Johnny Joyce (jvjj1u19@soton.ac.uk)

October 12, 2019

1 Applying the perceptron algorithm

The objective of this session was to implement a linear classifier to identify coordinates sampled from two distributions. We began by using 200 samples from a bivariate Gaussian distribution with mean $(0, 5)$ and covariance matrix $\mathbf{C} := \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, and another 200 samples from a bivariate Gaussian distribution with mean $(5, 0)$ and covariance matrix $\mathbf{C}$. Let us henceforth call these distributions 1 and 2, respectively. All samples from distribution 1 were assigned the label "+1" and all samples from distribution 2 were assigned the label "-1".

The data was then randomly split equally into training and testing data and a random vector $\mathbf{a} := (a_0, a_1)$ was initialised. Given any $(x, y) \in \mathbb{R}^2$, we can classify (x, y) by checking whether $(x, y) \cdot \mathbf{a}$ is greater than or less than 0 and assigning it the corresponding label. For 400 iterations, a random coordinate from our training data was chosen and checked whether it was correctly classified; if it was wrongly classified the vector $\mathbf{a}$ would be adjusted by adding or subtracting a small multiple of the chosen coordinate (depending on which label it should be assigned). The percentage of correctly classified coordinates per iteration can be seen in Figure 1.

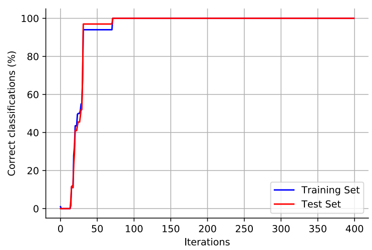


Figure 1: A graph of the percentage of coordinates that were successfully classified on each iteration of our algorithm.

The decision boundary is the set of coordinates (x, y) satisfying $(x, y) \cdot \mathbf{a} = 0$, which we can rearrange into $xa_0 + ya_1 = 0$, giving $y = \frac{-a_0}{a_1}x$. We can see this line plotted on the graph of Figure 2. All coordinates are correctly classified, as each coordinate above our boundary has label +1 and each coordinate below has label -1.

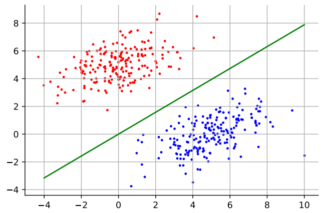


Figure 2: Graph showing the decision boundary. Red coordinates are from distribution 1 and blue coordinates are from distribution 2.

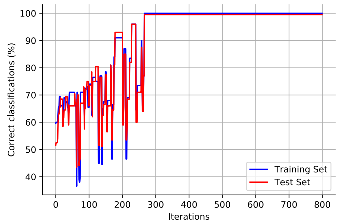
2 Adding a constant term

We were next tasked with solving the same problem with the means changed to $(2.5, 2.5)$ and $(10, 10)$ for distributions 1 and 2, respectively. This required our decision boundary to include a constant term, so the vector $\mathbf{a}$ was changed to (a_0, a_1, a_2). Instead of calculating $(x, y) \cdot \mathbf{a}$, we would then calculate $(x, y, 1), \cdot \mathbf{a}$, making our decision boundary the linear equation $xa_0 + ya_1 + a_2 = 0$. We can rearrange this into $y = \frac{-a_0}{a_1}x - \frac{a_2}{a_1}$, allowing us to plot the boundary as in Figure 3(b).

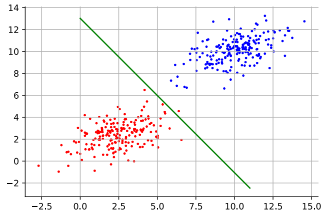
We can see that there is 1 incorrectly classified coordinate from distribution 1 in the graph of Figure 3(b). Figure 3(a) shows us that this coordinate must have been part of the test data, so our algorithm would not have accounted for it no matter how many iterations were run.

3 Further Data Set: Banknote Authentication

We can also apply our perceptron algorithm to higher-dimensional two-class classification problems. The algorithm was applied to a data set from [1], obtained at `http://archive.ics.uci.edu/ml/datasets/banknote+authentication`. This data has 1372 instances, each with 5 attributes:



(a)



(b)

Figure 3: Graphs for our revised problem showing (a) The percentage of correctly classified data per iteration. (b) The decision boundary.

0. variance of Wavelet Transformed image (continuous) **1.** skewness of Wavelet Transformed image (continuous) **2.** curtosis of Wavelet Transformed image (continuous) **3.** entropy of image (continuous) **4.** class (integer)

The first four attributes relate to properties of images of banknotes, and the last attribute is either 0 for a real banknote or 1 for a fake banknote (this was transformed into +1 and -1, respectively). Since there are more real banknotes in this data set than fake notes (762 vs 610), let us also calculate the F-measure on each iteration (as seen in Figure 4).

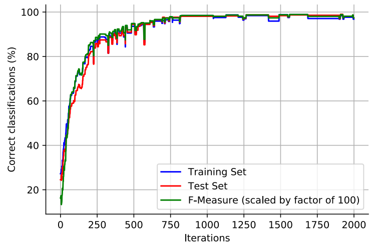
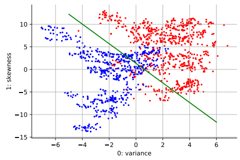
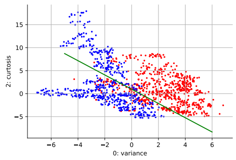


Figure 4: Percentage of correctly classified instances over time. The green line represents the $100\times$F-measure.

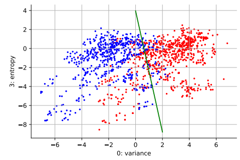
Since we have four attributes for training, we can graph the 2-dimensional projection of the decision boundary for any two variables. Since our boundary now lies on $y_0a_0 + y_1a_1 + y2a_2 + y_3a_3 + a_4 = 0$ (where y_i are the values of each attribute), we set all y_i equal to zero except for the two we wish to plot. The result of doing so can be seen in Figure 5 for every combination of any two attributes.



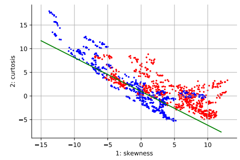
(a) (0,1)



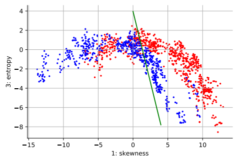
(b) (0,2)



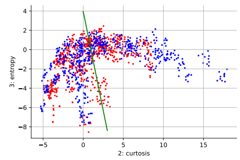
(c) (0,3)



(d) (1,2)



(e) (1,3)



(f) (2,3)

Figure 5: Two dimensional projections of the decision boundaries for every combination of two attributes. The caption (x, y) for each graph represents the two attributes x and y being plotted. Red coordinates indicate real banknotes, blue coordinates indicate fake.

References

[1] V. Lohweg, H. Dörksen, J. Hoffmann, R. Hildebrand, E. Gillich, J. Schaede, and J. Hofmann, "Banknote authentication with mobile devices," vol. 8665, 02 2013.