Machine Learning

Assignment 3 - Binary Classification
Summer 2023

Introduction

In this assignment you will implement Linear Discriminant Analysis (LDA) and Logistic Regression classifiers for the purpose of binary classification.

You may **not** use any functions from an ML library in your code. And as always your code should work on any dataset that has the same general form as the provided one.

Grading

Although all assignments will be weighed equally in computing your homework grade, below is the grading rubric we will use for this assignment:

Part 1 (Theory)	10pts
Part 2 (LDA)	30pts
Part 3 (Logistic Regression)	60pts
TOTAL	100 pts

Datasets

Spambase Dataset (spambase.data) This dataset consists of 4601 instances of data, each with 57 features and a class label designating if the sample is spam or not. The features are *real valued* and are described in much detail here:

https://archive.ics.uci.edu/ml/machine-learning-databases/spambase/spambase.names

Data obtained from: https://archive.ics.uci.edu/ml/datasets/Spambase

1 Theory

1. For the function $J = (x_1 w_1 - 5x_2 w_2 - 2)^2$, where $w = [w_1, w_2]$ are our weights to learn:

   (a) What are the partial gradients, $\frac{\partial J}{\partial w_1}$ and $\frac{\partial J}{\partial w_2}$? Show work to support your answer (6pts).

   $$\frac{\partial J}{\partial w_1} = 2(x_1 w_1 - 5x_2 w_2 - 2) \cdot x_1$$
   $$\frac{\partial J}{\partial w_2} = 2(x_1 w_1 - 5x_2 w_2 - 2) \cdot 5x_2$$

   (b) That are the value of the partial gradients given current values of $w = [0, 0], x = [1, 1]$ (4pts)?

   $$\frac{\partial J}{\partial w_1} = 2(0 \cdot 1 - 5(1) \cdot 0 - 2) \cdot 1 = \textbf{-4}$$
   $$\frac{\partial J}{\partial w_2} = 2(0 \cdot 1 - 5(1) \cdot 0 - 2) \cdot 5 = \textbf{-20}$$

2 Linear Descriment Analysis (LDA) Classifier

For your first programming task, you'll implement, train and test a *Linear Discriminant Classifier*.

Download the dataset *spambase.data* from Blackboard. As mentioned in the Datasets area, this dataset contains 4601 rows of data, each with 57 continuous valued features followed by a binary class label (0=not-spam, 1=spam). Your code should work on any dataset that lacks header information and has several comma-separated continuous-valued features followed by a class id $\in 0, 1$.

Write a script that:

1. Reads in the data.
2. Randomizes the data.
3. Selects the first 2/3 (round up) of the data for training and the remaining for validation.
4. Standardizes (z-scores) the data (except for the last column of course) using the training data
5. Deterimines the optimal direction of projection for linear discriminant analysis.
6. Classifies each *validation* sample by projecting it and assinging it the class whose training post-projection mean it is closes to.
7. Computes the *accuracy* of the training and validation data and the *precision, recall, and f-measure* for the validation set.

Implementation Details

1. Seed the random number generate with zero prior to randomizing the data

Results:
Accuracy: 0.897
Precision: 0.843
Recall: 0.894
F-Measure: 0.868

3 Logistic Regression

Finally, lets design, implement, train and test a *Logistic Regression Classifier*. For training and validation, we'll use the same dataset as in the previous programming part, and as always, your code should work on any dataset that lacks header information and has several comma-separated continuous-valued features followed by a class id $\in 0, 1$.

Write a script that:

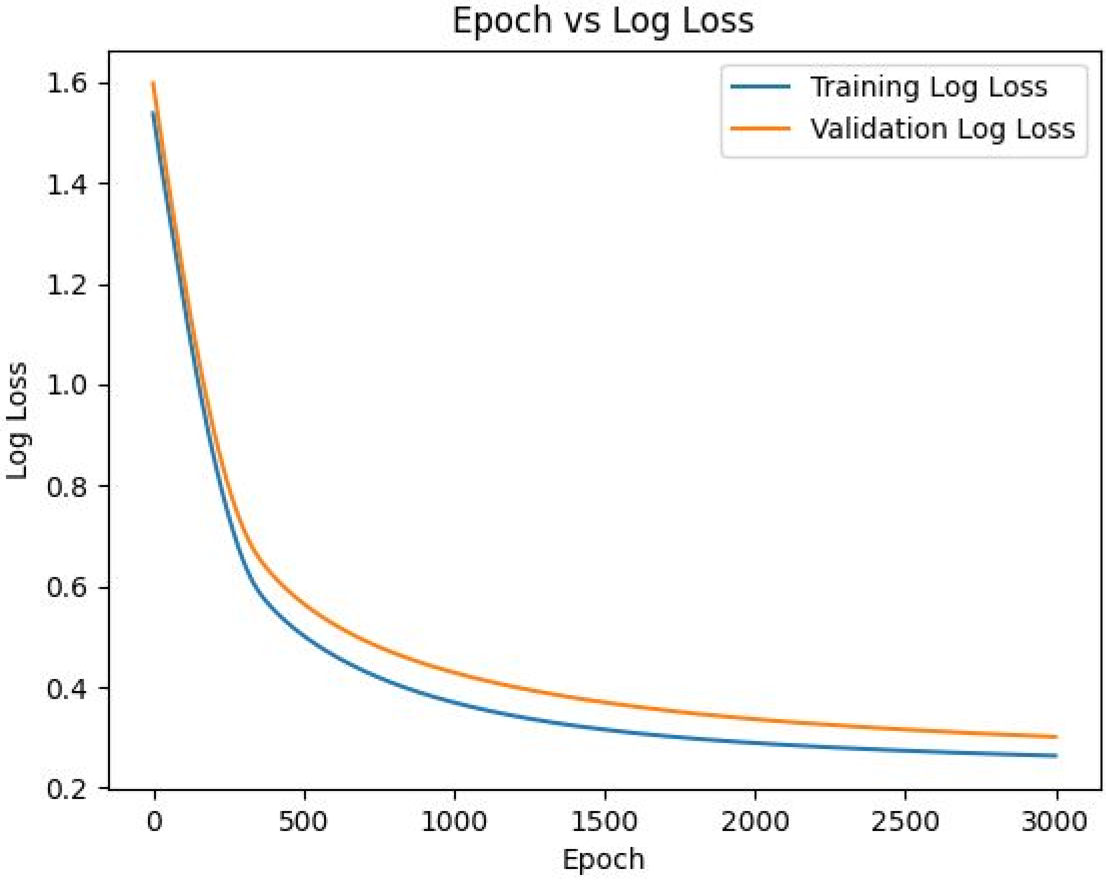
1. Reads in the data.
2. Randomizes the data.
3. Selects the first 2/3 (round up) of the data for training and the remaining for validation.
4. Standardizes (z-scores) the data (except for the last column of course) using the training data.
5. Trains a logistic classifier, keeping track of the log loss for *both* the training and validation data as you train.
6. Classify each validation sample using your trained model, choosing an observation to be spam if the output of the model is $\geq 50\%$.
7. Compute the following statistics using the validation data results:
   (a) Precision
   (b) Recall
   (c) F-measure
   (d) Accuracy (expect around 90%)
8. Plots epoch vs log-loss of both the training and validation data sets.

Implementation Details

1. Seed the random number generate with zero prior to randomizing the data
2. We will let you determine appropriate values for the learning rate, η, the initial parameter values, as well as an appropriate termination criteria.
3. You'll need to figure how to deal with $log(0)$ issues.

In your report you will need:

1. The statistics requested for your Logistic Classifier.
2. The plot of epoch vs log-loss for the training and validation data sets (on the same graph).



Results:

Accuracy: 0.919
Precision: 0.905
Recall: 0.877
F-Measure: 0.891

Submission

For your submission, upload to Blackboard a single zip file (again no spaces or non-underscore special characters in file or directory names) containing:

1. PDF Writeup
2. Source Code
3. readme.txt file

The readme.txt file should contain information on how to run your code to reproduce results for each part of the assignment.

The PDF document should contain the following:

1. Part 1: Answer to theory questions
2. Part 2:
   (a) Requested Classification Statistics
3. Part 3:
   (a) Requested Classification Statistics
   (b) Requested plot.