Blind Kriging: Implementation and performance analysis

I. Couckuyt[a], A. Forrester[b], D. Gorissen[b], F. De Turck[a], T. Dhaene[a]

[a] *Ghent University-IBBT, Dept. of Information Technology (INTEC), Gaston Crommenlaan 8, 9000 Ghent*
[b] *University of Southampton, School of Engineering Sciences, University Road, Southampton, United Kingdom*

Abstract

When analysing data from computationally expensive simulation codes or process measurements, surrogate modelling methods are firmly established as facilitators for design space exploration, sensitivity analysis, visualisation and optimisation. Kriging is a popular surrogate modelling technique for data based on deterministic computer experiments. There exist several types of Kriging, mostly differing in the type of regression function used. Recently a promising new variable selection technique was proposed to identify a regression function in the Kriging framework. In this paper this type of Kriging, i.e., blind Kriging, has been efficiently implemented in Matlab® and has been extended. The implementation is validated and tested on several examples to illustrate the strength and weaknesses of this new, promising modelling technique. It is shown that the performance of blind Kriging is as good as, or better than ordinary Kriging. Though, blind Kriging comes at double the computational cost with respect to ordinary Kriging.

Keywords: blind Kriging, surrogate modelling, feature selection, variable subset selection, benchmark

1. Introduction

Many complex real world phenomena are difficult to study directly using controlled experiments. Instead, the use of computer simulations has become

Email address: ivo.couckuyt@ugent.be (I. Couckuyt)

commonplace as a feasible alternative. However, due to the computational cost of these high fidelity simulations, the use of surrogate modelling techniques has become indispensable. A popular surrogate model to approximate deterministic noise-free data is Kriging. First conceived by Danie Krige in geostatistics, these Gaussian Process [1] based surrogate models are compact and cheap to evaluate, and have proven to be very useful for tasks such as optimisation [2], design space exploration, visualisation, prototyping, and sensitivity analysis [3].

A thorough mathematically treatment of Kriging is given by [4]. Basically, Kriging models fit data first on a regression function $f(\mathbf{x})$, and, subsequently, construct a stochastic process $Z(\mathbf{x})$ through the residuals.

$$Y(\mathbf{x}) = f(\mathbf{x}) + Z(\mathbf{x}) \tag{1}$$

By approaching the approximation problem from a Bayesian point of view, Kriging inherits a solid mathematical foundation with several useful properties, e.g., a closed formula for approximating the prediction variance.

Intuitively, the regression function in Kriging can be seen as the part trying to capture the general trend and thus the largest variations of the data. While the stochastic part takes care of small details and the interpolation of the data. However, choosing the right regression function for a set of data is a difficult and widely researched problem [5]. A simple approach would be just to apply a ranking method that assigns scores to the individual variables of the dataset. Afterwards, the most promising variables according to this score are selected to participate in a regression function. However, to also identify interactions between variables, features should be defined, i.e., an interaction between a set of variables (quadratic, linear-linear, etc.). The whole process of choosing the right terms is referred to as variable or feature selection. Note that the terms "*variable*" and "*feature*" are often used interchangeably, while in theory they have a different meaning [1].

[1] the term "*variables*" is used to denote the raw input variables. While, "*features*" denote artificial variables constructed from the raw input variables.

Several feature selection methods are discussed in the literature. [6] proposed to apply a Bayesian feature selection method to Kriging. This form of Kriging is known as blind Kriging and is the main focus of this paper.

The contributions of this paper are an efficient Matlab® implementation of blind Kriging, and a thorough analysis and performance study of blind Kriging on a highly distinct set of mathematical and real life problems from the literature. In particular, blind Kriging is adapted to include a re-estimation of the hyperparameters during the feature selection process, as well as a normalization of the training dataset, which requires modified formulae. Furthermore, these changes are compared against traditional blind Kriging, as explained in [6], using several statistical methods, e.g., error on a dense test set, histogram plots, robustness analysis, etc.

First, a brief introduction of variable selection is given in section 2. Then, in subsections 3.1-3.3, blind Kriging and the associated Bayesian feature selection method is explained. Subsequently, in subsection 3.4 the Matlab® implementation of blind Kriging is discussed in detail. Section 4 describes the experimental setup used for testing this implementation against several applications. Subsections 4.1-4.6 discuss the results of the various application benchmarks, illustrating strengths and weaknesses of blind Kriging. The last section concludes this paper and describes future work.

2. Feature selection

The time and space complexity of many surrogate modelling techniques (polynomials, radial basis functions, Kriging, etc.) scale exponentially with the dimensionality of the problem. In literature this is often noted as the *"curse of dimensionality"*. By taking advantage of feature selection methods this can often be (somewhat) alleviated.

Feature selection is important if the data contains a large amount of variables. Usually, not all of these variables are relevant for the problem at hand, and several sets of variables might be correlated. Therefore, feature selection methods are applied to simplify the data. Irrelevant variables may be either left out completely (dimension reduction) or aggregated into variable subsets (known as a *features*). This results in an updated dataset, expressed in terms of these features, which is easier to analyse. The machine learning community is particularly active in the domain of feature selection methods as they often have to deal with enormous amounts of high dimensional data. Most research is focused on classification techniques, however, a lot of these techniques are also applicable to regression problems. It should be noted that the problem of selection the right features in regression is more commonly known as variable subset selection or subset selection in regression [5].

According to [7] feature selection methods can be classified into three categories: filters, wrappers and embedded methods.

Filters are pre-processing methods and, thus, not associated with a particular prediction model. Variable ranking is an important filter method and forms the foundation of many feature selection algorithms. Variable ranking methods assign a score to the features corresponding to their influence on the response according to some correlation criteria, e.g., Pearson's correlation. This allows the practitioner to select only those variables that are most sensitive to the response of interest. Note that this method does not consider the effects between variables. More advanced methods consider subsets of variables. These so-called variable subset selection mechanisms often use variable ranking as a principal component in their inner workings. Filter methods are said to be fast to compute and the resulting dataset is not particularly tuned for any prediction model and thus no bias is introduced, though, under certain assumptions optimality to a predictor can be proven.

Wrappers, popularised by [7], view the prediction model as a black box and only use them to assess the usefulness of a subset of variables. In general, algorithms in the wrapper category must define approximately four components: a search space of all possible variable subsets (1), a search strategy (2), a prediction model (3) and a performance measure of the prediction model to guide (and halt) the search (4). Two popular greedy search strategies are *forward selection* and *backwards elimination*, progressively incorporating features or discarding features respectively. Though, other strategies such as genetic algorithms may be used. While easy to use and applicable to any prediction model this approach requires a retraining of the prediction model several times.

The last category, *embedded methods*, tightly integrates with the prediction method (e.g., embedded in the training phase) and thus is often faster than wrappers. An intuitive way to determine the influence of a feature is to calculate the derivative. This can be computed exactly for some models and training methods [8, 9]. Other prediction models, e.g., kernel based methods, inherently provide an approximation of the derivatives (sensitivity) [10]. A simple approach is to fit a polynomial on the data using least squares, including all candidates features. The coefficients of the polynomial represent the influence of a particular feature on the response and can be used to guide the selection of features in the final prediction model. An example of a kernel based method is Kriging. The Kriging model parameters quantify the variance of the response in each dimension. In fact, many kernel based methods, such as support vector machines [11], have this benefit as long as the appropriate kernel function is used. This approach, referred to as "Automatic Relevance Detection", basically attaches a weight to each dimension in the kernel function (the weights are determined using normal model parameter optimisation routines). Depending on the kernel function the weights denote the influence, sensitivity, variance, etc. of the associated dimensions. Though this approach only works on the raw input variables and, hence, are only suited for dimension reduction (or adding dimensions). For more information see [12].

The feature selection mechanism detailed in this paper belongs to the embedded class of methods, due to the tight coupling with Kriging. In particular, a Bayesian ranking method is used to rank several candidate features and a (greedy) forward selection procedure constructs a regression function. Subsequently, a Kriging model is used to measure the accuracy of the chosen regression function. Thus, the forward selection procedure is guided by a Bayesian ranking method and halted when the accuracy of the Kriging model increases again.

3. Blind Kriging

3.1. Introduction

We assume that the reader is acquainted with traditional Kriging, and refer to [4] for an in-depth description. Though, a couple of notations and definitions are introduced to explain the Bayesian forward selection mechanism of blind Kriging further on.

Assume a set of n samples, $(\mathbf{x}_1, \ldots, \mathbf{x}_n)'$ in d dimensions (see Equation 2) and associated function values, $\mathbf{y} = (y_1, \ldots, y_n)$. Furthermore, for ease of notation only one output is assumed. Though, the algorithm is easily generalised to multi-output approximation.

$$X = \left(\mathbf{x}_1, \ldots, \mathbf{x}_n \right)' = \begin{pmatrix} x_{1,1} & \ldots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \ldots & x_{n,d} \end{pmatrix} \tag{2}$$

A Kriging model is often regarded as a combination of a regression function $f(\mathbf{x})$ (also referred to as trend function) and a stochastic process $Z(\mathbf{x})$, see Equation 1. Depending on the form of the regression function Kriging has been prefixed with different names. *Simple* Kriging assumes the regression function to be a known constant, i.e., $f(\mathbf{x}) = 0$. A more popular version is *ordinary* Kriging, which assumes a constant but unknown regression function $f(\mathbf{x}) = \hat{\mu}$. Though, other, more complex, trend functions are possible such as linear or quadratic polynomials. In general, *universal* Kriging treats the trend function as a multivariate polynomial, namely,

$$f(\mathbf{x}) = \sum_{i=0}^{p_1} \ldots \sum_{k=0}^{p_d} a_{i \ldots k} x_1^i \ldots x_d^k, \tag{3}$$

where $p_1 \ldots p_d$ denote the maximum order of each variable $x_1 \ldots x_d$ and $a_{i \ldots k}$ are the coefficients.

As the actual full behaviour of the response is unknown it is often hard to choose which trend function to use for the given problem. Feature selection methods offer the possibility to identify the most plausible interactions occurring in the data. Therefore, it is often useful to apply feature selection techniques prior to fitting the data and, thus, by ranking the candidate features, automatically select a trend function that captures most variations of the data. As reviewed in section 2, many approaches are available, though one Bayesian method [13] in particular seems promising due to its connection with Kriging. [6] applied this Bayesian feature selection method to Kriging with interesting results. The method, referred to as *blind* Kriging, is outlined below.

3.2. Fundamentals

A blind Kriging model utilizes a linear trend function $f(\mathbf{x})$ (i.e., linear in the parameters a_i),

$$f(\mathbf{x}) = a_0 + \sum_{i=1}^{m} a_i f_i(\mathbf{x}), \tag{4}$$

where $f_i(\mathbf{x})$ are $i = 1 \ldots m$ unknown basis functions and $\mathbf{a} = (a_0, a_1, \ldots, a_m)$ denotes the coefficients. Let Ψ be the correlation matrix of the samples, $\psi(\mathbf{x})$ the correlation vector between a point $\mathbf{x}$ and the samples and M the $n \times m$ model matrix of the samples. Then the Best Linear Unbiased Predictor (BLUP) of the blind Kriging model is given as,

$$\hat{y}(\mathbf{x}) = a_0 + \sum_{i=1}^{m} a_i f_i(\mathbf{x}) + \psi(\mathbf{x})\Psi^{-1}(y - M \cdot a'). \tag{5}$$

The goal of blind Kriging is to efficiently determine the basis functions f_i (features) that captures the most variance in the sample data. To that end, a set of candidate functions is considered from which to choose from. In the ideal case the sample data is almost fully represented by the chosen trend function and the stochastic process $Z(\mathbf{x})$ has little or no influence.

Consider an existing Kriging model, the goal is to select new features to be incorporated in the regression function of this Kriging model, taking into account features that are already part of the regression function of this Kriging model. Hence, subsequently, the whole set of candidate functions b_i is used to fit the data in a linear model, i.e.,

$$g(\mathbf{x}) = a_0 + \sum_{i=1}^{m} a_i f_i(\mathbf{x}) + \sum_{i=1}^{t} \beta_i b_i(\mathbf{x}), \tag{6}$$

where t denotes the number of candidate functions.

The first part of this Equation is the regression function of Kriging and, hence, the coefficients $\mathbf{a}$ have already been determined independently of $\beta = (\beta_1, \ldots, \beta_t)$. The estimation of β provides a relevance score of the candidate features. A frequentist estimation of β (e.g., least-squares solution) would be a straightforward approach to rank the features. However, this is not always possible as the number of candidate features is often higher than the number of samples available. For instance, considering all possible interactions up to the quadratic effect in four dimensions the number of candidate features is $t = 3^4 = 81$. To that end, a Gaussian Prior distribution is introduced for β. Let $g(\mathbf{x})$ be a Gaussian Process,

$$g(\mathbf{x}) \sim GP(\mu, \sigma^2 \psi), \tag{7}$$

where μ is the mean and $\tau^2 \psi(\cdot)$ is the covariance function of the Gaussian Process. Then,

$$\beta \sim \mathcal{N}(0, \tau^2 R), \tag{8}$$

where $R = U^{-1} \Psi (U^{-1})'$ and U is the model matrix, namely, a design matrix with t rows. Furthermore, the choice of correlation functions is restricted to the product correlation form,

$$\psi(\mathbf{h}) = \prod_{j=1}^{d} \psi_j(|h_j|), \tag{9}$$

the variance-covariance matrix R can be constructed independently of the number of dimensions,

$$R_j = U_j^{-1} \psi_j (U_j^{-1})', \tag{10}$$

where U_j is the model matrix of the samples for factors $j = 1 \ldots d$. Thus the number of considered features can be chosen per dimension and afterwards the full matrix R is obtained by taking the Kronecker product,

$$R = \bigotimes_{j=1}^{d} R_j. \tag{11}$$

While the Bayesian variable selection is able to handle arbitrarily high-order effects, the matrix R grows quite rapidly. Hence, it may be useful to consider the special case where only linear effects, quadratic effects and two-factor interactions are identified. The total set of candidate functions is then defined by $b_i(\mathbf{x})$, where $i = 1 \ldots t = 2d^2$. Note that t does not scale exponentially as above, but still the matrix R would already require $(t+1) \times (t+1)$ $(> 4d^4)$ entries.

Let U_j be 3×3 orthogonal polynomial coded [14] matrices, then

$$R_j = U_j^{-1}\psi_j(U_j^{-1})' = U_j'\psi_j U_j = \\ \begin{pmatrix} 3 + 4\psi_j(1) + 2\psi_j(2) & 0 & -\sqrt{2}(\psi_j(1) - \psi_j(2)) \\ 0 & 3(1 - \psi_j(2)) & 0 \\ -\sqrt{2}(\psi_j(1) - \psi_j(2)) & 0 & 3 - 4\psi_j(1) + \psi_j(2) \end{pmatrix}, \tag{12}$$

this requires scaling of the sample data to the interval $[1, 3]$. The encoded samples for linear and quadratic effects are then, respectively, defined by,

$$x_{j,l} = \frac{\sqrt{3}}{\sqrt{2}}(\mathbf{x}_j - 2), \tag{13}$$

$$x_{j,q} = \frac{1}{\sqrt{2}}(3(\mathbf{x}_j - 2)^2 - 2), \tag{14}$$

for $j = 1 \ldots d$, where $\mathbf{x}_j$ denotes the j^{th} column of X. Other candidate terms can be constructed from these basic effects, e.g., the linear-quadratic effect between x_{15} and x_6 is represented by $b_i = x_{15,l} \cdot x_{6,q}$ for a particular i. As x_j takes on the values 1, 2 and 3 the column lengths of $x_{j,l}$ and $x_{j,q}$ will be $\sqrt{3}$.

While there is some (negative) correlation between mean and quadratic effects (see Equation 12), [15, 13] propose to only use the information of the diagonal of R_j. Normalising to the mean $3 + 4\psi_j(1) + 2\psi(2)$ (first entry of the diagonal of R_j) the variance-covariance matrix R can be expressed as follows. For ease of notation let $\psi(\mathbf{x})$ be a vector of length d,

$$\psi(\mathbf{x}) = \begin{pmatrix} \psi_1(\mathbf{x}) \\ \vdots \\ \psi_d(\mathbf{x}) \end{pmatrix}, \tag{15}$$

the vectors $\mathbf{r_l}$ and $\mathbf{r_q}$ of length d are then defined by,

$$\mathbf{r}_l = \frac{3 - 3\psi(2)}{3 + 4\psi(1) + 2\psi(2)}, \tag{16}$$

$$\mathbf{r}_q = \frac{3 - 4\psi(1) + \psi(2)}{3 + 4\psi(1) + 2\psi(2)}, \tag{17}$$

finally let $\mathbf{l_i}$ be the vector where element $l_{i,j} = 1$ if β_i includes the linear effect of factor j and 0 otherwise. In addition, define $\mathbf{q_i}$ as the vector where element $q_{i,j} = 1$ if β_i includes the quadratic effect of factor j and 0 otherwise. Then the diagonal matrix R is defined as,

$$R = \begin{pmatrix} \mathbf{r}_l^{l_1} \cdot \mathbf{r}_q^{q_1} & 0 & \dots & 0 \\ 0 & \ddots & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{r}_l^{l_{t+1}} \cdot \mathbf{r}_q^{q_{t+1}} \end{pmatrix}. \tag{18}$$

Note that, as the correlations between mean and quadratic effects have been dropped from Equation 12 the matrix R is in fact an estimation of the real correlation matrix.

Having constructed the variance-covariance matrix R by any means explained above, the posterior of β is estimated by,

$$\hat{\beta} = \frac{\tau^2}{\sigma^2} R M_c' \Psi^{-1}(\mathbf{y} - M \cdot \mathbf{a}), \tag{19}$$

$$var(\hat{\beta}) = \tau^2 (R - \frac{\tau^2}{\sigma^2} R M_c' \Psi^{-1} M_c R), \tag{20}$$

where M_c is the $n \times (t+1)$ model matrix of all candidate variables, M is the model matrix of all currently chosen variables and Ψ is the correlation matrix of the samples.

3.3. Selecting features

The coefficients $\hat{\beta}$ obtained through this Bayesian variable ranking method quantifies the importance of the associated candidate feature with respect to the data. There are several heuristics proposed to identify the best set of variables to approximate the data. Originally [13], the feature selection consisted of a greedy forward selection procedure, iteratively adding candidate variables with highest standardised coefficients to the Kriging model. In blind Kriging [6] the standardised coefficient is replaced with the absolute value of $\hat{\beta}$, delivering similar results while easier to compute. Note that, the first term of Equation 19 is a constant and does not influence the end results, i.e., $\frac{\tau^2}{\sigma^2}$ is set to 1.

The advantage of choosing this Bayesian variable selection method over other techniques is the tight coupling with Kriging's correlation matrix Ψ, thus taking advantage of already available information. Moreover, this variable selection method incorporates the important variable selection principles - effect hierarchy[2] and effect heredity[3] [16] - in the prior belief of β. In other words, the chosen features should form a simple and interpretable regression function.

In summary, constructing blind Kriging models can be seen as a two stage process. In the first phase an ordinary Kriging model, namely, a Kriging model with a constant regression function, is constructed and θ parameters are estimated. In a second phase the regression function of this initial Kriging model is extended with promising features according to the estimated $\hat{\beta}$ coefficients, generating a series of intermediate Kriging models. When these intermediate Kriging models stop improving on the leave-one out cross validation prediction error, the search is halted (though a look-ahead of n steps can be used to avoid

[2] Effect hierarchy denotes that low order effects (e.g., individual variables) should be chosen before high order effects (e.g., interactions of variables)

[3] Effect heredity states that an effect cannot be important until its parent effect is also important

Algorithm 1 Pseudo-code of the blindDACE toolbox.

```
$X \leftarrow$samples
$b_1, \ldots, b_t$ {Candidate features}
$\mathbf{b} = C^{te}$ {Selected features}
$\theta_0 = \max_\theta \mathit{likelihood}(X, \theta, \mathbf{b})$
$M_0 = \mathit{construct}(X, \theta_0, \mathbf{b})$ {Construct ordinary Kriging model}
$\alpha_0 = \mathit{evaluateMeasure}(M_0)$ {Assess accuracy}
$i = 0$
while $\mathit{improvement}(\alpha)$ {Accuracy improves?}
  $i = i + 1$
  $\beta =\mathit{rank}(b_1, \ldots, b_t)$
  $j = \mathit{max}_j(|\beta_j|)$
  $\mathbf{b} = \mathbf{b} \cup b_j$
  $\theta_i = \max_\theta \mathit{likelihood}(X, \theta, \mathbf{b})$ {Optional}
  $M_i = \mathit{update}(M_{i-1}, \theta_i, \mathbf{b})$ {Intermediate Kriging model}
  $\alpha_i = \mathit{evaluateMeasure}(M_i)$ {Assess accuracy}
endwhile
$\theta_{final} = \max_\theta \mathit{likelihood}(X, \theta)$
$M_{final} = \mathit{update}(M_i, \theta_{final}, \mathbf{b})$ {Final blind Kriging model}
```

local optima). The current best set of features is then chosen to construct the final blind Kriging model, re-estimating the θ parameters (see Figure 1).

3.4. Software

Blind Kriging has been implemented in a stand alone Matlab® toolbox, the blindDACE toolbox[4], partly based on the original paper by [6] and associated R code. Pseudo-code of the complete blind Kriging algorithm is found in Algorithm 1.

As discussed in previous sections, the estimation of $\hat{\beta}$ requires the correct parameters θ of the Kriging model to be available. Therefore, first, an ordinary Kriging model is built and this involves estimating θ. Further on, the identified θ parameters are kept fixed throughout the Bayesian forward selection procedure while appropriate regression terms are being selected. In the ideal case θ is optimised (fine-tuned) in each iteration once a new every term is added. However, this adds a relative huge computational burden when constructing blind

[4] The blindDACE Toolbox is freely available under an open source license (AGPLv3) for download at `http://sumo.intec.ugent.be/?q=blindDACE`

models, i.e., the computational cost is roughly equal to ordinary Kriging multiplied by the number of features considered. Hence, θ is only identified once before the Bayesian feature selection procedure starts, and a second time once the complete trend function has been chosen. Though, this still requires twice the computational cost compared to traditional Kriging.

The optimisation strategy for the parameters θ of the Kriging model is of utmost importance as it has a large impact on the performance of (blind) Kriging. A nice overview of hyperparameter tuning strategies is given by [17], where a two-stage approach is suggested. First use a genetic algorithm (global search) to quickly explore the search space and obtain a rough estimate of θ. As Kriging is very sensitive to the last steps, while converging to the final accurate optimum, a hill climbing method (local search) is used to fine-tune the optimum. However, while such a two-stage process, i.e., a memetic algorithm [18], performs well with respect to finding the global optimum, it can be prohibitively expensive in higher dimensions. Therefore, another optimisation strategy is used in the blindDACE toolbox which is outlined below.

The efficient calculation of the likelihood involves the factorisation of the correlation matrix Ψ, and in this work a Cholesky decomposition is used requiring a time complexity of the order $O(n^3)$. To improve the efficiency, and to reduce the number of likelihood evaluations, derivative information is utilised in a Sequential Quadratic Programming (SQP) framework. The derivative of the concentrated likelihood can be calculated analytically or an adjoint [19] can be used. In both cases the SQP method is found to find competitive optima in comparison with the thorough search of a memetic algorithm while using significantly fewer likelihood evaluations. In addition, unlike a genetic algorithm, the used SQP method is deterministic resulting in a very robust blind Kriging implementation. In this work, the derivatives of the likelihood function are calculated analytically instead of using the adjoint method since the difference in computation time is negligible for the problems in this paper.

The basic algorithm is depicted in Figure 1. After choosing the initial set of candidate features and constructing the initial Kriging model, the candidate fea-

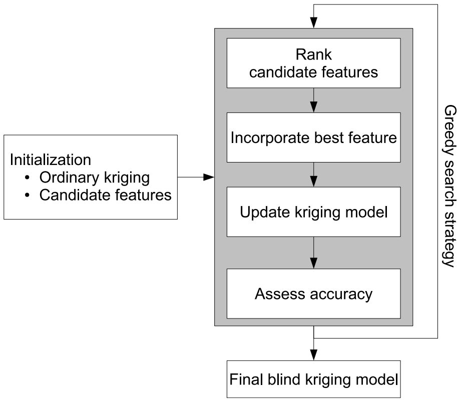


Figure 1: Flow chart of the blind Kriging construction process.

tures are ranked using the Bayesian variable selection method. Subsequently, a search strategy selects a new promising feature (according to the ranking) which is incorporated in the ordinary Kriging model. This intermediate Kriging model is scored against a measure and the features are ranked again. This process is repeated until the accuracy of the Kriging model stops improving. Several other adjustments were made to this basic algorithm to improve efficiency. These improvements are discussed in the remainder of this section.

Arguably the most important factor when determining relevant variable interactions is the method used to guide the search through the feature space, and, closely related, the criterion to stop adding terms. The leave-one-out cross validation has been used here, combined with a heuristic stopping criterion, as described in [6]. In addition, the blindDACE toolbox also supports the use of an unbiased hold-out set to validate the intermediate blind Kriging models as well as the likelihood itself, though this functionality has not been used in this paper.

Recall that Kriging requires inverting the correlation matrix Ψ. Depending on the number and the distribution of samples the correlation matrix may become close to singular, resulting in inaccuracies. Analogous to the DACE toolbox [20] and as suggested by [21] the blindDACE toolbox uses Cholesky and QR decompositions to efficiently compute the BLUP and likelihood. In addition, the likelihood score is set to infinity when a bad condition number is encountered, and so unstable Kriging models are effectively avoided.

The scaling of the sample data is also an important issue. In [6] all sample data used to be in the interval $[1, 3]$ (three-level factorial design). In the blind-DACE toolbox, as in the DACE toolbox, all data is normalised (into a standardised dataset), i.e., it scales the input and output data such that the data is distributed with mean 0 and variance 1,

$$\mu(\mathbf{x}) = 0 \textit{ and } \sigma(\mathbf{x}) = 1, \tag{21}$$

$$\mu(\mathbf{y}) = 0 \textit{ and } \sigma(\mathbf{y}) = 1, \tag{22}$$

this scaling is based on the hypothesis that, although being a distribution-free spatial interpolator, Kriging achieves its maximal efficiency only when the training data follows a Gaussian distribution. Hence, samples and values should be transformed to the *"Gaussian domain"* [22] when fitting, and afterwards, the BLUP, the uncertainty measures and all other estimates are transformed back to the original domain. While, the normalisation of the data as described above is far from being an exact conversion to the *"Gaussian domain"*, it does help to reduce the effects of outliers, i.e., extreme values in the input as well as output domain, and makes the scaled data easier to model. However, this scaling requires changing the orthogonal polynomial coding and R matrix equations. For arbitrarily bounds on the data, equations 13-14 can be written as,

$$x_{j,l} = \frac{\sqrt{3}}{\sqrt{2}} \left(\frac{\mathbf{x_j}}{\mathbf{l}_3} \right), \tag{23}$$

$$x_{j,q} = \frac{1}{\sqrt{2}} (3 \left(\frac{\mathbf{x_j}}{\mathbf{l}_3} \right)^2 - 2), \tag{24}$$

and equations 16-17 as,

$$\mathbf{r}_l = \frac{3 - 3\psi(\mathbf{l}_3 - \mathbf{l}_1)}{3 + 4\psi(\mathbf{l}_2 - \mathbf{l}_1) + 2\psi(\mathbf{l}_3 - \mathbf{l}_1)}, \tag{25}$$

$$\mathbf{r}_q = \frac{3 - 4\psi(\mathbf{l}_2 - \mathbf{l}_1) + \psi(\mathbf{l}_3 - \mathbf{l}_1)}{3 + 4\psi(\mathbf{l}_2 - \mathbf{l}_1) + 2\psi(\mathbf{l}_3 - \mathbf{l}_1)}, \tag{26}$$

where $\mathbf{l}_i$ denotes the i^{th} level in a factorial design. These can be defined as,

$$\mathbf{l}_1 = min(X), \tag{27}$$

$$\mathbf{l}_3 = max(X), \tag{28}$$

$$\mathbf{l}_2 = mean(X), \tag{29}$$

min and max take the columnwise minimum and maximum, respectively, of the sample matrix X such that $\mathbf{l}_1$ and $\mathbf{l}_3$ effectively provide the tightest bounding box of the data. $\mathbf{l}_2$ is the columnwise mean of the sample matrix X, when using the aforementioned scaling this is equal to $\frac{\mathbf{l}_1+\mathbf{l}_3}{2} \cong \mathbf{0}$.

Currently, the blindDACE toolbox only supports linear and quadratic trend functions. In fact, quoting from [23],

> "...for most well-defined physical system, only relatively low-order correlations of the input variables are expected to have a significant impact upon the output, and high-order correlated behaviour of the input variables is expected to be weak".

Thus, it is not crucial to consider higher interactions than quadratic. This is noticed by other authors such as [24].

Identifying higher order interactions (cubic and higher) is only possible when a greater number of levels is considered in the factorial design. Moreover, the R matrix and $\mathbf{r_l}$,$\mathbf{r_q}$ equations have to be adapted accordingly and appropriate coding and $\mathbf{r}$ equations should be added. In addition, the R matrix will grow rapidly in size when more candidate features are considered.

Finally, the blindDACE toolbox is limited to the Gaussian correlation function and the exponential correlation function, though, other correlation functions could easily be added as long as they can be written in the required product correlation structure notation (see Equation 9). For the Gaussian and exponential correlation function the formulae are given by

$$\psi_j(d_j) = e^{-\theta_j \cdot d_j^2}, \tag{30}$$

and,

$$\psi_j(d_j) = e^{-\theta_j \cdot |d_j|}, \tag{31}$$

respectively.

4. Performance

Intuitively blind Kriging can be expected to do better on problems where the regression function is able to approximate the general trend of the data. For instance, when using linear and quadratic effects as candidate variables then problems that (strongly) exhibit these effects would do rather well of course. Naturally, the behaviour of the response is unknown and thus no regression function can be defined a priori.

The performance of the blindDACE toolbox is applied to a very distinct set of real-life problems. First, the performance of this blind Kriging implementation is validated on three examples taken from the original blind Kriging paper [6], namely, a sealing experiment (subsection 4.1), piston slap (subsection 4.2) and the borehole model (subsection 4.3). Whenever a comparison is made to the findings of Joseph et al. this will be referred to as *the reference paper*. Afterwards, blind Kriging is applied to a novel problem from hydrology, quantifying the effects of pesticide leaching on ground water in Europe. Thereafter, blind Kriging is applied to a problem from Mechanical Engineering which has no clear trend, i.e., subsection 4.5, the application data has been obtained from [25]. Subsequently, blind Kriging is tested on a two dimensional mathematical function, i.e., the Branin function. The results are found in the following subsections.

All tests were performed using the standard Gaussian correlation function. For several examples, the implementation introduced in this paper is also compared against the freely available DACE toolbox [20]. The DACE toolbox is configured as ordinary Kriging, i.e., a constant regression function (*@regpoly0*), and the correlation function is set to Gaussian (*@corrgauss*). The ordinary Kriging model produced by the blind Kriging code is also taken into account.

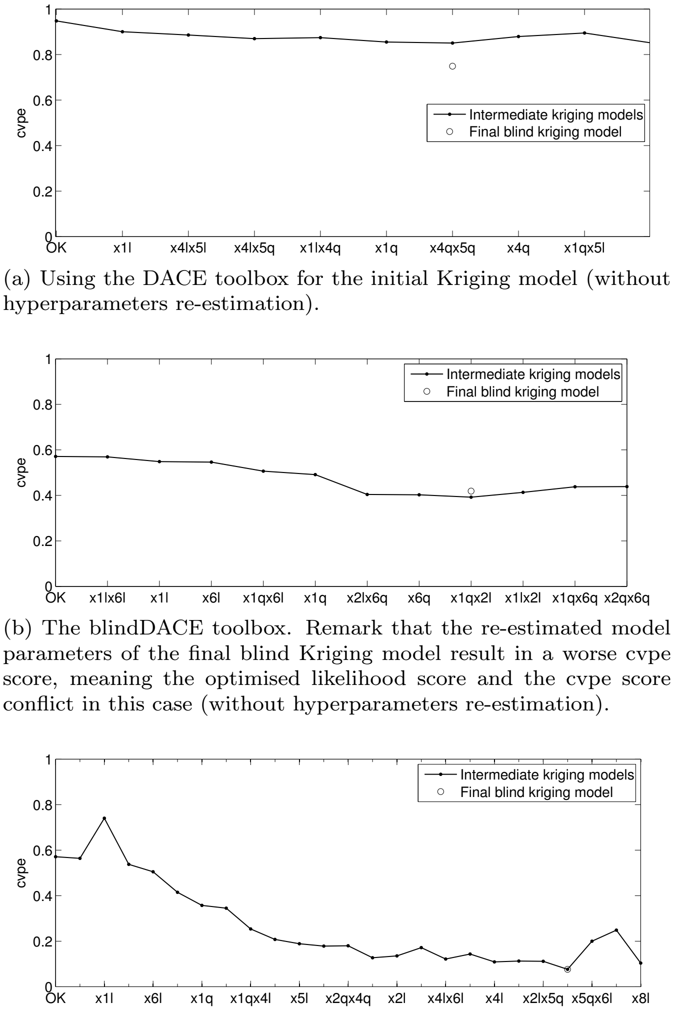
4.1. Sealing experiment

The first dataset was directly obtained from [6]. Therefore, a full description of the problem can be found in there. In summary, the data consists of eight input parameters and one output parameter resembling the gap lift in an engine block and head sealing experiment. There are 27 observations which form an orthogonal array. The goal of this experiment is to validate the correctness of the implementation and compare against the original findings of Joseph et al.

First the DACE toolbox was used to construct an ordinary Kriging model of this dataset. The obtained ordinary Kriging model serves as a base for the Bayesian forward selection procedure to produce the final blind Kriging model. The evolution of the cross validated prediction error (*cvpe*; leave-one-out) versus the chosen terms is shown in Figure 2a. Starting from a rather high leave-one-out score (in comparison to the reference paper) for the ordinary Kriging model (OK), the score decreases and settles at a value of approximately 0.87, after choosing six extra terms. After re-estimating the θ parameters, this time using the optimisation strategy discussed in subsection 3.4, there is much improvement to be found (the circle beneath the curve). Although, curiously, these results are much worse in comparison to the reference paper. Looking at the chosen features, see Equation 32, they do not correspond to the regression function found by Joseph et al.

$$1 + x_{1,l} + x_{4,l} \cdot x_{5,l} + x_{4,l} \cdot x_{5,q} + x_{1,l} \cdot x_{4,q} + x_{1,q} + x_{4,q} \cdot x_{5,q} \qquad (32)$$

Considering the poor score of the first constructed ordinary Kriging model it was decided to use the optimisation strategy of the blindDACE toolbox to



(a) Using the DACE toolbox for the initial Kriging model (without hyperparameters re-estimation).

(b) The blindDACE toolbox. Remark that the re-estimated model parameters of the final blind Kriging model result in a worse cvpe score, meaning the optimised likelihood score and the cvpe score conflict in this case (without hyperparameters re-estimation).

(c) The blindDACE toolbox. Note that not all chosen features are shown on the x-axis to avoid cluttering (with hyperparameters re-estimation; the cvpe scores are *after* the hyperparameters re-estimation).

Figure 2: Evolution of the Bayesian feature selection phase (sealing experiment).

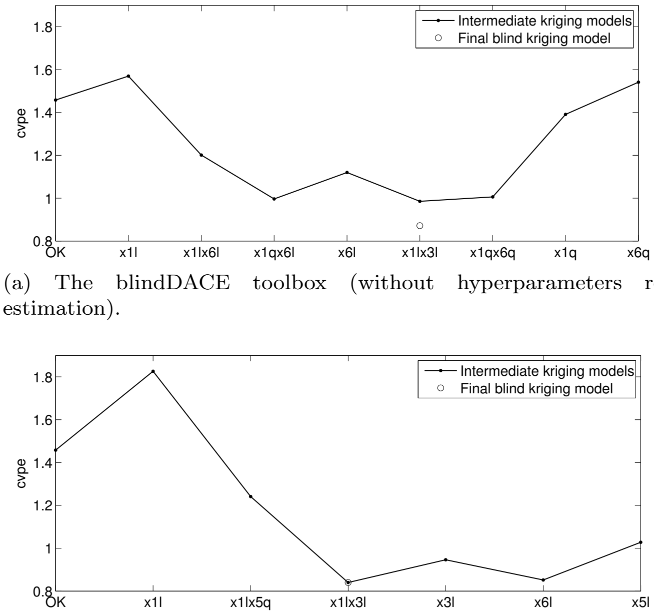
identify the θ parameters of the ordinary Kriging model. Results are depicted in Figure 2b. As can be seen a slightly better initial leave-one-out score is calculated and using this set of θ parameters the feature selection procedure is able to select the right features, reducing the score to 0.39 after eight terms. The final regression function of the blind Kriging model (without the coefficients) is given by (33). This regression function contains the same terms as the reference paper, though selected in a slightly different order. The difference is found in the last two terms $x_{6,q}$ and $x_{1,q} \cdot x_{2,l}$, which give lead to a rise in cross validation score in the reference paper and thus are not selected. Here it is found that they slightly lower the score, explained by a different set of θ parameters. Re-estimating the θ parameters it can be seen that the cross validation score for the blind Kriging model is actually worse with the newly identified parameters. After some testing it is found that those two last terms were not exactly appropriate. If those terms are left out of the regression function the final cross validation is in the same range as the reference paper.

$$\begin{aligned} & 1 + x_{1,l} \cdot x_{6,l} + x_{1,l} + x_{6,l} + x_{1,q} \cdot x_{6,l} \\ & +x_{1,q} + x_{2,l} \cdot x_{6,q} + x_{6,q} + x_{1,q} \cdot x_{2,l} \end{aligned} \tag{33}$$

This shows that the calibration of the initial ordinary Kriging model greatly influences the Bayesian feature selection procedure. Moreover, the second approach clearly shows that the implementation of blind Kriging is competitive with the reference paper. Any difference between the two is most likely due to a different set of θ parameters. In addition, note that the original dataset takes on exactly three levels (1, 2 and 3). Whereas in the proposed implementation the models are fitted on normalised data (see section

3.4).

The behaviour described above is mostly due to the optimised θ parameters not matching the current regression function anymore. In essence, those θ parameters are describing the function that is the difference between the original observations *minus* the current regression function. As optimisation is not



(a) The blindDACE toolbox (without hyperparameters re-estimation).

(b) The blindDACE toolbox (with hyperparameters re-estimation; the cvpe scores are *after* the hyperparameters re-estimation).

Figure 3: Evolution of the Bayesian feature selection phase (piston slap).

that expensive when using derivative information and for testing purposes it is decided to re-estimate the θ parameters after every added term. Results using this approach are shown in Figure 2c. Many more terms are included in the regression function resulting in a cross validation score as low as 0.1.

4.2. Piston slap

The second experiment is a dataset describing engine noise due to piston secondary motion (piston slap). For more information the reader is again referred to [6]. The dataset consists of six inputs and one output (the noise). This time, the dataset is directly approximated using solely the blind Kriging implementation and leave-one-out cross validation to guide the feature selection. The evolution of the feature selection stage can be found in Figure 3a. The resulting regression function is very similar to the reference paper, except for the last two terms.

$$1 + x_{1,l} + x_{1,l} \cdot x_{6,l} + x_{1,q} \cdot x_{6,l} + x_{6,l} + x_{1,l} \cdot x_{3,l} \tag{34}$$

At the fourth step the term $x_{6,l}$ is chosen, giving an increase in the cross validation score. However, by adding $x_{1,l} \cdot x_{3,l}$ the score decreases again, settling at an even lower value than before. While $x_{1,l} \cdot x_{3,l}$ may be a good feature, $x_{6,l}$ is clearly not. However, by using a greedy forward selection strategy previously selected terms are never reconsidered. It may be worthwhile to adapt the search strategy to eliminate (or skip) terms that result in a (temporary) increase of the score, though, one should of course not violate the principles of effect hierarchy and heredity.

Comparing these results with the paper of Joseph et al. it is immediately clear that the cross validation score is substantially lower for the proposed blind Kriging implementation. Here, two more terms are included in the regression function causing a lower cross validation score.

As in the previous example the configuration is slightly changed so as to re-estimate the hyperparameters after every added term. The surprising results are depicted in Figure 3b. The only common feature with the no hyperparameters re-estimation case (aside from the mean) is $x_{1,l}$ and only three extra terms are identified. Nevertheless, with this limited set of features blind Kriging is still able to achieve a cross validation score approximately the same as without re-estimating the hyperparameters.

4.3. Borehole model

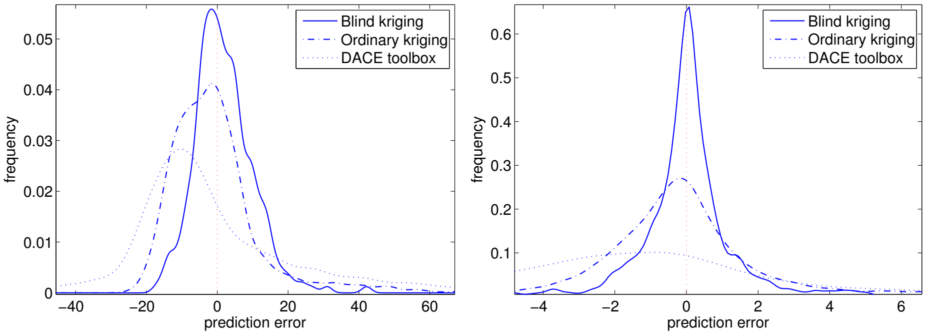
The final validation experiment is a simple analytical model to calculate the flow rate through a borehole [26]. There are eight inputs and the process is described by the following functions,

$$rr_w = \ln(\frac{r}{r_w}), \tag{35}$$

$$flow\,rate = \frac{2\pi T_u (H_u - H_l)}{rr_w[1 + (\frac{2LT_u}{rr_w r_w^2 K_w} + \frac{T_u}{T_l}]}, \tag{36}$$

	27 samples		200 samples	
	AEE	Improvement	AEE	Improvement
Blind Kriging	6.31	66%	0.92	84%
Ordinary Kriging	9.33	51%	2.24	61%
DACE toolbox	19.62	baseline	5.73	baseline

Table 1: Accuracy of the approximations on a test set (borehole model)



(a) 27 samples.

(b) 200 samples.

Figure 4: Density plot for the prediction errors (borehole model).

The same orthogonal array of 27 samples of subsection 4.1 is used to create a dataset of the borehole model. In addition, an optimised maximin Latin Hypercube Design (LHD; [27]) of 200 samples is constructed. These two datasets are used to create an ordinary Kriging model and blind Kriging model (without hyperparameters re-estimation) using the code presented in this paper. In addition, another ordinary Kriging produced by the DACE toolbox is taken into account. Hence, in total there are three Kriging models of which an error is calculated on a separate test set of $k = 6561$ samples. The Average Euclidean Error (AEE) is used, which is defined by,

$$AEE(\mathbf{a}, \mathbf{b}) = \frac{1}{k} \sum_{i=1}^{k} \sqrt{(a_i - b_i)^2}. \tag{37}$$

Results are found in Table 1. In both cases ordinary Kriging achieves a great improvement over the DACE toolbox due to a better optimisation strategy. Furthermore, blind Kriging offers for this problem a vast improvement over ordinary Kriging itself and, hence, the DACE toolbox. In addition, histogram plots, see Figure 4, of the prediction errors on the test set also compare in favour of blind Kriging. The resulting regression functions (in the order they are selected) of blind Kriging for the case of 27 (Equation 38) and 200 samples (Equation 39), respectively, are,

$$\begin{aligned}1 + x_{1,l} + x_{4,l} + x_{7,l} + x_{1,q} + x_{6,l} \\ + x_{1,l} \cdot x_{4,l} + x_{1,l} \cdot x_{7,l} + x_{1,l} \cdot x_{6,l} + x_{8,l}\end{aligned} \tag{38}$$

$$\begin{aligned}1 + x_{1,l} + x_{7,l} + x_{4,l} + x_{6,l} + x_{1,q} + x_{8,l} + x_{1,l} \cdot x_{4,l} \\ + x_{1,l} \cdot x_{7,l} + x_{1,l} \cdot x_{6,l} + x_{1,l} \cdot x_{8,l} + x_{4,l} \cdot x_{7,l} \\ + x_{6,l} \cdot x_{7,l} + x_{7,q} + x_{1,q} \cdot x_{4,l} + x_{4,l} \cdot x_{8,l} + x_{6,l} \cdot x_{8,l} \\ + x_{2,l} + x_{7,l} \cdot x_{8,l} + x_{1,l} \cdot x_{7,q} + x_{1,q} \cdot x_{7,l}.\end{aligned} \tag{39}$$

It is obvious that the two regression functions are quite similar aside from the order in which the features were chosen. In fact, the regression function of the second case is a superset of the one from the first case. In particular, 11 extra terms are identified likely due to more information being available (more samples). While Joseph et al. reported only a linear effect for x_1 as regression function, it is not immediately clear whether this term was chosen manually or using a search strategy as we have in this paper.

In summary, from the experiments of subsections 4.1, 4.2 and 4.3 it is clear that the implementation produces similar results as Joseph et al. [6].

4.4. EuroPEARL

This test case consists of a dataset of 51319 samples generated by the EuroPEARL model [28, 29]. EuroPEARL models the leaching of pesticides, taking into account transient flown, hydrodynamic dispersion, nonlinear adsorption, degradation, and uptake of pesticides by plant roots. The model is

	AEE	Improvement
Blind Kriging	0.92	51%
Ordinary Kriging	1.94	−3%
DACE toolbox	1.88	baseline

Table 2: Accuracy of the approximations on a test set (EuroPEARL model)

developed at the European scale mainly due ground water being a major drinking source for Europe. In particular EuroPEARL consists of a link between the one-dimensional, multi-layer, mechanistic pesticide leaching model PEARL and a Geographical Information System (GIS). More details about EuroPEARL and the associated metamodeling efforts, denoted MetaPEARL, is found in [30].

The disadvantage of such a process-based pesticide-leaching model are the rather large number of parameters of which some are significantly less relevant than others. Hence, the use of feature selection techniques may prove to be indispensable for this problem. The EuroPEARL configuration for this paper has six input parameters describing the response.

For this test case, a training dataset of 100 samples arranged in a LHD is subsampled from the full dataset. The remaining samples of the dataset are taken as a test set to calculate the true error, using the AEE function. The same approximation models as the previous section are considered, namely, blind Kriging, ordinary Kriging and the DACE toolbox. Note, in contrast to the previous section, the blind Kriging model is configured to re-estimate the hyperparameters after every added term.

The true errors of the constructed Kriging models are found in Table 2. While ordinary Kriging has a slightly worse accuracy than the DACE toolbox, the blind Kriging approximation is more than twice as accurate. Looking at the histogram plots, Figure 5, it is seen that ordinary Kriging and the DACE toolbox are biased towards the right, namely, predicting smaller values than the real response. Using only 100 samples, blind Kriging is able to select a trend function that follows the behaviour of the whole test set.

EuroPEARL is the perfect example for blind Kriging, a very small data-

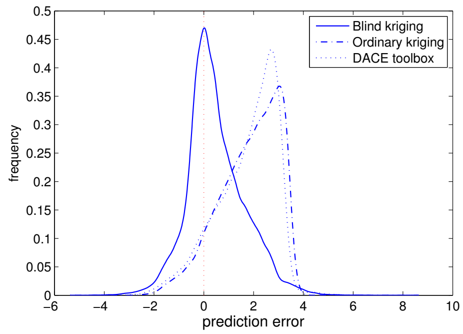


Figure 5: Histogram plot (EuroPEARL model).

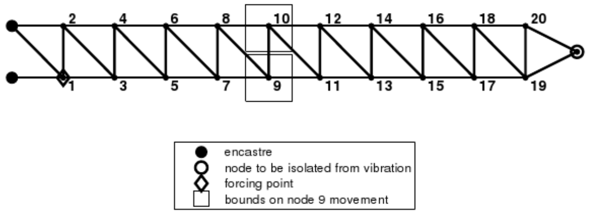

2
4
6
8
10
12
14
16
18
20
1
3
5
7
9
11
13
15
17
19
● encastre
○ node to be isolated from vibration
◇ forcing point
□ bounds on node 9 movement

Figure 6: Truss structure.

set which represents the features of the full response quite well. Equation 40 represents the final identified regression function of blind Kriging.

$$1 + x_{1,l} \cdot x_{5,q} + x_{3,l} \cdot x_{5,q} + x_{1,q} \cdot x_{3,q} + x_{4,q} \cdot x_{5,q} + x_{3,q} \cdot x_{5,l} \tag{40}$$

4.5. Truss dataset

The fifth dataset describes a structural dynamics problem. The problem is the optimal design of a two-dimensional truss, constructed by 42 Euler-Bernoulli beams, see Figure 6. The goal is to identify a design that is optimal (or close to) with respect to passive vibration isolation. To that end, a force is induced on a base node of the structure and the force attenuated through the structure is measured on the tip of the structure. A full description is given in [25].

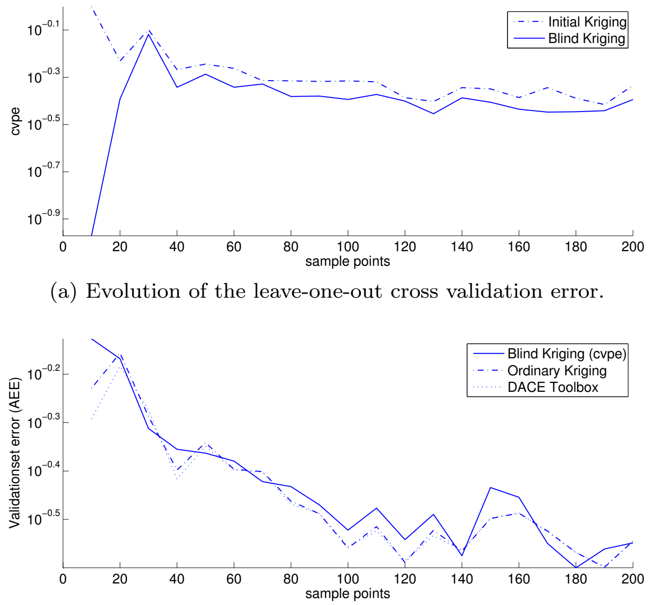
There are four input parameters defining the position of nodes nine and ten in the structure and one output parameter, i.e., the stress that the outermost node (the tip) receives. Note that the truss dataset is not used for optimisation purposes, instead the goal is the reproduce the landscape as accurate as possible. To that end, 20 datasets are constructed each arranged in an optimised maximin LHD. The datasets only differ in the number of observations, which ranges from 10 to 200 samples with steps of 10 samples.

For each dataset, a DACE toolbox model and a blind Kriging model (with hyperparameters re-estimation) have been constructed. In addition, the ordinary Kriging model resulting from the blind Kriging construction process is included as well. An estimation of the true error is obtained for each model on a separate random test set of $k = 100$ samples. Results are shown in Figure 7. While, the evolution of the leave-one-out score promises increased accuracy (Figure 7a), little of this can be seen in the final AEE scores (Figure 7b). Looking at this second plot it is seen that adding terms rarely increases the accuracy. This is explained by the fact that the truss datasets have no clearly defined trend. At least, no obvious linear or quadratic effects can be seen. Perhaps if the set of candidate variables are extended with more complex interactions terms a good regression function can be found. However, this requires some alterations in the existing implementation and thus is considered future work.

Density plots of the prediction errors for the case of 10 and 140 samples are shown in Figure 8. Not much improvement is found, on the contrary, the histogram for the case of 10 samples is hardly noticeable in favour of ordinary Kriging. Despite its attractiveness, blind Kriging is not suited for every problem available. Care should be taken when choosing the candidate variables. Ideally a domain expert is able to select plausible interactions for the given problem.

4.6. Branin function

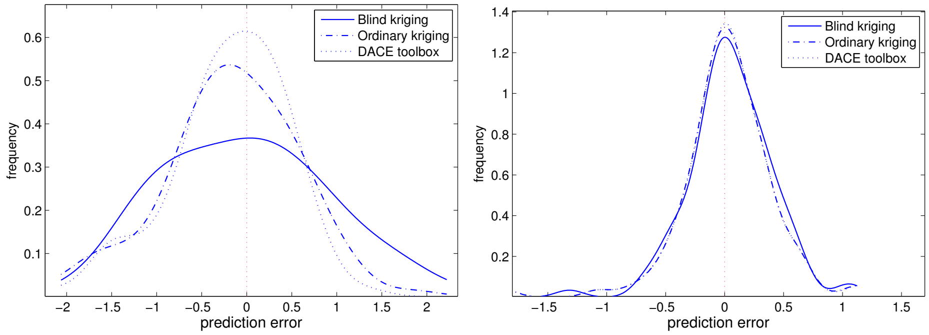
The Branin function is a well-known benchmark function for optimisation. The Branin function is defined by Equation 41. Again, 20 datasets arranged in an optimal maximin LHD [31] were used to construct the different Kriging models. Regard that the Branin function is not used here for optimisation purposes but the intent is to reproduce the landscape of the Branin function as accurately as possible.



(a) Evolution of the leave-one-out cross validation error.

(b) Evolution of the AEE versus the number of samples.

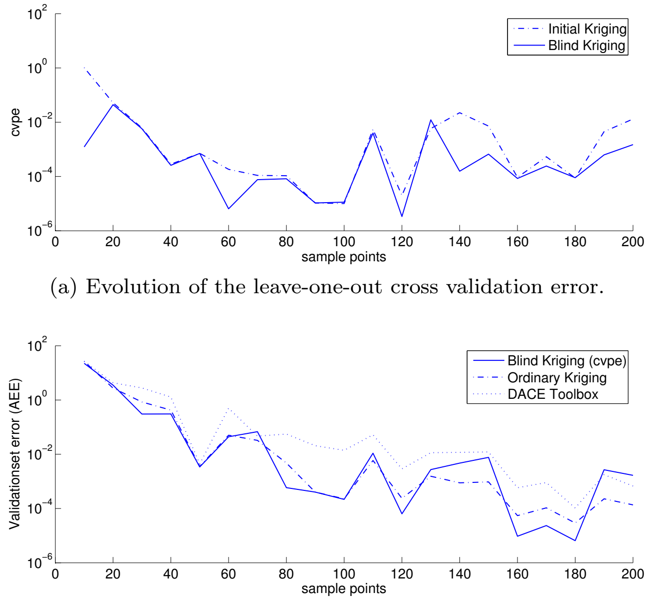
Figure 7: Accuracy of the prediction models (truss).



(a) 10 samples.

(b) 140 samples.

Figure 8: Density plot for the prediction errors (truss).



(a) Evolution of the leave-one-out cross validation error.

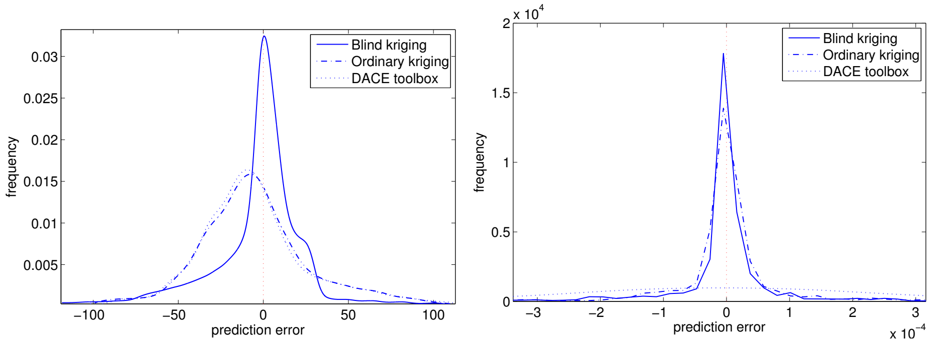
(b) Evolution of the AEE versus the number of samples.

Figure 9: Accuracy of the prediction models (Branin 1).

$$f(x_1, x_2) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10 \qquad (41)$$

Figure 9a depicts the final cross validated leave-one-out score (*cvpe*) of the blind Kriging model (with hyperparameters re-estimation) for each dataset, while the initial baseline score of the ordinary Kriging model is included as well. As expected the *cvpe* of blind Kriging is always lower than ordinary Kriging. An estimation of the true error is obtained for each model on a separate and very dense dataset arranged in an uniform grid of $k = 2500$ samples. The AEE errors on this dataset versus the number of samples are shown in Figure 9b.

The performance differences between the blind Kriging and ordinary Kriging models is quite small. There is no obvious better Kriging model for this problem which can explained by the abundance of data on this low dimensional problem, i.e., the stochastic part is always able to capture most of the variance making



(a) Dataset of 10 samples.

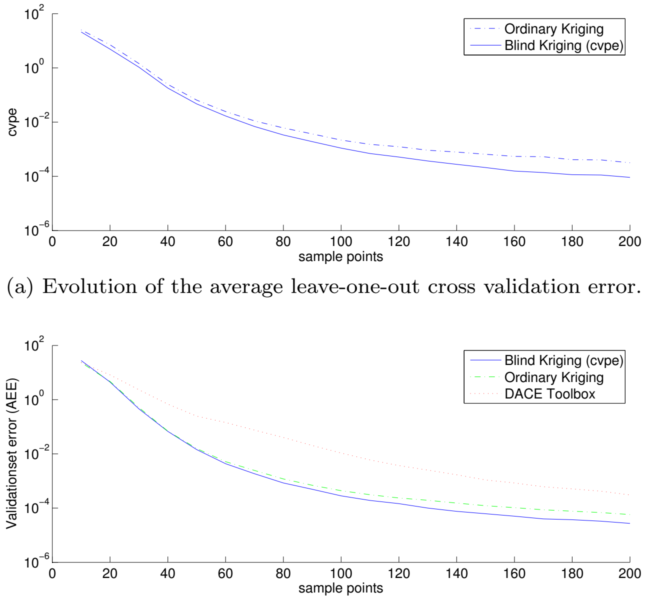
(b) Dataset of 140 samples.

Figure 10: Density plot for the prediction errors (Branin 1).

the regression function less important. Hence, the differences on the AEE scores are insignificant. More of interest are the distributions of the prediction errors. Therefore, two density plots of the prediction errors for the datasets of 10 and 140 samples are depicted in Figure 10.

Surprisingly, looking at Figure 10a the blind Kriging model has a much better distribution of the errors than the other two models, even though it has a slightly higher AEE score. It is clear that using a suitable regression function removes much bias and provides a more stable foundation for the stochastic process of Kriging to build on. However, as the number of samples increase blind Kriging has a harder time distinguishing itself from the classical Kriging models (see Figure 10b). While considering the simplicity of the problem, i.e., only 2 dimensions, it is straightforward that any type of Kriging model has little trouble approximating the landscape well. Easily reaching an accuracy score of 10^{-4}.

To have an idea of the robustness of blind Kriging with respect to the sample distribution, the Branin function test is repeated using random designs instead of an optimal LHD. For each number of samples, ranging from 10 to 200 samples in steps of 10 samples, 1000 uniform random designs are constructed. Thus, the fitting of the Kriging models under consideration is repeated 1000 times for each sample size.



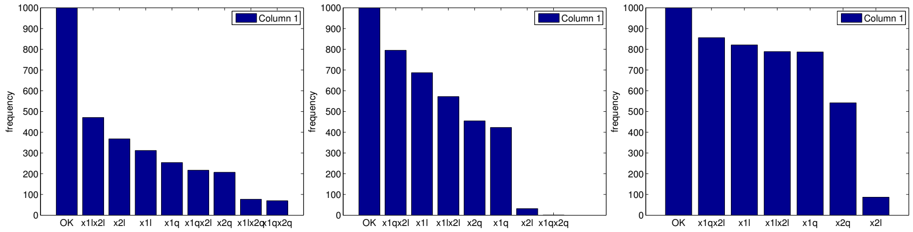
(a) Evolution of the average leave-one-out cross validation error.

(b) Evolution of the average AEE versus the number of samples.

Figure 11: Average accuracy of the prediction models (Branin 2).

Similar plots as above are possible, including error bars. Though, the error bars make the evolution plot somewhat cluttered, as such only the mean accuracy is plotted in Figure 11. Note that the evolution is much smoother than the previous evolution plots using a LHD. This time, blind Kriging performs consistently better than the other approximations on the true error, though not by much. It is obvious that blind Kriging is still somewhat sensitive to the distribution of the samples. Logically, correct feature selection is only possible as long as the dataset is a decent representation of the general behaviour of the simulation code.

More interesting is Figure 12 which depicts three histogram snapshots (10, 100 and 200 samples) describing the frequency each feature has been chosen (out of 1000). The leftmost bar (OK) is the mean which is always included in the regression function. As expected, the more samples are available the more accurate the selection of terms is, more specific, less terms are selected wrongfully and correct terms are selected more often.



(a) 10 samples. (b) 100 samples. (c) 200 samples.

Figure 12: Histogram plot of the chosen features (Branin 2).

5. Conclusion

This paper discussed blind Kriging, the associated Bayesian forward selection method and how it compares to traditional Kriging. An efficient implementation of blind Kriging, with numerous additions such as hyperparameters re-estimation and normalization of the training data, has been tested and validated on six different problems to illustrate strengths and weaknesses. Blind Kriging is able to identify good regression functions for problems that have a clear trend. In particular, the training data should cover non-linearities in the domain for blind Kriging to work nicely. On the other hand, if the considered features do not contain the interactions exhibited by the data then blind Kriging is not able to identify a good regression function and does not improve on ordinary Kriging. In some cases blind Kriging can be misguided by deceptive data and select wrong features, effectively decreasing the accuracy of the final model. In that regard more work is to be done on the search strategy to select candidate features and the associated metric to guide this search. Finally, it should be noted that when enough data is available to cover the domain ordinary Kriging performs equally well as blind Kriging and even slightly outperforms blind Kriging for some datasets.

In summary, blind Kriging is a valuable tool to approximate sparse data obtained from expensive simulation codes. Though, the resulting blind Kriging model should be carefully analysed, if possible using an independent test set. In particular, blind Kriging is more interesting for difficult, high dimensional, problems where limited data is available.

Future work includes investigating the impact of 4-level and higher factorial designs to identify higher order interactions, researching other optimisation strategies for choosing the best subset of variables (this includes performance metrics to guide the optimisation), etc. The blind Kriging implementation explained in this paper has been coded in Matlab® and is available for free at `http://sumo.intec.ugent.be/?q=blindDACE` under the AGPLv3 license and will be updated when new features are added.

Acknowledgements

Ivo Couckuyt is funded by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

[1] M. Gibbs, D. J. C. Mackay, Efficient implementation of gaussian processes, Tech. rep., Department of Physics, Cavendish Laboratory, Cambridge University (1997).

[2] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, J. of Global Optimization 13 (4) (1998) 455–492.

[3] G. Wang, S. Shan, Review of metamodeling techniques in support of engineering design optimization, Journal of Mechanical Design 129 (4) (2007) 370–380.

[4] J. Sacks, W. J. Welch, T. Mitchell, H. P. Wynn, Design and analysis of computer experiments, Statistical science 4 (4) (1989) 409–435.

[5] A. Miller, Subset Selection in Regression, London: Chapman & Hall, 1990.

[6] V. R. Joseph, Y. Hung, A. Sudjianto, Blind kriging: A new method for developing metamodels, ASME Journal of Mechanical Design 130 (3) (2008) 031102–1–8.

[7] R. Kohavi, G. John, Wrappers for feature selection, Artificial Intelligence 97 (1997) 273–324.

[8] H. Stoppiglia, G. Dreyfus, R. Dubois, Y. Oussar, Ranking a random feature for variable and feature selection, Journal of Machine Learning Research 3 (2003) 1399–1414.

[9] I. Rivals, L. Personnaz, Mlps (mono-layer polynomials and multi-layer perceptrons) for non-linear modeling, Journal of Machine Learning Research 3 (2003) 1383–1398.

[10] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (2002) 389–422.

[11] J. de Brabanter, LS-SVM regression modelling and its applications, Ph.D. thesis, Katholieke Universitiet Leuven (2004).

[12] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Machine Learning 3 (2003) 1157–1182.

[13] V. R. Joseph, J. D. Delaney, Functionally induced priors for the analysis of experiments, Technometrics 49 (2007) 1–11.

[14] C. Wu, M. Hamada, Experiments: Planning, Analysis, and Parameter Design Optimization, Wiley, New York, 2000.

[15] V. R. Joseph, A bayesian approach to the design and analysis of fractionated experiments, Technometrics 48 (2) (2006) 221–229.

[16] M. Hamada, C. F. J. Wu, Analysis of designed experiments with complex aliasing, Quality Technology 24 (3) (1992) 130–137.

[17] D. J. Toal, N. W. Bressloff, A. J. Keane, Kriging hyperparameter tuning strategies, AIAA Journal 46 (5) (2008) 1240–1252.

[18] P. Moscato, C. Cotta, Memetic algorithms, in: Optimization Techniques in Engineering, Springer-Verlag, 2004, pp. 53–85.

[19] D. J. J. Toal, A. I. Forrester, N. W. Bresslof, A. J. Keane, C. Holden, An adjoint for likelihood maximization, Royal Society pre-print.

[20] S. N. Lophaven, H. B. Nielsen, J. Søndergaard, Aspects of the matlab toolbox DACE, Tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby (2002).

[21] A. Forrester, A. Sobester, A. Keane, Engineering Design Via Surrogate Modelling: A Practical Guide, Wiley, Chichester, 2008.

[22] E. Pardos-Igúzquiza, P. Dowd, Empirical maximum likelihood kriging: The general case, Mathematical Geology 37 (5) (2005) 477–492.

[23] S. Shan, G. Wang, Development of adaptive rbf-hdmr model for approximating high dimensional problems, in: Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2009, San Diego, USA, DETC2009-86531, 2009.

[24] A. J. Keane, P. B. Nair, Computational approaches for Aerospace Design, The Pursuit of Excellence, John Wiley & Sons Ltd., The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2005.

[25] A. I. J. Forrester, D. R. Jones, Global optimization of deceptive functions with sparse sampling, in: Proceedings of the AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA, 2008, p. 15.

[26] B. Worley, Deterministic uncertainty analysis, Nuclear Science and Engineering (ASCE) 94 (1986) 180.

[27] V. R. Joseph, Y. Hung, Orthogonal-Maximin Latin hypercube designs, Statistica Sinica 18 (2008) 171–186.

[28] J. Pineros-Garcet, M. Vanclooster, A. Tiktak, D. D. Nie, A. Jones, Methodological approach for evaluating higher tier pec groundwater scenarios supporting the prediction of environmental concentrations of pesticides at the pan european scale, in: A. D. Re, M. Trevisan, E. Capri (Eds.), Pesticide chemistry symposium. Pesticide in air, plant soil and water system, 2003, pp. 951–962.

[29] A. Tiktak, D. D. Nie, J. Pineros-Garcet, A. Jones, M. Vanclooster, Assessing the pesticide leaching risk at the pan European level: the EuroPEARL approach, Journal of Hydrology 289 (2004) 222–238.

[30] A. Tiktak, J. Boesten, A. V. der Linden, M. Vanclooster, Mapping the vulnerability of european groundwater to leaching of pesticides with a process based meta-model of europearl, Journal of Environmental Quality 35 (2006) 1213–1226.

[31] E. Dam, B. van Husslage, D. den Hertog, J. Melissen, Maximin Latin hypercube designs in two dimensions, Operations Research 55 (1) (2007) 158–169.