HOSVD Tensor Emulator for Spatiotemporal Emulators

Evgeny Noi

UC Santa Barbara, Department of Geography

November 23, 2021

Sources:

1. Pratola, M. T., & Chkrebtii, O. (2018). Bayesian calibration of multistate stochastic simulators. Statistica Sinica, 693-719.

2. Gopalan, G., & Wikle, C. K. (2021). A Higher-Order Singular Value Decomposition Tensor Emulator for Spatiotemporal Simulators. Journal of Agricultural, Biological and Environmental Statistics, 1-24.

Problem

- Modeling spatiotemporal processes is **complicated** due to inherent complex `dependencies` across time, space and processes themselves (non-linearity, non-stationarity, non-separability). The process cannot be easily speicified as deterministic black-box. How do we add randomness and uncertainty to simulator for *ensemble realizations*?

Part 1: Emulators

Emulators

> “ Emulator is a function that mimics the output of a simulator at a fraction of simulator's cost. The function is most frequently specified by Gaussian Processes, polynomial basis expansions, and non-linear surrogate models. ”

Can we infer unknown calibration parameters from noisy indirect observation of system states at discrete spatial locations?

Calibration Setting

$$y|\theta, \delta, \kappa, \Lambda_f \sim N(\kappa x(\theta) + \delta, \Lambda_f I)$$

We model a state $x(s; \theta)$ at spatiotemporal locations $s_i \in S$ and unknown calibration parameters $\theta \in \Theta$ from partial and indirect observations $y(s)$. $\delta(s)$ - additive systemic discrepency between the simulator and the true state, and κ is a multiplicative discrepency. Λ_f denotes $n \times n$ precision matrix denoting uncertainty. Likelihood is described by equation above, state is modeled as a realization of the Gaussian Process (GP), discrepancy is also modeled as GP.

Part 2: Model Formulation

- u - refers to u-th realization from multistate stochastic simulation
- k - refers to k-th state ouptut from multistate stochastic simulation
- i - denotes i-th output grid setting
- j - refers to j-th setting of the calibration parameter vector
- n - number of locations, p - number of variables, n_S - number of states, N - number of simulation realizations (calibrations) using MCMC,
- the u-th realization of $x_k(s_i, \theta_j)$ for $u = 1, \ldots, N$.
- tensorized representation: $\mathcal{X} \in R^{m \times n_s \times n \times N} = x_{u,k,i,j}$

$$[\mathcal{X}]_{u,k,i,j} = \sum_{r_1}^{R} \sum_{r_2}^{R} \sum_{r_3}^{R} \sum_{r_4}^{R} \mathcal{E}_{r_1,r_2,r_3,r_4} \, a^{(1)}_{u,r_1} \, a^{(2)}_{k,r_2} \, a^{(3)}_{i,r_3} \, a^{(4)}_{j,r_4}$$

where R_1, R_2, R_3, R_4 denote the rank of approximation, $\mathcal{E}$ is the $R_1 \times R_2 \times R_3 \times R_4$ core tensor (analogous to diagonal weight matrix in SVD), and $a^{(1)}_{u,r_1} \in A^{(1)}$, is an entry in $N \times R_1$ factor matrix $A^{(1)}$, the analogue of eigenvector in the SVD. The $\mathcal{X}$ is a tensor-variate Gaussian Process. Thus HOSVD decomposes tensor into: A) effects of variability across runs, B) variability across states; C) variability across spatio-temporal grid; D) variability across stochastic realizations of the simulator.

Putting it into a tensor

Left singular vectors of R_4 can be seen as arising from latent eigenfunctions that describe the variability of the tensor across the m simulator runs, motivating the use of a Gaussian process prior. Thus we reconstruct a missing entry in our tensor representation of simulator outputs, the trajectory of the simulator at the unknown calibration parameter setting θ, by modeling the appropriate eigenvectors.

Modeling Simulator Realizations

Let $X_{(4)}^T = UDV^T$ be the SVD of the transposed mode-4 matricization: $(N \cdot n \cdot n_s) \times m$, U is the same shape as $X_{(4)}^T$ and D and V are both $m \times m$. To decrease the computational burden we utilize low-rank approximation using n_c EOFs, such that $n_c < m$:

$$x_{ukij} \approx \sum_{l=1}^{n_c} v_l(\theta_j) U_{ukil}$$

where $v_l(\theta_j) = [V]_{jl}$ and the number of bases n_c is determined via cross-validation.

Part 3: Application

Application: stochastic water temperature model

The simulator requires `nearby air temperature` data to capture the **short-term fluctuations** of observed `water temperatures`. The **annual trend** is separated from the short term fluctuation by fitting a simple **sinusoid** to capture annual seasonal variation.

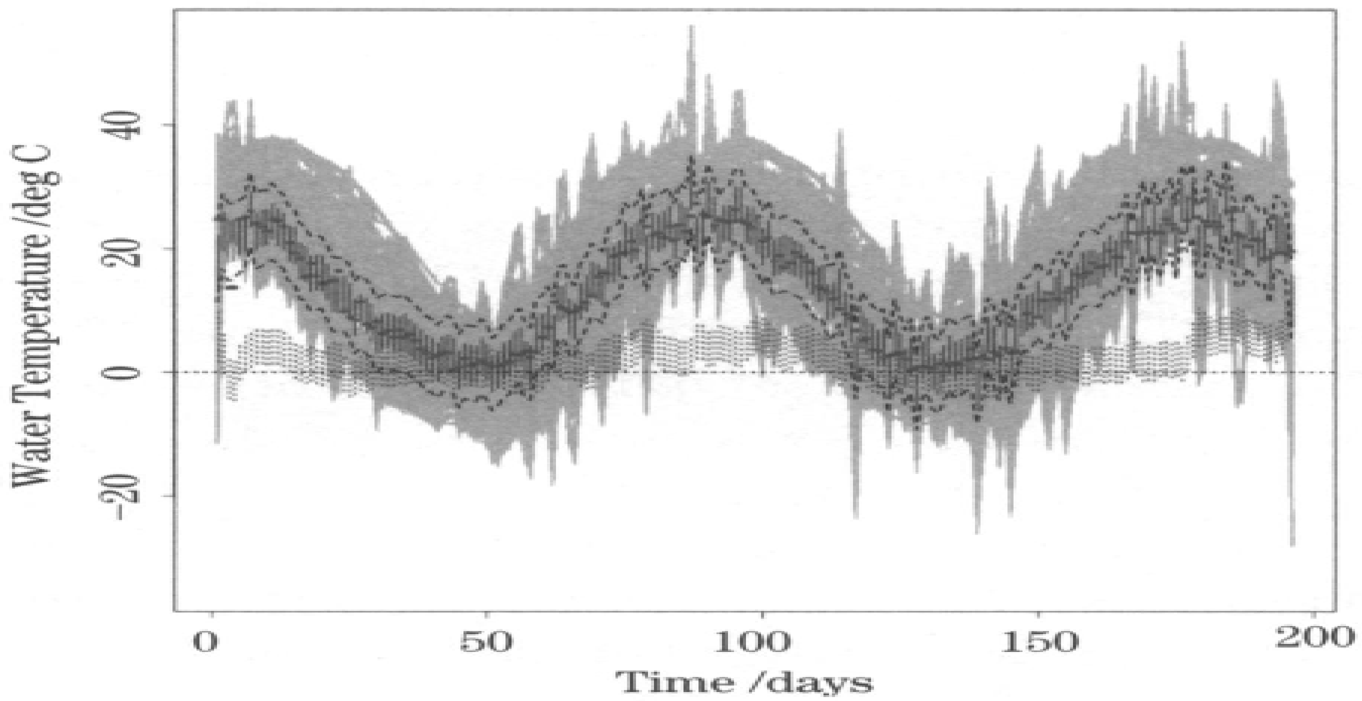
$$T_w(t) = T_a(t) + R_w(t)$$

$$T_a(t) = a_1 + a_2 sin(\frac{2\pi}{365}(t - t_0))$$

$$R_w(t) = KR_a(t) + \epsilon$$

Application: stochastic water temperature model

Calibration parameters: $\theta = (a_1, a_2, t_0, K, \sigma)^T$ - overall temperature level, seasonal component scale, seasonal component offset, thermal diffusivity, short-term fluctuation deviation.



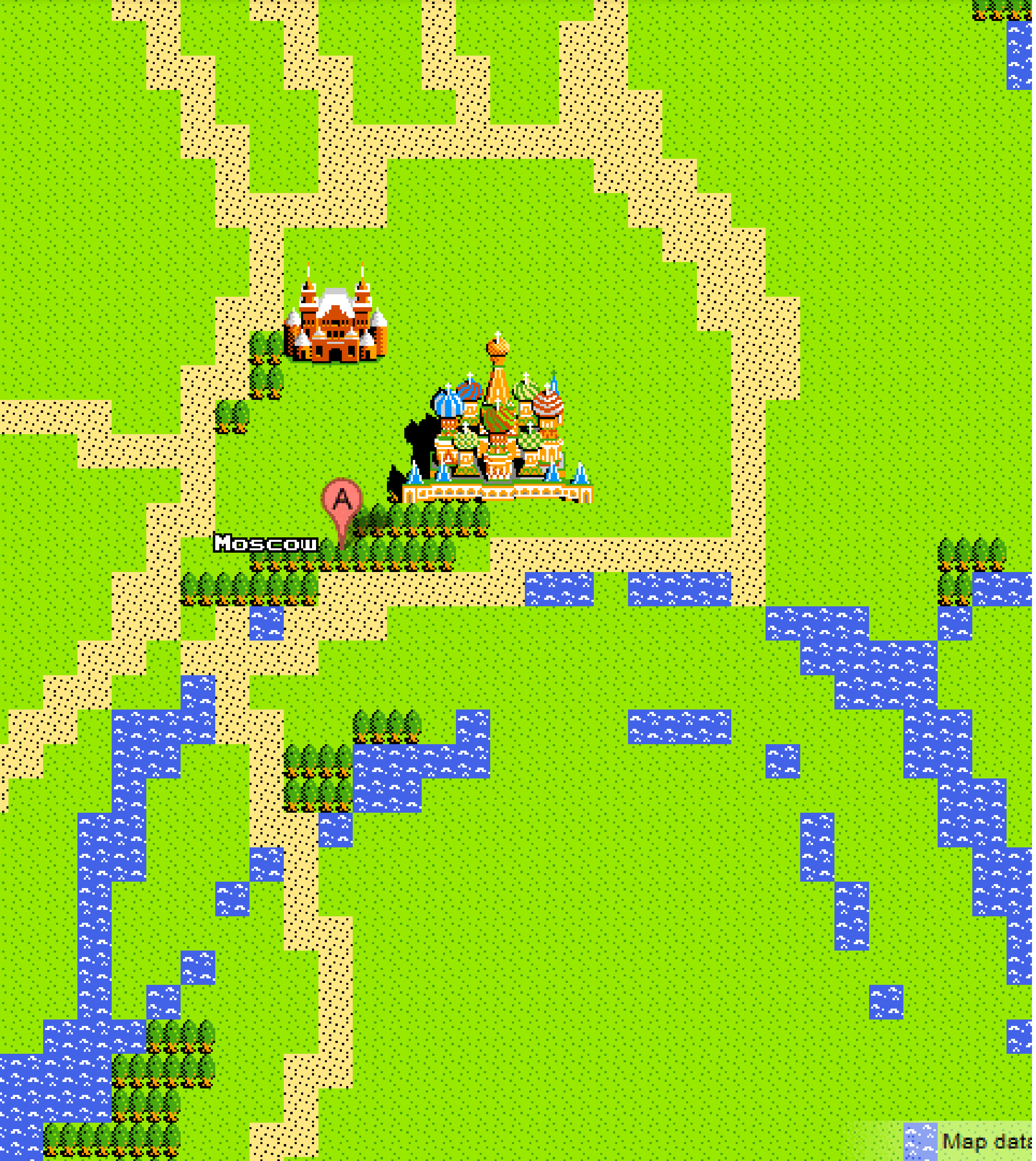
Application: stochastic water temperature model

> “ The standard deviations of the posterior distributions for the predicted process are similar for both models, but the standard deviation for the discrepancy when accounting for simulator uncertainty (0.776) was about 10% smaller than when p ”

Part 4: Conclusion

Conclusion

- This presentation examplified the usage of tensorization and HOSVD for model calibration in PDE and AB models
- In both papers the authors describe procedure for identifying basis vectors via cross-validation/scree plots. However, the higher order orthogonal iteration (HOOI) can help calculate orthonormal basis of the dominant subspace.



Questions?

Evgeny Noi

> “If geography is a prose, maps are iconography. (Lennar Meri)”

noi@ucsb.edu