University of BRISTOL

# UNSUPERVISED LEARNING

Introduction to Artificial Intelligence

bristol.ac.uk

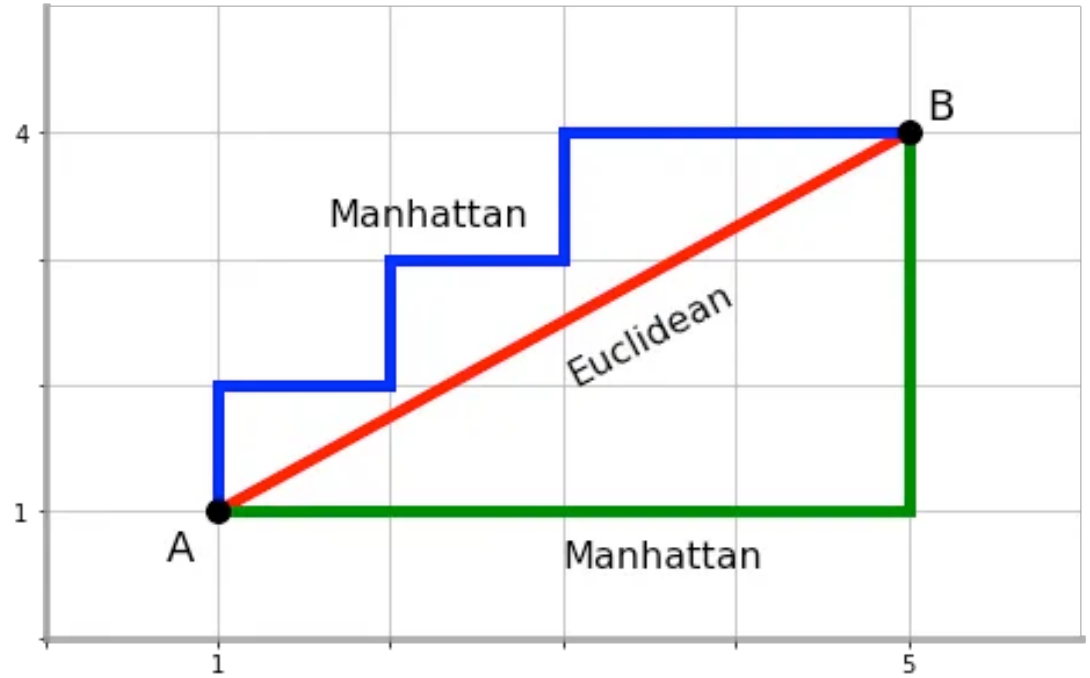
# When do we want to use unsupervised learning?

- When we don’t have labelled data
- Examples:
  - Document classification, e.g. Google News
  - Anomaly detection, e.g. faulty equipment or security breaches
  - Customer personas
  - Recommendation engines
- Do you have examples? Post on the discussion board!

# Clustering Algorithms

- Aim: Given unlabelled data the aim is to **partition the dataset into distinct clusters of unlabelled elements**
- **Similarity:** based on some notion of **distance** between datapoints
- Challenges:
  - Knowing the **optimal (best) number of clusters**
  - **Stability** and **convergence** of algorithms

# Similarity and Distance






https://www.omnicalculator.com/math/manhattan-distance

# Distance

- Euclidean distance: For a feature space $V = \mathbb{R}^n$, $\overrightarrow{x} = (x_1, \ldots, x_n)$ and $\overrightarrow{y} = (y_1, \ldots, y_n)$

$$d(\overrightarrow{x}, \overrightarrow{y}) = \| \overrightarrow{x} - \overrightarrow{y} \|_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

# Distance

- Manhattan distance: For a feature space $V = \mathbb{R}^n$, $\overrightarrow{x} = (x_1, \ldots, x_n)$ and $\overrightarrow{y} = (y_1, \ldots, y_n)$

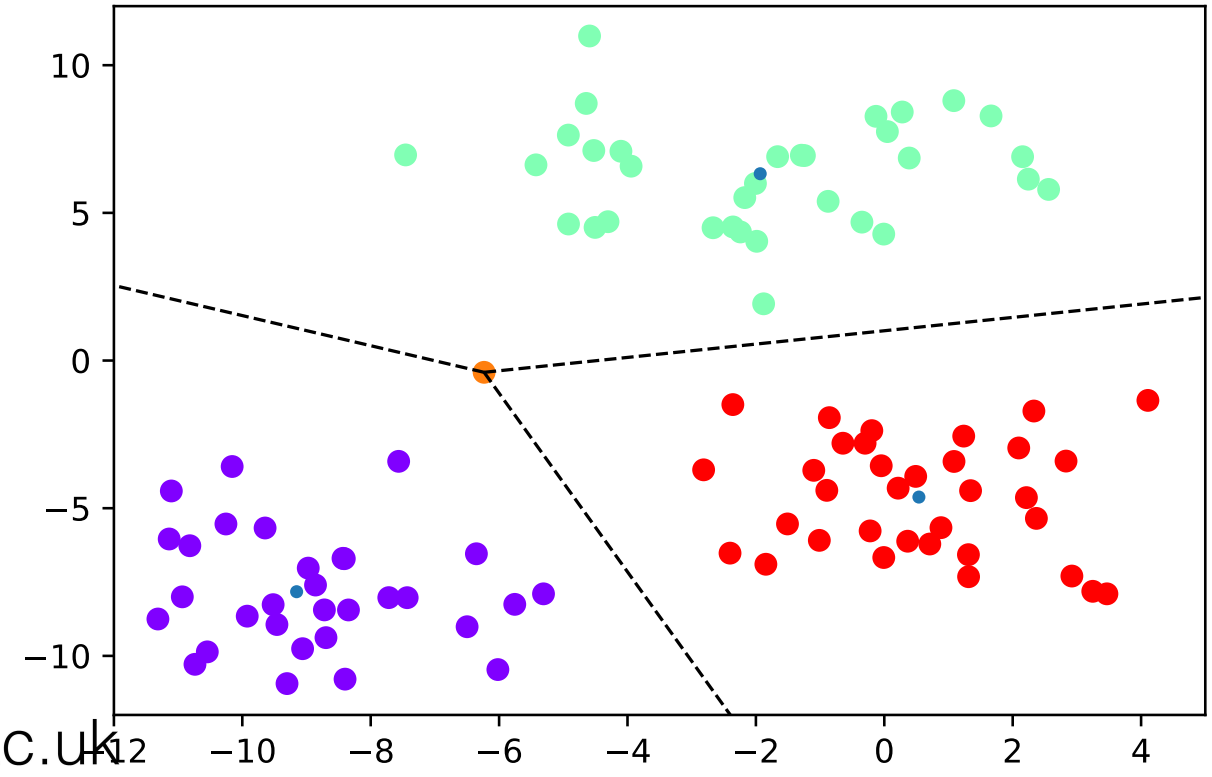
$$d(\overrightarrow{x}, \overrightarrow{y}) = \| \overrightarrow{x} - \overrightarrow{y} \|_1 = \sum_{i=1}^{n} | x_i - y_i |$$

# Distance

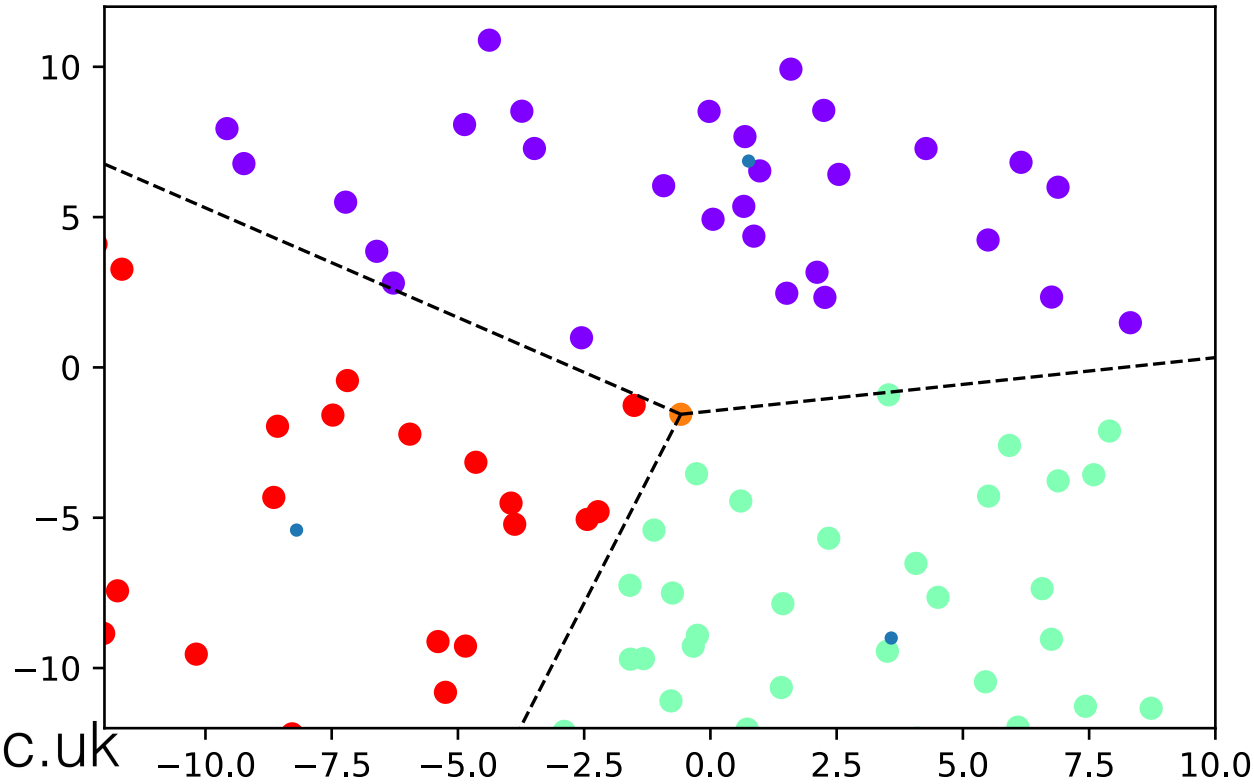
- Hamming distance: For a feature space $V$ that is a set of symbols, Hamming distance between two strings of symbols is the number of places at which they differ

- 000, 111 - Hamming distance is 3
- abc, abd - Hamming distance is 1

# k-Means Clustering



# k-Means Clustering




bristol.ac.uk

# k-Means Clustering Algorithm

- Suppose we want to partition a set of vectors $\overrightarrow{x}_i \in V = \mathbb{R}^n$ into $k$ clusters.

1. Start by randomly partitioning the vectors into $k$ sets.
2. Find the centroid of each set of vectors
3. Re-label each vector based on which centroid it is closest to
4. If there is no change in the labelling of the vectors stop, else go to 2

# Centroid

- Find the centroid of each set of vectors
- Means: add up all the vectors and divide by the number of vectors
- Finds the middle point.

$$\hat{x}_P = \left( \frac{\sum_{i=1}^{|P|} x_{i1}}{|P|}, \ldots, \frac{\sum_{i=1}^{|P|} x_{ij}}{|P|}, \ldots, \frac{\sum_{i=1}^{|P|} x_{in}}{|P|} \right)$$

# How good is the clustering?

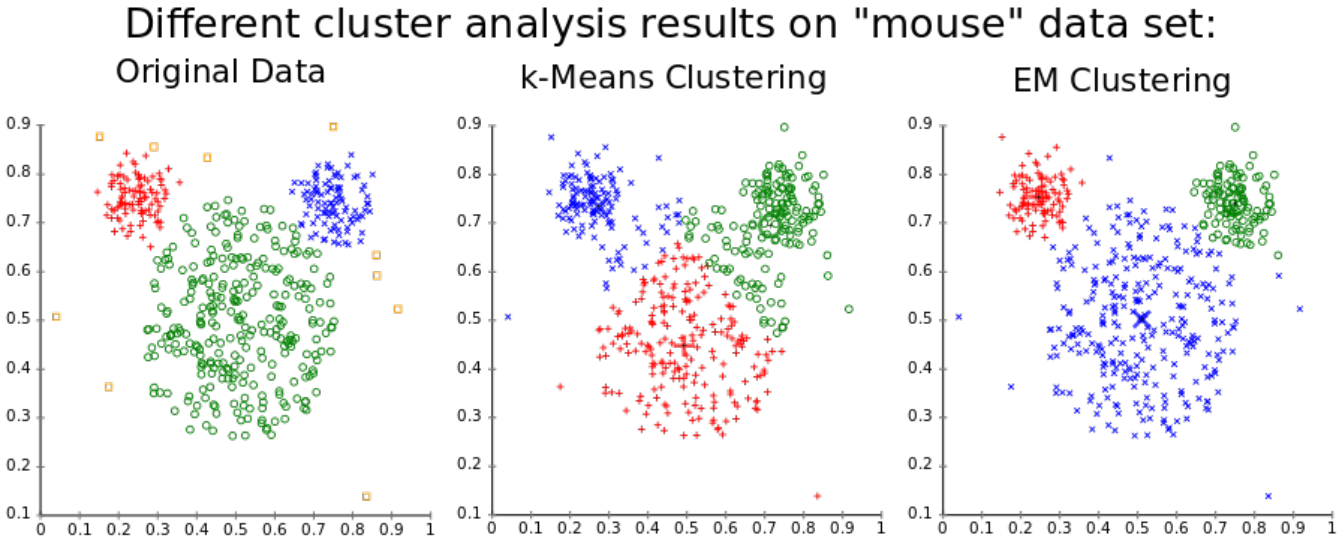
- We use the *degree of dissimilarity* to measure how good a clustering is. Suppose we have a partition $\mathbf{P} = \{P_1, \ldots, P_k\}$ of our data.

$$J(\mathbf{P}) = \sum_{i=1}^{k} \sum_{j=1}^{|P_i|} \| \overrightarrow{x}_{ij} - \hat{x}_{P_i} \|^2$$

- We call this the *within cluster sum of squares* or the *inertia*
- K-means aims to minimise this quantity.

# Drawbacks of k-means

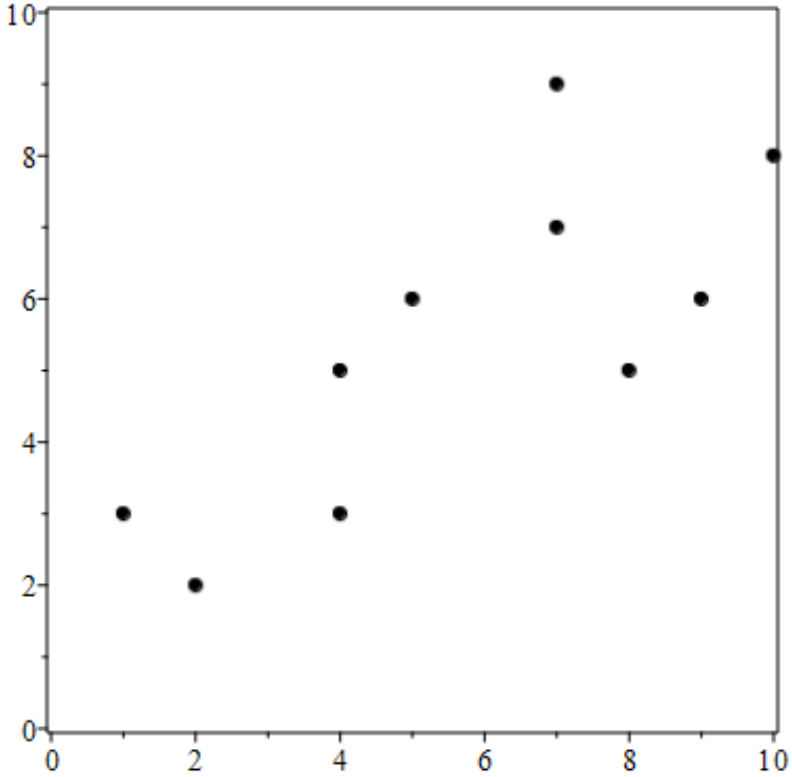
- Assumes that clusters are spherical and equally sized
- Number of clusters is an input parameter
- Can converge to local minima



# K-means example

- Consider the following points in $\mathbb{R}^2$:

$$\{(10,8), (7,9), (1,3), (2,2), (4,3),$$
$$(8,5), (7,7), (5,6), (4,5), (9,6)\}$$



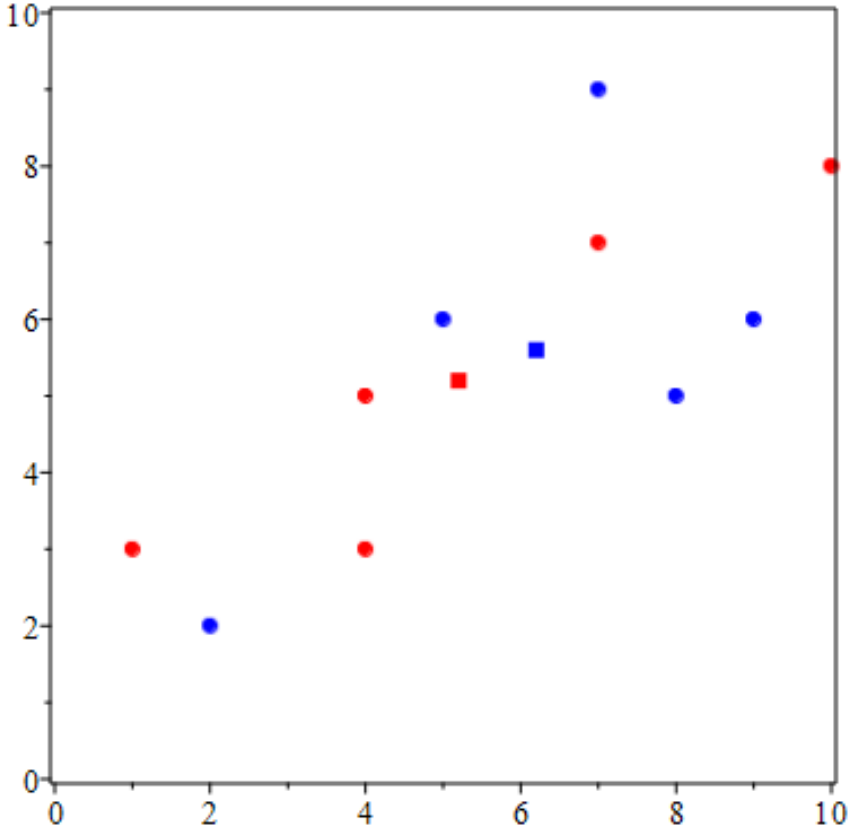
# K-means example

Let $k = 2$ and let the initial partition be:

$P_1 = \{(10,8), (1,3), (4,3), (7,7), (4,5)\}$

$P_2 = \{(7,9), (2,2), (8,5), (5,6), (9,6)\}$

With centroids $\hat{x}_{P_1} = (5.2,5.2)$ and $\hat{x}_{P_2} = (6.2,6.5)$

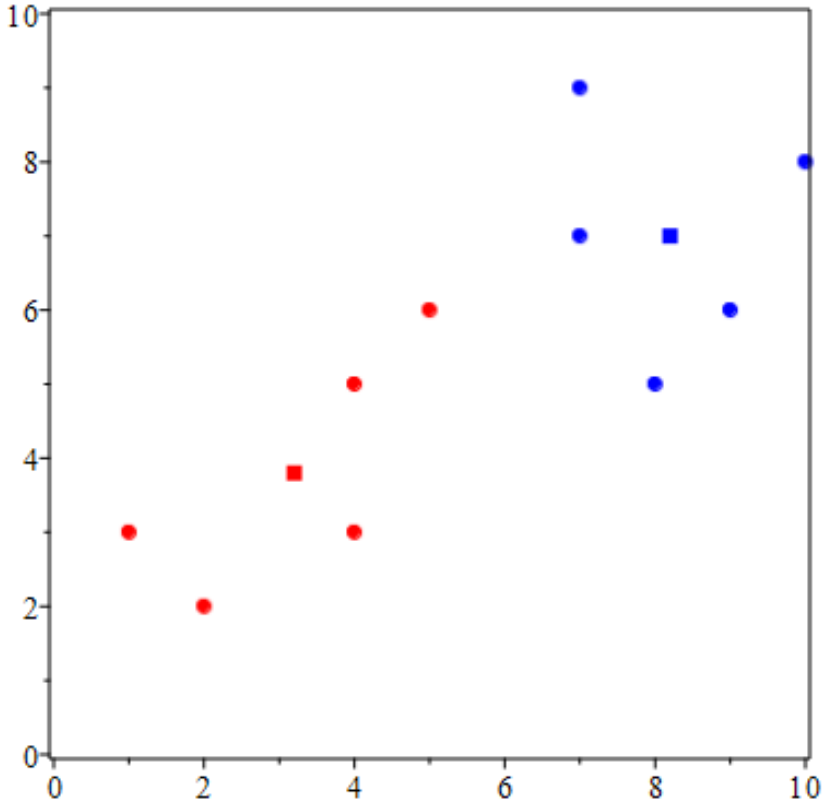


# K-means example

Reallocating points according to distance from means gives us:

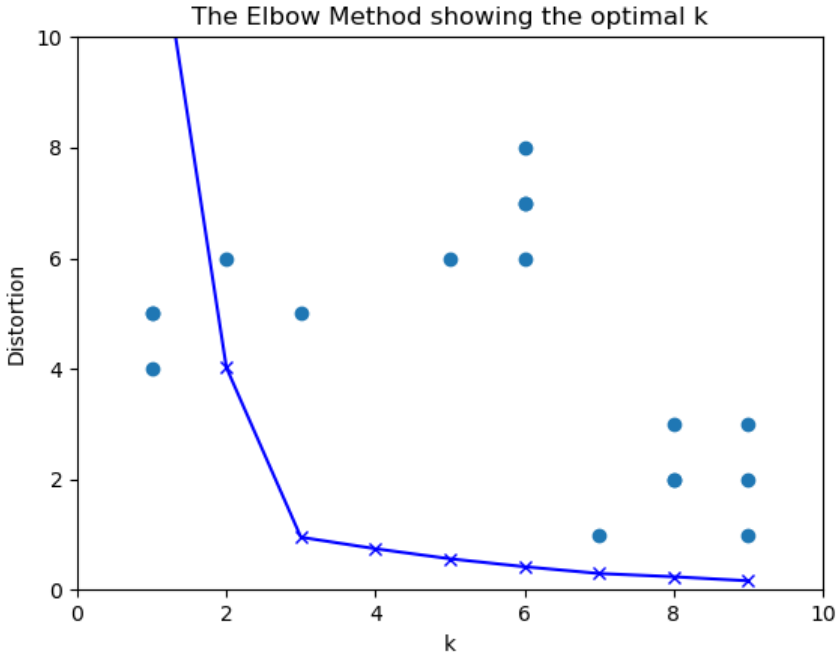
$$P_1 = \{(1,3), (2,2), (4,3), (5,6), (4,5)\}$$
$$P_2 = \{(10,8), (7,9), (8,5), (7,7), (9,6)\}$$

Calculating new means and updating gives no change so terminate.



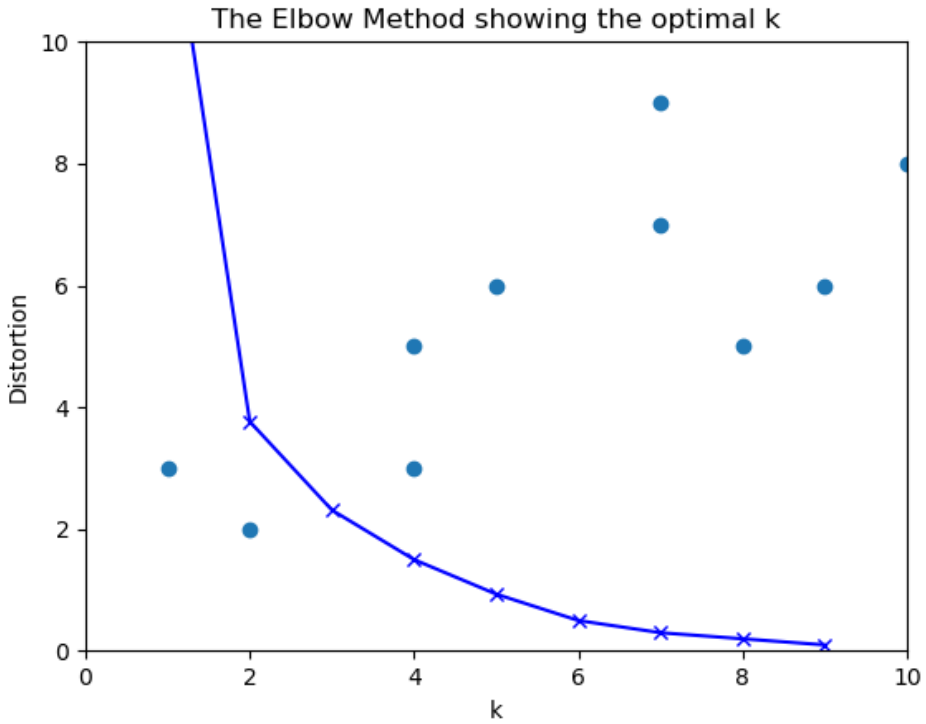
# Elbow Plots

- Elbow plots help to identify the optimal $k$
- Plot $J(\mathbf{P})$ against $k$
- The ‘elbow’ gives the optimal $k$



# Elbow Plots

- Not always clear what the optimal $k$ is.

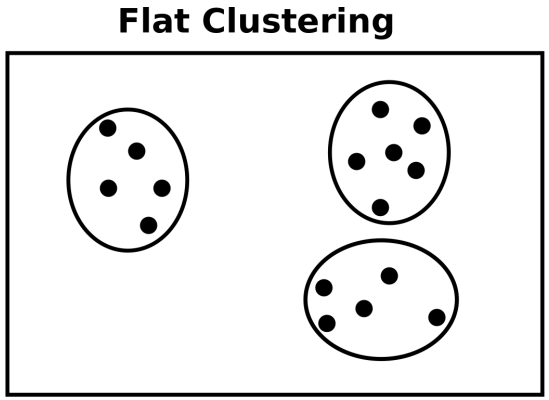


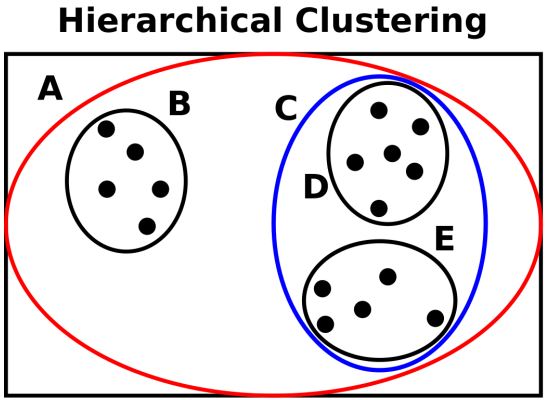
# Summary - k-means

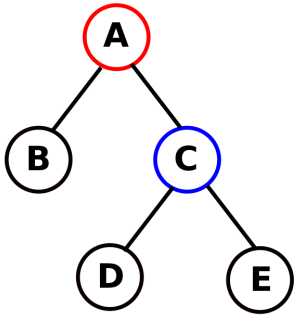
- k-means is a simple and intuitive algorithm to cluster data.
- k-means works by minimizing the variance of clusters of data that are determined by the cluster centroids.
- At each point in the algorithm, the cluster centroids are updated, and the datapoints reallocated to the clusters
- k-means suffers from a number of drawbacks: it assumes that the clusters are spherical and equally sized, it requires that we choose the number of clusters up front, and it can converge to local minima
- Further reading:
  - Pattern Recognition and Machine Learning section 9.1, Bishop 2006
  - https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/
  - Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval https://nlp.stanford.edu/IR-book/ Chapter 16, up to 16.4

# Hierarchical clustering

- Hierarchical clustering recognises that we may have more structure in our data
- Top-down clustering starts with all datapoint in one cluster and divides into child clusters
- Bottom-up clustering builds up clustering from individual data points by merging

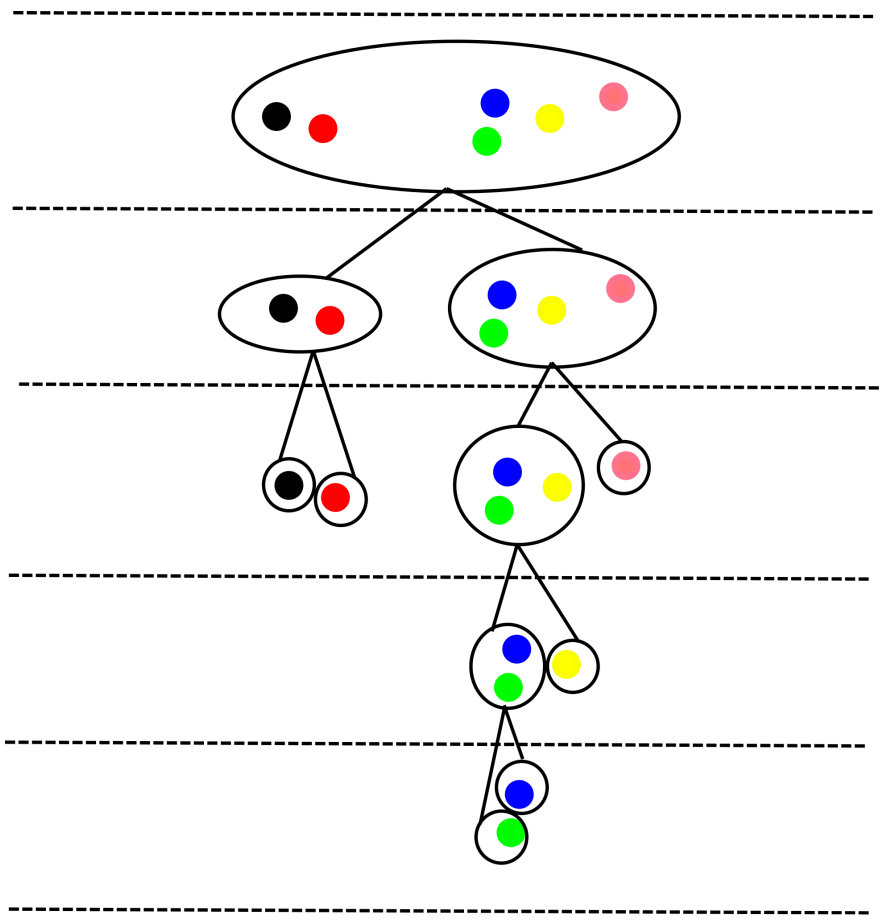


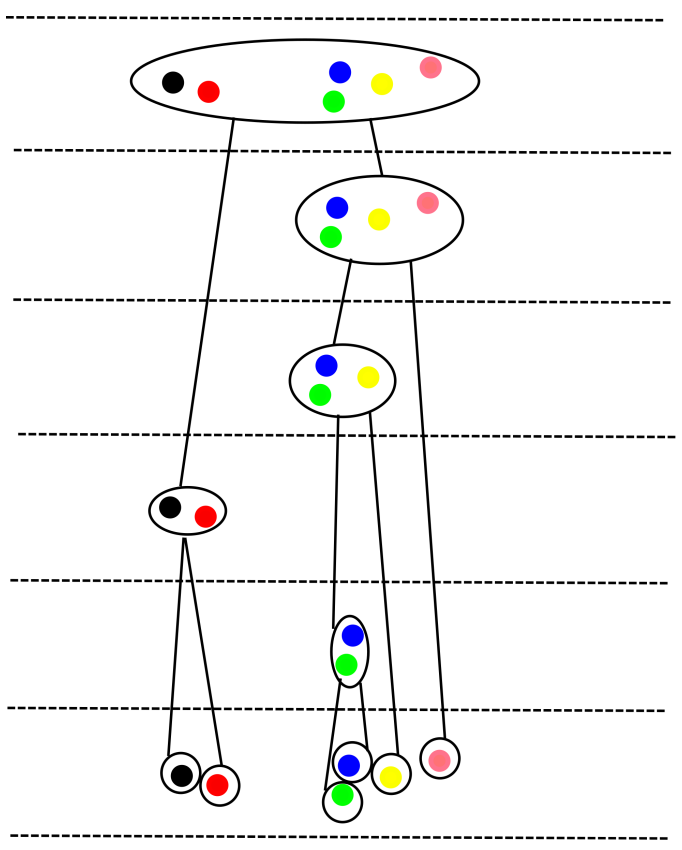
# Hierarchical k-means

- For fixed $k$ recursively run $k$-means until only single-element clusters remain

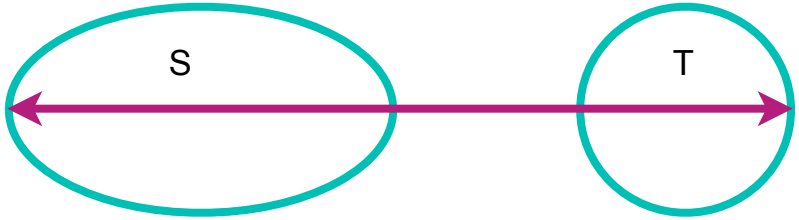


# Agglomerative clustering

- In agglomerative clustering we start from single points and merge into clusters based on proximity
- We need a notion of distance between sets

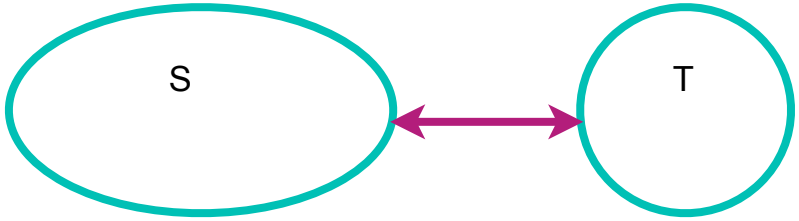


# Distance between sets




Maximum distance:

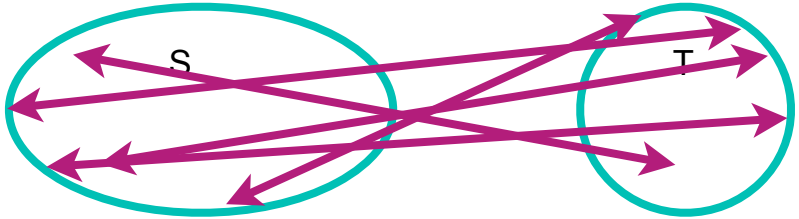
$$d(S,T) = \max\{d(x,y) : x \in S, y \in T\}$$




Minimum distance:

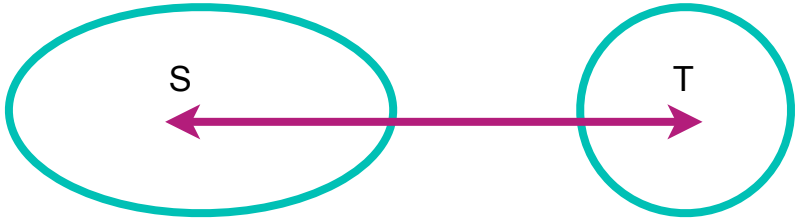
$$d(S,T) = \min\{d(x,y) : x \in S, y \in T\}$$

# Distance between sets




Average distance:

$$d(S,T) = \frac{1}{|S||T|} \sum_{x \in S} \sum_{y \in T} d(x,y)$$




Minimum distance:

$$d(S,T) = d(\frac{\sum_{x \in S} x}{|S|}, \frac{\sum_{x \in T} x}{|T|})$$

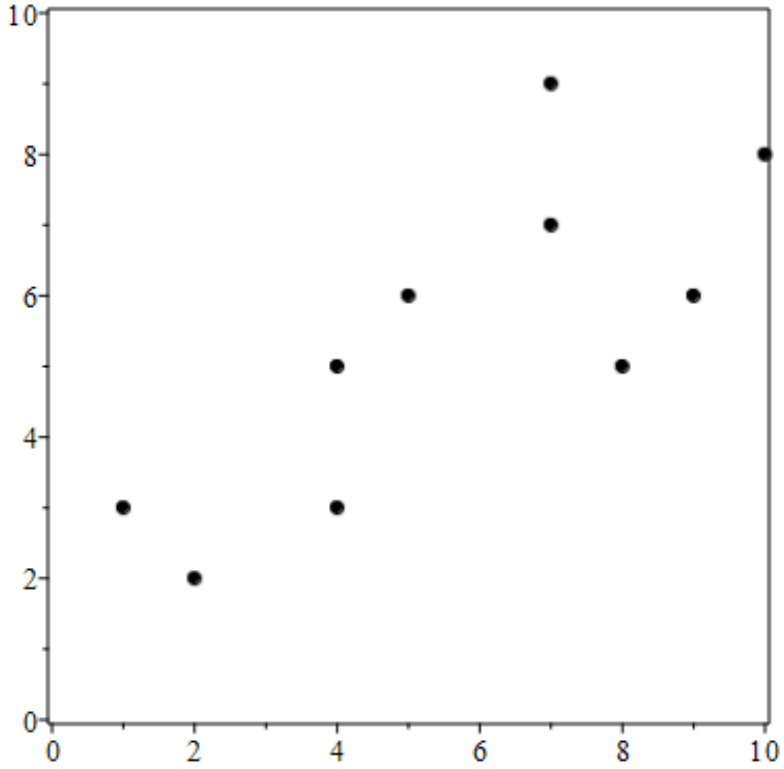
# Agglomerative Clustering Algorithm

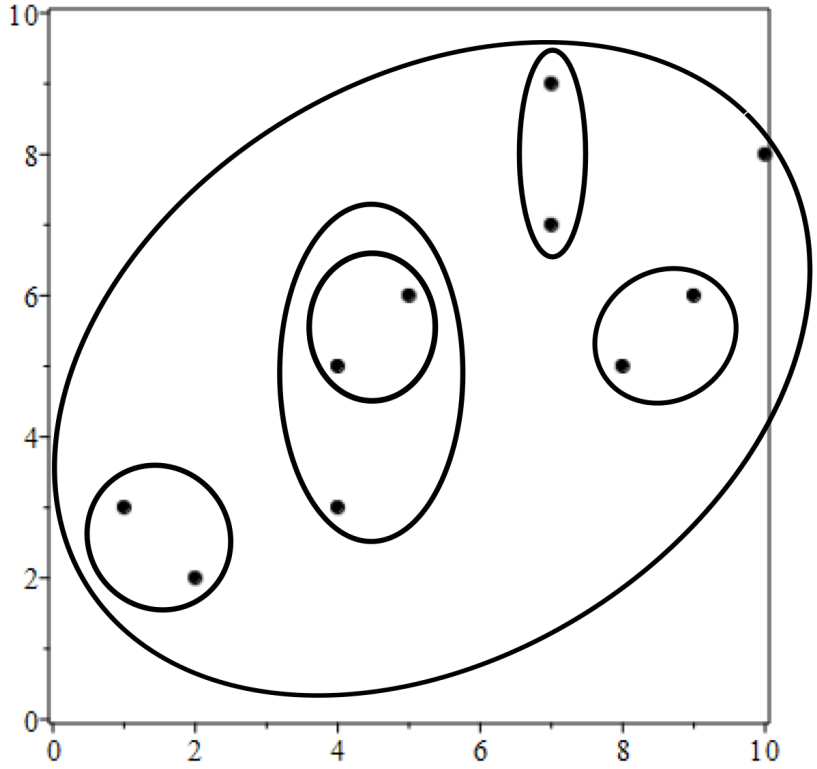
SIMPLEHAC$(d_1, \ldots, d_N)$
1 **for** $n \leftarrow 1$ **to** $N$
2 **do for** $i \leftarrow 1$ **to** $N$
3 **do** $C[n][i] \leftarrow \text{SIM}(d_n, d_i)$
4 $I[n] \leftarrow 1$ *(keeps track of active clusters)*
5 $A \leftarrow []$ *(assembles clustering as a sequence of merges)*
6 **for** $k \leftarrow 1$ **to** $N - 1$
7 **do** $\langle i, m \rangle \leftarrow \arg\max_{\{\langle i,m \rangle : i \neq m \wedge I[i]=1 \wedge I[m]=1\}} C[i][m]$
8 $A$.APPEND$(\langle i, m \rangle)$ *(store merge)*
9 **for** $j \leftarrow 1$ **to** $N$
10 **do** $C[i][j] \leftarrow \text{SIM}(i, m, j)$
11 $C[j][i] \leftarrow \text{SIM}(i, m, j)$
12 $I[m] \leftarrow 0$ *(deactivate cluster)*
13 **return** $A$

► **Figure 17.2** A simple, but inefficient HAC algorithm.

- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.
- https://nlp.stanford.edu/IR-book/

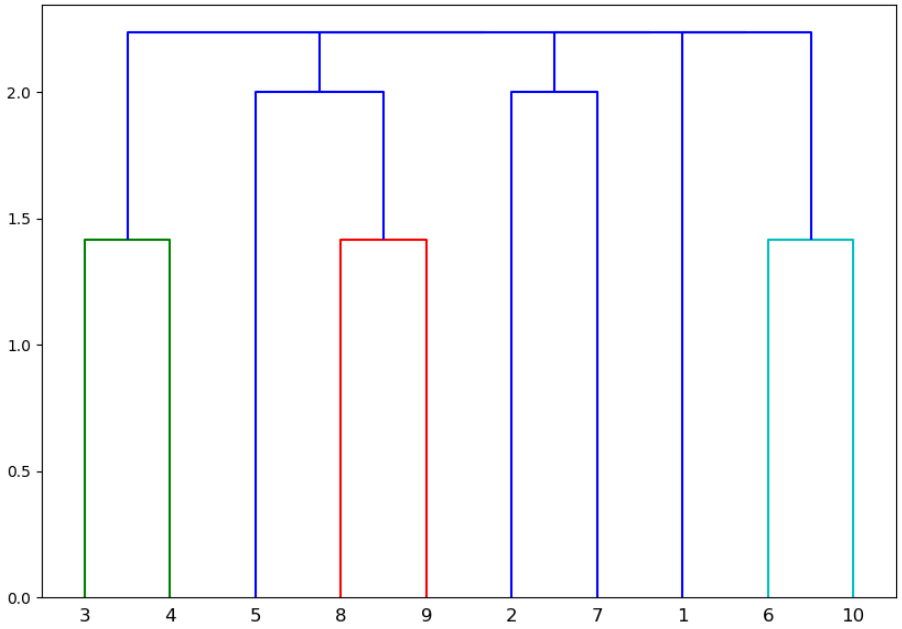
# Agglomerative Clustering Example



# Agglomerative Clustering Example

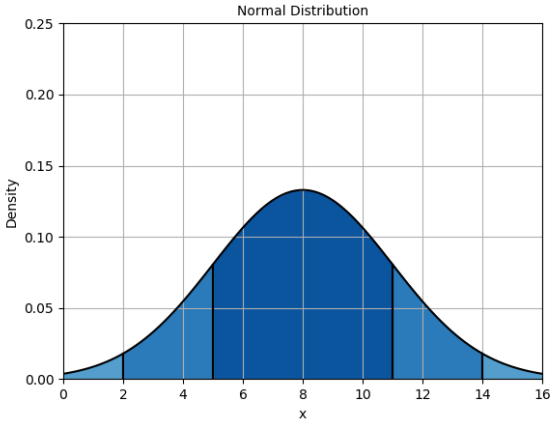
- The outcome of the clustering algorithm is represented in a dendrogram.
- The y-axis of the dendrogram indicates cluster similarity
- ‘Natural’ clusters of the data can be formed by cutting the dendrogram where the similarity between clusters changes most.

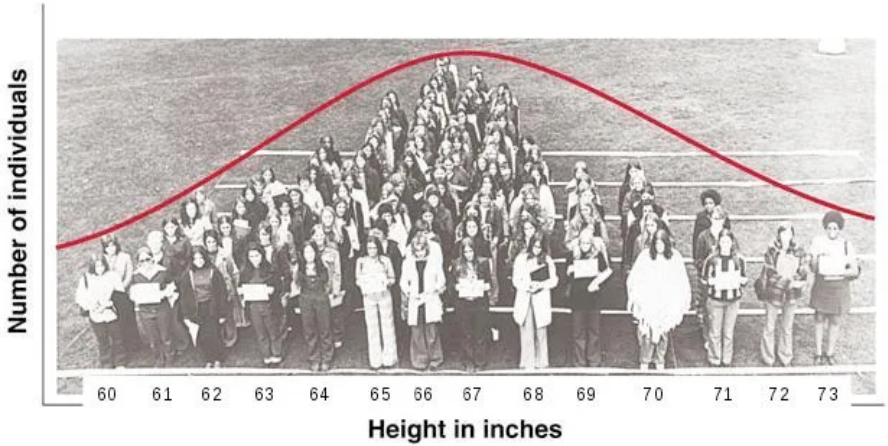


# Hierarchical Clustering Summary

- Hierarchical clustering gives you a set of clusters that can be applied at different levels of hierarchy.
- Hierarchical clustering can be done top-down, or divisively, or bottom-up, using agglomeration.
- The results of the clustering can be visualized in a dendrogram.
- The dendrogram shows the order of clustering and distance between clusters. A ‘natural’ division of the data into clusters can be inferred by cutting the dendrogram where the distance between clusters is greatest.
- Further reading: Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval https://nlp.stanford.edu/IR-book/ Section 17.1

# Gaussian Mixture Models

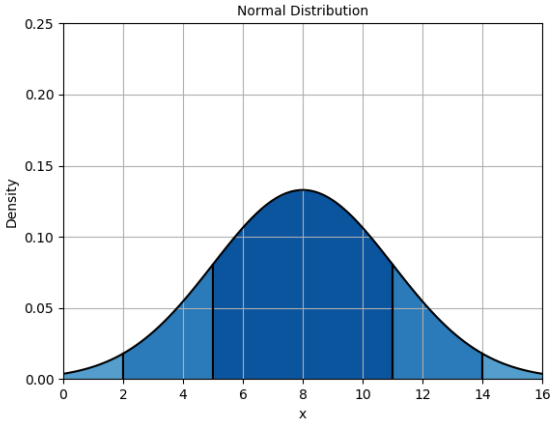


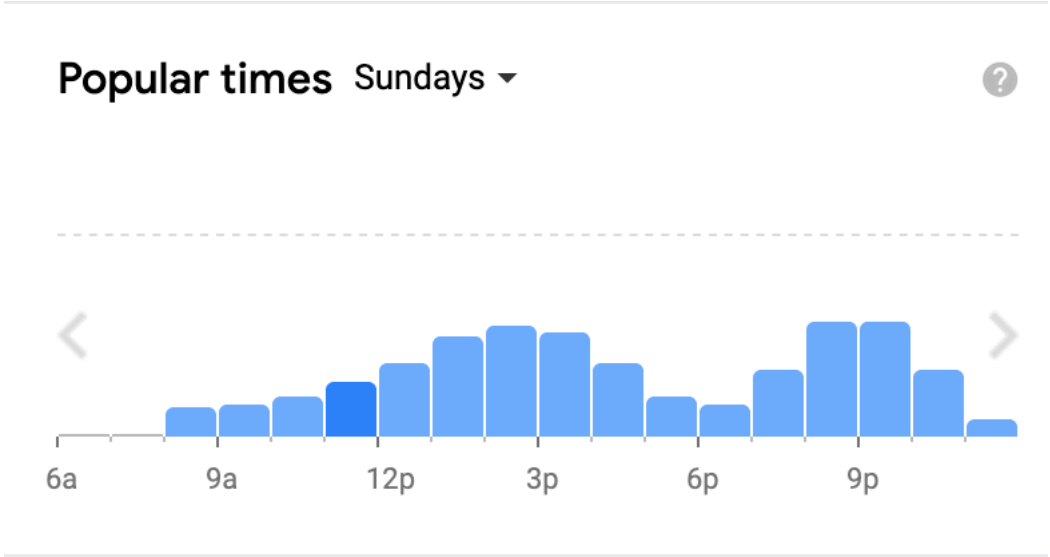

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

bristol.ac.uk

# Gaussian Mixture Models



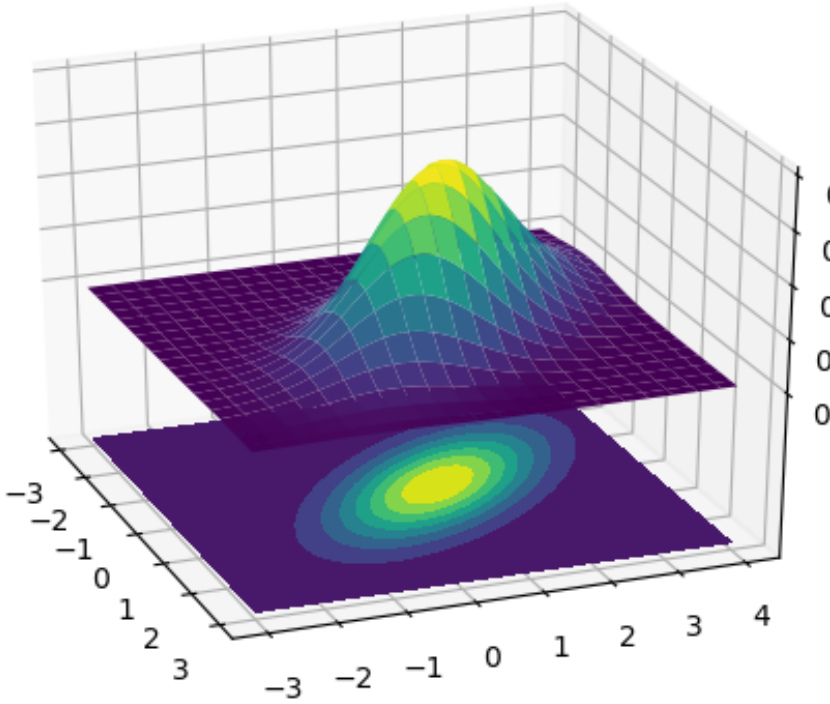

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$




bristol.ac.uk

# Multivariate Gaussian

- Let $\vec{x} = (x_1, \ldots, x_n)$ be a vector of random variables
- Let $\vec{\mu} = (\mu_1, \ldots, \mu_n)$ be the vector of means of the $x_i$
- Let $\Sigma$ be the $n$ by $n$ covariance matrix such that $\Sigma_{ij} = E((x_i - \mu_i)(x_j - \mu_j))$




$$\mathcal{N}(\vec{x} \mid \vec{\mu}, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} det(\Sigma)^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1} (\vec{x}-\vec{\mu})}$$

# Height vs shoe size

# Gaussian Mixture Distributions

We combine Gaussians together to make a *mixture distribution*

$$p(\vec{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\vec{x} \mid \vec{\mu}_k, \Sigma_k)$$

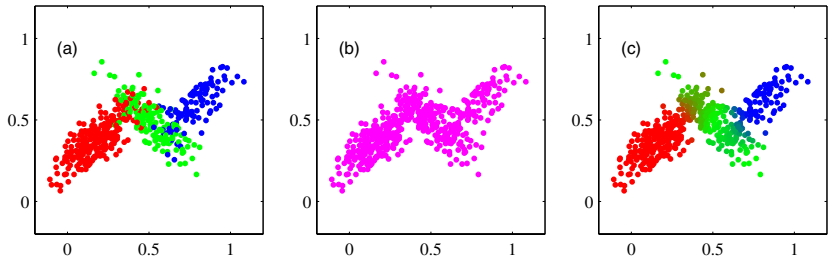
Each of the $\mathcal{N}(\vec{x} \mid \vec{\mu}_k, \Sigma_k)$ is called a *component* of the distribution

We require that $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k > 0$ for all $k$

# Learning Gaussian Mixture Models

1. Initialise the means $\vec{\mu}_k$, covariances $\Sigma_k$ and mixing coefficients $\pi_k$
2. Calculate the *responsibilities*

$$r_k(\vec{x}_n) = \frac{\pi_k \mathcal{N}(\vec{x}_n \mid \vec{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\vec{x}_n \mid \vec{\mu}_j, \Sigma_j)}$$




4. Calculate new means $\vec{\mu}_k^{new}$, covariances $\Sigma_k^{new}$ and mixing coefficients $\pi_k^{new}$ (next slide)

# Learning Gaussian Mixture Models

Bishop, Pattern Recognition and Machine Learning, section 9.2

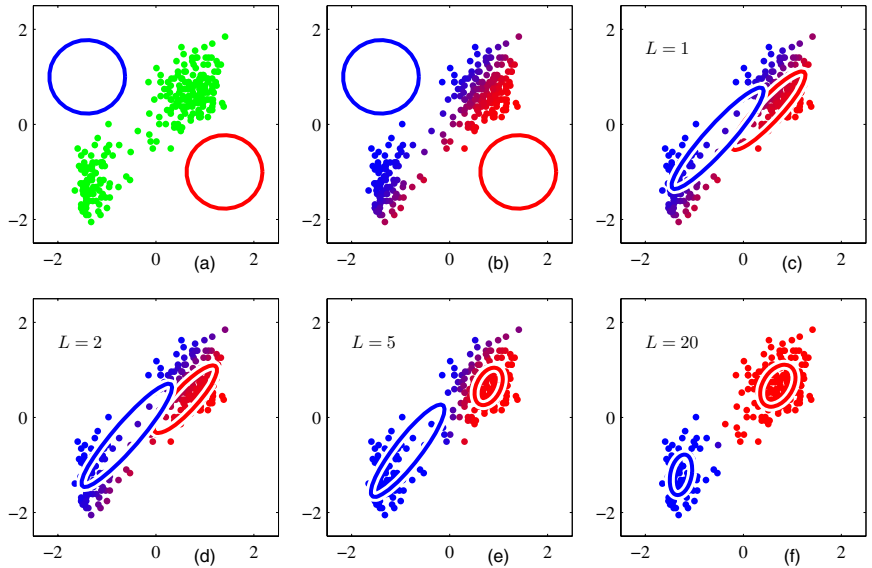
$$\vec{\mu}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} r_k(\vec{x}_n) \vec{x}_n$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} r_k(\vec{x}_n)(\vec{x}_n - \vec{\mu}_k^{new})(\vec{x}_n - \vec{\mu}_k^{new})^T$$

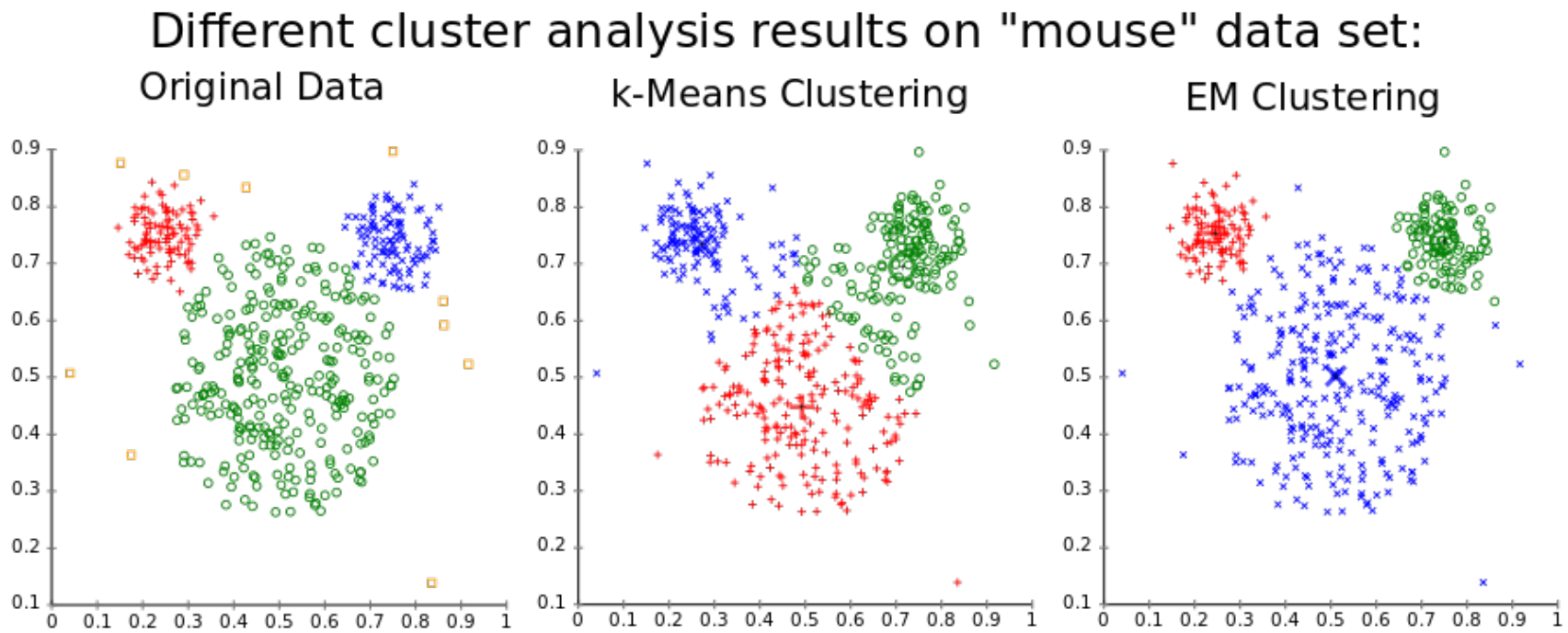
$$\pi_k^{new} = \frac{N_k}{N}$$

where $N_k = \sum_{n=1}^{N} r_k(\vec{x}_n)$

Continue until convergence



# GMMs have more flexibility in clustering




bristol.ac.uk

# GMMs Summary

- Gaussian mixture models can be used to cluster data in a probabilistic way
- We represent the data as a weighted sum of multivariate Gaussians
- The model parameters are learnt using an iterative algorithm, which is a kind of expectation maximisation algorithm.
- Further reading:
  - Pattern Recognition and Machine Learning section 9.2, Bishop 2006
  - https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/
  - AIMA section 20.3.1

# Worksheet

- Covering k-means, hierarchical clustering, and Gaussian mixture models
- Interpretation of results and the effect of different distance metrics