Advances in Engineering Software 99 (2016) 137–146

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Adaptive classification under computational budget constraints using sequential data gathering

Joachim van der Herten*, Ivo Couckuyt[1], Dirk Deschrijver, Tom Dhaene

Ghent University – iMinds, Technologiepark 15, Gent B-9052, Belgium

ARTICLE INFO

Article history:
Received 12 October 2015
Revised 30 May 2016
Accepted 30 May 2016
Available online 7 June 2016

Keywords:
Classification
SUMO Toolbox
Sequential design
Surrogate modeling

ABSTRACT

Classification algorithms often handle large amounts of labeled data. When a label is the result of a very expensive computer experiment (in terms of computational time), sequential selection of samples can be used to limit the overall cost of acquiring the labeled data. This paper outlines the concept of sequential design for classification, and the extension of an existing state-of-the-art research platform for surrogate modeling to handle classification problems with sequential design. The capabilities of the platform are illustrated on a number of use cases including real-world applications such as an ElectroMagnetic Compatibility (EMC) and a Computational Fluid Dynamics (CFD) problem. The CFD problem also illustrates how classification can be used together with regression techniques to solve multi-objective constrained optimization problems of complex systems.

1. Introduction

Supervised learning algorithms learn the relation between an input space and a corresponding output space based on multiple examples (*samples*). After learning, the predictor can be used to predict the output(s) of unseen data points. In case an output varies continuously, this task is referred to as *regression*. When only a distinct number of discrete outcomes are possible (*labels*), the term *classification* is used. In literature, classification algorithms usually label large data sets. To limit the massive computational requirements of the learning process, the data is often sub-sampled to obtain a smaller representative set of training data.

Sometimes, obtaining the label for a sample is a very *expensive* task: it might be the result of a lengthy computer simulation or a (possibly dangerous) real-life experiment. Assuming there are *budget constraints* limiting the total amount of labels that can be acquired, obtaining the labels for all samples in the data set might not be possible. Although budget constraints also include applications where time and money is required for instance preparation [1], this article focuses on labels obtained through evaluation of complex physics-based (deterministic) simulators. These are used frequently in computer-aided design and engineering (CAD/CAE) to avoid building and testing several prototypes of new products. As these simulations have become significantly more accurate over the years, their computational requirements have also become more expensive.

This article describes a state-of-the-art platform for surrogate modeling [2] with sequential design. A surrogate model is a cheap-to-evaluate mathematical regression model mimicking the response of computationally intensive simulators with continuous response range, and is trained from a small set of (sequentially) well-chosen evaluations. The platform was recently expanded with classification models and some state-of-the-art sequential design methods targeting classification applications. The SUMO Toolbox is introduced in Section 2 with a focus on these new extensions. The concept of sequential design is introduced in Section 3, and the sequential sampling step for classification is discussed in more detail in Section 4. The integrated platform is then illustrated on a number of use cases in Section 5.

2. SUMO Toolbox

Designed as a research platform for sequential sampling and adaptive modeling using MATLAB, the SUMO Toolbox [3] has grown into a mature design tool for surrogate modeling with sequential design offering a large variety of algorithms for simulators with continuous output. The software design is fully object-oriented allowing high-extensibility of its capabilities. By default, the platform follows the integrated modeling flow as shown in

* Corresponding author.
E-mail addresses: joachim.vanderherten@ugent.be (J. van der Herten), ivo.couckuyt@ugent.be (I. Couckuyt), dirk.deschrijver@ugent.be (D. Deschrijver), tom.dhaene@ugent.be (T. Dhaene).
URL: http://sumo.intec.ugent.be (T. Dhaene)

[1] Ivo Couckuyt is a post-doctoral research fellow of the Research Foundation Flanders (FWO).

http://dx.doi.org/10.1016/j.advengsoft.2016.05.016
0965-9978/© 2016 Elsevier Ltd. All rights reserved.

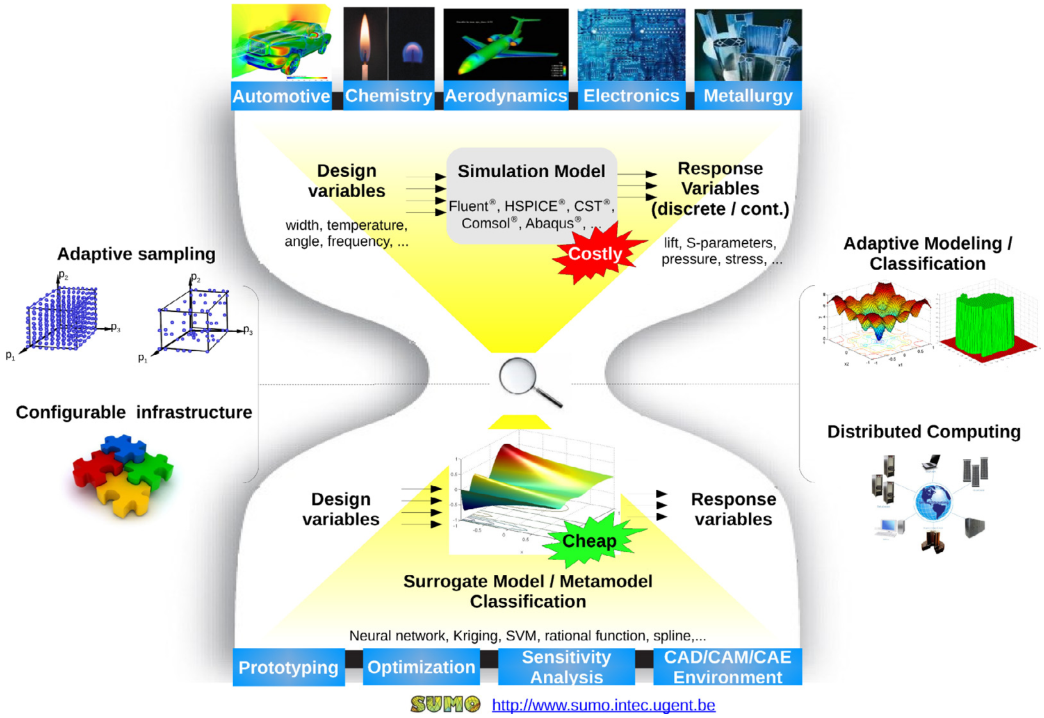
Automotive
Chemistry
Aerodynamics
Electronics
Metallurgy
Design variables
width, temperature, angle, frequency, ...
Simulation Model
Fluent®, HSPICE®, CST®, Comsol®, Abaqus®, ...
Response Variables (discrete / cont.)
lift, S-parameters, pressure, stress, ...
Costly
Adaptive sampling
Configurable infrastructure
Adaptive Modeling / Classification
Distributed Computing
Design variables
Response variables
Cheap
Surrogate Model / Metamodel Classification
Neural network, Kriging, SVM, rational function, spline,...
Prototyping
Optimization
Sensitivity Analysis
CAD/CAM/CAE Environment
SUMO
http://www.sumo.intec.ugent.be

Fig. 1. Design philosophy of the SUMO Toolbox for surrogate modeling. The Toolbox was recently extended to support classification applications under budget constraints.

Fig. 3, but can also be configured to model data sets, use a one-shot setup etc. Recently, the platform has been extended to offer support for several classification algorithms by including several implementations and linking the WEKA library [4].

Fig. 1 illustrates the design goals of the SUMO Toolbox. Expensive computer simulations of complex black-box systems with several design parameters are approximated by a cheap-to-evaluate model, and the toolbox can also approximate outputs with a discrete set of labels by training a classifier. To obtain these goals, the SUMO Toolbox offers sequential sampling and adaptive modeling in a highly configurable environment which is easy to extend due to the microkernel design philosophy as illustrated in Fig. 2. Distributed computing support for evaluations of data points is also available, as well as multi-threading to support the usage of multi-core architectures for regression modeling and classification.

Many different plugins are available for each of the different sub-problems: model types (rational functions, Kriging [5], splines, Support Vector Machines (SVM) [6–8], Artificial Neural Networks (ANN), Extreme Learning Machines (ELM) [9], Least Squares-SVM (LS-SVM) [10], Random Forests [11]), hyperparameter optimization algorithms (Particle Swarm Optimization [12], Efficient Global Optimization [13], simulated annealing, Genetic Algorithm), sample selection (random, error based, density based [14,15], hybrid [16]), Design of Experiments (Latin Hypercube [17,18], Box-Bhenken), 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 implementation.

During the adaptive modeling step, the Toolbox uses the following methodology for model selection to guide the hyperparameter optimization: the quality of a model $\tilde{f}_\theta$ parametrized by θ of a dataset D is denoted as:

$$\Lambda\left(\epsilon, \tilde{f}_\theta, D\right). \tag{1}$$

Λ denotes a quality estimator for model selection: the SUMO Toolbox supports several algorithms such as a validation set, cross validation, Akaike Information Criterion (AIC) [19], SampleError, jackknife and LRM [2]. The quality estimator uses an error function ϵ: popular choices are Root Mean Square Error (RMSE), Root Relative Square Error (RRSE) for regression [20], or the misclassification rate for classification.

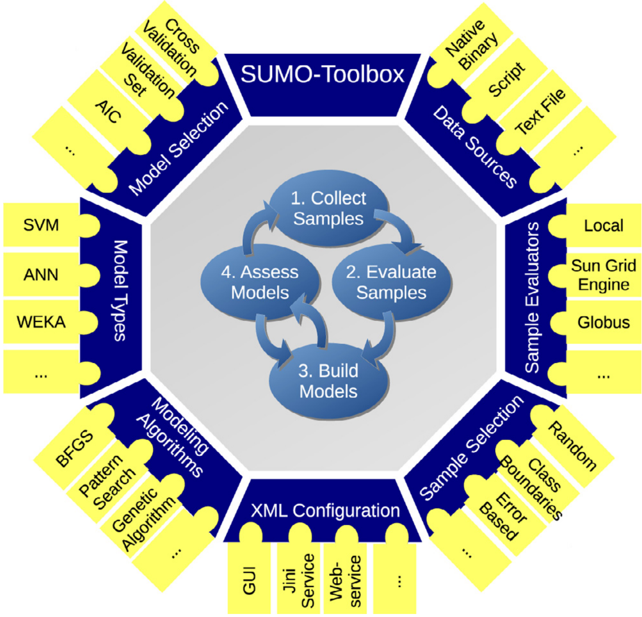

Fig. 2. Microkernel architecture of the SUMO Toolbox.

The architecture for hyperparameter optimization of the Toolbox also allows optimization of the classifier parameters to improve its position in the ROC space, a popular method to present the accuracy of a classifier. This can be seen as a multi-objective goal: minimizing the false positive rate and maximizing the true positive rate. Both rates can be determined by evaluation of a quality estimator. By combining these objectives into a single multi-