Journal of Machine Learning Research () Submitted 03/10; Revised 06/10; Published

A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design

Dirk Gorissen DIRK.GORISSEN@UGENT.BE
Ivo Couckuyt IVO.COUCKUYT@UGENT.BE
Piet Demeester PIET.DEMEESTER@UGENT.BE
Tom Dhaene TOM.DHAENE@UGENT.BE
Ghent University - IBBT
Department of Information Technology (INTEC)
Gaston Crommenlaan 8 bus 201
9050 Gent, Belgium

Karel Crombecq KAREL.CROMBECQ@UA.AC.BE
University of Antwerp
Department of Maths and Computer Science
Middelheimlaan 1
2020 Antwerpen, Belgium

Editor: Cheng Soon Ong

Abstract

An exceedingly large number of scientific and engineering fields are confronted with the need for computer simulations to study complex, real world phenomena or solve challenging design problems. However, due to the computational cost of these high fidelity simulations, the use of neural networks, kernel methods, and other surrogate modeling techniques have become indispensable. Surrogate models are compact and cheap to evaluate, and have proven very useful for tasks such as optimization, design space exploration, prototyping, and sensitivity analysis. Consequently, in many fields there is great interest in tools and techniques that facilitate the construction of such regression models, while minimizing the computational cost and maximizing model accuracy. This paper presents a mature, flexible, and adaptive machine learning toolkit for regression modeling and active learning to tackle these issues. The toolkit brings together algorithms for data fitting, model selection, sample selection (active learning), hyperparameter optimization, and distributed computing in order to empower a domain expert to efficiently generate an accurate model for the problem or data at hand.

Keywords: surrogate modeling, metamodeling, function approximation, model selection, adaptive sampling, active learning, distributed computing

1. Background and Motivation

In many science and engineering problems researchers make heavy use of computer simulation codes in order to replace expensive physical experiments and improve the quality and performance of engineered products and devices. Such simulation activities are collectively referred to as computational science/engineering. Unfortunately, while allowing scientists more flexibility to study phenomena under controlled conditions, computer simulations require a substantial investment of computation time. One simulation may take many minutes, hours, days or even weeks, quickly rendering parameter studies impractical (Forrester et al., 2008; Simpson et al., 2008).

Of the different ways to deal with this problem, this paper is concerned with the construction of simpler approximation models to predict the system performance and develop a relationship between

the system inputs and outputs. When properly constructed, these approximation models mimic the behavior of the simulation accurately while being computationally cheap(er) to evaluate. Different approximation methods exist, each with their relative merits. This work concentrates on the use of data-driven, global approximations using compact surrogate models (also known as metamodels, replacement models, or response surface models). Examples include: rational functions, Kriging models, Artificial Neural Networks (ANN), splines, and Support Vector Machines (SVM). Once such a global approximation is available it is of great use for gaining insight into the behavior of the underlying system. The surrogate may be easily queried, optimized, visualized, and seamlessly integrated into CAD/CAE software packages.

The challenge is thus how to generate an approximation model that is as accurate as possible over the *complete* domain of interest while minimizing the simulation cost. Solving this challenge involves multiple sub-problems that must be addressed: how to interface with the simulation code, how to run simulations (locally, or on a cluster or cloud), which model type to approximate the data with and how to set the model complexity (e.g., topology of a neural network), how to estimate the model quality and ensure the domain expert trusts the model, how to decide which simulations to run (data collection), etc. The data collection aspect is worth emphasizing. Since data is computationally expensive to obtain and the optimal data distribution is not known up front, data points should be selected iteratively, there where the information gain will be the greatest. A sampling function is needed that minimizes the number of sample points selected in each iteration, yet maximizes the information gain of each iteration step. This process is called adaptive sampling but is also known as active learning, or sequential design.

There is a complex dependency web between these different options and dealing with these dependencies is non-trivial, particularly for a domain expert for whom the surrogate model is just an intermediate step towards solving a larger, more important problem. Few domain experts will be experts in the intricacies of efficient sampling and modeling strategies. Their primary concern is obtaining an accurate replacement metamodel for their problem as fast as possible and with minimal overhead (Gorissen et al., 2009d). As a result these choices are often made in a pragmatic, sometimes even ad-hoc, manner.

This paper discusses an advanced, and integrated software framework that provides a flexible and rigorous means to tackle such problems. This work lies at the intersection of Machine Learning/AI, Modeling and Simulation, and Distributed Computing. The methods developed are applicable to any domain where a cheap, accurate, approximation is needed to replace some expensive reference model. Our experience has been that the availability of such a framework can facilitate the transfer of knowledge from surrogate modeling researchers and lower the barrier of entry for domain experts.

2. SUMO Toolbox

The platform in question is the Matlab SUrrogate MOdeling (SUMO) Toolbox, illustrated in figure 1. Given a simulation engine (Fluent, Cadence, Abaqus, HFSS, etc.) or other data source (data set, Matlab script, Java class, etc.), the toolbox drives the data source to produce a surrogate model within the time and accuracy constraints set by the user.

The SUMO Toolbox adopts a microkernel design philosophy with many different plugins available for each of the different sub-problems[1]: model types (rational functions, Kriging, splines, SVM, ANN, etc.), hyperparameter optimization algorithms (Particle Swarm Optimization, Efficient Global Optimization, simulated annealing, Genetic Algorithm, etc.), model selection algorithms (cross validation, AIC, Leave-out set, etc.), sample selection (random, error based, density based, hybrid, etc.), Design of Experiments (Latin hypercube, Box-Bhenken, etc.), and sample evaluation methods (local, on a cluster or grid). The behavior of each software component is configurable through a central XML file and components can easily be added, removed or replaced by custom implementa-

1. The full list of plugins and features can be found at `http://www.sumowiki.intec.ugent.be`

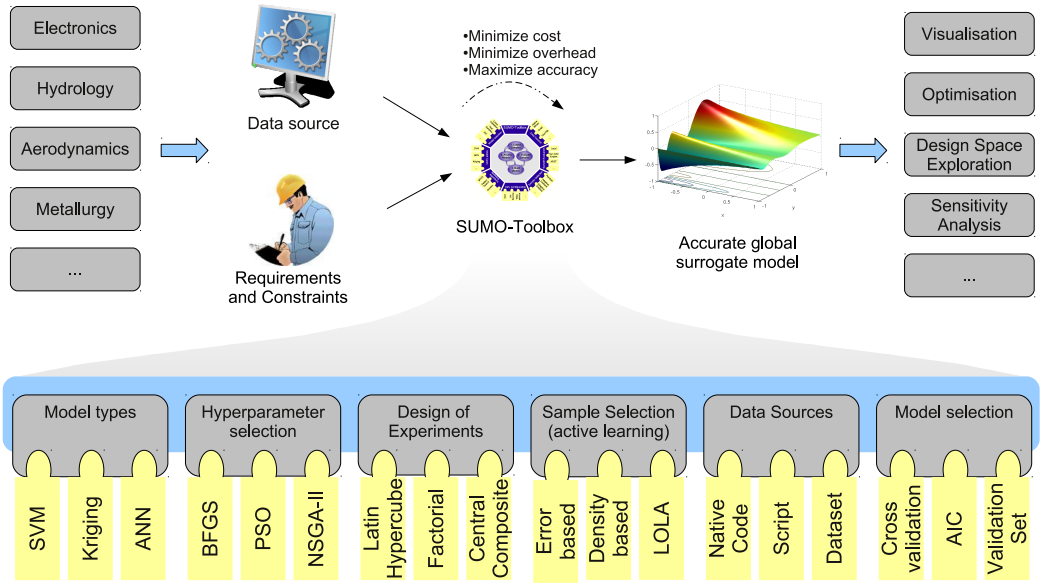

Figure 1: The SUMO Toolbox is a flexible framework for accurate global surrogate modeling and adaptive sampling (active learning). It features a rich set of plugins, is applicable to a wide range of domains, and can be applied in an autonomous, black-box fashion, or under full manual control. Written in Matlab and Java it is fully cross platform and comes with a large (60+) number of example problems.

tions. In addition the toolbox provides 'meta' plugins. For example to automatically select the best model type for a given problem (Gorissen et al., 2009d) or to use multiple model selection or sample selection criteria in concert (Gorissen et al., 2010).

Furthermore, there is built-in support for high performance computing. On the modeling side, the model generation process can take full advantage of multi-core CPUs and even of a complete cluster or grid. This can result in significant speedups for model types where the fitting process can be expensive (e.g., neural networks). Likewise, sample evaluation (simulation) can occur locally (with the option to take advantage of multi-core architectures) or on a separate compute cluster or grid (possibly accessed through a remote head-node). All interfacing with the grid middleware (submission, job monitoring, rescheduling of failed/lost simulation points, etc.) is handled transparently and automatically (see (Gorissen et al., 2009c) for more details). Also, the sample evaluation component runs in parallel with the other components (non-blocking) and not sequentially. This allows for an optimal use of computational resources.

In addition the SUMO Toolbox contains extensive logging and profiling capabilities so that the modeling process can easily be tracked and the modeling decisions understood. Once a final model has been generated, a GUI tool is available to visually explore the model (including derivatives and prediction uncertainty), assess its quality, and export it for use in other software tools.

3. Applications

The SUMO Toolbox has already been applied successfully to a very wide range of applications, including RF circuit block modeling (Gorissen et al., 2009b), hydrological modeling (Couckuyt et al., 2009), Electronic Packaging (Zhu and Franzon, 2009), aerodynamic modeling (Gorissen et al., 2009a), process engineering (Stephens et al., 2009), and automotive data modeling (Gorissen et al., 2010).

Besides global modeling capabilities, the SUMO Toolbox also includes a powerful optimization framework based on the Efficient Global Optimization framework developed by Jones (Jones et al., 1998). As of version 6.1, the toolbox also contains an example of how the framework can also be applied to solve classification problems.

In sum, the goal of the toolbox is to fill the void in machine learning software when it comes to the challenging, costly, real-valued, problems faced in computational engineering. The toolbox is in use successfully at various institutions and we are continuously refining and extending the set of available plugins as the number of applications increase. Usage instructions, design documentation, and stable releases for all major platforms can be found at `http://www.sumo.intec.ugent.be`.

References

I. Couckuyt, D. Gorissen, H. Rouhani, E. Laermans, and T. Dhaene. Evolutionary regression modeling with active learning: An application to rainfall runoff modeling. In *International Conference on Adaptive and Natural Computing Algorithms*, volume LNCS 5495, pages 548–558, Sep. 2009.

A. Forrester, A. Sobester, and A. Keane. *Engineering Design Via Surrogate Modelling: A Practical Guide*. Wiley, 2008.

D. Gorissen, K. Crombecq, I. Couckuyt, and T. Dhaene. *Foundations of Computational Intelligence, Volume 1: Learning and Approximation: Theoretical Foundations and Applications*, volume 201, chapter Automatic Approximation of Expensive Functions with Active Learning, pages 35–62. Springer Verlag, Series Studies in Computational Intelligence, 2009a.

D. Gorissen, L. De Tommasi, K. Crombecq, and T. Dhaene. Sequential modeling of a low noise amplifier with neural networks and active learning. *Neural Computing and Applications*, 18(5): 485–494, Jun. 2009b.

D. Gorissen, T. Dhaene, P. Demeester, and J. Broeckhove. *Handbook of Research on Grid Technologies and Utility Computing: Concepts for Managing Large-Scale Applications*, chapter Grid enabled surrogate modeling, pages 249–258. IGI Global, May 2009c.

D. Gorissen, T. Dhaene, and F. DeTurck. Evolutionary model type selection for global surrogate modeling. *Journal of Machine Learning Research*, 10:2039–2078, 2009d.

D. Gorissen, I. Couckuyt, E. Laermans, and T. Dhaene. Multiobjective global surrogate modeling,dealing with the 5-percent problem. *Engineering with Computers*, 26(1):81–89, Jan. 2010.

D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, Nov. 1998. ISSN 0925-5001.

T. W. Simpson, V. Toropov, V. Balabanov, and F. A. C. Viana. Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come or not. In *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2008 MAO, Victoria, Canada*, 2008.

D. Stephens, D. Gorissen, and T. Dhaene. Surrogate based sensitivity analysis of process equipment. In *Proc. of 7th International Conference on CFD in the Minerals and Process Industries, CSIRO, Melbourne, Australia*, Dec. 2009.

T. Zhu and P. D. Franzon. Application of surrogate modeling to generate compact and PVT-sensitive IBIS models. In *Proceedings of the 18th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, Oct. 2009.