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

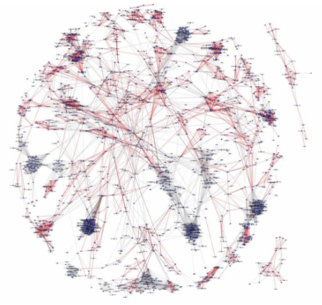
Beilun Wang[1]
Advisor: Yanjun Qi[1]
Dissertation Committees:
Mahmoody Mohammad (Committee Chair)[1]
Xiaojin (Jerry) Zhu[2]
Farzad Farnoud[1]
Tingting Zhang[1]

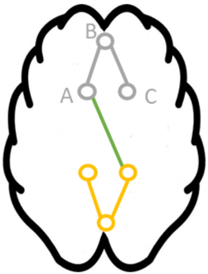
[1]University of Virginia
[2]University of Wisconsin–Madison

August 24, 2018

Background

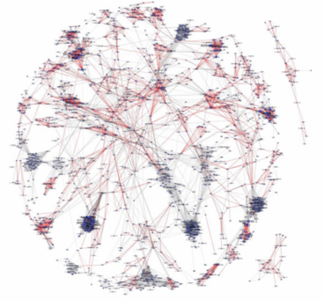
Background: Entity Graph

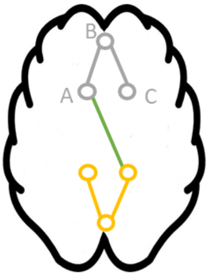




- Many applications need to know interactions among entities:
  - Gene Interactions
  - Brain connectivity

Background: Entity Graph

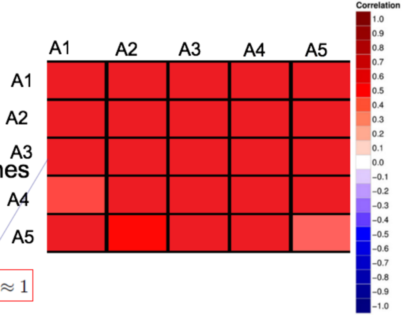




- Many applications need to know interactions among entities:
  - Gene Interactions
  - Brain connectivity
- Why to study the entity graph
  - Understanding
  - Diagnosis, e.g., marker
  - Treatment, e.g., drug development.

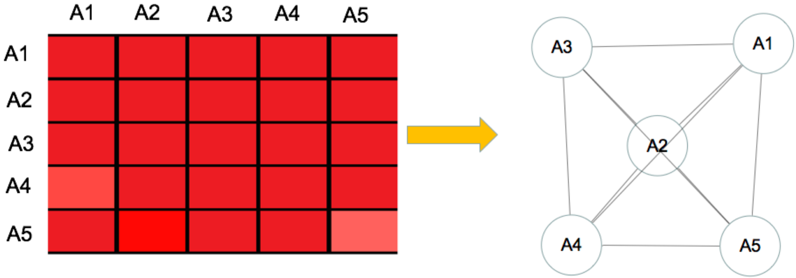
Background: What Type of Edges? Correlation to Conditional dependency

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

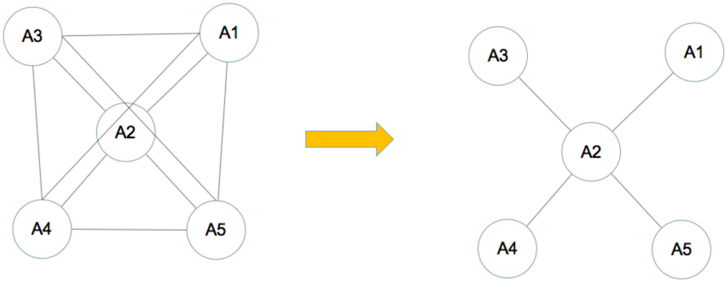


$\boxed{Cor(A_1, A_3) \approx 1}$

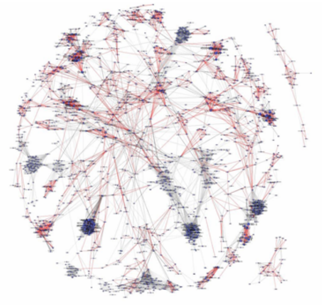
Background: What Type of Edges? Correlation to Conditional dependency

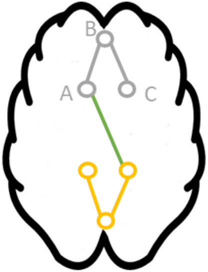


Background: What Type of Edges? Correlation to Conditional dependency



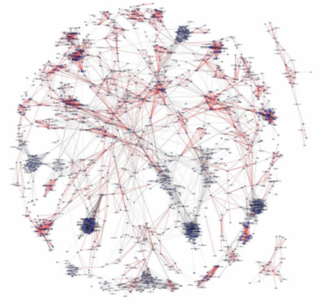
Background: How to Infer Entity Graph?

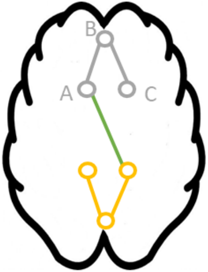




- To measure conditional dependency interactions physically.
- Largely unknown and hard to measure physically.

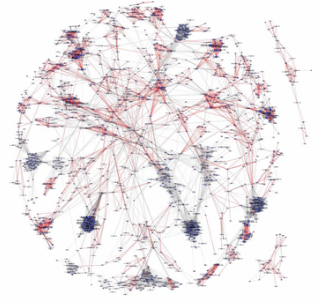
Background: How to Infer Entity Graph?

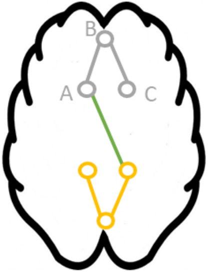




- To measure conditional dependency interactions physically.
- Largely unknown and hard to measure physically.
    - #Physical check for all possible conditional dependency edges = 2^p (binary experiments)
    - For example, $p = 160$ important regions in human brain
    - For example, $p = 30000$ genes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

Background: How to Infer Entity Graph?

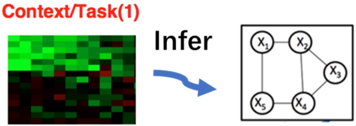




- To measure conditional dependency interactions physically.
- Largely unknown and hard to measure physically.
  - #Physical check for all possible conditional dependency edges = 2^p (binary experiments)
  - For example, $p = 160$ important regions in human brain
  - For example, $p = 30000$ genes in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

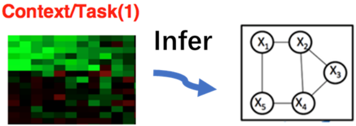
Background: Entity graphs from Observed Samples (Entity as Feature)

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



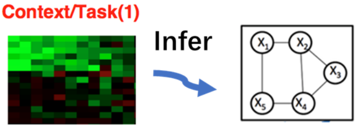
Background: Entity graphs from Observed Samples (Entity as Feature)

- ~~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.



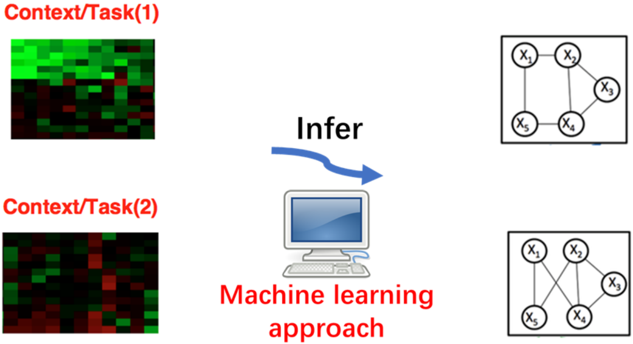
Background: Entity graphs from Observed Samples (Entity as Feature)

- ~~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



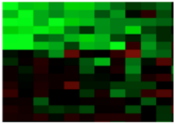
Background: Entity graphs from Heterogeneous Data (Entity as Feature)

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

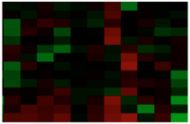


Background: Entity graphs from Heterogeneous Data

Context/Task(1)



Context/Task(2)



Case I:

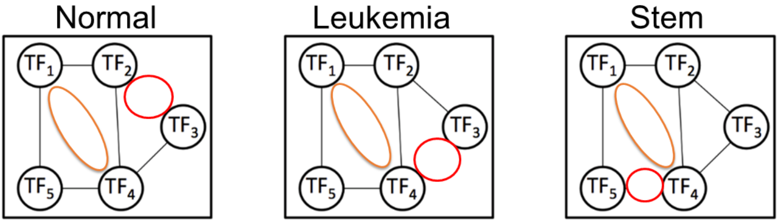
Leukemia Dataset → Commonality ← Breast Cancer Dataset

Case II:

Normal Cell Dataset → Differences ← Cancer Cell Dataset

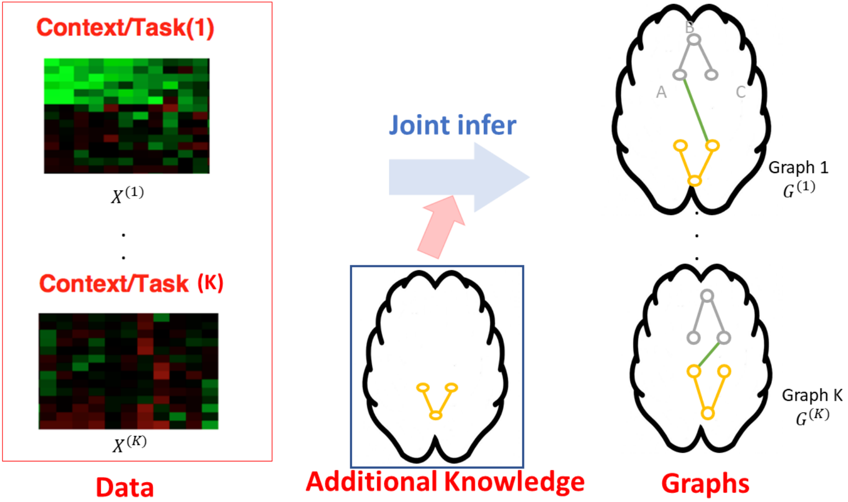
Task I: Learning multiple related graphs

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

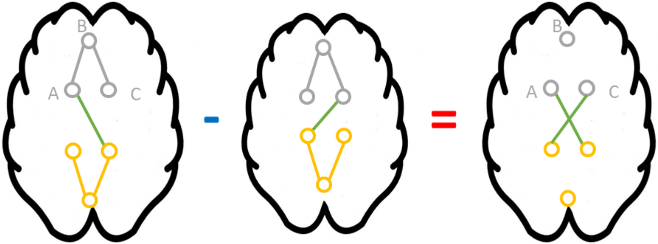


Task II: Integrating additional knowledge

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



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} $(W_S, W_S, \ldots, W_S)$

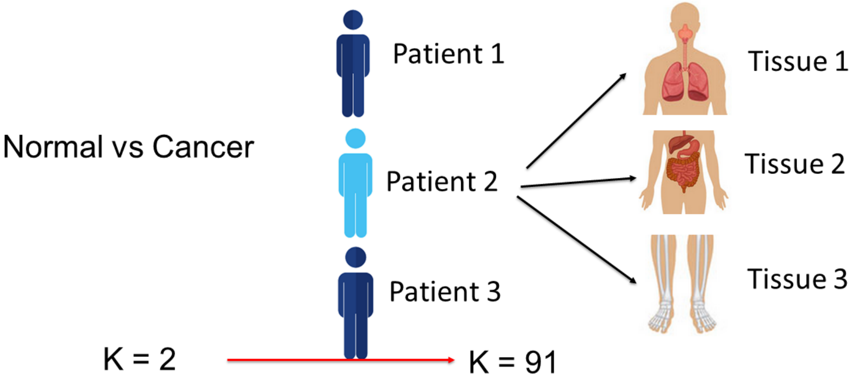
Motivation

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

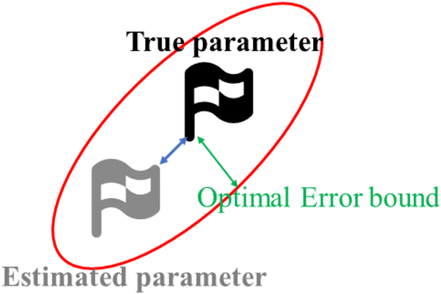
Computer Clusters

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



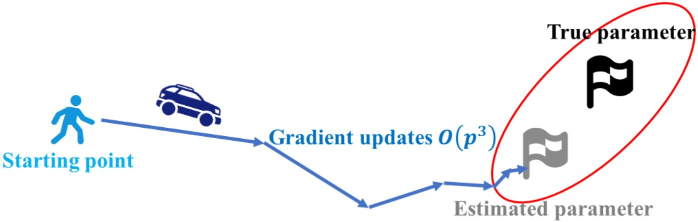
Our Aim: Fast and Scalable estimators for three types of joint graphs estimation

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

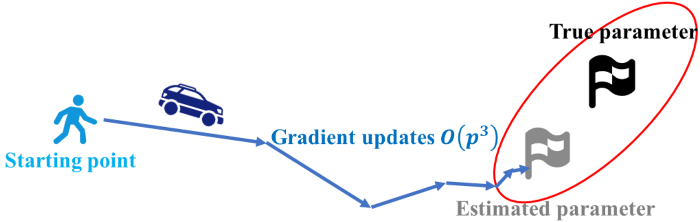
Solution for Limitations - Elementary Estimator

Background: summary of the previous optimization strategy

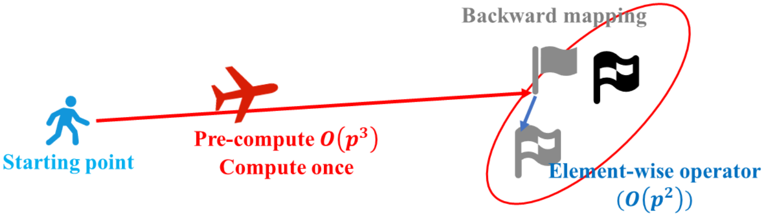
- e.g., ADMM algorithm



- Previous studies:

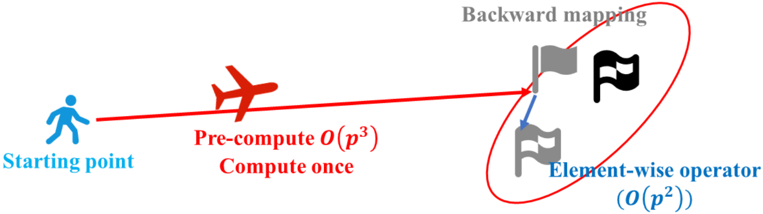


- Elementary Estimator:



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: $\operatorname*{argmax}_{\theta} \mathcal{L}(\theta)$
  - Already close to true parameter
  - But without assumptions e.g., sparse
  - For instance, linear regression solution $(X^T X)^{-1} X^T Y$



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

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

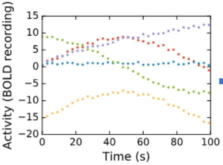
$$[S_\lambda(A)]_{ij} = \text{sign}(A_{ij}) \max(|A_{ij}| - \lambda, 0) \tag{3.3}$$

- 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

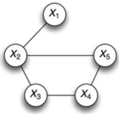
sGGM

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



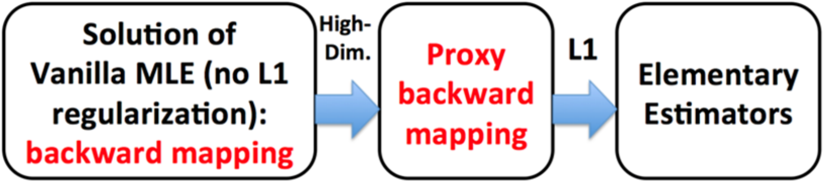
Connectome

EE-GM: Elementary Estimator for sGGM

- Vanilla MLE: $\underset{\Omega}{\text{argmin}} - \log(\det(\Omega)) + < \Omega, \Sigma >$
- 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, $\widehat{\theta}_n = [T_v(\widehat{\Sigma})]^{-1}$



EE-GM: Elementary Estimator for sGGM

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

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

$\Downarrow$

EE-sGGM

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

-

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

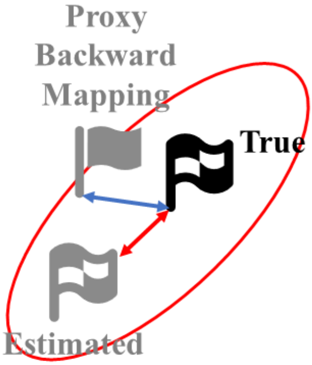
- Error bound:

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

- Condition:

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

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



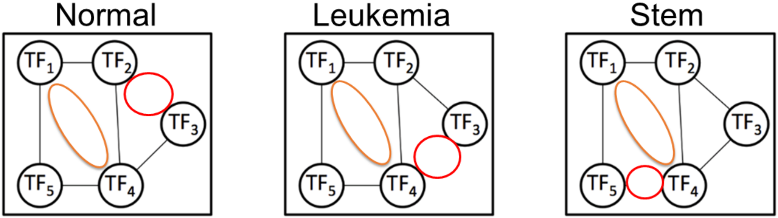
Method I: FASJEM

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

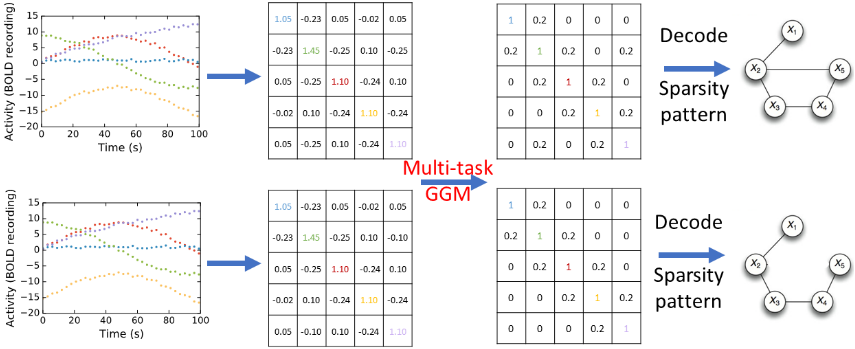
Task I: Learning multiple related graphs

- 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] 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

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

- 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

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

FASJEM: Variations

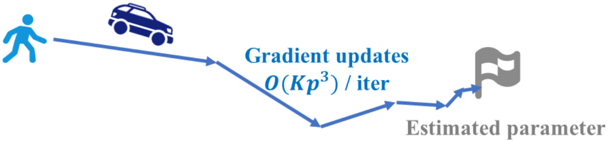
- FASJEM-G:

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

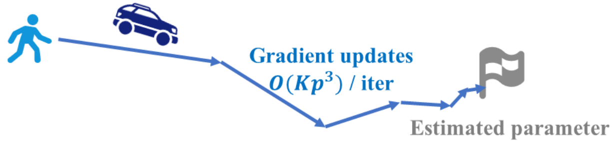
- FASJEM-I:

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

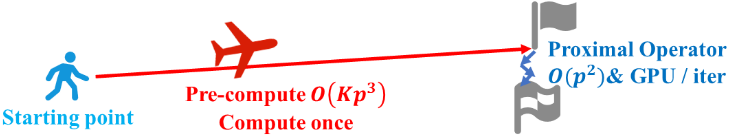
- JGL solution:



- JGL solution:

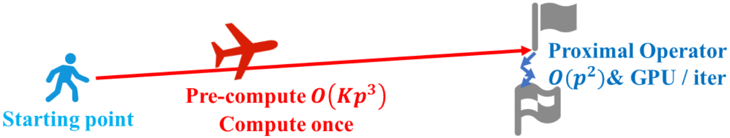


- FASJEM solution:



FASJEM: Optimization Solution – Proximal algorithm

- FASJEM solution:



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

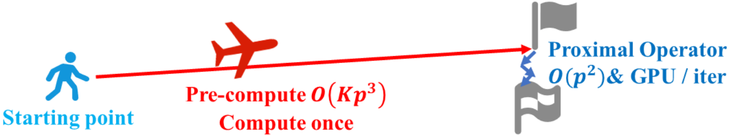
$$\mathrm{prox}_{\gamma||\cdot||_1}(x) = \begin{cases} x_{j,k}^{(i)} - \gamma, \; x_{j,k}^{(i)} > \gamma \\ 0, \; |x_{j,k}^{(i)}| \leq \gamma \\ x_{j,k}^{(i)} + \gamma, \; x_{j,k}^{(i)} < -\gamma \end{cases} \tag{4.9}$$

$$\Longrightarrow$$

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

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

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

$\Longrightarrow$

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

FASJEM: Computational Complexity

The best baseline of	Task I	Task II	Task III
Computational complexity	$O(Kp^3)$ / iter	$O(K^4p^5)$	$O(p^3)$ / 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 $\frac{\log(Kp)}{n_{tot}} < K \frac{\log p}{n_{tot}}$

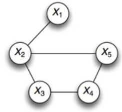
Results: Theoretical Analysis

- $p' = \max(Kp, n_{tot})$
- 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

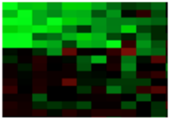


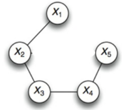
1	0.2	0	0	0
0.2	1	0.2	0	0.2
0	0.2	1	0.2	0
0	0	0.2	1	0.2
0	0.2	0	0.2	1

Inverse

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

Context/Task(1)

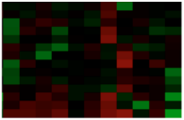




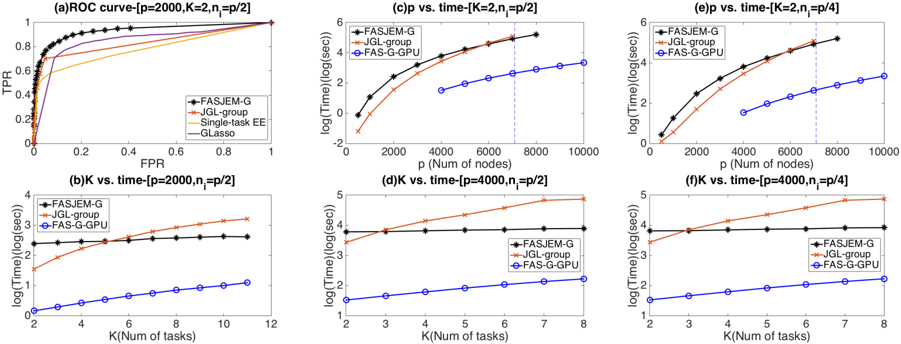
1	0.2	0	0	0
0.2	1	0.2	0	0
0	0.2	1	0.2	0
0	0	0.2	1	0.2
0	0	0	0.2	1

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

Context/Task(2)

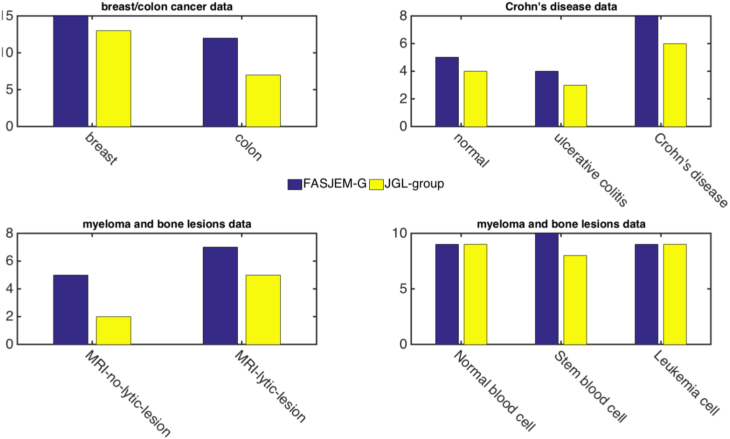


Results: Synthetic Data Results



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

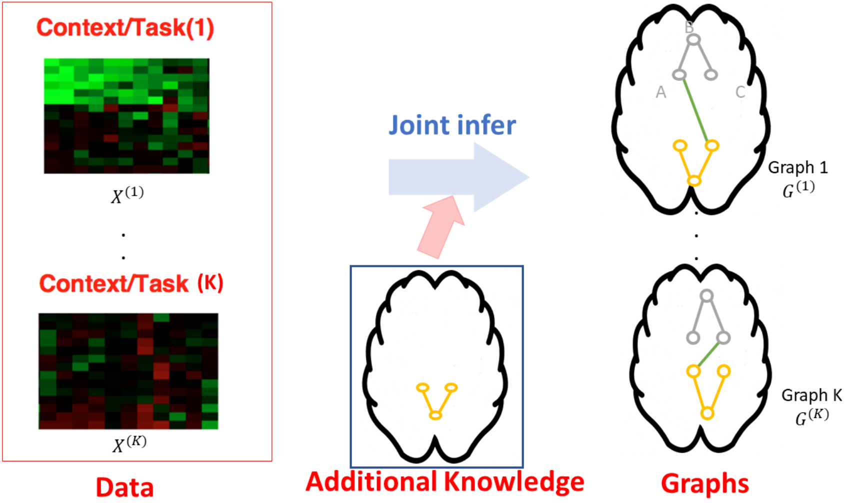
5 Method II: JEEK
- Background
- Method
- Results

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



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

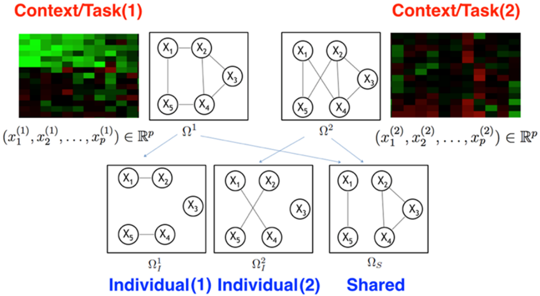
- Integrating additional knowledge through a novel regularization function $\mathcal{R}(\cdot)$

KW-norm

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

- $\Omega^{(i)} = \Omega_I^{(i)} + \Omega_S$
- $\{W_I^{(i)}\}$: 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

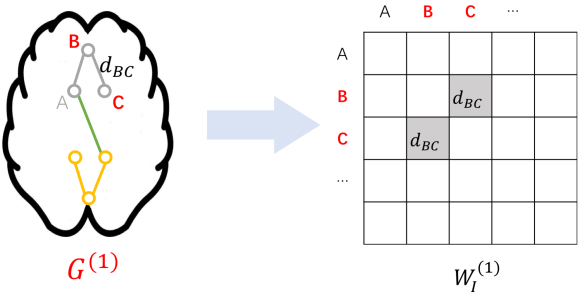
KW-norm

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

- 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.

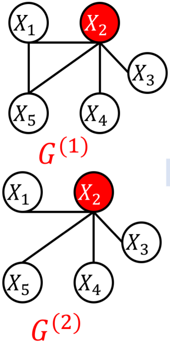
Example I: KW-norm representing the edge-level knowledge

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



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

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

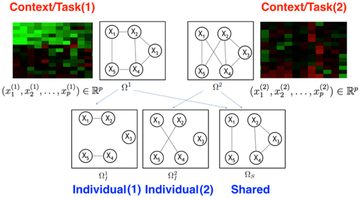


$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & & 1/\gamma & 1 & 1 & 1 \\
2 & 1/\gamma & & 1/\gamma & 1/\gamma & 1/\gamma \\
3 & 1 & 1/\gamma & & 1 & 1 \\
4 & 1 & 1/\gamma & 1 & & 1 \\
5 & 1 & 1/\gamma & 1 & 1 &
\end{array}
$$

W_s

Background: SIMULE

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



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

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

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

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

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

W-SIMULE

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

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

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

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

$$+$$

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

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

Proposed method: JEEK – Solution

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

$$\operatorname*{argmin}_{a_i,b} \sum_i |w_i a_i| + K|w_s b|$$
$$\text{Subject to: } |a_i + b - c_i| \le \frac{\lambda_n}{\min(w_i, w_s)}, \tag{5.7}$$
$$i = 1, \ldots, K$$

[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

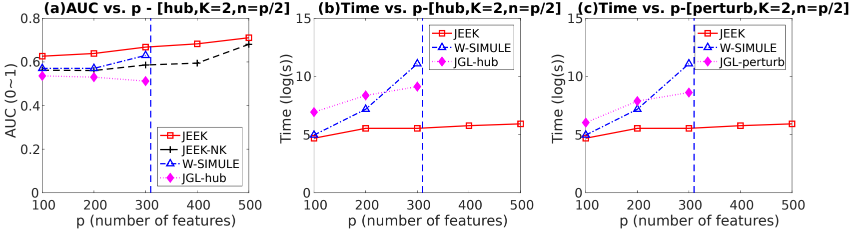
- Sharp convergence rate as the state-of-art

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

Where a, c, κ_1 and κ_2 are constants

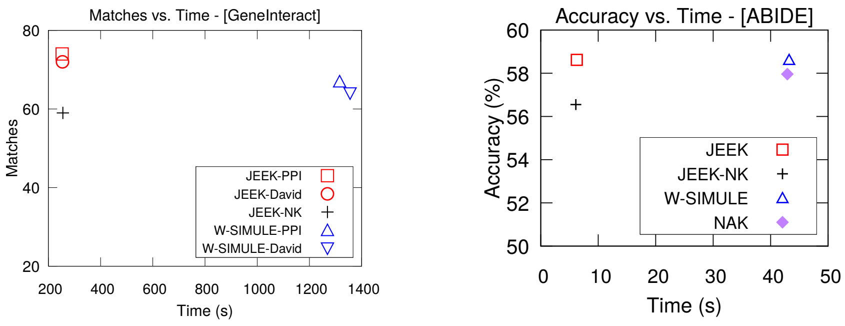
$$
\begin{aligned}
&||\widehat{\Omega}^{tot} - \Omega^{tot*}||_F \\
&\leq \frac{16\kappa_1 a \max_{j,k}(W_I^{tot}{}_{j,k}, W_S^{tot}{}_{j,k})}{\kappa_2}\sqrt{\frac{(k_i + k_s)\log(Kp)}{n_{tot}}}
\end{aligned}
\tag{5.9}
$$

Empirical Results on Multiple Synthetic Datasets



- JEEK outperforms the speed of the state-of arts significantly faster ($\sim$ 5000$\times$ 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

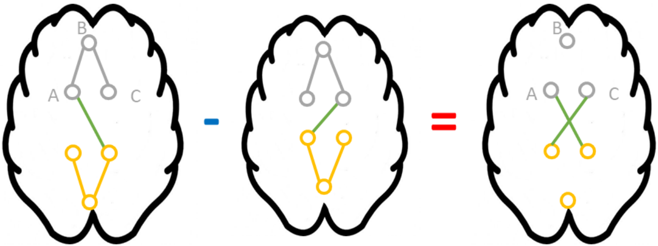
5 Method II: JEEK
- Background
- Method
- Results

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



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

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

Subject to:

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

$$\Delta = \Omega_d - \Omega_c$$
$$\Longrightarrow$$

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

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)

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

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

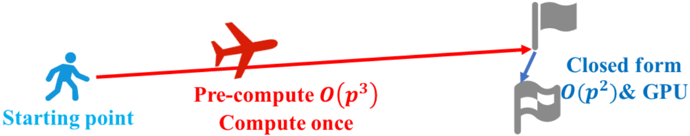
$$\underset{\Delta}{\text{argmin}} \, ||\Delta||_1$$
$$\text{Subject to: } ||\Delta - \Big([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}\Big)||_\infty \leq \lambda_n \tag{6.4}$$

- Close form

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

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

- 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 program-ming	SVD
Our ap-proach	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

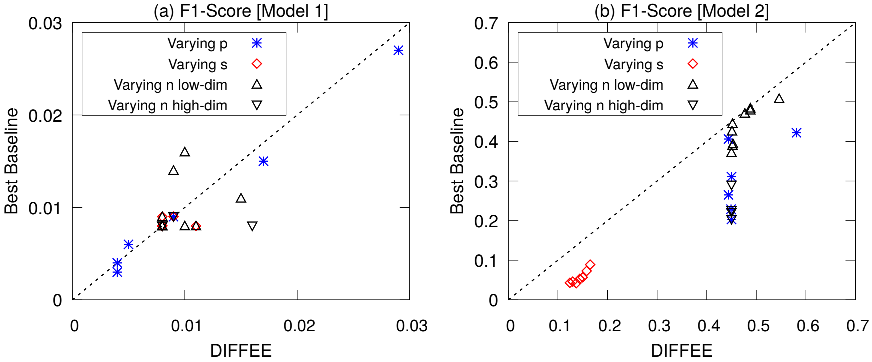
1 Background
2 Motivation
3 Solution for Limitations - Elementary Estimator
4 Method I: FASJEM
- Background
- Method
- Results
5 Method II: JEEK
- Background
- Method
- Results
6 Method III: DIFFEE
- Method
- Results
7 Discussion
- Questions from Proposal
- Future works

Results: Theoretical Analysis

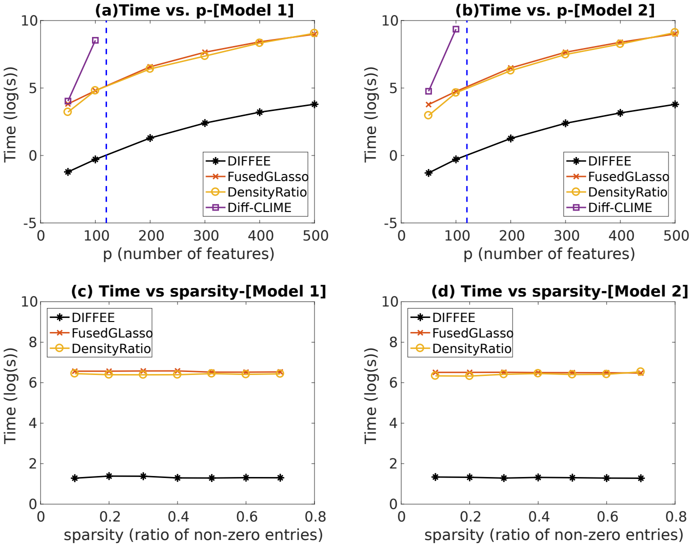
- Sharp convergence rate as the state-of-art

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

Results: Synthetic Data Results



Results: Synthetic Data Results



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

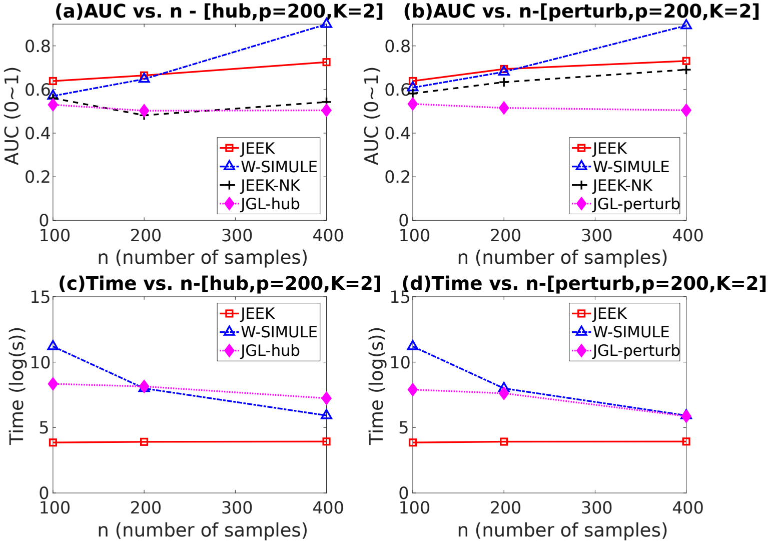
Trade-off

- 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)
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))

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

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

- $\{W^{(i)}\}$: 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

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

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

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

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

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

- 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

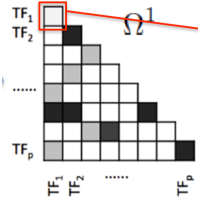
Back-up: Connecting to Bayesian Statistics

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

Back-up: Proximal algorithm Basics

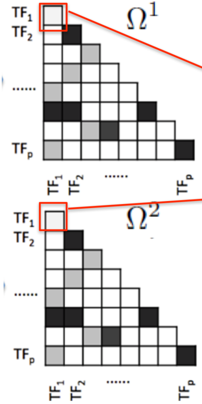
- proximity definition:
- $\text{prox}_h(x) = \underset{u}{\operatorname{argmin}}(h(u) + \frac{1}{2}||u - x||_2^2)$
- $\underset{x}{\operatorname{argmin}} f(x) = g(x) + h(x)$
- proximal gradient descent:
- $x^{(k)} = \text{prox}_{t_k h}(x^{(k-1)} - t_k \bigtriangledown g(x^{(k-1)}))$

Back-up: Proximal algorithm for FASJEM



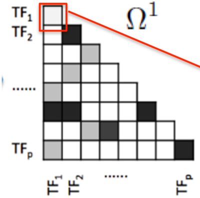
$$(\mathrm{prox}_{\gamma||\cdot||_1}(x))_i = \begin{cases} x_i - \gamma,\, x_i > \gamma \\ 0,\, |x_i| \le \gamma \\ x_i + \gamma,\, x_i < -\gamma \end{cases}$$

Back-up: Proximal algorithm for FASJEM



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

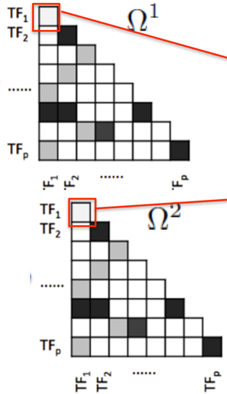
Back-up: Proximal algorithm for FASJEM



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

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

Back-up: Proximal algorithm for FASJEM



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

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

$$\text{prox}_{\gamma g_2}(x) = \text{proj}_{||x-a||^*_{\mathcal{G},2} \le \lambda} = \begin{cases} x_g, & ||x_g - a_g||_2 \le \lambda \\ \lambda \frac{x_g - a_g}{||x_g - a_g||_2} + a_g, & ||x_g - a_g||_2 > \lambda \end{cases}$$