Week 8

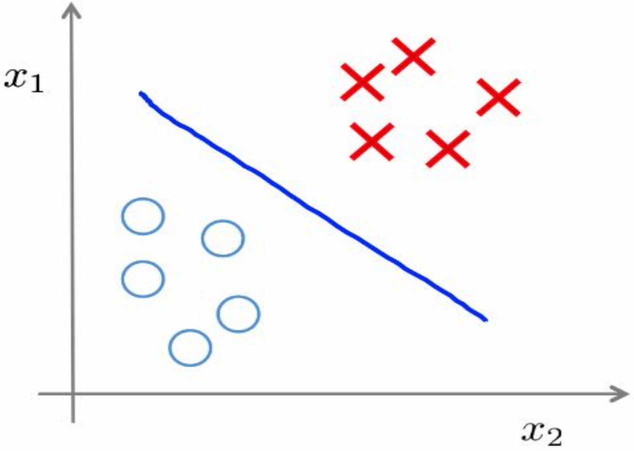
AI Seminar Series

K-Means Clustering

Use-case jupyternotebooks

Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13

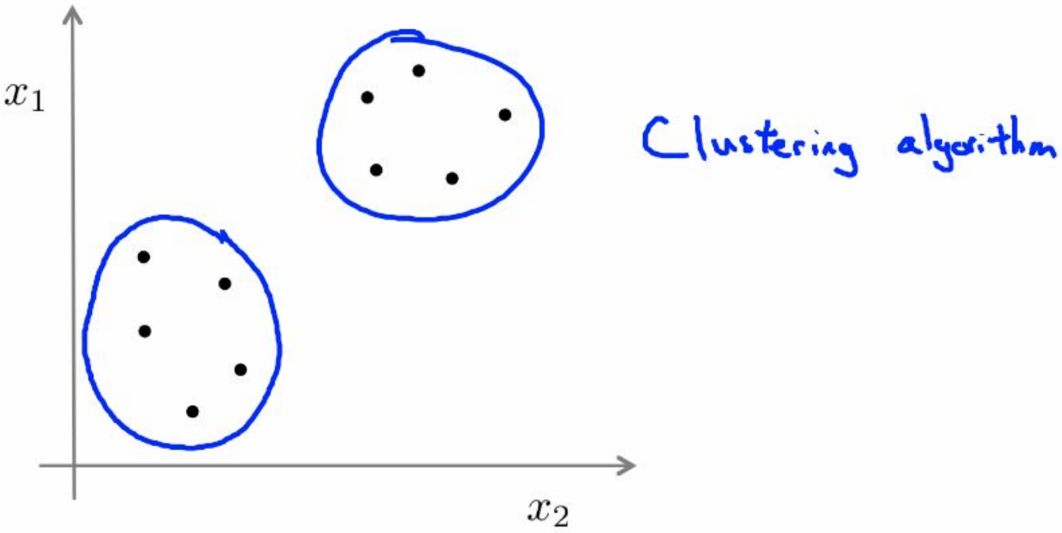
Supervised learning



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

Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

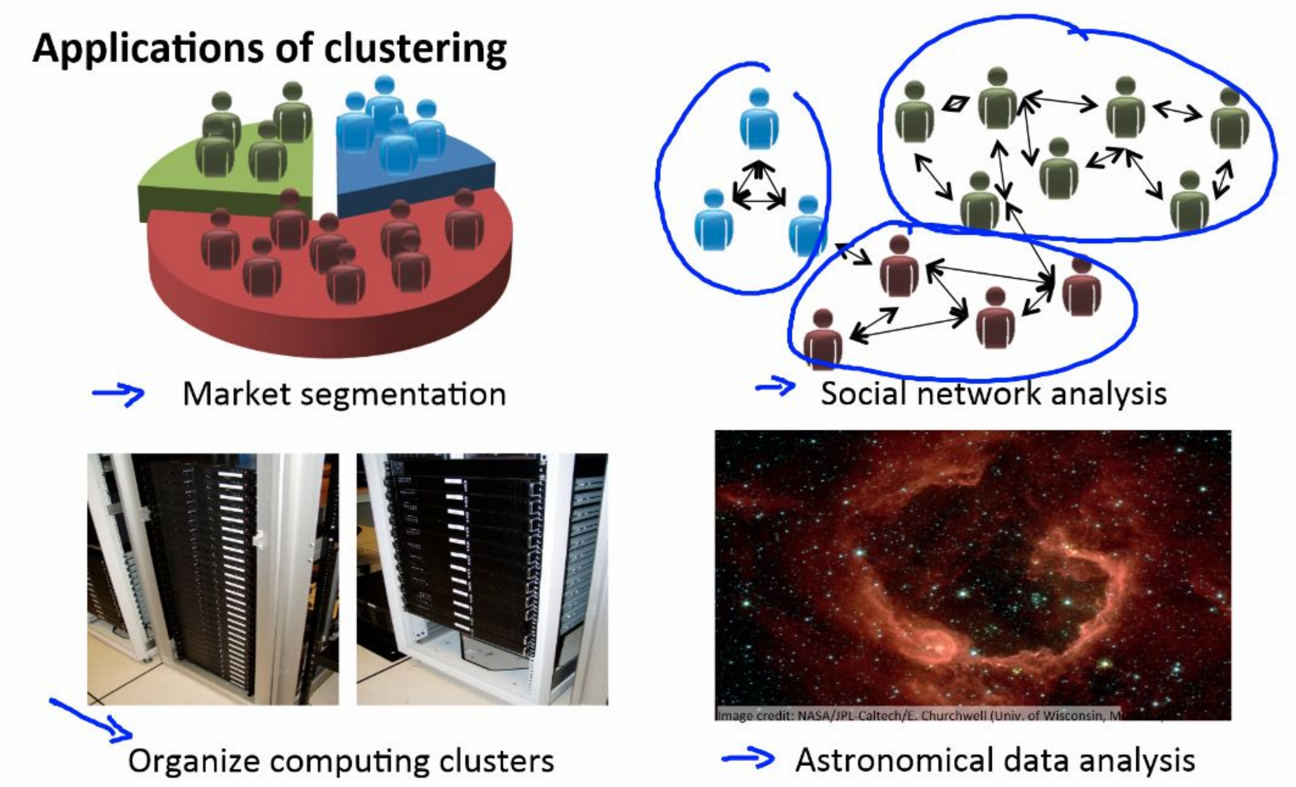
Unsupervised learning



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

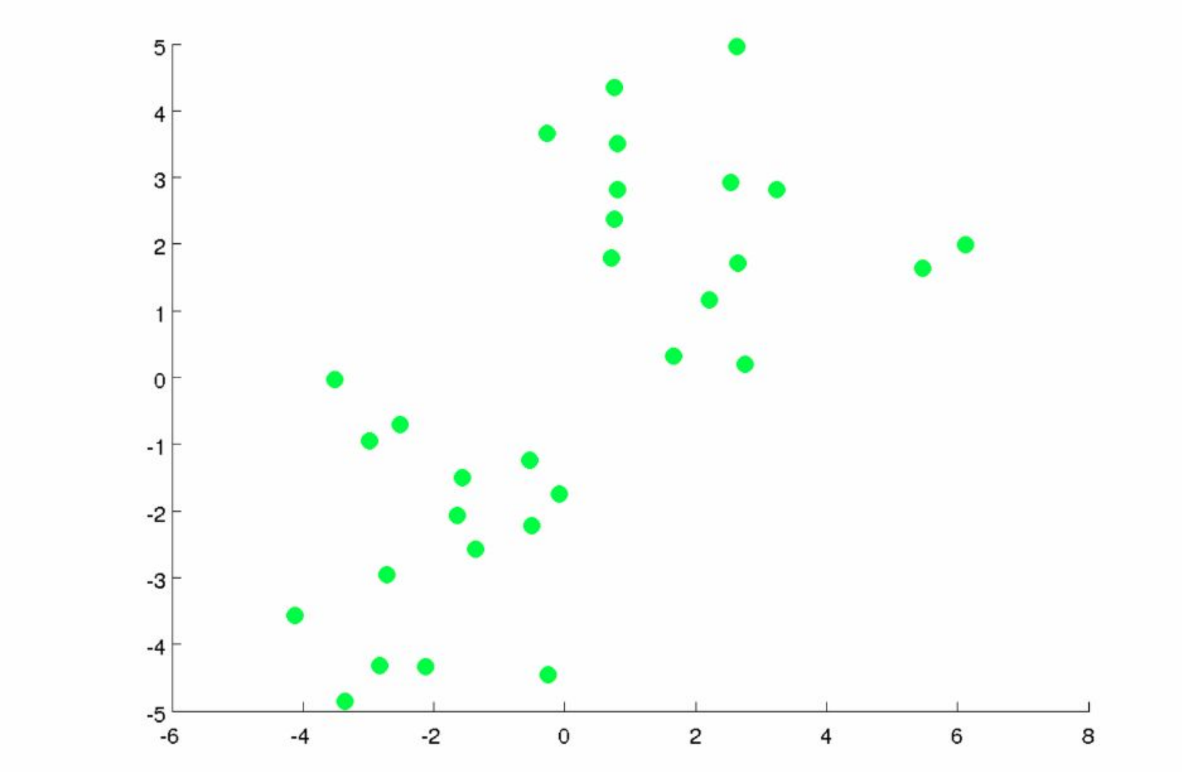
Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

Applications



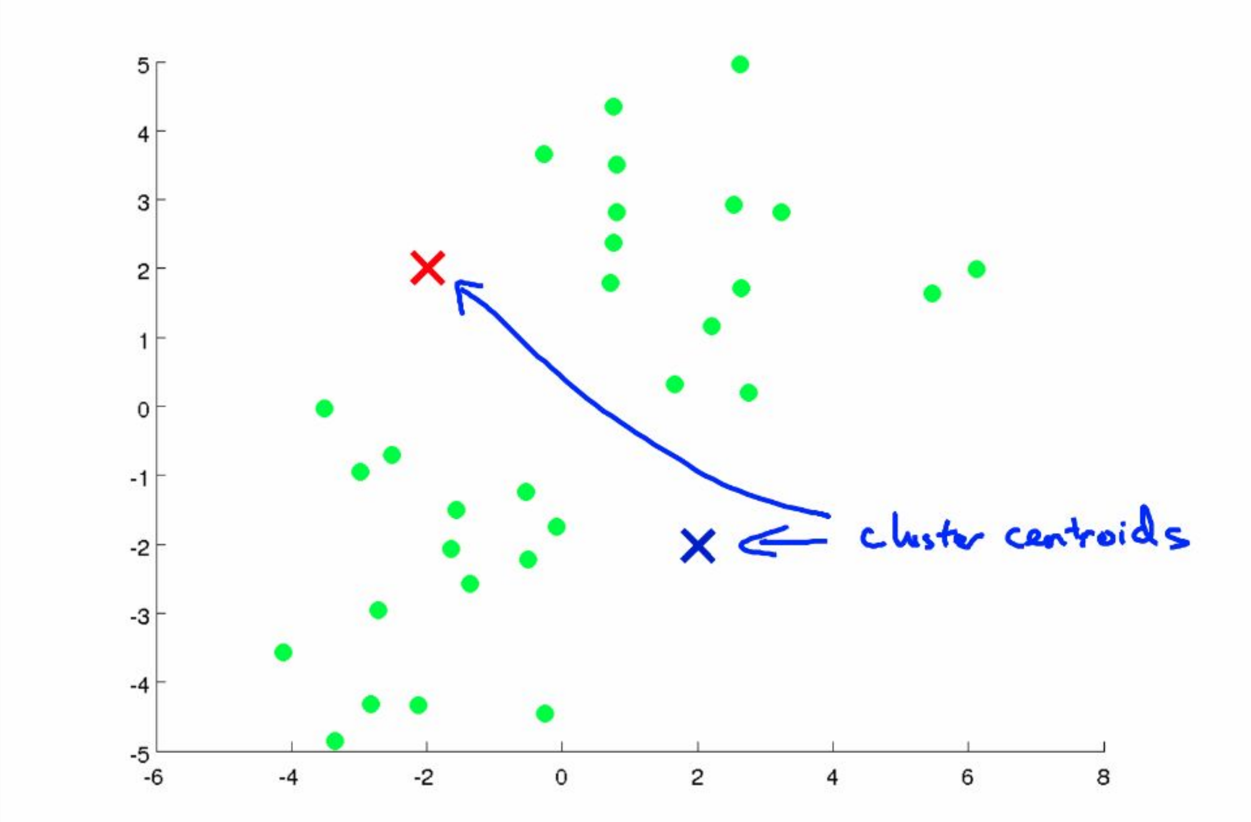
Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

K-Means Clustering



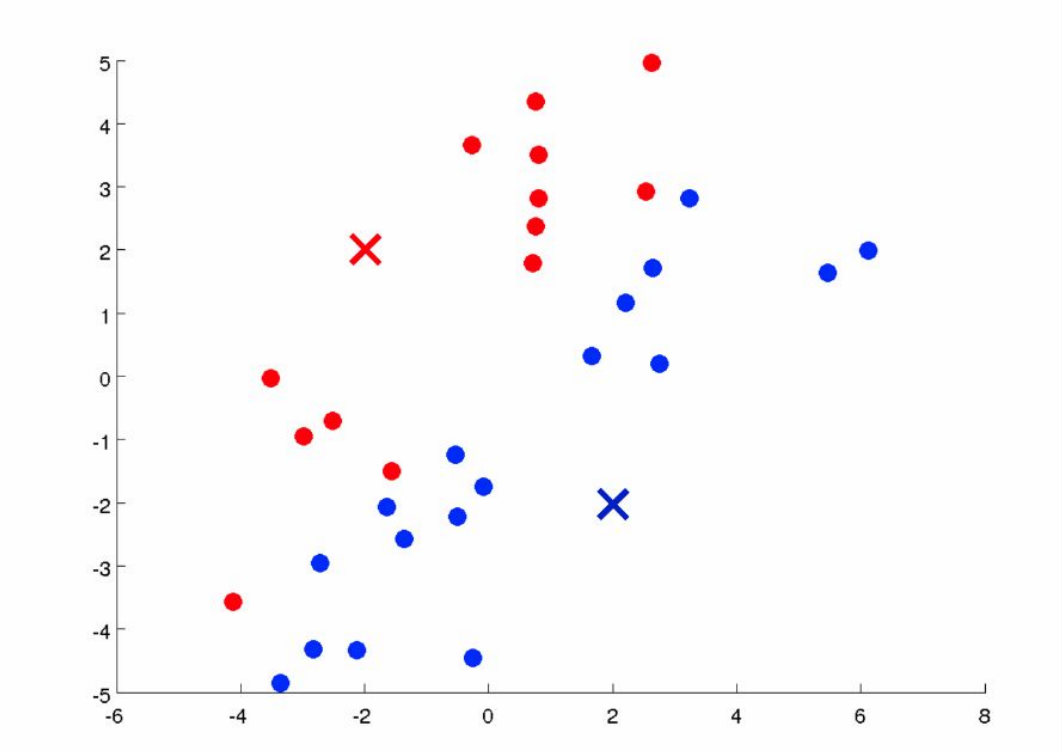
Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

K-means Clustering



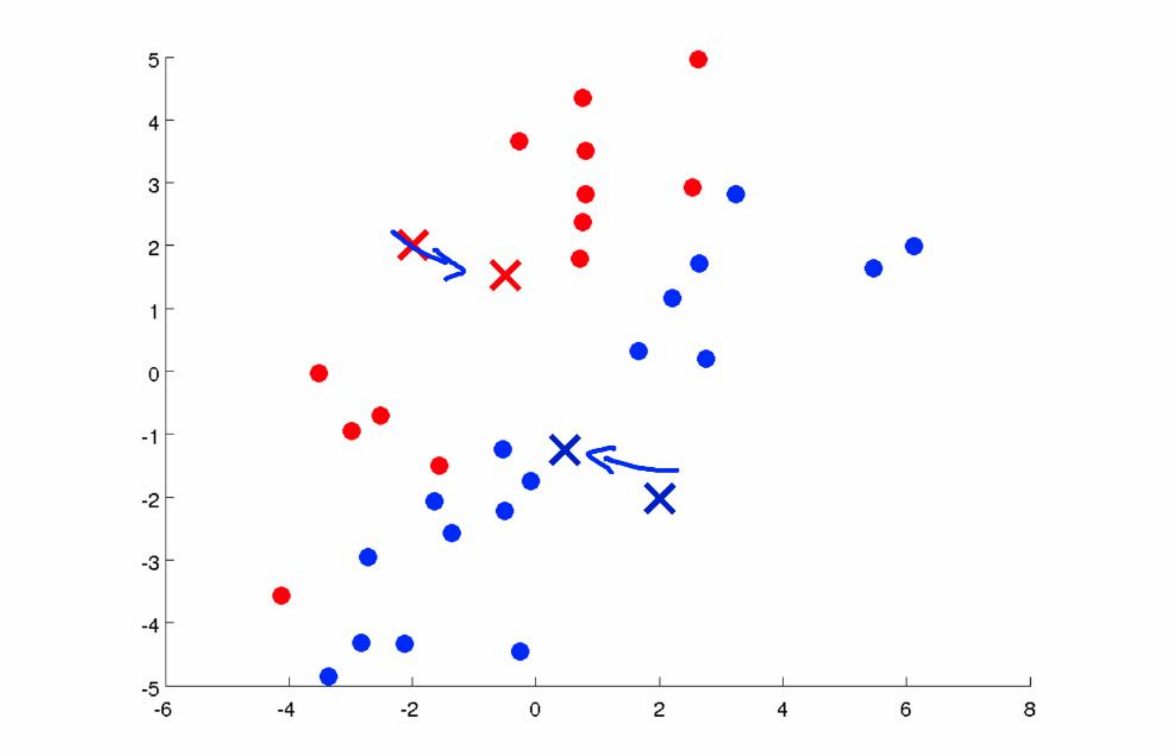
Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

K-means Clustering



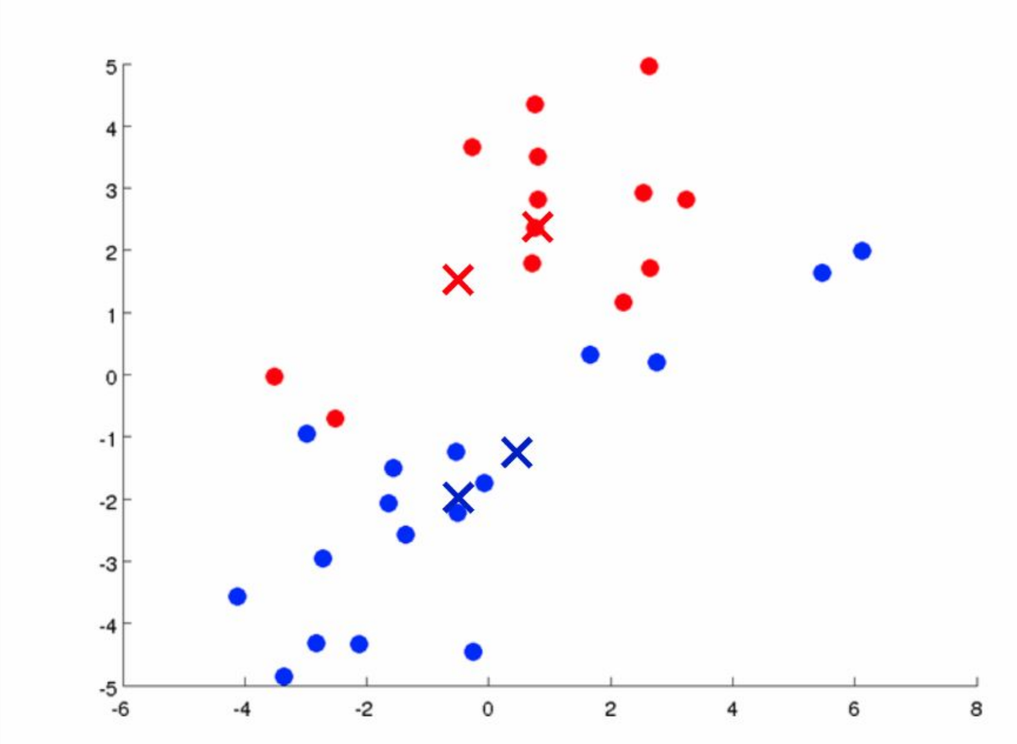
Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

K-means Clustering



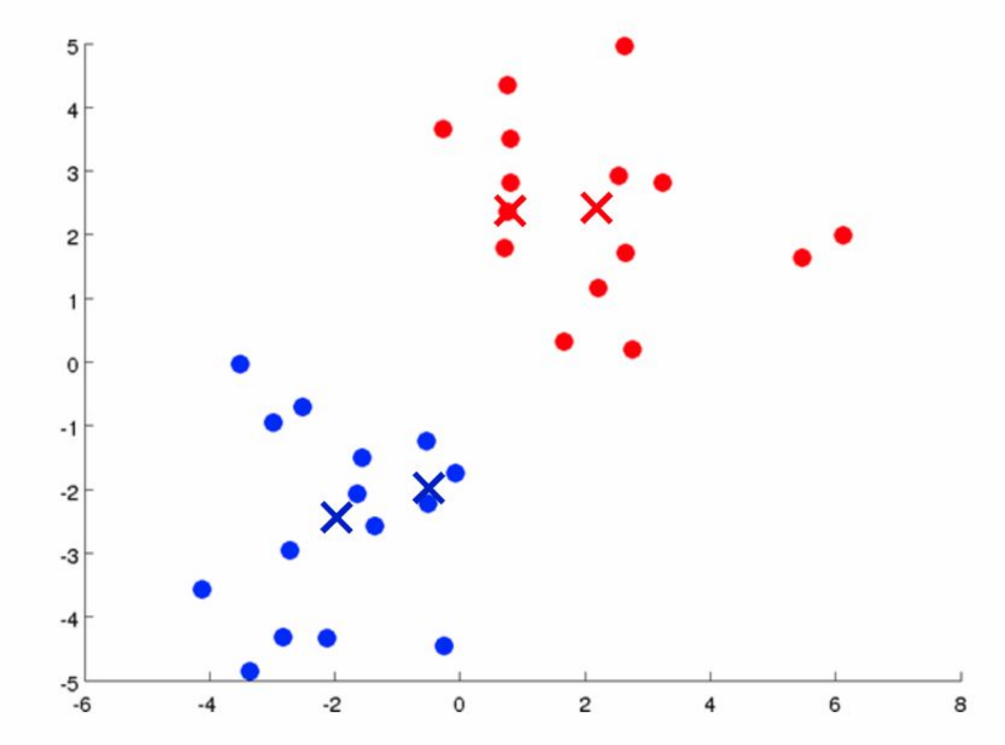
Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

K-means Clustering



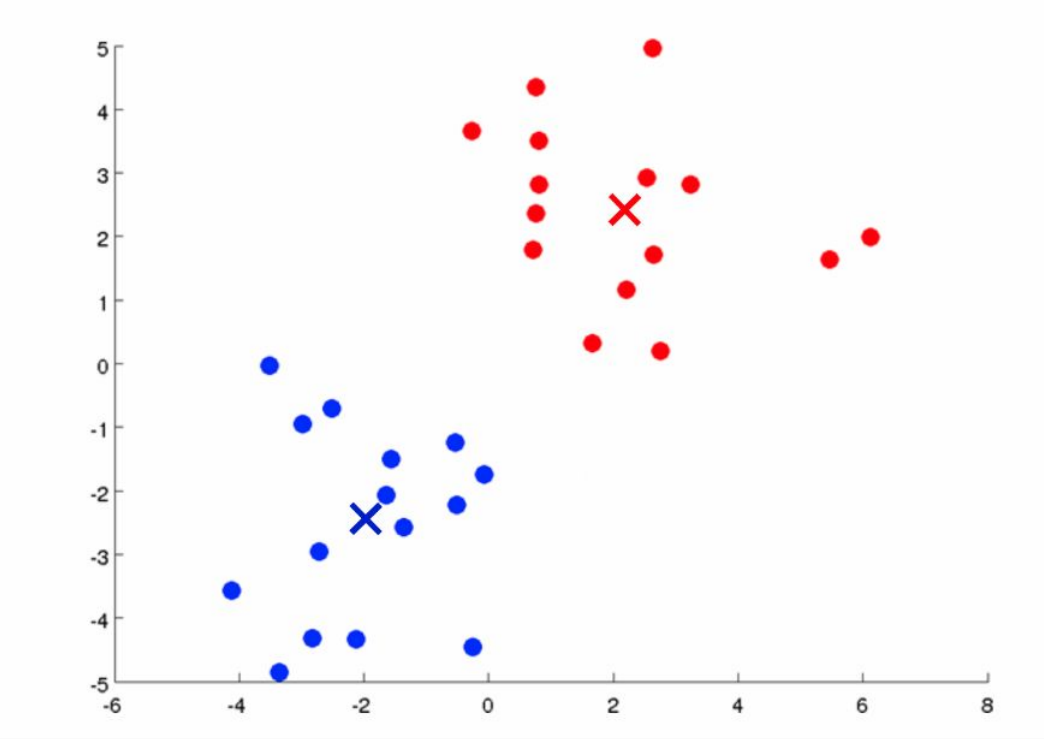
Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

K-means Clustering



Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

K-means Clustering



Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

K-means Clustering

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)

Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

K-means Clustering

K-means algorithm

μ_1 $\times$ μ_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

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

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

}

Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

K-means Clustering

K-means optimization objective

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

→ μ_k = cluster centroid $\underline{k}$ ($\mu_k \in \mathbb{R}^n$) K $k \in \{1, 2, \ldots, K\}$

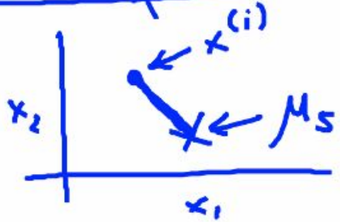
$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned $x^{(i)} \rightarrow \underline{5}$ $\underline{c^{(i)} = 5}$ $\underline{\mu_{c^{(i)}}} = \mu_5$

Optimization objective:

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

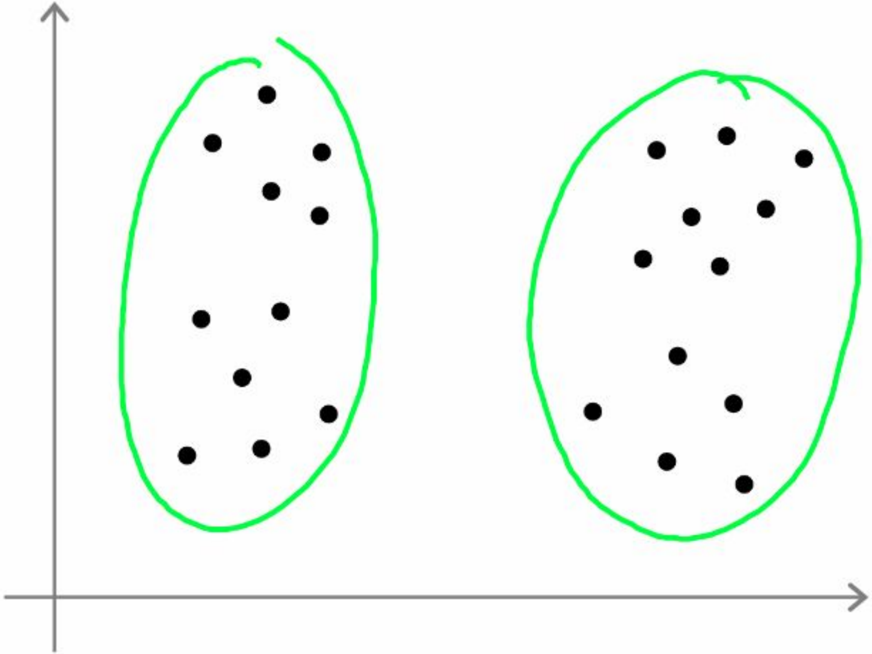
$$\min_{\substack{\rightarrow c^{(1)}, \ldots, c^{(m)}, \\ \rightarrow \mu_1, \ldots, \mu_K}} J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$$

Distortion



Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

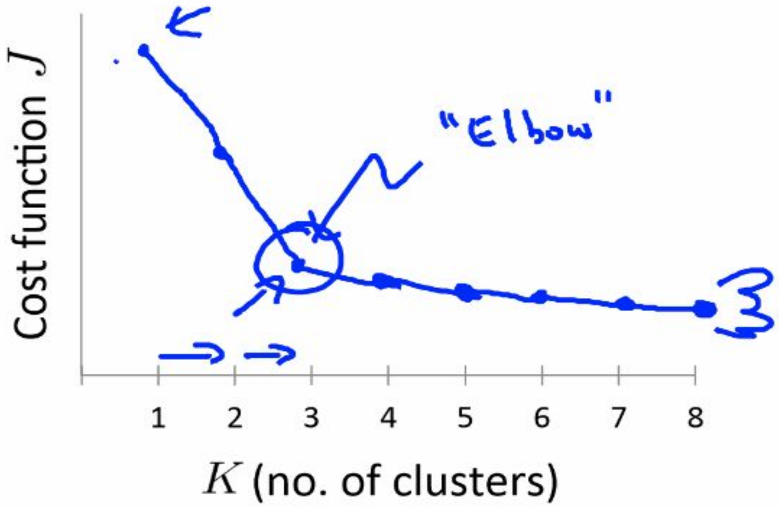
What is the right value of K?

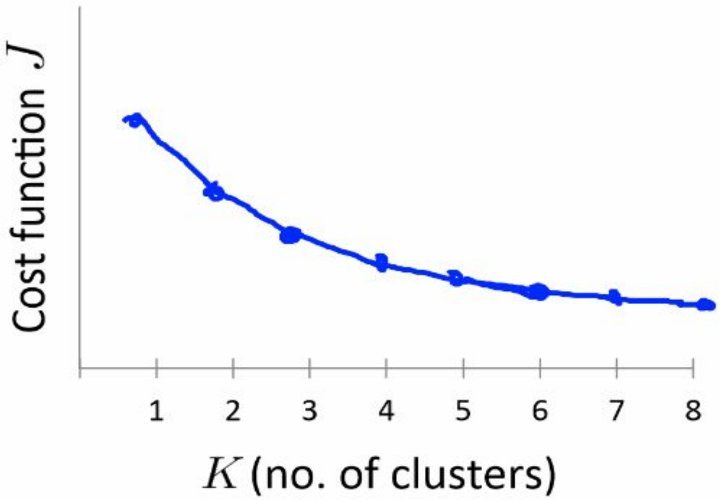


Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

Choosing the value of K

Elbow method:





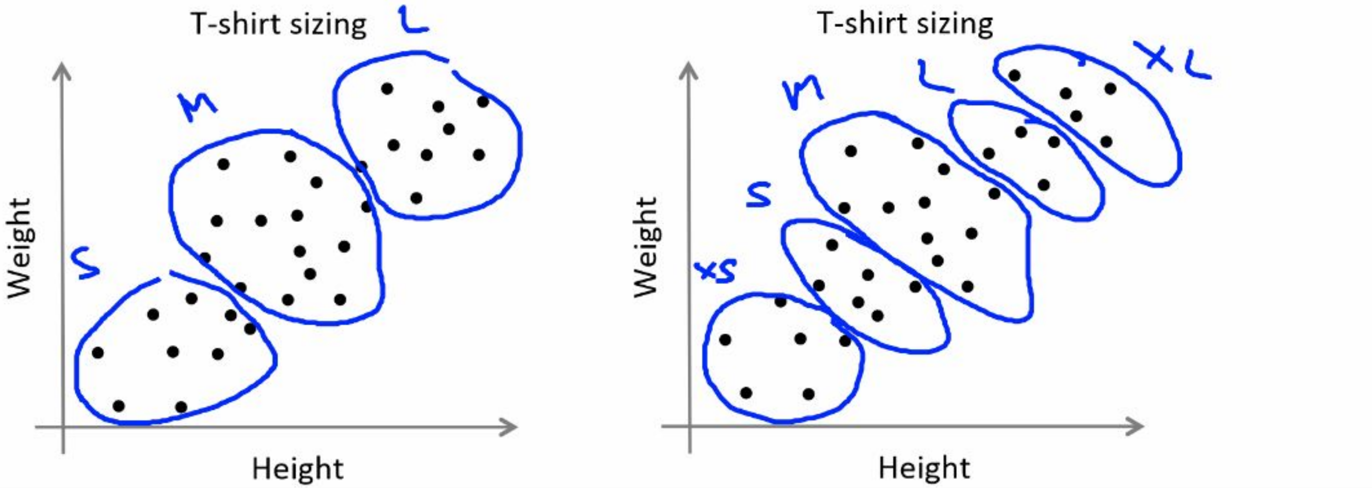
Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

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

E.g.



Credit: Slides from Andrew Ng's Coursera ML Course: lecture 13
https://github.com/vkosuri/CourseraMachineLearning/blob/master/home/week-8/lectures/pdf/Lecture13.pdf

Silhouette Score

sklearn.metrics.silhouette_score

`sklearn.metrics.silhouette_score`(*X*, *labels*, *, *metric='euclidean'*, *sample_size=None*, *random_state=None*, ***kwds*) [source]

Compute the mean Silhouette Coefficient of all samples.

The Silhouette Coefficient is calculated using the mean intra-cluster distance (`a`) and the mean nearest-cluster distance (`b`) for each sample. The Silhouette Coefficient for a sample is `(b - a) / max(a, b)`. To clarify, `b` is the distance between a sample and the nearest cluster that the sample is not a part of. Note that Silhouette Coefficient is only defined if number of labels is 2 <= n_labels <= n_samples - 1.

This function returns the mean Silhouette Coefficient over all samples. To obtain the values for each sample, use `silhouette_samples`.

The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar.

Use-cases

Inputs

> What topics would you like to see more working examples?

> Ideas/presentation on projects?

> What topics could be covered in coming class?

> Any ML work you are currently leveraging in your project?