Do we need "Harmless" Bayesian Optimization and "First-Order" Bayesian Optimization?

Mohamed Osama Ahmed, Bobak Shahriari, Mark Schmidt

University of British Columbia

December 2016

Bayesian Optimization (BO) vs. Random Search

- Bayesian optimization has a long history and is used in many fields.
- In machine learning, it has been proposed for automatic hyper-parameter tuning.

- Li et al. [2016] questions usefulness of BO for hyper-parameter tuning:
  - 117 hyper-parameter tuning problems.
  - BO typically has small gains over random guesses.
  - BO typically outpeformed by doubling number of random guesses.

- But BO practioners know BO is often much faster than random.

- ??

- This talk:
  - Why BO might/not-might beat random, and how we can improve BO.

Iteration Complexity Framework

- We consider a minimizing a real f over upper/lower bounds $\mathcal{X}$,

$$\operatorname*{argmin}_{x \in \mathcal{X}} f(x).$$

- At each iteration t of the iteration complexity game:
  - Algorithm can pick parameter vector x^t.
  - Algorithm receives function value $f(x^t)$ (noiseless).
- We want to minimize number of iterations t before algoirthm guarantees

$$f(\hat{x}^t) - f^* \leq \epsilon,$$

where $\hat{x}^t$ is algorithm's guess of global optimum f^*, and accuracy $\epsilon > 0$.

Iteration Complexity vs. Error after Fixed Time

- Iteration complexitystuides how big t need to be to guarantee ϵ accuracy.
- Example:
  - For high-dimensional convex functions, we need $O(1/\epsilon^2)$ iterations.
- Can equivalently state results in terms of error ϵ after fixed iterations t.
  - If we need $t = O(1/\epsilon^2)$ iterations, then error after t steps is $\epsilon = O(1/\sqrt{t})$.

Difficulty of Real-Valued Optimization

- We're minimizing a real f over bounds $\mathcal{X}$,

$$\operatorname*{argmin}_{x \in \mathcal{X}} f(x).$$

- How many iterations t before any algorithm could guarantee $f(\hat{x}^t) - f^* \leq \epsilon$?

- Impossible!

- Given any algorithm, we can construct an f where error $> \epsilon$ forever.
  - Make $f(x) = 0$ everywhere except 1 real number x^* where $f(x^*) = -\epsilon - 2^{\text{whatever}}$.
    (The x^* is algorithm-specific.)

- To say anything about runtime we need assumptions on f.

Difficulty of Lipschitz-Continuous Optimization

- One of the simplest assumptions is Lipschitz-continuity (others are possible):

$$|f(x) - f(y)| \leq L\|x - y\|,$$

  for all x and y and some $L < \infty$.
- Function can't change arbitrarily fast as you change x.

- Under this assumption, any algorithm requires at least $\Omega(1/\epsilon^d)$ iterations.
- An optimal $O(1/\epsilon^d)$ worst-case rate is achieved by a grad-based search method.
  - See Chapter 1 of Nesterov's book.
- An optimal $O(1/\epsilon^d)$ worst-case rate is achieved by random guesses.
  - Probability that a random guess is ϵ-optimal is $\Omega(\epsilon^d)$.

- So random guessing is optimal.

Bayesian Optimization for Lipschitz-Continuous Optimization

- So we have that convergence rate of random guesses is $O(1/\epsilon^d)$.
- Under certain assumptions, BO convergence rate is $\tilde{O}(1/\epsilon^{\nu/d})$ [Bull, 2011].
  - Parameter ν is a masure of "smoothness" of f.
- If $\nu > 1$, BO can be exponentially faster than random guessing.
  - Supports empirical experiments where BO crushes random.
- If $\nu < 1$, BO can be slower than random guessing.

Harmless Bayesian Optimization (HBO)

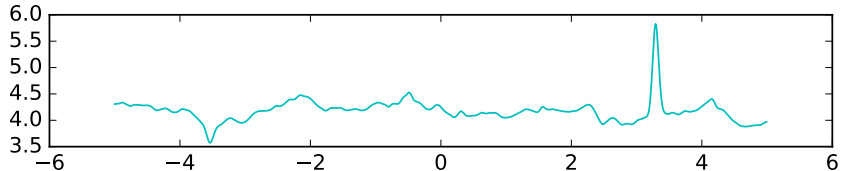
- We typically don't know η, so we don't know if BO will beat random.
- Motivates harmless Bayesian optimization (HBO).
  - "An HBO algorithm requires at most $O(1/\epsilon^d)$ iterations to achieve accuracy ϵ on a Lipschitz-continuous function."
- HBO algorithms guaranteed to perform within constant factor of random.

A Simple Harmless Method

- A simple way to make an existing BO method harmless:
  - On odd iterations, pick a random x^t.
  - On even iterations, apply the BO method.
- Achieves a faster rate of $\tilde{O}(1/\epsilon^{\min\{d,d/\nu\}})$ under Bull's assumptions.
- Similar to ϵ-greedy algorithms for exploration vs. exploitation.
  - We could use random iterations for any fixed porition of the time.
- There are probably better methods that:
  - Share information between random/BO iterations, and/or locally exploit smoothness.

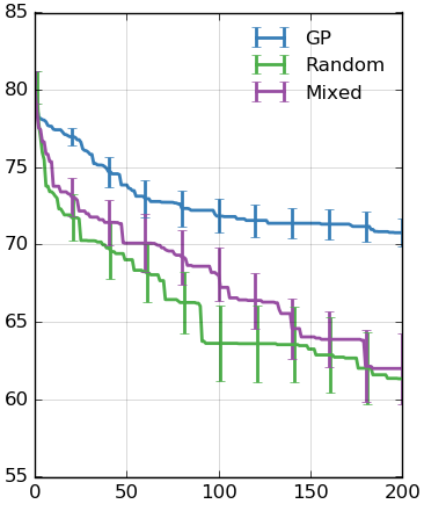
Experiment with Harmless Bayesian Optimization

- We applied a kernel smoother to samples from a 10-dimensional t-distribution.



- Yields a differentiable function where BO converges slowly.

Experiment with Harmless Bayesian Optimization



Beating Random: Exploiting Structure

- HBO ensures we aren't beaten by random, but this is a bar for "success".
- How can we do go significantly faster than random?

- Usually, we aren't really optimizing a black box:
  - Problems have structure, and we can exploit this to give faster methods.

- Structure in convex optimization giving faster algorithms:
  - Convexity, smoothness, projections, proximal operators, linear oracles, analytic optimization over subsets, finite-sum problems, strong-convexity, self-concordance.
- Structure in non-convex optimization giving faster algorithms:
  - Polyak-Lojasiewicz condition, label switching arguments, instability of non-global critical points.

First-Order Bayesian Optimization (FOBO)

- We can do significantly better than random using structure in f.
- We focus on one of the simplest structures: f is differentiable.

- First-Order Bayesian optimization: Bayesian optimization with derivatives.

- Using derivatives in GPs/BOs is not a new idea.

[Morris et al., 1993, Solak et al., 2003, Rasmussen & Williams, 2006, Lizotte, 2008, Osborne, 2010, ??????]

- But it's under-utilized:
  - Many problems where we apply BO are differentiable:
    - Gradient-based hyper-parameter learning [Bengio, 2000, Maclaurin et al., 2015].
  - Cost of getting gradient is same order as getting function value.
  - For sufficiently smooth functions, convergence rate should be faster (conjecture).

First-Order Bayesian Optimization (FOBO)

- Key idea: assume funtion value and all first derivatives are jointly Gaussian.
- If covariance kernel is twice-differentiable, extra covariance matrix elements are

$$\begin{aligned}\operatorname{cov}(f(x^i), \partial_p f(x^j)) &= \partial_p k(x^i, x^j),\\ \operatorname{cov}(\partial_p f(x^i), \partial_q f(x^j)) &= \partial_p \partial_q k(x^i, x^j),\end{aligned}$$

where $\partial_p f$ is the directional derivative of f in the direction p.

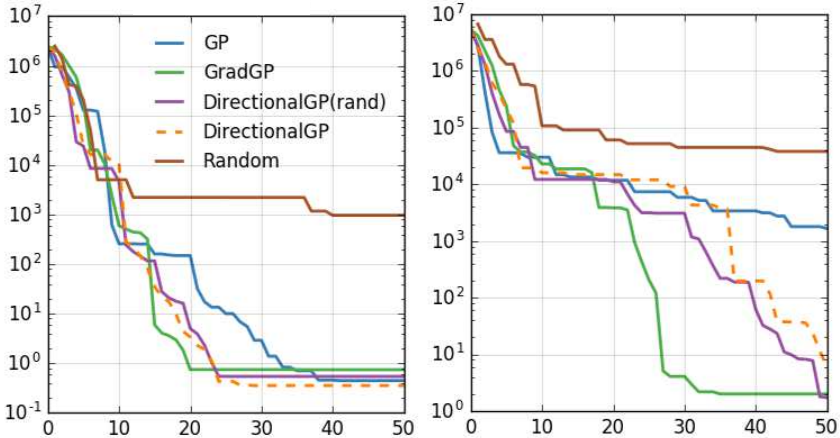
FOBO with Directional Derivatives

- FOBO increases space from $O(t^2)$ to $O(t^2 d)$.
- FOBO increases time from $O(t^3)$ to $O(t^3 d^3)$.

- If this is too large, we can focus on modeling directional derivatives.
  - We considered using gradient direction or a random direction.
  - Has same time/space complexity as function-only BO.
  - Can be computed exactly using forward-mode automatic differentiation.
    - Don't need gradient code or doesn't increase cost.

Experiment with First-Order Bayesian Optimization

Experiments with 2D and 3D Rosenbrock function:



Summary

- Effectiveness of continuous optimizers depends on assumptions.
- For fairly-general functions, random is optimal.

- We proposed harmless Bayesian optimization (HBO):
  - Similar to random for "hard" functions.
  - Can be much faster for "easy" functions.

- If we want to beat random, we need extra structure in the problem.

- We explored first-order Bayesian optimization (FOBO):
  - Incorporates derivatives to converge faster.
  - Can use directional derivatives to reduce cost.