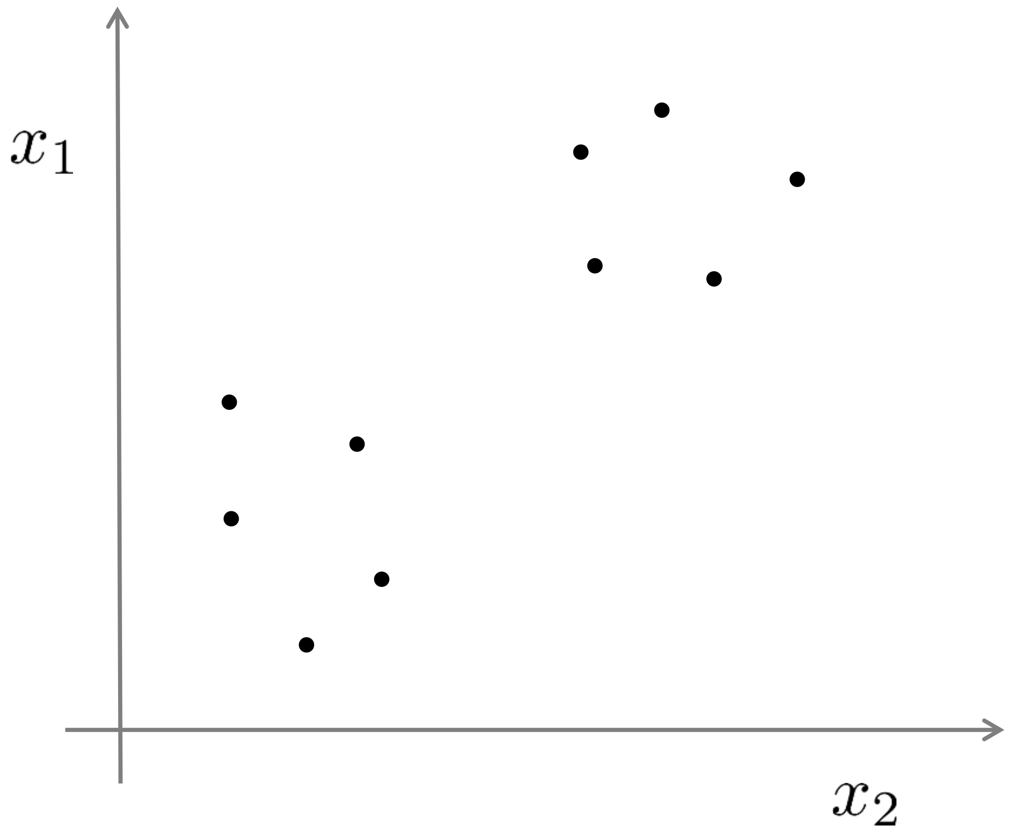
Clustering

Unsupervised Learning

Unsupervised learning



Clustering is the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often according to some defined distance measure.

Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

Different Types of clustering:

1. **Hierarchical algorithms**: these find successive clusters using previously established clusters.

   1. Agglomerative ("bottom-up"): Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.

   2. Divisive ("top-down"): Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

2. **Partitional clustering:** Partitional algorithms determine all clusters at once. They include:
   - ***K*-means**
   - Fuzzy *c*-means clustering
   - QT clustering algorithm

Common Distance measures:

- ***Distance measure*** will determine how the *similarity* of two elements is calculated and it will influence the shape of the clusters.

  1. The Euclidean distance (also called 2-norm distance) is given by:

$$d(x,y) = \sum_{i=1}^{p} | x_i - y_i |$$

2. The Manhattan distance (also called taxicab norm or 1-norm) is given by:

$$d(x,y) = \sqrt[2]{\sum_{i=1}^{p} | x_i - y_i |^2}$$

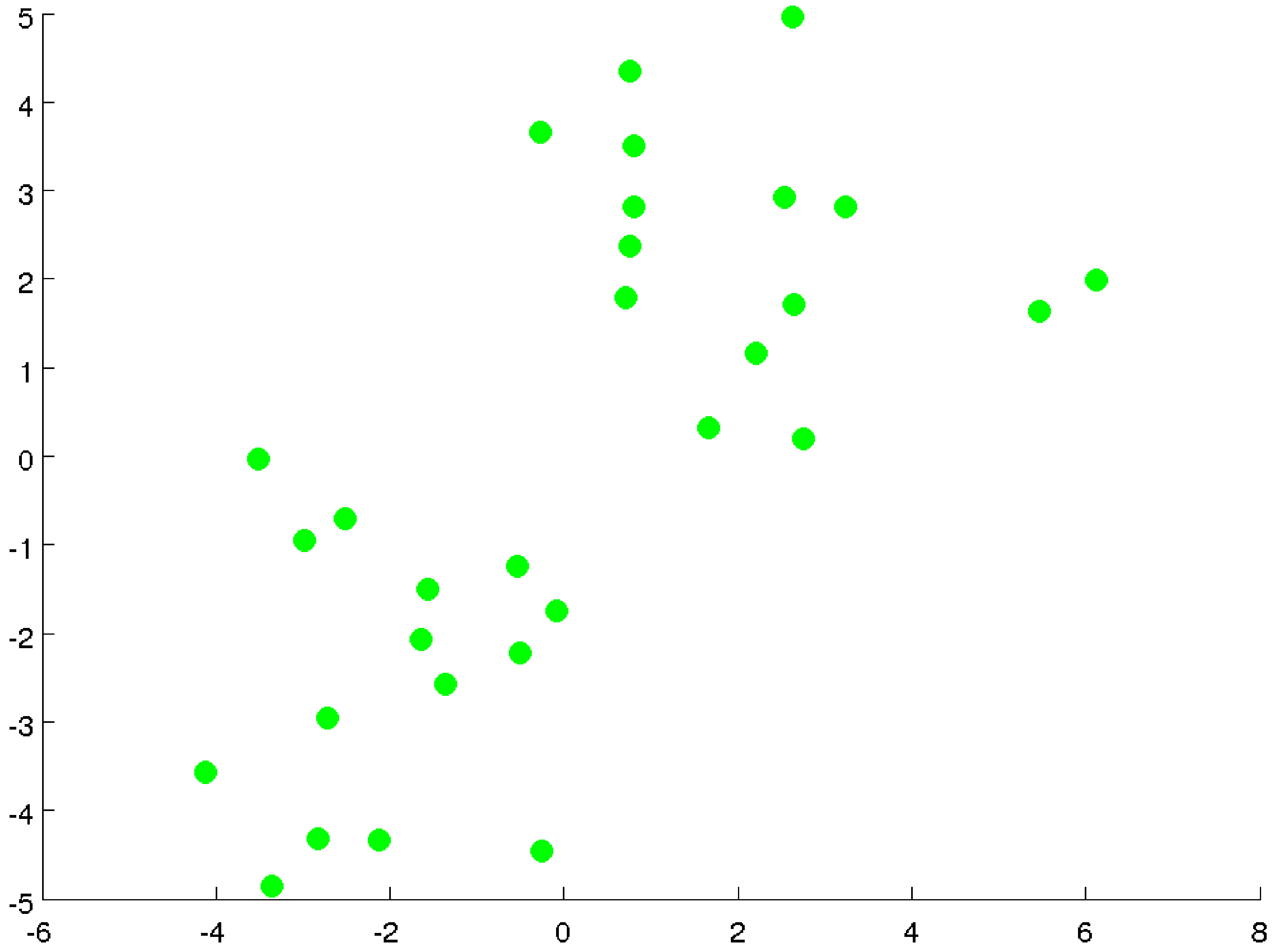
3.The maximum norm is given by:

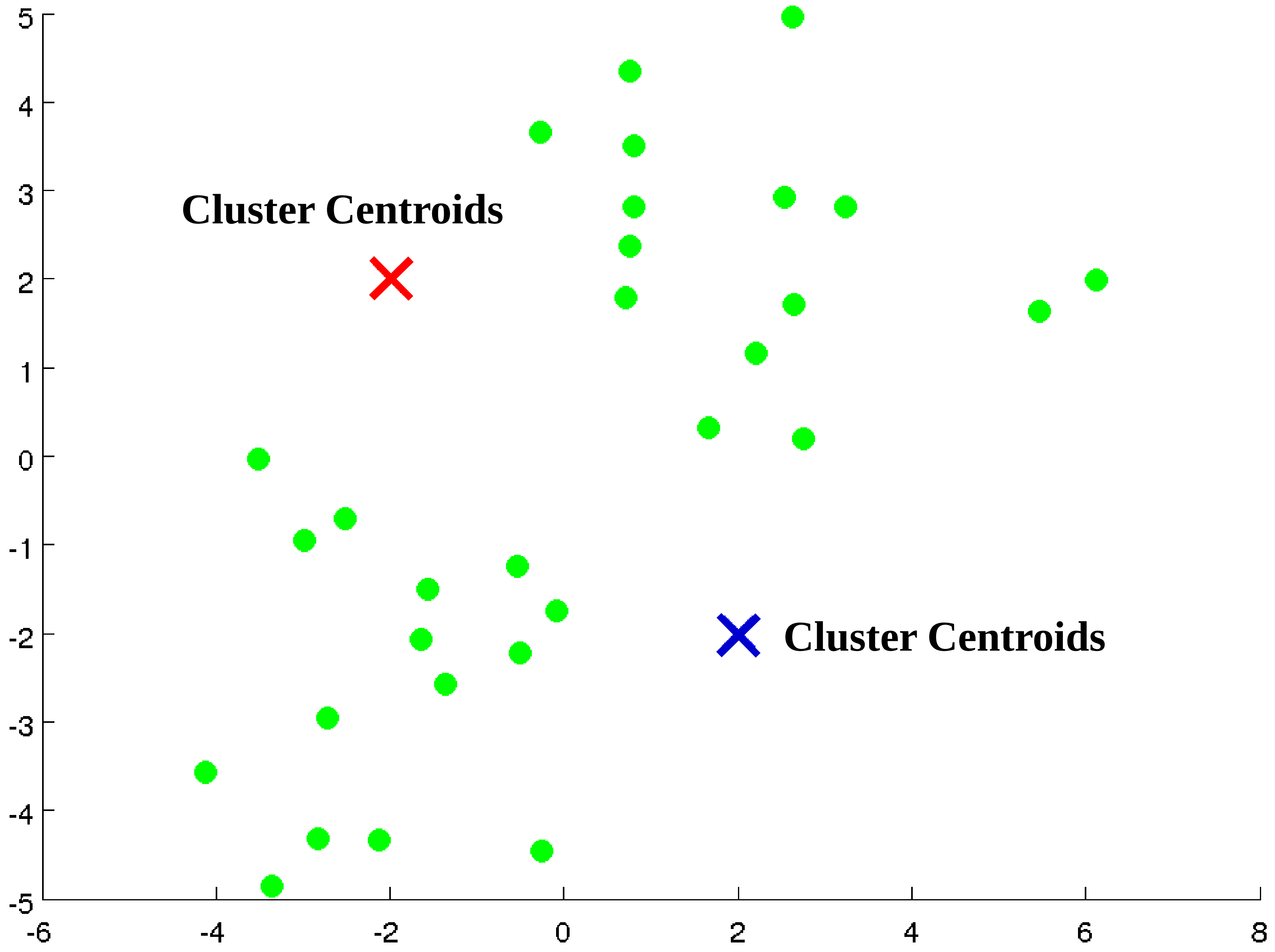
$$d(x, y) = \max_{1 \le i \le p} |x_i - y_i|$$

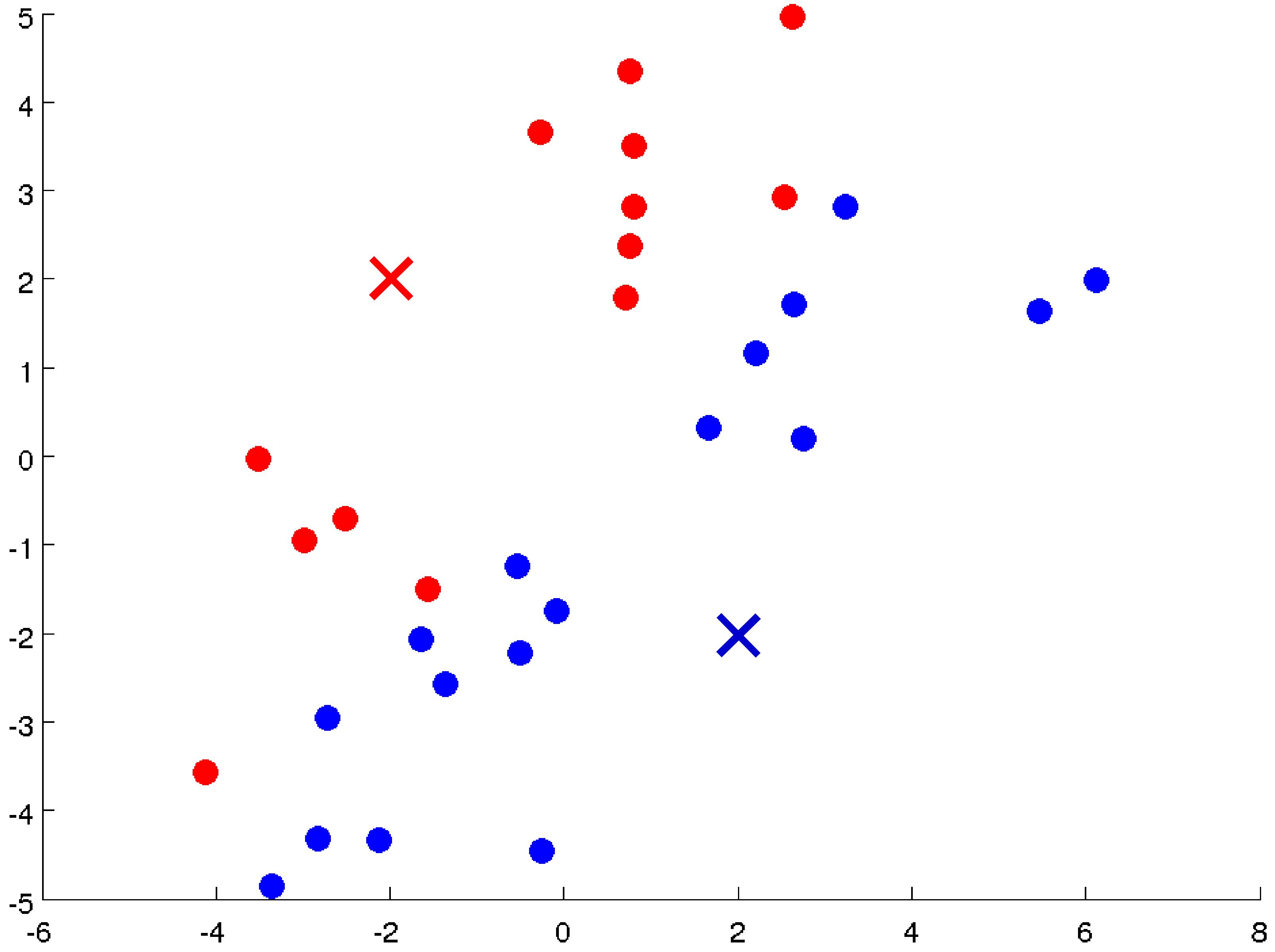
4. The Mahalanobis distance corrects data for different scales and correlations in the variables.

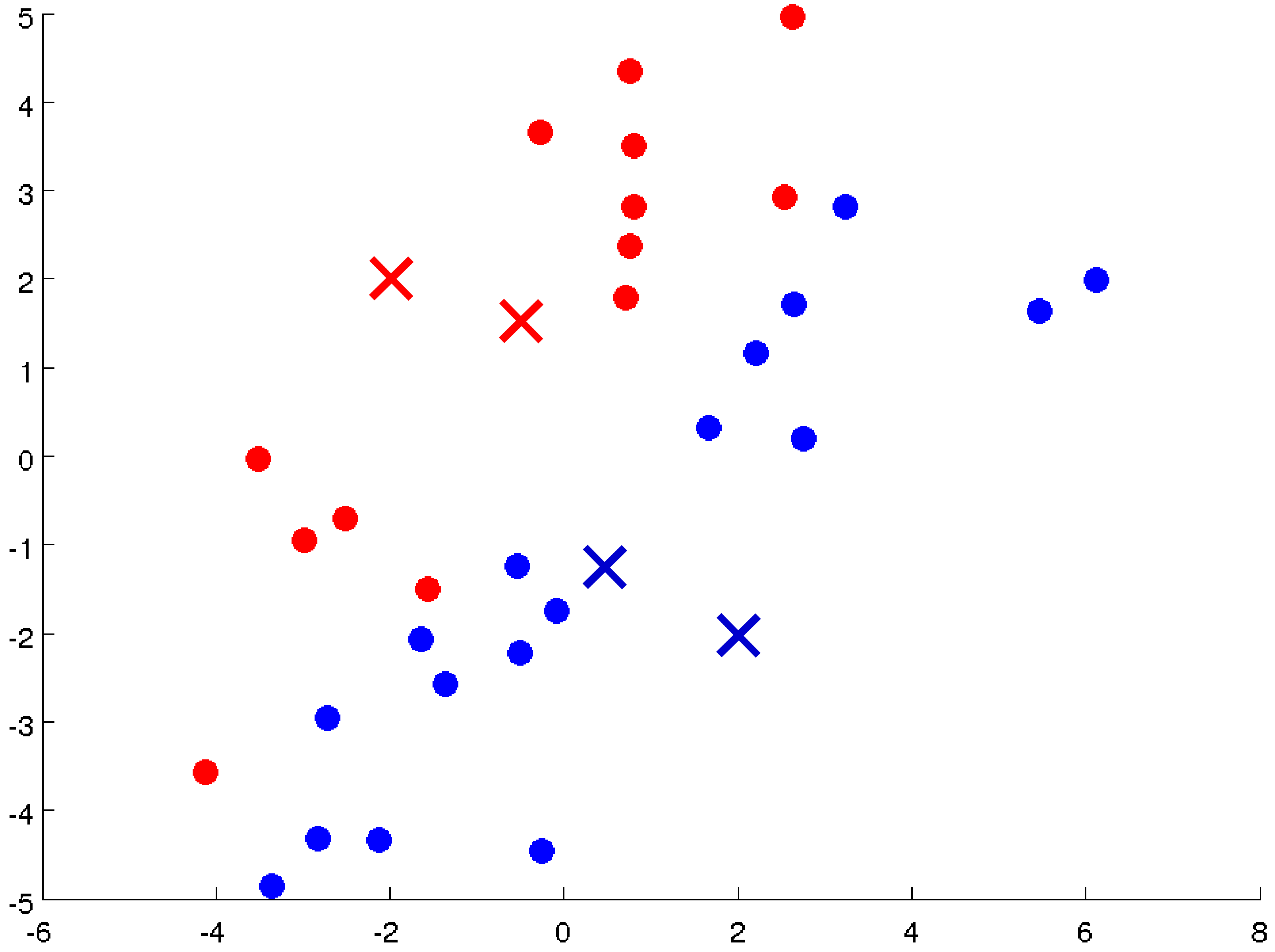
5. Inner product space: The angle between two vectors can be used as a distance measure when clustering high dimensional data.

6. Hamming distance (sometimes edit distance) measures the minimum number of substitutions required to change one member into another.

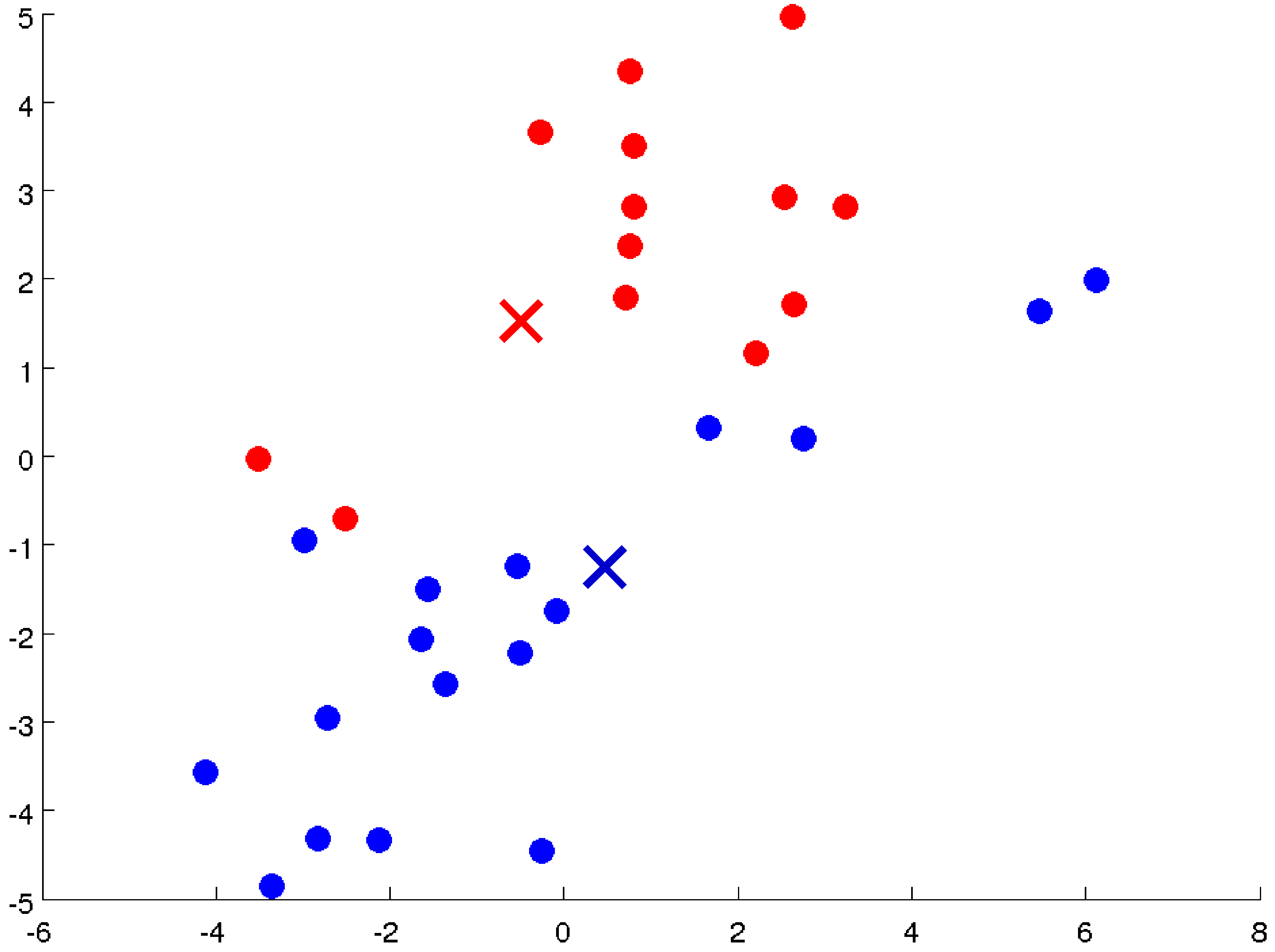

5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8

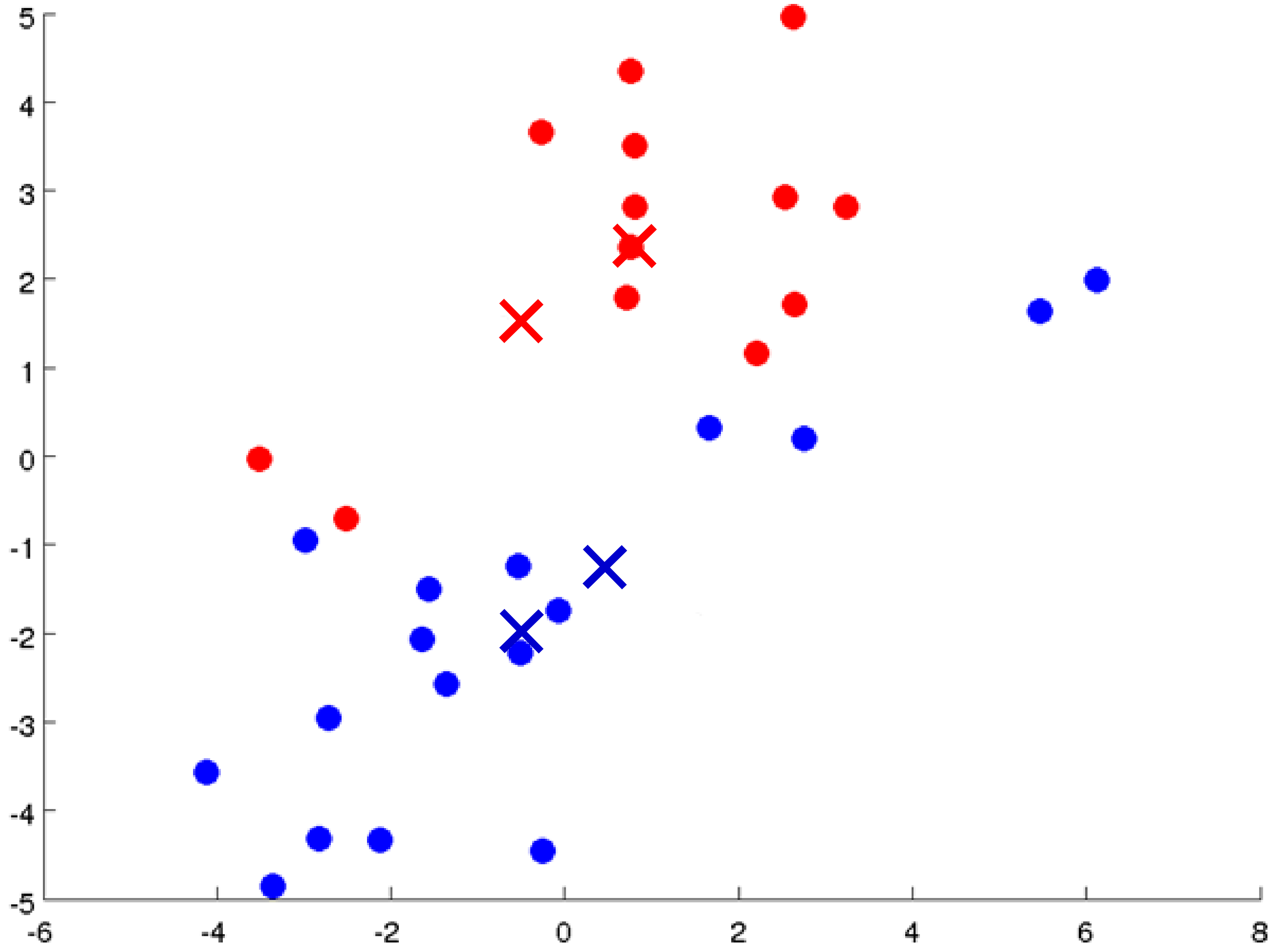

Cluster Centroids
Cluster Centroids

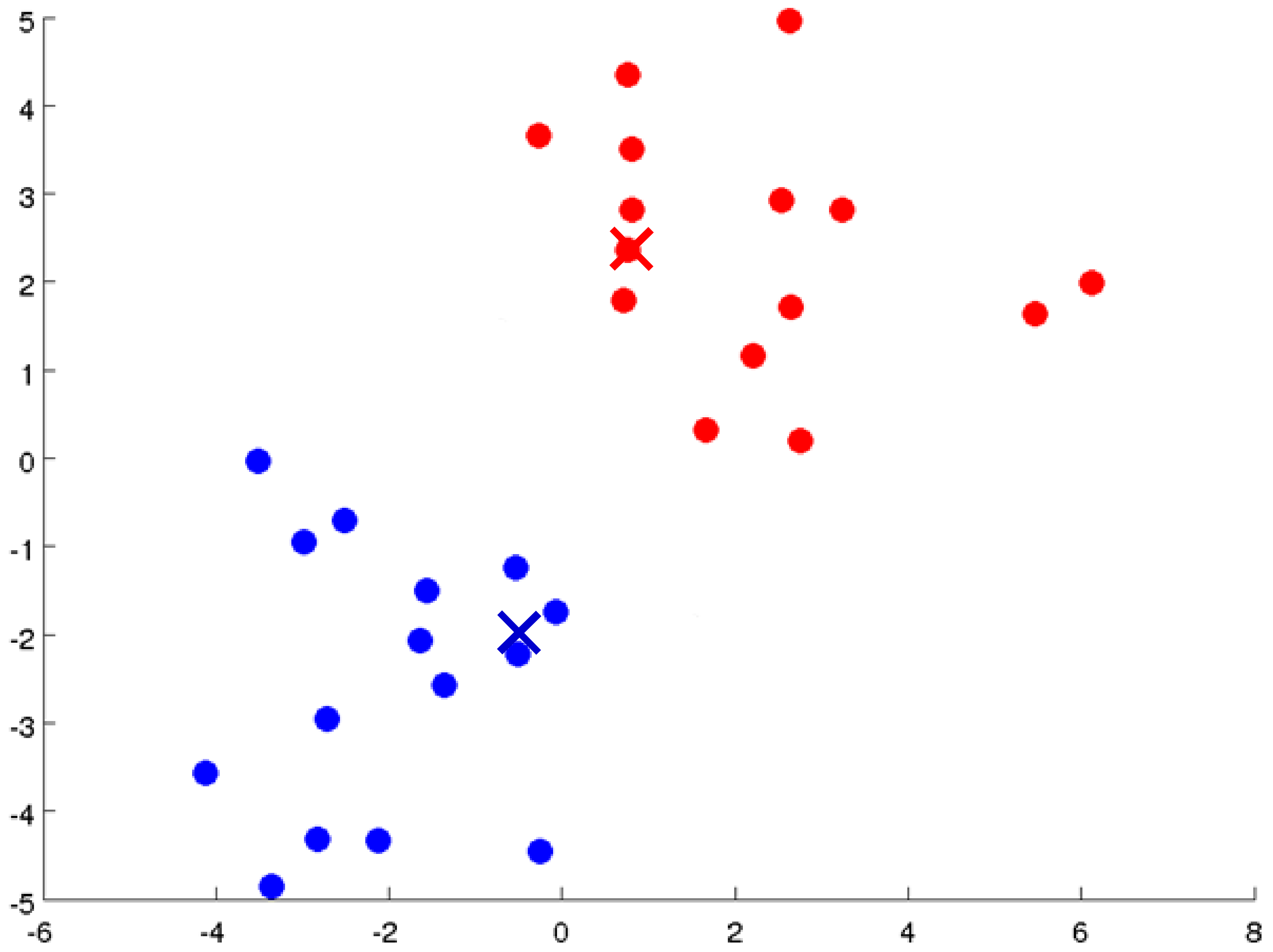

5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8

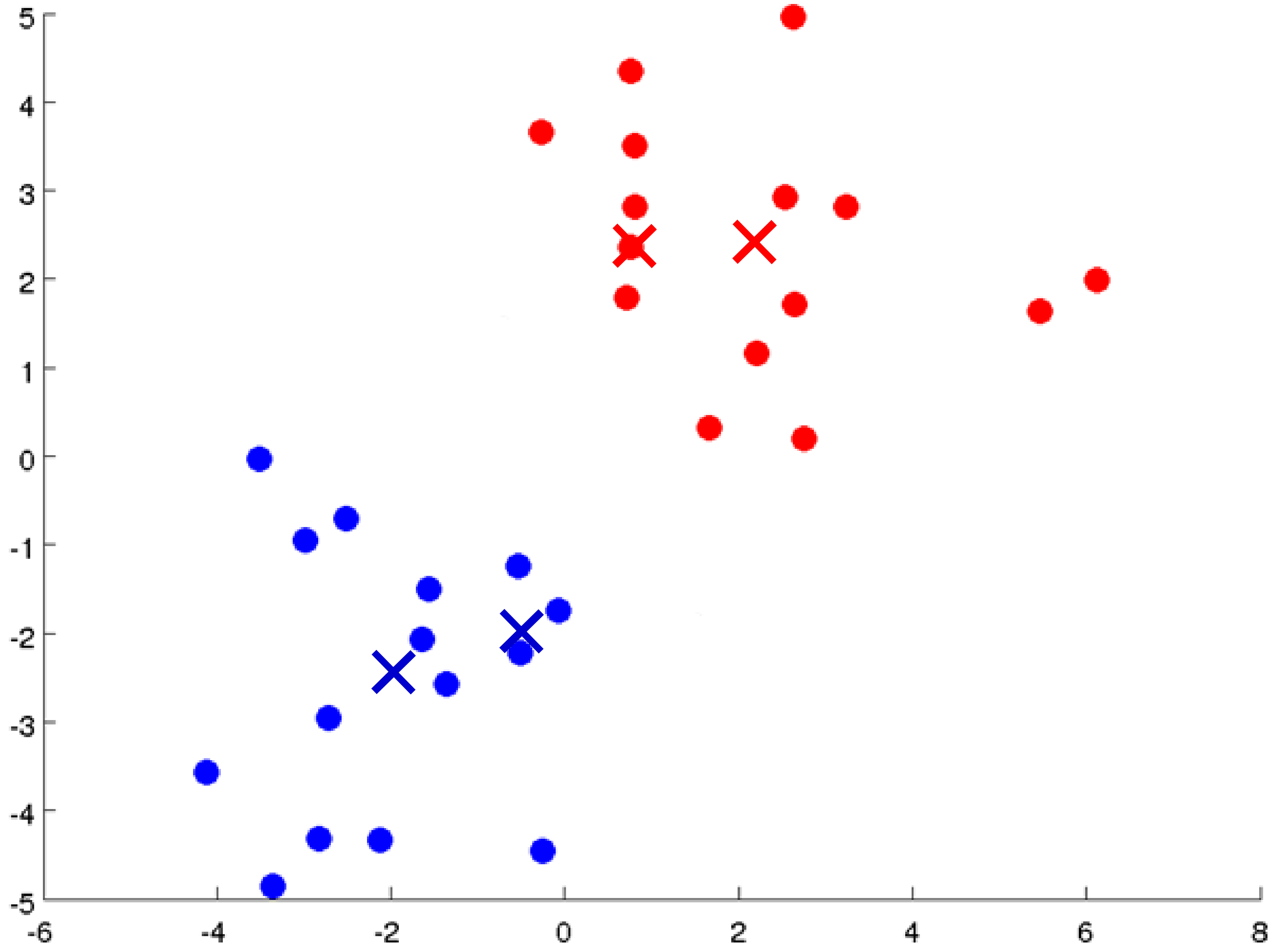


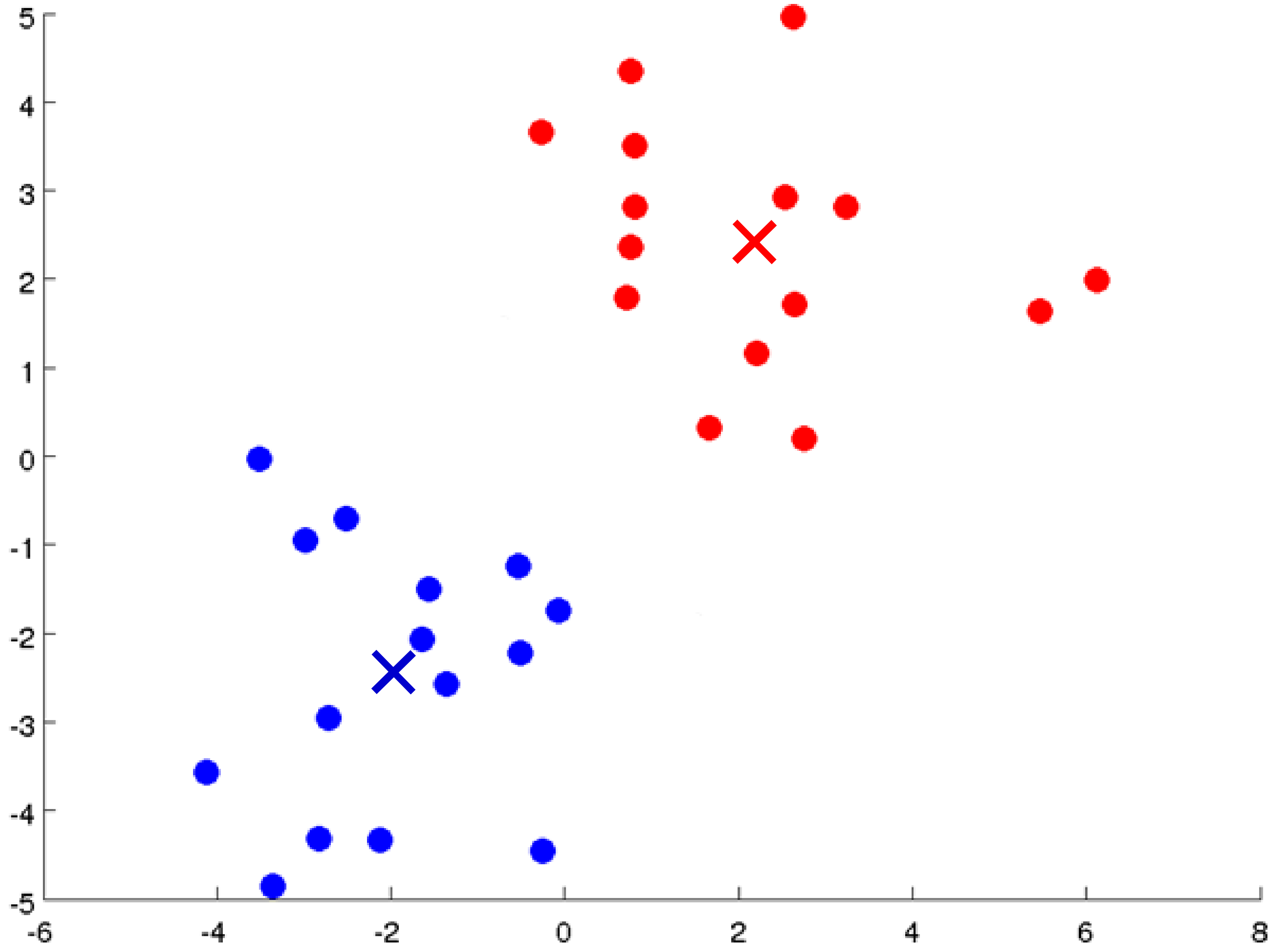
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


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


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


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


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


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

K-means algorithm

assumption

Input:

- K (number of clusters)
- Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

$x^{(i)} \in \mathbb{R}^n$ (drop $x_0 = 1$ convention)

K-means algorithm

```
Randomly initialize $K$ cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$
  Repeat {
        for $i$ = 1 to $m$
            $c^{(i)}$ := index (from 1 to $K$) of cluster centroid
                    closest to $x^{(i)}$
        for $k$ = 1 to $K$
            $\mu_k$ := average (mean) of points assigned to cluster $k$
        }
```

K-means optimization objective

$c^{(i)}$ = index of cluster (1,2,…,K) to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k ($\mu_k \in \mathbb{R}^n$)

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned

Optimization objective:

$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

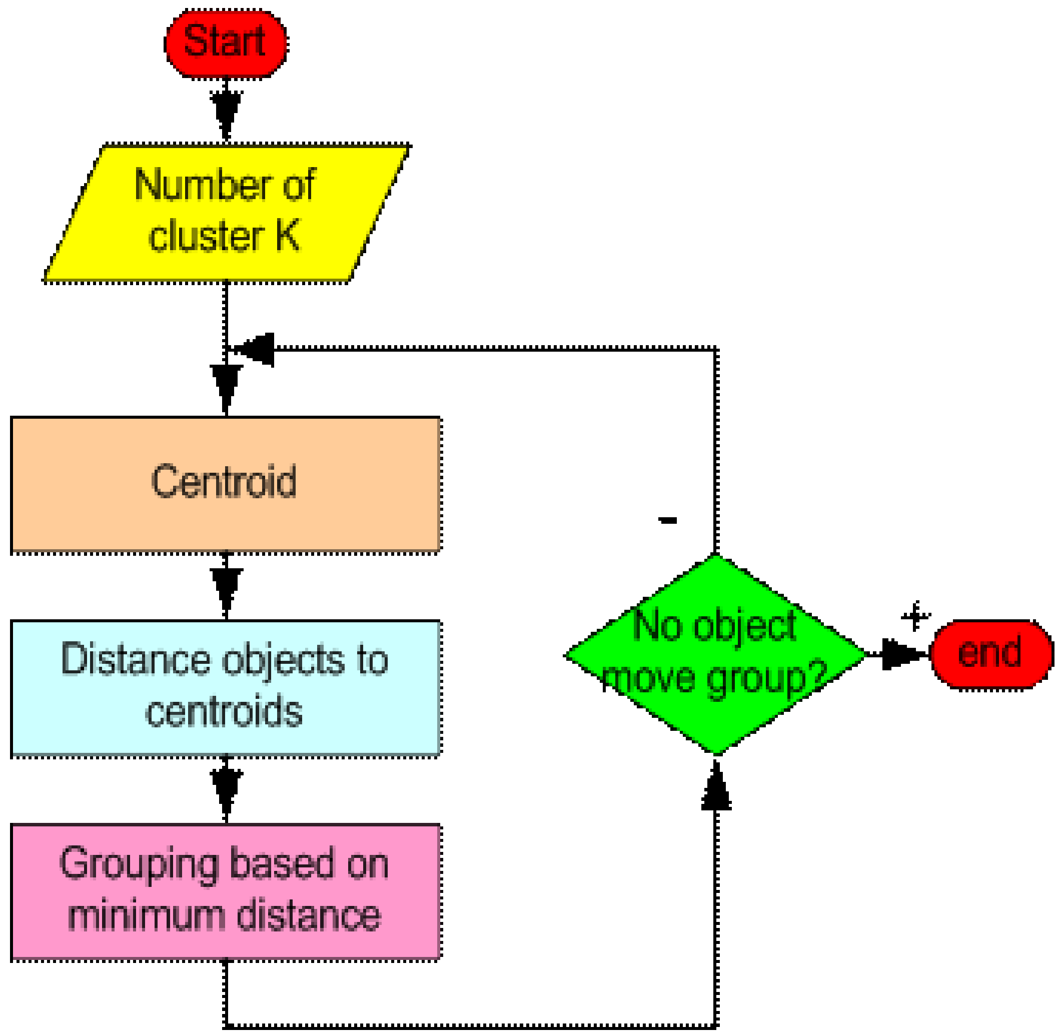
$$\min_{\substack{c^{(1)},\dots,c^{(m)}, \\ \mu_1, \dots, \mu_K}} J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K)$$

K-means algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

```
Repeat {
        for $i$ = 1 to $m$
            $c^{(i)}$ := index (from 1 to $K$) of cluster centroid
                    closest to $x^{(i)}$
        for $k$ = 1 to $K$
            $\mu_k$ := average (mean) of points assigned to cluster $k$
        }
```

K-Mean Clustering algorithm

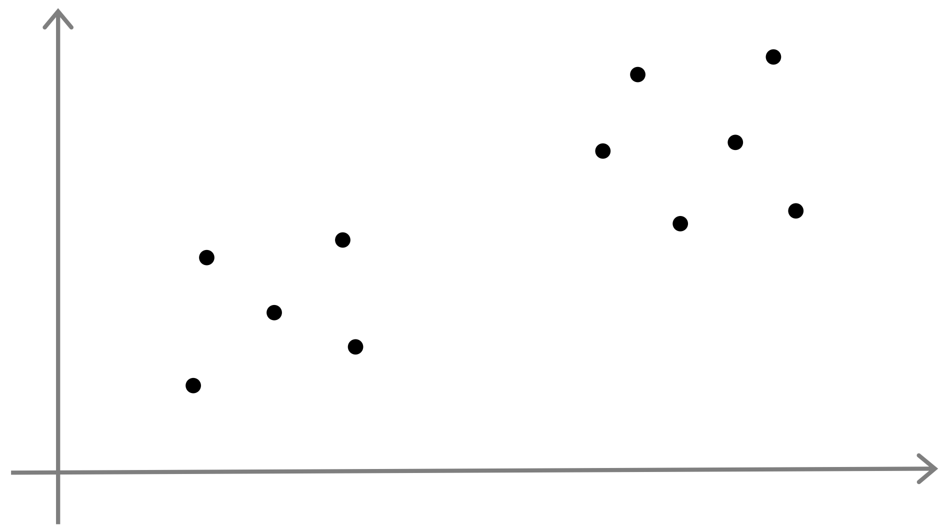

Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

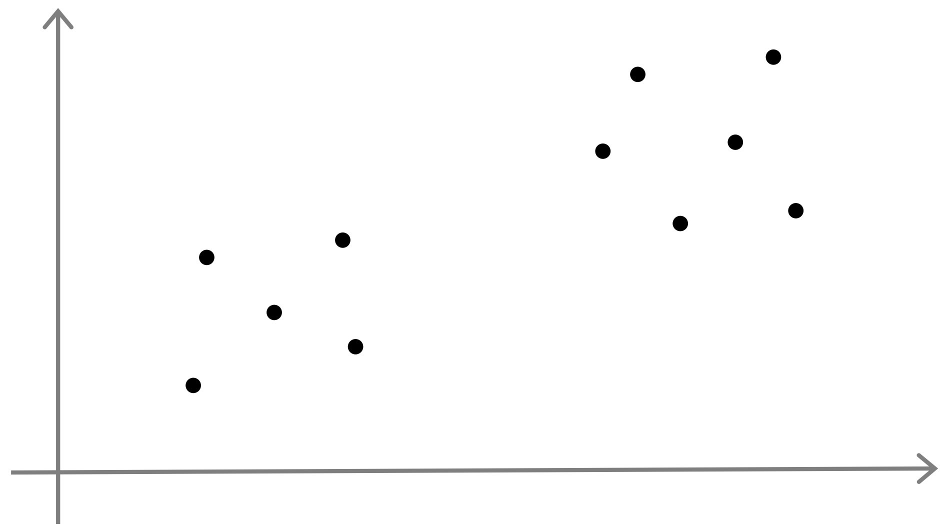
Random initialization

Should have $K < m$

Randomly pick K training examples.

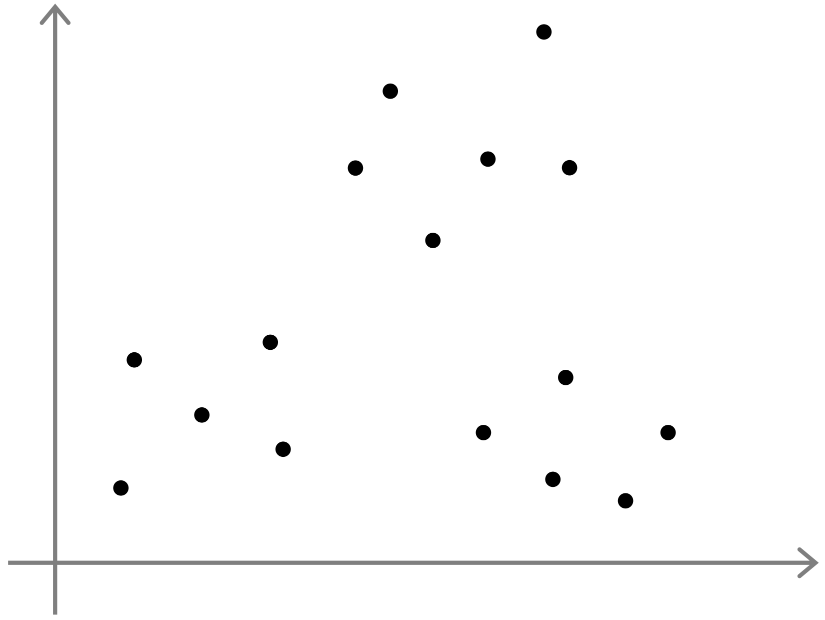
Set $\mu_1, \ldots, \mu_K$ equal to these K examples.

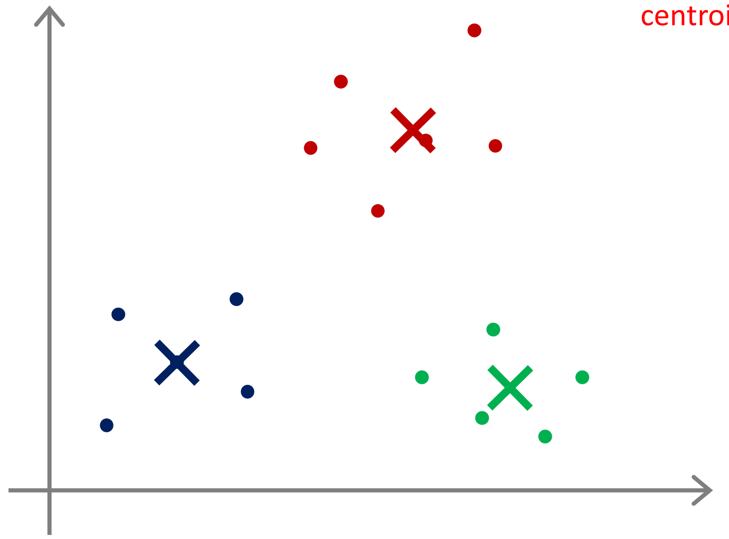


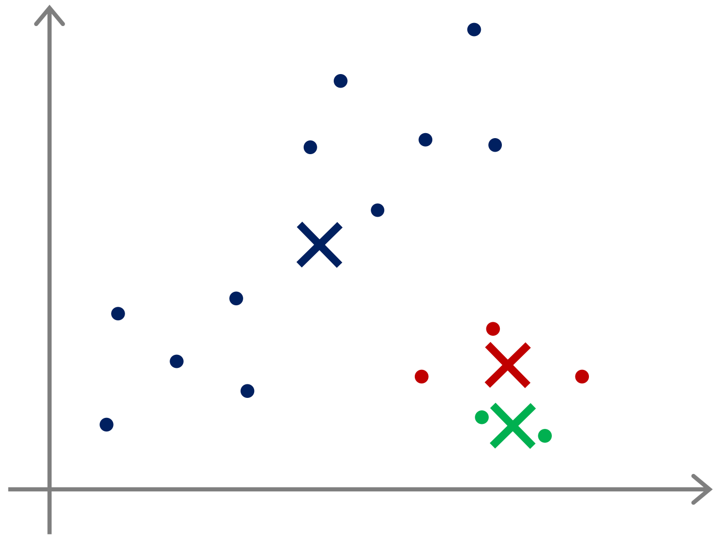


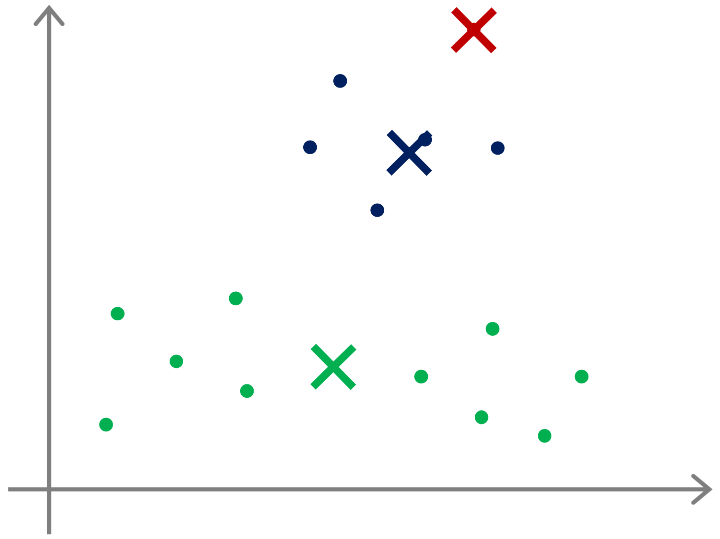
Local optima

Depending on the initialization of cluster centroids K-means can produce different results









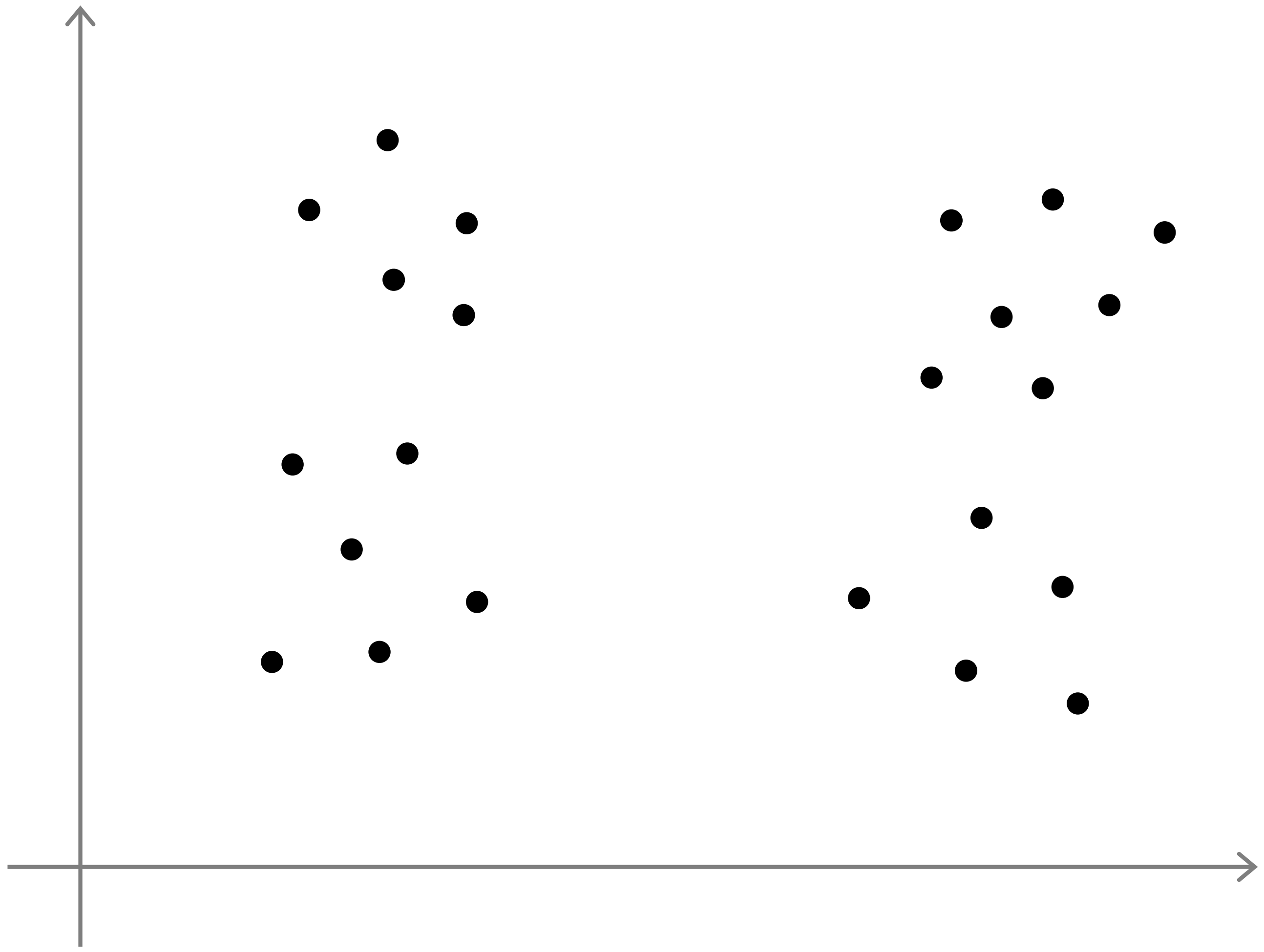
Random initialization

```
For i = 1 to 100 {

        Randomly initialize K-means.
        Run K-means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$
        Compute cost function (distortion)
                $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$
        }
```

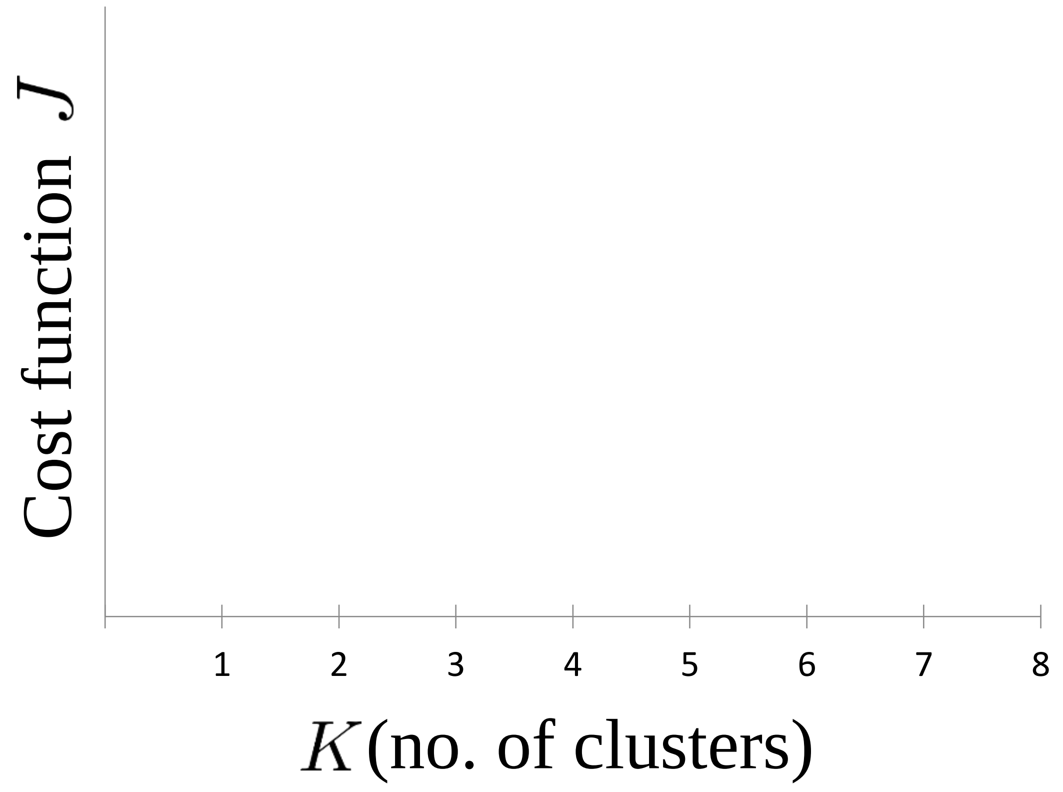
Pick clustering that gave lowest cost $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

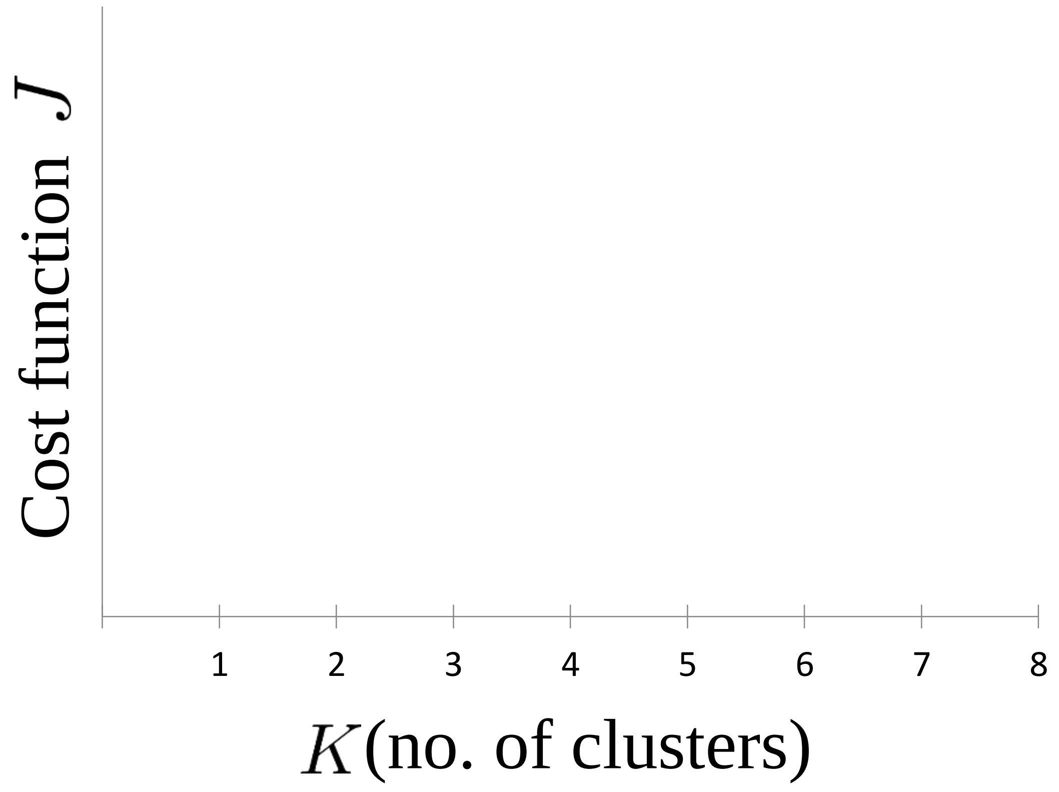
What is the right value of K?



Choosing the value of K

Elbow method:

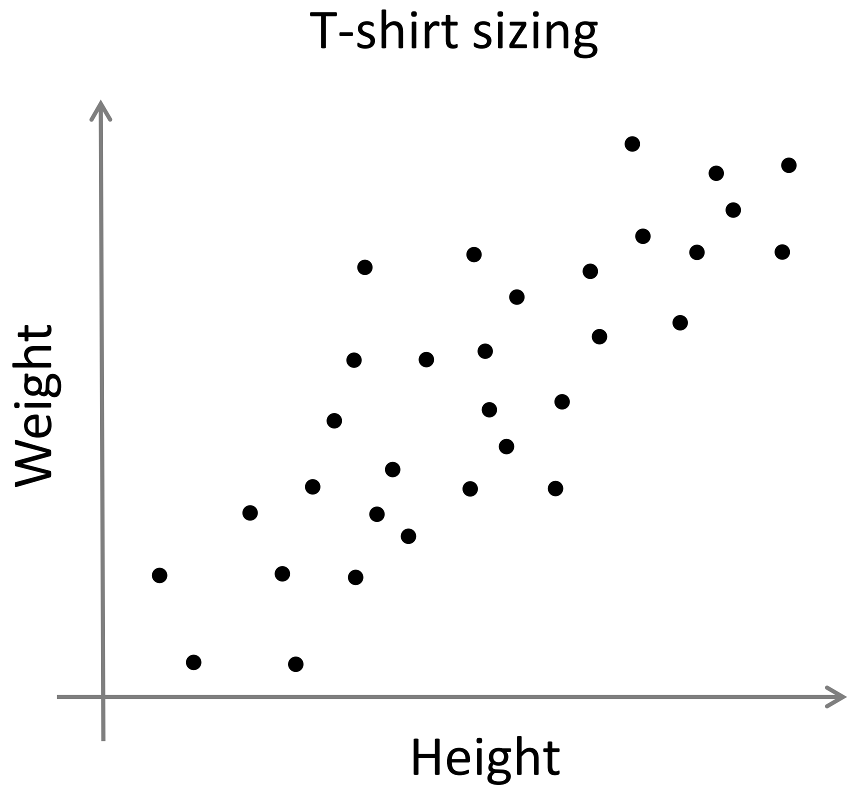


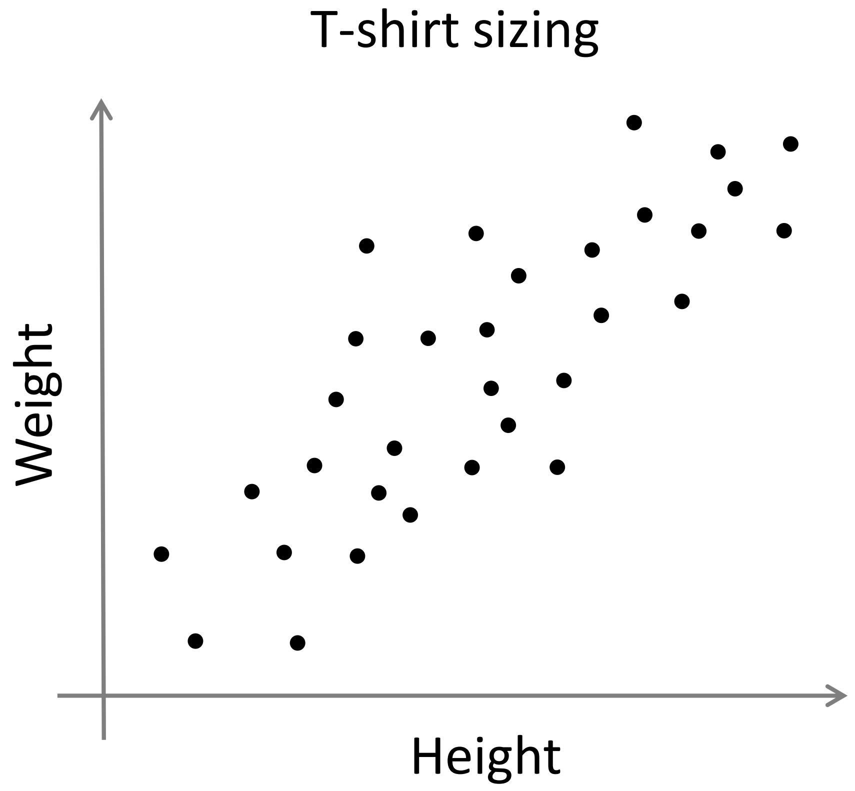


Choosing the value of K

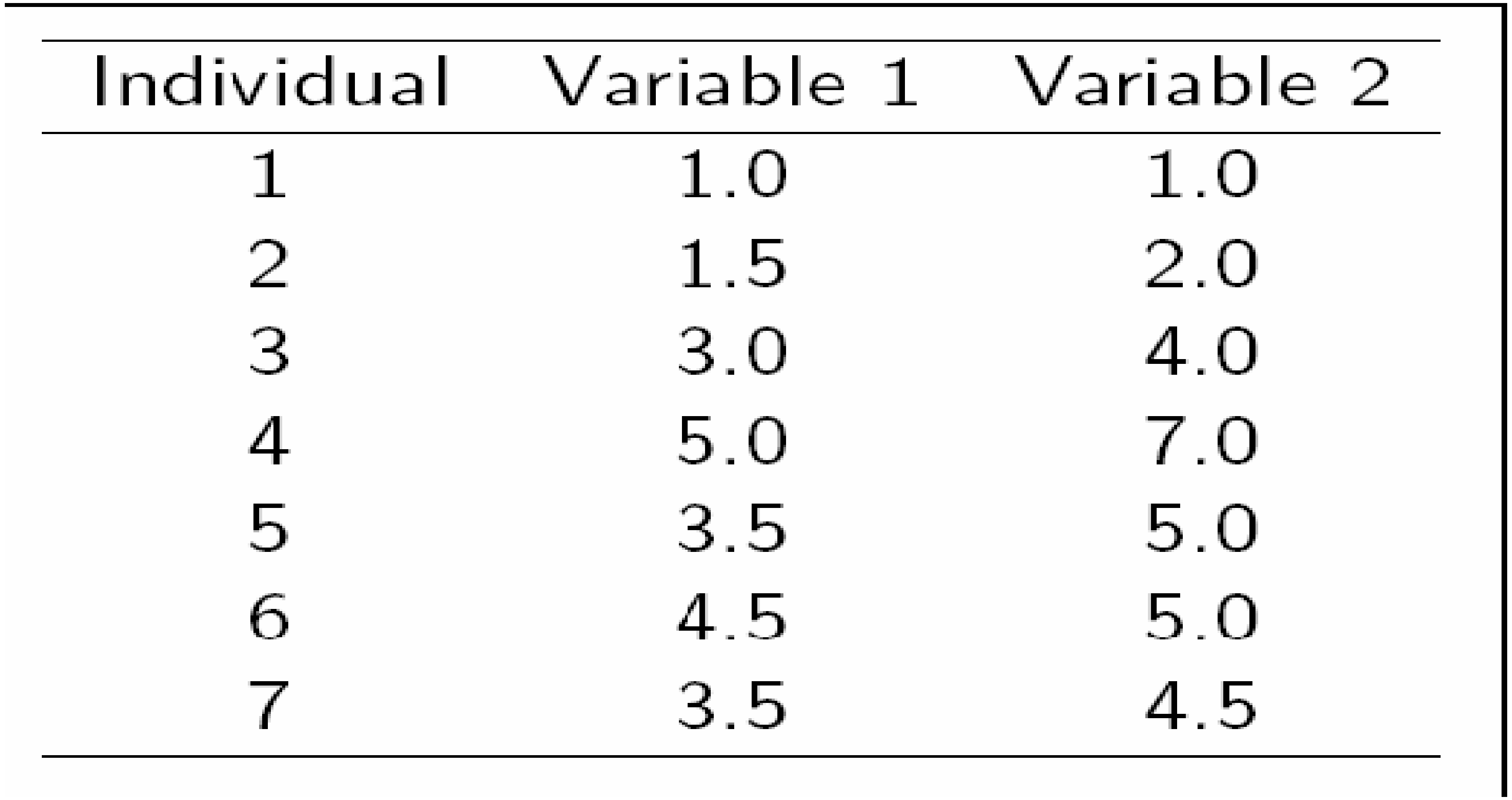
Sometimes, you're running K-means to get clusters to use for some later/downstream purpose. Evaluate K-means based on a metric for how well it performs for that later purpose.

E.g.





A Simple example showing the implementation of k-means algorithm (using K=2)

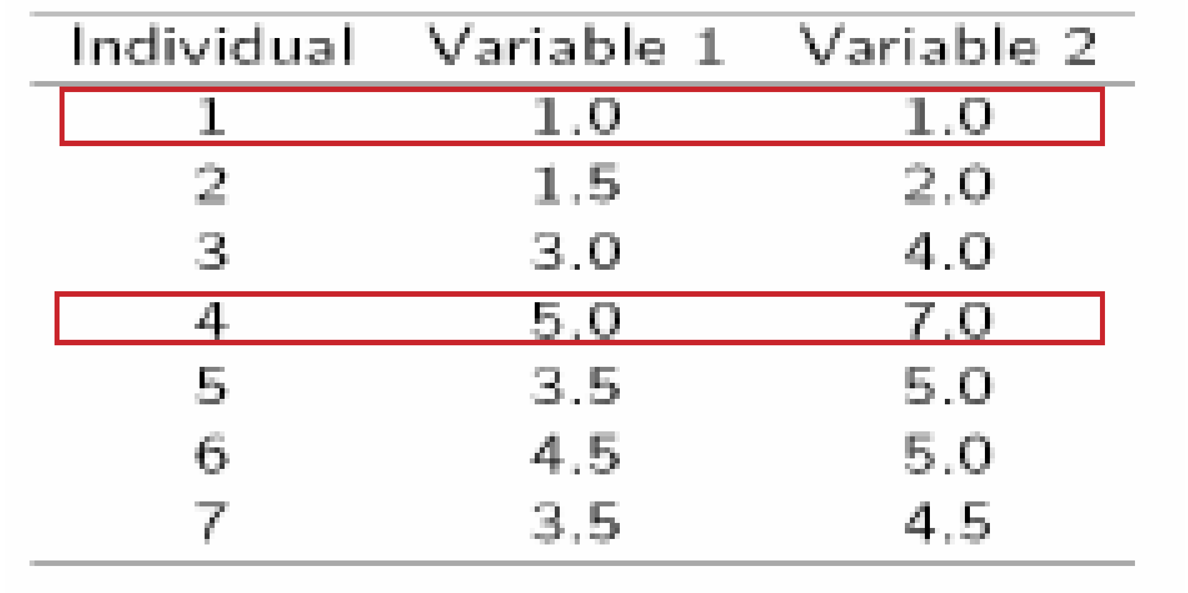


Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

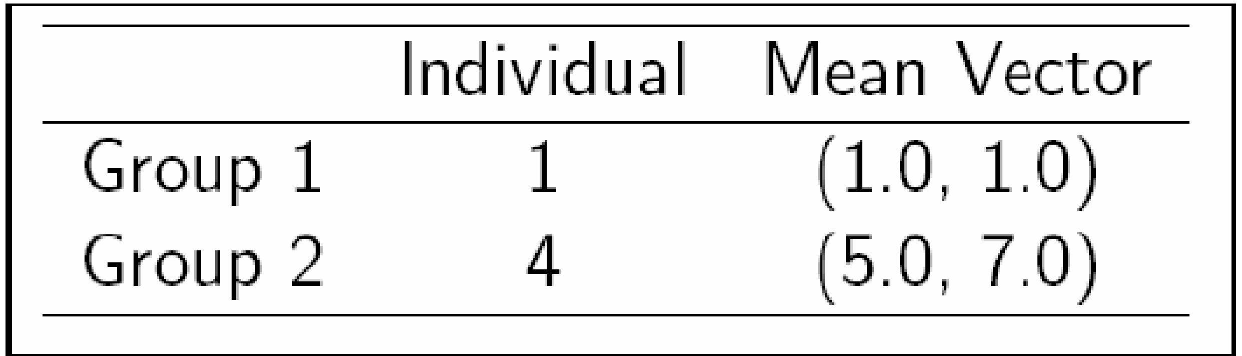
<u>Step 1</u>:

<u>Initialization</u>: 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:

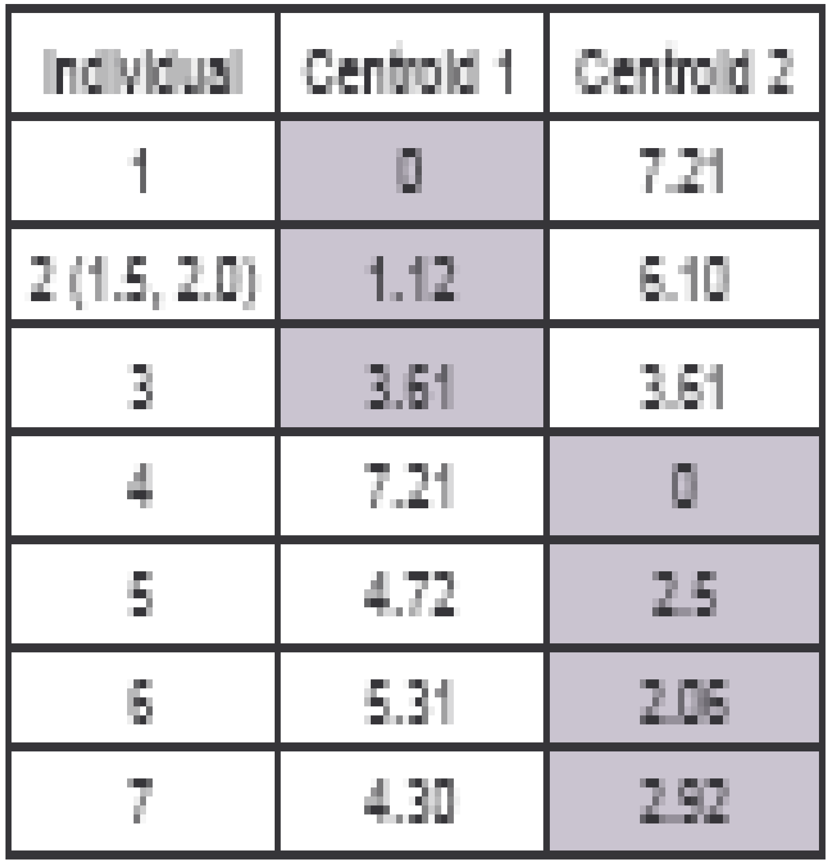
  {1,2,3} and {4,5,6,7}.

- 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

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

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

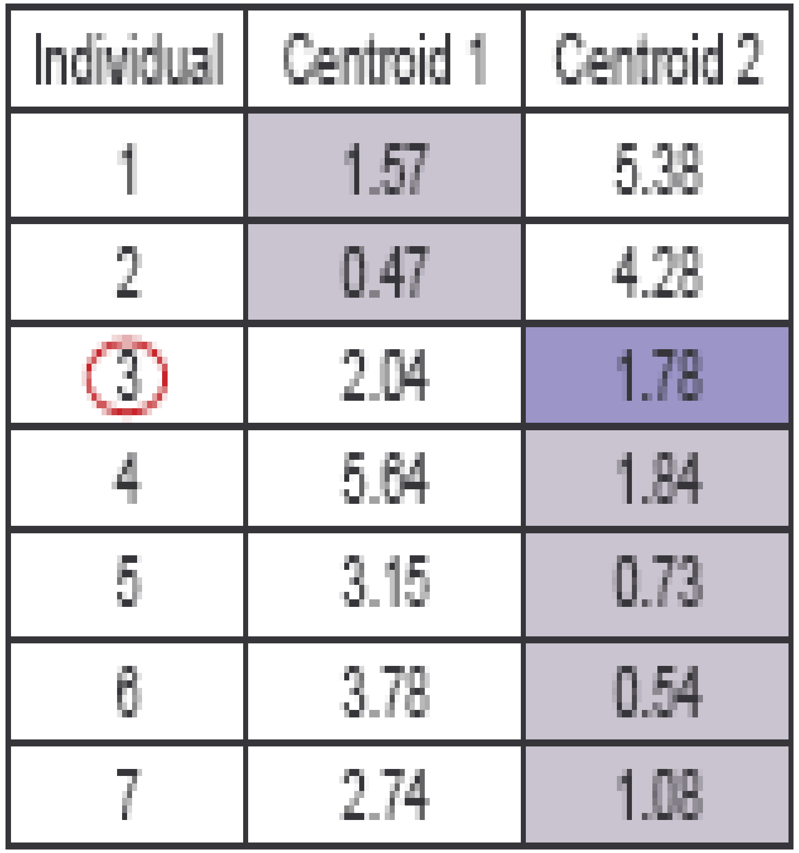
<u>Step 3:</u>

- 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)



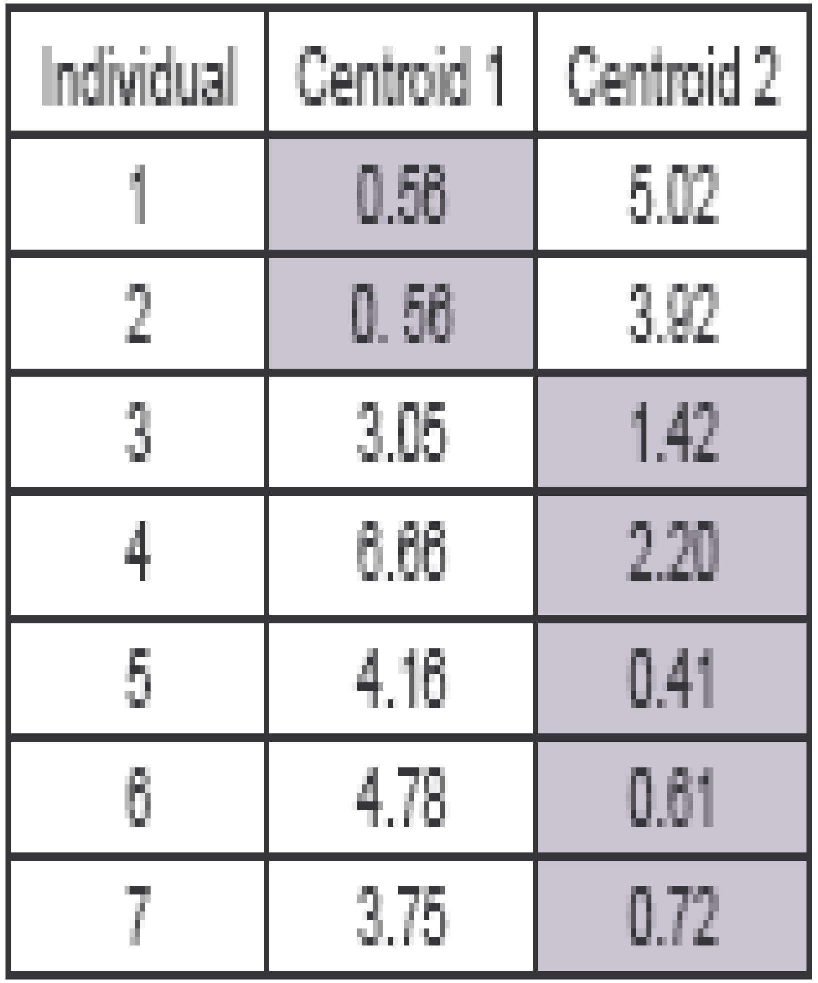
Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

- Step 4 :

  The clusters obtained are:

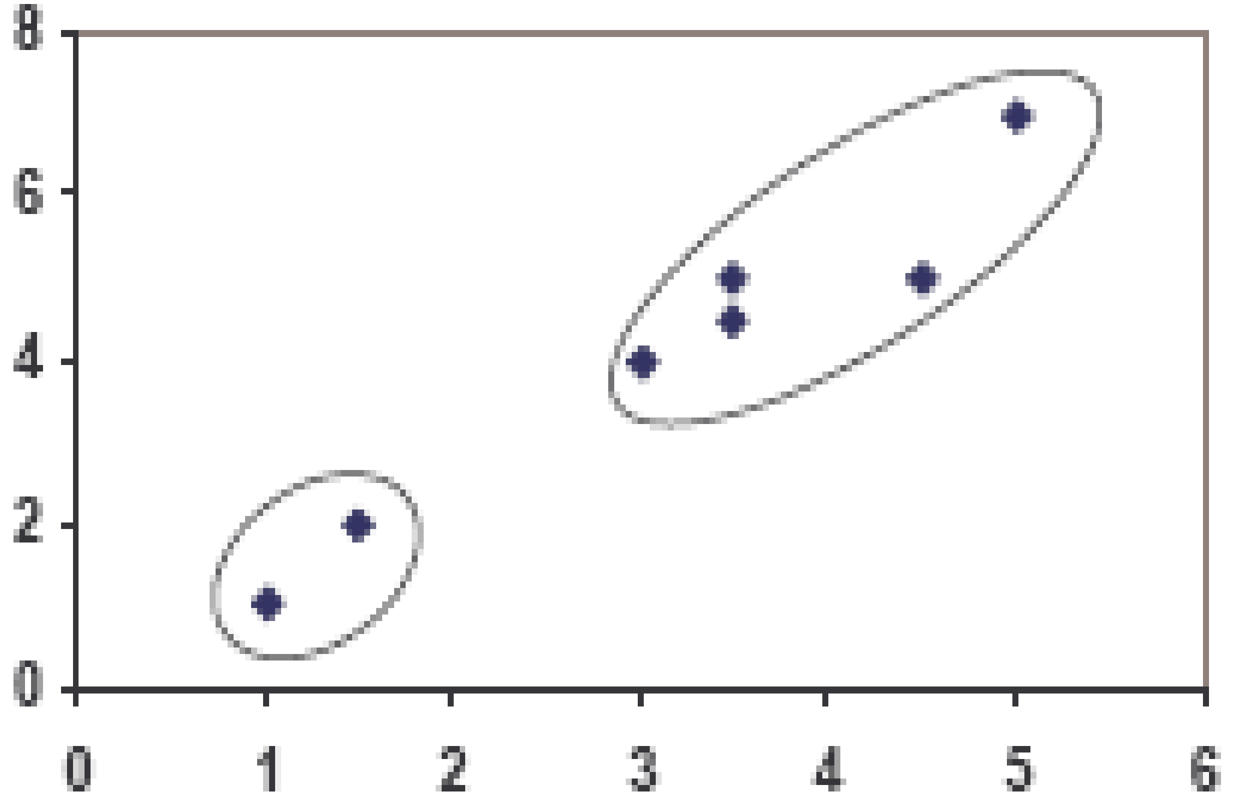
  {1,2} and {3,4,5,6,7}

- 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

PLOT



Weaknesses of K-Mean Clustering

1. When the numbers of data are not so many, initial grouping will determine the cluster significantly.
2. The number of cluster, K, must be determined before hand. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments.
3. We never know the real cluster, using the same data, because if it is inputted in a different order it may produce different cluster if the number of data is few.
4. It is sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the *local optimum*.

Applications of K-Mean Clustering

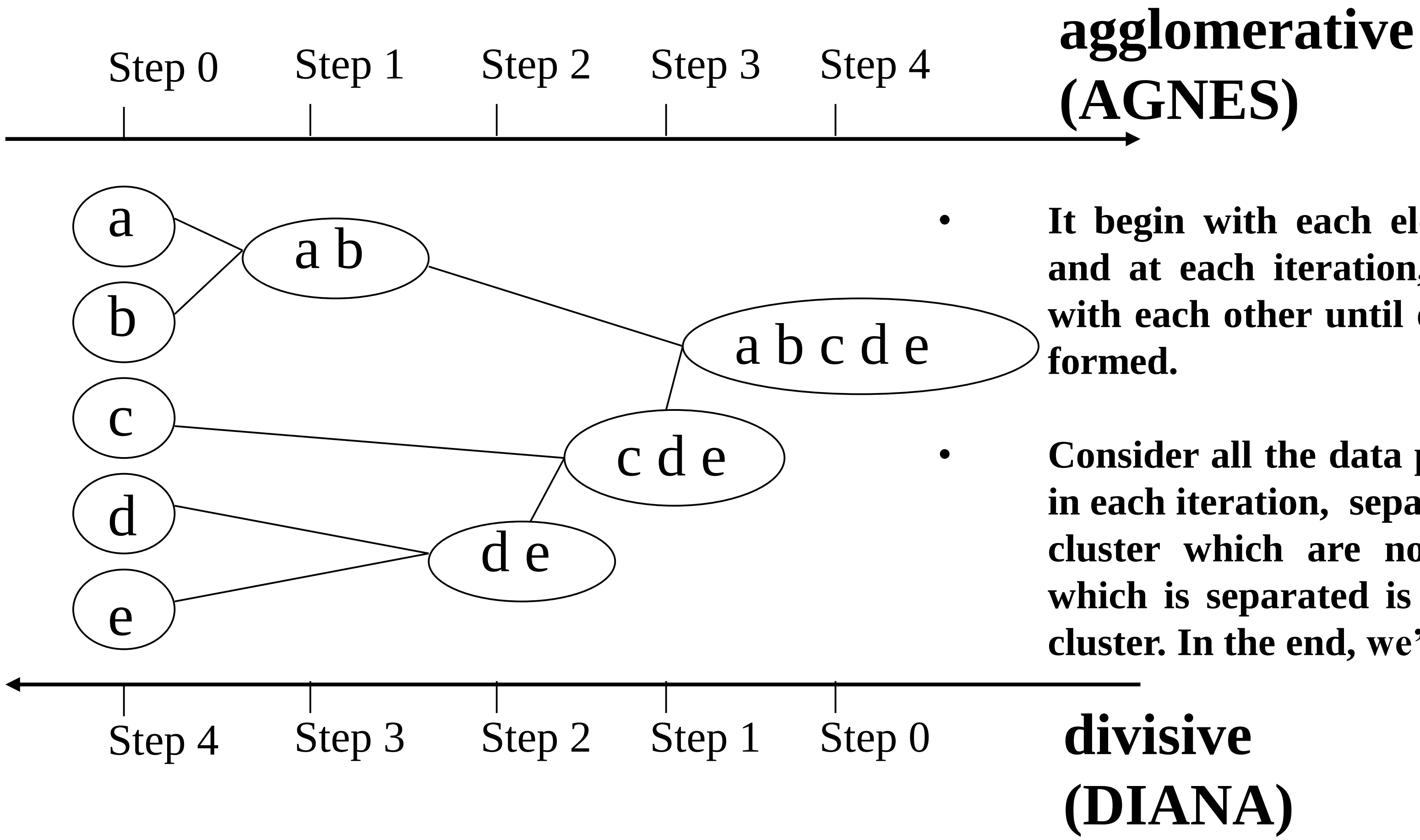
- It is relatively *efficient and fast*. It computes result at **O(tkn),** where n is number of objects or points, k is number of clusters and t is number of iterations.
- k-means clustering can be applied to *machine learning or data mining*
- *Used on acoustic data in speech understanding to convert waveforms into one of k categories (known as Vector Quantization or Image Segmentation).*
- *Also used for choosing color palettes on old fashioned graphical display devices and Image Quantization.*

CONCLUSION

- *K-means algorithm is* useful for undirected knowledge discovery and is relatively simple. K-means has found wide spread usage in lot of fields, ranging from unsupervised learning of neural network, Pattern recognitions, Classification analysis, Artificial intelligence, image processing, machine vision, and many others.

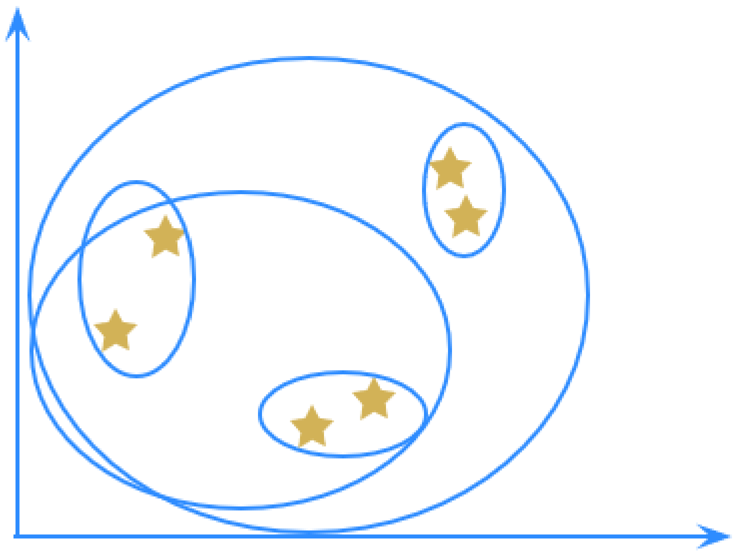
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition Lets say we have five data points {a, b, c, d, e}.

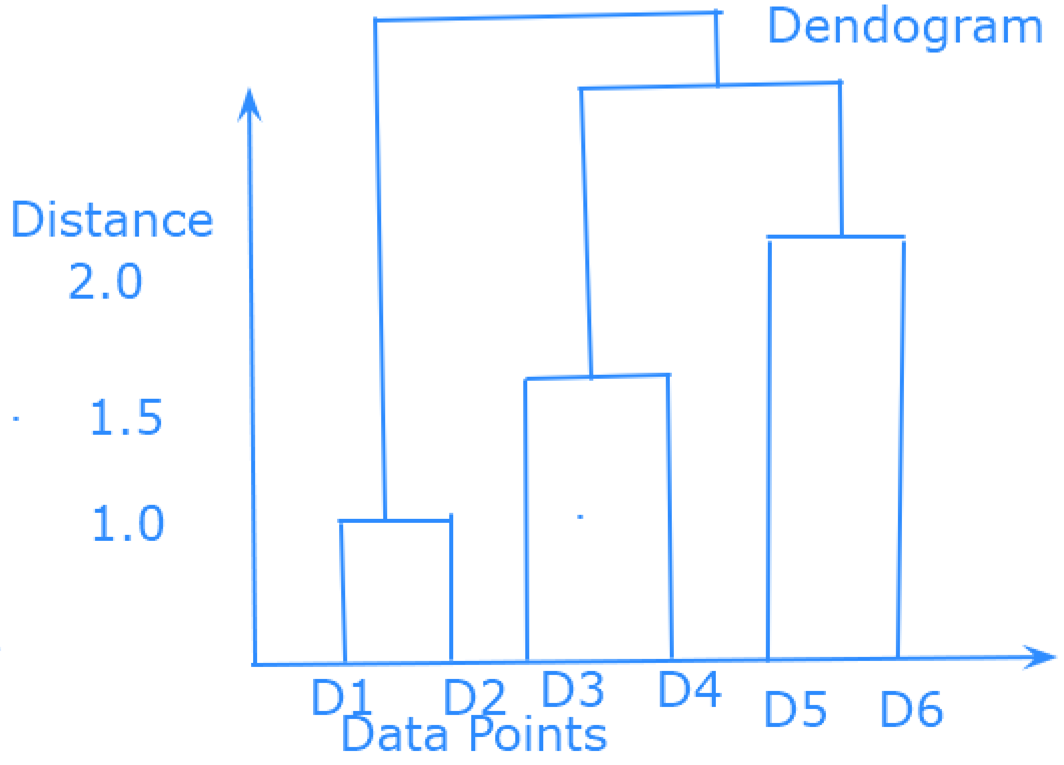


- **It begin with each element as a separate cluster and at each iteration, the similar clusters merge with each other until one cluster or K clusters are formed.**

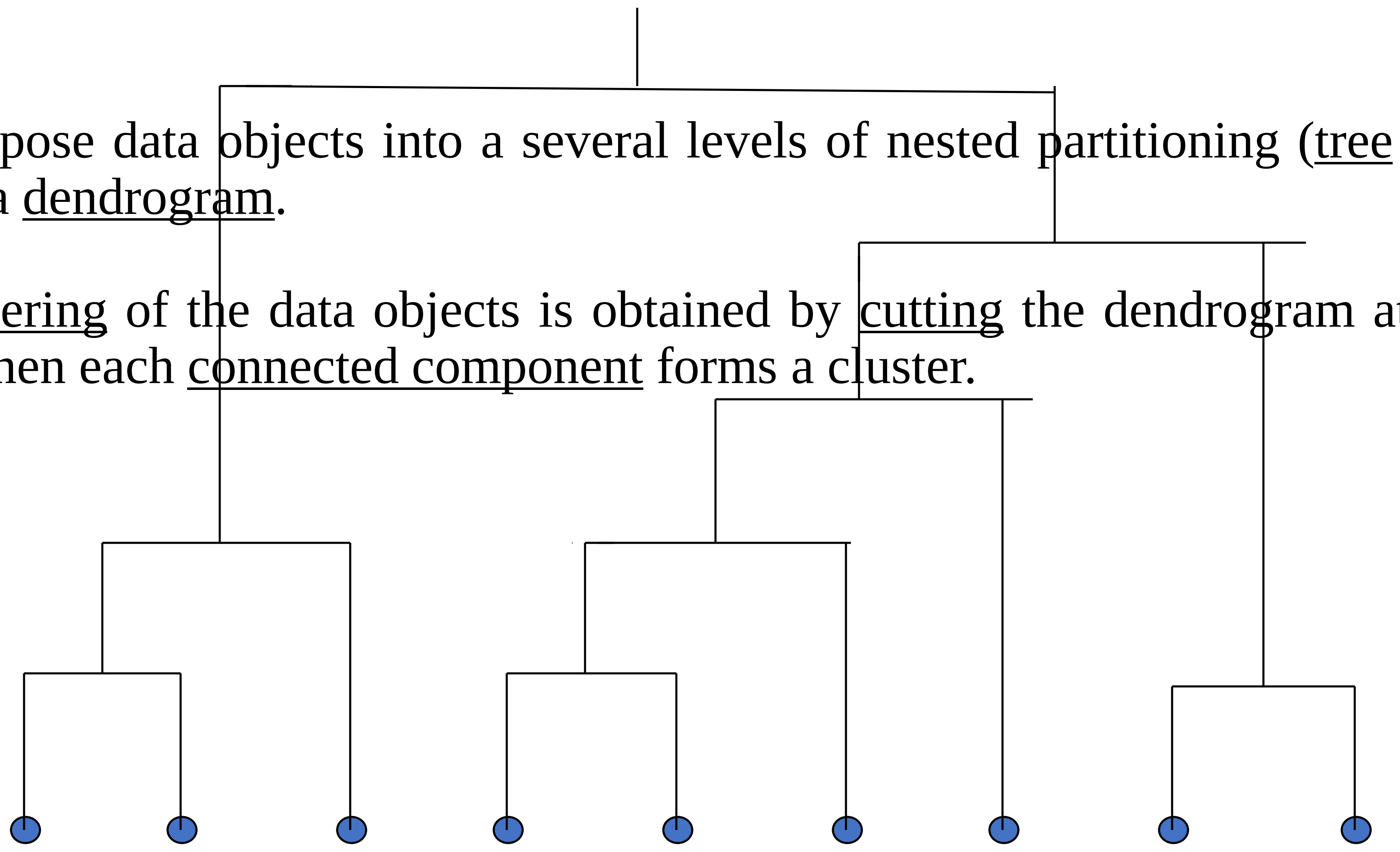
- **Consider all the data points as a single cluster and in each iteration, separate the data points from the cluster which are not similar. Each data point which is separated is considered as an individual cluster. In the end, we'll be left with n clusters.**



Hierarchical Clustering



Dendrogram: Shows How the Clusters are Merged



Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.

“Calculation of the Similarity Between Two Clusters?”

Calculating the similarity between two clusters is important to merge or divide the clusters.

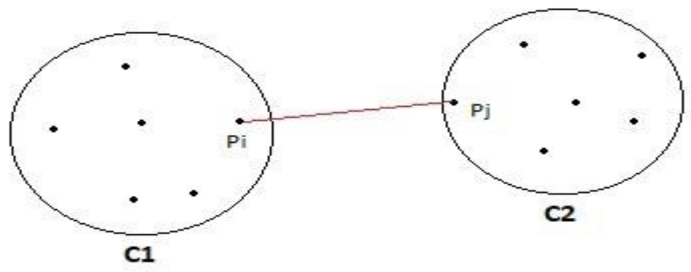
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Ward’s Method

"Calculation of the Similarity Between Two Clusters?"

- MIN

Sim(C1,C2) = Min Sim(Pi, Pj) such that Pi ∈ C1 & Pj ∈ C2.

Also known as single-linkage algorithm. The similarity of two clusters C1 and C2 is equal to the **minimum** of the similarity between points Pi and Pj such that Pi belongs to C1 and Pj belongs to C2.



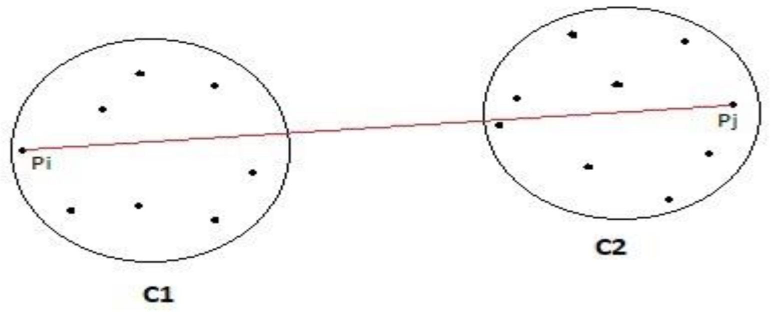
It can separate non-elliptical shapes as long as the gap between the two clusters is not small but cannot separate clusters properly if there is noise between clusters.

“Calculation of the Similarity Between Two Clusters?”

- MAX

Sim(C1,C2) = Max Sim(Pi, Pj) such that Pi ∈ C1 & Pj ∈ C2

Pick the two farthest points such that one point lies in cluster one and the other point lies in cluster 2



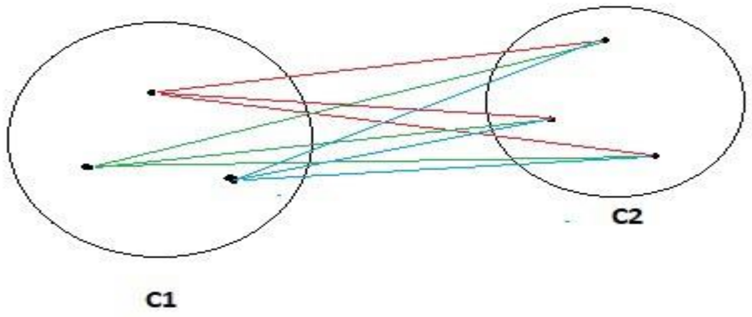
Complete Linkage; it does well in separating clusters if there is noise between clusters; but it is biased towards globular clusters, and tends to break large clusters.

"Calculation of the Similarity Between Two Clusters?"

- Group Average

sim(C1,C2) = ∑ sim(Pi, Pj)/|C1|*|C2|; where, Pi ∈ C1 & Pj ∈ C2

Take all the pairs of points and compute their similarities and calculate the average of the similarities.

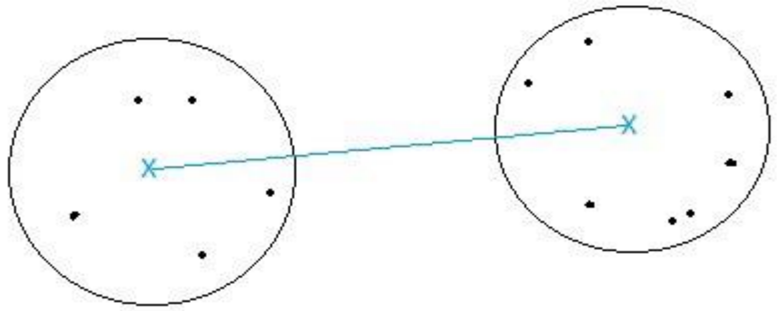


It does well in separating clusters if there is noise between clusters; but is biased towards globular clusters.

“Calculation of the Similarity Between Two Clusters?”

- Distance between centroids

Compute the centroids of two clusters C1 & C2 and take the similarity between the two centroids as the similarity between two clusters. This is a less popular technique in the real world.



“Calculation of the Similarity Between Two Clusters?”

- Ward’s Method

$\mathrm{sim}(C1,C2) = \sum (\mathrm{dist}(Pi, Pj))^2/|C1|*|C2|$

calculate the similarity between two clusters in same way as computed by Group Average except it calculates the sum of the square of the distances Pi and PJ.

It also does well in separating clusters if there is noise between clusters. But it is also biased towards globular clusters.

Space and Time Complexity of Hierarchical clustering Technique

- The space required is very high to store the similarity matrix in the RAM for large number of data points in case of Hierarchical clustering Technique.

Space complexity = $O(m^2)$ where m is the number of data points.

- To perform “m” iterations and in each iteration, we need to update the similarity matrix and restore the matrix, the time complexity is also very high.

Time complexity = $O(m^3)$ where m is the number of data points.

Limitations of Hierarchical clustering Technique

- There is no mathematical objective for Hierarchical clustering.

- All the approaches to calculate the similarity between clusters has its own disadvantages.

- High space and time complexity for Hierarchical clustering. Hence this clustering algorithm cannot be used when we have huge data.