CMPE 462 Machine Learning
Spring 2020 Project II
Due: May 18 by 11.59pm

Project Description

In this project, you are going to implement SVM. For this purpose, a data set (data.mat) is given to you. You can load the mat dataset into Python using the function `loadmat` in `Scipy.io`. When you load the data, you will obtain a dictionary object, where 'X' stores the data matrix and 'Y' stores the labels. You can use the first 150 samples for training and the rest for testing. In this project, you will use the software package LIBSVM[1] to implement SVM. Note that LIBSVM has a Python interface[2], so you can call the SVM functions in Python.

Tasks

1. (30 points) Train a hard margin linear SVM and report both train and test classification accuracy.

2. (40 points) Train soft margin SVM for different values of the parameter C, and with different kernel functions. Systematically report your results. For instance, report the performances of different kernels for a fixed C, then report the performance for different C values for a fixed kernel, and so on.

3. (15 points) Please report how the number of support vectors changes as the value of C increases (while all other parameters remain the same). Discuss whether your observations match the theory.

4. (15 points) Please investigate the changes in the hyperplane when you remove one of the support vectors, vs., one data point that is not a support vector.

Bonus Task (10 points)

Use Python and CVXOPT (`http://cvxopt.org`) QP solver to implement the hard margin SVM.

Note: The notations used in the CVXOPT tutorial (`http://cvxopt.org/userguide/coneprog.html#quadratic-programming`) is different from what we learned in class. Specifically, we used (Q, $\mathbf{p}$, A, $\mathbf{c}$) in class and the corresponding notations in CVXOPT are (P, $\mathbf{q}$, G, $\mathbf{h}$). Verify your results using the toy data in class.

[1] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[2] `https://github.com/cjlin1/libsvm/tree/master/python`

Submission Guidelines

- Download the provided dataset.

- Prepare a comprehensive report in pdf format. Include your tables, and comments for each task. Follow academic writing rules. Prefer a concise and clear language. You do not need to include code snippets in your report.

- Submit the completed .ipynb and .pdf files through the assignment *Project 2* on Moodle. Name your submission files with the student IDs of the group members (i.e. 2015400XXX_2015400YYY.ipynb). Please submit your pdf report through Turnitin assignment.

- The submitted .ipynb file should run from scratch without errors when accompanied with the provided data. Note that iPython kernel of Jupyter stores variables in the kernel over time and if your code depends on any such variable, it will not run on new sessions. Therefore, make sure that you can run your code from scratch (restart kernel & run all) before submitting the notebook. If your code does not run for any reason, you will be deducted 10 pts.

- Please fairly share the tasks among the group members. Write the student ID/IDs next to each task/code cell to specify who completed which task. If you do not specify any IDs, the instructor will assume each student put equal effort in that task.