# Clustering

# Outline

- Why cluster data?
- Clustering as unsupervised learning
- Clustering algorithms
  - k-means, k-medoids
  - agglomerative clustering
  - Brown's clustering
  - Spectral clustering
- Cluster evaluation measures
  - Purity
  - Normalised Mutual Information
  - Rand Index
  - B-CUBED
  - Precision, Recall, F-score

# Why cluster data?

- Data mining has two main objectives:
  - Prediction: classification, regression etc.
  - Description: pattern mining, rule extraction, visualisation, *clustering*
- Clustering is:
  - Unsupervised learning
    - no label data is required (consider classification algorithms we discussed so far in the lectures which are supervised algorithms)

# Unsupervised Learning

- Supervised learning
    - labels for training instances are provided
- Unsupervised learning
    - no labels for training instances are provided
- Semi-Supervised learning
    - Both labeled and unlabeled training instances are provided
- What can we learn about training data if we do not have any labels?
    - The similarity and distribution of the features can still be learnt and this can be used to create rich feature spaces for supervised learning (if required )

# Clustering: Example

## Headlines

More Headlines

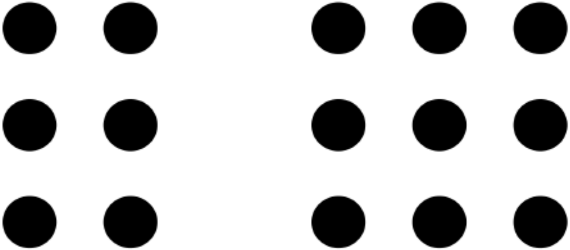
**Coronavirus: Boris Johnson announces plan for 'delay' phase**

Daily Mail · 1 hour ago

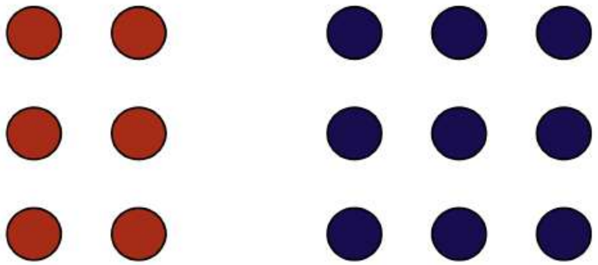
- **Coronavirus: Boris Johnson to hold emergency Cobra meeting**
  BBC South East Wales · 7 hours ago
- **BREAKING: UK cases of coronavirus rise to 319**
  Sky News · 6 hours ago
- **Coronavirus brings a reminder of the iron law of politics**
  Financial Times · 4 hours ago · Opinion
- **Nigel Farage: Yes, Protecting Us All from an Epidemic Should be Prioritized Over the Economy | Opinion**
  Newsweek · 2 hours ago · Opinion

View Full coverage

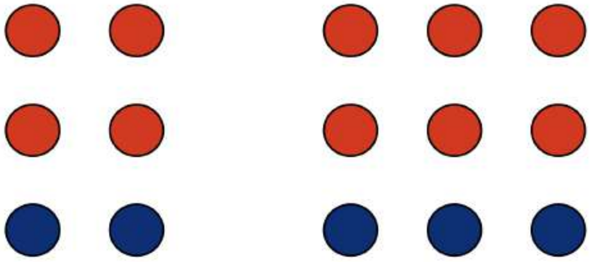
# Quiz: Cluster the following data



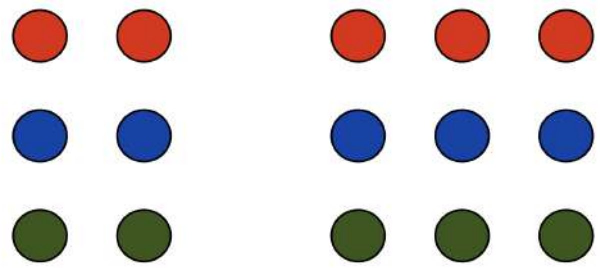
# Quiz: Cluster the following data



# Quiz: Cluster the following data



# Quiz: Cluster the following data



How many clusters?

# General Remarks

- A single dataset can be clustered into several ways
- There is no single right or wrong clustering
  - Simply different views of the same data
- how to measure the quality of clustering algorithm?
  - Two ways
    - Compare clusters produced by clustering algorithm against some reference (gold standard) set of clusters (direct evaluation)
    - Use the clusters for some other (eg. supervised learning) task and measure the difference in performance of the second task (indirect evaluation)

# Clustering as Optimisation

- Given a dataset $\{\mathbf{x}_1, \ldots \mathbf{x}_N\}$ of $N$ instances represented as $d$ dimensional real vectors ($\mathbf{x}_i \in \mathbb{R}^d$), partition these N instances into $k$ clusters $S_1, \ldots S_k$ such that some objective function $f(S_1, \ldots S_k)$ is minimised.
- Observations
  - $k$ and $f$ are given
  - $f$ can be similarity between the clusters (good to create dissimilar clusters as much as possible), information gain, correlation and various other such goodness measures (heuristics)

# Partioning - k-means algorithm

$$\arg\min_{S_1,\ldots,S_k} \sum_{i=1}^{k} \sum_{\boldsymbol{x}_j \in S_i} ||\boldsymbol{x}_j - \boldsymbol{\mu}_i||^2$$

We want to minimize the distance between data instances ($x_j$) and some cluster centres ($\mu_i$)

$$f(S_1,\ldots,S_k) = \sum_{i=1}^{k} \sum_{\boldsymbol{x}_j \in S_i} ||\boldsymbol{x}_j - \boldsymbol{\mu}_i||^2$$

This objective function is called the *within cluster sum of squares* (WCSS) objective

# Partioning - cluster centroids

$$\frac{\partial f(S_1, \ldots, S_k)}{\partial \mu_i} = 0$$

$$\frac{\partial f(S_1, \ldots, S_k)}{\partial \mu_i} = \sum_{\boldsymbol{x}_j \in S_i} 2(\boldsymbol{x}_j - \boldsymbol{\mu}_i)$$

$$\mu_i = \frac{1}{|S_i|} \sum_{\boldsymbol{x}_j \in S_i} \boldsymbol{x}_j$$

Just compute the centroid (mean) of each cluster and that will give you the cluster centers

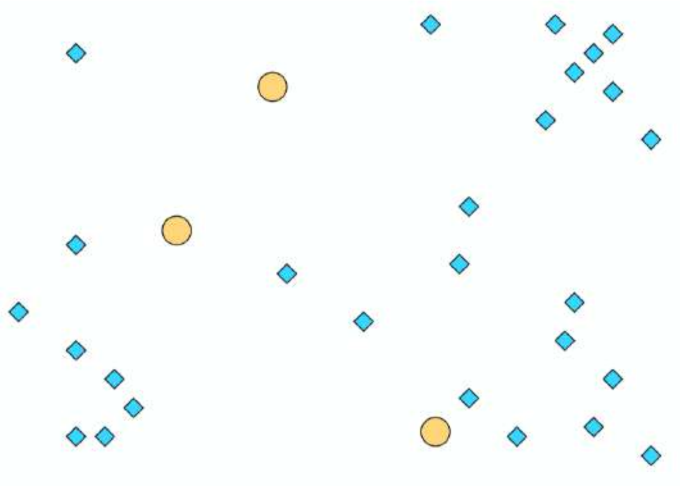
# k-Means clustering

- Input
  - The number of clusters $k$
  - Dataset $\{\mathbf{x}_1, \ldots \mathbf{x}_N\}$ of $N$ instances represented as $d$ dimensional real vectors ($\mathbf{x}_i \in \mathbb{R}^d$)

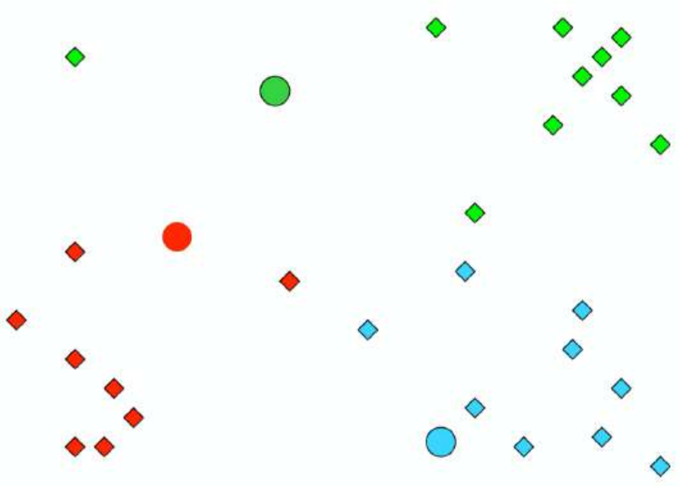
1. Set $k$ instances from the dataset randomly (initial cluster means / centers)
2. Assign all other instances to the closest cluster centre
3. Compute the mean of each cluster
4. Until convergence repeat between steps 2 and 3

convergence = no instances have moved among clusters (often after a fixed number of iterations specified by the user)

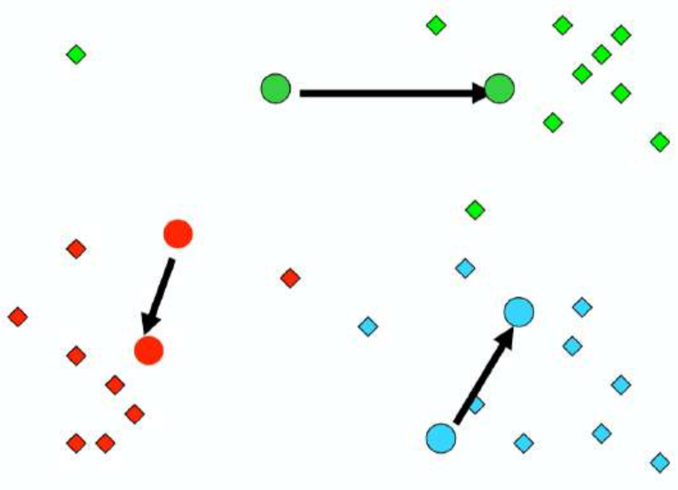
# k-means clustering



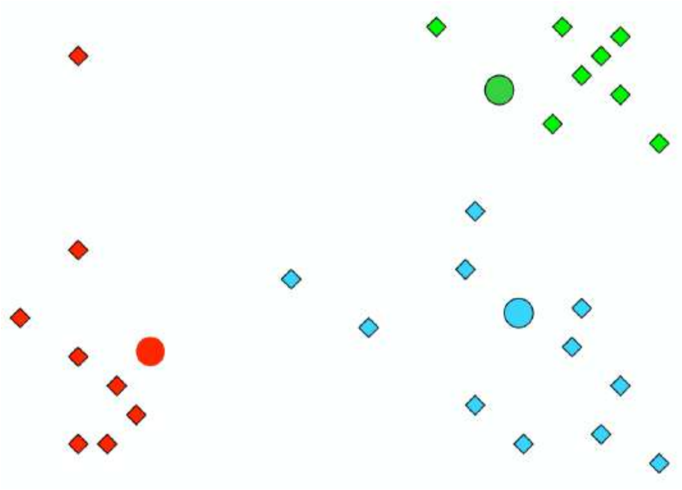
# k-means clustering



# k-means clustering



# k-means clustering



# Quiz: k-means clustering

- Given five data points: $\{(0,0),(1,0),(1,1),(0,1),(-1,0)\}$
- Create two clusters $K = 2$: $(c_1$ and $c_2)$
- Choose $x_2 = (1,0)$ and $x_3 = (1,1)$ as initial centroids
- Use Euclidean Distance as the similarity metric

# Quiz: k-means clustering (soln)

| | | $c_1$ (1,0) | $c_2$ (1,1) | Assignment |
|---|---|---|---|---|
| $x_1$ | 0,0 | $\sqrt{(0-1)^2+(0-0)^2}=\sqrt{1}$ | $\sqrt{(0-1)^2+(0-1)^2}=\sqrt{2}$ | $c_1$ |
| $x_2$ | 1,0 | $\sqrt{(1-1)^2+(0-0)^2}=\sqrt{0}$ | $\sqrt{(1-1)^2+(0-1)^2}=\sqrt{1}$ | $c_1$ |
| $x_3$ | 1,1 | $\sqrt{(1-1)^2+(1-0)^2}=\sqrt{1}$ | $\sqrt{(1-1)^2+(1-1)^2}=\sqrt{0}$ | $c_2$ |
| $x_4$ | 0,1 | $\sqrt{(0-1)^2+(1-0)^2}=\sqrt{2}$ | $\sqrt{(0-1)^2+(1-1)^2}=\sqrt{1}$ | $c_2$ |
| $x_5$ | -1,0 | $\sqrt{(-1-1)^2+(0-0)^2}=\sqrt{4}$ | $\sqrt{(-1-1)^2+(0-1)^2}=\sqrt{5}$ | $c_1$ |

- $c_1 = \{x_1, x_2, x_5\}; c_2 = \{x_3, x_4\}$
- $c_1 = \{(0,0),(1,0),(-1,0)\}; c_2 = \{(1,1),(0,1)\}$
- $\mu_{c_1} = (0,0); \mu_{c_2} = (0.5,1)$
- computing clusters using new $\mu$ gives the same clusters

# Evaluating Clustering - Purity

- Purity is an external evaluation criterion for cluster quality
- It gives the percentage of total number of items that were classified correctly.
- range $[0, 1]$

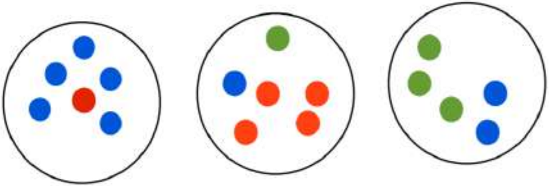
# Evaluating Clustering - Purity

- Let us assume we have a set $\Omega = \{\omega_1, \ldots, \omega_K\}$ clusters for a set of classes $C = \{c_1, \ldots, c_j\}$.
- Purity measures the ratio of the items that are in the cluster with the same class of its own.

$$\text{purity}(\Omega, C) = \frac{1}{N} \sum_k \max_j \left| \omega_k \bigcap c_j \right|$$

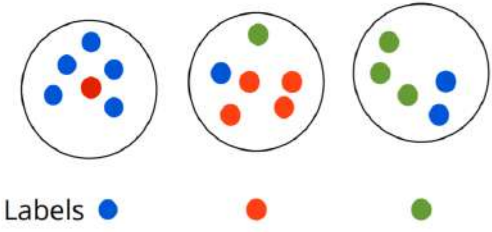
$N$ is the number of items, $\omega_k$ is a cluster in $\Omega$ and and $c_j$ is the class which has the maximum count for cluster $\omega_k$.

# Purity



Quiz: Compute purity for this clustering.

# Purity




purity = (5 + 4 + 3) / 17 = 12/17 = 0.71

Purity achieves its maximum value of 1 for singletons (each item is in a cluster containing only that single item)! Obviously this is not good "clustering" and purity does not recognise this.

# Purity

- bad clusterings have purity values close to 0
- perfect clustering has a purity of 1
- high purity is easy to achieve when the number of clusters is large
- particularly, purity is 1 is each item (singleton) gets its own cluster
- thus, purity cannot be used to trade off the quality of clustering against number of clusters.

# Evaluating Clustering - NMI

- Let us assume we have a set $\Omega = \{\omega_1, \ldots, \omega_K\}$ clusters for a set of classes $C = \{c_1, \ldots, c_j\}$.
- Normalised Mutual Information (NMI) computes the ratio of information that we can know about the classes $C$ given the clusters $\Omega$ to the averaged information that is contained in $C$ and $\Omega$.

# Evaluating Clustering - NMI

- NMI is computed using:

$$\mathrm{NMI}(\Omega, \mathcal{C}) = \frac{I(\Omega, \mathcal{C})}{[H(\Omega) + H(\mathcal{C})]/2}$$

where,

  - $\Omega$ = set of clusters
  - $\mathcal{C}$ = set of classes
  - $I(\Omega, \mathcal{C})$ = mutual information between $\Omega$ and $\mathcal{C}$
  - $H(.)$ = Entropy

# Evaluating Clustering - NMI

- Mutual information $I(\Omega, \mathcal{C})$ is given by:

$$\boxed{\begin{aligned} I(\Omega, \mathcal{C}) &= \sum_k \sum_j p(\omega_k \cap c_j) \log \left( \frac{p(\omega_k \cap c_j)}{p(\omega_k) p(c_j)} \right) \\ &= \sum_k \sum_j \frac{|\omega_k \cap c_j|}{N} \log \left( \frac{N |\omega_k \cap c_j|}{|\omega_k||c_j|} \right) \end{aligned}}$$

where,

  - $P(\omega_k)$ = probability of an object being in cluster $\omega_k$
  - $P(c_j)$ = probability of an object being in class $c_j$
  - $P(\omega_k \cap c_j)$ = probability of an object being in the intersection of $\omega_k$ and $c_j$

# NMI - Mutual Information

- mutual information measures the amount of information by which our knowledge about the classes increases when we are told about what clusters are.
- minimum of mutual information is 0
  - clustering is random w.r.t class
  - knowing an item in a cluster does not give any information about what classes could be
- mutual information achieves maximum for clustering that perfectly recreates the classes

# NMI - Mutual Information

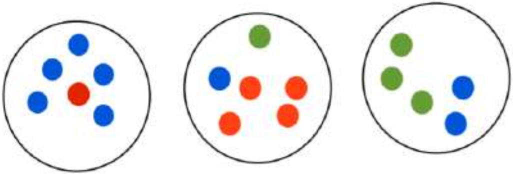
- clustering $K = N$ (one-document clusters) has maximum MI.
- thus MI has same problem of purity
- cannot penalise large cardinalities
- fewer clusters are better

# Entropy

$$\text{NMI}(\Omega, \mathcal{C}) = \frac{I(\Omega, \mathcal{C})}{[H(\Omega) + H(\mathcal{C})]/2}$$

- the entropy function used in the denominator of NMI fixes this problem.
- entropy tends to increase with different number of clusters
- for example $H(\Omega)$ reaches its maximum $\log N$ for $K = N$, which ensures NMI is low for $K = N$
- NMI is always a number between 0 and 1

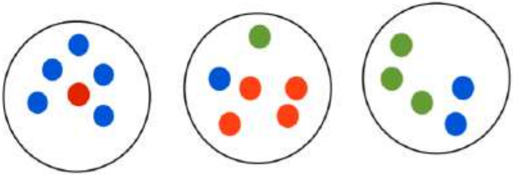
# Calculating NMI for Clustering



- Given C $=$ 3 classes
- Let $c_1 =$ Blue, $c_2 =$ Red and $c_3 =$ Green
- Thus, $P(c_1) = \frac{8}{17}$, $P(c_2) = \frac{5}{17}$, $P(c_3) = \frac{4}{17}$
- Entropy of class: $H(C) = -\sum_{i=1}^{3} P(c_j) \log P(c_j)$

$= -[\frac{8}{17} \log \frac{8}{17} + \frac{5}{17} \log \frac{5}{17} + \frac{4}{17} \log \frac{4}{17}] = 1.055$

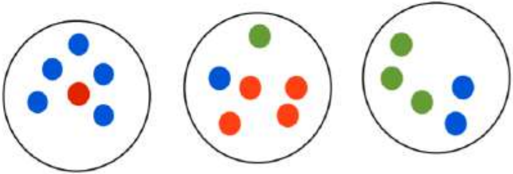
# Calculating NMI for Clustering



- Likewise, $P(w_1) = \frac{6}{17}$, $P(w_2) = \frac{6}{17}$, $P(w_3) = \frac{5}{17}$

$$= -[\tfrac{6}{17} \log \tfrac{6}{17} + \tfrac{6}{17} \log \tfrac{6}{17} + \tfrac{5}{17} \log \tfrac{5}{17}] = 1.095$$

# Calculating NMI for Clustering



- $P(w_1 \cap c_1) = \frac{5}{17}$, $P(w_1 \cap c_2) = \frac{1}{17}$, $P(w_1 \cap c_3) = \frac{0}{17}$
- $P(w_2 \cap c_1) = \frac{1}{17}$, $P(w_2 \cap c_2) = \frac{4}{17}$, $P(w_2 \cap c_3) = \frac{1}{17}$
- $P(w_3 \cap c_1) = \frac{2}{17}$, $P(w_3 \cap c_2) = \frac{0}{17}$, $P(w_3 \cap c_3) = \frac{3}{17}$

# Calculating NMI for Clustering

- Mutual information $I(\Omega, \mathcal{C})$ is given by:

$$\boxed{\begin{aligned} I(\Omega, \mathcal{C}) &= \sum_k \sum_j p(\omega_k \cap c_j) \log\left(\frac{p(\omega_k \cap c_j)}{p(\omega_k)p(c_j)}\right) \\ &= \sum_k \sum_j \frac{|\omega_k \cap c_j|}{N} \log\left(\frac{N|\omega_k \cap c_j|}{|\omega_k||c_j|}\right) \end{aligned}}$$

- substituting the values, we get $I(\Omega, \mathcal{C}) = 0.4496$
- Finally NMI is given by:

$$\mathrm{NMI}(\Omega, \mathcal{C}) = \frac{I(\Omega, \mathcal{C})}{[H(\Omega) + H(\mathcal{C})]/2}$$

- thus, $NMI(\Omega, C) = \frac{0.4496}{1.055+1.095} = 0.4182$

# NMI

- NMI is a good measure for determining the quality of clustering
- it is an external measure as we need class labels of instances to determine NMI
- Since it is normalised, NMI between clusterings having different number of clusters can be measured.

# Rand Index (RI)

- RI is a metric used to evaluate the quality of clustering technique
- RI measures the percentage of decisions that are correct
- decision - assigning a pair of data points to a cluster
- Total number of decisions: $\frac{N(N-1)}{2}$, where N is the total number of data points
- RI is given by: $\frac{TP+TN}{TP+FP+TN+FN}$

# Rand Index (RI)

- TP = No. of item pairs that are in the same cluster and belong to the same class
- FP = No. of item pairs that are in the same cluster but belong to different classes
- TN = No. of item pairs that are in different clusters and belong to different classes
- FN = No. of item pairs that are in different clusters but belong to the same class

# Rand Index (RI)

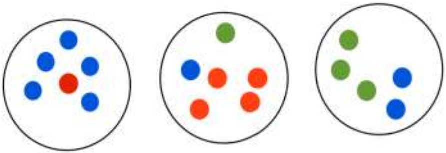
## Contingency Table

| contingency table | same cluster | different clusters |
|---|---|---|
| same class | TP | FN |
| different classes | FP | TN |

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}$$

(accuracy of the clustering)

# Compute Rand Index (RI)



Quiz: Compute RI for this clustering.

# Compute Rand Index (RI)

- Three classes: blue, red, green
- Total items: 17
- Total number of pairs:

$$\frac{N(N-1)}{2} = \frac{17(17-1)}{2} = 136 \tag{1}$$

# Compute Rand Index (RI)

- To start, let us compute Total Positives = TP+FP

TP+FP $= {}^6C_2 + {}^6C_2 + {}^5C_2$

${}^nC_r = \frac{n!}{r!(n-r)!}$. Thus ${}^6C_2 = \frac{6!}{2!(6-2)!} = \frac{6\times5}{2} = 15$;
${}^5C_2 = \frac{5!}{2!(5-2)!} = \frac{5\times4}{2} = 10$

TP + FP = 15 + 15 + 10 = 40

# Compute Rand Index (RI)

- To start, let us compute TP

  $$TP = {}^{5}C_2 + {}^{4}C_2 + {}^{3}C_2 + {}^{2}C_2$$

  $$TP = 10 + 6 + 3 + 1 = \mathbf{20}$$

- Thus, FP = 40 -20 = **20**

# Compute Rand Index (RI)

- let us calculate negatives!

  Total Negatives = Total Pairs - Total Positives (TP + FP)

  Total Negatives = 136 - 40 = 96

- FN is calculated by looking at pairs that should be grouped together but are not!

  $$FN = [(3 \times 5) + (1 \times 2)] + (1 \times 4) + (1 \times 3) = \mathbf{24}$$

- TN = Total Negatives - FN = 96 -24 = **72**

# Compute Rand Index (RI)

| | same cluster | different clusters |
|---|---|---|
| same class | 20 | 24 |
| different classes | 20 | 72 |

$$\begin{aligned} RI &= (20+72) / (20+24+20+72) \\ &= 0.676 \end{aligned}$$

# Evaluating Clustering - P/R/F

- We can use Precision (P), Recall (R), and F-measure (F) at to evaluate the accuracy of a clustering.
- For this purpose we must first create the contingency table as we did for RI and then compute P, R, F as follows

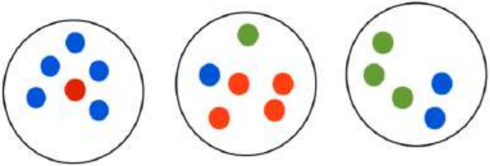
$$P = TP / (TP + FP)$$

$$R = TP / (TP + FN)$$

$$F = 2PR / (P + R)$$

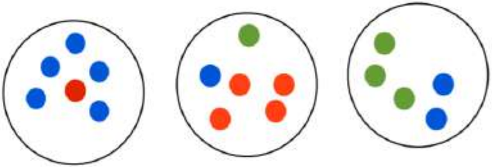
Ref: https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

# Evaluating Clustering - P/R/F



Quiz: Compute P/R/F for this clustering.

# Evaluating Clustering - P/R/F



| | same cluster | different clusters |
|---|---|---|
| same class | TP=20 | FN=24 |
| different classes | FP=20 | TN=72 |

$$P = TP / (TP + FP) = 20 / (20+20) = 0.5$$

$$R = TP / (TP + FN) = 20 / (20 + 24) = 0.45$$

$$F = 2PR / (P + R) = 0.47$$

# B-CUBED Measure

- Proposed in (Bagga B. Baldwin = B³)
  - A. Bagga and B. Baldwin. Entity-based cross document coreference resolution using the vector space model, In Proc. of 36th COLING-ACL, pages 79--85, 1998.
- We would like to evaluate clustering without labelling any clusters.

$$\text{precision}(x) = \frac{\text{No. of items in C(x) with A(x)}}{\text{No. of items in C(x)}}$$

$$\text{recall}(x) = \frac{\text{No. of items in C(x) with A(x)}}{\text{Total no. of items with A(x)}}$$

C(x): The ID of the cluster that x belongs to

A(x): label of x

# B-CUBED Measure

- Compute the average over all the items (instances) that appear in all clusters (N)

$$\text{Precision} = \frac{1}{N} \sum_{p \in DataSet} \text{Precision}(p)$$

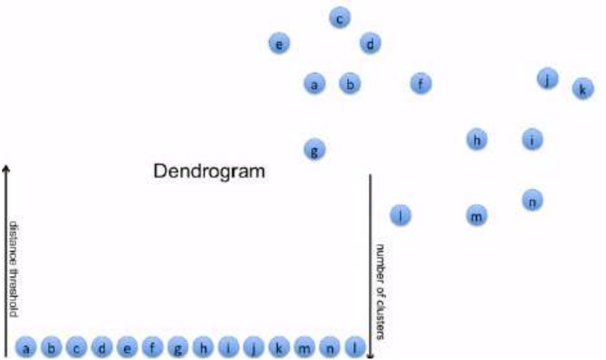
$$\text{Recall} = \frac{1}{N} \sum_{p \in DataSet} \text{Recall}(p)$$

$$F\text{-Score} = \frac{1}{N} \sum_{p \in DataSet} \text{F}(p)$$

# Hierarchical Clustering

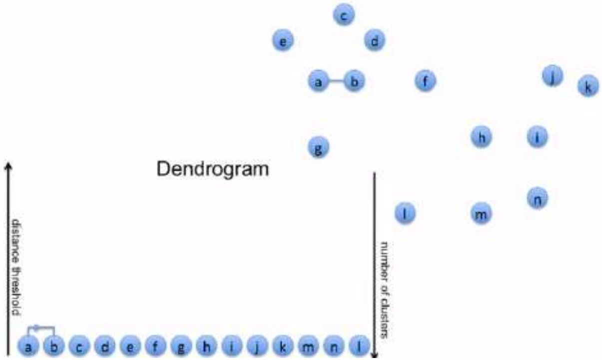
- Sometimes we might want to organise the data into a hierarchy of subsuming concepts for visualisation (abstraction) purposes
- Two methods exists
  - Conglomerative clustering
    - Start from one big cluster with all data instances and repeatedly partition it
    - Top-down approach
  - Agglomerative clustering
    - Start singletons (clusters with exactly one instance) and iteratively merge the most *similar* two clusters
      - Bottom-up approach
      - computationally more efficient (O(log$n$) merges required )

# Hierarchical Clustering (example)



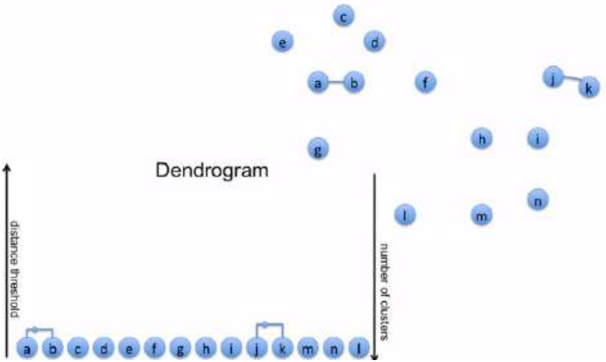

Source: Victor Lavrenko

# Hierarchical Clustering (example)



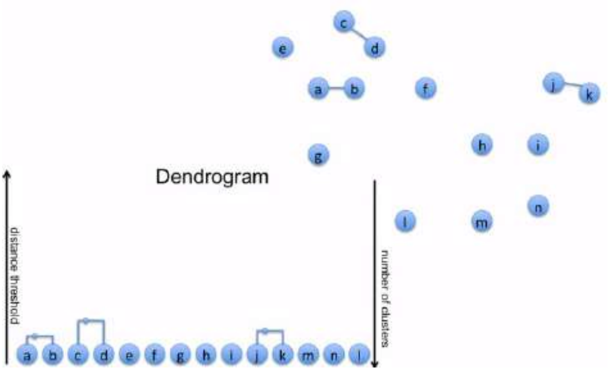

Source: Victor Lavrenko

# Hierarchical Clustering (example)



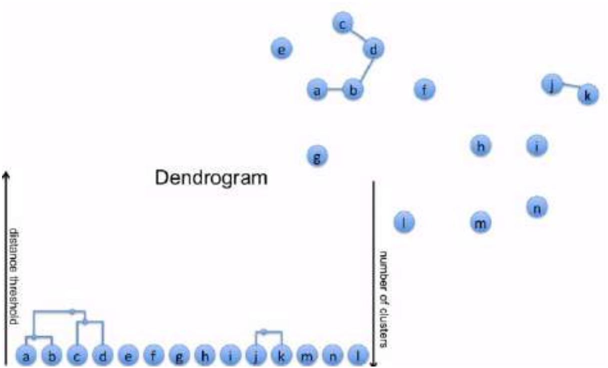

Source: Victor Lavrenko

# Hierarchical Clustering (example)



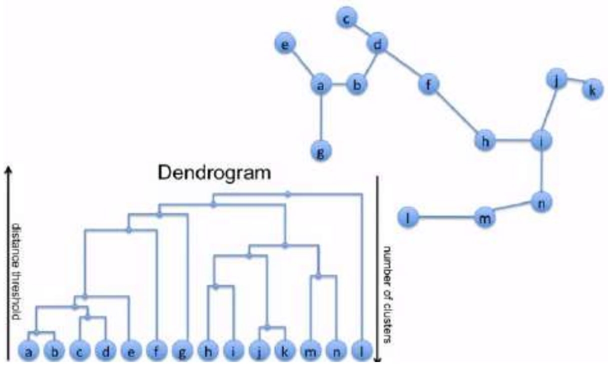

Source: Victor Lavrenko

# Hierarchical Clustering (example)



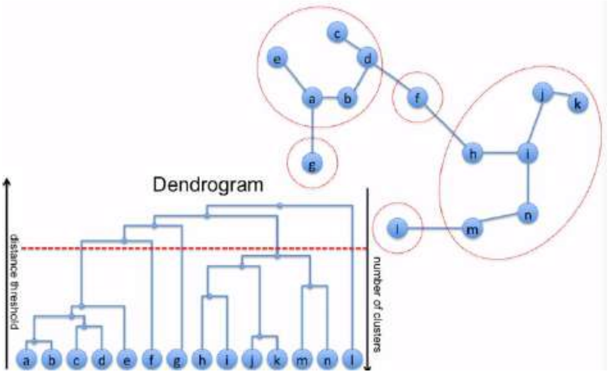

Source: Victor Lavrenko

# Hierarchical Clustering (example)




Source: Victor Lavrenko

# Hierarchical Clustering (example)




Source: Victor Lavrenko