Bayesian Optimization: A Review

Fan Bu

Lab meeting discussion, May 5 2023

Reference: Shahriari et al., 2016. "Taking the human out of the loop: A review of Bayesian optimization". Proceedings of the IEEE.

Roadmap

- The problem of global optimization with “hard” objective functions
- Logic and components of “Bayesian Optimization”
- Technical details & practical challenges
- Dicussion

Problem: global optimization with limited evaluation budget

$$x^* = \arg\max_x f(x).$$

where $f(x)$ is assumed continuous, but

- "black-box"
- expensive to evaluate
- doesn't admit gradients
- (dimension of $x \sim O(10)$ not huge [Fra18])

Problem: global optimization with limited evaluation budget

$$x^* = \arg\max_x f(x).$$

where $f(x)$ is assumed continuous, but

- “black-box”
- expensive to evaluate
- doesn’t admit gradients
- (dimension of $x \sim O(10)$ not huge [Fra18])

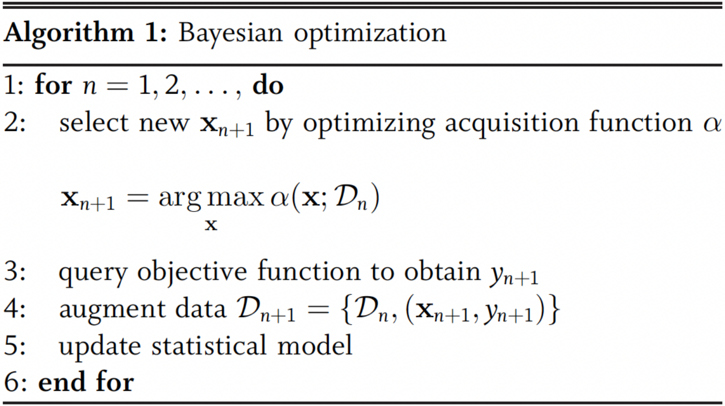
Goal: find global optimizer x with as few evaluations of f as possible

The logic of Bayesian Optimization

Given data $\mathcal{D}_n = \{(x_n, y_n)\}$, evaluate $y_{n+1} = f(x_{n+1})$ at x_{n+1} with highest gain:

1. "approximate" $f(x)$ with a statistical model
   - usually a Gaussian process (GP)
2. find the next point x_{n+1} to maximize an "acquisition function"
   - multiple choices balancing exploitation & exploration

Algorithm sketch



Algorithm 1: Bayesian optimization

```
for $n = 1, 2, \dots,$ do
    select new $\mathbf{x}_{n+1}$ by optimizing acquisition function $\alpha$

        $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$

    query objective function to obtain $y_{n+1}$
    augment data $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (\mathbf{x}_{n+1}, y_{n+1})\}$
    update statistical model
end for
```

Source: Shahriari et al., 2016 [SSW+15].

Statistical model for $f(x)$

Common practice: GP model

$$y \sim N(f(x), \sigma^2_{\text{noise}})$$
$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')).$$

- $\mu(x)$: mean function
- $k(x, x')$: kernel

Statistical model for $f(x)$

Common practice: GP model

$$y \sim N(f(x), \sigma^2_{\text{noise}})$$
$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')).$$

- $\mu(x)$: mean function
- $k(x, x')$: kernel
- **Closed-form predictive distribution** of $f(x_{\text{new}})$, conditioned on $\mathcal{D}_n$:

$$f(x_{\text{new}}) \mid \mathcal{D}_n \sim N(\mu_n(x_{\text{new}}), \sigma^2_n(x_{\text{new}})).$$

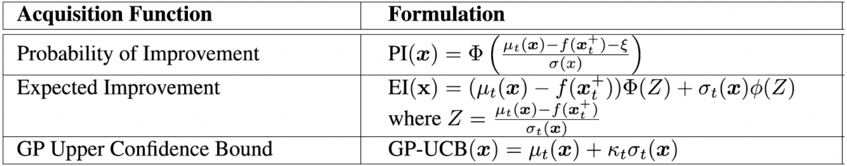
- (See pg.157, (29) & (30) of [SSW$^+$15])

Acquisition functions

In general, choose next x to maximize:

$$a \times \text{Exploitation term} + b \times \text{Explorartion term}.$$

Some common choices:



Acquisition Function	**Formulation**
Probability of Improvement	$\mathrm{PI}(\boldsymbol{x}) = \Phi\left(\frac{\mu_t(\boldsymbol{x}) - f(\boldsymbol{x}_t^+) - \xi}{\sigma(x)}\right)$
Expected Improvement	$\mathrm{EI}(\mathbf{x}) = (\mu_t(\boldsymbol{x}) - f(\boldsymbol{x}_t^+))\Phi(Z) + \sigma_t(\boldsymbol{x})\phi(Z)$ where $Z = \frac{\mu_t(\boldsymbol{x}) - f(\boldsymbol{x}_t^+)}{\sigma_t(\boldsymbol{x})}$
GP Upper Confidence Bound	$\text{GP-UCB}(\boldsymbol{x}) = \mu_t(\boldsymbol{x}) + \kappa_t \sigma_t(\boldsymbol{x})$

Source: Greenhill et al., 2020 [GRG$^+$20].

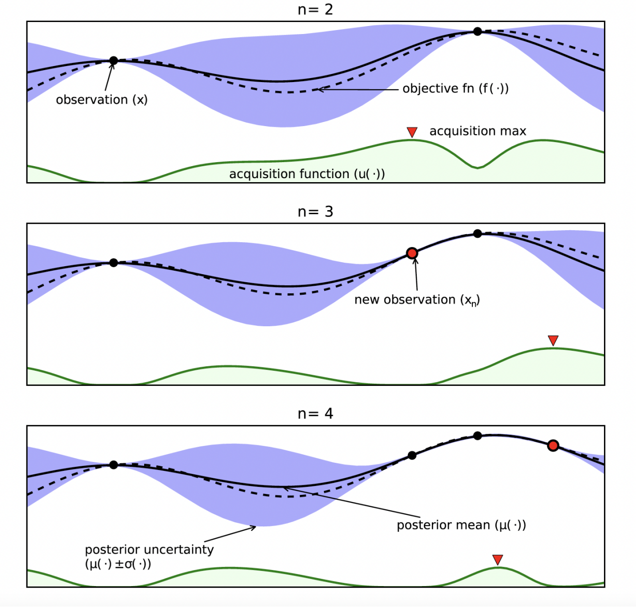


Figure 1: Example of BO in action. Source: [SSW$^+$15].

BO has a lot of applications

- (Hyperparameter) Tuning of large/complex models, e.g.
  - deep neural nets, language models
- Optimization/Simulation of complex dynamical systems, e.g.,
  - systems in cosmology, meteorology, traffic flows
- Online learning / reinforcement learning tasks, e.g.,
  - A/B testing, recommender systems, etc.
  - with connections to "multi-armed bandits"
- Experiment design in engineering (see [GRG$^+$20] for a nice review)

The dirty truth: BO is hard

- GPs are hard
  - choice of kernel k
  - hyperparameters of GP
  - computational burden in inference (matrix inversion)
- Acquisition function can be hard to optimize
  - can be multi-modal and complex
  - computational cost can be high
  - (See Section V. B in [SSW^{+}15] for review.)

GP kernel choice

Usually stationary functions w.r.t. $r = \|x - x'\|_2$ for different levels of smoothness. See [SSW$^+$15] pg. 157, (31-34) for examples.

GP kernel choice

Usually stationary functions w.r.t. $r = \|x - x'\|_2$ for different levels of smoothness. See [SSW$^+$15] pg. 157, (31-34) for examples.

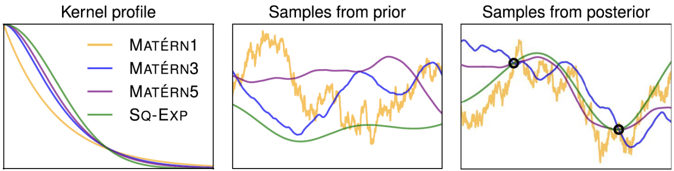


***Fig. 3.** (Left): Visualization of various kernel profiles. The horizontal axis represents the distance $r > 0$. (Middle): Samples from GP priors with the corresponding kernels. (Right): Samples from GP posteriors given two data points (black circles). Note the sharper drop in the Matérn1 kernel leads to rough features in the associated samples, while samples from a GP with the Matérn3 and Matérn5 kernels are increasingly smooth.*

Handling hyperparameters

Hyperparameters: scale parameters in k, initial mean function μ_0, noise variance σ^2_{noise}, etc.

- Optimal: marginalize over hyperparameters
  - analytical solution if conjugate priors exist (and make sense)
  - numerical solution through Monte Carlo simulation (or even MCMC)
- Ad hoc: plug-in with estimates of hyperparameters

Computational burden

Each iteration of GP inference involves

$$\left[K + \sigma^2_{\text{noise}} I_n\right]^{-1},$$

where $K \in \mathbb{R}^{n \times n}$ with $K_{i,j} = k(x_i, x_j)$.

Computational burden

Each iteration of GP inference involves

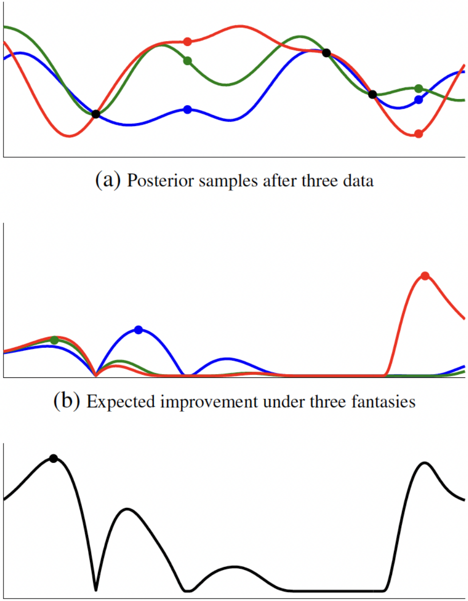
$$\left[K + \sigma^2_{\text{noise}} I_n\right]^{-1},$$

where $K \in \mathbb{R}^{n \times n}$ with $K_{i,j} = k(x_i, x_j)$.

- $O(n^3)$ if exact
- $O(n^2)$ with decomposition (e.g., Cholesky), **but** has to update every time
- $O(nm^2 + m^3)$ with approximation using m pseudopoints (Section III. E. of [SSW+15])
- might further reduce if sparsity enforced (e.g., enforcing CAR-ish structure for conditional independence; similar to INLA [RRS+17])

Parallelization

- Pseudo-parallel: propose J fantasies and get Monte Carlo estimate of α; e.g., with EI [SLA12].
- Parallel: get a set of J evaluation points simultaneously with various acquisition function tuning parameters; e.g, with GP-UCB [HHLB12, Jon01].



(a) Posterior samples after three data

(b) Expected improvement under three fantasies

(c) Expected improvement across fantasies

Example: using 3 pending evaluations as “fantasies” to get “expected” acquisition function (EI in this example)

Discussion

- High-dimensional case?
  - model order reduction techniques to reduce the “effective dimensionality”?
  - e.g., cost-efficient online learning for splines...?

Discussion

- High-dimensional case?
  - model order reduction techniques to reduce the "effective dimensionality"?
  - e.g., cost-efficient online learning for splines...?
- Do we still care about uncertainty quantification?
  - can obtain/approximate marginal posterior of $x^* \mid \mathcal{D}_n$

Discussion

- High-dimensional case?
    - model order reduction techniques to reduce the "effective dimensionality"?
    - e.g., cost-efficient online learning for splines...?
- Do we still care about uncertainty quantification?
    - can obtain/approximate marginal posterior of $x^* \mid \mathcal{D}_n$
- If pure exploration (no need for optimization)?
    - what happens if $\alpha := \sigma_n(x)$?
    - next evaluation solely to reduce uncertainty
    - (look more closely at $\sigma_n(x)$...)
    - similar to mesh refinement in finite-element methods...? [Lo98, JP97]

References I

Peter I Frazier, *A tutorial on bayesian optimization*, arXiv preprint arXiv:1807.02811 (2018).

Stewart Greenhill, Santu Rana, Sunil Gupta, Pratibha Vellanki, and Svetha Venkatesh, *Bayesian optimization for adaptive experimental design: A review*, IEEE access **8** (2020), 13937–13948.

Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown, *Parallel algorithm configuration*, Learning and Intelligent Optimization: 6th International Conference, LION 6, Paris, France, January 16-20, 2012, Revised Selected Papers, Springer, 2012, pp. 55–70.

Donald R Jones, *A taxonomy of global optimization methods based on response surfaces*, Journal of global optimization **21** (2001), 345–383.

References II

- Mark T Jones and Paul E Plassmann, *Adaptive refinement of unstructured finite-element meshes*, Finite Elements in Analysis and Design **25** (1997), no. 1-2, 41–60.

- SH Lo, *3d mesh refinement in compliance with a specified node spacing function*, Computational mechanics **21** (1998), no. 1, 11–19.

- Håvard Rue, Andrea Riebler, Sigrunn H Sørbye, Janine B Illian, Daniel P Simpson, and Finn K Lindgren, *Bayesian computing with inla: a review*, Annual Review of Statistics and Its Application **4** (2017), 395–421.

- Jasper Snoek, Hugo Larochelle, and Ryan P Adams, *Practical bayesian optimization of machine learning algorithms*, Advances in neural information processing systems **25** (2012).

References III

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** (2015), no. 1, 148–175.