Clustering and the K-means Algorithm

Lecture 11 - DAMLF | ML1

Ingredients:

Task *to perform*

Type of **Experience**

Performance *measure*

What is a Learning Algorithm?

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"

–Tom Mitchell

Types of Learning Algorithms

Different Experiences with Data

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

Machine learning algorithms are grouped (roughly) into three categories, each corresponding to what type of **experience** an algorithm has with data.

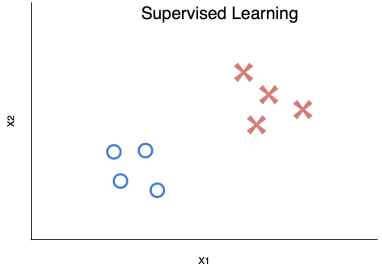
Supervised Learning

Supervised learning problems

Involve datasets where each training example is labeled.

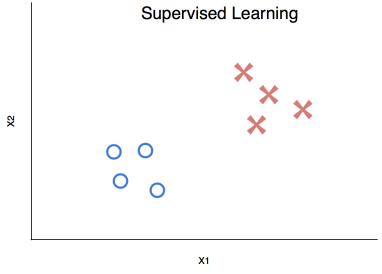
The goal is to fit a hypothesis to data.

Supervised Learning



Example: Classification Problems

Supervised Learning



Example: Classification Problems

Training set:

$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$

Goal:

Find **decision boundary** which separates positive labeled examples ($y = 1$) from negative labeled examples ($y = 0$)

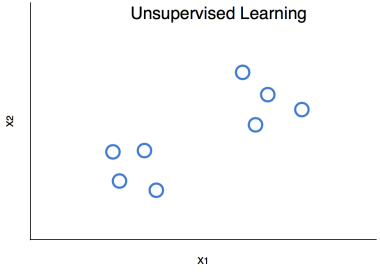
Unsupervised Learning

Unsupervised learning problems

Involve unlabeled datasets.

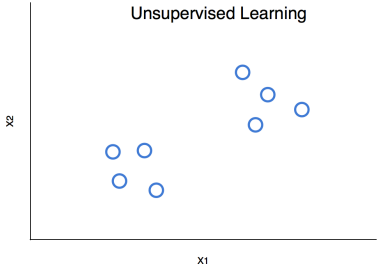
The goal is to find "some pattern" in data.

Unsupervised Learning



Example: Clustering Problems

Unsupervised Learning



Example: Clustering Problems

Training set:

$\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

Goal:

Find clusters within the data

Applications of Clustering

Deploying police patrols

Applications of Clustering

Portfolio Diversification

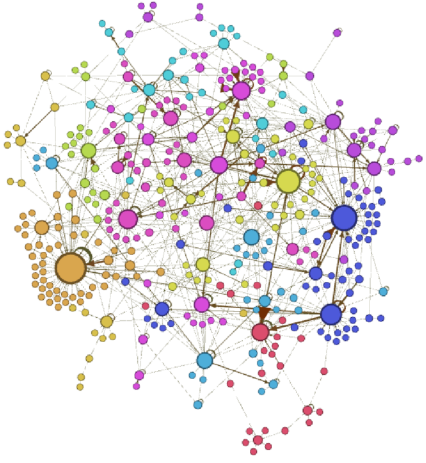
Diversifying assets is the most effective way to ensure low risk-reward ratios.

Karina Marvin (2015, Princeton TR) has proposed a measure of similarity that places companies in the same cluster which have a similar average of two ratios,

$$\frac{\text{Revenues}}{\text{Assets}} \quad \text{and} \quad \frac{\text{Net Income}}{\text{Assets}}$$

G. Kudesia's iPython notebook implementation:
https://github.com/garvit-kudesia91/kmeans-portfolio

Applications of Clustering

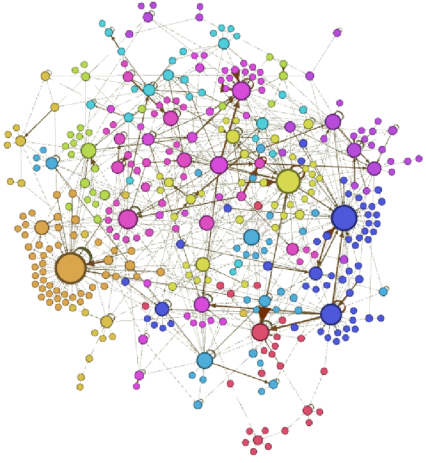


Diffusion Networks

Disease Transmission

scale-free networks

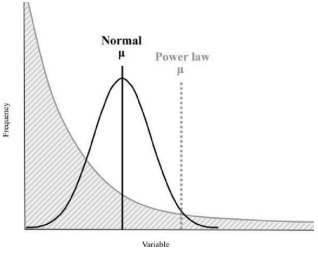
Applications of Clustering



Diffusion Networks

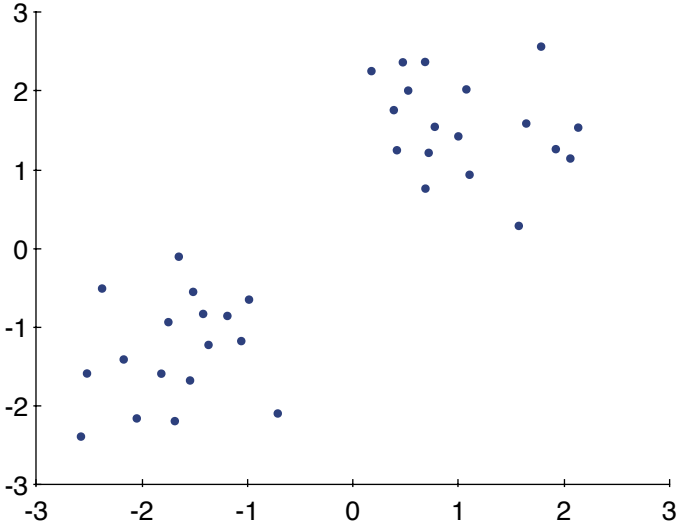
Disease Transmission

scale-free networks



K-means algorithm

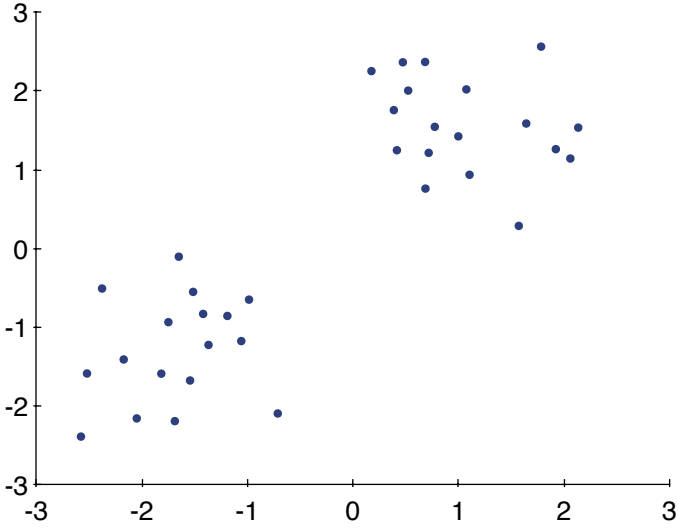
K-means Algorithm



Goal

Group unlabeled data into "coherent" clusters.

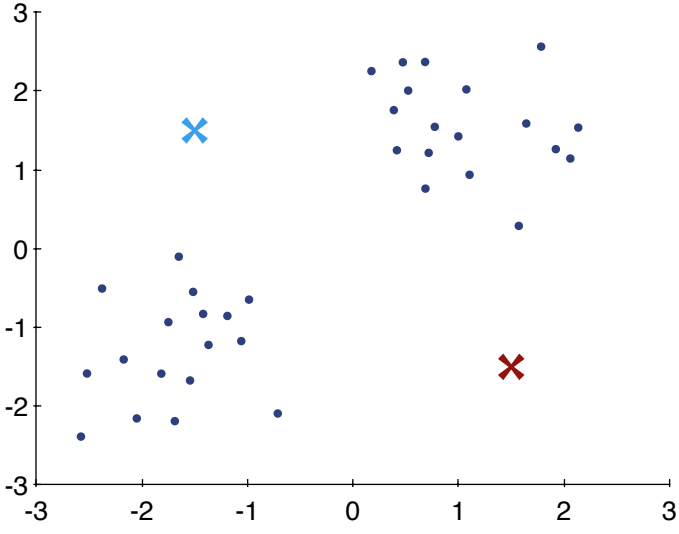
K-means Algorithm



Intuition

- Start with a data set

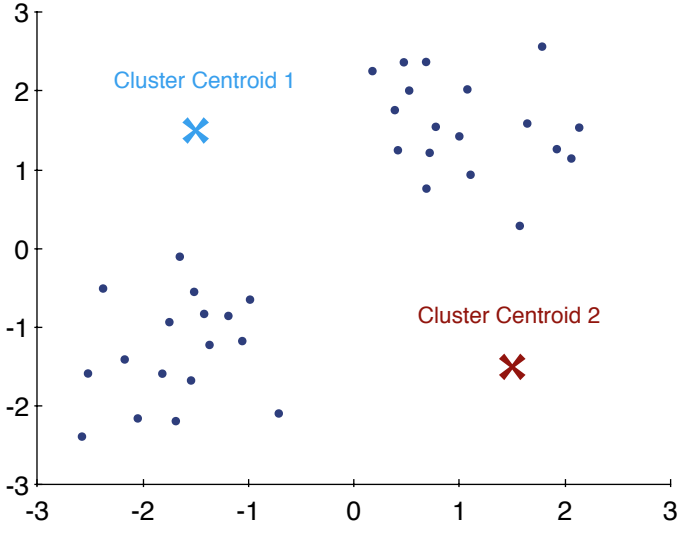
K-means Algorithm



Intuition

- Start with a data set
- Initialize centroids

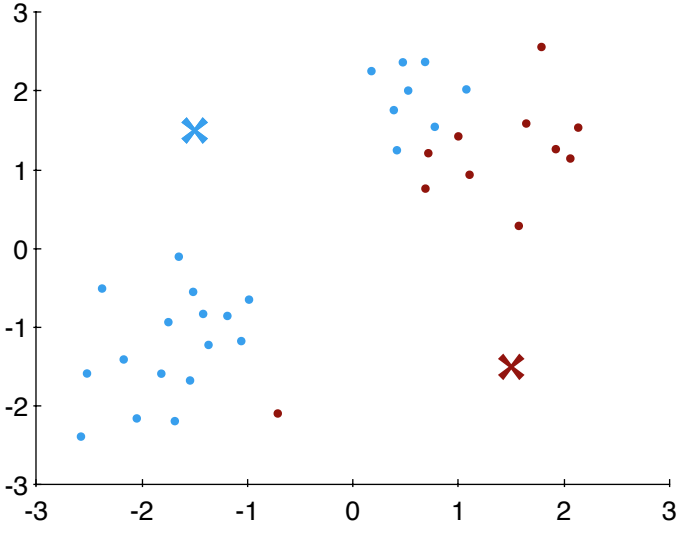
K-means Algorithm



Intuition

- Start with a data set
- Initialize centroids

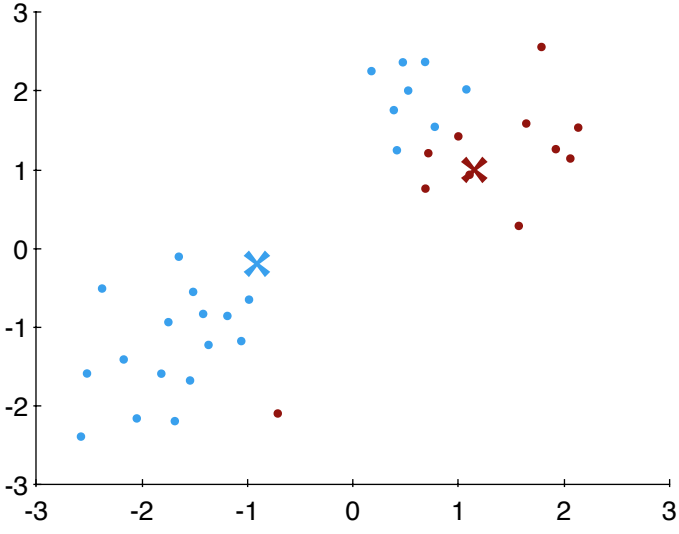
K-means Algorithm



Intuition

- Start with a data set
- Initialize centroids
- Nearest cluster step:
  assign data to nearest centroid

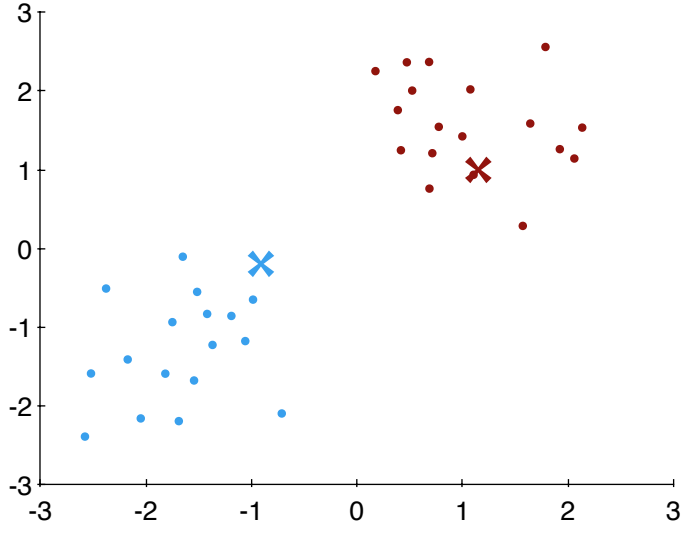
K-means Algorithm



Intuition

- Start with a data set
- Initialize centroids
- Nearest cluster step:
  assign data to nearest centroid
- Move centroids step
  by computing the mean

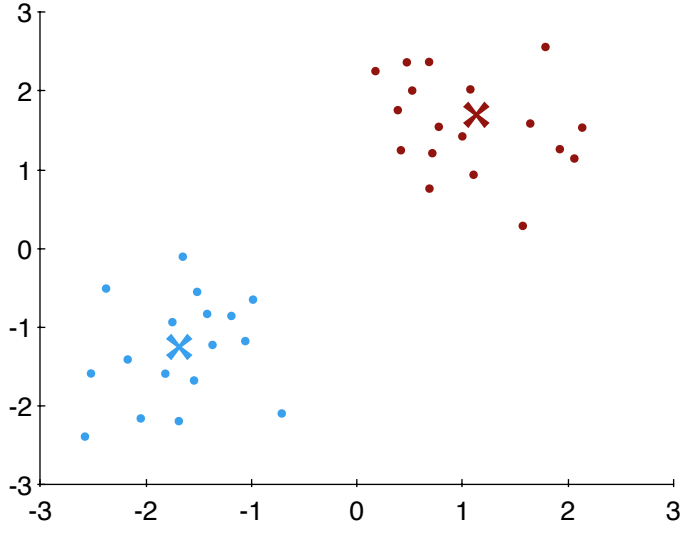
K-means Algorithm



Intuition

- Start with a data set
- Initialize centroids
- Nearest cluster step:
  assign data to nearest centroid
- Move centroids step
  by computing the mean
- Nearest cluster step

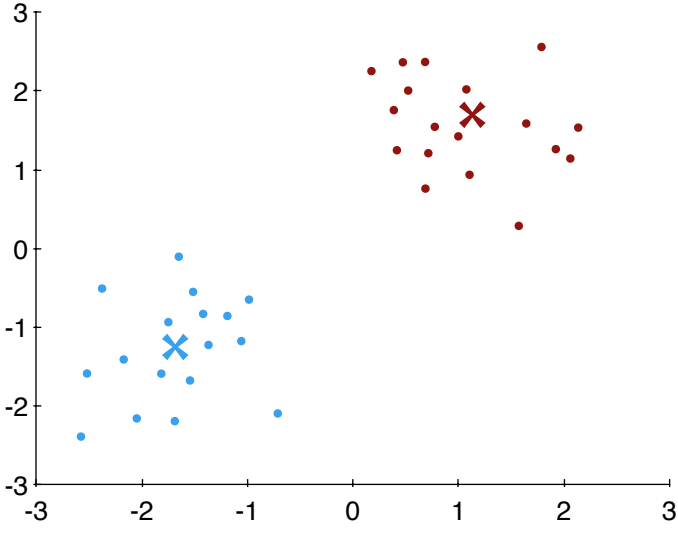
K-means Algorithm



Intuition

- Start with a data set
- Initialize centroids
- Nearest cluster step:
  assign data to nearest centroid
- Move centroids step
  by computing the mean
- Nearest cluster step
- Move centroids step

K-means Algorithm



Intuition

- Start with a data set
- Initialize centroids
- Nearest cluster step:
  assign data to nearest centroid
- Move centroids step
  by computing the mean
- Nearest cluster step
- Move centroids step
- Nearest cluster
  (*same subsets*)
- Move centroids (*no change*)

K-means Algorithm

K-means Input

K # Number of clusters

X_{train} # Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$.

Note that $x^{(i)} \in \mathbb{R}^n$; we do **not** have $x_0 = 1$

K-means Algorithm

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

Repeat:

for i **in** range(1, m):

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

for k **in** range(1, K):

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

K-means Algorithm

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

Repeat:

for i **in** range(1, m):

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

Cluster assignment step.

To compute $c^{(i)}$, first $\|x^{(i)} - \mu_k\|^2$, for $k \in \{1, 2, \ldots, K\}$;
then $c^{(i)} = k$ for $\min_k \|x^{(i)} - \mu_k\|^2$

for k **in** range(1, K):

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

K-means Algorithm

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

Repeat:

for i **in** range(1, m):

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

Cluster assignment step.

To compute $c^{(i)}$, first $\|x^{(i)} - \mu_k\|^2$, for $k \in \{1, 2, \ldots, K\}$;
then $c^{(i)} = k$ for $\min_k \|x^{(i)} - \mu_k\|^2$

for k **in** range(1, K):

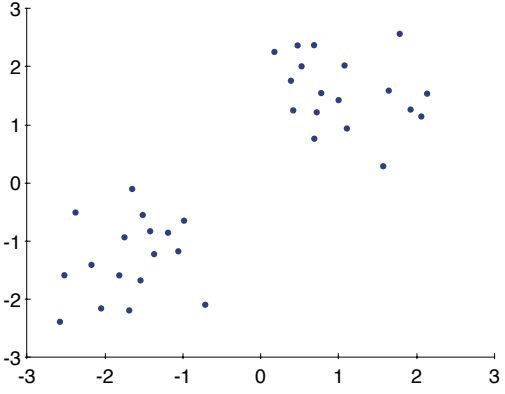
$\mu_k :=$ average (mean) of points assigned to cluster k

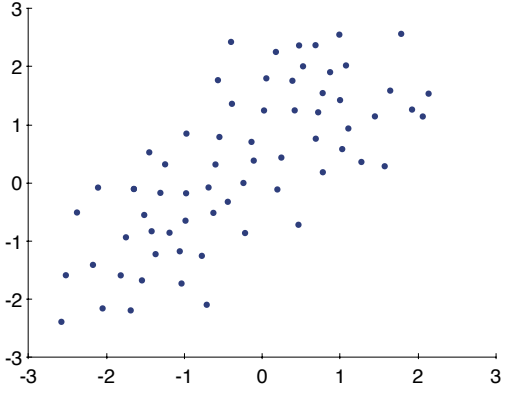
Move centroid step.

Compute the average $\mu_k = \frac{1}{r}\left(\sum_{j=1}^{r} x^{(r)}\right)$, where μ_k is of dimension $\mathbb{R}^n$

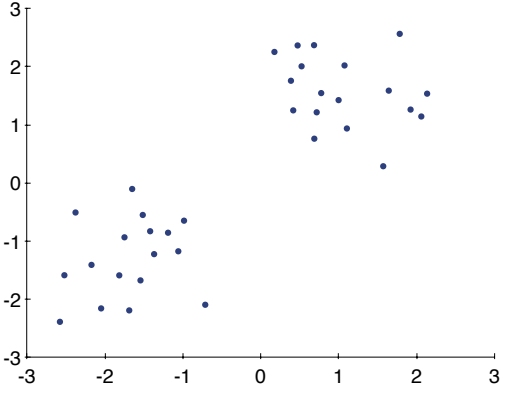
Example:
Suppose $c^{(3)} = 2, c^{(7)} = 2, c^{(11)} = 2$ and $|\{x^{(i)} : c^{(i)} = 2\}| = 3$;
then: $\mu_2 = \frac{1}{3}\left(x^{(3)} + x^{(7)} + x^{(11)}\right)$, where $\mu_2 \in \mathbb{R}^n$

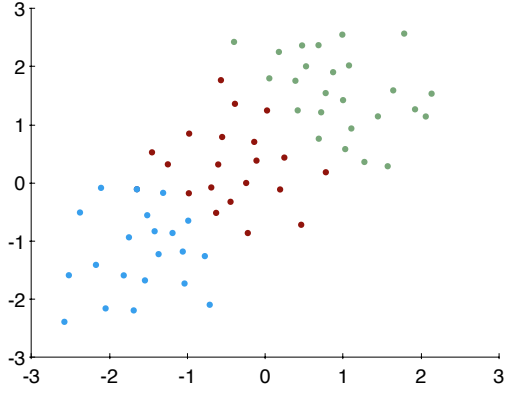
Separated vs Non-separated Clusters





Separated vs Non-separated Clusters





K-means Optimization Objective

K-means Cost Function

Notation:

$c^{(i)}$:= index of the cluster centroid ($c = 0, 1, \ldots, K$) that $x^{(i)}$ currently assigned to $x^{(i)}$
$c^{(i)}$ is an integer

μ_k := the current location (in $\mathbb{R}^n$) of the cluster centroid k
μ_k is a vector in $\mathbb{R}^n$

K-means Cost Function

Notation:

$c^{(i)}$:= index of the cluster centroid ($c = 0, 1, \ldots, K$) that $x^{(i)}$ currently assigned to $x^{(i)}$
$c^{(i)}$ is an integer

μ_k := the current location (in $\mathbb{R}^n$) of the cluster centroid k
μ_k is a vector in $\mathbb{R}^n$

$\mu_{c^{(i)}}$:= the cluster centroid (in $\mathbb{R}^n$) to which example $x^{(i)}$ is currently assigned.
$\mu_{c^{(i)}}$ is a vector in $\mathbb{R}^n$

K-means Cost Function

Notation:

$c^{(i)}$:= index of the cluster centroid ($c = 0, 1, \ldots, K$) that $x^{(i)}$ currently assigned to $x^{(i)}$
$c^{(i)}$ is an integer

μ_k := the current location (in $\mathbb{R}^n$) of the cluster centroid k
μ_k is a vector in $\mathbb{R}^n$

$\mu_{c^{(i)}}$:= the cluster centroid (in $\mathbb{R}^n$) to which example $x^{(i)}$ is currently assigned.
$\mu_{c^{(i)}}$ is a vector in $\mathbb{R}^n$

Example:
if $x^{(i)}$ is assigned to cluster 3, then $c^{(i)} = 3$;
therefore, $\mu_{c^{(i)}} = \mu_3$.

K-means Cost Function

Notation:

$c^{(i)}$:= index of the cluster centroid ($c = 0, 1, \ldots, K$) that $x^{(i)}$ currently assigned to $x^{(i)}$
$c^{(i)}$ is an integer

μ_k := the current location (in $\mathbb{R}^n$) of the cluster centroid k
μ_k is a vector in $\mathbb{R}^n$

$\mu_{c^{(i)}}$:= the cluster centroid (in $\mathbb{R}^n$) to which example $x^{(i)}$ is currently assigned.
$\mu_{c^{(i)}}$ is a vector in $\mathbb{R}^n$

Example:
if $x^{(i)}$ is assigned to cluster 3, then $c^{(i)} = 3$;
therefore, $\mu_{c^{(i)}} = \mu_3$.

K-means Cost Function:

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

where $\left\| x^{(i)} - \mu_{c^{(i)}} \right\|^2$ is **squared Euclidean distance**, not Euclidean distance.

Squared Euclidean Distance

Euclidean Distance: $d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$

Squared Euclidean Distance: $d^2 = \sum_{i=1}^{n} (x_i - y_i)^2$

Squared Euclidean Distance

Euclidean Distance: $d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$

Squared Euclidean Distance: $d^2 = \sum_{i=1}^{n} (x_i - y_i)^2$

Technical note:

Squared Euclidean distance is **not** a metric since it does not satisfied the triangle inequality.

However, for K-means, there is **no difference in outcomes** between using Euclidean distance or squared Euclidean distance, but the latter is computationally more efficient.

Some clustering algorithms (e.g., hierarchical clustering) require the properties of a metric (e.g., Euclidean distance).

K-means Optimization Objective

K-means Distortion (Cost Function):

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

Optimization objective:

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

K-means Algorithm

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

Repeat:

for i **in** range(1, m):

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

for k **in** range(1, K):

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

K-means Algorithm

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

Repeat:

for *i* **in** range(1, m):

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

Cluster assignment step.

Minimize distortion function $J(\cdots)$ wrt $c^{(1)}, \ldots, c^{(m)}$ while holding $\mu_1, \ldots, \mu_K$ fixed.

for *k* **in** range(1, K):

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

K-means Algorithm

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

Repeat:

for *i* **in** range(1, m):

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

Cluster assignment step.

Minimize distortion function $J(\cdots)$ wrt $c^{(1)}, \ldots, c^{(m)}$ while holding $\mu_1, \ldots, \mu_K$ fixed.

for *k* **in** range(1, K):

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

Move centroid step.

Minimize distortion function $J(\cdots)$ wrt $\mu_1, \ldots, \mu_K$ while holding $c^{(1)}, \ldots, c^{(m)}$ fixed.

K-means Algorithm

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

Repeat:

```
for i in range(1, m):
  c^(i) := index (from 1 to K) of cluster centroid closest to x^(i)
  # Cluster assignment step.
  # Minimize distortion function J(···) wrt c^(1), ..., c^(m) while holding
    μ_1, ..., μ_K fixed.

for k in range(1, K):
  μ_k := average (mean) of points assigned to cluster k
  # Move centroid step.
  # Minimize distortion function J(···) wrt μ_1, ..., μ_K while holding
    c^(1), ..., c^(m) fixed.
```

Random Initialization

K-means Algorithm

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

Repeat:

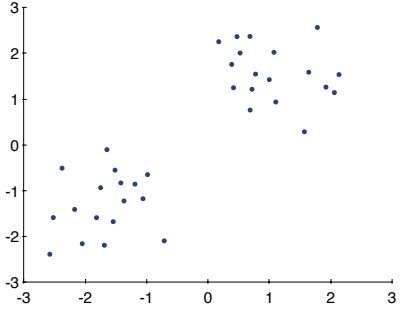
for *i* **in** range(1, m):

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

for *k* **in** range(1, K):

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

Random Initialization



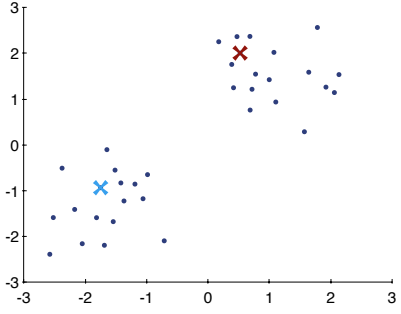
Implementation

Should set $K < m$

Randomly pick K training examples

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

Random Initialization



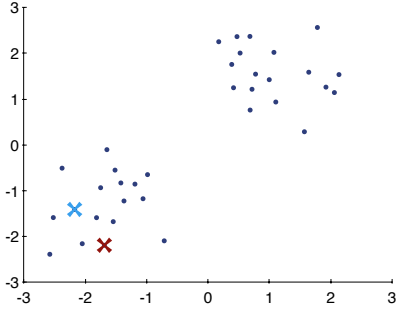
Implementation

Should set $K < m$

Randomly pick K training examples

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

Random Initialization



Implementation

Should set $K < m$

Randomly pick K training examples

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

Random Initialization

Implementation

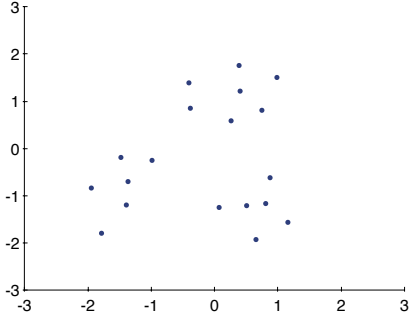
Should set $K < m$

Randomly pick K training examples

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

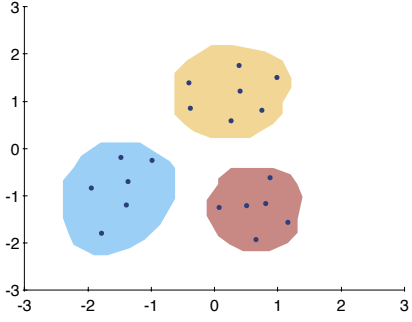
However, depending on the initial randomization, K-means can end up with different solutions

Local Optima



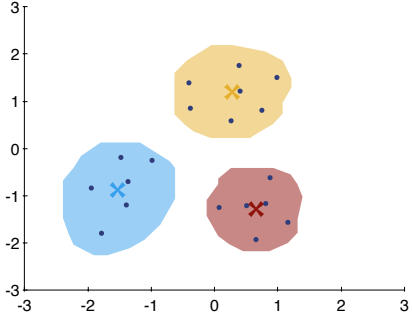
However, depending on the initial randomization, K-means can end up with *different solutions*

Local Optima



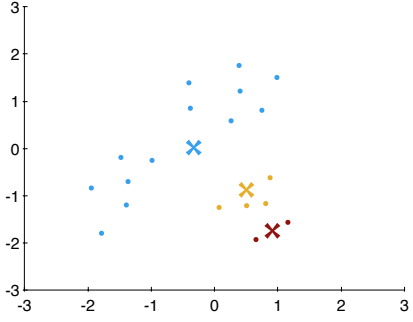
However, depending on the initial randomization, K-means can end up with *different solutions*

Local Optima



However, depending on the initial randomization, K-means can end up with *different solutions*

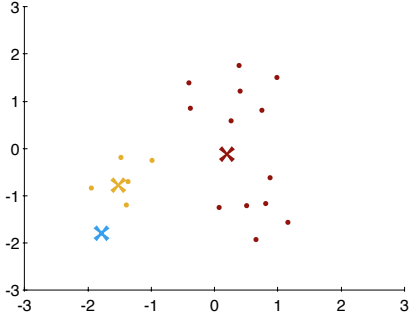
Local Optima



However, depending on the initial randomization, K-means can end up with *different solutions*

In particular, K-means can end up at **local optima**

Local Optima

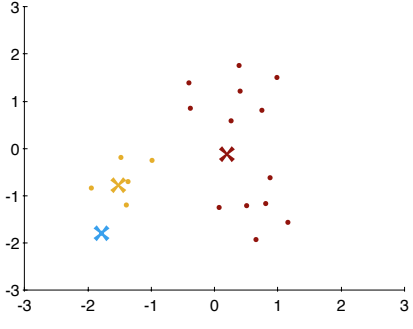


However, depending on the initial randomization, K-means can end up with *different solutions*

In particular, K-means can end up at **local optima**

'Local optima' refers to local minima of the distortion function $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_k)$

Local Optima



However, depending on the initial randomization, K-means can end up with *different solutions*

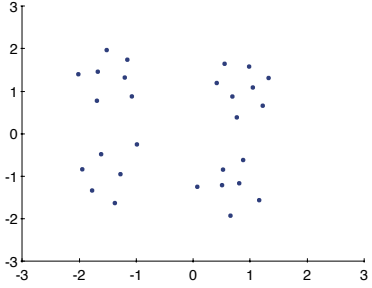
In particular, K-means can end up at **local optima**

'Local optima' refers to local minima of the distortion function $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_k)$

Local minima is an issue for small clusters (e.g., $K = 2$ to 10), but becomes less of an issue for $K > 10$)

Choosing K

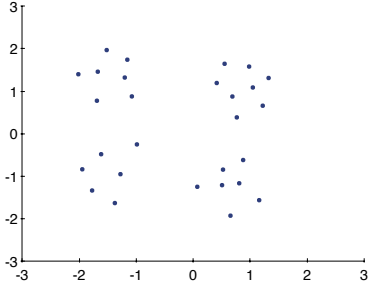
What is the right value for K?



Ambiguity

People often choose the number of clusters by hand.

What is the right value for K?

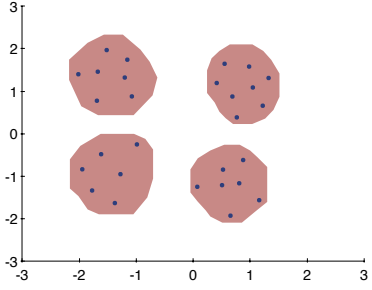


Ambiguity

People often choose the number of clusters by hand.

However, sometimes this can be ambiguous.

What is the right value for K?



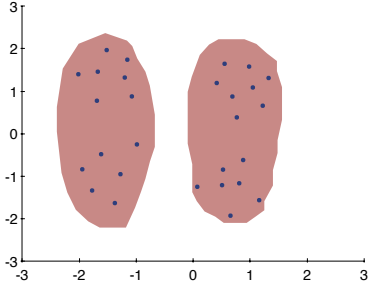
Ambiguity

People often choose the number of clusters by hand.

However, sometimes this can be ambiguous.

- Some might see $K = 4$

What is the right value for K?



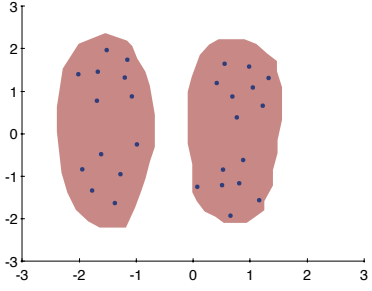
Ambiguity

People often choose the number of clusters by hand.

However, sometimes this can be ambiguous.

- Some might see $K = 4$
- Others might see $K = 2$

What is the right value for K?



Ambiguity

People often choose the number of clusters by hand.

However, sometimes this can be ambiguous.

- Some might see $K = 4$
- Others might see $K = 2$

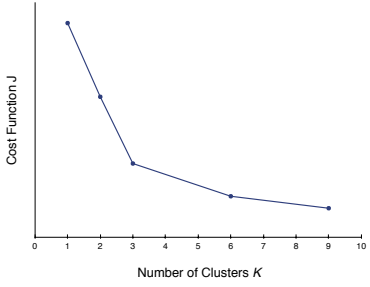
The ambiguity problem is inherent in clustering, since this is an ***unsupervised learning*** *problem; we are not given labeled data.*

What is the right value for K?

Elbow Method

Vary the number of clusters, plotting the distortion (cost function) J for each value of K.

What is the right value for K?



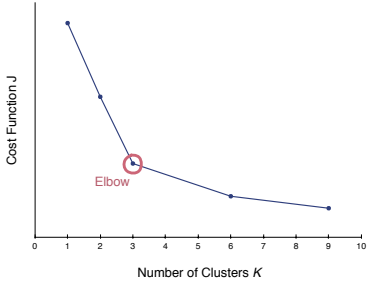
Elbow Method

Vary the number of clusters, plotting the distortion (cost function) J for each value of K.

- Run K-means for $K = 1$, then plot J
- Run K-means for $K = 2$, then plot J
- Run K-means for $K = 3$, then plot J

⋮

What is the right value for K?



Elbow Method

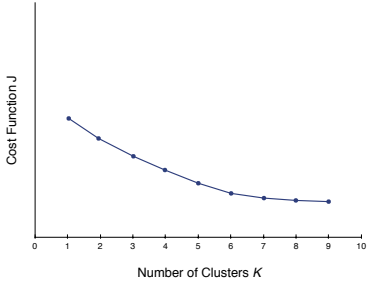
Vary the number of clusters, plotting the distortion (cost function) J for each value of K.

- Run K-means for $K = 1$, then plot J
- Run K-means for $K = 2$, then plot J
- Run K-means for $K = 3$, then plot J

$\vdots$

The **elbow method** looks at the plot of J vs K, and picks the value of K after which the rapid improvement in reducing J transitions (if at all) to more gradual improvements.

What is the right value for K?



Elbow Method

Vary the number of clusters, plotting the distortion (cost function) J for each value of K.

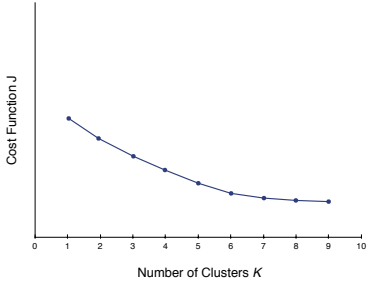
- Run K-means for $K = 1$, then plot J
- Run K-means for $K = 2$, then plot J
- Run K-means for $K = 3$, then plot J

$\vdots$

The **elbow method** looks at the plot of J vs K, and picks the value of K after which the rapid improvement in reducing J transitions (if at all) to more gradual improvements.

However, just as often there isn't a sharp elbow.

What is the right value for K?



Elbow Method

Vary the number of clusters, plotting the distortion (cost function) J for each value of K.

- Run K-means for $K = 1$, then plot J
- Run K-means for $K = 2$, then plot J
- Run K-means for $K = 3$, then plot J

$\vdots$

The **elbow method** looks at the plot of J vs K, and picks the value of K after which the rapid improvement in reducing J transitions (if at all) to more gradual improvements.

However, just as often there isn't a sharp elbow.

Take-away: *The elbow method is worth a shot when there is an unambiguous "elbow", but it doesn't always yield a clear unambiguous answer.*

What is the right value for K?

Pragmatic Considerations

Given a value for *K*, how well does this suit my purpose?

What is the right value for K?

Pragmatic Considerations

Given a value for *K*, how well does this suit my purpose?

How you answer this question, can provide you with an evaluation criteria for judging the appropriate number of clusters *K* to use.

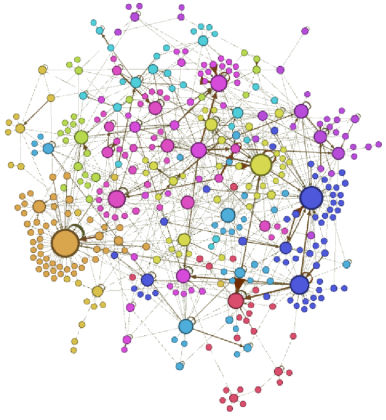
What is the right value for K?

Pragmatic Considerations

Given a value for *K*, how well does this suit my purpose?

How you answer this question, can provide you with an evaluation criteria for judging the appropriate number of clusters *K* to use.

What is the right value for K?



Pragmatic Considerations

Given a value for *K*, how well does this suit my purpose?

How you answer this question, can provide you with an evaluation criteria for judging the appropriate number of clusters *K* to use.

References