Bayesian Optimization Tutorial

Module 3: The BO Feedback Loop in Practice

Joel Paulson
Assistant Professor, Department of Chemical and Biomolecular Engineering, The Ohio State University

Great Lakes PSE Student Workshop, 2023

For copies of slides & code, see
https://github.com/joelpaulson/Great_Lakes_PSE_Workshop_2023

Bird's-eye View of Bayesian Optimization

while {budget not exhausted}

Fit a Bayesian machine learning model
(usually Gaussian process regression)
to observations {x, f(x)}

Find x that maximizes acquisition(x, posterior)

Sample x & then observe f(x)

end

Standard Implementation

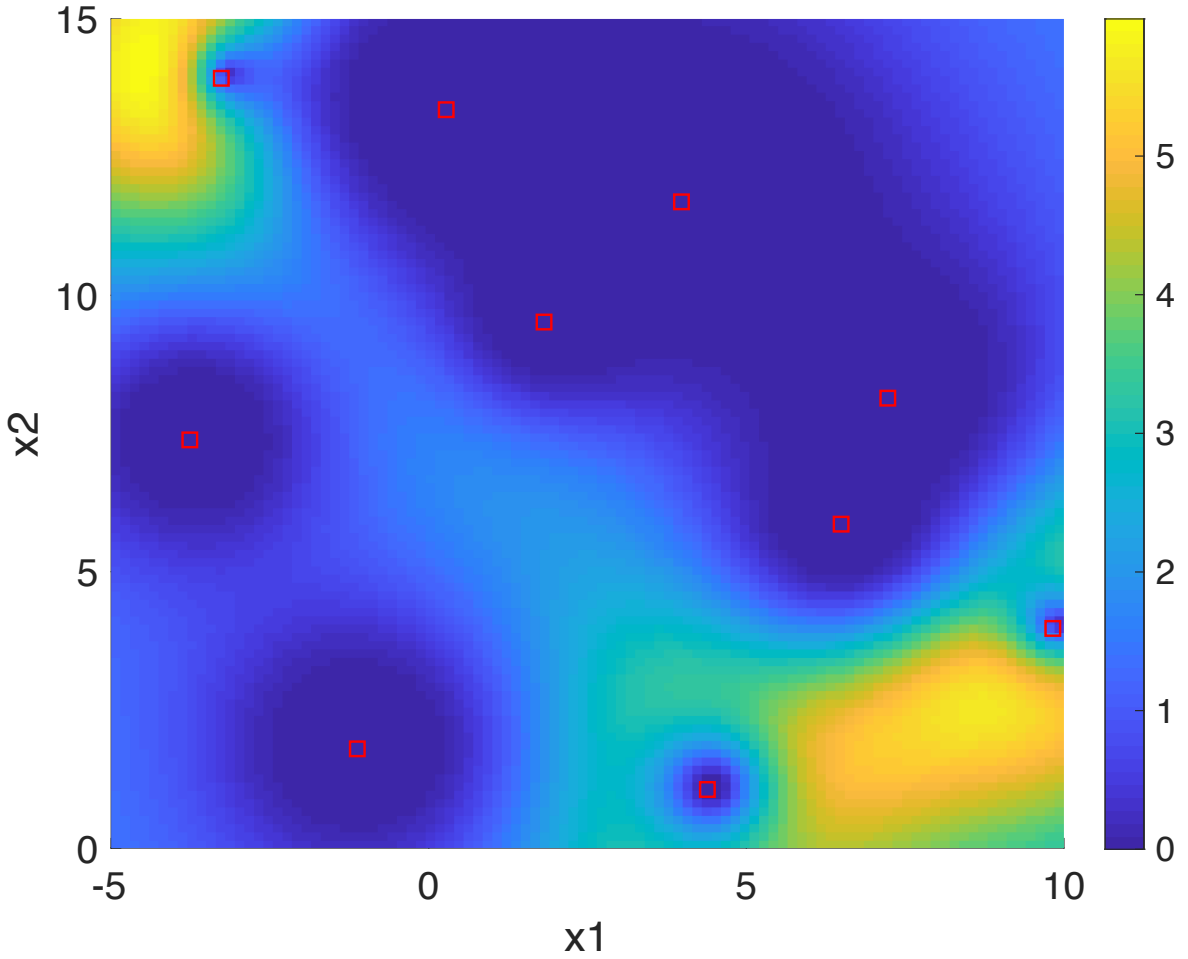
Two Major Tasks at Each Iteration in BO

1. Train hyperparameters of Gaussian process model
   - The more accurate estimates we achieve for kernel + hyperparameters, the better the decision we can make for the next sample
   - In practice, we re-optimize the hyperparameters at every iteration (given the most recent data) by maximizing the log-likelihood function
   - A trick to reduce cost is to "warm start" the initial guess for the hyperparameters, which works well once they have roughly "stabilized"

2. Maximize the acquisition function, $x_{n+1} \in \operatorname{argmax}_{x \in \Omega} \alpha_n(x)$
   - Many methods exist, no consensus in literature (use your favorite method)
   - Complexity of GP model + operators in acquisition function both important

Common Practice for Maximizing Acquisition Function

- We expect $\alpha_n(x)$ to be non-convex but almost always differentiable
- Easily apply local optimization methods (e.g., L-BFGS, IPOPT)
- Important to perform some type of multi-start procedure to globalize
  - How effective is this method?



How to Measure Performance?

- The majority of BO papers use simple regret, which is the minimum over the recommended point after a finite number of iterations

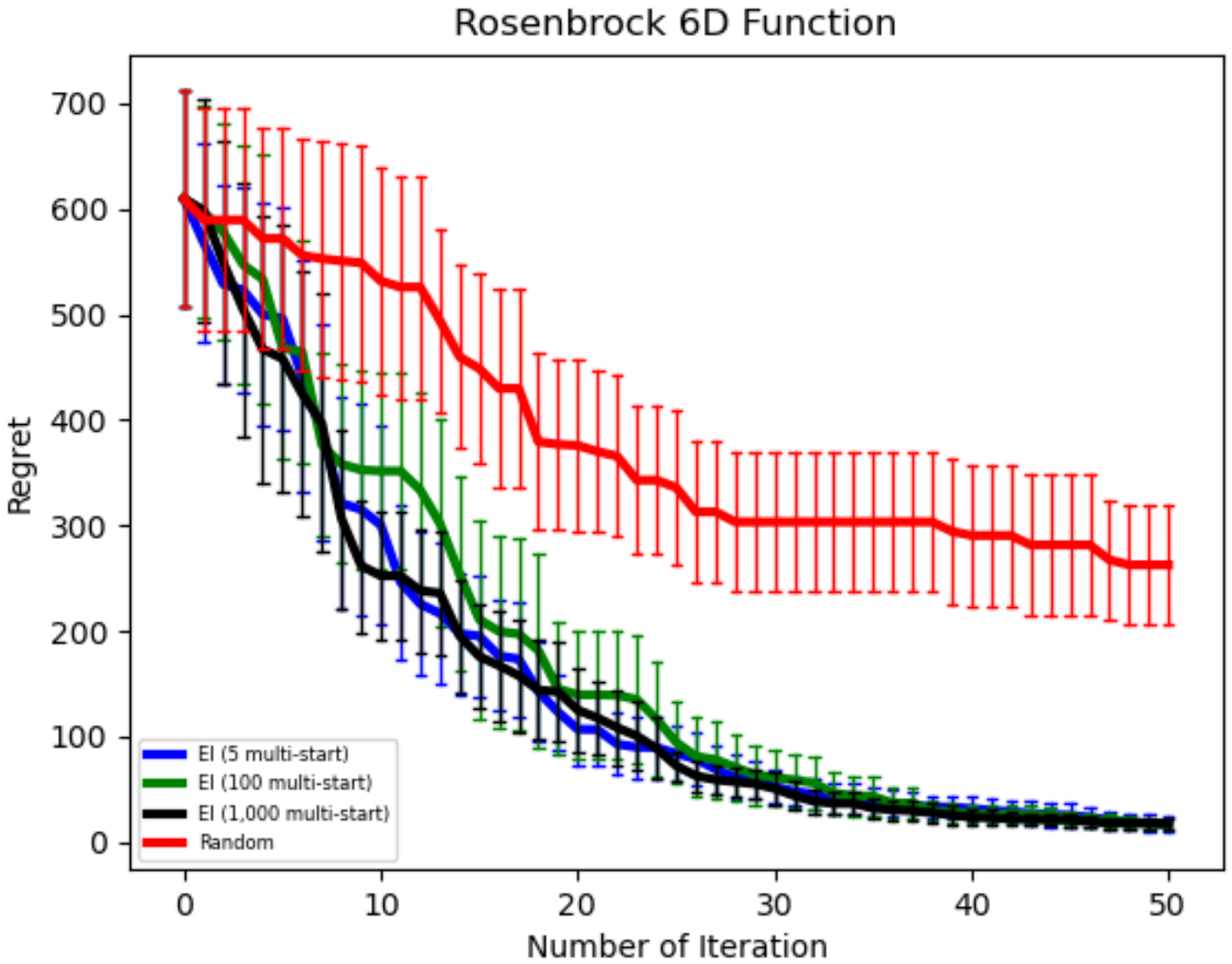
$$\text{SimpleRegret}_n = \min_{i \in \{1,\ldots,n\}} r_i = \min_{i \in \{1,\ldots,n\}} \{f(x_i) - \underline{f(x^\star)}\}$$

Use best known solution when true solution unknown

- Since the regret sequence depends on the initial data, it is very common to do 50-100 replicates of the entire procedure (with initial data randomly generated) to assess average performance
  - We cannot conclude that every run performs – not obvious what the distribution of performance is either, so often do not try to estimate

Common Practice for Maximizing Acquisition Function

[100 replicates over initial data]



random

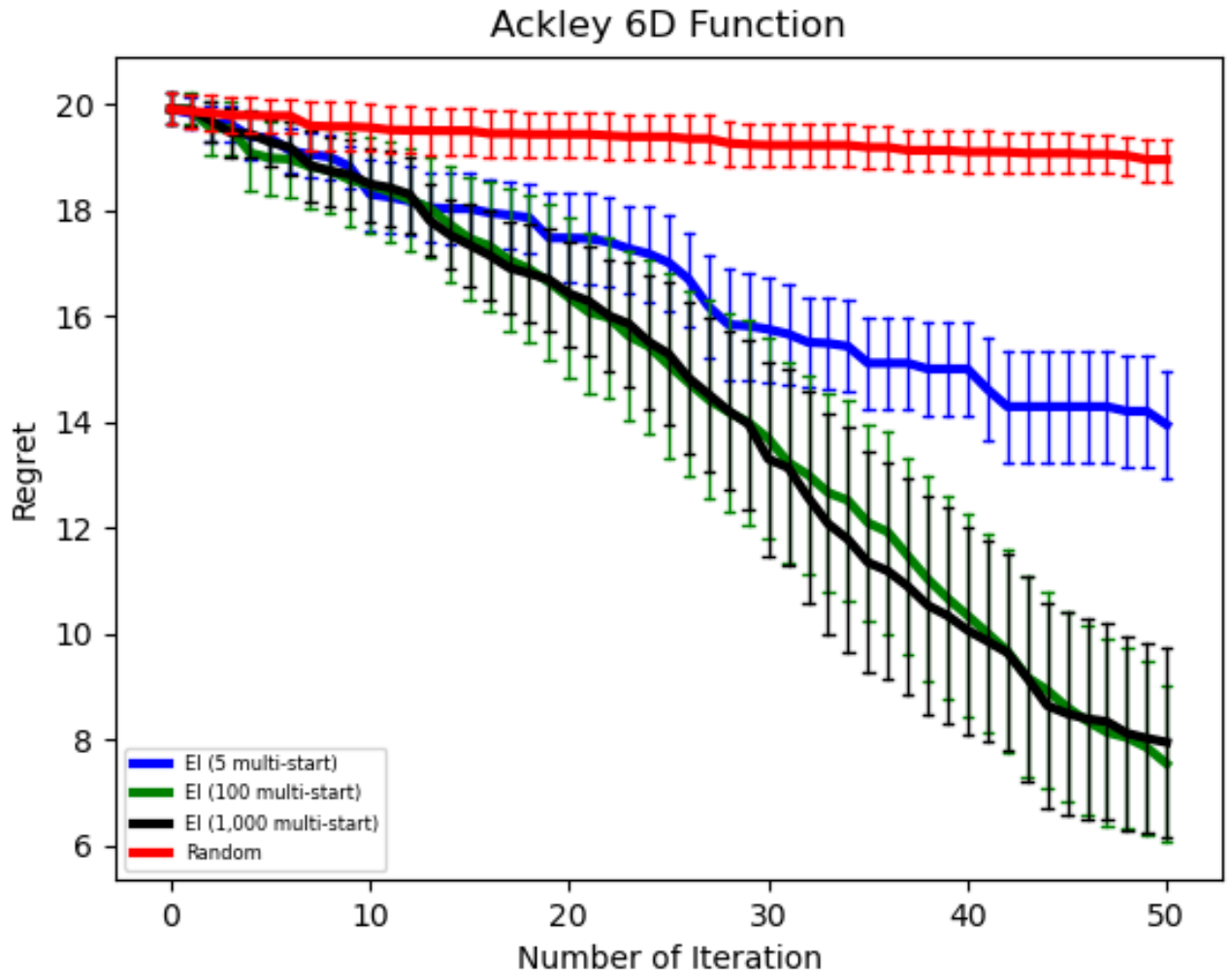
5 restarts

100 restarts

1000 restarts

Common Practice for Maximizing Acquisition Function

[100 replicates over initial data]



random

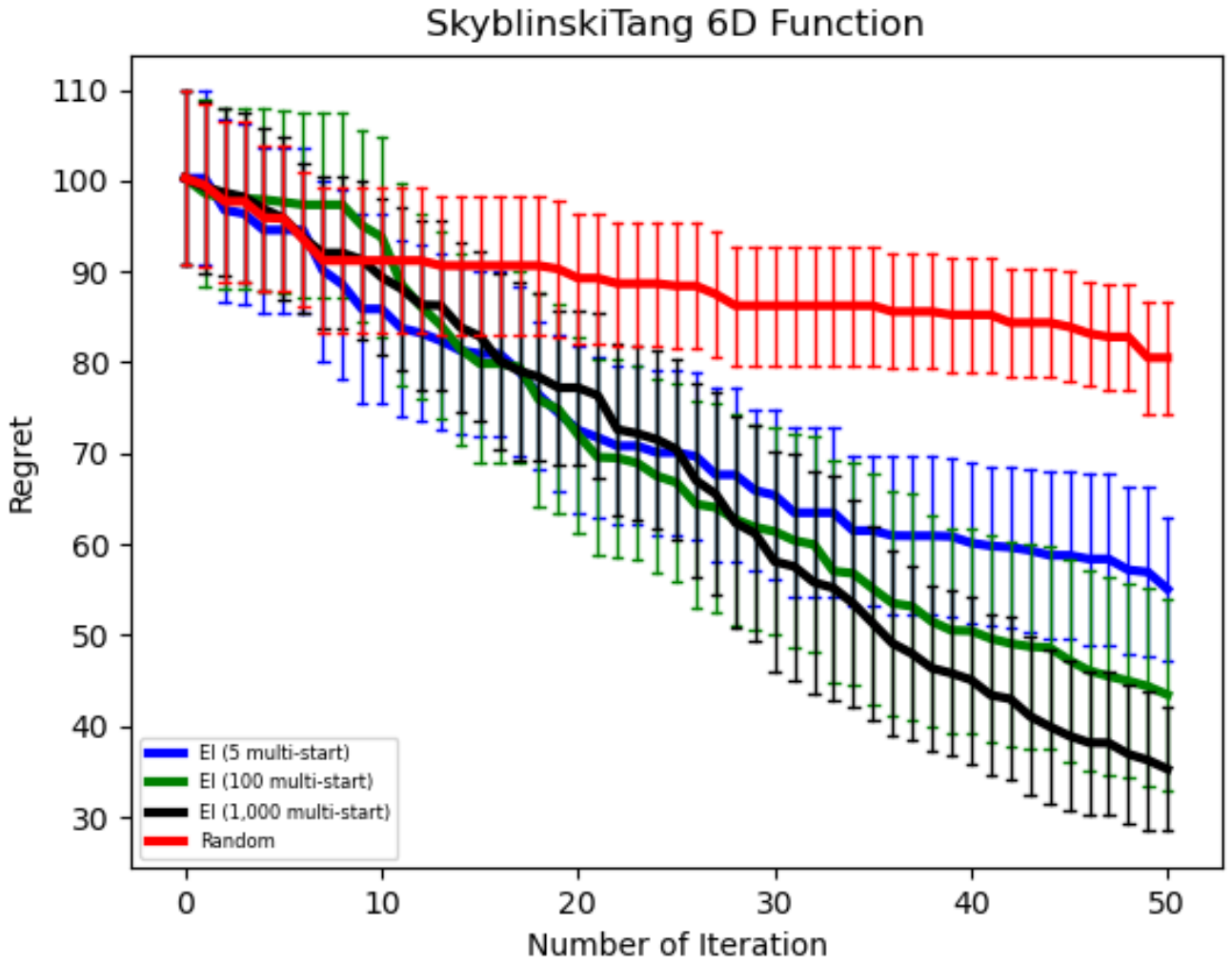
5 restarts

100 restarts

1000 restarts

Common Practice for Maximizing Acquisition Function

[100 replicates over initial data]



random

5 restarts

100 restarts

1000 restarts

Why Not Global Optimization?
Short Answer: Existing Methods Struggle

- McCormick relaxations end up being very weak for the posterior mean and covariance functions since they are the sum over several terms that can have alternating sign

- Would be great if we could (cheaply) construct underestimators that are not too weak → active research area in my group

- Since multi-start is trivially parallelizable (run local solves in parallel), it seems that is best approach to use for now
  - First-order methods, like ADAM, also can take advantage of GPU acceleration, so end up having fast wall-times in practice

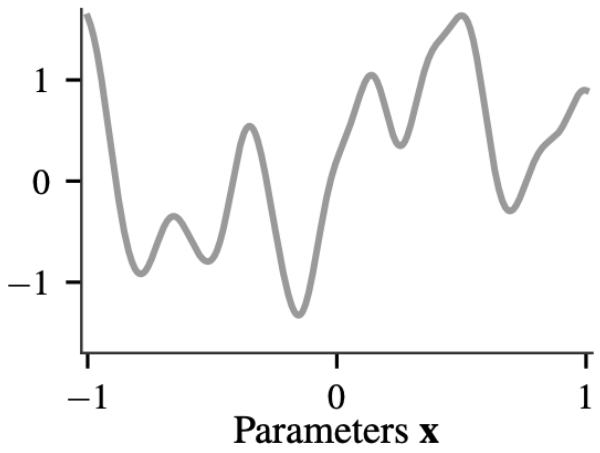
Some Adaptive Modifications to (Potentially) Improve Performance

Dynamic Kernel Selection

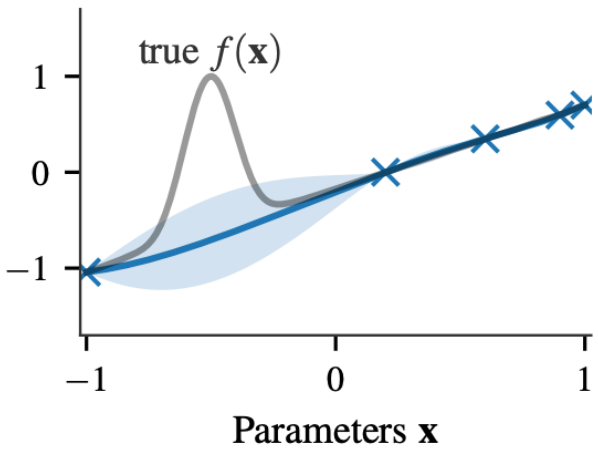
- Normally, we pick a single kernel and keep it fixed at every iteration
- There has been recent work suggesting that some performance gains can be obtained by training multiple GP models (with different kernels) and using some criteria to dynamically select the one to use at each iteration
- Heuristic in nature, so more work needed to find best ways to systematically select between GP models → usually based on some random select process to induce more exploration in the BO process

Roman, Ibai, et al. *BayesOpt 2014: NIPS Workshop on Bayesian Optimization*, 2014

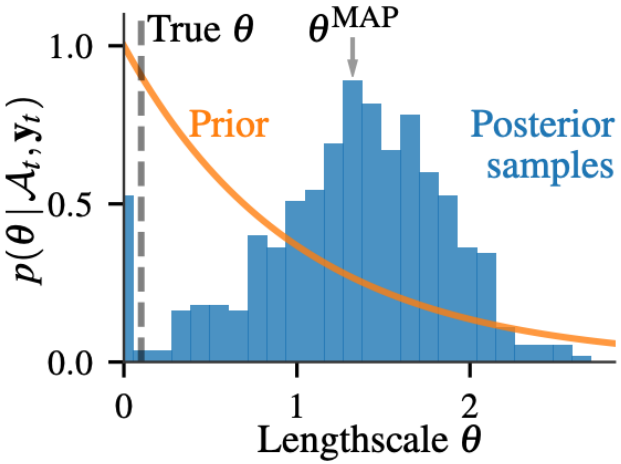
Adaptation of Kernel Hyperparameters



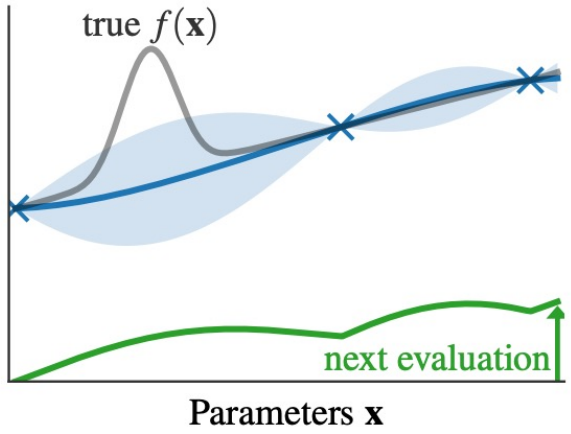
(a) Sample from GP prior.



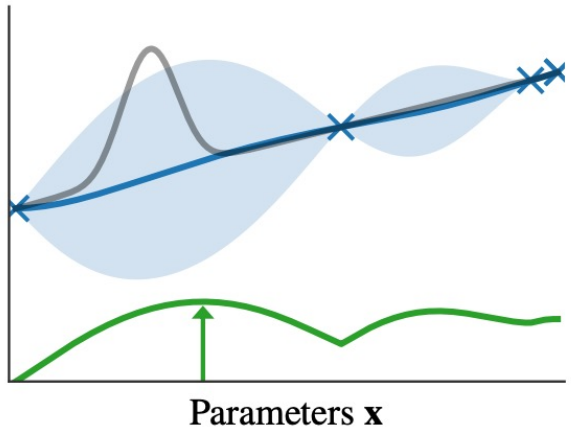
(b) GP estimate (RKHS).



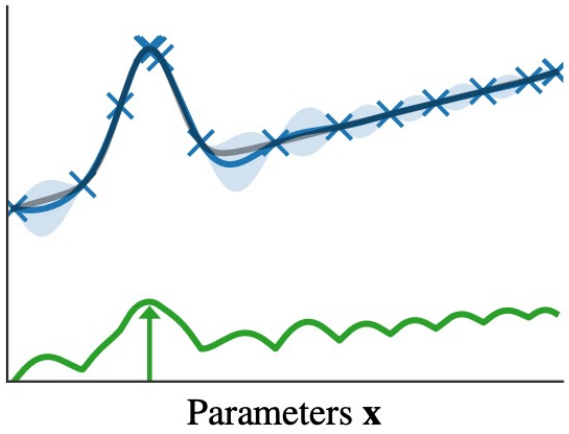
(c) Lengthscale distribution.



(a) Stuck in local optimum.



(b) Expanding the function class.



(c) Global optimum found.

Berkenkamp, et al. *Journal of Machine Learning Research*, 2019

Dynamic Scheduling of Acquisition Functions

- General practice in the BO literature has been for a practitioner to pick their favorite acquisition and use it for the entire optimization process

- Recent work has discussed the value of adopting an adaptive sampling that chooses different acquisition functions at different iterations instead of attempting to pick "the best one"

- Different sampling strategies exist, a simple one is to assume m acquisition functions with weights $\{w_i\}_{i=1}^{m}$. The probability of sampling i^{th} acquisition is $w_i / \sum_{i=1}^{n} w_i$ and, if the acquisition takes a successful move, then the weight is updated

Kandasamy, et al. *Journal of Machine Learning Research*, 2020

Knowledge Gradient Maximization [More Advanced Optimization]

Let's recall Knowledge Gradient (KG)

- In Module 2, we saw that the KG acquisition function is:

$$\mathrm{KG}_n(x) = \mathbb{E}_n\{\mu_n^\star - \mu_{n+1}^\star | x_{n+1} = x\}$$

$$= \mathbb{E}_n\left\{\cancel{\mu_n^\star} - \min_{x' \in \Omega} \mu_{n+1}(x') | x_{n+1} = x\right\}$$

- Our goal is to maximize this function, so can ignore constant and convert the max to a min (due to the -1)

Maximization of KG is a two-stage stochastic program

$$\min_{x\in\Omega} \mathbb{E}_n \left\{ \min_{x'\in\Omega} \mu_{n+1}(x') | x_{n+1} = x \right\}$$

How can we express the expectation in terms of things we can compute?

$$\mu_{n+1}(x') = \mu_n(x') + \tilde{\sigma}_n(x', x_{n+1})Z, \qquad Z \sim \mathcal{N}(0,1)$$

$$\tilde{\sigma}_n(x, x') = \frac{k_n(x, x')}{\sqrt{k_n(x', x') + \sigma^2}}$$

Maximization of KG is a two-stage stochastic program

$$\min_{x_{n+1}\in\Omega} \mathbb{E}_Z \left\{ \min_{x'\in\Omega} \{\mu_n(x') + \tilde{\sigma}_n(x', x_{n+1})Z \right\}$$

- Two main approaches for solving this problem:

1. Stochastic gradient descent (SGD) (+ envelope theorem)

2. Sample average approximation (SAA)

this has become more popular recently

Sample average approximation for KG acquisition

$$\min_{x_{n+1}\in\Omega} \frac{1}{N}\sum_{i=1}^{N}\left\{\min_{x^{(i)}\in\Omega}\{\mu_n(x^{(i)})+\tilde{\sigma}_n(x^{(i)},x_{n+1})Z_i\}\right\}$$

The point that minimizes the next mean function depends on our sample selection

$$\min_{x_{n+1},x^{(1)},\ldots,x^{(N)}\in\Omega} \frac{1}{N}\sum_{i=1}^{N}\{\mu_n(x^{(i)})+\tilde{\sigma}_n(x^{(i)},x_{n+1})Z_i\}$$

"here-and-now" "wait-and-see"

CODE REVIEW

Workshop Schedule

9:00 – 9:20	Introduction: Why Go Beyond Traditional Optimization?
9:20 – 10:20	Module 1: Probabilistic Surrogate Modeling*
10:20 – 10:30	Break
10:30 – 11:20	Module 2: Quantifying the Value of Information*
11:20 – 12:20	Module 3: The BO Feedback Loop*
12:20 – 12:30	Break
12:30 – 1:00	Module 4: Beyond Bayesian Optimization

*module includes Python code review / exercises