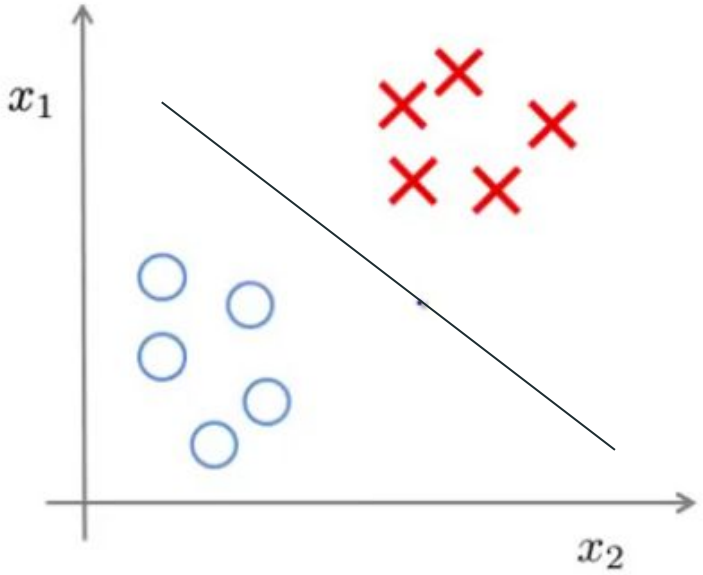
Clustering algorithms

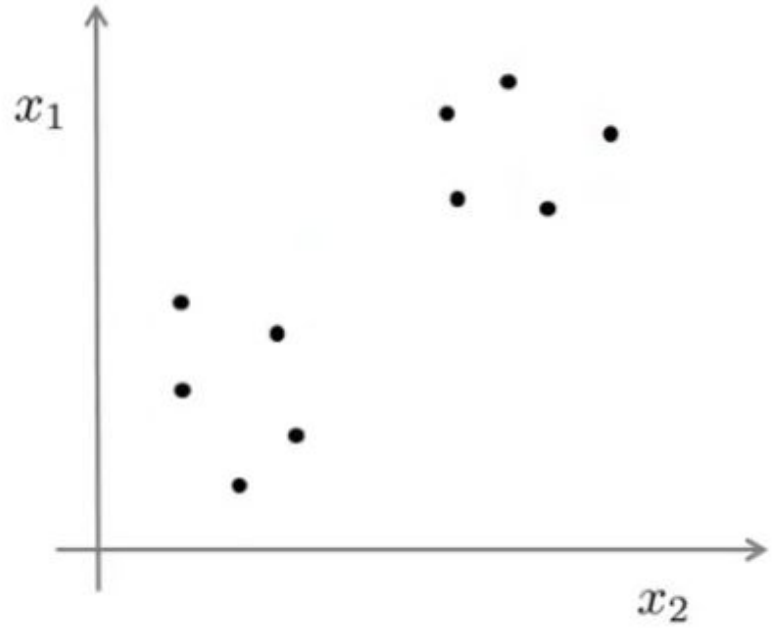
Aleksanyan Lida

Supervised learning



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

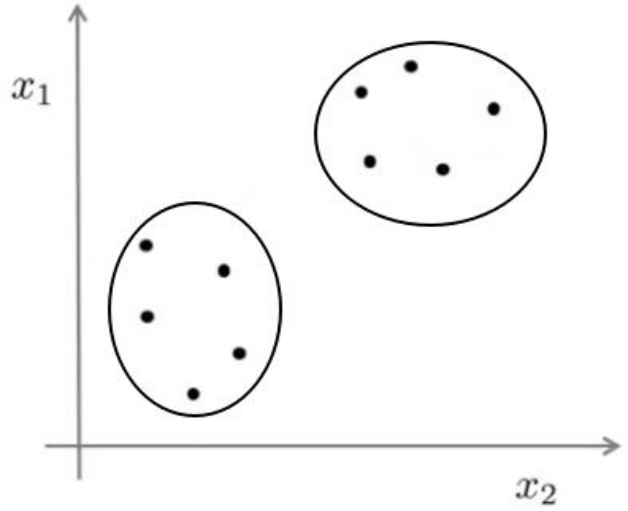
Unsupervised learning



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

Clustering algorithm

Unsupervised learning



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

We are going to divide our dataset into ***clusters***

- **Clustering** means grouping of objects based on the information found in data describing the objects or their relationship.
- The goal is that the objects in one group must be similar to each other but different from objects in another group.
- It deals with finding a structure in the collection of unlabeled data.

Clustering algorithms applications

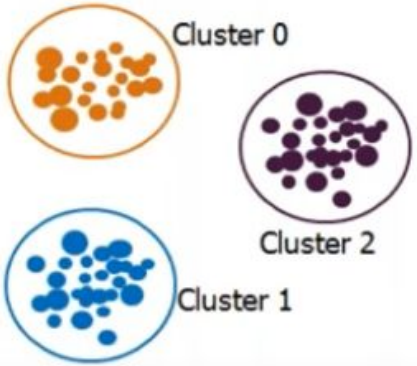
- Identifying fake news
- Spam filter
- Marketing and sales
- Social network analysis
- Pattern recognition

etc.

Types of clustering

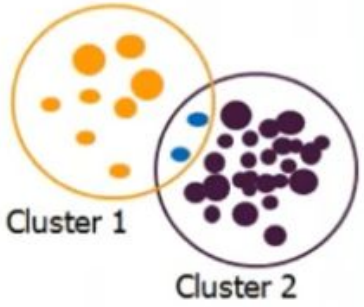
Exclusive Clustering

- An item belongs exclusively to one cluster, not several.
- K-means does this sort of exclusive clustering.



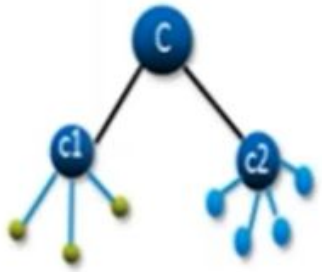
Overlapping Clustering

- An item can belong to multiple clusters
- Its degree of association with each cluster is known
- Fuzzy/ C-means does this sort of exclusive clustering.

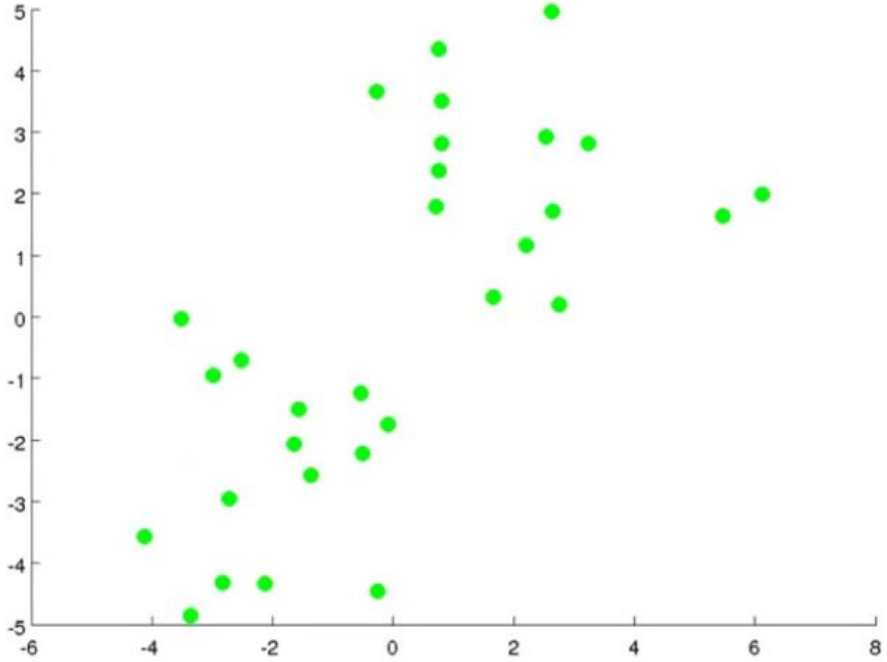


Hierarchial Clustering

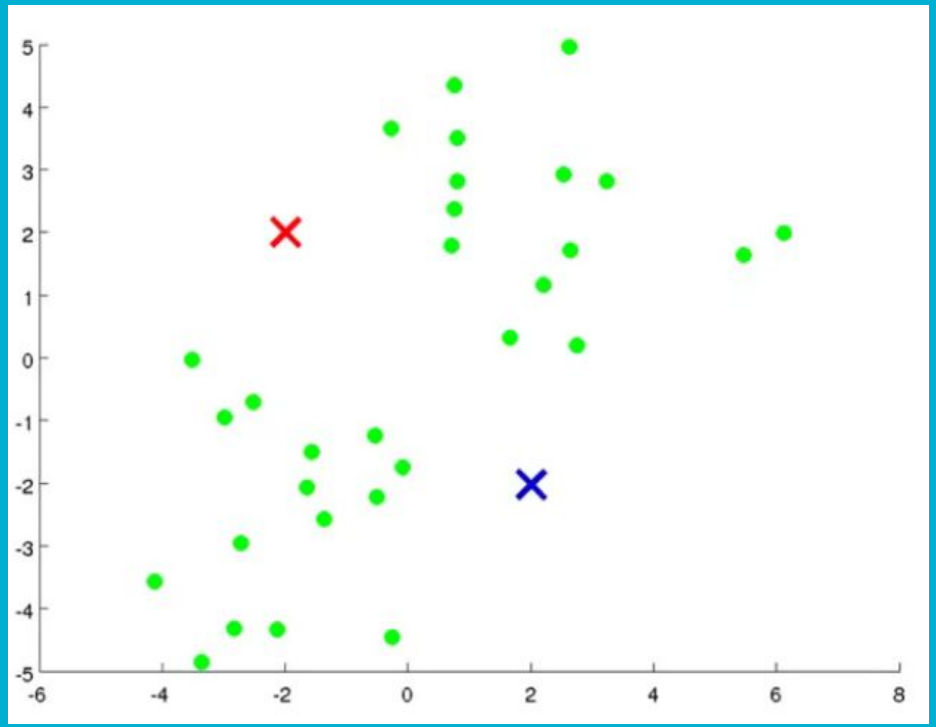
- When two cluster have a parent-child relationship or a tree-like structure then it is Hierarchical clustering



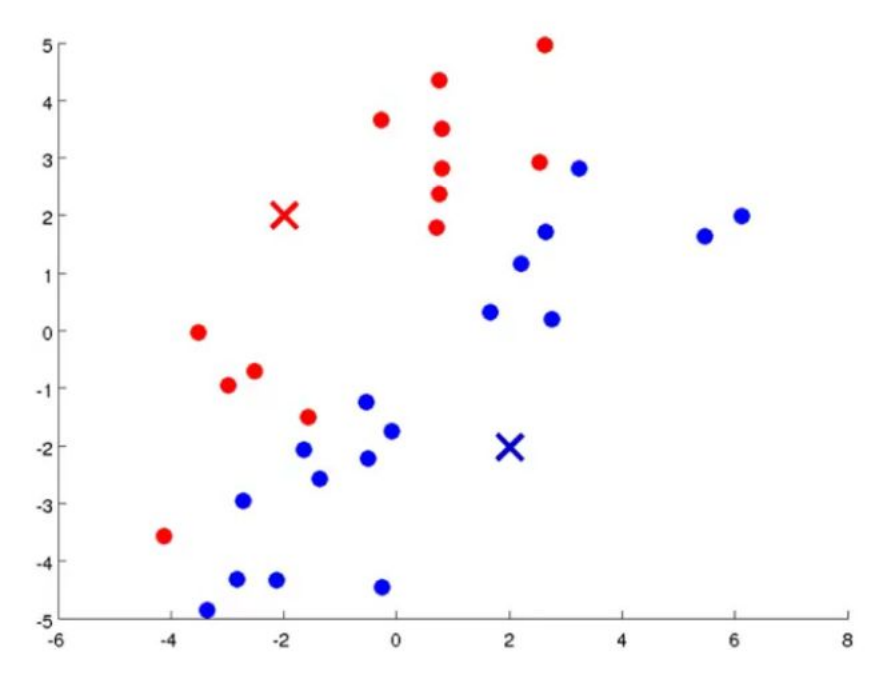
K-Means algorithm



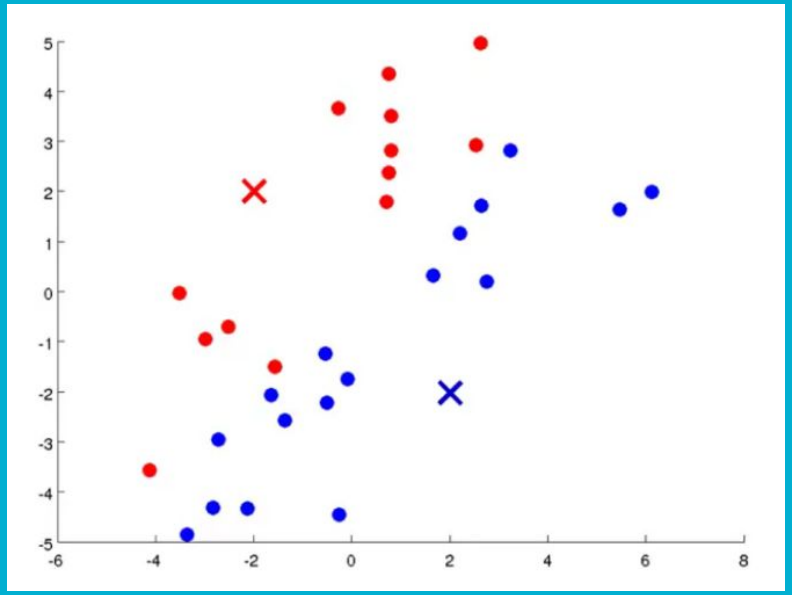
The first step is to randomly initialize two points, called the *cluster centroids*

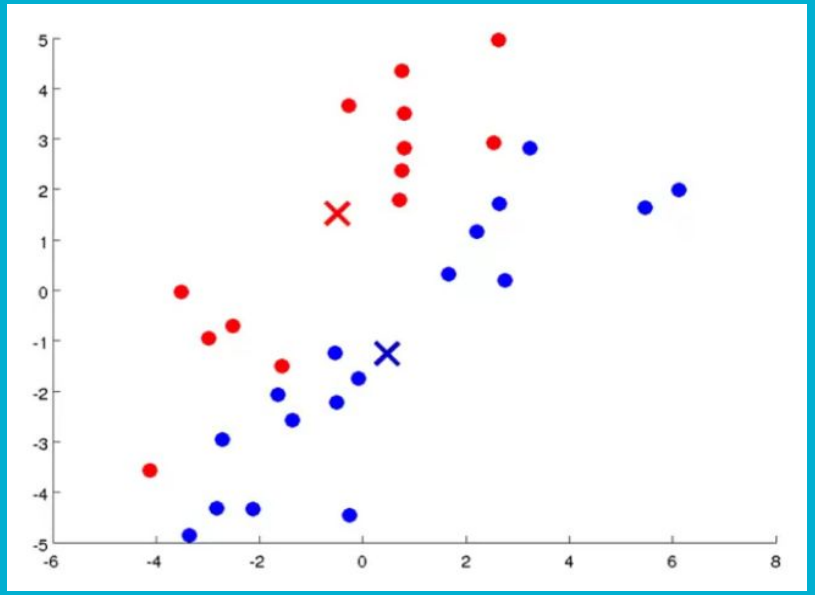


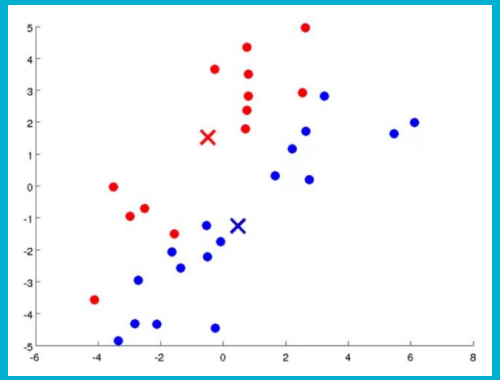
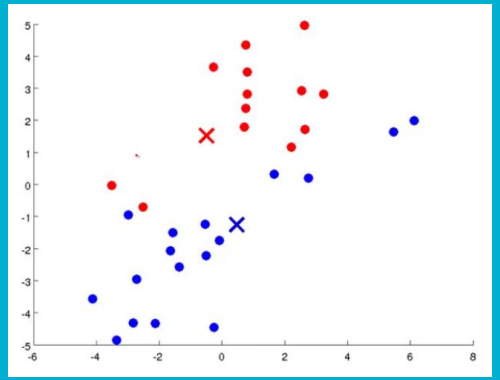
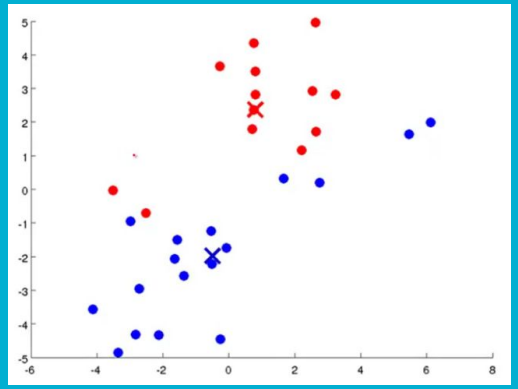
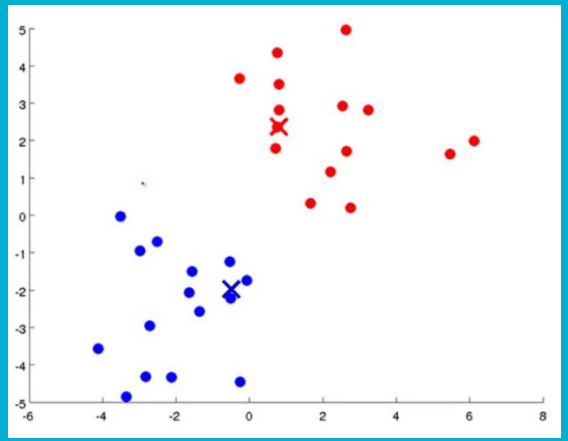
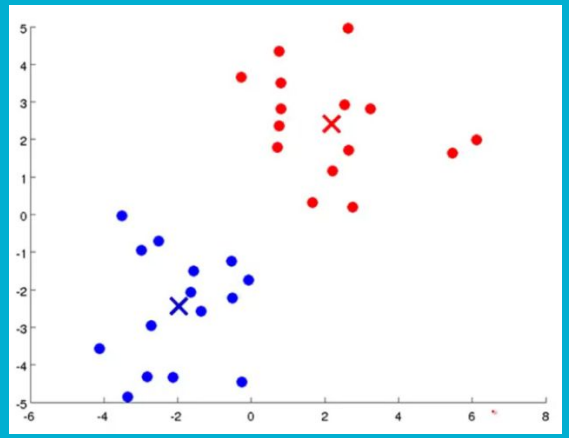
Cluster assignment step



Move centroid step





K-means algorithm

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

μ_k := average (mean) of points assigned to cluster k

}

- The first for loop is for cluster assignment step.
- The second one is for move centroid step.

Pseudo code for K-Means algorithm

Step 1. Initialise the function centres

Set the initial function centres to the first m training data or to the m randomly chosen training data.

Step 2. Group all patterns with the closet function centre

For each pattern x_i, assign x_i to group j*, where $\left\| x_i - c_{j^*} \right\| = \min_j \left\| x_i - c_j \right\|^2$

Step 3. Compute the sample mean for the function centre

For each group c_j, $c_j = \frac{1}{m_j} \sum_{x_i \in group\ j} x_i$

where m_j is the number of patterns in group j.

Step 4. Repeat by going to step 2, until no change in cluster assignments

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:

Cost function

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

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

K-means algorithm

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

Repeat {

cluster assignment step

for i = 1 to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

move centroid step

for k = 1 to K

μ_k := average (mean) of points assigned to cluster k

}

In the **cluster assignment step**, our goal is to:

Minimize J(...) with $c^{(1)}, \ldots, c^{(m)}$ (holding $\mu_1, \ldots, \mu_K$ fixed)

In the **move centroid** step, our goal is to:

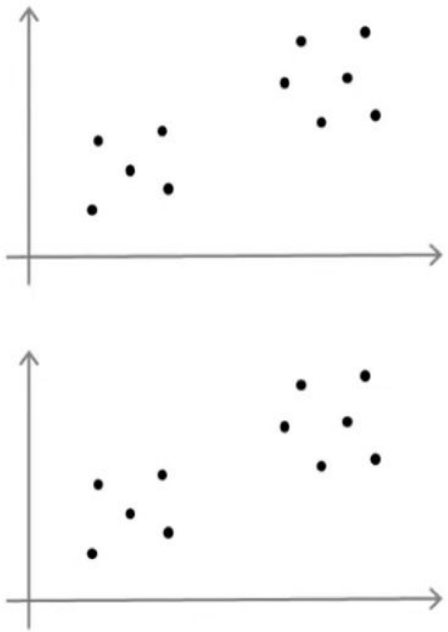
Minimize J(...) with $\mu_1, \ldots, \mu_K$

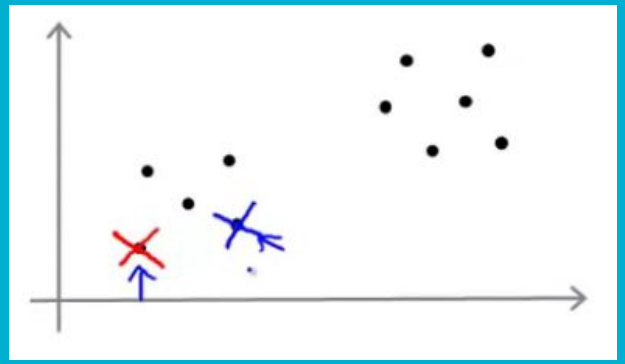
Random initialization

Should have $K < m$

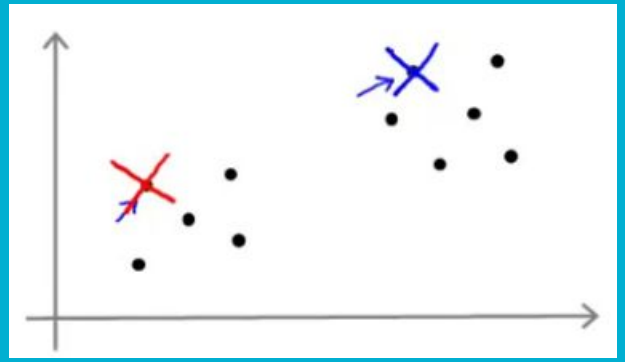
Randomly pick K training examples.

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



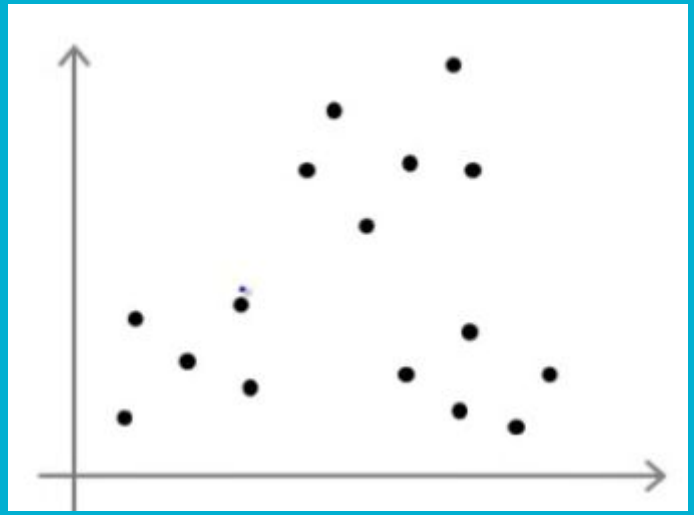


or



Local optima

out training data



Random initialization

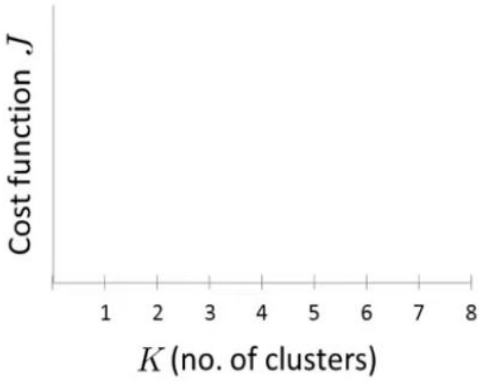
For i = 1 to 100 {

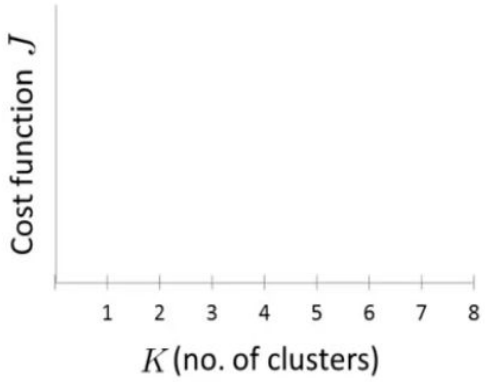
Randomly initialize K-means.
Run K-means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$.
Compute cost function (distortion)
$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$
}

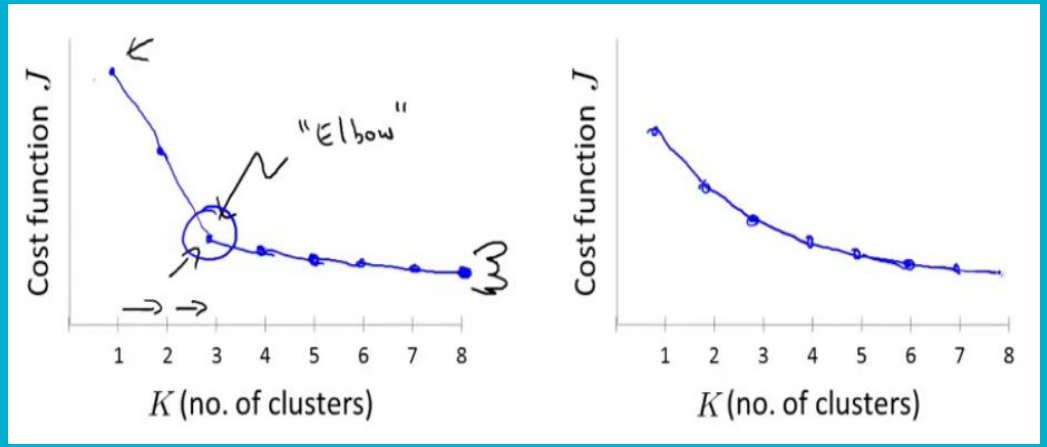
Pick clustering that gave lowest cost $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

Choosing the value of K

Elbow method:







FOREL algorithm

- FOREL- FORmal ELement
- FOREL- clustering algorithm based on the principle of joining in one cluster those data points which are more closely situated.

$$P_n = \sum_{k=1}^{L} s_{nk}$$

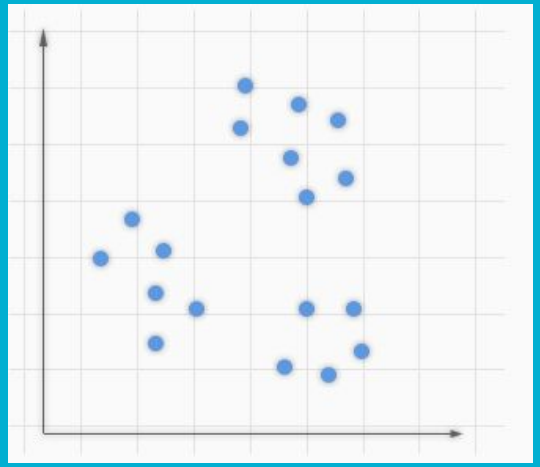
s_{nk} is the distance between n's cluster's centre and point k

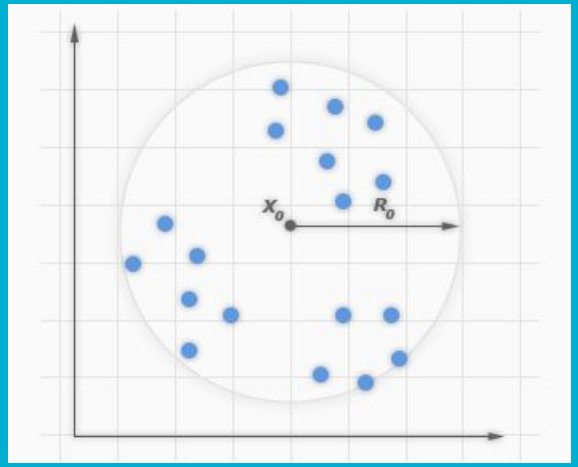
P_n is the distance between n's cluster and all points in it

$$P = \sum_{n=1}^{N} P_n$$

So the goal of FOREL algorithm is finding such set of clusters which will minimize P.

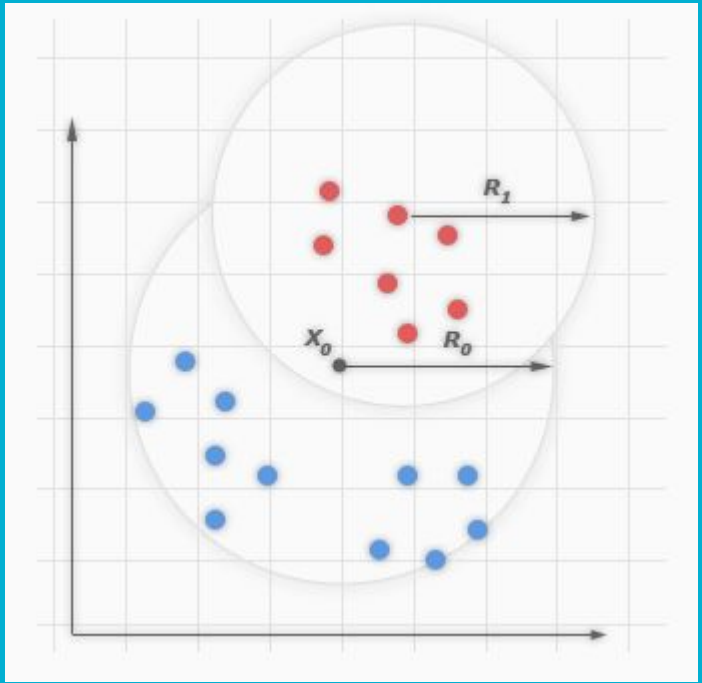
Algorithm work example



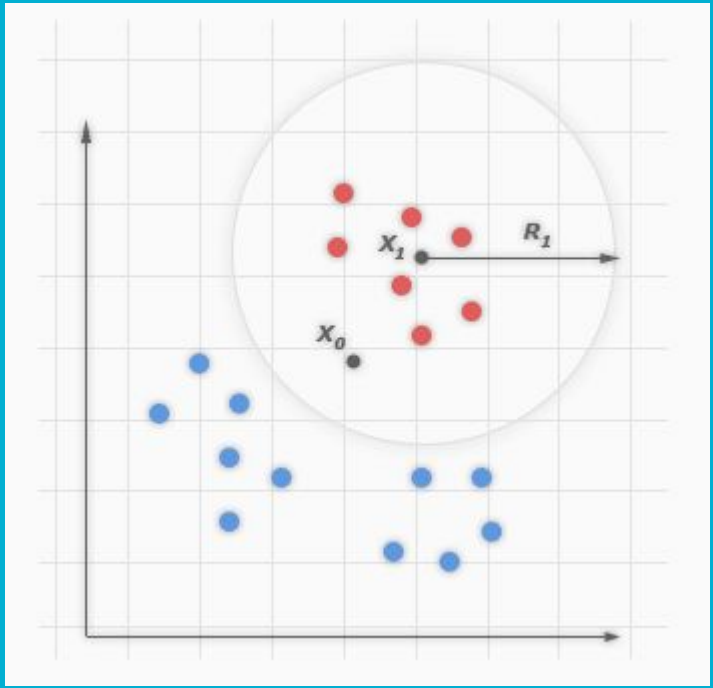


Step1. We are choosing some radius R0, so that many of our data points will be in the sphere of radius R0.

Step2. Then we are changing radius R0 to R1: R1 = 0.9R0, and moving the centre of the sphere to any point which is in the sphere with radius R1.



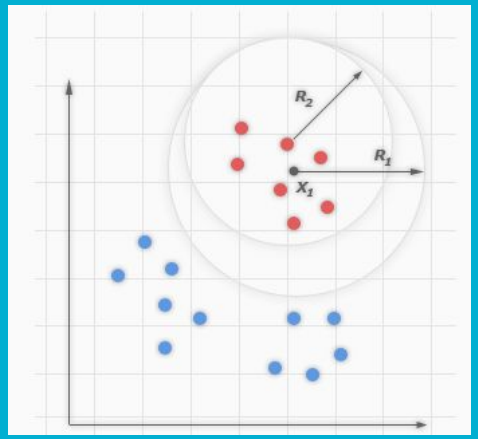
Step3. Now we have to calculate new centre point (mean of the points) and move there our centroid.

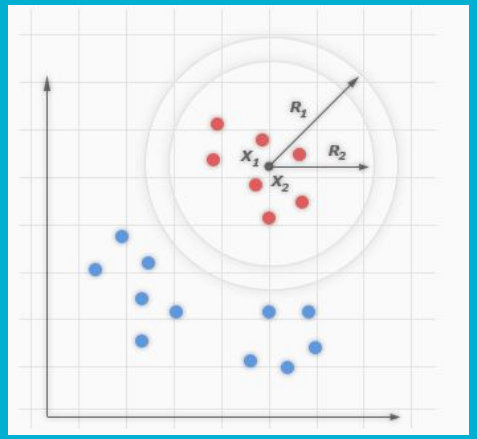


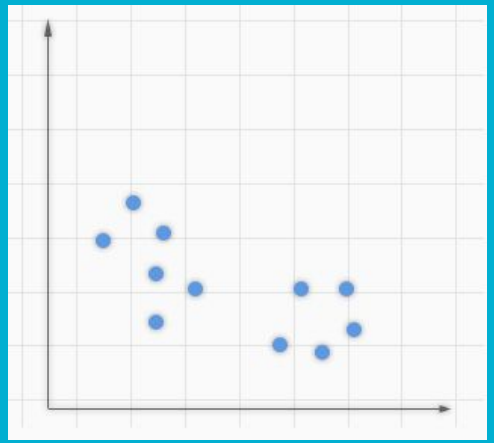
Step4. If the new centroid is different than previous one we must go to step 2 and repeat the loop. The loop will continue until the centre stabilizes.

Step5. After changing the centre point we see that new centre X2=X1, so we found one cluster, now we delete points of that cluster.

Step6. Continue from step 1 for remaining points.







One of the main advantages: *FOREL algorithm founds number of clusters itself!*
One of the main disadvantages: *we must choose appropriate radius experimentally, or find some algorithm which will do it.*

Fuzzy K-Means Clustering

- In fuzzy clustering, each point has a probability of belonging to each cluster, rather than completely belonging to just one cluster as it is the case in the traditional k-means.
- Fuzzy k-means specifically tries to deal with the problem where points are somewhat in between centers or otherwise ambiguous by replacing distance with probability.
- Fuzzy k-means uses a weighted centroid based on those probabilities.
- Processes of initialization, iteration, and termination are the same as the ones used in k-means.
- The resulting clusters are best analyzed as probabilistic distributions rather than a hard assignment of labels.

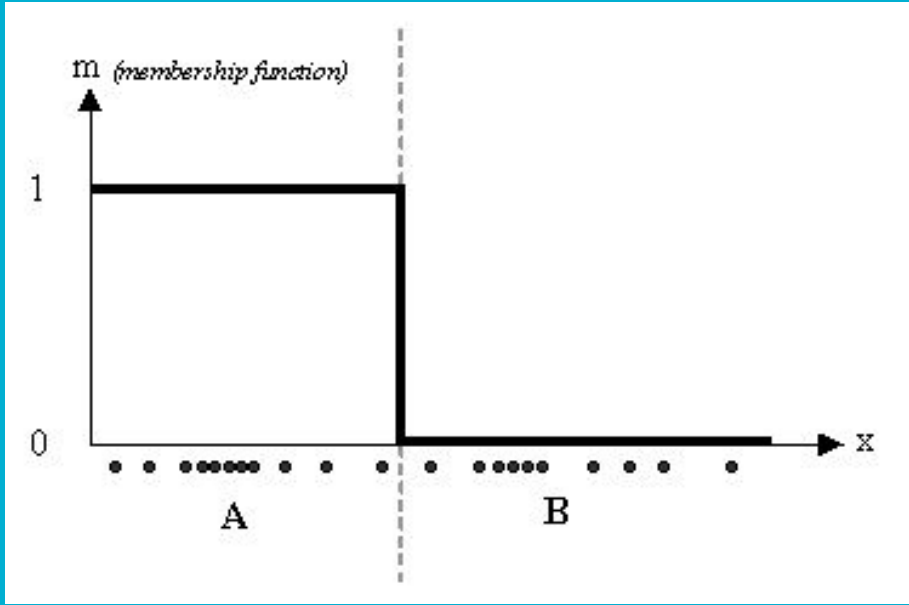
The fuzzy k-means algorithm is the following:

- **Assume** a fixed number of clusters *K*.
- **Initialization:** Randomly initialize the k-means μk associated with the clusters and compute the probability that each data point *Xi* is a member of a given cluster *K*, *P(PointXiHasLabelK|Xi,K)*.
- **Iteration:** Recalculate the centroid of the cluster as the weighted centroid given the probabilities of membership of all data points *Xi* :

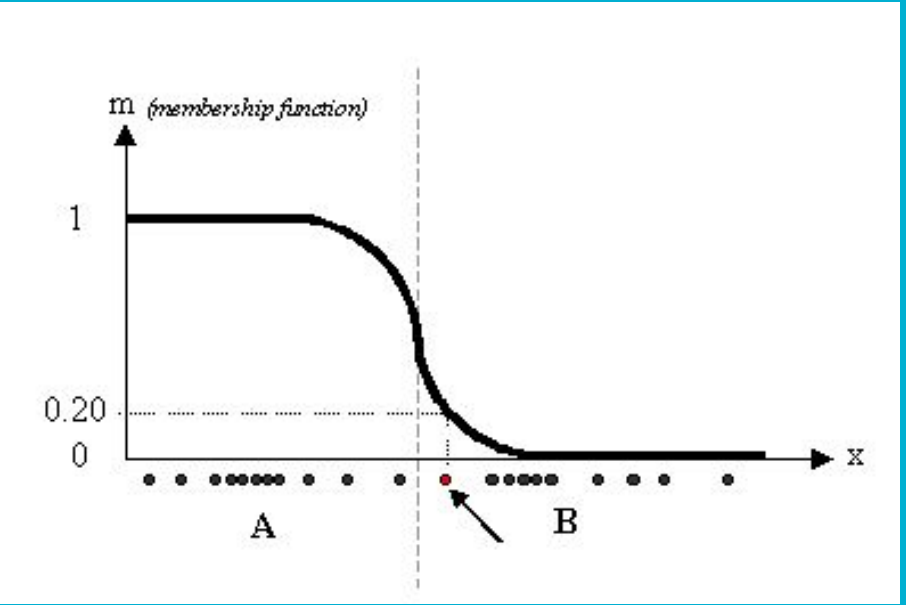
$$\mu_k(n+1) = \frac{\sum_{x_i \in k} x_i \times P(\mu_k|x_i)^b}{\sum_{x_i \in k} P(\mu_k|x_i)^b}$$

- **Termination**: Iterate until convergence or until a user-specified number of iterations has been reached (the iteration may be trapped at some local maxima or minima)

K-Means



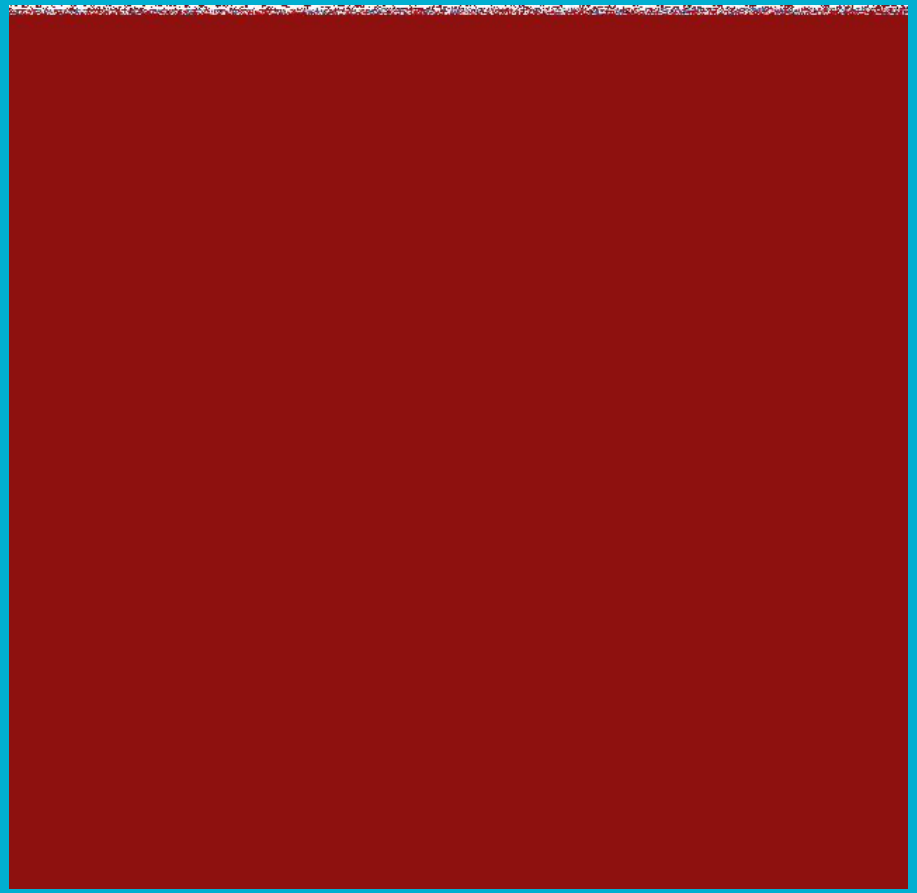
Fuzzy K-Means



Hierarchical Clustering Algorithms

Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering is this:

- Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
- Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
- Compute distances (similarities) between the new cluster and each of the old clusters.
- Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.



Linkage criteria

- Single-linkage (two most similar parts of a cluster)
- Complete-linkage (two least similar parts of a cluster)
- Mean or average-linkage (the center of the clusters)

Thank you!