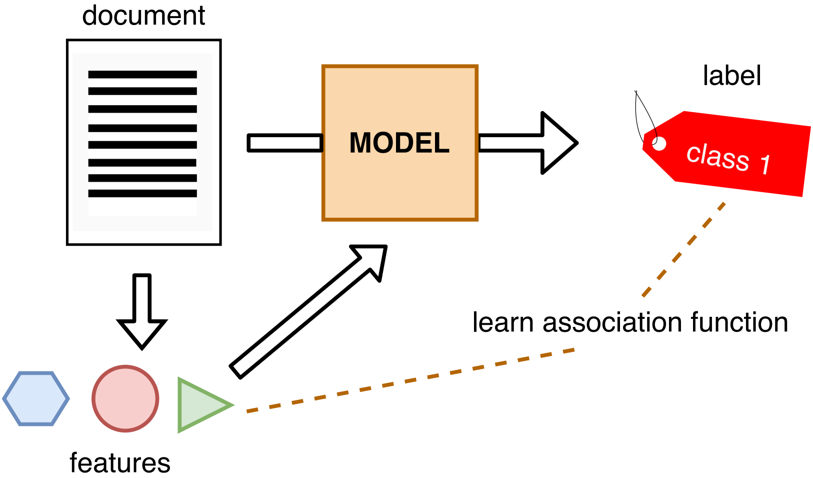
3. Unsupervised Learning: k-means and GMMs

NLP for CogSci Research

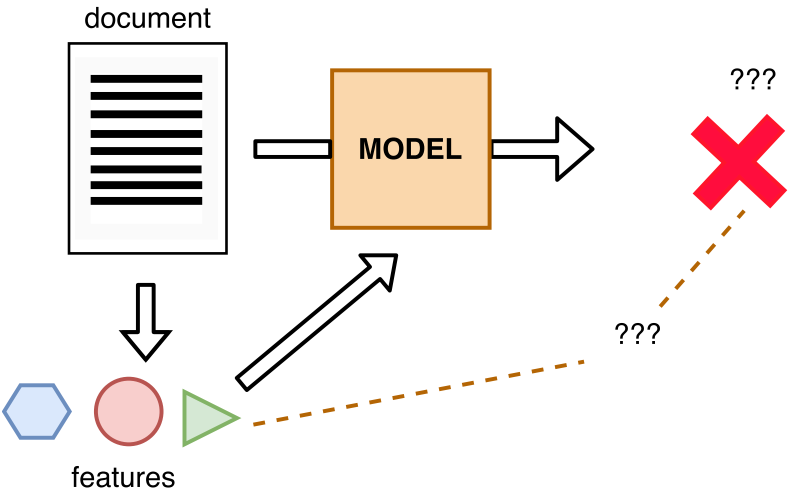
Marlene Staib

September 20, 2018

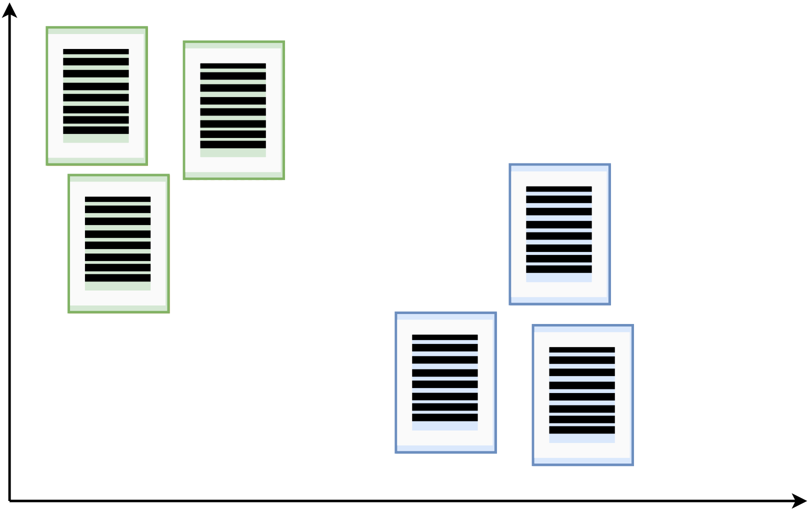
The story so far...



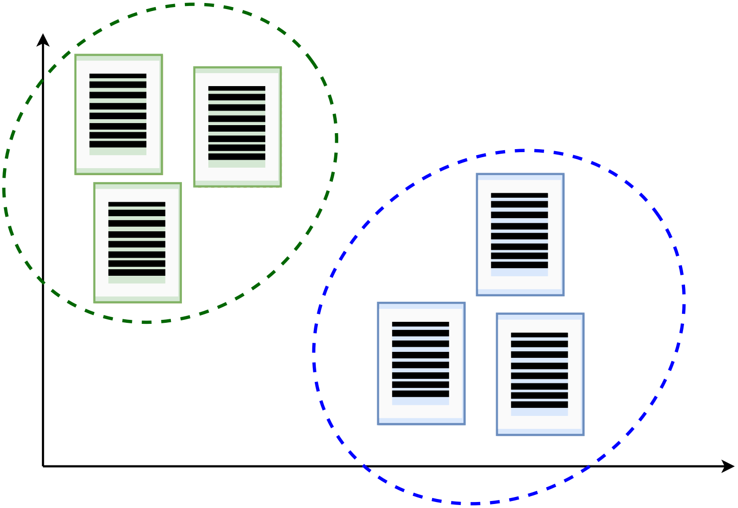
What if...



Unsupervised Learning

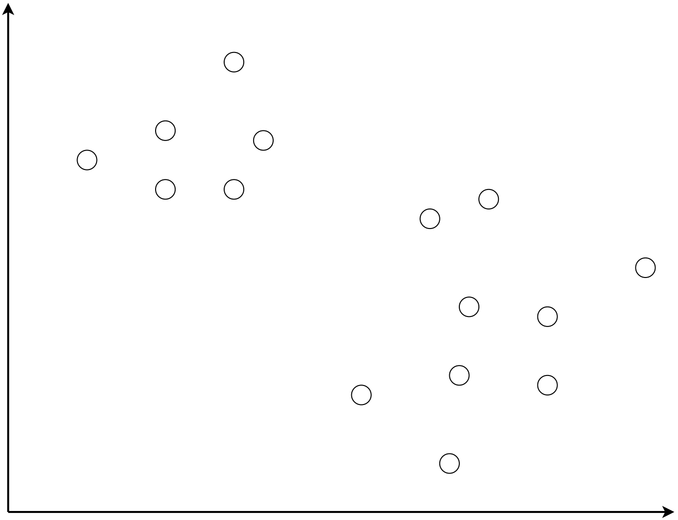


Unsupervised Learning



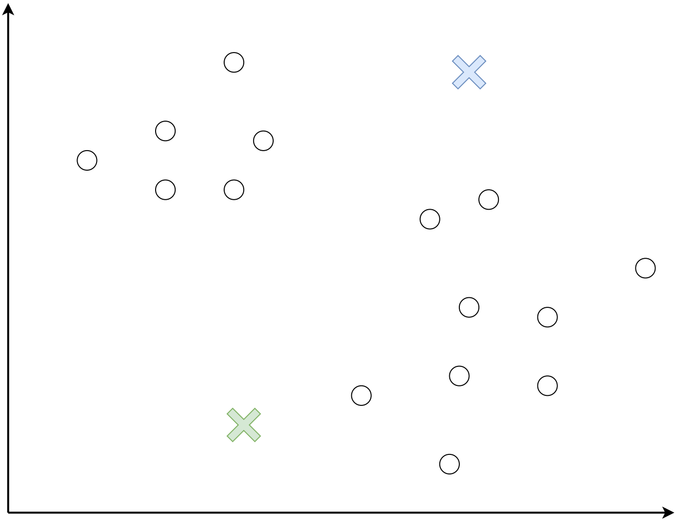
k-means clustering

k-means algorithm



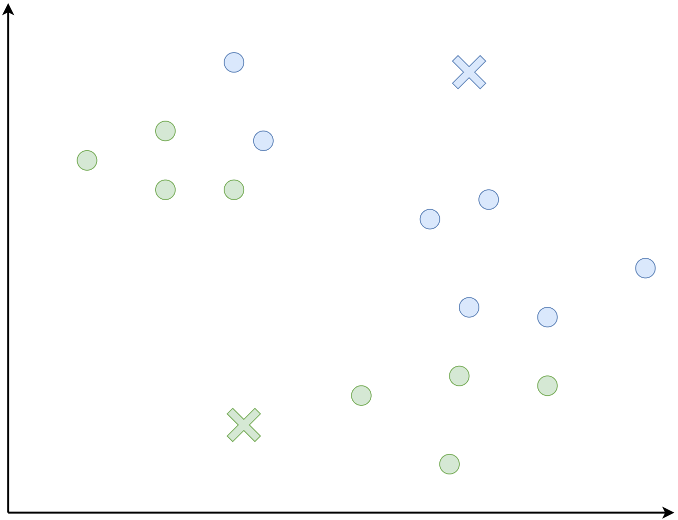
k-means algorithm

0. Initialize cluster centres randomly:



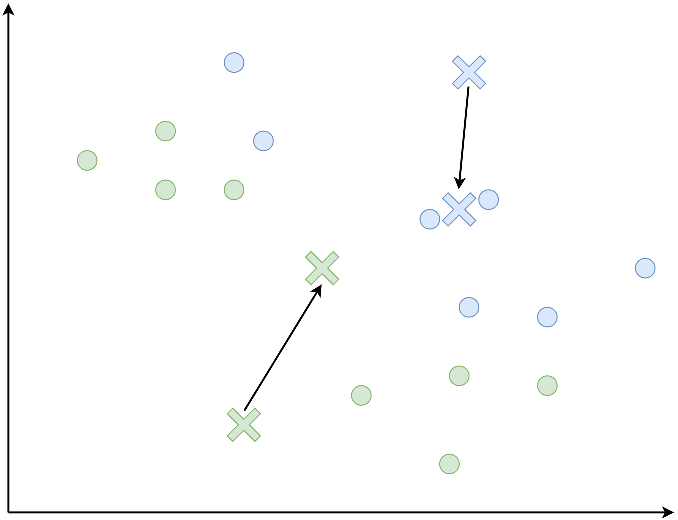
k-means algorithm

1. Assign all data points to their nearest cluster centre:



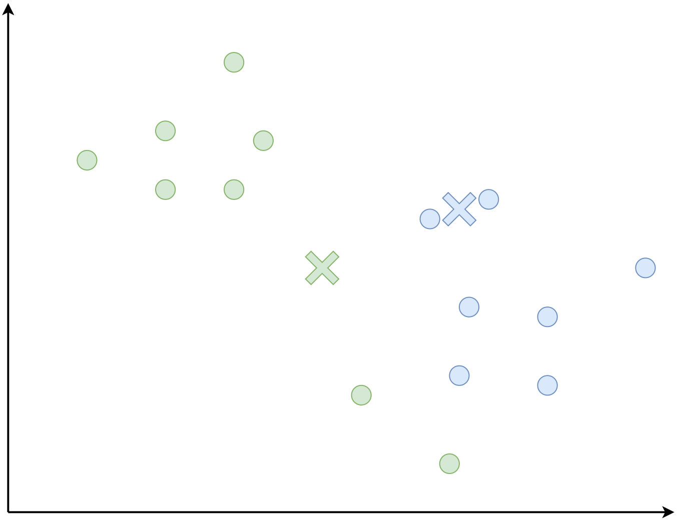
k-means algorithm

2. Move the cluster centre to the mean of its assigned data points:



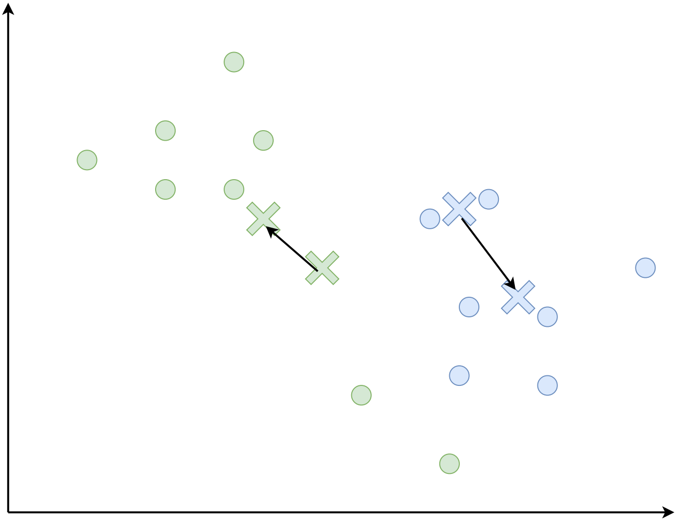
k-means algorithm

Reassign the data points; iterate 2 and 3:



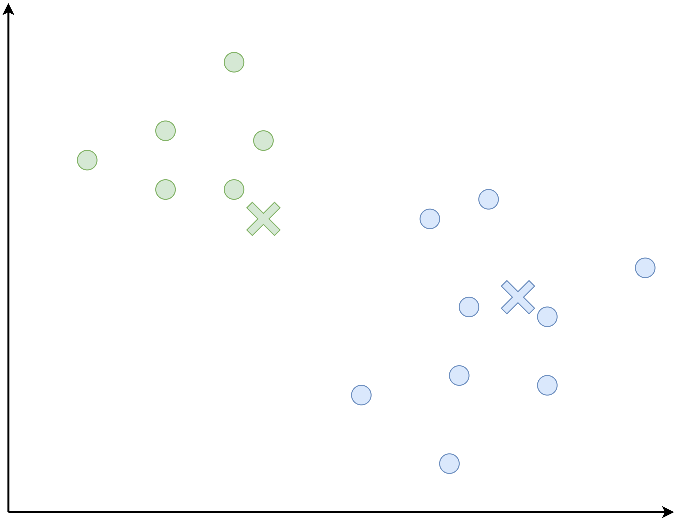
k-means algorithm

Iterate 2 and 3:



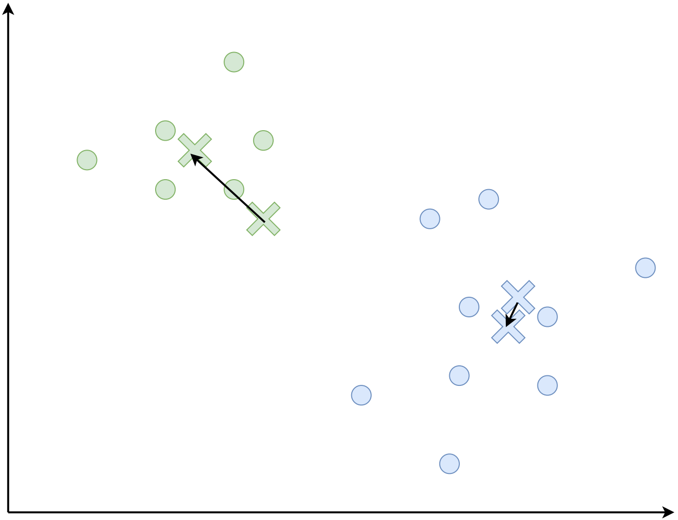
k-means algorithm

Iterate 2 and 3:



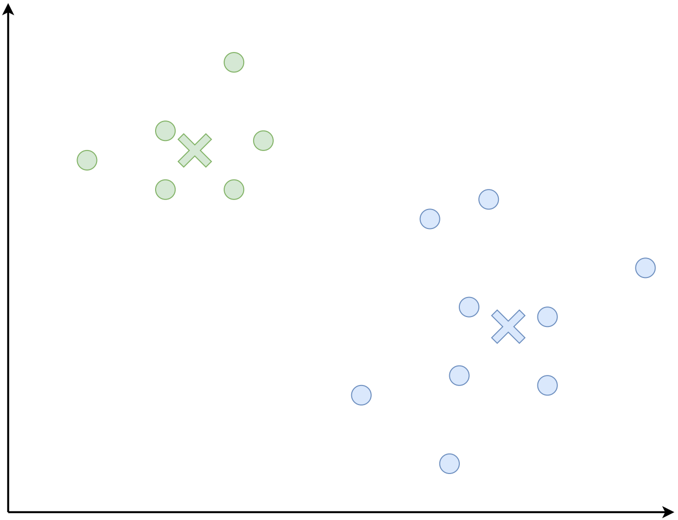
k-means algorithm

Iterate 2 and 3:



k-means algorithm

Convergence:



Worked example

data points:

$$A : \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ B : \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \ C : \begin{bmatrix} 5 \\ 2 \end{bmatrix}, \ D : \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \ E : \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \ F : \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

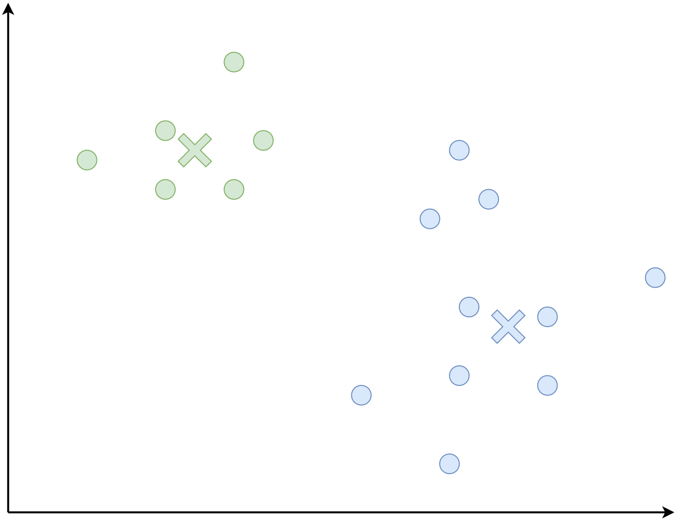
Use the Euclidean distance to determine the closest centre for each point:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Iteration	μ_1	data c_1	μ_2	data c_2
0	[1,1]	-	[4,4]	-
1	[1.5,1]	A,B,D,F	[4,2.5]	C,E
2	[1.5,1]	A,B,D,F	[4,2.5]	C,E

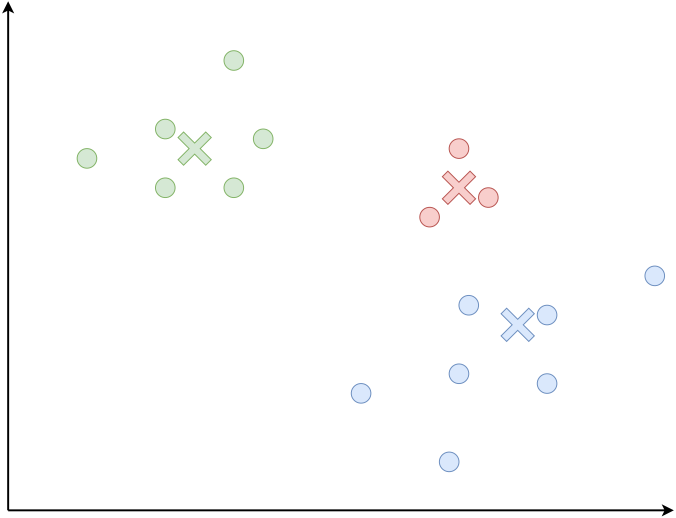
Finding k

$k = 2$



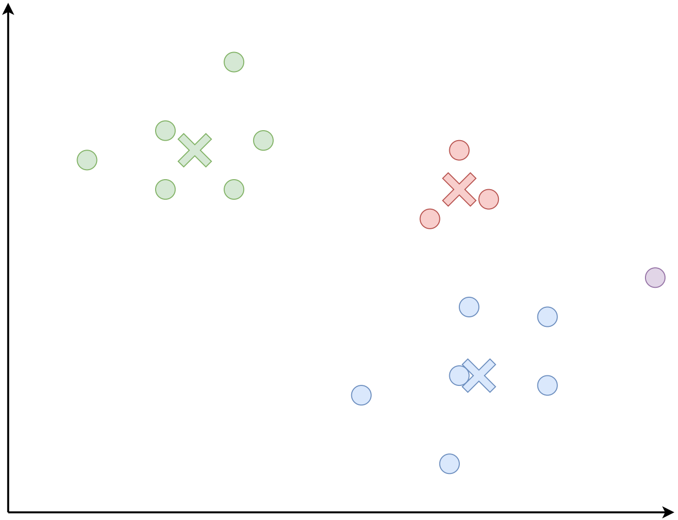
Finding k

k = 3



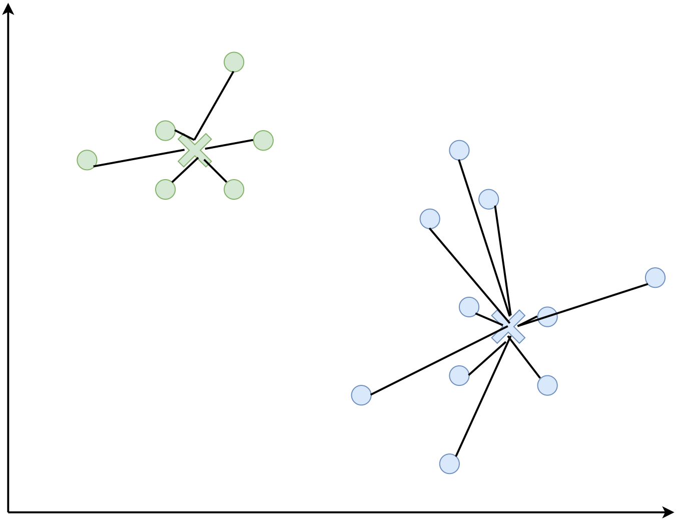
Finding k

k = 4



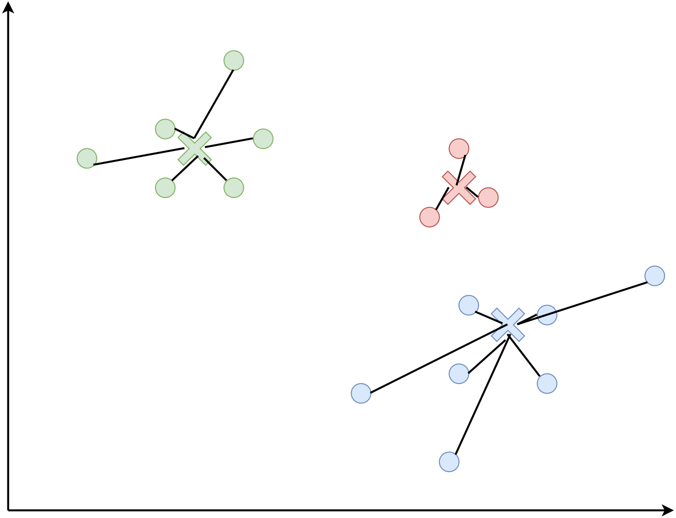
Finding k: Mean Squared Error (MSE)

k = 2



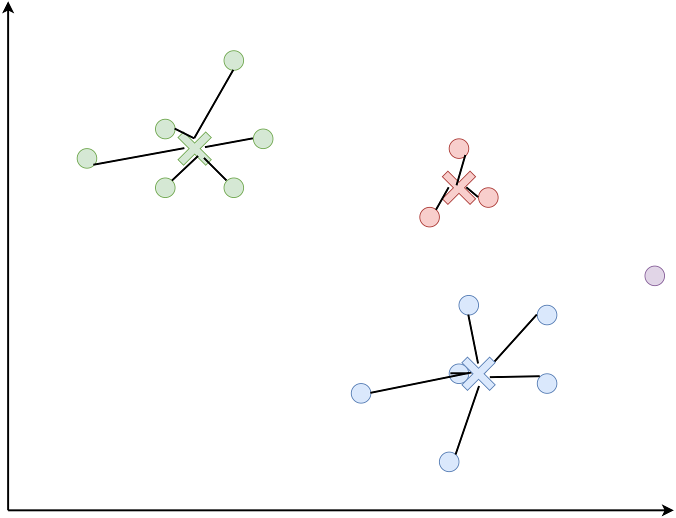
Finding k: Mean Squared Error (MSE)

k = 3



Finding k: Mean Squared Error (MSE)

k = 4



Finding k: Scree plot

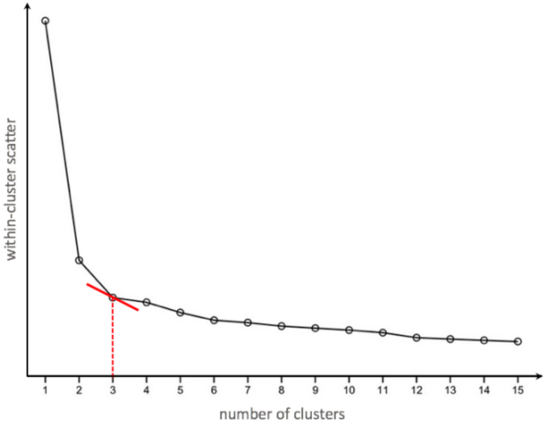
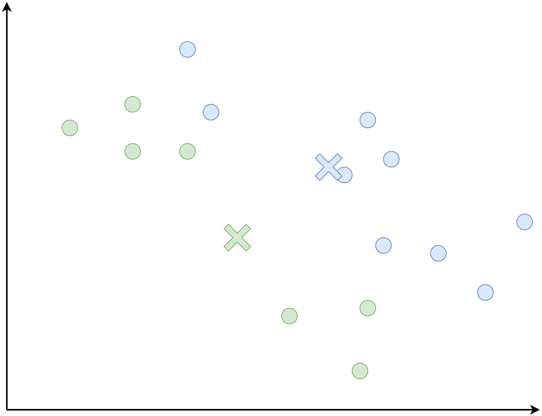
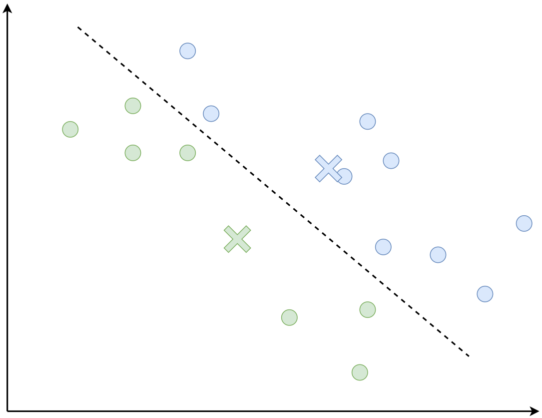


image from:
https://algobeans.com/2015/11/30/k-means-clustering-laymans-tutorial/

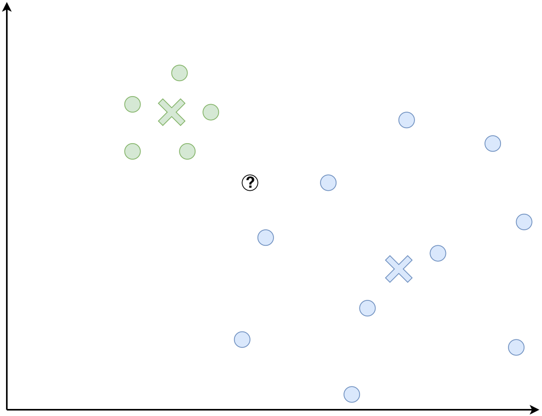
Solution depends on initialization!



Solution depends on initialization!



No variance

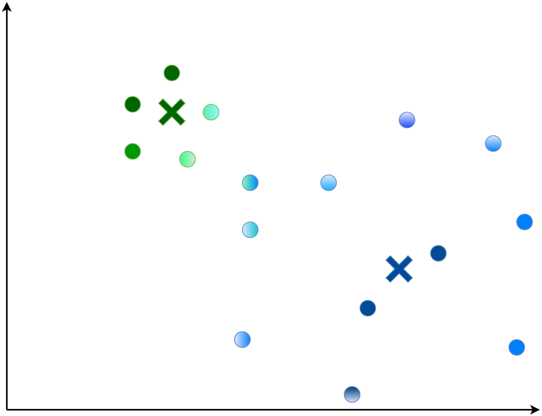


Disadvantages of k-means

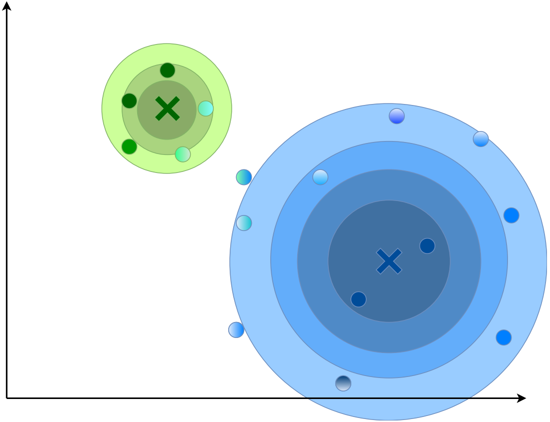
- Have to know k!
- Dependent on initialization
- Hard cluster assignment
- Doesn't take variance into account

Gaussian Mixture Models (GMMs)

GMMs: From hard clustering to probabilities



GMMs: From hard clustering to probabilities



GMMs: Model parameters

For each mixture m:

- Mean μ_m
- Variance σ_m^2
- Prior $P(m)$

The likelihood of any given data point x_i under m is calculated as:

$$P(x_i|m) = \frac{1}{2\pi\sigma_m^2}\ exp(-\frac{(x_i - \mu_m)^2}{2\sigma_m^2})$$

The posterior probability of the mixture, given x_i is:

$$P(m|x_i) = \frac{P(x_i|m)P(m)}{\sum\limits_{m'} P(x_i|m')P(m')}$$

Learning GMMs: The EM algorithm

Similar to k-means:

- Start with random values for μ_m, σ_m; uniform priors
- Calculate $P(m|x_i)$ for every data point
- Update μ_m, σ_m (and priors), weighing each data point proportional to its probability
- Iterate until convergence

Applications of clustering

Clustering documents

- Group by topic, author, time, ...
- Semi-supervised: use a few known examples to link the clusters to a class
- See if there exists a grouping by features
- In networks: Discover sub-networks

Clustering features

- Use clusters/likelihoods as features in supervised task
  - e.g. topics
  - word classes
  - style elements that are typical of a specific class

Clustering speakers