Procedia Computer Science

Volume 51, 2015, Pages 2708–2713

ICCS 2015 International Conference On Computational Science

A Matlab toolbox for Kriging metamodelling

Selvakumar Ulaganathan[1]*, Ivo Couckuyt[2], Dirk Deschrijver[3], Eric Laermans[4], and Tom Dhaene[5]

Ghent University - iMinds, Department of Information Technology (INTEC), Gaston Crommenlaan 8, 9050 Gent, Belgium

selvakumar.ulaganathan@ugent.be, ivo.couckuyt@ugent.be, dirk.deschrijver@ugent.be, eric.laermans@ugent.be, tom.dhaene@ugent.be

Abstract

Metamodelling offers an efficient way to imitate the behaviour of computationally expensive simulators. Kriging based metamodels are popular in approximating computation-intensive simulations of deterministic nature. Irrespective of the existence of various variants of Kriging in the literature, only a handful of Kriging implementations are publicly available and most, if not all, free libraries only provide the standard Kriging metamodel. ooDACE toolbox offers a robust, flexible and easily extendable framework where various Kriging variants are implemented in an object-oriented fashion under a single platform. This paper presents an incremental update of the ooDACE toolbox introducing an implementation of Gradient Enhanced Kriging which has been tested and validated on several engineering problems.

Keywords: Metamodelling, Kriging, Gradient Enhanced Kriging, ooDACE

1 Introduction

Constructing approximation models or metamodels of computation-intensive simulators to perform routine activities, such as optimisation, sensitivity analysis, design space exploration, etc. is a well-known approach to ease the computational burden. Metamodelling offers a proven efficient way to imitate the behaviour of computationally expensive simulators. An overview of various metamodelling techniques is given in [5], [15] and [21]. Kriging based metamodels are popular in approximating computer generated data which are deterministic in nature. Kriging for design and analysis of computer experiments was introduced by [13] and is further used in computer aided applications by various researchers [6, 12, 14].

In general, the primary motivation of any metamodelling approach is to accurately mimic the behaviour of a computation-intensive simulator over a design space of interest with as few training data as possible. Hence, the ideas of incorporating (computationally cheap) secondary

*Corresponding author

 Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015
© The Authors. Published by Elsevier B.V.

doi:10.1016/j.procs.2015.05.395

data in the form of gradients, data of different degrees of accuracy, etc. are very attractive. In this context, gradient incorporation in Kriging, known as Gradient Enhanced Kriging (GEK), can be found in the literature extensively [2, 7, 8, 10]. However, not many publicly available implementations exist. Moreover, many implementations are limited to standard Kriging or one specific type of Kriging. For example, the popular Matlab[1] DACE toolbox [9] provides only the standard Kriging model and does not deal with gradient incorporation in Kriging. Other publicly available Kriging implementations where none of them deal with gradient incorporation in Kriging include: Stochastic Kriging [16], DiceKriging[2], Gaussian processes for Machine Learning [11], a Small Toolbox for Kriging (STK)[3] and the Matlab Krigeage toolbox[4]. However, some publicly available implementations of GEK such as beta version of Bayesian formulation based GEK code[5], Fortran Kriging (ForK) library[6], etc. do exist. But, unfortunately, none of them provide a flexible framework to implement and test new Kriging algorithms. In this context, this paper presents a Matlab toolbox, known as ooDACE[7], which offers a robust and easily extendable object-oriented framework where various Kriging variants are implemented.

2 ooDACE Toolbox

The ooDACE Toolbox (Design and Analysis of Computer Experiments) is designed to meet the needs of researchers by providing a flexible and easily extendable Matlab implementation of various Kriging variants and is well-suited to test and benchmark new Kriging algorithms. The most significant features of the toolbox include:

- Kriging metamodels:
  - **Kriging (or Gaussian process)** : models single-fidelity data.
  - **Simple**, **Ordinary**, **Universal Kriging** : models single-fidelity data using a Gaussian process with the mean as known constant, unknown constant or polynomial function, respectively.
  - **Co-Kriging** : models multi-fidelity data.
  - **Blind Kriging** : models single-fidelity data using a Gaussian process where the mean is a polynomial function that is automatically identified.
  - **Stochastic Kriging** : models data of stochastic nature.
  - **GradientKriging** : models single-fidelity data of function and gradient values.
- Efficient computation of derivatives of the prediction model.
- Efficient hyper-parameter optimisation.
- Proper Object Oriented (OO) design.

[1]MATLAB, The MathWorks, Inc., Natick, Massachusetts, USA
[2]http://cran.r-project.org/web/packages/DiceKriging/index.html
[3]http://sourceforge.net/projects/kriging/files/stk/
[4]http://globec.whoi.edu/software/kriging/
[5]https://aerodynamics.lr.tudelft.nl/∼bayesiancomputing/
[6]http://w3.uwyo.edu/∼blockwoo/ForKlib/krigingwrapperGEK_8f90.html
[7]http://sumo.intec.ugent.be/?q=ooDACE

- Useful utilities include: cross-validation, integrated mean squared error, etc.

A simplified UML class diagram that shows only the most important public operations of the toolbox, is shown in Figure 1. The basic functionalities required for building and evaluating a Kriging metamodel are implemented in the `GaussianProcess` superclass. Similarly, a new hyper-parameter optimisation technique can be implemented by inheriting from the `Optimizer` class. New Kriging variants (e.g., `CoKriging`, `GradientKriging`, `BlindKriging`) can be implemented by forming subclass that inherits the common functionalities of the `GaussianProcess` superclass. In those subclasses, one can extend or override the existing methods to offer additional functionality (e.g., a method to compute the GEK correlation matrix in `GradientKriging`).

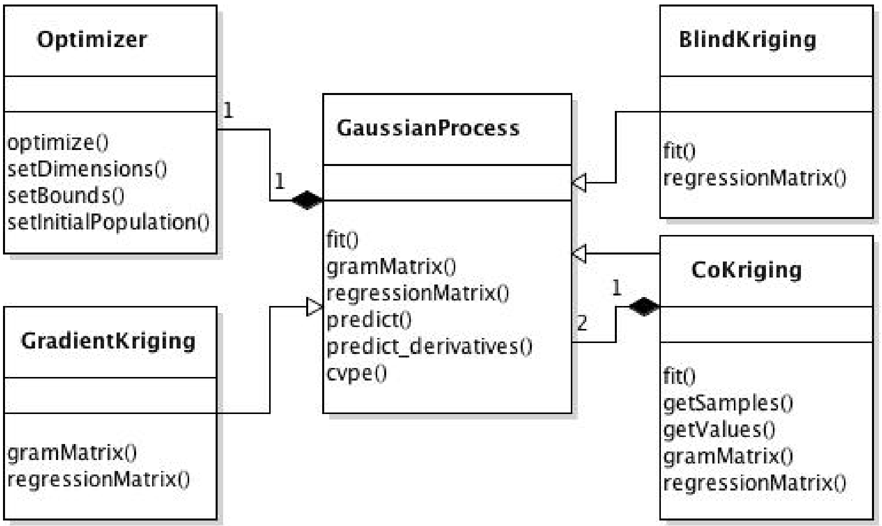
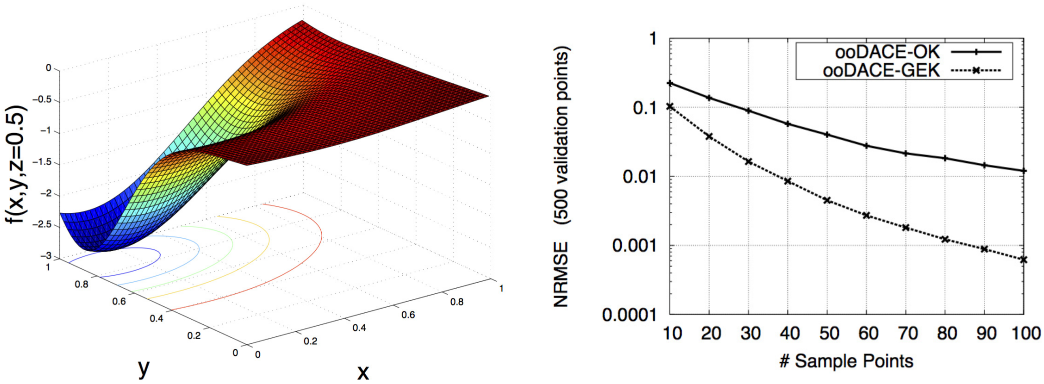


Figure 1: The UML class diagram of the ooDACE toolbox.

3 Applications

The performance of GEK in ooDACE toolbox is assessed with the modelling of the 3D Hartmann function. Figure 2 compares the evolution of (averaged) Normalized Root Mean Square Error (NRMSE) of Ordinary Kriging (OK) and GEK as a function of number of training samples. For each number of training samples, 50 different data sets where the samples are chosen randomly in the design space are constructed. Thus, fitting and accuracy assessment of the OK and the GEK models are repeated 50 times for each training sample size. Then the error metric is calculated by making predictions at 500 randomly chosen validation points and the resulting score is averaged over 50 runs. In Figure 2, it is important to note that GEK contains additional gradient data at any given sample size. For example, at the sample size of 10, OK contains only 10 function values whereas GEK contains 10 function values and 30 additional gradient values. GEK can reduce the amount of function values in the training data set by more than 50% and still reach a similar accuracy level as OK, due to the inclusion of gradient values. It was shown in [20] that the use of gradient values in GEK results in an improved hyper-parameter estimation.



(a) Slice of the actual function

(b) Evolution of averaged NRMSE

Figure 2: Hartmann 3D function. At every sample set, GEK incorporates additional data in the form of gradients in all the dimensions.

The performance of the ooDACE toolbox is also assessed with various analytical (e.g., Branin, Peaks, Ackley, Sphere, Hartmann, Rosenbrock, etc. [4, 17, 18]) and real-life (e.g., identification of the elasticity of the middle-ear drum [1], optimisation of a textile antenna [3], modelling of wall displacement in an artery [17], etc.) functions of varying dimensionality (2D-20D). One can refer to [17, 18, 19, 20] for more information on the performance analysis of GEK and other Kriging flavours available in the ooDACE toolbox.

4 Conclusions

This paper presented ooDACE toolbox, a free object-oriented Matlab toolbox for building metamodels using various Kriging variants. By providing a researcher-friendly framework, the vision of ooDACE is to offer a unique platform catered to scientists and engineers where ease of use, transparency and extensibility exist together. Further, with the recent implementation of Gradient Enhanced Kriging, the ooDACE toolbox aims to provide an up to date Kriging based metamodelling framework to a large scientific community. Furthermore, a strong emphasis is placed on implementation and code clarity, documentation, demo scripts, thorough usage instructions and stable releases.

5 Acknowledgments

This research has been funded by the Interuniversity Attraction Poles Programme BESTCOM initiated by the Belgian Science Policy Office. Additionally, this research has been supported by the Fund for Scientific Research in Flanders (FWO-Vlaanderen). Ivo Couckuyt and Dirk Deschrijver are post-doctoral research fellows of the Research Foundation Flanders (FWO-Vlaanderen).

References

[1] Jef Aernouts, Ivo Couckuyt, Karel Crombecq, and Joris J.J. Dirckx. Elastic characterization of membranes with a complex shape using point indentation measurements and inverse modelling. *International Journal of Engineering Science*, 48(6):599 – 611, 2010.

[2] Hyoung-Seog Chung and Juan J. Alonso. Using gradients to construct cokriging approximation models for high-dimensional design optimization problems. In *Problems, 40th AIAA Aerospace Sciences Meeting and Exhibit, AIAA*, pages 2002–0317, Reno, NV, January 2002.

[3] Ivo Couckuyt, Frederick Declercq, Tom Dhaene, Hendrik Rogier, and Luc Knockaert. Surrogate-based infill optimization applied to electromagnetic problems. *International Journal of RF and Microwave computer-aided Engineering*, 20(5):492–501, 2010.

[4] Ivo Couckuyt, Tom Dhaene, and Piet Demeester. ooDACE toolbox: A flexible object-oriented kriging implementation. *Journal of Machine Learning Research*, 15:3183–3186, 2014.

[5] Dirk Gorissen, Ivo Couckuyt, Piet Demeester, Tom Dhaene, and Karel Crombecq. A surrogate modeling and adaptive sampling toolbox for computer based design. *Journal of Machine Learning Research*, 11:2051–2055, 2010.

[6] J.P.C. Kleijnen. Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3):707–716, January 2009.

[7] Julien Laurenceau and P Sagaut. Building efficient response surfaces of aerodynamic functions with kriging and cokriging. *AIAA*, 46(2):498–507, January 2008.

[8] Weiyu Liu. *Development of Gradient-Enhanced Kriging approximations for multidisciplinary design optimisation*. PhD thesis, University of Notre Dame, Notre Dame, Indiana, 2003.

[9] 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.

[10] Max D. Morris, Toby J. Mitchell, and Donald Ylvisaker. Bayesian design and analysis of computer experiments: Use of gradients in surface prediction. *Technometrics*, 35(3):243–255, August 1993.

[11] CE. Rasmussen and H. Nickisch. Gaussian processes for machine learning (GPML) toolbox. *Journal of Machine Learning Research*, 11:3011–3015, November 2010.

[12] Jerome Sacks, Susannah B. Schiller, and William J. Welch. Designs for computer experiments. *Technometrics*, 31(1):41–47, February 1989.

[13] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, November 1989.

[14] Wenze Shao, Haisong Deng, Yizhong Ma, and Zhuihui Wei. Extended Gaussian kriging for computer experiments in engineering design. *Engineering with Computers*, 28(2):161–178, 2012.

[15] T.W. Simpson, J.D. Poplinski, P. N. Koch, and J.K. Allen. Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, 17(2):129–150, 2001.

[16] J. Staum. Better simulation metamodeling: The why, what, and how of stochastic kriging. In *Simulation Conference (WSC), Proceedings of the 2009 Winter*, pages 119–133, Dec 2009.

[17] Selvakumar Ulaganathan, Ivo Couckuyt, Tom Dhaene, Joris Degroote, and Eric Laermans. Performance study of gradient enhanced kriging. *Engineering with computers*, Accepted paper, 2015.

[18] Selvakumar Ulaganathan, Ivo Couckuyt, Tom Dhaene, Eric Laermans, and Joris Degroote. On the use of gradients in kriging surrogate models. In *Proceedings of the 2014 Winter Simulation Conference*, WSC '14, pages 2692–2701, Piscataway, NJ, USA, 2014. IEEE Press.

[19] Selvakumar Ulaganathan, Ivo Couckuyt, Francesco Ferranti, Tom Dhaene, and Eric Laermans. Variable-fidelity surrogate modelling with kriging. *World Academy of Science, Engineering and Technology*, 2(9):129, 2014.

[20] Selvakumar Ulaganathan, Ivo Couckuyt, Francesco Ferranti, Eric Laermans, and Tom Dhaene.

Performance study of multi-fidelity gradient enhanced kriging. *Structural and Multidisciplinary Optimization*, DOI: 10.1007/s00158-014-1192-x, 2014.

[21] G. Gary Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4):370–380, April 2006.