

Joint Gaussian Graphical Model Review Series – I

Probability Foundations

Beilun Wang
Advisor: Yanjun Qi

¹Department of Computer Science, University of Virginia
<http://jointggm.org/>

June 23rd, 2017

Outline

- 1 Notation
- 2 Probability
- 3 Dependence and Correlation
- 4 Conditional Dependence and Partial Correlation

Notation

Notation

\mathbb{P} The probability measure.

Ω The sample space.

\mathcal{F} The event set.

X, Y, Z The random variables.

Probability

Probability Space

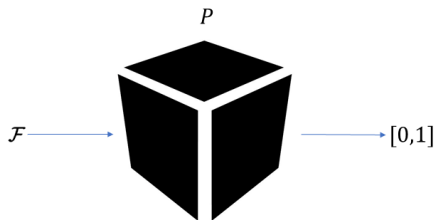
Probability Space

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be the probability space.

- Ω be an arbitrary non-empty set.
- $\mathcal{F} \subset 2^\Omega$ is a set of events.
- \mathbb{P} is the probability measure. In another word, a function : $\mathcal{F} \rightarrow [0, 1]$.

- \mathcal{F} contains Ω .
- \mathcal{F} is closed under complements.
- \mathcal{F} is closed under countable unions.

Probability Measure

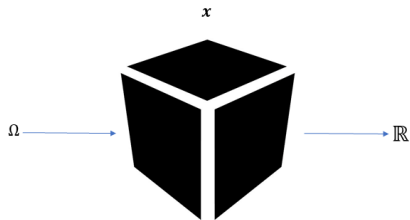


Random Variable

Random Variable

Let $X : \Omega \rightarrow \mathbb{R}$ be a random variable. X is a measurable function.

Random Variable



Probability Distribution

Probability Distribution function

Let $F(x) : \mathbb{R} \rightarrow [0, 1] = \mathbb{P}[X < x]$ where $x \in \mathbb{R}$.

- $X = Y$, they follow same distribution?
- $F_X = F_Y$, then $X = Y$?

Joint Probability

Joint Probability

The probability distribution of random vector (X, Y) .

Joint Probability



Twice

{Head, Head} {Tail, Tail} {Head, Tail}

Marginal Probability

Marginal Probability

A pair of random variable (X, Y) , the probability distribution of X .



Twice

Head or Tail for the first one?

Conditional Distribution

Conditional Distribution

Given the information of Y , the probability distribution of X . Notation $X|Y$.



Twice

I know the second one is Head.
Head or Tail for the first one?

Relationship

Relationship

$$\mathbb{P}(X = x, Y = y) = \mathbb{P}(Y = y)\mathbb{P}(X = x|Y = y)$$

Dependence and Correlation

Independence

Independence

X and Y are independent if and only if $p_{X,Y}(x,y) = p_X(x)p_Y(y)$, where p is the probability density function.

Independence

$$Y|X = Y$$

- Flip coin example
- Causal relationship

Correlation

Covariance

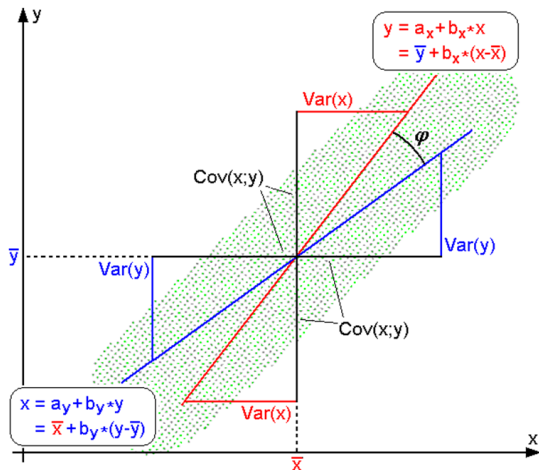
$\text{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$, where μ_X, μ_Y is the mean vector.

Correlation

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

- Linear relationship
- Linear dependency between X and Y .
- $\rho(X, Y) = 1$ means that X and Y are in the same linear direction while $\rho(X, Y) = -1$ means that X and Y are in the reverse linear direction.
- $\rho(X, Y) = 1$ means that when X increase, Y increase with all the points lying on the same line.
- $\rho(X, Y) = 0$ means that X and Y are perpendicular with each other.

Correlation



Dependence and Correlation

- Correlation is easy to estimate the value while independence is a relationship to infer.
- Dependence is stronger relationship than correlation.
- In another word, if X and Y are independent, $\rho(X, Y) = 0$. However, the reverse doesn't hold.
- For example, suppose the random variable X is symmetrically distributed about zero and $Y = X^2$.

Gaussian Example

The distribution of bivariate Gaussian is:

$$f(x, y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)} * \left(\frac{(x-\mu_X)^2}{\sigma_X^2} + \frac{(y-\mu_Y)^2}{\sigma_Y^2} - \right.\right. \\ \left.\left.\right) \quad (3.1)$$

Gaussian Example

Suppose (X, Y) are uncorrelated. i.e., $(X, Y) \sim N(0, \text{diag}(\sigma_X^2, \sigma_Y^2))$.

$$\begin{aligned} f(x, y) &= \frac{1}{2\pi\sigma_X\sigma_Y} \exp\left(-\frac{1}{2}\left(\frac{(x - \mu_X)^2}{\sigma_X^2} + \frac{(y - \mu_Y)^2}{\sigma_Y^2}\right)\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_X} \exp\left(-\frac{1}{2}\frac{(x - \mu_X)^2}{\sigma_X^2}\right) \frac{1}{\sqrt{2\pi}\sigma_Y} \exp\left(-\frac{1}{2}\frac{(y - \mu_Y)^2}{\sigma_Y^2}\right) \quad (3.2) \\ &= f(x)f(y) \end{aligned}$$

Therefore, if (X, Y) follows bivariate Gaussian, (X, Y) are uncorrelated if and only if (X, Y) are independent.

Summary

- Correlation is easy to estimate the value while independence is a relationship to infer.
- In the Gaussian Case, they are equivalent.
- From the structure learning angle, dependence is about the causal relationship, while correlation is, more specifically, the linear relationship.

Conditional Dependence and Partial Correlation

Conditional Dependence

Let's consider a more complicated case. There is another third random variable Z . There are two ways to view the conditional dependence.

- X and Y are independent conditional on Z
- $X|Z$ and $Y|Z$ are independent

Conditional Dependence

X and Y are independent on Z if and only if

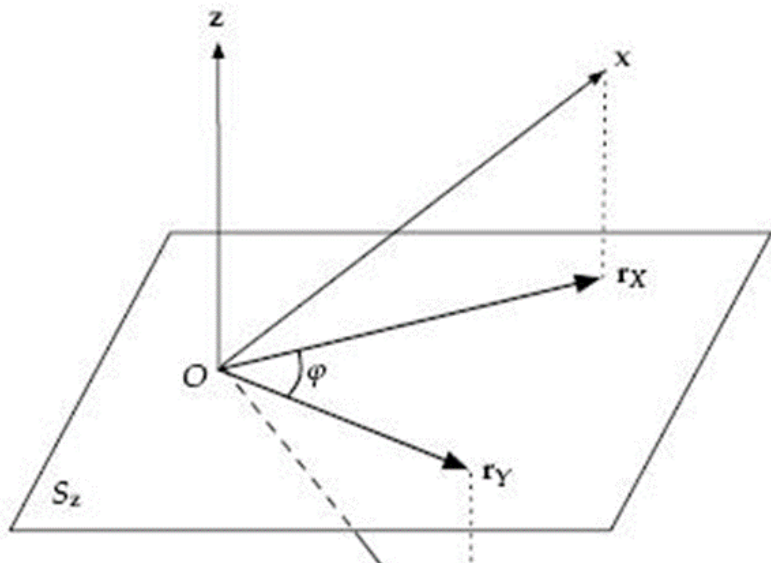
$p_{X,Y|Z}(x,y) = p_{X|Z}(x)p_{Y|Z}(y)$, where p is the probability density function.

Partial Correlation

Partial Correlation

Formally, the partial correlation between X and Y given random variable Z , written $\rho_{XY.Z}$, is the correlation between the residuals R_X and R_Y resulting from the linear regression of X with Z and of Y with Z , respectively.

Partial Correlation



Partial Correlation

Partial Correlation Calculation

Suppose $P = \Sigma^{-1}$ (Σ is covariance matrix or Correlation matrix)

$$\rho_{X_i X_j \cdot \mathbf{V} \setminus \{X_i, X_j\}} = -\frac{P_{ij}}{\sqrt{P_{ii} P_{jj}}}.$$

The value is exactly related to the precision matrix (the inverse of covariance matrix)!

Conditional Dependence and Partial Correlation

- Similarly, in the Gaussian Case, they are equivalent.
- A detailed derivation is in the next talk.

Gaussian Case

- Partial Correlation is easy to estimate the value while conditional independence is a relationship to infer.
- Conditional Dependence is stronger relationship than partial correlation.
- In another word, if $X|Z$ and $Y|Z$ are independent, $\rho(X, Y \cdot Z) = 0$. However, the reverse doesn't hold.

Summary

- Partial correlation is easy to estimate the value while conditional independence is a relationship to infer.
- In the Gaussian Case, they are equivalent.
- From the structure learning angle, conditional dependence is about the causal relationship, while partial correlation is, more specifically, the linear relationship.

Joint Gaussian Graphical Model Review Series – II

Gaussian Graphical Model Basics

Beilun Wang
Advisor: Yanjun Qi

¹Department of Computer Science, University of Virginia
<http://jointggm.org/>

June 30th, 2017

Outline

- 1 Notation
- 2 Reviews
- 3 Why partial correlation and condition dependence are equivalent in the Gaussian case?
- 4 Maximum Likelihood Method
- 5 Regression Method

Notation

Notation

Σ The covariance matrix.

Ω The precision matrix.

μ The mean vector.

x_i The i -th sample follows multivariate normal distribution.

Reviews

Reviews

- Probability basics
- Dependency vs. Correlation
- Conditional dependency vs. partial Correlation

Summary from last talk

- Partial correlation is easy to estimate the value while conditional independence is a relationship to infer.
- In the Gaussian Case, they are equivalent.
- From the structure learning angle, conditional dependence is about the causal relationship, while partial correlation is, more specifically, the linear relationship.

So the remaining question is why in the Gaussian case they are equivalent and how to infer this relationship.

Review: Gaussian Example

Suppose (X, Y) are uncorrelated. i.e., $(X, Y) \sim N(0, \text{diag}(\sigma_X^2, \sigma_Y^2))$.

$$\begin{aligned} f(x, y) &= \frac{1}{2\pi\sigma_X\sigma_Y} \exp\left(-\frac{1}{2}\left(\frac{(x - \mu_X)^2}{\sigma_X^2} + \frac{(y - \mu_Y)^2}{\sigma_Y^2}\right)\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_X} \exp\left(-\frac{1}{2}\frac{(x - \mu_X)^2}{\sigma_X^2}\right) \frac{1}{\sqrt{2\pi}\sigma_Y} \exp\left(-\frac{1}{2}\frac{(y - \mu_Y)^2}{\sigma_Y^2}\right) \quad (2.1) \\ &= f(x)f(y) \end{aligned}$$

Therefore, if (X, Y) follows bivariate Gaussian, (X, Y) are uncorrelated if and only if (X, Y) are independent.

Why partial correlation and condition dependence are equivalent in the Gaussian case?

Multivariate Gaussian Distribution

Density function

Let $X \sim N(\mu, \Sigma)$. $f(x) = (2\pi)^{-\frac{p}{2}} \det(\Sigma)^{-\frac{1}{2}} \exp(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu))$

Partition X , μ , and Σ

Partition X , μ , Σ , Ω .

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

$$\Omega = \Sigma^{-1} = \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix}$$

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

Conditional Distribution of Multivariate Gaussian

If $X \sim N(\mu, \Sigma)$, it holds that $X_2 \sim N(\mu_2, \Sigma_{22})$.

If Σ_{22} is regular, it further holds that

$$X_1 | (X_2 = a) \sim N(\mu_{1|2}, \Sigma_{1|2})$$

where $\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(a - \mu_2)$, and

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} = (\Omega_{11})^{-1}.$$

Partial correlation and condition dependence are equivalent in the Gaussian case

$$X_1 | X_2 = a \sim N(\mu_{1|2}, (\Omega_{11})^{-1}),$$

If X_1 only contains x_i and x_j , then x_i and x_j are conditional independent on others iff $\Omega_{ij} = 0$.

Estimate the condition dependence graph/Partial correlation

Now the only thing left is to estimate $\Omega = \Sigma^{-1}$. There are three potential ways to do that. We call this problem as Gaussian Graphical model.

- Directly calculate the inverse of the sample covariance matrix $\hat{\Sigma}$.
However, we cannot do that when the sample covariance matrix is not invertible.
- Maximum Likelihood Method
- Regression method

For the first one, the sample covariance matrix $\hat{\Sigma}$ may not be invertible.

Maximum Likelihood Method

The MLE of μ

$$\mathcal{L}(\mu, \Omega) = (2\pi)^{-\frac{np}{2}} \prod_{i=1}^n \det(\Omega^{-1})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mu)^T \Omega (\mathbf{x}_i - \mu)\right).$$

After take a first derivative, it is easy to show that $\bar{\mathbf{x}} = \frac{\mathbf{x}_1 + \dots + \mathbf{x}_n}{n}$

The Likelihood of Ω

$$\mathcal{L}(\bar{x}, \Omega) = (2\pi)^{-\frac{np}{2}} \prod_{i=1}^n \det(\Omega^{-1})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_i - \bar{x})^T \Omega (x_i - \bar{x})\right).$$

Notice that $(x_i - \bar{x})^T \Omega (x_i - \bar{x})$ is a scalar. Therefore,

$$(x_i - \bar{x})^T \Omega (x_i - \bar{x}) = \text{trace}((x_i - \bar{x})^T \Omega (x_i - \bar{x})).$$

The Likelihood of Ω

Since $\text{tr}(A, B) = \text{tr}(B, A)$.

$$\mathcal{L}(\bar{x}, \Omega) \propto \det(\Omega^{-1})^{-\frac{n}{2}} \exp \left(-\frac{1}{2} \sum_{i=1}^n \text{tr} \left((x_i - \bar{x})^T \Omega (x_i - \bar{x}) \right) \right) \quad (4.1)$$

$$= \det(\Omega^{-1})^{-\frac{n}{2}} \exp \left(-\frac{1}{2} \sum_{i=1}^n \text{tr} \left((x_i - \bar{x}) (x_i - \bar{x})^T \Omega \right) \right) \quad (4.2)$$

$$= \det(\Omega^{-1})^{-\frac{n}{2}} \exp \left(-\frac{1}{2} \text{tr} \left(\sum_{i=1}^n (x_i - \bar{x}) (x_i - \bar{x})^T \Omega \right) \right) \quad (4.3)$$

$$= \det(\Omega^{-1})^{-\frac{n}{2}} \exp \left(-\frac{1}{2} \text{tr}(S\Omega) \right) \quad (4.4)$$

where, $S = \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \in \mathbb{R}^{p \times p}$.

The Log-Likelihood of Ω

$$\ln \mathcal{L}(\bar{x}, \Omega) = \text{const} - \frac{n}{2} \ln \det(\Omega^{-1}) - \frac{1}{2} \text{tr} \left(\Omega \sum_{i=1}^n (\bar{x} - \mu)(\bar{x} - \mu)^T \right).$$

Since $\det(A^{-1}) = 1/\det(A)$,

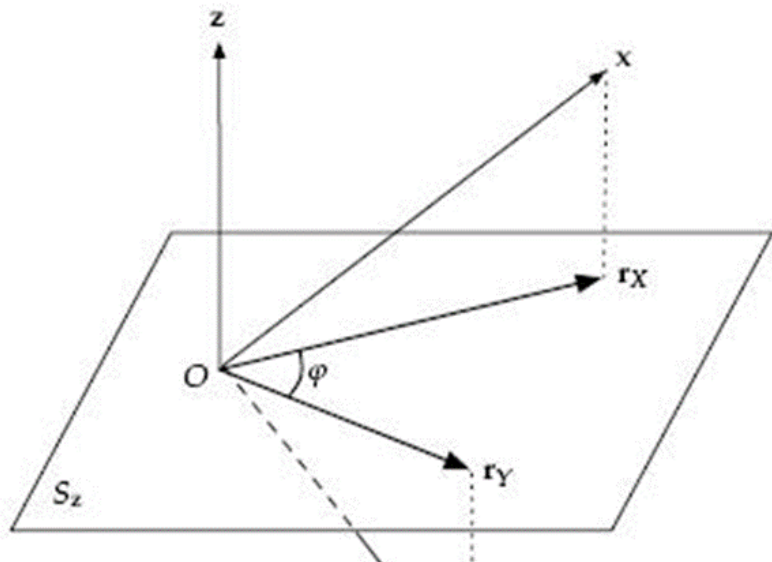
$$\ln \mathcal{L}(\bar{x}, \Omega) \propto \ln \det(\Omega) - \text{tr} \left(\Omega \frac{1}{n} \sum_{i=1}^n (\bar{x} - \mu)(\bar{x} - \mu)^T \right) \quad (4.5)$$

$$= \ln \det(\Omega) - \text{tr} \left(\Omega \hat{S} \right) \quad (4.6)$$

where \hat{S} is the sample covariance matrix.

Regression Method

Partial Correlation



Partial correlation

- As we know, the partial correlation can also be solved by the linear regression.
- In the Gaussian case, we can use so-called neighborhood approach.

Conditional Distribution of Multivariate Gaussian

If $X \sim N(\mu, \Sigma)$, it holds that $X_2 \sim N(\mu_2, \Sigma_{22})$.

If Σ_{22} is regular, it further holds that

$$X_1 | X_2 = a \sim N(\mu_{1|2}, \Sigma_{1|2})$$

where $\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(a - \mu_2)$, and

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} = (\Omega_{11})^{-1}.$$

Neighborhood approach

If $X \sim N(0, \Sigma)$ and let $X_1 = X_j$.

$X_j | X_{\setminus j} \sim N(\Sigma_{\setminus j, j} \Sigma_{\setminus j, \setminus j}^{-1} X_{\setminus j}, \Sigma_{jj} - \Sigma_{\setminus j, j} \Sigma_{\setminus j, \setminus j}^{-1} \Sigma_{\setminus j, j})$

Let $\alpha_j := \Sigma_{\setminus j, j} \Sigma_{\setminus j, \setminus j}^{-1}$ and $\sigma_j^2 := \Sigma_{jj} - \Sigma_{\setminus j, j} \Sigma_{\setminus j, \setminus j}^{-1} \Sigma_{\setminus j, j}$. We have that

$$X_j = \alpha_j^T X_{\setminus j} + \epsilon_j \quad (5.1)$$

where $\epsilon_j \sim N(0, \sigma_j^2)$ is independent of $X_{\setminus j}$.

Neighborhood approach

- We can estimate the α_j by solving p simple linear regression.
- if i -th entry of α_j equals to 0, it means that X_i and X_j are partial uncorrelated and conditional independent.
- Perhaps we want more assumption on α_j like sparsity.

Summary

- In Gaussian case, the partial correlation and the conditional dependence are equivalent
- We have two ways to estimate them. First, directly estimate the precision matrix by MLE. Second, solve p linear regression problem by neighborhood approach.
- None of them have any assumptions on the partial correlation coefficient.
- In the next talk, let's introduce the solutions of these two estimators.

Joint Gaussian Graphical Model Review Series – III

Markov Random Field and Log Linear Model

Beilun Wang
Advisor: Yanjun Qi

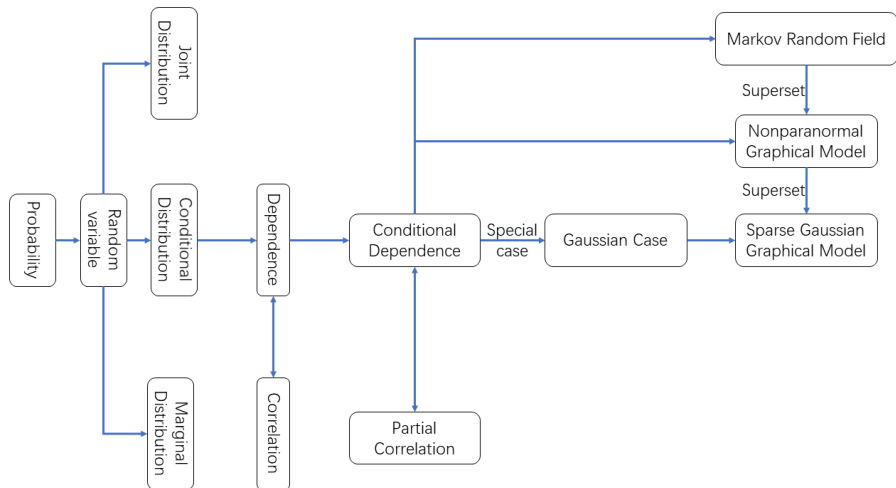
¹Department of Computer Science, University of Virginia
<http://jointggm.org/>

July 7th, 2017

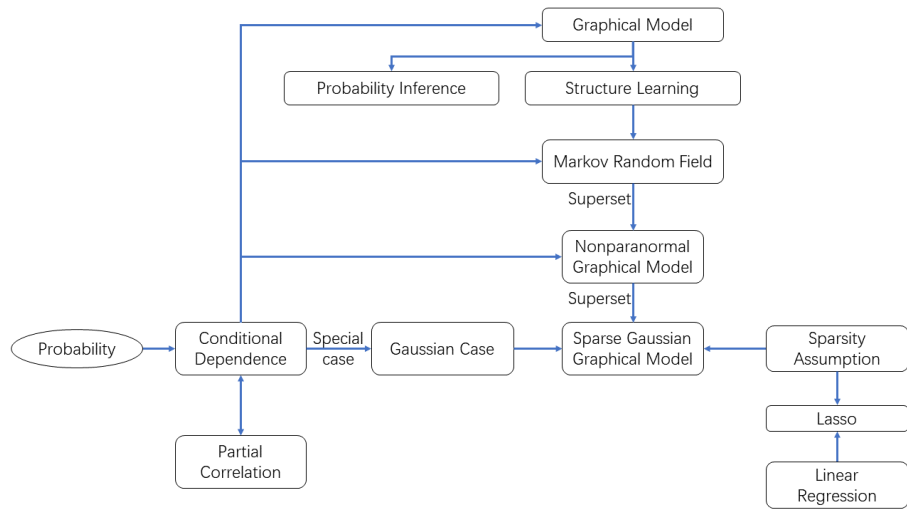
Outline

- 1 Why we need Graphical Model?
- 2 Graphical Model
- 3 Markov Random Field

Road Map



Road Map

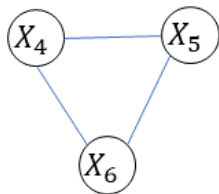
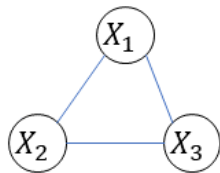


Review: Gaussian Case

- In the Gaussian case, we know the conditional dependence and partial correlation are equivalent.
- This pairwise relationship can be naturally represented by a graph $G = (V, E)$.
- $|\Omega| > 0$ is a natural adjacency matrix.
- We call the pairwise conditional dependence relationship among variables as undirected Graphical Model.

Why we need Graphical Model?

A Toy Example



A Toy Example

Suppose $X = (X_1, X_2, X_3, X_4, X_5, X_6)$. Each variable only takes either 0 or 1. To estimate the joint probability $p(X)$, you need to estimate 2^6 values. However, if we know the conditional independence graph, $p(X) = p(X_1, X_2, X_3)p(X_4, X_5, X_6)$. You only need to estimate 2^4 values.

Proof of the decomposition

First, let's prove that if $X_1 \perp\!\!\!\perp X_3 | X_2$, then $p(X_1 | X_3, X_2) = p(X_1 | X_2)$.
 $p(X_1 | X_2)p(X_3 | X_2) = p(X_1, X_3 | X_2) = p(X_1 | X_3, X_2)p(X_3 | X_2)$. Cancel out $p(X_3 | X_2)$ in the both sides, we can have the conclusion.

It is easy to obtain the similar result under the local markov property:
 $p(X_v | X_{v \setminus N(v)}, X_{N(v)}) = p(X_v | X_{N(v)})$.

Proof of the decomposition

$$p(X_1, X_2, X_3, X_4, X_5, X_6) = p(X_1|X_2, X_3, X_4, X_5, X_6)p(X_2|X_3, X_4, X_5, X_6)p(X_3|X_4, X_5, X_6)p(X_4, X_5, X_6)$$

By the conclusion we have in the last page, the left equals to

$$p(X_1|X_2, X_3)p(X_2|X_3)p(X_3)p(X_4, X_5, X_6) \quad (1.1)$$

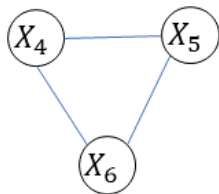
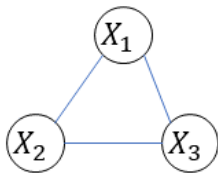
$$= p(X_1, X_2, X_3)p(X_4, X_5, X_6) \quad (1.2)$$

Graphical Model

Graphical Model

- **Probability Inference:** estimate joint probability, marginal probability, and conditional probability.
- **Structure learning:** Give dataset \mathbf{X} , learn the Graph structure from \mathbf{X} (i.e., learn the edge patterns between variables).

A Toy Example



Probability Inference: Calculate the joint Probability

You know that $p(X) = p(X_1, X_2, X_3)p(X_4, X_5, X_6)$. Traditionally,
$$p(X_1, X_2 = a) = \sum_{X_3, X_4, X_5, X_6} p(X_1, X_2 = a, X_3, X_4, X_5, X_6).$$

16 operators.

By the graph, we can have

$$p(X_1, X_2 = a) = \sum_{X_3} p(X_1, X_2 = a, X_3) \sum_{X_4, X_5, X_6} p(X_4, X_5, X_6).$$

10 operators.

Probability Inference: Calculate the joint Probability

You know that $p(X) = p(X_1, X_2, X_3)p(X_4, X_5, X_6)$. Traditionally,
$$p(X_1, X_2 = a) = \sum_{X_3, X_4, X_5, X_6} p(X_1, X_2 = a, X_3, X_4, X_5, X_6).$$

16 operators.

By the graph, we can have

$$p(X_1, X_2 = a) = \sum_{X_3} p(X_1, X_2 = a, X_3) \sum_{X_4, X_5, X_6} p(X_4, X_5, X_6).$$

10 operators.

Markov Random Field

Markov Random Field

Markov Random Field

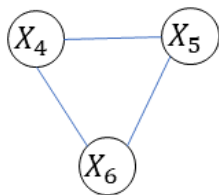
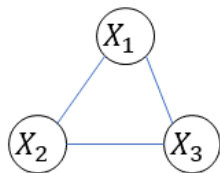
Given an undirected graph $G = (V, E)$, a set of random variables $X = (X_v)_{v \in V}$ indexed by V form a Markov random field with respect to G if they satisfy the local Markov property:

A variable is conditionally independent of all other variables given its neighbors: $X_v \perp\!\!\!\perp X_{V \setminus N(v)} \mid X_{N(v)}$

This property is stronger than the pairwise Markov property:

Any two non-adjacent variables are conditionally independent given all other variables: $X_u \perp\!\!\!\perp X_v \mid X_{V \setminus \{u, v\}}$ if $\{u, v\} \notin E$.

A Toy Example



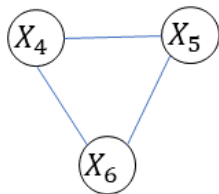
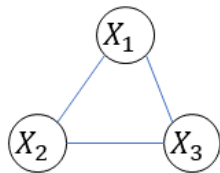
Clique factorization

If this joint density can be factorized over the cliques of G :

$$p(X = x) = \prod_{C \in \text{cl}(G)} \phi_C(x_C)$$

then X forms a Markov random field with respect to G . Here, $\text{cl}(G)$ is the set of cliques of G .

A Toy Example



Log-linear Model

Any Markov random field can be written as log-linear model with feature functions f_k such that the full-joint distribution can be written as:

$$P(X = x) = \frac{1}{Z} \exp \left(\sum_k w_k^\top f_k(X) \right)$$

. Notice that the reverse doesn't hold.

Example I: Pairwise Model

Pairwise Model

$$P(X = x) = \frac{1}{Z(\Theta)} \exp \left(\sum_{s \in V} \theta_s^\top x_s^2 + \sum_{(s,t) \in E} \theta_{st}^\top x_s x_t \right)$$

Examples:

- Gaussian Graphical Model
- Ising Model

These two models have good estimators to infer the MRF. Generally, estimate Θ is difficult. Since it involves computing $Z(\Theta)$ or its derivatives.

Example I: Pairwise Model – Gaussian Case

Gaussian Case

$$f(x_1, \dots, x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right)}{\sqrt{(2\pi)^k |\Sigma|}}$$

Solution:

$$\ln \mathcal{L}(\bar{x}, \Omega) \propto \ln \det(\Omega) - \text{tr} \left(\Omega \frac{1}{n} \sum_{i=1}^n (\bar{x} - \mu)(\bar{x} - \mu)^T \right) \quad (3.1)$$

$$= \ln \det(\Omega) - \text{tr} \left(\Omega \hat{S} \right) \quad (3.2)$$

where \hat{S} is the sample covariance matrix.

For the Ising model, we use generalized covariance matrix to avoid the normalization term.

Example II: Non-pairwise model – Nonparanormal Graphical Model

Are there any non-pairwise model which is easy to estimate?

Nonparanormal Graphical Model

$$P(X = x) = \frac{1}{Z} \exp \left(-\frac{1}{2} (f(x) - \mu)^T \Sigma^{-1} (f(x) - \mu) \right)$$

where $f(X) = (f_1(X_1), f_2(X_2), \dots, f_p(X_p))$ and each f_i is a univariate monotone function. $f(X) \sim N(\mu, \Sigma)$.

Summary

- The formal definition of Markov Random Field (undirected Graphical Model)
- General formulation: Clique factorization
- log-linear Model
- Two examples: pairwise model and nonparanormal Graphical Model.
- In the next talk, let's introduce the solutions of these two estimators for sGGM.

Joint Gaussian Graphical Model Review Series – IV

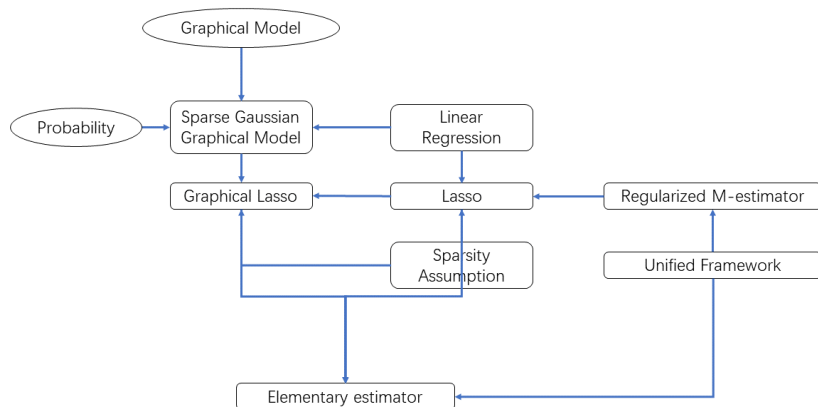
A Unified Framework for M-estimator and Elementary Estimators

Beilun Wang
Advisor: Yanjun Qi

¹Department of Computer Science, University of Virginia
<http://jointggm.org/>

July 21st, 2017

Road Map



Outline

- 1 Notation
- 2 Review
- 3 Regularized M-estimator
- 4 A unified framework
- 5 Elementary Estimator

Notation

Notation

\mathcal{L} The loss function.

\mathcal{R} The Regularization function (norm).

\mathcal{R}^* The Dual norm of \mathcal{R} .

Review

Review from last talk

- Likelihood of the precision matrix in the Gaussian case
- Graphical Model Basics

Regularized M-estimator

Example

We want to buy a TV.

Target:



Constraints: 4K, 65 inch

Result:

SAMSUNG



Regularized M-estimator

M-estimator

In statistics, M-estimators are a broad class of estimators, which are obtained as the minima of sums of functions of the data.

The parameters are estimated by argmin the sums of functions of the data.

target

$\mathcal{L}(X, \theta)$ the loss function

Conditions

$\mathcal{R}(\theta)$ the Regularization function

Therefore, the whole objective function is:

$$\operatorname{argmin}_{\theta} \mathcal{L}(X, \theta) + \lambda_n \mathcal{R}(\theta) \quad (3.1)$$

Example: Linear Model

Let's use the linear regression model as an example.

Target

Find β , such that $X\beta = y$.

Constraints: Sparsity

- **Prediction Accuracy:** Sacrifice a little bias and reduce the variance. Improve the overall performance.
- **Interpretation:** With a large number of predictors, we often would like to determine a smaller subset that exhibits the strongest effect.

$$\operatorname{argmin}_{\beta} \|y - X\beta\|_2 \quad (3.2)$$

$$\text{Subject to: } \|\beta\|_0 \leq t \quad (3.3)$$

Example: Lasso

Since ℓ_0 -norm is not a convex function, we need the closest convex function of ℓ_0 -norm.

$$\operatorname{argmin}_{\beta} \|y - X\beta\|_2 \quad (3.4)$$

$$\text{Subject to: } \|\beta\|_1 \leq t \quad (3.5)$$

Lasso

$$\operatorname{argmin}_{\beta} \|y - X\beta\|_2 + \lambda_n \|\beta\|_1$$

Other equivalent formulation

$$\operatorname{argmin}_{\beta} \|\beta\|_1 \quad (3.6)$$

$$\text{Subject to: } y = X\beta \quad (3.7)$$

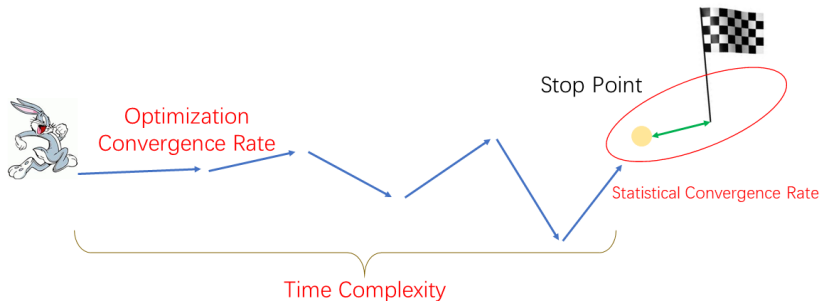
Dantzig selector

$$\operatorname{argmin}_{\beta} \|\beta\|_1 \quad (3.8)$$

$$\text{Subject to: } \|X^T(X\beta - y)\|_{\infty} \leq \lambda_n \quad (3.9)$$

A unified framework

Three major Criteria



Three major Criteria

- Statistical Convergence Rate: How close is between your estimated parameter and the true parameter. It corresponds to estimation error and approximation error.
- Computational Complexity: How fast the algorithm is with respect to certain parameters, e.g., n and p .
- Optimization Rate of Convergence: How fast each optimization step move to the estimated parameter, such as linear or quadratic.

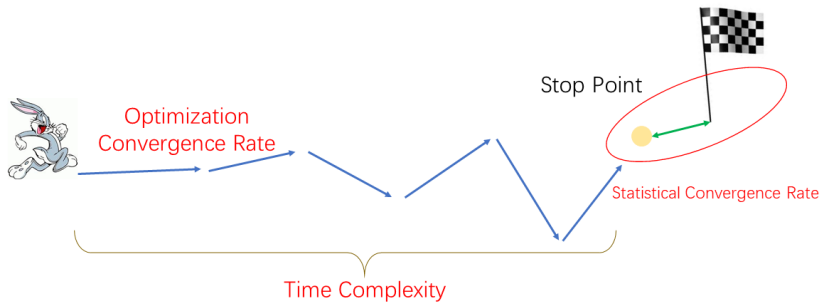
Traditional statisticians focus on the statistical convergence rate (Accuracy).

High dimension vs low dimension

- low dimension: when n is large, the error is asymptotic 0 by the law of large number.
- high dimension (i.e., $p/n \rightarrow c \neq 0$): the error is not asymptotic 0.

High dimensional analysis is relative hard. Traditionally, we need carefully proof for every estimator.

Three major Criteria



A unified framework for M-estimator

[Negahban et al.(2009)Negahban, Yu, Wainwright, and Ravi

Decomposability of \mathcal{R}

Suppose a subspace $\mathcal{M} \subset \mathbb{R}^p$, a norm-based regularizer \mathcal{R} is decomposable with respect to $(\mathcal{M}, \bar{\mathcal{M}}^\perp)$ if

$$\mathcal{R}(\theta + \gamma) = \mathcal{R}(\theta) + \mathcal{R}(\gamma)$$

for all $\theta \in \mathcal{M}$ and $\gamma \in \bar{\mathcal{M}}^\perp$, where

$$\bar{\mathcal{M}}^\perp := \{v \in \mathbb{R}^p \mid \langle u, v \rangle = 0 \forall u \in \mathcal{M}\}.$$

Subspace compatibility constant

$$\Phi(\mathcal{M}) := \sup_{u \in \mathcal{M} \setminus \{0\}} \frac{\mathcal{R}(u)}{\|u\|}$$

with respect to the pair $(\mathcal{R}, \|\cdot\|)$.

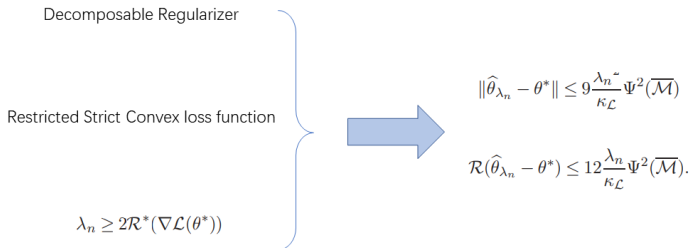
A unified framework for M-estimator

[Negahban et al.(2009)Negahban, Yu, Wainwright, and Ravi

Example: ℓ_1

ℓ_1 is decomposable and the $\Phi(\mathcal{M}) = \sqrt{s}$ with respect to (ℓ_1, ℓ_2) .

A unified framework for M-estimator



Example: Lasso

$$\|\hat{\theta}_{\lambda_n} - \theta^*\|_2^2 \leq O\left(\frac{s \log p}{n}\right)$$

In high dimensional setting, the sparsity assumption actually improves the convergence rate a lot.

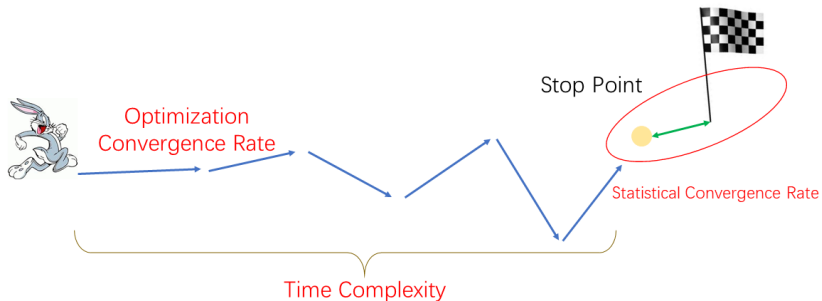
Elementary Estimator

We have a very powerful tool to easily prove the convergence rate. We can also follow the similar process to prove the convergence rate for estimators like Dantzig Selector.

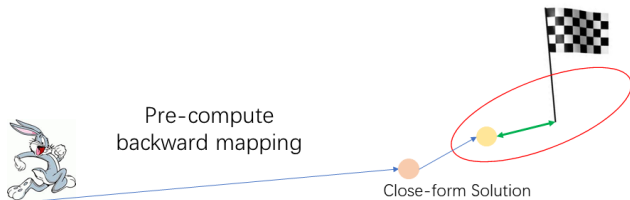
However, a lot of statistical method is slow when p and n are large and they are not scalable at all.

Are there any estimators with close form solution for the statistic problem, which also achieve the optimal convergence rate?

Three major Criteria



Three major Criteria



Elementary

Estimator [Yang et al. (2014b) Yang, Lozano, and Ravikumar]

$$\operatorname{argmin}_{\theta} \mathcal{R}(\theta) \quad (5.1)$$

$$\text{Subject to: } \mathcal{R}^*(\theta - \mathcal{B}^*(\hat{\phi})) \leq \lambda_n \quad (5.2)$$

Here $\mathcal{B}^*(\hat{\phi})$ is a backward mapping for $\hat{\phi}$.

Example: sparse linear regression [Yang et al. (2014a), Yang, Lozano, and Ravikumar]




$$\operatorname{argmin}_{\theta} \|\theta\|_1 \quad (5.3)$$

$$\text{Subject to: } \|\theta - (X^T X + \epsilon I)^{-1} X^T y\|_{\infty} \leq \lambda_n \quad (5.4)$$

Summary

- We review the unified framework for M-estimator, which can be applied to most regularized M-estimator problem
- Following the similar proof strategy, we have the set of elementary estimators.

References I

-  S. Negahban, B. Yu, M. J. Wainwright, and P. K. Ravikumar.
A unified framework for high-dimensional analysis of m -estimators with decomposable regularizers.
In Advances in Neural Information Processing Systems, pages 1348–1356, 2009.
-  E. Yang, A. Lozano, and P. Ravikumar.
Elementary estimators for high-dimensional linear regression.
In Proceedings of the 31st International Conference on Machine Learning (ICML-14), pages 388–396, 2014a.
-  E. Yang, A. C. Lozano, and P. K. Ravikumar.
Elementary estimators for graphical models.
In Advances in Neural Information Processing Systems, pages 2159–2167, 2014b.

Joint Gaussian Graphical Model Series – V

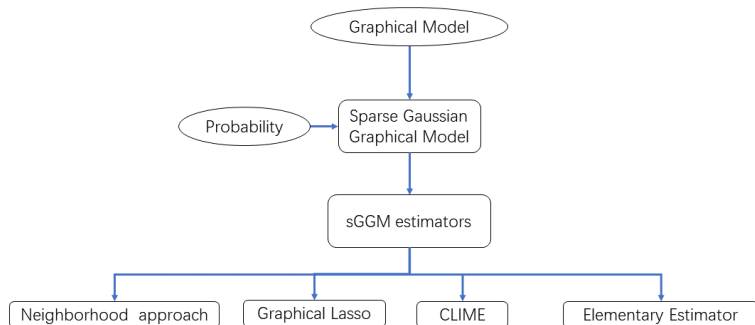
sparse Gaussian Graphical Model estimators

Beilun Wang
Advisor: Yanjun Qi

¹Department of Computer Science, University of Virginia
<http://jointggm.org/>

July 28th, 2017

Road Map



Outline

- 1 Notation
- 2 Review
- 3 Neighborhood Method
- 4 Graphical Lasso
- 5 CLIME
- 6 Elementary Estimator for Gaussian Graphical Model

Notation

Notation

- Σ The covariance matrix.
- Ω The precision matrix.
- p The number of features.
- n The number of samples.

Review

Review from last talk

- Regularized M-estimator $\operatorname{argmin}_{\theta} \mathcal{L}(\theta) + \lambda_n \mathcal{R}(\theta)$
- a unified framework to analyze the statistical convergence rate for high-dimensional statistics
- Elementary Estimator

Review of Gaussian Graphical Model

Suppose the precision matrix $\Omega = \Sigma^{-1}$.

The log-likelihood of Ω equals to $\ln \det(\Omega) - \text{tr}(\Omega \hat{S})$

In this talk, we will use this likelihood to derive several estimators of sparse Gaussian Graphical Model (sGGM)

Neighborhood Method

Neighborhood approach

If $X \sim N(0, \Sigma)$ and let $X_1 = X_j$.

$X_j | X_{\setminus j} \sim N(\Sigma_{\setminus j, j} \Sigma_{\setminus j, \setminus j}^{-1} X_{\setminus j}, \Sigma_{jj} - \Sigma_{\setminus j, j} \Sigma_{\setminus j, \setminus j}^{-1} \Sigma_{\setminus j, j})$

Let $\alpha_j := \Sigma_{\setminus j, j} \Sigma_{\setminus j, \setminus j}^{-1}$ and $\sigma_j^2 := \Sigma_{jj} - \Sigma_{\setminus j, j} \Sigma_{\setminus j, \setminus j}^{-1} \Sigma_{\setminus j, j}$. We have that

$$X_j = \alpha_j^T X_{\setminus j} + \epsilon_j \quad (3.1)$$

where $\epsilon_j \sim N(0, \sigma_j^2)$ is independent of $X_{\setminus j}$.

Neighborhood approach with sparse assumption

By the sparse assumption, we estimate each α_j by a lasso estimator

$$\alpha_j = \underset{\alpha_j}{\operatorname{argmin}} \|\alpha_j^T \mathbf{X}_{\setminus j} - \mathbf{X}_j\|_2^2 + \lambda \|\alpha_j\|_1 \quad (3.2)$$

Review of Lasso solution

Lasso

$$\beta = \underset{\beta}{\operatorname{argmin}} \|\beta^T X - y\|_2^2 + \lambda \|\beta\|_1 \quad (3.3)$$

subgradient method

$$g(\beta; \lambda) = -2X^T(y - X\beta) + \lambda \operatorname{sgn}(\beta) \quad (3.4)$$

Review of Lasso solution: State of the Art

We see that the proximity operator is important because x^* is a minimizer to the problem $\min_{x \in \mathcal{H}} F(x) + R(x)$ if and only if $x^* = \text{prox}_{\gamma R}(x^* - \gamma \nabla F(x^*))$, where $\gamma > 0$. γ is any positive real number.

Proximal gradient method

$$\left(\text{prox}_{\gamma R}(x)\right)_i = \begin{cases} x_i - \gamma, & x_i > \gamma \\ 0, & |x_i| \leq \gamma \\ x_i + \gamma, & x_i < -\gamma, \end{cases} \quad (3.5)$$

By using the fixed point method, you can obtain the estimation of β .

Graphical Lasso

Graphical

Lasso [Friedman et al. (2008) Friedman, Hastie, and Tibshirani]

We already have the log-likelihood as the loss function. Can we use it to obtain a similar estimator as Lasso?

$$\operatorname{argmin}_{\Omega} -\ln \det(\Omega) + \operatorname{tr}(\Omega \hat{S}) + \lambda_n \|\Omega\|_1 \quad (4.1)$$

Proximal gradient method to solve it

Let's do a practice in the white board.

Super Linear algorithm.

$$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|} = 0.$$

State of the art method: Big & QUIC[Hsieh et al.(2011)Hsieh, Sustik, Dhillon, and Ravikum

Parallelized Coordinate descent.

approximated quadratic algorithm.

$$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^2} < M$$

CLIME

CLIME

$$\operatorname{argmin}_{\Omega} \|\Omega\|_1, \text{ subject to: } \|\Sigma\Omega - I\|_{\infty} \leq \lambda \quad (5.1)$$

Here $\lambda > 0$ is the tuning parameter.

By taking the first derivative of Eq. (4.1) and setting it equal to zero, the solution $\hat{\Omega}_{glasso}$ also satisfies:

$$\hat{\Omega}_{glasso}^{-1} - \hat{\Sigma} = \lambda \hat{Z} \quad (5.2)$$

where \hat{Z} is an element of the subdifferential $\partial \|\hat{\Omega}_{glasso}\|_1$.

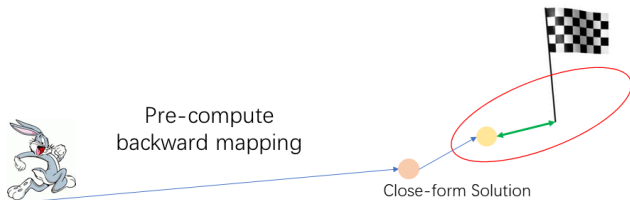
Column-wise estimator

$$\operatorname{argmin} \|\beta\|_1 \quad \text{subject to} \quad \|\Sigma\beta - \mathbf{e}_j\|_\infty \leq \lambda$$

CLIME can be estimated column-by-column.

Elementary Estimator for Gaussian Graphical Model

Elementary Estimator



Elementary

Estimator [Yang et al. (2014b) Yang, Lozano, and Ravikumar]

$$\operatorname{argmin}_{\theta} \mathcal{R}(\theta) \quad (6.1)$$

$$\text{Subject to: } \mathcal{R}^*(\theta - \mathcal{B}^*(\hat{\phi})) \leq \lambda_n \quad (6.2)$$

Here $\mathcal{B}^*(\hat{\phi})$ is a backward mapping for $\hat{\phi}$.

Example: sparse linear regression [Yang et al. (2014a) Yang, Lozano, and Ravikumar]

$$\operatorname{argmin}_{\theta} \|\theta\|_1 \quad (6.3)$$

$$\text{Subject to: } \|\theta - (X^T X + \epsilon I)^{-1} X^T y\|_{\infty} \leq \lambda_n \quad (6.4)$$

Elementary Estimator for sGGM

$$\begin{aligned} & \underset{\Omega}{\operatorname{argmin}} |\Omega|_{1,off} \\ \text{subject to: } & |\Omega - [T_v(\Sigma)]^{-1}|_{\infty,off} \leq \lambda_n \end{aligned} \quad (6.5)$$

Summary

- We review most sGGM estimators.

References I



T. Cai, W. Liu, and X. Luo.

A constrained l_1 minimization approach to sparse precision matrix estimation.

Journal of the American Statistical Association, 106(494):594–607, 2011.



J. Friedman, T. Hastie, and R. Tibshirani.

Sparse inverse covariance estimation with the graphical lasso.

Biostatistics, 9(3):432–441, 2008.



C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. D. Ravikumar.

Sparse inverse covariance matrix estimation using quadratic approximation.

In *NIPS*, pages 2330–2338, 2011.

References II



E. Yang, A. Lozano, and P. Ravikumar.

Elementary estimators for high-dimensional linear regression.

In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 388–396, 2014a.



E. Yang, A. C. Lozano, and P. K. Ravikumar.

Elementary estimators for graphical models.

In *Advances in Neural Information Processing Systems*, pages 2159–2167, 2014b.

Joint Gaussian Graphical Model Series – VI

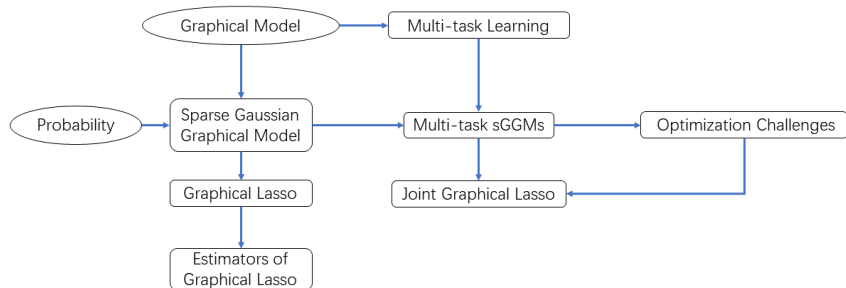
Multi-task sGGMs and its optimization challenges

Beilun Wang
Advisor: Yanjun Qi

¹Department of Computer Science, University of Virginia
<http://jointggm.org/>

August 4th, 2017

Road Map



Outline

- 1 Notation
- 2 Review
- 3 Multi-task Learning
- 4 Multi-task sGGMs
- 5 Optimization Challenge of Multi-task sGGMs
- 6 Joint Graphical Lasso Example

Notation

Notation

$X^{(i)}$ The i -th data matrix

$\Sigma^{(i)}$ The i -th covariance matrix.

$\Omega^{(i)}$ The i -th precision matrix.

p The number of features.

n_i The number of samples in the i -th data matrix.

K The number of tasks.

Review

Review from last talk

- We introduce four estimators of sparse Gaussian Graphical Model.
- We finish most contents about sparse Gaussian Graphical Model in the last five talks.

Review of Gaussian Graphical Model

Suppose the precision matrix $\Omega = \Sigma^{-1}$.

The log-likelihood of Ω equals to $\ln \det(\Omega) - \text{tr}(\Omega \hat{S})$

Multi-task Learning

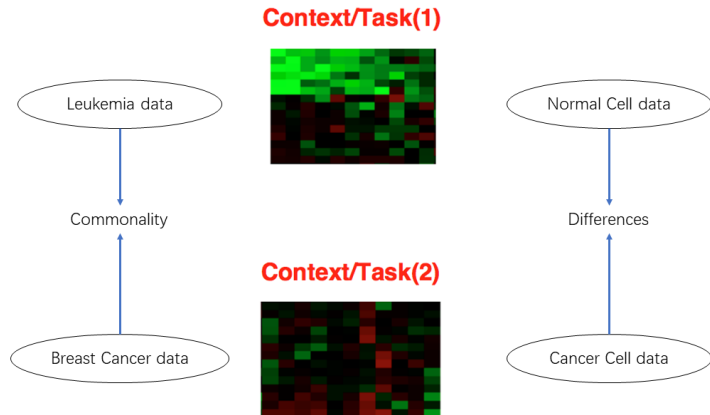
Multi-task Learning

Multi-task Learning

Multi-task learning (MTL) is a subfield of machine learning in which multiple learning tasks are solved at the same time, while exploiting commonalities and differences across tasks.

This can result in improved learning efficiency and prediction accuracy for the task-specific models, when compared to training the models separately.

Multi-task Learning



Multi-task Learning–Linear Classifier Example

Linear Classifier

$$f(x) = \text{sgn}(w^T x + b) \quad (3.1)$$

Multi-task Linear Classifiers

For the i -th task,

$$f_i(x) = \text{sgn}((w_S^T + w_i^T)x + b) \quad (3.2)$$

Multi-task sGGMs

Multi-task sGGMs

Problem

- Input: $\{X^{(i)}\}$
- Output: $\{\Omega^{(i)}\}$
- Assumption I: Sparsity
- Assumption II: Commonalities and Differences

Multi-task sGGMs

Likelihood

$$\sum_i n_i (\ln \det(\Omega^{(i)}) - \text{tr}(\Omega^{(i)} \widehat{\mathcal{S}}^{(i)})) \quad (4.1)$$

Likelihood with sparsity assumption

$$\sum_i n_i (\ln \det(\Omega^{(i)}) - \text{tr}(\Omega^{(i)} \widehat{\mathcal{S}}^{(i)})) \quad (4.2)$$

$$\text{Subject to: } \|\Omega^{(i)}\|_1 \leq t \quad (4.3)$$

Multi-task sGGMs

Likelihood with multi-task setting

$$\sum_i n_i (\ln \det(\Omega^{(i)}) - \text{tr}(\Omega^{(i)} \widehat{S}^{(i)})) \quad (4.4)$$

$$\text{Subject to: } \|\Omega^{(i)}\|_1 \leq t \quad (4.5)$$

$$P(\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(K)}) \leq t_2 \quad (4.6)$$

Joint Graphical Lasso

[Danaher et al.(2013) Danaher, Wang, and Witten]

$$-\sum_i n_i (\ln \det(\Omega^{(i)}) + \text{tr}(\Omega^{(i)} \widehat{S}^{(i)})) + \lambda_1 \|\Omega^{(i)}\|_1 + \lambda_2 P(\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(K)}) \quad (4.7)$$

Optimization Challenge of Multi-task sGGMs

General formulation

Likelihood with multi-task setting

$$-\sum_i n_i (\ln \det(\Omega^{(i)}) + \text{tr}(\Omega^{(i)} \widehat{S}^{(i)})) \quad (5.1)$$

$$\text{Subject to: } \|\Omega^{(i)}\|_1 \leq t \quad (5.2)$$

$$P(\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(K)}) \leq t_2 \quad (5.3)$$

General formulation

$$\sum_{x,z} f(x) + g(z) \quad (5.4)$$

$$\text{Subject to: } Ax + Bz = c \quad (5.5)$$

Optimization Challenge



Alternating direction method of multipliers

- ▶ ADMM problem form (with f, g convex)

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

– two sets of variables, with separable objective

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

- ▶ ADMM:

$$x^{k+1} := \operatorname{argmin}_x L_\rho(x, z^k, y^k) \quad // \textit{x-minimization}$$

$$z^{k+1} := \operatorname{argmin}_z L_\rho(x^{k+1}, z, y^k) \quad // \textit{z-minimization}$$

$$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \quad // \textit{dual update}$$

Optimization Challenges

- For $K > 2$ tasks, you need carefully derive the whole optimization solution.
- Each step in each iteration is still a sub-optimization problem. Sometimes, it is already difficult to solve.
- This method is at most linear Convergence.

Joint Graphical Lasso Example

JGL-group Lasso example

$$L_\rho(\{\Theta\}, \{\mathbf{Z}\}, \{\mathbf{U}\}) = - \sum_{k=1}^K n_k \left(\log \det \Theta^{(k)} - \text{trace}(\mathbf{S}^{(k)} \Theta^{(k)}) \right) + P(\{\mathbf{Z}\}) \\ + \frac{\rho}{2} \sum_{k=1}^K \|\Theta^{(k)} - \mathbf{Z}^{(k)} + \mathbf{U}^{(k)}\|_F^2,$$

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

- (a) $\{\Theta_{(i)}\} \leftarrow \arg \min_{\{\Theta\}} \{L_\rho(\{\Theta\}, \{\mathbf{Z}_{(i-1)}\}, \{\mathbf{U}_{(i-1)}\})\}$.
- (b) $\{\mathbf{Z}_{(i)}\} \leftarrow \arg \min_{\{\mathbf{Z}\}} \{L_\rho(\{\Theta_{(i)}\}, \{\mathbf{Z}\}, \{\mathbf{U}_{(i-1)}\})\}$.
- (c) $\{\mathbf{U}_{(i)}\} \leftarrow \{\mathbf{U}_{(i-1)}\} + (\{\Theta_{(i)}\} - \{\mathbf{Z}_{(i)}\})$.

JGL solution – updating $\Theta^{(i)}$

For $k = 1, \dots, K$, update $\Theta_{(i)}^{(k)}$ as the minimizer (with respect to $\Theta^{(k)}$) of

$$-n_k \left(\log \det \Theta^{(k)} - \text{trace}(\mathbf{S}^{(k)} \Theta^{(k)}) \right) + \frac{\rho}{2} \|\Theta^{(k)} - \mathbf{Z}_{(i-1)}^{(k)} + \mathbf{U}_{(i-1)}^{(k)}\|_F^2.$$

Letting \mathbf{VDV}^T denote the eigendecomposition of $\mathbf{S}^{(k)} - \rho \mathbf{Z}_{(i-1)}^{(k)} / n_k + \rho \mathbf{U}_{(i-1)}^{(k)} / n_k$, the solution is given (Witten & Tibshirani 2009) by $\mathbf{V}\tilde{\mathbf{D}}\mathbf{V}^T$, where $\tilde{\mathbf{D}}$ is the diagonal matrix with j th diagonal element

$$\frac{n_k}{2\rho} \left(-D_{jj} + \sqrt{D_{jj}^2 + 4\rho/n_k} \right).$$

Set the gradient to be 0, we can get the SVD part of the solution.

JGL solution – updating $Z^{(i)}$

$$\text{minimize}_{\{\mathbf{Z}\}} \left\{ \frac{\rho}{2} \sum_{k=1}^K \|\mathbf{Z}^{(k)} - \mathbf{A}^{(k)}\|_F^2 + P(\{\mathbf{Z}\}) \right\},$$

where

$$\mathbf{A}^{(k)} = \Theta_{(i)}^{(k)} + \mathbf{U}_{(i-1)}^{(k)}.$$

$$\text{minimize}_{\{\mathbf{Z}\}} \left\{ \frac{\rho}{2} \sum_{k=1}^K \|\mathbf{Z}^{(k)} - \mathbf{A}^{(k)}\|_F^2 + \lambda_1 \sum_{k=1}^K \sum_{i \neq j} |Z_{ij}^{(k)}| + \lambda_2 \sum_{i \neq j} \sqrt{\sum_k Z_{ij}^{(k)2}} \right\}.$$

$$\hat{Z}_{ij}^{(k)} = S(A_{ij}^{(k)}, \lambda_1/\rho) \left(1 - \frac{\lambda_2}{\rho \sqrt{\sum_{k=1}^K S(A_{ij}^{(k)}, \lambda_1/\rho)^2}} \right)_+,$$

An example for difficulty of ADMM

Algorithm 1: ADMM algorithm for the PNJGL optimization problem (6)

input: $\rho > 0, \mu > 1, t_{\max} > 0$;

Initialize: Primal variables to the identity matrix and dual variables to the zero matrix;

for $t = 1:t_{\max}$ **do**

$\rho \leftarrow \mu\rho$;

while *Not converged* **do**

$\Theta^1 \leftarrow \text{Expand} \left(\frac{1}{2}(\Theta^2 + V + W + Z^1) - \frac{1}{2\rho}(Q^1 + n_1 S^1 + F), \rho, n_1 \right)$;

$\Theta^2 \leftarrow \text{Expand} \left(\frac{1}{2}(\Theta^1 - (V + W) + Z^2) - \frac{1}{2\rho}(Q^2 + n_2 S^2 - F), \rho, n_2 \right)$;

$Z^i \leftarrow \mathcal{T}_1 \left(\Theta^i + \frac{Q^i}{\rho}, \frac{\lambda_i}{\rho} \right)$ for $i = 1, 2$;

$V \leftarrow \mathcal{T}_q \left(\frac{1}{2}(W^T - W + (\Theta^1 - \Theta^2)) + \frac{1}{2\rho}(F - G), \frac{\lambda_1}{2\rho} \right)$;

$W \leftarrow \frac{1}{2}(V^T - V + (\Theta^1 - \Theta^2)) + \frac{1}{2\rho}(F + G^T)$;

$F \leftarrow F + \rho(\Theta^1 - \Theta^2 - (V + W))$;

$G \leftarrow G + \rho(V - W^T)$;

$Q^i \leftarrow Q^i + \rho(\Theta^i - Z^i)$ for $i = 1, 2$

Summary

- We introduce the multi-task sGGMs problem.
- We introduce the challenges of the optimization for this problem.
- We introduce the ADMM method and its drawbacks.

References I

 P. Danaher, P. Wang, and D. M. Witten.

The joint graphical lasso for inverse covariance estimation across multiple classes.

Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2013.

Joint Gaussian Graphical Model Series – VII

Multi-task sGGMs estimators

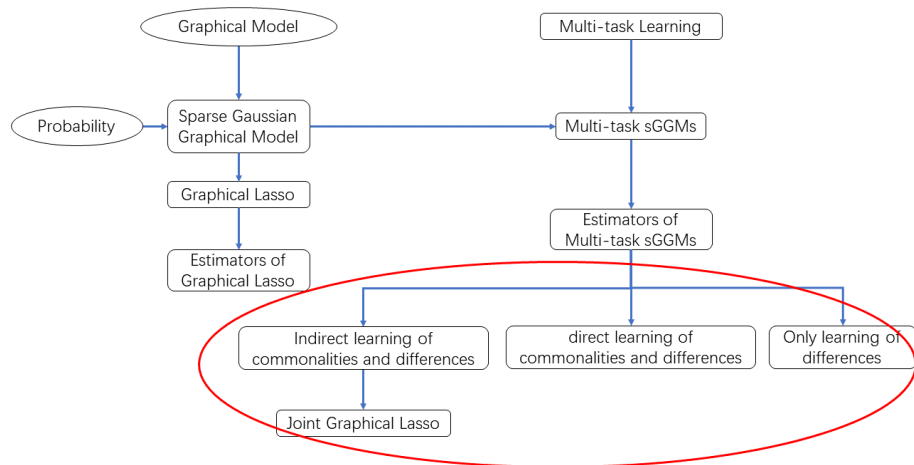
Beilun Wang

Advisor: Yanjun Qi

¹Department of Computer Science, University of Virginia
<http://jointggm.org/>

August 18th, 2017

Road Map



Outline

- 1 Notation
- 2 Review
- 3 Multi-task Learning
- 4 Multi-task sGGMs
- 5 Multi-task sGGMs estimators
 - Joint Graphical Lasso
 - Directly learn the commonalities and differences among tasks
 - Directly learn the differences between case and control

Notation

Notation

$X^{(i)}$ The i -th data matrix

$\Sigma^{(i)}$ The i -th covariance matrix.

$\Omega^{(i)}$ The i -th precision matrix.

p The number of features.

n_i The number of samples in the i -th data matrix.

K The number of tasks.

Review

Review from last talk

- We introduce multi-task learning sparse Gaussian Graphical Models (sGGMs).
- We introduce the optimization challenges in the multi-task sGGMs.
- We introduce the ADMM method and the solution of Joint Graphical Lasso.

Review of Gaussian Graphical Model

Suppose the precision matrix $\Omega = \Sigma^{-1}$.

The log-likelihood of Ω equals to $\ln \det(\Omega) - \text{tr}(\Omega \hat{S})$

Multi-task Learning

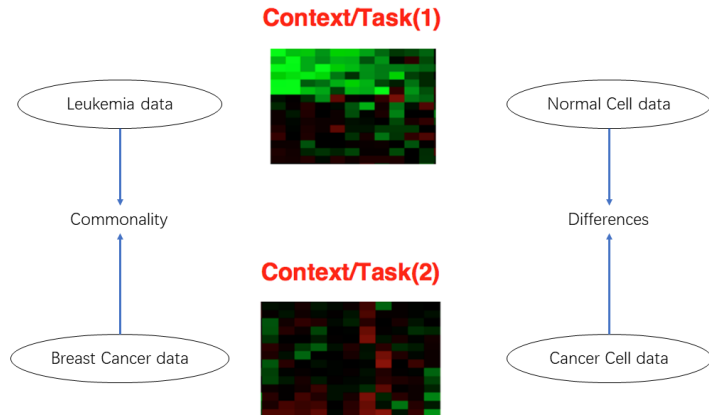
Multi-task Learning

Multi-task Learning

Multi-task learning (MTL) is a subfield of machine learning in which multiple learning tasks are solved at the same time, while exploiting commonalities and differences across tasks.

This can result in improved learning efficiency and prediction accuracy for the task-specific models, when compared to training the models separately.

Multi-task Learning



Multi-task Learning–Linear Classifier Example

Linear Classifier

$$f(x) = \text{sgn}(w^T x + b) \quad (3.1)$$

Multi-task Linear Classifiers

For the i -th task,

$$f_i(x) = \text{sgn}((w_S^T + w_i^T)x + b) \quad (3.2)$$

Multi-task sGGMs

Problem

- Input: $\{X^{(i)}\}$
- Output: $\{\Omega^{(i)}\}$
- Assumption I: Sparsity
- Assumption II: Commonalities and Differences

Multi-task sGGMs

Likelihood

$$\sum_i n_i (\ln \det(\Omega^{(i)}) - \text{tr}(\Omega^{(i)} \widehat{\mathcal{S}}^{(i)})) \quad (4.1)$$

Likelihood with sparsity assumption

$$\operatorname{argmax}_{\Omega^{(i)}} \sum_i n_i (\ln \det(\Omega^{(i)}) - \text{tr}(\Omega^{(i)} \widehat{\mathcal{S}}^{(i)})) \quad (4.2)$$

$$\text{Subject to: } \|\Omega^{(i)}\|_1 \leq t \quad (4.3)$$

Multi-task sGGMs

Likelihood with multi-task setting

$$\operatorname{argmax}_{\Omega^{(i)}} \sum_i n_i (\ln \det(\Omega^{(i)}) - \operatorname{tr}(\Omega^{(i)} \widehat{\mathcal{S}}^{(i)})) \quad (4.4)$$

$$\text{Subject to: } \|\Omega^{(i)}\|_1 \leq t \quad (4.5)$$

$$P(\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(K)}) \leq t_2 \quad (4.6)$$

Joint Graphical Lasso

[Danaher et al.(2013) Danaher, Wang, and Witten]

$$\operatorname{argmin}_{\Omega^{(i)}} - \sum_i n_i (\ln \det(\Omega^{(i)}) + \operatorname{tr}(\Omega^{(i)} \widehat{\mathcal{S}}^{(i)})) + \lambda_1 \|\Omega^{(i)}\|_1 + \lambda_2 P(\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(K)}) \quad (4.7)$$

Multi-task sGGMs estimators

Multi-task sGGMs estimators

- Joint Graphical Lasso type estimators
- Directly learn the commonalities and differences among tasks
- Directly learn the differences between case and control

Joint Graphical Lasso estimators

Different Joint Graphical Lasso

In the end, different multi-task sGGMs estimators choose different $P(\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(K)})$.

Solutions

Most methods use ADMM as the solution of the estimators.

JGL:Problem

- Input: $\{X^{(i)}\}$
- Output: $\{\Omega^{(i)}\}$
- Assumption I: Sparsity
- Assumption II: Commonalities and Differences

Multi-task sGGMs estimators

Group Lasso [Danaher et al. (2013) Danaher, Wang, and Witten]

$$P(\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(K)}) = \|\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(K)}\|_{\mathcal{G}, 2}.$$

SIMONE [Chiquet et al. (2011) Chiquet, Grandvalet, and Ambroise]

$$P(\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(K)}) = \sum_{i \neq j} \left(\left(\sum_{k=1}^T (\Omega_{ij}^{(k)})_+^2 \right) \right)^{\frac{1}{2}} + \left(\left(\sum_{k=1}^K (-\Omega_{ij}^{(k)})_+^2 \right) \right)^{\frac{1}{2}}.$$

Node

JGL [Mohan et al. (2013) Mohan, London, Fazel, Lee, and Witten]

$$P(\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(K)}) = \sum_{ij, i > j} RCON(\Omega^{(i)} - \Omega^{(j)}).$$

Definition 1 The row-column overlap norm (RCON) induced by a matrix norm $\|\cdot\|$ is defined as

$$\Omega(\Theta^1, \Theta^2, \dots, \Theta^K) = \min_{V^1, V^2, \dots, V^K} \left\| \begin{bmatrix} V^1 \\ V^2 \\ \vdots \\ V^K \end{bmatrix} \right\|$$

subject to $\Theta^k = V^k + (V^k)^T$ for $k = 1, \dots, K$.

Directly learn the commonalities and differences among tasks: Problem

- Input: $\{X^{(i)}\}$
- Output: $\{\Omega_I^{(i)}, \Omega_S\}$
- Assumption I: Sparsity
- Assumption II: Commonalities and Differences

Multi-task sGGMs estimators – Direct modeling

The second penalty function is still an indirect way to model the commonality and differences among tasks. Some works try to directly model this relationship.

Mixed Neighborhood Selection

(MSN)[Monti et al.(2015)Monti, Anagnostopoulos, and Montana]

the neighborhood edges of a given node v in the i -task is modeled as $\beta^v + \tilde{\mathbf{b}}^{(i),v}$. Here $\tilde{\mathbf{b}}^{(i),v} \sim N(0, \Phi^v)$.

Consider the CLIME estimator, we can directly model the graphs as the sum of commonality and differences

SIMULE

$$\Omega^{(i)} = \epsilon\Omega_S + \Omega_I^{(i)}.$$

SIMULE

$$\hat{\Omega}_I^{(1)}, \hat{\Omega}_I^{(2)}, \dots, \hat{\Omega}_I^{(K)}, \hat{\Omega}_S = \operatorname{argmin}_{\Omega_I^{(i)}, \Omega_S} \sum_i \|\Omega_I^{(i)}\|_1 + \epsilon K \|\Omega_S\|_1$$

Subject to: $\|\Sigma^{(i)}(\Omega_I^{(i)} + \Omega_S) - I\|_\infty \leq \lambda_n, i = 1, \dots, K$

Multi-task sGGMs estimators – Direct modeling the differential networks: Problem

- Input: $\{X^{(i)}\}$
- Output: $\{\Delta\}$
- Assumption I: Sparse Differential networks

Multi-task sGGMs estimators – Direct modeling the differential networks I

Fused GLasso

By adding a regularization to enforce the sparsity of $\Delta = \Omega_c - \Omega_d$, we have the following formulation:

$$\underset{\Omega_c, \Omega_d \succ 0, \Delta}{\operatorname{argmin}} \mathcal{L}(\Omega_c) + \mathcal{L}(\Omega_d) \lambda_n (\|\Omega_c\|_1 + \|\Omega_d\|_1) + \lambda_2 \|\Delta\|_1 \quad (5.1)$$

The Fused Lasso assumes $\Omega_{case}, \Omega_{control}, \Delta$. However, many real world applications, like brain imaging data, only assume the differential network Δ is sparse.

Direct modeling the differential networks II: Differential CLIME

A recent study proposes the following model, which only assume the sparsity of Δ .

Differential CLIME

$$\begin{aligned} & \underset{\Delta}{\operatorname{argmin}} \|\Delta\|_1 \\ & \text{Subject to: } \|\widehat{\Sigma}_c \Delta \widehat{\Sigma}_d - (\widehat{\Sigma}_c - \widehat{\Sigma}_d)\|_\infty \leq \lambda_n \end{aligned} \tag{5.2}$$

However, this method is solved by a linear programming. It has p^2 variables in this method. Therefore, the time complexity is at least $O(p^8)$. In practice, it takes more than 2 days to finish running the method when $p = 120$.

Direct modeling the differential networks III: Density Ratio

The above methods all make the Gaussian assumption. This method relaxes the model to the exponential family distribution.

Density Ratio

$$\frac{p_c(x, \theta_c)}{p_d(x, \theta_d)} \propto \exp\left(\sum_t \Delta_t f_t(x)\right) \quad (5.3)$$

Here Δ_t encodes the difference between two Networks for factor f_t .

Density Ratio

$$r(x; \theta) = \frac{1}{N(\theta)} \exp\left(\sum_t \Delta_t f_t(x)\right) \quad (5.4)$$

Here Δ_t encodes the difference between two Networks for factor f_t . $N(\theta)$ is a normalization term.

Density Ratio for Markov Random Field

$$\begin{aligned} \hat{p}(x) &= p_d(x)r(x; \theta) \\ \text{KL}[p_c || \hat{p}] &= \text{Const.} - \int p_c(x) \log r(x; \theta) dx. \end{aligned} \tag{5.5}$$

Summary

- We introduce the multi-task sGGMs estimators.
- We introduce the multi-task sGGMs estimators, which directly model the commonalities and differences.
- We introduce the multi-task sGGMs estimators, which directly model the differences.

References I

 J. Chiquet, Y. Grandvalet, and C. Ambroise.

Inferring multiple graphical structures.

Statistics and Computing, 21(4):537–553, 2011.

 P. Danaher, P. Wang, and D. M. Witten.

The joint graphical lasso for inverse covariance estimation across multiple classes.

Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2013.

 K. Mohan, P. London, M. Fazel, S.-I. Lee, and D. Witten.

Node-based learning of multiple gaussian graphical models.

arXiv preprint arXiv:1303.5145, 2013.

References II



R. P. Monti, C. Anagnostopoulos, and G. Montana.

Learning population and subject-specific brain connectivity networks via mixed neighborhood selection.

arXiv preprint arXiv:1512.01947, 2015.

Joint Gaussian Graphical Model Series – VIII

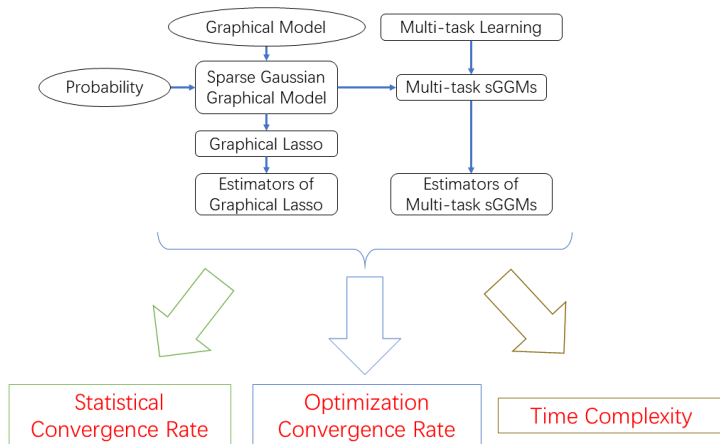
A deep introduction of the metrics for evaluating an/a estimator/learner

Beilun Wang
Advisor: Yanjun Qi

¹Department of Computer Science, University of Virginia
<http://jointggm.org/>

Sep 22nd, 2017

Road Map



Outline

- 1 Notation
- 2 Review
- 3 The metrics for evaluating an estimator
- 4 Statistical Convergence Rate
- 5 Optimization Convergence Rate
- 6 Computational Complexity

Notation

Notation

- X The data matrix
- Σ The covariance matrix.
- Ω The precision matrix.
- p The number of features.
- n The number of samples in the data matrix.
- s The number of non-zero entries in the precision matrix.

Review

Review from last talk

- We introduce different sGGM estimators and their solution.

Review from last talk

- We introduce different sGGM estimators and their solution.
- We briefly introduce the three metrics used in evaluating an estimator.

Review from last talk

- We introduce different sGGM estimators and their solution.
- We briefly introduce the three metrics used in evaluating an estimator.
- We introduce different multi-task sGGMs estimators and their optimization challenges.

The metrics for evaluating an estimator

Motivation I: Select a proper estimator

- There may be a lot of similar estimators.



Motivation I: Select a proper estimator

- There may be a lot of similar estimators.
- You need to decide which one to use.



Motivation I: Select a proper estimator

- There may be a lot of similar estimators.
- You need to decide which one to use.
- You need some metrics to make the decision.



Motivation II: Evaluate a novel method

- You may come out a new estimator.

Motivation II: Evaluate a novel method

- You may come out a new estimator.
- You want to know whether this novel estimator is no worse than the previous ones.

Motivation II: Evaluate a novel method

- You may come out a new estimator.
- You want to know whether this novel estimator is no worse than the previous ones.
- Then you need some metrics to evaluate the estimator.

Background: Two major properties

- Two major properties: **Accuracy** and **Speed**.

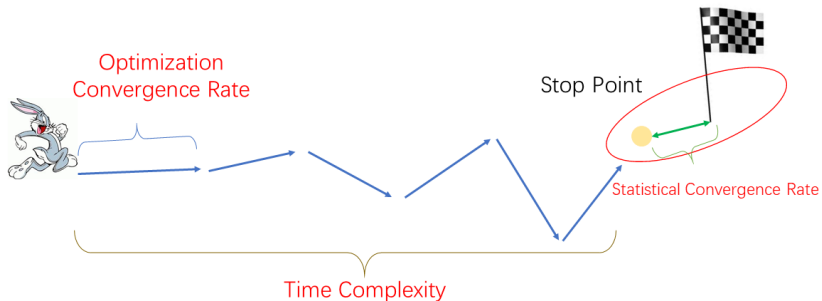
Background: Two major properties

- Two major properties: **Accuracy** and **Speed**.
- Accuracy:
 - ▶ Statistical Convergence rate
 - ▶ how close to the Truth
 - ▶ Statisticians

Background: Two major properties

- Two major properties: **Accuracy** and **Speed**.
- Accuracy:
 - ▶ Statistical Convergence rate
 - ▶ how close to the Truth
 - ▶ Statisticians
- Speed:
 - ▶ Optimization convergence rate
 - ▶ Optimization researchers
 - ▶ Computational complexity
 - ▶ Computer Scientists

Overview Figure



Statistical Convergence Rate

Statistical Convergence Rate : Definition

- The task for an estimator is parameter estimation.

Statistical Convergence Rate : Definition

- The task for an estimator is parameter estimation.
- Suppose the parameter you need to estimate is θ , the truth is θ^*

Statistical Convergence Rate : Definition

- The task for an estimator is parameter estimation.
- Suppose the parameter you need to estimate is θ , the truth is θ^*
- $\| \theta - \theta^* \|$ or $\mathcal{R}(\theta - \theta^*)$

A simple example: Estimate the mean

On the whiteboard.

Elementary

Estimator [Yang et al. (2014b) Yang, Lozano, and Ravikumar]

$$\operatorname{argmin}_{\theta} \mathcal{R}(\theta) \quad (4.1)$$

$$\text{Subject to: } \mathcal{R}^*(\theta - \mathcal{B}^*(\hat{\phi})) \leq \lambda_n \quad (4.2)$$

Here $\mathcal{B}^*(\hat{\phi})$ is a backward mapping for $\hat{\phi}$.

Example: sparse linear regression [Yang et al. (2014a) Yang, Lozano, and Ravikumar]

$$\operatorname{argmin}_{\theta} \|\theta\|_1 \quad (4.3)$$

$$\text{Subject to: } \|\theta - (X^T X + \epsilon I)^{-1} X^T y\|_{\infty} \leq \lambda_n \quad (4.4)$$

Hands on: Elementary Estimator for high-dimensional linear regression

On the whiteboard.

Hands on: DIFFEE

On the whiteboard.

Conclusions

- In high-dimensional setting, related to $\frac{\log p}{n}$.
- Equivalent estimators still have differences in constants or constraints

Optimization Convergence Rate

Optimization Convergence Rate : Definition

- Linearly Converge: $\lim_{k \rightarrow \infty} \frac{|\theta_{k+1} - L|}{|\theta_k - L|} = \mu_k$

Optimization Convergence Rate : Definition

- Linearly Converge: $\lim_{k \rightarrow \infty} \frac{|\theta_{k+1} - L|}{|\theta_k - L|} = \mu_k$
- - ▶ Linearly, if $\mu_k \in (0, 1)$
 - ▶ Superlinearly, if $\mu_k \rightarrow 0$ when $k \rightarrow \infty$.
 - ▶ Sublinearly, if $\mu_k \rightarrow 1$ when $k \rightarrow \infty$

Optimization Convergence Rate : Definition

- Linearly Converge: $\lim_{k \rightarrow \infty} \frac{|\theta_{k+1}-L|}{|\theta_k-L|} = \mu_k$
- - ▶ Linearly, if $\mu_k \in (0, 1)$
 - ▶ Superlinearly, if $\mu_k \rightarrow 0$ when $k \rightarrow \infty$.
 - ▶ Sublinearly, if $\mu_k \rightarrow 1$ when $k \rightarrow \infty$
- Higher order: $\lim_{k \rightarrow \infty} \frac{|x_{k+1}-L|}{|x_k-L|^q} > 0$.

Optimization Convergence Rate : Definition

- Linearly Converge: $\lim_{k \rightarrow \infty} \frac{|\theta_{k+1}-L|}{|\theta_k-L|} = \mu_k$
 - ▶ Linearly, if $\mu_k \in (0, 1)$
 - ▶ Superlinearly, if $\mu_k \rightarrow 0$ when $k \rightarrow \infty$.
 - ▶ Sublinearly, if $\mu_k \rightarrow 1$ when $k \rightarrow \infty$
- Higher order: $\lim_{k \rightarrow \infty} \frac{|x_{k+1}-L|}{|x_k-L|^q} > 0$.
- Closed form solution

Optimization Convergence Rate : Definition

- Linearly Converge: $\lim_{k \rightarrow \infty} \frac{|\theta_{k+1}-L|}{|\theta_k-L|} = \mu_k$
 - ▶ Linearly, if $\mu_k \in (0, 1)$
 - ▶ Superlinearly, if $\mu_k \rightarrow 0$ when $k \rightarrow \infty$.
 - ▶ Sublinearly, if $\mu_k \rightarrow 1$ when $k \rightarrow \infty$
- Higher order: $\lim_{k \rightarrow \infty} \frac{|x_{k+1}-L|}{|x_k-L|^q} > 0$.
- Closed form solution
- Closed form \geq Higher order \geq linear

Optimization Convergence Rate: Basic Results

- Gradient Descent based method: Linear

Optimization Convergence Rate: Basic Results

- Gradient Descent based method: Linear
- - ▶ gradient descent
 - ▶ SGD
 - ▶ ADMM / proximal gradient descent

Optimization Convergence Rate: Basic Results

- Gradient Descent based method: Linear
 - ▶ gradient descent
 - ▶ SGD
 - ▶ ADMM / proximal gradient descent
- Newton method based method: Quadratic

Optimization Convergence Rate: Basic Results

- Gradient Descent based method: Linear
 - ▶ gradient descent
 - ▶ SGD
 - ▶ ADMM / proximal gradient descent
- Newton method based method: Quadratic
- Elementary Estimator: Closed form solution

Optimization Convergence Rate: Different methods

	Single sGGM			Multiple sGGMs	
Method:	GLasso	CLIME	EEGM	JGL	FASJEM
Rate of Convergence	Linear	NA	Closed form	Linear	Linear

Computational Complexity

Computational Complexity: Definition

- Complexity of an algorithm is the amount of resources required for running it.

Computational Complexity: Definition

- Complexity of an algorithm is the amount of resources required for running it.
- In machine learning, it is mainly related to n and p .

Computational Complexity: Definition

- Complexity of an algorithm is the amount of resources required for running it.
- In machine learning, it is mainly related to n and p .
- Use big O notation

Computational Complexity: how to calculate

- Some cases:

- ▶ Matrix Multiplication: $O(np^2)$
- ▶ Matrix inversion $O(p^3)$
- ▶ SVD inversion $O(p^3)$
- ▶ soft-thresholding $O(p^2)$

Computational Complexity: how to calculate

- Some cases:
 - ▶ Matrix Multiplication: $O(np^2)$
 - ▶ Matrix inversion $O(p^3)$
 - ▶ SVD inversion $O(p^3)$
 - ▶ soft-thresholding $O(p^2)$
- How to calculate:
 - ▶ Num of Iter \times Computational complexity of each Iter
 - ▶ Direct calculate e.g., Closed form solution
 - ▶ Use existing method e.g., linear programming
 - ▶ Special case: linear convergence.

Computational Complexity: Different methods

	Single sGGM			Multiple sGGMs		
Method:	GLasso	CLIME	EEGM	JGL	FASJEM	SIMUL
Computational Complexity	$O(Tp^2)$	$O(p^5)$	$O(p^2)$	$O(Tp^3)$	$O(Tp^2)$	$O(K^4 p^5)$

Summary

- We introduce the statistical convergence rate.
- We introduce the optimization convergence rate.
- We introduce the computational complexity.

References I



E. Yang, A. Lozano, and P. Ravikumar.

Elementary estimators for high-dimensional linear regression.

In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 388–396, 2014a.



E. Yang, A. C. Lozano, and P. K. Ravikumar.

Elementary estimators for graphical models.

In *Advances in Neural Information Processing Systems*, pages 2159–2167, 2014b.