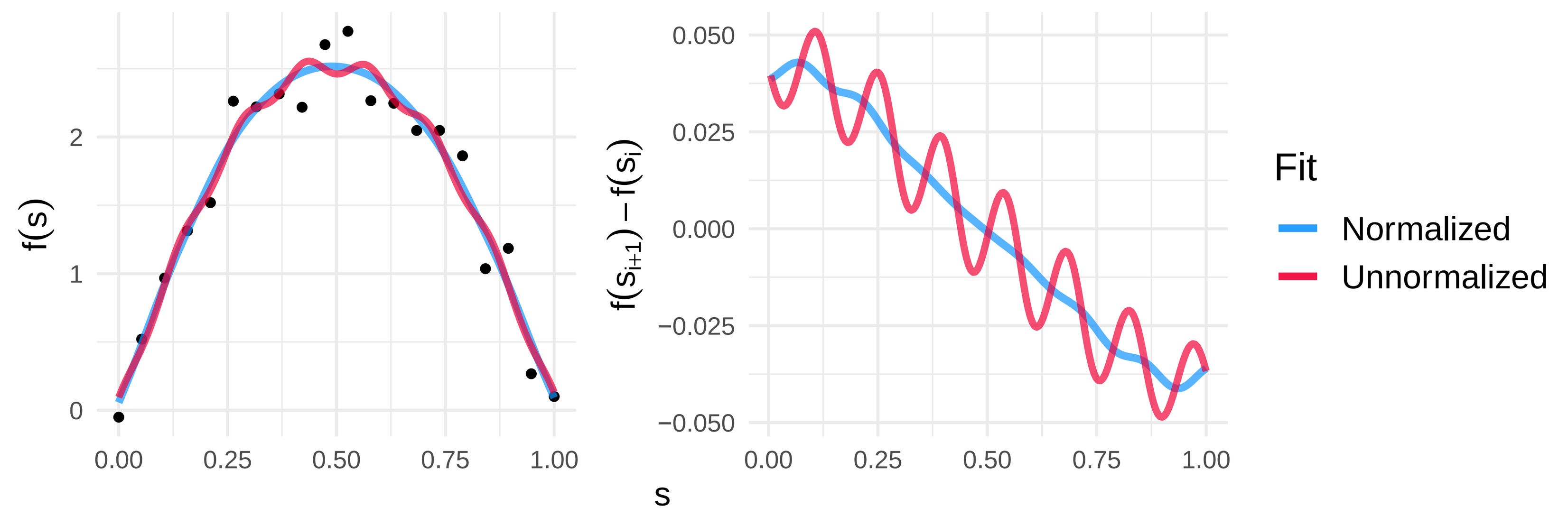
Basis for Change: Approximate Stationary Models for Large Spatial Data

Antony Sikorski   Daniel McKenzie   Douglas Nychka

Department of Applied Mathematics and Statistics, Colorado School of Mines

Introduction

- Fitting a Gaussian Process (GP) is slow for big data ($\Sigma^{-1}, |\Sigma|$).
- Basis function models efficiently approximate the GP [1].
- To approximate a stationary process and remove undesirable artifacts, computationally expensive basis function normalization is needed.
- We propose two fast methods for this normalization, and implement them within the `LatticeKrig` R package [3].



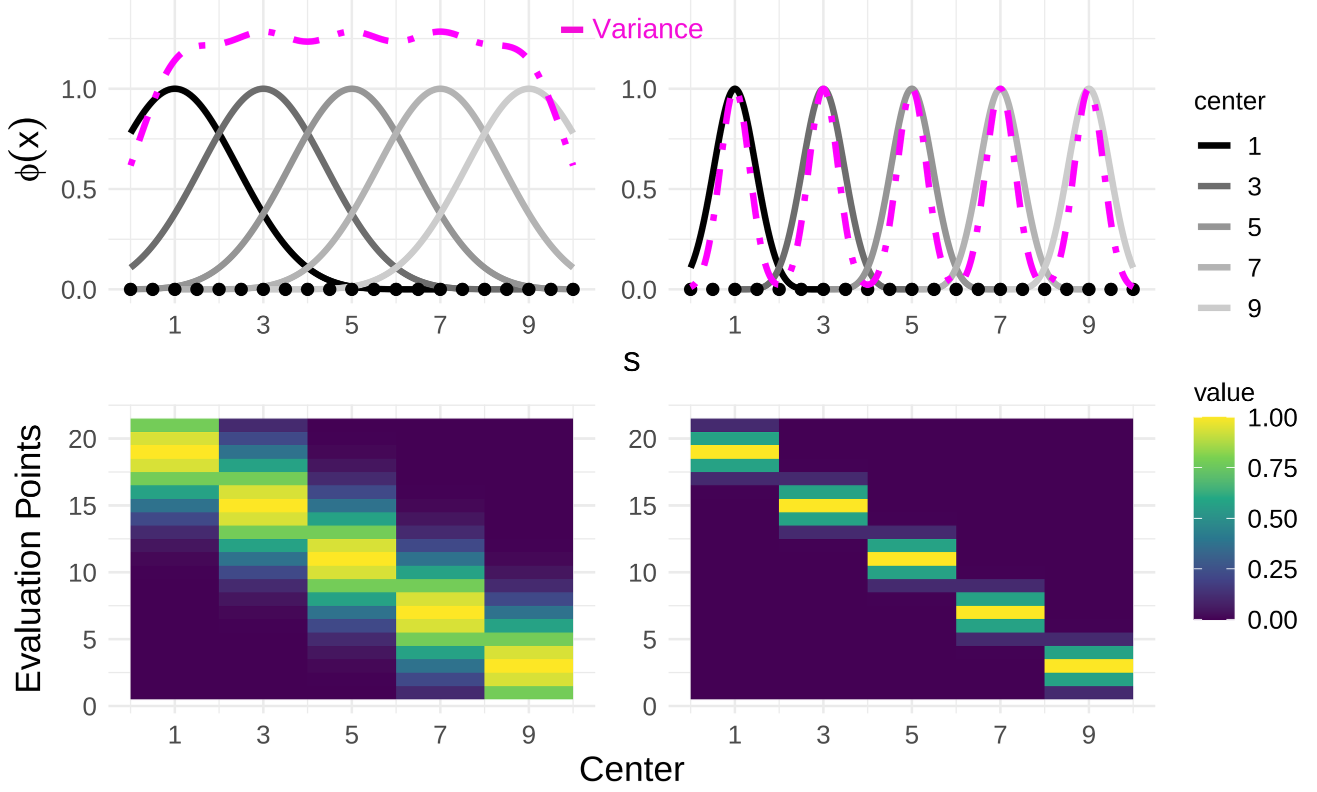
LatticeKrig Spatial Model

The LatticeKrig framework [2] introduces sparsity into key matrices to allow for a large number of basis functions:

- Basis function model for spatial observations Z at 2-d locations $\mathbf{s}$:

$$Z(\mathbf{s}) = \mathbf{x}(\mathbf{s})^\top \boldsymbol{\beta} + g(\mathbf{s}) + \varepsilon(\mathbf{s}), \quad \text{where } g(\mathbf{s}) = \sum_{i=1}^{R} c_i \phi_i(\mathbf{s})$$

  where $\mathbf{x}$ is a vector of covariates, $\boldsymbol{\beta}$ are linear coefficients, and ε is error.
- g is a GP, approximated by the sum of R compact radial basis functions ϕ with random coefficients $\mathbf{c}$.
- $\mathbf{c}$ follow SAR model, $B\mathbf{c} = \mathbf{e}$, where B is sparse and $\mathbf{e} \sim \mathcal{N}(0, 1)$. This directly gives the sparse precision matrix $Q = BB^\top$.
- Direct prescription of $Q = \Sigma^{-1}$ lets you avoid expensive GP operations.
- Problem: Low basis function overlap (sparse) creates undesirable artifacts



Slow Solution: Normalize functions to have constant marginal variance by performing $\phi_i^*(\boldsymbol{s}) = \phi_i(\boldsymbol{s})/\sqrt{\mathrm{Var}(g(\boldsymbol{s}))}$.

- This is expensive! Let $\boldsymbol{\phi_s} = [\phi_1(\boldsymbol{s}) \ldots \phi_R(\boldsymbol{s})]^T$, then

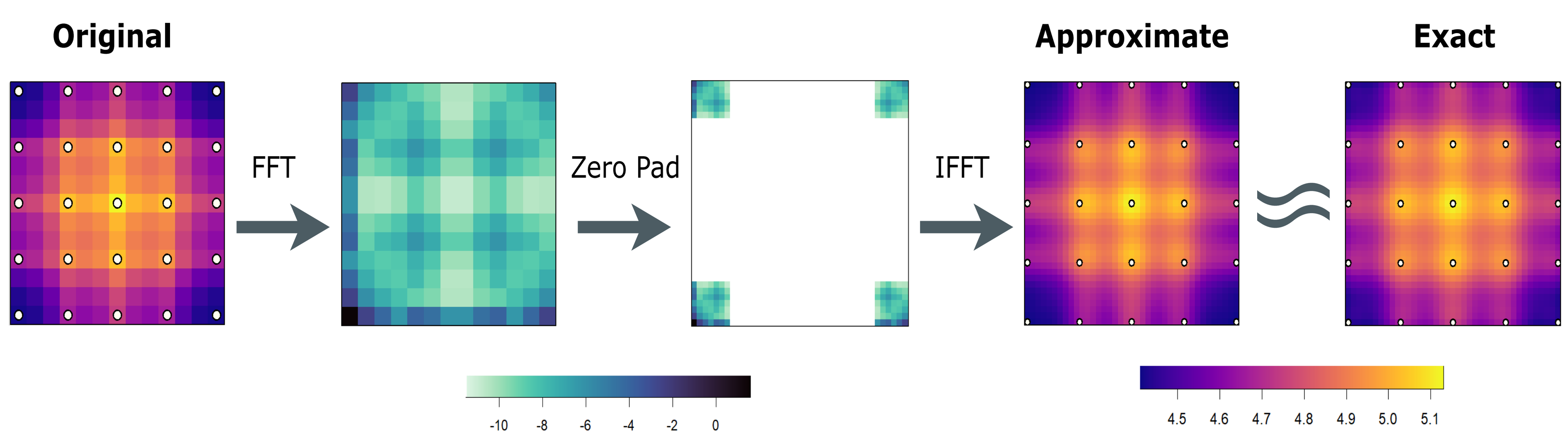
$$\mathrm{Var}(g(\boldsymbol{s})) = \boldsymbol{\phi_s}^T \Sigma\, \boldsymbol{\phi_s} = \boldsymbol{\phi_s}^T Q^{-1} \boldsymbol{\phi_s} = \boldsymbol{\phi_s}^T (BB^T)^{-1} \boldsymbol{\phi_s}$$

- Default method is **exact**: first computes sparse Cholesky $Q = DD^T$, then solves $D\mathbf{v} = \boldsymbol{\phi_s}$, whence

$$\mathrm{Var}(g(\mathbf{s})) = \|\mathbf{v}\|_2^2 = \sum_i^R v_i^2.$$

Two Fast Methods

- FFT-based method is **approximate**: calculates variance on coarse grid $n \geq (2r-1)^2$ where $R = r \times r$, then performs 2D-FFT upsampling to a finer grid N.
- Reduces complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(n^3 + N\log(N))$, where $n < N$.



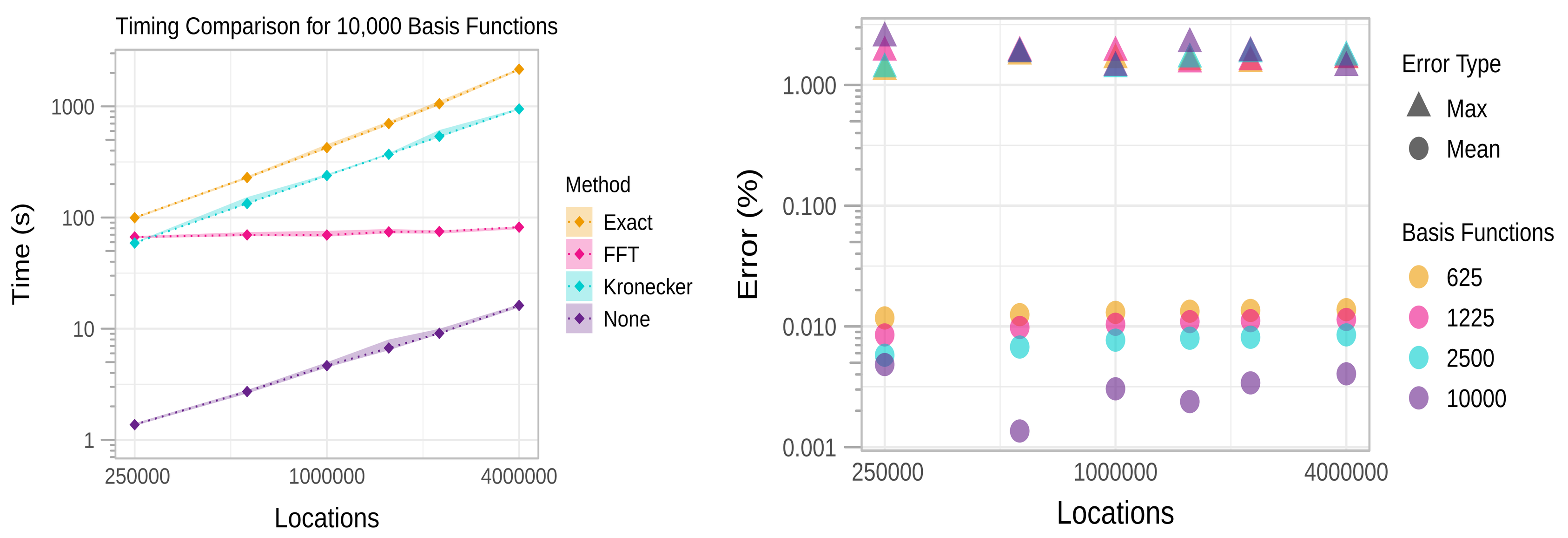
- Kronecker-based method is **exact**: further decomposes B using Kronecker products and discretizes across two-dimensions:

$$B = (A \otimes I_r + I_r \otimes A) = (UDU^T \otimes I_r + I_r \otimes UDU^T) = \mathcal{UDU}^T$$

  where U is small, D is diagonal, and $B^{-1} = \mathcal{UD}^{-1}\mathcal{U}^T$
- Reduces complexity to $\mathcal{O}(NR^{3/2})$.

Timing & Error

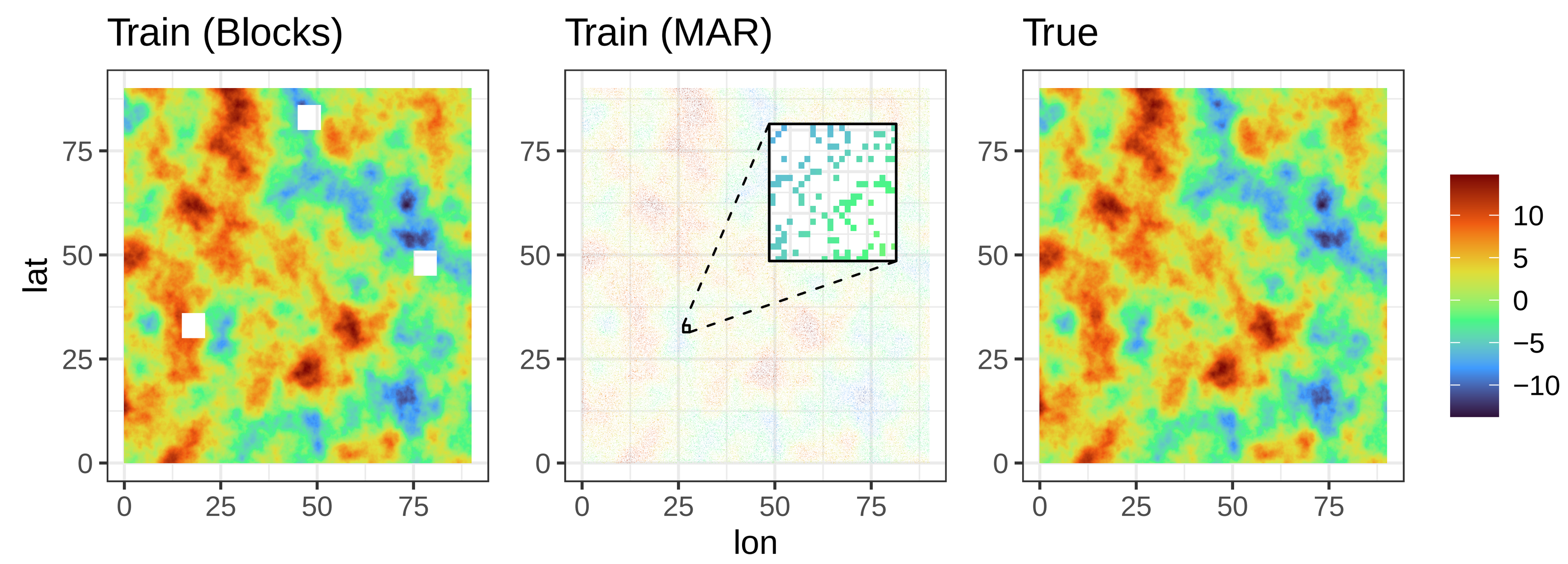
Basis function calculation timed for varying spatial locations N.



- Error calculated by comparing FFT result (approximation) to the exact variance at each grid location $\mathbf{s}$.

Big Data Prediction Example

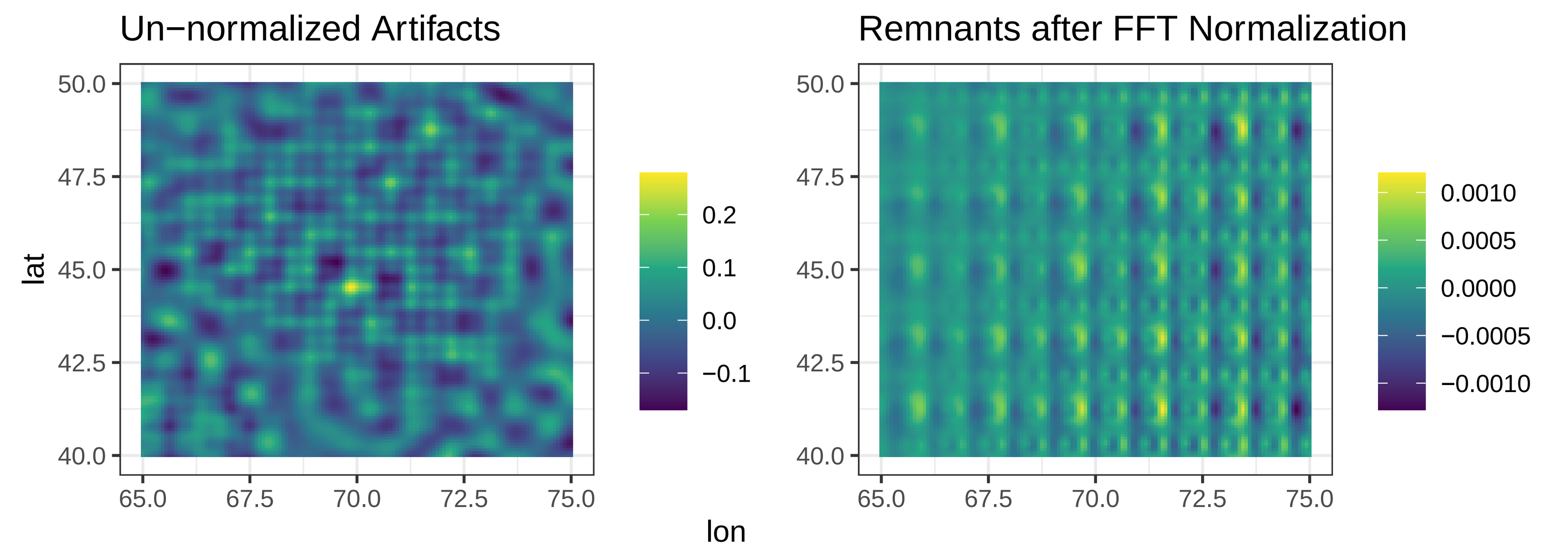
GP with Matern covariance and smoothness $\nu = 1$ simulated on a 1153×1153, $(N = 1,329,409)$ grid. Training cases include missing 80% of the data at random (MAR) and missing three 100×100 pixel regions (Blocks).



Method	Blocks			MAR		
	MAE ↓	RMSPE ↓	Time (min)	MAE ↓	RMSPE ↓	Time (min)
None	1.1672	1.6432	28.68	0.1983	0.2486	16.57
*FFT	1.0527	1.4735	78.82	0.2051	0.2570	45.08
Kronecker	1.0508	1.4712	111.49	0.2051	0.2570	57.90
Default	1.0508	1.4712	173.75	0.2051	0.2570	95.51

Table 1. 4 layers of basis functions used ($R = 65,844$ total basis functions). *Kronecker is used at the layer with the finest resolution.

Reducing (or Removing) Artifacts



FFT normalization removes 99% of artifacts, exact methods remove the artifacts completely.

Summary & Future Work

Main Contributions:

- We show that accelerated and approximate normalization methods are feasible in practice.
- We provide software implementation of two such methods.

Future research directions could explore:

- More advanced image upsampling algorithms.
- Faster solvers of discretizations.
- Other basis function forms more amenable to normalization.
- Adaptation to irregular data with fast and local interpolation.

Source Code & Article

All code for figures, timing, and big data experiments, along with a link to the `arXiv` preprint of the paper, can be found at this `Github` repository.

References

[1] N. Cressie, M. Sainsbury-Dale, and A. Zammit-Mangion. Basis-function models in spatial statistics. *Annual Review of Statistics and Its Application*, 9:373–400, 2022.

[2] D. Nychka et al. A multiresolution gaussian process model for the analysis of large spatial datasets. *Journal of Computational and Graphical Statistics*, 24(2):579–599, 2015.

[3] D. Nychka, D. Hammerling, S. Sain, N. Lenssen, and M. D. Nychka. Package 'latticekrig', 2019.