Unsupervised learning
Part I: Clustering

Jing Wang

Outline

- Introduction of Unsupervised Learning
- Introduction of Clustering
  - What is clustering
  - Applications
- K-means Algorithm
  - What is k-means
  - How does k-means work
  - Loss function
  - Strengths and weakness

Introduction of unsupervised learning

Supervised learning vs.Unsupervised learning

Supervised learning: discover patterns in the data that relate data attributes with a target (class) attribute.

- ➢ These patterns are then utilized to predict the values of the target attribute in future data instances.

Examples: Regression, Classification

Unsupervised learning: The data have no target attribute.

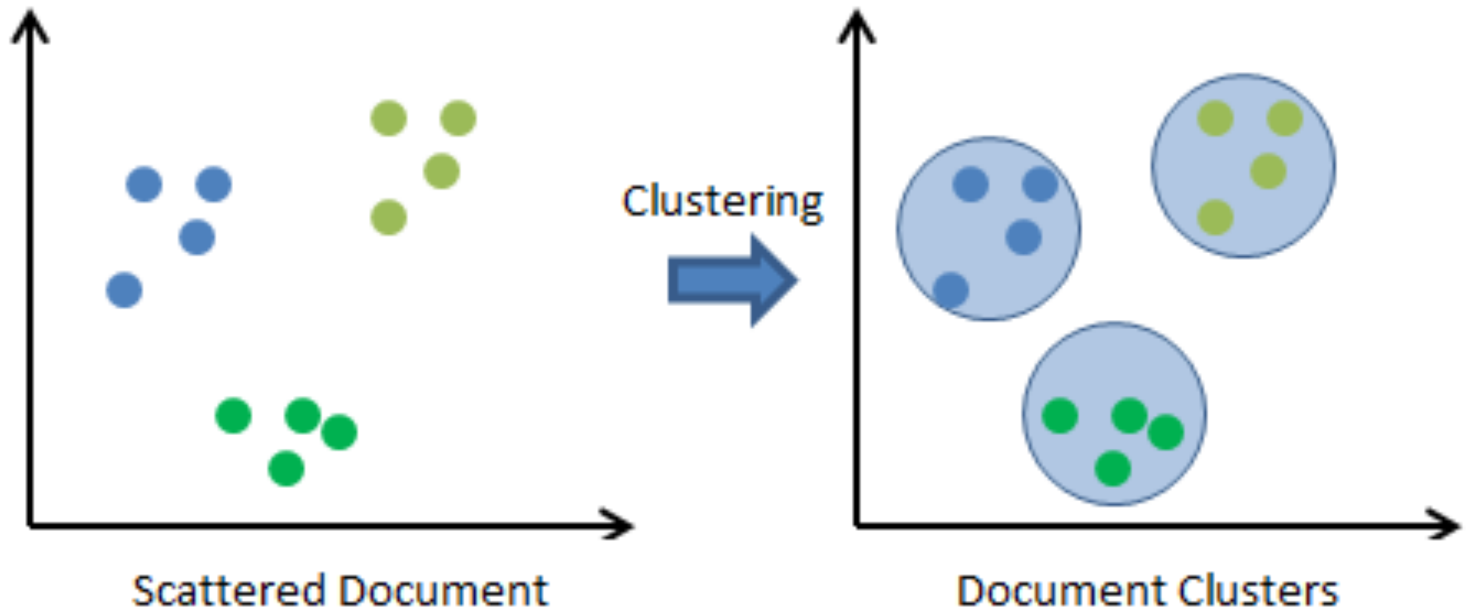
- ➢ We want to explore the data to find some intrinsic structures in them.

Examples: Clustering, Dimensionality reduction

Clustering

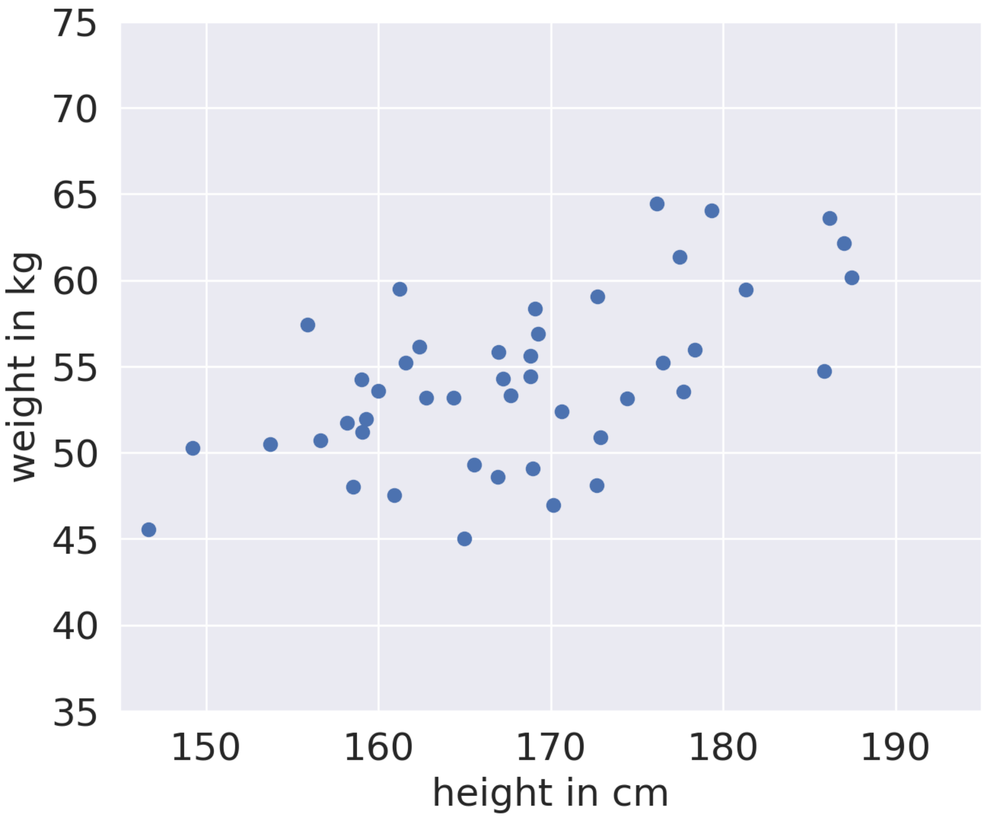
What is clustering?

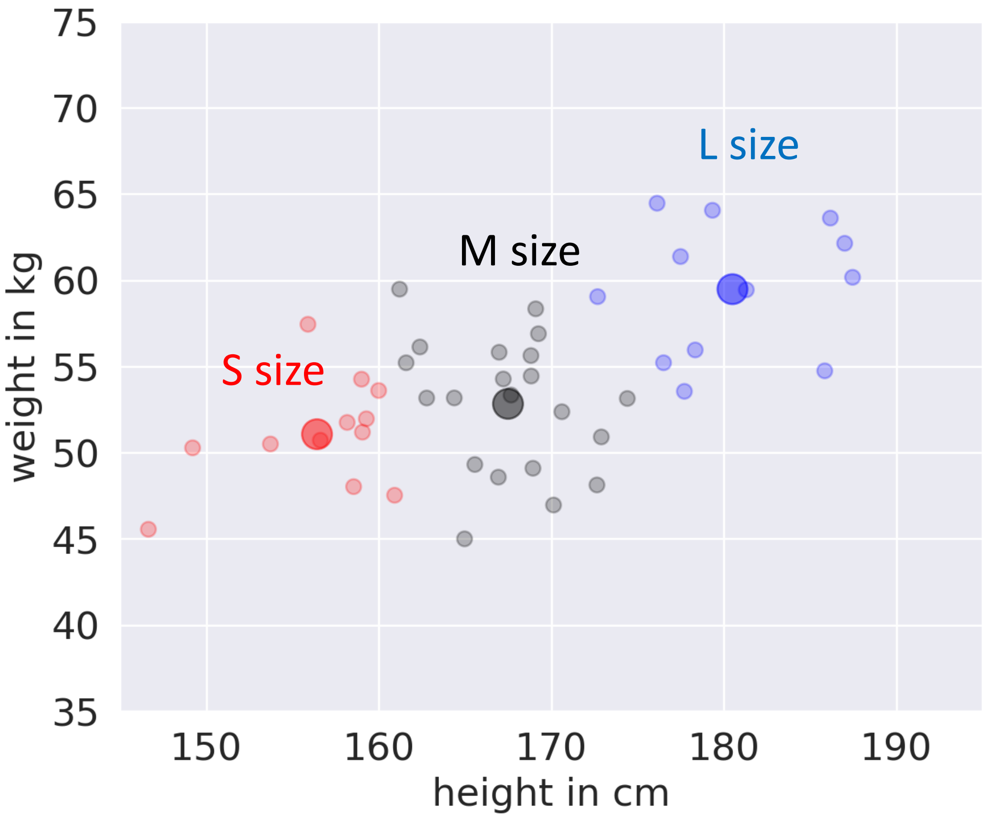
Clustering is to group a set of objects such that similar objects are in the same group and dissimilar objects are separated into different groups



Applications

-Customer categorisation





Applications

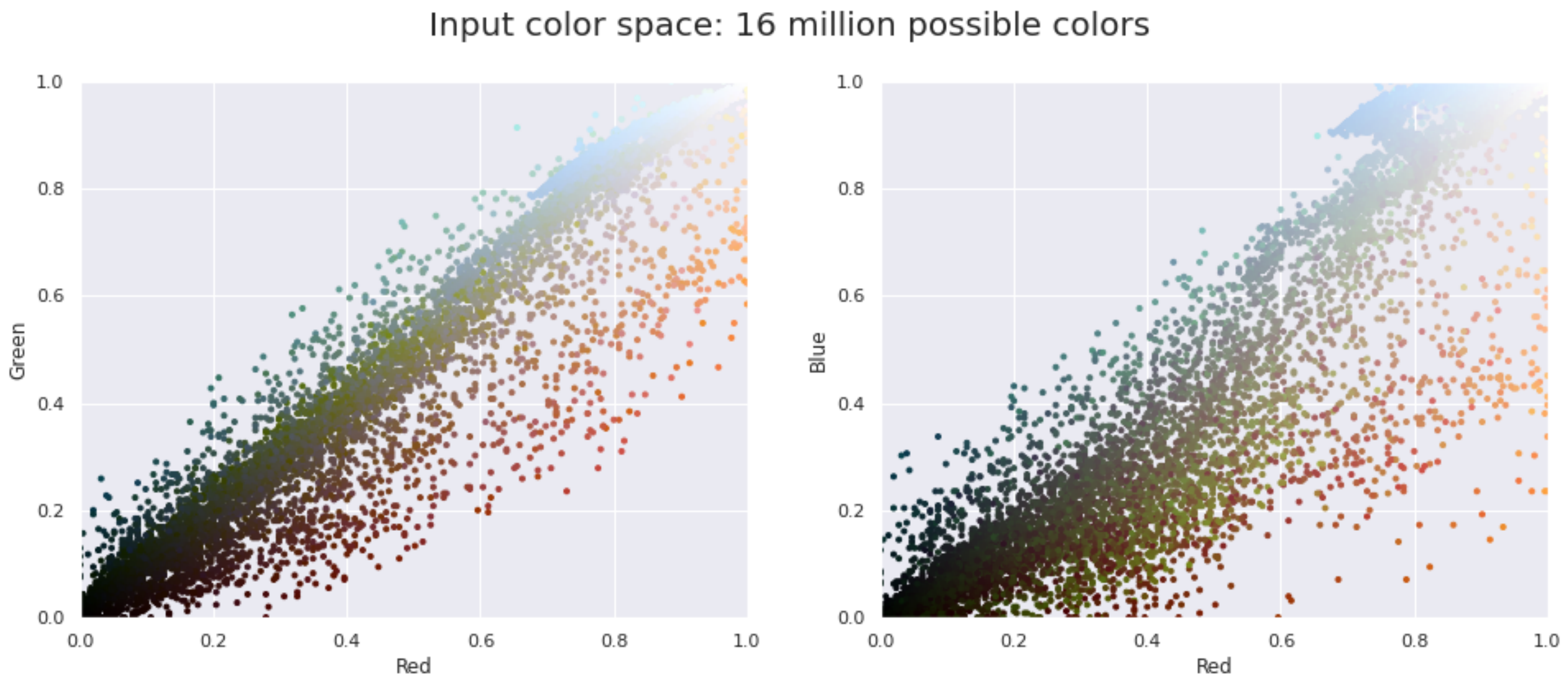
-Image compression

Number of possible colours: $256^3 = 16777216$

From scikit-learn (https://scikit-learn.org/stable/index.html) sample images

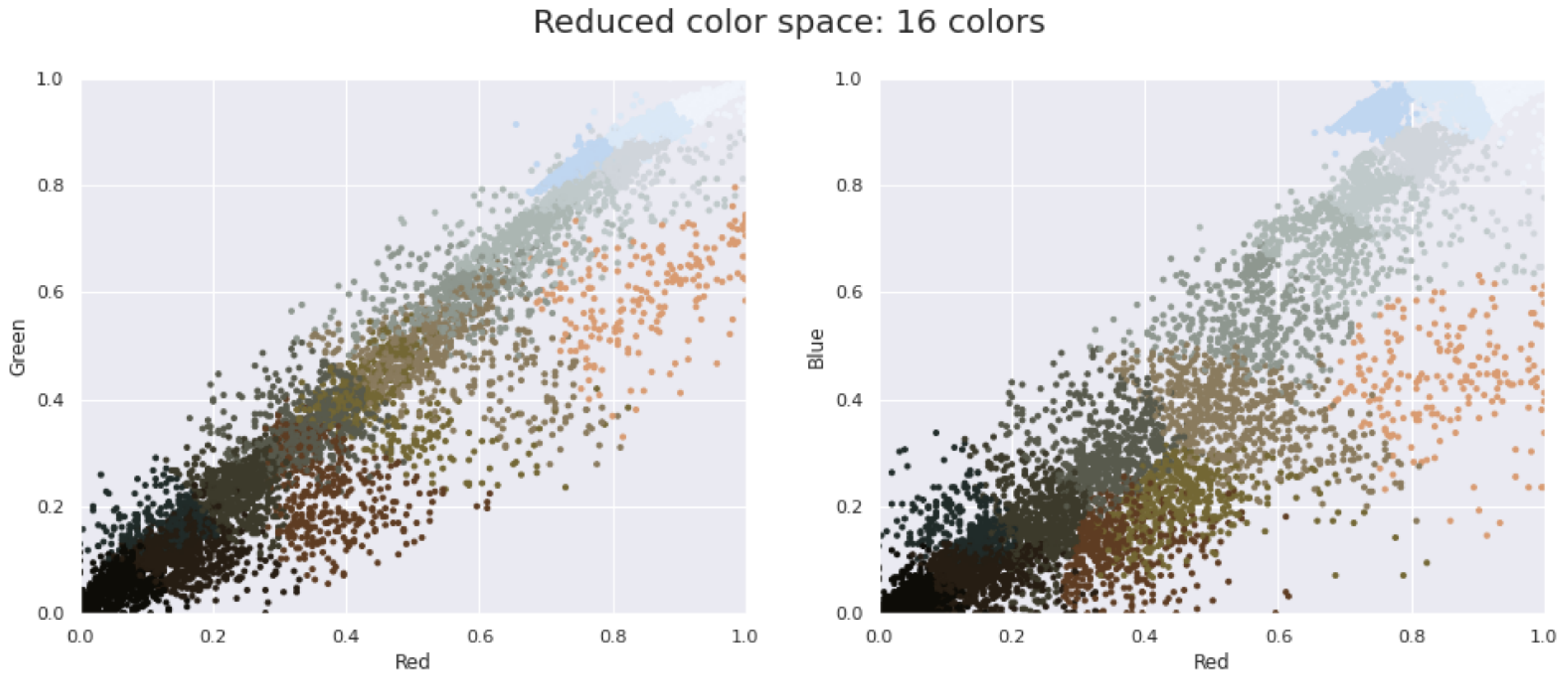
Applications

-Image compression



Applications

-Image compression

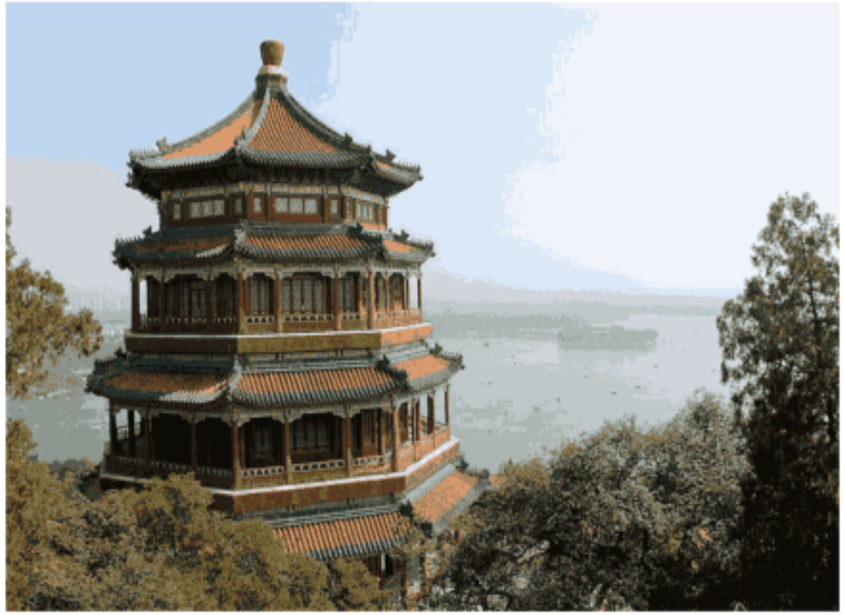


Applications

-Image compression

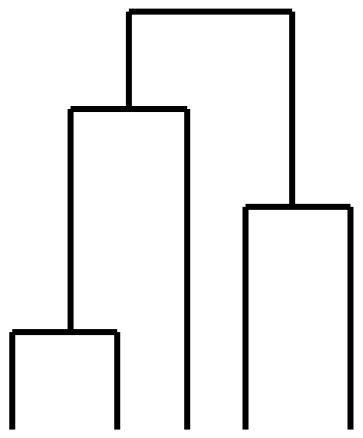
Original Image

16-color Image

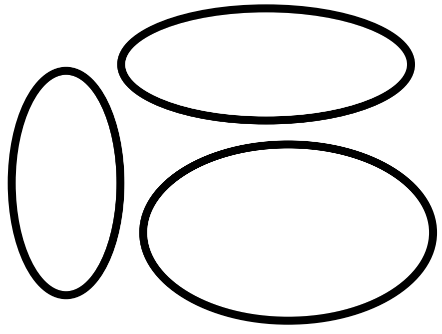


Types of clustering

1. **Hierarchical algorithms**: find successive clusters using previously established clusters.
   - Agglomerative("bottom-up"): begin with each element as a separate cluster and merge them into successively larger clusters.
   - Divisive("top-down"): begin with the whole set and proceed to divide it into successively smaller clusters.



2. **Partitional clustering:** determine all clusters at once.
   - ***K*-means**
   - Fuzzy *c*-means clustering



K-means

What is k-means?

The k-means algorithm is an algorithm to cluster m objects based on attributes into k clusters, where k < m.

- Each cluster has a cluster center, called centroid.
- The centroid of the cluster is the mean of the value of the data points that belong to the cluster
- *k* is specified by the user

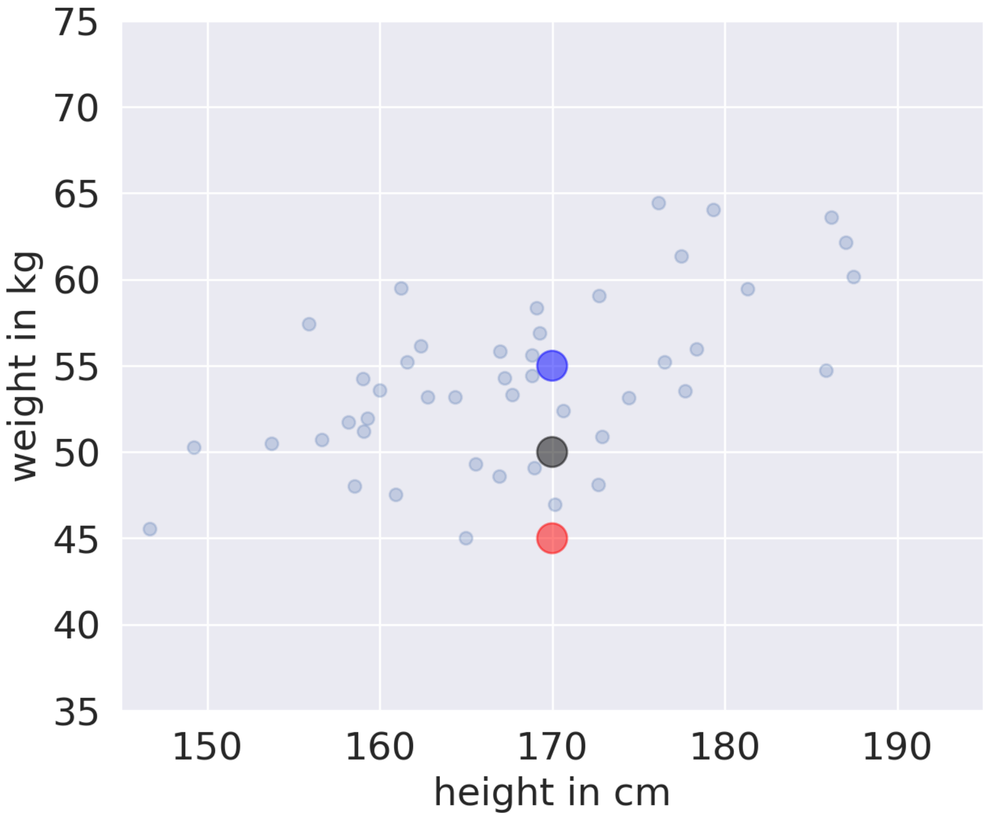
How does K-means work?

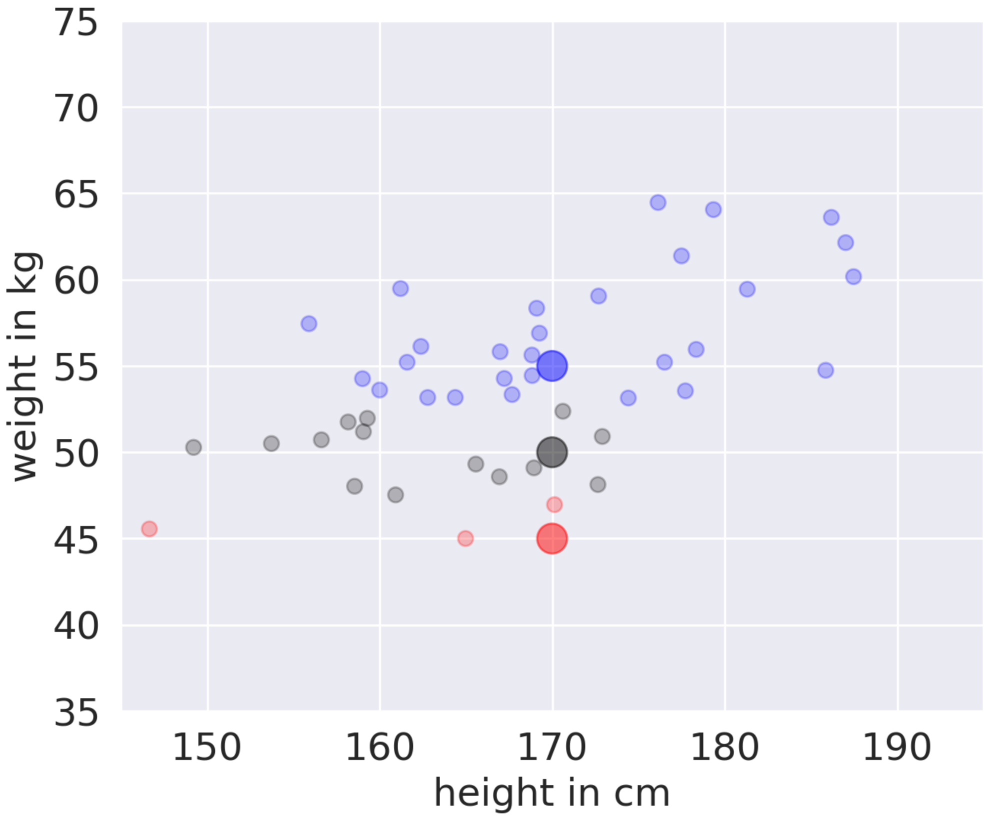
- Step 1: Begin with a decision of the value of k
- Step 2: randomly select initial cluster centroids

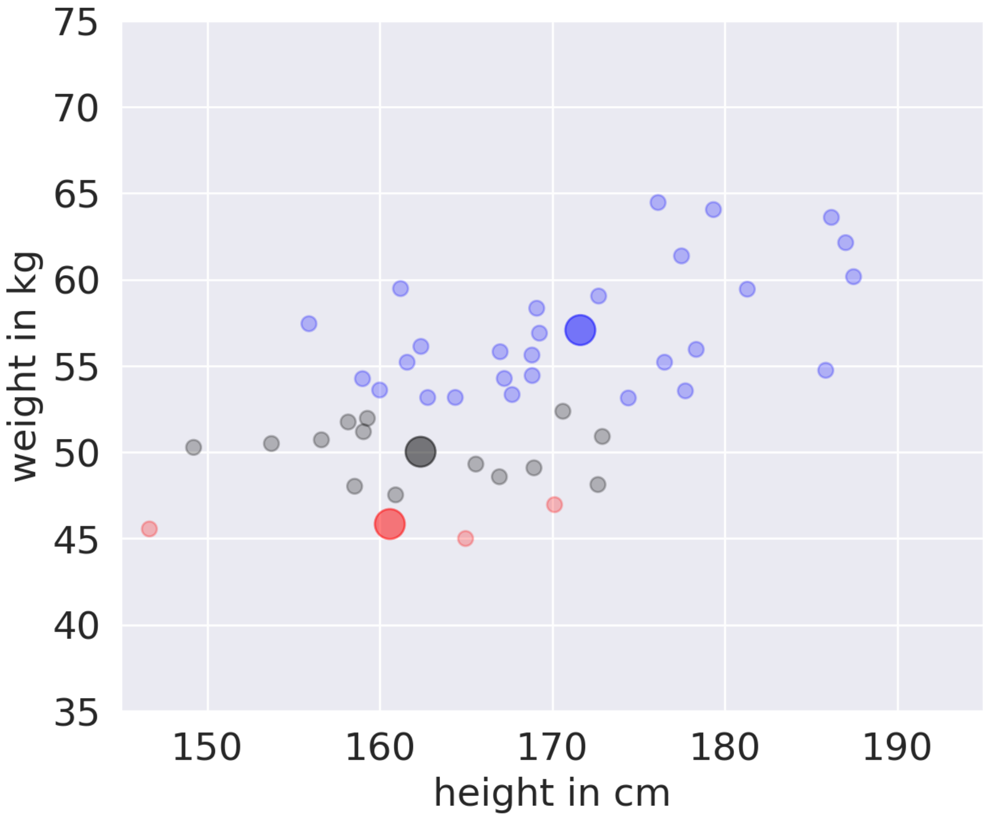
Repeat

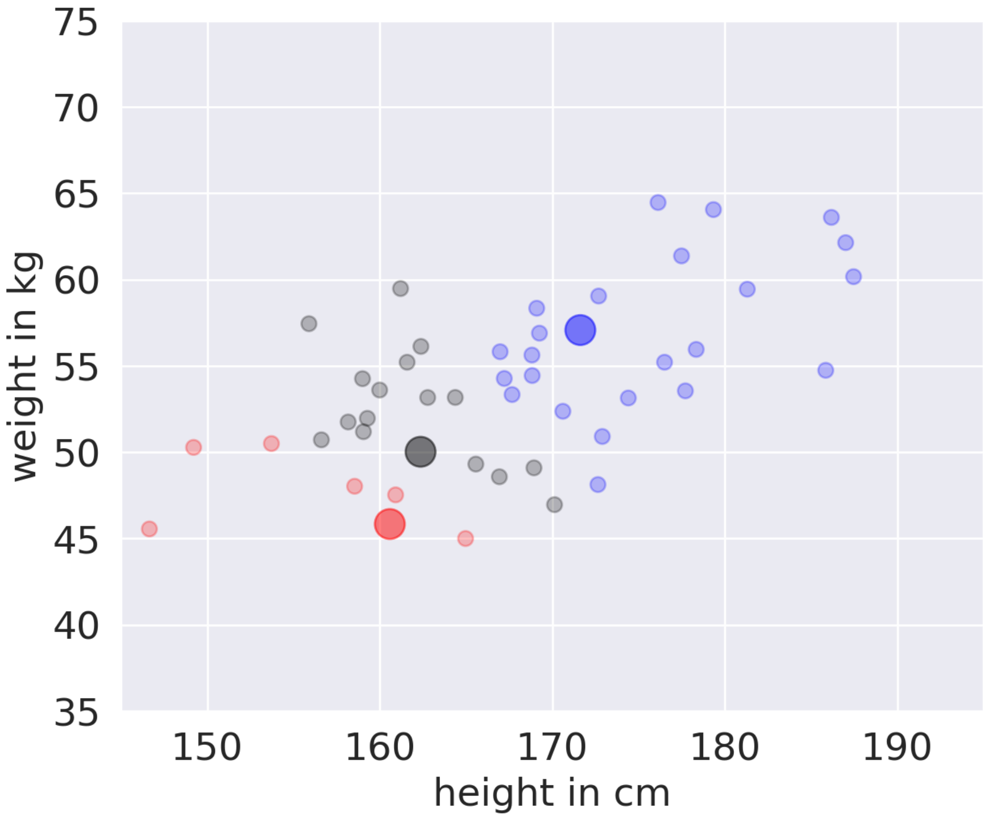
- Step 3: calculate distance from each object to each cluster centroid.
- Step 4: Assign each object to the closest cluster
- Step 5: Compute the new centroid for each cluster

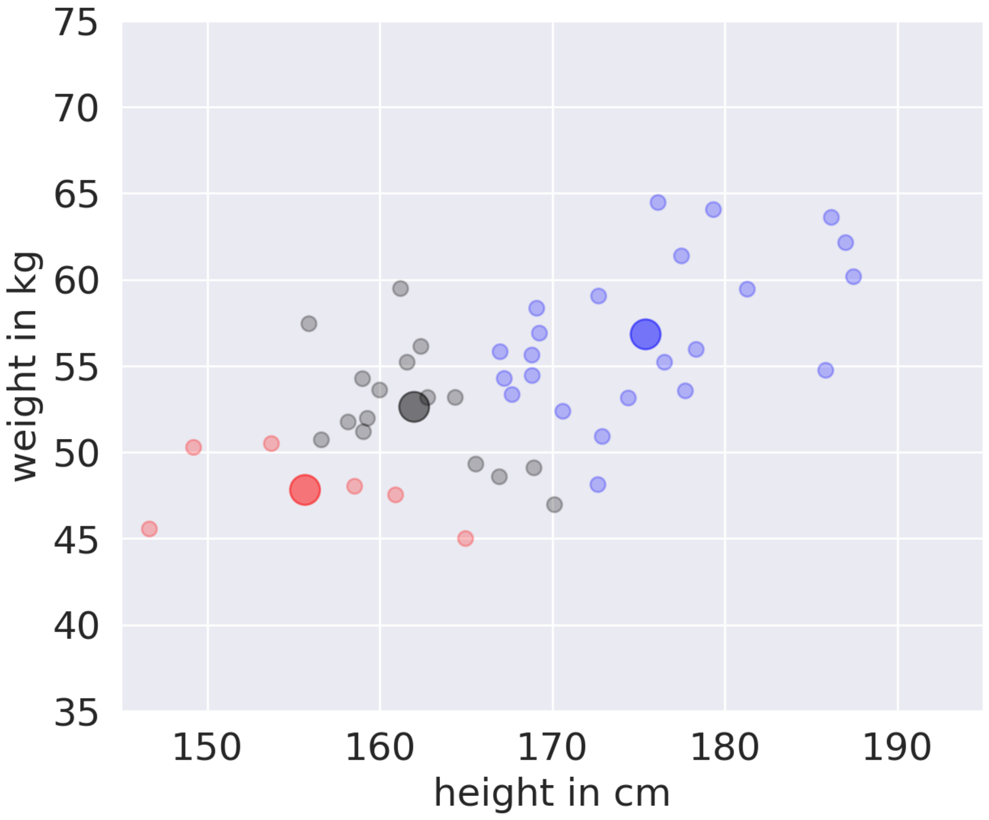
until *no change in clusters*


75
70
65
60
55
50
45
40
35
weight in kg
150
160
170
180
190
height in cm

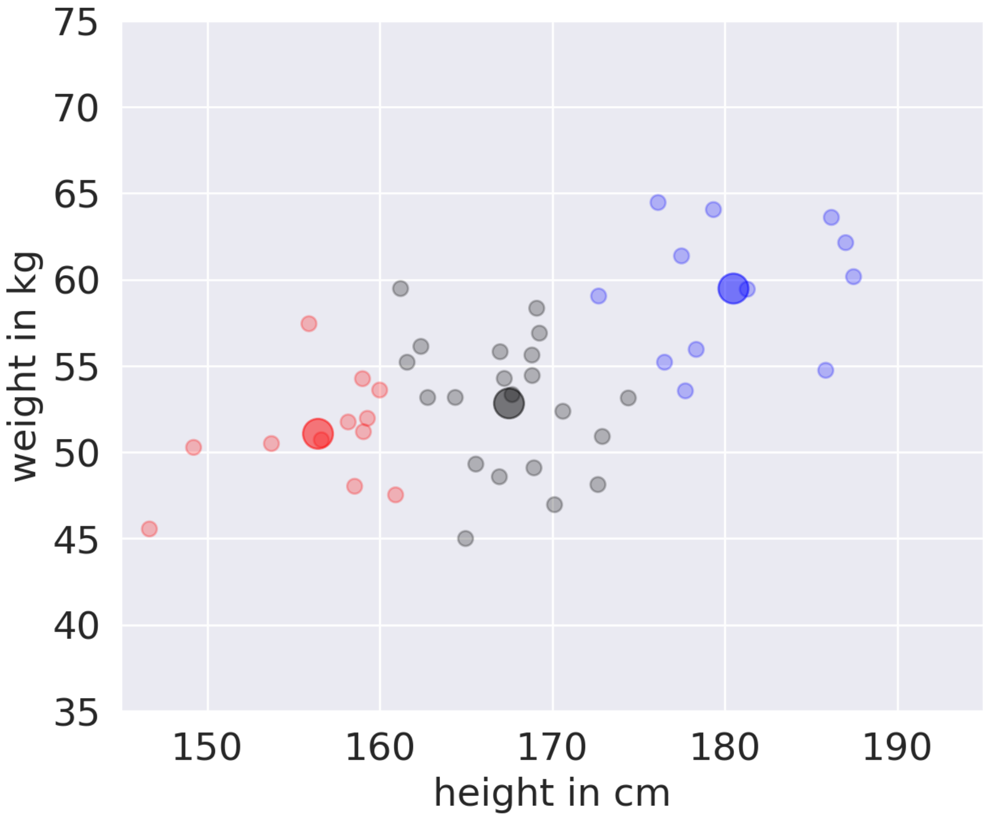

75
70
65
60
55
50
45
40
35
weight in kg
150
160
170
180
190
height in cm


75
70
65
60
55
50
45
40
35
150
160
170
180
190
height in cm
weight in kg


75
70
65
60
55
50
45
40
35
weight in kg
150
160
170
180
190
height in cm



75
70
65
60
55
50
45
40
35
weight in kg
150
160
170
180
190
height in cm



75
70
65
60
55
50
45
40
35
weight in kg
150
160
170
180
190
height in cm

A simple example of implementation of k-means (using k=2)

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Step 1:

Initialization: Randomly we choose following two centroids (k=2) for two clusters.

In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

Step 2:

- Thus, we obtain two clusters containing:

  {1,2,3} and {4,5,6,7}.

- Their new centroids are:

$$m_1 = \left(\frac{1}{3}(1.0+1.5+3.0), \frac{1}{3}(1.0+2.0+4.0)\right) = (1.83, 2.33)$$

$$m_2 = \left(\frac{1}{4}(5.0+3.5+4.5+3.5), \frac{1}{4}(7.0+5.0+5.0+4.5)\right)$$
$$= (4.12, 5.38)$$

Individual	Centroid 1	Centroid 2
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.31	2.06
7	4.30	2.92

$$d(m_1, 2) = \sqrt{|1.0-1.5|^2 + |1.0-2.0|^2} = 1.12$$
$$d(m_2, 2) = \sqrt{|5.0-1.5|^2 + |7.0-2.0|^2} = 6.10$$

Step 3:

- Now using these centroids we compute the Euclidean distance of each object, as shown in table.

- Therefore, the new clusters are:

  {1,2} and {**3**,4,5,6,7}

- Next centroids are: m1=(1.25,1.5) and m2 = (3.9,5.1)

Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

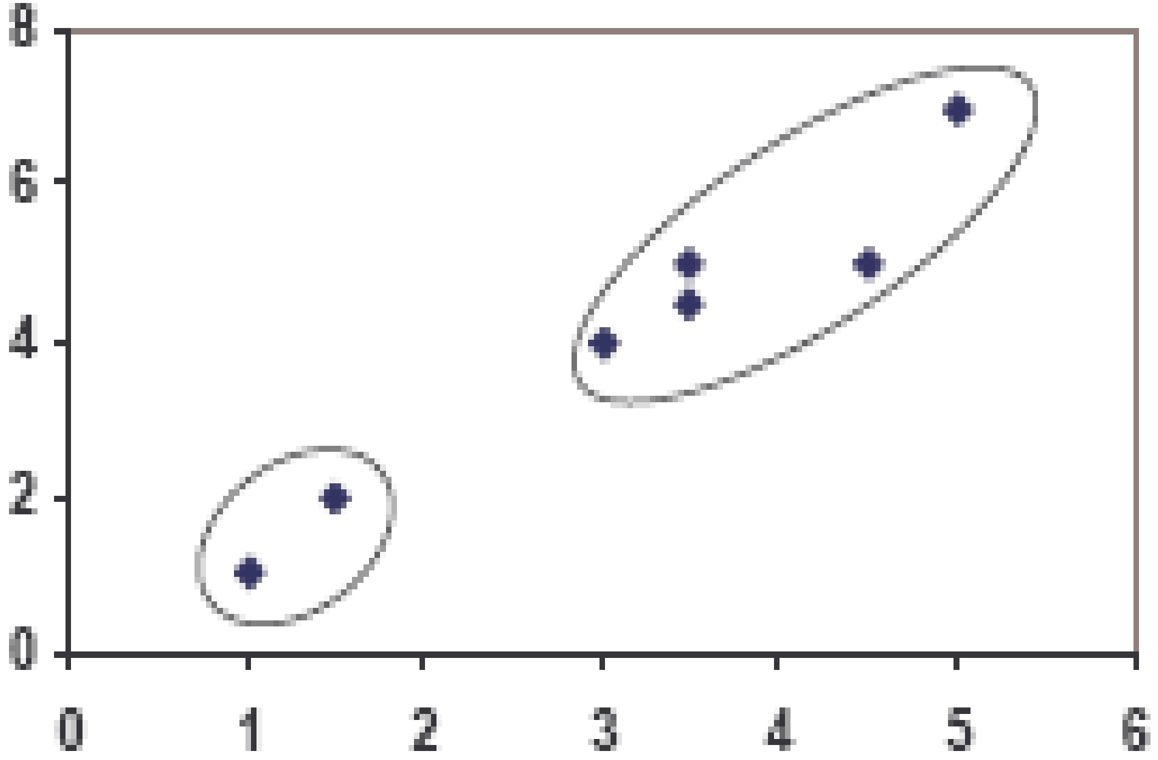
- **Step 4** :

  The clusters obtained are:

  {1,2} and {3,4,5,6,7}

- Therefore, there is no change in the cluster.
- Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and {3,4,5,6,7}.

Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0. 56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72



8
6
4
2
0
0
1
2
3
4
5
6

Loss function

Data (m examples): $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m-1)} \in \mathbb{R}^n$

The centroid of ℓ-th cluster: $\boldsymbol{\mu}^{[\ell]} \in \mathbb{R}^n$ ($\ell = 0,1,\dots,k-1$)

Cluster indicator: $\boldsymbol{z}^{(i)} \in \mathbb{R}^k$ (an one hot vector, $i = 0,1,\dots,m-1$).

$$z_\ell^{(i)} = \begin{cases} 1 & \text{if the } i\text{–th example belongs to the } \ell\text{–th cluster,} \\ 0 & \text{otherwise.} \end{cases}$$

Loss function $J\left(\boldsymbol{\mu}^{[0]}, \boldsymbol{\mu}^{[1]}, \dots, \boldsymbol{\mu}^{[k-1]}, \boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m-1)}\right)$

$$:= \sum_{i=0}^{m-1} \sum_{\ell=0}^{k-1} z_\ell^{(i)} \left\|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{[\ell]}\right\|^2$$

Non-zero only for the cluster $\boldsymbol{x}^{(i)}$ belongs to

Loss function

Data (m examples): $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m-1)} \in \mathbb{R}^n$

The centroid of ℓ-th cluster: $\boldsymbol{\mu}^{[\ell]} \in \mathbb{R}^n$ $(\ell = 0,1,\dots,k-1)$

Cluster label: $\boldsymbol{z}^{(i)} \in \mathbb{R}^k$ (an one hot vector, $i = 0,1,\dots,m-1$).

$$z_\ell^{(i)} = \begin{cases} 1 & \text{if the } i\text{–th example belongs to the } \ell\text{–th cluster,} \\ 0 & \text{otherwise.} \end{cases}$$

Loss function $J\left(\boldsymbol{\mu}^{[0]}, \boldsymbol{\mu}^{[1]}, \dots, \boldsymbol{\mu}^{[k-1]}, \boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m-1)}\right)$

$$:= \sum_{i=0}^{m-1} \sum_{\ell=0}^{k-1} z_\ell^{(i)} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{[\ell]} \right\|^2$$

The square distance between $\boldsymbol{x}^{(i)}$
and the centroid of the cluster $\boldsymbol{x}^{(i)}$ belongs to

Update of cluster indicators:

Minimise $J\left(\boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m-1)}, \boldsymbol{\mu}^{[0]}, \boldsymbol{\mu}^{[1]}, \dots, \boldsymbol{\mu}^{[k-1]}\right)$

$$\coloneqq \sum_{i=0}^{m-1} \sum_{\ell=0}^{k-1} z_{\ell}^{(i)} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{[\ell]} \right\|^2$$

w.r.t. $\boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m-1)}$ with $\boldsymbol{\mu}^{[0]}, \boldsymbol{\mu}^{[1]}, \dots, \boldsymbol{\mu}^{[k-1]}$ fixed

$$z_{\ell}^{(i)} \leftarrow \begin{cases} 1 & \text{if } \ell = \operatorname*{argmin}_{\ell'} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{[\ell']} \right\|, \\ 0 & \text{otherwise.} \end{cases}$$

Update of cluster indicators:

Minimise $J\left(\boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m-1)}, \boldsymbol{\mu}^{[0]}, \boldsymbol{\mu}^{[1]}, \dots, \boldsymbol{\mu}^{[k-1]}\right)$

$$:= \sum_{i=0}^{m-1} \sum_{\ell=0}^{k-1} z_\ell^{(i)} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{[\ell]} \right\|^2$$

w.r.t. $\boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m-1)}$ with $\boldsymbol{\mu}^{[0]}, \boldsymbol{\mu}^{[1]}, \dots, \boldsymbol{\mu}^{[k-1]}$ fixed

$$z_\ell^{(i)} \leftarrow \begin{cases} 1 & \text{if } \ell = \operatorname*{argmin}_{\ell'} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{[\ell']} \right\|, \\ 0 & \text{otherwise.} \end{cases}$$

Updating the cluster indicator $\boldsymbol{z}^{(i)}$ of $\boldsymbol{x}^{(i)}$
with the cluster that has the nearest centroid to $\boldsymbol{x}^{(i)}$.

Update of centroids:

Minimise $J\left(\boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m-1)}, \boldsymbol{\mu}^{[0]}, \boldsymbol{\mu}^{[1]}, \dots, \boldsymbol{\mu}^{[k-1]}\right)$

$$:= \sum_{i=0}^{m-1} \sum_{\ell=0}^{k-1} z_\ell^{(i)} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{[\ell]} \right\|^2$$

w.r.t. $\boldsymbol{\mu}^{[0]}, \boldsymbol{\mu}^{[1]}, \dots, \boldsymbol{\mu}^{[k-1]}$ with $\boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m-1)}$ fixed

The sum of the data points that belongs to cluster ℓ

$$\boldsymbol{\mu}^{[\ell]} \leftarrow \frac{\sum_{i=0}^{m-1} z_\ell^{(i)} \boldsymbol{x}^{(i)}}{\sum_{i=0}^{m-1} z_\ell^{(i)}}$$

The number of the data points that belongs to cluster ℓ

Update of centroids:

Minimise $J\left(\boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m-1)}, \boldsymbol{\mu}^{[0]}, \boldsymbol{\mu}^{[1]}, \ldots, \boldsymbol{\mu}^{[k-1]}\right)$

$$:= \sum_{i=0}^{m-1} \sum_{\ell=0}^{k-1} z_\ell^{(i)} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{[\ell]} \right\|^2$$

w.r.t. $\boldsymbol{\mu}^{[0]}, \boldsymbol{\mu}^{[1]}, \ldots, \boldsymbol{\mu}^{[k-1]}$ with $\boldsymbol{z}^{(0)}, \boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m-1)}$ fixed

The sum of the data points that belongs to cluster ℓ

$$\boldsymbol{\mu}^{[\ell]} \leftarrow \frac{\sum_{i=0}^{m-1} z_\ell^{(i)} \boldsymbol{x}^{(i)}}{\sum_{i=0}^{m-1} z_\ell^{(i)}}$$

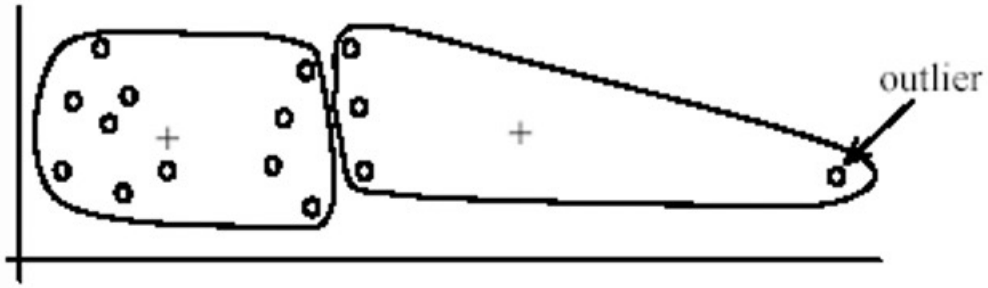
The mean of the data points that belongs to cluster ℓ

The number of the data points that belongs to cluster ℓ

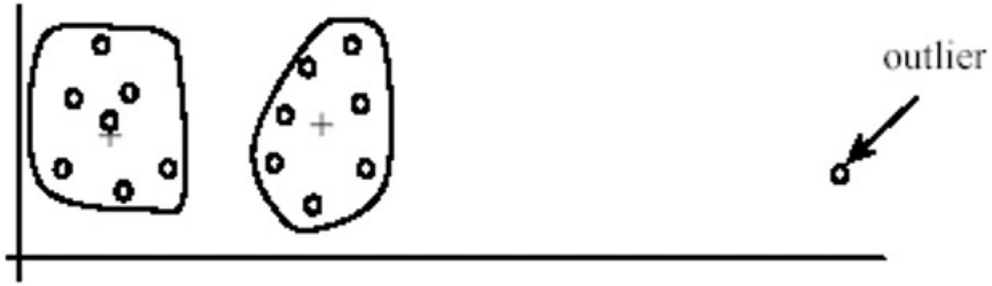
Strengths and weakness

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient (linear complexity): Time complexity: $O(mkt)$, where m is the number of data points, k is the number of clusters and t is the number of iterations.
- Weakness:
  - The k needs to be specified.
  - Sensitive to outliers.
  - Not suitable for data with hyper-sphere structure.

Sensitive to outliers

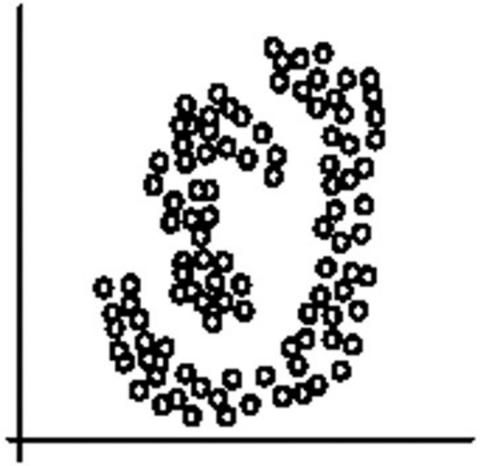


(A): Undesirable clusters

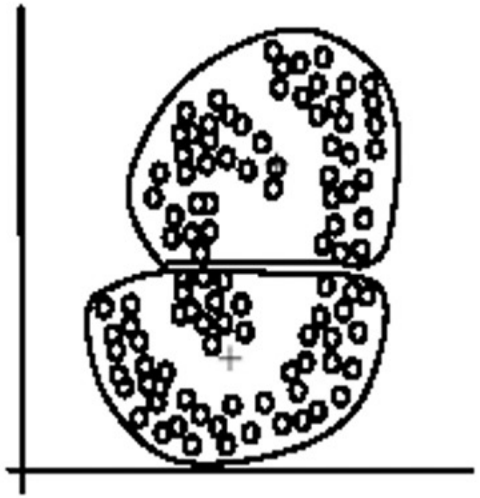


(B): Ideal clusters

Unsuitable for hyper-sphere



(A): Two natural clusters



(B): *k*-means clusters