Homework 3

DSE 220: Machine Learning

Due Date: May 14 11:59pm

1 Turn-in Instructions

The answers should be submitted on Gradescope. You should submit the PDF of the Jupyter Notebook. Explain your approach as clearly as possible whereever needed. Please make sure that the questions are clearly segmented and labeled. To secure full marks for a question both the answer and the code should be correct. Completely wrong (or missing) code with correct answer will result in zero marks. Please complete this homework individually.

2 Logistic Regression

Question 1: This question was included in the previous homework and no submission is needed.

3 Perceptron & Support Vector Machines

3.1 Data

In this section, we will work with text data. Download the newsgroups train and test data using fetch 20newsgroups for categories: ’alt.atheism’, ’comp.graphics’, ’sci.space’ and ’talk.politics.mideast’ after removing ’headers’, ’footers’ and ’quotes’ from the data.

```
# function to get data
corpus = fetch_20newsgroups(subset=<VALUE HERE>,
                            remove=<VALUE HERE>,
                            categories=<VALUE HERE>)
```

Next, we need to vectorize the *documents* to train our classifier. Use the TfidfVectorizer to get vectors of the documents (after smoothing[1]). A common practice is to convert all the documents to lowercase and remove stopwords like a, and, the etc. Use the stopwords set provided by *’nltk.corpus.stopwords’*. Take advantage of the parameters provided by the *TfidfVectorizer* to convert to lowercase and remove stopwords. (10 marks)

Note: Fit the *TfidfVectorizer* only on the train data and re-use the same on the test data. Do not fit on the test data again.
Note: You might have to run *’nltk.download(’stopwords’)’* before using nltk’s stopwords.
[1] Smoothing the next data is the same as computing the idf values after adding a document with all the words in the vocabulary.

3.2 Learning

Question 2: After obtaining the tf-idf vectors for train and test data, use the perceptron model (no penalty) to train on the training vectors and compute the accuracy on the test vectors. (5 marks)

Question 3: Keeping all the above data processing steps same, observe how the test accuracy changes by varying the number of top features selected for 100, 200, 500, 1000, 1500, 2000, 4000, 10000, 20000, 30000 for a perceptron model. Report and plot the results. Provide a brief explanation of the observed results. (10 mark)

Question 4: After obtaining the tf-idf vectors for train and test data, use the SVM model to train on the training vectors and compute the accuracy on the test vectors. Use linear kernel and default parameters. (5 mark)

Question 5: Keeping all the above data processing steps same observe how the test accuracy changes by varying the number of top features selected for 100, 200, 500, 1000, 1500, 2000, 4000, 10000, 20000, 30000 for a linear SVM model. Report and plot the results. Provide a brief explanation of the observed results. (10 mark)

Question 6: Perform 80-20 split of the training data to obtain validation data using train_test_split (random state=10). Use this validation data to tune the regularization parameter ’C’ for values 0.01,0.1,1,10,100. Select the best ’C’ and compute the accuracy for the test data. Report the validation and test accura-

cies. *Use feature vectors of 2000 dimensions.*(10 marks)
Note: Retrain on train + validation data while reporting accuracy on test data

Question 7: Use the same train and validation split as the previous question. Train a kernelized SVM (with 'C'=10000) with kernel values - 'poly' with degree 1, 2, 3, 'rbf ' and 'sigmoid', and report the one with best accuracy on validation data. Report the test accuracy for the selected kernel. (10 marks)

3.3 Custom Kernels

Now we introduce the concept of custom kernels in Support Vector Machines. In class we discussed how kernel functions are a form of similarity measure for our data. There are good chances that we need some other form of similarity measure for our data which works better with the SVM classifier.

Question 8: Use *Cosine Similarity* and *Laplacian Kernel* ($exp^{-||x-y||_1}$) measures, and report the test accuracies using these kernels with SVM. (15 marks)

Question 9: Another way to construct a kernel is use a linear combination of 2 kernels. Let K be a kernel represented as:

$$K(x, y) = \alpha K_1(x, y) + (1 - \alpha)K_2(x, y) \quad (0 \leq \alpha \leq 1)$$

Provide a brief explanation of why K is a valid kernel. Does your reasoning hold true for other values of α as well? Let K_1 be the *Cosine Similarity* and K_2 be the *Laplacian Kernel.* Using K as kernel, train a SVM model to tune the value of α (upto one decimal) and report the accuracy on the test data using the selected parameter. (15 marks)