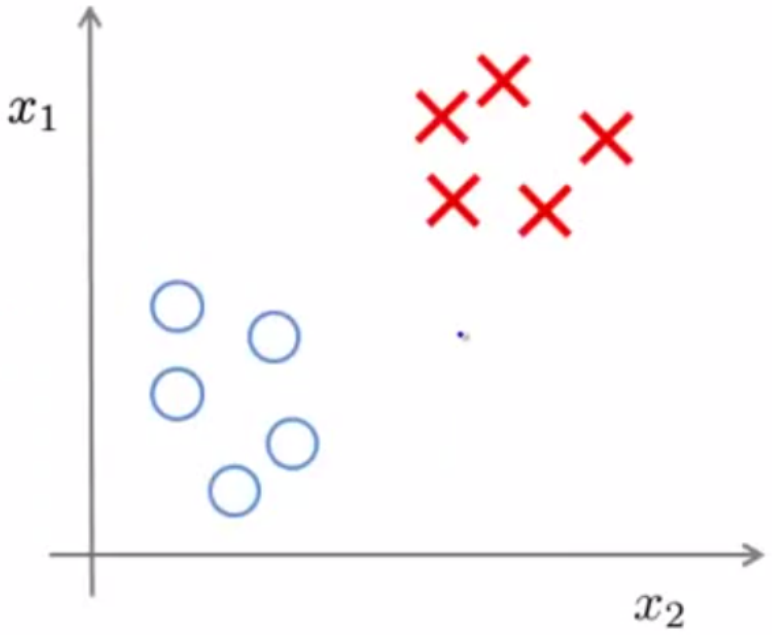
Clustering

Prof.: Eric A. Antonelo

Slides baseados no curso de *Machine Learning* de Andrew Ng

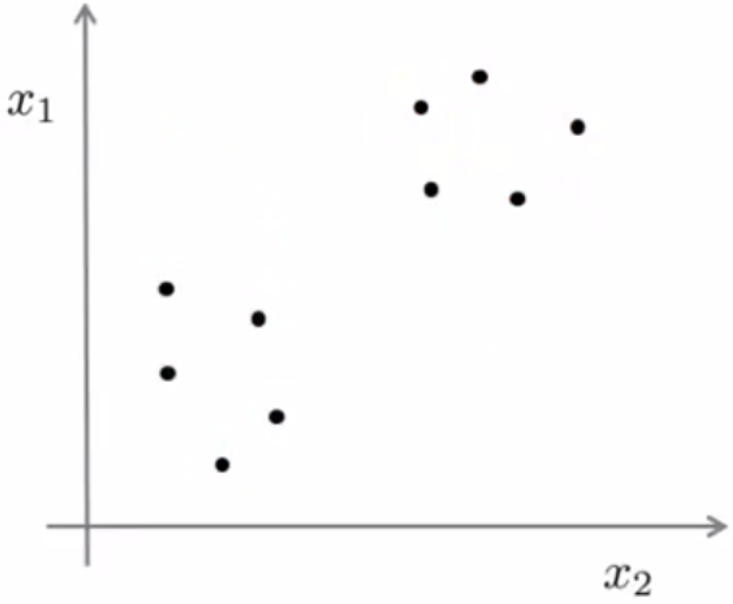
DAS-UFSC

Aprendizagem supervisionada



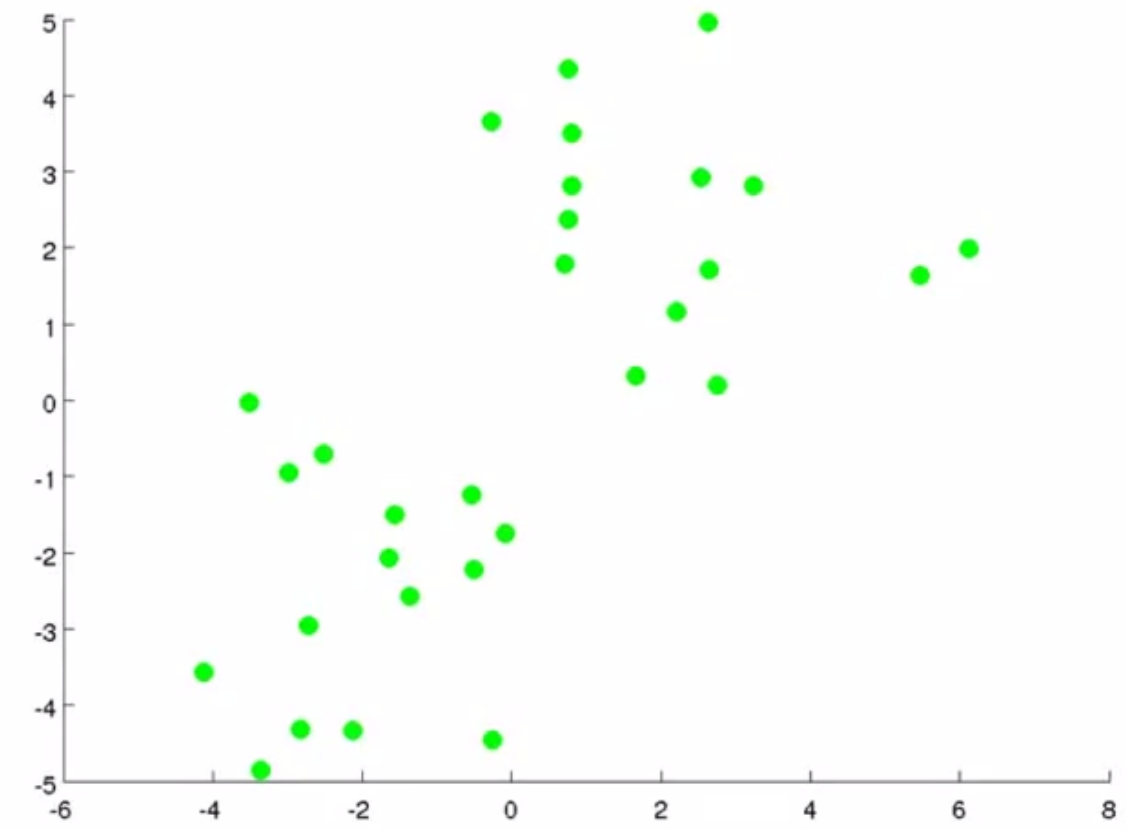
Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

Aprendizagem não-supervisionada

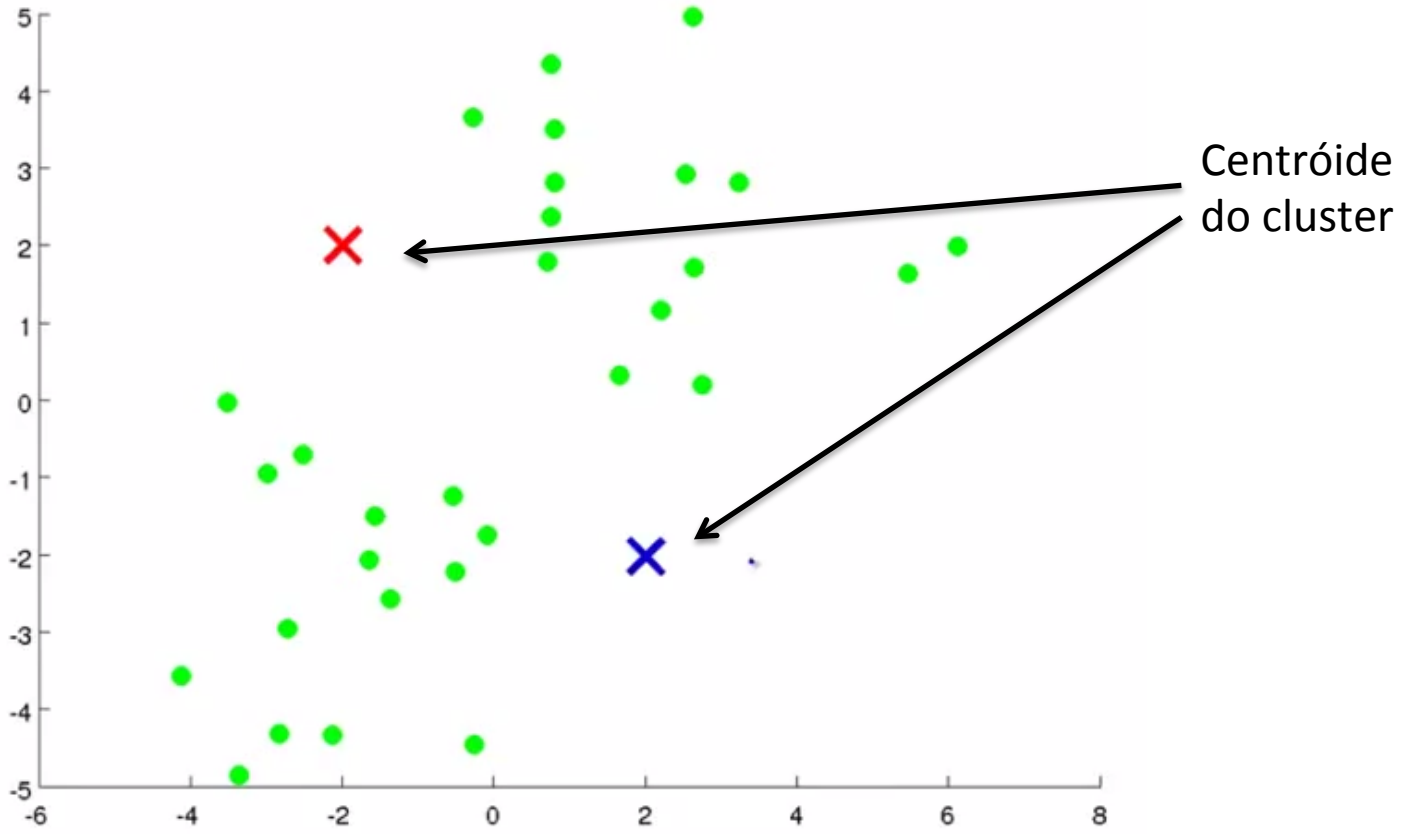


Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

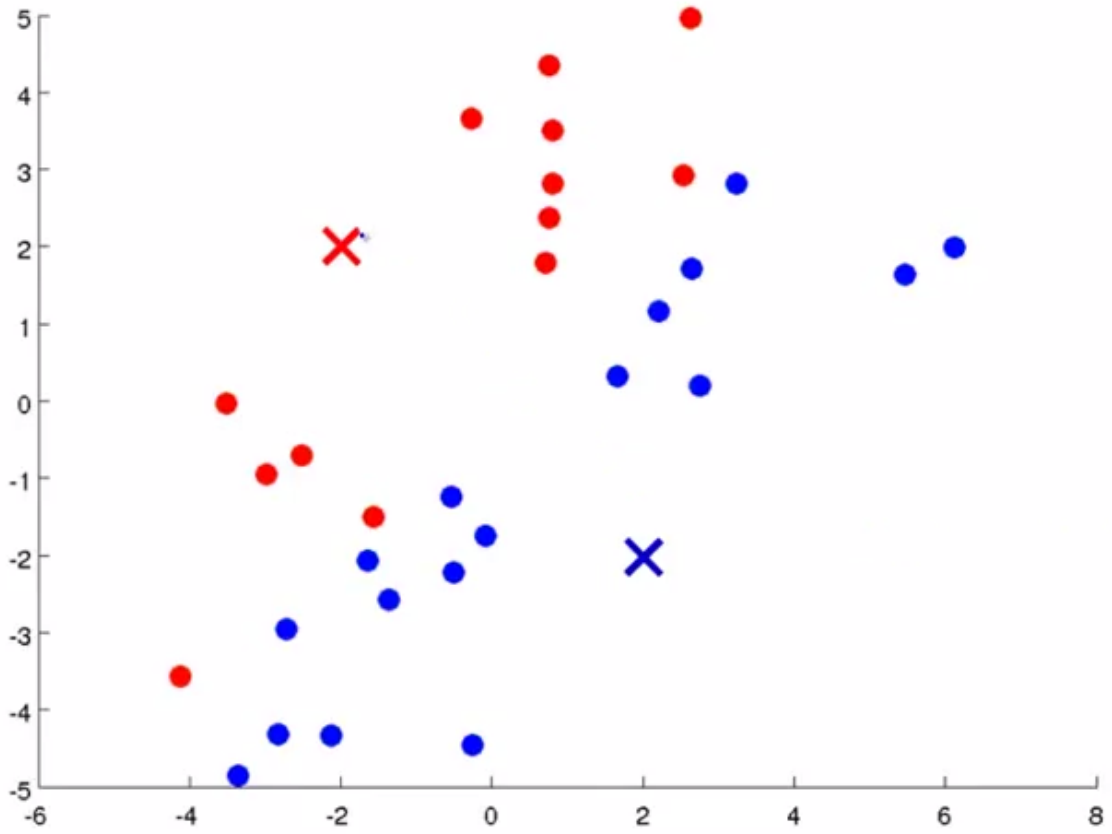
K-means clustering



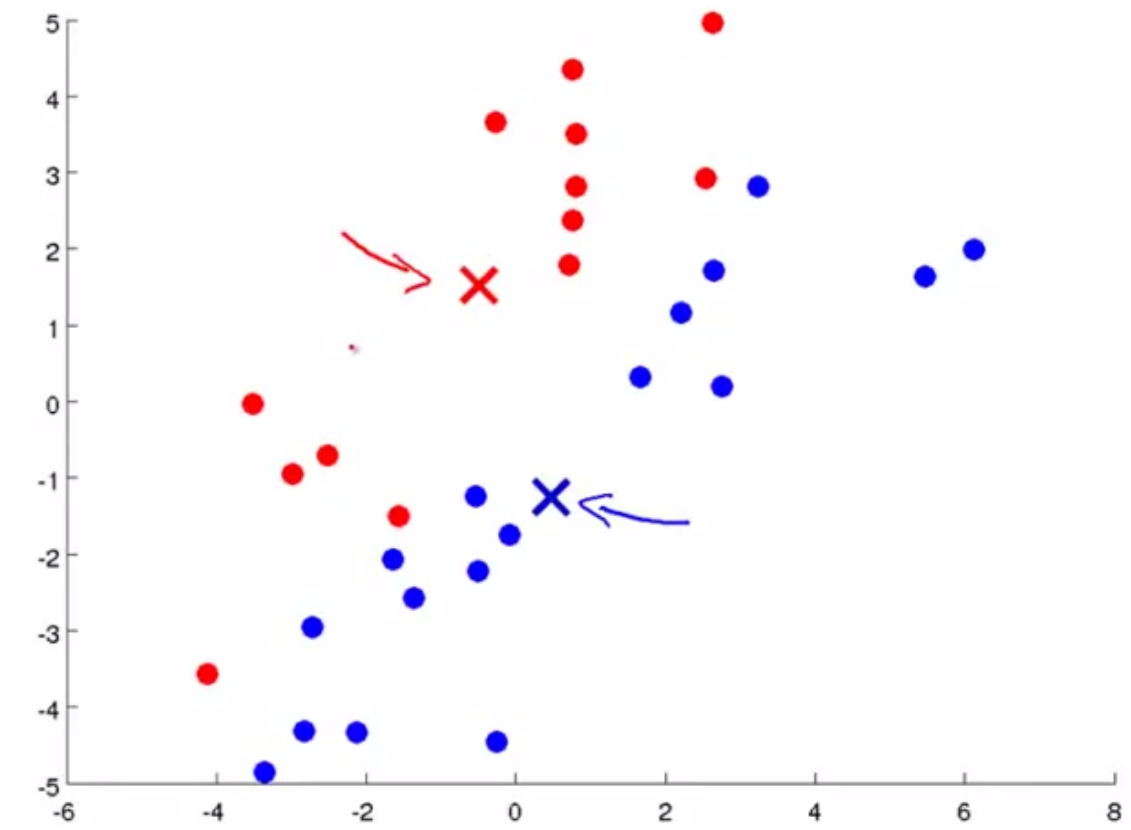
K-means clustering



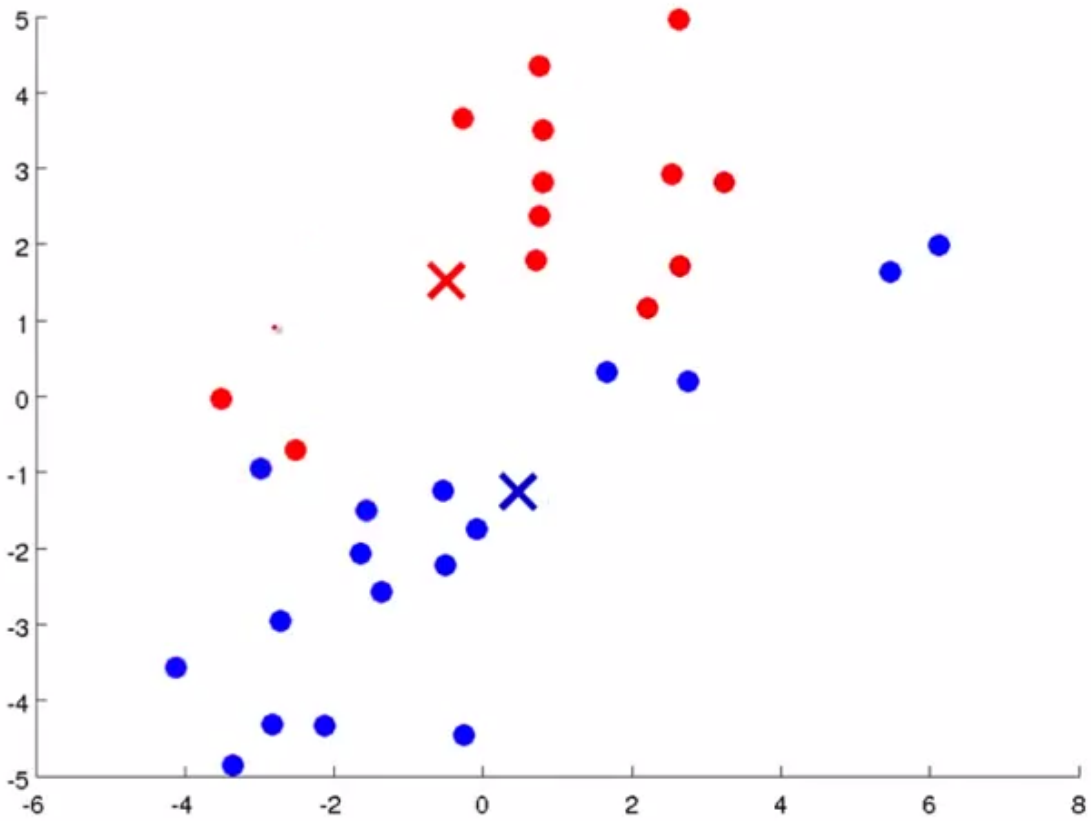
K-means clustering



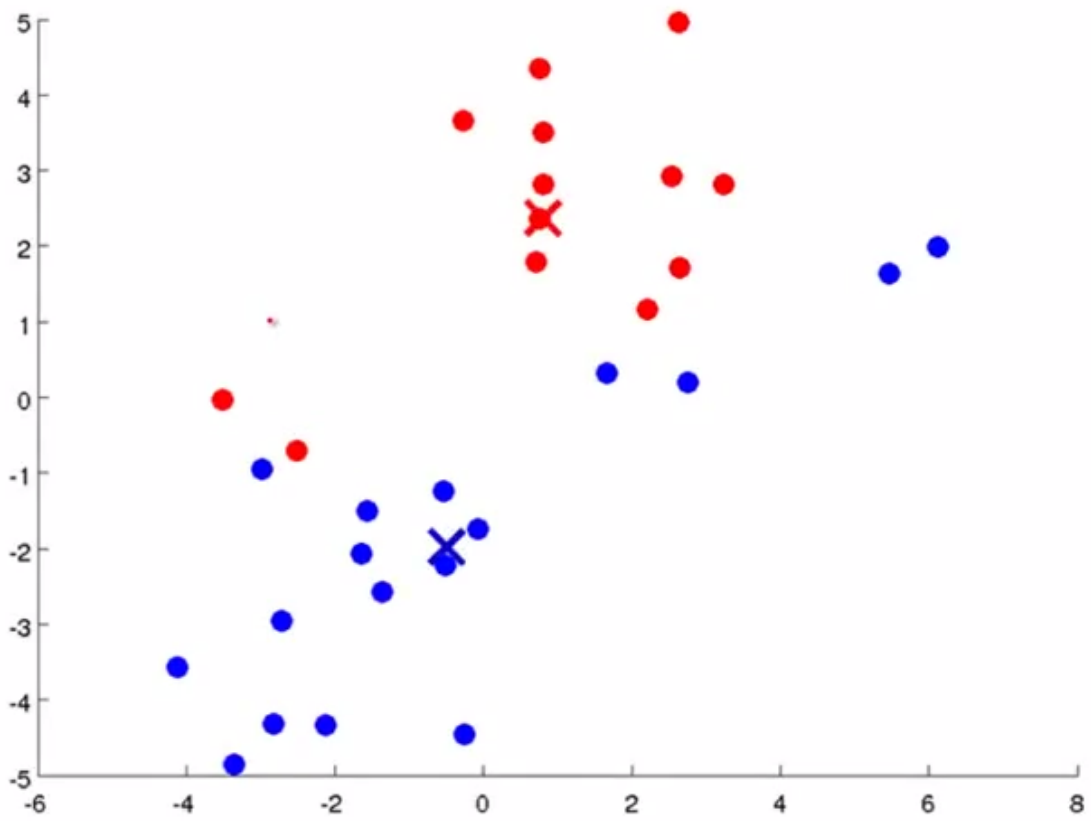
K-means clustering



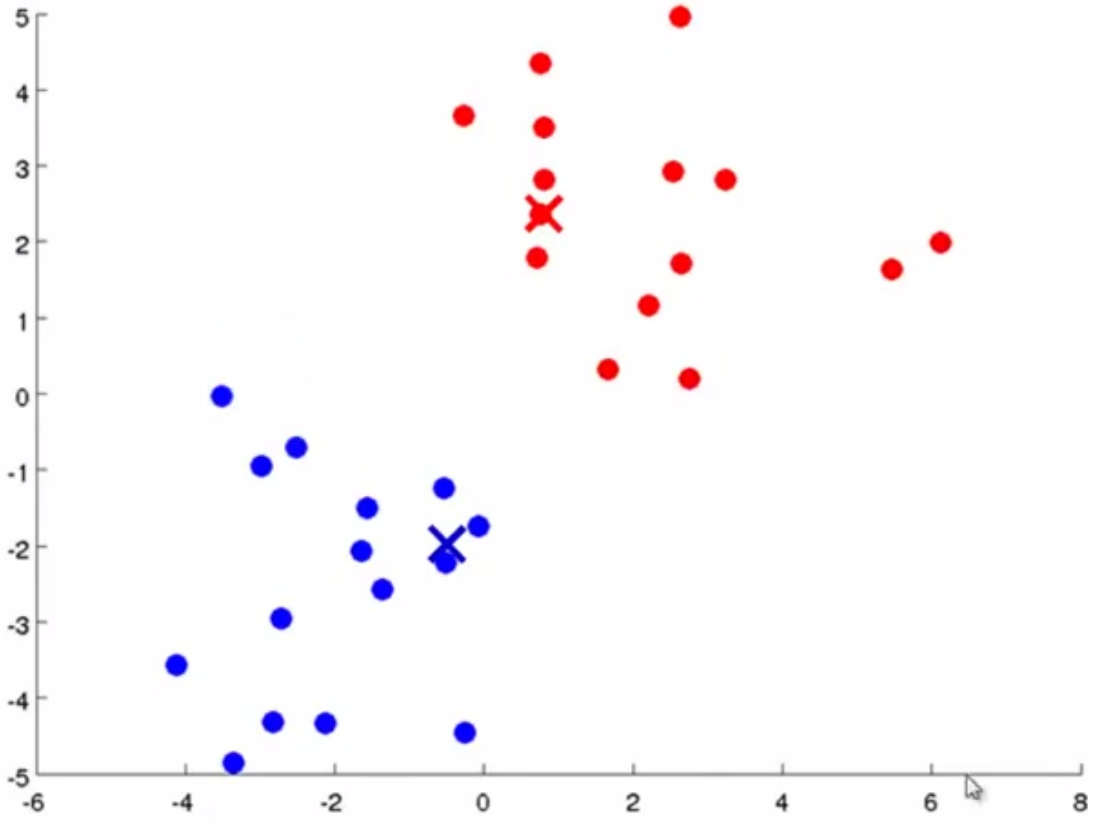
K-means clustering



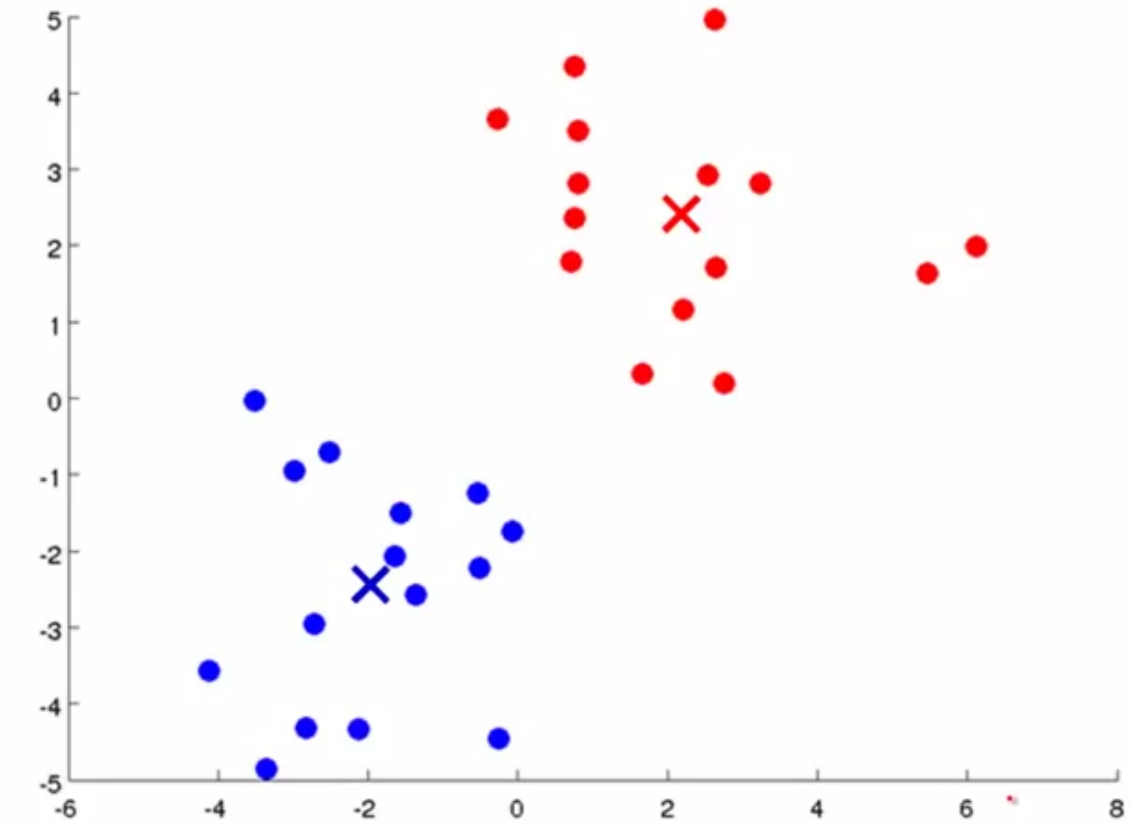
K-means clustering



K-means clustering



K-means clustering



Algoritmo K-means

- Entrada:
  - *K* (número de clusters)
  - *Conjunto de treinamento* $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

Algoritmo K-means

Inicializar aleatoriamente *K* centróides de cluster

$$\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$$

Repetir {

  para *i* = 1 até *m*

    c(*i*) := índice (de 1 a *K*) do centróide mais próximo de *x*(*i*)

  para *k* = 1 até *K*

    μ(*k*) := média dos pontos pertencentes ao cluster *k*

}

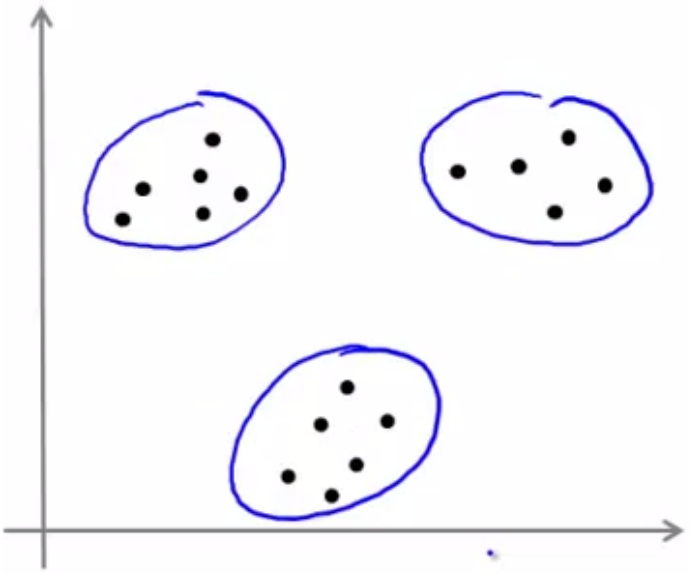
PASSO 1

Atribuição de clusters

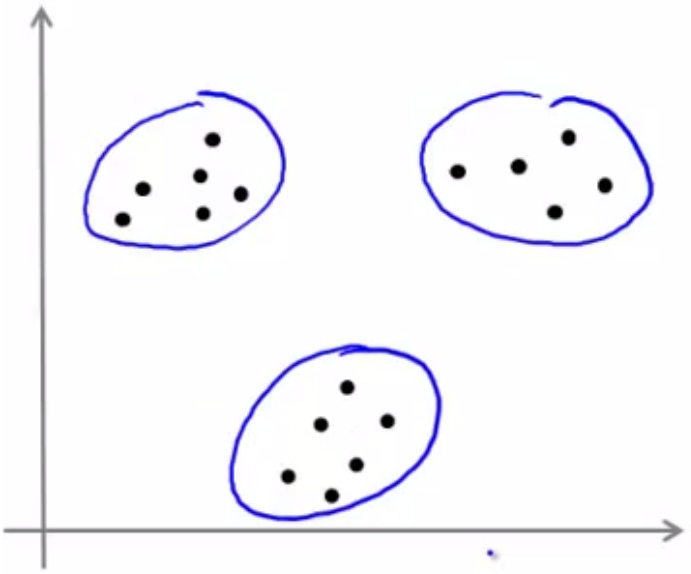
PASSO 2

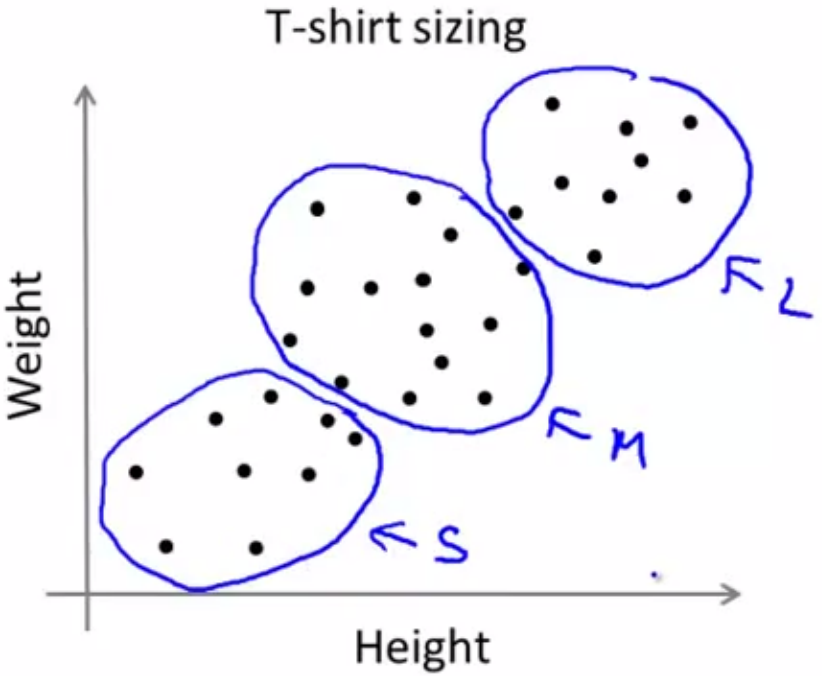
Move centróides

K-means para clusters não-separados



K-means para clusters não-separados





Objetivo de Otimização

$c^{(i)}$ = Índice do Cluster ao qual $x^{(i)}$ pertence no momento

μ_k = Centróide do Cluster *k* $(\mu_k \in \mathbb{R}^n)$

$\mu_{c^{(i)}}$ = Centróide do Cluster $c^{(i)}$, referente ao cluster atual de $x^{(i)}$

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

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

Algoritmo K-means

Inicializar aleatoriamente *K* centróides de cluster

$$\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$$

Repetir {

  para *i* = 1 até *m*

    c(*i*) := índice (de 1 a *K*) do centróide mais próximo de *x*(*i*)

  para *k* = 1 até *K*

    μ(*k*) := média dos pontos pertencentes ao cluster *k*

}

PASSO 1

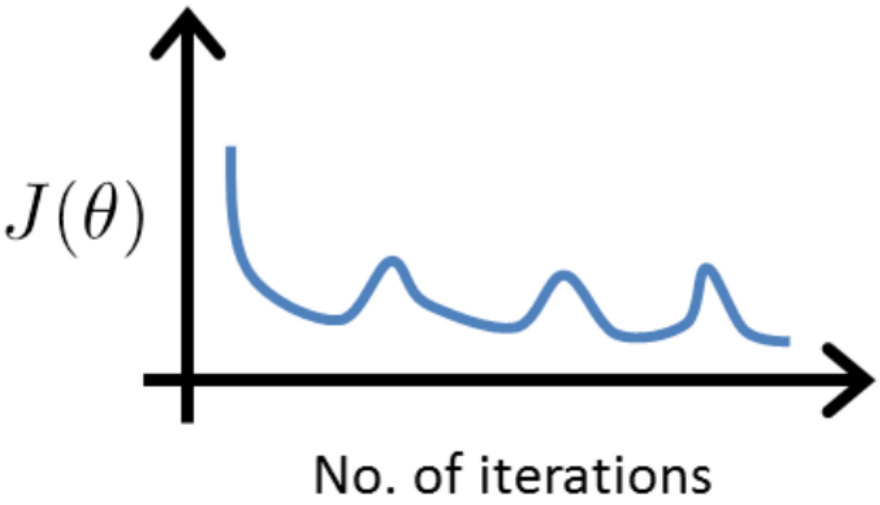
Atribuição de clusters

PASSO 2

Move centróides

Questão: Algoritmo K-means

Após implementar o K-means e plotar a função de custo em função do número de iterações, você obtém:



1. Taxa de aprendizagem alta.
2. Algoritmo está correto.
3. Está funcionando, mas *k* está alto.
4. A função de custo não pode diminuir. A implementação tem um bug.

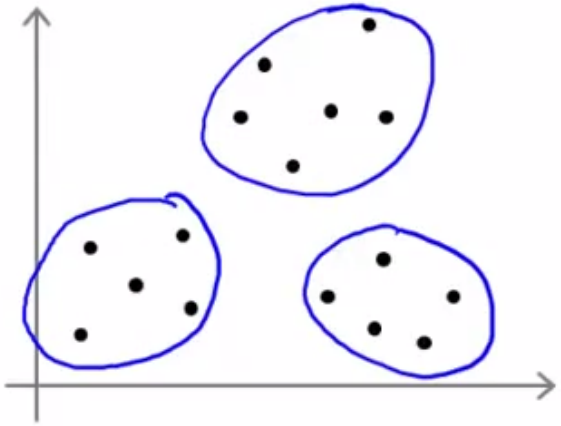
Inicialização aleatória

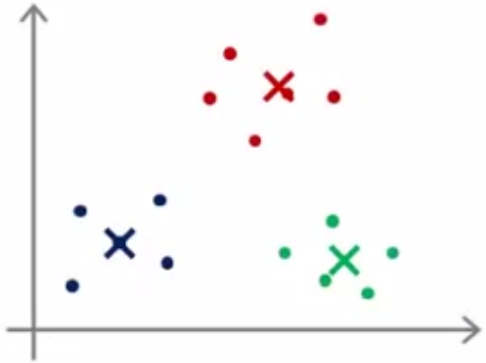
$K < m$

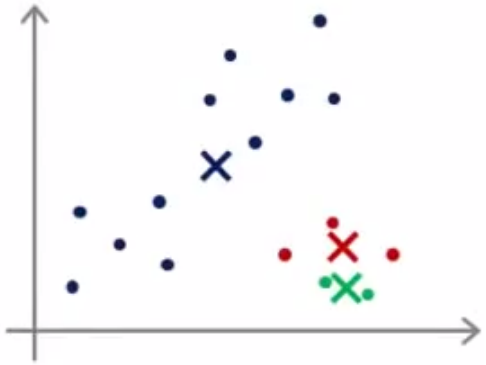
Escolher K exemplos de treinamento.

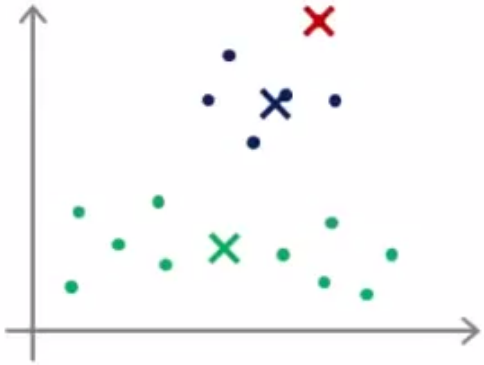
Fazer os centróides $\mu_1, \ldots, \mu_K$ serem iguais a esses K exemplos.

Ótimos locais









Inicialização aleatória

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

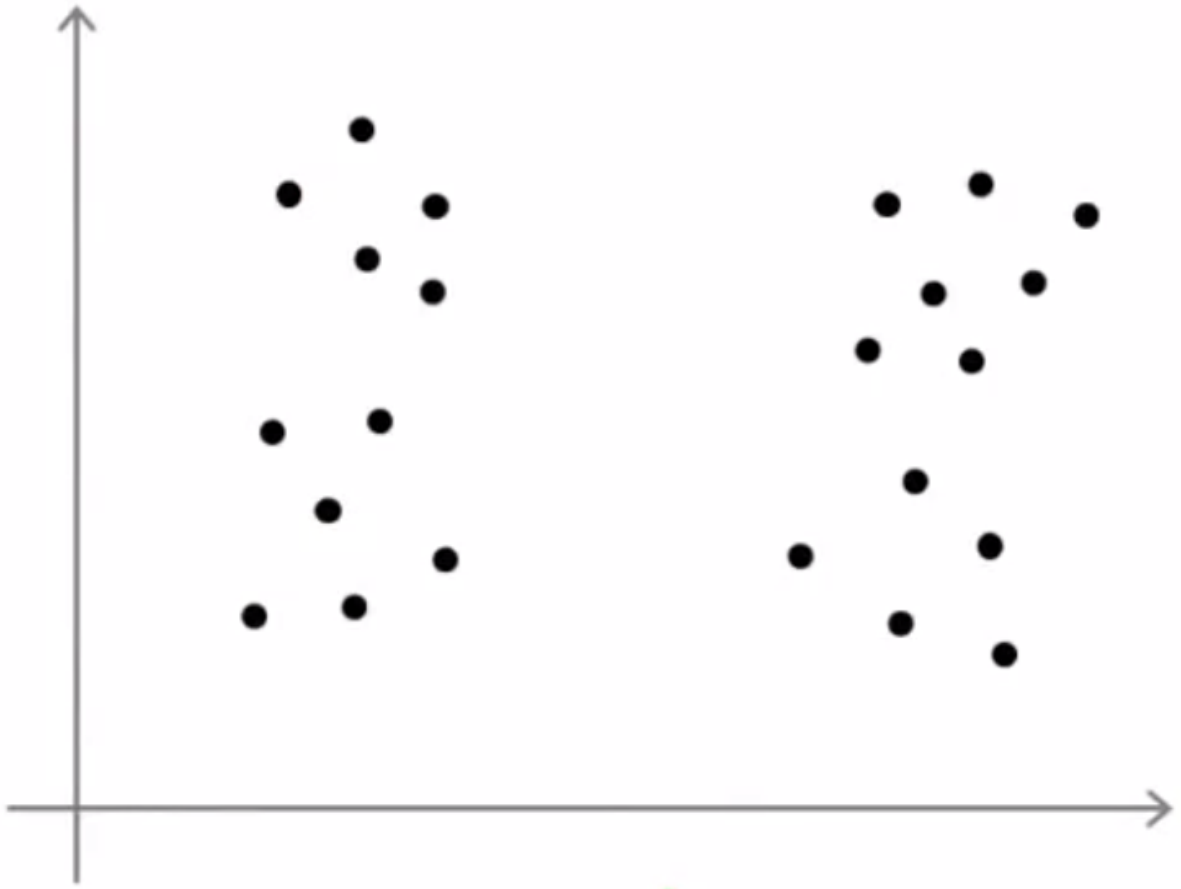
Repetidas inicializações aleatórias valem a pena quando:

2 <= *K* <= 10

Escolhendo K?

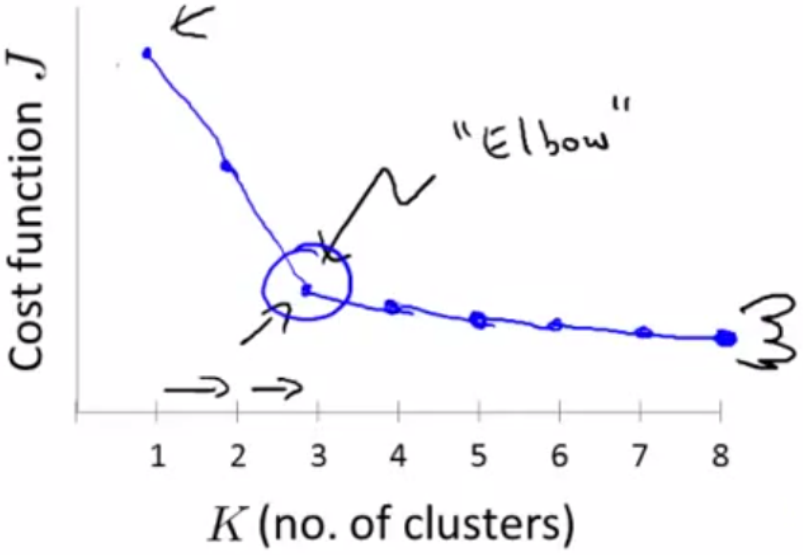
Muito comum escolher **manualmente**.

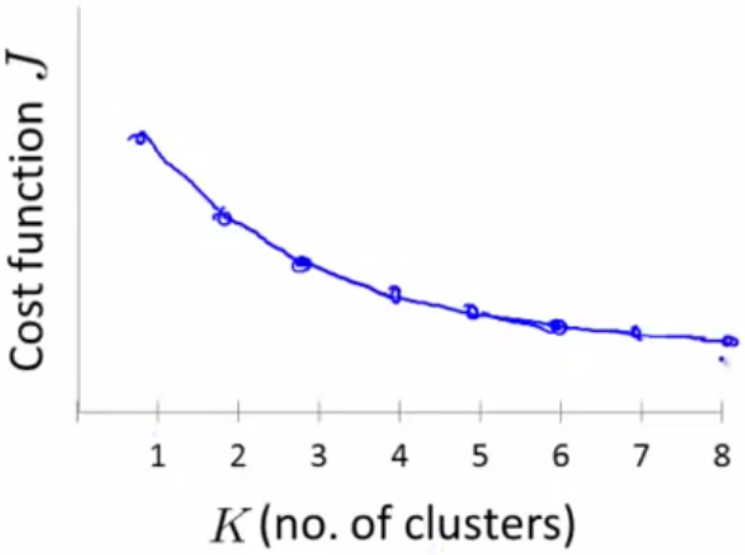
Escolhendo K?



Método “Elbow” (Cotovelo)

Método “Elbow” (Cotovelo)





Questão

Suponha que você rode K-means com K=3 e K=5.

A função de custo para K=5 é muito maior que para K=3. O que você conclui?

1. Matematicamente impossível. Algum bug?
2. O número correto de clusters é k = 3.
3. Na execução de k = 5, k-means ficou preso em um mínimo local. Deve-se rodar k-means múltiplas vezes.
4. Na execução de k = 3, k-means teve sorte. Deve-se rodar k-means até não ser melhor que k = 5.

Escolhendo K?

Escolher K de forma a atender uma métrica ou objetivo relacionada ao problema/área de negócio.