Mini-Project (ML for Time Series) - MVA 2025/2026

Antoine Le Maguet antoine.lemaguet@free.fr
Alexandre Mallez alexandre.mallez@gmail.com

December 19, 2025

1 Introduction and contributions

Context The analysis of high-dimensional multivariate time series has become ubiquitous in many fiels (finance, health). In this context, the classical Vector Autoregressive (VAR) model constitutes a fundamental tool for capturing dynamic interdependencies among variables. However, the application of this model quickly encounters dimensionality issues, as the number of parameters to estimate grows quadratically with the number of time series (N) and linearly with the autoregressive order (P). Consequently, estimating approximately N^2P parameters becomes computationally prohibitive and leads to a significant loss of estimation efficiency when dimensions are high relative to the sample size.

The scientific context of our project is based on the approach proposed by Wang et al. (2020) [5], which suggests rearranging the transition matrices of the VAR model into a third-order tensor. This structure allows for the application of a Tucker decomposition to restrict the parameter space along three modes simultaneously: response variables, predictor variables, and time lags. Our objective is to implement and analyze the two estimators derived from this method: the Multilinear Low-Rank (MLR) least squares estimator, and the Sparse Higher-Order Reduced-Rank (SHORR) estimator, which imposes additional sparsity for high-dimensional settings.

Contributions and Work Distribution

- **Work Distribution:** Antoine Le Maguet focused on the mathematical derivation, the implementation of traditional base estimators (OLS, RRR), rank selection mechanisms, and statistical tests. Alexandre Mallez took charge of implementing the MLR estimator, the ADMM algorithm for the SHORR estimator, and the analysis of real-world economic data.
- **Source Code:** We utilized a 0% reuse rate of the original source code. We implemented the algorithms completely from scratch based on the mathematical descriptions. We just utilized numpy for implementing tensor operation and tensorly for Tucker decomposition.
- **Experiments:** Regarding existing experiments, we reproduced the authors' validation tests, including asymptotic variance verification, rank consistency checks, and bias analysis using Monte Carlo experiments on self-generated stationary processes. Regarding new experiments, we applied the models to a macro-economic dataset to validate the robustness of the implementation on data.
- **Improvements:** Our primary improvement is the development of a fully modular Python implementation. Furthermore, we explicitly coded the computation of asymptotic variance equations to verify theoretical bounds numerically.

2 Method

Our objective is to solve this problem by finding an estimate of matrices $A_1, \ldots, A_P$ of size $N \times N$, where ϵ_t is an *i.i.d.* centered noise:

$$Y_t = \sum_{k=1}^{p} A_k Y_{t-k} + \epsilon_t \tag{1}$$

2.1 Notations

We follow the authors' notation for vector, matrix and tensor (small, capital and round letter). We keep the classical notation for vector and matrix norms. We define the l_0 and **Frobenius norm** for tensors as: $\|\mathcal{X}\|_F = \left(\sum_{i=1}^{p_1}\sum_{j=1}^{p_2}\sum_{k=1}^{p_3} x_{ijk}^2\right)^{1/2}$ and $\|\mathcal{X}\|_0 = \sum_{i=1}^{p_1}\sum_{j=1}^{p_2}\sum_{k=1}^{p_3} \mathbb{1}(x_{ijk} \neq 0)$. We can matricize the tensor along its modes (1), (2), and (3). The mode-1 matricization is denoted $\mathcal{X}_{(1)} \in \mathbb{R}^{p_1 \times (p_2 p_3)}$ where the index $(i, (k-1)p_2 + j)$ corresponds to x_{ijk}. We can also define multiplication with a matrix in three ways. The first is: $(\mathcal{X} \times_1 Y)_{sjk} = \sum_{i=1}^{p_1} \mathcal{X}_{ijk} Y_{si}, \quad \forall\, 1 \leq s \leq q_1,\ 1 \leq j \leq p_2,\ 1 \leq k \leq p_3$. The three ranks of a tensor correspond to the ranks of the matricizations in the different modes.

Proposition 1. *For a tensor of rank (r_1, r_2, r_3), we can decompose it into three matrices and a core tensor $\mathcal{Y} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$:*

$$\mathcal{X} = \mathcal{Y} \times_1 Y_1 \times_2 Y_2 \times_3 Y_3 \quad \textit{(Tucker Decomposition)} \tag{2}$$

We denote it $\mathcal{X} = [\![\mathcal{Y}; Y_1, Y_2, Y_3]\!]$. [2] [1]

2.2 The Tensor Approach

Context First, we rewrite the VAR equation using a tensor and the Tucker decomposition. With x_t containing the stacked lagged vectors y, we have:

$$y_t = (\mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3)_{(1)} x_t + \epsilon_t = U_1 \mathcal{G}_{(1)} (U_3 \otimes U_2)' x_t + \epsilon_t = U_1 \mathcal{G}_{(1)} \mathrm{vec}(U_2' X_t U_3) + \epsilon_t \tag{3}$$

This reduces the number of parameters to estimate. The count becomes $r_1 r_2 r_3 + (N - r_1) r_1 + (N - r_2) r_2 + (P - r_3) r_3$, which increases linearly with N and P, in contrast to the quadratic growth of the standard VAR model. We can interpret the matrices U_1, U_2, U_3 as follows:

- U_1 is associated with the **response factor**. It reduces the N signals to r_1 signals that describe the dynamics between the different y_t.
- U_2 is associated with the **predictor factor**. It corresponds to the information contained in the variables x, reducing N to r_2.
- U_3 is the **temporal factor**, reducing the information contained in the lags ($r_3 \ll P$).

Low-Dimensional Estimation Before addressing high-dimensional settings, the authors establish the theoretical properties of the proposed model for fixed dimensions N and P. They introduce the Multilinear Low-Rank (MLR) estimator (see properties below), defined as the solution to the least squares minimization:

$$\hat{\mathcal{A}}_{MLR} = \arg \min_{\mathcal{G}, U_1, U_2, U_3} \sum_{t=1}^{T} \|y_t - (\mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3)_{(1)} x_t\|_2^2 \tag{4}$$

- **Consistency:** The estimator converges to the true solution with probability 1.
- **Asymptotic Normality:** It converges to a Gaussian distribution as $T \to \infty$.
- **Efficiency:** The asymptotic variance of MLR is lower than that of OLS and RRR.

We use the Alternating Least Squares (ALS) algorithm (describe in appendix) to find the final estimator by exploiting the convexity of each block. The algorithm iteratively updates U_1, U_2, U_3, and $\mathcal{G}$ in a cyclic manner. It converges to a stationary point of the objective function.

High-Dimensional Adaptation In high-dimensional settings, merely restricting the rank is often insufficient for interpretability and estimation efficiency. The factor matrices U_i estimated by MLR are typically dense. To address this, the authors propose the **Sparse Higher-Order Reduced-Rank (SHORR)** estimator (algorithm in appendix). The SHORR estimator imposes sparsity on the factor matrices by adding an l_1-regularization penalty to the objective function. It is defined as:

$$\hat{\mathcal{A}}_{SHORR} = \arg \min_{\mathcal{G},U_1,U_2,U_3} (L(\mathcal{G}, U_1, U_2, U_3) + \lambda \|U_3 \otimes U_2 \otimes U_1\|_1) \tag{5}$$

subject to the orthogonality constraints $U_i' U_i = I_{r_i}$ and the structure of the core tensor $\mathcal{G}$. Here, λ is a parameter controlling the level of sparsity on U_1, U_2, U_3, using the least squares error as the loss. The estimation error is bounded by a term proportional to $\sqrt{S \log(N^2 P)/T}$, where S represents the sparsity level (number of non-zero elements). This result proves that the estimator is consistent even when the dimensions N and P grow, provided the true model is sufficiently sparse.

Solving (5) is challenging due to the non-smooth l_1 penalty and the non-convex orthogonality constraints. The authors develop an algorithm based on the **Alternating Direction Method of Multipliers (ADMM** that consists in minimizing the loss functions with orthogonality constraints and next applying the sparsity penalty (handled via soft-thresholding).

Rank Selection Strategy The theoretical properties of MLR and SHORR rely on knowledge of the true multilinear ranks (r_1, r_2, r_3), which are unknown in practice. To address this, the authors propose a consistent rank selection procedure based on a **Ridge-Type Ratio Estimator**. First, a preliminary consistent estimator $\hat{\mathcal{A}}_{init}$ is obtained, typically using nuclear norm regularization [3]. Then, the ranks are estimated by identifying the drop in singular values for each mode $i \in \{1, 2, 3\}$:

$$\hat{r}_i = \arg \min_{1 \le j \le p_i - 1} \frac{\sigma_{j+1}((\hat{\mathcal{A}}_{init})_{(i)}) + c}{\sigma_j((\hat{\mathcal{A}}_{init})_{(i)}) + c} \tag{6}$$

where $\sigma_j(\cdot)$ denotes the j-th singular value and $c > 0$ is a tuning parameter chosen to dominate the estimation error. Under mild assumptions on the gap between singular values, the probability of selecting the correct triplet of ranks converges to 1 as the sample size T tends to infinity. This justifies using these estimated ranks for the subsequent MLR or SHORR estimation steps.

3 Data

3.1 Generation of Stationary Processes

To simulate realistic multivariate time series governed by a transition tensor $\mathcal{A} \in \mathbb{R}^{N \times N \times P}$, we implemented a generation procedure strictly enforcing stationarity, a critical condition to avoid

explosive variance. The transition tensor is constructed via a Tucker decomposition with random orthogonal factor matrices U_i and a fixed low-rank core tensor $\mathcal{G}$. A crucial diagnostic step involves computing the spectral radius ρ of the associated companion matrix; we iteratively scale $\mathcal{G}$ until $\rho < 1$, ensuring the process remains stable. We designed two specific experimental regimes to evaluate the estimators:

- **Low-Rank Structure (MLR):** We fixed dimensions at $N = 10$ and $P = 5$ (500 parameters) and varied $T \in [100, 4000]$ to analyze convergence in high-dimensional settings where OLS typically overfits.
- **Sparse and Orthogonal Structure (SHORR):** To assess the sparsity-inducing estimator, we enforced strictly orthogonal factor matrices and introduced sparsity prior to the spectral scaling. This setup ensures the identifiability of the components and tests the model's ability to recover support in high dimensions.

Finally, to validate the rank selection mechanism, we analyzed the singular value spectrum of the generated tensors. We implemented a new generation process to create a distinct spectral gap, allowing the "Ridge-Type Ratio Estimator" to distinguish signal from noise using a singular value perturbation approach.

3.2 Real Data

The analysis is based on the standard US Macroeconomic dataset (Statsmodels [4]), from which we selected 11 core indicators capturing real economic activity (*realgdp*, *realcons*, *realinv*, *realgovt*, *realdpi*), price dynamics (*cpi*, *infl*), and labor/monetary market conditions (*m1*, *tbilrate*, *realint*, *unemp*). These data should be correlated, as it's what we want for multimodal analysis. There are one data for each quarter of a year (4 data/year).

To ensure the statistical robustness of the vector autoregressive (VAR) framework, the sample period is restricted to 1959-2005. This temporal truncation intentionally excludes the 2008 financial crisis to remove this period of high volatility and structural breaks. The goal is to use our estimator to forecast these time series.

4 Results

4.1 Numerical Simulations on Generated Data

Asymptotic Efficiency of MLR Our Monte Carlo simulations quantitatively confirmed the theoretical efficiency hierarchy $\Sigma_{MLR} \leq \Sigma_{RRR} \leq \Sigma_{OLS}$ 3, validating that tensor restrictions effectively mitigate variance in high-dimensional settings. A key technical contribution was the successful implementation of the theoretical asymptotic variance formula $\Sigma_{MLR} = H(H'JH)^{\dagger}H'$. The convergence of our empirical variance towards this theoretical bound validates the correctness of our complex tensor algebra implementation and confirms the estimator's consistency. 4

SHORR and High-Dimensional Convergence We validated the consistency of the SHORR estimator in high-dimensional regimes, observing a monotonic decrease in estimation error $\|\hat{\mathcal{A}}_{SHORR} - \mathcal{A}\|_F$ with sample size T 5. SHORR outperformed the dense MLR estimator in sparse settings by effectively shrinking insignificant coefficients. While the error decay generally followed the expected theoretical scaling factor, the simulations exhibited minor deviations attributed to the computational limits of the ADMM algorithm. 6

Rank Estimation For rank initialization, we prioritized OLS over the Nuclear Norm estimator due to better empirical stability in our experiments. The ridge-type ratio minimization proved consistent for identifying ranks (r_1, r_2, r_3), particularly in balanced scenarios where $r_i > 1$. However, the method showed sensitivity to the data generation process, requiring a sufficient spectral gap to robustly distinguish the signal subspace from noise. 8

4.2 Real Data

Our experiment were in three steps. We have first estimate the tensor modelizing the model. After that, we have reconstructed the signal with this estimated tensor to analyze if estimation was sufficient and 10. Finally, we have predict 11 the end of the signals. For this, we have take a part of the data for the estimation and the other part for the validation.

The experimental results obtained on the autoregressive task are generally satisfactory, demonstrating the model's capacity to capture underlying economic dynamics. However, it is important to note that despite distinct preprocessing steps the dataset exhibits residual non-stationarity. This persistent instability affects the model's convergence properties over extended periods.

Regarding the comparative analysis of estimators, we observe a performance trade-off based on the forecasting horizon. For short-term predictions (one-step ahead), the MLR yields slightly superior accuracy, effectively capturing immediate local variations. Conversely, for longer forecasting horizons, the SHORR estimator proves to be the most robust method. This superiority suggests that the regularization constraints inherent to SHORR provide better generalization and stability when propagating predictions over time in a high-dimensional, noisy environment.

4.3 Limitations

SHORR Implementation Implementing the SHORR estimator presented significant challenges due to the non-convexity of the optimization. We observed that the ADMM algorithm was highly sensitive to the scale of the data fidelity term; normalizing the objective function by $1/T$ was a crucial technical improvement to ensure convergence. However, the iterative nature of ADMM remains computationally expensive compared to ALS. This high cost limited the number of Monte Carlo replications we could perform, resulting in higher residual variance in our non-asymptotic validation plots 5. The choice of regularizations terms (they are 4 differents one in the ADMM) is also challenging and couldn't be done using grid-search due to not enough computational ressources.

Rank Estimation We found that the consistency of the "Ridge-Type Ratio" estimator was linked to the spectral properties of the data. The method relies on a perceptible spectral gap to distinguish signal from noise. Consequently, we had to iteratively refine our data generation process to ensure that the core tensor $\mathcal{G}$ possessed sufficiently large singular values.

Future Directions Our analysis has two main limitations. First, due to computational costs, we did not implement the BIC-based grid search for hyperparameter tuning, nor did we compare against the most recent state-of-the-art estimators mentioned in the literature. Second, the application to macroeconomic data highlighted the limits of assuming a constant transition tensor $\mathcal{A}$. A promising extension would be the **Time-Varying Coefficient VAR**, where transition matrices evolve dynamically. As suggested by the authors, this could be modeled via a fourth-order tensor decomposition to capture structural changes without parameter explosion.

References

[1] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[2] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[3] Sahand Negahban and Martin J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, 39(2):1069–1097, 2011.

[4] Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

[5] Di Wang, Yao Zheng, Heng Lian, and Guodong Li. High-dimensional vector autoregressive time series modeling via tensor decomposition. *Journal of the American Statistical Association*, 2020.

A Plot

A.1 MLR

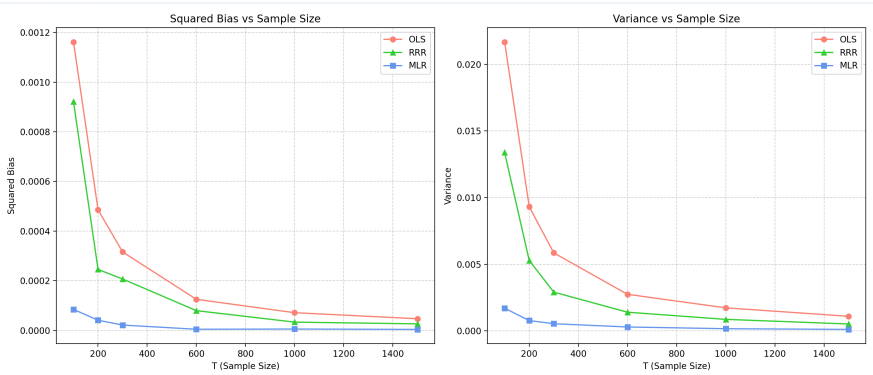


Figure 1: Analyze the squared bias and the variance of MLS estimators against the sample size

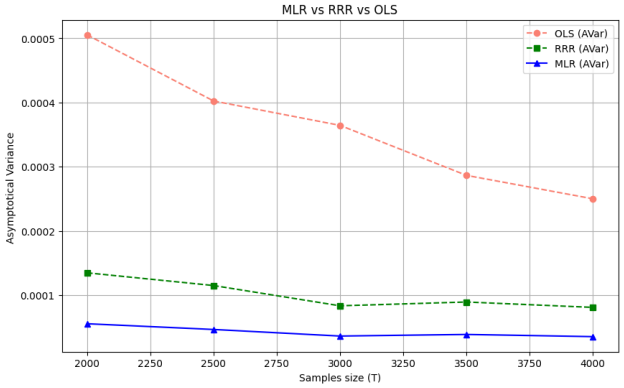


Figure 2: Asymptotical Variance for OLS, MLR and RRR (consistency and Σ_{MLR} is smaller than the other

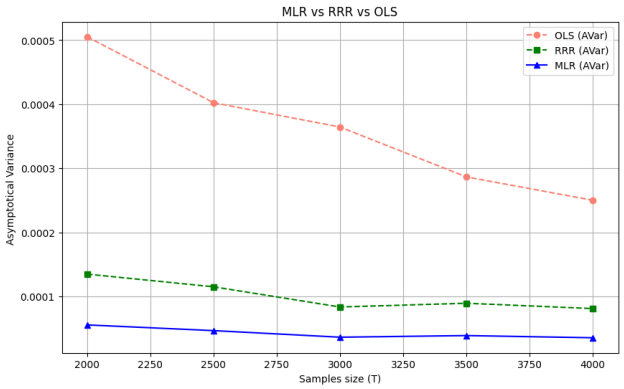


Figure 3: Asymptotical Variance for OLS, MLR and RRR (consistency and Σ_{MLR} is smaller than the other

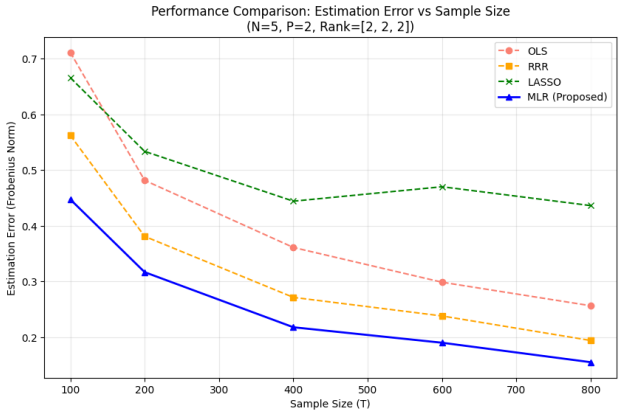


Figure 4: Convergence of the error for different estimators

A.2 SHORR

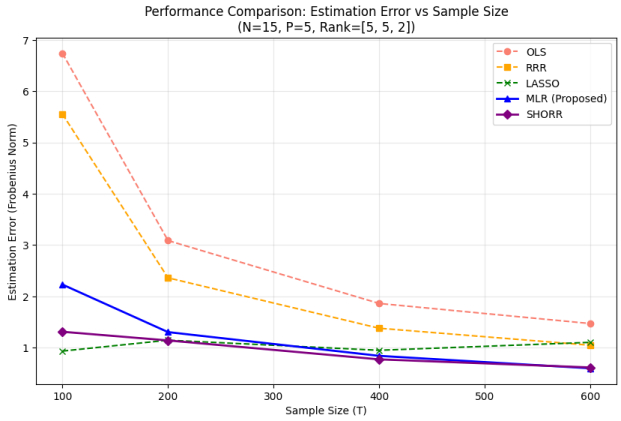


Figure 5: Convergence for the SHORR estimator

The next figure must be affine to show this consistency, it shows a bit of unstability in our estimators.

Figure 6: estimation of the error depending on $\gamma = S\log(N^2P)/T$, S defines the sparsity level.

A.3 rank

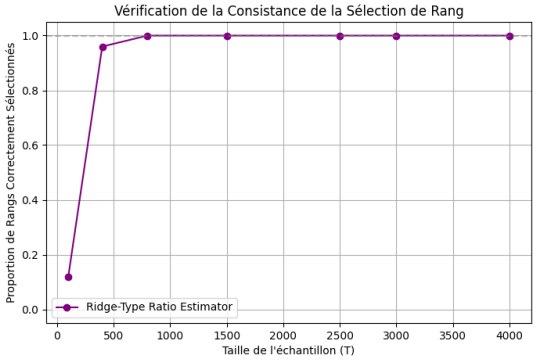


Figure 7: Rank Consistency; The probability of estimating the good rank converge to 1

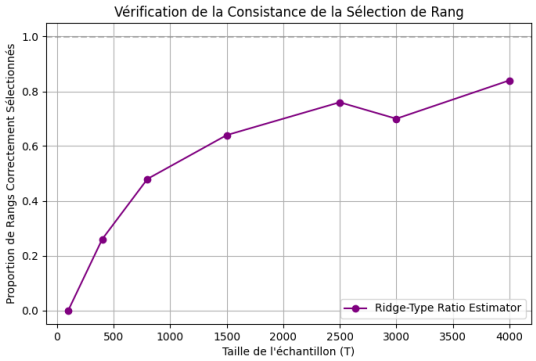


Figure 8: Rank Consistency; this plot shows the problem relating to process which have not significant singular values or if our tensor cotains gap between it : Denoising step is important

A.4 Real Data

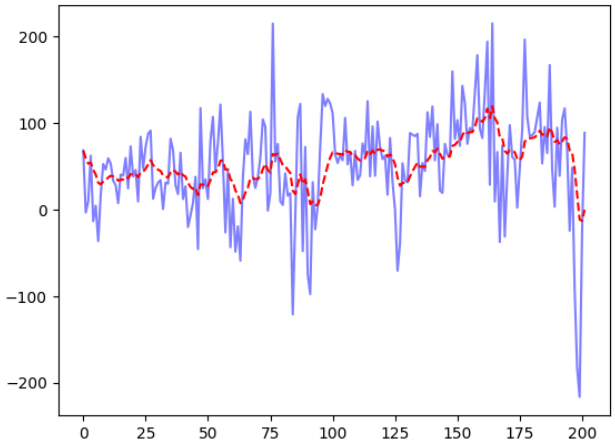


Figure 9: Making data stationary (cancel trend and velocity)

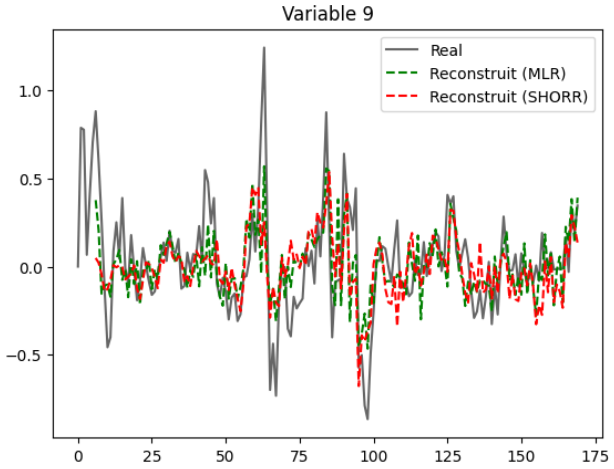
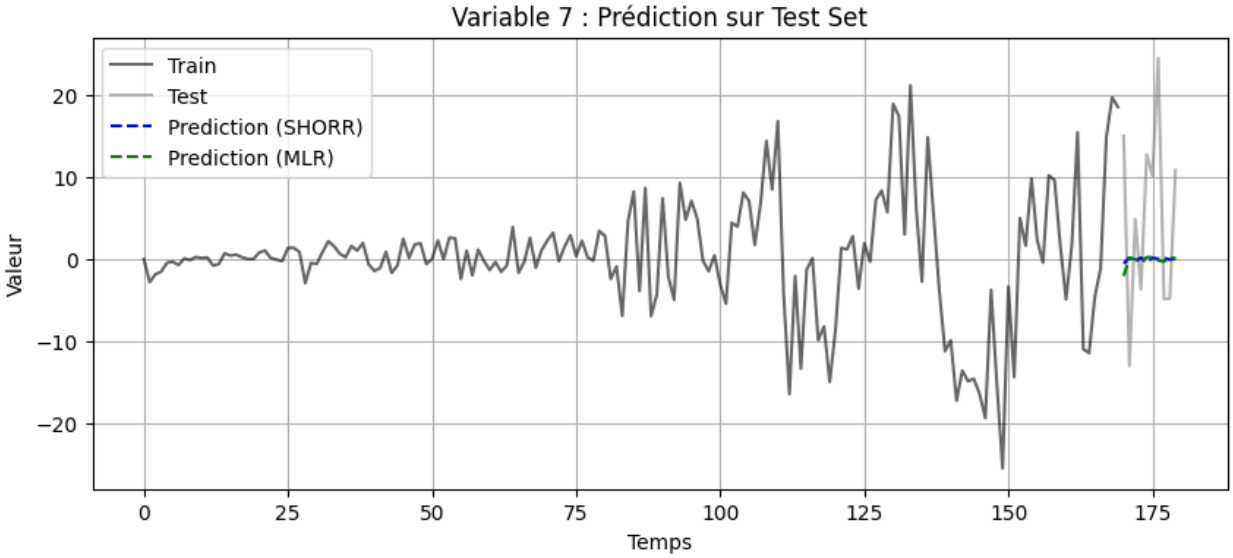


Figure 10: Reconstruction of the signal after estimation of tensors

[b]0.45



[b]0.45

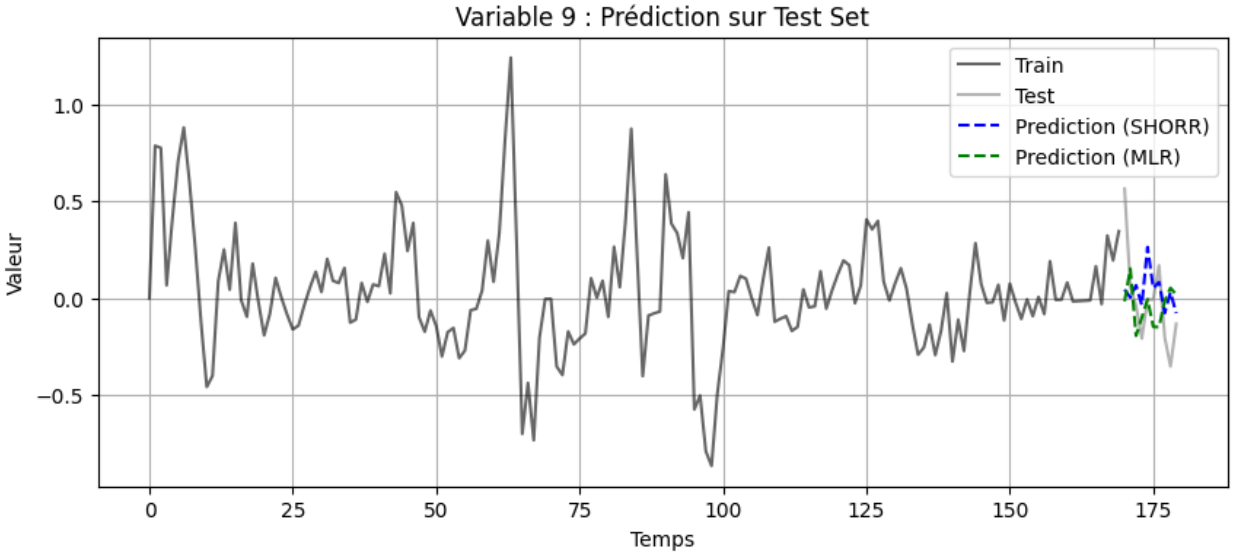


Figure 11: Comparison of signal prediction between estimators (problem when time series is far from an AR model)

B Algorithms

B.1 ALS algorithm

Algorithm 1 Alternating Least Squares (ALS) for MLR Estimator

Initialization: Choose an initial value $\mathcal{A}^{(0)}$ (e.g., via OLS or RRR).
Compute initial HOSVD: $\mathcal{A}^{(0)} \approx \mathcal{G}^{(0)} \times_1 U_1^{(0)} \times_2 U_2^{(0)} \times_3 U_3^{(0)}$
repeat
  Update U_1:
  $U_1^{(k+1)} \leftarrow \arg\min_{U_1} \sum_{t=1}^{T} \left\| y_t - \left((x_t'(U_3^{(k)} \otimes U_2^{(k)}) \mathcal{G}_{(1)}^{(k)'}) \otimes I_N \right) \mathrm{vec}(U_1) \right\|_2^2$
  Update U_2:
  $U_2^{(k+1)} \leftarrow \arg\min_{U_2} \sum_{t=1}^{T} \left\| y_t - U_1^{(k+1)} \mathcal{G}_{(1)}^{(k)} ((X_t U_3^{(k)})' \otimes I_{r_2}) \mathrm{vec}(U_2') \right\|_2^2$
  Update U_3:
  $U_3^{(k+1)} \leftarrow \arg\min_{U_3} \sum_{t=1}^{T} \left\| y_t - U_1^{(k+1)} \mathcal{G}_{(1)}^{(k)} (I_{r_3} \otimes (U_2^{(k+1)'} X_t)) \mathrm{vec}(U_3) \right\|_2^2$
  Update $\mathcal{G}$:
  $\mathcal{G}^{(k+1)} \leftarrow \arg\min_{\mathcal{G}} \sum_{t=1}^{T} \left\| y_t - \left(((U_3^{(k+1)} \otimes U_2^{(k+1)})' x_t)' \otimes U_1^{(k+1)} \right) \mathrm{vec}(\mathcal{G}_{(1)}) \right\|_2^2$
  Reconstruction:
  $\mathcal{A}^{(k+1)} \leftarrow \mathcal{G}^{(k+1)} \times_1 U_1^{(k+1)} \times_2 U_2^{(k+1)} \times_3 U_3^{(k+1)}$
until convergence (stopping criterion met)
Finalization (for uniqueness):
$\hat{U}_i \leftarrow$ top r_i left singular vectors of $\hat{\mathcal{A}}_{(i)}$ (with positive first element), for $i = 1, 2, 3$.
$\hat{\mathcal{G}} \leftarrow \hat{\mathcal{A}} \times_1 \hat{U}_1' \times_2 \hat{U}_2' \times_3 \hat{U}_3'$
return $\hat{\mathcal{A}}_{MLR} = [\![\hat{\mathcal{G}}; \hat{U}_1, \hat{U}_2, \hat{U}_3]\!]$

B.2 ADMM algorithm

Algorithm 2 ADMM Algorithm for SHORR Estimator

Initialization: Choose $\mathcal{A}^{(0)}$ (e.g., via Nuclear Norm estimator).
Initial HOSVD: $\mathcal{A}^{(0)} \approx \mathcal{G}^{(0)} \times_1 U_1^{(0)} \times_2 U_2^{(0)} \times_3 U_3^{(0)}$ with ranks (r_1, r_2, r_3).
repeat
▷ *Update Factors (Sparse & Orthogonal Subproblems)*
$U_1^{(k+1)} \leftarrow \arg\min_{U_1'U_1=I}\{L(\mathcal{G}^{(k)}, U_1, U_2^{(k)}, U_3^{(k)}) + \lambda\|U_1\|_1\|U_2^{(k)}\|_1\|U_3^{(k)}\|_1\}$
$U_2^{(k+1)} \leftarrow \arg\min_{U_2'U_2=I}\{L(\mathcal{G}^{(k)}, U_1^{(k+1)}, U_2, U_3^{(k)}) + \lambda\|U_1^{(k+1)}\|_1\|U_2\|_1\|U_3^{(k)}\|_1\}$
$U_3^{(k+1)} \leftarrow \arg\min_{U_3'U_3=I}\{L(\mathcal{G}^{(k)}, U_1^{(k+1)}, U_2^{(k+1)}, U_3) + \lambda\|U_1^{(k+1)}\|_1\|U_2^{(k+1)}\|_1\|U_3\|_1\}$
▷ *Update Core $\mathcal{G}$ (All-Orthogonal constraint handling)*
$\mathcal{G}^{(k+1)} \leftarrow \arg\min_{\mathcal{G}}\{L(\mathcal{G}, U_1^{(k+1)}, U_2^{(k+1)}, U_3^{(k+1)}) + \sum_{i=1}^{3}\rho_i\|\mathcal{G}_{(i)} - D_i^{(k)}V_i^{(k)'} + (\mathcal{C}_i^{(k)})_{(i)}\|_F^2\}$
▷ *Update auxiliary and dual variables*
for $i = 1, 2, 3$ **do**
$D_i^{(k+1)} \leftarrow \arg\min_{D_i=\text{diag}(d_i)} \|\mathcal{G}_{(i)}^{(k+1)} - D_iV_i^{(k)'} + (\mathcal{C}_i^{(k)})_{(i)}\|_F^2$
$V_i^{(k+1)} \leftarrow \arg\min_{V_i'V_i=I} \|\mathcal{G}_{(i)}^{(k+1)} - D_i^{(k+1)}V_i' + (\mathcal{C}_i^{(k)})_{(i)}\|_F^2$
$(\mathcal{C}_i^{(k+1)})_{(i)} \leftarrow (\mathcal{C}_i^{(k)})_{(i)} + \mathcal{G}_{(i)}^{(k+1)} - D_i^{(k+1)}V_i^{(k+1)'}$
end for
Reconstruction:
$\mathcal{A}^{(k+1)} \leftarrow \mathcal{G}^{(k+1)} \times_1 U_1^{(k+1)} \times_2 U_2^{(k+1)} \times_3 U_3^{(k+1)}$
until convergence

For this SOC problem:

$$\mathbf{B}^{(k+1)} \leftarrow \arg\min_{\mathbf{B}^T\mathbf{B}=I}\left\{\frac{1}{n}\|\mathbf{y} - \mathbf{X}\text{vec}(\mathbf{B})\|_2^2 + \kappa\|\mathbf{B} - \mathbf{W}^{(k)} + \mathbf{M}^{(k)}\|_F^2\right\}$$

Posing $\mathbf{C} = \mathbf{W}^{(k)} - \mathbf{M}^{(k)}$

Without constraint :

$$\min_{\mathbf{v}} f(\mathbf{v}) = \frac{1}{n}\|\mathbf{y} - \mathbf{X}\mathbf{v}\|_2^2 + \kappa\|\mathbf{v} - \text{vec}(\mathbf{C})\|_2^2$$

with $\mathbf{v} = \text{vec}(\mathbf{G})$ we need to minimize

Using $\|\mathbf{B} - \mathbf{C}\|_F^2 = \|\text{vec}(\mathbf{B}) - \text{vec}(\mathbf{C})\|_2^2$

We have the close form solution :

$$\text{vec}(\mathbf{G}) = \left(\frac{1}{n}\mathbf{X}^T\mathbf{X} + \kappa\mathbf{I}\right)^{-1}\left(\frac{1}{n}\mathbf{X}^T\mathbf{y} + \kappa\mathbf{C}_{\text{vec}}\right)$$

To find back the solution we need to project on Stiefel Variety

$$\mathbf{G} = \mathbf{U}_{\text{svd}}\mathbf{\Sigma}\mathbf{V}_{\text{svd}}^T$$

Then:

$$\mathbf{B}^{(k+1)} = \mathbf{U}_{\text{svd}}\mathbf{V}_{\text{svd}}^T$$

C Benchmark

This is the different estimators we implemented.

Sparsity We can assume that the A_k matrices are sparse, meaning they contain a large number of zero entries. This involves introducing a regularization term $\lambda\Omega(Z)$ in the Least Squares equation (e.g., l_1-norm). While widely used, this method faces specific challenges in time series analysis. First, it ignores the temporal and cross-sectional dependencies inherent in the data, which makes estimation unstable. Finally, standard Lasso, for example, does not exploit the low-rank structures in transition matrices.

Reduced-Rank Regression (RRR) As mentioned, it is interesting to identify the low rank that can restrict the parameter structure. The rank of the concatenated matrix r is typically much smaller than N and NP. This implies that the dynamic relationship between the predictors and the responses is driven by a smaller number of latent factors. The reduced-rank estimator minimizes the weighted least squares objective function subject to this rank constraint. While RRR effectively reduces the number of parameters, it restricts the parameter space in only one direction: the column space. It fails to exploit potential low-rank structures along other dimensions, specifically the row space (variables) and the temporal space (time lags).

Nuclear norm (NN) To overcome the computational challenges associated with non-convex rank constraints, the Nuclear Norm estimator is frequently employed as a convex relaxation of the rank minimization problem. This method estimates the parameters by minimizing the least squares objective function regularized by the nuclear norm of the transition matrix A_1 (the sum of its singular values). By doing so, it effectively encourages a low-rank solution, making it suitable for high-dimensional scaling where the true underlying dynamic structure is governed by a few factors. However, similar to RRR, the standard nuclear norm approach typically operates on the unfolded matrix A_1, which restricts the parameter space along only one specific matricization direction. Consequently, it may fail to fully capture the simultaneous low-rank structures across the row and temporal dimensions of the transition tensor, and unlike sparse estimators, it does not exploit entry-wise sparsity, which can result in lower efficiency if the true model is sparse.