Lecture 12 – Clustering

Bulat Ibragimov

bulat@di.ku.dk

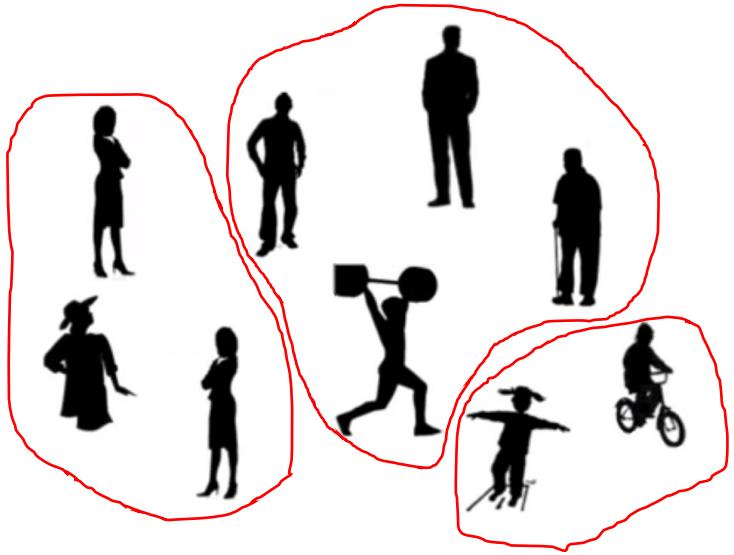
Department of Computer Science
University of Copenhagen

UNIVERSITY OF COPENHAGEN

Clustering

- Unsupervised
  - Try to understand underlying structure of the data and nothing specific

- What sub-populations exist in the data?
  - How many clusters?
  - How big?
  - Outliers?



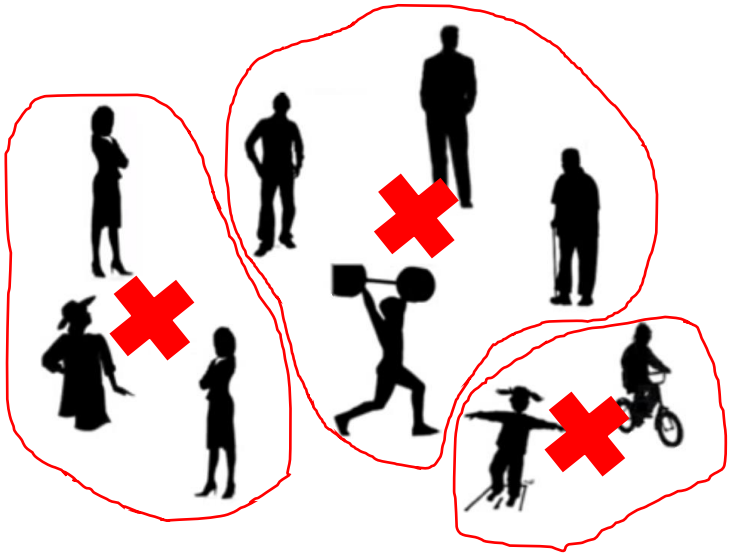
http://homepages.inf.ed.ac.uk/vlavrenk

Clustering types

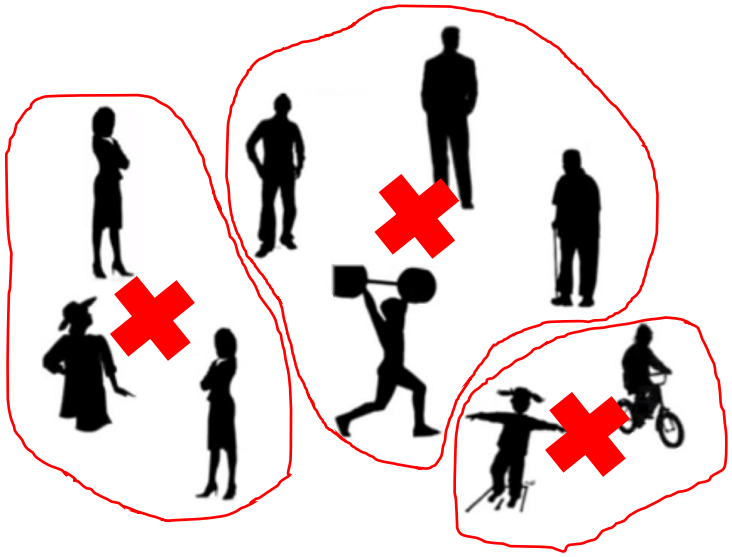
- Goal
  - Monothetic: looking for specific properties for cluster members
    - e. g. all people younger than 15 – cluster of kids
  - Polythetic: cluster members are similar to each other
    - We compute distance between elements for clustering
- Overlap
  - Hard clustering – no overlap is allowed
  - Soft clustering – estimate “membership strength” for datapoints
- Depth
  - Flat
  - Hierarchy

k-mean clustering

- Properties
  - Polythetic clustering
  - Data partitioned into k sub-populations (k must be specified)
  - Datapoints in sub-population are similar to its "centroid"
  - Clusters are hard
  - Clusters form flat structure



k-mean clustering



- Input: K, set of points $x_1, \dots, x_n$
- Generate random $c_1, \dots, c_k$ centroids
- Repeat:
  - Assign each x_i to the nearest centroid c_j $\mathbf{argmin}_{j} \boldsymbol{D}(\boldsymbol{x_i}, \boldsymbol{c_j})$
  - Compute new centroids $\boldsymbol{c_j} \leftarrow \frac{\mathbf{1}}{\boldsymbol{n_j}} \sum_{\boldsymbol{x_i} \in \boldsymbol{c_j}} \boldsymbol{x_i}$
- Stop when assignment of x_i does not change

Distance metric

- Selecting a suitable distance metric is paramount for success

Euclidian distance:

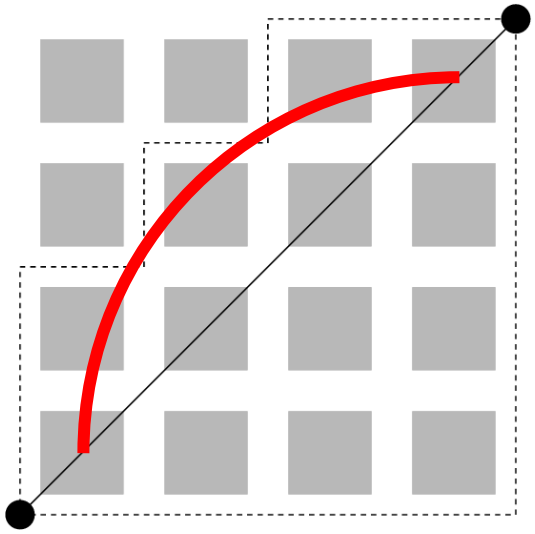
$$D(x, x') = \sqrt{\sum_d |x_d - x'_d|^2}$$

Manhattan distance:

$$D(x, x') = \sum_d |x_d - x'_d|$$

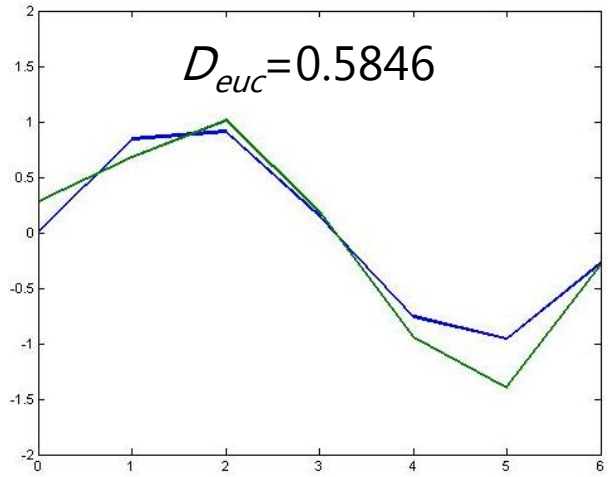
Logical distance (categorical attributes):

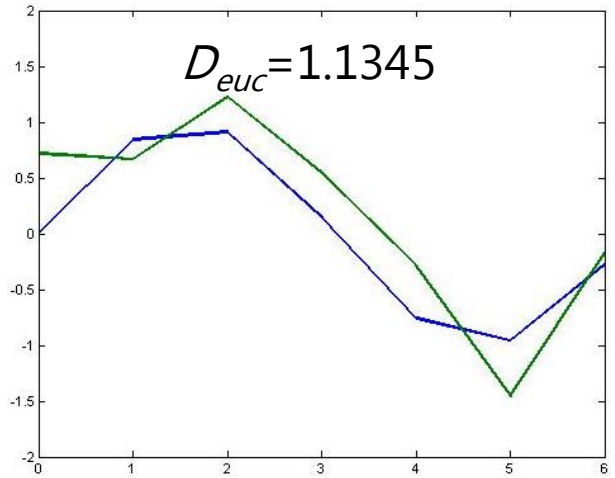
$$D(x, x') = \sum_d 1_{x_d \neq x'_d}$$

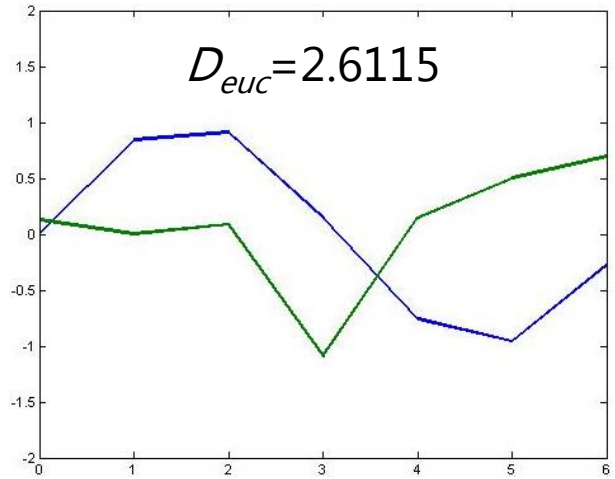


Distance metric

- The Euclidian distance works well most of the times:

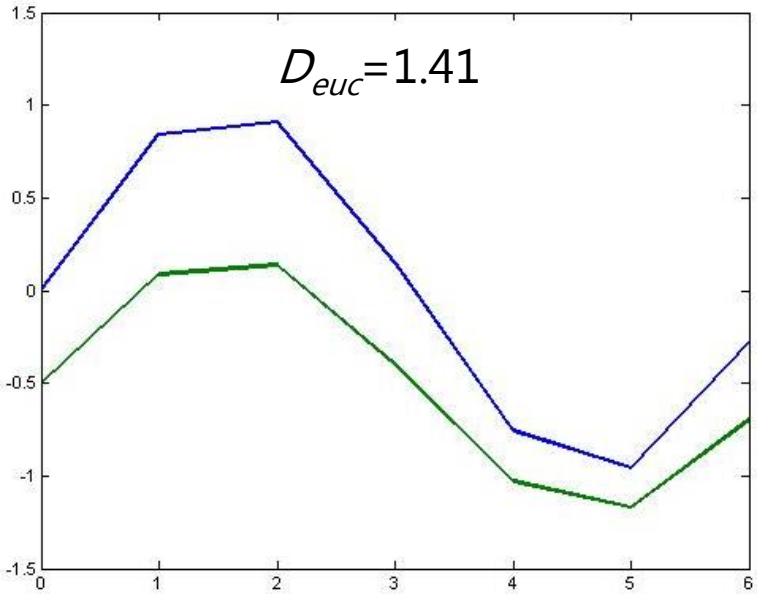


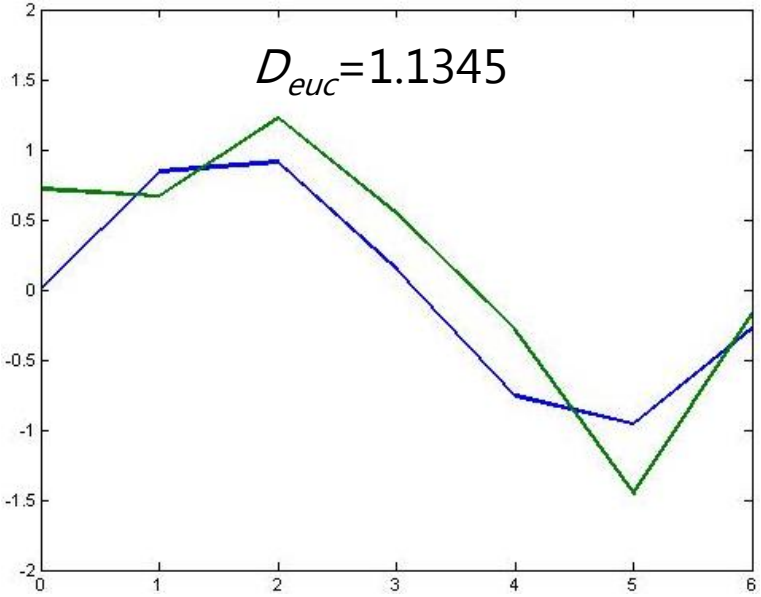




Distance metric

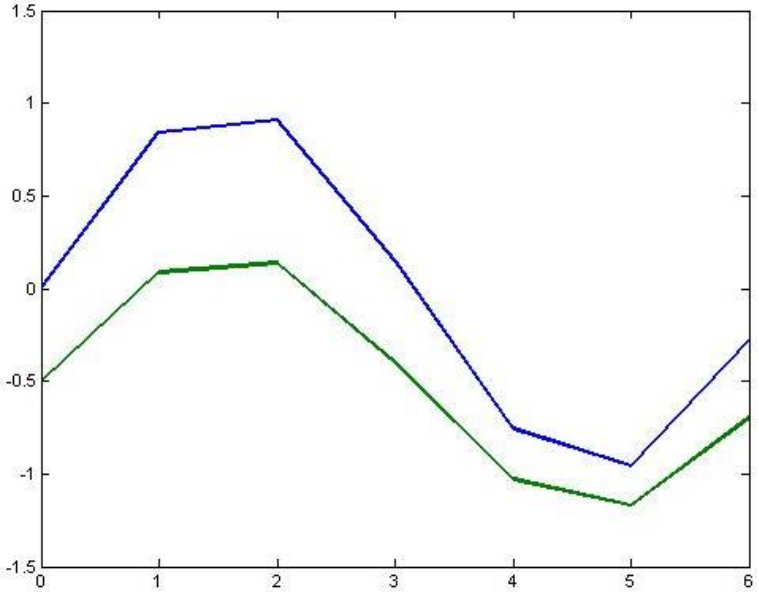
- Let's compare these two examples.
- Is the left one worse than the right one?

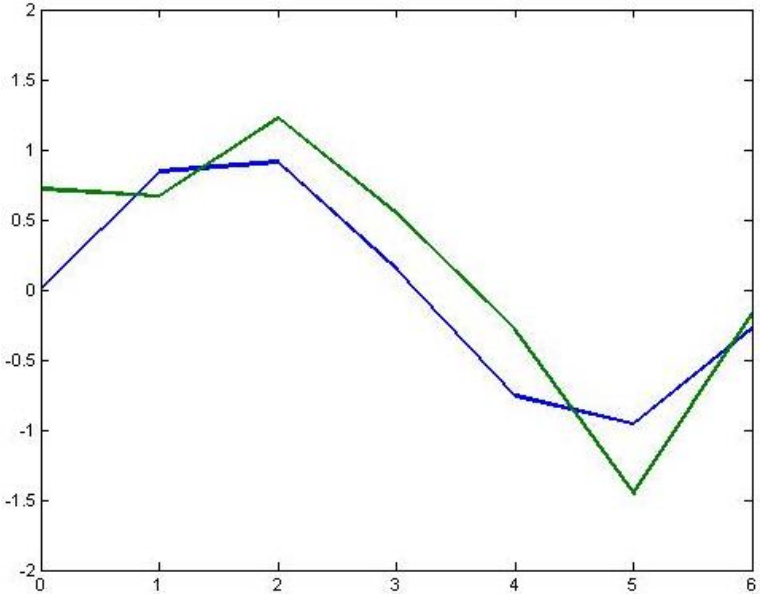




Distance metric: Pearson correlation coefficient

- We can put stress on matching patterns in features but not the actual values.
- Do features go “up” and “down” together for two data





Distance metric: Pearson correlation coefficient

- We're shifting the expression profiles down (subtracting the means) and scaling by the standard deviations (i.e., making the data have mean = 0 and std = 1)

$$\rho(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

$$\bar{x} = \frac{1}{n}\sum_{i}^{n} x_i$$

$$\bar{y} = \frac{1}{n}\sum_{i}^{n} y_i$$

Distance metric: Pearson correlation coefficient

- Pearson linear correlation (PLC) is a measure that is invariant to scaling and shifting (vertically) of the expression values

- Always between –1 and +1 (perfectly anti-correlated and perfectly correlated)

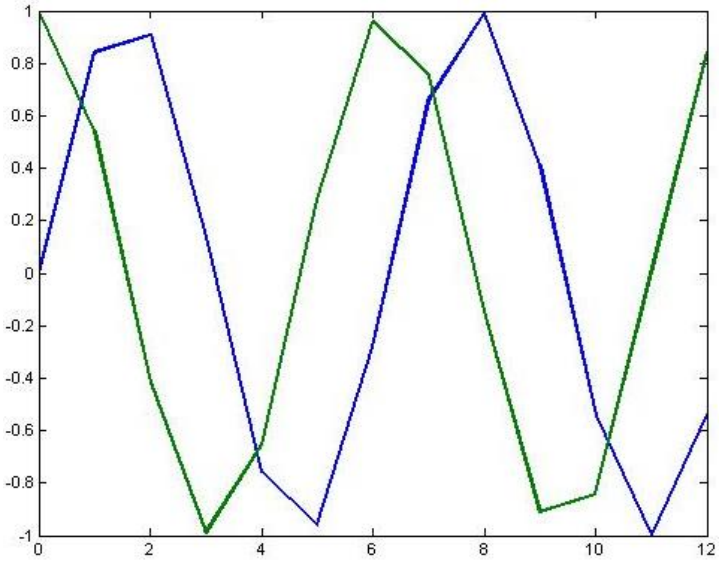
- This is a similarity measure, but we can easily make it into a dissimilarity measure:

$$d_p = \frac{1 - \rho(\mathbf{x}, \mathbf{y})}{2}$$

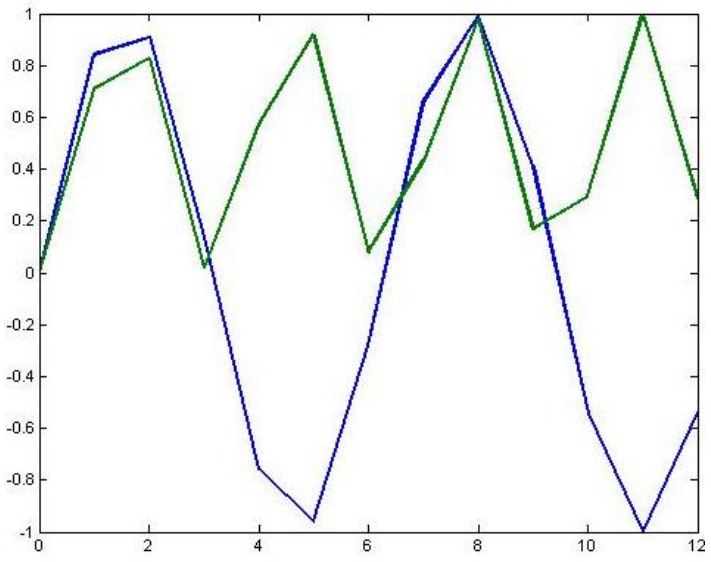
Distance metric: Complex examples

- More sophisticated metrics may be needed to capture more complex relationships between samples

- Hopefully, it is very unlikely



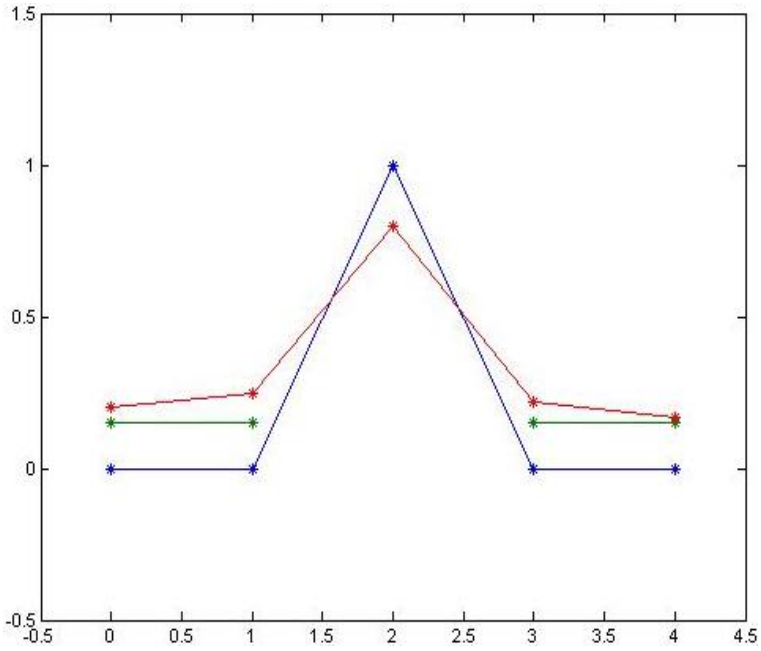
Perfect correlation



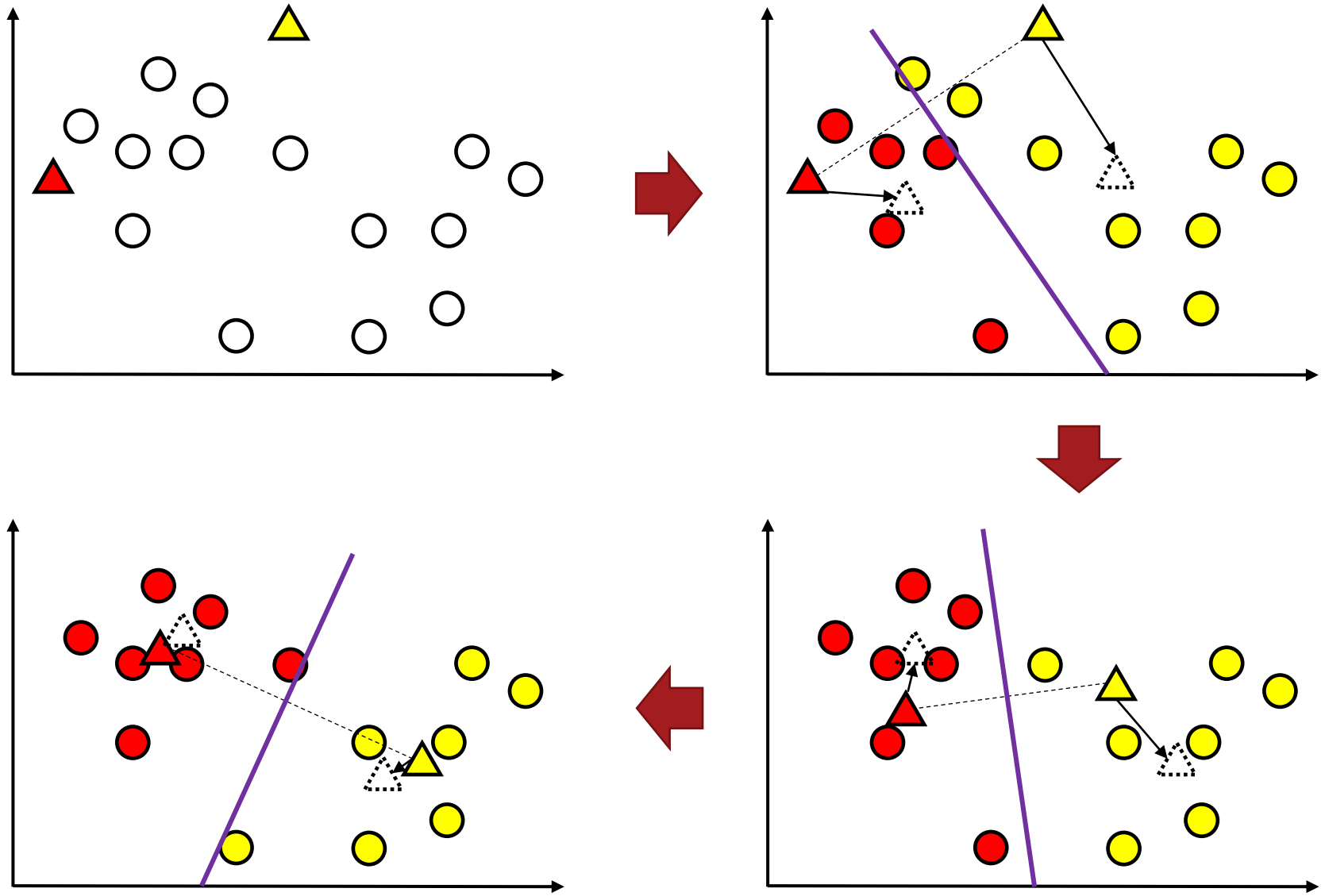
Green = Blue^2

Distance metric: Missing values

- What if some values are missed in the data?
- We can simply ignore a feature, if its value is missed in one sample
- We can train regressors to reconstruct missing feature values
- If we miss a categorical feature, we can duplicate the sample with all possible feature values, cluster them and average the clustering result

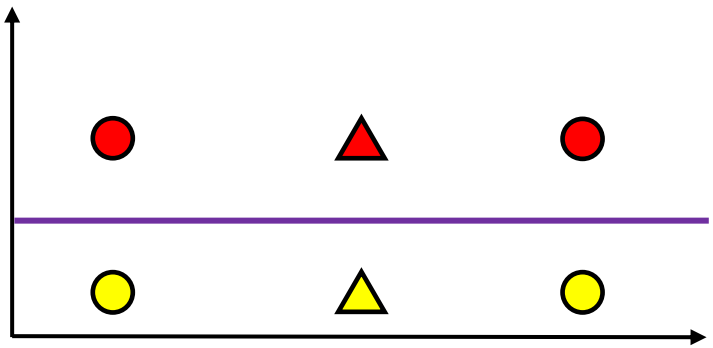


k-mean clustering: visualization



k-mean clustering

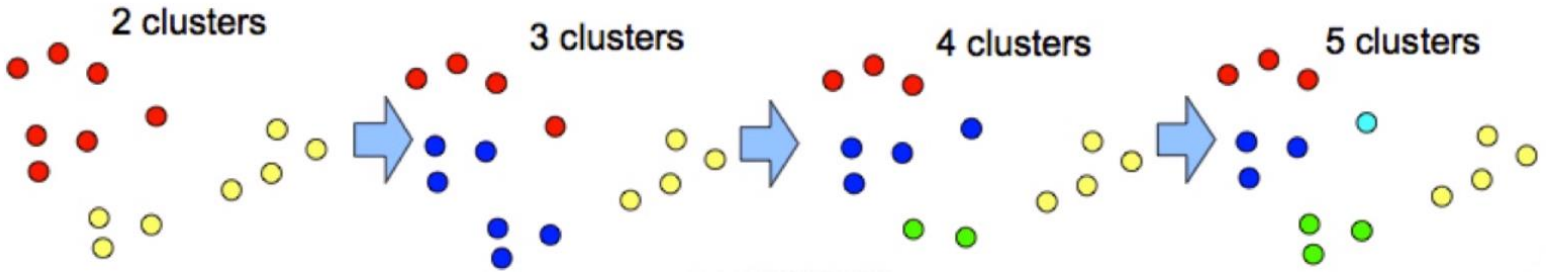
- Solution:
  - Minimized intra-cluster distance $\sum_j \sum_{x_i \in c_j} D(c_j, x_i)^2$
  - Converges to local minimum.
    - Different seed centroids -> different results
    - Run several times and select solution with lowest cost, i.e. intra-cluster distance
  - Nearby points may be assigned to different clusters



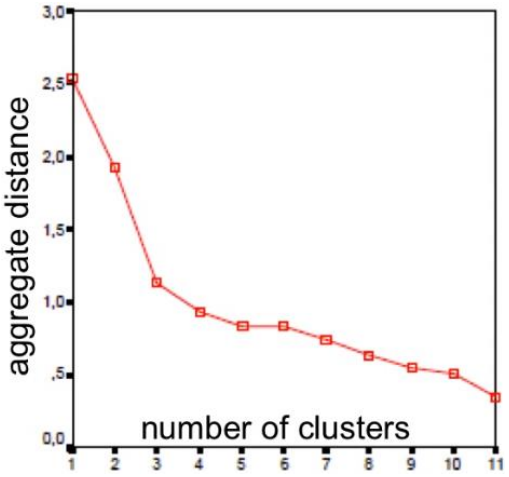
k-mean clustering: optimal k

- How many cluster are in the data:
  - Prior knowledge, e.g. for clustering of digits $k = 10$
  - Try different Ks
    - Record the behavior of cost function

What is the problem with this idea?

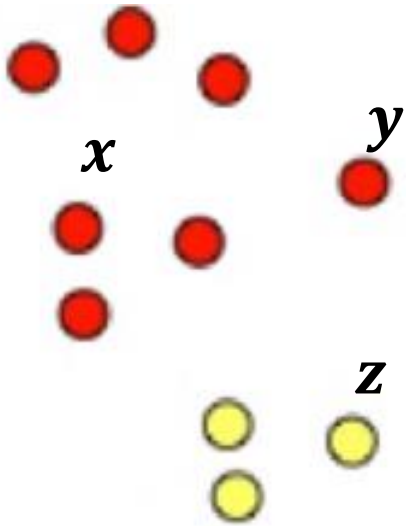


- The cost will always go down. The optimal $k = n$.
- Elbow method



k-mean clustering: silhouette score

- Intra-cluster distance $\boldsymbol{x} \in \boldsymbol{c_j}$:
  - $a(\boldsymbol{x}) = \frac{1}{|\boldsymbol{c_i}|-1} \sum_{\boldsymbol{y} \in \boldsymbol{c_i}} \boldsymbol{D}(\boldsymbol{x}, \boldsymbol{y})$
- Inter-cluster distance $\boldsymbol{x} \in \boldsymbol{c_j}$:
  - $b(\boldsymbol{x}) = \min_{i \neq j} \frac{1}{|\boldsymbol{c_i}|} \sum_{\boldsymbol{z} \in \boldsymbol{c_i}} \boldsymbol{D}(\boldsymbol{x}, \boldsymbol{z})$
- Silhouette:
  - $$s(\boldsymbol{x}) = \begin{cases} 1 - a(\boldsymbol{x})/b(\boldsymbol{x}) & if\ a(\boldsymbol{x}) < b(\boldsymbol{x}) \\ 0 & if\ a(\boldsymbol{x}) = b(\boldsymbol{x}), or\ |\boldsymbol{c_i}| = 1 \\ b(\boldsymbol{x})/a(\boldsymbol{x}) - 1 & if\ a(\boldsymbol{x}) > b(\boldsymbol{x}) \end{cases}$$
- Score $= \underset{x}{\text{mean}}\, s(\boldsymbol{x})$



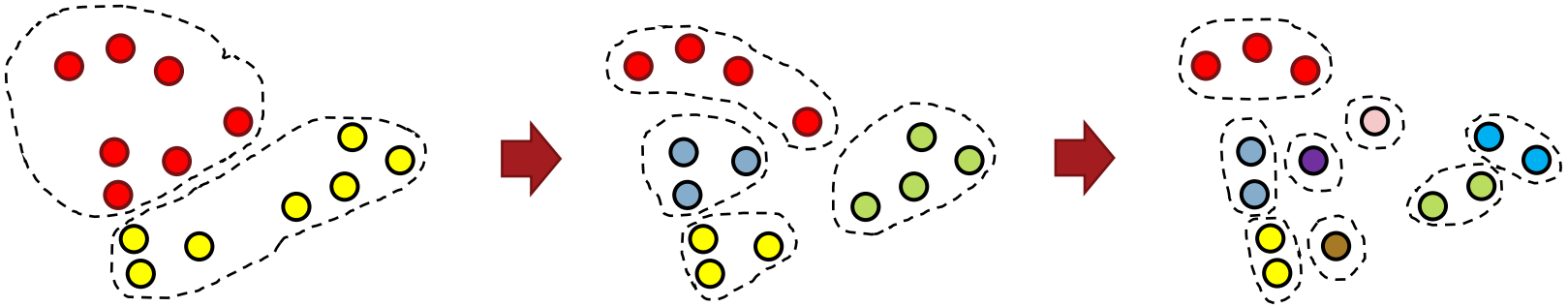
Hierarchical clustering

- Selecting k is difficult, even for a human
- Alternative is to create hierarchy of clusters:
  - Top-down approach: start with one cluster, and recursively split clusters
  - Bottom-up approach: start with individual datapoints, iteratively merge by some criterion

How many clusters?

Hierarchical k-means

- Top-down approach
- Select a relatively low k, e.g. 2.
- Run k-mean clustering on original data $x_1, \dots, x_n$
- For each of the resulting clusters $c_i : i = 1, \dots, k$:
  - Recursively run k-means on centroids in c_i

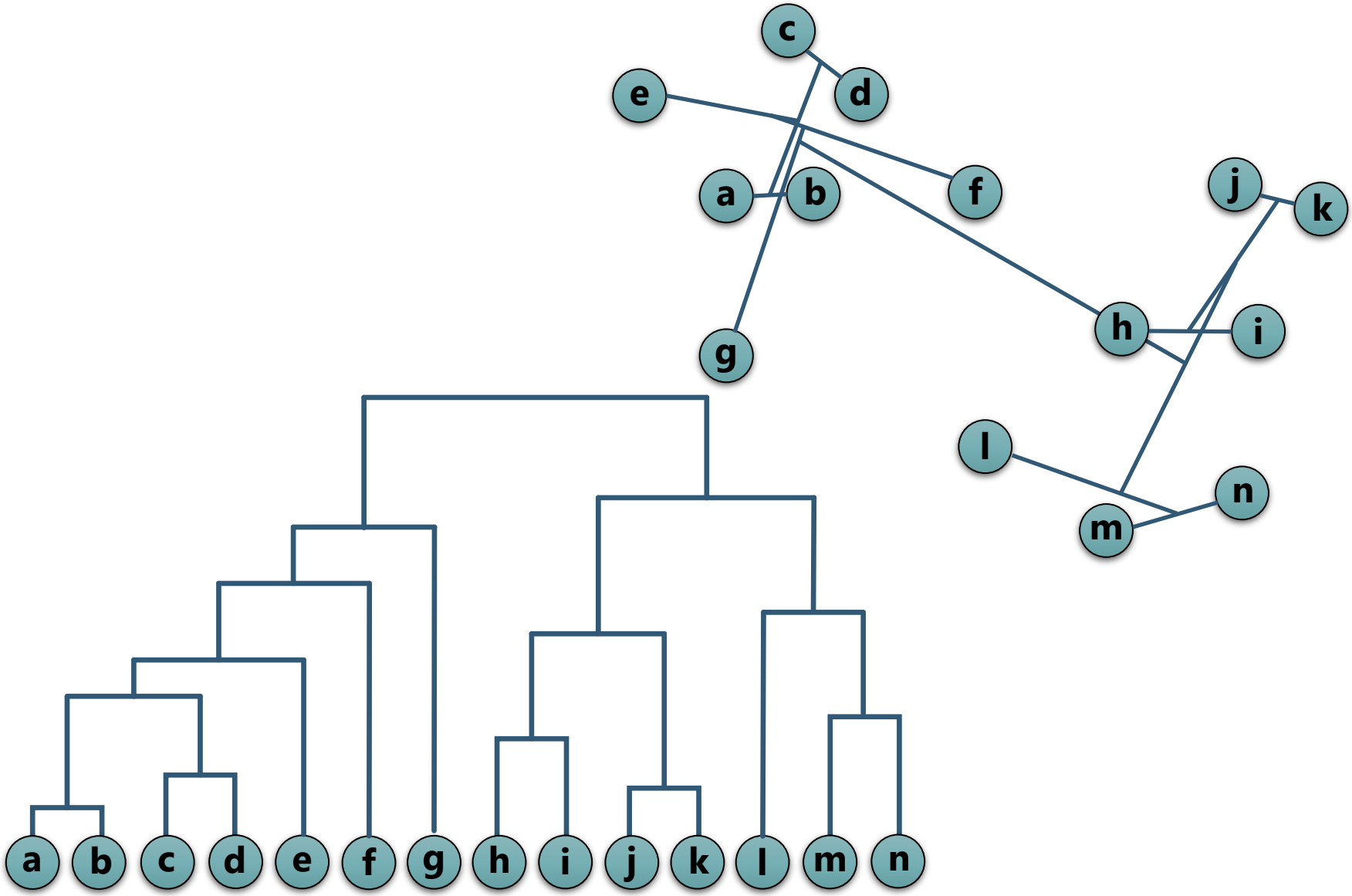


- There is still a problem of close points being assigned to different clusters

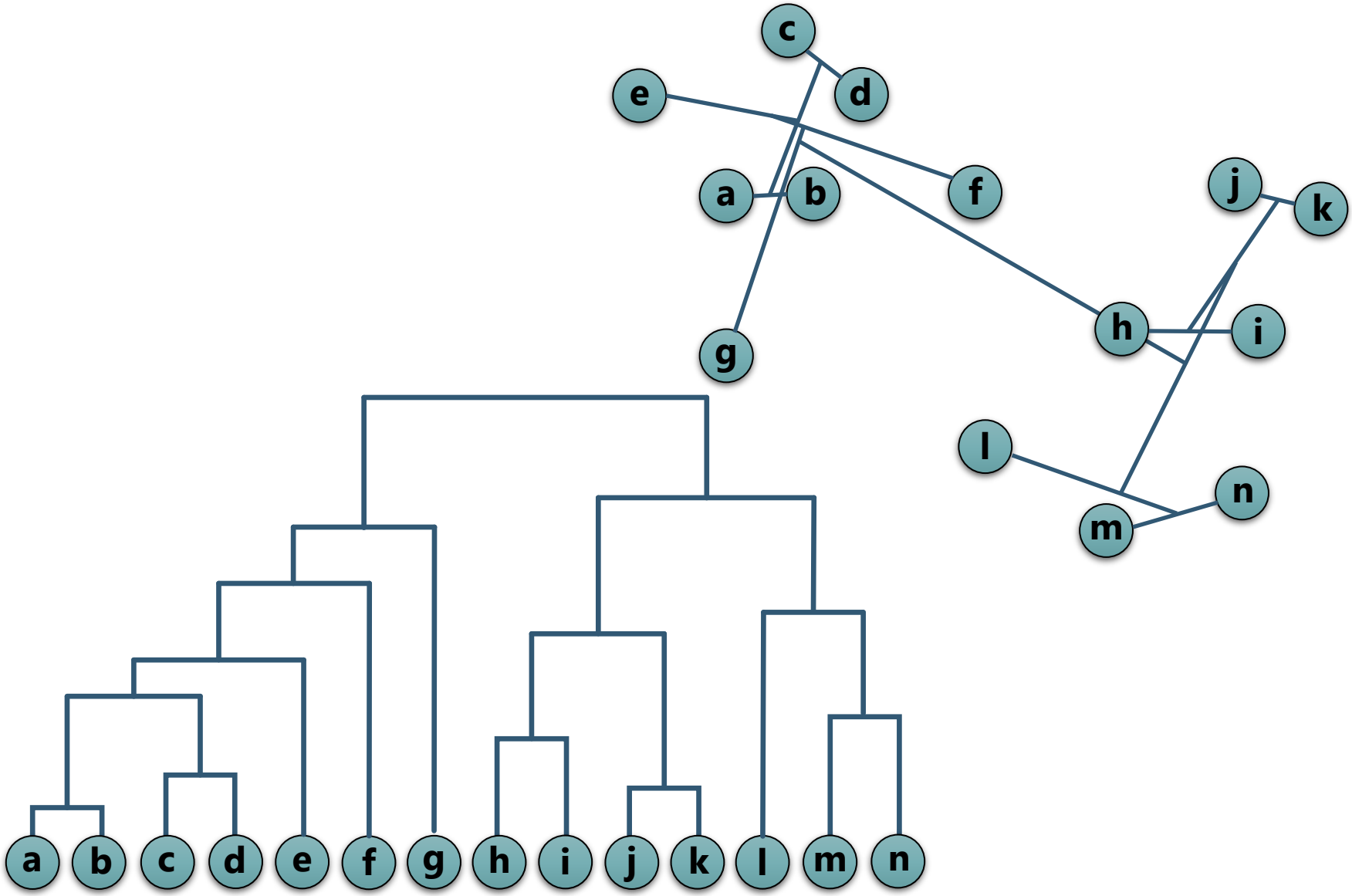
Agglomerative clustering

- Bottom-up approach
- Start with a collection C of n single-datapoint clusters
  - Each cluster contains one single point: $c_i = \{x_i\}$
- Repeat until only one cluster left:
  - Find a pair of clusters that is closest using some **distance** metric
  - Merge clusters c_i, c_j into a new cluster c_r
  - Remove c_i, c_j from the collection C, add c_r
- Slow: $O(n^2 d + n^3)$, because need to recompute distance matrices multiple times
- There is still a problem of close points being assigned to different clusters

Agglomerative clustering: example



Agglomerative clustering: example



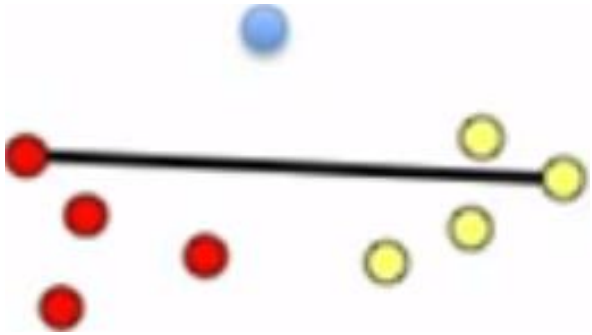
Agglomerative clustering: distance metric

What kind of distance metrics are possible?

- Maximum distance (complete-linkage):

$$D = \max\{d(a,b): a \in c_i, b \in c_j\}$$

- Tends to produce very tight clusters



Agglomerative clustering: distance metric

What kind of distance metrics are possible?

- Minimum distance (single-linkage):

$$D = \min\{d(a, b): a \in c_i, b \in c_j\}$$

- Tends to produce long, "loose" clusters

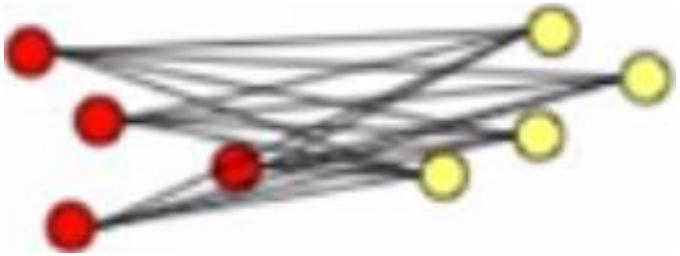
Agglomerative clustering: distance metric

What kind of distance metrics are possible?

- Average linkage distance:

$$D = \frac{1}{|c_i||c_j|} \sum_{a \in c_i} \sum_{b \in c_j} d(a, b)$$

- Tends to produce tight clusters

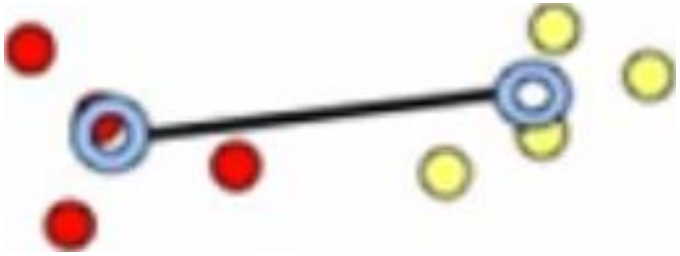


Agglomerative clustering: distance metric

What kind of distance metrics are possible?

- Centroid distance metric:

$$D = \left\| \bar{c}_i - \bar{c}_j \right\|$$

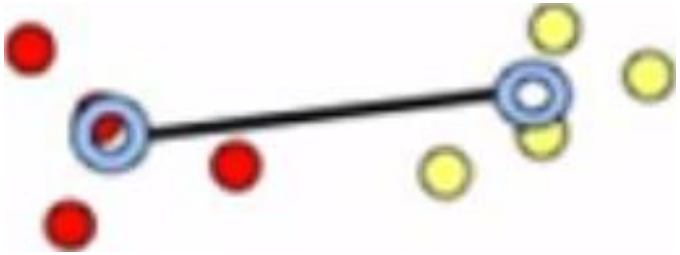


Agglomerative clustering: distance metric

What kind of distance metrics are possible?

- Centroid distance metric:

$$D = \|\bar{c}_i - \bar{c}_j\|$$



Agglomerative clustering: distance metric

What kind of distance metrics are possible?

- Minimum energy distance:

$$D = \frac{2}{|c_i||c_j|}\sum_{a\in c_i}\sum_{b\in c_j} d(a,b) - \frac{1}{|c_i|^2}\sum_{a\in c_i}\sum_{a'\in c_i} d(a,a') - \frac{1}{|c_j|^2}\sum_{b\in c_j}\sum_{b'\in c_j} d(b,b')$$

Minimal energy distance: example

Minimal energy distance:

$$D = \frac{2}{|c_i||c_j|}\sum_{a\in c_i}\sum_{b\in c_j} d(a,b) - \frac{1}{|c_i|^2}\sum_{a\in c_i}\sum_{a'\in c_i} d(a,a') - \frac{1}{|c_j|^2}\sum_{b\in c_j}\sum_{b'\in c_j} d(b,b')$$

- Let's define:
  - $d(a,a') = 1, \quad d(b,b') = 1, \quad d(a,b) = 3$
- Scenario $|c_i| = 3$, $|c_j| = 3$:

$$D = \frac{2}{3\cdot 3}9\cdot 2 - \frac{1}{3^2}6\cdot 1 - \frac{1}{3^2}6\cdot 1 = 4 - \frac{2}{3} - \frac{2}{3} = \frac{8}{3}$$

- Scenario $|c_i| = 3$, $|c_j| = 1$:

$$D = \frac{2}{3\cdot 1}3\cdot 2 - \frac{1}{3^2}6\cdot 1 - \frac{1}{1^2}1\cdot 1 = 4 - \frac{2}{3} - 1 = \frac{7}{3}$$

- Scenario $|c_i| = 1$, $|c_j| = 1$:

$$D = \frac{2}{1\cdot 1}1\cdot 2 - \frac{1}{1^2}1\cdot 1 - \frac{1}{1^2}1\cdot 1 = 4 - 1 - 1 = \frac{6}{3}$$

Clustering: summary

- Clustering discover underlying sub-populations in the data
- K-means:
  - Fast, iterative. Leads to a local minimum
  - K is the key parameter of the algorithm
- Hierarchical clustering:
  - Top-down k-means clustering
    - Fast, iterative
  - Bottom-up approach. Agglomerative clustering
    - Slow, iterative
    - Requires specific distance metric

Questions?