CSE5243

HW5: K-means clustering Analysis

Yifei Zhang (zhang.5387)

**Overview**

This report is concentrated on the clustering analysis on three dataset: TwoDimEasy, TwoDimHard, and wine. Part1 is creating program for K-means algorithm and conducting related clustering analysis among three dataset. For each dataset, different k-value is tested. Part 2 is conducting k-means clustering analysis with off-the –self clustering method in R. And compare the result between these two methods.

* 1. **Part1**

**1.1 Program description:**

**1.1.1 Input data:**

This program is written by R. User can input data with file directory.

Before run program, user need to intall.package( ggplot2, SpatialTools, gridExtra,cowplot)which are help for generate plots with following code.

library(ggplot2)   
install.packages("SpatialTools")  
library(SpatialTools)   
install.packages("gridExtra")  
library("gridExtra")  
install.packages("cowplot")  
library("cowplot")

**1.1.2 K-means algorithm convergence condition**

At K means algorithm, it terminates when there is no change on centroids. K means reach a state that no points shifting from one cluster to another cluster. Since convergence occurs in iterations step, the weak condition that iteration repeat until 1% of the points change cluster ( set threshold of the percentage of changing point as 0.01. The corresponding code is:

while(change==TRUE)   
{

….  
if(sum(oldcluster[,2]!=kis2.easy1[,4])/nrow(kis2.easy1)<=0.01)   
{ change=FALSE }  
……..  
}

When less than 1% of the points change cluster, change=false, the while loop stops.

**1.1.3 K-means algorithm**

The initial centroid point was selected randomly from the entire data set. Each cluster associated with centroids. Then calculate the distance from each point to centroid i, assign the point to the closest centroid. The corresponding code is show as below(this is for TwodimEasy Dataset when k=2):

for (i in 1:n)   
 {  
 dist[i,1]<-sqrt((kis2.easy1[i,2]-centroid.easy1[1,1])^2+(kis2.easy1[i,3]-centroid.easy1[1,2])^2)  
 dist[i,2]<-sqrt((kis2.easy1[i,2]-centroid.easy1[2,1])^2+(kis2.easy1[i,3]-centroid.easy1[2,2])^2)  
 if(dist[i,1]==min(dist[i,]))   
 {  
 kis2.easy1[i,4]=1  
 }  
 else{  
 kis2.easy1[i,4]=2  
 }   
 }

* 1. **K- means clustering for TwoDimEasy Dataset**

In this part, the true cluster SSE, overall SSE, and SSB between clusters are calculated. K-mean algorithm is applied to clustering analysis when k=2, and k=3. For each k means run, SSE, overall SSE, SSB, silhouette width was calculated. And also compared cluster result from k mean algorithm with the true cluster to evaluate how well the k-mean algorithm performed.

**1.2.1 SSE, SSB for true cluster.**

Based on the original data, we can know that there are two different cluster, cluster1 and cluster2.In this part, the coordinates of centroid point for each cluster was calculated by extracting the mean of each coordinate of observation in each cluster.

#centroid for cluster1  
centroid\_easy<-matrix(0,nrow = 2,ncol=3)  
centroid\_easy[1,1]<-mean(subset(twodimeasy,cluster==1)$X.1)  
centroid\_easy[1,2]<-mean(subset(twodimeasy,cluster==1)$X.2)  
#centriod for cluster2  
centroid\_easy[2,1]<-mean(subset(twodimeasy,cluster==2)$X.1)  
centroid\_easy[2,2]<-mean(subset(twodimeasy,cluster==2)$X.2)

SSE for each cluster is calculated by following formula( which is used to quality clustering :

The function code in the programing is :

*##SSE funtion*SSE<-function(df,cluster1,i,j,k,centroid){   
SSE=sum((subset(df,cluster==cluster1)$X.1-centroid[i,j])^2+(subset(df,cluster==cluster1)$X.2-centroid[i,k])^2)  
*}*

SSB( sum of squared error between cluster ) is also calculated by following code:

SSB\_easy=(N\_cluster1.easy\*((centroid\_easy[1,1]-mean(twodimeasy$X.1))^2+(centroid\_easy[1,2]-mean(twodimeasy$X.2))^2)+ N\_cluster2.easy\*((centroid\_easy[2,1]-mean(twodimeasy$X.1))^2+(centroid\_easy[2,2]-mean(twodimeasy$X.2))^2))

The SSE for each cluster, overall SSE, SSB value for true cluster in TwoDimEasy Dataset is shown as below, the SSB is relatively larger than the overall of SSE, which indicate, true cluster splits data very well..

Table 1.1 SSE for each cluster, overall SSE, SSB



* + 1. **K means algorithm**

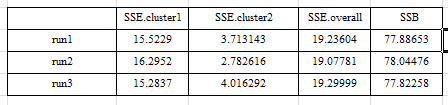
In this part, K means (k=2, k=3) algorithm is used for clustering. For each k, the algorithm is run 3times. For each run the SSE, SSB, ASW (average silhouette width) for each cluster and for entire dataset was calculated. In this K-means, the initial centroid points are selected randomly from entire data set.

***K=2***

When K=2, SSE for each cluster, overall SSE, and SSB between clusters is shown in table 1.2.From this table, we can know that there is no significant difference between 3 runs, which means the initial centroid point won’t affect the cluster result . All of overall SSE is about 19; the SSE is about 77 to 78. The SSE for each cluster is similar. Comparing to the value of true cluster in table1.1, we can know that this cluster result is very close to true cluster.

1. SSE and SSB

Table1.2 SSE for each cluster, overall SSE, SSB for K=2 in TwoDimEasy Dataset



1. ASW

Table1.2 ASW for K=2 in TwoDimEasy Dataset

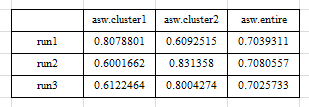


Table 1.2 shows the ASW (average silhouette width) for each cluster and for entire dataset. From the table, we can know that all of these three runs have high ASW value (>0.7), which reveals that the clustering perform excellently.

1. Confusion matrix of three runs:

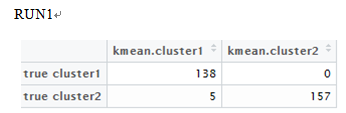
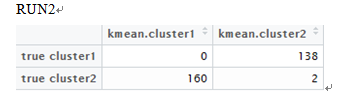
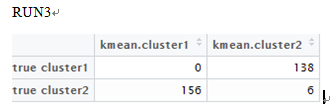
**  **

Table1.3 Accuracy for k=2 clustering in three runs



From above these three confusion matrix and table 1.3, we can know that k-mean algorithm ( k=2 ) performs very well. The accuracy is more than 98%. (Due to the random of initial centroid point, the k-mean cluster number may not consistent in each run. So the confusion matrix may display like run 3. For run 3, the accuracy is ( 156+138)/300. This won’t affect the accuracy of clustering)

1. Scatterplot for clustering of three runs

Figure1.1 Scatterplot for true cluster and K-means clustering ( K=2)

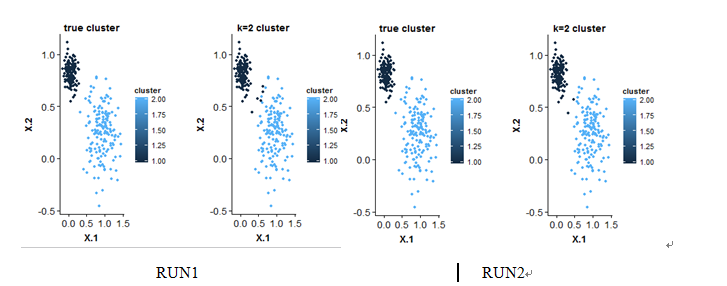


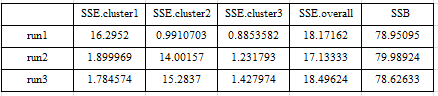
Figure 1.1 shows the comparison between true cluster and K-means clustering (K=2) for each run. From this figure, we can know that:

1. For each run, the K-means clustering result is close to true cluster.
2. There are no many points shifting between clusters during iteration.

***K=3***

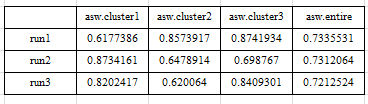
1. SSE and SSB

Table1.4 SSE for each cluster, overall SSE, SSB for K=3 in TwoDimEasy Dataset



1. ASW

Table1.5 ASW for K=3 in TwoDimEasy Dataset



According to table 1.4 and table1.5, we can know that:

1. The overall SSE and SSB of K=3 is close to K=2 and the true cluster. But SSE of K=3 is smaller than SSE of k=2, the SSB is larger than k =2, which means K=3 performs better than K=2 based on SSE and SSB value.
2. The ASW of each cluster when k=3 is large than 0.6, the ASW of entire dataset is about 0.72 -0.73( >0.7),which means K=3 split data very well.
3. Confusion matrix for each run

Table1.6 Confusion matrix for Run=1

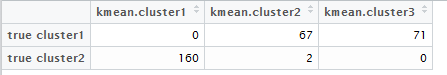


Table1.7 Confusion matrix for Run2

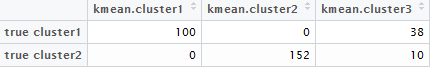
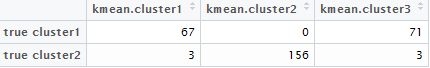
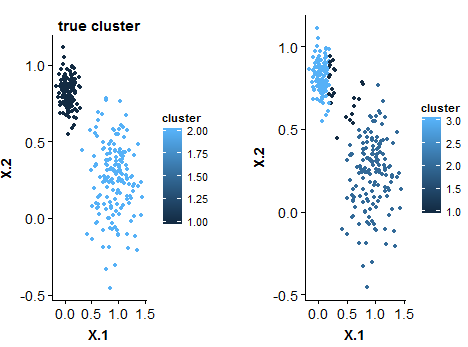
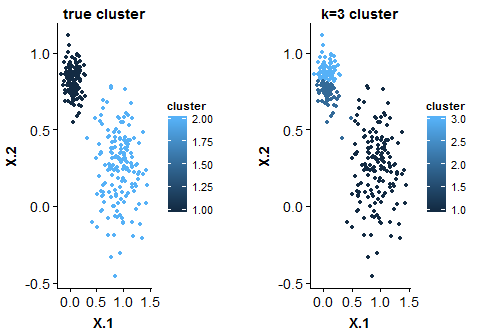
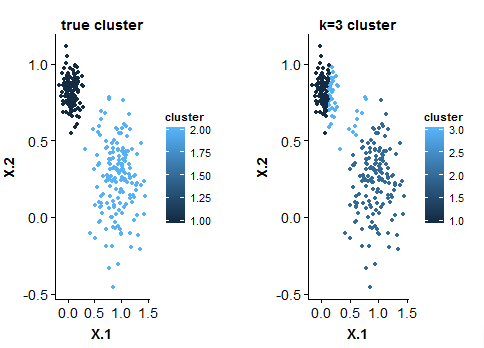


Table1.8 Confusion matrix for Run3



1. Scatterplot for clustering of three runs

Figure1.2 Scatterplot for true cluster and K-means clustering (K=3)

According to the confusion matrix for three runs (Table1.6, Table1.7, Table1.8), and scatterplot or true cluster and K-means clustering (Figure1.2), we can know the accuracy of k-means (k=3) algorithm decrease comparing with k=2. And in each run, many point shift from one cluster to other clusters.

**1.2 K-means clustering of TwoDimHard Dataset**

1.2.1 SSE, SSB for true cluster.

Table 1.9 SSE for each cluster, overall SSE, SSB

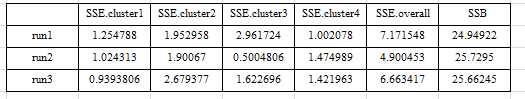


1.2.2 k-means algorithm.

***K=4***

a. SSE, SSB for true cluster for k-means clustering

Table1.10 SSE for each cluster, overall SSE, SSB for K=4 in TwoDimHard Dataset

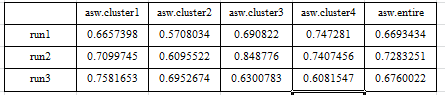


From this table, we can know that:

1. There is difference between 3 runs, which means the initial centroid point might affect the cluster result. The overall SSE and the SSE of each cluster varies among tree runs.
2. Comparing to the value of true cluster in table1.9, the overall SSE sometimes is larger than true cluster, sometimes smaller than true cluster. However, the SSB value of K-means clustering is larger than true cluster, which indicates the good performance of k-means clustering(k=4)

b. ASW

Table1.11 ASW for K=3 in TwoDimhHard Dataset



According to table1.11, the ASW of each cluster of entire dataset and when k=4 is large than 0.6, that means, K=3 splits data very well.

c. Confusion Matrix for 3 runs

Table1.12 Confusion matrix for Run=1

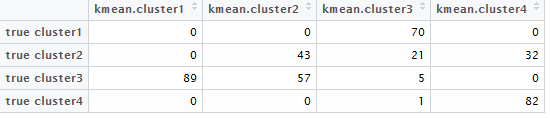


Table1.13 Confusion matrix for Run=2

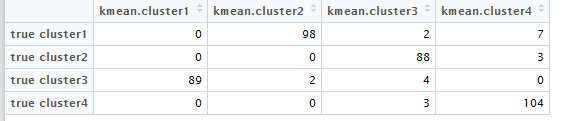
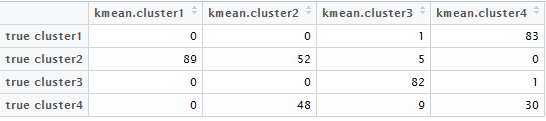
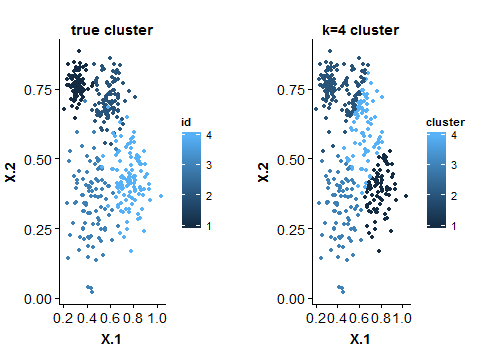
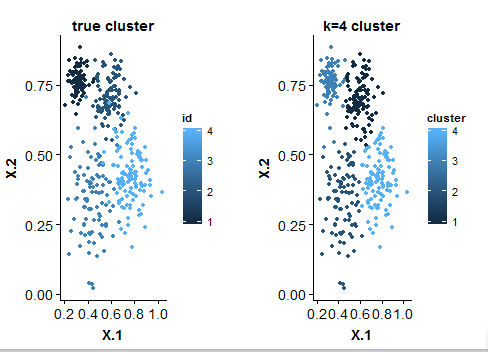
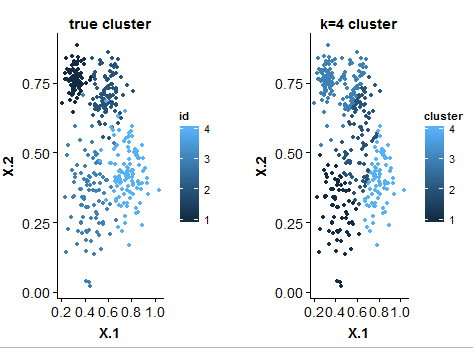


Table1.14 Confusion matrix for Run=3



d.Scatterplot for clustering of three runs

Figure1.3 Scatterplot for true cluster and K-means clustering ( K=4)

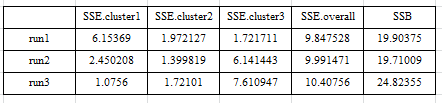


1. According to the confusion matrix for three runs (Table1.6, Table1.7, Table1.8), and scatterplot of clustering (Figure1.2). We can know for run1 and 3, there is no many point shifting from one cluster to others, the cluster result is similar .However, when compare to run2, there is some points switched among clusters. That means, the initial centroid points affects the clustering result.
2. Comparing the true cluster with k-means clustering, we can know that even though some points are clustered incorrectly .So the accuracy of k –means (k=4) cluster is not very low.

***K=3***

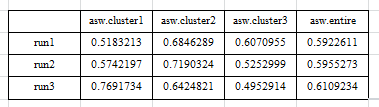
a. SSE, SSB for true cluster for k-means clustering

Table1.15 SSE for each cluster, overall SSE, SSB for K=3 in TwoDimHard Dataset



b.ASW

Table1.16 ASW for K=3 in TwoDimhHard Dataset



According to table 1.15 and table1.16, we can know that:

1. The overall SSE of K=3 has no significant difference among runs. However SSB varies among runs.
2. Comparing The overall SSE and SSB between k=3 and k=4 ,we can know that SSE value of k=3 is large than k=4, the SSB value is smaller than k=4 and also the true cluster, which mean k=3 doesn’t perform as well as k=4. K=4 is a better choice for this dataset.
3. The ASW of each cluster when k=3 varies from 0.49 to 0.76, the ASW of entire dataset is around 0.6. Compare with k=4, the ASW of entire dataset when k=3 is smaller than k=4,which mean k=3 doesn’t as well as k=4.

c. Confusion Matrix for 3 runs

Table1.17 Confusion matrix for Run=1

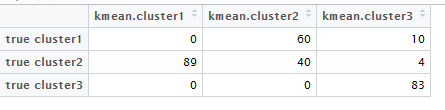


Table1.17 Confusion matrix for Run=2

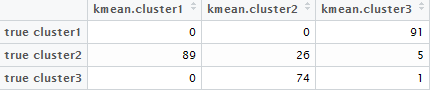
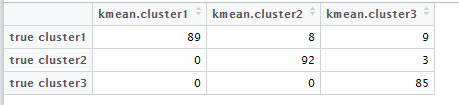
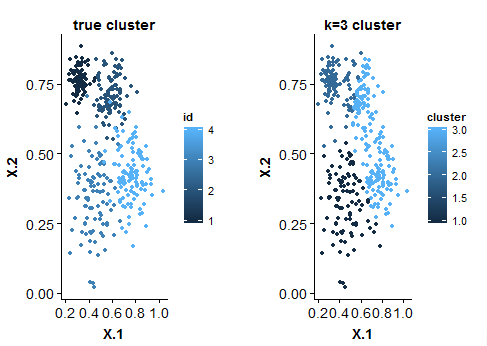
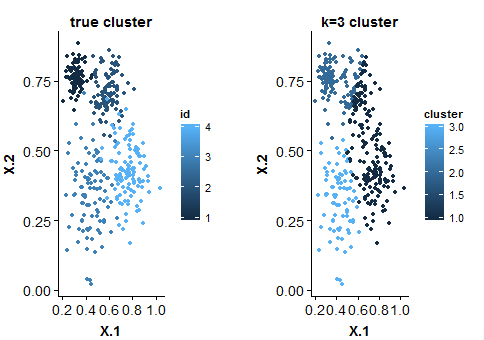


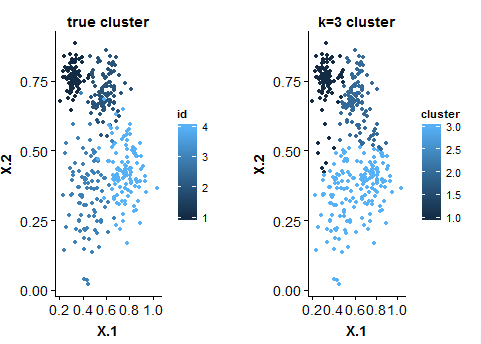
Table1.18 Confusion matrix for Run=3



d. Scatterplot for clustering of three runs

Figure1.3 Scatterplot for true cluster and K-means clustering ( K=3)





1. According to the confusion matrix for three runs (Table1.6, Table1.7, Table1.8), and scatterplot of clustering (Figure1.2). We can know for run1 and 2, there is no many point shifting from one cluster to others, the cluster result is similar .However, when compare to run3, there is some points switched among clusters. That means, the initial centroid points affect s the clustering result,
2. Comparing the true cluster with k-means clustering, we can know that many points are clustered incorrectly. K=3 might just split one or two cluster of the dataset. So the accuracy of k –means (k=3) cluster is as good as k=4.
   1. **K-means clustering of wine Dataset**

**Pre-procession:**

For wine Dataset, before clustering, we eliminate the class column and quality column. And normalize the residual attribute with min\_max normalization which is robust to outliers.

**K-means algorithm:**

For this dataset, k=2, 3, 4, 6 was experimented with tree runs. The SSE and SSB were calculated for each k, shown as following tables (Table1.19, Table1.20, Table1.21, and Table1.22).

Table1.19 SSE for each cluster, overall SSE, SSB for K=2 in Wine Dataset

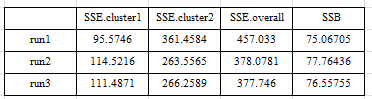


Table1.20 SSE for each cluster, overall SSE, SSB for K=3 in Wine Dataset

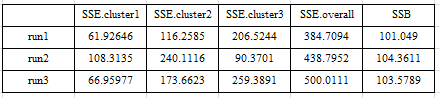


Table1.21 SSE for each cluster, overall SSE, SSB for K=4 in Wine Dataset

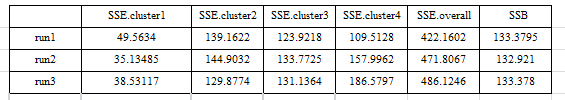
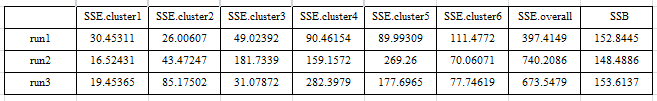


Table1.22 SSE for each cluster, overall SSE, SSB for K=6 in Wine Dataset

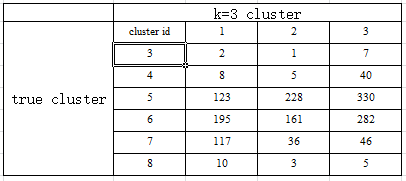
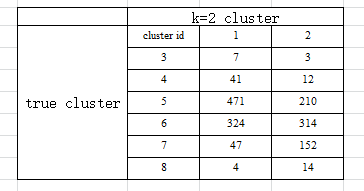


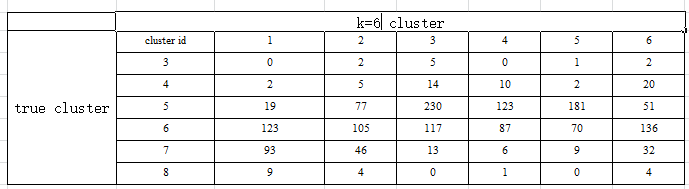
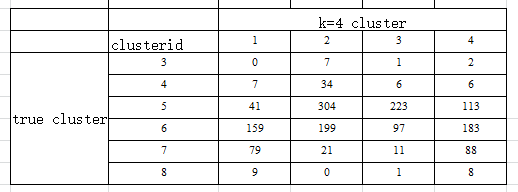
From these tables, we can see that:

1. SSE of each cluster and the overall SSE vary on runs, but SSB value has not significant difference among different runs.
2. SSB value increase when k value increase, which means higher k-value implies better clustering. However there is no obvious pattern on SSE value. Especially, when k=6, the range of the SSE.overall is very large.

Based on above SSE and SSB value, I would like to choose k=4

Table1.23 Confusion matrix for k=2,3,4,6 in Wine Dataset





From the confusion matrix, we can know that:

1. When k=2, the predicted cluster from k-means algorithm (cluster1 and cluster2) are spread over the true cluster), that means, it cannot even split out one correct cluster. The accuracy would be low.
2. When k =3, the predicted cluster from k-means algorithm is cluster1, cluster2, and cluster3. Most points are in true cluster 5,6,7 . So the cluster 3,4,8 will be omitted in this clustering.
3. When k =4, the predicted cluster from k-means algorithm is cluster1, cluster2, and cluster3,cluster4. For each cluster data in a true cluster will be clustered into 2 clusters in k-means algorithm. So, it hard to slip out one correct cluster.
4. When k=6, for most true cluster has one or two predicted clusters from k-means algorithm dominated.The accuracy is enhanced somehow because this.
   1. **Part2 Use off-the-shelf clustering method**

In this part, the k-means cluster (,i.e. kmeans()) and hierarchal clustering function (i.e. hclust() )inside R is used.

**2.1 K-means clustering**

We can use following code to determine k-value, which can create plot that shows the relationship between SSE and k –value. (For example, for twodimhard dataset, when k=1 to 4 , the SSE drop obviously, but when k>=4, there is no much different on SSE. So, we can chose k=4 for k-means clustering.)

wss <- (nrow(data)-1)\*sum(apply(data,2,var))   
for (i in 2:15) wss[i] <- sum(kmeans(data 1,   
 centers=i)$withinss)   
plot(1:15, wss, type="b", xlab="Number of Clusters",  
 ylab="Within groups sum of squares")

**2.1.1 K-means clustering of TwoDimEasy Dataset**

By using above code, we can obtain the following plot (Figure 2.1).In this plot, we can know than, the SSE decrease harshly from k=1 to k=2,but when k>=2, the SSE doesn’t have significant decrease. So in this case, we set k=2.

Figure2.1 SSE vs K-value for TwoDimEasy Dataset

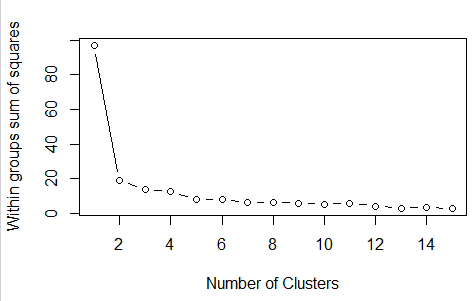


Table2.1 SSE for each cluster, overall SSE, SSB/SST for K=2 in TwoDimEasy Dataset

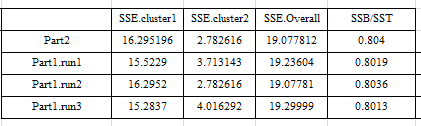
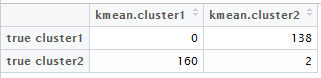


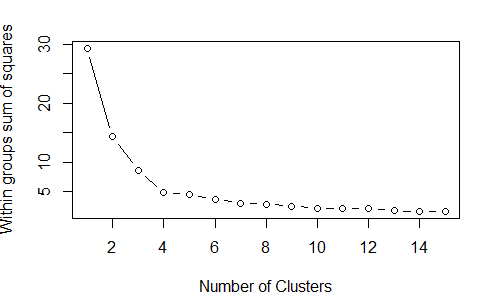
Table 2.2 Confusion matrix when k=2



In Table 2.1, we compare SSE value and SSB/SST value between the k-means algorithm created in part1 and the existing K-means function in R. From the output, we can know that these two k-means method reveals similar output. The SSE and SSB/SST are almost same. And also by comparing the confusion matrix of this part (Table2.2) with the confusion matrix generated by part, they are also almost same. SSB/SST is larger than 80%, which means this clustering method split data very well.

**2.1.2 K-means clustering of TwoDimHard Dataset**

Figure2.2 SSE vs K-value for TwoDimHard Dataset



In this plot, we can know than, the SSE decrease harshly from k=1 to k=4,but when k>=4, the SSE doesn’t have significant decrease. So in this case, we set k=4.

Table2.3 SSE for each cluster, overall SSE, SSB/SST for K=2 in TwoDimHard Dataset

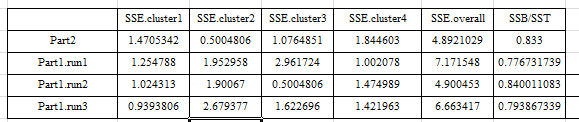
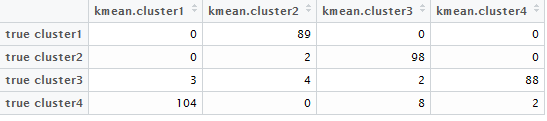


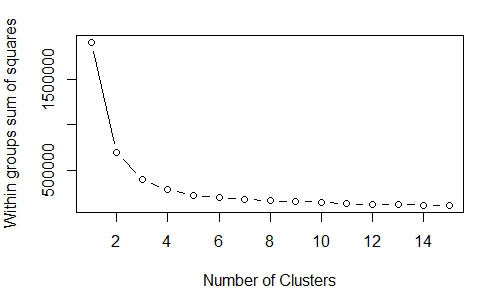
Table 2.4 Confusion matrix when k=4



In Table 2.3, we compare SSE value and SSB/SST value between the k-means algorithm created in part1 and the existing K-means function in R. From the output, we can know that these two k-means method reveals similar output. The SSE and SSB/SST has no big difference. And also by comparing the confusion matrix of this part (Table2.2) with the confusion matrix generated by part 1, they are also almost same. The accuracy of this clustering is (104+89+98+88)/400=94.75% . By comparing with the confusion matrix from k-means algorithm in part1, this performs better. SSB/SST is larger than 80%, which means this clustering method split data very well.

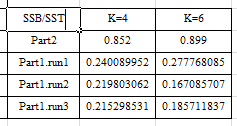
**2.1.3 K-means clustering of wine Dataset**

Figure2.3 SSE vs K-value for wine Dataset



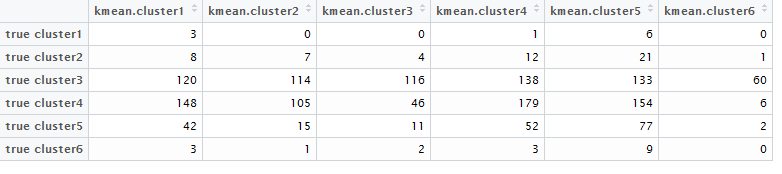
In this plot, we can know than, the SSE decrease harshly from k=1 to k=4,but when k>=4, the SSE doesn’t have significant decrease. So in this case, we set k=4, and k=6 to experiment.

Table2.5 SSB/SST for K=4,k=6 in wine Dataset



Based on the SSB/SST of k=4 and k=6, we can know that in this case the k-means algorithm in R is better than the algorithm SSB/SST of k=4 and k=6 are greater than 80%. However, SSB/SST obtained on part1 is only about 20%.

Table 2.6 Confusion matrix when k=6



Comparing with the confusion matrix between part1 algorithm and off-the-shelf algorithm, we can know these 2 different methods are similar. For true cluster1 is 3, 6, there is no dominating cluster, and for the point which in true cluster1,2,6 which has a few point , they are difficult to cluster correctly by this method . This clustering don’t perform very well on this dataset

**2.2 Hierarchical Clustering**

Hierarchical clustering code in R is shown as below. The distance matrix of the dataset need be calculated before clustering. In this part, the ward’s method is used.

d <- dist(twodimeasy1, method = "euclidean") # distance matrix  
fit <- hclust(d, method="ward")

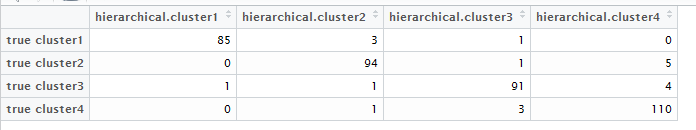
The clustering result for these three dataset and comparison to true cluster are shown in Confusion Matrix (Table2.7,Table2.8 and Table2.9).

Table 2.7 Confusion matrix for TwoDimEasy Dataset when k=2



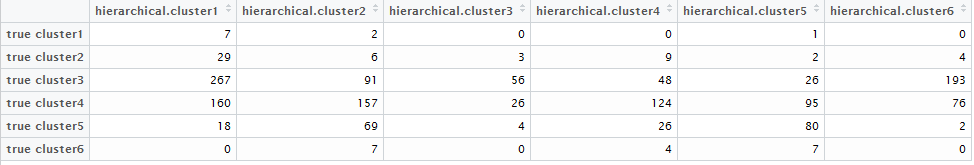
From the Table 2.7, we can know Hierarchical clustering performed very well. The accuracy is (138+161)/300=99.67%. Comparing to the k-means algorithm which is in part 1 and part, they all performed very well for this dataset.

Table 2.8 Confusion matrix for TwoDimHard Dataset when k=4



From the Table 2.7, we can know Hierarchical clustering performed very well. The accuracy is (85+94+91+110)/300=95%. Comparing to the k-means algorithm which is in part 1, this one performs better..

Table 2.9 Confusion matrix for Wine Dataset when k=6



Comparing with the confusion matrix between part1 algorithm and part 2 K-means function in R, we can know they are similar. For the point in true cluster is 3, 4, 5, there is no dominating cluster in hierarchal clustering. They spread into many cluster. This clustering doesn’t perform very well on this dataset