Evaluation System for a Bayesian Optimization Service

Ian Dewancker[1], Michael McCourt[1], Scott Clark[1], Patrick Hayes[1], Alexandra Johnson[1] and George Ke[2]

[1]SigOpt, [2]University of Waterloo

Objectives

An evaluation framework for rigorously testing and quickly understanding the impact of changes to the SigOpt optimization service.

- Perform end-to-end testing of service
- Compare between algorithm versions
- Compare against external baselines

Introduction

SigOpt offers an optimization service to help customers tune complex systems, simulations and models. Our optimization engine applies several concepts from Bayesian optimization [1] and machine learning to optimize customers metrics as quickly as possible. In particular, we consider problems where the maximum is sought for an expensive function $f : \mathcal{X} \to \mathbb{R}$,

$$\mathbf{x}_{opt} = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}),$$

SigOpt's core optimization engine is a closed-source fork of the open-source MOE project [2]. The SigOpt service supports a succinct set of HTTP API endpoints for optimizing objective functions.

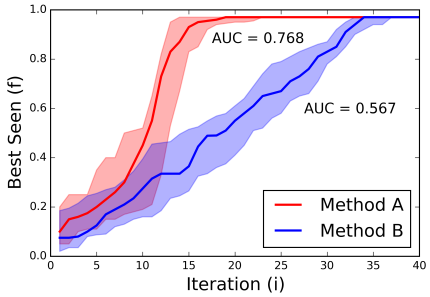

Figure 1: Hypothetical optimization methods A and B both achieve the same **Best Found** of 0.97 after 40 evaluations. Method A however finds the optimum in fewer evaluations.

Metrics

The performance metrics we consider for comparisons on a given objective function are the best value seen by the end of the optimization (**Best Found**), and the area under the best seen curve (**AUC**). The **AUC** metric can help to better differentiate performance, as shown in Figure 1. Optimization algorithms are run at least 20 times on each function. The distributions of the performance metrics are compared using the non-parametric **Mann-Whitney** U **test**, used in previous studies of Bayesian optimization methods [3] [4].

Benchmark Suite

The tests for our evaluation system consist of closed-form optimization functions [5] which are extensions of an earlier set proposed for black-box optimization evaluation. The benchmark suite is open source.

Infrastructure

Lightweight function evaluation processes are run concurrently on a master machine with many cores. External baseline methods including TPE, SMAC, Spearmint, particle swarm optimization and random search are run locally on master.

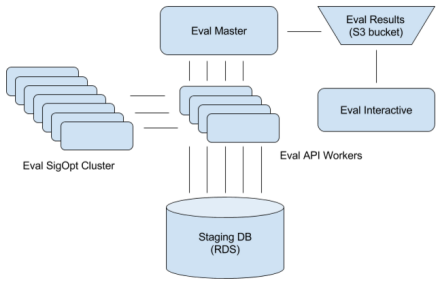

Figure 2: Architecture of evaluation system infrastructure

Each evaluation process communicates with an on-demand cluster of SigOpt API workers, which in turn co-ordinate each optimization request with a cluster of instances running the SigOpt optimization engine as well as a database used by the service. This design closely follows the architecture of the production system. Result data is archived in a simple, extensible JSON format. The result data and summarizations are then visualized on a small web application.

Visualization Tools

To better summarize the relative performance between two methods (A, B) using metric M, we create histograms of the p-values returned by the Mann-Whitney U tests. Test functions are first partitioned by the relative expected value of the metric M, then histograms are generated for the two sets.

$$\mathbf{wins}(A > B)_M = \{\ func \mid \mathbb{E}[M_A] > \mathbb{E}[M_B]\ \}$$
$$\mathbf{wins}(B > A)_M = \{\ func \mid \mathbb{E}[M_B] > \mathbb{E}[M_A]\ \}$$

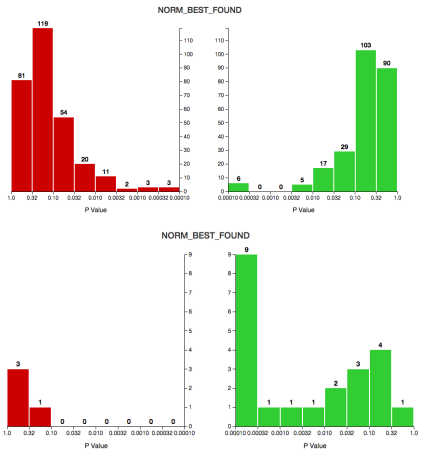

Figure 3: *Above*: Histograms for two comparable methods. *Below*: Method B (in green) has highly significant wins on test functions, method A (in red) has only low significance wins

For inspecting comparative performance on individual functions, we plot the best value of the objective metric seen after each function evaluation.

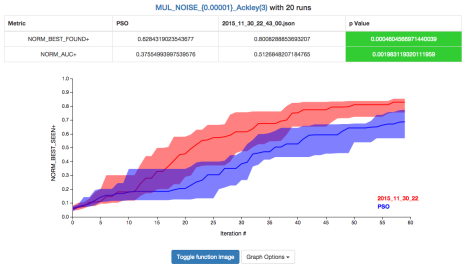

Figure 4: Comparative best seen trace on test function

Figure 4 shows the best seen traces of SigOpt (in red) and particle swarm optimization (in blue) on a given test function.

Conclusions

Our evaluation system has been become a valuable analysis tool when considering algorithm or system changes to the SigOpt optimization service. Data driven performance analysis is an effective way to enable faster iteration and evaluation of a wide spectrum of ideas. The system continues to guide improvements to the core SigOpt service by providing empirical comparisons between internal changes, alternative optimization methods, as well helping to expose errors and bugs.

References

[1] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[2] Scott Clark, Eric Liu, Peter Frazier, JiaLei Wang, Deniz Oktay, and Norases Vesdapunt. MOE: A global, black box optimization engine for real world metric optimization. https://github.com/Yelp/MOE, 2014.

[3] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*, pages 507–523. Springer, 2011.

[4] Ian Dewancker, Michael McCourt, Scott Clark, Patrick Hayes, Alexandra Johnson, and George Ke. A strategy for ranking optimization methods using multiple criteria. In *ICML AutoML Workshop*, 2016.

[5] Michael McCourt. Optimization Test Functions. https://github.com/sigopt/evalset, 2016.

Contact Information

- Web: http://www.sigopt.com
- Email: contact@sigopt.com