A Fast and Scalable Joint Estimator for Learning Multiple Related Sparse Gaussian Graphical Models

Beilun Wang, Ji Gao, Yanjun Qi

Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTAT17), PMLR 54:1168-1177, 2017.

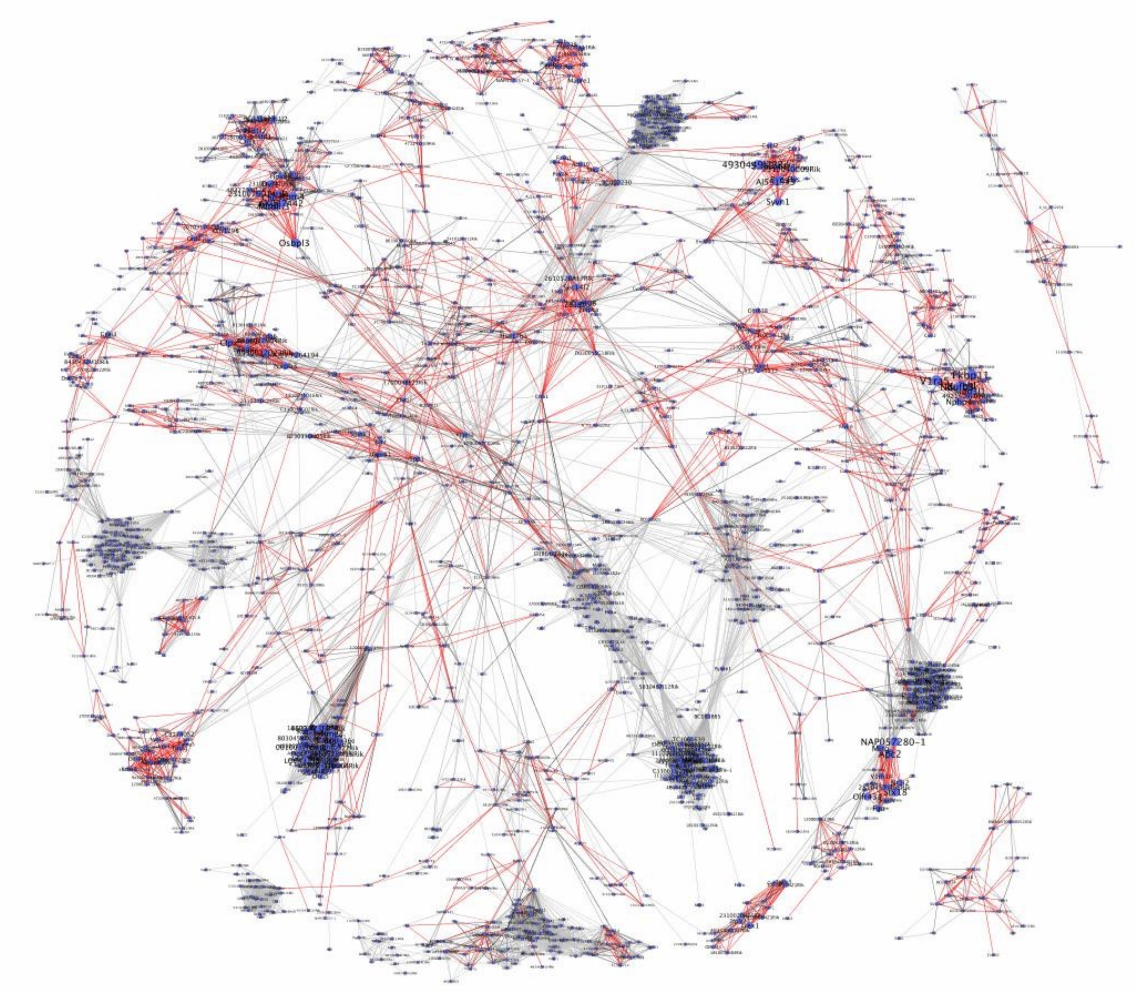
R package: fasjem

http://jointggm.org

```
install.packages("fasjem")
library(fasjem)
demo(fasjem)
```

Motivation: Entity Graph

Interaction among genes

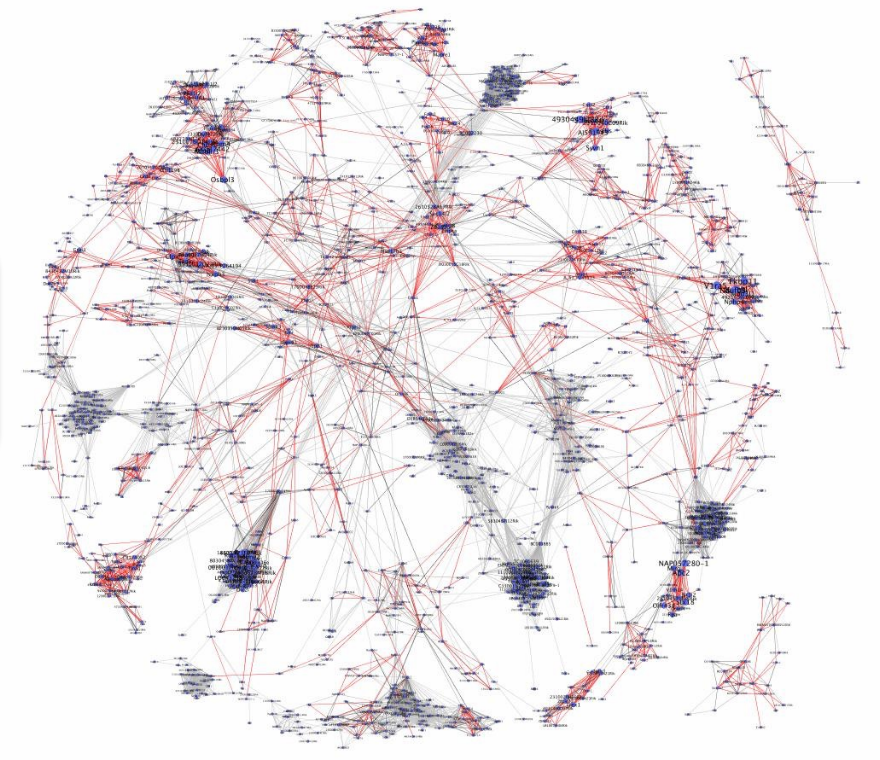


Social Network

Motivation: Data to Graph

Samples-->	T1	T2	T3	T4	T5	T6	T7	N1	N2	N3	N4	*P* value
Genes				Expression level relative to non-tumor pool								
Gene 1	2.4		2	2.5	1.5	2	2.3	1.7	1	1	0.81	9.5E-06
Gene 2	2.9	3.2	1.4	1.7	2.6	3.7	2.5	1	0.9	0.7	0.7	8.0E-05
Gene 3	2	3.5	1.4	1.8	2	2.5	2.2	0.7	1	0.7	0.6	8.6E-05
Gene 4	2.2	2.2	2.7	1.3	2.3	3.7	1.7	0.7	0.9	0.9	0.9	0.00012
Gene 5	5.2	2	3.7	2	5.8	3.2	1.6	0.7	0.7	0.7	0.6	0.00014
Gene 6	6.9	15	18	5.8	12	21	2.3	0.9	1.6	0.7	1.2	0.00015
Gene 7	5.4	2.1	3	2.2	3.5	2.8	1.5	0.8	1.2	0.9	0.8	0.00022
Gene 8	3.1	2.3	1.8	1.7	2.9	1.5	1.2	0.7	0.7	1	0.7	0.00023
Gene 9	9.7	25	23	6.1	9.5	23	2.4	1.2	1.6	0.8	1.1	0.00024
Gene 10	7.6	14	13	4.7	8.2	24	2.1	0.9	1	0.8	1.1	0.00025
Gene 11	4.8	7.7	2.1	2.3	6.6	3.7	7.4	1.1	1.2	0.6	1.2	0.00028
Gene 12	3.6	5.7	3.8	3.3	4.7	6.6	1.8	1.7	0.9	1.1	1	0.00029
Gene 13	5.7	9.8	12	4.5	6	17	1.7	1	1.3	0.8	0.8	0.00031
Gene 14	1.5	2.1	1	1.1	1.2	1.2	1.4	0.6	0.8	0.6	0.8	0.00031
Gene 15	2.5	2.9	1.9	1.8	5.5	2	1.3	1	0.8	0.7	0.7	0.0004
Gene 16	2.2	1.5	1.3	1.2	1.4	1.8	1.1	0.8	0.8	0.8	0.8	0.00042
Gene 17	5.9	2	3.4	2.5	4.3	3.1	2.1	1.2	1.3	1.5	1	0.00048
Gene 18	4	1.6	2.8	1.4	2.9	2.2	1.6	0.9	0.9	1	0.8	0.00052
Gene 19	1.6	1.5	2.3	1.4	1.4	1.8	1.6	1.1	1.2	1	1.1	0.00059
Gene 20	3.9	6.7	6.6	2.3	4	11	1.5	0.8	1.1	0.8	0.8	0.00059
Gene 21	5.3	1.8	2.6	1.4	3.4	2.2	1.6	0.7	0.8	0.8	0.8	0.0006
Gene 22	4.2	1.9	1	2	4.2	4.3	7.9	16	1.1	1.3	10.9	0.00061
Gene 23	2.3	1.3	2.5	1.8	5.7	2.2	1.8	1.1	0.7	0.6	0.7	0.00066
Gene 24	2.9	1	2.9	2.2	3.8	1.9	2.3	0.9	0.8	0.9	0.8	0.00071
Gene 25	2.6	1.4	1.7	1.4	2.4	1.7	1.3	0.9	0.9	0.9	0.9	0.00079
Gene 26	5.8	2.3	3.4	2.1	5	4.7	1.7	1.2	0.9	1.2	0.6	0.0009
Gene 27	5.7	2	4	2.4	5	3.5	1.5	0.8	1.2	1.3	1.1	0.00093
Gene 28	1.6	2.8	1.7	1.7	1.5	2.8	1.9	1.2	1.1	0.8	1.1	0.00094

trends in Biotechnology



Many entities' data

Inference Important

Few entities graph

Motivation: Data Heterogeneity across context

Samples of the same set of genes(human genes)
Vary across Normal vs Leukemia vs Stem

Normal	Leukemia	Stem
rows 1, 2, ..., p: $X^{1}_{p*n_1}$	rows 1, 2, ..., p: $X^{2}_{p*n_2}$	rows 1, 2, ..., p: $X^{3}_{p*n_3}$

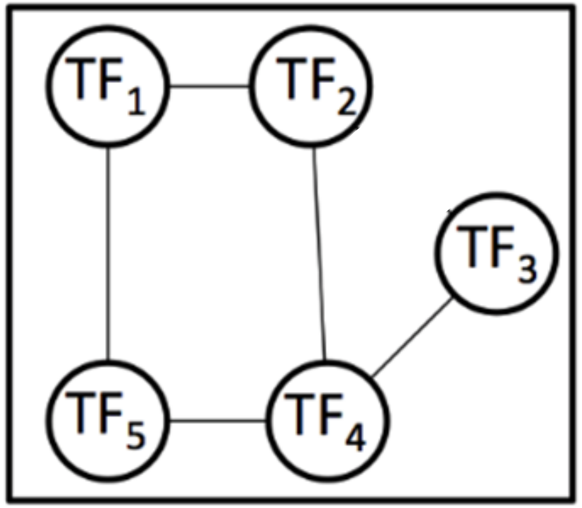
Notation

p represents the number of nodes or features

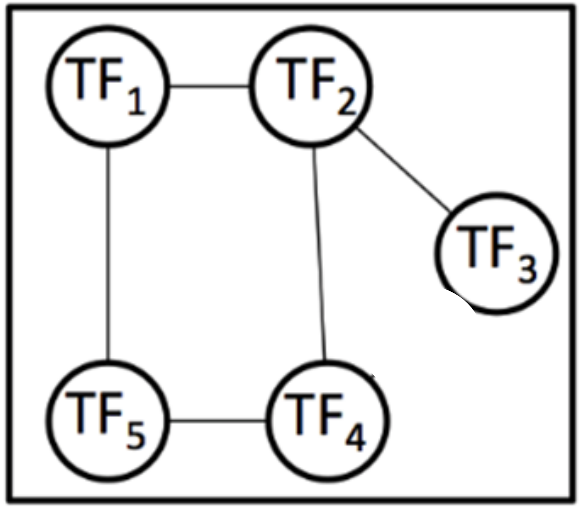
K represents the number of tasks or contexts

Motivation: Entity Graphs vary across contexts

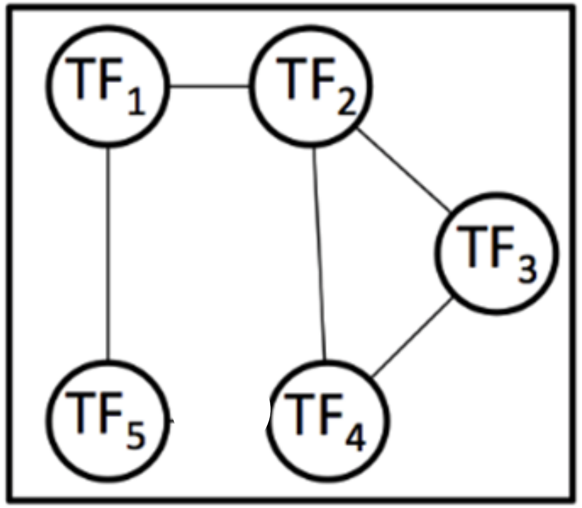
Different but related entity graphs



Normal



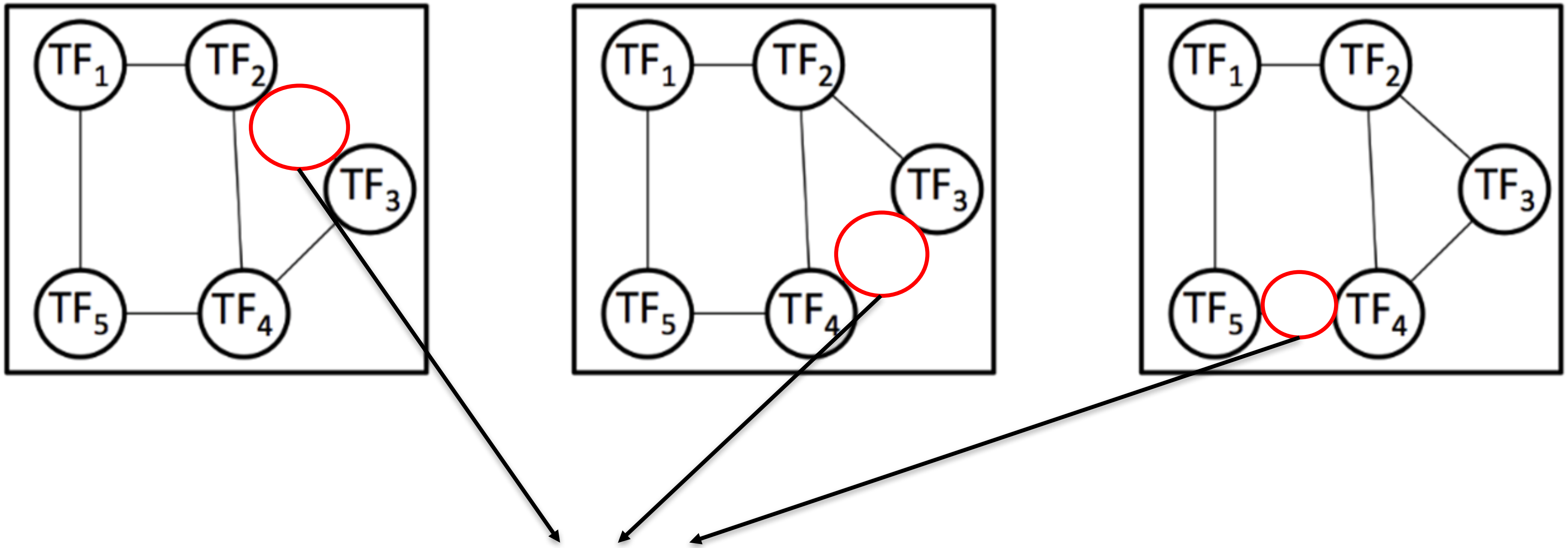
Leukemia



Stem

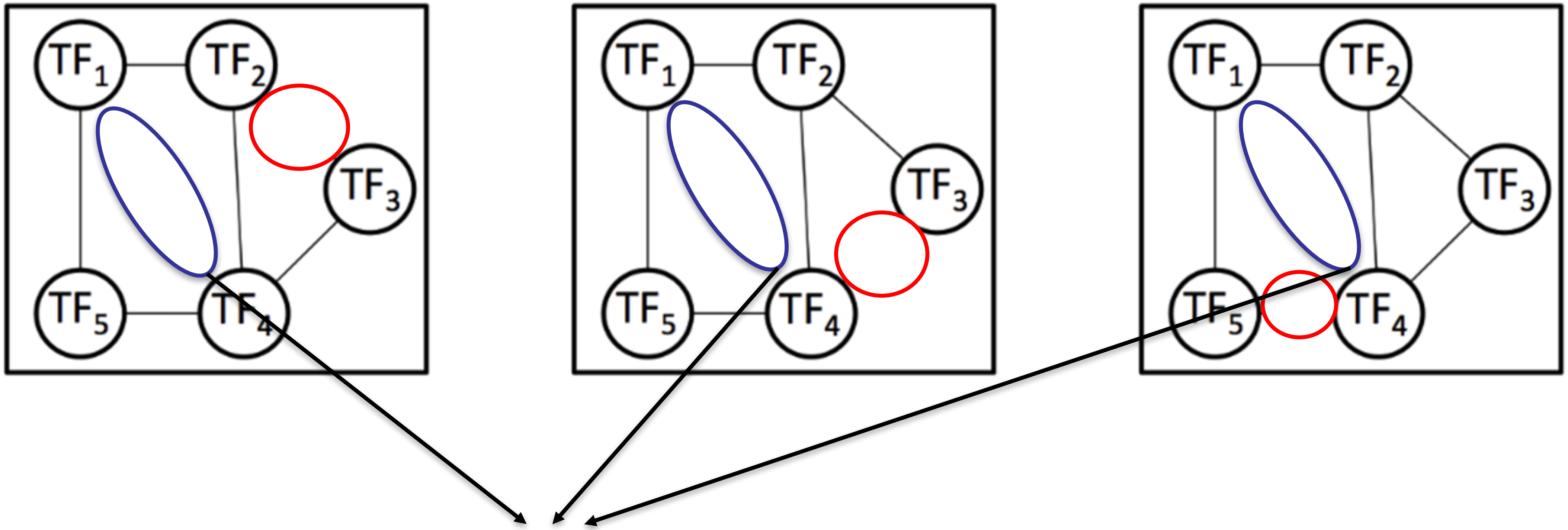
Graphs are sparse

Difference among related graphs: Sparsity



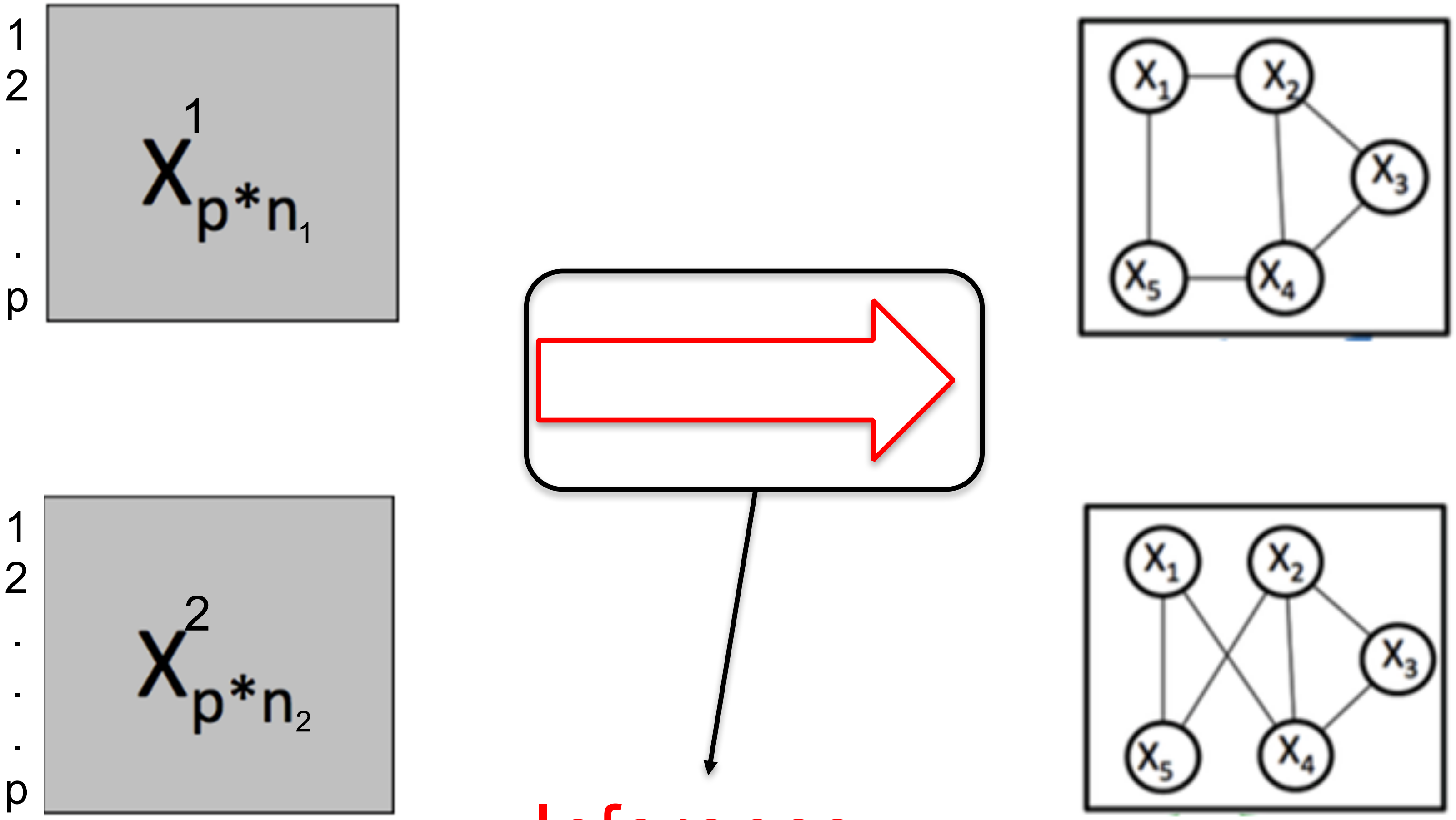
Sparsity Patterns are different
e.g., (TF_2, TF_3)

Similarity among related graphs: Group Sparsity



Group Sparsity means
e.g., (TF_1, TF_4) no edge pattern across three

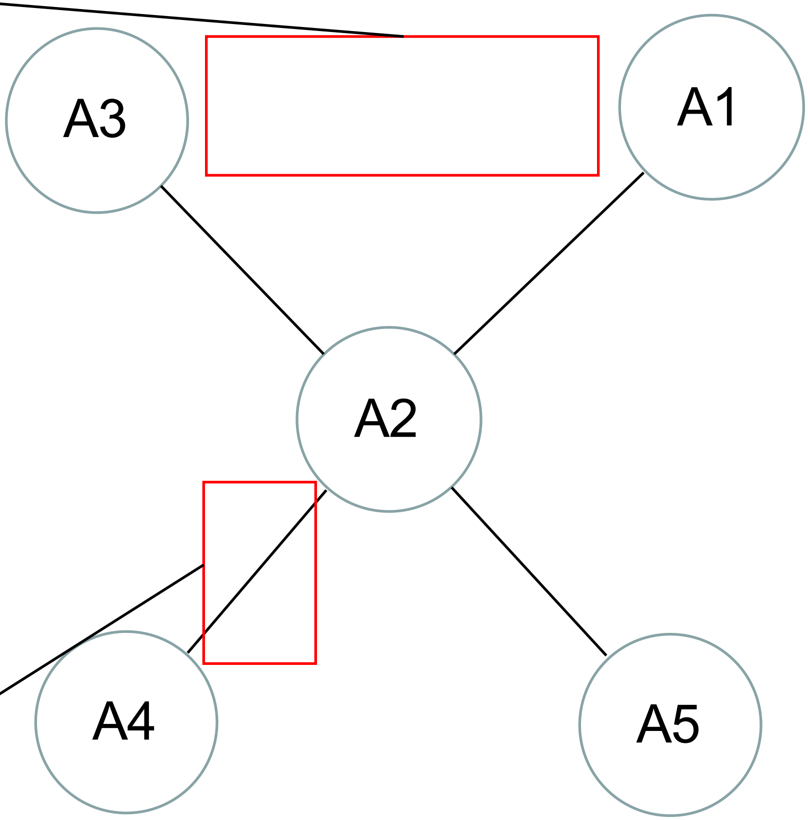
Motivation: Data to Graphs across context



Inference Important

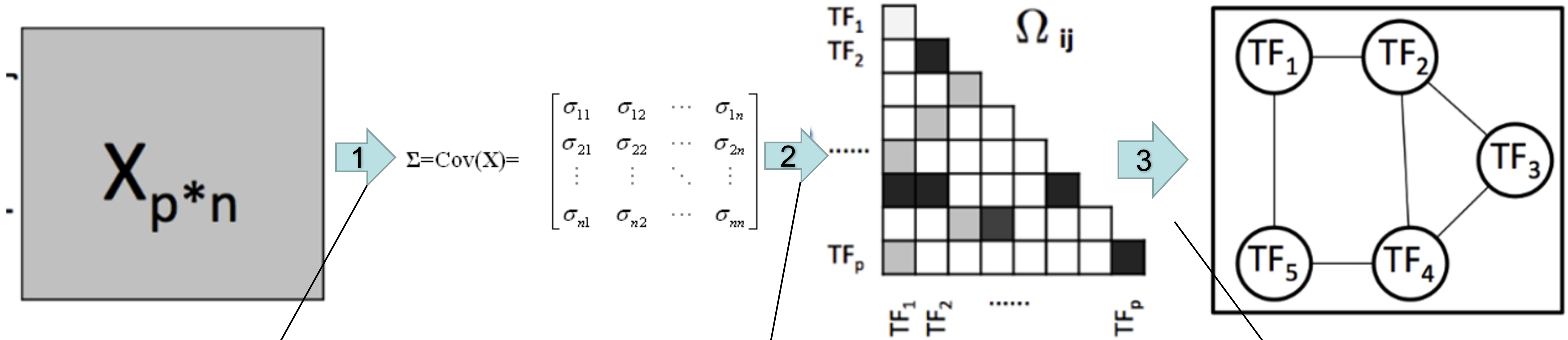
Motivation: Entity Graph – Conditional Independence Graph

Conditional Independent



Conditional Dependent

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



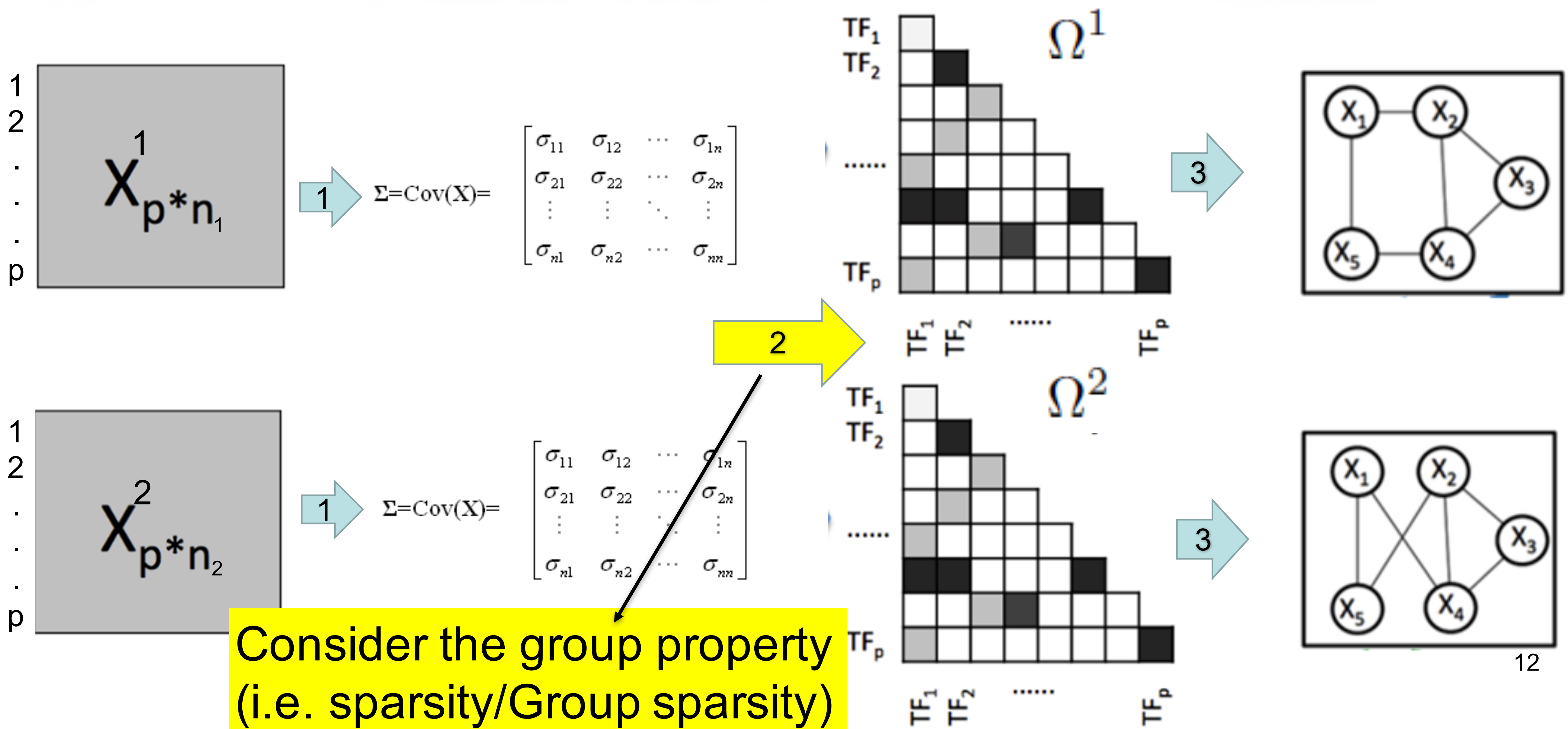
Step1:
Calculate the Covariance matrix
X → Σ

Step2:
Estimate Sparse Inverse of Covariance matrix

$$\boxed{\Sigma^{-1} = \Omega}$$

Step3:
Transfer sparse matrix to Conditional Independence graph

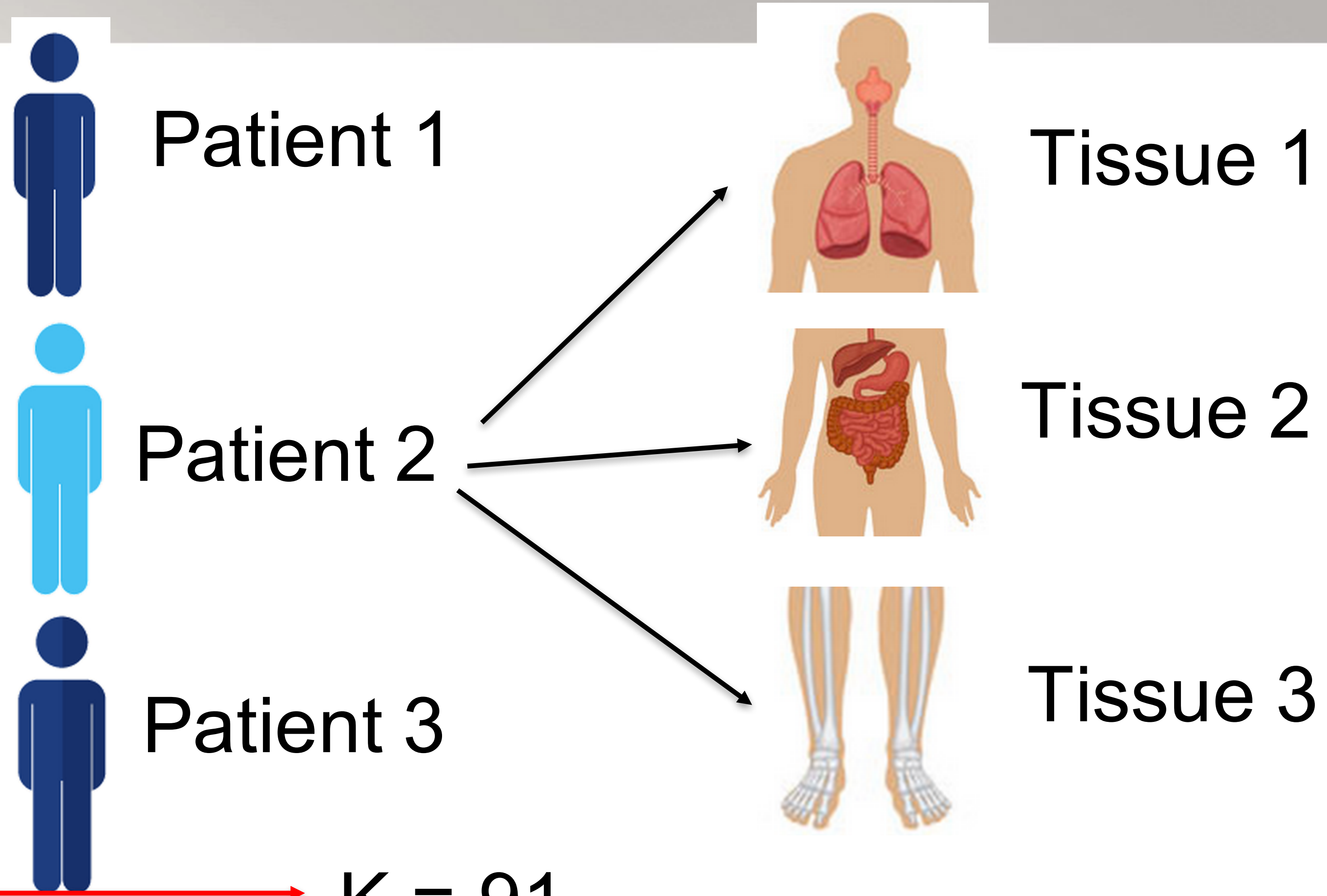
Background Model: Multi-task sGGM



Consider the group property
(i.e. sparsity/Group sparsity)

Motivation: More tasks(K) to be considered

Normal vs Cancer



K = 2 → K = 91

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

Motivation: More Num of features(p) to consider

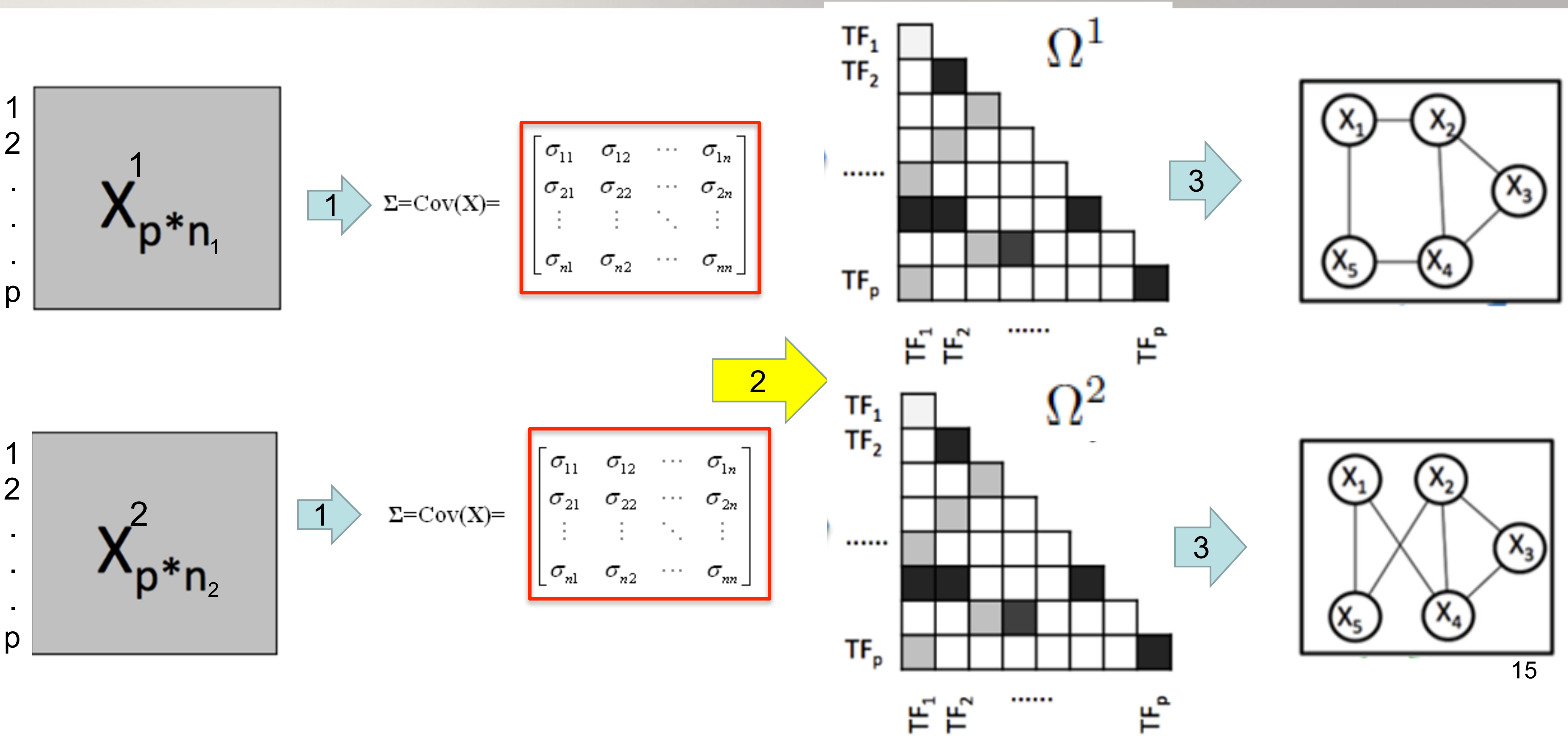
e.g. Yeast gene: 6K

↓

Human gene: 30K

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

Limitation of Previous Methods: Storage



Limitation of Previous Methods : Storage

e.g., calculate the gradient

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

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

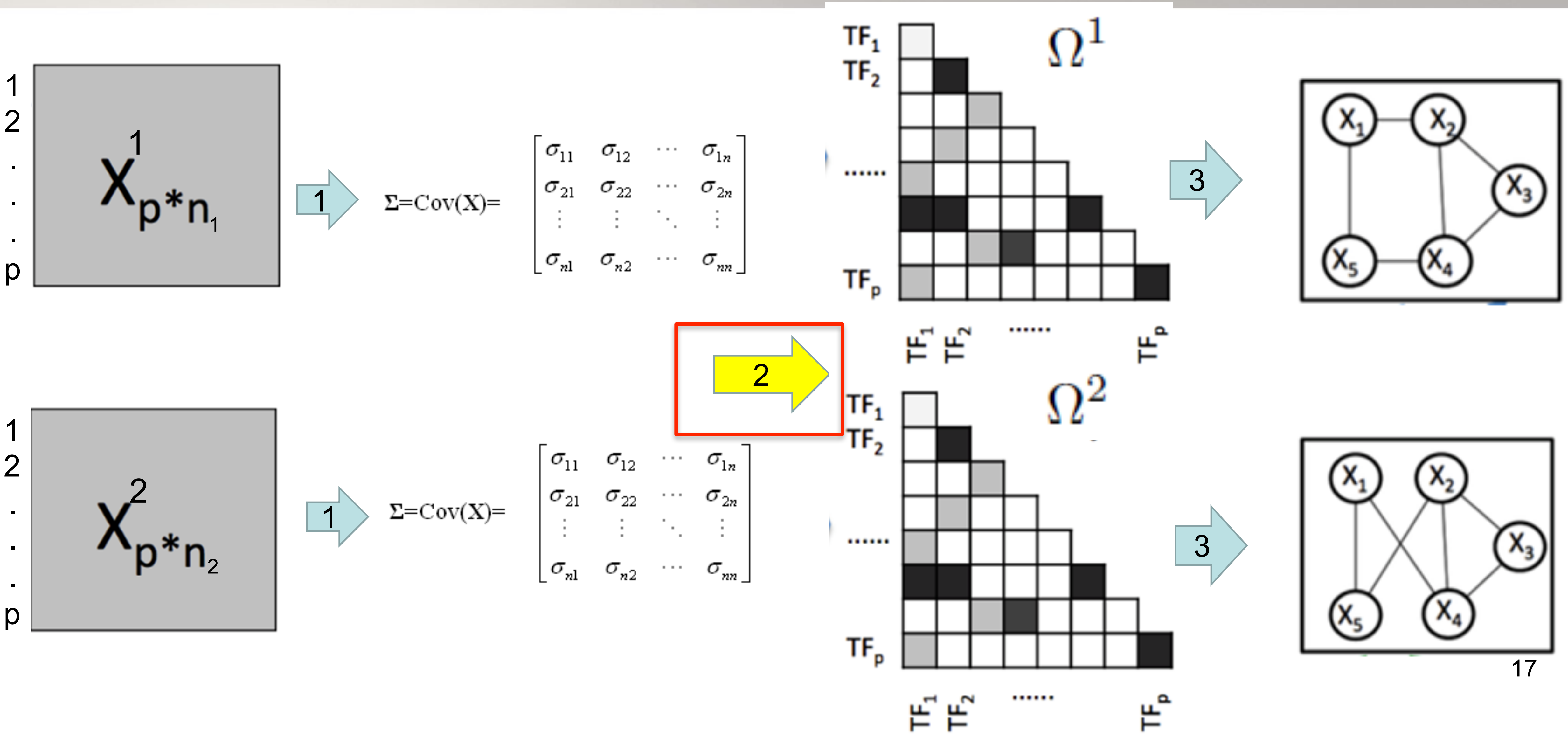
$$\Sigma=\mathrm{Cov}(X)=\boxed{\begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix}}$$

K = 91, p= 30K

$O(Kp^2)$ in memory

Double type: 65 TB

Limitation of Previous Methods: Speed



Limitation of Previous Methods: Speed

Suppose they have same iteration number T

Traditional Optimization Method

K = 91, p= 30K

---- Block Coordinate Descent : $O(K^3p^4)$ / Itera

more than 2 billion years

Improved Optimization:

---- Still needs SVD for each covariance matrix

SVD for the matrices needs $O(Kp^3)$ / Itera $\longrightarrow$ 3.5 days

Roadmap

1. Goal & Background
2. Proposed
3. Evaluation
4. Conclusion

Goal

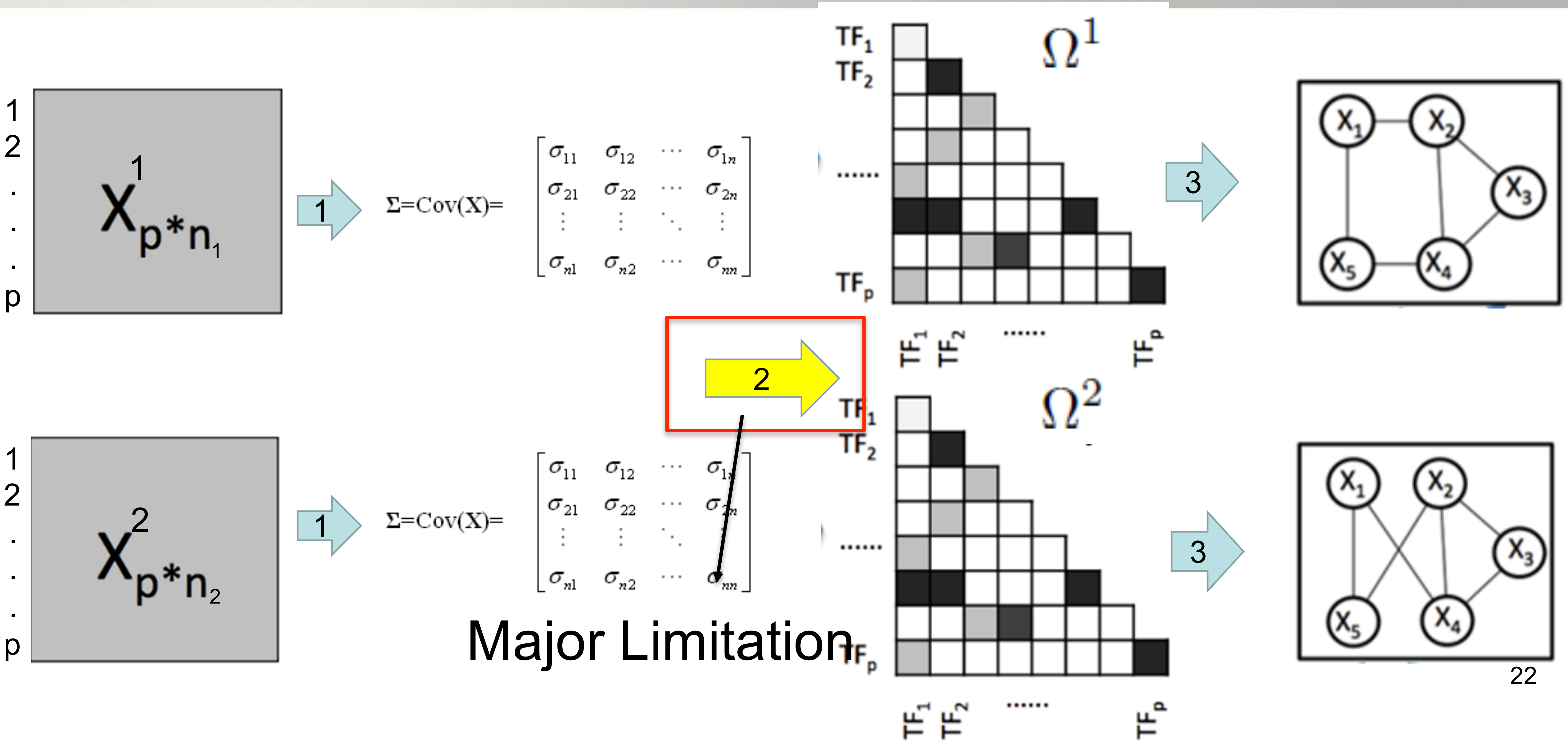
1. Design a fast and scalable joint estimator for multi-task sparse Gaussian Graphical Model

2. Prove the theoretical Bound for our estimator

Roadmap

1. Goal & Background
2. <u>Proposed</u>
3. Evaluation
4. Conclusion

Major Limitation of Previous : Optimization

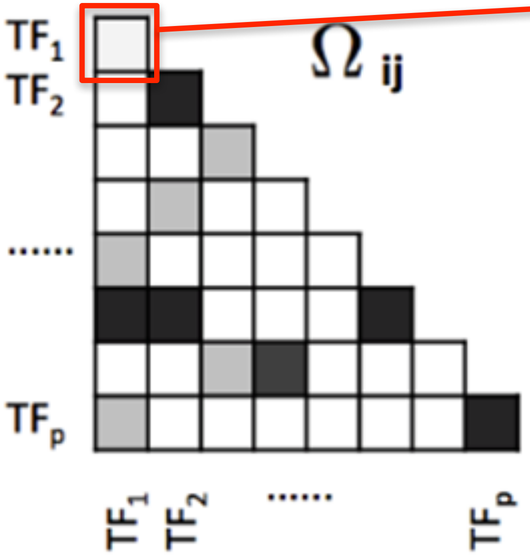


Notation: Entry

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

Ω^1

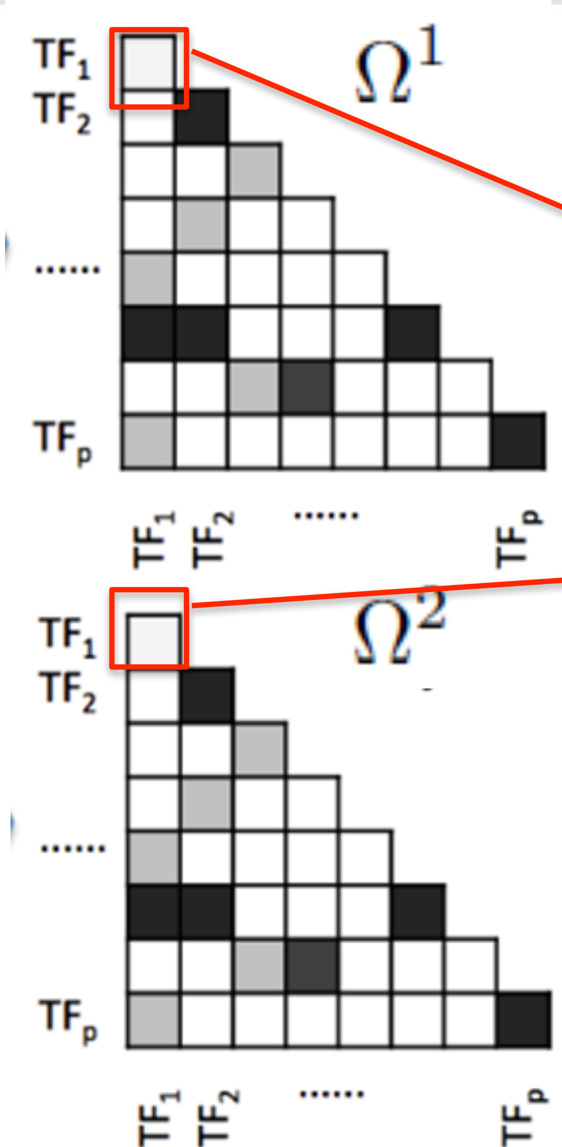
Ω_{ij}



Entry

(Ω_{ij})

Notation: Group Entries among all tasks



Group Entries

$$(\Omega^1_{ij}, \Omega^2_{ij}, \ldots, \Omega^K_{ij})$$

Our Model

- ✓ Traditional Models:

Penalized log-likelihood model $\longrightarrow$ Better optimization method

Some expensive computation is because of model itself!

- ✓ Proposed Model:

New model $\longrightarrow$ Entry-wise(group entry-wise) optimization method

Equivalent Forms of Constrained Optimization

$$\boxed{\begin{aligned}&\min_x f(x)\\&\text{subject to: } g(x) \le c_1\end{aligned}} \Leftrightarrow \boxed{\min_x f(x) + \lambda g(x)} \Leftrightarrow \min_x \tfrac{1}{\lambda} f(x) + g(x) \Leftrightarrow \begin{aligned}&\min_x g(x)\\&\text{subject to: } f(x) \le c_2\end{aligned}$$

$$\underset{\Omega^{(i)} \succ 0}{\operatorname{argmin}} \sum_i \boxed{\left(-L(\Omega^{(i)})\right)}$$

Subject to: 1. sparse
2. group sparse

Initial problem

$$\underset{\Omega^{(i)} \succ 0}{\operatorname{argmin}} \sum_i \boxed{\left(-L(\Omega^{(i)})\right)} + \lambda_1 \sum_i ||\Omega^{(i)}||_1 + \lambda_2 P(\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(K)})$$

Traditional Models

Equivalent Forms of Constrained Optimization

$$\min_x f(x) \quad \text{subject to: } g(x) \le c_1 \iff \boxed{\min_x f(x) + \lambda g(x)} \iff \min_x \tfrac{1}{\lambda} f(x) + g(x) \iff \boxed{\min_x g(x) \quad \text{subject to: } f(x) \le c_2}$$

$$\operatorname*{argmin}_{\Omega^{(i)}>0} \sum_i (-L(\Omega^{(i)}) + \lambda_1 \sum_i ||\Omega^{(i)}||_1 + \lambda_2 P(\Omega^{(1)}, \Omega^{(2)}, \ldots, \Omega^{(K)})$$

Traditional Models

$$\operatorname*{argmin}_{\Omega_{tot}} |\Omega_{tot}|_1 + \epsilon R'(\Omega_{tot})$$

$$s.t. |\Omega_{tot} - inv(T_v(\Sigma_{tot}))|_\infty \le \lambda_n$$

$$\mathcal{R}'^*(\Omega_{tot} - inv(T_v(\Sigma_{tot}))) \le \epsilon\lambda_n$$

Proposed Models

Our Model: FASJEM

Fast and Scalable Joint Estimator for Multiple related sparse Gaussian Graphical Model

$$\underset{\Omega_{tot}}{\operatorname{argmin}} |\Omega_{tot}|_1 + \epsilon R'(\Omega_{tot})$$

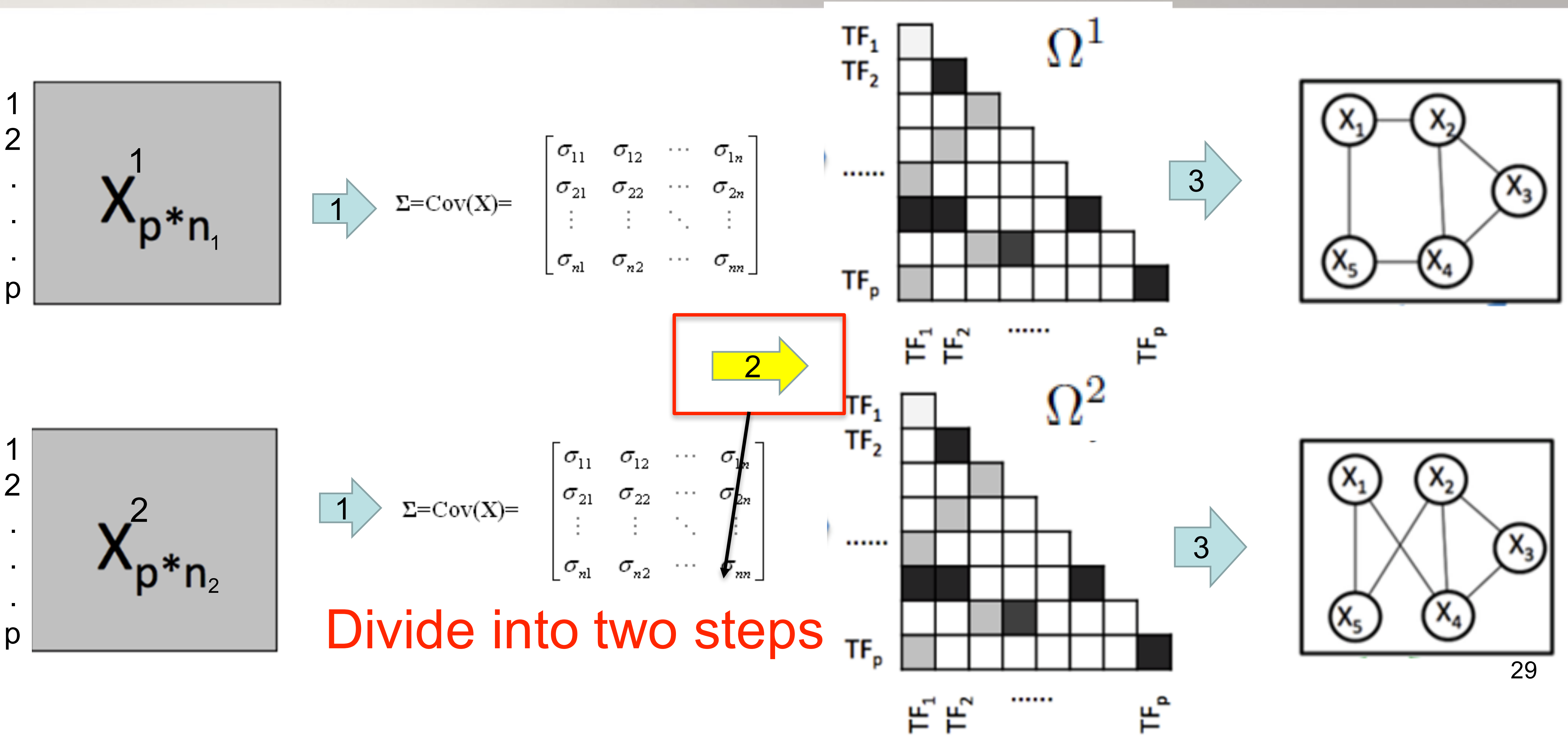
$$s.t. |\Omega_{tot} - inv(T_v(\Sigma_{tot}))|_\infty \leq \lambda_n$$

$$\mathcal{R}'^*(\Omega_{tot} - inv(T_v(\Sigma_{tot}))) \leq \epsilon\lambda_n$$

Here R' is another penalty norm and R'* is the dual norm of R'

$$\Omega_{tot} = (\Omega^{(1)}, \Omega^{(2)}, \ldots, \Omega^{(K)}) \qquad \Sigma_{tot} = (\Sigma^{(1)}, \Sigma^{(2)}, \ldots, \Sigma^{(K)})$$

Optimization: Structure



Divide into two steps

Optimization: Structure

- Step I: Pre-compute and pre-store(not in the memory) approximated backward mapping matrix $\mathcal{B}^*(\Sigma_{tot})$

$$\Sigma_{tot} \longrightarrow \mathcal{B}^*(\Sigma_{tot})$$

- Step II: Use proximity algorithm(entry-wise and group entry-wise) to solve the optimization problem.

$$\mathcal{B}^*(\Sigma_{tot}) \longrightarrow \Omega_{tot}$$

Optimization: Structure

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

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

Pre-compute

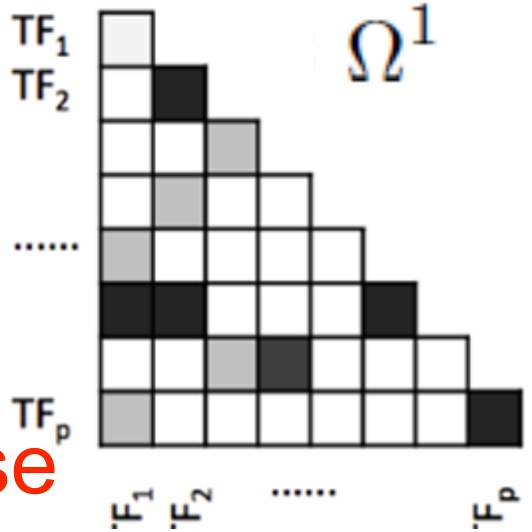
1

Pre-store

entry-wise

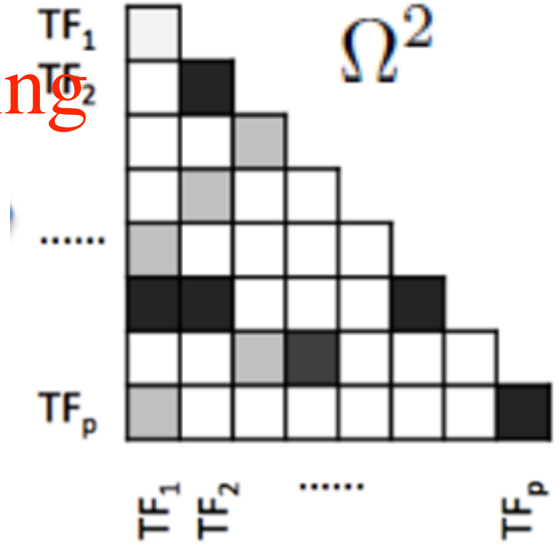
2

soft-thresholding



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

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



Optimization: Step I

Precompute $\mathcal{B}^*(\Sigma_{tot}) = \boxed{inv}(\boxed{T_v(\Sigma_{tot})})$

A matrix inversion + A soft-thresholding operator

Note: this only compute and pre-store(no need to store them in the memory) once. No need to use the whole matrix Σ_{tot} repeatedly.

Here $inv(\Sigma_{tot}) := ({\Sigma^{(1)}}^{-1}, {\Sigma^{(2)}}^{-1}, \ldots, {\Sigma^{(K)}}^{-1})$

Optimization: Step2

$$\underset{\Omega_{tot}}{\operatorname{argmin}} |\Omega_{tot}|_1 + \epsilon \boxed{R'}(\Omega_{tot})$$

$$s.t. |\Omega_{tot} - \mathcal{B}^*(\Sigma_{tot})|_\infty \leq \lambda_n$$

$$\mathcal{R}'^*(\Omega_{tot} - \mathcal{B}^*(\Sigma_{tot})) \leq \epsilon\lambda_n$$

Multiple choices for R'( )

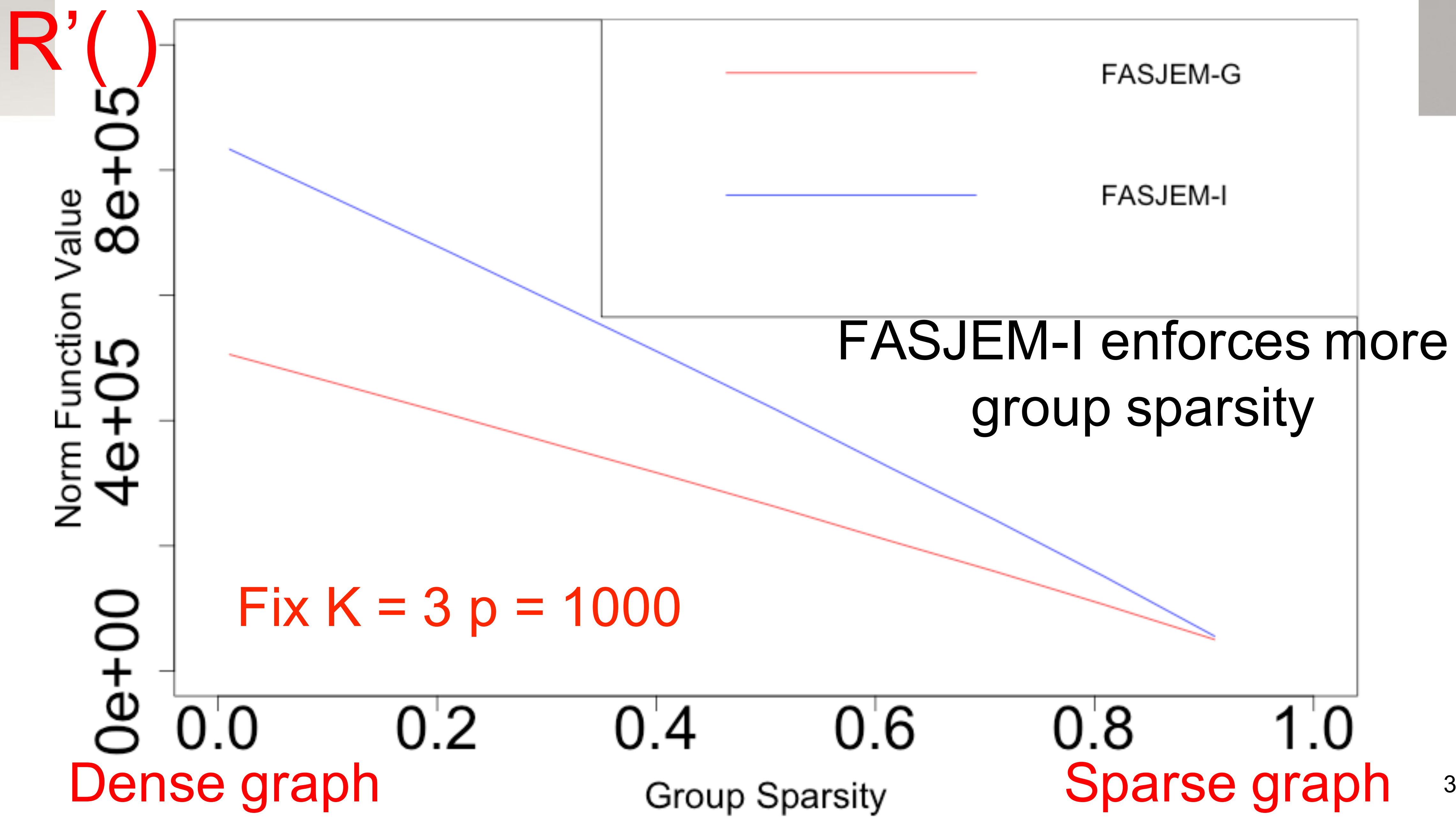
Two Variations

- Case I -- FASJEM-G:

$$\mathcal{R}'(\cdot) = |\cdot|_{\mathcal{G},2}$$

- Case II – FASJEM-I:

$$R'(\cdot) = |\cdot|_{\mathcal{G},\infty}$$


R'( )
FASJEM-G
FASJEM-I
FASJEM-I enforces more group sparsity
Fix K = 3 p = 1000
Norm Function Value
0e+00
4e+05
8e+05
0.0
0.2
0.4
0.6
0.8
1.0
Group Sparsity
Dense graph
Sparse graph

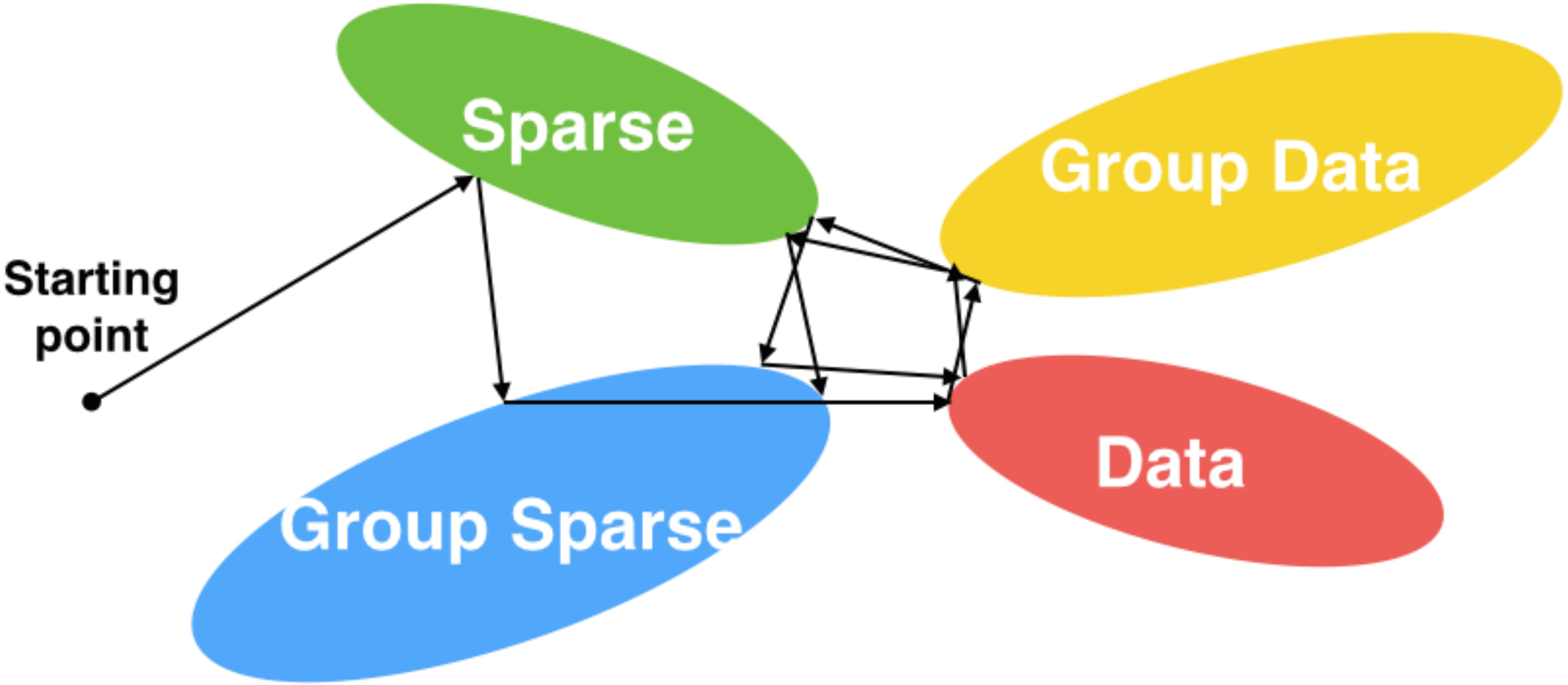
Optimization: Step II for FASJEM-G

We only need to compute the following four soft-thresholding operators for each entry(element) or group entries(element)

We choose FASJEM-G as an example.

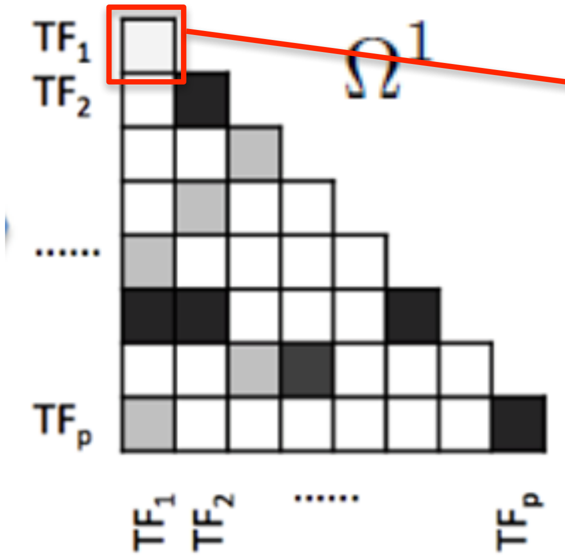
Other second norm is similar to this one.

Optimization Step2: Overall



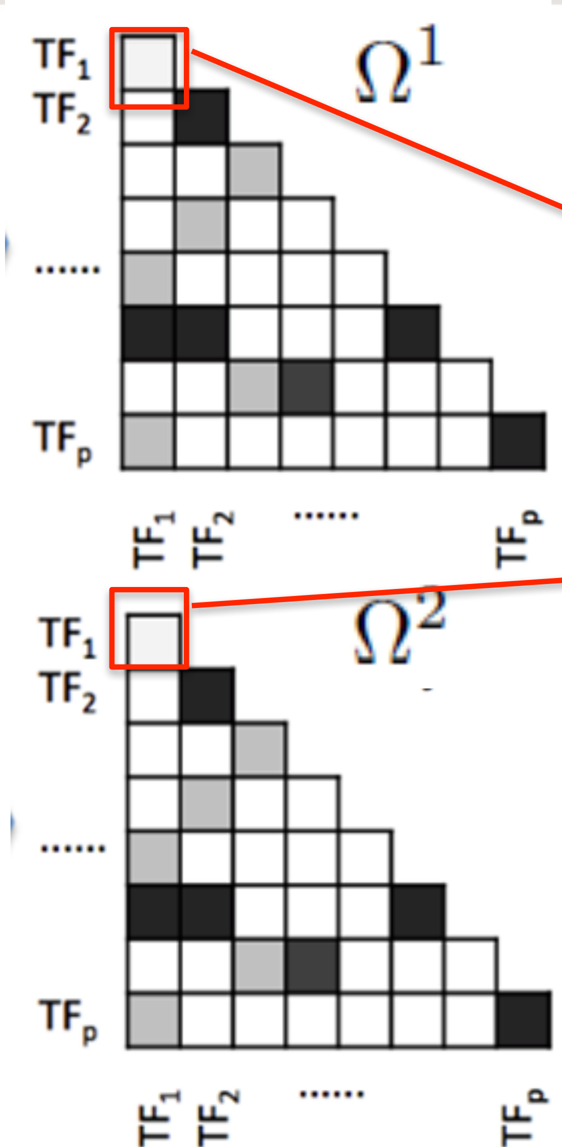
Optimization: Step II(1) - Sparse

In each iteration,



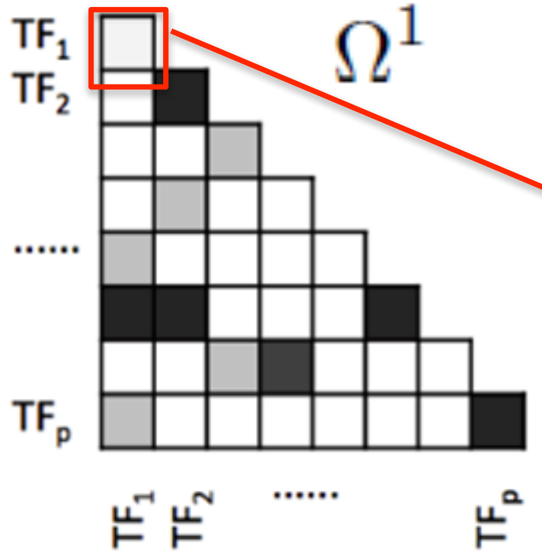
$$(\text{prox}_{\gamma||\cdot||_1}(x))_i = \begin{cases} x_i - \gamma,\, x_i > \gamma \\ 0,\, |x_i| \le \gamma \\ x_i + \gamma,\, x_i < -\gamma \end{cases}$$

Optimization: Step II(2) – Group sparse



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

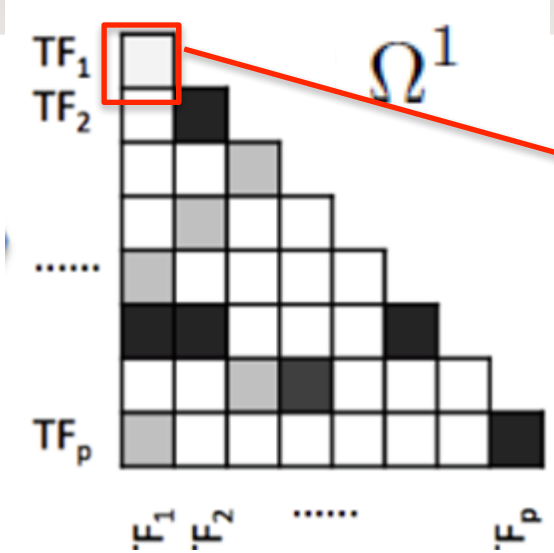
Optimization: Step II (3) – Data



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

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

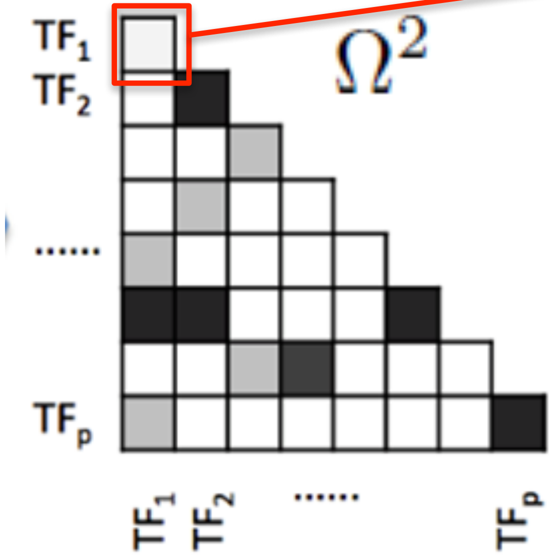
Optimization: Step II (4) – Group Data



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

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

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



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

Advantage of Optimization: Space

Suppose they have same iteration number T

K = 91, p= 30K

$O(Kp^2)$ space to store / Itera ⇨ Pre-store, only $O(K)$ / Itera

Double type: 65 TB ⇨ Double type: 728B < 1KB

Advantage of Optimization: Time

Suppose they have same iteration number T

K = 91, p= 30K

Previous Multi-sGGM (SVD) needs $O(Kp^3)$/ Itera

Totally entry-wise, $O(Kp^2)$/ Itera also can be paralled

300000 times faster

3.5 days ⟷ 1 second

Roadmap

1. Goal & Background
2. Proposed
3. <u>Evaluation</u>
4. Conclusion

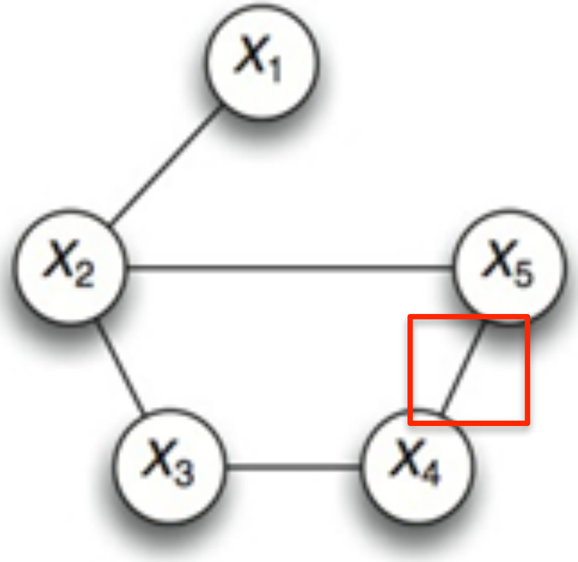
My Work: Evaluation

1. Simulation test
    - random graph models

2. Real world datasets

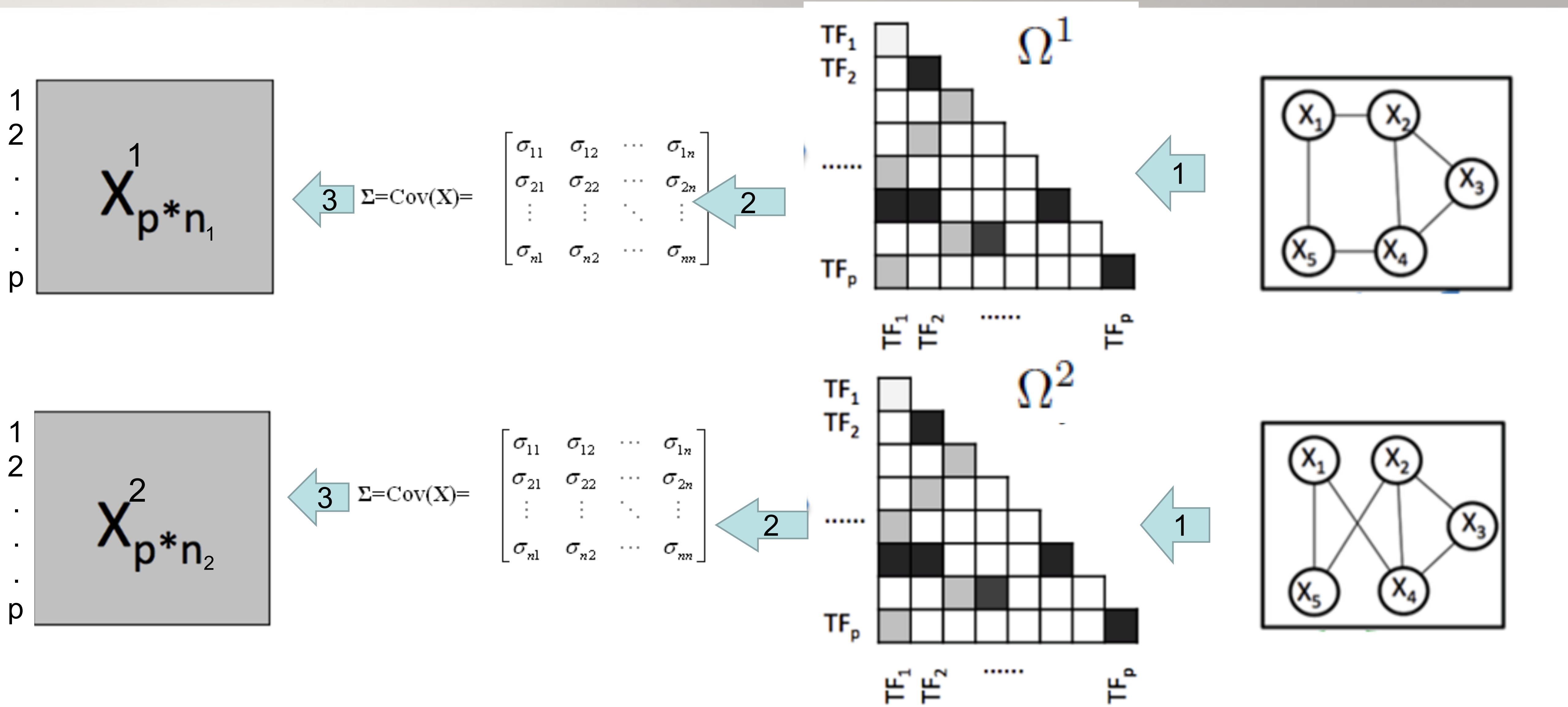
3. Theoretical Performance
    - e.g., Convergence rate

EXP I – Generating Simulation Data



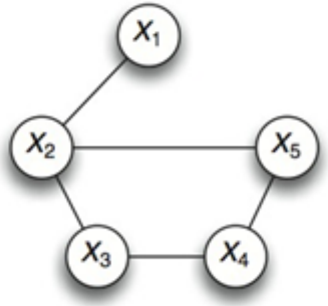
Generate edges randomly by Bernoulli distribution with probability q

EXP I – Generating Simulation Data



EXP I – Generating Simulation Data

Step1:



Non-edge → Zero entry

Edge → Nonzero entry

(b) Inverse Covariance Matrix

$$\begin{pmatrix} 1 & 0.2 & 0 & 0 & 0 \\ 0.2 & 1 & 0.2 & 0 & 0.2 \\ 0 & 0.2 & 1 & 0.2 & 0 \\ 0 & 0 & 0.2 & 1 & 0.2 \\ 0 & 0.2 & 0 & 0.2 & 1 \end{pmatrix}$$

EXP I – Generating Simulation Data

Step2:

$$\begin{pmatrix} 1 & 0.2 & 0 & 0 & 0 \\ 0.2 & 1 & 0.2 & 0 & 0.2 \\ 0 & 0.2 & 1 & 0.2 & 0 \\ 0 & 0 & 0.2 & 1 & 0.2 \\ 0 & 0.2 & 0 & 0.2 & 1 \end{pmatrix} \xrightarrow{\text{Inverse}} \Sigma = \begin{pmatrix} 1.05 & -0.23 & 0.05 & -0.02 & 0.05 \\ -0.23 & 1.45 & -0.25 & 0.10 & -0.25 \\ 0.05 & -0.25 & 1.10 & -0.24 & 0.10 \\ -0.02 & 0.10 & -0.24 & 1.10 & -0.24 \\ 0.05 & -0.25 & 0.10 & -0.24 & 1.10 \end{pmatrix}$$

EXP I – Generating Simulation Data

Step3:

Suppose $X \sim N(\mu, \Sigma)$ and X is a p-dimensional vector

Use MCMC to simulate data set based on Covariance matrix

Evaluation: metric

1. ROC curve varying different tuning parameter, compare AUC(area under curve)

2. computation time
   - fix K varying p
   - fix p varying K

3. memory
   - how large p and K will cause the programme to terminate

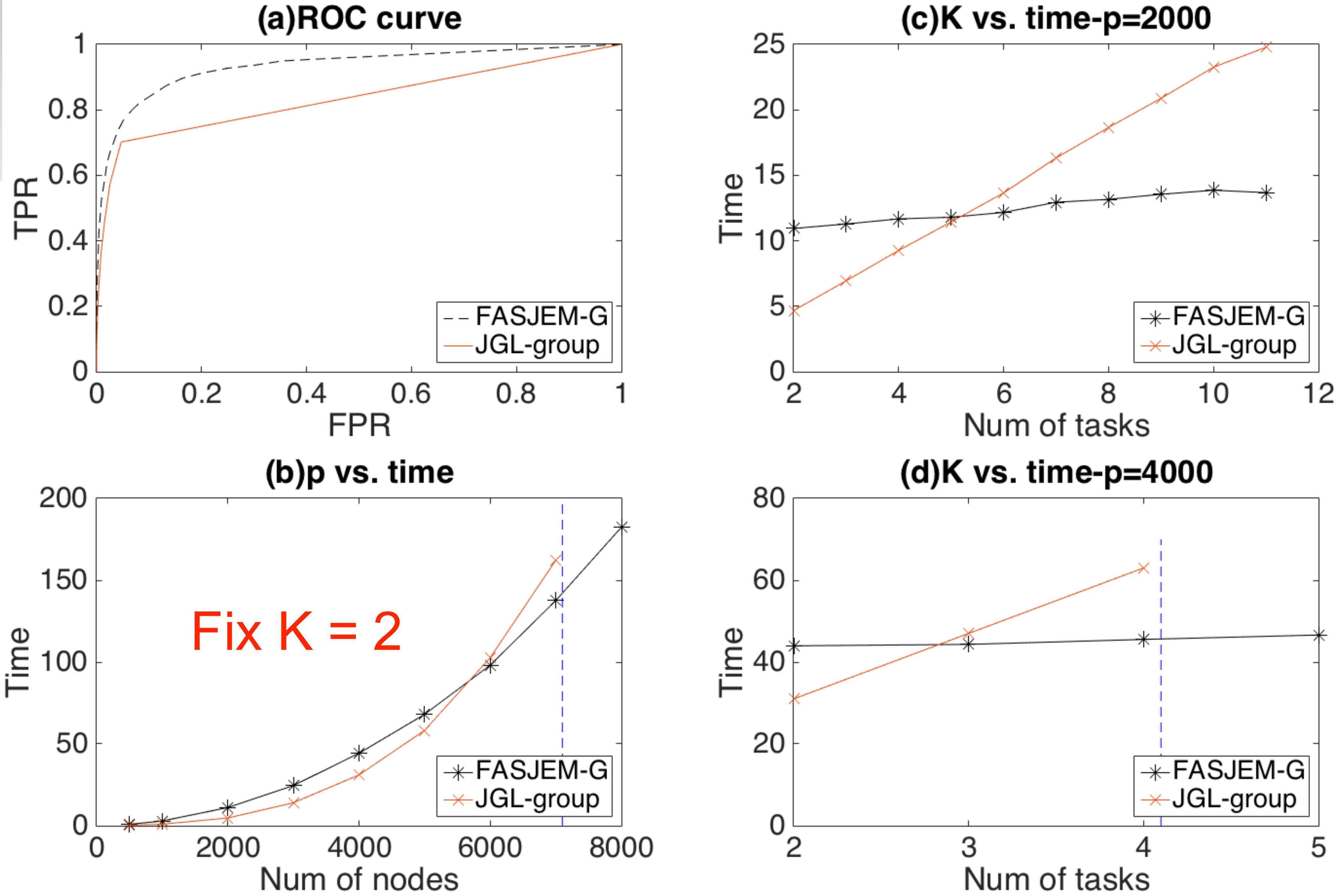
Evaluation : Experimental Setting

- simulation test datasets:
  - Model : random sparse graph model
  - Case I -- FASJEM-G:

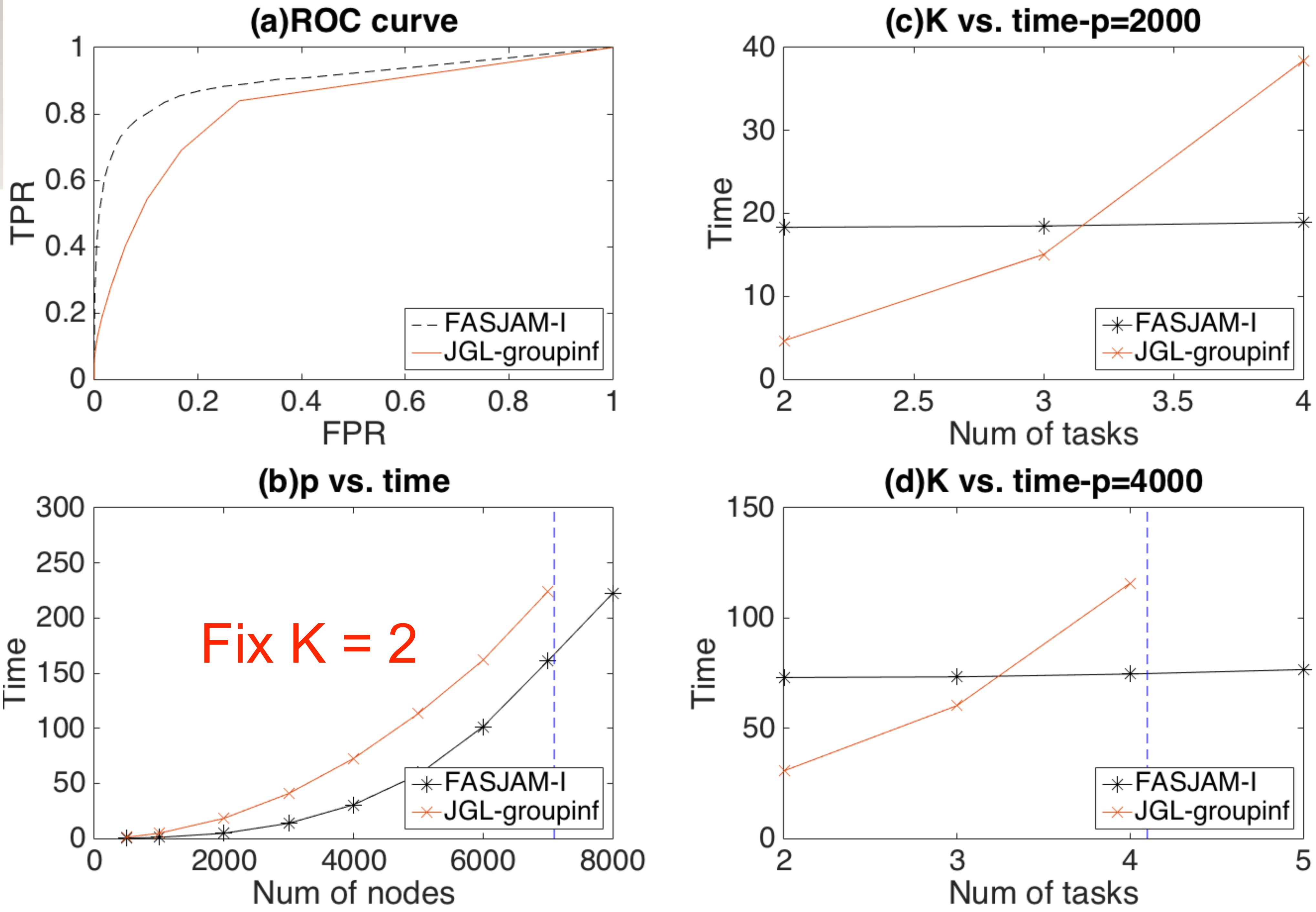
$$\mathcal{R}'(\cdot) = |\cdot|_{\mathcal{G},2}$$

  - Case II – FASJEM-I:

$$R'(\cdot) = |\cdot|_{\mathcal{G},\infty}$$



(a)ROC curve
TPR
FPR
FASJEM-G
JGL-group
(c)K vs. time-p=2000
Time
Num of tasks
FASJEM-G
JGL-group
(b)p vs. time
Time
Num of nodes
Fix K = 2
FASJEM-G
JGL-group
(d)K vs. time-p=4000
Time
Num of tasks
FASJEM-G
JGL-group



(a)ROC curve
TPR
FPR
FASJAM-I
JGL-groupinf
(c)K vs. time-p=2000
Time
Num of tasks
FASJAM-I
JGL-groupinf
(b)p vs. time
Time
Num of nodes
Fix K = 2
FASJAM-I
JGL-groupinf
(d)K vs. time-p=4000
Time
Num of tasks
FASJAM-I
JGL-groupinf

Evaluation-Experiment Result I

- Our models obtain the best accuracy result.
- Our models are faster than the baseline methods when $p>6000$&$K=2$ or $p = 2000$&$K > 6$
- Our models still work when $p>8000$&$K=2$ or $p=4000$&$K>4$
  - On 8GB memory desktop

Evaluation-Experiment Result II

Two existing
gene interactions databases

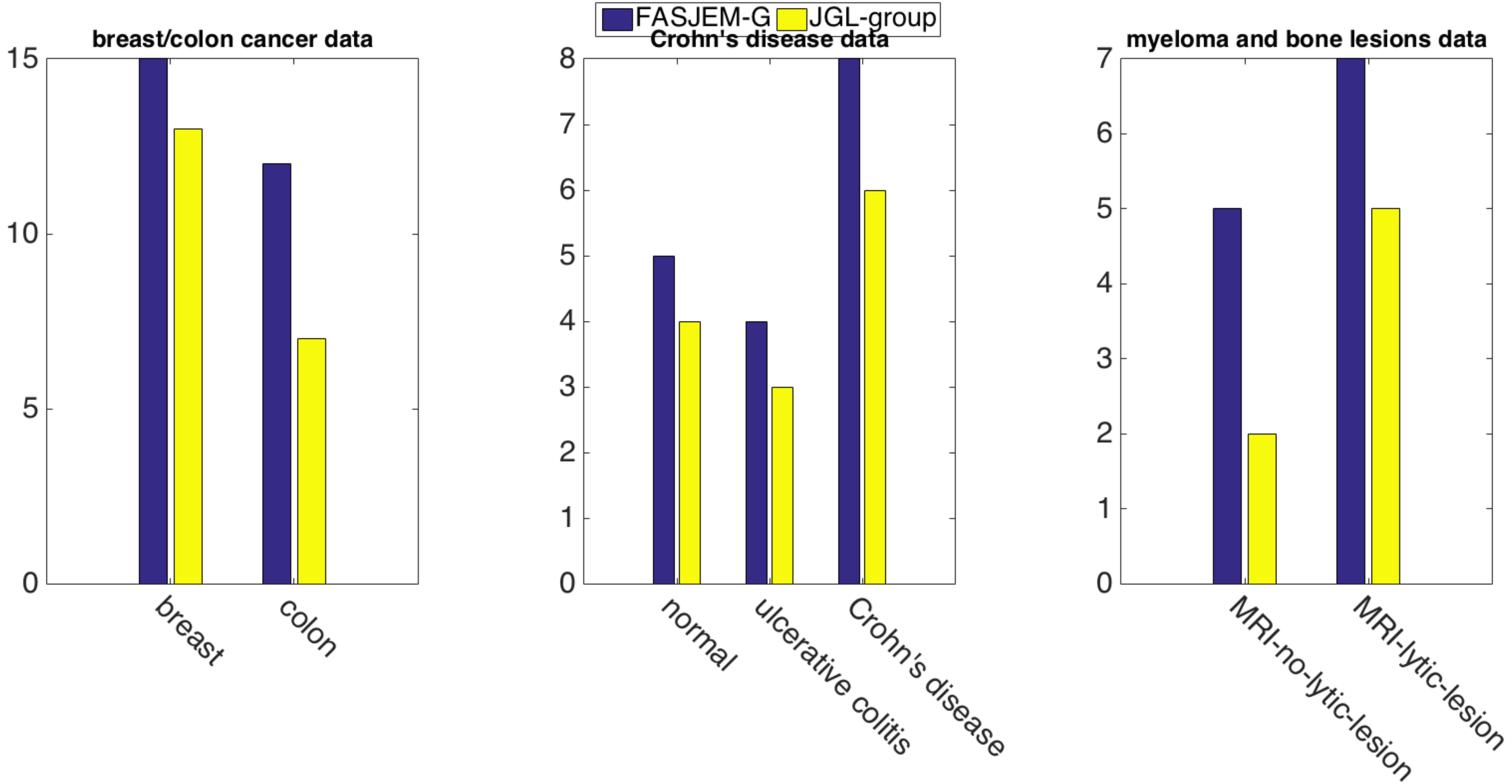
Three real world datasets

implement
different
methods

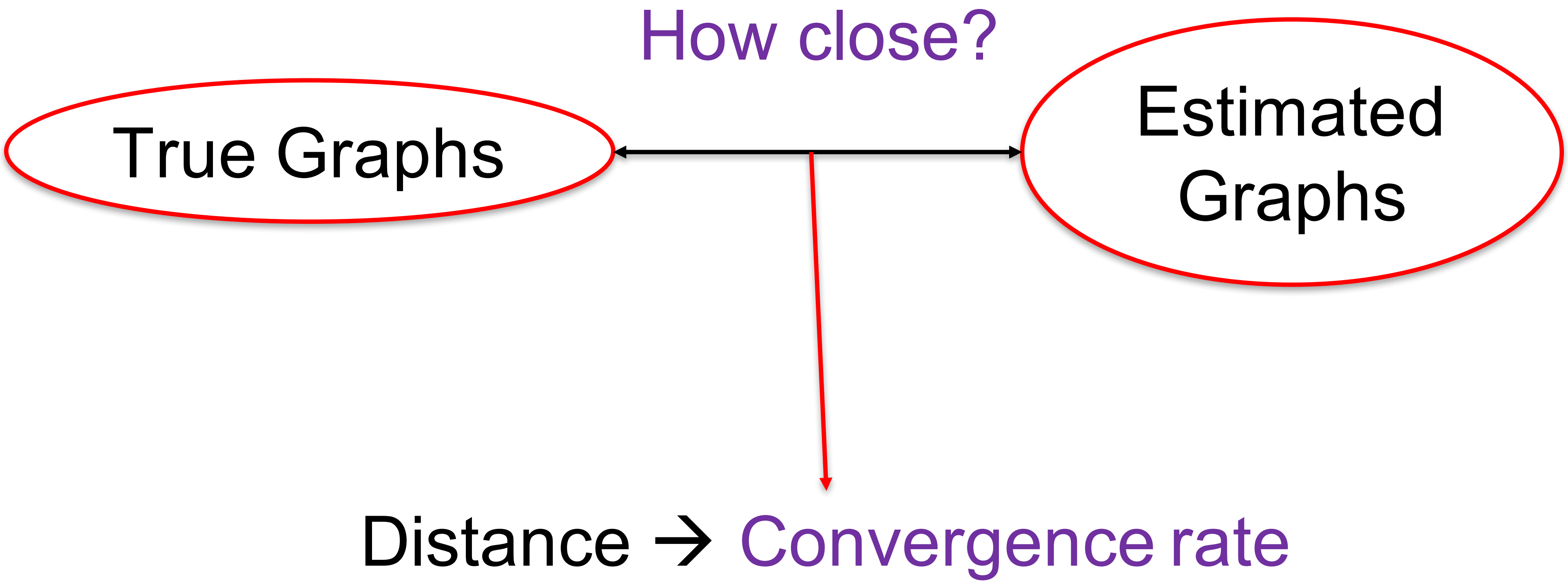
Results ⟺ Interactions

matching

Evaluation-Experiment Result II



Evaluation-Theoretical Analysis



Theoretical Analysis – Best convergence rate

- General case > some experiments

- The best convergence rate:

$$|\widehat{\mu} - \mu^*|_F \leq 8 \boxed{\max\{M_1\sqrt{\frac{k_1 \log Kp}{n_{tot}}}, M_2\sqrt{\frac{k_2 p \log Kp}{n_{tot}}}\}}$$

Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decom- posable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009

Theoretical Analysis – Best convergence rate

- General cases' conclusion

- The best convergence rate:

$$\|\widehat{\mu} - \mu^*\|_F \leq 8 \max\{M_1 \sqrt{\frac{k_1 \log Kp}{n_{tot}}}, M_2 \sqrt{\frac{k_2 p \log Kp}{n_{tot}}}\}$$

- We prove it!

Theoretical Analysis – When compared to single task case

- Single task:

$$|\widehat{\mu} - \mu^*|_F \le O(\sqrt{\frac{\log p}{n}})$$

- Our case:

$$|\widehat{\mu} - \mu^*|_F \le 8 \max\{M_1\sqrt{\frac{k_1 \log Kp}{n_{tot}}}, M_2\sqrt{\frac{k_2 p \log Kp}{n_{tot}}}\}$$

Theoretical Analysis – Multi-task helps !

Suppose $n_i = n_1$

$$\frac{\log Kp}{Kn_1} \leq \frac{\log p}{n_1}$$

$K = 91$, $p = 30K$, $n_1 = 1K$

Our case 0.0001 << 0.01 Single case

Multi-task —> n increase —> closer distance

Conclusion

- We design a novel algorithm to solve the Multi-task sGGM
- We have the best simulation test result in
  - Accuracy
  - Time
  - Memory
- Our method achieve the best convergence rate

Thank You!

http://jointggm.org

R package: fasjem

```
install.packages("fasjem")
library(fasjem)
demo(fasjem)
```

Background: Multi-task sGGM to derive Conditional Independence Graph from data

Step1：

Suppose $X \sim N(\mu, \Sigma)$ and X is a p-dimensional vector

$$\widehat{\Sigma} = (X - \bar{X})^T (X - \bar{X})$$

Background: Multi-task sGGM to derive Conditional Independence Graph from data

Step2: We solve the following optimization problem:

$$\min_{\Omega^k > 0} \sum_k \left(-\boxed{L(\Omega^k)} + \rho_1 \boxed{|\Omega^k|_1}\right) + \rho_2 \boxed{P(\Omega^1, \Omega^2, \ldots, \Omega^K)}$$

where $k = 1, 2, \ldots, K$.

Log likelihood

L1 penalty

Group penalty

Background: Multi-task sGGM to derive Conditional Independence Graph from data

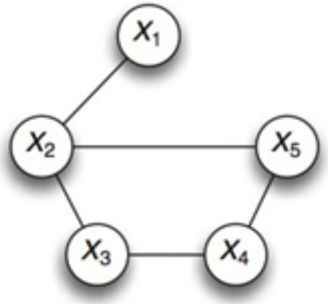
Step3:

(b) Inverse Covariance Matrix

$$\begin{pmatrix} 1 & 0.2 & 0 & 0 & 0 \\ 0.2 & 1 & 0.2 & 0 & 0.2 \\ 0 & 0.2 & 1 & 0.2 & 0 \\ 0 & 0 & 0.2 & 1 & 0.2 \\ 0 & 0.2 & 0 & 0.2 & 1 \end{pmatrix}$$

Zero entry → Non-edge

Nonzero entry → Edge



Background: Dual Norm

$$\mathcal{R}^*(v) := \sup_{u \in \mathbb{R}^p \setminus \{0\}} \frac{< u, v >}{\mathcal{R}(u)} = \sup_{\mathcal{R}(u) \leq 1} < u, v >$$

Experiment: Real world datasets

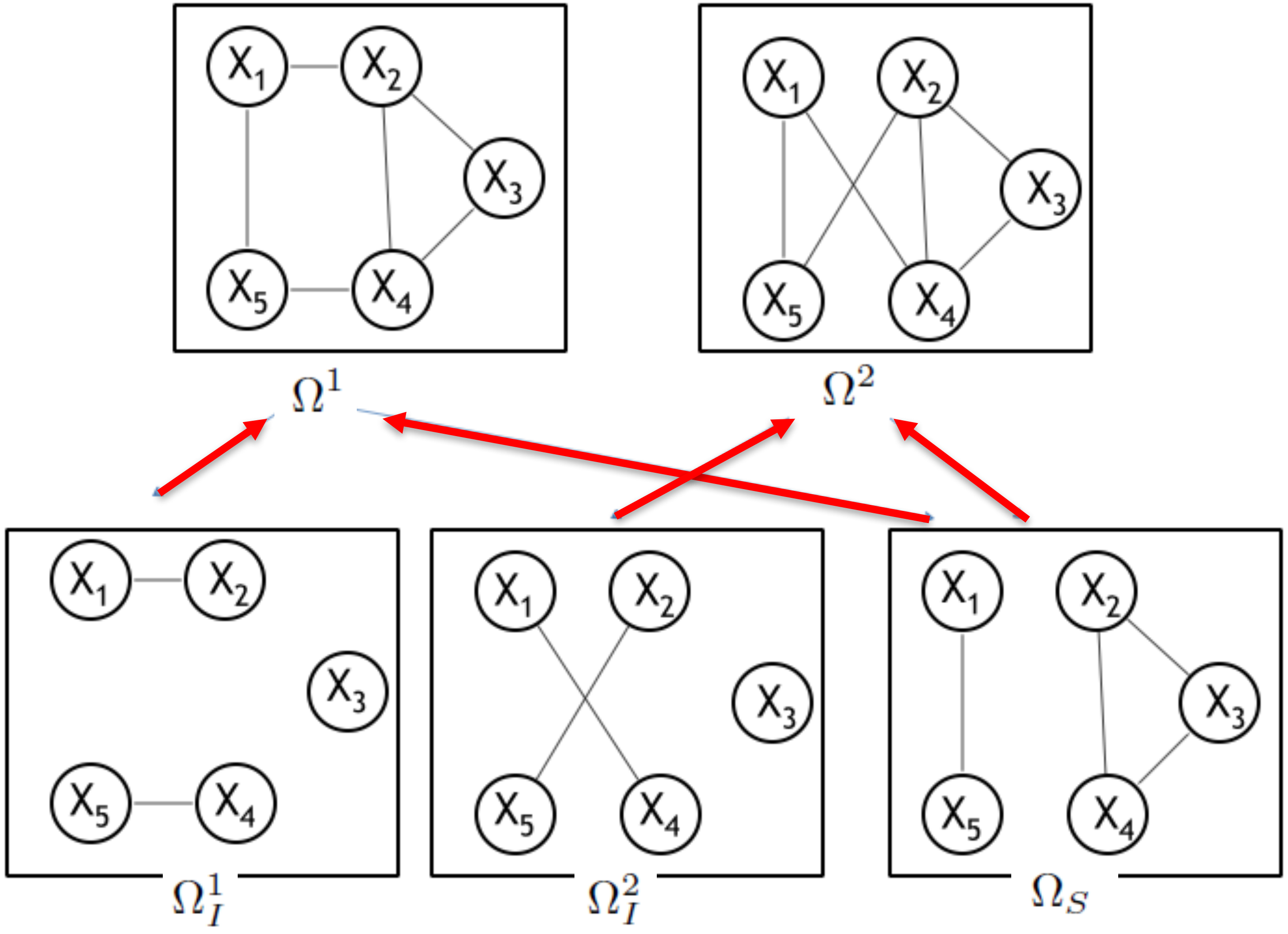
(1) The breast/colon cancer data (with 2 cell type and 104 samples, each of which has 22283 features);

(2) Chrohn's disease data (with 3 cell type and 127 samples, each of which has 22283 features)

(3) The myeloma and bone lesions data set (with 2 cell type and 173 samples, each of which has 12625 features)

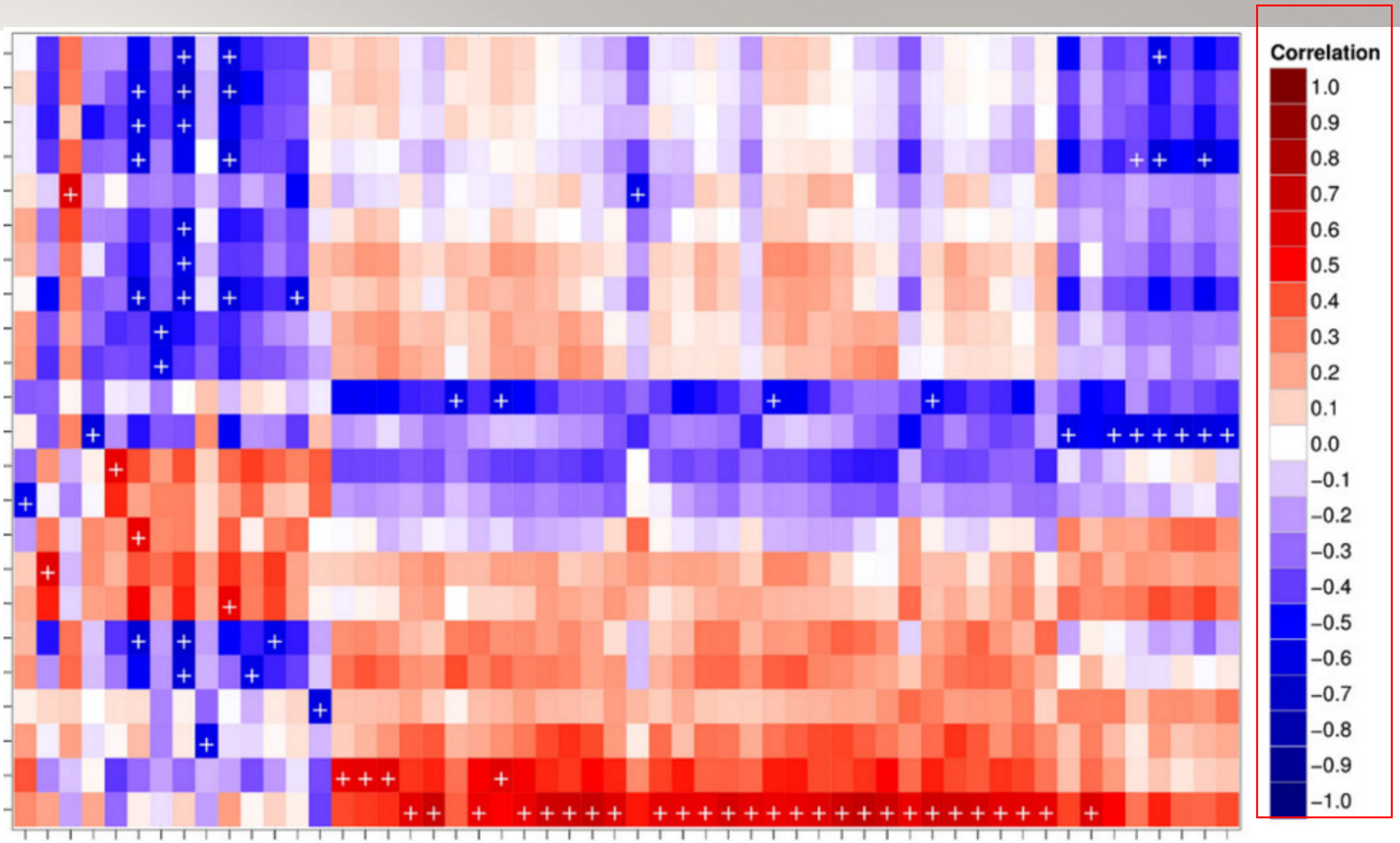
We select top 500 features based variable variance for all three datasets.

Experiment: Synthesizing Data With Random Graph Model



Motivation: Correlation is not enough

Correlation Heatmap:



Motivation: Correlation is not enough

Example:

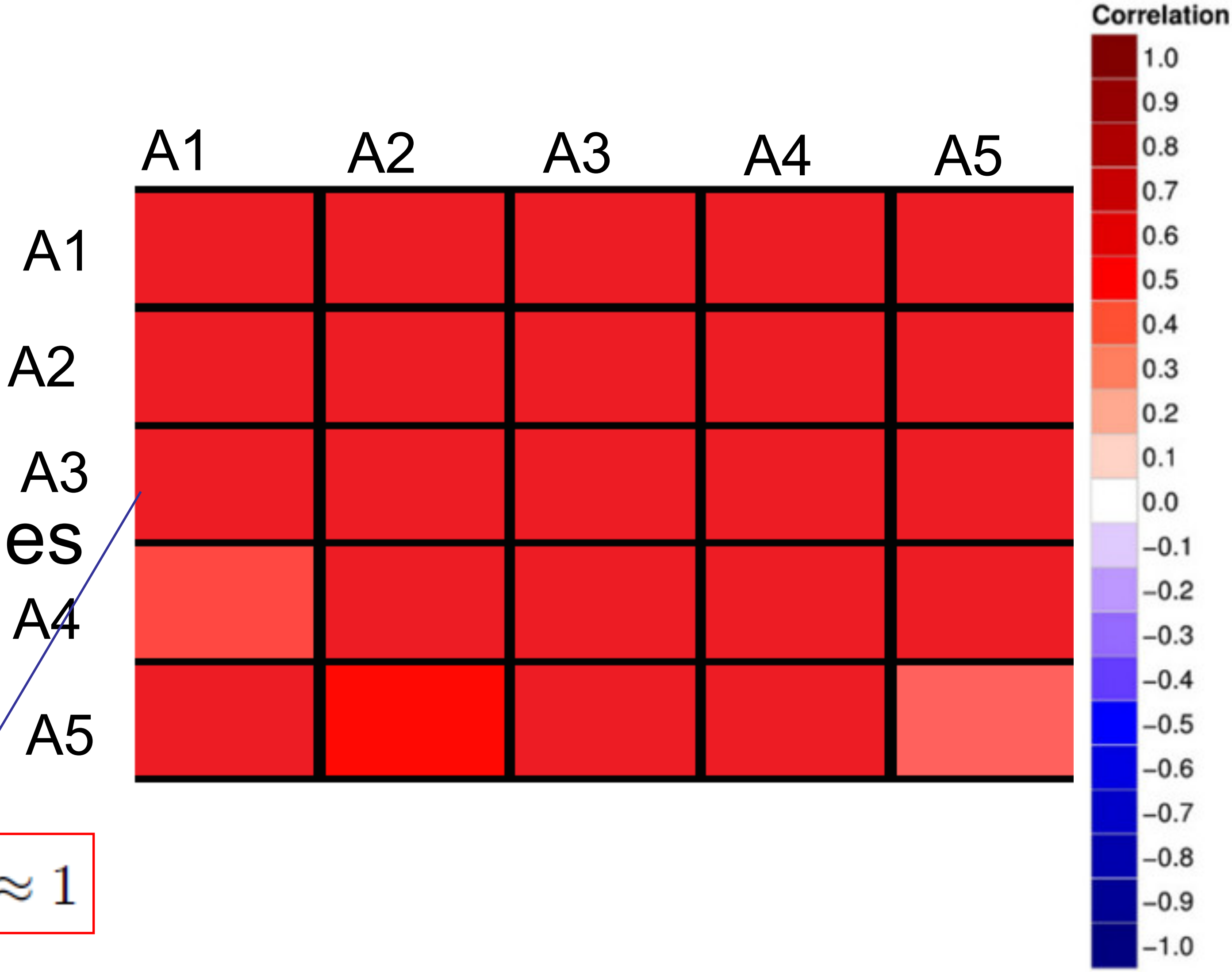
If X is a random variable follows $N(0,1)$, let $Y = X^2$.

Then $Cor(X,Y) = 0$, but X and Y have dependent relationship.

Motivation: Correlation is not enough

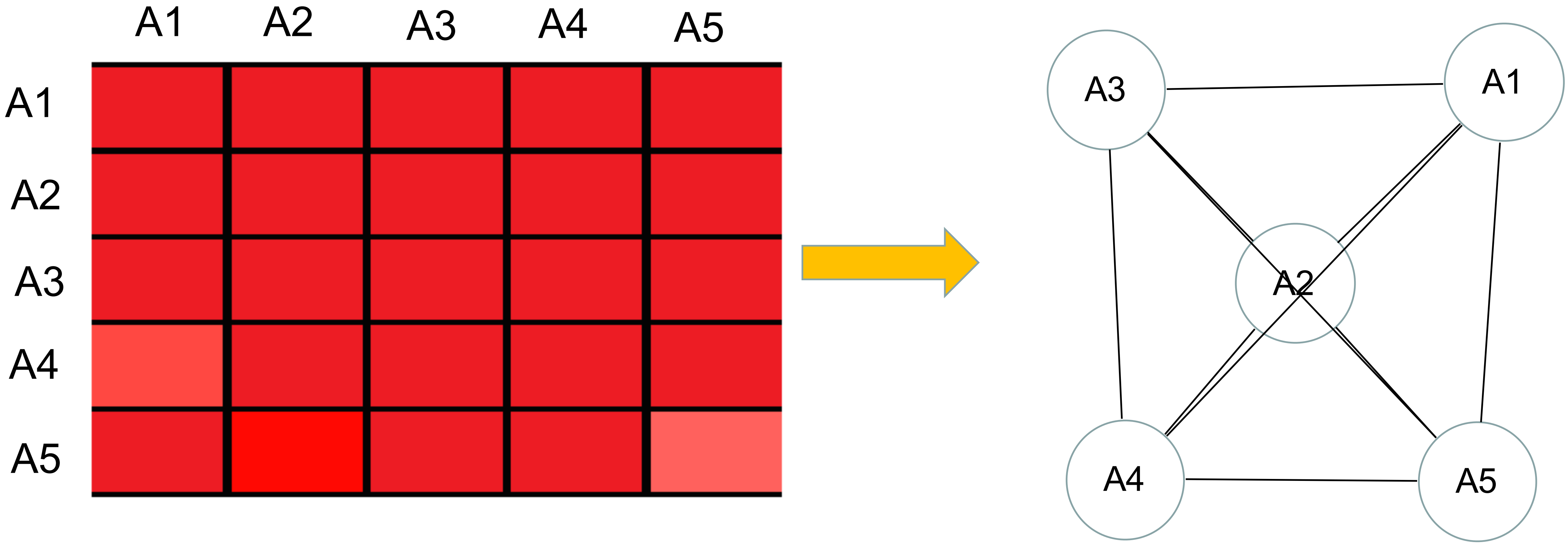
Another Example:

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

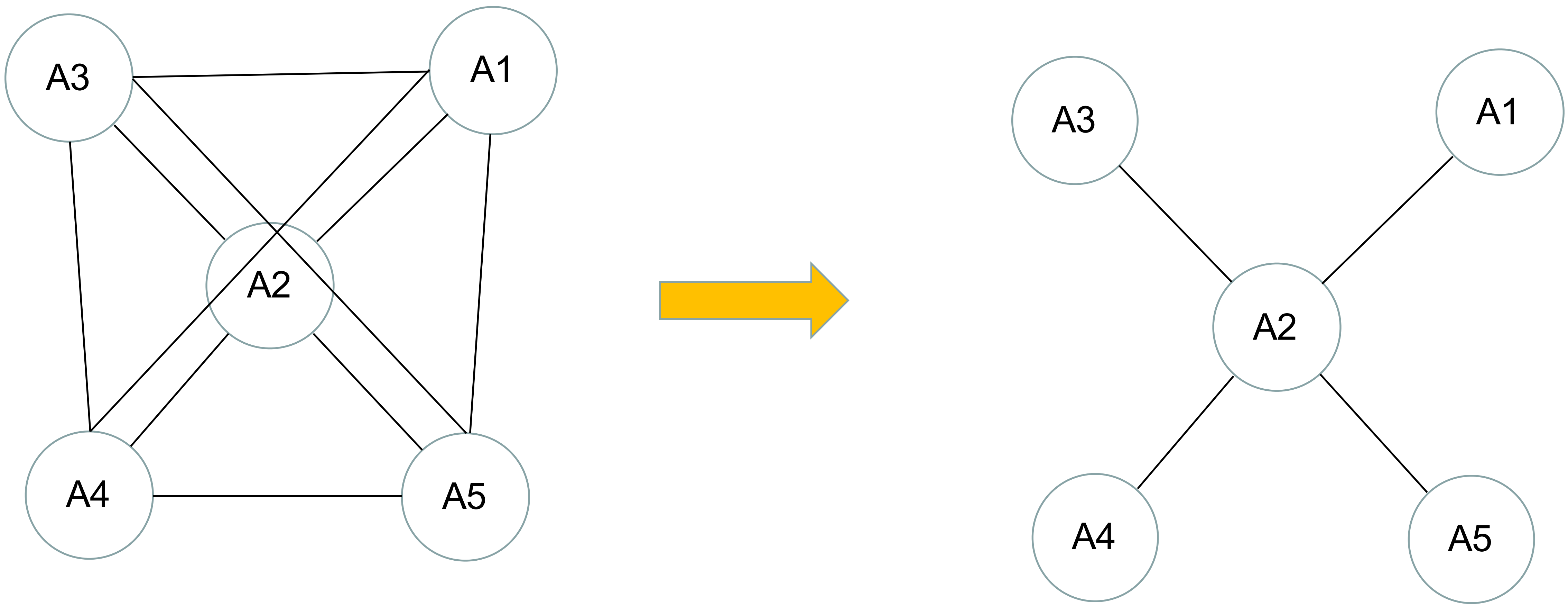


$$\boxed{Cor(A_1, A_3) \approx 1}$$

Motivation: Correlation is not enough



Motivation: Conditional independence is better



Motivation: Conditional independence is better

A1: Children are drown

A2: Weather is hot

A3: High sale of ice cream

A4: Wear less amount of clothes

A5: High Electricity Consumption