Huanying Yeh

Mr. Tivnan

EN.580.140.12.IN21 Statistical Foundations of Machine Learning

15 January 2021

Final Project: Binary Classification Model of Annual Income based on Demographic Information

**Introduction**

This project is a classification model implementation that predicts whether a person’s income exceeds $50K/yr based on demographic data from the UC Irvine Machine Learning Repository: <http://archive.ics.uci.edu/ml/datasets/Census+Income>.

The assignment instructions are adopted by professor Hung-Yi Lee’s undergraduate Machine Learning course at National Taiwan University. The slideshow can be found [here](https://drive.google.com/file/d/1DOqby_K_9TK2QVUr-C2qYg0amZkRu21U/view?usp=sharing). The dataset is revised so make the features easier to manipulate. Download the data files [here](https://drive.google.com/drive/folders/1nnbMOILn2FM_CzdScgw7zRkTZhOKhQ32?usp=sharing).

As seen from above, the datasets’ input, x\_train, has vectors with 510 dimensions. The training outputs, y\_train, are binary values of 0 (income below $50k) or 1 (income above $50k). There are 48842 instances of labeled data. Because we have ample amounts of data and simple outputs, we use supervised learning to train a model with logistic regression. Simple implementations of Adagrad and mini-batch training are also made.

The outputs of the code are:

* A graph of the loss values of the training set and validation sets over time
* A graph of the accuracy % of the model over time
* A .csv file that predicts the y values of the testing data

**Model Design and Statistical Foundation:**

The workflow is as follows:

* Import and clean up the data files: x\_train, y\_train, and x\_test.
* Split up the training and validation sets.
* Normalize the training sets so that the means of all dimensions of x are 0 and the variances are all 1.
* Use the dot product to compute the loss values of each set of parameters passed in.
* Write functions for shuffling, sigmoid(), creating a proposed function \_f(x, w, b), prediction, accuracy, cross entropy, and gradient descent.
* Implement mini-batch and Adagrad.
* Print and update the parameters.
* Plot the graphs mentioned above.
* Write the prediction file, y\_test.

**Code Implementation:**

The python file can be accessed [here](https://drive.google.com/file/d/17P2tvqLgAPKj59Pv-mbzilidP-8VtRMo/view?usp=sharing). All files are also uploaded to Blackboard. When running the code, edit the file paths.

Works Cited

Bishop, Christopher M. 2006 Pattern Recognition and Machine Learning 205-206

Lee, Hung-yi 2017 Machine Learning

<https://www.youtube.com/channel/UC2ggjtuuWvxrHHHiaDH1dlQ/playlists>