Homework 2
Applied Machine Learning
Fall 2017
CSCI-P 556/INFO-I 526

Ashay H Sawant
ahsawant@umail.iu.edu

October 6, 2017

"All work herein is solely mine."

Problem 1 [20 points]

From textbook, Chapter 10 exercise 2 (Page 414).
Answer:
From the given dissimilarity matrix, first, let's compute the distance matrix for better understanding and calculation.

```
dissimilarityMatrix = matrix(c(0, 0.3, 0.4, 0.7, 0.3, 0, 0.5, 0.8,
0.4, 0.5, 0.0, 0.45, 0.7, 0.8, 0.45, 0.0), nrow=4)
distanceMatrix = as.dist(dissimilarityMatrix)
```

	1	2	3	4
1	0	0	0	0
2	0.30	0	0	0
3	0.40	0.50	0	0
4	0.70	0.80	0.45	0

1. Sketch dendrogram that results from hierarchically clustering above observations using complete linkage.

   Complete linkage is the distance between members that are farthest apart. So, every time we'll look for the minimum value in distance matrix and cluster the respective elements. After clustering two element, we'll update distance matrix to work further.

   Iteration 1:
   0.30 is the smallest element in the distanceMatrix, hence we'll cluster elements (1,2).
   To update the distanceMatrix for point 3, we'll find
   MAX(dist((P1,P3),(P2,P3))) = MAX(0.40,0.50) = 0.50
   Similarly, we can update distanceMatrix in following iterations.
   Let's update the distanceMatrix.

	(1,2)	3	4
(1,2)	0	0	0
3	0.50	0	0
4	0.80	0.45	0

   Iteration 2:
   0.45 is the smallest element in the distanceMatrix, hence we'll cluster elements (3,4).

   Let's update the distanceMatrix.

	(1,2)	(3,4)
(1,2)	0	0
(3,4)	0.80	0

   Iteration 3:
   0.80 is the smallest and final element in the distanceMatrix, hence we'll cluster elements ((1,2),(3,4)).

Figure 1 shows the Dendrogram of above calculations for hierarchical clustering with complete linkage.

We can use R code for hierarchical clustering with complete linkage.

```
plot(hclust(distanceMatrix, method="complete"))
```

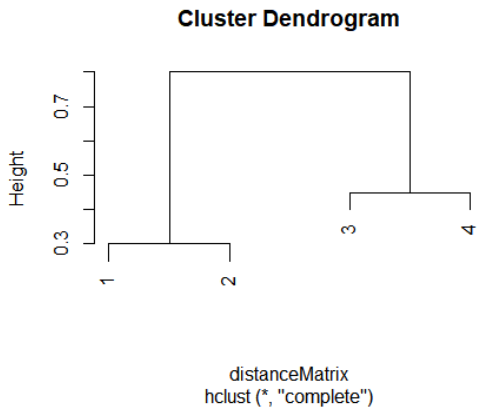

Figure 1: Hierarchical Clustering (Complete Linkage)

2. Sketch dendrogram that results from hierarchically clustering above observations using single linkage.

   Single linkage is the distance between closest members of two clusters. So, every time we'll look for the minimum value in distance matrix and cluster the respective elements. After clustering two element, we'll need to update distance matrix to work further.

   Iteration 1:
   0.30 is the smallest element in the distanceMatrix, hence we'll cluster elements (1,2).
   To update the distanceMatrix for point 3, we'll find
   MIN(dist((P1,P3),(P2,P3))) = MIN(0.40,0.50) = 0.40
   Similarly, we can update distanceMatrix in following iterations.
   Let's update the distanceMatrix.

	(1,2)	3	4
(1,2)	0	0	0
3	0.40	0	0
4	0.70	0.45	0

   Iteration 2:
   0.40 is the smallest element in the distanceMatrix, hence we'll cluster elements ((1,2),3).

   Let's update the distanceMatrix.

	((1,2),3)	4
((1,2),3)	0	0
4	0.45	0

Iteration 3:
0.45 is the smallest and final element in the distanceMatrix, hence we'll cluster elements (((1,2),3),4).

Figure 2 shows the Dendrogram of above calculations of hierarchical clustering with single linkage.

We can use R code for hierarchical clustering with single linkage.

```
plot(hclust(distanceMatrix, method="single"))
```

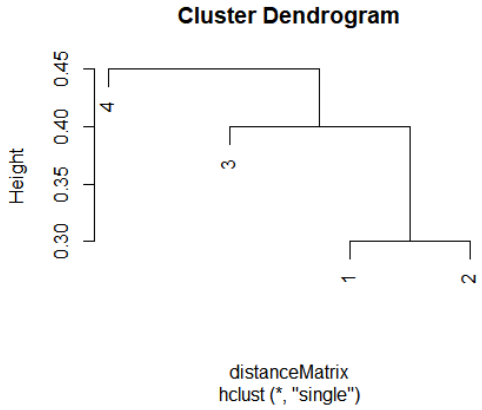

Figure 2: Hierarchical Clustering (Single Linkage)

3. Suppose that we cut the dendrogram obtained in 1 (complete linkage) such that two clusters result. Which observations are in each cluster?

   - (1,2) and (3,4)

4. Suppose that we cut the dendrogram obtained in 2 (single linkage) such that two clusters result. Which observations are in each cluster?

   - ((1,2),3) and (4)

5. It is mentioned in the chapter that at each fusion in the dendrogram, the position of the two clusters being fused can be swapped without hanging the meaning of the dendrogram. Draw a dendrogram that is equivalent to the dendrogram in (a), for which two or more of the leaves are repositioned, but for which the meaning of the dendrogram is the same.

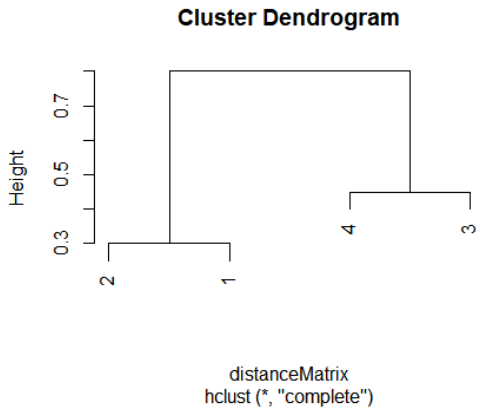

Figure 3: Hierarchical Clustering (Complete Linkage) with repositioned leaves

Problem 2 [50 points]

Implement expectation-maximization algorithm for Gaussian mixture models (see the EM algorithm below) in R and call this program G_k. As you present your code explain your protocol for

3.1 Initializing each Gaussian:
- Expectation Maximization algorithm is parametric algorithm. That means, we assume parameters such as k Gaussian models and prior probability for every Gaussian model before preceding with algorithm. Each Gaussian model is characterized by mean μ and variance ϵ.
- In EM algorithm, we initialize k number of means μ, as random sample from data set and with every iteration, we re-estimate the mean.

```
meanMatrix = ionosphere_matrix[sample(nrow(ionosphere_matrix),size=k,replace=TRUE),]
```

- Instead of randomly initializing mean μ as uniform distribution, we can induce non-uniform distribution method like k-means++ to select initial means, to improve the prediction.
- Since, we will be dealing with multidimensional data, we will use co-variance matrix for each Gaussian model.
- I have initialized it to Identity matrix of (d*d) dimensions.

```
covarianceMatrix = rep(list(data.matrix(diag(dimension))), k)
```

- Finally, we have to initialize prior probabilities considering equally likely chances of a point belonging to k Gaussians. priors = (1/k)

```
priors = rep((1 / k), k)
```

3.2 Maintaining k Gaussian:
After initializing parameters, we'll perform 2 steps to maintain Gaussians.

(a) Expectation step: We will find the posterior probability for every data point with respect to every Gaussian model(W_ij).

```
for (i in 1:k) {
for (j in 1:nrow(data_matrix)) {
weightedProbability[j, i] = findWeightedProbability(j,i,data_matrix,clusterMean,
covarianceMatrix,priors) + 0.00000005
}
}

findWeightedProbability = function(j,i,data_matrix,clusterMean,
covarianceMatrix,priors) {
mvnd_currentCluster = 0
mvnd_allCluster = 0
for (a in 1:nrow(clusterMean)) {
mvnd = multivariateNormalDistribution(ringnorm_matrix[j, ],
clusterMean[a, ],covarianceMatrix[[a]],priors[a])
```

```
if (a == i) {
mvnd_currentCluster = mvnd
}
mvnd_allCluster = mvnd + mvnd_allCluster
}
f = mvnd_currentCluster / mvnd_allCluster
return(f)
}
```

(b) Maximization step: In Maximization part we'll re-estimate the Gaussian mean μ, Co-variance matrix and Prior probabilities.

```
#Re-estimate cluster means
for (i in 1:k) {
clusterMean_new[i, ] = updateClusterMean(i, weightedProbability,ringnorm_matrix)
}

#Re-estimate Covariance Matrix
for (i in 1:k) {
temp = cov.wt(ringnorm_matrix,weightedProbability[, i],clusterMean_new[i, ])
covarianceMatrix[[i]] = temp$cov
}

#Re-estimate priors
for (i in 1:k) {
priors[i] = updatePriors(i, weightedProbability, nrow(ringnorm_matrix))
}

updateClusterMean = function(i, weightedProbability, ringnorm_matrix) {
numerator = sum(weightedProbability[, i] * ringnorm_matrix[i, ])
denominator = sum(weightedProbability[, i])
return(numerator / denominator)
}



updatePriors = function(i, weightedProbability, n) {
s = sum(weightedProbability[, i])
p = s / n
return(p)
}
```

3.3 Deciding ties:
EM Algorithm is soft clustering algorithm, that means every data point is assigned a probability in terms of likelyhood of a point to be in particular cluster. There is very rare chance that this probability to be equal for all k-clusters. In this case, I'm assigning that point to first cluster.

3.4 Stopping criteria: After estimating Gaussian models, I'm computing the squared distance between old Gaussian Mean(μ_{old}) and new Gaussian Mean(μ_{new}), and calling it as a threshold to stop my algorithm.

Problem 3 [70 points]

In this questions, you are asked to run your program, G_k, against the Ringnorm and Ionosphere data sets and compare G_k with C_k (k-means algorithm from previous homework). Click on the below links to download the data sets.

- Ringnorm Data Set
- Ionosphere Data Set

Answer the following questions:

3.1 Initialize G_k and C_k with the same set of initial points (initial centroids for C_k and μ_i-s for G_k are identical) and run them for $k = 2, \ldots, 5$ for 20 runs each. Report error rates and iteration counts for each k using whisker plots that reveal comparison of C_k and G_k. An example of whisker plot is given below. A simple error rate can be calculated as follows:

- If $k = 2$: C_k and G_k will predict two clusters. Error calculation is trivial for two clusters.
- If $k > 2$: after C_k and G_k converge, combine the clusters as follows to ended up with two clusters: since the true clusters are known for a given arbitrary blocks number, final clusters are determined by measuring the Euclidean (this is the easiest choice) distances between true cluster centers and predicted cluster centers.

In other words, you will always calculate the error for $k = 2$ since there are only 2 clusters in the given data sets. Below is an example of error calculation for Ionosphere data set. You can similarly calculate an error rate for Ringnorm data set.

For each centroid C_i, and each Gaussian G_k form two counts (over Ionosphere Data Set) :

$$g_i \leftarrow \sum_{\delta \in c_i.B} [\delta.C = \text{“g”}], \quad \text{good}$$
$$b_i \leftarrow \sum_{\delta \in c_i.B} [\delta.C == \text{“b”}], \quad \text{bad}$$

where $[x = y]$ returns 1 if True, 0 otherwise. For example, $[2 = 3] + [0 = 0] + [34 = 34] = 2$

The centroid C_i and Gaussian G_k is classified as good if $g_i > b_i$ and bad otherwise. We can now calculate a simple error rate. Assume C_i is good. Then the error is:

$$error(C_i) = \frac{b_i}{b_i + g_i} \quad [\text{same for error}(G_i)]$$

We can find the total error rate easily:

$$Error(\{C_1, C_2\}) = \sum_{i=1}^{2} error(C_i)$$

Discuss your results, i.e., which one performs better.

Answer:

(a) Ionosphere Dataset:
Figure 4 depicts the comparison between EM Algorithm and K-means Algorithm in terms of error in clustering over Ionosphere Dataset. As we can see in boxplot, for all of the values of k, error in EM Algorithm is much less than error in K-means Algorithm.

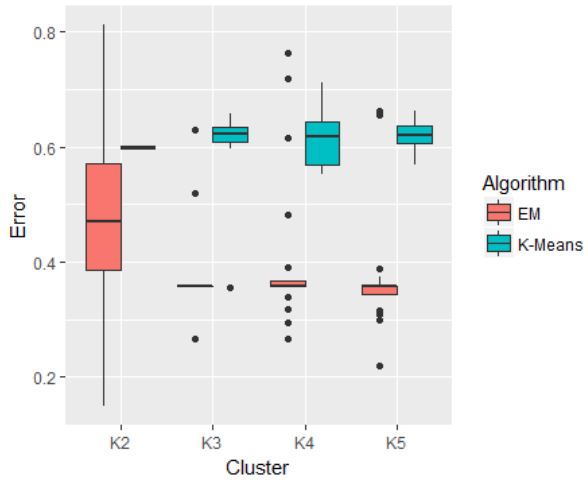

Figure 4: Boxplot of Error for k=2,3,..5 in EM and K-means Algorithm

Figure 5 depicts the comparison between EM Algorithm and K-means Algorithm in terms of number of iterations while clustering over Ionosphere Dataset. As we can see in boxplot, for all of the values of k, number of Iterations in EM Algorithm is much less than number of Iteration in K-means Algorithm.

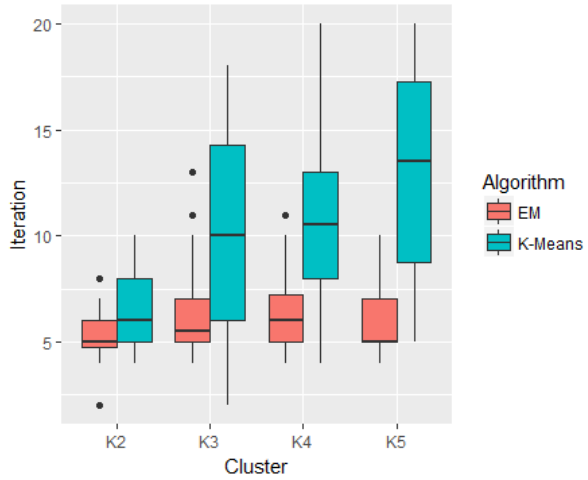

Figure 5: Boxplot of Iterations for k=2,3,..5 in EM and K-means Algorithm

(b) Ringnorm Dataset:
Figure 6 depicts the comparison between EM Algorithm and K-means Algorithm in terms of error in clustering over Ringnorm Dataset. As we can see in boxplot, for all of the values of k, error in EM Algorithm is comparitively more than error in K-means Algorithm.

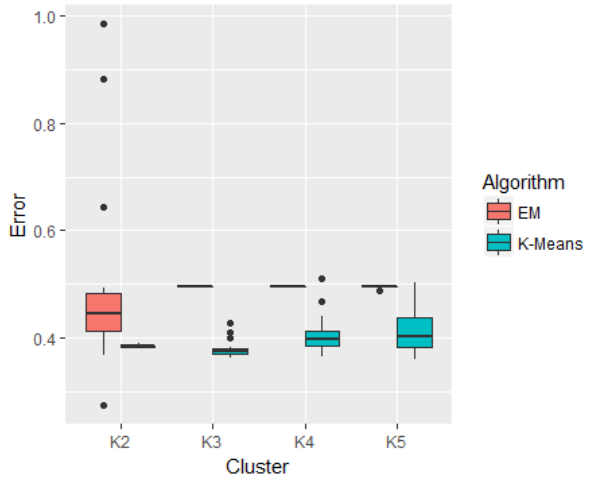

Figure 6: Boxplot of Error for k=2,3,..5 in EM and K-means Algorithm

Figure 7 depicts the comparison between EM Algorithm and K-means Algorithm in terms of number of iterations while clustering over Ringnorm Dataset. As we can see in boxplot, for all of the values of k, number of Iterations in EM Algorithm is much less than number of Iteration in K-means Algorithm.

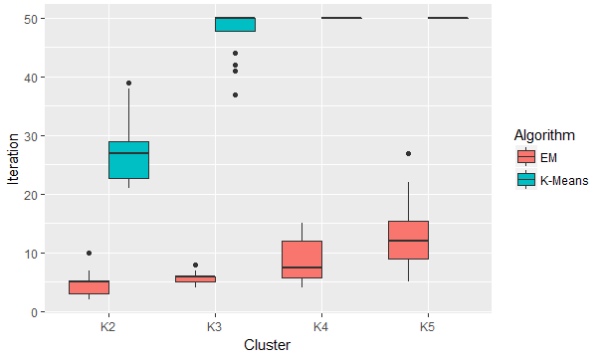

Figure 7: Boxplot of Iterations for k=2,3,..5 in EM and K-means Algorithm

3.2 In this question, we will run your G_k with fixing the variances to ones and the priors to be uniform. Do not update the variances and priors throughout iterations. As explained in question 3.1, compare your new G_k and C_k using whisker plots. Discuss your results, i.e., which one performed better.
Answer:

(a) Ionosphere Dataset:
Figure 8 depicts the comparison between EM Algorithm and K-means Algorithm in terms of error in clustering over Ionosphere Dataset. As we can see in boxplot, for all of the values of k, error in EM Algorithm is much less than error in K-means Algorithm.

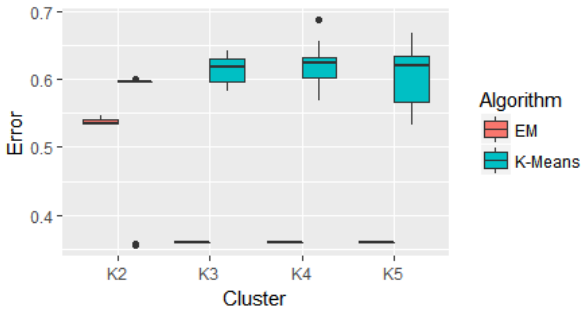

Figure 8: Boxplot of Error for k=2,3,..5 in EM and K-means Algorithm

Figure 9 depicts the comparison between EM Algorithm and K-means Algorithm in terms of number of iterations while clustering over Ionosphere Dataset. As we can see in boxplot, for all of the values of k, number of Iterations in EM Algorithm is much less than number of Iteration in K-means Algorithm.

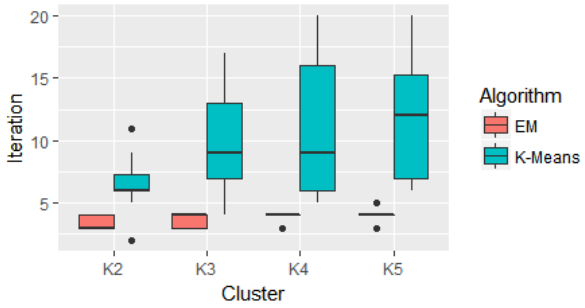

Figure 9: Boxplot of Iterations for k=2,3,..5 in EM and K-means Algorithm

(b) Ringnorm Dataset:
Figure 10 depicts the comparison between EM Algorithm and K-means Algorithm in terms of error in clustering over Ringnorm Dataset. As we can see in boxplot, for all of the values of k, error in EM Algorithm is comparitively more than error in K-means Algorithm.

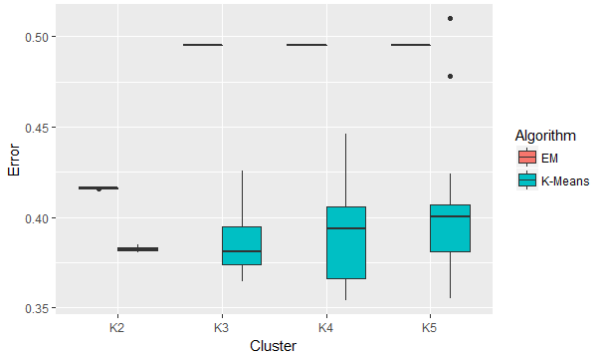

Figure 10: Boxplot of Error for k=2,3,..5 in EM and K-means Algorithm

Figure 11 depicts the comparison between EM Algorithm and K-means Algorithm in terms of number of iterations while clustering over Ringnorm Dataset. As we can see in boxplot, for all of the values of k, number of Iterations in EM Algorithm is much less than number of Iteration in K-means Algorithm.

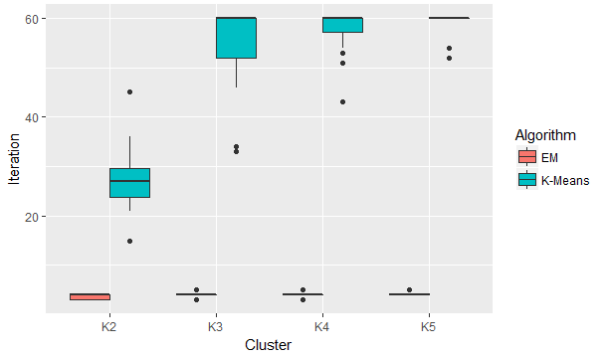

Figure 11: Boxplot of Iterations for k=2,3,..5 in EM and K-means Algorithm

From the above boxplots, we can conclude that by re-estimating co-variance and prior probabilities there is not much significant change in observation than not doing it.

Problem 4 [50 points]

In this question, you will first perform principal component analysis (PCA) over Ionosphere and Rignorm data sets and then cluster the reduced data sets using G_k (from question 3.1) and C_k. You are allowed to use R packages for PCA. Ignore the class variables (35th and 1st variables for Ionosphere and Ringnorm data sets, respectively) while performing PCA. Answer the questions below:

4.1 Make a scatter plot of PC1 and PC2 for both data sets. Discuss principal components (The first and second principal components). What are PC1 and PC2?
Answer:
Principal Components Analysis (PCA) is a statistical procedure to convert possibly co-related attributesof a dataset, into set of linearly uncorrelated variables called Principal Component.

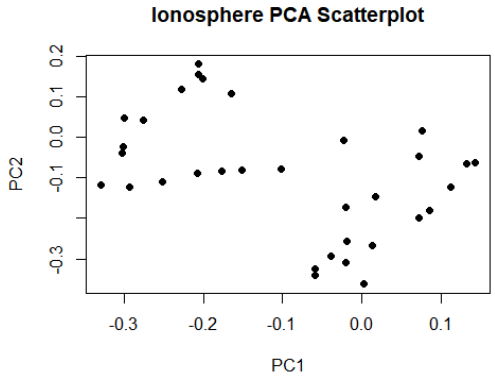

Figure 12: Scatterplot of PC1 and PC2 for Ionosphere Dataset

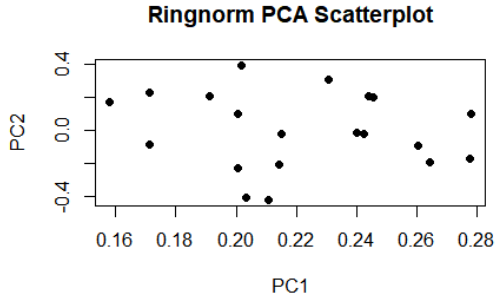

Figure 13: Scatterplot of PC1 and PC2 for Ringnorm Dataset

4.2 Create scree plots after PCA and explain the plots.
Answer:
- A screepolt displays the proportion of total variation in dataset that is explained by each of the components of in Principal Component Analysis. It is a method to determine optimal number of components useful for describing data in multidimensional scaling.
- From the screeplot below, we can conclude that Principal Components PC1 - PC7 are enough to describe data in multidimensional scaling.

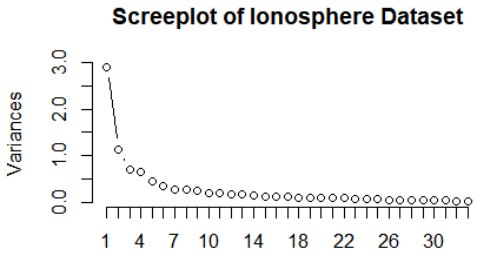

Figure 14: Screeplot of Ionosphere Dataset

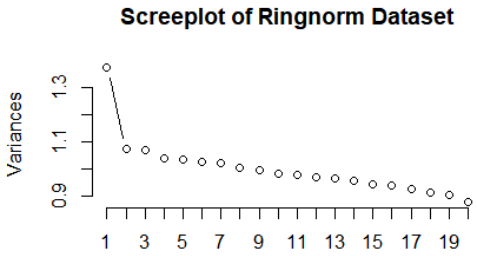

Figure 15: Screeplot of Ringnorm Dataset

4.3 Observe the loadings using prcomp() or princomp() functions in R and discuss loadings in PCA?i.e., how are principal components and original variables related?
Answer:
Loadings in PCA are basically the co-relation coeffiecients between rows and columns in dataset. Additionally, sum of squares within each component are eigen values.

4.4 Keep 90% of variance after PCA and reduce Ionosphere and Rignorm data sets. Run C_k and G_k with the reduced data sets and compare them using whisker plots as shown in question 3.1
Answer:

(a) Ionosphere Dataset:
Figure 16 depicts the comparison between EM Algorithm and K-means Algorithm in terms of error in clustering over reduced Ionosphere Dataset with the help of PCA algorithm. As we can see in boxplot, for all of the values of k, error in EM Algorithm is much less than error in K-means Algorithm.

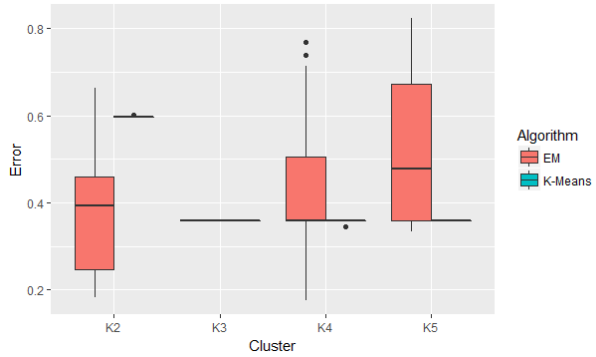

Figure 16: Boxplot of Error for k=2,3,..5 in EM and K-means Algorithm

Figure 17 depicts the comparison between EM Algorithm and K-means Algorithm in terms of number of iterations while clustering over reduced Ionosphere Dataset with the help of PCA algorithm. As we can see in boxplot, for all of the values of k, number of Iterations in EM Algorithm is much less than number of Iteration in K-means Algorithm.

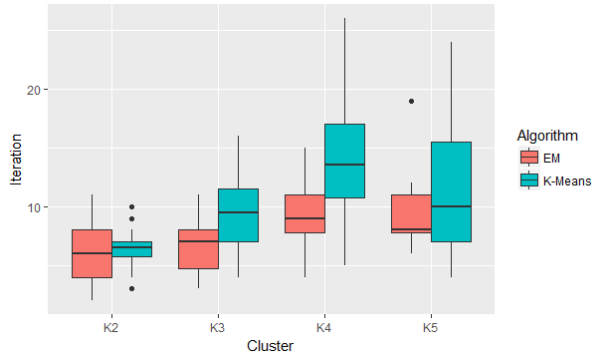

Figure 17: Boxplot of Iterations for k=2,3,..5 in EM and K-means Algorithm

(b) Ringnorm Dataset:
Figure 18 depicts the comparison between EM Algorithm and K-means Algorithm in terms of error in clustering over reduced Ringnorm Dataset with the help of PCA algorithm. As we can see in boxplot, for all of the values of k, error in EM Algorithm is comparitively more than error in K-means Algorithm.

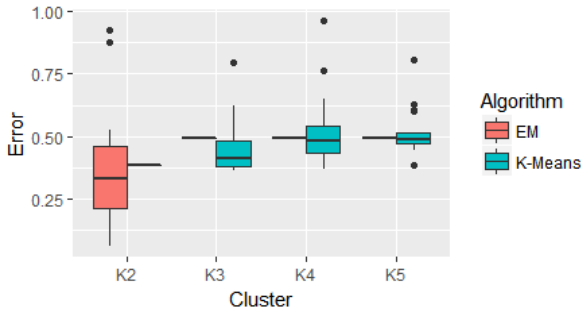

Figure 18: Boxplot of Error for k=2,3,..5 in EM and K-means Algorithm

Figure 19 depicts the comparison between EM Algorithm and K-means Algorithm in terms of number of iterations while clustering over reduced Ringnorm Dataset with the help of PCA algorithm. As we can see in boxplot, for all of the values of k, number of Iterations in EM Algorithm is much less than number of Iteration in K-means Algorithm.

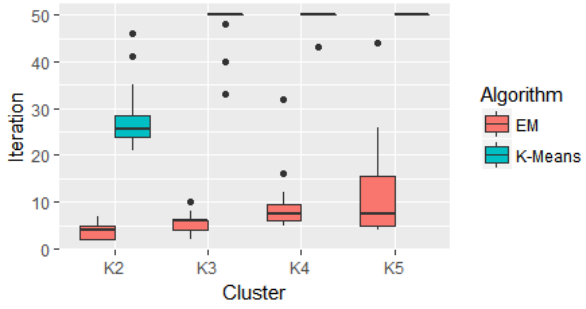

Figure 19: Boxplot of Iterations for k=2,3,..5 in EM and K-means Algorithm

4.5 Discuss that how PCA affects the performance of C_k and G_k.
Answer:
After applying PCA algorithm over dataset to reduce the dimentions of dataset based on 90% variance, there is not much significant change in the result. Though number of iteration taken by EM Algorithm to converge is much less than K-means algorithm. But at the same time, Error in clustering by EM algorithm is slightly greater than mean error caused by K-means Algorithm.

Problem 5 [50 points]

Randomly choose 50 points from Ionosphere data set (call this data set I_{50}) and perform hierarchical clustering. You are allowed to use R packages for this question. (Ignore the class variable while performing hierarchical clustering.)

5.1 Using hierarchical clustering with complete linkage and Euclidean distance cluster I_{50}. Plot the dendrogram.
Answer:

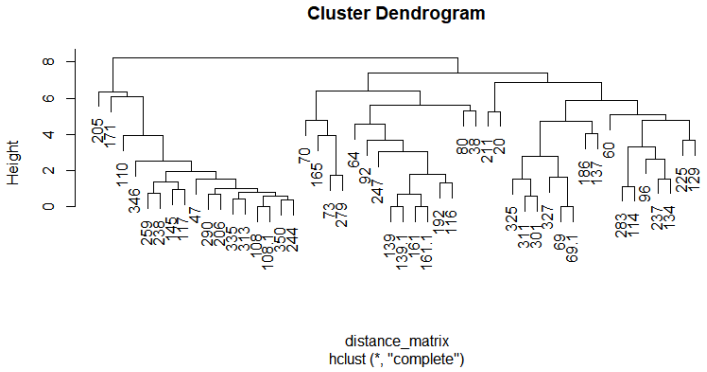

Figure 20: Dendrogram of I_{50} dataset

5.2 Cut the dendrogram at a height that results in two distinct clusters. Calculate an error rate.
Answer:
Total Error before performing PCA : 0.3061224

5.3 First, perform PCA on I_{50} (Keep 90% of variance). Then hierarchically cluster the reduced data using complete linkage and Euclidean distance. Plot the dendrogram
Answer:

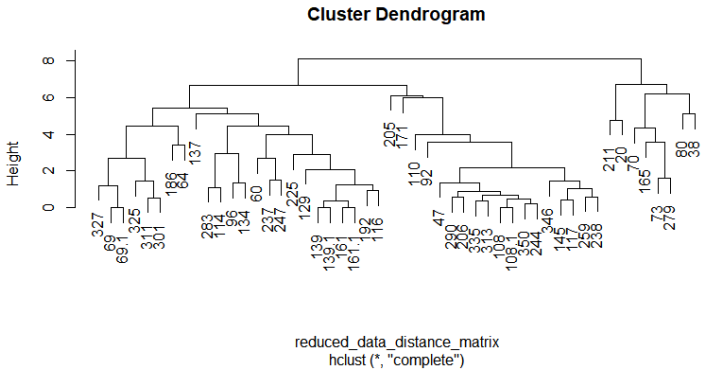

Figure 21: Dendrogram of reduced I_{50} dataset after PCA

5.4 Cut the dendrogram at a height that results in two distinct clusters. Calculate an error rate. How did PCA affect hierarchical clustering?

Answer:

Total Error after performing PCA : 0.3061224

Since, the error before and after performing PCA is exact same. PCA did not affect hierarchical clustering.

Extra credit [60 points]

This part is optional.

1 Improve the EM algorithm through initialization. k-means ++ is an extended k-means clustering algorithm and induces non-uniform distributions over the data that serve as the initial centroids. Read the paper and implement this idea to improve your G_k program (from question 3.1). Run your new G_k and old one (question 3.1) for $k = 2, \ldots, 5$ and compare the results using whisker plots. [30 points]
Answer:
- K-means ++ is the extension of traditional k-means clustering algorithm which focuses on centroid initialization with non-uniform distribution.
- Figure 22 depicts the comparison between EM Algorithm with centroid initialization as uniform distribution and K-means++ (non-uniform distribution) initialization in terms of error in clustering over Ionosphere Dataset. As we can see in boxplot, for all of the values of k, error in both the cases is almost same.

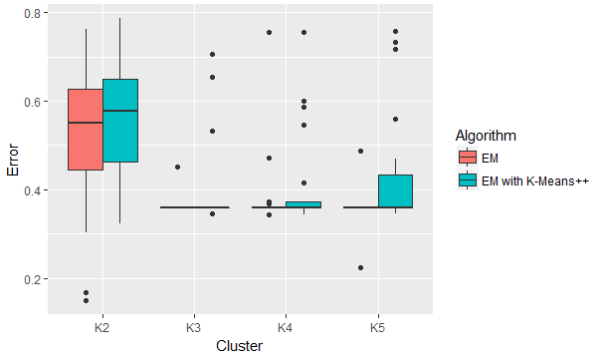

Figure 22: Boxplot of Error for k=2,3,..5 in normal and K-means++ centroid initialization

Figure 23 depicts the comparison between EM Algorithm with centroid initialization as uniform distribution and K-means++ (non-uniform distribution) initialization in terms of total number of iteration for convergence in clustering over Ionosphere Dataset. As we can see in boxplot, for all of the values of k, number of iteration taken by K-means++ centroid initialization is much lower compared to normal initialization.

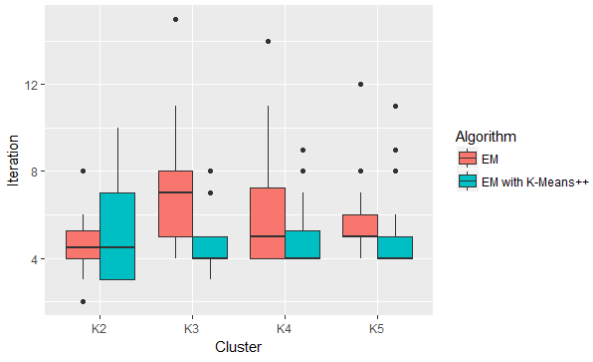

Figure 23: Boxplot of Iterations for k=2,3,..5 in normal and K-means++ centroid initialization