Optimization Method

1. Bayesian Optimization

Usage: Typically, Bayesian optimization is used to solve the maximum/minimum optimization problem $\min / \max_{x \in S} f(x)$. The function $f(x)$ will be viewed as a black box and typically difficult to evaluate.

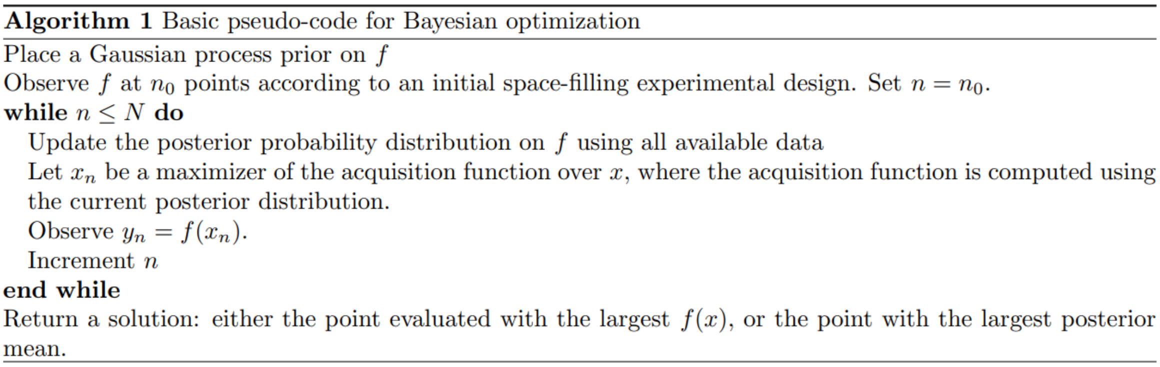
1.1 Main Idea

1. using Gaussian process to calculate the **mean value** and the **variance** of the posterior probability distribution from current dataset.

   Note that we can also use other models to model the function $f(x)$ as long as the surrogate model can output a posterior distribution.

   In the paper "Model-based RL for biological sequence", they used two BO baselines. One uses Gaussian process as the surrogate model, while the other uses the ensemble of neural networks as the surrogate model.

2. calculate the **Acquisition function** (AF) $\mu(x)$ using the mean value and the variance. The details about the AF will be described later.
3. get the next data point x to be sampled by maximizing $\mu(x)$
4. calculate the observation $f(x)$ with the new data point and add $(x, f(x))$ to the dataset.
5. repeat the above 4 steps until stop



```
**Algorithm 1** Basic pseudo-code for Bayesian optimization
Place a Gaussian process prior on $f$
Observe $f$ at $n_0$ points according to an initial space-filling experimental design. Set $n = n_0$.
**while** $n \leq N$ **do**
    Update the posterior probability distribution on $f$ using all available data
    Let $x_n$ be a maximizer of the acquisition function over $x$, where the acquisition function is computed using the current posterior distribution.
    Observe $y_n = f(x_n)$.
    Increment $n$
**end while**
Return a solution: either the point evaluated with the largest $f(x)$, or the point with the largest posterior mean.
```

1.2 Some popular Acquisition functions

1.2.1 Probability of Improvement (PI)

The next sampled data point should have the highest probability of improvement over the current maximum value $f(x^+)$.

$$PI(x) = P(f(x) \geq f(x^+) + \epsilon) = \Phi(\frac{\mu(x) - f(x^+) - \epsilon}{\sigma(x)})$$

where ϵ is a small positive number that can be used to **balance exploration and exploitation**. A larger ϵ encourages more exploration. With larger ϵ, there could be less data points with values greater than x^+ by ϵ. In this case, most data points will be sampled with the equal probability.

x^+ is the current max value.

Φ denotes the CDF of the standard normal distribution.

$\mu(x)$ and $\sigma(x)$ are the mean and covariance of $f(x)$.

1.2.2 Expected improvement (EI)

Compared with PI which only considers how likely the improvement is, EI can also take how much the improvement is.

Main idea: choose the next query point as the one which has the highest expected improvement over the current max $f(x^+)$.

Similar to PI, ϵ can be used to balance the exploration and exploitation. Larger ϵ leads to smoother distribution of Z value.

$$EI(x) = \begin{cases} (\mu_t(x) - f(x^+) - \epsilon)\Phi(Z) + \sigma_t(x)\phi(Z), & \text{if } \sigma_t(x) > 0 \\ 0, & \text{if } \sigma_t(x) = 0 \end{cases}$$

$$Z = \frac{\mu_t(x) - f(x^+) - \epsilon}{\sigma_t(x)}$$

where $\Phi(\cdot)$ indicates CDF and $\phi(\cdot)$ indicates pdf.

1.2.3 UCB bound

$UCB(x) = \mu(x) + \epsilon\sigma(x)$

ϵ can be used to balance the exploration and exploitation. The mean value $\mu(x)$ represents exploitation. The variance $\sigma(x)$ measures the uncertainty of the prediction, and thus represents exploration.

The next data point $x = \arg\max UCB(x)$

1.2.4 Thompson Sampling

Choose the action that can maximize the expected reward.

- Draw the model parameter θ from the Gaussian process posterior $P(f|D)$ over the function $f(x)$.
- Select the data point $x \sim \arg\max_{x \in X} \hat{f}_\theta(x)$
- Get the value $f(x)$ and add the data point $(x, f(x))$ into the dataset D.

In the above algorithm, the exploration is done by sampling the model parameters θ. Because data points which haven't been visited will have large uncertainty, the predictions of these data points with GP have large uncertainty and can be large values. These large values, in turn, encourage the model to visit unexplored data points.

Unlike UCB which is a deterministic algorithm, Thompson sampling incorporates the uncertainty or probability by sampling from the posterior distribution and thus is a stochastic algorithm. According to the paper[1],

> It appears that Thompson sampling is more robust than UCB when the delay is long. Thompson sampling alleviates the influence of delayed feedback by randomizing over actions; on the other hand, UCB is deterministic and suffers a larger regret in case of a sub-optimal choice.

2. Thompson Sampling for the multi-armed bandit problem

[1] Chapelle, Olivier, and Lihong Li. "An empirical evaluation of thompson sampling." *Advances in neural information processing systems* 24 (2011): 2249-2257. ↩