Mini-Project (ML for Time Series) - "High-dimensional vector autoregressive time series modeling via tensor decomposition" by Di Wang, Yao Zheng, Heng Lian and Guodong Li

Dorian Gailhard dorian.gailhard@telecom-paris.fr
Denis Duval denis.duval13@gmail.com

June 27, 2023

1 Introduction and contributions

The article we chose is "High-dimensional vector autoregressive time series modeling via tensor decomposition" [link] by Di Wang, Yao Zheng, Heng Lian and Guodong Li. It focuses on autoregressive models and introduces a novel formalization of them, combining the matrices inside a tensor, which can then benefit from tensor decomposition to restrict the parameter space into three directions. It also adapts regularization methods to this new formalism.

The work carried out in the paper is quite complex and exhaustive. We chose to only focus on certain parts of their work. More precisely we focused on the first part of the article which, in the end, proposes a new estimator for VAR models and a corresponding algorithm for computing it. We hence do not focused on the comparison with factor models nor we had dived into the theoretical guarantees of the proposed methods.

Our work consisted to implement from scratch the alternating least squares algorithm that computes the proposed MLR estimator (Part 4 of the paper). We then wanted to test this estimator on real data for forecasting. Unfortunately, we couldn't reach this task and we only studied the fitting of the VAR model to our data. We also led some preliminary experiments to test an initial estimator, and various estimators for the multilinear ranks.

Also:

- The work has been divided as follows : Denis did most of the implementation and Dorian factored the code and corrected some bugs. Both of us wrote the report.
- No previous implementation has been reused, everything was done by ourselves from scratch.
- We tested the implementations on artificial generated data in the same conditions as the authors did. All the experiments were not reproduced and those that were were done with lighter parameters due to constraints on computational power.

2 Method

2.1 Rewriting an AR process with tensors

The classical way of formalizing an AR process problem is the following :

$$y_t = A^1 y_{t-1} + \cdots + A^P y_{t-P} + \epsilon_t, 1 \leq t \leq T$$

where y_t being the observed time series (with $y_t \in \mathbb{R}^N$, ϵ_t being i.i.d innovations of mean 0 and finite variance, also in $\mathbb{R}^N$, and $\{A^i\}_{1 \leq i \leq P}$ being $N \times N$ transitions matrices, which are unknown).

The previous problem can be expressed with a tensor combining the transitions matrices, grouping them along a third dimension such that :

$$\forall (i,j,k) \in [1,N] \times [1,N] \times [1,P], \mathcal{A}_{ijk} = A^k_{ij}$$

Denoting (r_1, r_2, r_3) the multilinear ranks of this tensor, there exists the following Tucker decomposition :

$$\mathcal{A} = \mathcal{G} \times_1 \mathcal{U}_1 \times_2 \mathcal{U}_2 \times_3 \mathcal{U}_3$$

where $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ is the core tensor, and $\mathcal{U}_1 \in \mathbb{R}^{N \times r_1}$, $\mathcal{U}_2 \in \mathbb{R}^{N \times r_2}$ and $\mathcal{U}_3 \in \mathbb{R}^{P \times r_3}$ are factor matrices.

Ultimately, the AR model can be rewritten as :

$$y_t = (\mathcal{G} \times_1 \mathcal{U}_1 \times_2 \mathcal{U}_2 \times_3 \mathcal{U}_3)_{(1)} x_t + \epsilon_t$$

where $\mathcal{A}_{(1)}$ is the mode-1 matricization of $\mathcal{A}$ (please refer to section 2.1 of the article for additional details).

2.2 The first algorithm

To estimate the unknown tensors, the authors introduce a multilinear low-rank least squares estimation algorithm which targets the minimization of the following loss :

$$L(\mathcal{G}, \mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3) = \frac{1}{T} \sum_{t=1}^{T} \| y_t - (\mathcal{G} \times_1 \mathcal{U}_1 \times_2 \mathcal{U}_2 \times_3 \mathcal{U}_3)_{(1)} x_t \|_2^2$$

The algorithm is the following :

This allows to capture the dynamic information along three dimensions, with the three $\mathcal{U}_i$ representing the factor loadings.

2.3 The second algorithm

When the problem becomes highly dimensional, a lot of the values of these three matrices become small. It is thus possible to enforce sparsity to reduce the number of unknown parameters and increase interpretability. The authors adapt the classical regularization methods with an algorithm targeting the following loss :

$$L_{sparse}(\mathcal{G}, \mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3) = L(\mathcal{G}, \mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3) + \lambda \| \mathcal{U}_1 \otimes \mathcal{U}_2 \otimes \mathcal{U}_3 \|_1 = L(\mathcal{G}, \mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3) + \lambda \| \mathcal{U}_1 \|_1 \| \mathcal{U}_2 \|_1 \| \mathcal{U}_3 \|_1$$

Algorithm 1

Initialize: $\mathcal{A}^{(0)}$
Tensor decomposition : $\mathcal{A}^{(0)} = \mathcal{G}^{(0)} \times_1 \mathcal{U}_1^{(0)} \times_2 \mathcal{U}_2^{(0)} \times_3 \mathcal{U}_3^{(0)}$ with multilinear ranks (r_1, r_2, r_3)
Repeat until convergence :
$\mathcal{U}_1^{(k+1)} \leftarrow argmin_{\mathcal{U}_1} \sum_{t=1}^{T} \|y_t - ((x_t^T(\mathcal{U}_3^{(k)} \otimes \mathcal{U}_2^{(k)})\mathcal{G}_{(1)}^{(k)T}) \otimes I_N)vec(\mathcal{U}_1)\|_2^2$
$\mathcal{U}_2^{(k+1)} \leftarrow argmin_{\mathcal{U}_2} \sum_{t=1}^{T} \|y_t - \mathcal{U}_1^{(k+1)}\mathcal{G}_{(1)}^{(k)}((X_t\mathcal{U}_3^{(k)})^T \otimes I_{r_2})vec(\mathcal{U}_\in^T)\|_2^2$
$\mathcal{U}_3^{(k+1)} \leftarrow argmin_{\mathcal{U}_3} \sum_{t=1}^{T} \|y_t - \mathcal{U}_1^{(k+1)}\mathcal{G}_{(1)}^{(k)}(I_{r_3} \otimes (\mathcal{U}_2^{(k+1)T}X_t))vec(\mathcal{U}_3)\|_2^2$
$\mathcal{G}^{(k+1)} \leftarrow argmin_{\mathcal{G}} \sum_{t=1}^{T} \|y_t - (((\mathcal{U}_3^{(k+1)} \otimes \mathcal{U}_2^{(k+1)})^T x_t)^T \otimes \mathcal{U}_1^{(k+1)})vec(\mathcal{G}_{(1)})\|_2^2$
$\mathcal{A}^{(k+1)} \leftarrow \mathcal{G}^{(k+1)} \times_1 \mathcal{U}_1^{(k+1)} \times_2 \mathcal{U}_2^{(k+1)} \times_3 \mathcal{U}_3^{(k+1)}$
Finalize :
$\mathcal{U}_i \leftarrow$ top r_i left singular vectors of $\mathcal{A}_{(i)}$ with positive first elements, $1 \leq i \leq 3$
$\mathcal{G} \leftarrow \mathcal{A} \times_1 \mathcal{U}_1 \times_2 \mathcal{U}_2 \times_3 \mathcal{U}_3$

with the constraints $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$, $\mathcal{G}_{(i)}$ is rwo orthogonal for each i, and $\mathcal{U}_i^T\mathcal{U}_i = I_{r_i}$.

The algorithm is the following (the argmins with orthogonality constraints are split into a subroutine, please refer to the paper for all the details at section 4.3, and the l_i are regularization parameters) :

Algorithm 2

Initialize: $\mathcal{A}^{(0)}$
Tensor decomposition : $\mathcal{A}^{(0)} = \mathcal{G}^{(0)} \times_1 \mathcal{U}_1^{(0)} \times_2 \mathcal{U}_2^{(0)} \times_3 \mathcal{U}_3^{(0)}$ with multilinear ranks (r_1, r_2, r_3)
repeat until convergence :
$\mathcal{U}_1^{k+1} \leftarrow argmin_{\mathcal{U}_1^T\mathcal{U}_1 = I_{r_1}} L_{sparse}(\mathcal{G}^{(k)}, \mathcal{U}_1, \mathcal{U}_2^{(k)}, \mathcal{U}_3^{(k)})$
$\mathcal{U}_2^{k+1} \leftarrow argmin_{\mathcal{U}_2^T\mathcal{U}_1 = I_{r_2}} L_{sparse}(\mathcal{G}^{(k)}, \mathcal{U}_1^{(k+1)}, \mathcal{U}_2, \mathcal{U}_3^{(k)})$
$\mathcal{U}_3^{k+1} \leftarrow argmin_{\mathcal{U}_3^T\mathcal{U}_3 = I_{r_3}} L_{sparse}(\mathcal{G}^{(k)}, \mathcal{U}_1^{(k+1)}, \mathcal{U}_2^{(k+1)}, \mathcal{U}_3)$
$\mathcal{G}^{k+1} \leftarrow argmin_{\mathcal{G}} L(\mathcal{G}, \mathcal{U}_1^{(k+1)}, \mathcal{U}_2^{(k+1)}, \mathcal{U}_3^{(k+1)}) + \sum_{i=1}^{3} l_i \|\mathcal{G}_{(i)} - D_i^{(k)}V_i^{(k)T} + (\mathcal{C}_i^{(k)})_{(i)}\|_2^2$
for $i \in \{1, 2, 3\}$ **do** :
$D_i^{(k+1)} \leftarrow argmin_{D_i = diag(d_i)} \|\mathcal{G}_{(i)}^{(k+1)} - D_iV_i^{(k)T} + (\mathcal{C}_i^{(k)})_{(i)}\|_2^2$
$V_i^{(k+1)} \leftarrow argmin_{V_iV_i^T = I_{r_i}} \|\mathcal{G}_{(i)}^{(k+1)} - D_i^{(k+1)}V_i + (\mathcal{C}_i^{(k)})_{(i)}\|_2^2$
$(\mathcal{C}_i^{(k+1)})_{(i)} \leftarrow (\mathcal{C}_i^{(k)})_{(i)} + \mathcal{G}^{k+1} - D_i^{(k+1)}V_i^{(k+1)T}$
end for
$\mathcal{A}^{(k+1)} \leftarrow \mathcal{G}^{(k+1)} \times_1 \mathcal{U}_1^{(k+1)} \times_2 \mathcal{U}_2^{(k+1)} \times_3 \mathcal{U}_3^{(k+1)}$

3 Data

3.1 Datasets used

For the experiments that have been conducted in this project, we used two sets of data:

- Synthetic data: we simulated VAR models, according to the classical VAR model on which

the paper is based. The transition tensors were generated as in the paper. For this, we fixed the multilinear ranks to (3, 3, 3). Then we generated the core tensors being super diagonal : we hence fixed the singular values. Then we generated random factor matrices, then recover the 3-modes transition tensor thanks to the Tucker decomposition. We did this with different singular values.
Having the transitions tensors, we can easily generate VAR samples recursively. We took zero mean Gaussian noise with covariance matrix equal to I_N.
This set of data was only used to judge the consistency of the multilinear ranks estimators : recall that the idea of this paper is to leverage low-rankness in order to lower the number of parameters in high dimension. Indeed it is necessary to have an idea of the multilinear ranks of the true transition tensor of the VAR model. The paper proposes to that end an estimator of the multilinear ranks from an initial estimator.

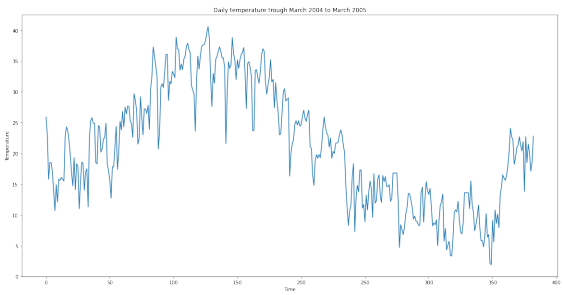
- The second dataset is a real dataset that we found online. This is the dataset on which we tested the alternating least squares algorithm proposed in the paper. It contains information on the quality of the air, on an (approximately) hourly basis from March 2004 to February 2005. More precisely it contains averaged responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device. The device was located on the field in a significantly polluted area, at road level, within an Italian city. Examples of features : concentrations for CO, Non Metanic Hydrocarbons, Benzene, Total Nitrogen Oxides (NOx) and Nitrogen Dioxide (NO2)... as well as the temperature.

3.2 Air Quality Dataset : Preprocessing and Analysis

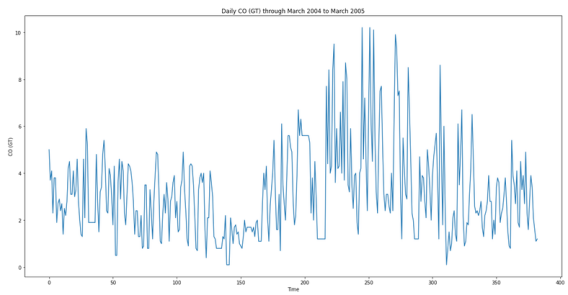
The dataset is available here. The data is of initial dimensionality 9358x16. Here are the important preprocessing steps that we carried out in order to clean the data:

- We first dropped the last two columns, that were unnamed and were full of NaN values, as well as the first column providing a date/time information.
- There were then a non negligeable amount of missing values, which couldn't be just dropped. Hence we adopted a classical filling procedure for time series that was to use the previous (in time) value.
- Then, for the alternating least squares algorithm to not take too much time, we reduce the number of samples. We then took 1/24 of the total number of samples which is equivalent to having one sample per day which seems reasonable when one knows the characteristic times of such climate processes.

The plots below illustrate two features of this dataset, representing measure through (approximately) a year.



Temperature

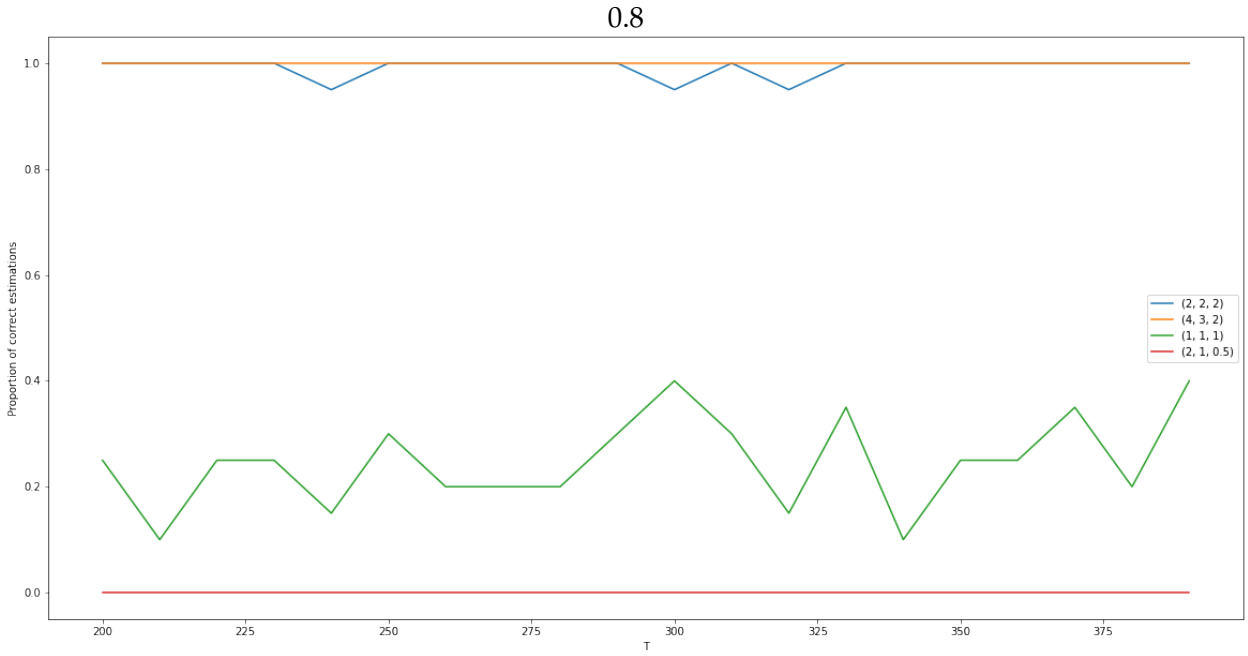


Concentration in CO

A Johansen test has also been carried out though one already knows that this kind of data is not stationary. The values that we obtained were significantly different from one which confirmed the non stationarities that the data presents.

4 Results

We first tested the consistency of the estimator for the multilinear ranks.

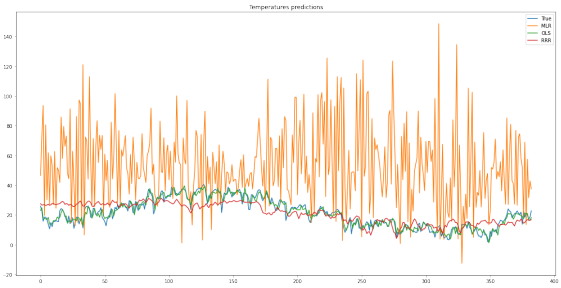


Ranks predictions on different tensors

We can see that the ranks were quite accurate for some singular values and very bad for other singular values. Overall we noticed that the initial estimator was quite good and that allows us to have quite accurate multilinear ranks in the next steps of our experiments.

Moreover we tried to fit the dataset on the Air quality. In the appendix are the the fittings of two features.

5 Appendix



Temperature

Concentration in NO2