Computers and Chemical Engineering 96 (2017) 103–114

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Smart Sampling Algorithm for Surrogate Model Development

Sushant Suhas Garud[a,b], I.A. Karimi[a,b,*], Markus Kraft[a,c,d]

[a] Cambridge Centre for Carbon Reduction in Chemical Technology (C4T), 05-05 CARES, CREATE Tower, 1 CREATE Way, 138602, Singapore
[b] Department of Chemical and Biomolecular Engineering, 4 Engineering Drive 4, National Univeristy of Singapore, 117585, Singapore
[c] Department of Chemical Engineering and Biotechnology, New Museums Site, Pembroke Street, Cambridge, CB2 3RA, United Kingdom
[d] School of Chemical and Biomedical Engineering, Nanyang Technological University, 62 Nanyang Drive, 637459, Singapore

ARTICLE INFO

Article history:
Received 4 May 2016
Received in revised form 4 September 2016
Accepted 12 October 2016
Available online 19 October 2016

Keywords:
Smart sampling
Adaptive surrogate modelling
Point placement

ABSTRACT

Surrogate modelling aims to reduce computational costs by avoiding the solution of rigorous models for complex physicochemical systems. However, it requires extensive sampling to attain acceptable accuracy over the entire domain. The well-known space-filling techniques use sampling based on uniform, quasi-random, or stochastic distributions, and are typically non-adaptive. We present a novel technique to select sample points systematically in an adaptive and optimized manner, assuring that the points are placed in regions of complex behaviour and poor representation. Our proposed smart sampling algorithm (SSA) solves a series of surrogate-based nonlinear programming problems for point placement to enhance the overall accuracy and reduce computational burden. Our extensive numerical evaluations using 1-variable test problems suggest that our SSA performs the best, when its initial sample points are generated using uniform sampling. For now, this conclusion is valid for 1-variable functions only, and we are testing our algorithm for n-variable functions.

1. Introduction

Naturally, we grasp the understanding of a complex phenomenon by opting for a simpler format. Similarly, we use process simulators to model, study, and analyze complex and nonlinear physicochemical processes. However, such simulations can be compute-intensive, and running them repeatedly in an optimization/analysis procedure can be computationally prohibitive. Furthermore, numerical models can pose significant hurdles within a continuous optimization algorithm. Therefore, it helps to convert a high-fidelity simulation model into a computationally inexpensive surrogate model that captures its essential features with prescribed numerical accuracy.

Surrogate modelling, also known as metamodeling, is a technique to generate a mathematical or numerical representation of a complex system based on some sampled input-output data. Many surrogate modelling techniques have been developed over the past few decades such as Polynomial Surface Response Models (PRSM) (Forrester and Keane, 2009; Myers and Montgomery, 2002; Queipo et al., 2005), Kriging (Cressie, 1990; Forrester et al., 2008a; Martin and Simpson, 2005; Sakata et al., 2003; Simpson, 1998), Radial Basis Functions (RBF) (Hardy, 1971; Hussain et al., 2002), Support Vector Regression (SVR) (Clarke et al., 2005), and Artificial Neural Networks (ANNs) (Yegnanarayana, 2004). The literature (Forrester and Keane, 2009; Henao and Maravelias, 2010, 2011; Queipo et al., 2005; Shan and Wang, 2010; Wang and Shan, 2007) has compared them and discussed their applications to various systems.

Irrespective of the technique, building a surrogate model requires sample points. The process of generating such points is known as sampling. We can classify the existing sampling methods into two broad categories: non-adaptive and adaptive. The four types of non-adaptive methods are grid-based, pattern/geometry-based, stochastic, and quasi-random. The grid-based method simply distributes sample points to form a uniform grid (Cartesian grid). The second type employs statistics-driven methods such as the design of experiments to fill space. These include full/half factorial designs (Fisher, 1935), central composite (CC) designs (Box and Wilson, 1951), Box-Behnken (Box and Behnken, 1960), Plackett-Burman (Plackett and Burman, 1946), Delaunay triangulations and their dual structures (Delaunay, 1934), and Voronoi tessellations (Voronoï, 1908). These methods work well for low dimensions ($N \leq 3$) (Davis and Ierapetritou, 2010); (Crombecq et al., 2009), but become extremely costly for large N as in the case of geometry and factorial designs, or inaccurate due to the lack of spatial coverage as in the case of CC, Plackett-Burman, and Box-Beheken designs. Moreover, these methods rapidly face the *curse of dimensionality*

* Corresponding author at: Department of Chemical and Biomolecular Engineering, 4 Engineering Drive 4, National Univeristy of Singapore, 117585, Singapore.
E-mail address: cheiak@nus.edu.sg (I.A. Karimi).

http://dx.doi.org/10.1016/j.compchemeng.2016.10.006
0098-1354/

(Forrester et al., 2008c). The third type of sampling methods relies on stochastic sampling with the aim to fill space. For instance, sampling based on random distribution is the most straightforward. The more sophisticated methods include Monte Carlo sampling (Metropolis and Ulam, 1949) and variations (Koehler and Owen, 1996; Niederreiter, 2010) that combine random sampling and probabilistic filtering; Latin hypercube sampling (McKay et al., 1979) that fills hypercube bins via random placements but subject to projection filters; and orthogonal arrays (Hedayat et al., 2012; Rao, 1946, 1947) that generalize Latin hypercube sampling (Giunta et al., 2003). Finally, the fourth type generates sample points quasi-randomly using low-discrepancy sequences such as Hammersley (Hammersley and Handscomb, 1964), Sobol (Sobol', 1967), and Halton (Halton and Smith, 1964). While these methods manage the sample size much better (Mascagni and Hongmei, 2004; Queipo et al., 2005), they may fail to capture the characteristics of the space properly at higher N. Typically, the system output is computed from the high-fidelity model at these sample points to generate the required input-output data for surrogate development. If the resulting surrogate does not meet expectations, then the model is reconstructed by adding more sample points.

The adaptive sampling methods attempt to address the drawbacks of the non-adaptive techniques discussed above. An adaptive method starts with a small set of sample points, and then adds points sequentially using some criteria or procedures. Provost et al. (1999) have shown that such methods normally require fewer sample points than the non-adaptive ones for the same surrogate accuracy. The approaches and objectives behind the adaptive techniques are varied. The most commonly used grid sampling technique, namely the Cartesian grid, can be made adaptive by evolving the entire grid rather than individual sample points. If a grid at hand is inadequate, then midpoints are inserted as additional sample points to make a finer grid. Thus, if a 5×5 grid (25 sample points) is inadequate for a $2D$ surrogate, then 56 additional points are added to get a 9×9 grid of 81 points. It is clear that the sample size increases exponentially with N in this method. Therefore, it is important to explore the idea of strategic sampling that not only fills the space, but also exploits the system knowledge for smart placements (Forrester et al., 2008b). Crombecq et al. (2009) proposed a novel sequential strategy involving both exploration and exploitation. They used a combination of derivative-based local linear approaximations and Voronoi tesselations to place new sample points. But, in most cases, the derivatives of a system to be modelled are not available a priori, and estimating them accurately and efficiently can be an arduous task due to the black-box and compute-intensive nature of the system. Moreover, as discussed earlier, the Voronoi tesselations can rapidly become computationally expensive at higher dimensions. A recent work by Eason and Cremaschi (2014) proposes an adaptive sampling strategy for ANN surrogates. Instead of generating all sample points in one shot, they generate them sequentially and randomly in a piecemeal manner. They use a *score* to select new sample points from the random points generated at each iteration. The score considers the normalized nearest neighbor distance of a potential point from the current sample points and its normalized expected variance evaluated using jackknifing (Quenouille, 1956). Though their selection of sample points is systematic, it is still from randomly generated points. Cozad et al. (2014, 2015) propose adaptive sampling for their surrogate modelling tool called ALAMO. They add sample points one at a time to an initial DoE-based set. For each sample point, they solve a derivative-free optimization problem to maximize the deviation of the surrogate from the real function. This can obviously be compute-intensive, as it requires the evaluation of the real function during optimization. Jin et al. (2016) essentially made two modifications to the work of (Eason and Cremaschi, 2014). One, they improved ANN modelling by introducing auto-node selection, and second, they used maximum predicted error instead of expected variance.

In this work, we develop a smart sampling algorithm (SSA) that differs from the current approaches described above in four novel aspects. First, unlike Cremaschi and coworkers, it does not use any random sample points, potential or otherwise. Second, unlike Cozad et al., it employs surrogate-based optimization using derivative information rather than a black-box-based, derivative-free optimization. Third, it exploits all sample points as one set instead of dividing them into small subsets. Lastly, it integrates both spatial and quality considerations in a single objective to place new sample points.

In the rest of this article, we first define our problem and the key concepts behind our algorithm. Then, we present our algorithm, and illustrate it with a simple example and a practical case study. Finally, we evaluate its performance numerically using a variety of single-variable test problems from the literature.

2. Problem Statement

Let $y = f(\boldsymbol{x}); f : \Re^N \rightarrow \Re$ for $\boldsymbol{x}^L \leq \boldsymbol{x} \leq \boldsymbol{x}^U$ describe the behaviour of a unit/process/system whose experimental or computational quantification is complex and compute-intensive. We need an analytical or numerical surrogate model $\tilde{y} = S(\boldsymbol{x})$ to replace $f(\boldsymbol{x})$ in an optimization and/or analysis task. Hence, our problem is as follows.

Given:

- $y = f(\boldsymbol{x}); f : \Re^N \rightarrow \Re$ for $\boldsymbol{x}^L \leq \boldsymbol{x} \leq \boldsymbol{x}^U$.
- A mathematical form for $S(\boldsymbol{x})$.
- Upper limit (K_{max}) on the number of sample points at which $f(\boldsymbol{x})$ may be evaluated to obtain $S(\boldsymbol{x})$, or a desired accuracy for $S(\boldsymbol{x})$.

Obtain:

- K_{max} sampling points $\boldsymbol{x}_k$ ($k = 1, 2, \ldots, K_{max}$) that give the best $S(\boldsymbol{x})$ for approximating $f(\boldsymbol{x})$
- Or, the sampling points that give $S(x)$ with a prescribed accuracy for approximating $f(\boldsymbol{x})$.

3. Motivation

The most common sampling methods employed in surrogate construction are uniform (US), random (RS), statistical (LHS, e.g. Latin Hypercube), central composite design (CCD), and quasi-random low-discrepancy (QS, e.g. Sobol sequence). Each generates a set of sample points to achieve a spatially uniform coverage of $\Re^N$, and evaluates $f(\boldsymbol{x})$ at these points to obtain the required input-output data to construct $S(\boldsymbol{x})$. This surrogate development paradigm involves several key issues: How many sample points should we use? How (one-shot or adaptively) should we generate them? How do they affect the quality of surrogate approximation? Which sampling method gives the best approximation for a given form of $S(\boldsymbol{x})$?

Let us look at these questions with the help of an example. Consider using US and RS to obtain $S(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$ for $f(x) = x \times \sin(2 \times \pi \times x)$ on $0 \leq x \leq 1$. For US, we compute $f(x)$ at $x = 0$, $x = 1$, and nine equidistant points between them. We get $S(x) = -0.02 + 0.75x + 6.65x^2 - 25.27x^3 + 17.94x^4$. Now, let us say that we have a new sampling method that gives us the seven points shown by diamonds in Fig. 1, which give us a surrogate as good as the one from US. In other words, US amounted to oversampling. Now, let us assume that seven points are sufficient to get an acceptable $S(x)$, so we generate seven random points between 0 and 1. Fig. 1 shows that $S(x)$ does well only for $x \in [0.08, 0.80]$. In other words, RS can mean poor sample placement. Hence, the