Homework 2: Clustering

ISYE 8803: Introduction to Analytics Modeling

Joseph Phillips, Kaustubh Mohite, and Sophia Smith

Georgia Institute of Technology

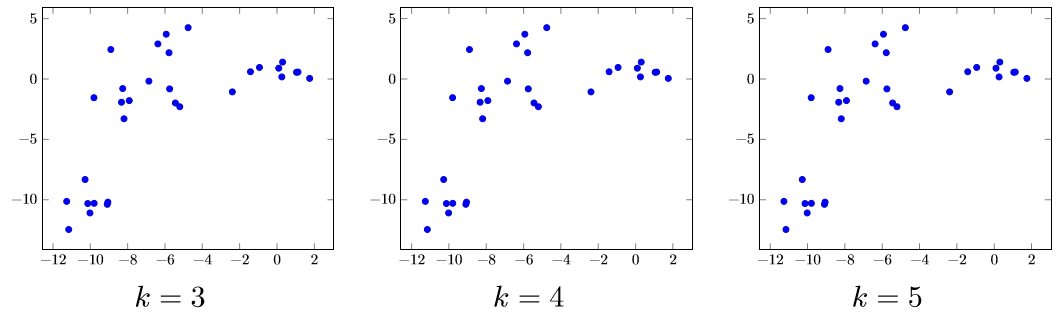
Problem 1.

Because of the variety of universities, it is often difficult for applicants to choose the best university for their needs. Before an applicant invests time and money researching different universities and applying to them, it may be prudent to use a clustering model to group all possible universities according to the applicant’s priorities. Some possible predictor variables for such a clustering model are:

1. Tuition per credit hour
2. University rankings (e.g. US News, Times Higher Education, ARWU – the Shanghai Ranking, QS, etc.)
3. Distance from applicant’s current location
4. Average student to instructor ratio
5. Average annual starting salary
6. Likert scale alumni feedback
7. Majors offered

Problem 2.

a.





b. Visually, four seems to be the best value of *k* for this graph. However, the problem being addressed and the type of data being used should also be considered when making this choice.

Problem 3.

a. Model for Classification:

Where:

b. Model for Clustering:

c. The basic data for clustering does not include values.

d. The classification model requires values, and the clustering model does not.

e. 1. Clustering

2. Classification

3. Clustering

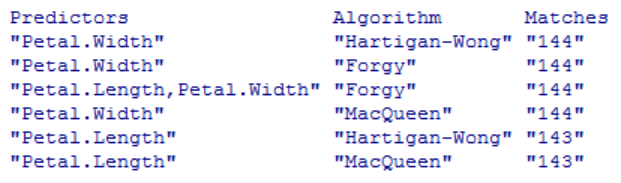
4. Classification

Problem 4.

For this clustering problem, the iris dataset which is provided in the R dataset library was used. From this dataset, four variables were evaluated as predictors of clusters within the data. These variables were measurements of septal lengths and widths and petal lengths and widths for 150 irises. The fifth variable in the iris dataset indicated the type of iris for each observation. This variable was not used to identify clusters. However, it was used to test the accuracy of the clusters.

Python was utilized to generate R script which tested every combination of the four predictor variables with each of the provided algorithms for the kmeans function. The results for each permutation were compared to the actual iris types observed. The iris type which comprised the majority of a cluster was considered the match for that cluster. Observations which were grouped in a cluster for which their iris types were not the majority were considered mismatches. The permutations of different combinations of predictor variables and algorithms were sorted by the number of correct matches. Based on the evidence, petal width or petal width and petal length are the best combinations of predictors for this dataset. Both combinations produced groups that matched 144 observations (Table 1).

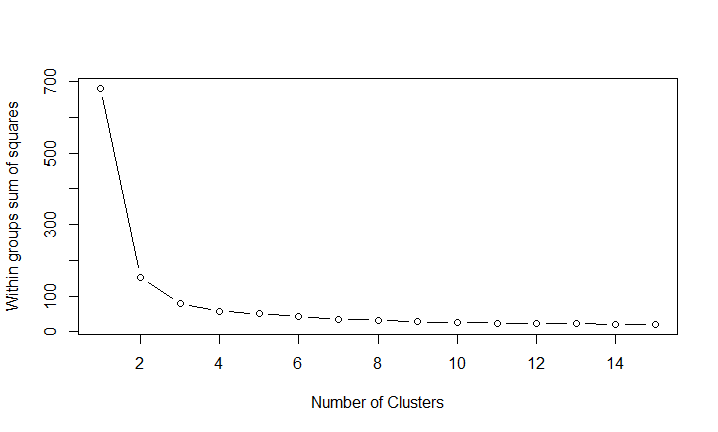
Table 1. Top 6 Combinations of Predictors for Clustering (k = 3) the Iris Dataset



Because the number of Iris types is known to be three, the best value for *k* for this dataset is three. Using values of two or four for *k* for this dataset would cause a greater number of mismatches. Though the exact groups included in the dataset will usually not be known when using a clustering model, the context of the problem would probably indicate a best value for *k*. However, a technique suggested by Rob Kabacoff in his book ‘R in Action’ (excerpt published on rbloggers.com) can be used to determine a value for k if no knowledge of a correct number of groups in a dataset were available.

For various values of k, we calculate for each group (cluster) the sum of squared distances from the centre. We then plot k versus the sum of within-group squared distances as a line and dot graph. The value of k for which there is the most pronounced bend in the graph (i.e. the sum of squared distances reduces the most sharply) is a good candidate.

**Figure 1. Selection of appropriate value of k – number of clusters**

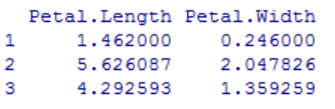
****

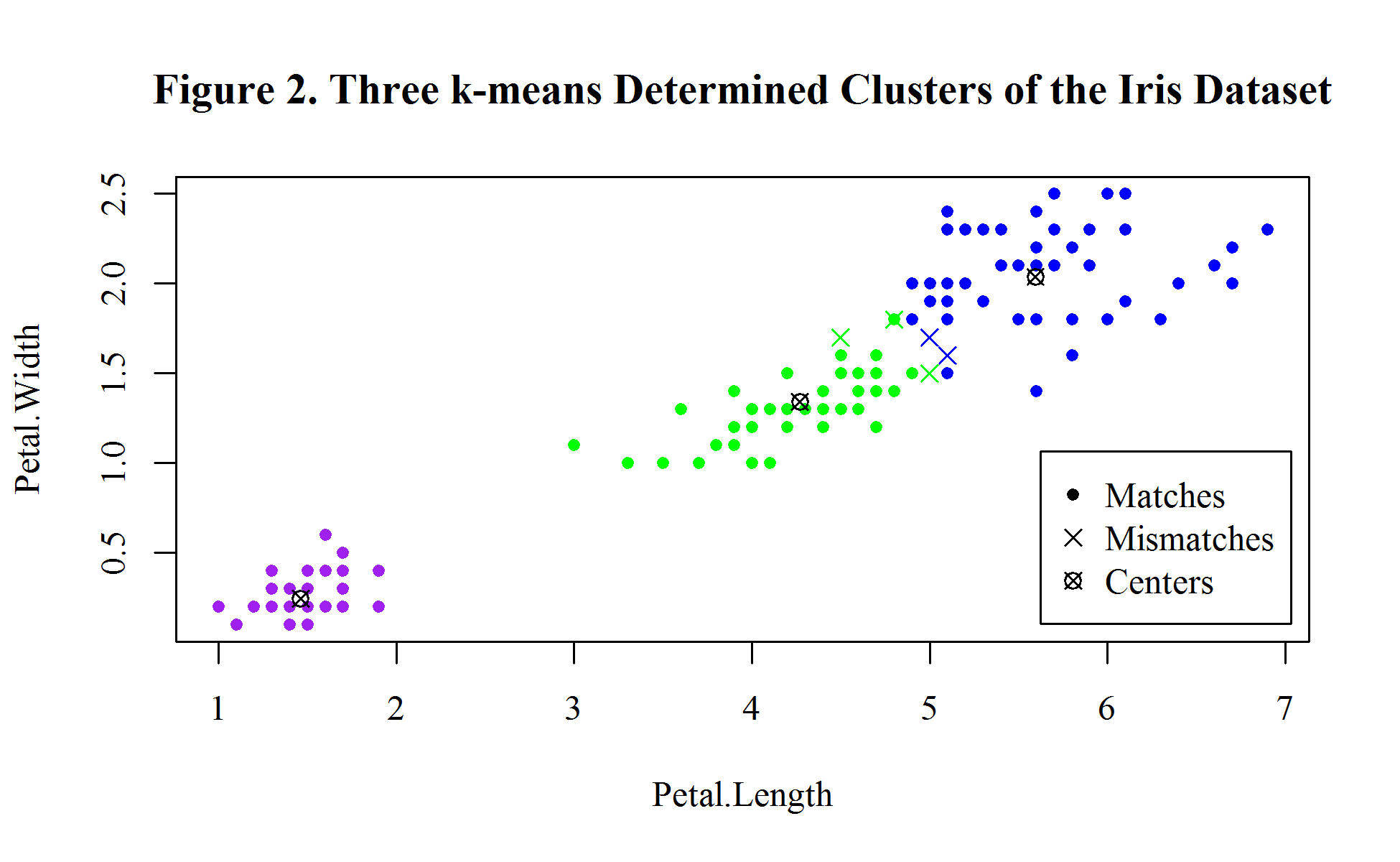
In our case, we find that the sum of squared distances decreases the most sharply between k = 1 and k = 2. However, there is still significant improvement achieved by changing k from 2 to 3. Beyond this, further increase in the value of k leads to only marginal improvement.

Hence we select k = 3.

The clustering model using petal length and petal width as predictors with the Forgy algorithm and three clusters was one of the four most accurate models. This model had an accuracy of 96%. The centers for the clustered groups are listed in Table 2. Figure 2 displays the groups created by this clustering model. The different colors in Figure 2 indicate the different groups. The X’s in Figure 2 indicate members of a particular group that were incorrectly clustered with that group. All of the mismatches occurred within the region that represents the largest of the versicolor irises (majority green) and the smallest of the virginica irises (majority blue). The arrow points to two observations from the iris dataset which demonstrate why a perfect clustering may not be possible. The observations are collocated and both were grouped into the green group which is comprised primarily of versicolor irises. However, the X indicates that one of these two observations is a virginica iris. Because these two observations are collocated, they will always be clustered together even though they are two different types of iris. Conditions like this one are likely to occur when clustering other data.

Table 2. Cluster Centers for the Iris Dataset by Petal Length and Width.





Appendix 1: R code

library("datasets")

iristype <- iris[,5]

#Run Python generated R-code

bestxi <- head(testmatrix[sort.list(as.numeric(testmatrix[,3]),decreasing = TRUE), ])

colnames(bestxi) <- c("Predictors","Algorithm","Matches")

kdata <- iris[,c(3:4)]

clustiris <- kmeans(kdata,3,iter.max = 1000,,algorithm = "Forgy")

tclust <- matrix(,150,4)

tclust[,1] <- iristype

tclust[,2] <- clustiris$cluster

key <- aggregate(tclust[,2]~tclust[,1],,median)

tclust[,3] <- ifelse(tclust[,1]==key[1,1],key[1,2],

ifelse(tclust[,1]==key[2,1],key[2,2],key[3,2]))

tclust[,4] <- ifelse(tclust[,2]==tclust[,3],1,0)

sum(tclust[,4])

aggregate(tclust[,4]~tclust[,1],,sum)

plotdata1 <- cbind(iris,tclust[,c(2,4)])

plotdata2 <- matrix(,3,7)

colnames(plotdata2) <- colnames(plotdata1)

plotdata2[,c(3:4)] <- clustiris$centers

plotdata2[,7] <- 9

plotdata <- rbind(plotdata1,setNames(plotdata2,names(plotdata1)))

attach(plotdata)

#png("Figure 2.png", width = 1950, height = 1200, res = 300)

#par(family="serif")

plot(Petal.Length,Petal.Width,

main = "Figure 2. Three k-means Determined Clusters of the Iris Dataset",

col = ifelse(plotdata[7] == 9, "black",

ifelse(plotdata[6] == 1, "green",

ifelse(plotdata[6] == 2, "purple","blue"))),

pch = ifelse(plotdata[7] == 9, 13,

ifelse(plotdata[7] == 0, 4,20)),

)

legend(5.875,1.18,c("Hits","Misses","Centers"),pch = c(20,4,13))

#dev.off()

Appendix 2: Python code

var\_list = [1,2,3,4]

master\_list = [[1],[2],[3],[4]]

count = -1

while len(master\_list) <= (2\*\*4)-2:

count += 1

for i in var\_list:

if i not in master\_list[count]:

new\_var = sorted(master\_list[count] + [i])

if new\_var not in master\_list:

master\_list.append(new\_var)

x\_list = ["Septal.Length","Septal.Width","Petal.Length","Petal.Width"]

xi\_list = []

str\_list = []

for i in master\_list:

count = -1

str\_var = "c("

xi\_var = ""

while count < (len(i)-1):

count += 1

if count < (len(i) - 1):

str\_var += str(i[count]) + ","

xi\_var += x\_list[i[count]-1] + ","

else:

str\_var += str(i[count]) + ")"

xi\_var += x\_list[i[count]-1]

str\_list.append(str\_var)

xi\_list.append(xi\_var)

alg\_list = ["Hartigan-Wong", "Forgy", "MacQueen"]

count = 0

line3 = "tclust <- matrix(,150,4)\n"

line4 = "tclust[,1] <- iristype\n"

line5 = "tclust[,2] <- clustiris$cluster\n"

line6 = "key <- aggregate(tclust[,2]~tclust[,1],,median)\n"

line7 = "tclust[,3] <- ifelse(tclust[,1]==key[1,1],key[1,2],\n"

line8 = " ifelse(tclust[,1]==key[2,1],key[2,2],key[3,2]))\n"

line9 = "tclust[,4] <- ifelse(tclust[,2]==tclust[,3],1,0)\n"

with open("clustertest.r",'w') as tmat:

tmat.write("testmatrix <- matrix(,nrow = 45,ncol = 3)\n")

for ai in alg\_list:

count2 = -1

for i in str\_list:

count += 1

count2 += 1

line1 = "kdata <- iris[,"+i+"]\n"

line2 = 'clustiris <- kmeans(kdata,3,iter.max = 1000,, algorithm = ' + '"' + ai + '")\n'

line10 = 'testmatrix['+str(count)+',1] <- ' + '" + xi\_list[count2] + '"\n'

line11= 'testmatrix['+str(count)+',2] <- "' + ai + '"\n'

line12 = "testmatrix["+str(count)+",3] <- sum(tclust[,4])\n"

for ii in list(range(12)):

tmat.write(eval("line"+str(ii+1)))

Appendix 3: R-code adapted from [www.r-bloggers.com/k-means-clustering-from-r-in-action/](http://www.r-bloggers.com/k-means-clustering-from-r-in-action/)

wssplot <- function(data, nc=15, seed=1234){

wss <- (nrow(data)-1)\*sum(apply(data,2,var))

for (i in 2:nc){

set.seed(seed)

wss[i] <- sum(kmeans(data, centers=i)$withinss)}

plot(1:nc, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")}