Optimization Algorithms
Exercise 10

Marc Toussaint
Learning & Intelligent Systems Lab, TU Berlin
Marchstr. 23, 10587 Berlin, Germany

Winter 2020/21

In this exercise, you will implement Gaussian Process (GP) regression and based on that Bayesian optimization.

1 Gaussian Process Regression

You can find many existing implementations of Gaussian Processes (GPs) in the web, but for transparence (and later use within Global Optimization), you'll develop an own minimalistic implementation here.

We are given data $D = \{(x_i, y_i)\}_{i=1}^n$ of the unknown function $f : \mathbb{R}^d \to \mathbb{R}$, where $f(x_i) = y_i$ (noiseless case).

See slide 28/41. A GP approximates f as a Gaussian distribution $P(f(x) \,|\, D) = \mathcal{N}(f(x) \,|\, \mu(x), \sigma^2(x))$ with

$$\mu(x) = \kappa(x)^T (K + \sigma_0^2 I)^{-1} Y \tag{1}$$

$$\sigma^2(x) = k(x, x) - \kappa(x)^T (K + \sigma_0^2 I)^{-1} \kappa(x) \ , \tag{2}$$

where the vector $\kappa(x) = (k(x, x_1), \dots, k(x, x_n))^T \in \mathbb{R}^n$ contains covariances of x to all data points; and the matrix $K = (k(x_i, x_j))_{i,j=1}^{n,n}$ contains covariances between all data points. The vector $Y = (y_1, \dots, y_n)^T \in \mathbb{R}^n$ contains all data output values. $\sigma_0 > 0$ describes the data noise (sdv of the y_i observations), which you can choose small.

For a length-scale parameter $l \in \mathbb{R}$, and global variance parameter $a \in \mathbb{R}$, we define the squared exponential covariance function (also called "kernel") as

$$k(x, x') = a \exp(-\|x - x'\|^2 / l^2). \tag{3}$$

This is the most commonly used kernel.

First consider the function $g(x) = \sin(x) + 0.5 \cos(4x) + 0.05x^2$ for $x \in [-2\pi, 2\pi]$.

a) Uniformly sample x_i, generate a data set, compute the GP mean and variance, and plot $\mu(x)$ and $\mu(x) \pm \sigma(x)$ together with the dataset itself and the true function for increasing numbers of points in the dataset.

b) Play around with the hyperparameters a, l, σ_0. What do you observe?

c) Repeat the same for the Rastrigin function from the previous exercise with $d = 2$. In this case, plot the mean only.

You can look into the code skeleton we provide for python as a starting point.

Bonus: For those of you that are interested in coding, think about a more efficient way: The matrix $(K + \sigma_0^2 I)$ is symmetric positive definite, hence there exists a decomposition $(K + \sigma_0^2 I) = LL^T$ where L is a lower triangular matrix (called Cholesky decomposition), which is numerically more stable and requires about half the floating point operations than computing matrix inverses. Implement a GP that stores the Cholesky decomposition of $(K + \sigma_0^2 I)$. How can you compute $\kappa(x)^T (K + \sigma_0^2 I)^{-1} \kappa(x)$ using L? Tips for python: `scipy.linalg.cholesky`, `scipy.linalg.cho_solve`, `scipy.linalg.solve_triangular`.

2 Bayesian Optimization

Implement Bayesian optimization with the UCB acquisition function for both the Rastrigin as well as the function g from above.

a) Think about how to solve the acquisition function optimization problem.

b) Plot the points chosen by the algorithm and how the GP evolves.

c) Choose as an acquisition function the variance of the GP only. How does this compare to random sampling?

3 Symmetric quadratic forms

Assume $f(x) = x^T A x$ for $A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$. In what sense is it sufficient to consider symmetric A only?