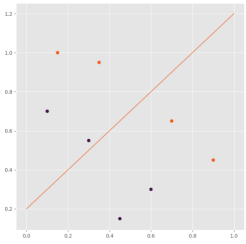
Amin Fadaee

Pattern Recognition HW6

Python 3.5, Libraries: Numpy, Matplotlib, Documentation: HTML, Bootstrap 4, jqMath

Problem 1

a. Here is the plot depicting the points and the initial separator:



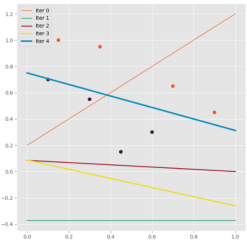
As can be seen, 4 points are misclassified using this line.

b. *(along with the next section)*

c. Using the following pseudocode we update the weight with single sample rule:

```
1. Create the augmented data (add a 1 column and multiple each row by its label)
2. For iter=1..t:
     For i=1..n:
       Compute w'y_i
       If w'y_i<0:
         w = w + rate * y_i
```

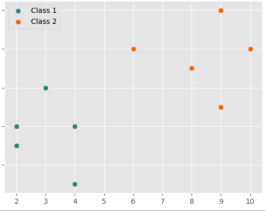
Using the procedure above (in `Problem 1.py`), rate=1 and 4 iterations we would derive the following plot:



As can be seen from the plot, the line in the 4th iteration correctly classifies all instances.

Problem 2

a. Here is a plot depicting the data:



b. Now let's compute the LDA projection line. We first need to compute the means of each class:

$$\mu_1 = [3, 3.6]^T$$
$$\mu_2 = [8.4, 7.6]^T$$

And now the scatter for each class:

$$S = \sum (x_i - \mu)(x_i - \mu)^T$$
$$S_1 = \begin{bmatrix} 4 & -2 \\ -2 & 13.2 \end{bmatrix}$$
$$S_2 = \begin{bmatrix} 9.2 & -0.2 \\ -0.2 & 13.2 \end{bmatrix}$$

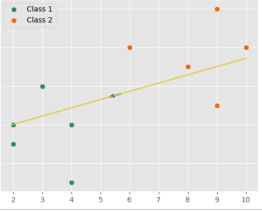
Now we need to find S_W and S_B

$$S_W = S_1 + S_2 = \begin{bmatrix} 13.2 & -2.2 \\ -2.2 & 26.4 \end{bmatrix}$$
$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T = \begin{bmatrix} 29.16 & 21.6 \\ 21.6 & 16 \end{bmatrix}$$

If S_W is full ranked, we can find the best projection vector v by the following formula:

$$v = S_W^{(-1)}(\mu_1 - \mu_2)$$
$$S_W^{(-1)} = \frac{1}{13.2 * 26.4 - 2.2^2} \begin{bmatrix} 26.4 & 2.2 \\ 2.2 & 13.2 \end{bmatrix}$$
$$v = \begin{bmatrix} 0.07682458 & 0.00640205 \\ 0.00640205 & 0.03841229 \end{bmatrix} [-5.4, -4]^T = [-0.44, -0.18]^T$$

Now if we plot this vector along with the projection line, the following figure is obtained:



c. Now let's project the data to this line. We can obtain the new data Y in the following way:

$$Y = v^T X$$

Therefore:

$Y_1 =$ `-1.94 -1.6 -1.42 -2.4 -2.48`

$Y_2 =$ `-5.76 -4.08 -4.86 -4.78 -5.84`

And here is a plot depicting these points:

d. As can be seen in the plot above, the data of each class are separated perfectly and there is no collision between them. Also as the margin between the two class is quite high (more than 1.5), the classification in this projection would have good results.

Problem 3

Since the decision function only depends on the support vectors, removing a nonsupport vector from the training data and then re-training an SVM would lead to the same decision function. Also, non-support vectors must be classified correctly. As a result, errors found in the leave-one-out validation must be caused by removing the support vectors, proving the desired result.