Gaussian Graphical Models.

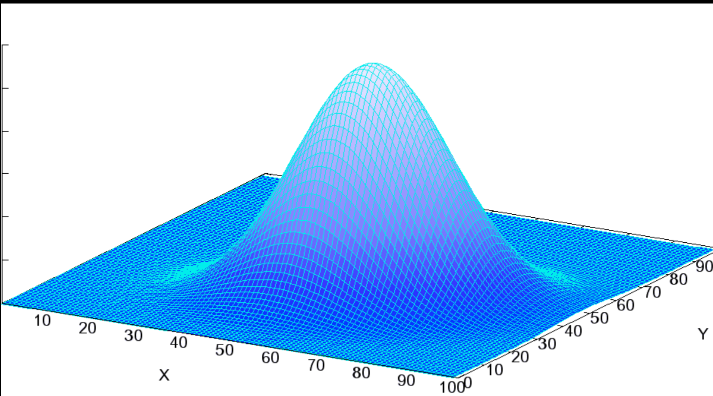
SUMMARY:

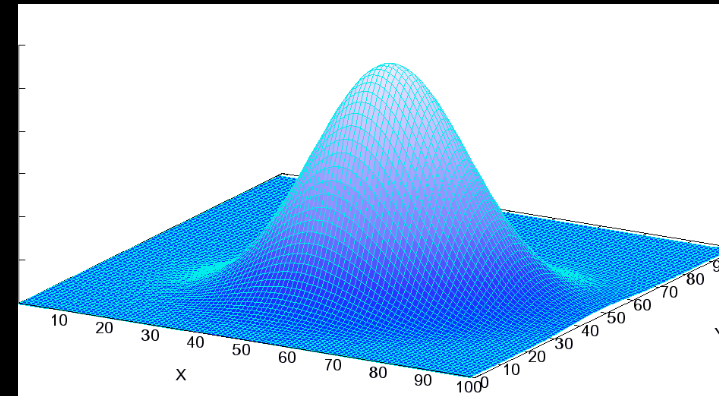
Can learn Attractive and Walk-Summable GGMs with $O(d^2 \log p / \kappa^6)$ samples and quadratic run-time.

d - degree (max)
p - total number of parameters

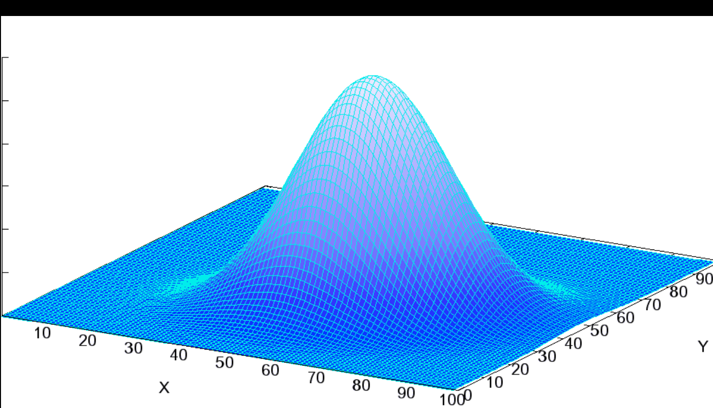
Open: Can any sparse GGM be learnt with $O\left(\frac{d \log p}{\kappa^2}\right)$ samples and a fixed polynomial run-time?

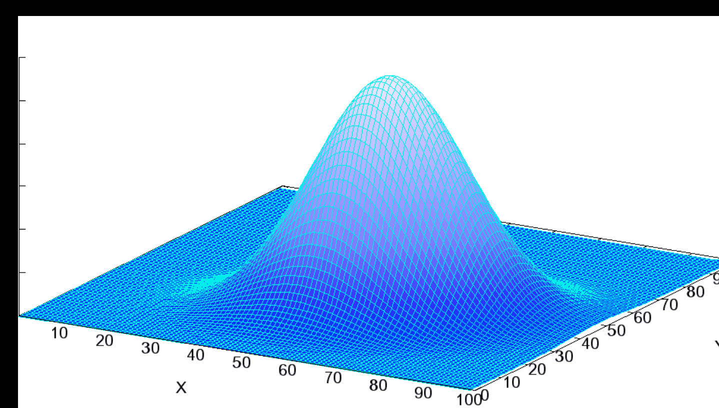
$$\kappa(\Theta) = \min_{i,j:\Theta_{ij}\neq 0} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$$



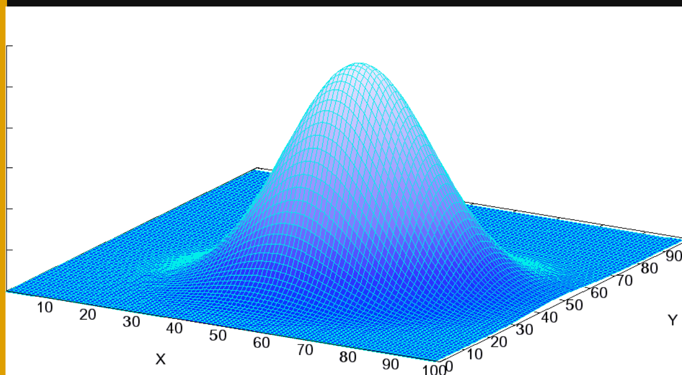


Learning Gaussian Graphical Models w/o condition number bounds





GGMs



Motivation

Algorithm

While $|S| < t$:

Add $\arg\min_{j} Var(X_i \mid X_{S \cup j})$.

Special Models

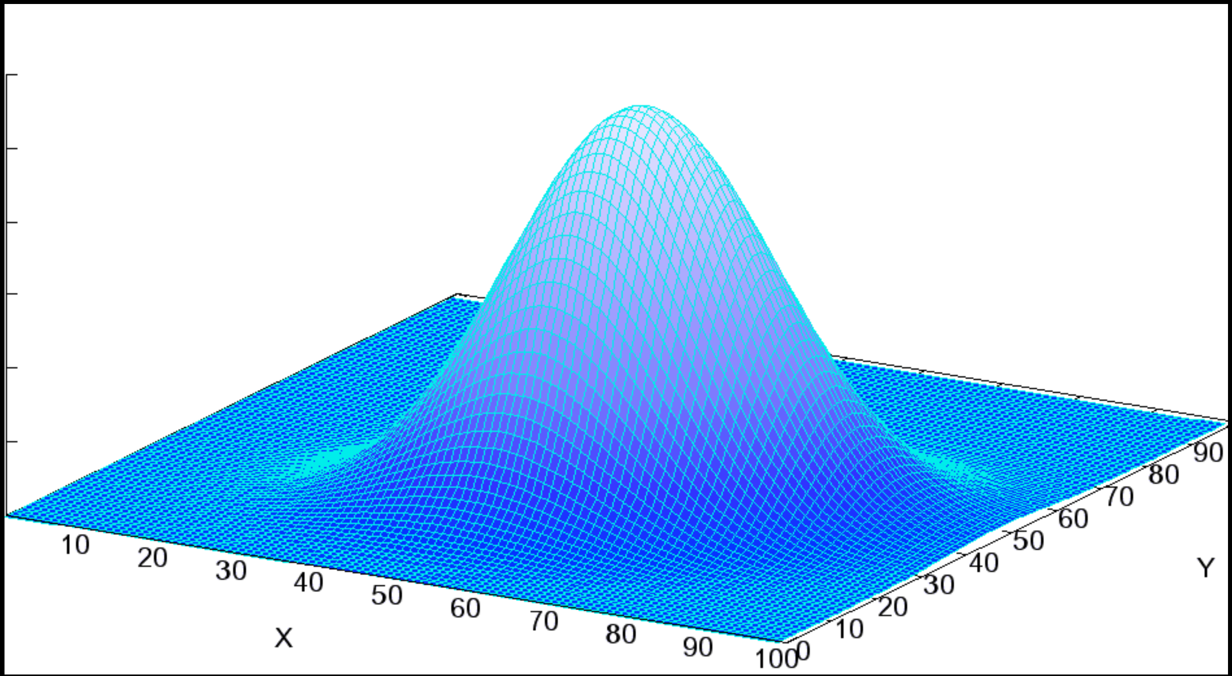
Analysis

Attractive. SDD

Gaussian Graphical Models (GGMs)

$X \sim N(0, \Sigma)$. $\Sigma \in R^{p \times p}$ covariance matrix.

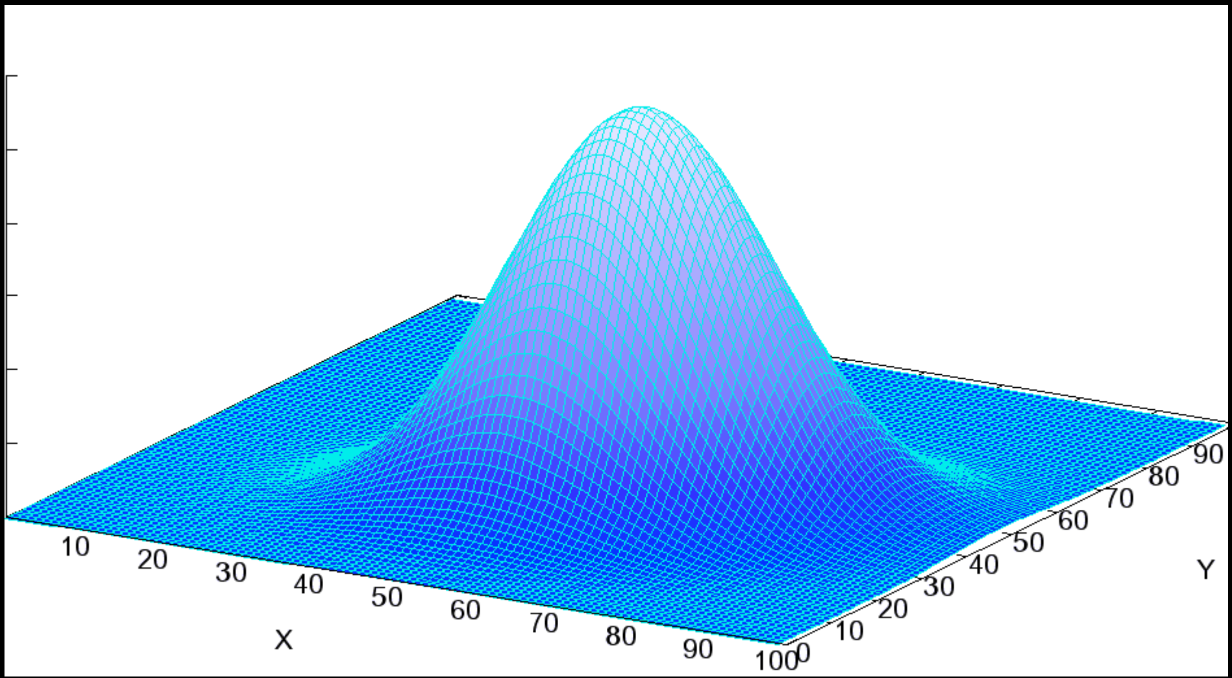
p = number of features.



$$Pr[X = x] = \frac{1}{\sqrt{(2\pi)^p det(\Sigma)}} exp(-x^T \Sigma^{-1} x/2)$$

Gaussian Graphical Models (GGMs)

$X \sim N(0, \Sigma)$. $\Sigma \in R^{p \times p}$ covariance matrix.

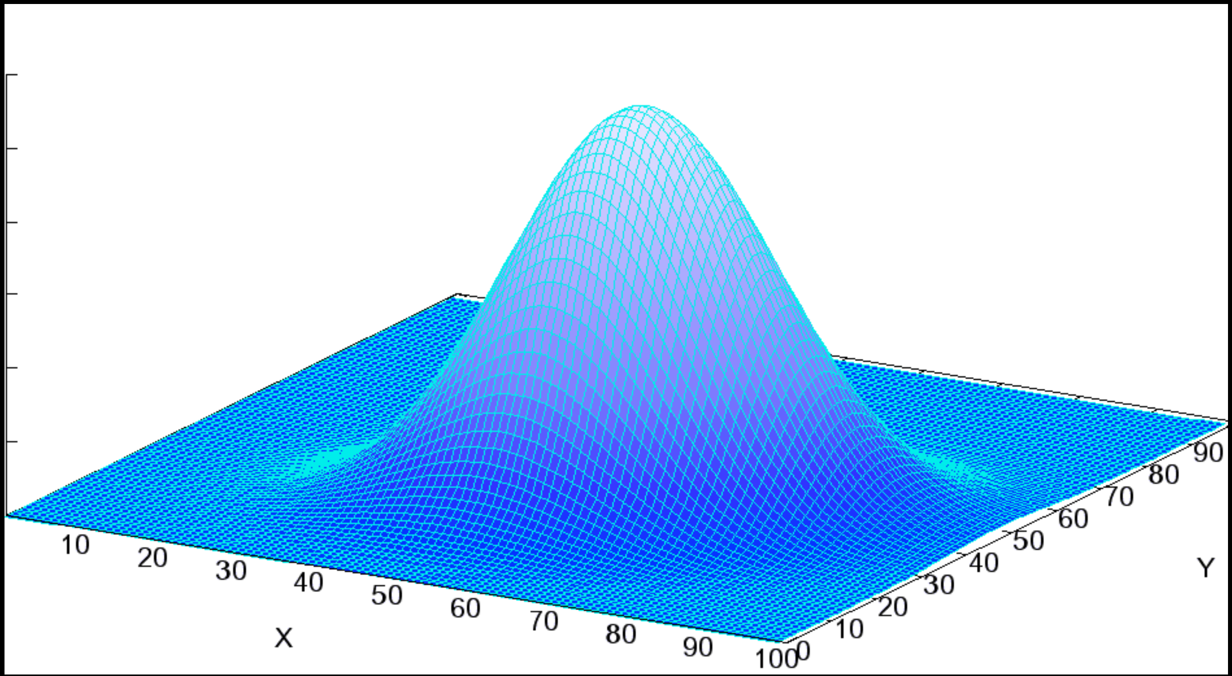


Today's Focus: Precision Matrix $\Theta = \Sigma^{-1}$.

$Cov(x)_{ij} = E[x_i, x_j]$

Gaussian Graphical Models (GGMs)

$X \sim N(0,\Sigma)$. $\Sigma \in R^{p\times p}$ covariance matrix.



Today's Focus: Precision Matrix $\Theta = \Sigma^{-1}$.

Dempster 72: Encodes conditional independence structure

Gaussian Graphical Models (GGMs)

$X \sim N(0,\Sigma)$. $\Sigma \in R^{p \times p}$ covariance matrix.

Precision Matrix $\Theta = \Sigma^{-1}$.

Gaussian Graphical Models (GGMs)

$X \sim N(0, \Sigma)$. $\Sigma \in R^{p \times p}$ covariance matrix.

Precision Matrix $\Theta = \Sigma^{-1}$.

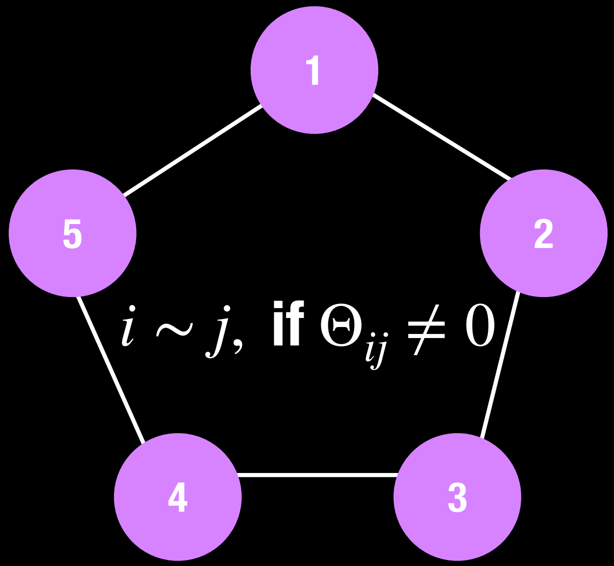
Dependency graph $G = Supp(\Theta)$.

Gaussian Graphical Models (GGMs)

$X \sim N(0, \Sigma)$. $\Sigma \in R^{p \times p}$ covariance matrix.

Precision Matrix $\Theta = \Sigma^{-1}$.

Dependency graph $G = Supp(\Theta)$.

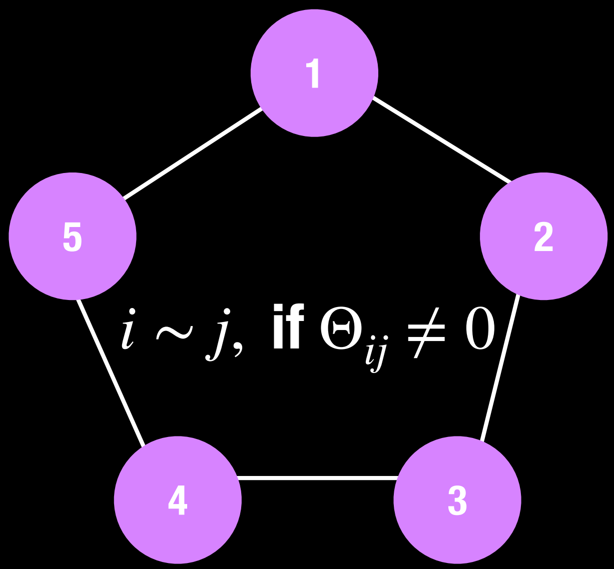


Gaussian Graphical Models (GGMs)

$X \sim N(0,\Sigma)$. $\Sigma \in R^{p \times p}$ covariance matrix.

Precision Matrix $\Theta = \Sigma^{-1}$.

Dependency graph $G = Supp(\Theta)$.



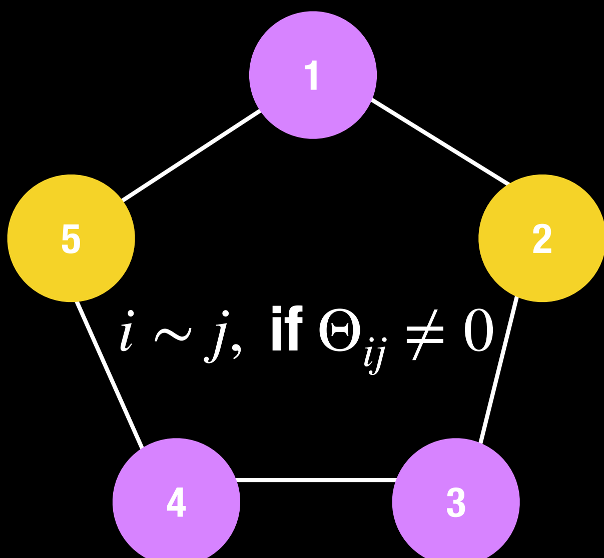
Markov property: $\Theta_{ij} = 0 \Rightarrow X_i, X_j$ **are independent conditioned on neighbors of** i**.**

Gaussian Graphical Models (GGMs)

$X \sim N(0,\Sigma)$. $\Sigma \in R^{p\times p}$ covariance matrix.

Precision Matrix $\Theta = \Sigma^{-1}$.

Dependency graph $G = Supp(\Theta)$.



Example: $(X_1 \mid X_2, X_5)$ **independent of** $(X_3 \mid X_2, X_5)$

Markov property: $\Theta_{ij} = 0 \Rightarrow X_i, X_j$ **are independent conditioned on neighbors of** i.

Gaussian Graphical Models (GGMs)

$X \sim N(0,\Sigma)$. $\Sigma \in R^{p \times p}$ covariance matrix.

Precision Matrix $\Theta = \Sigma^{-1}$.

Dependency graph $G = Supp(\Theta)$.

Complexity of GGMs: max-degree of G $d \ll p$.

Gaussian Graphical Models (GGMs)

$X \sim N(0,\Sigma)$. $\Sigma \in R^{p \times p}$ covariance matrix.

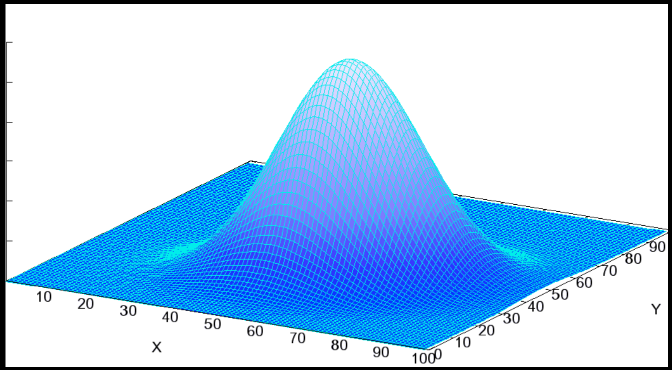
Precision Matrix $\Theta = \Sigma^{-1}$.

Dependency graph $G = Supp(\Theta)$.

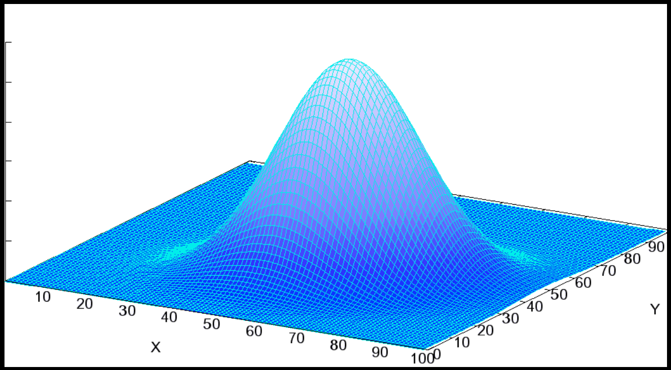
Complexity of GGMs: max-degree of G $d \ll p$.

The assumption that makes the formalism non-trivial and useful …

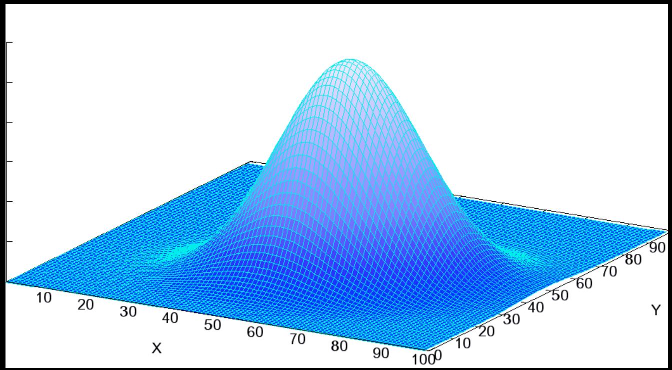
Why?



Fundamental model for modeling statistical relations between variables



Fundamental model for modeling statistical relations between variables



Many applications in machine learning, sciences:

Gene-regulatory networks

Brain connection networks from fMRI

...

Bigger Picture

Can we learn sparse dependency graphs from few samples?

(aka learning Markov random fields, undirected graphical models)

Bigger Picture

Can we learn sparse dependency graphs from few samples?

(aka learning Markov random fields, undirected graphical models)

$$X \sim \{1, -1\}^p\,.$$

Dependency graph of X: $i \nsim j \Rightarrow X_i, X_j$ **are independent conditioned on neighbors of** i.

Example: "Random Walk" Model

$$X_i = X_{i-1} + Z_i, \text{ where } Z_1, \ldots, Z_p \text{ i.i.d } N(0,1)$$

1	1	1	1	1	1	1	1	1	1
1	2	2	2	2	2	2	2	2	2
1	2	3	3	3	3	3	3	3	3
1	2	3	4	4	4	4	4	4	4
1	2	3	4	5	5	5	5	5	5
1	2	3	4	5	6	6	6	6	6
1	2	3	4	5	6	7	7	7	7
1	2	3	4	5	6	7	8	8	8
1	2	3	4	5	6	7	8	9	9
1	2	3	4	5	6	7	8	9	10

Σ : Covariance Matrix

$E[x_1, x_2] = 1$

$E[x_2, x_3] = 2$

$x_1 = z_1 \quad x_2 = z_1 + z_2 \quad x_3 = z_1 + z_2 + z_3$

Example: "Random Walk" Model

$X_i = X_{i-1} + Z_i$, **where** $Z_1, \ldots, Z_p$ **i.i.d** $N(0,1)$

1	1	1	1	1	1	1	1	1	1
1	2	2	2	2	2	2	2	2	2
1	2	3	3	3	3	3	3	3	3
1	2	3	4	4	4	4	4	4	4
1	2	3	4	5	5	5	5	5	5
1	2	3	4	5	6	6	6	6	6
1	2	3	4	5	6	7	7	7	7
1	2	3	4	5	6	7	8	8	8
1	2	3	4	5	6	7	8	9	9
1	2	3	4	5	6	7	8	9	10

Σ : Covariance Matrix

2	-1	0	0	0	0	0	0	0	0
-1	2	-1	0	0	0	0	0	0	0
0	-1	2	-1	0	0	0	0	0	0
0	0	-1	2	-1	0	0	0	0	0
0	0	0	-1	2	-1	0	0	0	0
0	0	0	0	-1	2	-1	0	0	0
0	0	0	0	0	-1	2	-1	0	0
0	0	0	0	0	0	-1	2	-1	0
0	0	0	0	0	0	0	-1	2	-1
0	0	0	0	0	0	0	0	-1	2

Θ : Precision Matrix

Example: “Random Walk” Model

$$X_i = X_{i-1} + Z_i, \text{ where } Z_1, \ldots, Z_p \text{ i.i.d } N(0,1)$$

1	1	1	1	1	1	1	1	1	1
1	2	2	2	2	2	2	2	2	2
1	2	3	3	3	3	3	3	3	3
1	2	3	4	4	4	4	4	4	4
1	2	3	4	5	5	5	5	5	5
1	2	3	4	5	6	6	6	6	6
1	2	3	4	5	6	7	7	7	7
1	2	3	4	5	6	7	8	8	8
1	2	3	4	5	6	7	8	9	9
1	2	3	4	5	6	7	8	9	10

Σ : Covariance Matrix

Dependency graph

2	-1	0	0	0	0	0	0	0	0
-1	2	-1	0	0	0	0	0	0	0
0	-1	2	-1	0	0	0	0	0	0
0	0	-1	2	-1	0	0	0	0	0
0	0	0	-1	2	-1	0	0	0	0
0	0	0	0	-1	2	-1	0	0	0
0	0	0	0	0	-1	2	-1	0	0
0	0	0	0	0	0	-1	2	-1	0
0	0	0	0	0	0	0	-1	2	-1
0	0	0	0	0	0	0	0	-1	2

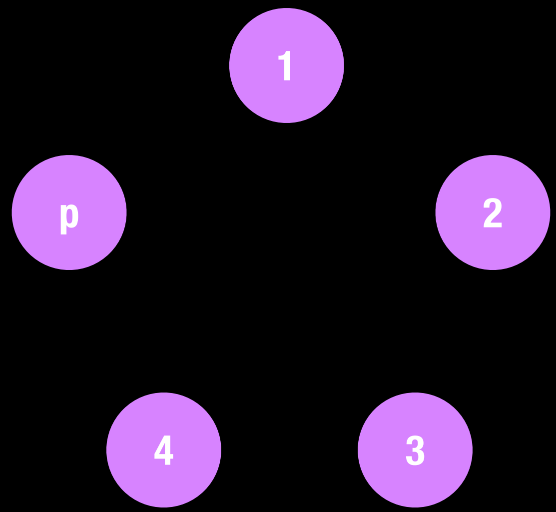
Θ : Precision Matrix

Learning Sparse GGMs

- Structure learning - learn dependency graph.
- Parameter learning - learn the matrix.

Structure Learning for GGMs

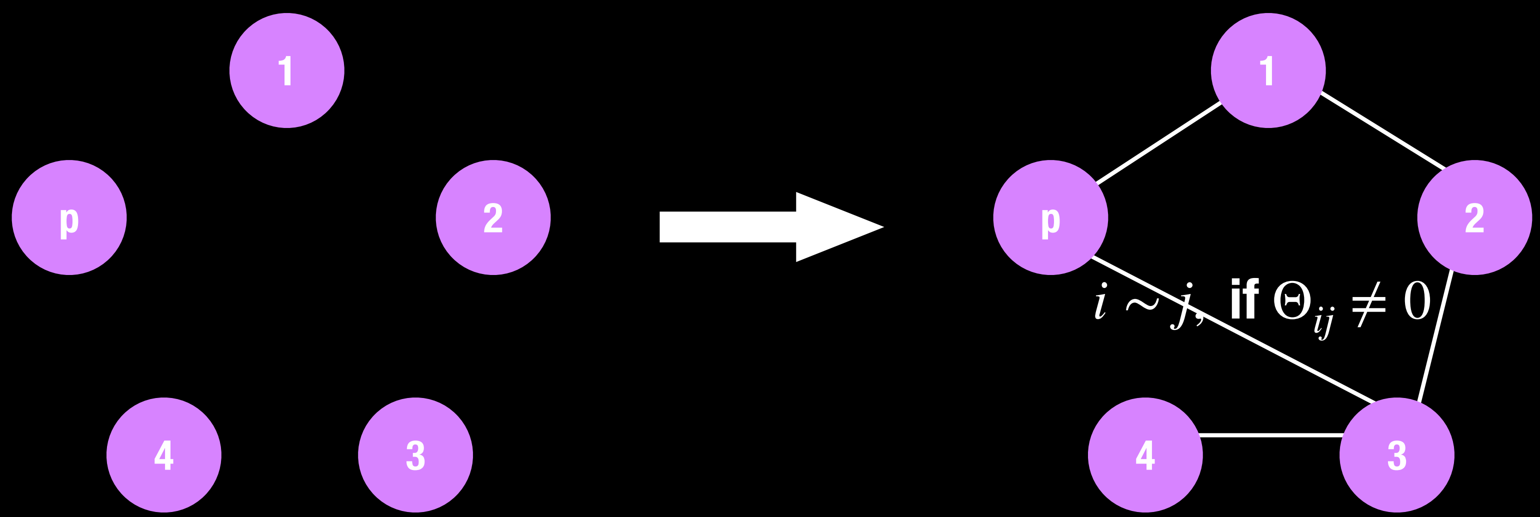
- Structure learning - learn dependency graph.



Input: Given samples $X^1, X^2, \ldots, X^n$ from a GGM
Output: Dependency graph of the GGM.

Structure Learning for GGMs

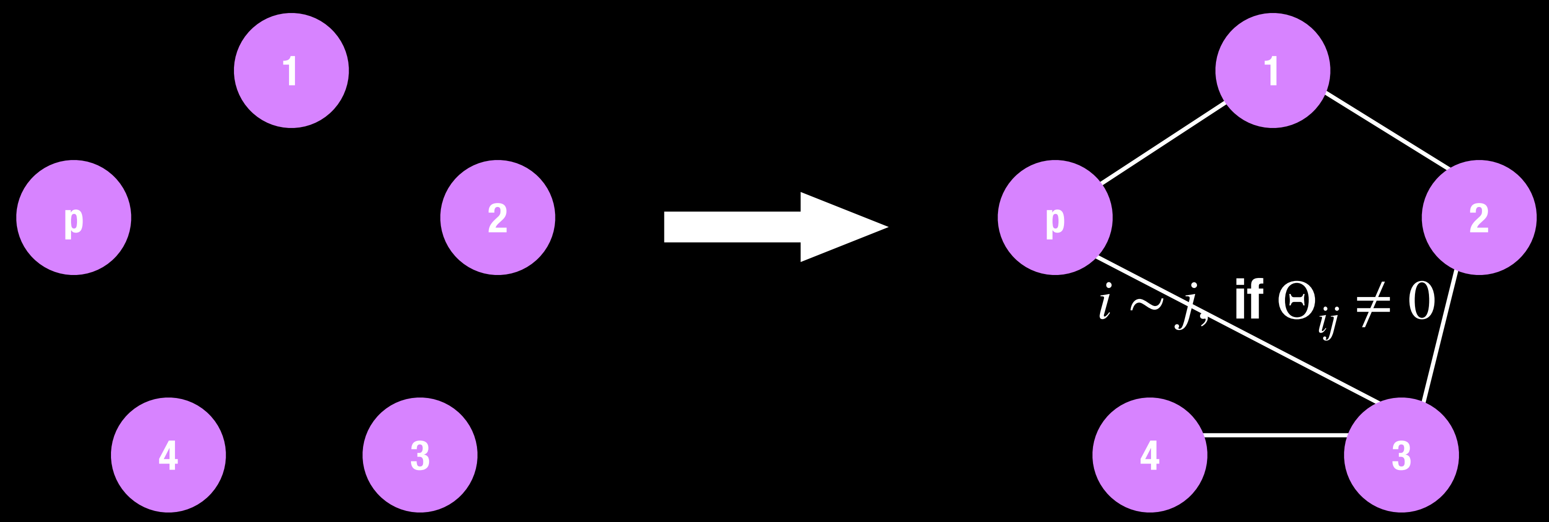
- **Structure learning - learn dependency graph.**



Input: Given samples $X^1, X^2, \ldots, X^n$ from a GGM
Output: Dependency graph of the GGM.

Structure Learning for GGMs

- Structure learning - learn dependency graph.

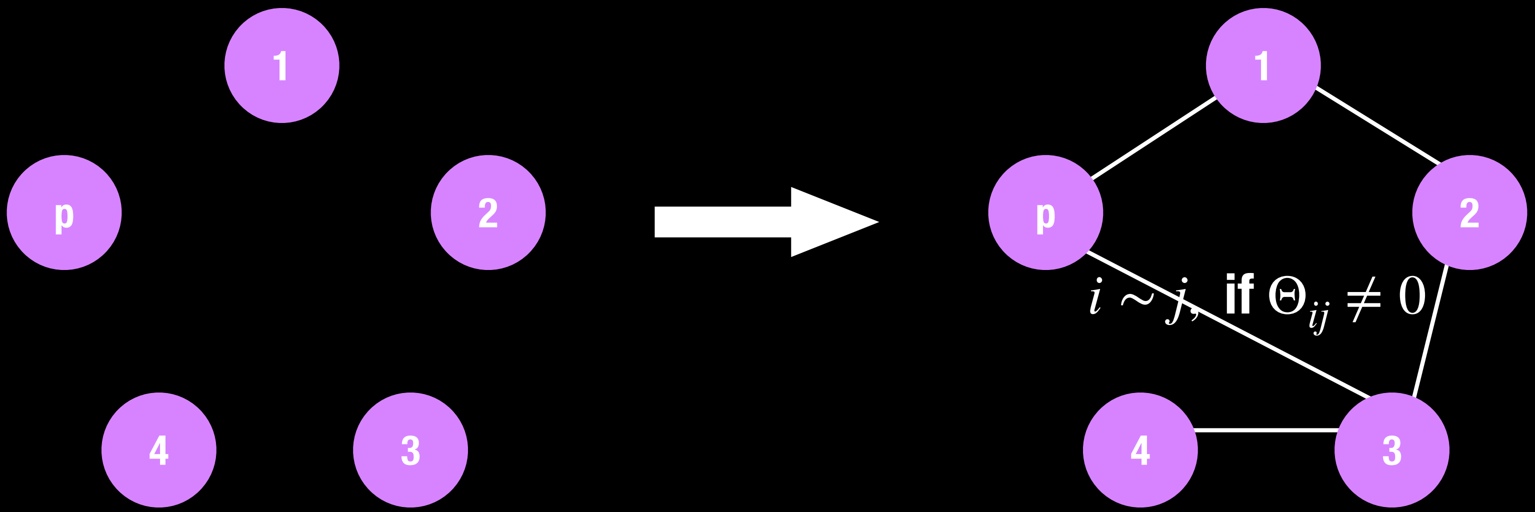


Input: Given samples $X^1, X^2, \ldots, X^n$ from a GGM
Output: Dependency graph of the GGM.

Challenge: Often $n \ll p$.

Structure Learning for GGMs

- Structure learning - learn dependency graph.



Assumption: Unknown dependency graph is sparse - each vertex has at most d edges.

Challenge: Often $n \ll p$.

Structure Learning for GGMs

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

(Think: $n = O_d(\log p)$. **)**

Structure Learning for GGMs

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

(Think: $n = O_d(\log p)$. **)**

Core of problem. Can learn parameters easily afterward.

Structure Learning for GGMs

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

(Think: $n = O_d(\log p)$. **)**

Ideal: Practical algorithms with provable guarantees.

Example: Unknown order Random Walk

$$X_{\pi(i)} = X_{\pi(i-1)} + Z_i, \text{ where } Z_1, \ldots, Z_p \text{ i.i.d } N(0,1)$$

7	1	5	6	7	4	3	7	2	7
1	1	1	1	1	1	1	1	1	1
5	1	5	5	5	4	3	5	2	5
6	1	5	6	6	4	3	6	2	6
7	1	5	6	8	4	3	8	2	8
4	1	4	4	4	4	3	4	2	4
3	1	3	3	3	3	3	3	2	3
7	1	5	6	8	4	3	9	2	9
2	1	2	2	2	2	2	2	2	2
7	1	5	6	8	4	3	9	2	10

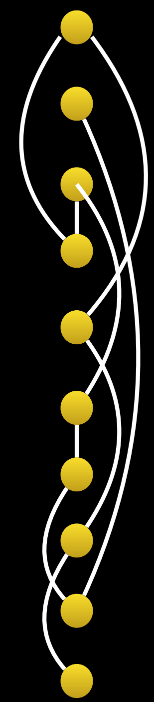
Σ : Covariance Matrix

Example: Unknown order Random Walk

$$X_{\pi(i)} = X_{\pi(i-1)} + Z_i, \text{ where } Z_1, \ldots, Z_p \text{ i.i.d } N(0,1)$$

7	1	5	6	7	4	3	7	2	7
1	1	1	1	1	1	1	1	1	1
5	1	5	5	5	4	3	5	2	5
6	1	5	6	6	4	3	6	2	6
7	1	5	6	8	4	3	8	2	8
4	1	4	4	4	4	3	4	2	4
3	1	3	3	3	3	3	3	2	3
7	1	5	6	8	4	3	9	2	9
2	1	2	2	2	2	2	2	2	2
7	1	5	6	8	4	3	9	2	10

Σ : Covariance Matrix



Dependency graph?

2	0	0	-1	-1	0	0	0	0	0
0	2	0	0	0	0	0	0	-1	0
0	0	2	-1	0	-1	0	0	0	0
-1	0	-1	2	0	0	0	0	0	0
-1	0	0	0	2	0	0	-1	0	0
0	0	-1	0	0	2	-1	0	0	0
0	0	0	0	0	-1	2	0	-1	0
0	0	0	0	-1	0	0	2	0	-1
0	-1	0	0	0	0	-1	0	2	0
0	0	0	0	0	0	0	-1	0	1

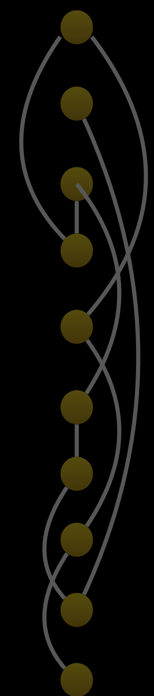
Θ : Precision Matrix

Example: Unknown order Random Walk

$$X_{\pi(i)} = X_{\pi(i-1)} + Z_i,\ \textbf{where } Z_1, \ldots, Z_p \textbf{ i.i.d } N(0,1)$$

7	1	5	6	7	4	3	7	2	7
1	1	1	1	1	1	1	1	1	1
5	1	5	5	5	4	3	5	2	5
6	1	5	6	6	4	3	6	2	6
7	1	5	6	8	4	3	8	2	8
4	1	4	4	4	4	3	4	2	4
3	1	3	3	3	3	3	3	2	3
7	1	5	6	8	4	3	9	2	9
2	1	2	2	2	2	2	2	2	2
7	1	5	6	8	4	3	9	2	10

Σ : Covariance Matrix



Dependency graph?

2	0	0	-1	-1	0	0	0	0	0
0	2	0	0	0	0	0	0	-1	0
0	0	2	-1	0	-1	0	0	0	0
-1	0	-1	2	0	0	0	0	0	0
-1	0	0	0	2	0	0	-1	0	0
0	0	-1	0	0	2	-1	0	0	0
0	0	0	0	0	-1	2	0	-1	0
0	0	0	0	-1	0	0	2	0	-1
0	-1	0	0	0	0	-1	0	2	0
0	0	0	0	0	0	0	-1	0	1

Θ : Precision Matrix

How many samples of X to find the hidden permutation?

Previous Work

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

Well studied with several popular software packages:
GLASSO, CLIME, ACLIME

Previous Work

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

Previous Work

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

- **GLASSO: Friedman, Hastie, Tibshirani 08**

  Can learn with $O(d^2 \log p)$ samples if precision matrix is incoherent.

Previous Work

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

Strong assumption: Violated by random walk …

- GLASSO: Friedman, Hastie, Tibshirani 08

  Can learn with $O(d^2 \log p)$ samples if precision matrix is incoherent.

Previous Work

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

- GLASSO: Friedman, Hastie, Tibshirani 08

  Can learn with $O(d^2 \log p)$ samples if precision matrix is incoherent.

- CLIME: Cai, Liu, Luo 2011

  Can learn with $O(M^4 \log p)$ samples where M ~ maximum ratio of entries in Σ, Θ .

Previous Work

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

- GLASSO: Friedman, Hastie, Tibshirani 08

  Can learn with $O(d^2 \log p)$ samples if precision

  Strong assumption: Violated by random walk …

- CLIME: Cai, Liu, Luo 2011

  Can learn with $O(M^4 \log p)$ samples where M ~ maximum ratio of entries in Σ, Θ .

Previous Work

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

- **GLASSO: Friedman, Hastie, Tibshirani 08**

  Can learn with $O(d^2 \log p)$ samples if precision matrix is incoherent.

- **CLIME: Cai, Liu, Luo 2011**

  Can learn with $O(M^4 \log p)$ samples where M ~ maximum ratio of entries in Σ, Θ .

Previous Work

$\boxed{\text{Given samples } X^1, X^2, \ldots, X^n \text{ from a GGM of degree } d \ll p\text{, can we efficiently find the dependency graph with } n \ll p?}$

Summary: GLASSO, CLIME, ACLIME need `well-conditioned' precision matrix.

Previous Work

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

Summary: GLASSO, CLIME, ACLIME need `well-conditioned' precision matrix.

- Violated by simple models
- Not scale invariant
- Not just analysis … **fail empirically**

Previous Work

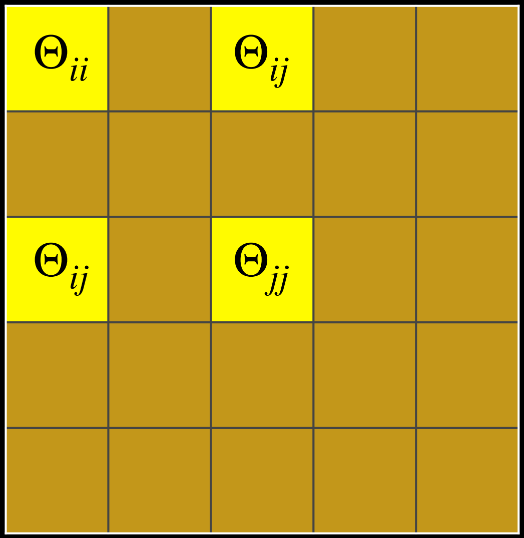
Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

What is possible information theoretically?

Previous Work: MVL18

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

$$\kappa(\Theta) = \min_{i,j:\Theta_{ij}\neq 0} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$$



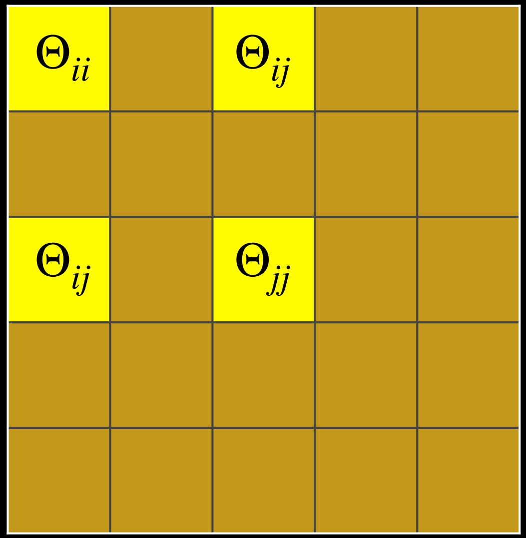
Θ : Precision Matrix

Previous Work: MVL18

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

$$\kappa(\Theta) = \min_{i,j:\Theta_{ij}\neq 0} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$$

Measures signal in 2x2 submatrices … not whole matrix.



Θ : Precision Matrix

Previous Work: MVL18

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

MVL18: $\kappa(\Theta) = \min_{i,j:\Theta_{ij}\neq 0} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$

Dependency graph identifiable with $n = O(d \log p / \kappa^2)$ samples!

Previous Work: MVL18

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

MVL18: $\kappa(\Theta) = \min_{i,j:\Theta_{ij}\neq 0} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$

Dependency graph identifiable with $n = O(d \log p/\kappa^2)$ samples!

Example: Random walk model - $\kappa = 1/2$.

Information-Theoretic Limits: MVL18

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

MVL18: $\kappa(\Theta) = \min_{i,j:\Theta_{ij}\neq 0} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$

Dependency graph identifiable with $n = O(d \log p/\kappa^2)$ samples!

Wang, Wainwright, Ramachandran 10: Need $n = \Omega(\log p/\kappa^2)$.

Information-Theoretic Limits: MVL18

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

MVL18: $\kappa(\Theta) = \min_{i,j:\Theta_{ij}\neq 0} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$

Dependency graph identifiable with $n = O(d \log p / \kappa^2)$ samples!

Run-time of algorithm: $p^{O(d)}$.

Information-Theoretic Limits: MVL18

Given samples $X^1, X^2, \ldots, X^n$ from a GGM of degree $d \ll p$, can we efficiently find the dependency graph with $n \ll p$?

MVL18: $\kappa(\Theta) = \min_{i,j:\Theta_{ij} \neq 0} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$

Dependency graph identifiable with $n = O(d \log p / \kappa^2)$ samples!

Run-time of algorithm: $p^{O(d)}$.

Problematic for even moderate sized instances
… Can we do better?

GGMS: Main Learning Challenge

Given n samples from a GGM of degree $d \ll p$,
can we find the dependency graph with $n \approx d \log p / \kappa^2$, and
run-time fixed polynomial in p? (Or even $p^{o(d)}$.)

GGMS: Main Learning Challenge

Given n samples from a GGM of degree $d \ll p$, can we find the dependency graph with $n \approx d \log p / \kappa^2$, and run-time fixed polynomial in p? (Or even $p^{o(d)}$.)

This work — YES for a large class of models:
Attractive, SDD,… more generally, Walk-Summable

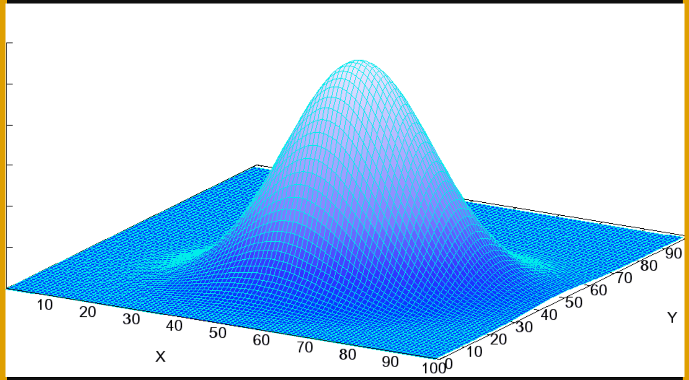
GGMS: Main Learning Challenge

Given n samples from a GGM of degree $d \ll p$,
can we find the dependency graph with $n \approx d \log p/\kappa^2$, and
run-time fixed polynomial in p? (Or even $p^{o(d)}$.)

This work — YES for a large class of models:
Attractive, SDD,… more generally, Walk-Summable

Main: A simple greedy algorithm solves above ill-conditioned cases
(and recovers guarantees of GLASSO, CLIME, …)

GGMs



Motivation

Algorithm

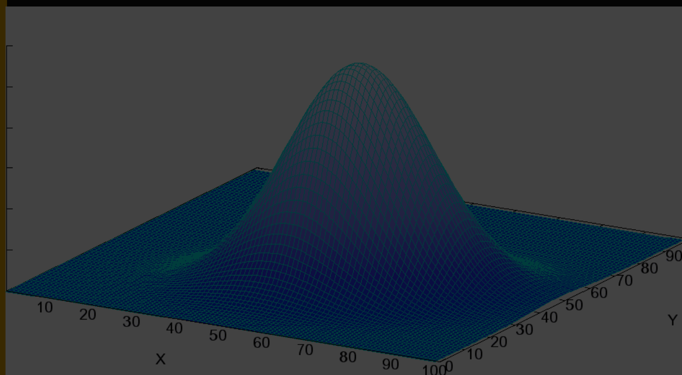
$$\boxed{\begin{array}{c}\text{While } |S| < t : \\ \text{Add } \arg\min_j Var(X_i \mid X_{S \cup j}) \,.\end{array}}$$

Special Models

Analysis

Attractive. SDD.

GGMs



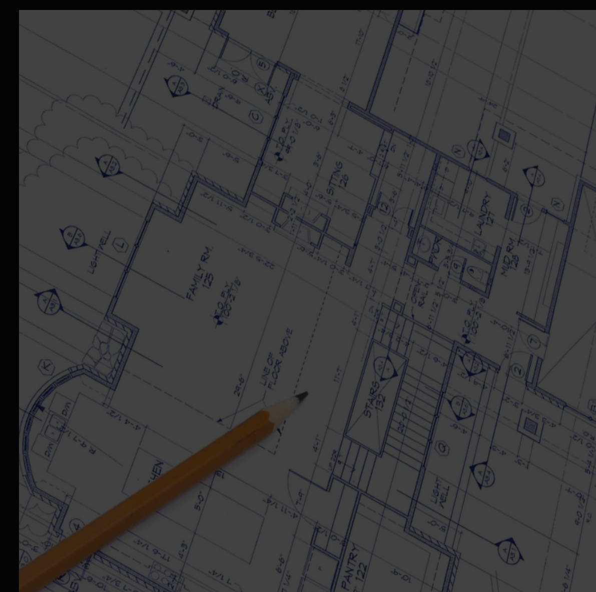
Motivation

Algorithm

While $|S| < t$:
Add $\arg\min_j Var(X_i \,|\, X_{S\cup j})$.

Special Models

Analysis

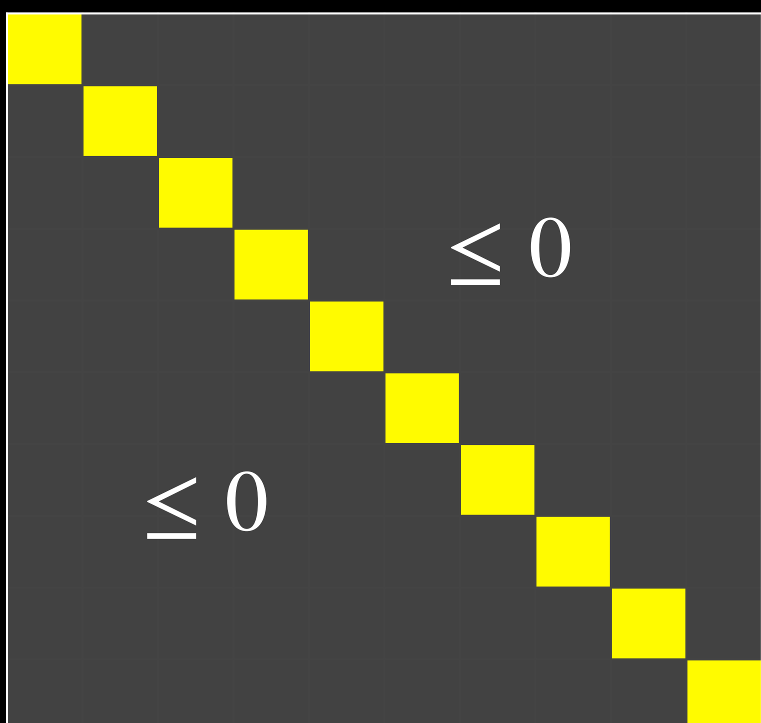


Attractive. SDD.

Attractive GGMs

GGM is attractive if all covariances are non-negative.

(Equivalently, Θ has non-positive off-diagonals.)

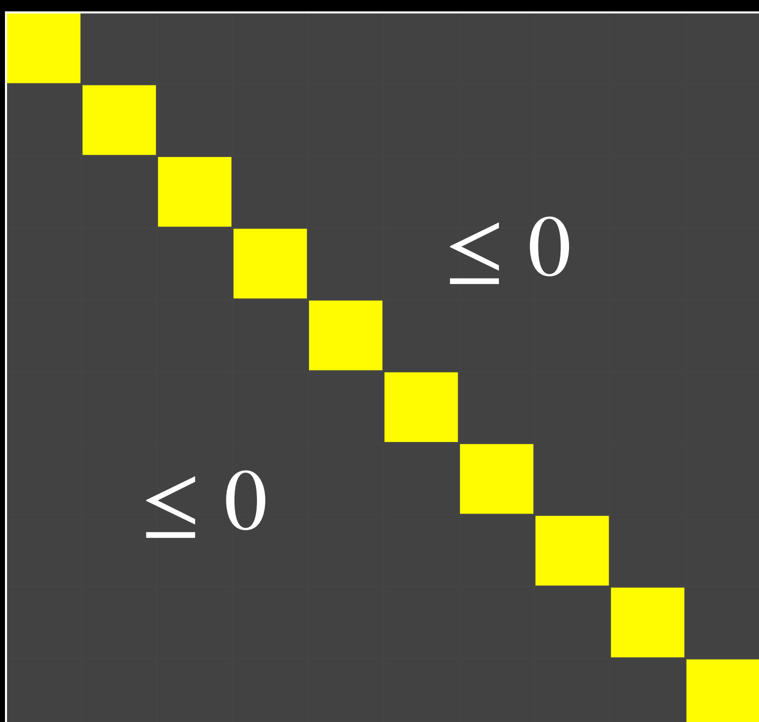


Θ : Precision Matrix

Attractive GGMs

GGM is attractive if all covariances are non-negative.

(Equivalently, Θ has non-positive off-diagonals.)



Θ : Precision Matrix

Ex: Gaussian Free Fields

- **Many applications via Gaussian processes**

Attractive GGMs

GGM is attractive if all covariances are non-negative.

(Equivalently, Θ has non-positive off-diagonals.)

2	-1	0	0	0	0	0	0	0	0
-1	2	-1	0	0	0	0	0	0	0
0	-1	2	-1	0	0	0	0	0	0
0	0	-1	2	-1	0	0	0	0	0
0	0	0	-1	2	-1	0	0	0	0
0	0	0	0	-1	2	-1	0	0	0
0	0	0	0	0	-1	2	-1	0	0
0	0	0	0	0	0	-1	2	-1	0
0	0	0	0	0	0	0	-1	2	-1
0	0	0	0	0	0	0	0	-1	2

Θ : Precision Matrix

Ex: Gaussian Free Fields

- Many applications via Gaussian processes
- Ex: Random walk model

Attractive GGMs

GGM is attractive if all covariances are non-negative.

(Equivalently, Θ has non-positive off-diagonals.)

2	-1	0	0	0	0	0	0	0	0
-1	2	-1	0	0	0	0	0	0	0
0	-1	2	-1	0	0	0	0	0	0
0	0	-1	2	-1	0	0	0	0	0
0	0	0	-1	2	-1	0	0	0	0
0	0	0	0	-1	2	-1	0	0	0
0	0	0	0	0	-1	2	-1	0	0
0	0	0	0	0	0	-1	2	-1	0
0	0	0	0	0	0	0	-1	2	-1
0	0	0	0	0	0	0	0	-1	2

Θ : Precision Matrix

Ex: Gaussian Free Fields

- Many applications via Gaussian processes
- Ex: Random walk model
- Ill-conditioned if ‘long paths’

Attractive GGMs

GGM is attractive if all covariances are non-negative.

Previous: No efficient algorithms with $O_d(\log p)$ sample complexity known.

Attractive GGMs

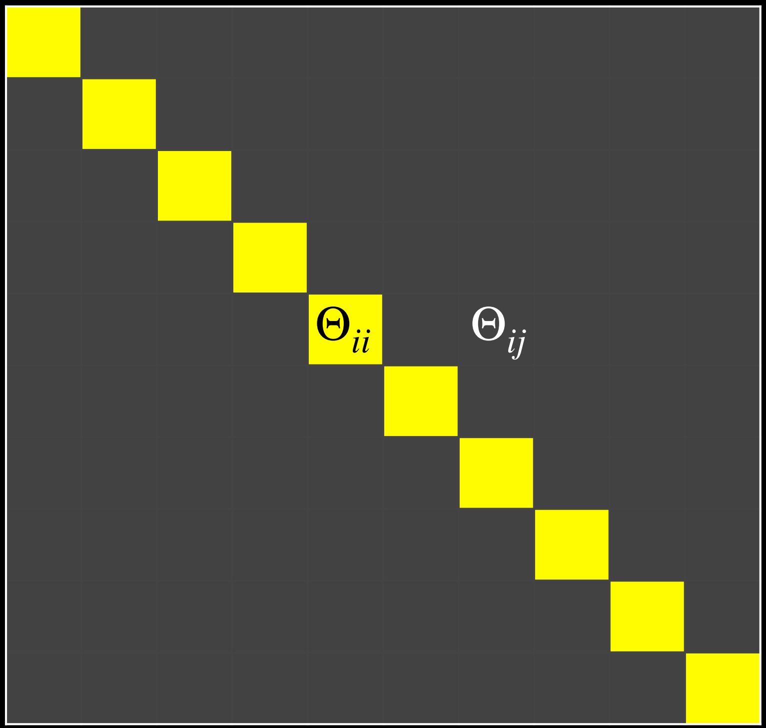
GGM is attractive if all covariances are non-negative.

KKMM: `GreedyPrune` learns attractive models with $\tilde{O}(d \log p/\kappa^2)$ samples and quadratic run-time.

Previous: No efficient algorithms with $O_d(\log p)$ sample complexity known.

Walk-Summable GGMs

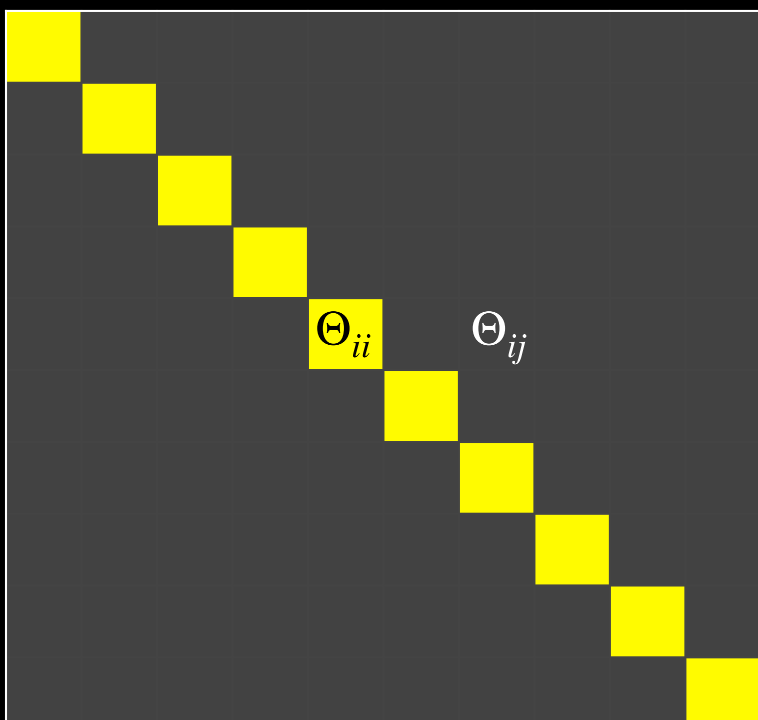
GGM walk-summable if making off-diagonals of precision matrix negative preserves positive semi-definiteness.



Θ : Precision Matrix

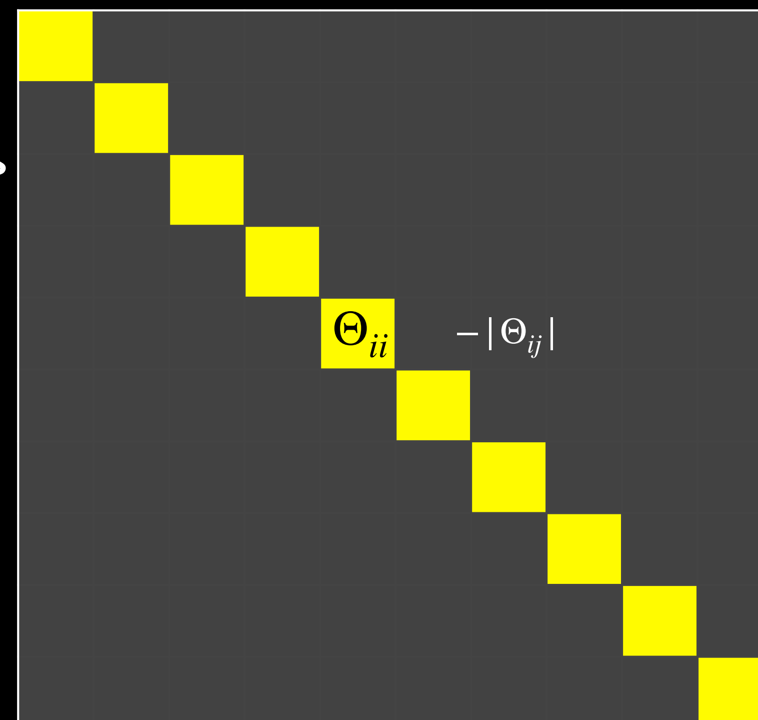
Walk-Summable GGMs

GGM walk-summable if making off-diagonals of precision matrix negative preserves positive semi-definiteness.



Θ : Precision Matrix

all eigenvalues > 0

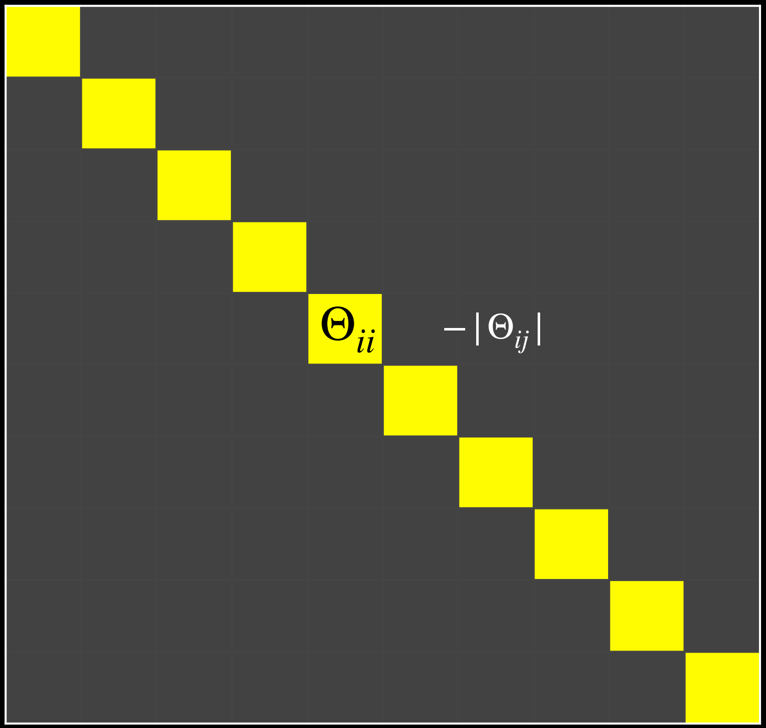


Offdiagonals negative $\succeq 0$

Walk-Summable GGMs

GGM walk-summable if making off-diagonals of precision matrix negative preserves positive semi-definiteness.

- Introduced by Maliutov, Johnson, Willsky 2006
- Generalize many classes
  - Attractive
  - Pairwise normalizable
  - Non-frustrated
  - Symmetric diagonally-dominant



Offdiagonals negative $\succeq 0$

Walk-Summable GGMs

GGM walk-summable if making off-diagonals of precision matrix negative preserves positive semi-definiteness.

Previous: ATHW12, MVL18 - $p^{O(d)}$ run-time algorithm with logarithmic samples.

Walk-Summable GGMs

GGM walk-summable if making off-diagonals of precision matrix negative preserves positive semi-definiteness.

KKMM: **GreedyPrune** learns walk-summable models with $O(d^2 \log p/\kappa^6)$ samples and quadratic run-time.

Previous: ATHW12, MVL18 - $p^{O(d)}$ run-time algorithm with logarithmic samples.

Walk-Summable GGMs

GGM walk-summable if making off-diagonals of precision matrix negative preserves positive semi-definiteness.

KKMM: **Hybrid** (Greedy+Lasso) learns walk-summable models with $O(d \log p / \kappa^4)$ samples.

Previous: ATHW12, MVL18 - $p^{O(d)}$ run-time algorithm with logarithmic samples.

Learning GGMs Greedily

Input: Samples from a sparse GGM ~ X.
Output: Dependency graph of X.

Similar greedy approaches for discrete GMs: [Bresler10], [HKM17], [BKM19]

Learning GGMs Greedily

Input: Samples from a sparse GGM ~ X.
Output: Dependency graph of X.

GREEDYPRUNE

1. **Recover neighborhood of each vertex in parallel.**
2. **Grow a candidate neighborhood.**
3. **Prune out some vertices.**

Similar greedy approaches for discrete GMs: [Bresler10], [HKM17], [BKM19]

Phase 1: Growing a neighborhood

Input: Samples from a sparse GGM ~ X.
Goal: Neighborhood of vertex 1.

Phase 1: Growing a neighborhood

Input: Samples from a sparse GGM ~ X.
Goal: Neighborhood of vertex 1.

GREEDY-GROWING

1. **Set** $S \leftarrow \emptyset$
2. **While S is small enough:**
   1. **Find j to minimize estimate of** $Var(X_1 \,|\, X_{S \cup j})$.

      ↳ For each $var(x_1 | x_{S \cup j})$ we need $1/\varepsilon^2$ samples to get $(1-\varepsilon)$ accuracy.
   2. $S \leftarrow S \cup \{j\}$.

Intuition: Add vertex that gives maximum decrease in conditional variance.

Phase 2: Pruning a neighborhood

Input: Samples from a sparse GGM ~ X.
Goal: Neighborhood of vertex 1.

Phase 2: Pruning a neighborhood

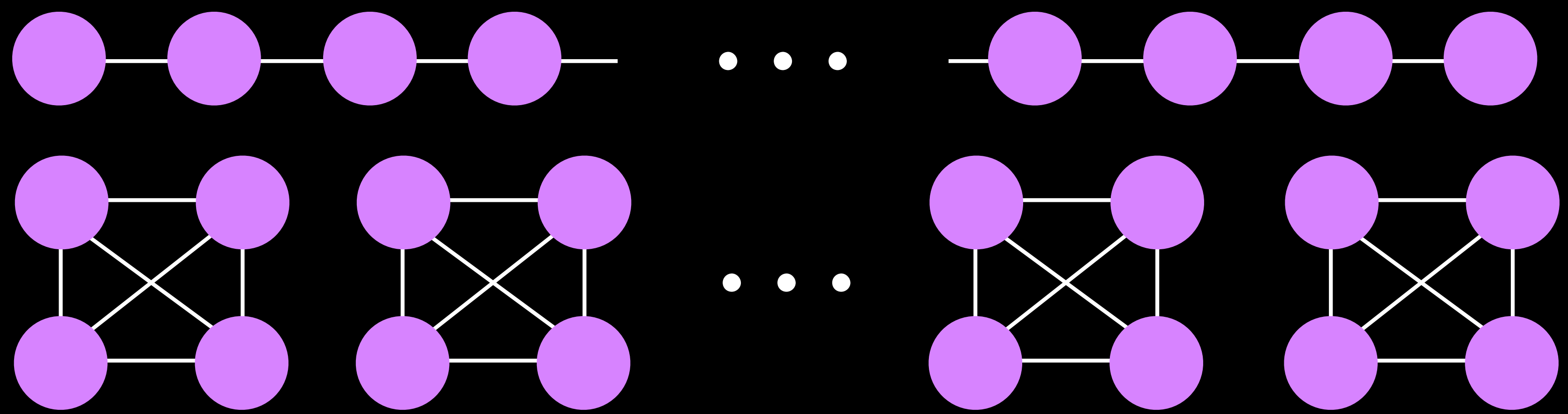
Input: Samples from a sparse GGM ~ X.
Goal: Neighborhood of vertex 1.

GREEDY-PRUNING

1. For each j in S:
    1. If $Var(X_1 \,|\, X_{S\setminus\{j\}}) < (1+\tau)Var(X_1 \,|\, X_S)$, drop j from S.

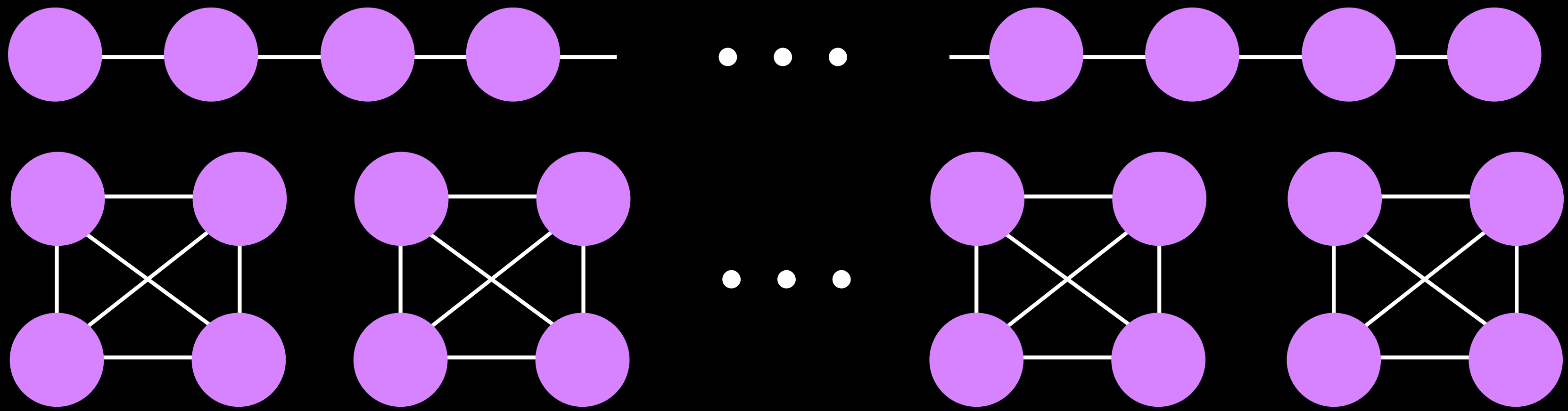
Intuition: If dropping a vertex, does not hurt too much, drop it.

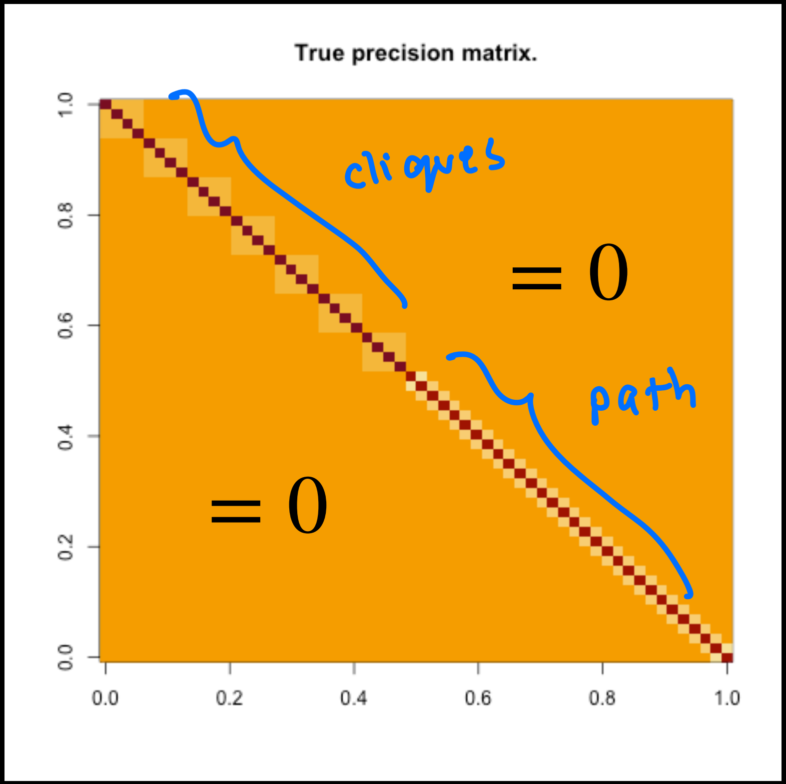
Experiments: A Simple Challenge



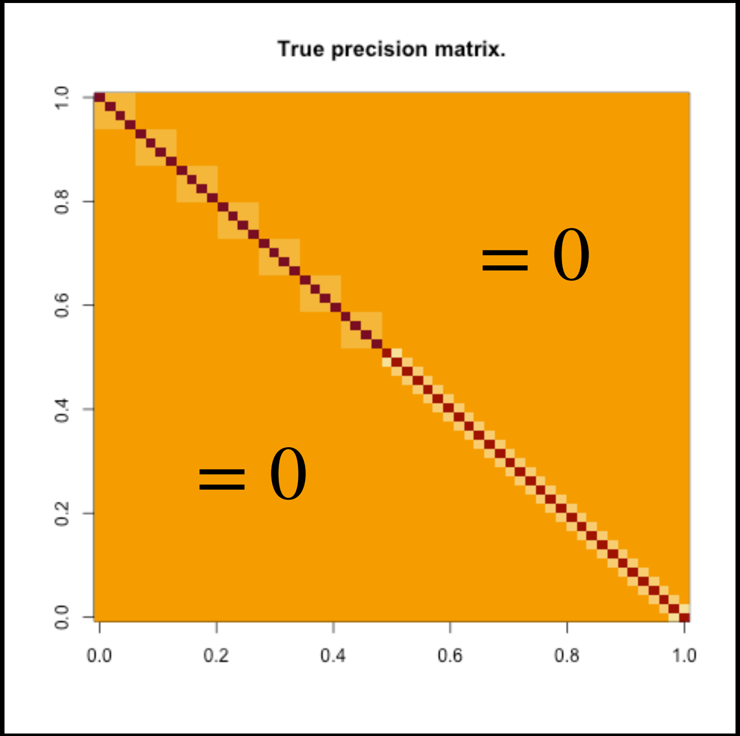
Precision: Path + Cliques
Max-degree = 3
Ill-conditioned

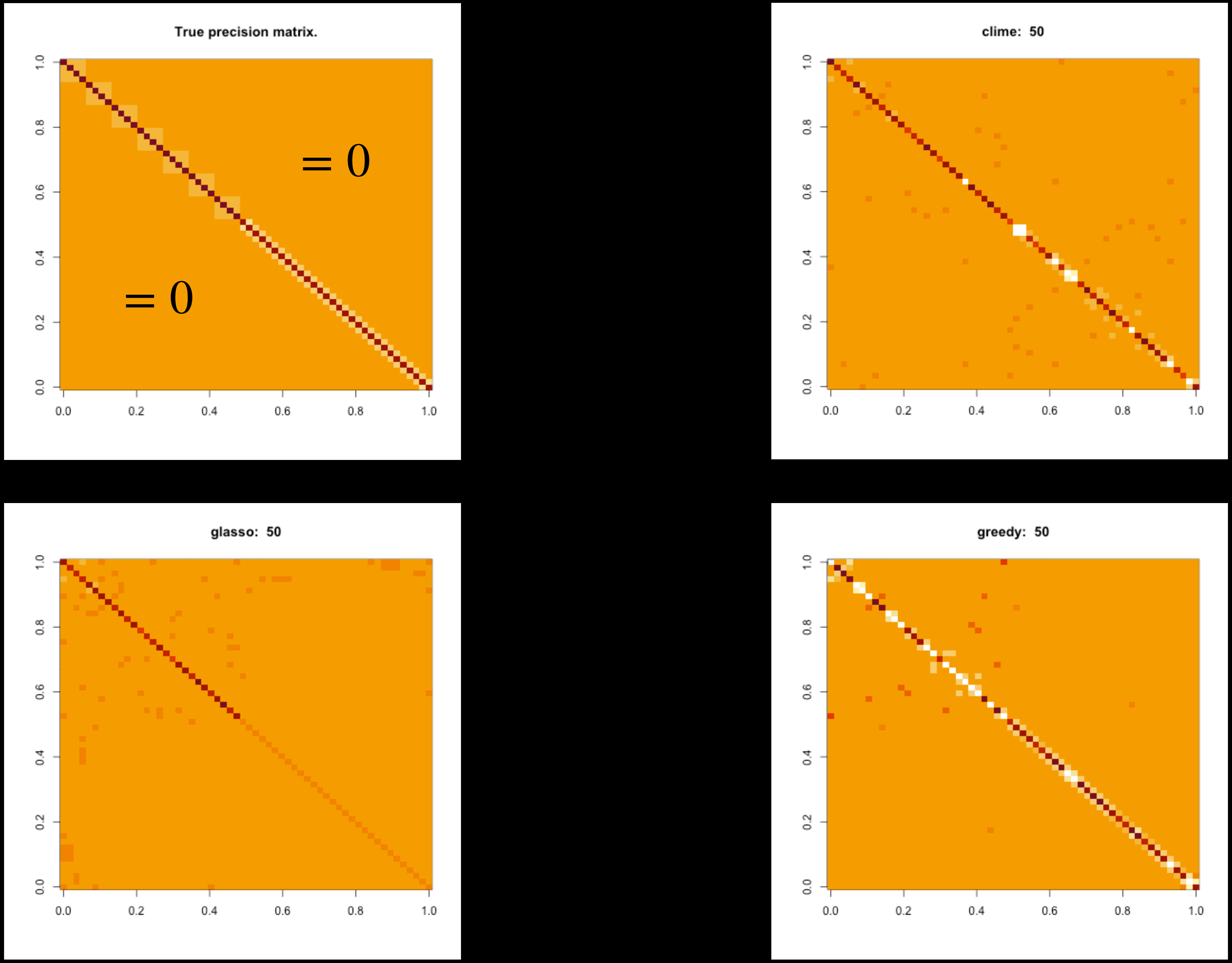
Experiments: A Simple Challenge

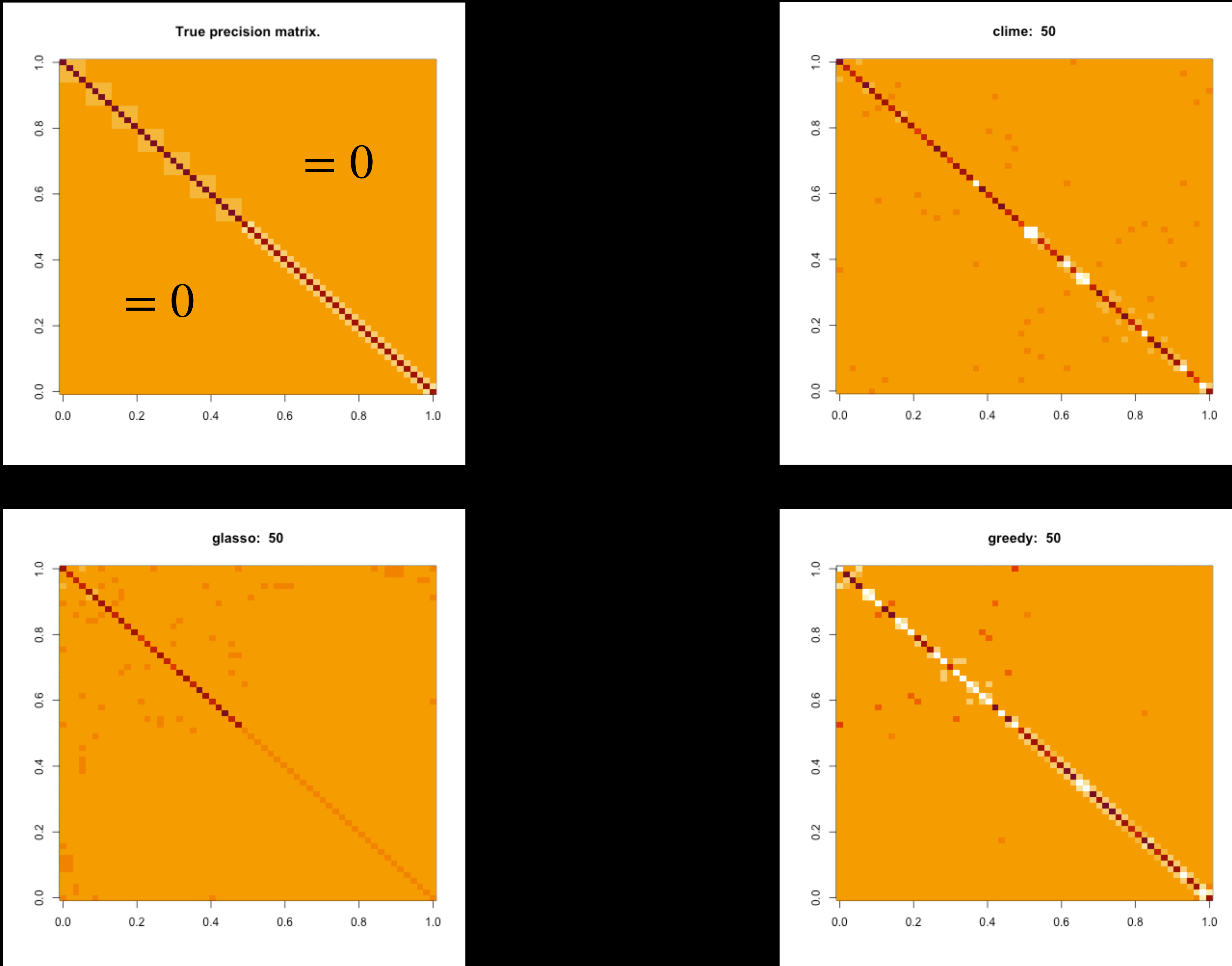


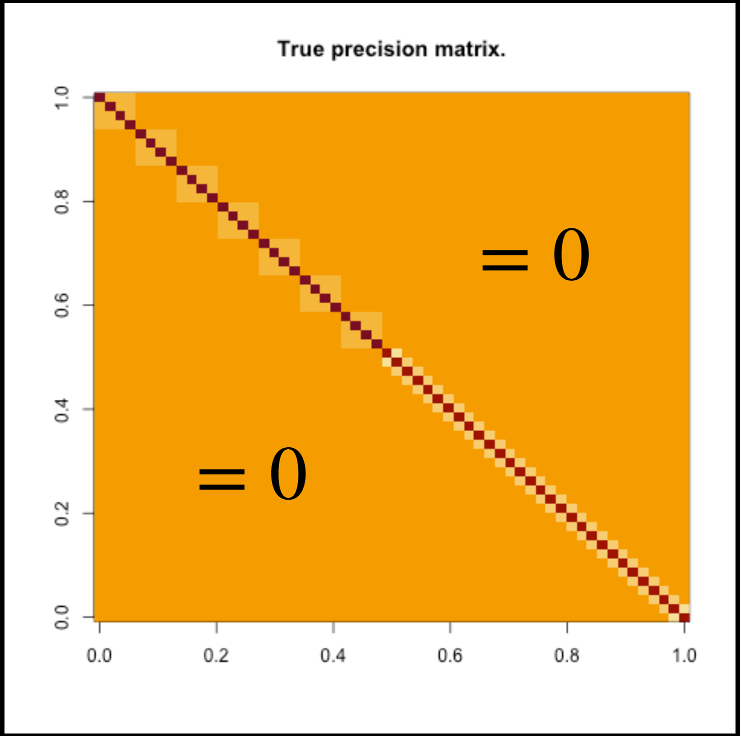


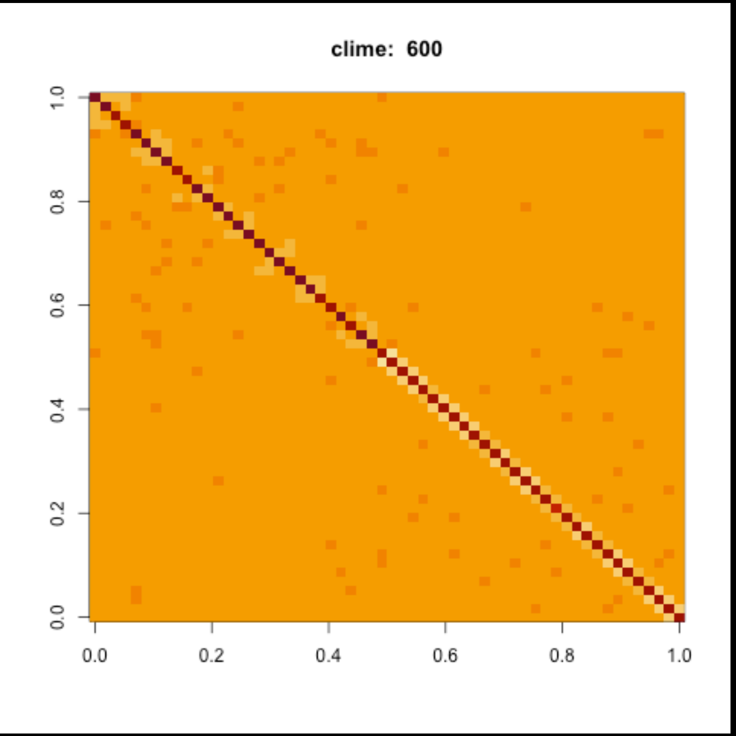
Precision: Path + Cliques
Max-degree = 3
Ill-conditioned


True precision matrix.
= 0
= 0
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

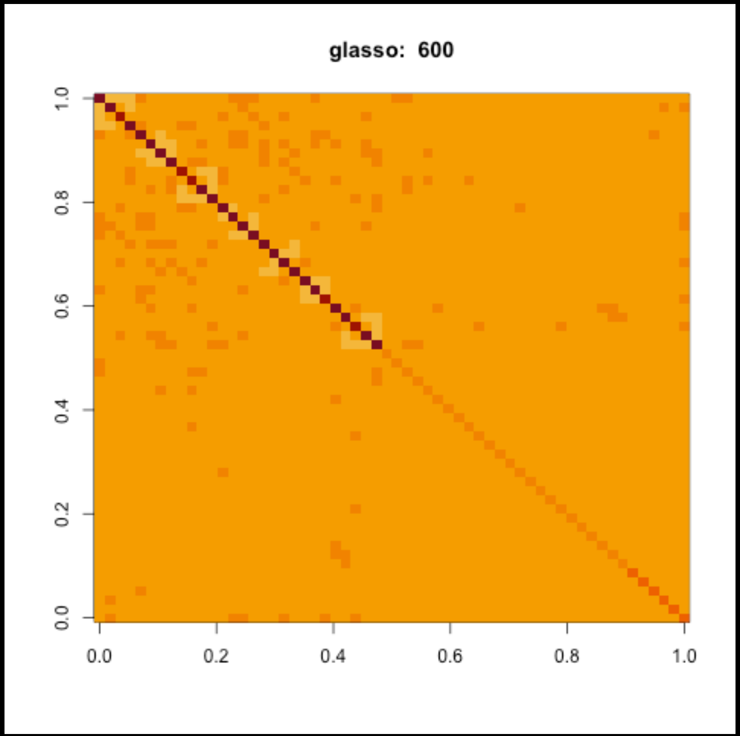

True precision matrix.
= 0
= 0
clime: 50
glasso: 50
greedy: 50
0.0
0.2
0.4
0.6
0.8
1.0

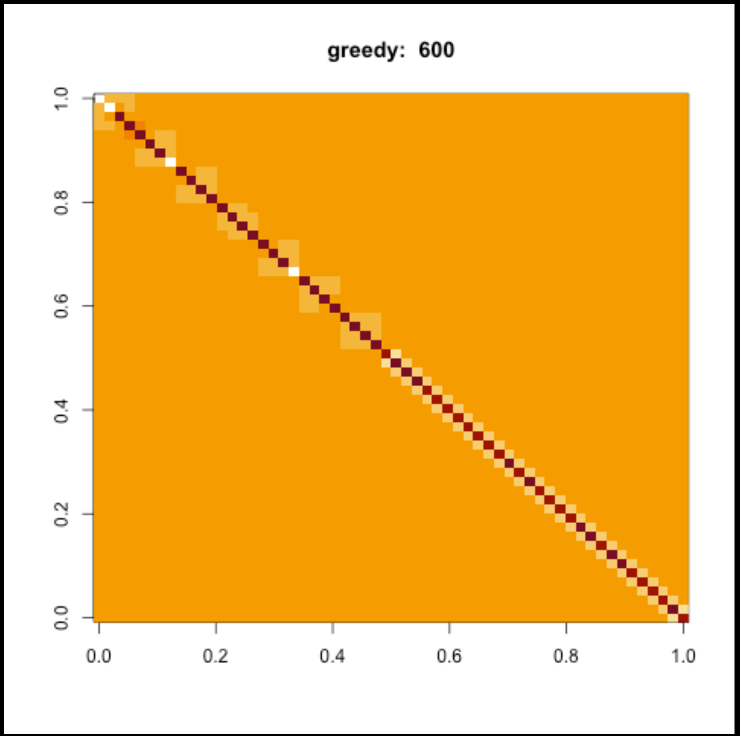

True precision matrix.
= 0
= 0
clime: 50
glasso: 50
greedy: 50
0.0
0.2
0.4
0.6
0.8
1.0



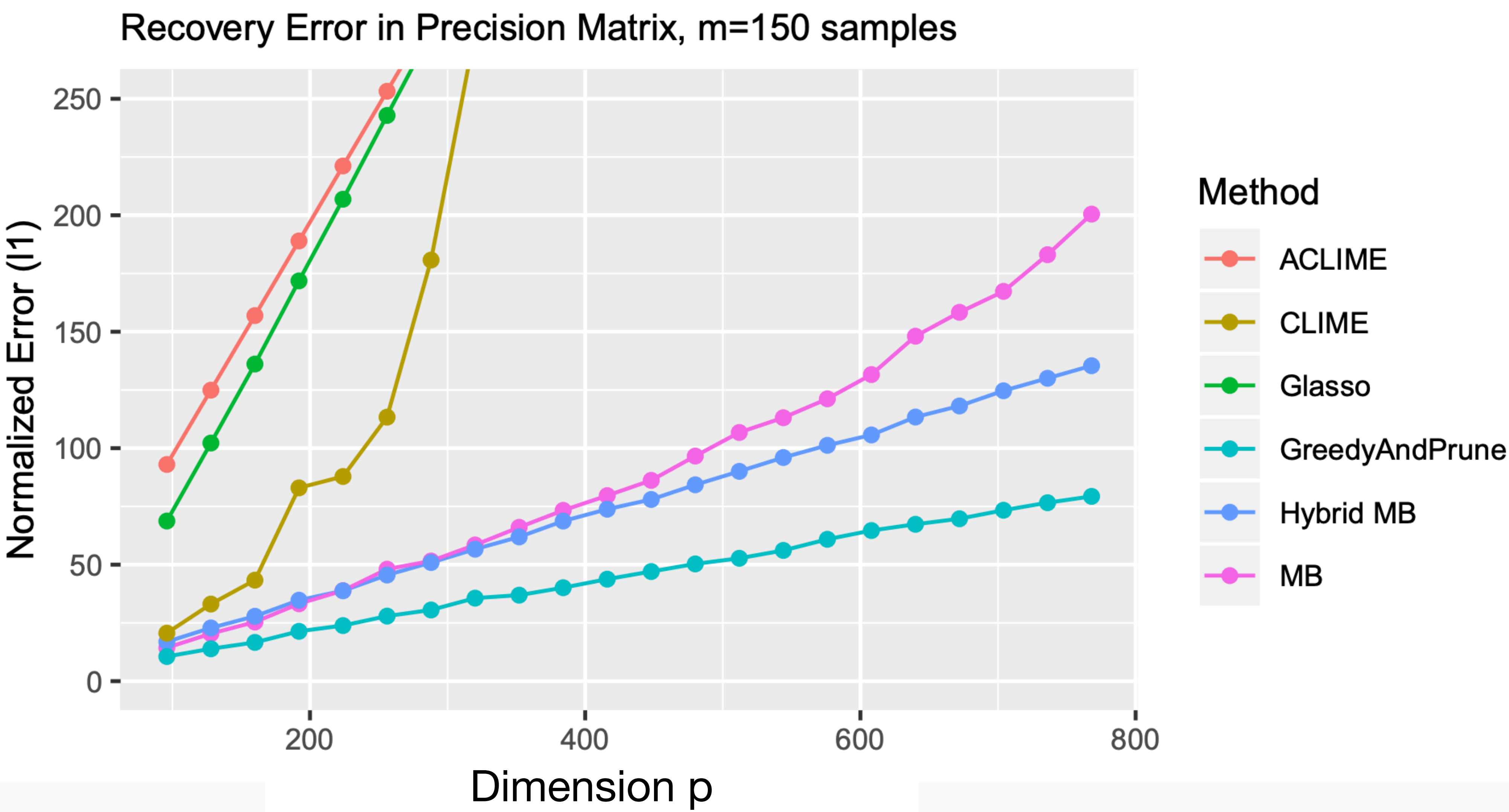


Final Outputs



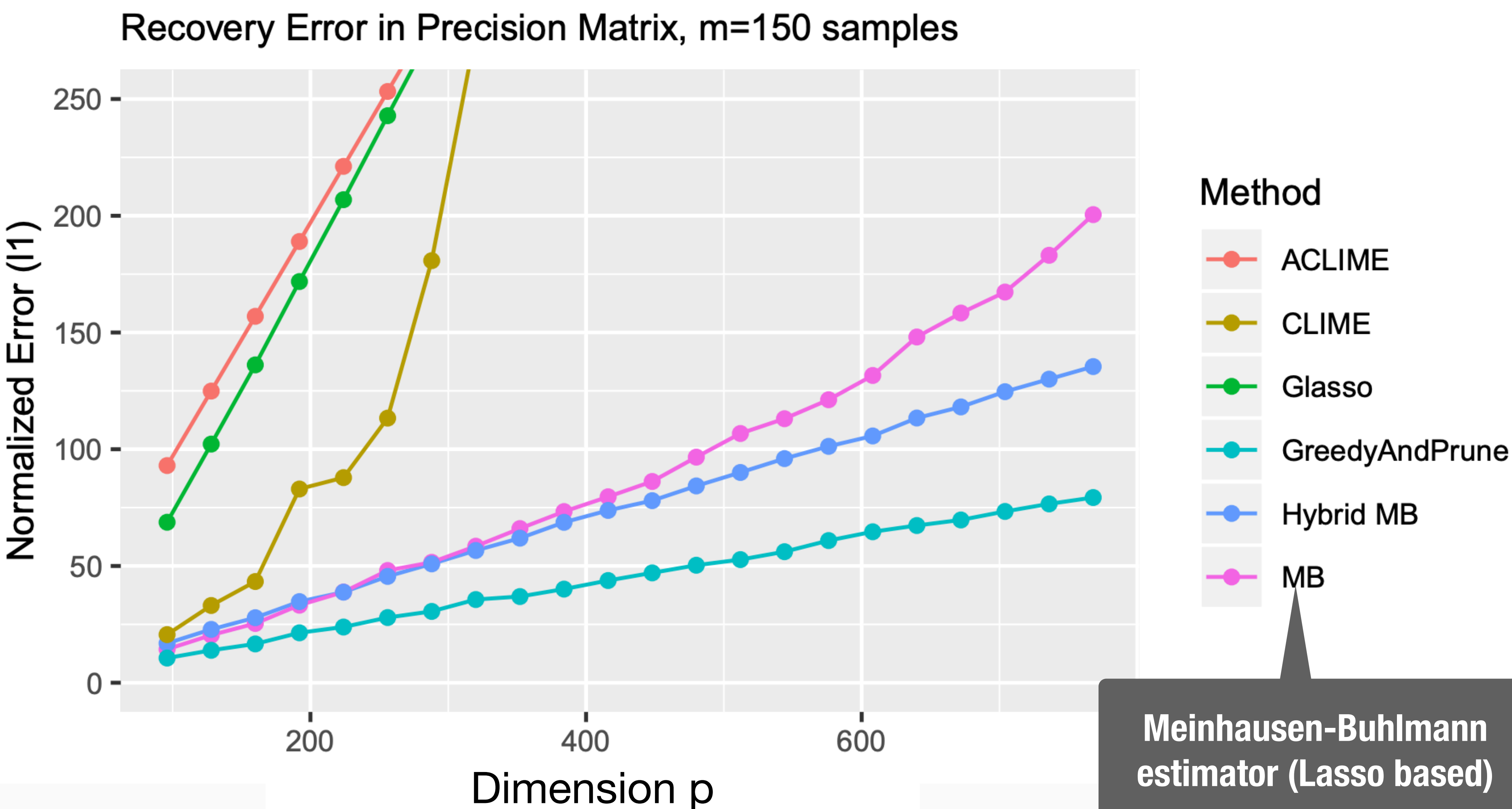


A Simple Challenge: Path + Clique



GreedyPrune has best error

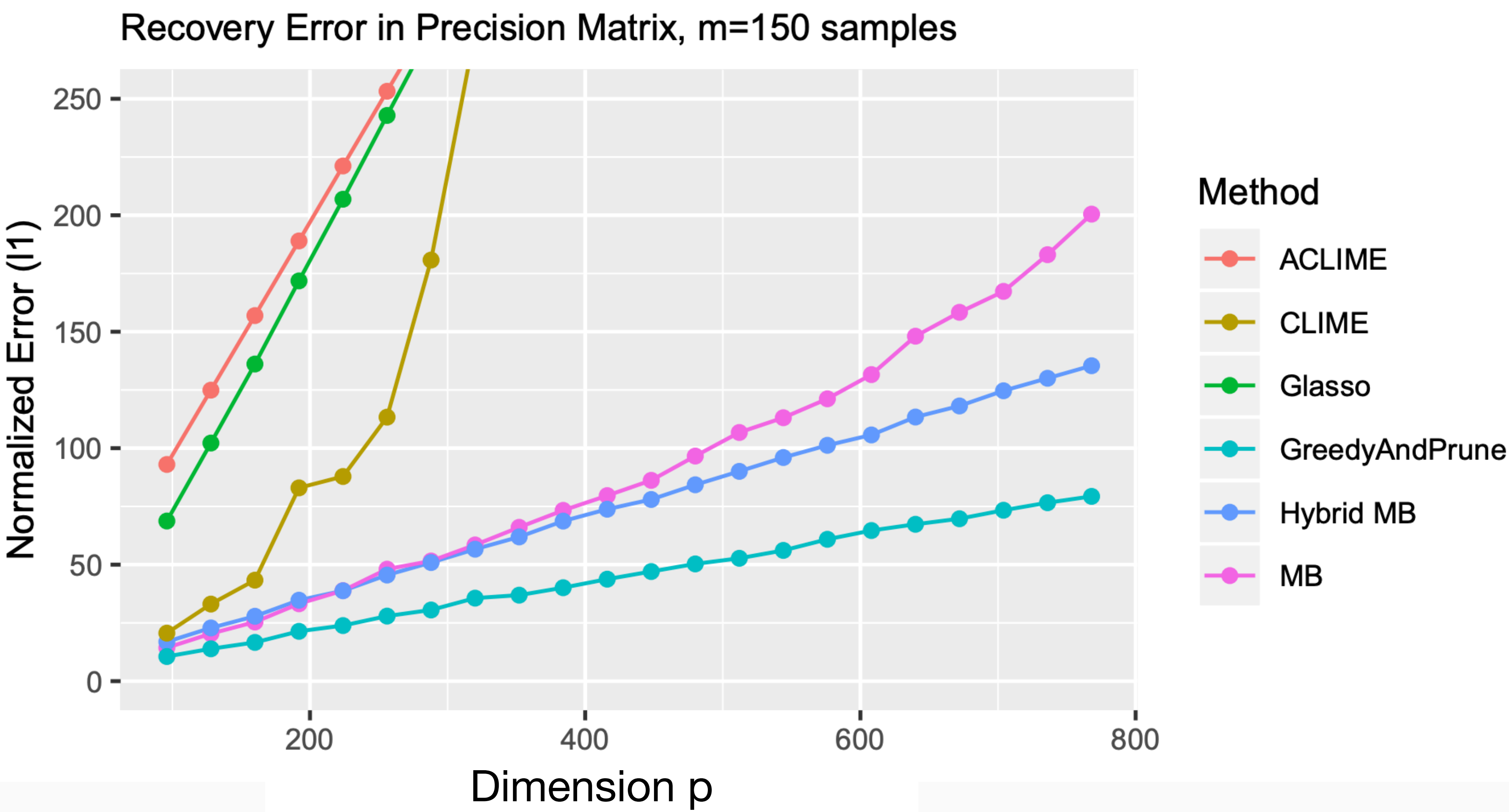
A Simple Challenge: Path + Clique



Meinhausen-Buhlmann estimator (Lasso based)

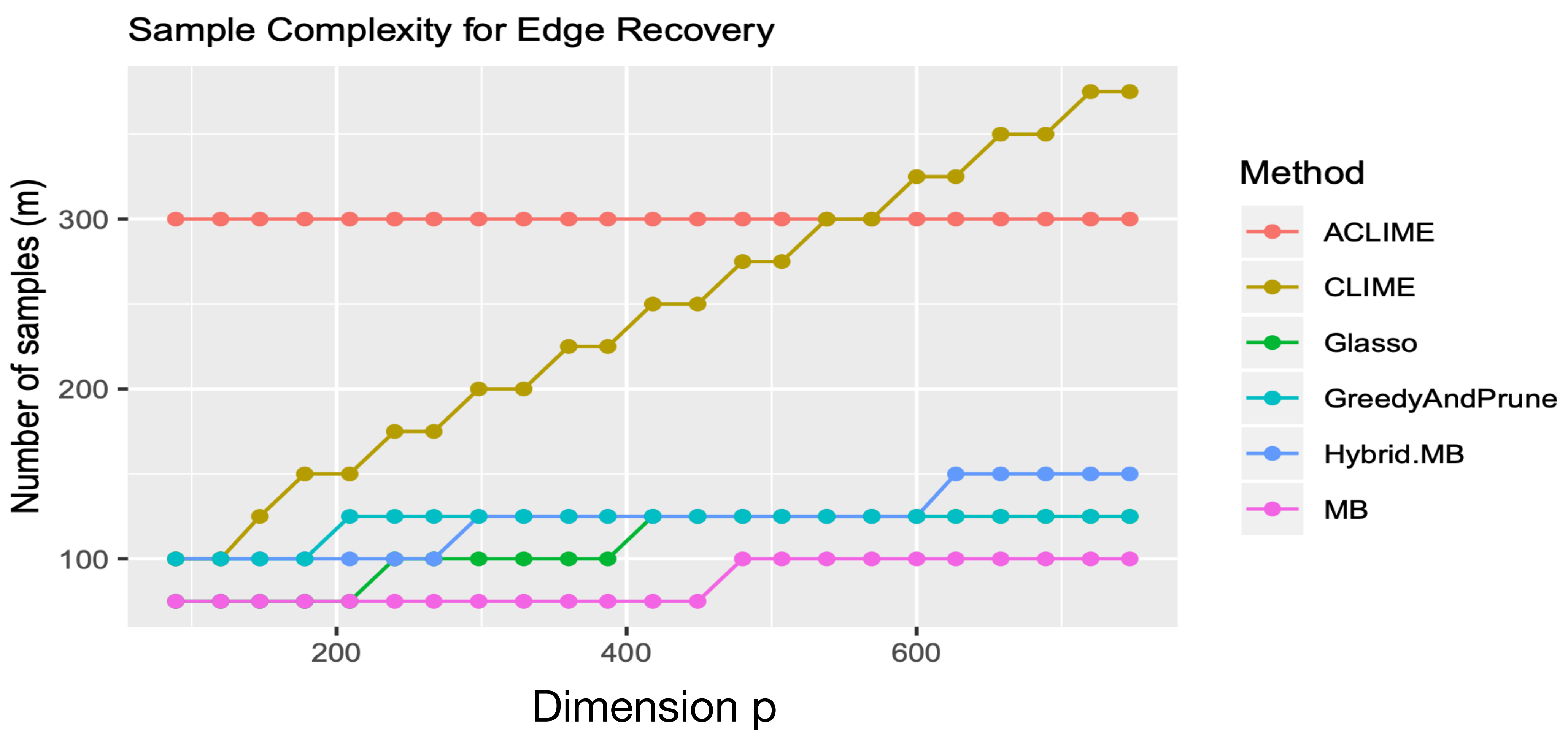
GreedyPrune has best error

A Simple Challenge: Path + Clique



GreedyPrune has best error

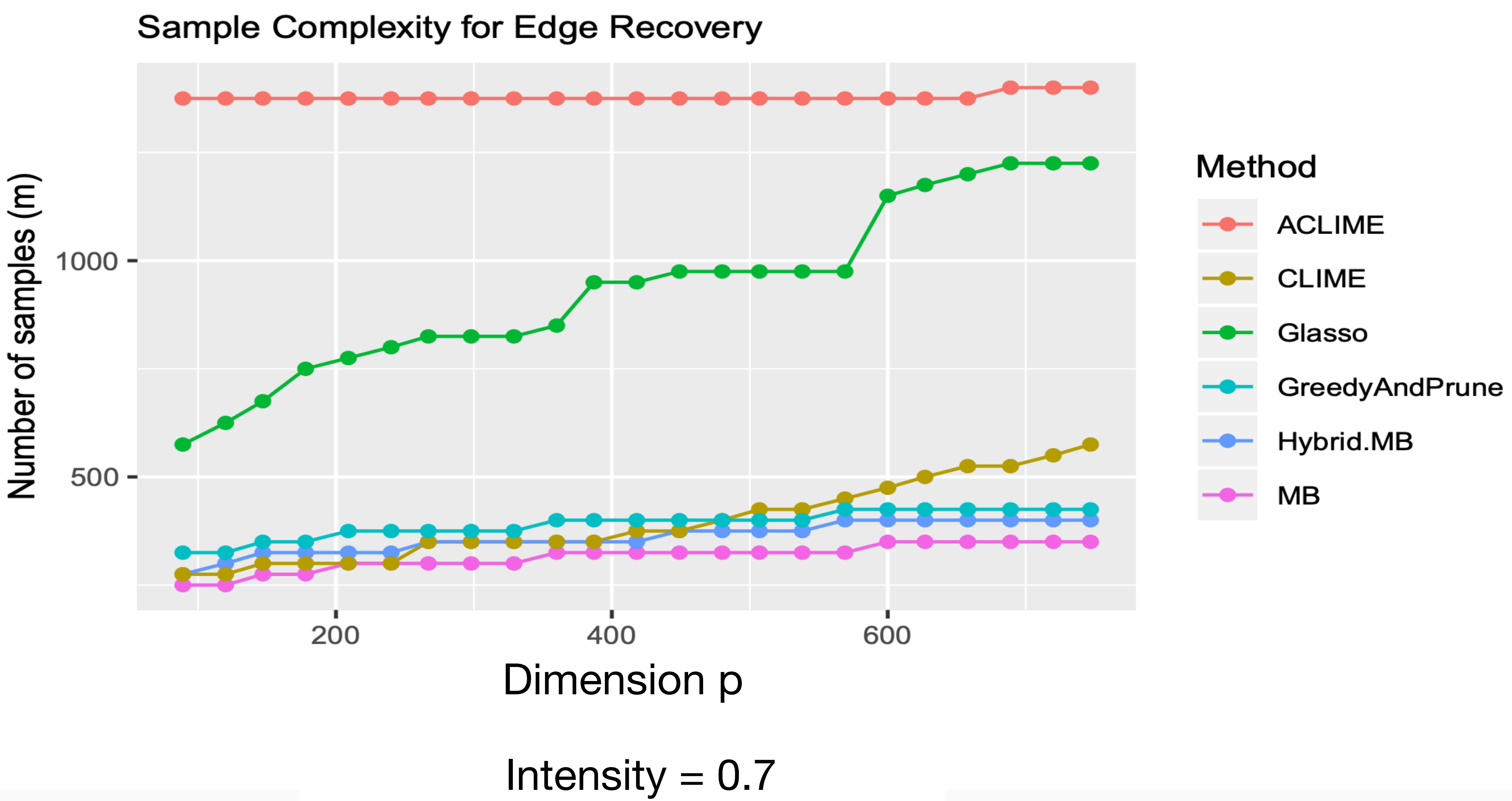
A Simple Challenge: Path + Clique



Intensity = 0.95

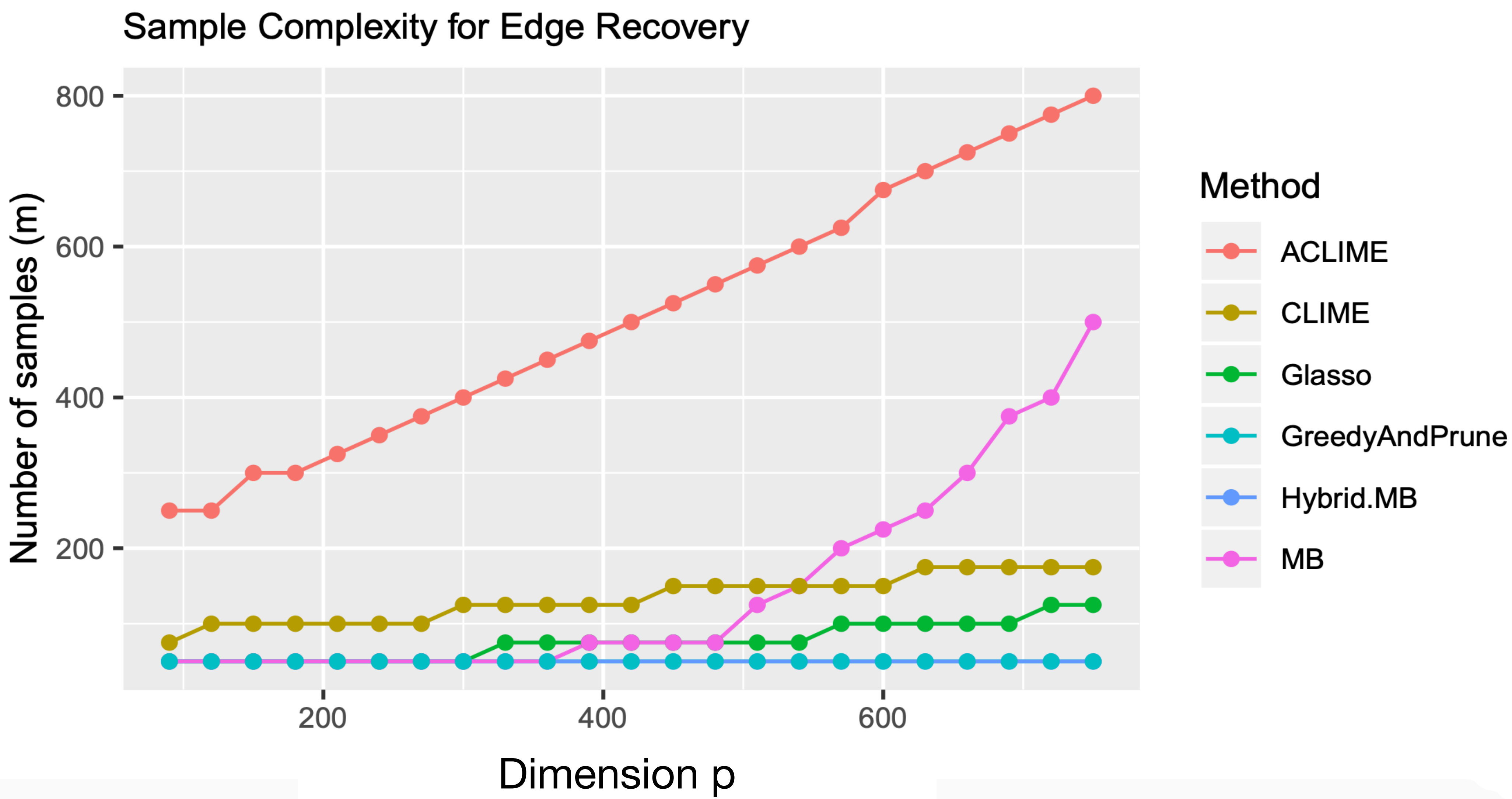
GreedyPrune needs very few samples.
CLIME grows nearly linearly …

A Simple Challenge: Path + Clique



GreedyPrune needs very few samples.
GLASSO grows nearly linearly …

A Simple Challenge: Random walk



GreedyPrune needs very few samples.
MB grows nearly linearly …

GreedyPrune Summary

KKMM: **GreedyPrune** learns attractive models with $\tilde{O}(d \log p/\kappa^2)$ samples and quadratic run-time.

KKMM: **GreedyPrune** learns walk-summable models with $O(d^2 \log p/\kappa^6)$ samples and quadratic run-time.

GreedyPrune Summary

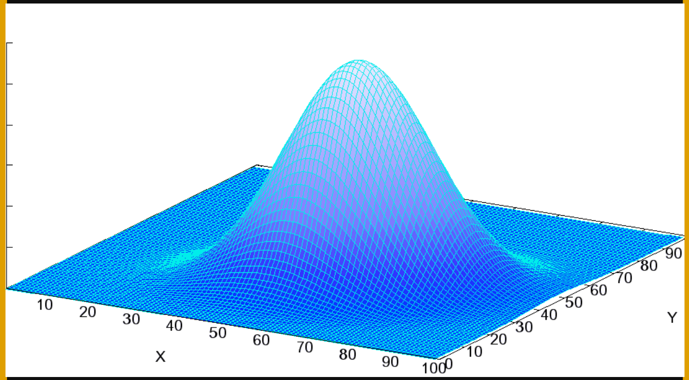
KKMM: **GreedyPrune** learns attractive models with $\tilde{O}(d \log p/\kappa^2)$ samples and quadratic run-time.

KKMM: **GreedyPrune** learns walk-summable models with $O(d^2 \log p/\kappa^6)$ samples and quadratic run-time.

Also …

- Recovers guarantees of GLASSO, CLIME
- Empirically better
- Non-Gaussian distributions: Can learn precision matrices if good tail behaviour

GGMs



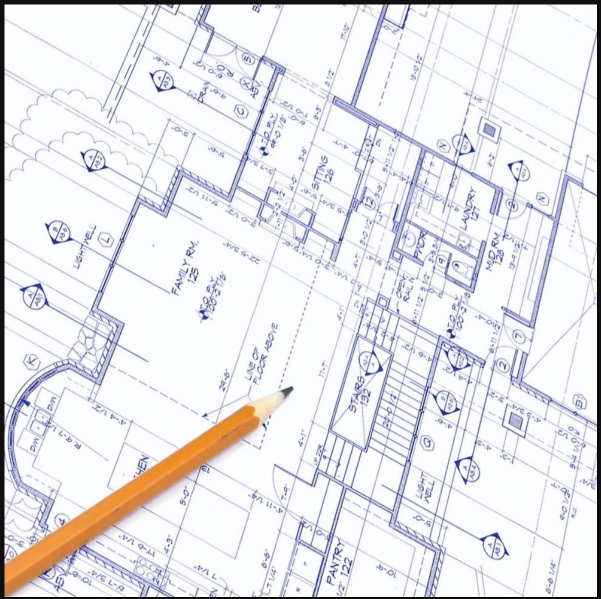
Motivation

Algorithm

While $|S| < t$:
Add $\arg\min_{j} Var(X_i \,|\, X_{S \cup j})$.

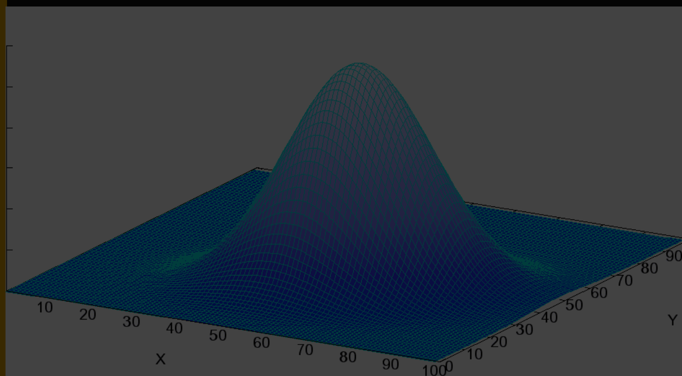
Special Models

Analysis



Attractive. SDD

GGMs



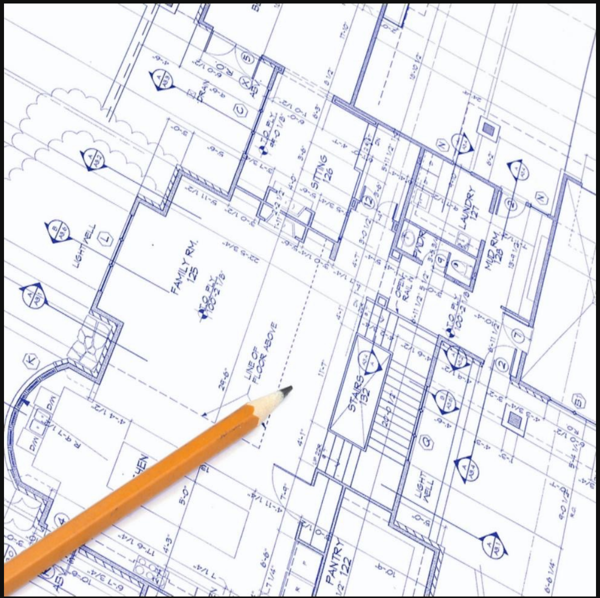
Motivation

Algorithm

While $|S| < t$:
Add $\arg\min_j Var(X_i | X_{S \cup j})$.

Special Models

Analysis



Attractive. SDD

Summary

Main Challenge: Can we recover structure of GGM of degree d, pairwise-correlation κ with run-time $p^{o(d)}$ and sample complexity $n \approx O_{d,\kappa}(poly(\log p))$?

Very well-stuided, practically important!

Summary

Main Challenge: Can we recover structure of GGM of degree d, pairwise-correlation κ with run-time $p^{o(d)}$ and sample complexity $n \approx O_{d,\kappa}(poly(\log p))$?

Very well-stuided, practically important!

Today: A simple greedy algorithm solves interesting special classes

Many Questions for GGMs …

- What other classes can we solve efficiently without condition number assumptions?
- Testing if a model is correct?
- Hidden variables?
- Computational hardness?

Many Questions for GGMs …

- What other classes can we solve efficiently without condition number assumptions?
- Testing if a model is correct?
- Hidden variables?
- Computational hardness?

Conjecture: No $p^{o(d)}$ algorithm to recover structure of general GGMs with $poly(d, 1/\kappa, \log p)$ samples.

Bigger Picture

Can we learn sparse dependency graphs from few samples?

(aka learning Markov random fields, undirected graphical models)

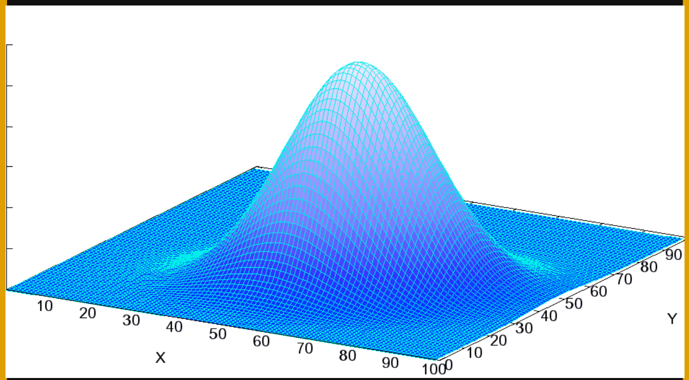
Bigger Picture

Can we learn sparse dependency graphs from few samples?

(aka learning Markov random fields, undirected graphical models)

Lots of work … [Bresler10], [KM17], [HKM17], …

GGMs



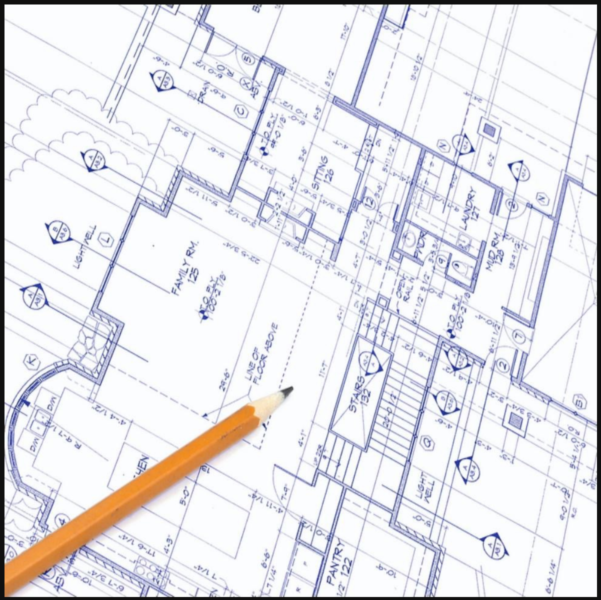
Motivation

Algorithm

While $|S| < t$:
Add $\arg\min_j Var(X_i \mid X_{S \cup j})$.

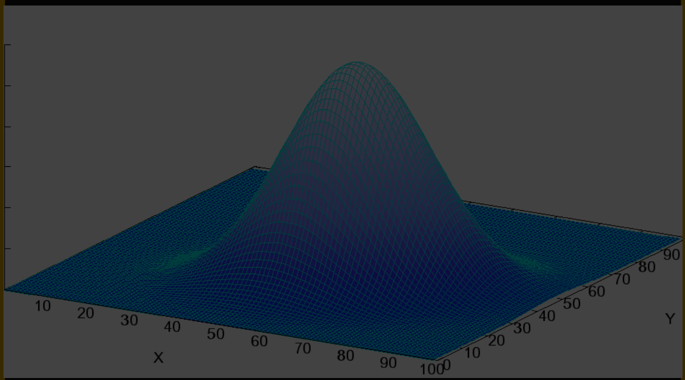
Special Models

Analysis



Attractive. SDD

GGMs



Motivation

Algorithm

While $|S| < t$:
Add $\arg\min_j Var(X_i \,|\, X_{S \cup j})$.

Special Models

Analysis

Attractive. SDD

Learning GGMs Greedily

Input: Samples from a sparse GGM ~ X.
Output: Dependency graph of X.

(d: max-degree; p: dimension;

n: num. samples; κ: pairwise correlation)

Learning GGMs Greedily

Input: Samples from a sparse GGM ~ X.
Output: Dependency graph of X.

(d: max-degree; p: dimension;

n: num. samples; κ: pairwise correlation)

Thm 1: `GreedyPrune` learns attractive models with $\tilde{O}(d \log p/\kappa^2)$ samples and quadratic run-time.

Learning GGMs Greedily

Input: Samples from a sparse GGM ~ X.
Output: Dependency graph of X.

(d: max-degree; p: dimension;

n: num. samples; κ: pairwise correlation)

Thm 1: **GreedyPrune** learns attractive models with $\tilde{O}(d \log p/\kappa^2)$ samples and quadratic run-time.

Thm 2: **GreedyPrune** learns walk-summable models with $O(d^2 \log p/\kappa^6)$ samples and quadratic run-time.

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2,\ldots,n\}$, define

$$f(S) = Var(X_1 \,|\, X_S)\,.$$

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2, \ldots, n\}$, define

$$f(S) = Var(X_1 \,|\, X_S) \,.$$

Claim [MJW06, MS12]: X is attractive GGM. Then, $f(\quad)$ is monotonically decreasing and a supermodular function.

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2, \ldots, n\}$, define

$$f(S) = Var(X_1 \,|\, X_S)\,.$$

Claim [MJW06, MS12]: X is attractive GGM. Then, $f(\ \ )$ is monotonically decreasing and a **supermodular** function.

Supermodular: $S \subset T$,

$$f(S) - f(S \cup \{j\}) \geq f(T) - f(T \cup \{j\})\,.$$

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2, \ldots, n\}$, define
$$f(S) = Var(X_1 \mid X_S)\,.$$

Claim [MS, MJW]: X is attractive GGM. Then, $f(\;)$ is monotonically decreasing and a **supermodular** function.

Why useful?

- Optimizing supermodularity very well understood
- Greedy algorithm finds minimizer

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2, \ldots, n\}$, define
$f(S) = Var(X_1 \,|\, X_S)$.

Claim [MS, MJW]: X is attractive GGM. Then, $f(\ \ )$ is monotonically decreasing and a **supermodular** function.

Are we done?

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2, \ldots, n\}$, define
$$f(S) = Var(X_1 \mid X_S) .$$

Claim [MS, MJW]: X is attractive GGM. Then, $f(\ \ )$ is monotonically decreasing and a **supermodular** function.

Are we done?

Almost …

- Only have estimates for f
- Crucial: missing a vertex means noticeably away from optimum

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2, \ldots, n\}$, define
$f(S) = Var(X_1 \,|\, X_S)\,.$

Claim [MS, MJW]: X is attractive GGM. Then, $f(\quad)$ is monotonically decreasing and a supermodular function.

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2, \ldots, n\}$, define
$f(S) = Var(X_1 \,|\, X_S)$.

Claim [MS, MJW]: X is attractive GGM. Then, $f(\;\;)$ is monotonically decreasing and a supermodular function.

Proof: …

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2,\ldots,n\}$, define
$f(S) = Var(X_1 \,|\, X_S)\,.$

Claim [MS, MJW]: X is attractive GGM. Then, $f(\ \ )$ is monotonically decreasing and a supermodular function.

Proof: …

- Assume $\Theta_{ii} = 1$.

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2, \ldots, n\}$, define
$f(S) = Var(X_1 \,|\, X_S)$.

Claim [MS, MJW]: X is attractive GGM. Then, $f(\ \ )$ is monotonically decreasing and a supermodular function.

Proof: …

- **Assume $\Theta_{ii} = 1$.**
- **Linear algebra:**

$$Var(X_1 \,|\, X_S) = \sum_{i_1, i_2, \ldots, i_k \notin S} (-\Theta_{ii_1})(-\Theta_{i_1 i_2}) \cdots (-\Theta_{i_k i})$$

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2,\ldots,n\}$, define
$f(S) = Var(X_1 \mid X_S)$.

Claim [MS, MJW]: X is attractive GGM. Then, $f(\;)$ is monotonically decreasing and a supermodular function.

Proof: …

-
-

Product over cycles not touching S …

$$Var(X_1 \mid X_S) = \sum_{i_1, i_2, \ldots, i_k \notin S} (-\Theta_{ii_1})(-\Theta_{i_1 i_2})\cdots(-\Theta_{i_k i})$$

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2, \ldots, n\}$, define
$f(S) = Var(X_1 \mid X_S)$.

Claim [MS, MJW]: X is attractive GGM. Then, $f(\ )$ is monotonically decreasing and a supermodular function.

Proof: …

-

-

Product over cycles not touching S …

$$Var(X_1 \mid X_S) = \sum_{i_1, i_2, \ldots, i_k \notin S} (-\Theta_{ii_1})(-\Theta_{i_1 i_2}) \cdots (-\Theta_{i_k i})$$

- Thus: $f(S) - f(S \cup \{j\})$ is sum on cycles that touch j but don't touch S.

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2,\ldots,n\}$, define
$f(S) = Var(X_1 \mid X_S)$.

Claim [MS, MJW]: X is attractive GGM. Then, $f(\ \ )$ is monotonically decreasing and a supermodular function.

Supermodular: $S \subset T$,
$f(S) - f(S \cup \{j\}) \geq f(T) - f(T \cup \{j\})$.

$$Var(X_1 \mid X_S) = \sum_{i_1, i_2, \ldots, i_k \notin S} (-\Theta_{ii_1})(-\Theta_{i_1 i_2}) \cdots (-\Theta_{i_k i})$$

- **Thus:** $f(S) - f(S \cup \{j\})$ **is sum on cycles that touch j but don't touch S.**

Analysis for Attractive: Supermodularity

Fix vertex 1. For $S \subseteq \{2,\ldots,n\}$, define
$f(S) = Var(X_1 \mid X_S)$.

Claim [MS, MJW]: X is attractive GGM. Then, $f(\ \ )$ is monotonically decreasing and a supermodular function.

Supermodular: $S \subset T$,
$f(S) - f(S \cup \{j\}) \geq f(T) - f(T \cup \{j\})$.

$$Var(X_1 \mid X_S) = \sum_{i_1,i_2,\ldots,i_k \notin S} (-\Theta_{ii_1})(-\Theta_{i_1 i_2})\cdots(-\Theta_{i_k i})$$

- Thus: $f(S) - f(S \cup \{j\})$ is sum on cycles that touch j but don't touch S.
- Is decreasing with S, so f is supermodular.

Analysis for Walk-Summable

Fix vertex 1. For $S \subseteq \{2, \ldots, n\}$, define

$$f(S) = Var(X_1 \,|\, X_S)\,.$$

No longer supermodular!

(Not even weakly-supermodular …)

Analysis for Walk-Summable

Fix vertex 1. For $S \subseteq \{2, \ldots, n\}$, define
$$f(S) = Var(X_1 \,|\, X_S)\,.$$

No longer supermodular!
(Not even weakly-supermodular …)

- Walk-summable = Symmetric Diagonally Dominant (SDD)
- SDD implies neighbors have noticeable effect

Analysis for Walk-Summable

Def: Θ **is SDD if for all i**

$$|\Theta_{ii}| \geq \sum_{j \neq i} |\Theta_{ij}|$$

- **Walk-summable = Symmetric Diagonally Dominant (SDD)**
- **SDD implies neighbors have noticeable effect**

Analysis for Walk-Summable

Def: Θ **is SDD if for all i**

$$|\Theta_{ii}| \geq \sum_{j \neq i} |\Theta_{ij}|$$

Claim [RTT09]: Θ **walk-summable iff a diagonal** $D \geq 0$, $D\Theta D$ **is SDD.**

- Walk-summable = Symmetric Diagonally Dominant (SDD)
- SDD implies neighbors have noticeable effect

Analysis for Walk-Summable

Def: Θ **is SDD if for all i**

$$|\Theta_{ii}| \geq \sum_{j \neq i} |\Theta_{ij}|$$

Claim [RTT09]: Θ **walk-summable iff a diagonal** $D \geq 0$, $D\Theta D$ **is SDD.**

Suffices to show greedy works for SDD!

- Walk-summable = Symmetric Diagonally Dominant (SDD)
- SDD implies neighbors have noticeable effect

Analysis: Bounded Conditional Variances

Lemma: If Θ is SDD, then for any vertex i, there exists a neighbor j such that $Var(X_i \mid X_j) \leq 4Var(X_i \mid X_{-i})/\kappa^2$.

Analysis: Bounded Conditional Variances

There exists a neighbor, that is comparable to the entire neighborhood in conditioning …

Lemma: If Θ is SDD, then for any vertex i, there exists a neighbor j such that $Var(X_i \mid X_j) \leq 4Var(X_i \mid X_{-i})/\kappa^2$.

Analysis: Bounded Conditional Variances

Why useful …

- $Var(X_i)$ can be very large!
- But conditioning on one neighbor brings the variance down.
- We can detect it with few samples.

Lemma: If Θ is SDD, then for any vertex i, there exists a neighbor j such that $Var(X_i \mid X_j) \leq 4Var(X_i \mid X_{-i})/\kappa^2$.

Analysis: Bounded Conditional Variances

Proof …

- Triangle inequality of **effective resistance** metric on graphs
- Properties of conditional variance …

Lemma: If Θ is SDD, then for any vertex i, there exists a neighbor j such that $Var(X_i | X_j) \leq 4Var(X_i | X_{-i})/\kappa^2$.

Analysis: Bounded Conditional Variances

For intuition: Assume $\Theta_{ii} = 1, \forall i$.

Recall: $\kappa(\Theta) = \min\limits_{i,j:\Theta_{ij}\neq 0} \dfrac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$.

So $i \sim j \Rightarrow |\Theta_{ij}| \geq \kappa$.

Analysis: Bounded Conditional Variances

For intuition: Assume $\Theta_{ii} = 1, \forall i$.

Recall: $\kappa(\Theta) = \min_{i,j:\Theta_{ij}\neq 0} \dfrac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$.

So $i \sim j \Rightarrow |\Theta_{ij}| \geq \kappa$.

Lemma 1: If Θ is SDD, then conditional variances are bounded — $i \sim j \Rightarrow Var(X_i | X_j) \leq 1/\kappa$.

Analysis: Bounded Conditional Variances

For intuition: Assume $\Theta_{ii} = 1, \forall i$.

Recall: $\kappa(\Theta) = \min_{i,j:\Theta_{ij}\neq 0} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$.

So $i \sim j \Rightarrow |\Theta_{ij}| \geq \kappa$.

Lemma 1: If Θ is SDD, then conditional variances are bounded — $i \sim j \Rightarrow Var(X_i | X_j) \leq 1/\kappa$.

Analysis: Bounded Conditional Variances

For intuition: Assume $\Theta_{ii} = 1, \forall i$.

Recall: $\kappa(\Theta) = \min_{i,j:\Theta_{ij}\neq 0} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$.

So $i \sim j \Rightarrow |\Theta_{ij}| \geq \kappa$.

Lemma 1: If Θ is SDD, then conditional variances are bounded — $i \sim j \Rightarrow Var(X_i | X_j) \leq 1/\kappa$.

Example: In random walk model

$Var(X_{i+1} | X_i) = 1; \quad Var(X_{i+1}) = i + 1!$

Analysis: Bounded Conditional Variances

For intuition: Assume $\Theta_{ii} = 1, \forall i$.

Recall: $\kappa(\Theta) = \min_{i,j:\Theta_{ij}\neq 0} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$.

So $i \sim j \Rightarrow |\Theta_{ij}| \geq \kappa$.

Lemma 1: If Θ is SDD, then conditional variances are bounded — $i \sim j \Rightarrow Var(X_i | X_j) \leq 1/\kappa$.

Not true for general sparse precison matrices …

Analysis: Bounded Conditional Variances

For intuition: Assume $\Theta_{ii} = 1, \forall i$.

Recall: $\kappa(\Theta) = \min_{i,j:\Theta_{ij}\neq 0} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$.

So $i \sim j \Rightarrow |\Theta_{ij}| \geq \kappa$.

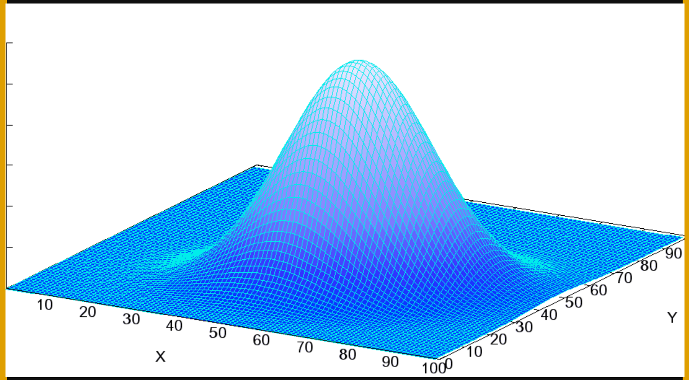
Lemma 1: If Θ is SDD, then conditional variances are bounded — $i \sim j \Rightarrow Var(X_i | X_j) \leq 1/\kappa$.

"Proof …":

- **If Laplacian …**

$$Var(X_i | X_j) = \frac{1}{2} R_{eff}(i, j) \leq \frac{1}{|\Theta_{ij}|} \leq 1/\kappa .$$

GGMs



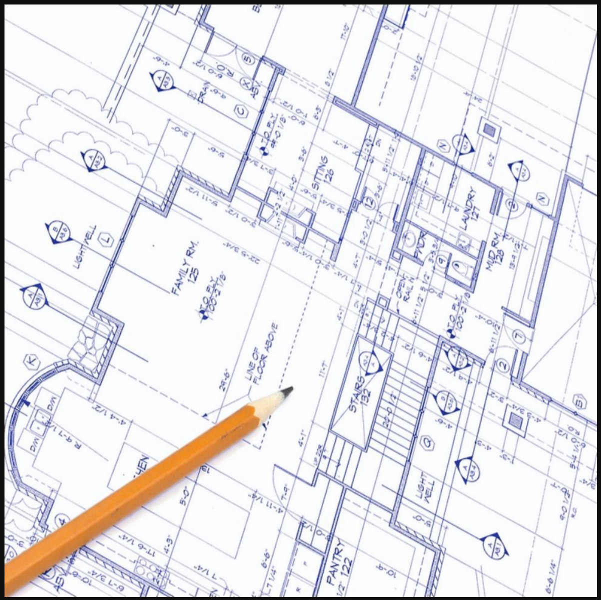
Motivation

Algorithm

While $|S| < t$:
Add $\arg\min_j Var(X_i | X_{S \cup j})$.

Special Models

Analysis



Attractive. SDD