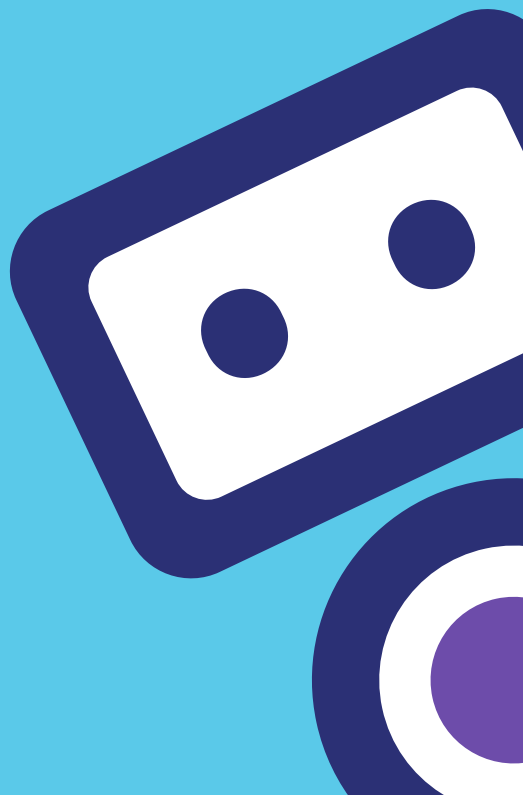
Gaussian Process Summer School

Second introduction to GPs and Kernel Design

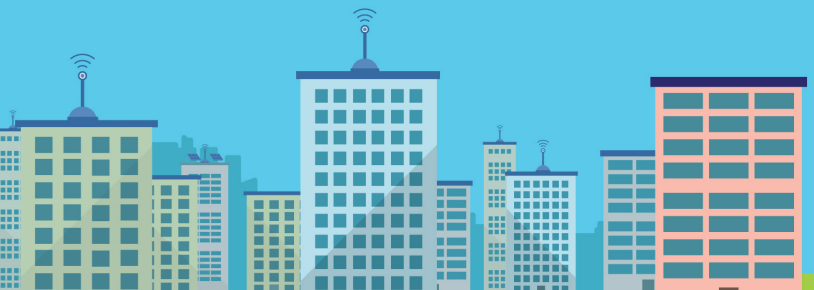
Nicolas Durrande – PROWLER.io

@NicolasDurrande – nicolas@prowler.io

September 2020

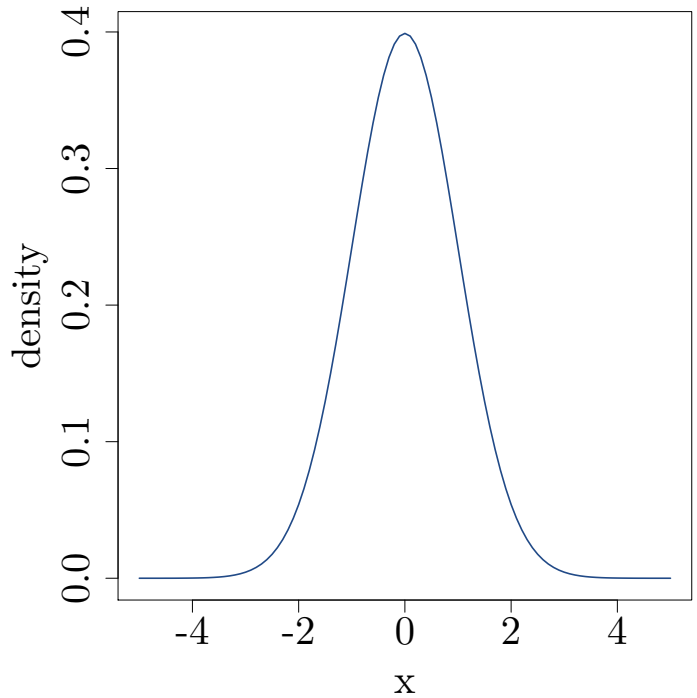


Second Introduction to GPs and GP Regression



The pdf of a Gaussian random variable is:

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



The parameters μ and σ^2 correspond to the mean and variance

$$\mu = \mathrm{E}[X]$$
$$\sigma^2 = \mathrm{E}[X^2] - \mathrm{E}[X]^2$$

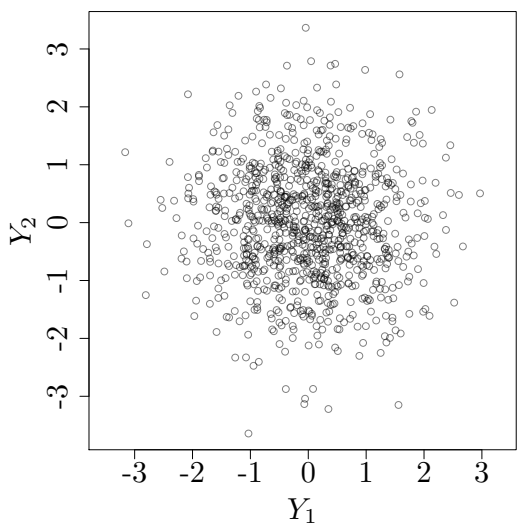
The variance is positive.

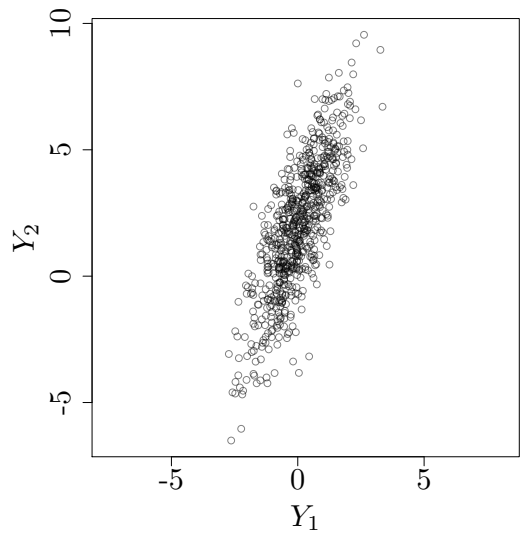
Definition

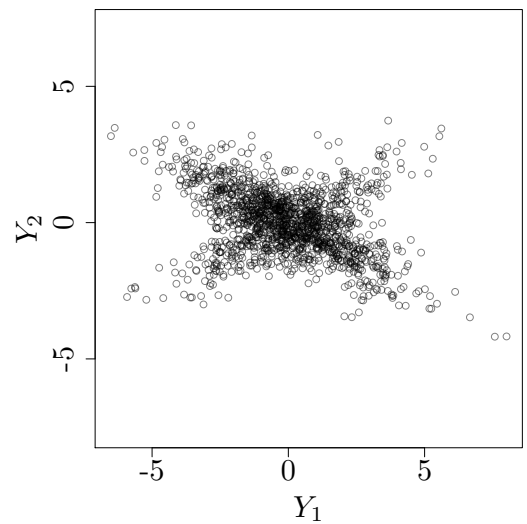
We say that a vector $Y = (Y_1, \ldots, Y_n)^T$ follows a multivariate normal distribution if any linear combination of Y follows a normal distribution:

$$\forall \alpha \in \mathbb{R}^n,\ \alpha^T Y \sim \mathcal{N}$$

Two examples and one **counter-example**:





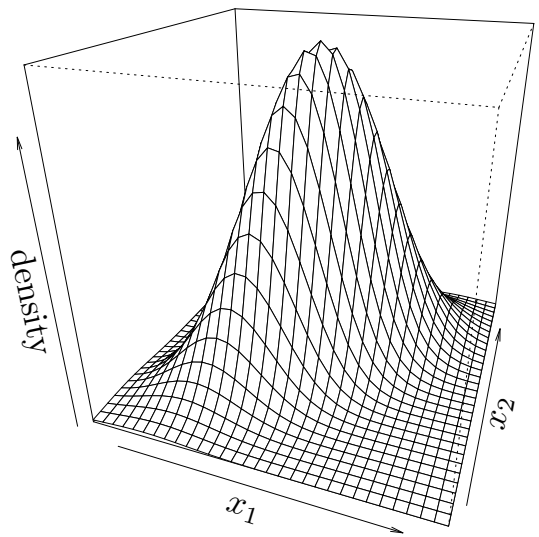


The pdf of a multivariate Gaussian is:

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

It is parametrised by

- mean vector $\mu = \mathrm{E}[Y]$
- covariance matrix
  $\Sigma = \mathrm{E}[YY^T] - \mathrm{E}[Y]\mathrm{E}[Y]^T$
  (i.e. $\Sigma_{i,j} = \mathrm{cov}(Y_i, Y_j)$)



A covariance matrix is **symmetric** $\Sigma_{i,j} = \Sigma_{j,i}$ and **positive semi-definite**

$$\forall \alpha \in \mathbb{R}^n, \alpha^T\Sigma\alpha \geq 0.$$

Conditional distribution

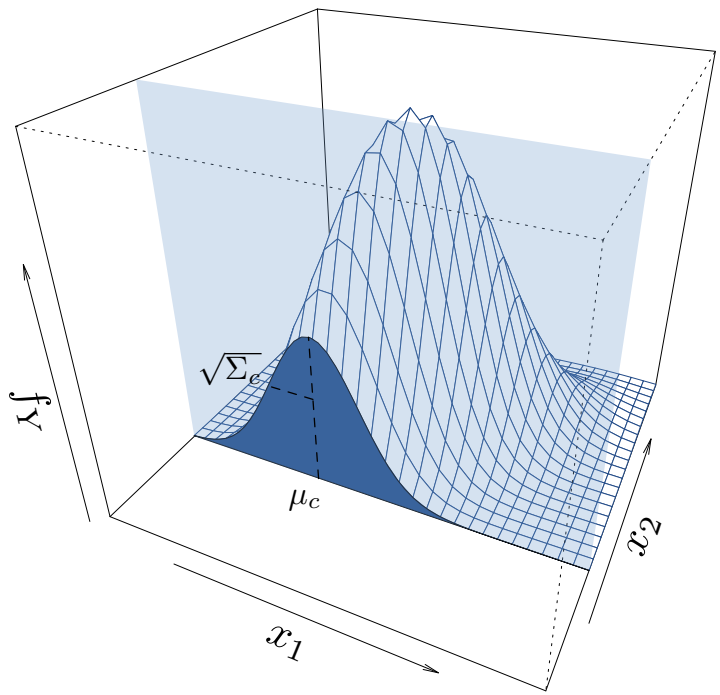
2D multivariate Gaussian conditional distribution:

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

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

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

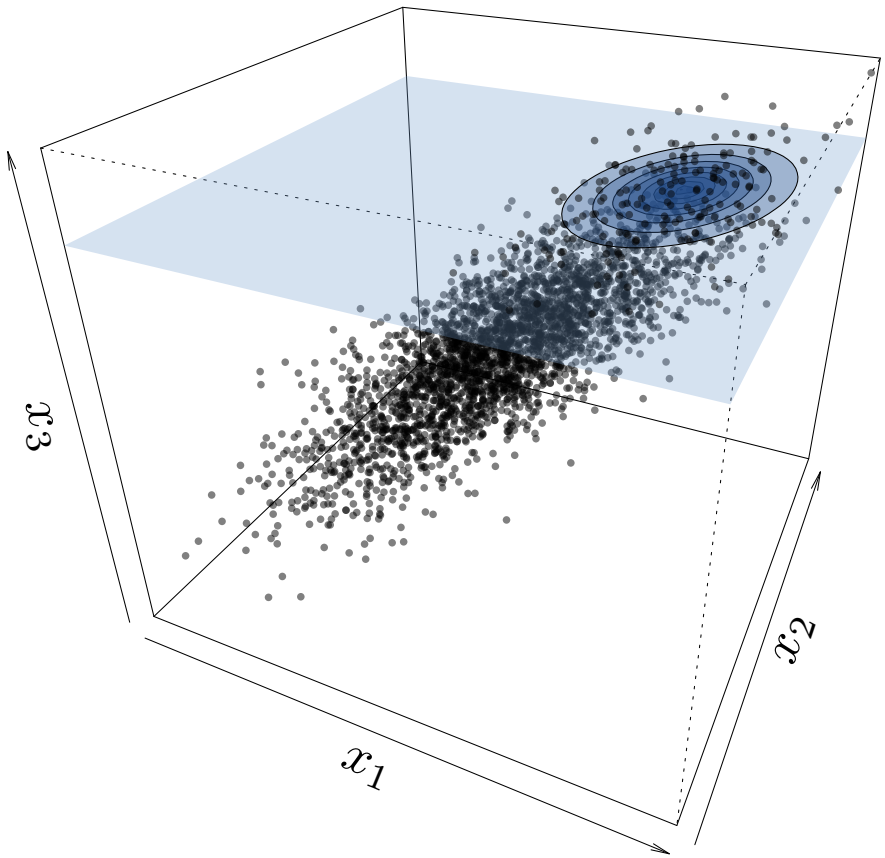
$$= \text{Gaussian distribution!}$$



The conditional distribution is still Gaussian!

3D Example

3D multivariate Gaussian conditional distribution:



Conditional distribution

Let (Y_1, Y_2) be a Gaussian vector (Y_1 and Y_2 may both be vectors):

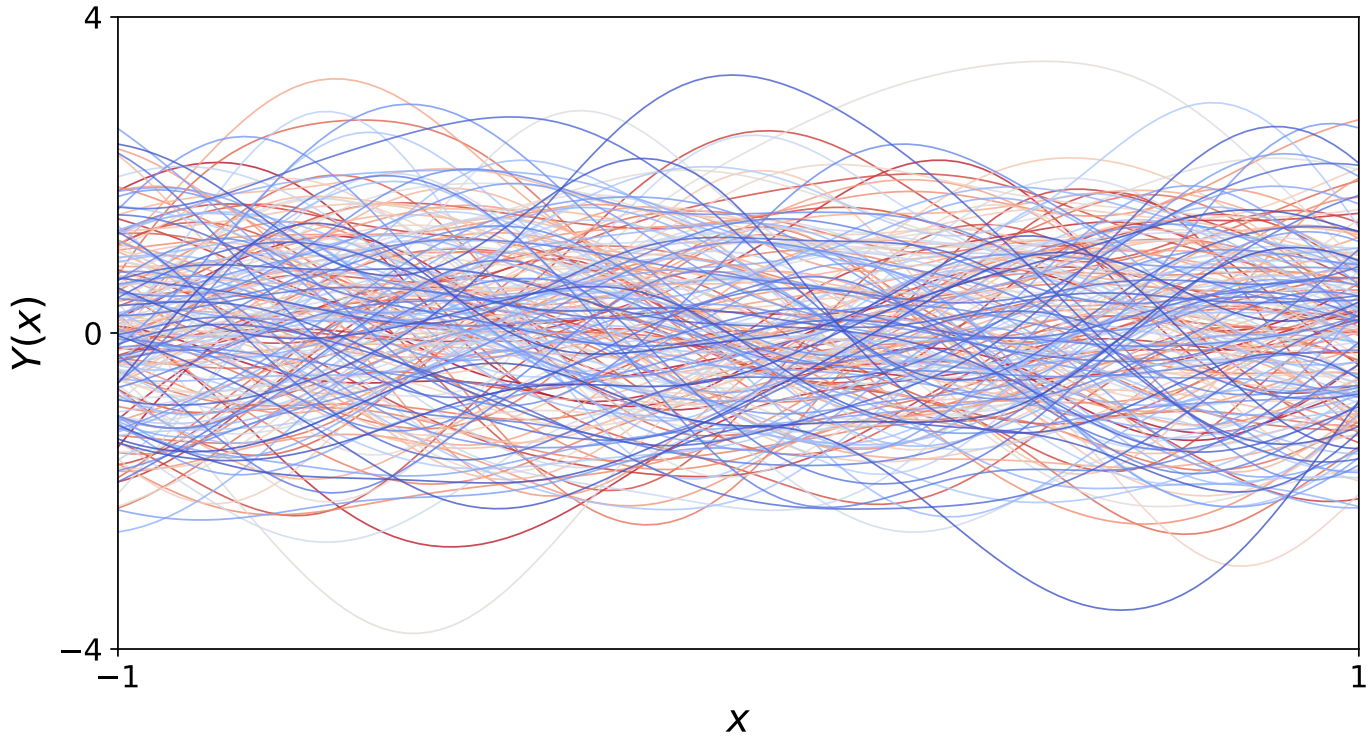
$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \right).$$

The conditional distribution of Y_1 given Y_2 is:

$$Y_1 | Y_2 \sim \mathcal{N}(\mu_{\text{cond}}, \Sigma_{\text{cond}})$$

with
$$\mu_{\text{cond}} = \mathrm{E}[Y_1|Y_2] = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(Y_2 - \mu_2)$$
$$\Sigma_{\text{cond}} = \mathrm{cov}[Y_1, Y_1|Y_2] = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

Gaussian processes



Definition

A random process Z over $D \subset \mathbb{R}^d$ is said to be Gaussian if

$$\forall n \in \mathbb{N}, \forall x_i \in D, (Z(x_1), \ldots, Z(x_n)) \text{ is multivariate normal.}$$

$\Rightarrow$ Demo: https://github.com/awav/interactive-gp

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

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

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

$$k(x, y) = \text{cov}(Z(x), Z(y))$$

The mean m can be any function, but not the kernel:

Theorem (Loeve)

k is a GP covariance

$$\Updownarrow$$

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

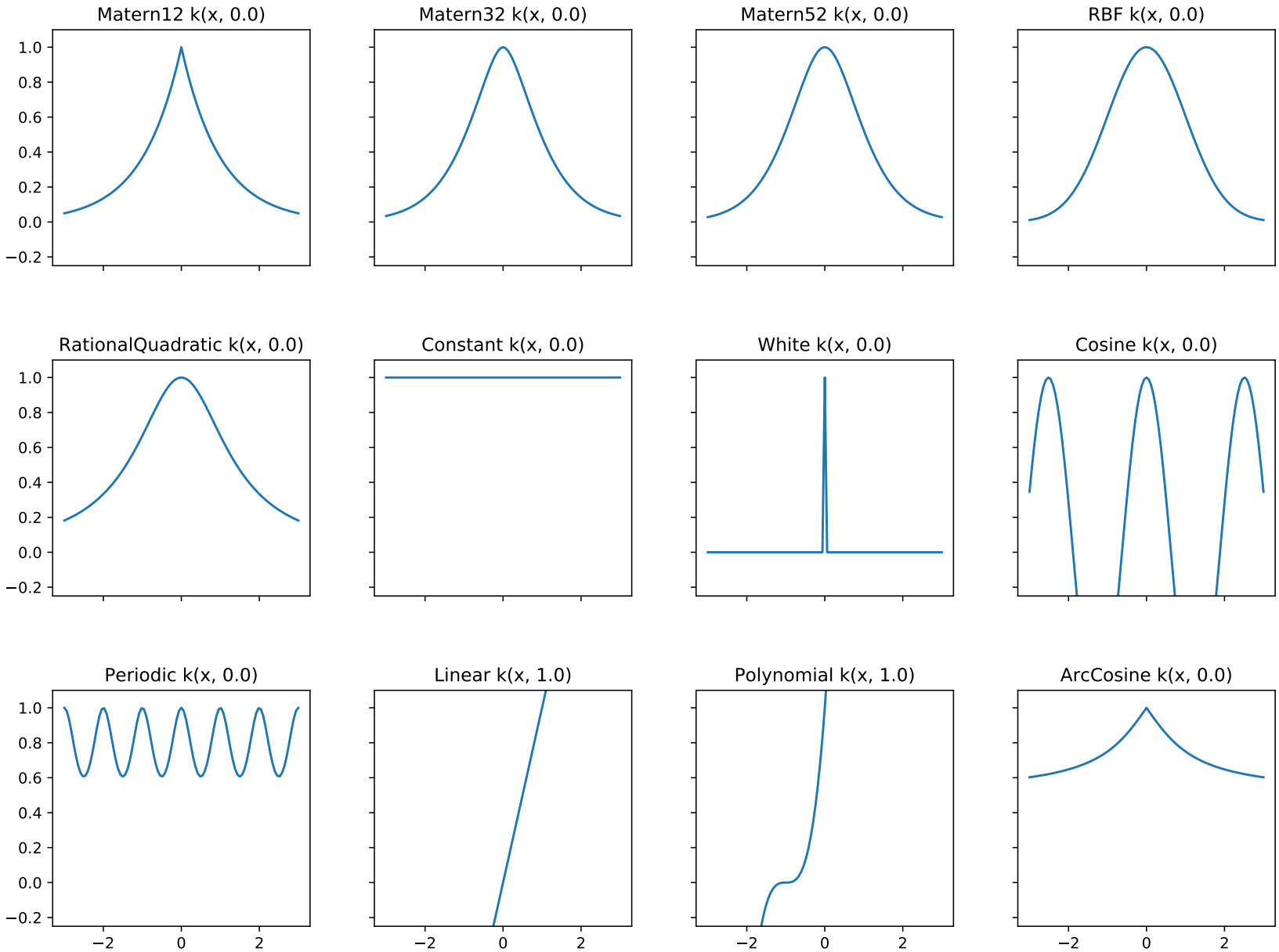
$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \geq 0$$

Proving that a function is psd is often difficult. However there are a lot of functions that have already been proven to be psd:

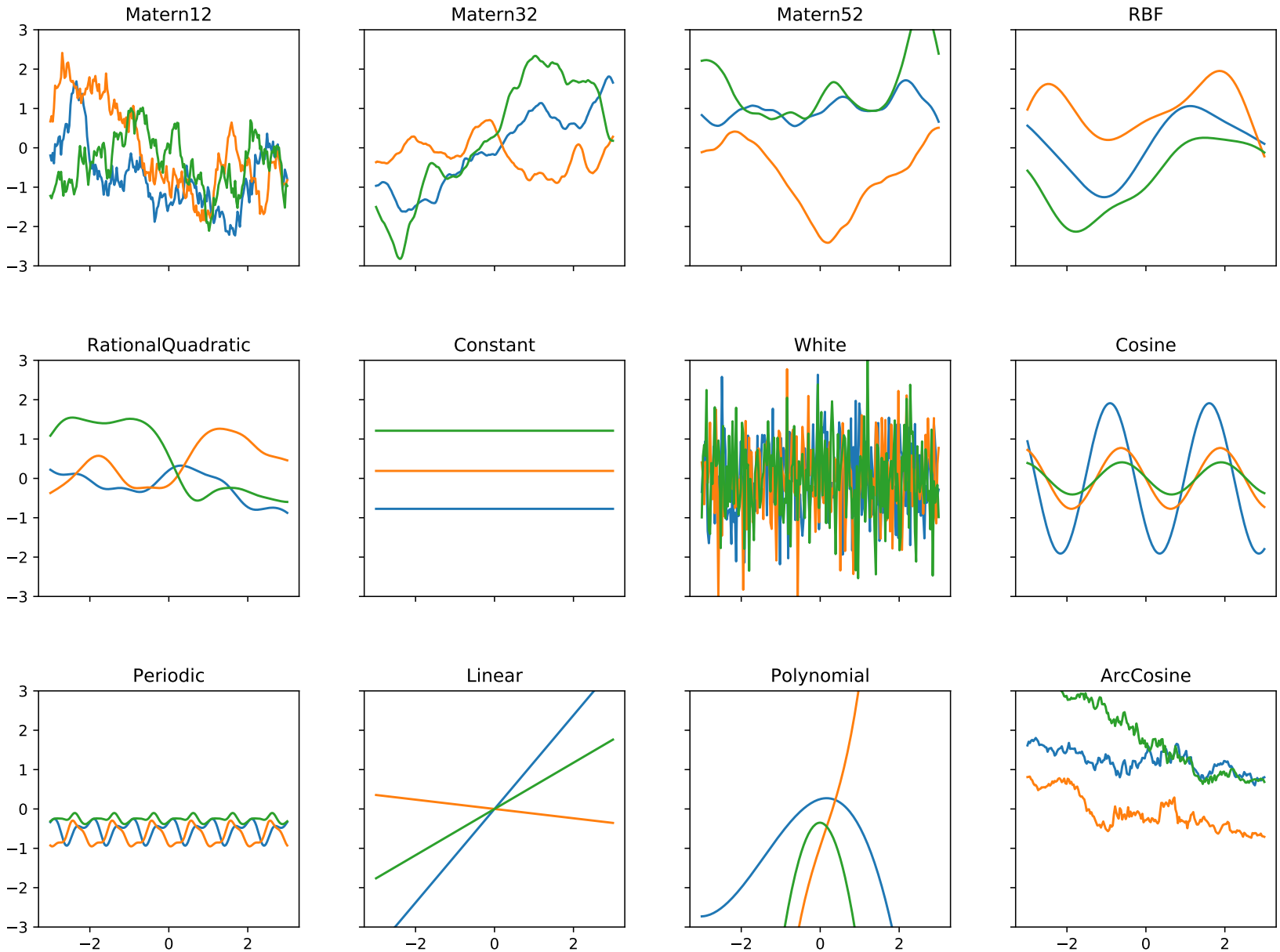
squared exp.	$k(x,y) = \sigma^2 \exp\left(-\frac{(x-y)^2}{2\theta^2}\right)$
Matern 5/2	$k(x,y) = \sigma^2 \left(1 + \frac{\sqrt{5}\lvert x-y\rvert}{\theta} + \frac{5\lvert x-y\rvert^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}\lvert x-y\rvert}{\theta}\right)$
Matern 3/2	$k(x,y) = \sigma^2 \left(1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

When k is a function of $x - y$, the kernel is called **stationary**. σ^2 is called the **variance** and θ the **lengthscale**.

Examples of kernels in gpflow:

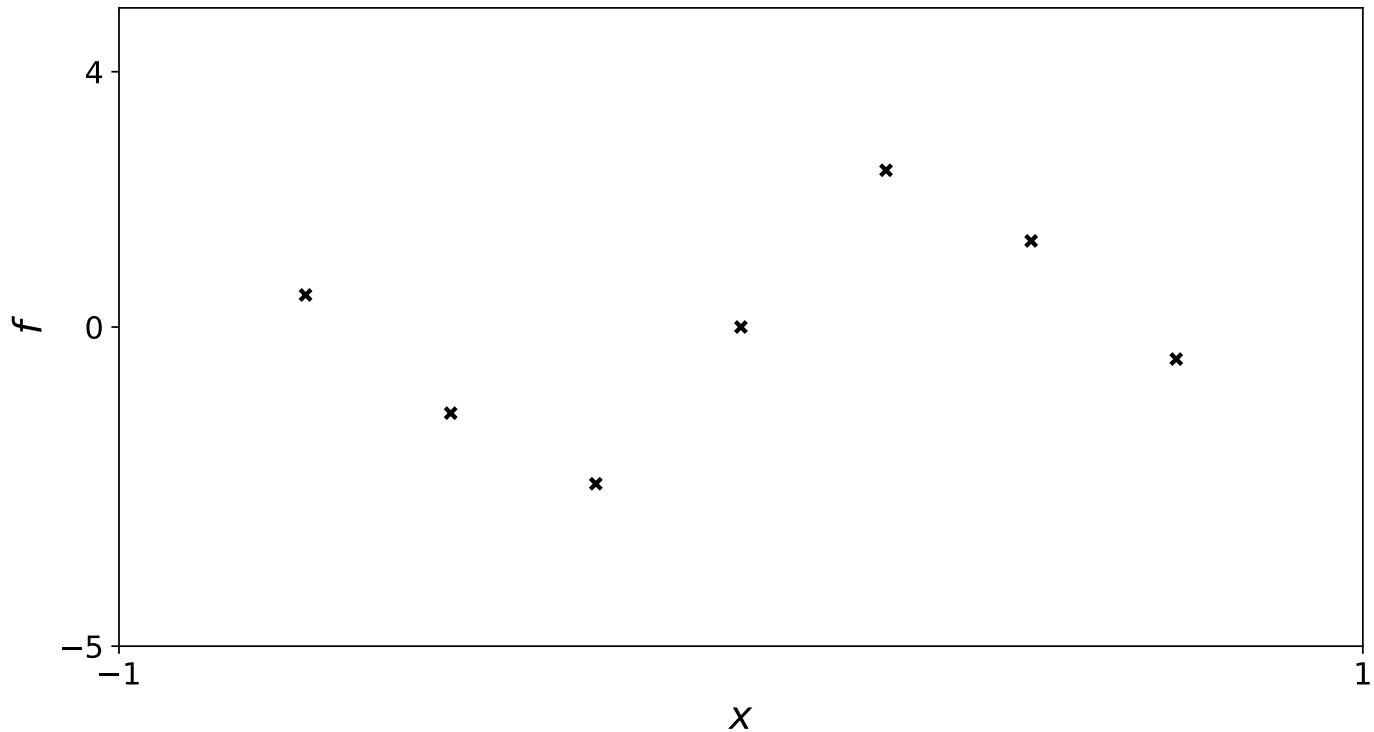


Associated samples



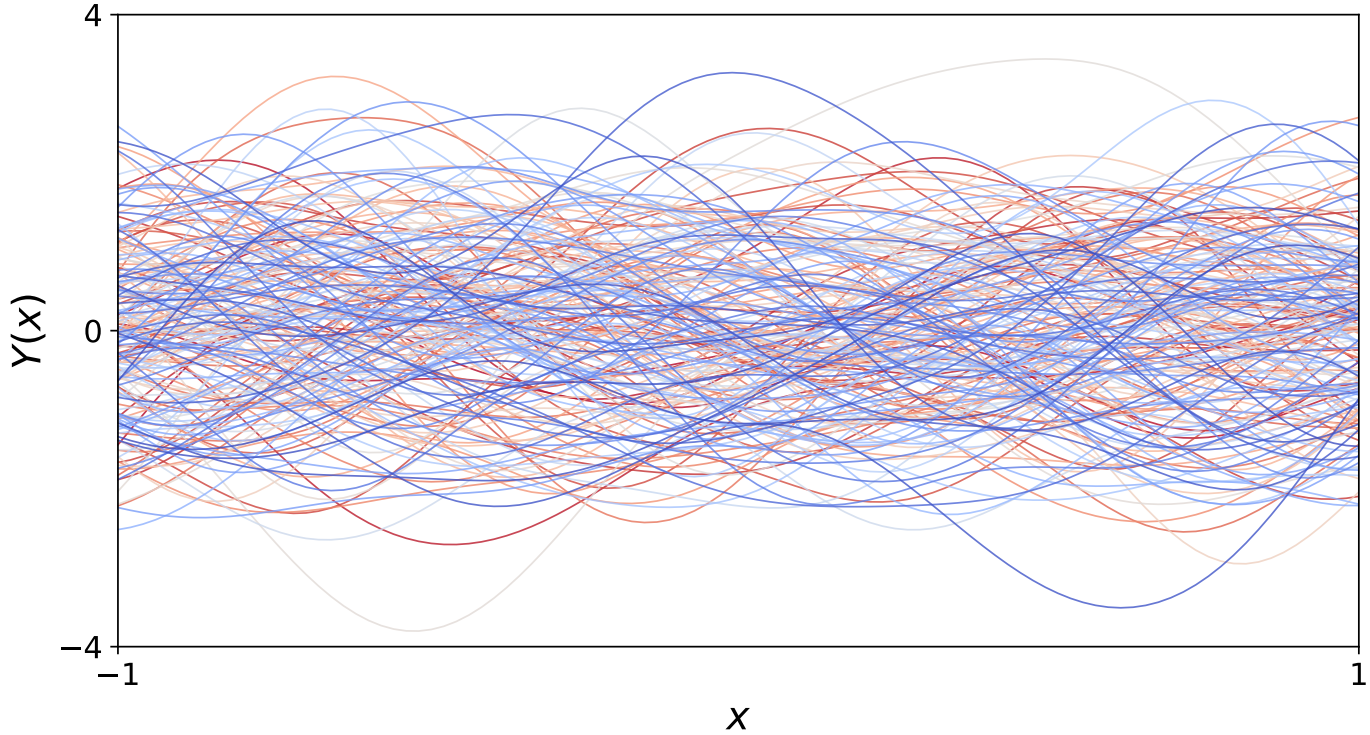
Gaussian process regression

We assume we have observed a function f for a set of points $X = (X_1, \ldots, X_n)$:



The vector of observations is $F = f(X)$ (ie $F_i = f(X_i)$).

Since f in unknown, we make the general assumption that it is the sample path of a Gaussian process $Z \sim \mathcal{N}(0, k)$:



The posterior distribution $Z(\cdot)|Z(X) = F$:

- Is still a Gaussian process
- Can be computed analytically

It is $\mathcal{N}(m(\cdot), c(\cdot,\cdot))$ with:

$$\begin{aligned} m(x) &= \mathrm{E}[Z(x)|Z(X)=F] \\ &= k(x,X)k(X,X)^{-1}F \end{aligned}$$

$$\begin{aligned} c(x,y) &= \mathrm{cov}[Z(x),Z(y)|Z(X)=F] \\ &= k(x,y) - k(x,X)k(X,X)^{-1}k(X,y) \end{aligned}$$

A few words on GPR **Complexity**

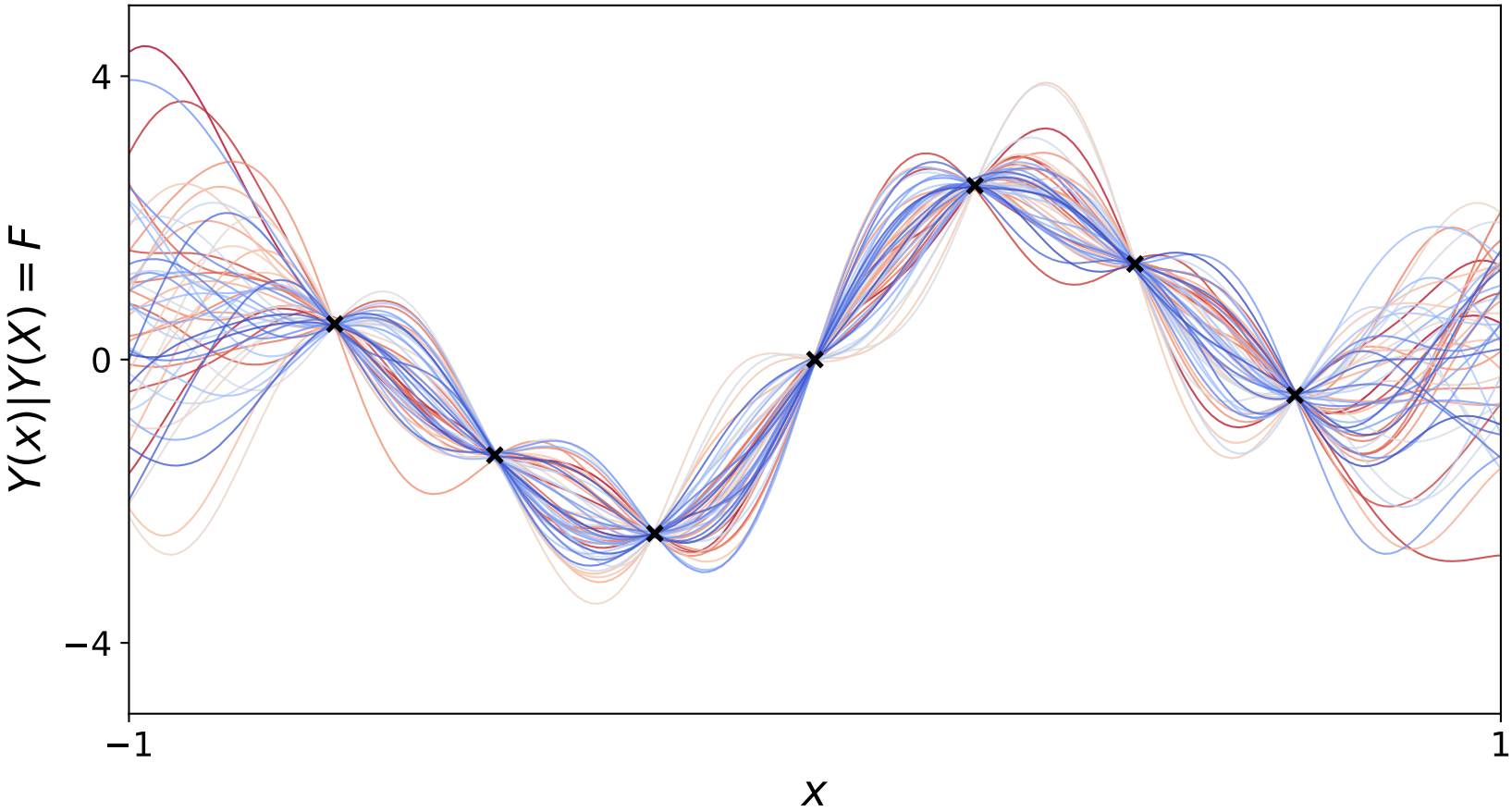
- Storage footprint is $\mathcal{O}(n^2)$: We have to store the covariance matrix which is $n \times n$.
- Complexity is $\mathcal{O}(n^3)$: We have to invert the covariance matrix (or compute the Cholesky factor and apply triangular solves).

Storage footprint is often the first limit to be reached.

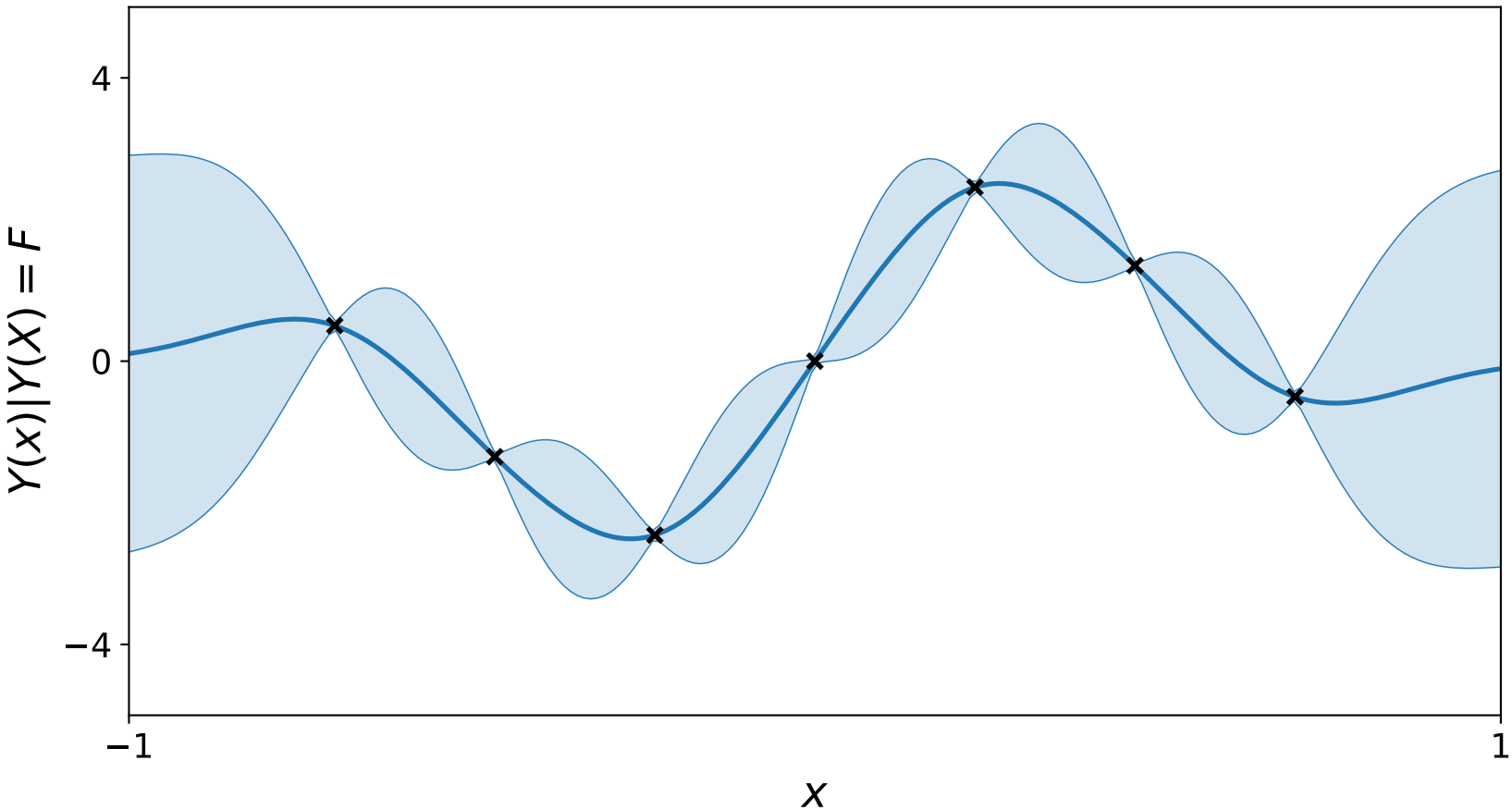
The maximal number of observation points is between 1000 and 10 000.

What if we have more data? $\Rightarrow$ Talk from Zhenwen tomorrow

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:

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

$\Rightarrow$ Demo https://durrande.shinyapps.io/gp_playground

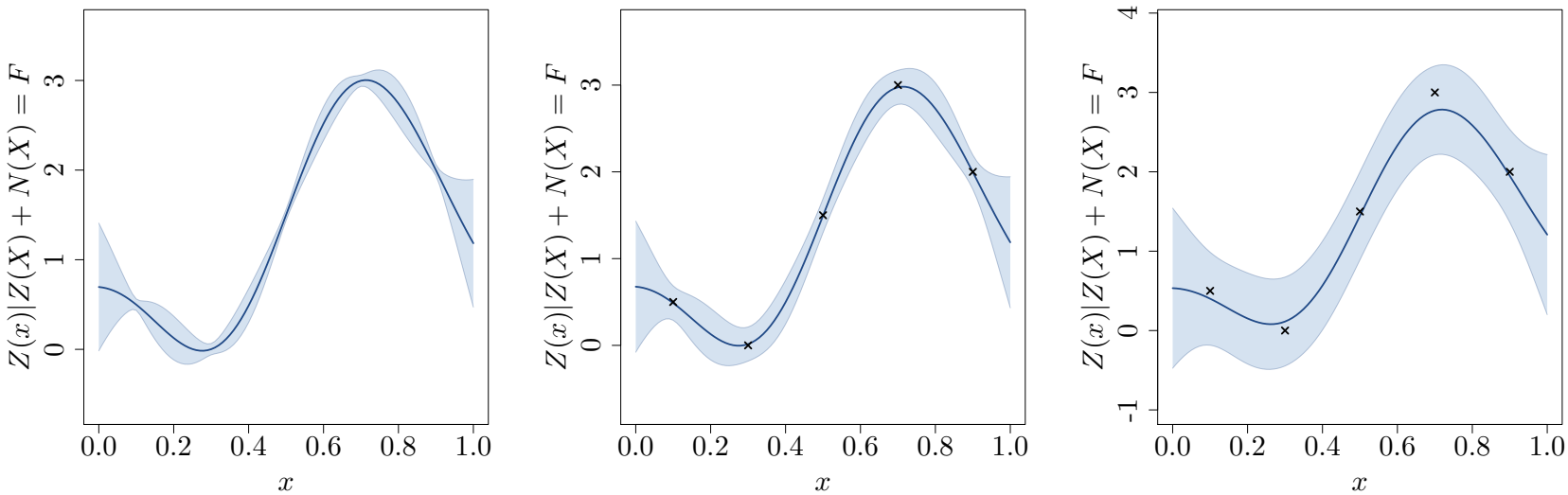
We are not always interested in models that interpolate the data. For example, if there is some observation noise: $F = f(X) + \varepsilon$.

Let N be a process $\mathcal{N}(0, n(.,.))$ that represent the observation noise. The expressions of GPR with noise are

$$\begin{aligned} m(x) &= \mathrm{E}[Z(x)|Z(X) + N(X)=F] \\ &= k(x,X)(k(X,X) + n(X,X))^{-1}F \end{aligned}$$

$$\begin{aligned} c(x,y) &= \mathrm{cov}[Z(x), Z(y)|Z(X) + N(X)=F] \\ &= k(x,y) - k(x,X)(k(X,X) + n(X,X))^{-1}k(X,y) \end{aligned}$$

Examples of models with observation noise for $n(x, y) = \tau^2 \delta_{x,y}$:



The values of τ^2 are respectively 0.001, 0.01 and 0.1.

What if $F = f(X) + \varepsilon$ isn't appropriate? $\Rightarrow$ Talks from Carl Henrik and Neil (days 2 and 3).

Parameter estimation

The choice of the kernel parameters has a great influence on the model. $\Rightarrow$ 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 f_X given some observations $X_1, \dots, X_p$ is:

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

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

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

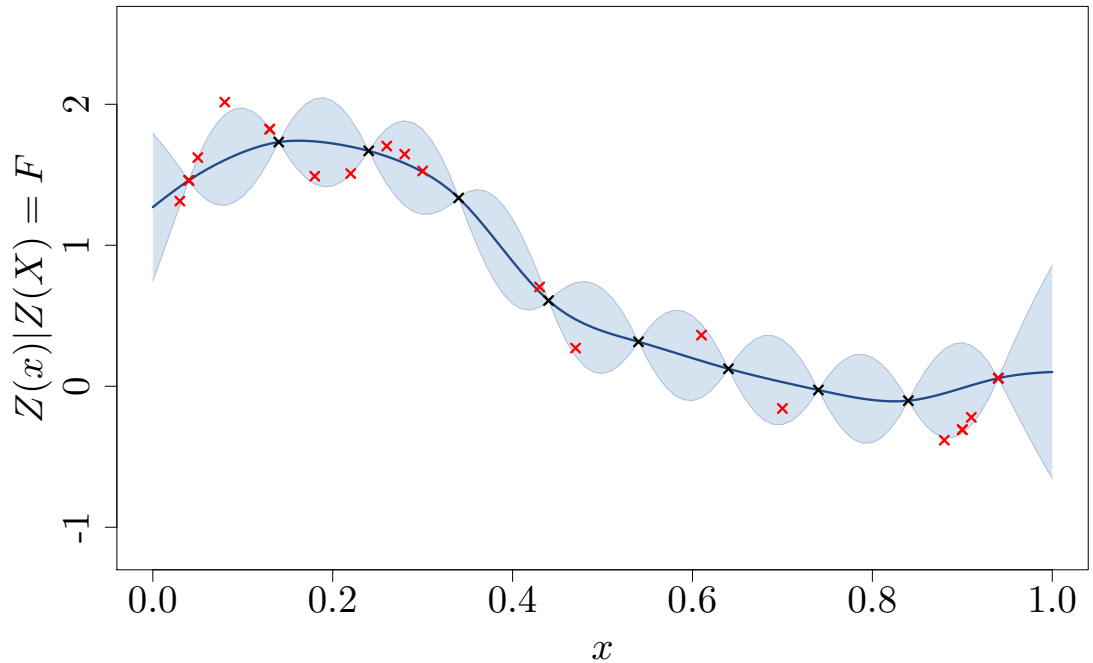
It is thus possible to maximise L – or $\log(L)$ – with respect to the kernel's parameters in order to find a well suited prior.

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

$$f_{Z(X)}(F) = f(F_1) \times f(F_2|F_1) \times f(F_3|F_1, F_2) \times \cdots \times f(F_n|F_1, \ldots, F_{n-1})$$

Model validation

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?
- Do the confidence intervals make sense?
- Are the predicted covariances right?

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:

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

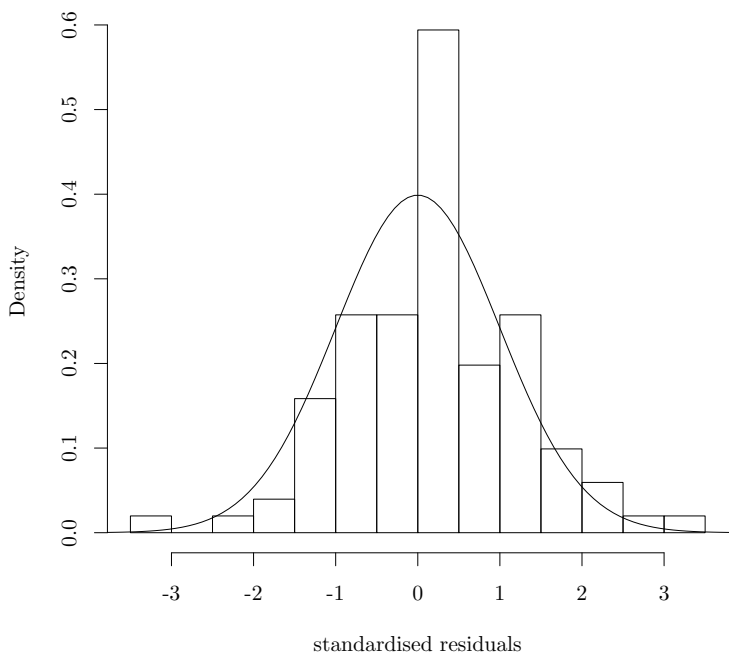
$$\text{A "normalised" criterion} \qquad Q_2 = 1 - \frac{\sum(F_t - m(X_t))^2}{\sum(F_t - \text{mean}(F_t))^2}$$

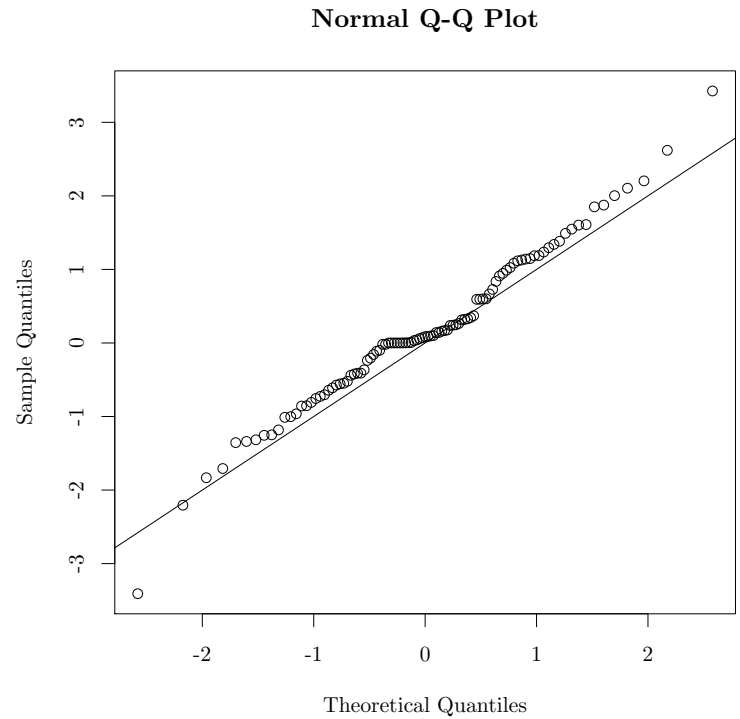
On the above example we get $MSE = 0.038$ and $Q_2 = 0.95$.

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





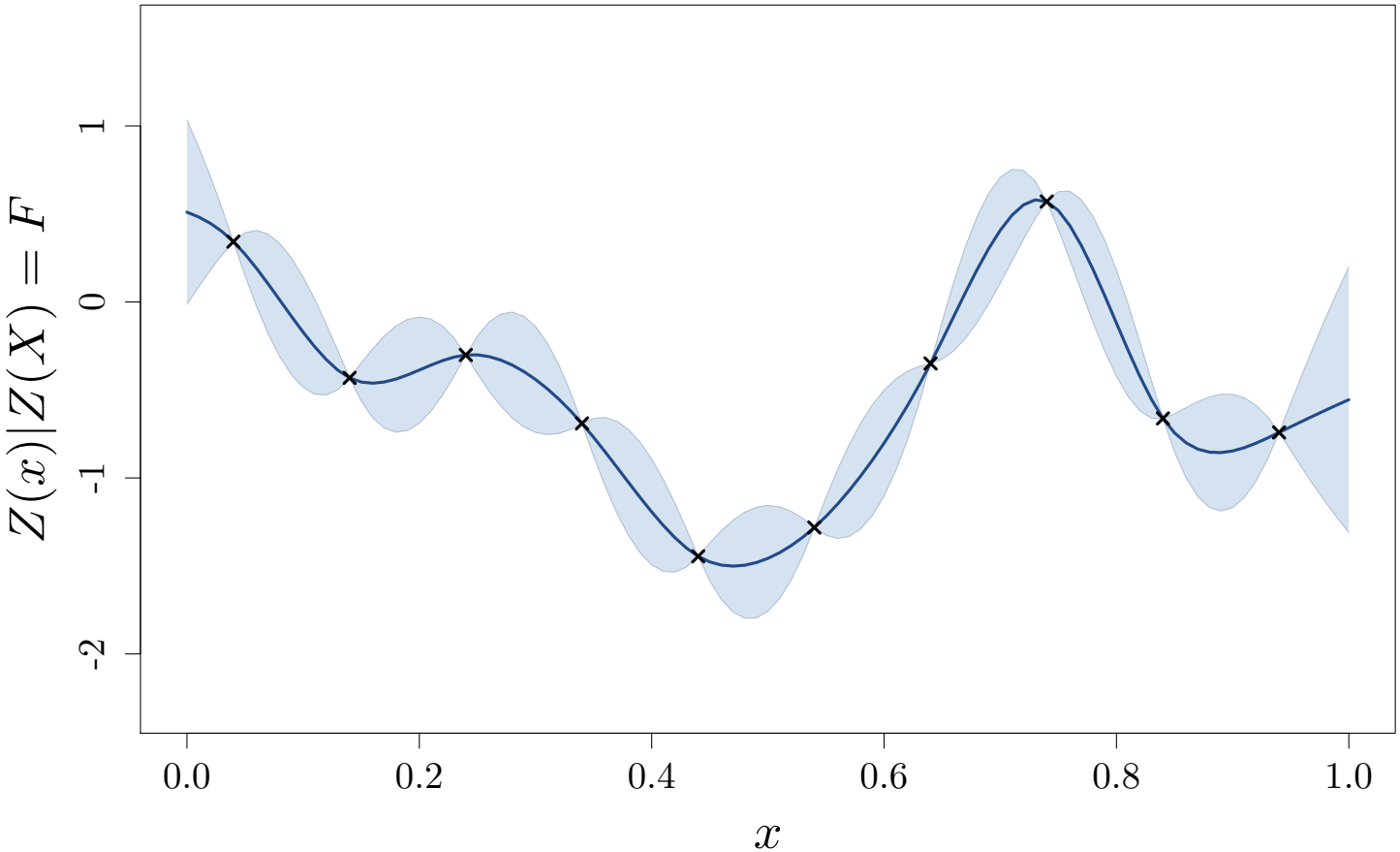
When no test set is available, another option is to consider cross validation methods such as leave-one-out.

The steps are:

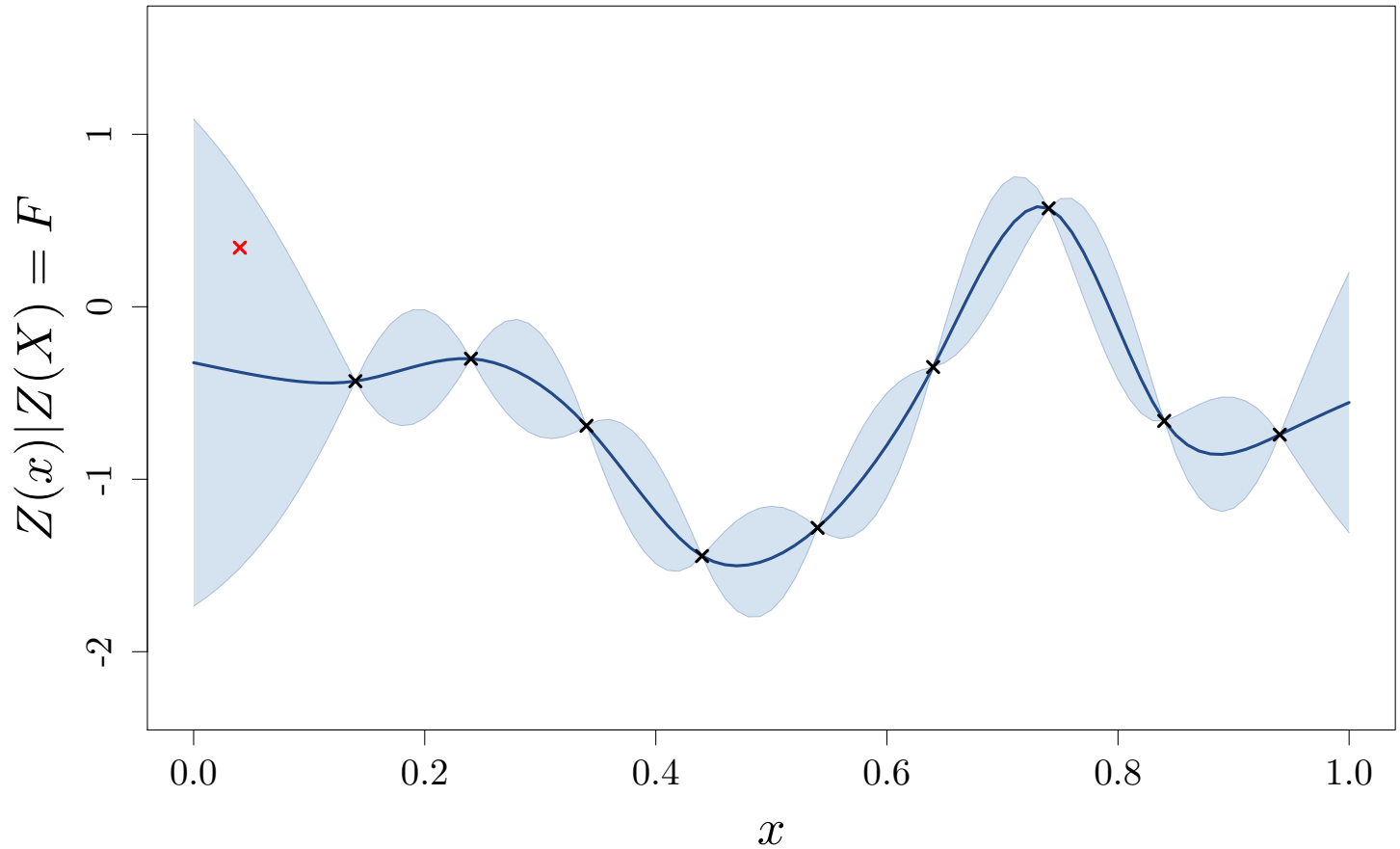
1. build a model based on all observations except one
2. compute the model error at this point

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

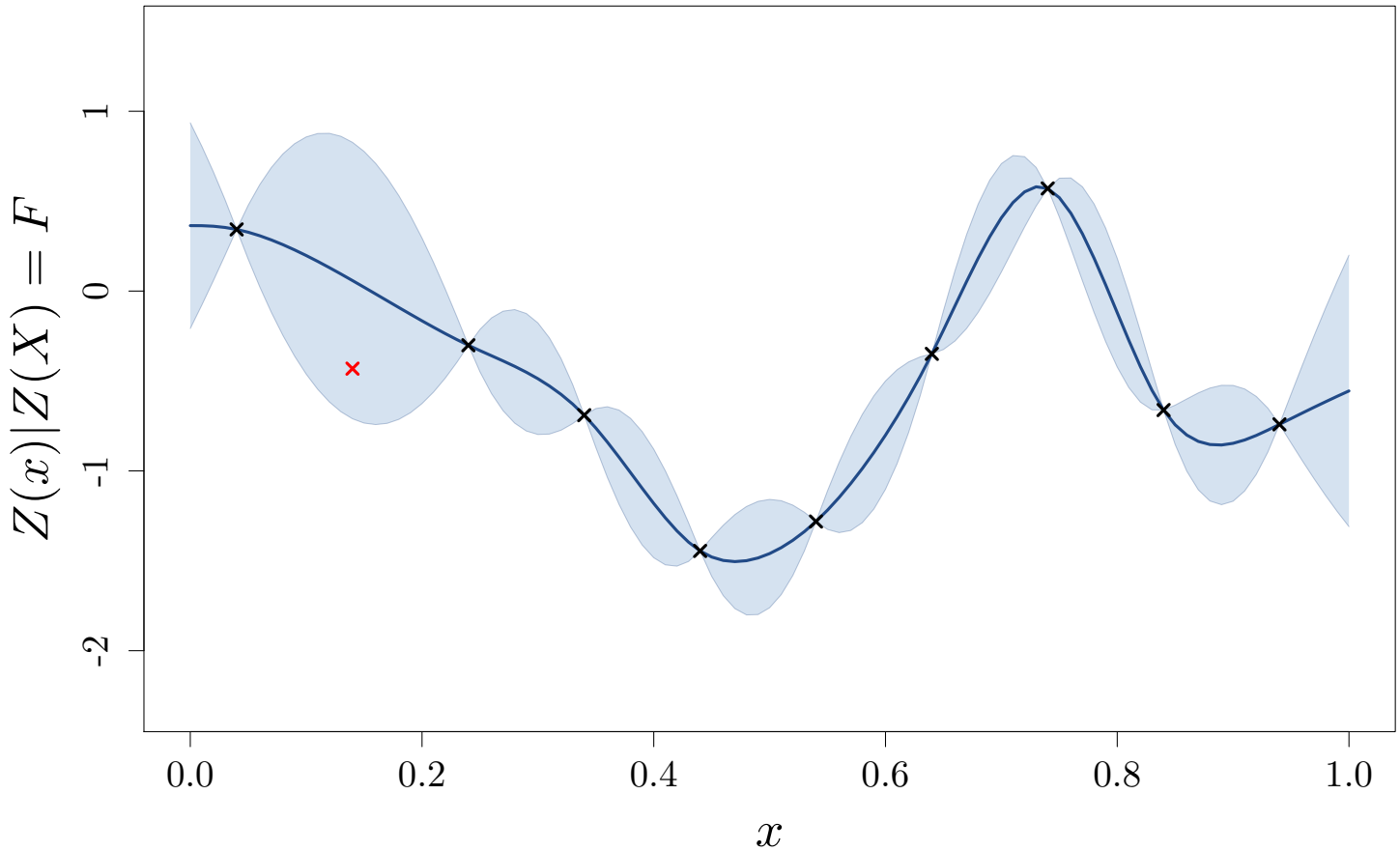
Model to be tested:



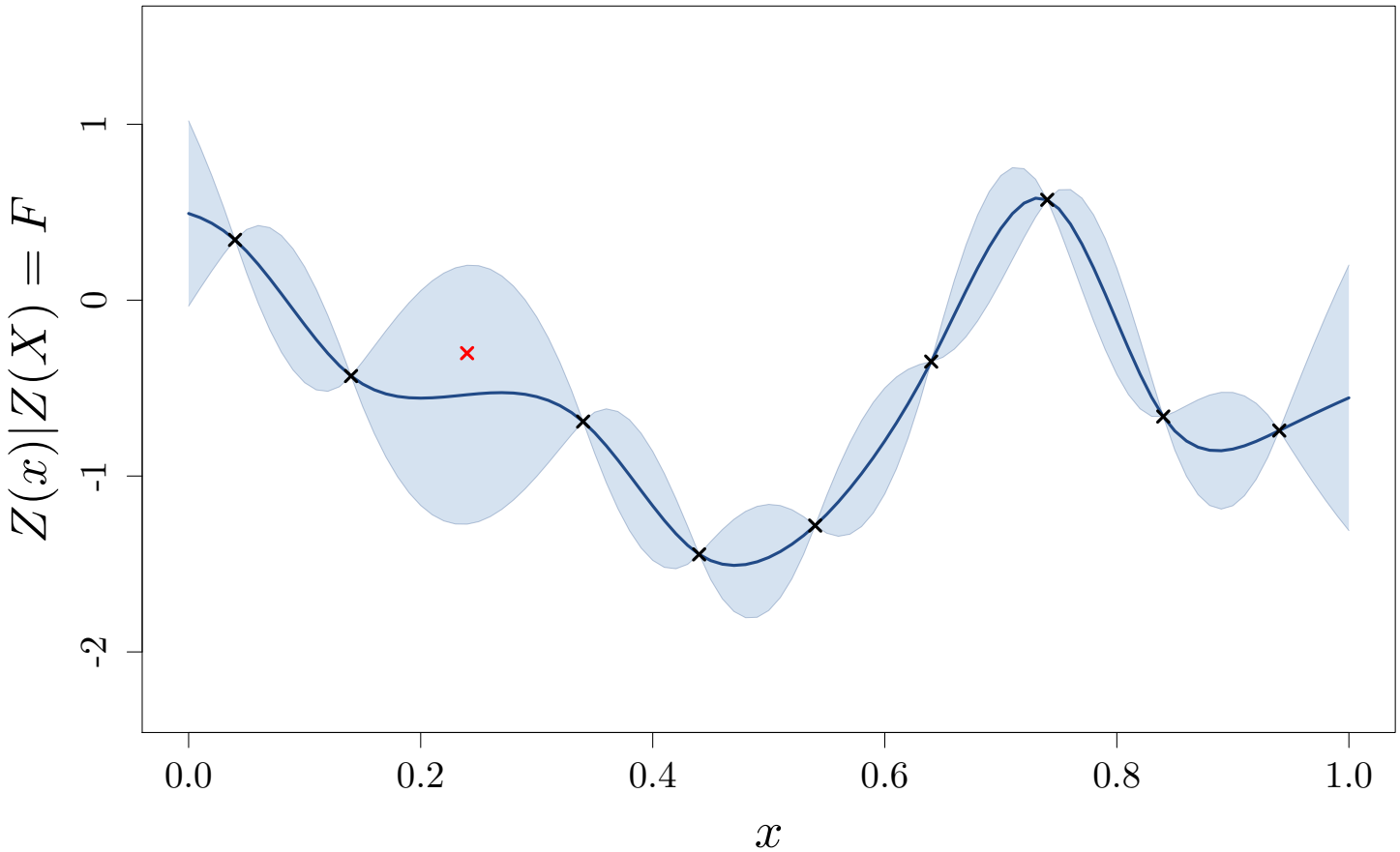
Step 1:



Step 2:



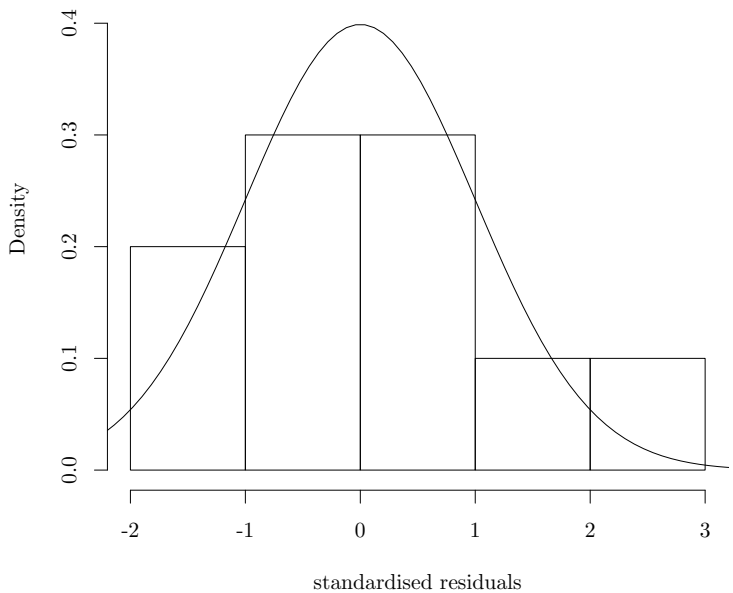
Step 3:

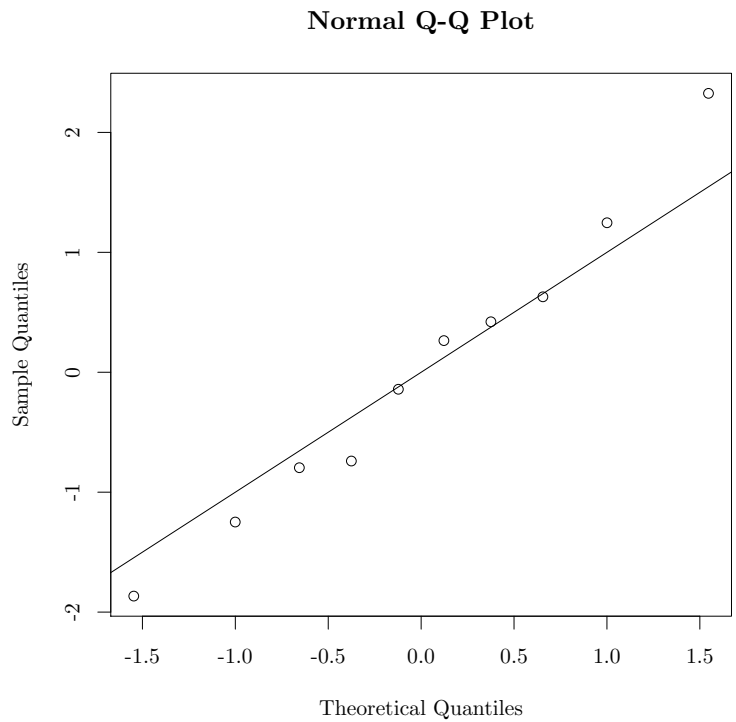


We finally obtain:

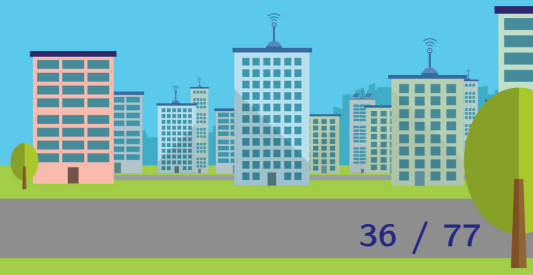
$$MSE = 0.24 \text{ and } Q_2 = 0.34.$$

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



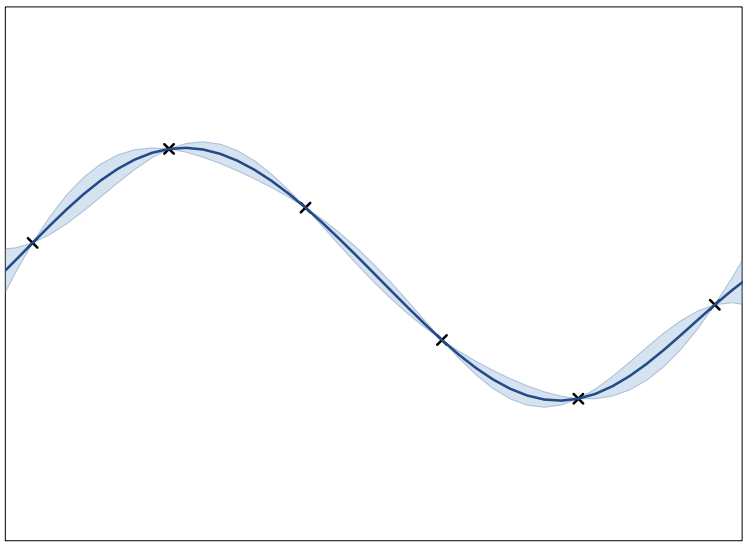


Choosing the kernel

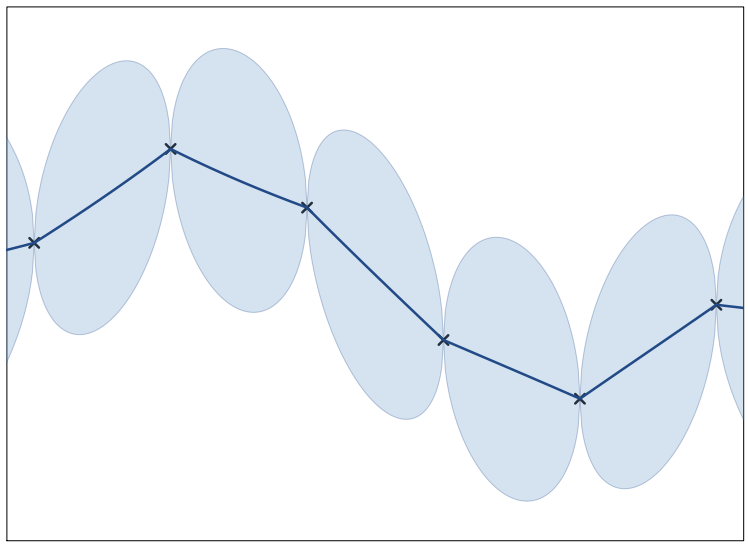


Changing the kernel has a huge impact on the model:

Gaussian kernel:

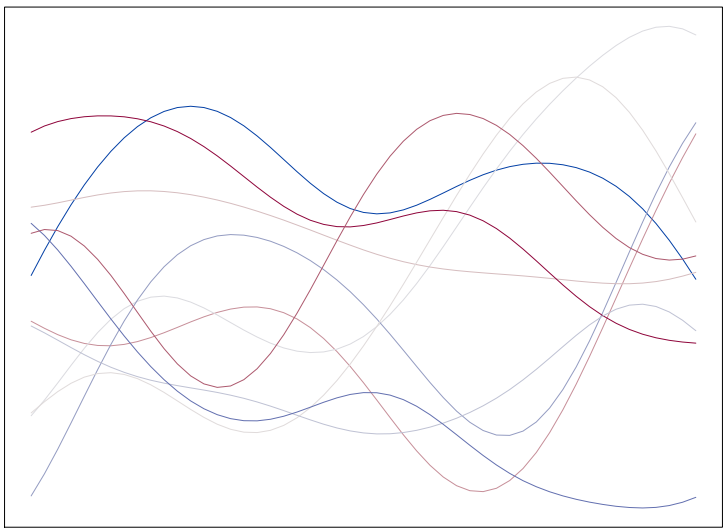


Exponential kernel:

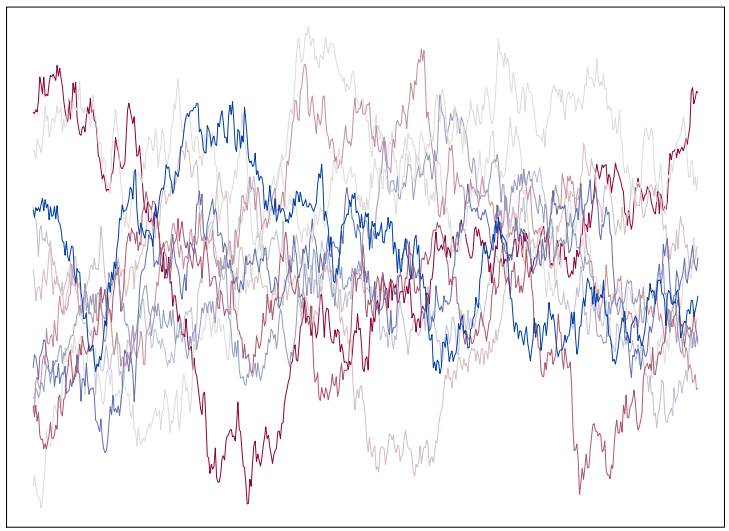


This is because changing the kernel means changing the prior on f

Gaussian kernel:



Exponential kernel:



Kernels encode the prior belief about the function to approximate...
they should be chosen accordingly!

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:

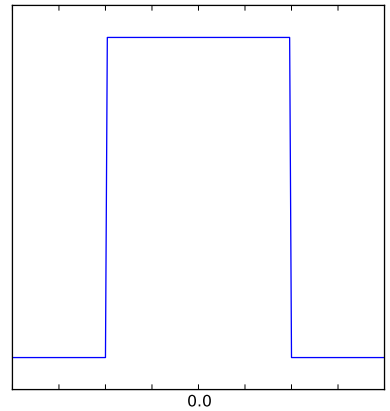
Theorem (Bochner)

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:

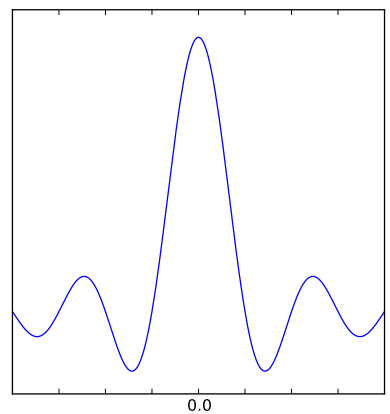
$$\tilde{k}(t) = \int_{\mathbb{R}} e^{-i\omega t} \mathrm{d}\mu(\omega)$$

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.

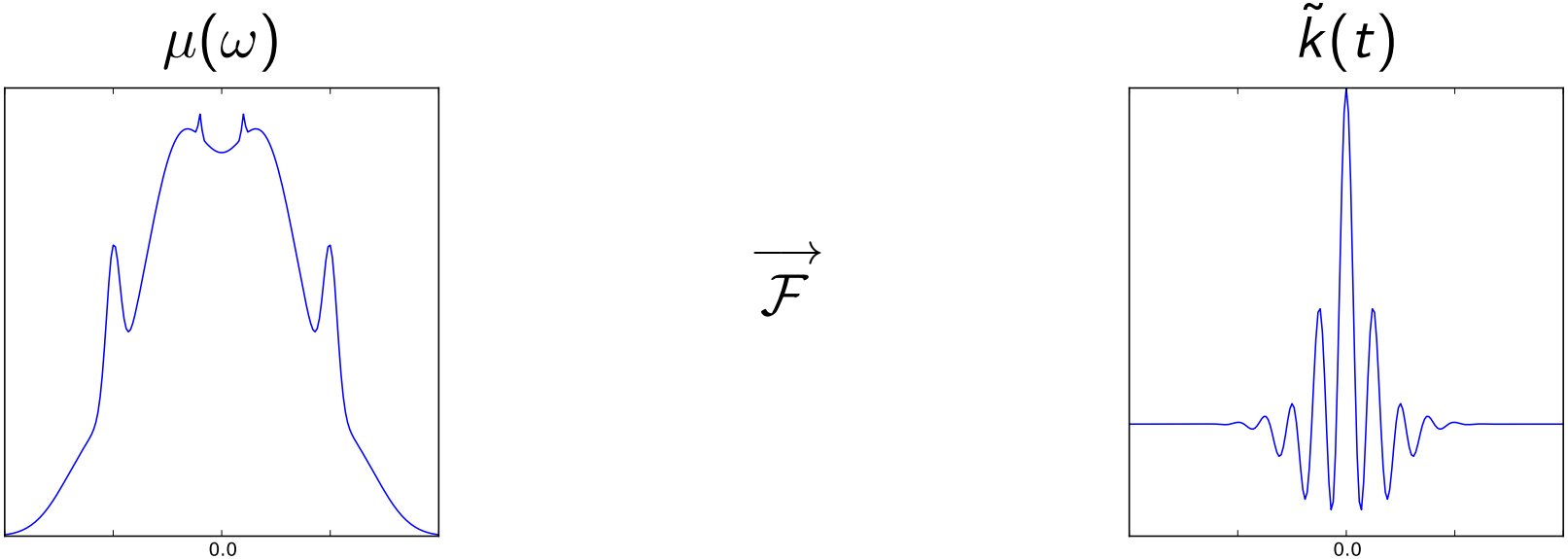
Usual kernels

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

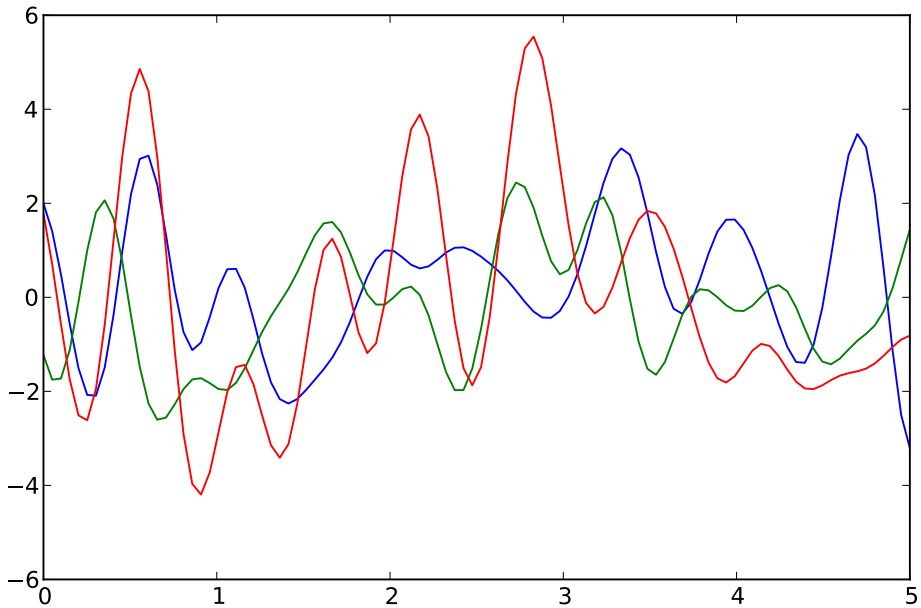
Inverse Fourier transform of a (symmetrised) sum of Gaussian gives (A. Wilson, ICML 2013):



The obtained kernel is parametrised by its spectrum.

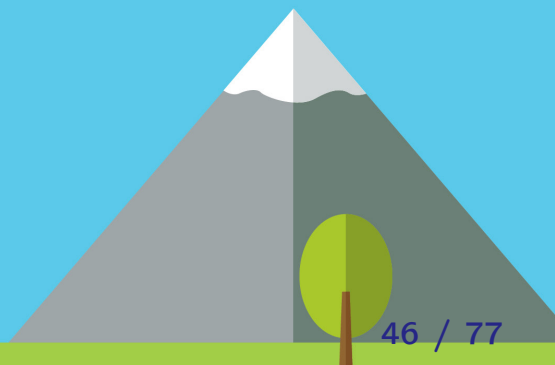
Unusual kernels

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

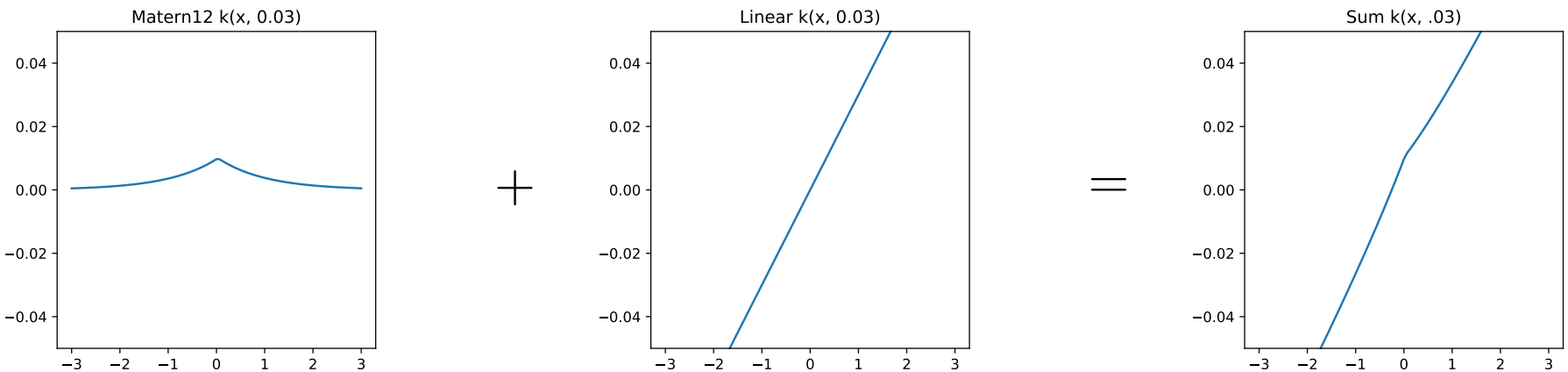
Property

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

is a valid covariance structure.

This can be proved directly from the p.s.d. definition.

Example

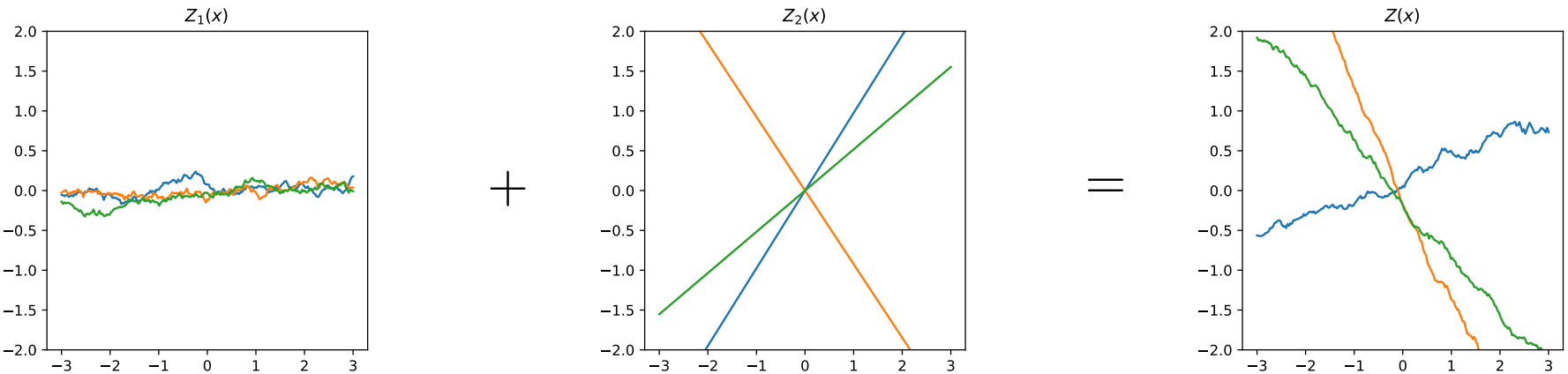


Sum of kernels over the same input space

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

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

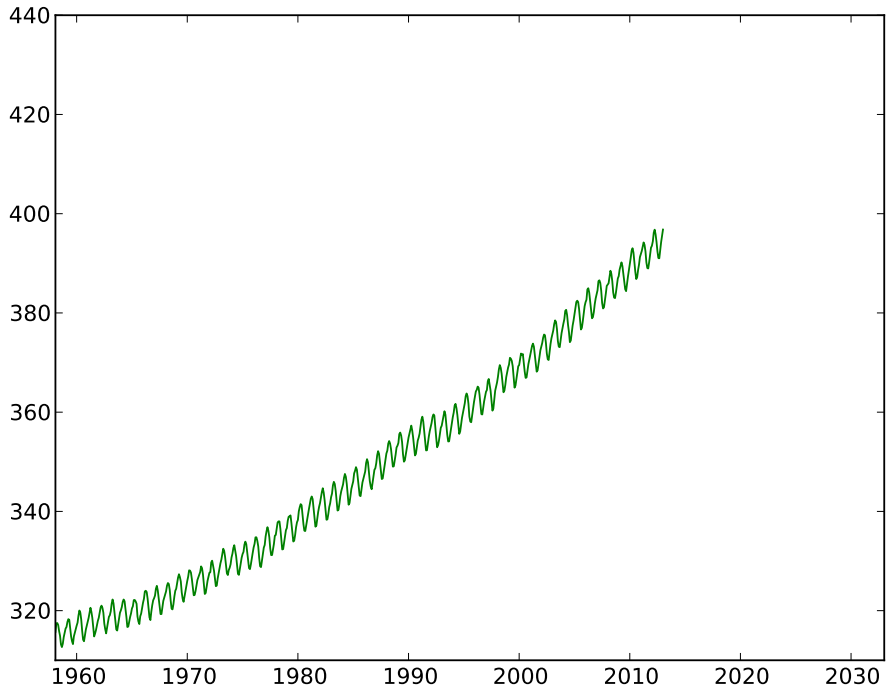
Example



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.

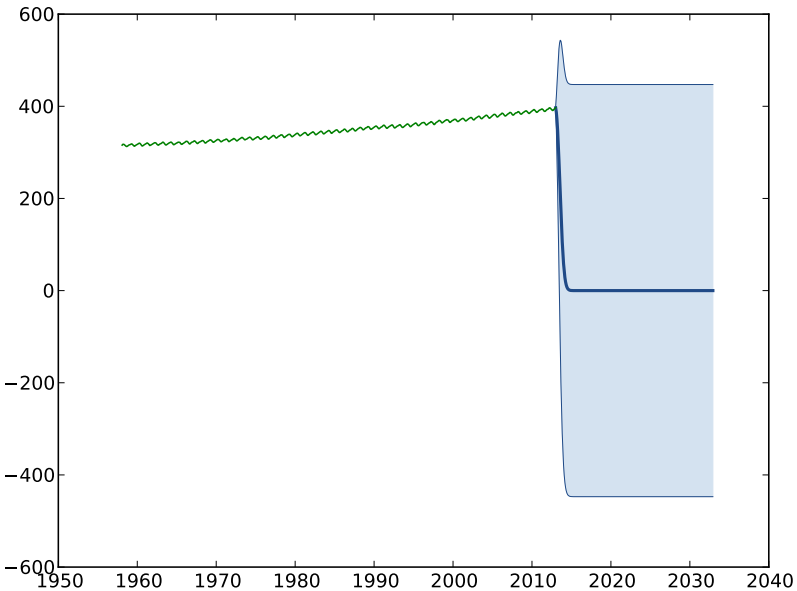


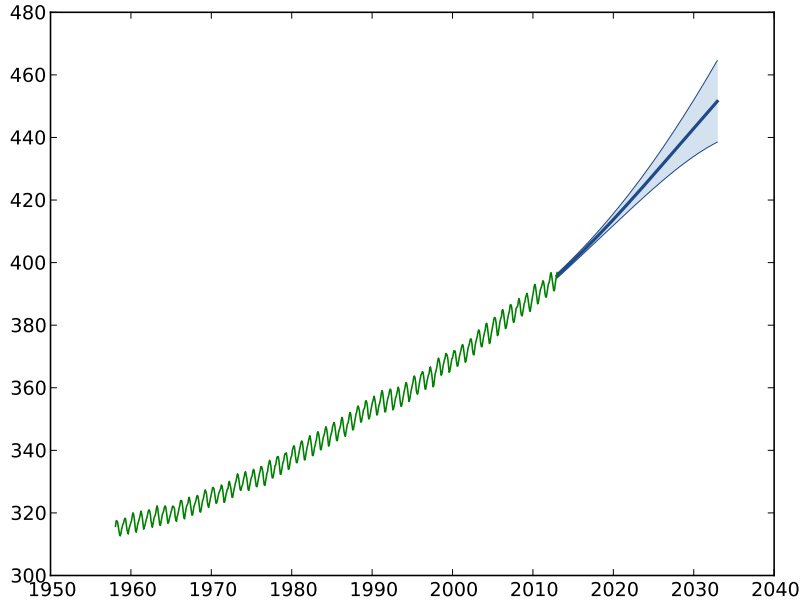
Let's try to predict the concentration for the next 20 years.

Sum of kernels over the same space

We first consider a squared-exponential kernel:

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





The results are terrible!

Sum of kernels over the same space

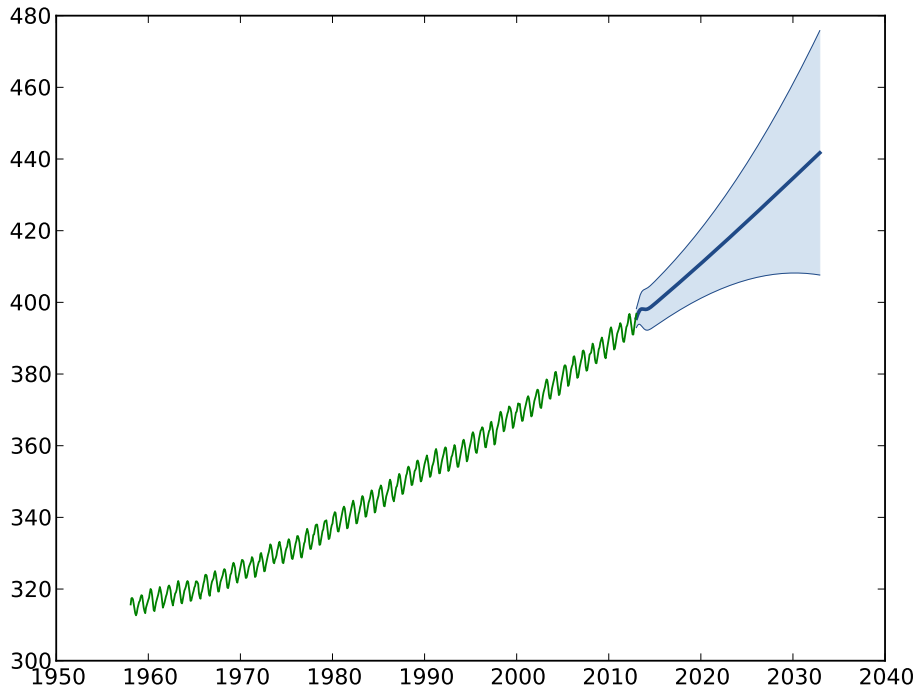
What happen if we sum both kernels?

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

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

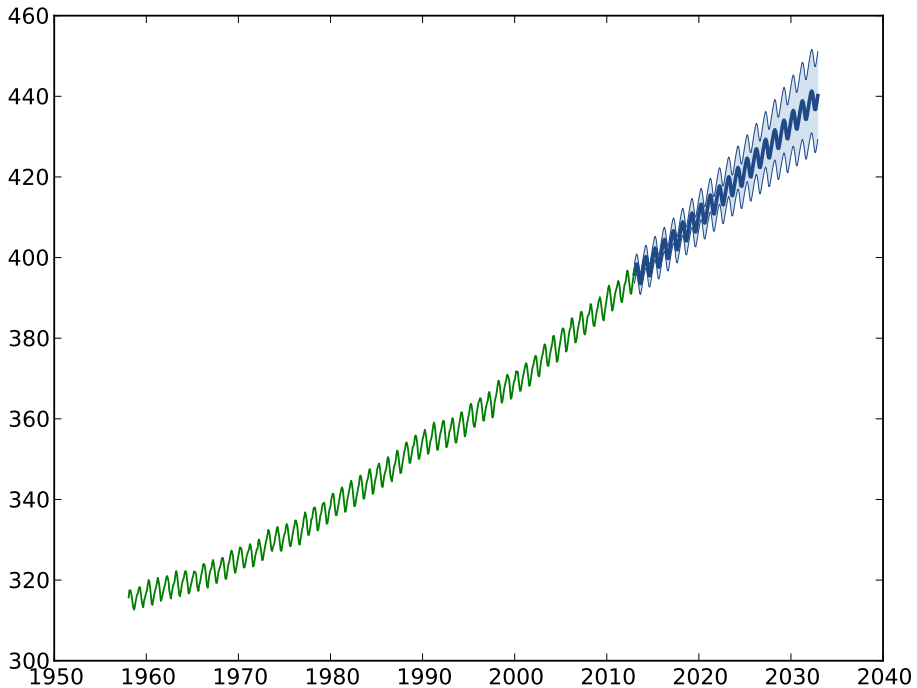
We can try the following kernel:

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

Sum of kernels over the same space

We can try the following kernel:

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



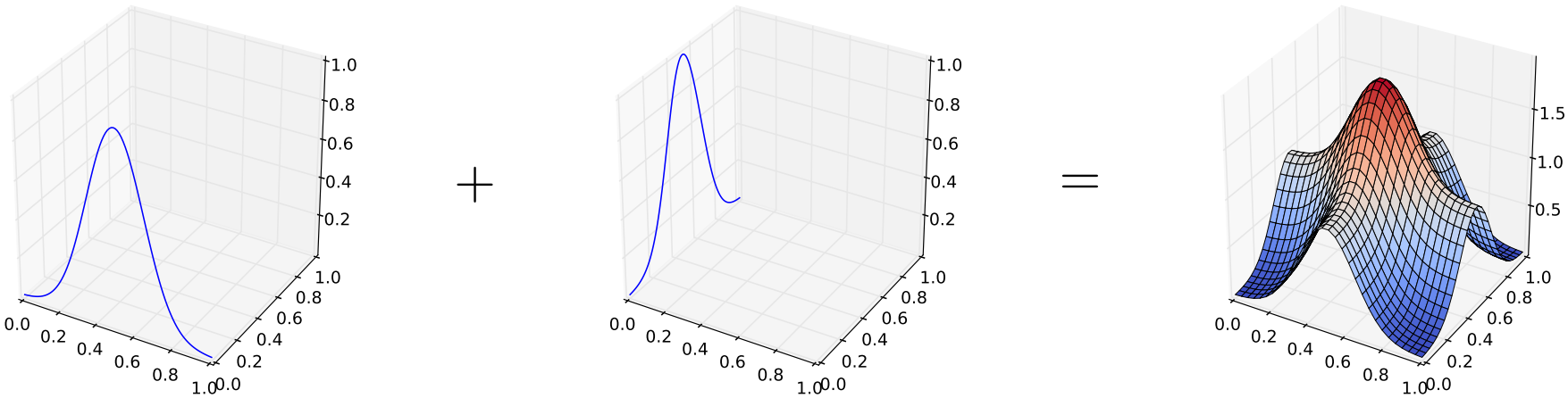
Once again, the model is significantly improved.

Sum of kernels over tensor space

Property

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

is a valid covariance structure.

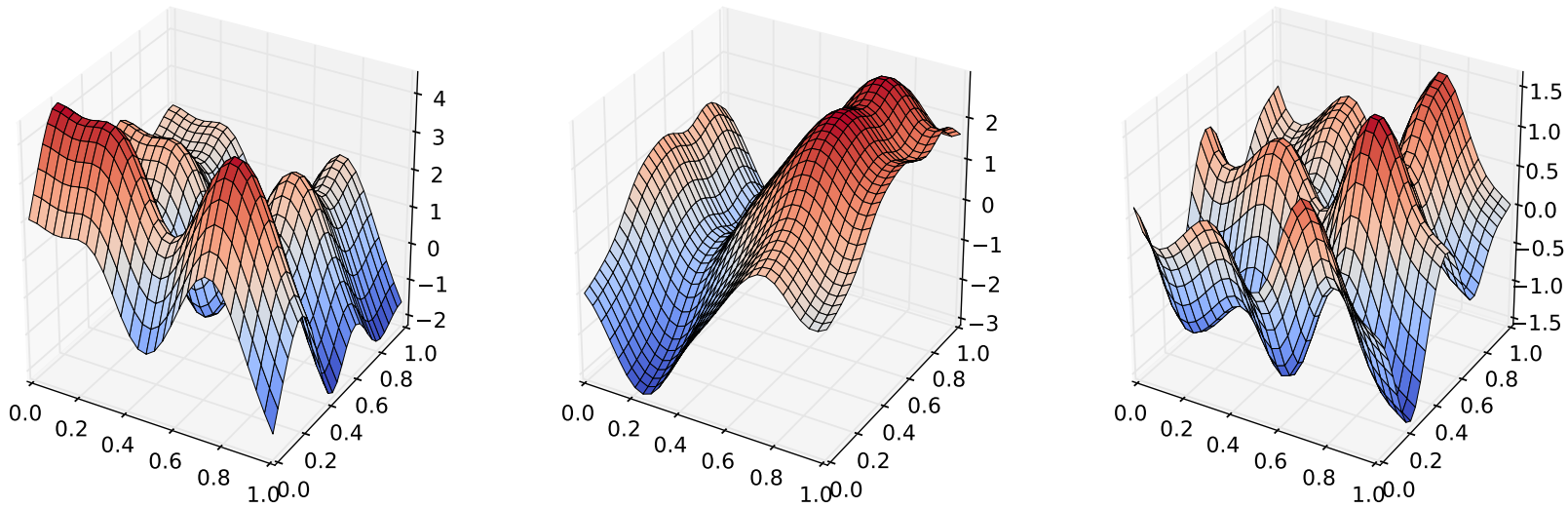


Remark:

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

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:

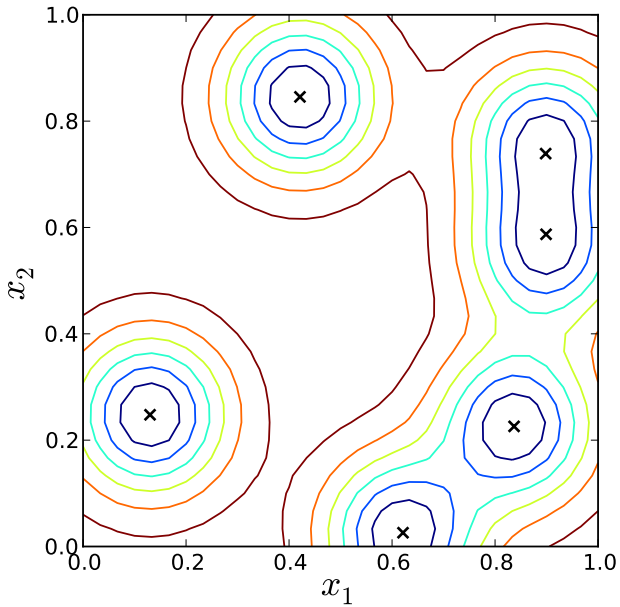
$$
m_1(x_1) = \mathrm{E}\left(Z_1(x_1) \mid Z_1(X_1) + Z_2(X_2) = F \right)
$$

Sum of kernels over tensor space

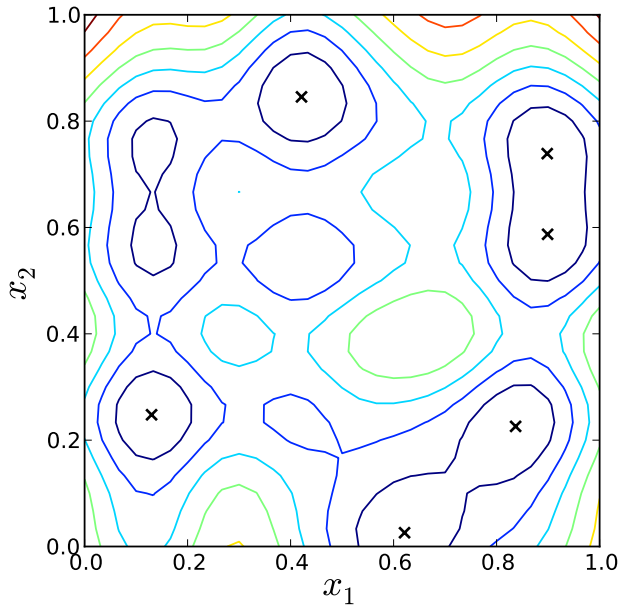
Remark

- The prediction variance has interesting features

pred. var. with kernel product

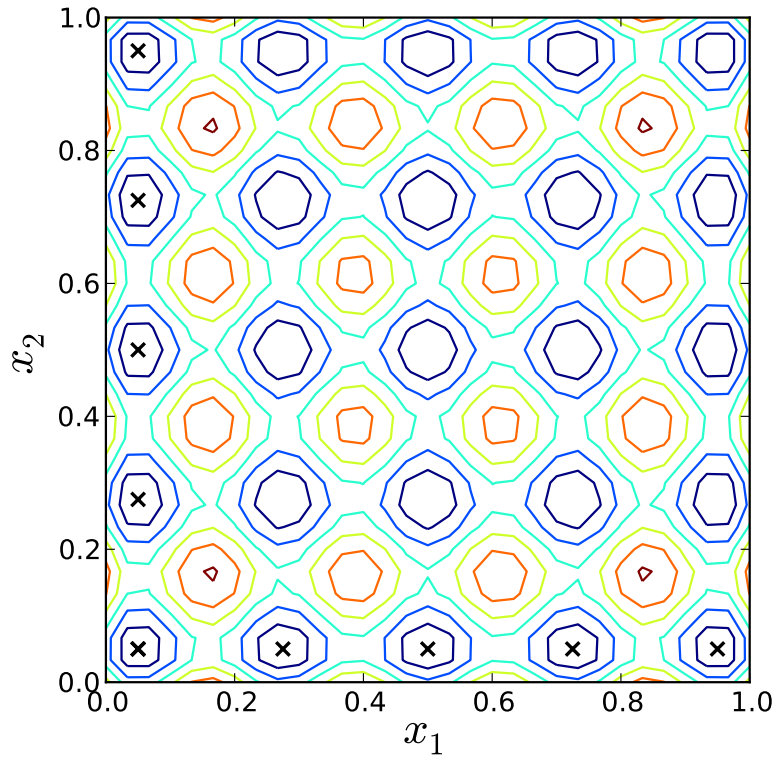


pred. var. with kernel sum



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

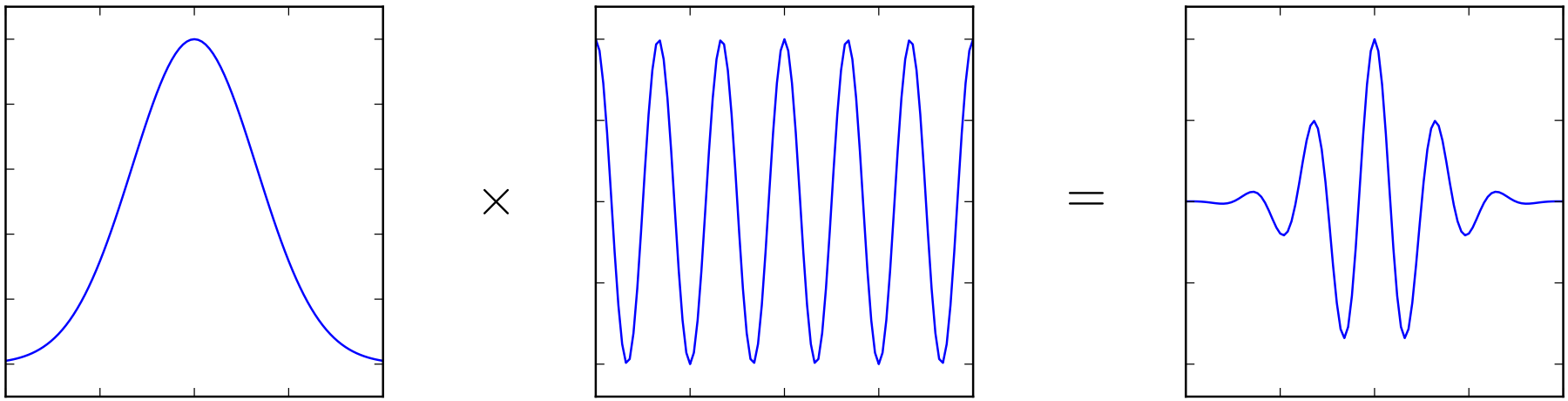
Property

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

is valid covariance structure.

Example

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



Product over the tensor space

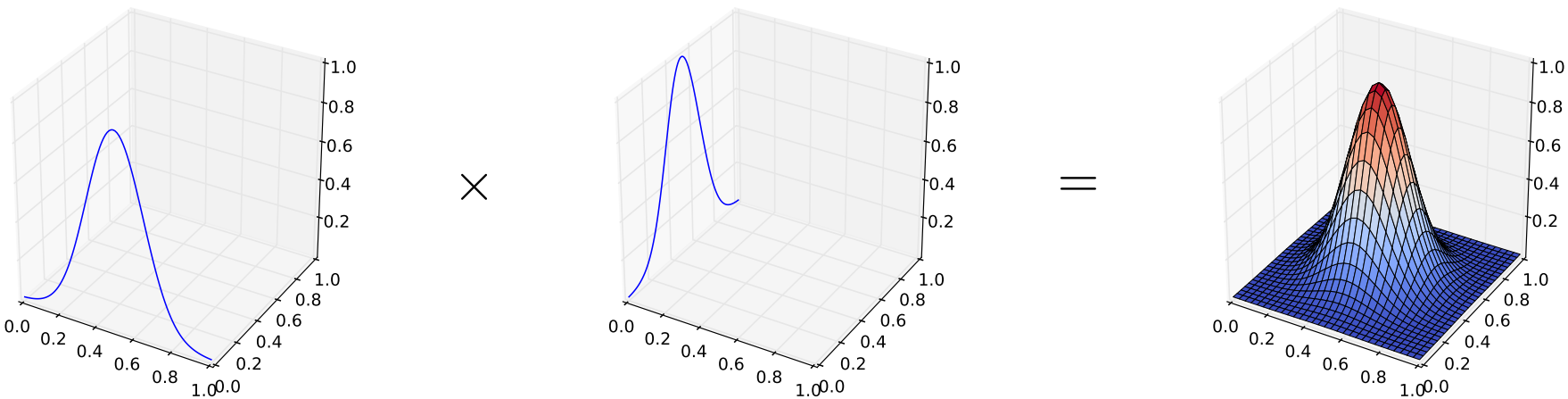
Property

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

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:

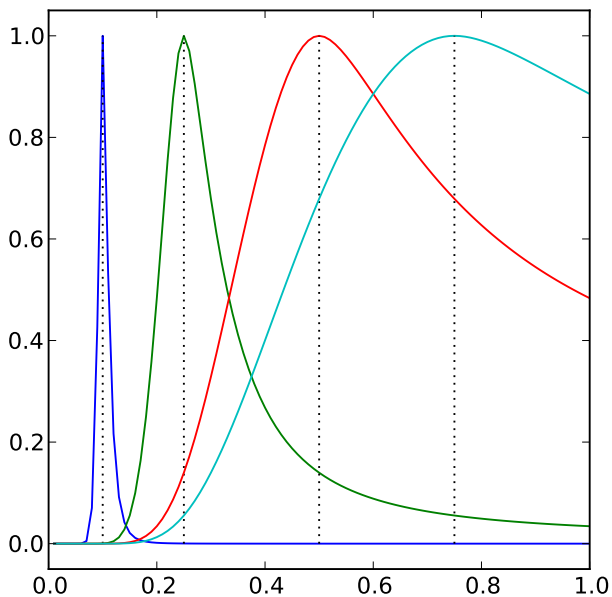
- k corresponds to the covariance of $Z(x) = Z_1(f(x))$
- This can be seen as a (nonlinear) rescaling of the input space

Example

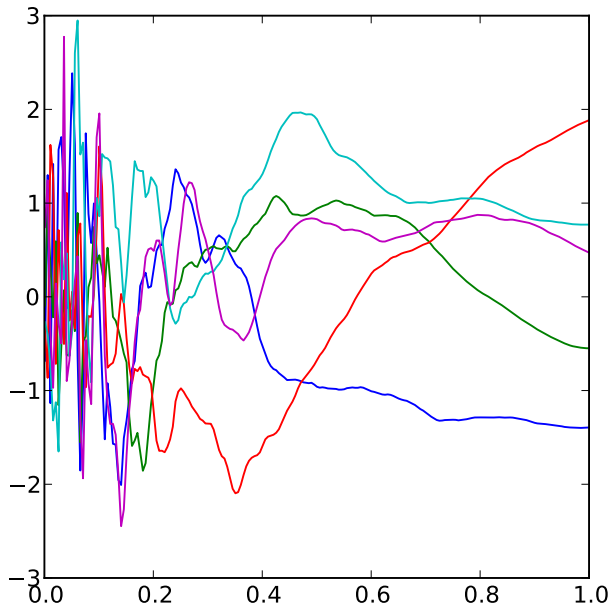
We consider $f(x) = \frac{1}{x}$ and a Matérn 3/2 kernel $k_1(x, y) = (1 + |x - y|)e^{-|x-y|}$.

We obtain:

Kernel



Sample paths



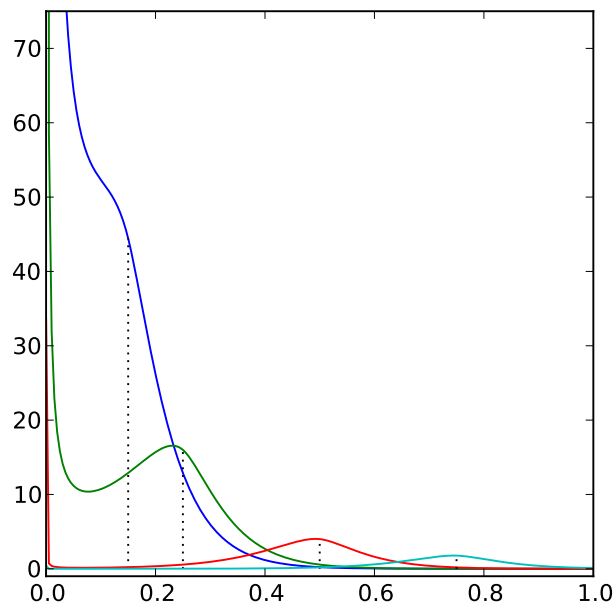
All these transformations can be combined!

Example

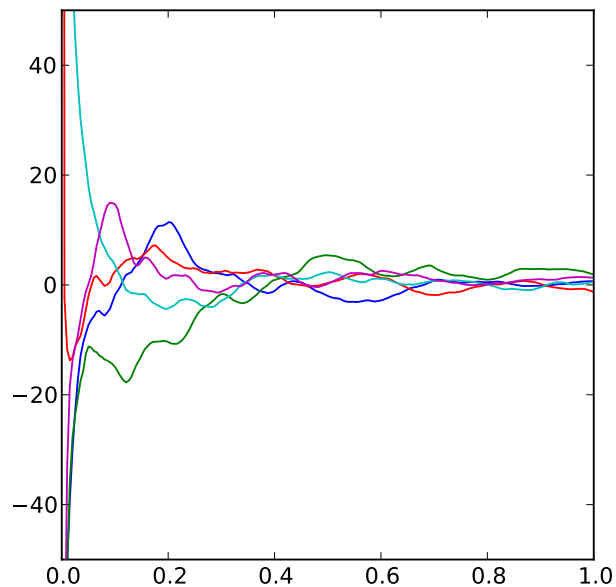
$k(x, y) = f(x)f(y)k_1(x, y)$ is a valid kernel.

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

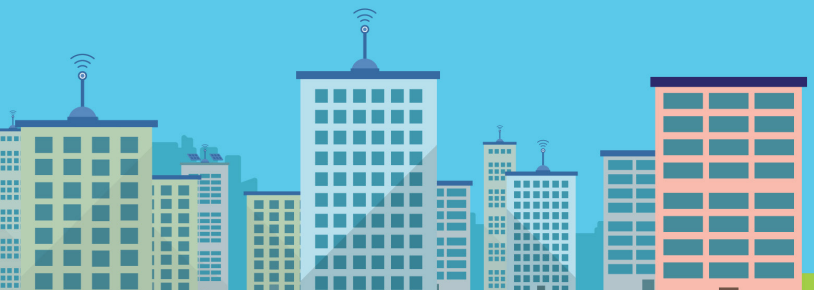
- kernel functions
- kernel combinations ($+$, $\times$, change-point)

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!

Applying linear operators to GPs



Effect of a linear operator

Property (Ginsbourger 2013)

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:

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

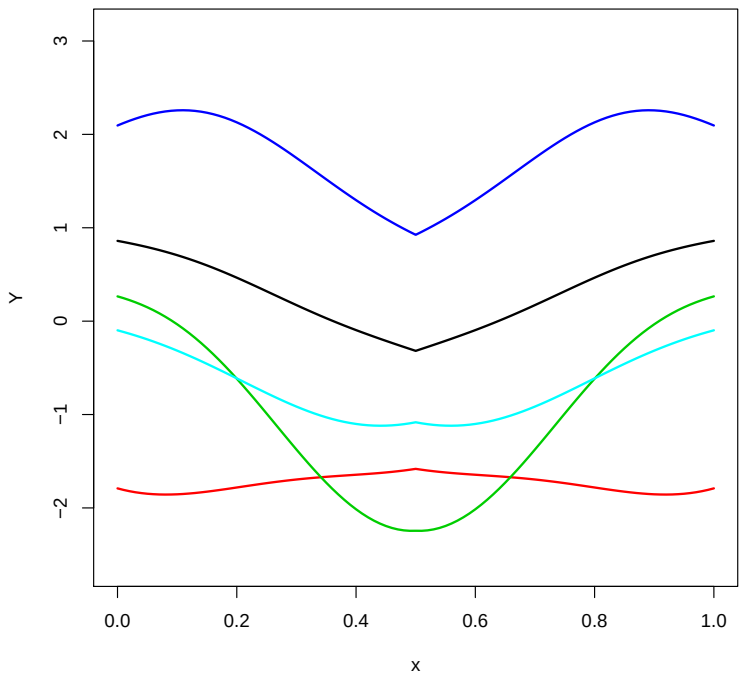
$$L_2 : f(x) \to \frac{f(x) + f(1-x)}{2}.$$

Effect of a linear operator

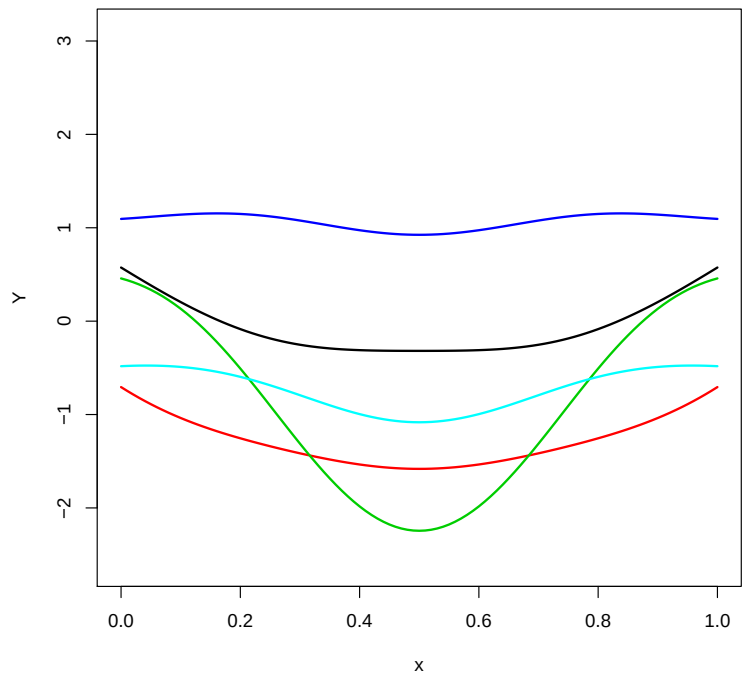
Example

Associated sample paths are

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



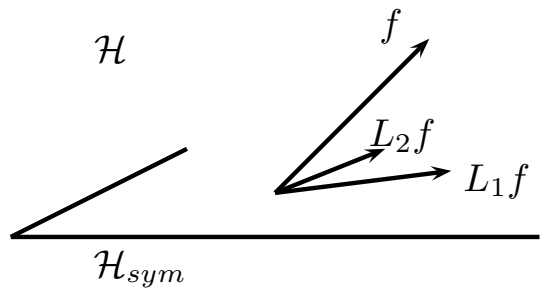
$$k_2 = L_2(L_2(k))$$



The differentiability is not always respected!

Effect of a linear operator

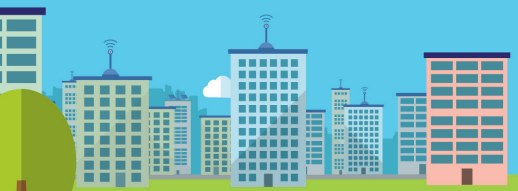
These linear operator are projections onto a space of symmetric functions:

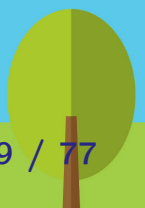


What about the optimal projection?

$\Rightarrow$ This can be difficult... but it raises interesting questions!

Application to sensitivity analysis





The analysis of the influence of the various variables of a d-dimensional function f is often based on the FANOVA:

$$f(\mathbf{x}) = f_0 + \sum_{i=1}^{d} f_i(x_i) + \sum_{i<j} f_{i,j}(x_i, x_j) + \cdots + f_{1,\ldots,d}(\mathbf{x})$$

where $\int f(x_I)\mathrm{d}x_i = 0$ if $i \in I$.

The expressions of the f_I are:

$$f_0 = \int f(\mathbf{x})\mathrm{d}\mathbf{x}$$

$$f_i(x_i) = \int f(\mathbf{x})\mathrm{d}\mathbf{x}_{-i} - f_0$$

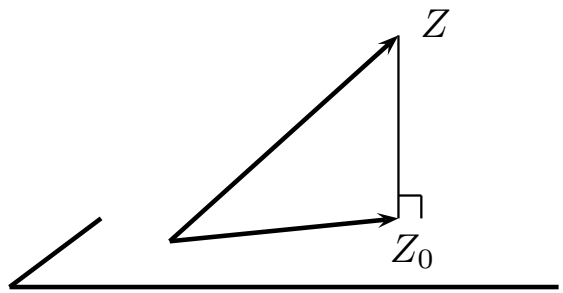
$$f_{i,j}(x_i, x_j) = \int f(\mathbf{x})\mathrm{d}\mathbf{x}_{-ij} - f_i(x_i) - f_j(x_j) + f_0$$

Can we obtain a similar decomposition for a GP?

samples with zero integrals

We are interested in building a GP such that the integral of the samples are exactly zero.

idea: project a GP onto a space of functions with zero integrals:



It can be proved that the orthogonal projection is

$$Z_0(x) = Z(x) - \frac{\int k(x,s)\mathrm{d}s \int Z(s)\mathrm{d}s}{\iint k(s,t)\mathrm{d}s\mathrm{d}t}$$

The associated kernel is:

$$k_0(x,y) = k(x,y) - \frac{\int k(x,s)\mathrm{d}s \int k(y,s)\mathrm{d}s}{\iint k(s,t)\mathrm{d}s\mathrm{d}t}$$

Such 1-dimensional kernels are great when combined as ANOVA kernels:

$$\begin{aligned} k(\mathbf{x},\mathbf{y}) &= \prod_{i=1}^{d}(1 + k_0(x_i,y_i)) \\ &= 1 + \underbrace{\sum_{i=1}^{d} k_0(x_i,y_i)}_{\textit{additive part}} + \underbrace{\sum_{i<j} k_0(x_i,y_i)k_0(x_j,y_j)}_{2^{nd}\ \textit{order interactions}} + \cdots + \underbrace{\prod_{i=1}^{d} k_0(x_i,y_i)}_{\textit{full interaction}} \end{aligned}$$

10d example

Let us consider the test function $f : [0,\ 1]^{10} \to \mathbb{R}$ with $\varepsilon \sim \mathcal{N}(0, 1)$ observation noise:

$$x \mapsto 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon$$

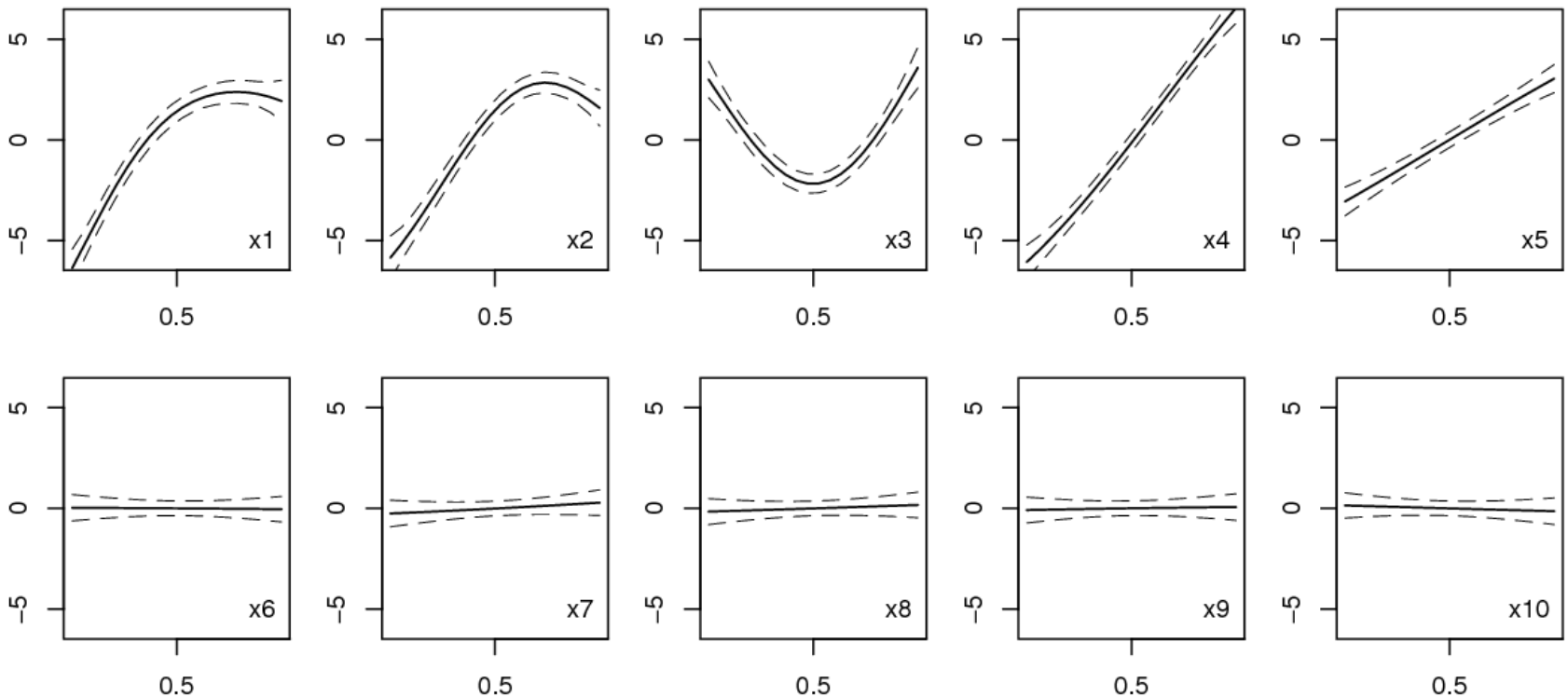
The steps for approximating f with GPR are:

1. Learn f on a DoE (here LHS maximin with 180 points)
2. get the optimal values for the kernel parameters using MLE,
3. build a model based on kernel $\prod(1 + k_0)$

The structure of the kernel allows to split m in sub-models.

$$\begin{aligned} m(\mathbf{x}) &= \left(1 + \sum_i k_0(x_i, X_i) + \sum_{i \neq j} k_0(x_i, X_i) k_0(x_j, X_j) + \dots\right) k(X, X)^{-1} F \\ &= m_0 + \sum m_i(x_i) + \sum_{i \neq j} m_{i,j}(x_i, x_j) + \cdots + m_{1,\dots,d}(x) \end{aligned}$$

The univariate sub-models are:



$\Big($ we had $f(x) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \mathcal{N}(0,1)$ $\Big)$

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

Remarks

- 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

In practice, invertibility issues arise when observation points are close-by. This is specially true if

- 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
- impose a maximum bound to length-scales
- impose a minimal bound for noise variance
- avoid using the Gaussian kernel