

Robust Distributed Learning of Functional Data From Simulators through Data Sketching

Jacob Andros and Dr. Rajarshi Guhaniyogi
Stat Cafe 2024

Background

Suppose we fit a Gaussian process model to some dataset of interest:

$$\begin{aligned}\mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{w} + \boldsymbol{\epsilon} \\ \mathbf{w} &\sim \text{GP}(\mathbf{0}, \mathbf{C}_\theta) \\ \boldsymbol{\epsilon} &\sim \text{N}(\mathbf{0}, \tau^2 \mathbf{I})\end{aligned}\tag{1}$$

To carry out posterior inference with n_{iter} MCMC iterates, we will need to invert \mathbf{C}_θ repeatedly (n_{iter} times). This becomes costly as the sample size increases, since the order of inversion is $\mathcal{O}(n^3)$ and the storage cost is $\mathcal{O}(n^2)$.

How can we address the computational cost associated with inverting the covariance matrix at each iteration?

1. Divide and conquer: Split \mathbf{Y} and \mathbf{X} into K distinct subsets, and fit the model on each subset independently.
2. Random compression: Multiply \mathbf{Y} and \mathbf{X} by a random $m \times n$ matrix Φ to create a condensed “sketch” of the original data with only m rows.

Both of these are forms of distributed learning. Study of distributed statistical methods has gained substantial attention in the recent years (Guhaniyogi and Banerjee, 2019; McMahan et al., 2017).

What if the data is confidential or privacy-protected?

1. Divide and conquer: Since the data is split into different shards, the entirety of the dataset is not accessible from one place or file, reducing the risk of leaked information.
2. Random compression: Transforms the data in such a way that it is not possible to recover the original \mathbf{Y} and \mathbf{X} .

Motivating Case Study: Flood Data

New Jersey Flood Data

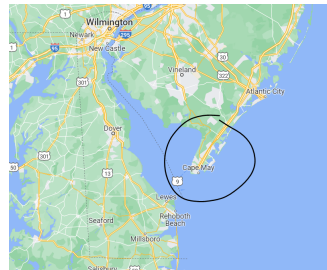
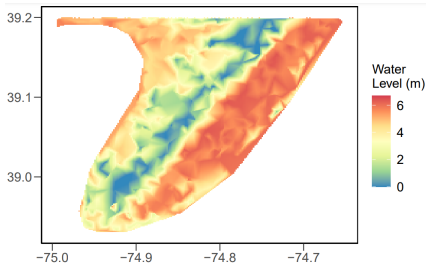


Figure 1: Flood level analysis on southern NJ peninsula coastline, first attempted by Hutchings et al., 2023.

- The number of coordinate locations is very high (nearly 50 thousand).
- The locations of the coordinates, some of which contain power stations, are confidential.
- We wish to incorporate both fixed effects (location-specific predictors) and random effects (storm-specific predictors) into the model.

*** Random compression can help us solve the first two issues here, but we still need a way to incorporate both fixed and random effects (AKA local and global attributes).

Incorporate Storm-Specific Predictors

Normally, for local predictors only, our likelihood function for the full Gaussian process model would be something like:

$$N(\mathbf{y} | \mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{K}(\theta) + \tau^2 \mathbf{I}_n), \quad (2)$$

But now, we formulate it as the product of likelihoods as follows:

$$\prod_{s=1}^S N(\mathbf{y}_s | (\mathbf{1}_n \otimes \mathbf{z}_s^T) \boldsymbol{\gamma} + \mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{K}(\theta) + \tau^2 \mathbf{I}_n), \quad (3)$$

Where \mathbf{y}_s is the water level for all locations resulting from storm s , and \mathbf{z}_s are the storm-specific predictors.

Methodology

Model Specification

Priors:

$$\begin{aligned}\sigma^2 &\sim \text{IG}(1, 1) \\ \tau^2 &\sim \text{IG}(1, 1) \\ \boldsymbol{\beta} &\sim \text{N}(0, I_p) \\ \theta &\text{ follows a discrete prior.}\end{aligned}\tag{4}$$

Likelihood:

$$\prod_{s=1}^S N\left(\boldsymbol{\Phi} \mathbf{y}_s \mid (\mathbf{1}_m \otimes \mathbf{z}_s^T) \boldsymbol{\gamma} + \boldsymbol{\Phi} \mathbf{X} \boldsymbol{\beta}, \boldsymbol{\Phi} \left[\sigma^2 \mathbf{K}(\theta) + \tau^2 \mathbf{I}_n \right] \boldsymbol{\Phi}^\top\right), \tag{5}$$

Where $\boldsymbol{\Phi}$ is an $m \times n$ random compression matrix, generated from $\Phi_{ij} \sim \text{N}(0, \frac{1}{n})$, and $m \ll n$.

Sampling Procedure

- Sample σ^2 and τ^2 through Metropolis-Hastings. Since θ_h is kept fixed throughout the analysis of Π_h , we need to compute $\Phi^T \mathbf{K}(\theta_h) \Phi$ only once, which leads to substantial computational benefit.
- Sample $(\beta^T, \gamma^T)^T | - \sim N(\boldsymbol{\mu}_{\beta, \gamma}, \boldsymbol{\Sigma}_{\beta, \gamma})$, where $\boldsymbol{\Sigma}_{\beta, \gamma} = \left\{ (m/n) \sum_{s=1}^S \mathbf{A}_s^T \boldsymbol{\Sigma}^{-1} \mathbf{A}_s + \mathbf{I}_p \right\}^{-1}$, $\boldsymbol{\mu}_{\beta, \gamma} = \boldsymbol{\Sigma}_{\beta, \gamma} \left\{ (m/n) \sum_{s=1}^S \mathbf{A}_s^T \boldsymbol{\Sigma}^{-1} \mathbf{y}_{s, \Phi_h} \right\}$. Here $\mathbf{A}_s = [\mathbf{X}_{\Phi_h} : \mathbf{Z}_{s, \Phi_h}]$ is an $m \times n$ matrix and $\boldsymbol{\Sigma} = (\Phi_h \mathbf{K}(\theta_h) \Phi_h^T + \tau^2 \mathbf{I}_m)$ is an $m \times m$ matrix. This step incurs a computation complexity of $O(m^3)$, since $\boldsymbol{\Sigma}$ is an $m \times m$ covariance matrix that needs to be inverted.

Distributed Learning

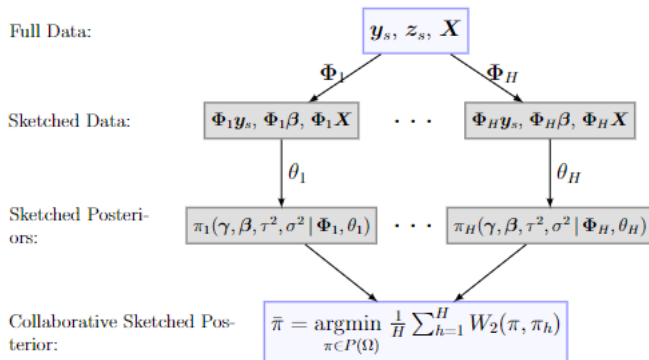


Figure 2: The Wasserstein mean averages the H different posterior distributions obtained (Guhaniyogi et al., 2023).

We leverage the notion of the Wasserstein barycenter (Srivastava et al., 2018) to aggregate parameter estimates and predictions from the H different model fits.

$$\hat{\alpha}_{\xi} = \frac{1}{H} \sum_{h=1}^H \hat{\alpha}_{\xi,h}, \quad (6)$$

Distributed Storage

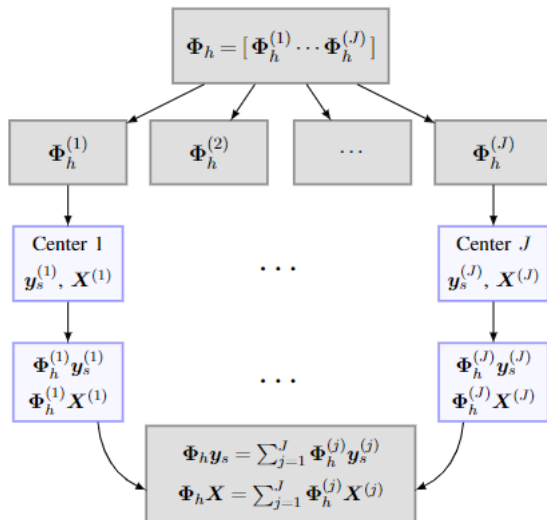


Figure 3: Distributed storage of data for J separate storage centers.

Results (Simulations)

Simulation Study - Parameters

Model	Method	σ^2	τ^2	β_1	β_2
Truth		2.00	0.20	2.00	-1.00
Full GP	DISK-Sub	2.13 (2.09, 2.16)	0.21 (0.21, 0.22)	1.77 (1.39, 2.14)	-1.00 (-1.01, -0.98)
	DISK-Str	1.94 (1.91, 1.96)	0.20 (0.19, 0.21)	1.99 (1.89, 2.08)	-1.02 (-1.02, -0.98)
	Sketching	2.07 (2.03, 2.10)	0.21 (0.19, 0.22)	2.00 (1.98, 2.01)	-0.99 (-1.00, -0.97)
MPP	DISK-Sub	1.62 (1.59, 1.65)	0.18 (0.17, 0.20)	1.79 (1.55, 2.03)	-1.00 (-1.02, -0.98)
	DISK-Str	1.80 (1.79, 1.82)	0.05 (0.04, 0.05)	1.94 (1.86, 2.01)	-0.99 (-1.02, -0.97)
	Sketching	2.07 (1.99, 2.16)	0.11 (0.10, 0.13)	2.01 (2.00, 2.03)	-0.95 (-0.96, -0.95)
NNGP	DISK-Sub	2.14 (2.10, 2.18)	0.21 (0.21, 0.22)	1.76 (1.37, 2.14)	-1.00 (-1.01, -0.98)
	DISK-Str	1.99 (1.97, 2.01)	0.21 (0.20, 0.21)	1.99 (1.90, 2.08)	-1.00 (-1.01, -0.98)
	Sketching	2.04 (2.00, 2.07)	0.19 (0.18, 0.20)	1.97 (1.95, 1.99)	-0.98 (-0.99, -0.97)

Table 1: We calculate the posterior median with 95% confidence intervals for all model parameters for all the distributed Bayesian competitors. We set both the sketching dimension for our approach and the size of each subset for the DISK approach to be $m = 500$ to ensure comparability.

Simulation Study - Predictions

Model	Competitors	MSPE	Coverage	Interval Score	Energy Score
Full GP	DISK-Subdomain	2.44	0.95	7.02	0.89
	DISK-Stratified	2.30	0.94	6.89	0.86
	Sketching	2.27	0.95	6.87	0.86
MPP	DISK-Subdomain	2.50	0.91	7.24	0.90
	DISK-Stratified	2.32	0.92	7.05	0.89
	Sketching	2.30	0.95	6.87	0.86
NNGP	DISK-Subdomains	2.43	0.95	7.00	0.89
	DISK-Stratified	2.29	0.94	6.88	0.86
	Sketching	2.28	0.95	6.88	0.86

Table 2: MSPE, coverage, interval score, and energy score for all competing methods.

Interval Score: Favors model with the smallest possible prediction intervals that still contain the true data (Francom and Sansó, 2020).

Energy Score: Takes into account both predictive accuracy as well as predictive uncertainty (Heaton et al., 2019).

Additional - How to choose m ?

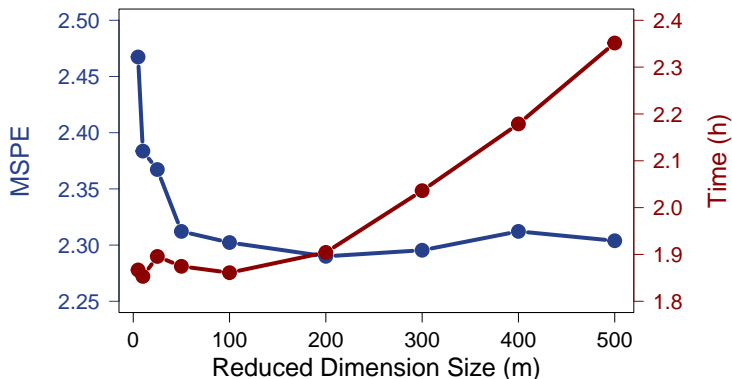


Figure 4: MSPE and computation time (in hours) for model fit and prediction in our simulation data, as a function of the compressed dimension size m .

Results (Flood Analysis)

Predicted Water Levels

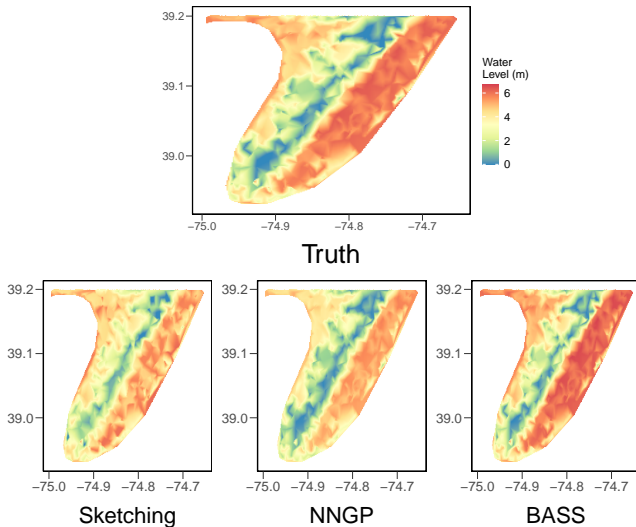


Figure 5: Actual water level (in meters) for a randomly selected storm at each coordinate in the testing dataset, along with the predicted water level under each model.

	MSPE	Error %	Coverage	Interval Score	Energy score
Sketching	1.07	0.06	0.88	5.60	0.63
NNGP-ind	0.83	0.09	0.18	20.45	1.52
BASS	1.45	0.10	0.59	11.23	1.02

Table 3: Predictive diagnostics for the storm surge analysis. For interval score and Energy score, lower values indicate better scores.

Posterior Densities of Parameters

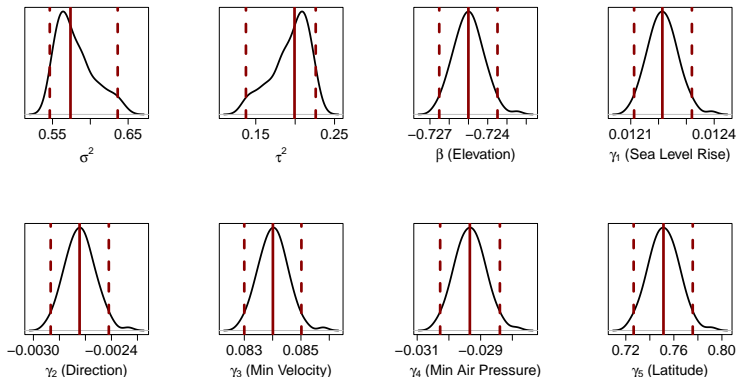








Figure 6: Posterior densities, means, and credible intervals for each parameter in the flood model.

Conclusion

Distributed inference for Gaussian processes using random compression proves to be:

- Capable of inference for both local and global effect parameters.
- Free of the sensitivity associated with partitioning a dataset.
- Resilient even in settings where relatively few simulations can be obtained.
- A viable solution to substantially reducing both the computation cost and storage cost in fitting GP models to massive datasets.

References

-  Francom, D., & Sansó, B. (2020). **BASS: An R package for fitting and performing sensitivity analysis of Bayesian adaptive spline surfaces.** *Journal of Statistical Software*, 94(LA-UR-20-23587).
-  Guhaniyogi, R., & Banerjee, S. (2019). **Multivariate spatial meta kriging.** *Statistics & probability letters*, 144, 3–8.
-  Guhaniyogi, R., Li, C., Savitsky, T., & Srivastava, S. (2023). **Distributed bayesian inference in massive spatial data.** *Statistical science*, 38(2), 262–284.
-  Heaton, M. J., Datta, A., Finley, A. O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., et al. (2019). **A case study competition among methods for analyzing large spatial data.** *Journal of Agricultural, Biological and Environmental Statistics*, 24, 398–425.
-  Hutchings, G., Sansó, B., Gattiker, J., Francom, D., & Pasqualini, D. (2023). **Comparing emulation methods for a high-resolution storm surge model.** *Environmetrics*, 34(3), e2796.
-  McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A.