Jared Yu

Module 11 Assignment

Generate 1,000 two-dimensional samples for each of two Gaussians, $p(\mathbf{x}|\omega_i) \sim N(\boldsymbol{\mu}_i, \Sigma_i)$ with

$$\boldsymbol{\mu}_1 = \begin{pmatrix}2\\0\end{pmatrix}, \boldsymbol{\mu}_2 = \begin{pmatrix}-2\\0\end{pmatrix}, \Sigma_1 = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}, \text{and } \Sigma_2 = PDP^{\top}$$

where

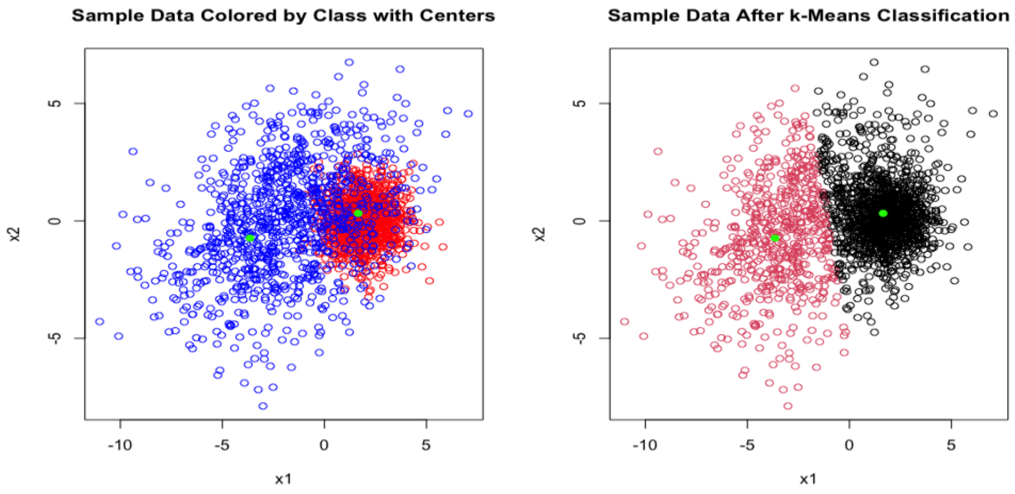
$$P = \frac{1}{2}\begin{pmatrix}\sqrt{3} & -1\\1 & \sqrt{3}\end{pmatrix} \text{ and } D = \begin{pmatrix}9 & 0\\0 & 4\end{pmatrix}.$$

Run k-means clustering, fuzzy k-means clustering, and Expectation-Maximization on these data. Plot the resulting membership sets for each run and briefly discuss the results.

<u>Ans:</u> Reference: [1]

The steps for the problem were to first generate the random samples of data. This was done in R, using the "mvtnorm" package. Once 1,000 samples for each of the two multivariate Gaussian distributions were generated, the following algorithms were applied: k-means clustering (KM), fuzzy k-means clustering (FKM), and Expectation-Maximization (EM). The corresponding packages used for each algorithm are: "stats," "e1071," and "mclust" respectively.

The results can be seen below in Figures 1-3. The left-side of the plots each show the two samples colored in red and blue. The red denotes class 1 and blue denotes class 2. Within the left and right side of each figure are two green dots which are the estimated centers of the two clusters. On the right side of each figure are the estimated classes according to each of the algorithms. On the right side, the black denotes the estimate for class 1 and red denotes the estimate for class 2.


Sample Data Colored by Class with Centers

Sample Data After k-Means Classification

Figure 1 The above figure shows on the left-hand side the generated data colored by their corresponding class and on the right-hand side the corresponding estimated classification by the k-means algorithm. The green dots denote the estimated centers for each of the two clusters.

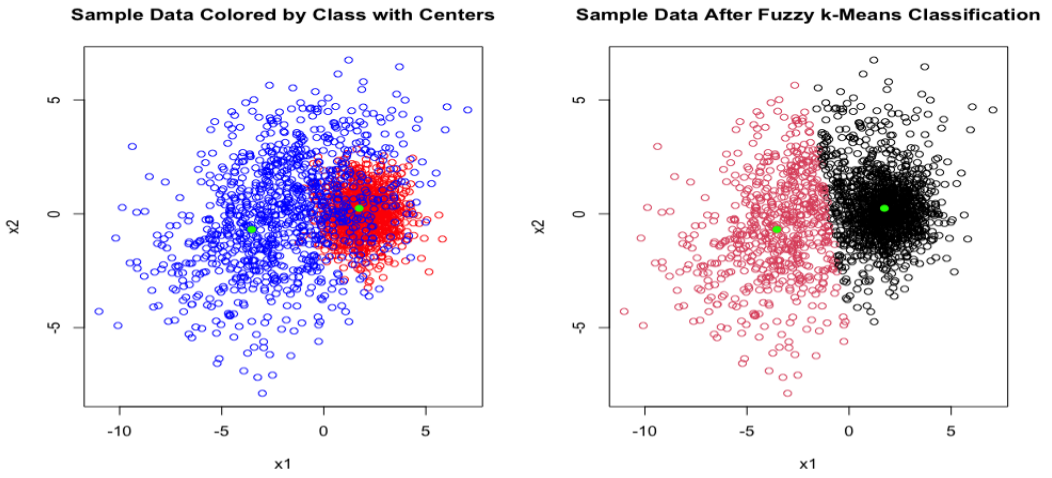


Figure 2 The above figure shows on the left-hand side the generated data colored by their corresponding class and on the right-hand side the corresponding estimated classification by the fuzzy k-means algorithm. The green dots denote the estimated centers for each of the two clusters.

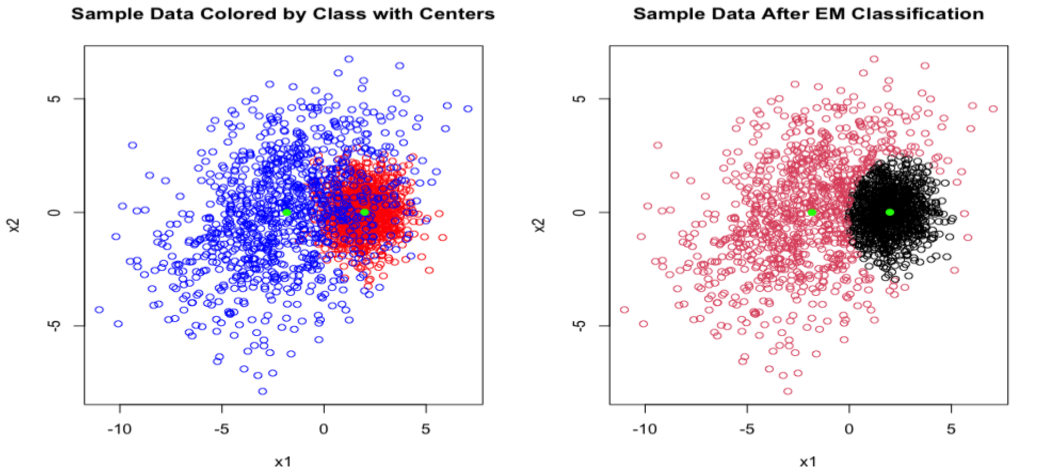


Figure 3 The above figure shows on the left-hand side the generated data colored by their corresponding class and on the right-hand side the corresponding estimated classification by the EM algorithm. The green dots denote the estimated centers for each of the two clusters.

From the three figures, they all look roughly similar. It is noticeable however that the EM algorithm generates a more curved boundary, while KM and FKM generate a more linear boundary. Based on this sample of data, it would seem that EM possibly has the advantage since there is considerable overlap between the two points of data.

We can look closer at the generated centers to see how similar they are. The below table (Table 1) shows the estimated center points for each of the three algorithms. It is apparent that the estimates generates by EM are much closer to the actual means. The difference is comparatively minor, while for KM and FKM the difference noticeably is larger.

Table 1 The below table shows the estimated centers for each of the clusters.

Algorithm	$\hat{\boldsymbol{\mu}}_1$	$\hat{\boldsymbol{\mu}}_2$
k-means	$(1.6537,0.3218)^\top$	$(-3.6461,-0.7312)^\top$
Fuzzy k-means	$(1.7306,0.2419)^\top$	$(-3.5342,-0.6873)^\top$
Expectation-Maximization	$(1.9858,0.0069)^\top$	$(-1.8262,-0.0044)^\top$

Another possibility is to calculate the 0-1 loss for each of the estimated classifications from each algorithm. The 0-1 loss here will be defined as follows,

$$L(\hat{\mathbf{y}},\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} I(\hat{y}_i \neq y_i),$$

where $I(\cdot)$ is an indicator function used to output 1 if the estimate is false and 0 otherwise. The result is averaged over all n observations. The 0-1 loss for each algorithm can be seen below in Table 2. It seems from the results that EM perform a more accurate classification, leading to fewer misclassified examples on average. The difference is also noticeable between EM and the other two KM and FKM algorithms. The EM leads to almost half as many errors as the other two.

Table 2 The table below shows the 0-1 loss function from each of the algorithms.

Algorithm	$L(\hat{\mathbf{y}},\mathbf{y})$
k-means	0.1975
Fuzzy k-means	0.193
Expectation-Maximization	0.1005

The conclusion then is that from this generated sample of data that EM outperforms the other KM and FKM algorithms. This is based on looking at the estimated centers along with the 0-1 loss comparison for the algorithms. In each of the cases, the difference was noticeable and seemed significant. It is likely that with an alternate distribution of data that the difference would not be as noticeable. However, generating more samples would not likely improve the performance for either KM or FKM.

Reference:

[1] https://piazza.com/class/kc0jkwru805u1?cid=199

Code Appendix:

```
# Load libraries
library(mvtnorm); library(mclust); library(e1071)

# Initialize variables
mu_1 <- matrix(c(2,0), nrow=2)
mu_2 <- matrix(c(-2,0), nrow=2)
sigma_1 <- matrix(c(1, 0, 0, 1), nrow=2)
P <- (1 / 2) * matrix(c(sqrt(3), 1, -1, sqrt(3)), ncol=2)
D <- matrix(c(9, 0, 0, 4), nrow=2)
sigma_2 <- P %*% D %*% t(P)

set.seed(1) # Generate sample data
n <- 1e3
sample_1 <- rmvnorm(n = n, mean = mu_1, sigma = sigma_1)
sample_2 <- rmvnorm(n = n, mean = mu_2, sigma = sigma_2)
sample_data <- rbind(sample_1, sample_2)
```

```
# k-means
par(mfrow = c(1,2))
k_mean_clustering <- kmeans(x = sample_data, centers = 2)
plot(x = sample_data[,1], y = sample_data[,2],
    col = c(rep('red', 1e3), rep('blue', 1e3)),
    main = 'Sample Data Colored by Class with Centers',
    xlab = 'x1', ylab = 'x2')
points(k_mean_clustering$centers, col = 'green', pch = 19)

plot(x = sample_data[,1], y = sample_data[,2],
    col=k_mean_clustering$cluster,
    main = 'Sample Data After k-Means Classification',
    xlab = 'x1', ylab = 'x2')
points(k_mean_clustering$centers, col = 'green', pch = 19)

# Error rate 0.1975
sum(k_mean_clustering$cluster != c(rep(1, 1e3), rep(2, 1e3))) / 2e3

# fuzzy k-means
fuzzy_k_mean_clustering <- cmeans(x = sample_data, centers = 2)
plot(x = sample_data[,1], y = sample_data[,2],
    col = c(rep('red', 1e3), rep('blue', 1e3)),
    main = 'Sample Data Colored by Class with Centers',
    xlab = 'x1', ylab = 'x2')
points(fuzzy_k_mean_clustering$centers, col = 'green', pch = 19)

plot(x = sample_data[,1], y = sample_data[,2],
    col=fuzzy_k_mean_clustering$cluster,
    main = 'Sample Data After Fuzzy k-Means Classification',
    xlab = 'x1', ylab = 'x2')
points(fuzzy_k_mean_clustering$centers, col = 'green', pch = 19)

# Error rate 0.193
sum(fuzzy_k_mean_clustering$cluster != c(rep(1, 1e3), rep(2, 1e3))) / 2e3

# EM
em_clustering = Mclust(data = sample_data, G = 2)
plot(x = sample_data[,1], y = sample_data[,2],
    col = c(rep('red', 1e3), rep('blue', 1e3)),
    main = 'Sample Data Colored by Class with Centers',
    xlab = 'x1', ylab = 'x2')
points(t(em_clustering$parameters$mean), col = 'green', pch = 19)
plot(x = sample_data[,1], y = sample_data[,2],
    col=em_clustering$classification,
    main = 'Sample Data After EM Classification',
    xlab = 'x1', ylab = 'x2')
points(t(em_clustering$parameters$mean), col = 'green', pch = 19)

# Error rate 0.1005
sum(em_clustering$classification != c(rep(1, 1e3), rep(2, 1e3))) / 2e3

# Center comparison
round(k_mean_clustering$centers, 4)
round(fuzzy_k_mean_clustering$centers, 4)
round(t(em_clustering$parameters$mean), 4)
```