Kernel ANOVA Decomposition for Gaussian process modeling

N. Durrande [1], D. Ginsbourger [2], O. Roustant [1], *L. Carraro* [3]

MASCOT NUM 2011 workshop

Villard de Lans, the 23th of March

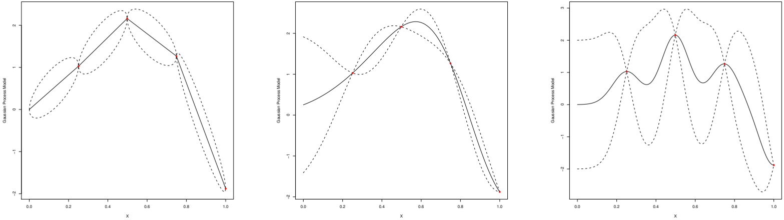
1. CROCUS - Ecole des Mines de St Etienne
2. Institute of Mathematical Statistics and Actuarial Science - University of Berne
3. Telecom St Etienne

Gaussian process models

Let $f : D \subset R^d \to R$ be a function which value is known on a DoE $X = (x^1, \ldots, x^n)$.

The kriging model relies on the choice of the kernel K

$$m(x) = k(x)^T \mathrm{K}^{-1} Y \quad \text{and} \quad v(x) = K(x,x) - k(x)^T \mathrm{K}^{-1} k(x)$$



Gaussian process models

When the dimension of the input space increases, the kriging model really becomes a black-box.

$$m(x) = k(x)^T \mathrm{K}^{-1} Y$$

Major drawbacks for usual kernels :

- The models cannot easily be interpreted.
  - *Without computation, what is the effect of x^1 on $m(x)$?*
- The importance of the variables x^i is supposed to be similar.
  - *What if the variance is not the same in each direction ?*

outline

We present here a method inspired from the ANOVA decomposition that allows to tackle those issues.

The talk is organized as follow :

- **Kernel ANOVA Decomposition (KAD)**
- **Selection of relevant terms :** the HKL method.
- **Example of application :** The MARTHE benchmark.

Kernel ANOVA Decomposition

Any square integrable function $f : D \to \mathbb{R}$ may be written

ANOVA Decomposition

$$f(x) = f_0 + \sum_{i=1}^{d} f_i(x_i) + \sum_{1 \leq i < j \leq d} f_{i,j}(x_i, x_j) + \cdots + f_{1,\ldots,d}(x_1, \ldots, x_d)$$

where :

- Any two terms of the decomposition are $\perp$ in $L^2(D)$,
- the integral of $f_{\alpha_1,\ldots,\alpha_p}(x)$ with respect to any x_{α_i} is null.

Kernel ANOVA Decomposition

For $D \subset \mathbb{R}$, the space $L^2(D)$ may be decomposed as follows :

$$f(x) = \int_D f(s)\mathrm{d}s + \left(f(x) - \int_D f(s)\mathrm{d}s\right)$$
$$L^2(D) = \mathcal{L}_0 \overset{\perp}{\oplus} \mathcal{L}_1$$

where $\mathcal{L}_0$ is the space of the functions equal to a constant and $\mathcal{L}_1$ the space of function with zero mean.

For $\mathcal{D} = D_1 \times \cdots \times D_d \subset \mathbb{R}^d$, we obtain

$$L^2(\mathcal{D}) = \prod_{i=1}^{d} L^2(D_i) = \prod_{i=1}^{d} \left(\mathcal{L}_0^i \overset{\perp}{\oplus} \mathcal{L}_1^i\right) = \sum_{I \in \{0,1\}^d} \mathcal{L}_I$$

Kernel ANOVA Decomposition

Similarly, let $\mathcal{H}$ be a one-dimensional RKHS with kernel k.
We call $\mathcal{H}_1$ the subspace of $\mathcal{H}$ with zero mean functions :

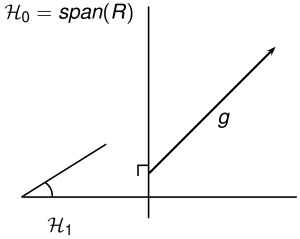
$$g \in \mathcal{H}_1 \Leftrightarrow \int_D g(s)\mathrm{d}s = 0$$

The Riesz theorem gives

$$\exists ! R \in \mathcal{H} \text{ such that } \forall g \in \mathcal{H}, \int_D g(s)\mathrm{d}s = \langle R, g \rangle_{\mathcal{H}}$$

We have an orthogonal decomposition of $\mathcal{H}$:

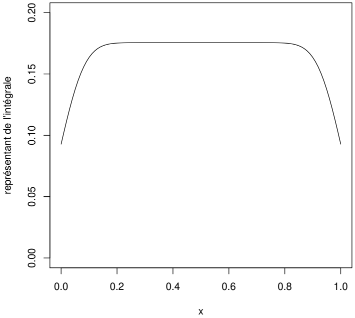
$$\mathcal{H} = \mathcal{H}_0 \overset{\perp}{\oplus} \mathcal{H}_1$$



Kernel ANOVA Decomposition

Using the reproducing property of k, we get the expression of $R(x)$:

$$R(x) = \langle R, k(x,.)\rangle_{\mathcal{H}} = \int_D k(x,s)\mathrm{d}s$$



proposed ANOVA-like decomposition

Let k_0 and k_1 be the reproducing kernels of $\mathcal{H}_0$ and $\mathcal{H}_1$.
As $\mathcal{H} = \mathcal{H}_0 + \mathcal{H}_1$, we have :

$$k(x, y) = k_0(x, y) + k_1(x, y)$$

Using the orthogonal projection on $\mathcal{H}_0$ one can calculate :

$$k_0(x, y) = \frac{\int_D k(x, s)\mathrm{d}s \int_D k(y, s)\mathrm{d}s}{\int_{D\times D} k(s, t)\mathrm{d}s\mathrm{d}t}$$

$$k_1(x, y) = k(x, y) - \frac{\int_D k(x, s)\mathrm{d}s \int_D k(y, s)\mathrm{d}s}{\int_{D\times D} k(s, t)\mathrm{d}s\mathrm{d}t}$$

proposed ANOVA-like decomposition

Probabilistic interpretation

Let Z_0 and Z_1 be centered GP with kernels k_0 and k_1

ANOVA Decomposition for GP

$$Z(x) = Z_0(x) + Z_1(x)$$

with

- Z_0 and Z_1 independent
- $\int_D Z_1(x)\mathrm{d}x = 0$ (with proba. 1)

proposed ANOVA-like decomposition

Probabilistic interpretation

Z_0 and Z_1 may also be defined as :

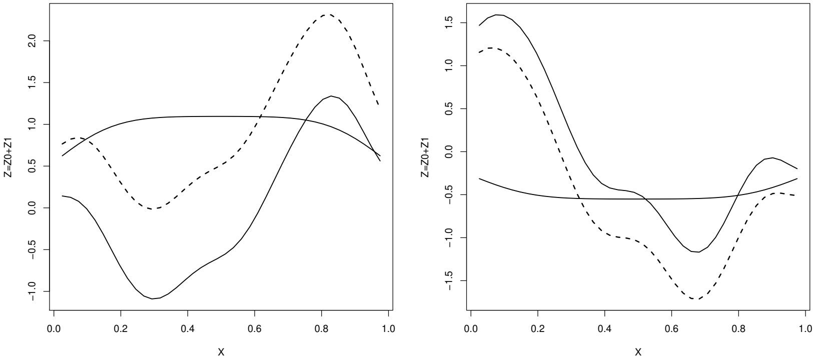
$$Z_0(x) = \mathrm{E}\left[Z(x) \middle| \int_D Z(s) \mathrm{d}s \right] = \frac{\displaystyle\int_D k(x,s) \mathrm{d}s}{\displaystyle\int_{D \times D} k(s,t) \mathrm{d}s \mathrm{d}t} \int_D Z(s) \mathrm{d}s$$

$$Z_1(x) = Z(x) - Z_0(x)$$

Then Z_0 and Z_1 have kernel k_0 and k_1.

proposed ANOVA-like decomposition

Given Z, we can decompose any path $Z(\omega)$ as $Z_0(\omega) + Z_1(\omega)$



Reciprocally, given K_0 and K_1 we can build paths of Z by summing $Z_0(\omega)$ and $Z_1(\omega)$.

proposed ANOVA-like decomposition

What happens for the multi-dimensional case ?

If K is a tensor product kernel, the generalization is straightforward :

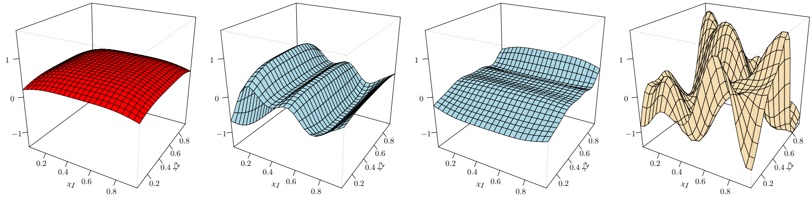
$$\begin{aligned} K &= k \times k = (k_0 + k_1) \times (k_0 + k_1) \\ &= k_0 k_0 + k_1 k_0 + k_0 k_1 + k_1 k_1 \\ &= K_{00} + K_{10} + K_{01} + K_{11} \end{aligned}$$

Or similarly

$$\begin{aligned} \mathcal{H}_K &= \mathcal{H} \otimes \mathcal{H} \\ &= (\mathcal{H}_0 \overset{\perp}{\oplus} \mathcal{H}_1) \otimes (\mathcal{H}_0 \overset{\perp}{\oplus} \mathcal{H}_1) \\ &= \mathcal{H}_0 \otimes \mathcal{H}_0 \ \overset{\perp}{\oplus} \ \mathcal{H}_1 \otimes \mathcal{H}_0 \ \overset{\perp}{\oplus} \ \mathcal{H}_0 \otimes \mathcal{H}_1 \ \overset{\perp}{\oplus} \ \mathcal{H}_1 \otimes \mathcal{H}_1 \end{aligned}$$

proposed ANOVA-like decomposition

We use those kernels to simulates paths of Z_{00}, Z_{10}, Z_{01} and Z_{11} :



As previously, the paths have original properties.

KAD $\neq$ ANOVA kernels

Link with usual ANOVA kernels [4] :

$$K_{ANOVA}(x, y) = \prod_i (1 + k(x_i, y_i))$$

For this decomposition, we have

- $\mathcal{H}_0$ is a space of constant functions.
- $\mathcal{H}_1$ **is not** the space of zero-mean functions.
- We do not have anymore $\mathcal{H}_0 \perp \mathcal{H}_1$

4. Stitson et Al, Support vector regression with ANOVA decomposition kernels. Technical report, Royal Holloway, University of London, 1997.

Kernel ANOVA Decomposition

This decomposition may be used for many tasks :

- visualize main effects without computation.
- modify the weight of the sub-kernels :

$$K^* = \lambda_{00}K_{00} + \lambda_{10}K_{10} + \lambda_{01}K_{01} + \lambda_{11}K_{11}$$

or built sparse models

$$K^* = K_{00} + \cancel{K_{10}} + K_{01} + \cancel{K_{11}}$$

We will now consider those two points on two test functions.

Application 1 : interpretation

We consider a test function [5] with observation's noise $\mathcal{N}(0, 1)$:

$$f : [0,\ 1]^{10} \to \mathbb{R}$$
$$x \mapsto 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5$$

The steps for approximating f with a GP model are :

1. Learn f on a DoE (here LHS maximin with 180 points)
2. estimate the kernel parameters ψ (MLE),
3. build the kriging mean predictor $\hat{f}$ based on K^ψ

As $\hat{f}$ is a function of 10 variables, the model can not easily be represented : it is usually considered as a black-box.

5. S.R. Gunn and J.S. Kandola. Structural modelling with sparse kernels. Machine learning, 2002

Application 1 : interpretation

with KAD, $\hat{f}$ can be written as the sum of sub-models

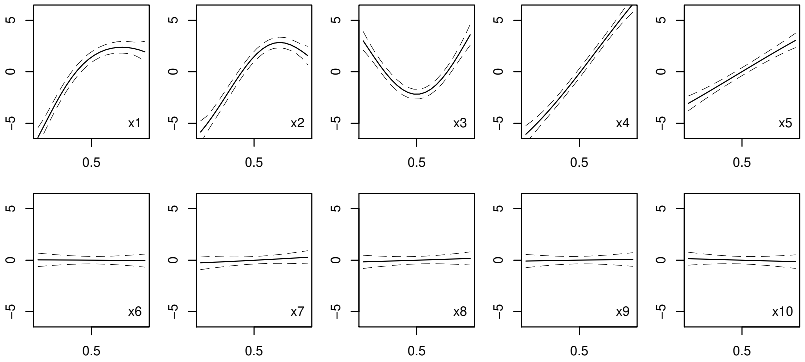
$$K^{\psi}(x,y) = \sum_{I \in \{0,1\}^d} K_I(x,y)$$

$$\Downarrow$$

$$\begin{aligned}
\hat{f}(x) &= k(x)^T(\mathrm{K} + \tau^2 \mathrm{Id})^{-1} Y \\
&= \left(\sum_{I \in \{0,1\}^d} k_I(x) \right)^T (\mathrm{K} + \tau^2 \mathrm{Id})^{-1} Y \\
&= \sum_{I \in \{0,1\}^d} \left(k_I(x)^T (\mathrm{K} + \tau^2 \mathrm{Id})^{-1} Y \right) = \sum_{I \in \{0,1\}^d} \hat{f}_I(x)
\end{aligned}$$

Application 1 : interpretation

The univariate sub-models are :



$$\left(\text{we had } f(x) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 \right)$$

Application 2 : HKL

In order to

- Construct parsimonious models,
- Change the weights of the sub-kernels,

we will use a method called **Hierarchical Kernel Learning** (HKL) developed by F. Bach in 2009.

Application 2 : HKL

Hierarchical kernel Learning

Given a set of kernel $\{K_1, \ldots, K_n\}$ the point is to select a limited number of them adapted to the data :

$$\{K_1, \ldots, K_n\} \rightarrow K^* = \lambda_1 K_1 + \cancel{\lambda_2 K_2} + \lambda_3 K_3 + \cdots + \cancel{\lambda_n K_n}$$

Like other methods (COSSO, SUPANOVA), the sparsity and the coefficients are obtained by minimizing a trade off between 2 norms :

$$criterion = \text{``}\ ||f - \hat{f}||_2 + c||\hat{f}||_1\ \text{''}$$

Application 2 : HKL

Let us combine KAD and HKL to model the test function f.

The steps for modeling f are :

1. Construct a DoE X, and calculate the response $Y = f(X)$
2. Estimate the kernels parameter ψ (MLE),
3. Decompose K_ψ using KAD.
4. Apply HKL.
5. Get the final GP model.

Application 2 : HKL

Here, the total number of kernels is $\mathbf{2^d = 1024}$.

As $f(x) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon(x)$, we could expect HKL to find **7 active kernels.**

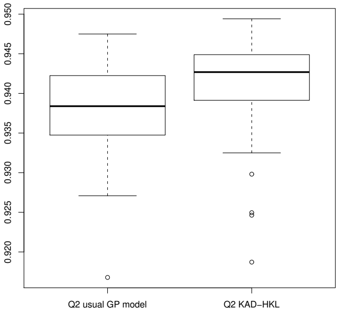
The algorithm gives 84 active kernels but the weight associated to the unexpected ones is around 0.

To evaluate the quality of the model, we compare it to a usual GP on 2000 test points. We compute

$$Q_2 = 1 - \frac{\sum(\hat{f}_i - f_i)^2}{\sum(f_i - \bar{f})^2}$$

Application 2 : HKL

Varying X, we finally obtain :



On this example, KAD-HKL performs significantly better.

The Marthe case study

The MARTHE case study is part of the GDR-mascotnum benchmark.

Objective : estimation of an environmental impact

- Radioactive waste storage on a Russian site from 1943 to 1974
- Upper groundwater contamination in ^{90}Sr.

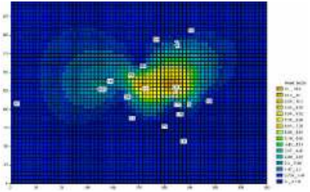
The aim is to model the evolution of the radioactive plume.

The Marthe case study

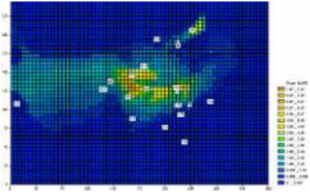
The MARTHE computer code has

- 20 input variables (7 permeabilities, 1 porosity, ...)
- 10 output variables (locations to predict the ^{90}Sr concentration)

We know the concentration for 2002, we want to predict it for 2010.



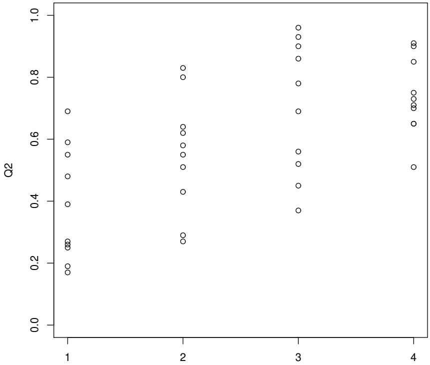
$\xrightarrow{\textit{MARTHE}}$



The Marthe case study

The design is composed of 300 points. 250 are used for training and 50 for external validation.

Results



1. Regression
2. Boosting Trees
3. Marrel and Iooss
4. KAD-HKL

Conclusion

Advantages of the proposed Kernel Anova Decomposition

- Interpretation of High dimensional GP models
- Allows to set various variance parameters
- Allows to split multi-dimensional problems into low-dimensional ones
- Well designed for HKL

Applications

- Model accuracy improvement
- Calculation of Sobol indices.
- Can be coupled with any kriging software

Conclusion

Thank you for your attention

F. Bach, *High-Dimensional Non-Linear Variable Selection through Hierarchical Kernel Learning*, hal-00413473, 2009.
B. Iooss and A. Marrel, *Benchmark of GdR MASCOT NUM – Données MARTHE*, 2008.