CSCI4390-6390 Assign6

Assign6: Support Vector Machines

Due Date: Nov 9, before midnight (11:59:59PM, Alofi Time; GMT-11)

You will use the Appliances energy prediction data set. You should ignore the first attribute, which is a date-time variable, and you should also remove the last attribute, which is a duplicate of the previous one. Use the first attribute (after removing the date-time variable), which denotes the **Appliances Energy Use**, as the response variable, with the remaining attributes as predictor variables.

Note that the **Appliances Energy Use** attribute takes values in the range $[10, 1080]$. However, for binary classification, we need only two values, so for the purpose of this assignment you should consider energy use less than or equal to 50 as the positive class (1), and energy use higher than 50 as negative class (-1).

You should shuffle the data points before selecting 70% of the data training and 30% for testing, so that there is an equal mix of the classes in both.

You will implement the dual SVM Algorithm 21.1 (Chapter 21, page 540). You must implement both the "hinge" and "quadratic" loss cases, which will be command line options.

CSCI4390

You must implement two kernels, namely, both linear and Gaussian.

CSCI6390

You must implement three kernels, namely, linear, Gaussian, and polynomial.

What to submit

- Write a script named as **Assign6.py**, which will be run as

Assign6.py FILENAME LOSS C EPS MAXITER KERNEL KERNEL_PARAM

FILENAME is the datafile name, LOSS is either the string "hinge" or "quadratic", C is the regularization constant, EPS is the convergence threshold, MAXITER is the max number of iterations to perform (in case you do not get convergence within EPS), KERNEL is one of the strings "linear", "gaussian" or "polynomial", and finally KERNEL_PARAM is either a float that represents the spread σ^2 for gaussian kernel (see Eq 5.10 on pg 147), or it is a comma separated pair q, c for the polynomial kernel (see Eq 5.9 on pg 144), with q being the degree (an int) and c the kernel constant (a float); note that polynomial kernel is only for CSCI6390.

You should implement the algorithms using NumPy; you cannot use any other library. Note that computing the full kernel matrix for 19K+ points will be memory intensive, so if you do not have enough memory, one option is for you to repeatedly compute the required kernel values for each point $\mathbf{x}_k$. You can see in line 12, that we need only the k-th row of the kernel matrix, so you can recompute that row as needed instead of storing the entire kernel matrix in memory. Alternatively, you can show results on at least 5000 points. These points should be selected after shuffling the data, and then you should create the training/test splits.

One final note in the implementation is that you must shuffle the points/indexes in line 11 so that you get different permutations for the stochastic gradient ascent in each iteration. This results in better performance than using the fixed order.

Your script should print out the number of support vectors on the training set (those points that are exact support vectors, with $0 < \alpha < C$ for hinge, and with $\alpha > 0$ for quadratic loss), and the final accuracy value on the test data, where you use 70% of the data for training and 30% for testing. For the linear kernel you must also print the weight vector and bias (see Eq 21.34 and 21.35).

Note that accuracy is defined as the fraction of correct class label predictions. So for each test point, you should predict its class as the sign of the hyperplane equation (see Eq 21.37). Divide the number of correct predictions by the test data size to get the accuracy. Report the results for the combination of LOSS, C, KERNEL and KERNEL_PARAM values that yields the best accuracy.

You can compare your results with those obtained from sklearn SVM implementation.

- Submit a PDF file named Assign6.pdf that should include your answers to each of the questions (just cut and paste the output from python). **Failure the submit the PDF will result in lost points.**

- Submit the scripts and pdf file via submitty

Policy on Academic Honesty

You are free to discuss how to tackle the assignment, but all coding must be your own. Please do not copy or modify code from anyone else, including code on the web. Any students caught violating the academic honesty principle will get an automatic F grade on the course and will be referred to the dean of students for disciplinary action.