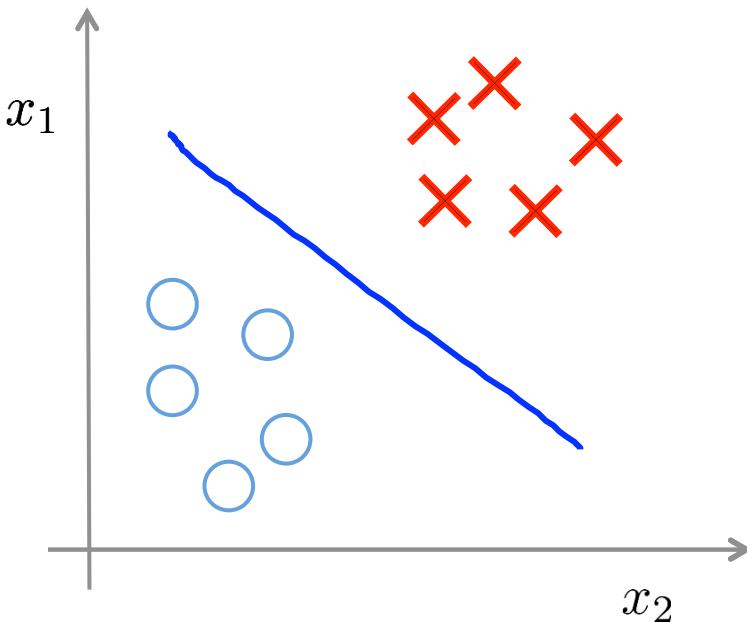


Machine Learning

Clustering

Unsupervised learning introduction

Supervised learning



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



Unsupervised learning

Unsupervised learning application

80%

1. For which of the following tasks might K-means clustering be a suitable algorithm? Select all that apply.

Given a database of information about your users, automatically group them into different market segments

Correct
You can use K-means to cluster the database entries, and each cluster will correspond to a different market segment.

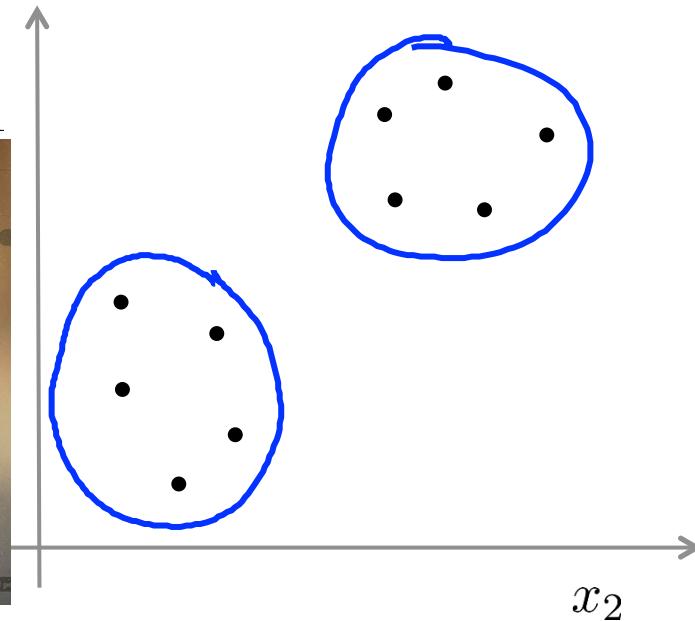
Given sales data from a large number of products in a supermarket, figure out which products tend to form coherent groups (say are frequently purchased together) and thus should be put on the same shelf.

Correct
If you cluster the sales data with K-means, each cluster should correspond to coherent groups of items.

Given historical weather records, predict the amount of rainfall tomorrow (this would be a real-valued output)

Given sales data from a large number of products in a supermarket, estimate future sales for each of these products.

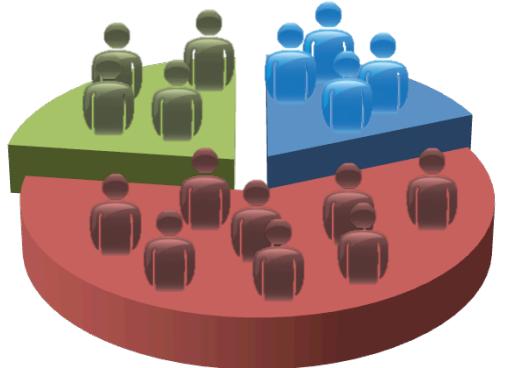
Suppose we have three cluster centroids $\mu_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} -3 \\ 0 \end{bmatrix}$ and $\mu_3 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$. Furthermore, we have a training



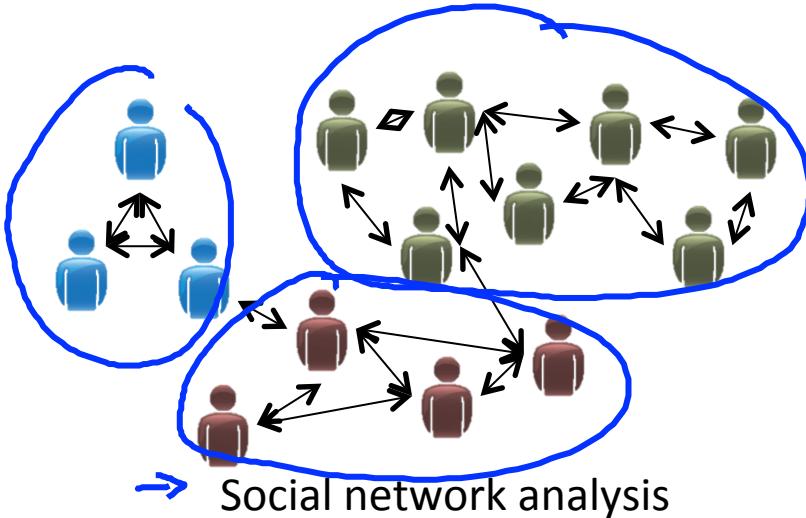
Clustering algorithm

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

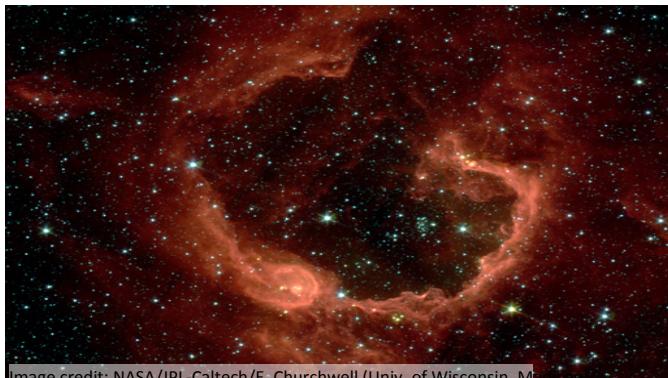
Applications of clustering



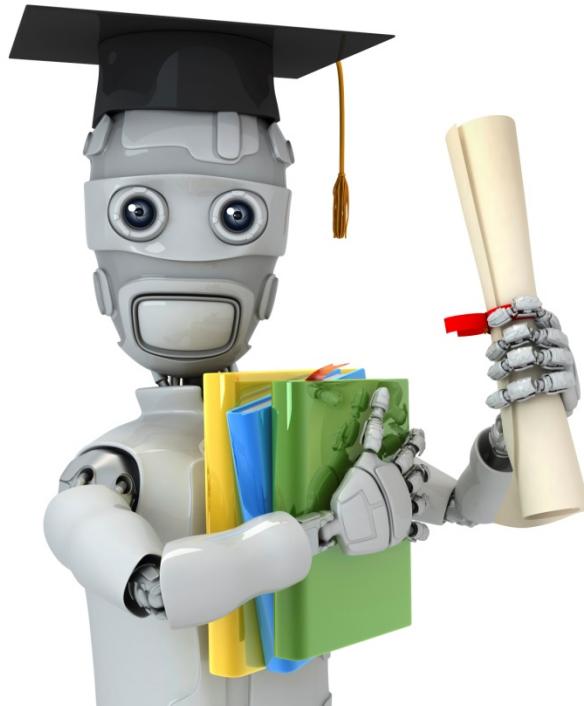
→ Market segmentation



Organize computing clusters



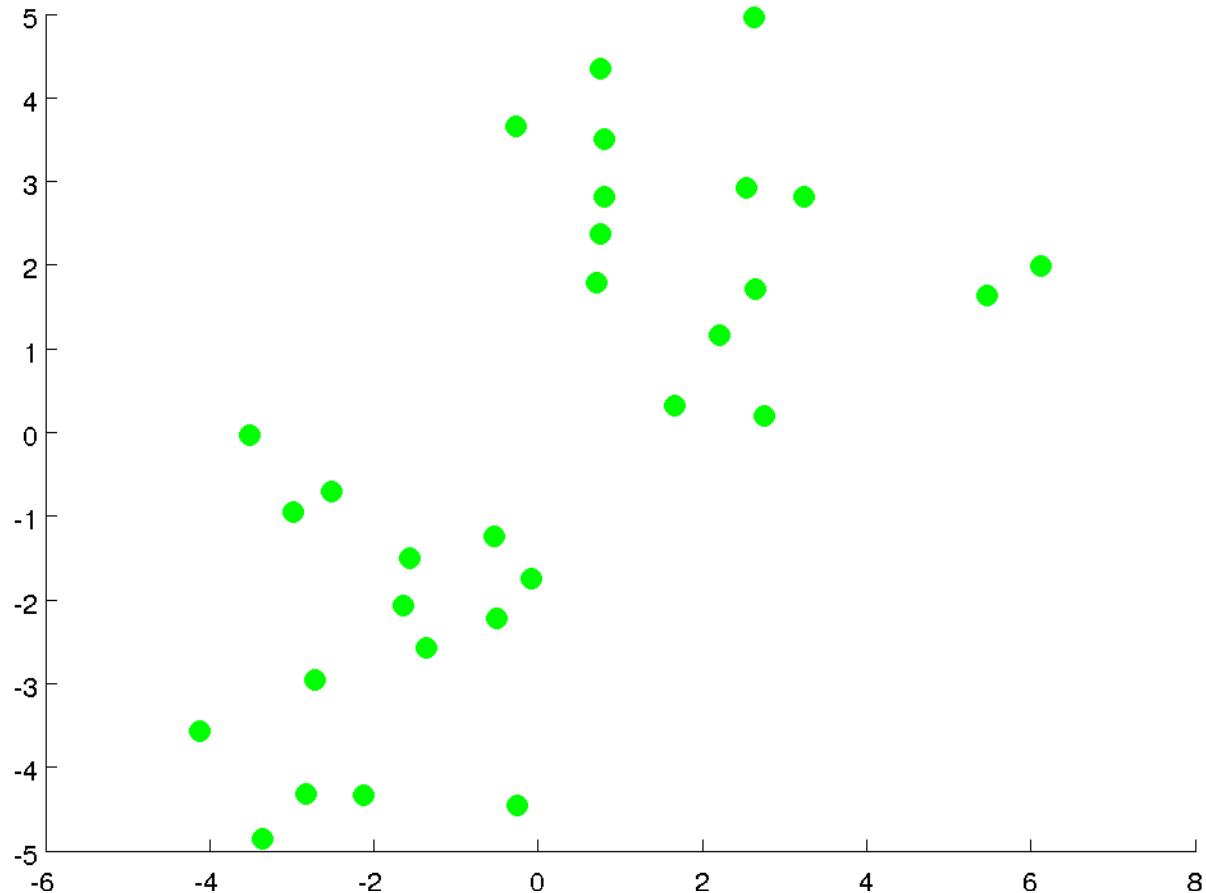
→ Astronomical data analysis

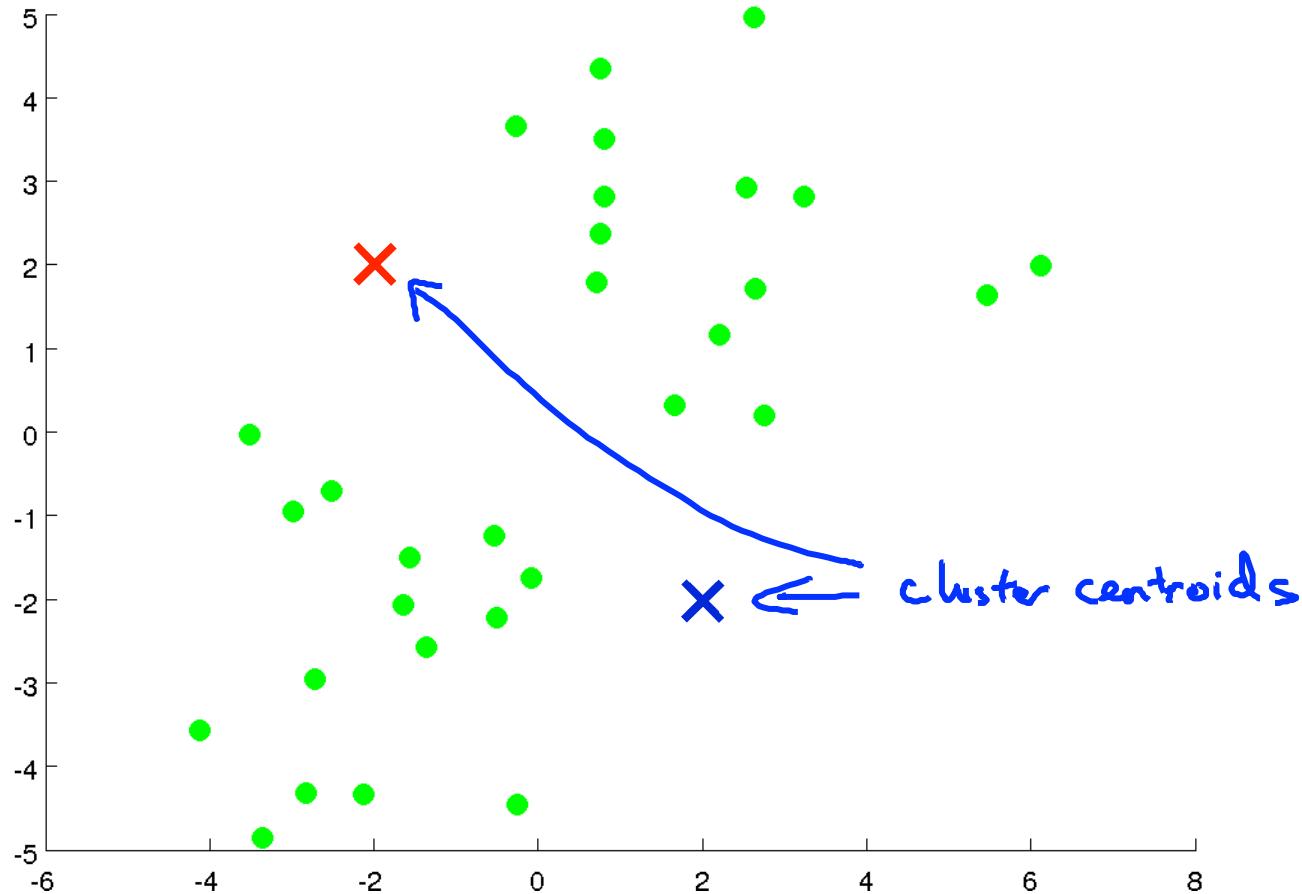


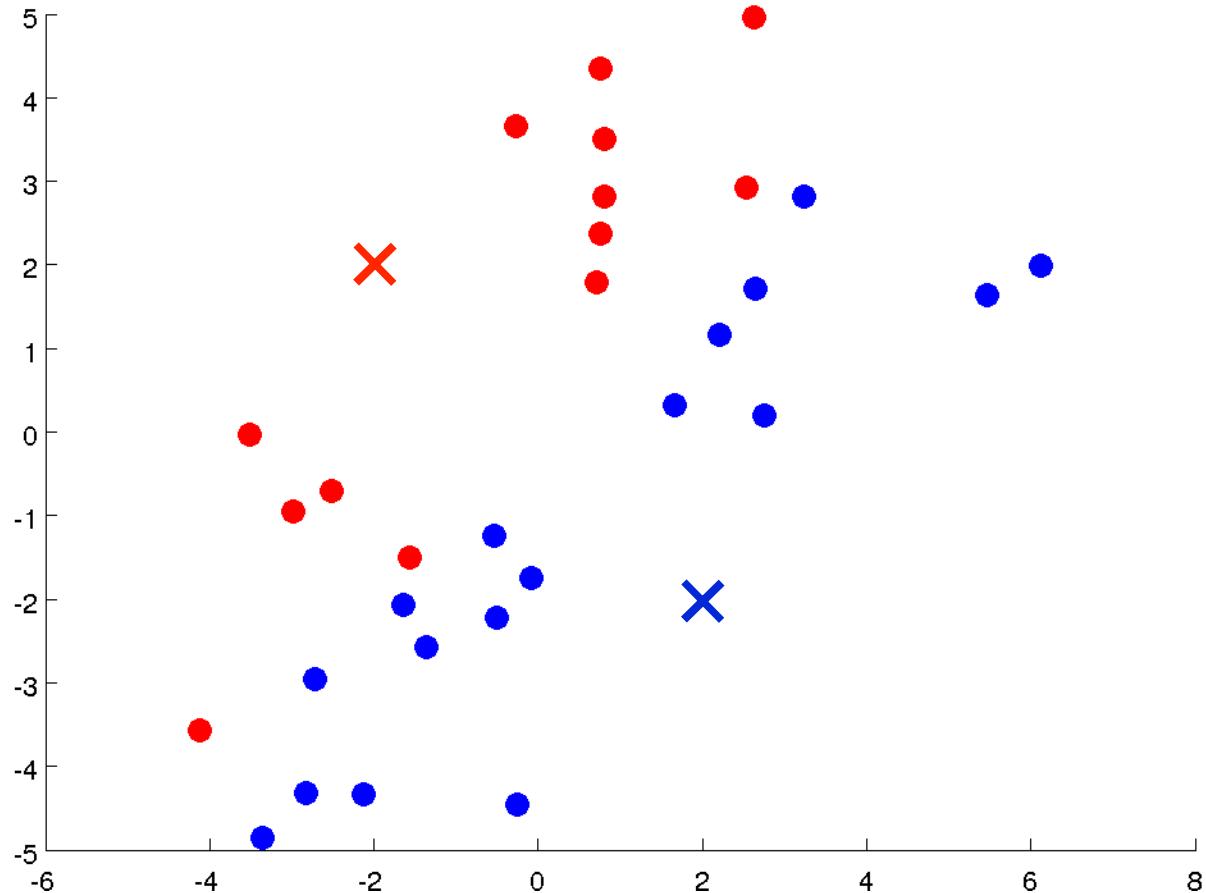
Machine Learning

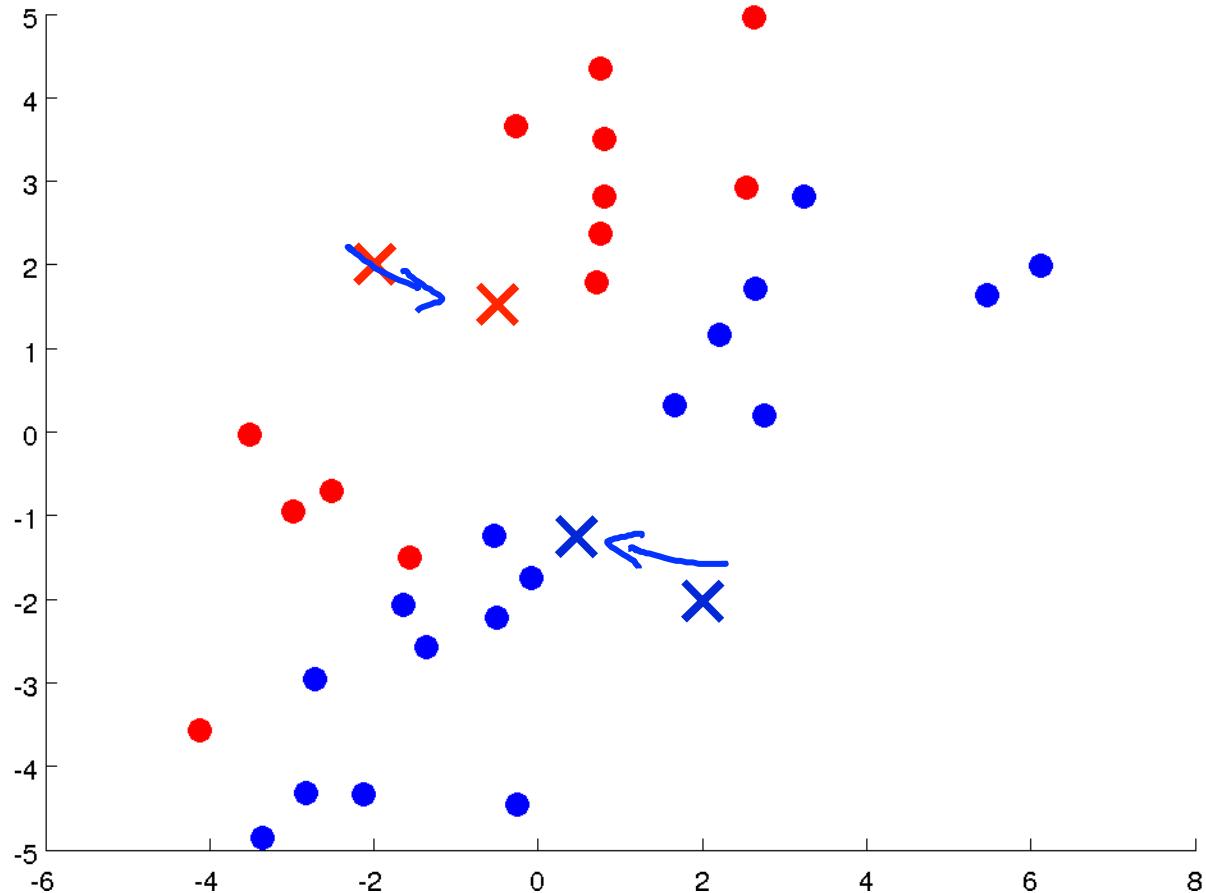
Clustering

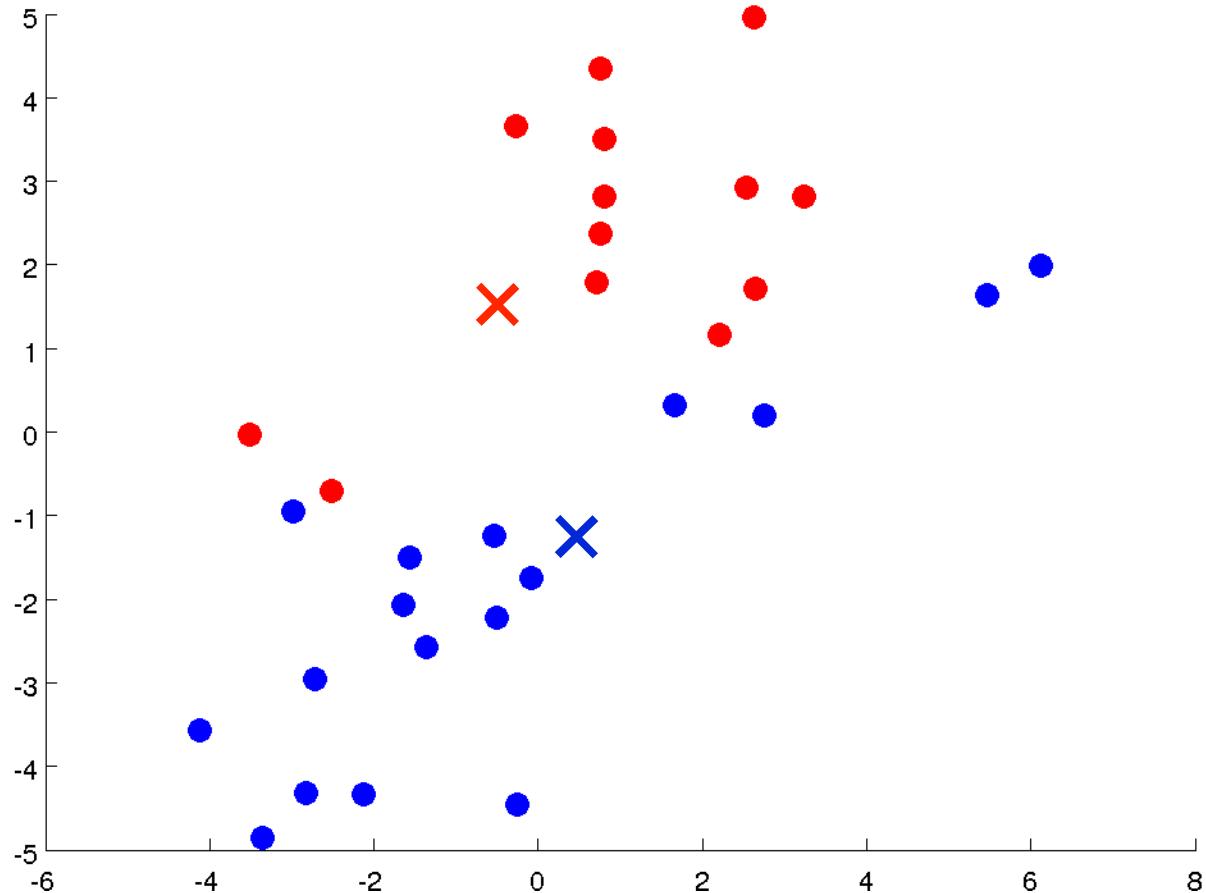
K-means
algorithm

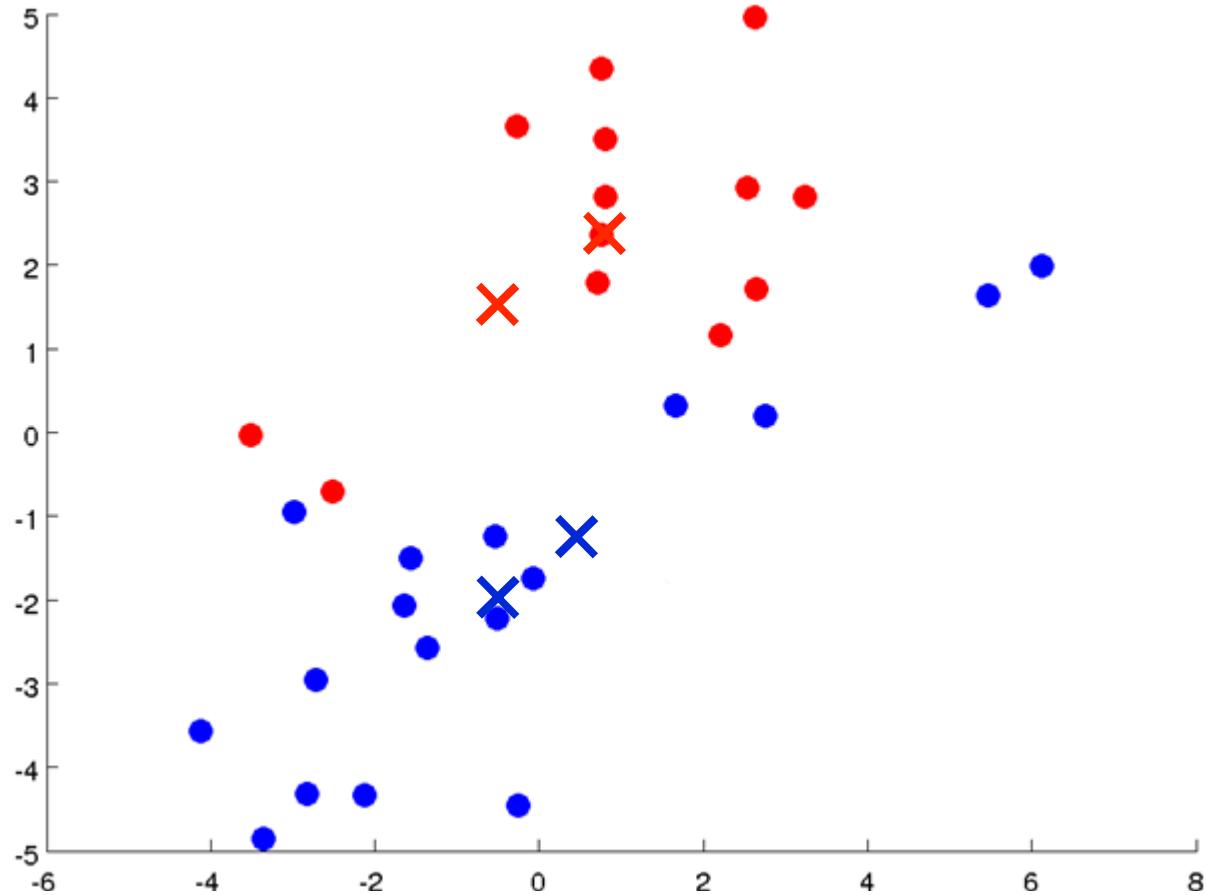


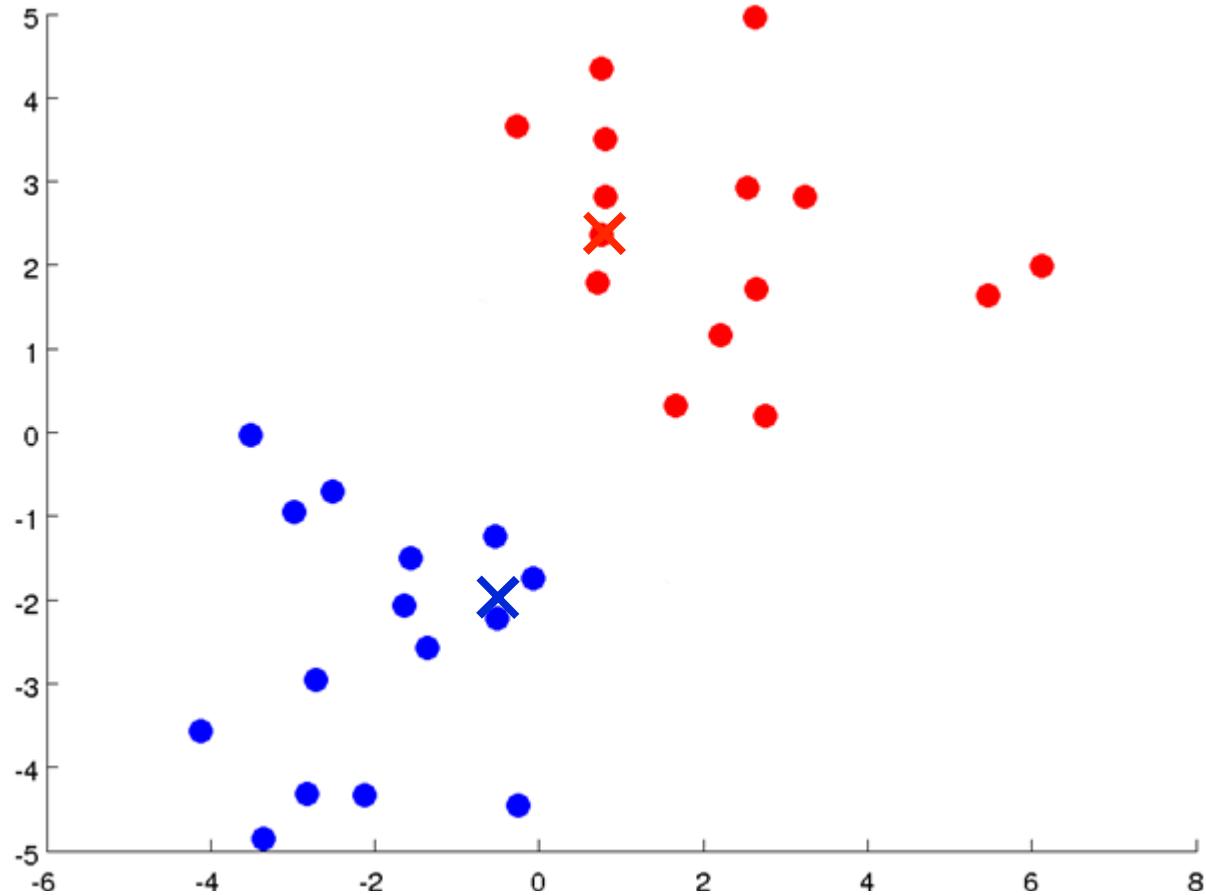


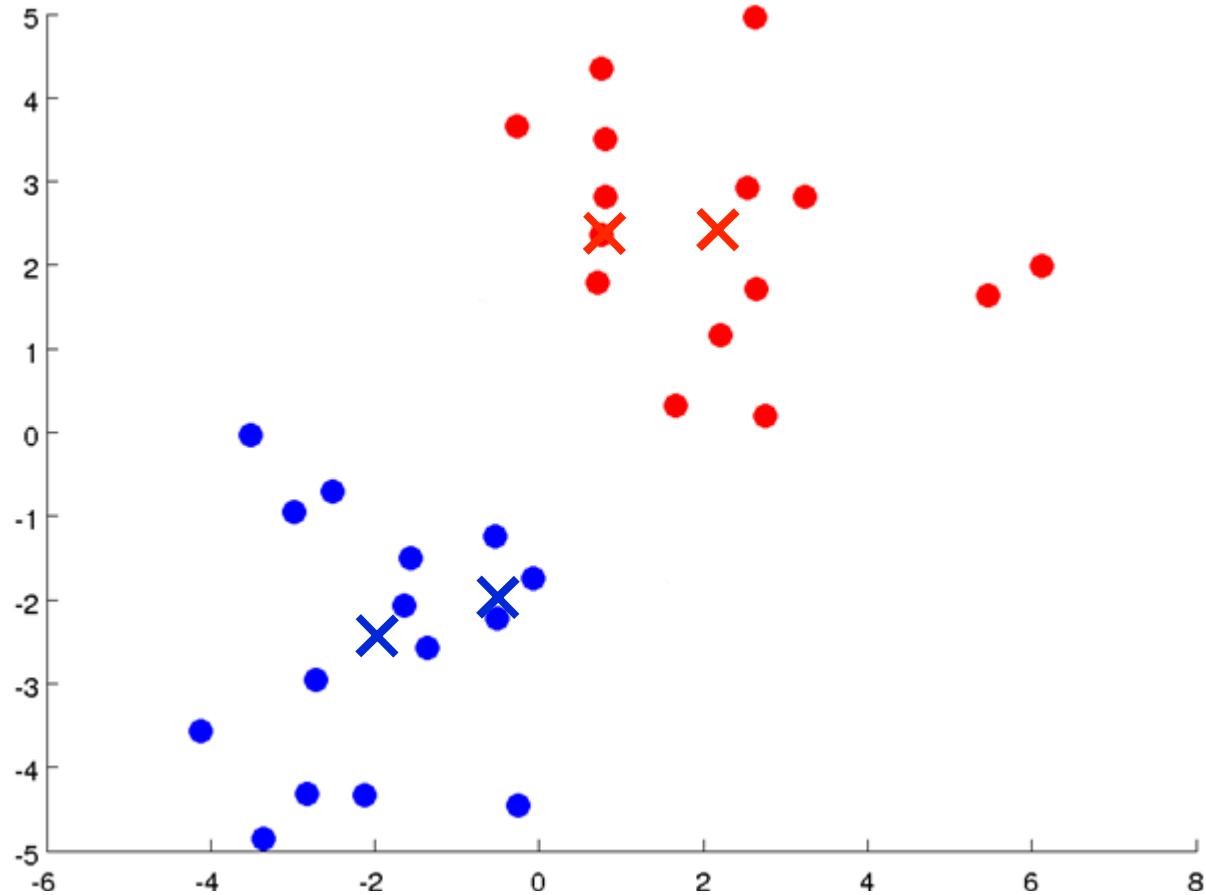


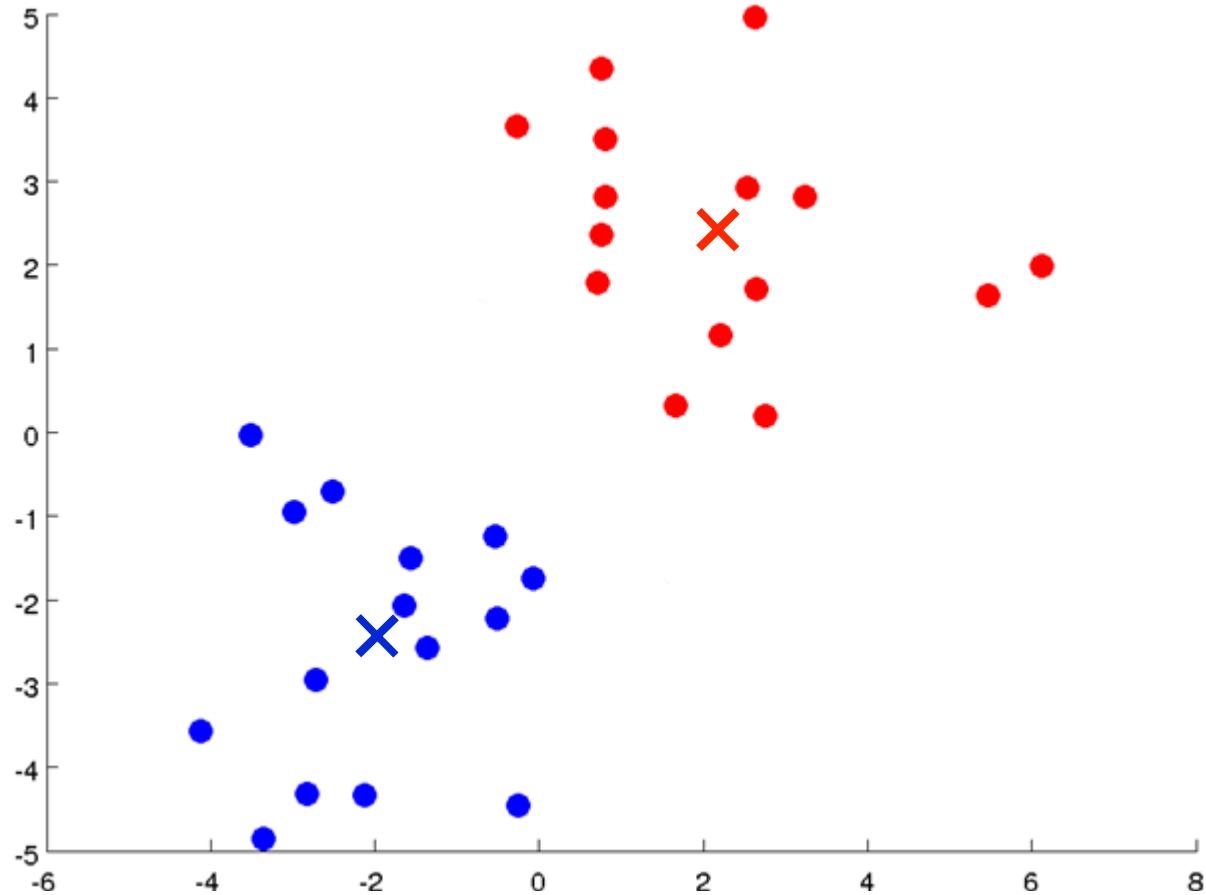












K-means algorithm

Input:

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



Way to determine $c(i)$



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

The screenshot shows a question from a machine learning exam. The question asks to suppose three cluster centroids $\mu_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} -3 \\ 0 \end{bmatrix}$, and $\mu_3 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$. It also mentions a training example $x^{(i)} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$. The question asks what $c^{(i)}$ will be after a cluster assignment step. The correct answer is $c^{(i)} = 2$.

2. Suppose we have three cluster centroids $\mu_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} -3 \\ 0 \end{bmatrix}$ and $\mu_3 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$. Furthermore, we have a training example $x^{(i)} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$. After a cluster assignment step, what will $c^{(i)}$ be?

$c^{(i)}$ is not assigned
 $c^{(i)} = 3$
 $c^{(i)} = 2$
 $c^{(i)} = 1$

Incorrect
 $x^{(i)}$ is closest to μ_3 , so $c^{(i)} = 3$, not 2

3. K-means is an iterative algorithm, and two of the following steps are repeatedly carried out in its inner-loop. Which two?

Feature scaling, to ensure each feature is on a comparable scale to the others.

K-means algorithm



Randomly initialize K cluster centroids $\underline{\mu}_1, \underline{\mu}_2, \dots, \underline{\mu}_K \in \mathbb{R}^n$

Repeat {

Cluster
assignment
step

for $i = 1$ to m

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

$$\min_k \|\underline{x}^{(i)} - \underline{\mu}_k\|^2$$

for $k = 1$ to K

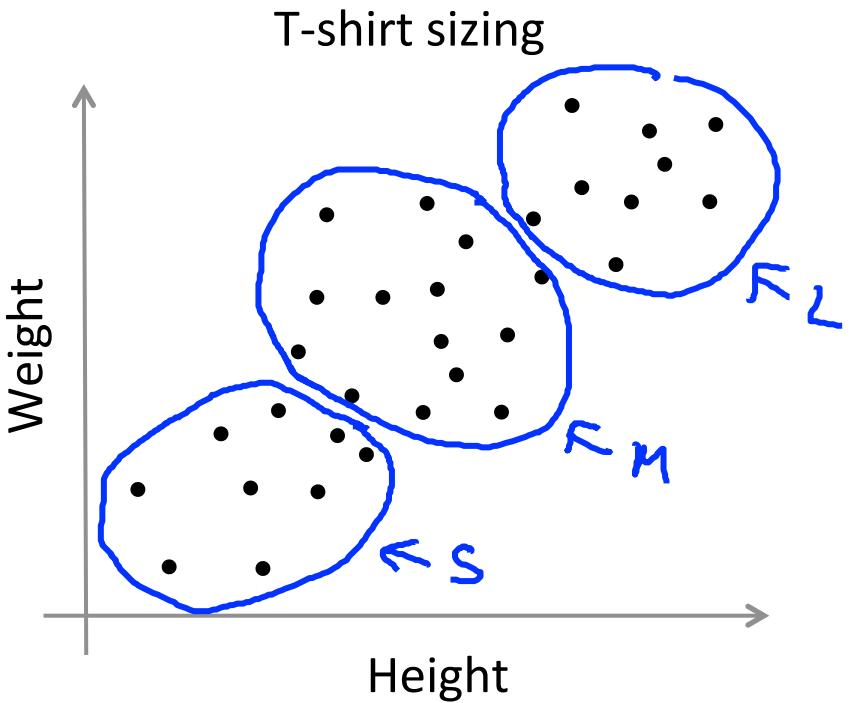
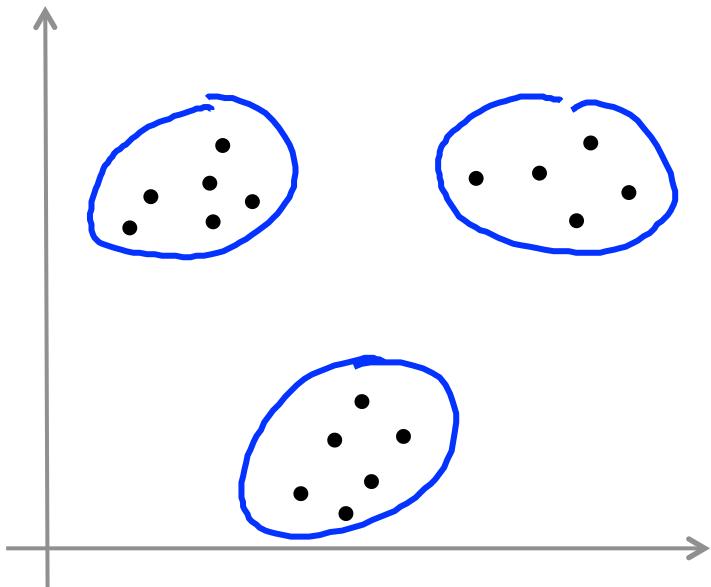
$\rightarrow \underline{\mu}_k$:= average (mean) of points assigned to cluster k
 $\underline{x}^{(1)}, \underline{x}^{(2)}, \underline{x}^{(3)}, \underline{x}^{(4)}$

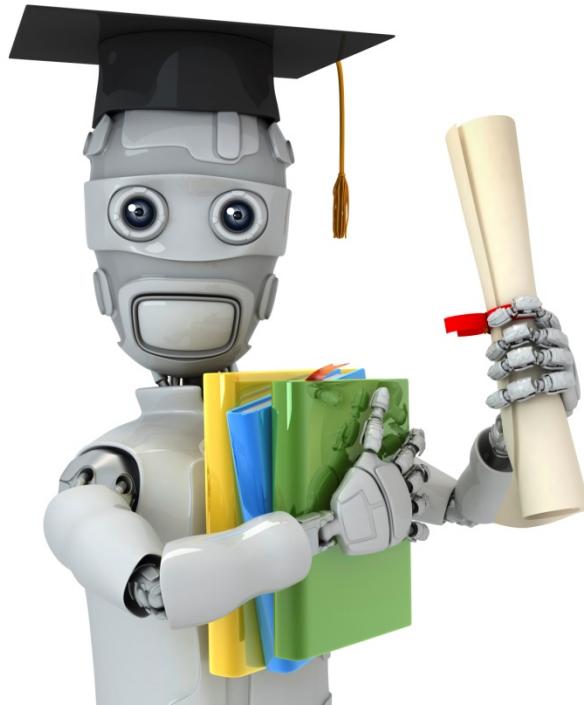
$$\rightarrow \underline{c}^{(1)}=2, \underline{c}^{(2)}=2, \underline{c}^{(3)}=2, \underline{c}^{(4)}=2$$

$$\underline{\mu}_2 = \frac{1}{4} \left[\underline{x}^{(1)} + \underline{x}^{(2)} + \underline{x}^{(3)} + \underline{x}^{(4)} \right] \in \mathbb{R}^n$$

K-means for non-separated clusters

S, M, L





Machine Learning

Clustering Optimization objective

K-means optimization objective

- $c^{(i)}$ = index of cluster ($1, 2, \dots, K$) to which example $x^{(i)}$ is currently assigned
- μ_k = cluster centroid k ($\mu_k \in \mathbb{R}^n$) K $k \in \{1, 2, \dots, K\}$
- ✓ $\underline{\mu_{c^{(i)}}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned $x^{(i)} \rightarrow S$ $c^{(i)} = s$ $\underline{\mu_{c^{(i)}}} = \mu_s$ ✓

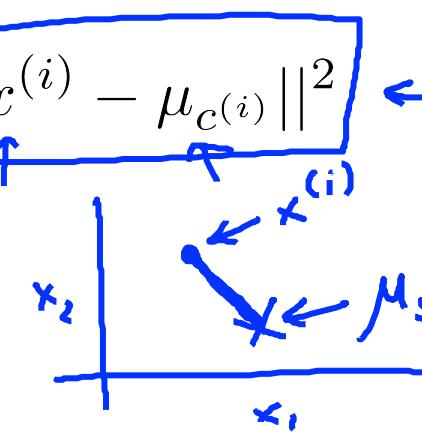
Optimization objective:

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

↑ ↓
 $x^{(i)}$ $c^{(i)}$

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

Distortion



K-means algorithm

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

Cluster assignment step
Repeat {
Minimize $J(\dots)$ wrt $[c^{(1)}, c^{(2)}, \dots, c^{(n)}] \leftarrow$
(holding μ_1, \dots, μ_K fixed)}

for $i = 1$ to m

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

for $k = 1$ to K

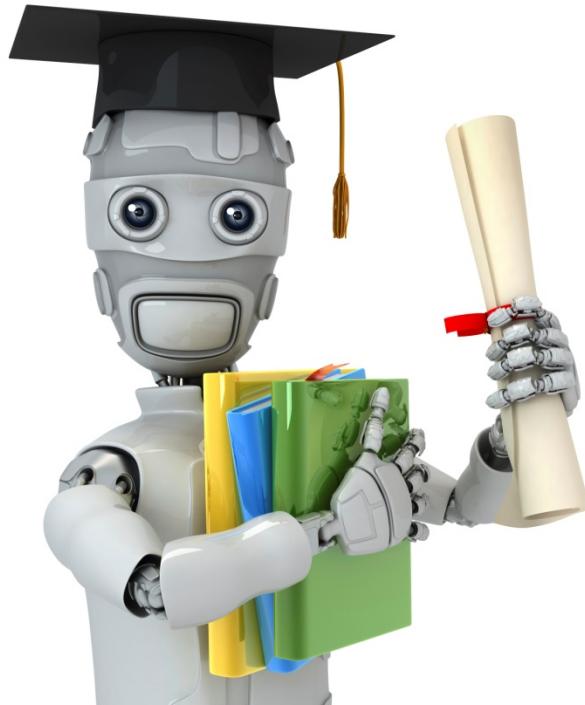
$\mu_k :=$ average (mean) of points assigned to cluster k

}

minimize $J(\dots)$ wrt

μ_1, \dots, μ_K

move
centroid



Machine Learning

Clustering

Random initialization

K-means algorithm

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

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

}

Random initialization

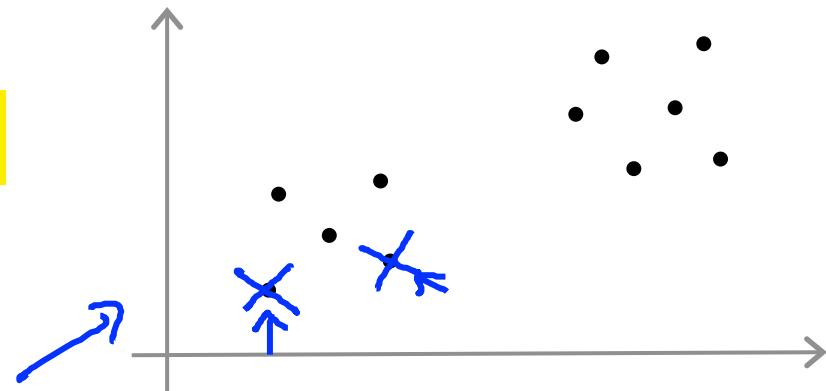
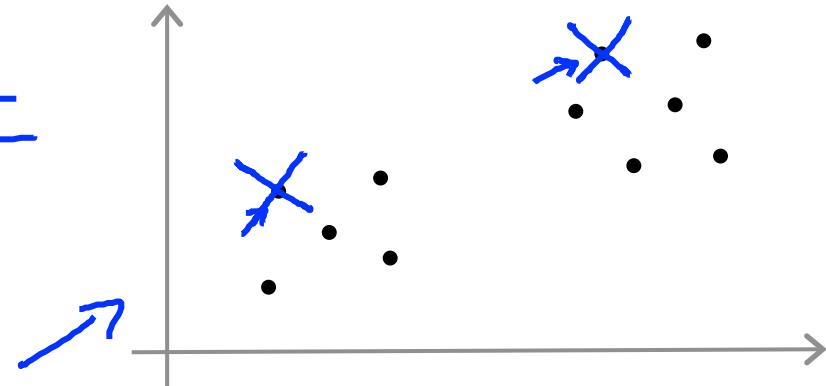
Should have $K < m$

$$\underline{K=2}$$

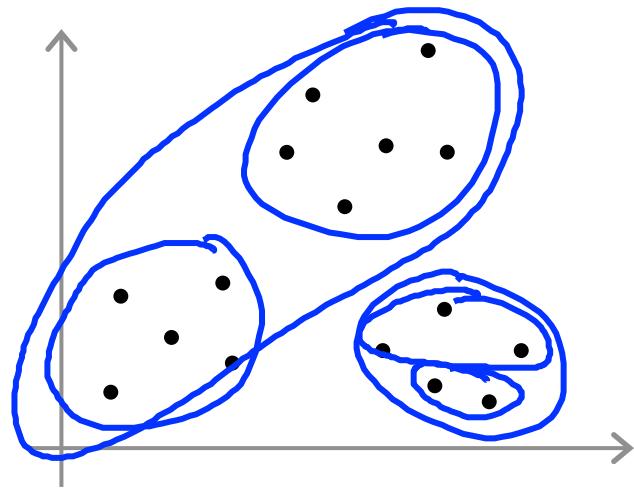
Randomly pick K training examples.

Set μ_1, \dots, μ_K equal to these K examples.

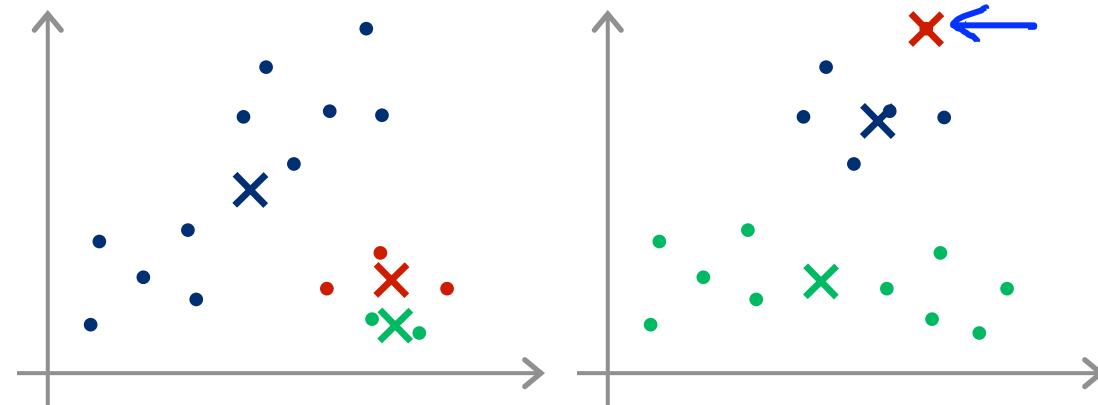
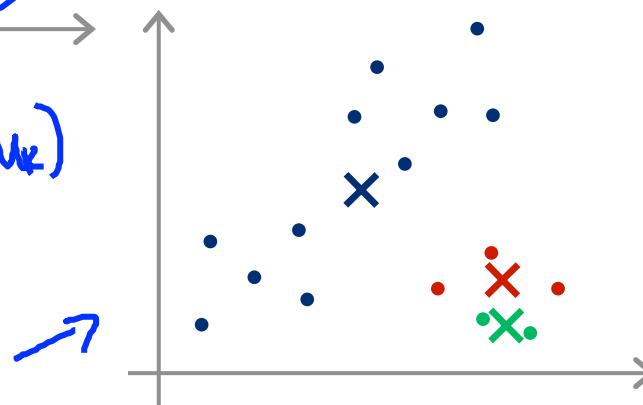
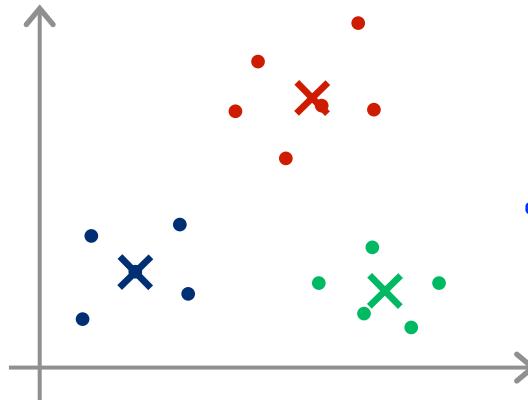
$$\begin{aligned}\mu_1 &= x^{(1)} \\ \mu_2 &= x^{(2)} \\ &\vdots\end{aligned}$$



Local optima



$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_k)$$



Random initialization

For i = 1 to 100 {

 Randomly initialize K-means.

 Run K-means. Get $c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K$.

 Compute cost function (distortion)

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

}

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

This function is the distortion function. Since a lower value for the distortion function implies a better clustering, you should choose the clustering with the smallest value for the distortion function.

5. Which of the following statements are true? Select all that apply.

For some datasets, the "right" or "correct" value of K (the number of clusters) can be ambiguous, and hard even for a human expert looking carefully at the data to decide.

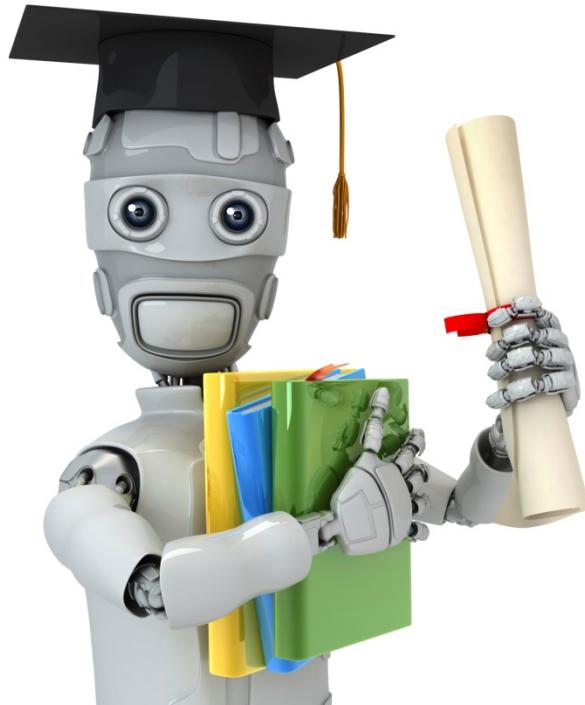
Correct
In many datasets, different choices of K will give different clusterings which appear quite reasonable. With no labels on the data, we cannot say one is better than the other.

The standard way of initializing K-means is setting $\mu_1 = \dots = \mu_k$ to be equal to a vector of zeros.

Since K-Means is an unsupervised learning algorithm, it cannot overfit the data, and thus it is always better to have as large a number of clusters as is computationally feasible.

If we are worried about K-means getting stuck in bad local optima, one way to ameliorate (reduce) this problem is if we try using multiple random initializations.

Correct
Since each run of K-means is independent, multiple runs can find different optima, and some should avoid bad local optima.

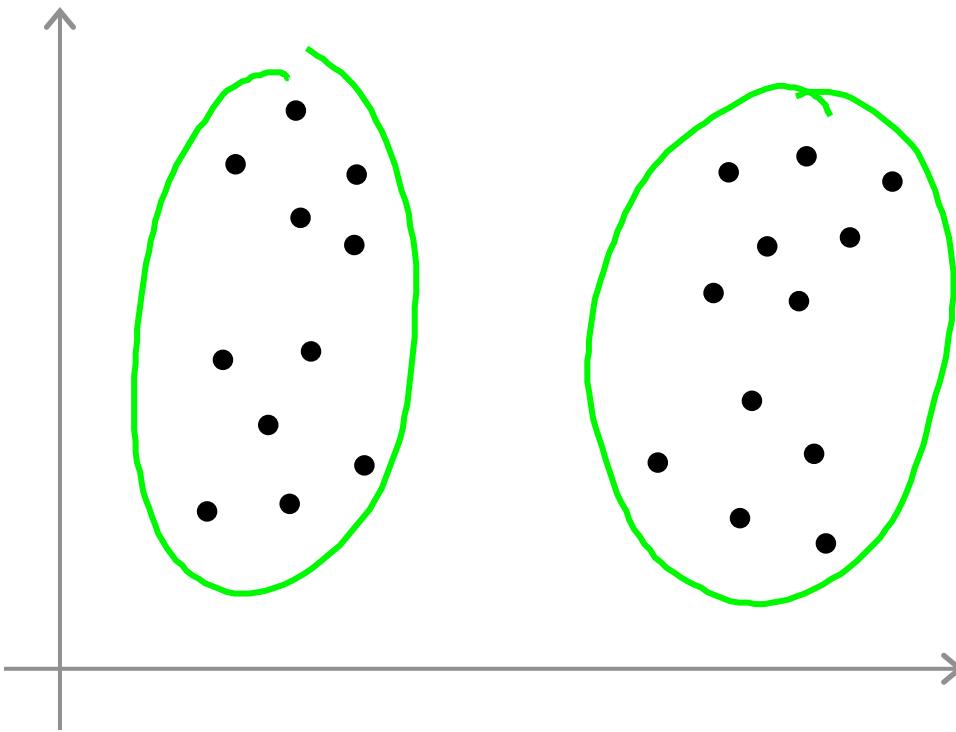


Machine Learning

Clustering

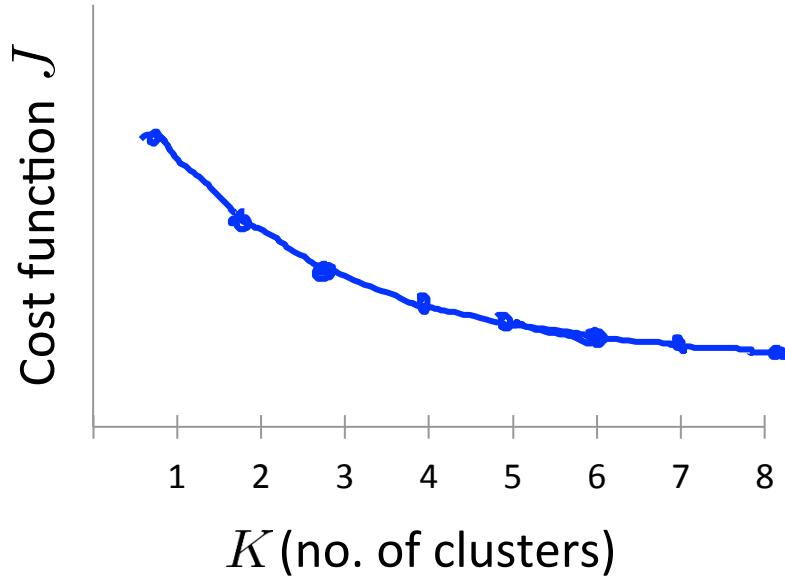
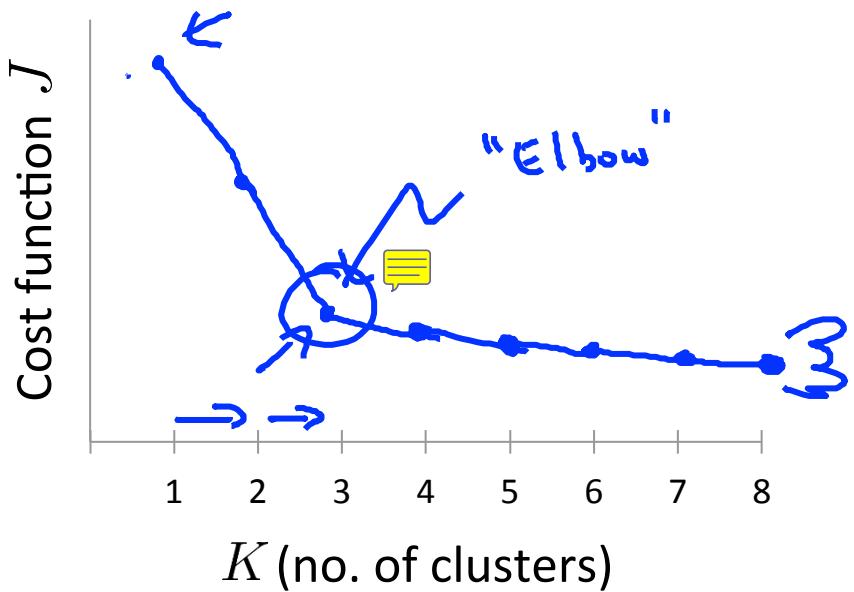
Choosing the number of clusters

What is the right value of K?



Choosing the value of K

Elbow method:

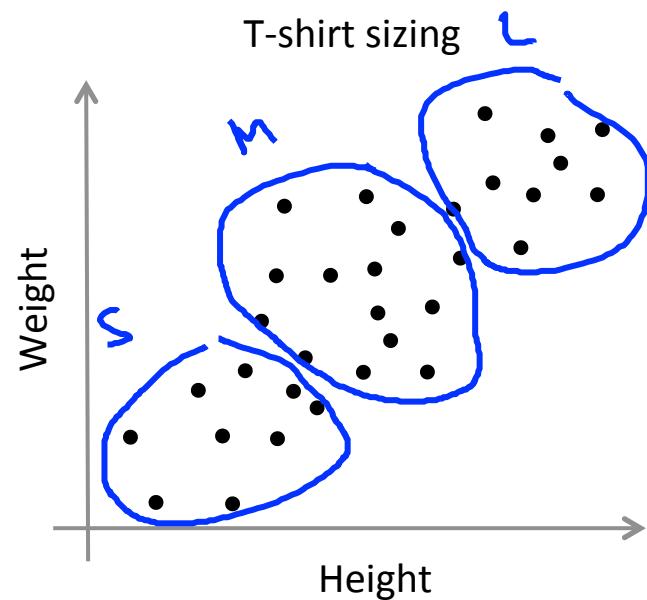


Choosing the value of K

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.

$K=3$ S, M, L

E.g.



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

