Final Projects for MATH320, Fall 2023

1 Project 1: Support Vector Machine for binary and multiclass classification

In this project we will develop a support vector machine for classification with various kernels. We will work on the multiclass classification of the types of *E. coli* (`https://en.wikipedia.org/wiki/Escherichia_coli`). The *E. coli* data file can be downloaded on the class Canvas page.

You may build upon your own SVM code of from your Homework V for binary classification, and then develop approaches to extend the code for multiclass classification. We did not implement the kernels in Homework V, though. To further improve the accuracy of classification you will implement and compare various kernels in your own SVM code. Your implementations will be compared to the SVM in sklearn (`https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`) which also allows you to use various type of kernels. You can then compare results with and without kernels, for both binary and multiclass classifications.

You are expected to

(a) Finish up your Python SVM code for multiclass classification without kernel.

(b) Implement kernels in your own SVM code.

(c) Implement SVM in sklearn.

(d) Understand the functinality of various kernels for SVM.

(e) Compare your own implementation of SVM and that in sklearn, with and without kernels, for binary and multiclass classifications.

(f) Apply the methods you develop above to the classification problem. Quantify the accuracy of your classifications.

(g) Wrap up the results and finish a project report with five or more pages (excluding your code)

2 Project 2: Classifications using Neural Network

In this project we will implement neural network for classification. We will use the same dataset as Project 1 about the the types of *E. coli* (`https://en.wikipedia.org/wiki/Escherichia_coli`). The *E. coli* data file can be downloaded on the class Canvas page.

You are not required to build your own neural network and its training. Instead, you may use the Multilayer Perception Classifier in sklearn. More details information about the classification function can be found at `https://scikit-learn.org/stable/modules/neural_networks_supervised.html` and `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html`. Make sure you understand the various parameters and options of this function, and are able to provide proper values. These include

(1) Depth and width of the network.

(2) Solver of the minimization probelm.

(3) Strength of regularization.

(4) Batch size.

(5) Initial and adaptive earning rate.

(6) Initilization of the network.

(7) Momentum method.

(8) Termination of the training.

(9) Use of well-trained network for prediction on test dataset.

You will need to split the dataset into a training set and a test set so you can verify the useage of your network.

You are expected to

(a) Develop a new Python code to implement `klearn.neural_network.MLPClassifier` of sklearn.

(b) Read and understand the ecoli dataset.

(c) Apply your code for the classification of *E. coli*.

(d) Describe the training process. Quantify the accuracy of your classifications.

(e) Wrap up the results and finish a project report with five or more pages (excluding your code)