Elif Küçük
0040851
COMP 421
HW 07 Report.

COMP/INDR 421/521 HW 07: EXPECTATION MAXIMIZATION CLUSTERING– Report

This file briefly explains each function and variable defined in the project. The defined function names and variables are written in bold. The code also has the comment lines to make understating easier.

Question 1

Using mvrnorm function we generate random data points from given Gausian densities.

You can find the generated sample in the Figure 1.

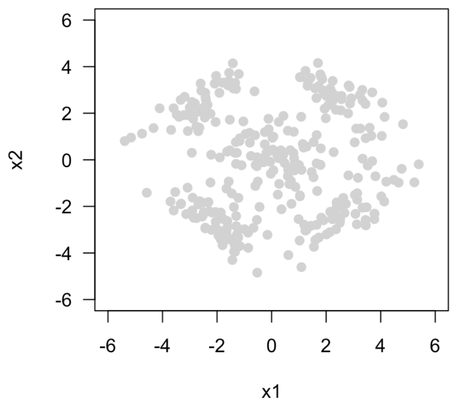


Figure 1

Question 2

To initialize our EM algorithm we need to run K-means clustering algorithm for 2 times.

To implement K-means algorithm I defined two methods:

updateAssignments: Assignment of each data point is updated according to closest centroids.
In order to initialize K-means algorithm I randomly choose 5 data points that will represent **centroids** (the mean point of each clustering) in the beginning.

assignments: is element of 1:5, represents the cluster data point belongs to. Vector of size N.

updateCentroid_K_Means: by checking the assignments of data sample updates the cluster means.

After running the K-means algorithm for 2 iteration the following assignment and centroids in Figure 2 are achieved.

Elif Küçük
0040851
COMP 421
HW 07 Report.

```
centroids<-X[sample(seq(1:N),5),]
my_plot(X,centroids, NULL)
updateAssignments()
updateCentroid_K_Means()
updateAssignments()
```

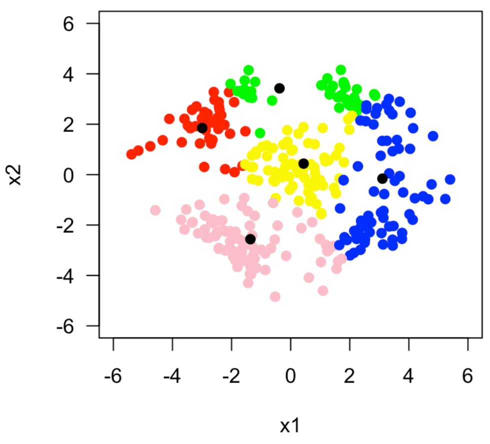


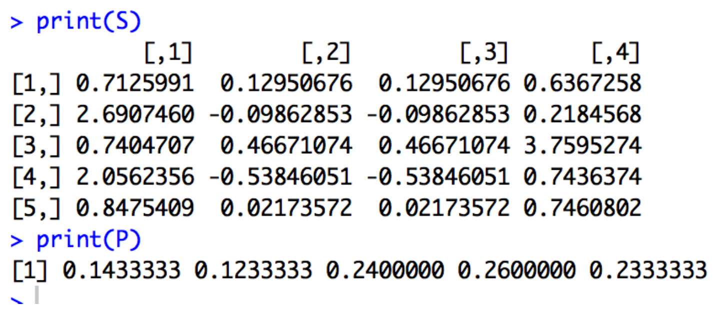
Figure 2

Question 3:

To be able to run EM algorithm, in S and P vectors we initialize covariance and prior probabilities using the assignments that were achieved in question 2.

S: covariance matrices. (5*4). Each row stores cov matrices for each cluster.

P: (5*1) Prior probabilities.



```
> print(S)
          [,1]        [,2]        [,3]      [,4]
[1,] 0.7125991  0.12950676  0.12950676 0.6367258
[2,] 2.6907460 -0.09862853 -0.09862853 0.2184568
[3,] 0.7404707  0.46671074  0.46671074 3.7595274
[4,] 2.0562356 -0.53846051 -0.53846051 0.7436374
[5,] 0.8475409  0.02173572  0.02173572 0.7460802
> print(P)
[1] 0.1433333 0.1233333 0.2400000 0.2600000 0.2333333
> 
```

Figure 3

Question 4:

In order to implement EM algorithm I defined two method E_Step and M_Step.

h:

E_Step: Updates the h (300*5 vector). h vector’s role is similar to assignments in K-Means.

$$h_i^t = \frac{\pi_i |S_i|^{-\frac{1}{2}} \exp\left[-\left(\frac{1}{2}\right)(x^t - m_i)^T S_i^{-1}(x^t - m_i)\right.}{\sum_j \pi_i |S_i|^{-\frac{1}{2}} \exp\left[-\left(\frac{1}{2}\right)(x^t - m_i)^T S_i^{-1}(x^t - m_i)\right.}$$

Also according to result of h, we also update Prior probability P.

Elif Küçük
0040851
COMP 421
HW 07 Report.

M_Step: Updates cluster means **centroids**, and cluster covariance **S**.

$$m_i^{l+1} = \frac{\sum_t h_i^t x^t}{\sum_t h_i^t}$$

$$S_i^{l+1} = \frac{\sum_t h_i^t \left(x^t - m_i^{l+1}\right)\left(x^t - m_i^{l+1}\right)^T}{\sum_t h_i^t}$$

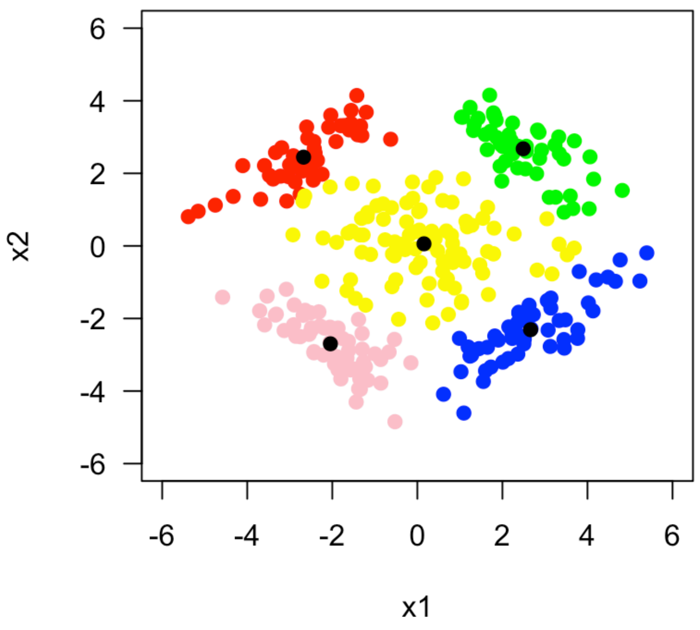


Figure 4

Question 5:

In question 5 to draw the Gaussian density I have used different approach than was suggested.

```
for(k in seq(1:K)){
mu     <- centroids[k,]
S_i       <- matrix(S[k,],2,2)
decom      <- chol(S_i)
t <- seq(0, 2*pi, length.out=200)
elips     <- 1 * cbind(cos(t), sin(t)) %*% decom
center_elips <- sweep(elips, 2, mu, "+")
lines(center_elips, type="l", lwd=2, asp=1)
}
```

By taking the cholesky decomposition of covariance and using the elips formula and by adding the centroids, we are able to find the elips that represents the distribution.

The final Figure can be seen in Figure 5.

Elif Küçük
0040851
COMP 421
HW 07 Report.

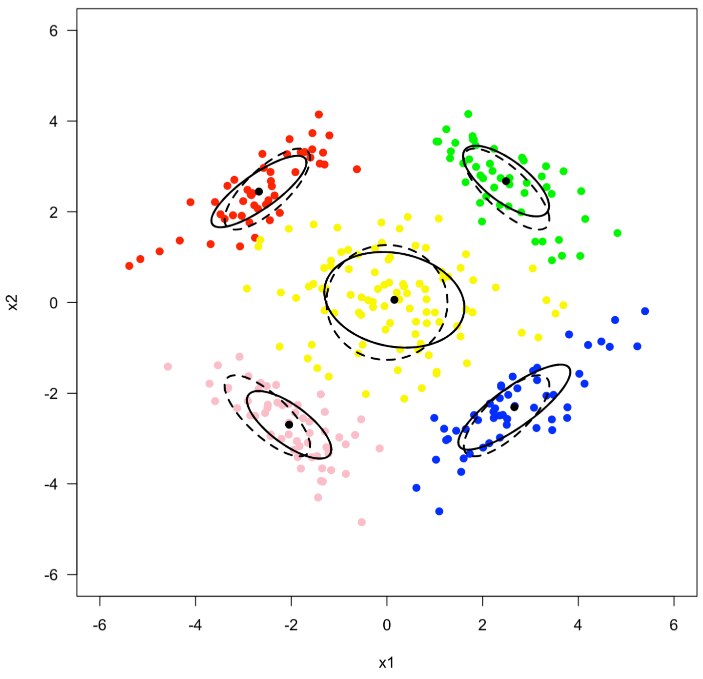


Figure 5

The solution to question 5 was taken from following website:
https://stats.stackexchange.com/questions/9898/how-to-plot-an-ellipse-from-eigenvalues-and-eigenvectors-in-r