High-Dimensional Vector Autoregressive Time Series Modeling via Tensor Decomposition

Di Wang, Yao Zheng, Heng Lian, and Guodong Li (report by Ilia Lomasov)

December 7, 2023

1 Introduction

Disclaimer: symbol (∗) means there is a proof or just extra information in the appendix

VAR model (Lütkepohl 2005; Tsay 2010)

Consider the VAR model of the form:

$$\begin{gathered} y_t = A_1 y_{t-1} + \ldots + A_P y_{t-P} + \epsilon_t \\ \{y_t\}_{t=1}^T \in \mathbb{R}^N, \quad \{A_j\}_{j=1}^P \in \mathbb{R}^{N\times N} \\ \epsilon_t \in \mathbb{R}^N \overset{iid}{\sim} (\mathbf{0}, \Sigma_\epsilon), \Sigma_\epsilon \succ 0, \Sigma_\epsilon < \inf \end{gathered} \tag{1}$$

The issue with the standard VAR model is that it is difficult to perform estimation as number of parameters, N^2P, is very large. Even in low-dimensional setting, for example, $N = 5, P = 2$, the number of unknown parameters to estimate is already 50, hence, if a time series only has $T = 100$ observations, we already run into a curse of dimensionality problem. In the literature, there has been a number of alternatives to estimating N^2P parameters. PCA and Factor Models, for example, are the approaches we are familiar with because of STAT 556 course. All other approaches generally follow one of the two paths:

1. Assume sparsity of A_j and apply regularization (e.g., ℓ_1, LASSO or Dantzig selector). Papers on this topic include Basu & Michailidis 2015; Han, Lu & Liu 2015 and many others. Among the common drawbacks, this approach sacrifices certain temporal and cross-sectional dependencies. Moreover, according to Bai (1997), the average magnitude of parameters is bounded by $O(N^{-1/2})$, which limits sparsity-inducing regularization.

2. Reduced-rank regression approaches, that rely on assumption that $A^{(C)}$ in the following equation is a low-rank matrix.

$$y_t = (A_1 \ldots A_P)\left(y'_{t-1} \ldots y'_{t-P}\right)' + \epsilon_t =: A^{(C)} x_t + \epsilon_t \tag{2}$$

Papers on this topic include Velu and Reinsel 2013; Carriero, Kapetanios, and Marcellino 2011 (Bayesian extension) and many others. This approach, however, allows for only 1 type of low-rank structure, while the Tensor decomposition approach allows to restrict the parameter space along three directions simultaneously, thus drastically reducing the number of parameters needed to be estimated.

2 Tensor

3-Dimensional Tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$

A tensor of order M is an M-dimensional array of numbers. In the 2 simplest cases, 1-dimensional tensor is a vector, and a 2-dimensional tensor is a matrix. In this paper, authors were focused on tensors of order

3, which is an $I_1 \times I_2 \times I_3$ cuboid of numbers. Fibers are the vectors (column/mode-1, row/mode-2 or tube/mode-3) it consists of, just like a matrix consists of row or column vectors . And slices (horizontal, lateral or frontal) are the matrices it consists of (∗). $\{\mathcal{X}_{(n)}\}_{n=1}^{3}$ – **mode-n matricization** of $\mathcal{X}$. It is a rearrangement of mode-n fibers (vectors) into a matrix. $\mathcal{X}_{(n)} \in \mathbb{R}^{I_n \times (I_1 I_2 I_3 / I_n)}$(∗). To provide a simple example, let's look at matricization of a 2-dimensional tensor – a matrix. Let's consider $\mathcal{A} = A \in \mathbb{R}^{m \times n}$. Its row and column fibers are just its rows and columns respectively. Hence, rearranging its column fibers into a matrix yields A, and rearranging its row fibers into a matrix yields $A^\top$. (∗) Another tensor operation that is essential for this paper is **mode-n multiplication** defined as follows:

$$\underset{I_1 \times I_2 \times I_3}{\mathcal{X}} \times_n \underset{J \times I_n}{U} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 i_3} u_{j i_n} \ , \quad n \in \{1, 2, 3\}$$

In our matrix example, they can be thought of as equivalent to multiplications from the left and from the right (∗).

***I*mportant *F*acts (*IF*):**

1. For $m \neq n$, $\mathcal{X} \times_m A \times_n B = \mathcal{X} \times_n B \times_m A$ (∗)
2. $\mathcal{X} \times_n A \times_n B = \mathcal{X} \times_n (BA)$ (∗)
3. If $rank_n(\mathcal{Y}) < I_n$ for some $n \in \{1, 2, 3\}$, then there exists a Tucker decomposition $\mathcal{Y} = \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3$. (Notation: $[\![\mathcal{G}; U_1, U_2, U_3]\!]$). In 2-dimensional case, this is equivalent to low-rank matrix decomposition $A_{m \times n} = B_{m \times r} C_{r \times n}$, $r < \min\{m, n\}$.
4. (Following from 1 & 2): Tucker decomposition is not unique: for any non-singular matrices $\left\{O_n \in \mathbb{R}^{I_n \times I_n}\right\}_{n=1}^{3}$, $[\![\mathcal{G}; U_1, U_2, U_3]\!] = [\![\mathcal{G} \times_1 O_1 \times_2 O_2 \times_3 O_3; U_1 O_1^{-1}, U_2 O_2^{-1}, U_3 O_3^{-1}]\!]$. In 2-dimensional case, it is equivalent to $A = BC = (BO)(O^{-1}C)$.
5. (Kolda, 2006): $(\mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3)_{(1)} = U_1 \mathcal{G}_{(1)} (U_3 \otimes U_2)'$

3 Multilinear Low Rank (MLR) VAR

We can rearrange transition matrices $A_1, \ldots, A_P$ into a tensor $\mathcal{A} \in \mathbb{R}^{N \times N \times P}$, assuming those are its frontal slices. Below is the visualizaion.

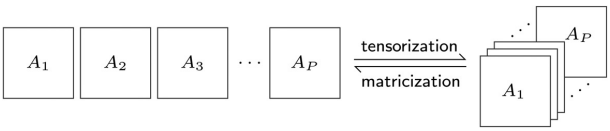

Figure 1: Tensorization from $A_1, \ldots, A_P$ to $\mathcal{A}$

Recall representation (2): $y_t = (A_1, \ldots, A_P)\, x_t + \epsilon_t$. It's equivalent to $y_t = \mathcal{A}_{(1)} x_t + \epsilon_t$. The key idea how to transition from that structure to tensors is to assume that there exists a Tucker decomposition $\mathcal{A} = \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3$. Then (2) can be rewritten as follows:

$$y_t = (\mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3)_{(1)}\, x_t + \epsilon_t \tag{3}$$

Authors of this paper used a special case of Tucker decomposition – high-order SVD (HOSVD) (De Lathauwer, DeMoor & Vandewalle 2000) (∗)

Number of parameters comparison

The number of parameters for regular VAR model is $N^2 P$ – we have to estimate N independent rows and NP independent columns. It is a cubical polynomial of N and P. In reduced rank regression setting,

matrix $A^{(c)}$ has reduced row rank, meaning only $(NP + N - r_1)r_1$ parameters have to be estimated (r_1 independent rows with NP elements each plus $N - r_1$ dependent rows, with r_1 dependency coefficients each). It is now a quadratic polynomial in N and P. Multilinear low rank setting allows to further reduce the number of parameters, specifically to $r_1r_2r_3 + (N - r_1)r_1 + (N - r_2)r_2 + (P - r_3)r_3$, which is now a linear polynomial in N and P.

Connection with Factor Models

Recall unknown factor model with r_1 common factors: $Y = F\Lambda' + E$, $F'F/T = \mathbb{I}_{r_1}$, $\Lambda'\Lambda$ – full-rank and diagonal. Using ***IF5***, we can rewrite (3) as $Y = X\left(U_3 \otimes U_2\right)\mathcal{G}'_{(1)}U'_1 + E$. Consider SVD: $\left(X\left(U_3 \otimes U_2\right)\mathcal{G}'_{(1)}\right) = U_x D_x V'_x$, where $D_x \in \mathbb{R}^{r_1 \times r_1}$ is diagonal, and U_x and V_x are orthonormal. Define $F = \sqrt{T}U_x$ and $\Lambda = U_1 V_x D_x/\sqrt{T}$. Note that $F'F/T = \mathbb{I}_{r_1}$ and that $\Lambda'\Lambda$ is diagonal. Thus, $Y = X\left(U_3 \otimes U_2\right)\mathcal{G}'_{(1)}U'_1 + E = F\Lambda' + E$. Although, the **key difference** is that MLR can be used directly for predictions, while a factor model cannot.

4 MLR estimation

The authors used a quadratic loss function (usage of alternative loss functions is outlined by the authors as one of the possible future research directions)

$$L\left(\mathcal{G}, U_1, U_2, U_3\right) = \frac{1}{T}\sum_{t=1}^{T}\left\|y_t - \left(\mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3\right)_{(1)} x_t\right\|_2^2$$

Low-dimensional case

The MLR LS estimator is $\widehat{\mathcal{A}}_{\text{MLR}} = \widehat{\mathcal{G}} \times_1 \widehat{U}_1 \times_2 \widehat{U}_2 \times_3 \widehat{U}_3 = \arg\min L\left(\mathcal{G}, U_1, U_2, U_3\right)$. Its asymptotic properties are derived using CLT (∗).
L – convex w.r.t any of $\mathcal{G}, U_1, U_2$, and U_3 when the other three are fixed. Therefore, the estimator itself is computed by Alternating Least Squares (ALS) algorithm (∗). Below is the general structure of ALS:
1. Initialize $\mathcal{A}^{(0)}$; perform HOSVD to obtain $U_1^{(0)}, U_2^{(0)}, U_3^{(0)}, \mathcal{G}^{(0)}$
2. Update individually $U_1^{(k+1)}, U_2^{(k+1)}, U_3^{(k+1)}, \mathcal{G}^{(k+1)}$ (in that order), other 3 fixed
3. When convergence reached, obtain $\widehat{\mathcal{A}}$
Authors recommend to initialize $\mathcal{A}^{(0)} = \widehat{\mathcal{A}}_{\text{prelim}} + T^{-1/2}\mathcal{T}$, where $\widehat{\mathcal{A}}_{\text{prelim}}$ is $\widehat{\mathcal{A}}_{\text{OLS}}$ for large T, $\widehat{\mathcal{A}}_{\text{RRR}}$ for small T, and $vec(\mathcal{T}) \sim N(0, \mathbb{I}_{NNP})$. Global minimum is not guaranteed.

High-dimensional case

For the high-dimensional case ($N, P \to \infty$), authors introduce regularization and all-orthogonality constraint. Sparse Higher-Order Reduced-Rank (SHORR) estimator is defined below:

$$\widehat{\mathcal{A}}_{\text{SHORR}} \equiv [\![\widehat{\mathcal{G}}; \widehat{U}_1, \widehat{U}_2, \widehat{U}_3]\!] = \underset{\mathcal{G}, U_1, U_2, U_3}{\arg\min}\left\{L\left(\mathcal{G}, U_1, U_2, U_3\right) + \lambda\left\|U_3 \otimes U_2 \otimes U_1\right\|_1\right\}$$

$$\text{subject to } U'_iU_i = \mathbb{I}_{r_i} \text{ and } \mathcal{G} \in \left\{\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3} : \left(\mathcal{G}_{(i)}\right)_{i=1}^{3} \text{ – row-orthogonal}\right\}$$

Sparsity assumption is introduced: each column of U_i has at most s_i nonzero entries. The key difference with LASSO is that $\mathcal{A}$ is not required to be sparse now, which lets us avoid the related problems identified in the introduction section of this report.

Developing an algorithm for SHORR

The **issue** is that ℓ_1 regularization is non-smooth while orthogonality constraint is non-convex. The **idea** is to assume a decomposition $\mathcal{G}_{(i)} = D_i V_i'$ exists, where $D_i \in \mathbb{R}^{r_i \times r_i}$, $V_i \in \mathbb{R}^{(r_1 r_2 r_3 / r_i) \times r_i}$, $V_i' V_i = \mathbb{I}_{r_i}$. We can write the Augmented Lagrangian in this case:

$$\mathcal{L}_\varrho \left(\mathcal{G}, \{U_i\}, \{D_i\}, \{V_i\}; \{\mathcal{C}_i\} \right) = L\left(\mathcal{G}, U_1, U_2, U_3 \right) + \lambda \left\| U_3 \otimes U_2 \otimes U_1 \right\|_1$$
$$+ 2 \sum_{i=1}^{3} \varrho_i \left\langle \left(\mathcal{C}_i \right)_{(i)}, \mathcal{G}_{(i)} - D_i V_i' \right\rangle + \sum_{i=1}^{3} \varrho_i \left\| \mathcal{G}_{(i)} - D_i V_i' \right\|_{\mathrm{F}}^2$$

Where ϱ_i – regularization constants, $\mathcal{C}_i \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ – dual variables.
Below is ADMM algorithm (Boyd, 2011) for solving this optimization problem. As for initialization, authors recommend to choose $\mathcal{A}^{(0)} = \widehat{\mathcal{A}}_{\mathrm{NN}} + (NP/T)^{1/2} \mathcal{T}$, where entries of $\mathcal{T}$ are iid random Standard Gaussian, and $\widehat{\mathcal{A}}_{\mathrm{NN}}$ – Nuclear Norm (NN) estimator (*)

Initialize: $\mathcal{A}^{(0)}$
HOSVD: $\mathcal{A}^{(0)} \approx \mathcal{G}^{(0)} \times_1 U_1^{(0)} \times_2 U_2^{(0)} \times_3 U_3^{(0)}$ with multilinear ranks (r_1, r_2, r_3).
repeat
  $U_1^{(k+1)} \leftarrow \underset{U_1' U_1 = I_{r_1}}{\arg\min} \left\{ L(\mathcal{G}^{(k)}, U_1, U_2^{(k)}, U_3^{(k)}) + \lambda \|U_1\|_1 \|U_2^{(k)}\|_1 \|U_3^{(k)}\|_1 \right\}$
  $U_2^{(k+1)} \leftarrow \underset{U_2' U_2 = I_{r_2}}{\arg\min} \left\{ L(\mathcal{G}^{(k)}, U_1^{(k+1)}, U_2, U_3^{(k)}) + \lambda \|U_1^{(k+1)}\|_1 \|U_2\|_1 \|U_3^{(k)}\|_1 \right\}$
  $U_3^{(k+1)} \leftarrow \underset{U_3' U_3 = I_{r_3}}{\arg\min} \left\{ L(\mathcal{G}^{(k)}, U_1^{(k+1)}, U_2^{(k+1)}, U_3) + \lambda \|U_1^{(k+1)}\|_1 \|U_2^{(k+1)}\|_1 \|U_3\|_1 \right\}$
  $\mathcal{G}^{(k+1)} \leftarrow \arg\min \left\{ L(\mathcal{G}, U_1^{(k+1)}, U_2^{(k+1)}, U_3^{(k+1)}) + \sum_{i=1}^{3} \varrho_i \|\mathcal{G}_{(i)} - D_i^{(k)} V_i^{(k)\prime} + (\mathcal{C}_i^{(k)})_{(i)}\|_{\mathrm{F}}^2 \right\}$
  for $i \in \{1, 2, 3\}$ **do**
    $D_i^{(k+1)} \leftarrow \underset{D_i = \mathrm{diag}(d_i)}{\arg\min} \|\mathcal{G}_{(i)}^{(k+1)} - D_i V_i^{(k)\prime} + (\mathcal{C}_i^{(k)})_{(i)}\|_{\mathrm{F}}^2$
    $V_i^{(k+1)} \leftarrow \underset{V_i' V_i = I_{r_i}}{\arg\min} \|\mathcal{G}_{(i)}^{(k+1)} - D_i^{(k+1)} V_i' + (\mathcal{C}_i^{(k)})_{(i)}\|_{\mathrm{F}}^2$
    $(\mathcal{C}_i^{(k+1)})_{(i)} \leftarrow (\mathcal{C}_i^{(k)})_{(i)} + \mathcal{G}_{(i)}^{(k+1)} - D_i^{(k+1)} V_i^{(k+1)\prime}$
  end for
  $\mathcal{A}^{(k+1)} \leftarrow \mathcal{G}^{(k+1)} \times_1 U_1^{(k+1)} \times_2 U_2^{(k+1)} \times_3 U_3^{(k+1)}$
until convergence

From the equations above we can see that updating $\mathcal{G}, D_i, V_i$ is an LS problem, but updating U_i directly is very complicated. Original problem: $U_i = \arg\min_B \left\{ n^{-1} \|y - X \operatorname{vec}(B)\|_2^2 + \lambda \|B\|_1 \right\}$ s.t. $B'B = \mathbb{I}$. The **idea** is to separate orthogonality and regularization: $\min_B \left\{ n^{-1} \|y - X \operatorname{vec}(B)\|_2^2 + \lambda \|W\|_1 \right\}$ s.t. $B'B = \mathbb{I}$, $B = W$
Augmented Lagrangian (M – dual variable): $n^{-1} \|y - X \operatorname{vec}(B)\|_2^2 + \lambda \|W\|_1 + 2\kappa \langle M, B - W \rangle + \kappa \|B - W\|_{\mathrm{F}}^2$
Apply ADMM once again to find $B = W = U_i$:

Initialize: $\boldsymbol{B}^{(0)} = \boldsymbol{W}^{(0)}, \boldsymbol{M}^{(0)} = \boldsymbol{0}$
repeat
  $\boldsymbol{B}^{(k+1)} \leftarrow \arg\min_{\boldsymbol{B}'\boldsymbol{B} = \boldsymbol{I}} \left\{ n^{-1} \|\boldsymbol{y} - \boldsymbol{X}\mathrm{vec}(\boldsymbol{B})\|_2^2 + \kappa \|\boldsymbol{B} - \boldsymbol{W}^{(k)} + \boldsymbol{M}^{(k)}\|_{\mathrm{F}}^2 \right\}$
  $\boldsymbol{W}^{(k+1)} \leftarrow \arg\min_{\boldsymbol{W}} \left\{ \kappa \|\boldsymbol{B}^{(k+1)} - \boldsymbol{W} + \boldsymbol{M}^{(k)}\|_{\mathrm{F}}^2 + \lambda \|\boldsymbol{W}\|_1 \right\}$
  $\boldsymbol{M}^{(k+1)} \leftarrow \boldsymbol{M}^{(k)} + \boldsymbol{B}^{(k+1)} - \boldsymbol{W}^{(k+1)}$
until convergence

Therefore, SHORR estimator is computed by this 2-stage ADMM algorithm. Under certain conditions on $\mathcal{L}_\varrho$, algorithm converges to local minimum of our objective function.

Rank selection

Let $\widehat{\mathcal{A}}$ be a consistent initial estimator of $\mathcal{A}$ (e.g., $\widehat{\mathcal{A}}_{\text{NN}}$). Then we can use Ridge-type ratio estimator (Xia, Xu, and Zhu 2015):

$$\widehat{r}_i = \arg\min_{1\le j\le p_i-1} \frac{\sigma_{j+1}\left(\widehat{\mathcal{A}}_{(i)}\right)+c}{\sigma_j\left(\widehat{\mathcal{A}}_{(i)}\right)+c} \quad \text{where } p_1=p_2=N, p_3=P \text{ and } \sigma_j - \text{j-th largest singular value}$$

The logic behind this procedure is that if rank of a matrix is r, then its $(r+1)$–st largest singular value should be zero, and hence $\frac{\sigma_{r+1}(\widehat{\mathcal{A}}_{(i)})}{\sigma_r(\widehat{\mathcal{A}}_{(i)})} \approx 0$, so, r_i that minimizes the Ridge-type ratio is likely the true mode-i rank.
c needs to be chosen very carefully (∗). Authors recommend to choose $c = \sqrt{NP\ln(T)/(10T)}$

5 Trying the model on simulations and real data

Rank selection consistency – simulation

Let $(N,P)=(10,5)$, $(r_1,r_2,r_3)=(3,3,3)$, and $\epsilon_t \overset{\text{iid}}{\sim} N(\mathbf{0},\mathbb{I}_N)$. $\mathcal{G}$ – a diagonal cube with $(\mathcal{G}_{111},\mathcal{G}_{222},\mathcal{G}_{333}) = (2,2,2)$ (case a), $(4,3,2)$ (case b), $(1,1,1)$ (case c), or $(2,1,0.5)$ (case d). Then nonzero singular values of $\mathcal{A}_{(i)}$ are $\mathcal{G}_{111},\mathcal{G}_{222}$, and $\mathcal{G}_{333}$. Generate U_i's as the first r_i left singular vectors of Gaussian random matrices while ensuring the stationarity.

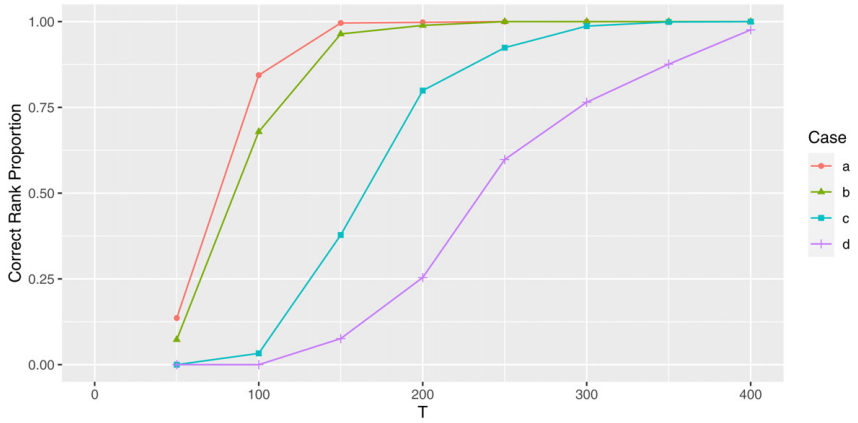

Figure 2: Proportion of correct rank out of 1000 replications for each $T \in \{50, 100, \ldots, 400\}$ in each case

From this simulation we can see that rank selection procedure is not always accurate, especially when singular values of matricizations are small and different from each other (case d), and especially for smaller T (200). However, it is still very accurate for other cases, and for larger T.
Remark: The outcomes of other simulations can be found in appendix.

Modeling real data

Data: $N = 40$ quarterly macroeconomic sequences of the United States from 1959 to 2007 (from Koop, 2013). Lag $P = 4$ for the VAR model is suggested by that same paper. $N >> P \Rightarrow$ penalty on U_3 is not needed. New penalty – $\|U_2 \otimes U_1\|_1$. $(r_1,r_2,r_3) = (4,3,2)$ are selected by the ridge-type ratio estimator. Tuning parameter λ is selected by BIC.

	Unregularized methods				Regularized methods				
Criterion	OLS	RRR	DFM	MLR	SHORR	LASSO	NN	RSSVD	SOFAR
ℓ_2 norm	20.16	13.31	6.36	**5.81**	5.35	6.72	8.16	6.33	6.28
ℓ_∞ norm	8.32	4.55	2.85	**2.56**	2.44	3.06	3.36	3.02	3.02

Figure 3: Forecasting errors for different existing methods

MLR performed the best among unregularized methods, SHORR – among regularized. NN performs the worst, which might be a sign that some other initialization might be used for SHORR and for rank selection to make the algorithms better.

6 Conclusion

Remark: The drawbacks that I found to this paper are highlighted in purple, with possible solutions.

The novelty of the approach is in its ability to jointly enforce three different reduced-rank structures at the same time. Not only is it more computationally efficient, the simulations and real data show that MLR estimation is also the most accurate among its peers.
One of the issues is that the order P of VAR is not estimated. Possible solution to that is to use Information Criteria for model selection. Another issue is that Selecting r_i is dependent on initialization $\widehat{\mathcal{A}}_0$, derived from other methods and which can even be consistent but biased/inefficient and hence make low-T estimation incorrect. IC-based selection or hypothesis testing is problematic in this case (3 parameters, too many combinations).
NN estimator perform the worst in both simulations and real data. This is connected to the previous point, and both have already been outlined at the end of the previous section. A possible solution is to use other estimators at initialization $\widehat{\mathcal{A}}_0$ (e.g., SOFAR or RSSVD).
However, despite those outlined limitations, MLR and SHORR perform the best among the existing models, both on the simulated and real macroeconomic data.

7 Appendix

Tensor visualization

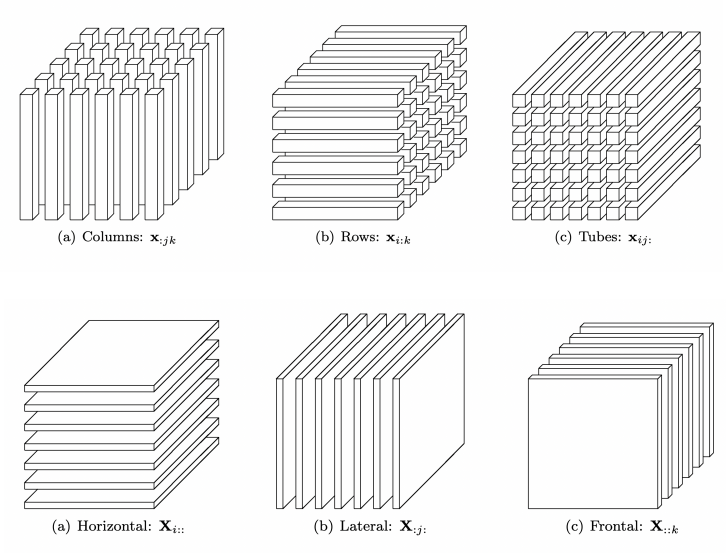

Figure 4: Fibers (top) and Slices (bottom) of $\mathcal{X}$ (Kolda, 2006)

Matricizations of a 2-dimensional tensor

$$\mathcal{A} = A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} = \mathcal{A}_{(1)}; \qquad \mathcal{A}_{(2)} = A^\top = \begin{pmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{mn} \end{pmatrix}$$

Mode-n multiplication in a 2-dimensional tensor

$$\left(\underset{m\times n}{A} \times_1 \underset{p\times m}{U} \right)_{ij} = \sum_{k=1}^{m} a_{kj} u_{ki} = \langle a_{\cdot j}, u_{\cdot i} \rangle = \langle a_{\cdot j}, u_{\cdot i} \rangle = \left(U^\top A \right)_{ij}$$

$$\left(\underset{m\times n}{A} \times_2 \underset{n\times p}{U} \right)_{ij} = \sum_{k=1}^{n} a_{jk} u_{ki} = \langle a_{i\cdot}, u_{\cdot j} \rangle = (AU)_{ij}$$

Proof of important facts 1 & 2

$$\mathbf{1}.\ [\mathcal{X} \times_m A \times_m B]_{i_{-n,-m}jk} = \sum_{i_n=1}^{I_n} \left[\sum_{i_m=1}^{I_m} x_{i_1 i_2 i_3} a_{j i_m} \right] b_{k i_n} =$$

$$= \sum_{i_m=1}^{I_m} \left[\sum_{i_n=1}^{I_n} x_{i_1 i_2 i_3} b_{k i_n} \right] a_{j i_m} = [\mathcal{X} \times_n B \times_m A]_{i_{-n,-m}jk}$$

$$\mathbf{2}.\ [\mathcal{X} \times_n A \times_n B]_{i_{-n},k} = \sum_{j=1}^{k} \left[\sum_{i_n=1}^{I_n} x_{i_1 i_2 i_3} a_{j i_n} \right] b_{kj} =$$

$$= \sum_{i_n=1}^{I_n} x_{i_1 i_2 i_3} \sum_{j=1}^{k} b_{kj} a_{j i_n} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 i_3} (BA)_{k i_n} = [\mathcal{X} \times_n (BA)]_{i_{-n},k}$$

HOSVD elaborated

Given $n \in \{1,2,3\}$ and $r_n = rank_n(\mathcal{A}) \leq I_n$, construct $U_n \in \mathbb{R}^{I_n \times r_n}$ as a matrix of top-r_n left singular vectors of $\mathcal{A}_{(n)}$ (i.e., eigenvectors of $\mathcal{A}_{(n)}\mathcal{A}_{(n)}{}'$). $\mathcal{A}_{(n)}\mathcal{A}_{(n)}{}'$ is symmetric, therefore $U_n' U_n = \mathbb{I}_{I_n},\ U_n U_n' = \mathbb{I}_{r_n}$.

Asymptotic Properties of $\widehat{\mathcal{A}}_{\text{MLR}}, \widehat{A}_{\text{RRR}}, \widehat{A}_{\text{OLS}}$

$$\text{Define } \left(\widehat{\mathcal{A}}_{\text{OLS}}\right)_{(1)} = \widehat{A}_{\text{OLS}} = \arg\min_{B \in \mathbb{R}^{N \times NP}} \sum_{t=1}^{T} \|y_t - Bx_t\|_2^2$$

$$\text{and } \left(\widehat{\mathcal{A}}_{\text{RRR}}\right)_{(1)} = \widehat{A}_{\text{RRR}} = \arg\min_{B \in \mathbb{R}^{N \times NP}, \text{rank}(B) \leq r_1} \sum_{t=1}^{T} \|y_t - Bx_t\|_2^2$$

Assume true (r_1, r_2, r_3) are known, N, P – fixed **(low-dim. setup)**. Also assume $\mathbb{E}\|\epsilon_t\|_2^4 < \inf$, and that all roots of the matrix polynomial $A(z) = \mathbb{I}_N - A_1 z - \ldots - A_P z^P, z \in \mathbb{C}$ lie outside unit circle. Then for method $\in$ {"MLR","RRR","OLS"}, the following is true:

$$\sqrt{T}\left\{\text{vec}\left(\left(\widehat{\mathcal{A}}_{\text{method}}\right)_{(1)}\right) - \text{vec}\left(\mathcal{A}_{(1)}\right)\right\} \xrightarrow[T\to\infty]{D} N\left(0, \Sigma_{\text{method}}\right)$$

Where Σ_{MLR} is a function of $\mathcal{G}, U_1, U_2, U_3$, $\Sigma_{\text{OLS}}, \Sigma_{\text{RRR}}$ - of $A_1, \ldots, A_P$.
Moreover, $\Sigma_{\text{MLR}} \preceq \Sigma_{\text{RRR}} \preceq \Sigma_{\text{OLS}}$

ALS update equations

$$U_1^{(k+1)} \leftarrow \arg\min_{U_1} \sum_{t=1}^{T} \|y_t - \left(\left(x_t' \left(U_3^{(k)} \otimes U_2^{(k)}\right) \mathcal{G}_{(1)}^{(k)\prime}\right) \otimes I_N\right) \text{vec}\left(U_1\right)\|_2^2$$

$$U_2^{(k+1)} \leftarrow \arg\min_{U_2} \sum_{t=1}^{T} \|y_t - U_1^{(k+1)} \mathcal{G}_{(1)}^{(k)} \left(\left(X_t U_3^{(k)}\right)' \otimes I_{r_2}\right) \text{vec}\left(U_2'\right)\|_2^2$$

$$U_3^{(k+1)} \leftarrow \arg\min_{U_3} \sum_{t=1}^{T} \|y_t - U_1^{(k+1)} \mathcal{G}_{(1)}^{(k)} \left(I_{r_3} \otimes \left(U_2^{(k+1)\prime} X_t\right)\right) \text{vec}\left(U_3\right)\|_2^2$$

$$\mathcal{G}^{(k+1)} \leftarrow \arg\min_{\mathcal{G}} \sum_{t=1}^{T} \|y_t - \left(\left(\left(U_3^{(k+1)} \otimes U_2^{(k+1)}\right)' x_t\right)' \otimes U_1^{(k+1)}\right) \text{vec}\left(\mathcal{G}_{(1)}\right)\|_2^2$$

Nuclear Norm (NN) estimator

$\widehat{\mathcal{A}}_{\text{NN}} = \arg\min \frac{1}{T} \sum_{t=1}^{T} \left\|y_t - \mathcal{A}_{(1)} x_t\right\|_2^2 + \lambda \left\|\mathcal{A}_{(1)}\right\|_*$, $\left\|\mathcal{A}_{(1)}\right\|_*$ – nuclear norm, or sum of all singular values of $\mathcal{A}_{(1)}$

Rank selection

Denote $\zeta_i = \frac{1}{\sigma_{r_i}\left(\mathcal{A}_{(i)}\right)} \cdot \max_{1 \leq j < r_i} \frac{\sigma_j\left(\mathcal{A}_{(i)}\right)}{\sigma_{j+1}\left(\mathcal{A}_{(i)}\right)}$.
$c > 0$ is chosen such that: **(1)** $\|\widehat{\mathcal{A}} - \mathcal{A}\|_{\text{F}} = o_p(c)$ and **(2)** $\max_{1 \leq i \leq 3} \zeta_i = o(1/c)$

Other simulations

OLS vs. RRR vs. MLR.

N, P, U_i – same as in rank selection consistency example. Number of replications – same $r_1 = r_2 = 3$, and $r_3 \in \{2, 3, 4\}$. $\mathcal{G}$ is generated by scaling a random iid Gaussian tensor s.t. $\min_{1 \leq i \leq 3} \sigma_{r_i}\left(\mathcal{G}_{(i)}\right) = 1$

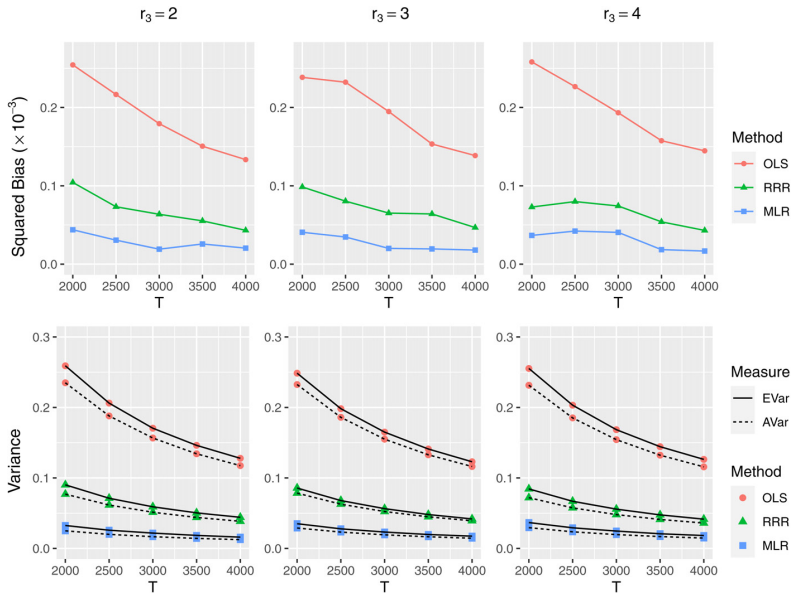

Figure 5: Squared bias, empirical variance (EVar) and asymptotic variance (AVar) for $\widehat{\mathcal{A}}_{\mathrm{OLS}}\widehat{\mathcal{A}}_{\mathrm{RRR}}$, and $\widehat{\mathcal{A}}_{\mathrm{MLR}}$ under various multilinear ranks.

Comparison with existing methods

$(N, P) = (10, 5)$ (case a), $(15, 8)$ (case b); $(r_1, r_2, r_3) = (3, 3, 3), (s_1, s_2, s_3) = (3, 3, 2)$
For case a, $\mathcal{G}$ and U_i 's are generated by the same methods as in RRR vs. MLR vs. OLS. For case b, zeros rows are added below the U_i 's in case a. In both cases, $\|\mathcal{A}\|_0 = 500$. Hence, $\mathcal{A}$ is not sparse in case a, but is sparse in case b.

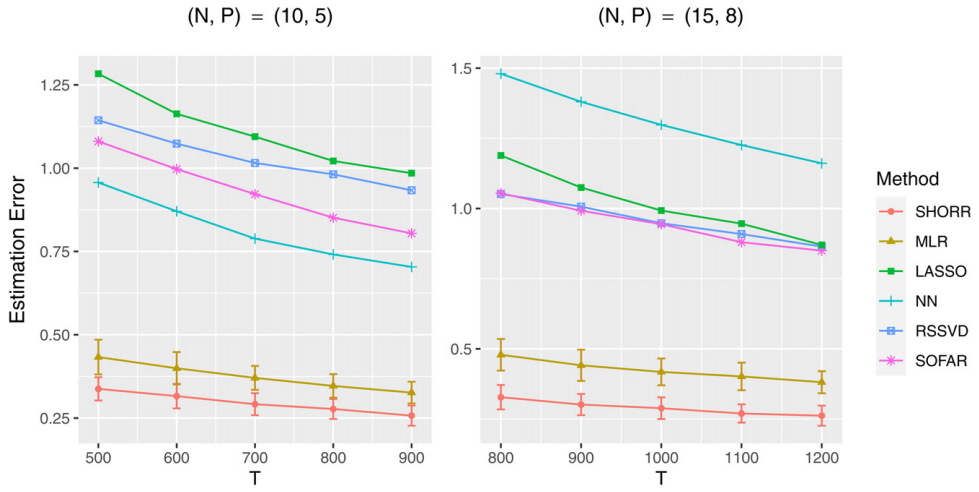

Figure 6: Plots of the estimation error $\|\widehat{\mathcal{A}} - \mathcal{A}\|_{\mathrm{F}}$ against T for six estimation methods in two cases. NN performes the worst in the sparse case, just like it performed the worst on real data.