Bayesian Learning and Spatio-temporal Statistics

Part II

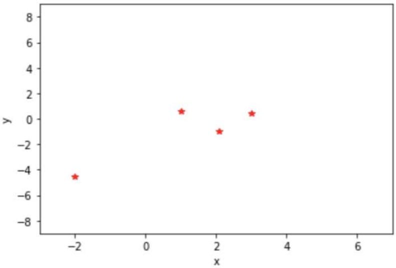
François Septier

http://www.univ-ubs.fr/septier/

francois.septier@univ-ubs.fr

Outline

Reminder: the regression problem



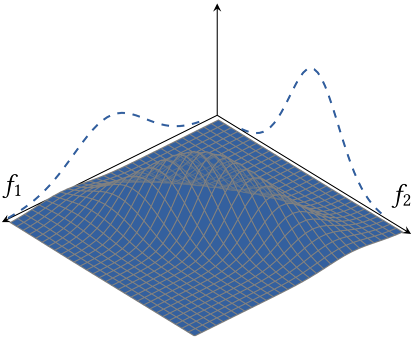
Problem Learn a model from data for how the output y depends on the input x, say $f(x)$.

We will now see what it means to use the Gaussian process as a regression model.

Outline

A binary input

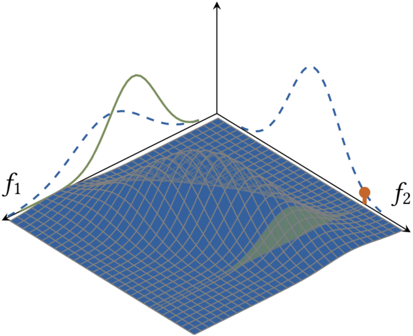
If $x \in \{1, 2\}$, we only have to find a model for $f(1)$ and $f(2)$.
Why not a multivariate normal? *(We have to estimate its parameters somehow, let's talk about that later.)*



$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix} \right)$$

A binary input

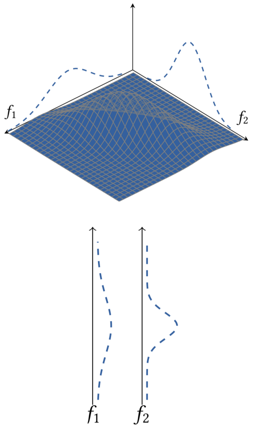
If the training data contains an observation of f_2, then our multivariate normal will automatically give us an updated prediction of f_1 as $p(f_1|f_2)$ (Thm 2, *lecture on normal distribution*)

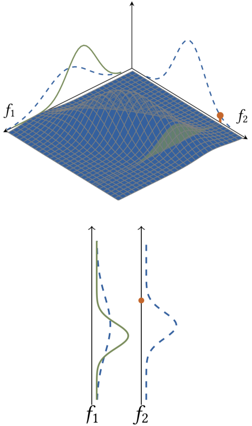


$$f_1|f_2 \sim \mathcal{N}\left(\mu_1 + \frac{\sigma_{12}}{\sigma_2^2}(f_2 - \mu_2), \sigma_1^2 - \frac{\sigma_{12}\sigma_{21}}{\sigma_2^2}\right)$$

A binary input

Another way to illustrate this is to plot only the marginal distributions





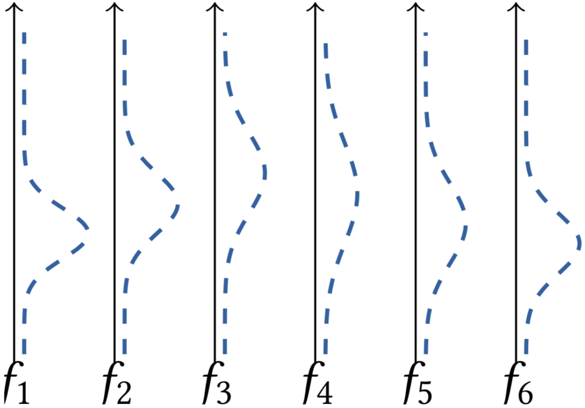
$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix}\right)$$

$$f_1 | f_2 \sim \mathcal{N}\left(\mu_1 + \frac{\sigma_{12}}{\sigma_2^2}(f_2 - \mu_2), \sigma_1^2 - \frac{\sigma_{12}\sigma_{21}}{\sigma_2^2}\right)$$

A discrete input

Now, if $x \in \{1, 2, 3, 4, 5, 6\}$, we can do the same thing. With $\boldsymbol{f} = \begin{bmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \end{bmatrix}^T$ we assume

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$



A discrete input

If we block $\boldsymbol{f}$ in two parts $\boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_a \\ \boldsymbol{f}_b \end{bmatrix}$, we can write

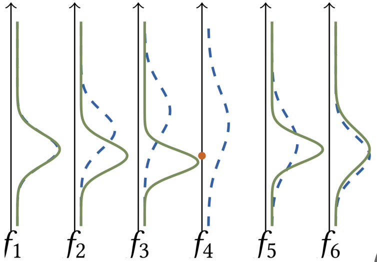
$$\begin{bmatrix} \boldsymbol{f}_a \\ \boldsymbol{f}_b \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix} \right)$$

and get (Thm 2 - *lecture on normal distribution*)

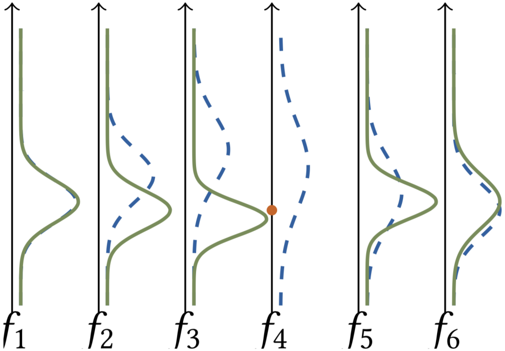
$$\boldsymbol{f}_a | \boldsymbol{f}_b \sim \mathcal{N}\left(\boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\boldsymbol{f}_b - \boldsymbol{\mu}_b), \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba} \right)$$

if we observe $\boldsymbol{f}_b$, we get an updated prediction for $\boldsymbol{f}_a$ as $p(\boldsymbol{f}_a | \boldsymbol{f}_b)$

for example, let $\boldsymbol{f}_b = f_4 \Rightarrow$



A discrete input



Can we generalize this idea to continuous inputs?

That is, $x \in \mathbb{R}$ instead of $x \in \{1, 2, 3, 4, 5, 6\}$?
The response is.... YES $\Rightarrow$ **Gaussian Process !**

A discrete input

For the case of a finite set of input values $x \in \{1, 2, \ldots, n\}$ we can use the multivariate Gaussian as a model for $f(x)$.

$$\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_n^2 \end{bmatrix} \right)$$

For simplicity we use a prior with zero mean $\boldsymbol{\mu} = \mathbf{0}$ - still useful in practice.

How do we generalize this to conitnuous inputs?

The Gaussian process

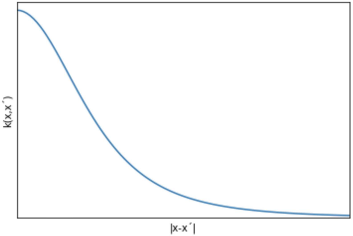
We have to introduce a **covariance function** $\kappa(x, x')$ such that

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \kappa(x_1, x_1) & \kappa(x_1, x_2) & \cdots & \kappa(x_1, x_n) \\ \kappa(x_2, x_1) & \kappa(x_2, x_2) & \cdots & \kappa(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(x_n, x_1) & \kappa(x_n, x_2) & \cdots & \kappa(x_n, x_n) \end{bmatrix} \right)$$

for any choice of $\{x_1, x_2, \ldots, x_n\}$

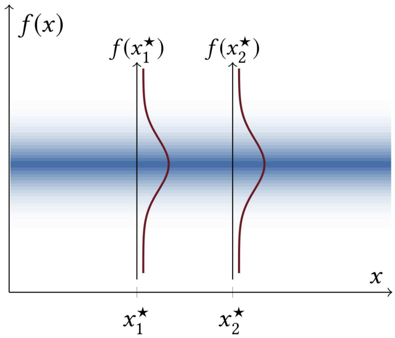
One choice of $\kappa(x, x')$, out of many, is

$$\kappa(x, x') = \left(1 + \frac{|x - x'|^2}{2\alpha l}\right)^{-\alpha}$$

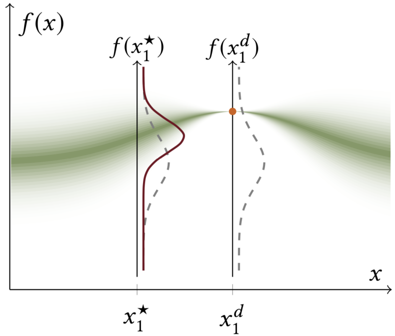


Given a $\kappa(x, x')$ everything follows as before

The Gaussian process



The distribution for $f(x^*)$ without any observations



The distribution for $f(x^*)$ conditional on an observation at x_1^d

The Gaussian process: definition

Definition: Gaussian Process

Let $\mathcal{X} \subset \mathbb{R}^d$ be some bounded domain of a d-dimensional real valued vector space. Denote by $f(\boldsymbol{x}) : \mathcal{X} \mapsto \mathbb{R}$ a stochastic process parametrized by $\boldsymbol{x} \in \mathcal{X}$. Then, the random function $f(\boldsymbol{x})$ is a Gaussian process if all its finite dimensional distributions are Gaussian, i.e. where for any $m \in \mathbb{N}$, the random vector $[f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_m)]$ is normally distributed.

The Gaussian process: the "core" equations

With $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$, $f(\boldsymbol{x}) = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{bmatrix}$, $K(x^*, x^*) = \kappa(x^*, x^*)$,

$$K(\boldsymbol{x}, x^*) = \begin{bmatrix} \kappa(x_1, x^*) \\ \kappa(x_2, x^*) \\ \vdots \\ \kappa(x_N, x^*) \end{bmatrix} = K(x^*, \boldsymbol{x})^T,\ K(\boldsymbol{x}, \boldsymbol{x}) = \begin{bmatrix} \kappa(x_1, x_1) & \cdots & \kappa(x_1, x_N) \\ \vdots & \vdots & \\ \kappa(x_N, x_1) & \cdots & \kappa(x_N, x_N) \end{bmatrix}$$

we have

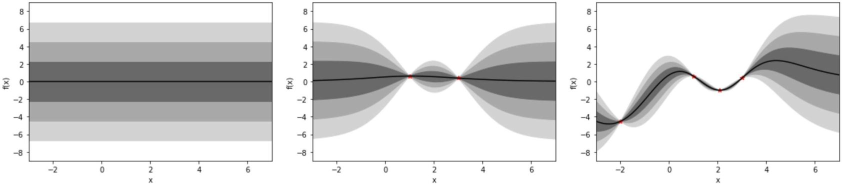
$$\begin{bmatrix} f(\boldsymbol{x}) \\ f(x^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{x}) & K(x^*, \boldsymbol{x}) \\ K(\boldsymbol{x}, x^*) & K(x^*, x^*) \end{bmatrix} \right)$$

and thus most importantly the distribution of the function at the prediction value (Thm 2, *lecture on normal distribution*)

$$f(x^*)|f(\boldsymbol{x}) \sim \mathcal{N}\left(K(x^*, \boldsymbol{x})K(\boldsymbol{x}, \boldsymbol{x})^{-1} f(\boldsymbol{x}), K(x^*, x^*) - K(x^*, \boldsymbol{x})K(\boldsymbol{x}, \boldsymbol{x})^{-1} K(\boldsymbol{x}, x^*)\right)$$

The Gaussian process as a regression model

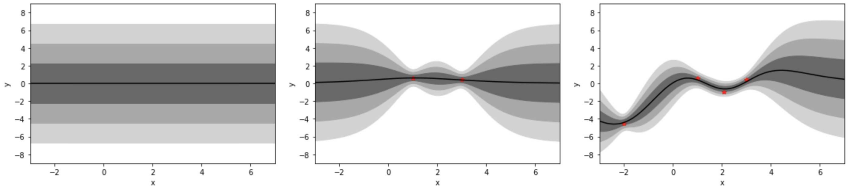
$$f(x^*)|f(\boldsymbol{x}) \sim \mathcal{N}\left(K(x^*,\boldsymbol{x})K(\boldsymbol{x},\boldsymbol{x})^{-1}f(\boldsymbol{x}), K(x^*,x^*) - K(x^*,\boldsymbol{x})K(\boldsymbol{x},\boldsymbol{x})^{-1}K(\boldsymbol{x},x^*)\right)$$



The Gaussian process as a regression model

But what if we don't observe $f(x)$ exactly, but observe $y = f(x) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$?

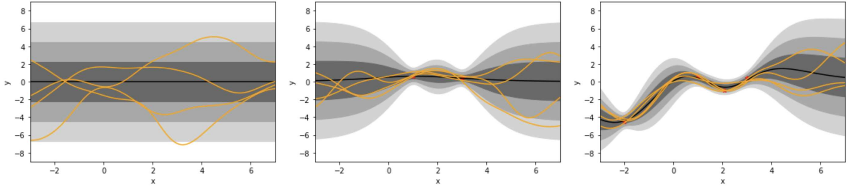
$$f(x^*)|\boldsymbol{y} \sim \mathcal{N}(K(x^*, \boldsymbol{x})(K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_\varepsilon^2 \boldsymbol{I})^{-1} f(\boldsymbol{x}),$$
$$K(x^*, x^*) - K(x^*, \boldsymbol{x})(K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_\varepsilon^2 \boldsymbol{I})^{-1} K(\boldsymbol{x}, x^*))$$



Samples from the Gaussian process

we can also draw samples from $p(f(x^*)|\boldsymbol{y})$ But what if we don't observe $f(x)$ exactly, but observe $y = f(x) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$?

$$f(x^*)|\boldsymbol{y} \sim \mathcal{N}(K(x^*, \boldsymbol{x})(K(\boldsymbol{x}, \boldsymbol{x})+\sigma_\varepsilon^2 \boldsymbol{I})^{-1}\boldsymbol{y},$$
$$K(x^*, x^*) - K(x^*, \boldsymbol{x})(K(\boldsymbol{x}, \boldsymbol{x})+\sigma_\varepsilon^2 \boldsymbol{I})^{-1}K(\boldsymbol{x}, x^*))$$



Here, x^* is a vector on a very fine grid on $[-3; 7]$, so the samples look continuous! The Gaussian process defines a **distribution over functions**

Outline

Bayesian linear regression

Lets assume that we want to model our N data points $\mathcal{D} = \left\{ \boldsymbol{x}^{(i)}, y^{(i)} \right\}_{i=1}^{N}$ as, for $j = 1, \dots, N$:

$$y^{(j)} = \sum_{i=1}^{k} \beta_i \boldsymbol{x}_i^{(j)} + \varepsilon^{(j)} \quad \text{with} \quad \varepsilon^{(j)} \sim \mathcal{N}(\mathbf{0}, \sigma_\varepsilon^2)$$

In a matrix form, the complete set of observations can be written as follows:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where $\mathbf{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}$, $\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_{k-1}^{(1)} \\ \vdots & & \ddots & & \vdots \\ 1 & x_1^{(N)} & x_2^{(N)} & \cdots & x_{k-1}^{(N)} \end{bmatrix}$

Bayesian linear regression

- We have $\boldsymbol{y} = \boldsymbol{X}\beta + \boldsymbol{\varepsilon}$ with $\varepsilon^{(j)} \sim \mathcal{N}(\boldsymbol{0}, \sigma_\varepsilon^2 \boldsymbol{I})$ *(σ^2 is assumed to be known)*

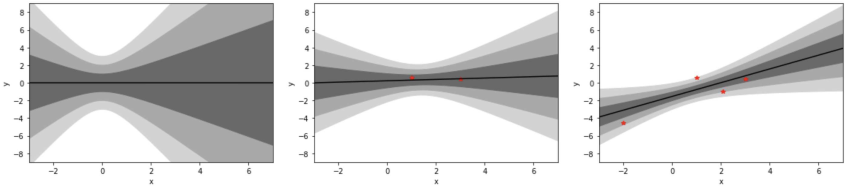
  $\Rightarrow$ $p(\boldsymbol{y}|\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{X}\boldsymbol{\beta}, \sigma_\varepsilon^2 \boldsymbol{I})$
- $p(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta}; \boldsymbol{0}, \boldsymbol{I})$

  $\Rightarrow$ $p(\boldsymbol{\beta}|\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{\beta}; \left(\boldsymbol{I} + \frac{1}{\sigma_\varepsilon^2}\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\left(\frac{1}{\sigma_\varepsilon^2}\boldsymbol{X}^T\boldsymbol{y}\right), \left(\boldsymbol{I} + \frac{1}{\sigma_\varepsilon^2}\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\right)$

  $\Rightarrow$ $p(f(\boldsymbol{x}^*)|\boldsymbol{y}) =$
  $\mathcal{N}\left(f(\boldsymbol{x}^*); \boldsymbol{x}^{*T}\left(\boldsymbol{I} + \frac{1}{\sigma_\varepsilon^2}\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\left(\frac{1}{\sigma_\varepsilon^2}\boldsymbol{X}^T\boldsymbol{y}\right), \boldsymbol{x}^{*T}\left(\boldsymbol{I} + \frac{1}{\sigma_\varepsilon^2}\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{x}^*\right)$

  $\boldsymbol{x}^*$ is an arbitrary *test input*.

Bayesian linear regression



$$p(f(\boldsymbol{x}^*)|\boldsymbol{y}) =$$

$$\mathcal{N}\left(f(\boldsymbol{x}^*); \underbrace{\boldsymbol{x}^{*T}\left(\boldsymbol{I} + \frac{1}{\sigma_\varepsilon^2}\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\left(\frac{1}{\sigma_\varepsilon^2}\boldsymbol{X}^T\boldsymbol{y}\right)}_{\text{Predictive Posterior mean; black line}}, \underbrace{\boldsymbol{x}^{*T}\left(\boldsymbol{I} + \frac{1}{\sigma_\varepsilon^2}\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{x}^*}_{\text{Predictive Posterior variance; gray areas}}\right)$$

Red dots: observed data $\mathcal{D} = \left\{\boldsymbol{x}^{(i)}, y^{(i)}\right\}_{i=1}^{N}$

Bayesian linear regression

And now if we use non-linear transformations of input using basis functions, such as $\boldsymbol{b}(x) = \begin{bmatrix}1 & x & x^2\end{bmatrix}^T$

$$p(f(x^*)|\boldsymbol{y}) =$$
$$\mathcal{N}\left(f(x^*); \boldsymbol{b}(x^*)^T \left(\boldsymbol{I} + \frac{1}{\sigma_\varepsilon^2}\boldsymbol{B}^T\boldsymbol{B}\right)^{-1} \left(\frac{1}{\sigma_\varepsilon^2}\boldsymbol{B}^T\boldsymbol{y}\right), \boldsymbol{b}(x^*)^T \left(\boldsymbol{I} + \frac{1}{\sigma_\varepsilon^2}\boldsymbol{B}^T\boldsymbol{B}\right)^{-1} \boldsymbol{b}(x^*)\right)$$

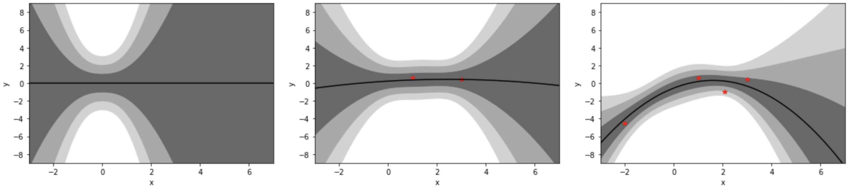
with

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}(x^{(1)})^T \\ \boldsymbol{b}(x^{(2)})^T \\ \vdots \\ \boldsymbol{b}(x^{(N)})^T \end{bmatrix}$$

Bayesian linear regression

With

$$\boldsymbol{b}(x) = \begin{bmatrix} 1 & x & x^2 \end{bmatrix}^T$$



Bayesian linear regression

$$p(f(x^*)|\boldsymbol{y}) =$$

$$\mathcal{N}\left(f(x^*); \boldsymbol{b}(x^*)^T \left(\sigma_\varepsilon^2 \boldsymbol{I} + \boldsymbol{B}^T \boldsymbol{B}\right)^{-1} \left(\boldsymbol{B}^T \boldsymbol{y}\right), \boldsymbol{b}(x^*)^T \left(\boldsymbol{I} + \frac{1}{\sigma_\varepsilon^2} \boldsymbol{B}^T \boldsymbol{B}\right)^{-1} \boldsymbol{b}(x^*)\right)$$

For any matrix $\boldsymbol{A}$, $(\boldsymbol{I} + \boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{I} + \boldsymbol{A}\boldsymbol{A}^T)^{-1}$. Hence,

$$\boldsymbol{b}(x^*)^T \left(\sigma_\varepsilon^2 \boldsymbol{I} + \boldsymbol{B}^T \boldsymbol{B}\right)^{-1} \boldsymbol{B}^T \boldsymbol{y} = \boldsymbol{b}(x^*)^T \boldsymbol{B}^T \left(\sigma_\varepsilon^2 \boldsymbol{I} + \boldsymbol{B}\boldsymbol{B}^T\right)^{-1} \boldsymbol{y}$$

The matrix inversion lemma $(\boldsymbol{I} + \boldsymbol{U}\boldsymbol{V})^{-1} = \boldsymbol{I} - \boldsymbol{U}(\boldsymbol{I} + \boldsymbol{V}\boldsymbol{U})^{-1}\boldsymbol{V}$ gives

$$\boldsymbol{b}(x^*)^T \left(\boldsymbol{I} + \frac{1}{\sigma_\varepsilon^2} \boldsymbol{B}^T \boldsymbol{B}\right)^{-1} \boldsymbol{b}(x^*) =$$

$$\boldsymbol{b}(x^*)^T \boldsymbol{b}(x^*) - \boldsymbol{b}(x^*)^T \boldsymbol{B}^T \left(\sigma_\varepsilon^2 \boldsymbol{I} + \boldsymbol{B}^T \boldsymbol{B}\right)^{-1} \boldsymbol{B}\boldsymbol{b}(x^*)$$

Kernels

Let $\kappa(x, x') = \boldsymbol{b}(x)^T \boldsymbol{b}(x')$, we refer to $\kappa(\cdot, \cdot)$ as a **kernel**.
$K(x^*, x^*) = \kappa(x^*, x^*) = \boldsymbol{b}(x^*)^T \boldsymbol{b}(x^*)$

$$K(\boldsymbol{x}, x^*) = \begin{bmatrix} \kappa(x^{(1)}, x^*) \\ \vdots \\ \kappa(x^{(N)}, x^*) \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}(x^{(1)})^T \boldsymbol{b}(x^*) \\ \vdots \\ \boldsymbol{b}(x^{(N)})^T \boldsymbol{b}(x^*) \end{bmatrix} = \boldsymbol{B}\boldsymbol{b}(x^*) = K(x^*, \boldsymbol{x})$$

$$K(\boldsymbol{x}, \boldsymbol{x}) = \begin{bmatrix} \kappa(x^{(1)}, x^{(1)}) & \cdots & \kappa(x^{(1)}, x^{(N)}) \\ \vdots & & \vdots \\ \kappa(x^{(N)}, x^{(1)}) & \cdots & \kappa(x^{(N)}, x^{(N)}) \end{bmatrix} =$$

$$\begin{bmatrix} \boldsymbol{b}(x^{(1)})^T \boldsymbol{b}(x^{(1)}) & \cdots & \boldsymbol{b}(x^{(1)})^T \boldsymbol{b}(x^{(N)}) \\ \vdots & & \vdots \\ \boldsymbol{b}(x^{(N)})^T \boldsymbol{b}(x^{(1)}) & \cdots & \boldsymbol{b}(x^{(N)})^T \boldsymbol{b}(x^{(N)}) \end{bmatrix} = \boldsymbol{B}\boldsymbol{B}^T$$

The kernel trick

$$p(f(x^*)|\boldsymbol{y}) = \mathcal{N}\left(f(x^*); K(x^*, \boldsymbol{x})\left(\sigma_\varepsilon^2 \boldsymbol{I} + K(\boldsymbol{x}, \boldsymbol{x})\right)^{-1} \boldsymbol{y},\right.$$
$$\left. K(x^*, x^*) - K(x^*, \boldsymbol{x})\left(\sigma_\varepsilon^2 \boldsymbol{I} + K(\boldsymbol{x}, \boldsymbol{x})\right)^{-1} K(\boldsymbol{x}, x^*)\right)$$

The input x only appears in $p(f(x^*)|\boldsymbol{y})$ via the kernel $\kappa(x, x') = \boldsymbol{b}(x)^T \boldsymbol{b}(x')$

The kernel trick:

Do not compute (or even choose!) the nonlinear transformation $\boldsymbol{b}(x)$, work directly with $\kappa(x, x')$ instead

The kernel trick

The kernel trick:

Do not compute (or even choose!) the nonlinear transformation $\boldsymbol{b}(x)$, work directly with $\kappa(x, x')$ instead

Only requirement on $\kappa(x, x')$: $K(\boldsymbol{x}, \boldsymbol{x})$ has to be positive semidefinite for all possible values on $\boldsymbol{x}$

One possible choice of $\kappa(x, x')$, out of many, is

$$\kappa(x, x') = \left(1 + \frac{|x - x'|^2}{2\alpha l}\right)^{-\alpha},$$

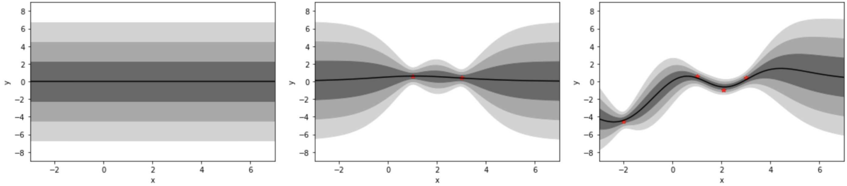
the rational quadratic kernel

For any kernel $\kappa(\cdot, \cdot)$, a corresponding (possibily infinite) nonlinear feature transformation $\boldsymbol{b}(\cdot)$ can be constructed, which lives in a *reproducing kernel Hilbert space.*

Gaussian process regression

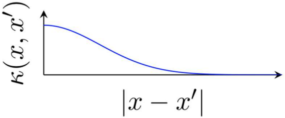
$$p(f(x^*)|\boldsymbol{y}) = \mathcal{N}(f(x^*), K(x^*,\boldsymbol{x})(K(\boldsymbol{x},\boldsymbol{x})+\sigma_\varepsilon^2\boldsymbol{I})^{-1}\boldsymbol{y},$$
$$K(x^*,x^*) - K(x^*,\boldsymbol{x})(K(\boldsymbol{x},\boldsymbol{x})+\sigma_\varepsilon^2\boldsymbol{I})^{-1}K(\boldsymbol{x},x^*))$$

$$\kappa(x,x') = \left(1 + \frac{|x-x'|^2}{2\alpha l}\right)^{-\alpha},$$

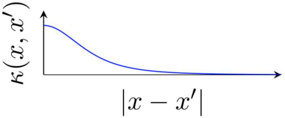


Gaussian process regression - Kernel = covraiance function

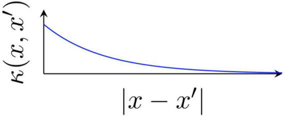
Squared exponential/RBF
$\kappa(x,x') = \sigma^2 \exp(-\frac{1}{2l^2}(x-x')^2)$



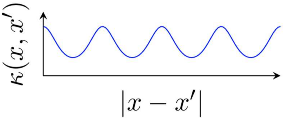
Rational quadratic
$\kappa(x,x') = \left(1 + \frac{|x-x'|^2}{2\alpha l}\right)^{-\alpha}$



Matérn 1
$\kappa(x,x') = \sigma^2 \exp(-\frac{1}{l^2}|x-x'|)$



Periodic kernel
$\kappa(x,x') = \sigma^2 \exp(-\frac{2}{l^2}\sin^2(\pi\frac{|x-x'|}{p}))$

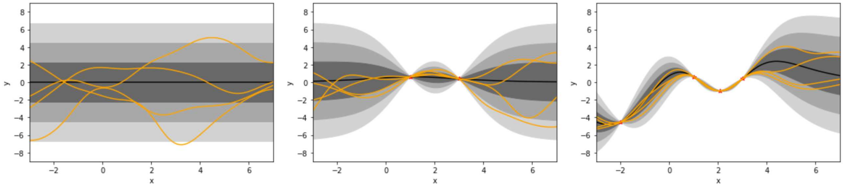


Outline

The Gaussian Process - Kernel choice

$$f(x^*)|\boldsymbol{y} \sim \mathcal{N}(\underbrace{K(x^*,\boldsymbol{x})(K(\boldsymbol{x},\boldsymbol{x})+\sigma_\varepsilon^2\boldsymbol{I})^{-1}\boldsymbol{y}}_{\text{Predictive posterior mean}}, \underbrace{K(x^*,x^*) - K(x^*,\boldsymbol{x})(K(\boldsymbol{x},\boldsymbol{x})+\sigma_\varepsilon^2\boldsymbol{I})^{-1}K(\boldsymbol{x},x^*)}_{\text{Predictive posterior covariance}})$$

$$\kappa(x,x') = \left(1 + \frac{|x-x'|^2}{2\alpha l}\right)^{-\alpha}, \quad \sigma^2 = 5, \alpha = 2, l = 3$$

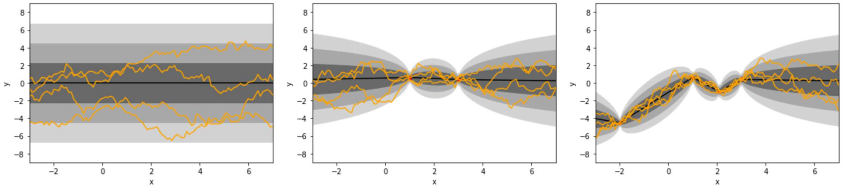


The choice of kernel and hyperparameter is crucial!

The Gaussian Process - Kernel choice

$$f(x^*)|\boldsymbol{y} \sim \mathcal{N}(\underbrace{K(x^*,\boldsymbol{x})(K(\boldsymbol{x},\boldsymbol{x})+\sigma_\varepsilon^2\boldsymbol{I})^{-1}\boldsymbol{y}}_{\text{Predictive posterior mean}}, \underbrace{K(x^*,x^*) - K(x^*,\boldsymbol{x})(K(\boldsymbol{x},\boldsymbol{x})+\sigma_\varepsilon^2\boldsymbol{I})^{-1}K(\boldsymbol{x},x^*)}_{\text{Predictive posterior covariance}})$$

$$\kappa(x,x') = \sigma^2 \exp(-\frac{|x-x'|}{l^2}), \quad \sigma^2 = 5, l = 3$$



The choice of kernel and hyperparameter is crucial!

Importance of kernel choice

- The kernel $\kappa(x, x')$ encodes assumptions on how much correlation there is between $f(x)$ and $f(x')$
- The kernel tells how the model should generalize the traning data

Even with prior mean 0, the predictive posterior does not have mean 0 thanks to the kernel

Constructing new kernels

For a kernel to be valid for Gaussian processes, the matrix

$$K(\boldsymbol{x}, \boldsymbol{x}) = \begin{bmatrix} \kappa(x^{(1)}, x^{(1)}) & \cdots & \kappa(x^{(1)}, x^{(N)}) \\ \vdots & & \vdots \\ \kappa(x^{(N)}, x^{(1)}) & \cdots & \kappa(x^{(N)}, x^{(N)}) \end{bmatrix}$$

must be positive semidefinite for all possible $\boldsymbol{x}$

- you can invent completely new kernels, as long as they fulfill this criterion
- you can create composite kernels by multiplying or adding existing ones

$$\kappa_\times(x, x') = \kappa_1(x, x')\kappa_2(x, x')$$
$$\kappa_+(x, x') = \kappa_1(x, x') + \kappa_2(x, x')$$

In the end, the choice of kernel is a design choice left to the machine learning engineer.

Choosing hyperparameters

How to choose the hyperparameters $\xi = \{\sigma_\varepsilon^2, l, \alpha, \ldots\}$

- The go-to solution for machine learning: **(k-fold) cross validation**
- A more probabilistic alternative: **maximizing the marginal likelihood**

Both approaches can be used in practice !

Gaussian processes in machine learning

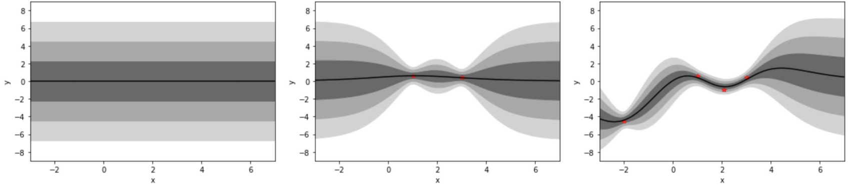
- The idea dates back to the 60's ("kriging"); model the presence of gold in South Africa based on information from boreholes (a regression problem!)
- Big interest within the machine learning research, because of its Bayesian and non-parametric nature
- Has not (yet?) become as popular among practitioners as, e.g., random forests and neural networks
- Many interesting research directions!

Gaussian process Summary

A Bayesian/probabilistic nonparametric model for regression

- Bayesian/probabilistic: the predictions $f(x^*)|\boldsymbol{y}$ are not only points, but distributions.
- Nonparametric: the predictions $f(x^*)|\boldsymbol{y}$ depends on all observed data, and not just a fixed set of parameters

$$f(x^*)|\boldsymbol{y} \sim \mathcal{N}(\underbrace{K(x^*,\boldsymbol{x})(K(\boldsymbol{x},\boldsymbol{x})+\sigma_\varepsilon^2\boldsymbol{I})^{-1}\boldsymbol{y}}_{\text{Predictive posterior mean}}, \underbrace{K(x^*,x^*) - K(x^*,\boldsymbol{x})(K(\boldsymbol{x},\boldsymbol{x})+\sigma_\varepsilon^2\boldsymbol{I})^{-1}K(\boldsymbol{x},x^*)}_{\text{Predictive posterior covariance}})$$



More details in Rasmussen and Williams (2006)

References I

Rasmussen, C. and Williams, C. (2006). *Gaussian processes for machine learning*. The MIT Press.