NSF GRANT # 0521953
NSF PROGRAM NAME: Operations Research

Classification-Based Global Search: an Application to a Simulation for Breast Cancer

Michael Ferris
Computer Sciences Department, University of Wisconsin, Madison
Geng Deng
Mathematics Department, University of Wisconsin, Madison

Abstract: In simulation-based optimization, we seek the optimal parameter settings that minimize or maximize certain performance measures of the simulation system. In this paper, we use a two-phase approach to calibrate simulation parameters using classification tools. This classification-based method is used in Phase I to facilitate the global search process and it is followed by local optimization in Phase II. By learning knowledge from existing data the approach identifies potentially high-quality parameter settings. We present an example of its use on a Wisconsin breast cancer simulation.

1. Introduction: Over the past few decades, computer simulation has become a powerful tool for developing predictive outcome of real systems. For example, simulations consisting of dynamic econometric models of travel behavior are used for nationwide demographic and travel demand forecasting. The choice of optimal simulation parameters can lead to improved operation, but configuring them remains a challenging problem. Traditionally, the parameters are chosen by heuristics with expert advice, or by selecting the best from a set of candidate parameter settings. *Simulation-based optimization* is an emerging field which integrates optimization techniques into the simulation analysis. The corresponding objective function is an associated measurement of an experimental simulation. Due to the complexity of simulation, the objective function may act as a black-box function and be time-consuming to evaluate. Moreover, since the derivative information of the objective function is typically unavailable, many derivative-dependent methods are not applicable.

In this paper, calibration of simulation parameters is formulated as a general stochastic unconstrained minimization problem:

$$\min F(x) = E\,[f(x,\xi)]$$

Here the variable x is the set of input parameters and ξ is a random variable which is internally or externally originated from simulation. We assume that only a moderate level of noise exists in our case.

In the literature of simulation-based optimization, a general two-phase framework [1,5] is widely accepted. Each phase has a distinct purpose:

- Phase I is a global exploration step. The algorithm explores the entire domain and proceeds to determine promising subregions for future investigation.
- Phase II is a local exploitation step, in which local optimization algorithms are applied to solve for the exact optimum.

Phase I typically produces one or multiple excellent points which indicate the locations of good subregions. These points are used as starting points for the Phase II local search methods. In this paper we present a *classification-based* Phase I global exploration procedure and apply it to the calibration of a simulation.

2. Classification-based Global Search: The idea of our global exploring approach centers around using machine learning methods to discover valuable simulation settings. As we have known, the goal in Phase I is to locate promising subregions rather than to determine the exact solution, therefore, Phase I is only a rough search step. One may not care about how an underlying function F behaves over the whole space, instead, caring about the behavior of a simple indicator function that is 1 for x residing in a promising subregion and 0 otherwise. This function gives sufficient information to determine where a promising subregion is located. Approximating the indicator function is simpler than approximating the underlying function F.

In our approach, a classifier works as a surrogate function for the indicator function. A classifier is a cheap mechanism to predict whether new samples are in a promising subregion or not. The target promising subregion is often defined as a certain level set L(c) where c is an adjustable parameter that quantifies the volume of the set. The value of c may be determined, for example, as a quantile value of the responses.

The classifier is built on the training data to create appropriate decision rules. In the implementation, we use a voting scheme to derive a robust decision rule by combining various classification techniques by a voting scheme. The decision rule is then used to evaluate potential new evaluation set as follows. We generate an alternative sample library as an evaluation set from more refined space-filling points. The classifier is applied to assign these points to the corresponding class As a consequence, the classification implicitly partitions the domain space into positive and negative zones. Typically, we expect the process to greatly increase the chances of generating refined points in promising subregions. At the end of Phase I, we can validate the subset of the identified promising points by performing additional simulation evaluations.

The general procedure is now summarized in detail (refer to the flow chart in Figure 1).

1. Generate coarse space-filling samples and evaluate them via simulation. Choose an appropriate value of c, and split the samples into positive (points in L(c)) and negative samples (points outside L(c)). Typically, we suggest to set c as the 10% quantile value.
2. Use a pre-processing procedure to generate a balanced dataset. 6 classifiers are considered, but only those passing a performance test are used.
3. Given the training set, derive an ensemble of classifiers using the voting scheme.
4. Generate a fine space-filling evaluation set either by the grid sampling or the latin hypercube sampling. Determine the membership points by classification.
5. For those points that are predicted to lie in L(c), evaluate them via simulation.

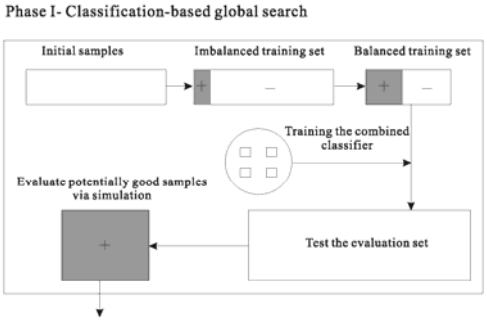


Figure 1: Flow Chart for Phase I

All the evaluated points are passed to Phase II for local optimization methods. One may be concerned about how we can identify distinct subregions from the observation of these discrete points. In fact, when the dimension of the input is small, we may visualize the scatter plot of points to recognize disjoint subregions; when the dimension is high, the situation becomes complicated; we propose elsewhere [1] a nonparametric statistical approach for this. The classification-based approach returns a set of representative points for a bunch of subregions, from which the Phase II local optimization proceeds [2,3]. The figure below outlines the overall algorithm which we call WISOPT [1]; further details can be found in [1].

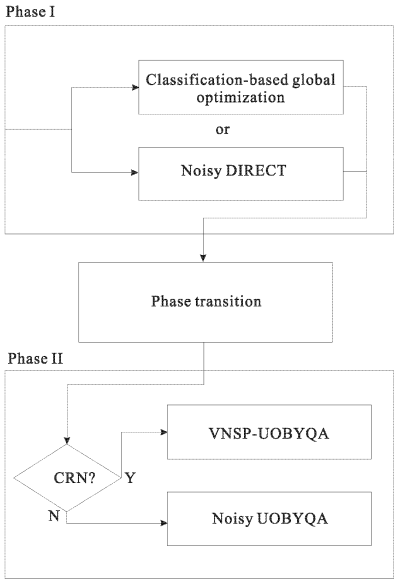


Figure 2: Flow Chart for WISOPT

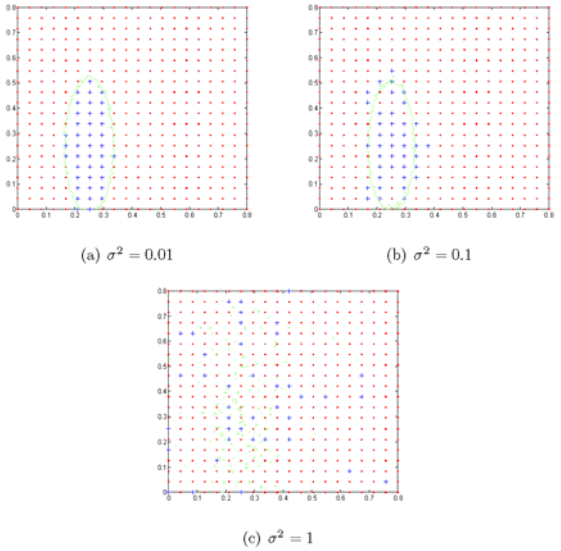
3. Numerical Examples: The classification-based global search is relatively insensitive to noise. The accuracy of the method is highly related to the accuracy of the training set, i.e., whether a positive sample is indeed a positive sample, which is equivalent to the positive sample being located in a subregion of the underlying objective function. We show here that the estimated level sets (represented by positive points) are quite insensitive to the noisy data.

We plot several estimated level sets of a small example in Figure 3. The test function was the Gauss function with additive noise:

$$F(x,\xi) = 1-\exp(-20(x_1-0.25)^2-2(x_2-0.25)^2) + \xi$$

where ξ is a noise term distributed as $N(0,\sigma^2)$. We applied the grid sampling method to generate 400 samples in the range [0, 0.8] x [0, 0.8]. Of all the samples, the top 10% were considered as positive

samples and plotted as `+', and the rest were considered as negative samples and plotted as `.'. As we observed, when we simplified all the samples as positive or negative, most samples (in the first two figures) were correctly labeled. When the noise was intense, i.e., the third figure, it could produce a biased training set.



(a) $\sigma^2 = 0.01$ (b) $\sigma^2 = 0.1$

(c) $\sigma^2 = 1$

Figure 3: Effect of noise on classifier

4. The Wisconsin Breast Cancer Epidemiology Simulation: The Wisconsin Breast Cancer Epidemiology Simulation uses detailed individual-woman level discrete event simulation of four processes (breast cancer natural history, detection, treatment and non-breast cancer mortality among US women) to replicate breast cancer incidence rates according to the Surveillance, Epidemiology, and End Results (SEER) Program data from 1975 to 2000. Incidence rates are calculated for four different stages of tumor growth, namely in-situ, localized, regional and distant; these correspond to increasing size and/or progression of the disease. Each run involves the simulation of 3 million women, and takes approximately 8 minutes to execute on a 1GHz Pentium machine with 1Gb of RAM. The four simulated processes overlap in very complex ways, and thus it is very difficult to formulate analytical models of their interactions. However, each of them can be modelled by simulation; these models need to take into account the increase in efficiency of screening processes that has occurred since 1975, the changes in non-screen detection due to increased awareness of the disease and a variety of other changes during that time. The simulations are grounded in mathematical and statistical models that are formulated using a parametrization. For example, the natural history process in the simulation can be modelled using a Gompertzian growth model that is parameterized by a mean and variance that is typically unknown exactly, but for which a range of reasonable values can be estimated. The overall simulation facilitates interaction between the various components, but it is extremely difficult to determine values for the parameters that ensure the simulation replicates known data patterns across the time period studied. In all there are 37 of these parameters, most of which interact with each other and are constrained by linear relationships. Further details can be found in [4].

A score is calculated that measures how well the simulation output replicates an estimate of the incidence curves in each of the four growth stages. Using SEER and Wisconsin Cancer Reporting System (WCRS) data, we generate an envelope that captures the variation in the data that might naturally be expected in a population of the size we simulated The purpose of this study is to determine parameter values x that generate small values for the scoring function. Prior to the work described here, acceptance sampling had been used to fit the parameters. Essentially, the simulation was run tens of thousands of times with randomly chosen inputs to determine a set of good values. With over 450,000 simulations, only 363 were found that had a score no more than 10. That is, for a single replication ξ, 363 vectors x had $F(x,\xi)$ below 10.

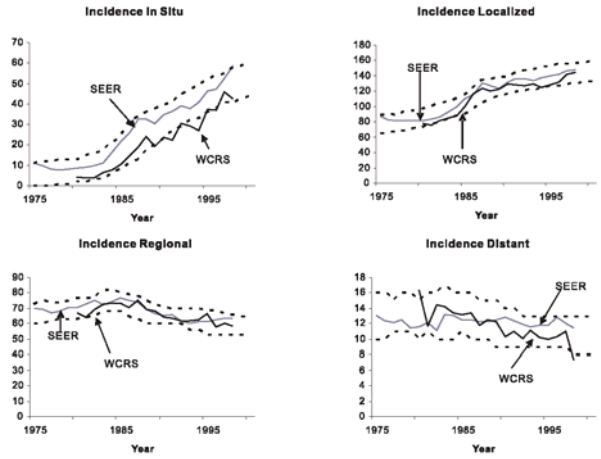


Figure 4: Envelope function data used to determine objective score function

Our first goal was to generate many more vectors x with scores no more than 10. To do this, we attempted to use the classification-based global search based on the scoring function data..

Since we have a vast majority of negative samples, the data-preprocessing step was applied to yield a much balanced training data set. A great portion of the negative samples were removed, resulting a training set containing all the 363 positive samples and 500 negative samples. We trained an ensemble of classifiers that predicted membership of L(c). Each resulting

classifier was evaluated on the testing set using the true positive and negative measures. Classifiers were discarded if the value of TP was less than 0.9 (TN typically is around 0.4). The value was chosen to guarantee the probability of removing positive points in error is small. 100,000 potential values for x were uniformly generated in the feasible domain. Each of the classifiers selected was used to determine if the point x was negative (and hence removed from consideration). At that stage, there were 220 points that were hypothesized to be positive. Evaluating these points via simulation, 195 were found to be in L(10). Thus, with very high success rate (89%), our classification-based global search is able to predict values of x that has a lower score $F(x,\xi)$.

Since the classifiers are cheap to evaluate, this process facilitates a more efficient exploration of the parameter space. Clearly, instead of using a single replication, we could instead replace $F(x,\xi)$ by max $F(x,\xi_i)$ for some i = 1,…,N. In fact this was carried out. The difficulty is that we require replication data (for our experiments we choose N = 10) and we update the definition of L(c) appropriately. However, the process we follow is identical to that outlined above. In our setting, x has dimension 37. Using expert advice, we only allowed 9 dimensions to change; the other 28 values were fixed to the feasible values that have highest frequency of occurrence over the positive samples.
In Phase II local search, we employed the sample-path method [8] with fixed number of replications N=10. Since we carried out multiple local optimizations, the best solution is treated as an approximate solution to the underlying solution of the problem.

1558 samples were selected for further evaluations with replications. In this, we included the original 363 positive samples, another 195 positive samples selected by the classifiers and 1000 negative samples from the original data set. The 1000 negative samples were chosen such that their original scores were less than or equal to 30. In this research, we fixed the other 28 parameters using the `optimal setting' we calculated. We found that 310 out of the 1558 samples had maximum score less than 10.

Given the 10 replications of each sample, we used the DACE toolbox to fit a Kriging model to the data, which we considered as a surrogate function for our objective function. We used the Nelder-Mead [6] simplex method (a derivative-free method) to optimize the surrogate function and generated several local minimizers based on different trial starting points found by the classifier.
The parameter values found using this process outperform all previous values found. Our best parameter generated a score distribution with a mode of 2. Furthermore, expert analysis of various output curves generated from the simulation results with the best set of parameter values confirms the quality of this solution. All the results showed that the surrogate function using the DACE toolbox performs well.

5. Conclusions: We summarize several of our conclusions:

1. The classifier technique is cheap to use and predicts good parameter values very accurately without performing additional simulations.
2. An ensemble of classifiers significantly improves classification accuracy.
3. Imbalanced training data has a detrimental effect on classifier behavior. Ensuring the data is balanced in size is crucial before generating classifiers.

The two-phase optimization framework (WISOPT) has been successfully applied to simulation-based optimization problems [4,7], and specifically to simulation calibration; more generally, the framework is applicable to handle noisy functions. Phase I involves global exploration of the entire domain space to identify promising local subregions which are returned as a collection of samples (over the domain), densely distributed in promising regions. The classification-based global search simplifies the objective function as a 0-1 indicator function, which is approximated by an ensemble of classifiers. A phase transition module is applied to derive the locations of a collection of promising subregions. Phase II applies local optimization techniques to determine optimal solutions in each local subregion. A Matlab implementation of the WISOPT two-phase framework optimization is available.

The WISOPT code couples many contemporary statistical tools (Bayesian statistics and nonparametric statistics) and optimization techniques, and shows effectiveness in processing noisy function optimizations [1]. Moreover, the WISOPT is applicable to deterministic global optimization problem as well.

Acknowledgements: This research was partially supported by National Science Foundation Grants NSF DMI-0521953, DMS-0427689 and IIS-0511905.

References:

[1] G. Deng. Simulation Based Optimization, PhD Thesis, University of Wisconsin, 2007.

[2] G. Deng and M.C. Ferris. Adaptation of the UOBYQA algorithm for noisy functions. In L.F. Perrone, F.P. Wieland, B.G. Lawson J. Liu, D.M.

Nicol, and R.M. Fujimoto, editors, *Proceedings of the 2006 Winter Simulation Conference*, 312-319, 2006.

[3] G. Deng and M.C. Ferris. Variable-number sample-path optimization. To appear in Mathematical Programming, Series B, 2007.

[4] M.C. Ferris, G. Deng, D.G. Fryback, and V. Kuruchittham. Breast cancer epidemiology: calibrating simulations via optimization. *Oberwolfach Reports*, 2:89--92, 2005.

[5] M. Fu. Optimization via simulation: A review. *Annals of Operations Research*, 53:199--248, 1994.

[6] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal,* 7:308--313, 1965.

[7] P. Prakash, G. Deng, M.C. Converse, J.G. Webster, D.M. Mahvi, and M.C. Ferris, Design optimization of a robust sleeve antenna for hepatic microwave ablation. Technical report, Computer Sciences Department, University of Wisconsin, 2007.

[8] S. M. Robinson. Analysis of sample-path optimization. *Mathematics of Operations Research*, 21:513--528, 1996.