

ooDACE Toolbox: A Flexible Object-Oriented Kriging Implementation

Ivo Couckuyt*

Tom Dhaene

Piet Demeester

Ghent University - iMinds

Department of Information Technology (INTEC)

Gaston Crommenlaan 8

9050 Gent, Belgium

IVO.COUCKUYT@UGENT.BE

TOM.DHAENE@UGENT.BE

PIET.DEMEESTER@UGENT.BE

Editor: Mikio Braun

Abstract

When analyzing data from computationally expensive simulation codes, surrogate modeling methods are firmly established as facilitators for design space exploration, sensitivity analysis, visualization and optimization. Kriging is a popular surrogate modeling technique used for the Design and Analysis of Computer Experiments (DACE). Hence, the past decade Kriging has been the subject of extensive research and many extensions have been proposed, e.g., co-Kriging, stochastic Kriging, blind Kriging, etc. However, few Kriging implementations are publicly available and tailored towards scientists and engineers. Furthermore, no Kriging toolbox exists that unifies several Kriging flavors. This paper addresses this need by presenting an efficient object-oriented Kriging implementation and several Kriging extensions, providing a flexible and easily extendable framework to test and implement new Kriging flavors while reusing as much code as possible.

Keywords: Kriging, Gaussian process, co-Kriging, blind Kriging, surrogate modeling, metamodeling, DACE

1. Introduction

This paper is concerned with efficiently solving complex, computational expensive problems using surrogate modeling techniques (Gorissen et al., 2010). Surrogate models, also known as metamodels, are cheap approximation models for computational expensive (black-box) simulations. Surrogate modeling techniques are well-suited to handle, for example, expensive finite element (FE) simulations and computational fluid dynamic (CFD) simulations.

Kriging is a popular surrogate model type to approximate deterministic noise-free data. First conceived by Danie Krige in geostatistics and later introduced for the Design and Analysis of Computer Experiments (DACE) by Sacks et al. (1989), these Gaussian process (Rasmussen and Williams, 2006) based surrogate models are compact and cheap to evaluate, and have proven to be very useful for tasks such as optimization, design space exploration, visualization, prototyping, and sensitivity analysis (Viana et al., 2014). Note

*. Ivo Couckuyt is a post-doctoral research fellow of FWO-Vlaanderen.

that Kriging surrogate models are primarily known as Gaussian processes in the machine learning community. Except for the utilized terminology there is no difference between the terms and associated methodologies.

While Kriging is a popular surrogate model type, not many publicly available, easy-to-use Kriging implementations exist. Many Kriging implementations are outdated and often limited to one specific type of Kriging. Perhaps the most well-known Kriging toolbox is the DACE toolbox¹ of Lophaven et al. (2002), but, unfortunately, the toolbox has not been updated for some time and only the standard Kriging model is provided. Other freely available Kriging codes include: stochastic Kriging (Staum, 2009),² DiceKriging,³ Gaussian processes for Machine Learning (Rasmussen and Nickisch, 2010) (GPML),⁴ demo code provided with Forrester et al. (2008),⁵ and the Matlab Krigeage toolbox.⁶

This paper addresses this need by presenting an object-oriented Kriging implementation and several Kriging extensions, providing a flexible and easily extendable framework to test and implement new Kriging flavors while reusing as much code as possible.

2. ooDACE Toolbox

The ooDACE toolbox is an object-oriented Matlab toolbox implementing a variety of Kriging flavors and extensions. The most important features and Kriging flavors include:

- Simple Kriging, ordinary Kriging, universal Kriging, stochastic Kriging (regression Kriging), blind- and co-Kriging.
- Derivatives of the prediction and prediction variance.
- Flexible hyperparameter optimization.
- Useful utilities include: cross-validation, integrated mean squared error, empirical variogram plot, debug plot of the likelihood surface, robustness-criterion value, etc.
- Proper object-oriented design (compatible interface with the DACE toolbox¹ is available).

Documentation of the ooDACE toolbox is provided in the form of a getting started guide (for users), a wiki⁷ and doxygen documentation⁸ (for developers and more advanced users). In addition, the code is well-documented, providing references to research papers where appropriate. A quick-start demo script is provided with five surrogate modeling use cases, as well as script to run a suite of regression tests.

A simplified UML class diagram, showing only the most important public operations, of the toolbox is shown in Figure 1. The toolbox is designed with efficiency and flexibility in mind. The process of constructing (and predicting) a Kriging model is decomposed in several smaller, logical steps, e.g., constructing the correlation matrix, constructing the

1. The DACE toolbox can be downloaded at <http://www2.imm.dtu.dk/~hbn/dace/>.
 2. The stochastic Kriging toolbox can be downloaded at <http://stochasticKriging.net/>.
 3. The DiceKriging toolbox can be downloaded at <http://cran.r-project.org/web/packages/DiceKriging/index.html>.
 4. The GPML toolbox can be downloaded at <http://mloss.org/software/view/263/>.
 5. Demo code of Kriging can be downloaded at <http://www.wiley.com//legacy/wileychi/forrester/>.
 6. The Krigeage toolbox can be downloaded at <http://globec.who.edu/software/kriging/>.
 7. The wiki documentation of the ooDACE toolbox is found at http://sumowiki.intec.ugent.be/index.php/ooDACE:ooDACE_toolbox.
 8. The doxygen documentation of the ooDACE toolbox is found at <http://sumo.intec.ugent.be/buildbot/ooDACE/doc/>.

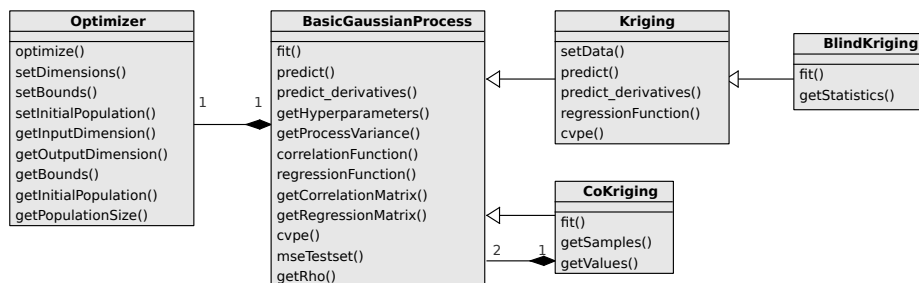


Figure 1: Class diagram of the ooDACE toolbox.

regression matrix, updating the model, optimizing the parameters, etc. These steps are linked together by higher-level steps, e.g., fitting the Kriging model and making predictions. The basic steps needed for Kriging are implemented as (protected) operations in the BASICGAUSSIANPROCESS superclass. Implementing a new Kriging type, or extending an existing one, is now done by subclassing the Kriging class of your choice and inheriting the (protected) methods that need to be reimplemented. Similarly, to implement a new hyperparameter optimization strategy it suffices to create a new class inherited from the OPTIMIZER class.

To assess the performance of the ooDACE toolbox a comparison between the ooDACE toolbox and the DACE toolbox¹ is performed using the 2D Branin function. To that end, 20 data sets of increasing size are constructed, each drawn from a uniform random distribution. The number of observations ranges from 10 to 200 samples with steps of 10 samples. For each data set, a DACE toolbox¹ model, a ooDACE ordinary Kriging and a ooDACE blind Kriging model have been constructed and the accuracy is measured on a dense test set using the Average Euclidean Error (AEE). Moreover, each test is repeated 1000 times to remove any random factor, hence the average accuracy of all repetitions is used. Results are shown in Figure 2a. Clearly, the ordinary Kriging model of the ooDACE toolbox consistently outperforms the DACE toolbox for any given sample size, mostly due to a better hyperparameter optimization, while the blind Kriging model is able to improve the accuracy even more.

3. Applications

The ooDACE Toolbox has already been applied successfully to a wide range of problems, e.g., optimization of a textile antenna (Couckuyt et al., 2010), identification of the elasticity of the middle-ear drum (Aernouts et al., 2010), etc.

In sum, the ooDACE toolbox aims to provide a modern, up to date Kriging framework catered to scientists and engineers. Usage instructions, design documentation, and stable releases can be found at <http://sumo.intec.ugent.be/?q=ooDACE>.

References

J. Aernouts, I. Couckuyt, K. Crombecq, and J.J.J. Dirckx. Elastic characterization of membranes with a complex shape using point indentation measurements and inverse

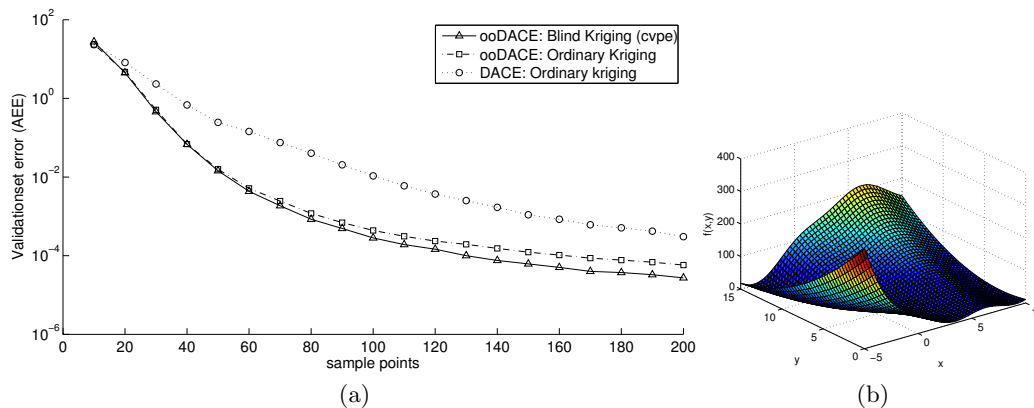


Figure 2: (a) Evolution of the average AEE versus the number of samples (Branin function).
 (b) Landscape plot of the Branin function.

modelling. *International Journal of Engineering Science*, 48:599–611, 2010.

- I. Couckuyt, F. Declercq, T. Dhaene, and H. Rogier. Surrogate-based infill optimization applied to electromagnetic problems. *Journal of RF and Microwave Computer-Aided Engineering: Advances in Design Optimization of Microwave/RF Circuits and Systems*, 20(5):492–501, 2010.
- A. Forrester, A. Sobester, and A. Keane. *Engineering Design Via Surrogate Modelling: A Practical Guide*. Wiley, Chichester, 2008.
- D. Gorissen, K. Crombecq, I. Couckuyt, P. Demeester, and T. Dhaene. A surrogate modeling and adaptive sampling toolbox for computer based design. *Journal of Machine Learning Research*, 11:2051–2055, 2010. URL <http://sumo.intec.ugent.be/>.
- S. N. Lophaven, H. B. Nielsen, and J. Søndergaard. Aspects of the Matlab toolbox DACE. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2002.
- C. E. Rasmussen and H. Nickisch. Gaussian processes for machine learning (GPML) toolbox. *Journal of Machine Learning Research*, 11:3011–3015, 2010.
- C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- J. Sacks, W. J. Welch, T.J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.
- J. Staum. Better simulation metamodeling: The why, what, and how of stochastic Kriging. In *Proceedings of the Winter Simulation Conference*, 2009.
- F. A. C. Viana, T. W. Simpson, V. Balabanov, and V. Toropov. Metamodeling in multi-disciplinary design optimization: How far have we really come? *AIAA Journal*, 52(4):670–690, 2014.