MIT 6.7220/15.084 — Nonlinear Optimization Thu, Apr 25th 2024

Lecture 16
Bayesian optimization

Instructor: Prof. Gabriele Farina (✉ gfarina@mit.edu)*

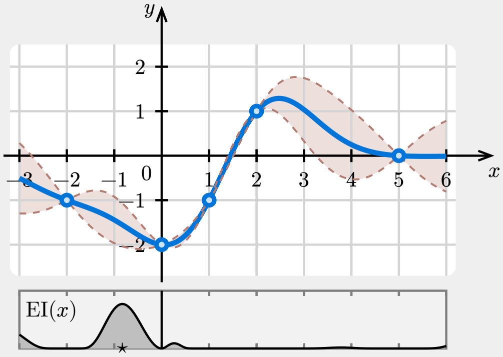
Having seen first-order and second-order methods, today we briefly take a look at a completely different approach to optimization: Bayesian optimization. This method is particularly useful when the function to be optimized is expensive to evaluate, and we have no information about its gradient. Bayesian optimization is a *heuristic* approach that is applicable to *low-dimensional* optimization problems. Since it avoids using gradient information altogether, it is a popular approach for hyperparameter tuning, architecture search, *et cetera*.

1 A Bayesian approach to optimization

In Bayesian optimization, we are still interested in finding the minimizer of a function. However, unlike first- and second-order methods, we do not assume access to the Gradient or Hessian of the function. Rather, we assume only that given any point x, we can query f at x and obtain the value $f(x)$.

In the Bayesian approach, instead of finding the best x given f, we *try to figure out the best model of f given its evaluation at the previously observed points x*. By conditioning on the observed points, we can then extract an expected function f and a confidence interval, and decide what point to query next based on this information. In other words, Bayesian optimization is a sophisticated way of fitting the data points with a function, and then optimistically querying the point where the function is expected to be the smallest.

Example 1.1. To build intuition, let us consider the example depicted on the right. Five points (the blue dots) have been computed for f; the minimum value observed so far is $f_\star = -2$. Out of the prior over all functions, a posterior was computed by conditioning on these points. The **blue** solid curve represents the mean $\mu(x)$ of the posterior, while the **brown** shaded area represents one standard deviation $\pm\sigma(x)$ away from the mean.



The **expected improvement** (EI) is shown below, which is a heuristic that suggests where to query next. Specifically,

$$\mathrm{EI}(x) := \mathbb{E}_{\tilde f}\Big[\max\Big\{0, f_\star - \tilde f(x)\Big\}\Big],$$

where at each x, the expectation is with respect to the conditional distribution of $f(x)$.

[*] These notes are class material that has not undergone formal peer review. The TAs and I are grateful for any reports of typos.

Of course, many details are missing from this example: What was the prior over the functions? How was the posterior computed? How was the expected improvement computed?

In the following, we will address these aspects in more detail.

2 Gaussian processes and regression

In order to keep the conditioning tractable, a very popular modeling choice is that the function f is sampled from a *Gaussian process*. A Gaussian process postulates that the function f must be such that for any finite set of points $x_1, x_2, ..., x_t \in \mathbb{R}^n$, the vector $(f(x_1), f(x_2), ..., f(x_t))$ is distributed as a multivariate Gaussian. This means that a Gaussian process is completely defined by the following two quantities:

- the *mean* function $m(x)$, assigning to each x the expected value $\mathbb{E}[f(x)]$; and
- the *covariance* function $\mathrm{K}(x, y)$, assigning to each pair of points x, y the covariance between $f(x)$ and $f(y)$.

2.1 Kernel functions

The covariance function $\mathrm{K}(x, y)$ is also called the *kernel* of the Gaussian process. Intuitively, it measures how correlated the values of $f(x)$ and $f(y)$ are. It is reasonable that as x and y get father apart, values of $f(x)$ do not inform the values of $f(y)$, and *vice versa*; so the covariance should decrease. In practice, the most popular choice for the kernel is the *radial basis function* (RBF) kernel,[1] defined as follows.

Definition 2.1. The *radial basis function* (RBF) kernel—also known as squared exponential kernel or Gaussian kernel—is defined as

$$\mathrm{K}(x, y) \coloneqq k \cdot \exp\left\{ -\frac{\|x - y\|_2^2}{2\sigma^2} \right\},$$

where $k > 0$ and $\sigma > 0$ are parameters of the kernel.

The radial basis function kernel is a popular choice because it is smooth and has a simple form. It is also a *universal kernel*, meaning that any continuous function can be approximated arbitrarily well by a Gaussian process with an RBF kernel [MXZ06].

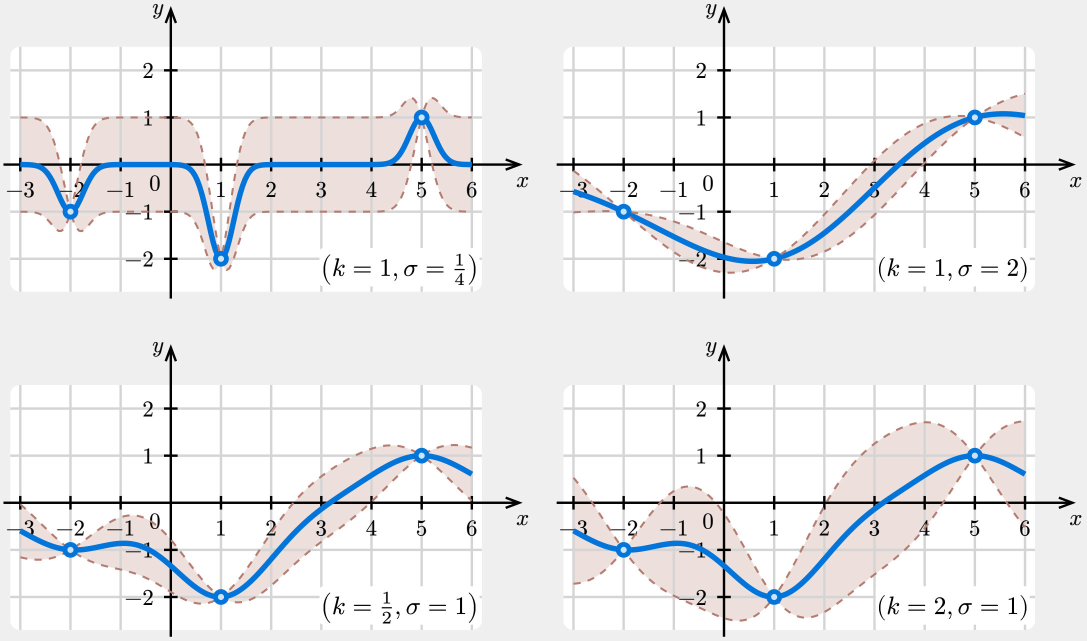
2.2 The effect of the RBF kernel parameters

While we have not yet discussed how to carry out the computations necessary for conditioning a Gaussian process on observed points, we can already speculate how the parameters of the RBF kernel affect the regression. Specifically,

- The parameter σ controls how fast the correlation between neighboring points decays. If σ is large, the function will be very smooth, because values will remain correlated for a long distance; if σ is small, the function will be very wiggly and regressing often to the mean.
- The parameter k controls the *amplitude* of the function. In particular, there will be more variance in the function values (applied uniformly throughout the domain). This is easy to appreciate by considering that the variance of the Gaussian process at any point x is $\mathrm{K}(x, x) = k$.

[1] For example, this is the default in `scikit-learn`, a popular Python machine learning toolkit.

Example 2.1. In the examples below, we confirm the effect of varying the RBF kernel parameters k and σ parameters when conditioning on $f(-2) = -1$, $f(1) = -2$, and $f(5) = 1$, under the common assumption that $m(x) = 0$ at all x.



In all plots, the blue curve represents the mean of the conditioned Gaussian process, while the brown shaded area represents one standard deviation.

2.3 Gaussian process conditioning

Given a set of points $x_1, x_2, ..., x_n$ and their corresponding function values, the Gaussian process is conditioned on these points to extract the posterior over the function. An important result in the theory of Gaussian processes—and the reason why these stochastic processes are often considered—is that it is possible to characterize the conditional distribution of the value of $f(x)$ for any x not in the set of observed points, in closed form.

Theorem 2.1. Let f be drawn from a Gaussian process, and $f(x_1), f(x_2), ..., f(x_t)$ be the observed function values at distinct points $x_1, x_2, ..., x_t$. The posterior distribution of the function values at any point x is a Gaussian distribution with mean and variance given by

$$\mu(x) \coloneqq m(x) + \begin{pmatrix} \mathrm{K}(x, x_1) \\ \vdots \\ \mathrm{K}(x, x_t) \end{pmatrix}^{\top} \begin{pmatrix} \mathrm{K}(x_1, x_1) & \cdots & \mathrm{K}(x_1, x_t) \\ \vdots & \ddots & \vdots \\ \mathrm{K}(x_t, x_1) & \cdots & \mathrm{K}(x_t, x_t) \end{pmatrix}^{-1} \begin{pmatrix} f(x_1) - m(x_1) \\ \vdots \\ f(x_t) - m(x_t) \end{pmatrix},$$

and

$$\sigma^2(x) = \mathrm{K}(x, x) - \begin{pmatrix} \mathrm{K}(x, x_1) \\ \vdots \\ \mathrm{K}(x, x_t) \end{pmatrix}^{\top} \begin{pmatrix} \mathrm{K}(x_1, x_1) & \cdots & \mathrm{K}(x_1, x_t) \\ \vdots & \ddots & \vdots \\ \mathrm{K}(x_t, x_1) & \cdots & \mathrm{K}(x_t, x_t) \end{pmatrix}^{-1} \begin{pmatrix} \mathrm{K}(x, x_1) \\ \vdots \\ \mathrm{K}(x, x_t) \end{pmatrix}.$$

The formulas above are known results regarding conditioning of multivariate Gaussians. While we will not prove them here, we can easily verify that indeed they guarantee interpolating the data correctly. Specifically, we have the following.

Remark 2.1. For any observed datapoint x_i, one has $\mu(x_i) = f(x_i)$ and $\sigma^2(x_i) = 0$ Furthermore, $\sigma^2(x) \geq 0$ for all $x \in \mathbb{R}^n$.

Proof. To simplify notation, let

$$\mathrm{K}(x, X) \coloneqq \begin{pmatrix} \mathrm{K}(x, x_1) \\ \vdots \\ \mathrm{K}(x, x_t) \end{pmatrix}, \qquad \text{and} \qquad \mathrm{K}(X, X) \coloneqq \begin{pmatrix} \mathrm{K}(x_1, x_1) & \cdots & \mathrm{K}(x_1, x_t) \\ \vdots & \ddots & \vdots \\ \mathrm{K}(x_t, x_1) & \cdots & \mathrm{K}(x_t, x_t) \end{pmatrix}.$$

Letting $e_i \in \mathbb{R}^n$ denote the i-th indicator vector (that is, the vector containing a 1 in the i-th position and 0 elsewhere), it is easy to check that

$$\mathrm{K}(x_i, X) = \mathrm{K}(X, X) e_i.$$

Hence, plugging in the previous relationship in the definition of $\mu(x_i)$ we can write

$$\begin{aligned} \mu(x_i) &= m(x_i) + \mathrm{K}(x_i, X)^\top \mathrm{K}(X, X)^{-1} \begin{pmatrix} f(x_1) - m(x_1) \\ \vdots \\ f(x_t) - m(x_t) \end{pmatrix} \\ &= m(x_i) + e_i^\top \mathrm{K}(X, X)^\top \mathrm{K}(X, X)^{-1} \begin{pmatrix} f(x_1) - m(x_1) \\ \vdots \\ f(x_t) - m(x_t) \end{pmatrix} \\ &= f(x_i). \end{aligned}$$

Similarly, plugging in the previous relationship in the definition of $\sigma^2(x_i)$ we have

$$\begin{aligned} \sigma^2(x_i) &= \mathrm{K}(x_i, x_i) - \mathrm{K}(x_i, X)^\top \mathrm{K}(X, X)^{-1} \mathrm{K}(x_i, X) \\ &= \mathrm{K}(x_i, x_i) - e_i^\top \mathrm{K}(X, X)^\top \mathrm{K}(X, X)^{-1} \mathrm{K}(X, X) e_i \\ &= \mathrm{K}(x_i, x_i) - \mathrm{K}(x_i, x_i) = 0. \end{aligned}$$

Finally, the nonnegativity of $\sigma^2(x)$ follows from the observation that

$$\begin{aligned} \sigma^2(x) &= \mathrm{K}(x, x) - \mathrm{K}(x, X)^\top \mathrm{K}(X, X)^{-1} \mathrm{K}(x, X) \\ &= \begin{pmatrix} 1 \\ -\mathrm{K}(X, X)^{-1} \mathrm{K}(x, X) \end{pmatrix}^\top \begin{pmatrix} \mathrm{K}(x, x) & \mathrm{K}(x, X)^\top \\ \mathrm{K}(x, X) & \mathrm{K}(X, X) \end{pmatrix} \begin{pmatrix} 1 \\ -\mathrm{K}(X, X)^{-1} \mathrm{K}(x, X) \end{pmatrix}. \end{aligned}$$

Since the middle square matrix is a covariance matrix, it is positive semidefinite, and therefore any quadratic form involving it is nonnegative. □

3 Acquisition functions

Finally, we need a heuristic to decide what point $x \in \mathbb{R}^n$ to query next. This is where the *acquisition function* comes in. The acquisition function is a heuristic that suggests where to query next based on the posterior distribution of the function. The most popular acquisition function is the *expected improvement* (EI), defined as follows.

Definition 3.1. The *expected improvement* (EI) at a point x is defined as

$$\mathrm{EI}(x) := \mathbb{E}_{\tilde f}\Big[\max\Big\{0, f_\star - \tilde f(x)\Big\}\Big],$$

where $f_\star$ is the minimum value observed so far, and the expectation is taken with respect to $\tilde f$ sampled from the conditional distribution of f. In particular, for a Gaussian process, the conditional distribution of $f(x)$ is the Gaussian with mean $\mu(x)$ and variance $\sigma^2(x)$, which means that

$$\begin{aligned}\mathrm{EI}(x) &= \frac{1}{\sigma(x)\sqrt{2\pi}}\int_{-\infty}^{+\infty} \max\{0, f_\star - v\} \exp\left\{-\frac{1}{2}\left(\frac{v-\mu(x)}{\sigma(x)}\right)^2\right\} \mathrm{d}v\\ &= \frac{1}{\sigma(x)\sqrt{2\pi}}\int_{-\infty}^{f_\star} (f_\star - v) \exp\left\{-\frac{1}{2}\left(\frac{v-\mu(x)}{\sigma(x)}\right)^2\right\} \mathrm{d}v.\end{aligned}$$

This can be computed in closed form starting from the PDF and CDF of the Gaussian distribution. [▷ Try to work out the details!]

The next point to query is then the one that maximizes the expected improvement.

Alternatively, one could use other acquisition functions, such as the *probability of improvement* (PI)

$$\mathrm{PI}(x) := \mathbb{P}_{\tilde f}\Big[f_\star - \tilde f(x) \geq 0\Big] = \mathbb{E}_{\tilde f}\Big[\mathbb{1}_{\{f_\star - \tilde f(x)\geq 0\}}\Big] = \frac{1}{\sigma(x)\sqrt{2\pi}}\int_{-\infty}^{f_\star} \exp\left\{-\frac{1}{2}\left(\frac{v-\mu(x)}{\sigma(x)}\right)^2\right\} \mathrm{d}v.$$

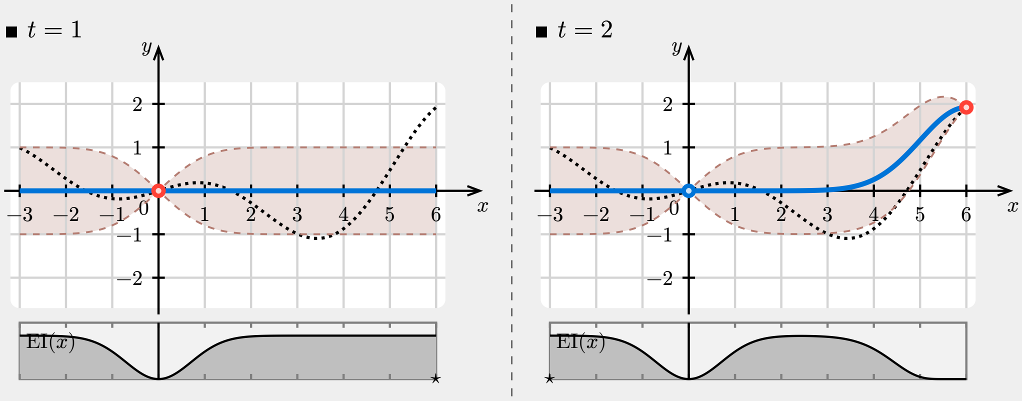
or select the minimum *lower confidence bound* (LCB)

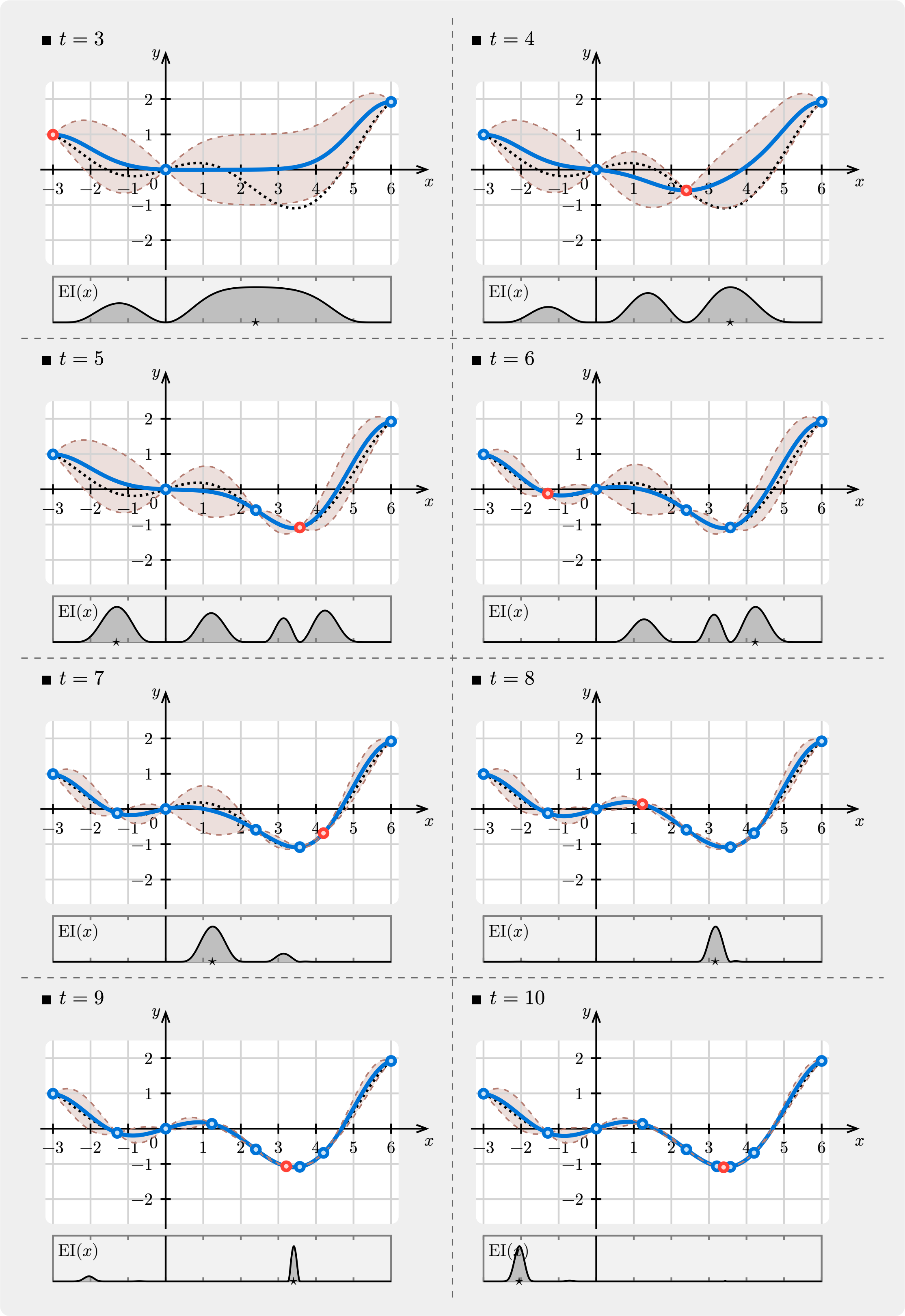
$$\mathrm{LCB}(x) := \mu(x) - \beta\sigma(x), \qquad \text{where } \beta \geq 0.$$

The choice of acquisition function is a crucial aspect of Bayesian optimization, and it is often problem-dependent.

4 Simulation

Example 4.1. We provide an extended example with the function $f(x) = \frac{1}{3}x\cos(x)$, which is superimposed as a dotted curve in the simulation of the Bayesian optimization process below. The acquisition function was set to the expected improvement, and the Gaussian process regression uses the RBF kernel with $k = 1$ and $\sigma = 1$. The last-queried point is shown in red.




■ $t = 3$
■ $t = 4$
■ $t = 5$
■ $t = 6$
■ $t = 7$
■ $t = 8$
■ $t = 9$
■ $t = 10$
y
x
−3 −2 −1 0 1 2 3 4 5 6
2 1 −1 −2
EI(x)
⋆

Bibliography

[MXZ06] C. A. Micchelli, Y. Xu, and H. Zhang, “Universal Kernels,” *Journal of Machine Learning Research*, vol. 7, no. 12, 2006.