Introduction to K-means Clustering

Aleksandra Ma

December 3, 2017

Introduction

When I was doing the CS61A homework, I was asked to make a map showing a person's favorite restaurant rating through python. In this project, Berkeley is segmented into regions, where each region is shaded by the predicted rating of the closest restaurant. The idea is very interesting, reminding me of the k-means clustering in statistics. And post 2 gave me an opportunity to dig deeper in to K-means clustering and how we could do it through R.

Background Information on K-means Clustering

K-means clustering is a kind of unsupervised learning for data without defined categories or groups. The goal of this algorithm is to find patterns in data through grouping the data with the number of groups represented by the variable K. In other words, it aims to partition n observations into k clusters. The algorithm randomly assigns each observation to a cluster, and finds the centroid of each cluster. Then it iterates through two steps:

- Reassign data points to the cluster whose centroid is closest.
- Calculate new centroid of each cluster.

These two steps will be repeated till variation cannot be reduced any further within cluster. The within cluster variation is calculated as the sum of euclidean distance between the data points and their respective cluster centroids.

Example in R

We'll use the example in R using *Chatterjee-Price Attitude Data* from the dataset built in R. The dataset is a survey of clerical employees of a large financial organization aggregated from questionnaires of 35 employees for each of the 30 randomly selected departments. The numbers give the percent proportion of favorable responses to 7 questions in each department.

```
#Load necessary data
library(datasets)
str(attitude)
```

```
## 'data.frame':    30 obs. of  7 variables:
##  $ rating    : num  43 63 71 61 81 43 58 71 72 67 ...
##  $ complaints: num  51 64 70 63 78 55 67 75 82 61 ...
##  $ privileges: num  30 51 68 45 56 49 42 50 72 45 ...
##  $ learning  : num  39 54 69 47 66 44 56 55 67 47 ...
##  $ raises    : num  61 63 76 54 71 54 66 70 71 62 ...
##  $ critical  : num  92 73 86 84 83 49 68 66 83 80 ...
##  $ advance   : num  45 47 48 35 47 34 35 41 31 41 ...
```

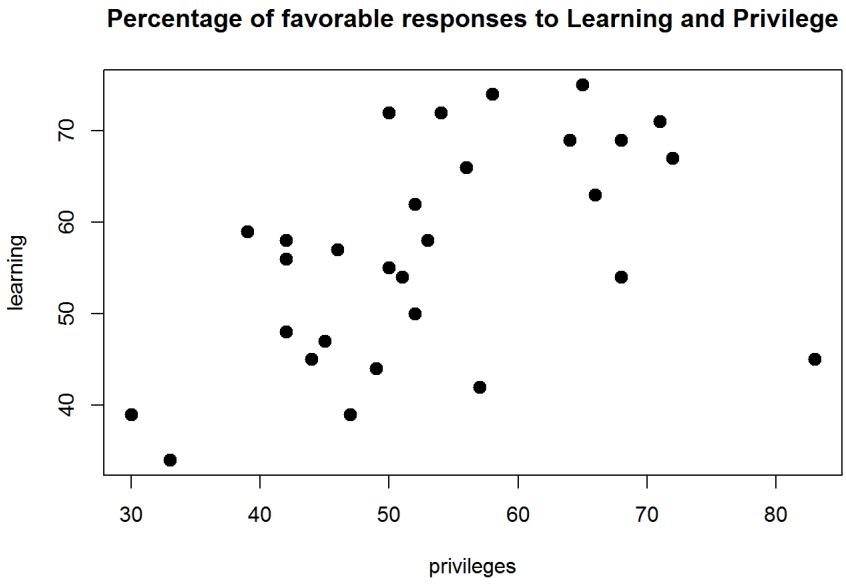
```
#summarise data
summary(attitude)
```

```
##      rating        complaints     privileges       learning
##  Min.   :40.00   Min.   :37.0   Min.   :30.00   Min.   :34.00
##  1st Qu.:58.75   1st Qu.:58.5   1st Qu.:45.00   1st Qu.:47.00
##  Median :65.50   Median :65.0   Median :51.50   Median :56.50
##  Mean   :64.63   Mean   :66.6   Mean   :53.13   Mean   :56.37
##  3rd Qu.:71.75   3rd Qu.:77.0   3rd Qu.:62.50   3rd Qu.:66.75
##  Max.   :85.00   Max.   :90.0   Max.   :83.00   Max.   :75.00
##      raises          critical        advance
##  Min.   :43.00   Min.   :49.00   Min.   :25.00
##  1st Qu.:58.25   1st Qu.:69.25   1st Qu.:35.00
##  Median :63.50   Median :77.50   Median :41.00
##  Mean   :64.63   Mean   :74.77   Mean   :42.93
##  3rd Qu.:71.00   3rd Qu.:80.00   3rd Qu.:47.75
##  Max.   :88.00   Max.   :92.00   Max.   :72.00
```

As mentioned above, this data gives the percent of favorable responses for each department. For example, in summary of the data, we can see that for *learning* among the 30 departments, the department with the least favorable responses only have 34% of their responses as favorable, whereas the department with the most favorable responses have 75% of their responses favorable.

For simplicity, we'll take a subset of the attitude dataset and consider only two variables in our example. We will cluster the attitude dataset of both *learning* and *privileges* with the responses from all 30 departments and try to understand whether there are commonalities among certain departments when it comes to these two variables.

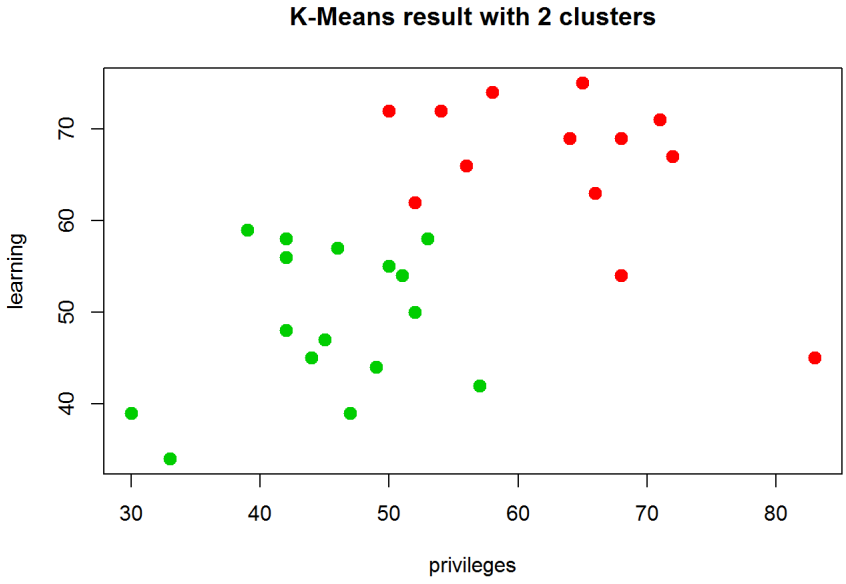
```
#Subset the attitude data
newdata = attitude[,c(3,4)]

#Plot subset data
plot(newdata, main = "Percentage of favorable responses to Learning and Privilege", pch = 20, cex = 2)
```



K-means Clustering When k = 2

As shown above, we can see how each department's score behave across "privilege" and "learning." Now we can apply K-means clustering and assign each department to a specific number of clusters that are "similar." We'll use the *kmeans* function from R base package:

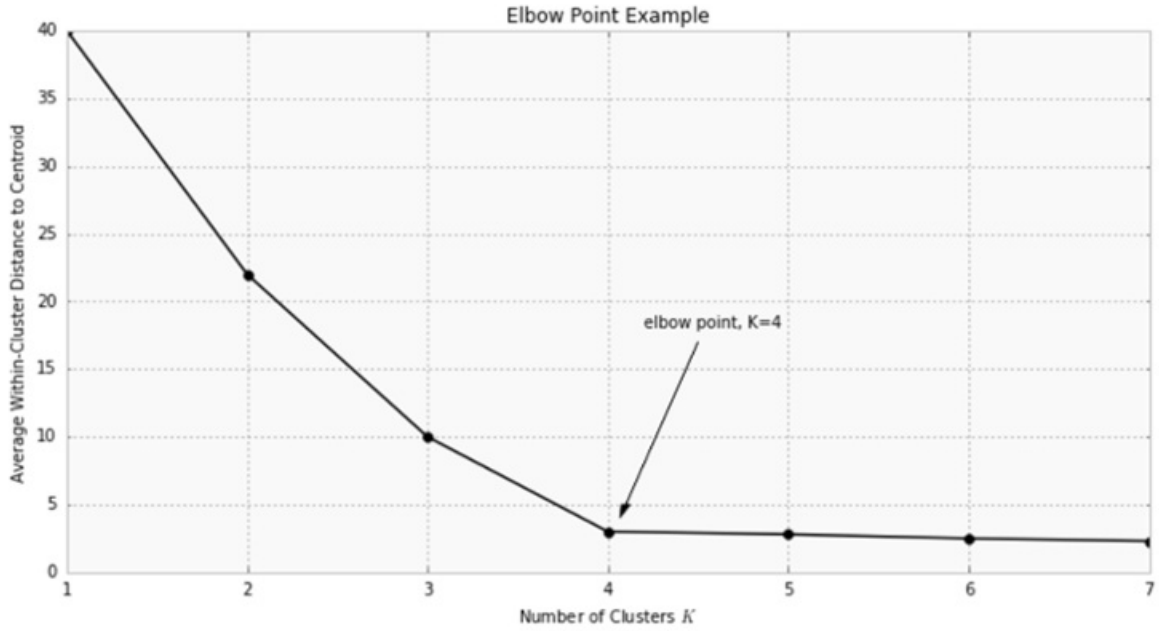
```
# Perform K-Means Clustering with 2 clusters
set.seed(7)
km1 = kmeans(newdata, 2, nstart = 100)

#Plot results
plot(newdata, col = (km1$cluster + 1), main = "K-Means result with 2 clusters", pch = 20, cex = 2)
```



How do we choose K?

We don't always have a pre-determined K and we don't always want K to be 2 as in our example above. We could try to choose the number of clusters by visually inspecting our data points, but there is a lot of ambiguity in this process unless it's a really simple data set. Sometimes it's not always bad because it is unsupervised learning and certain degree of inherent subjectivity in the labeling process is inevitable, but there is a more technical way of choosing the right K value.

One of the metrics that is used to choose the right K is through comparing the mean distance between data points and their cluster centroid. But through this way, increasing K will always decrease the mean distance and it will reach zero when the number of data points is the same as K. Thus, this metric cannot be used as the sole target. Instead, the mean distance to the centroid as a function of K is plotted and the "elbow point," where the rate of decrease sharply shifts, can be used to approximately determine K.

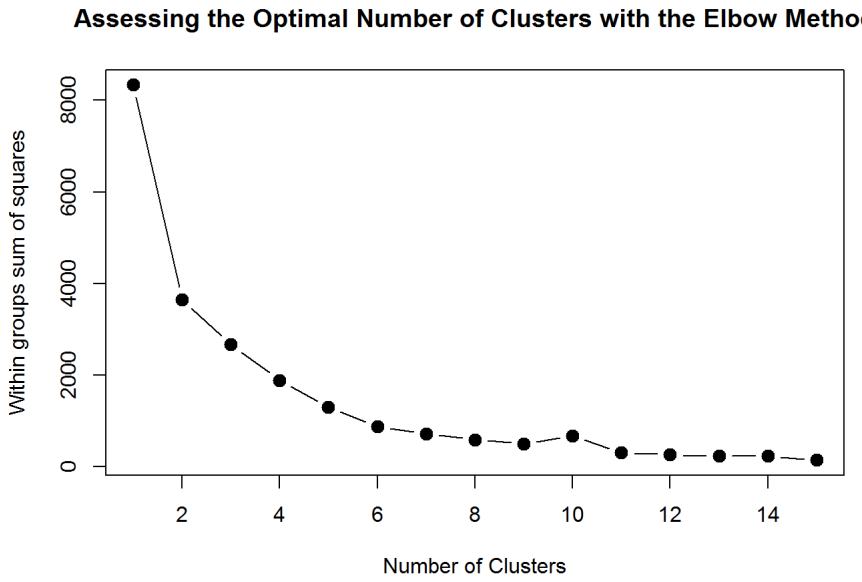


In the example above, as we increase the number of clusters K, there is an elbow point at k = 4 where the slope of the graph suddenly decreases. In this case, 4 is our elbow point and will be our K.

Back to Our Example

Now it's time to do the Elbow method on our own dataset.

```
# Check for the optimal number of clusters given the data

wss <- (nrow(newdata)-1)*sum(apply(newdata,2,var))
  for (i in 2:15) wss[i] <- sum(kmeans(newdata, centers = i)$withinss)

plot(1:15, wss, type = "b", xlab= "Number of Clusters",
     ylab="Within groups sum of squares",
     main="Assessing the Optimal Number of Clusters with the Elbow Method",
     pch=20, cex=2)
```



With the Elbow method, the y-axis(within groups sum of square) will tend to decrease substantially with each successive increase in the number of clusters. Simplistically, an optimal number of clusters is identified once a "kink" in the line plot is observed. In our example, we can say with confidence that the optimal number of clusters to be used is 6 because after 6 clusters the observed difference in the within-cluster dissimilarity is not substantial.

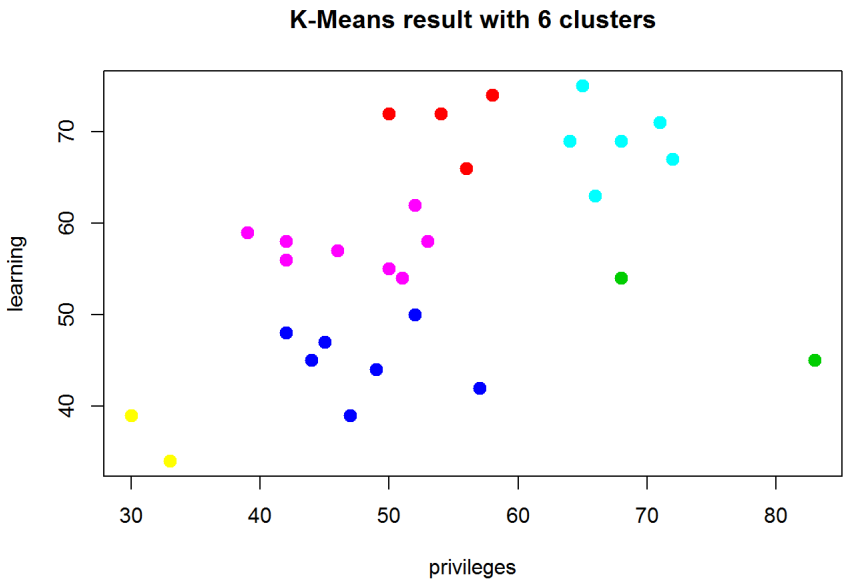
K-means Clustering When k = 6

Assuming this assertion is valid, we can now apply the identified number of clusters onto the K-Means algorithm and plot the results:

```r
# Perform K-Means with the optimal number of clusters identified from the Elbow method
set.seed(7)
km2 = kmeans(newdata, 6, nstart = 100)

# Examine the result of the clustering algorithm
km2
```

```
## K-means clustering with 6 clusters of sizes 4, 2, 8, 6, 8, 2
##
## Cluster means:
##   privileges learning
## 1   54.50000   71.000
## 2   75.50000   49.500
## 3   47.62500   45.250
## 4   67.66667   69.000
## 5   46.87500   57.375
## 6   31.50000   36.500
##
## Clustering vector:
##  [1] 6 5 4 3 1 3 5 5 4 3 5 3 3 2 1 1 4 4 5 2 6 5 3 5 3 4 1 3 4 5
##
## Within cluster sum of squares by cluster:
## [1]  71.0000 153.0000 255.3750 133.3333 244.7500  17.0000
##  (between_SS / total_SS =  89.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
# Plot results
plot(newdata, col = (km2$cluster + 1), main = "K-Means result with 6 clusters", pch = 20, cex = 2)
```



As shown above, there is a relatively well defined set of groups of departments that are relatively distinct when it comes to the attitude around “privileges” and “learning” in the survey. The next steps in further analyzing this set of data is to devise strategies to understand why certain departments rate these two different measures like the results and what to do about it.

Real Life Example:

1. Ever wondered how professors set the curve? Through k-means squaring they can choose how many tiers they want the grades to be divided into and decide which students get to be in which grade brackets.
2. Clustering is the backbone behind the search engines. It can show people the most related information including images, articles, videos, etc.
3. Uses in business:

- Behavioral segmentation:
  - segment by purchase history
  - segment by activities on application, website, or platform
  - define personas based on interests
  - create profiles based on activity monitoring
- Inventory Categorization:
  - group inventory by sales activity
  - group inventory by manufacturing metrics

- Sorting sensor measurements:
    - detect activity types in motion sensors
    - group images
    - separate audio
    - identify groups in health monitoring
- Detecting bots or anomalies:
    - separate valid activity groups from bots
    - group valid activity to clean up outlier detection

Take-home message

After writing up this post, I hope it is more clear what k-means clustering is, and the amazing things that it can do. K-means clustering is really helpful in grouping a dataset into different segments, and it is also very useful in analyzing the well-being, anomalies, or performing patterns of a business.

Reference

1. https://www.quora.com/How-can-we-choose-a-good-K-for-K-means-clustering
2. https://www.datascience.com/blog/k-means-clustering
3. https://en.wikipedia.org/wiki/K-means_clustering
4. https://www.r-bloggers.com/k-means-clustering-in-r/
5. https://rpubs.com/FelipeRego/K-Means-Clustering
6. https://discuss.analyticsvidhya.com/t/what-is-within-cluster-sum-of-squares-by-cluster-in-k-means/2706/2
7. https://www.quora.com/What-is-the-use-of-k-means-clustering-Where-is-it-applied