Elif Küçük
0040851
COMP 421
HW 01 Report.

COMP/INDR 421/521 HW 01: Multivariate Parametric Classification - Report

In this project using the **mvrnorm** function of R, I generated random data points for three bivariate Gaussian densities with the given three mean matrix and three covariance matrix.

Using the **mean**, **cov** in-built function of R I estimated the sample mean and sample covariance matrix from the random generated data points.

In order to help me visualise each sample mean and sample covariance along with one general **sample_mean** matrix and **sample_covariance** matrix, I stored sample_mean1, sample_mean2, sample_mean3 and similarly sample_cov1, sample_cov2, sample_cov3 matrices.

class_priors was estimated as the ratio of data points belonging each class agains total data points.

You can see the print result in screen shots below.

```
> print(sample_means)
           [,1]      [,2]      [,3]
[1,] -0.1262465 -2.832581  2.673704
[2,]  1.4688810 -2.812404 -2.922399
> print(sample_covariance)
, , 1

          [,1]      [,2]
[1,] 0.9319606 0.1850652
[2,] 0.1850652 2.5167750

, , 2

          [,1]       [,2]
[1,]  1.754739 -0.7959760
[2,] -0.795976  0.9684586

, , 3

          [,1]      [,2]
[1,] 1.5325311 0.7512647
[2,] 0.7512647 1.1182344

> print(class_priors)
[1] 0.3333333 0.3333333 0.3333333
```

In order to create the confusion matrix :

1. We first defined the score function for all 3 classes using the parametric classification rule.

$$g_i(x) = x^t W_i x + W^T x + w_{i0}$$
$$W_i = -\frac{1}{2} S_i^{-1}$$
$$w_i = S_i^{-1} m_i$$

Elif Küçük
0040851
COMP 421
HW 01 Report.

$$w_{i0} = -\frac{1}{2} m_i^T{}_i S_i^{-1} m_i - \frac{1}{2}\log|S_i| + \log \hat{P}(C_i)$$

2. Then using the y and y_predicted and y values we increased the corresponding matrix point in the confusion matrix.

You can see the print result of confusion matrix in screen shots below.

```
> print(confusion_matrix)
     [,1] [,2] [,3]
[1,]   99    1    0
[2,]    0   99    2
[3,]    1    0   98
>
```

In order to draw decision boundaries.

1. First, to be able to find our way in plot, for each point in plot we mark corresponding **x1 grid points** and **x2 grid points** in the rectangular area.

2. For each point in the rectangular plot area using the score functions defined for class 1,2,3 **we define a function that estimates the class that point belongs to.**

3. Using the function f defined we estimate and store the **class_values** for each data point in 6 by 6 plot area.

4. We draw black circle around the data points in which y_predicted is not same with actual y values. By checking y_predictedlist values against actual y.

5. For each grid point take the points that have class_value 1,2 and 3 and paint them in corresponding color, red, green and blue, by choosing the grid point according to **class_values** matrix defined.

You can see the resulting plot in screen shot below.