Yixiao Zhang
31353282
yz3a19@soton.ac.uk

Perceptron Algorithm
COMP6245 Lab 2

1 Linear Classifier (no basis)

Two bivariate Gaussian densities are generated with means $m1 = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$ and $m2 = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$ and identical covariance matrix $C = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ as Figure 1 shows. Each Gaussian densities has 200 points. A linear classifier with no basis is built and data are separated to two dataset with same size randomly.

During 200 epochs, with learning rate $alpha = 0.01$, the training process is shown as Figure 2. The percentage of correct increased from 52% to 99.0%(training set) and 98.5%(testing set). Figure 3 shows the trained linear classifier at the end of training process.

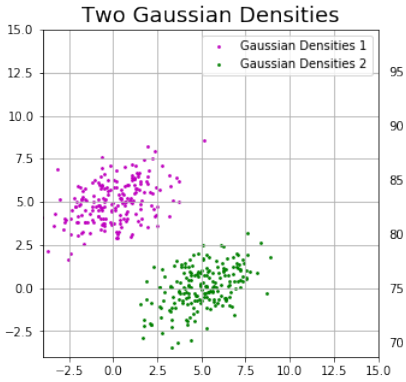

Figure 1: Gaussian Densities

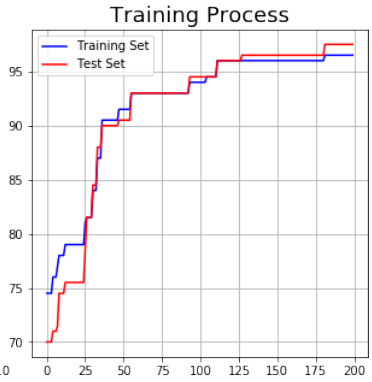

Figure 2: Training Process

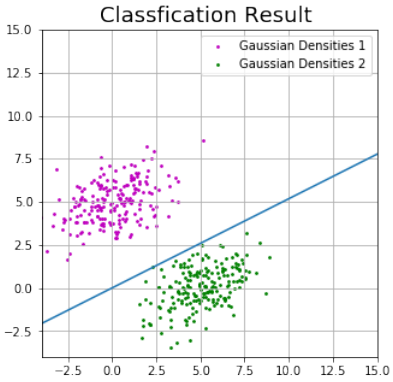

Figure 3: Classification Result

2 Linear Classifier (with basis)

Two bivariate Gaussian densities are generated with means $m1 = \begin{bmatrix} 2.5 \\ 2.5 \end{bmatrix}$ and $m2 = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$ and identical covariance matrix $C = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ as Figure 4 shows. A linear classifier with no basis and a linear classifier with basis are built.

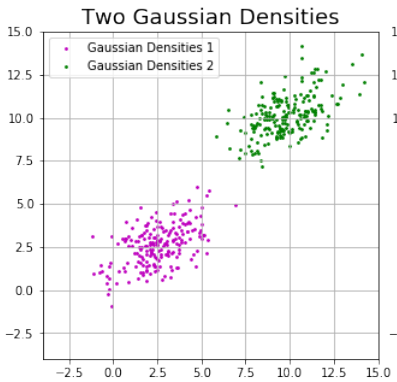

Figure 4: Gaussian Densities

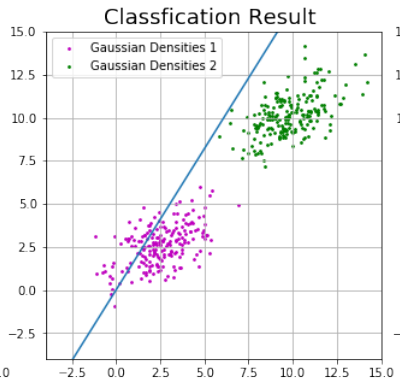

Figure 5: Classification Result

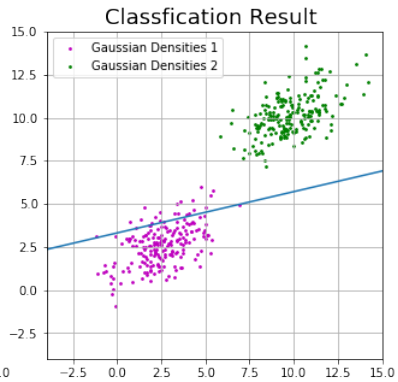

Figure 6: Classification Result

Figure 5 shows the classification result of no basis classifier and Figure 6 shows the classification result of classifier with basis. The testing percentage of correct of these two classifiers are 52.5% and 75.5% respectively. From the result, the classifier with basis could better deal with the distributions which cannot separated by a line pass though the origin point.

3 Linear Classifier in scikit-learn

A automatic linear classifier ***Perceptron()*** is import from sklearn package ***sklearn.linear_model***. The classifier is trained on the training set and testing set which used in the section above. The data distribution is shown in Figure 4. The trained linear classifier is shown in Figure 7. The testing percentage of correct of this classifier is 100%. The reason why this classifier has better performance than the linear classifier before is that package sklearn provide a better algorithm.

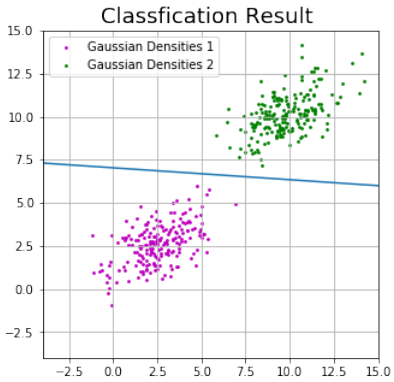

Figure 7: Classification Result

4 Iris Classification Problems from UCI

Iris dataset has 3 class and 4 attributes. In data pre-processing, Virginica class is removed as only two classes can be used. Weka which is one of the most popular data mining tool is used to choose the top two important attributes and the result shows that petal length and petal width should be choose.

Function ***DataFrame.read_csv("filename.csv")*** from package ***pandas*** is used to import dataset to program and function ***DataFrame.values*** is used to convert DataFrame data structure to NumPy vector.

The training process is shown as Figure 8. The percentages of correct achieve 98.0% on both training set and testing set. Figure 9 shows the trained linear classifier at the end of training process.

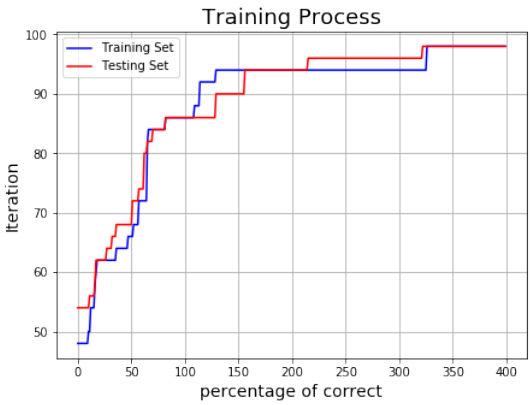

Figure 8: Classification Result

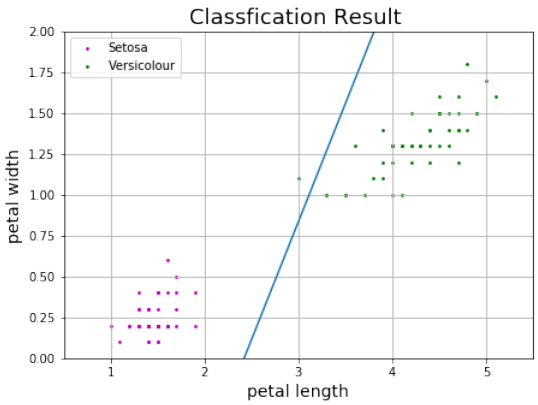

Figure 9: Classification Result