DS 801: Advanced Optimization for Data Science **KAIST, Spring 2024**
Lecture #24: Bayesian Optimization June 5, 2024
Lecturer: Dabeen Lee

1 Outline

In this lecture, we cover

- motivation for Bayesian optimization,
- Bayesian linear regression,
- Gaussian process,
- Gaussian process regression,
- Gaussian process optimization.

2 Motivation for Bayesian Optimization

So far, we have covered the simultaneous optimistic optimization (SOO) algorithm, which is basically a grid search-based method, and two supervised learning-based methods, kernel ridge regression and optimization with a neural network with ReLU activation. In particular, the latter two are based on learning "best-fit" models.

While search-based methods and best-fit methods are widely used in practice, they often fail to deal with "expensive-to-evaluate" functions and high-dimensional search spaces. Moreover, best-fit methods typically require a large-data regime. Hence, it is difficult to apply those methods to application settings where it is not feasible to test the underlying objective function for many times. Let us provide a few of such applications.

- Drug Discovery and Material Science: Evaluating the effectiveness of new compounds in drug discovery or the properties of new materials involves expensive and time-consuming laboratory experiments or simulations.
- A/B Testing in Marketing: In marketing, evaluating different strategies or configurations (e.g., ad placements, pricing strategies) often involves running A/B tests that can be expensive in terms of time and resources.
- Engineering Design: Designing components such as aircraft wings, automotive parts, or microprocessors often involves running sophisticated simulations (e.g., computational fluid dynamics or finite element analysis) that can take hours or days to complete.
- Hyperparameter Tuning in Machine Learning: For a given set of hyperparameters, training complex models like deep neural networks, support vector machines, or ensemble methods can be time-consuming and computationally expensive.

Hence, we want to develop a method that can suggest and find a good-quality solution even with a small amount of data on function evaluations. In this lecture, we introduce **Bayesian optimization**, which is suitable for a data-poor regime.

Bayesian optimization essentially takes the **Bayesian approach** of maintaining a probabilistic model of the unknown objective function. Based on the probabilistic model, we choose next decision points to be evaluated. This probabilistic approach provides a measure of uncertainty about the function's behavior, which allows for a more strategic exploration of the search space. By cleverly balancing exploration and exploitation, Bayesian optimization can identify optimal solutions more efficiently and effectively. This makes Bayesian optimization particularly useful in scenarios where function evaluation is costly.

3 Bayesian Linear Regression

As before, let us start with the case of linear models. Our hypothesis is that the unknown objective function f is given by

$$f(x) = \theta^\top x$$

where the coefficient vector $\theta \in \mathbb{R}^d$ is unknown to the decision-maker. In **Bayesian linear regression**, we assume that θ is sampled from a known **prior distribution** with density $p(\theta)$. A common choice is a multivariate Gaussian distribution of the form $\mathcal{N}(0, \tau^2 I_d)$ where $\tau > 0$ and I_d is the $d \times d$ identity matrix.

Suppose that we have a set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ of n i.i.d. data points sampled from some unknown distribution. Moreover, we have

$$y_i = \theta^\top x_i + \epsilon_i, \quad i \in [n]$$

where $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. noise variables sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$ for some $\sigma > 0$. Then we have $y_i - \theta^\top x_i \sim \mathcal{N}(0, \sigma^2)$ with density

$$p\left(y_i \mid x_i, \theta\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \theta^\top x_i)^2}{2\sigma^2}\right).$$

Having obtained the data set $\mathcal{D}$, we deduce the **posterior distribution** of the coefficient vector θ with density

$$p\left(\theta \mid \mathcal{D}\right) = \frac{p(\theta)p(\mathcal{D} \mid \theta)}{\int_\omega p(\omega)p(\mathcal{D} \mid \omega)d\omega} = \frac{p(\theta)\prod_{i=1}^n p(y_i \mid x_i, \theta)}{\int_\omega p(\omega)\prod_{i=1}^n p(y_i \mid x_i, \omega)d\omega}$$

where the first equality is due to **Bayes' rule** and the second equality follows from

$$p(\mathcal{D} \mid \theta) = \prod_{i=1}^n p(y_i \mid x_i, \theta).$$

Based on the posterior distribution for θ, we want to predict the response value y associated with a solution x. This can be done by considering the **posterior predictive distribution** with density

$$p(y \mid x, \mathcal{D}) = \int_\theta p(y \mid x, \theta)p(\theta \mid \mathcal{D})d\theta.$$

If we assume that the prior distribution is given by $\mathcal{N}(0, \tau^2 I_d)$ and $y_i - \theta^\top x_i \sim \mathcal{N}(0, \sigma^2)$, then we can derive we can characterize $p(y \mid x, \theta)$ and $p\left(\theta \mid \mathcal{D}\right)$. Based on these, we can also compute the posterior predictive distribution. To explain this, we define Φ and y as

$$\Phi = (x_1, \ldots, x_n) \in \mathbb{R}^{d \times n}, \quad y = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n.$$

Then it is known that

$$\begin{aligned}\theta \mid \mathcal{D} \quad &\sim \quad \mathcal{N}\left(\left(\Phi\Phi^\top + \frac{\sigma^2}{\tau^2}I_d\right)^{-1}\Phi y, \quad \sigma^2\left(\Phi\Phi^\top + \frac{\sigma^2}{\tau^2}I_d\right)^{-1}\right),\\ y \mid x, \mathcal{D} \quad &\sim \quad \mathcal{N}\left(x^\top\left(\Phi\Phi^\top + \frac{\sigma^2}{\tau^2}I_d\right)^{-1}\Phi y, \quad \sigma^2 x^\top\left(\Phi\Phi^\top + \frac{\sigma^2}{\tau^2}I_d\right)^{-1}x + \sigma^2\right).\end{aligned}$$

In the last lecture, we observed that for any $\lambda > 0$,

$$\left(\Phi\Phi^\top + \lambda I_d\right)^{-1}\Phi = \Phi\left(\Phi^\top\Phi + \lambda I_n\right)^{-1}.$$

Therefore,

$$y \mid x, \mathcal{D} \quad \sim \quad \mathcal{N}\left(x^\top\Phi\left(\Phi^\top\Phi + \frac{\sigma^2}{\tau^2}I_n\right)^{-1}y, \quad \sigma^2 x^\top\Phi\left(\Phi^\top\Phi + \frac{\sigma^2}{\tau^2}I_n\right)^{-1}x + \sigma^2\right).$$

One may select a solution x, based on this posterior predictive distribution. We will discuss this further after explaining the kernel extension of Bayesian linear regression.

Note that the mean value of $y \mid x, \mathcal{D}$ is equivalent to the prediction from the ridge regression formulation with $\lambda = \sigma^2/\tau^2$. Note that Bayesian linear regression considers noise in function evaluation, which results in the variance term. As a result, Bayesian linear regression takes into account uncertainty in prediction as well.

4 Gaussian Process

Before we extend the framework of Bayesian linear regression to general non-linear functions, let us discuss what is known as **Gaussian processes** defined based on multivariate Gaussian distributions. We say that a function f is sampled from a Gaussian process if for any finite collection of solutions $x_1, \ldots, x_n \in C$, their function values $f(x_1), \ldots, f(x_n)$ follow a multivariate Gaussian distribution. To be more specific, we assume that there exist a mean function $\mu(x)$ and a covariance function $k(x, x')$ such that

$$\begin{aligned}\mu(x) &= \mathbb{E}\left[f(x)\right],\\ k(x,x') &= \mathbb{E}\left[(f(x)-\mu(x))(f(x')-\mu(x'))\right].\end{aligned}$$

Then f sampled from the Gaussian process with mean function μ and covariance function k satisfies the following. For any $x_1, \ldots, x_n \in C$,

$$\begin{pmatrix} f(x_1)\\ \vdots \\ f(x_n)\end{pmatrix} \quad \sim \quad \mathcal{N}\left(\begin{pmatrix}\mu(x_1)\\ \vdots\\ \mu(x_n)\end{pmatrix}, \quad \begin{pmatrix} k(x_1,x_1) & \cdots & k(x_n,x_1)\\ \vdots & \ddots & \vdots\\ k(x_1,x_n) & \cdots & k(x_n,x_n)\end{pmatrix}\right).$$

To simplify, we often use notation

$$f(x) \quad \sim \quad \mathcal{GP}\left(\mu(x), k(x,x')\right).$$

Moreover, the covariance function k is also referred to as a kernel function, as in kernel ridge regression. Recall that for kernel ridge regression, we want the kernel function k to induce the associated matrix

$$K = \begin{pmatrix} k(x_1,x_1) & \cdots & k(x_n,x_1)\\ \vdots & \ddots & \vdots\\ k(x_1,x_n) & \cdots & k(x_n,x_n)\end{pmatrix}$$

symmetric and positive semidefinite. To define a proper Gaussian process, we also require this condition. As before, the linear kernel, the polynomial kernel, the squared exponential kernel, and the Matérn kernel satisfy the condition.

5 Gaussian Process Regression

Next we present the framework of **Gaussian process regression**, which generalizes Bayesian linear regression. As before, we have a set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ of n i.i.d. data points sampled from some unknown distribution. We assume that the data points are given by a function f sampled from a Gaussian process. To be specific, we have

$$y_i = f(x_i) + \epsilon_i, \quad i \in [n]$$

where $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. noise variables sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$ for some $\sigma > 0$. As f is a sample from a Gaussian process, there exists some mean function μ and kernel function k such that $f(x) \sim \mathcal{GP}\left(\mu(x), k(x, x')\right)$. In practice, we often assume that $\mu(x) = 0$. Hence,

$$f(x) \quad \sim \quad \mathcal{GP}\left(0, k(x, x')\right).$$

Based on our development of Gaussian processes from the previous section, let us explain how to predict the function value $f(x)$ of a given solution. By definition, $f(x_1), \ldots, f(x_n)$, and $f(x)$ follow

$$\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \\ f(x) \end{pmatrix} \quad \sim \quad \mathcal{N}\left(\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_n, x_1) & k(x, x_1) \\ \vdots & \ddots & \vdots & \vdots \\ k(x_1, x_n) & \cdots & k(x_n, x_n) & k(x, x_n) \\ k(x_1, x) & \cdots & k(x_n, x) & k(x, x) \end{pmatrix}\right).$$

For the kernel function k, we define $k(x)$ and K as follows.

$$k(x) = \begin{pmatrix} k(x_1, x) \\ \vdots \\ k(x_n, x) \end{pmatrix} \in \mathbb{R}^n, \quad K = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_n, x_1) \\ \vdots & \ddots & \vdots \\ k(x_1, x_n) & \cdots & k(x_n, x_n) \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

Then

$$\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \\ f(x) \end{pmatrix} \quad \sim \quad \mathcal{N}\left(0, \quad \begin{bmatrix} K & k(x) \\ k(x)^\top & k(x, x) \end{bmatrix}\right).$$

Since $y_i = f(x_i) + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, it follows that

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \\ f(x) \end{pmatrix} \quad \sim \quad \mathcal{N}\left(0, \quad \begin{bmatrix} K + \sigma^2 I_n & k(x) \\ k(x)^\top & k(x, x) \end{bmatrix}\right).$$

Then the posterior distribution of f is given by

$$f(x) \mid \mathcal{D} \quad \sim \quad \mathcal{N}\left(k(x)^\top \left(K + \sigma^2 I_n\right)^{-1} y, \quad k(x, x) - k(x)^\top \left(K + \sigma^2 I_n\right)^{-1} k(x)\right)$$

where $y = (y_1, \ldots, y_n)^\top$. Again, the mean function of the posterior distribution is the same as the prediction from the kernel ridge regression formulation with $\lambda = \sigma^2$.

6 Gaussian Process Optimization

In this section, we consider black-box optimization where the goal is to "maximize" an unknown objective function:

$$\max_{x\in C} \quad f(x).$$

Note that a maximization problem can be equivalently expressed as a minimization problem by taking $-f$. In contrast to our previous frameworks for black-box optimization, we adopt the regret minimization framework described as follows. Suppose that we make a sequence $x_1, \ldots, x_T$ of solutions in C over T steps. Then we define the notion of regret as

$$\mathrm{Regret}(T) = T \cdot \max_{x\in C} f(x) - \sum_{t=1}^{T} f(x_t).$$

A Bayesian approach for this framework would assume a Gaussian process prior and proceed by updating the posterior. In particular, we will discuss the seminal work by Srinivas et al. [SKKS10].

Given $x_1, \ldots, x_t$ and their associated noisy evaluations $y_1, \ldots, y_t$, we define empirical mean function $\mu_t(x)$ and empirical covariance function $\sigma_t(x)$ based on Gaussian process regression as follows. For a given kernel function k, let $k_t(x)$ and K_t be defined as

$$k_t(x) = \begin{pmatrix} k(x_1, x) \\ \vdots \\ k(x_t, x) \end{pmatrix} \in \mathbb{R}^t, \quad K_t = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_t, x_1) \\ \vdots & \ddots & \vdots \\ k(x_1, x_t) & \cdots & k(x_t, x_t) \end{pmatrix} \in \mathbb{R}^{t\times t}.$$

Then we define μ_t and σ_t so that they satisfy

$$\mu_t(x) = k_t(x)^\top (K_t + \sigma^2 I_t)^{-1} y,$$
$$\sigma_t(x)^2 = k(x,x) - k_t(x)^\top (K_t + \sigma^2 I_t)^{-1} k_t(x).$$

For $t = 0$, we have $\mu_0(x) = 0$ and $\sigma_0(x)^2 = k(x,x)$. Then the celebrated **GP-UCB** algorithm works as follows. Here, the parameter $\beta_1, \ldots, \beta_T$ are chosen so that for all $t \in [T]$, we satisfy the

Algorithm 1 The GP-UCB algorithm

```
Input: parameters $\beta_1, \ldots, \beta_T$.
for $t = 1, \ldots, T$ do
    Choose
        $x_t = \mathrm{argmax}_{x\in C} \mu_{t-1}(x) + \sqrt{\beta_t}\sigma_{t-1}(x)$
    Obtain a noisy evaluation of $x_t$ given by
        $y_t = f(x_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma).$
end for
```

following with high probability.

- When C is finite,
$$|f(x) - \mu_{t-1}(x)| \le \sqrt{\beta_t}\sigma_{t-1}(x)$$
holds for every $x \in C$.

- When C is compact,

$$|f(x_t) - \mu_{t-1}(x_t)| \leq \sqrt{\beta_t}\sigma_{t-1}(x_t).$$

This is why the algorithm is called GP-UCB where GP stands for Gaussian process and UCB stands for upper confidence bounds. Basically, by satisfying the condition, $\mu_{t-1}(x) + \sqrt{\beta_t}\sigma_{t-1}(x)$ is an upper bound on $f(x)$ with high probability. The GP-UCB algorithm chooses a solution that maximizing the upper bound function at each step. This is similar in spirit to the UCB algorithm for multi-armed bandits.

References

[SKKS10] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 1015–1022, Madison, WI, USA, 2010. Omnipress. 6