Bayesian Optimization of Composite Functions

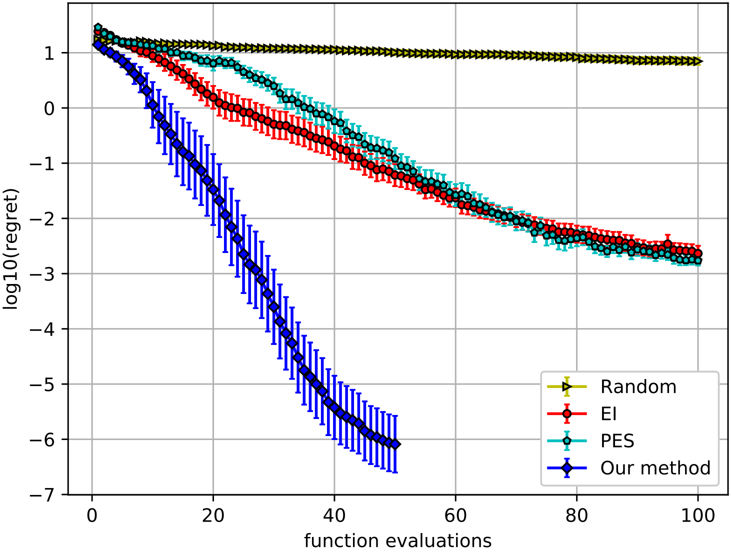
To appear at ICML 2019

Raúl Astudillo

Cornell University

Joint work with Peter I. Frazier

2nd Uber Science Symposium, May 3, 2019


1
0
−1
−2
−3
−4
−5
−6
−7
log10(regret)
0
20
40
60
80
100
function evaluations
Random
EI
PES
Our method

How it works: an illustration

Suppose

- x is a parameter of a simulator
- $h(x)$ is simulator's prediction under x
- y is our observed data

We want to solve

$$\min_x (h(x) - y)^2.$$

Standard BO

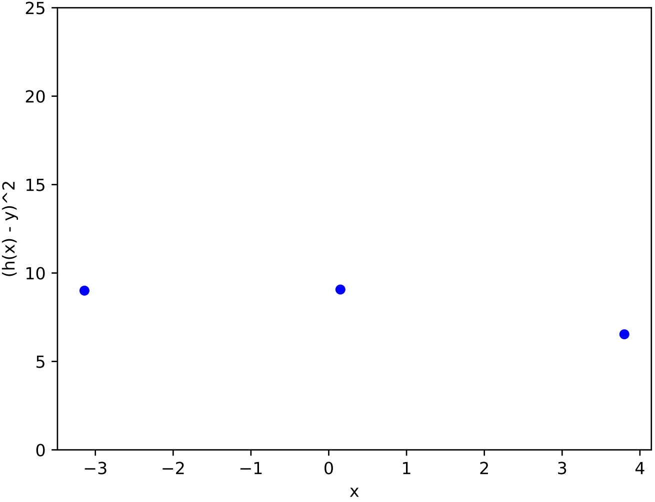


Figure: Evaluations of $(h(x) - y)^2$

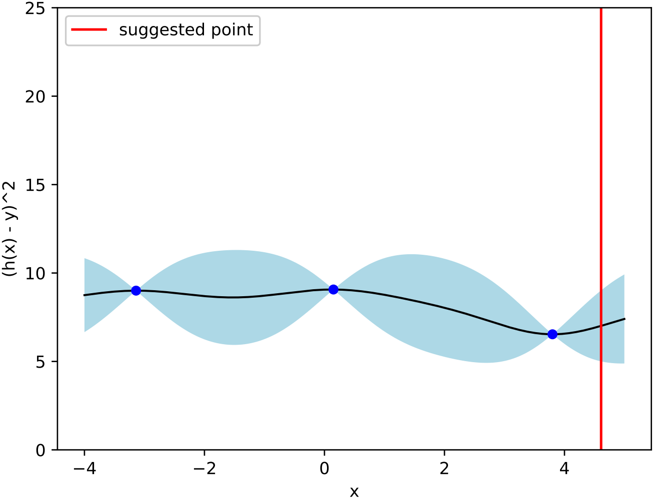


Figure: GP posterior on $(h(x) - y)^2$

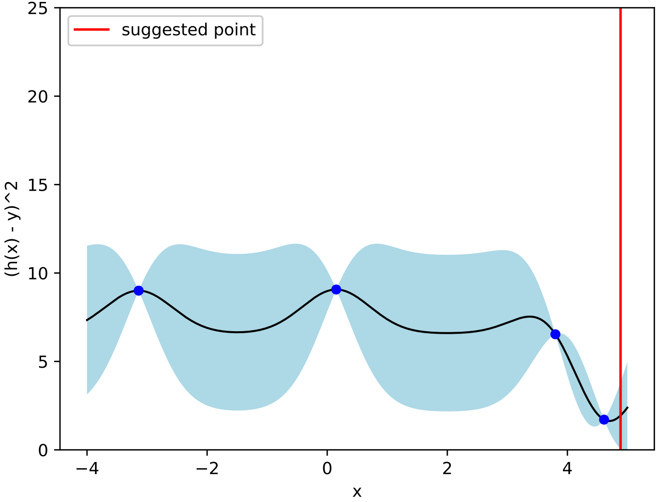
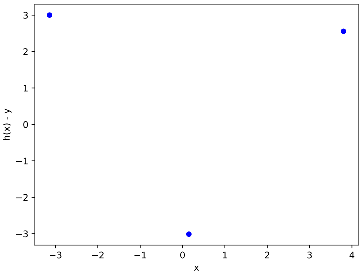
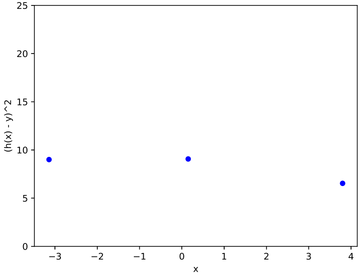


Figure: GP posterior on $(h(x) - y)^2$

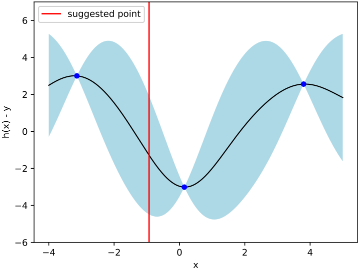
Our approach



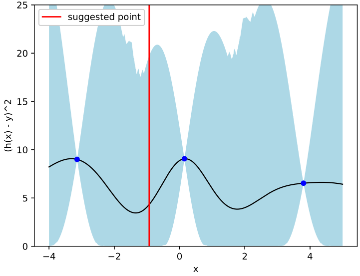
(a) Evaluations of $h(x) - y$



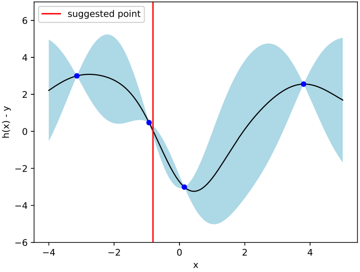
(b) Evaluations of $(h(x) - y)^2$



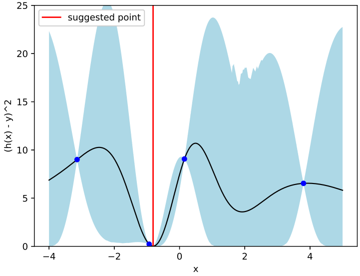
(a) GP posterior on $h(x) - y$



(b) Implied posterior on $(h(x) - y)^2$



(a) GP posterior on $h(x) - y$



(b) Implied posterior on $(h(x) - y)^2$

Our problem

We consider problems of the form

$$\max_{x \in \mathcal{X}} f(x),$$

where

$$f(x) = g(h(x))$$

and

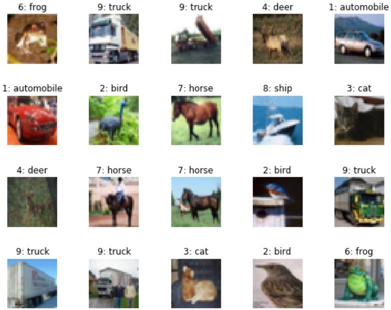
- $h : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}^m$ is a time-consuming-to-evaluate black-box
- $g : \mathbb{R}^m \to \mathbb{R}$ (and its gradient) are known in closed form and fast-to-evaluate

Composite functions arise naturally in practice

Example: Hyperparameter tuning of classification algorithms

$$g(h(x)) = -\sum_{j=1}^{m} h_j(x),$$

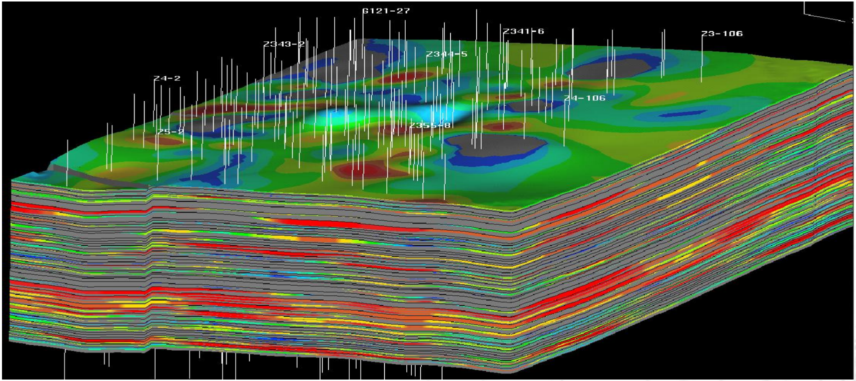
where h_j is the classification error on the j-th class.



Example: Calibration of an oil reservoir simulator

$$g(h(x)) = -\sum_{j=1}^{m} (h_j(x) - y_j)^2,$$

where y is a vector of observed data.



Example: Optimization of posteriors with expensive likelihoods

$$\log p(x \mid y) = \log \underbrace{L(y \mid x)}_{\text{likelihood}} + \log \underbrace{\pi(x)}_{\text{prior}}.$$

Very often, $L(y \mid x) \propto g(y \mid h(x))$, where g is known in closed form and $h(x)$ is a vector of parameters governing properties of the data's distribution.

E.g., for a Gaussian likelihood,

$$g(y \mid h(x)) \propto -\|h(x) - y\|_2^2.$$

Related work

BO for sums of functions:

- Swersky, K., Snoek, J. and Adams, R. P. Multi-task bayesian optimization. In *Advances in neural information processing systems* (pp. 2004-2012). 2013.
- Toscano-Palmerin, S. and Frazier, P.I. Bayesian optimization with expensive integrands. arXiv preprint arXiv:1803.08661. 2018.
- Several others.

Constrained BO:

- Gardner, J.R., Kusner, M.J., Xu, Z.E., Weinberger, K.Q. and Cunningham, J.P. Bayesian Optimization with Inequality Constraints. *In International Conference on Machine Learning* (pp. 937-945). 2014.
- Several others.

Our contribution

1. A **statistical approach** for modeling f that greatly improves over the standard BO approach

2. An efficient **way to optimize EI** under this new statistical model

Our approach

- Model h using a multi-output Gaussian process instead of f directly

- This implies a (non-Gaussian) posterior on $f(x) = g(h(x))$

- To decide where to sample next: compute and optimize the expected improvement acquisition function under this new posterior

Background: Expected Improvement (EI)

The most widely used acquisition function in standard BO is:

$$\mathrm{EI}_n(x) = \mathbb{E}_n\left[\{f(x) - f_n^*\}^+\right],$$

where

- f_n^* is the best observed value so far
- $\mathbb{E}_n$ is the conditional expectation under the posterior after n evaluations

Background: Expected Improvement (EI)

The most widely used acquisition function in standard Bayesian optimization is:

$$\mathrm{EI}_n(x) = \mathbb{E}_n\left[\{f(x) - f_n^*\}^+\right],$$

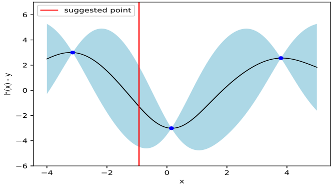
When $f(x)$ is Gaussian, EI and its derivative have a closed form which make it easy to optimize.

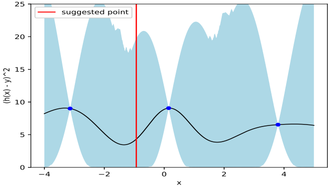
Expected Improvement for Composite Functions

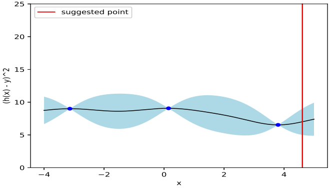
Our acquisition function is Expected Improvement for Composite Functions (EI-CF):

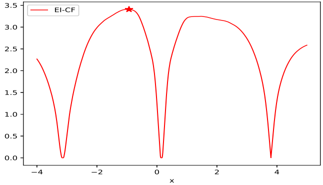
$$\text{EI-CF}_n(x) = \mathbb{E}_n\left[\{g(h(x)) - f_n^*\}^+\right],$$

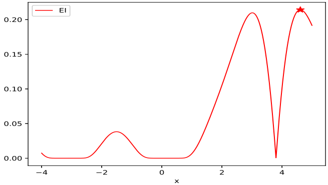
where h is a GP, making $h(x)$ Gaussian.


suggested point
h(x) - y
x


suggested point
(h(x) - y)^2
x


suggested point
(h(x) - y)^2
x


EI-CF
x


EI
x

Challenge: maximizing EI-CF is hard

Expected Improvement for Composite Functions (EI-CF):

$$\text{EI-CF}_n(x) = \mathbb{E}_n\left[\{g(h(x)) - f_n^*\}^+\right],$$

where h is a GP, making $h(x)$ Gaussian.

Challenge:

- When h is a GP and g is nonlinear, $f(x) = g(h(x))$ is **not Gaussian**
- EI no longer has a closed form, making it hard to optimize

Calculating EI-CF

To estimate $\mathrm{EI\text{-}CF}_n(x)$, repeat the following N times:

1. Sample $h(x)$ from the Gaussian process posterior
2. Calculate the improvement $\{g(h(x)) - f_n^*\}^+$

Then average the results.

Challenge: maximizing EI-CF is hard

- **Naive optimization method:** Maximize EI-CF directly, e.g., using a genetic algorithm
- **Problem:** this will be really slow because we don't have gradients and the evaluations are noisy

A better way to maximize EI-CF

1. Reparametrization trick
2. Evaluate using Monte Carlo
3. Optimize using a novel gradient estimator

Reparametrization trick

$$h(x) \stackrel{d}{=} \mu_n(x) + C_n(x)Z,$$

where

- μ_n and K_n are the posterior mean and covariance functions of h
- $C_n(x)$ is the Cholesky factor of $K_n(x,x)$
- Z is a m-variate standard normal random vector

Reparametrization trick

$$h(x) \overset{d}{=} \mu_n(x) + C_n(x)Z,$$

where

- μ_n and K_n are the posterior mean and covariance functions of h
- $C_n(x)$ is the Cholesky factor of $K_n(x,x)$
- Z is a m-variate standard normal random vector

Thus,

$$\text{EI-CF}_n(x) = \mathbb{E}\left[\{g(\mu_n(x) + C_n(z)\textcolor{red}{Z}) - f_n^*\}^+\right].$$

Evaluate using Monte Carlo

$$\text{EI-CF}_n(x) \approx \frac{1}{L}\sum_{\ell=1}^{L}\{g(\mu_n(x) + C_n(x)Z^{(\ell)}) - f_n^*\}^+,$$

where $Z^{(1)}, \ldots, Z^{(L)} \sim \mathcal{N}(0, I_m)$.

Gradient of EI-CF

Lemma.

Under mild regularity conditions, $\mathrm{EI\text{-}CF}_n$ is differentiable almost everywhere and its gradient, when it exists, is given by

$$\nabla \mathrm{EI\text{-}CF}_n(x) = \mathbb{E}_n\left[\gamma_n(x, Z)\right],$$

where

$$\gamma_n(x, Z) = \begin{cases} 0, & \text{if } g(\mu_n(x) + C_n(x)Z) \leq f_n^*. \\ \nabla g(\mu_n(x) + C_n(x)Z), & \text{otherwise.} \end{cases}$$

Our improved method for maximizing EI-CF

To get a stochastic gradient, i.e., an unbiased estimate of $\nabla_x \text{EI-CF}_n(x)$:

1. Sample a standard normal random vector Z
2. Return $\gamma_n(x, Z)$

Our improved method for maximizing EI-CF

To get a stochastic gradient, i.e., an unbiased estimate of $\nabla_x \mathrm{EI\text{-}CF}_n(x)$:

1. Sample a standard normal random vector Z
2. Return $\gamma_n(x, Z)$

We use these stochastic gradients within multi-start stochastic gradient ascent to efficiently maximize $\mathrm{EI\text{-}CF}_n$.

Computational complexity of posterior inference

When outputs of h are modeled independently, the complexity of exact posterior inference is $\mathcal{O}(mn^2)$ (with a precomputation of complexity $\mathcal{O}(mn^3)$).

Recent advances on fast approximate GP prediction allow a $\mathcal{O}(m)$ computational complexity.

Asymptotic consistency

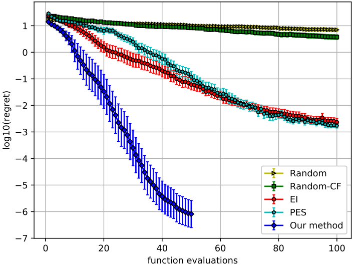
Theorem.

If g is continuous and under additional suitable regularity conditions, EI-CF is asymptotically consistent, i.e., it finds the true global optimum as the number of evaluations goes to infinity.

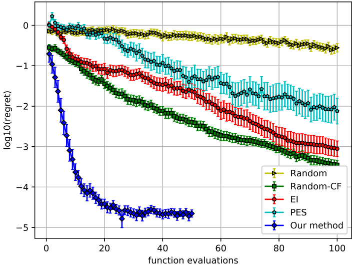
Numerical experiments

GP-generated test problems

Problem	$\mathcal{X}$	g	m
a	$[0,1]^4$	$g(h(x)) = -\sum_{j=1}^{5}(h_j(x) - y_j^*)^2$	5
b	$[0,1]^3$	$g(h(x)) = -\sum_{j=1}^{4}\exp(h_j(x))$	4



(a)



(b)

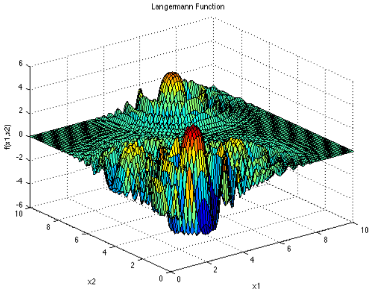
Langermann test problem

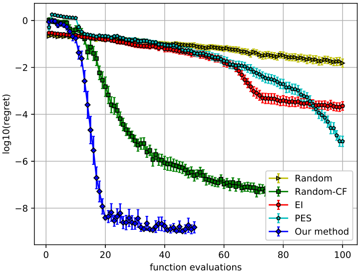
$f(x) = g(h(x))$ where

$$h_j(x) = \sum_{i=1}^{d} (x_i - A_{ij}), \ j = 1, \ldots 5,$$

and

$$g(h(x)) = -\sum_{j=1}^{5} c_j \exp(-h_j(x)/\pi) \cos(\pi h_j(x)).$$





5d Rosenbrock test problem

$$f(x) = -\sum_{j=1}^{d-1} 100(x_{j+1} - x_j^2)^2 + (x_j - 1)^2$$

Adapted to our framework by taking $d = 5$ and

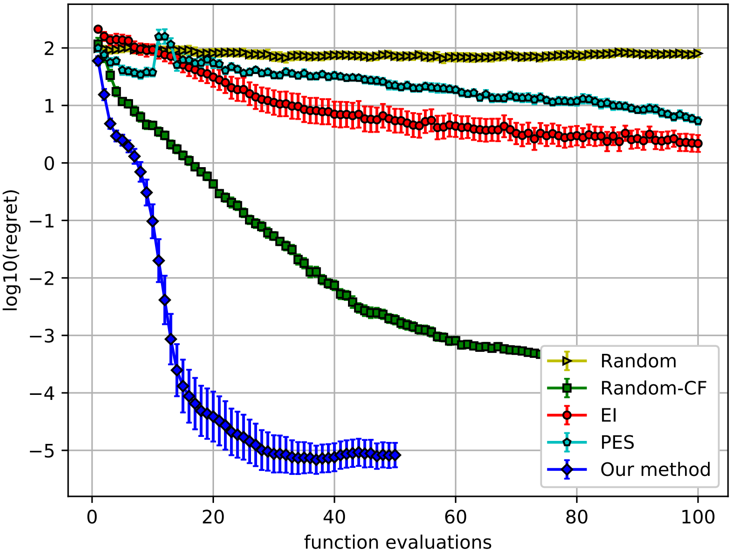
$$h_j(x) = x_{j+1} - x_j^2,\ j = 1, \ldots, 4,$$

$$h_{j+4}(x) = x_j - 1,\ j = 1, \ldots, 4,$$

and

$$g(h(x)) = -\sum_{j=1}^{4} 100 h_j(x)^2 + h_{j+4}(x)^2.$$

5d Rosenbrock test problem

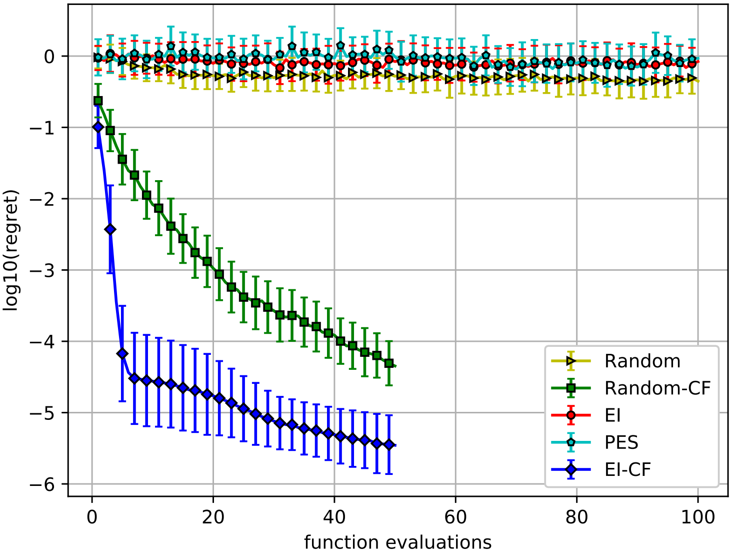


Environmental model test problem

- Models a chemical accident that has caused a pollutant to spill at two locations
- Given 12 measurements at different geospatial locations, invert the 4 parameters of this simulator
- We solve

$$\min_{x \in \mathcal{X}} \sum_{j=1}^{12} \left(s(\theta_j; x^*) - s(\theta_j; x)\right)^2$$

Environmental model test problem



Conclusion and future work

- Exploiting composite objective functions can substantially improve the performance of BO
- Develop efficient implementatios of other acquisitions in this setting
- Some of them would allow noisy and decoupled evaluations

Check out our paper

Astudillo, R. and P. I. Frazier. Bayesian Optimization of Composite Functions. To appear in *Proceedings of the International Conference on Machine Learning*, 2019.

Code

- Check out our code: `https://github.com/RaulAstudillo06/BOCF`
- Coming to Cornell-MOE: `https://github.com/wujian16/Cornell-MOE`
- (Cornell-MOE is now easier to install for python 2 or 3 via `https://anaconda.org/frazierlab`)

Thanks!