Fast and Scalable Joint Estimators for Learning Sparse Gaussian Graphical Models from Heterogeneous Data with Additional Knowledge

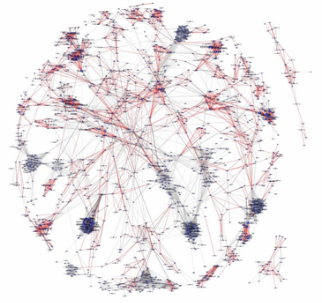
Beilun Wang[1]
Advisor: Yanjun Qi[1]
Dissertation Committees:
Mahmoody Mohammad (Committee Chair)[1]
Xiaojin (Jerry) Zhu[2]
Farzad Farnoud[1]
Tingting Zhang[1]

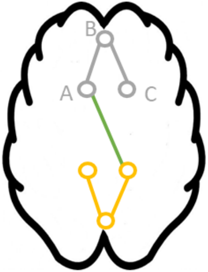
[1] University of Virginia
[2] University of Wisconsin–Madison

August 24, 2018

Background

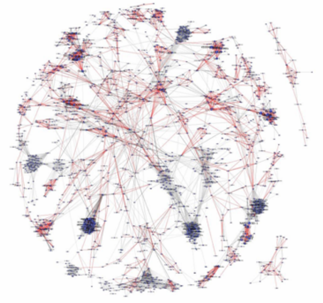
Background: Entity Graph

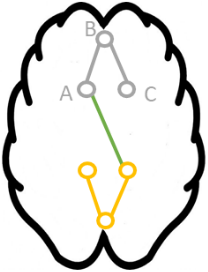




- Many applications need to know interactions among entities:
  - Gene Interactions
  - Brain connectivity

Background: Entity Graph



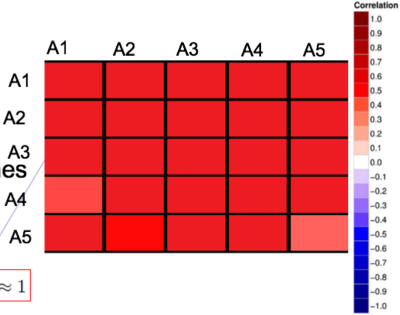


- Many applications need to know interactions among entities:
  - Gene Interactions
  - Brain connectivity

- Why to study the entity graph
  - Understanding
  - Diagnosis, e.g., marker
  - Treatment, e.g., drug development.

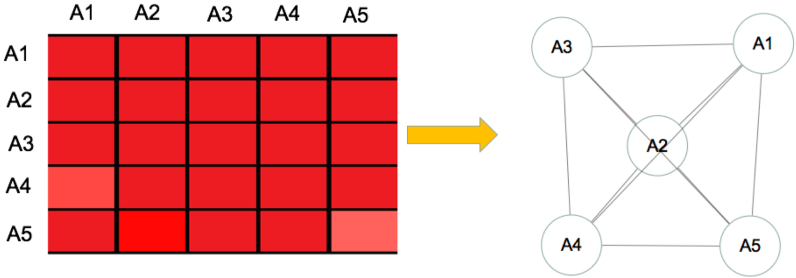
Background: What Type of Edges? Correlation to Conditional dependency

A1: Children swim
A2: Weather is hot
A3: High sale of ice cream
A4: Wear less amount of clothes
A5: High Electricity Consumption

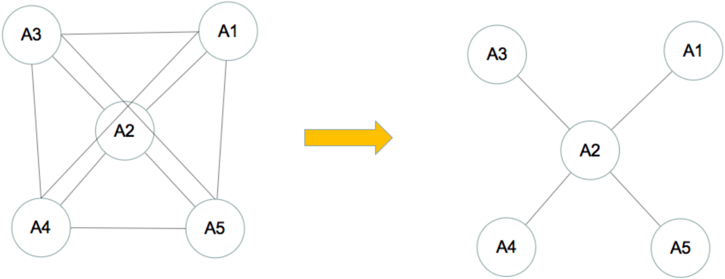


$\boxed{Cor(A_1, A_3) \approx 1}$

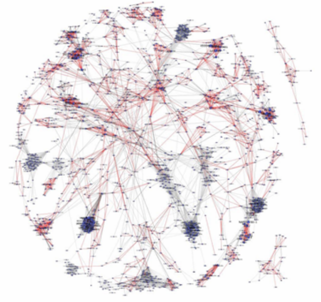
Background: What Type of Edges? Correlation to Conditional dependency

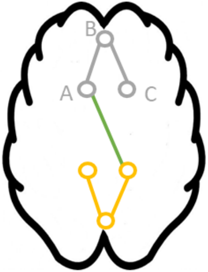


Background: What Type of Edges? Correlation to Conditional dependency



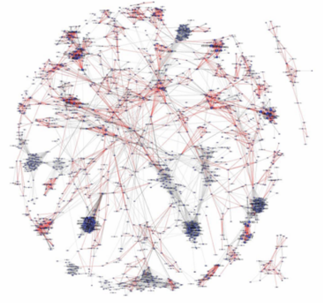
Background: How to Infer Entity Graph?

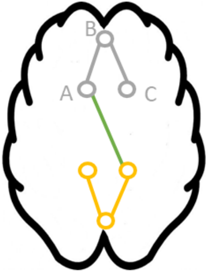




- To measure conditional dependency interactions physically.
- Largely unknown and hard to measure physically.

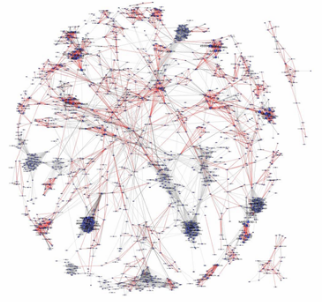
Background: How to Infer Entity Graph?

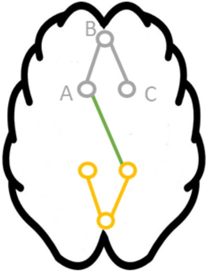




- To measure conditional dependency interactions physically.
- Largely unknown and hard to measure physically.
    - #Physical check for all possible conditional dependency edges = 2^p (binary experiments)
    - For example, $p = 160$ important regions in human brain
    - For example, $p = 30000$ genes in human cell

Background: How to Infer Entity Graph?

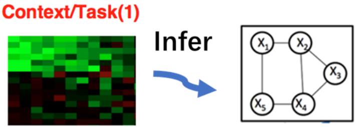




- To measure conditional dependency interactions physically.
- Largely unknown and hard to measure physically.
  - #Physical check for all possible conditional dependency edges = 2^p (binary experiments)
  - For example, $p = 160$ important regions in human brain
  - For example, $p = 30000$ genes in human cell
  - Much more than Trillions (2^{40}) of biological experiments

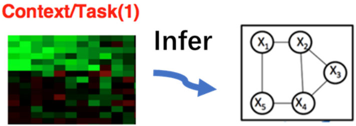
Background: Entity graphs from Observed Samples (Entity as Feature)

- ~~Trillions of biological experiments~~ ⟹ Data-driven approach
- Experiments (not physically check) ⟹ Data ⟹ Entity Graph



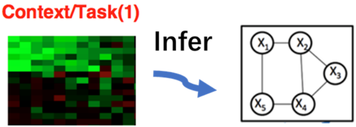
Background: Entity graphs from Observed Samples (Entity as Feature)

- ~~Trillions of biological experiments~~ $\Longrightarrow$ Data-driven approach
- Experiments (not physically check) $\Longrightarrow$ Data $\Longrightarrow$ Entity Graph
- *n* experiments $\rightarrow$ *n* data samples
  - Each sample is a snapshot of all the entities.
  - Each sample has measurements of *p* features/entities.



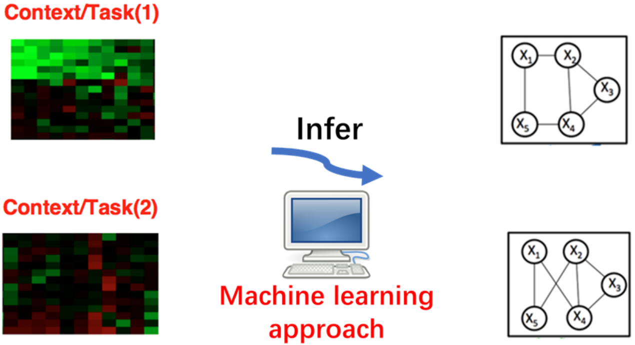
Background: Entity graphs from Observed Samples (Entity as Feature)

- ~~Trillions of biological experiments~~ $\Longrightarrow$ Data-driven approach
- Experiments (not physically check) $\Longrightarrow$ Data $\Longrightarrow$ Entity Graph
- n experiments $\rightarrow$ n data samples
    - Each sample is a snapshot of all the entities.
    - Each sample has measurements of p features/entities.
- n data samples is enough $\rightarrow$ a well estimated entity graph of p when $n >> p$ (low-dimensional).
- $p > n$ (high-dimensional) needs novel approaches



Background: Entity graphs from Heterogeneous Data (Entity as Feature)

- Most applications have heterogeneous samples.
- For example:
  - Totally n_{tot} data samples
  - From K different but related contexts, each has n_i data samples



Context/Task(1)

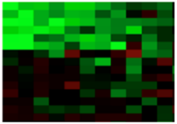
Context/Task(2)

Infer

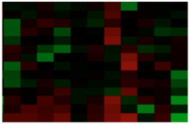
Machine learning approach

Background: Entity graphs from Heterogeneous Data

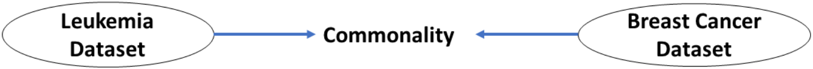
Context/Task(1)



Context/Task(2)



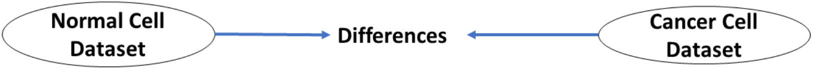
Case I:


Leukemia Dataset

Commonality

Breast Cancer Dataset

Case II:

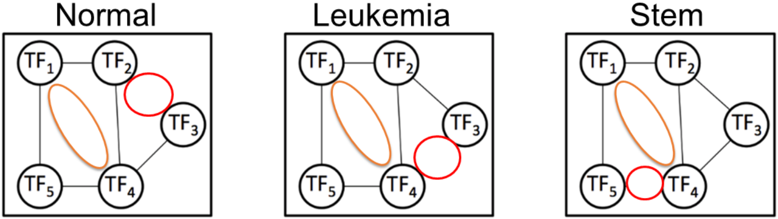

Normal Cell Dataset

Differences

Cancer Cell Dataset

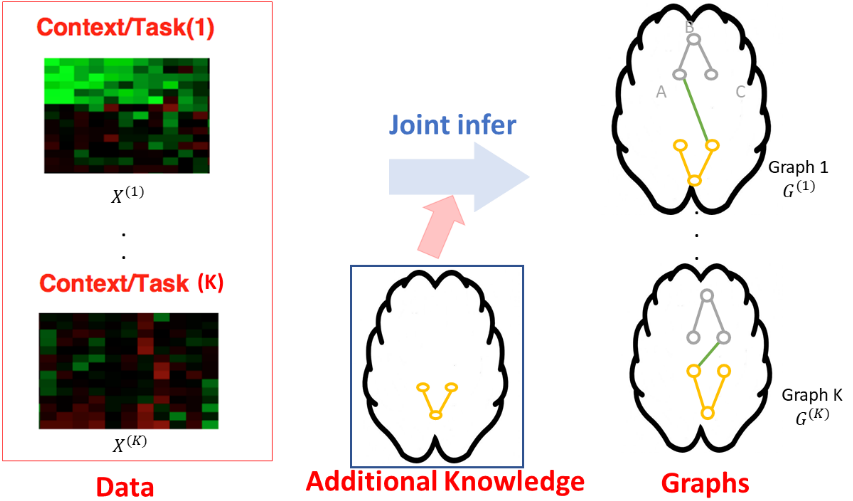
Task I: Learning multiple related graphs

- Learning multiple related graphs
- E.g., TF-TF interactions
  - Three graphs are similar



Task II: Integrating additional knowledge

- Integrating known knowledge in Learning multiple related graphs
  - E.g., known knowledge in Brain Connection



Context/Task(1)

$X^{(1)}$

Context/Task (K)

$X^{(K)}$

Data

Joint infer

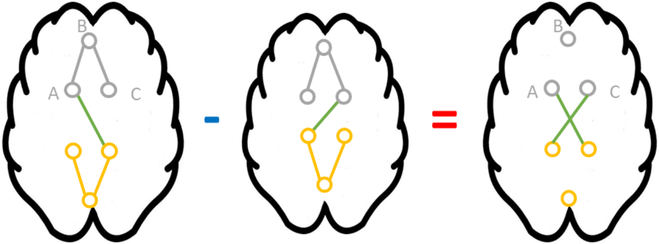
Additional Knowledge

Graph 1 $G^{(1)}$

Graph K $G^{(K)}$

Graphs

Task III: Learning sparse changes between two graphs



- A very interesting task:
  - Find differences in the brains of people with diseases, e.g. Autism, Alzheimer's
  - Use for understanding
  - Use for diagnosis

Notations

$X^{(i)}$ *i*-th Data matrix.

$\Sigma^{(i)}$ *i*-th Covariance matrix.

$\Omega^{(i)}$ *i*-th Inverse of covariance matrix (precision matrix).

p The total number of feature variables.

n_{tot} The total number of samples.

X^{tot} the concatenation of all Data matrices.

Σ^{tot} the concatenation of all Covariance matrices.

Ω^{tot} the concatenation of all Inverse of covariance matrices (precision matrices).

W_I^{tot} $(W_I^{(1)}, W_I^{(2)}, \ldots, W_I^{(K)})$

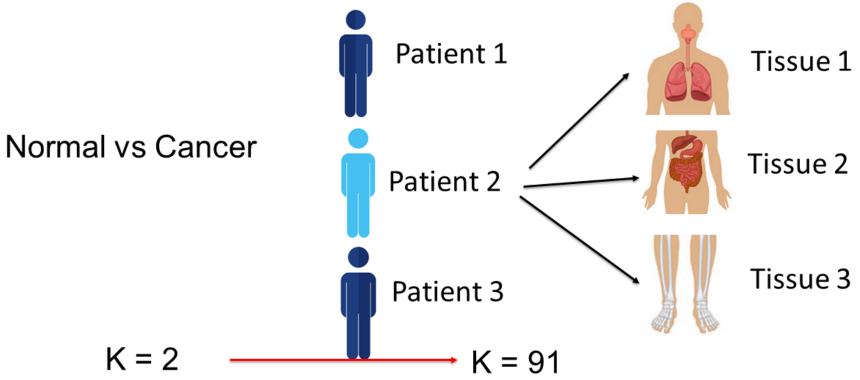
W_S^{tot} $(W_S, W_S, \ldots, W_S)$

Motivation

Motivation: More Num of features (p) to consider

- Yeast gene: 6K
  $\downarrow$
  Human gene: 30K
- Words interaction, millions of words ($p > 1,000,000$)

Motivation: More num of tasks (K) to consider



ENCODE Project Consortium et al. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57–74, 2012.

Motivation: Limitation I – Slow Computation

The best baseline of	Task I	Task II	Task III
Computational complexity	$O(Kp^3)$ / iter	$O(K^4p^5)$	$O(p^3)$ / iter
Bottle neck	SVD	Linear programming	SVD

- If $K = 91$ and $p = 30$K $\Downarrow$

The best baseline of	Task I	Task II	Task III
Time	3.5 days / iter	6 trillion years	1 hour/ iter

- Can we have a $O(p^2)$ method?

Motivation: Limitation II – No consideration of parallelization

✖

Computer Clusters

✖

GPU

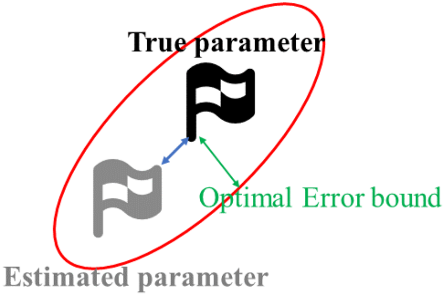
- Reduce $O(p^2)$ to $O(1)$.

Motivation: Limitation III: Lack of error bound analysis

- $||\widehat{\theta} - \theta^*||$

- Missing analysis under a high-dimensional setting ($p \geq n$)

- No sacrifices of the accuracy from speeding-up and scaling-up the algorithm



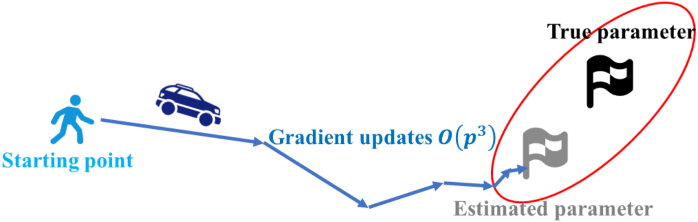
Our Aim: Fast and Scalable estimators for three types of joint graphs estimation

- Fast and scalable estimators for the three tasks
- Parallelizable algorithms
- Integrating additional knowledge
- Sharp convergence rate

Solution for Limitations - Elementary Estimator

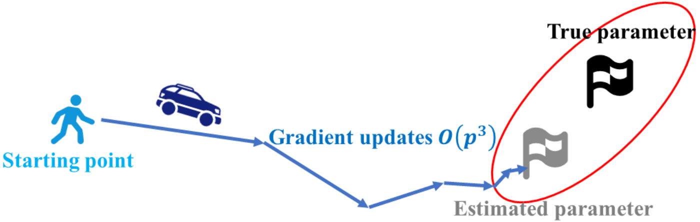
Background: summary of the previous optimization strategy

- e.g., ADMM algorithm

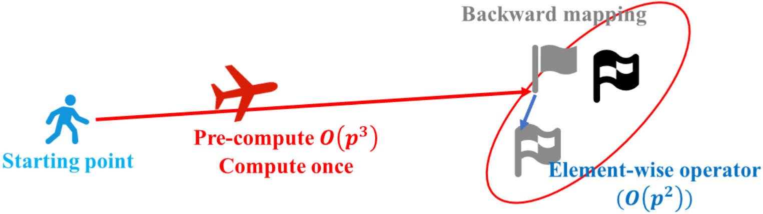


Elementary Estimator (EE) for joint sGGMs tasks

- Previous studies:

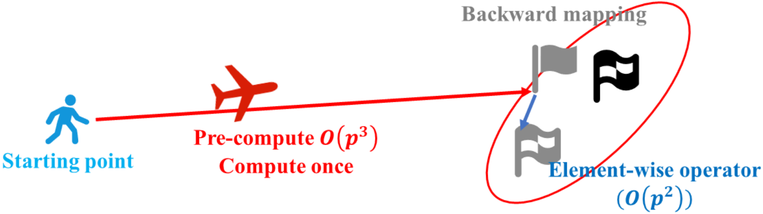


- Elementary Estimator:



Elementary Estimator (EE): Step I – Backward mapping

- Backward mapping $\mathcal{B}^*(\widehat{\phi})$ of the parameter (Solution of Vanilla Maximum Likelihood Estimator (MLE))
- Vanilla MLE: $\underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\theta)$
  - Already close to true parameter
  - But without assumptions e.g., sparse
  - For instance, linear regression solution $(X^TX)^{-1}X^TY$



Elementary Estimator: Step II – Optimization formulation

Elementary Estimator (EE)

$$\underset{\theta}{\operatorname{argmin}} \mathcal{R}(\theta) \\ \text{Subject to: } \mathcal{R}^*(\theta - \mathcal{B}^*(\widehat{\phi})) \leq \lambda_n \tag{3.1}$$

- Let $\mathcal{R}(\cdot) = \| \cdot \|_1$

$$\Downarrow$$

$$\underset{\theta}{\operatorname{argmin}} ||\theta||_1 \\ \text{Subject to: } ||\theta - \mathcal{B}^*(\widehat{\phi})||_\infty \leq \lambda_n \tag{3.2}$$

- Easy to prove the sharp convergence rate when $\mathcal{R}$ and $\mathcal{B}^*$ satisfy certain conditions.

EE-Benefit: Fast and scalable solution

- A soft-thresholding operator (closed form)
- Closed form & $O(p^2)$
- Easy to parallelize in GPU

$$\widehat{\theta} = S_{\lambda_n}(\mathcal{B}^*(\widehat{\phi}))$$

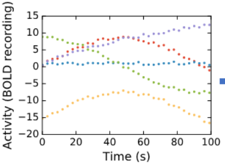
$$[S_\lambda(A)]_{ij} = \text{sign}(A_{ij})\max(|A_{ij}| - \lambda, 0) \tag{3.3}$$

- Element-wise

$$\Sigma\text{=Cov(X)=}\begin{bmatrix} \boxed{\sigma_{11}} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix} \quad \Sigma\text{=Cov(X)=}\begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \boxed{\sigma_{21}} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix} \quad \Sigma\text{=Cov(X)=}\begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \boxed{\sigma_{nn}} \end{bmatrix}$$

Apply same operator
Independent calculation

Background: sparse Gaussian Graphical Model (sGGM) to derive Conditional Independence Graph from data



Data

1.05	-0.23	0.05	-0.02	0.05
-0.23	1.45	-0.25	0.10	-0.25
0.05	-0.25	1.10	-0.24	0.10
-0.02	0.10	-0.24	1.10	-0.24
0.05	-0.25	0.10	-0.24	1.10

Covariances matrix Σ

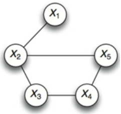
sGGM

1	0.2	0	0	0
0.2	1	0.2	0	0.2
0	0.2	1	0.2	0
0	0	0.2	1	0.2
0	0.2	0	0.2	1

Sparse inversion Ω
(Precision Matrix)

Decode

Sparsity pattern



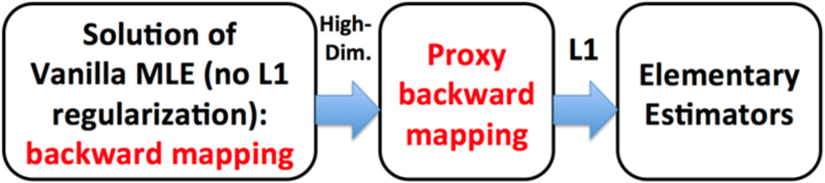
Connectome

EE-GM: Elementary Estimator for sGGM

- Vanilla MLE: $\underset{\Omega}{\operatorname{argmin}} - \log(\det(\Omega)) + < \Omega, \Sigma >$
- Backward mapping of Ω is Σ^{-1}
- Not invertible when $p \geq n$

EE-GM: Elementary Estimator for sGGM

- Vanilla MLE: $\underset{\Omega}{\operatorname{argmin}} - \log(\det(\Omega)) + < \Omega, \Sigma >$
- Backward mapping of Ω is Σ^{-1}
- Not invertible when $p \geq n$
- Need apporximated backward mapping
  - proxy backward mapping $\widehat{\theta}_n \approx \mathcal{B}^*(\widehat{\phi})$
  - In sGGM, $\widehat{\theta}_n = [T_v(\widehat{\Sigma})]^{-1}$



Solution of Vanilla MLE (no L1 regularization): backward mapping

High-Dim.

Proxy backward mapping

L1

Elementary Estimators

EE-GM: Elementary Estimator for sGGM

$$\underset{\theta}{\operatorname{argmin}} ||\theta||_1 \quad \text{Subject to: } ||\theta - \mathcal{B}^*(\widehat{\phi})||_\infty \le \lambda_n \tag{3.4}$$

$\widehat{\theta}_n = [T_v(\widehat{\Sigma})]^{-1}$

$$\Downarrow$$

EE-sGGM

$$\underset{\Omega}{\operatorname{argmin}} ||\Omega||_{1,,\text{off}} \quad \text{subject to:} ||\Omega - [T_v(\widehat{\Sigma})]^{-1}||_{\infty,\text{off}} \le \lambda_n \tag{3.5}$$

EE	$\mathcal{R}(\cdot)$	θ	$\widehat{\theta}_n$	$\mathcal{R}^*$
EE-sGGM	$\|\| \cdot \|\|_1$	Ω	$[T_v(\widehat{\Sigma})]^{-1}$	$\|\| \cdot \|\|_\infty$

EE-Benefit: Easy to prove error bound

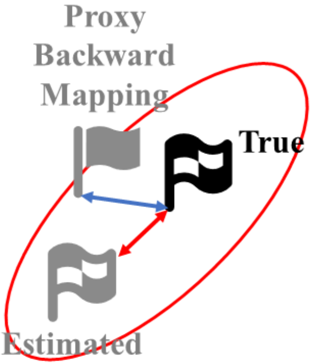
- Error bound:

$$
\begin{aligned}
||\widehat{\theta} - \theta^*||_\infty &\leq 2\lambda_n \\
||\widehat{\theta} - \theta^*||_F &\leq 4\sqrt{s}\lambda_n \qquad (3.6) \\
||\widehat{\theta} - \theta^*||_1 &\leq 8s\lambda_n
\end{aligned}
$$

- Condition:

$$\lambda_n \geq ||\widehat{\theta}_n - \theta^*||_\infty \qquad (3.7)$$

- Constant: s is the num of non-zero entries.



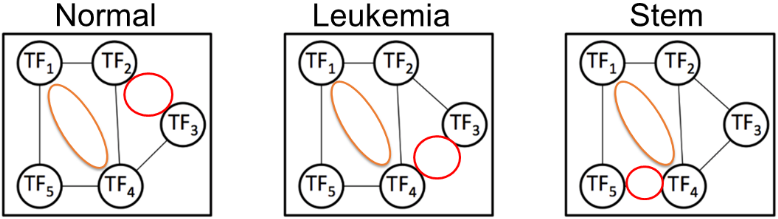
Method I: FASJEM

Outline

1. Background
2. Motivation
3. Solution for Limitations - Elementary Estimator
4. Method I: FASJEM
    - Background
    - Method
    - Results
5. Method II: JEEK
    - Background
    - Method
    - Results
6. Method III: DIFFEE
    - Method
    - Results
7. Discussion
    - Questions from Proposal
    - Future works

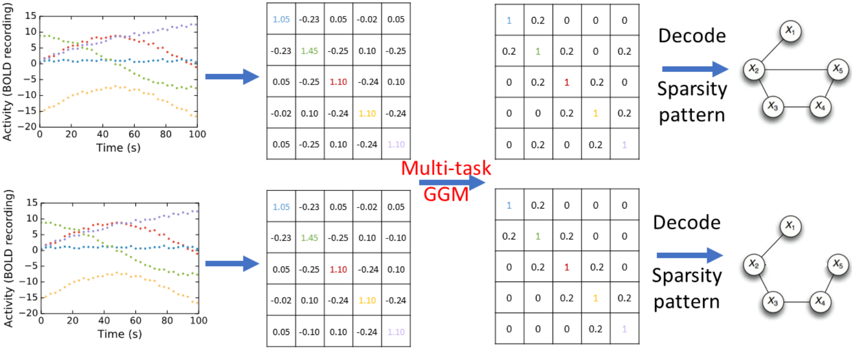
Task I: Learning multiple related graphs

- Learning multiple related graphs
- E.g., TF-TF interactions
    - Three graphs are similar



Background: Multi-task sGGMs

- A pipeline to infer Multiple Related Graphs from heterogeneous datasets $\mathbf{X}^{(1)},\ldots\mathbf{X}^{(K)}$[1].



1.05	-0.23	0.05	-0.02	0.05
-0.23	1.45	-0.25	0.10	-0.25
0.05	-0.25	1.10	-0.24	0.10
-0.02	0.10	-0.24	1.10	-0.24
0.05	-0.25	0.10	-0.24	1.10

1	0.2	0	0	0
0.2	1	0.2	0	0.2
0	0.2	1	0.2	0
0	0	0.2	1	0.2
0	0.2	0	0.2	1

Decode

Sparsity pattern

Multi-task GGM

1.05	-0.23	0.05	-0.02	0.05
-0.23	1.45	-0.25	0.10	-0.10
0.05	-0.25	1.10	-0.24	0.10
-0.02	0.10	-0.24	1.10	-0.24
0.05	-0.10	0.10	-0.24	1.10

1	0.2	0	0	0
0.2	1	0.2	0	0
0	0.2	1	0.2	0
0	0	0.2	1	0.2
0	0	0	0.2	1

Decode

Sparsity pattern

[1] X^{tot}: the concatenation of $(X^{(1)}, X^{(2)}, \ldots, X^{(K)})$.
Σ^{tot}: the concatenation of $(\Sigma^{(1)}, \Sigma^{(2)}, \ldots, \Sigma^{(K)})$.
Ω^{tot}: the concatenation of $(\Omega^{(1)}, \Omega^{(2)}, \ldots, \Omega^{(K)})$.

Background: Joint Graphical Lasso

Graphical Lasso

$$\underset{\Omega}{\operatorname{argmin}} - \log\det(\Omega) + < \Omega, \Sigma > + \lambda_n ||\Omega||_1 \tag{4.1}$$

- Add $\mathcal{R}'(\cdot)$ $\Downarrow$

Joint Graphical Lasso

$$\underset{\Omega^{(i)}>0}{\operatorname{argmin}} \sum_i (-L(\Omega^{(i)}) + \lambda_1 \sum_i ||\Omega^{(i)}||_1 + \lambda_2 \mathcal{R}'(\Omega^{(1)}, \Omega^{(2)}, \ldots, \Omega^{(K)}) \tag{4.2}$$

- $\Omega_{tot} = (\Omega^{(1)}, \Omega^{(2)}, \ldots, \Omega^{(K)})$.

Outline

1. Background
2. Motivation
3. Solution for Limitations - Elementary Estimator
4. Method I: FASJEM
   - Background
   - Method
   - Results
5. Method II: JEEK
   - Background
   - Method
   - Results
6. Method III: DIFFEE
   - Method
   - Results
7. Discussion
   - Questions from Proposal
   - Future works

Enforcing relatedness of multiple graphs through Regularization: FASJEM-norm

EE-sGGM

$$\underset{\Omega}{\text{argmin}} ||\Omega||_{1,,\text{off}}$$
$$\text{subject to:} ||\Omega - [T_v(\widehat{\Sigma})]^{-1}||_{\infty,\text{off}} \leq \lambda_n \tag{4.3}$$

- Add $\mathcal{R}'(\cdot)$

$\Downarrow$

FASJEM-norm

$$\mathcal{R}(\Omega_{tot}) = ||\Omega_{tot}||_1 + \mathcal{R}'(\Omega_{tot}) \tag{4.4}$$

Elementary Estimator (EE)

$$\underset{\theta}{\operatorname{argmin}} \mathcal{R}(\theta)$$
$$\text{Subject to: } \mathcal{R}^*(\theta - \mathcal{B}^*(\widehat{\phi})) \leq \lambda_n \tag{4.5}$$

EE	$\mathcal{R}(\cdot)$	θ	$\widehat{\theta}_n$	$\mathcal{R}^*(\cdot)$
EE-sGGM	$\|\|\cdot\|\|_1$	Ω	$[T_v(\widehat{\Sigma})]^{-1}$	$\|\|\cdot\|\|_\infty$
FASJEM	$\|\|\cdot\|\|_1 + \mathcal{R}'$	Ω^{tot}	$inv[T_v(\widehat{\Sigma}^{tot})]$	$\max(\|\|\cdot\|\|_\infty, \mathcal{R}'^*)$

FASJEM

$$\begin{aligned} &\underset{\Omega_{tot}}{\operatorname{argmin}} \,||\Omega_{tot}||_1 + \mathcal{R}'(\Omega_{tot}) \\ &s.t. ||\Omega_{tot} - \text{inv}(T_v(\widehat{\Sigma}_{tot}))||_\infty \leq \lambda_n \\ &\mathcal{R}'^*(\Omega_{tot} - \text{inv}(T_v(\widehat{\Sigma}_{tot}))) \leq \lambda_n \end{aligned} \tag{4.6}$$

FASJEM: Variations

- FASJEM-G:

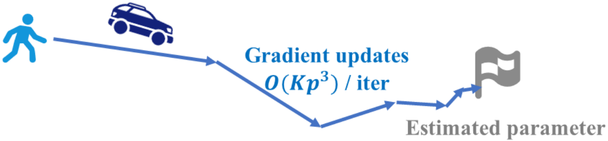
$$\begin{aligned} &\mathcal{R}'(\cdot) = ||\cdot||_{\mathcal{G},2} \\ &||\Omega_{tot}||_{\mathcal{G},2} = \sum_{j=1}^{p}\sum_{k=1}^{p} ||(\Omega_{j,k}^{(1)}, \Omega_{j,k}^{(2)}, \ldots, \Omega_{j,k}^{(i)}, \ldots, \Omega_{j,k}^{(K)})||_2 \end{aligned} \tag{4.7}$$

- FASJEM-I:

$$\begin{aligned} &\mathcal{R}'(\cdot) = ||\cdot||_{\mathcal{G},\infty} \\ &||\Omega_{tot}||_{\mathcal{G},\infty} = \sum_{j=1}^{p}\sum_{k=1}^{p} ||(\Omega_{j,k}^{(1)}, \Omega_{j,k}^{(2)}, \ldots, \Omega_{j,k}^{(i)}, \ldots, \Omega_{j,k}^{(K)})||_\infty \end{aligned} \tag{4.8}$$

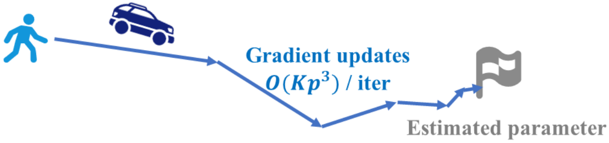
FASJEM: Optimization Solution

- JGL solution:

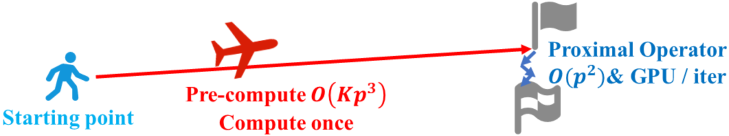


FASJEM: Optimization Solution

- JGL solution:

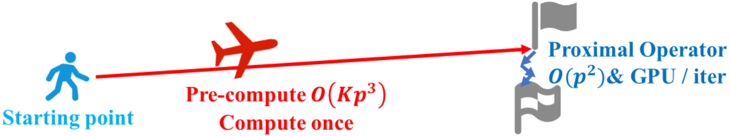


- FASJEM solution:



FASJEM: Optimization Solution – Proximal algorithm

- FASJEM solution:



Starting point

Pre-compute $O(Kp^3)$

Compute once

Proximal Operator

$O(p^2)$& GPU / iter

- In each iteration, a proximal operator
- Element-wise operator, $O(p^2)$

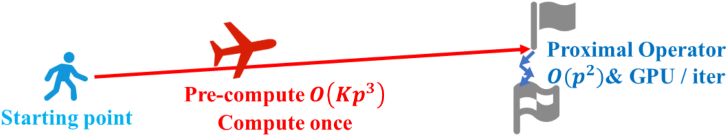
$$\mathrm{prox}_{\gamma||\cdot||_1}(x) = \begin{cases} x_{j,k}^{(i)} - \gamma,\ x_{j,k}^{(i)} > \gamma \\ 0,\ |x_{j,k}^{(i)}| \leq \gamma \\ x_{j,k}^{(i)} + \gamma,\ x_{j,k}^{(i)} < -\gamma \end{cases} \tag{4.9}$$

$$\Longrightarrow$$

$$\mathrm{prox}_{\gamma||\cdot||_1}(x) = \max((x_{j,k}^{(i)} - \gamma), 0) + \min(0, (x_{j,k}^{(i)} + \gamma)) \tag{4.10}$$

FASJEM: Optimization Solution – Proximal algorithm

- FASJEM solution:



- In each iteration, a proximal operator
- Element-wise operator, $O(p^2)$
- GPU-parallelizable $O(1)$
  - e.g., proximity of ℓ_1

$$\mathrm{prox}_{\gamma||\cdot||_1}(x) = \begin{cases} x^{(i)}_{j,k} - \gamma,\ x^{(i)}_{j,k} > \gamma \\ 0,\ |x^{(i)}_{j,k}| \leq \gamma \\ x^{(i)}_{j,k} + \gamma,\ x^{(i)}_{j,k} < -\gamma \end{cases} \tag{4.9}$$

$$\Longrightarrow$$

$$\mathrm{prox}_{\gamma||\cdot||_1}(x) = \max((x^{(i)}_{j,k} - \gamma), 0) + \min(0, (x^{(i)}_{j,k} + \gamma)) \tag{4.10}$$

FASJEM: Computational Complexity

The best baseline of	Task I	Task II	Task III
Computational complexity	$O(Kp^3)$ / iter	$O(K^4 p^5)$	$O(p^3)$ / iter
Bottle neck	SVD	Linear programming	SVD
Our approach	FASJEM		
Computational complexity	$O(Kp^2)$ / iter		
Parallelization	$O(K)$ / iter		

Summary

	EE	$\mathcal{R}(\cdot)$	θ	$\widehat{\theta}_n$	$\mathcal{R}^*(\cdot)$
	EE-sGGM	$\|\|\cdot\|\|_1$	Ω	$[T_v(\widehat{\Sigma})]^{-1}$	$\|\|\cdot\|\|_\infty$
Task I	FASJEM	$\|\|\cdot\|\|_1 + \mathcal{R}'$	Ω^{tot}	$inv[T_v(\widehat{\Sigma}^{tot})]$	$\max(\|\|\cdot\|\|_\infty, \mathcal{R}'^*)$
Task II					
Task III					

Outline

1. Background
2. Motivation
3. Solution for Limitations - Elementary Estimator
4. **Method I: FASJEM**
   - Background
   - Method
   - **Results**
5. Method II: JEEK
   - Background
   - Method
   - Results
6. Method III: DIFFEE
   - Method
   - Results
7. Discussion
   - Questions from Proposal
   - Future works

Results: Theoretical Analysis

- $p' = \max(Kp, n_{tot})$
- Error Bound: $||\widehat{\Omega}_{tot} - \Omega^*_{tot}||_F \leq 32\frac{4\kappa_1 a}{\kappa_2}\sqrt{\frac{s \log p'}{n_{tot}}}$

Multi-task:	K Single-task:
$O(\frac{\log(Kp)}{n_{tot}})$	$O(\frac{\log p}{n_i}))$

- By assuming $n_i = \frac{n_{tot}}{K}$:

Results: Theoretical Analysis

- $p' = \max(Kp, n_{tot})$
- Error Bound: $||\widehat{\Omega}_{tot} - \Omega^*_{tot}||_F \leq 32\frac{4\kappa_1 a}{\kappa_2}\sqrt{\frac{s \log p'}{n_{tot}}}$

Multi-task:	K Single-task:
$O(\frac{\log(Kp)}{n_{tot}})$	$O(\frac{\log p}{n_i}))$

- By assuming $n_i = \frac{n_{tot}}{K}$:
- We can conclude that $\frac{\log(Kp)}{n_{tot}} < K\frac{\log p}{n_{tot}}$

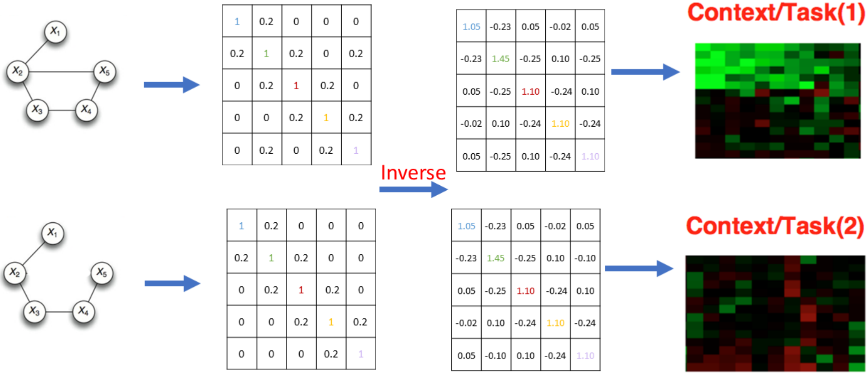
Results: Theoretical Analysis

- $p' = \max(Kp, n_{tot})$
- Error Bound: $||\widehat{\Omega}_{tot} - \Omega^*_{tot}||_F \leq 32\frac{4\kappa_1 a}{\kappa_2}\sqrt{\frac{s\log p'}{n_{tot}}}$

Multi-task:	K Single-task:
$O(\frac{\log(Kp)}{n_{tot}})$	$O(\frac{\log p}{n_i}))$

- By assuming $n_i = \frac{n_{tot}}{K}$:
- We can conclude that $\frac{\log(Kp)}{n_{tot}} < K\frac{\log p}{n_{tot}}$
- This indicates that the multi-task estimator is better!!!

Results: Synthetic Data generation process



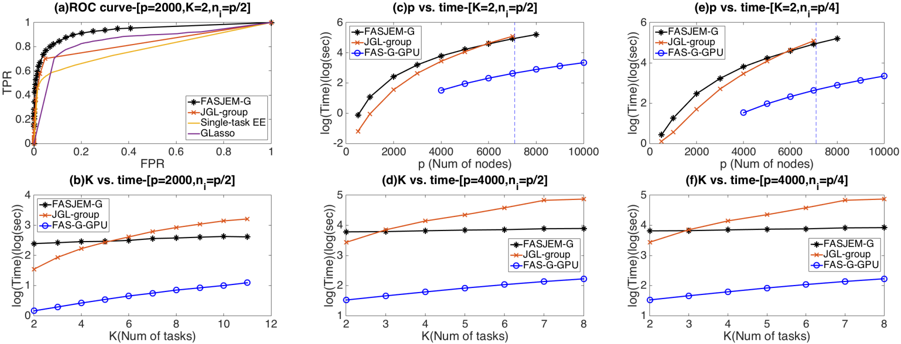
1	0.2	0	0	0
0.2	1	0.2	0	0.2
0	0.2	1	0.2	0
0	0	0.2	1	0.2
0	0.2	0	0.2	1

1.05	-0.23	0.05	-0.02	0.05
-0.23	1.45	-0.25	0.10	-0.25
0.05	-0.25	1.10	-0.24	0.10
-0.02	0.10	-0.24	1.10	-0.24
0.05	-0.25	0.10	-0.24	1.10

1	0.2	0	0	0
0.2	1	0.2	0	0
0	0.2	1	0.2	0
0	0	0.2	1	0.2
0	0	0	0.2	1

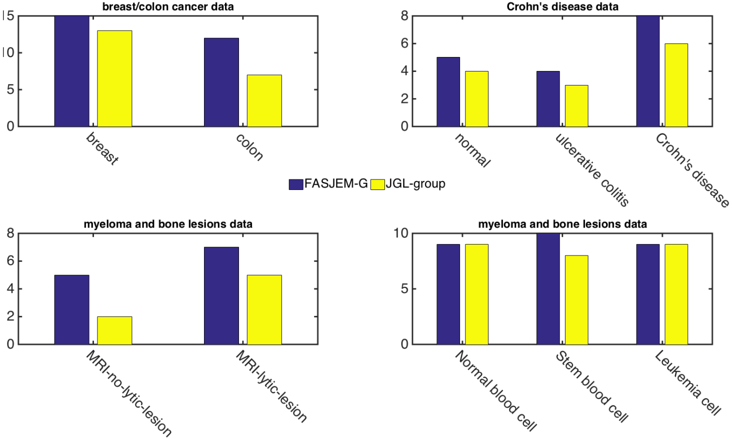
1.05	-0.23	0.05	-0.02	0.05
-0.23	1.45	-0.25	0.10	-0.10
0.05	-0.25	1.10	-0.24	0.10
-0.02	0.10	-0.24	1.10	-0.24
0.05	-0.10	0.10	-0.24	1.10

Results: Synthetic Data Results



Results: Real-world Data Results – Number of Matched Edges versus the Existing Domain Databases

- Validation by counting the overlapped interactions according to the existing bio-databases (MInact)



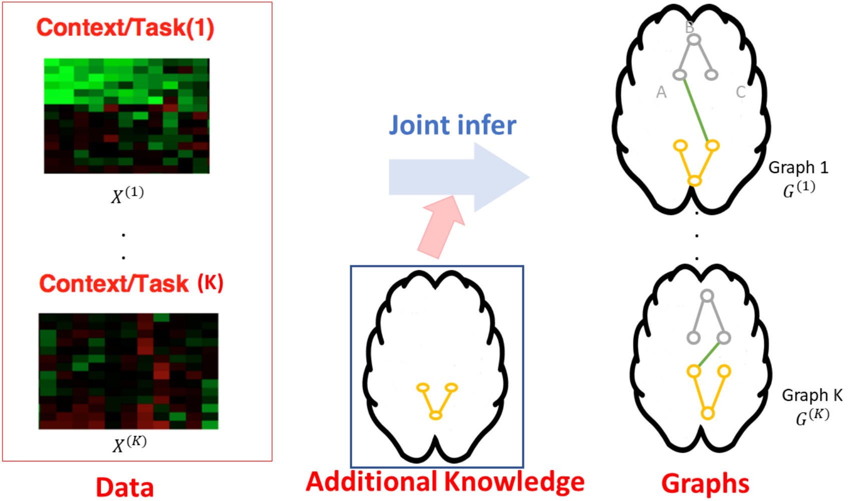
Method II: JEEK

Outline

1. Background
2. Motivation
3. Solution for Limitations - Elementary Estimator
4. Method I: FASJEM
    - Background
    - Method
    - Results
5. Method II: JEEK
    - Background
    - Method
    - Results
6. Method III: DIFFEE
    - Method
    - Results
7. Discussion
    - Questions from Proposal
    - Future works

Task II: Integrating additional knowledge

- Integrating known knowledge in Learning multiple related graphs
  - E.g., known knowledge in Brain Connection



Context/Task(1)

$X^{(1)}$

Context/Task (K)

$X^{(K)}$

Data

Joint infer

Additional Knowledge

Graph 1 $G^{(1)}$

Graph K $G^{(K)}$

Graphs

Solution: Using Knowledge as Weight in Regularization (KW-norm)

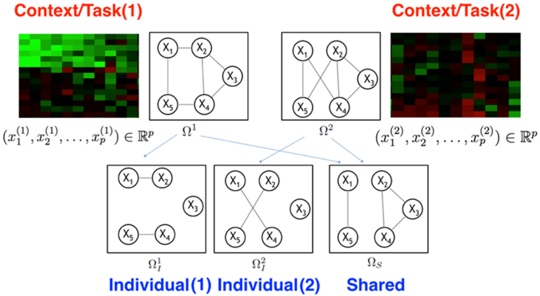
- Integrating additional knowledge through a novel regularization function $\mathcal{R}(\cdot)$

KW-norm

$$\mathcal{R}(\{\Omega^{(i)}\}) = \sum_{i=1}^{K} ||W_I^{(i)} \circ \Omega_I^{(i)}||_1 + \sum_{i=1}^{K} ||W_S \circ \Omega_S||_1 \qquad (5.1)$$

- $\Omega^{(i)} = \Omega_I^{(i)} + \Omega_S$
- $\{W_I^{(i)}\}$: weights describing knowledge of each individual graph.
- W_S: weights describing knowledge of the shared graph.

Background: Shared and Task-Specific Subgraph Representation



- Know both
  - House keeping interactions
  - Context-specific networks

Solution: Using Knowledge as Weight in Regularization (KW-norm)

- Use *tot* notation

KW-norm

$$\mathcal{R}(\Omega^{tot}) = ||W_I^{tot} \circ \Omega_I^{tot}||_1 + ||W_S^{tot} \circ \Omega_S^{tot}||_1 \tag{5.2}$$

- W_I^{tot}: weights describing knowledge of each individual graph.
- W_S^{tot}: weights describing knowledge of the shared graph.

Solution: Using Knowledge as Weight in Regularization (KW-norm)

- Use *tot* notation

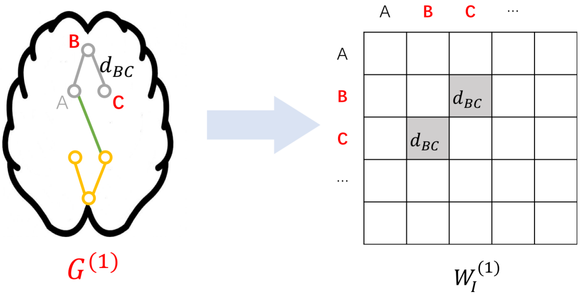
KW-norm

$$\mathcal{R}(\Omega^{tot}) = ||W_I^{tot} \circ \Omega_I^{tot}||_1 + ||W_S^{tot} \circ \Omega_S^{tot}||_1 \tag{5.2}$$

- W_I^{tot}: weights describing knowledge of each individual graph.
- W_S^{tot}: weights describing knowledge of the shared graph.
- No need to design knowledge-specific optimization
- KW-norm is flexible.

Example I: KW-norm representing the edge-level knowledge

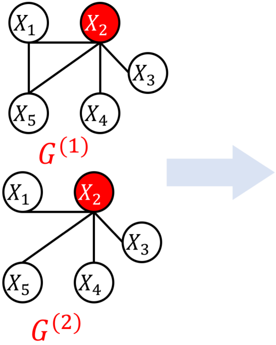
- e.g., Spatial distance among brain regions;



$$\begin{array}{c|ccccc}
 & A & B & C & \cdots & \\
\hline
A & & & & & \\
B & & & d_{BC} & & \\
C & & d_{BC} & & & \\
\cdots & & & & & \\
 & & & & &
\end{array}$$

Example II: KW-norm describing the node-level knowledge

- e.g., X_2 is a known hub node;

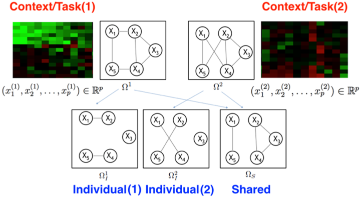


$$
\begin{array}{c|c|c|c|c|c|}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & & 1/\gamma & 1 & 1 & 1 \\
\hline
2 & 1/\gamma & & 1/\gamma & 1/\gamma & 1/\gamma \\
\hline
3 & 1 & 1/\gamma & & 1 & 1 \\
\hline
4 & 1 & 1/\gamma & 1 & & 1 \\
\hline
5 & 1 & 1/\gamma & 1 & 1 & \\
\hline
\end{array}
$$

W_s

Background: SIMULE

- Decompose $\Omega^{(i)} = \Omega_I^{(i)} + \Omega_S$
- An ℓ_1 minimization approach



$$\widehat{\Omega}_I^{(1)}, \widehat{\Omega}_I^{(2)}, \dots, \widehat{\Omega}_I^{(K)}, \widehat{\Omega}_S =$$

$$\underset{\Omega_I^{(i)}, \Omega_S}{\operatorname{argmin}} \sum_i ||\Omega_I^{(i)}||_1 + \epsilon K ||\Omega_S||_1$$

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

Background: WSIMULE: A weighted SIMULE estimator

SIMULE

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

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

- ADD $W_I^{(i)}, W_S$ $\Downarrow$

W-SIMULE

$$\widehat{\Omega}_I^{(1)}, ..., \widehat{\Omega}_I^{(K)}, \widehat{\Omega}_S = \sum_i \underset{\Omega_I^{(i)}, \Omega_S}{\operatorname{argmin}} ||W_I^{(i)} \circ \Omega_I^{(i)}||_1 + K||W_S \circ \Omega_S||_1 \tag{5.3}$$

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

Outline

1. Background
2. Motivation
3. Solution for Limitations - Elementary Estimator
4. Method I: FASJEM
    - Background
    - Method
    - Results
5. Method II: JEEK
    - Background
    - Method
    - Results
6. Method III: DIFFEE
    - Method
    - Results
7. Discussion
    - Questions from Proposal
    - Future works

Proposed Method: Combine EE and KW-norm

Elementary Estimator

$$\underset{\theta}{\operatorname{argmin}} \mathcal{R}(\theta)$$
$$\text{Subject to: } \mathcal{R}^*(\theta - \mathcal{B}^*(\widehat{\phi})) \leq \lambda_n \tag{5.4}$$

$$+$$

KW-norm

$$\mathcal{R}(\Omega^{tot}) = ||W_I^{tot} \circ \Omega_I^{tot}||_1 + ||W_S^{tot} \circ \Omega_S^{tot}||_1 \tag{5.5}$$

Proposed Method: Joint Elementary Estimator incorporating additional Knowledge (JEEK)

EE	$\mathcal{R}(\cdot)$	θ	$\widehat{\theta}_n$	$\mathcal{R}^*(\cdot)$
EE-sGGM	$\|\|\cdot\|\|_1$	Ω	$[T_v(\widehat{\Sigma})]^{-1}$	$\|\|\cdot\|\|_\infty$
JEEK	kw-norm	Ω^{tot}	$inv[T_v(\widehat{\Sigma}^{tot})]$	kw-dual

JEEK

$$\begin{aligned}
&\underset{\Omega_I^{tot},\Omega_S^{tot}}{\operatorname{argmin}} ||W_I^{tot} \circ \Omega_I^{tot}||_1 + ||W_S^{tot} \circ \Omega_S^{tot}|| \\
\text{Subject to: } & ||W_I^{tot} \circ (\Omega^{tot} - inv(T_v(\widehat{\Sigma}^{tot})))||_\infty \leq \lambda_n \\
& ||W_S^{tot} \circ (\Omega^{tot} - inv(T_v(\widehat{\Sigma}^{tot})))||_\infty \leq \lambda_n \\
& \Omega^{tot} = \Omega_S^{tot} + \Omega_I^{tot}
\end{aligned} \tag{5.6}$$

Proposed method: JEEK – Solution

- Fast and Scalable solution[2] – p^2 small linear programming subproblems with only $K+1$ variables:

$$\begin{aligned} &\underset{a_i,b}{\operatorname{argmin}} \sum_i |w_i a_i| + K|w_s b| \\ &\text{Subject to: } |a_i + b - c_i| \leq \frac{\lambda_n}{\min(w_i, w_s)}, \\ &i = 1, \dots, K \end{aligned} \tag{5.7}$$

[2] $a_i := \Omega_I^{(i)}{}_{j,k}$ (the $\{j,k\}$-th entry of $\Omega^{(i)}$)
$b := \Omega_{S_{j,k}}$
$c_i = [T_v(\widehat{\Sigma}^{(i)})]^{-1}_{j,k}$.
$W^{(i)}_{j,k} = w_i$ and $W^S_{j,k} = w_s$.

Why JEEK is better

- Rich and flexible for integrating additional knowledge
  - e.g., spatial, anatomy, hub, pathway, location, known edges;

Why JEEK is better

- Rich and flexible for integrating additional knowledge
  - e.g., spatial, anatomy, hub, pathway, location, known edges;
- Parallelizable optimization with small sub-problems.

Why JEEK is better

- Rich and flexible for integrating additional knowledge
  - e.g., spatial, anatomy, hub, pathway, location, known edges;
- Parallelizable optimization with small sub-problems.
- Theoretical guaranteed

JEEK: Computational Complexity

The best baseline of	Task I	Task II	Task III
Computational complexity	$O(Kp^3)$ / iter	$O(K^4p^5)$	$O(p^3)$ / iter
Bottle neck	SVD	Linear programming	SVD
Our approach	FASJEM	JEEK	
Computational complexity	$O(Kp^2)$ / iter	$O(K^4p^2)$	
Parallelization	$O(K)$ / iter	$O(K^4)$	

Summary

	EE	$\mathcal{R}(\cdot)$	θ	$\widehat{\theta}_n$	$\mathcal{R}^*(\cdot)$
	EE-sGGM	$\|\|\cdot\|\|_1$	Ω	$[T_v(\widehat{\Sigma})]^{-1}$	$\|\|\cdot\|\|_\infty$
Task I	FASJEM	$\|\|\cdot\|\|_1 + \mathcal{R}'$	Ω^{tot}	$inv[T_v(\widehat{\Sigma}^{tot})]$	$\max(\|\|\cdot\|\|_\infty, \mathcal{R}'^*)$
Task II	JEEK	kw-norm	Ω^{tot}	$inv[T_v(\widehat{\Sigma}^{tot})]$	kw-dual
Task III					

Outline

1. Background
2. Motivation
3. Solution for Limitations - Elementary Estimator
4. Method I: FASJEM
   - Background
   - Method
   - Results
5. **Method II: JEEK**
   - Background
   - Method
   - **Results**
6. Method III: DIFFEE
   - Method
   - Results
7. Discussion
   - Questions from Proposal
   - Future works

Theoretical Results

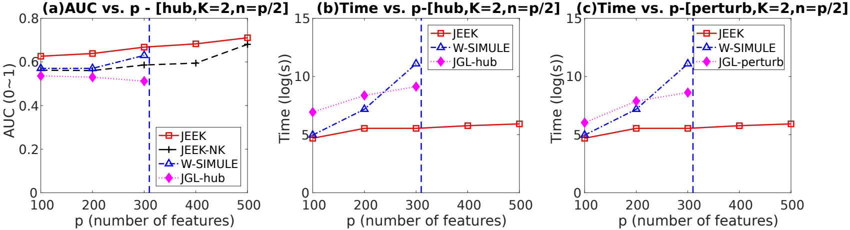
- Sharp convergence rate as the state-of-art

$$
\begin{aligned}
&||\widehat{\Omega}^{tot} - \Omega^{tot*}||_F \leq 4\sqrt{k_i + k_s}\lambda_n \\
&\max(||W_I^{tot} \circ (\widehat{\Omega}^{tot} - \Omega^{tot*})||_\infty, ||W_S^{tot} \circ (\widehat{\Omega}^{tot} - \Omega^{tot*}||_\infty) \leq 2\lambda_n \\
&||W_I^{tot} \circ (\widehat{\Omega}_I^{tot} - \Omega_I^{tot*})||_1 + ||W_S^{tot} \circ (\widehat{\Omega}_S^{tot} - \Omega_S^{tot*})||_1 \leq 8(k_i + k_s)\lambda_n
\end{aligned}
\tag{5.8}
$$

Where a, c, κ_1 and κ_2 are constants

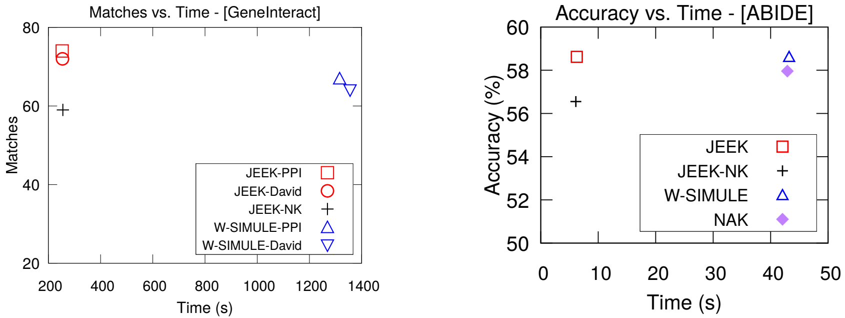
$$
\begin{aligned}
&||\widehat{\Omega}^{tot} - \Omega^{tot*}||_F \\
&\qquad \leq \frac{16\kappa_1 a \max_{j,k}({W_I^{tot}}_{j,k}, {W_S^{tot}}_{j,k})}{\kappa_2} \sqrt{\frac{(k_i + k_s)\log(Kp)}{n_{tot}}}
\end{aligned}
\tag{5.9}
$$

Empirical Results on Multiple Synthetic Datasets



- JEEK outperforms the speed of the state-of arts significantly faster ($\sim$ 5000× improvement);
- JEEK obtains better AUC as the state-of-the-art;
- JEEK obtains better AUC than JEEK-NK (no additional knowledge).

Empirical Results on Two Real-world Datasets



(a)

(b)

- (a). On real-world gene expression data about leukemia cells vs. normal blood cells. Used multiple types of additional knowledge;
- (b). On real-world Brain fMRI dataset: ABIDE. Using LDA as a downstream classification for evaluating JEEK vs. baselines.

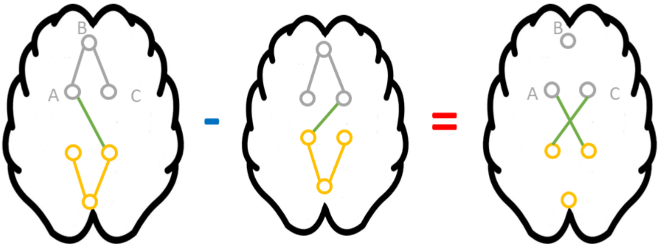
Method III: DIFFEE

Outline

1. Background
2. Motivation
3. Solution for Limitations - Elementary Estimator
4. Method I: FASJEM
   - Background
   - Method
   - Results
5. Method II: JEEK
   - Background
   - Method
   - Results
6. Method III: DIFFEE
   - Method
   - Results
7. Discussion
   - Questions from Proposal
   - Future works

Takes III: Learning sparse changes between two graphs

- Each graph may be dense or sparse, differential net is sparse



Proposed Method III: DIFFEE

- Two cases : d (disease) & c (control)

$$\underset{\theta}{\text{argmin}} ||\theta||_1$$

Subject to:

$$||\theta - \mathcal{B}^*(\widehat{\phi})||_\infty \leq \lambda_n \quad (6.1)$$

$$\underset{\Longrightarrow}{\Delta = \Omega_d - \Omega_c}$$

$$\underset{\Delta}{\text{argmin}} ||\Delta||_1$$

Subject to:

$$||\Delta - \mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)||_\infty \leq \lambda_n \quad (6.2)$$

Proposed Method III: DIFFEE

Elementary Estimator (EE)

$$\underset{\theta}{\operatorname{argmin}} \mathcal{R}(\theta)$$

$$\text{Subject to: } \mathcal{R}^*(\theta - \mathcal{B}^*(\widehat{\phi})) \leq \lambda_n \tag{6.3}$$

EE	$\mathcal{R}(\cdot)$	θ	$\widehat{\theta}_n$	$\mathcal{R}^*(\cdot)$
EE-sGGM	$\|\|\cdot\|\|_1$	Ω	$[T_v(\widehat{\Sigma})]^{-1}$	$\|\|\cdot\|\|_\infty$
DIFFEE	$\|\|\cdot\|\|_1$	Δ	$\Big([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}\Big)$	$\|\|\cdot\|\|_\infty$

DIFFEE

$$\underset{\Delta}{\operatorname{argmin}} \, ||\Delta||_1$$

$$\text{Subject to: } ||\Delta - \Big([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}\Big)||_\infty \leq \lambda_n \tag{6.4}$$

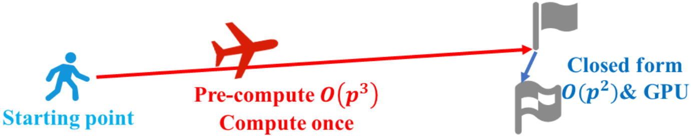
DIFFEE: Optimization Solution

- Close form

$$\widehat{\Delta} = S_{\lambda_n}([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}) \tag{6.5}$$

$$[S_\lambda(A)]_{ij} = \text{sign}(A_{ij}) \max(|A_{ij}| - \lambda, 0) \tag{6.6}$$

- GPU-parallelizable



DIFFEE: Computational Complexity

The best baseline of	Task I	Task II	Task III
Computational complexity	$O(Kp^3)$ / iter	$O(K^4p^5)$	$O(p^3)$ / iter
Bottle neck	SVD	Linear programming	SVD
Our approach	FASJEM	JEEK	DIFFEE
Computational complexity	$O(Kp^2)$ / iter	$O(K^4p^2)$	$O(p^3)$
Parallelization	$O(K)$ / iter	$O(K^4)$	$O(p^3)$

Summary

	EE	$\mathcal{R}(\cdot)$	θ	$\widehat{\theta}_n$	$\mathcal{R}^*(\cdot)$
	EE-sGGM	$\|\|\cdot\|\|_1$	Ω	$[T_v(\widehat{\Sigma})]^{-1}$	$\|\|\cdot\|\|_\infty$
Task I	FASJEM	$\|\|\cdot\|\|_1 + \mathcal{R}'$	Ω^{tot}	$inv[T_v(\widehat{\Sigma}^{tot})]$	$\max(\|\|\cdot\|\|_\infty, \mathcal{R}'^*)$
Task II	JEEK	kw-norm	Ω^{tot}	$inv[T_v(\widehat{\Sigma}^{tot})]$	kw-dual
Task III	DIFFEE	$\|\|\cdot\|\|_1$	Δ	$[T_v(\widehat{\Sigma}_d)]^{-1}$ $-[T_v(\widehat{\Sigma}_c)]^{-1}$	$\|\|\cdot\|\|_\infty$

Outline

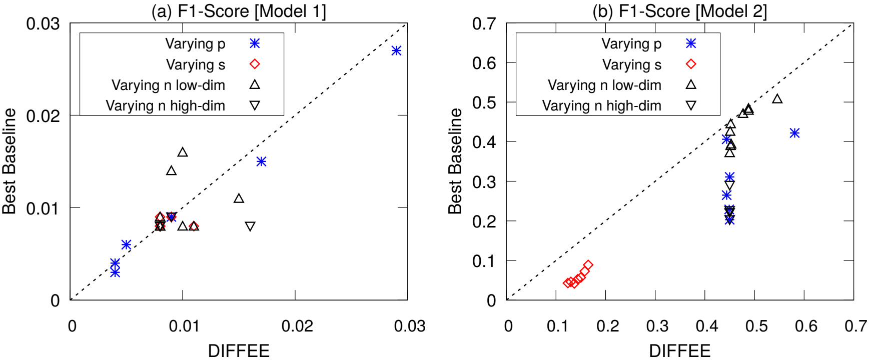
1. Background
2. Motivation
3. Solution for Limitations - Elementary Estimator
4. Method I: FASJEM
   - Background
   - Method
   - Results
5. Method II: JEEK
   - Background
   - Method
   - Results
6. Method III: DIFFEE
   - Method
   - Results
7. Discussion
   - Questions from Proposal
   - Future works

Results: Theoretical Analysis

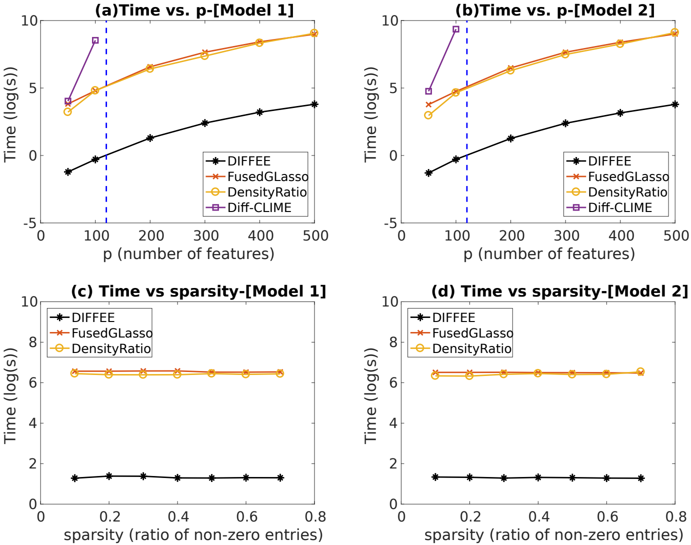
- Sharp convergence rate as the state-of-art

$$
\begin{aligned}
||\widehat{\Delta} - \Delta^*||_\infty &\leq \frac{16\kappa_1 a}{\kappa_2}\sqrt{\frac{\log p}{\min(n_c, n_d)}} \\
||\widehat{\Delta} - \Delta^*||_F &\leq \frac{32\kappa_1 a}{\kappa_2}\sqrt{\frac{k \log p}{\min(n_c, n_d)}} \\
||\widehat{\Delta} - \Delta^*||_1 &\leq \frac{64\kappa_1 a}{\kappa_2} k\sqrt{\frac{\log p}{\min(n_c, n_d)}}
\end{aligned}
\tag{6.7}
$$

Results: Synthetic Data Results



Results: Synthetic Data Results



(a)Time vs. p-[Model 1]

(b)Time vs. p-[Model 2]

(c) Time vs sparsity-[Model 1]

(d) Time vs sparsity-[Model 2]

Results: Real-world Data Results

- Apply to Brain image data (fMRI)
- Use the estimated different network in LDA
- Compare the accuracy with the state-of-art methods

Method	DIFFEE	FusedGLasso	Diff-CLIME
Accuracy (%)	**57.58%**	56.90%	53.79%

Discussion

Outline

1. Background
2. Motivation
3. Solution for Limitations - Elementary Estimator
4. Method I: FASJEM
   - Background
   - Method
   - Results
5. Method II: JEEK
   - Background
   - Method
   - Results
6. Method III: DIFFEE
   - Method
   - Results
7. Discussion
   - Questions from Proposal
   - Future works

Support Analysis Results

- DIFFEE as an example

Lemma

$$||\Delta^* - \mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)||_\infty \leq \lambda_n \tag{7.1}$$

Support Analysis Results

- DIFFEE as an example

Lemma

$$||\Delta^* - \mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)||_\infty \leq \lambda_n \tag{7.1}$$

- $$\Downarrow$$

Corollary

$$\Delta^*_{i,j} = 0 \Longrightarrow |\mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)_{i,j}| \leq \lambda_n \tag{7.2}$$

- $$\widehat{\Delta} = S_{\lambda_n}(\mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)) \tag{7.3}$$

Support Analysis Results

- DIFFEE as an example

Lemma

$$||\Delta^* - \mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)||_\infty \leq \lambda_n \tag{7.1}$$

- $$\Downarrow$$

Corollary

$$\Delta^*_{i,j} = 0 \Longrightarrow |\mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)_{i,j}| \leq \lambda_n \tag{7.2}$$

- $$\widehat{\Delta} = S_{\lambda_n}(\mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)) \tag{7.3}$$

Result

$$\Delta^*_{i,j} = 0 \Longrightarrow \widehat{\Delta}_{i,j} = 0 \tag{7.4}$$

- $supp(\widehat{\Delta}) \subseteq supp(\Delta^*)$

Support Analysis Result

- Additional Assumption:

Assumption

$$\min_{s \in supp(\Delta^*)} |\Delta^*_s| \geq 3||\Delta^* - \mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)||_\infty \tag{7.5}$$

Support Analysis Result

- Additional Assumption:

Assumption

$$\min_{s \in supp(\Delta^*)} |\Delta^*_s| \geq 3||\Delta^* - \mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)||_\infty \tag{7.5}$$

- $$supp(\Delta^*) \subseteq supp(\widehat{\Delta}) \tag{7.6}$$

Support Analysis Result

- Additional Assumption:

Assumption

$$\min_{s \in supp(\Delta^*)} |\Delta^*_s| \geq 3||\Delta^* - \mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)||_\infty \tag{7.5}$$

- $$supp(\Delta^*) \subseteq supp(\widehat{\Delta}) \tag{7.6}$$
- Combine the above results

$$supp(\Delta^*) = supp(\widehat{\Delta}) \tag{7.7}$$

Standardized Covariance Matrices

- Real world: Different tasks $\rightarrow$ different value scale
  - e.g., fMRI vs RNA squencing
- Problem: hard to choose λ_n in different scales

Standardized Covariance Matrices

- Real world: Different tasks $\rightarrow$ different value scale
    - e.g., fMRI vs RNA squencing
- Problem: hard to choose λ_n in different scales
- Solution: ~~Covariance matrices~~ $\Longrightarrow$ Correlation matrices

Theorem

The inverse of Correlation matrices have the same support set as the inverse of covariance matrices

- Nonparanormal extensions – Relax the Gaussian Assumption
- Added in all the packages

Iteration number T

- linearly converge method: $T = O(n \log(\frac{1}{TOL}))$
- TOL is the error bound

Iteration number T

- linearly converge method: $T = O(n \log(\frac{1}{TOL}))$
- TOL is the error bound

- FASJEM error bound: $O(\frac{\log(Kp)}{n_{tot}})$

Iteration number T

- linearly converge method: $T = O(n\log(\frac{1}{TOL}))$
- TOL is the error bound

- FASJEM error bound: $O(\frac{\log(Kp)}{n_{tot}})$
- $T = O(\frac{n_{tot}\log(n_{tot})}{\log(\log(Kp))})$

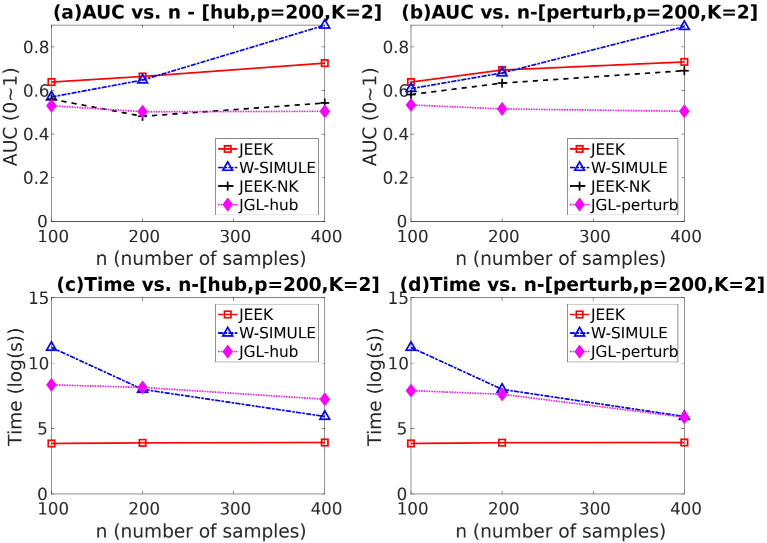
Trade-off

- proxy backward mapping still $O(p^3)$

- In practice, fast in our three tasks
- Thanks to excellent low-level implementation

- Not well performed in low-dimensional case
- $p' = \max(n, p)$

Trade-off


(a)AUC vs. n - [hub,p=200,K=2]
(b)AUC vs. n-[perturb,p=200,K=2]
AUC (0~1)
0
0.2
0.4
0.6
0.8
100
200
300
400
n (number of samples)
JEEK
W-SIMULE
JEEK-NK
JGL-hub
JGL-perturb
(c)Time vs. n-[hub,p=200,K=2]
(d)Time vs. n-[perturb,p=200,K=2]
Time (log(s))
0
5
10
15

Outline

1. Background
2. Motivation
3. Solution for Limitations - Elementary Estimator
4. Method I: FASJEM
   - Background
   - Method
   - Results
5. Method II: JEEK
   - Background
   - Method
   - Results
6. Method III: DIFFEE
   - Method
   - Results
7. Discussion
   - Questions from Proposal
   - Future works

KW-norm for FASJEM

- Revise the ℓ_1 norm in FASJEM to a KW-norm

KW-norm for FASJEM

$$\begin{aligned} \mathcal{R}(\{\Omega^{(i)}\}) &= \sum_{i=1}^{K} ||W^{(i)} \circ \Omega^{(i)}||_1 \\ &= ||W^{tot} \circ \Omega^{tot}||_1 \end{aligned} \quad (7.8)$$

- $\{W^{(i)}\}$: weights describing knowledge of each graph.

Future work: FASJEM with additional knowledge – FASJEM-K

FASJEM-K

$$
\begin{aligned}
&\underset{\Omega_{tot}}{\operatorname{argmin}} ||W_{tot} \circ \Omega_{tot}||_1 + \epsilon \mathcal{R}'(\Omega_{tot}) \\
&s.t. ||W_{tot} \circ (\Omega_{tot} - \text{inv}(T_v(\widehat{\Sigma}_{tot})))||_\infty \leq \lambda_n \\
&\mathcal{R}'^*(\Omega_{tot} - \text{inv}(T_v(\widehat{\Sigma}_{tot}))) \leq \epsilon \lambda_n
\end{aligned}
\tag{7.9}
$$

KW-norm for Differential Network: kEV-norm

- Integrating both edge-level and node-level additional knowledge through a novel regularization function $\mathcal{R}(\cdot)$

kEV-norm

$$\mathcal{R}(\Delta) = ||W_E \circ \Delta_{E \setminus \mathcal{G}_V}||_1 + \epsilon ||\Delta_{\mathcal{G}_V}||_{\mathcal{G}_V,2} \tag{7.10}$$

- $\mathcal{G}_V$ is a node group.
- W_E represents the weights for edges.

Future work: DIFFEE-K

- Combine kEV-norm and Elementary Estimator

DIFFEE-K

$$\underset{\Delta}{\text{argmin}} \, ||W_E \circ \Delta_{E \setminus \mathcal{G}_V}||_1 + \epsilon ||\Delta_{\mathcal{G}_V}||_{\mathcal{G}_V,2}$$

$$\text{Subject to: } ||W_E \circ \left(\Delta - \left([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}\right)\right)||_\infty \leq \lambda_n \quad (7.11)$$

$$\epsilon||\Delta - \left([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}\right)||^*_{\mathcal{G}_V,2} \leq \lambda_n$$

Publications

- FASJEM
  - A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models, B Wang, J Gao, Y Qi, AISTATS 2017
- DIFFEE
  - Fast and Scalable Learning of Sparse Changes in High-Dimensional Gaussian Graphical Model Structure, B Wang, A Sekhon, Y Qi, AISTATS 2018
- W-SIMULE
  - A constrainedℓ 1 minimization approach for estimating multiple sparse Gaussian or nonparanormal graphical models, B Wang, R Singh, Y Qi, Machine Learning 106 (9-10), 1381-1417
  - A Constrained, Weighted-L1 Minimization Approach for Joint Discovery of Heterogeneous Neural Connectivity Graphs, C Singh, B Wang, Y Qi, Advances in Modeling and Learning Interactions from Complex Data, NIPS 2017 Workshop

Publications

- JEEK
    - A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models, B Wang, A Sekhon, Y Qi, ICML 2018
- DIFFEE-K
    - A Fast and Scalable Estimator for Using Additional Knowledge in Learning Sparse Structure Change of High-Dimensional Gaussian Graphical Models, B Wang, A Sekhon, Y Qi, submit to NIPS 2018

R Package is Available !!!

- The project website: `http://jointggm.org/`
- R package ”simule”:
  - `install.packages("simule")`
  - `demo(simule)` !
- R package ”fasjem”:
  - `install.packages("fasjem")`
  - `demo(fasjem)` !
- R package ”diffee”:
  - `install.packages("diffee")`
  - `demo(diffee)` !
- R package ”jeek”:
  - `install.packages("jeek")`
  - `demo(jeek)` !
- A complete package ”jointNet” will be ready by this summer.

Acknowledgement

- Advisor: Yanjun Qi
- Co-authors: Rita, Arshdeep, Ji, Chandan
- Lab mates: Zhaoyang, Jack, Weilin
- My Family
- Thanks!

Back-up: Difficulty in combining FASJEM and JEEK

$$
\begin{aligned}
\underset{\Omega_I^{tot},\Omega_S^{tot}}{\operatorname{argmin}} & ||W_I^{tot} \circ \Omega_I^{tot}||_1 + ||W_S^{tot} \circ \Omega_S^{tot}|| + \epsilon \mathcal{R}'(\Omega^{tot}) \\
\text{Subject to: } & ||W_I^{tot} \circ (\Omega^{tot} - inv(T_v(\widehat{\Sigma}^{tot})))||_\infty \le \lambda_n \\
& ||W_S^{tot} \circ (\Omega^{tot} - inv(T_v(\widehat{\Sigma}^{tot})))||_\infty \le \lambda_n \\
& \mathcal{R}^{*\prime}(\Omega^{tot}) \le \epsilon \lambda_n
\end{aligned}
\tag{7.12}
$$

- Hard to optimize
- Lose fast and scalable property

Back-up: How to choose v in $T_v(\widehat{\Sigma})$

- line search
- v from the set $\{0.001i | i = 1, 2, \ldots, 1000\}$
- pick a value that makes $T_v(\widehat{\Sigma})$ and be invertible

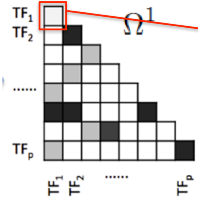
Back-up: Connecting to Bayesian Statistics

$$
\begin{aligned}
&-\log(\mathbb{P}(\Omega^{(i)}|X^{(i)},\mu^{(i)},{W_I^{(i)}}_{j,k},W_{S j,k})) \\
&\propto -\log(det({\Omega^{(i)}}^{-1}))+<\Omega^{(i)},\widehat{\Sigma}^{(i)}> \\
&+\sum_{j,k}({W_I^{(i)}}_{j,k}|{\Omega_I^{(i)}}_{j,k}|+W_S|\Omega_{S j,k}|)
\end{aligned}
\tag{7.13}
$$

Back-up: Proximal algorithm Basics

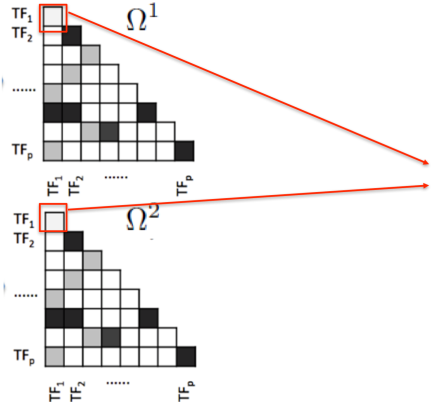
- proximity definition:
- $\text{prox}_h(x) = \underset{u}{\operatorname{argmin}}(h(u) + \frac{1}{2}||u - x||_2^2)$
- $\underset{x}{\operatorname{argmin}}\, f(x) = g(x) + h(x)$
- proximal gradient descent:
- $x^{(k)} = \text{prox}_{t_k h}(x^{(k-1)} - t_k \bigtriangledown g(x^{(k-1)}))$

Back-up: Proximal algorithm for FASJEM



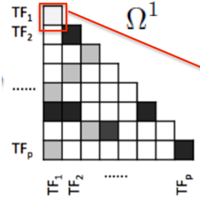
$$(\text{prox}_{\gamma||\cdot||_1}(x))_i = \begin{cases} x_i - \gamma, x_i > \gamma \\ 0, |x_i| \leq \gamma \\ x_i + \gamma, x_i < -\gamma \end{cases}$$

Back-up: Proximal algorithm for FASJEM



$$\mathrm{prox}_{\gamma||\cdot||_{\mathcal{G},2}}(x_g) = \begin{cases} x_g - \gamma \frac{x_g}{||x_g||_2}, ||x_g||_2 > \gamma \\ 0, ||x_g||_2 \leq \gamma \end{cases}$$

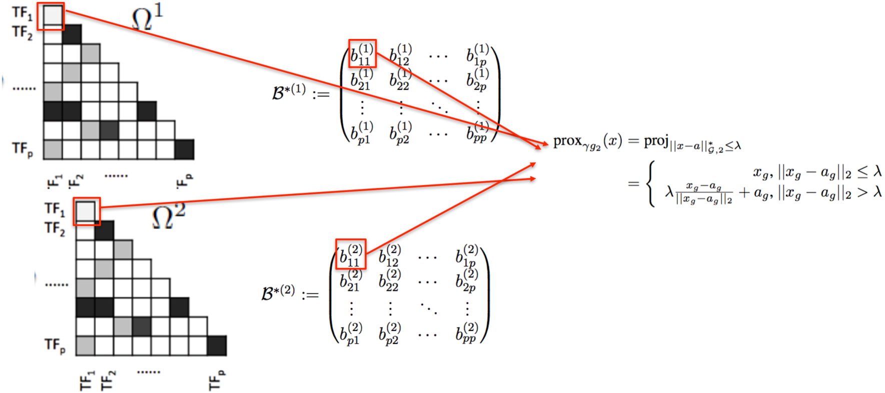
Back-up: Proximal algorithm for FASJEM



$$\mathcal{B}^{*(1)} := \begin{pmatrix} \boxed{b_{11}^{(1)}} & b_{12}^{(1)} & \cdots & b_{1p}^{(1)} \\ b_{21}^{(1)} & b_{22}^{(1)} & \cdots & b_{2p}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1}^{(1)} & b_{p2}^{(1)} & \cdots & b_{pp}^{(1)} \end{pmatrix}$$

$$\mathrm{proj}_{||x-a||_\infty \le \lambda} = \begin{cases} x_i, |x_i - a_i| \le \lambda \\ a_i + \lambda, x_i > a_i + \lambda \\ a_i - \lambda, x_i < a_i - \lambda \end{cases}$$

Back-up: Proximal algorithm for FASJEM



$$\mathcal{B}^{*(1)} := \begin{pmatrix} b_{11}^{(1)} & b_{12}^{(1)} & \cdots & b_{1p}^{(1)} \\ b_{21}^{(1)} & b_{22}^{(1)} & \cdots & b_{2p}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1}^{(1)} & b_{p2}^{(1)} & \cdots & b_{pp}^{(1)} \end{pmatrix}$$

$$\mathcal{B}^{*(2)} := \begin{pmatrix} b_{11}^{(2)} & b_{12}^{(2)} & \cdots & b_{1p}^{(2)} \\ b_{21}^{(2)} & b_{22}^{(2)} & \cdots & b_{2p}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1}^{(2)} & b_{p2}^{(2)} & \cdots & b_{pp}^{(2)} \end{pmatrix}$$

$$\text{prox}_{\gamma g_2}(x) = \text{proj}_{||x-a||^*_{\mathcal{G},2} \le \lambda}$$

$$= \begin{cases} x_g, & ||x_g - a_g||_2 \le \lambda \\ \lambda \frac{x_g - a_g}{||x_g - a_g||_2} + a_g, & ||x_g - a_g||_2 > \lambda \end{cases}$$

Joint Gaussian Graphical Model Review Series – I

Probability Foundations

Beilun Wang
Advisor: Yanjun Qi

[1]Department 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

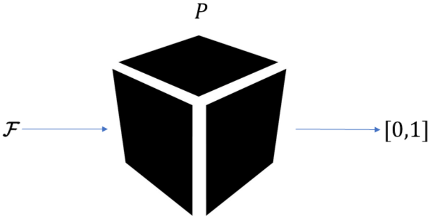
Let $(\Omega, \mathcal{F}, \mathbb{P})$ be the probability space.

- 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 : $\mathcal{F} \to [0, 1]$.

Events

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

Probability Measure



P

$\mathcal{F}$

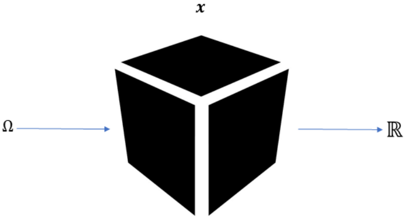
$[0,1]$

Random Variable

Random Variable

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

Random Variable



Probability Distribution

Probability Distribution function

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

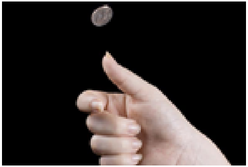
- $X = Y$, they follow same distribution?
- $F_X = F_Y$, then $X = Y$?

Joint Probability

Joint Probability

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

Joint Probability



Twice

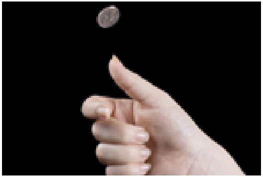
{Head, Head} {Tail, Tail} {Head, Tail}

Marginal Probability

Marginal Probability

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

Joint Probability



Twice

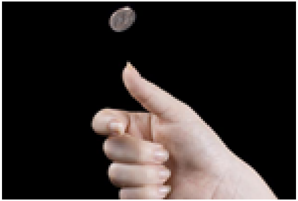
Head or Tail for the first one?

Conditional Distribution

Conditional Distribution

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

Joint Probability



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

$Y|X = Y$

- Filp coin example
- Causal relationship

Correlation

Covariance

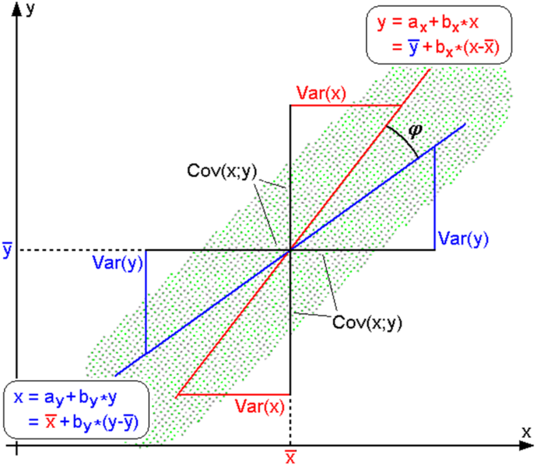
$Cov(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$, where μ_X, μ_Y is the mean vector.

Correlation

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

- 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:

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

Gaussian Example

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

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

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

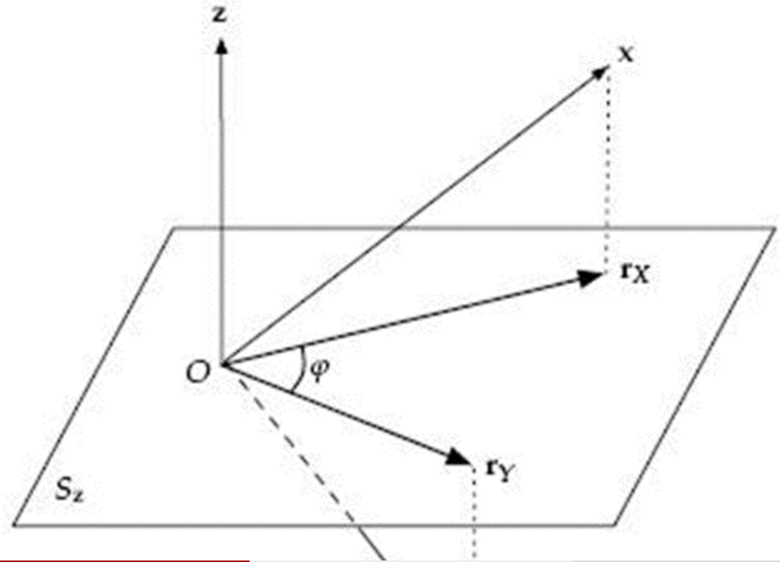
X and Y are independent on Z if and only if $p_{X,Y|Z}(x,y) = p_{X|Z}(x)p_{Y|Z}(y)$, where p is the probability density function.

Partial Correlation

Partial Correlation

Formally, the partial correlation between X and Y given random variable Z , written $\rho_{XY \cdot Z}$, 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

[1]Department 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.,$(X, Y) \sim N(0, diag(\sigma_X^2, \sigma_Y^2))$.

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

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 $\widehat{\Sigma}$. 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 $\widehat{\Sigma}$ may not be invertible.

Maximum Likelihood Method

The MLE of μ

$\mathcal{L}(\mu, \Omega) = (2\pi)^{-\frac{np}{2}} \prod_{i=1}^{n} \det(\Omega^{-1})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_i - \mu)^T \Omega (x_i - \mu)\right)$.
After take a first derivative, it is easy to show that $\bar{x} = \frac{x_1 + \cdots + x_n}{n}$

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

The Likelihood of Ω

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

$$\begin{aligned}
\mathcal{L}(\bar{x}, \Omega) &\propto \det(\Omega^{-1})^{-\frac{n}{2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{n} \mathrm{tr}\left(\left(x_i - \bar{x}\right)^T \Omega \left(x_i - \bar{x}\right)\right)\right) && (4.1)\\
&= \det(\Omega^{-1})^{-\frac{n}{2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{n} \mathrm{tr}\left(\left(x_i - \bar{x}\right)\left(x_i - \bar{x}\right)^T \Omega\right)\right) && (4.2)\\
&= \det(\Omega^{-1})^{-\frac{n}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\sum_{i=1}^{n} \left(x_i - \bar{x}\right)\left(x_i - \bar{x}\right)^T \Omega\right)\right) && (4.3)\\
&= \det(\Omega^{-1})^{-\frac{n}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(S\Omega\right)\right) && (4.4)
\end{aligned}$$

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

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

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

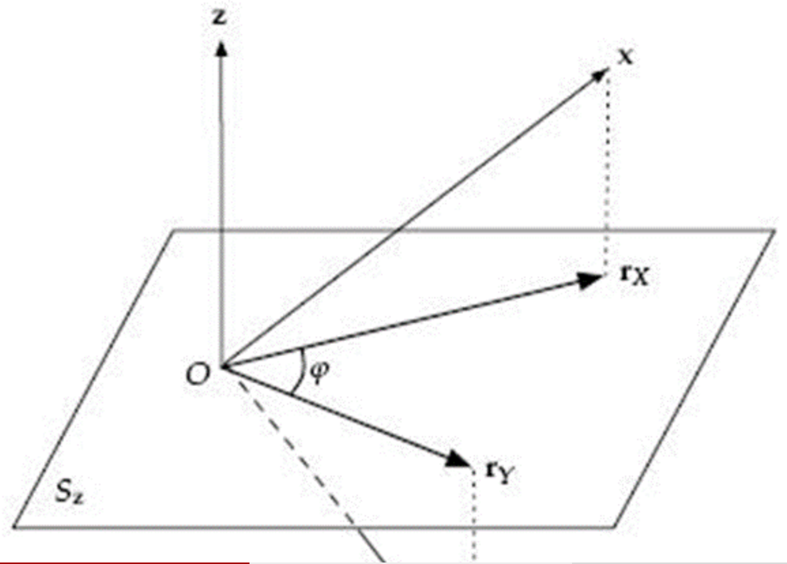
$$\ln \mathcal{L}(\bar{x}, \Omega) \propto \ln \det(\Omega) - \text{tr} \left(\Omega \frac{1}{n} \sum_{i=1}^{n} (\bar{x} - \mu)(\bar{x} - \mu)^T \right) \tag{4.5}$$

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

where $\widehat{S}$ 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 $X_1 = X_j$.
$X_j | X_{\setminus j} \ N(\Sigma_{\setminus j,j}\Sigma^{-1}_{\setminus j,\setminus j}X_{\setminus j}, \Sigma_{jj} - \Sigma_{\setminus j,j}\Sigma^{-1}_{\setminus j,\setminus j}\Sigma_{\setminus j,j})$
Let $\alpha_j := \Sigma_{\setminus j,j}\Sigma^{-1}_{\setminus j,\setminus j}$ and $\sigma_j^2 := \Sigma_{jj} - \Sigma_{\setminus j,j}\Sigma^{-1}_{\setminus j,\setminus j}\Sigma_{\setminus j,j}$. We have that

$$X_j = \alpha_j^T X_{\setminus j} + \epsilon_j \tag{5.1}$$

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

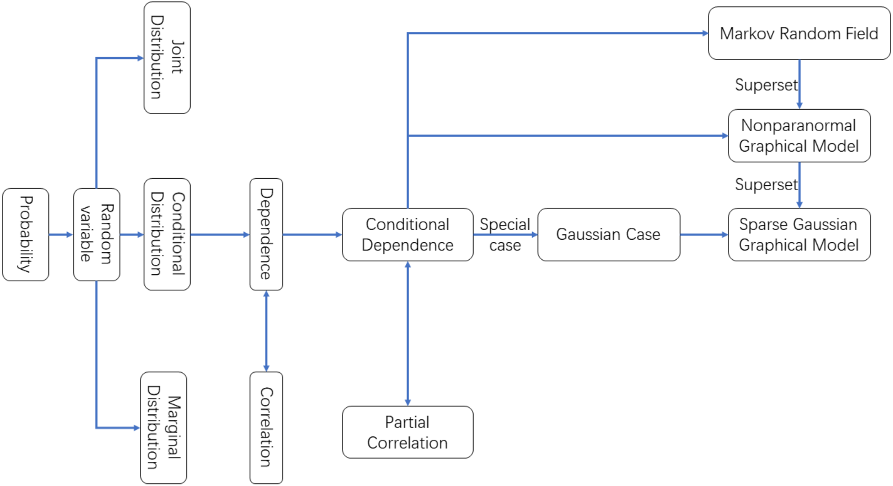
[1]Department of Computer Science, University of Virginia
http://jointggm.org/

July 7th, 2017

Outline

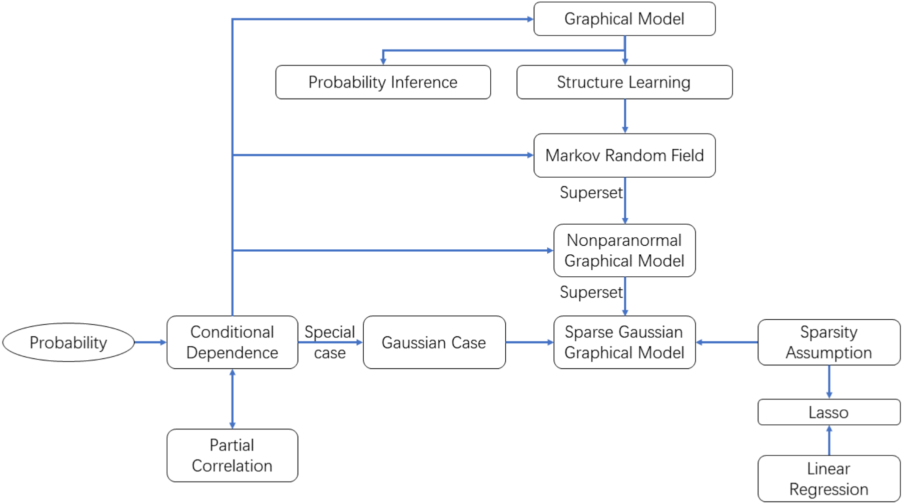
1. Why we need Graphical Model?
2. Graphical Model
3. Markov Random Field

Road Map



Probability
Random variable
Joint Distribution
Conditional Distribution
Marginal Distribution
Dependence
Correlation
Conditional Dependence
Partial Correlation
Special case
Gaussian Case
Markov Random Field
Superset
Nonparanormal Graphical Model
Superset
Sparse Gaussian Graphical Model

Road Map

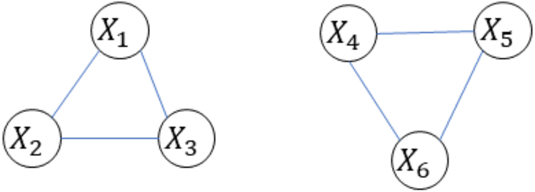

Graphical Model
Probability Inference
Structure Learning
Markov Random Field
Superset
Nonparanormal Graphical Model
Superset
Probability
Conditional Dependence
Special case
Gaussian Case
Sparse Gaussian Graphical Model
Sparsity Assumption
Lasso
Linear Regression
Partial Correlation

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

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

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

$$p(X_1|X_2, X_3)p(X_2|X_3)p(X_3)p(X_4, X_5, X_6) \tag{1.1}$$

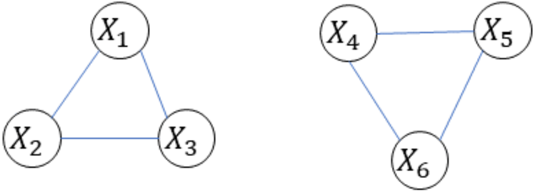
$$=p(X_1, X_2, X_3)p(X_4, X_5, X_6) \tag{1.2}$$

Graphical Model

Graphical Model

- **Probability Inference:** estimate joint probability, marginal probability, and conditional probability.
- **Structure learning:** Give dataset **X**, learn the Graph structure from **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

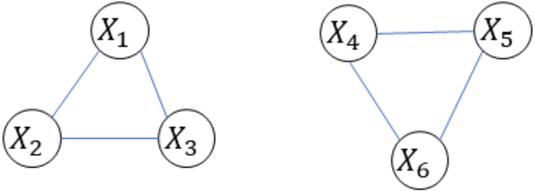
Markov Random Field

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: $X_v \perp\!\!\!\perp X_{V \setminus N(v)} | X_{N(v)}$

This property is stronger than the pairwise Markov property:
Any two non-adjacent variables are conditionally independent given all other variables: $X_u \perp\!\!\!\perp X_v \mid X_{V \setminus \{u,v\}} \quad \text{if } \{u, v\} \notin E.$

A Toy Example



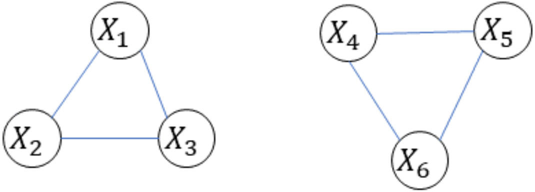
Clique factorization

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

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

then X forms a Markov random field with respect to G. Here, $\mathrm{cl}(G)$ 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

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

.

Solution:

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

$$= \ln \det(\Omega) - \mathrm{tr}\left(\Omega \widehat{S}\right) \tag{3.2}$$

where $\widehat{S}$ 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

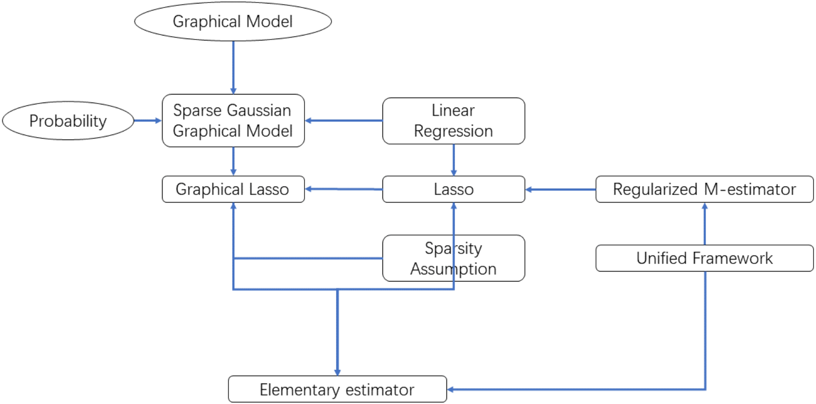
A Unified Framework for M-estimaotr and Elementary Estimators

Beilun Wang
Advisor: Yanjun Qi

[1]Department of Computer Science, University of Virginia
http://jointggm.org/

July 21st, 2017

Road Map


Graphical Model
Probability
Sparse Gaussian Graphical Model
Linear Regression
Graphical Lasso
Lasso
Regularized M-estimator
Sparsity Assumption
Unified Framework
Elementary estimator

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:

Constrains: 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:

$$\underset{\theta}{\mathrm{argmin}}\, \mathcal{L}(X, \theta) + \lambda_n \mathcal{R}(\theta) \tag{3.1}$$

Example: Linear Model

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

Target

Find β, such that $X\beta = y$.

Constrains: 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.

$$\underset{\beta}{\mathrm{argmin}} ||y - X\beta||_2 \tag{3.2}$$

$$\text{Subject to: } ||\beta||_0 \leq t \tag{3.3}$$

Example: Lasso

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

$$\underset{\beta}{\operatorname{argmin}} ||y - X\beta||_2 \tag{3.4}$$

$$\text{Subject to: } ||\beta||_1 \leq t \tag{3.5}$$

Lasso

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

Other equivalent formulation

$$\underset{\beta}{\operatorname{argmin}} ||\beta||_1 \tag{3.6}$$

$$\text{Subject to: } y = X\beta \tag{3.7}$$

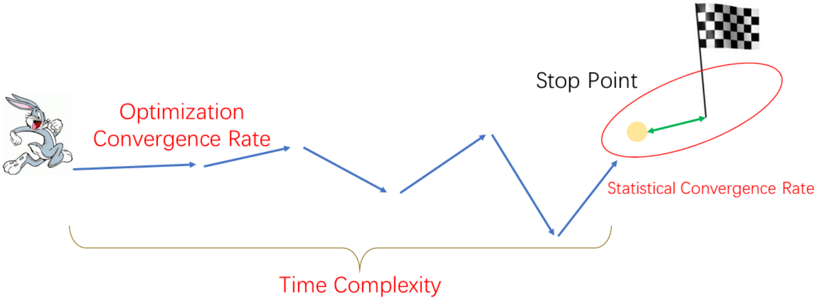
Dantzig selector

$$\underset{\beta}{\operatorname{argmin}} ||\beta||_1 \tag{3.8}$$

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

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.

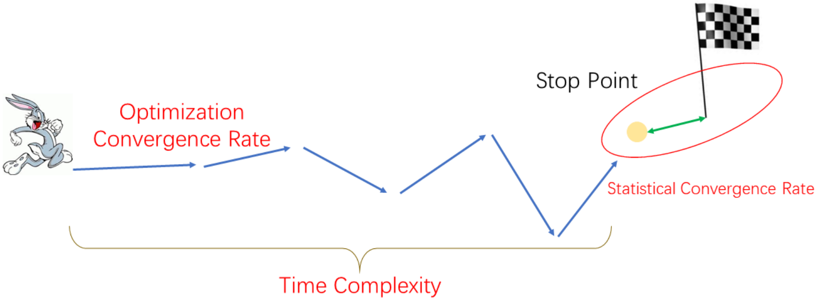
Traditional statisticians focus on the statistical convergence rate (Accuracy).

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.,$p/n \to c \neq 0$): 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
$\bar{\mathcal{M}}^{\perp} := \{v \in \mathbb{R}^p | < u, v >= 0 \forall u \in \bar{\mathcal{M}}\}$.

Subspace compatibility constant

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

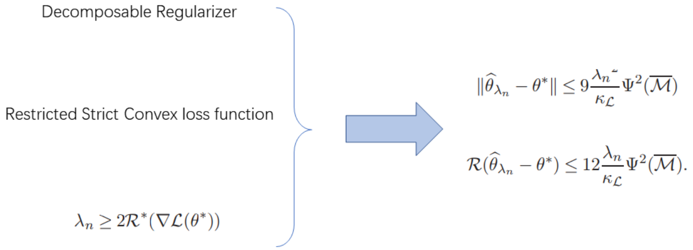
with respect to the pair $(\mathcal{R}, || \cdot ||)$.

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



Decomposable Regularizer

Restricted Strict Convex loss function

$$\lambda_n \geq 2\mathcal{R}^*(\nabla\mathcal{L}(\theta^*))$$

$$\|\widehat{\theta}_{\lambda_n} - \theta^*\| \leq 9\frac{\lambda_n{}^{2}}{\kappa_{\mathcal{L}}}\Psi^2(\overline{\mathcal{M}})$$

$$\mathcal{R}(\widehat{\theta}_{\lambda_n} - \theta^*) \leq 12\frac{\lambda_n}{\kappa_{\mathcal{L}}}\Psi^2(\overline{\mathcal{M}}).$$

Example: Lasso

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

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

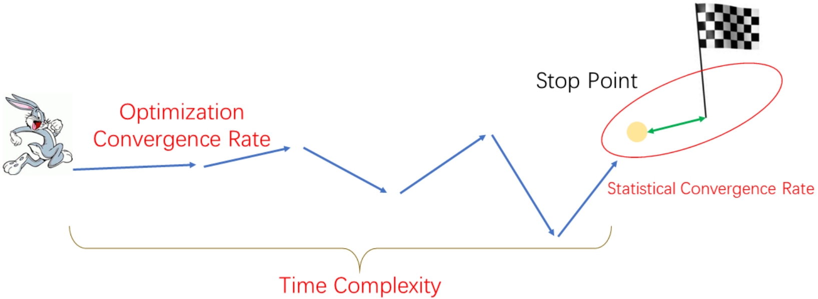
Elementary Estimator

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.

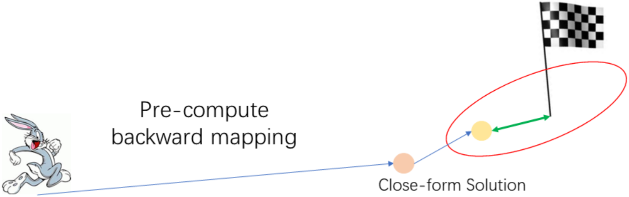
However, a lot of statistical method is slow when p and n are large and they are not scalable at all.

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]

$$\underset{\theta}{\operatorname{argmin}} \mathcal{R}(\theta) \tag{5.1}$$

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

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

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

$$\underset{\theta}{\operatorname{argmin}} ||\theta||_1 \tag{5.3}$$

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

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

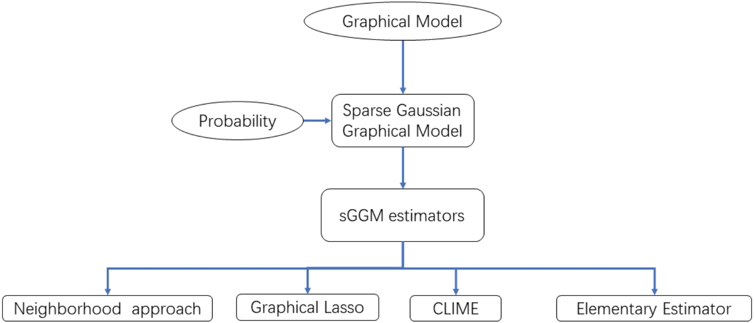
sparse Gaussian Graphical Model estimators

Beilun Wang
Advisor: Yanjun Qi

[1]Department of Computer Science, University of Virginia
http://jointggm.org/

July 28th, 2017

Road Map


Graphical Model
Probability
Sparse Gaussian Graphical Model
sGGM estimators
Neighborhood approach
Graphical Lasso
CLIME
Elementary Estimator

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 $\underset{\theta}{\operatorname{argmin}} \mathcal{L}(\theta) + \lambda_n \mathcal{R}(\theta)$
- 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 $\ln\det(\Omega) - \mathrm{tr}\left(\Omega\widehat{S}\right)$

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 $X_1 = X_j$.
$X_j | X_{\setminus j}\ N(\Sigma_{\setminus j,j}\Sigma^{-1}_{\setminus j,\setminus j}X_{\setminus j}, \Sigma_{jj} - \Sigma_{\setminus j,j}\Sigma^{-1}_{\setminus j,\setminus j}\Sigma_{\setminus j,j})$
Let $\alpha_j := \Sigma_{\setminus j,j}\Sigma^{-1}_{\setminus j,\setminus j}$ and $\sigma_j^2 := \Sigma_{jj} - \Sigma_{\setminus j,j}\Sigma^{-1}_{\setminus j,\setminus j}\Sigma_{\setminus j,j}$. We have that

$$X_j = \alpha_j^T X_{,\setminus j} + \epsilon_j \tag{3.1}$$

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

$$\alpha_j = \operatorname*{argmin}_{\alpha_j} ||\alpha_j^T X_{,\setminus j} - X_j||_2^2 + \lambda ||\alpha_j||_1 \tag{3.2}$$

Review of Lasso solution

Lasso

$$\beta = \operatorname*{argmin}_{\beta} ||\beta^T X - y||_2^2 + \lambda||\beta||_1 \tag{3.3}$$

subgradient method

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

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 $x^* = \text{prox}_{\gamma R}\left(x^* - \gamma \nabla F(x^*)\right)$, where $\gamma > 0$. γ is any positive real number.

Proximal gradient method

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

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?

$$\operatorname*{argmin}_{\Omega} - \ln \det(\Omega) + \operatorname{tr}\left(\Omega \widehat{S}\right) + \lambda_n ||\Omega||_1 \tag{4.1}$$

Proximal gradient method to solve it

Let's do a practice in the white board.

Super Linear algorithm.

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

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

Parallelized Coordinate descent.

approximated quadratic algorithm.

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

CLIME

CLIME[Cai et al.(2011)Cai, Liu, and Luo]

CLIME

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

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

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

$$\widehat{\Omega}_{glasso}^{-1} - \widehat{\Sigma} = \lambda \widehat{Z} \tag{5.2}$$

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

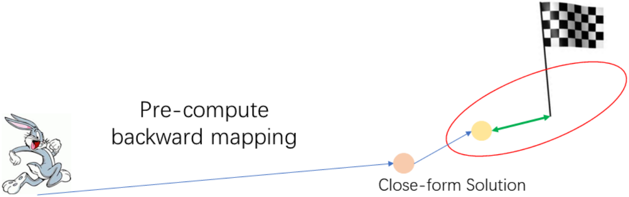
Column-wise estimator

$$\operatorname{argmin} ||\beta||_1 \quad \text{subject to} \quad ||\Sigma\beta - e_j||_\infty \le \lambda$$

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]

$$\underset{\theta}{\operatorname{argmin}} \mathcal{R}(\theta) \tag{6.1}$$

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

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

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

$$\operatorname*{argmin}_{\theta} ||\theta||_1 \tag{6.3}$$

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

Elementary Estimator for sGGM

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

Summary

- We review most sGGM estimators.

References I

T. Cai, W. Liu, and X. Luo.
A constrained l1 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

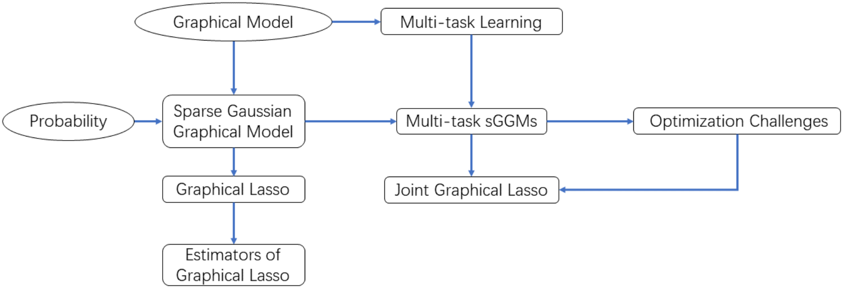
Multi-task sGGMs and its optimization challenges

Beilun Wang
Advisor: Yanjun Qi

[1] Department of Computer Science, University of Virginia
http://jointggm.org/

August 4th, 2017

Road Map


Graphical Model
Multi-task Learning
Probability
Sparse Gaussian Graphical Model
Multi-task sGGMs
Optimization Challenges
Graphical Lasso
Joint Graphical Lasso
Estimators of Graphical Lasso

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 $\ln \det(\Omega) - \operatorname{tr}\left(\Omega\widehat{S}\right)$

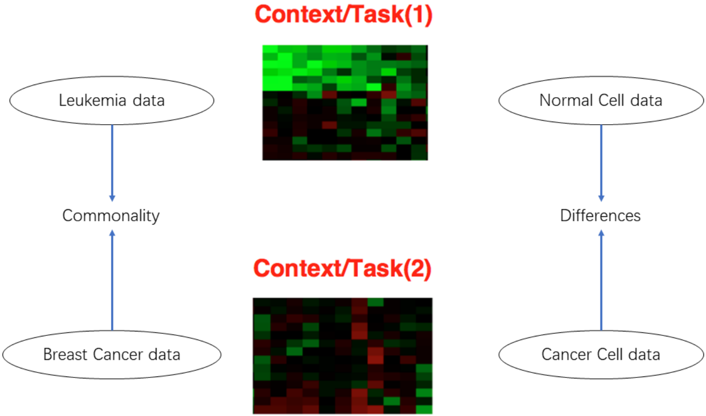
Multi-task Learning

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



Context/Task(1)

Leukemia data

Commonality

Breast Cancer data

Normal Cell data

Differences

Cancer Cell data

Context/Task(2)

Multi-task Learning–Linear Classifier Example

Linear Classifier

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

Multi-task Linear Classifiers

For the *i*-th task,

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

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

$$\sum_i n_i(\ln\det(\Omega^{(i)}) - \operatorname{tr}\left(\Omega^{(i)}\widehat{S}^{(i)}\right)) \tag{4.1}$$

Likelihood with sparsity assumption

$$\sum_i n_i(\ln\det(\Omega^{(i)}) - \operatorname{tr}\left(\Omega^{(i)}\widehat{S}^{(i)}\right)) \tag{4.2}$$

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

Multi-task sGGMs

Likelihood with multi-task setting

$$\sum_i n_i(\ln\det(\Omega^{(i)}) - \mathrm{tr}\left(\Omega^{(i)}\widehat{S}^{(i)}\right)) \tag{4.4}$$

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

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

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

$$-\sum_i n_i(\ln\det(\Omega^{(i)}) + \mathrm{tr}\left(\Omega^{(i)}\widehat{S}^{(i)}\right)) + \lambda_1||\Omega^{(i)}||_1 + \lambda_2 P(\Omega^{(1)}, \Omega^{(2)}, \ldots, \Omega^{(K)}) \tag{4.7}$$

Optimization Challenge of Multi-task sGGMs

General formulation

Likelihood with multi-task setting

$$-\sum_i n_i(\ln\det(\Omega^{(i)}) + \operatorname{tr}\left(\Omega^{(i)}\widehat{S}^{(i)}\right)) \tag{5.1}$$

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

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

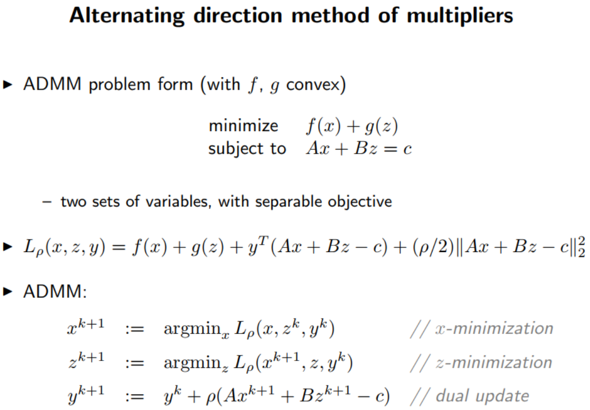
General formulation

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

$$\text{Subject to: } Ax + Bz = c \tag{5.5}$$

Optimization Challenge

Solution–Distributed optimization



Alternating direction method of multipliers

- ADMM problem form (with f, g convex)

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

  – 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:

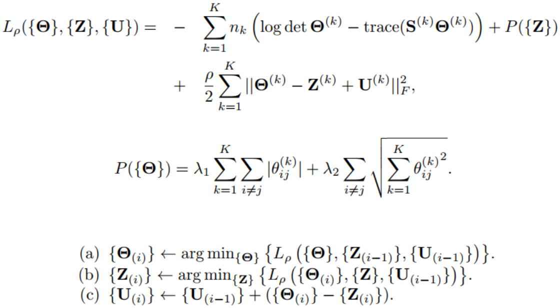
$$\begin{array}{lll} x^{k+1} & := & \operatorname{argmin}_x L_\rho(x, z^k, y^k) & \textit{// } x\textit{-minimization} \\ z^{k+1} & := & \operatorname{argmin}_z L_\rho(x^{k+1}, z, y^k) & \textit{// } z\textit{-minimization} \\ y^{k+1} & := & y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) & \textit{// dual update} \end{array}$$

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



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

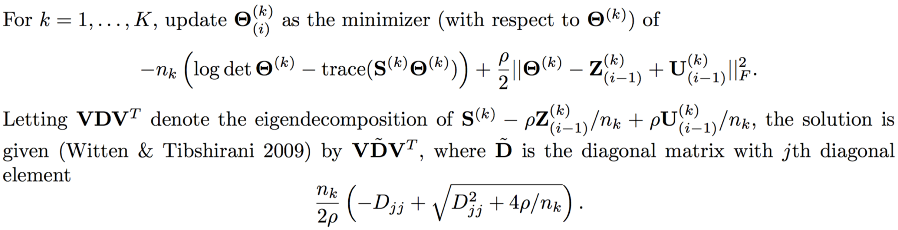
$$P(\{\mathbf{\Theta}\}) = \lambda_1 \sum_{k=1}^{K}\sum_{i\neq j} |\theta_{ij}^{(k)}| + \lambda_2 \sum_{i\neq j}\sqrt{\sum_{k=1}^{K} {\theta_{ij}^{(k)}}^2}.$$

(a) $\{\mathbf{\Theta}_{(i)}\} \leftarrow \arg\min_{\{\mathbf{\Theta}\}} \left\{L_\rho\left(\{\mathbf{\Theta}\},\{\mathbf{Z}_{(i-1)}\},\{\mathbf{U}_{(i-1)}\}\right)\right\}.$

(b) $\{\mathbf{Z}_{(i)}\} \leftarrow \arg\min_{\{\mathbf{Z}\}} \left\{L_\rho\left(\{\mathbf{\Theta}_{(i)}\},\{\mathbf{Z}\},\{\mathbf{U}_{(i-1)}\}\right)\right\}.$

(c) $\{\mathbf{U}_{(i)}\} \leftarrow \{\mathbf{U}_{(i-1)}\} + (\{\mathbf{\Theta}_{(i)}\} - \{\mathbf{Z}_{(i)}\}).$

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



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

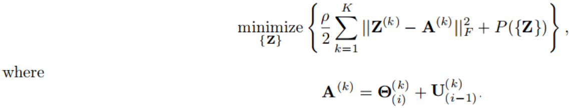
$$-n_k \left(\log \det \boldsymbol{\Theta}^{(k)} - \operatorname{trace}(\mathbf{S}^{(k)}\boldsymbol{\Theta}^{(k)})\right) + \frac{\rho}{2}||\boldsymbol{\Theta}^{(k)} - \mathbf{Z}_{(i-1)}^{(k)} + \mathbf{U}_{(i-1)}^{(k)}||_F^2.$$

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 jth 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.

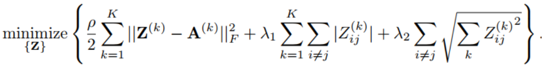
JGL solution – updating $Z^{(i)}$



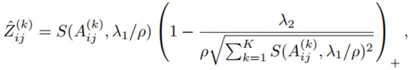
$$\underset{\{\mathbf{Z}\}}{\text{minimize}} \left\{ \frac{\rho}{2} \sum_{k=1}^{K} ||\mathbf{Z}^{(k)} - \mathbf{A}^{(k)}||_F^2 + P(\{\mathbf{Z}\}) \right\},$$

where

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

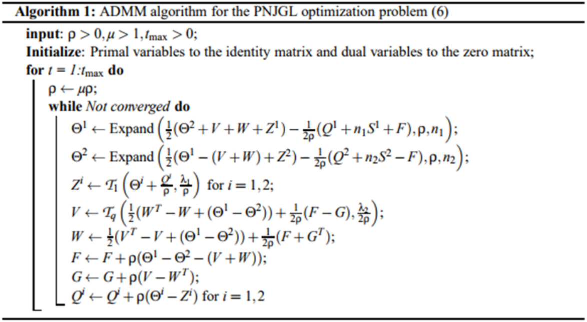


$$\underset{\{\mathbf{Z}\}}{\text{minimize}} \left\{ \frac{\rho}{2} \sum_{k=1}^{K} ||\mathbf{Z}^{(k)} - \mathbf{A}^{(k)}||_F^2 + \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |Z_{ij}^{(k)}| + \lambda_2 \sum_{i \neq j} \sqrt{\sum_k Z_{ij}^{(k)2}} \right\}.$$



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

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
        $\Theta^1 \leftarrow \text{Expand}\left(\frac{1}{2}(\Theta^2 + V + W + Z^1) - \frac{1}{2\rho}(Q^1 + n_1 S^1 + F), \rho, n_1\right)$;
        $\Theta^2 \leftarrow \text{Expand}\left(\frac{1}{2}(\Theta^1 - (V + W) + Z^2) - \frac{1}{2\rho}(Q^2 + n_2 S^2 - F), \rho, n_2\right)$;
        $Z^i \leftarrow \mathcal{T}_1\left(\Theta^i + \frac{Q^i}{\rho}, \frac{\lambda_1}{\rho}\right)$ for $i = 1, 2$;
        $V \leftarrow \mathcal{T}_q\left(\frac{1}{2}(W^T - W + (\Theta^1 - \Theta^2)) + \frac{1}{2\rho}(F - G), \frac{\lambda_2}{2\rho}\right)$;
        $W \leftarrow \frac{1}{2}(V^T - V + (\Theta^1 - \Theta^2)) + \frac{1}{2\rho}(F + G^T)$;
        $F \leftarrow F + \rho(\Theta^1 - \Theta^2 - (V + W))$;
        $G \leftarrow G + \rho(V - W^T)$;
        $Q^i \leftarrow Q^i + \rho(\Theta^i - Z^i)$ 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

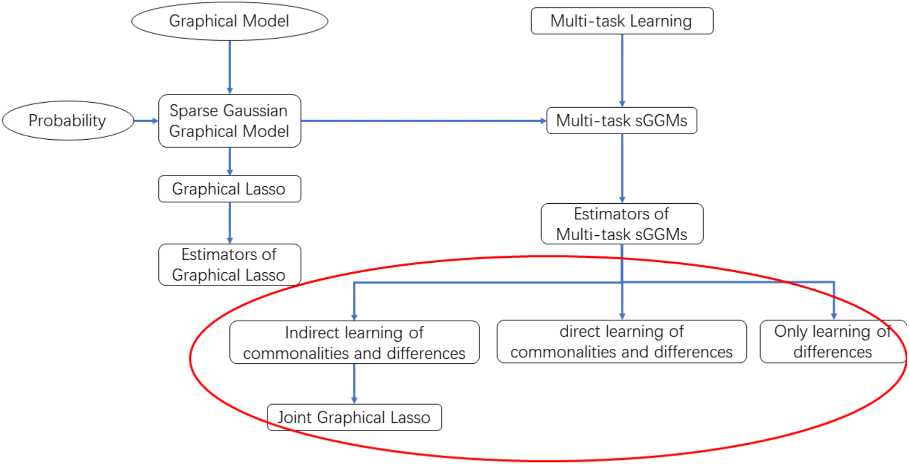
Multi-task sGGMs estimators

Beilun Wang
Advisor: Yanjun Qi

[1]Department of Computer Science, University of Virginia
http://jointggm.org/

August 18th, 2017

Road Map


Graphical Model
Multi-task Learning
Probability
Sparse Gaussian Graphical Model
Multi-task sGGMs
Graphical Lasso
Estimators of Multi-task sGGMs
Estimators of Graphical Lasso
Indirect learning of commonalities and differences
direct learning of commonalities and differences
Only learning of differences
Joint Graphical Lasso

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 chanllenges 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 $\ln\det(\Omega) - \operatorname{tr}\left(\Omega\widehat{S}\right)$

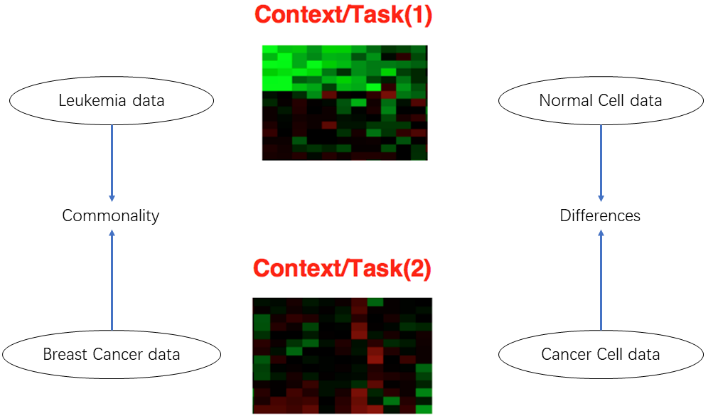
Multi-task Learning

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



Context/Task(1)

Leukemia data

Normal Cell data

Commonality

Differences

Context/Task(2)

Breast Cancer data

Cancer Cell data

Multi-task Learning–Linear Classifier Example

Linear Classifier

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

Multi-task Linear Classifiers

For the *i*-th task,

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

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

$$\sum_i n_i(\ln\det(\Omega^{(i)}) - \operatorname{tr}\left(\Omega^{(i)}\widehat{S}^{(i)}\right)) \tag{4.1}$$

Likelihood with sparsity assumption

$$\operatorname*{argmax}_{\Omega^{(i)}} \sum_i n_i(\ln\det(\Omega^{(i)}) - \operatorname{tr}\left(\Omega^{(i)}\widehat{S}^{(i)}\right)) \tag{4.2}$$

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

Multi-task sGGMs

Likelihood with multi-task setting

$$\underset{\Omega^{(i)}}{\operatorname{argmax}} \sum_i n_i(\ln \det(\Omega^{(i)}) - \operatorname{tr}\left(\Omega^{(i)}\widehat{S}^{(i)}\right)) \tag{4.4}$$

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

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

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

$$\underset{\Omega^{(i)}}{\operatorname{argmin}} - \sum_i n_i(\ln \det(\Omega^{(i)}) + \operatorname{tr}\left(\Omega^{(i)}\widehat{S}^{(i)}\right)) + \lambda_1||\Omega^{(i)}||_1 + \lambda_2 P(\Omega^{(1)}, \Omega^{(2)}, \dots, \tag{4.7}$$

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 $P(\Omega^{(1)}, \Omega^{(2)}, \ldots, \Omega^{(K)})$.

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]

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

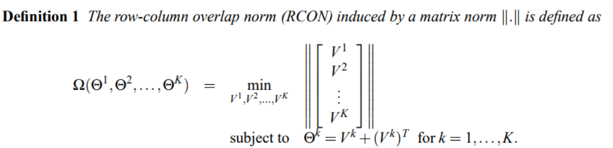
SIMONE[Chiquet et al.(2011)Chiquet, Grandvalet, and Ambroise]

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

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

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

Node JGL[Mohan et al.(2013)Mohan, London, Fazel, Lee, and Wit



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

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

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

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 $\beta^v + \widetilde{\boldsymbol{b}}^{(i),v}$. Here $\widetilde{\boldsymbol{b}}^{(i),v} \sim N(0, \Phi^v)$.

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

SIMULE

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

Direct modeling commonalities and differences – SIMULE

SIMULE

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

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

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:

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

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

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

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

Density Ratio

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

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

Direct modeling the differential networks III: Density Ratio

Density Ratio for Markov Random Field

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

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

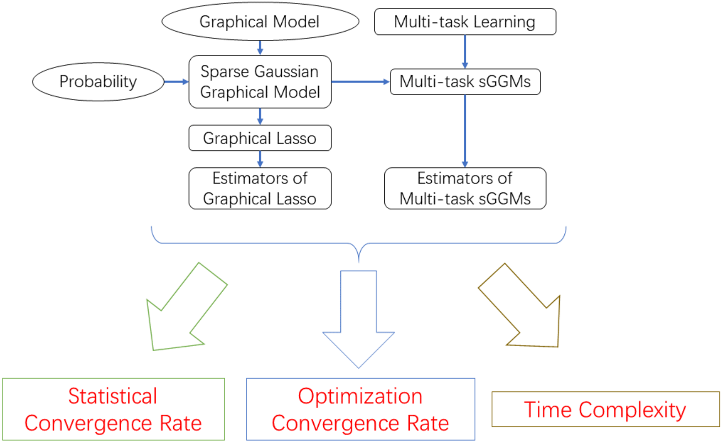
A deep introduction of the metrics for evaluating an/a estimator/learner

Beilun Wang
Advisor: Yanjun Qi

[1]Department of Computer Science, University of Virginia
http://jointggm.org/

Sep 22nd, 2017

Road Map


Graphical Model
Multi-task Learning
Probability
Sparse Gaussian Graphical Model
Multi-task sGGMs
Graphical Lasso
Estimators of Graphical Lasso
Estimators of Multi-task sGGMs
Statistical Convergence Rate
Optimization Convergence Rate
Time Complexity

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.

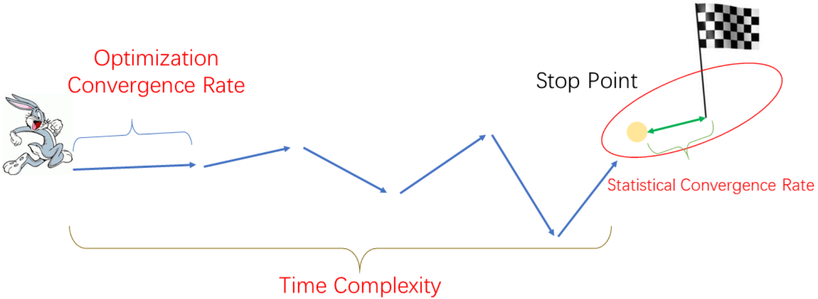
Background: Two major properties

- 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



Optimization Convergence Rate

Stop Point

Statistical Convergence Rate

Time Complexity

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]

$$\underset{\theta}{\operatorname{argmin}} \mathcal{R}(\theta) \tag{4.1}$$

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

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

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

$$\underset{\theta}{\operatorname{argmin}} ||\theta||_1 \tag{4.3}$$

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

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

Optimization Convergence Rate : Definition

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

Optimization Convergence Rate : Definition

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

Optimization Convergence Rate : Definition

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

Optimization Convergence Rate : Definition

- Linearly Converge: $\lim\limits_{k\to\infty} \frac{|\theta_{k+1}-L|}{|\theta_k-L|} = \mu_k$
-
  - Linearly, if $\mu_k \in (0,1)$
  - Superlinearly, if $\mu_k \to 0$ when $k \to \infty$.
  - Sublinearly, if $\mu_k \to 1$ when $k \to \infty$
- Higher order: $\lim\limits_{k\to\infty} \frac{|x_{k+1}-L|}{|x_k-L|^q} > 0$.
- 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 × 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	$O(Tp^2)$	$O(p^5)$	$O(p^2)$	$O(Tp^3)$	$O(Tp^2)$	$O(K^4p^5$

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.