Computing transport maps by cumulant matching

Stephen Huan

https://cgdct.moe

2024-03-15 group meeting

Overview

Fast inference in Gaussian processes

Scientific applications at exascale

Towards general preconditioners

Transport by cumulant matching

Part 1: Fast inference in Gaussian processes

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

The problem

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

Posterior predictions

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

Likelihood $-2 \log \eta(\boldsymbol{y}) = \operatorname{logdet}(\Theta) + \boldsymbol{y}^{\top} \Theta^{-1} \boldsymbol{y} + N \log(2\pi)$

Sampling $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \mathsf{Id}), L^{-\top} \boldsymbol{z} + \boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\mu}, \Theta)$

The problem

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

Posterior predictions

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

Likelihood $-2 \log \eta(\boldsymbol{y}) = \operatorname{logdet}(\Theta) + \boldsymbol{y}^\top \Theta^{-1} \boldsymbol{y} + N \log(2\pi)$

Sampling $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathsf{Id}), L^{-\top} \boldsymbol{z} + \boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\mu}, \Theta)$

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

Statistical Cholesky factorization

Cholesky factorization $\Leftrightarrow$ iterative conditioning of process

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

Statistical Cholesky factorization

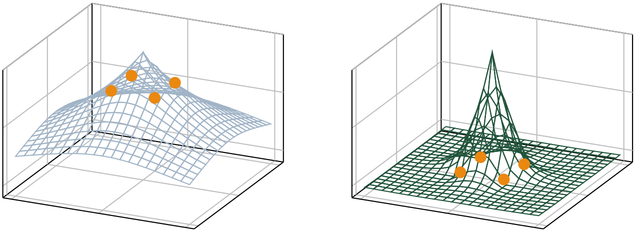
Cholesky factorization $\Leftrightarrow$ iterative conditioning of process

$$L = \mathrm{chol}(\Theta^{-1})$$

$$-\frac{L_{i,j}}{L_{j,j}} = \frac{\mathbb{C}\mathrm{ov}[y_i, y_j \mid y_{k>j,k\neq i}]}{\mathbb{V}\mathrm{ar}[y_j \mid y_{k>j,k\neq i}]}$$

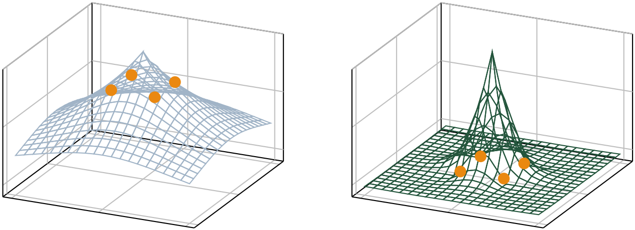
Conditional (near)-independence $\Leftrightarrow$ (approximate) sparsity

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

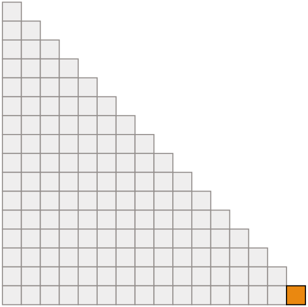
Procedure for computing $LL^\top \approx \Theta^{-1}$

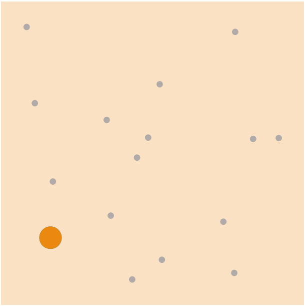
1. Pick an ordering on the rows/columns of Θ
2. Select a sparsity pattern lower triangular w.r.t. ordering
3. Compute entries by minimizing objective over all factors

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

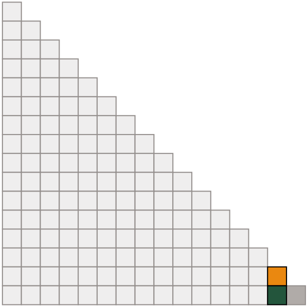


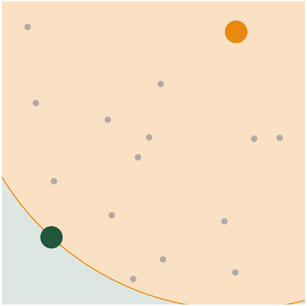


Ordering and sparsity pattern

(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

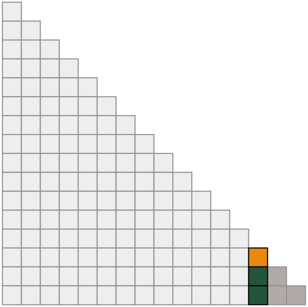


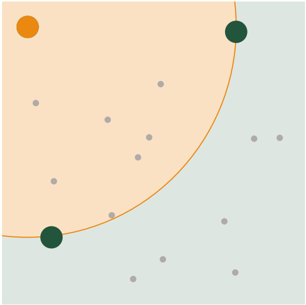


Ordering and sparsity pattern

(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

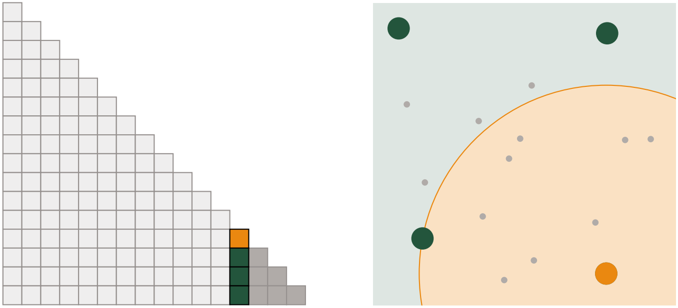




Ordering and sparsity pattern

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

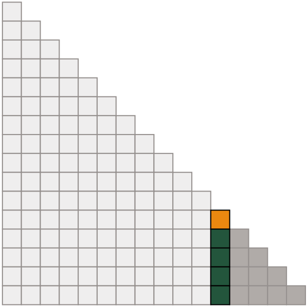
The ith column selects all points within a radius of $\rho \ell_i$ from x_i

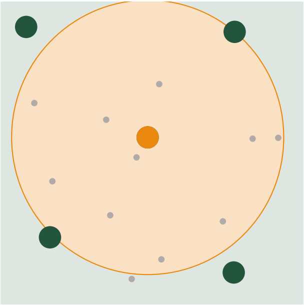


Ordering and sparsity pattern

(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

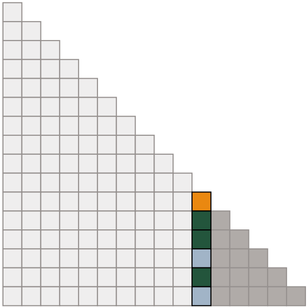


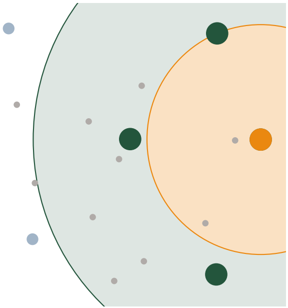


Ordering and sparsity pattern

(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

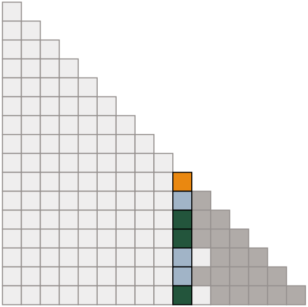


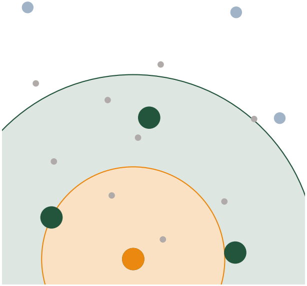


Ordering and sparsity pattern

(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

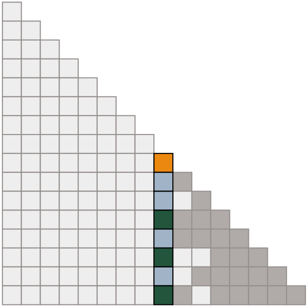


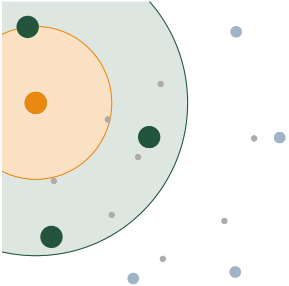


Ordering and sparsity pattern

(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

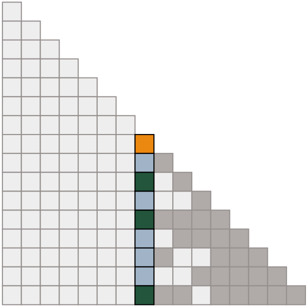


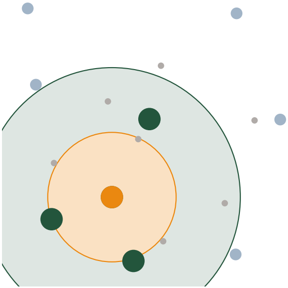


Ordering and sparsity pattern

(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

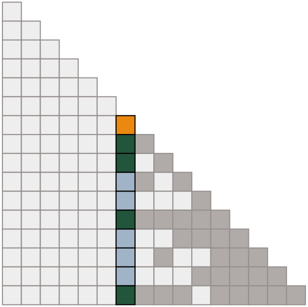


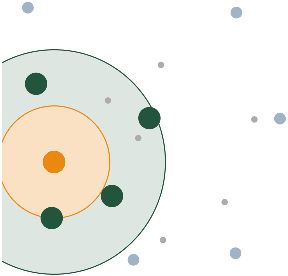


Ordering and sparsity pattern

(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

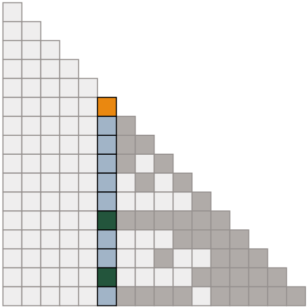


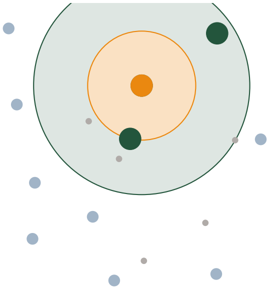


Ordering and sparsity pattern

(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

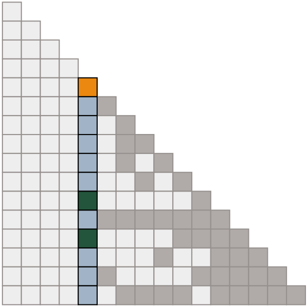


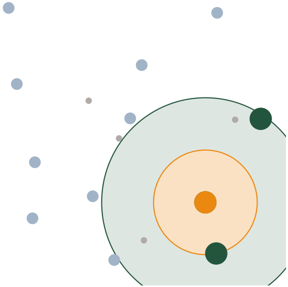


Ordering and sparsity pattern

(Reverse) maximin ordering [Guinness 2018] selects the next point 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

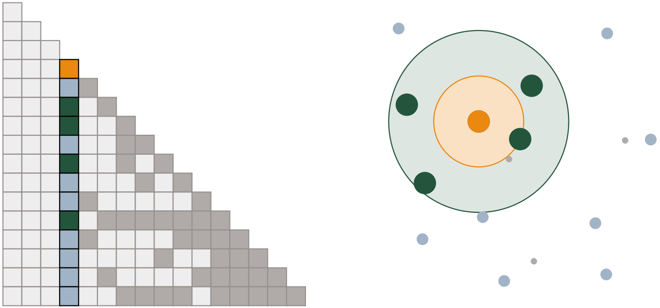




Ordering and sparsity pattern

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

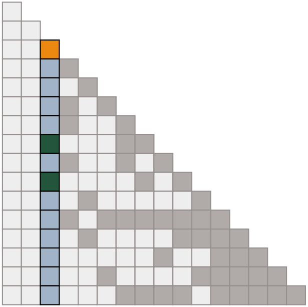
The ith column selects all points within a radius of $\rho \ell_i$ from x_i

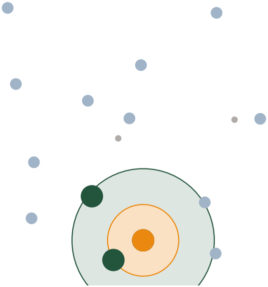


Ordering and sparsity pattern

(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

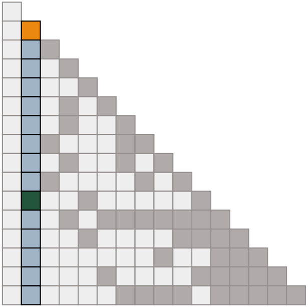


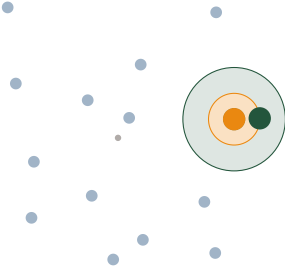


Ordering and sparsity pattern

(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

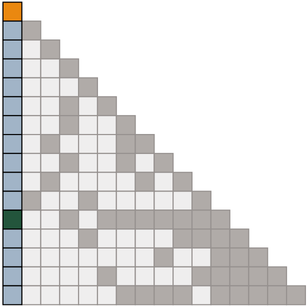


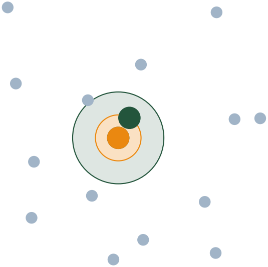


Ordering and sparsity pattern

(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 $\boldsymbol{x}_i$





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

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

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 $\mathcal{O}\left(N \log^{2d}\left(\frac{N}{\varepsilon}\right)\right)$ with $\mathcal{O}\left(N \log^{d}\left(\frac{N}{\varepsilon}\right)\right)$ nonzero entries [Schäfer, Katzfuss, and Owhadi 2021]

KL-minimization, revisited

Plug optimal L back into the KL divergence

$$\mathbb{D}_{\mathrm{KL}}\left(\Theta \,\middle\|\, (LL^\top)^{-1}\right) = \sum_{i=1}^{N} \left[\log\left(\Theta_{i,i \mid s_i \setminus \{i\}}\right) - \log\left(\Theta_{i,i \mid i+1:}\right)\right]$$

KL-minimization, revisited

Plug optimal L back into the KL divergence

$$\mathbb{D}_{\mathrm{KL}}\left(\Theta \,\middle\|\, (LL^\top)^{-1}\right) = \sum_{i=1}^{N} \left[\log\left(\Theta_{i,i|s_i \setminus \{i\}}\right) - \log\left(\Theta_{i,i|i+1:}\right)\right]$$

KL is accumulated error over independent regression problems

KL-minimization, revisited

Plug optimal L back into the KL divergence

$$\mathbb{D}_{\mathrm{KL}}\Big(\Theta \,\Big\|\, (LL^{\top})^{-1}\Big) = \sum_{i=1}^{N} \left[\log\left(\Theta_{i,i|s_i \setminus \{i\}}\right) - \log\left(\Theta_{i,i|i+1:}\right)\right]$$

KL is accumulated error over independent regression problems

Goal: minimize posterior variance of i-th prediction point by selecting training points s_i *most informative* to that point

KL-minimization, revisited

Plug optimal L back into the KL divergence

$$\mathbb{D}_{\mathrm{KL}}\left(\Theta \,\middle\|\, (LL^{\top})^{-1}\right) = \sum_{i=1}^{N} \left[\log\left(\Theta_{i,i|s_i\setminus\{i\}}\right) - \log\left(\Theta_{i,i|i+1:}\right)\right]$$

KL is accumulated error over independent regression problems

Goal: minimize posterior variance of i-th prediction point by selecting training points s_i *most informative* to that point

Variance $\Leftrightarrow$ mutual information $\Leftrightarrow$ mean squared error

$$\mathbb{H}[\boldsymbol{y}_{\mathsf{Pr}}] = \mathbb{H}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}] + \mathbb{I}[\boldsymbol{y}_{\mathsf{Pr}}; \boldsymbol{y}_{\mathsf{Tr}}]$$

$$\mathbb{V}\mathrm{ar}[\boldsymbol{y}_{\mathsf{Pr}}] = \mathbb{E}[\mathbb{V}\mathrm{ar}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}]] + \mathbb{V}\mathrm{ar}[\mathbb{E}[\boldsymbol{y}_{\mathsf{Pr}} \mid \boldsymbol{y}_{\mathsf{Tr}}]]$$

https://cgdct.moe/projects/cholesky/

Part 2: Scientific applications at exascale

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

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

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

Kernels on paths

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

Linear transformation of a GP is also a GP

Kernels on paths

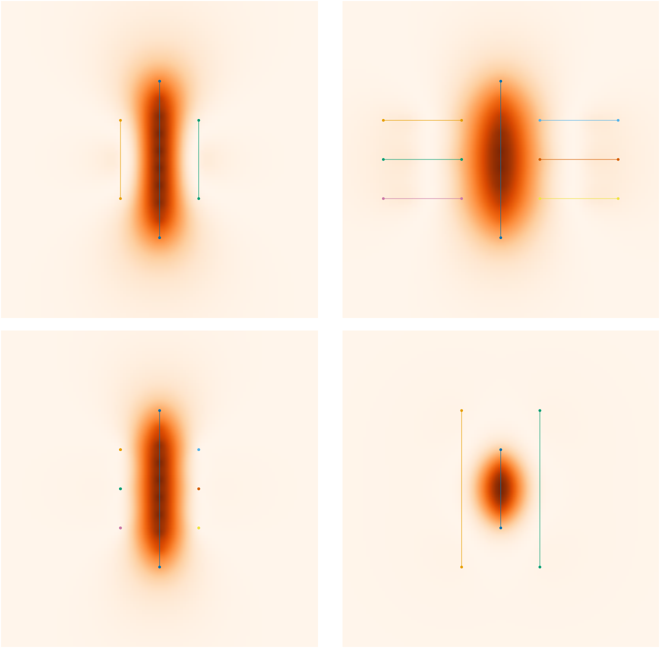
For $f \sim \mathcal{GP}(\mathbf{0}, k)$, define $\widetilde{f} = \int_0^1 f(\boldsymbol{x} + t(\boldsymbol{x}' - \boldsymbol{x}))\,\mathsf{d}t$

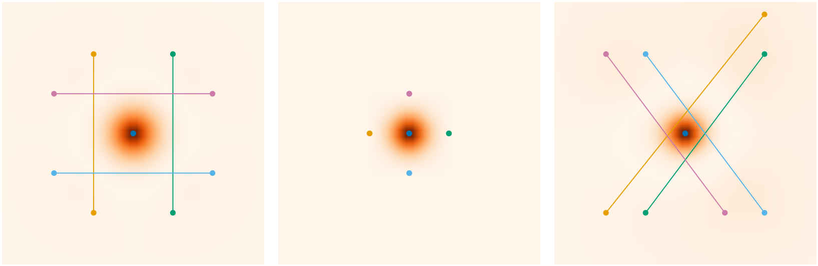
Linear transformation of a GP is also a GP

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.





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

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

Geometric dependence

Screening effect motivated by geometric considerations

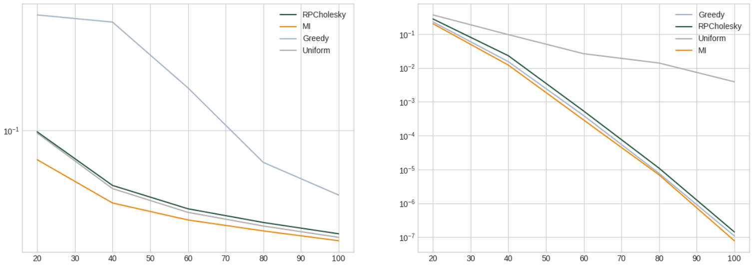
Maximin ordering worse than random for spatial dimension ≥ 4

Nearest neighbors unclear for paths

Quick fix: correlation distance

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

Information-theoretic orderings?



RPCholesky
MI
Greedy
Uniform
10^{-1}
20
30
40
50
60
70
80
90
100
Greedy
RPCholesky
Uniform
MI
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
20
30
40
50
60
70
80
90
100

https://kolesky.cgdct.moe/

Part 3: Towards general preconditioners

Sparse versus low rank

Sparsity doesn't scale to high dimension

"All high dimensional data is low rank"

Best-of-both-worlds approximation?

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

Applications to GPs, optimal transport, optimization, etc.

Implications for algorithmic design

Automation essential from a user perspective

User-friendly software libraries (Cython, JAX, Julia)

Part 4: Transport by cumulant matching

A *partially ordered set* is a set X equipped with a partial order

$$\leq \subseteq X \times X.$$

A *partially ordered set* is a set X equipped with a partial order

$$\leq \subseteq X \times X.$$

The relation $\leq$ satisfies

1. Reflexivity: $x \leq x$.
2. Antisymmetry: If $x \leq y$ and $y \leq x$, $x = y$.
3. Transitivity: If $x \leq y$ and $y \leq z$, $x \leq y$.

Lattice theory

A *partially ordered set* is a set X equipped with a partial order

$$\leq \subseteq X \times X.$$

The relation $\leq$ satisfies

1. Reflexivity: $x \leq x$.
2. Antisymmetry: If $x \leq y$ and $y \leq x$, $x = y$.
3. Transitivity: If $x \leq y$ and $y \leq z$, $x \leq y$.

A *lattice* $\mathcal{L}$ has the extra property that for $a, b \in \mathcal{L}$, there are unique greatest lower and least upper bounds $a \wedge b, a \vee b \in \mathcal{L}$.

Partition lattice

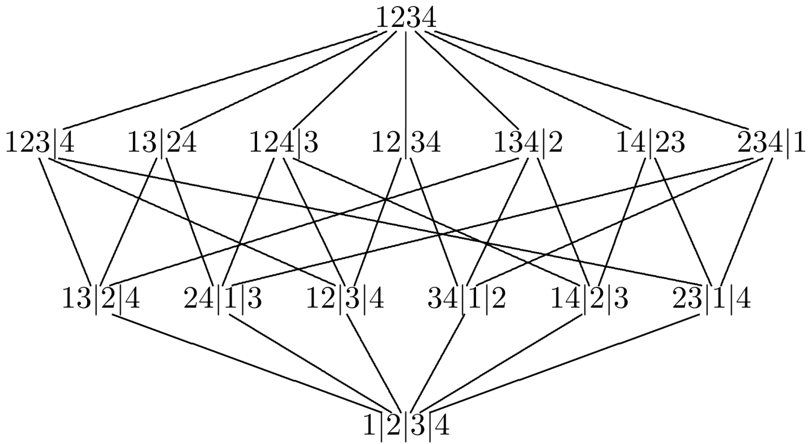


Figure: Partition lattice: $a \leq b$ if a is a sub-partition of b.

For a lattice function $f : \mathcal{L} \to \mathbb{R}$, define the *integral*

$$F(a) = \sum_{b \leq a} f(b).$$

Lattice calculus

For a lattice function $f : \mathcal{L} \to \mathbb{R}$, define the *integral*

$$F(a) = \sum_{b \leq a} f(b).$$

Formal inverse the *derivative*

$$f(a) = \sum_{b \leq a} m(b,a) F(b),$$

where m is the *Möbius function* of the lattice.

Möbius function of the partition lattice

On the partition lattice, it can be shown that

$$m(\sigma, \mathbf{1}) = (-1)^{\#\sigma - 1}(\#\sigma - 1)!.$$

Möbius function of the partition lattice

On the partition lattice, it can be shown that

$$m(\sigma, \mathbf{1}) = (-1)^{\#\sigma - 1}(\#\sigma - 1)!.$$

In addition, for $\sigma \leq \tau$ we have the isomorphism

$$[\sigma, \tau] \cong \bigotimes_{b \in \tau} \Upsilon_b$$

inspiring the formula

$$\prod_{b \in \tau} m(\mathbf{0}_b, \mathbf{1}_b).$$

Möbius function of the partition lattice

On the partition lattice, it can be shown that

$$m(\sigma, \mathbf{1}) = (-1)^{\#\sigma - 1}(\#\sigma - 1)!.$$

In addition, for $\sigma \leq \tau$ we have the isomorphism

$$[\sigma, \tau] \cong \bigotimes_{b \in \tau} \Upsilon_b$$

inspiring the formula

$$\prod_{b \in \tau} m(\mathbf{0}_b, \mathbf{1}_b).$$

Note that $m(\sigma, \tau) = 0$ if both $\sigma \not\leq \tau$ and $\tau \not\leq \sigma$.

Statistical interpretation

If f is the cumulant product

$$f(\tau) = \kappa(\tau_1) \cdots \kappa(\tau_\nu),$$

then the integral F is the moment product

$$F(\tau) = \mu(\tau_1) \cdots \mu(\tau_\nu).$$

Statistical interpretation

If f is the cumulant product

$$f(\tau) = \kappa(\tau_1) \cdots \kappa(\tau_\nu),$$

then the integral F is the moment product

$$F(\tau) = \mu(\tau_1) \cdots \mu(\tau_\nu).$$

The generalized cumulants $g(\tau) = \kappa(\tau)$ are given by

$$g(\tau) = \sum_{\sigma:\sigma\vee\tau=\mathbf{1}} f(\sigma).$$

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & & 1 & 1 & 1 & 1 & 1 & 1 & & & 1 & & 1 & 1 & \\
1 & 1 & & 1 & 1 & 1 & 1 & 1 & & 1 & & 1 & & 1 & \\
1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & & & 1 & 1 & & \\
1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & & & & \\
1 & 1 & 1 & 1 & 1 & & 1 & 1 & & 1 & 1 & 1 & 1 & & \\
1 & 1 & 1 & 1 & 1 & 1 & & 1 & 1 & & 1 & 1 & & 1 & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & & 1 & 1 & & & 1 & 1 & \\
1 & & & 1 & 1 & & 1 & 1 & & & & & & & \\
1 & & 1 & & 1 & 1 & & 1 & & & & & & & \\
1 & 1 & & & 1 & 1 & 1 & & & & & & & & \\
1 & & 1 & 1 & & 1 & 1 & & & & & & & & \\
1 & 1 & & 1 & & 1 & & 1 & & & & & & & \\
1 & 1 & 1 & & & & 1 & 1 & & & & & & & \\
1 & & & & & & & & & & & & & &
\end{pmatrix}
$$

References I

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.