Introduction to Data Science

DSA1101

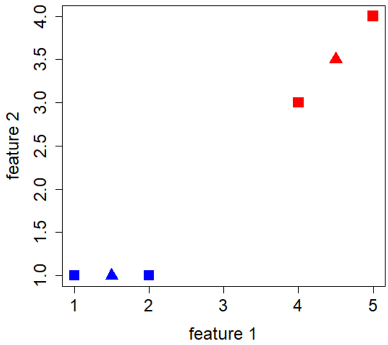
Semester 1, 2018/2019
Week 3

Clustering Methods

Determining the Number of Clusters

- Recall that the selection of the number of clusters k can be guided by the metric *Within Sum of Squares* (WSS).
- To ground ideas, let us calculate the WSS for our earlier example involving four data points and two clusters.

Determining the Number of Clusters



- We have determined that at convergence, there are two clusters:
- blue centroid: $(1.5, 1)$
- red centroid: $(4.5, 3.5)$

Determining the Number of Clusters

- We first calculate the sum of squared distances from each of the two points in the blue cluster, $(1, 1)$ and $(2, 1)$, to the blue centroid

$$
\begin{aligned}
SS_{blue} &= \left(\sqrt{(1-1.5)^2+(1-1)^2}\right)^2 + \\
&\quad \left(\sqrt{(2-1.5)^2+(1-1)^2}\right)^2 \\
&= \left(\sqrt{(-0.5)^2}\right)^2 + \left(\sqrt{(0.5)^2}\right)^2 \\
&= (-0.5)^2 + (0.5)^2 = 0.25 + 0.25 \\
&= 0.5
\end{aligned}
$$

Determining the Number of Clusters

- Then calculate the sum of squared distances from each of the two points in the red cluster, $(4, 3)$ and $(5, 4)$, to the red centroid

$$
\begin{aligned}
SS_{red} &= \left(\sqrt{(4-4.5)^2 + (3-3.5)^2}\right)^2 + \\
&\left(\sqrt{(5-4.5)^2 + (4-3.5)^2}\right)^2 \\
&= \left(\sqrt{(-0.5)^2 + (-0.5)^2}\right)^2 + \left(\sqrt{(0.5)^2 + (0.5)^2}\right)^2 \\
&= (-0.5)^2 + (-0.5)^2 + (0.5)^2 + (0.5)^2 \\
&= 0.25 + 0.25 + 0.25 + 0.25 \\
&= 1.0
\end{aligned}
$$

Determining the Number of Clusters

- The *Within Sum of Squares* (WSS) for our example when $k = 2$ will be

$$WSS = SS_{blue} + SS_{red} = 0.5 + 1.0 = 1.5$$

Determining the Number of Clusters

- In general, for M data points $z_1, z_2, \dots, z_M$ with p features, the *Within Sum of Squares* (WSS) is calculated via

$$
\begin{aligned}
WSS &= \sum_{i=1}^{M} dist(z_i, D_i)^2 \\
&= \sum_{i=1}^{M} \sum_{j=1}^{p} \left(x(j)_i - x(j)_{D_i} \right)^2 ,
\end{aligned}
$$

where D_i is the centroid of the cluster to which the i^{th} data point z_i belongs.

Determining the Number of Clusters

- So for our example with 4 data points, the formula for WSS is

$$\begin{aligned} WSS = {} & dist(z_1, D_{blue})^2 + dist(z_2, D_{blue})^2 \\ & + dist(z_3, D_{red})^2 + dist(z_4, D_{red})^2 \end{aligned}$$

  which we have shown to be equal to 1.5

K-means clustering with `kmeans()`

- The *R* function `kmeans()` perform k-means clustering
- We will illustrate the output from `kmeans()` using our simple example
- Recall that our four data points are

  (1 ,1)

  (2, 1)

  (4, 3)

  (5, 4)

K-means clustering with `kmeans()`

- Perform k-means clustering on our four data points with $k = 2$:

```
x=c(1,2,4,5)
y=c(1,1,3,4)

kout=kmeans(cbind(x,y),center=2)
```

column binding

K-means clustering with `kmeans()`

- Perform k-means clustering on our four data points with $k = 2$:

```
> kout
K-means clustering with 2 clusters of sizes 2, 2

Cluster means:
    x   y
1 1.5 1.0
2 4.5 3.5

Clustering vector:the cluster each data point belongs to
[1] 1 1 2 2

Within cluster sum of squares by cluster:
[1] 0.5 1.0
```

K-means clustering with `kmeans()`

- The available output components from `kmeans()`:

```
Available components:

[1] "cluster"       "centers"
[3] "totss"         "withinss"
[5] "tot.withinss"  "betweenss"
[7] "size"          "iter"
[9] "ifault"
```

- The WSS can be computed by `sum(kout$withinss)` or equivalently given by `kout$tot.withinss`:

```
> sum(kout$withinss)
[1] 1.5
> kout$tot.withinss
[1] 1.5
```

K-means clustering with kmeans()

- The WSS can be computed by sum(kout$withinss) or equivalently given by kout$tot.withinss:

```
> sum(kout$withinss)
[1] 1.5
> kout$tot.withinss
[1] 1.5
```

Example:academic performance of high school students

- The task is to group 620 high school seniors based on their grades in three subject areas: English, mathematics, and science.
- The grades are averaged over their high school career and assume values from 0 to 100.
- Available as the CSV file `grades_km_input.csv` on the course website.

Example: *k*-means clustering using *R*

```
> grade_input = read.csv("grades_km_input.csv")
> head(grade_input)
  Student English Math Science
1       1      99   96      97
2       2      99   96      97
3       3      98   97      97
4       4      95  100      95
5       5      95   96      96
6       6      96   97      96
```

- We can use *R* to calculate WSS for each value of k, the number of clusters.

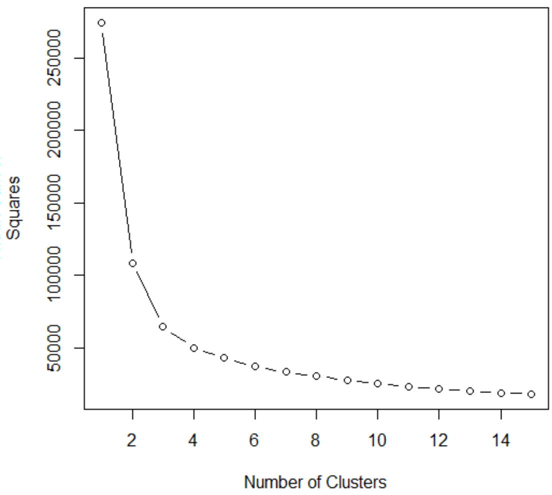
```
wss <- numeric(15) construct a vector of length 15

for (k in 1:15) {

  wss[k] <- sum(kmeans(grade_input[,c("English","
      Math","Science")],
  centers=k, nstart=25)$withinss)
   do 25 times of K-means Clustering to arrive at the best answer
}
```

Determining the Number of Clusters using *R*

- We can plot the WSS against the number of clusters, k.

```r
plot(1:15, wss, type="b",
xlab="Number of Clusters",
ylab="Within Sum of Squares")
```

Determining the Number of Clusters using *R*



Determining the Number of Clusters using R

- WSS is greatly reduced when k increases from one to two. Another substantial reduction in WSS occurs at $k = 3$.
- However, the improvement in WSS is fairly linear for $k > 3$.
- Therefore, the k-means analysis will be conducted for $k = 3$.
- The process of identifying the appropriate value of k is referred to as finding the “elbow” of the WSS curve

Example:academic performance of high school students

- We proceed with clustering the high school students with $k = 3$.

```
> km = kmeans(grade_input[,c("English","Math","Science")],3, nstart=25)
> km
K-means clustering with 3 clusters of sizes 158, 244, 218

Cluster means:
   English     Math  Science
1 97.21519 93.37342 94.86076
2 85.84426 79.68033 81.50820
3 73.22018 64.62844 65.84862
```

Example:academic performance of high school students

- Visualization is vital for understanding data analytic results
- We will use the ggplot2 package to visualize the identified student clusters and centroids
- In the 'Packages' option within R console, select 'Install package(s)' then select ggplot2
- Load ggplot2 with the command

```
library(ggplot2)
```

Example:academic performance of high school students

- The following code is adapted from *Data Science & Big Data Analytics*, pages 126-127.

```r
#prepare the student data and clustering results
    for plotting
df = as.data.frame(grade_input[,2:4])
df$cluster = factor(km$cluster)
centers=as.data.frame(km$centers)

g1= ggplot(data=df, aes(x=English, y=Math, color=
    cluster )) +
geom_point() + theme(legend.position="right") +
geom_point(data=centers,
aes(x=English,y=Math, color=as.factor(c(1,2,3))),
size=10, alpha=.3, show_guide=FALSE)
```

Example:academic performance of high school students

- The following code is adapted from *Data Science & Big Data Analytics*, pages 126-127.

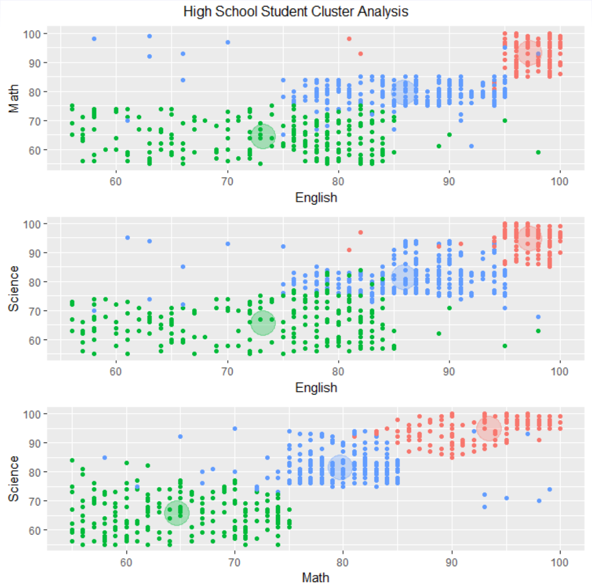
```
g2 =ggplot(data=df, aes(x=English, y=Science, color=cluster )) +
geom_point() +
geom_point(data=centers,
aes(x=English,y=Science, color=as.factor(c(1,2,3)))),
size=10, alpha=.3, show_guide=FALSE)


g3 = ggplot(data=df, aes(x=Math, y=Science, color=cluster )) +
geom_point() +
geom_point(data=centers,
aes(x=Math,y=Science, color=as.factor(c(1,2,3))),
size=10, alpha=.3, show_guide=FALSE)
tmp = ggplot_gtable(ggplot_build(g1))
```

Example:academic performance of high school students

- The following code is adapted from *Data Science & Big Data Analytics*, pages 126-127.

```

library(gridExtra)
library(grid)

grid.arrange(arrangeGrob(g1 + theme(legend.
    position="none"),
g2 + theme(legend.position="none"),
g3 + theme(legend.position="none"),
top ="High School Student Cluster Analysis",
ncol=1))
```

Example:academic performance of high school students

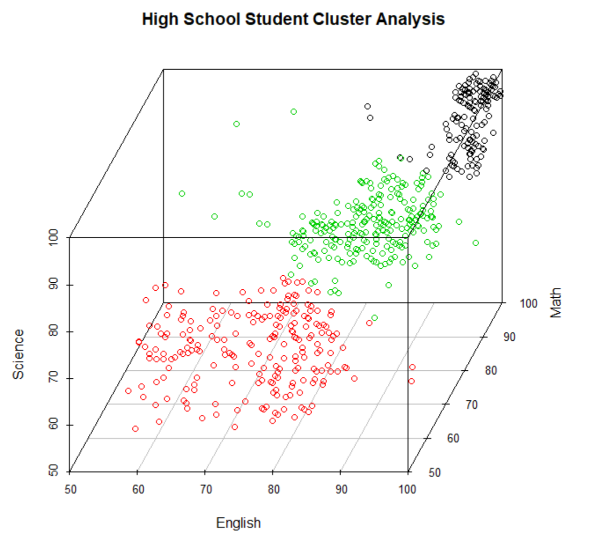


Example:academic performance of high school students

- We can also create a 3D scatter-plot for our cluster analysis

```
install.packages("scatterplot3d")
library(scatterplot3d)


scatterplot3d(df, main="High School Student
    Cluster Analysis",
                  angle=65,color=df$cluster)
```

Example:academic performance of high school students



Example:academic performance of high school students

- Interactive spinning 3D scatter-plot

```

install.packages("rgl")
library(rgl)

plot3d(df, col=df$cluster, size=3)
```

K-means clustering

- Assigning labels to the identified clusters is useful to communicate the results of an analysis.
- In a marketing context, it is common to label a group of customers as frequent shoppers or big spenders.
- Such designations are especially useful when communicating the clustering results to business users or executives.
- It is better to describe the marketing plan for big spenders rather than Cluster #1.

K-means clustering: feature selection

- Although k-means is considered an unsupervised method, there are still several decisions that the practitioner must make:

(i) Which features should be included in the analysis?

(ii) What unit of measure (for example, miles or kilometers) should be used for each feature?

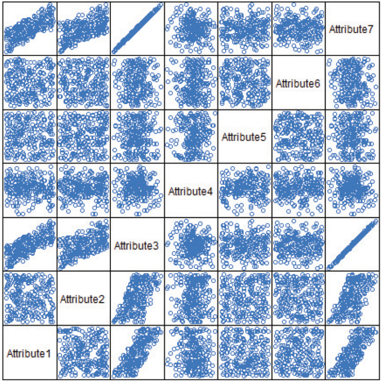
(iii) Do the features need to be rescaled so that one feature does not have a disproportionate effect on the results?

(iv) What other considerations might apply?

Which features should be included in the analysis?

- The Data Scientist may have a choice of a dozen or more attributes to use in the clustering analysis.
- Whenever possible and based on the data, it is best to reduce the number of attributes to the extent possible.
- Too many attributes can minimize the impact of the most important variables.
- The use of several similar attributes can place too much importance on one type of attribute.
- For example, if five attributes related to personal wealth are included in a clustering analysis, the wealth attributes dominate the analysis and possibly mask the importance of other attributes, such as age.

Which features should be included in the analysis?



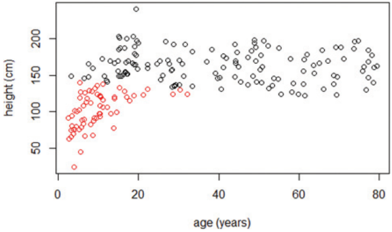
- When dealing with the problem of too many attributes, one useful approach is to identify any highly correlated attributes and use only one or two of the correlated attributes in the clustering analysis.
- The strongest relationship is observed to be between Attribute 3 and Attribute 7.

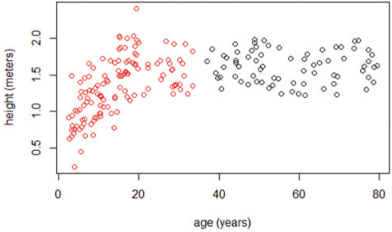
-> can only use either Attribute 3 or 7.

Units of measure

- From a computational perspective, the k-means algorithm is somewhat indifferent to the units of measure for a given attribute (for example, meters or centimeters for a patient's height).
- However, the algorithm will identify different clusters depending on the choice of the units of measure. similar attributes can place too much importance on one type of attribute.
- For example, suppose that k-means is used to cluster patients based on age in years and height in centimeters.

Units of measure





- When the height is expressed in meters, the magnitude of the ages dominates the distance calculation between two points.
- The height attribute provides only as much as the square between the difference of the maximum height and the minimum height or $(2.0 - 0)^2 = 4$ to the sum in the square root of the distance formula

$$dist = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

but age: (80-40)^2 = 1600 >> 4
age difference contributes more to the distance formula