K-Means Clustering Using R

Kent Ng

May 24, 2018

The iris data set iris.txt contains 150 data points, each with four predictor variables and one categorical response. The predictors are the width and length of the sepal and petal of flowers and the response is the type of flower. The data is available from the R library datasets and can be accessed with iris once the library is loaded. It is also available at the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Iris (https://archive.ics.uci.edu/ml/datasets/Iris)). The response values are only given to see how well a specific method performed and should not be used to build the model. We will be using K means the cluster the points as well as possible.

As best practice, let's set seed value.

```
set.seed(42)
```

Next, we have to load the data from the given .txt file into our "data_4" variable.

```
data_4 <- read.table("4.2irisSummer2018.txt")
head(data_4)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Let's take a look and see the structure and distribution of our data.

```
str(data_4)
```

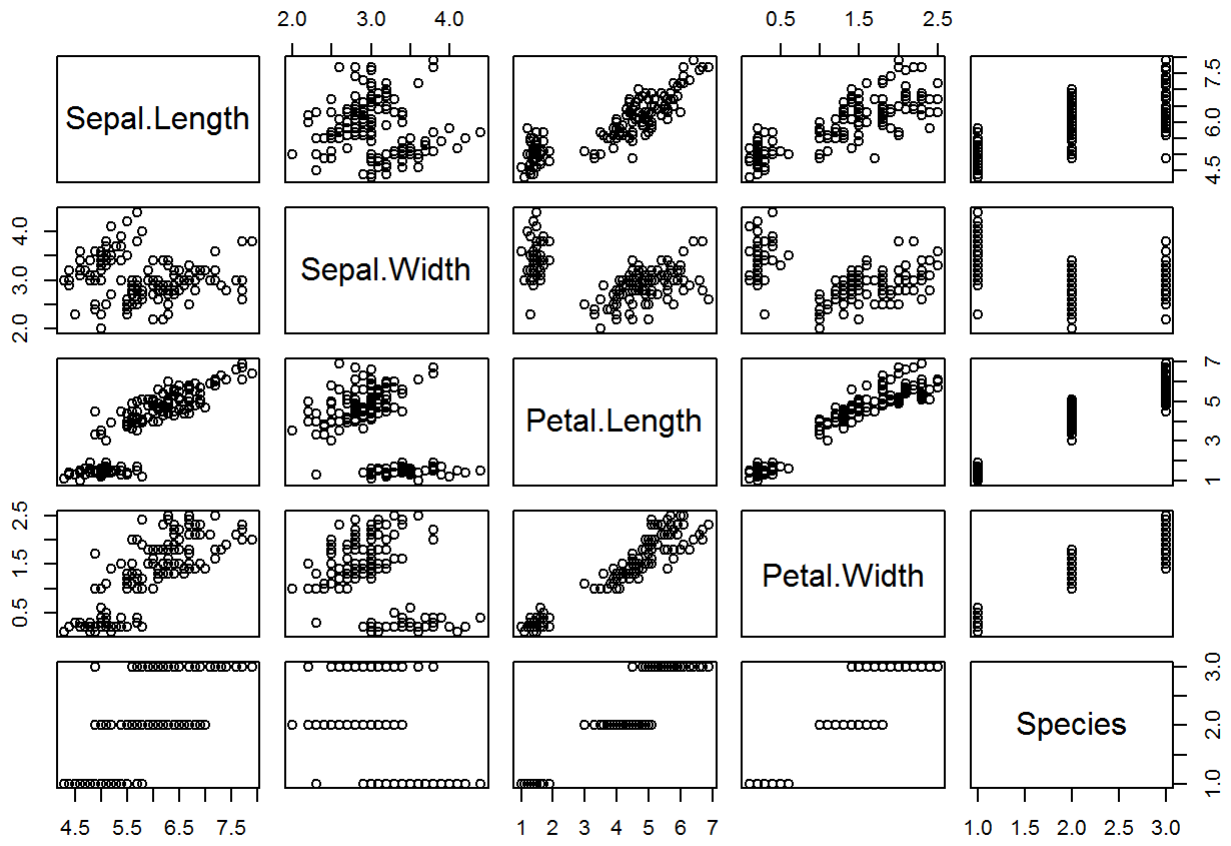
```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(data_4)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

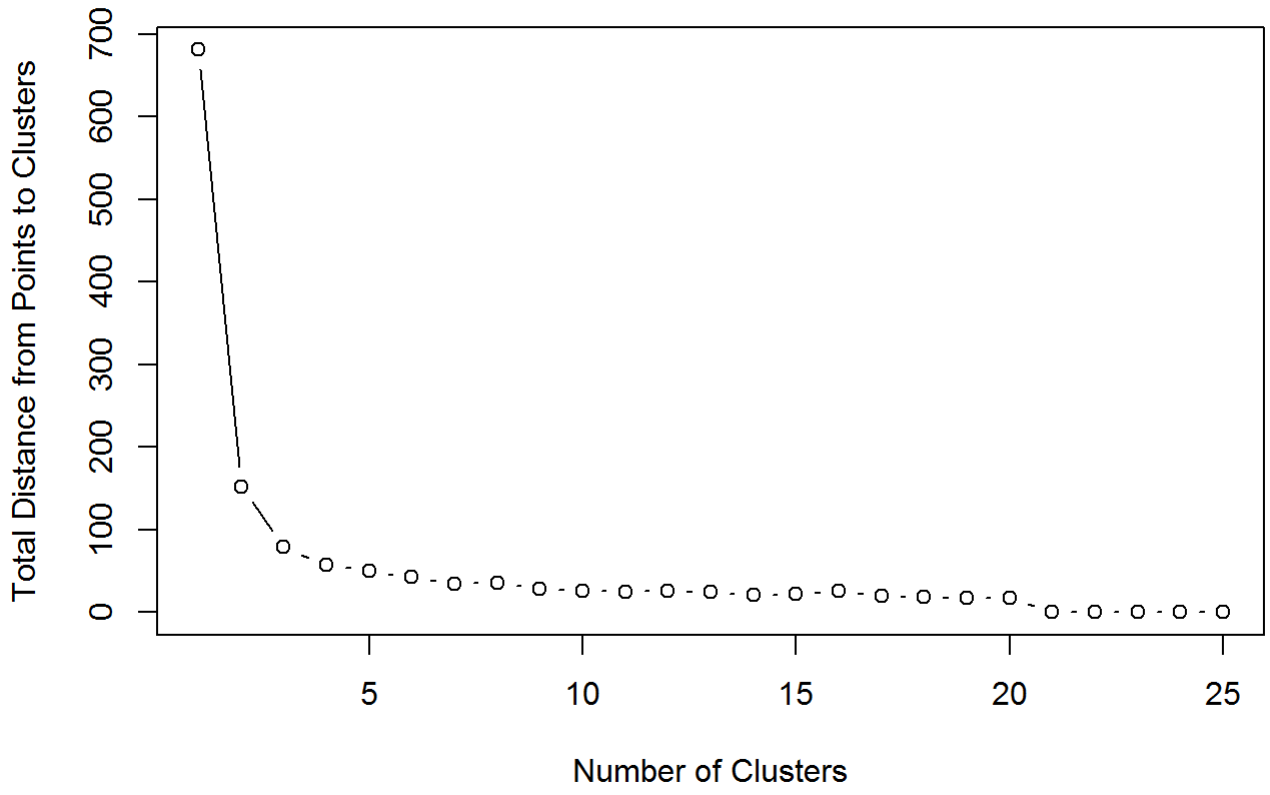
Next, let's add a plot to better understand and visualize the data. Based on these plots, we can see which pair of predictors have strong correlation with each other.

```
plot(data_4)
```



Let's call the k-means clustering method. Since we are trying to classify 3 different types of species, it makes sense to set k =3. However, technically, we're not supposed to know what the response variable is when using clustering techniques (hence why this is a type of unsupervised learning). So intead, let's leverage the elbow diagram (introduced in the Week 2 lectures) to determine a good value for k.

```
distances<-rep(0,25)
for(k_clusters in 1:20){
  cluster <- kmeans(data_4[,1:4],k_clusters)
  distances[k_clusters]<-cluster$tot.withinss
}
plot(distances, xlab="Number of Clusters",ylab="Total Distance from Points to Clusters", type =
"b", frame = TRUE)
```



Looking at the elbow diagram, it also makes sense to use k = 3; after k = 3, the benefits realized from adding additional clusters become comparatively small.

Now that we've selected a k value, we can also try out different combinations of predictors to see which clustering model fits our data the best.

Let's first select Sepal.Length and Sepal.Width as our initial set of predictors.

```
kmeans_cluster1<-kmeans(data_4[,1:2],centers=3,nstart=30)
head(kmeans_cluster1)
```

```
## $cluster
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   1   1   1   2
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   1   2   1   2   1   2   2   2   2   2   2   1   2   2   2   2   2   2
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   2   2   1   1   1   1   2   2   2   2   2   2   2   2   1   2   2   2
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   2   2   2   2   2   2   2   2   2   2   1   2   1   1   1   1   2   1
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   1   1   1   1   1   2   2   1   1   1   1   2   1   2   1   2   1   1
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   2   2   1   1   1   1   1   2   2   1   1   1   2   1   1   1   2   1
## 145 146 147 148 149 150
##   1   1   2   1   1   2
##
## $centers
##   Sepal.Length Sepal.Width
## 1     6.812766    3.074468
## 2     5.773585    2.692453
## 3     5.006000    3.428000
##
## $totss
## [1] 130.4753
##
## $withinss
## [1] 12.6217 11.3000 13.1290
##
## $tot.withinss
## [1] 37.0507
##
## $betweenss
## [1] 93.42456
```

We can see how well the clustering predicted the flower types by comparing the results to the original Species data. Note once again that, in a real world situation, we wouldn't know the response variable (Species in this case) and so normally we wouldn't be able to do this type of validation.

```
table(kmeans_cluster1$cluster,data_4$Species)
```

```
##
##     setosa versicolor virginica
##   1      0         12        35
##   2      0         38        15
##   3     50          0         0
```

We can see that the data points for Setosa were grouped into Cluster 1. On the other hand, it is not as clear cut for the species Versicolor and Virginica. One can assume that data points for Versicolor were grouped into Cluster 2 and, likewise, data points for Virginica into Cluster 3. If that is the case, this means that 12 points were incorrectly classified as Versicolor and 15 points were incorrectly classified as Virginica. Let's try other sets of predictors to see if we can get better results.

This time, let's try using Petal.Length and Petal.Width as our predictors

```
kmeans_cluster2<-kmeans(data_4[,3:4],centers=3,nstart=30)
table(kmeans_cluster2$cluster,data_4$Species)
```

```
##
##     setosa versicolor virginica
##   1      0          2        46
##   2      0         48         4
##   3     50          0         0
```

Here we can see that the results are much better! We can confidently see that Cluster 1,2 and 3 contain the data points for Virginica, Setosa and Versicolor respectively. Here, we only have 6 points that were misclassified.

But what if we add more predictors in our clustering model? To answer this question, let's try Using Sepal.Length, Sepal.Width, Petal.Length and Petal.Width all as predictors

```
kmeans_cluster3<-kmeans(data_4[,1:4],centers=3,nstart=30)
table(kmeans_cluster3$cluster,data_4$Species)
```

```
##
##     setosa versicolor virginica
##   1      0          2        36
##   2      0         48        14
##   3     50          0         0
```

Using this set of predictors, we have 16 misclassifications. This shows that having more predictors does not mean the clustering prediction will be better! One explanation is that introducing uncorrelated predictors doesn't really help or improve our clustering model, yet it introduces more noise in our clustering.

Let's try other combinations of predictors

```
#Sepal.Length and Petal.Length as predictors - Result: 18 misclassifications.
kmeans_cluster5<-kmeans(data_4[,c(1,3)],centers=3,nstart=30)
table(kmeans_cluster5$cluster,data_4$Species)
```

```
##
##     setosa versicolor virginica
##   1     50          1         0
##   2      0         45        13
##   3      0          4        37
```

```
#Sepal.Length and Petal.Width as predictors - Result: 28 misclassifications.
kmeans_cluster6<-kmeans(data_4[,c(1,4)],centers=3,nstart=30)
table(kmeans_cluster6$cluster,data_4$Species)
```

```
##
##     setosa versicolor virginica
##   1      0         37        15
##   2     50          4         0
##   3      0          9        35
```

```
#Sepal.Width and Petal.Length as predictors - Result: 11 misclassifications.
kmeans_cluster7<-kmeans(data_4[,c(2,3)],centers=3,nstart=30)
table(kmeans_cluster7$cluster,data_4$Species)
```

```
##
##     setosa versicolor virginica
##   1     50          0         0
##   2      0         48         9
##   3      0          2        41
```

```
#Sepal.Width and Petal.Width as predictors - Result: 11 misclassifications.
kmeans_cluster8<-kmeans(data_4[,c(2,4)],centers=3,nstart=30)
table(kmeans_cluster7$cluster,data_4$Species)
```

```
##
##     setosa versicolor virginica
##   1     50          0         0
##   2      0         48         9
##   3      0          2        41
```

```
#Sepal.Length, Sepal.Width and Petal.Length as predictors - Result: 18 misclassifications.
kmeans_cluster4<-kmeans(data_4[,1:3],centers=3,nstart=30)
table(kmeans_cluster4$cluster,data_4$Species)
```

```
##
##     setosa versicolor virginica
##   1      0         45        13
##   2      0          5        37
##   3     50          0         0
```

```
#Sepal.Width, Petal.Length, Petal.Width as predictors - Result: 7 misclassifications.
kmeans_cluster9<-kmeans(data_4[,2:4],centers=3,nstart=30)
table(kmeans_cluster9$cluster,data_4$Species)
```

```
##
##     setosa versicolor virginica
##   1     50          0         0
##   2      0         48         5
##   3      0          2        45
```

```
#Sepal.Width, Petal.Length, Petal.Width as predictors - Result: 26 misclassifications.
kmeans_cluster10<-kmeans(data_4[,c(1,2,4)],centers=3,nstart=30)
table(kmeans_cluster10$cluster,data_4$Species)
```

```
##
##     setosa versicolor virginica
##   1      0         11        35
##   2     50          0         0
##   3      0         39        15
```

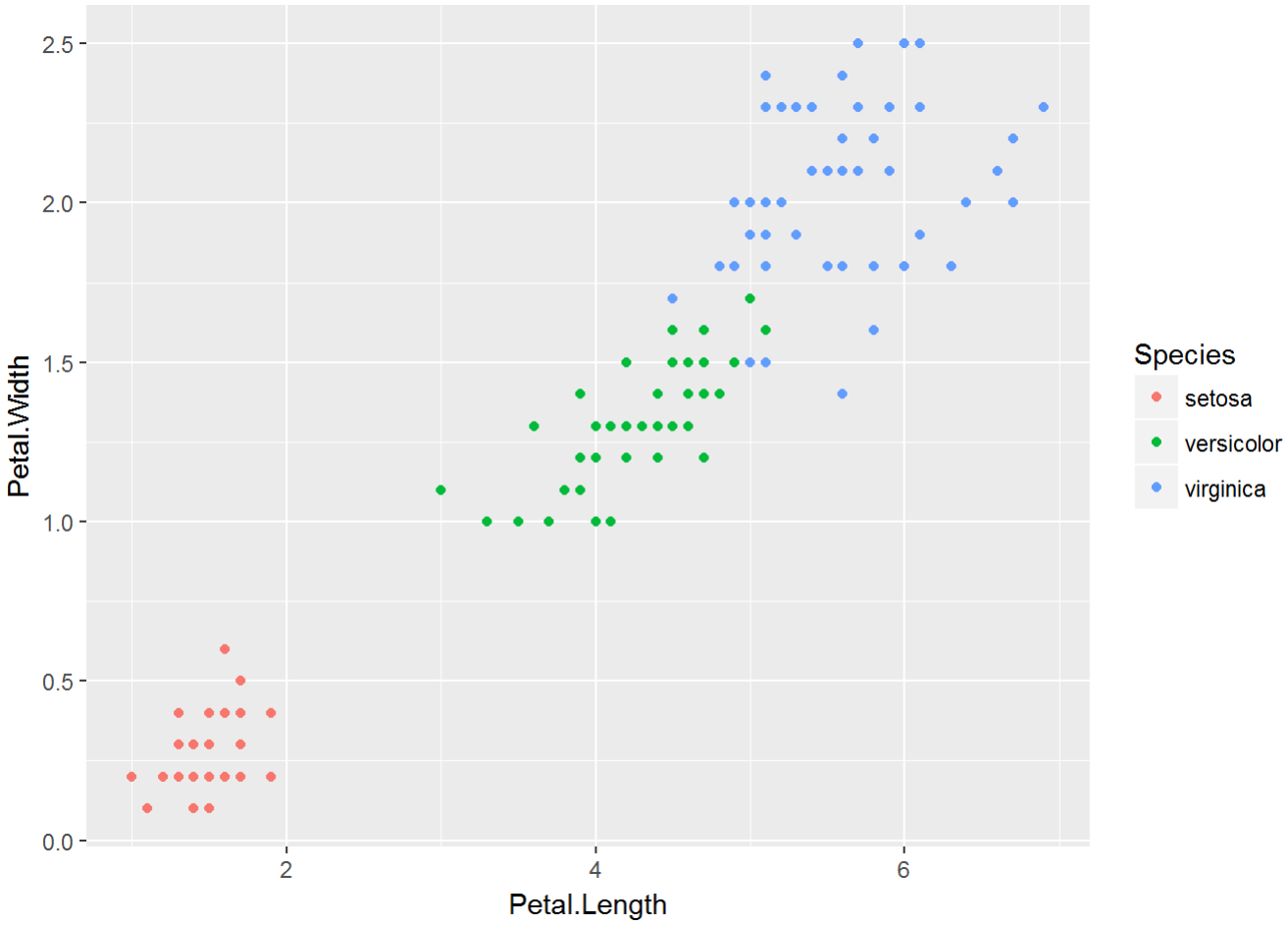
```
#Sepal.Width, Petal.Length, Petal.Width as predictors - Result: 16 misclassifications.
kmeans_cluster11<-kmeans(data_4[,c(1,3,4)],centers=3,nstart=30)
table(kmeans_cluster11$cluster,data_4$Species)
```

```
##
##     setosa versicolor virginica
##   1     50          0         0
##   2      0         48        14
##   3      0          2        36
```

Therefore, we can see that using Petal.Length and Petal.Width as predictors produced the best clustering for predicting the flower types. K was set to 3 given that it was the sweet spot in our elbow diagram, indicating that it is the most “bang for the buck” from a computation perspective. Overall, the clustering had 6 misclassifications out of 150 points (96% accuracy). Note that for the purpose of this question, we will not be testing the accuracy of our model using a test dataset.

We can use ggplot to visually confirm that the cluster model (using petal length and width as predictors) indeed did a good job in grouping the data by flower species.

```
library(ggplot2)
ggplot(data_4,aes(Petal.Length,Petal.Width,color=Species))+geom_point()
```


2.5
2.0
1.5
1.0
0.5
0.0
Petal.Width
2
4
6
Petal.Length
Species
setosa
versicolor
virginica