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

Additive covariance kernels for high-dimensional Gaussian process modeling.
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

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

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

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

$$f = f_C + f_{U_1} + \ldots + f_{U_d}$$
Considerations in $\mathcal{L}^2$

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

$$f = f_C + f_{U_1} + \ldots + f_{U_d} + f_O$$
Considerations in $\mathcal{L}^2$

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

\[ f = f_C + f_{U_1} + \ldots + f_{U_d} + f_O \]

\[ f_C = \int_D f \, d\mu \cdot 1_D \]

\[ f_{U_i} = \int_{D_{-i}} f - f_C \, d\mu_{-i} \cdot 1_{D_{-i}} \]

\[ f_A = f_C + \sum_{i=1}^{d} f_{U_i} \]

\[ f_O = f - f_A \]
Considerations in $L^2$

$f \in L^2$ can be decomposed

\[
f = f_C + f_{U_1} + \ldots + f_{U_d} + f_O
\]

\[
f_C = \int_D f \, d\mu \cdot 1_D =: \pi_C f
\]

\[
f_{U_i} = \int_{D_{-i}} f - f_C \, d\mu_{-i} \cdot 1_{D_{-i}} =: \pi_{U_i} f
\]

\[
f_A = f_C + \sum_{i=1}^d f_{U_i} =: \pi_A f
\]

\[
f_O = f - f_A =: \pi_O f
\]
Realizations $Z(\omega)$ of a GRF, generated with an isotropic kernel

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

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

$$\pi_A Z(\omega)$$
Realizations $Z(\omega)$ of a GRF, generated with an isotropic kernel

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

$\pi_A Z(\omega)$

$\pi_O Z(\omega)$
Projecting a random field

Realizations $Z(\omega)$ of a GRF, generated with an isotropic kernel

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

$$\pi_A Z(\omega)$$

$$\pi_O Z(\omega)$$

$$\pi_A Z(\omega) + \pi_O Z(\omega)$$
Let $\mathcal{P}$ be a finite family of projections such that

$$\text{Id}_{\mathcal{L}^2} = \sum_{\pi \in \mathcal{P}} \pi$$
"Double" decomposition of a kernel

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

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

With these projections we can equally decompose a kernel

$$\text{Id}_{L^2 \times L^2} = \left( \sum_{\pi \in \mathcal{P}} \pi \right) \otimes \left( \sum_{\tilde{\pi} \in \mathcal{P}} \tilde{\pi} \right) = \sum_{\pi \in \mathcal{P}} \sum_{\tilde{\pi} \in \mathcal{P}} (\pi \otimes \tilde{\pi})$$
Let \( P \) be a finite family of projections such that

\[
\text{Id}_{L^2} = \sum_{\pi \in P} \pi
\]

With these projections we can equally decompose a kernel

\[
\text{Id}_{L^2 \times L^2} = \left( \sum_{\pi \in P} \pi \right) \otimes \left( \sum_{\tilde{\pi} \in P} \tilde{\pi} \right) = \sum_{\pi \in P} \sum_{\tilde{\pi} \in P} (\pi \otimes \tilde{\pi})
\]

\[
k(x, y) = \text{Cov}(Z_x, Z_y) = \text{Cov} \left( \sum_{\pi \in P} \pi Z_x, \sum_{\tilde{\pi} \in P} \tilde{\pi} Z_y \right)
\]

\[
= \sum_{\pi \in P} \sum_{\tilde{\pi} \in P} \text{Cov}(\pi Z_x, \tilde{\pi} Z_y) = \left( \sum_{\pi \in P} \sum_{\tilde{\pi} \in P} (\pi \otimes \tilde{\pi})k \right)(x, y)
\]
Applying $\mathcal{P} = \{ \pi_C, \pi_{U_1}, \ldots, \pi_{U_d}, \pi_O \}$ 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.
Applying $\mathcal{P} = \{\pi_C, \pi_{U_1}, \ldots, \pi_{U_d}, \pi_O\}$ 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
```
Applying $\mathcal{P} = \{\pi_C, \pi_{U_1}, \ldots, \pi_{U_d}, \pi_\emptyset\}$ 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.
Applying $\mathcal{P} = \{\pi_C, \pi_{U_1}, \ldots, \pi_{U_d}, \pi_O\}$ 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 $\mathcal{P} = \{\pi_C, \pi_{U_1}, \ldots, \pi_{U_d}, \pi_{O}\}$ 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

\[ ((\pi \otimes \pi) k)(x, y) = \mathcal{E} \left[ \frac{k(x, y)}{\mathcal{E}} + \sum_{i=1}^{d} \left( \frac{k_i(x_i, y_i)}{\mathcal{E}_i} - \frac{E_i(x_i)E_i(y_i)}{\mathcal{E}_i^2} \right) \right. \]

\[ \left. - \frac{E(x)}{\mathcal{E}} \left( 1 + \sum_{i=1}^{d} \left( \frac{k_i(x_i, y_i)}{E_i(x_i)} - 1 \right) \right) \right] \]

\[ \left. - \frac{E(y)}{\mathcal{E}} \left( 1 + \sum_{i=1}^{d} \left( \frac{k_i(x_i, y_i)}{E_i(y_i)} - 1 \right) \right) \right] \]

\[ + \left( 1 + \sum_{i=1}^{d} \left( \frac{E_i(x_i)}{\mathcal{E}_i} - 1 \right) \right) \cdot \left( 1 + \sum_{i=1}^{d} \left( \frac{E_i(y_i)}{\mathcal{E}_i} - 1 \right) \right) \]

where

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

\[(\pi \otimes \pi) k(x, y) = \mathcal{E} \left[ \frac{k(x, y)}{\mathcal{E}} + \sum_{i=1}^{d} \left( \frac{k_i(x_i, y_i)}{\mathcal{E}_i} - \frac{E_i(x_i)E_i(y_i)}{\mathcal{E}_i^2} \right) \right. \]

\[\left. - \frac{E(x)}{\mathcal{E}} \left( 1 + \sum_{i=1}^{d} \left( \frac{k_i(x_i, y_i)}{E_i(x_i)} - 1 \right) \right) \right. \]

\[\left. - \frac{E(y)}{\mathcal{E}} \left( 1 + \sum_{i=1}^{d} \left( \frac{k_i(x_i, y_i)}{E_i(y_i)} - 1 \right) \right) \right. \]

\[\left. + \left( 1 + \sum_{i=1}^{d} \left( \frac{E_i(x_i)}{\mathcal{E}_i} - 1 \right) \right) \cdot \left( 1 + \sum_{i=1}^{d} \left( \frac{E_i(y_i)}{\mathcal{E}_i} - 1 \right) \right) \right] \]

where

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

\[ \hat{Z} \]

\[ \text{error} \]

Kriging

Concrete Experiment

Realization of a GRF generated with a Gaussian kernel

\[ Z \]

Predictions

\[ \hat{Z} \]

\[ \text{error} \]

Covariance kernels accounting for non-additivity in kriging
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

Nicolas Lenz  Covariance kernels accounting for non-additivity in kriging
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 $\int_D (\hat{Z}(x) - Z(x))^2 d\mu$
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 $\int_D (\hat{Z}(x) - Z(x))^2 d\mu$

Repeat the procedure 200 times and take the mean over all results
## Results

### General Setting

- Projections
- Numerical experiments
- Conclusions and Perspectives

### Kriging

<table>
<thead>
<tr>
<th>1</th>
<th>(0.086)</th>
<th>(0.033)</th>
<th>(0.055)</th>
<th>(0.088)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>(0.456)</td>
<td>(0.032)</td>
<td>(0.429)</td>
<td>(0.457)</td>
</tr>
<tr>
<td>4</td>
<td>(6.472)</td>
<td>(5.927)</td>
<td>(0.043)</td>
<td>(5.579)</td>
</tr>
<tr>
<td>2</td>
<td>(0.087)</td>
<td>(0.032)</td>
<td>(0.055)</td>
<td>(0.087)</td>
</tr>
</tbody>
</table>

Nicolas Lenz

Covariance kernels accounting for non-additivity in kriging
## Results

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Setting</td>
<td>(0.086)</td>
<td>(0.087)</td>
<td>(0.033)</td>
<td>(0.032)</td>
</tr>
<tr>
<td>Projections</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Numerical experiments</td>
<td>(0.032)</td>
<td>(0.032)</td>
<td>(0.055)</td>
<td>(0.055)</td>
</tr>
<tr>
<td>Conclusions and Perspectives</td>
<td>(0.043)</td>
<td>(0.043)</td>
<td>(0.088)</td>
<td>(0.087)</td>
</tr>
</tbody>
</table>

Covariance kernels accounting for non-additivity in kriging
### Results

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0.086)</td>
<td>(0.033)</td>
<td>(0.055)</td>
<td>(0.088)</td>
</tr>
<tr>
<td>2</td>
<td>(0.456)</td>
<td>(0.032)</td>
<td>(0.429)</td>
<td>(0.457)</td>
</tr>
<tr>
<td>3</td>
<td>(6.472)</td>
<td>(5.927)</td>
<td>(0.043)</td>
<td>(5.579)</td>
</tr>
<tr>
<td>4</td>
<td>(0.087)</td>
<td>(0.032)</td>
<td>(0.055)</td>
<td>(0.087)</td>
</tr>
</tbody>
</table>

Covariance kernels accounting for non-additivity in kriging

Nicolas Lenz
## Results

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>3</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.086</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.033</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>0.055</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>0.088</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>4</th>
<th>1</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.456</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>0.032</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>0.429</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>0.457</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>4</th>
<th>1</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6.472</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>5.927</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>0.043</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>5.579</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.087</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.032</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>0.055</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>0.087</td>
</tr>
</tbody>
</table>
### Results

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>(0.086) 1</td>
<td>(0.033) 3</td>
<td>(0.055) 3</td>
<td>(0.088) 2</td>
</tr>
<tr>
<td>(0.456) 3</td>
<td>(0.032) 1</td>
<td>(0.429) 4</td>
<td>(0.457) 3</td>
</tr>
<tr>
<td>(6.472) 4</td>
<td>(5.927) 4</td>
<td>(0.043) 1</td>
<td>(5.579) 4</td>
</tr>
<tr>
<td>(0.087) 2</td>
<td>(0.032) 2</td>
<td>(0.055) 2</td>
<td>(0.087) 1</td>
</tr>
</tbody>
</table>

Nicolas Lenz  
Covariance kernels accounting for non-additivity in kriging
### Kriging

#### Results

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>3</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td>(0.086)</td>
<td>(0.033)</td>
<td>(0.055)</td>
<td>(0.088)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Values</td>
<td>(0.456)</td>
<td>(0.032)</td>
<td>(0.429)</td>
<td>(0.457)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Values</td>
<td>(6.472)</td>
<td>(5.927)</td>
<td>(0.043)</td>
<td>(5.579)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Values</td>
<td>(0.087)</td>
<td>(0.032)</td>
<td>(0.055)</td>
<td>(0.087)</td>
</tr>
</tbody>
</table>
Results

Kriging

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(0.086)</td>
<td>(0.033)</td>
<td>(0.055)</td>
</tr>
<tr>
<td>1</td>
<td>(0.456)</td>
<td>(0.032)</td>
<td>(0.429)</td>
</tr>
<tr>
<td>3</td>
<td>(6.472)</td>
<td>(5.927)</td>
<td>(0.043)</td>
</tr>
<tr>
<td>4</td>
<td>(0.087)</td>
<td>(0.032)</td>
<td>(0.055)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Nicolas Lenz
Covariance kernels accounting for non-additivity in kriging
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.
Development of the mean squared error with respect to the dimension

Simulation of GRFs with a kernel of the form
\[ \alpha (\pi_A \otimes \pi_A) k + (1 - \alpha) (\pi_O \otimes \pi_O) k, \quad \alpha \in [0, 1] \]

Recover the value of \( \alpha \) 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
This presentation is based on...

  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


- **F. Y. Kuo, I. H. Sloan, G. W. Wasilkowski and H. Wozniakowski (2010)**