Elif Küçük
0040851
COMP 421
HW 02 Report.

COMP/INDR 421/521 HW 02: Multiclass Linear Discrimination – Report

Data generation in this project were done by the exact same function mvrnorm using the same mean and covariance matrixes that were used in the first homework.

We learned the linear discrimination rule using the softmax function for the multiclass classification problem.

We define **softmax** function which takes weight w and w0 along with the data point x to predict corresponding y value.

$$y_i = \hat{P}(C_i|x) = \frac{\exp\,[w_i^t x + w_{i0}]}{\sum_{j=1}^{K} \exp\,[\,w_j^t x + w_{j0}]}, i = 1,2 \ldots K \ where\ K = class\ number$$

In the **softmax(w,w0,x)** function we calculated softmax value for each class with the given data point and weights. Then the fumction returns column vector with these 3 values.

In order to find the correct weights for each attribute of each class type we initialize w and w0 matrices using runif function around 0 and define gradient_w and gradient_w0 functions for updating the weights.

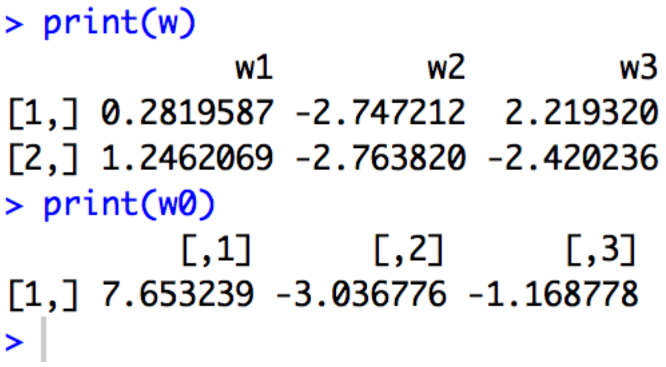
Gradient_w function takes y, y_pred and X to return updated w matrix of which each column corresponds to weight of each class.

$$gradient\_wj = stepSize * \sum_t (r_j^t - y_j^t)\, x^t$$

$$gradien_{wj0} = stepSize * \sum_t (r_j^t - y_j^t)$$

Similarly gradient_w0 function takes y, y_pred to return updated w0 matrix of which each column corresponds to weight0 of each class.

You can see the resulting w and w0 matrixes after the training data.



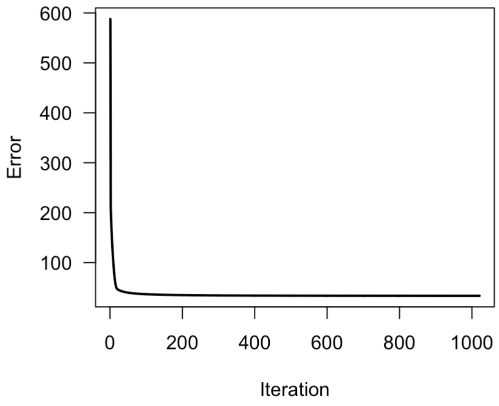
```
> print(w)
            w1        w2        w3
[1,] 0.2819587 -2.747212  2.219320
[2,] 1.2462069 -2.763820 -2.420236
> print(w0)
         [,1]      [,2]      [,3]
[1,] 7.653239 -3.036776 -1.168778
> 
```

Elif Küçük
0040851
COMP 421
HW 02 Report.

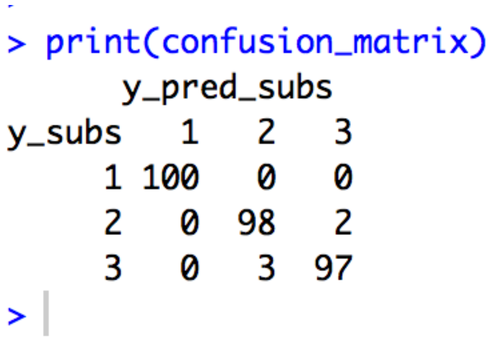
In order to train data I defined a while loop in which firstly each y_pred points are predicted using the softmax function and weight found in previous step.

Then using the gradient_w and gradient_w0 new weights are calculated according to y_pred generated. The while loop continues until weight converges.

In order to see the objective value chamge over time in the while loop we store each objective value in a vector and than make a graph of objective value vs iteration number.



In order to make a confusion matrix I defined y_pred_subs and y_subs which are 300*1 matrices that store the corresponding class value as 1, 2 or 3. Using the table function with y_pred_subs and y_subs I created the confusion matrix that you can see below.

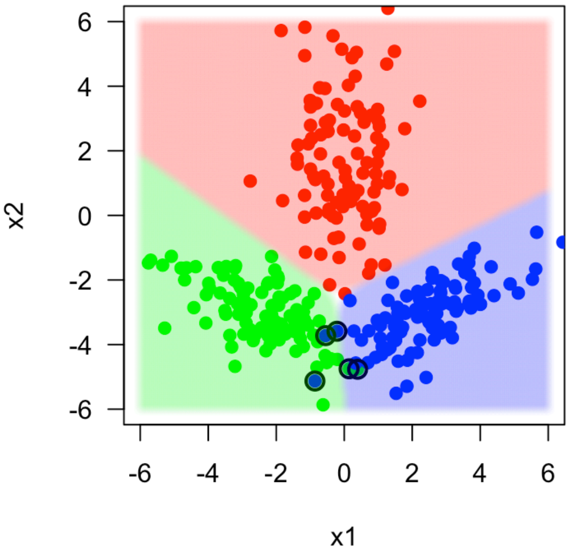


```
> print(confusion_matrix)
      y_pred_subs
y_subs   1   2   3
     1 100   0   0
     2   0  98   2
     3   0   3  97
> 
```

Elif Küçük
0040851
COMP 421
HW 02 Report.

After creating the y_pred_subs and y_subs drawing the decision boundaries were similar to the steps used in homework 1. You can find the commented code in the R file.

Here is thre graph with the decision boundaries.


6
4
2
0
-2
-4
-6
x2
-6
-4
-2
0
2
4
6
x1