COL 774: Assignment 2

Due Date: 11:50 pm, March 8 (Thursday), 2018. Total Points: 26 + x

Notes:

- This assignment has two parts - Text Classification using Naïve Bayes and Handwritten digit classification using SVM.
- You should submit all your code (including any pre-processing scripts written by you) and any graphs that you might plot.
- Do not submit the datasets. Do not submit any code that we have provided to you for processing.
- Include a **single write-up (pdf) file** which includes a brief description for each question explaining what you did. Include any observations and/or plots required by the question in this single write-up file.
- You should use Python/MATLAB for all your programming solutions.
- Your code should have appropriate documentation for readability.
- You will be graded based on what you have submitted as well as your ability to explain your code.
- Refer to the course website for assignment submission instructions.
- This assignment is supposed to be done individually. You should carry out all the implementation by yourself.
- We plan to run Moss on the submissions. We will also include submissions from previous years since some of the questions may be repeated. Any cheating will result in a zero on the assignment, a penalty of -10 points and possibly much stricter penalties (including a **fail grade** and/or a **DISCO**).

1. **(26 points) Text Classification**
   In this problem, we will use the Naïve Bayes algorithm for text classification. The dataset for this problem is a subset of the IMDB movie review dataset and has been obtained from this website (look at the IMDB movie review dataset). Given a movie review, task is to predict the rating given by the reviewer. Read the website for more details about the dataset. You have been provided with separate training and test files containing 25,000 reviews (samples) each. Data is available at this link. A review comes from one of the eight categories (class label). Here, class label represents rating given by the user along with the review. You are provided four files i) Train text ii) Train labels iii) Test text iv) Test labels. Text files contain one review in each line and label files contain the corresponding rating.

   (a) **(10 points)** Implement the Naïve Bayes algorithm to classify each of the articles into one of the given categories. Report the accuracy over the training as well as the test set. In the remaining parts below, we will only worry about test accuracy.
   Notes:

   - Make sure to use the Laplace smoothing for Naïve Bayes (as discussed in class) to avoid any zero probabilities. Use $c = 1$.
   - You should implement your algorithm using logarithms to avoid underflow issues.
   - You should implement Naïve Bayes from the first principles and not use any existing Matlab/Python modules.

(b) **(2 points)** What is the test set accuracy that you would obtain by randomly guessing one of the categories as the target class for each of the articles (random prediction). What accuracy would you obtain if you simply predicted the class which occurs most of the times in the training data (majority prediction)? How much improvement does your algorithm give over the random/majority baseline?

(c) **(4 points)** Read about the confusion matrix. Draw the confusion matrix for your results in the part (a) above (for the test data only). Which category has the highest value of the diagonal entry? What does that mean? What other observations can you draw from the confusion matrix? Include the confusion matrix in your submission and explain your observations.

(d) **(4 points)** The dataset provided to is in the raw format i.e., it has all the words appearing in the original set of articles. This includes words such as 'of', 'the', 'and' etc. (called stopwords). Presumably, these words may not be relevant for classification. In fact, their presence can sometimes hurt the performance of the classifier by introducing noise in the data. Similarly, the raw data treats different forms of the same word separately, e.g., 'eating' and 'eat' would be treated as separate words. Merging such variations into a single word is called stemming.

- Read about stopword removal and stemming (for text classification) online.
- Use the script provided at this link to you to perform stemming and remove the stopwords in the training as well as the test data.
- Learn a new model on the transformed data. Again, report the accuracy.
- How does your accuracy change over test set? Comment on your observations.

(e) **(6 points)** Feature engineering is an essential component of Machine Learning. It refers to the process of manipulating existing features/constructing new features in order to help improve the overall accuracy on the prediction task. For example, instead of using each word as a feature, you may treat bi-grams (two consecutive words) as a feature. Come up with at least two alternative features and learn a new model based on those features. Add them on top of your model obtained in part (d) above. Compare with the test set accuracy that you obtained in parts (a) and parts (d). Which features help you improve the overall accuracy? Comment on your observations.

2. **(Coming soon)(x points) MNIST Handwritten digit Classification**
In this problem, we will use Support Vector Machines (SVMs) to build a handwritten digit classifier. We will be solving the SVM optimization problem using the Pegasos algorithm and also use a customized solver known as LIBSVM. You will be provided with separate training and test example files along with the corresponding label files. Each row in the (train/test) data file corresponds to an image of size 28x28, represented as a vector of pixel intensities and its label. Every column represents a feature where the feature value denotes the grayscale value of the corresponding pixel in the image (there is a feature for every pixel). Last entry in each row gives the corresponding label. You will be provided with a subset of the dataset available at this link. For details of the original dataset, you are encouraged to look at this webpage. More details about the specific tasks will follow soon.