Surrogate models and Gaussian Process regression – lecture 3/5

Tuning of Kriging models

Mines St-Étienne – Majeure Data Science – 2016/2017

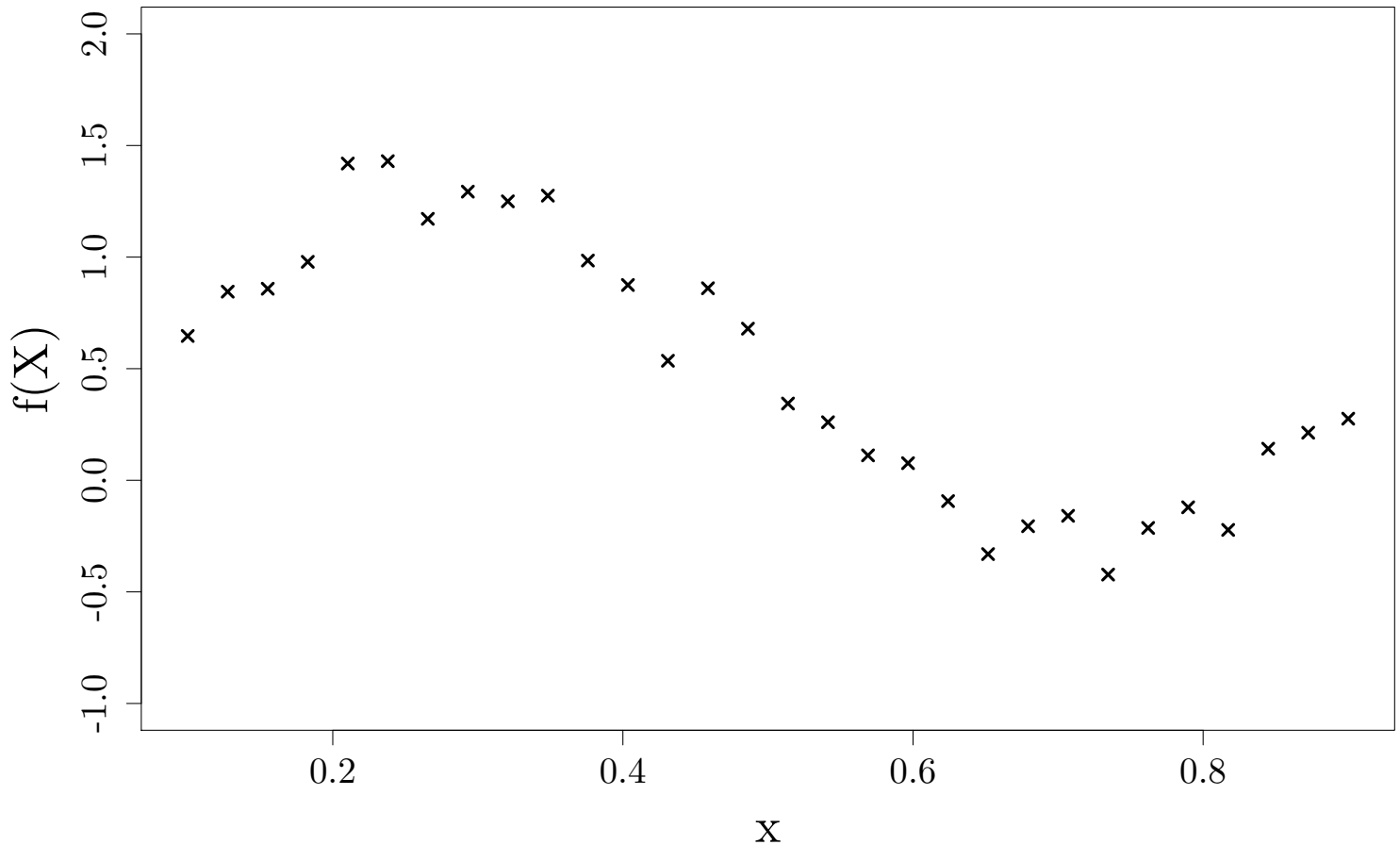
Nicolas Durrande (durrande@emse.fr)

Outline of the lecture:

1. Approximation
   - GPR with noisy observations
2. GPR in practice
   - Overall Steps for GPR
   - Numerical stability
   - Numerical complexity
3. GPR with trend
   - Ordinary kriging
   - Universal kriging
4. Kernel design
   - Making new from old
   - Linear applications

Approximation

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



Exercise:

Let f be a function of interest and let F be a vector of "noisy" observations of f at some input locations X:

$$F = f(X) + \varepsilon \qquad \text{with } \varepsilon \sim \mathcal{N}(0, \tau^2 Id).$$

Let Z be a Gaussian Process corresponding that can be used as prior distribution for f.

1. Compute the conditional mean of $Z(x)|Z(X) + \varepsilon = F$
2. Compute the conditional covariance function

Solution:

1. The conditional mean is

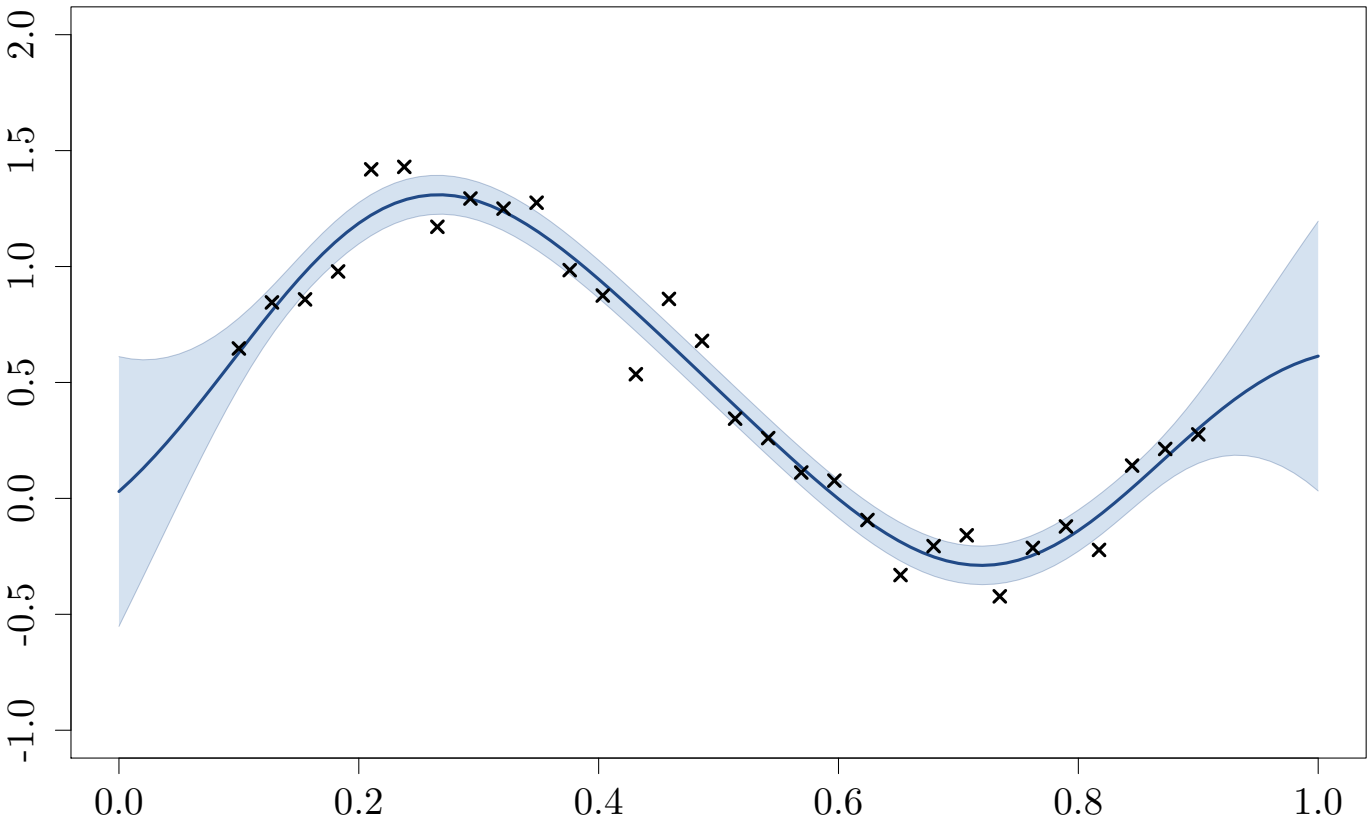
$$\begin{aligned} m(x) &= \mathrm{E}[Z(x)|Z(X)+\varepsilon = F] \\ &= k(x,X)(k(X,X)+\tau^2 Id)^{-1}F \end{aligned}$$

2. The conditional variance is

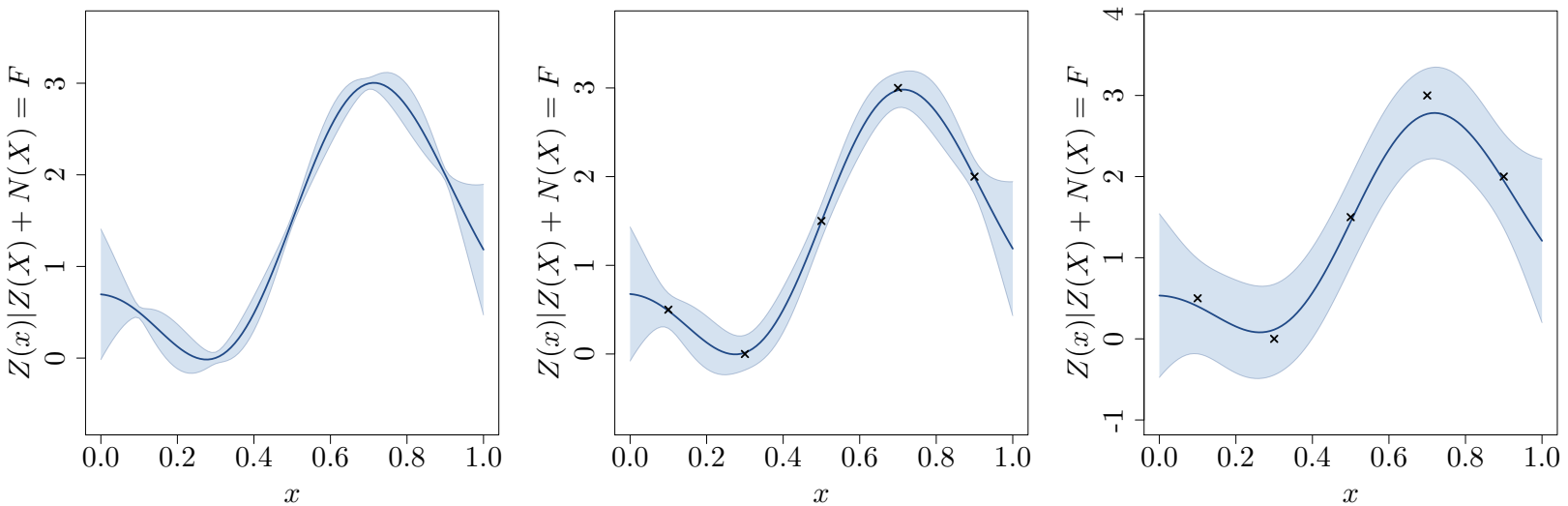
$$\begin{aligned} c(x,y) &= \mathrm{cov}[Z(x),Z(y)|Z(X)+\varepsilon = F] \\ &= k(x,y) - k(x,X)(k(X,X)+\tau^2 Id)^{-1}k(X,y) \end{aligned}$$

Note that is si straightforward to generalize for a noise that is not i.i.d (as long as the noise is Gaussian distributed).

We obtain the following model



Influence of observation noise τ^2:



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

In practice, τ^2 can be estimated with Maximum Likelihood.

GPR in practice

The various steps for building a GPR model are:

1. Create a DoE
   - What is the overall evaluation budget?
   - What is my model for?
2. Choose a kernel
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: the matrix is not invertible
- Error: the matrix is not positive definite

Covariance matrices are positive semi-definite. Null eigenvalues arise if one information is repeated.

Example

For $X = (0.1, 0.1, 0.4, 0.6, 0.8)$, the covariance of a squared exponential kernel with parameters $\sigma^2 = 1$, $\theta = 0.2$ is:

$$k(X, X) = \begin{pmatrix} 1.00 & 1.00 & 0.32 & 0.04 & 0.00 \\ 1.00 & 1.00 & 0.32 & 0.04 & 0.00 \\ 0.32 & 0.32 & 1.00 & 0.61 & 0.14 \\ 0.04 & 0.04 & 0.61 & 1.00 & 0.61 \\ 0.00 & 0.00 & 0.14 & 0.61 & 1.00 \end{pmatrix}$$

The first two columns are the same, so the matrix is not invertible.

It is particularly interesting to look at the eigenvectors associated with null eigenvalues. On the previous example, this eigenvector is

$$P_0 = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, 0, 0\right)^t$$

Now, 2 situations can be distinguished

(A) The observations are compatible with model: $P_0^t F = 0$.
- the model is appropriate and one observation can be removed without any loss of information.

(B) The observations **are not** compatible with model: $P_0^t F \neq 0$.
- the model is not appropriate and it should be modified. For example, observation noise can be added.

In both cases, the covariance matrix will become invertible.

In practice, invertibility issues may arise if observations 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 a (very) small observation noise
- impose a maximum bound to length-scales
- impose a minimal bound for noise variance
- choose a Matérn kernel

A few words on the Complexity of GPR

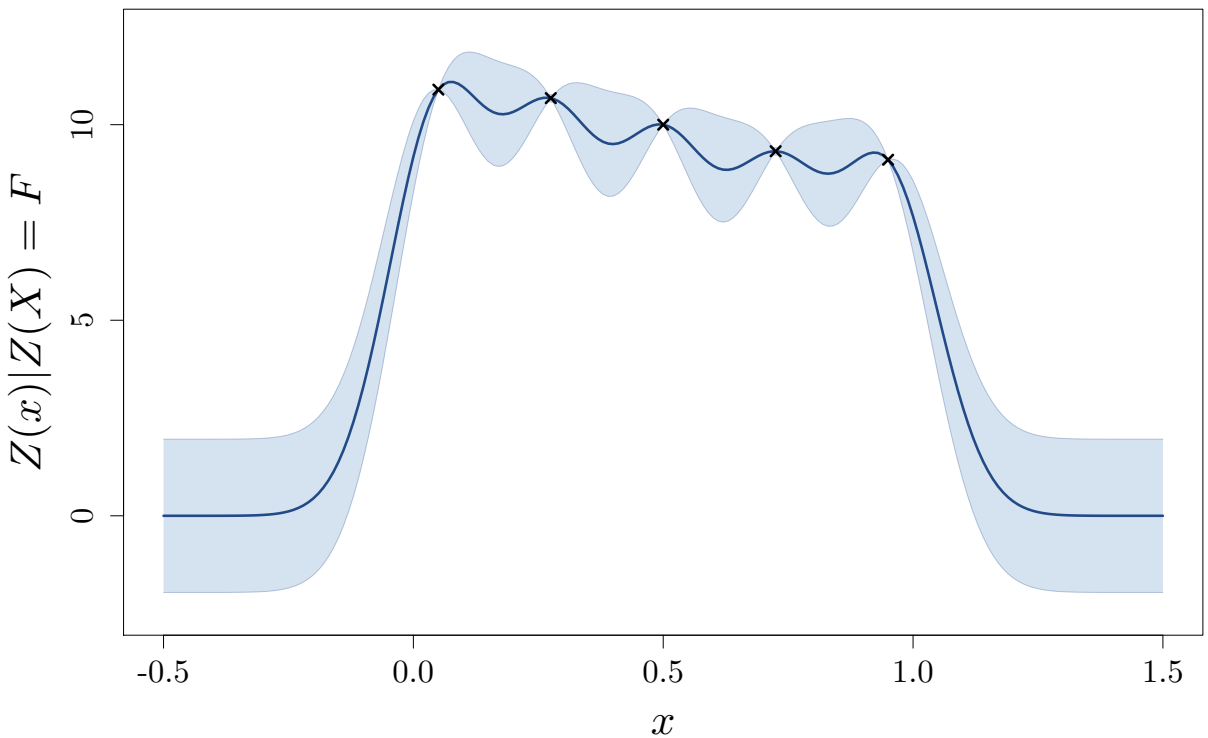
- Storage footprint: We have to store the covariance matrix which is $n \times n$.
- Complexity: We have to invert the covariance matrix, which requires is $\mathcal{O}(n^3)$.
  $\Rightarrow$ This limits the number of observation points is between 1000 and 10 000.

Note that the complexity do not depend on the dimension of the input space!

GPR with trend

We have seen that GPR models go back to zero if we consider a centred prior.

This behaviour is not always wanted

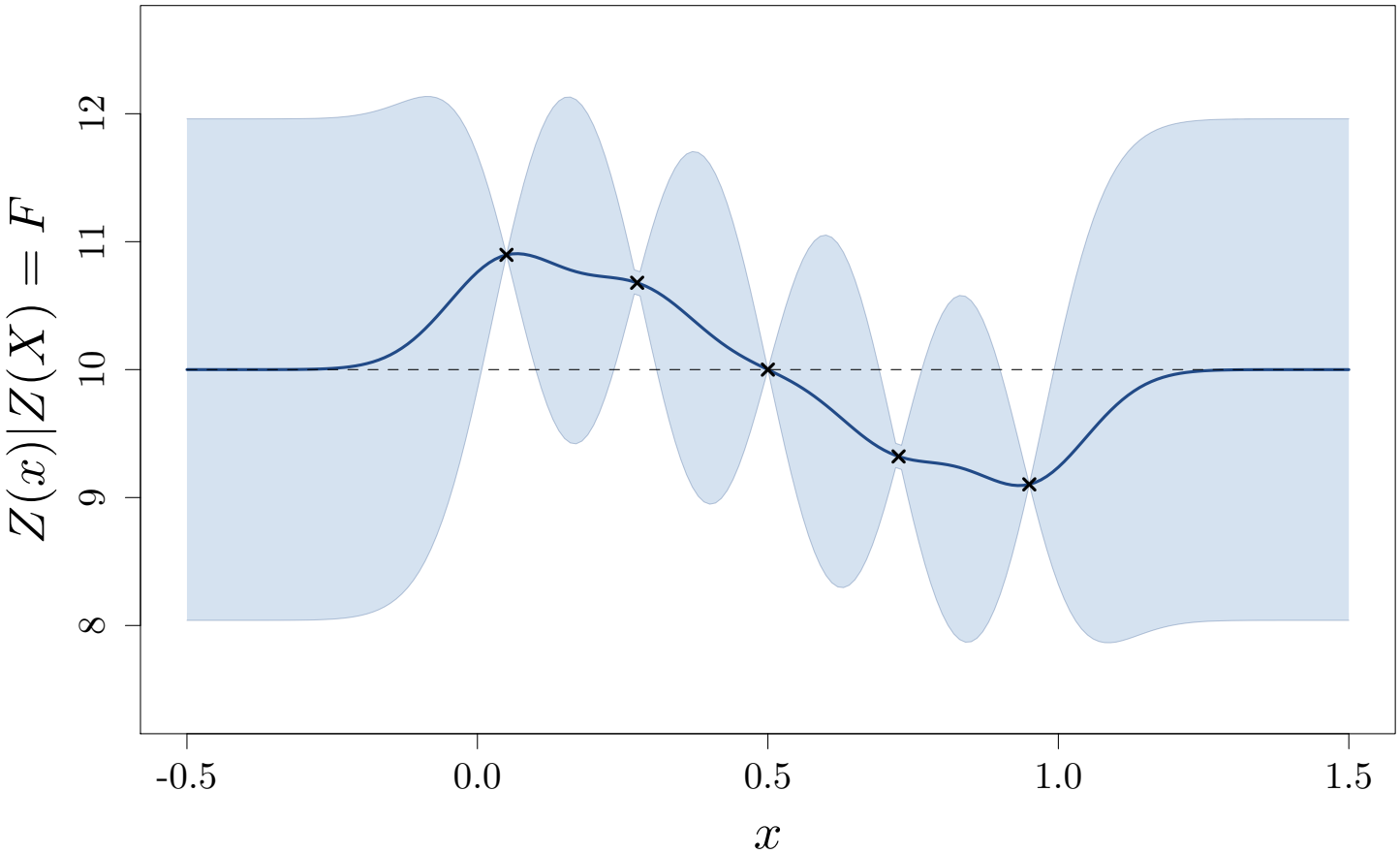


If the trend $t(.)$ is known, the usual formulas for multivariate normal conditional distribution apply:

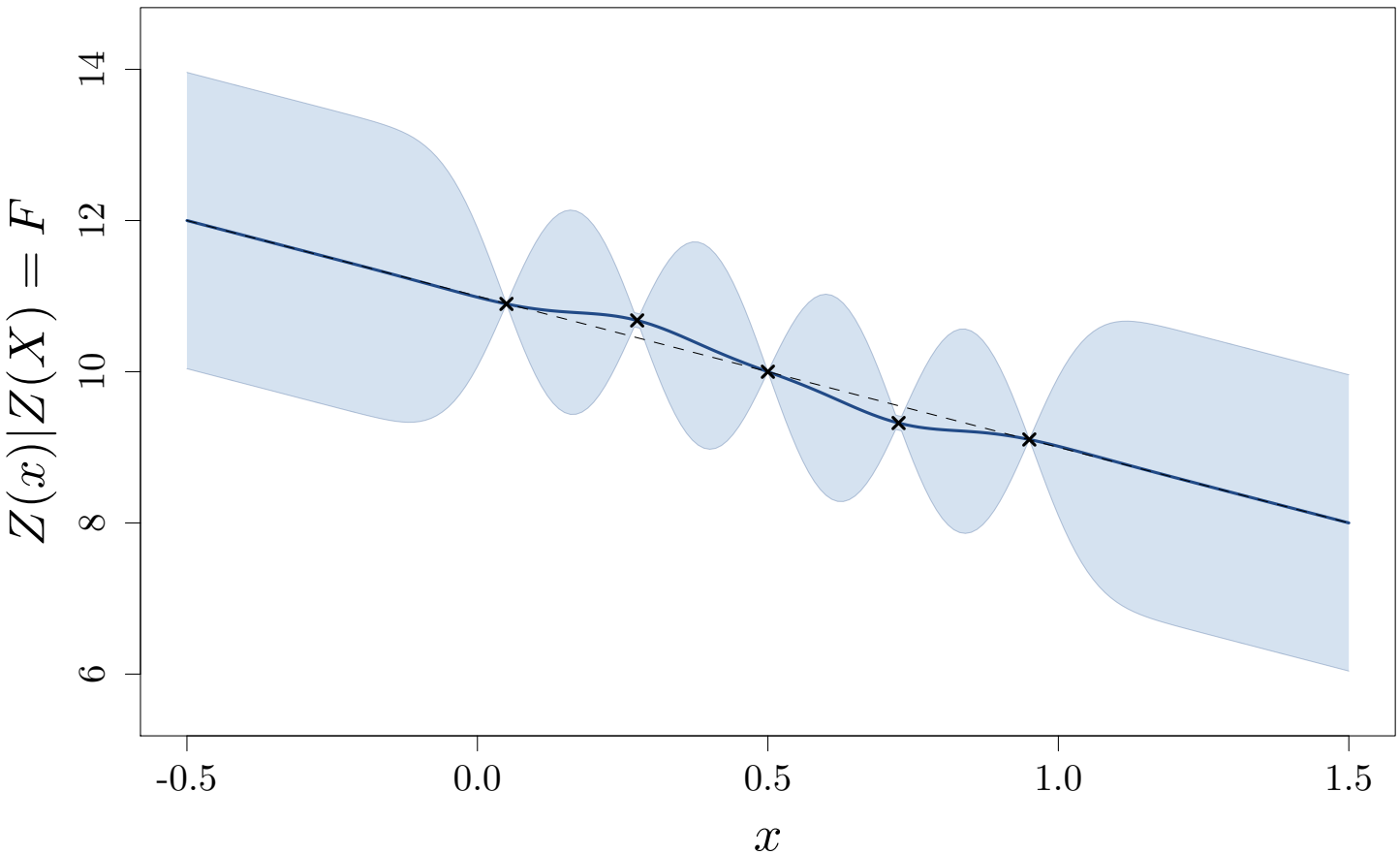
$$\begin{aligned} m(x) &= \mathrm{E}[Z(x)|Z(X) = F] \\ &= t(x) + k(x, X)k(X, X)^{-1}(F - t(X)) \\ c(x, y) &= \mathrm{cov}[Z(x), Z(y)|Z(X) = F] \\ &= k(x, y) - k(x, X)k(X, X)^{-1}k(X, y) \end{aligned}$$

We can see that the trend is subtracted first and then added in the end.

In the previous example, we can consider that trend is constant $t(x) = 10$:



We can also try a linear trend $t(x) = 11 - 2x$:

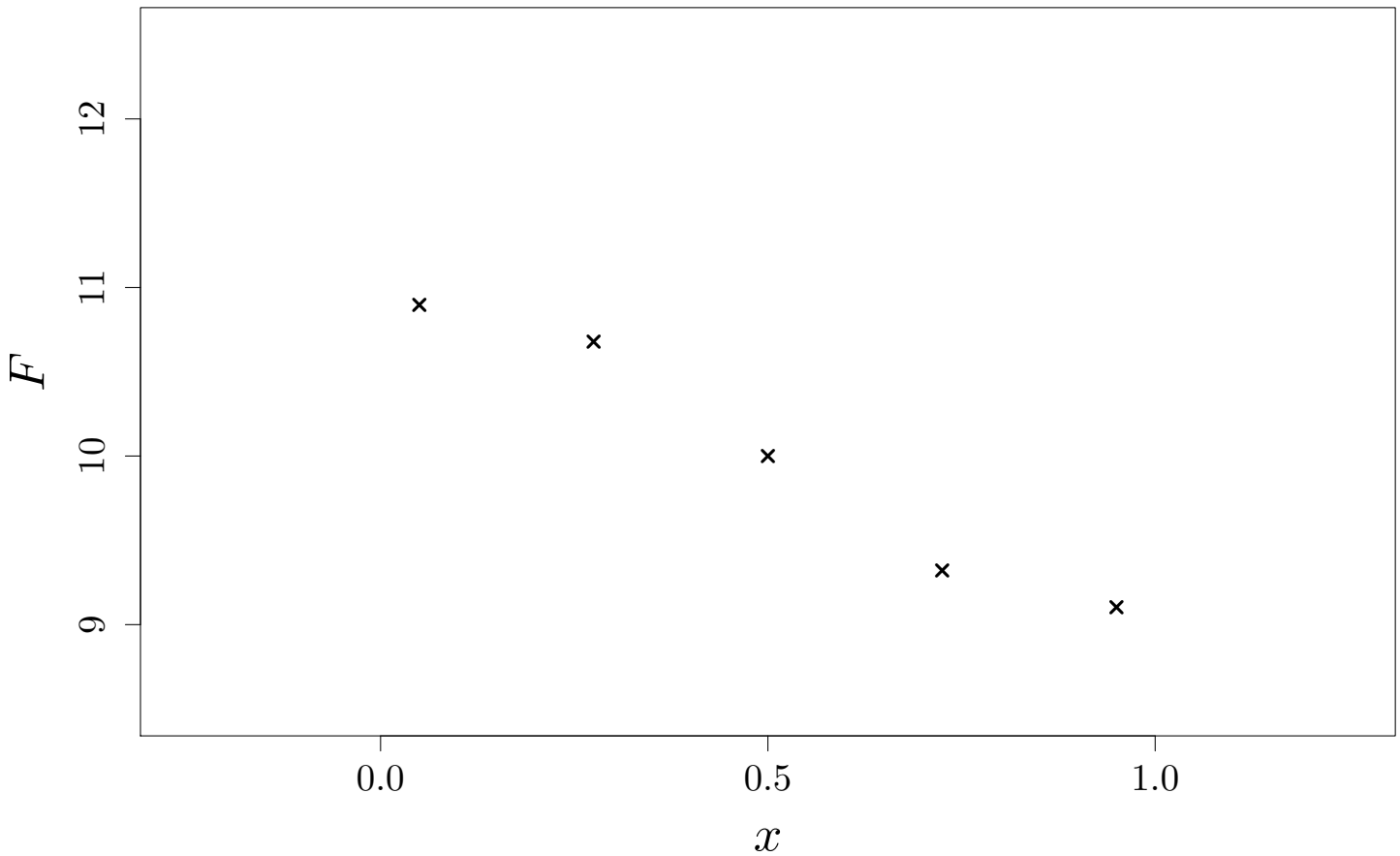


In practice, the trend is often unknown... The question is then how to estimate it.

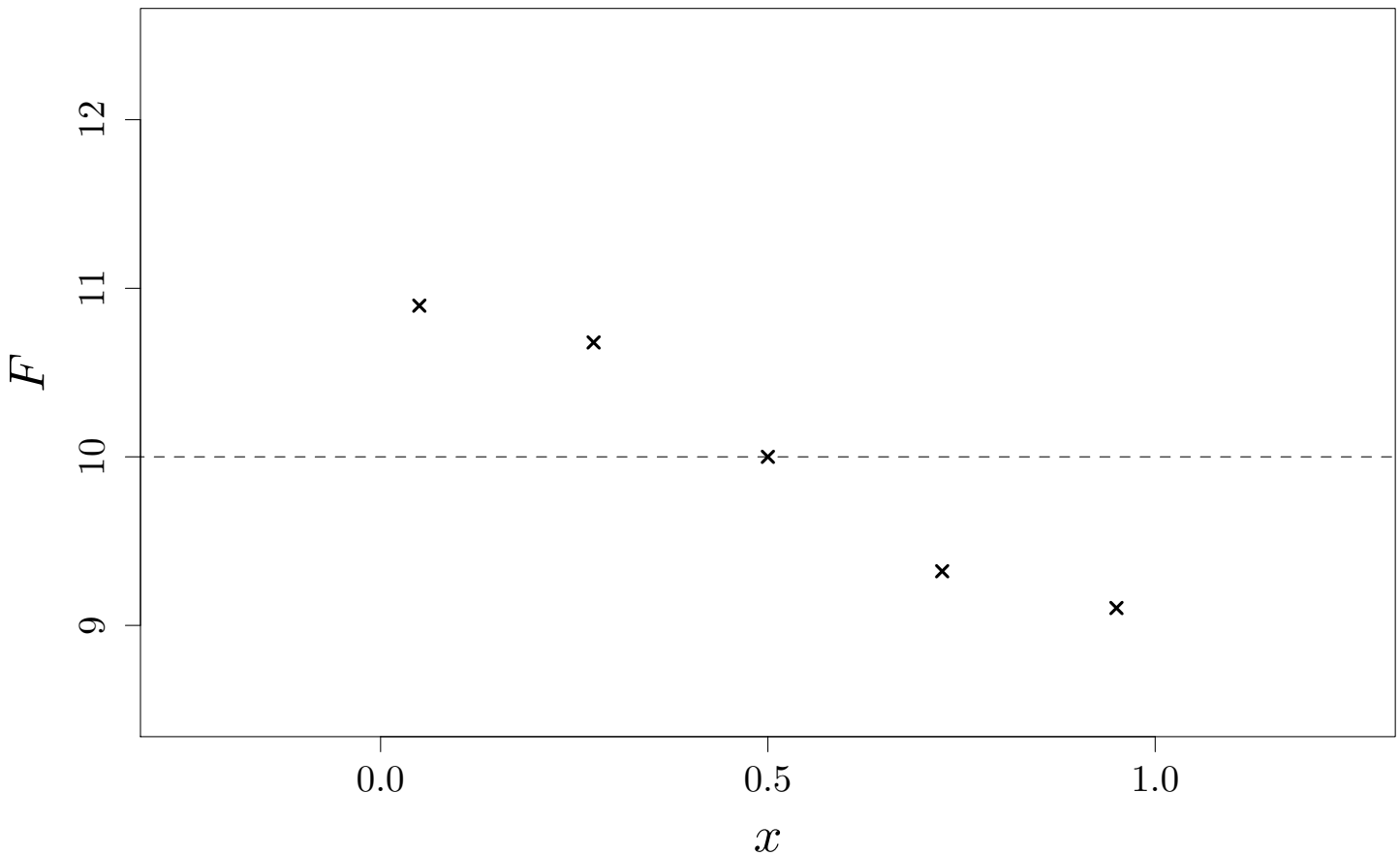
We will distinguish:

- **simple kriging**: there is no trend or it is known
- **ordinary kriging**: the trend is a constant
- **universal kriging**: the trend is given by basis functions

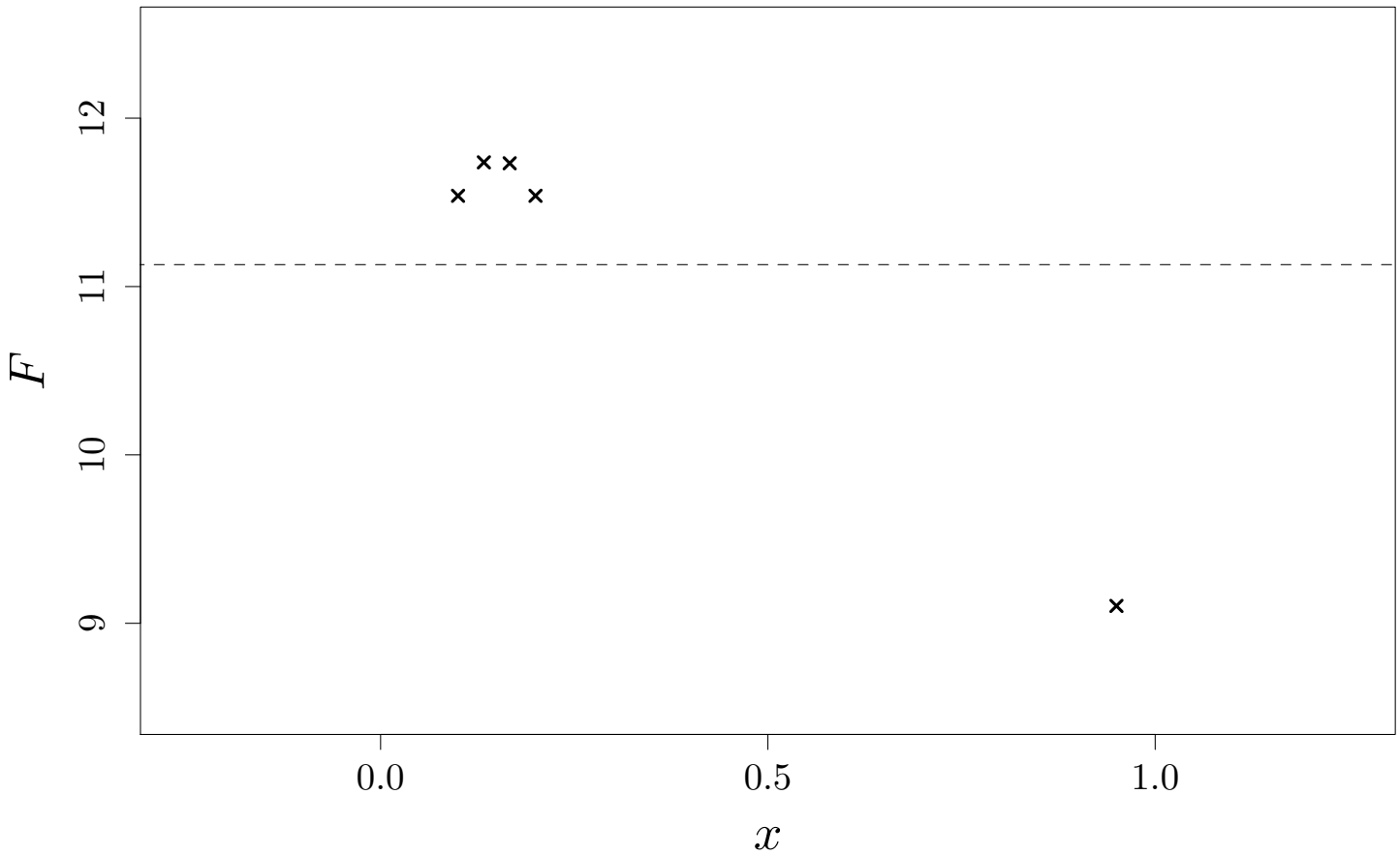
We will first focus on **ordinary kriging**. We thus need to estimate a constant:



The idea of considering $t(x) = \text{mean}(F)$ looks all right on this example...



but not on this one.



Any other idea?

We have considered maximum likelihood estimation for the kernel's parameter... why not doing the same thing here ?

Exercise

1. Compute the maximum likelihood estimation $\hat{t}$ of t. A few hints
   - consider the log-likelihood
   - take the derivative
   - find where it is null
2. What can we recognize in this expression ?

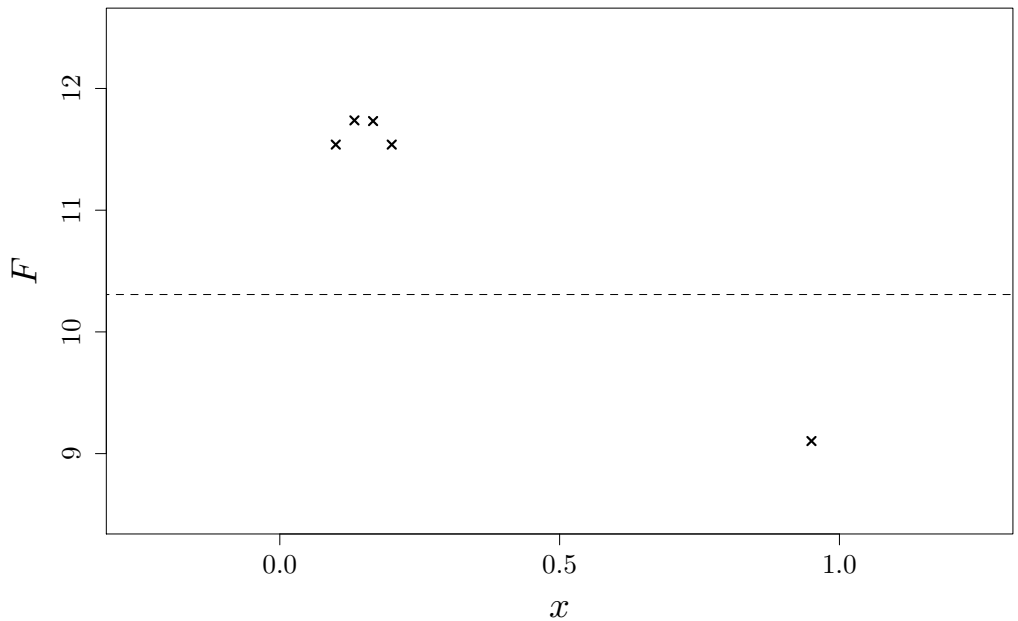
We recall that the likelihood is

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

Solution

1. We obtain $\hat{t} = \dfrac{\mathbf{1}^t k(X,X)^{-1} F}{\mathbf{1}^t k(X,X)^{-1} \mathbf{1}}$
2. It can be seen as an orthogonal projection $t = \dfrac{\langle \mathbf{1}, F \rangle}{\langle \mathbf{1}, \mathbf{1} \rangle}$ for a inner product given by $k(X,X)^{-1}$.

On the previous example we obtain $t = 10.3$:



Under the hypothesis $F = Z(X)$, the estimation $\hat{t} = \dfrac{\mathbf{1}^t k(X,X)^{-1} F}{\mathbf{1}^t k(X,X)^{-1}\mathbf{1}}$ is a sample from $T = \dfrac{\mathbf{1}^t k(X,X)^{-1} Z(X)}{\mathbf{1}^t k(X,X)^{-1}\mathbf{1}}$.

The distribution of T is Gaussian with moments:

$$\mathrm{E}[T] = \frac{\mathbf{1}^t k(X,X)^{-1}\,\mathrm{E}[Z(X)]}{\mathbf{1}^t k(X,X)^{-1}\mathbf{1}} = t$$

$$\mathrm{var}[T] = \frac{\mathbf{1}^t k(X,X)^{-1}\,\mathrm{var}[Z(X)] k(X,X)^{-1}\mathbf{1}}{(\mathbf{1}^t k(X,X)^{-1}\mathbf{1})^2} = \frac{1}{\mathbf{1}^t k(X,X)^{-1}\mathbf{1}}$$

The expression of the **best predictor** is given by the usual conditioning of a GP:

$$m(x) = \mathrm{E}[Z(x)|Z(X) = F] = \hat{t} - k(x, X)k(X, X)^{-1}(F - \hat{t})$$

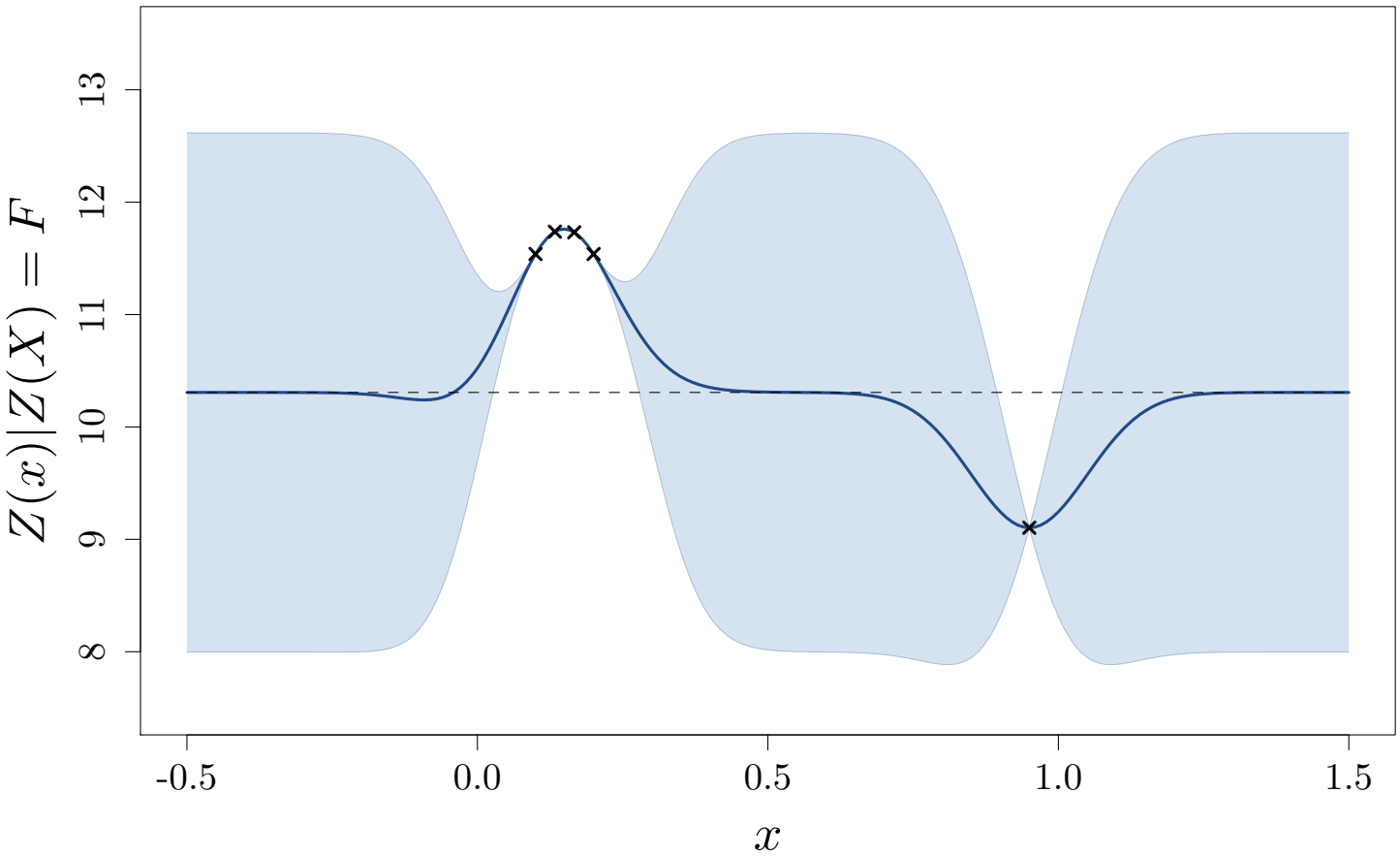
Regarding the **model variance**, it must account for the estimator's variance. We will use the law of total Variance :

$$\mathrm{var}[X] = \mathrm{E}[\mathrm{var}(X|Y)] + \mathrm{var}[\mathrm{E}(X|Y)]$$

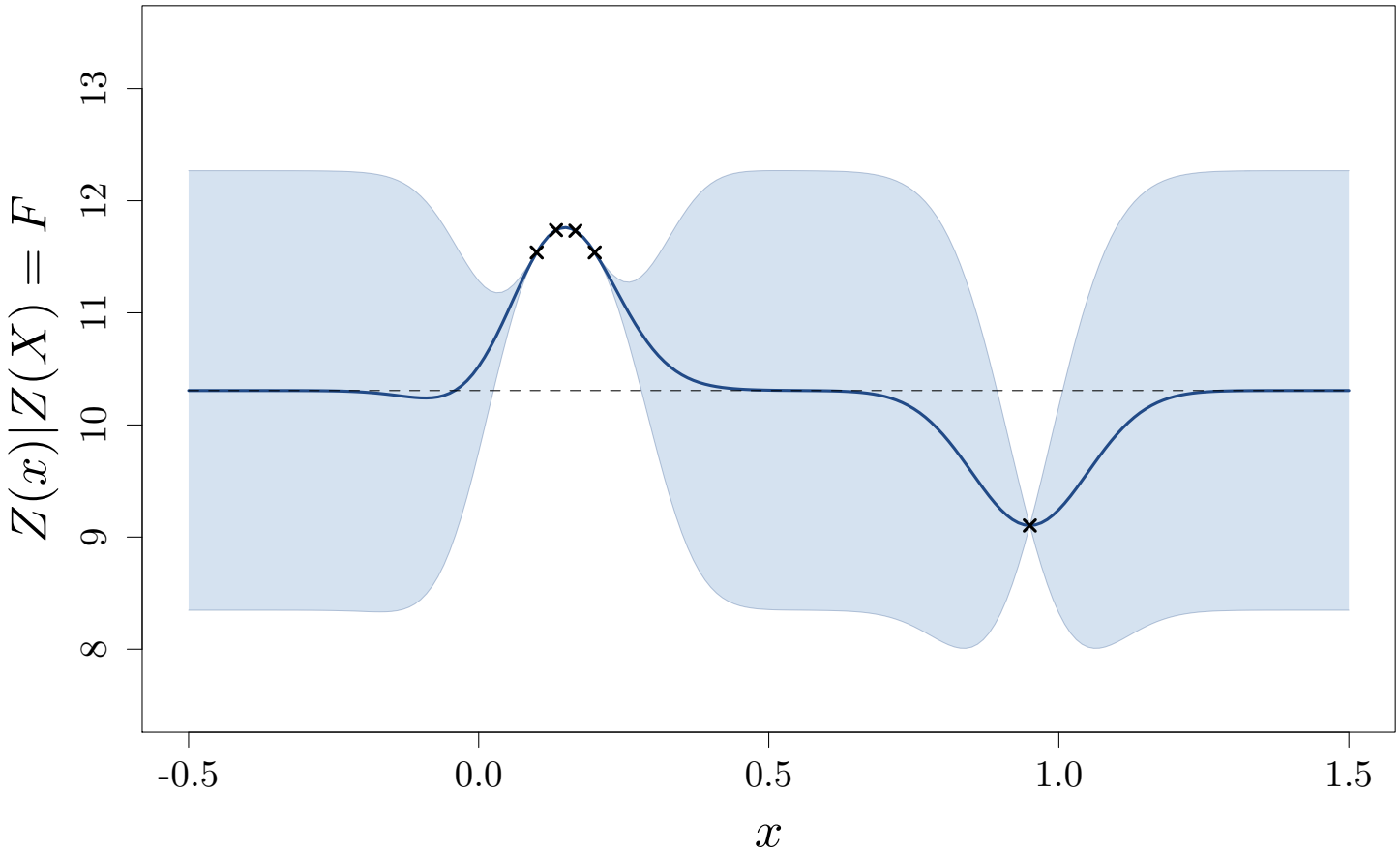
If we apply this to the GPR variance prediction we get:

$$\mathrm{var}[Z(x)|Z(X)] = k(x, x) - k(x, X)k(X, X)^{-1}k(X, x) + \frac{(\mathbf{1} + k(x, X)k(X, X)^{-1}\mathbf{1})^t(\mathbf{1} + k(x, X)k(X, X)^{-1}\mathbf{1})}{\mathbf{1}^t k(X, X)^{-1}\mathbf{1}}$$

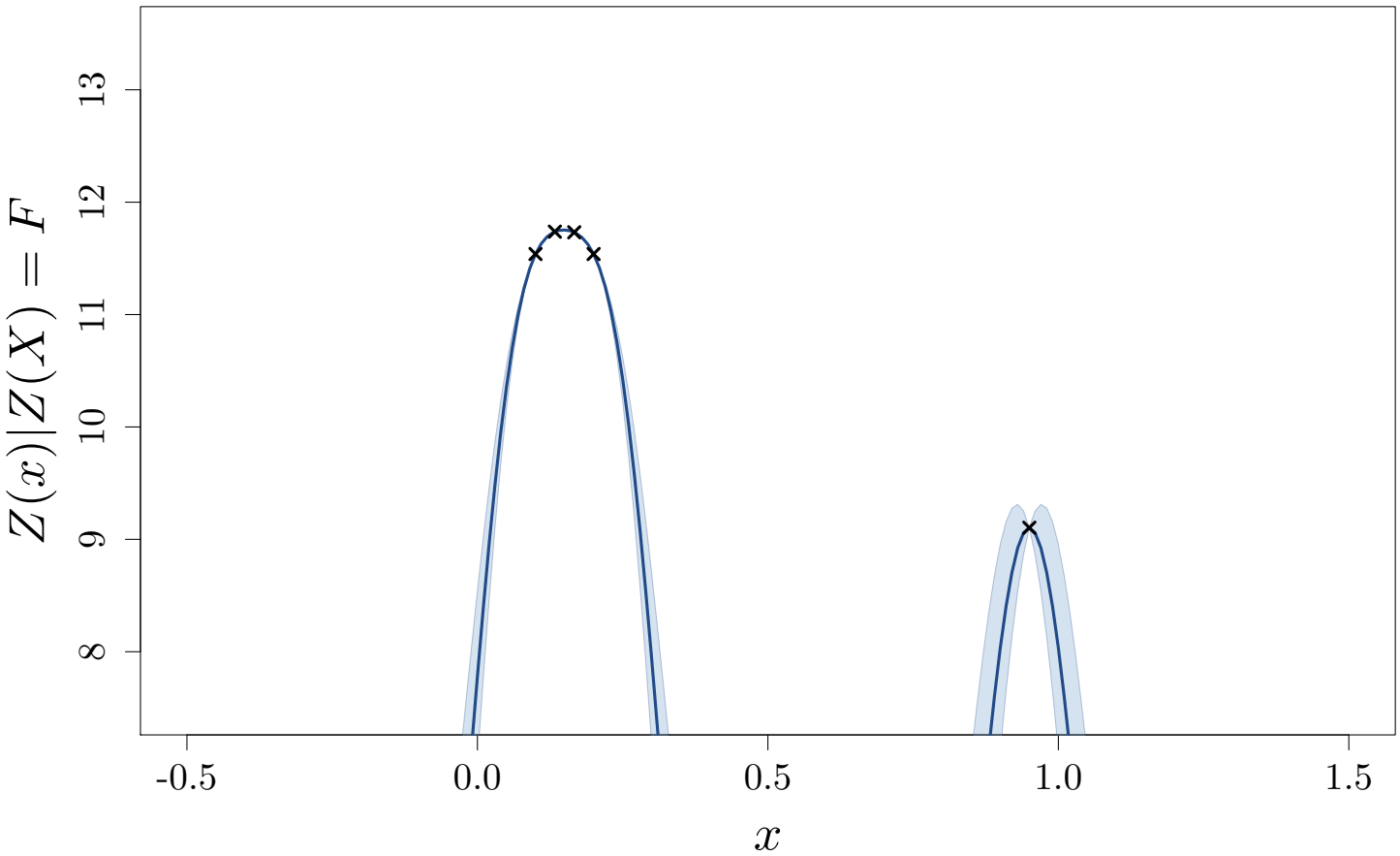
On the previous example we obtain:



We would have obtain this the mean were considered known.



it can be compared with simple kriging



If the trend is not constant but linear, quadratic, etc. it is interesting to consider the following probabilistic model for the prior:

$$Z(x) = Y(x) + \sum_i \beta_i h_i(x)$$

where the $h_i(x)$ are basis functions and the β_i are unknown scalars.

As previously, we can consider the maximum likelihood estimator

$$\hat{\beta} = (H^t k(X,X)^{-1} H)^{-1} H^t k(X,X)^{-1} F$$

where H is the matrix of general term $H_{i,j} = h_j(X_i)$.

The final equations are very similar to ordinary kriging:

Universal kriging

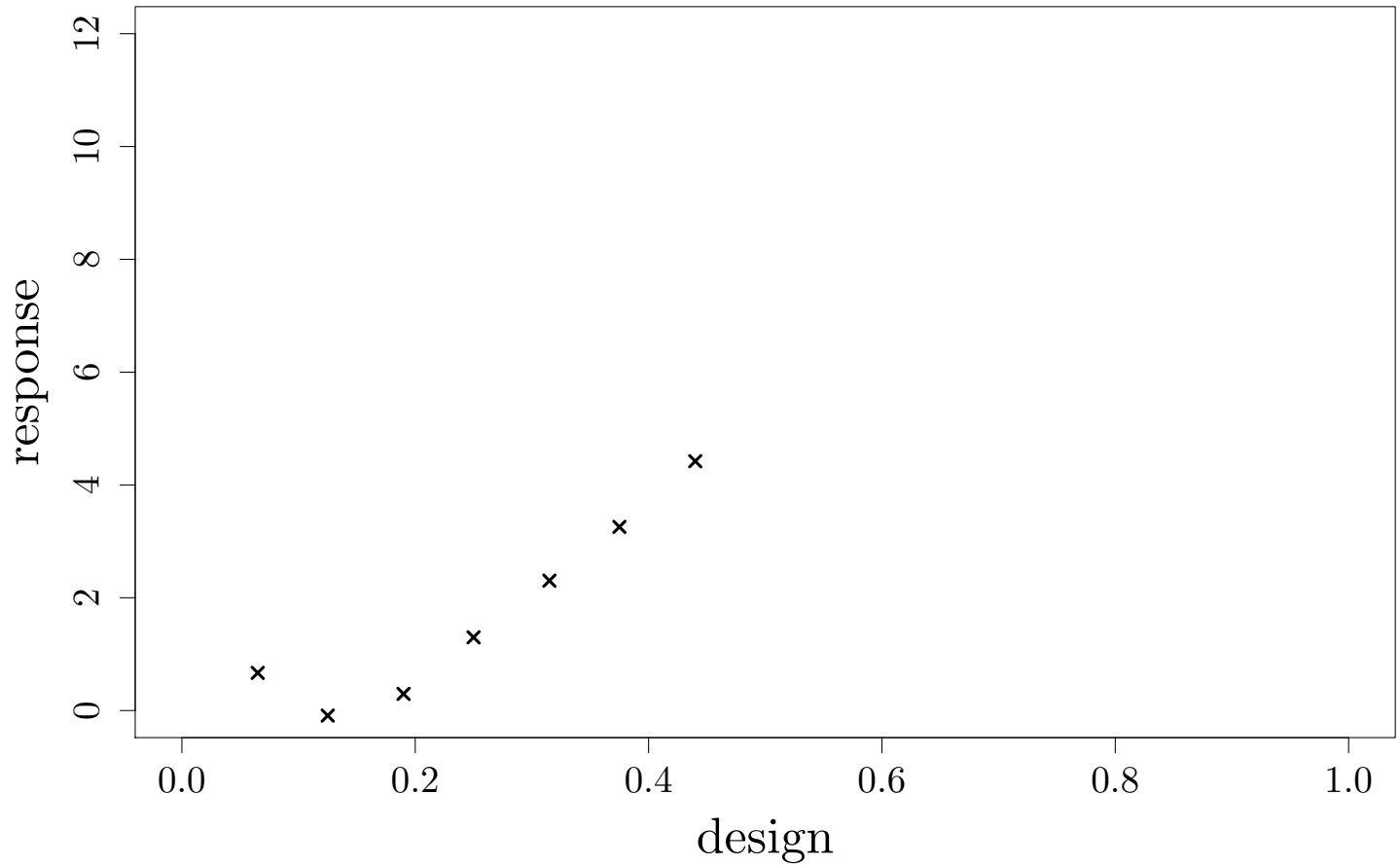
$$
\begin{aligned}
m(x) &= h(x)^t\hat{\beta} - k_x K^{-1}(F - h(X)^t\hat{\beta}) \\
c(x,y) &= k(x,y) - k_x K^{-1} k_y^t \\
&\quad + (h(x)^t + k_x K^{-1} H)^t (H^t K^{-1} H)^{-1} (h(y)^t + k_y K^{-1} H)
\end{aligned}
$$

where $k_x = k(x, X)$ and $K = k(X, X)$.

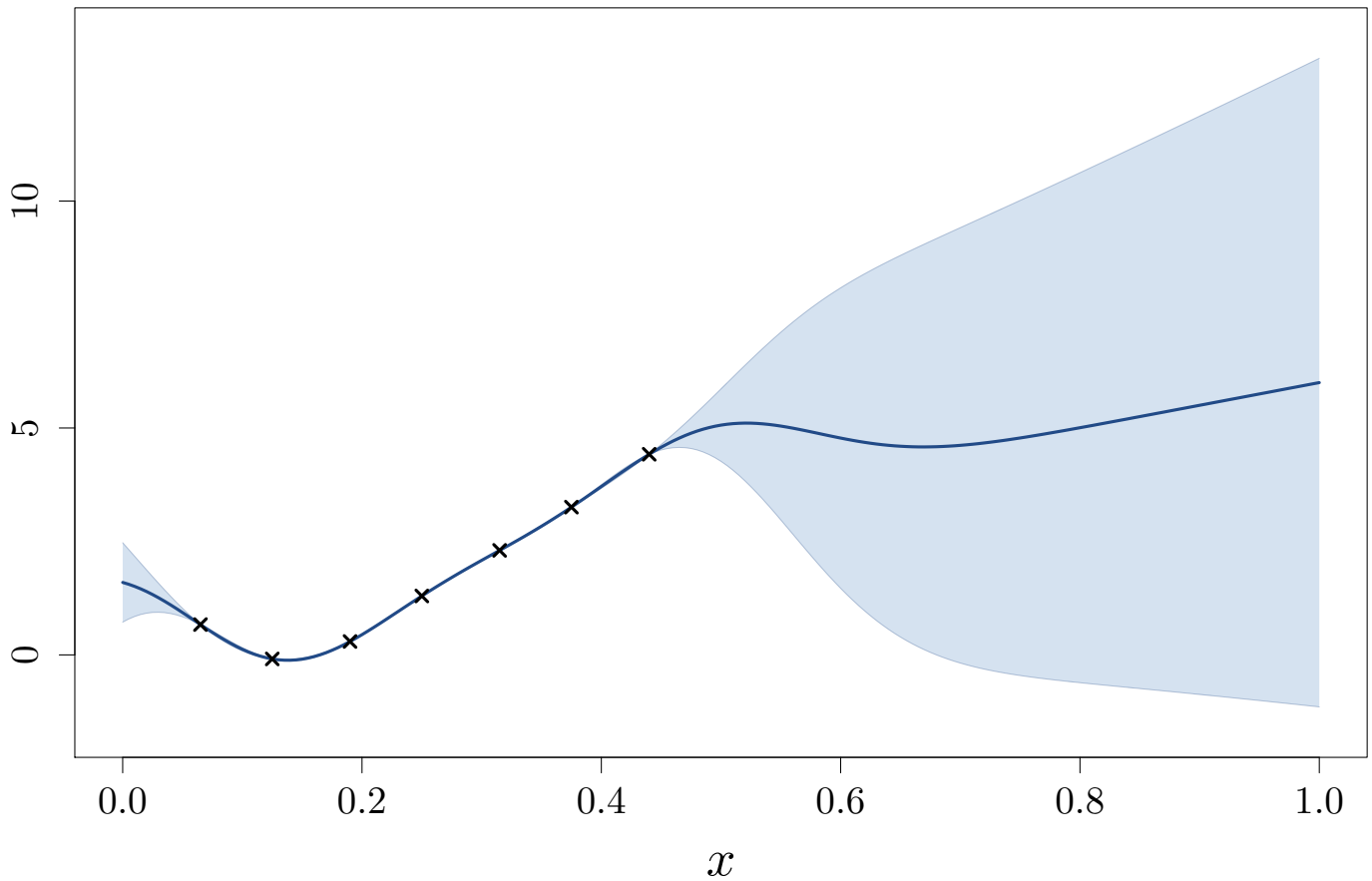
Remarks

- Ordinary kriging is a special case of universal kriging with only one constant basis function.
- The model always interpolates whatever $\hat{\beta}$ is.
- the trend part can be seen as generalised least square (regression with correlated residuals)

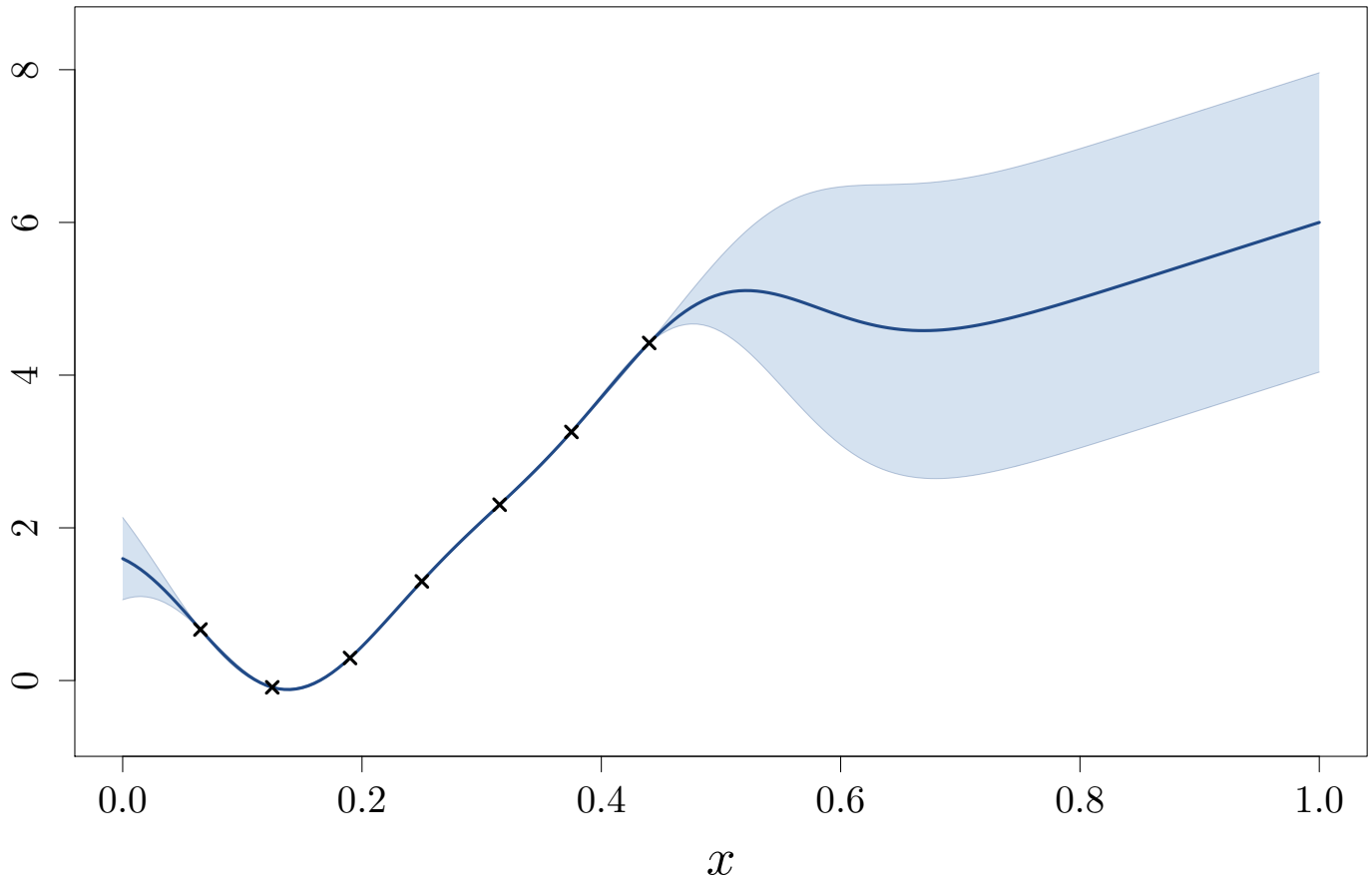
We consider the following example



Universal kriging model with linear trend: $h_1(x) = 1$, $h_2(x) = x$.



It can be compared to simple kriging with known trend



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

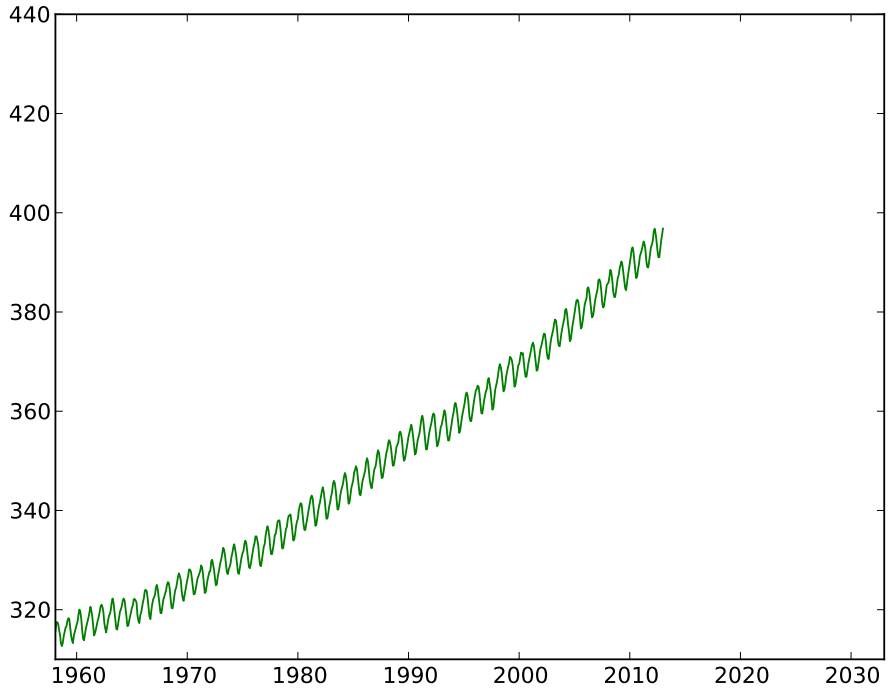
All these operations will preserve the positive definiteness.

How can this be useful?

Sum of kernels over the same space

Example (The Mauna Loa observatory dataset)

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

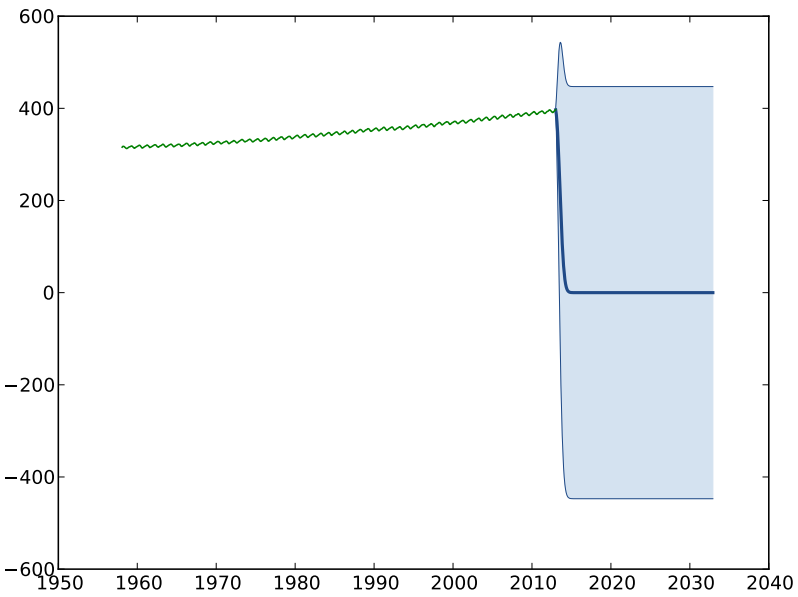


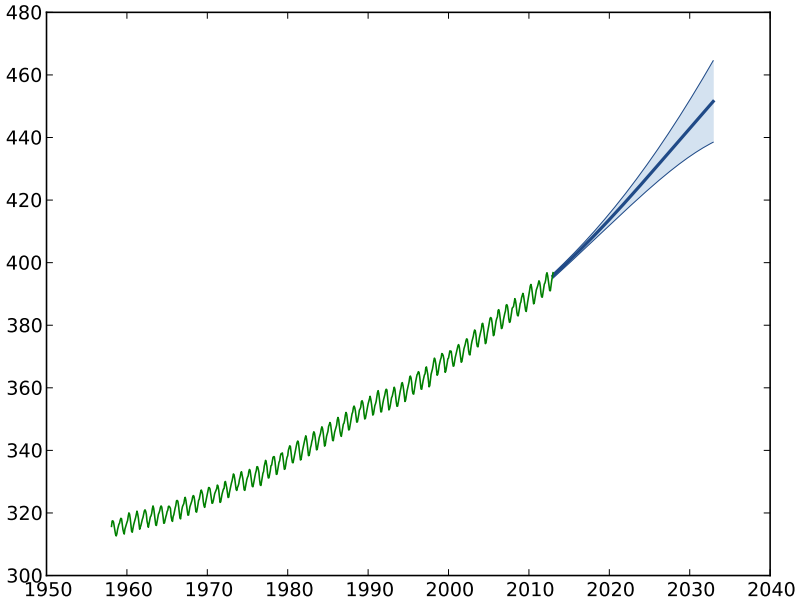
Let's try to predict the concentration for the next 20 years.

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

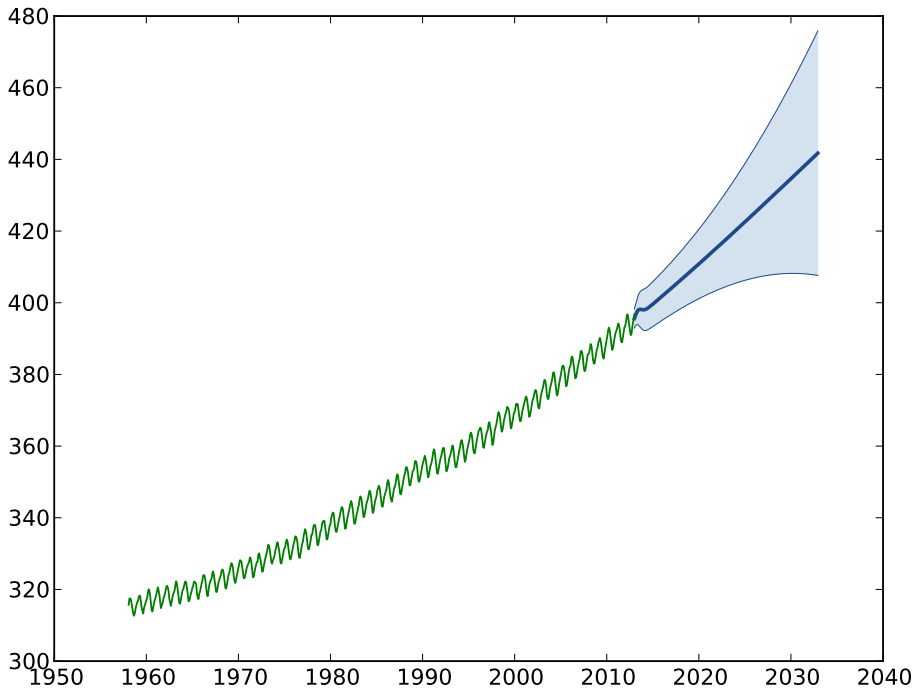
What happen if we sum both kernels?

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

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

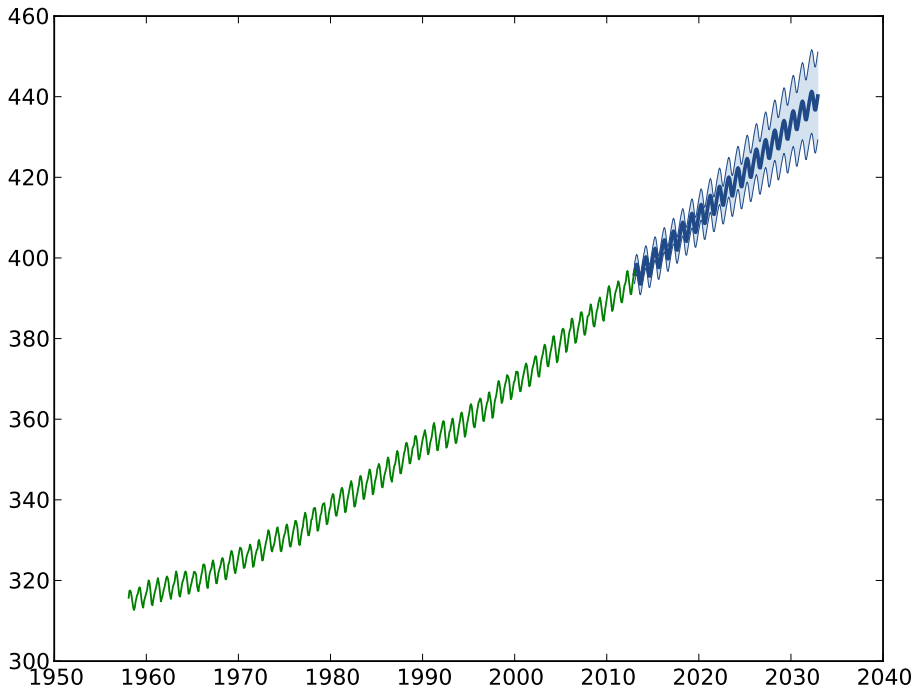
We can try the following kernel:

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

Sum of kernels over the same space

We can try the following kernel:

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



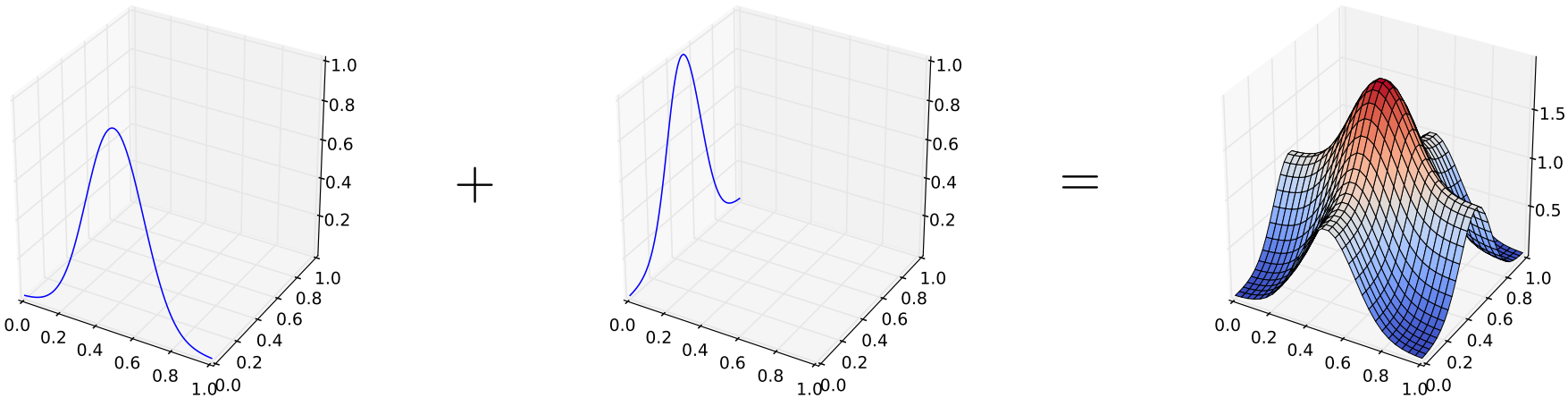
Once again, the model is significantly improved.

Sum of kernels over tensor space

Property

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

is a valid covariance structure.

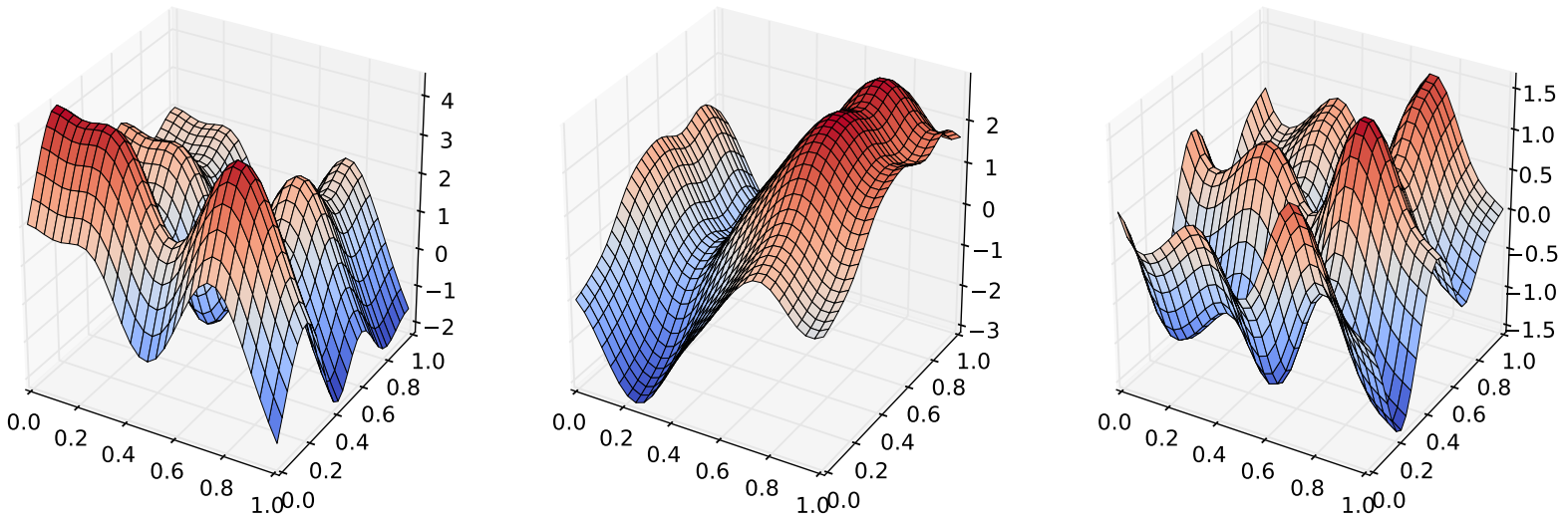


Remark:

- 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)

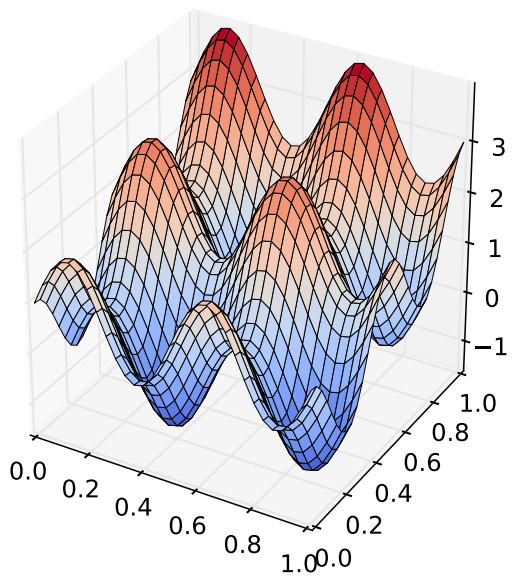
Tensor Additive kernels are very useful for

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

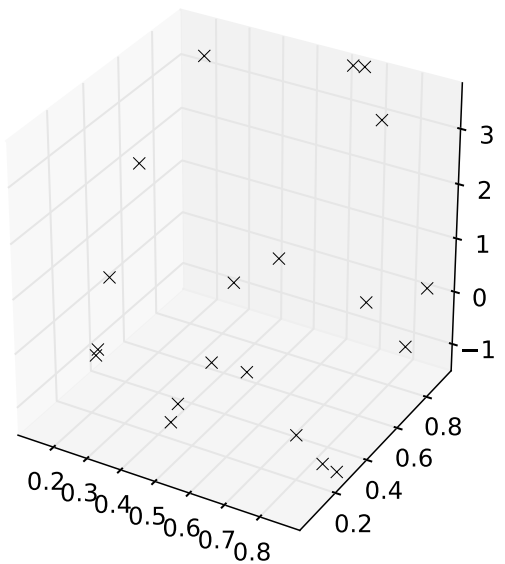
Sum of kernels over tensor space

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

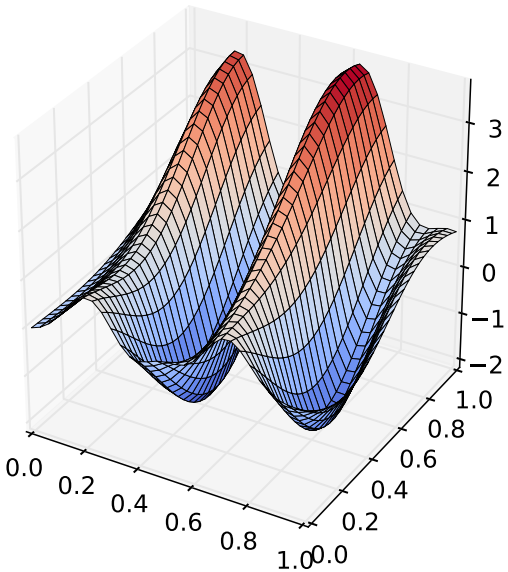


Sum of kernels over tensor space

We obtain the following models:

Gaussian kernel

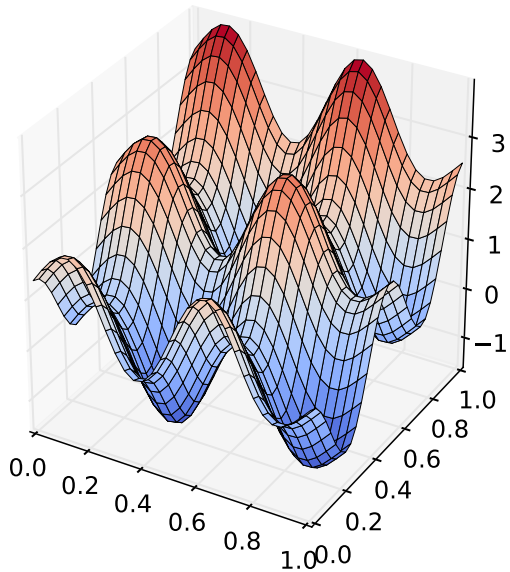
Mean predictor



RMSE is 1.06

Additive Gaussian kernel

Mean predictor



RMSE is 0.12

Sum of kernels over tensor space

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:

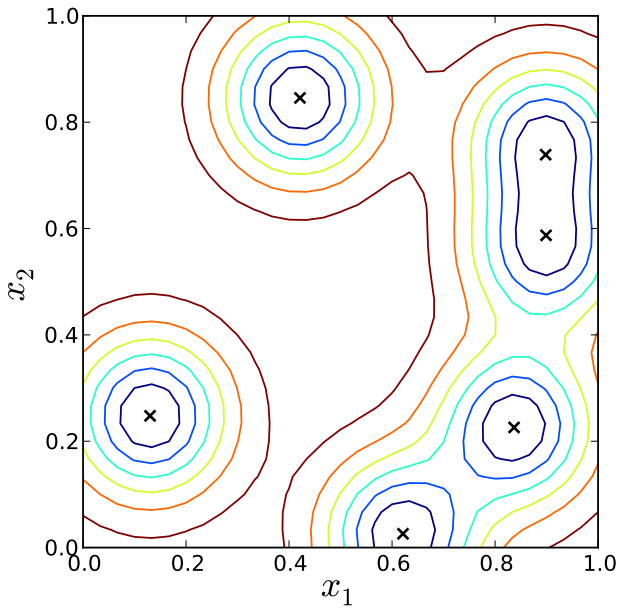
$$m_1(x_1) = \mathsf{E}\left[Z_1(x_1) \mid Z_1(X_1) + Z_2(X_2) = F \right]$$

Sum of kernels over tensor space

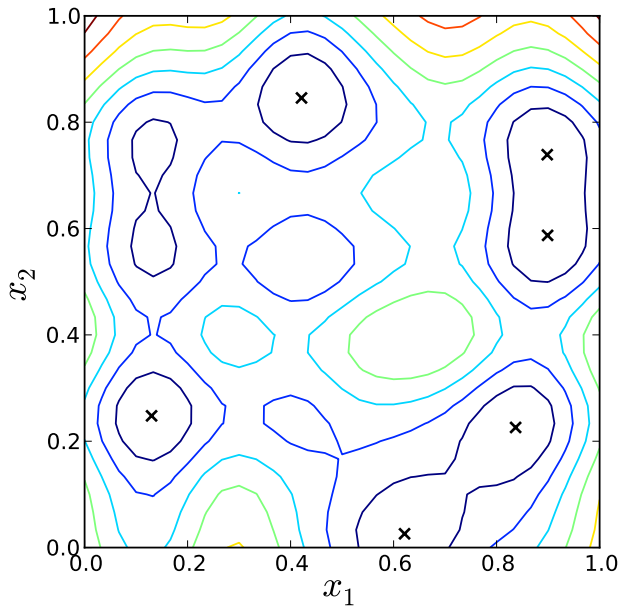
Remark

- The prediction variance has interesting features

pred. var. with kernel product

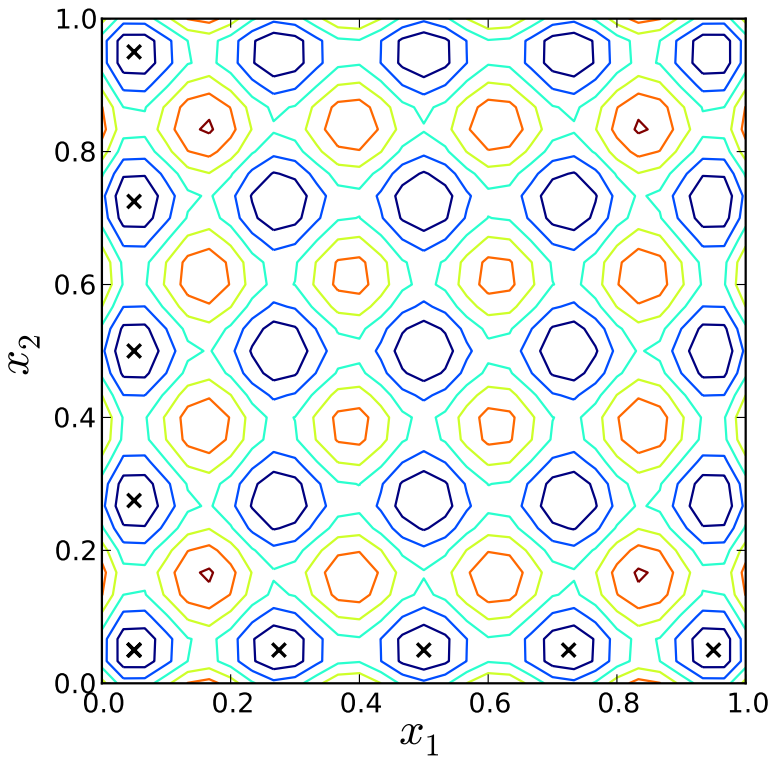


pred. var. with kernel sum



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

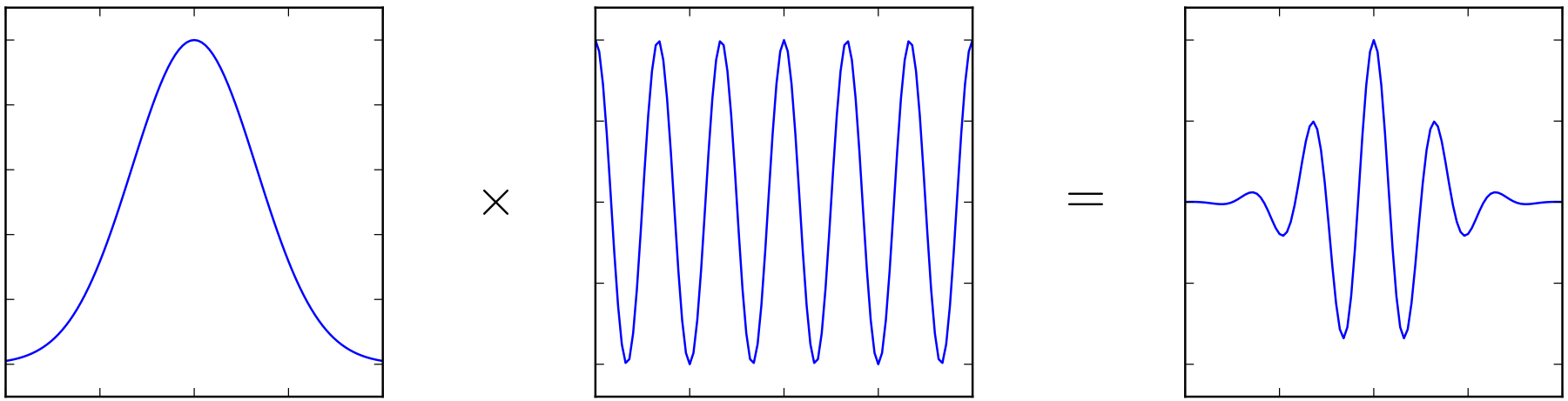
Property

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

is valid covariance structure.

Example

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



Product over the tensor space

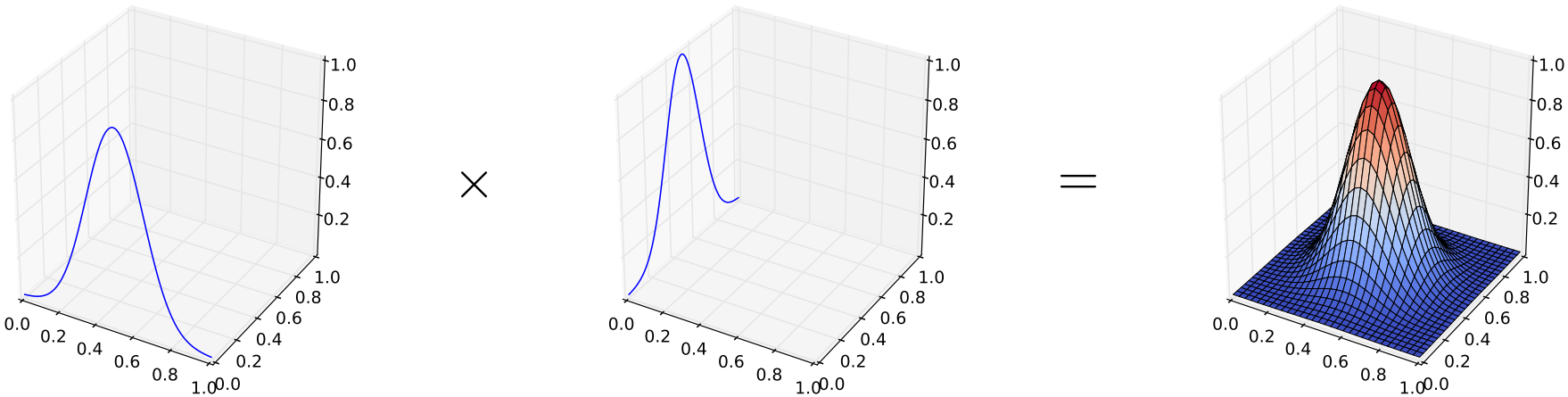
Property

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

is valid covariance structure.

Example

We multiply 2 squared exponential kernel



Question: What kernel do we end up with?

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:

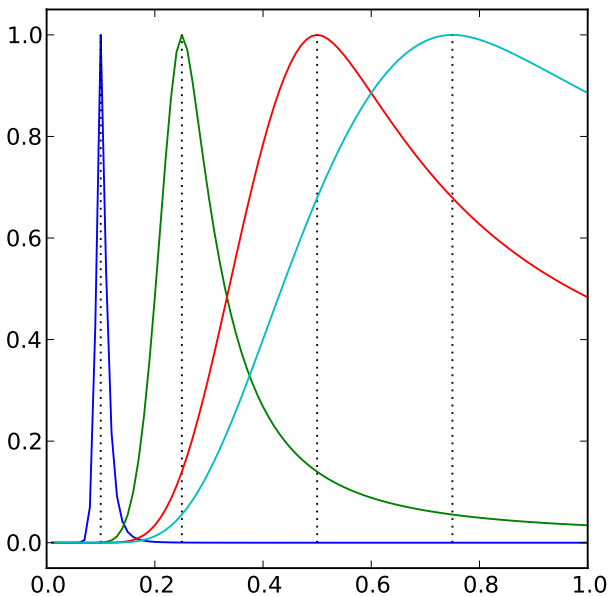
- k corresponds to the covariance of $Z(x) = Z_1(f(x))$
- This can be seen as a (nonlinear) rescaling of the input space

Example

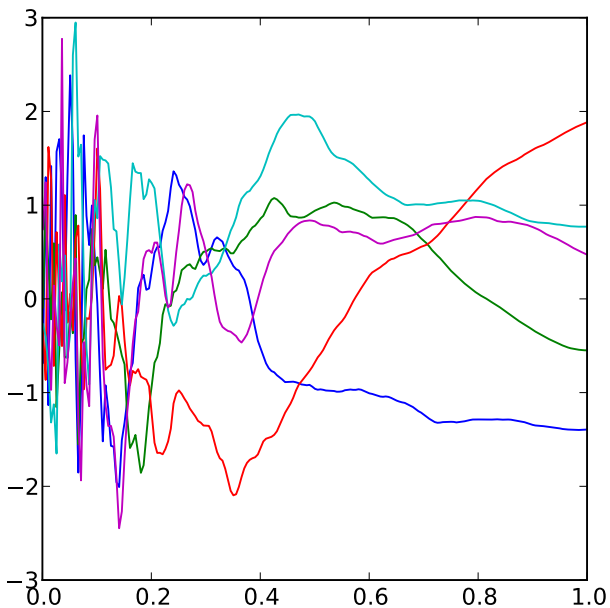
We consider $f(x) = \frac{1}{x}$ and a Matérn 3/2 kernel $k_1(x, y) = (1 + |x - y|)e^{-|x-y|}$.

We obtain:

Kernel



Sample paths



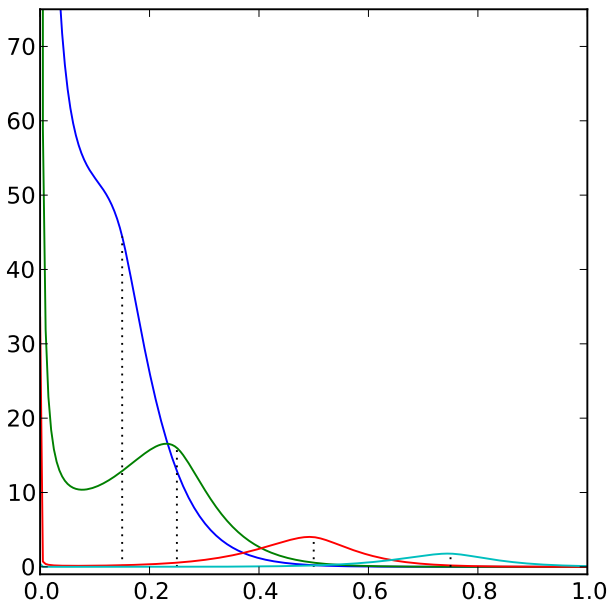
All these transformations can be combined!

Example

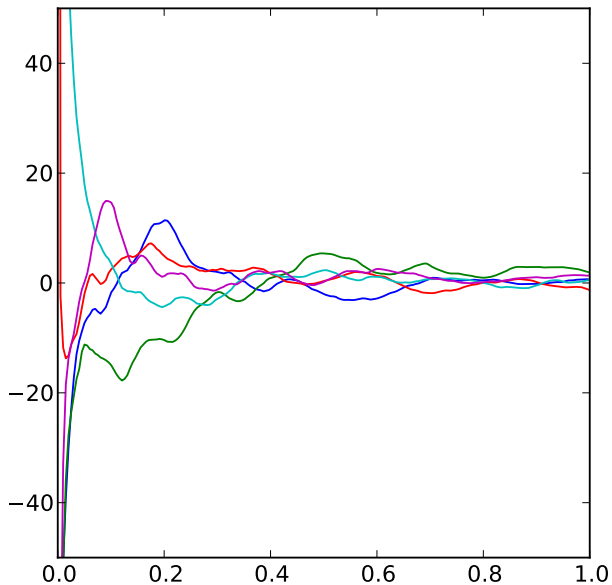
$k(x, y) = f(x)f(y)k_1(x, y)$ is a valid kernel.

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

Kernel



Sample paths



Effect of a linear operator

Effect of a linear operator

Property

Let L be a linear operator that can be applied to samples of a process Z (such that [...]). Then

$$k(x, y) = L_x(L_y(k(x, y)))$$

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:

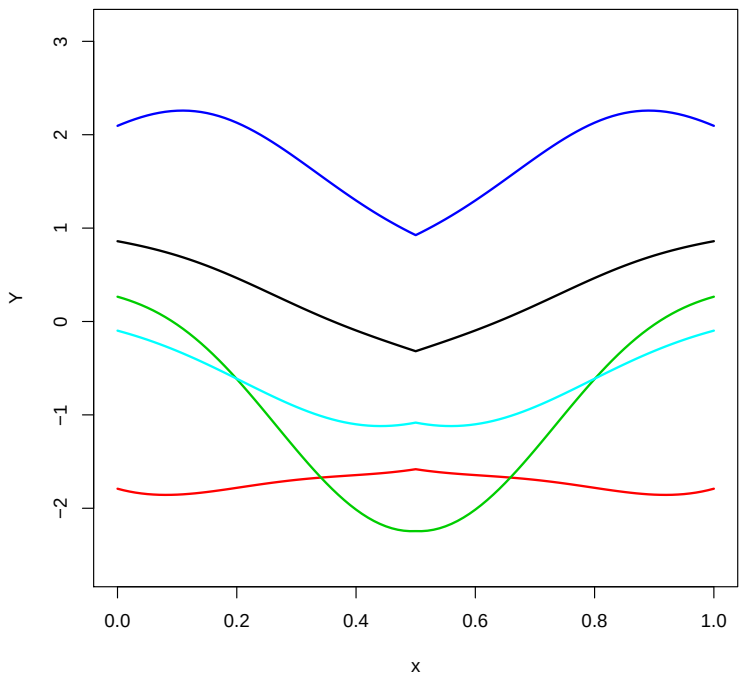
$$L_1 : f(x) \to \begin{cases} f(x) & x < 0.5 \\ f(1-x) & x \geq 0.5 \end{cases}$$

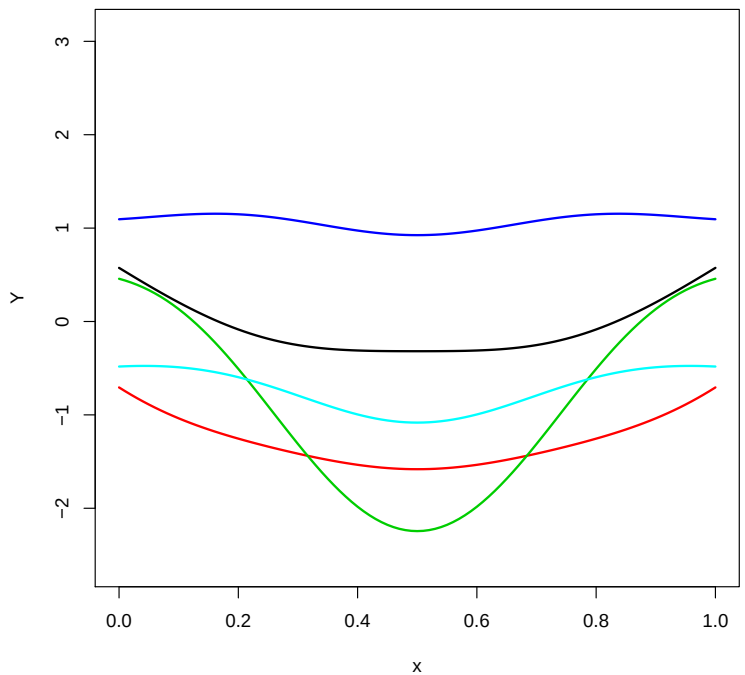
$$L_2 : f(x) \to \frac{f(x) + f(1-x)}{2}.$$

Effect of a linear operator: example [Ginsbourger 2013]

Examples of associated sample paths are

$$k_1 = L_1(L_1(k)) \qquad k_2 = L_2(L_2(k))$$

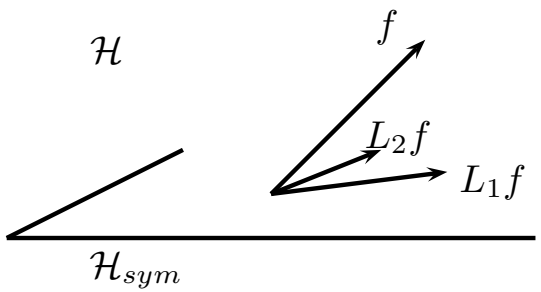




The differentiability is not always respected!

Effect of a linear operator

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



What about the optimal projection?

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

Conclusion

We have seen that

- Kriging model do not necessarily interpolate, but they can if we want them to!
- If we have information about a trend, it is possible to incorporate it in models.
- Kernels encode the prior belief on the function to approximate.
  - They should be chosen accordingly
- It is sometimes possible to design kernels tailored to the problem at hand.