Secondmind

GPs and Kernel Design

GPSS 2021 - Nicolas Durrande

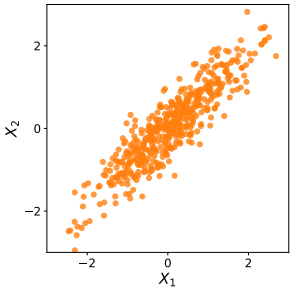
Outline

Talk is organised in 5 sections

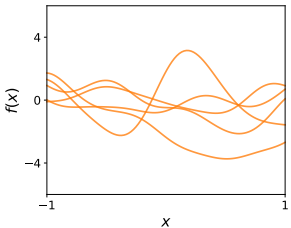
1. From random variables to Gaussian Processes
2. Gaussian process Regression
3. Parameter estimation and model validation
4. Choosing the kernel
5. Making new from old

From random variables to Gaussian processes

VS



VS



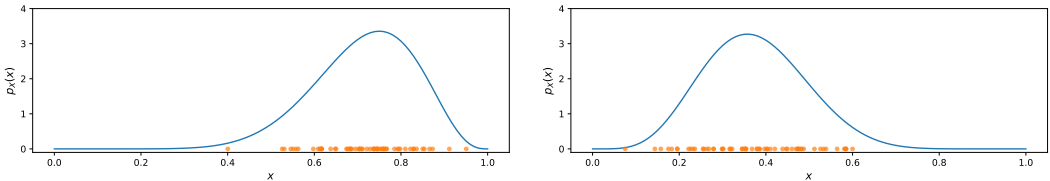
random variable

random vector

random function

Random variables

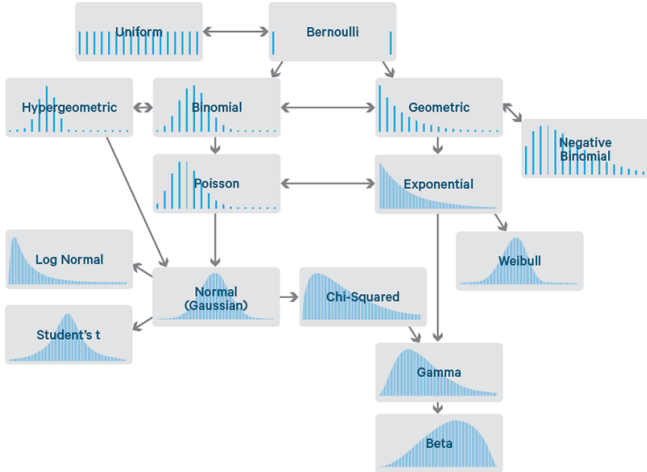
One way to describe the distribution of a random variable X is through its probability density function p_X:



Probability density functions give the "probability mass" for each interval:

$$P(X \in [a, b]) = \int_a^b p_X(x)\, dx.$$

There are many of commonly used families of distributions for random variables:



Unsurprisingly, the Gaussian (or Normal) distribution will be particularly relevant to us!

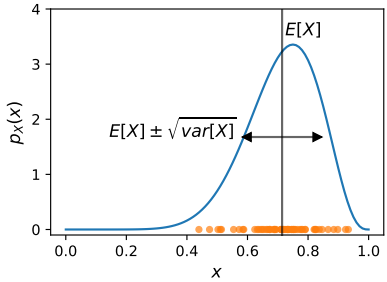
The expectation captures the mean value of a random variable:

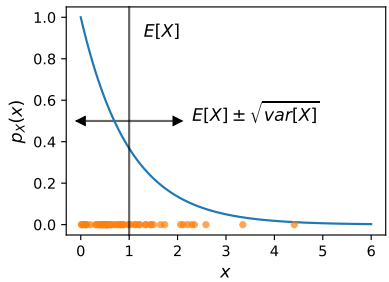
$$\mathsf{E}[X] = \int_{\mathbb{R}} x\, p_X(x)\, dx$$

The variance quantifies the dispersion around the mean value:

$$\mathsf{var}[X] = \mathsf{E}[(X - \mathsf{E}[X])^2] = \mathsf{E}[X^2] - \mathsf{E}[X]^2$$

Examples

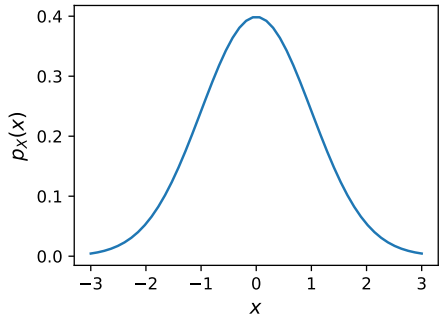




1D normal distribution

We say that X is normally distributed if its pdf writes:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



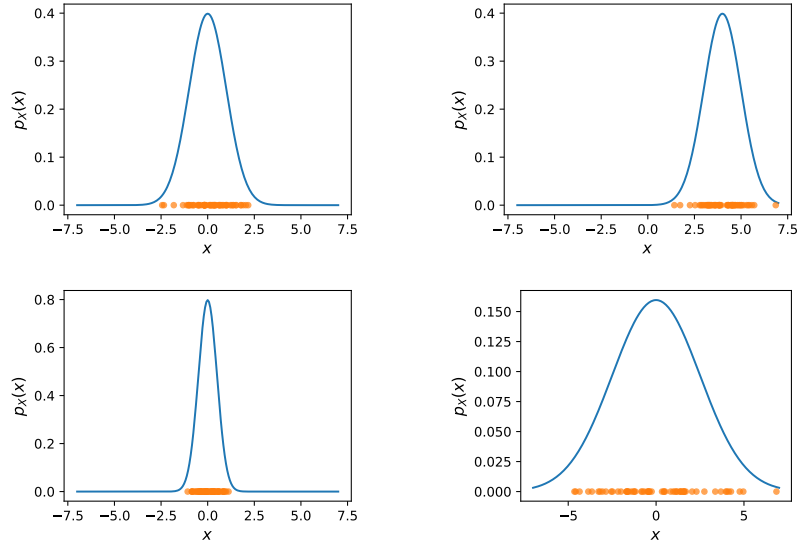
Computing the expectation and variance of X yields:

$$\mathsf{E}[X] = \mu \quad \text{and} \quad \mathsf{var}[X] = \sigma^2.$$

These two quantities are thus enough to characterise the distribution of X.

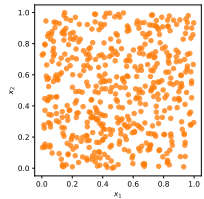
"X is normally distributed with mean μ and variance σ^2" is written: $X \sim \mathcal{N}(\mu, \sigma^2)$.

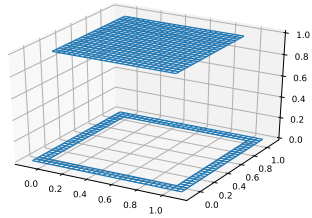
Examples of normally distributed random variables for various values of μ and σ: ($\mu \in \{0, 1\}$, and $\sigma \in \{0.5, 1, 2.5\}$)

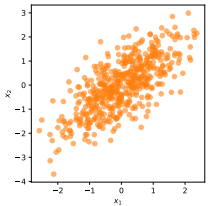


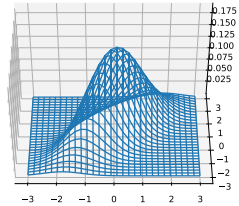
Random vectors

Probability density functions are also defined for random vectors:









Multivariate normal

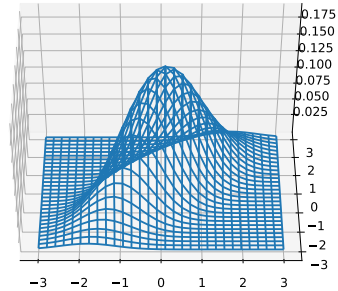
We say that $X = (X_1, ..., X_n)$ is multivariate normal if its pdf writes:

$$p_x(x) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

μ is the mean vector, and Σ is the covariance matrix:

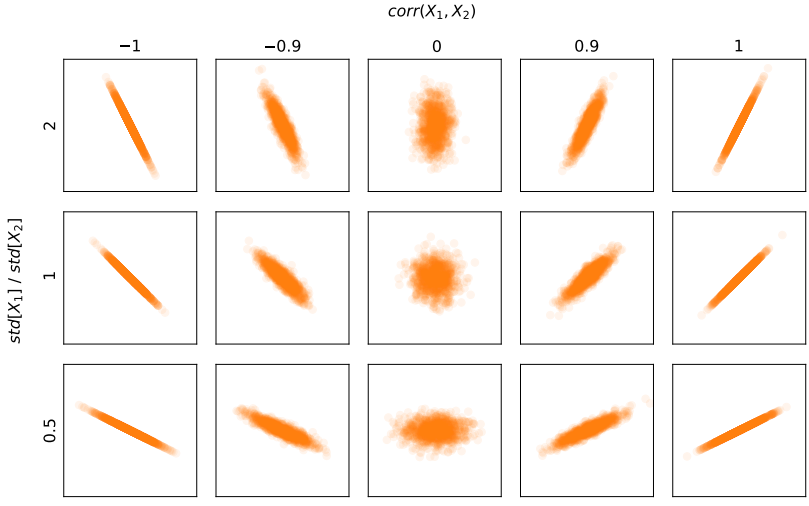
$$\mu = \mathsf{E}[X] = \begin{pmatrix} \mathsf{E}[X_1] \\ \\ \mathsf{E}[X_2] \end{pmatrix}$$

$$\Sigma = \mathsf{cov}(X, X) = \begin{pmatrix} \mathsf{var}[X_1] & \mathsf{cov}(X_1, X_2) \\ \\ \mathsf{cov}(X_2, X_1) & \mathsf{var}[X_2] \end{pmatrix}$$



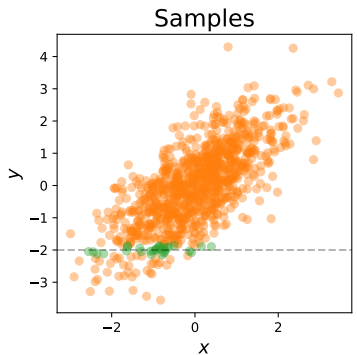
As previously we make use of the notation $X \sim \mathcal{N}(\mu, \Sigma)$.

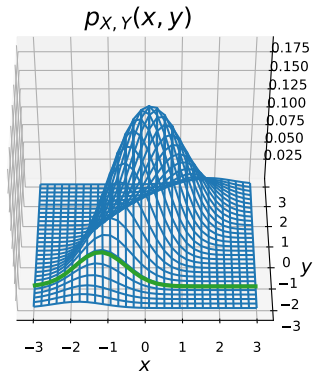
The covariance matrix Σ captures both the dispersion of each component, and their correlation:



Conditional distribution

Knowing the value of one component of a random vector $X = (X_1, X_2)$ can give some information on the other:





The conditional distribution of X_1 given $X_2 = a$ is proportional to $p_{X_1,Y_2}(x, a)$.

Conditional distribution

For a multivariate normal distribution $X = (X_1, X_2)$, the conditional distribution of $X_1|X_2 = a$ has two great properties

+ It is still normally distributed
+ It's mean and variance are known analytically

More precisely, let

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \right)$$

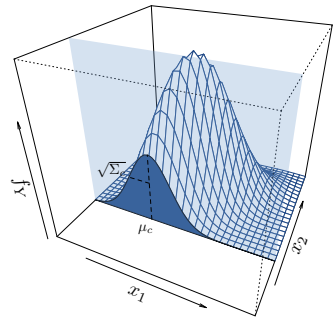
Then the conditional distribution of X_1 given $X_2 = a$ is:

$$X_1|\{X_2 = a\} \sim \mathcal{N}(\mu_{\text{cond}}, \Sigma_{\text{cond}})$$

with

$$\mu_{\text{cond}} = \mathsf{E}[X_1|X_2 = a] = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(a - \mu_2)$$

$$\Sigma_{\text{cond}} = \mathsf{cov}(X_1, X_1|X_2) = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$



Conditional distribution

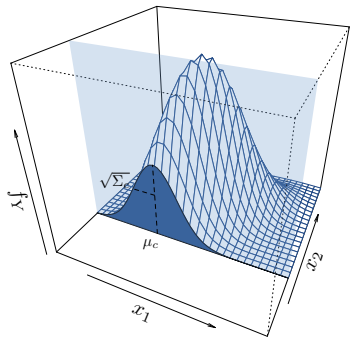
Intuition why...

$$p(y_1|y_2 = \alpha) = \frac{p(y_1, \alpha)}{p(\alpha)}$$

$$= \frac{\exp(\text{quadratic in } y_1 \text{ and } \alpha)}{\text{const}}$$

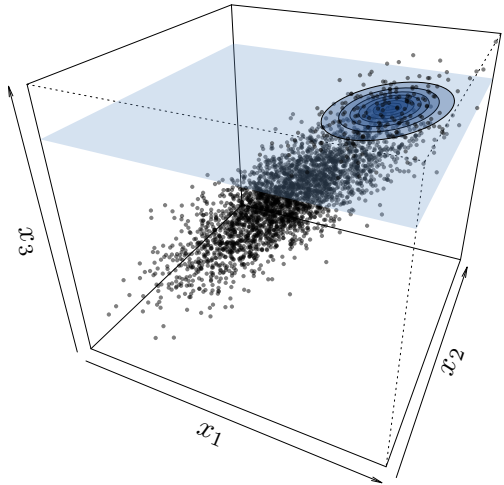
$$= \frac{\exp(\text{quadratic in } y_1)}{\text{const}}$$

$$= \text{normal distribution!}$$



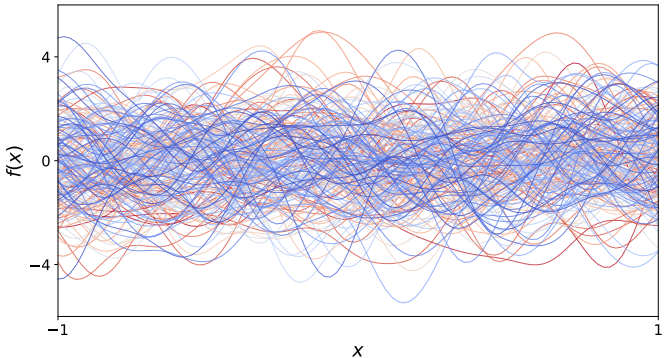
The conditional distribution is still Gaussian!

Illustration of the conditional distribution of a 3D multivariate normal vector



Random processes

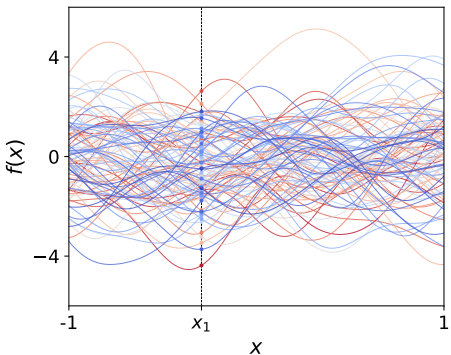
A random process is a generalisation of random variables/vectors, where each draw is a function

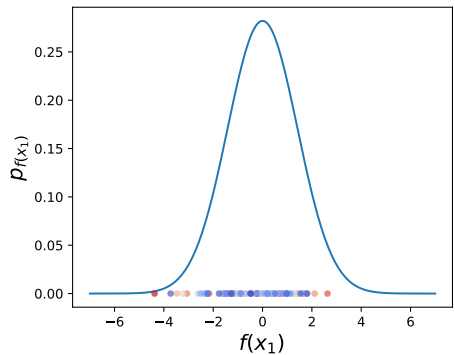


Gaussian Process

A Gaussian Process is a random process f where any finite set $\{f(x_1), f(x_2), ..., f(x_n)\}$ is multivariate normal:

Example For $n = 1$, we get

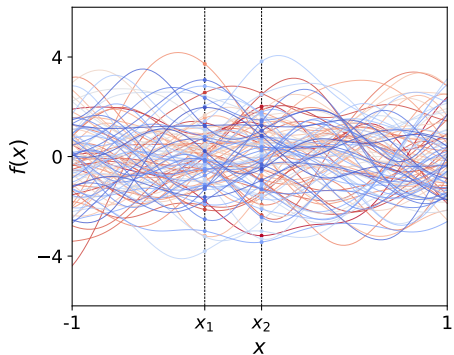


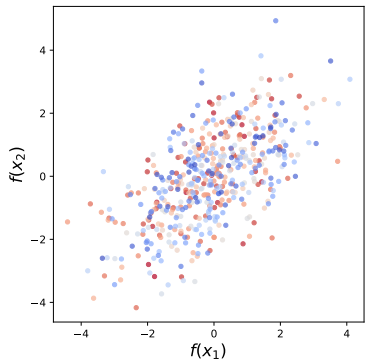


Gaussian Process

A Gaussian Process is a random process f where any finite set of the process value is multivariate normal:

Example For $n = 2$, we get





The distribution of a GP is fully characterised by its mean function and covariance function.

We write $f \sim \mathcal{N}(m(.), k(.,.))$:

$m : D \to \mathbb{R}$ is the mean function $m(x) = \mathsf{E}[f(x)]$

$k : D \times D \to \mathbb{R}$ is the covariance function (i.e. kernel):

$$k(x, y) = \mathsf{cov}(f(x), f(y))$$

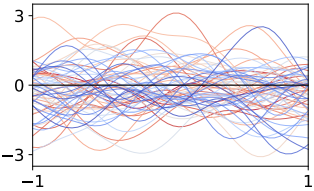
The mean m can be any function.

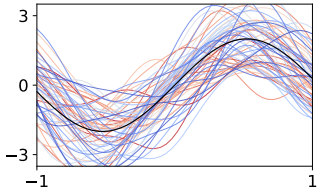
The kernel must satisfy:

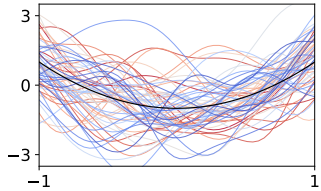
+ symmetric: $k(x, y) = k(y, x)$
+ positive semi-definite: for all $n \in \mathbb{N}$, for all $x_i \in D$, $\forall \alpha_i \in \mathbb{R}$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \geq 0$$

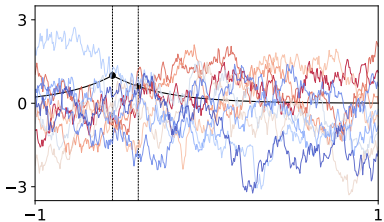
Influence of the mean function:

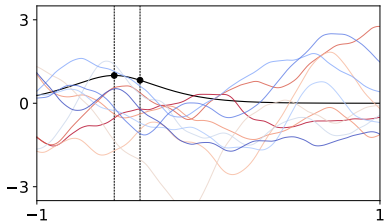






Influence of the kernel (i.e. covariance function)



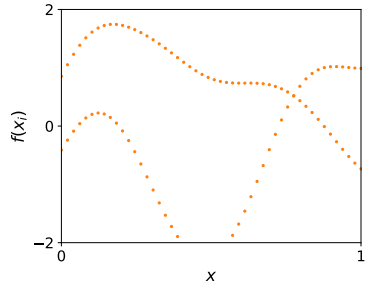


Sampling from a GP

Let $f \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$, and let $(x_1, \ldots, x_n)$ be a regular grid on $[0, 1]$. Then $(f(x_1), \ldots, f(x_n)) \sim \mathcal{N}(\mu, \Sigma)$ with:

$$\mu = \begin{pmatrix} m(x_1) \\ m(x_1) \\ \\ m(x_n) \end{pmatrix} \qquad \Sigma = \begin{pmatrix} k(x_1,x_1) & k(x_1,x_2) & & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & & k(x_2,x_n) \\ & & \vdots & \\ k(x_n,x_1) & k(x_n,x_2) & & k(x_n,x_n) \end{pmatrix}$$

One can use any random variable generator capable of sampling from a multivariate normal distribution to get a draw from $\mathcal{N}(\mu, \Sigma)$. The obtained vector can then be plotted against the indices x_i.



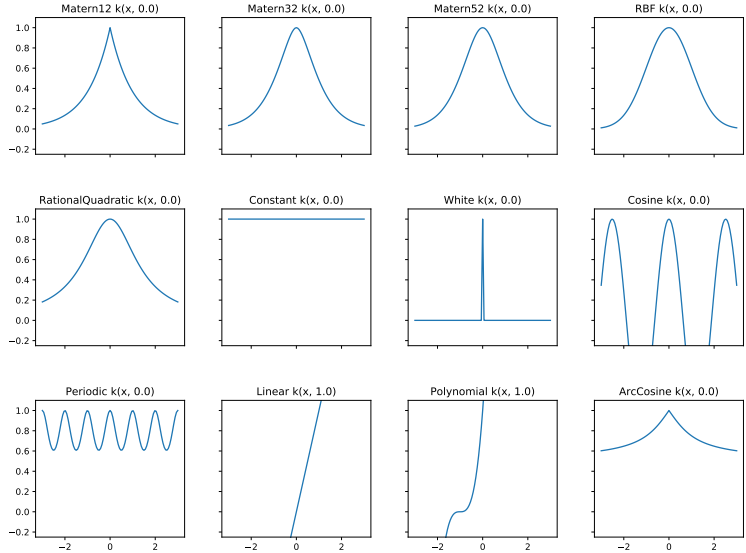
Kernels

There are lots of common kernels:

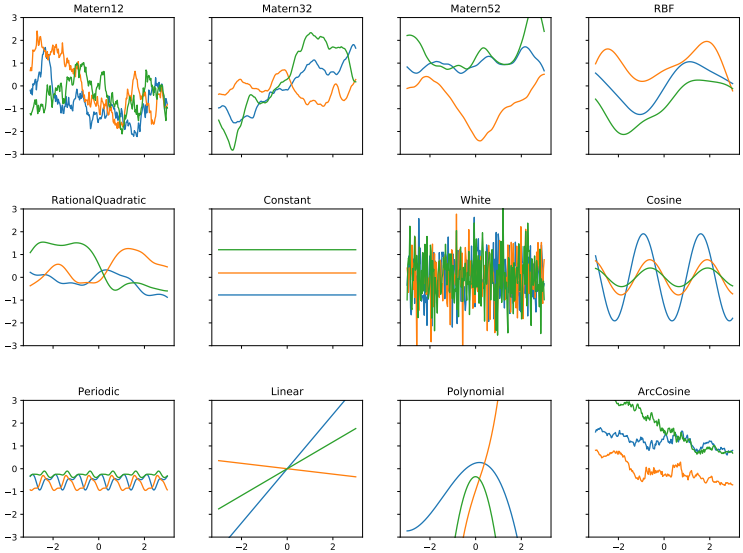
constant	$k(x, x') = \sigma^2$
white noise	$k(x, x') = \sigma^2 \delta_{x,x'}$
Brownian	$k(x, x') = \sigma^2 \min(x, x')$
exponential	$k(x, x') = \sigma^2 \exp\left(-\lvert x - x' \rvert / \theta\right)$
Matérn 3/2	$k(x, x') = \sigma^2 \left(1 + \lvert x - x' \rvert\right) \exp\left(-\lvert x - x' \rvert / \theta\right)$
Matérn 5/2	$k(x, x') = \sigma^2 \left(1 + \lvert x - x' \rvert / \theta + 1/3 \lvert x - x' \rvert^2 / \theta^2\right) \exp\left(-\lvert x - x' \rvert / \theta\right)$
squared exponential	$k(x, x') = \sigma^2 \exp\left(-(x - x')^2 / \theta^2\right)$
linear	$k(x, x') = \sigma^2 x\, x'$

The parameter σ^2 is called the variance and θ the lengthscale.

Examples of kernels in gpflow:

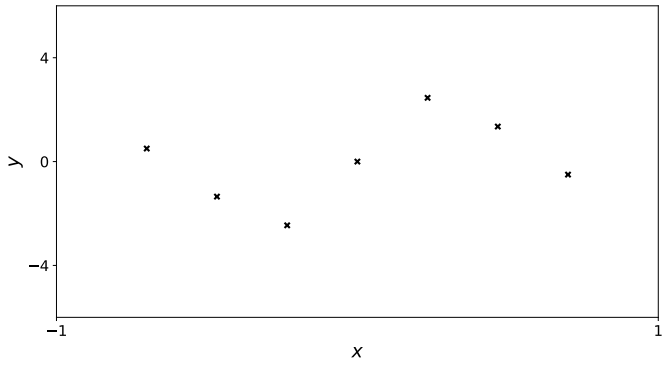


Associated samples



Gaussian Process Regression

We have some data corresponding to input/output tuples (x_i, y_i)



We want to predict the output value for any input point.

We cannot make predictions without making assumptions

In a GP regression model, we assume that the input x and the output y are related as follow:

$$y = f(x) + \varepsilon, \qquad \text{where} \qquad f \sim \mathcal{GP}(0, k(\cdot, \cdot)) \quad \text{and} \quad \varepsilon \sim \mathcal{N}(0, \tau^2)$$

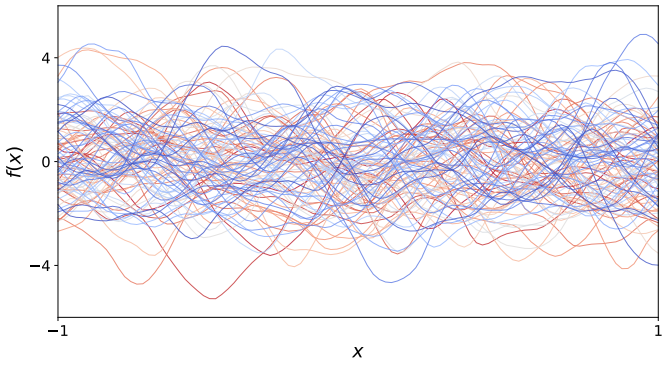
Inference consists in combining this model with the observed data in order to derive the posterior distribution

$$prior \ + \ data \ \rightarrow \ posterior$$

The rules of probability are here to help!

$$p(f|y) = \frac{p(y|f)p(f)}{p(y)}$$

We consider the model $y = f(x) + \varepsilon$ with $f \sim \mathcal{GP}(0, k)$ and $\varepsilon \sim \mathcal{N}(0, \tau^2)$:

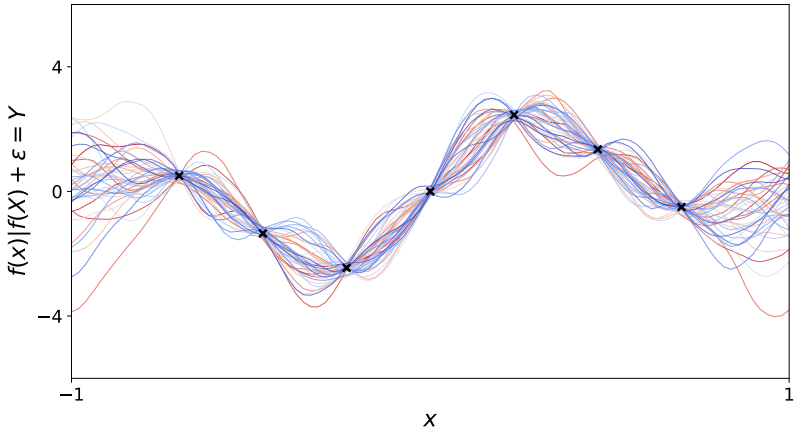


The posterior distribution $f(\cdot)|f(X) + \varepsilon = Y$ is still a Gaussian process.
The posterior mean and covariance can be computed analytically:

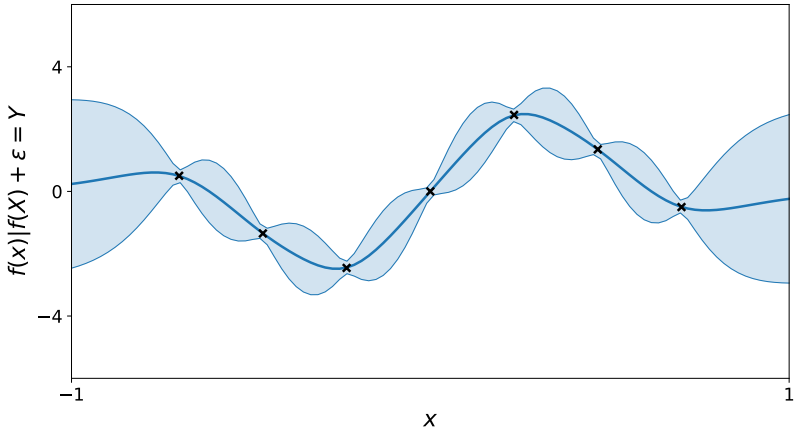
$$m(x) = k(x, X)(k(X, X) + \tau^2 I)^{-1} Y$$

$$c(x, x') = k(x, x') - k(x, X)(k(X, X) + \tau^2 I)^{-1} k(X, x')$$

Samples from the posterior distribution



It can be summarized by a mean function and 95% confidence intervals.



A few remarkable properties of GPR models

+ They (can) interpolate the data-points.
+ The prediction variance does not depend on the observations.
+ The mean predictor does not depend on the variance parameter.
+ The mean (usually) come back to zero when predicting far away from the observations.

Can we prove them?
Reminder:

$$m(x) = k(x, X)k(X, X)^{-1}F$$
$$c(x, y) = k(x, y) - k(x, X)k(X, X)^{-1}k(X, y)$$

We do we like GP so much?

They offer great features:

+ Quantification of uncertainty
  - The risk associated with decisions based on predictions can be controlled
  - Can be used to derive exploration/exploitation trade-off

    ⇒ Talk from Javier Gonzales on day 3

+ Versatile framework: non-conjugate likelihood, etc.

  ⇒ Talk from Ti John on day 3

+ Comfortable both with small and big data regimes
  - Extremely data efficient in low data regime
  - Can also cope with large datasets

    ⇒ Talk from Zhenwen Dai on day 3

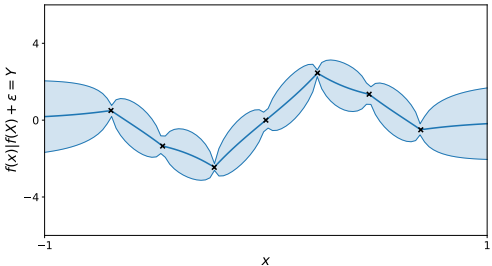
+ Principled approach: marginal likelihood as a training objective, etc.
+ (mathematically tractable!)

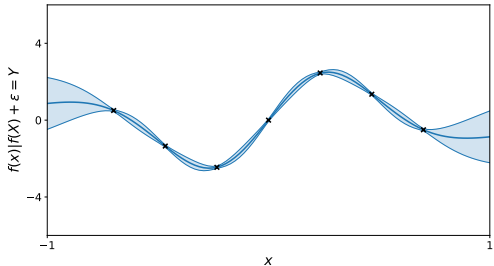
What are their limitations?

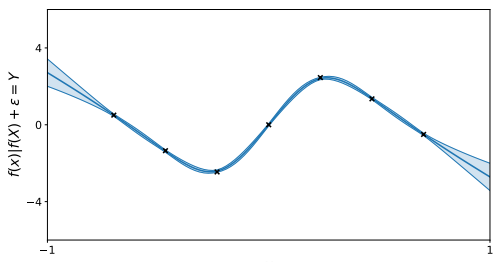
+ Complexity
  - Storage footprint is $\mathcal{O}(n^2)$: We have to store the covariance matrix which is $n \times n$.
  - Complexity is $\mathcal{O}(n^3)$: We have to invert the covariance matrix (or compute the Cholesky factor and apply triangular solves).

+ Numerical stability (you want double precision!)

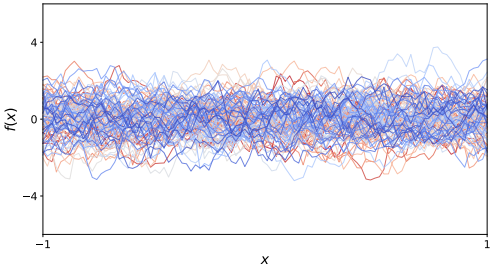
The kernel has a huge impact on the model: (Matérn 1/2, Matérn 5/2 and Arccosine kernels)

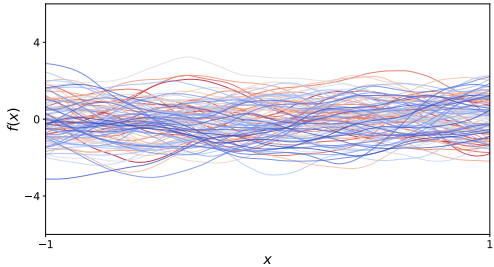


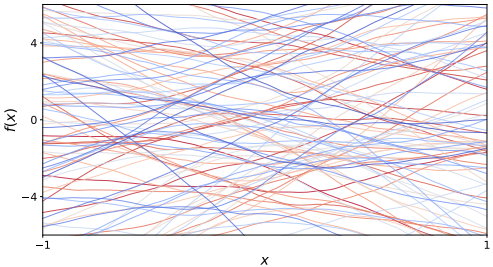




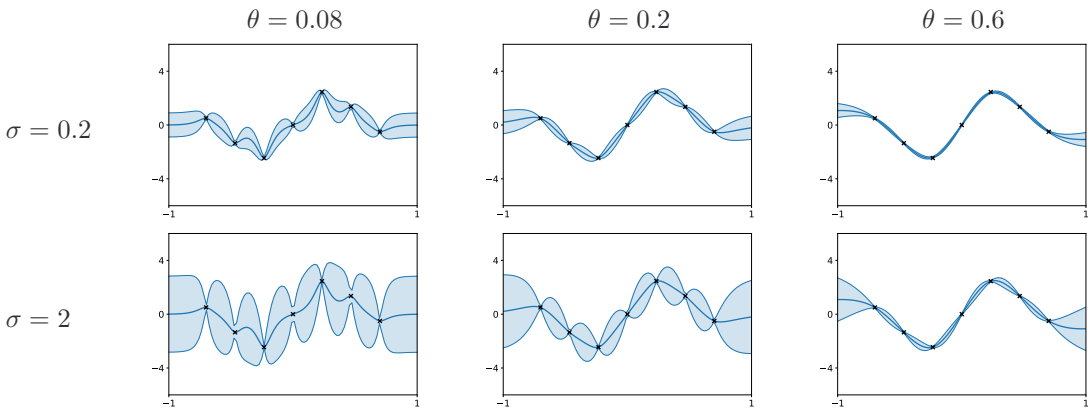
This is because changing kernel means changing prior (Matérn 1/2, 5/2 and Arccosine kernels)



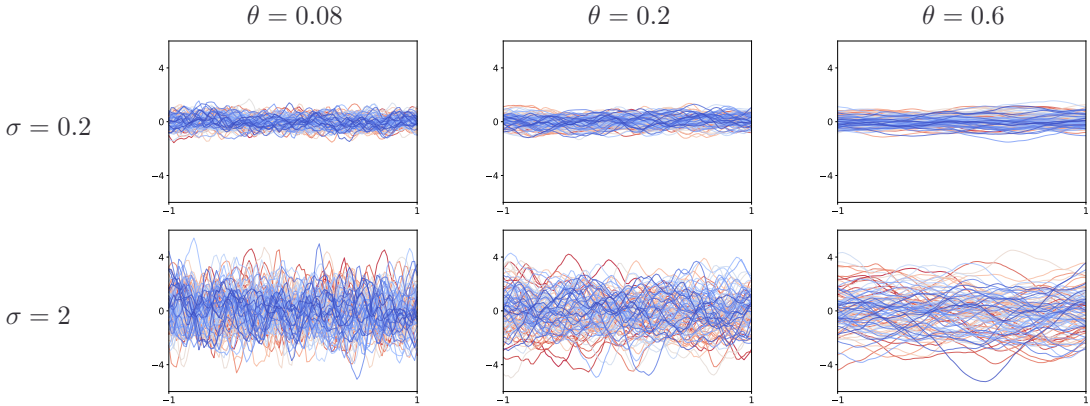




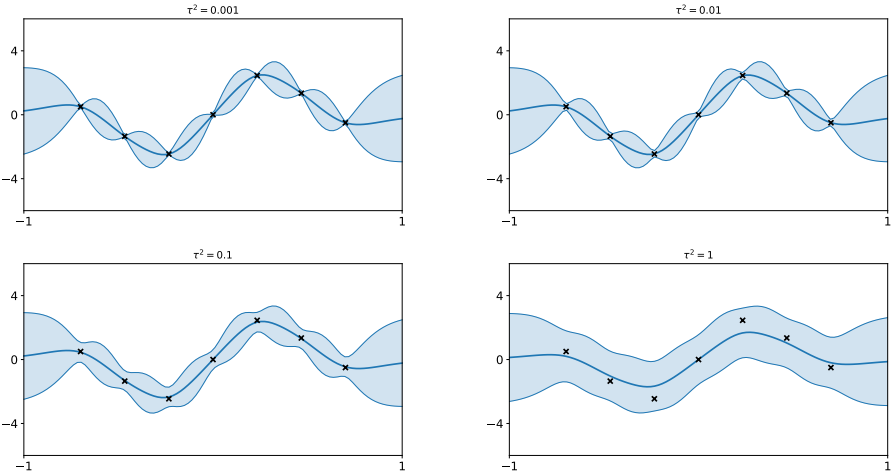
Similarly, changing the kernel parameters has a huge impact on the model:



Again, this is because it means changing the hypothesis we include in our model



Finally the noise variance τ^2 also has a big influence



Parameter estimation and Model validation

The choice of the kernel parameters has a great influence on the model.
⇒ Demo https://durrande.shinyapps.io/gp_playground

In order to choose a prior that is suited to the data at hand, we can search for the parameters that maximise the model likelihood.

Definition

The likelihood of a distribution with a density p_X given some observations $X_1, \dots, X_n$ is:

$$L = \prod_{i=1}^{n} p_X(X_i)$$

In the GPR context, we often have only one observation of the vector Y. The likelihood is then:

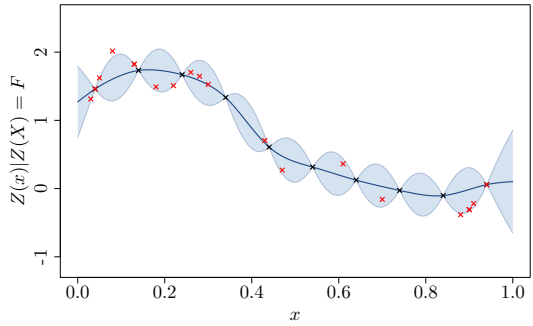
$$L(\sigma^2, \theta) = p_{f(X)}(Y) = \frac{1}{(2\pi)^{n/2}|k(X,X)|^{1/2}} \exp\left(-\frac{1}{2}Y^T k(X,X)^{-1} Y\right).$$

It is thus possible to maximise L – or $\log(L)$ – with respect to the kernel's parameters in order to find a well suited prior.

Why is the likelihood linked to good model predictions? They are linked by the product rule:

$$p_{f(X)}(Y) = p(Y_1) \times p(Y_2|Y_1) \times p(Y_3|Y_1, Y_2) \times \cdots \times p(Y_n|Y_1, \ldots, Y_{n-1})$$

The idea is to introduce new data and to compare the model prediction with reality



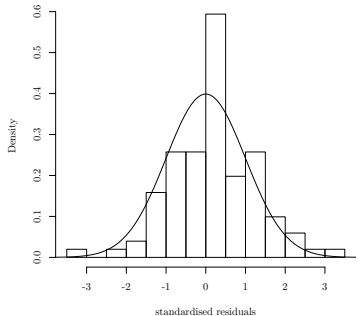
Two (ideally three) things should be checked:

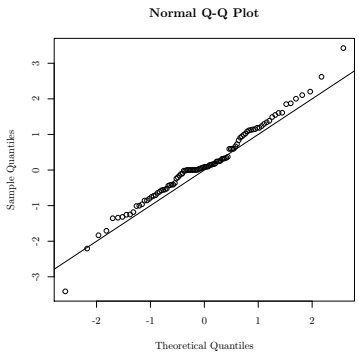
+ Is the mean accurate?
  ⇒ Mean square error (0.038 for the plot above)
+ Do the confidence intervals make sense?
  ⇒ Percentage of points in confidence intervals...
+ Are the predicted covariances right?

The predicted distribution can be tested by normalising the residuals.

According to the model, $F_t \sim \mathcal{N}(m(X_t), c(X_t, X_t))$.

$c(X_t, X_t)^{-1/2}(F_t - m(X_t))$ should thus be independents $\mathcal{N}(0, 1)$:





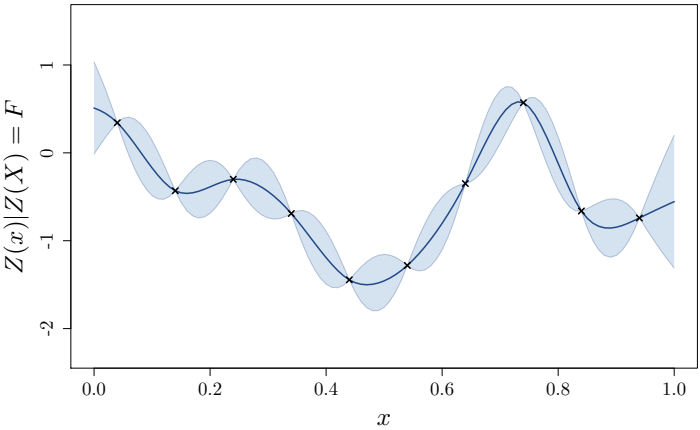
When no test set is available, another option is to consider cross validation methods such as leave-one-out.

The steps are:

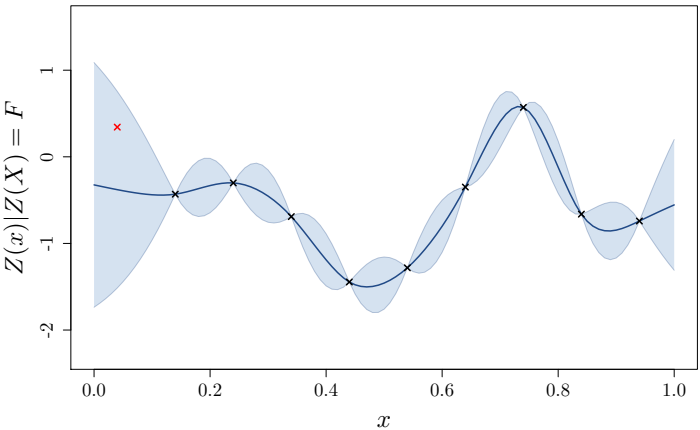
1. build a model based on all observations except one
2. compute the model error at this point

This procedure can be repeated for all the design points in order to get a vector of error.

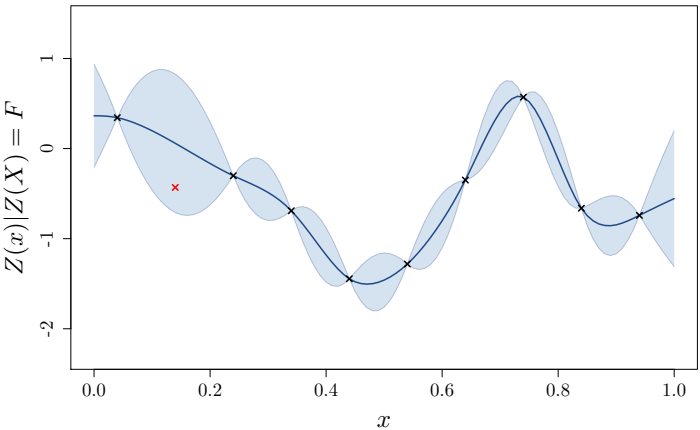
Model to be tested:



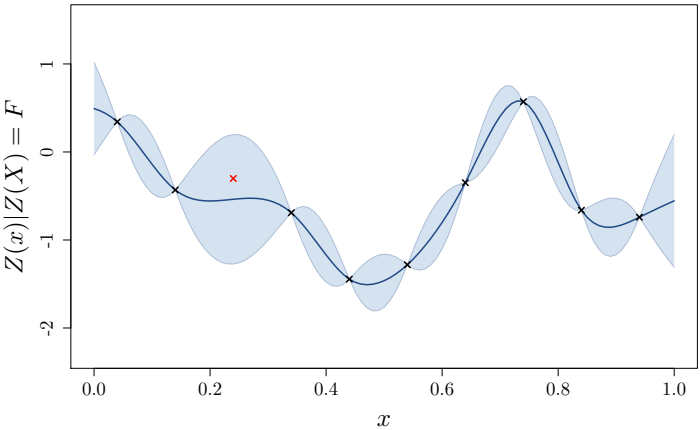
Step 1:



Step 2:



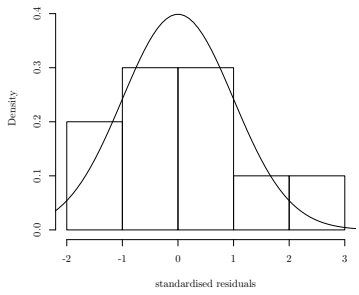
Step 3:

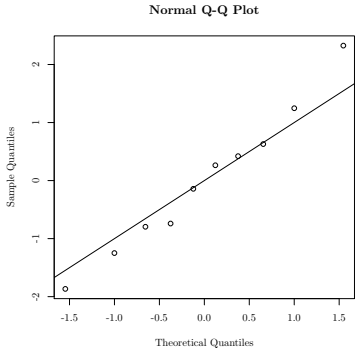


We finally obtain:

$$MSE = 0.24 \text{ and } Q_2 = 0.34.$$

We can also look at the residual distribution, but computing their joint distribution is not as straightforward as previously.





Choosing the kernel

In order to choose a kernel, one should gather all possible informations about the function to approximate...

+ Is it stationary?
+ Is it differentiable, what's its regularity?
+ Do we expect particular trends?
+ Do we expect particular patterns (periodicity, cycles, additivity)?

It is common to try various kernels and to asses the model accuracy (test set or leave-one-out).

Furthermore, it is often interesting to try some input remapping such as $x \to \log(x)$, $x \to \exp(x)$, ...

We have seen previously:

Theorem (Loeve)

k corresponds to the covariance of a GP

$$\Updownarrow$$

k is a symmetric positive semi-definite function

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j k(x_i, x_j) \geq 0$$

for all $n \in \mathbb{N}$, for all $x_i \in D$, for all $\alpha_i \in \mathbb{R}$.

For a few kernels, it is possible to prove they are psd directly from the definition.

+ $k(x, y) = \delta_{x,y}$
+ $k(x, y) = 1$

For most of them a direct proof from the definition is not possible. The following theorem is helpful for stationary kernels:

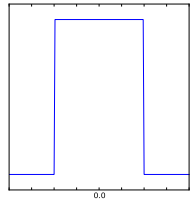
Theorem (Bochner)

A continuous stationary function $k(x, y) = \tilde{k}(|x - y|)$ is positive definite if and only if $\tilde{k}$ is the Fourier transform of a finite positive measure:

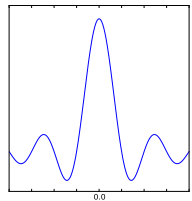
$$\tilde{k}(t) = \int_{\mathbb{R}} e^{-i\omega t} \mathsf{d}\mu(\omega)$$

Example

We consider the following measure:



Its Fourier transform gives $\tilde{k}(t) = \dfrac{\sin(t)}{t}$:



As a consequence, $k(x, y) = \dfrac{\sin(x - y)}{x - y}$ is a valid covariance function.

Usual kernels

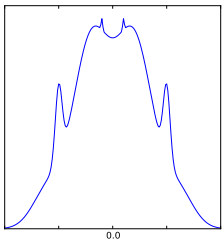
Bochner theorem can be used to prove the positive definiteness of many usual stationary kernels

+ The Gaussian is the Fourier transform of itself
  $\Rightarrow$ it is psd.
+ Matérn kernels are the Fourier transforms of $\frac{1}{(1+\omega^2)^p}$
  $\Rightarrow$ they are psd.

Unusual kernels

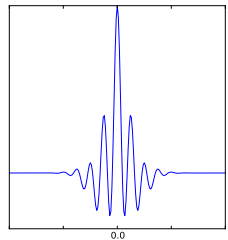
Inverse Fourier transform of a (symmetrised) sum of Gaussian gives (A. Wilson, ICML 2013):

$\mu(\omega)$



$\overrightarrow{\mathcal{F}}$

$\tilde{k}(t)$

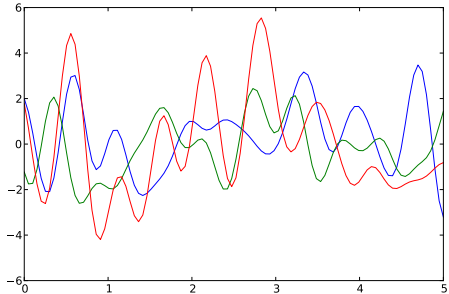


The obtained kernel is parametrised by its spectrum.

More details this afternoon $\Rightarrow$ Talk from Markus Heinonen

Unusual kernels

The sample paths have the following shape:



Making new from old

Making new from old

Kernels can be:

+ Summed together
  - On the same space $k(x, y) = k_1(x, y) + k_2(x, y)$
  - On the tensor space $k(\mathbf{x}, \mathbf{y}) = k_1(x_1, y_1) + k_2(x_2, y_2)$
+ Multiplied together
  - On the same space $k(x, y) = k_1(x, y) \times k_2(x, y)$
  - On the tensor space $k(\mathbf{x}, \mathbf{y}) = k_1(x_1, y_1) \times k_2(x_2, y_2)$
+ Composed with a function
  - $k(x, y) = k_1(f(x), f(y))$

All these operations will preserve the positive definiteness.

How can this be useful?

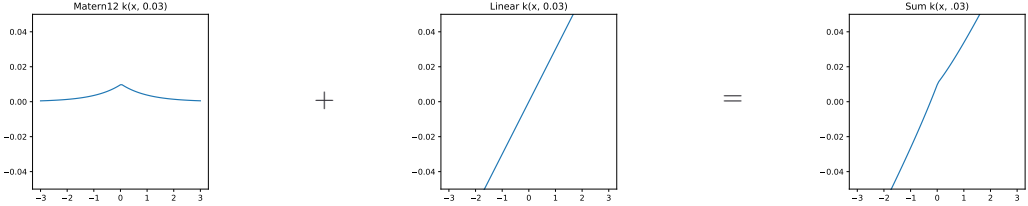
Sum of kernels over the same input space

Property

$$k(x, y) = k_1(x, y) + k_2(x, y)$$

is a valid covariance structure.

Example

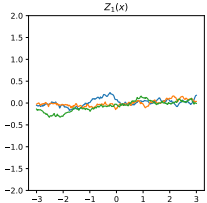


Sum of kernels over the same input space

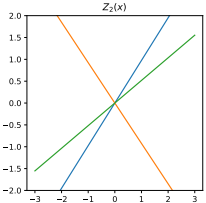
$Z \sim \mathcal{N}(0, k_1 + k_2)$ can be seen as $Z = Z_1 + Z_2$ where $Z_1,\ Z_2$ are indenpendent and $Z_1 \sim \mathcal{N}(0, k_1)$, $Z_2 \sim \mathcal{N}(0, k_2)$

$$k(x, y) = k_1(x, y) + k_2(x, y)$$

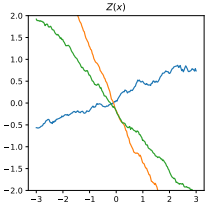
Example



+



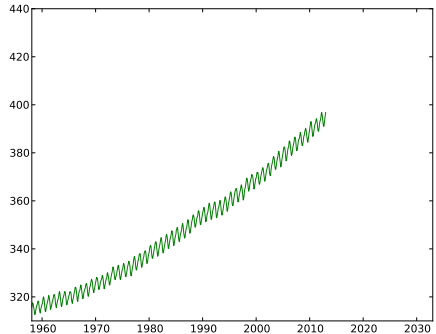
=



Sum of kernels over the same space

Example: The Mauna Loa observatory dataset [GPML 2006]

This famous dataset compiles the monthly CO_2 concentration in Hawaii since 1958.

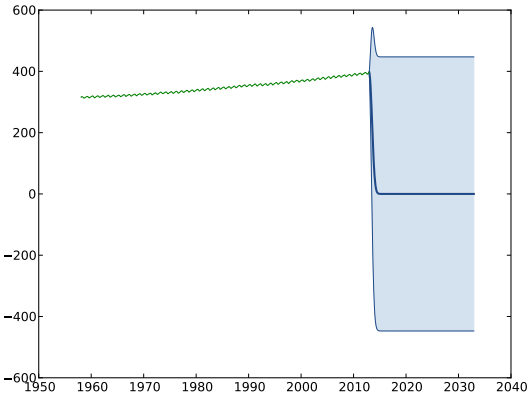


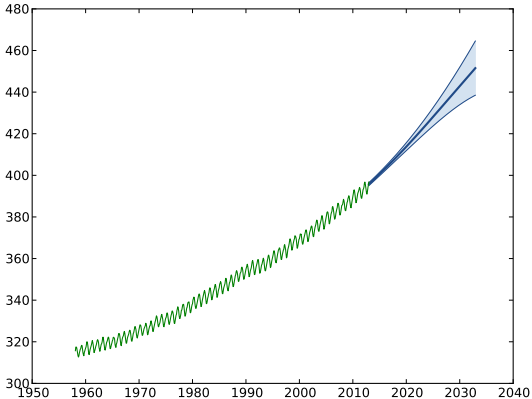
Let's try to predict the concentration for the next 20 years.

Sum of kernels over the same space

We first consider a squared-exponential kernel:

$$k(x, y) = \sigma^2 \exp\left(-\frac{(x-y)^2}{\theta^2}\right)$$





Sum of kernels over the same space

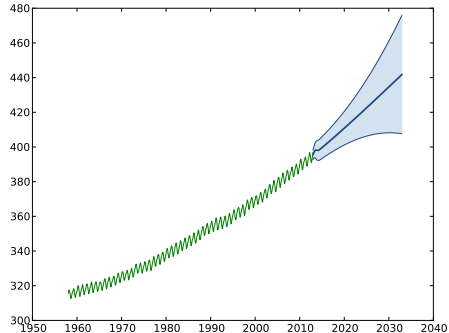
What happen if we sum both kernels?

$$k(x, y) = k_{rbf1}(x, y) + k_{rbf2}(x, y)$$

Sum of kernels over the same space

What happen if we sum both kernels?

$$k(x, y) = k_{rbf1}(x, y) + k_{rbf2}(x, y)$$



The model is drastically improved!

Sum of kernels over the same space

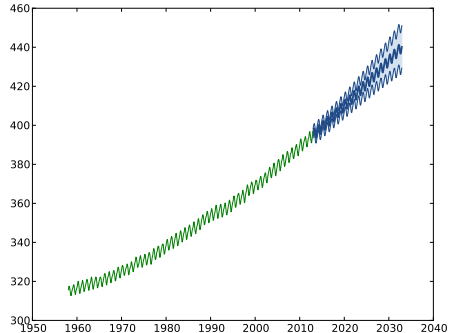
We can try the following kernel:

$$k(x, y) = \sigma_0^2 x^2 y^2 + k_{rbf1}(x, y) + k_{rbf2}(x, y) + k_{per}(x, y)$$

Sum of kernels over the same space

We can try the following kernel:

$$k(x, y) = \sigma_0^2 x^2 y^2 + k_{rbf1}(x, y) + k_{rbf2}(x, y) + k_{per}(x, y)$$



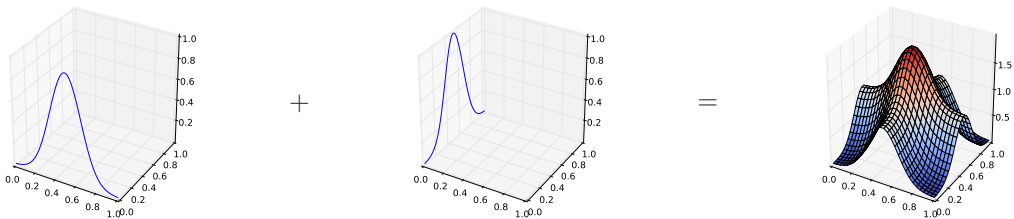
Once again, the model is significantly improved.

Sum of kernels over tensor space

Property

$$k(\mathrm{x}, \mathrm{y}) = k_1(x_1, y_1) + k_2(x_2, y_2)$$

is a valid covariance structure.

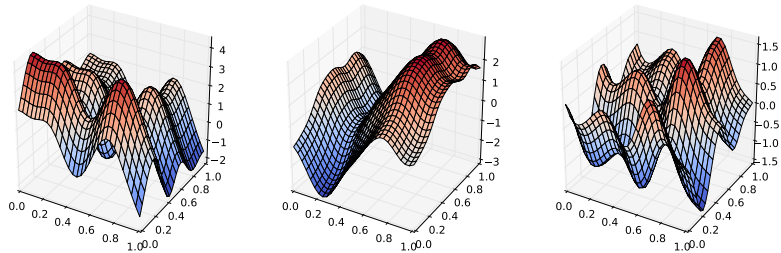


Remark:

From a GP point of view, k is the kernel of $Z(\mathrm{x}) = Z_1(x_1) + Z_2(x_2)$

Sum of kernels over tensor space

We can have a look at a few sample paths from Z:



$\Rightarrow$ They are additive (up to a modification)

Tensor Additive kernels are very useful for

+ Approximating additive functions
+ Building models over high dimensional input space

Sum of kernels over tensor space

Remarks

+ It is straightforward to show that the mean predictor is additive

$$\begin{aligned} m(\mathbf{x}) &= (k_1(x, X) + k_2(x, X))k(X, X)^{-1}F \\ &= \underbrace{k_1(x_1, X_1)k(X, X)^{-1}F}_{m_1(x_1)} + \underbrace{k_2(x_2, X_2)k(X, X)^{-1}F}_{m_2(x_2)} \end{aligned}$$

$\Rightarrow$ The model shares the prior behaviour.

+ The sub-models can be interpreted as GP regression models with observation noise:

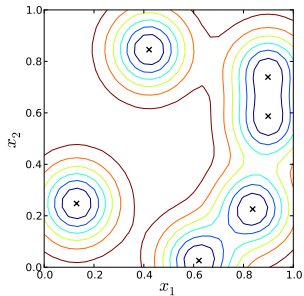
$$m_1(x_1) = \mathsf{E}\left(Z_1(x_1) \mid Z_1(X_1) + Z_2(X_2) = F \right)$$

Sum of kernels over tensor space

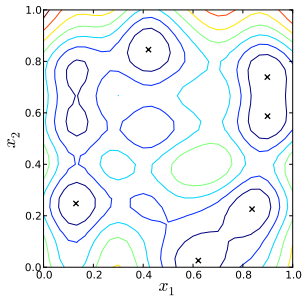
Remark

+ The prediction variance has interesting features

pred. var. with kernel product

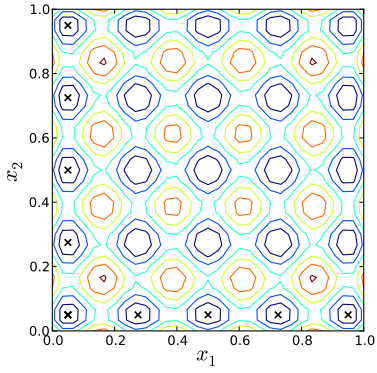


pred. var. with kernel sum



Sum of kernels over tensor space

This property can be used to construct a design of experiment that covers the space with only $cst \times d$ points.



Prediction variance

Product over the same space

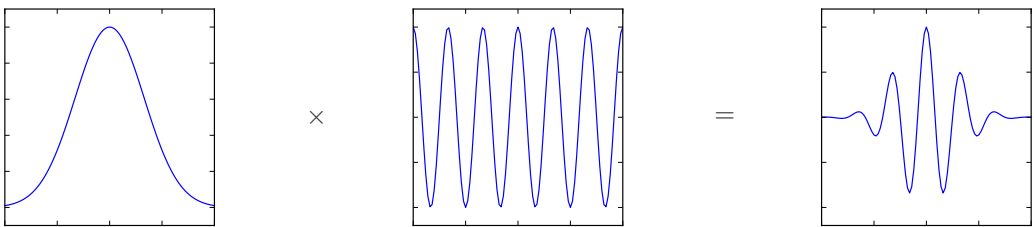
Property

$$k(x, y) = k_1(x, y) \times k_2(x, y)$$

is valid covariance structure.

Example

We consider the product of a squared exponential with a cosine:



Product over the tensor space

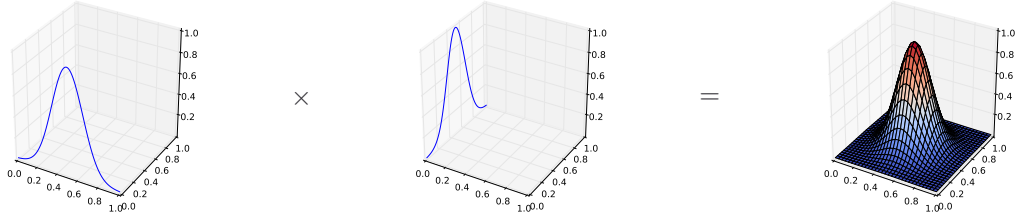
Property

$$k(\mathbf{x}, \mathbf{y}) = k_1(x_1, y_1) \times k_2(x_2, y_2)$$

is valid covariance structure.

Example

We multiply two squared exponential kernels



Calculation shows we obtain the usual 2D squared exponential kernels.

Composition with a function

Property

Let k_1 be a kernel over $D_1 \times D_1$ and f be an arbitrary function $D \to D_1$, then

$$k(x, y) = k_1(f(x), f(y))$$

is a kernel over $D \times D$.
proof

$$\sum\sum a_i a_j k(x_i, x_j) = \sum\sum a_i a_j k_1(\underbrace{f(x_i)}_{y_i}, \underbrace{f(x_j)}_{y_j}) \geq 0$$

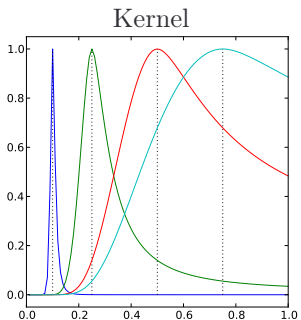
Remarks:

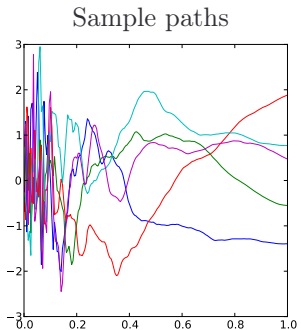
+ k corresponds to the covariance of $Z(x) = Z_1(f(x))$
+ This can be seen as a (nonlinear) rescaling of the input space

Example

We consider $f(x) = \frac{1}{x}$ and a Matérn 3/2 kernel $k_1(x, y) = (1 + |x - y|)e^{-|x-y|}$.

We obtain:





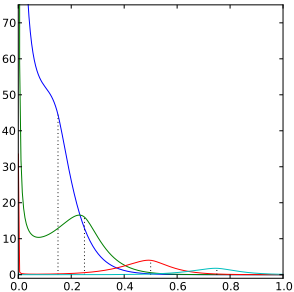
All these transformations can be combined!

Example

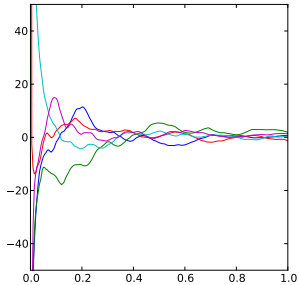
$k(x, y) = f(x)f(y)k_1(x, y)$ is a valid kernel.

This can be illustrated with $f(x) = \frac{1}{x}$ and $k_1(x, y) = (1 + |x - y|)e^{-|x-y|}$:

Kernel



Sample paths



Can we automate the construction of the covariance?

Automatic statistician [Duvenaud 2013, Steinruecken 2019]

It considers a set of possible

+ kernel functions
+ kernel combinations ($+$, $\times$, change-point)

and uses a greedy approach to find the kernel that minimises

$$BIC = -2\log(L) + \#_{param}\log(n)$$

The automatic statistician also generates human readable reports!

Kernel identification through transformers [preprint Simpson 2021]

idea: train a transformer neural network to add the kernel name as a label to kernel samples

+ Uses a vocabulary of kernels and a grammar to combine them
+ Outputs the probability associated to various kernel combinations

Conclusion: GPR and kernel design in practice

The various steps for building a GPR model are:

1. Get the Data (Design of Experiment)
   - What is the overall evaluation budget?
   - What is my model for?

2. Choose a kernel. Do we have any specific knowledge we can include in it?

3. Estimate the parameters
   - Maximum likelihood
   - Cross-validation
   - Multi-start

4. Validate the model
   - Test set
   - Leave-one-out to check mean and confidence intervals
   - Leave-k-out to check predicted covariances

Remark

It is common to iterate over steps 2, 3 and 4.

In practice, the following errors may appear:

- Error: Cholesky decomposition failed
- Error: the matrix is not positive definite

Invertibility issues arise typically when

+ observations are close-by
+ the kernel corresponds to very regular sample paths (squared-exponential for example)
+ the range (or length-scale) parameters are large

In order to avoid numerical problems during optimization, one can:

+ add some (very) small observation noise (jitter or nugget)
+ impose a maximum bound to length-scales
+ impose a minimal bound for noise variance
+ avoid using the Gaussian kernel


Secondmind