PLA Without Linear Regression

The PLA method takes the generated sample data. We are considering two random point from the sample and will consider a line with these points as a reference to identify the labels. This line will divide the points into two regions (positive and negative points) on the fact that if the point falls above or on this line we will mark those point with label 1 otherwise the points will be label as -1.

Algorithm Description:

generateData takes two random points from a sample of size N and then identify each sample with these two random points, if the point falls above the line then it is marked as positive otherwise it is marked as negative.

Post the data generation with the labels, next we are trying to find a misclassified point and the weight will be updated accordingly with the rule w=w+y*X and then test all the sample with the updated weights. The outer while loop will execute as long as we have misclassified points in the sample.

1. In the first iteration of the ***pla*** method, the while loop will evaluate to true since the initial weight vectors are all zero. This is achieved using the ***misclassified*** method which return a boolean value as true if still there are misclassified points left in the sample otherwise this method returns false.
2. Choose a random misclassified point with the weight vector. This is done using the method ***getMisclassifiedPoint*** which returns a list of misclassified point with their labels.
3. Update the weight vector in **pla** method and then test all the samples with the updated weights. Also increase the count of iterations since the sample was a misclassified point.
4. Check again if all the points are correctly classified with the updated weight vector.
   - If YES exit the while loop and we have found the weights which correctly classify the data.
   - If NO, then repeat the process again from Step (1) until there are no more misclassified points

At the end of this method we will have a final weight vector and the number of iterations taken by PLA to linear classify the data.

The above procedure is repeated 100 times and then a mean of iterations is evaluated from that which gives the final iteration value. For now, the sample size in the program is kept as 20.The observation chart show in Figure (1) depicts the count of iterations for the above method.

Note: The print statement is commented in the code which displays the count of iteration for PLA and a weight vector.

PLA with Linear Regression (using pseudoinverse)

This is implemented similar to the PLA, the only difference here is the calculation of learnt weight vector using the pseudo inverse method. The output of learnt weight vector is supplied to PLA method again and then the number of iterations are noted for this.

- Determine the weight vector using the pseudo inverse method. ***Pseudoinverse*** take X, Y as input and returns learnt weight vector w1
- Next the ***pla*** method is invoked using the w1 calculated from above step (This will now be the same as the pla method described in pla without linear regressions)

At the end of this method we will again have a final weight vector and the number of iterations taken by PLA with pseudo inverse to linear classify the data

The above procedure is repeated 100 times and then a mean of iterations is evaluated from that which gives the final iteration value. For now, the sample size in the program is kept as 20.

The observation chart show in Figure (1) depicts the count of iterations for the above method.

Assumptions:

Currently the value of N is set as 20, it needs to be changed accordingly while running the program. For simplicity and to avoid ambiguity of the weighted vector name, the name **w1 and w2** are interchangeably used in place of **w0** and **w1** for pseudo inverse matrix. The learnt weighted vectors are stored in **w1** which is supplied as an input to **pla** method. The output of pla with this will be again a weighted vector with notation **w2**

Also the plot graph is commented in the code for brevity. The graphs are shown using the plot method of python. Sometimes it has been observed that repeated testing increases the number of iterations for PLA with pseudo inverse.

Note: The print statement is commented in the code which displays the count of iteration for PLA with pseudo inverse and a weight vector.

Observation Chart of PLA (with and without linear regressions)

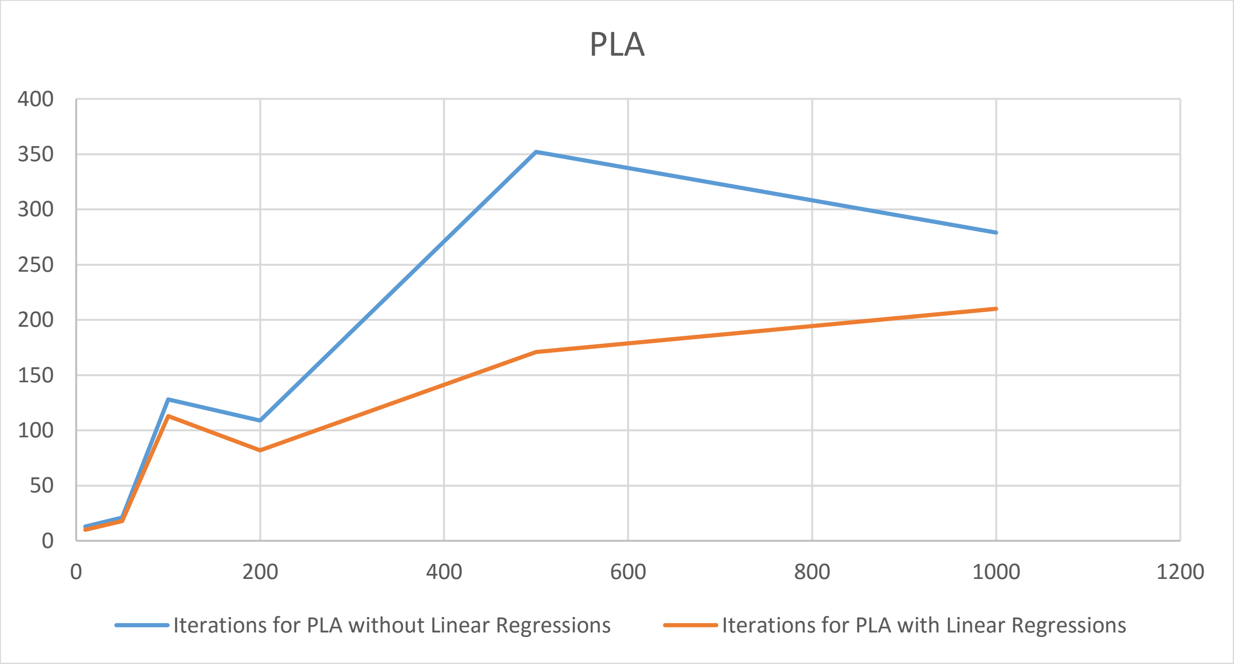
N	**Iterations for PLA without Linear Regressions**	**Iterations for PLA with Linear Regressions**
10	13	10
50	21	18
100	128	113
200	109	82
500	352	171
1000	279	210

Figure (1)

Note: *The results shown above can change significantly depending on the random samples selected. The output changes every time the algorithm runs. The above is one of the outputs measured while running the experiments.*

What does two experiments say:

- On the basis of observations for two experiments we can say the count of iterations reduce for PLA method if the weighted vectors are supplied as an initial input to the PLA. The initializing of weighted vector is done using the pseudo inverse method.
- Also if the data is linearly separable, then both the PLA methods will converge and will linearly classify the data.



Evidently the above chart shows that PLA with linear regression takes less number of iteration as compared to PLA without linear regressions to converge the data if it is linearly separable.

Visualizations Charts (Figures and descriptions are together):

Note: The plot generation code is currently commented in the code and the green and blue dashed line are for PLA with and without linear regressions.

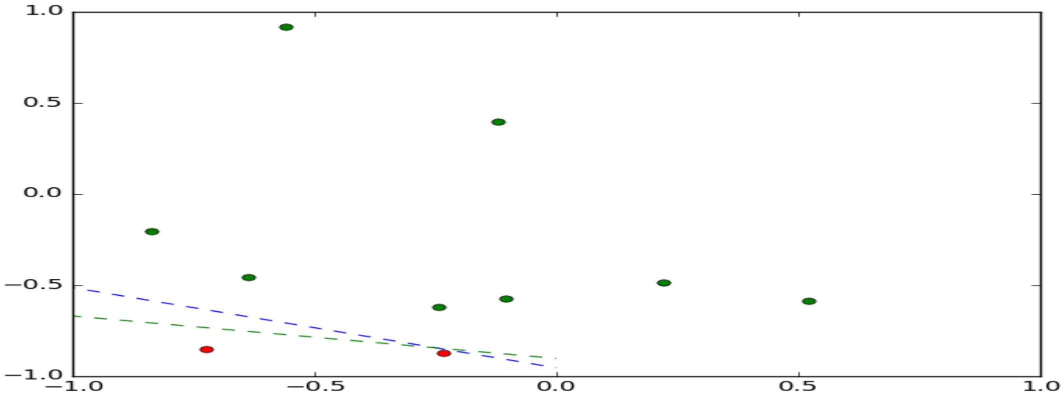


Figure 2

Here the sample size is 10 and blue and green dotted line shows linear separatability by PLA algorithm with and without linear regressions.

- ***Weight Vector with PLA = [-1.   -0.7268259 -2.51424827]***
- ***Weight Vector of PLA with pseudo inverse= [-1. -0.64956727 -2.63190509]***

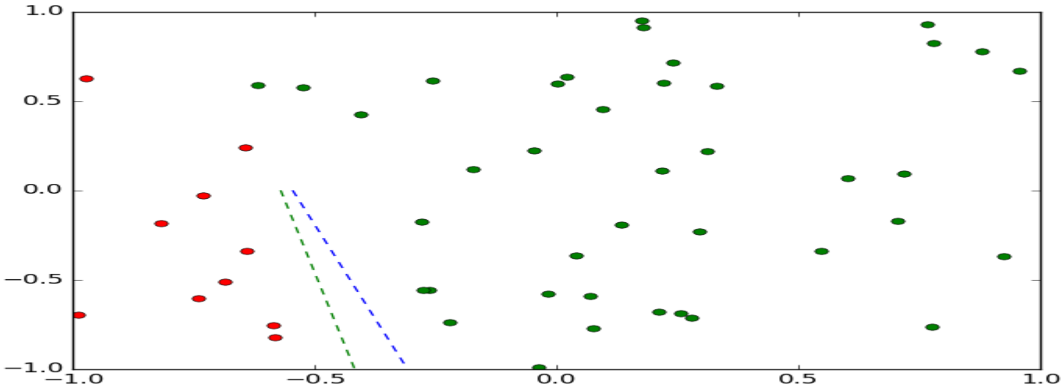


Figure 3

Here the sample size is 50 and green and blue dotted line shows linear separable by PLA algorithm with and without linear regressions.

- ***Weight Vector with PLA =[* 3. 1.4901018 4.33730689]**
- ***Weight Vector of PLA with pseudo inverse=* [3. 1.41665884 4.09385204]**

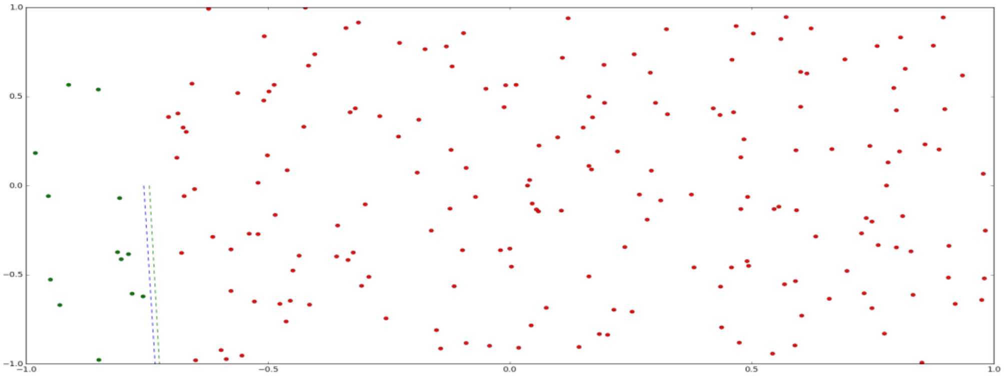


Figure 4

Here the sample size is 100 and green and blue dotted line shows linear separatability by PLA algorithm with and without linear regressions.

- ***Weight Vector with PLA= [0. -2.53125953 2.48899149]***
- ***Weight Vector of PLA with pseudo inverse=* [0. -3.61922549 3.79114546]**

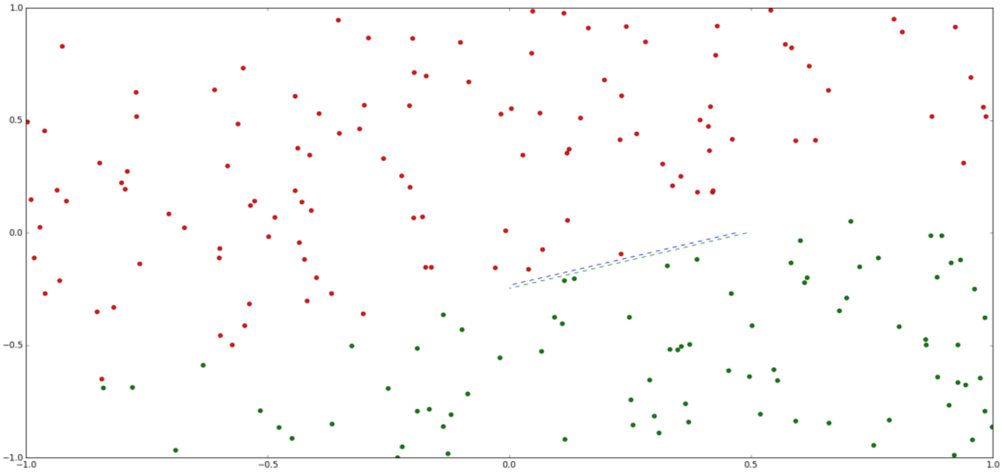


Figure 5

Here the sample size is 200 and green and blue dotted line shows linear separability by PLA algorithm with and without linear regressions.

- ***Weight Vector with PLA= [-1. -8.91411933 8.11619674]***
- ***Weight Vector of PLA with pseudo inverse=*** **[-1. -8.68444131 7.65274165]**

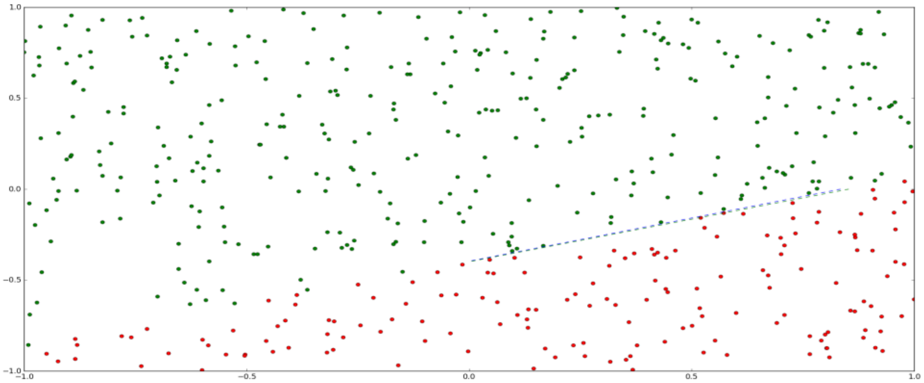


Figure 6

Here the sample size is 500 and green and blue dotted line shows linear separability by PLA algorithm with and without linear regressions.

- ***Weight Vector with PLA= [-8. 4.44634494 9.78227646]***
- ***Weight Vector of PLA with pseudo inverse=*** **[-9. 4.97966508 11.00138994]**

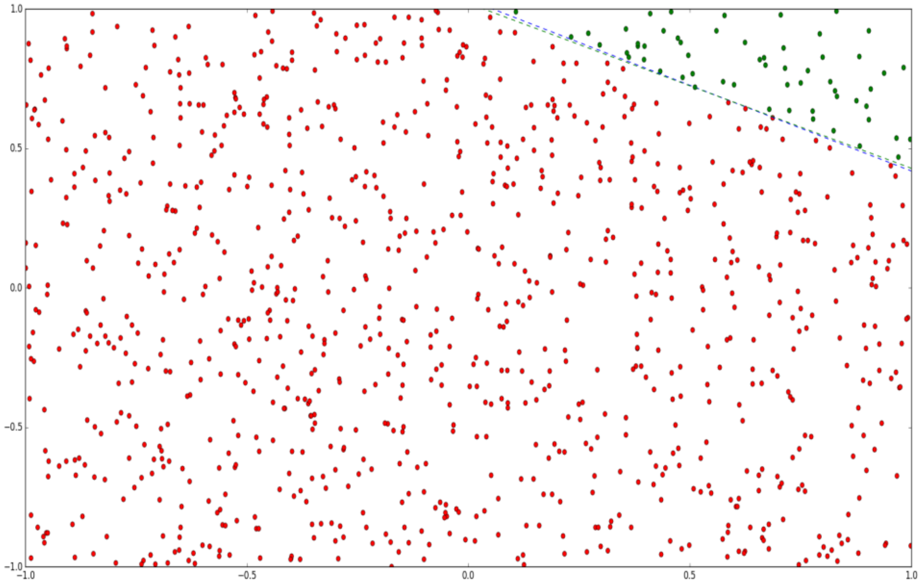


Figure 7

Here the sample size is 1000 and green and blue dotted line shows linear separability by PLA algorithm with and without linear regressions.

- ***Weight Vector with PLA= [-8. 4.44634494 9.78227646]***
- ***Weight Vector of PLA with pseudo inverse=* [-9. 4.97966508 11.00138994]**