

COMP6245 : Lab 2 Report

Thanakorn Panyapiang(31446612)

1 Implementing the Perceptron algorithm

Given the data which has the distribution like Figure 1a, using the Perceptron algorithm to train a linear classifier gives a decision boundary as shown in Figure 1b. It can be noticed that the accuracy improves as numbers of iteration increase which can be observed from Figure 1c.

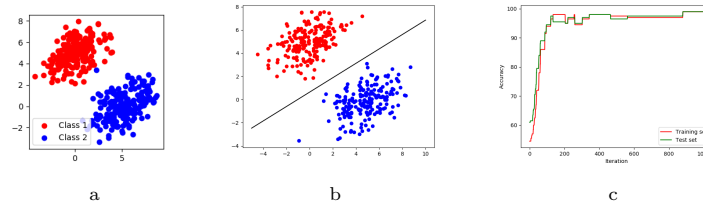


Figure 1

However, the algorithm fails to train a decent linear classifier for the sample which has the distribution like Figure 2a as can be seen from the decision boundary and the accuracy in Figure 2b. and 2c. respectively.

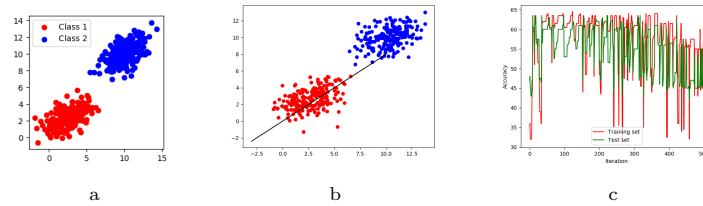


Figure 2

The cause of this issue is the algorithm tries to find the decision boundary that passes the origin, but unlike the first dataset, there is no line which passes the origin that can split two classes in this dataset.

To fix this issue, the algorithm needs to be modified so that the decision boundary doesn't have to pass the origin. This can be done by adding a bias feature to the dataset which will allow the algorithm to shift the decision boundary. The result after adding a bias feature is shown in Figure 3.

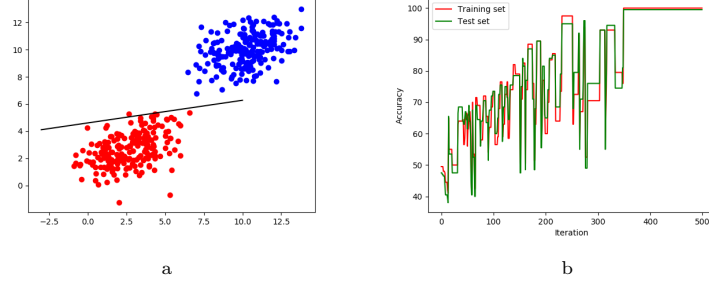


Figure 3

2 Testing with a benchmark dataset

The benchmark dataset which is used to test the algorithm is the Iris dataset. From the scatter plot in Figure 4a and 4b, it can be noticed that the data can be separated using only two attributes.

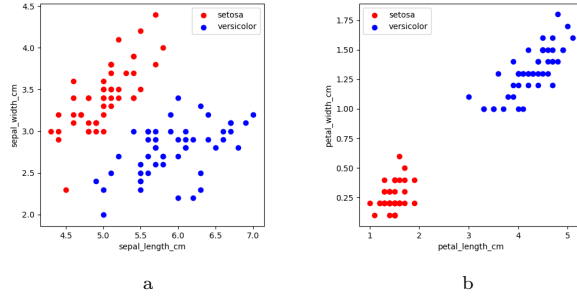


Figure 4

Training a linear classifier for the Iris dataset using the Perceptron algorithm gives the result as illustrated in Figure 5. The model trained from sepal width and sepal length(Figure 5a) give 100 percent accuracy while the classifier trained using petal width and petal length(Figure 5b) has 99 percent accuracy.

