Introduction to Gaussian process models

Nicolas Durrande (Ecole des Mines de St Etienne)

durrande@emse.fr

UTC – AVENUE Roberval, the 20\textsuperscript{th} of May 2014
Outline:

1. Introduction to GP models
2. What is a kernel?
3. Design of Experiments
4. Sensitivity analysis
5. Co-Kriging
Introduction
We assume we have observed a function $f$ for a limited number of time points $x_1, \ldots, x_n$:

The observations are denoted by $f_i = f(x_i)$ (or $F = f(X)$).
Since \( f \) is unknown, we make the general assumption that it is to the sample path of a Gaussian process \( Y \):

\[
Y \text{ is characterised by its mean (here 0) and covariance function (or kernel) } k(x, y) = \text{cov}(Y(x), Y(y)).
\]
We can look at the sample paths of $Y$ that interpolate the data points:
The conditional distribution is still Gaussian. It has mean and variance

\[ m(x) = \mathbb{E}(Y(x)|Y(X)=F) = k(x, X)k(X, X)^{-1}F \]
\[ v(x) = \text{var}(Y(x)|Y(X)=F) = k(x, x) - k(x, X)^tk(X, X)^{-1}k(x, X) \]

where \( k \) is the kernel: \( k(x, y) = \text{cov}(Y(x), Y(y)) \).

It can be represented as a mean function with confidence intervals.
In order to build $m$ we can take any kernel off the shelf:

- **white noise**: $k(x, y) = \delta_{x,y}$
- **bias**: $k(x, y) = 1$
- **linear**: $k(x, y) = xy$
- **exponential**: $k(x, y) = \exp(-|x - y|)$
- **Brownian**: $k(x, y) = \min(x, y)$
- **Gaussian**: $k(x, y) = \exp(-(x - y)^2)$
- **Matérn 3/2**: $k(x, y) = (1 + |x - y|) \times \exp(-|x - y|)$
- **sinc**: $k(x, y) = \frac{\sin(|x - y|)}{|x - y|}$
GPy.kern.rbf

GPy.kern.exponential

GPy.kern.Matern32

GPy.kern.Matern52

GPy.kern.Brownian

GPy.kern.bias

GPy.kern.linear

GPy.kern.spline

GPy.kern.periodic_exponential

GPy.kern.periodic_Matern32

GPy.kern.periodic_Matern52

GPy.kern.white
Changing the kernel has a huge impact on the model:

squared exponential kernel: \( k(x, y) = \sigma^2 \exp\left(\frac{-(x-y)^2}{\theta^2}\right) \)

exponential kernel: \( k(x, y) = \sigma^2 \exp\left(\frac{-|x-y|}{\theta}\right) \)
This is because changing the kernel means changing the prior on $f$. 

Squared exponential kernel: 
\[ k(x, y) = \sigma^2 \exp \left( -\frac{(x-y)^2}{\theta^2} \right) \]

Exponential kernel: 
\[ k(x, y) = \sigma^2 \exp \left( -\frac{|x-y|}{\theta} \right) \]

---

N. Durrande (durrande@emse.fr)  Intro to GP models  20th of May 2014
What is a kernel?
A kernel is a **positive semi-definite function**:

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

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

It has to be chosen accordingly to the prior believe on the function to approximate:

- What is the regularity of the phenomenon?
- Is it stationary?
- ...
If we have some knowledge about the behaviour of $f$, can we design a kernel accordingly?

Designing appropriate kernels results in great improvements of the models. Basic tools such as

- Making new from old
- Linear operators

ensure to preserve the positive definiteness.
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!
What is a kernel?

What happen if we sum both kernels?

\[ k(x, y) = k_{rbf_1}(x, y) + k_{rbf_2}(x, y) \]
What is a kernel?

What happen if we sum both kernels?

\[ k(x, y) = k_{rbf_1}(x, y) + k_{rbf_2}(x, y) \]

The model is drastically improved!
What is a kernel?

We can try the following kernel:

\[ k(x, y) = \sigma_0^2 x^2 y^2 + k_{rbf_1}(x, y) + k_{rbf_2}(x, y) + k_{per}(x, y) \]
What is a kernel?

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.
Design of Experiment
Depending on the context, it is sometimes possible to choose the data points $X$ where the function will be evaluated. This set of points is called the Design of Experiments (DoE).

We often look for about space filling design in order to minimize the (integrated) prediction variance.
Example

Space filling designs:

- full factorial
- LHS
minimizing the prediction variance

The variance of a GP regression model depends on the DoE and the kernel parameter but it does not depend on the vector of observations:

\[ \nu(x) = \text{var} (Y(x)| Y(X)=F) = k(x, x) - k(x, X)^t k(X, X)^{-1} k(x, X) \]

With known parameters, it is thus possible to optimize the DoE to minimize the prediction variance.
DoE and Parameter estimation

Parameter estimation is influenced by the DoE.

\[ X = (0, 1, 2, 3, 4, ..., 9) \]

\[ X = (0, 0.1, 0.2, 0.4, 0.8, 1.6, ...) \]

Estimation of the variance
DoE and Parameter estimation

Parameter estimation is influenced by the DoE.

$X = (0, 1, 2, 3, 4, \ldots, 9)$

$X = (0, 0.1, 0.2, 0.4, 0.8, 1.6, \ldots)$

Estimation of the lengthscale
Sensitivity analysis
Sensitivity analysis

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

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

where $\int f(x_i) dx_i = 0$ if $i \in I$.

The expressions of the $f_i$ are:

$$f_0 = \int f(x) dx$$

$$f_i(x_i) = \int f(x) dx_{-i} - f_0$$

$$f_{i,j}(x_i, x_j) = \int f(x) dx_{-ij} - f_i(x_i) - f_j(x_j) + f_0$$
Can we obtain a similar decomposition for the model?

The following kernel will be of particular interest:

\[
k(x, y) = \prod_{i=1}^{d} \left( 1 + k_0(x_i, y_i) \right)
\]

\[
= 1 + \sum_{i=1}^{d} k_0(x_i, y_i) + \sum_{i<j} k_0(x_i, y_i)k_0(x_j, y_j) + \cdots + \prod_{i=1}^{d} k_0(x_i, y_i)
\]

where

\[
k_0(x, y) = k(x, y) - \frac{\int k(x, s)ds \int k(y, s)ds}{\int \int k(s, t)dsdt}
\]
A decomposition of the best predictor is naturally associated to those kernels.

**Example:** we have in 2D $k = 1 + k_1 + k_2 + k_1 k_2$ so the best predictor can be written as

$$m(x) = (1 + k(x_1) + k(x_2) + k(x_1)k(x_2))^t K^{-1} F$$

$$= m_0 + m_1(x_1) + m_2(x_2) + m_{12}(x)$$

This decomposition satisfies the conditions of the ANOVA representation:

$$\int_{D_i} m_i(x_i) dx_i = 0$$
Let us consider the random test function $f : [0, 1]^{10} \rightarrow \mathbb{R}$:

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

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 the Kriging predictor $m$ based on $\prod (1 + k_0)$

As $m$ is a function of 10 variables, the model can not easily be represented: it is usually considered as a “blackbox”. However, the structure of the kernel allows to split $m$ in sub-models.
The univariate sub-models are:

\[
\begin{align*}
W(x) &= 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \mathcal{N}(0, 1)
\end{align*}
\]
Co-Kriging
In some cases, we may observe not only the function of interest $f$ but also another related function $f_c$ that can provide some insights on $f$.

**Example**

- $f(x)$ is the observed value and $f_c(x)$ is a computed approximation
- $f(x)$ is a high fidelity computer code and $f_c(x)$ is a low fidelity one

Can we use this additional information?
In some cases, we may observe not only the function of interest $f$ but also another related function $f_c$ that can provide some insights on $f$.

**Example**

- $f(x)$ is the observed value and $f_c(x)$ is a computed approximation
- $f(x)$ is a high fidelity computer code and $f_c(x)$ is a low fidelity one

Can we use this additional information? Yes!
In some cases, we may observe not only the function of interest $f$ but also another related function $f_c$ that can provide some insights on $f$.

**Example**

- $f(x)$ is the observed value and $f_c(x)$ is a computed approximation
- $f(x)$ is a high fidelity computer code and $f_c(x)$ is a low fidelity one

Can we use this additional information? 
Yes!

What should be adapted?
In some cases, we may observe not only the function of interest $f$ but also another related function $f_c$ that can provide some insights on $f$.

**Example**

- $f(x)$ is the observed value and $f_c(x)$ is a computed approximation
- $f(x)$ is a high fidelity computer code and $f_c(x)$ is a low fidelity one

Can we use this additional information?  
Yes!

What should be adapted?  
Nothing!
Let \( k_{cc} \) be the kernel of \( Y_c \) and \( k_c(x, y) = \text{cov}(Y(x), Y_c(y)) \) be the cross covariance.

**The Co-Kriging equations are:**

\[
\begin{align*}
m(x) &= E(Y(x)|Y(X)=F, Y_c(X_c)=F_c) = k(x)K^{-1}F \\
v(x) &= \text{var}(Y(x)|Y(X)=F, Y_c(X_c)=F_c) = k(x,x) - k(x)K^{-1}k(x)^t
\end{align*}
\]

where

\[
k(x) = (k(x, X), k_c(x, X_c))
\]

\[
K = \begin{pmatrix}
  k(X, X) & k_c(X, X_c) \\
  k_c(X, X_c)^t & k_{cc}(X_c, X_c)
\end{pmatrix}
\]

The only issue here can be to get the cross covariance function \( k_c \).
Example

3 high fidelity observations and 5 low fidelity ones:
Conclusion
Conclusion

We have seen that

- GP regression is a very powerful framework for dealing with limited observations of a phenomenon.
- The choice of the kernel has to reflect the prior belief about the function to approximate.

Kriging can be used for sensitivity analysis, multi-fidelity outputs, variable selection...

Various implementations are already available including:

- Python: GPy
- R: DiceKriging
- Matlab: GPmat, GPML
C. E. Rasmussen and C. Williams

A. Berlinet and C. Thomas-Agnan

N. Durrande, D. Ginsbourger, O. Roustant, L. Carraro
ANOVA kernels and RKHS of zero mean functions for model-based sensitivity analysis, JMA 2013.

L. Le Gratiet
Multi-fidelity Gaussian process regression for computer experiments, PhD thesis 2013.