Bayesian Optimization

Kangcheng Hou
kangchenghou@gmail.com

July 15, 2018

Agenda

1. Introduce some attractive examples of Bayesian Optimization.
2. Basic procedure of bayesian optimization without the details of gaussian process or acquisition function.
3. Introduction of gaussian process.
4. Introduction of acquisition function.

Why Bayesian Optimization?

Bayesian optimization applies when

- the evaluation of the function that we are trying to optimize is costly.
- function is very hard to optimize (hard to compute the gradients)
- it looks appropriate to model the function using a gaussian process.

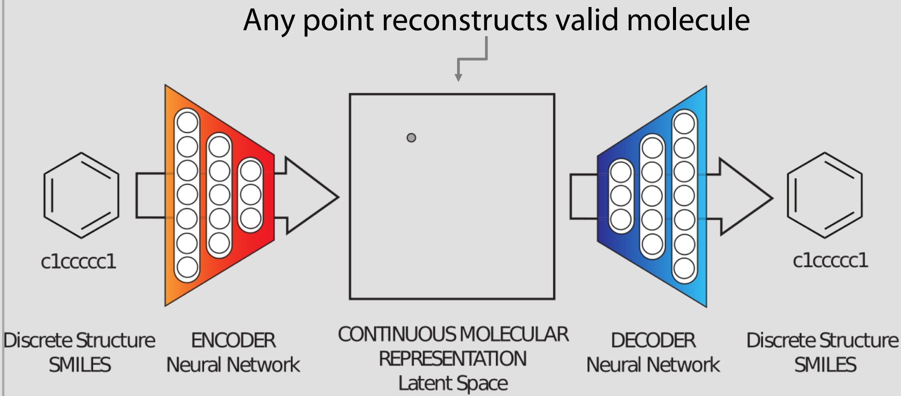
Attractive Examples

Some good and successful examples and scenerios where it is great to try bayesopt.

- hyper-parameter tuning (to evaluate, we need to train the model)
- Design wet-lab experiments saving time and money. (this is much like the hyper-parameter tuning. Only after that we have done the experiments can we see the final results. Using BayesOpt might help you to find better configuration for the experiments faster with proper prior.
- usually finds better optima than when tuned by hand
- honest comparison with other methods for research.
- the input parameters of the function is formed with discrete and continuous variables.

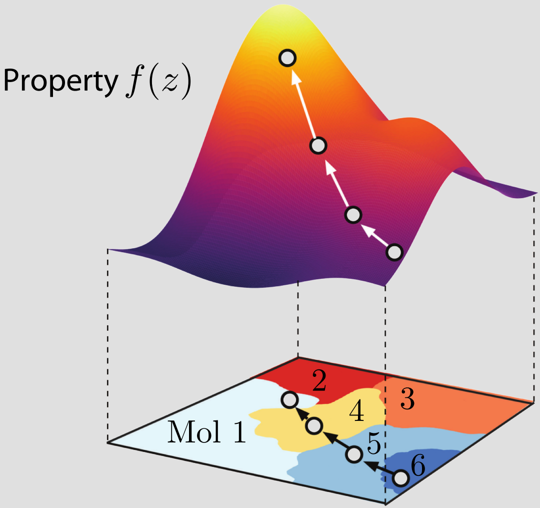
Encoding-decoding

Train VAE on SMILES:


Any point reconstructs valid molecule
c1ccccc1
Discrete Structure SMILES
ENCODER Neural Network
CONTINUOUS MOLECULAR REPRESENTATION Latent Space
DECODER Neural Network
c1ccccc1
Discrete Structure SMILES

Rafael Gómez-Bombarelli et. al. https://arxiv.org/abs/1610.02415

Bayesian optimization



Rafael Gómez-Bombarelli et. al. https://arxiv.org/abs/1610.02415

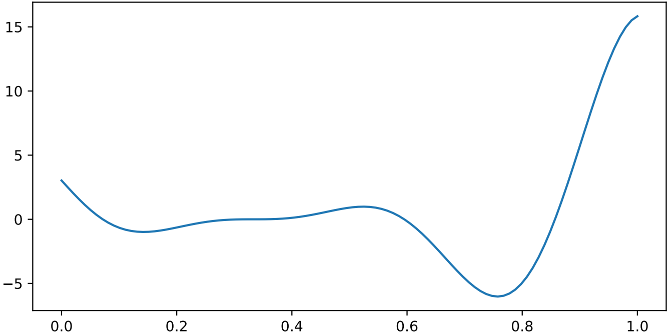
Optimize property like effectiveness against cancer.

While **True**:

1. Find maximum of acquisition function
2. Perform trials on new molecule

Basic procedure of bayesian optimization I

Take a look as the following function



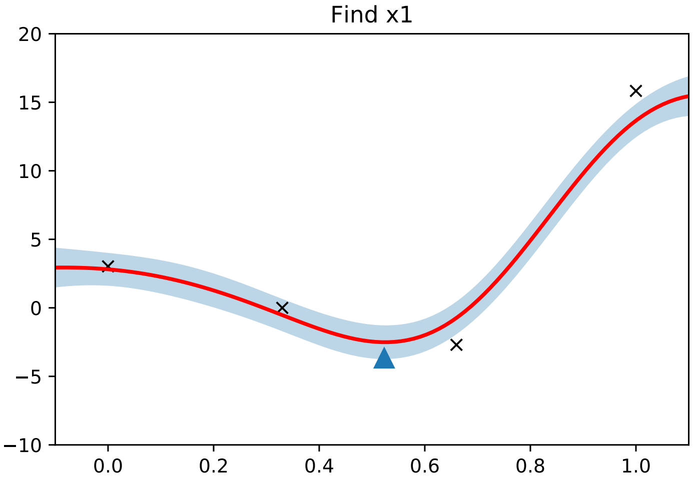
It has two local minimums and one global minimum. We might think of this stange-shape function as the `test error - regularization coefficient` graph of some machine learning model(which we don't have access to this function!)

Basic procedure of bayesian optimization II

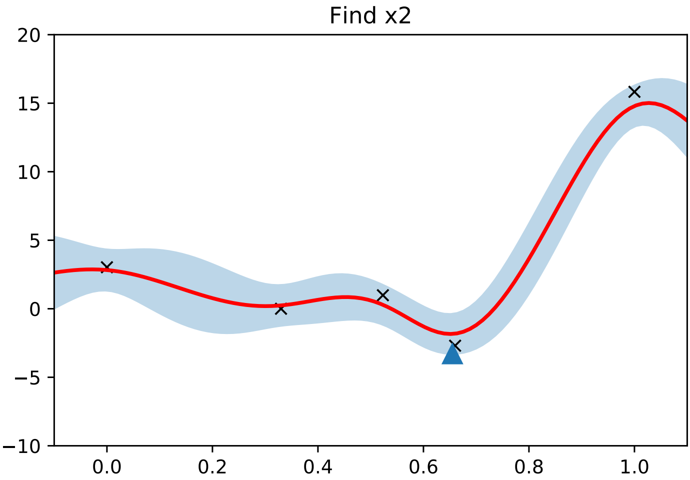
Think about the process of how we tune the hyper parameter of a machine learning model.

- We first try some points and evaluate the function value on these points and get some initial function estimates to have a general idea of the function.
- Then from these estimated points and their function values, we have some sense what the whole function should be like(This is called the surrogate of the original function). We choose to evaluate some point on the original function according to some criterion based on the surrogate to update our posterior to the problem.

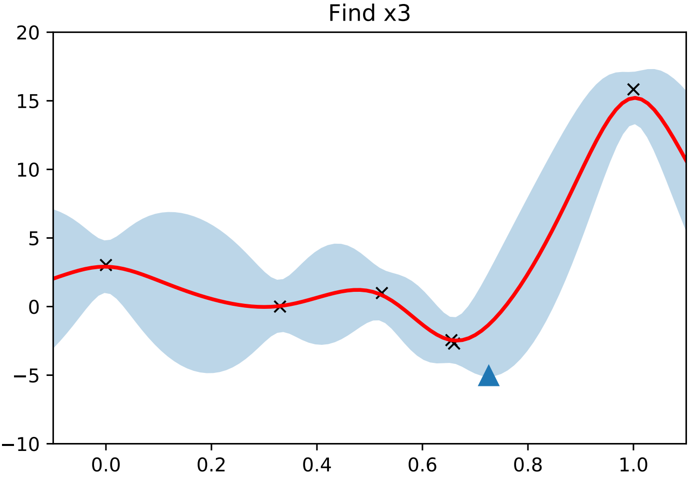
The procedure I



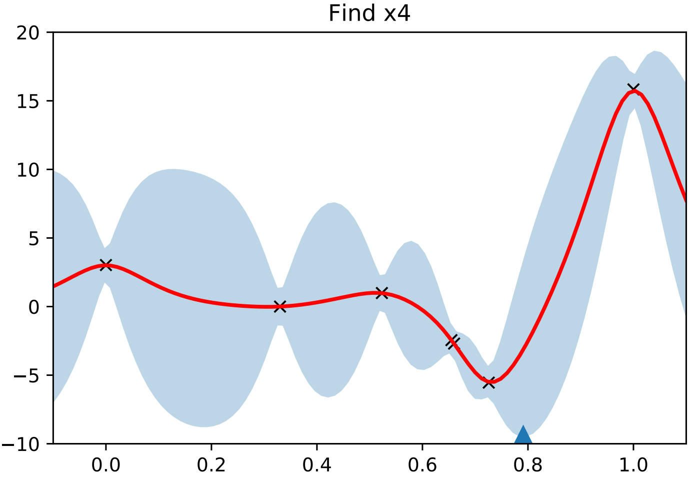
The procedure II



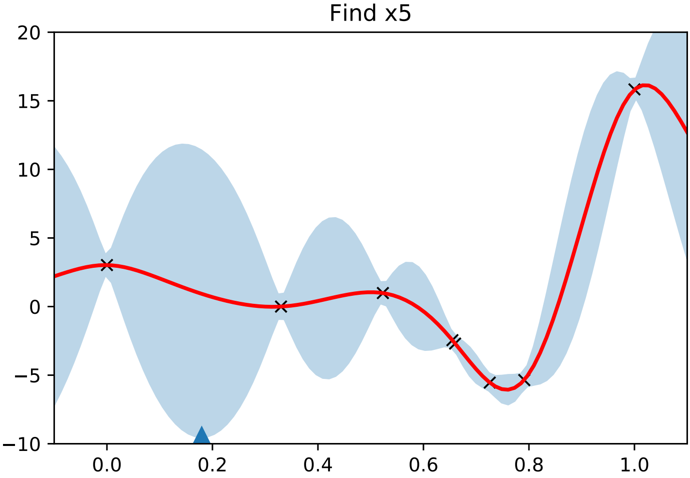
The procedure III



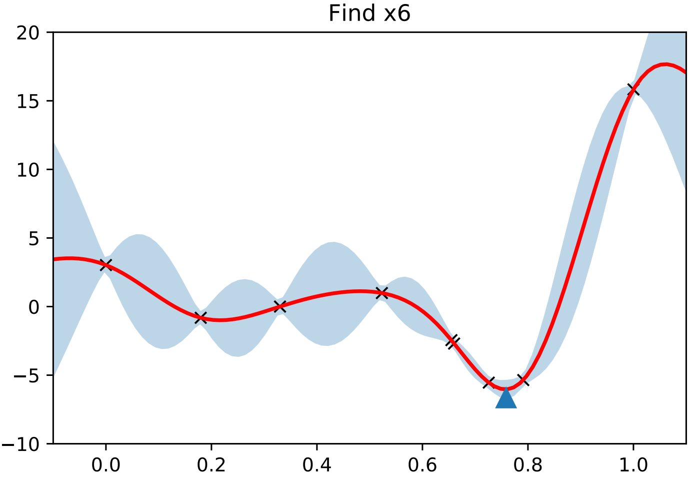
The procedure IV



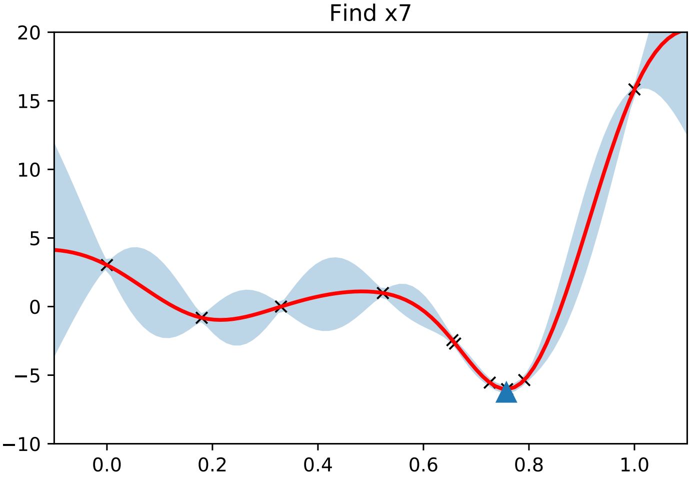
The procedure V



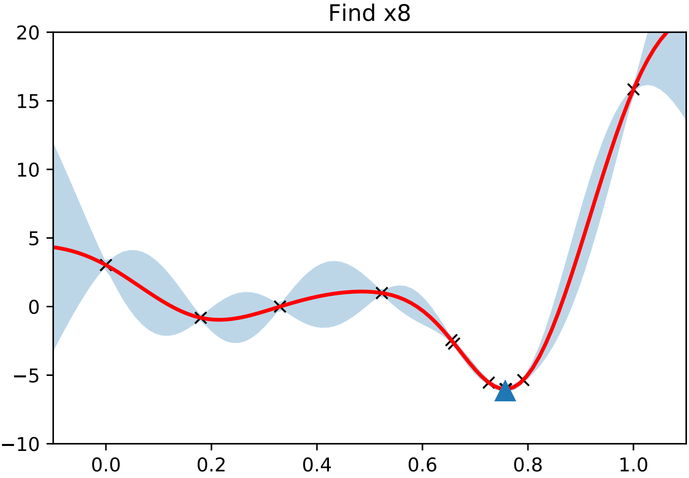
The procedure VI



The procedure VII



The procedure VIII



The procedure IX

The red line represents the mean of the distribution of surrogate

The procedure X

function while the blue area represents the error area.

Summary of the procedure

- Like human, choosing the prior that capture the structure of the problems is very important. (For example, think about why do we assume that the surrogate function is smooth.)
- the general procedure
  - The procedure of bayesian optimization is like we build a surrogate of the black box function that we are trying to optimize.
  - We perform optimization on the surrogate function distributions based on some criterion to find the good point to evaluate.
  - We update our belief of the blackbox function i.e. surrogate function with the newly evaluated point.

Introduction of Gaussian Process

So how do we represent the surrogate function distribution and update our belief about this?

- Gaussian Process a.k.a gp provides a great framework to model the distribution of functions. i.e. each sample from gp is a function. or we can think of it as a very high dimension vector.
- gp provide a systematic way to update our belief i.e. inference for the posterior distribution.

Formal representation

Formally, a gaussian process is determined by two functions, the mean function $m(\cdot)$ and the covariance function $k(\cdot,\cdot)$. A sample from gp is described as follows:

$$f(\cdot) \sim \mathcal{GP}(\mu(\cdot), k(\cdot,\cdot))$$

The sample from GP $f(\cdot)$ satisfies that any d dimensional samples $(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_d))$ is from Gaussian distribution

$$(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_d))^\top \sim \mathcal{N}(\mu(\mathbf{x}_1),\ldots,\mu(\mathbf{x}_d)^\top, \mathbf{K}(\mathbf{x}_1,\ldots,\mathbf{x}_d))$$

The selection of the mean function $\mu(\cdot)$ and kernel function $k(\cdot,\cdot)$ represents our belief about the sampled function from $\mathcal{GP}$.

Two popular kernels

Now let's see what kind of belief we make when specifying different $\mu(\cdot)$ and $k(\cdot,\cdot)$. We will see two kinds of kernels(there are actually tons of other kernels out there!)

Squared exponential

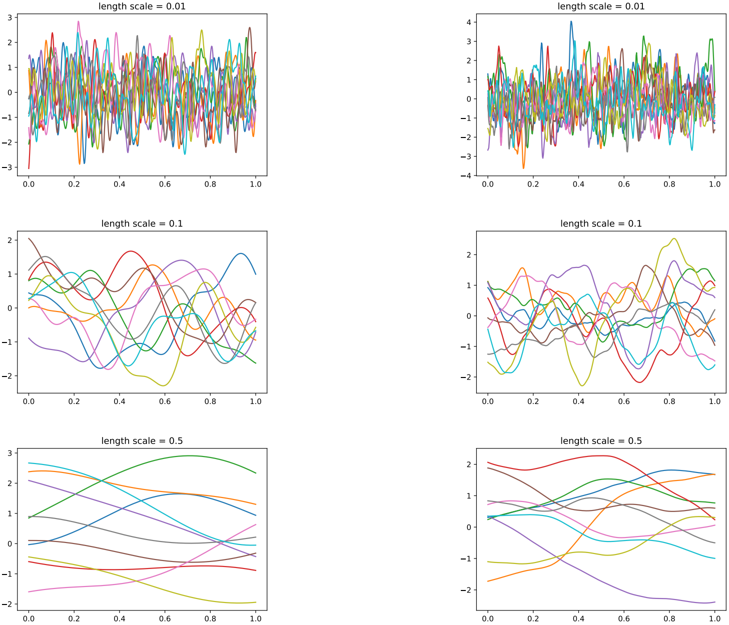
$$k(x, x') = \exp(-\frac{1}{2\tau^2}||x - x'||^2)$$

Function samples from squared exponential kernel GP is inifinitely differentiable which is not that desired in real life. People turn to use Matérn kernel[1] for less smooth function samples. Samples from Matérn $\frac{5}{2}$ kernel GP is 2-times differentiable.

$$k(x, x') = \sigma^2(1 + \frac{\sqrt{5}d}{\rho} + \frac{5d^2}{3\rho^2})\exp(-\sqrt{5d}\rho)$$

The three figures on the left are sampled from RBF kernel, while the right ones are sampled from Matérn kernel GP.

[1] https://en.wikipedia.org/wiki/Mat%C3%A9rn_covariance_function


length scale = 0.01

length scale = 0.01

length scale = 0.1

length scale = 0.1

length scale = 0.5

length scale = 0.5

Make our assumptions in mean and covariance function

Jittering samples may be from financial data while the smoother samples may come from something like weather data.
Choosing the prior depends on what you know about the problems.
For example:

- If we think our data is uptrending. We may assume the mean function has the form $\mu(x) = ax + b, \quad a > 0$.
- We can choose different kinds of Matérn kernel based on our assumptions about the order of differentiability of the function that we are trying to model.

Posterior Inference

We already know how to sample function from a GP model. Also, we know what are samples like when we use different kernel functions. Once specifying a form of kernel function, we want to know what infer the parameter. There are basically three kinds of inference methods.(From computationally efficient to inefficient and from approximate to accurate under our prior assumption.) They are

- Maximum likelihood estimation
- Maximum a posterior
- Full bayesian inference

Criterion on what point to evaluate next I

We already know how to construct the surrograte function based on our assumption and update this surrogate after observing new data. Now we introduce how to select the new point to evaluate based on this surrogate function.
The basic principle of point selection is to weight the tradeoff between **exploration** and **exploitation**. Noting that our ultimate goal is to find the lowest point of some black box function. We already have some estimated function points. Either we can go deeper for lower function value points or we can explore some points that we are highly uncertain.

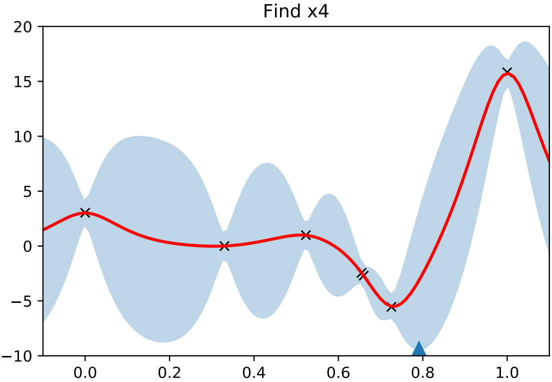
Criterion on what point to evaluate next II

An creterion that is easy to understand is Lower Confidence Bound acquisition function

$$\alpha(x) = \mu(x) - \kappa\sigma(x)$$

If we take $\kappa = 2$ this is just the $\mu(x) - 2\sigma(x)$ line. Think about how it balance the exploration and exploitation. $\mu(x)$ corresponds to exploration while $\sigma(x)$ corresponds to the exploitation.

Criterion on what point to evaluate next III



Alternaitve acquisition functions

Alternative choices of acquisition function includes:

- Expected Improvement
- Knowledge gradient
- Entropy Search

Extension of the Bayesian Optimization

- Noisy evaluations
- Parallel Evaluations
- Constraints
- Multi-Fidelity and Multi-Information Source Evaluations
- Random Environmental Conditions and Multi-Task Bayesian Optimization

Summary