COL 774: Assignment 2 - Part B

Due Date: 11:50 pm, Mar 6 (Friday), 2019. Total Points: 36 (for Part B)

Notes:

- This is the second part of Assignment 2 - Article 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 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**).

(36 points) Fashion MNIST Article Classification

In this problem, we will use Support Vector Machines (SVMs) to build a article classifier, to classify whether an image contains a sneaker, shirt, bag, etc. We will be solving the SVM optimization problem using a general purpose convex optimization package as well as using a customized solver from sklearn. The dataset that we would be using can be downloaded from the course website. Note that we have given you a subset of the original Fashion MNIST Dataset. You can read more about the original dataset from this link. Every column except the last represents a feature where the feature value denotes the grayscale value (0-255) of the corresponding pixel in the image. The last column gives the corresponding label. Since the features represent grayscale values, we may not want to normalize the data to zero mean and unit variance as described in the class. But for this problem, you may find it helpful to simply scale all the values to the range [0,1] (down from [0 255]).

1. **Binary Classification:**
   Let d be the last digit of your entry number. Then, we would start with binary classification problem over the images of article classes (d vs $(d+1)$ mod 10) in this Section. In particular, you should take the subset of images for the article classes d and $(d+1)$ mod 10 from the train/validation/test data provided to you and perform the following experiments [1]. The article classes and their corresponding labels can be found here.

[1] Note that different students will be experimenting with different class pairs depending on their entry number

(a) **(8 points)** Download and install the CVXOPT package. Express the SVM dual problem (with a linear kernel) in the a form that the CVXOPT package can take. You will have to think about how to express the SVM dual objective in the form $\alpha^T P\alpha + q^T\alpha + c$ matrix where P is an $m \times m$ matrix (m being the number of training examples), q 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 $C = 1.0$ (i.e. C in the expression $\frac{1}{2}w^T w + C * \sum_i \xi_i$). You can refer this link to get a working overview of cvxopt module and it's formulation. Obtain the set of support vectors from your optimization for the binary classification problem. Furthermore, calculate the weight vector w and the intercept term b and classify each of the examples in the validation and test file into one of the two labels. Report the validation and test set accuracy obtained. You will need to carefully think about how to represent w and b in this case.

(b) **(6 points)** Use CVXOPT package to solve the dual SVM problem using a Gaussian kernel. Think about how the P 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 validation and test examples and report the accuracies obtained. Use $C = 1.0$ and $\gamma = 0.05$ (i.e. γ in $K(x, z) = \exp^{-\gamma * \|x-z\|^2}$) for this part. How do these compare with the ones obtained with in the linear kernel?

(c) **(6 points)** Repeat part-a & b with Scikit SVM package available from this link. Report accuracy on validation and test set for both linear and Gaussian kernel. Furthermore, compare weight (w), bias (b) and nSV (# of Support Vectors) with your implementation in part (a) for linear kernel and part (b) for Gaussian kernel. Also compare the computational cost (training time) of the two implementations. Report your observations.

2. **Multi-Class Classification:**
In this section, we will work with the entire subset of the data provided to you focusing on a multi-class classification problem. We will work with a Gaussian kernel for this section.

(a) **(4 points)** In class, we described the SVM formulation for a binary classification problem. In order to extend this to the multi-class setting, we train a model on each pair of classes to get $\binom{k}{2}$ classifiers, k being the number of classes. During prediction time, we output the class which has the maximum number of votes from all the $\binom{k}{2}$ classifiers. You can read more about one-vs-one classifier setting at the following link. Using your solver from previous section, implement one-vs-one multi-class SVM. Use a Gaussian Kernel with $C = 1.0$ and $\gamma = 0.05$. Classify given dataset and report validation and test set accuracy. In case of ties, choose the label with the highest score.

(b) **(4 points)** Now train a multi-class SVM on this dataset using the Scikit SVM library . Repeat part (a) using a a Gaussian kernel with $\gamma = 0.05$. Use $C = 1.0$ as earlier. Report the validation as well as test set accuracies. How do your results compare with those obtained in part (a) above. As earlier, compare the computational cost (training time) of the two implementations? Comment.

(c) **(4 points)** Draw the confusion matrix as done in the first Part (Naive Bayes) of this assignment for both of the above parts (2(a) and 2(b)). What do you observe? Which articles are mis-classified into which ones most often? Do the results make sense?

(d) **(4 points)** Validation set is typically used to estimate the best value of the model parameters (e.g., C in our problem with linear kernel) by randomly selecting small subset of data as validation set, training on train set (minus the validation set) and making predictions on validation set. For a detailed introduction, you can refer to this video. You can check the correctness of your intuition by trying this test. K-fold cross validation is another such techniques in this regard that we use in practice. In this technique we divide our training data into K-folds or parts and then treat each part as our validation set once and train on the remaining K-1 parts. You can read more about cross validation here [2] (see Section 1). for more details. This process is repeated for a range of model parameter values and the parameters which give best K-fold cross validation accuracy are reported as the best parameters. We will use Scikit for this part.

For this problem, we will do a 5-fold cross validation to estimate the best value of the C parameter for the Gaussian kernel case. Test data should not be touched. Fix γ as 0.05 and vary the value of C in the set $\{10^{-5}, 10^{-3}, 1, 5, 10\}$ and compute the 5-fold cross validation accuracy for each value

[2] These are from Andrew Ng notes posted on the course website, and the link is available only from the internal IIT Delhi network. Feel free to read additional material online about this topic

of C. Also, compute the corresponding accuracy on the test set. Now, plot both the 5-fold cross validation accuracy as well as the test set accuracy on a graph as you vary the value of C on x-axis (you may use log scale on x-axis). What do you observe? Which value of C gives the best 5-fold cross validation accuracy? Does this value of the C also give the best test set accuracy? Comment on your observations.

3. **Extra Fun - no credits!**

   (a) Use mini-batch version of Pegasos algorithm described in "Pegasos: Primal Estimated sub-GrAdient SOlver for SVM" to optimize SVM problem and solve for w, b. You should start with Algorithm 1 given in the paper. Further, Algorithm 1 ignores the intercept term b. To incorporate b, use the description provided in Section 6 of the paper (use Equation 23). Your final implementation should compute the values of both w and b. Note that the paper uses a slightly different notation from the one covered in class and you should map the symbols accordingly.

   (b) Multi-class problem can also be tackled using one-vs-rest strategy. Compare one-vs-rest and one-vs-one in terms of training time and accuracy. You can read more about Multiclass SVMs in Sec. 7.1.3 of PRML book [by Christopher Bishop] or on the the link given earlier.