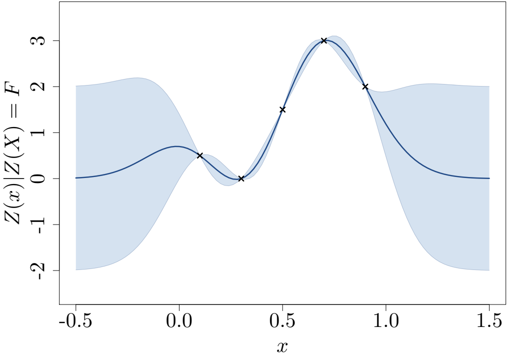
Short course on Statistical Modelling for Optimization – lecture 3/4

Gaussian Process regression

June 2016 – Universidad Tecnológica de Pereira – Colombia

Nicolas Durrande (durrande@emse.fr)
Jean-Charles Croix (jean-charles.croix@emse.fr)
Mines St-Étienne – France

We have seen on day 1 how to build Gaussian process regression models:



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

We will discuss these models in more details today.

Outline of todays lecture

- Parameters estimation
- Model validation
- Kernel designs
- “Exotic” informations

Parameter estimation

We have seen previously that the choice of the kernel and its parameters have a great influence on the model.

In order to choose a prior that is suited to the data at hand, we can consider:

- minimising the model error
- Using maximum likelihood estimation

We will now detail the second one.

Definition

The **likelihood** of a distribution with a density f_X given some observations $X_1, \ldots, X_p$ is:

$$L = \prod_{i=1}^{p} f_X(X_i)$$

This quantity can be used to measure the adequacy between observations and a distribution.

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

$$L = f_{Z(X)}(F) = \frac{1}{(2\pi)^{n/2}|k(X,X)|^{1/2}} \exp\left(-\frac{1}{2}F^t k(X,X)^{-1}F\right).$$

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.

Example

We consider 100 sample from a Matern 5/2 process with parameters $\sigma^2 = 1$ and $\theta = 0.2$, and n observation points. We then try to recover the kernel parameters using MLE.

n	5	10	15	20
σ^2	1.0 (0.7)	1.11 (0.71)	1.03 (0.73)	0.88 (0.60)
θ	0.20 (0.13)	0.21 (0.07)	0.20 (0.04)	0.19 (0.03)

MLE can be applied regardless to the dimension of the input space.

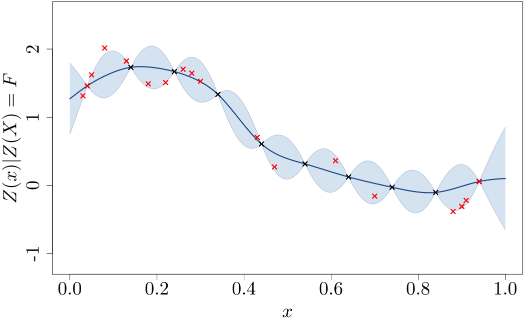
Model validation

We have seen that given some observations $F = f(X)$, it is very easy to build lots of models, either by changing the kernel parameters or the kernel itself.

The interesting question now is to know how to get a good model. To do so, we will need to answer the following questions:

- What is a good model?
- How to measure it?

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



Since GPR models provide a mean and a covariance structure for the error they both have to be assessed.

Let X_t be the test set and $F_t = f(X_t)$ be the associated observations.

The accuracy of the mean can be measured by computing:

$$\text{Mean Square Error} \qquad MSE = \mathrm{mean}((F_t - m(X_t))^2)$$

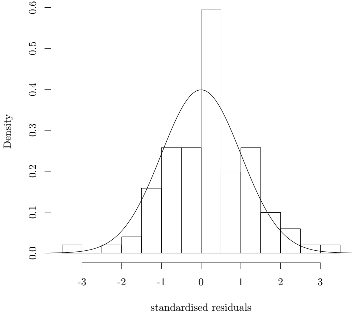
$$\text{A ``normalised'' criterion} \qquad Q_2 = 1 - \frac{\sum (F_t - m(X_t))^2}{\sum (F_t - \mathrm{mean}(F_t))^2}$$

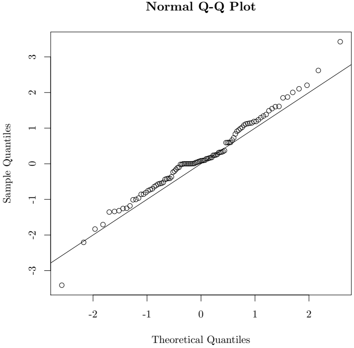
On the above example we get $MSE = 0.038$ and $Q_2 = 0.95$.

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 $\mathcal{N}(0, 1)$:





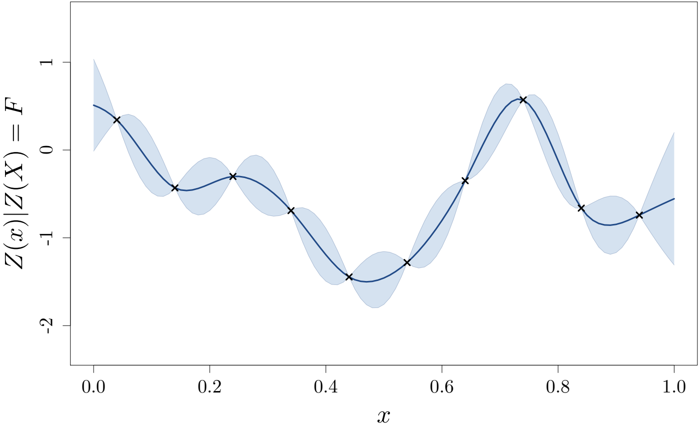
When no test set is available, another option is to consider cross validation methods such as leave-one-out.

The steps are:

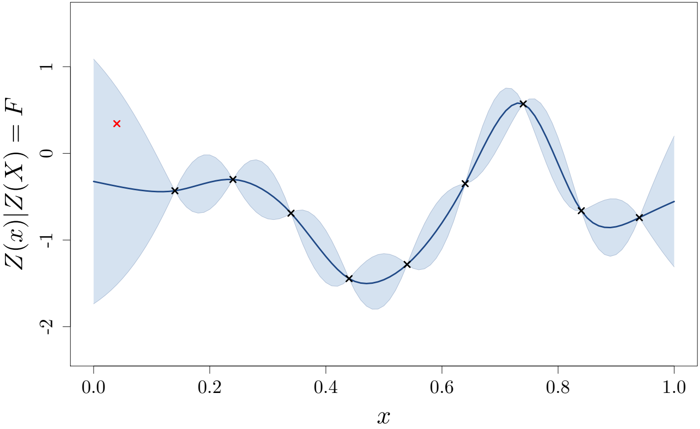
1. build a model based on all observations except one
2. compute the model error at this point

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

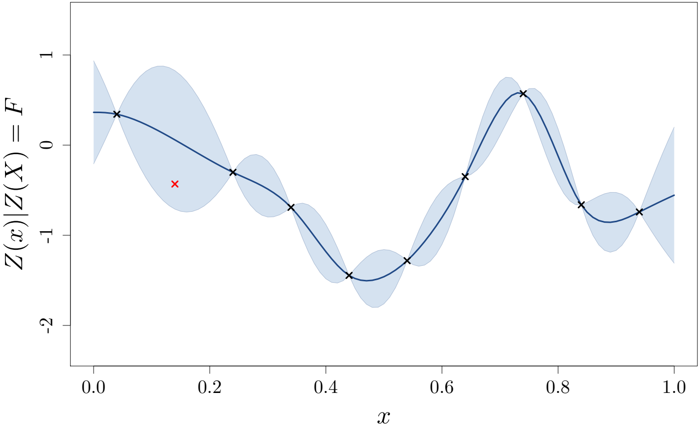
Model to be tested:



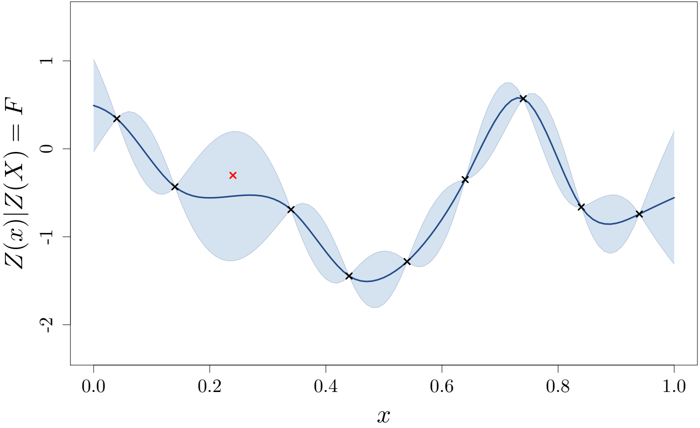
Step 1:



Step 2:



Step 3:



If we apply LOO to previous example we obtain:

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

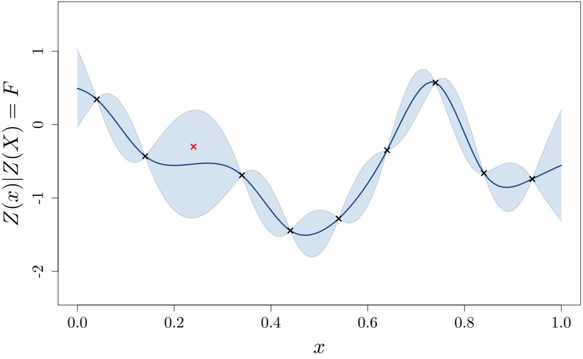
Why doesn't the model perform as good as previously?

If we apply LOO to previous example we obtain:

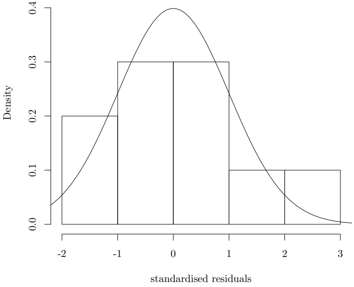
$$MSE = 0.24 \text{ and } Q_2 = 0.34.$$

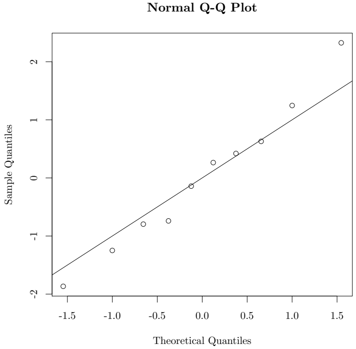
Why doesn't the model perform as good as previously?

It turns out that errors are always computed at the 'worst' location!



We can also look at the residual distribution. For leave-one-out, there is no joint distribution for the residuals so they have to be standardised independently. We obtain:





Making new from old

Making new from old: Many operations can be applied to psd functions while retaining this property

Kernels can be:

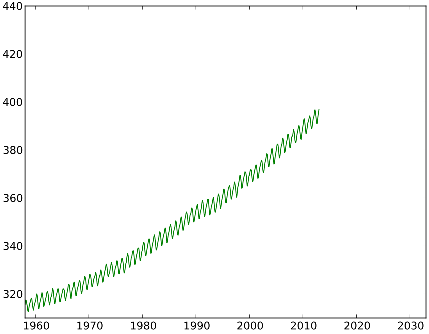
- Summed together
  - On the same space $k(x, y) = k_1(x, y) + k_2(x, y)$
  - On the tensor space $k(x, y) = k_1(x_1, y_1) + k_2(x_2, y_2)$
- Multiplied together
  - On the same space $k(x, y) = k_1(x, y) \times k_2(x, y)$
  - On the tensor space $k(x, y) = k_1(x_1, y_1) \times k_2(x_2, y_2)$
- Composed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

How can this be useful?

Sum of kernels over the same space

Example (The Mauna Loa observatory dataset)

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

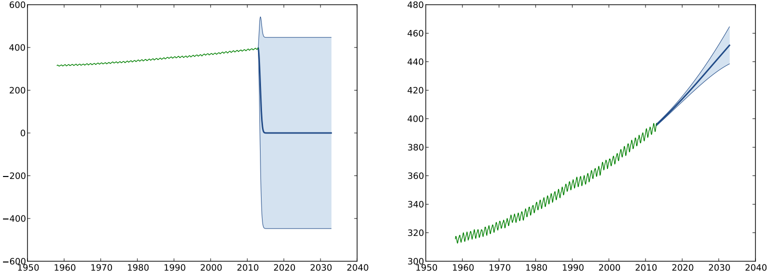


Let's try to predict the concentration for the next 20 years.

Sum of kernels over the same space

We first consider a squared-exponential kernel:

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



The results are terrible!

Sum of kernels over the same space

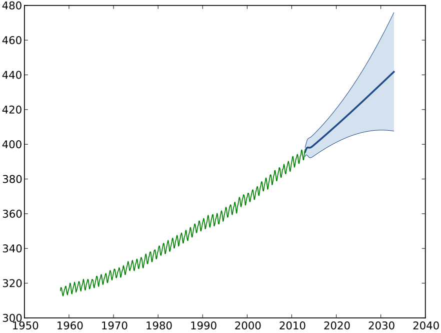
What happen if we sum both kernels?

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

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

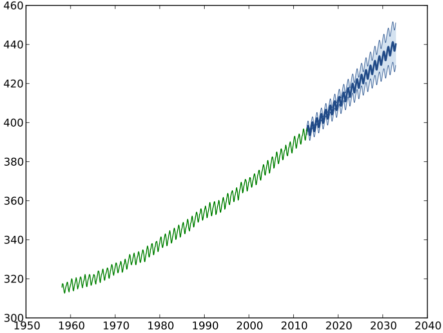
We can try the following kernel:

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

Sum of kernels over the same space

We can try the following kernel:

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



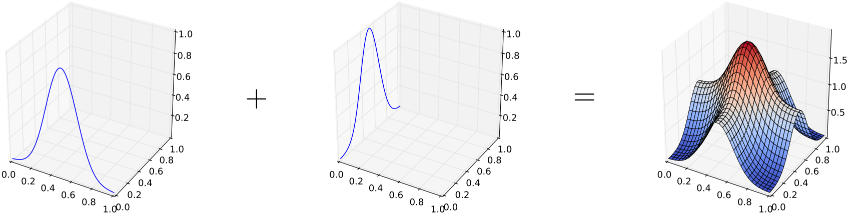
Once again, the model is significantly improved.

Sum of kernels over tensor space

Property

$$k(x, y) = k_1(x_1, y_1) + k_2(x_2, y_2)$$

is valid covariance structure.

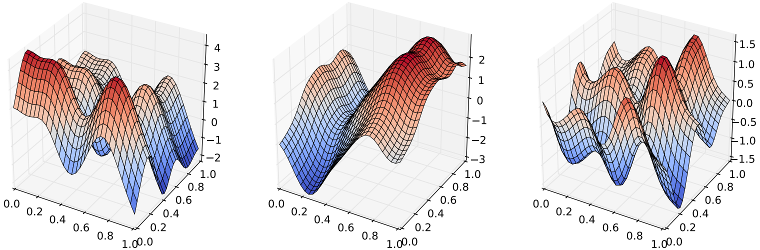


Remark: From a GP point of view, k is the kernel of

$$Z(x) = Z_1(x_1) + Z_2(x_2)$$

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)

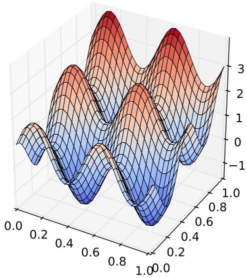
Tensor Additive kernels are very useful for

- Approximating additive functions
- Building models over high dimensional inputs spaces

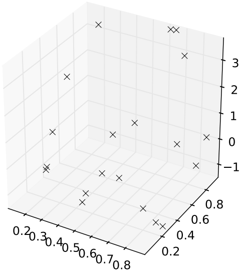
Sum of kernels over tensor space

We consider the test function $f(x) = \sin(4\pi x_1) + \cos(4\pi x_2) + 2x_2$ and a set of 20 observation in $[0,1]^2$.

Test function



Observations

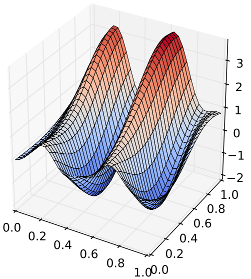


Sum of kernels over tensor space

We obtain the following models:

Gaussian kernel

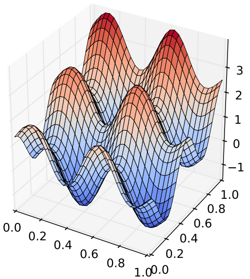
Mean predictor



RMSE is 1.06

Additive Gaussian kernel

Mean predictor



RMSE is 0.12

Sum of kernels over tensor space

Remark

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

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

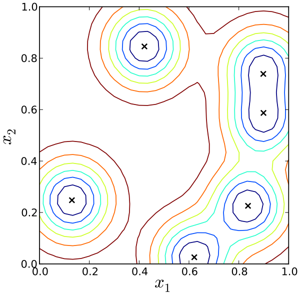
$\Rightarrow$ The mean predictor shares the prior behaviour.

Sum of kernels over tensor space

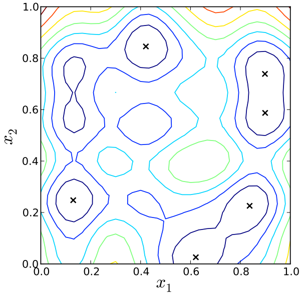
Remark

- The prediction variance has interesting features

pred. var. with kernel product

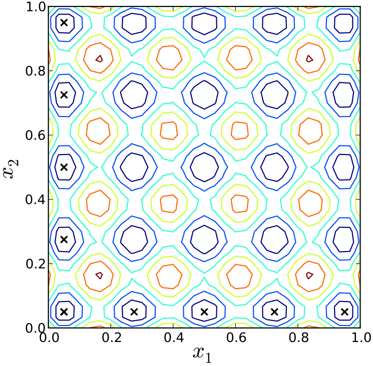


pred. var. with kernel sum



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

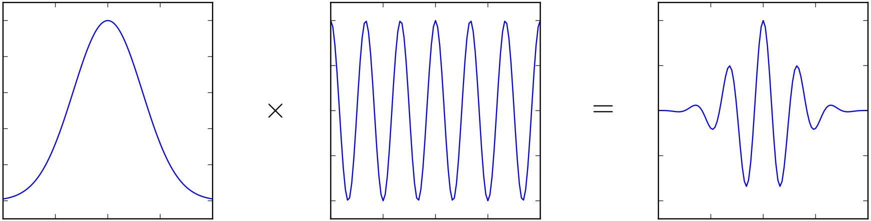
Property

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

is valid covariance structure.

Example

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



Product over the tensor space

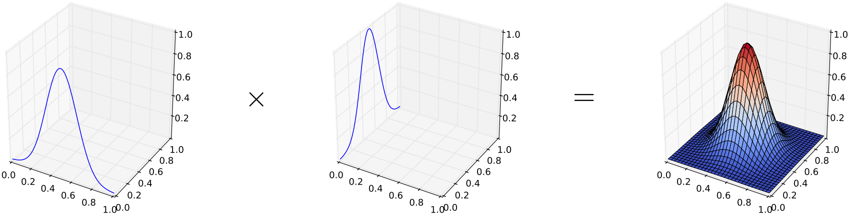
Property

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

is valid covariance structure.

Example

We multiply 2 squared exponential kernel



Calculation shows this is the usual 2D squared exponential kernel.

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

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

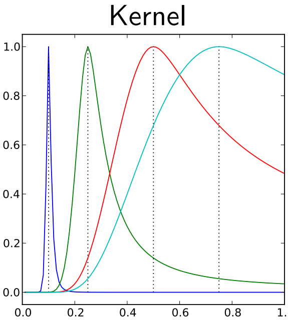
Remarks:

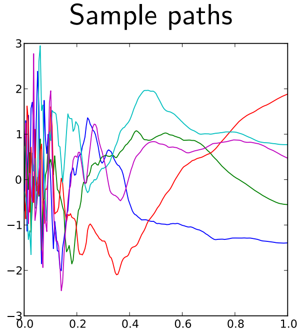
- k corresponds to the covariance of $Z(x) = Z_1(f(x))$
- This can be seen as a (non-linear) rescaling of the input space

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:



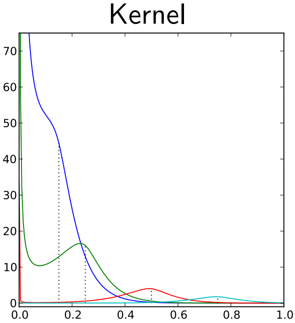


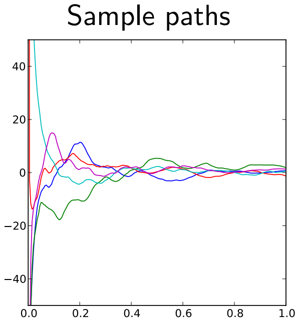
All these transformations can be combined!

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

Bochner's theorem

Theorem (Bochner)

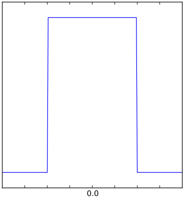
A continuous stationary function $k(x,y) = \tilde{k}(x-y)$ is positive definite if and only if $\tilde{k}$ is the Fourier transform of a finite positive measure:

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

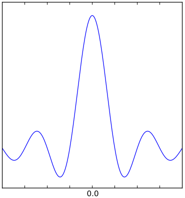
This result is very useful to prove the positive definiteness of stationary functions.

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.

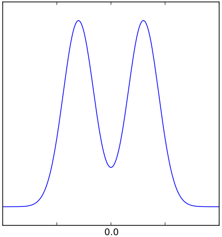
Bochner theorem can be used to prove the positive definiteness of many usual stationary kernels

- The Gaussian is the Fourier transform of itself
  $\Rightarrow$ it is psd.
- Matern kernels are the Fourier transforms of $\frac{1}{(1+\omega^2)^p}$
  $\Rightarrow$ they are psd.
- the constant function is the Fourier transform of $\delta_{x,y}$
  $\Rightarrow$ it is psd.

Spectral approximation with a mixture of Gaussian (A. Wilson, ICML 2013)

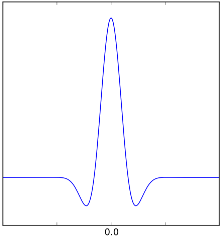
The inverse Fourier transform of a (symmetrised) non centred Gaussian is:

$\mu(\omega)$



$$\xrightarrow[\mathcal{F}]{}$$

$\tilde{k}(t)$

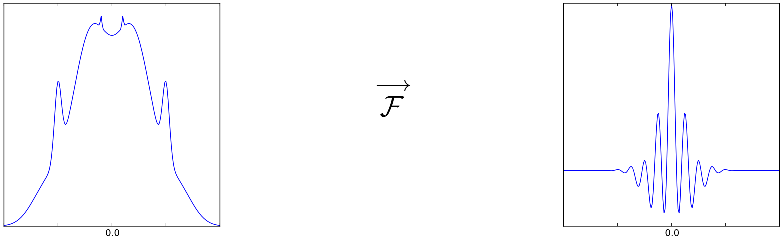


This can be generalised to a measure based on the sum of Gaussians.

Spectral approximation with a mixture of Gaussian (A. Wilson, ICML 2013)

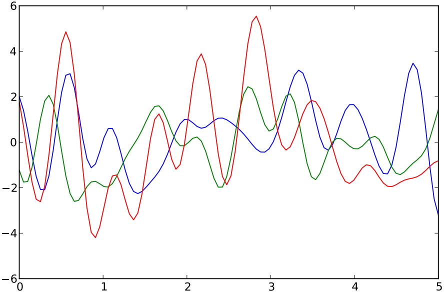
We obtain a kernel that is parametrised by the means and the bandwidths of Gaussian bells in the measure space:

$\mu(\omega)$ $\qquad \underset{\mathcal{F}}{\longrightarrow} \qquad$ $\tilde{k}(t)$



Spectral approximation with a mixture of Gaussian (A. Wilson, ICML 2013)

The sample paths have the following aspect:



Effect of a linear operator

Effect of a linear operator

Property

Let L be a linear operator that commutes with the covariance, then $k(x,y) = L_x(L_y(k_1(x,y)))$ is a kernel.

Example

We want to approximate a function $[0,1] \to \mathbb{R}$ that is symmetric with respect to 0.5. We will consider 2 linear operators:

$$L_1 : f(x) \to \begin{cases} f(x) & x < 0.5 \\ f(1-x) & x \geq 0.5 \end{cases}$$

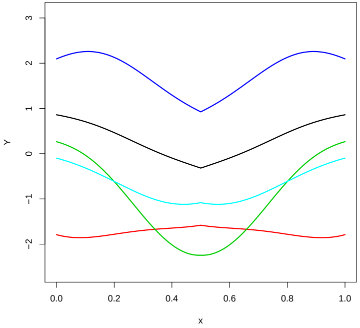
$$L_2 : f(x) \to \frac{f(x) + f(1-x)}{2}.$$

Exercice: Compute the kernel associated to the second operator.

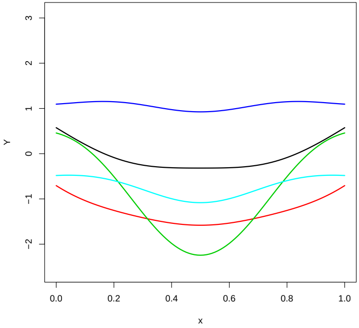
Effect of a linear operator: example [Ginsbourger 2013]

Examples of associated sample paths are

$$k_1 = L_1(L_1(k))$$



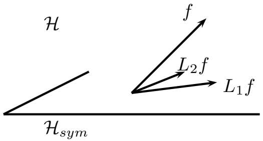
$$k_2 = L_2(L_2(k))$$



The differentiability is not always respected!

Effect of a linear operator

These linear operator are projections onto a space of symmetric functions:



Is there an optimal projection?

⇒ This can be difficult... but it raises interesting questions!

Effect of a linear operator

Can we construct a GP such that the integrals of the paths are null?

We can think of the following application:

$$L : f(x) \rightarrow f(x) - \int f(s) \,\mathrm{d}s.$$

More generally, for all $g : [0,1] \rightarrow \mathbb{R}$, the application

$$L_g : f(x) \rightarrow f(x) - \frac{g(x)}{\int g(s) \,\mathrm{d}s} \int f(s) \,\mathrm{d}s$$

will center f. It turns out that the optimal g is $g(x) = \int k(x,s) \,\mathrm{d}s$

Exercice

1. Compute the associated kernel.
2. What is the distribution of $Z | \int Z = 0$?

Adding “exotic” observations

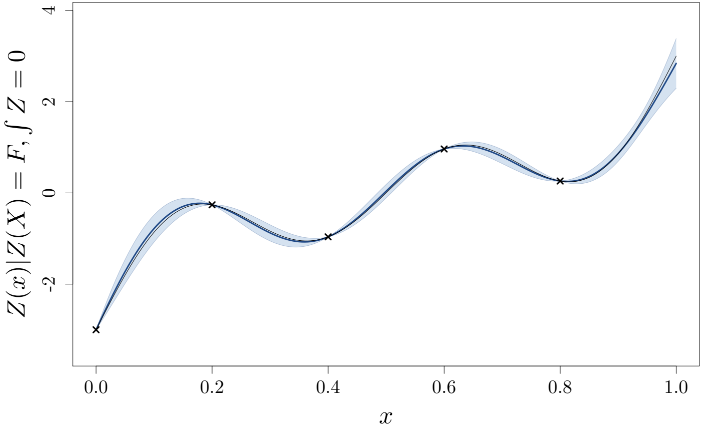
Up to now, we have only considered regular observation of the kind $f(X)$.

According to what we have just seen, the conditioning can also include observation of the integral. This can be generalised to other linear operators such as the derivative:

$$Z \;\Big|\; Z(X) = F, \int Z = a, \frac{\mathrm{d}Z}{\mathrm{d}x}(X') = F'$$

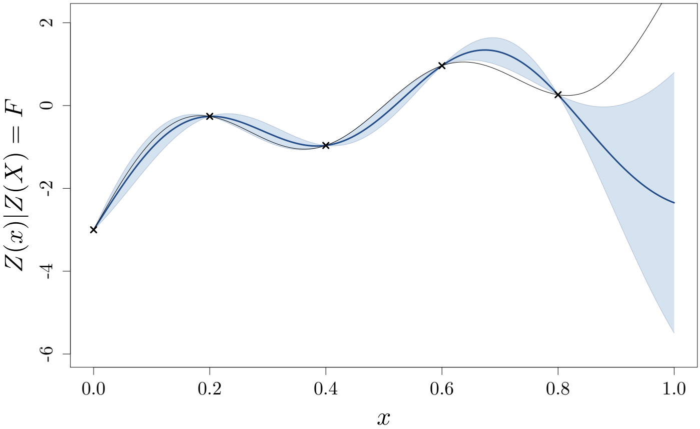
Example

If we take into account that the function is centred, we obtain:

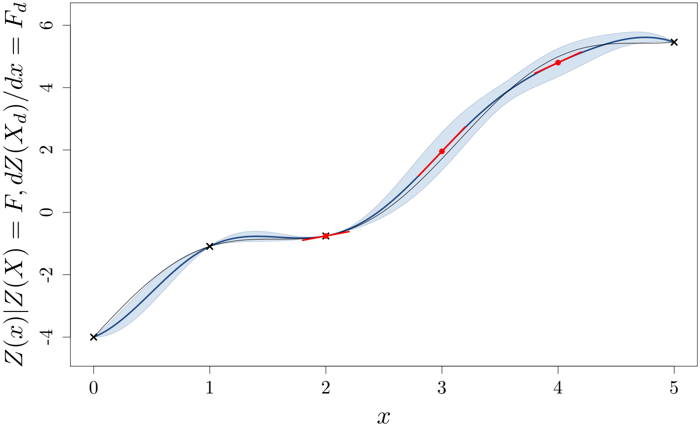


Example

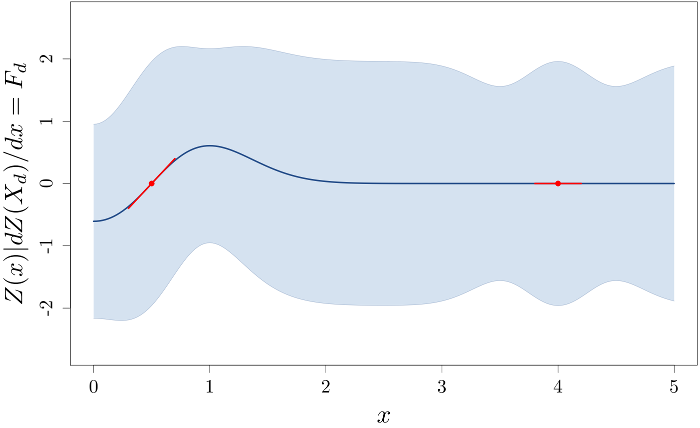
Whereas if we ignore it:



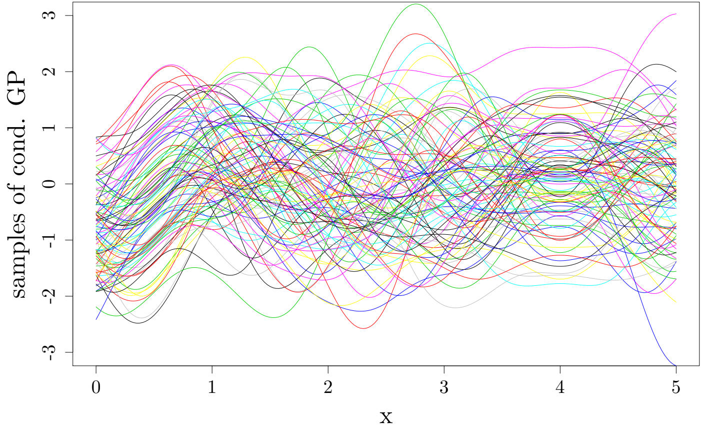
Similarly, we can include in the model some derivatives observations:



We can see interesting behaviour if we look at a model with only derivatives.



As always, we can simulate conditional paths:



Conclusion

Small Recap We have seen that

- It is possible to build as many kernels as you want
  - Given some data, there is not one GP model but an infinity...
- Kernels encode the prior belief on the function to approximate.
  - They can (and should) be tailored to the problem at hand.
- It is possible to include more than regular function observations.
- If you want the decisions based on your model to be reliable, model validation is of the utmost importance.

3 tools for designing new kernels:

Making new from old

Various operations can be applied to kernels while keeping the psd :

- sum
- product
- composition with a function

Bochner Theorem

is very useful to prove a stationary kernel is psd.

Linear operators

If we have a linear application that transforms any function into a function satisfying the desired property, it is possible to build a GP fulfilling the requirements.