CSE/ECE 343/543: Machine Learning
Assignment-1 Linear Regression & Logistic Regression
Max Marks: 100 (Programming:80, Theory:20) **Due Date**: 19/9/2020, 11:59 PM

Instructions

- Keep collaborations at high level discussions. Copying/Plagiarism will be dealt with strictly.
- Late submission penalty: As per course policy.
- Your submission should be a single zip file **2018xxx_HW1.zip** (Where *2018xxx* is your roll number). Including only the **relevant files** arranged with proper names. A single **.pdf report** explaining your codes with relevant graphs and visualization and solution to theory questions. The structure of submission should follow:

  2018xxx_HW1
  |– scratch.py download
  |– test.py download
  |– (All other files for submission)
- Remember to **turn in** after uploading on Google Classroom. No excuses or issues would be taken regarding this after the deadline.
- Start the assignment early. Resolve all your doubts from TAs in their office hours **two days before the deadline.**
- **Document** your code. Lack of comments and documentation would result in loss of 20% of the *obtained* score.

Template for Programming Questions

- You are provided with two files, **scratch.py** and **test.py**
  - *scratch.py* has two classes, **MyLinearRegression** and **MyLogisticRegression**.
    - MyLinearRegression: implement your algorithm for Linear Regression in this class
    - MyLogisticRegression: Implement your algorithm for Logistic Regression in this class.
  - *test.py* is an example file and the final test file used during the demo will follow a similar structure. Ensure that *scratch.py* works fine with this file.
- Functions mandatory to implement have been provided with a docstring to explain their functionality.
- You are free to implement other functions as required.

- The code written using sklearn's algorithms should be present in another file. You can see a similarity in the function definition of sklearn's algorithms and your algorithms which will help you in implementation.

---

1. (45 points) **Linear Regression**

   You need to implement gradient descent from scratch (you may use Numpy, but libraries like sklearn are not allowed).

   Download the datasets, **Dataset 1**, **Dataset 2** (README)

   Perform Linear Regression on both the datasets. Also perform K-Fold cross-validation (implemented from scratch) in this exercise.

   - Choose an appropriate value of K and justify it in your report along with the preprocessing strategy. (5 points)
   - Implement gradient descent using two losses - RMSE loss and MAE loss (from scratch). (5 points)

   Analysis to be included in your report:

   (a) Include plots between training loss v/s iterations and validation loss v/s iterations.(total 4 plots - 2 plots for each dataset). (4x4 = 16 points)

   (b) Include the best RMSE and MAE value achieved (as well as which fold achieves this) in your report. (2x2 = 4 points)

   (c) For each dataset, analyze and describe which of the loss leads to better performance. (Hint: Compare the values of RMSE and MAE). (5 points)

   (d) What is the relationship between MAE and RMSE? Under what conditions are RMSE and MAE expected to give similar values? Which loss will you prefer in such a case and why? (1+1+3 = 5 points)

   (e) Implement the normal equation form (closed form) of linear regression and get the optimal parameters directly. Consider the **Dataset 1** and the most appropriate loss function you've described for this dataset in part(c). Compute the training and validation loss for the best fold for this loss described in part(b) using these optimal parameters. (5 points)

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

   Use **Banknote Authentication Dataset** for this question. Conduct EDA (exploratory data analysis) given below:

   - Analyze the class distributions and comment on the feature values for each of the given features. (5 points)

   Perform a train:val:test split in the ratio 7:1:2 and implement Logistic Regression based on the given template functions. (5x4 = 20 points)

(a) Using Stochastic Gradient Descent (SGD), choose an appropriate learning rate and the number of epochs (iterations). Report the accuracy obtained on both the training and test set.

(b) Include plots between training loss v/s iterations and validation loss vs iterations.

(c) Re-run your implementation for 3 variations in learning rates - 0.0001, 0.01, 10

(d) Now, implement Batch Gradient Descent (BGD) and re-run (a), (b), (c) with BGD.

Compare the performance of BGD and SGD, w.r.t the following observations: (2.5x4 = 10 points)

(a) Loss plots

(b) Number of epochs taken to converge.

(c) Use sklearn's LogisticRegression implementation on the same dataset above.

(d) Report the accuracy obtained on both the training and test set. Use the same hyper-parameters as in the SGD implementation in 2(a), and compare sklearn's performance with that.

3. (10 points) You have a logistic regression model and you are using mean squared error loss along with it. Assume you have a datapoint for which the model produces really wrong results (eg : $y\_true = 1$ and y_pred is approaching 0 or vice versa). For this datapoint, show that the gradient calculated during gradient descent would approach 0. What are the implications of this, would the model be able to learn effectively? What would happen if we use cross entropy loss instead?

4. (2.5x4 = 10 points) A cancer research institute was conducting a study on the recurrence of neuroblastoma in children. They were examining the feasibility of using logistic regression to find out the likelihood that a patient will have disease recurrence in the next 5 years. A random sample of 33 patients was selected. The data collected from the patient consists of the extent to which the spread of the disease has occurred ($X1$, in percentage) and the age of the patient ($X2$, in years). The pilot study was followed by a revaluation of the patient condition after 5 years which was used to determine whether the disease recurred ($Y = 1$) or didn't recur ($Y = 0$) in 5 years. The data pertaining to the patients **download**:

   - Estimate $\beta 0$, $\beta 1$, and $\beta 2$ using MLE.
   - State the fitted response function.
   - Obtain $\exp(\beta 1)$ and $\exp(\beta 2)$ and interpret these numbers.
   - What is the estimated probability that a patient with 75% of disease spread and an age of 2 years will have a recurrence of disease in the next 5 years?

5. (Bonus 5 points) The linear model with several explanatory variables is given by equation:

$$y_i = \beta_1 + \beta_2 x_{2i} + \beta_3 x_{3i} + ... + \beta_k x_{ki} + \epsilon_i \tag{1}$$

For the purpose of analysis, it is convenient to express the above linear model in *matrix form* as shown in equation 2:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2}$$

Where,

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \ \mathbf{X} = \begin{pmatrix} 1 & x_{21} & \dots x_{k1} \\ \vdots & \vdots & \vdots \\ 1 & x_{2n} & \dots x_{kn} \end{pmatrix}, \ \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_k \end{pmatrix}, \ \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix} \tag{3}$$

Write the matrix expression for the sum of squared errors loss function. Derive an expression to find the $\boldsymbol{\beta}^*$ that minimizes this loss for the above linear regression problem. In order to derive the *least squares solution*, you would need to differentiate the matrix form directly. Once you have the solution expression, write the conditions under which the solution (in the matrix form) will exist.