Homework 3

DSE 220: Machine Learning

Due Date: 14 May 2017

1 Instructions

The answers to the questions should be submitted on Gradescope and the code should be submitted on github by 14 May 2017. You don't need to explain your approach so please be concise in your Gradescope submission. To secure full marks for a question both the answer (on Gradescope) and the code (on github) should be correct. Completely wrong (or missing) code with correct answer will result in zero marks. Please complete this homework individually.

2 Discriminative Learning

For the questions in this section, load the wine dataset (wine_original.csv).

Question 1: Perform a 80-20 split using train_test_split on the data to obtain the train and the test data (random_state=3). Use Logistic Regression to classify the wines according to their cultivators. Tune parameters *'penalty'* and *'C'* using *GridSearchCV* implementation. Report the accuracy on test data. (10 marks)

3 Perceptron and Support Vector Machines

3.1 Data:

In this section, we will work on the text data. Download the newsgroups data (train and test) using fetch_20newsgroups for categories: 'alt.atheism', 'comp.graphics', 'sci.space' and 'talk.politics.mideast' after removing 'headers', 'footers' and 'quotes' from the data. Convert all the words in the text to lower case. A common practice is to remove the stopwords like a, and, the etc. from the text. Use nltk to get the stopwords list (nltk.corpus.stopwords) and remove the stopwords from the text. Use TfidfVectorizer to obtain the tfidf vectors (after smoothing*) for the train and test data and select only top 2000 features

(words). You can also perform the above stated actions (lowercase and stop-words) using the TfidfVectorizer. *Note: You'll fit the tf-idf vectors on the train data and use the same to transform the test data.* (10 marks)
*: Smoothing the text data is same as computing the idf values after adding a document with all words in the vocabulary.

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, 3000 for a perceptron model. Report and plot the 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, 3000 for a linear SVM model. Report and plot the 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 cost parameter 'C' for values 0.01,0.1,1,10,100. Select the best value compute the accuracy for the test data. Report the validation and test accuracies. *Note: Use full data of 2000 vectors here.* (10 marks)

Question 7: 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. Also report the test accuracy for the selected kernel. (10 marks)

3.2 Custom Kernels

Now we introduce the concept of custom kernels in Support Vector Machines. There are good chances that we need some other form of similarity measure for our data, for which we need to pass our own function as kernel to SVM.

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:

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

Why is K 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 '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)