Exercise on Bayesian Optimisation

Tinkle Chugh

Overview

This exercise consists of three tasks. In the first task, you will compare three different acquisition functions, lower confidence bound (LCB), expected improvement (EI) and probability of improvement (PoI) on a single-objective minimisation problem. In the second task, you will compare the closed-form expression of EI with the Monte Carlo approximation. In the third task, you will utilise Bayesian optimisation to optimise the parameters of a machine learning algorithm. You may want to go through the lecture slides for more details about acquisition functions and Bayesian optimisation.

Task 1 - Comparison of acquisition functions

The one-dimensional objective function $f(x) = \sin(3\pi x^3) - 0.5\sin(8\pi x^3)$, where $0 \leq x \leq 1$ is to be minimised. Given a training data set, $\mathbf{x} = [0.1, 0.5, 0.75, 0.2, 0.59]$, compare the performance of three acquisition functions: Expected improvement (EI), Probability of improvement (PoI) and Lower confidence bound (LCB). You can use the following libraries when implementing BO:

- Gaussian process regression: `https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html`
- Optimiser to optimise the acquisition function: `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html`

Plot the regret (difference between a minimum of objective function values and true optima) with a number of iterations for all three acquisition functions. The optimal value for the objective function is -1.3667.

Task 2 - Monte Carlo Estimation of Expected Improvement

In Task 1, you used a closed-form expression for EI. Instead of using a closed-form expression, estimate EI using Monte Carlo sampling. Compare two different methods of estimating EI.

Task 3 - Bayesian optimisation to estimate the parameters of a Random Forest Classifier

The objective of this task is to utilise BO to optimise the hyperparameters of a machine learning algorithm. Perform the following steps to utilise BO for tuning the parameters of a Random Forest Classifier.

- Load the Iris dataset and split it into training and testing sets (`https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html`).
- Define an objective function to train a Random Forest classifier with the given hyperparameters. For example, maximising accuracy in terms of minimising cross-validation error can be an objective function. You can select other objective functions, such as minimising root mean squared error.
- The parameters to be optimised are (`https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`):
  - Number of trees or estimators (n-estimators). Bounds: 10 - 200
  - Maximum depth of tree (max-depth). Bounds: 1-20
  - The minimum number of samples required to split an internal node (min-samples-split). Bounds: 2-10
  - The minimum number of samples required to be at a leaf node (min-samples-leaf). Bounds: 1-10

For BO, you can use the acquisition function of your choice. Plot the objective function value with the number of iterations.