Precision Matrix Estimation

Shihua Zhang

Fall 2019

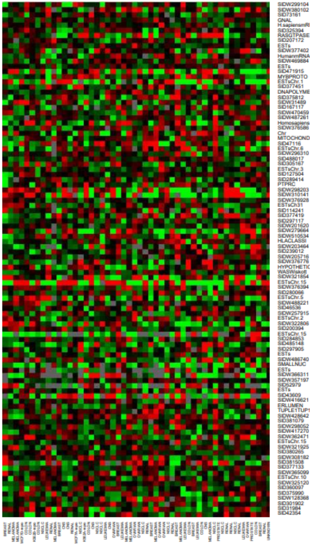
Contents

1. Introduction
2. Gaussian Graphical Model
3. Main Approaches
4. Variants
5. Discussion

Outline

1. Introduction
2. Gaussian Graphical Model
3. Main Approaches
4. Variants
5. Discussion

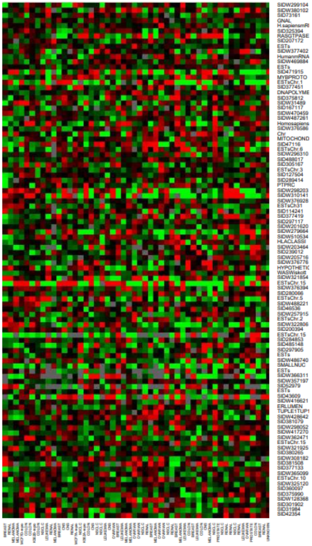
Network of Variables



Gene expression data are typical high-dimensional data.

- Tens of thousands of genes
- Only a few hundreds of samples

Network of Variables



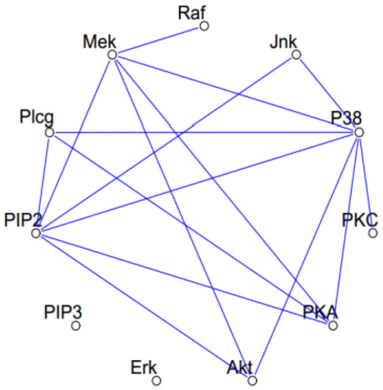
Gene expression data are typical high-dimensional data.

- Tens of thousands of genes
- Only a few hundreds of samples

Network of Variables

- Covariance matrix
- Inverse covariance matrix

Undirected Graphical Model



Consider the simplest undirect graphical model:

- Nodes correspond to random variables
- Edges represent conditional dependencies between the variables

Definition of Precision Matrix

In statistics, the covariance matrix: $\mathrm{C(i,j)}$ is the covariance between the i-th and j-th elements of a random vector.

$$\mathrm{K}_{\mathbf{XX}} = \begin{bmatrix} \mathrm{E}[(X_1 - \mathrm{E}[X_1])(X_1 - \mathrm{E}[X_1])] & \mathrm{E}[(X_1 - \mathrm{E}[X_1])(X_2 - \mathrm{E}[X_2])] & \cdots & \mathrm{E}[(X_1 - \mathrm{E}[X_1])(X_n - \mathrm{E}[X_n])] \\ \mathrm{E}[(X_2 - \mathrm{E}[X_2])(X_1 - \mathrm{E}[X_1])] & \mathrm{E}[(X_2 - \mathrm{E}[X_2])(X_2 - \mathrm{E}[X_2])] & \cdots & \mathrm{E}[(X_2 - \mathrm{E}[X_2])(X_n - \mathrm{E}[X_n])] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n - \mathrm{E}[X_n])(X_1 - \mathrm{E}[X_1])] & \mathrm{E}[(X_n - \mathrm{E}[X_n])(X_2 - \mathrm{E}[X_2])] & \cdots & \mathrm{E}[(X_n - \mathrm{E}[X_n])(X_n - \mathrm{E}[X_n])] \end{bmatrix}$$

Definition of Precision Matrix

In statistics, the covariance matrix: $\mathrm{C(i,j)}$ is the covariance between the i-th and j-th elements of a random vector.

$$\mathrm{K}_{\mathbf{XX}} = \begin{bmatrix} \mathrm{E}[(X_1 - \mathrm{E}[X_1])(X_1 - \mathrm{E}[X_1])] & \mathrm{E}[(X_1 - \mathrm{E}[X_1])(X_2 - \mathrm{E}[X_2])] & \cdots & \mathrm{E}[(X_1 - \mathrm{E}[X_1])(X_n - \mathrm{E}[X_n])] \\ \mathrm{E}[(X_2 - \mathrm{E}[X_2])(X_1 - \mathrm{E}[X_1])] & \mathrm{E}[(X_2 - \mathrm{E}[X_2])(X_2 - \mathrm{E}[X_2])] & \cdots & \mathrm{E}[(X_2 - \mathrm{E}[X_2])(X_n - \mathrm{E}[X_n])] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n - \mathrm{E}[X_n])(X_1 - \mathrm{E}[X_1])] & \mathrm{E}[(X_n - \mathrm{E}[X_n])(X_2 - \mathrm{E}[X_2])] & \cdots & \mathrm{E}[(X_n - \mathrm{E}[X_n])(X_n - \mathrm{E}[X_n])] \end{bmatrix}$$

The precision matrix (also known as concentration matrix) is the matrix inverse of the covariance matrix.

Challenges of Precision Matrix Estimation

The sample covariance based on the observed data is singular when the dimension is larger than the sample size.

- Many data are of high dimensionality (p $\gg$ n)
- Results will be unstable due to the limited number of samples
- The aggregation of a massive amount of estimation errors can lead to considerable adverse impacts on the estimation accuracy

Challenges of Precision Matrix Estimation

The sample covariance based on the observed data is singular when the dimension is larger than the sample size.

- Many data are of high dimensionality ($\mathrm{p} \gg \mathrm{n}$)
- Results will be unstable due to the limited number of samples
- The aggregation of a massive amount of estimation errors can lead to considerable adverse impacts on the estimation accuracy

Computational complexity is another challenging problem.

- The dimensions of a covariance matrix is $\mathrm{p} \times \mathrm{p}$.
- The computational complexity of estimating precision matrix directly is $\mathrm{O}(\mathrm{p}^3)$.

The estimation of large covariance and precision matrices is generally challenging.

Why Precision Matrix is Important?

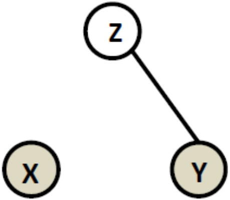
Estimation of a precision matrix is a fundamental problem in many areas of statistical analysis.

- high-dimensional linear discriminant analysis
- complex data visualization
- portfolio allocation
- graphical model

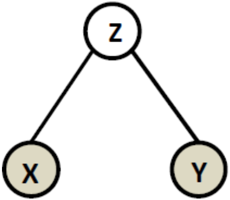
Outline

1. Introduction
2. Gaussian Graphical Model
3. Main Approaches
4. Variants
5. Discussion

Independence vs Conditional Independence



$\mathrm{P}(\mathrm{X} \cap \mathrm{Y}) = \mathrm{P}(\mathrm{X})\mathrm{P}(\mathrm{Y})$
$\mathrm{P}(\mathrm{X} \cap \mathrm{Y}|\mathrm{Z}) = \mathrm{P}(\mathrm{X}|\mathrm{Z})\mathrm{P}(\mathrm{Y}|\mathrm{Z})$

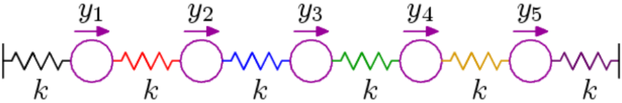


$\mathrm{P}(\mathrm{X} \cap \mathrm{Y}) \neq \mathrm{P}(\mathrm{X})\mathrm{P}(\mathrm{Y})$
$\mathrm{P}(\mathrm{X} \cap \mathrm{Y}|\mathrm{Z}) = \mathrm{P}(\mathrm{X}|\mathrm{Z})\mathrm{P}(\mathrm{Y}|\mathrm{Z})$

Independence: Two events A and B are independent if and only if $\mathrm{P}(\mathrm{A} \cap \mathrm{B}) = \mathrm{P}(\mathrm{A})\mathrm{P}(\mathrm{B})$.

Conditional independence: Two events A and B are independent if and only if $\mathrm{P}(\mathrm{A} \cap \mathrm{B}|\mathrm{C}) = \mathrm{P}(\mathrm{A}|\mathrm{C})\mathrm{P}(\mathrm{B}|\mathrm{C})$.

Covariance Matrix vs Precision Matrix



inverse-covariance matrix or covariance matrix?

$$\mathbf{K}^{-1} = \frac{k}{T}\begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \qquad \mathbf{K} = \frac{T}{k}\begin{bmatrix} 0.83 & 0.67 & 0.50 & 0.33 & 0.17 \\ 0.67 & 1.33 & 1.00 & 0.67 & 0.33 \\ 0.50 & 1.00 & 1.50 & 1.00 & 0.50 \\ 0.33 & 0.67 & 1.00 & 1.33 & 0.67 \\ 0.17 & 0.33 & 0.50 & 0.67 & 0.83 \end{bmatrix}$$

For the multivariate Gaussian distribution, precision matrix encodes the conditionally independent between variables.

Gaussian Graphical Model

Background:

Multivariate Gaussian Distribution

$$X \sim N_p(\mu, \Sigma)$$

- If Σ is positive definite, distribution has density on $\mathbb{R}^p$

$$f(x|\mu, \Sigma) = (2\pi)^{-p/2}(\det \Theta)^{1/2} e^{-(x-\mu)^T \Theta (x-\mu)/2}$$

  where $\Theta = \Sigma^{-1}$ is the precision matrix of the distribution.

- Conditional distribution:

$$\begin{gathered} X_1|X_2 \sim \mathcal{N}\left(\mu_{1|2}, \Sigma_{1|2}\right) \\ \text{where: } \mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \\ \Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \end{gathered}$$

Gaussian Graphical Model

Question:

Precision matrix ⇔ conditional dependencies?

Suppose that observations $x_1, x_2, \ldots, x_n \in \mathbb{R}^P$ are i.i.d. $N_p(\mu, \Sigma)$ where $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{\Sigma}$ is a $p \times p$ positive definite matrix.

- Partition $X = (Z, Y)$ where $Z = (X_1, \ldots, X_{p-1})$ and $Y = X_p$.
- Partitioned Σ and Θ as

$$\boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_{ZZ} & \sigma_{ZY} \\ \sigma_{ZY}^T & \sigma_{YY} \end{pmatrix}, \quad \boldsymbol{\Theta} = \begin{pmatrix} \Theta_{ZZ} & \theta_{ZY} \\ \theta_{ZY}^T & \theta_{YY} \end{pmatrix}$$

Gaussian Graphical Model

Because $\Theta = \Sigma^{-1}$ standards formulas for partitioned inverses given:

- $\theta_{YY} = (\sigma_{YY} - \sigma_{ZY}^{T}\Sigma_{ZZ}^{-1}\sigma_{ZY})^{-1} \succ 0$
- $\theta_{ZY} = -\theta_{YY} \cdot \Sigma_{ZZ}^{-1}\sigma_{ZY}$

And the conditional distribution of Y given Z:

$$Y(Z = z) \sim N\left(\mu_Y + (z - \mu_Z)^T \Sigma_{ZZ}^{-1}\sigma_{ZY}, \sigma_{YY} - \sigma_{ZY}^{T}\Sigma_{ZZ}^{-1}\sigma_{ZY}\right)$$

Gaussian Graphical Model

Because $\Theta = \Sigma^{-1}$ standards formulas for partitioned inverses given:

- $\theta_{YY} = (\sigma_{YY} - \sigma_{ZY}^{T}\Sigma_{ZZ}^{-1}\sigma_{ZY})^{-1} \succ 0$
- $\theta_{ZY} = -\theta_{YY} \cdot \Sigma_{ZZ}^{-1}\sigma_{ZY}$

And the conditional distribution of Y given Z:

$$Y(Z = z) \sim N\left(\mu_Y + (z - \mu_Z)^T \Sigma_{ZZ}^{-1}\sigma_{ZY}, \sigma_{YY} - \sigma_{ZY}^{T}\Sigma_{ZZ}^{-1}\sigma_{ZY}\right)$$

Comparing these two equations,

- If $(\theta_{ZY})_i = 0$, then $Y(Z = z)$ is nothing to do with the value of z_i

Gaussian Graphical Model

Because $\Theta = \Sigma^{-1}$ standards formulas for partitioned inverses given:

- $\theta_{YY} = (\sigma_{YY} - \sigma_{ZY}^{T}\Sigma_{ZZ}^{-1}\sigma_{ZY})^{-1} \succ 0$
- $\theta_{ZY} = -\theta_{YY} \cdot \Sigma_{ZZ}^{-1}\sigma_{ZY}$

And the conditional distribution of Y given Z:

$$Y(Z = z) \sim N\left(\mu_Y + (z - \mu_Z)^T \Sigma_{ZZ}^{-1}\sigma_{ZY}, \sigma_{YY} - \sigma_{ZY}^{T}\Sigma_{ZZ}^{-1}\sigma_{ZY}\right)$$

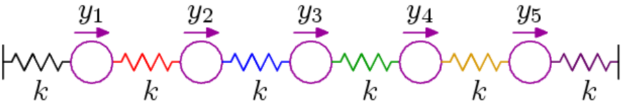
Comparing these two equations,

- If $(\theta_{ZY})_i = 0$, then $Y(Z = z)$ is nothing to do with the value of z_i

$$\theta_{i,p} = 0 \Leftrightarrow i \perp p|V\backslash\{i,p\} \Leftrightarrow \text{Edge } (i,p) \text{ doesn't exist}$$

Gaussian Graphical Model

We can construct a graphical model by estimating the precision matrix.



inverse-covariance matrix or covariance matrix?

$$\mathbf{K}^{-1} = \frac{k}{T}\begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \qquad \mathbf{K} = \frac{T}{k}\begin{bmatrix} 0.83 & 0.67 & 0.50 & 0.33 & 0.17 \\ 0.67 & 1.33 & 1.00 & 0.67 & 0.33 \\ 0.50 & 1.00 & 1.50 & 1.00 & 0.50 \\ 0.33 & 0.67 & 1.00 & 1.33 & 0.67 \\ 0.17 & 0.33 & 0.50 & 0.67 & 0.83 \end{bmatrix}$$

Outline

1. Introduction

2. Gaussian Graphical Model

3. Main Approaches
   - Sparsity
   - Penalized Likelihood Methods
   - Column-by-column Estimation Methods
   - CLIME

4. Variants

5. Discussion

1. Introduction

2. Gaussian Graphical Model

3. Main Approaches
   - Sparsity
   - Penalized Likelihood Methods
   - Column-by-column Estimation Methods
   - CLIME

4. Variants

5. Discussion

Sparsity

The key assumption of the precision matrix estimation is that the target matrix of interest is sparse (i.e., many entries are either zeros or nearly so).

- Question 1: why do we need a sparse solution?

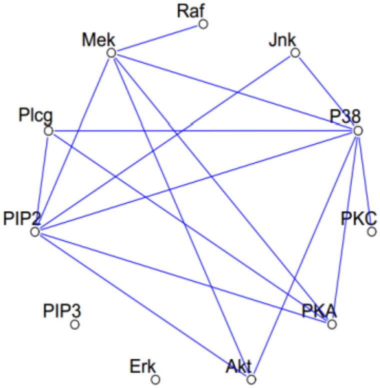
Sparsity

The key assumption of the precision matrix estimation is that the target matrix of interest is sparse (i.e., many entries are either zeros or nearly so).

- Question 1: why do we need a sparse solution?
  - feature/variable selection
  - better to interpret the data
  - shrink the size of model
  - computational savings
  - discourage overfitting

Sparsity

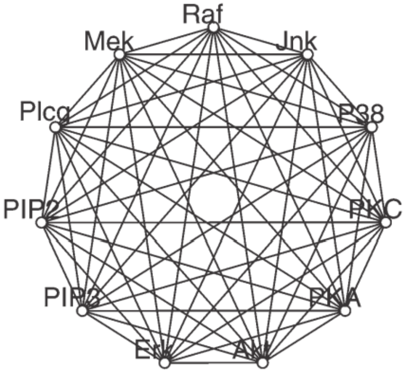
A real network is a set of links with direct dependencies.



- Spare and structured

Sparsity

Estimated network without sparsity constraint.



- Dense and meaningless

Sparsity

The key assumption of precision matrix estimation is that the target matrix of interest is sparse (i.e., many entries are either zeros or nearly so).

- Question 1: why do we need a sparse solution?

Sparsity

The key assumption of precision matrix estimation is that the target matrix of interest is sparse (i.e., many entries are either zeros or nearly so).

- Question 1: why do we need a sparse solution?
    - feature/variable selection
    - better to interpret the data
    - shrink the size of model
    - computational savings
    - discourage overfitting
- Question 2: how to achieve a sparse solution?

Sparsity

Take the linear regression as an example $(\mathrm{f(x) = w^T * x + b})$.

Subset selection: $\mathrm{l_0}$-norm regularization

$$\min \mathcal{L} = \sum_{\mathrm{i=1}}^{\mathrm{N}} \left|\mathrm{y_i - f(x_i)}\right|^2 + \frac{\lambda}{2}\|\mathrm{w}\|_0$$

where

$$\|\mathrm{w}\|_0 = \#\left(\mathrm{i}|\mathrm{w_i} \neq 0\right)$$

Sparsity

Take the linear regression as an example ($f(x) = w^T * x + b$).

Subset selection: l_0-norm regularization

$$\min \mathcal{L} = \sum_{i=1}^{N} |y_i - f(x_i)|^2 + \frac{\lambda}{2} \|w\|_0$$

where

$$\|w\|_0 = \#(i | w_i \neq 0)$$

- sparse solution
- but non-convex and hard to optimize

Sparsity

Ridge: l_2-norm regularization

$$\min \mathcal{L} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \left|\mathrm{y}_\mathrm{i} - \mathrm{f}\left(\mathrm{x}_\mathrm{i}\right)\right|^2 + \frac{\lambda}{2}\|\mathrm{w}\|_2^2$$

Its equivalent form is (constrained optimization):

$$\begin{gathered} \min \mathcal{L} = \textstyle\sum_{\mathrm{i}=1}^{\mathrm{N}} \left|\mathrm{y}_\mathrm{i} - \mathrm{f}\left(\mathrm{x}_\mathrm{i}\right)\right|^2 \\ \mathrm{s.t.} \|\mathrm{w}\|_2^2 \leqslant \mathrm{C} \end{gathered}$$

Sparsity

Ridge: l_2-norm regularization

$$\min \mathcal{L} = \sum_{i=1}^{N} |y_i - f(x_i)|^2 + \frac{\lambda}{2}\|w\|_2^2$$

Its equivalent form is (constrained optimization):

$$\begin{gathered} \min \mathcal{L} = \textstyle\sum_{i=1}^{N} |y_i - f(x_i)|^2 \\ s.t. \|w\|_2^2 \leqslant C \end{gathered}$$

- convex but generate a non-sparse solution (values close to zeros)

Sparsity

Lasso: l_1-norm regularization

$$\min \mathcal{L} = \sum_{i=1}^{N} |y_i - f(x_i)|^2 + \frac{\lambda}{2}\|w\|_1$$

Its equivalent form is (constrained optimization):

$$\begin{gathered}\min \mathcal{L} = \textstyle\sum_{i=1}^{N} |y_i - f(x_i)|^2 \\ \text{s.t.} \|w\|_1 \leqslant C\end{gathered}$$

Sparsity

Lasso: l_1-norm regularization

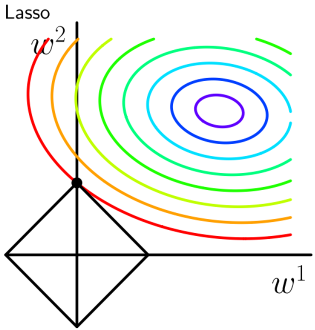
$$\min \mathcal{L} = \sum_{i=1}^{N} |y_i - f(x_i)|^2 + \frac{\lambda}{2}\|w\|_1$$

Its equivalent form is (constrained optimization):

$$\begin{gathered} \min \mathcal{L} = \textstyle\sum_{i=1}^{N} |y_i - f(x_i)|^2 \\ \text{s.t.} \|w\|_1 \leqslant C \end{gathered}$$

- convex
- sparse solution

Why Lasso Leads to Sparsity?



l_1-norm regularization helps to generate sparse estimation.

1 Introduction

2 Gaussian Graphical Model

3 Main Approaches
- Sparsity
- Penalized Likelihood Methods
- Column-by-column Estimation Methods
- CLIME

4 Variants

5 Discussion

Penalized Likelihood Methods

One of the most commonly used approaches to estimate sparse precision matrices is the penalized maximum likelihood.

- When $x_1, x_2, \ldots, x_n \in \mathbb{R}^P$ are i.i.d. $N(0, \Sigma)$

$$f(x_1, \ldots, x_n | \mu, \Sigma) = (2\pi)^{-d/2} (\det \Theta)^{1/2} e^{-\sum_i \mathrm{tr}\left((x_i-\mu)^T (x_i-\mu)\Theta/2\right)}$$

- The negative Gaussian log-likelihood function is given by

$$\ell(\boldsymbol{\Theta}) = \mathrm{tr}(S\boldsymbol{\Theta}) - \log|\boldsymbol{\Theta}|$$

- Penalized likelihood method:

$$\widehat{\Theta} = \underset{\boldsymbol{\Theta}}{\mathrm{argmin}} \left\{ \mathrm{tr}(S\boldsymbol{\Theta}) - \log|\boldsymbol{\Theta}| + \sum_{i \neq j} P\left(|\theta_{ij}|\right) \right\}$$

Penalized Likelihood Methods

One of the commonly used convex penalties is the l_1 penalty [Meinshausen et al., 2006].

$$\widehat{\Theta} = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} \{ \operatorname{tr}(S\boldsymbol{\Theta}) - \log|\boldsymbol{\Theta}| + \lambda \|\boldsymbol{\Theta}\|_1 \}$$

Solutions

- Interior-point optimization methods [Banerjee et al., 2008]
- Graphical Lasso (most popular) [Friedman et al., 2008]
- Alternating direction method of multipliers [Boyd et al., 2011]
- QUIC [Hsieh et al., 2014]

Trick and variants

- Block screen [Mazumder and Hastie, 2012]
- D-trace loss [Zhang and Zou, 2014]

Graphical Lasso

Graphical Lasso [Friedman et al., 2008]

- Problem: maximize the l_1 penalized log-likelihood:

$$\log\det\boldsymbol{\Theta} - \mathrm{tr}(S\boldsymbol{\Theta}) - \lambda\|\boldsymbol{\Theta}\|_1$$

- Optimization by blockwise coordinate descent.
- Fast: it solves a 1000-variable problem (about 500,000 parameters) in at most one minute.

Optimization of Graphical Lasso

- Graphical Lasso considers estimation of Σ (rather than Σ^{-1})
- Objective function:

$$\log\det\Sigma^{-1} - \mathrm{tr}(\mathbf{S}\Sigma^{-1}) - \lambda\|\Sigma^{-1}\|_1$$

- Let W be the estimate of Σ and partitioning W and S

$$\mathrm{W} = \begin{pmatrix} \mathrm{W}_{11} & \mathrm{w}_{12} \\ \mathrm{w}_{12}^{\mathrm{T}} & \mathrm{w}_{22} \end{pmatrix}, \quad \mathrm{S} = \begin{pmatrix} \mathrm{S}_{11} & \mathrm{s}_{12} \\ \mathrm{s}_{12}^{\mathrm{T}} & \mathrm{s}_{22} \end{pmatrix}$$

- Blockwise coordinate descent: Fix W_{11} to optimize w_{12}

Optimization of Graphical Lasso

Equivalence problem [Banerjee et al., 2008]

When fix W_{11} to optimize w_{12},

$$\operatorname*{argmax}_{\mathrm{w}_{12}} \left\{ \log\det\Sigma^{-1} - \mathrm{tr}(\mathbf{S}\Sigma^{-1}) - \lambda\|\Sigma^{-1}\|_1 \right\}$$

equals solving a Lasso problem

$$\min_{\beta} \left\{ \frac{1}{2} \left\| \mathrm{W}_{11}^{1/2}\beta - \mathrm{W}_{11}^{-1/2}\mathrm{s}_{12} \right\|^2 + \lambda\|\beta\|_1 \right\}$$

Optimization of Graphical Lasso

Proof: The subgradient equation of the log-likelihood:

$$w_{12} - s_{12} - \lambda \cdot \gamma_{12} = 0$$

where γ_{12} is the derivative of l_1-norm.
For the Lasso problem

$$\min_{\beta} \left\{ \frac{1}{2} \left\| W_{11}^{1/2} \beta - W_{11}^{-1/2} s_{12} \right\|^2 + \lambda \|\beta\|_1 \right\}$$

its subgradient equation

$$W_{11}\beta - s_{12} + \lambda \cdot v = 0$$

For (w_{12}, γ_{12}) soloves log-likelihood, then $\beta = W_{11}^{-1} w_{12}$ and $v = -\gamma_{12}$ solves the Lasso problem.

Optimization of Graphical Lasso

Graphical Lasso algorithm

1. Start with $W = S + \rho I$. The diagonal of W remains unchanged in what follows.
2. For each $j = 1, 2, \ldots p, 1, 2, \ldots p, \ldots,$ solve the Lasso problem, which takes as input the inner products W_{11} and s_{12}. This gives a $p - 1$ vector solution β. Fill in the corresponding row and column of W using $w_{12} = W_{11}\beta$.
3. Continue until convergence. Obtain estimation of Σ: $\Sigma = W$

Optimization of Graphical Lasso

After estimate $\Sigma = W$, we can recover $\Theta = W^{-1}$ relatively cheaply.

$$\begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix} \begin{pmatrix} \Theta_{11} & \theta_{12} \\ \theta_{12}^T & \theta_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0^T & 1 \end{pmatrix}$$

So

$$\begin{aligned} \theta_{12} &= -W_{11}^{-1} w_{12} \theta_{22} \\ \theta_{22} &= 1/\left(w_{22} - w_{12}^T W_{11}^{-1} w_{12}\right) \end{aligned}$$

But since $\beta = W_{11}^{-1} w_{12}$

$$\begin{aligned} \theta_{22} &= 1/\left(w_{22} - w_{12}^T \beta\right) \\ \theta_{12} &= -\beta \theta_{22} \end{aligned}$$

Using the stored the coefficients β, we can compute Θ cheaply after convergence.

Graphical Lasso

Graphical Lasso allows each inverse covariance element to be penalized differently,

$$\log\det\Theta - \mathrm{tr}(S\Theta) - \|\Theta * P\|_1$$

where $P = \{\rho_{jk}\}$ with $\rho_{jk} = \rho_{kj}$

Changes the Lasso problem to

$$\min_{\beta}\left\{\frac{1}{2}\left\|W_{11}^{1/2}\beta - W_{11}^{-1/2}s_{12}\right\|^2 + P_{12}\|\beta\|_1\right\}$$

Time Comparison

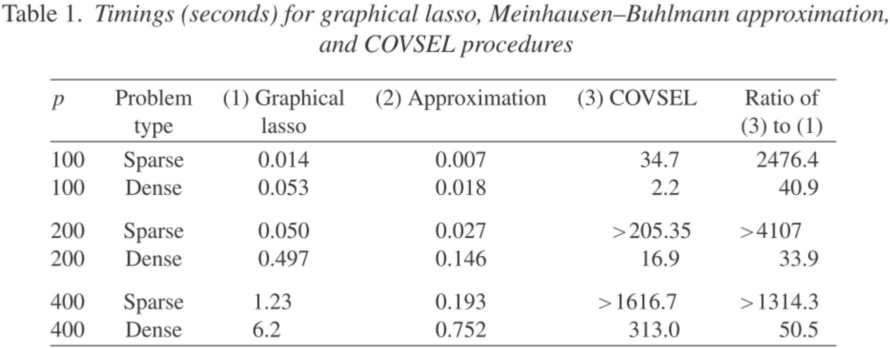
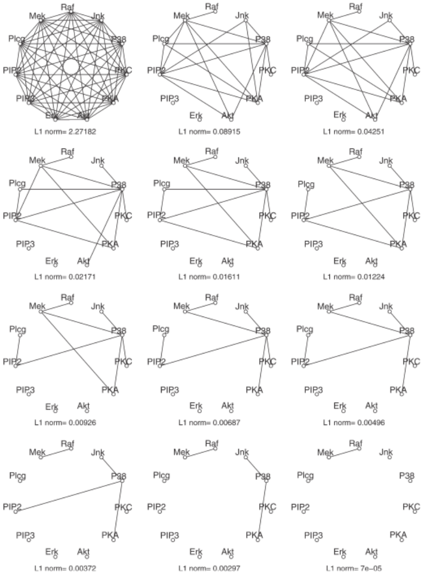


Table 1. *Timings (seconds) for graphical lasso, Meinhausen–Buhlmann approximation, and COVSEL procedures*

p	Problem type	(1) Graphical lasso	(2) Approximation	(3) COVSEL	Ratio of (3) to (1)
100	Sparse	0.014	0.007	34.7	2476.4
100	Dense	0.053	0.018	2.2	40.9
200	Sparse	0.050	0.027	>205.35	>4107
200	Dense	0.497	0.146	16.9	33.9
400	Sparse	1.23	0.193	>1616.7	>1314.3
400	Dense	6.2	0.752	313.0	50.5

Different Penalty Parameters



Penalized Likelihood Methods

One of the commonly used convex penalties is the l_1 penalty [Meinshausen et al., 2006].

$$\widehat{\Theta} = \underset{\boldsymbol{\Theta}}{\mathsf{argmin}}\{\mathsf{tr}(\mathrm{S}\boldsymbol{\Theta}) - \mathsf{log}\,|\boldsymbol{\Theta}| + \lambda\|\boldsymbol{\Theta}\|_1\}$$

Solutions

- Interior-point optimization methods [Banerjee et al., 2008]
- Graphical Lasso (most popular) [Friedman et al., 2008]
- Alternating direction method of multipliers (ADMM) [Boyd et al., 2011]
- QUIC [Hsieh et al., 2014]

Trick and variants

- Block screen [Mazumder and Hastie, 2012]
- D-trace loss [Zhang and Zou, 2014]

ADMM

- ADMM is a method with good robustness of method of multipliers, which can support decomposition.

ADMM problem form (with f, g convex)

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

The augmented Lagrangian:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)\|Ax + Bz - c\|_2^2$$

ADMM:

$$\begin{aligned} x^{k+1} &:= \mathrm{argmin}_x\, L_\rho\left(x, z^k, y^k\right) \\ z^{k+1} &:= \mathrm{argmin}_z\, L_\rho\left(x^{k+1}, z, y^k\right) \\ y^{k+1} &:= y^k + \rho\left(Ax^{k+1} + Bz^{k+1} - c\right) \end{aligned}$$

ADMM with scaled dual variables

Combine linear and quadratic terms in the augmented Lagrangian:

$$\begin{aligned} L_\rho(x, z, y) &= f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)\|Ax + Bz - c\|_2^2 \\ &= f(x) + g(z) + (\rho/2)\|Ax + Bz - c + u\|_2^2 + \text{ const.} \end{aligned}$$

with $u^k = (1/\rho)y^k$

ADMM (scaled dual form):

$$\begin{aligned} x^{k+1} &:= \underset{x}{\operatorname{argmin}} \left(f(x) + (\rho/2) \left\|Ax + Bz^k - c + u^k\right\|_2^2 \right) \\ z^{k+1} &:= \underset{z}{\operatorname{argmin}} \left(g(z) + (\rho/2) \left\|Ax^{k+1} + Bz - c + u^k\right\|_2^2 \right) \\ u^{k+1} &:= u^k + \left(Ax^{k+1} + Bz^{k+1} - c\right) \end{aligned}$$

Convergence

Assume (very little!)

- f, g convex, closed, proper
- L_0 has a saddle point

Then ADMM converges:

- Residual convergence: $\mathrm{Ax}^{\mathrm{k}} + \mathrm{Bz}^{\mathrm{k}} - \mathrm{c} \to 0$
- Objective convergence: $\mathrm{f}\left(\mathrm{x}^{\mathrm{k}}\right) + \mathrm{g}\left(\mathrm{z}^{\mathrm{k}}\right) \to \mathrm{p}^{\star}$
- Dual convergence: $\mathrm{u}^{\mathrm{k}} \to \mathrm{u}^{\star}$

Convergence rate: not known in general, theory is currently being developed, e.g., in Hong and Luo (2012), Nishihara et al. (2015). Roughly, it behaves like a first-order method (or a bit faster).

ADMM for Precision Matrix Estimation

Problem form

$$\widehat{\Theta} = \underset{\Theta}{\mathrm{argmin}}\{\mathrm{tr}(S\Theta) - \log|\Theta| + \lambda\|\Theta\|_1\}$$

ADMM for Precision Matrix Estimation

Problem form

$$\widehat{\Theta} = \underset{\Theta}{\mathsf{argmin}}\{\mathsf{tr}(S\Theta) - \log|\Theta| + \lambda\|\Theta\|_1\}$$

ADMM form

$$\begin{array}{ll} \text{minimize} & \mathsf{tr}(SX) - \log\det X + \lambda\|Z\|_1 \\ \text{subject to} & X - Z = 0 \end{array}$$

ADMM for Precision Matrix Estimation

Problem form

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \{\operatorname{tr}(S\Theta) - \log|\Theta| + \lambda\|\Theta\|_1\}$$

ADMM form

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}(SX) - \log\det X + \lambda\|Z\|_1 \\ \text{subject to} & X - Z = 0 \end{array}$$

The augmented Lagrangian:

$$L = \operatorname{tr}(SX) - \log\det X + \lambda\|Z\|_1 + (\rho/2)\|X - Z + U\|_2^2$$

ADMM for Precision Matrix Estimation

ADMM form

$$L = \mathsf{tr}(SX) - \mathsf{log\,det}\, X + \lambda\|Z\|_1 + (\rho/2)\|X - Z + U\|_2^2$$

ADMM:

$$\begin{aligned}
X^{k+1} &:= \underset{X}{\mathsf{argmin}} \left(\mathsf{tr}(SX) - \mathsf{log\,det}\, X + (\rho/2)\left\|X - Z^k + U^k\right\|_F^2\right) \\
Z^{k+1} &:= \underset{Z}{\mathsf{argmin}} \left((\rho/2)\left\|X - Z^k + U^k\right\|_F^2 + \lambda\|Z\|_1\right) \\
U^{k+1} &:= U^k + \left(X^{k+1} - Z^{k+1}\right)
\end{aligned}$$

Update for X

Problem

$$X^{k+1} = \operatorname*{argmin}_X \left(\mathsf{tr}(SX) - \mathsf{log\,det}\, X + (\rho/2) \left\| X - Z^k + U^k \right\|_F^2 \right)$$

Differentiating with respect to X, the minimum solves:

$$S - X^{-1} + \rho(X - Z^k + U^k) = 0$$

that is

$$\rho X - X^{-1} = \rho(Z^k - U^k) - S$$

which is a eigenvalue problem.

Update for X

$$\rho X - X^{-1} = \rho(Z^k - U^k) - S$$

Compute the eigendecomposition:

$$\rho\left(Z^k - U^k\right) - S = Q\Lambda Q^T$$

Then the eigendecomposition of X^{k+1} is QXQ^T, where X is a diagonal matrix and $\rho X - X^{-1} = \Lambda$
So $X^{k+1} := QXQ^T$ with

$$X_{ii} = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4\rho}}{2\rho}$$

Cost of X-update is an eigendecomposition.

Time Cost

For a 1000×1000 Σ^{-1} with 10^4 nonzeros

- Graphical Lasso (Fortran): 20 seconds – 3 minutes
- ADMM (Matlab): 3 – 10 minutes

It is flexible to extend (such as adding other convex penalty in the log-likelihood function).

Penalized Likelihood Methods

One of the commonly used convex penalties is the l_1 penalty [Meinshausen et al., 2006].

$$\widehat{\Theta} = \underset{\boldsymbol{\Theta}}{\mathsf{argmin}} \{ \mathsf{tr}(S\boldsymbol{\Theta}) - \mathsf{log}\,|\boldsymbol{\Theta}| + \lambda \|\boldsymbol{\Theta}\|_1 \}$$

Solutions

- Interior-point optimization methods [Banerjee et al., 2008]
- Graphical Lasso (most popular) [Friedman et al., 2008]
- Alternating direction method of multipliers (ADMM) [Boyd et al., 2011]
- QUIC [Hsieh et al., 2014]

Trick and variants

- Block screen [Mazumder and Hastie, 2012]
- D-trace loss [Zhang and Zou, 2014]

QUIC

- Existing methods are first-order iterative methods that mainly use gradient information at each step.
- Disadvantage: they are at most linear rates of convergence

Question

Can we achieve superlinearly rate of convergence by considering second-order methods?

QUIC (QUadratic approximation of Inverse Covariance matrices) performs Newton steps and achieve superlinearly rate of convergence.

QUIC

- Existing methods are first-order iterative methods that mainly use gradient information at each step.
- Disadvantage: they are at most linear rates of convergence

Question

Can we achieve superlinearly rate of convergence by considering second-order methods?

QUIC (QUadratic approximation of Inverse Covariance matrices) performs Newton steps and achieve superlinearly rate of convergence.

Difficulties: second-order methods at least in part use the Hessian of the objective function.

- This is too expensive for high-dimensional problem.

QUIC

- Existing methods are first-order iterative methods that mainly use gradient information at each step.
- Disadvantage: they are at most linear rates of convergence

Question

Can we achieve superlinearly rate of convergence by considering second-order methods?

QUIC (QUadratic approximation of Inverse Covariance matrices) performs Newton steps and achieve superlinearly rate of convergence.

Difficulties: second-order methods at least in part use the Hessian of the objective function.

- This is too expensive for high-dimensional problem.

QUIC reduces the computational cost of a coordinate descent update from the naive $\mathrm{O}(\mathrm{p}^2)$ to $\mathrm{O}(\mathrm{p})$ complexity.

The Newton Direction

The second-order Taylor expansion of a function f around x^{k} is

$$\mathrm{f}(\mathrm{x}) \approx \mathrm{f}\left(\mathrm{x}^{\mathrm{k}}\right) + \nabla \mathrm{f}\left(\mathrm{x}^{\mathrm{k}}\right)^{\mathrm{T}}\left(\mathrm{x}-\mathrm{x}^{\mathrm{k}}\right) + \frac{1}{2}\left(\mathrm{x}-\mathrm{x}^{\mathrm{k}}\right)^{\mathrm{T}} \mathrm{H}\left(\mathrm{x}-\mathrm{x}^{\mathrm{k}}\right)$$

where H is the Hessian matrix

$$\mathrm{H} = \nabla^2 \mathrm{f}\left(\mathrm{x}^{\mathrm{k}}\right) = \begin{bmatrix} \frac{\partial^2 \mathrm{f}(\mathrm{x}^{\mathrm{k}})}{\partial \mathrm{x}_1^2} & \cdots & \frac{\partial^2 \mathrm{f}(\mathrm{x}^{\mathrm{k}})}{\partial \mathrm{x}_1 \partial \mathrm{x}_\mathrm{n}} \\ \vdots & & \vdots \\ \frac{\partial^2 \mathrm{f}(\mathrm{x}^{\mathrm{k}})}{\partial \mathrm{x}_\mathrm{n} \partial \mathrm{x}_1} & \cdots & \frac{\partial^2 \mathrm{f}(\mathrm{x}^{\mathrm{k}})}{\partial \mathrm{x}_\mathrm{n}^2} \end{bmatrix}$$

The Newton direction is the solution of $\Delta \mathrm{x} = \mathrm{x} - \mathrm{x}^{\mathrm{k}}$ for the second-order expansion

$$\mathrm{D}^{\mathrm{k}} = (\mathrm{H})^{-1} \nabla \mathrm{f}\left(\mathrm{x}^{\mathrm{k}}\right)$$

QUIC

$$\mathrm{X}^* = \mathsf{arg}\min_{\mathrm{X}\succ 0}\{-\mathsf{log}\,\mathsf{det}\,\mathrm{X} + \mathsf{tr}(\mathrm{SX}) + \|\mathrm{X}\|_1\} = \mathsf{arg}\min_{\mathrm{X}\succ 0}\mathrm{f}(\mathrm{X})$$

Partition $\mathrm{f}(\mathrm{X}) = \mathrm{g}(\mathrm{X}) + \mathrm{h}(\mathrm{X})$, where

$$\mathrm{g}(\mathrm{X}) = -\mathsf{log}\,\mathsf{det}\,\mathrm{X} + \mathsf{tr}(\mathrm{SX}) \quad \text{and} \quad \mathrm{h}(\mathrm{X}) = \|\mathrm{X}\|_1$$

considering the second-order Taylor expansion of the smooth component $\mathrm{g}(\mathrm{X})$

$$\bar{\mathrm{g}}_{\mathrm{X}_\mathrm{t}}(\Delta) \equiv \mathrm{g}\left(\mathrm{X}_\mathrm{t}\right) + \mathsf{vec}\left(\nabla \mathrm{g}\left(\mathrm{X}_\mathrm{t}\right)\right)^{\mathrm{T}} \mathsf{vec}(\Delta) + \frac{1}{2}\mathsf{vec}(\Delta)^{\mathrm{T}} \nabla^2 \mathrm{g}\left(\mathrm{X}_\mathrm{t}\right) \mathsf{vec}(\Delta)$$

QUIC

The Newton direction D_t^* for the entire objective $f(X)$

$$D_t^* = \mathsf{arg}\,\min_{\Delta} \left\{ \bar{g}_{X_t}(\Delta) + h\left(X_t + \Delta\right) \right\}$$

Note that it can be rewritten as a standard Lasso regression problem

$$\mathsf{arg}\,\min_{\Delta} \frac{1}{2} \left\| H^{\frac{1}{2}}\,\mathsf{vec}(\Delta) + H^{-\frac{1}{2}} b \right\|_2^2 + \left\| X_t + \Delta \right\|_1$$

where $H = \nabla^2 g\left(X_t\right)$ and $b = \mathsf{vec}\left(\nabla g\left(X_t\right)\right)$

QUIC

Gradient and Hessian for the log-likelihood g(x) [Boyd and Vandenberghe, 2004]

$$\nabla g(X) = S - X^{-1} \quad \text{and} \quad \nabla^2 g(X) = X^{-1} \otimes X^{-1}$$

Proof: For Z near X, and $\Delta X = Z - X$

$$\begin{aligned} Z^{-1} &= (X + \Delta X)^{-1} \\ &= \left(X^{1/2}\left(I + X^{-1/2}\Delta X X^{-1/2}\right)X^{1/2}\right)^{-1} \\ &= X^{-1/2}\left(I + X^{-1/2}\Delta X X^{-1/2}\right)^{-1} X^{-1/2} \\ &\approx X^{-1/2}\left(I - X^{-1/2}\Delta X X^{-1/2}\right) X^{-1/2} \\ &= X^{-1} - X^{-1}\Delta X X^{-1} \end{aligned}$$

using the first-order approximation $(I + A)^{-1} \approx I - A$ (valid for A small).

QUIC

We have $\mathsf{tr}\left(X_t^{-1}\Delta X_t^{-1}\Delta\right) = \mathsf{vec}(\Delta)^T\left(X_t^{-1}\otimes X_t^{-1}\right)\mathsf{vec}(\Delta)$. So, the approximation of $g(x)$ can be rewritten as

$$\overline{g}_{X_t}(\Delta) = -\mathsf{log\,det}\,X_t + \mathsf{tr}\left(SX_t\right) + \mathsf{tr}\left(\left(S - W_t\right)^T\Delta\right) + \frac{1}{2}\,\mathsf{tr}\left(W_t\Delta W_t\Delta\right)$$

where $W_t = X_t^{-1}$.

QUIC

We have $\mathsf{tr}\left(X_t^{-1}\Delta X_t^{-1}\Delta\right) = \mathsf{vec}(\Delta)^T\left(X_t^{-1}\otimes X_t^{-1}\right)\mathsf{vec}(\Delta)$. So, the approximation of $g(x)$ can be rewritten as

$$\bar{g}_{X_t}(\Delta) = -\log\det X_t + \mathsf{tr}\left(SX_t\right) + \mathsf{tr}\left(\left(S - W_t\right)^T\Delta\right) + \frac{1}{2}\mathsf{tr}\left(W_t\Delta W_t\Delta\right)$$

where $W_t = X_t^{-1}$.

The Newton direction can be solved by a ordinary Lasso problem which requires $O(p^2)$ for each element (coordinate descent).

- $O(p^4)$ for computing the Newton direction
- not enough

The Key Step of QUIC

Use D to denote the current iterate approximating the Newton direction and D′ for the updated direction.

- Consider the coordinate descent update for the variable X_{ij}, with $i < j$ that preserves symmetry: $D' = D + \mu\left(e_i e_j^T + e_j e_i^T\right)$
- The solution of the one-variable problem is

$$\arg\min_{\mu} \bar{g}\left(D + \mu\left(e_i e_j^T + e_j e_i^T\right)\right) + 2\lambda_{ij}\left|X_{ij} + D_{ij} + \mu\right|$$

- Omit the terms not dependent on μ

$$\mathrm{tr}\left(\left(S - W_t\right)^T D'\right) \propto 2\mu\left(S_{ij} - W_{ij}\right)$$

$$\mathrm{tr}\left(WD'WD'\right) = \mathrm{tr}(WDWD) + 4\mu w_i^T D w_j + 2\mu^2\left(W_{ij}^2 + W_{ii}W_{jj}\right)$$

The Key Step of QUIC

So the one-variable problem is transformed into minimization of the following function of μ

$$\frac{1}{2}\left(W_{ij}^2 + W_{ii}W_{jj}\right)\mu^2 + \left(S_{ij} - W_{ij} + w_i^T D w_j\right)\mu + \lambda_{ij}\left|X_{ij} + D_{ij} + \mu\right|$$

Let $a = W_{ij}^2 + W_{ii}W_{jj}$, $b = S_{ij} - W_{ij} + w_i^T D w_j$, and $c = X_{ij} + D_{ij}$, the minimum is achieved for

$$\mu = -c + \mathcal{S}\left(c - b/a, \lambda_{ij}/a\right)$$

where

$$\mathcal{S}(z, r) = \mathsf{sign}(z)\,\mathsf{max}\{|z| - r, 0\}$$

a and c are easy to compute. The main computational cost is $w_i^T D w_j$

The Key Step of QUIC

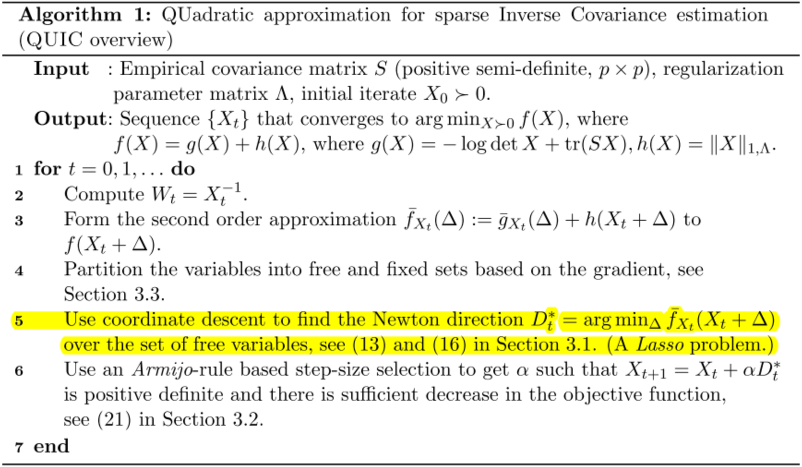
Calculating $\mathrm{w}_i^T D \mathrm{w}_j$ requires $O(p^2)$ times.

- Instead, we maintain matrix $U = DW$, and then compute $\mathrm{w}_i^T D \mathrm{w}_j$ by $\mathrm{w}_i^T \mathrm{u}_j$ using $O(p)$ flops.
- The maintain of $U = DW$ needs to update 2p elements

  $$\mathrm{u}_{i\cdot} \leftarrow \mathrm{u}_{i\cdot} + \mu \mathrm{w}_{j\cdot}$$
  $$\mathrm{u}_{j\cdot} \leftarrow \mathrm{u}_{j\cdot} + \mu \mathrm{w}_{i\cdot}$$

  where $\mathrm{u}_{i\cdot}$ refers to the i-th row vector of U.

Workflow of QUIC



Algorithm 1: QUadratic approximation for sparse Inverse Covariance estimation (QUIC overview)

Input : Empirical covariance matrix S (positive semi-definite, $p \times p$), regularization parameter matrix Λ, initial iterate $X_0 \succ 0$.

Output: Sequence $\{X_t\}$ that converges to $\arg\min_{X \succ 0} f(X)$, where $f(X) = g(X) + h(X)$, where $g(X) = -\log\det X + \operatorname{tr}(SX), h(X) = \|X\|_{1,\Lambda}$.

```
for $t = 0, 1, \ldots$ do
    Compute $W_t = X_t^{-1}$.
    Form the second order approximation $\bar{f}_{X_t}(\Delta) := \bar{g}_{X_t}(\Delta) + h(X_t + \Delta)$ to $f(X_t + \Delta)$.
    Partition the variables into free and fixed sets based on the gradient, see Section 3.3.
    Use coordinate descent to find the Newton direction $D_t^* = \arg\min_\Delta \bar{f}_{X_t}(X_t + \Delta)$ over the set of free variables, see (13) and (16) in Section 3.1. (A Lasso problem.)
    Use an Armijo-rule based step-size selection to get $\alpha$ such that $X_{t+1} = X_t + \alpha D_t^*$ is positive definite and there is sufficient decrease in the objective function, see (21) in Section 3.2.
end
```

Guarantee of Convergence

Theorem 1

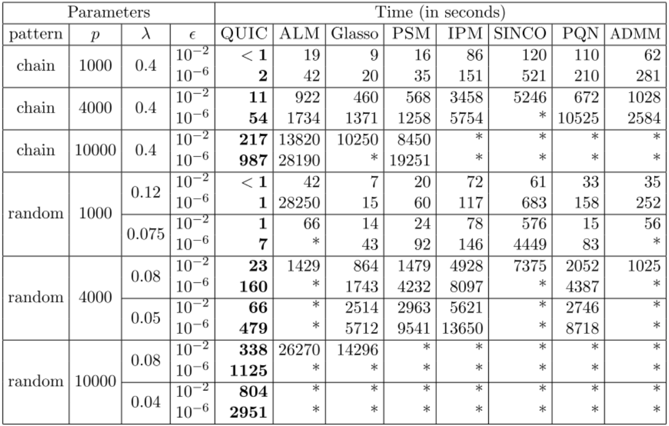
Algorithm converges to the unique global optimum Y^*.

Theorem 2

The sequence $\{X_t\}$ generated by the QUIC algorithm converges quadratically to X^*, that is for some constant $\kappa > 0$,

$$\lim_{t\to\infty} \frac{\|X_{t+1} - X^*\|_F}{\|X_t - X^*\|_F^2} = \kappa$$

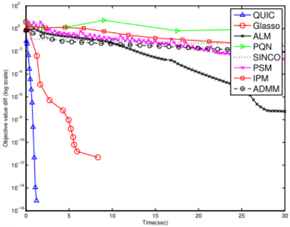
Time Comparison using Synthetic Data Sets



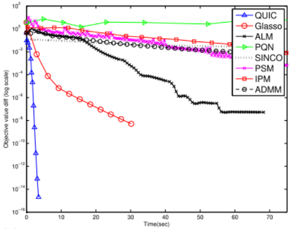
Parameters				Time (in seconds)							
pattern	p	λ	ϵ	QUIC	ALM	Glasso	PSM	IPM	SINCO	PQN	ADMM
chain	1000	0.4	10^{-2}	< **1**	19	9	16	86	120	110	62
			10^{-6}	**2**	42	20	35	151	521	210	281
chain	4000	0.4	10^{-2}	**11**	922	460	568	3458	5246	672	1028
			10^{-6}	**54**	1734	1371	1258	5754	*	10525	2584
chain	10000	0.4	10^{-2}	**217**	13820	10250	8450	*	*	*	*
			10^{-6}	**987**	28190	*	19251	*	*	*	*
random	1000	0.12	10^{-2}	< **1**	42	7	20	72	61	33	35
			10^{-6}	**1**	28250	15	60	117	683	158	252
		0.075	10^{-2}	**1**	66	14	24	78	576	15	56
			10^{-6}	**7**	*	43	92	146	4449	83	*
random	4000	0.08	10^{-2}	**23**	1429	864	1479	4928	7375	2052	1025
			10^{-6}	**160**	*	1743	4232	8097	*	4387	*
		0.05	10^{-2}	**66**	*	2514	2963	5621	*	2746	*
			10^{-6}	**479**	*	5712	9541	13650	*	8718	*
random	10000	0.08	10^{-2}	**338**	26270	14296	*	*	*	*	*
			10^{-6}	**1125**	*	*	*	*	*	*	*
		0.04	10^{-2}	**804**	*	*	*	*	*	*	*
			10^{-6}	**2951**	*	*	*	*	*	*	*

- Graphical Lasso (Glasso) is without block screen.

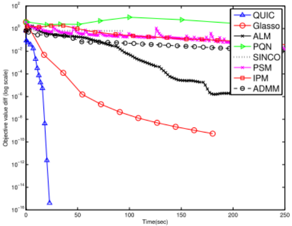
Time Comparison using Real Data Sets



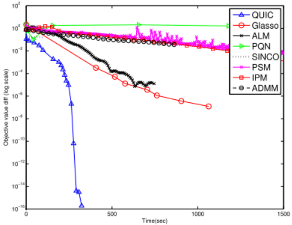
(a) Time taken on ER data set, $p = 692$, $\frac{\|X^*\|_0}{p^2} = 0.0222$



(b) Time taken on Arabidopsis data set, $p = 834$, $\frac{\|X^*\|_0}{p^2} = 0.0296$



(c) Time taken on Leukemia data set, $p = 1,255$, $\frac{\|X^*\|_0}{p^2} = 0.0221$



(d) Time taken on hereditarybc data set, $p = 1,869$, $\frac{\|X^*\|_0}{p^2} = 0.0198$

Penalized Likelihood Methods

One of the commonly used convex penalties is the l_1 penalty [Meinshausen et al., 2006].

$$\widehat{\Theta} = \underset{\boldsymbol{\Theta}}{\mathsf{argmin}} \{\mathsf{tr}(S\boldsymbol{\Theta}) - \mathsf{log}\,|\boldsymbol{\Theta}| + \lambda\|\boldsymbol{\Theta}\|_1\}$$

Solutions

- Interior-point optimization methods [Banerjee et al., 2008]
- Graphical Lasso (most popular) [Friedman et al., 2008]
- Alternating direction method of multipliers (ADMM) [Boyd et al., 2011]
- QUIC [Hsieh et al., 2014]

Trick and variants

- Block screen [Mazumder and Hastie, 2012]
- D-trace loss [Zhang and Zou, 2014]

Block Screen (Important)

Motivation

Suppose $\widehat{\Theta}$ has the following sparse pattern

$$\widehat{\Theta} = \begin{pmatrix} \widehat{\Theta}_1 & 0 & \cdots & 0 \\ 0 & \Theta_2 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \Theta_{k(\lambda)} \end{pmatrix}$$

The log-likelihood problem can be decomposed to subproblems:

$$\widehat{\Theta}_\ell = \operatorname*{arg\,min}_{\Theta_\ell} \left\{ -\log\det\left(\Theta_\ell\right) + \operatorname{tr}\left(S_\ell \Theta_\ell\right) + \lambda \sum_{ij} \left| \left(\Theta_\ell\right)_{ij} \right| \right\}$$

Can we learn such a sparse pattern?

Block Screen (Important)

The sparsity pattern of the solution $\widehat{\Theta}^{(\lambda)}$ is

$$E_{ij}^{(\lambda)} = \begin{cases} 1 & \text{if } \widehat{\Theta}_{ij}^{(\lambda)} \neq 0, i \neq j \\ 0 & \text{otherwise} \end{cases}$$

Block Screen [Mazumder and Hastie, 2012]

The graph edge skeleton E_{ij} is defined by

$$E_{ij}^{(\lambda)} = \begin{cases} 1 & \text{if } |S_{ij}| > \lambda, i \neq j \\ 0 & \text{otherwise} \end{cases}$$

Block Screen (Important)

Block Screen [Mazumder and Hastie, 2012]

The graph edge skeleton E_{ij} is defined by

$$E_{ij}^{(\lambda)} = \begin{cases} 1 & \text{if } |S_{ij}| > \lambda, i \neq j \\ 0 & \text{otherwise} \end{cases}$$

Proof: The KKT conditions of optimality of log-likelihood problem is:

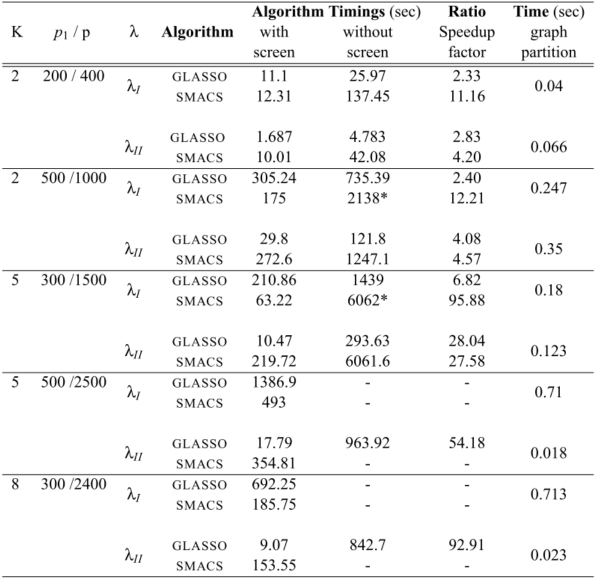
$$\begin{aligned} \left|S_{ij} - \widehat{W}_{ij}\right| \leqslant \lambda, &\qquad \forall\ \widehat{\Theta}_{ij} = 0 \\ \widehat{W}_{ij} = S_{ij} + \lambda, &\qquad \forall\ \widehat{\Theta}_{ij} > 0 \\ \widehat{W}_{ij} = S_{ij} - \lambda, &\qquad \forall\ \widehat{\Theta}_{ij} < 0 \end{aligned}$$

For the $E_{ij}^{(\lambda)} = 0$, set $\widehat{\Theta}_{ij} = 0$, so $\widehat{W}_{ij} = 0$, the KKT condition $\left|S_{ij} - \widehat{W}_{ij}\right| = |S_{ij}| \leqslant \lambda$ satisfied.

Block Screen (Important)

- It is not a specific algorithm for the penalized likelihood.
- It can be used as a wrapper around existing algorithms leads to enormous performance boosts.
- The optimization problem is completely separated into k(λ) separated optimization sub-problems of the form. Help to solve high-dimensional problem.
- Easy to compute in a distributed manner.

Synthetic Examples



K	p_1 / p	λ	Algorithm	Algorithm Timings (sec)		Ratio	Time (sec)
				with screen	without screen	Speedup factor	graph partition
2	200 / 400	λ_I	GLASSO	11.1	25.97	2.33	0.04
			SMACS	12.31	137.45	11.16	
		λ_{II}	GLASSO	1.687	4.783	2.83	0.066
			SMACS	10.01	42.08	4.20	
2	500 /1000	λ_I	GLASSO	305.24	735.39	2.40	0.247
			SMACS	175	2138*	12.21	
		λ_{II}	GLASSO	29.8	121.8	4.08	0.35
			SMACS	272.6	1247.1	4.57	
5	300 /1500	λ_I	GLASSO	210.86	1439	6.82	0.18
			SMACS	63.22	6062*	95.88	
		λ_{II}	GLASSO	10.47	293.63	28.04	0.123
			SMACS	219.72	6061.6	27.58	
5	500 /2500	λ_I	GLASSO	1386.9	-	-	0.71
			SMACS	493	-	-	
		λ_{II}	GLASSO	17.79	963.92	54.18	0.018
			SMACS	354.81	-	-	
8	300 /2400	λ_I	GLASSO	692.25	-	-	0.713
			SMACS	185.75	-	-	
		λ_{II}	GLASSO	9.07	842.7	92.91	0.023
			SMACS	153.55	-	-	

Penalized Likelihood Methods

One of the commonly used convex penalties is the l_1 penalty [Meinshausen et al., 2006].

$$\widehat{\Theta} = \underset{\boldsymbol{\Theta}}{\mathrm{argmin}} \{\mathrm{tr}(S\boldsymbol{\Theta}) - \log|\boldsymbol{\Theta}| + \lambda\|\boldsymbol{\Theta}\|_1\}$$

Solutions

- Interior-point optimization methods [Banerjee et al., 2008]
- Graphical Lasso (most popular) [Friedman et al., 2008]
- Alternating direction method of multipliers (ADMM) [Boyd et al., 2011]
- QUIC [Hsieh et al., 2014]

Trick and variants

- Block screen [Mazumder and Hastie, 2012]
- D-trace loss [Zhang and Zou, 2014]

Variants for Penalized Likelihood Methods

Motivation

- Existing methods for precision matrix estimation do not always guarantee that the final estimator is positive definite.
- The log-determinant term is hard to be analyzed theoretically.
- Work for non-Gaussian data

Variants for Penalized Likelihood Methods

Motivation

- Existing methods for precision matrix estimation do not always guarantee that the final estimator is positive definite.
- The log-determinant term is hard to be analyzed theoretically.
- Work for non-Gaussian data

Zhang et al. (2014) proposed the D-trace loss to estimate precision matrix.

D-trace Loss Estimator

D-trace loss estimator:

$$\Theta = \mathop{\arg\min}_{\Theta \succeq \epsilon I} \frac{1}{2} \left\langle \Theta^2, \Sigma \right\rangle - \mathrm{tr}(\Theta) + \lambda_n \|\Theta\|_1$$

$A \succeq B$ represents $A - B$ is positive semidefinite.

D-trace Loss Estimator

D-trace loss estimator:

$$\Theta = \underset{\Theta \succeq \epsilon I}{\arg\min} \frac{1}{2} \left\langle \Theta^2, \Sigma \right\rangle - \mathrm{tr}(\Theta) + \lambda_n \|\Theta\|_1$$

$A \succeq B$ represents $A - B$ is positive semidefinite.

- Condition 1. It is a smooth convex function of Θ.
- Condition 2. The unique minimizer occurs at $(\Sigma_0)^{-1}$.

ADMM for D-trace Loss Estimator

ADMM form

$$\arg\min_{\Theta_1 \succeq \epsilon I} \frac{1}{2}\left\langle \Theta^2, \Sigma \right\rangle - \mathsf{tr}(\Theta) + \lambda_n \left\|\Theta_0\right\|_1 \quad \text{s.t. } [\Theta, \Theta] = [\Theta_0, \Theta_1]$$

The augmented Lagrangian

$$\begin{aligned} L\left(\Theta, \Theta_0, \Theta_1, \Lambda_0, \Lambda_1\right) = & \frac{1}{2}\left\langle \Theta^2, \Sigma \right\rangle - \mathsf{tr}(\Theta) + \lambda_n \left\|\Theta_0\right\|_{1,\text{ off}} + h\left(\Theta_1 \succeq \epsilon I\right. \\ & + \left\langle \Lambda_0, \Theta - \Theta_0 \right\rangle + \left\langle \Lambda_1, \Theta - \Theta_1 \right\rangle \\ & + (\rho/2)\left\|\Theta - \Theta_0\right\|_F^2 + (\rho/2)\left\|\Theta - \Theta_1\right\|_F^2 \end{aligned}$$

where

$$h\left(\Theta_1 \succeq \epsilon I\right) = \begin{cases} 0, & \Theta_1 \succeq \epsilon I \\ \infty, & \text{otherwise} \end{cases}$$

ADMM for D-trace Loss Estimator

1. Update Θ

$$\Theta^{k+1} = \underset{\Theta=\Theta^\top}{\arg\min} \frac{1}{2}\left\langle \Theta^2, \Sigma + 2\rho I \right\rangle - \left\langle \Theta, I + \rho\Theta_0^k + \rho\Theta_1^k - \Lambda_0^k - \Lambda_1^k \right\rangle$$

2. Update Θ_0

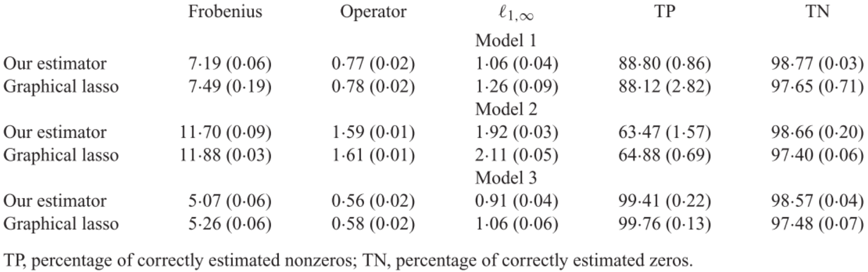
$$\Theta_0^{k+1} = \underset{\Theta_0=\Theta_0^T}{\arg\min} \frac{\rho}{2}\left\langle \Theta_0^2, I \right\rangle - \left\langle \Theta_0, \rho\Theta^{k+1} + \Lambda_0^k \right\rangle + \lambda_n \left\|\Theta_0\right\|_1$$

3. Update Θ_1

$$\Theta_1^{k+1} = \underset{\Theta_1 \succeq \in I}{\arg\min} \frac{\rho}{2}\left\langle \Theta_1^2, I \right\rangle - \left\langle \Theta_1, \rho\Theta^{k+1} + \Lambda_1^k \right\rangle$$

4. $\left[\Lambda_0^{k+1}, \Lambda_1^{k+1}\right] = \left[\Lambda_0^k, \Lambda_1^k\right] + \rho\left[\Theta^{k+1} - \Theta_0^{k+1}, \Theta^{k+1} - \Theta_1^{k+1}\right]$

ADMM for D-trace Loss Estimator



	Frobenius	Operator	$\ell_{1,\infty}$	TP	TN
			Model 1		
Our estimator	7·19 (0·06)	0·77 (0·02)	1·06 (0·04)	88·80 (0·86)	98·77 (0·03)
Graphical lasso	7·49 (0·19)	0·78 (0·02)	1·26 (0·09)	88·12 (2·82)	97·65 (0·71)
			Model 2		
Our estimator	11·70 (0·09)	1·59 (0·01)	1·92 (0·03)	63·47 (1·57)	98·66 (0·20)
Graphical lasso	11·88 (0·03)	1·61 (0·01)	2·11 (0·05)	64·88 (0·69)	97·40 (0·06)
			Model 3		
Our estimator	5·07 (0·06)	0·56 (0·02)	0·91 (0·04)	99·41 (0·22)	98·57 (0·04)
Graphical lasso	5·26 (0·06)	0·58 (0·02)	1·06 (0·06)	99·76 (0·13)	97·48 (0·07)

TP, percentage of correctly estimated nonzeros; TN, percentage of correctly estimated zeros.

Their results do not imply that the D-trace loss estimator is superior to the graphical Lasso.

Nonconvex Penalties

Nonconvex penalties also have been considered under the same normal likelihood model.
They are usually computationally more demanding.

- smoothly clipped absolute deviation (SCAD) penalty aims to ameliorate the bias problem of l_1 penalization
[Fan et al., 2009]

$$p_\lambda(\theta) = \lambda 1_{\{\theta \leqslant \lambda\}} + (a\lambda - \theta)_+ 1_{\{\theta > \lambda\}}/(a-1), \text{ for some } a > 2$$

1 Introduction

2 Gaussian Graphical Model

3 Main Approaches

- Sparsity
- Penalized Likelihood Methods
- Column-by-column Estimation Methods
- CLIME

4 Variants

5 Discussion

Column-by-column Estimation Methods

Column-by-column regression is another approach to estimate the precision matrix.

- Main idea: exploit the relationship between the conditional distribution of multivariate Gaussian and linear regressions.

Column-by-column Estimation Methods

Column-by-column regression is another approach to estimate the precision matrix.

- Main idea: exploit the relationship between the conditional distribution of multivariate Gaussian and linear regressions.

Problem

1. Precision matrix ⇔ conditional dependencies ✓

2. Precision matrix ⇔ conditional dependencies ⇔ linear regressions?

Gaussian Graphical Models

Suppose that observations $x_1, x_2, \ldots, x_n \in \mathbb{R}^P$ are i.i.d. $N_p(\mu, \Sigma)$ where $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{\Sigma}$ is a $p \times p$ positive definite matrix.

- Partition $X = (Z, Y)$ where $Z = (X_1, \ldots, X_{p-1})$ and $Y = X_p$.
- Partitioned Σ and Θ as

$$\boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_{ZZ} & \sigma_{ZY} \\ \sigma_{ZY}^T & \sigma_{YY} \end{pmatrix}, \boldsymbol{\Theta} = \begin{pmatrix} \Theta_{ZZ} & \theta_{ZY} \\ \theta_{ZY}^T & \theta_{YY} \end{pmatrix}$$

Gaussian Graphical Models

Suppose that observations $x_1, x_2, \ldots, x_n \in \mathbb{R}^P$ are i.i.d. $N_p(\mu, \Sigma)$ where $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{\Sigma}$ is a $p \times p$ positive definite matrix.

- Partition $X = (Z, Y)$ where $Z = (X_1, \ldots, X_{p-1})$ and $Y = X_p$.
- Partitioned Σ and Θ as

$$\boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_{ZZ} & \sigma_{ZY} \\ \sigma_{ZY}^T & \sigma_{YY} \end{pmatrix}, \boldsymbol{\Theta} = \begin{pmatrix} \Theta_{ZZ} & \theta_{ZY} \\ \theta_{ZY}^T & \theta_{YY} \end{pmatrix}$$

Because $\Theta = \Sigma^{-1}$ standards formulas for partitioned inverses given:

- $\theta_{YY} = (\sigma_{YY} - \sigma_{ZY}^T \Sigma_{ZZ}^{-1} \sigma_{ZY})^{-1} > 0$
- $\theta_{ZY} = -\theta_{YY} \cdot \Sigma_{ZZ}^{-1} \sigma_{ZY}$

Gaussian Graphical Model

If we perform multiple linear regression of Y on Z

$$\beta_{\mathrm{Y}} = \arg\min_{\beta} \|\mathrm{Y} - \mathrm{Z}\beta_{\mathrm{Y}}\|_2^2$$

Then

$$\beta_{\mathrm{Y}} = \left(\mathrm{Z}^{\mathrm{T}}\mathrm{Z}\right)^{-1}\mathrm{Z}^{\mathrm{T}}\mathrm{Y} = \Sigma_{\mathrm{ZZ}}^{-1}\sigma_{\mathrm{ZY}} = -\theta_{\mathrm{ZY}}/\theta_{\mathrm{YY}}$$

Gaussian Graphical Model

If we perform multiple linear regression of Y on Z

$$\beta_{\mathrm{Y}} = \arg\min_{\beta} \|\mathrm{Y} - \mathrm{Z}\beta_{\mathrm{Y}}\|_2^2$$

Then

$$\beta_{\mathrm{Y}} = \left(\mathrm{Z}^{\mathrm{T}}\mathrm{Z}\right)^{-1} \mathrm{Z}^{\mathrm{T}}\mathrm{Y} = \Sigma_{\mathrm{ZZ}}^{-1}\sigma_{\mathrm{ZY}} = -\theta_{\mathrm{ZY}}/\theta_{\mathrm{YY}}$$

Gaussian Graphical Model

If we perform multiple linear regression of Y on Z

$$\beta_Y = \mathop{\arg\min}_{\beta} \|Y - Z\beta_Y\|_2^2$$

Then

$$\beta_Y = \left(Z^T Z\right)^{-1} Z^T Y = \Sigma_{ZZ}^{-1} \sigma_{ZY} = -\theta_{ZY}/\theta_{YY}$$

We can learn about this dependence structure through multiple linear regression $((\beta_Y)_i = 0 \Leftrightarrow (\theta_{ZY})_i = 0)$.

Column-by-column Estimation Methods

Inspired by the linear regression model in and the fact that regression coefficient is sparse [Meinshausen et al., 2006]

$$\widehat{\boldsymbol{\beta}}_j = \underset{\alpha_j \in \mathbb{R}^{p-1}}{\mathsf{argmin}} \frac{1}{2n} \left\| Y_{*j} - Y_{*/\ j} \boldsymbol{\beta}_j \right\|_2^2 + \lambda_j \left\| \boldsymbol{\beta}_j \right\|_1$$

- Once $\widehat{\boldsymbol{\beta}}_j$ is obtained, we obtain the neighbourhood edges of node j by reading out the non-zero coefficients of $\widehat{\boldsymbol{\beta}}_j$.
- To estimate Θ

$$\widehat{\theta}_{jj}^2 = \frac{1}{n} \| Y_{*j} - Y_{*} \backslash_j \theta_{jj} \|_2^2$$

and plug it into $\theta_{ij} = -\theta_{jj} \beta_{ij}$

Time Costs

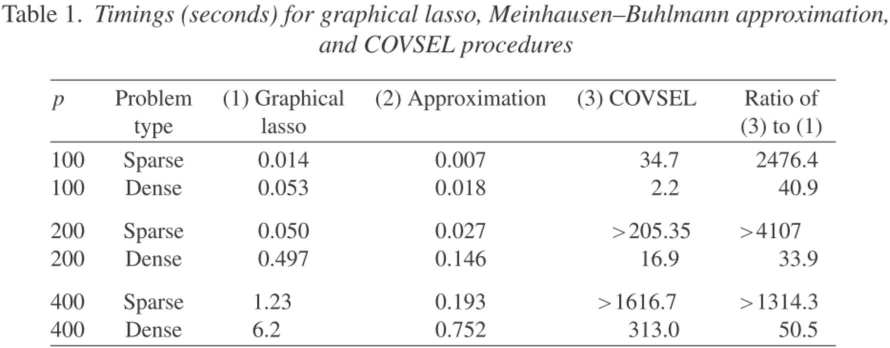


Table 1. *Timings (seconds) for graphical lasso, Meinhausen–Buhlmann approximation, and COVSEL procedures*

p	Problem type	(1) Graphical lasso	(2) Approximation	(3) COVSEL	Ratio of (3) to (1)
100	Sparse	0.014	0.007	34.7	2476.4
100	Dense	0.053	0.018	2.2	40.9
200	Sparse	0.050	0.027	>205.35	>4107
200	Dense	0.497	0.146	16.9	33.9
400	Sparse	1.23	0.193	>1616.7	>1314.3
400	Dense	6.2	0.752	313.0	50.5

- Fast and easy to implement and parallelize
- It is a solution to the quadratic approximation of the log-likelihood function [Banerjee et al., 2008].

TIGER

Problem:

How to choose tuning parameters?

Tuning-Insensitive Graph Estimation and Regression (TIGER) [Liu et al., 2017]

- based on the square-root-Lasso (SQRT-Lasso):

$$\widehat{\boldsymbol{\beta}}_j = \operatorname*{argmin}_{\alpha_j \in \mathbb{R}^{p-1}} \frac{1}{2\sqrt{n}} \left\| Y_{*j} - Y_{*\backslash} \boldsymbol{\beta}_j \right\|_2^2 + \lambda_j \left\| \boldsymbol{\beta}_j \right\|_1$$

  which is asymptotically tuning-free.

- tuning-insensitive

1 Introduction

2 Gaussian Graphical Model

3 Main Approaches
- Sparsity
- Penalized Likelihood Methods
- Column-by-column Estimation Methods
- CLIME

4 Variants

5 Discussion

CLIME

Problem:

Most methods are designed for the Gaussian distribution.

For some classes of non-Gaussian distributions, the problem of estimating the graph can also be reduced to estimating the precision matrix (e.g., the nonparanormal distribution [Liu et al., 2009]).

CLIME

Aim:

Estimating the precision matrix for both sparse and nonsparse matrices, without restricting to a specific sparsity pattern.

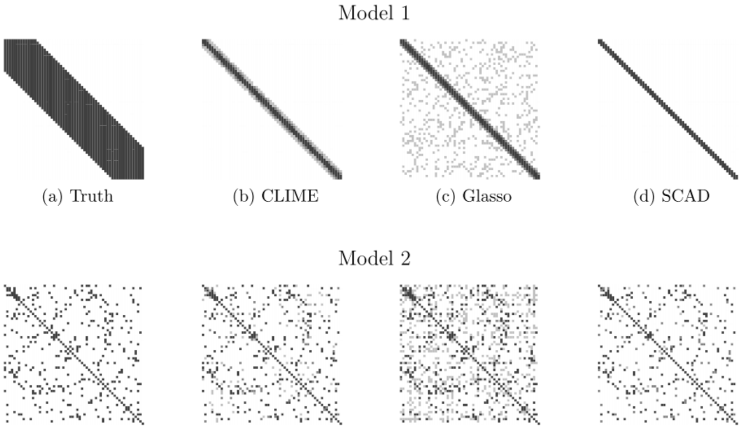
CLIME: a constrained l_1 minimization approach to sparse precision matrix estimation [Cai et al., 2011].

$$\Theta = \underset{\Theta}{\mathsf{argmin}} \|\Theta\|_1 \text{ s.t. } \|\Sigma\Theta - I\|_\infty \leqslant \delta_j, \Theta \in R^{p\times p}$$

CLIME is equivalent to solving the p optimization sub-problems:

$$\Theta_{*j} = \underset{\Theta_{*j}}{\mathsf{argmin}} \|\Theta_{*j}\|_1 \text{ s.t. } \|\Sigma\Theta_{*j} - e_j\|_\infty \leqslant \delta_j, \text{ for } j = 1, \dots, p$$

Simulated Experiments



Model 1

(a) Truth

(b) CLIME

(c) Glasso

(d) SCAD

Model 2

(e) Truth

(f) CLIME

(g) Glasso

(h) SCAD

Real Data Experiments

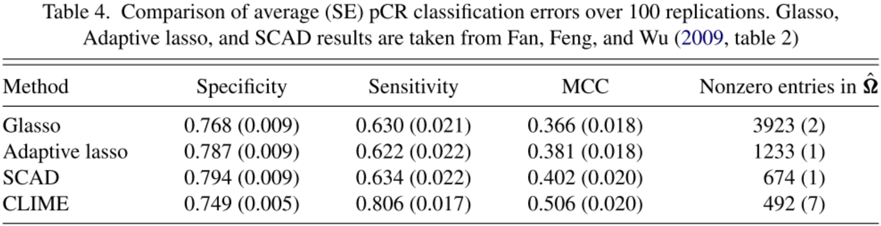


Table 4. Comparison of average (SE) pCR classification errors over 100 replications. Glasso, Adaptive lasso, and SCAD results are taken from Fan, Feng, and Wu (2009, table 2)

Method	Specificity	Sensitivity	MCC	Nonzero entries in $\hat{\boldsymbol{\Omega}}$
Glasso	0.768 (0.009)	0.630 (0.021)	0.366 (0.018)	3923 (2)
Adaptive lasso	0.787 (0.009)	0.622 (0.022)	0.381 (0.018)	1233 (1)
SCAD	0.794 (0.009)	0.634 (0.022)	0.402 (0.020)	674 (1)
CLIME	0.749 (0.005)	0.806 (0.017)	0.506 (0.020)	492 (7)

Outline

1. Introduction

2. Gaussian Graphical Model

3. Main Approaches

4. Variants
   - Graphical Model with Hubs
   - Joint Graphical Lasso
   - Differential Networks Estimation

5. Discussion

1. Introduction

2. Gaussian Graphical Model

3. Main Approaches

4. Variants
   - Graphical Model with Hubs
   - Joint Graphical Lasso
   - Differential Networks Estimation

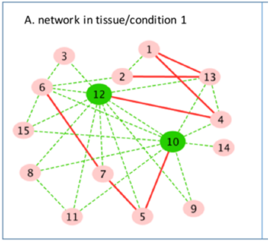
5. Discussion

Graphical Model with Hubs

Motivation

In many applications, there are a few hub nodes that are densely-connected to many other nodes.

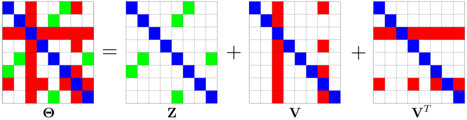
- such as critical genes for organisms to complete their life cycle.



Hub Graphical Lasso was proposed to estimate networks with hub nodes [Tan et al., 2014].

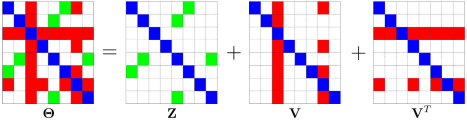
Hub Graphical Lasso

Decomposition of a symmetric matrix Θ into $\mathrm{Z} + \mathrm{V} + \mathrm{V}^{\mathrm{T}}$, where Z is sparse, and most columns of V are entirely zeros.



Hub Graphical Lasso

Decomposition of a symmetric matrix Θ into $Z + V + V^T$, where Z is sparse, and most columns of V are entirely zeros.



Hub Graphical Lasso [Tan et al., 2014]

$$\begin{aligned} \mathsf{minimize}_{\Theta \in \mathcal{S}, V, Z} \Big\{ & \ell(X, \Theta) + \lambda_1 \|Z - \mathsf{diag}(Z)\|_1 + \lambda_2 \|V - \mathsf{diag}(V)\|_1 \\ & + \lambda_3 \sum_{j=1}^{p} \|(V - \mathsf{diag}(V))_j\|_2 \Big\} \end{aligned}$$

$$\text{subject to} \qquad \Theta = V + V^T + Z$$

where $\ell(X, \Theta)$ is the negative log-likelihood.

ADMM Algorithm for Hub Graphical Lasso

ADMM form

Let $B = (\Theta, V, Z), B = (\Theta, V, Z)$

$$\begin{aligned} f(B) =& \ell(X, \Theta) + \lambda_1 \|Z - \mathsf{diag}(Z)\|_1 + \lambda_2 \|V - \mathsf{diag}(V)\|_1 \\ & + \lambda_3 \sum_{j=1}^{p} \|(V - \mathsf{diag}(V))_j\|_2 \end{aligned}$$

$$g(B) = \begin{cases} 0 & \text{if} \Theta = V + V^T + Z \\ \infty & \text{otherwise} \end{cases}$$

The ADMM form:

$$\underset{B,B}{\mathsf{minimize}} \{f(B) + g(B)\} \quad \text{s.t. } B = B$$

ADMM Algorithm for Hub Graphical Lasso

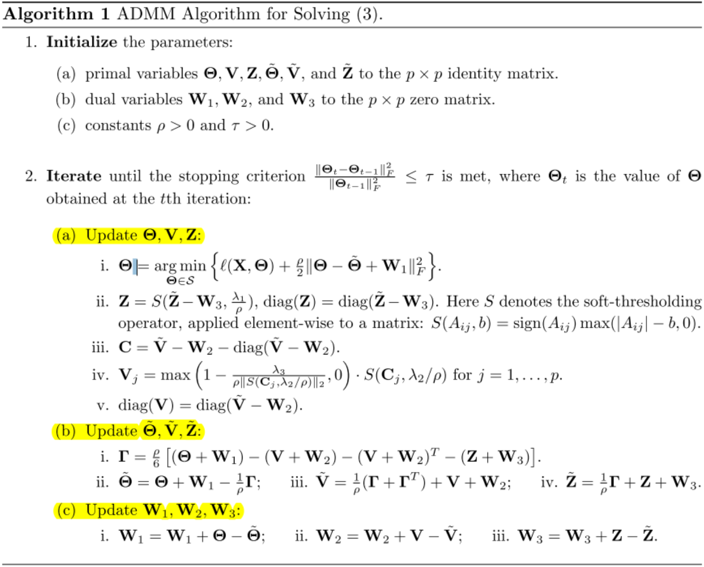
The scaled augmented Lagrangian

$$L(B, B, W) = \ell(X, \Theta) + \lambda_1 \|Z - \mathsf{diag}(Z)\|_1 + \lambda_2 \|V - \mathsf{diag}(V)\|_1 + \lambda_3 \sum_{j=1}^{p} \|(V - \mathsf{diag}(V))_j\|_2 + g(B) + \frac{\rho}{2} \|B - B + W\|_F^2$$

ADMM:

1. Update B (include Θ, V, Z)
2. Update B (include Θ, V, Z)
3. Update W

ADMM Algorithm for Hub Graphical Lasso



Algorithm 1 ADMM Algorithm for Solving (3).

```
1. Initialize the parameters:
    (a) primal variables $\mathbf{\Theta}, \mathbf{V}, \mathbf{Z}, \tilde{\mathbf{\Theta}}, \tilde{\mathbf{V}}$, and $\tilde{\mathbf{Z}}$ to the $p \times p$ identity matrix.
    (b) dual variables $\mathbf{W}_1, \mathbf{W}_2$, and $\mathbf{W}_3$ to the $p \times p$ zero matrix.
    (c) constants $\rho > 0$ and $\tau > 0$.

2. Iterate until the stopping criterion $\frac{\|\mathbf{\Theta}_t - \mathbf{\Theta}_{t-1}\|_F^2}{\|\mathbf{\Theta}_{t-1}\|_F^2} \leq \tau$ is met, where $\mathbf{\Theta}_t$ is the value of $\mathbf{\Theta}$ obtained at the $t$th iteration:
    (a) Update $\mathbf{\Theta}, \mathbf{V}, \mathbf{Z}$:
        i. $\mathbf{\Theta} = \underset{\mathbf{\Theta} \in \mathcal{S}}{\arg\min} \left\{ \ell(\mathbf{X}, \mathbf{\Theta}) + \frac{\rho}{2} \|\mathbf{\Theta} - \tilde{\mathbf{\Theta}} + \mathbf{W}_1\|_F^2 \right\}$.
        ii. $\mathbf{Z} = S(\tilde{\mathbf{Z}} - \mathbf{W}_3, \frac{\lambda_1}{\rho})$, $\mathrm{diag}(\mathbf{Z}) = \mathrm{diag}(\tilde{\mathbf{Z}} - \mathbf{W}_3)$. Here $S$ denotes the soft-thresholding operator, applied element-wise to a matrix: $S(A_{ij}, b) = \mathrm{sign}(A_{ij}) \max(|A_{ij}| - b, 0)$.
        iii. $\mathbf{C} = \tilde{\mathbf{V}} - \mathbf{W}_2 - \mathrm{diag}(\tilde{\mathbf{V}} - \mathbf{W}_2)$.
        iv. $\mathbf{V}_j = \max\left(1 - \frac{\lambda_3}{\rho \|S(\mathbf{C}_j, \lambda_2/\rho)\|_2}, 0\right) \cdot S(\mathbf{C}_j, \lambda_2/\rho)$ for $j = 1, \ldots, p$.
        v. $\mathrm{diag}(\mathbf{V}) = \mathrm{diag}(\tilde{\mathbf{V}} - \mathbf{W}_2)$.
    (b) Update $\tilde{\mathbf{\Theta}}, \tilde{\mathbf{V}}, \tilde{\mathbf{Z}}$:
        i. $\mathbf{\Gamma} = \frac{\rho}{6} \left[ (\mathbf{\Theta} + \mathbf{W}_1) - (\mathbf{V} + \mathbf{W}_2) - (\mathbf{V} + \mathbf{W}_2)^T - (\mathbf{Z} + \mathbf{W}_3) \right]$.
        ii. $\tilde{\mathbf{\Theta}} = \mathbf{\Theta} + \mathbf{W}_1 - \frac{1}{\rho}\mathbf{\Gamma}$;   iii. $\tilde{\mathbf{V}} = \frac{1}{\rho}(\mathbf{\Gamma} + \mathbf{\Gamma}^T) + \mathbf{V} + \mathbf{W}_2$;   iv. $\tilde{\mathbf{Z}} = \frac{1}{\rho}\mathbf{\Gamma} + \mathbf{Z} + \mathbf{W}_3$.
    (c) Update $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3$:
        i. $\mathbf{W}_1 = \mathbf{W}_1 + \mathbf{\Theta} - \tilde{\mathbf{\Theta}}$;   ii. $\mathbf{W}_2 = \mathbf{W}_2 + \mathbf{V} - \tilde{\mathbf{V}}$;   iii. $\mathbf{W}_3 = \mathbf{W}_3 + \mathbf{Z} - \tilde{\mathbf{Z}}$.
```

Simulated Experiments

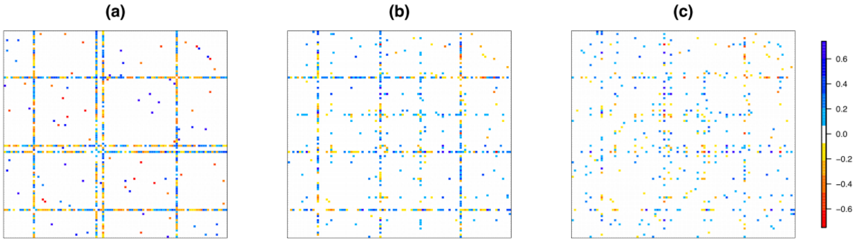


Figure 1: (a): Heatmap of the inverse covariance matrix in a toy example of a Gaussian graphical model with four hub nodes. White elements are zero and colored elements are non-zero in the inverse covariance matrix. Thus, colored elements correspond to edges in the graph. (b): Estimate from the *hub graphical lasso*, proposed in this paper. (c): Graphical lasso estimate.

1 Introduction

2 Gaussian Graphical Model

3 Main Approaches

4 Variants

- Graphical Model with Hubs
- Joint Graphical Lasso
- Differential Networks Estimation

5 Discussion

Joint Graphical Lasso (JGL)

Motivation

The standard formulation for estimating a precision matrix assumes that each observation is drawn from the same distribution.

However, in many datasets the observations may correspond to several distinct classes.

Joint Graphical Lasso (JGL)

Motivation

The standard formulation for estimating a precision matrix assumes that each observation is drawn from the same distribution.

However, in many datasets the observations may correspond to several distinct classes.

Consider the gene expression data of a set of cancer samples and normal samples, respectively.

- Solution 1: estimating graphical model using all sample
  - Ignore the heterogeneity, inappropriate
- Solution 2: estimating separate graphical models for the cancer and normal samples
  - This strategy does not exploit the similarity between the true graphical models.

Joint Graphical Lasso

Suppose one has a heterogeneous data with p variables and K classes.

- The k-th class contains n_k observations $\left(x_1^{(k)}, \ldots, x_{n_k}^{(k)}\right)$, where each $x_i^{(k)} = \left(x_{i,1}^{(k)}, \ldots, x_{i,p}^{(k)}\right)$ is a p-dim row vector.
- $\mathbf{S}^{(k)}$ is the sample covariance matrix of the k-th class.
- $\Theta^{(k)}$ is the inverse covariance matrix of the k-th class.

General formulation for JGL [Danaher et al., 2014]

$$\mathsf{maximize}_{\{\Theta\}} \left(\sum_{k=1}^{K} n_k \left[\mathsf{log} \left\{ \mathsf{det} \left(\Theta^{(k)} \right) \right\} - \mathsf{tr} \left(S^{(k)} \Theta^{(k)} \right) \right] - P(\{\Theta\}) \right)$$

$$P(\{\Theta\}) = \lambda_1 \sum_{k} \sum_{i \neq j} |\theta_{ij}^{(k)}| + \widehat{P}(\{\Theta\})$$

Joint Graphical Lasso

Fused graphical Lasso [Danaher et al., 2014]

The fused graphical lasso (FGL) is the solution to joint graphical model with the penalty:

$$P(\{\Theta\}) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} \left| \theta_{ij}^{(k)} \right| + \lambda_2 \sum_{k<k'} \sum_{i,j} \left| \theta_{ij}^{(k)} - \theta_{ij}^{(k')} \right|$$

- encourages similar edge values.

Joint Graphical Lasso

Group graphical Lasso (GGL) [Danaher et al., 2014]

GGL is the solution to joint graphical model with the penalty:

$$P(\{\Theta\}) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} \left| \theta_{ij}^{(k)} \right| + \lambda_2 \sum_{i \neq j} \left(\sum_{k=1}^{K} \left(\theta_{ij}^{(k)} \right)^2 \right)^{1/2}$$

- encourages a similar pattern of sparsity (i.e. there will be a tendency for the 0s in the K estimated precision matrices to occur in the same places)

ADMM Algorithm for JGL

ADMM form

$$\underset{\{\Theta\},\{Z\}}{\mathsf{lminimize}} \left(-\sum_{k=1}^{K} n_k \left[\mathsf{log}\left\{\mathsf{det}\left(\Theta^{(k)}\right)\right\} - \mathsf{tr}\left(S^{(k)(k)}\right)\right] + P(\{Z\}) \right)$$

$$\text{s.t.} Z^{(k)} = \Theta^{(k)} \text{ for } k = 1, \ldots, K$$

The scaled augmented Lagrangian

$$\begin{aligned} L_p(\{\boldsymbol{\Theta}\}, \{Z\}, \{U\}) = & -\sum_{k=1}^{K} n_k \left[\mathsf{log}\,\mathsf{det}\left(\Theta^{(k)}\right) - \mathsf{tr}\left(S^{(k)}(k)\right)\right] + P(\{Z\}) \\ & + \frac{\rho}{2}\sum_{k=1}^{K} \left\|\Theta^{(k)} - Z^{(k)} + U^{(k)}\right\|_F^2 - \frac{\rho}{2}\sum_{k=1}^{K} \left\|U^{(k)}\right\|_F^2 \end{aligned}$$

ADMM Algorithm for JGL

ADMM Algorithm for JGL
1. Update $\Theta^{(k)}, k = 1, \ldots, K$

$$\widehat{\Theta}^{(k)} = \mathsf{argmin}(-n_k \left[\log\left\{\det\left(\Theta^{(k)}\right)\right\} - \mathsf{tr}\left(S^{(k)}\Theta^{(k)}\right)\right] \\ +(\rho/2)\left\|\Theta^{(k)} - Z_{i-1}^{(k)} + U_{i-1}^{(K)}\right\|_F^2)$$

- Eigenvalue problem

ADMM Algorithm for JGL

2. Update $Z^{(k)}, k = 1, \dots, K$

$$\widehat{Z}^{(k)} = \operatorname{argmin} \sum_{k=1}^{K} \left\| Z^{(k)} - \left(\Theta_i^{(k)} + U_{i-1}^{(k)} \right) \right\|_F^2 + P(Z)$$

- Fused Graphical Lasso
    - The subproblem is a group lasso problem → explicit solution (Friedman et al., 2010)
- Fused Graphical Lasso
    - The subproblem is a fused lasso problem → for each (i, j), costs is $O\{K \log(K)\}$ (Hocking et al., 2011)

3. Update U $\left\{U_{(i)}\right\} \leftarrow \left\{U_{(i-1)}\right\} + \left(\left\{\Theta_{(i)}\right\} - \left\{Z_{(i)}\right\}\right)$

1. Introduction

2. Gaussian Graphical Model

3. Main Approaches

4. Variants
   - Graphical Model with Hubs
   - Joint Graphical Lasso
   - Differential Networks Estimation

5. Discussion

Differential Network Estimation

Motivation

Still consider the gene expression data of a set of cancer tissue samples and a set of normal tissue samples.

- A complete understanding of the molecular basis of cancer will require characterization of the differential network.
- Direct estimation of differential networks will help us to advance the understanding of caner development.

Differential Network Estimation

Motivation

Still consider the gene expression data of a set of cancer tissue samples and a set of normal tissue samples.

- A complete understanding of the molecular basis of cancer will require characterization of the differential network.
- Direct estimation of differential networks will help us to advance the understanding of caner development.

Zhao et al. applied precision matrix estimation to differential network analysis [Zhao et al., 2014].

Differential Network Estimation

Suppose we have two classes of data X, Y

- $\mathbf{S}^{\mathrm{X}}$, $\mathbf{S}^{\mathrm{Y}}$ are the smaple covariance matrix of X and Y.
- Θ^{X}, Θ^{Y} are the inverse covariance matrix of X and Y.

The differential network can be represented by $\Delta = \Theta^{\mathrm{Y}} - \Theta^{\mathrm{X}}$.

Differential Network Estimation

Suppose we have two classes of data X, Y

- $\mathbf{S}^{\mathrm{X}}$, $\mathbf{S}^{\mathrm{Y}}$ are the smaple covariance matrix of X and Y.
- Θ^{X}, Θ^{Y} are the inverse covariance matrix of X and Y.

The differential network can be represented by $\Delta = \Theta^{\mathrm{Y}} - \Theta^{\mathrm{X}}$.

Zhao et al. (2014) applied CLIME to estimate the Δ.

- $\Delta = \Theta^{\mathrm{Y}} - \Theta^{\mathrm{X}}$
- $\mathbf{S}^{\mathrm{X}}\Delta\mathbf{S}^{\mathrm{Y}} - \left(\mathbf{S}^{\mathrm{X}} - \mathbf{S}^{\mathrm{Y}}\right) = 0$

Thus, Δ can be estimated by

$$\min \|\Delta\|_1 \text{ s.t. } \left|\mathbf{S}^{\mathrm{X}}\Delta\mathbf{S}^{\mathrm{Y}} - \left(\mathbf{S}^{\mathrm{X}} - \mathbf{S}^{\mathrm{Y}}\right)\right|_\infty \leqslant \lambda_{\mathrm{n}}$$

Outline

1. Introduction

2. Gaussian Graphical Model

3. Main Approaches

4. Variants

5. Discussion

Discussion

Question 1:

In a general graph, whether a relationship exists between conditional independence and the structure of the precision matrix?

Remain unsolved. There are some progresses:

- High dimensional semiparametric Gaussian copula graphical models [Liu et al., 2012]
- The nonparanormal: Semiparametric estimation of high dimensional undirected graphs [Liu et al., 2009]

Discussion

Question 1:

In a general graph, whether a relationship exists between conditional independence and the structure of the precision matrix?

Remain unsolved. There are some progresses:

- High dimensional semiparametric Gaussian copula graphical models [Liu et al., 2012]
- The nonparanormal: Semiparametric estimation of high dimensional undirected graphs [Liu et al., 2009]

Question 2:

Can we learn directed graphical models from Gaussian data?

Solve it by considering data from stationary Gaussian processes.

- Learning Directed Graphical Models from Gaussian Data [Tank, 2016]

Summary

- Estimation of a precision matrix is a fundamental problem in many areas of statistical analysis.
- Sparsity is the key assumption for precision matrix estimation
- Methods for precision matrix estimation
    - Penalized likelihood methods
    - Column-by-column estimation methods
    - CLIME
- With the increase of data size, precision matrix estimation will become a more important problem.

Software

- Graphical Lasso: R package glasso
  `https://cran.r-project.org/web/packages/glasso/`
- QUIC: MATLAB and R package QUIC
  `https://bigdata.oden.utexas.edu/software/1035/`
- TIGER and CLIME: R package flare
  `https://cran.r-project.org/web/packages/flare/`
- Hub Graphical Lasso: R package hglasso
  `https://cran.r-project.org/web/packages/hglasso/`
- Joint Graphical Lasso: R package JGL
  `https://cran.r-project.org/web/packages/JGL/`

References I

Banerjee, O., Ghaoui, L. E., and d'Aspremont, A. (2008).
Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data.
Journal of Machine learning research, 9(Mar):485–516.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011).
Distributed optimization and statistical learning via the alternating direction method of multipliers.
Foundations and Trends® in Machine learning, 3(1):1–122.

Boyd, S. and Vandenberghe, L. (2004).
Convex optimization.
Cambridge university press.

Cai, T., Liu, W., and Luo, X. (2011).
A constrained l_1 minimization approach to sparse precision matrix estimation.
Journal of the American Statistical Association, 106(494):594–607.

Danaher, P., Wang, P., and Witten, D. M. (2014).
The joint graphical lasso for inverse covariance estimation across multiple classes.
Journal of the Royal Statistical Society: Series B (Statistical Methodology), 76(2):373–397.

Fan, J., Feng, Y., and Wu, Y. (2009).
Network exploration via the adaptive lasso and scad penalties.
The annals of applied statistics, 3(2):521.

Fitch, K. (2019).
Learning directed graphical models from gaussian data.
arXiv preprint arXiv:1906.08050.

References II

Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., et al. (2007).
Pathwise coordinate optimization.
The annals of applied statistics, 1(2):302–332.

Friedman, J., Hastie, T., and Tibshirani, R. (2008).
Sparse inverse covariance estimation with the graphical lasso.
Biostatistics, 9(3):432–441.

Hsieh, C.-J., Sustik, M. A., Dhillon, I. S., and Ravikumar, P. (2014).
Quic: quadratic approximation for sparse inverse covariance estimation.
The Journal of Machine Learning Research, 15(1):2911–2947.

Liu, H., Han, F., Yuan, M., Lafferty, J., Wasserman, L., et al. (2012).
High-dimensional semiparametric gaussian copula graphical models.
The Annals of Statistics, 40(4):2293–2326.

Liu, H., Lafferty, J., and Wasserman, L. (2009).
The nonparanormal: Semiparametric estimation of high dimensional undirected graphs.
Journal of Machine Learning Research, 10(Oct):2295–2328.

Liu, H., Wang, L., et al. (2017).
Tiger: A tuning-insensitive approach for optimally estimating gaussian graphical models.
Electronic Journal of Statistics, 11(1):241–294.

Mazumder, R. and Hastie, T. (2012).
Exact covariance thresholding into connected components for large-scale graphical lasso.
Journal of Machine Learning Research, 13(Mar):781–794.

References III

Meinshausen, N., Bühlmann, P., et al. (2006).
High-dimensional graphs and variable selection with the lasso.
The annals of statistics, 34(3):1436–1462.

Tan, K. M., London, P., Mohan, K., Lee, S.-I., Fazel, M., and Witten, D. (2014).
Learning graphical models with hubs.
The Journal of Machine Learning Research, 15(1):3297–3331.

Zhang, T. and Zou, H. (2014).
Sparse precision matrix estimation via lasso penalized d-trace loss.
Biometrika, 101(1):103–120.

Zhao, S. D., Cai, T. T., and Li, H. (2014).
Direct estimation of differential networks.
Biometrika, 101(2):253–268.