The SUMO Toolbox: a Tool for Automatic Regression Modeling and Active Learning

Ivo Couckuyt, Dirk Gorissen, Karel Crombecq, Dirk Deschrijver, Tom Dhaene
Department of Information Technology
iMinds-Ghent University
Ghent, Belgium
Email: ivo.couckuyt@ugent.be

***Abstract*—Many complex, real world phenomena are difficult to study directly using controlled experiments. Instead, the use of computer simulations has become commonplace as a feasible alternative. Due to the computational cost of these high fidelity simulations, surrogate models are often employed as a drop-in replacement for the original simulator, in order to reduce evaluation times. In this context, neural networks, kernel methods, and other modeling techniques have become indispensable. Surrogate models have proven to be very useful for tasks such as optimization, design space exploration, visualization, prototyping and sensitivity analysis. We present a fully automated machine learning tool for generating accurate surrogate models, using active learning techniques to minimize the number of simulations and to maximize efficiency.**

I. Introduction

For many problems from science and engineering, it is impractical or impossible to perform experiments in the physical world directly (e.g. airfoil design, earthquake propagation, car crash worthiness). Instead, complex, physics-based simulation codes are used to run experiments on computer hardware. While allowing scientists more flexibility to study phenomena under controlled conditions, computer experiments require a substantial investment of computation time (one simulation may take many minutes, hours, days or even weeks) [1].

As a result, the use of various approximation methods that mimic the behavior of the simulation model as closely as possible has become standard practice. This work concentrates on the use of data-driven, global approximations using compact surrogate models (also known as metamodels or response surface models). Neural networks, Kriging models, and Support Vector Machines (SVM) are often used in this context. Global surrogate modeling is illustrated in Figure 1.

Please note that we are concerned only with global surrogate modeling as opposed to local surrogate modeling. Global surrogate modeling tries to create a model that accurately mimics the original system over the entire design space, with the goal of creating a model that can be safely used as a replacement for the original simulation code. Local surrogate models are often used in optimization to help the optimizer locate an optimum; however, the local surrogate modeling is discarded afterwards, and is not the final goal.

Mathematically, the simulator can be defined as an unknown function $f : \Re^d \rightarrow \mathbb{C}$, mapping a vector of real inputs to a real or complex output. This function can be highly nonlinear and possibly even discontinuous. This unknown function has been sampled at a set of scattered data points $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$, for which the function values $\{f(\mathbf{p}_1), f(\mathbf{p}_2), \ldots, f(\mathbf{p}_n)\}$ are known. In order to approximate the function $\tilde{f}$, a function $f : \Re^d \rightarrow \mathbb{C}$ is chosen from the (possibly) infinite set of candidate approximation functions F.

The quality of this approximation depends on both the choice and exploration of the function space F and the data points P. Ideally, the function f itself would be in the search space F, in which case it is possible to achieve an exact approximation. However, this is rarely the case, due to the complexity of the underlying system. In practice the function $\tilde{f}$ is chosen according to a search strategy through the space F, in order to find the function that most closely resembles the original function, based on some error metric for the data points P [2], [3].

II. Motivation

Creating a sufficiently accurate surrogate model is no easy process, and there are several important design choices that need to be made and problems to overcome in order to develop a robust algorithm. Examples of problems often encountered are choosing the data sampling strategy (active learning), choosing the right model for the problem at hand (model selection), tuning the model parameters (hyperparameter optimization) and balancing between model accuracy and computational cost. Particularly important is the sampling strategy. Since data is computationally expensive to obtain, it can be infeasible to use traditional, one-shot space filling experimental designs such as Latin hypercubes or factorial designs. Data points should be selected iteratively and intelligently at locations where the information gain will be the greatest. This process is called active learning, but is also known as sequential design or adaptive sampling [4].

All these problems result in an overwhelming number of options available to the designer. In practice, it turns out that the designer rarely tries out more than one subset of options, because the search space is just too large to explore manually. All too often, surrogate model construction is done in a one-shot manner. Iterative and adaptive methods, on the other hand, have the potential to produce much more accurate surrogate models at a considerably lower cost (less data points). We present a state-of-the-art machine learning platform that provides an automatic, flexible and extensible framework to tackle such problems: the SUrrogate MOdeling (SUMO) Toolbox.

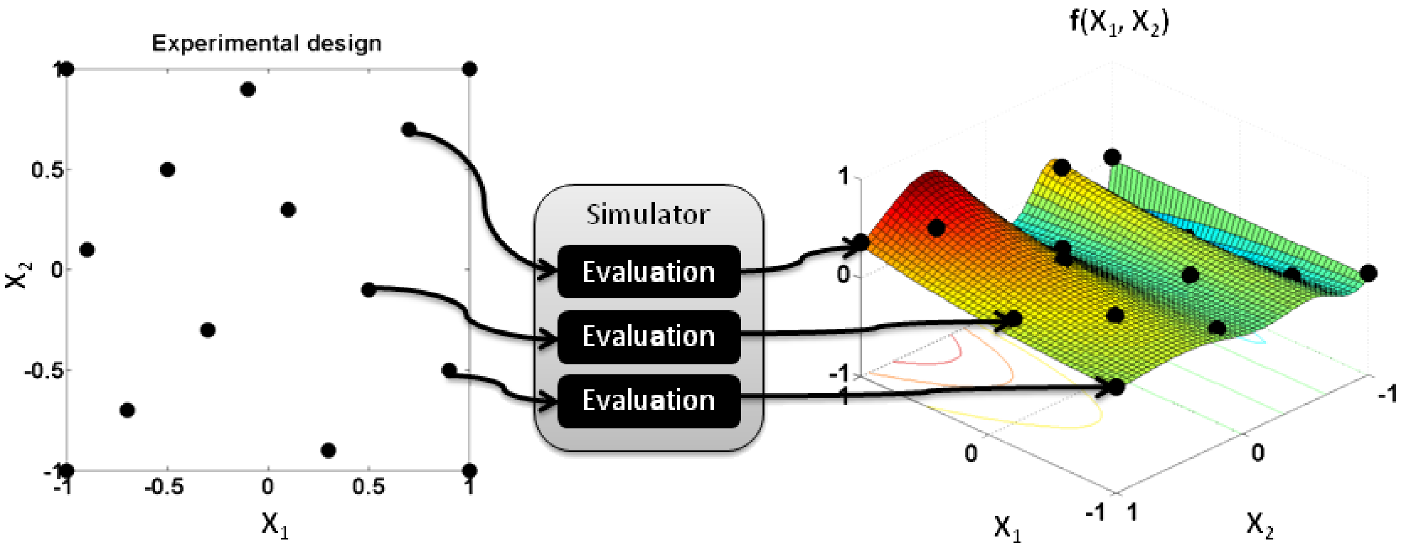

Figure 1. A set of data points is evaluated by the simulator, which outputs a response for every data point. An approximation model (global surrogate model) is fit to the data points.

III. The SUMO Toolbox

The SUMO Toolbox is an adaptive tool that integrates different modeling approaches and implements a fully automated, adaptive global surrogate model construction algorithm. Given a data source (a simulator or a dataset), the toolbox automatically generates a surrogate model within the predefined accuracy and time limits set by the user. Robustness is a primary concern, as different problems require different approaches to achieve optimal results. The toolbox aims to automate as much of the modeling process as possible by tweaking model parameters and selecting samples on the fly to optimize the models for the problem that is being tackled. The latest version of the SUMO Toolbox (v7.0.2) has been released as open software (including all the algorithms mentioned in this paper) and can be downloaded from http://www.sumo.intec.ugent.be, allowing for a full reproduction of all the experiments conducted and published by the authors.

The work-flow of SUMO is illustrated in Figure 2. First, an initial design (typically a sparse Latin hypercube or a fractional design) is generated and evaluated. Then, a set of surrogate models is built, and the accuracy of these models is estimated using a set of measures (for example: cross-validation or an external validation test set). Each model type has several hyperparameters which can be modified, such as the order of numerator and denominator for rational models, number and size of hidden layers in neural networks, smoothness parameters for RBF models, and so on. These parameters are adjusted using a hyperparameter optimization technique, and more models are built until no further improvement can be made by changing the hyperparameters. If the overall desired accuracy has not yet been reached, a call is made to the sequential design routine, which selects a new sample to be evaluated, and the algorithm starts all over again.

To make SUMO even more widely applicable, the toolbox was designed to be as modular and extensible as possible, without becoming too cumbersome to use or configure. Many different modules are readily available for use: model types (neural networks, support vector machines, rational functions, gaussian process models, ...), hyperparameter optimization algorithms (Pattern Search, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), ...), active learning (density based, error based, hybrid, ...) and data sources (datasets, local simulator codes, simulator execution on a cluster or grid). The behavior of each component is configurable through a central XML file and new components can easily be added, removed or replaced by custom, problem-specific implementations, if none of the available implementations are suitable for the problem at hand. Additionally, the toolbox supports ensemble models and heterogeneous model selection, in which several different model types are trained and compared against each other automatically [5].

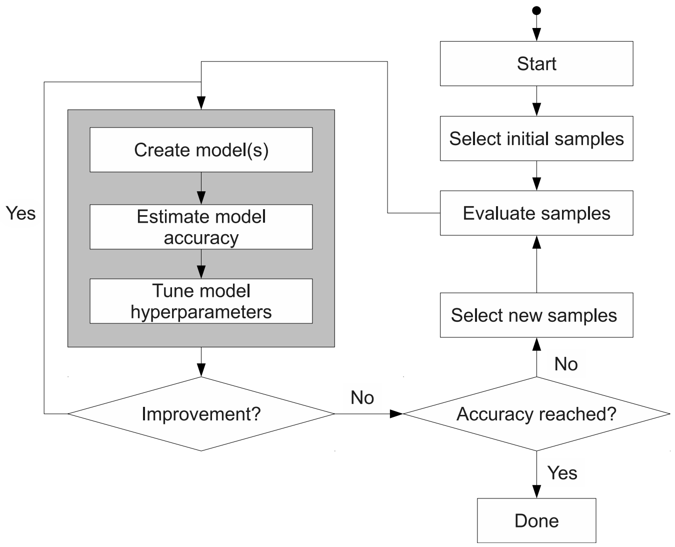

Figure 2. Flow-chart of the SUMO toolbox.

The difference with existing machine learning toolkits such as Rapidminer (formerly Yale), Spider, Shogun, Weka, and Plearn is that they are heavily biased towards classification and data mining, while SUMO focuses entirely on regression with expensive simulators (and therefore limited amounts of data) with getting an accurate global surrogate model as the final goal. Focusing on regression with expensive data allowed us to develop advanced and specialized hyperparameter selection and active learning methods. The other toolkits often assume that data is freely (and abundantly) available and cheap, and they lack advanced algorithms for the automatic selection of the model type and model parameters.

Our approach has been successfully applied to fields ranging from combustion modeling in chemistry and metallurgy, semi-conductor modeling (electro-magnetism) and aerodynamic modeling (aerospace) to structural mechanics modeling in the car industry [5], [6], [7], [8], [9], [10], [11], [12]. Its success is primarily due to its flexibility, self-tuning implementation, and its ease of integration into the larger computational science and engineering pipeline.

IV. Example

As a simple example of a problem that was successfully modeled using the SUMO toolbox, we present the following application from electro-magnetism (code courtesy of Robert Lehmensiek [13]): a 3-dimensional simulation model that computes the scattering parameters for a step discontinuity in a rectangular waveguide [14]. The inputs consists of the input frequency, the gap height and the gap length. The (complex) outputs are the scattering parameters S_{11} and S_{21}. The goal is to generate an accurate surrogate model using as little data points as possible. This surrogate can then be used by the engineer for further analysis or integration into a larger circuit simulation program.

The toolbox starts with an initial sparse Latin hypercube sample distribution of 20 samples (in the 3-dimensional design space) and each adaptive sampling iteration adds 5 new samples until a maximum of 500 or a cross validation error of 0.0001 is reached. Rational functions are used for the model type, and its model parameters (degrees of freedom, number of terms in the nominator and denominator, ...) are optimized using a stochastic hill climber.

Figure 3 shows the evolution of the error for S_{11} on an independent test set as the modeling progresses. We see that after only 60 samples, the toolbox has found a model which a generalization error less than 1%, which is already acceptable for most applications. Adding another 40 extra adaptively selected samples reduces the error even further to 0.1%. At this point the surrogate model can confidently be used to replace the simulation code in virtually all cases, thus avoiding any future time intensive simulations.

V. Conclusion

The SUMO Toolbox is a flexible tool for adaptive surrogate modeling which has been successfully applied to a large number of real-life problems, as well as many popular benchmark test cases. It contains a wide variety of different model types, active learning strategies, hyperparameter optimization methods and data sources, making it a very powerful and robust tool for automatically generating accurate surrogate models.

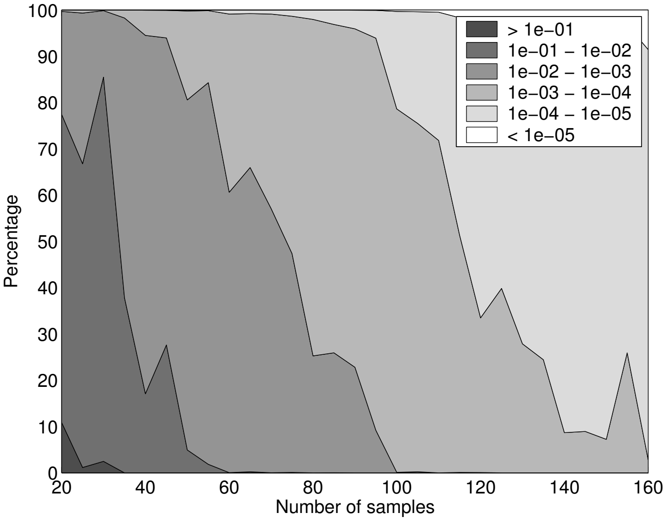

Figure 3. Evolution of the generalization error for S_{11}.

Acknowledgments

Ivo Couckuyt is funded by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). Dirk Deschrijver is a post-doctoral researcher of the Research Foundation Flanders (FWO-Vlaanderen). This research has (partially) been funded by the Interuniversity Attraction Poles Program BESTCOM initiated by the Belgian Science Policy Office.

References

[1] G. Wang and S. Shan, "Review of metamodeling techniques in support of engineering design optimization," *Journal of Mechanical Design*, vol. 129, no. 4, pp. 370–380, 2007.

[2] D. Busby, C. L. Farmer, and A. Iske, "Hierarchical nonlinear approximation for experimental design and statistical data fitting," *SIAM Journal on Scientific Computing*, vol. 29, no. 1, pp. 49–69, 2007.

[3] A. A. Jamshidi and M. J. Kirby, "Towards a black box algorithm for nonlinear function approximation over high-dimensional domains," *SIAM Journal on Scientific Computing*, vol. 29, pp. 941–963, 2007.

[4] M. Sugiyama and H. Ogawa, "Release from active learning/model selection dilemma: optimizing sample points and models at the same time," in *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, vol. 3, 12-17 May 2002, pp. 2917–2922.

[5] D. Gorissen, T. Dhaene, and F. DeTurck, "Evolutionary model type selection for global surrogate modeling," *Journal of Machine Learning Research*, vol. 10, pp. 2039–2078, 2009.

[6] 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*, vol. 18, no. 5, pp. 485–494, Jun. 2009.

[7] D. Gorissen, K. Crombecq, I. Couckuyt, and T. Dhaene, *Foundations of Computational Intelligence, Volume 1: Learning and Approximation: Theoretical Foundations and Applications*. Springer Verlag, Series Studies in Computational Intelligence, 2009, vol. 201, ch. Automatic Approximation of Expensive Functions with Active Learning, pp. 35–62.

[8] K. Crombecq, D. Gorissen, D. Deschrijver, and T. Dhaene, "A novel hybrid sequential design strategy for global surrogate modelling of computer experiments," *SIAM Journal of Scientific Computing*, vol. 33, no. 4, pp. 1948–1974, 2010.

[9] D. Deschrijver, F. Vanhee, D. Pissoort, and T. Dhaene, “Automated near-field scanning algorithm for the emc analysis of electronic devices,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 54, no. 3, pp. 502–510, 2012.

[10] K. Goethals, I. Couckuyt, T. Dhaene, and A. Janssens, “Sensitivity of night cooling performance to room/system design : surrogate models based on cfd,” *Building and Environment*, vol. 58, pp. 23–36, 2012.

[11] I. Couckuyt, S. Koziel, and T. Dhaene, “Surrogate modeling of microwave structures using kriging, co-kriging, and space mapping,” *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 26, no. 1, pp. 64–73, January/February 2012.

[12] J. Degroote, I. Couckuyt, J. Vierendeels, P. Segers, and T. Dhaene, “Inverse modelling of an aneurysm’s stiffness using surrogate-based optimization and fluid-structure interaction simulations,” *Structural and Multidisciplinary Optimization*, vol. 46, pp. 457–469, 2012.

[13] R. Lehmensiek, “Efficient Adaptive Sampling Applied to Multivariate, Multiple Output Rational Interpolation Models, with Applications in Electromagnetics-based Device Modeling,” Ph.D. dissertation, University of Stellenbosch, 2001.

[14] D. Gorissen, D. Deschrijver, T. Dhaene, and D. D. Zutter, “A software framework for automated behavioral modeling of electronic devices,” *IEEE Microwave Magazine*, vol. 13, no. 6, pp. 102–118, 2012.