6th ICML Workshop on Automated Machine Learning (2019)

Explainability Constraints for Bayesian Optimization

Michael Y. Li MYLI@PRINCETON.EDU
Department of Computer Science
Princeton University
Princeton, NJ 08544, USA

Ryan P. Adams RPA@PRINCETON.EDU
Department of Computer Science
Princeton University
Princeton, NJ 08544, USA

Abstract

Bayesian optimization is a powerful tool for solving optimization problems with noisy, expensive objective functions. Despite the success of Bayesian optimization in settings ranging from hyperparameter tuning to drug discovery, human operators are often reluctant to trust the decisions made by Bayesian optimization systems, due to the opacity of the selection procedure. This paper introduces techniques for achieving "explainable" Bayesian optimization and evaluates the impact of these techniques on empirical performance.

1. Introduction

The problem of minimizing an objective function $f(x)$, where the function captures the performance of a system, is ubiquitous. This problem becomes challenging when $f(x)$ has no known parametric form and the derivatives are inaccessible. In these situations, the function can only be evaluated point-wise with noisy observations. When function evaluation is also expensive, it is crucial to evaluate the function strategically. One approach to this problem is to treat $f(x)$ as a black box, sample noisy points from this black box, and use observations to build a model of $f(x)$. Bayesian optimization (Mockus et al., 1978) is one such approach, that explicitly reasons about the posterior over functions, given previous observations and a prior over the objective. Bayesian optimization has seen success in tuning hyperparameters of machine learning models (Shahriari et al., 2016), automated chemical design (Gomez-Bombarelli et al., 2016), and materials science (Gomez-Bombarelli et al., 2018). Even though Bayesian optimization can outperform human experts (Snoek et al., 2012), when experiments are extremely expensive, human users hesitate to leave decisions to an opaque algorithm. However, these expensive problem settings are precisely when Bayesian optimization is most valuable.

This work seeks to improve the human interface with Bayesian optimization by providing greater transparency and interpretability of its decisions. This effort is part of a larger trend within the machine learning community towards fairness and transparency, as machine learning models play a larger role in society (Doshi-Velez and Kim, 2017). While interpretability has largely been studied in the context of explaining supervised models (Guidotti et al., 2018), there is limited work on interpretability in sequential-decision making settings. This work seeks to close that gap in expensive model-based optimization tasks.

2. Bayesian Optimization Setup

In Bayesian optimization, the goal is to minimize a function $f(x)$ on a bounded set $\mathcal{X}$, where $f(x)$ can only be evaluated point-wise and evaluations are expensive. Bayesian optimization constructs a probabilistic model, known as the surrogate model, to approximate $f(x)$ and explicitly characterize the uncertainty about $f(x)$ across $\mathcal{X}$. This surrogate model is used to inform intelligent sampling from $f(x)$ through an acquisition function that balances exploration of the domain and greedy exploitation of good candidates.

The Gaussian process (GP) prior is used as the surrogate model due to its flexibility and expressiveness. The GP is defined by the property that any finite set of N points $\{x_n \in \mathcal{X}\}$ is distributed as a multivariate Gaussian on $\mathbb{R}^N$. The nth of these points is taken as the value of $f(x_n)$. Computing marginal and posterior distributions of Gaussian process can be done in closed form. This allows efficient updating of the surrogate model given new information from experiments.

The acquisition function $a : \mathcal{X} \to \mathbb{R}^+$ determines which point in $\mathcal{X}$ to evaluate next given the previously evaluated experiments and the surrogate model. The next point is determined by:

$$x_n = \arg\max_x a(x)\,.$$

Define the predictive mean function of the Gaussian process prior as $\mu(x; \{x_n, y_n\}, \theta)$ and the variance function as $\sigma^2(x; \{x_n, y_n\}, \theta)$ Although there are many acquisition functions in the literature with different appealing properties, e.g., Frazier (2018), Brochu et al. (2010), Hernndez-Lobato et al. (2016), in this paper, the focus is on the expected improvement (EI) acquisition function (Mockus et al., 1978) since it performs well empirically and does not have any parameters that require tuning. Expected improvement selects the point that "improves upon" the best minimum value of f observed so far.

$$a_{EI}(x; \{x_n, y_n\}, \theta) = \sigma(x; \{x_n, y_n\}, \theta)(\gamma(x)\Phi(\gamma(x)) + \mathcal{N}(\gamma(x); 0, 1))\,.$$

3. Explainability Constraints for Bayesian Optimization

The challenge that humans often have when interacting with Bayesian optimization systems is that the exploratory decisions seem *ad hoc.* The decision that maximizes the acquisition function may feel like it was "pulled out of thin air". Thus our goal in this work is to accompany every suggestion with an explanation that can be provided to the user. To achieve this explainability, we introduce a portfolio of heuristics with accompanying natural language descriptions, that correspond to human-understandable criteria for selecting candidate points. We call these heuristics *explainability constraints.* These heuristics can all be framed in terms of previously-evaluated points, where $\mathcal{X} \subset \mathbb{R}^D$. The first heuristic is *similarity to previous evaluation.* This heuristic corresponds to explanations like: "Experiment X wasn't the best so far, but there may be some interesting candidates nearby. Here is an experiment that's similar to X but with some small tweaks." The second heuristic is called *coordinate descent* and corresponds to explanations like: "Experiment X was good but I don't know what parameters were most important. Let's try an experiment where we fix every parameter except parameter z and perturb just z." The last heuristic is *interpolation* which corresponds to explanations like: "'Experiments X and Y were both good results, but were in different parts of the space. Let's see what happens if you move smoothly from X to Y. Here is a candidate that lies in between those previous experiments."

As written, these heuristics appear as *a priori* criteria for selection. However, since any experiment can play the role of X or Y in the above explanations, then there are a large number of possible points that would allow for such an explanation *post hoc.* We examine these more closely below.

Perturbation of Previous Evaluation Humans are suspicious of exploratory decisions that deviate significantly from previous evaluations. However, users may be comfortable with evaluating candidates similar to ones they have seen before. The perturbation constraint limits evaluations to the union of neighborhoods around previous points:

$$\mathcal{X}_{\mathsf{perturb}} \coloneqq \bigcup_{n=1}^{N} \mathcal{B}_\epsilon(x_n) \qquad \text{where} \quad \mathcal{B}_\epsilon(x_n) \coloneqq \{x : ||x - x_n||_\infty \leq \epsilon\} \tag{1}$$

Coordinate descent One intuitive approach a human might take to learn the behavior of a black-box function is to vary one parameter while keeping all other parameters fixed, and then repeat this procedure for every parameter of the function. The coordinate descent constraint formalizes this intuitive approach, but improves it with the reasoning of Bayesian optimization. This constraint corresponds to all possible single-parameter perturbations of previous points. Let $x_{d,\min}$ and $x_{d,\max}$ denote the minimum and maximum allowed value of the *dth* dimension:

$$\mathcal{X}_{\mathsf{coord}} \coloneqq \bigcup_{n=1}^{N} \bigcup_{d=1}^{D} \mathcal{C}_d(x_n) \qquad \text{where} \quad \mathcal{C}_d(x_n) \coloneqq \{x : x_{d,\min} \leq x_{n,d} \leq x_{d,\max}\} \tag{2}$$

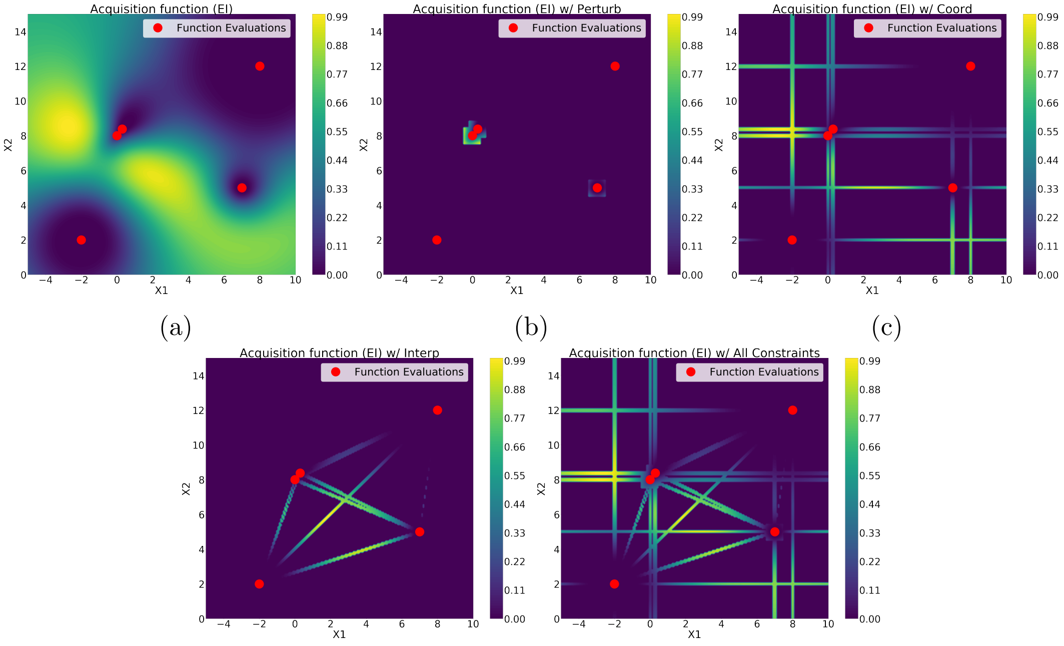
Interpolation between Previous Evaluations The interpolation constraint restricts future evaluations to convex combinations of any two previously evaluated points. Human users may find "smooth blends" of previous decisions explainable because these decisions aim to understand uncertain regions between well understood regions.

$$\mathcal{X}_{\mathsf{interp}} \coloneqq \bigcup_{n=1}^{N-1} \bigcup_{n'=n+1}^{N} \bigcup_{\alpha \in [0,1]} \alpha x_n + (1-\alpha) x_{n'} \tag{3}$$

Given a set of previous evaluations $\{x_n, y_n\}_{n=1}^N$, the set of explainable points is then the union of the sets defined above, i.e., the set of points where at least one explanation can be used:

$$\mathcal{X}_{\mathsf{explainable}} = \mathcal{X}_{\mathsf{perturb}} \cup \mathcal{X}_{\mathsf{coord}} \cup \mathcal{X}_{\mathsf{interp}} \,. \tag{4}$$

When candidates are selected, the acquisition function is maximized subject to the constraint that the maximum be in the set of explainable points $\mathcal{X}_{\mathsf{explainable}}$. Thus, identifying the best explainable candidate is a constrained optimization problem where we maximize $a(x)$ subject to $x \in \mathcal{X}_{\mathsf{explainable}}$. To solve this, one can use Limited-memory BFGS to identify the best candidate separately for each constraint set and then choose the best candidate



(a) (b) (c)

(d) (e)

Figure 1: EI unconstrained (a), perturbation constraint (b), coordinate descent constraint (c), interpolate constraint (d), and all three (e).

among all constraint sets. Instead of this, to obtain the best candidate from a constraint set, we densely sampled points from each constraint set and took the best among all sampled points. This was empirically faster and produced similar results.

Figures 1b, 1c, 1d, and 1e are visualizations of the regions captured by $\mathcal{X}_{\mathsf{perturb}}$, $\mathcal{X}_{\mathsf{coord}}$, $\mathcal{X}_{\mathsf{interp}}$, and $\mathcal{X}_{\mathsf{explainable}}$. For illustration, any point outside a constrained region has an acquisition value of zero. Figure 1a shows the unconstrained acquisition function. Figure 1b shows that any point not within a small neighborhood of the previous evaluations has acquisition value of zero. Figure 1c shows that the feasible region corresponds to all points that can be constructed by fixing the X_1/X_2 dimension of a previous point while varying the X_2/X_1 dimension. Figure 1d shows the acquisition function subject to the interpolate constraints. The feasible region corresponds to any points that are convex combinations of any two previous evaluations. Figure 1e shows that collectively the explainability constraints capture many points with high acquisition values. Nevertheless, these plots show explainability restricts where sampling can occur. A natural question is: *how does explainability damage performance?*

4. Results

Although explainability constraints can build user trust and understanding, it is critical to understand how these constraints damage the performance of Bayesian optimization. In this section, we present empirical results comparing various explainability constrained EI algorithms against baseline EI and random exploration. We plot the minimum function value observed so far against the number of function evaluations, averaged over different trials, along with the standard error. We used the open-source Python library GPyOpt to aid our analysis. The benchmark functions are the Branin-Hoo function, tuning the hyperparameters in Support Vector Regression (SVR), and finding the minimum of a 7-dimensional random function drawn from a Gaussian process prior. See appendix for details.

Figure 2a show the perturbation constraint slows the convergence rate relative to baseline EI for the Branin-Hoo problem. However, the final minimum value identified is comparable. In Figure 2b, we see a slight damage to the final minimum value identified for the SVR hyperparameter tuning problem. Figure 2c shows that the perturbation constraint causes convergence to a significantly smaller value than baseline EI for the 7-dimensional GP function. However, perturbation constrained EI still identifies a significantly smaller value than random exploration of the space.

In Figures 2a and 2b, coordinate descent constrained EI identifies the minimum as quickly as baseline EI. This performance holds reasonably well even in higher dimensions. For the 7-dimensional GP function, coordinate descent constrained EI converges to a minimum function value that is nearly as small as the value identified by baseline EI and significantly smaller than the value identified by random exploration and the perturbation constrained EI (Figure 2c).

The interpolation constraint needs to be combined with another constraint for good performance. Otherwise, the feasible region will be limited to the convex combinations of each pair of previous evaluations. Figures 2a, 2b, and 2c show that the performance of interpolate constraint combined with coordinate descent constraint is comparable to the coordinate descent constraint both with respect to convergence rate and minimum value identified. We do not include the perturbation constraint because experiments showed performance was insensitive to including it when the coordinate descent constraint was already incorporated.

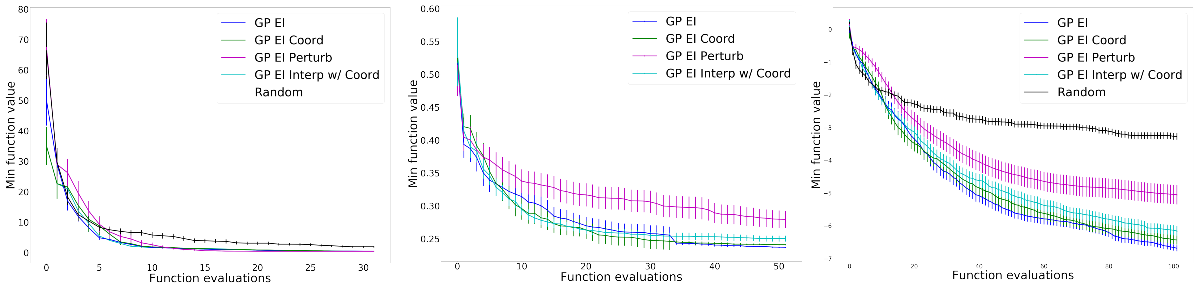
The results in Figure 2 demonstrated that restricting function evaluations to $\mathcal{X}_{\mathsf{coord}}$ did not damage performance significantly. Since the hyperparameters were the same, only the *acquisition values* of sampled points can account for performance discrepancies. Figure 3 compares the acquisition value of the best candidate in $\mathcal{X}$ and the best candidate in the constrained set $\mathcal{X}_{\mathsf{coord}}$. We define the “EI gap” as the difference in acquisition value between the best candidate from the restricted set and the best candidate from the entire domain. This gap quantifies how much the explainability constraints reduce the value of the best possible candidate. Figures 3a and 3b show that the mean EI value of the best candidate from the entire domain is consistently higher than the mean EI value of the best candidate from $\mathcal{X}_{\mathsf{coord}}$. However, this gap is statistically insignificant. Given that the acquisition function governs performance, the small EI Gap can account for the performance results in Figures 2a and 2b. This general behavior holds for higher dimensions. Figure 3c shows the EI Gap is mostly negligible for the 7-dimensional benchmark. However, in some function evaluations in the start and the end, there are significant gaps.

5. Discussion/Conclusion

Results showed explainability had limited damage to optimization performance. Coordinate descent constrained EI performed particularly well across all benchmarks. Figure 3 demonstrated that the best candidate in the explainable set was consistently as good as the best candidate in the entire domain. What might account for this? Bayesian optimization consists of two main regimes: the exploration phase and exploitation phase. In the exploration phase, little is known about the true function so any exploration of uncertain regions is near optimal. In the exploitation phase, the mean drives candidate selection so most good candidates will be close to previous evaluations. The coordinate descent constraint is flexible enough to allow both global exploration and local exploitation so the constrained set can consistently capture valuable points.

Combining the interpolate constraint with coordinate descent led to comparable performance to just the coordinate descent constraint. Combining two constraints has the advantage of providing more diverse explanations. Coordinate descent constrained EI outperformed perturbation consistently. Pertubation constrained EI also has the disadvantage that ϵ needs to be tuned for both performance and explainability, which requires user studies.

We introduced explainability constraints for Bayesian optimization. Benchmarking showed explainability had limited damage on empirical performance. However, user studies are needed to determine whether explainability constraints improve interpretability.

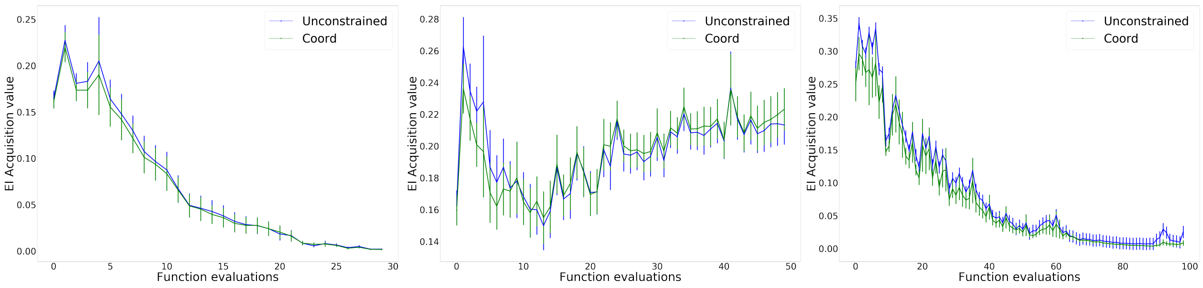


(a)

(b)

(c)

Figure 2: Benchmarks for Branin-Hoo (a), SVR (b), and 7-dimensional GP (c).



(a)

(b)

(c)

Figure 3: EI Gap for coordinate descent constraint on Branin-Hoo (a), SVR (b), and 7-dimensional GP (c).

Acknowledgements

This research was funded in part by JPMorgan Chase & Co. Any views or opinions expressed herein are solely those of the authors listed, and may differ from the views and opinions expressed by JPMorgan Chase & Co. or its affiliates. This material is not a product of the Research Department of J.P. Morgan Securities LLC. This material should not be construed as an individual recommendation for any particular client and is not intended as a recommendation of particular securities, financial instruments or strategies for a particular client. This material does not constitute a solicitation or offer in any jurisdiction.

References

Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. art. arXiv:1012.2599, 2010.

Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv e-prints*, art. arXiv:1702.08608, Feb 2017.

Peter I. Frazier. A Tutorial on Bayesian Optimization. art. arXiv:1807.02811, 2018.

Rafael Gomez-Bombarelli, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, David Duvenaud, Dougal Maclaurin, Martin A. Blood-Forsythe, Hyun Sik Chae, Markus Einzinger, Dong-Gwang Ha, Tony Wu, Georgios Markopolous, Soonok Jeon, Hosuk Kang, Hiroshi Miyazaki, Masaki Numata, Sunghan Kim, Wenliang Huang, Seong Ik Hong, Marc Baldo, Ryan P. Adams, and Alan Aspuru-Guzik. Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nature Materials*, 15(10):1120–1127, 2016.

Rafael Gomez-Bombarelli, Jennifer Wei, David Duvenaud, Jose Miguel Hernndez-Lobato, Benjamin Snchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alan Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5):93:1–93:42, August 2018.

Daniel Hernndez-Lobato, Jos Miguel Hernndez-Lobato, Amar Shah, and Ryan P. Adams. Predictive entropy search for multi-objective bayesian optimization. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. *Towards Global Optimization*. 1978.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NIPS) 25*, 2012.

Description of Problems

The first problem is optimization of the Branin-Hoo function, which is a classic benchmark function used in Bayesian optimization. We average our results over 30 different sample paths. The second problem we consider is tuning the hyperparameters of the Support Vector Regression function on the Olympic marathon dataset available in GPy.The hyperparameters considered are the penalty parameter of the error term, the kernel coefficient, and epsilon in the Epsilon-SVR model. We split the original dataset into the training data (first 20 data points) and testing data (last 7 data points). The performance of SVR (predicting gold-medal times based on the year) is evaluated in terms of Rooted Mean Squared Error (RMSE) on the testing data. We average our results over 40 different sample paths. The last optimization problem was optimizing a function generated from the mean of a 7-dimensional Gaussian Process with an RBF kernel. This function was generated by first fitting a Gaussian Process to a sinuosoidal function and then choosing the mean function as the truth. We average over 50 sample paths.