Stochastic gradient MCMC for massive geostatistical data

Mohamed A Abba[1], Brian J Reich[1], **Reetam Majumder**[2], Brandon Feng[1]

[1]North Carolina State University, [2]University of Arkansas

CMStatistics 2025

Dec 15, 2025

UNIVERSITY OF
ARKANSAS

Problem setup

- GPs are the workhorse of spatial statistics (Gelfand and Schliep, 2016)
- Let Y_i for $i \in \{1, ..., n\}$ be the observation at a spatial location $\mathbf{s}_i = (s_{i1}, s_{i2})$, and let $\mathbf{X}_i = (X_{i1}, ..., X_{ip})$ be covariates. Our model is:

$$Y_i = \mathbf{X}_i\boldsymbol{\beta} + Z(\mathbf{s}_i) + \varepsilon_i,$$

- Fixed effects $\boldsymbol{\beta}$ and measurement error $\varepsilon_i \overset{iid}{\sim} \text{Normal}(0, \tau^2)$
- $Z(\mathbf{s})$ is an isotropic spatial Gaussian process with mean $\text{E}\{Z(\mathbf{s})\} = 0$, spatial variance $\text{Var}\{Z(\mathbf{s})\} = \sigma^2$ and spatial correlation $\text{Cor}\{Z(\mathbf{s}_i), Z(\mathbf{s}_j)\} = \text{K}(d_{ij})$ for distance $d_{ij} = ||\mathbf{s}_i - \mathbf{s}_j||$.
- Matérn (Stein, 1999) correlation function with range ρ and smoothness ν:

$$\text{K}(d) = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{d}{\rho}\right)^{\nu} \mathcal{K}_{\nu}\left(\frac{d}{\rho}\right).$$

- Let $\boldsymbol{\theta} = (\sigma^2, \rho, \nu, \tau^2)$ be the collection of covariance parameters.

Problem setup

- The marginal distribution (over Z) of $\mathbf{Y}$ is multivariate normal with mean $\mathbb{E}[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta}$ and covariance matrix $[(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}} \mid \boldsymbol{\theta}] = \Sigma(\boldsymbol{\theta})$ with

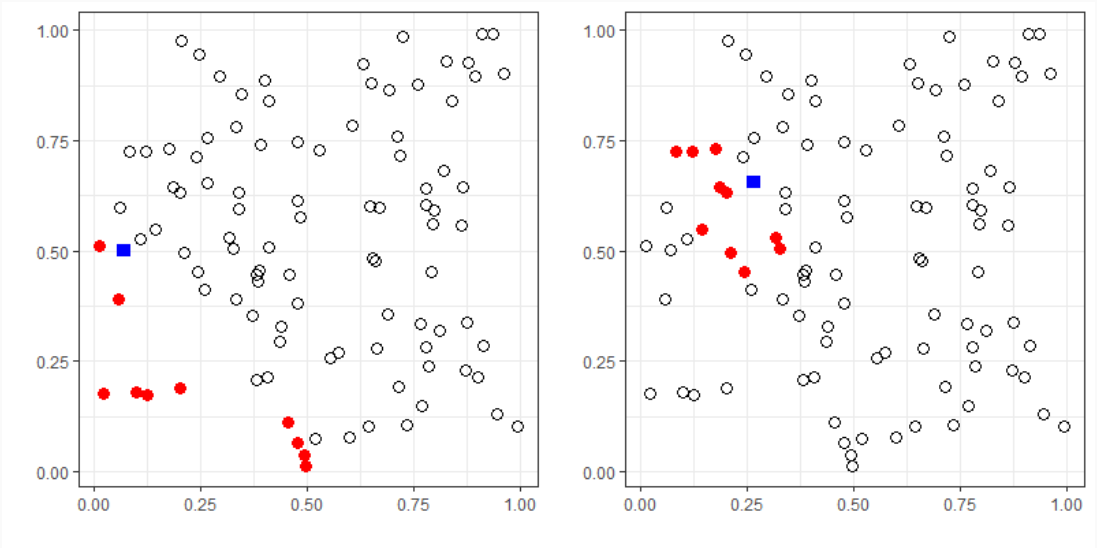
$$\Sigma(\boldsymbol{\theta}) = \sigma^2\mathbf{K} + \tau^2\mathbf{I}_n,$$
$$\mathbf{K}_{i,j} = \mathrm{K}(d_{ij}).$$

- The full log-likelihood then becomes

$$\ell_{\mathrm{full}}(\boldsymbol{\beta}, \boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det\Sigma(\boldsymbol{\theta}) - \frac{1}{2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}\Sigma(\boldsymbol{\theta})^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}).$$

- Evaluating ℓ_{full} involves computing the determinant and inverse of $\Sigma(\boldsymbol{\theta})$ which generally requires $O(n^3)$ operations.
- This cost becomes prohibitive for large spatial datasets ($\approx n > 10^4$).

Vecchia approximation



- Approximate $\mathbf{Y}$ via a Vecchia approximation (Vecchia, 1988):

$$f(Y(\mathbf{s}_1), ..., Y(\mathbf{s}_n)) = \prod_{i=1}^{n} f(Y(\mathbf{s}_i)|Y(\mathbf{s}_1), ..., Y(\mathbf{s}_{i-1})) \approx \prod_{i=1}^{n} f_i(Y(\mathbf{s}_i)|Y(\mathbf{s}_{(i)})),$$

  for $Y(\mathbf{s}_{(i)}) = \{Y(\mathbf{s}_j); j \in \mathcal{N}_i\}$, conditioning set $\mathcal{N}_i \subseteq \{1, ..., i-1\}$, with $|\mathcal{N}_i| \leq m$

- Valid joint PDF permits standard Bayesian analysis and interpretation

Motivation

- The Vecchia approximation reduces the complexity cost from $O(n^3)$ for the full likelihood to $O(nm^3)$
- When n is quite large, and we need to do MCMC, this is still substantial
- One option is subsampling:
  - Saha and Bradley (2023) propose an efficient composite sampling scheme
  - Heaton and Johnson (2023) use minibatches to approximate the complete conditional of conjugate parameters

We want:

1. Fast inference and predictions for $n \in \{10^4, 10^5, 10^6\}$
2. Precision quantification

Our approach: Stochastic gradient MCMC (Ma et al., 2015)

- Stochastic nature makes it faster than regular MCMC
- Second order gradient information leads to better exploration of parameter space and faster convergence
- MCMC gives uncertainty unlike likelihood-based methods (Guinness, 2019)

Unbiasedness of the stochastic gradient

- We can express the posterior $p(\boldsymbol{\beta}, \boldsymbol{\theta} \mid \mathbf{Y})$ as:

$$\log p(\boldsymbol{\beta}, \boldsymbol{\theta} \mid \mathbf{Y}) = \ell(\boldsymbol{\beta}, \boldsymbol{\theta}) + \log p(\boldsymbol{\beta}, \boldsymbol{\theta}), \tag{1}$$

$$\ell(\boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \log f(Y_i \mid Y_{(i)}, \boldsymbol{\beta}, \boldsymbol{\theta}).$$

- The log-likelihood and log-posterior of the parameters $\phi := \{\boldsymbol{\beta}, \boldsymbol{\theta}\}$ can consequently be written as a sum of conditional normal log-densities, where the conditioning set is at most of size m.
- Let $\mathcal{B} \subset \{1, \ldots, n\}$ be a minibatch index set of size $n_{\mathcal{B}}$, and let

$$\bar{\ell}_{\mathcal{B}}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \frac{n}{n_{\mathcal{B}}} \sum_{i \in \mathcal{B}} \log f(Y_i \mid Y_{(i)}, \boldsymbol{\beta}, \boldsymbol{\theta}).$$

- **A key result which will become important soon:**
  $\nabla \bar{\ell}_{\mathcal{B}}$ is a stochastic gradient, which is an unbiased estimator of $\nabla \ell(\boldsymbol{\beta}, \boldsymbol{\theta})$.

Unbiasedness of the stochastic gradient

- We can construct an unbiased estimate of the gradient of the Vecchia log-posterior based on a minibatch of the data:

$$\bar{g}_{\mathcal{B}}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \nabla \bar{\ell}_{\mathcal{B}}(\boldsymbol{\beta}, \boldsymbol{\theta}) + \nabla \log p(\boldsymbol{\beta}, \boldsymbol{\theta}),$$

- Reduces the cost of learning iterations to $O(m^3 n_{\mathcal{B}})$.

SGMCMC simulates continuous dynamics of a potential energy, viz., $-\log p(\boldsymbol{\beta}, \boldsymbol{\theta} \mid \mathbf{Y})$, in a manner that generates samples from the posterior distribution.

Langevin Monte Carlo

- Let $\phi = (\beta^{\mathrm{T}}, \theta^{\mathrm{T}})^{\mathrm{T}}$ be the vector of parameters.
- The Langevin diffusion over $\log p(\phi \mid \mathbf{Y})$ is given by the stochastic differential equation

$$d(\phi_t) = \nabla \log p(\phi_t \mid \mathbf{Y}) dt + \sqrt{2} dW_t,$$

  where dW_t is Brownian motion, t represents time.
- The distribution of samples ϕ_t converges to the true posterior as $t \to \infty$ (Roberts and Rosenthal, 1998)
- Since simulating a continuous time process is infeasible in practice, we use the Euler discretization method to approximate the Langevin dynamics:

$$\phi_{t+1} = \phi_t + h_t \nabla \log p(\phi_t \mid \mathbf{Y}) + \sqrt{2h_t} e_t,$$

  where h_t is the step size, e_t is random white noise.
- This recursive sampling algorithm is known as the Langevin Monte Carlo

Stochastic gradient Langevin dynamics

- Evaluating the gradient of $\log p(\phi_t \mid \mathbf{Y})$ is a bottleneck for large n
- SGLD replaces $\nabla \log p(\phi_t \mid \mathbf{Y})$ with an unbiased estimate, viz., $\bar{g}_{\mathcal{B}}(\phi)$

$$\phi_{t+1} = \phi_t + h_t \bar{g}_{\mathcal{B}}(\phi_t) + \sqrt{2h_t} e_t,$$

  for positive step sizes h_t that satisfy the Robbins-Monro conditions.

Why do we need unbiasedness?

- Most (all?) SGD methods require unbiasedness to guarantee convergence
- Arbitrary subsamples will *not* give an unbiased gradient estimate, but the Vecchia approximation does.

This idea works for other stochastic gradient methods too; e.g., preconditioned SGLD (pSGLD; Li et al., 2016), ADAMSGLD (Kim et al., 2022), momentum SGLD (MSGLD).

Stochastic gradient Reimannian Langevin dynamics

- SGLD updates all parameters using the same step size, which can cause slow mixing when different parameters have different curvature or scales
- SGRLD accounts for curvature and scale by using a Riemannian metric $G(\phi)$ as a preconditioner (Nemeth and Fearnhead, 2021)
- It takes steps with the steepest ascent on the manifold defined by $G(\phi_t)$:

$$\phi_{t+1} = \phi_t + h_t \left(G^{-1}(\phi_t)\bar{g}_{\mathcal{B}}(\phi_t) + \Gamma(\phi_t) \right) + \sqrt{2h_t} G^{-1/2}(\phi_t) e_t,$$

$$\Gamma(\phi_t)_i = \sum_j \frac{\partial G(\phi_t)^{-1}_{ij}}{\partial \phi_{tj}}.$$

- $\Gamma(\phi_t)$ represents a drift term (that vanishes in SGLD)
- Commonly choices for $G(\phi)$ include the Fisher information matrix and estimates of the Hessian of the log-posterior.
- The **Vecchia approximation makes the Fisher Information matrix tractable** to evaluate (Guinness, 2019; Guinness et al., 2018)

Application: Argo ocean temperature data

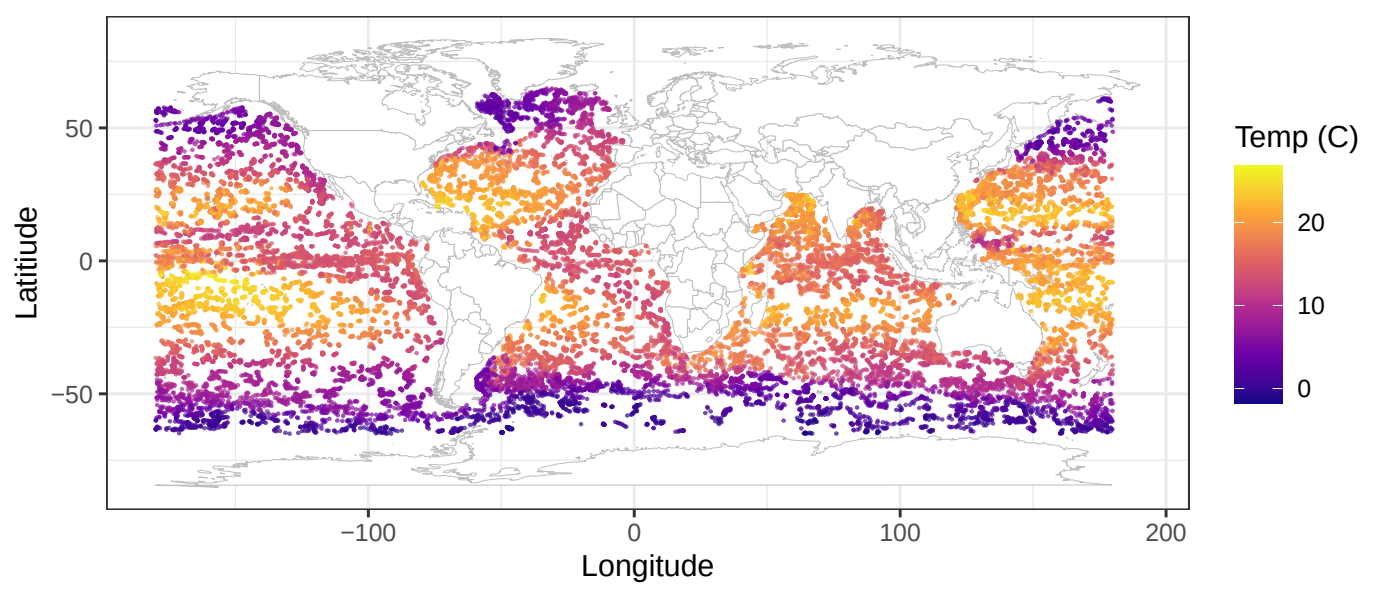


Figure 1: Argo ocean temperature measurements at a depth of 100 meters.

- $n = 32,436$ observations taken on buoys in the spring of 2016, measuring ocean temperature (C) at depths of roughly 100 meters.
- The mean function is taken to be quadratic in latitude and longitude.

Priors and hyperparameters

- Priors for the covariance parameters:

$$\rho \sim \text{Gamma}(9.0, 2.0)$$
$$\nu \sim \text{Log-Normal}(1.0, 1.0)$$
$$\tau^2, \sigma^2 \sim \text{Gamma}(0.1, 0.1)$$

- Flat priors for the regression coefficients
- 80/20 train-test split
- $n_{\mathcal{B}} \in \{100, 250, 500\}$
- $m \in \{10, 15, 30\}$
- Nearest neighbor Gaussian process (NNGP; Finley et al., 2019) used for comparisons
- $8,000$ iterations for NNGP, $40,000$ for SGRLD (400 epochs for $n_{\mathcal{B}} = 100$)

MCMC Convergence

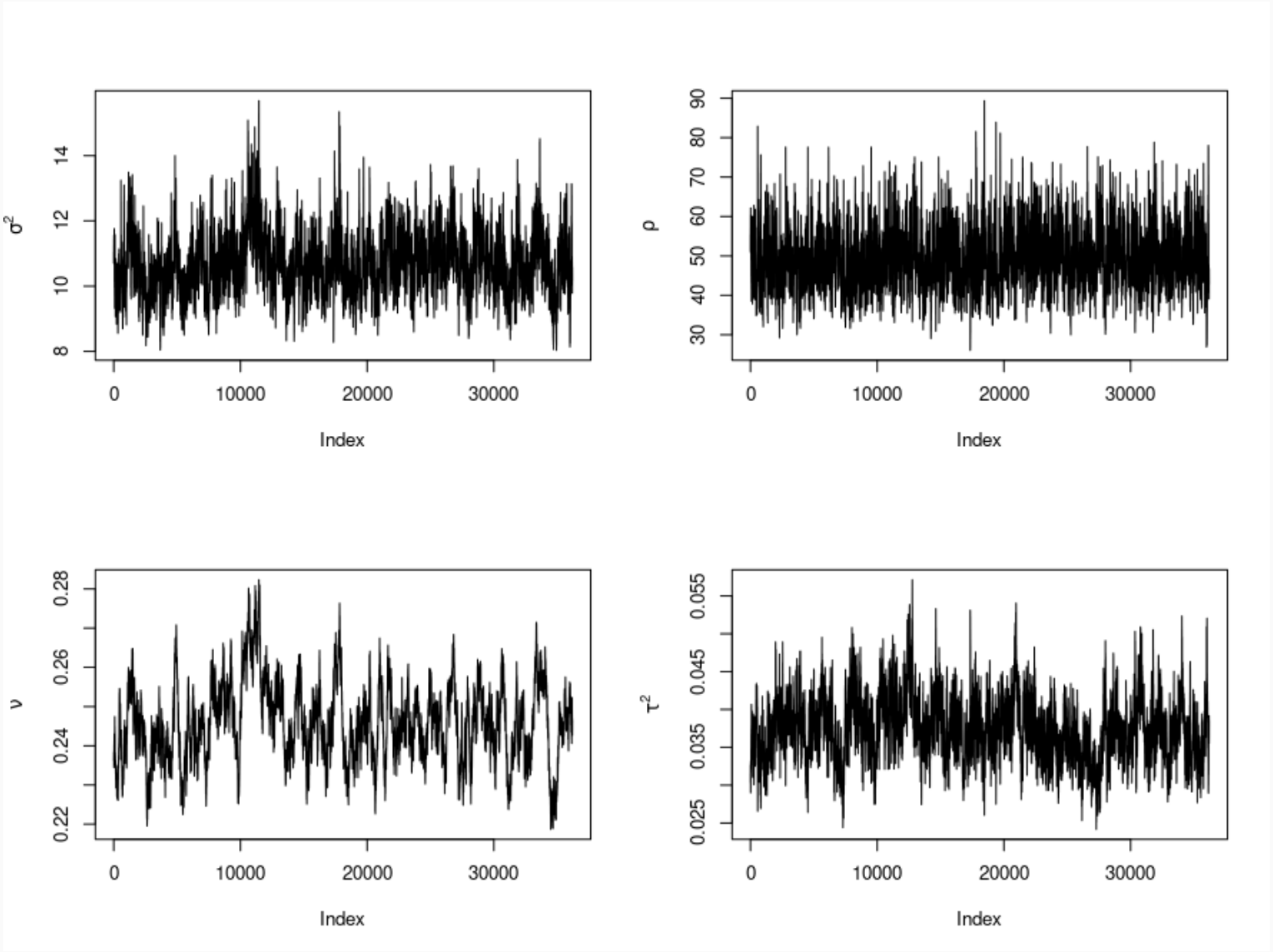


Figure 2: Evolution of SGRLD sampling from the posterior distribution of the covariance parameters.

Inference

Method	Parameter	Posterior mean	95% CI	ESS/min
NNGP	σ^2	6.72	$(6.32, 7.08)$	0.17
	ρ	0.10	$(0.10, 0.11)$	3.42
	ν	0.33	$(0.32, 0.34)$	0.04
	τ^2	0.08	$(0.08, 0.09)$	0.08
SGRLD	σ^2	10.64	$(7.41, 13.57)$	52.21
	ρ	48.93	$(22.94, 68.46)$	115.41
	ν	0.25	$(0.23, 0.27)$	18.68
	τ^2	0.04	$(0.03, 0.05)$	39.13

Table 1: Posterior mean, 95% credible intervals and effective sample size per minute for all the covariance parameters.

	MSE	Coverage	R^2	Time (in minutes)
NNGP	6.41	0.88	0.89	218.55
SGRLD	1.47	0.93	0.94	7.01

Table 2: Prediction MSE, R^2, and coverage rate of the 95% predictive credible intervals on the test set and the correlation between the predicted temperatures and true observed values. The last column gives the total training time.

Sensitivity study

$n_{\mathcal{B}}$	m	σ^2	ρ	ν	τ^2
100	10	$10.18_{(8.79,11.89)}$	$53.67_{(41.11,66.87)}$	$0.24_{(0.22,0.25)}$	$0.04_{(0.03,0.04)}$
	15	$11.29_{(9.08,13.53)}$	$54.95_{(39.18,72.83)}$	$0.25_{(0.22,0.27)}$	$0.04_{(0.03,0.04)}$
	30	$9.60_{(5.09,13.24)}$	$46.41_{(13.34,74.52)}$	$0.24_{(0.20,0.26)}$	$0.04_{(0.03,0.07)}$
250	10	$10.52_{(9.08,12.25)}$	$49.55_{(37.79,62.85)}$	$0.25_{(0.22,0.26)}$	$0.04_{(0.03,0.05)}$
	15	$10.64_{(7.41,13.57)}$	$48.93_{(22.94,68.46)}$	$0.25_{(0.23,0.27)}$	$0.04_{(0.03,0.05)}$
	30	$10.59_{(7.41,13.57)}$	$46.08_{(22.95,69.46)}$	$0.25_{(0.23,0.27)}$	$0.04_{(0.03,0.05)}$
500	10	$11.02_{(9.74,12.69)}$	$49.23_{(39.56,60.36)}$	$0.25_{(0.23,0.27)}$	$0.04_{(0.03,0.04)}$
	15	$11.41_{(9.75,13.20)}$	$48.42_{(37.98,60.26)}$	$0.25_{(0.24,0.27)}$	$0.04_{(0.03,0.04)}$
	30	$11.52_{(9.72,13.61)}$	$48.39_{(36.61,62.35)}$	$0.26_{(0.24,0.28)}$	$0.04_{(0.03,0.04)}$

Table 3: Sensitivity analysis to the choice of the conditioning set size m and the mini-batch size $n_{\mathcal{B}}$. Posterior mean and 95% credible intervals are displayed for each combination of $n_{\mathcal{B}}$ and m.

Summary

- SG methods offer considerable speed-ups when the data size is very large
- By leveraging the form of the Vecchia approximation, we derive unbiased gradient estimates based on minibatches of the data.
- We developed a new stochastic gradient based MCMC algorithm for scalable Bayesian inference in large spatial data settings.
- Exact Fisher information used to speed up convergence and explore the parameter space efficiently
- Can be extended to non Gaussian models *i.e.* classification.
- ArXiv preprint link (left) and GitHub repo link (right).


References

Finley, A. O., Datta, A., Cook, B. D., Morton, D. C., Andersen, H. E., and Banerjee, S. (2019). Efficient algorithms for Bayesian nearest neighbor Gaussian processes. *Journal of Computational and Graphical Statistics*, 28(2):401–414.

Gelfand, A. E. and Schliep, E. M. (2016). Spatial statistics and Gaussian processes: A beautiful marriage. *Spatial Statistics*, 18:86–104. Spatial Statistics Avignon: Emerging Patterns.

Guinness, J. (2019). Gaussian process learning via fisher scoring of vecchia's approximation.

Guinness, J., Katzfuss, M., and Fahmy, Y. (2018). Gpgp: fast Gaussian process computation using Vecchia's approximation. *R package version 0.1. 0*.

Heaton, M. J. and Johnson, J. A. (2023). Minibatch Markov chain Monte Carlo algorithms for fitting Gaussian processes. *arXiv preprint arXiv:2310.17766*.

Kim, S., Song, Q., and Liang, F. (2022). Stochastic gradient Langevin dynamics with adaptive drifts. *Journal of statistical computation and simulation*, 92(2):318–336.

Li, C., Chen, C., Carlson, D., and Carin, L. (2016). Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Ma, Y., Ma, Y.-A., Chen, T., and Fox, E. B. (2015). A complete recipe for stochastic gradient MCMC. In *Neural Information Processing Systems*.

References ii

Nemeth, C. and Fearnhead, P. (2021). Stochastic gradient Markov chain Monte Carlo. *Journal of the American Statistical Association*, 116(533):433–450.

Roberts, G. O. and Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60.

Saha, S. and Bradley, J. R. (2023). Incorporating subsampling into Bayesian models for high-dimensional spatial data. *arXiv preprint arXiv:2305.13221*.

Stein, M. L. (1999). *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media.

Vecchia, A. V. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):297–312.

Simulation results

Table 4: Mean squared error (Monte Carlo standard errors) of covariance parameters computed using 100 simulations, each having sample size N. The proposed SGRLD method compared with other SGMCMC methods (pSGLD, ADAMSGLD, MSGLD) and the full likelihood NNGP method.

N	Algorithm	Variance (σ^2)	Range (ρ)	Smoothness (ν)	Nugget (τ^2)
10^4	pSGLD	0.074(0.013)	0.039(0.008)	0.103(0.017)	$0.002(4 \cdot 10^{-4})$
	ADAMSGLD	0.075(0.017)	0.036(0.008)	0.129(0.023)	$0.002(6 \cdot 10^{-4})$
	MSGLD	0.066(0.014)	0.034(0.008)	0.108(0.0196)	$0.002(6 \cdot 10^{-4})$
	NNGP	0.414(0.131)	0.095(0.071)	0.162(0.106)	$0.093(2.4 \cdot 10^{-2})$
	SGRLD	0.056(0.016)	0.031(0.006)	0.077(0.013)	$0.001(10^{-4})$
10^5	pSGLD	0.008(0.001)	0.002(0.0003)	0.011(0.0019)	$1 \cdot 10^{-4}(2 \cdot 10^{-5})$
	ADAMSGLD	0.014(0.005)	0.008(0.002)	0.031(0.008)	$1 \cdot 10^{-4}(2 \cdot 10^{-4})$
	MSGLD	0.017(0.001)	$0.003(5 \cdot 10^{-4)}$	0.019(0.002)	$2 \cdot 10^{-4}(4 \cdot 10^{-5})$
	NNGP	0.116(0.030)	0.024(0.01)	0.118(0.08)	$4 \cdot 10^{-2}(0.01)$
	SGRLD	$0.005(8 \cdot 10^{-4})$	$0.001(1.0 \cdot 10^{-4})$	$0.008(1.8 \cdot 10^{-3})$	$10^{-4}(2 \cdot 10^{-}5)$
10^6	pSGLD	0.003(0.001)	0.003(0.0008)	0.002(0.0014)	$3.1 \cdot 10^{-4}(6 \cdot 10^{-5})$
	ADAMSGLD	0.009(0.002)	0.006(0.002)	0.026(0.007)	$2 \cdot 10^{-4}(9 \cdot 10^{-5})$
	MSGLD	$0.011(1.8 \cdot 10^{-3})$	$0.003(5 \cdot 10^{-4)}$	0.019(0.002)	$1 \cdot 10^{-5}(3 \cdot 10^{-5})$
	NNGP	0.078(0.055)	0.016(0.009)	0.126(0.086)	0.08(0.049)
	SGRLD	$0.002(3 \cdot 10^{-4})$	$0.001(1 \cdot 10^{-4})$	$0.004(6.1 \cdot 10^{-3})$	$0.4 \cdot 10^{-4}(1 \cdot 10^{-}5)$

Simulation results

Table 5: Coverage of the 95% credible intervals (Monte Carlo standard errors) for the covariance parameters computed using 100 simulations, each having sample size N. The proposed SGRLD method is compared with other SGMCMC methods (pSGLD, ADAMSGLD, MSGLD) and the full likelihood NNGP method.

N	Algorithm	Variance, σ^2	Range, ρ	Smoothness, ν	Nugget, τ^2
10^4	pSGLD	0.977(0.02)	0.845(0.06)	0.815(0.06)	0.931(0.05)
	ADAMSGLD	0.886(0.05)	0.791(0.08)	0.647(0.08)	0.636(0.05)
	MSGLD	0.793(0.03)	0.847(0.07)	0.709(0.07)	0.683(0.05)
	NNGP	0.783(0.06)	0.776(0.05)	0.614(0.07)	0.812(0.01)
	SGRLD	0.955(0.03)	0.924(0.05)	0.909(0.04)	0.935(0.01)
10^5	pSGLD	0.991(0.03)	0.913(0.04)	0.862(0.05)	0.965(0.02)
	ADAMSGLD	0.861(0.03)	0.754(0.07)	0.814(0.03)	0.738(0.05)
	MSGLD	0.896(0.04)	0.881(0.07)	0.774(0.08)	0.872(0.07)
	NNGP	0.826(0.05)	0.758(0.04)	0.714(0.03)	0.872(0.02)
	SGRLD	0.957(0.01)	0.964(0.01)	0.948(0.01)	$0.932(5 \cdot 10^{-3})$
10^6	pSGLD	$0.987(6 \cdot 10^{-3})$	0.934(0.02)	0.901(0.03)	0.961(0.01)
	ADAMSGLD	0.902(0.01)	$0.824(10^{-3})$	0.838(0.02)	0.781(0.03)
	MSGLD	$0.884(10^{-3})$	0.918(0.02)	0.846(0.01)	0.926(0.01)
	NNGP	0.866(0.03)	0.818(0.06)	0.834(0.04)	0.862(0.01)
	SGRLD	$0.968(6 \cdot 10^{-3})$	$0.941(8 \cdot 10^{-3})$	$0.929(5 \cdot 10^{-3})$	$0.941(2 \cdot 10^{-3})$

Simulation results

Table 6: Effective sample size per minute (Monte Carlo standard errors) of covariance parameters computed using 100 simulations, each having sample size N. The proposed SGRLD method is compared with other SGMCMC methods (pSGLD, ADAMSGLD, MSGLD) and the full likelihood NNGP method.

N	Algorithm	Variance, σ^2	Range, ρ	Smoothness, ν	Nugget, τ^2
10^4	pSGLD	42.97(1.57)	8.43(0.54)	4.33(0.26)	9.82(0.79)
	ADAMSGLD	9.12(0.45)	4.22(0.33)	2.85(0.28)	3.80(0.48)
	MSGLD	15.68(0.95)	6.48(0.70)	3.65(0.44)	5.11(0.78)
	NNGP	1.02(0.33)	0.99(0.24)	1.11(0.75)	0.51(0.14)
	SGRLD	23.8(1.15)	23.9(1.19)	25.2(1.25)	30.5(1.55)
10^5	pSGLD	66.87(2.09)	10.06(0.65)	3.59(0.21)	11.3(0.79)
	ADAMSGLD	7.87(0.38)	2.37(0.27)	1.15(0.13)	1.64(0.24)
	MSGLD	12.92(0.67)	3.15(0.36)	1.206(0.11)	1.71(0.13)
	NNGP	0.89(0.08)	0.75(0.31)	1.02(0.14)	0.47(0.07)
	SGRLD	22.7(0.33)	22.44(0.27)	22.69(0.13)	23.23(0.34)
10^6	pSGLD	96.49(3.37)	13.68(0.81)	3.04(0.11)	9.74(0.42)
	ADAMSGLD	6.17(0.13)	4.56(0.52)	1.98(0.62)	2.36(0.83)
	MSGLD	15.07(1.01)	3.78(0.81)	2.06(0.30)	5.01(0.97)
	NNGP	0.81(0.16)	1.01(0.34)	0.28(0.05)	0.52(0.03)
	SGRLD	25.8(0.14)	26.05(0.18)	29.62(0.28)	24.07(0.27)