Introduction to Data Science

DSA1101

Semester 1, 2018/2019
Week 2

Clustering Methods

- The k-means clustering algorithm can be generalized to cluster objects with more than two features.
- To generalize the algorithm, suppose there are M objects, where each object is described by p attributes or property values $(x(1), x(2), ..., x(p))$.
- Then the i^{th} object is described by $(x(1)_i, x(2)_i, ..., x(p)_i)$ for $i = 1, 2, ..., M$.

- For a given point z_i at $(x(1)_i, x(2)_i, ..., x(p)_i)$ and a centroid D at $(x(1)_d, x(2)_d, ..., x(p)_d)$, the distance between z_i and D is

$$dist(z_i, D) = \sqrt{\sum_{j=1}^{p} (x(j)_i - x(j)_d)^2}$$

$i = 1, 2, ..., M.$

- The centroid for a cluster of m points, $(x(1)_i, x(2)_i, \ldots, x(p)_i)$ for $i = 1, 2, \ldots, m$, is given by

$$\left(\frac{1}{m}\sum_{i=1}^{m} x(1)_i, \frac{1}{m}\sum_{i=1}^{m} x(2)_i, \ldots, \frac{1}{m}\sum_{i=1}^{m} x(p)_i\right)$$

Example: k-means clustering with more than two features

- Suppose we have $p = 3$ features for clustering 4 objects:
- $[x(1)_1, x(2)_1, x(3)_1] = (1, 1, 3)$
- $[x(1)_2, x(2)_2, x(3)_2] = (2, 1, 5)$
- $[x(1)_3, x(2)_3, x(3)_3] = (4, 3, 2)$
- $[x(1)_4, x(2)_4, x(3)_4] = (5, 4, 9)$

Example: k-means clustering with more than two features

- If we assign the centroid for the first cluster D_1 to be at

$$[x(1)_{D_1}, x(2)_{D_1}, x(3)_{D_1}] = (2, 2, 2),$$

  then the distance from the first object to this centroid will be

$$\sqrt{(1-2)^2 + (1-2)^2 + (3-2)^2} = \sqrt{3}.$$

- Similarly for the other three objects.

Example: k-means clustering with more than two features

- To calculate the centroid for the four objects, we use the formula

$$\left(\frac{1}{4}\sum_{i=1}^{4} x(1)_i, \frac{1}{4}\sum_{i=1}^{4} x(2)_i, \frac{1}{4}\sum_{i=1}^{4} x(3)_i\right)$$

Example: k-means clustering with more than two features

$$\left(\frac{1}{4}\sum_{i=1}^{4} x(1)_i, \frac{1}{4}\sum_{i=1}^{4} x(2)_i, \frac{1}{4}\sum_{i=1}^{4} x(3)_i\right)$$
$$= \left(\frac{1+2+4+5}{4}, \frac{1+1+3+4}{4}, \frac{3+5+2+9}{4}\right)$$
$$= (3, 2.25, 4.75)$$

Example: k-means clustering using R

- 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
```

Example: k-means clustering using R

- The output from the R function `kmeans` includes

(i) The location of the cluster means

(ii) A clustering vector that defines the membership of each student to a corresponding cluster

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

```
> kout <- kmeans(grade_input[,c("English","Math","Science")],centers=3)

> kout
K-means clustering with 3 clusters of sizes 244, 158, 218

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

Clustering vector:
  [1] 2 2 2 2 2 2 2 2 2 2 2
```

2 1 3

--> second cluster, first cluster, third cluster

*Final clusters may be different depending on the starting centroids.

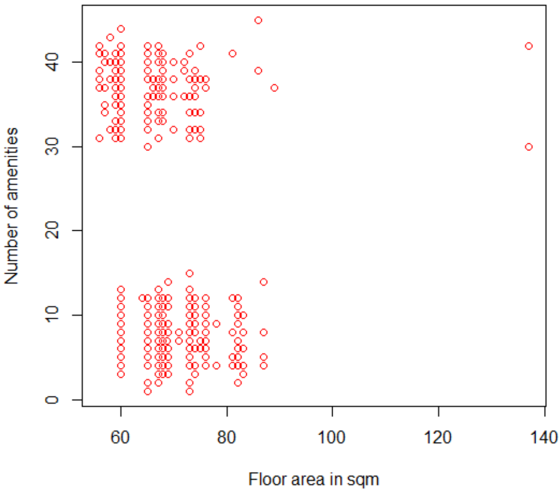
Example: k-means clustering using R

- Recall that in the HDB resale data example, we only have two features for clustering

(1) Floor area in square meters

(2) The number of amenities in the vicinity of the HDB unit

Example: k-means clustering using R



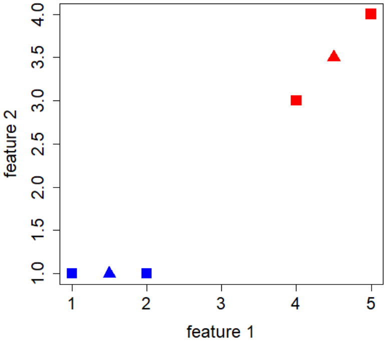
Example: k-means clustering using R

- It was relatively easy to see the data in two dimensions and determine that there are two clusters
- Is there a way to determine the number of clusters, k, when the number of features for clustering is of higher dimensions?

Determining the Number of Clusters

- 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

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

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

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

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 $z_1, z_2, ..., z_M$ with p features, the *Within Sum of Squares* (WSS) is calculated via

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

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

$$WSS = dist(z_1, D_{blue})^2 + dist(z_2, D_{blue})^2 + dist(z_3, D_{red})^2 + dist(z_4, D_{red})^2$$

which we have shown to be equal to 1.5

Determining the Number of Clusters using *R*

- The *R* function `kmeans` also returns the vector of values, `withinss`, which is the sum of squared distances within each cluster.
- For example, we earlier performed k-means clustering on student grade data with a cluster size of $k = 3$.

```
> kout <- kmeans(grade_input[,c("English","Math","
    Science")],centers=3)
> kout$withinss
[1] 34806.339  6692.589 22984.131
```

- We just have to sum up the entries in `withinss` to get the WSS.

Determining the Number of Clusters using *R*

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

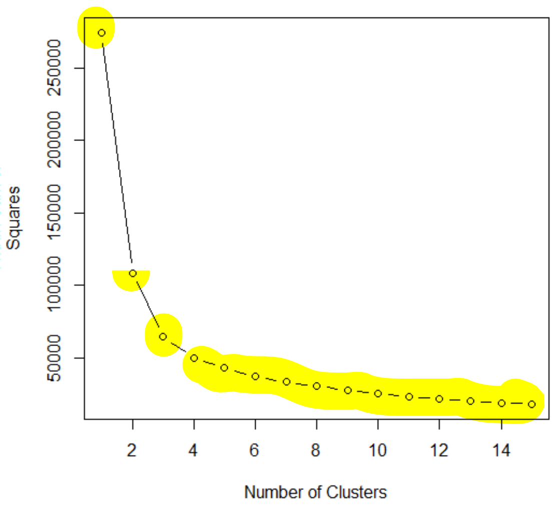
```
> wss <- numeric(15)
>
> for (k in 1:15) {
+
+   wss[k] <- sum(kmeans(grade_input[,c("English","Math","Science")],
+   centers=k, nstart=25)$withinss)
+
+ }
```

Determining the Number of Clusters using *R*

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

```
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 lower -> better clustering

- 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