COL 774: Assignment 2

Due Date: 11:50 pm, Friday Mar 10, 2017. Total Points: 58

Notes:

- This assignment will have a mix of theoretical as well as implementation questions.
- Only the implementation questions will be graded.
- You are strongly encouraged to try out theoretical questions though they are not graded.
- You should submit all your code (including any pre-processing scripts written by you) and any graphs that you might plot. Do not submit answers to theoretical questions.
- 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 MATLAB/Python 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**).
- Question 2 below has been adapted from a homework assigment question in a course offered at CMU by Carlos Guestrin.

1. **(22 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 Reuters-21578 dataset and has been obtained from this website (look at the R8 dataset). Read the website for more details about the dataset. You have been provided with separate training and test files containing 5485 and 2189 articles (examples), respectively. Each article comes from one of the eight categories (class label). Each row in the file contains the information about the class of the article followed by the list of words appearing in the article.

   (a) **(10 points)** Implement the Naïve Bayes algorithm to classify each of the articles into one of the given categories. Report the accuracies 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 in the confusion matrix? What does that mean? Which two categories are confused the most with each other i.e. which is the highest entry amongst the non-diagonal entries in the confusion matrix? Explain your observations. Include the confusion matrix in your submission.

(d) **(6 points)** The dataset provided to is in the raw format i.e., it has all the words appearning in the original set of articles are present. This includes words such as 'of', 'the', 'and' etc. (called stopwords). Presumably, these words are not relevant for classification. In fact, their presence can sometimes hurt the peroformance of the classifier by introducing noise in the data. Similarly, the raw data treats separately different forms of the same word 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 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 as well as the confusion matrix over the test data.
- How do your accuracies and confusion matrix change? Comment on your observations.

2. **(36 points) Facial Attractiveness Classification**
In this problem, we will use Support Vector Machines (SVMs) to build a facial attractiveness classifier. We will be solving the SVM optimization problem using a general purpose convex optimization package as well as using a customized solver known as LIBSVM. You are 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. 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). Each row in each of the label files gives the corresponding label (1:*attractive*,2: *not attractive*). You have been provided with a processed version of a subset of the dataset. For details of the original dataset, you are encouraged to look at the course project report referred here[1].

(a) **(8 points)** Download and install the CVX package. Express the SVM dual problem (with a linear kernel) in the a form that the CVX package can take. You will have to think about how to express the SVM dual objective in the form $\alpha^T Q\alpha + b^T\alpha + c$ matrix where Q is an $m \times m$ matrix (m being the number of training examples), b is an m-sized column vector and c is a constant. For your optimization problem, remember to use the constraints on α_i's in the dual. Use the SVM formulation which can handle noise and use $C = 500$ (i.e. C in the expression $\frac{1}{2}w^T w + C * \sum_i \xi_i$). Report the set of support vectors obtained from your optimization.

(b) **(6 points)** Calculate the weight vector w and the intercept term b using the solution in the part above. Classify each of the examples in the test file into one of the two labels. Report the average test set accuracy obtained.

(c) **(6 points)** Now solve the dual SVM problem using a Gaussian kernel with the bandwidth parameter $\gamma = 2.5$ (i.e. γ in $K(x, z) = \exp^{-\gamma * \|x-z\|^2}$). Think about how the Q matrix will be represented. What are the set of support vectors in this case? Note that you may not be able to explicitly store the weight vector (w) or the intercept term (b) in this case. Use your learned model to classify the test examples and report the accuracies obtained. How do these compare with the ones obtained with the linear SVM?

(d) **(10 points)** Now train an SVM on this dataset using the LIBSVM library, available for download from this link. Repeat the parts above using a linear Kernel as well as a Gaussian kernel with $\gamma = 2.5$.

[1]Ryan White, Ashley Eden, Michael Maire "Automatic Prediction of Human Attractiveness", CS 280 class report, December 2003.

Use $C = 500$ in both cases, as before. Report the set of support vectors obtained as well as the test set accuracies for both linear as well as the Gaussian kernel setting. How do these compare with the numbers obtained using the CVX package. Comment.

(e) **(6 points)** Cross-validation is a technique to estimate the best value of the model parameters (e.g., C, γ in our problem) by randomly dividing the data into multiple folds, training on all the folds but one, validating on the remaining fold, repeating this procedure so that every fold gets a chance to be the validation set and finally computing the average validation accuracy across all the folds. This process is repeated for a range of model parameter values and the paramters which give best average validation accruacy are reporeted as the best parameters. For a detailed introduction, you can watch this video. Use LIBSVM for this part, and you are free to use the cross-validation utility provided with the package.

In this problem, we will perform 10 fold cross-validation to estimate the value of the C parameter. We will fix γ to be 2.5. Vary the value of C in the range $\{1, 10, 10^2, 10^3, 10^4, 10^5, 10^6\}$ and compute the 10-fold cross-validation accuracy for each value of C. Also, compute the corresponding accuracy on the test set. Now, plot both the average validation set accuracy as well as the test set accuracy on a graph as you vary the value of C on x-axis (use log scale on x-axis). What do you observe? Which value of C gives the best validation accuracy? Does this value of the C also give the best test set accuracy? Comment on your observations.

(f) **(Extra fun! No Credits)**. You may argue that facial attractiveness is a subjective concept. In your test data, identify the top 3 images (each) with the highest confidence of being *attractive* as well as being *not attractive* based on your model. Think about how would you identify such images using your learned model parameters. Now, display these images using the matlab script provided with the dataset repository. You can also use any other utility in Python in case you are working with Python code. What do you observe? Does your idea of attractiveness align with what the model thinks is attractive. Why or why not?

Note: Do not submit the CVX or LibSVM code. You should only submit the code that you wrote by youself (including wrapper code, if any) to solve this problem.