Semi-blind Subgraph Reconstruction in Gaussian Graphical Models

†**Tianpei Xie**, ⋆Sijia Liu, ⋆Alfred O. Hero

†Transaction Risk Management Team @ Amazon [1]
⋆University of Michigan, Ann Arbor

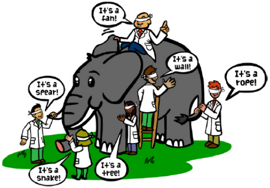
[1] This work was completed while Tianpei Xie was a Phd student at University of Michigan

Backgrounds

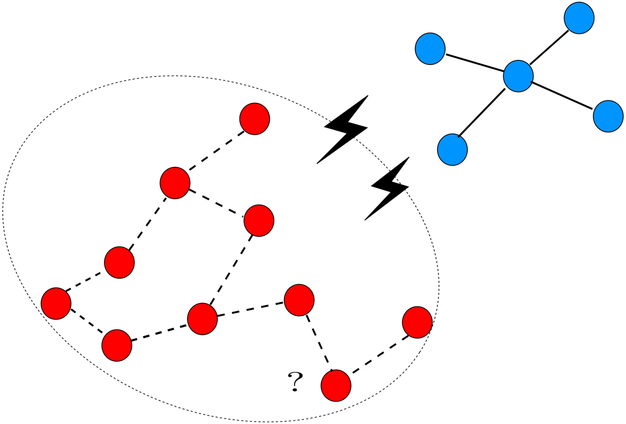
- Learning a dependency graph from relational data is a key step in data visualization and analysis. Examples include
  1. recommendation system
  2. social network analysis [Goyal et al., 2010]
  3. sensor network analysis [Joshi and Boyd, 2009, Liu et al., 2016]

- However, in many situations, only a limited set of data is accessible, due to
  - the limited budgets during data collections (e.g. labor, energy)
  - the restricted accessibility to data sources (e.g. data security, privacy)

- **Semi-blinded subgraph topology learning problem**: only see data on a subgraph but blind to the rest.



Semi-blinded subgraph topology learning problem



Challenges

- Challenges:
  - The influence of external latent data ⇒ the target network ⇒ bias in inference

    probabilistic models: ***marginalization*** ⇒ *false positives* in edge detection

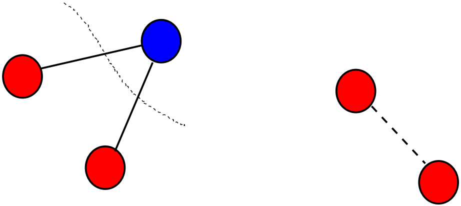


Figure: *The red nodes are conditional independent given the blue node. After marginalizing the blue node, it creates a false connection in the graph*

- Assumption: additional information from **external** sources ⇒ summary info. of latent data

Settings

- Random graph signal $\boldsymbol{x} \in \mathbb{R}^n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Theta}^{-1})$, Markovian w.r.t. $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$
  $\Rightarrow x_i \perp\!\!\!\perp x_j | \boldsymbol{x}_{-\{i,j\}} \Leftrightarrow \boldsymbol{\Theta}_{i,j} = 0$ iff $(i, j) \notin \mathcal{E}$.
- Consider
  - partition of $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$, non-overlapping, $|\mathcal{V}_1| = n_1$, $|\mathcal{V}_2| = n_2$, edge set between $\mathcal{V}_1, \mathcal{V}_2$ denoted as $\mathcal{E}_{1,2}$
  - accessible $\boldsymbol{x}_1 := \boldsymbol{x}_{\mathcal{V}_1} \in \mathbb{R}^{n_1}$, inaccessible *(latent)* $\boldsymbol{x}_2 := \boldsymbol{x}_{\mathcal{V}_2} \in \mathbb{R}^{n_2}$
  - precision matrix $\boldsymbol{\Theta} := \begin{bmatrix} \boldsymbol{\Theta}_1 & \boldsymbol{\Theta}_{12} \\ \boldsymbol{\Theta}_{21} & \boldsymbol{\Theta}_2 \end{bmatrix}$
- $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$, **target network**; $\mathcal{E}_1 := \mathcal{E} \cap (\mathcal{V}_1 \times \mathcal{V}_1) \Leftrightarrow \boldsymbol{\Theta}_1$

Settings

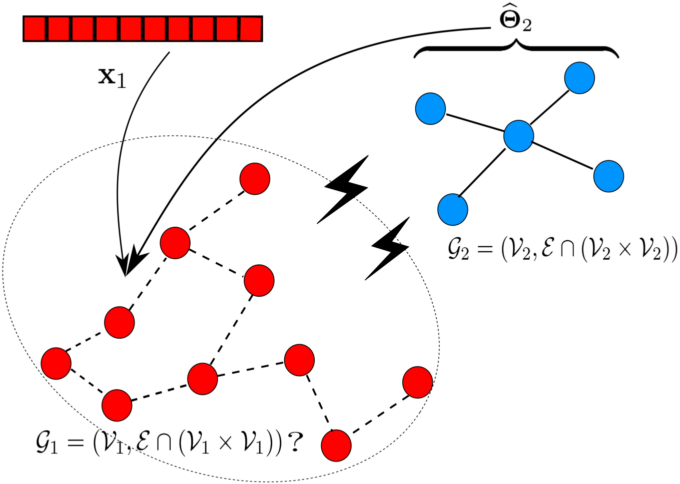
- Random graph signal $\boldsymbol{x} \in \mathbb{R}^n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Theta}^{-1})$, Markovian w.r.t. $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$
  $\Rightarrow x_i \perp\!\!\!\perp x_j | \boldsymbol{x}_{-\{i,j\}} \Leftrightarrow \boldsymbol{\Theta}_{i,j} = 0$ iff $(i, j) \notin \mathcal{E}$.

- Consider
  - partition of $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$, non-overlapping, $|\mathcal{V}_1| = n_1$, $|\mathcal{V}_2| = n_2$, edge set between $\mathcal{V}_1, \mathcal{V}_2$ denoted as $\mathcal{E}_{1,2}$
  - accessible $\boldsymbol{x}_1 := \boldsymbol{x}_{\mathcal{V}_1} \in \mathbb{R}^{n_1}$, inaccessible *(latent)* $\boldsymbol{x}_2 := \boldsymbol{x}_{\mathcal{V}_2} \in \mathbb{R}^{n_2}$
  - precision matrix $\boldsymbol{\Theta} := \begin{bmatrix} \boldsymbol{\Theta}_1 & \boldsymbol{\Theta}_{12} \\ \boldsymbol{\Theta}_{21} & \boldsymbol{\Theta}_2 \end{bmatrix}$

- $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$, **target network**; $\mathcal{E}_1 := \mathcal{E} \cap (\mathcal{V}_1 \times \mathcal{V}_1) \Leftrightarrow \boldsymbol{\Theta}_1$

- **Goal**: estimate $\boldsymbol{\Theta}_1$, given
  1. accessible data on $\mathcal{V}_1$, $\boldsymbol{x}_1 \Rightarrow \widehat{\boldsymbol{\Sigma}}_1 := \widehat{\mathbb{E}}[\boldsymbol{x}_1 \boldsymbol{x}_1^T]$, sample marginal covariance
  2. a noisy summary $\widehat{\boldsymbol{\Theta}}_2 \in \mathbb{R}^{n_2 \times n_2}$ of inverse covariance of $\boldsymbol{x}_2$, shared by external sources

Network topology learning from partially shared information



Related works

- w/o latent variables, many algorithms to estimate $\boldsymbol{\Theta}$ of Gaussian graphical model. e.g.
  1. ℓ_1 regularized ML, such as **gLasso**, [Friedman et al., 2008]
  2. quadratic approximation, **QUIC**, [Hsieh et al., 2011]
  3. ℓ_0 regularized ML, [Marjanovic and Hero, 2015]

Related works

- w/o latent variables, many algorithms to estimate $\boldsymbol{\Theta}$ of Gaussian graphical model. e.g.
  1. ℓ_1 regularized ML, such as **gLasso**, [Friedman et al., 2008]
  2. quadratic approximation, **QUIC**, [Hsieh et al., 2011]
  3. ℓ_0 regularized ML, [Marjanovic and Hero, 2015]

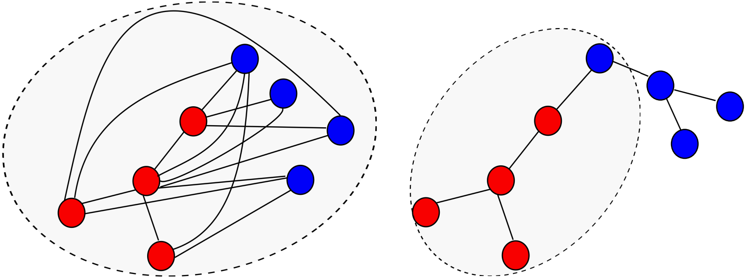
- w/ latent variables, to estimate sub-matrix $\boldsymbol{\Theta}_1$ of full precision $\boldsymbol{\Theta}$
  1. the *latent variable Gaussian graphical model* (**LV-GGM**) by [Chandrasekaran et al., 2012]
  2. **Key**

$$\widetilde{\boldsymbol{\Theta}}_1 := (\boldsymbol{\Sigma}_1)^{-1} = \underbrace{\boldsymbol{\Theta}_1}_{\text{sparse}} - \underbrace{\boldsymbol{\Theta}_{12}(\boldsymbol{\Theta}_2)^{-1}\boldsymbol{\Theta}_{21}}_{\text{low-rank}}$$

$$:= \boldsymbol{C} - \boldsymbol{M} \Rightarrow \text{signal} + \text{confounding factor}$$

  3. Disadvantages
     - the effect of latent variables is ***uniform and global***, not change during propagation
     - does not exploit the ***dependency structure*** among latent variables

Global influence model vs. decayed influence model



(a) Global influence by LV-GGM (b) **Decayed-influence latent variable model**

- $\mathcal{E}_{21}$ *dense*
- no edge among nodes in $\mathcal{V}_2$, i.e. $\boldsymbol{x}_2$ cond. indep given $\boldsymbol{x}_1$

- $\mathcal{E}_{21}$ *sparse*
- edges among nodes in $\mathcal{V}_2$, i.e. cond. **dep** $\sim \widehat{\Theta}_2$

Our contributions

- Propose the *decayed-influence latent variable Gaussian graphical model* **(DiLat-GGM)** that
  1. takes into account the decayed influence effect during the propagation of info.
  2. fully utilizes the shared **dependency** information from external sources
  3. latent variable inference and selection

LV-GGM vs. DiLat-GGM

	LV-GGM	**DiLat-GGM**
variables	$C \in \mathbb{R}^{n_1 \times n_1}$, $M \in \mathbb{R}^{n_1 \times n_1}$	$C \in \mathbb{R}^{n_1 \times n_1}$ $B := \Theta_{12}\Theta_2^{-1} \in \mathbb{R}^{n_1 \times n_2}$
known	$\widehat{\Sigma}_1, \alpha, \beta$	$\widehat{\Sigma}_1, \alpha, \beta,$ $\widehat{\Theta}_2 \succ \mathbf{0} \in \mathbb{R}^{n_2 \times n_2}$
constraint	$\widetilde{\Theta}_1 = C - M \succeq \mathbf{0}$	$\widetilde{\Theta}_1 = C - B\widehat{\Theta}_2 B^T \succeq \mathbf{0}$
key	$M \succeq \mathbf{0}$, low-rank	$\Theta_{21} = \widehat{\Theta}_2 B^T = \begin{bmatrix} \mathbf{0} \\ \Theta_{\delta\mathcal{V}_2,1} \end{bmatrix}$, row-sparse
infer. on latent var	No	**Yes**. $p(x_2 \mid x_1) = \mathcal{N}(\mu_{2\mid 1}, \widehat{\Theta}_2)$, $\mu_{2\mid 1} = B^T x_1$
latent feat. sel.	No	**Yes**.
convexity	**Yes**	No
implemt.	ADMM	Convex-concave procedure **(CCP)** + ADMM

The decayed-influence latent variable Gaussian graphical model

The proposed **DiLat-GGM** solves the following

$$\min_{\boldsymbol{C},\boldsymbol{B}} \; -\log\det\left(\boldsymbol{C}-\boldsymbol{B}\widehat{\Theta}_2\boldsymbol{B}^T\right)+\mathrm{tr}\left(\widehat{\Sigma}_1\left(\boldsymbol{C}-\boldsymbol{B}\widehat{\Theta}_2\boldsymbol{B}^T\right)\right)+\underbrace{\alpha_m\left\|\boldsymbol{C}\right\|_1}_{\text{sparsity of cond. graph}}$$

$$+\underbrace{\beta_m\left\|\widehat{\Theta}_2\boldsymbol{B}^T\right\|_{2,1}}_{\text{sparsity of }\mathcal{E}_{cross}\text{ \& latent feat. sel.}}$$

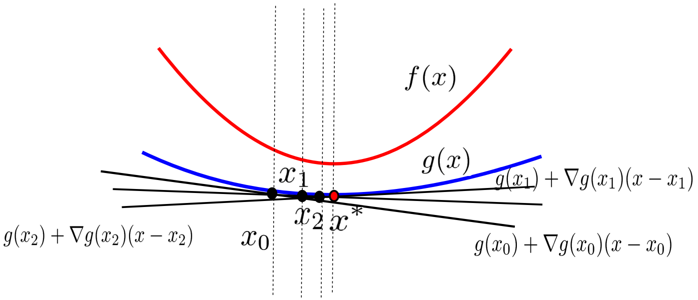
$$\text{s.t.} \quad \boldsymbol{C}-\boldsymbol{B}\widehat{\Theta}_2\boldsymbol{B}^T\succeq\boldsymbol{0},$$

where

- $\left\|\widehat{\Theta}_2\boldsymbol{B}^T\right\|_{2,1} := \sum_{i\in\mathcal{V}_2}\left\|[\widehat{\Theta}_2\boldsymbol{B}^T]_i\right\|_2$ is the mixed ℓ_{21} norm.
- An external source provides $\widehat{\Theta}_2 \Rightarrow$ partial corr. of $\boldsymbol{x}_2$
- DiLat-GGM is a Difference-of-Convex program and can be solved via ***convex-concave procedure*** (**CCP**) [Yuille et al., 2002, Lipp and Boyd, 2016]

The convex-concave procedure

- Example: find $x^* = \mathrm{argmin}(f(x) - g(x))$.



- Iteratively solve for $x_t := \mathrm{argmin}(f(x) - g(x_{t-1}) - \nabla g(x_{t-1})(x - x_{t-1}))$
- For DiLat-GGM, $g(\boldsymbol{B}) = \mathrm{tr}\left(\widehat{\Sigma}_1 \boldsymbol{B} \widehat{\Theta}_2 \boldsymbol{B}\right)^T$, the rest is $f(\cdot)$.

Experiments

- Compare algorithms:
  - **DiLat-GGM**
  - **GLasso** [Friedman et al., 2008]
  - **LV-GGM** [Chandrasekaran et al., 2012]
  - **EM-GLasso** [Yuan, 2012].
  - Generalized Laplacian learning (**GenLap**) [Pavez and Ortega, 2016]

- m i.i.d realizations of $\boldsymbol{x} = [x_1, \ldots, x_n]$. $m = 400$.

- Three types of graphs:
  1. The complete binary tree (h :=height)
  2. The grid (w :=width, h :=height)
  3. The Erdős-Rényi (n, p)

Experiments

- Compare algorithms:
  - **DiLat-GGM**
  - **GLasso** [Friedman et al., 2008]
  - **LV-GGM** [Chandrasekaran et al., 2012]
  - **EM-GLasso** [Yuan, 2012].
  - Generalized Laplacian learning (**GenLap**) [Pavez and Ortega, 2016]

- m i.i.d realizations of $\boldsymbol{x} = [x_1, \ldots, x_n]$. $m = 400$.

- Three types of graphs:
  1. The complete binary tree (h :=height)
  2. The grid (w :=width, h :=height)
  3. The Erdős-Rényi (n, p)

- The **Jaccard distance error** [Jaccard, 1901, Choi et al., 2010] for edge selection: between two sets A, B as

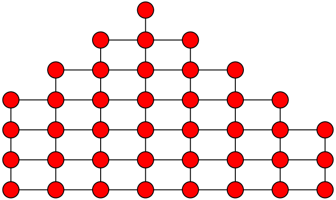
$$dist_J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|} \in [0, 1].$$

  1. $A :=$ non-zero support set of estimated $\widehat{\boldsymbol{\Theta}}_1$
  2. $B := \mathcal{E}_1$, the ground true edge set

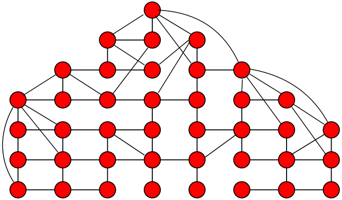
Comparison of mean edge selection error

Mean Jaccard distance error (×100%)					
Network	GLasso	EM-GLasso	GenLap	LV-GGM	DiLat-GGM
complete binary tree ($h = 3, n_1 = 10$)	55.7	65.2	**12.8**	36.4	18.8
complete binary tree ($h = 4, n_1 = 17$)	11.3	32.1	22.4	3.5	**2.2**
complete binary tree ($h = 5, n_1 = 36$)	15.0	26.6	50.9	3.3	**2.5**
grid ($w = 5, h = 5, n_1 = 15$)	39.3	40.7	**5.7**	23.3	12.8
grid ($w = 7, h = 7, n_1 = 30$)	10.4	18.0	20.8	7.7	**4.6**
grid ($w = 9, h = 9, n_1 = 49$)	10.3	25.1	32.7	7.8	**5.4**
Erdős-Rényi ($n = 15, p = 0.05, n_1 = 10$)	19.6	25.4	**7.9**	15.0	13.9
Erdős-Rényi ($n = 30, p = 0.05, n_1 = 20$)	9.6	22.3	23.0	6.2	**4.5**
Erdős-Rényi ($n = 60, p = 0.05, n_1 = 40$)	10.8	32.5	61.1	8.1	**6.5**
Erdős-Rényi ($n = 60, p = 0.1, n_1 = 40$)	39.3	43.5	63.4	34.1	**27.2**
Erdős-Rényi ($n = 60, p = 0.15, n_1 = 40$)	54.9	56.2	62.1	52.2	**50.2**

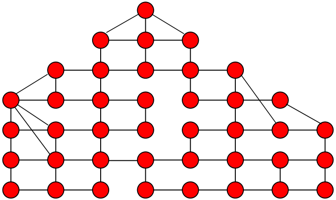
Comparison of Learned Network



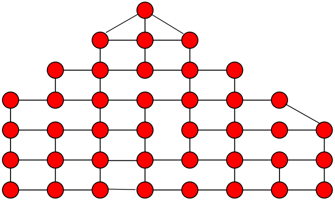
(a) Ground truth



(b) GLasso



(c) LV-GGM



(d) **DiLat-GGM**

Conclusion

- We propose the DiLat-GGM as a generalization of the LV-GGM
- The proposed model learns network topology given internal data and a summary of latent factors from external source
- Efficient algorithm based on CCP is proposed
- Future research direction: large-scale network learning, hierarchical models

Thank you !

References I

Venkat Chandrasekaran, Pablo A Parrilo, and Alan S Willsky. Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4):1935–1967, 2012.

Seung-Seok Choi, Sung-Hyuk Cha, and Charles C Tappert. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48, 2010.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250. ACM, 2010.

Cho-Jui Hsieh, Inderjit S Dhillon, Pradeep K Ravikumar, and Mátyás A Sustik. Sparse inverse covariance matrix estimation using quadratic approximation. *Advances in neural information processing systems*, pages 2330–2338, 2011.

References II

Paul Jaccard. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz, 1901.

Siddharth Joshi and Stephen Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, 2009.

Thomas Lipp and Stephen Boyd. Variations and extension of the convex–concave procedure. *Optimization and Engineering*, 17(2): 263–287, 2016.

Sijia Liu, Sundeep Prabhakar Chepuri, Makan Fardad, Engin Maşazade, Geert Leus, and Pramod K Varshney. Sensor selection for estimation with correlated measurement noise. *IEEE Transactions on Signal Processing*, 64(13):3509–3522, 2016.

Goran Marjanovic and Alfred O Hero. ℓ_0 sparse inverse covariance estimation. *IEEE Transactions on Signal Processing*, 63(12):3218–3231, 2015.

Eduardo Pavez and Antonio Ortega. Generalized laplacian precision matrix estimation for graph signal processing. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6350–6354. IEEE, 2016.

References III

Ming Yuan. Discussion: Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4):1968–1972, 2012.

Alan L Yuille, Anand Rangarajan, and AL Yuille. The concave-convex procedure (cccp). *Advances in neural information processing systems*, 2: 1033–1040, 2002.

DiLat-GGM as Difference-of-Convex program

$$\min_{\boldsymbol{C},\boldsymbol{B}} \underbrace{-\log\det\left(\boldsymbol{C}-\boldsymbol{B}\widehat{\Theta}_2\boldsymbol{B}^T\right)+\operatorname{tr}\left(\widehat{\Sigma}_1\boldsymbol{C}\right)}_{f(\boldsymbol{C},\boldsymbol{B})\text{ convex}} - \underbrace{\operatorname{tr}\left(\Sigma_1\boldsymbol{B}\widehat{\Theta}_2\boldsymbol{B}^T\right)}_{g(\boldsymbol{B})\text{ convex}} + \textit{regularizer}$$

$$\text{s.t.} \quad \boldsymbol{C}-\boldsymbol{B}\widehat{\Theta}_2\boldsymbol{B}^T \succeq \boldsymbol{0},$$

- $f(\boldsymbol{C},\boldsymbol{B}) = -\log\det\begin{bmatrix}\boldsymbol{C} & \boldsymbol{B}\\ \boldsymbol{B}^T & \widehat{\Theta}_2^{-1}\end{bmatrix} + \operatorname{tr}\left(\widehat{\Sigma}_1\boldsymbol{C}\right)$ convex

  $g(\boldsymbol{B}) = \operatorname{vec}\left(\boldsymbol{B}^T\right)^T\left(\widehat{\Sigma}_1\otimes\widehat{\Theta}_2\right)\operatorname{vec}\left(\boldsymbol{B}^T\right)$ convex
- can be solved via ***convex-concave procedure*** (**CCP**) [Yuille et al., 2002, Lipp and Boyd, 2016].

The convex sub-problem

At iteration t,

$$(\boldsymbol{C}_{t+1}, \boldsymbol{B}_{t+1}) = \min_{\boldsymbol{C},\boldsymbol{B}} \quad \ldots + \operatorname{tr}\left(\widehat{\Sigma}_1 \left(\boldsymbol{C} - 2\boldsymbol{B}\boldsymbol{D}_t^T\right)\right) \qquad (1)$$
$$\text{s.t.} \quad \ldots$$

where $\nabla_{\boldsymbol{B}} g(\boldsymbol{B}_t) = 2\widehat{\Sigma}_1 \boldsymbol{B}_t \widehat{\Theta}_2$, $\boldsymbol{D}_t := \boldsymbol{B}_t \widehat{\Theta}_2$.

- SDP problem $\Rightarrow$ convex
- CCP is a special form of *Majorization-minimization* (**MM**) algorithm.
- Guarantee to converge to local stationary point (regardless of choice of initial point)
- SDP time complexity $O(n^{6.5}) \Rightarrow$ an efficient solver based on ADMM, $O(n^3)$

Solving sub-problem using ADMM

- Define $\boldsymbol{R} := \begin{bmatrix} \boldsymbol{C} & \boldsymbol{B} \\ \boldsymbol{B}^T & \widehat{\Theta}_2^{-1} \end{bmatrix}$, $\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_1 & \boldsymbol{P}_{21}^T \\ \boldsymbol{P}_{21} & \boldsymbol{P}_2 \end{bmatrix} := \boldsymbol{R}$, $\boldsymbol{W} := \widehat{\Theta}_2 \boldsymbol{P}_{21}$

  We reformulate the convex sub-problem as

$$\min_{\boldsymbol{R},\boldsymbol{P},\boldsymbol{W}} \quad -\log\det \boldsymbol{R} + \mathrm{tr}\left(\boldsymbol{S}_t \boldsymbol{R}\right) + \mathbb{1}\left\{\boldsymbol{R} \succeq \boldsymbol{0}\right\} + \alpha_m \left\|\boldsymbol{P}_1\right\|_1 + \beta_m \left\|\boldsymbol{W}\right\|_{2,1} \qquad (2)$$

$$\text{s.t.} \quad \boldsymbol{P}_2 = \widehat{\Theta}_2^{-1}$$

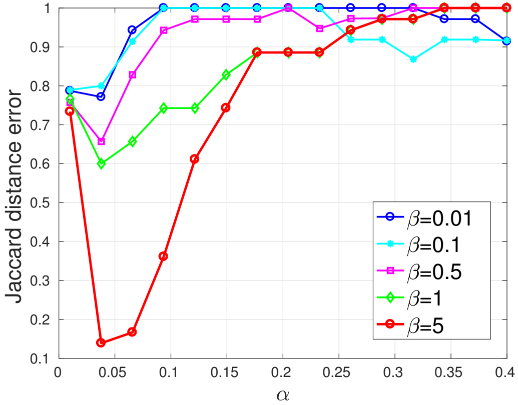
$$\boldsymbol{R} = \boldsymbol{P}$$

$$\boldsymbol{W} = \widehat{\Theta}_2 \boldsymbol{P}_{21}$$

  where $\mathbb{1}\{A\}$ is the indicator function, $\boldsymbol{S}_t := \begin{bmatrix} \widehat{\Sigma}_1 & -\widehat{\Sigma}_1 \boldsymbol{D}_t \\ -\boldsymbol{D}_t^T \widehat{\Sigma}_1 & \gamma_t \boldsymbol{I} \end{bmatrix}$

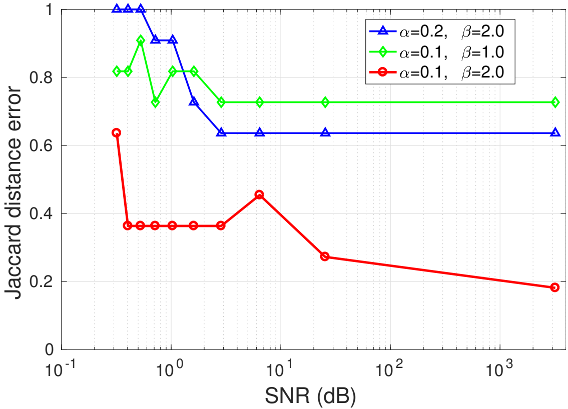
- ADMM solves three subproblems w.r.t. $\boldsymbol{R}$, $\boldsymbol{P}$, $\boldsymbol{W}$ iteratively

Sensitive to α, β



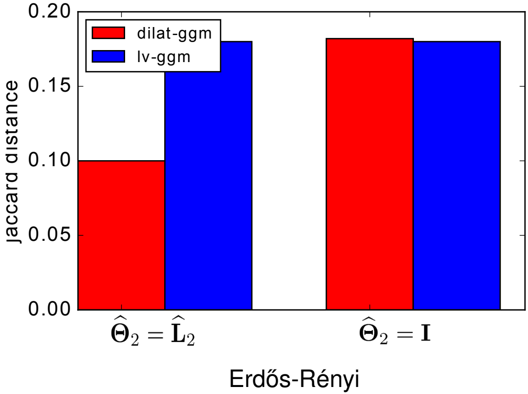
Erdős-Rényi $n = 30, p = 0.16$

Sensitivity to $\widehat{\Theta}_2$

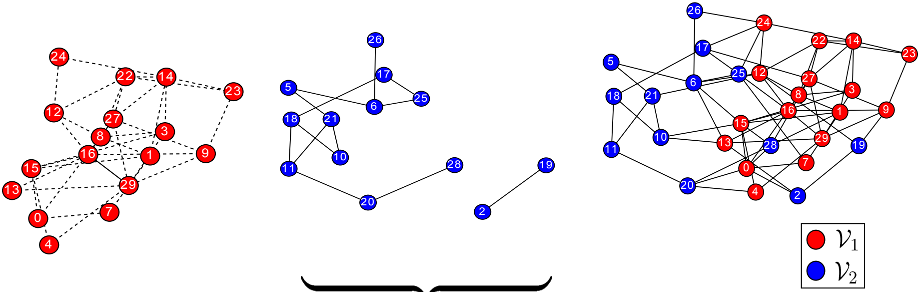


- $\widehat{\Theta}_2 = \widehat{\boldsymbol{L}}_2 + \sigma^2 \boldsymbol{G}$, where $\boldsymbol{G} = \boldsymbol{H}\boldsymbol{H}^T / n_2$, $H_{i,j} \sim N(0,1)$, $\widehat{\boldsymbol{L}}_2$ is the inverse covariance matrix of $\boldsymbol{x}_2$.
- The Signal-to-Noise Ratio (SNR) is defined as $\log \left(\frac{\|\widehat{\boldsymbol{L}}_2\|_F^2}{\sigma^2} \right)$ (dB)

Sensitivity to $\widehat{\Theta}_2$ (cond. correlated latent var.)

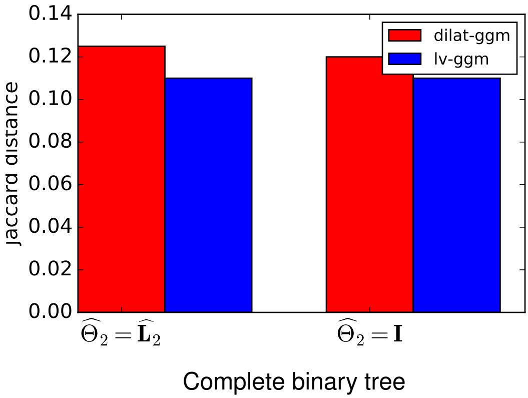


Sensitivity to $\widehat{\Theta}_2$ (cond. correlated latent var.)

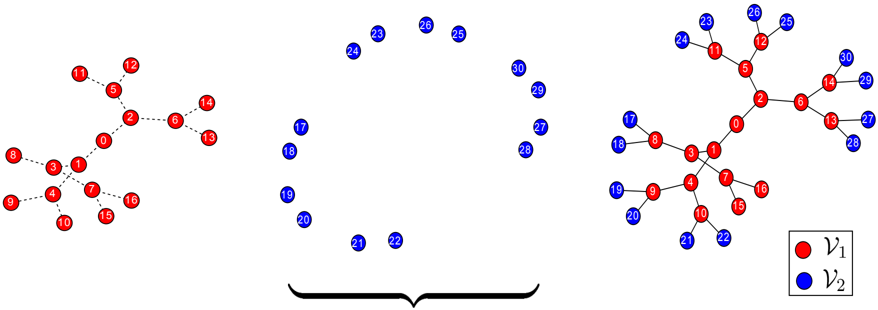


$$\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1) \; ? \leftarrow \mathbf{x}_{\mathcal{V}_1} \quad + \quad \underbrace{}\; \hat{\Theta}_{\mathcal{V}_2} \quad \Longleftarrow \quad \mathcal{G} = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E})$$

Sensitivity to $\widehat{\Theta}_2$ (cond. indep. latent var.)



Sensitivity to $\widehat{\Theta}_2$



$$\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1) \; ? \leftarrow \mathbf{x}_{\mathcal{V}_1} \quad + \quad \hat{\mathbf{\Theta}}_{\mathcal{V}_2} \quad \Longleftarrow \quad \mathcal{G} = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E})$$