Bayesian Optimization

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

April 17, 2018

Optimizing "Black-box" Functions

- Consider hyperparameter tuning for a machine learning model
- Goal: Find hyperparameters that minimize the test error
- Can think of this as minimizing the "test error function" w.r.t. hyperparams
  - But we don't know what the function is. Only have a black-box access to it
  - What it means: We can only evaluate the function for any given setting of hyperparam values on some held-out validation dataset
- Each evaluation is typically expensive since it requires
  - Training the model for a given setting of hyperparameters
  - Computing the test error on the validation set
- Would like to quickly find the optimal hyperparams (without using, say, an expensive grid-search)?
- Basically an optimization problem where we only have a black-box access to the function
- Bayesian Optimization enables us to solve such problems efficiently

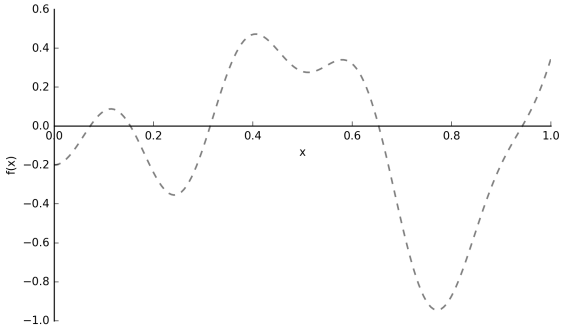
Bayesian Optimization: Many Applications

- Hyperparameter tuning in ML models

- Design of expensive experiments (e.g., drug design, robotic control, etc.)

Bayesian Optimization: The Basic Formulation

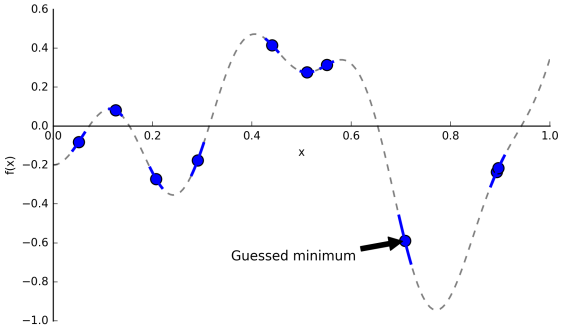
- Consider finding the optima x_* (say minima) of a function $f(x)$



- Caveat: We don't know the form of the function; can't get its gradient,Hessian, etc
- Suppose we can only query the function's values at certain points (i.e., only black-box access)

Bayesian Optimization

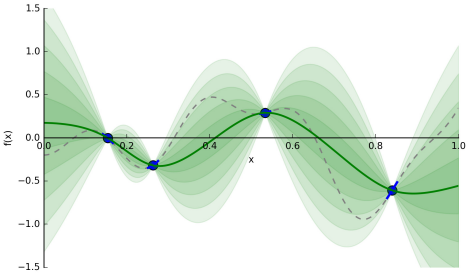
- We would like to locate the minima using the queries we made



- We would like to do so while making as few queries as possible
- Reason: Each query may be expensive cost-wise (where cost may be e.g., time, money, or both)

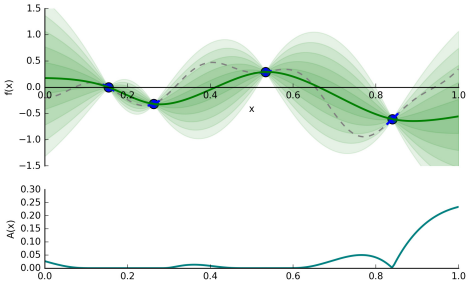
Bayesian Optimization

- Suppose we are allowed to make the queries sequentially
- Queries so far can help us guess what the function looks like (by solving a regression problem)



- Above: dotted = true $f(x)$, solid green = current estimate of $f(x)$, shaded = uncertainty in $f(x)$
- BO use past queries and the function's estimate+uncertainty to decide where to query next
- The next query point is chosen using an acquisition function $A(x)$

Bayesian Optimization



- So basically, Bayesian Optimization requires two ingredients
  - A regression model to learn the function given the current set of query-value pairs $\{x_n, f(x_n)\}_{n=1}^N$
  - An acquisition function $A(x)$ that tells us the utility of any future point x
- Note: Function evaluations given to us may be noisy, i.e., $f(x) + \epsilon$
- Important: The regression model must also have estimate of function's uncertainty, e.g.,
  - Gaussian Process, Bayesian Neural Network, or any nonlinear regression model with uncertainty

Some Acquisition Functions: Probability of Improvement (PI)

- Given the set of previous query points $\mathbf{X}$ and corresponding function values $\boldsymbol{f}$, suppose

$$f' = \min \ \boldsymbol{f}$$

- Suppose f denotes the function's value at some new point x
- We will have an improvement if $f \leq f'$
- Note that we actually have a posterior predictive $p(f|x) = \mathcal{N}(f|\mu(x), \sigma^2(x))$
- We can therefore define the probability of improvement based acquisition function

$$A_{PI}(x) = p(f \leq f') = \int_{-\infty}^{f'} \mathcal{N}(f|\mu(x), \sigma^2(x))df = \Phi\left(\frac{f' - \mu(x)}{\sigma(x)}\right)$$

- Point with the highest probability of improvement is selected as the next query point
- Note that BO involves optimizating the acquisition function to find the next query point x (this optimization to be cheaper than the original problem)

Some Acquisition Functions: Expected Improvement (EI)

- PI doesn't take into account the amount of improvement
- Expected Improvement (EI) takes this into account and is defined as

$$\begin{aligned} A_{EI}(x) = \mathbb{E}[f' - f] &= \int_{-\infty}^{f'} (f' - f)\mathcal{N}(f|\mu(x), \sigma^2(x))df \\ &= (f' - \mu(x))\Phi\left(\frac{f' - \mu(x)}{\sigma(x)}\right) + \sigma(x)\mathcal{N}\left(\frac{f' - \mu(x)}{\sigma(x)}; 0, 1\right) \end{aligned}$$

- Point with the highest expected improvement is selected as the next query point
- Trade-off b/w exploitation and exploration
  - Prefer points with low predictive mean (exploitation) and large predictive variance (exploration)

Some Acquisition Functions: Lower Confidence Bound (LCB)

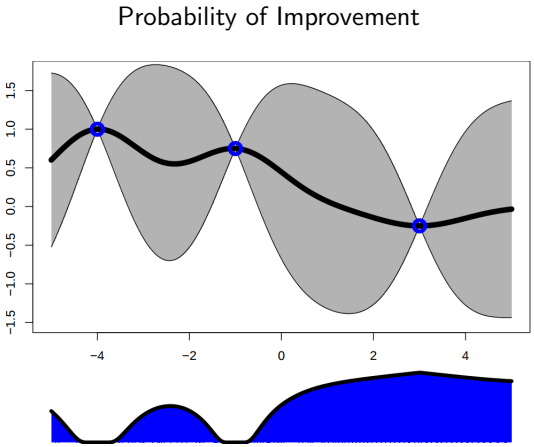
- Another acquisition function that takes into account exploitation vs exploration
- Used when the regression model is a Gaussian Process (GP)
- Assuming the posterior predictive for a new point x to be $\mathcal{N}(\mu(x), \sigma^2(x))$, the UCB is

$$A_{LCB}(x) = -(\mu(x) - \kappa\sigma(x))$$

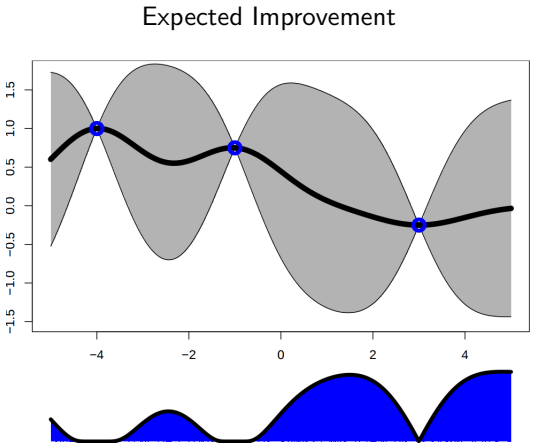
- κ is a parameter to trade-off b/w mean and variance
- Point with the largest LCB is selected as the next query point
- Strong theoretical results; under certain conditions, the iterative application of this acquisition function will converge to the true global optima of f (Srinivas et al. 2010)
- Note: When solving maximization problems, the term "Upper Confidence Bound" (UCB) is used

$$A_{UCB}(x) = \mu(x) - \kappa\sigma(x)$$

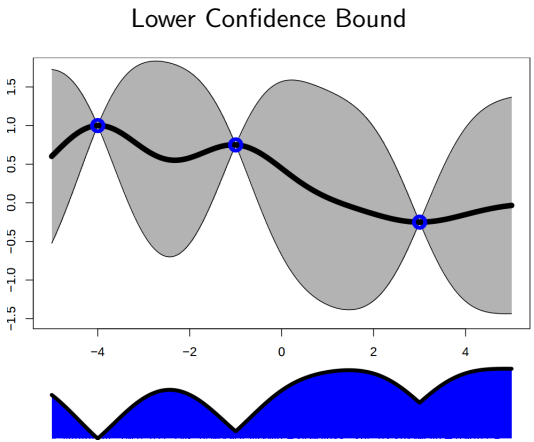
Acquisition Functions: An Illustration



Acquisition Functions: An Illustration



Acquisition Functions: An Illustration



Bayesian Optimization: Some Challenges/Open Problems

- Learning the regression model for the function
  - GPs can be expensive as N grows
    - Bayesian neural networks can be an more efficient alternative to GPs (Snoek et al, 2015)
  - Hyperparams of the regression model itself (e.g., GP cov. function, Bayesian NN hyperparam)
- High-dimensional Bayesian Optimization (optimizing functions of many variables)
  - Number of function evaluations required would be quite large in high dimensions
  - Lot of recent work on this (e.g., based on dimensionality reduction)
- Multitask Bayesian Optimization (optimizing several related functions)
  - Can leverage ideas from multitask learning

Further Resources

- Some survey papers:
  - A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning (Brochu *et al.*, 2010)
  - Taking the Human Out of the Loop: A Review of Bayesian Optimization (Shahriari *et al.*, 2015)
- Some open source software libraries

SOFTWARE	REGR. MODEL	ACQ. FUNCTION
SPEARMINT	GAUSSIAN PROCESS	EXP. IMPROV
MOE	GAUSSIAN PROCESS	EXP. IMPROV
HYPEROPT	TREE PARZEN EST.	EXP. IMPROV
SMAC	RANDOM FOREST	EXP. IMPROV

- Another one: SIGOPT (`https://sigopt.com/research`)