Linear Algebra and Optimization for Machine Learning – Project 2

Attached to this assignment you can find a 'Heart' dataset containing data of 270 patients. The first 13 columns represents the patients' features such as age etc., and the last column has a value $+1$ if the patient has a heart disease and -1 if not. The subsequent features used are:

- age
- sex
- chest pain type (1: typical angina; 2: atypical angina; 3: non-anginal pain; 4: asymptomatic)
- resting blood pressure
- serum cholestoral in mg/dl
- if fasting blood sugar $>$ 120 mg/dl (1 = true; 0 = false)
- resting electrocardiographic results (0: normal; 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV); 2: showing probable or definite left ventricular hypertrophy by Estes' criteria)
- maximum heart rate achieved
- exercise induced angina (1 = yes; 0 = no)
- ST depression induced by exercise relative to rest
- the slope of the peak exercise ST segment (1: upsloping; 2: flat; 3: downsloping)
- number of major vessels (0-3) colored by flourosopy
- blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)

The goal of this assignment is to perform hyperparameter tuning of a Radial Basis Function (RBF)-kernelized SVM (support vector machine) using two methods:

- grid search
- Bayesian optimization (BO)

In this exercise, fitting an ML tool will be outsourced to a ready-made Python function SVC ("support vector classification") from the scikit-learn Python package. If used in combination with the RBF kernel, SVC has only two, strictly positive hyperparameters:

- C: the regularization parameter,
- γ ('gamma'): the kernel scaling parameter.

Your goal is to find values of C and 'gamma' such that the classification accuracy of an SVC tool obtained via five-fold cross-validation is maximized. Specifically, perform the following:

1. To ensure that we only use positive values of C and γ, we shall work with their decimal logarithms. For a dense meshgrid of value pairs of $(\log_{10} C, \log_{10} \gamma) \in [0, 9] \times [-10, 0]$ compute the cross-validation performance and illustrate them using 3D plots or heatmaps. Report on the best parameter combination you found.

2. Implement the BO algorithm to search for the best possible parameter combination yourselves – the 'black box function' is the cross-validation (in)accuracy. In particular, the following are to be obeyed:

   - Decide how to initialize the BO algorithm.
   - Choose an acquisition function to be used to guide the search for new $(\log C, \log \gamma)$ pairs.
   - Implement a minimization algorithm that, given a set of N evaluated points, uses the Bayesian optimization mechanism to find a (local) maximizer of the acquisition function which will be the $N + 1$-st $(\log C, \log \gamma)$ pair. For example, you can design your own descent method that, starting from a randomly selected point and, based on the acquisition function you chose, selects a direction to go to and for which a step size is determined through a line-search procedure, including a clearly-stated stopping criterion. You are free here to use any direction-determining method. In your method, account for the optimization domain $(\log_{10} C, \log_{10} \gamma) \in [0, 9] \times [-10, 0]$.

3. For your BO method, perform 100 iterations. Present your results using at least the following:

   - Plot the best-so-far misclassification error versus iteration number. Include in this plot a flat line corresponding to the best grid search obtained solution.
   - Make a scatterplot of the 100 points found using BO inside the optimization domain.
   - For four random BO iterations (the process of finiding a new maximizer of the acquisition function), plot the corresponding acquisition function along with the path that the optimization method takes to find the new best point.

   Comment on the relative performance of the two methods used discussing both the end result as well as the number of points evaluated, and the number of descent steps performed inside the BO acquisition function minimizer.

4. If your group consists of 4 people:

   - Include two different acquisition functions in your analysis.

The ideal situation (grade 10) is that you implement all the requested methods. If you are not successful, you may use an available package to perform the Bayesian optimization. However, this will lead to some point deductions. As a point of reference, an assignment which uses only a ready package, will receive the grade 5.

Leave clear comments in your Python code – unclear assignments and their codes will result in a reduction of the grade. Also, explain if something does not work or works too slowly; if you can explain well the problem and how this could be addressed, we might subtract less points, if something does not work as expected.

The page limit for the entire assignment is 6 pages of text excluding images (font size 11). A submitted assignment consists of a PDF file with the assignment and a zipped folder including .py files + a short readme.txt file explaining how to use the code.

The deadline is January 7, 2023, 23:59 CET. Please submit one project per group through Brightspace.

As mentioned in the lectures, we expect you to work in groups of 3–4; for groups of 4, please be sure to also work on the last part.