Scalable Gaussian Process Regression with Gauss-Legendre Features

Paz Fink Shustin

University of Oxford

Probabilistic Numerics Spring School Southampton, April 2024

Joint work with Haim Avron

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

UNIVERSITY OF OXFORD
Mathematical Institute

Talk Outline

1. Gaussian Processes: From Linear to Functional Model
2. Gauss-Legendre Features
3. Parameter Computation
4. Spectral Equivalence
5. Efficient Compuations
6. Experiments

Linear Regression in Bayesian Modeling

Given: $\mathcal{D} = \{(\mathbf{x}_i, y_i) \,|i = 1, \ldots, n\} \subset \mathbb{R}^d \times \mathbb{R}$.
Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\mathsf{T}$.

Bayesian linear regression model:

$$f(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathbf{w}, \quad y = f(\mathbf{x}) + \varepsilon, \quad \varepsilon \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_n^2)\,.$$

- We specify a **prior** over the parameters: $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$.
- Gives rise to the **likelihood**: $p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \mathcal{N}\left(\mathbf{X}\mathbf{w}, \sigma_n^2\mathbf{I}\right)$.

Bayesian Modeling

Bayesian analysis is based on Bayes rule:

$$\underbrace{p\left(\mathbf{w}|\mathbf{y},\mathbf{X}\right)}_{\text{posterior}} = \frac{\overbrace{p(\mathbf{y}|\mathbf{X},\mathbf{w})}^{\text{likelihood}}\,\overbrace{p(\mathbf{w})}^{\text{prior}}}{\underbrace{p\left(\mathbf{y}|\mathbf{X}\right)}_{\text{marginal likelihood}}}$$

where the **marginal-likelihood** is:

$$p\left(\mathbf{y}|\mathbf{X}\right) = \int p\left(\mathbf{y}|\mathbf{X},\mathbf{w}\right) p\left(\mathbf{w}\right) d\mathbf{w}\,.$$

Thus, the **posterior** can be derived:

$$p\left(\mathbf{w}|\mathbf{y},\mathbf{X}\right) \propto \mathcal{N}\left(\sigma_n^{-2}\left(\sigma_n^{-2}\mathbf{X}^\mathsf{T}\mathbf{X} + \mathbf{\Sigma}^{-1}\right)^{-1}\mathbf{X}^\mathsf{T}\mathbf{y},\left(\sigma_n^{-2}\mathbf{X}^\mathsf{T}\mathbf{X} + \mathbf{\Sigma}^{-1}\right)^{-1}\right)\,.$$

Feature Maps in Function-space View

Replace $\mathbf{x}$ with $\phi(\mathbf{x})$, and write $f(\mathbf{x}) = \phi(\mathbf{x})^\mathsf{T}\mathbf{w}$ with a prior $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Then:

$$\mathbb{E}[f(\mathbf{x})] = \phi(\mathbf{x})^\mathsf{T}\mathbb{E}[\mathbf{w}] = 0$$

$$\mathbb{E}\left[f(\mathbf{x})f(\mathbf{x}')\right] = \phi(\mathbf{x})^\mathsf{T}\mathbb{E}\left[\mathbf{w}\mathbf{w}^\mathsf{T}\right]\phi(\mathbf{x}') = \phi(\mathbf{x})^\mathsf{T}\boldsymbol{\Sigma}\phi(\mathbf{x}')$$

Feature Maps in Function-space View

Replace $\mathbf{x}$ with $\phi(\mathbf{x})$, and write $f(\mathbf{x}) = \phi(\mathbf{x})^\mathsf{T}\mathbf{w}$ with a prior $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Then:

$$\mathbb{E}[f(\mathbf{x})] = \phi(\mathbf{x})^\mathsf{T}\mathbb{E}[\mathbf{w}] = 0$$

$$\mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] = \phi(\mathbf{x})^\mathsf{T}\mathbb{E}\left[\mathbf{w}\mathbf{w}^\mathsf{T}\right]\phi(\mathbf{x}') = \phi(\mathbf{x})^\mathsf{T}\mathbf{\Sigma}\phi(\mathbf{x}')$$

defining the feature map $\psi(\mathbf{x}) = \mathbf{\Sigma}^{1/2}\phi(\mathbf{x}')$ yields $k(\mathbf{x}, \mathbf{x}') = \psi(\mathbf{x})^\mathsf{T}\psi(\mathbf{x}) \Rightarrow$ the ***kernel trick***.

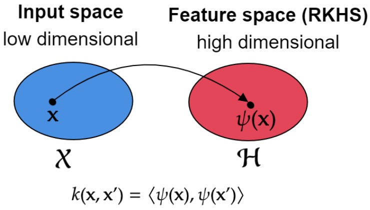
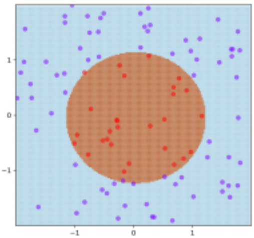


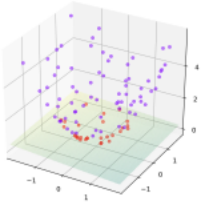
Figure: Reproducing Kernel Hilbert Space.

Feature Maps in Function-space View

For example: consider data $\mathbf{x} \in \mathbb{R}^2$.

Feature map	Kernel
$\psi(\mathbf{x}) = (x_1, x_2, x_1^2 + x_2^2)$	$k(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}' + \lVert\mathbf{x}\rVert_2^2 \lVert\mathbf{x}'\rVert_2^2$





Some Popular Kernels

- **Polynomial:** $k(\mathbf{x}, \mathbf{x}') = (\gamma \mathbf{x}^\mathsf{T} \mathbf{x}' + c)^q$.
- **Exponential:** $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2}{\sigma}\right)$.
- **Gaussian:** $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2\sigma^2}\right)$.
- **Matérn:** $k(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell}\|\mathbf{x}-\mathbf{x}'\|_2\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}}{\ell}\|\mathbf{x}-\mathbf{x}'\|_2\right)$.
- **Periodic kernel:** $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{2\sin^2(\pi\|\mathbf{x}-\mathbf{x}'\|_2/p)}{\sigma^2}\right)$.

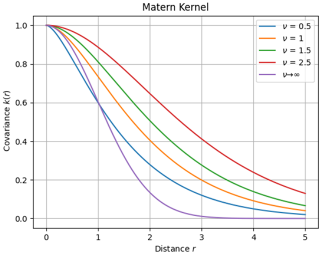


Figure: Matérn kernel.

Gaussian Processes

Definition

A *Gaussian Process* (GP) is a finite collection of random variables, which have a joint Gaussian distribution.

Gaussian Processes

Definition

A *Gaussian Process* (GP) is a finite collection of random variables, which have a joint Gaussian distribution.

Assume that $f(\mathbf{x}) \sim \mathcal{GP}\left(\mu\left(\mathbf{x}\right), k(\mathbf{x}, \mathbf{x}')\right)$, where:

- The mean is $\mu(\mathbf{x}) := \mathbb{E}\left[f(\mathbf{x})\right]$ (usually $= \mathbf{0}$).
- The covariance is:
  $k(\mathbf{x}, \mathbf{x}') := \mathbb{E}\left[\left(f(\mathbf{x}) - \mu(\mathbf{x})\right)\left(f(\mathbf{x}') - \mu(\mathbf{x}')\right)\right]$.

Gaussian Processes

Definition

A *Gaussian Process* (GP) is a finite collection of random variables, which have a joint Gaussian distribution.

Assume that $f(\mathbf{x}) \sim \mathcal{GP}\left(\mu\left(\mathbf{x}\right), k(\mathbf{x}, \mathbf{x}')\right)$, where:

- The mean is $\mu(\mathbf{x}) := \mathbb{E}\left[f(\mathbf{x})\right]$ (usually $= \mathbf{0}$).
- The covariance is: $k(\mathbf{x}, \mathbf{x}') := \mathbb{E}\left[\left(f(\mathbf{x}) - \mu(\mathbf{x})\right)\left(f(\mathbf{x}') - \mu(\mathbf{x}')\right)\right]$.

For any $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^d$, the vector $\mathbf{f} = \left[f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m)\right]^\mathsf{T}$ is Gaussian random vector with covariance matrix $\mathbf{K}(\mathbf{X}, \mathbf{X})$ defined by $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Gaussian Processes Regression

- The prior is Gaussian $\mathbf{f}|\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$.
- The log marginal-likelihood can be obtained:

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^{\mathsf{T}} \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y} - \frac{1}{2} \log |\mathbf{K} + \sigma_n^2 \mathbf{I}| - \frac{n}{2} \log 2\pi$$

- The predictive mean and variance are

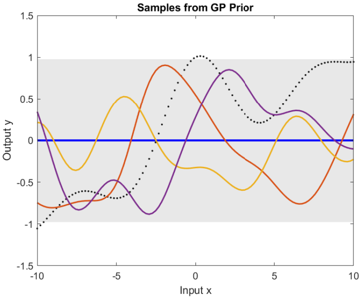
$$\mathbf{f}^{\star}(\mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{X}) \left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}_n\right)^{-1} \mathbf{y}$$

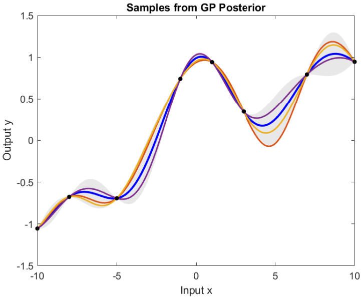
$$\mathbf{f}^{\star}_{\mathsf{var}}(\mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{x}) - \mathbf{K}(\mathbf{x}, \mathbf{X}) \left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}_n\right)^{-1} \mathbf{K}(\mathbf{x}, \mathbf{X})^{*}.$$

$O(n^3)$ is expensive!

Gaussian Processes Regression

Sampled functions from prior and posterior:





- Furher reading about GPs for machine learning [1].

[1] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT Press Cambridge, MA, 2006.

Maximum Likelihood

Typically, the kernel $k_{\boldsymbol{\theta}}$ depends on a *hyperparameters* vector $\boldsymbol{\theta}$, determined by:

$$\boldsymbol{\theta}^{\star} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}), \quad \mathcal{L}(\boldsymbol{\theta}) = \log p\left(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}\right) .$$

- Global maximum is not guaranteed (local search).
- Optimization of $\mathcal{L}(\boldsymbol{\theta})$ is **expensive**.

Feature Maps

Denote $\boldsymbol{\theta} = [\boldsymbol{\theta}_0, \sigma_f^2, \sigma_n^2]$. Our method builds feature maps for kernel families $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ that can be written as:

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \int_{\mathbb{R}^d} \varphi(\mathbf{x}, \boldsymbol{\eta})\varphi(\mathbf{x}', \boldsymbol{\eta})^* p(\boldsymbol{\eta}; \boldsymbol{\theta}_0) d\boldsymbol{\eta} + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'}$$

where for every $\boldsymbol{\theta}_0$, $p(\cdot; \boldsymbol{\theta}_0)$ is an even on $\mathbb{R}^d$, and

$$\varphi(\mathbf{x}, \boldsymbol{\eta}) = \varphi(\mathbf{x}, -\boldsymbol{\eta})^* \, \forall \boldsymbol{\eta} \in \mathbb{R}^d \, .$$

Feature Maps

Denote $\boldsymbol{\theta} = [\boldsymbol{\theta}_0, \sigma_f^2, \sigma_n^2]$. Our method builds feature maps for kernel families $\{k_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$ that can be written as:

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \int_{\mathbb{R}^d} \varphi(\mathbf{x}, \boldsymbol{\eta})\varphi(\mathbf{x}', \boldsymbol{\eta})^* p(\boldsymbol{\eta}; \boldsymbol{\theta}_0) d\boldsymbol{\eta} + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'}$$

where for every $\boldsymbol{\theta}_0$, $p(\cdot; \boldsymbol{\theta}_0)$ is an even on $\mathbb{R}^d$, and

$$\varphi(\mathbf{x}, \boldsymbol{\eta}) = \varphi(\mathbf{x}, -\boldsymbol{\eta})^* \, \forall \boldsymbol{\eta} \in \mathbb{R}^d \, .$$

Important case:

A shift-invariant kernel $k(\mathbf{x}, \mathbf{x}') = k_0(\mathbf{x} - \mathbf{x}')$ can be written as **(Bochner's theorem)**

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \int_{\mathbb{R}^d} e^{-i(\mathbf{x}-\mathbf{x}')^\mathsf{T}\boldsymbol{\eta}} p(\boldsymbol{\eta}; \boldsymbol{\theta}_0) d\boldsymbol{\eta} + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'} \, .$$

Random Fourier Features Method

For **shift-invariant kernels**:

$$\begin{aligned} k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') &= \sigma_f^2 \int_{\mathbb{R}^d} e^{-i(\mathbf{x}-\mathbf{x}')^\top \boldsymbol{\eta}} p(\boldsymbol{\eta}; \boldsymbol{\theta}_0) d\boldsymbol{\eta} + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'} \\ &\approx \frac{\sigma_f^2}{s} \sum_{j=1}^{s} e^{-i(\mathbf{x}-\mathbf{x}')^\top \boldsymbol{\eta}_j} + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'} \\ &=: \tilde{k}_{\boldsymbol{\theta}}^{(\mathsf{RFF})}(\mathbf{x}, \mathbf{x}') \,. \end{aligned}$$

RFF [2] is based on **Monte-Carlo** sampling, where $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ are sampled from $p(\boldsymbol{\eta}; \boldsymbol{\theta}_0)$.

[2] Rahimi and Recht, *Random features for large-scale kernel machines*, NeurIPS (2008).

Random Fourier Features Method

For **shift-invariant kernels**:

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \int_{\mathbb{R}^d} e^{-i(\mathbf{x}-\mathbf{x}')^\mathsf{T}\boldsymbol{\eta}} p(\boldsymbol{\eta}; \boldsymbol{\theta}_0) d\boldsymbol{\eta} + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'}$$
$$\approx \frac{\sigma_f^2}{s} \sum_{j=1}^{s} e^{-i(\mathbf{x}-\mathbf{x}')^\mathsf{T}\boldsymbol{\eta}_j} + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'}$$
$$=: \tilde{k}_{\boldsymbol{\theta}}^{(\mathsf{RFF})}(\mathbf{x}, \mathbf{x}') \,.$$

RFF [2] is based on **Monte-Carlo** sampling, where $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ are sampled from $p(\boldsymbol{\eta}; \boldsymbol{\theta}_0)$.

- We have $k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \mathbb{E}_{\boldsymbol{\eta}_1, \ldots \boldsymbol{\eta}_s} \left[\varphi(\mathbf{x}, \cdot)^* \varphi(\mathbf{x}', \cdot)\right]$ where $\varphi(\mathbf{x}, \boldsymbol{\eta}) = \sqrt{\frac{\sigma_f^2}{s}} \left(e^{-2\pi i \boldsymbol{\eta}_1^\mathsf{T}\mathbf{x}}, \ldots, e^{-2\pi i \boldsymbol{\eta}_s^\mathsf{T}\mathbf{x}}\right)^* \,.$

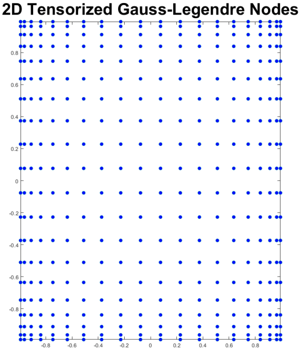
[2] Rahimi and Recht, *Random features for large-scale kernel machines*, NeurIPS (2008).

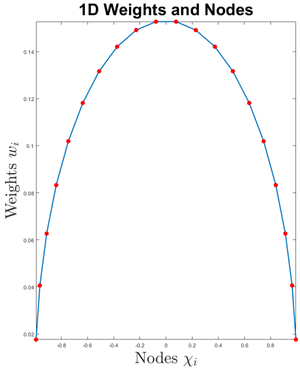
Reminder: Gauss-Legendre Quadrature

- Approximate (exactness holds for all $p \in \mathbb{P}_{2s-1}$)

$$\int_\Omega f(\boldsymbol{\eta}) p(\boldsymbol{\eta}) d\boldsymbol{\eta} \approx \sum_{j \in \{1,\ldots,s\}^d} w_j f(\boldsymbol{\eta}_j) \, .$$

- Gauss-Legendre rule (uniform pdf):
  η_j are the roots of L_s and $w_j = \frac{2}{(1-\eta_j^2)(L_s'(\eta_j))^2}$.





Gauss-Legendre Features: Integral Error

($\boldsymbol{\theta}$ is fixed).

The goal: set $\mathbf{U}$ and $\mathbf{s}$ s.t. for every dataset $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$

$$(1-\Delta)\mathbf{K} \preceq \tilde{\mathbf{K}} \preceq (1+\Delta)\mathbf{K}\,.$$

$$\Updownarrow$$

Gauss-Legendre Features: Integral Error

($\boldsymbol{\theta}$ is fixed).

The goal: set $\mathbf{U}$ and $\mathbf{s}$ s.t. for every dataset $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$

$$(1-\Delta)\mathbf{K} \preceq \tilde{\mathbf{K}} \preceq (1+\Delta)\mathbf{K}\,.$$

$$\Updownarrow$$

$$\left| \int_{\mathbb{R}^d} f(\boldsymbol{\eta}) p(\boldsymbol{\eta}; \boldsymbol{\theta}_0) d\boldsymbol{\eta} - \sum_{j=1}^{s} w_j(\boldsymbol{\theta}) f(\boldsymbol{\eta}_j) p(\boldsymbol{\eta}_j; \boldsymbol{\theta}_0) \right| \leq \frac{\Delta}{\sigma_f^2}$$

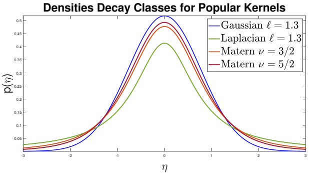
where

$$f(\boldsymbol{\eta}) = \sigma_f^2 \mathbf{z}(\boldsymbol{\eta}) \mathbf{z}(\boldsymbol{\eta})^*, \quad \mathbf{z}(\boldsymbol{\eta}) := [\varphi(\mathbf{x}_1, \boldsymbol{\eta}), \ldots, \varphi(\mathbf{x}_n, \boldsymbol{\eta})]^\mathsf{T}\ .$$

Gauss-Legendre Features for Gaussian Process Regression

1. Truncate the integral

$$\approx \int_{\Pi_{k=1}^{d}[-U_k,U_k]} f(\boldsymbol{\eta})p(\boldsymbol{\eta};\boldsymbol{\theta}_0)d\boldsymbol{\eta}$$



Faster density decay $\Leftrightarrow$ smaller $\mathbf{U}$.

2. Approximate via Gauss-Legendre quadrature

$$\approx \sum_{j=1}^{s} w_j(\boldsymbol{\theta}) f(\boldsymbol{\eta}_j) p(\boldsymbol{\eta}_j;\boldsymbol{\theta})$$

Faster Chebyshev's coefficients decay $\Leftrightarrow$ smaller s.

- Decay of Chebyshev's coefficients for analytic functions.
- For most kernels: $s = \prod_{k=1}^{d} s_k = O((\ln n)^d)$.

Truncating The Integral

We assume that p has a decay property. Consider a few decay classes of $p(\boldsymbol{\eta};\boldsymbol{\theta}_0)$ for well known kernels:

$$\begin{aligned}
\mathcal{P}_{C,\mathbf{L}} &= \left\{ p(\boldsymbol{\eta}) \le C \cdot \Pi_{k=1}^{d} \frac{1}{1+L_k^2\eta_k^2} \right\} \\
\mathcal{P}_{C,\mathbf{L}}^{(r)} &= \left\{ p(\boldsymbol{\eta}) \le C \cdot \left(1+\|\mathbf{L}\boldsymbol{\eta}\|_2^2\right)^{-r} \right\} \\
\mathcal{E}_{C,\mathbf{L}}^{(2)} &= \left\{ p(\boldsymbol{\eta}) \le C \cdot e^{-\|\mathbf{L}\boldsymbol{\eta}\|_2^2} \right\}
\end{aligned}$$

where $\mathbf{L} = \mathbf{diag}\left(L_1,\ldots,L_d\right) \ge 0$.
For each class, we set $\mathbf{U}$ s.t. the first integral error bound holds.

Truncating The Integral

$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')$	$p(\boldsymbol{\eta}; \boldsymbol{\theta}_0)$	Decay Class
Non-isotropic Laplacian $\exp(-\lVert \mathbf{L}^{-1}(\mathbf{x} - \mathbf{x}') \rVert_1)$	$\pi^{-d} \prod_{k=1}^{d} \frac{\ell_k}{1 + \ell_k^2 \eta_k^2}$ $L_k = \ell_k$	$\mathcal{P}_{C,\mathbf{L}}$ $C = \pi^{-d} \prod_{k=1}^{d} \ell_k$
Matèrn $\frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu} \lVert \mathbf{L}^{-1}(\mathbf{x} - \mathbf{x}') \rVert_2)^{\nu} \cdot K_{\nu}(\sqrt{2\nu} \lVert \mathbf{L}^{-1}(\mathbf{x} - \mathbf{x}') \rVert_2)$	$\frac{\Gamma(\nu + d/2)}{\pi^{d/2} \Gamma(\nu)(2\nu)^{d/2}} \prod_{k=1}^{d} \ell_k \cdot \left(1 + \lVert \mathbf{L}\boldsymbol{\eta} \rVert_2^2\right)^{-(\nu + d/2)}$ $L_k = \ell_k / \sqrt{2\nu}$	$\mathcal{P}_{C,\mathbf{L}}^{(r)}$ $r = \nu + d/2$ $C = \frac{\Gamma(\nu + d/2)}{\Gamma(\nu)(2\pi\nu)^{d/2}} \prod_{k=1}^{d} \ell_k$
Non-isotropic Gaussian $\exp(-\lVert \mathbf{L}^{-1}(\mathbf{x} - \mathbf{x}') \rVert_2^2 / 2)$	$(2\pi)^{-d/2} \prod_{k=1}^{d} \ell_k \exp(-\lVert \mathbf{L}\boldsymbol{\eta} \rVert_2^2 / 2)$ $L_k = \ell_k$	$\mathcal{E}_{C,\mathbf{L}}^{(2)}$ $C = (2\pi)^{-d/2} \prod_{k=1}^{d} \ell_k$

Truncating The Integral

$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')$	$p(\boldsymbol{\eta}; \boldsymbol{\theta}_0)$	Decay Class
Non-isotropic Laplacian $\exp(-\Vert\mathbf{L}^{-1}(\mathbf{x}-\mathbf{x}')\Vert_1)$	$\pi^{-d}\prod_{k=1}^{d}\frac{\ell_k}{1+\ell_k^2\eta_k^2}$ $L_k = \ell_k$	$\mathcal{P}_{C,\mathbf{L}}$ $C = \pi^{-d}\prod_{k=1}^{d}\ell_k$
Matèrn $\frac{2^{1-\nu}}{\Gamma(\nu)}(\sqrt{2\nu}\Vert\mathbf{L}^{-1}(\mathbf{x}-\mathbf{x}')\Vert_2)^{\nu}\cdot K_{\nu}(\sqrt{2\nu}\Vert\mathbf{L}^{-1}(\mathbf{x}-\mathbf{x}')\Vert_2)$	$\frac{\Gamma(\nu+d/2)}{\pi^{d/2}\Gamma(\nu)(2\nu)^{d/2}}\prod_{k=1}^{d}\ell_k\cdot$ $\left(1+\Vert\mathbf{L}\boldsymbol{\eta}\Vert_2^2\right)^{-(\nu+d/2)}$ $L_k = \ell_k/\sqrt{2\nu}$	$\mathcal{P}_{C,\mathbf{L}}^{(r)}$ $r = \nu + d/2$ $C = \frac{\Gamma(\nu+d/2)}{\Gamma(\nu)(2\pi\nu)^{d/2}}\prod_{k=1}^{d}\ell_k$
Non-isotropic Gaussian $\exp(-\Vert\mathbf{L}^{-1}(\mathbf{x}-\mathbf{x}')\Vert_2^2/2)$	$(2\pi)^{-d/2}\prod_{k=1}^{d}\ell_k\exp(-\Vert\mathbf{L}\boldsymbol{\eta}\Vert_2^2/2)$ $L_k = \ell_k$	$\mathcal{E}_{C,\mathbf{L}}^{(2)}$ $C = (2\pi)^{-d/2}\prod_{k=1}^{d}\ell_k$

1. For $\mathcal{P}_{C,\mathbf{L}}$: $U_k^{(\min)} = O(\cot(n^{-2/d}))$.
2. For $\mathcal{P}_{C,\mathbf{L}}^{(r)}$ and $r > d/2$: $U_k^{(\min)} = O(n^{2/(r-1)})$ for $d = 2$, and $O(n^{2/(2r-d)})$ otherwise.
3. For $\mathcal{E}_{C,\mathbf{L}}^{(2)}$: $U_k^{(\min)} = O(\sqrt{\ln n})$.

Spectrally Equivalent Kernel Approximations

- Scaling up GPR by $k_{\boldsymbol{\theta}} \approx \tilde{k}_{\boldsymbol{\theta}}$. How well $\tilde{k}$ approximates k?
- It depends on how much $\mathcal{N}(\boldsymbol{\mu}, \tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X}))$ is different from $\mathcal{N}(\boldsymbol{\mu}, \mathbf{K}(\mathbf{X}, \mathbf{X}))$.

Definition

We say $\tilde{\mathbf{K}}$ is *spectrally equivalent* to $\mathbf{K}$ if

$$(1 - n^{-1})\mathbf{K} \preceq \tilde{\mathbf{K}} \preceq (1 + n^{-1})\mathbf{K}\,.$$

Spectrally Equivalent Kernel Approximations

- Scaling up GPR by $k_{\boldsymbol{\theta}} \approx \tilde{k}_{\boldsymbol{\theta}}$. How well $\tilde{k}$ approximates k?
- It depends on how much $\mathcal{N}(\boldsymbol{\mu}, \tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X}))$ is different from $\mathcal{N}(\boldsymbol{\mu}, \mathbf{K}(\mathbf{X}, \mathbf{X}))$.

Definition

We say $\tilde{\mathbf{K}}$ is *spectrally equivalent* to $\mathbf{K}$ if

$$(1 - n^{-1})\mathbf{K} \preceq \tilde{\mathbf{K}} \preceq (1 + n^{-1})\mathbf{K}\,.$$

Lemma:

If $\tilde{\mathbf{K}}$ is *spectrally equivalent* to $\mathbf{K}$, then

$$D_{\mathbf{KL}}\left(\mathcal{N}(\boldsymbol{\mu}, \mathbf{K}), \mathcal{N}(\boldsymbol{\mu}, \tilde{\mathbf{K}})\right) \leq 1 - O(n^{-1})\,.$$

Spectrally Equivalent Kernel Approximations

- Scaling up GPR by $k_{\boldsymbol{\theta}} \approx \tilde{k}_{\boldsymbol{\theta}}$. How well $\tilde{k}$ approximates k?
- It depends on how much $\mathcal{N}(\boldsymbol{\mu}, \tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X}))$ is different from $\mathcal{N}(\boldsymbol{\mu}, \mathbf{K}(\mathbf{X}, \mathbf{X}))$.

Definition

We say $\tilde{\mathbf{K}}$ is *spectrally equivalent* to $\mathbf{K}$ if

$$(1 - n^{-1})\mathbf{K} \preceq \tilde{\mathbf{K}} \preceq (1 + n^{-1})\mathbf{K}\,.$$

Lemma:

If $\tilde{\mathbf{K}}$ is *spectrally equivalent* to $\mathbf{K}$, then

$$D_{\mathbf{KL}}\left(\mathcal{N}(\boldsymbol{\mu}, \mathbf{K}), \mathcal{N}(\boldsymbol{\mu}, \tilde{\mathbf{K}})\right) \leq 1 - O(n^{-1})\,.$$

- Use $(2n)^{-1}$ as the truncation error and as the quadrature error ($\Delta = n^{-1}$).

Efficient Gaussian Process Regression

The approximation has a low-rank structure. Given a dataset $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, let

$$\mathbf{Z} \in \mathbb{C}^{n \times s}, \quad \mathbf{Z}_{lj} := \varphi(\mathbf{x}_l, \boldsymbol{\eta}_j),$$

and

$$\mathbf{W} : \Theta \to \mathbb{R}_+^{s \times s}, \quad \mathbf{W}(\boldsymbol{\theta}) := \mathbf{diag}\left(w_1(\boldsymbol{\theta}_0), \ldots, w_s(\boldsymbol{\theta}_0)\right) .$$

Then, the approximate kernel matrix is

$$\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X}) = \sigma_f^2 \mathbf{Z}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^* + \sigma_n^2 \mathbf{I}_n .$$

Efficient Gaussian Process Regression

- **Hyperparameter learning:** for each gradient iteration of:

$$\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$$

we compute $O(ns + s^3 + sd)$ vs $O(n^3)$. We have:

Efficient Gaussian Process Regression

- **Hyperparameter learning:** for each gradient iteration of:

$$\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$$

we compute $O(ns + s^3 + sd)$ vs $O(n^3)$. We have:

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^\mathsf{T}\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\mathbf{y} - \frac{1}{2}\log\det\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X}) - \frac{n}{2}\log 2\pi$$

$$\frac{\partial\mathcal{L}}{\partial\theta_i} = -\frac{1}{2}\mathbf{Tr}\left(\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\theta_i}\right) + \frac{1}{2}\mathbf{y}^\mathsf{T}\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\theta_i}\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\mathbf{y}\,.$$

where

$$\frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\sigma_f^2} = \mathbf{Z}\mathbf{W}(\boldsymbol{\theta})\mathbf{Z}^*, \quad \frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\sigma_n^2} = \mathbf{I}_n, \quad \frac{\partial\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})}{\partial\theta_i} = \sigma_f^2\mathbf{Z}\frac{\partial\mathbf{W}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{Z}^*\,.$$

Efficient Gaussian Process Regression

- **Training:** we compute

$$\mathbf{w} := \mathbf{Z}^*\boldsymbol{\alpha} = \mathbf{W}(\boldsymbol{\theta}_0)^{-1}(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta}_0)^{-1})^{-1}\mathbf{Z}^*\mathbf{y}\,.$$

in $O(ns^2)$ vs $O(n^3)$ for computing $\boldsymbol{\alpha} := \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\mathbf{y}$.

Efficient Gaussian Process Regression

- **Training:** we compute

$$\mathbf{w} := \mathbf{Z}^*\boldsymbol{\alpha} = \mathbf{W}(\boldsymbol{\theta}_0)^{-1}(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta}_0)^{-1})^{-1}\mathbf{Z}^*\mathbf{y}\,.$$

  in $O(ns^2)$ vs $O(n^3)$ for computing $\boldsymbol{\alpha} := \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\mathbf{y}$.
- **Prediction:** Given a test set $\mathbf{x}_1^{(t)},\dots,\mathbf{x}_t^{(t)}$, let

$$\mathbf{Z}^{(t)} \in \mathbb{C}^{t\times s}, \quad \mathbf{Z}_{lj}^{(t)} := \varphi(\mathbf{x}_l^{(t)},\boldsymbol{\eta}_j)\,.$$

  We compute

$$\mathbf{y}_{\text{mean}}^{(t)} := \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}^{(t)},\mathbf{X})\boldsymbol{\alpha} = \sigma_f^2\mathbf{Z}^{(t)}\mathbf{W}(\boldsymbol{\theta})\mathbf{w}$$
$$\mathbf{Y}_{\text{var}}^{(t)} := \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}^{(t)},\mathbf{X}^{(t)}) - \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}^{(t)},\mathbf{X})\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X}^{(t)})$$

  in $O(st)$ vs $O(nt)$ for mean and $O(s^3+ts^2)$ vs $O(n^3+tn^2)$ for covariance.

Efficient Gaussian Process Regression

- **Training:** we compute

$$\mathbf{w} := \mathbf{Z}^*\boldsymbol{\alpha} = \mathbf{W}(\boldsymbol{\theta}_0)^{-1}(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta}_0)^{-1})^{-1}\mathbf{Z}^*\mathbf{y}\,.$$

  in $O(ns^2)$ vs $O(n^3)$ for computing $\boldsymbol{\alpha} := \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\mathbf{y}$.

- **Prediction:** Given a test set $\mathbf{x}_1^{(t)},\dots,\mathbf{x}_t^{(t)}$, let

$$\mathbf{Z}^{(t)} \in \mathbb{C}^{t\times s}, \quad \mathbf{Z}_{lj}^{(t)} := \varphi(\mathbf{x}_l^{(t)},\boldsymbol{\eta}_j)\,.$$

  We compute

$$\begin{aligned}
\mathbf{y}_{\text{mean}}^{(t)} &:= \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}^{(t)},\mathbf{X})\boldsymbol{\alpha} = \sigma_f^2\mathbf{Z}^{(t)}\mathbf{W}(\boldsymbol{\theta})\mathbf{w} \\
\mathbf{Y}_{\text{var}}^{(t)} &:= \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}^{(t)},\mathbf{X}^{(t)}) - \tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X}^{(t)},\mathbf{X})\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})^{-1}\tilde{\mathbf{K}}_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X}^{(t)})
\end{aligned}$$

  in $O(st)$ vs $O(nt)$ for mean and $O(s^3+ts^2)$ vs $O(n^3+tn^2)$ for covariance.

Woodbury matrix identity

$$\left(\sigma_f^2\mathbf{Z}\mathbf{W}(\boldsymbol{\theta}_0)\mathbf{Z}^* + \sigma_n^2\mathbf{I}_n\right)^{-1} = \sigma_n^{-2}\left(\mathbf{I}_n - \sigma_f^2\mathbf{Z}\left(\sigma_f^2\mathbf{Z}^*\mathbf{Z} + \sigma_n^2\mathbf{W}(\boldsymbol{\theta}_0)^{-1}\right)^{-1}\mathbf{Z}^*\right).$$

Main Competitive Method: RFF

- Consider a restricted class of **shift-invariant kernels** of the form:

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 k_0(\mathbf{L}^{-1}(\mathbf{x} - \mathbf{x}')) + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'}$$

  where $\mathbf{L} = \mathbf{diag}\,(L_1, \ldots, L_d) \geq 0$ (i.e., $\boldsymbol{\theta}_0$).

Main Competitive Method: RFF

- Consider a restricted class of **shift-invariant kernels** of the form:

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 k_0(\mathbf{L}^{-1}(\mathbf{x} - \mathbf{x}')) + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'}$$

  where $\mathbf{L} = \mathbf{diag}\,(L_1, \ldots, L_d) \geq 0$ (i.e., $\boldsymbol{\theta}_0$).
- The approximate kernel is

$$\tilde{k}_{\boldsymbol{\theta}}^{(\mathsf{RFF})}(\mathbf{x}, \mathbf{x}') = \frac{\sigma_f^2}{s} \sum_{j=1}^{s} e^{-i(\mathbf{x}-\mathbf{x}')^{\mathsf{T}} \boldsymbol{\eta}_j} + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'}$$

  with the associated matrix

$$\tilde{\mathbf{K}}_{\boldsymbol{\theta}}^{(\mathsf{RFF})}(\mathbf{X}, \mathbf{X}) = \sigma_f^2 \mathbf{Z}(\mathbf{L}) \mathbf{Z}(\mathbf{L})^* + \sigma_n^2 \mathbf{I}_n$$

  where

$$\mathbf{Z}(\mathbf{L}) = \frac{1}{\sqrt{s}} \exp\left(-i \mathbf{X} \mathbf{L}^{-1} \boldsymbol{\Sigma}\right), \quad \boldsymbol{\Sigma} = \left[\begin{array}{ccc} \boldsymbol{\eta}_1 & \ldots & \boldsymbol{\eta}_s \end{array}\right].$$

Comparison Between The Methods

Let I be the number of gradient computations for hyperparameter learning.

Table: Computational complexities (arithmetic operations) comparison between Gauss-Legendre Features and Random Fourier Features.

	Gauss-Legendre Features	Random Fourier Features
Training	$O(ns^2)$	$O(ns^2)$
Prediction	$O(st)$	$O(st)$
Hyperparameter learning	$O(ns^2+I(ns+s^3+sd))$	$O(I(ns^2+nsd^2))$

Other Types of Kernels

Semigroup kernels are defined on $\mathbb{R}^d_+$, and can be written as

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \int_{\mathbb{R}^d_+} e^{-\boldsymbol{\eta}^\mathsf{T}(\mathbf{x}+\mathbf{x}')} p(\boldsymbol{\eta}; \lambda) d\boldsymbol{\eta} + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'} \,.$$

For example, the *reciprocal semigroup kernel*:

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \prod_{k=1}^{d} \frac{\lambda}{x_k + x_k' + \lambda} + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'} \,.$$

All the methodology above can be adapted also for these types of kernels.

Hyperparameter Domain

We define a hyperparameter domain Θ for each problem, e.g.

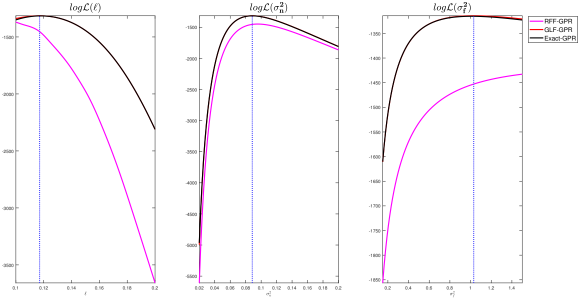
$$\Theta = \{[\ell, \sigma_n^2, \sigma_f^2] : \ell_0 \leq \ell \leq \ell_1, \sigma_{n0}^2 \leq \sigma_n^2 \leq \sigma_{n1}^2, \sigma_{f1}^2 \leq \sigma_f^2 \leq \sigma_{f0}^2\}$$

where $[\ell_0, \sigma_{f0}^2, \sigma_{n0}^2]$ is considered as the initial guess for optimization, for which we set the parameters $\mathbf{U}$ and $\mathbf{s}$.

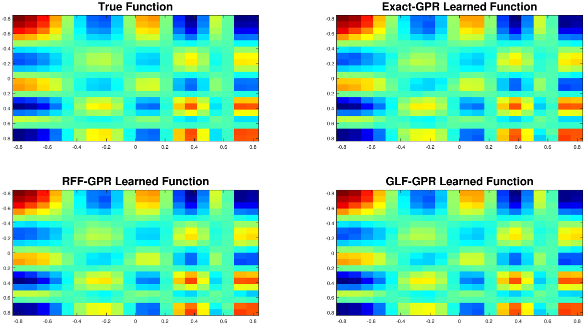
Synthetic Data

We consider $[-1,1] \times [-1,1]$ with $n = 4096$ training points and 400 test samples, generated by noisily sampling a predetermined function:

$$y_i = f^\star(\mathbf{x}_i) + \tau_i \,, \tau_i \sim \mathcal{N}(\mathbf{0}, 0.3^2\mathbf{I}_2)$$
$$f^\star(x_1, x_2) = (\sin(x_1) + \sin(10e^{x_1}))(\sin(x_2) + \sin(10e^{x_2}))\,.$$

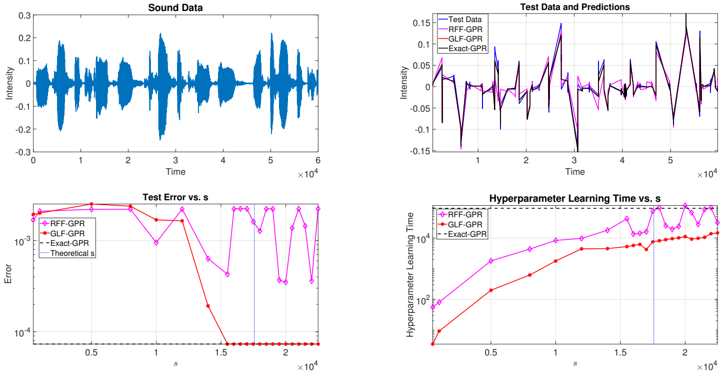


Synthetic Data



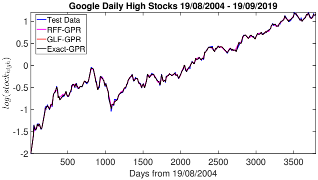
Natural Sound Modeling

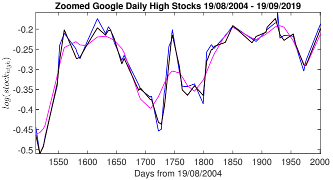
The goal is to recover contiguous missing regions in a waveform with $n = 59309$ training points. The test consists of 691 samples. The Gaussian kernel is used for learning.

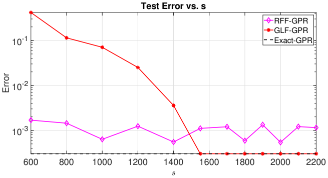


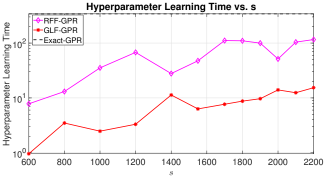
Google Daily High Stock Price

We consider a time series data of the daily high stock price of Google spanning 3797 days $19/08/2004 - 19/09/2019$. The data is $x \in \{1, \ldots, 3797\}$, $y = \log(Stock_{high})$. The test consists of 502 days. We use the Matèrn kernel with $\nu = 5/2$.



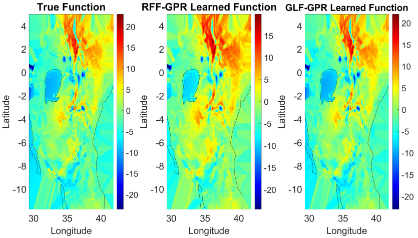


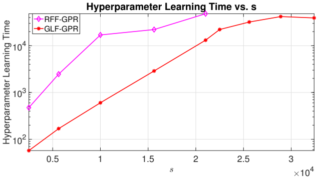




Spatial Temperature Anomaly for East Africa in 2016

We consider MOD11A2 Land Surface Temperature (LST) 8-day composite 2D data of synoptic yearly mean for 2016 in the East Africa region. Training data consists 77404 random locations $\mathbf{x} \in \{(Longitude, Latitude)\}$, $y = \{temperature\}$. The test consists of the remaining 6005 locations. We use the anisotropic Matèrn kernel with $\nu = 1$.





Thank you for your attention!

Questions?

P. Fink Shustin and Haim Avron,
Gauss-Legendre Features for Gaussian Process Regression,
Journal of Machine Learning Research (2022).