Scalable Gaussian processes for non-ergodic earthquake models

Stephen Huan

https://cgdct.moe

SURF Summer Seminar Day 2023

Collaborators

Houman Owhadi, Caltech

Greg Lavrentiadis, Caltech

Yifan Chen, Caltech

Pau Batlle, Caltech

Collaborators

Florian Schäfer,
Gatech

Overview

The problem

Non-ergodic ground-motion models [Lavrentiadis et al. 2022] estimate the probability an earthquake exceeds a fixed intensity

The problem

Non-ergodic ground-motion models [Lavrentiadis et al. 2022] estimate the probability an earthquake exceeds a fixed intensity

Ergodic refers to assumption of translation invariance

The problem

Non-ergodic ground-motion models [Lavrentiadis et al. 2022] estimate the probability an earthquake exceeds a fixed intensity

Ergodic refers to assumption of translation invariance

Gaussian process modeling provides uncertainty quantification

The problem

Non-ergodic ground-motion models [Lavrentiadis et al. 2022] estimate the probability an earthquake exceeds a fixed intensity

Ergodic refers to assumption of translation invariance

Gaussian process modeling provides uncertainty quantification

Seismic hazard at nuclear power plant locations

Gaussian process regression

Given dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, learn residual $y_i = f(\boldsymbol{x}_i)$

Gaussian process regression

Given dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, learn residual $y_i = f(\boldsymbol{x}_i)$

Gaussian process (GP) modeling $f \sim \mathcal{GP}(\mu(\cdot), K(\cdot, \cdot))$

Gaussian process regression

Given dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, learn residual $y_i = f(\boldsymbol{x}_i)$

Gaussian process (GP) modeling $f \sim \mathcal{GP}(\mu(\cdot), K(\cdot, \cdot))$

Use closed-form posterior predictions

$$\begin{aligned}
\mathbb{E}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] &= \boldsymbol{\mu}_{\mathsf{Pr}} + \Theta_{\mathsf{Pr},\mathsf{Tr}} \Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1} (\boldsymbol{y}_{\mathsf{Tr}} - \boldsymbol{\mu}_{\mathsf{Tr}}) \\
\mathbb{C}\mathrm{ov}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] &= \Theta_{\mathsf{Pr},\mathsf{Pr}} - \Theta_{\mathsf{Pr},\mathsf{Tr}} \Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1} \Theta_{\mathsf{Tr},\mathsf{Pr}}
\end{aligned}$$

Gaussian process regression

Given dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, learn residual $y_i = f(\boldsymbol{x}_i)$

Gaussian process (GP) modeling $f \sim \mathcal{GP}(\mu(\cdot), K(\cdot, \cdot))$

Use closed-form posterior predictions

$$\begin{aligned}
\mathbb{E}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] &= \boldsymbol{\mu}_{\mathsf{Pr}} + \Theta_{\mathsf{Pr},\mathsf{Tr}} \Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1} (\boldsymbol{y}_{\mathsf{Tr}} - \boldsymbol{\mu}_{\mathsf{Tr}}) \\
\mathbb{C}\mathrm{ov}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] &= \Theta_{\mathsf{Pr},\mathsf{Pr}} - \Theta_{\mathsf{Pr},\mathsf{Tr}} \Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1} \Theta_{\mathsf{Tr},\mathsf{Pr}}
\end{aligned}$$

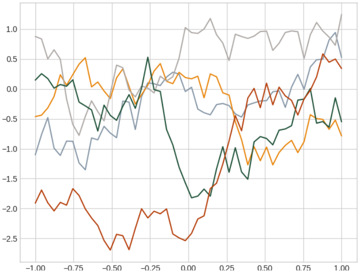
Direct computation scales as $\mathcal{O}(N^3)$, limiting data size (10^4)

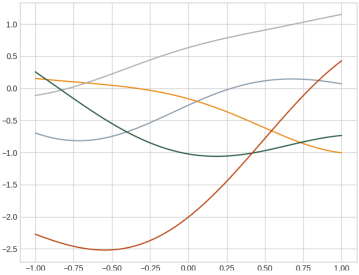
Matérn kernel functions

Matérn family of kernels with smoothness ν and length scale ℓ

$\nu = 1/2$ corresponds to the exponential kernel $\psi^2 \exp(-r/\ell)$

$\nu = \infty$ to the squared exponential kernel $\psi^2 \exp(-r^2/(2\ell^2))$





Kernel function

Use kernel

$$c_1(t_E) + c_2(t_S) + X_3 c_3(t_E, t_S) + [\Delta R \cdot c_{\mathsf{ca}}(t_C)] + \delta W + \delta B$$

where

- c_1 models earthquake interactions
- c_2 models site (receiver) interactions
- X_3 is the geometric scaling spreading
- c_3 models the interaction between earthquakes and sites
- ΔR is a cell path distance array
- c_{ca} models cell-specific path attenuation
- δW is a noise nugget
- δB is noise shared within the same earthquake event

Modeling overview

Pick (parametric) class of kernel functions

Learn hyperparameters (MLE, full Bayesian, kernel flows, . . .)

Make predictions

What do we need?

(log-)Likelihood, posterior statistics

$$
\begin{aligned}
-2\log\eta(\boldsymbol{y}) &= \operatorname{logdet}(\Theta) + \boldsymbol{y}^\top\Theta^{-1}\boldsymbol{y} + N\log(2\pi) \\
\mathbb{E}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] &= \Theta_{\mathsf{Pr},\mathsf{Tr}}\Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1}\boldsymbol{y}_{\mathsf{Tr}} \\
\mathbb{C}\mathrm{ov}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] &= \Theta_{\mathsf{Pr},\mathsf{Pr}} - \Theta_{\mathsf{Pr},\mathsf{Tr}}\Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1}\Theta_{\mathsf{Tr},\mathsf{Pr}}
\end{aligned}
$$

What do we need?

(log-)Likelihood, posterior statistics

$$
\begin{aligned}
-2\log\eta(\boldsymbol{y}) &= \operatorname{logdet}(\Theta) + \boldsymbol{y}^\top \Theta^{-1}\boldsymbol{y} + N\log(2\pi) \\
\mathbb{E}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] &= \Theta_{\mathsf{Pr},\mathsf{Tr}}\Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1}\boldsymbol{y}_{\mathsf{Tr}} \\
\mathbb{C}\mathrm{ov}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] &= \Theta_{\mathsf{Pr},\mathsf{Pr}} - \Theta_{\mathsf{Pr},\mathsf{Tr}}\Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1}\Theta_{\mathsf{Tr},\mathsf{Pr}}
\end{aligned}
$$

Log determinant, inversion

What do we need?

(log-)Likelihood, posterior statistics

$$-2 \log \eta(\boldsymbol{y}) = \operatorname{logdet}(\Theta) + \boldsymbol{y}^\top \Theta^{-1} \boldsymbol{y} + N \log(2\pi)$$
$$\mathbb{E}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] = \Theta_{\mathsf{Pr},\mathsf{Tr}} \Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1} \boldsymbol{y}_{\mathsf{Tr}}$$
$$\mathbb{C}\mathrm{ov}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] = \Theta_{\mathsf{Pr},\mathsf{Pr}} - \Theta_{\mathsf{Pr},\mathsf{Tr}} \Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1} \Theta_{\mathsf{Tr},\mathsf{Pr}}$$

Log determinant, inversion

Accelerated with *Cholesky factor* $\Theta = LL^\top$

What do we need?

(log-)Likelihood, posterior statistics

$$
\begin{aligned}
-2\log\eta(\boldsymbol{y}) &= \operatorname{logdet}(\Theta) + \boldsymbol{y}^\top \Theta^{-1}\boldsymbol{y} + N\log(2\pi) \\
\mathbb{E}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] &= \Theta_{\mathsf{Pr},\mathsf{Tr}}\Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1}\boldsymbol{y}_{\mathsf{Tr}} \\
\mathbb{C}\mathrm{ov}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] &= \Theta_{\mathsf{Pr},\mathsf{Pr}} - \Theta_{\mathsf{Pr},\mathsf{Tr}}\Theta_{\mathsf{Tr},\mathsf{Tr}}^{-1}\Theta_{\mathsf{Tr},\mathsf{Pr}}
\end{aligned}
$$

Log determinant, inversion

Accelerated with *Cholesky factor* $\Theta = LL^\top$

Seek *sparse* Cholesky factor for *dense* covariance matrix

Statistical Cholesky factorization

Cholesky factorization $\Leftrightarrow$ iterative conditioning of process

$$L = \mathrm{chol}(\Theta^{-1})$$
$$-\frac{L_{i,j}}{L_{j,j}} = \frac{\mathbb{C}\mathrm{ov}[y_i, y_j \mid y_{k>j,k\neq i}]}{\mathbb{V}\mathrm{ar}[y_j \mid y_{k>j,k\neq i}]}$$

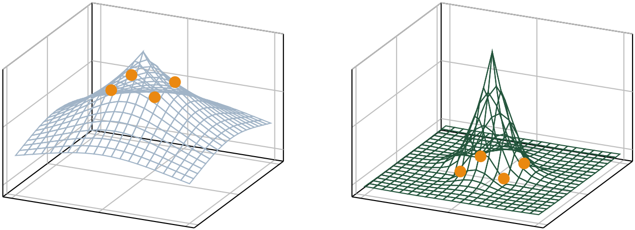
Statistical Cholesky factorization

Cholesky factorization $\Leftrightarrow$ iterative conditioning of process

$$L = \operatorname{chol}(\Theta^{-1})$$
$$-\frac{L_{i,j}}{L_{j,j}} = \frac{\mathbb{C}\mathrm{ov}[y_i, y_j \mid y_{k>j, k\neq i}]}{\mathbb{V}\mathrm{ar}[y_j \mid y_{k>j, k\neq i}]}$$

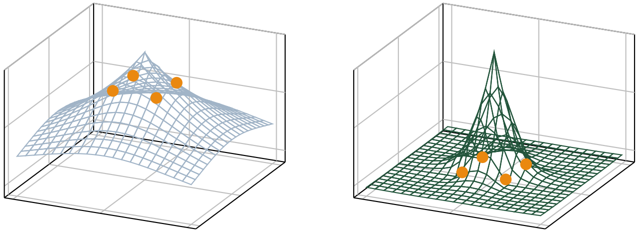
Conditional (near)-independence $\Leftrightarrow$ (approximate) sparsity

Screening effect



Conditional on points near a point of interest,
far away points are almost independent [Stein 2002]

Screening effect



Conditional on points near a point of interest,
far away points are almost independent [Stein 2002]

Suggests space-covering ordering and selecting nearby points

Cholesky factorization recipe

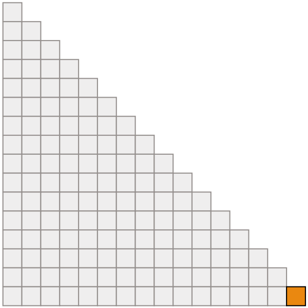
Implied procedure for computing $LL^\top \approx \Theta^{-1}$

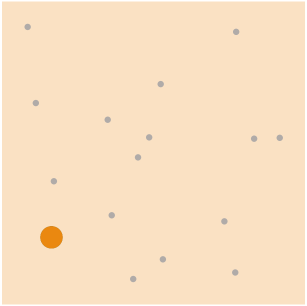
1. Pick an ordering on the rows/columns of Θ
2. Select a sparsity pattern lower triangular w.r.t. ordering
3. Compute entries by minimizing objective over all factors

Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho\ell_i$ from x_i

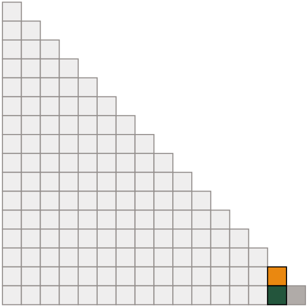


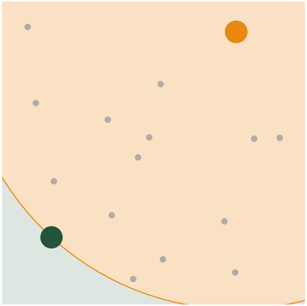


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho\ell_i$ from x_i

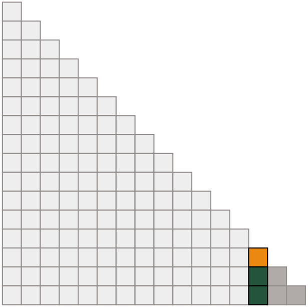


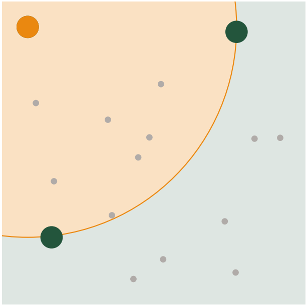


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho \ell_i$ from x_i

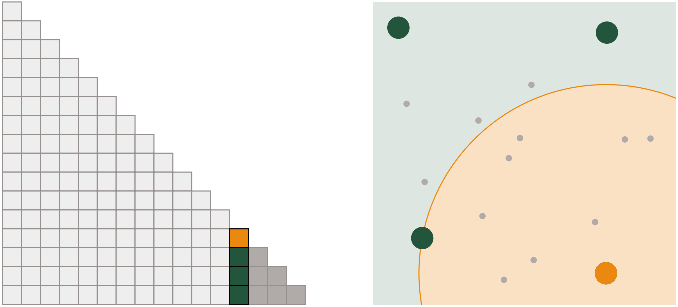




Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point $\mathbf{x}_i$ with largest distance ℓ_i to points selected before

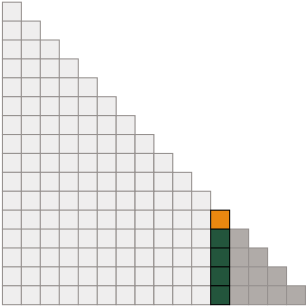
The ith column selects all points within a radius of $\rho\ell_i$ from $\mathbf{x}_i$

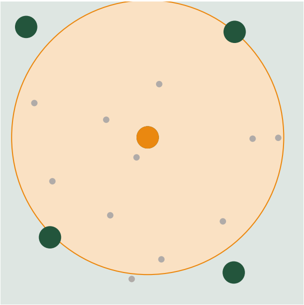


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho\ell_i$ from x_i

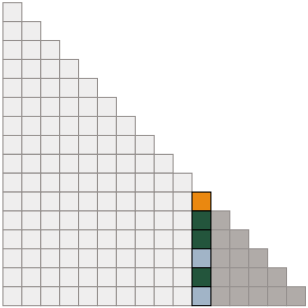


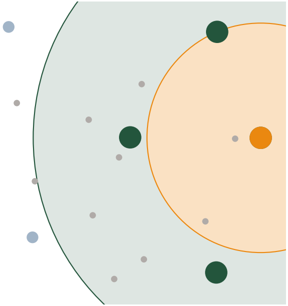


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho \ell_i$ from x_i

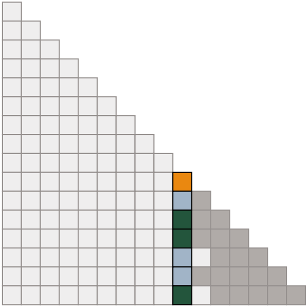


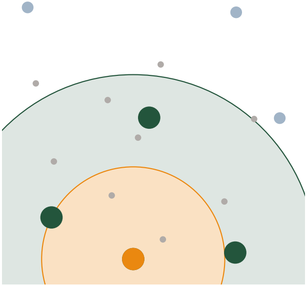


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho\ell_i$ from x_i

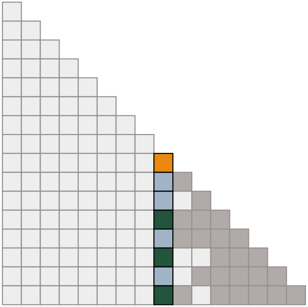


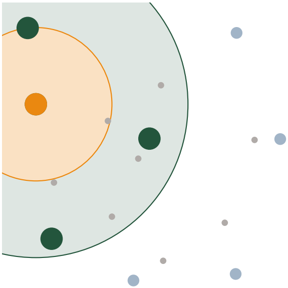


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho\ell_i$ from x_i

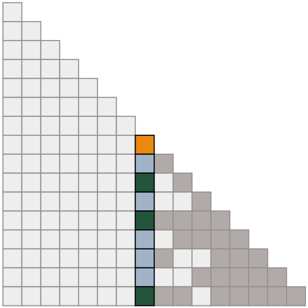


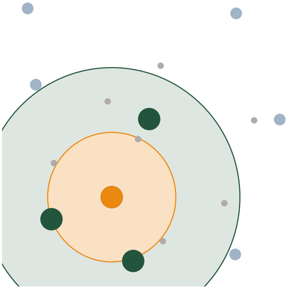


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho \ell_i$ from x_i

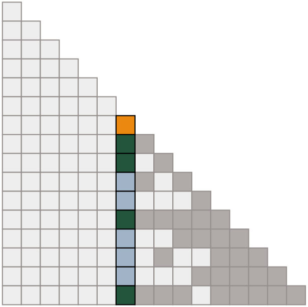


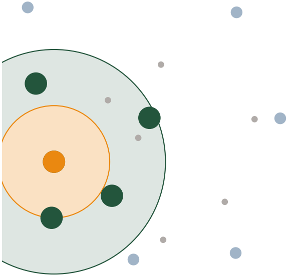


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho\ell_i$ from x_i

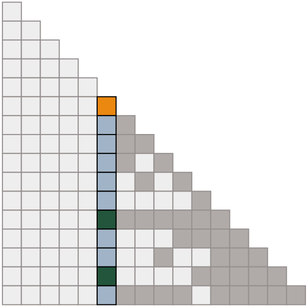


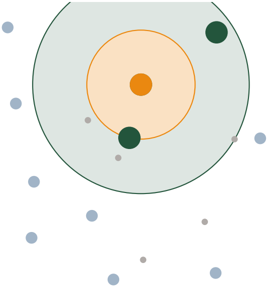


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho \ell_i$ from x_i

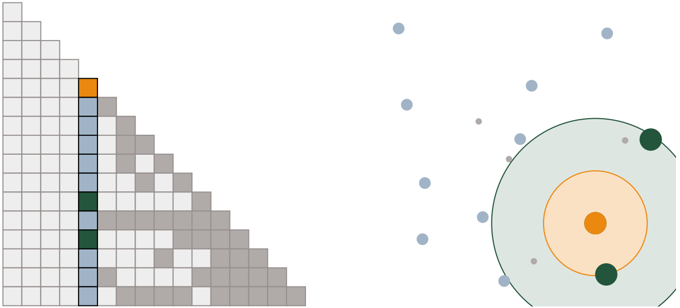




Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

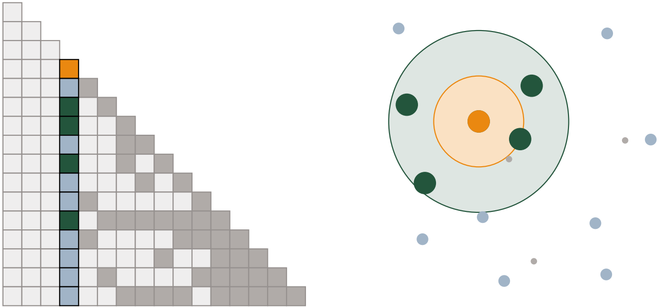
The ith column selects all points within a radius of $\rho\ell_i$ from x_i



Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point $\boldsymbol{x}_i$ with largest distance ℓ_i to points selected before

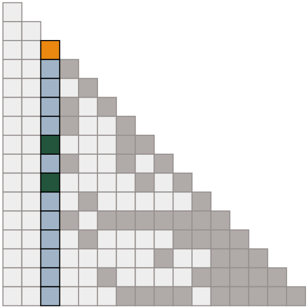
The ith column selects all points within a radius of $\rho\ell_i$ from $\boldsymbol{x}_i$

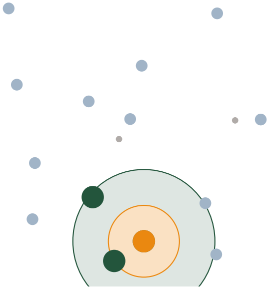


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho\ell_i$ from x_i

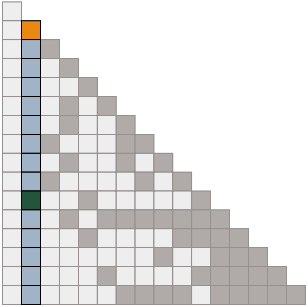


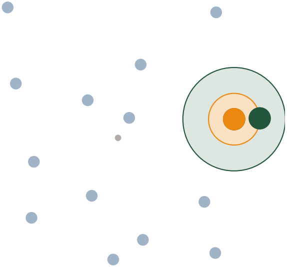


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho \ell_i$ from x_i

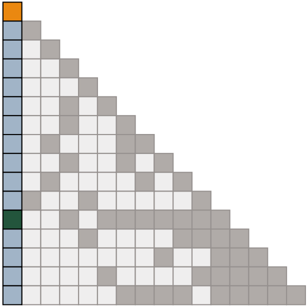


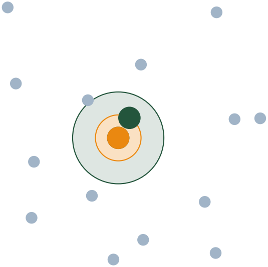


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point x_i with largest distance ℓ_i to points selected before

The ith column selects all points within a radius of $\rho \ell_i$ from x_i





Kullback-Leibler minimization

Compute entries by minimizing Kullback-Leibler divergence

$$L := \underset{\hat{L} \in \mathcal{S}}{\operatorname{argmin}}\ \mathbb{D}_{\mathrm{KL}}\Big(\mathcal{N}(\mathbf{0}, \Theta) \,\Big\|\, \mathcal{N}(\mathbf{0}, (\hat{L}\hat{L}^\top)^{-1})\Big)$$

Kullback-Leibler minimization

Compute entries by minimizing Kullback-Leibler divergence

$$L := \underset{\hat{L} \in \mathcal{S}}{\operatorname{argmin}} \; \mathbb{D}_{\mathrm{KL}}\Big(\mathcal{N}(\mathbf{0}, \Theta) \, \Big\| \, \mathcal{N}(\mathbf{0}, (\hat{L}\hat{L}^\top)^{-1})\Big)$$

Efficient and embarrassingly parallel closed-form solution

$$L_{s_i, i} = \frac{\Theta_{s_i, s_i}^{-1} \boldsymbol{e}_1}{\sqrt{\boldsymbol{e}_1^\top \Theta_{s_i, s_i}^{-1} \boldsymbol{e}_1}}$$

Kullback-Leibler minimization

Compute entries by minimizing Kullback-Leibler divergence

$$L := \underset{\hat{L} \in \mathcal{S}}{\operatorname{argmin}} \, \mathbb{D}_{\mathrm{KL}}\Big(\mathcal{N}(\mathbf{0}, \Theta) \, \Big\| \, \mathcal{N}(\mathbf{0}, (\hat{L}\hat{L}^{\top})^{-1}) \Big)$$

Efficient and embarrassingly parallel closed-form solution

$$L_{s_i,i} = \frac{\Theta_{s_i,s_i}^{-1} \boldsymbol{e}_1}{\sqrt{\boldsymbol{e}_1^{\top} \Theta_{s_i,s_i}^{-1} \boldsymbol{e}_1}}$$

Achieves state of the art ϵ-accuracy in time complexity $\mathcal{O}\left(N \log^{2d}\left(\frac{N}{\epsilon}\right)\right)$ with $\mathcal{O}\left(N \log^{d}\left(\frac{N}{\epsilon}\right)\right)$ nonzero entries [Schäfer, Katzfuss, and Owhadi 2021]

Geometric dependence

Screening effect motivated by geometric considerations

Geometric dependence

Screening effect motivated by geometric considerations

Maximin ordering worse than random for spatial dimension ≥ 4

Nearest neighbors unclear for paths

Geometric dependence

Screening effect motivated by geometric considerations

Maximin ordering worse than random for spatial dimension ≥ 4

Nearest neighbors unclear for paths

Quick fix: correlation distance

$$\mathsf{dist}(p, q) := \sqrt{1 - |\rho|}$$
$$\rho(p, q) := \frac{k(p, q)}{\sqrt{k(p, p)k(q, q)}}$$

Towards geometry-free Cholesky factors

RPCholesky [Chen et al. 2023] + random ordering

RPCholesky + nearest neighbors + random candidate sets

Conditional selection sparsity pattern [Huan et al. 2023]

Automatic interpolation between low rank/sparse

Summary

Non-ergodic earthquake models with Gaussian processes

Efficient computation with sparse Cholesky factors

Implemented in Julia, scale to HPC/supercomputers

Project website and additional resources can be found at

`https://kolesky.cgdct.moe`

Thank you!

Chen, Yifan et al. (Feb. 2023). *Randomly Pivoted Cholesky: Practical Approximation of a Kernel Matrix with Few Entry Evaluations*. DOI: 10.48550/arXiv.2207.06503. arXiv: 2207.06503 [cs, math, stat].

Guinness, Joseph (Oct. 2018). "Permutation and Grouping Methods for Sharpening Gaussian Process Approximations". In: *Technometrics* 60.4, pp. 415–429. ISSN: 0040-1706, 1537-2723. DOI: 10.1080/00401706.2018.1437476. arXiv: 1609.05372 [stat].

Huan, Stephen et al. (July 2023). *Sparse Cholesky Factorization by Greedy Conditional Selection*. DOI: 10.48550/arXiv.2307.11648. arXiv: 2307.11648 [cs, math, stat].

Lavrentiadis, Grigorios et al. (Aug. 2022). "Overview and Introduction to Development of Non-Ergodic Earthquake Ground-Motion Models". In: *Bulletin of Earthquake Engineering*. ISSN: 1573-1456. DOI: 10.1007/s10518-022-01485-x.

Schäfer, Florian, Matthias Katzfuss, and Houman Owhadi (Oct. 2021). "Sparse Cholesky Factorization by Kullback-Leibler Minimization". In: *arXiv:2004.14455 [cs, math, stat]*. arXiv: 2004.14455 [cs, math, stat].

Stein, Michael L. (Feb. 2002). "The Screening Effect in Kriging". In: *The Annals of Statistics* 30.1, pp. 298–323. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/aos/1015362194.

Kernels on paths

Integral of a Matérn kernel $k(\boldsymbol{x}, \boldsymbol{x}')$

If $f \sim \mathcal{GP}(\mathbf{0}, k)$, then define $\widetilde{f} = \int_0^1 f(\boldsymbol{x} + t(\boldsymbol{x}' - \boldsymbol{x}))\, \mathsf{d}t$

Linear transformation of a GP is also a GP

It has covariance

$$\widetilde{k}(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{y}, \boldsymbol{y}') = \int_0^1 \int_0^1 k(\boldsymbol{x} + t(\boldsymbol{x}' - \boldsymbol{x}), \boldsymbol{y} + s(\boldsymbol{y}' - \boldsymbol{y}))\, \mathsf{d}t\, \mathsf{d}s$$

which creates "paths" in the 2-d input space.