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

Mohamed Osama Ahmed (UBC), Bobak Shahriari (UBC), Mark Schmidt (UBC)

Motivation and Overview of Contribution

- Recent empirical study on hyper-parameter tuning [Li et al., 2016]:
  - Bayesian optimization outperformed by random run for twice as long.
- So should we use Bayesian optimization?
  - Random is an optimal solver in the worst-case ("hard" problems).
  - But on certain problems, Bayesian optimization is exponentially faster ("easy" problems).
- We propose two research directions to improve Bayesian optimization:
  - For "hard" problems; armless Bayesian optimization methods that do no worse than random.
  - For "easy" problems: first-order Bayesian optimization (FOBO) uses gradients to solve even faster.
    - And possibly using directional derivatives to reduce the cost.

Problem formulation

- We consider the problem of minimizing a real-valued f with lower and upper bounds,

$$\arg\min_{x \in \mathcal{X}} f(x). \tag{1}$$

- At iteration t, the algorithm chooses an x^t and receives $f(x^t)$.
- Goal: minimize number of iterations t before we have $f(\hat{x}^t) - f^* \leq \epsilon$
  - Equivalent to problem of minimizing sub-optimality on iteration t.
  - Impossible in any finite number of iterations without assumptions on f.
- A weak assumption is that f is Lipschitz-continuous:
  - In worst case, any algorithms requires $\Omega(1/\epsilon^d)$.
  - Random search requires $O(1/\epsilon^d)$ so it is optimal.
- For ν-smooth functions Bayesian optimization requires $O(1/\epsilon^{d/\nu})$:
  - Slower than random search when $\nu < 1$.
  - Faster than random search when $\nu > 1$.

Harmless Bayesian Optimization

- For "black-box" optimization, we don't want BO to be worse than random.
- A "harmless" BO algorithm is a BO method that requires at most $O(1/\epsilon^d)$ iterations.
- Harmless BO methods perform as well as random on "hard" problems.
- Achievable with a simple alternating algorithm:
  - Alternate between BO and random iterations to achieve a rate of $O(1/\epsilon^{\min\{d, d/\nu\}})$.

First Order Bayesian Optimization

- For "easy" functions, we should be able to improve BO with derivatives.
- If the kernel k is twice differentiable, GP directional derivatives are generated by a GP:

$$\mathrm{cov}(f(x^i), \partial_p f(x^j)) = \partial_p k(x^i, x^j), \text{ and} \tag{2}$$
$$\mathrm{cov}(\partial_p f(x^i), \partial_q f(x^j)) = \partial_p \partial_q k(x^i, x^j), \tag{3}$$

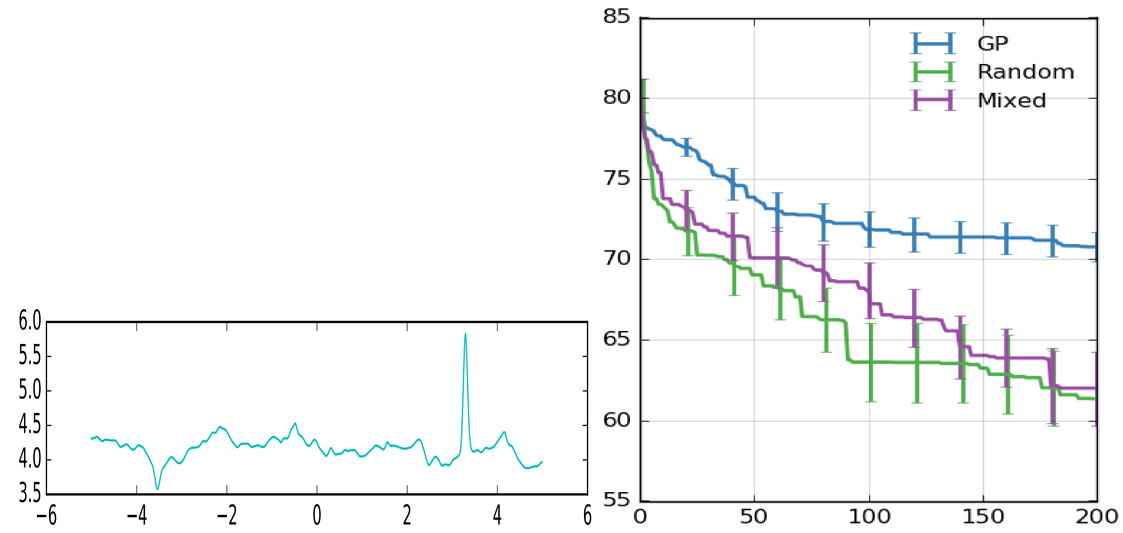
where $\partial_p f$ denotes the partial derivative of f with respect to direction p.

Directional Derivatives

- The memory and time requirement increase if we use full gradients:
  - Memory is increased from $O(t^2)$ to $O(t^2 d^2)$.
  - Cost of the GP is increased from $O(t^3)$ to $O(t^3 d^3)$.
- We can avoid this using directional derivatives $\partial_p f(x^t)$ for directions p:
  - If we have the gradient, we could use $p = \nabla f(x^t)$.
  - If not, we could set p to a random direction.
- Provides derivative information but only increases time/memory by constant.
  - Always cheap for analytic functions.
- We conjecture that gradient information improves the convergence rate.

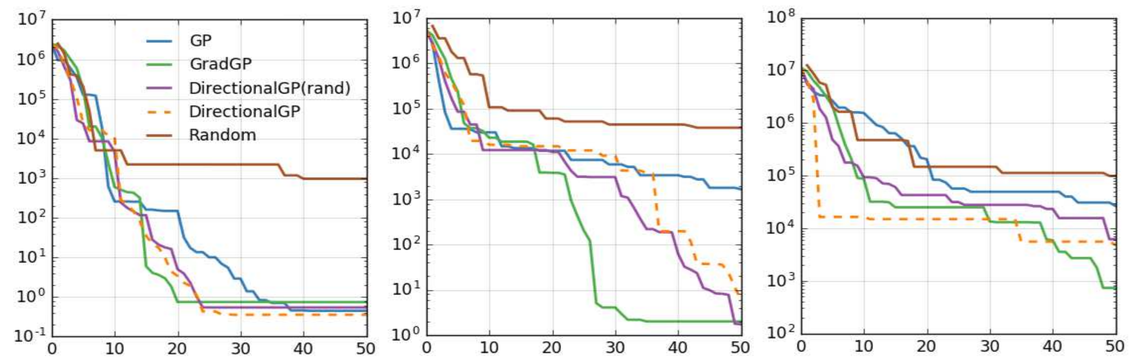
Experiment Results on Harmless BO

- To test harmless BO on a "hard" function, we applied kernel smoother to 10-dimensional samples from a student t-distribution (differentiable but not sufficiently smooth for BO to be effective).

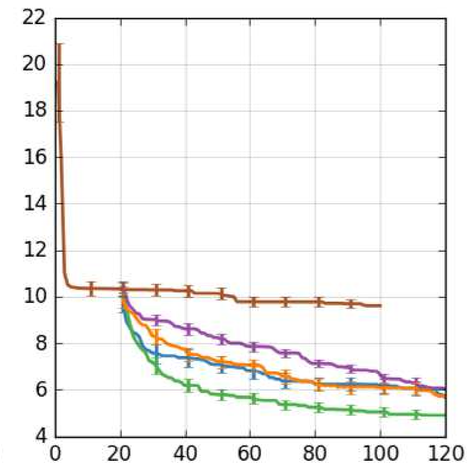


Experiment Results on FOBO

- To test FOBO on an "easy" function, we used the Rosenbrock function $f(x_1 \cdots x_d) = \sum_{i=1}^{d-1}(100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2)$ for different dimensions d.



- We also explored neural network training:



Extensions

- Is there a better way to combine random and BO?
- Developing black-box Harmless FOBO methods.
- Can we exploit *local* smoothness?