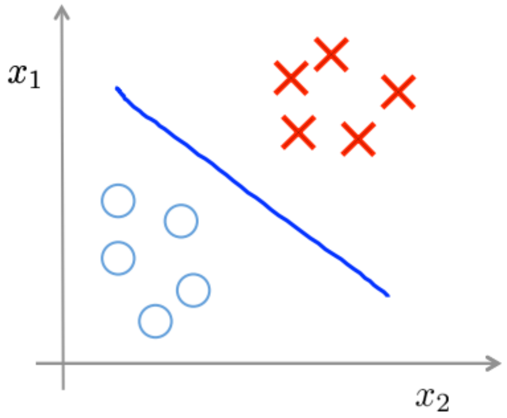
Week 8:
Unsupervised Learning
Clustering

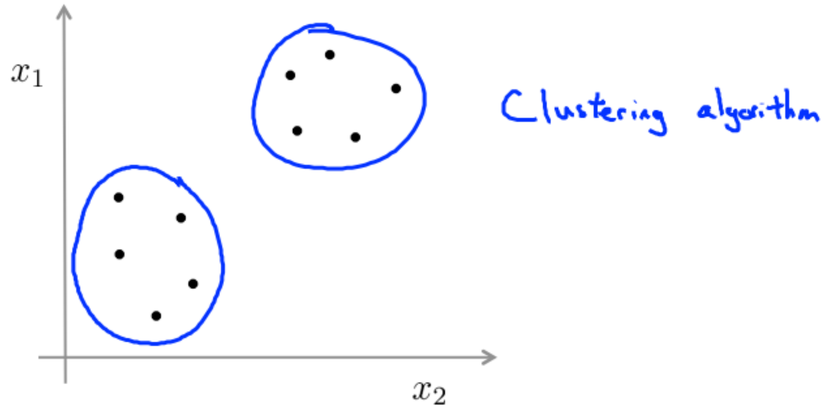
Difference between Supervised learning and Unsupervised learning

Supervised learning



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

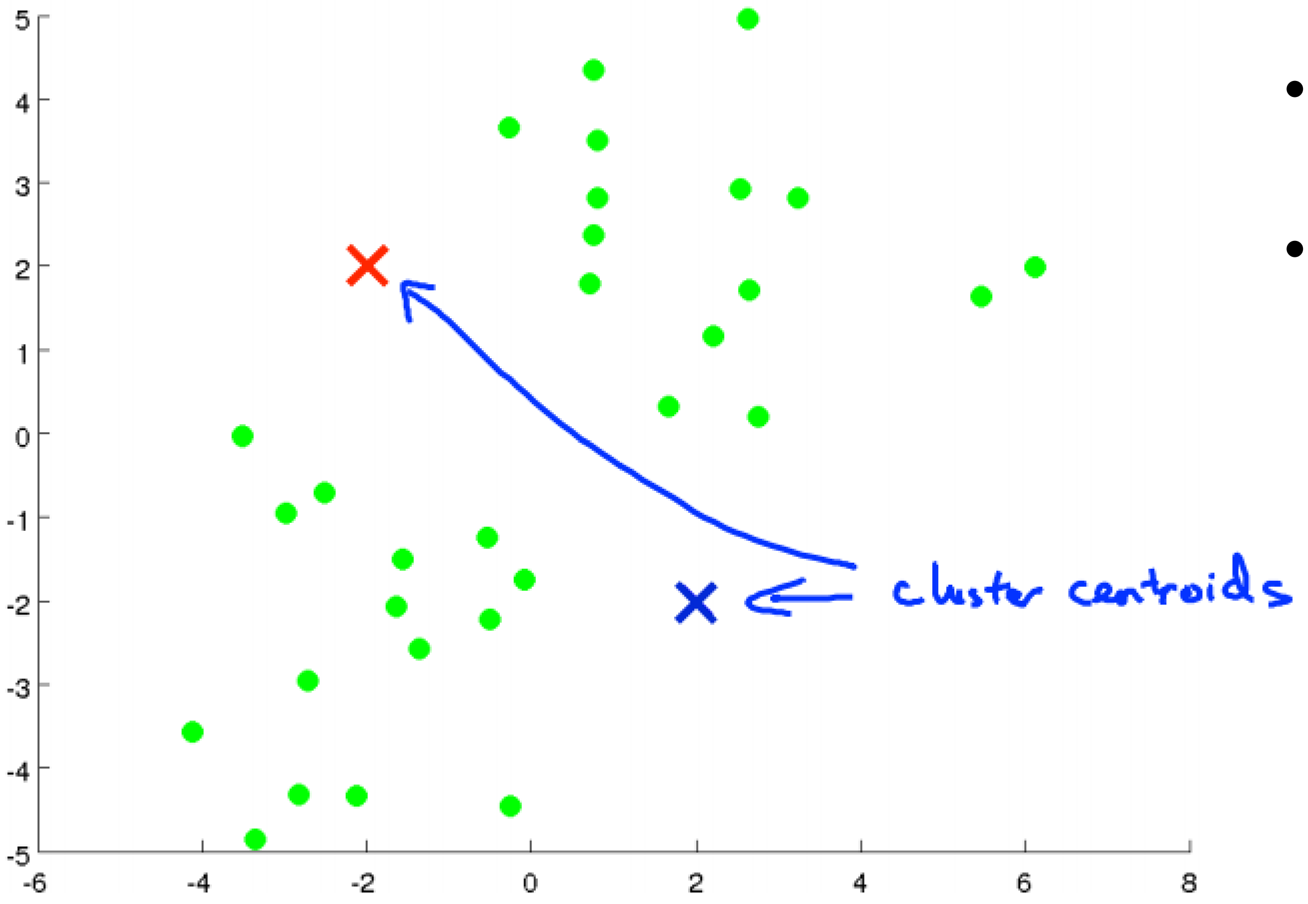
Unsupervised learning



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

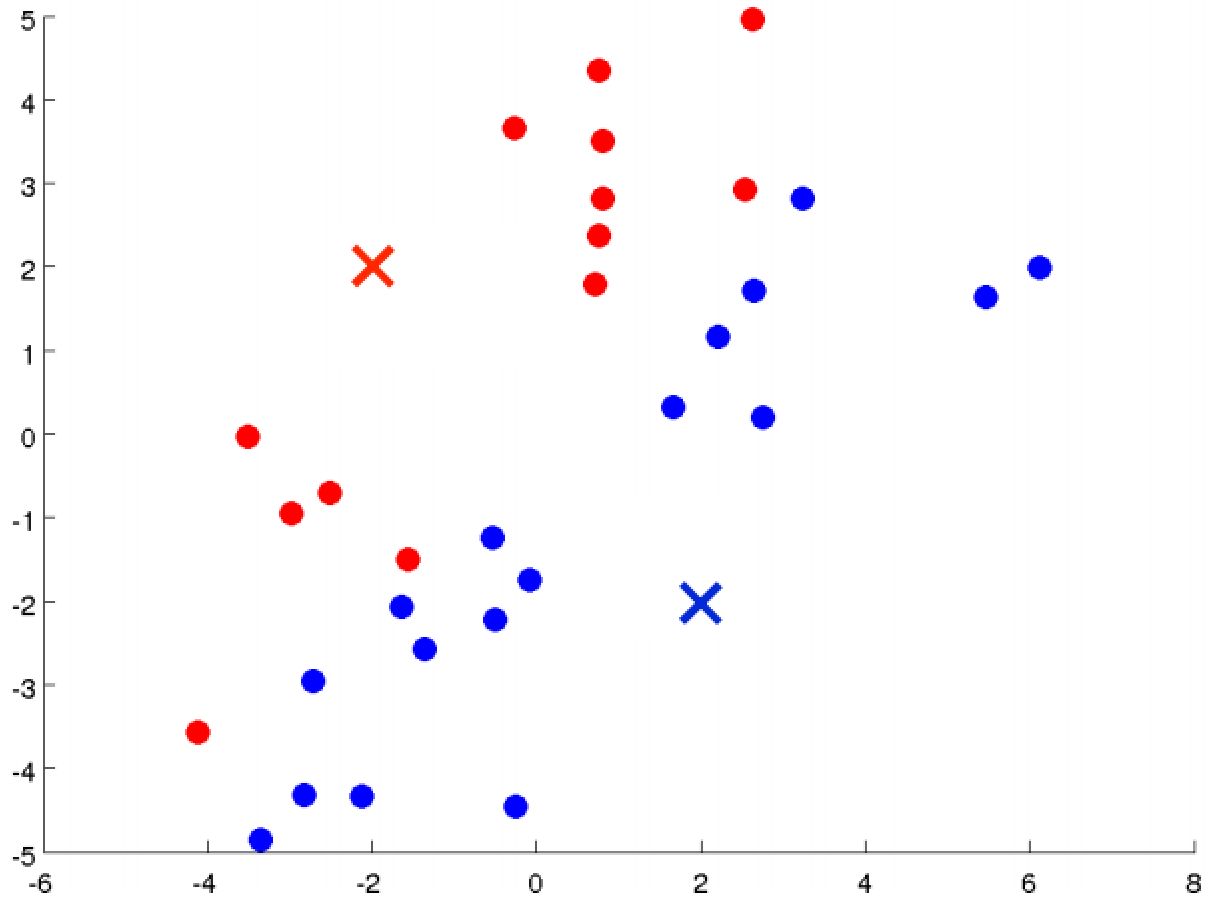
No label, Only X is given

K-means algorithm



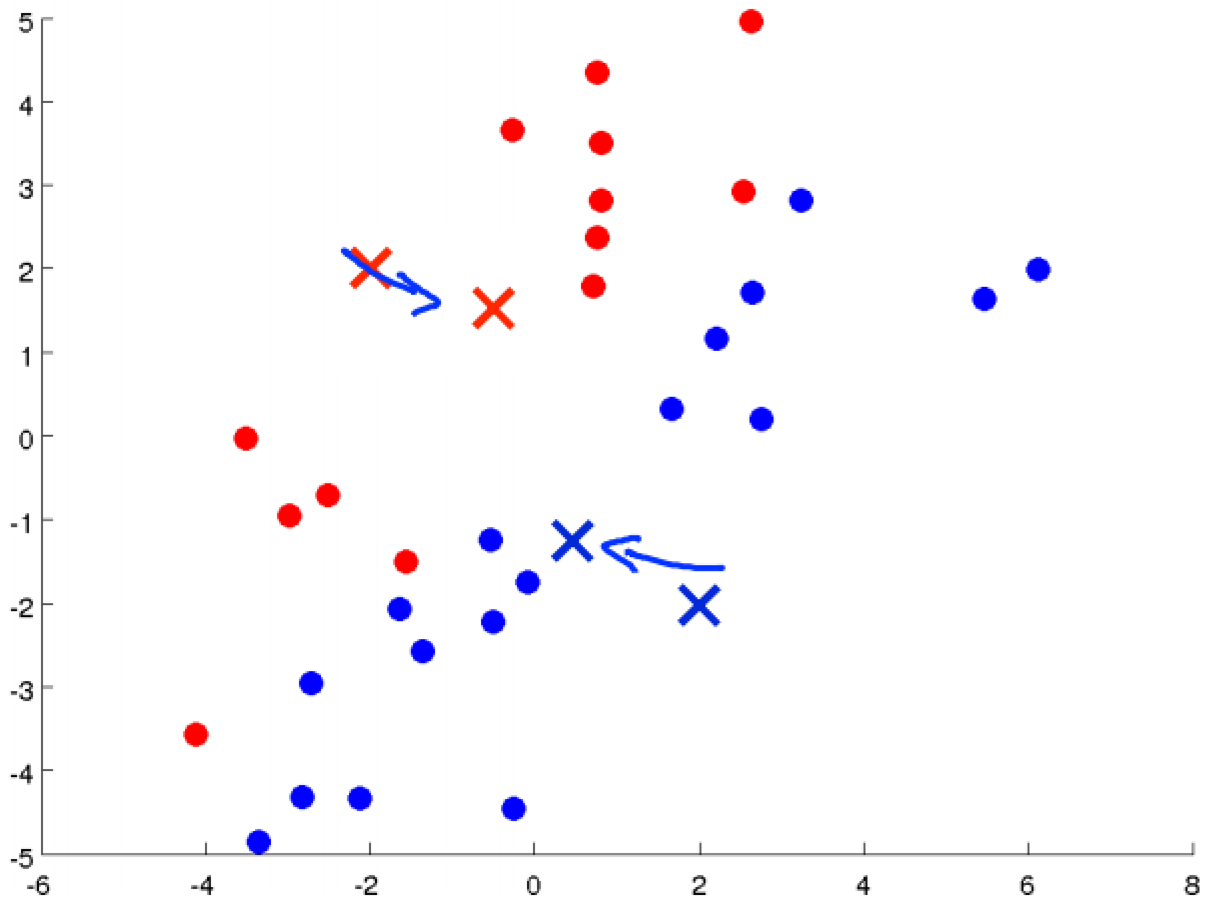
- Randomly select cluster centroids
- Goal is to locate centroids to center of each cluster

K-means algorithm

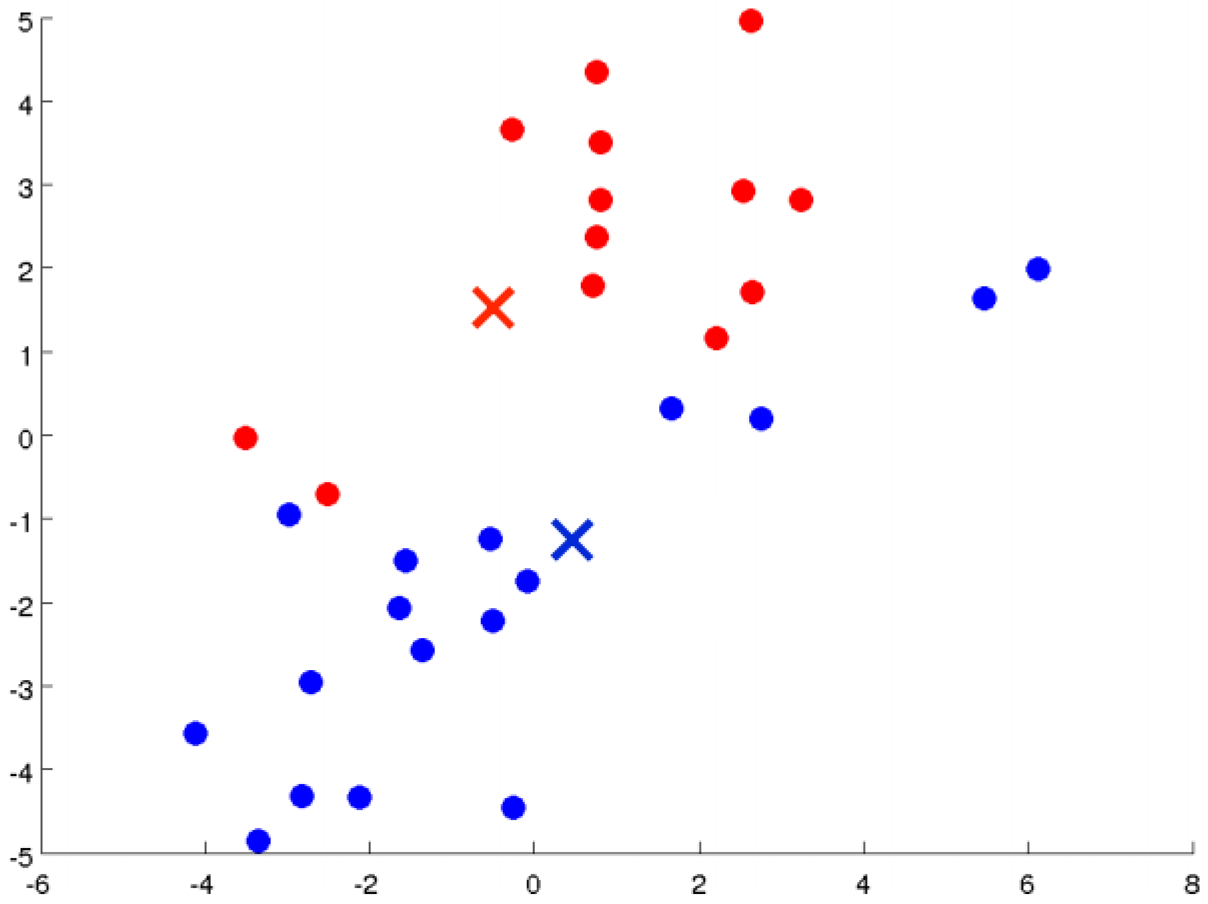


- Calculate distance between every training data with centroids, and pick closer centroid
- Get mean value of each clusters and move centroid to mean value

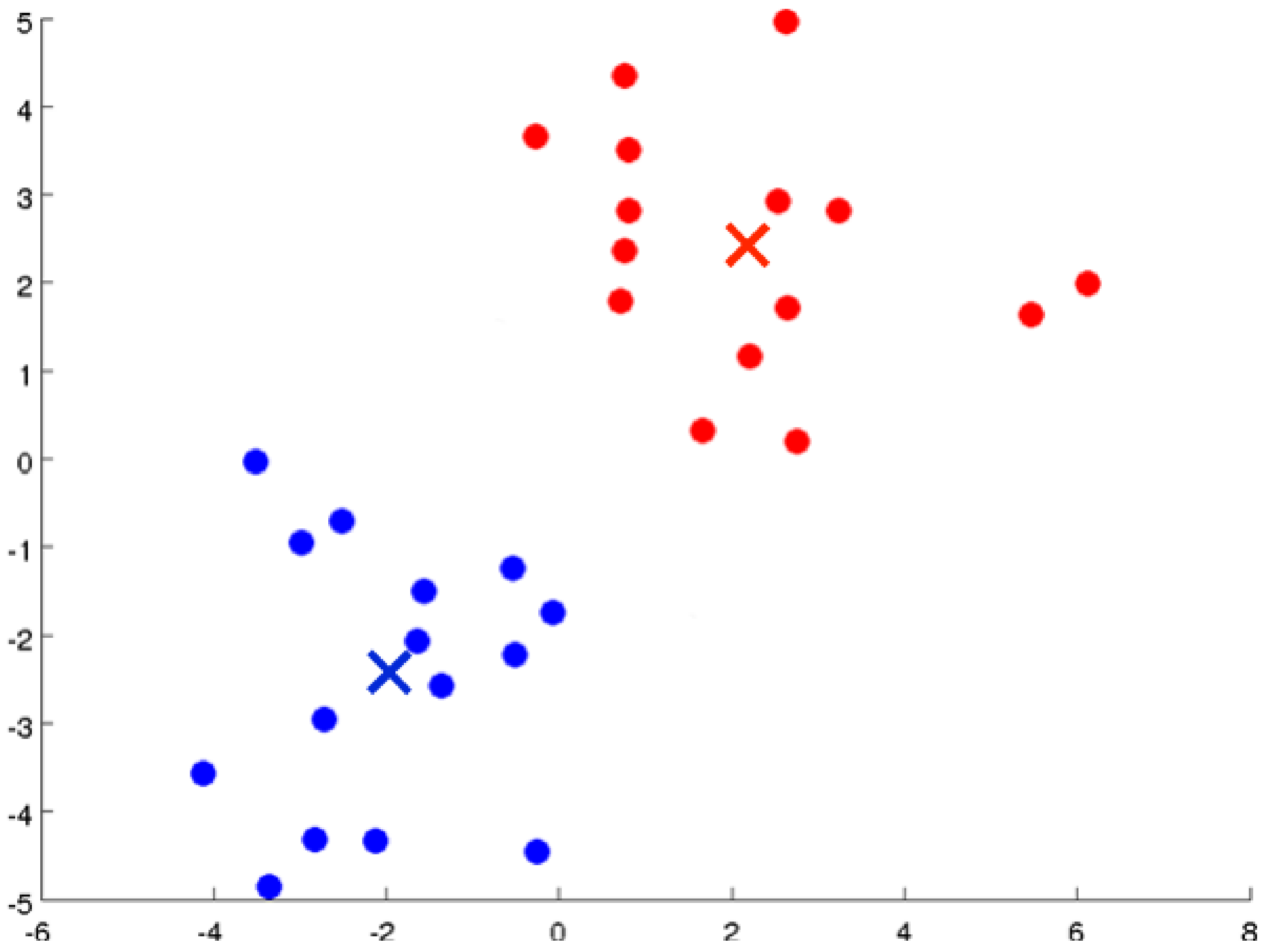
K-means algorithm



K-means algorithm



K-means algorithm



- After iterating, Cluster centroids converge to certain value

How k-means algorithm works

K-means algorithm

$\mu_1 \times \quad \mu_2 \times$

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 $x^{(i)}$

$\min_k \|x^{(i)} - \mu_k\|^2 \rightarrow c^{(i)}$

Move centroid

for k = 1 to K

$\rightarrow \mu_k$:= average (mean) of points assigned to cluster k

$x^{(1)}, x^{(5)}, x^{(6)}, x^{(10)} \rightarrow c^{(1)}=2, \; c^{(5)}=2, \; c^{(6)}=2, \; c^{(10)}=2$

$$\mu_2 = \frac{1}{4}\left[x^{(1)} + x^{(5)} + x^{(6)} + x^{(10)}\right] \in \mathbb{R}^n$$

}

- Iterate these two steps until centroids converge

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 ($\mu_k \in \mathbb{R}^n$) K $k \in \{1, 2, \ldots, K\}$

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned $x^{(i)} \to 5$ $c^{(i)} = 5$ $\mu_{c^{(i)}} = \mu_5$

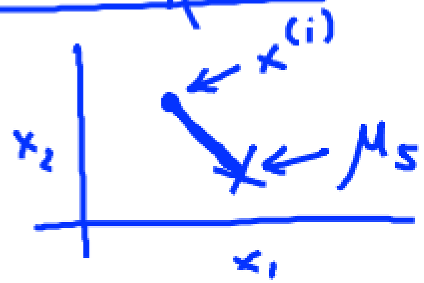
Optimization objective:

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

Difference between input data x and current centroid of x

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

Distortion



Like other algorithms, goal is to minimize cost function(distortion function)

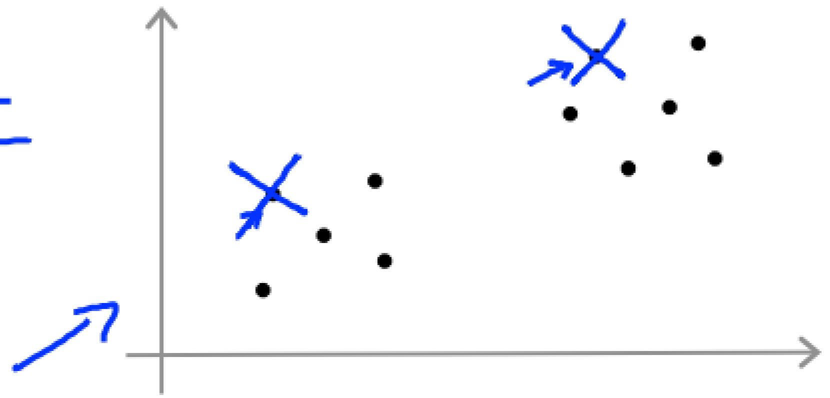
Selecting 'K', number of cluster centroids

Random initialization

$K=2$

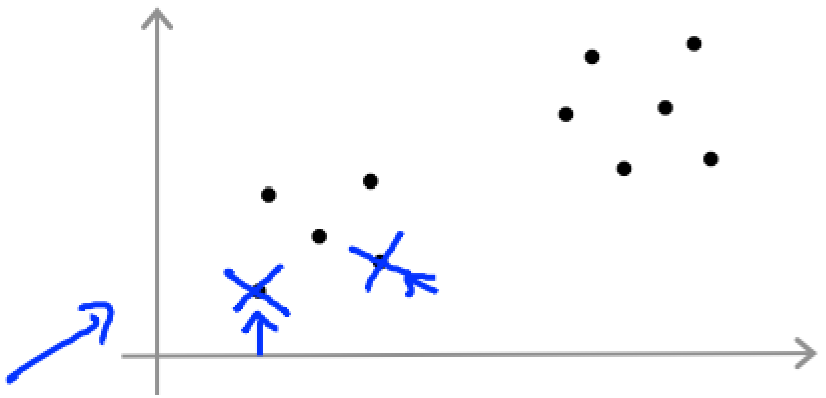
Should have $K < m$

Randomly pick K training examples.



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

$\mu_1 = x^{(i)}$
$\mu_2 = x^{(j)}$
$\vdots$



K should be less than data m
If randomly picked wrong examples, can result Local optima problem

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)

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

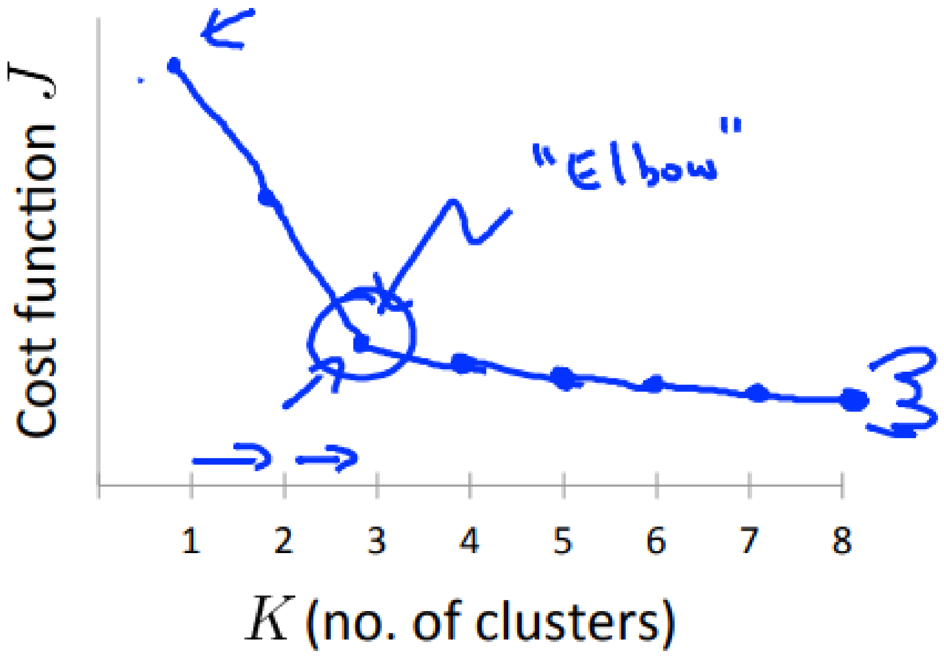
}

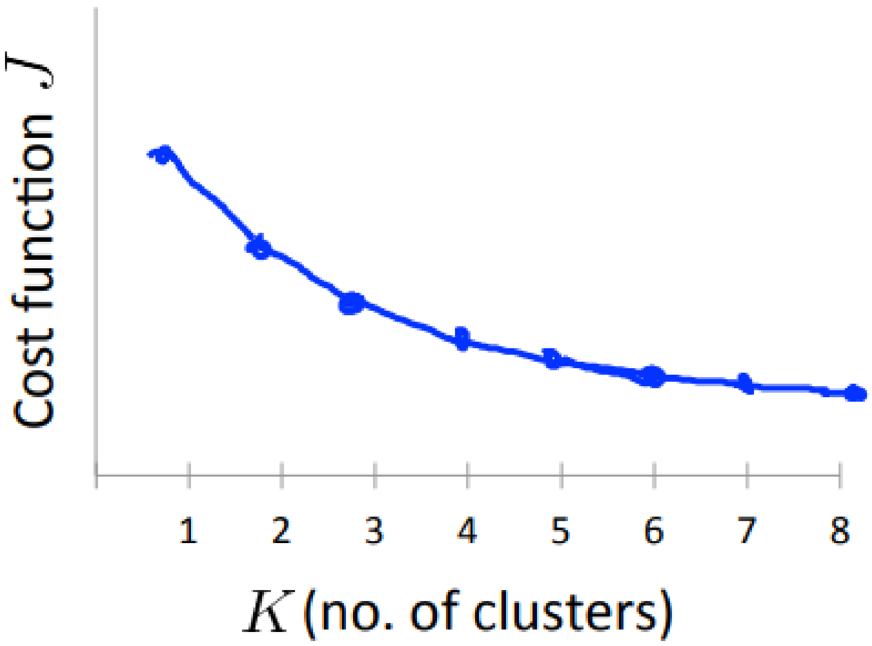
Pick clustering that gave lowest cost $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

Random initialize iteratively, pick lowest cost

Choosing the number of clusters

Elbow method:





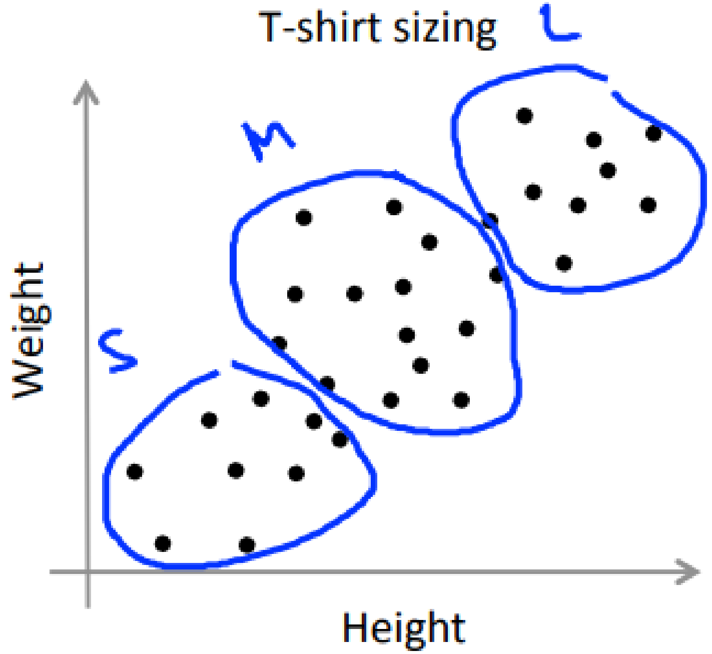
- Increase K one by one starting from 1
- No cost reduction after specific K (elbow point), then pick K
- Doesn't work if no elbow point

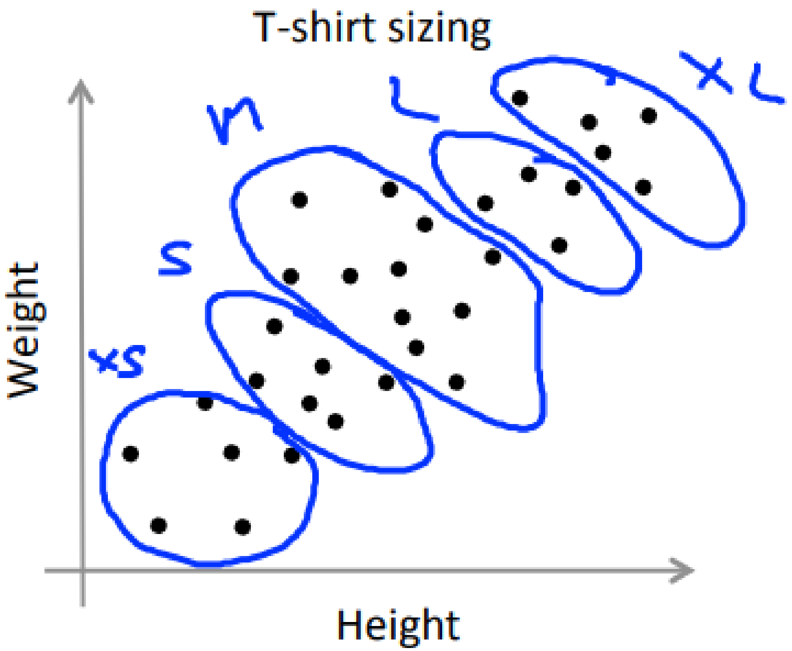
K-means for later purpose

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.

K=3 S, M, L K=5 XS, S, M, L, XL

E.g.





Pick K intuitively that well suits the clustering purpose