Gaussian Processes

Probabilistic Decision Making — Lecture 13

7th January 2026

Robert Peharz
Institute of Machine Learning and Neural Computation
Graz University of Technology

What are Gaussian Processes?

In a nutshell, **Gaussian processes (GPs)**

- generalize multivariate Gaussians to the **infinite** case
- GPs can be seen as **random real-valued functions**
- interestingly, inference (marginals, conditionals) remains tractable in GPs
- inference in GPs reduces to **inference in standard finite-dimensional Gaussians**

Recap: Inference in Gaussians

Recall the multivariate Gaussian

$$p(\boldsymbol{y} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D \, |\boldsymbol{\Sigma}|}} \, e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})}$$

with **mean vector** $\boldsymbol{\mu}$ and positive definite **covariance matrix** $\boldsymbol{\Sigma}$.

Also recall that **marginalization** and **conditioning** are the two key inference routines. Fortunately, these routines can be done analytically in Gaussians.

Marginalizing Gaussians is easy: one simply needs to discard the parameters mentioning marginalized RVs.

Example: Gaussian over 5 random variables Y_1, Y_2, Y_3, Y_4, Y_5. The marginal over Y_1, Y_4, Y_5 is Gaussian with parameters $\boldsymbol{\mu}_q$ and Σ_{qq} (q for "**q**uery") obtained by deleting rows and columns corresponding to Y_2, Y_3:

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \cancel{\mu_2} \\ \cancel{\mu_3} \\ \mu_4 \\ \mu_5 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} v_{1,1} & v_{1,2} & v_{1,3} & v_{1,4} & v_{1,5} \\ \cancel{v_{2,1}} & \cancel{v_{2,2}} & \cancel{v_{2,3}} & \cancel{v_{2,4}} & \cancel{v_{2,5}} \\ \cancel{v_{3,1}} & \cancel{v_{3,2}} & \cancel{v_{3,3}} & \cancel{v_{3,4}} & \cancel{v_{3,5}} \\ v_{4,1} & v_{4,2} & v_{4,3} & v_{4,4} & v_{4,5} \\ v_{5,1} & v_{5,2} & v_{5,3} & v_{5,4} & v_{5,5} \end{pmatrix}$$

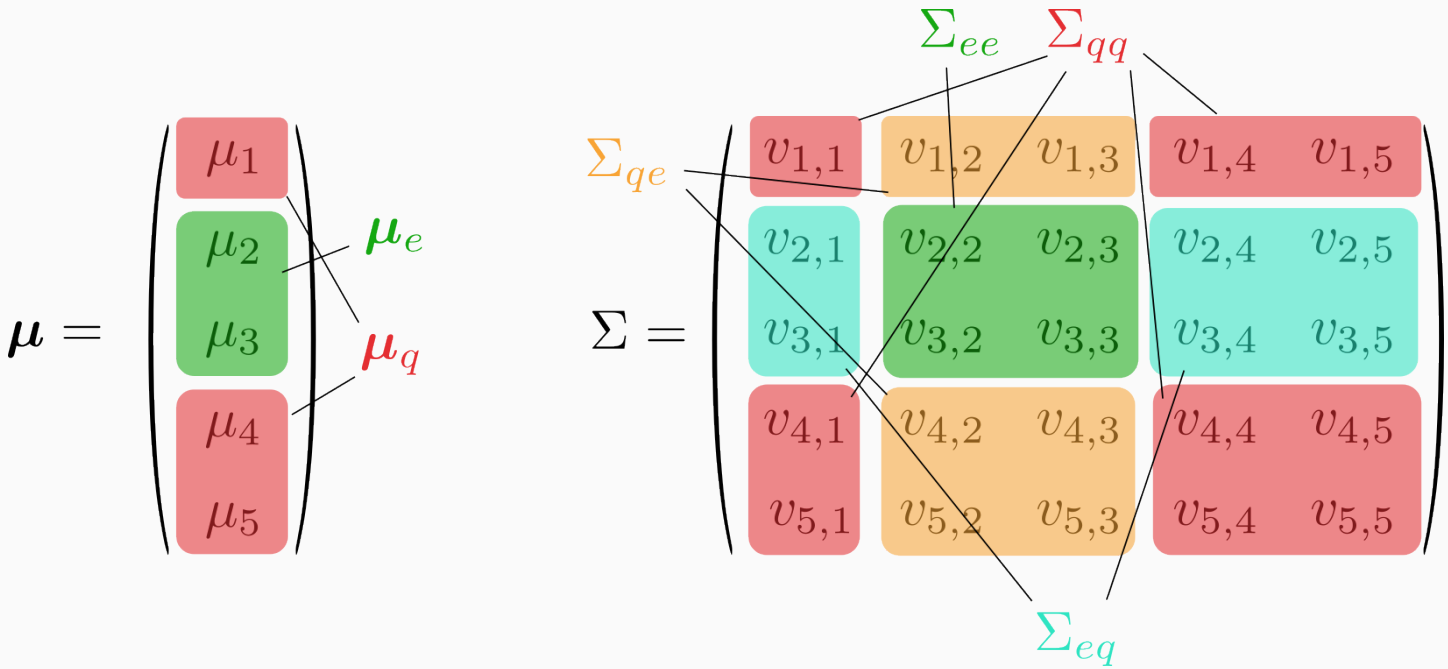
$$\boldsymbol{\mu}_q = \begin{pmatrix} \mu_1 \\ \mu_4 \\ \mu_5 \end{pmatrix} \qquad \Sigma_{qq} = \begin{pmatrix} v_{1,1} & v_{1,4} & v_{1,5} \\ v_{4,1} & v_{4,4} & v_{4,5} \\ v_{5,1} & v_{5,4} & v_{5,5} \end{pmatrix}$$

Gaussian Conditional Distributions

In Gaussians with parameters $\boldsymbol{\mu}$ and Σ, any conditional distribution $p(\boldsymbol{y}_q \mid \boldsymbol{y}_e)$ is again Gaussian with **closed form parameters**:

$$\boldsymbol{\mu}_{q|e} = \boldsymbol{\mu}_q + \Sigma_{qe}\Sigma_{ee}^{-1}(\boldsymbol{y}_e - \boldsymbol{\mu}_e)$$
$$\Sigma_{q|e} = \Sigma_{qq} - \Sigma_{qe}\Sigma_{ee}^{-1}\Sigma_{eq}$$

E.g., for **q**uery variables Y_1, Y_4, Y_5 and **e**vidence variables Y_2, Y_3:



query: Y_1, Y_4, Y_5 **e**vidence: Y_2, Y_3

$$\mu = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 1.00 & 0.61 & 0.32 & 0.01 & 0.00 \\ 0.61 & 1.00 & 0.88 & 0.14 & 0.07 \\ 0.32 & 0.88 & 1.00 & 0.32 & 0.20 \\ 0.01 & 0.14 & 0.32 & 1.00 & 0.96 \\ 0.00 & 0.07 & 0.20 & 0.96 & 1.00 \end{pmatrix}$$

$$\mu_q = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad \mu_e = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Sigma_{qq} = \begin{pmatrix} 1.00 & 0.01 & 0.00 \\ 0.01 & 1.00 & 0.96 \\ 0.00 & 0.96 & 1.00 \end{pmatrix} \quad \Sigma_{ee} = \begin{pmatrix} 1.00 & 0.88 \\ 0.88 & 1.00 \end{pmatrix} \quad \Sigma_{qe} = \begin{pmatrix} 0.61 & 0.32 \\ 0.14 & 0.32 \\ 0.07 & 0.20 \end{pmatrix} = \Sigma_{eq}^T$$

Take a concrete observation $\boldsymbol{y}_e = (1, -1)^T$, meaning $Y_2 = 1$, $Y_3 = -1$.

Then, with the parameters from the previous slide, the conditional distribution of the query variables Y_1, Y_4, Y_5 given the evidence is Gaussian with parameters

$$\boldsymbol{\mu}_{q|e} = \boldsymbol{\mu}_q + \Sigma_{qe}\Sigma_{ee}^{-1}(\boldsymbol{y}_e - \boldsymbol{\mu}_e) = \begin{pmatrix} 2.4 \\ -1.61 \\ -1.08 \end{pmatrix}$$

$$\Sigma_{q|e} = \Sigma_{qq} - \Sigma_{qe}\Sigma_{ee}^{-1}\Sigma_{eq} = \begin{pmatrix} 0.43 & 0.12 & 0.09 \\ 0.12 & 0.79 & 0.82 \\ 0.09 & 0.82 & 0.91 \end{pmatrix}$$

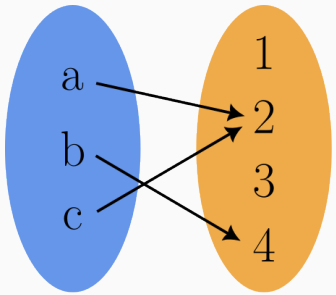
Gaussian Processes

Functions are one of **the** fundamental concepts in mathematics.

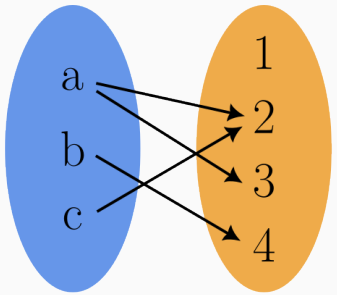
Given two **arbitrary** non-empty sets $\mathcal{X}$ (the **domain**) and $\mathcal{Y}$ (the **co-domain**), a **function**

$$f : \mathcal{X} \mapsto \mathcal{Y}$$

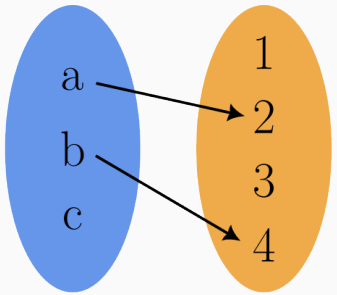
assigns to **each** element $x \in \mathcal{X}$ **exactly one** element $f(x) \in \mathcal{Y}$.



function



not a function



not a function

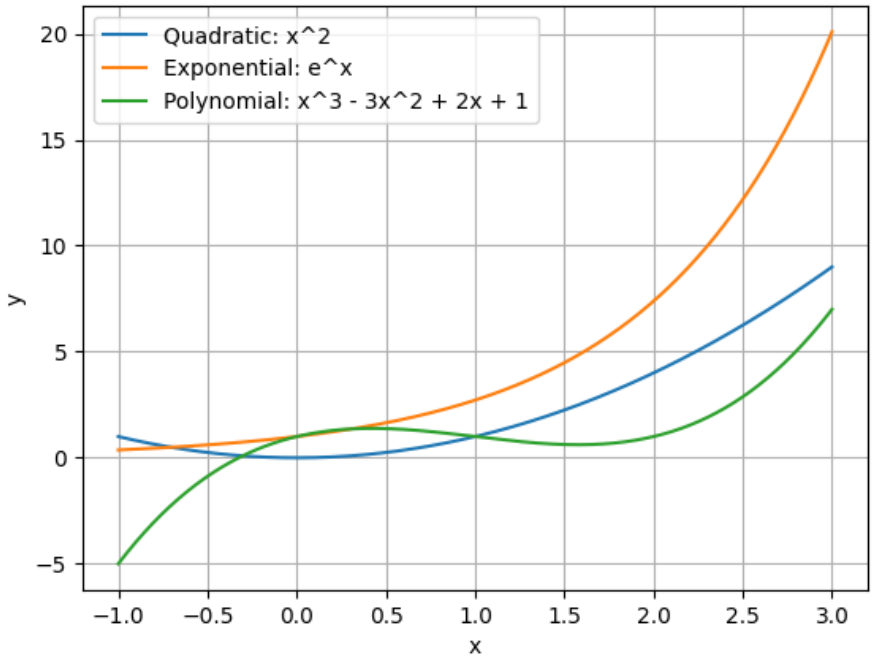
Real-Valued Functions

Very often one considers **real-valued functions** $f : \mathbb{R} \mapsto \mathbb{R}$ such as

$$f(x) = x^2$$

$$f(x) = \exp(x)$$

$$f(x) = x^3 - 3x^2 + 2x + 1$$



We will also denote **multi-variate** and **vector-valued** functions

$$f : \mathbb{R}^D \mapsto \mathbb{R}^K$$

as real-valued functions.

Finite Vectors are Functions

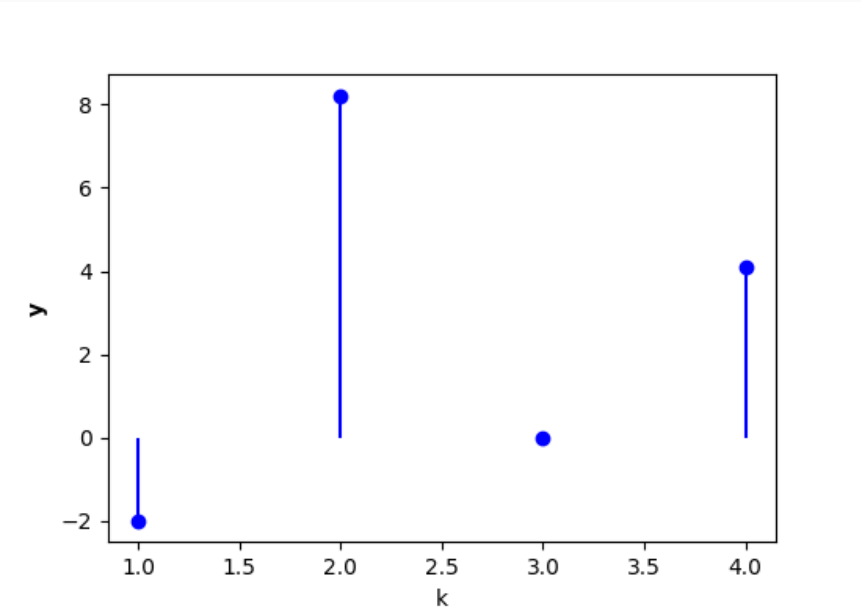
In order to understand GPs, it is helpful to re-interpret real vectors $\boldsymbol{y} \in \mathbb{R}^K$ as functions

$$\boldsymbol{y} \colon \{1, 2, \ldots, K\} \mapsto \mathbb{R},$$

mapping the integers $1 \ldots K$ to real numbers.

For example, the vector $\boldsymbol{y} = (y_1, y_2, y_3, y_4)^T = (-2, 8.2, 0, 4.1)^T$ can be seen as a function defined on $\{1, 2, 3, 4\}$:

- $\boldsymbol{y}(1) = y_1 = -2$
- $\boldsymbol{y}(2) = y_2 = 8.2$
- $\boldsymbol{y}(3) = y_3 = 0$
- $\boldsymbol{y}(4) = y_4 = 4.1$



Multivariate Gaussians as Random Functions

The multivariate Gaussian

$$p(\boldsymbol{y} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D \, |\boldsymbol{\Sigma}|}} \, e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})}$$

can be considered a **random function**:

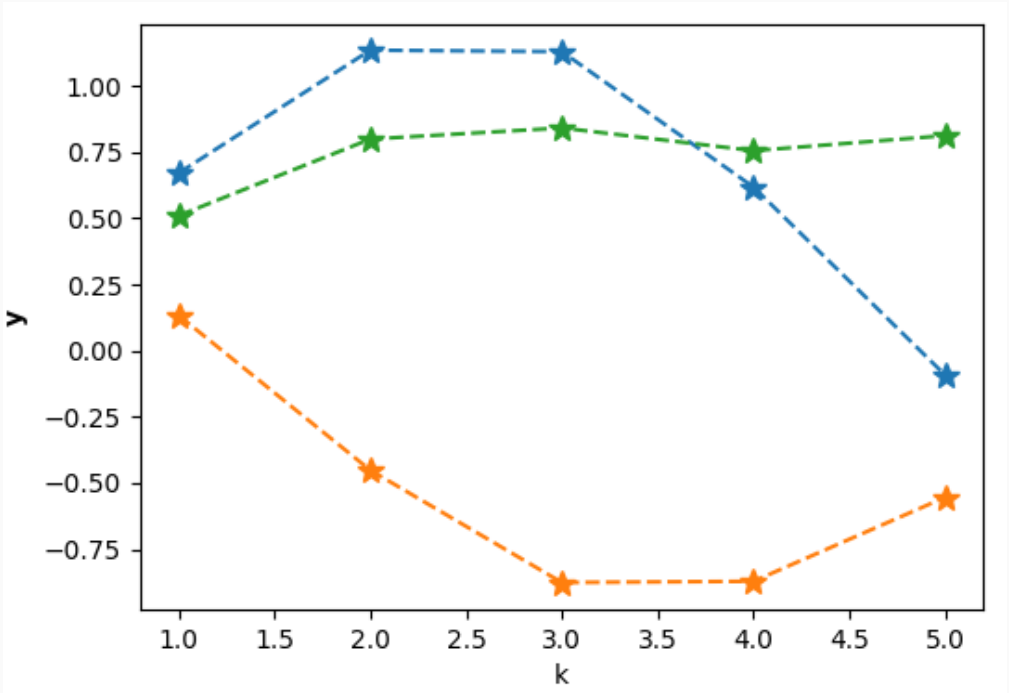
- draws from the Gaussian are K-dimensional vectors
- since K-dimensional vectors are functions defined on $\{1, 2, \ldots, K\}$, a multivariate Gaussian can be understood as a **random function** defined on $\{1, 2, \ldots, K\}$

Assume a 5-dimensional Gaussian with parameters

$$\mu = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 1.00 & 0.92 & 0.73 & 0.49 & 0.28 \\ 0.92 & 1.00 & 0.92 & 0.73 & 0.49 \\ 0.73 & 0.92 & 1.00 & 0.92 & 0.73 \\ 0.49 & 0.73 & 0.92 & 1.00 & 0.92 \\ 0.28 & 0.49 & 0.73 & 0.92 & 1.00 \end{pmatrix}$$

On the right are 3 iid draws, represented as functions.

Note: the dashed lines are just for orientation: **the functions are really defined on** $\{1, 2, 3, 4, 5\}$ **only!**



Generalization to Infinite Case?

- multivariate Gaussians are a collection of K (correlated) Gaussian RVs $\{Y_k\}_{k=1}^K$
- they represent a random function on the domain $\{1, 2, \ldots, K\}$
- in order to generalize to real functions, we need to specify **infinitly many** Gaussian RVs:

$$\{Y_x\}_{x \in \mathbb{R}},$$

  one for each $x \in \mathbb{R}$
- **note:** this is an **uncountable** set of RVs
- **note:** real functions are basically just **infinite-dimensional vectors**

A **Gaussian process** **(GP)** defined on domain $\mathcal{X}$ (e.g. $\mathcal{X}$ might be $\mathbb{R}$, $\mathbb{R}^D$, $\mathbb{C}$) is a collection of random variables

$$\{Y_x\}_{x \in \mathcal{X}},$$

such that for any **finite** subset $\{x_1, x_2, \ldots, x_K\} \subseteq \mathcal{X}$

$$\{Y_{x_k}\}_{k=1}^{K}$$

is a multivariate Gaussian.

Notes

- a finite multivariate Gaussian is a special case of GP with $\mathcal{X} = \{1, 2, \ldots, K\}$
- the existence of GPs on infinite index sets is non-trivial—but rest assured, they exist
- GPs can be defined on arbitrary domains, including $\mathbb{R}$, $\mathbb{R}^D$, $\mathbb{C}$, ...
- the finite sub-collections RVs mentioned in the definition are the **finite marginals** of the GP—**they all have a well-defined multivariate Gaussian density**
- the finite marginals must be **marginally consistent** with each other, i.e. two finite sets of RVs must have the same marginal on the set of overlapping RVs

Parameters of a Gaussian Process

Finite Gaussians are completely specified by the K-dimensional **mean vector** μ and $K \times K$ **covariance matrix** Σ.

Similarly, GPs are **completely specified** by a **mean function**

$$\mu \colon \mathcal{X} \mapsto \mathbb{R}$$

and a **covariance function** or **kernel** k:

$$k \colon \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$$

μ and k are the parameters of the GP. Note that they are **infinite-dimensional objects**.

Mean Function

- the mean function $\mu\colon \mathcal{X} \mapsto \mathbb{R}$ can be provided in any arbitrary form
  - analytic
  - linear model over non-linear feature functions
  - neural network
  - ...
- often, it is simply set to $\mathbf{0}$

Covariance Function, Kernel

The covariance function must be **positive definite**, meaning that for any finite subset $\{x_1, x_2, \ldots, x_K\} \subseteq \mathcal{X}$, the $K \times K$-matrix

$$\Sigma = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \ldots & k(x_1, x_K) \\ k(x_2, x_1) & k(x_2, x_2) & \ldots & k(x_2, x_K) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_K, x_1) & k(x_K, x_2) & \ldots & k(x_K, x_K) \end{pmatrix}$$

must be positive definite (symmetric and only positive eigen values).

Mean Function and Kernel Specify the Marginals

Recall that a GP is a (usually infinite) collection of Gaussian RV $\{Y_x\}_{x\in\mathcal{X}}$, such that for any **finite** subset is multivariate Gaussian.

The mean function and the kernel specify exactly these finite marginals.

Specifically, for inputs $\{x_1, x_2, \ldots, x_K\}$ the outputs $\{Y_{x_k}\}_{k=1}^K$ are Gaussian with parameters

$$\boldsymbol{\mu} = \begin{pmatrix} \mu(x_1) \\ \mu(x_2) \\ \vdots \\ \mu(x_K) \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} k(x_1,x_1) & k(x_1,x_2) & \ldots & k(x_1,x_K) \\ k(x_2,x_1) & k(x_2,x_2) & \ldots & k(x_2,x_K) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_K,x_1) & k(x_K,x_2) & \ldots & k(x_K,x_K) \end{pmatrix}$$

The RBF Kernel ("Squared Exponential")

The **radial basis function (RBF)** (**squared exponential**) kernel is defined as

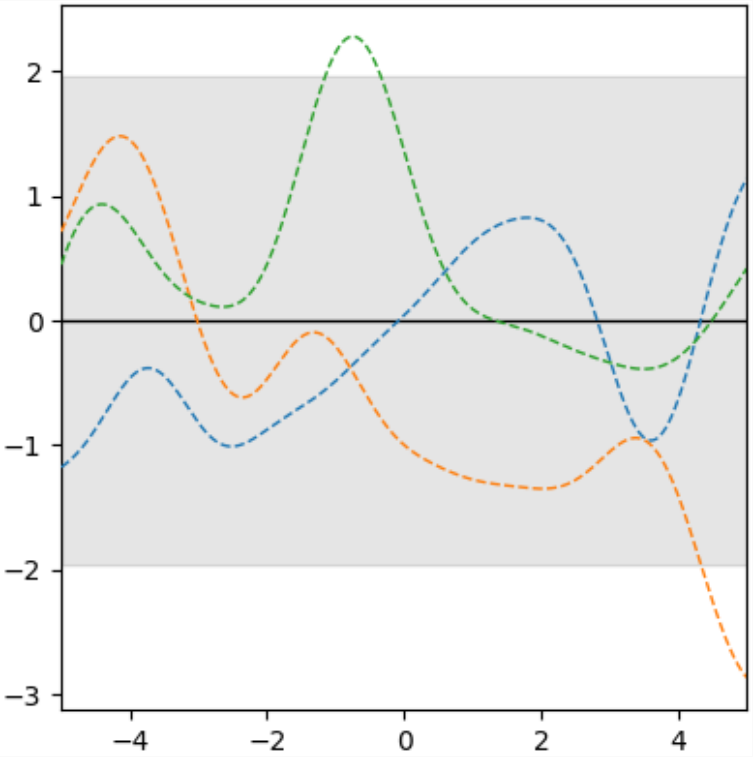
$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\ell^2}\right)$$

- $\boldsymbol{x}$, $\boldsymbol{x}'$ can be scalars or vectors
- it is positive definite
- the hyper-parameter σ controls the amplitude or **variance** of the random function
- ℓ controls the **length-scale** or rate of change of the random function
- note that it depends only on the difference $\boldsymbol{x} - \boldsymbol{x}'$; such kernels are called **stationary**

Gaussian Process with RBF Kernel

Example

- mean function (black line) is constant 0
- RBF kernel ($\sigma = 1$, $\ell = 1$)
- the gray area is the 95% confidence interval (1.96σ)
- this **"sausage plot"** represents only the (diagonal) variance at each position, not the covariance
- the dashed plots are three independent draws from the GP—**these draws are entire functions!**

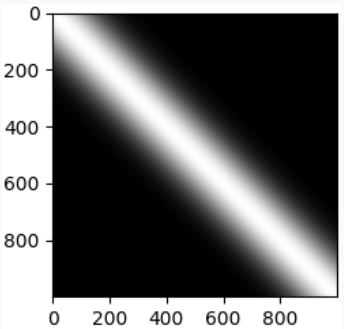


How to draw and plot GPs?

How did we draw entire functions in the previous plot?

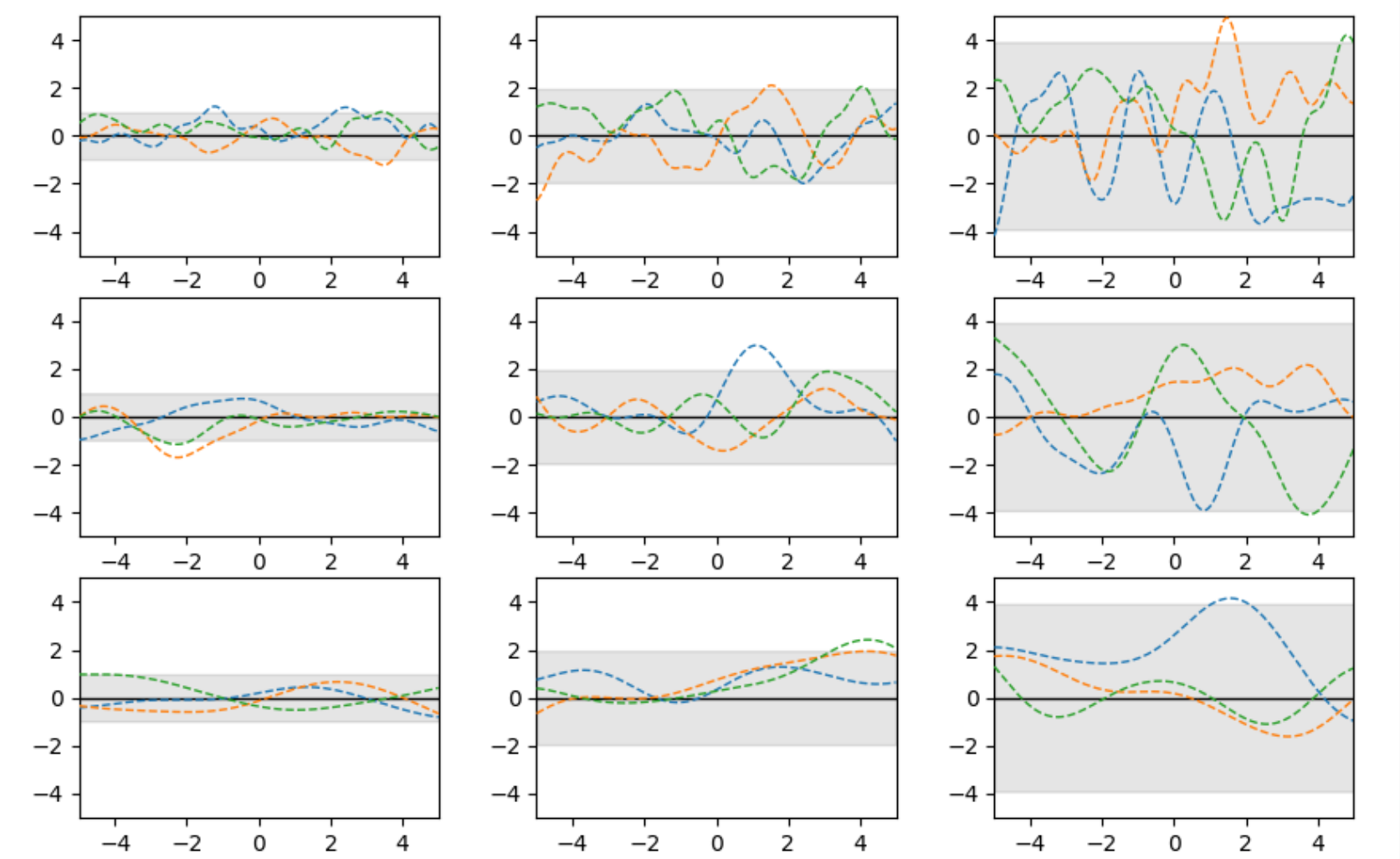
Well, we didn't—we just plotted a finely sampled finite marginal:

- we uniformly sampled thousand points $x_1, \ldots, x_{1000}$ on $[-5, 5]$
- computed the 1000×1000 covariance matrix by evaluating the RBF-kernel $\sigma^2 \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\ell^2}\right)$ on each pair x_i, x_j:



- used this covariance to draw 3 times from a multivariate Gaussian and plotted over $x_1, \ldots, x_{1000}$
- **key message:** working with GPs always reduces to multivariate Gaussians!

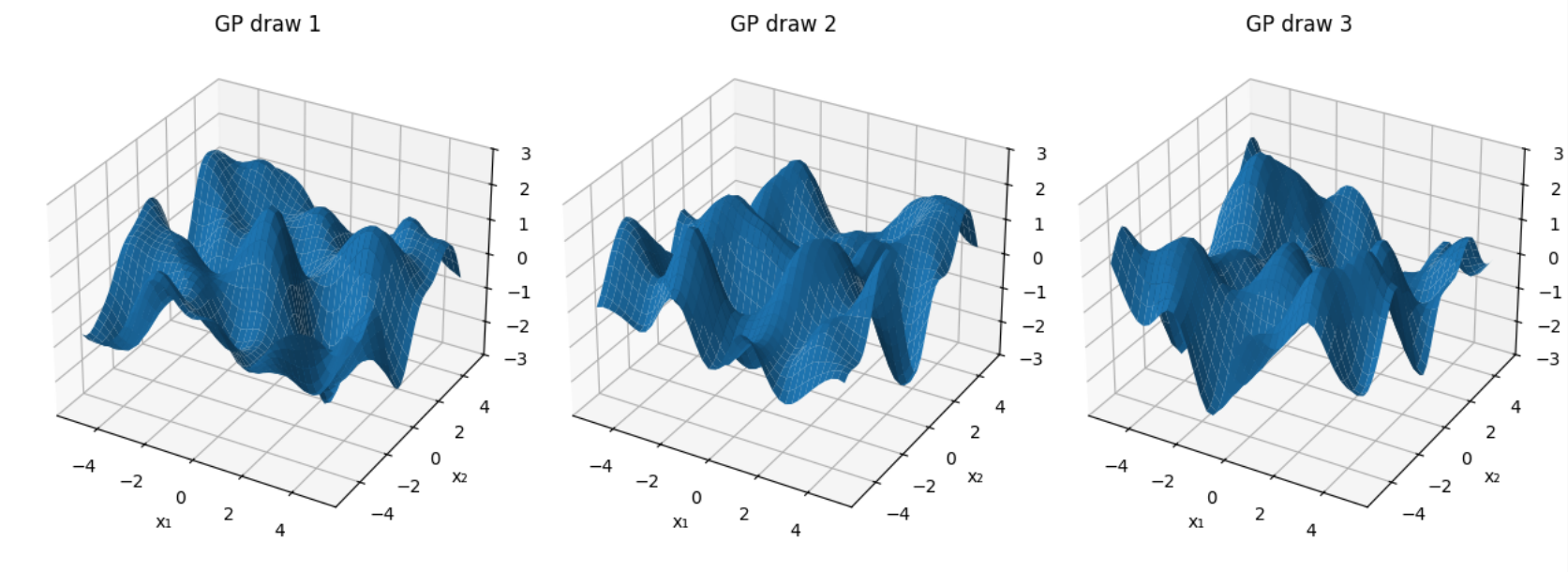
Influence of Hyperparameters



GPs with RBF kernel, with $\sigma \in \{0.5, 1, 2\}$ (left to right) and $\ell \in \{0.5, 1, 2\}$ (top to bottom)

Gaussian Process in 2D Space

Example



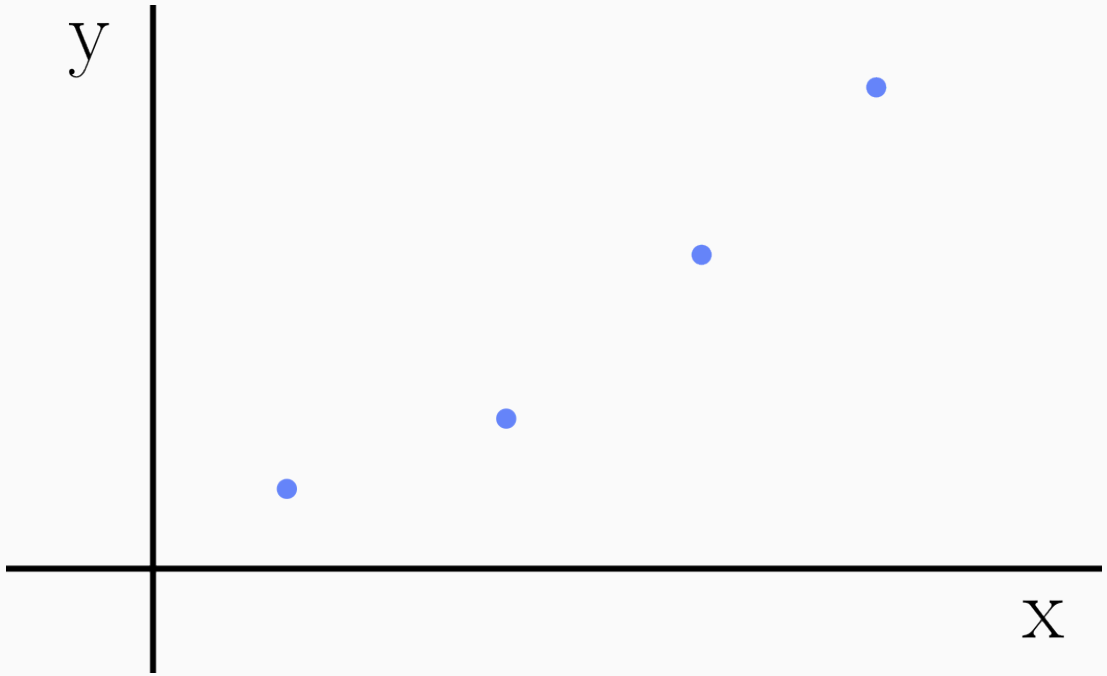
The RBF kernel is defined for any input space dimensionality. Above we see three GP draws in 2D space for $\sigma = 1$ and $\ell = 1$. These draws are random functions of the form $f : \mathbb{R}^2 \mapsto \mathbb{R}$.

For this plot, we discretized the input space x_1, x_2 with 40 points in each dimension, yielding $40 \times 40 = 1600$ points in 2D. The RBF-kernel is evaluated on each pair of points, yielding a 1600×1600 covariance matrix. I sampled from the corresponding Gaussian and plotted.

Inference, Bayesian Regression

A Regression Problem

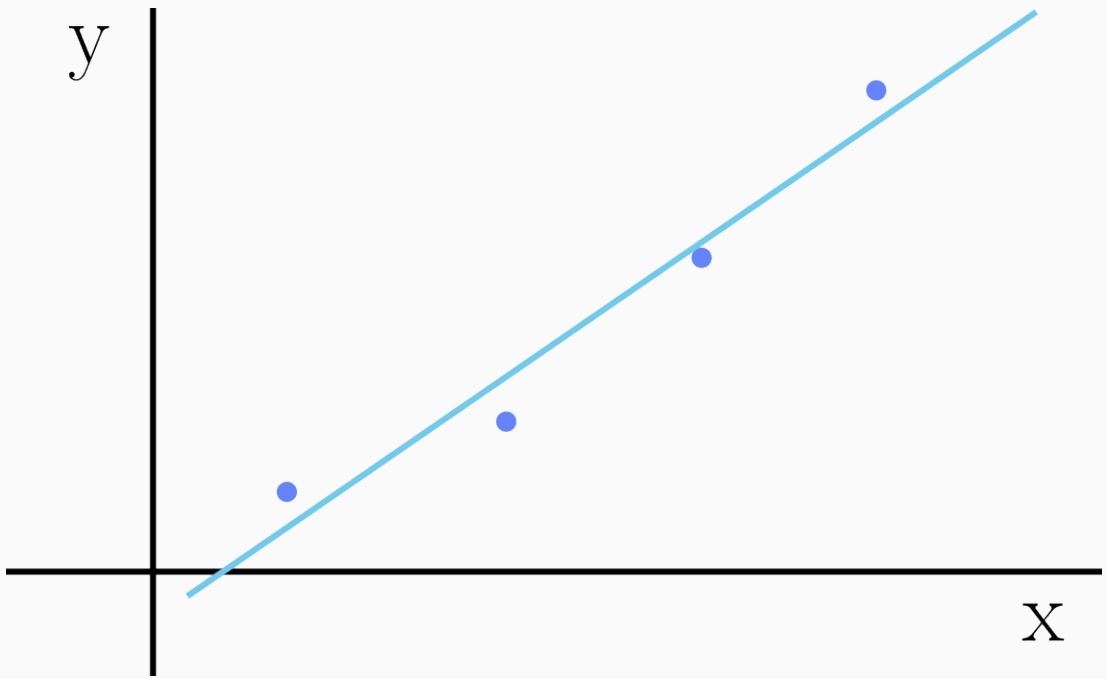
Example



Consider the regression problem above, where we aim to predict y from x. We have got 4 data points.

A Regression Problem

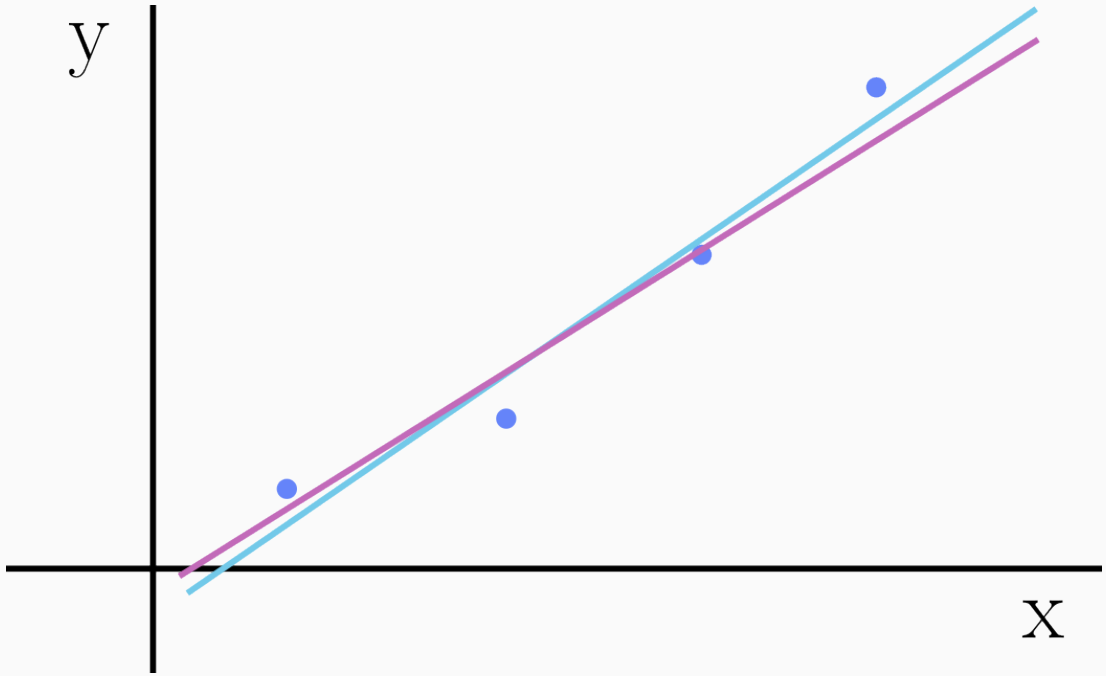
Example



We might try a linear fit, for example using a **least squares** solution.

A Regression Problem

Example

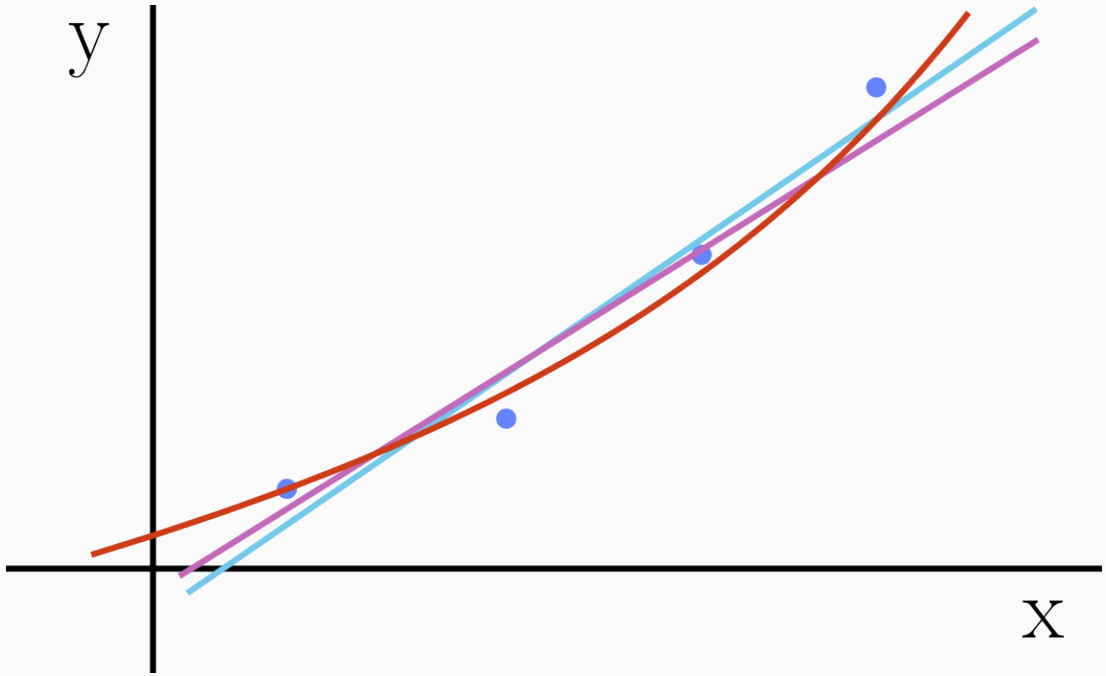


However, also other functions are good candidates, even of they are not the global optimum of the least squares objective.

After all, we just got finitely many data points. How can we be sure, what is the "correct" function?

A Regression Problem

Example



Perhaps, we should also go a for different classes of functions, like quadratic?

Take away: if we have only finitely many data, we should not overly commit to a single function.

Frequentist vs. Bayesian Regression

Let $\mathcal{F}$ be a **class of candidate functions**, mapping inputs $\boldsymbol{x}$ to output y. Examples for $\mathcal{F}$ are

- the set of **linear functions**,
- **polynomials**,
- **the set of continuous functions**,
- **the set of smooth functions**, etc.

The **frequentist approach to regression** aims to estimate a **single** "best fitting" function $f^* \in \mathcal{F}$ that explains the data $\mathcal{D}$ best.

The **Bayesian approach to regression** considers **all** functions in $\mathcal{F}$ and **performs Bayesian inference about it**.

Frequentist vs. Bayesian Regression cont'd

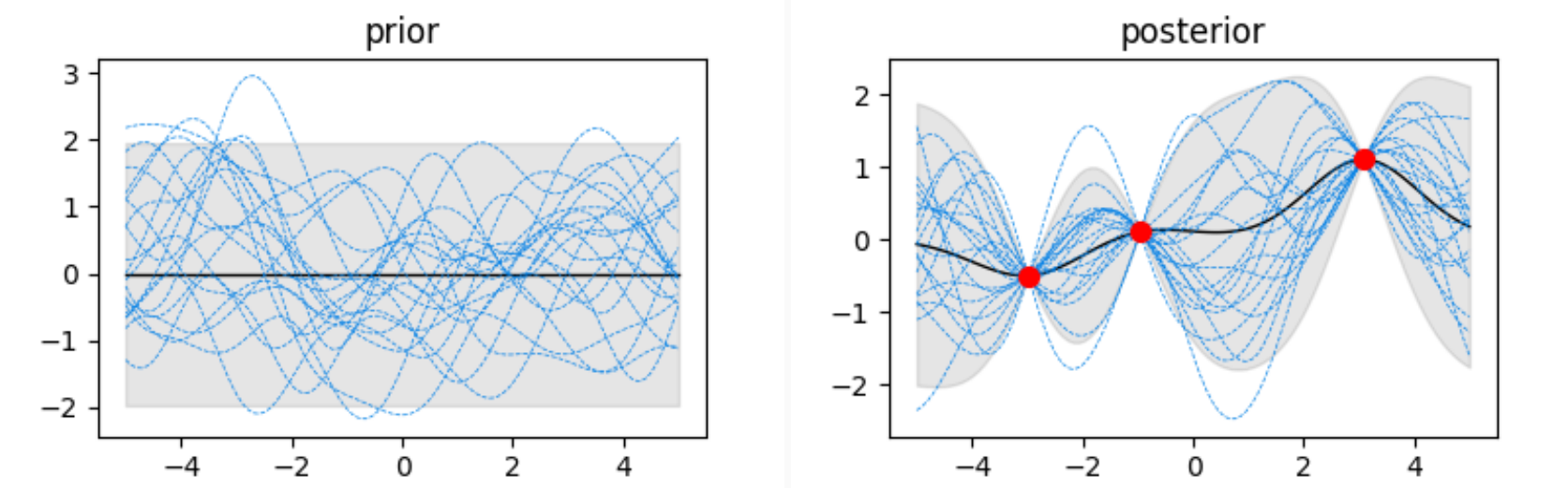
Specifically, the Bayesian approach

- puts a **prior** $p(f)$ on the function class $\mathcal{F}$
- and computes the **posterior** via Bayes law:

$$\overbrace{p(f \mid \mathcal{D})}^{\text{posterior}} = \frac{\overbrace{p(\mathcal{D} \mid f)}^{\text{likelihood}} \overbrace{p(f)}^{\text{prior}}}{p(\mathcal{D})}$$

The posterior usually does **not** assign probability 1 to a single function—hence, the Bayesian approach **considers many possible functions as solution.**

Note: writing "$p(f)$" and "$p(f \mid \mathcal{D})$" is a stretch of notation, as distributions over functions don't admit a density in the classical sense.



Bayesian regression with a Gaussian process, using an RBF kernel with $\sigma = 1$ and $\ell = 1$.

The dashed blue lines are 20 draws from prior (left) and posterior (right). The black line is the mean function and the gray area the 95% interval (diagonal of covariance).

The red dots are observed data points. Note that the posterior has ruled out functions that are not consistent with the data.

The GP Posterior

- let $\bar{\boldsymbol{x}}$ be the $N \times D$ matrix containing the **training inputs**
- let $\boldsymbol{y}$ be the N-dimensional vector containing the **training targets**
- let $\boldsymbol{x}_*$ and $\boldsymbol{x}'_*$ be two arbitrary points (**test inputs**)

For a GP prior with parameters μ and k, the posterior is again a GP with

$$\mu_{\text{post}}(\boldsymbol{x}_*) = \mu(\boldsymbol{x}_*) + k(\boldsymbol{x}_*, \bar{\boldsymbol{x}})^T k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})^{-1}(\boldsymbol{y} - \mu(\bar{\boldsymbol{x}}))$$

$$k_{\text{post}}(\boldsymbol{x}_*, \boldsymbol{x}'_*) = k(\boldsymbol{x}_*, \boldsymbol{x}'_*) - k(\boldsymbol{x}_*, \bar{\boldsymbol{x}})^T k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})^{-1} k(\bar{\boldsymbol{x}}, \boldsymbol{x}'_*)$$

- $k(\boldsymbol{x}_*, \bar{\boldsymbol{x}})$ (resp. $k(\bar{\boldsymbol{x}}, \boldsymbol{x}'_*)$) are N-dimensional vectors containing the kernel evaluation between $\boldsymbol{x}_*$ (resp. $\boldsymbol{x}'_*$) and the N training samples
- $k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})$ is the $N \times N$-matrix containing the kernel evaluation between all pairs of training inputs
- $\mu(\bar{\boldsymbol{x}})$ is the N-dimensional vector containing the evaluations of μ on the training inputs

The GP Posterior—Vectorized Version

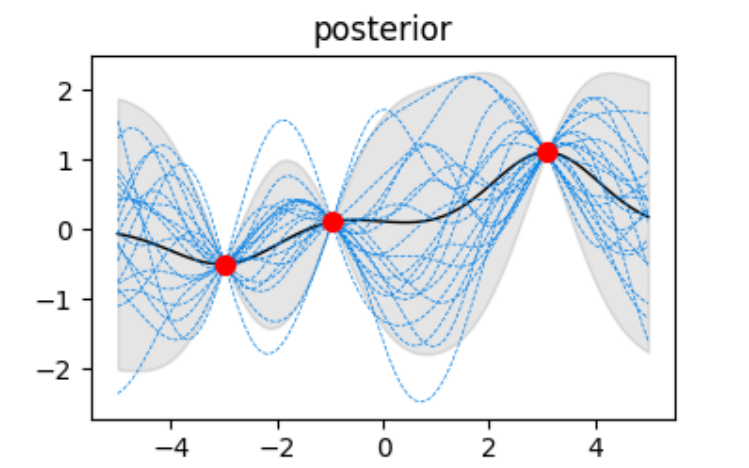
Let $\bar{\boldsymbol{x}}_*$ be a $N_* \times D$ matrix containing N_* **test inputs**. Then the mean (N_*-dimensional) and covariance ($N_* \times N_*$-matrix) evaluated on the test points are

$$\mu_{\text{post}}(\bar{\boldsymbol{x}}_*) = \mu(\bar{\boldsymbol{x}}_*) + k(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}}) k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})^{-1} (\boldsymbol{y} - \mu(\bar{\boldsymbol{x}}))$$

$$k_{\text{post}}(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}}_*) = k(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}}_*) - k(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}}) k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})^{-1} k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}}_*)$$

- $k(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}}_*)$ is the $N_* \times N_*$-matrix containing the kernel evaluation between all pairs of test inputs
- $k(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}}) = k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}}_*)^T$ is the $N_* \times N$-matrix containing all kernel evaluations between test and training inputs

$\bar{\boldsymbol{x}}_*$: 1000 points uniform from $[-5, 5]$
$\bar{\boldsymbol{x}} = (-3, -1, 3)^T$
$\boldsymbol{y} = (-0.5, 0.1, 1.1)^T$



The dimensionalities of the formula for the GP posterior are:

$$\overbrace{\mu_{\text{post}}(\bar{\boldsymbol{x}}_*)}^{1000} = \overbrace{\mu(\bar{\boldsymbol{x}}_*)}^{1000} + \overbrace{k(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}})}^{1000\times 3} \overbrace{k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})^{-1}}^{3\times 3} \overbrace{(\boldsymbol{y} - \mu(\bar{\boldsymbol{x}}))}^{3}$$

$$\underbrace{k_{\text{post}}(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}}_*)}_{1000\times 1000} = \underbrace{k(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}}_*)}_{1000\times 1000} - \underbrace{k(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}})}_{1000\times 3} \underbrace{k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})^{-1}}_{3\times 3} \underbrace{k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}}_*)}_{3\times 1000}$$

GPs and Multivariate Gaussians

Compare the GP posterior

$$\mu_{\text{post}}(\bar{\boldsymbol{x}}_*) = \mu(\bar{\boldsymbol{x}}_*) + k(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}})k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})^{-1}(\boldsymbol{y} - \mu(\bar{\boldsymbol{x}}))$$
$$k_{\text{post}}(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}}_*) = k(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}}_*) - k(\bar{\boldsymbol{x}}_*, \bar{\boldsymbol{x}})k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})^{-1}k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}}_*)$$

with the conditional Gaussian

$$\boldsymbol{\mu}_{q|e} = \boldsymbol{\mu}_q + \Sigma_{qe}\Sigma_{ee}^{-1}(\boldsymbol{y}_e - \boldsymbol{\mu}_e)$$
$$\Sigma_{q|e} = \Sigma_{qq} - \Sigma_{qe}\Sigma_{ee}^{-1}\Sigma_{eq}$$

It is really the same formula! GPs really always reduce to multivariate Gaussians. The only difference is that in GPs the mean and covariance depend on some given inputs $\boldsymbol{x}$.

More Kernels

The RBF kernel puts probability 1 on very smooth functions (infinitely often differentiable). The **Matérn** kernel allow to control for the smoothness of the modeled function.

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\, \|\boldsymbol{x} - \boldsymbol{x}'\|}{\ell} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}\, \|\boldsymbol{x} - \boldsymbol{x}'\|}{\ell} \right)$$

- Γ is the Gamma-function
- K_ν the Bessel function of the second kind
- σ is a amplitude hyper-parameter
- ℓ is a length-scale hyper-parameter
- ν controls the **roughness** of the random function

Matérn Kernel, Special Cases

The Bessel function and Gamma functions are non-trivial to evaluate. Special cases for $\nu = \{0.5, 1.5, 2.5\}$ (from top to bottom) are:

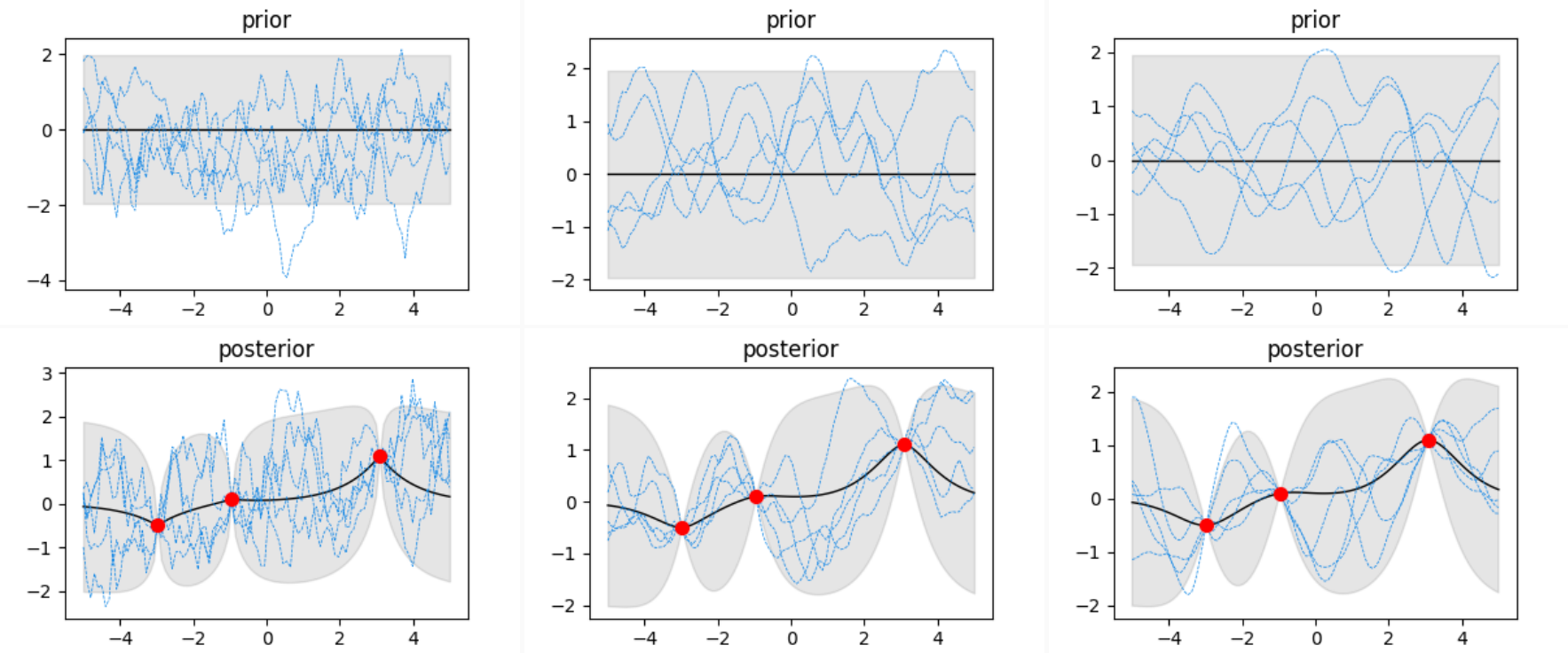
$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|}{\ell}\right)$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \left(1 + \frac{\sqrt{3}\,\|\boldsymbol{x} - \boldsymbol{x}'\|}{\ell}\right) \exp\left(-\frac{\sqrt{3}\,\|\boldsymbol{x} - \boldsymbol{x}'\|}{\ell}\right)$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \left(1 + \frac{\sqrt{5}\,\|\boldsymbol{x} - \boldsymbol{x}'\|}{\ell} + \frac{5\,\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}\,\|\boldsymbol{x} - \boldsymbol{x}'\|}{\ell}\right)$$

- $\nu = 0.5$: not differentiable random function
- $\nu = 1.5$: once differentiable random function
- $\nu = 2.5$: twice differentiable random function
- $\nu \to \infty$: RBF kernel

$\nu = 0.5$ $\nu = 1.5$ $\nu = 2.5$



$\sigma = 1$ and $\ell = 1$ for all three examples.

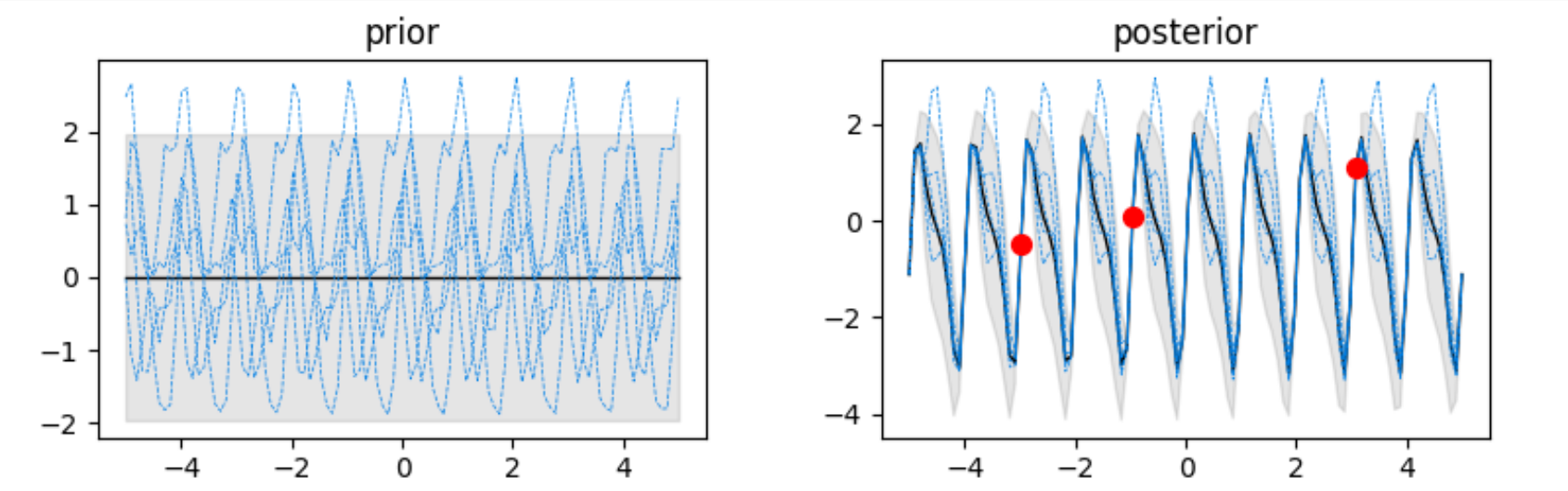
The **periodic kernel** (**sinusoidal kernel**) puts probability 1 on **periodic functions**. Hence, if we know that the function we aim to fit is periodic, it is a good choice:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-2\frac{\sin^2(\pi \|\boldsymbol{x} - \boldsymbol{x}'\|/p)}{\ell^2}\right)$$

- σ is a amplitude hyper-parameter
- ℓ is a length-scale hyper-parameter
- p is the period

Periodic Kernel

Example



$\sigma = 1$, $\ell = 1$ and $p = 1$ in this example.

Noisy Observations

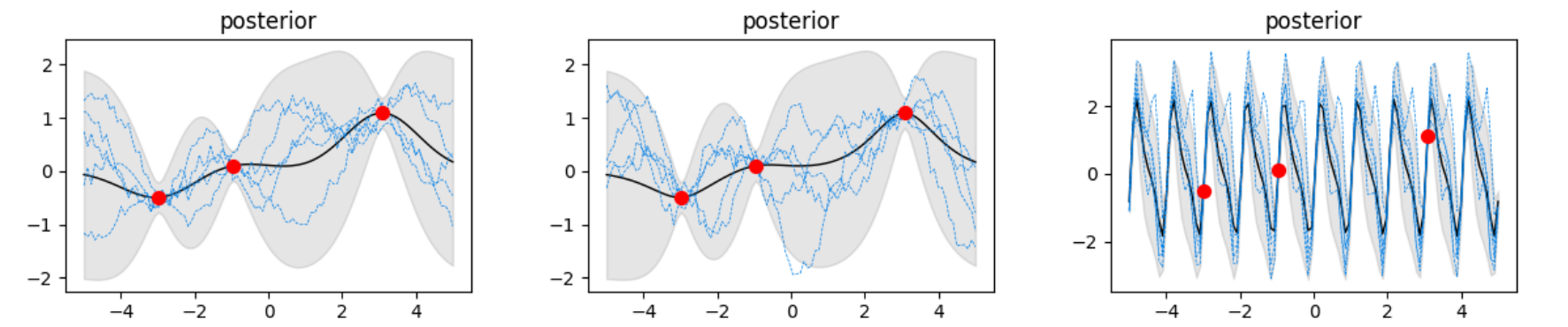
So far, we treated the noise-less case, meaning that the posterior puts all probability on functions going **exactly** through the training points.

We can incorporate white Gaussian measurement noise simply by adding a noise kernel

$$k_{\text{noise}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_{\text{n}}^2 \delta(\boldsymbol{x}, \boldsymbol{x}')$$

where δ is the **Kronecker delta**, which is 1 iff $\boldsymbol{x} = \boldsymbol{x}'$ and 0 otherwise. In practice, this just means that we add σ_{n}^2 to the diagonal of any covariance matrices.

GP posteriors with noise, using RBF kernel, Matérn kernel and Sinusoidal kernel (left to right):



How to set Hyper-Parameters

All kernels have some hyper-parameters, like **length-scales**, **amplitude/variance**, **roughness**, etc. These hyper-parameters govern the prior characteristics under the GP.

There are several ways how to set them:

- **full Bayesian approach**: equip the hyper-parameters with a **hyper-prior** and perform Bayesian inference over them (intractable, requires Monte Carlo or Variational Inference)
- optimize them with maximum likelihood: the training data just yields a finite multivariate Gaussian and we maximize this w.r.t. the hyper-parameters (**maximum likelihood type 2**)

The full Bayesian approach is cleaner, but ML type 2 is very common, oven though it should lead to some slight overfitting.

- Gaussian processes: generalizations of multivariate Gaussians to infinite case
- model random functions
- completely described by a **mean function** μ and a positive definite **covariance function** (**kernel**) k
- **Bayesian regression**: perform Bayesian inference over functions
- GP posterior is analytic and reduces to finite Gaussians