CIS 490 MACHINE LEARNING HOMEWORK 3
(Full Score: **35** + 5 bonus points)

Due: Nov 29th, 2017 Fall

Group number: 08

Group leader: Kevin D'Cruz(G) (01656193)

Group members and student IDs: Chetan Kumar(G) (01672511),
Abhishek Manoj Kumar(G) (01675536),
Madhuri Gawali(G) (01654141),
Apoorva Shivshankar(G) (01631727)

(Please mark "G" after names for Graduate Students in your group)

INSTRUCTOR: Dr. Julia Hua Fang

Overview of Homework 3 Score Distribution (35 points + 5 Bonus points)

1. CART, Bagging, Random Forest, and Boosting: 8 points.
2. Kmeans: 7 points
3. Hierarchical Clusteirng: 7 points
4. Gaussian Mixture Models: 6 points
5. Principal components Analyses: 7 points

Submission instruction:

1. Submit one **Word** file (e.g., *.docx) to address all questions below and add an "Appendix" section at the end of your file to attach all scripts and outputs
2. Submit all your zipped folders that contain all your original data, scripts and output as instructed for specific questions in this homework.

1. (8 points) CART, Bagging, Random Forest, and Boosting, monte Carlo:

1) What does CART stand for? (0.25 points)
Answer: CART stands for Classification and Regression Trees

2) If you have a continuous outcome, what kind of trees would you use? (0.25 points)
A. Regression Tree; B. Classification Tree
Answer: (A) Regression Trees

3) If you have a discrete outcome, what kind of trees would you use? (0.25 points)
A. Regression Tree; B. Classification Tree
Answer: (B) Classification Tree

4) What algorithm does CART typically use? (0.25 points)
Answer: CART typically uses Decision Trees, which is a type of Supervised Learning.

5) What term is similar to "Regions" used in CART? (0.25 points)
Answer: These regions can be termed as high dimensional rectangles or boxes.

6) What is the goal of regression trees? (0.25 points)
Answer: Goal of Regression trees is to create a model which predicts the target's value (dependent variable) based on the several input values (independent variables).

7) Why should we use cost complexity pruning (also called weakest link pruning)? (0.25 points)
Answer: Using a top-down greedy approach, called as recursive binary splitting for a tree building process may produce good predictions on the training set, but is likely to overfit the data. Here, each step of the tree-building process, best split is made that step, instead of looking ahead and looking ahead and picking a split leading to a better tree. This may lead to poor test set performance. Hence, a better strategy is to grow a very large tree and then prune it back for obtaining a subtree. The technique used for this is cost complexity pruning, also weakest link pruning. It typically uses misclassification loss for pruning regardless of the method used to grow the tree. Also, it is the easiest to implement via cross-validation.

8) In the pruning procedure, what technique do we use to select the tuning/penalty parameter *which* controls a trade-off between the subtree's complexity and its fit to the training data? (0.25 points)
Answer: In the pruning procedure, the technique used to select the tuning/penalty parameter (α) which controls a trade-off between the subtree's complexity and its fit to the training data is Cross Validation. An optimal value is selected using this technique, followed by which the subtree corresponding to that optimal value is returned to the full data-set.

9) What are the options to be used as a criterion for making the binary splits in the classification trees? (0.25 points)
A. Classification error
B. Gini index (a measure of purity; select the split with the smallest Gini index)
C. Cross entropy/deviance/information gain
Answer: B and C (Gini Index and Cross Entropy/Deviance/Information Gain)

10) What is the tree size? (0.25 points)

Answer: Tree size indicates as to how many terminal nodes or leaves the Decision Tree has.

11) What computational approach does the tree-building process use? What is the weakness of a greedy approach? (0.25 points)

Answer: The tree building process uses a top-down and greedy approach, called recursive binary splitting. The weakness of using a greedy approach is that at each step of the tree-building process, the best split is made at that step, instead of looking ahead and picking a split which would lead to a better tree in a future step.

12) Cross-validation (CV) can be used for (Hint: more than one option) (0.25 points)

A. choosing the optimal tuning parameter in regularized regression
B. choosing the optimal number of k for nearest neighbor classification
C. choosing the optimal tuning parameter in the pruning process of CART

Answer: A, B, C (Cross Validation can be used for all the 3 approaches)

13) Write out decision tree algorithms including pruning. (2 points)

Answer:

Decision Tree Algorithms including pruning are as follows:

1. ID3 (Iterative Dichotomiser 3)
2. C4.5 (successor of ID3)
3. CART (Classification And Regression Tree)
4. CHAID (CHi-squared Automatic Interaction Detector): Performs multi-level splits when computing classification trees
5. MARS: extends decision trees to handle numerical data better

Conditional Inference Trees: Statistics-based approach that uses non-parametric tests as splitting criteria, corrected for multiple testing to avoid overfitting. This approach results in unbiased predictor selection and does not require pruning

Decision Tree Algorithms including pruning have 4 steps:

1. Use recursive binary splitting for growing a large tree on the training data and stopping only when each terminal node has fewer than some minimum number of observations
2. Apply Cost Complexity pruning to the large tree for obtaining a sequence of best subtrees, as a function of α
3. Use K-fold cross-validation to choose α. For each k = 1, …K:
    a. Repeat Steps 1 and 2 on the (K-1)/K th fraction of the training data, excluding the kth fold
    b. Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of α. Average the results, and pick α to minimize the average error
4. Return the subtree from Step 2 that corresponds to the chosen value of α

Apart from this a few Decision Tree Pruning Methods include:

a. Validation set: withhold a subset (~1/3) of training data to use for pruning
b. Reduced Error Pruning: Classify examples in validation set, some may be errors
c. Pessimistic Pruning: Avoids needs to use validation set, can train on more examples

d. Cost Complexity Pruning: a measure of average error reduced per leaf
e. Rule Post-Pruning: Convert tree to rules (one for each path from root to a leaf)

14) Please use heart disease data, posted under Homework 3 on myCourses, to implement classification decision trees and replicate Figure 1, 2 and 3 shown on our lecture slides and below. (required for both undergraduates and graduates) (3 points)

Additionally, for graduates, please replicate figure 4 and 5 using bagging, random forest or boosting algorithms (2 points; for undergraduates, this will be counted as Bonus points)

Instruction:
Please (1) attach your R or Matlab codes and output in your word file as Appendices; (2) save and name your original scripts as "Group#_Trees.R" (e.g., Group1_Trees.R) or "Group#_Trees.m" (e.g., "Group1_Trees.m").

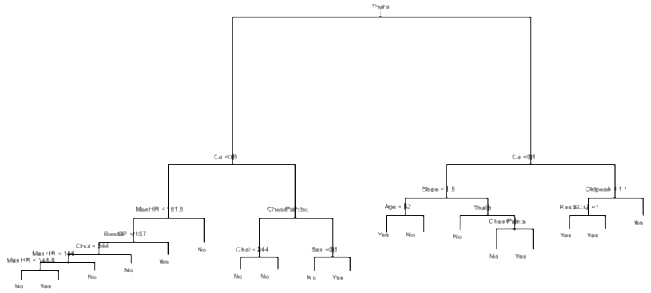


Figure 1

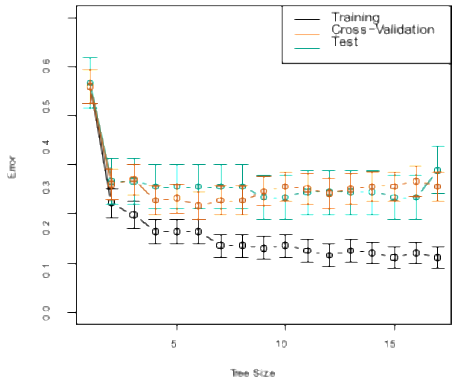


Figure 2

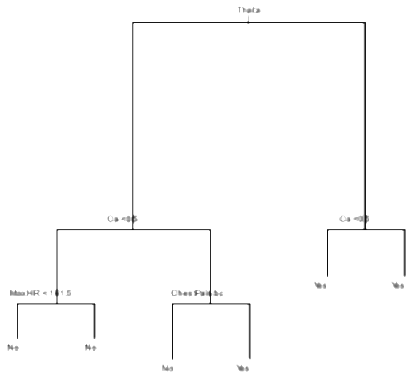


Figure 3

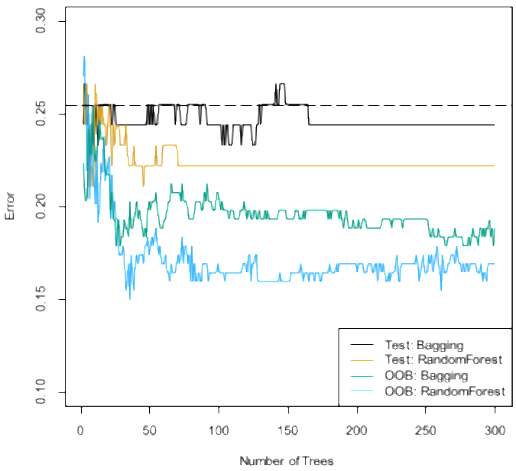


Figure 4

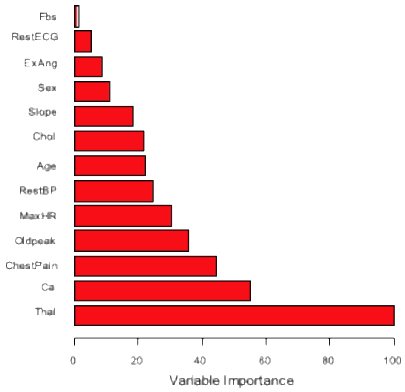
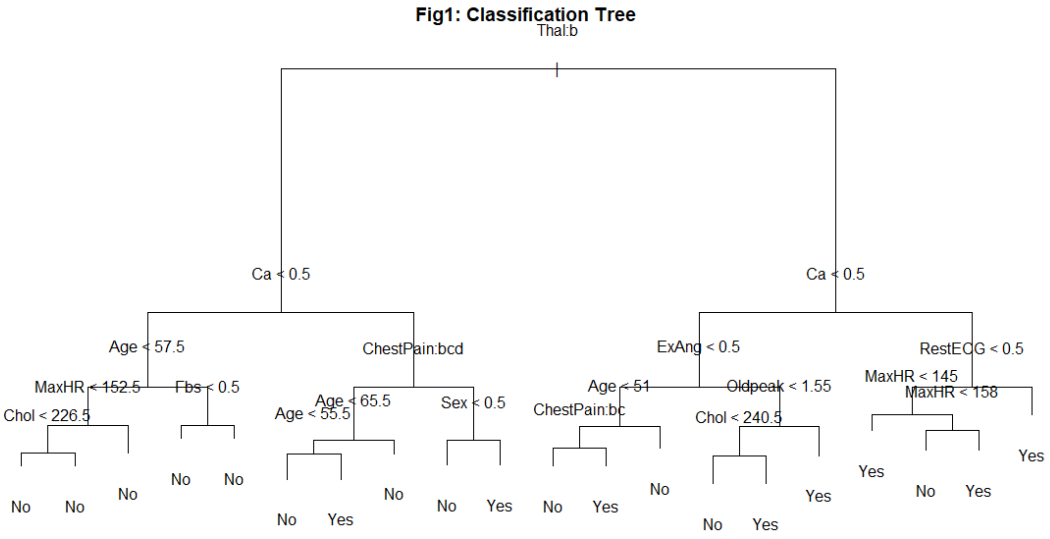


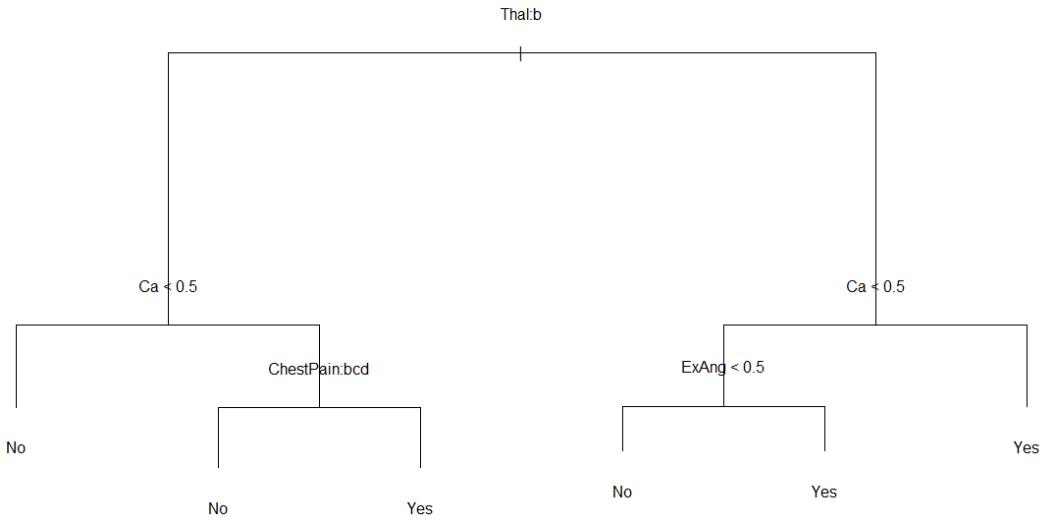
Figure 5

Figures:

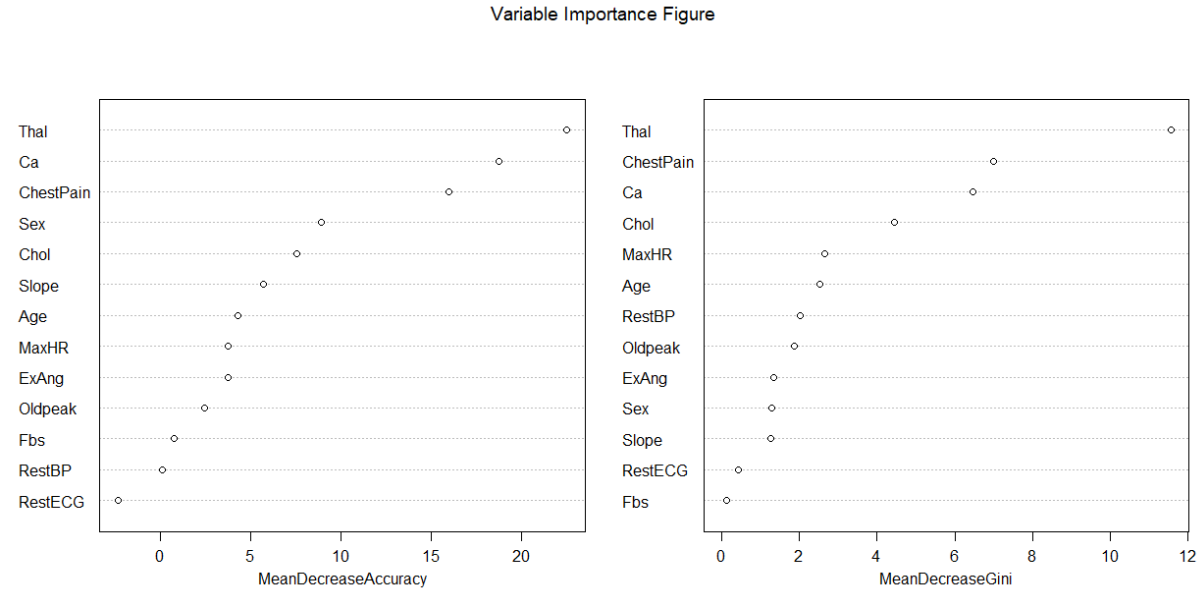
Classification Tree:



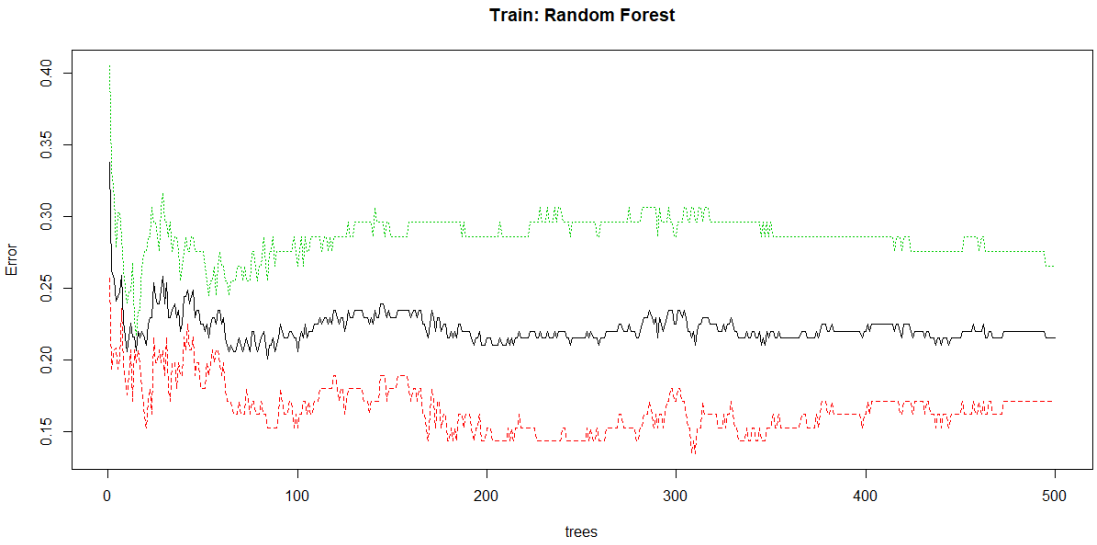
Pruned Tree

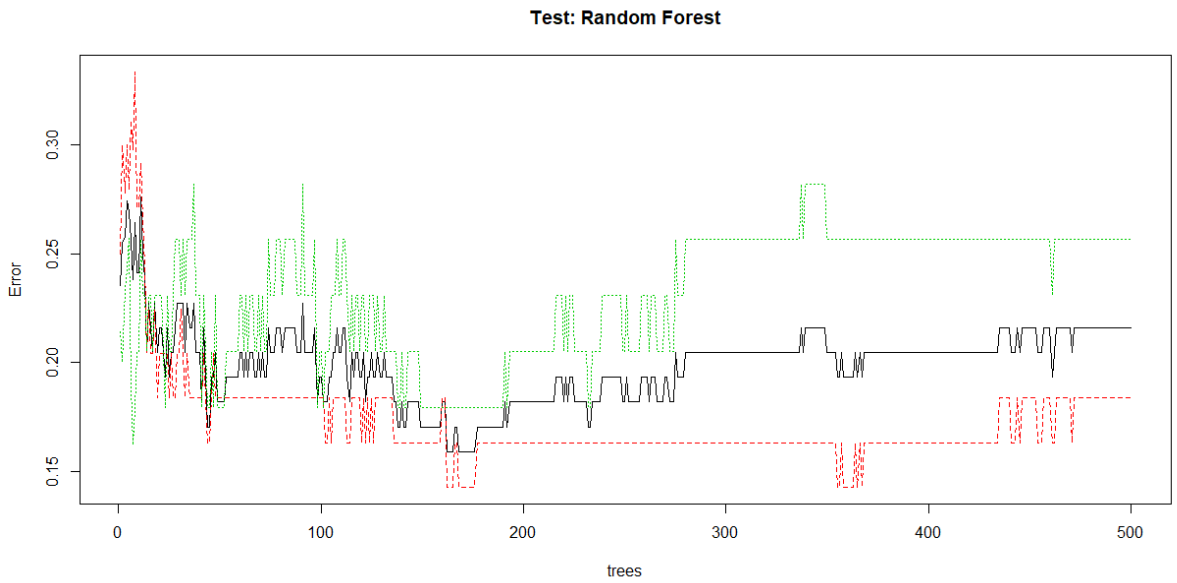


Variable Importance Graph



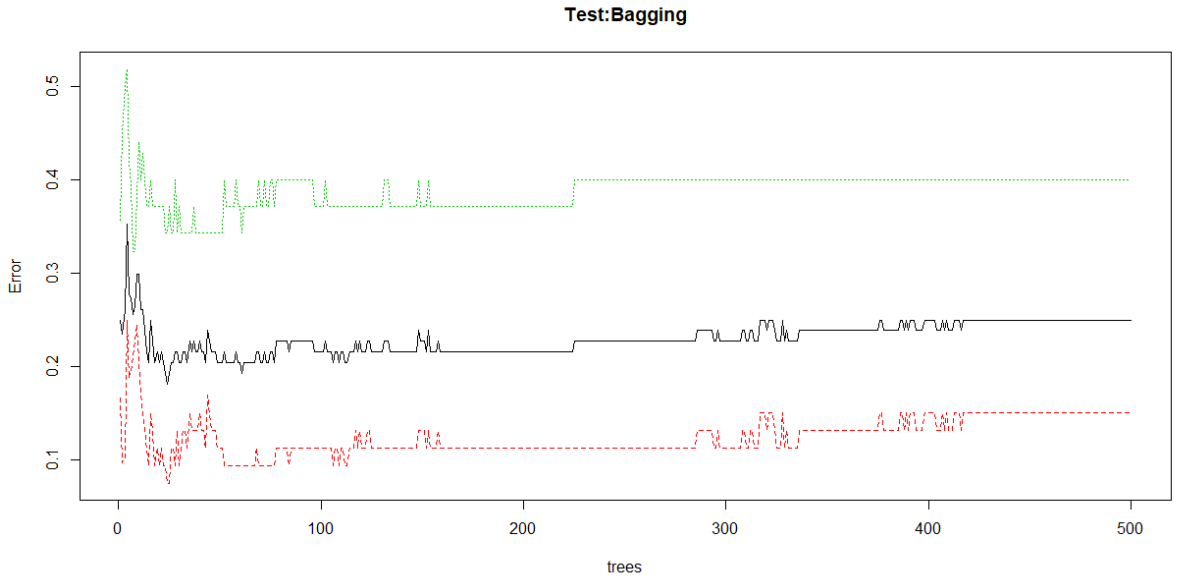
Random Forest



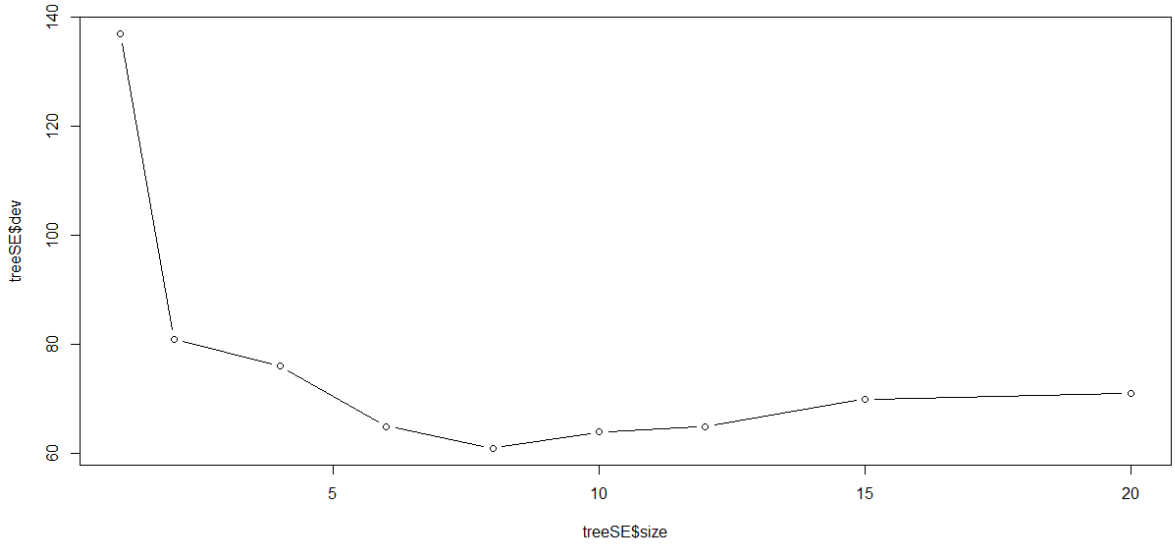

Test: Random Forest
Error
0.15
0.20
0.25
0.30
0
100
200
300
400
500
trees

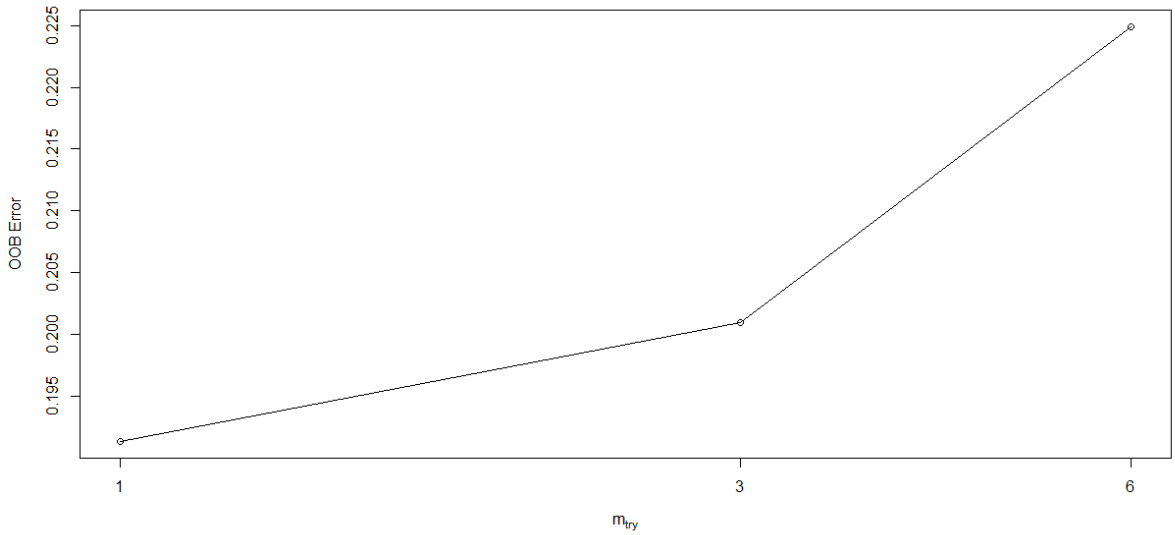
Bagging

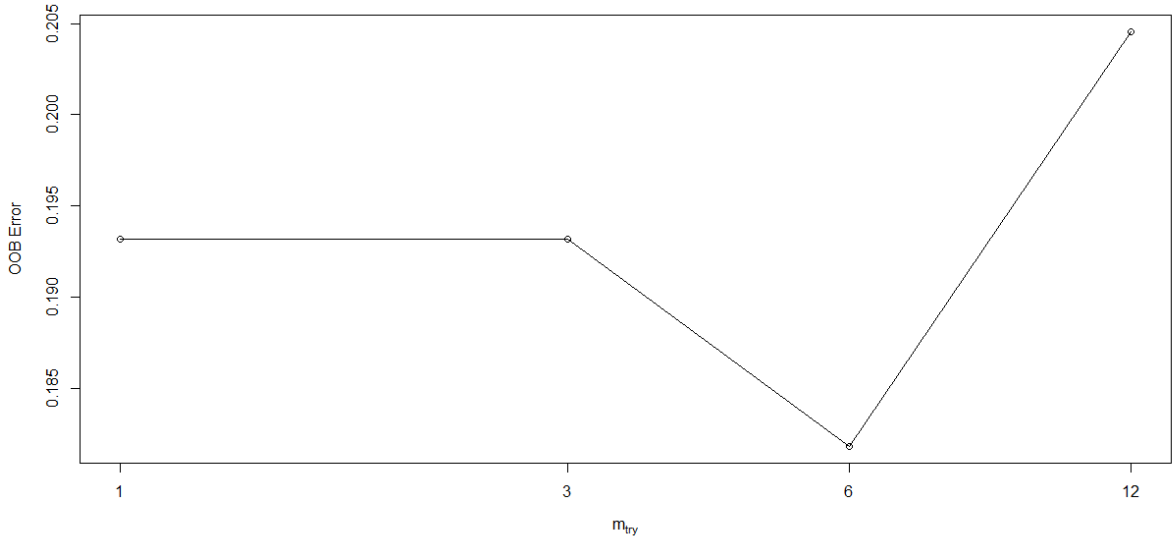

Train:Bagging
Error
0.10
0.15
0.20
0.25
0.30
0
100
200
300
400
500
trees


Test:Bagging
Error
0.1
0.2
0.3
0.4
0.5
0
100
200
300
400
500
trees

Tree Size Vs Error







2. Kmeans: 7 points

1) Describe Kmeans objective function (0.5 points)
   Answer: The objective function of k-mean is to minimize the average squared euclidean distance of point from cluster centers

2) Describe Kmeans algorithm (0.5 points)
   Answer: In this each cluster is associated with center point and each point is assigned to cluster whose center point is nearest to number of cluster i.e K
   a) Initially select K point as centroids
   b) Then repeat this until K clusters are formed by assigning all appoints to the closest centroid.
   c) After that recomputed the centroid of each clusters until the center point haven't change

3) What's difference between forgy and random partition initialization? (0.5 points)
   Answer:
   Forgy: It randomly chooses K observation from the data set and uses these as the initial means.
   Random Partition: In this initially it assigns a cluster to each observation randomly and the proceeds to update step.

4) How to choose the number of K for the K-means algorithm? (0.5 points)
   Answer: Sum of squares is used to choose the number of K for the k-means, squaring the error and summing them.

5) How to deal with random initialization issues in K-means? (0.5 points)
   Answer: To deal with random initialization issues in K-means following things should be done.
   a) Multiple runs must be done.
   b) Sample and hierarchical clustering used which will help to determine initial centroids.
   c) More than K,Centroids to be selected and then we must select among those centroids.
   d) With the help of bisecting K-means this problem can be solved.

6) What algorithm can be used to deal with outliers, if k-means is sensitive to outliers? (0.5 points)
   Answer:
   K-median algorithm is used to deal with outliers due to the presence of outliers K-median is more robust.

7) What are the assumptions for K-means? How to handle empty clusters in k-means? (0.5 points)
   Answer:
   a) Clusters are of similar size
   b) Clusters are spatially grounded or spherical
   c) The size of the cluster is similar, each data point belongs to the closest cluster to handle empty clusters we should select the point that offers most to SSE and also the highest SSE point from the cluster, we need to repeat the process if

there is any empty clusters.

8) How can we prevent local minima resulting from K-means (0.5 points)

Answer:

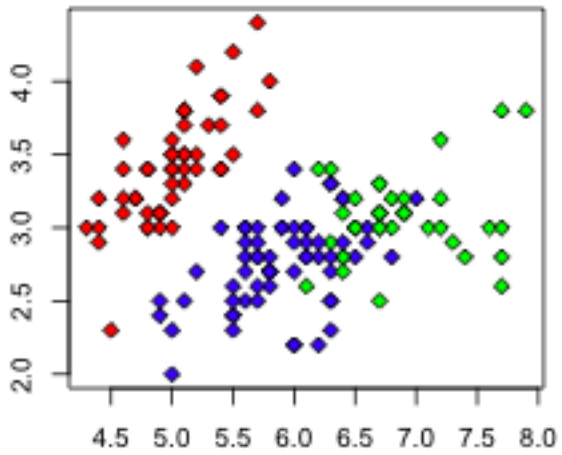
a) Initially select random center

b) Select new center [because it is proportional to (Xi-C)^2]

c) Repeat the above process until K centers.

9) Implement Kmeans on Iris data. Generate plots as illustrated below in a) and b): (3points)

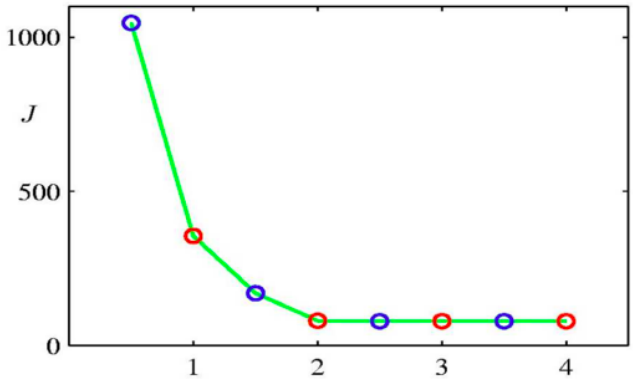
a) (1.5 points) the actual "scatter" plot when the number of clusters is chosen as 3, E.g,



b) (1.5 points) the plot where x = number of clusters, y = J, which is the cost value computed from

$$J(\mu, r) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \mu_k||^2$$

, similar to the graph below



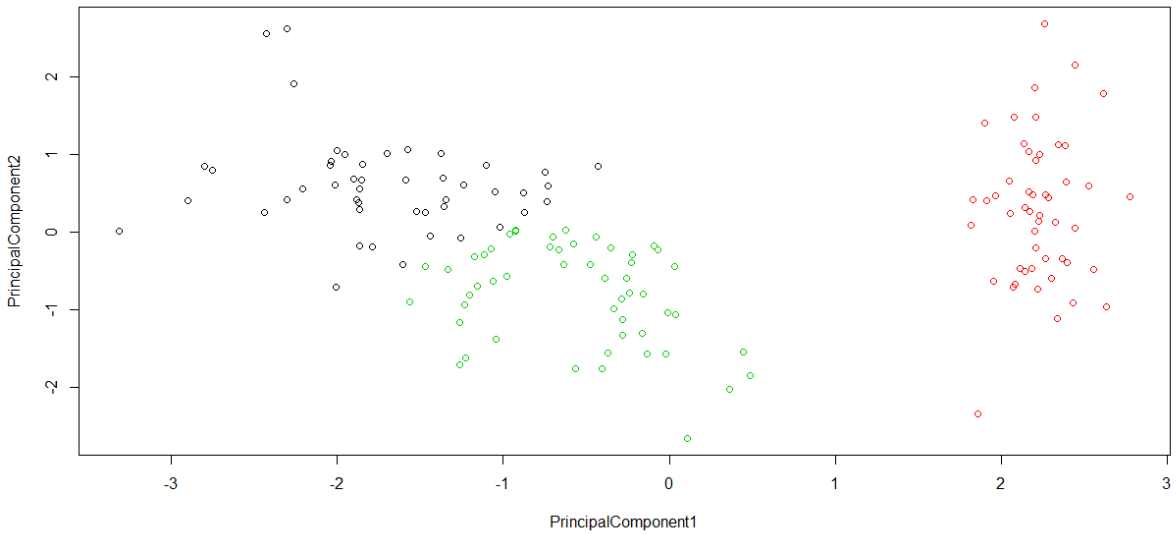
(c) Implement Kmeans++ on IRIS and report the findings you find in comparison to Kmeans. (1.5 Bonus points for both Graduates and Undergraduates)

Instruction:

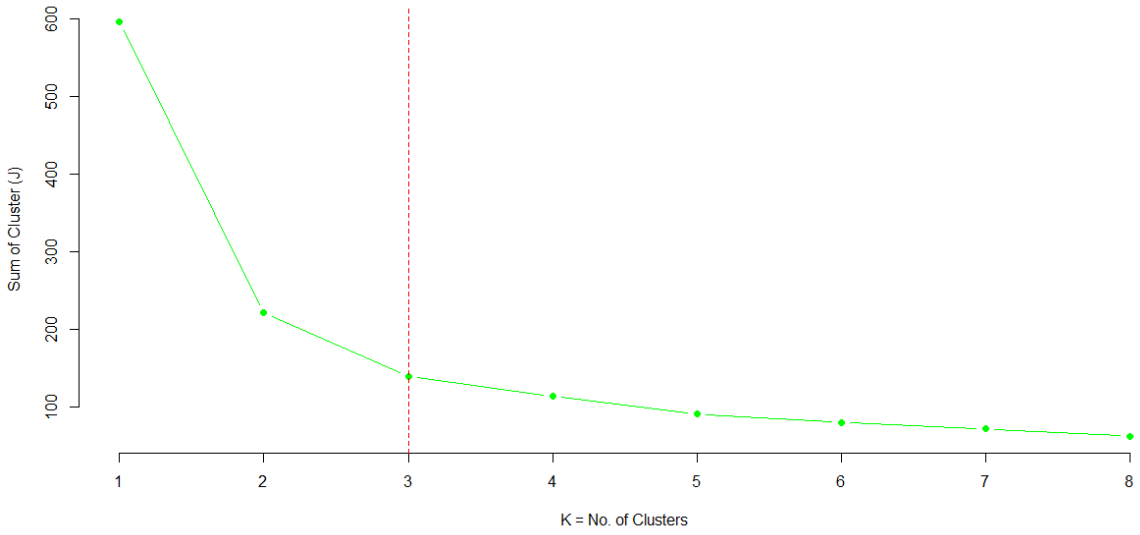
Please (1) attach your R or Matlab codes and output in your word file as Appendices; (2) save and name your original scripts as "Group#_Kmeans.R" (e.g., Group1_Kmeans.R) or "Group#_Kmeans.m" (e.g., "Group1_Kmeans.m").

Figures:

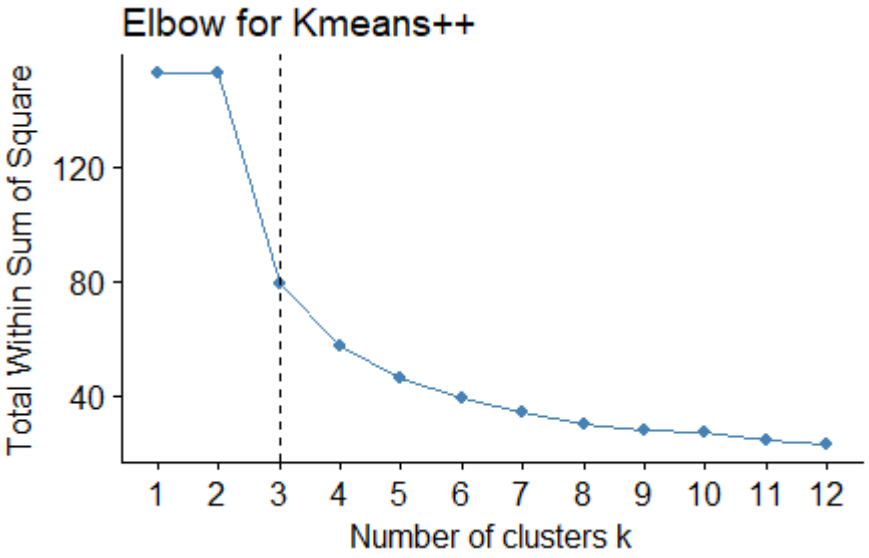
part(9-a)Kmeans



Part (9-b)



Part (9-C)



3. Hierarchical clustering (7 points)

1) What are the two hierarchical clustering methods/approaches? What are the differences between the two approaches? (0.5 points)

Answer:

Hierarchical clustering is a method for analysis which builds the hierarchy of clusters. The two HCA approaches are as follows:

- Agglomerative(combined): It is the bottom-up approach, starts with its own cluster or its singleton cluster. With each step, two most similar clusters are merged in a greedy manner and this process is stopped when there is only one cluster left. Otherwise, the process continues further.
- Divisive: it is the top-down approach. It starts in the same cluster and as each step progresses, the outsider is removed from the least cohesive cluster. This is stopped when each example is in its own cluster. Else, the process is further repeated.

2) What are the 3 dissimilarity measures? What are the differences among these three measures? (0.5 points)

Answer:

Min-link or single link: when the distance between the data point of each cluster is minimum. This link results in chaining whereas clusters can be very large at times.

$d(R, S) = \min_{x_R \in R, x_S \in S} d(x_R, x_S)$

Max-link or complete link: when the distance between the data points of each cluster is maximum. This link results in small round shaped clusters.

$d(R, S) = \max_{x_R \in R, x_S \in S} d(x_R, x_S)$

Average link: when the mean of the distance between the data points of each cluster is measured. This link is a compromise between the max-link (complete link) and min-link(single link).

3) What is the dendrogram? What does the vertical axis or the height indicate? Do we draw conclusions about the similarity of two observations based on their proximity along the horizontal axis or the location on the vertical axis? (0.5 points)

Answer:

Dendrogram is a tree like visual representation of the given observation which allows us to view the clustering obtained for each possible number of clusters at once, from 1 to n. it provides a highly interpretable complete description about the hierarchical clustering(graphical) format which counts as one of the main reason for the popularity and preferred factor for the usage of the hierarchical clustering.

The vertical axis or height indicates how different the measured observations are for the given clusters of the data points. Conclusions are drawn about the similarity of the two observations based on their proximity along not on the horizontal axis but on the location on the vertical axis where the branches containing those of the two observations. The location on the vertical axis is used because mathematically, there are 2^{n-1} possible recordings of the dendrogram, where n is the number of leaves at each of the n-1 points where the fused branches could be swapped without affecting the meaning of the dendrogram.

4) Would different linkage affect the hierarchical clustering behavior? Would scaling change the clustering results? (0.5 points)

Answer:

The choice off dissimilarity measure is important has a strong effect on the resulting dendrogram. Attention needs to be paid to the type of data that is being clustered.

From the above figure, it can be seen that when the average, complete, single link were applied on the dataset, the output of the average and complete link has more balanced clusters in comparison to the single link where the clusters are messed up and unbalanced.

Scaling will change the clustering results.

5) Describe Agglomerative Clustering Algorithm (1 point)

Answer:

The Hierarchical clustering algorithm or the Agglomerative clustering algorithm can be described as follows:

- Start with each point in its own cluster.
- Identifies the closest two clusters and merge with them.
- This process is repeated on loop.

It ends when all the points are emerged as one single cluster.

6) What criteria can we use to select the optimal number of clusters in hierarchical clustering? (0.5 points)

Answer:

The gap statistic which compares the curve with $\log W_k$ estimates the optimal number of the clusters to be placed where the gap between the two curves is largest in the hierarchical clustering analysis.

7) How many clusters do we have if we cut the height at 9 based on Fig (a) below? How about when cutting at a height of 5 in Fig (b) (0.5 points)

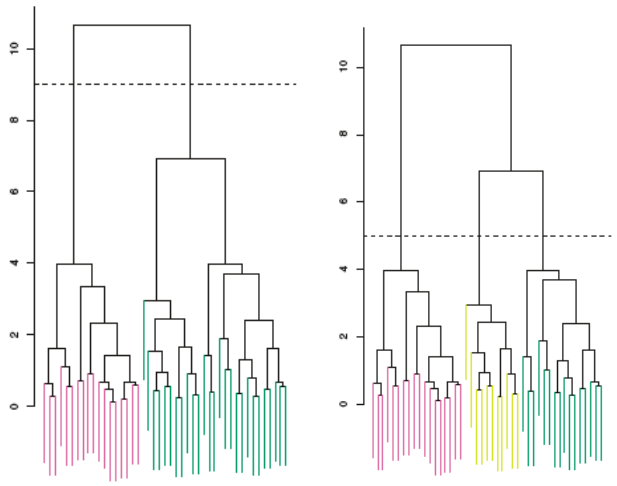


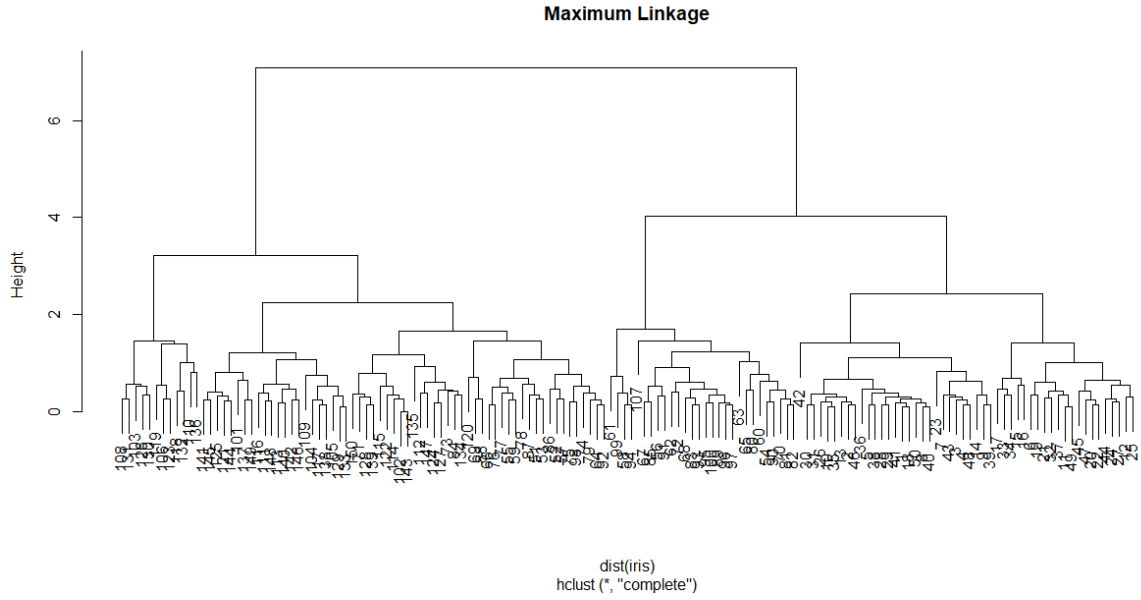
Fig (a) Fig (b)

Answer: If we cut 9, we have two distinct clusters (as the cluster is cut to its center). If we cut at 5, we have3 distinct clusters (this is interpreted as the right panel view for the given dendrogram).

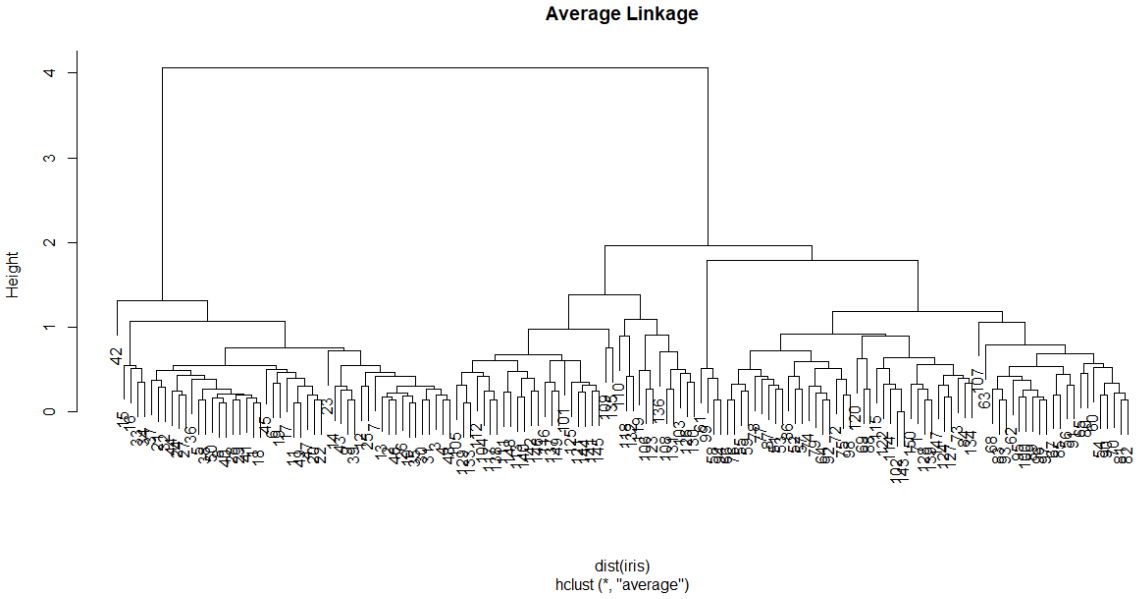
8) Implement Agglomerative hierarchical clustering on IRIS data, using three linkage methods, respectively, and output dendrograms for each method. (3 points)
Instruction:
Please (1) attach your R or Matlab codes and output in your word file as Appendices; (2) save and name your original scripts as "Group#_Hierarchical.R" (e.g., Group1_ Hierarchical.R) or "Group#_ Hierarchical.m" (e.g., "Group1_ Hierarchical.m").

Figures

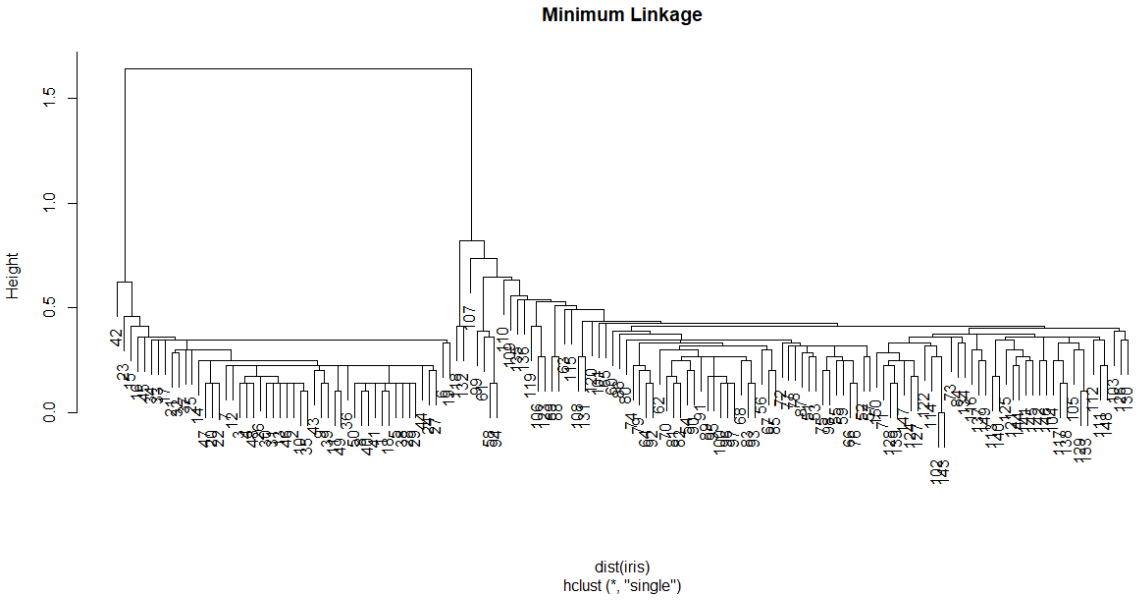
Maximum Linkage



Average Linkage



Minimum Linkage



4. Gaussian Mixture Models(GMM) (6 points)

1) When do we use GMM? What is the difference between GMM and Kmeans? (1.5 points)

(1.5 points)

Answer: Expectation Maximization (EM)

i) Initial guesses

ii) Iterative algorithm

Two main steps:

i) E-step: guess the values of the latent random variables.

ii) M-step: update the parameter of the model by assuming our guesses were correct.

iii) Repeat until convergence.

Difference between K-means and GMM

K-means-Clustering of samples is done by minimizing the distance between sample and the centroid i.e., assign the centroid and optimize the centroid based on the distances from the point to it. They are easy to interpret and implement.

GMM-It uses probability of a sample to determine the feasibility of it belonging to a cluster, when we are uncertain about the data points where they belong to which group we use GMM method. They are difficult to interpret.

2) What is responsibility in the context of GMM? (0.5 points)

Answer: In the context of GMM the relative guassians densities g1(x)/(g1(x) +g2(x)) and g2(x)/(g1(x)+g2(x), called the responsibility.

3) Describe EM algorithm. (3 points)

Answer:

Mixture Model: $f(x) = (1-\pi)\, g_1(x) + \pi g_2(x)$

Gaussian mixture: $g_j(x) = \phi_{\Theta j}(x)$, $\Theta_j = (\mu_j, \sigma^2_j)$

Considering the Gaussian Mixture Model (GMM) has 2 panels: left and right. Considering Left Panel has two Gaussian densities g1(x) and g2(x) and right panel has relative densities called the responsibility of each cluster.

The EM (Expectation and Maximization) algorithm uses these responsibilities for making a "soft" assignment of each data point to each of the two clusters. When σ is large, the responsibilities can be near 0.5.

As $\sigma \rightarrow 0$, the responsibilities $\sigma \rightarrow 1$, for the cluster center closest to the target point, and 0 for all other clusters. This "hard" assignment could be seen in the bottom right panel.

The algorithm is an iterative algorithm that starts from some initial estimate of Θ (e.g., random), and then proceeds to iteratively update Θ until convergence is detected. Each iteration consists of an E-step and an M-step.

Taking an initial guess for the parameters: $\hat{u}_1, \sigma^2_1, \hat{u}_2, \sigma^2_2, \hat{\pi}$

Expectation: Compute the responsibilities

$$\hat{\gamma}_i = \frac{\hat{\pi}\phi_{\hat{\theta}_2}(y_i)}{(1-\hat{\pi})\phi_{\hat{\theta}_1}(y_i) + \hat{\pi}\phi_{\hat{\theta}_2}.(y_i)},\ i = 1, 2, \ldots, N. \quad (3)$$

Maximization Step: Computing the weighted means and variances:

$$\hat{\mu}_1 = \frac{\sum_{i=1}^{N}(1-\hat{\gamma}_i)y_i}{\sum_{i=1}^{N}(1-\hat{\gamma}_i)}, \qquad \hat{\sigma}_1^2 = \frac{\sum_{i=1}^{N}(1-\hat{\gamma}_i)(y_i-\hat{\mu}_1)^2}{\sum_{i=1}^{N}(1-\hat{\gamma}_i)},$$

$$\hat{\mu}_2 = \frac{\sum_{i=1}^{N}\hat{\gamma}_i y_i}{\sum_{i=1}^{N}\hat{\gamma}_i}, \qquad \hat{\sigma}_2^2 = \frac{\sum_{i=1}^{N}\hat{\gamma}_i(y_i-\hat{\mu}_1)^2}{\sum_{i=1}^{N}\hat{\gamma}_i},$$

and the mixing probability (a.k.a: mixing proportion/mixing coefficient)

$$\hat{\pi} = \sum_{i=1}^{N} \hat{\gamma}_i / N.$$

Iterate these steps until convergence.

4) Why do we use log-likelihood instead of likelihood in EM (1 points)

Answer:

The total likelihood is the product of the likelihood for each point. i.e.:

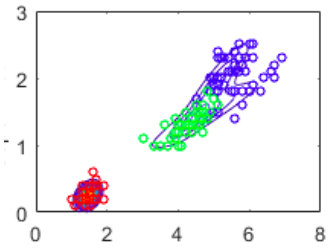
$P(X|\Theta) = \pi i = 1Np(x_i|\Theta)p(X|\Theta) = \pi i = 1Np(x_i|\Theta)$

Where Θ is the model parameters: vector of mean μ and covariance matrix ∑.
If the log-likelihood is used, we get the sum in place of the product:

$lnp(X|\Theta) = \sum i = 1Nlnp(x_i|\Theta)$

From a standpoint of computational complexity, it is imagined that summing is less expensive than multiplication (summing and multiplication are almost equal in recent times). Most importantly, the likelihoods become very small and we may run out of floating point precision very quickly, which results in an underflow. Hence, is would be more convenient to use logarithms of the likelihood.

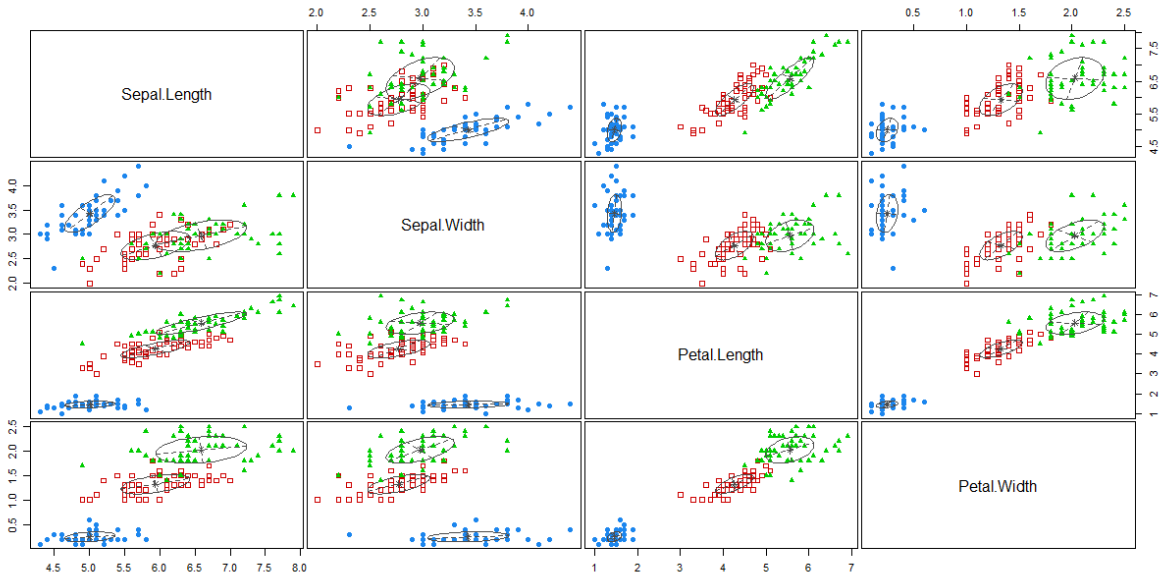
5) Based on IRIS data, simulate/sample data from the GMM using the GMM sampling procedure. Please use the seed 999 to simulate data and plot your simulated data, e.g., a faked plot as below. (2 Bonus points for undergraduates; required for graduates)



Instruction:

Please (1) attach your R or Matlab codes and output in your word file as Appendices; (2) save and name your original scripts as "Group#_GMM.R" (e.g., Group1_ GMM.R) or "Group#_ GMM.m" (e.g., "Group1_ GMM.m").

Figures:



5. Principal Components Analysis (PCA) (7 points)

1) What is the difference between PCA and clustering in the context of unsupervised learning? (1 point)

Answer: PCA tends to look for low dimensional representation of the observations which explains a good fraction of the variance.
Where has clustering looks for homogeneous subgroups among the observations.

2) PCA looks at means or variance in the high-dimensional data? By saying principal component are assumed orthogonal to each other, what does orthogonal mean? (1 point)
Answer: In high dimensional data, PCA looks for set of related variables which explain most of the variance and creates a feature of it which becomes the first component and then does it for the next set of variables different/ unrelated to the first which becomes the next component.
By saying principal component are assumed orthogonal to each other , "orthogonal" refers to the components being perpendicular to each other and that the cross product is zero.

3) Describe SVD based PCA algorithm (1.5 bonus points for both undergraduates and graduates)
Answer:

- Start with a data matrix X of m x n dimension.
- Subtract mean from each row of X : re-center
  $X - X_c = X - X$
- Apply SVD algorithm on Xc to obtain K singular vectors.
- Principle components: K singular vectors with highest singular values.
- Coefficient: Project each point onto the new vectors.

4) What are the loading, eigenvalues, eigenvector in the context of PCA? (1.5 point)
Answer:

Loading are the covariance/correlations between the original variables and the unit scaled components.
In PCA, we split the covariance/correlation matrix into scale part i.e. eigenvalues and direction part i.e. eigenvectors.
Eigenvector is just a coefficient of orthogonal *transformation* or projection and Eigenvalues are the variances of Principal components and when eigenvectors are multiplied by the square root of eigenvalues, we load the bare coefficient by the amount of variance. We make the coefficient to be the measure of association and co-variability.

5) Is the total variance of a dataset equal to the variance explained by components identified in PCA? (0.5 point)

Answer: Yes, I can be proved by the fallowing equations, the total variance of a dataset i.e.,

$$\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2,$$

And the variance explained by components identified in PCA i.e.,

$$\mathrm{Var}(Z_m) = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2.$$

It is seen that,

$\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{m=1}^{M} \mathrm{Var}(Z_m)$, with M = min(n-1,p).

6) What closeness measure does PCA use to find the hyperplane closest to the observations? (0.5 point)

Answer: PCA uses average squared Euclidean distance to obtain the closeness measure to find the hyperplane closest to the observations.

7) Based on the loading matrix from the USarrests data, which variables will be counted into PC1 and which one will be counted into PC2? (0.5 point)

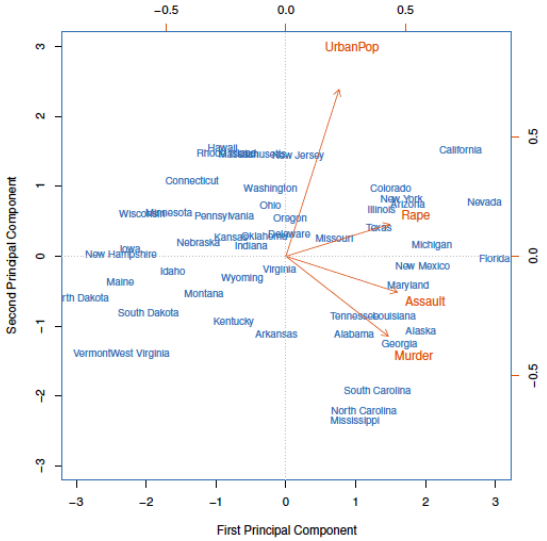
	PC1	PC2
Murder	0.5358995	-0.4181809
Assault	0.5831836	-0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186

Answer: Murder, Assault and Rape are variables counted in PC1
Assault and Rape are variables counted in PC2. This is decided by the proportion Weights which are imposed on the variables.

8)

(a) For the USarrest data, what do the blue state names represent on this bi-plot shown below ? What does the orange arrows indicate? (1 point)

Answer: The blue shows the state names scores for the first two principal components and biplot focuses on display of both principal component scores and the principal component loadings. The orange arrow indicates the first two principal component loading vectors.



(b) For the USarrest data, what does the scree plot (Fig 1) depict? What does Fig 2 depict? (1 point)

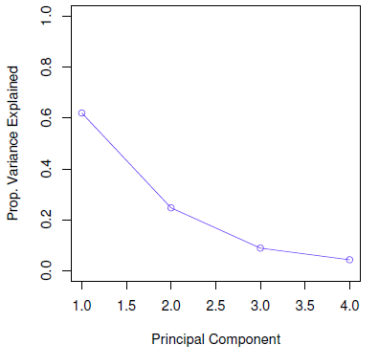


Fig 1.

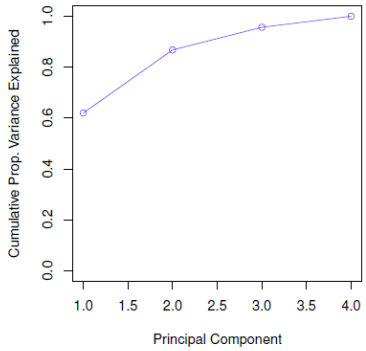


Fig 2.

Answer: Left: a screen plot depicting the proportion of variance explained by each of the four principal components in the USArrests data.

Right: the cumulative proportion of variance explained by the four principal components in the USArrests data.

Appendices

Q1

```
# Read data
data <- read.csv(file.choose(), header = T)
data$X <- NULL
data <- na.omit(data)

# Tree
library(tree)
tree1 <- tree(AHD ~., data)
plot(tree1)
title("Fig1: Classification Tree")
text(tree1, pretty = NULL)

# Pruned Tree
ptree <- prune.misclass(tree1, best = 6)
plot(ptree)
text(ptree, pretty = NULL)

# Variable Importance Plot
importance(rf)
varImpPlot(rf,main = "Variable Importance Figure")

## Splitting data
set.seed(511)
ind <-sample(2,nrow(data), replace = T, prob = c(0.7, 0.3))
train <- data[ind ==1,]
test <- data[ind ==2,]

# Train Random Forest
rf <- randomForest(AHD ~ ., train,
                   importance = T,
                   ntree = 500,
                   mtry = 8)
plot(rf, main="Train: Random Forest")

# Test Random Forest
rf <- randomForest(AHD ~ ., test,
                   importance = T,
                   ntree = 500,
                   mtry = 8)
plot(rf, main="Test: Random Forest")


# Train Begging
train1 = sample(1:nrow(data), nrow(train))
bagging <- randomForest(AHD ~., data, subset = train1, mtry = 8, importance = TRUE)
plot(bagging, main="Train:Bagging")
```

```
# Test Begging
test1 = sample(1:nrow(data), nrow(test))
bagging <- randomForest(AHD ~., data, subset = test1, mtry = 8, importance = TRUE)
plot(bagging, main="Test:Bagging")

## Tree size and error
## Reference: http://www.stat.cmu.edu/~cshalizi/350/lectures/22/lecture-22.pdf
treeSE = cv.tree(tree1, FUN = prune.misclass)
plot(treeSE$size, treeSE$dev, type='b')

## OOB Error - Train
t <- tuneRF(train[,-14], train[,14],
        stepFactor = 0.5,
        plot = T,
        ntreeTry = 100,
        trace = T,
        improve = 0.05)

## OOB Error - Test
t <- tuneRF(test[,-14], test[,14],
        stepFactor = 0.5,
        plot = T,
        ntreeTry = 100,
        trace = T,
        improve = 0.05)
```

Q2

```
data(iris)

pca <- princomp(iris[, 0:4], cor=T)
pc.comp <- pca$scores
PrincipalComponent1 <- -1*pc.comp[,1]
PrincipalComponent2 <- -1*pc.comp[,2]

clustering.data <- cbind(PrincipalComponent1, PrincipalComponent2)

## kmeans
set.seed(511)
km <- kmeans(clustering.data, 3,iter.max = 15,nstart=20)
plot(PrincipalComponent1, PrincipalComponent2, col=km$cluster)

## elbow kmeans
library(cluster)
library(ggplot2)

data("iris")

iris.scaled <- scale(iris[, -5])
k.max <- 8
```

```
data <- iris.scaled
wss <- sapply(1:k.max,
          function(k){kmeans(data, k, nstart=10 )$tot.withinss})
plot(1:k.max, wss,
    type="b", col="green",pch = 19, frame = FALSE,
    xlab="K = No. of Clusters",
    ylab="Sum of Cluster (J)")

abline(v = 3, col="red", lty =2)


## Kmeanspp
head(iris)
data <- iris[, -5]
head(data)

#elbow

library(factoextra)
library(LICORS)
# Elbow Method

fviz_nbclust(data,
          FUNcluster = kmeanspp,
          method = "wss",
          k.max = 12) +
  labs(title="Elbow for Kmeans++") +
  geom_vline(xintercept = 3,
          linetype = 2)
```

Q3

```
data("iris")
## reference: https://www.r-bloggers.com/hierarchical-clustering-in-r-2/
iris <- iris[,1:4]
hcMaxCluster <- hclust(dist(iris))
plot(hcMaxCluster,main="Maximum Linkage")

hcAvgCluster <- hclust(dist(iris), method = "average")
plot(hcAvgCluster,main="Average Linkage")

hcMinCluster <- hclust(dist(iris), method = "single")
plot(hcMinCluster,main="Minimum Linkage")
```

Q4

```
## https://stats.stackexchange.com/questions/111145/how-to-fit-mixture-model-for-
clustering
set.seed(999)
data(iris)
class <- iris$Species
X <- iris[,1:4]
```

```
plot(X,class,  pch = 19, col = "gray40")


library("mclust")
mod2 <- MclustDA(X, class, modelType = "EDDA")
plot(mod2, what = "scatterplot")
```