A new class of covariance kernels accounting for non-additivity in high-dimensional kriging

Nicolas Lenz
University of Bern
Supervised by D. Ginsbourger, D. Schuhmacher, L. Dümbgen

Nice, July 4, 2013

General setting

Point of departure

N. Durrande, D. Ginsbourger and O. Roustant (2012)
Additive covariance kernels for high-dimensional Gaussian process modeling.
Ann. Fac. Sci. Toulouse 21 481-499.

General setting

Point of departure

N. Durrande, D. Ginsbourger and O. Roustant (2012)
Additive covariance kernels for high-dimensional Gaussian process modeling.
Ann. Fac. Sci. Toulouse 21 481-499.

We consider a GRF $(Z_x)_{x \in D}$ over the domain $D = [0,1]^d$, $d \in \mathbb{N}$. We assume that expectation and covariance kernel exist and call them respectively

$$m(x) = \mathbb{E}[Z_x]$$
$$k(x,y) = \mathrm{Cov}(Z_x, Z_y)$$

Under mild conditions the trajectories of Z are $\mathcal{L}^2$

Considerations in $\mathcal{L}^2$

$f \in \mathcal{L}^2$ can be decomposed

$$f = f_{\mathcal{C}} + f_{\mathcal{U}_1} + \ldots + f_{\mathcal{U}_d}$$

Considerations in $\mathcal{L}^2$

$f \in \mathcal{L}^2$ can be decomposed

$$f = f_{\mathcal{C}} + f_{\mathcal{U}_1} + \ldots + f_{\mathcal{U}_d} + f_{\mathcal{O}}$$

Considerations in $\mathcal{L}^2$

$f \in \mathcal{L}^2$ can be decomposed

$$f = f_{\mathcal{C}} + f_{\mathcal{U}_1} + \ldots + f_{\mathcal{U}_d} + f_{\mathcal{O}}$$

$$\begin{aligned}
f_{\mathcal{C}} &= \int_D f \, d\mu \cdot \mathbf{1}_D \\
f_{\mathcal{U}_i} &= \int_{D_{-i}} f - f_{\mathcal{C}} \, d\mu_{-i} \cdot \mathbf{1}_{D_{-i}} \\
f_{\mathcal{A}} &= f_{\mathcal{C}} + \sum_{i=1}^{d} f_{\mathcal{U}_i} \\
f_{\mathcal{O}} &= f - f_{\mathcal{A}}
\end{aligned}$$

Considerations in $\mathcal{L}^2$

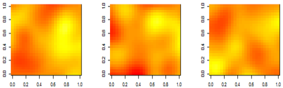
$f \in \mathcal{L}^2$ can be decomposed

$$f = f_{\mathcal{C}} + f_{\mathcal{U}_1} + \ldots + f_{\mathcal{U}_d} + f_{\mathcal{O}}$$

$$\begin{aligned}
f_{\mathcal{C}} &= \int_D f \, d\mu \cdot \mathbf{1}_D &&=: \pi_{\mathcal{C}} f \\
f_{\mathcal{U}_i} &= \int_{D_{-i}} f - f_{\mathcal{C}} \, d\mu_{-i} \cdot \mathbf{1}_{D_{-i}} &&=: \pi_{\mathcal{U}_i} f \\
f_{\mathcal{A}} &= f_{\mathcal{C}} + \sum_{i=1}^{d} f_{\mathcal{U}_i} &&=: \pi_{\mathcal{A}} f \\
f_{\mathcal{O}} &= f - f_{\mathcal{A}} &&=: \pi_{\mathcal{O}} f
\end{aligned}$$

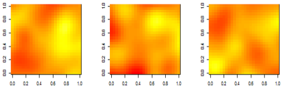
Projecting a random field

Realizations $Z(\omega)$ of a GRF, generated with an isotropic kernel $k(x,y) = \sigma^2 \cdot e^{-(\frac{\|x-y\|}{\theta})^2}$

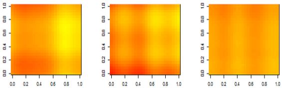


Projecting a random field

Realizations $Z(\omega)$ of a GRF, generated with an isotropic kernel $k(x,y) = \sigma^2 \cdot e^{-\left(\frac{\|x-y\|}{\theta}\right)^2}$

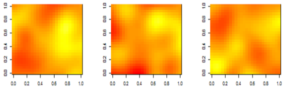


$\pi_{\mathcal{A}} Z(\omega)$

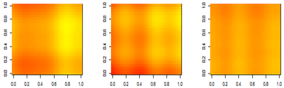


Projecting a random field

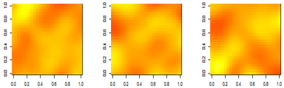
Realizations $Z(\omega)$ of a GRF, generated with an isotropic kernel $k(x, y) = \sigma^2 \cdot e^{-(\frac{\|x-y\|}{\theta})^2}$



$\pi_{\mathcal{A}} Z(\omega)$

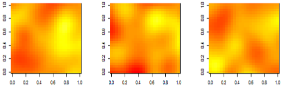


$\pi_{\mathcal{O}} Z(\omega)$

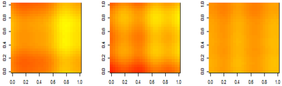


Projecting a random field

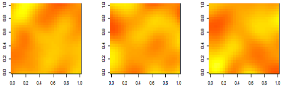
Realizations $Z(\omega)$ of a GRF, generated with an isotropic kernel $k(x, y) = \sigma^2 \cdot e^{-\left(\frac{\|x-y\|}{\theta}\right)^2}$



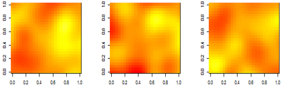
$\pi_{\mathcal{A}} Z(\omega)$



$\pi_{\mathcal{O}} Z(\omega)$



$\pi_{\mathcal{A}} Z(\omega) + \pi_{\mathcal{O}} Z(\omega)$



"Double" decomposition of a kernel

Let $\mathcal{P}$ be a finite family of projections such that

$$\mathrm{Id}_{\mathcal{L}^2} = \sum_{\pi \in \mathcal{P}} \pi$$

"Double" decomposition of a kernel

Let $\mathcal{P}$ be a finite family of projections such that

$$\mathrm{Id}_{\mathcal{L}^2} = \sum_{\pi \in \mathcal{P}} \pi$$

With these projections we can equally decompose a kernel

$$\mathrm{Id}_{\mathcal{L}^2 \times \mathcal{L}^2} = \Big(\sum_{\pi \in \mathcal{P}} \pi\Big) \otimes \Big(\sum_{\tilde{\pi} \in \mathcal{P}} \tilde{\pi}\Big) = \sum_{\pi \in \mathcal{P}} \sum_{\tilde{\pi} \in \mathcal{P}} (\pi \otimes \tilde{\pi})$$

"Double" decomposition of a kernel

Let $\mathcal{P}$ be a finite family of projections such that

$$\mathrm{Id}_{\mathcal{L}^2} = \sum_{\pi \in \mathcal{P}} \pi$$

With these projections we can equally decompose a kernel

$$\mathrm{Id}_{\mathcal{L}^2 \times \mathcal{L}^2} = (\sum_{\pi \in \mathcal{P}} \pi) \otimes (\sum_{\tilde{\pi} \in \mathcal{P}} \tilde{\pi}) = \sum_{\pi \in \mathcal{P}} \sum_{\tilde{\pi} \in \mathcal{P}} (\pi \otimes \tilde{\pi})$$

$$\begin{aligned} k(x,y) = \mathrm{Cov}(Z_x, Z_y) &= \mathrm{Cov}(\sum_{\pi \in \mathcal{P}} \pi\, Z_x, \sum_{\tilde{\pi} \in \mathcal{P}} \tilde{\pi}\, Z_y) \\ &= \sum_{\pi \in \mathcal{P}} \sum_{\tilde{\pi} \in \mathcal{P}} \mathrm{Cov}(\pi\, Z_x, \tilde{\pi}\, Z_y) = \left(\sum_{\pi \in \mathcal{P}} \sum_{\tilde{\pi} \in \mathcal{P}} (\pi \otimes \tilde{\pi}) k \right)(x,y) \end{aligned}$$

Schematic representation of kernels

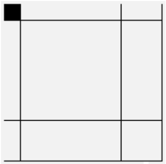
Applying $\mathcal{P} = \{\pi_{\mathcal{C}}, \pi_{\mathcal{U}_1}, \ldots, \pi_{\mathcal{U}_d}, \pi_{\mathcal{O}}\}$ to a kernel gives us a decomposition into $(d+2)^2$ parts.

We identify a projected kernel figuratively by a $(d+2) \times (d+2)$ matrix

Schematic representation of kernels

Applying $\mathcal{P} = \{\pi_{\mathcal{C}}, \pi_{\mathcal{U}_1}, \dots, \pi_{\mathcal{U}_d}, \pi_{\mathcal{O}}\}$ to a kernel gives us a decomposition into $(d+2)^2$ parts.

We identify a projected kernel figuratively by a $(d+2) \times (d+2)$ matrix, e.g.

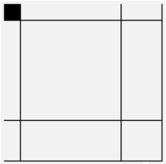


constant

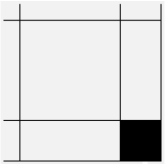
Schematic representation of kernels

Applying $\mathcal{P} = \{\pi_{\mathcal{C}}, \pi_{\mathcal{U}_1}, \ldots, \pi_{\mathcal{U}_d}, \pi_{\mathcal{O}}\}$ to a kernel gives us a decomposition into $(d+2)^2$ parts.

We identify a projected kernel figuratively by a $(d+2) \times (d+2)$ matrix, e.g.



constant

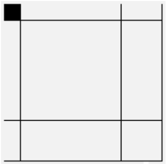


ortho-add.

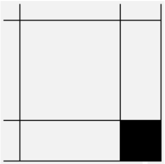
Schematic representation of kernels

Applying $\mathcal{P} = \{\pi_\mathcal{C}, \pi_{\mathcal{U}_1}, \ldots, \pi_{\mathcal{U}_d}, \pi_\mathcal{O}\}$ to a kernel gives us a decomposition into $(d+2)^2$ parts.

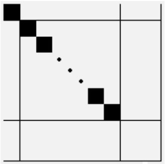
We identify a projected kernel figuratively by a $(d+2) \times (d+2)$ matrix, e.g.



constant



ortho-add.

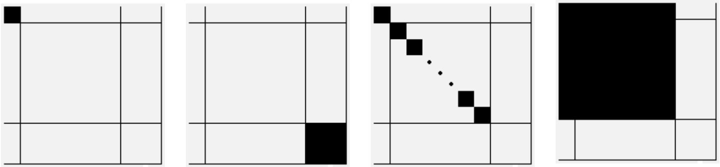


additive

Schematic representation of kernels

Applying $\mathcal{P} = \{\pi_{\mathcal{C}}, \pi_{\mathcal{U}_1}, \ldots, \pi_{\mathcal{U}_d}, \pi_{\mathcal{O}}\}$ to a kernel gives us a decomposition into $(d+2)^2$ parts.

We identify a projected kernel figuratively by a $(d+2) \times (d+2)$ matrix, e.g.



constant | ortho-add. | additive | (full) add.

Decomposition of a product kernel

$$
\begin{aligned}
((\pi_{\mathcal{O}} \otimes \pi_{\mathcal{O}})\, k)(\mathbf{x}, \mathbf{y}) = \mathcal{E} \Bigg[& \frac{k(\mathbf{x}, \mathbf{y})}{\mathcal{E}} + \sum_{i=1}^{d} \left(\frac{k_i(x_i, y_i)}{\mathcal{E}_i} - \frac{E_i(x_i) E_i(y_i)}{\mathcal{E}_i^2} \right) \\
& - \frac{E(\mathbf{x})}{\mathcal{E}} \left(1 + \sum_{i=1}^{d} \left(\frac{k_i(x_i, y_i)}{E_i(x_i)} - 1 \right) \right) \\
& - \frac{E(\mathbf{y})}{\mathcal{E}} \left(1 + \sum_{i=1}^{d} \left(\frac{k_i(x_i, y_i)}{E_i(y_i)} - 1 \right) \right) \\
& + \left(1 + \sum_{i=1}^{d} \left(\frac{E_i(x_i)}{\mathcal{E}_i} - 1 \right) \right) \cdot \left(1 + \sum_{i=1}^{d} \left(\frac{E_i(y_i)}{\mathcal{E}_i} - 1 \right) \right) \Bigg]
\end{aligned}
$$

where

- $E_i(x_i) := E_i(x_i, a_i, b_i) = \int_{a_i}^{b_i} k_i(x_i, y_i)\, dy_i$
- $E(\mathbf{x}) := E(\mathbf{x}, \mathbf{a}, \mathbf{b}) = \prod_{i=1}^{d} E_i(x_i, a_i, b_i)$
- $\mathcal{E}_i := \mathcal{E}_i(a_i, b_i) = \int_{a_i}^{b_i} E(x_i, a_i, b_i)\, dx_i$
- $\mathcal{E} := \mathcal{E}(\mathbf{a}, \mathbf{b}) = \prod_{i=1}^{d} \mathcal{E}_i(a_i, b_i)$

Decomposition of a product kernel

$$\begin{aligned}((\pi_{\mathcal{O}} \otimes \pi_{\mathcal{O}})\, k)(\mathbf{x}, \mathbf{y}) = \mathcal{E} \Bigg[& \frac{k(\mathbf{x}, \mathbf{y})}{\mathcal{E}} + \sum_{i=1}^{d} \Big(\frac{k_i(x_i, y_i)}{\mathcal{E}_i} - \frac{E_i(x_i) E_i(y_i)}{\mathcal{E}_i^2} \Big) \\ & - \frac{E(\mathbf{x})}{\mathcal{E}} \Big(1 + \sum_{i=1}^{d} \Big(\frac{k_i(x_i, y_i)}{E_i(x_i)} - 1 \Big) \Big) \\ & - \frac{E(\mathbf{y})}{\mathcal{E}} \Big(1 + \sum_{i=1}^{d} \Big(\frac{k_i(x_i, y_i)}{E_i(y_i)} - 1 \Big) \Big) \\ & + \Big(1 + \sum_{i=1}^{d} \Big(\frac{E_i(x_i)}{\mathcal{E}_i} - 1 \Big) \Big) \cdot \Big(1 + \sum_{i=1}^{d} \Big(\frac{E_i(y_i)}{\mathcal{E}_i} - 1 \Big) \Big) \Bigg]\end{aligned}$$

where

- $E_i(x_i) := E_i(x_i, a_i, b_i) = \int_{a_i}^{b_i} k_i(x_i, y_i)\, dy_i$
- $E(\mathbf{x}) := E(\mathbf{x}, \mathbf{a}, \mathbf{b}) = \prod_{i=1}^{d} E_i(x_i, a_i, b_i)$
- $\mathcal{E}_i := \mathcal{E}_i(a_i, b_i) = \int_{a_i}^{b_i} E(x_i, a_i, b_i)\, dx_i$
- $\mathcal{E} := \mathcal{E}(\mathbf{a}, \mathbf{b}) = \prod_{i=1}^{d} \mathcal{E}_i(a_i, b_i)$

Kriging

Kriging is done under the assumption that we know the true covariance kernel.

What is the impact of a misspecified kernel in the context of the "double" decomposition?

Kriging

Kriging is done under the assumption that we know the true covariance kernel.

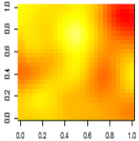
What is the impact of a misspecified kernel in the context of the "double" decomposition?

Controlled experiment:

- generate a realization of a random field using some kernel
- Split the data into a learning set and a test set
- Based on the learning set predict the other values using a misspecified kernel!
- Assess the quality of the predictions

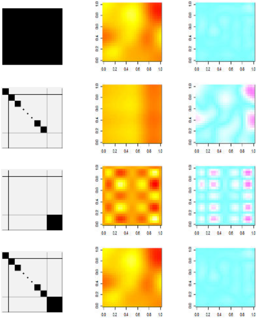
Concrete Experiment

Realization of a GRF generated with a Gaussian kernel



Z

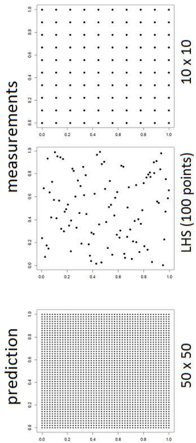
Predictions



!

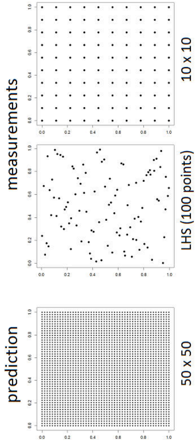
$\hat{Z}$ error

Concrete Experiment

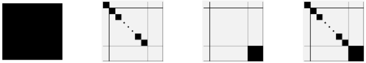


- Define learning and test set on a domain $D = [0, 1]^2$

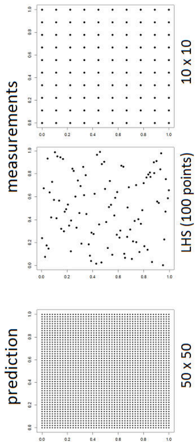
Concrete Experiment



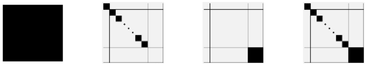
- Define learning and test set on a domain $D = [0,1]^2$
- Generate $Z := Z(\omega)$ using



Concrete Experiment

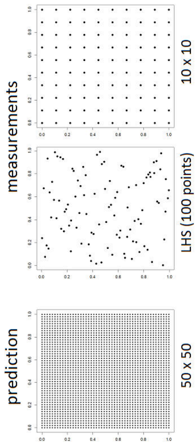


- Define learning and test set on a domain $D = [0,1]^2$
- Generate $Z := Z(\omega)$ using

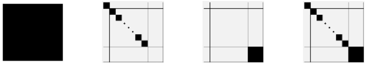


- Calculate the predictor $\hat{Z} := \hat{Z}(\omega)$ for every trajectory with all four kernels (using the measurements)

Concrete Experiment

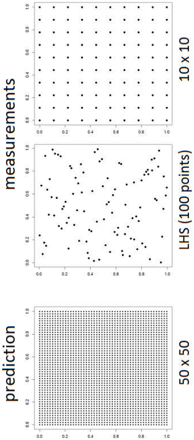


- Define learning and test set on a domain $D = [0, 1]^2$
- Generate $Z := Z(\omega)$ using

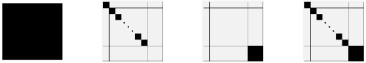


- Calculate the predictor $\hat{Z} := \hat{Z}(\omega)$ for every trajectory with all four kernels (using the measurements)
- Estimate $\int_D \left(\hat{Z}(x) - Z(x)\right)^2 d\mu$

Concrete Experiment

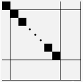
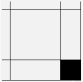
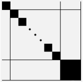
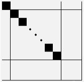
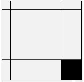
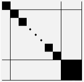


- Define learning and test set on a domain $D = [0,1]^2$
- Generate $Z := Z(\omega)$ using

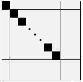
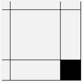
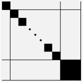
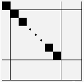
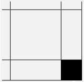
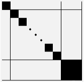


- Calculate the predictor $\hat{Z} := \hat{Z}(\omega)$ for every trajectory with all four kernels (using the measurements)
- Estimate $\int_D \left(\hat{Z}(x) - Z(x)\right)^2 d\mu$
- Repeat the procedure 200 times and take the mean over all results

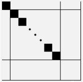
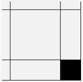
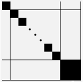
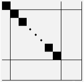
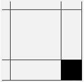
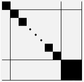
Results

				
	(0.086) 1	(0.033) 3	(0.055) 3	(0.088) 2
	(0.456) 3	(0.032) 1	(0.429) 4	(0.457) 3
	(6.472) 4	(5.927) 4	(0.043) 1	(5.579) 4
	(0.087) 2	(0.032) 2	(0.055) 2	(0.087) 1

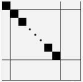
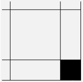
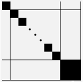
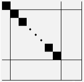
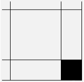
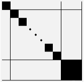
Results

				
	(0.086) 1	(0.033) 3	(0.055) 3	(0.088) 2
	(0.456) 3	(0.032) 1	(0.429) 4	(0.457) 3
	(6.472) 4	(5.927) 4	(0.043) 1	(5.579) 4
	(0.087) 2	(0.032) 2	(0.055) 2	(0.087) 1

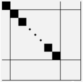
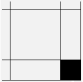
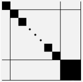
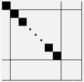
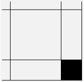
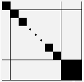
Results

				
	(0.086) 1	(0.033) 3	(0.055) 3	(0.088) 2
	(0.456) 3	(0.032) 1	(0.429) 4	(0.457) 3
	(6.472) 4	(5.927) 4	(0.043) 1	(5.579) 4
	(0.087) 2	(0.032) 2	(0.055) 2	(0.087) 1

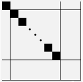
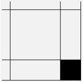
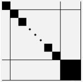
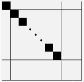
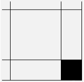
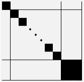
Results

				
	(0.086) 1	(0.033) 3	(0.055) 3	(0.088) 2
	(0.456) 3	(0.032) 1	(0.429) 4	(0.457) 3
	(6.472) 4	(5.927) 4	(0.043) 1	(5.579) 4
	(0.087) 2	(0.032) 2	(0.055) 2	(0.087) 1

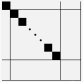
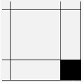
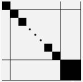
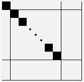
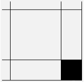
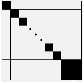
Results

				
	(0.086) 1	(0.033) 3	(0.055) 3	(0.088) 2
	(0.456) 3	(0.032) 1	(0.429) 4	(0.457) 3
	(6.472) 4	(5.927) 4	(0.043) 1	(5.579) 4
	(0.087) 2	(0.032) 2	(0.055) 2	(0.087) 1

Results

				
	(0.086) 1	(0.033) 3	(0.055) 3	(0.088) 2
	(0.456) 3	(0.032) 1	(0.429) 4	(0.457) 3
	(6.472) 4	(5.927) 4	(0.043) 1	(5.579) 4
	(0.087) 2	(0.032) 2	(0.055) 2	(0.087) 1

Results

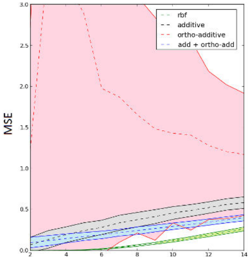
				
	**(0.086)** **1**	**(0.033)** **3**	**(0.055)** **3**	**(0.088)** **2**
	(0.456) 3	(0.032) 1	(0.429) 4	(0.457) 3
	(6.472) 4	(5.927) 4	(0.043) 1	(5.579) 4
	**(0.087)** **2**	**(0.032)** **2**	**(0.055)** **2**	**(0.087)** **1**

Some first Conclusions

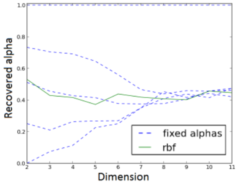
Summary of the presented work

- The kernel used for simulating the data always did the best predictions
- The additive kernel was less stable under the chosen circumstances
- The ortho-additive kernel much worse
- The combined additive and ortho-additive kernel performed as reliable as the full kernel
- A sparse kernel can carry almost the same information as a full one

Work in progress: Considerations in high dimensions



Development of the mean squared error with respect to the dimension



Simulation of GRFs with a kernel of the form

$$\alpha(\pi_{\mathcal{A}} \otimes \pi_{\mathcal{A}})k + (1-\alpha)(\pi_{\mathcal{O}} \otimes \pi_{\mathcal{O}})k, \ \alpha \in [0,1]$$

Recover the value of α by MLE

Summary of the presented work

- Ortho-additivity was introduced along with according projections of functions
- A kernel "double" decomposition was presented, and explicitly derived in the case of product kernels over $\mathbb{R}^d$
- Experiments suggested that neglecting cross-correlations between additive and ortho-additive parts have little influence on prediction for data generated with a Gaussian kernel

Summary of the presented work

- Ortho-additivity was introduced along with according projections of functions
- A kernel "double" decomposition was presented, and explicitly derived in the case of product kernels over $\mathbb{R}^d$
- Experiments suggested that neglecting cross-correlations between additive and ortho-additive parts have little influence on prediction for data generated with a Gaussian kernel

Selected perspectives

- Analyse which term is negligible by calculating relevant norms
- Define classes of kernels enabling to further exploit synergies between Kriging and Global Sensitivity Analysis
- Investigate further estimation procedures for high dimensions

References

Thank you for your attention!

This presentation is based on...

N. Durrande and D. Ginsbourger and O. Roustant and L. Carraro (2013)
ANOVA kernels and RKHS of zero mean functions for model-based sensitivity analysis. Journal of Multivariate Analysis 115 57 - 67

D. Ginsbourger and O. Roustant and N. Durrande (in preparation)
Invariances of random field paths, with applications in Gaussian Process Regression

J.E. Oakley and A. O'Hagan (2004)
Probabilistic Sensitivity Analysis of Complex Models: A Bayesian Approach. J. Roy. Statist. Soc. 66 751–769

F. Y. Kuo, I. H. Sloan, G. W. Wasilkowski and H. Wozniakowski (2010)
On decompositions of multivariate functions. Mathematics of Computation 79 953-966.