Bayesian Optimization

Exercise 1: Expected Improvement

Derive the closed form expression of the Expected Improvement:

$$a_{\mathrm{EI}}(\mathbf{x}) = \left(f_{\min} - \hat{f}(\mathbf{x})\right) \Phi\left(\frac{f_{\min} - \hat{f}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) + \hat{s}(\mathbf{x})\phi\left(\frac{f_{\min} - \hat{f}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right).$$

Assume that $Y(\mathbf{x}) \sim \mathcal{N}\left(\hat{f}(\mathbf{x}), \hat{s}^2(\mathbf{x})\right)$.

Hints:

- For notational clarity, let's introduce y for the random variable $Y(\mathbf{x})$ and $p(y) := P(Y|\mathbf{x}, \mathcal{D}^{[t]}) = \mathcal{N}\left(\hat{f}(\mathbf{x}), \hat{s}^2(\mathbf{x})\right)$ for its probability density function.
- Start with $a_{\mathrm{EI}}(\mathbf{x}) = \mathbb{E}_y(\max\{f_{\min} - y, 0\}) = \int_{-\infty}^{\infty} \max\{f_{\min} - y, 0\}p(y)dy$.
- Decompose the integral additively depending on whether $y < f_{\min}$ or $y \geq f_{\min}$ to get rid of the maximum operator.
- It is helpful to substitute y by $u := \frac{y - \hat{f}(\mathbf{x})}{\hat{s}(\mathbf{x})}$ which implies that $y = u\hat{s}(\mathbf{x}) + \hat{f}(\mathbf{x})$. This allows you to work with standard normal distributions. Note however, that this implies performing a change of variable within the integral.
- Denote the standard normal probability density function by $\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-z^2}{2}\right)$ and the standard normal cumulative distribution function by $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} \exp\left(\frac{-u^2}{2}\right) du$.
- There is a useful identity: $\int_{-\infty}^{z} u\phi(u)du = -\phi(z)$.

Exercise 2: BO Loop

We want to implement our own BO algorithm using a Gaussian Process (GP) as surrogate model and Expected Improvement as acquisition function. Our goal is to minimize the following univariate function:

$$f : [0, 1] \to \mathbb{R}, x \mapsto 2x \cdot \sin(14x).$$

We start with an initial design of 4 points sampled uniformly at random.

(a) Write down the BO algorithm in pseudocode style.

(b) Implement the algorithm. For the GP you can for example use the `DiceKriging` package (see `?DiceKriging::km`). Use an RBF kernel. Optimize the Expected Improvement via a univariate method such as Brent's method (see `?optimize`). Use your BO algorithm to minimize f and terminate after 10 function evaluations in total.