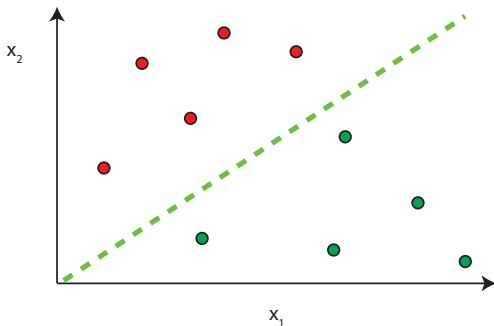


Overview

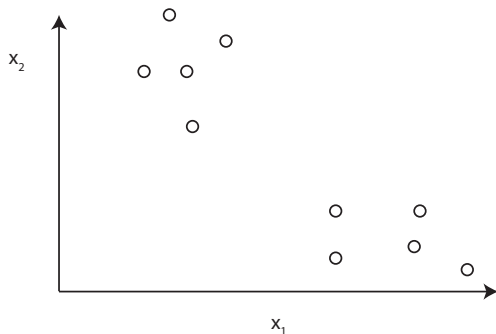
- Supervised vs Unsupervised Learning
- Clustering: k -means algorithm

Supervised Learning



- Training data: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$
- Training data is labelled i.e. we know $y^{(1)}, y^{(2)}$ etc

Unsupervised Learning



- Training data: $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$
- Training data is unlabelled i.e. we do not know $y^{(1)}, y^{(2)}$ etc
- We need algorithms that try to cluster the training data ...

Applications

- Google News

The screenshot shows the Google News interface in a web browser. The address bar displays the URL: https://news.google.com/news/headlines/section/topic/WORLD.en_ie/World?ned=en_ie&hl=en-IE. The page features a search bar at the top with the text "Search". Below the search bar, there are navigation tabs: "Headlines" (selected), "Local", "For You", and "Ireland".

On the left side, there is a "SECTIONS" menu with the following options: "Top Stories", "World" (selected), "Ireland", "Business", "Technology", "Entertainment", "Sport", "Science", "Health", and "Manage sections".

The main content area is titled "World" and displays a large article with the headline "10 Missing After US Navy Ship and Oil Tanker Collide Off Singapore". The article is attributed to "New York Times" and is "55m ago". Below the headline, there is a "RELATED COVERAGE" section with the following items:

- "7 things about US warship USS John S. McCain or 'Big Bad John'" - From Singapore - The Straits Times - 2h ago
- "USS John S. McCain: 10 US sailors missing after destroyer collides with oil tanker" - The Guardian - 2h ago
- "USS Navy ship and oil tanker collide near Singapore" - BBC News - 2h ago
- "Stricken US destroyer arrives in Singapore after collision, 10 sailors missing" - In-depth - Washington Post - 24m ago
- "Ten missing after USS McCain collides with oil tanker near Strait of Malacca" - Aljazeera.com

Below the related coverage, there is a "View full coverage" link with a right arrow. To the right of the main article, there is a "Related" section with the following items:

- Spain
- La Rambla, Barcelona
- Barcelona
- Singapore
- Australia
- Brexit
- Bashar al-Assad
- United Kingdom
- European Union
- Damascus

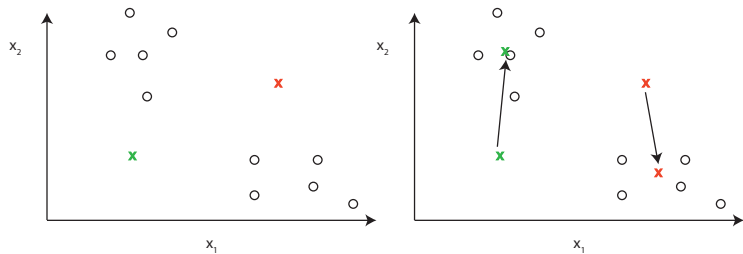
At the bottom of the page, there is another article with the headline "(LEAD) S. Korea urges NK to end provocations as allies start military drills" attributed to "Yonhap News" and "3h ago". Below this, there is a partial view of an article with the headline "Britain has to compromise on trade in agriculture".

Applications

- Fraud detection - try to cluster into normal and anomalous activity based on observed features
- Market segmentation e.g. try to detect customers about to leave a service
- Social network analysis e.g. try to detect communities/groupings



k -means algorithm



k -means algorithm

Input:

- k , number of clusters
- Training data: $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$
- We'll drop the $x_0 = 1$ convention and use x_1, \dots, x_n as elements of x .

Randomly initialise k cluster centres $\mu^{(1)}, \dots, \mu^{(k)}$. e.g. choose k points from training set and use these (need $k < m$).

- Repeat:
 - cluster assignment:**
for $i = 1$ to m ,
 $c^{(i)} :=$ index of cluster centres closest to $x^{(i)}$
 - update centres:**
for $j = 1$ to k
 $\mu^{(j)} :=$ average (mean) of points assigned to cluster j
- Stop when assignments no longer change

k-means algorithm: optimisation objective

$c^{(i)}$ = index of cluster to which example $x^{(i)}$ is assigned

μ_j = centre of cluster j

$\mu_{c^{(i)}}$ = cluster centre to which example $x^{(i)}$ is assigned

$\|x - c\|^2 = \sum_{j=1}^n (x_j - c_j)^2$ (Euclidean distance)

Goal: minimise

$$J(c^{(1)}, \dots, c^{(m)}, \mu^{(1)}, \dots, \mu^{(k)}) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu^{(c^{(i)})}\|^2$$

k -means algorithm: optimisation objective

Goal: minimise

$$J(c^{(1)}, \dots, c^{(m)}, \mu^{(1)}, \dots, \mu^{(k)}) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu^{(c^{(i)})}\|^2$$

- Repeat:

cluster assignment:

for $i = 1$ to m ,

$c^{(i)} :=$ index of cluster centres closest to $x^{(i)}$

i.e. select $c^{(1)}, \dots, c^{(m)}$ to minimise

$$J(c^{(1)}, \dots, c^{(m)}, \mu^{(1)}, \dots, \mu^{(k)})$$

update centres:

for $j = 1$ to k

$\mu_j :=$ average (mean) of points assigned to cluster j

$$= \frac{1}{|C_j|} \sum_{k \in C_j} x^k \text{ where } C_j = \{i : c^{(i)} = j\}$$

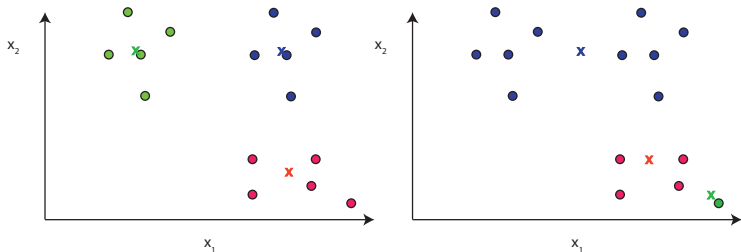
i.e. select $\mu^{(1)}, \dots, \mu^{(k)}$ to minimise

$$J(c^{(1)}, \dots, c^{(m)}, \mu^{(1)}, \dots, \mu^{(k)}) \text{ (a least squares task)}$$

- Stop when assignments no longer change

k -means algorithm: local optima

- k -means algorithm can converge to a local optimum, rather than a global optimum. e.g.



k -means algorithm: local optima

Use random initialisation and multiple runs of algorithm:

for $i = 1$ to 100

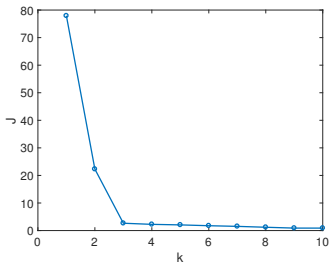
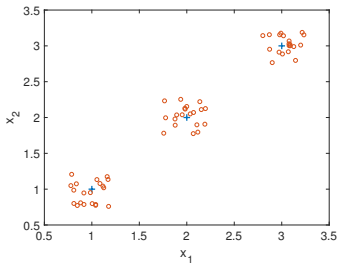
 randomly initialise the k centres $\mu^{(1)}, \dots, \mu^{(k)}$

 run k -means algorithm

 compute cost function $J(c^{(1)}, \dots, c^{(m)}, \mu^{(1)}, \dots, \mu^{(k)})$

Pick clustering that gives lowest cost $J(c^{(1)}, \dots, c^{(m)}, \mu^{(1)}, \dots, \mu^{(jk)})$

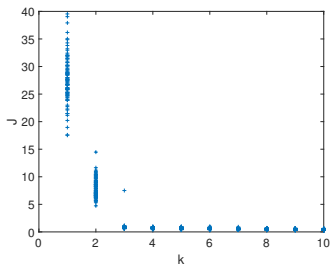
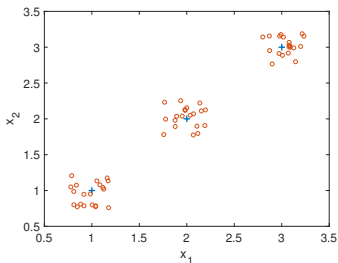
k -means algorithm: choosing the number of clusters



Elbow method:

- Vary k and pick value at “elbow”
- Problem: there might not be an elbow, or at least not a clear one

k -means algorithm: choosing the number of clusters



Cross-validation:

- Randomly select a subset of training data
- run k means algorithm
- Calculate cost $J(c^{(1)}, \dots, c^{(m)}, \mu^{(1)}, \dots, \mu^{(k)})$ for the test data not used for training
- Repeat for multiple random subsets and several values of k .