Signal processing on graphs: Causal modeling of unstructured data

Presenter: Hong-Ming Chiu

Email: hongmingchiu0217@gmail.com

Advisor: Professor Carrson C. Fung

National Chiao Tung University

May. 8, 2019

Citiation :

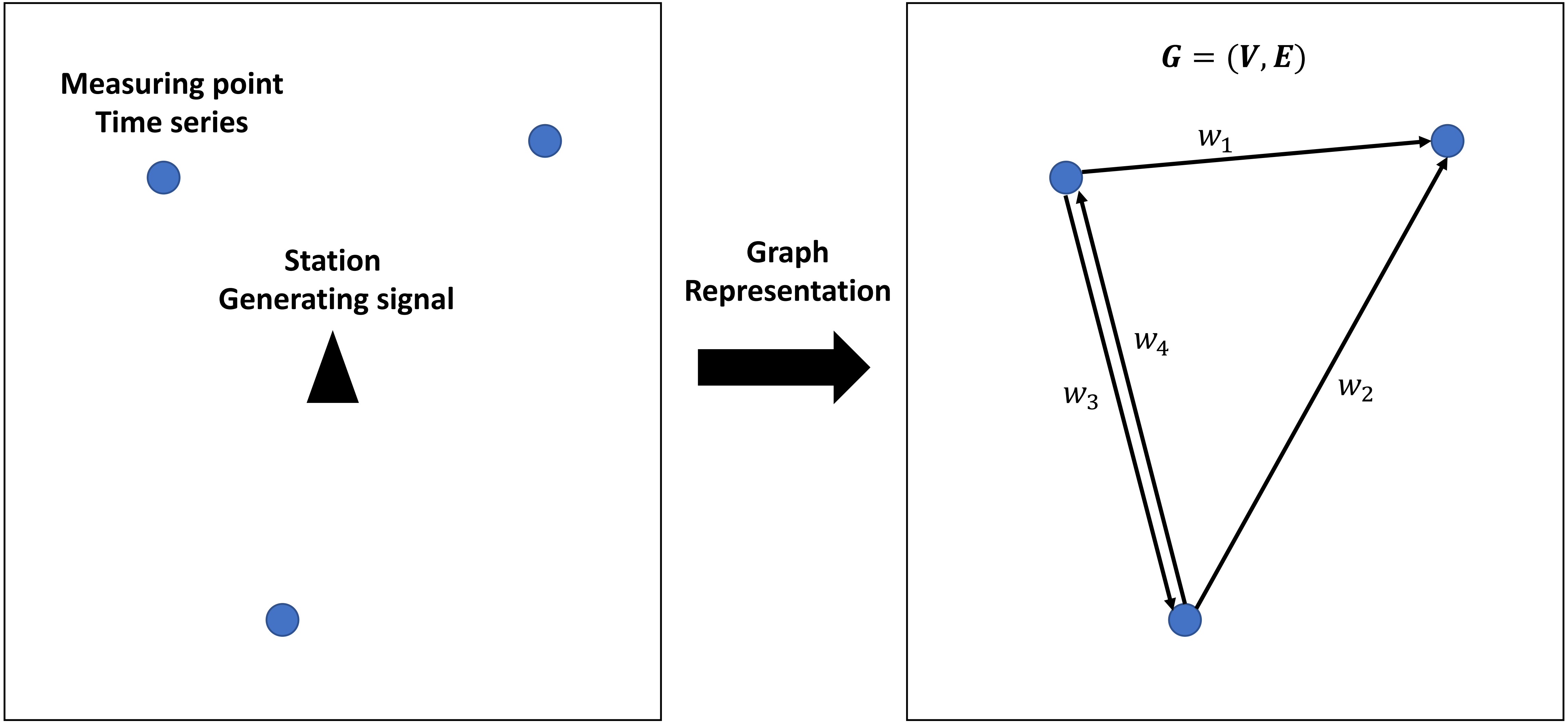
1 : J. Mei and J.M.F. Moura, “Signal processing on graphs: Causal modeling of unstructured data” IEEE Trans. on Signal Processing, vol. 65(8), pp. 2077−2092, 2017.

Content

I. Introduction

Introduction

- Many applications collect a large number of time series, for example, measuring receiving signal power at different locations. These data are often **unstructured**.
- Therefore, it is often useful to derive a low dimensional representation among these time series.
- A graph could be used to describe the **interrelations** among the **time series** and their **intrarelations** across **time**.


Measuring point
Time series
Station
Generating signal
Graph
Representation
$G = (V, E)$
w_1
w_4
w_3
w_2

Introduction

- The paper proposes a model call “**Causal Graph Process (CGP)**” to learning the graph.
- “**Sparse Vector Autoregressive model (SVAR)**” is another model for learning the graph, although it is not proposed by this paper, we will compare it to the CGP model.

Citiation :

2 : A. Davis, Richard & Zang, Pengfei & Zheng, Tian. (2012). “Sparse Vector Autoregressive Modeling.” Journal of Computational and Graphical Statistics. 30. 10.1080/10618600.2015.1092978.

Introduction

- The graph representation could be either directed or undirected, depending on your application. However, SVAR can only handle unweighted graph while CGP can handle both weighted and unweighted graph.
- The graph representation among time series allows us to capture the **correlation** between time series as well as **predict** the future date on each measuring points.

II. Graph learning

Parameter

- N denotes the number of nodes.
- K denotes the number of time samples.
- $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix of graph $G = (V, E)$.
- $\boldsymbol{x}_n[k] \in \mathbb{C}^K$ a time series on node n in graph $G = (V, E)$.
- $\boldsymbol{x}[k_0] = \left[\boldsymbol{x}_0[k_0] \ \dots \boldsymbol{x}_{N-1}[k_0]\right]^T \in \mathbb{C}^N$ a graph signal at time k_0.
- $\boldsymbol{w}[k]$: $white\ Gaussian\ noise$.

1. Sparse Vector Autoregressive Model (SVAR)

SVAR Model

- $\boldsymbol{x}[k] = \sum_{i=1}^{M} \boldsymbol{A}^{(i)} \boldsymbol{x}[k-i] + \boldsymbol{w}[k]$
- $\boldsymbol{A}^{(i)} \in \mathbb{R}^{N \times N}$ is the evolution matrix which contains the autoregressive coefficients describing the influence of nodes in graph at a delay of i time samples.
- M is the memory depth or the order of the system.
- Each $\boldsymbol{A}^{(i)}$ has the **same sparse structure** governed by $\boldsymbol{A}'$

$$where\ \boldsymbol{A}' \in \{0,1\}^{N \times N}\ such\ that\ \boldsymbol{A}'_{ij} = 0 \rightarrow \boldsymbol{A}^{(i)}_{ij} = 0$$

Estimating $\boldsymbol{A}$

- Assume the graph signals are generated from SVAR model.
- Using all the time series we observe, we can estimate by solving following optimization problem :

$$\left\{\widehat{\boldsymbol{A}}^{(i)}\right\} = \underset{\{\boldsymbol{A}^{(i)}\}}{\operatorname{argmin}} \frac{1}{2} \sum_{k=M}^{K-1} \left\| \boldsymbol{x}[k] - \sum_{i=1}^{M} \boldsymbol{A}^{(i)} \boldsymbol{x}[k-i] \right\|_2^2 + \lambda \sum_{i,j} \left\| \boldsymbol{a}_{ij} \right\|_2$$

$$where\ \boldsymbol{a}_{ij} = \left(\boldsymbol{A}_{ij}^{(1)} \ldots \boldsymbol{A}_{ij}^{(M)} \right)^T$$

Estimating $\boldsymbol{A}$

- $\widehat{\boldsymbol{A}}_{ij}^{(1)} \dots \widehat{\boldsymbol{A}}_{ij}^{(M)}$ can be solve by using convex optimization approach and group Lasso.
- The group Lasso term $\lambda \sum_{i,j} \lVert \boldsymbol{a}_{ij} \rVert_2$ cannot be change to $\lambda \sum_{i,j} \lVert \boldsymbol{a}_{ij} \rVert_2^2$ since we need to equally penalize the variables in each $\boldsymbol{a}_{ij} = \left(\boldsymbol{A}_{ij}^{(1)} \dots \boldsymbol{A}_{ij}^{(M)} \right)^T$ so that all the evolution matrices will have the same sparse structure.

Citiation :
3 : Yuan, Ming and Yi Juain Lin. “Model selection and estimation in regression with grouped variables.” (2006).

Estimating $\boldsymbol{A}$

- Since $\widehat{\boldsymbol{A}}_{ij}^{(1)} \dots \widehat{\boldsymbol{A}}_{ij}^{(M)}$ should have the same sparse structure. We can set a proper threshold ϵ such that

$$\widehat{\boldsymbol{A}}_{ij} = \begin{cases} 1 \;\; if \; \forall n \in \{1 \dots M\} \, \widehat{\boldsymbol{A}}_{ij}^{(n)} > \epsilon \\ 0 \;\; otherwise \end{cases}$$

- The adjacency matrix $\widehat{\boldsymbol{A}}$ is unweighted.

2. Causal Graph Process (CGP)

CGP Model

- $\boldsymbol{x}[k] = \boldsymbol{w}[k] + \sum_{i=1}^{M} P_i(\boldsymbol{A}, \boldsymbol{c})\boldsymbol{x}[k-i]$
- $P_i(\boldsymbol{A}, \boldsymbol{c}) = c_{i0}\boldsymbol{I} + c_{i1}\boldsymbol{A}^1 + \cdots + c_{ii}\boldsymbol{A}^i$ (matrix polynomial)
- $\boldsymbol{c} = \left[c_{10}\ c_{11}\ \ldots c_{ij} \ldots c_{MM}\right]^T$ coefficients of matrix polynomial.
- M is the maximum order of matrix polynomial. (memory)
- This model allows a signal on a node at current time to be affected through network effects by signals on other nodes at past times.

CGP Model

- $$\begin{aligned}\boldsymbol{x}[k] &= \boldsymbol{w}[k] + \sum_{i=1}^{M} P_i(\boldsymbol{A}, \boldsymbol{c})\boldsymbol{x}[k-i] \\ &= \boldsymbol{w}[k] + (c_{10}\boldsymbol{I} + c_{11}\boldsymbol{A})\boldsymbol{x}[k-1] \\ &\quad +(c_{20}\boldsymbol{I} + c_{21}\boldsymbol{A} + c_{22}\boldsymbol{A}^2)\boldsymbol{x}[k-2] + \cdots \\ &\quad +(c_{M0}\boldsymbol{I} + \cdots + c_{MM}\boldsymbol{A}^M)\boldsymbol{x}[k-M]\end{aligned}$$
- Set $c_{10} = 0$ and $c_{11} = 1$ to ensure that $\boldsymbol{A}$ and $\boldsymbol{c}$ are uniquely specified, that is, $P_1(\boldsymbol{A}, \boldsymbol{c}) = \boldsymbol{A}$.

Estimating $\boldsymbol{A}$

- Assume the graph signals are generated from CGP model.
- Using all the time series we observe, we can estimate by solving following optimization problem :

$$\left(\widehat{\boldsymbol{A}}\,,\hat{\boldsymbol{c}}\right) = \underset{\{\boldsymbol{A}\,,\boldsymbol{c}\}}{\operatorname{argmin}} \frac{1}{2}\sum_{k=M}^{K-1}\left\|\boldsymbol{x}[k] - \sum_{i=1}^{M} P_i(\boldsymbol{A},\boldsymbol{c})\boldsymbol{x}[k-i]\right\|_2^2$$

$$+\lambda_1\|vec(\boldsymbol{A})\|_1 + \lambda_2\|\boldsymbol{c}\|_1$$

Estimating $\boldsymbol{A}$

- The matrix polynomial $P_i(\boldsymbol{A}, \boldsymbol{c})$ makes this problem nonconvex, therefore directly optimize $\left(\widehat{\boldsymbol{A}}, \hat{\boldsymbol{c}}\right)$ using convex optimization approach would only optimize $\left(\widehat{\boldsymbol{A}}, \hat{\boldsymbol{c}}\right)$ locally.
- Breaking the problem into three more traceable steps :
  1. Solve $\mathrm{R_i} = P_i(\boldsymbol{A}, \boldsymbol{c})$
  2. Recover $\widehat{\boldsymbol{A}}$
  3. Recover $\hat{\boldsymbol{c}}$

Estimating $\boldsymbol{A}$: Solve $\mathrm{R_i} = P_i(\boldsymbol{A}, \boldsymbol{c})$

- $\widehat{\boldsymbol{R}}_i \; = \underset{\{\boldsymbol{R}_i\}}{\operatorname{argmin}} \frac{1}{2} \sum_{k=M}^{K-1} \left\| \boldsymbol{x}[k] - \sum_{j=1}^{M} \boldsymbol{R}_j \boldsymbol{x}[k-j] \right\|_2^2 + \lambda_1 \| vec(\boldsymbol{R}_1) \|_1$

$$+\lambda_3 \sum_{j \neq i} \left\| [\boldsymbol{R}_i, \boldsymbol{R}_j] \right\|_F^2$$

$$where \left[\boldsymbol{R}_i, \boldsymbol{R}_j\right] = \; \boldsymbol{R}_i \boldsymbol{R}_j - \boldsymbol{R}_j \boldsymbol{R}_i$$

- $\widehat{\boldsymbol{R}}_1 \; = \widehat{\boldsymbol{A}}$, due to the assumption that $c_{10} = 0$ and $c_{11} = 1$, $P_1(\boldsymbol{A}, \boldsymbol{c}) = \widehat{\boldsymbol{R}}_1 = \widehat{\boldsymbol{A}}$
- Can be solved by using block coordinate descent.

Estimating $\boldsymbol{A}$: Recover $\widehat{\boldsymbol{A}}$

- $\widehat{\boldsymbol{A}} = \underset{\{\boldsymbol{A}\}}{\operatorname{argmin}} \frac{1}{2} \left\| \widehat{\boldsymbol{R}}_1 - \boldsymbol{A} \right\|_2^2 + \lambda_1 \| vec(\boldsymbol{A}) \|_1 + \lambda_3 \sum_{i=2}^{M} \left\| \left[\boldsymbol{A}, \widehat{\boldsymbol{R}}_i \right] \right\|_F^2$

$$where \left[\boldsymbol{A}, \widehat{\boldsymbol{R}}_i \right] = \boldsymbol{A} \widehat{\boldsymbol{R}}_i - \widehat{\boldsymbol{R}}_i \boldsymbol{A}$$

- Or you could just skip this step and let $\widehat{\boldsymbol{A}} = \widehat{\boldsymbol{R}}_1$.
- The above optimization problem seems redundant but it is useful in simplified estimation algorithm.

Estimating $\boldsymbol{A}$: Recover $\hat{\boldsymbol{c}}$

- $\widehat{\boldsymbol{c}_i} = \underset{\boldsymbol{c}_i}{\operatorname{argmin}} \frac{1}{2} \left\| vec(\widehat{\boldsymbol{R}}_i) - \boldsymbol{Q}_i \boldsymbol{c}_i \right\|_2^2 + \lambda_2 \|\boldsymbol{c}_i\|_1$
  $Where\ \boldsymbol{Q}_i = \left[vec(\boldsymbol{I})\ \ vec\left(\widehat{\boldsymbol{A}}\right) \dots vec\left(\widehat{\boldsymbol{A}}^{\,i}\right)\right], \boldsymbol{c}_i = [c_{i0}\ \ c_{i1}\ \ \dots c_{ii}]^T$
- Estimating $\widehat{\boldsymbol{c}_i}$ from $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{R}}_i$.
- $\hat{\boldsymbol{c}} = [\hat{\boldsymbol{c}}_1^T \dots \hat{\boldsymbol{c}}_M^T]^T$ but it may not be accurate since both $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{R}}_i$ are estimated.
- To achieve higher accuracy, estimating $\hat{\boldsymbol{c}}$ from $\widehat{\boldsymbol{A}}$ and the time series we have instead.

Estimating $\boldsymbol{A}$: Recover $\hat{\boldsymbol{c}}$

- $\hat{\boldsymbol{c}} = \underset{\boldsymbol{c}}{\operatorname{argmin}} \frac{1}{2} \left\| \boldsymbol{Y}(\widehat{\boldsymbol{A}}) - B(\widehat{\boldsymbol{A}})\boldsymbol{c} \right\|_F^2 + \lambda_2 \|\boldsymbol{c}\|_1$

$$Where\ \boldsymbol{Y}(\widehat{\boldsymbol{A}}) = vec\left(\boldsymbol{X}_M - \widehat{\boldsymbol{A}}\,\boldsymbol{X}_{M-1}\right)$$

$$\boldsymbol{X}_m = \left[\boldsymbol{x}[m]\ \boldsymbol{x}[m+1] \ldots \boldsymbol{x}[m+K-M-1]\right]$$

$$B(\widehat{\boldsymbol{A}}) = \left[vec(\boldsymbol{X}_{M-2}) \ldots vec\left(\widehat{\boldsymbol{A}}^{\,j}\boldsymbol{X}_{M-i}\right) \ldots vec\left(\widehat{\boldsymbol{A}}^{\,M}\boldsymbol{X}_0\right)\right]$$

$$\boldsymbol{c} = \left[c_{2j} \ldots c_{ij} \ldots c_{MM}\right]^T, i = 2\sim M\{j = 0\sim i\}$$

- estimating $\hat{\boldsymbol{c}}$ from $\widehat{\boldsymbol{A}}$ and time series.
- Minimize K-M steps error.

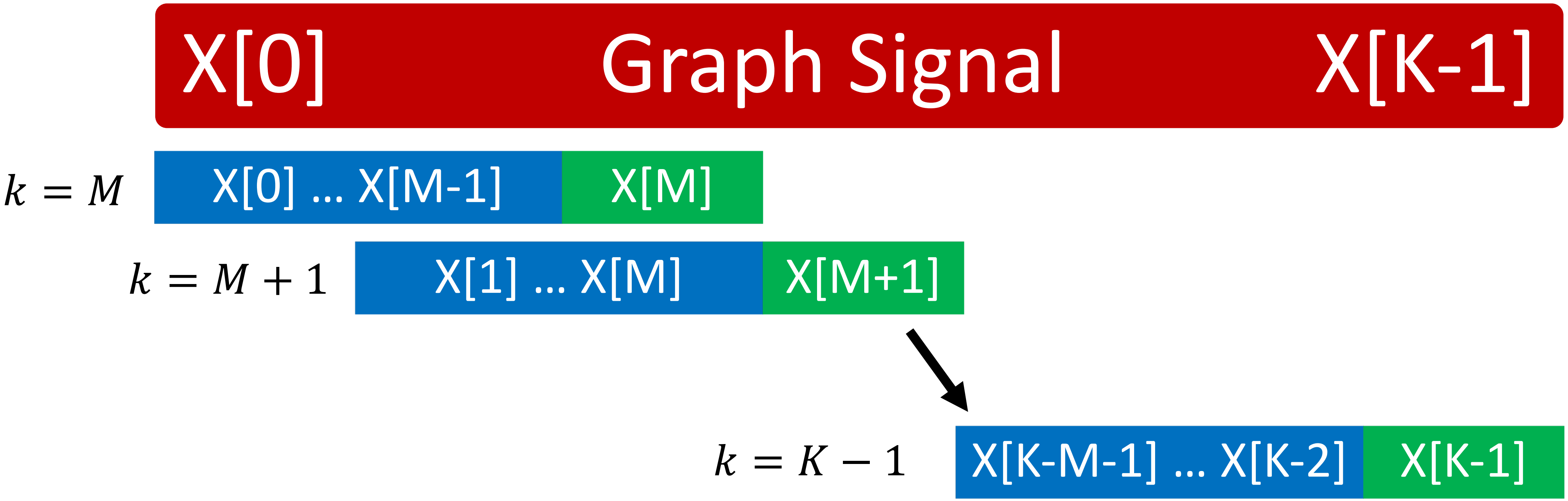
Estimating $\boldsymbol{A}$: Recover $\hat{\boldsymbol{c}}$

- Consider the first column of $\boldsymbol{X}_m$, $\left[\boldsymbol{Y}\left(\widehat{\boldsymbol{A}}\right) - B\left(\widehat{\boldsymbol{A}}\right)c\right]$ will become :
$vec\left(\boldsymbol{x}[M] - \widehat{\boldsymbol{A}}\,\boldsymbol{x}[M-1]\right) - vec(c_{20}\boldsymbol{x}[M-2]) - vec\left(c_{21}\widehat{\boldsymbol{A}}\,\boldsymbol{x}[M-2]\right)$
$- vec\left(c_{22}\widehat{\boldsymbol{A}}^{\,2}\boldsymbol{x}[M-2]\right) - \cdots - vec(c_{M0}\boldsymbol{x}[0]) - \cdots - vec\left(c_{MM}\widehat{\boldsymbol{A}}^{\,M}\boldsymbol{x}[0]\right)$
- Which is exactly the same as $\boldsymbol{x}[k] - \sum_{i=1}^{M} P_i(\boldsymbol{A},\boldsymbol{c})\boldsymbol{x}[k-i]\Big|_{k=M}$
- The rest of the columns in $\boldsymbol{X}_m$ is just the case where

$$k = M+1 \sim K-1$$

- This process is illustrated in next page.

Estimating $\boldsymbol{A}$: Recover $\hat{\boldsymbol{c}}$



Algorithm

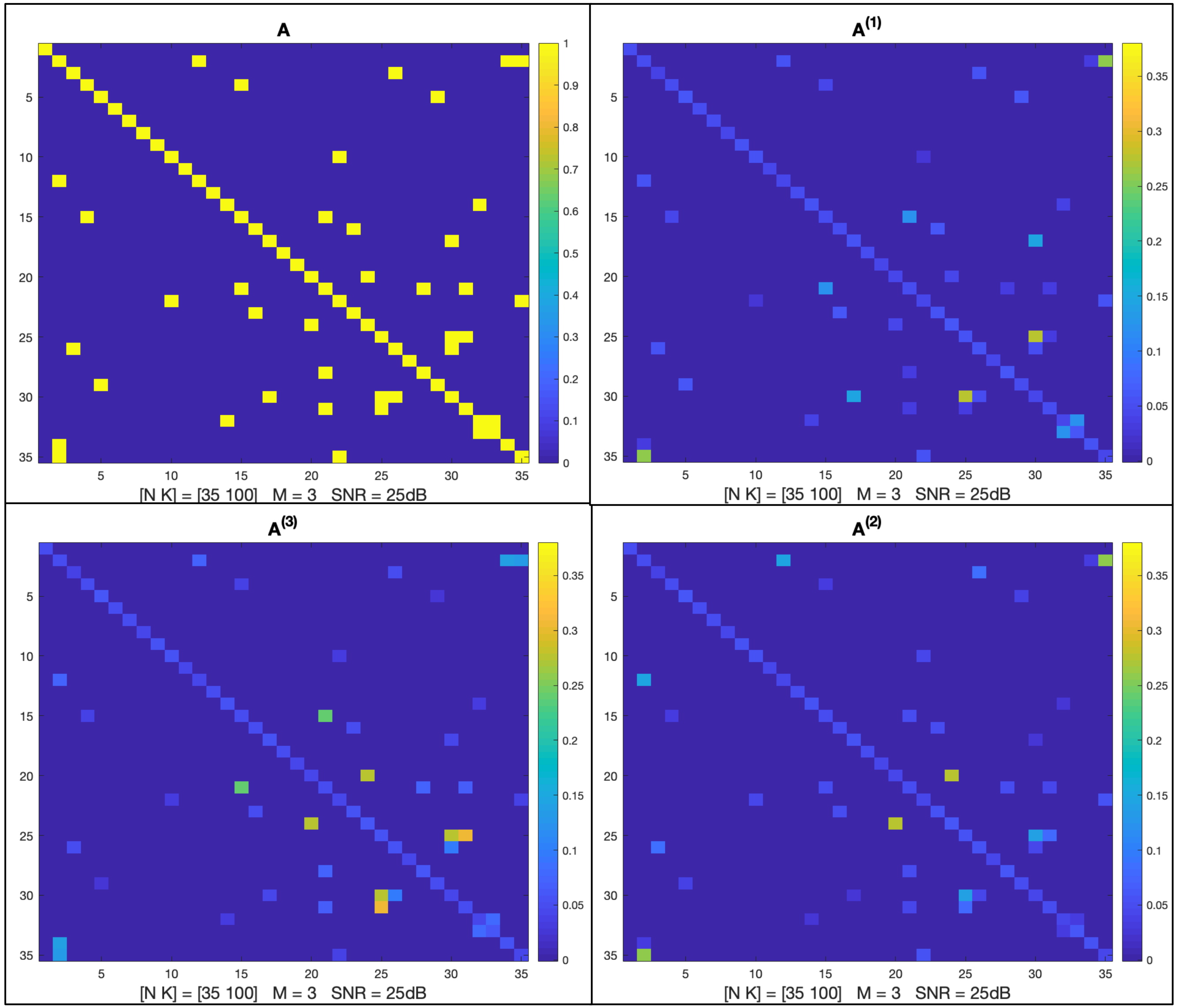
- **Base estimation algorithm** : what we have discussed earlier.
- **Simplified estimation algorithm** : convexify the first step in the base estimation algorithm (remove the commutive enforced term) and then use step 2 to recover the commutive property and then step 3.
- **Extended estimation algorithm** : solving nonconvex problem in one shot, repeat until finding batter local minimum.

iii. Simulation

Simulation

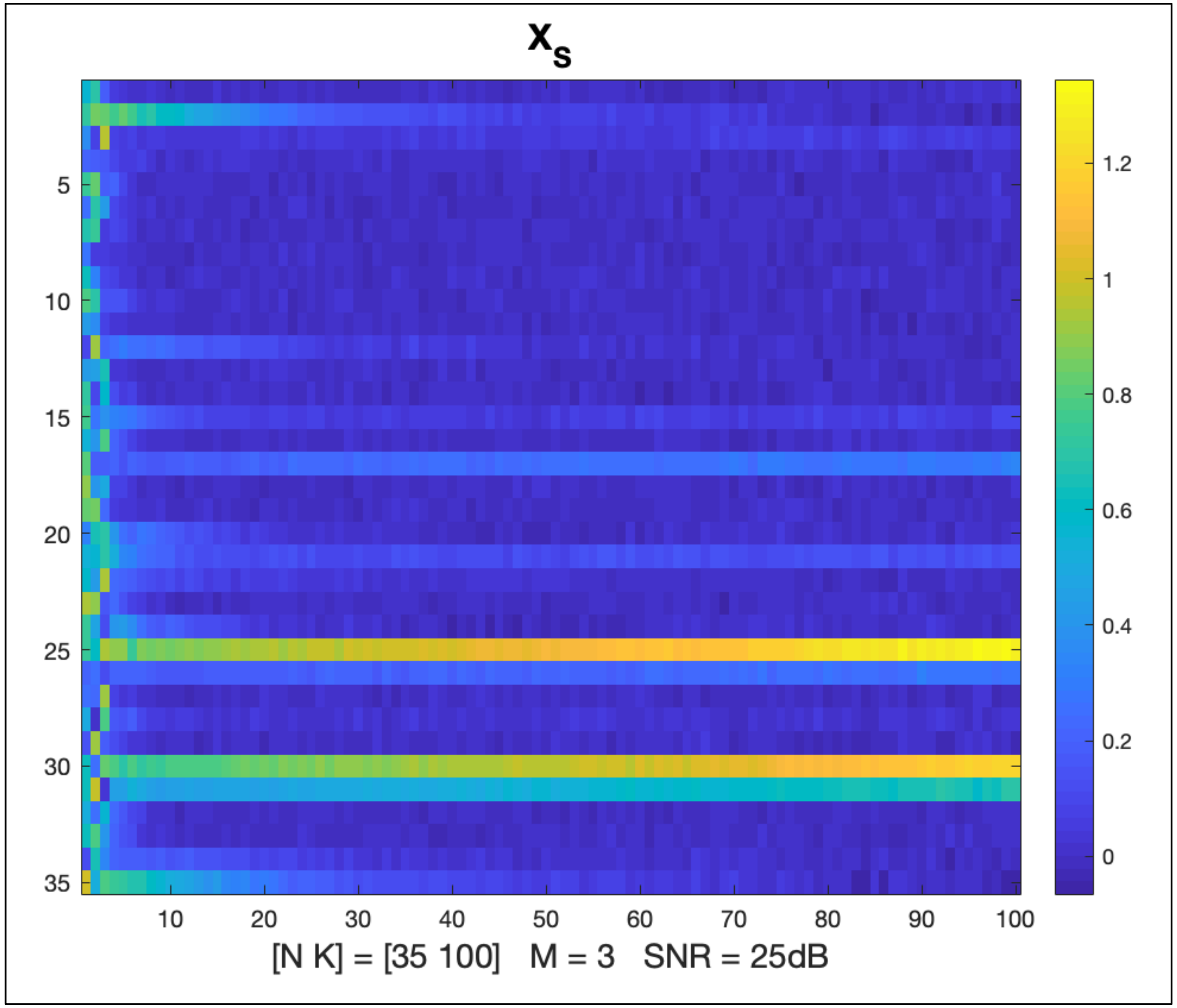
- Generating ground true $\boldsymbol{A}$, $\boldsymbol{c}$, $\boldsymbol{A}^{(1)} \sim \boldsymbol{A}^{(M)}$.
- Randomly generating M graph signal ($\boldsymbol{x}[0] \sim \boldsymbol{x}[M-1]$) then using CGP and SVAR to get the graph signal from $k = M \sim 99$.
- Estimate $\widehat{\boldsymbol{A}}$, $\widehat{\boldsymbol{A}}_{ij}^{(1)}$ … $\widehat{\boldsymbol{A}}_{ij}^{(M')}$ using M' order SVAR.
- Estimate $\widehat{\boldsymbol{A}}$, $\hat{\boldsymbol{c}}$ using M' order CGP model.
- Doing prediction on future graph signals form $k = 100 \sim 299$ by using CGP and SVAR respectively.

$A, A^{(1)} \sim A^{(3)}$ for SVAR



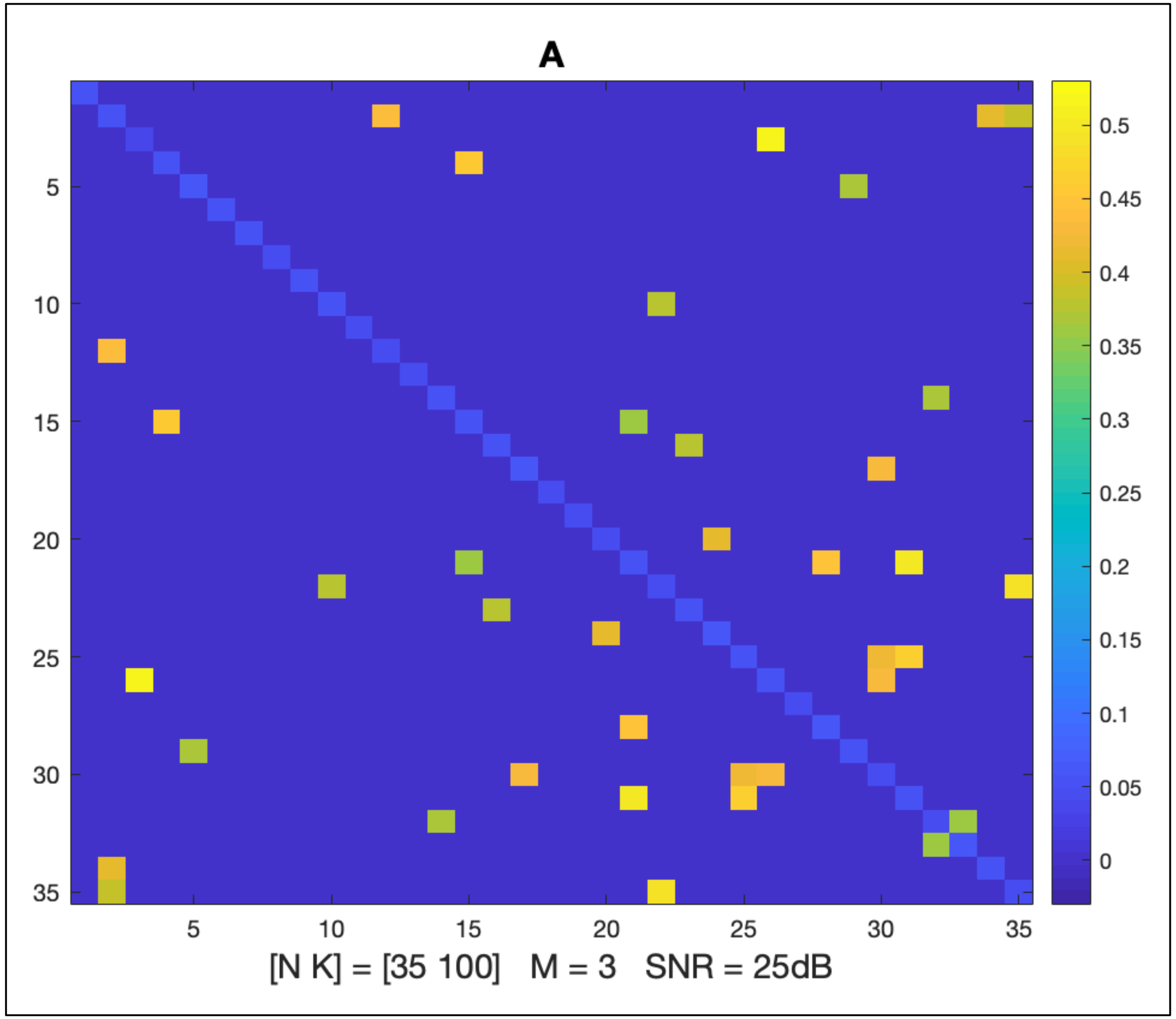
- $A^{(1)} \sim A^{(3)}$ have the same sparse structure.
- $A \in \{1,0\}$

Generating $\boldsymbol{X}_S$ from SVAR



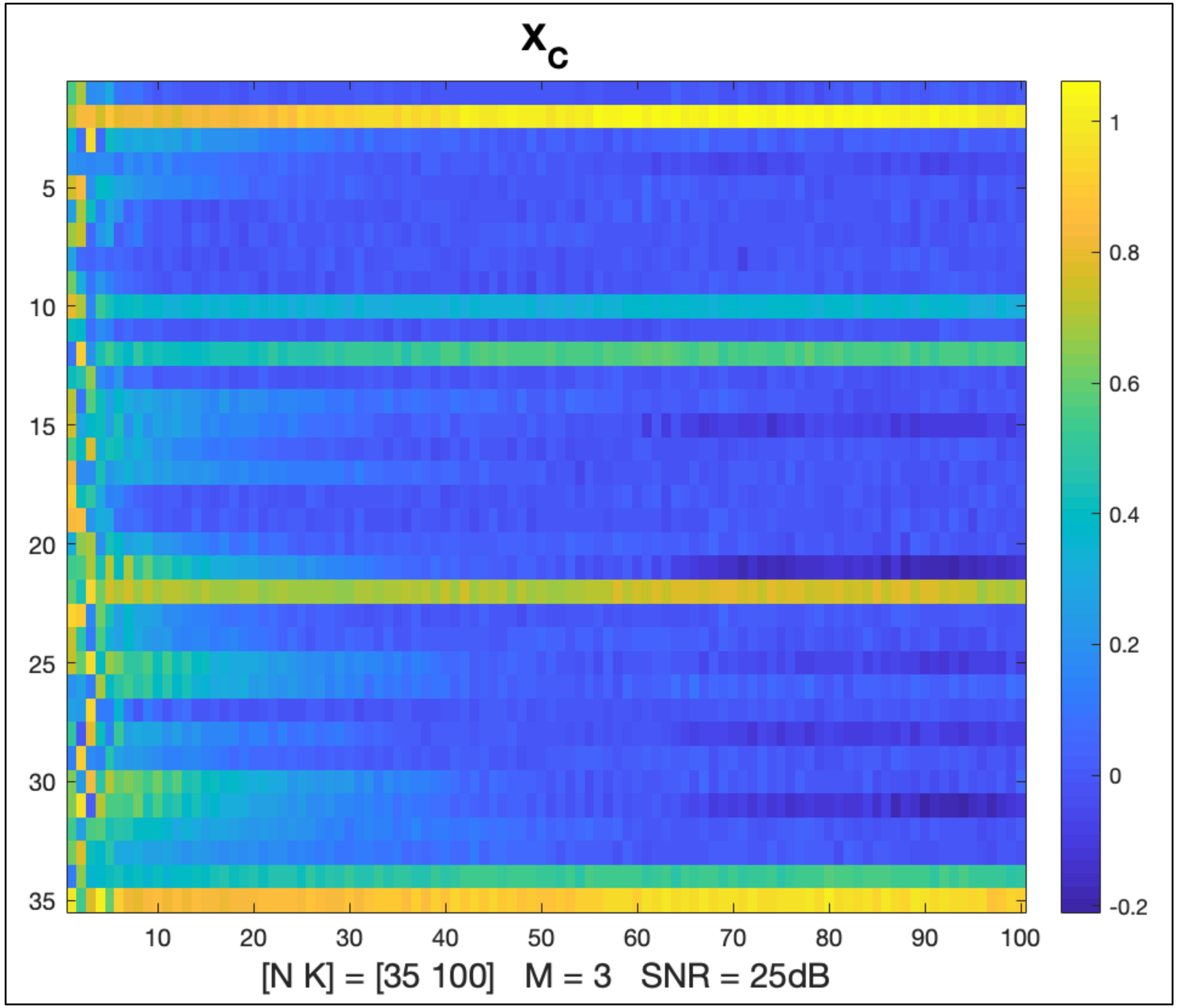
- Generating $\boldsymbol{X}_S$ using $\boldsymbol{A}^{(1)} \sim \boldsymbol{A}^{(3)}$.
- $M = 3$
- $\boldsymbol{X} = \begin{bmatrix} | & & | \\ \boldsymbol{x}[1] & \dots & \boldsymbol{x}[K] \\ | & & | \end{bmatrix} \in R^{N \times K}$
- $\boldsymbol{X}_S$ denote the ground true data generated from SVAR.

$\boldsymbol{A}$ and $\boldsymbol{c}$ for CGP



- For $\boldsymbol{c}$
- $[c_{10}\ c_{11}] = [0\ \ 1]$
- $[c_{20}\ c_{21}\ c_{22}] =$ $[-0.0175\ \ 0.1356\ \ 0.2878]$
- $[c_{30}\ c_{31}\ c_{32}\ c_{33}] =$ $[0.3474\ \ 0.1868\ -0.5378\ \ -0.4398]$

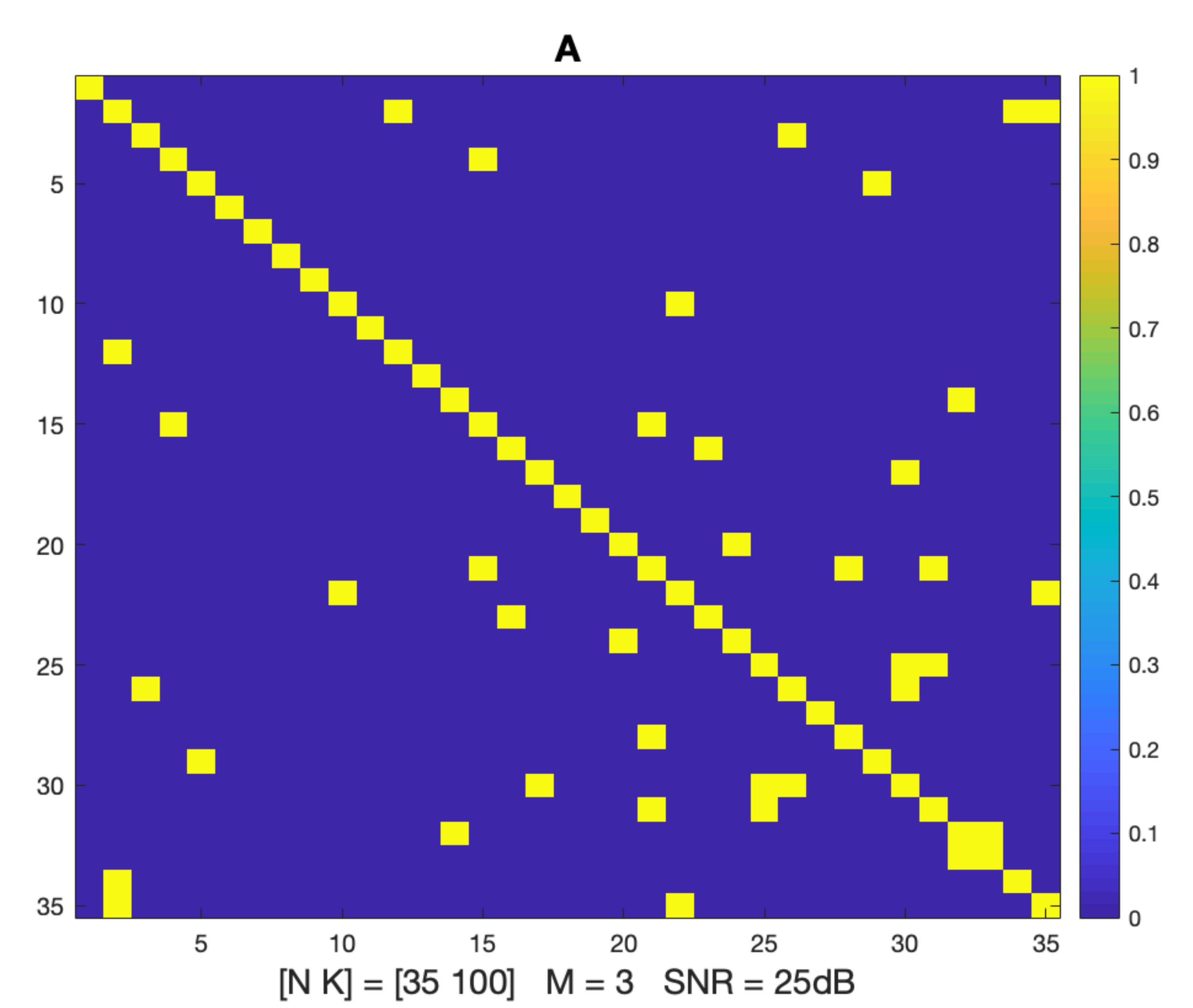
Generating $\boldsymbol{X}_C$ from CGP

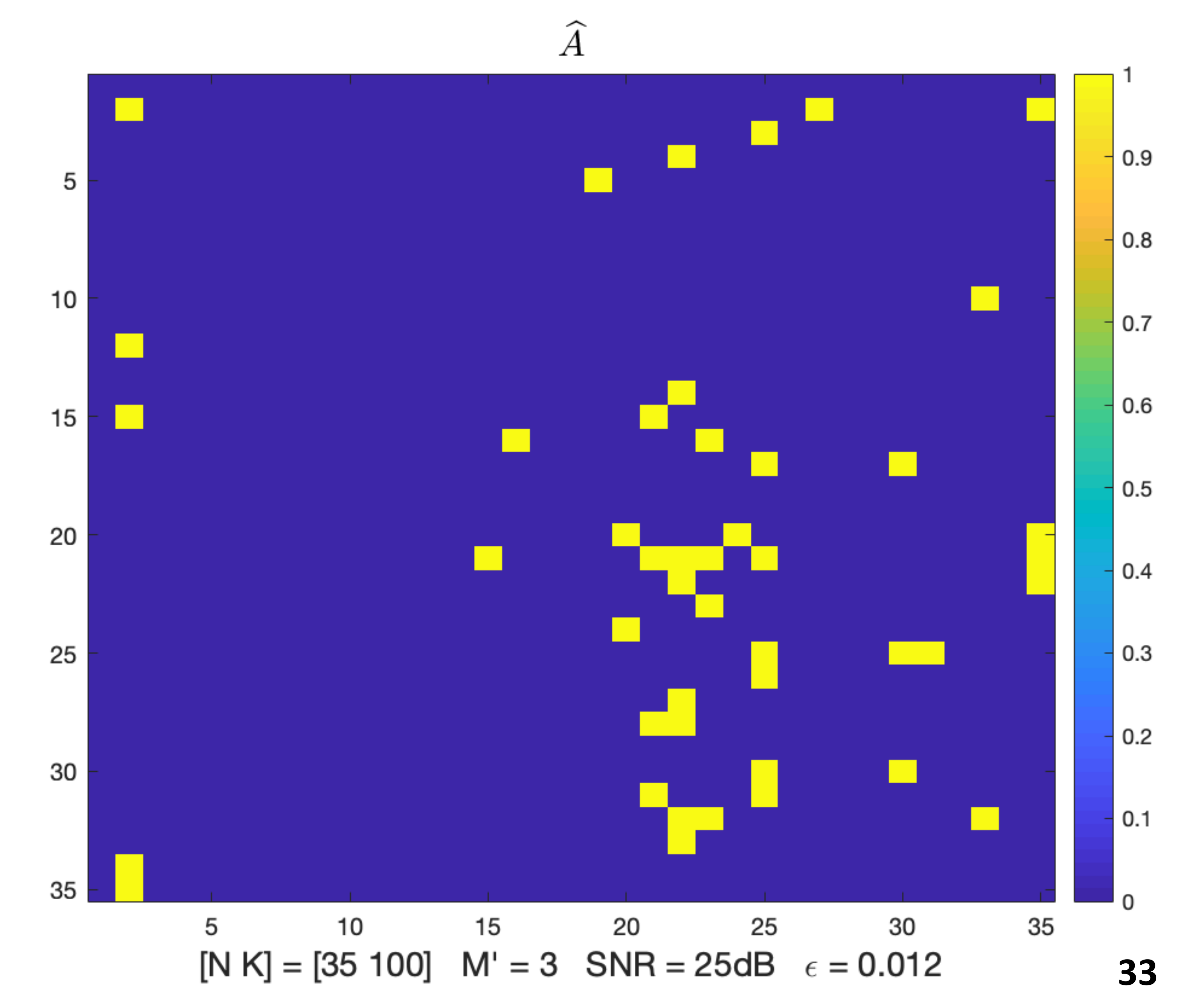


- Generating $\boldsymbol{X}_C$ using $\widehat{\boldsymbol{A}}$, $\hat{\boldsymbol{c}}$.
- $M = 3$
- $\boldsymbol{X} = \begin{bmatrix} | & & | \\ \boldsymbol{x}[1] & \dots & \boldsymbol{x}[K] \\ | & & | \end{bmatrix} \in R^{N \times K}$
- $\boldsymbol{X}_C$ denote the ground true data generated from CGP.

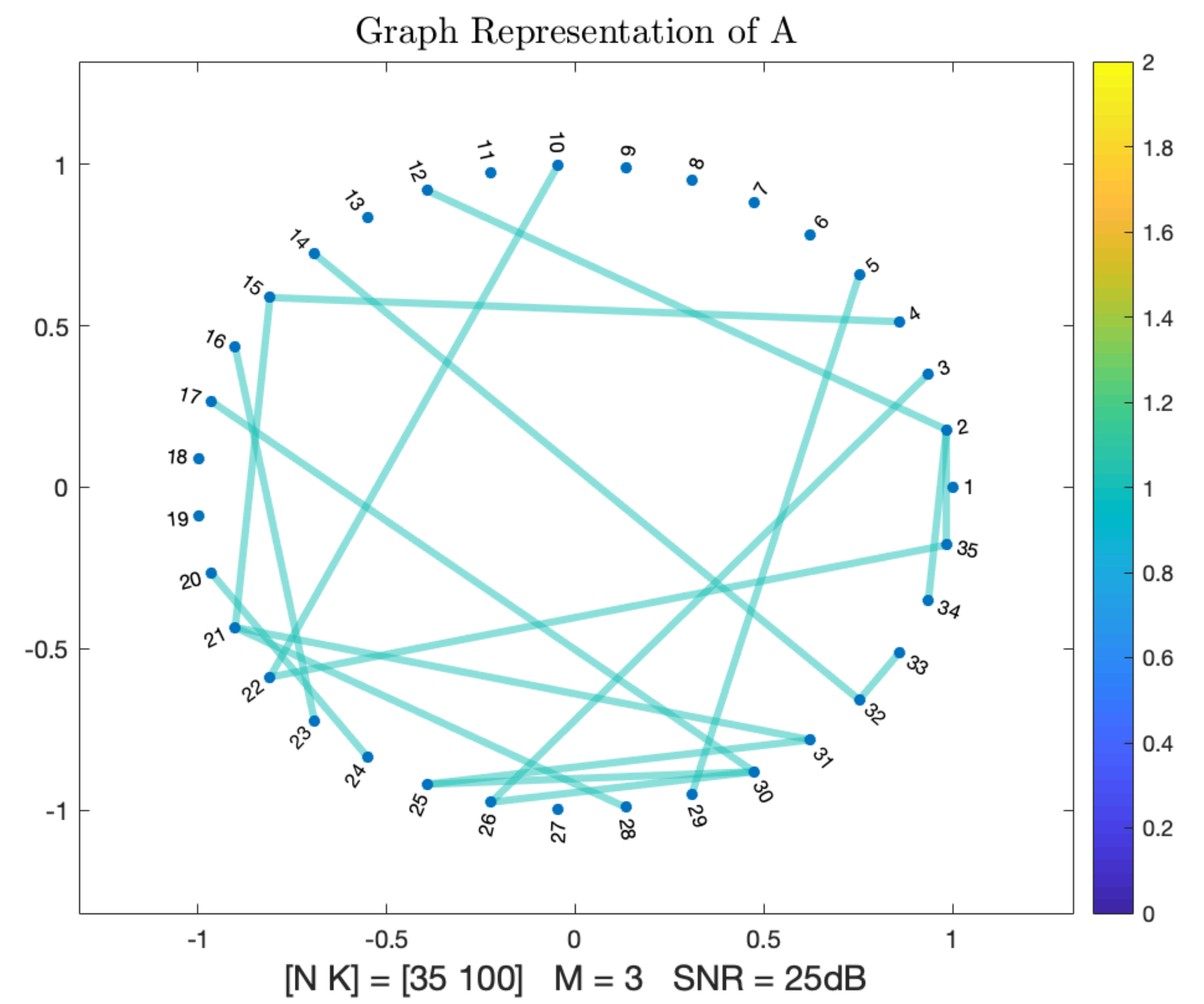
1. The same order as the ground true $M' = M$

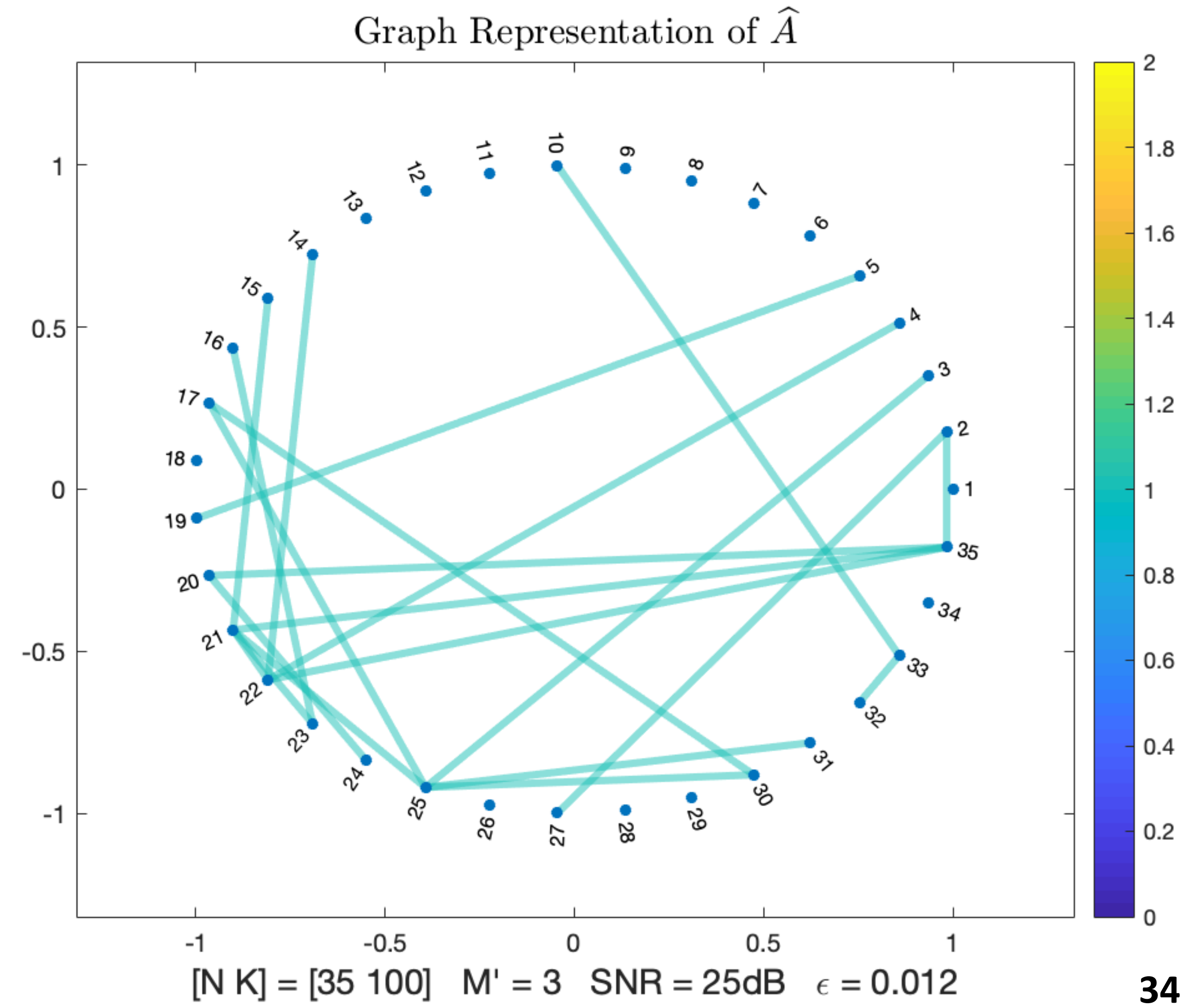
Recover $\widehat{\boldsymbol{A}}$ from $\boldsymbol{X}_S$ using SVAR



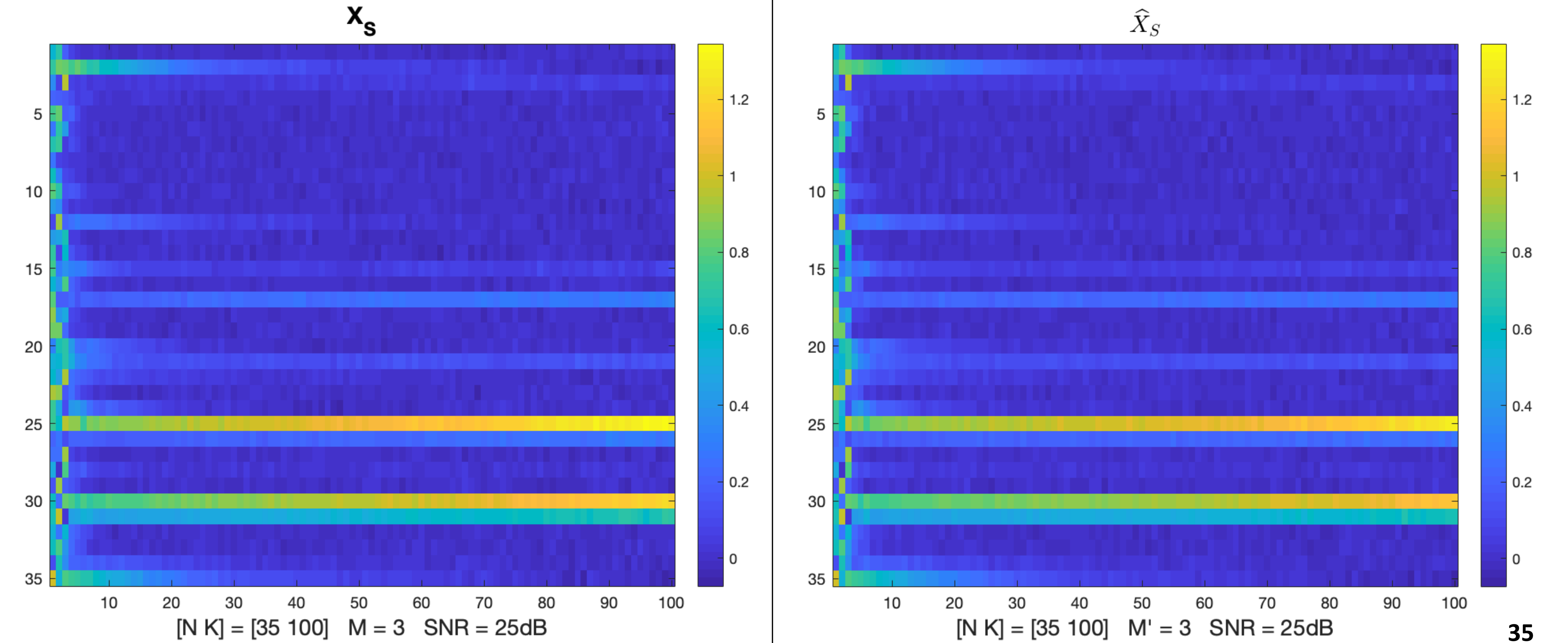


Recover $\hat{A}$ from X_S using SVAR

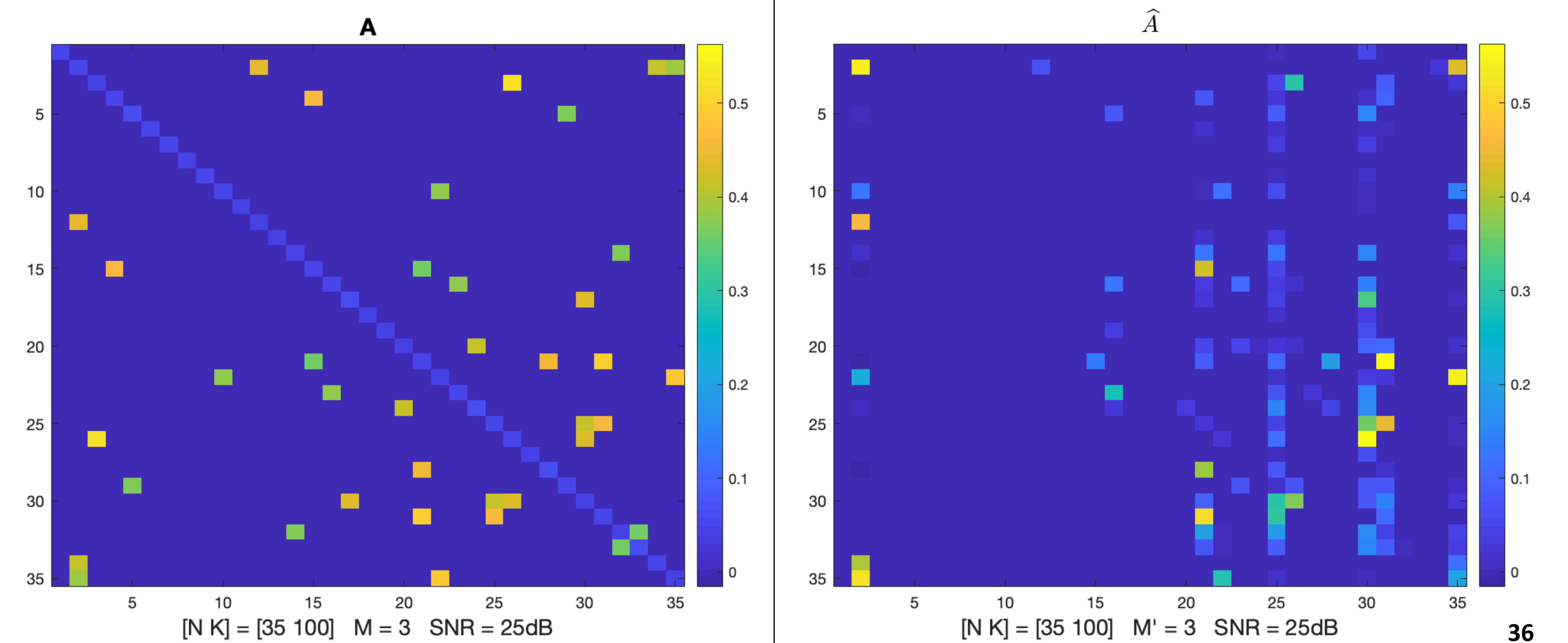




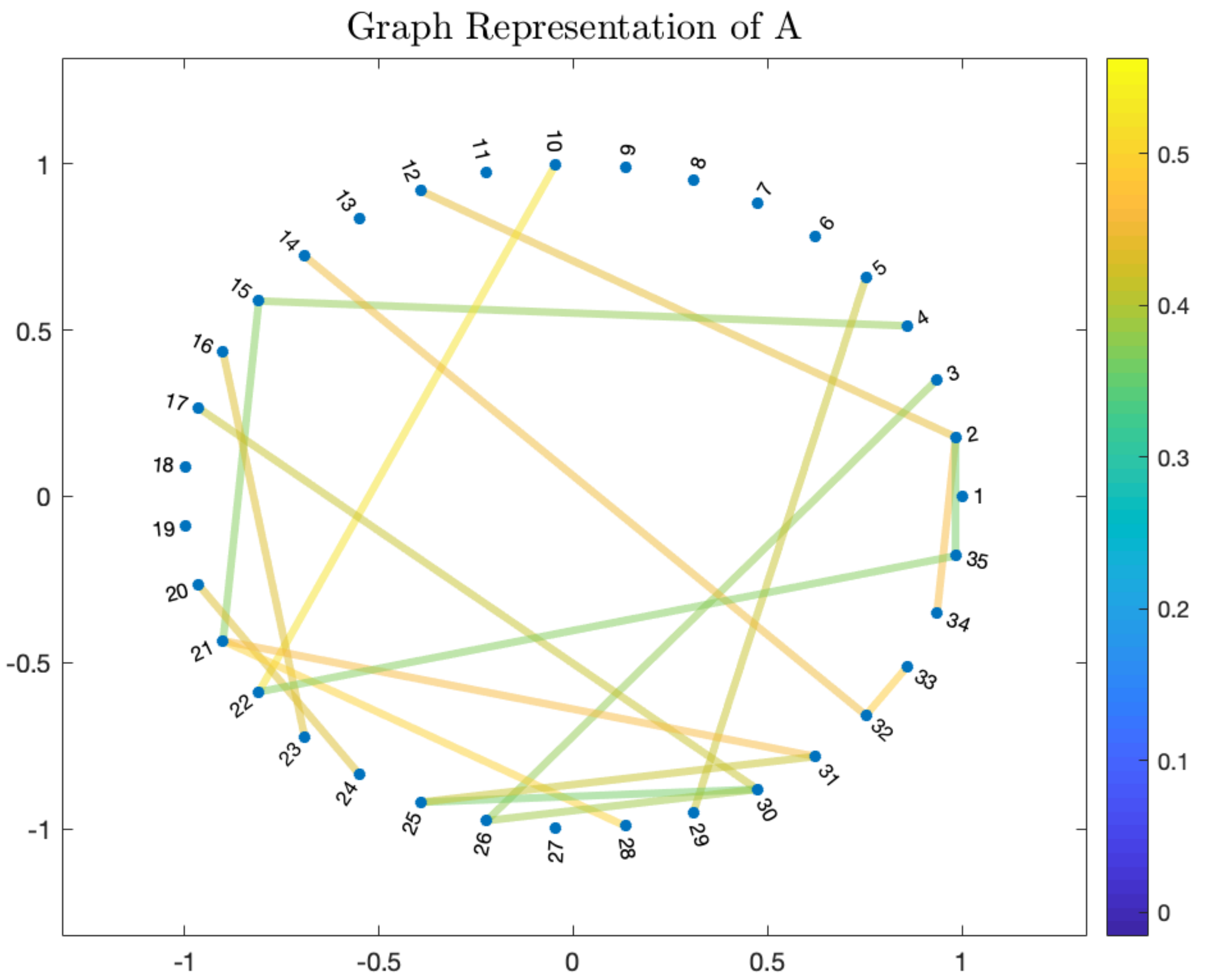
Recover $\widehat{\boldsymbol{X}}_S$ from $\widehat{\boldsymbol{A}}$ using SVAR



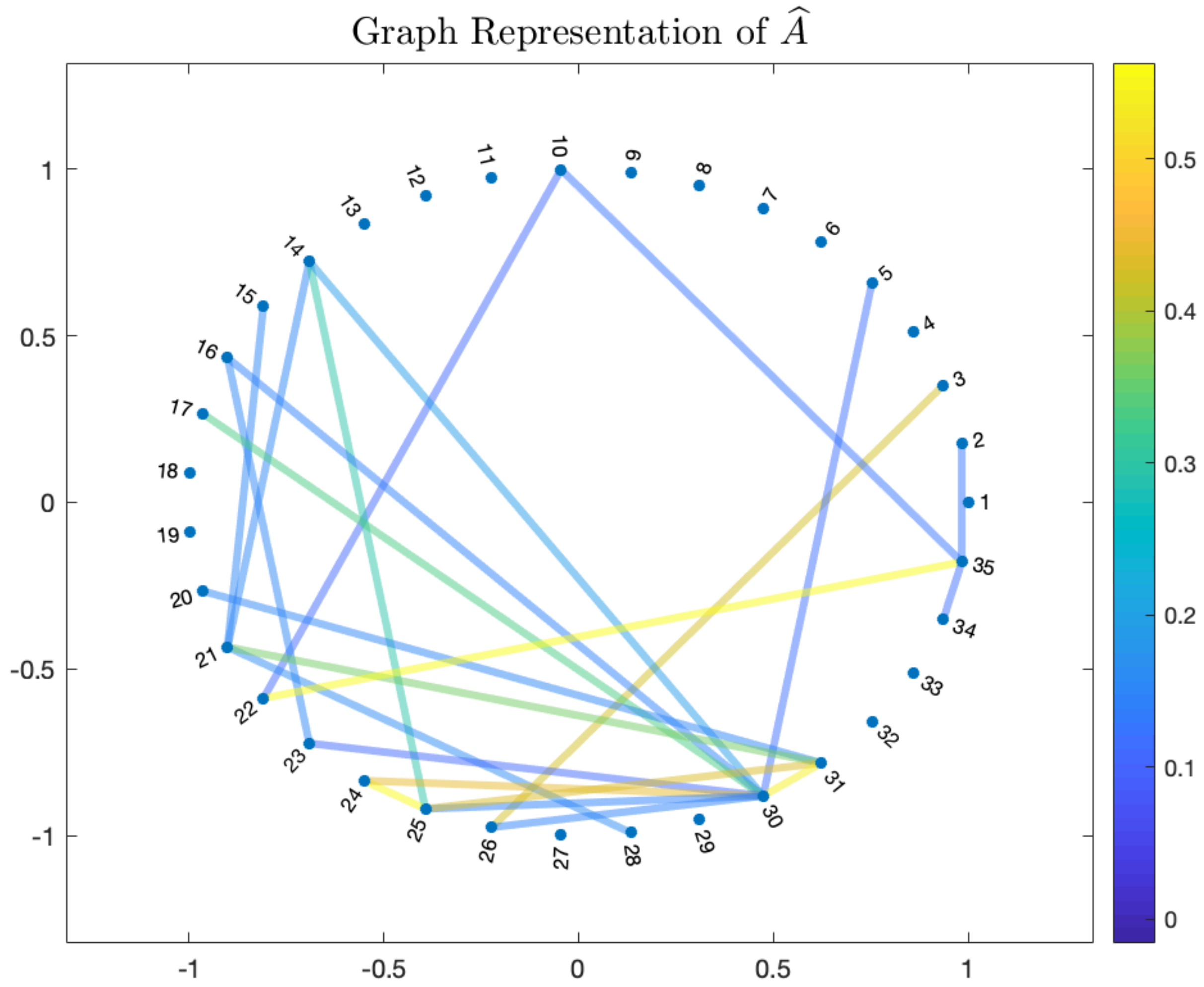
Recover $\hat{\boldsymbol{A}}$ from $\boldsymbol{X}_C$ using CGP



Recover $\widehat{A}$ from X_C using CGP

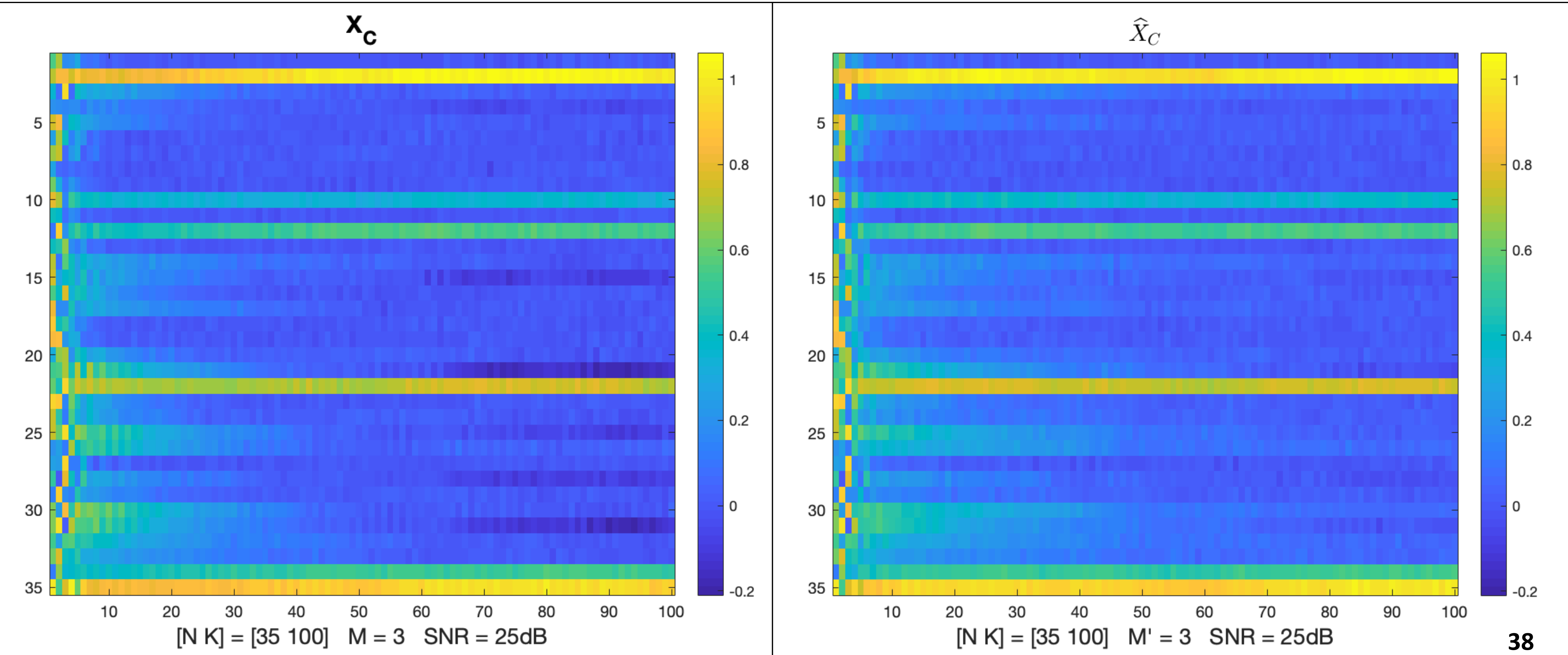


[N K] = [35 100]   M = 3   SNR = 25dB   Threshold = 0.11



[N K] = [35 100]   M' = 3   SNR = 25dB   Threshold = 0.11

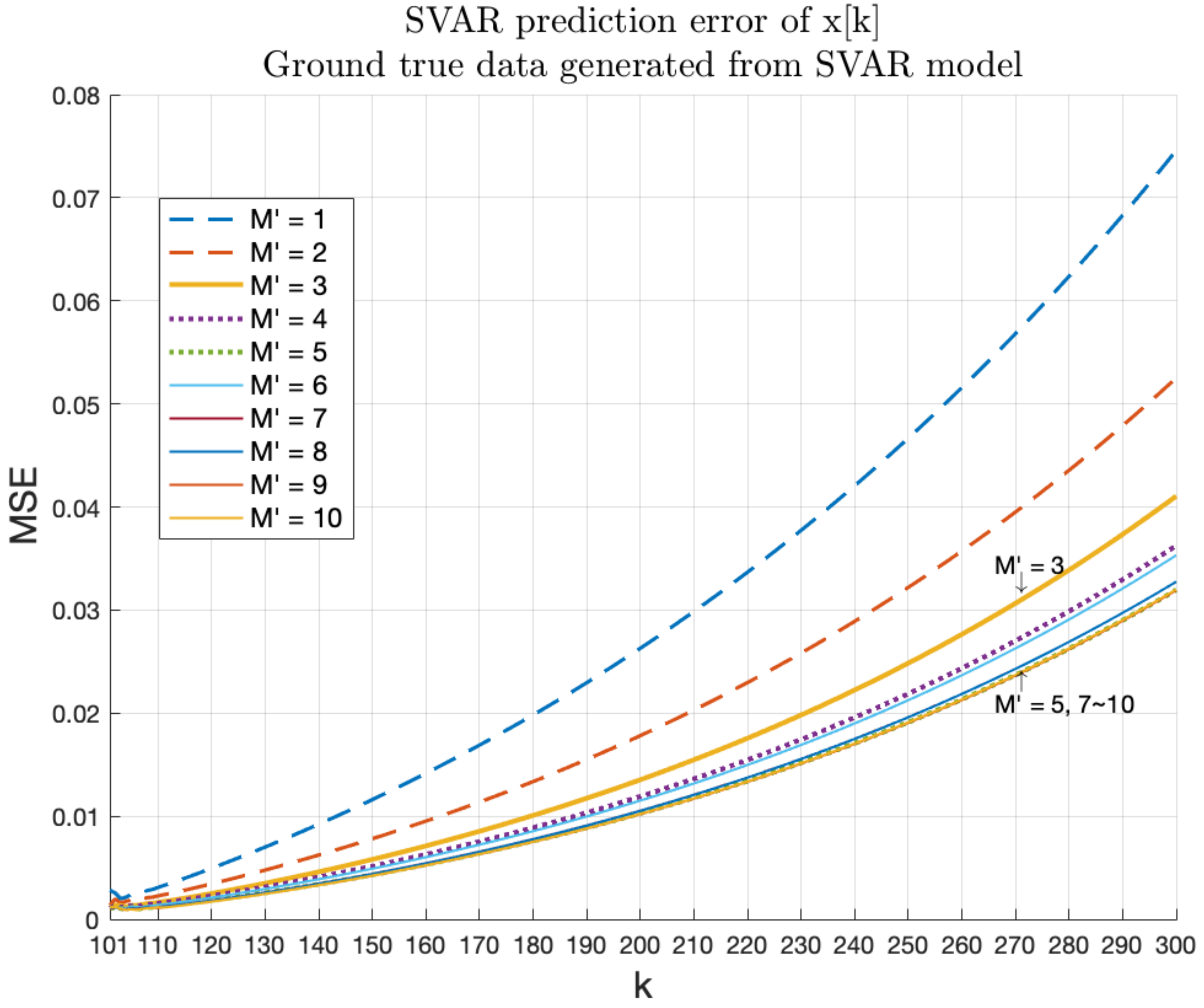
Recover $\widehat{X}_C$ from $\widehat{A}$ using CGP

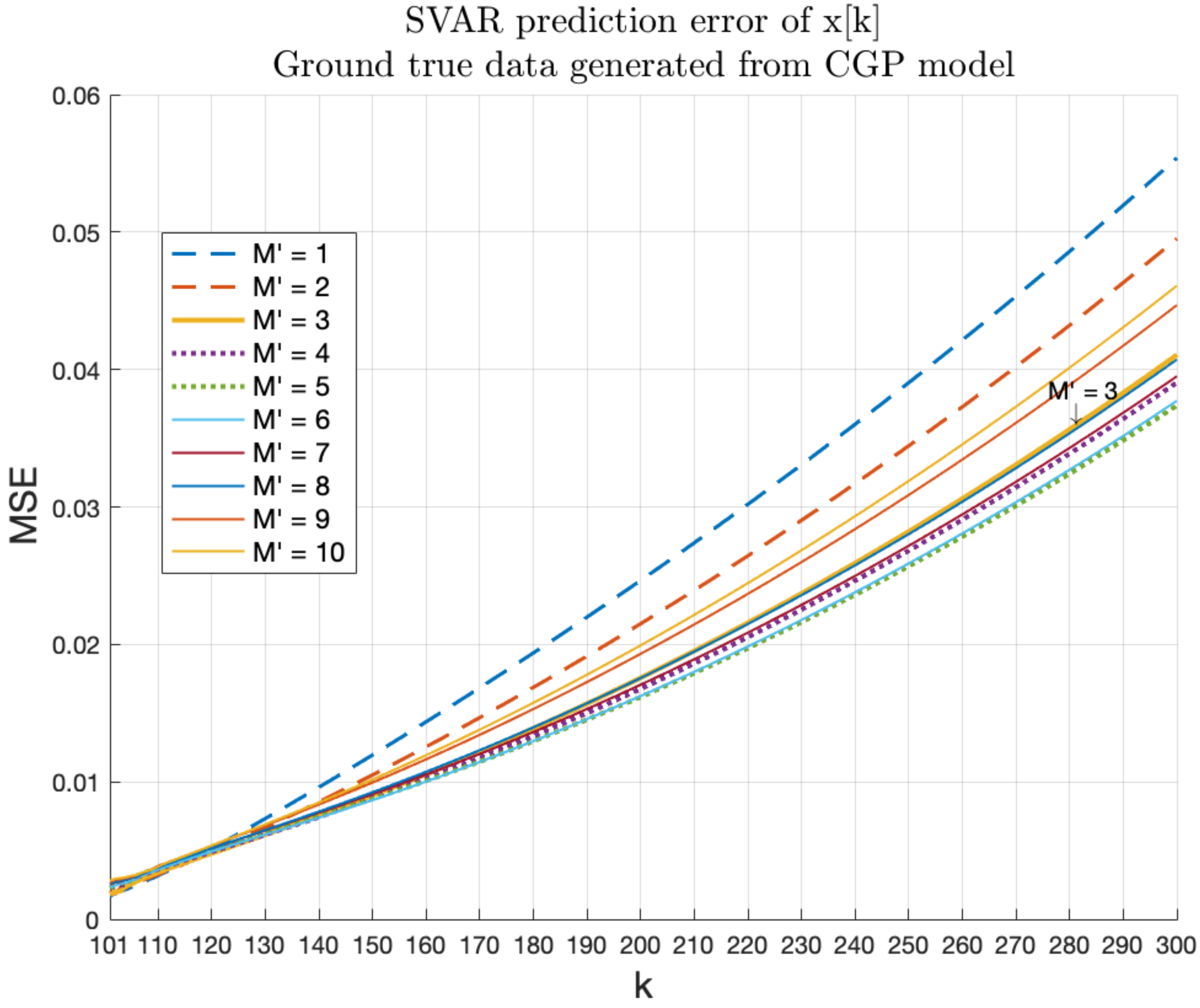


2. MSE error, underfitting, overfitting

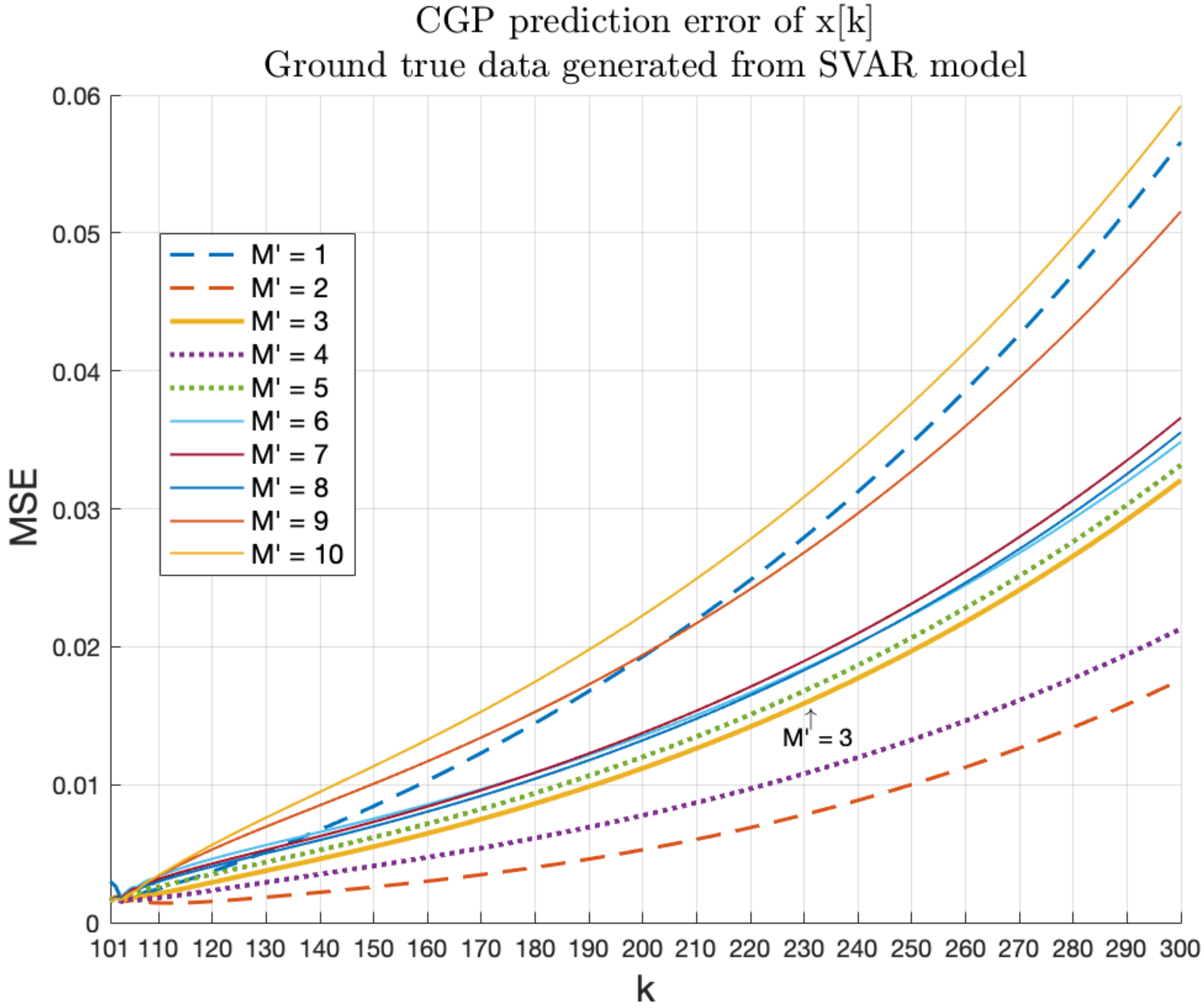
About this part

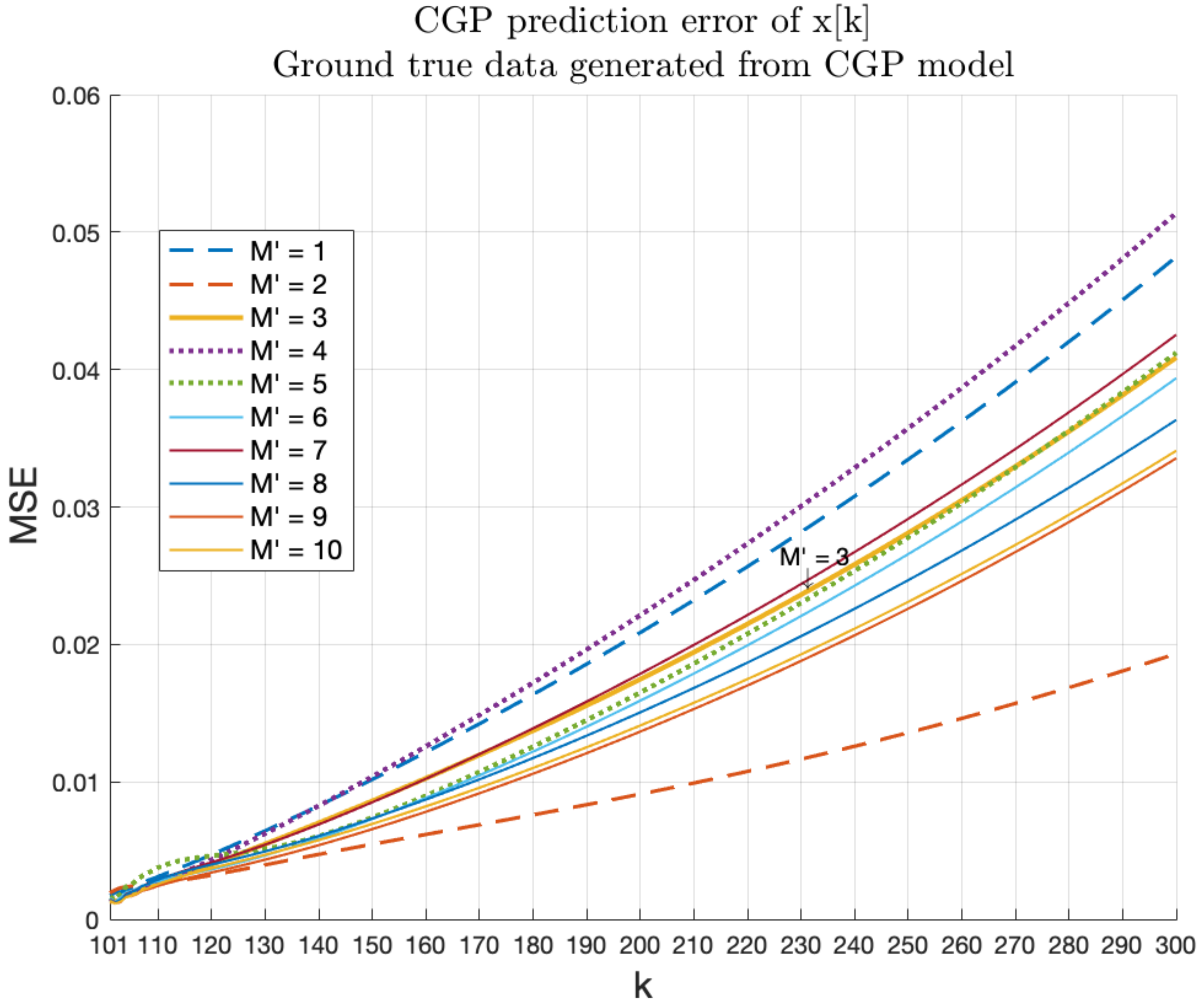
- Modeling the graph signals with different orders M' of SVAR and CGP model then predict on future graph signal.
- $M' > M$ corresponding the overfit and $M' < M$ corresponding to underfit.
- To be fair, using both SVAR and CGP to model $\boldsymbol{X}_S$ and $\boldsymbol{X}_C$.
- MSE error is given by : $MSE = \frac{1}{N}\|\boldsymbol{x}[k] - \widehat{\boldsymbol{x}}[k]\|_F$


SVAR prediction error of x[k]
Ground true data generated from SVAR model
MSE
k
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
101 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250 260 270 280 290 300
M' = 1
M' = 2
M' = 3
M' = 4
M' = 5
M' = 6
M' = 7
M' = 8
M' = 9
M' = 10
M' = 3
M' = 5, 7~10



SVAR prediction error of x[k]
Ground true data generated from CGP model
MSE
k
M' = 1
M' = 2
M' = 3
M' = 4
M' = 5
M' = 6
M' = 7
M' = 8
M' = 9
M' = 10
M' = 3
0.06
0.05
0.04
0.03
0.02
0.01
0
101 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250 260 270 280 290 300


CGP prediction error of x[k]
Ground true data generated from SVAR model
MSE
k
M' = 1
M' = 2
M' = 3
M' = 4
M' = 5
M' = 6
M' = 7
M' = 8
M' = 9
M' = 10
M' = 3
0.06
0.05
0.04
0.03
0.02
0.01
0
101
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300


CGP prediction error of x[k]
Ground true data generated from CGP model
MSE
k
M' = 1
M' = 2
M' = 3
M' = 4
M' = 5
M' = 6
M' = 7
M' = 8
M' = 9
M' = 10
M' = 3
0.06
0.05
0.04
0.03
0.02
0.01
0
101 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250 260 270 280 290 300

IV. Comparison & Conclusion

Comparison

- MSE error

  SVAR $>$ CGP(simplified) $>$ CGP(normal)

- Computational speed

  SVAR $>$ CGP(simplified) $>$ CGP(normal)

Conclusion

- CGP and SVAR model could be useful for solving the spatial-temporal interpolation problem, since they provides a very good way to model the relationship across time.
- There are several model to model the relationship across space as well, for solving interpolation problem.
- It we were able to put these two kind of model together, that would be a way to model the spatial-temporal relationship, which we believe haven’t be done by anyone so far.

Reference

1. J. Mei and J.M.F. Moura, “Signal processing on graphs: Causal modeling of unstructured data” IEEE Trans. On Signal Processing, vol. 65(8), pp. 2077−2092, 2017.
2. A. Davis, Richard & Zang, Pengfei & Zheng, Tian. (2012). “Sparse Vector Autoregressive Modeling.” Journal of Computational and Graphical Statistics. 30. 10.1080/10618600.2015.1092978.
3. Yuan, Ming and Yi Juain Lin. “Model selection and estimation in regression with grouped variables.” (2006).

Thanks