Lecture 1:
Introduction to Gaussian processes and kernel methods

Aretha Teckentrup

School of Mathematics, University of Edinburgh

KCDS Summer School 2025 - 27 August 2025

THE UNIVERSITY *of* EDINBURGH
School of Mathematics

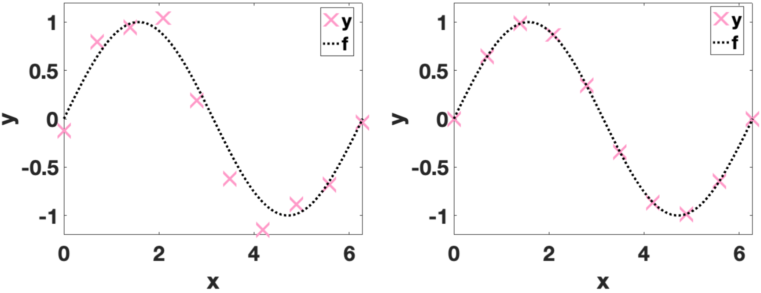
Outline

1. Problem formulation
2. Kernel methods
3. Gaussian process regression
4. Convergence analysis

Problem formulation

Interpolation and regression

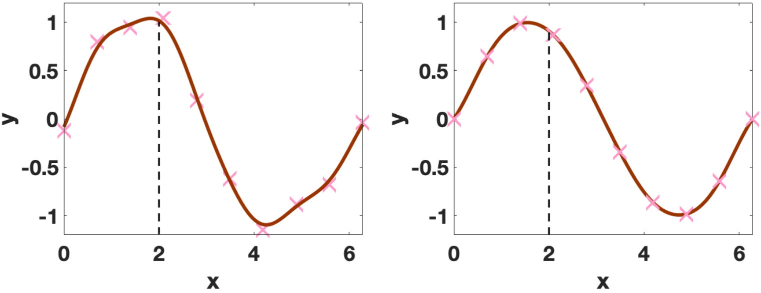
- Given N function values $y = \{\mathbf{x}^n, f(\mathbf{x}^n) + \varepsilon_n\}_{n=1}^N$, we want to learn, or approximate, the underlying function $f : D \to \mathbb{R}$, $D \subseteq \mathbb{R}^d$.
  - Real data y usually comes with noise, e.g. $\varepsilon_n \sim N(0, \delta^2)$ i.i.d..
  - Synthetic data y from computer runs is often noise-free, i.e. $\varepsilon_n \equiv 0$.



Problem formulation

Interpolation and regression

- Given N function values $y = \{\mathbf{x}^n, f(\mathbf{x}^n) + \varepsilon_n\}_{n=1}^N$, we want to learn, or approximate, the underlying function $f : D \to \mathbb{R}$, $D \subseteq \mathbb{R}^d$.
  - Real data y usually comes with noise, e.g. $\varepsilon_n \sim N(0, \delta^2)$ i.i.d..
  - Synthetic data y from computer runs is often noise-free, i.e. $\varepsilon_n \equiv 0$.



- We want to find, or *predict*, $f(\mathbf{x})$, for $\mathbf{x} \in D \setminus \{\mathbf{x}^n\}_{n=1}^N$, i.e. we want to perform regression or interpolation.

Problem formulation

Motivation and applications

This abstract framework appears in numerous disciplines and applications:

Data fitting:

- The function f linking input $\mathbf{x}$ to output y is unknown.
- Since $f(\mathbf{x}^n)$ is obtained from real-world observations, it contains measurement errors and is hence noisy.

  e.g. predicting water pollution levels in rivers, with spatial location $\mathbf{x}$ and nitrogen concentration $f(\mathbf{x})$

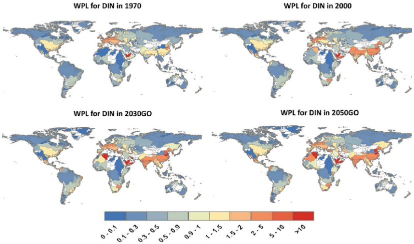


Figure taken from [Liu et al, Ecol. Indic., 2012]

Problem formulation

Motivation and applications

This abstract framework appears in numerous disciplines and applications:

Surrogate models:

- The function f linking input $\mathbf{x}$ to output y is known, but computationally very expensive to evaluate.
- Since $f(\mathbf{x}^n)$ is obtained from running computer code, it is noise-free.

e.g. parametric partial differential equations

$$-\nabla_{\mathbf{z}} \cdot (a(\mathbf{z}, \mathbf{x}) \nabla_{\mathbf{z}} u(z, \mathbf{x})) = g(\mathbf{z}), \quad \mathbf{z} \in \tilde{D}, \qquad (+\,\text{bound. cond.}),$$

$$f(\mathbf{x}) = \mathcal{F}(u(\cdot, \mathbf{x})), \qquad \text{e.g. } f(\mathbf{x}) = \|u(\cdot, \mathbf{x})\|_{L^2(\tilde{D})}.$$

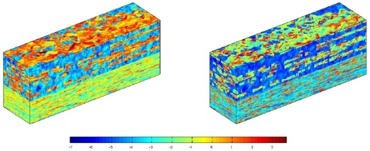


Figure taken from [Aarnes et al, Adv. Water Resour., 2005]

Kernel methods

Interpolation set-up, see e.g. [Wendland '05]

For now, I will focus on noise-free data $y = \{\mathbf{x}^n, f(\mathbf{x}^n)\}_{n=1}^N$ and interpolation. I will later discuss the extension to noisy data and regression.

Kernel methods

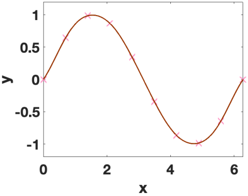
Interpolation set-up, see e.g. [Wendland '05]

For now, I will focus on noise-free data $y = \{\mathbf{x}^n, f(\mathbf{x}^n)\}_{n=1}^N$ and interpolation. I will later discuss the extension to noisy data and regression.

To approximate $f : D \to \mathbb{R}$ from y using kernel interpolation:

- we choose a kernel $k : D \times D \to \mathbb{R}$, and
- we compute, with $X_N := \{\mathbf{x}^n\}_{n=1}^N$,

$$f(\mathbf{x}) \approx s^f_{X_N,k}(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}, \mathbf{x}^n).$$



Kernel methods

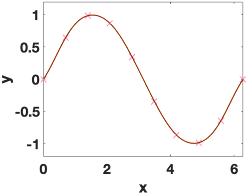
Interpolation set-up, see e.g. [Wendland '05]

For now, I will focus on noise-free data $y = \{\mathbf{x}^n, f(\mathbf{x}^n)\}_{n=1}^N$ and interpolation. I will later discuss the extension to noisy data and regression.

To approximate $f : D \to \mathbb{R}$ from y using kernel interpolation:

- we choose a kernel $k : D \times D \to \mathbb{R}$, and
- we compute, with $X_N := \{\mathbf{x}^n\}_{n=1}^N$,

$$f(\mathbf{x}) \approx s_{X_N,k}^f(\mathbf{x}) = \sum_{n=1}^N \alpha_n k(\mathbf{x}, \mathbf{x}^n).$$



- The coefficients $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^{\mathrm{T}} \in \mathbb{R}^N$ are determined by the interpolating conditions

$$f(\mathbf{x}^n) = s_{X_N,k}^f(\mathbf{x}^n), \qquad n = 1, \ldots, N.$$

A unique $\boldsymbol{\alpha}$ exists provided k is symmetric positive-definite and the interpolation points $\{\mathbf{x}^n\}_{n=1}^N$ are distinct.

Kernel methods

Interpolation set-up (ctd.), see e.g. [Wendland '05]

- Writing the interpolating conditions in vector form, we have

$$\mathbf{f}(X_N) := \begin{bmatrix} f(\mathbf{x}^1) \\ f(\mathbf{x}^2) \\ \vdots \\ f(\mathbf{x}^N) \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^N \alpha_n k(\mathbf{x}^1, \mathbf{x}^n) \\ \sum_{n=1}^N \alpha_n k(\mathbf{x}^2, \mathbf{x}^n) \\ \vdots \\ \sum_{n=1}^N \alpha_n k(\mathbf{x}^N, \mathbf{x}^n) \end{bmatrix}$$
$$= K(X_N, X_N)\,\boldsymbol{\alpha},$$

where $K(X_N, X_N) \in \mathbb{R}^{N \times N}$ is the matrix with entries $k_{ij} = k(\mathbf{x}^i, \mathbf{x}^j)$.

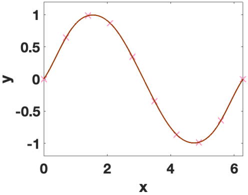
K is symmetric positive-definite provided k is symmetric positive-definite and the interpolation points $\{\mathbf{x}^n\}_{n=1}^N$ are distinct.

Kernel methods

Choice of kernel function, see e.g. [Rasmussen, Williams '06]

- The choice of kernel k is very important in practice, especially in the small N regime.

  ⇒ Behaviour in-between interpolation points!

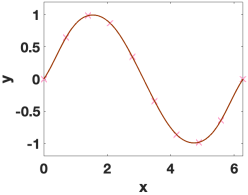


Kernel methods

Choice of kernel function, see e.g. [Rasmussen, Williams '06]

- The choice of kernel k is very important in practice, especially in the small N regime.

  $\Rightarrow$ Behaviour in-between interpolation points!



- A wide variety of kernels exists, aimed at being flexible or specialised to capture specific behaviours.
- Kernels can incorporate information about regularity, stationarity, isotropy, periodicity, amplitudes, multiple scales, . . .

Kernel methods

Matérn kernel functions, see e.g. [Porcu, Bevilacqua, Schaback, Oates '24]

- Kernels often used in applications include the Matérn kernels:

$$k_{\nu,\lambda}(\mathbf{x}, \mathbf{x}') = \frac{\sigma^2}{\Gamma(\nu) 2^{\nu-1}} \left(\frac{\|\mathbf{x} - \mathbf{x}'\|_2}{\lambda} \right)^{\nu} B_\nu \left(\frac{\|\mathbf{x} - \mathbf{x}'\|_2}{\lambda} \right),$$

with regularity parameter $\nu > 0$, lengthscale $\lambda > 0$, scaling $\sigma^2 > 0$

Special cases: $\nu = \frac{1}{2} \Rightarrow \sigma^2 \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2}{\lambda})$ and $\nu \to \infty \Rightarrow \sigma^2 \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2\lambda^2})$

Kernel methods

Matérn kernel functions, see e.g. [Porcu, Bevilacqua, Schaback, Oates '24]

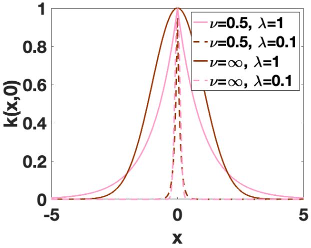
- Kernels often used in applications include the Matérn kernels:

$$k_{\nu,\lambda}(\mathbf{x},\mathbf{x}') = \frac{\sigma^2}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\|\mathbf{x}-\mathbf{x}'\|_2}{\lambda}\right)^{\nu} B_\nu \left(\frac{\|\mathbf{x}-\mathbf{x}'\|_2}{\lambda}\right),$$

  with regularity parameter $\nu > 0$, lengthscale $\lambda > 0$, scaling $\sigma^2 > 0$

  Special cases: $\nu = \frac{1}{2} \Rightarrow \sigma^2 \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2}{\lambda})$ and $\nu \to \infty \Rightarrow \sigma^2 \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2\lambda^2})$

- The kernel function $k_{\nu,\lambda}(\mathbf{x},\mathbf{x}^n)$ decays with distance $r = \|\mathbf{x}-\mathbf{x}^n\|_2$.



Kernel methods

Matérn kernel functions, see e.g. [Porcu, Bevilacqua, Schaback, Oates '24]

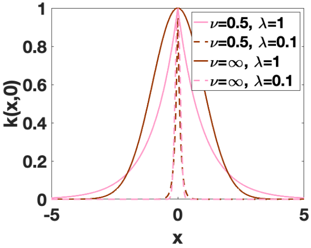
- Kernels often used in applications include the Matérn kernels:

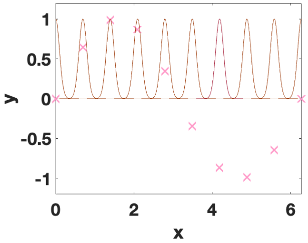
$$k_{\nu,\lambda}(\mathbf{x},\mathbf{x}') = \frac{\sigma^2}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\|\mathbf{x}-\mathbf{x}'\|_2}{\lambda}\right)^{\nu} B_\nu \left(\frac{\|\mathbf{x}-\mathbf{x}'\|_2}{\lambda}\right),$$

with regularity parameter $\nu > 0$, lengthscale $\lambda > 0$, scaling $\sigma^2 > 0$

Special cases: $\nu = \frac{1}{2} \Rightarrow \sigma^2 \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2}{\lambda})$ and $\nu \to \infty \Rightarrow \sigma^2 \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2\lambda^2})$

- The kernel function $k_{\nu,\lambda}(\mathbf{x},\mathbf{x}^n)$ decays with distance $r = \|\mathbf{x}-\mathbf{x}^n\|_2$.

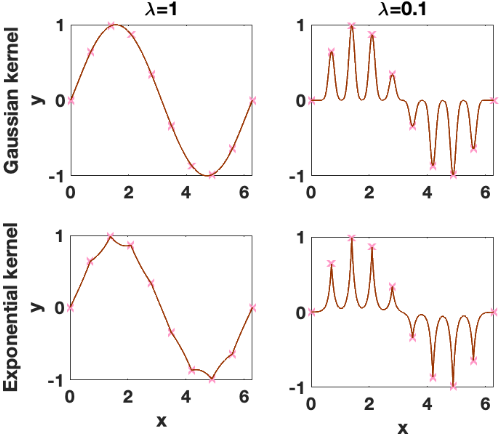




Kernel methods

Matérn kernel functions (ctd.)

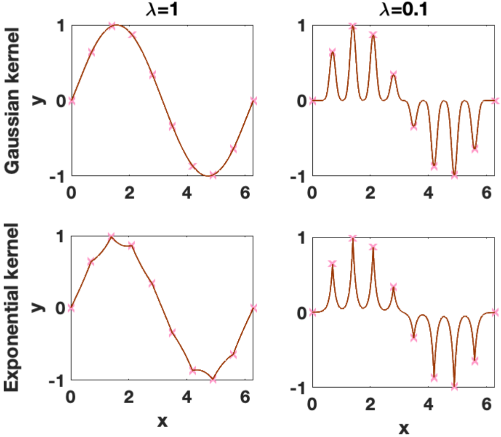
- The choice of ν and λ strongly influences the shape of $s^f_{X_N,k}$.
- But $s^f_{X_N,k}$ does not depend on σ^2, since it only scales the kernel k and the coefficients α.



Kernel methods

Matérn kernel functions (ctd.)

- The choice of ν and λ strongly influences the shape of $s^f_{X_N,k}$.
- But $s^f_{X_N,k}$ does not depend on σ^2, since it only scales the kernel k and the coefficients α.

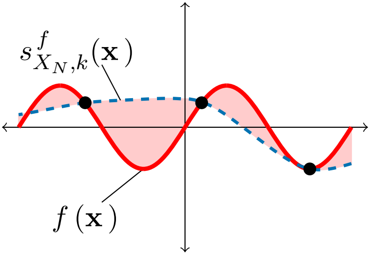


The choice of k should reflect properties of f.

Gaussian process regression

Motivation

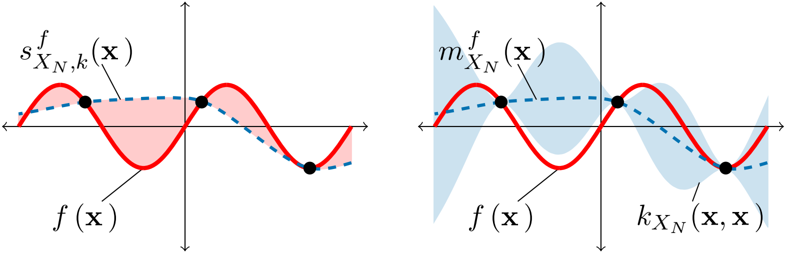
- A drawback of kernel interpolation is that it only provides an approximation $s^f_{X_N,k} \approx f$, and it does not provide a computable error estimate.



Gaussian process regression

Motivation

- A drawback of kernel interpolation is that it only provides an approximation $s^f_{X_N,k} \approx f$, and it does not provide a computable error estimate.



- Embedding the method into a Bayesian framework allows for uncertainty quantification and hence (a form of) error estimation.

Gaussian process regression

Bayesian framework, see e.g. [Rasmussen, Williams '06]

- In a Bayesian statistical framework, we place a prior distribution on the function f we want to recover.
  - This is a probability measure on a space of functions, e.g. on the space of continuous functions $C^0(D)$.
  - The prior distribution incorporates any properties of f we know, e.g. typical lengthscales, smoothness, periodicity, ...

Gaussian process regression

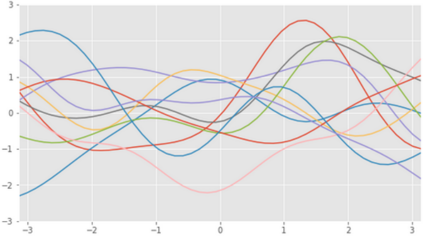
Bayesian framework, see e.g. [Rasmussen, Williams '06]

- In a Bayesian statistical framework, we place a prior distribution on the function f we want to recover.
  - This is a probability measure on a space of functions, e.g. on the space of continuous functions $C^0(D)$.
  - The prior distribution incorporates any properties of f we know, e.g. typical lengthscales, smoothness, periodicity, ...
- We obtain a posterior distribution on f (or $f|y$) by conditioning the prior distribution on the observations $y = \{\mathbf{x}^n, f(\mathbf{x}^n)\}_{n=1}^N$.
  - The posterior distribution is more *informative* than the prior distribution, i.e. more concentrated.
  - The posterior distribution may or may not be available in closed form.

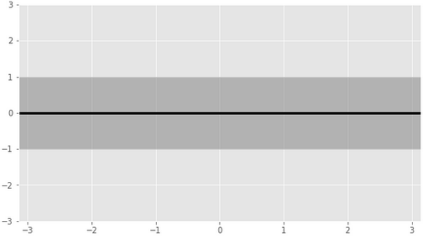
Gaussian process regression

Set-up, see e.g. [Rasmussen, Williams '06]

- Gaussian process regression is an instance of the Bayesian framework.
- We put a Gaussian process prior $\mathrm{GP}(0, k)$ on f, where we choose zero mean for ease of presentation.
  For $\{\mathbf{x}_i\}_{i=1}^m \subseteq D$, the random variables $\{f(\mathbf{x}_i)\}_{i=1}^m$ follow a joint Gaussian distribution with $\mathbb{E}[f(\mathbf{x}_i)] = 0$ and $\mathbb{C}ov[f(\mathbf{x}_i), f(\mathbf{x}_j)] = k(\mathbf{x}_i, \mathbf{x}_j)$.



Sample paths



Mean and standard deviation

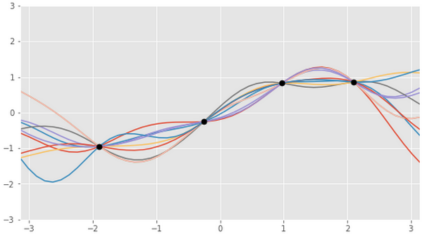
Gaussian process regression

Set-up (ctd.), see e.g. [Rasmussen, Williams '06]

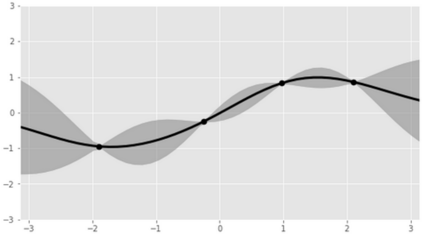
- The Gaussian process posterior $\mathrm{GP}(m^f_{X_N}, k_{X_N})$ on $f|y$ is obtained by conditioning the prior on the observed data $y = \{\mathbf{x}^n, f(\mathbf{x}^n)\}_{n=1}^N$:

$$
\begin{aligned}
m^f_{X_N}(\mathbf{x}) &= \mathbf{k}(\mathbf{x}, X_N)^\top K(X_N, X_N)^{-1} \mathbf{f}(X_N), \\
k_{X_N}(\mathbf{x}, \mathbf{x}') &= k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x}, X_N)^\top K(X_N, X_N)^{-1} \mathbf{k}(\mathbf{x}', X_N),
\end{aligned}
$$

where $\mathbf{k}(\mathbf{x}, X_N) = [k(\mathbf{x}, \mathbf{x}^1), \ldots, k(\mathbf{x}, \mathbf{x}^N)]^\top \in \mathbb{R}^N$, $K(X_N, X_N) \in \mathbb{R}^{N \times N}$ has ij^{th} entry $k(\mathbf{x}^i, \mathbf{x}^j)$, and $\mathbf{f}(X_N) = [f(\mathbf{x}^1), \ldots, f(\mathbf{x}^N)]^\top \in \mathbb{R}^N$.



Sample paths



Mean and standard deviation

Gaussian process regression

Derivation of posterior, see e.g. [Rasmussen, Williams '06]

- The form of the posterior distribution follows from the conditioning formula for Gaussian random variables.

Proposition

Suppose the n-dimensional multivariate Gaussian vector $\mathbf{Z}$ is partitioned as

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{bmatrix},$$

with $\mathbf{Z}_1$ taking values in $\mathbb{R}^{n_1}$ and $\mathbf{Z}_2$ taking values in $\mathbb{R}^{n_2}$. Writing

$$\mathbf{Z} \sim N(\boldsymbol{\mu}, \Sigma) = N\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right),$$

the conditioned random variable $\mathbf{Z}_1 | \mathbf{z}_2$ is multivariate Gaussian with

$$\mathbf{Z}_1 | \mathbf{z}_2 \sim N(\boldsymbol{\mu}_{1|2}, \Sigma_{1|2})),$$

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\boldsymbol{z}_2 - \mu_2), \qquad \Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}.$$

Gaussian process regression

Derivation of posterior, see e.g. [Rasmussen, Williams '06]

- Under the Gaussian process prior, we have by definition

$$\begin{bmatrix} \mathbf{f}(X_N) \\ f(\mathbf{x}) \end{bmatrix} := \begin{bmatrix} f(\mathbf{x}^1) \\ \vdots \\ f(\mathbf{x}^N) \\ f(\mathbf{x}) \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K(X_N, X_N) & \mathbf{k}(\mathbf{x}, X_N)^\top \\ \mathbf{k}(\mathbf{x}, X_N) & k(\mathbf{x}, \mathbf{x}) \end{bmatrix} \right).$$

- Applying the conditioning formula from the previous slide gives the Gaussianity, and the desired formulas for the mean and variance, of $f(\mathbf{x})|\mathbf{f}(X_N)$:

$$\begin{aligned} m^f_{X_N}(\mathbf{x}) &= \mathbf{k}(\mathbf{x}, X_N)^\top K(X_N, X_N)^{-1}\mathbf{f}(X_N), \\ k_{X_N}(\mathbf{x}, \mathbf{x}') &= k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x}, X_N)^\top K(X_N, X_N)^{-1}\mathbf{k}(\mathbf{x}', X_N). \end{aligned}$$

Gaussian process regression

Derivation of posterior, see e.g. [Rasmussen, Williams '06]

- Under the Gaussian process prior, we have by definition

$$\begin{bmatrix} \mathbf{f}(X_N) \\ f(\mathbf{x}) \end{bmatrix} := \begin{bmatrix} f(\mathbf{x}^1) \\ \vdots \\ f(\mathbf{x}^N) \\ f(\mathbf{x}) \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K(X_N, X_N) & \mathbf{k}(\mathbf{x}, X_N)^\top \\ \mathbf{k}(\mathbf{x}, X_N) & k(\mathbf{x}, \mathbf{x}) \end{bmatrix} \right).$$

- Applying the conditioning formula from the previous slide gives the Gaussianity, and the desired formulas for the mean and variance, of $f(\mathbf{x})|\mathbf{f}(X_N)$:

$$\begin{aligned} m^f_{X_N}(\mathbf{x}) &= \mathbf{k}(\mathbf{x}, X_N)^\top K(X_N, X_N)^{-1} \mathbf{f}(X_N), \\ k_{X_N}(\mathbf{x}, \mathbf{x}') &= k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x}, X_N)^\top K(X_N, X_N)^{-1} \mathbf{k}(\mathbf{x}', X_N). \end{aligned}$$

- Note that since $K(X_N, X_N)$ is symmetric positive-definite, we have $k_{X_N}(\mathbf{x}, \mathbf{x}) \leq k(\mathbf{x}, \mathbf{x})$, i.e. the posterior marginal variance is less than or equal to the prior marginal variance.

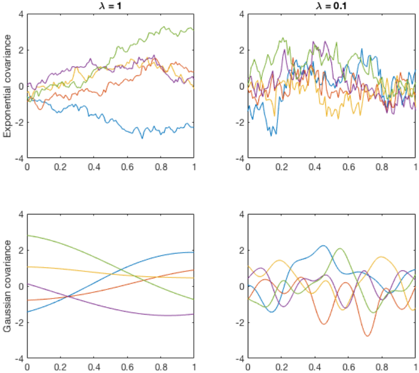
Gaussian process regression

Choice of prior distribution

- The prior $\mathrm{GP}(0, k)$ should be chosen to reflect properties of f. Assume we choose a Matérn covariance kernel.

The covariance kernel k determines properties of the Gaussian process and its sample paths:

- smoothness ν (sample path differentiability),
- amplitude σ^2 (marginal variance),
- length scales of fluctuations λ (correlation length),



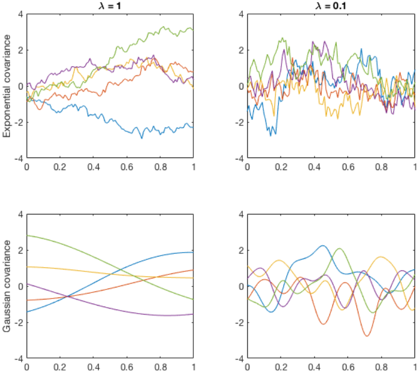
Gaussian process regression

Choice of prior distribution

- The prior $\mathrm{GP}(0, k)$ should be chosen to reflect properties of f. Assume we choose a Matérn covariance kernel.

The covariance kernel k determines properties of the Gaussian process and its sample paths:

- smoothness ν (sample path differentiability),
- amplitude σ^2 (marginal variance),
- length scales of fluctuations λ (correlation length),



- Challenge: hyper-parameters θ are usually unknown a-priori!

Gaussian process regression

Uncertainty quantification, see e.g. [Stuart, T. '18]

- The posterior mean $m^f_{X_N}$ is equal to the kernel interpolant,

$$m^f_{X_N}(\mathbf{x}) = s^f_{X_N,k}(\mathbf{x}),$$

and hence provides an approximation to f.

Gaussian process regression

Uncertainty quantification, see e.g. [Stuart, T. '18]

- The posterior mean $m^f_{X_N}$ is equal to the kernel interpolant,

$$m^f_{X_N}(\mathbf{x}) = s^f_{X_N,k}(\mathbf{x}),$$

  and hence provides an approximation to f.

- The posterior variance $k_{X_N}(\mathbf{x}, \mathbf{x})$ provides uncertainty quantification: how certain am I in the prediction of $f(\mathbf{x})$?

Gaussian process regression

Uncertainty quantification, see e.g. [Stuart, T. '18]

- The posterior mean $m^f_{X_N}$ is equal to the kernel interpolant,

$$m^f_{X_N}(\mathbf{x}) = s^f_{X_N,k}(\mathbf{x}),$$

  and hence provides an approximation to f.

- The posterior variance $k_{X_N}(\mathbf{x},\mathbf{x})$ provides uncertainty quantification: how certain am I in the prediction of $f(\mathbf{x})$?

- The posterior standard deviation is in fact the worst case error in the reproducing kernel Hilbert space[1] (RKHS) $\mathcal{H}_k$ of k:

$$\sqrt{k_{X_N}(\mathbf{x},\mathbf{x})} = \sup_{\substack{g\in\mathcal{H}_k(D)\\ \|g\|_{\mathcal{H}_k(D)}=1}} |s^g_{X_N,k}(\mathbf{x}) - g(\mathbf{x})|.$$

[1] A Hilbert space where point evaluation $g(\mathbf{x})$ is a bounded linear functional and $k(\cdot,\mathbf{x})$ is the Riesz representer.

Gaussian process regression

Uncertainty quantification (ctd.), see e.g. [Stuart, T. '18]

- The posterior standard deviation $\sqrt{k_{X_N}(\mathbf{x},\mathbf{x})}$ can hence be used to model the error in the approximation of f:

$$|s^f_{X_N,k}(\mathbf{x}) - f(\mathbf{x})| \quad \overset{?}{\approx} \sup_{\substack{g \in \mathcal{H}_k(D) \\ \|g\|_{\mathcal{H}_k(D)}=1}} |s^g_{X_n,k}(\mathbf{x}) - g(\mathbf{x})|.$$

Gaussian process regression

Uncertainty quantification (ctd.), see e.g. [Stuart, T. '18]

- The posterior standard deviation $\sqrt{k_{X_N}(\mathbf{x},\mathbf{x})}$ can hence be used to model the error in the approximation of f:

$$|s^f_{X_N,k}(\mathbf{x}) - f(\mathbf{x})| \quad \overset{?}{\approx} \sup_{\substack{g \in \mathcal{H}_k(D) \\ \|g\|_{\mathcal{H}_k(D)}=1}} |s^g_{X_n,k}(\mathbf{x}) - g(\mathbf{x})|.$$

- Including $\sqrt{k_{X_N}(\mathbf{x},\mathbf{x})}$ as an error estimate in computational pipelines can avoid over-confident and biased predictions, see e.g. [Bai, T., Zygalakis '24] for a case study in surrogate models in Bayesian inverse problems.

- Note that $\sqrt{k_{X_N}(\mathbf{x},\mathbf{x})}$ is given as part of the methodology and can be computed explicitly.

Gaussian process regression

Extension to noisy data

- Suppose we have noisy observations $y = \{\mathbf{x}^n, f(\mathbf{x}^n) + \varepsilon_n\}_{n=1}^N$, with $\varepsilon_n \sim N(0, \delta^2)$ i.i.d..
- Under the Gaussian process prior, we have by definition

$$\begin{bmatrix} f(\mathbf{x}^1) + \varepsilon_1 \\ \vdots \\ f(\mathbf{x}^N) + \varepsilon_N \\ f(\mathbf{x}) \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K(X_N, X_N) + \delta^2 \mathrm{I} & \mathbf{k}(\mathbf{x}, X_N)^\top \\ \mathbf{k}(\mathbf{x}, X_N) & k(\mathbf{x}, \mathbf{x}) \end{bmatrix} \right).$$

- Applying the conditioning formula for Gaussian random variables gives the Gaussianity, and the desired formulas for the mean and variance, of $f(\mathbf{x})|y$:

$$\begin{aligned} m^f_{X_N}(\mathbf{x}) &= \mathbf{k}(\mathbf{x}, X_N)^\top (K(X_N, X_N) + \delta^2 \mathrm{I})^{-1} y, \\ k_{X_N}(\mathbf{x}, \mathbf{x}') &= k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x}, X_N)^\top (K(X_N, X_N) + \delta^2 \mathrm{I})^{-1} \mathbf{k}(\mathbf{x}', X_N). \end{aligned}$$

Gaussian process regression

Advantages and challenges

Kernel methods offer many advantages, including:

- Flexibility and adaptation through the choice of kernel k.
- Ability to handle scattered interpolation points X_N in arbitrary dimension d, opening the possibility of experimental design.
- Providing an error estimate through the Gaussian process framework.

Gaussian process regression

Advantages and challenges

Kernel methods offer many advantages, including:

- Flexibility and adaptation through the choice of kernel k.
- Ability to handle scattered interpolation points X_N in arbitrary dimension d, opening the possibility of experimental design.
- Providing an error estimate through the Gaussian process framework.

Open challenges remain, including:

- Computational bottlenecks: solving linear systems with dense, typically ill-conditioned matrix $K(X_N, X_N)$.
- Kernel design: incorporating known structure into kernel k, and analysing the benefits.

Gaussian process regression

Advantages and challenges

Kernel methods offer many advantages, including:

- Flexibility and adaptation through the choice of kernel k.
- Ability to handle scattered interpolation points X_N in arbitrary dimension d, opening the possibility of experimental design.
- Providing an error estimate through the Gaussian process framework.

Open challenges remain, including:

- Computational bottlenecks: solving linear systems with dense, typically ill-conditioned matrix $K(X_N, X_N)$.
- Kernel design: incorporating known structure into kernel k, and analysing the benefits.

In this course, we will focus on methodology in physics-constrained and non-stationary settings.

Convergence analysis

Relation to Kernel Interpolation [T., '20], [Wendland '04]

- To prove convergence as $N \to \infty$, we can make use of results from numerical analysis. Recall: Want $m^f_{X_N} \to f$ and $k_{X_N} \to 0$.

Convergence analysis

Relation to Kernel Interpolation [T., '20], [Wendland '04]

- To prove convergence as $N \to \infty$, we can make use of results from numerical analysis. Recall: Want $m^f_{X_N} \to f$ and $k_{X_N} \to 0$.
- The posterior mean $m^f_{X_N}$ is a linear combination of kernel functions:

$$m^f_{X_N}(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}, \mathbf{x}^n), \qquad \text{for known } \alpha \in \mathbb{R}^N.$$

- We have $m^f_{X_N}(\mathbf{x}^n) = f(\mathbf{x}^n)$, for $n = 1, \dots, N$.
- The predictive mean $m^f_{X_N}$ is a kernel interpolant of f, and in the special case of isotropic kernels $k(\mathbf{x}, \mathbf{x}') = k(\|\mathbf{x} - \mathbf{x}'\|_2)$, a radial basis function interpolant.

Convergence analysis

Relation to Kernel Interpolation [T., '20], [Wendland '04]

- To prove convergence as $N \to \infty$, we can make use of results from numerical analysis. Recall: Want $m^f_{X_N} \to f$ and $k_{X_N} \to 0$.

- The posterior mean $m^f_{X_N}$ is a linear combination of kernel functions:

$$m^f_{X_N}(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}, \mathbf{x}^n), \qquad \text{for known } \alpha \in \mathbb{R}^N.$$

- We have $m^f_{X_N}(\mathbf{x}^n) = f(\mathbf{x}^n)$, for $n = 1, \dots, N$.

- The predictive mean $m^f_{X_N}$ is a kernel interpolant of f, and in the special case of isotropic kernels $k(\mathbf{x}, \mathbf{x}') = k(\|\mathbf{x} - \mathbf{x}'\|_2)$, a radial basis function interpolant.

- Convergence properties will depend on the specific choice of k.

Convergence analysis

Convergence for $k = k_{\mathrm{Mat}}$ - well-specified setting

Theorem [Arcangéli, de Silanes, Torres '12]

Let D be a Lipschitz domain that satisfies an interior cone condition. Then for any $f \in H^{\nu+d/2}(D)$ and $h_{X_N,D} \leq h_0$ sufficiently small, we have

$$\|f - m^f_{X_N}(\theta)\|_{L^2(D)} \leq C \underbrace{h_{X_N,D}^{\nu+\frac{d}{2}}}_{\substack{\text{decreasing in } N \\ \rightarrow \textit{convergence}}} \|f\|_{H^{\nu+d/2}(D)}.$$

Furthermore, $\|k_{X_N}(\theta)^{\frac{1}{2}}\|_{L^2(D)} \leq C' h^{\nu}_{X_N,D}$.

- The Sobolev space $H^{\nu+d/2}(D)$ is the reproducing kernel Hilbert space (RKHS) of k_{Mat}.

Convergence analysis

Convergence for $k = k_{\mathrm{Mat}}$ - well-specified setting

Theorem [Arcangéli, de Silanes, Torres '12]

Let D be a Lipschitz domain that satisfies an interior cone condition. Then for any $f \in H^{\nu+d/2}(D)$ and $h_{X_N,D} \leq h_0$ sufficiently small, we have

$$\|f - m^f_{X_N}(\theta)\|_{L^2(D)} \leq C \underbrace{h_{X_N,D}^{\nu+\frac{d}{2}}}_{\substack{\text{decreasing in } N \\ \to \textit{convergence}}} \|f\|_{H^{\nu+d/2}(D)}.$$

Furthermore, $\|k_{X_N}(\theta)^{\frac{1}{2}}\|_{L^2(D)} \leq C' h^{\nu}_{X_N,D}$.

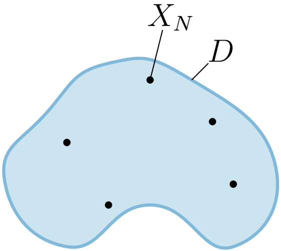
- The Sobolev space $H^{\nu+d/2}(D)$ is the reproducing kernel Hilbert space (RKHS) of k_{Mat}.
- With design points $X_N = \{\mathbf{x}^n\}_{n=1}^N$, define the fill distance

$$h_{X_N,D} = \sup_{\mathbf{x} \in D} \min_{\mathbf{x}^n \in X_N} \|\mathbf{x} - \mathbf{x}^n\|_2.$$

Convergence analysis

Convergence for $k = k_{\mathrm{Mat}}$ - well-specified setting (ctd.)

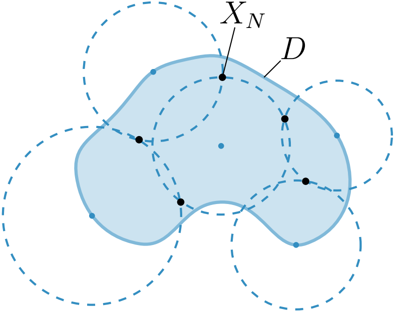
$$h_{X_N,D} \coloneqq \sup_{\mathbf{x}\in D} \inf_{\mathbf{x}^n \in X_N} \|\mathbf{x} - \mathbf{x}^n\|_2$$



Convergence analysis

Convergence for $k = k_{\mathrm{Mat}}$ - well-specified setting (ctd.)

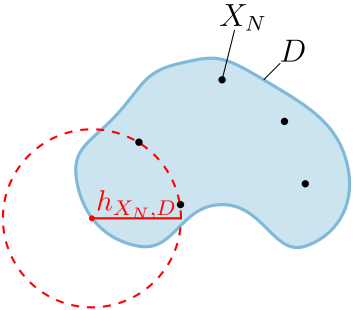
$$h_{X_N,D} \coloneqq \sup_{\mathbf{x}\in D} \inf_{\mathbf{x}^n \in X_N} \|\mathbf{x} - \mathbf{x}^n\|_2$$



Convergence analysis

Convergence for $k = k_{\mathrm{Mat}}$ - well-specified setting (ctd.)

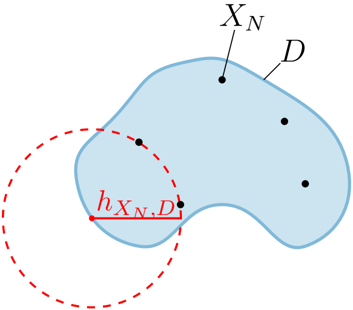
$$h_{X_N,D} := \sup_{\mathbf{x} \in D} \inf_{\mathbf{x}^n \in X_N} \|\mathbf{x} - \mathbf{x}^n\|_2$$



Convergence analysis

Convergence for $k = k_{\mathrm{Mat}}$ - well-specified setting (ctd.)

$$h_{X_N,D} := \sup_{\mathbf{x}\in D} \inf_{\mathbf{x}^n\in X_N} \|\mathbf{x} - \mathbf{x}^n\|_2$$
$$\sim N^{-\frac{1}{d}}$$



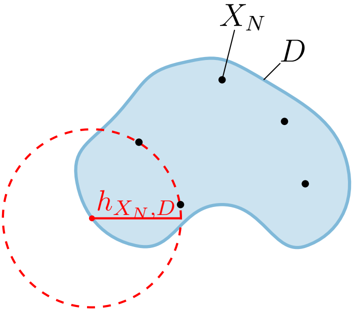
- To ensure $h_{X_N,D} \to 0$ as $N \to \infty$, we need a space-filling design.
- To obtain a fill distance $h_{X_N,D} = \varepsilon$, the number of interpolation points N needs to grow with ε^{-d}.

Convergence analysis

Convergence for $k = k_{\text{Mat}}$ - well-specified setting (ctd.)

$$h_{X_N,D} := \sup_{\mathbf{x}\in D} \inf_{\mathbf{x}^n \in X_N} \|\mathbf{x} - \mathbf{x}^n\|_2$$

$$\sim N^{-\frac{1}{d}}$$



- To ensure $h_{X_N,D} \to 0$ as $N \to \infty$, we need a space-filling design.
- To obtain a fill distance $h_{X_N,D} = \varepsilon$, the number of interpolation points N needs to grow with ε^{-d}.
- To handle high-dimensional problems, we need to assume structure in f and incorporate this structure into k and X_N.

Stationary Gaussian process regression

Convergence for $k = k_{\mathrm{Mat}}$ - misspecified setting

Theorem [Narcowich, Ward, Wendland '06] + previous theorem

Let D be a Lipschitz domain that satisfies an interior cone condition. Then for any $f \in H^\tau(D)$, with $\frac{d}{2} < \tau < \nu + \frac{d}{2}$, and $h_{X_N,D} \le h_0$ sufficiently small, we have

$$\|f - m^f_{X_N}(\theta)\|_{L^2(D)} \le C \quad \underbrace{h^\tau_{X_N,D}}_{\substack{\text{decreasing in } N \\ \to \textit{convergence}}} \quad \underbrace{\rho^{\nu+\frac{d}{2}-\tau}_{X_N}}_{\substack{\text{non-decreasing in } N \\ \to \textit{stability}}} \quad \|f\|_{H^\tau(D)}.$$

Furthermore, $\|k_{X_N}(\theta)^{\frac{1}{2}}\|_{L^2(D)} \le C' h^{\tau-\frac{d}{2}}_{X_N,D} \rho^{\nu+\frac{d}{2}-\tau}_{X_N,D}$.

- With design points $X_N = \{\mathbf{x}^n\}_{n=1}^N$, define the mesh ratio

$$\rho_{X_N,D} = \frac{\sup_{\mathbf{x}\in D} \ \min_{\mathbf{x}^n \in X_N} \ \|\mathbf{x} - \mathbf{x}^n\|_2}{\min_{n\neq m} \|\mathbf{x}^n - \mathbf{x}^m\|_2} \qquad \rho_{X_N,D} \ge 1$$

- $\rho_{X_N,D} = \text{constant}$: quasi-uniform.

Convergence analysis

Empirical Bayes'

- In a hierarchical Bayesian approach, we obtain the posterior $f|y$ as a marginal distribution of the joint posterior $f, \theta|y$. This is often intractable.

- We use an empirical Bayes' (or plug-in) approach, where we estimate values of any hyper-parameters θ from $y = \{\mathbf{x}^n, f(\mathbf{x}^n)\}_{n=1}^N$ and plug the estimate $\widehat{\theta}_N$ into the prior distribution.

Convergence analysis

Empirical Bayes'

- In a hierarchical Bayesian approach, we obtain the posterior $f|y$ as a marginal distribution of the joint posterior $f, \theta|y$. This is often intractable.

- We use an empirical Bayes' (or plug-in) approach, where we estimate values of any hyper-parameters θ from $y = \{\mathbf{x}^n, f(\mathbf{x}^n)\}_{n=1}^N$ and plug the estimate $\widehat{\theta}_N$ into the prior distribution.

- The sequence of estimates $\widehat{\theta}_N$ can be found via maximum likelihood estimation, maximum a-posteriori estimation, cross validation, ...

- Under what conditions do we get convergence for the Gaussian process posterior $\mathrm{GP}(m^f_{X_N}(\widehat{\theta}_N), k_{X_N}(\widehat{\theta}_N))$?

Convergence analysis

Convergence for $k = k_{\mathrm{Mat}}$ - estimated hyperparameters

Theorem [T. '20]

Let D be a Lipschitz domain that satisfies an interior cone condition, and for $N^* \in \mathbb{N}$ define the quantities $\nu^- := \inf_{N \geq N^*} \widehat{\nu}_N$ and $\nu^+ := \sup_{N \geq N^*} \widehat{\nu}_N$. Then for any $f \in H^{\nu^\dagger + d/2}(D)$, $h_{X_N,D} \leq h_0$ sufficiently small and $N \geq N^*$, we have

$$\|f - m^f_{X_N}(\widehat{\theta}_N)\|_{L^2(D)} \leq C \underbrace{h_{X_N,D}^{\min\{\nu^\dagger,\nu^-\}+\frac{d}{2}}}_{\substack{\text{decreasing in } N \\ \to\textit{convergence}}} \underbrace{\rho_{X_N,D}^{\max\{\nu^+-\nu^\dagger,0\}}}_{\substack{\text{non-decreasing in } N \\ \to\textit{stability}}} \|f\|_{H^{\nu^\dagger+d/2}(D)}.$$

Furthermore, $\|k_{X_N}(\widehat{\theta}_N)^{\frac{1}{2}}\|_{L^2(D)} \leq C' h_{X_N,D}^{\min\{\nu^\dagger,\nu^-\}} \rho_{X_N,D}^{\max\{\nu^+-\nu^\dagger,0\}}$.

- C, C' independent of N requires $0 < \widehat{\sigma}^2_N, \widehat{\lambda}_N, \widehat{\nu}_N < \infty$ uniformly, but we can also explicitly track dependence.
- We don't need identifiability or convergence of parameter estimates.

Convergence analysis

Convergence for $k = k_{\text{Mat}}$ - estimated hyperparameters

Theorem [T. '20]

Let D be a Lipschitz domain that satisfies an interior cone condition, and for $N^* \in \mathbb{N}$ define the quantities $\nu^- := \inf_{N \ge N^*} \widehat{\nu}_N$ and $\nu^+ := \sup_{N \ge N^*} \widehat{\nu}_N$. Then for any $f \in H^{\nu^\dagger + d/2}(D)$, $h_{X_N,D} \le h_0$ sufficiently small and $N \ge N^*$, we have

$$\|f - m^f_{X_N}(\widehat{\theta}_N)\|_{L^2(D)} \le C \underbrace{h_{X_N,D}^{\min\{\nu^\dagger,\nu^-\}+\frac{d}{2}}}_{\substack{\text{decreasing in } N \\ \to \textit{convergence}}} \underbrace{\rho_{X_N,D}^{\max\{\nu^+-\nu^\dagger,0\}}}_{\substack{\text{non-decreasing in } N \\ \to \textit{stability}}} \|f\|_{H^{\nu^\dagger+d/2}(D)}.$$

Furthermore, $\|k_{X_N}(\widehat{\theta}_N)^{\frac{1}{2}}\|_{L^2(D)} \le C' h_{X_N,D}^{\min\{\nu^\dagger,\nu^-\}} \rho_{X_N,D}^{\max\{\nu^+-\nu^\dagger,0\}}$.

- C, C' independent of N requires $0 < \widehat{\sigma}^2_N, \widehat{\lambda}_N, \widehat{\nu}_N < \infty$ uniformly, but we can also explicitly track dependence.
- We don't need identifiability or convergence of parameter estimates.
- Optimal rates $N^{-\frac{\nu+d/2}{d}}$ are obtained with $\nu^- = \nu^+ = \nu$, and with $\nu^- \ge \nu$ if the points X_N are quasi-uniform.

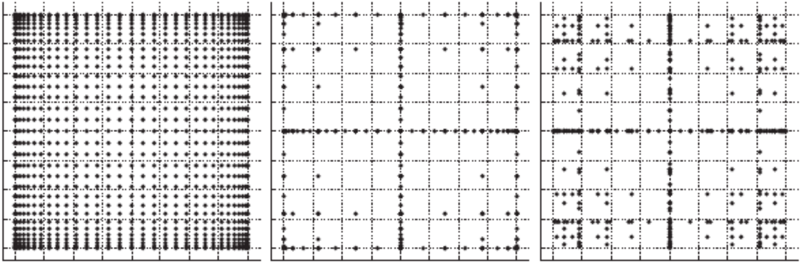
Convergence analysis

Separable Matérn kernels

- Suppose we use the family of separable Matèrn covariances

$$k_{\mathrm{sepMat}}(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^{d} k_{\mathrm{Mat}}(x_i, x_i'), \qquad D = \prod_{j=1}^{d} D_j.$$

- Suppose X_N is a Smolyak sparse grid built on nested points.



Convergence analysis

Convergence for $k = k_{\mathrm{sepMat}}$ - estimated hyperparameters

Theorem [T. '20]

With covariance kernel k_{sepMat} and sparse grid design points, under the same conditions as previous theorem, we have with $\alpha = \alpha(\nu^\dagger, \nu^+, \nu^-)$ independent of d,

$$\|f - m^f_{X_N}(\widehat{\theta}_N)\|_{L^2(D)} \leq C\, N^{-\alpha}\, (\log N)^{(1+\alpha')(d-1)} \|f\|_{\otimes_{j=1}^d H^{\nu^\dagger + d/2}(D_j)}.$$

Furthermore, $\|k_{X_N}(\widehat{\theta}_N)^{\frac{1}{2}}\|_{L^2(D)} \leq C' N^{-\alpha''}\, (\log N)^{(1+\alpha''')(d-1)}$.

- Requires dominating mixed smoothness of f. $H^1(D)$ needs $\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots$, but $\otimes_{j=1}^d H^1(D_j)$ needs $\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial^2 f}{\partial x_1 \partial x_2}, \ldots, \frac{\partial^d f}{\partial x_1 \ldots \partial x_d}$
- When $\nu_j = \nu$ and the sparse grid is based on uniform points, we have $\alpha = \frac{1}{2} + \min\{\nu^\dagger, \nu^-\}$ and $\alpha' = \min\{\nu^\dagger, \nu^-\}$, which are the rates obtained for $d = 1$ in previous theorem.

References I

E. J. Addy, J. Latz, and A. L. Teckentrup, *Lengthscale-informed sparse grids for kernel methods in high dimensions*, arXiv:2506.07797, (2025).

T. Bai, A. L. Teckentrup, and K. C. Zygalakis, *Gaussian processes for Bayesian inverse problems associated with linear partial differential equations*, Stat. Comput., 34 (2024), p. 139.

G. E. Fasshauer, F. J. Hickernell, and H. Woźniakowski, *On dimension-independent rates of convergence for function approximation with Gaussian kernels*, SIAM J. Numer. Anal., 50 (2012), pp. 247–271.

G. Finocchio and J. Schmidt-Hieber, *Posterior contraction for deep Gaussian process priors*, J. Mach. Learn. Res., 24 (2023), pp. 1–49.

M. Gnewuch, M. Hefter, A. Hinrichs, K. Ritter, and G. W. Wasilkowski, *Embeddings for infinite-dimensional integration and L^2-approximation with increasing smoothness*, J. Complexity, 54 (2019), p. 101406.

F. Nobile, R. Tempone, and S. Wolfers, *Sparse approximation of multilinear problems with applications to kernel-based methods in UQ*, Numer. Math., 139 (2018).

References II

- C. Osborne and A. L. Teckentrup, *Convergence rates of non-stationary and deep Gaussian process regression*, Found. Data Sci., (2025).

- M. Plumlee, *Fast Prediction of Deterministic Functions Using Sparse Grid Experimental Designs*, J. Am. Stat. Assoc., 109 (2014), pp. 1581–1591.

- E. Porcu, M. Bevilacqua, R. Schaback, and C. J. Oates, *The Matérn model: A journey through statistics, numerical analysis and machine learning*, Stat. Sci., 39 (2024), pp. 469–492.

- C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*, MIT press, 2006.

- A. M. Stuart and A. L. Teckentrup, *Posterior Consistency for Gaussian Process Approximations of Bayesian Posterior Distributions*, Math. Comput., 87 (2018), pp. 721–753.

- H. Wendland, *Scattered Data Approximation*, Cambridge University Press, 2005.