Unsupervised learning

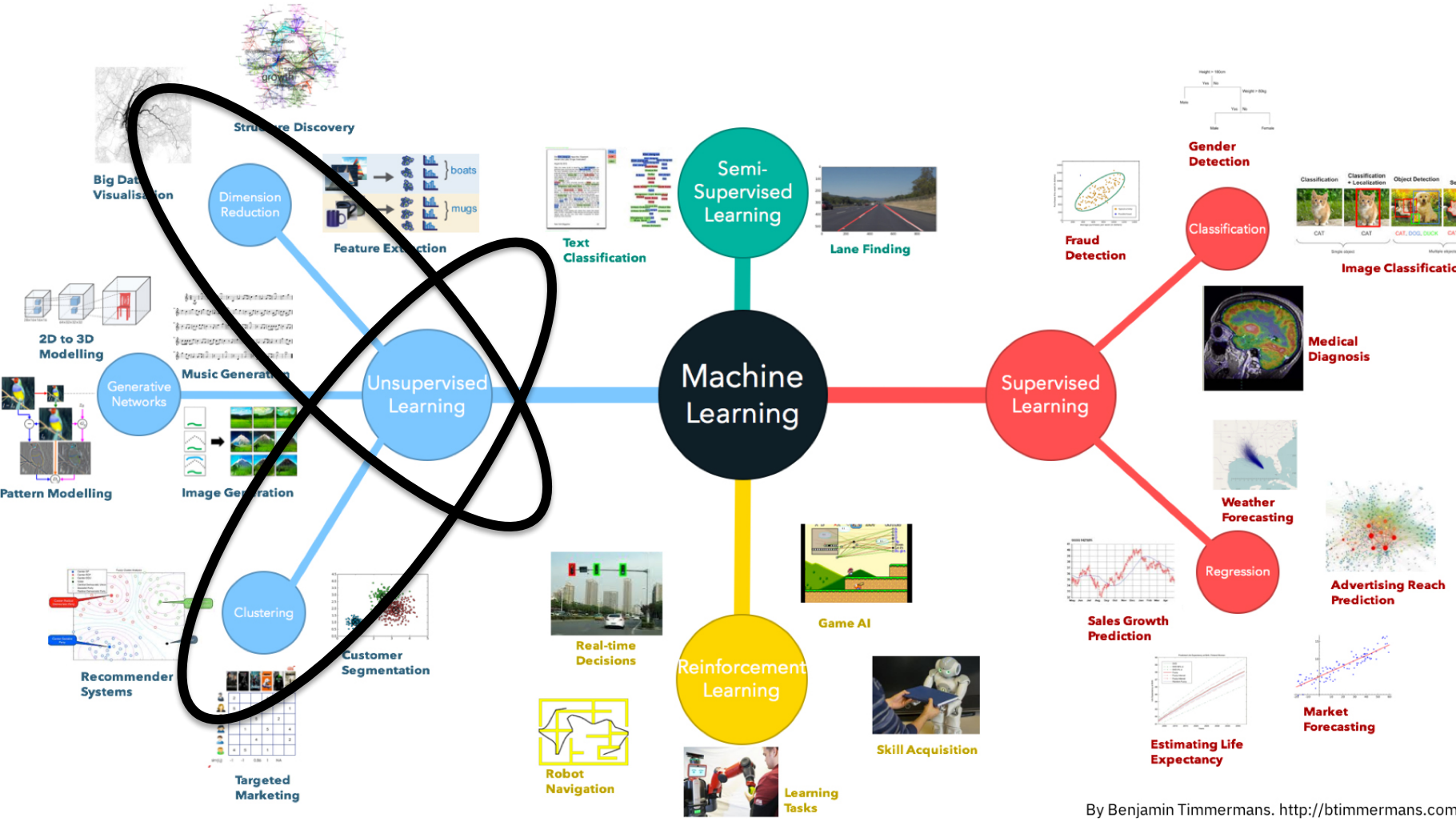
Gosia Migut

Slides credit: David Tax

Admin stuff

- Exam practice questions next week

Machine learning



By Benjamin Timmermans. http://btimmermans.com

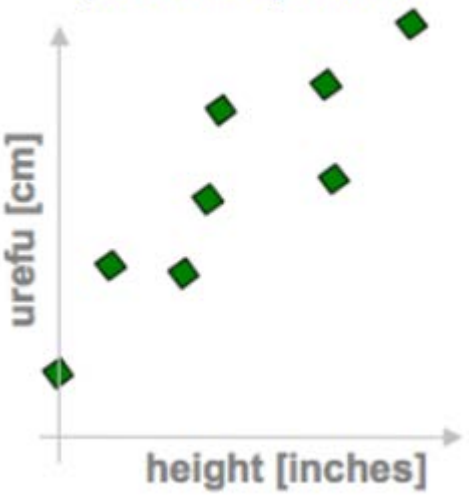
Learning goals of today

- Explain what clustering is and it’s applications
- Explain k-means algorithm
- Explain hierarchical clustering, single and complete link
- Pros and cons of k-means and hierarchical clustering
- Implement k-means

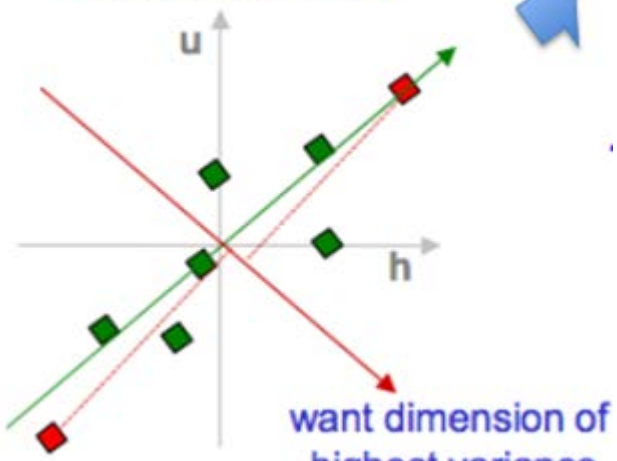
Recap PCA

1. hi-d data

("urefu" means "height" in Swahili)



2. center the points



3. compute covariance matrix

	h	u
h	2.0	0.8 → cov(h,u)
u	0.8	0.6

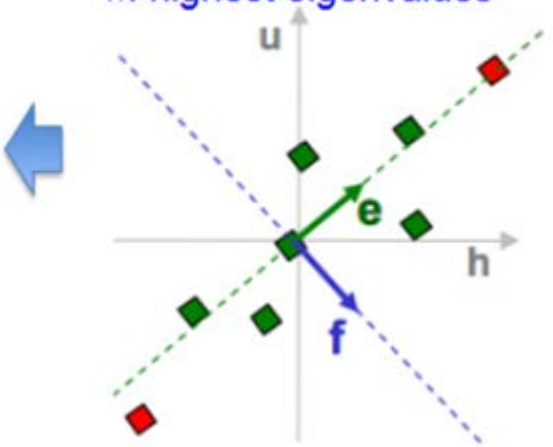
4. eigenvectors + eigenvalues

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} e_h \\ e_u \end{pmatrix} = \lambda_e \begin{pmatrix} e_h \\ e_u \end{pmatrix}$$

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} f_h \\ f_u \end{pmatrix} = \lambda_f \begin{pmatrix} f_h \\ f_u \end{pmatrix}$$

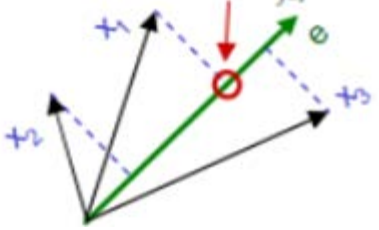
```
eig(cov(data))
```

5. pick m<d eigenvectors w. highest eigenvalues

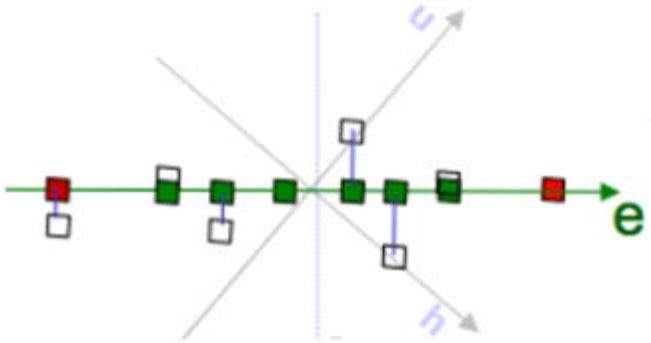


6. project data points to those eigenvectors

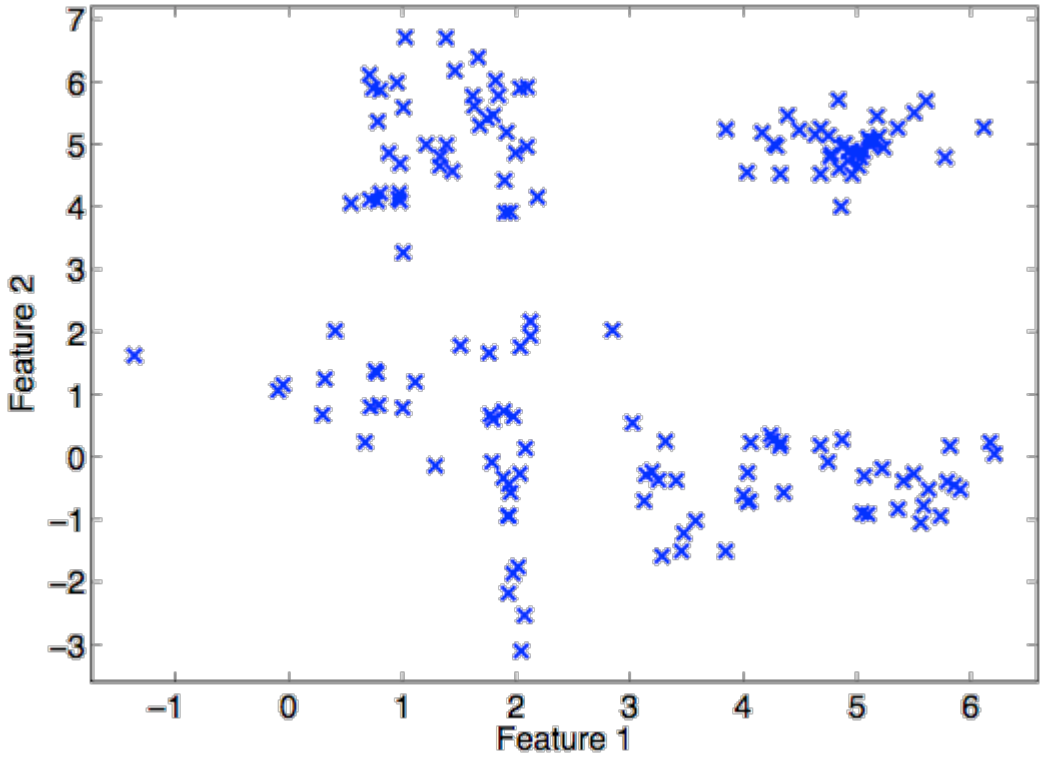
$$x_e' = x^T e = \sum_{j=1}^{d} x_{ij} e_j$$



7. low-d data



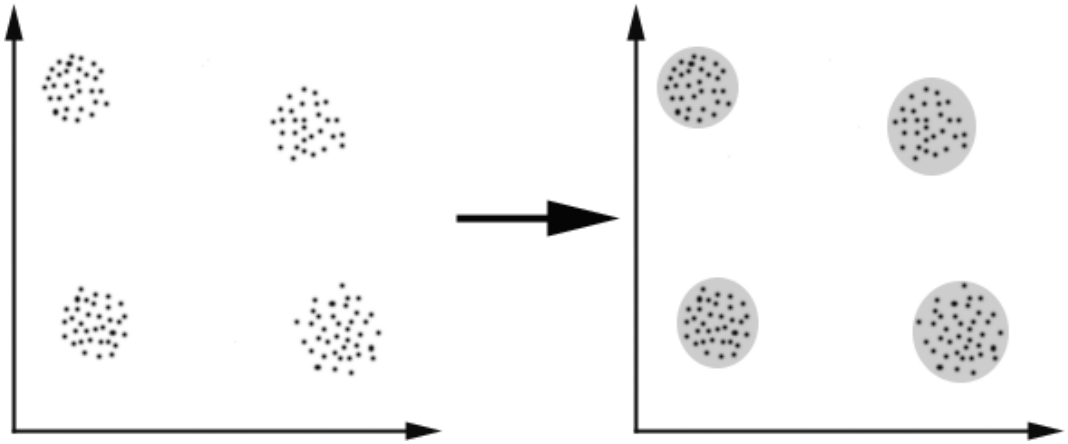
Unlabelled data: what now?



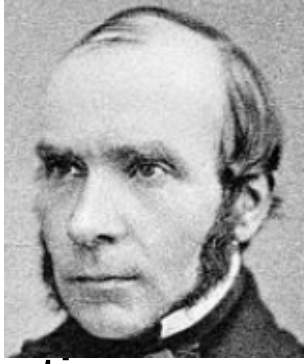
- Unsupervised learning: no labels/targets present

Clustering

- Finding natural groups in data where
  - Items within the group are close together
  - Items between groups are far apart



Historic application of clustering



- John Snow, a London physician plotted the locations of cholera deaths on a map during an outbreak in 1850s.

- The locations indicated that cases were clustered around certain intersections where there were polluted wells – exposing both the problem and the solution.

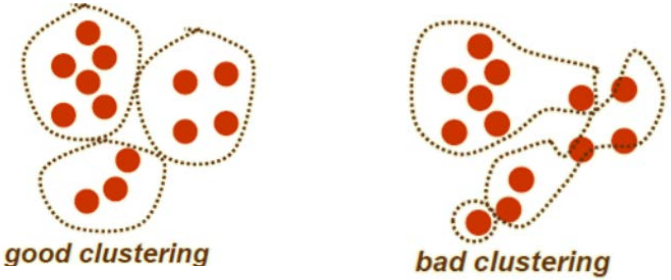
Clustering applications

- Market research: find groups of similar customers
- Social networks: find communities with similar interests / characteristics
- Recommender systems: find groups of users with similar ratings

What do we need for clustering?

1. Proximity measure, either
   - Similarity measure $s(x_i, x_k)$: large if $x_i \; and \; x_k$ are similar, or
   - Dissimilarity (distance) measure d(x_i, x_k): small if $x_i \; and \; x_k$ are similar

2. Criterion function to evaluate a clustering



3. Algorithm to compute clustering
   - Eg. By optimizing the criterion function

Distance measure

- Typically, we need to define a distance between objects first.
- Euclidean:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{l} (x_i - y_i)^2}$$

- Manhattan

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{l} |x_i - y_i|$$

- Minkowski (l_p-norm)

$$d_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{l} |x_i - y_i|^p \right)^{1/p}$$

More similarity measures

- Cosine similarity

$$s_{cos}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

- Pearson's correlation coefficient

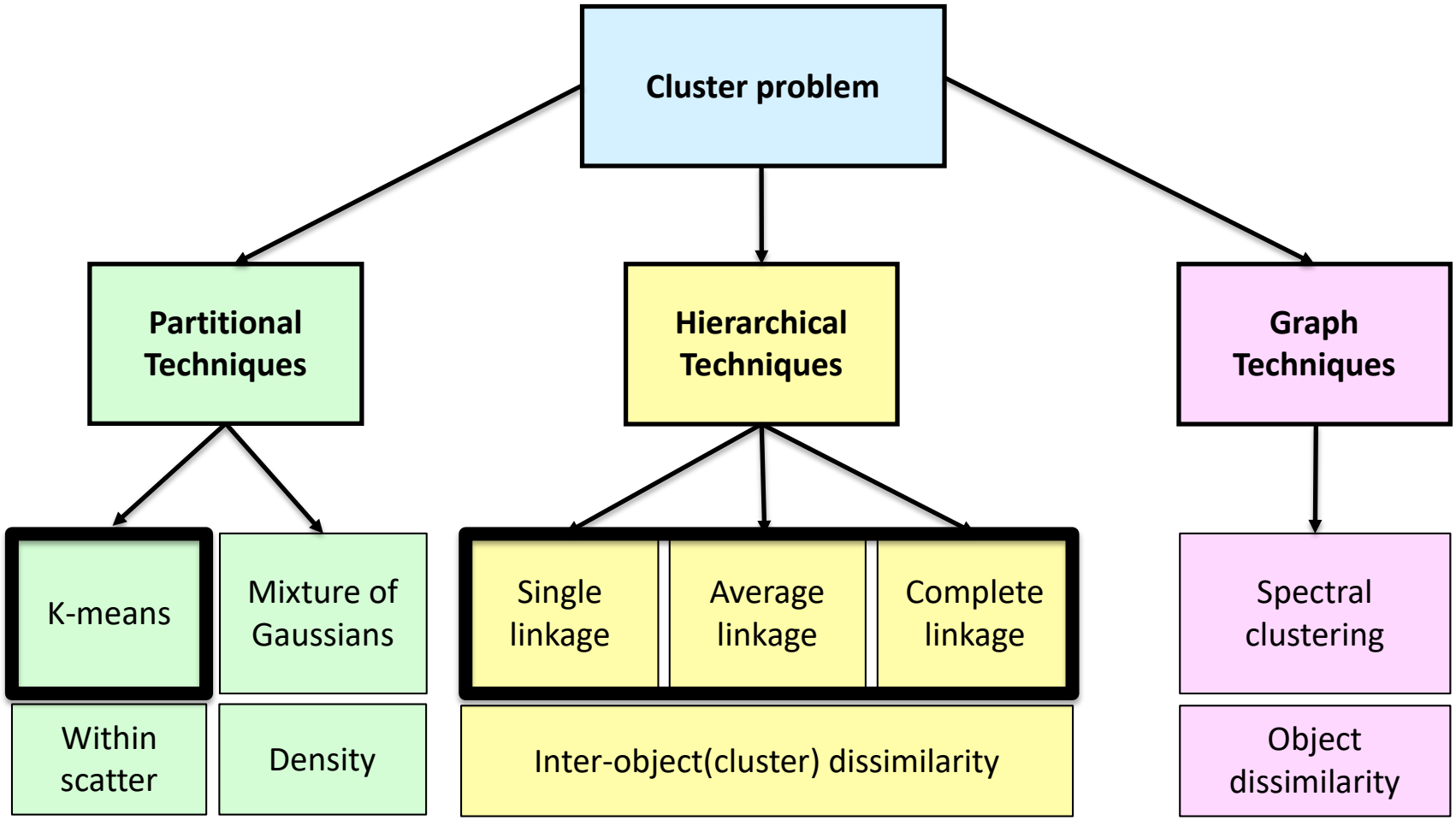
$$r_{Pearson}(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \mu_x)^T (\mathbf{y} - \mu_y)}{\|\mathbf{x} - \mu_x\| \|\mathbf{y} - \mu_y\|}$$

- and more... (for discrete features, mixed features, categorical features, ...)

Cluster evaluation (a hard problem)

- Intra-cluster cohesion (compactness):
  - Cohesion measures how near the data points in a cluster are to the cluster's mean.
  - Sum of squared errors (SSE) is a commonly used measure.
- Inter-cluster separation (isolation):
  - Separation means that different cluster means should be far away from one another.
- In most applications, expert judgments are still the key

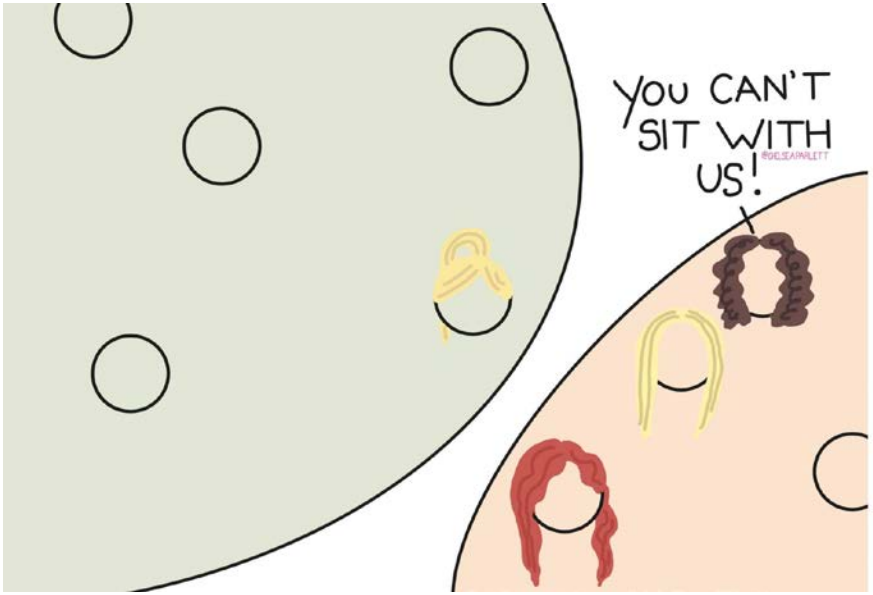
Clustering techniques



Hard vs. soft

- Hard assignments: each point assigned to 1 cluster
  - K-Means
  - Hierarchical clustering
- Soft assignments: each point assigned cluster membership
  - Fuzzy C-means
  - Probabilistic mixture models

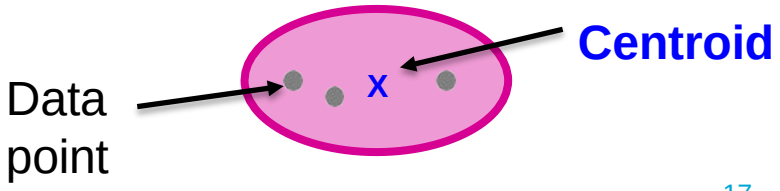
K-means clustering



K-mean (girls)

K-means algortihm

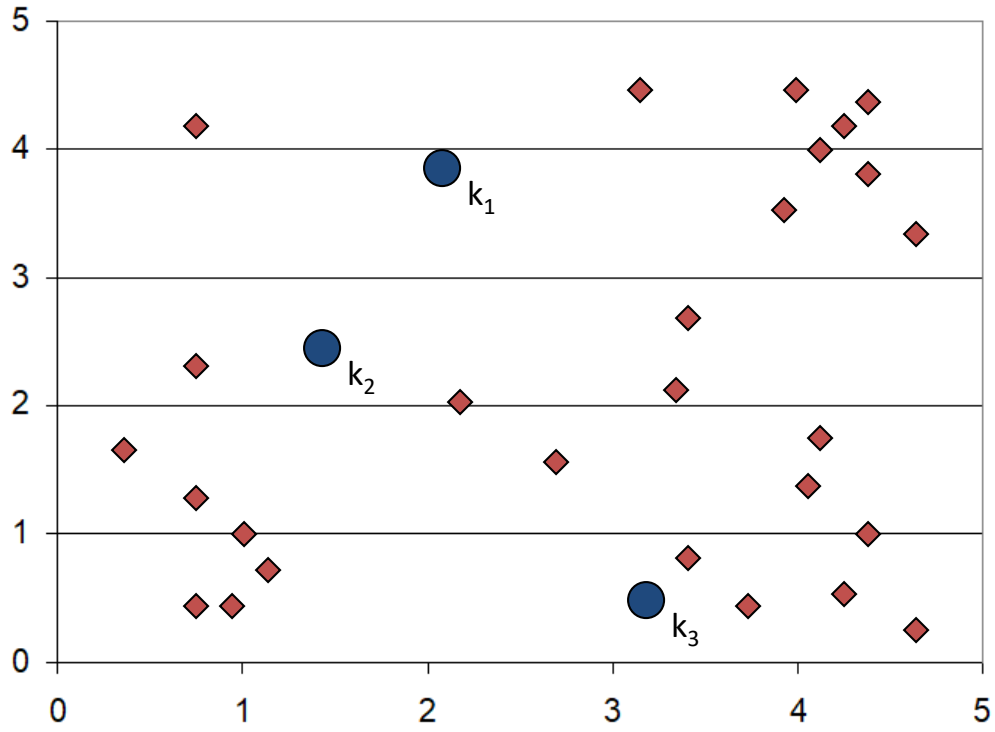
- K-means (MacQueen, 1967) is a partitional clustering algorithm
- Let the set of n data points be $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}\}$ where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip}\}$ is a feature vector and p the number of dimensions.
- The k-means algorithm partitions the given data into k clusters:
  - Each cluster has a cluster centre (cluster mean), called centroid.
  - K is specified by the user



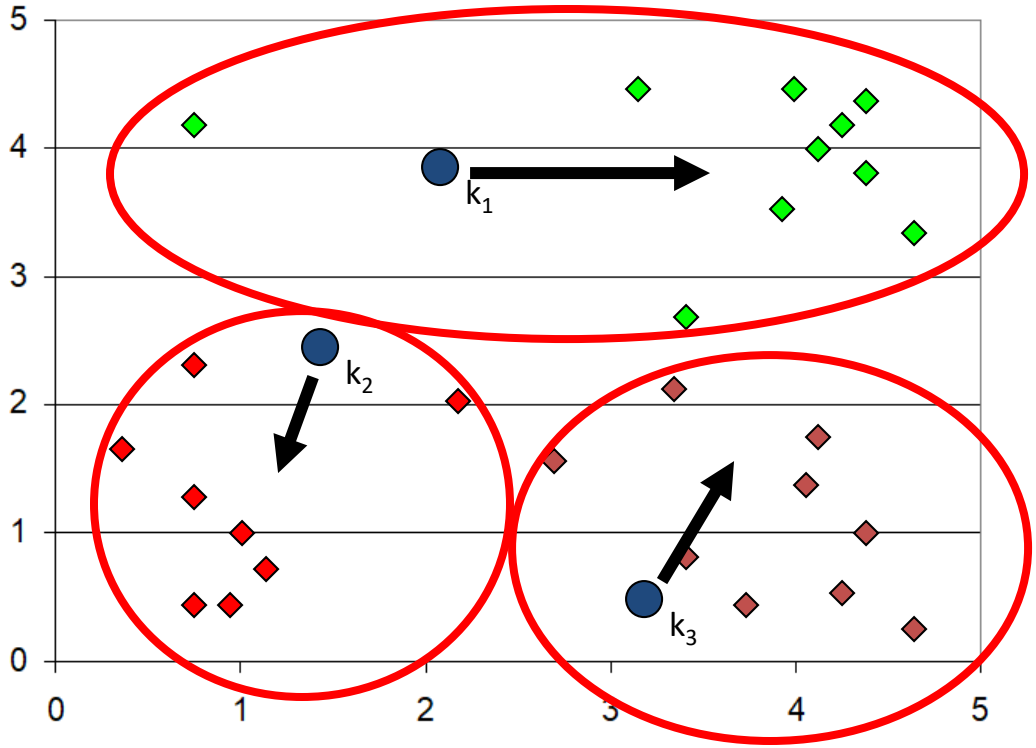
K-means algorithm

- Given k, the k-means algorithm works as follows:
  1. Choose k (random) data points (seeds) to be the initial centroids, cluster centers
  2. Assign each data point to the closest centroid
  3. Re-compute the centroids using the current cluster memberships
  4. If a convergence criterion is not met, repeat steps 2 and 3

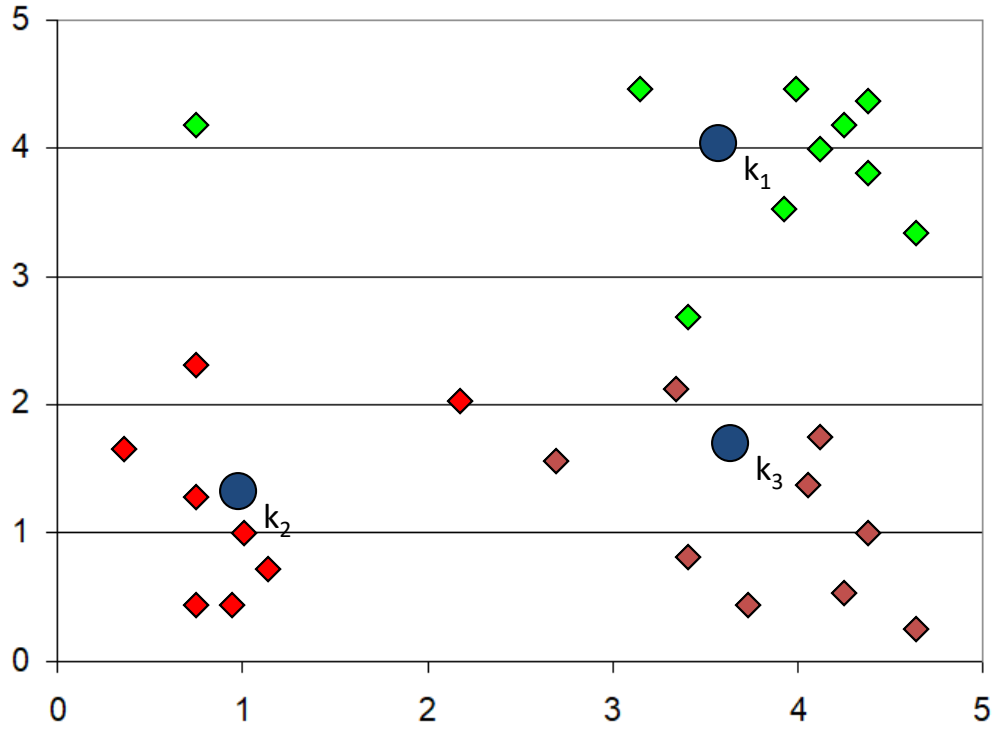
K-means: how it works



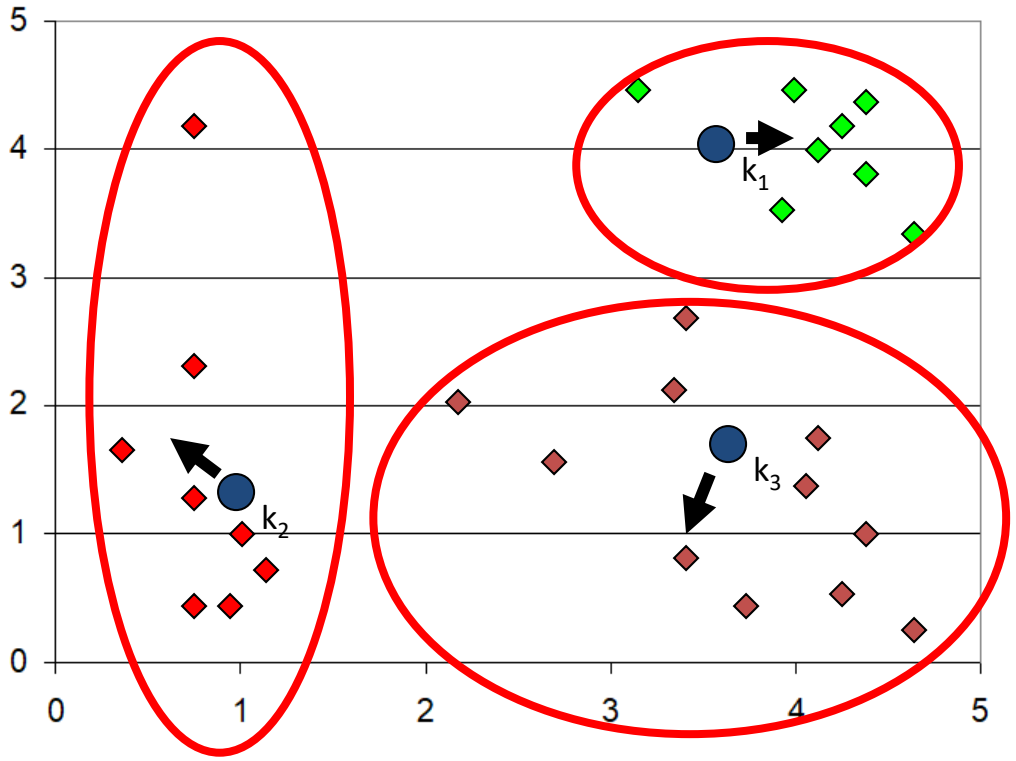
K-means: how it works



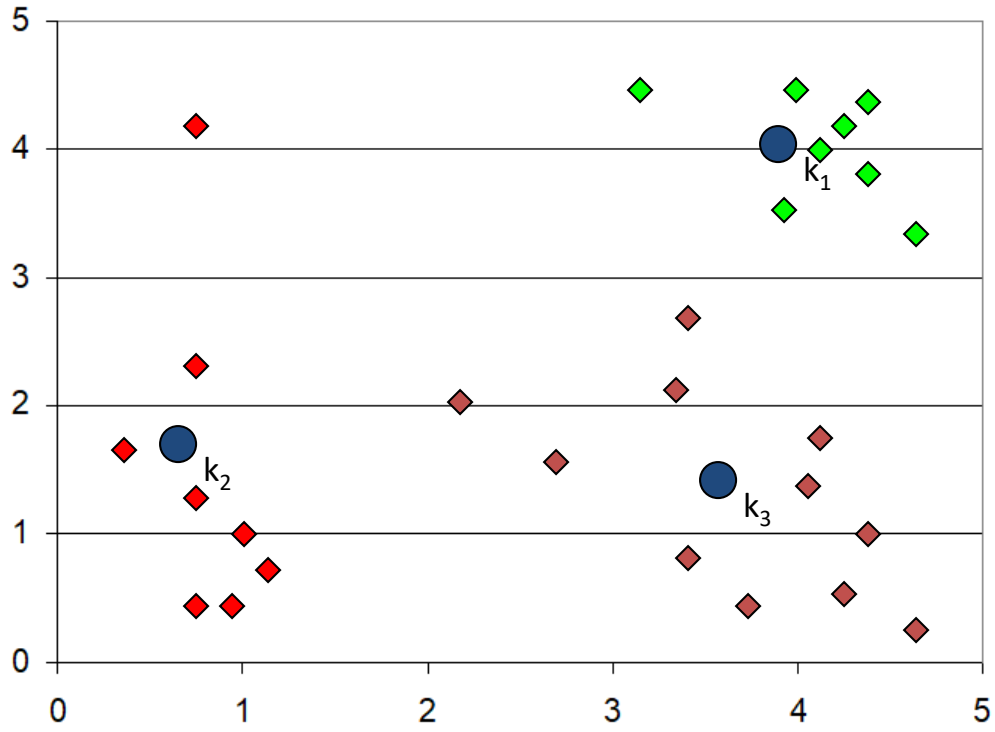
K-means: how it works



K-means: how it works



K-means: how it works



K-means questions

- How do we choose the number of centers?
- What is it trying to optimize?
- Are we sure it will terminate?
- Are we sure it will find an optimal clustering?

K-means convergence (stopping) criterion

- no (or minimum) re-assignments of data points to different clusters, or
- no (or minimum) change of centroids, or
- minimum decrease in the sum of squared errors (SSE)

Sum of squared errors

- Cost function (distortion)

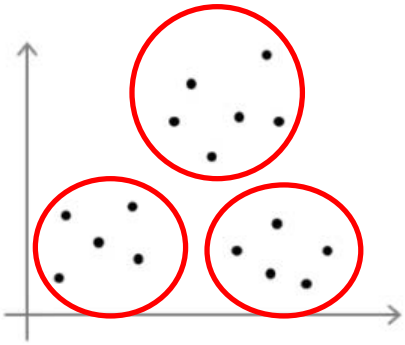
$$J(c, \mu) = \frac{1}{n} \sum_{i=1}^{m} ||x_i - \mu_{C_i}||^2$$

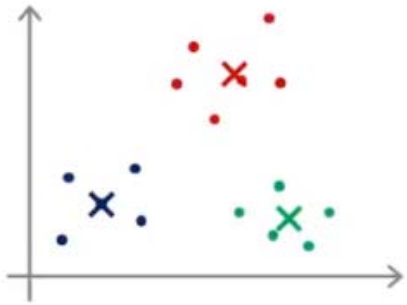
- μ_{C_i} is cluster center to which x_i is assigned

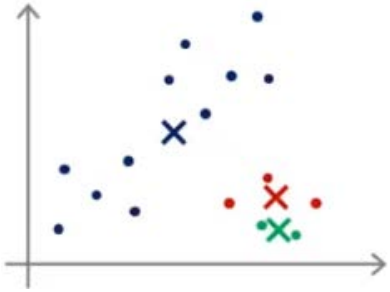
Random initialization

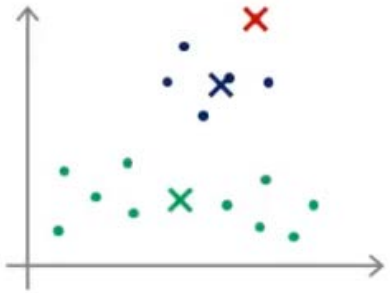
- $2 \leq k < m$
- Random pick K training examples
- Set $\mu_1, \mu_2, \ldots, \mu_k$ equal to these examples

Local optima







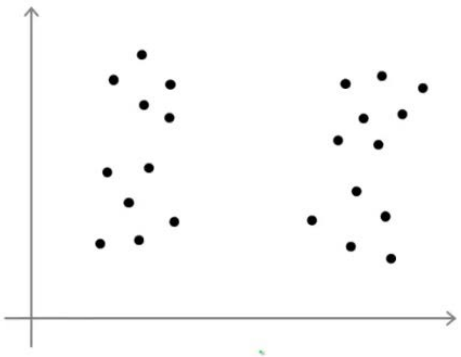


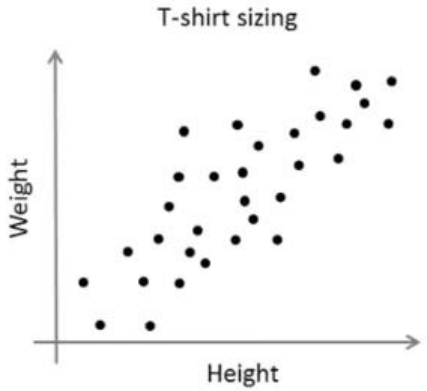
Random initialization

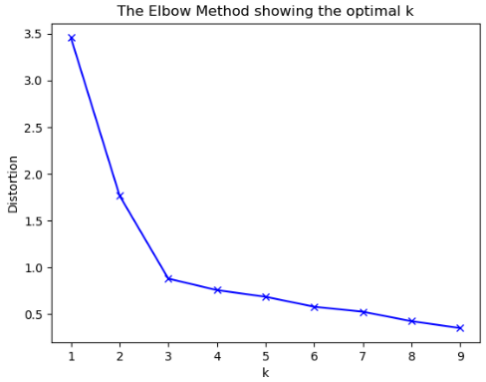
- For i=1 to 10000
  {
    Randomly initialize k means
    Run k-means. Get centroids and means
    Compute cost function J
  }
- Pick clustering that gave lowest cost
- For high-dimensional data, many restarts are necessary (e.g. I = 10000)!

Choosing the number of clusters

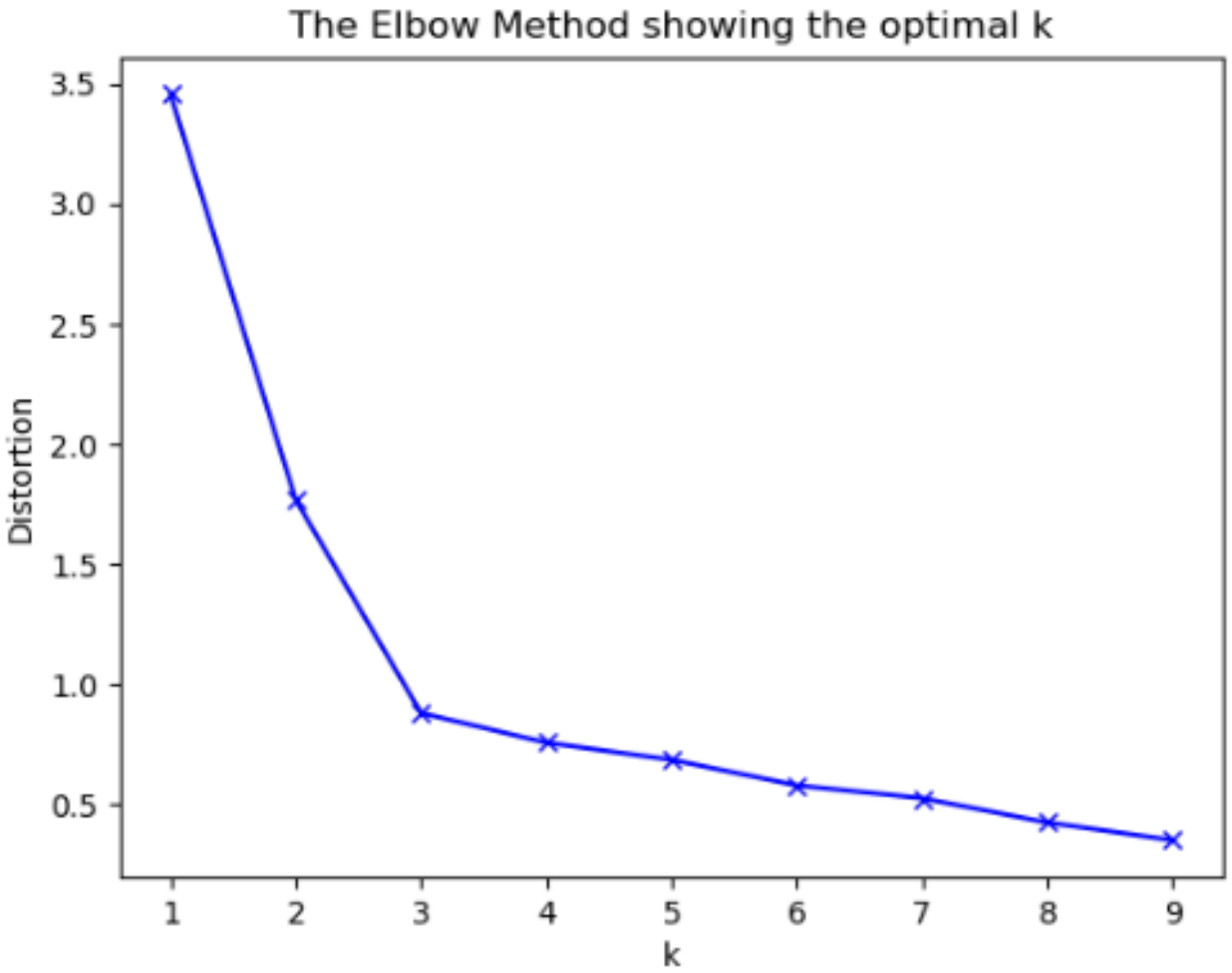
- Inspect visually
- Known purpose
- Elbow method





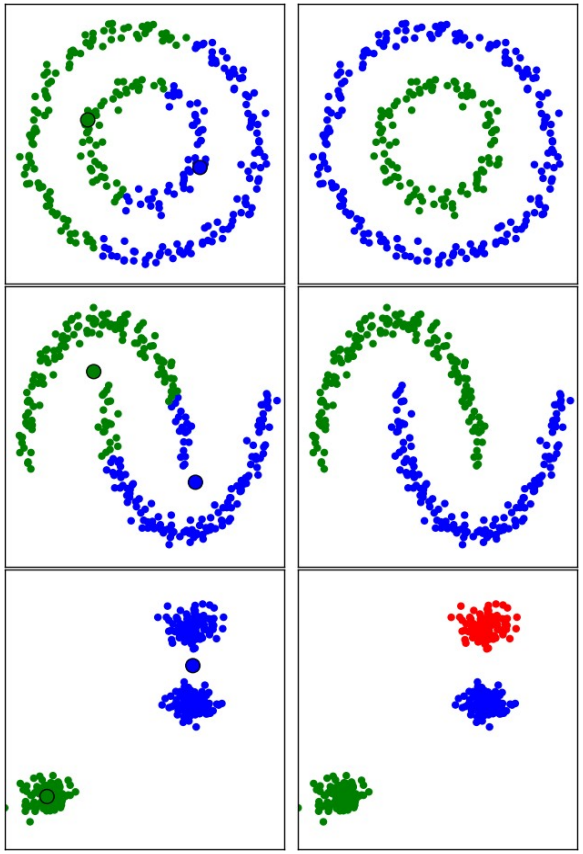


Elbow method



K-means summary

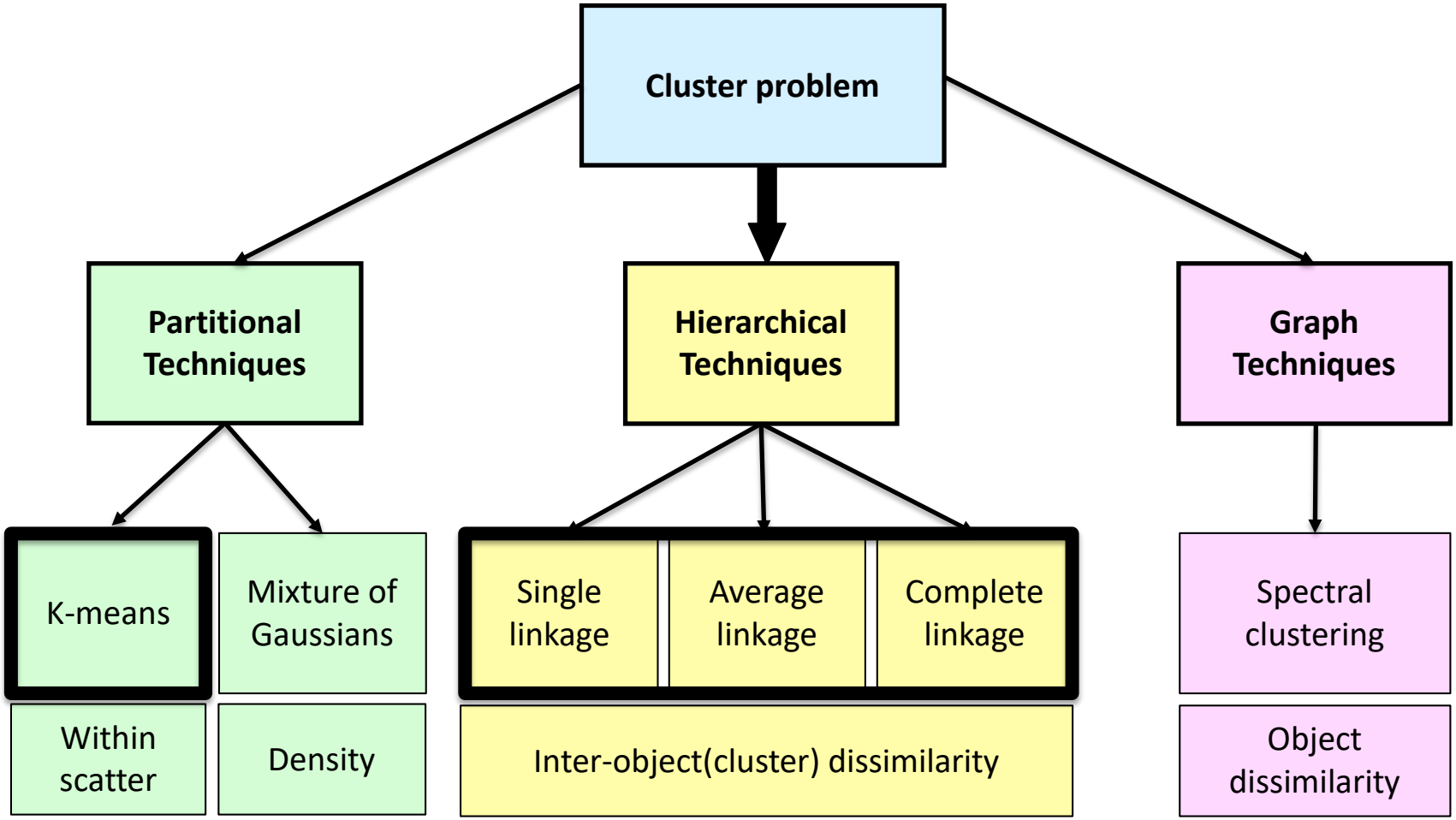
- Disadvantages:
  - Finds only convex clusters (“round shapes”)
  - Sensitive to initialization
  - Can get stuck in local minima
- Advantages:
  - Very simple
  - Fast



Example exercise

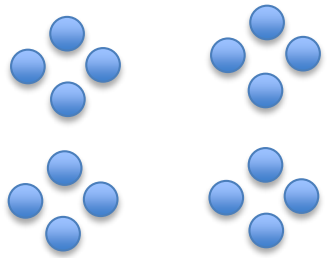
- We have the following points (1, 4), (2, 2), (5, 5) and (4, 6).
- We also have two cluster centroids $\mu_1 = (1, 2)$ and $\mu_2 = (6, 6)$.
- What is the value of the k-means cost function (SSE)?

Clustering techniques



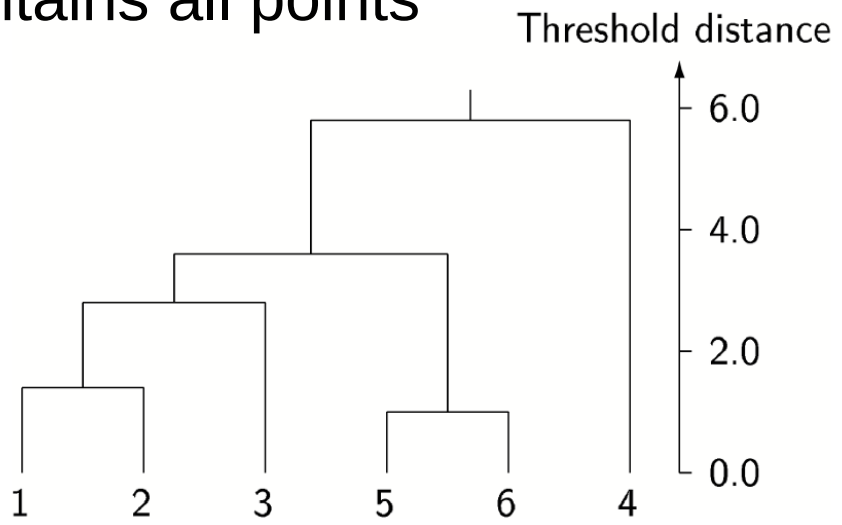
Hierarchical clustering

Hierarchical clustering



- Selecting k is a problem of granularity
  - How course or fine-grained is the structure in the data?
  - No cluster algorithm able to pick k

- Instead of picking k find a hierarchy of structure
  - Course effects: top level contains all points
  - Fine-grained: bottom level
    one cluster per data point



Hierarchical clustering approaches

- Agglomerative (bottom-up):
  - each point starts as cluster
  - group two closest clusters
  - stop at some point

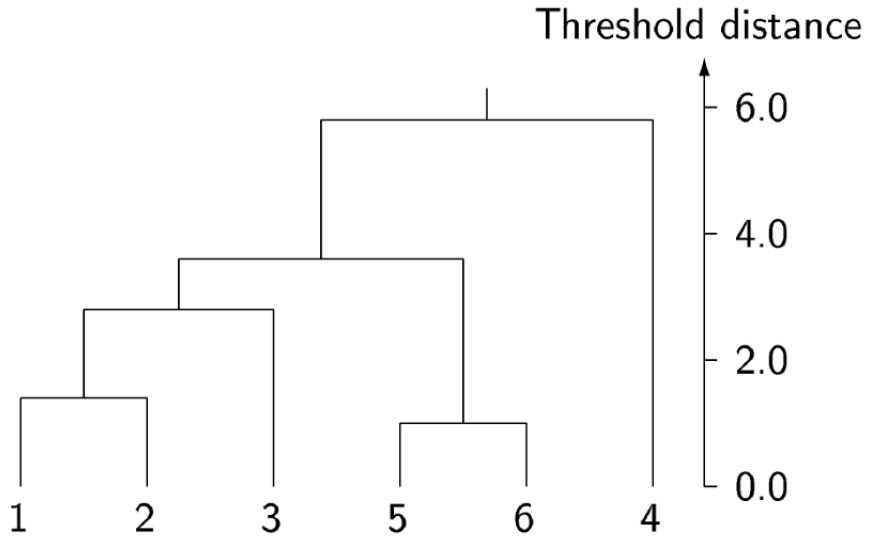


Figure 11.1 Dendrogram.

Hierarchical clustering approaches

- Divisive (top-down):
  - all points start in one cluster
  - split cluster in some sensible way
  - stop at some point

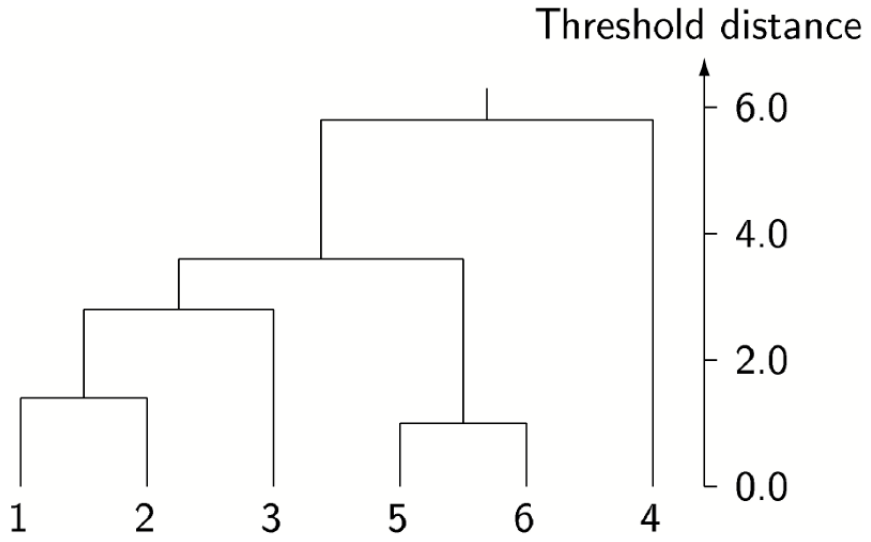
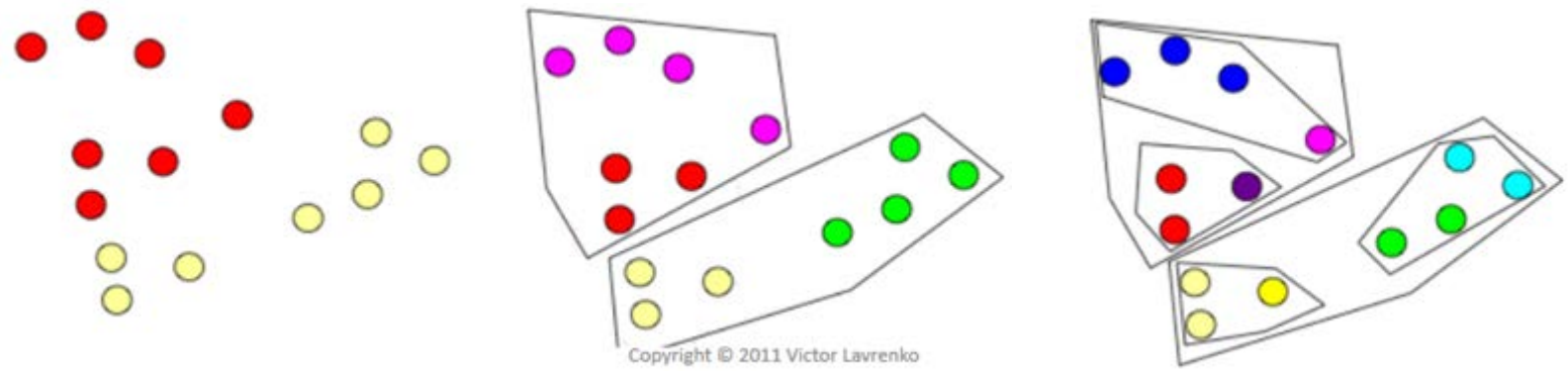


Figure 11.1 Dendrogram.

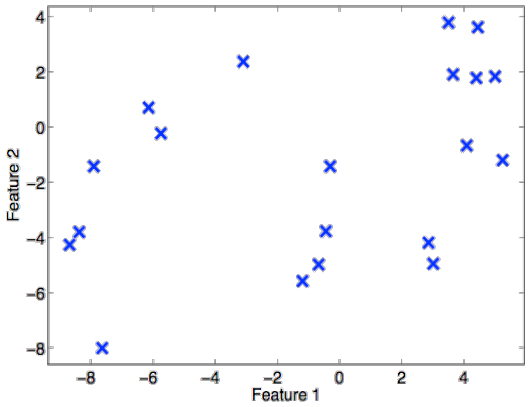
Divisive: hierarchical k-means

- Apply k-means recursively:
  - Run k-mean on the original data for k=2
  - For each of the resulting clusters run k-means with k=2

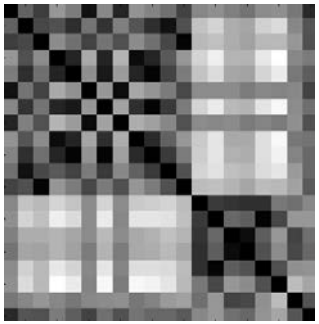


Aglomerative clustering

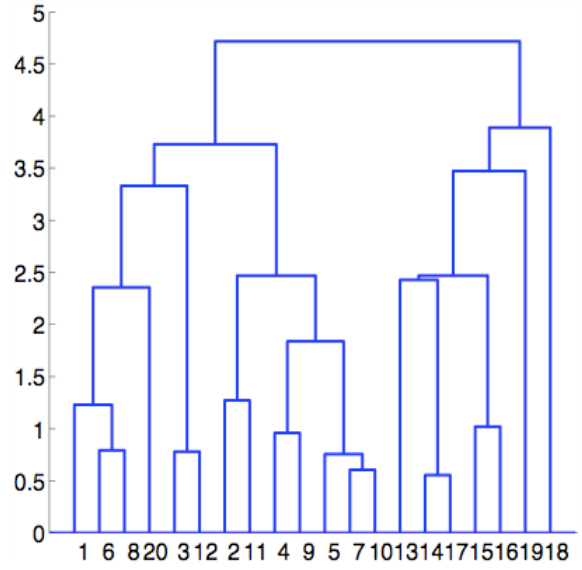
- Starting from individual observations, produce sequence of clusterings of increasing size
- At each level, two clusters chosen by criterion are merged



2D scatter plot of data



dissimilarity matrix



dendrogram

Aglomerative clustering

1. Determine distances between all clusters
2. Merge clusters that are closest
3. IF #clusters>1 THEN GOTO 1

- Which clusters to start with?
- What is the distance between clusters?
- Final number of clusters?

Different merging rules

- **Single linkage:** two nearest objects in the clusters :

$$g(R,S) = min_{ij}\{d(x_i, x_j): x_i \in R, x_j \in S\}$$

- **Complete linkage**: two most remote objects in the clusters :

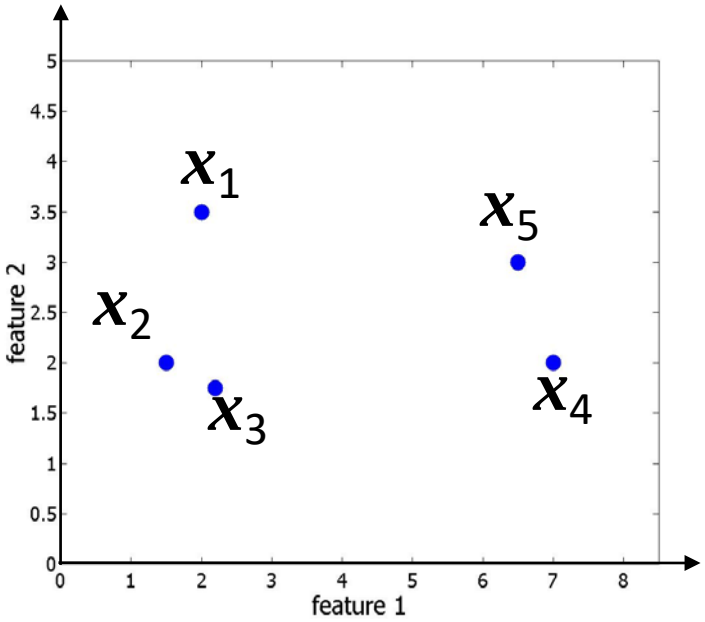
$$g(R,S) = max_{ij}\{d(x_i, x_j): x_i \in R, x_j \in S\}$$

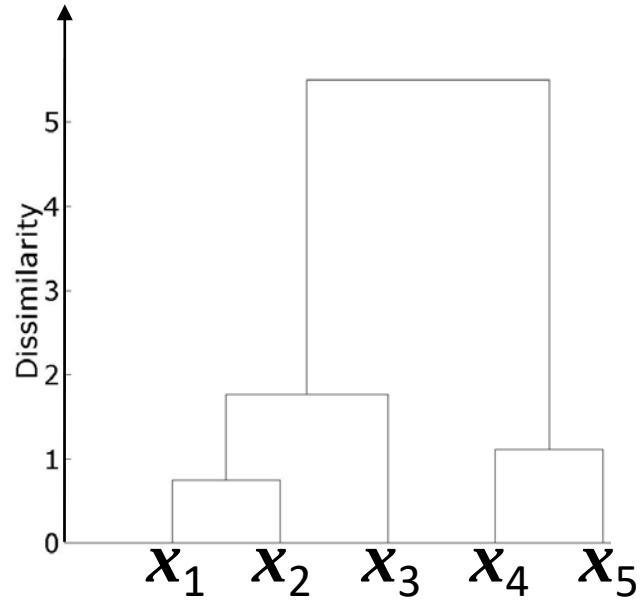
- **Average linkage**: cluster centres :

$$g(R,S) = \frac{1}{|R||S|}\sum_{ij}\{d(x_i, x_j): x_i \in R, x_j \in S\}$$

Hierarchical clustering: how it works

- Input:
  - dataset, $\boldsymbol{X}$: [n x p], or directly:
  - dissimilarity matrix, $\boldsymbol{D}$: [n x n]
  - linkage type
- Output:
  - dendrogram

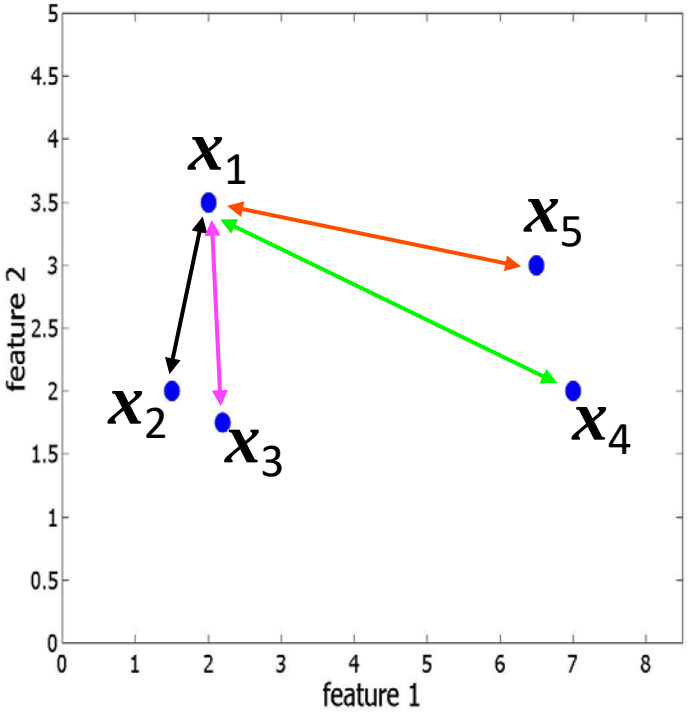




Hierarchical clustering

- **Step 0:** all objects are a cluster:

Dataset

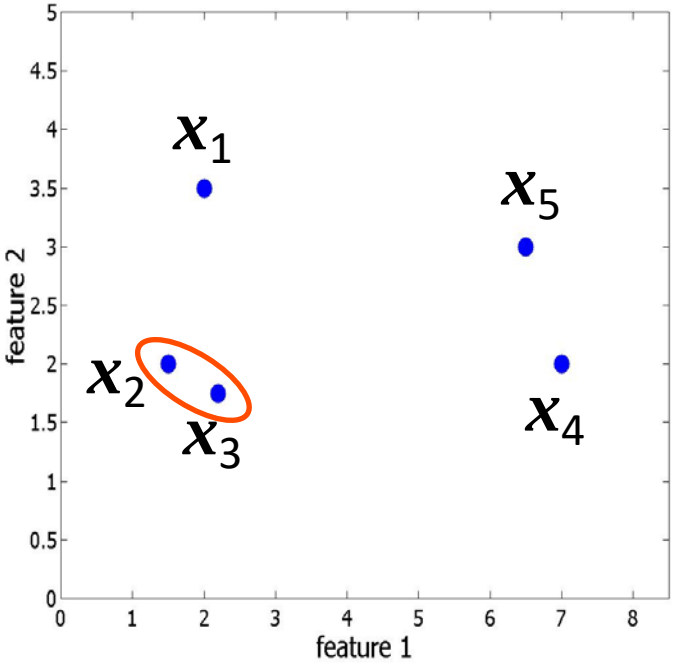


(Euclidean) distance matrix, $\boldsymbol{D}$

	x_1	x_2	x_3	x_4	x_5
x_1	0.00	1.58	1.76	5.22	4.53
x_2		0.00	0.74	5.50	5.10
x_3			0.00	4.81	4.48
x_4				0.00	1.12
x_5					0.00

Hierarchical clustering

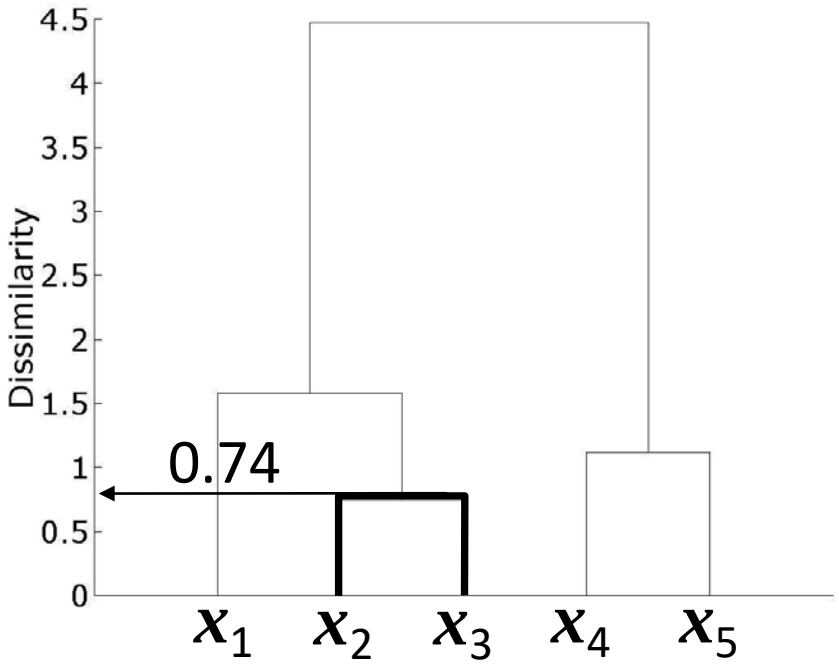
- **Step 1:**
  Find the most similar pair: $\min_{(i,j)}\{d(i,j)\} = d(2,3)$



	x_1	x_2	x_3	x_4	x_5
x_1	0.00	1.58	1.76	5.22	4.53
x_2		0.00	0.74	5.50	5.10
x_3			0.00	4.81	4.48
x_4				0.00	1.12
x_5					0.00

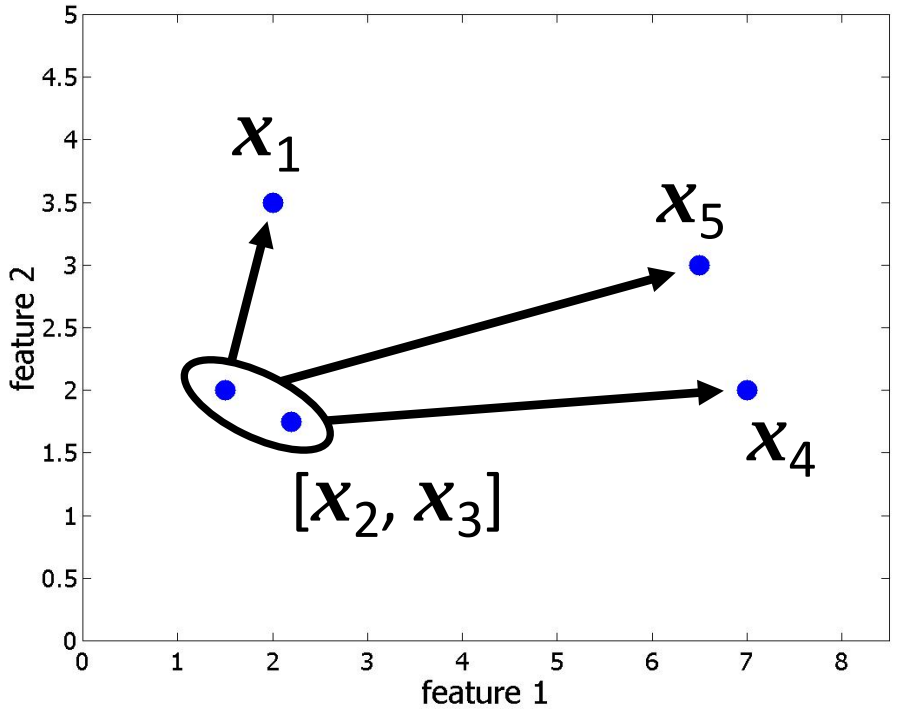
Hierarchical clustering

- **Step 2:**
  Merge x_2 and x_3 into a single object, $[x_2, x_3]$;



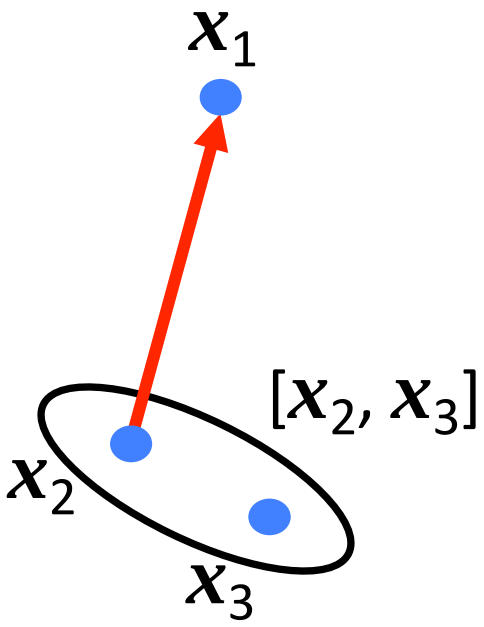
Hierarchical clustering

- **Step 3:**
  Recompute $\boldsymbol{D}$ – what is the distance between $[\boldsymbol{x}_2, \boldsymbol{x}_3]$ and the rest?



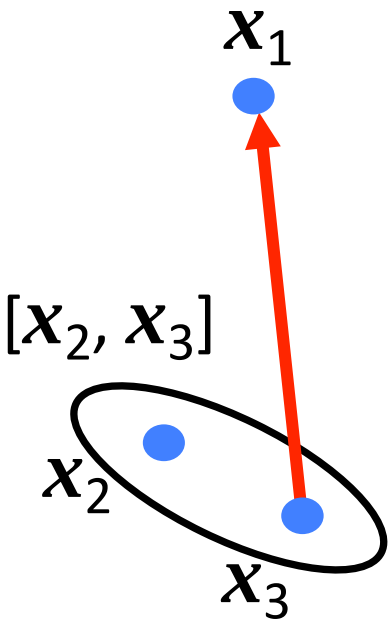
Hierarchical clustering

- **Step 3:**
  Recompute $\boldsymbol{D}$ – **single linkage:** $d([\boldsymbol{x}_2,\boldsymbol{x}_3],\boldsymbol{x}_1) = \min(d(\boldsymbol{x}_1,\boldsymbol{x}_2),d(\boldsymbol{x}_1,\boldsymbol{x}_3))$



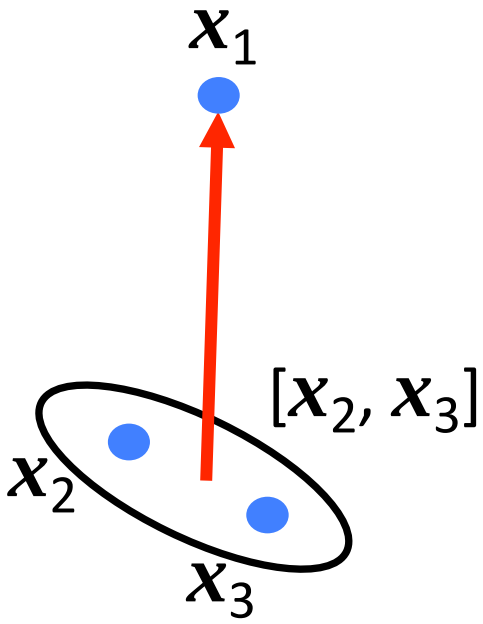
Hierarchical clustering

- **Step 3:**
  Recompute $\boldsymbol{D}$ – **complete linkage:** $d([\boldsymbol{x}_2,\boldsymbol{x}_3],\boldsymbol{x}_1) = \max(d(\boldsymbol{x}_1,\boldsymbol{x}_2),d(\boldsymbol{x}_1,\boldsymbol{x}_3))$



Hierarchical clustering

- **Step 3:**
  Recompute $\boldsymbol{D}$ – **average linkage:** $d([\boldsymbol{x}_2,\boldsymbol{x}_3],\boldsymbol{x}_1) = \text{mean}(d(\boldsymbol{x}_1,\boldsymbol{x}_2),d(\boldsymbol{x}_1,\boldsymbol{x}_3))$



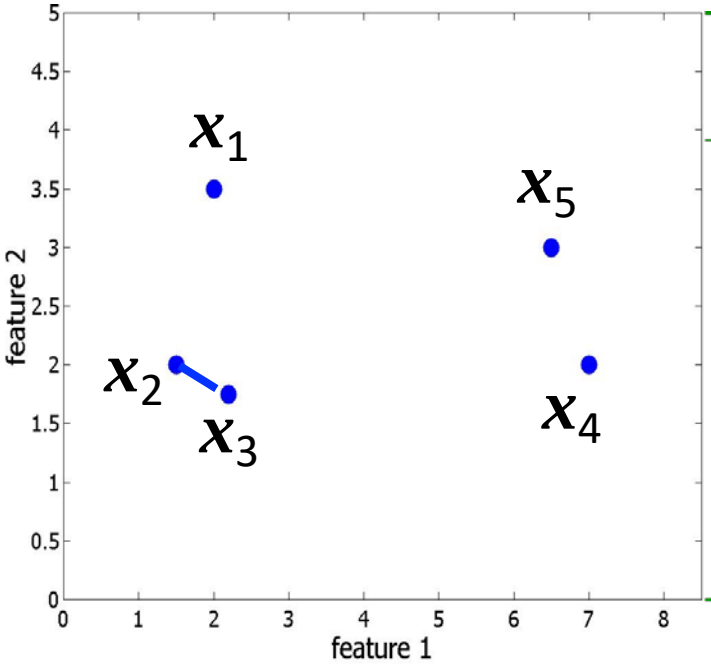
Hierarchical clustering

- **Step 3:**
  Recompute $\boldsymbol{D}$ – **single linkage:**

	x_1	$[x_2,x_3]$	x_4	x_5
x_1	0.00	1.58	5.22	4.53
$[x_2,x_3]$		0.00	4.81	4.48
x_4			0.00	1.12
x_5				0.00

Hierarchical clustering

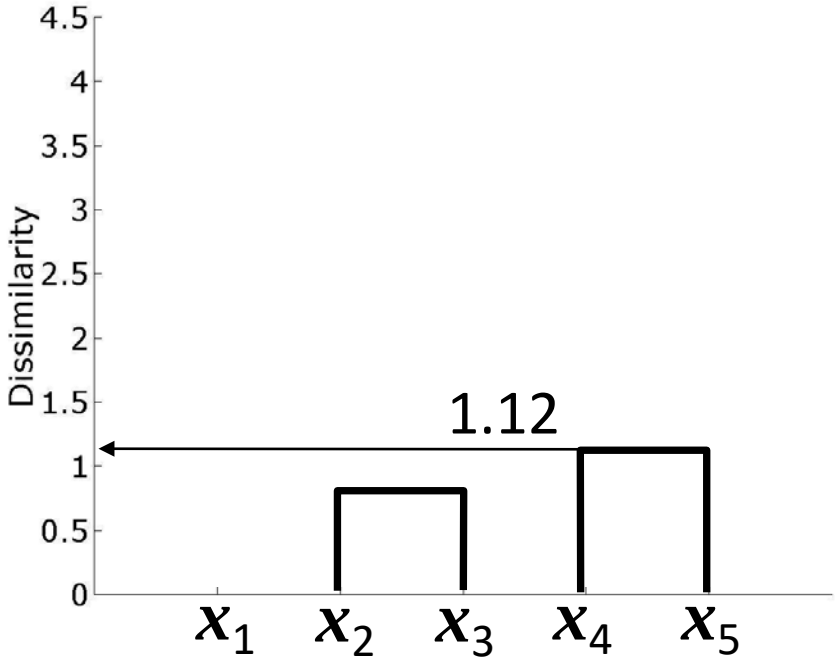
- **Repeat, step 1:**
  Find the most similar pair of objects: $\min_{(i,j)}\{d(i,j)\}$ $= d(4,5)$



	x_1	$[x_2,x_3]$	x_4	x_5
x_1	0.00	1.58	5.22	4.53
$[x_2,x_3]$	1.58	0.00	4.81	4.48
x_4	5.22	4.81	0.00	1.12
x_5	4.53	4.48	1.12	0.00

Hierarchical clustering

- **Repeat, step 2:**
  Merge x_4 and x_5 into a single object, $[x_4,x_5]$;



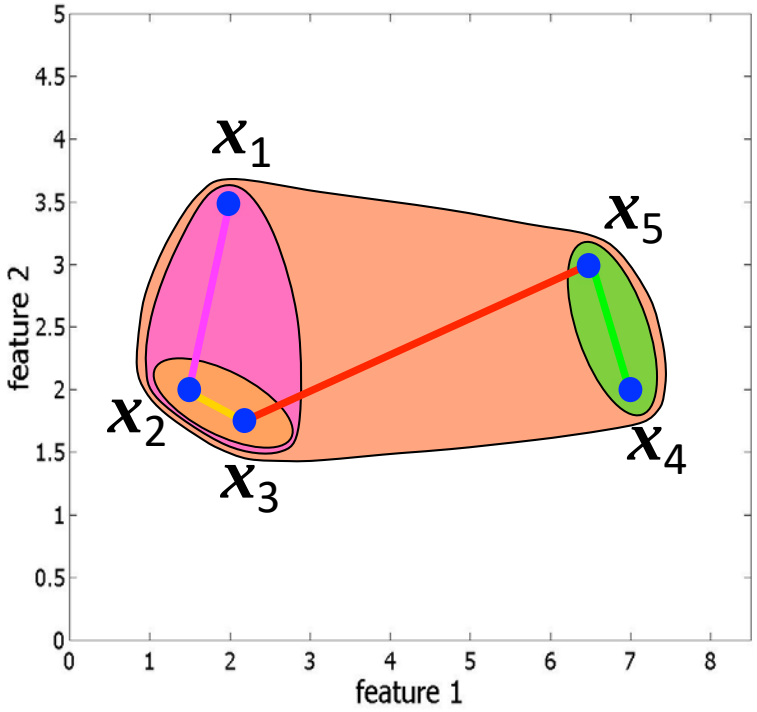
Hierarchical clustering

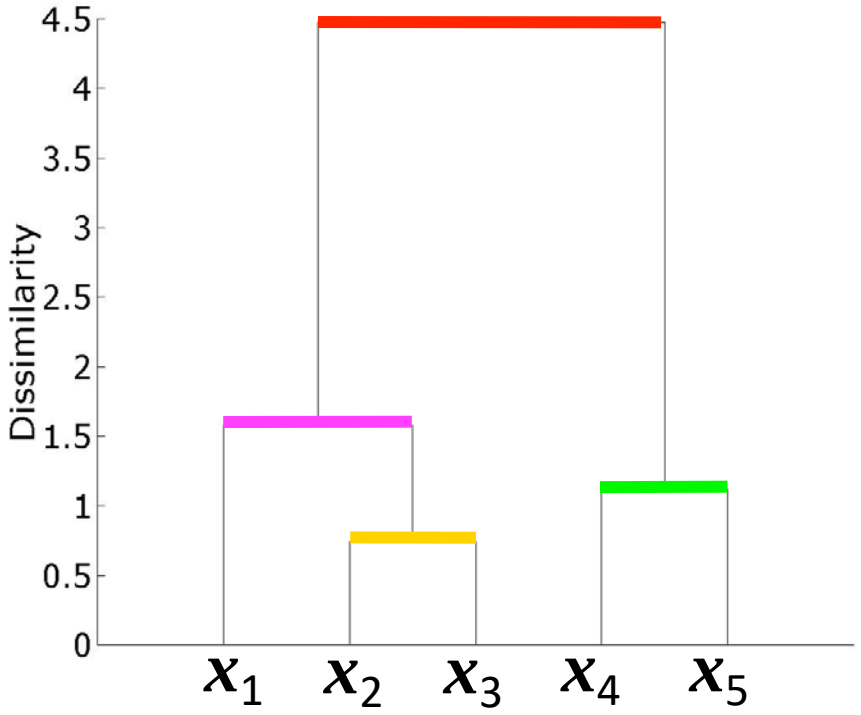
- **Repeat, step 3:**
  Recompute $\boldsymbol{D}$ (single linkage):

	x_1	$[x_2,x_3]$	$[x_4,x_5]$
x_1	0.00	1.58	4.53
$[x_2,x_3]$		0.00	4.48
$[x_4,x_5]$			0.00

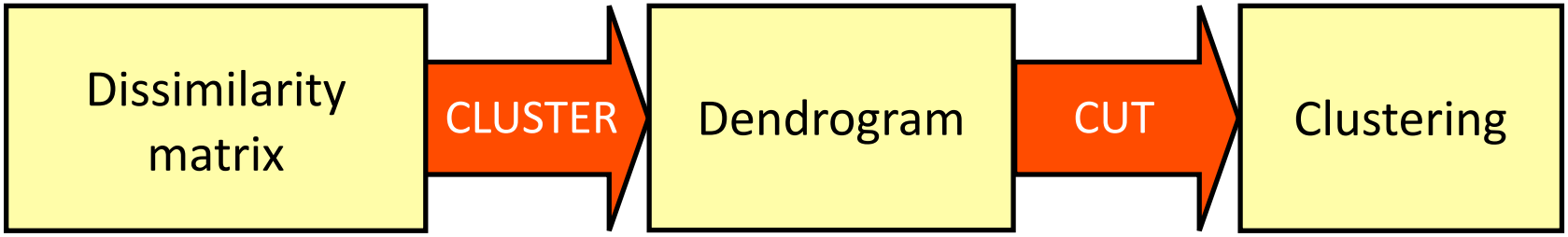
Hierarchical clustering

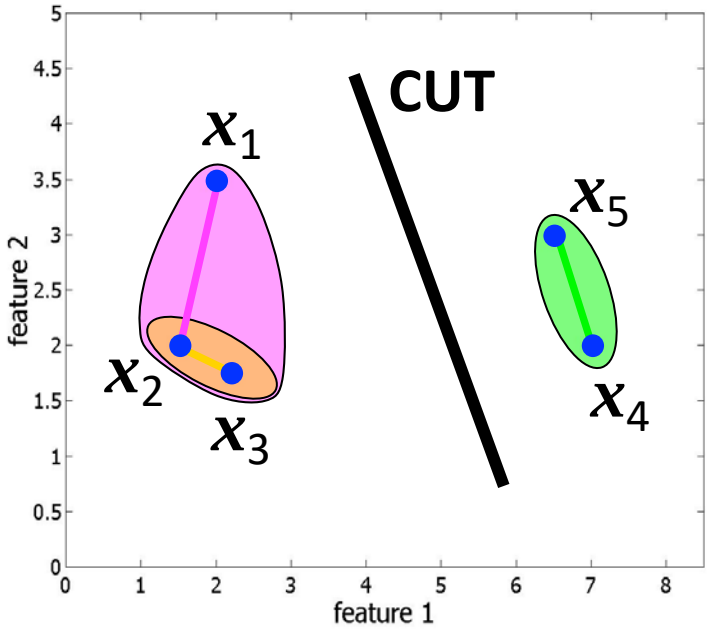
- Repeat steps 1-3 untill a single cluster remains

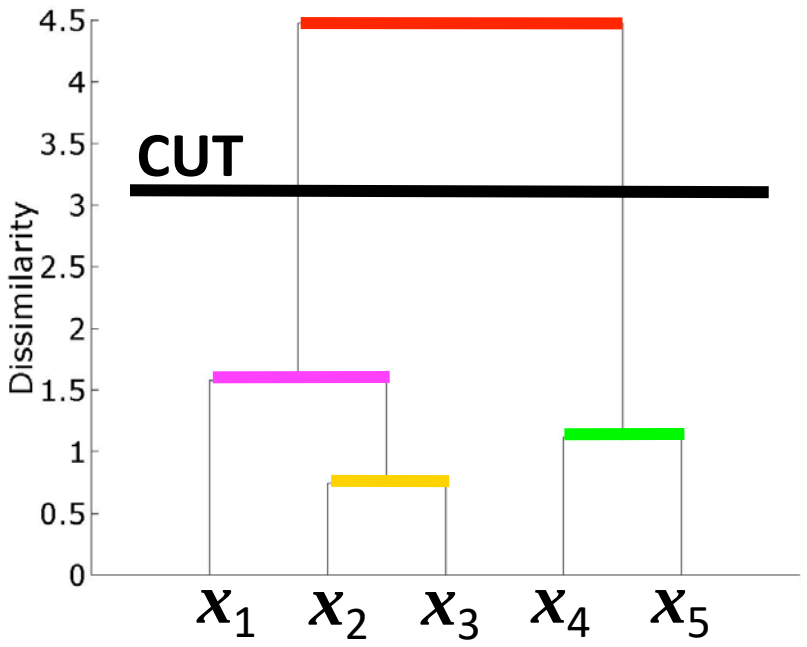




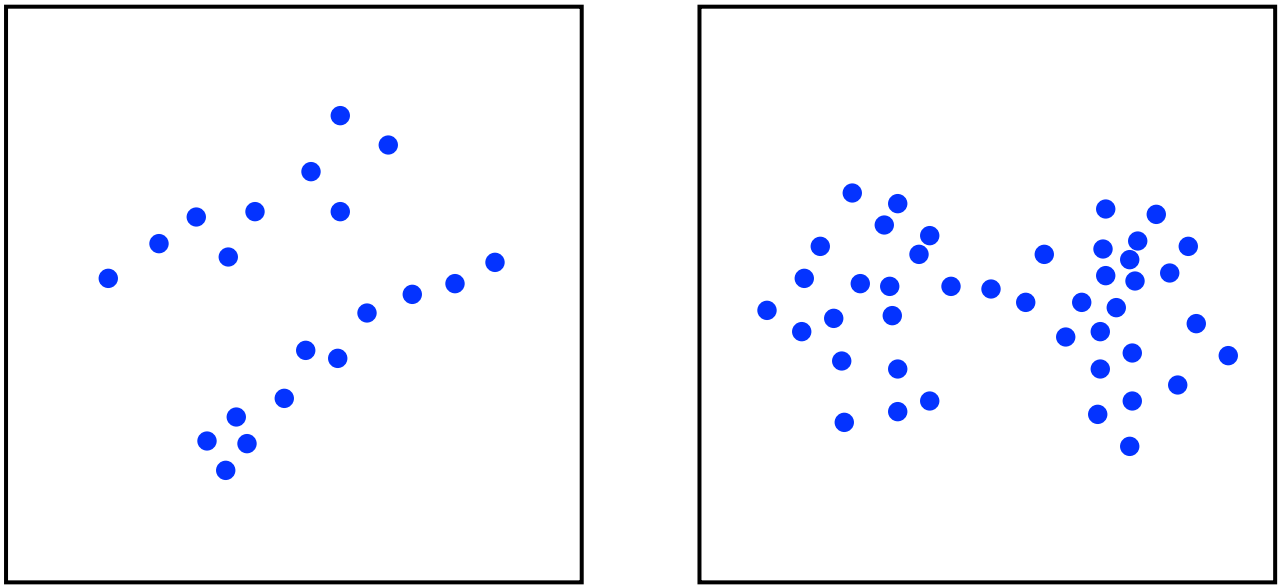
Hierarchical clustering







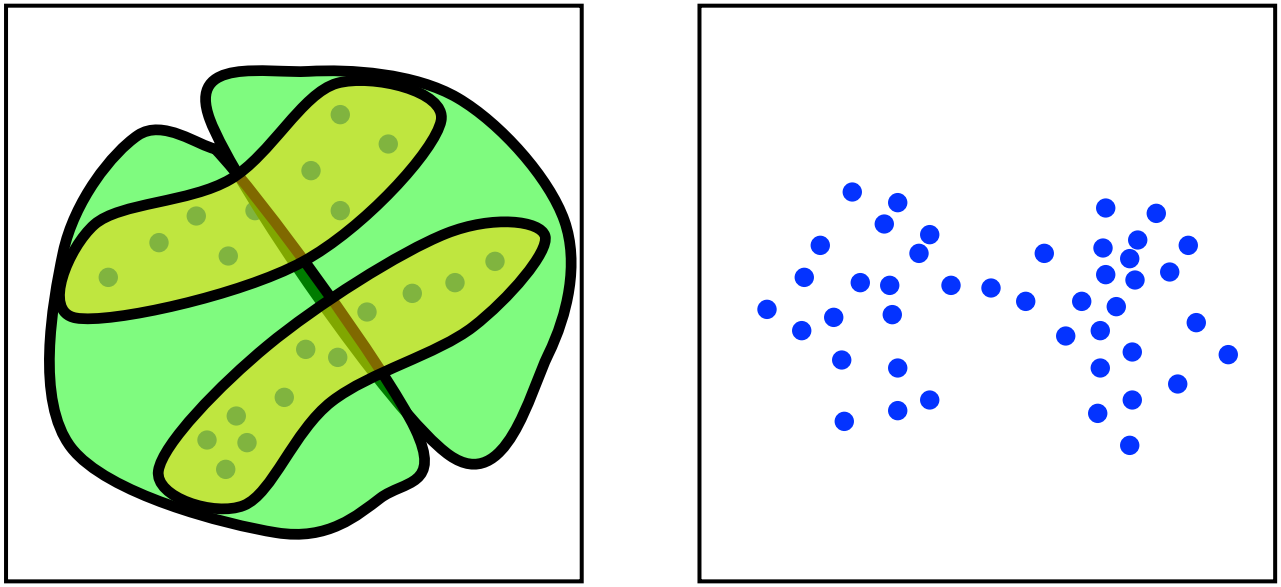
Linkage and cluster shape



complete linkage

single linkage

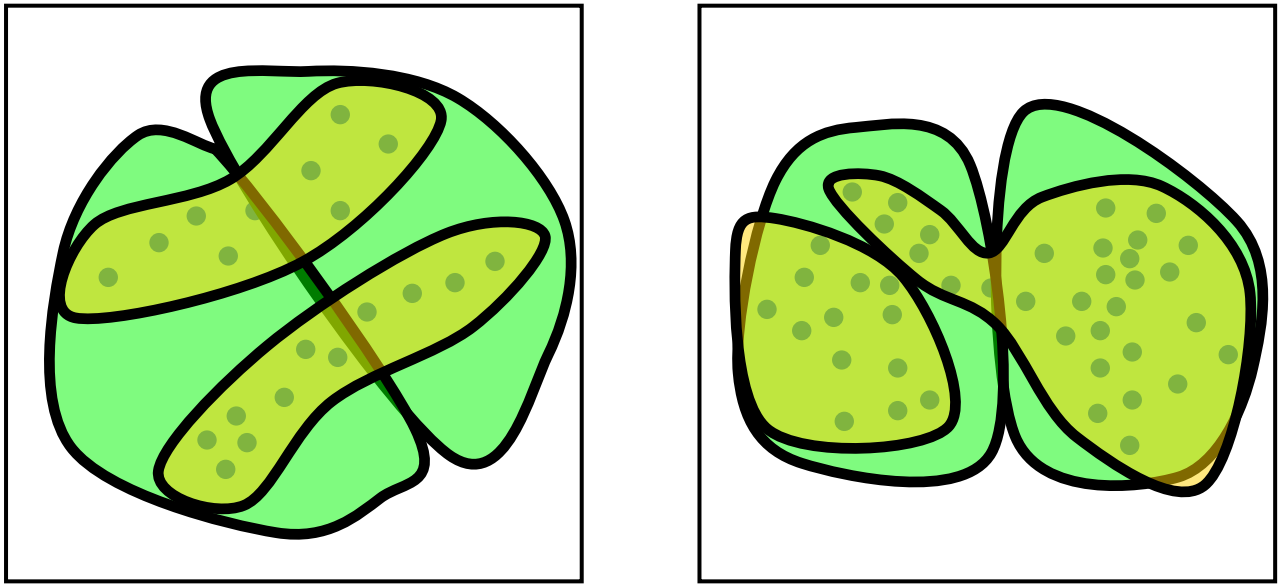
Linkage and cluster shape (2)



Complete linkage

Single linkage

Linkage and cluster shape (3)



Complete linkage

Single linkage

Question: hierarchical clustering

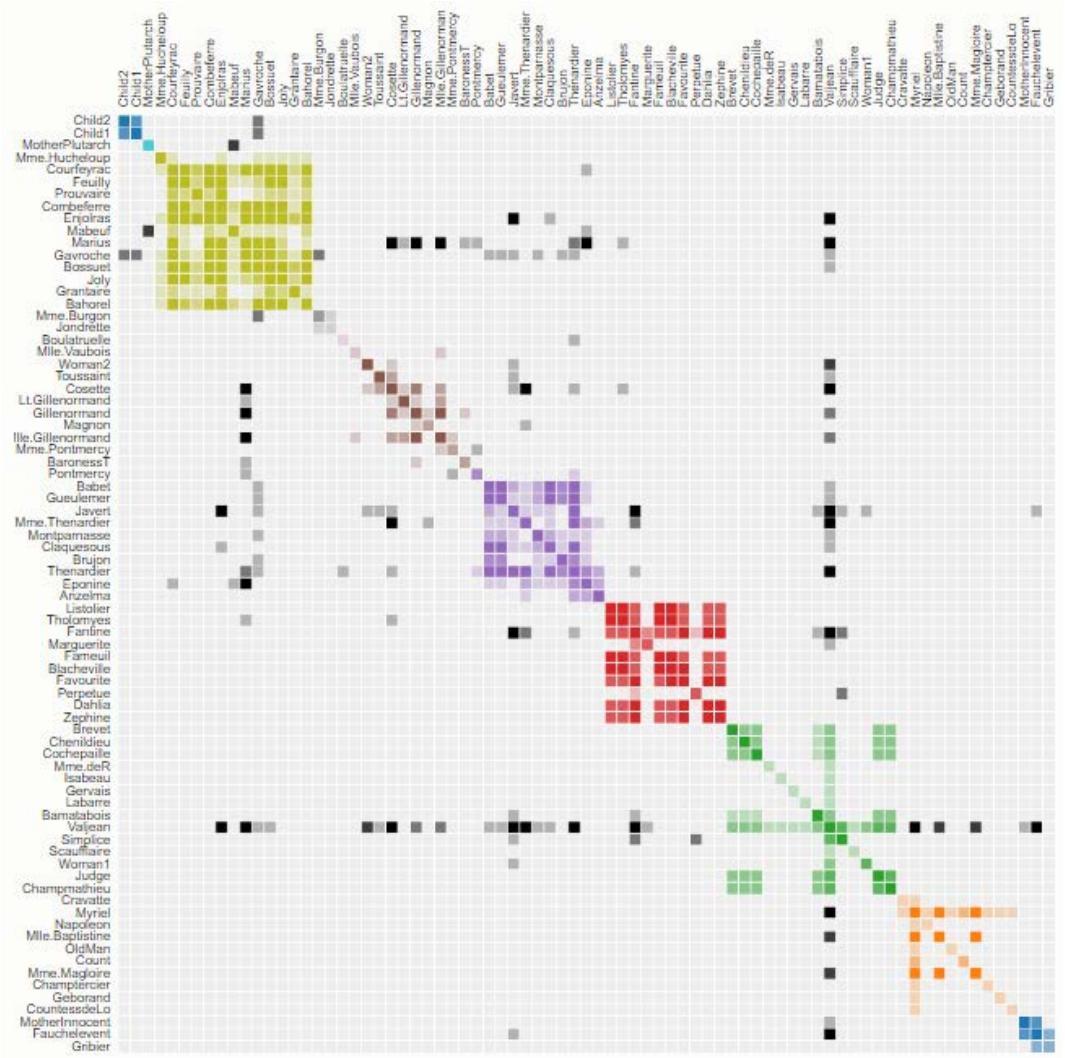
- Given is a dataset: (4, 10), (7,10), (4, 8), (10, 5), (11, 4), (3, 4), (9, 3), (5, 2)
- Cluster the points using agglomerative clustering
- Use single link method with Euclidean distance
- Stopping criterion: 3 clusters
- Detail your methodology, show steps and dendrogram

Hierarchical clustering summary

- Pros
  - Dendrogram gives overview of all possible clusterings
  - Linkage type allows to find clusters of varying shapes
  - Different dissimilarity measures can be used

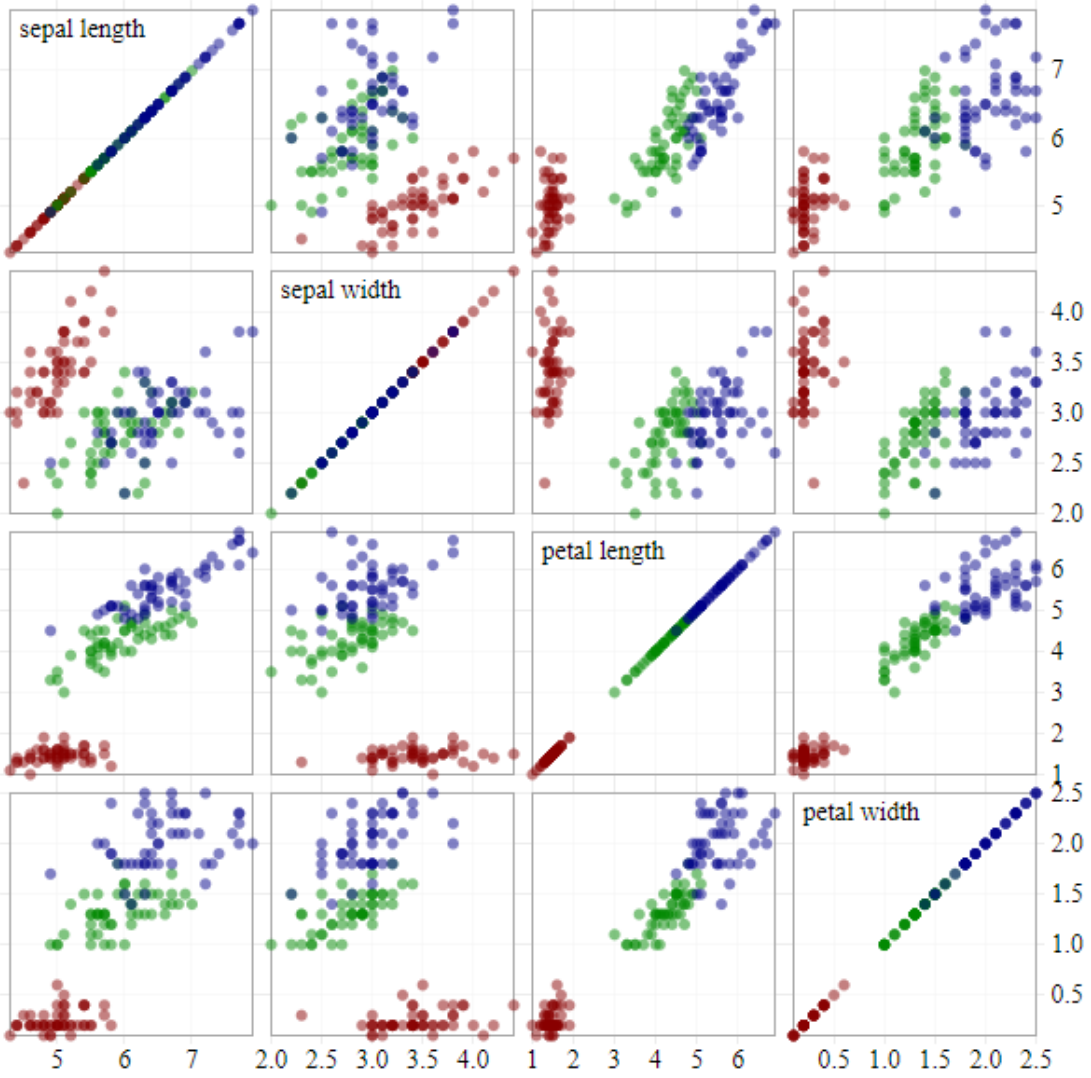
- Cons
  - Computationally intensive
  - Clustering limited to “hierarchical nestings”

Clusters visualized: Co-occurrence heatmap



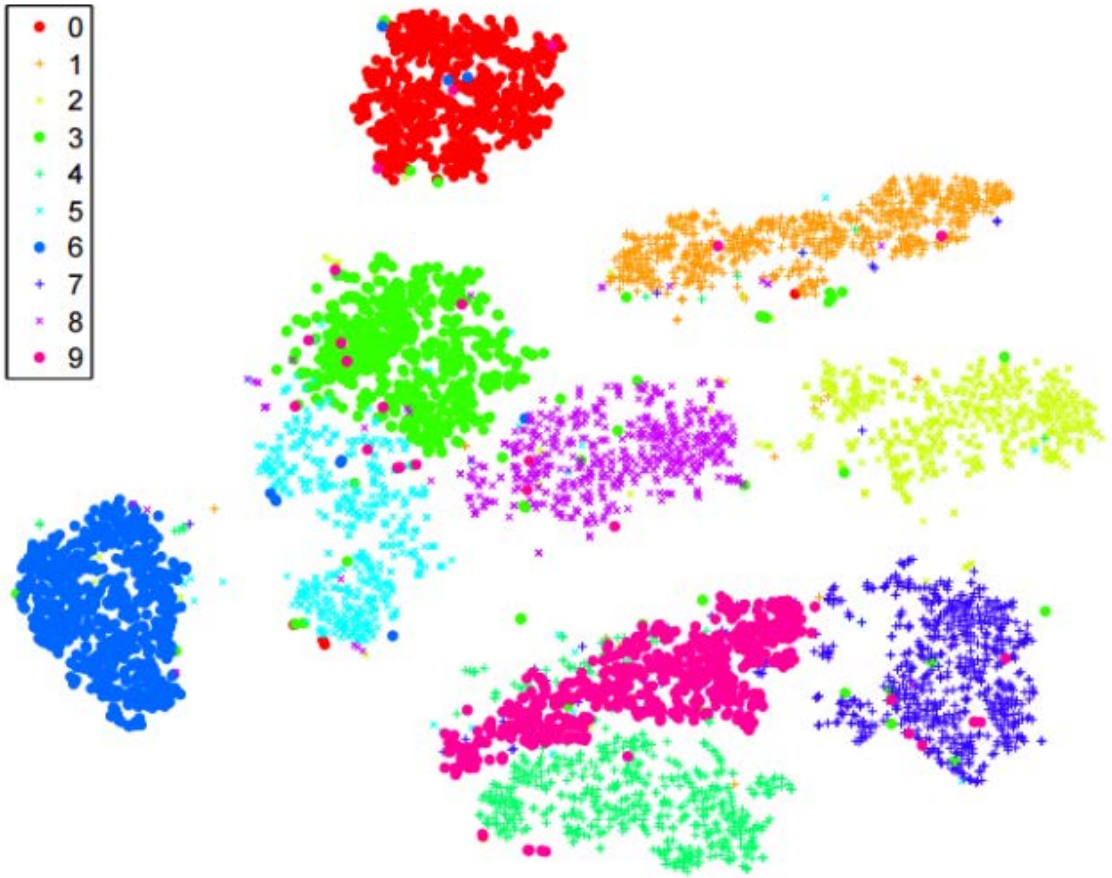
https://bost.ocks.org/mike/miserables/

Clusters visualized: scatterplot matrix



http://benjiec.github.io/scatter-matrix/demo/demo.html#

Clusters visualized: 2D embedding



https://lvdmaaten.github.io/tsne/

Clustering summary

- We can classify when we don’t have (training) labels: clustering
- Definition of cluster is vague
- For clustering we need to :
  - Define distance measure
  - Define criterion function to evaluate a clustering
  - select clustering algorithm
- Discussed clustering algorithms
  - Hierarchical clustering
  - k-means clustering