CSE/ECE 343/543: Machine Learning
Assignment-1 Programming Questions
Max Marks: 100 (Programming:70, Theory:30) **Due Date**: 25/09/2019, 11:59PM

Instructions

- Try to attempt all questions.
- Keep collaborations at high level discussions. Copying/Plagiarism will be dealt with strictly.
- Start early, solve the problems yourself. Some of these questions may be asked in Quiz/Exams.
- Late submission penalty: As per course policy.
- Your submission would be a single .zip file (rollno_HW1_programming.zip) file, that would contain two items (codes + .pdf file) . You have to include all your plots, results, analysis, conclusion in the pdf file.

PROGRAMMING QUESTIONS

1. (35 points) **Linear Regression.**
   - (15 points) : Implement Linear Regression for the Abalone Dataset. The dataset contains 9 variables out of which the last column is the output variable and the other 8 are input attributes. You need to implement gradient descent from scratch i.e. you cannot use any libraries for training the model (You may use numpy, but libraries like sklearn are not allowed). Choose an appropriate learning rate. You may need feature normalisation.
     (a) Include plots for the root mean squared error (RMSE) vs. gradient descent iterations for both training as well as validation set (one for each set computed over the 5 folds – 2 plots in total). The curves should show the mean RMSE.
     (b) Also implement the normal equation (closed form) for linear regression and get the optimal parameters directly. Compute the RMSE (after getting the optimal parameters) for both training and validation set and report them for all the 5 folds.
     (c) Compare the final RMSE obtained from (a) after convergence and the RMSE from (b) and make a note of any observations you might have.
   - (15 points) : **Regularization**
     From the previous part, identify the validation set that has the lowest RMSE, and hold it out as the test set. Use the remaining 80% of the data as the new training + validation set. **Note that your model should never see the test set (neither for training, nor for validation)**.

Use 5-fold cross validation with grid search on the train + val set (without using the test set) to find the appropriate regularization parameter (hyperparameter). You may use Ridge, Lasso and GridSearchCV routines from the sklearn library to perform the following:

(a) Find the optimal regularization parameter for L_2

(b) Find the optimal regularization parameter for L_1

Write the above two regularization parameters in the report.

Once the optimal regularization parameters for both L_2 and L_1 have been found out, modify the gradient descent algorithm implemented in (1) to accommodate for the L_2 and the L_1 regularization term.

Use the values of the "optimal" regularization parameter found out in (a) and (b) and use it with the modified gradient descent algorithm and plot the RMSE error vs iterations curve. Also report the RMSE on the test set. [Two plots – one for L_1, one for L_2]

- (5 points) : **Best Fit Line**

  Use the following data that contains only 1 input variable and 1 output variable i.e the brain-weight to the body-weight proportion for varying species. Consider the dataset as a whole i.e. do not split it into train, val or test. Perform the following tasks:

  (a) Plot the data points using a scatter plot along with the best fit line found out using linear regression (without regularisation).

  (b) Use L2 regularisation and plot the data points and the new best fit line.

  (c) Use L1 regularisation and plot the data points and the new best fit line.

  Use the gradient descent algorithm implemented in (1) to perform regression on the whole dataset in all these 3 parts.

  Compare how the best fit line changes visually with adding different kinds of regularisation, was it a better fit, worse fit than the regression performed without regularisation?

2. (35 points) **Logistic Regression.**

   i (15 points) Download the dataset from here. It's a standard dataset, where you have to predict whether the income of a person is above or below 50k$. Implement a classifier using logistic regression **from scratch** (you are allowed to use only NumPy and Pandas). Also implement L1 and L2 regularization. Report the accuracy on the train, val and test set while using L1 and L2 regularization separately. You may use scikit-learn only for encoding categorical data. You can play around with the data - it may or may not be necessary to use all the features. Can you come up with a reason as to why L1 regularization works better than L2 regularization, or vice versa? Also plot an error vs iteration and accuracy vs iteration graphs for both models (L1 and L2).

ii (15 points) Download the MNIST dataset from here. Implement an L1 and L2 regularized logistic regression model using the scikit-learn library. Compute and report the accuracy obtained using one-vs-rest approach for each of the 10 classes, for both the training and test sets. See this tutorial to understand how the one-vs-rest approach works while using a binary classifier for a multi-class classification problem. Report the train and test accuracy for both L1 and L2 regularized logistic regression. Comment on whether or not it is a good fit, i.e, underfitting or overfitting.

iii (5 points) For the MNIST dataset, plot the Receiver Operating Curve (ROC) curve for each class (Do this just for L2 regularized Regression). Plot all ROC curves on the same graph. A tutorial on ROC curves is given here and here.