Autoregressive models for matrix-valued time series

Antonio Fioravanti

RuhrMetrics Reading Group

January 13, 2025

Overview

In the previous episodes...

General VAR(1) model:

$$\underline{Y}_t = \mathbf{\Phi}\underline{Y}_{t-1} + \underline{\epsilon}_t$$

- $\underline{Y}_t$: $m \times 1$ vector of variables
- $\underline{Y}_{t-1}$: $m \times 1$ vector of lag-1 variables.
- $\mathbf{\Phi}$: $m \times m$ coefficient matrix
- $\underline{\epsilon}_t$: $m \times 1$ vector of shocks (innovations)
- $\text{Cov}(\underline{\epsilon}_t) := \mathbf{\Sigma}_\epsilon$

VAR(1) model structure

In case of a bivariate ($m = 2$) model, the previous VAR(1) can be written explicitly as:

$$\underbrace{\begin{bmatrix} y_{1t} \\ y_{2t} \end{bmatrix}}_{\underline{Y}_t} = \underbrace{\begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix}}_{\mathbf{\Phi}} \underbrace{\begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix}}_{\underline{Y}_{t-1}} + \underbrace{\begin{bmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{bmatrix}}_{\underline{\epsilon}_t}$$

or equivalently,

$$\begin{aligned} y_{1t} &= \phi_{11} y_{1,t-1} + \phi_{12} y_{2,t-1} + \epsilon_{1t} \\ y_{2t} &= \phi_{21} y_{1,t-1} + \phi_{22} y_{2,t-1} + \epsilon_{2t} \end{aligned}$$

What happens in case of multidimensional time series?

Consider Matrix valued data

Consider a DGP where the structure matters. For example, a time series of economic indicators ($m = 4$) measured across five countries ($n = 5$). They form a $m \times n$ matrix at each time point $t = 1, \ldots, T$.

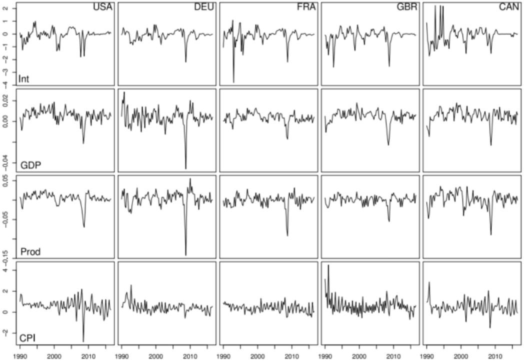


Fig. 1. Time series of four economic indicators: first differenced 3 month interbank interest rate, GDP growth (log difference), Total Manufacturing Production growth (log difference), and CPI core inflation growth (log difference) from five countries.

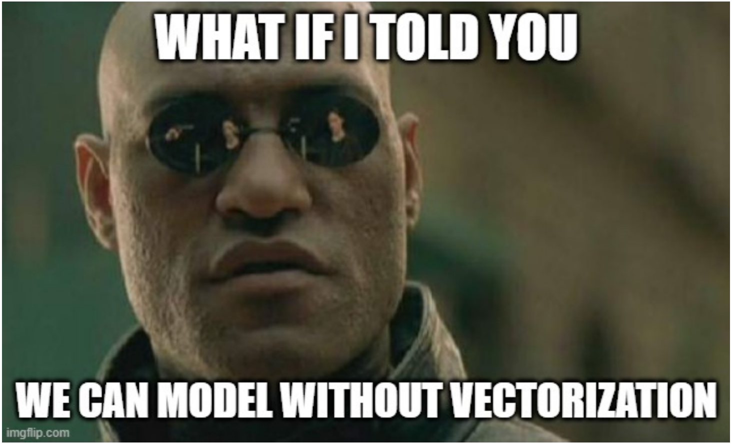
Motivation for Matrix valued time seris

Traditional VAR models handle the multidimensional component by vectorizing, i.e. stacking the variables in a vector.

Potential Drawbacks in Multidimensional Setting:

- **Vectorization**: the intrinsic structure of potentially multidimensional data is not preserved, due to the vectorization process.
- **Large number of parameters**: In a multidimensional setting the number of parameters to be estimated is high $m^2 n^2$.

"When you are ready, you won't have to"



Matrix Autoregressive (MAR) model

Let $\mathbf{X}_t$ be the $m \times n$ matrix observed at time t, then the **Matrix Autoregressive** (MAR) model (Chen, Xiao, and Yang 2021)is given by the bilinear form:

$$\underbrace{\mathbf{X}_t}_{m \times n} = \underbrace{\mathbf{A}}_{\color{blue}{m \times m}} \underbrace{\mathbf{X}_{t-1}}_{m \times n} \underbrace{\mathbf{B}'}_{\color{red}{n \times n}} + \underbrace{\mathbf{E}_t}_{m \times n}$$

- $\mathbf{X}_t$: observed matrix at time t.
- $\mathbf{X}_{t-1}$: observed matrix at time $t-1$ /lag-1 variables.
- $\mathbf{A}$: row-wise coefficient matrix.
- $\mathbf{B}$: column-wise coefficient matrix.
- $\mathbf{E}_t$: matrix of error terms / white noise.

Example: $m = 3, n = 2$

For $m = 3$ rows and $n = 2$ columns, the model becomes:

$$\underbrace{\begin{bmatrix} x_{11,t} & x_{12,t} \\ x_{21,t} & x_{22,t} \\ x_{31,t} & x_{32,t} \end{bmatrix}}_{\mathbf{X}_t} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}}_{\textcolor{blue}{\text{Row effects } (\mathbf{A})}} \cdot \underbrace{\begin{bmatrix} x_{11,t-1} & x_{12,t-1} \\ x_{21,t-1} & x_{22,t-1} \\ x_{31,t-1} & x_{32,t-1} \end{bmatrix}}_{\text{Lagged matrix } (\mathbf{X}_{t-1})} \cdot \underbrace{\begin{bmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \end{bmatrix}}_{\textcolor{red}{\text{Column effects } (\mathbf{B}')}}$$
$$+ \underbrace{\begin{bmatrix} e_{11,t} & e_{12,t} \\ e_{21,t} & e_{22,t} \\ e_{31,t} & e_{32,t} \end{bmatrix}}_{\text{Error terms } (\mathbf{E}_t)}.$$

Model Interpretations

Each entry $x_{ij,t}$ in $\mathbf{X}_t$ is influenced by:

$$x_{ij,t} = \sum_{k=1}^{m} \sum_{\ell=1}^{n} a_{ik}\, x_{k\ell,t-1}\, b_{j\ell} + e_{ij,t}.$$

Interpretations:

- **Row-wise effects (A)**:
  - a_{ik}: Influence of the Indicator k at time $t-1$ on Indicator i at time t.
  - **Example**: a_{21} How Interest Rate (column 1 of $\mathbf{A}$) at time $t-1$ affects GDP (row 2 of $\mathbf{A}$) at time t.
- **Column-wise effects (B)**:
  - $b_{j\ell}$: Influence of Country ℓ at time $t-1$ on Country j at time t.
  - **Example**: b_{13} How the indicators of Country 3 at time $t-1$ affect the indicators of Country 1 at time t.

Equivalence Between MAR(1) and VAR(1)

We can rewrite the MAR(1) model

$$\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1}\mathbf{B}' + \mathbf{E}_t$$

in the form of a VAR(1) model:

$$\mathrm{vec}(\mathbf{X}_t) = (\mathbf{B} \otimes \mathbf{A})\,\mathrm{vec}(\mathbf{X}_{t-1}) + \mathrm{vec}(\mathbf{E}_t)$$

Key Points:

- MAR(1) is a special case of the VAR(1) model
- $\mathbf{B} \otimes \mathbf{A}$: Kronecker product acts as the coefficient matrix $\mathbf{\Phi}$ in VAR(1).

General Assumptions on the Error Terms

A1: White Noise Error Matrix

The sequence $\{\mathbf{E}_t\}$ is assumed to be **Matrix White Noise**, with no temporal correlation:

$$\mathrm{Cov}(\mathbf{E}_t, \mathbf{E}_s) = 0, \quad \text{for } t \neq s.$$

Note: Entries of $\mathbf{E}_t$ are allowed to have concurrent correlations among its own entries.

Covariance of the Error Matrix:

- As a matrix, the covariances of $\mathbf{E}_t$ form a 4-dimensional tensor.
- Instead, $\mathrm{Cov}(\mathrm{vec}(\mathbf{E}_t)) =: \mathbf{\Sigma}$, a $mn \times mn$ matrix is used.
- The vectorization operator (vec) flattens the error matrix into a vector:

$$\mathrm{vec}(\mathbf{E}_t) \in \mathbb{R}^{mn}$$

- Σ captures all entry-wise correlations in $\mathbf{E}_t$ in matrix form.

Further Covariance Assumptions

Stronger Assumptions:

- **Independent Entries**: Covariance matrix $\mathbf{\Sigma}$ is diagonal.
- **Separable /Structured Covariance:**

$$\mathrm{Cov}(\mathrm{vec}(\mathbf{E}_t)) = \mathbf{\Sigma}_c \otimes \mathbf{\Sigma}_r,$$

  where:
  - $\mathbf{\Sigma}_r$: Row-wise covariances ($m \times m$).
  - $\mathbf{\Sigma}_c$: Column-wise covariances ($n \times n$).

  Interpretation Under Normality:

$$\mathbf{E}_t = \Sigma_r^{1/2}\mathbf{Z}_t\Sigma_c^{1/2}, \quad \mathbf{Z}_t \sim \mathcal{N}(0, \mathbf{I}_{mn}).$$

  where all the entries of $\mathbf{Z}_t$ are independent and following a standard Normal Distribution.

Identifiability Issue in MAR Models

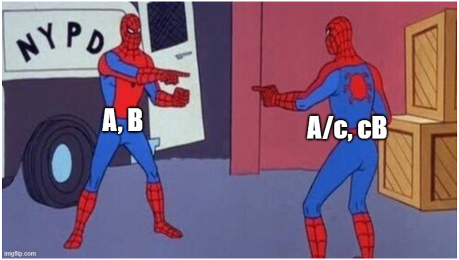
The MAR(1) model:

$$\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1}\mathbf{B}',$$

has an **identifiability issue**. Specifically, for any scalar $c \neq 0$:

$$\mathbf{A} \to \frac{\mathbf{A}}{c}, \quad \mathbf{B} \to c\mathbf{B},$$

leaves the model unchanged since the product $\mathbf{AB}'$ remains the same. This makes $\mathbf{A}$ and $\mathbf{B}$ non-unique.



Resolution to the Identifiability Issue

A2: Normalization of A

To resolve the scaling ambiguity, a normalization constraint is imposed on the matrix $\mathbf{A}$:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} a_{ij}^2} = 1.$$

This ensures that the scale of $\mathbf{A}$ is fixed, such that the matrix $\mathbf{B}$ adjusts accordingly to maintain the product $\mathbf{A}\mathbf{B}'$.

Normalization of $\mathbf{A}$ achieves the following:

- the model becomes **identifiable** and unique solutions for $(\mathbf{A}, \mathbf{B})$ are obtained.
- simplifies the estimation, as it **restricts the parameter space** to a well-defined region.

Stationarity and Causality of MAR(1)

Proposition 1: Stationarity and Causality

Let $\rho(\mathbf{C})$ be the spectral radius of any square matrix. If

$$\rho(\mathbf{A}) \cdot \rho(\mathbf{B}) < 1,$$

then the MAR(1) model is **stationary** and **causal** (or "stable" in Lütkepohl 2005).

$$\begin{aligned} \Gamma_k &:= \operatorname{Cov}\left(\operatorname{vec}\left(\boldsymbol{X}_t\right), \operatorname{vec}\left(\boldsymbol{X}_{t-k}\right)\right) \\ &= \sum_{l=0}^{\infty} \left(\boldsymbol{B}^{k+l} \otimes \boldsymbol{A}^{k+l}\right) \Sigma \left(\boldsymbol{B}^{l} \otimes \boldsymbol{A}^{l}\right)', k \geq 0 \end{aligned}$$

Proposition 1 guarantees that the infinite matrix series is summable.

Impulse Response Function (IRF) for VAR(1)

In the VAR(1) Model, the covariance matrix of the innovation undergoes orthogonalization via Cholesky decomposition to ensure that the shocks are uncorrelated:

$$\boldsymbol{\Sigma}_\epsilon = \mathsf{E}(\underline{\epsilon}_t \underline{\epsilon}_t') = \mathbf{P}\mathbf{P}'$$

in order to analyze the **Impulse Response Function**:

$$\mathsf{IRF}_j(k) = \boldsymbol{\Phi}^k \mathbf{P} \underline{s}_j.$$

where $\underline{s}_j = [0, 0, \ldots, 1, 0, \ldots, 0]'$ is the vector with the shock in the j-th variable.
It measures the effect of a one-unit shock in $\underline{\epsilon}_t$ on $\underline{y}_t$ over time lag k.

Key Points

- Dependent on variable ordering.

Impulse Response Function for MAR(1)

With a matrix data, it is difficult to order the innovations. Instead of relying on the ordering of the variables, only the series in position (i,j)th innovation is fixed, and all other innovations are orthogonalized.

shock-first Impulse Response Function with orthogonal innovations (s1-oIRF)

Let σ_{ij} be the (i,j)th entry of $\mathbf{\Sigma}$ Then, the Impulse Response Function (s1-oIRF) measuring the effect of a unit standard deviation shock in $e_{t,ij}$ (entry i,j of the error matrix $\mathbf{E}_t$) on $\mathbf{X}_t$ over time lag k is:

$$\mathbf{F}_{i,j}(k) = \left(\mathbf{B}^k \otimes \mathbf{A}^k\right) \mathbf{\Sigma}[\, , m(j-1)+i]$$

Key Points:

- The ordering of the variables does not matter in this case. Only the entry in position (i,j) is fixed for the orthogonalization.

IRF in Case of Separable Covariance

The impulse response function (IRF) exhibits a unique structure when the error covariance matrix $\mathbf{\Sigma} = \mathbf{\Sigma}_c \otimes \mathbf{\Sigma}_r$ is **separable**. Namely the IRF can be represented as:

$$\mathbf{F}_{ij}(k) = \left(\mathbf{B}^k \mathbf{\Sigma}_{c,j}\right) \otimes \left(\mathbf{A}^k \Sigma_{r,i}\right),$$

where $\mathbf{\Sigma} r, i$ and $\mathbf{\Sigma} c, j$ are the i-th and j-th columns of $\mathbf{\Sigma}_r$ and $\mathbf{\Sigma}_c$, respectively.

The IRF of the (i, j)-th series at lag k is:

$$f_{i,j}(k) = f_i^r(k) \cdot f_j^c(k) = \left(\mathbf{A}^k \mathbf{\Sigma}_{r,\cdot 1}\right)[i] \cdot \left(\mathbf{B}^k \mathbf{\Sigma} c, \cdot 1\right)[j].$$

Key Points:

- The IRF decomposes into row-wise effects $(\mathbf{A}^k \Sigma_r)$ and column-wise effects $(\mathbf{B}^k \Sigma_c)$.

Estimation Methods for MAR(1)

Overview: The paper proposes three primary methods for estimating the parameters of the MAR(1) model:

- **Projection Method:**
  - Estimates the coefficient matrices by projecting a general VAR(1) onto the space of Kronecker products.
- **Iterative Least Squares (LSE):**
  - Iteration between optimizing the **A** and **B** coefficient matrices.
- **Maximum Likelihood Estimation (MLE):**
  - Most efficient method in simulation.
  - Requires **structured covariance** assumptions (but works well even when this is not met in the simulation).

Projection Estimation (PROJ)

Solve the **nearest Kronecker Product (NKP)** problem (Van Loan 2000; Van Loan and Pitsianis 1993) starting with VAR(1) form:

$$\mathrm{vec}(\mathbf{X}_t) = (\mathbf{B} \otimes \mathbf{A})\,\mathrm{vec}(\mathbf{X}_{t-1}) + \mathrm{vec}(\mathbf{E}_t).$$

Objective:

$$\arg\min_{\mathbf{A},\mathbf{B}} \|\hat{\mathbf{\Phi}} - \mathbf{B} \otimes \mathbf{A}\|_F^2$$

Procedure:

1. Estimate $\mathbf{\Phi}$ with VAR(1) model
2. Project onto the space of Kronecker products.

Frame Title

Explicit Solution Exists:

1. Define **re-arrangement** operator $\mathcal{G} : \mathbb{R}^{mn} \times \mathbb{R}^{mn} \to \mathbb{R}^{m^2} \times \mathbb{R}^{n^2}$ such that $\mathcal{G}(\mathbf{B} \otimes \mathbf{A}) = \text{vec}(\mathbf{A})\text{vec}(\mathbf{B})$
2. Singular Value Decomposition of $\mathcal{G}(\hat{\mathbf{\Phi}}) = d_1 \underline{u}_1 \underline{v}_1^{'}$ is the solution for $\text{vec}(\hat{\mathbf{A}})\text{vec}(\hat{\mathbf{B}})$
3. Convert back into matrices to obtain $\hat{\mathbf{A}}_1$ and $\hat{\mathbf{B}}_1$
4. Normalize $\hat{\mathbf{A}}_1$ such that $\|\hat{\mathbf{A}}_1\|_F = 1$.

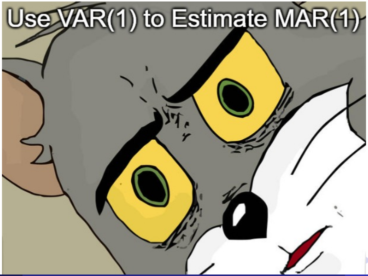
PROJ: Pros and Cons

Pros:

- **Closed form Solution**: due to Singular Value Decomposition (SVD).
- **Useful as Initialization:** for iterative methods like LSE or MLE.
- **Multiterm model**: can use the remaining singular values and vectors directly.

Cons:

- **Expensive Estimation**: m^2n^2 VAR(1) coefficient matrix $\mathbf{\Phi}$ is needed first.
- **Low Accuracy**: Highly dependent on the quality of $\hat{\mathbf{\Phi}}$ estimation.



Iterative Least Squares (LSE)

Objective:

$$\min_{\mathbf{A},\mathbf{B}} \sum_{t=1}^{T} \|\mathbf{X}_t - \mathbf{A}\mathbf{X}_{t-1}\mathbf{B}'\|_F^2 \iff \min_{\mathbf{A},\mathbf{B}} \|\mathcal{Y} - (\mathbf{B} \otimes \mathbf{A})\mathcal{X}\|_F^2$$

with:

$$\mathcal{Y} = [\mathrm{vec}(\boldsymbol{X}_2), \mathrm{vec}(\boldsymbol{X}_3), \ldots, \mathrm{vec}(\boldsymbol{X}_T)],$$

$$\mathcal{X} = [\mathrm{vec}(\boldsymbol{X}_1), \mathrm{vec}(\boldsymbol{X}_2), \ldots, \mathrm{vec}(\boldsymbol{X}_{T-1})]$$

Key points:

- **Inverse NKP problem:** It does not have an explicit SVD solution.
- **Solutions Exist**: the objective has at least one global minimum.
- **Unique Global Minimum**: if $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ solve the objective, so do $\tilde{\mathbf{A}} := \hat{\mathbf{A}}/c$, $\tilde{\mathbf{B}} := \hat{\mathbf{B}} \cdot c$, for $c \neq 0$. However, they yield the same matrix product $\hat{\mathbf{A}}\mathbf{X}_{t-1}\hat{\mathbf{B}}' = \tilde{\mathbf{A}}\mathbf{X}_{t-1}\tilde{\mathbf{B}}'$.

LSE: Condition and Iterative Procedure

Condition: If the innovations $\mathbf{E}_t$ are i.i.d. and absolutely continous w.r.t Lebesgue measure, then the solutions have full ranks.

Procedure:

- Start with some initial estimates $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ (e.g. PROJ).
- Alternate updates:

  Update **B** given **A**:

$$\boldsymbol{B} \leftarrow \left(\sum_t \boldsymbol{X}_t' \boldsymbol{A} \boldsymbol{X}_{t-1} \right) \left(\sum_t \boldsymbol{X}_t' \boldsymbol{A}' \boldsymbol{A} \boldsymbol{X}_{t-1} \right)^{-1}$$

  Update **A** given **B**:

$$\boldsymbol{A} \leftarrow \left(\sum_t \boldsymbol{X}_t \boldsymbol{B} \boldsymbol{X}_{t-1}' \right) \left(\sum_t \boldsymbol{X}_{t-1} \boldsymbol{B}' \boldsymbol{B} \boldsymbol{X}_{t-1}' \right)^{-1}$$

LSE: Pros and Cons

Pros:

- **General Covariance:** Does not assume any specific covariance structure for the error term $\mathbf{E}_t$.
- **Guaranteed Solution:** Minimizes the Frobenius norm - sum of quadratic terms is a convex function.

Cons:

- **No Closed-Form Solution:** Requires iterative optimization.
- **Local Minima Risk:** Solution may converge to a local minimum.

Maximum Likelihood Estimation (MLE) Procedure

Objective:

$$
\begin{aligned}
&-m(T-1)\log|\Sigma_c| - n(T-1)\log|\Sigma_r| \\
&-\sum_{t=1}^{T-1} \operatorname{tr}\left(\Sigma_r^{-1}(\mathbf{X}_t - \mathbf{A}\mathbf{X}_{t-1}\mathbf{B}')\Sigma_c^{-1}(\mathbf{X}_t - \mathbf{A}\mathbf{X}_{t-1}\mathbf{B}')'\right).
\end{aligned}
$$

is the Log-Likelihood under Normality.

Key points:

- **4 Matrix Parameters**: $\mathbf{A}, \mathbf{B}, \mathbf{\Sigma}_c, \mathbf{\Sigma}_r$.
- **Iterative**: Update one and keep the other three fixed.
- **Structured Covariance Assumption**: $\text{Cov}(\text{vec}(\mathbf{E}_t)) = \Sigma_c \otimes \Sigma_r$.

MLE: Iterative Procedure

1. Iteratively update one of the four:

$$\mathbf{A} \leftarrow \frac{\sum_{t=1}^{T-1} \mathbf{X}_t \Sigma_c^{-1} \mathbf{B} \mathbf{X}_{t-1}'}{\sum_{t=1}^{T-1} \mathbf{X}_{t-1} \mathbf{B}' \Sigma_c^{-1} \mathbf{B} \mathbf{X}_{t-1}'}$$

$$\mathbf{B} \leftarrow \frac{\sum_{t=1}^{T-1} \mathbf{X}_t' \Sigma_r^{-1} \mathbf{A} \mathbf{X}_{t-1}}{\sum_{t=1}^{T-1} \mathbf{X}_{t-1}' \mathbf{A}' \Sigma_r^{-1} \mathbf{A} \mathbf{X}_{t-1}}$$

$$\boldsymbol{\Sigma}_c \leftarrow \frac{\sum_{t=1}^{T-1} \mathbf{R}_t' \Sigma_r^{-1} \mathbf{R}_t}{m(T-1)}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{r}} \leftarrow \frac{\sum_{t=1}^{T-1} \mathbf{R}_t \Sigma_c^{-1} \mathbf{R}_t'}{n(T-1)}$$

2. Normalize $\|\mathbf{A}\|_F = 1$ and $\|\Sigma_r\|_F = 1$ for stability at each iteration.

Pros and Cons of Maximum Likelihood Estimation (MLE)

Pros:

- **Efficiency:** MLE is asymptotically the most efficient estimator when the covariance structure is correctly specified as **separable**.
- **Good even when not**: simulation studies from the authors show that it still performs well even if not separable.

Cons:

- **Computational Complexity:** Requires iterative updates, making it computationally intensive.
- **Correct Specification:** Relies on the correct specification of the covariance structure.

Simulation Settings

1. **Setting I**: Covariance matrix $\mathrm{Cov}(\mathrm{vec}(\mathbf{E}_t))$ is set to $\mathbf{\Sigma} = \mathbf{I}$.
2. **Setting II**: The covariance matrix $\mathrm{Cov}\left(\mathrm{vec}\left(\boldsymbol{E}_t\right)\right) = \mathbf{\Sigma}$ is randomly generated according to $\mathrm{Cov}\left(\mathrm{vec}\left(\boldsymbol{E}_t\right)\right) = \mathbf{Q} \wedge \mathbf{Q}'$, where the eigenvalues in the diagonal matrix Λ are the absolute values of i.i.d. standard normal random variates, and the eigenvector matrix $\mathbf{Q}$ is a random orthonormal matrix.
3. **Setting III**: The covariance matrix $\mathrm{Cov}\left(\mathrm{vec}\left(\boldsymbol{E}_t\right)\right)$ takes the Kronecker product form (4), where $\mathbf{\Sigma}_c$ and $\mathbf{\Sigma}_r$ are generated similarly as the $\mathbf{\Sigma}$ in Setting II.

Goal:

$$\log\left(\|\hat{\boldsymbol{B}} \otimes \hat{\boldsymbol{A}} - \boldsymbol{B} \otimes \boldsymbol{A}\|_F^2\right)$$

I guess (?) for VAR(1) (benchmark):

$$\log\left(\|\hat{\mathbf{\Phi}} - \mathbf{\Phi}\|_F^2\right)$$

Setting I

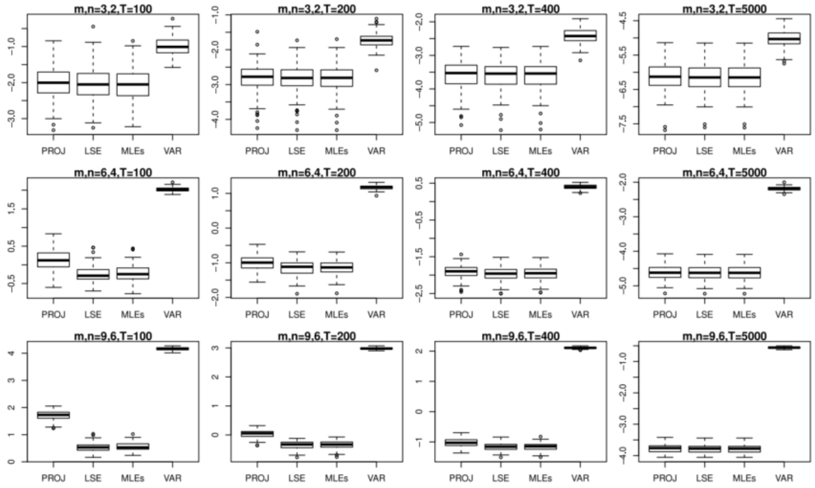


Fig. 2. Comparison of four estimators, PROJ, LSE, MLEs, and VAR, under Setting I. The three rows correspond to $(m, n) = (3, 2)$, $(6, 4)$ and $(9, 6)$ respectively, and the four columns $T = 100, 200, 400$ and 5000 respectively.

Setting II

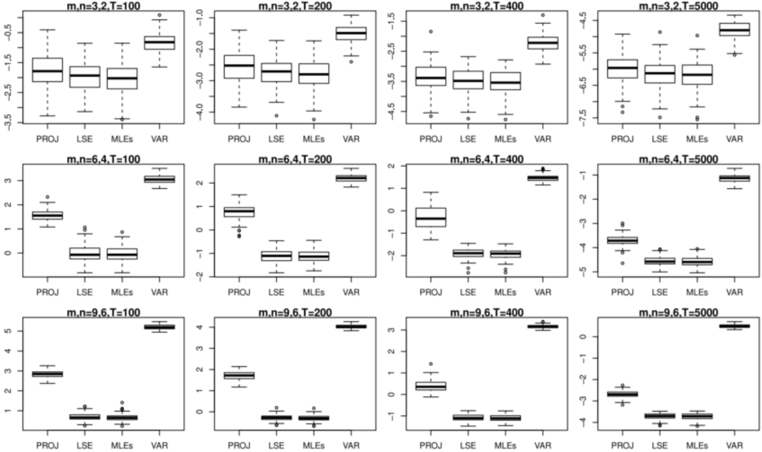


Fig. 3. Comparison of four estimators, PROJ, LSE, MLEs, and VAR, under Setting II. The three rows correspond to $(m, n) = (3, 2)$, $(6, 4)$ and $(9, 6)$ respectively, and the four columns $T = 100$, 200, 400 and 5000 respectively.

Setting III

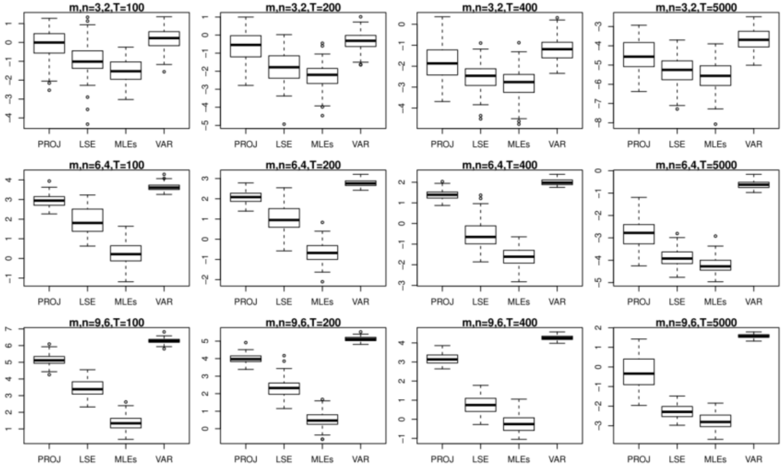
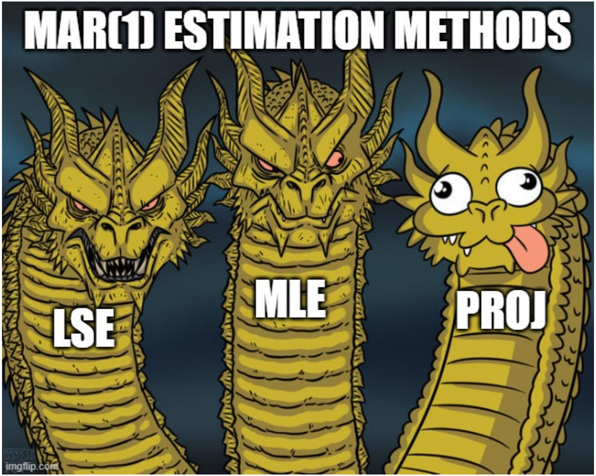


Fig. 4. Comparison of four estimators, PROJ, LSE, MLEs, and VAR, under Setting III. The three rows correspond to $(m, n) = (3, 2)$, $(6, 4)$ and $(9, 6)$ respectively, and the four columns $T = 100$, 200, 400 and 5000 respectively.

Winners?



Real Data Application of MAR(1)

Dataset Description:

- Quarterly observations of four economic indicators from 1991 to 2016.
- **4 Indicators**:
  - Interest Rate (3-month interbank, first-differenced).
  - GDP Growth (log difference, first-differenced).
  - Total Manufacturing Production Growth (log difference, first-differenced).
  - Consumer Price Index (CPI) Core Inflation (log difference, first-differenced).
- **5 Countries**: Canada, France, Germany, United Kingdom, United States.

Data Preprocessing:

- Adjusted for seasonality (quarterly means subtracted for CPI).
- Normalized each row's variance to 1 for comparability.

Left Coefficient Matrix **A** Results

Table: Estimated left coefficient matrix $\boldsymbol{A}$ of MAR(1) using LS method. Standard errors are shown in parentheses. The right panel indicates the positively significant, negatively significant, and insignificant parameters at 5% level using symbols $(+, -, 0)$, respectively.

	Int	GDP	Prod	CPI	Int	GDP	Prod	CPI
Int	0.177	0.215	0.132	-0.171	$+$	$+$	0	-
	(0.061)	(0.082)	(0.088)	(0.063)				
GDP	-0.190	0.341	0.346	-0.080	-	$+$	$+$	0
	(0.050)	(0.086)	(0.081)	(0.062)				
Prod	-0.223	0.318	0.424	-0.095	-	$+$	$+$	0
	(0.054)	(0.092)	(0.087)	(0.068)				
CPI	-0.028	0.048	-0.045	0.502	0	0	0	$+$
	(0.050)	(0.070)	(0.078)	(0.052)				

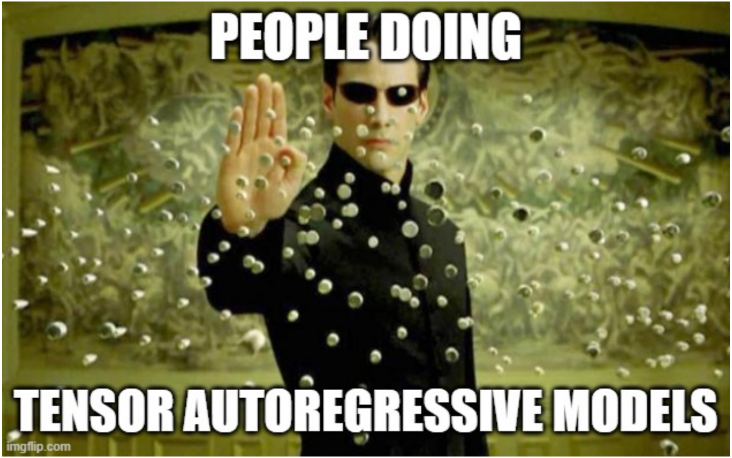
Right Coefficient Matrix **B** Results

Table: Estimated right coefficient matrix ***B*** of MAR(1) using LS method. Standard errors are shown in parentheses. The right panel indicates the positively significant, negatively significant, and insignificant parameters at 5% level using symbols $(+, -, 0)$, respectively.

	USA	DEU	FRA	GBR	CAN	USA	DEU	FRA	GBR	CAN
USA	0.878	-0.044	0.150	0.359	-0.043	+	0	0	+	0
	(0.134)	(0.202)	(0.138)	(0.132)	(0.156)					
DEU	0.722	0.072	0.801	0.308	-0.212	+	0	+	+	-
	(0.076)	(0.124)	(0.083)	(0.078)	(0.092)					
FRA	0.440	0.064	0.438	0.208	0.024	+	0	+	0	0
	(0.120)	(0.197)	(0.136)	(0.125)	(0.148)					
GBR	0.545	0.032	0.272	0.406	-0.018	+	0	+	+	0
	(0.089)	(0.153)	(0.101)	(0.101)	(0.118)					
CAN	0.553	0.023	-0.002	0.531	0.324	+	0	0	+	+
	(0.079)	(0.130)	(0.087)	(0.085)	(0.100)					

Is there an end to the madness?

Wang, Zheng, and Li 2024 have published a paper on *High-dimensional low-rank tensor autoregressive time series modeling*...



References

Van Loan, C. F., and N. Pitsianis. 1993. *Approximation with Kronecker products.* Springer.

Van Loan, C. F. 2000. "The ubiquitous Kronecker product." *Journal of computational and applied mathematics* 123 (1-2): 85–100.

Lütkepohl, H. 2005. *New introduction to multiple time series analysis.* Springer Science & Business Media.

Chen, R., H. Xiao, and D. Yang. 2021. "Autoregressive models for matrix-valued time series." *Journal of Econometrics* 222 (1): 539–560.

Wang, D., Y. Zheng, and G. Li. 2024. "High-dimensional low-rank tensor autoregressive time series modeling." *Journal of Econometrics* 238 (1): 105544. ISSN: 0304-4076.

The End ?