Machine Learning

Lab 4 - Logistic Regression and Gradient-Based Learning
Spring 2025

Introduction

In this lab you will implement 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.
Your task will be to write a *single script* such so that we can run it in the command line and it output (and/or displays) the requested information/figures.

Grading

- +1pts Parses dataset correctly.
- +1pt Zscores correctly.
- +1pt Does some gradient based computations.
- +1pt Does some learning.
- +1pt Correct termination criteria set up.
- +1pt Creates some plot.
- +1pt Creates correct plot.
- +1pt Runs to convergence.
- +2pts Statistics within a reasonable range ($Accuracy \approx 90\%$).

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 Logistic Regression

Lets design, implement, train and test a *Logistic Regression Classifier*. For training and validation, we'll use the dataset mentioned in the *Dataset* section, but 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 target column of course) using the training data.
5. Trains a logistic classifier, keeping track of the mean log loss for *both* the training and validation data as you train.
6. Classifies each training and validation sample using your trained model, choosing an observation to be spam if the output of the model is $\geq 50\%$.
7. Computes the the accuracy for both the training and validation sets (expect around 90%. In addition, provide the precision, recall and f-measure for the validation set.
8. Plots epoch vs mean 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. Make sure to add in a numeric stability constant, as needed.
4. Make sure to add a bias feature.
5. For full credit, you must run your algorithm to *convergence* (or at least until visually it pretty much flattens out).

Your script should produce the following when run from the command line:

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