CEUR-WS.org/Vol-2946/paper-08.pdf

Surrogate modeling assistant software

Anastasia V. Demidova, Tatyana R. Velieva[1,3], Anna V. Korolkova[1] and Dmitry S. Kulyabov[1,2]

[1]Peoples' Friendship University of Russia (RUDN University), 6 Miklukho-Maklaya St, Moscow, 117198, Russian Federation

[3]Plekhanov Russian University of Economics, 36 Stremyanny lane, Moscow, 117997, Russian Federation

[2]Laboratory of Information Technologies, Joint Institute for Nuclear Research, 6 Joliot-Curie St., Dubna, Moscow region, 141980, Russian Federation

Abstract

Computer modeling is designed to predict the behavior of complex systems by solving the corresponding mathematical equations of the physical process. The simulation requires a huge number of parameters to be found, and they all require a large number of simulation runs, where each run takes a different combination of design parameters as input. Computer simulations tend to be expensive. Research that requires a large number of computer calculations will lead to exorbitant computational costs, which will make them practically impracticable. This is what surrogate modeling is for. Surrogate modeling builds a statistical model to accurately approximate the simulation result. Subsequently, this trained model can replace the original computer simulation when performing system analysis.

Keywords

surrogate modeling, Julia language, Python language

1. Introduction

Recently, statistical modeling, a subsection of mathematical modeling that uses the mathematical apparatus of machine learning [1, 2], has been intensively developing. Surrogate modeling is a subsection of statistical modeling. When building a surrogate model, a so-called surrogate is built, a substitute for a computationally complex mathematical model for a simpler one that imitates the behavior of the main model. Such a model should satisfy the requirement of fast computation at arbitrary points. Surrogate modeling is the replacement of a computationally complex function with a computationally simpler function.

Surrogate modeling builds a statistical model to accurately approximate the simulation result. Subsequently, this trained model can replace the original computer simulation when performing system analysis. Surrogate modeling methods make a resource-intensive analysis

Workshop on information technology and scientific computing in the framework of the XI International Conference Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems (ITTMM-2021), Moscow, Russian, April 19–23, 2021

demidova-av@rudn.ru (A. V. Demidova); velieva-tr@rudn.ru (T. R. Velieva); korolkova-av@rudn.ru (A. V. Korolkova); kulyabov-ds@rudn.ru (D. S. Kulyabov)
https://yamadharma.github.io/ (D. S. Kulyabov)
0000-0003-1000-9650 (A. V. Demidova); 0000-0003-4466-8531 (T. R. Velieva); 0000-0001-7141-7610 (A. V. Korolkova); 0000-0002-0877-7063 (D. S. Kulyabov)

© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

of a computational model more accessible, since evaluating the trained statistical model is much faster than evaluating the original simulation, performing many experiments with various combinations of design parameters.

2. Surrogate modeling

The process of building a surrogate model can be divided into three main stages, which can be alternated iteratively:

- Receiving and preprocessing data. As data can be used as data obtained as a result of tests and computational experiments or an analytical method. In addition, this stage involves reducing the dimension of the data by removing unnecessary or redundant parameters from the dataset and forming a training sample.
- Construction of a surrogate model and optimization of model parameters. At this stage, the method for implementing the model should be chosen. The most common methods include models such as Kriging, Artificial Neural Networks, Support Vector Regression, Multivariate Nonparametric Regression, Polynomial Regression, etc. The model allows constructing an approximation based on the selected training sample. For best results, you can plot multiple approximation curves.
- Estimating the accuracy of the surrogate model. The accuracy of the surrogate model depends on the quantity and quality of the data. In addition, a poor fit can arise from noise in the data or from poorly chosen surrogate model building techniques.

2.1. Data collection methods

The first step in building a surrogate model is collecting training data. After collecting the input data and the corresponding output data into a training set, we can build a statistical model based on this set.

Let's list the main methods of data selection when building surrogate models:

- random selection;
- Latin sample of the hypercube;
- factorial sampling;
- low discrepancy sequences.

2.2. Methods of construction of surrogates

After collecting the data, we build the model. Now there are many surrogate models, and the choice of model depends on the task: classification, approximation, forecast, etc.

Let's list the main methods for constructing and training surrogate models:

- linear models;
- radially basic models;
- kriging;
- inverse distance weighted method;
- spline method;
- 2nd order polynomial method.

3. Surrogate modeling toolkits

Let's list the most popular general-purpose surrogate modeling packages.

- Matlab toolbox SUrrogate MOdeling (SUMO) [3] is an extremely powerful package that is widely used in scientific research.
- A specialized library named SMT [4, 5] has been developed for the Python language, which supports many surrogate modeling methods, such as kriging, linear methods, radial basis method, etc.
- The machine learning library scikit-learn [6] has several methods for constructing surrogates, including kriging and linear methods.
- SciML/Surrogates.jl library [7, 8] is implemented for Julia programming language. It is ideologically similar to the SMT library. This package supports many necessary methods for working with surrogate models: linear models, kriging, splines, etc.

4. Surrogate modeling package on Julia SciML/Surrogates.jl

SciML/Surrogates.jl package is for surrogate modeling and optimization for scientific machine learning. To install the package, just run the commands as follows:

```
using Pkg
Pkg.add("Surrogates")
```

The SciML/Surrogates.jl package provides all the necessary tools for the complete process of creating a surrogate model. We will build a surrogate model in three stages.

- Sample selection. The following methods are implemented for sampling:
  - Grid sample;
  - Uniform sample;
  - Sobol sample;
  - Latin Hypercube sample;
  - Low discrepancy sample.
- Building a surrogate model. The following surrogates are available:
  - Linear;
  - Radial Basis;
  - Kriging;
  - Neural Network;
  - Support Vector Machine;
  - Random Forest;
  - Second Order Polynomial;
  - Inverse Distance.
- Optimization. After the surrogate is built, it needs to be optimized for some objective function. The following optimization methods are available:

- Stochastic RBF (SRBF);
- Lower confidence bound strategy (LCBS);
- Expected improvement (EI);
- Dynamic coordinate search (DYCORS).

5. Surrogate modeling package on Python SMT

Surrogate Modeling Toolkit (SMT) [5, 4] is an open-source Python package consisting of libraries of surrogate modeling techniques (e.g. radial basis functions, kriging), sampling techniques, and the tasks of comparative analysis. SMT is designed to make it easy to develop new test models on a well-tested and well-documented platform.

SMT consists of three main modules that implement a set of sampling methods, benchmarking functions, and surrogate modeling methods, respectively. Each module contains a common interface inherited by the corresponding methods, and each method implements the functionality required by the interface.

There are two main steps required to build a surrogate model. First, we generate a set of training points from the input space. This step can be done using one of the sampling methods implemented in SMT, or by loading an existing training dataset. Second, we train the desired surrogate model at these points and make a prediction of the outputs and derivatives.

SMT contains a library of sampling techniques used to generate sets of points in the input space for both training and prediction.

6. An example of surrogates construction

Let's consider the construction of surrogates using the SciML/Surrogates.jl library as an example.

Let us consider various methods using the example of collecting samples to build a surrogate model for the simplest function as follows

$$f(x) = \sin x. \tag{1}$$

6.1. Sample selection

To generate samples by various methods for the function under consideration on the interval $[0, 30]$ at 15 points using the built-in functions of the SciML/Surrogates.jl package:

```
# Grid method
x_grid = sample(n_samples, lower_bound, upper_bound, GridSample(0.2))
y_grid = f.(x_grid)

# Uniform method
x_uniform = sample(n_samples, lower_bound, upper_bound, UniformSample())
y_uniform = f.(x_uniform)

# Sobol method
```

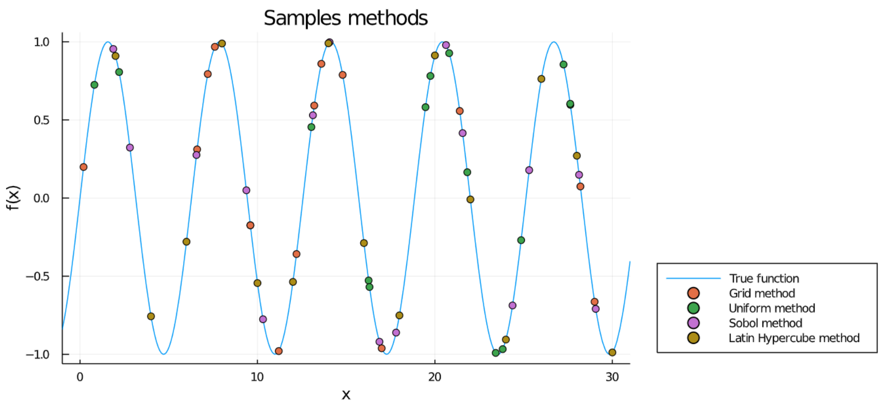

Figure 1: Sampling methods

```
x_sobol = sample(n_samples, lower_bound, upper_bound, SobolSample())
y_sobol = f.(x_sobol)

# Latin hypercube method
x_lh = sample(n_samples, lower_bound, upper_bound, LatinHypercubeSample())
y_lh = f.(x_lh)
```

The graphic 1 shows the results of generating samples. It can be seen from the graph that for this example, the best examples are provided by the Sobol sequence methods and the Latin hypercube method.

6.2. Construction of surrogates

To construct a surrogate model of the (1) function, we use various methods implemented in the SciML/Surrogates.jl library:

```
# Radial basis method
RBF_surrogate = RadialBasis(x, y, lower_bound, upper_bound)

# Kriging
kriging_surrogate = Kriging(x, y, lower_bound, upper_bound, p=1.9)

# Lobachesky method
lobachevsky_surrogate = LobacheskySurrogate(x, y, lower_bound, upper_bound,
 ↪  alpha = 2.0, n = 6)

# Inverse distance method
InverseDistance = InverseDistanceSurrogate(x, y, lower_bound, upper_bound)
```

The simulation results using these methods for the sequences obtained by the Sobol selection method and the Latin Hypercube selection method are presented in the graphics 2 and 3.

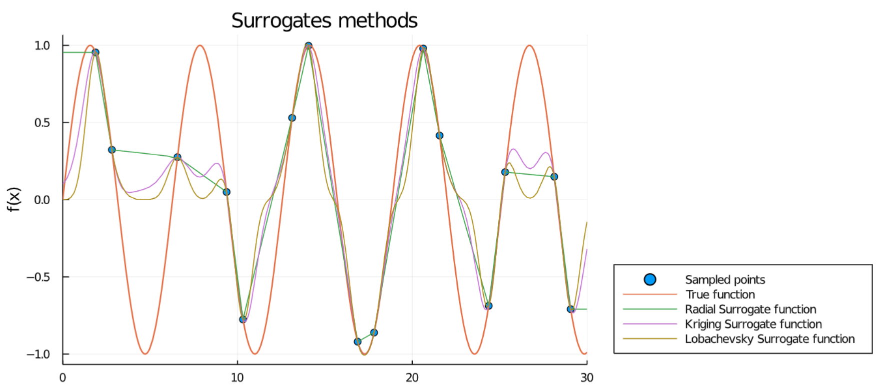

Figure 2: Construction of surrogates (Sobol selection method)

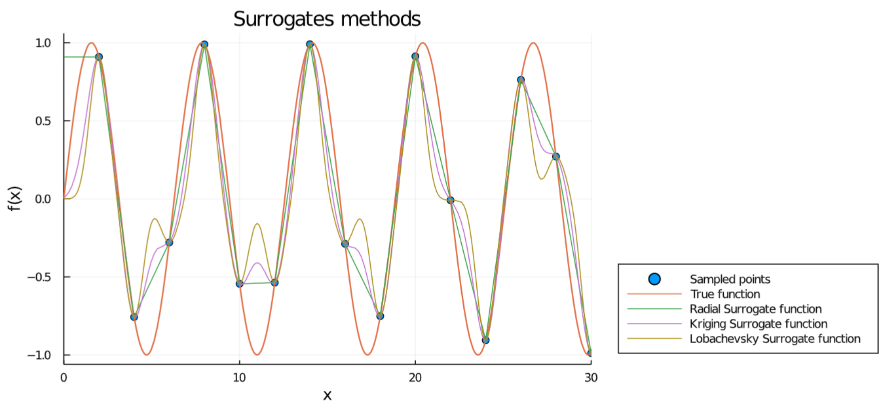

Figure 3: Construction of surrogates (Latin Hypercube method)

It can be concluded that for the studied example for constructing a surrogate model, the radially basic method gives the worst of the presented results. While kriging and Lobachevsky's method give approximately the same picture.

7. Conclusion

In this article, we reviewed the main opensource surrogate modeling toolkits. To study the features of software implementations, a simple surrogate model was built. The Surrogates.jl and SMT libraries are given as examples of surrogate modeling implementation. From the point of view of the authors, the Python SMT library is currently slightly superior to the Julia library Surrogates.jl.

Acknowledgments

This paper has been supported by the RUDN University Strategic Academic Leadership Program (Anastasia V. Demidova), by the Russian Science Foundation grant No. 19-71-30008 (Tatyana R. Velieva) by the Russian Foundation for Basic Research (RFBR) according to the research project No 19-01-00645 (Anna V. Korolkova, Dmitry S. Kulyabov).

References

[1] L. A. Sevastianov, A. L. Sevastianov, E. A. Ayrjan, A. V. Korolkova, D. S. Kulyabov, I. Pokorny, Structural approach to the deep learning method, in: V. Korenkov, T. Strizh, A. Nechaevskiy, T. Zaikina (Eds.), Proceedings of the 27th Symposium on Nuclear Electronics and Computing (NEC-2019), volume 2507 of *CEUR Workshop Proceedings*, Budva, 2019, pp. 272–275.

[2] M. N. Gevorkyan, A. V. Demidova, D. S. Kulyabov, Comparative analysis of machine learning methods by the example of the problem of determining muon decay, Discrete and Continuous Models and Applied Computational Science 28 (2020) 105–119. doi:10.22363/2658-4670-2020-28-2-105-119.

[3] SUrrogate MOdeling (SUMO) Toolbox, 2021. URL: http://www.sumo.intec.ugent.be/SUMO_toolbox.

[4] Smt: Surrogate modeling toolbox, 2021. URL: https://github.com/SMTorg/SMT.

[5] M. A. Bouhlel, J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier, J. R. Martins, A python surrogate modeling framework with derivatives, Advances in Engineering Software 135 (2019) 102662. doi:10.1016/j.advengsoft.2019.03.005.

[6] scikit-learn: Machine learning in python, 2021. URL: https://scikit-learn.org/stable/index.html.

[7] Surrogate modeling and optimization for scientific machine learning (sciml), 2021. URL: https://github.com/SciML/Surrogates.jl.

[8] Surrogates.jl, 2021. URL: https://surrogates.sciml.ai/latest/.