Contents

Project 4: Linear Classifier

Jaudat Raza JAUDAT.RAZA@JHU.EDU
Master of Computer Science
John Hopkins University
Baltimore, MD 21218

Editor: Jaudat Raza

Abstract

This document will cover the implementation process of Logistic Regression and Adaline, which are linear classifier. The resource used will be linked below in the resource section of the document. The document will follow a simple pattern for which a less than 5-minute video will be submitted as well. The Jupyter Notebook starts with getting the data into Pandas Data from the link provided in the data source. After that, the data set give in the project will run through both the algorithm and we will see how the update rule for both the algorithm impacts the results we get out of our tests.

1 Problem Statement

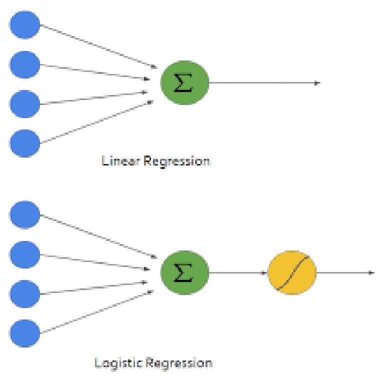
The purpose of this assignment is to give us a firm foundation in comparing a variety of linear classifiers. We will compare two different algorithms. These algorithms include Adaline and Logistic Regression. All five data set will run through each algorithm. The data we are running is from UCI data base. The data set we will run is Breast Cancer, Glass, Iris, Soybeans, and Vote. My hypothesis is that I expect to see from the result is that whenever a data set contain more continuous values in its features, Adaline will perform better than the Logistic regression because of updating rule, where there is only one class, but with the implementation of multiclass Logistics regression, it will perform better with data set that have multiple classes.

2 Algorithm Studied

In this project we are covering Adaline and Logistic Regression. Both the algorithm has similarities and difference, but to make some sequential sense. Each will be described from start to end separately. Let us start with Logistic Regression

2.1 Logistic Regression

Logistic Regression is used for classification purposes. As we remember Linear Regression is used to determine the value of continuous dependent variable. Though linear regression can do, but logistic regression is used when the dependent variable is categorical. It could have multiple categories, but not continuous. When there are only two possible outcomes, the regression is then referred as Binary Logistic Regression. If the there are three or more categories and they are not in order, then it is called multinomial logistic regression. An example of this would be Red, Yellow and Green. If the categories are in order, then the regression is called Ordinal Logistic Regression. An example of this would be 1st, 2^{nd,} and 3rd. In Linear Regression, we output the weighted sum input. In Logistic Regression we find the weighted sum and pass it through an activation function which output the category.



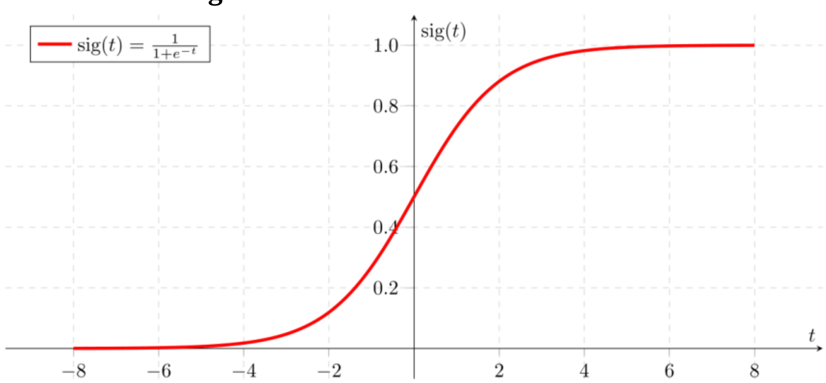
2.1.1 Activation Function

So, we use Sigmoid Function as our activation function for Logistic regression. If the let say Z goes to infinity, Y predicted will become 1 and if Z goes to negative infinity Y predicted will become 0.

2.1.2 Sigmoid Function Equation:

$$sig(t) = \frac{1}{1 + e^{-t}}$$

2.1.3 Sigmoid Function Diagram



2.1.4 Decision Boundary

To Predict which class the data belongs too a threshold is set. Based upon the threshold the data is classified. So, in Sigmoid Function value above 0.5 would be classify as 1 and anything below would be classified as 0. Decision boundaries can be linear and nonlinear. For more complex boundaries increase in polynomial order will make the decision boundaries more complex.

2.1.5 Cost Function

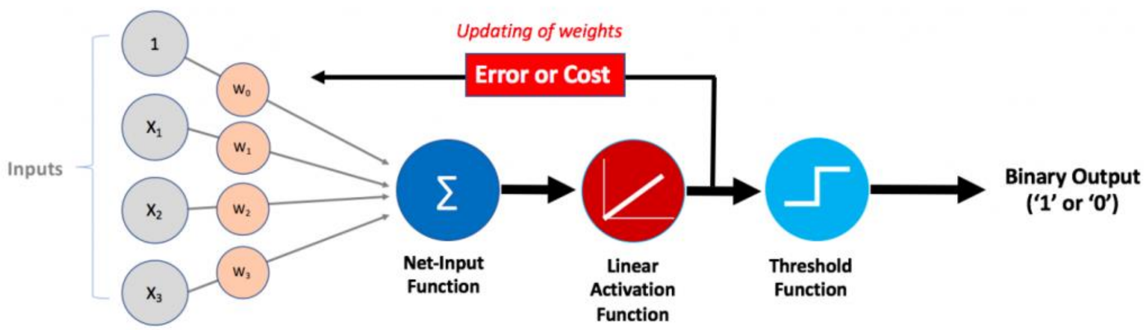
This negative function is because when we train, we need to maximize the probability by minimizing loss function. Decreasing the cost will increase the maximum likelihood

$$Cost\left(h_{\emptyset}(x), Y(actual)\right) = -\log\left(h_{\emptyset}(x)\right) if\ y = 1, -\log\left(1 - h_{\emptyset}(x)\right) if\ y = 0$$

2.2 Adaline

Adaline stands for Adaptive Linear Neuron. Adaline is also called as single-layer neural network. It is very similar to perceptron which we learned in module 8. Just like perceptron, it is important to understand Adaline because is a foundation for learning neural network. Adaline could find to be useful in understanding how gradient descent can be used to learn the weights which when combined with input signals is used to make predictions based on unit step function output.

2.2.1 Adaline representation Diagram



My Algo and the explanation below with using the following representation of the diagram above.

2.2.1.1 Input Function

Input signals of different weights get added to be fed into the activation function. The combined input or sum of weighted inputs can also be called as net input. Do note that the first input is constant 1. That is the case for weight multiplication with inputs.

2.2.1.2 Activation Function

The input from is given to the activation function, in which we are using a linear function. If Z is the net input from the previous step, then the equation would be as follows.

$$F(X) = Z$$

2.2.1.3 Learn Weights

The output from the activation function is used to calculate the changes in weights related to different input which will be updated to learn new weights. This is the feedback loop that we see in the diagram above and that is why Adaline does continuous update instead of batch update.

2.2.1.4 Threshold Function

This is just like the boundary decision. Binary prediction based on unit step function. Prediction is made based on the unit step function which provides binary output as 1 or 0 based on whether the output of activation function is greater than or equal to zero. If the output of activation function is greater than or equal to zero, the prediction is 1 or else 0.

3 Process

The runs are done with 5-fold cross-validation so we can compare results statistically. The project allowed us to pick from classification error, cross entropy loss, or mean squared error (as appropriate) for the loss function.

4 Tuning

This is the section that describes all the data and how it was updated to make it work with the Algorithm. This will be described in the video as well.

4.1 Breast Cancer

1) Dropped missing data from the set
2) Change the class, where 2 is 1 and 4 is 0 now
3) Converted BareNuclei to numeric type so the math could be done on it.

4.2 Glass

4) Changes the possible Glass Types to in order from 0 to 5

4.3 Iris

1) If Iris-setosa was in the class, the class value was changed to 0
2) If Iris-virginica was in the class, the class value was changed to 1
3) If Iris- versicolor was in the class, the class value was changed to 2

4.4 Soybean

1) Removed columns that did nothing which included
   a. leafspotshalo
   b. leafshread
   c. leafmalf
   d. leafmild
   e. seed
   f. moldgrowth
   g. seeddiscolor
   h. seedsize
   i. shriveling
2) Updated the Distribution to 1, 2, 3, 4

4.5 House Vote

1) The Missing Data was replaced with 2, which show no vote given for the class
2) One Hot Encoding on Class for demo and Republican
3) Rest of the attributes y is 1 and no was 0
4) Place the Class as the lass column

5 Results

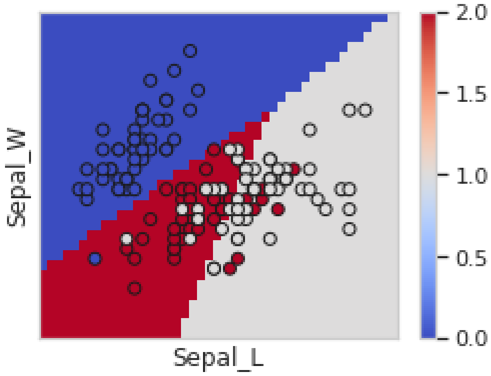
I have decided to use learning rate of 0.001 and the threshold is set to 0.001 as well. This is my step to make sure that either the error is this close to zero within the given iteration or the error starts to increment, that is when the loop ends. The steps are taken by multiplying the slope of error times the learning rate, so at first, the algo will take big steps and as we get close to 0, the steps will become smaller and smaller. For logistics max 10000 iteration. For Adaline 200.

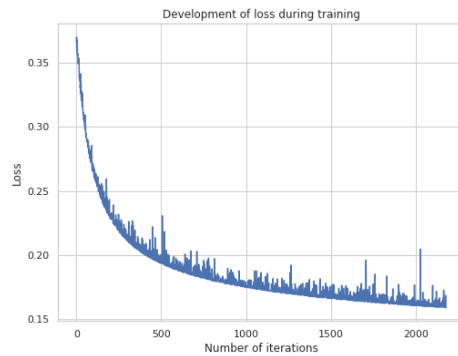
5.1 Iris Data

5.1.1 Logistic Regression

1. Sepal Length value is from: [4.3, 7.9]

2. Sepal Width value is from: [2.0, 4.4]
3. Possible Class for the Data is: [0 1 2]
4. Accuracy = 80.5%





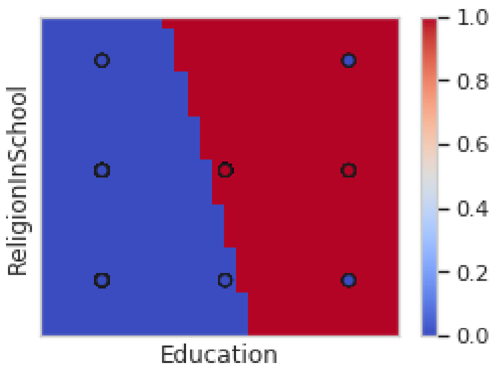
5.1.2 Adaline

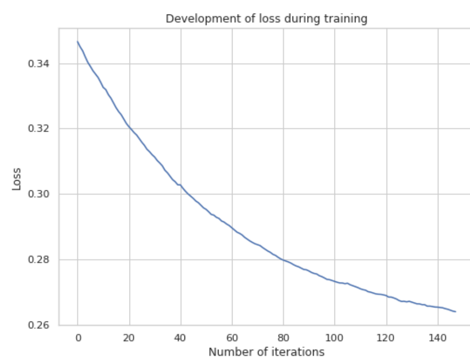
Fold	Score
Fold 1	0.423077
Fold 2	0.444444
Fold 3	0.444444
Fold 4	0.444444
Fold 5	0.444444

5.2 House Vote

5.2.1 Logistic Regression

1. Education value is from: [0, 2]
2. Religion in School value is from: [0, 2]
3. Possible Class for the Data is: [1 0]
4. Accuracy = 82.02%





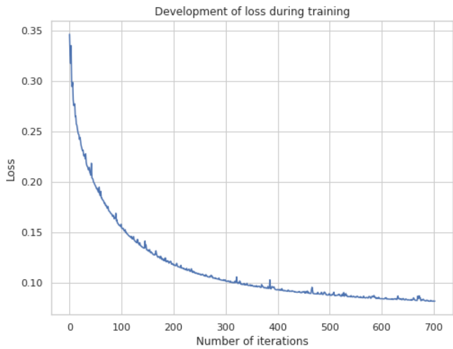
5.2.2 Adaline

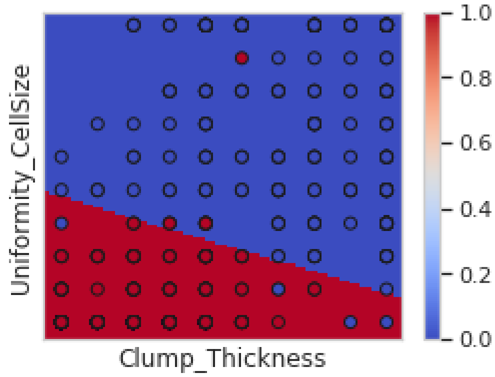
Fold	Score
Fold 1	0.74359
Fold 2	0.74359
Fold 3	0.74359
Fold 4	0.74359
Fold 5	0.74359

5.3 Breast Cancer

5.3.1 Logistic Regression

1. Clump Thickness value is from: [1, 10]
2. Uniformity Cell Size value is from: [1, 10]
3. Possible Class for the Data is: [1 0]
4. Accuracy = 94.28%





5.3.2 Adaline

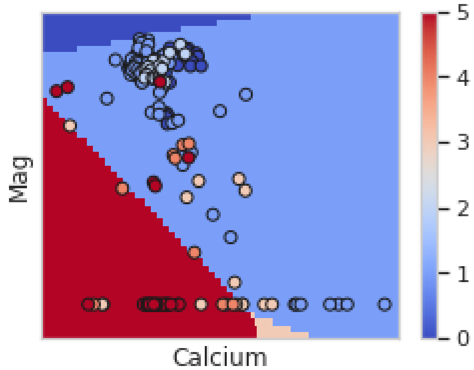
Fold	Score
Fold 1	0.540984
Fold 2	0.544715
Fold 3	0.540984
Fold 4	0.544715
Fold 5	0.544715

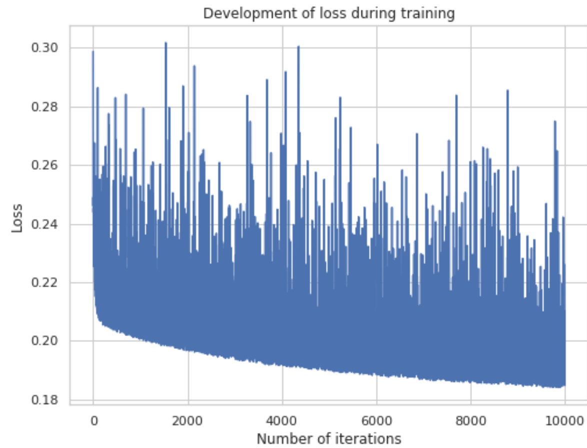
5.4 Glass

5.4.1 Logistic Regression

1. Calcium value is from: [5.43, 16.19]
2. Mag value is from: [0.0, 3.98]
3. Possible Class for the Data is: [0 1 2 3 4 5]

4. Accuracy = 46.09%





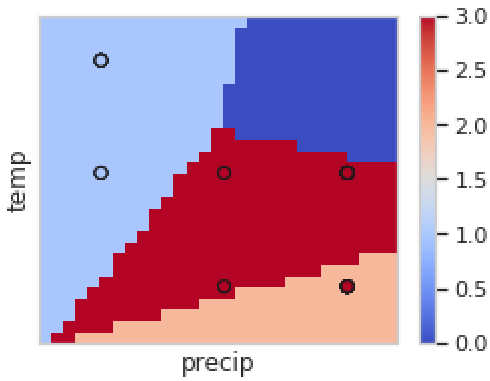
5.4.2 Adaline

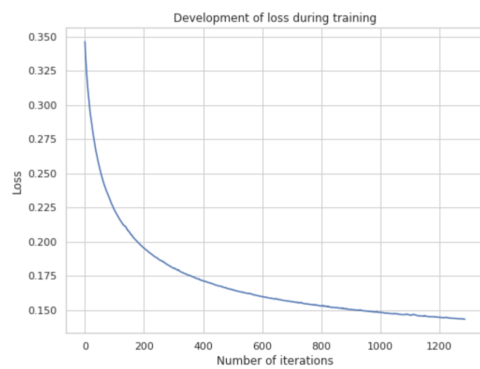
Fold	Score
Fold 1	0.421053
Fold 2	0.421053
Fold 3	0.421053
Fold 4	0.421053
Fold 5	0.435897

5.5 SoyBean

5.5.1 Logistic Regression

1. precip value is from: [0, 2]
2. temp value is from: [0, 2]
3. Possible Class for the Data is:[0 1 2 3]
4. Accuracy = 69.56%





5.5.2 Adaline

Fold	Score
Fold 1	0.375
Fold 2	0.375
Fold 3	0.375
Fold 4	0.375
Fold 5	0.444444

6 Conclusion

My hypothesis was incorrect. Logistic Regression did better in almost all the cases. Though the accuracy did decrement whenever there were more then two classes and the features did not help the data like it should. Like we see for Glass data set there are like 5 possible classification and that data set came back with the least accuracy in Logistic Regression and Adaline. Breast Cancer data performed the best because with only 2 classes and feature that really did help the classification. Depending on which features picked really did impact the accuracy of the results. A step to pick the features that best impact the result would be something to work on for further work into these two algorithms.

7 Resources

- StatQuest with Joh Starmer. “Gradient Descent, Step-by-Step.” YouTube, 05 Feb. 2019, https://www.youtube.com/watch?v=sDv4f4s2SB8.
- Jurafsky, Daniel, and James H Martin. “Logistic Regression.” *Standford Edu*, 2 Oct. 2019, web.stanford.edu/~jurafsky/slp3/5.pdf.
- B. Chen, Y. Zhu and J. Hu, "Mean-Square Convergence Analysis of ADALINE Training With Minimum Error Entropy Criterion," in IEEE Transactions on Neural Networks, vol. 21, no. 7, pp. 1168-1179, July 2010, doi: 10.1109/TNN.2010.2050212.