# Homogeneity fusion for Sparse Grouped Network VAR models

Panagiotis Andreou

University of North Carolina at Chapel Hill

*Joint work with Vladas Pipiras and Changryong Baek*

VTSS

April 10, 2025

# Motivation

- Multivariate time series show up in many modern applications:
  - Rumor spreading in a social network
  - Neural activity across brain regions
  - Evolution of stock prices for competing companies

# Motivation

- Multivariate time series show up in many modern applications:
  - Rumor spreading in a social network
  - Neural activity across brain regions
  - Evolution of stock prices for competing companies
- VAR(1) model: $\mathbf{Y}_t = B\mathbf{Y}_{t-1} + \boldsymbol{\varepsilon}_t$
- *Goal*: estimate the $d^2$ entries of $B$
- Sample size: $dT$

# Motivation

- Multivariate time series show up in many modern applications:
  - Rumor spreading in a social network
  - Neural activity across brain regions
  - Evolution of stock prices for competing companies
- VAR(1) model: $\mathbf{Y}_t = B\mathbf{Y}_{t-1} + \boldsymbol{\varepsilon}_t$
- *Goal*: estimate the $d^2$ entries of $B$
- Sample size: $dT$
- If $d < T$, use standard OLS estimator.

# Motivation

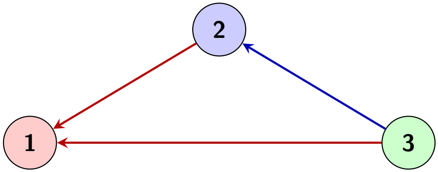
- Multivariate time series show up in many modern applications:
  - Rumor spreading in a social network
  - Neural activity across brain regions
  - Evolution of stock prices for competing companies
- VAR(1) model: $\mathbf{Y}_t = B\mathbf{Y}_{t-1} + \boldsymbol{\varepsilon}_t$
- *Goal*: estimate the $d^2$ entries of $B$
- Sample size: $dT$
- If $d < T$, use standard OLS estimator.
- If $d > T$, OLS is impossible. Instead, assume structure:
  - Sparsity (Lasso) [Basu and Michailidis, 2015]
  - Low Rank (factor models) [Stock and Watson, 2011]
  - Network models with few parameters [Zhu et al., 2017]

# Network Time Series

- Directed network $G$ captured by adjacency matrix $A = (a_{ij})_{1 \leq i,j \leq d}$
- Directed edge $a_{ij} = 1$ means "$i$ follows $j$"
- Each node has an associated time series

# Network Time Series

- Directed network $G$ captured by adjacency matrix $A = (a_{ij})_{1 \le i,j \le d}$
- Directed edge $a_{ij} = 1$ means "$i$ follows $j$"
- Each node has an associated time series




$$Y_{1,t} = \nu_1 Y_{1,t-1} + \varepsilon_{1,t}$$
$$Y_{2,t} = \beta_{21} Y_{1,t-1} + \nu_2 Y_{2,t-1} + \varepsilon_{2,t}$$
$$Y_{3,t} = \beta_{31} Y_{1,t-1} + \beta_{32} Y_{2,t-1} + \nu_3 Y_{3,t-1} + \varepsilon_{3,t}$$

# Related Work

- **Network VAR models** introduced by [Zhu et al., 2017], modeling temporal dependence through networks.

- Types of **group/community** structures incorporated in [Zhu and Pan, 2020, Chen et al., 2023, Zhu et al., 2023].

- **Unknown networks:** community detection in [Gudhmundsson and Brownlees, 2021] (Blockbuster algorithm) and network estimation in [Martin et al., 2024] (NIRVAR model).

# The GNAR model

- **Unknown** directed network with adjacency matrix $A = (a_{ij})_{1 \leq i,j \leq d}$.
- Latent partition $\mathbb{G} = \{G_1, \ldots, G_K\}$ of $\{1, \ldots, d\}$ into $K$ groups. We denote the group membership of node $i$ by $g_i$.

# The GNAR model

- **Unknown** directed network with adjacency matrix $A = (a_{ij})_{1 \le i,j \le d}$.
- Latent partition $\mathbb{G} = \{G_1, \ldots, G_K\}$ of $\{1, \ldots, d\}$ into $K$ groups. We denote the group membership of node $i$ by $g_i$.
- GNAR model [Zhu et al., 2023]:

$$Y_{i,t} = \sum_{j=1}^{d} \phi_{g_i g_j} \frac{a_{ij}}{n_i} Y_{j,t-1} + \nu_{g_i} Y_{i,t-1} + \varepsilon_{i,t}, \quad i \in [d],\ t \in [T],$$

where $n_i = \sum_{j=1}^{d} a_{ij}$ the out-degree of $i$, and $\varepsilon_{i,t} \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$.

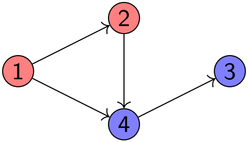
# The GNAR model

- **Unknown** directed network with adjacency matrix $A = (a_{ij})_{1 \leq i,j \leq d}$.
- Latent partition $\mathbb{G} = \{G_1, \ldots, G_K\}$ of $\{1, \ldots, d\}$ into $K$ groups. We denote the group membership of node $i$ by $g_i$.
- GNAR model [Zhu et al., 2023]:

$$Y_{i,t} = \sum_{j=1}^{d} \phi_{g_i g_j} \frac{a_{ij}}{n_i} Y_{j,t-1} + \nu_{g_i} Y_{i,t-1} + \varepsilon_{i,t}, \quad i \in [d],\ t \in [T],$$

  where $n_i = \sum_{j=1}^{d} a_{ij}$ the out-degree of $i$, and $\varepsilon_{i,t} \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$.
- *Remark:* When $K = 1$, the model reduces to the original model of [Zhu et al., 2017] and there are two parameters: network effect $\phi$ and autoregressive effect $\nu$.

# Example




Updating equations:

$$\begin{cases} Y_{1,t} = \nu_{1} Y_{1,t-1} + \dfrac{\phi_{11}}{2} Y_{2,t-1} + \dfrac{\phi_{12}}{2} Y_{4,t-1} + \varepsilon_{1,t} \\ Y_{2,t} = \nu_{1} Y_{2,t-1} + \phi_{12} Y_{4,t-1} + \varepsilon_{2,t} \\ Y_{3,t} = \nu_{2} Y_{3,t-1} + \varepsilon_{3,t} \\ Y_{4,t} = \phi_{22} Y_{3,t-1} + \nu_{2} Y_{4,t-1} + \varepsilon_{4,t} \end{cases}$$

# Fusion estimator

We want to solve the following optimization problem:

## Fusion estimator

$$\underset{\substack{A\in\{0,1\}^{d\times d},\mathbb{G}\in[K]^d \\ \Phi\in\mathbb{R}^{K\times K},\boldsymbol{\nu}\in\mathbb{R}^K}}{\operatorname{argmin}} \sum_{t=2}^{T}\sum_{i=1}^{d}\left(Y_{i,t}-\sum_{j=1}^{d}\phi_{g_i g_j}\frac{a_{ij}}{n_i}Y_{j,t-1}-\nu_{g_i}Y_{i,t-1}\right)^2$$

$$\text{s.t.}\quad \sum_{i=1}^{d}\sum_{j=1}^{d}a_{ij}\le s$$

# Theoretical goals

1. The misspecification probability for grouping/network goes to 0, i.e.,

$$\mathbb{P}(\mathbb{G}(\hat{B}) \neq \mathbb{G}(B^*)) \to 0, \quad \text{as } d, T \to \infty,$$

with an explicit convergence rate.

# Theoretical goals

1. The misspecification probability for grouping/network goes to 0, i.e.,

$$\mathbb{P}(\mathbb{G}(\hat{B}) \neq \mathbb{G}(B^*)) \to 0, \quad \text{as } d, T \to \infty,$$

   with an explicit convergence rate.

2. Estimation and prediction errors go to 0, i.e.,

$$\|\hat{\boldsymbol{\beta}}^{\mathsf{ol}} - \boldsymbol{\beta}^*\|_2 \to 0 \quad \text{and} \quad \|Z(\hat{\boldsymbol{\beta}}^{\mathsf{ol}} - \boldsymbol{\beta}^*)\|_2 \to 0,$$

   with specified convergence rates.

# Roadmap

1. Bound the misspecification probability:

$$\mathbb{P}\left(\mathbb{G}(\hat{B}) \neq \mathbb{G}(B^*)\right)$$

$$\leq \sum_{\omega \in \{\mathbb{G}(B) \mid B \in \Theta(K^*, s^*), \mathbb{G}(B) \neq \mathbb{G}(B^*)\}} \mathbb{P}\left( \min_{\substack{\widetilde{B} \in \Theta(K^*, s^*) \\ \mathbb{G}(\widetilde{B}) = \omega}} \|\mathbf{Y} - Z\widetilde{\boldsymbol{\beta}}\|_2^2 < \|\mathbf{Y} - Z\hat{\boldsymbol{\beta}}^{\mathrm{ol}}\|_2^2 \right)$$

# Roadmap

1. Bound the misspecification probability:

$$\mathbb{P}\left(\mathbb{G}(\hat{B}) \neq \mathbb{G}(B^*)\right)$$

$$\leq \sum_{\omega \in \{\mathbb{G}(B) | B \in \Theta(K^*, s^*), \mathbb{G}(B) \neq \mathbb{G}(B^*)\}} \mathbb{P}\left( \min_{\substack{\widetilde{B} \in \Theta(K^*, s^*) \\ \mathbb{G}(\widetilde{B}) = \omega}} \|\mathbf{Y} - Z\widetilde{\boldsymbol{\beta}}\|_2^2 < \|\mathbf{Y} - Z\hat{\boldsymbol{\beta}}^{\mathsf{ol}}\|_2^2 \right)$$

2. Handle each term

$$\mathbb{P}\left( \min_{\substack{\widetilde{B} \in \Theta(K^*, s^*) \\ \mathbb{G}(\widetilde{B}) = \omega}} \|\mathbf{Y} - Z\widetilde{\boldsymbol{\beta}}\|_2^2 < \|\mathbf{Y} - Z\hat{\boldsymbol{\beta}}^{\mathsf{ol}}\|_2^2 \right) = \mathbb{P}\left( \|\mathbf{Y} - Z\hat{\boldsymbol{\beta}}\|_2^2 < \|\mathbf{Y} - Z\hat{\boldsymbol{\beta}}^{\mathsf{ol}}\|_2^2 \right)$$

$$= \mathbb{P}\left(2\boldsymbol{\varepsilon}'(I - P_{\mathbb{G}(B)})Z\boldsymbol{\beta}^* + \|(I - P_{\mathbb{G}(B)})Z\boldsymbol{\beta}^*\|_2^2 - \boldsymbol{\varepsilon}'(P_{\mathbb{G}(B)} - P_{\mathbb{G}(B^*)})\boldsymbol{\varepsilon} < 0\right),$$

where $P_{\mathbb{G}(B)}$ is a $dT \times dT$ projection matrix including all lagged values.

# Remarks

1. In past work, $P_{\mathbb{G}(B)}$ was fixed. Here, we replace this by a population analog and use concentration inequalities to control the error.

# Remarks

1. In past work, $P_{\mathbb{G}(B)}$ was fixed. Here, we replace this by a population analog and use concentration inequalities to control the error.

2. The main technical difficulty lies in the dependence structure between $\boldsymbol{\varepsilon}$ and $Z$. To handle this, we carefully express $Z$ as function of $\boldsymbol{\varepsilon}$ and use Hanson-Wright on the residuals.

# Remarks

1. In past work, $P_{\mathbb{G}(B)}$ was fixed. Here, we replace this by a population analog and use concentration inequalities to control the error.

2. The main technical difficulty lies in the dependence structure between $\boldsymbol{\varepsilon}$ and $Z$. To handle this, we carefully express $Z$ as function of $\boldsymbol{\varepsilon}$ and use Hanson-Wright on the residuals.

3. *Why joint estimation?*

    - A natural alternative idea: estimate the network structure first, and conditionally on that estimate groups and parameters.

# Remarks

1. In past work, $P_{\mathbb{G}(B)}$ was fixed. Here, we replace this by a population analog and use concentration inequalities to control the error.

2. The main technical difficulty lies in the dependence structure between $\boldsymbol{\varepsilon}$ and $Z$. To handle this, we carefully express $Z$ as function of $\boldsymbol{\varepsilon}$ and use Hanson-Wright on the residuals.

3. *Why joint estimation?*
    - A natural alternative idea: estimate the network structure first, and conditionally on that estimate groups and parameters.
    - Conditional estimation does not guarantee solution to the optimization problem of interest.

# Remarks

1. In past work, $P_{\mathbb{G}(B)}$ was fixed. Here, we replace this by a population analog and use concentration inequalities to control the error.

2. The main technical difficulty lies in the dependence structure between $\boldsymbol{\varepsilon}$ and $Z$. To handle this, we carefully express $Z$ as function of $\boldsymbol{\varepsilon}$ and use Hanson-Wright on the residuals.

3. *Why joint estimation?*
   - A natural alternative idea: estimate the network structure first, and conditionally on that estimate groups and parameters.
   - Conditional estimation does not guarantee solution to the optimization problem of interest.
   - Our fusion estimator handles all components jointly, allowing for better theoretical guarantees.

# Mixed Integer Programming

- *Idea:* formulate the original optimization problem as a MIP and use Gurobi to solve it, as in [Bertsimas et al., 2016].

# Mixed Integer Programming

- *Idea:* formulate the original optimization problem as a MIP and use Gurobi to solve it, as in [Bertsimas et al., 2016].

- General MIP formulation:

$$\min \left( \mathbf{x}' Q \mathbf{x} + \mathbf{x}' \mathbf{y} \right) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{X},$$

  where $\mathcal{X}$ is the feasible set and the vector **x** includes:

  - *Binary variables:* $x_i \in \{0, 1\}$ for $i \in \mathcal{I}$
  - *Continuous variables:* $x_i \geq 0$ for $i \notin \mathcal{I}$

# MIP formulation of GNAR optimization problem

$$\underset{\substack{B\in\mathbb{R}^{d\times d},\, A\in\{0,1\}^{d\times d}\\ \Phi\in\mathbb{R}^{K\times K},\, \boldsymbol{\nu}\in\mathbb{R}^{K},\\ \Omega\in\{0,1\}^{d\times d\times K\times K}}}{\operatorname{argmin}} \sum_{t=2}^{T}\sum_{i=1}^{d}\left(Y_{i,t}-\sum_{j=1}^{d} b_{ij}Y_{j,t-1}\right)^2$$

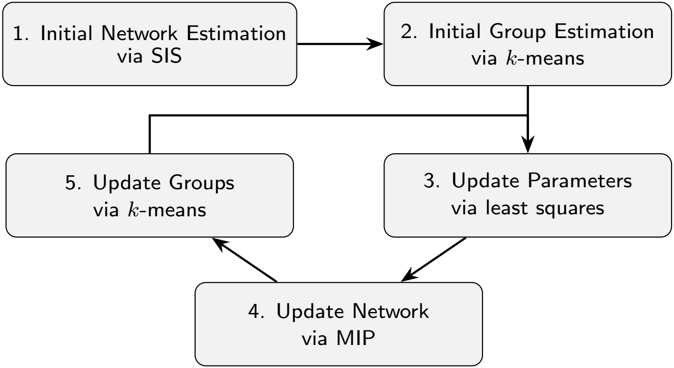
$$\begin{aligned}
\text{s.t.}\quad & a_{ij}\in\{0,1\},\, a_{ii}=0,\, n_i=\sum_{j=1}^{d} a_{ij},\quad i,j\in[d]\\
& \omega_{ijkl}\in\{0,1\},\quad i,j\in[d],\, k,l\in[K]\\
& \sum_{k=1}^{K}\sum_{j=1}^{d}\sum_{l=1}^{K}\omega_{ijkl}=1,\quad i\in[d]\\
& \sum_{j=1}^{d}\sum_{l=1}^{K}\omega_{ijkl}(b_{ii}-\nu_k)=0,\quad i\in[d],k\in[K]\\
& \omega_{ijkl}\left(n_i b_{ij}-\phi_{kl}a_{ij}\right)=0,\quad i,j\in[d],\, i\neq j,\, k,l\in[K]\\
& \sum_{i=1}^{d}\sum_{j=1}^{d} a_{ij}\leq s.
\end{aligned}$$

# Implementation

- The previous MIP is often intractable in practice (e.g., for $d = 100$ and $K = 3$, it has over 100k variables, mostly binary).

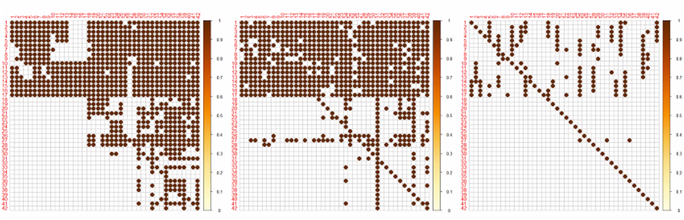
# Implementation

- The previous MIP is often intractable in practice (e.g., for $d = 100$ and $K = 3$, it has over 100k variables, mostly binary).
- Instead, follow the iterative scheme:



# Real Data: FRED-MD Example

- *Data:* 134 monthly U.S. macroeconomic variables.
- We focus on a subset of 42 variables in two groups:
  - Group 6: Interest & Exchange Rates (22 nodes)
  - Group 7: Prices (20 nodes)
- We treat variables as nodes in a network and apply our method to estimate:
  - The edge structure (adjacency matrix)
  - Group memberships



*Estimated adjacency matrices. Left: Fusion, Middle: Best Subset, Right: Adaptive Lasso*

# Conclusions

- We studied the problem of **homogeneity fusion** in the challenging setup of high-dimensional network time series with latent group structure.

# Conclusions

- We studied the problem of **homogeneity fusion** in the challenging setup of high-dimensional network time series with latent group structure.
- Our key contribution is a **joint estimation framework**, combining group structure, network topology, and autoregressive dynamics, with strong theoretical guarantees on misspecification error.

# Conclusions

- We studied the problem of **homogeneity fusion** in the challenging setup of high-dimensional network time series with latent group structure.
- Our key contribution is a **joint estimation framework**, combining group structure, network topology, and autoregressive dynamics, with strong theoretical guarantees on misspecification error.
- We formulated our problem as a mixed integer program and developed a practical **iterative algorithm** to solve this problem at scale.

# Conclusions

- We studied the problem of **homogeneity fusion** in the challenging setup of high-dimensional network time series with latent group structure.
- Our key contribution is a **joint estimation framework**, combining group structure, network topology, and autoregressive dynamics, with strong theoretical guarantees on misspecification error.
- We formulated our problem as a mixed integer program and developed a practical **iterative algorithm** to solve this problem at scale.
- Applied to real macroeconomic data, our method reveals interesting networks and groupings, complementing existing empirical approaches.

Basu, S. and Michailidis, G. (2015).
Regularized estimation in sparse high-dimensional time series models.
*The Annals of Statistics*, 43(4):1535 – 1567.

Bertsimas, D., King, A., and Mazumder, R. (2016).
Best subset selection via a modern optimization lens.
*The Annals of Statistics*, 44:813–852.

Chen, E. Y., Fan, J., and Zhu, X. (2023).
Community network auto-regression for high-dimensional time series.
*Journal of Econometrics*, 235(2):1239–1256.

Gudhmundsson, G. S. and Brownlees, C. (2021).
Detecting groups in large vector autoregressions.
*Journal of Econometrics*, 225(1):2–26.

Martin, B., Passino, F. S., Cucuringu, M., and Luati, A. (2024).
NIRVAR: Network informed restricted vector autoregression.
*arXiv preprint arXiv:2407.13314*.

Stock, J. H. and Watson, M. W. (2011).
Dynamic factor models.

Zhu, X. and Pan, R. (2020).
Grouped network vector autoregression.
*Statistica Sinica*, 30(3):1437–1462.

Zhu, X., Pan, R., Li, G., Liu, Y., and Wang, H. (2017).
Network vector autoregression.
*The Annals of Statistics*, 45(3):1096 – 1123.

Zhu, X., Xu, G., and Fan, J. (2023).
Simultaneous estimation and group identification for network vector autoregressive model with heterogeneous nodes.
*Journal of Econometrics*, page 105564.