

Markov Switching Multiple-equation Tensor Regression

Qing Wang (Ca' Foscari University of Venice)

Joint work with
Roberto Casarin¹ (Ca' Foscari University of Venice)
and
Radu Craiu (University of Toronto)

AHIDI 2024

Nov. 2024

¹The authors acknowledge support from: the MUR - PRIN project '*Discrete random structures for Bayesian learning and prediction*' under g.a. n. 2022CLTYP4 and the Next Generation EU - '*GRINS - Growing Resilient, INclusive and Sustainable*' project (PE0000018), National Recovery and Resilience Plan (NRRP) - PE9 - Mission 4, C2, Intervention 1.3. The views and opinions expressed are only those of the authors and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Table of Contents

- 1 Motivation
- 2 Tensor Algebra
- 3 The Model
- 4 Bayesian Framework
- 5 Simulation Results
- 6 Applications
- 7 Concluding Remarks

Motivation

Data arises naturally in **high-dimensional array (tensor)** structure in many applications, neural-imaging, spatial-temporal analysis, computer vision, financial networks, etc.

Often people are interested in characterizing the relationship between a **scalar outcome** and **tensor covariates** (predictors), high-dimensionality of the covariates introduces a natural challenge in estimating a large number of parameters given a limited sample size.

Contributions:

- Introduce a new flexible tensor model for **multiple-equation** regression that accounts for **common latent regime changes**.
- Provide a suitable inference framework to deal with **over-parametrization** and **overfitting**.
- Propose an efficient MCMC algorithm for posterior approximation (**Random Scan Gibbs Sampling and back-fitting strategy**).

High-dimensional data:

- **A brutal way**: vectorize the tensor predictors and regress the response variable on a large vector of tensor entries with some form of penalization and variable selection.
Overparametrization, ignorance of structural relationship of tensor predictors.
- **Dimensionality reduction**:
 - **Covariates/Predictors**: Zhang et al. (2019) and Caffo et al. (2010) performed PCA and SVD on tensor predictors to obtain a lower dimensional summaries of the predictors.
Unsupervised nature of PCA and interpretability issues.
 - **Coefficients**: Yu and Liu (2016); Wang and Xu (2022); Spencer et al. (2022) performed **Tucker decomposition** on the tensor coefficients, Guhaniyogi et al. (2017); Billio et al. (2023); Papadogeorgou et al. (2021) performed **PARAFAC decompositions**.

Inference

- **Frequentist**: Yu and Liu (2016) proposed an algorithm to minimize the empirical loss function. Zhou et al. (2013) proposed a MLE estimator.
- **Bayesian**: Guhaniyogi et al. (2017) proposed a novel multi-way shrinkage prior to induce further sparsity on the PARAFAC decomposed tensor elements. Papadogeorgou et al. (2021) extended their work by adding another layer of flexibility on the PARAFAC decomposition to achieve better inference performance.

Sampling strategy:

- Random Partial Scan Gibbs Sampling: Łatuszyński et al. (2013), Yang et al. (2019).
- Backfitting strategy: Härdle and Hall (1993).

Tensor Algebra

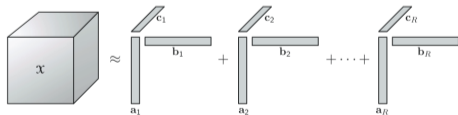
Tensor Representation

Two major tensor representation methods often used in the literature: CP (CANDECOMP/PARAFAC) and Tucker. (Kolda and Bader (2009))

CP Representation: given a 3-mode tensor $\mathcal{B} \in \mathbb{R}^{I \times J \times K}$

$$\mathcal{B} = \sum_{d=1}^D \mathbf{a}_d \otimes \mathbf{b}_d \otimes \mathbf{c}_d$$

where $\mathbf{a}_d \in \mathbb{R}^I$, $\mathbf{b}_d \in \mathbb{R}^J$, $\mathbf{c}_d \in \mathbb{R}^K$ are the marginals from the CP decomposition, D is the rank of the tensor, \otimes represents the *outer* product.



Dimensionality reduction: $I \times J \times K \rightarrow (I + J + K)D$.

Tensor Algebra

Hard vs Soft PARAFAC

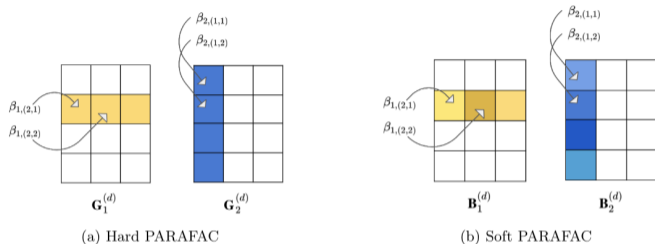


Figure: Hard vs Soft PARAFAC (Papadogeorgou et al., 2021)

$$\beta_{m,jm}^{(d)} \sim \mathcal{N}_{q_m} \left(\gamma_{m,jm}^{(d)}, \tau \sigma_m^2 \zeta^{(d)} I_{q_m} \right), q_m = \prod_{l \neq m} p_l$$

$$\mathbb{E} \left(\mathbf{B} | \gamma_1^{(d)}, \gamma_2^{(d)} \right) = \sum_{d=1}^D \mathbb{E} \left(\mathbf{B}_1^{(d)} \right) * \mathbb{E} \left(\mathbf{B}_2^{(d)} \right) = \sum_{d=1}^D \mathbf{G}_1^{(d)} * \mathbf{G}_2^{(d)} = \sum_{d=1}^D \gamma_1^{(d)} \circ \gamma_2^{(d)}$$

The Model

A Markov-Switching Multiple-equation Tensor Regression Model:

$$\begin{cases} y_{1t} = \mu_1(\mathbf{s}_t) + \langle \mathbf{B}_1(\mathbf{s}_t), \mathbf{X}_t \rangle + \sigma_1(\mathbf{s}_t)\varepsilon_{1t} \\ \vdots \\ y_{Nt} = \mu_N(\mathbf{s}_t) + \langle \mathbf{B}_N(\mathbf{s}_t), \mathbf{X}_t \rangle + \sigma_N(\mathbf{s}_t)\varepsilon_{Nt} \end{cases} \quad (1)$$

where $t = 1, \dots, T$, $\mathbf{X}_t, \mathbf{B}_\ell(\mathbf{s}_t)$ are $p_1 \times p_2$ matrices, $\langle \cdot, \cdot \rangle$ denotes inner product. The latent process is a K -state Markov chain process and the parametrization used is

$$\mu_\ell(\mathbf{s}_t) = \sum_{k=1}^K \mu_{\ell k} \mathbb{I}(\mathbf{s}_t = k), \quad \mathbf{B}_\ell(\mathbf{s}_t) = \sum_{k=1}^K \mathbf{B}_{\ell k} \mathbb{I}(\mathbf{s}_t = k), \quad \sigma_\ell(\mathbf{s}_t) = \sum_{k=1}^K \sigma_{\ell k} \mathbb{I}(\mathbf{s}_t = k)$$

Assume the following decomposition:

$$\mathbf{B}_{\ell k} = \sum_{d=1}^D \mathbf{B}_{\ell, k, 1}^{(d)} * \mathbf{B}_{\ell, k, 2}^{(d)}$$

where $*$ is the *Hadamard product*, $\mathbf{B}_{\ell, k, m}^{(d)}$, $m = 1, 2$ are the multiplicative factors. D is the number of components used to decompose the tensor.

Hierarchical Priors

We use shrinkage priors to favor sparsity:

$$B_m^{(d)} \sim \mathcal{MN}_{p_1, p_2} \left(G_m^{(d)}, \tau \sigma_m^2 \zeta^{(d)} I_{p_1}, I_{p_2} \right) \quad (2)$$

$$\gamma_m^{(d)} \sim \mathcal{N}_{p_m}(\mathbf{0}, \tau \zeta^{(d)} W_m^{(d)}) \quad (3)$$

$$w_{m, j_m}^{(d)} \sim \text{Exp}((\lambda_m^{(d)})^2 / 2) \quad (4)$$

$$\lambda_m^{(d)} \sim \mathcal{Ga}(a_\lambda, b_\lambda) \quad (5)$$

$$\sigma_m^2 \sim \mathcal{Ga}(a_\sigma, b_\sigma) \quad (6)$$

$$\tau \sim \mathcal{Ga}(a_\tau, b_\tau) \quad (7)$$

$$(\zeta^{(1)}, \dots, \zeta^{(D)}) \sim \text{Dir}(\alpha/D, \dots, \alpha/D) \quad (8)$$

where $m = \{1, 2\}$ is the number of mode, p_1, p_2 are the size of each mode.

$$G_m^{(d)} = \begin{cases} \gamma_1^{(d)} \circ \iota_{p_2} & m = 1 \\ \iota_{p_1} \circ \gamma_2^{(d)} & m = 2 \end{cases}$$

Selection of Hyperparameters

The choice of **hyperparameters** can have a large effect on the performance of the model. We follow the strategy in (Papadogeorgou et al. (2021)) to choose the hyperparameters by studying the properties of **induced prior variance** on the coefficients B .

In particular, we choose the hyperparameters such that $\text{Var}(B_{ij}) = V^*$ and the additional variance introduced by the softening equals to AV^* . B_{ij} denotes the entry of B .

We found that:

$$\mathbb{V}(B_{ij}) = \frac{a_\tau(a_\tau + 1)}{b_\tau^2} C \left(\frac{a_\sigma}{b_\sigma} + \frac{2b_\lambda^2}{(a_\lambda - 1)(a_\lambda - 2)} \right)^2 \quad (9)$$

$$\frac{a_\sigma}{b_\sigma} = \frac{b_\tau}{a_\tau} \sqrt{\frac{a_\tau V^*}{(a_\tau + 1)C}} \left(1 - \sqrt{1 - AV^*} \right) \quad (10)$$

where $C = \frac{\frac{\alpha}{D} + 1}{\alpha + 1}$.

In simulation we use $V^* = 1$ and $AV^* = 10\%$.

Full Conditionals

Let $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K)$ be the collection of the **state-specific** parameters $\boldsymbol{\theta}_k = (\boldsymbol{\beta}_k, \gamma_k, \zeta_k, \tau_k, \boldsymbol{\lambda}_k, \mathbf{w}_k, \sigma_k^2, \mu_k)$ and denote with $\mathbf{y} = (y_1, \dots, y_T)$, $\mathbf{X} = (X_1, \dots, X_T)$ and $\mathbf{s} = (s_1, \dots, s_T)$ the collection of **response variables**, **covariates** and **state variables**, respectively.

The joint posterior of the unknowns of the model is given by

$$p(\boldsymbol{\theta}, \mathbf{s} \mid \mathbf{y}, \mathbf{X}) \quad (11)$$

The joint posterior is not tractable, we approximate using the full conditionals for each of the parameters.

MCMC-Gibbs Sampler

We propose a MCMC procedure based on Gibbs sampling to sample the unknowns from 3 blocks.

- Block 1: Sampling $\beta_{m,j_m}^{(d)}, \gamma_{m,j_m}^{(d)}, \sigma_m^2, \sigma^2, \mu$ from $p(\beta_{m,j_m}^{(d)}, \gamma_{m,j_m}^{(d)}, \sigma_m^2, \sigma^2, \mu \mid \mathbf{Y}, \mathbf{X}_1, \dots, \mathbf{X}_T)$
- Block 2: Sampling $\zeta^{(d)}$ and τ from $p(\zeta^{(d)}, \tau \mid \mathbf{B}, \gamma, \mathbf{w})$
- Block 3: Sampling $\lambda_m^{(d)}$ and $w_{m,j_m}^{(d)}$ from $p(\lambda_m^{(d)}, w_{m,j_m}^{(d)} \mid \gamma_{m,j_m}^{(d)}, \tau, \zeta^{(d)})$

For the hidden states, we apply a **Forward Filtering Backward Sampling** (FFBS) strategy:

- Draw transitional probabilities (p_{1k}, \dots, p_{Kk}) from Dirichlet distribution $p(p_{1k}, \dots, p_{Kk} \mid \mathbf{s})$.
- Compute iteratively the vector of smoothed probabilities $\xi_{t|T}$ by using Hamilton Filter, and draw the state vector s_t from a multinomial distribution $\mathcal{M}(\mathbf{1}, \xi_{t|T})$.

For the first 10 Gibbs iterations, we run full scan for every rank and every mode to recover the main structure of the coefficients. Then we perform Random-Partial-Scan Gibbs to randomly select a subset of components to update for each iteration.

Simulations

Tensor Regression

Simulation settings:

- 4 experimental settings ranging from different ranks and **different levels of sparsity**.
- Matrix predictor with dimensions 20×20
- Number of observations: 400
- Gibbs iterations: 3000

▶ robustness checks

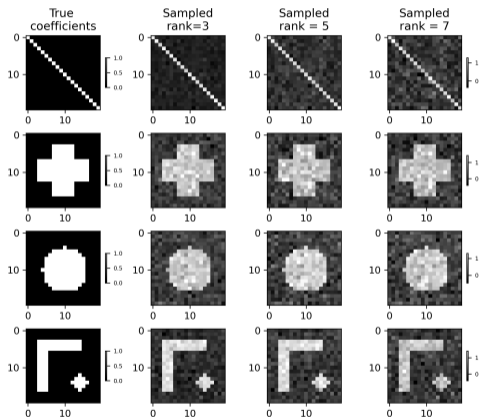


Figure: Estimated coefficients for four experimental settings using three different ranks

Simulations

Tensor Regression

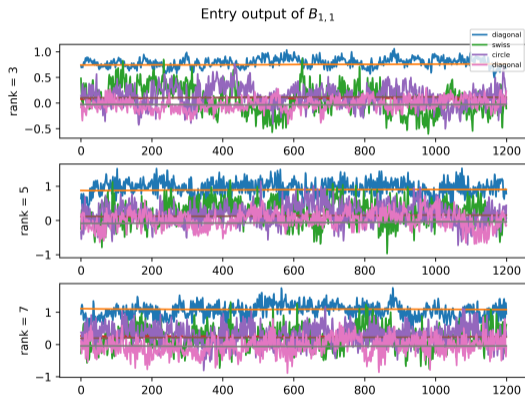


Figure: Raw MCMC output and progressive average of entry $B_{1,1}$ for different types of coefficients

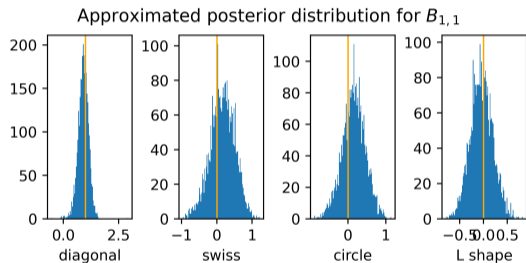


Figure: Approximated posterior distribution

Simulation

Markov Switching

Simulation settings

- 2 sets of true coefficients are used to represent 2 different regimes, both **i.i.d** covariates and **AR(1)** covariates are used in the simulation.
- Matrix predictor with dimensions 12×12
- Regime specific intercepts: $\mu_1 = \mu_2 = 0$
Regime specific variances: $\sigma_1^2 = 2, \sigma_2^2 = 0.1$.
- Number of observations: 800
- Gibbs iterations: 3000

▶ more results

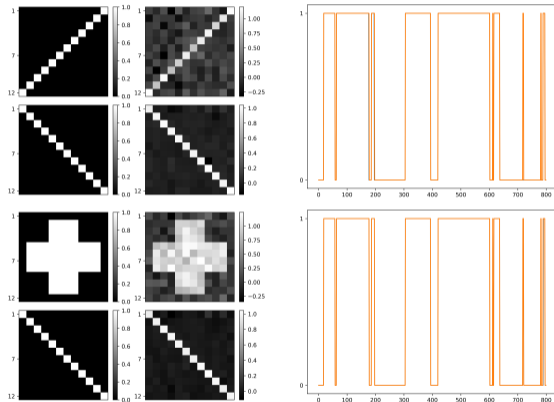


Figure: Markov-switching model with Diagonal and Anti-diagonal coefficients (first row) and with Cross and Diagonal coefficients (second row).

Table: MCMC convergence and efficiency

Setting S_1^{MS} (anti-diag / diag)					
	ACF(1)	ACF(5)	ACF(10)	MSE(10)	MSE(100)
Coefficients (B)	0.4085 (0.3145)	0.3279 (0.2328)	0.3158 (0.0980)	0.0559	0.0083
States(s_t)	0.5624 (0.5448)	0.5437 (0.3878)	0.5333 (0.1942)	0.2725	0.0113
Setting S_2^{MS} (cross / diag)					
Coefficients (B)	0.5139 (0.4247)	0.4425 (0.2819)	0.4410 (0.1650)	0.1773	0.0106
States(s_t)	0.5294 (0.5153)	0.5166 (0.3649)	0.5077 (0.1831)	0.3013	0.0050

Table 1 documents the results on convergence for the two different experimental settings. The second column of the table reports the ACFs of the parameters and the hidden states before and after thinning, where the results after thinning are reported in parentheses. The third column reports the MSE of the parameters and hidden states at the 10th and 100th Gibbs iteration.

Motivation

Time series model with many lags, mixed-frequency data sampling can take the advantage of tensor data structure.

Applications

Two relevant applications: i) financial market volatility and ii) oil prices data.

Macro Application

Stock return and oil prices

We examine the impact of oil price volatility on the stock market returns (S&P 500) at an **aggregate level** and on the **financial** sector, **energy** sector and other sectors of S&P500 at **disaggregate level**. We follow Xiao and Wang (2022) to distinguish the oil price volatility into **Good Oil Volatility** (GV) where the realized volatility is positive and **Bad Oil Volatility** (BV) where the realized volatility is negative.

Furthermore, we explore the problem in a **Mixed Data Sampling** (MIDAS) (Ghysels et al., 2004) framework and construct the covariates into a **3d array** to take advantage of the tensor regression. To fix the idea:

- Response variable: stock return $R_{\ell,t}$ (4-weekly data), $\ell = \{\text{S\&P 500, financial sector, energy sector, other sectors in S\&P 500}\}$
- Regressors: Good oil (GV) and Bad oil (BV) volatilities, exchange rate volatility (ER), TED spread volatility (IR) and VIX index (VI). (Weekly data)

Macro Application

Stock return and oil prices

The tensor regression model

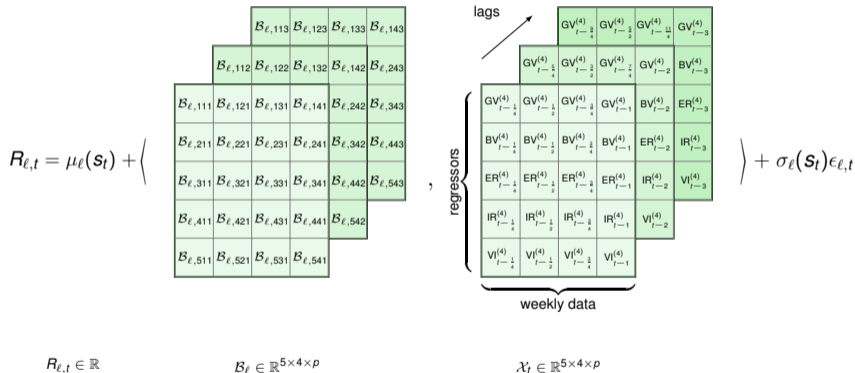


Figure: Graphic Representation of Tensor Regression for Macro Application

Macro Application

Stock return and oil prices

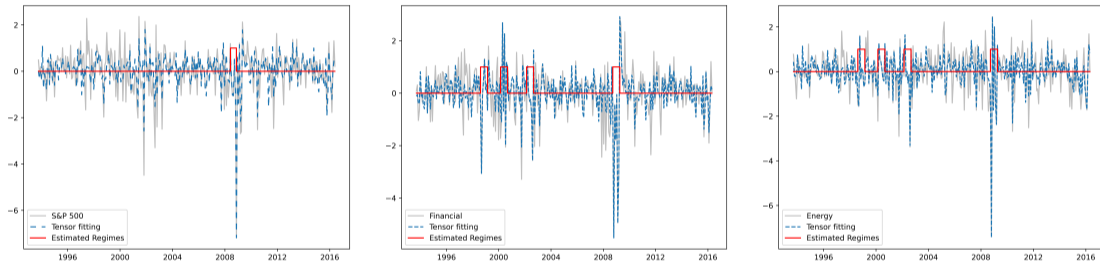


Figure: Tensor Regression with Markov Switching (blue dashed line) and estimated hidden states (red solid line). True data is shown in solid silver line)

Model Performance

	In Sample		Out of Sample			
			1-day (month) ahead		5-day (month) ahead	
	MSE	MAE	MSE	MAE	MSE	MAE
<i>Application 1: VIX and OVX on macro indicators</i>						
Least Square	0.3049	0.4266	0.1945	0.3474	0.3668	0.5211
LASSO	0.4207	0.5259	0.5199	0.6363	0.6940	0.7589
Tensor	0.3097	0.4324	0.2540	0.4232	0.3581	0.5182
MS Tensor	0.0907	0.2393	0.1409	0.3342	0.1379	0.3063
<i>Application 2: S&P 500 on oil prices (Aggregate Analysis)</i>						
Least Square	0.4418	0.5126	0.2394	0.4893	0.1986	0.4073
LASSO	0.4692	0.5264	0.2058	0.4537	0.1487	0.3548
Tensor	0.6073	0.6084	0.0445	0.2111	0.3442	0.4227
MS Tensor	0.3901	0.4794	0.5248	0.7244	0.2664	0.4522
<i>Application 2: Financial Sector on oil prices (Disaggregated S&P 500 analysis)</i>						
Least Square	0.5198	0.5530	0.3082	0.5551	0.1107	0.2631
LASSO	0.5532	0.5650	0.2684	0.5181	0.1111	0.2555
Tensor	0.6968	0.6204	0.0035	0.0594	0.0574	0.2025
MS Tensor	0.3370	0.4308	0.0586	0.2421	0.0878	0.2696
<i>Application 2: Energy Sector on oil prices (Disaggregated S&P 500 analysis)</i>						
Least Square	0.4587	0.5354	0.6606	0.8128	0.2879	0.4931
LASSO	0.4869	0.5516	0.1756	0.4191	0.2740	0.4545
Tensor	0.5379	0.5731	0.2414	0.4914	0.2788	0.4949
MS Tensor	0.3323	0.4362	0.0090	0.0949	0.3325	0.5153
<i>Application 2: Other sectors on oil prices (Disaggregated S&P 500 analysis)</i>						
Least Square	0.4389	0.5097	0.5919	0.7694	0.4054	0.5672
LASSO	0.4675	0.5254	0.3940	0.6277	0.2908	0.5071
Tensor	0.4643	0.5219	0.1072	0.3274	0.2935	0.4871
MS Tensor	0.3064	0.4105	0.9498	0.9746	0.7229	0.7488

Table: In-sample fitting and out-of-sample forecasting performance. Results for the best performing model are in boldface.

Concluding Remarks

- A new Markov switching multiple-equation tensor regression model capable of extracting a common latent factor (latent regime changes) is proposed.
- A low-rank representation of the coefficient tensor and hierarchical prior distribution are proposed to introduce shrinkage effects to overcome overparametrization.
- An efficient MCMC sampler is proposed based on back-fitting and random scan strategies.
- The tensor regression model is readily to be used with tensor covariates with order 2 or 3.

Thanks for your attention!

qing.wang@unive.it

Preprint available at:

<https://arxiv.org/abs/2407.00655>

References I

- Billio, M., Casarin, R., Iacopini, M., and Kaufmann, S. (2023). Bayesian dynamic tensor regression. *Journal of Business & Economic Statistics*, 41(2):429–439.
- Caffo, B. S., Crainiceanu, C. M., Verduzco, G., Joel, S., Mostofsky, S. H., Bassett, S. S., and Pekar, J. J. (2010). Two-stage decompositions for the analysis of functional connectivity for fmri with application to alzheimer's disease risk. *NeuroImage*, 51(3):1140–1149.
- Ghysels, E., Santa-Clara, P., and Valkanov, R. (2004). The midas touch: Mixed data sampling regression models.
- Guhaniyogi, R., Qamar, S., and Dunson, D. B. (2017). Bayesian tensor regression. *The Journal of Machine Learning Research*, 18(1):2733–2763.
- Härdle, W. and Hall, P. (1993). On the backfitting algorithm for additive regression models. *Statistica neerlandica*, 47(1):43–57.

References II

- Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500.
- Papadogeorgou, G., Zhang, Z., and Dunson, D. B. (2021). Soft tensor regression. *J. Mach. Learn. Res.*, 22:219–1.
- Spencer, D., Guhaniyogi, R., Shinohara, R., and Prado, R. (2022). Bayesian tensor regression using the tucker decomposition for sparse spatial modeling.
- Wang, K. and Xu, Y. (2022). Bayesian tensor-on-tensor regression with efficient computation. *arXiv preprint arXiv:2210.11363*.
- Xiao, J. and Wang, Y. (2022). Good oil volatility, bad oil volatility, and stock return predictability. *International Review of Economics & Finance*, 80:953–966.
- Yang, J., Levi, E., Craiu, R. V., and Rosenthal, J. S. (2019). Adaptive component-wise multiple-try metropolis sampling. *Journal of Computational and Graphical Statistics*, 28(2):276–289.

References III

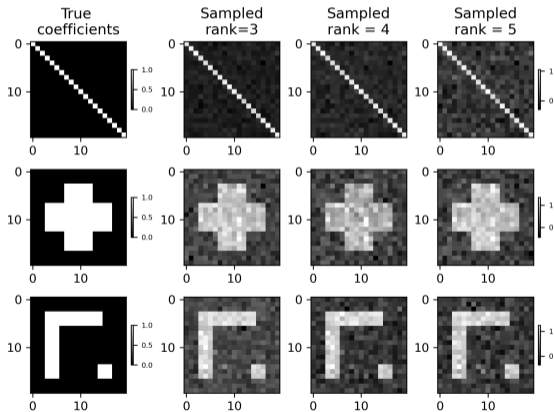
- Yu, R. and Liu, Y. (2016). Learning from multiway data: Simple and efficient tensor regression. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 373–381, New York, New York, USA. PMLR.
- Zhang, Z., Allen, G. I., Zhu, H., and Dunson, D. (2019). Tensor network factorizations: Relationships between brain structural connectomes and traits. *Neuroimage*, 197:330–343.
- Zhou, H., Li, L., and Zhu, H. (2013). Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552.
- Łatuszyński, K., Roberts, G. O., and Rosenthal, J. S. (2013). Adaptive Gibbs samplers and related MCMC methods. *The Annals of Applied Probability*, 23(1):66 – 98.

Robustness Check

We tweaked a bit with hyperparameters to change the prior mean and variance of the scales while still maintaining $V^* = 1, AV^* = 10\%$.

	benchmark	robustness
α	1	1
a_σ	0.5	0.5
b_σ	$8.5\sqrt{C}$	$2\sqrt{C}$
a_τ	3	3
b_τ	$33.75\sqrt{C}/b_\sigma$	$33.75\sqrt{C}/b_\sigma$
a_λ	3	3
b_λ	$a_\lambda^{1/4}$	$a_\lambda^{1/2}$

◀ back



Robustness Check

Noisy True Coefficients

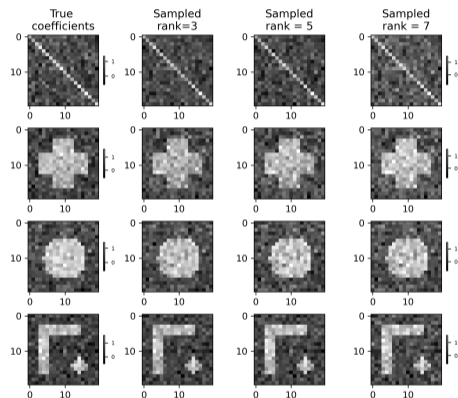


Figure: Estimation results with noisy true coefficients

Simulations Results

Computational cost

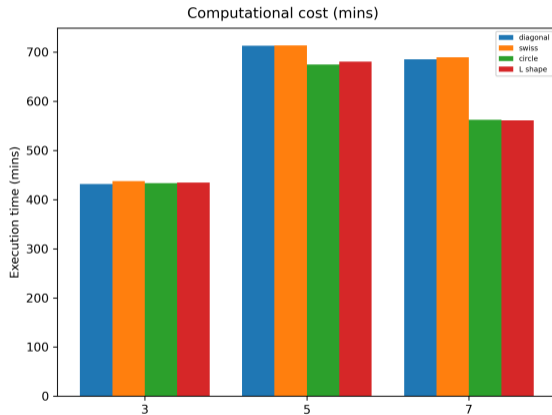


Figure: Computational cost (mins)

Simulations Results

Autocorrelation

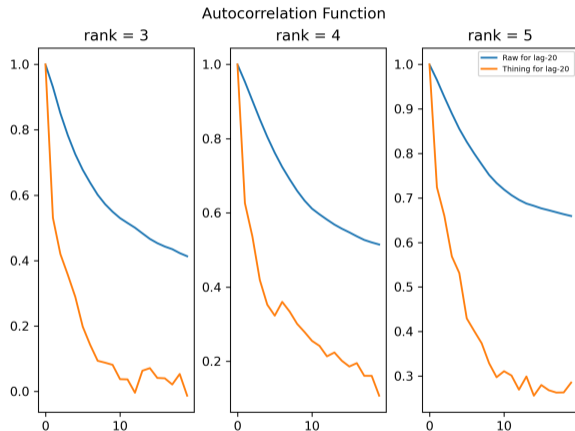


Figure: Autocorrelation before and after thinning

Simulation Results

Intercepts and Variance of MS

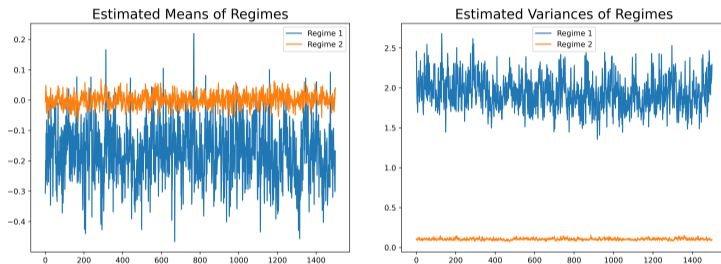


Figure: Trace plots after removing the burn-in samples for regime specific intercepts (left) and variances (right) for the S_2^{MS} experimental setting. True values are $\mu_1 = \mu_2 = 0$, $\sigma_1^2 = 2$, $\sigma_2^2 = 0.1$.

← back