AMS 598: Big Data Analysis (Fall 2024)

Project #1

Due Sep 24th, 2024

1. You need to submit (1) a report in PDF and (2) your .ipynb code file, both to Brightspace.
2. Your PDF report should include (1) answers to the non-programming part, and (2) results and analysis of the programming part. For the programming part, your PDF report should at least include the results you obtained, for example the accuracy, training curves, parameters, etc. You should also analyze your results as needed.
3. Please put all your files (PDF report and code files) into a compressed file named "Proj#_FirstName_LastName.zip"
4. Unlimited number of submissions are allowed on Brightspace and the latest one will be timed and graded.
5. All students are highly encouraged to typeset their reports using Word or LaTeX. In case you decide to hand-write, please make sure your answers are clearly readable in scanned PDF.
6. Only write your code between the following lines. Do not modify other parts.

   ### YOUR CODE HERE

   ### END YOUR CODE
7. Please read and follow submission instructions. No exception will be made to accommodate incorrectly submitted files/reports.
8. Please start your submission to Brightspace at least 15-30 minutes before the deadline, as there might be latency. We do NOT accept E-mail submissions.

Linear Models for Handwritten Digits Classification: In this project, you will implement the binary logistic regression model on a partial dataset from MNIST. In this classification task, the model will take a 16×16 image of handwritten digits as inputs and classify the image into different classes. For the binary case, the classes are 1 and 2. The "data" fold contains the dataset which has already been split into a training set and a testing set. All data examples are saved in dictionary-like objects using "npz" file. For each data sample, the dictionary key 'x' indicates its raw features, which are represented by a 256-dimensional vector where the values between $[-1, 1]$ indicate grayscale pixel values for a 16×16 image. In addition, the key 'y' is the label for a data example, which can be 0, 1, or 2. The starting code is given, and you must implement the models using the starting code.

1. Data Preprocessing (35 points)

   (a) (10 points) We use two hand-crafted features:

   The first feature is a measure of symmetry. For a 16×16 image x, it is defined as

$$F_{symmetry} = -\frac{\sum_{pixel} |x - flip(x)|}{256},$$

where 256 is the number of pixels and $flip(\cdot)$ means left and right flipping.

The second feature is a measure of intensity. For a 16 × 16 image x, it is defined as

$$F_{intensity} = \frac{\sum_{pixel} x}{256},$$

which is simply the average of pixel values.

Implement them in the function *prepare_X*.

(b) (4 points) In the function *prepare_X*, there is a third feature which is always 1. Explain why we need it.

(c) (6 points) The function *prepare_y* is already finished. Note that the returned indices stores the indices for data from class 1 and 2. Only use these two classes for binary classification and convert the labels to +1 and -1 if necessary.

(d) (15 points) Test your code and visualize the training data from class 1 and 2 by implementing the function *visualize_features*. The visualization should not include the third feature. Therefore it is a 2-D scatter plot. Include the figure in your submission. [Hint: For visualizing features, we have two classes in the training data, then the figure should be similar to the Lecture04-Regular_Linear_Models slide titled "Intensity and Symmetry Features".]

2. Sigmoid logistic regression (65 points): In this problem, **please follow the instructions in the starting code. Please use data from class 1 and 2 for the binary classification.**

   (a) (10 points) Based on the gradient for logistic regression introduced in class, implement the function *_gradient*.

   (b) (20 points) There are different ways to train a logistic regression model. In this assignment, you need to implement gradient descent and stochastic gradient descent in the functions *fit_GD*, and *fit_SGD*, respectively. [Hint: Pay attention to the "randomness" for SGD as we introduced in class.]

   (c) (10 points) Implement the functions *predict* and *score* for prediction and evaluation, respectively. Additionally, please implement the function *predict_proba* which outputs the probabilities of both classes.

   Let's use *fit_SGD* for all the following questions (d), (e), (f). Please refer to the last slide of the T04 lecture for all the following questions.

   (d) (10 points) Test your code and find the best hyper-parameters, including learning rates and iteration numbers. You will be guided to try different learning rates and different iteration numbers. Then you will need to find the best combination (of learning rate and iteration number) based on the validation set. [Hint: Use two "for loops" to find the best combination.]

   (e) (15 points) Visualize the best model so far after training by implementing the function *visualize_results*. The visualization should include the 2-D scatter plot of training data from classes 1 and 2 and your decision boundary. Include the figure in your submission. [Hint: For visualizing the logistic regression results, you will include the features from above in 1.(d), as well as the linear boundary (the trained model) that separates the data.]