Lecture: Machine Learning for Data Science

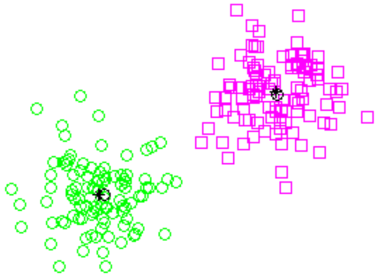
Winter semester 2021/22

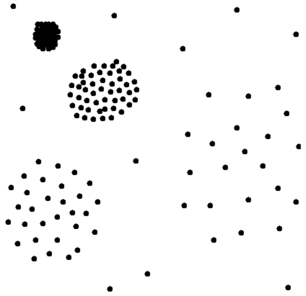
Lecture 11: Unsupervised learning –Hierarchical clustering

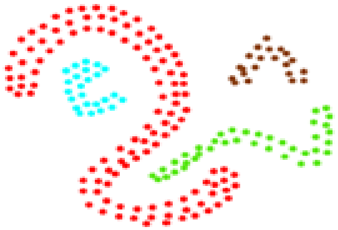
Prof. Dr. Eirini Ntoutsi

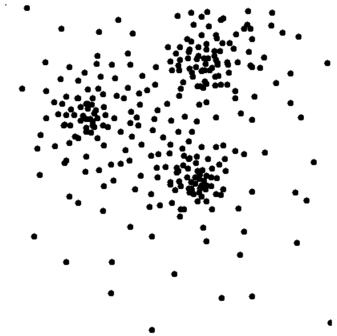
Clustering topics covered in this lecture

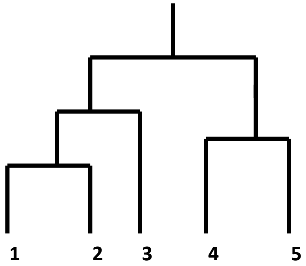
- Partitioning-based clustering
  - k-Means, k-Medoids
- Hierarchical clustering
- Density-based clustering
- Grid-based clustering
- Soft clustering
- Clustering evaluation

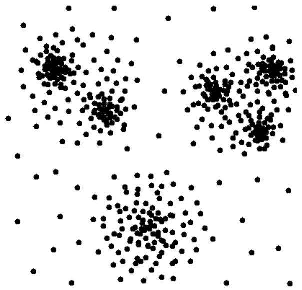


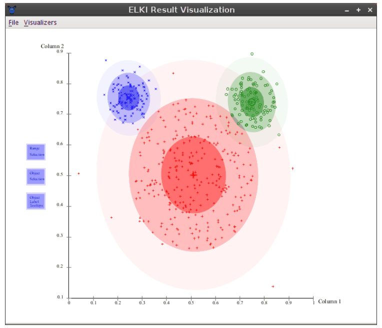












Outline

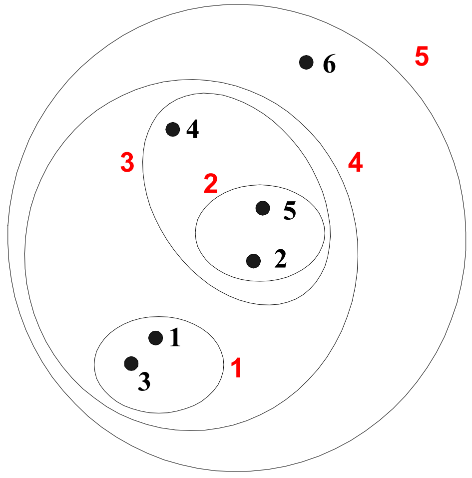
- Hierarchical clustering basics
- Hierarchical clustering methods
- Bisecting k-Means
- Things you should know from this lecture & reading material

Hierarchical-based clustering

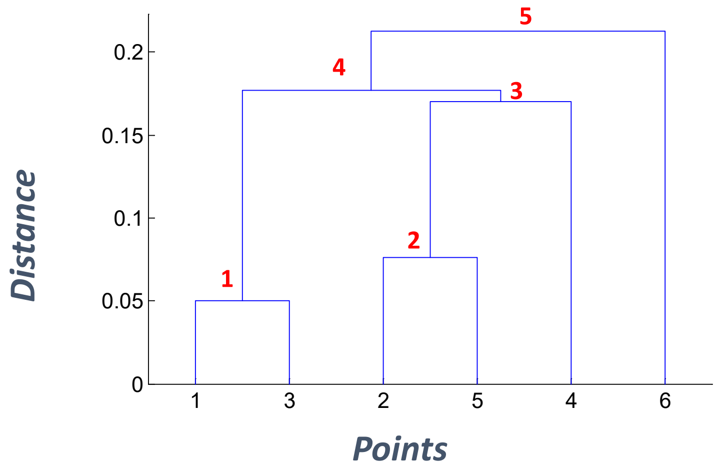
- Produces a set of nested clusters organized as a hierarchical tree

- Can be visualized also as a dendrogram
  - A tree like diagram that records the sequences of merges or splits & cluster memberships
  - The height at which two clusters are merged in the dendrogram reflects their distance

- An instance can belong to multiple clusters.
  - The assignement though is still hard



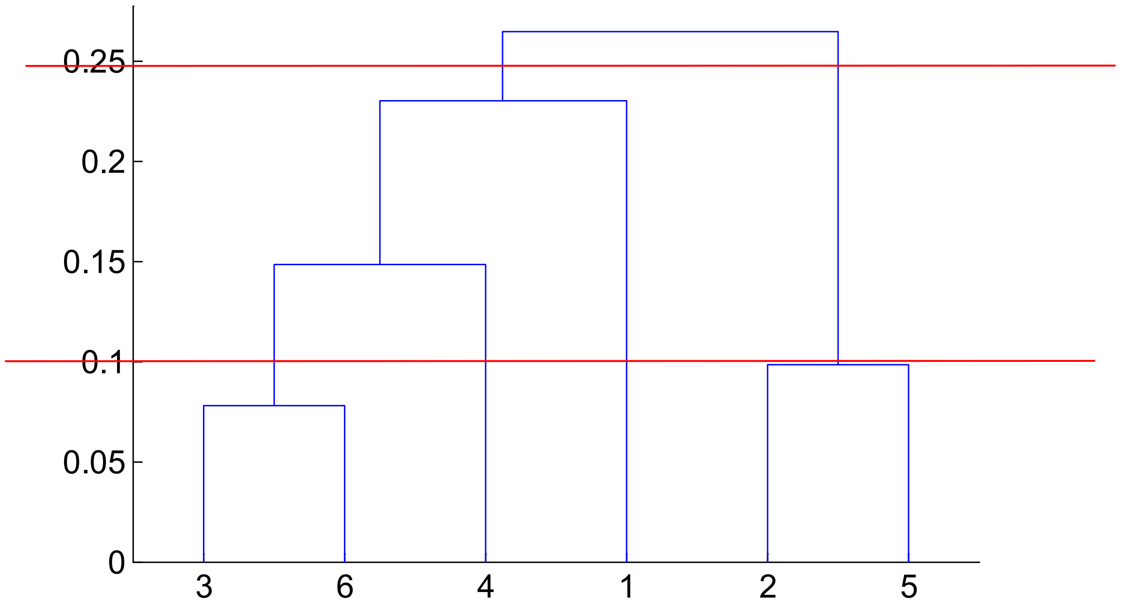
Nested clusters



Dendrogram

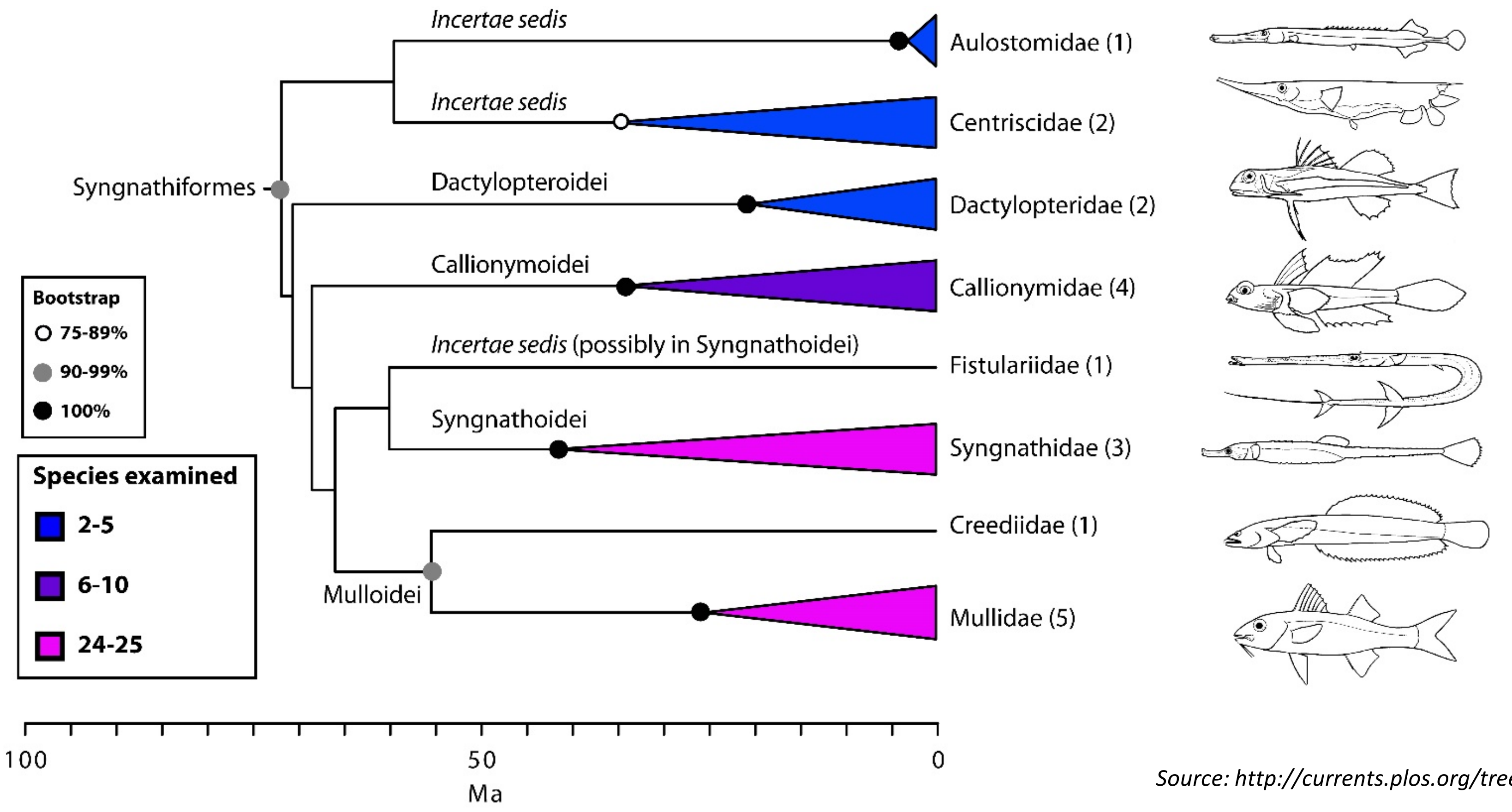
Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
- A clustering can be obtained by 'cutting' the dendrogram at the proper level
  - Cutting based on distance (i.e., I want ≤ 0.1 distance)
  - Cutting based on the number of clusters (i.e., I want 2 clusters)



Applications of hierarchical clustering 1/3

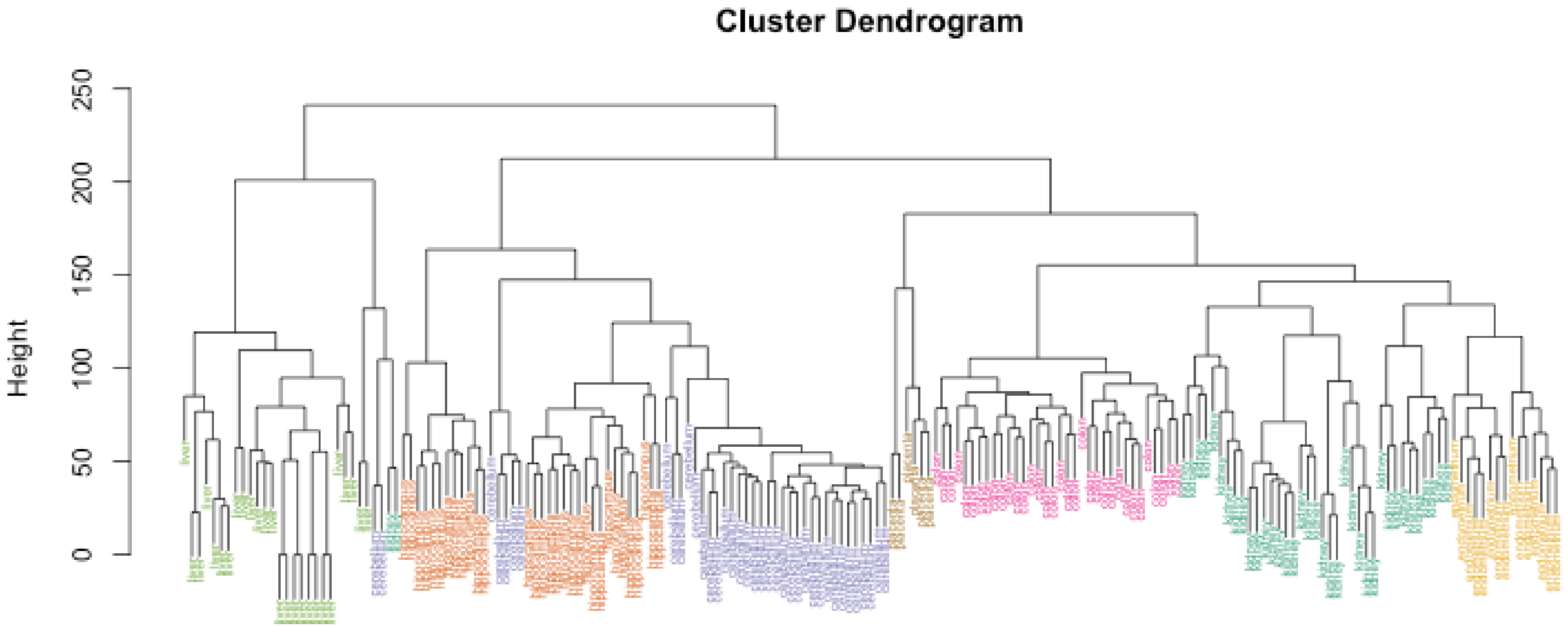
- The dendrogram of clusters may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)



Source: http://currents.plos.org/treeoflife/article/the-tree-of-life-and-a-new-classification-of-bony-fishes/

Applications of hierarchical clustering 2/3

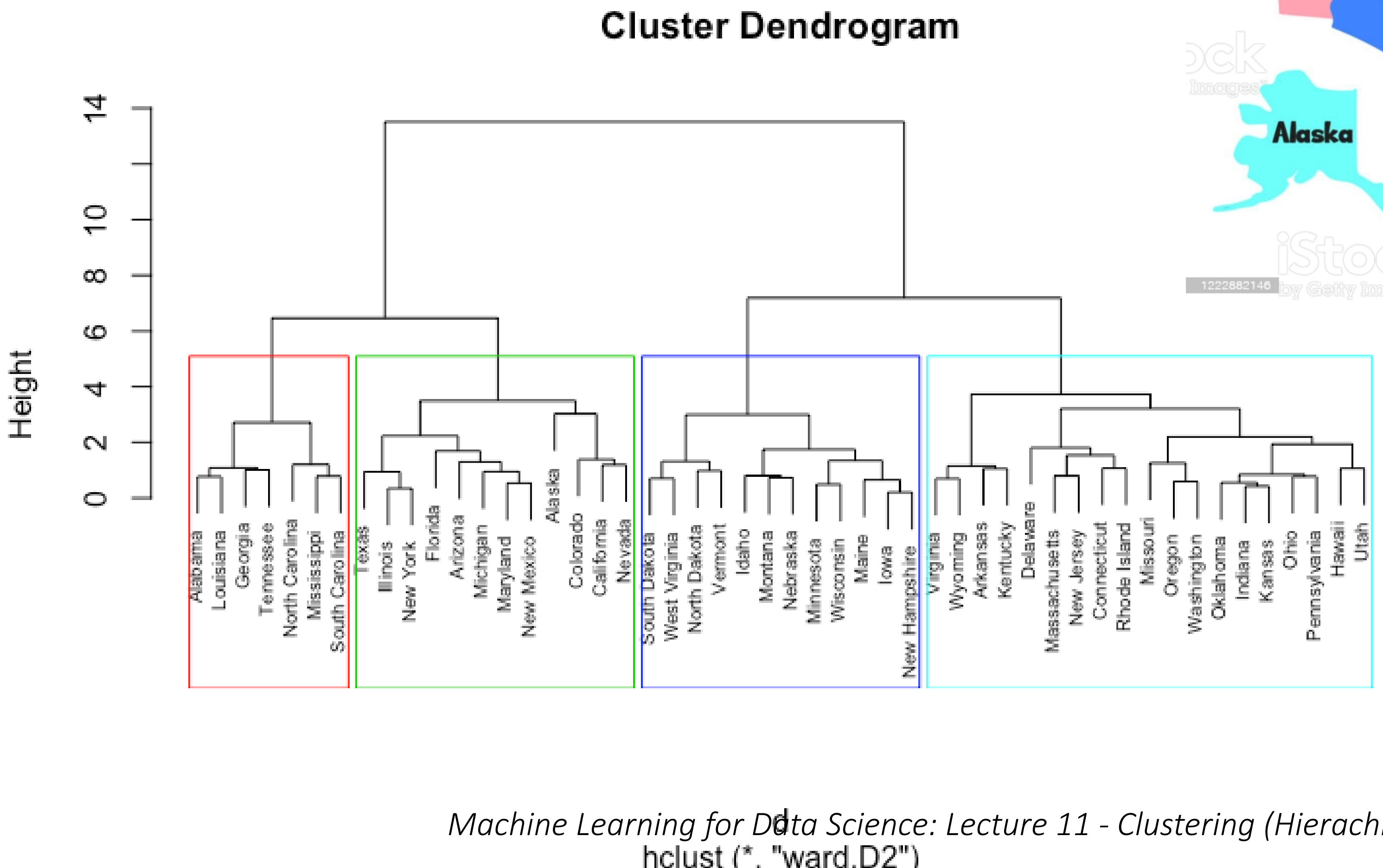
- The dendrogram of clusters may correspond to meaningful taxonomies
  - Dendrogram showing hierarchical clustering of tissue gene expression data with colours denoting tissues.



Source: http://genomicsclass.github.io/book/pages/clustering_and_heatmaps.html

Applications of hierarchical clustering 3/3

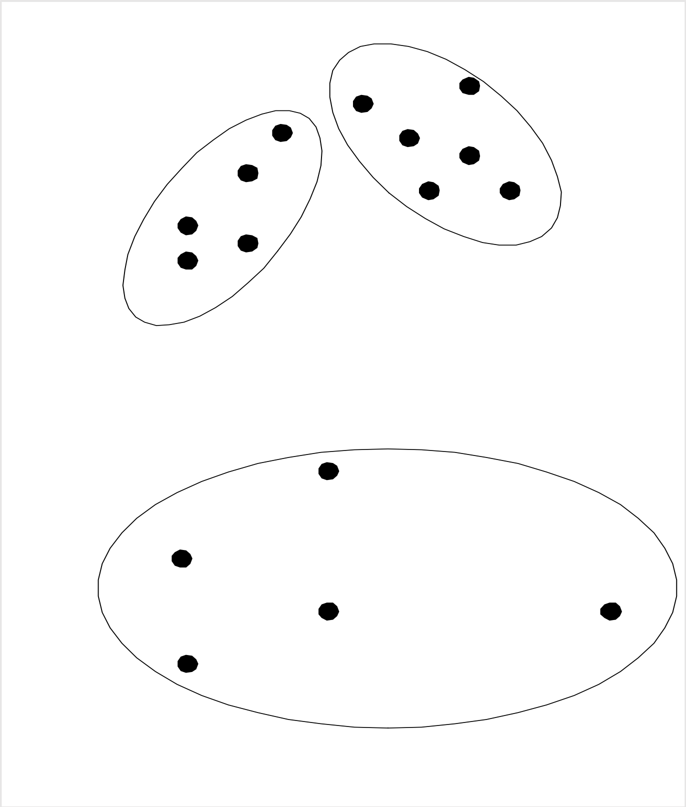
- The dendrogram of clusters may correspond to meanir
  - USArrests dataset: statistics in arrests per 100,000 resid
    US states in 1973.



Source: https://uc-r.github.io/hc_clustering

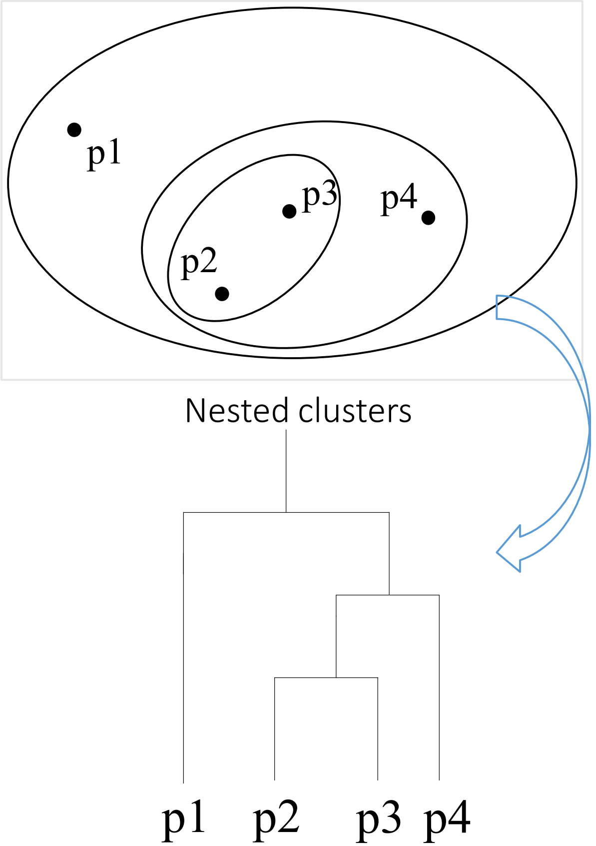
Hierarchical vs Partitioning

Partitioning clustering



Partitioning algorithms typically have global objectives, e.g., *k*-Means

Dendrogram



Hierarchical clustering algorithms typically have local objectives

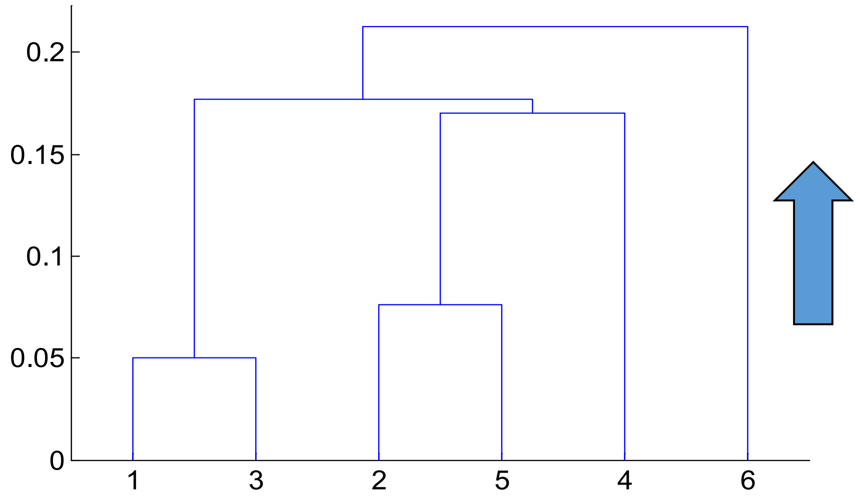
Outline

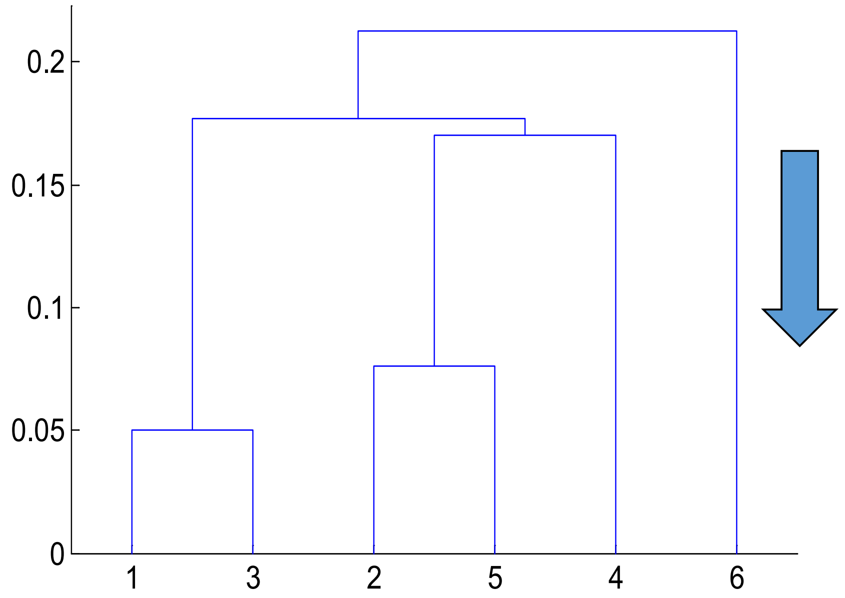
- Hierarchical clustering basics
- Hierarchical clustering methods
- Bisecting k-Means
- Things you should know from this lecture & reading material

Hierarchical clustering methods

Two main types of hierarchical clustering

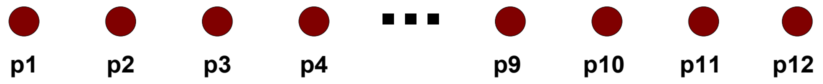
- Agglomerative or AGNES (Agglomerative Nesting)
  - Bottom-up approach
  - Start with the points as individual clusters
  - At each step, merge the closest pair of clusters
    - until only one cluster (or *k* clusters) left
- Divisive or DIANA (Divisive analysis)
  - Top-down approach
  - Start with one, all-inclusive cluster
  - At each step, split a cluster until each cluster contains a single point (or there are *k* clusters)
- Merge or split one cluster at a time





Hierarchical clustering methods

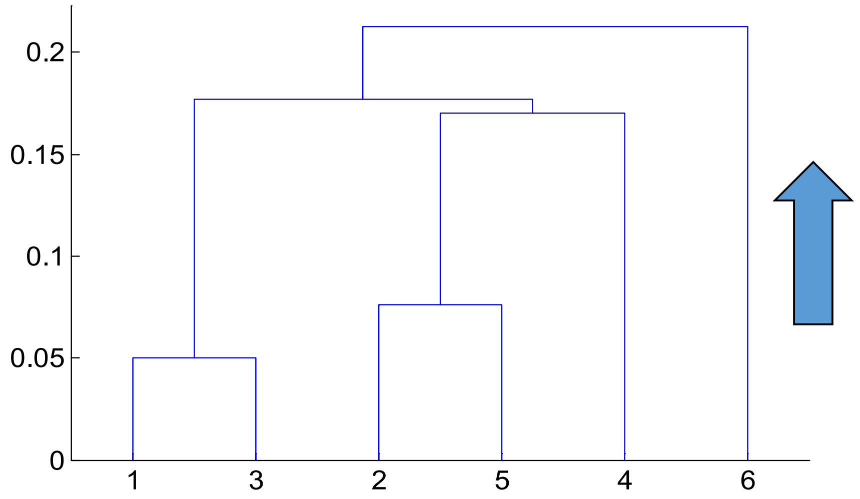
- Hierarchical algorithms use a similarity or distance matrix to decide on which cluster to split/merge next
  - Employed distance/similarity function depends on the application



	p1	p2	p3	...	p12
p1					
p2					
p3					
...					
p12					

Proximity matrix

Agglomerative clustering algorithm



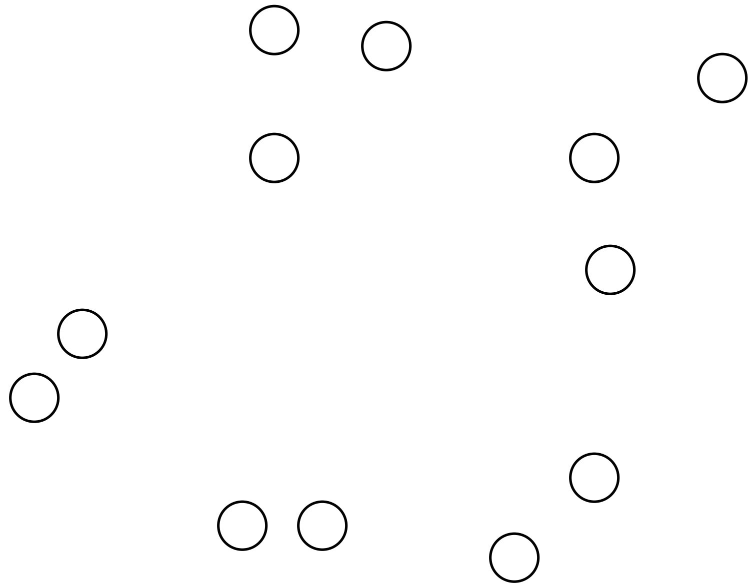
- Most popular hierarchical clustering technique
- Basic algorithm is straightforward

```
Compute the proximity matrix
Let each data point be a cluster
Repeat
    Merge the two closest clusters
    Update the proximity matrix
Until only a single cluster remains
```

- Key operation: the computation of the proximity of two clusters
  - Different approaches (single link, complete link, …..) which lead to different algorithms

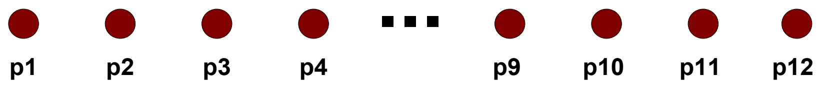
Starting situation

- Start with clusters of individual points and a proximity matrix



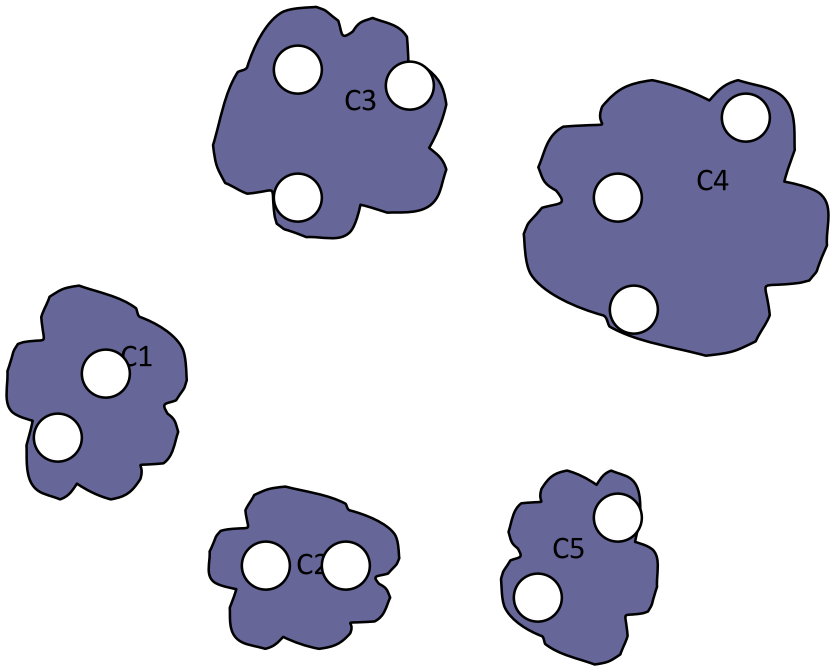
	p1	p2	p3	...	p12
p1					
p2					
p3					
...					
p12					

Proximity matrix



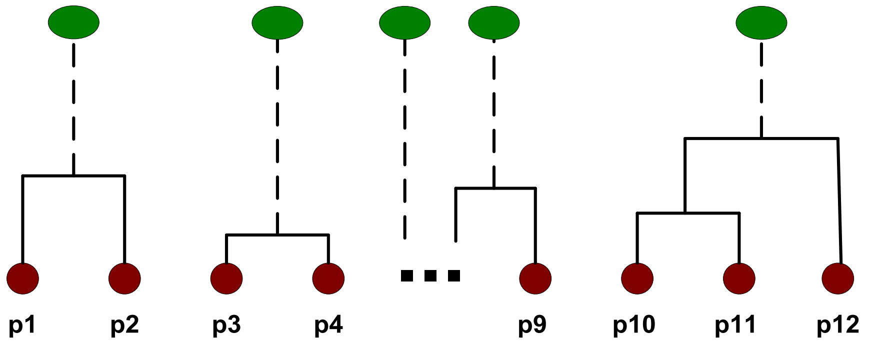
Intermediate situation I

- After some merging steps, we have some clusters



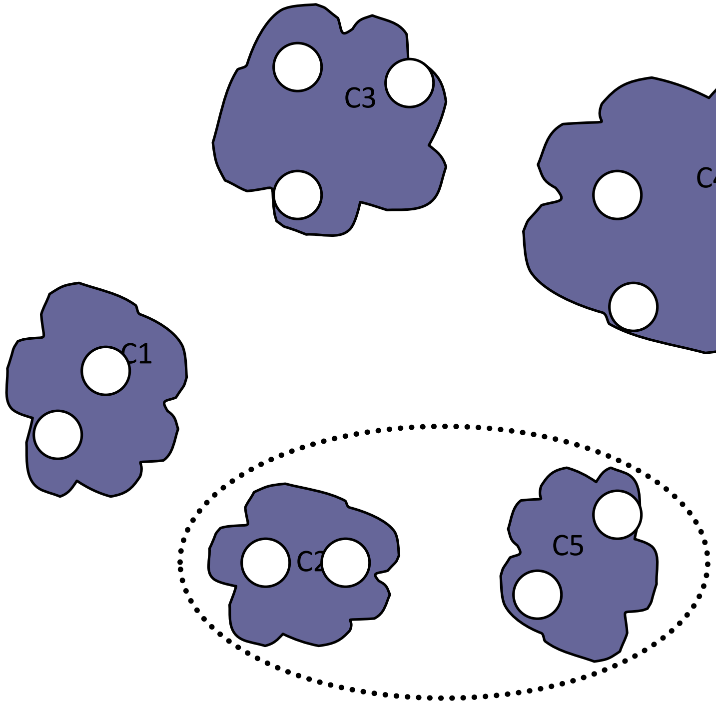
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

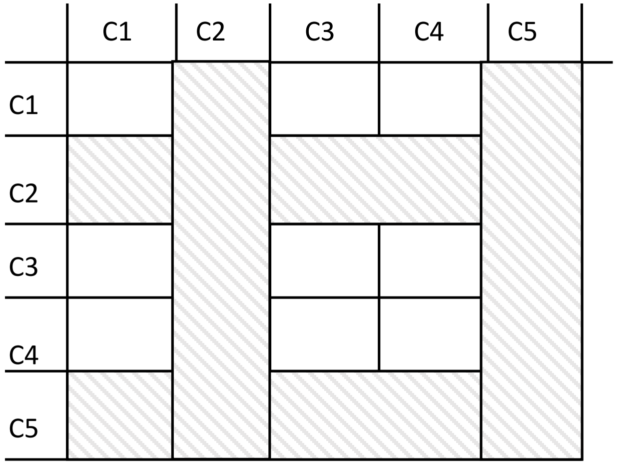
Proximity matrix



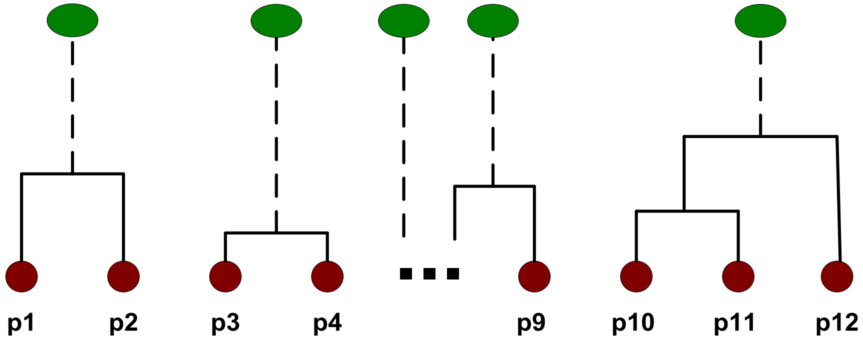
Intermediate situation II

- We decide to merge the two closest clusters (C_2 and C_5) and update the proximity matrix.



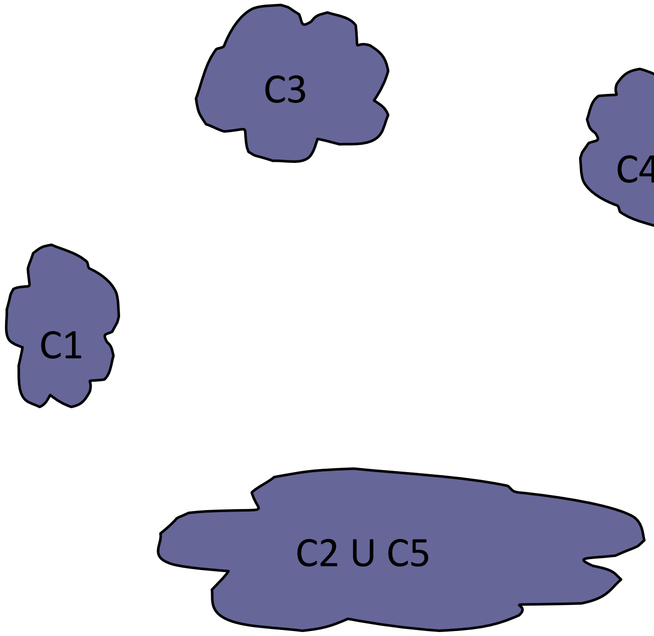


Proximity matrix



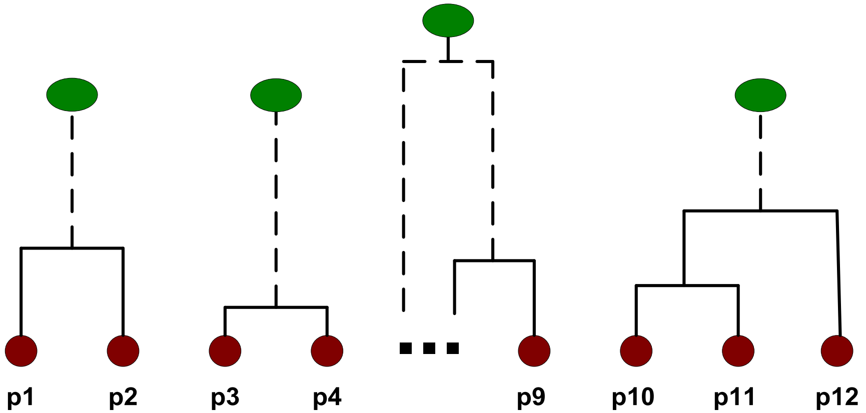
Merging

- Two major questions for merging
  - How we identify the closest pair of clusters to be merged?
  - How do we update the proximity matrix?



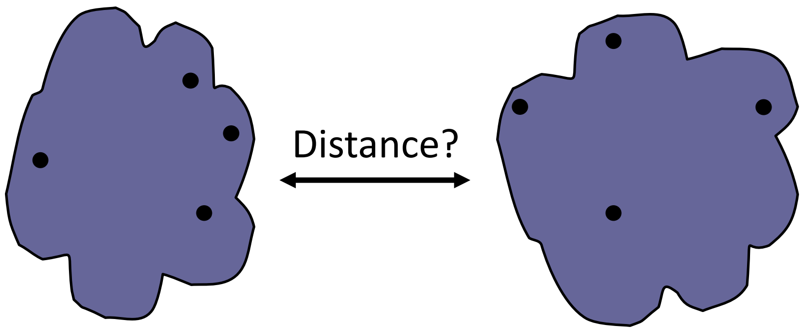
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity matrix



Distance between clusters

- Each cluster is a set of points
  - How do we compare two sets of points/clusters?



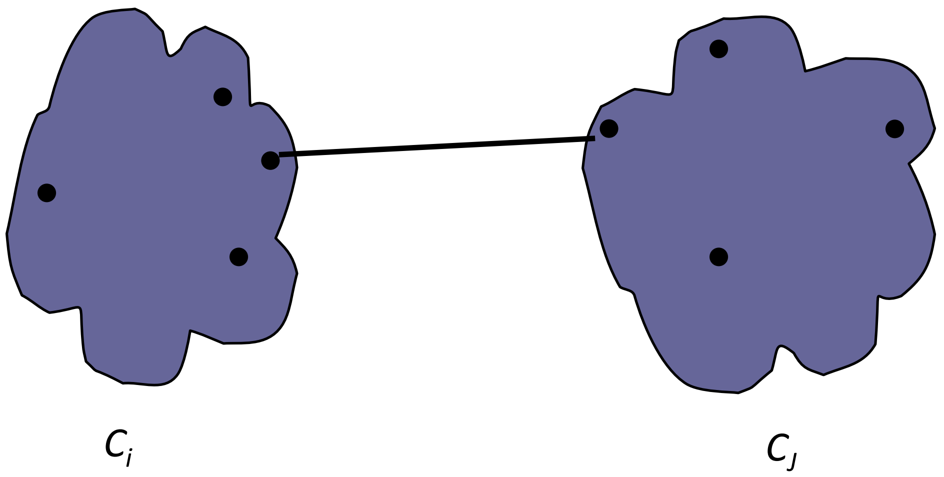
- A variety of different methods
  - Single link (or MIN)
  - Complete link (or MAX)
  - Group average
  - Centroid-distance
  - Medoid-distance
  - Other methods driven by an objective function
    - Ward's Method uses squared error

Distance between clusters: Single link distance or MIN

- Single link (or MIN) distance between C_i and C_j is the *minimum* distance between *any* object in C_i and any object in C_j, i.e.,

$$dis_{sl}\left(C_i, C_j\right) = \min_{x,y}\left\{d(x,y) \middle| x \in C_i, y \in C_j\right\}$$

  - i.e., the distance is defined by the two closest objects (shortest edge)

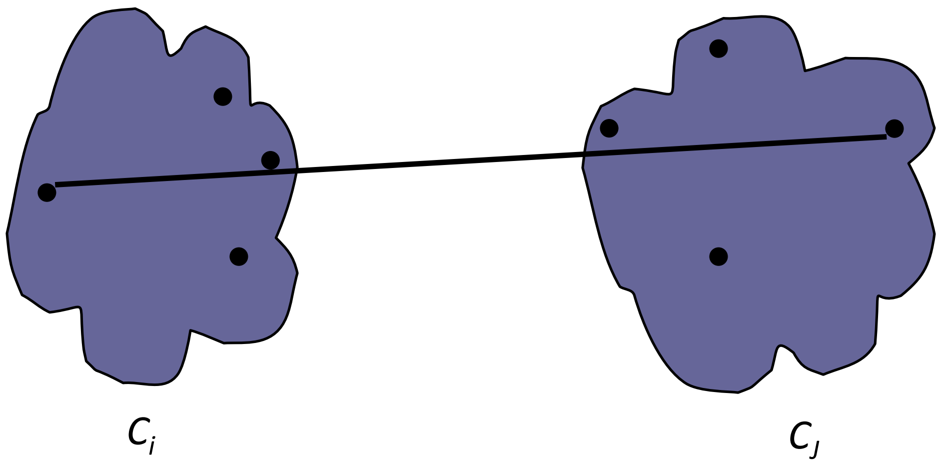


Distance between clusters: Complete link or MAX

- Complete link (or MAX) distance between C_i and C_j is the *maximum* distance between *any* object in C_i and any object in C_j, i.e.,

$$dis_{cl}\left(C_i, C_j\right) = \max_{x,y}\left\{d(x,y) \middle| x \in C_i, y \in C_j\right\}$$

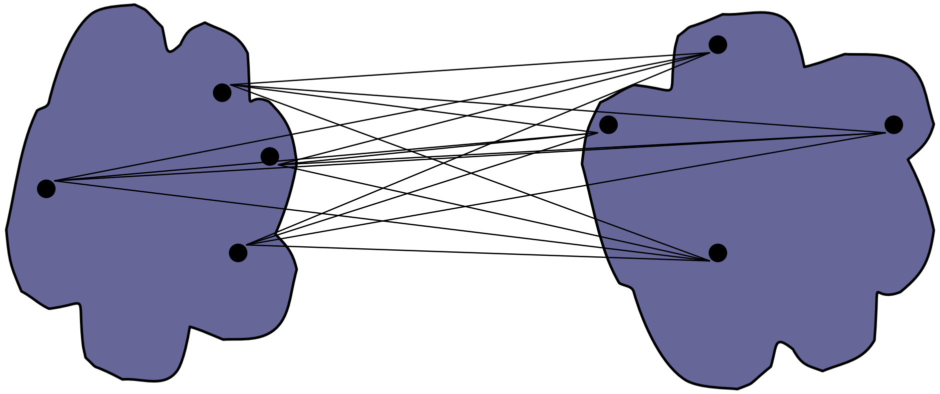
  - i.e., the distance is defined by the two most dissimilar objects (longest edge)



Distance between clusters: Group average

- Group average distance between C_i and C_j is the average distance between any object in C_i and any object in C_j, i.e.,

$$dis_{avg}\left(C_i, C_j\right) = \frac{\sum_{x \in C_i, y \in C_j} d(x, y)}{|C_i||C_j|}$$



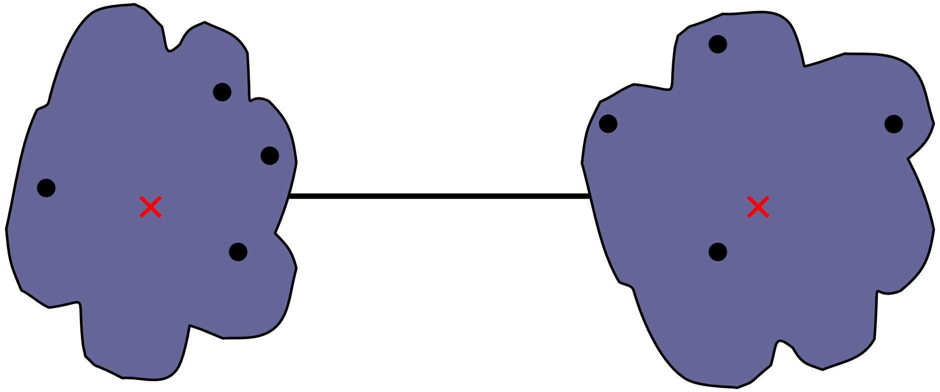
Distance between clusters: Centroid distance

- Centroid distance between C_i and C_j is the distance between the centroid c_i of C_i and the centroid c_j of C_j, i.e.,

$$dis_{centroids}\left(C_i, C_j\right) = d(c_i, c_j)$$

$$c_m = \frac{\sum_{i=1}^{n} p_i}{n}$$

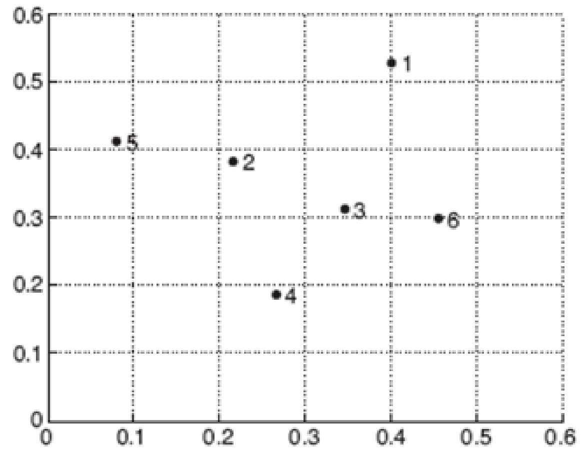
Centroid of a cluster



Example

Dataset (6 2D points)

Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30



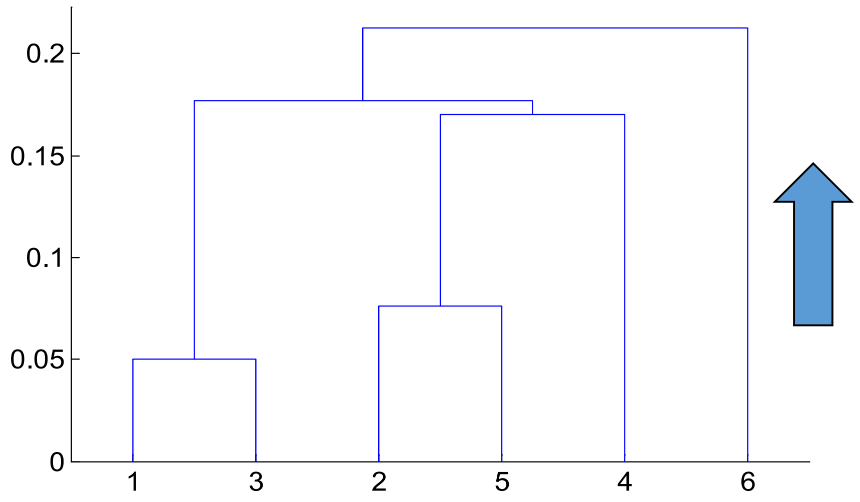
Distance matrix (Euclidean distance)

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Back to the pseudocode of the agglomerative clustering algorithm

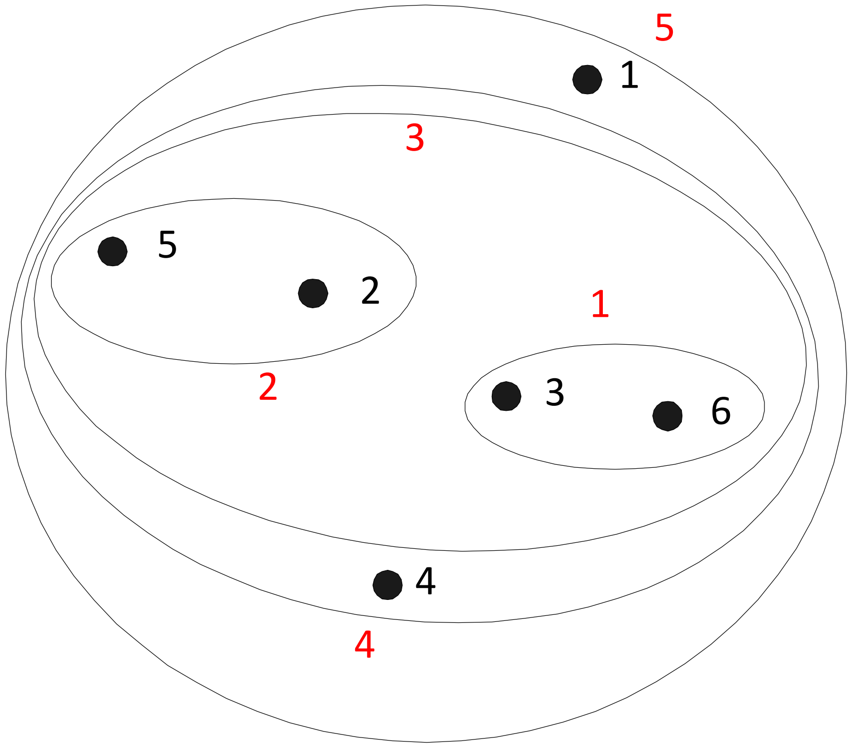
- Pseudocode of the algorithm

```
Compute the proximity matrix
Let each data point be a cluster
Repeat
    Merge the two closest clusters
    Update the proximity matrix
Until only a single cluster remains
```



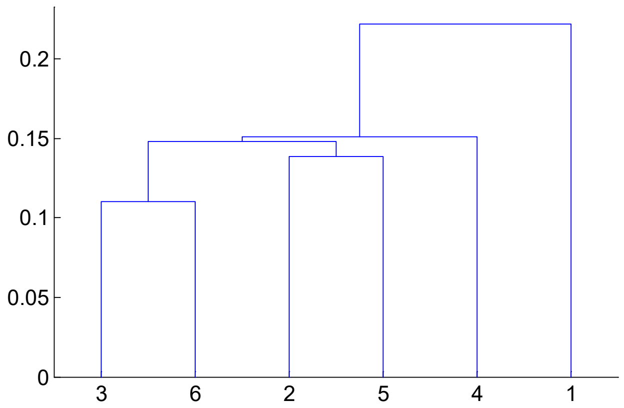
Single link distance or MIN agglomerative clustering algorithm

- Similarity of two clusters is based on the most similar (closest) pair of objects
  - Determined by one pair of points $dis_{sl}(C_i, C_j) = \min_{x,y}\{d(x,y) | x \in C_i, y \in C_j\}$



Nested clusters

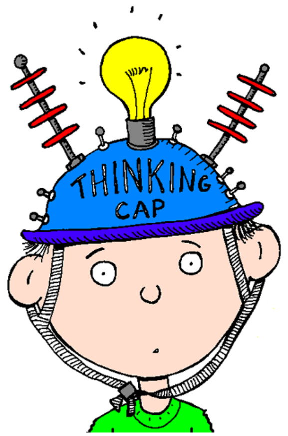
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00



Dendrogram

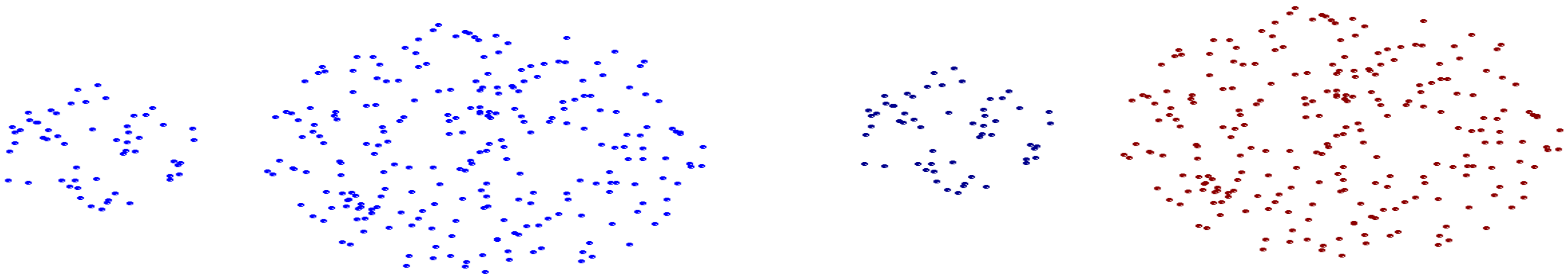
Short break (5')

- Given the following 1-dimensional dataset, build a hierarchical agglomerative clustering using single-link distance



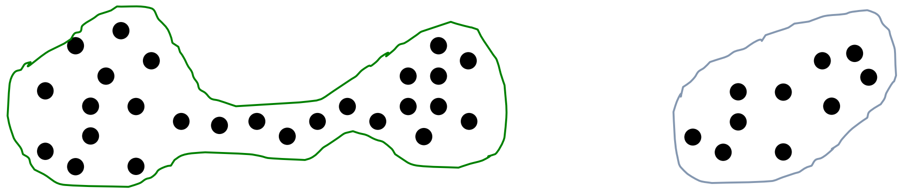
Single link distance (MIN): strengths

- Can discover clusters of arbitrary shapes



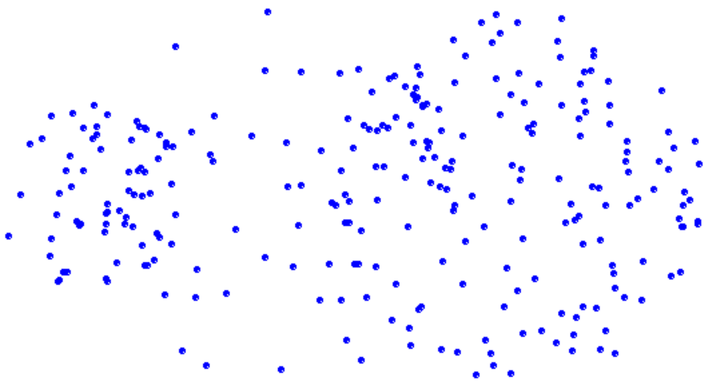
Original points

Two clusters

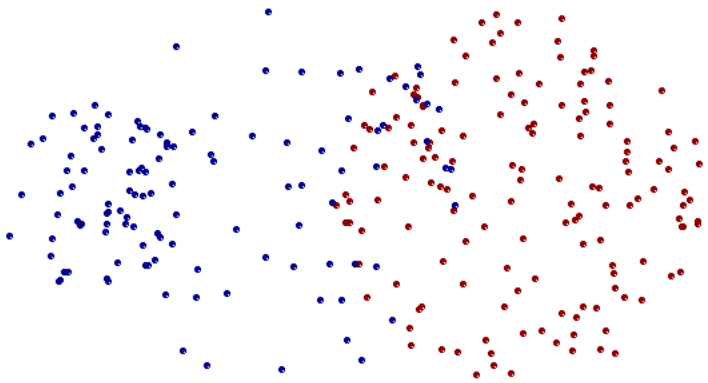


Single link distance (MIN): limitations

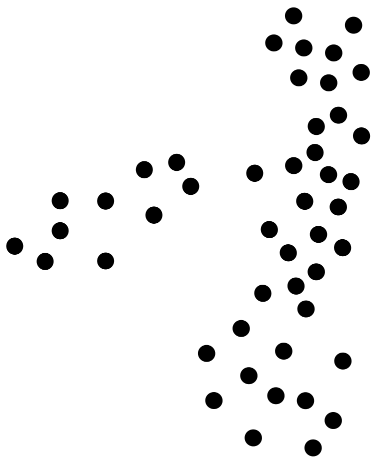
- Sensitive to noise and outliers



Original points



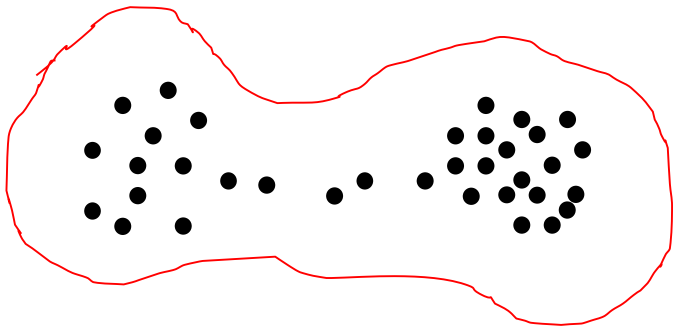
Two clusters



- DBSCAN (next lecture) can be viewed as a robust variant of single link distance
  - It excludes noisy points between clusters to avoid undesirable chaining effects.

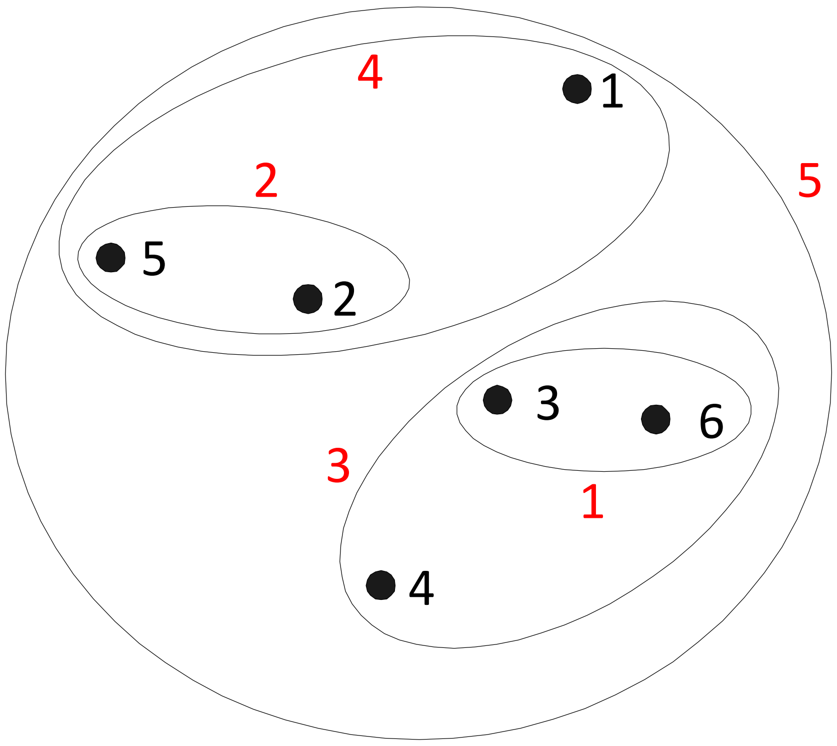
Single link distance (MIN): limitations

- Produces long, elongated clusters (chain-like clusters)



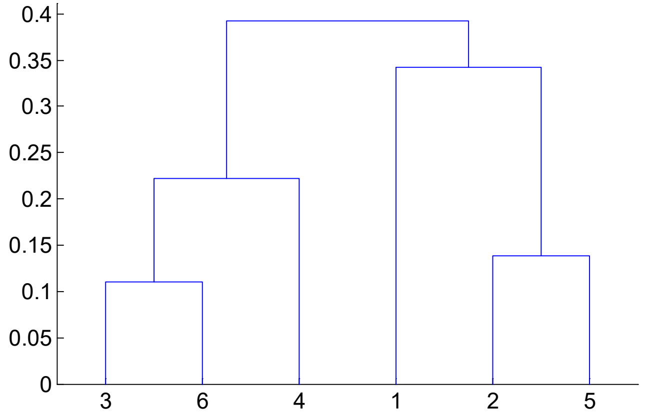
Complete link distance or MAX agglomerative clustering algorithm

- Similarity of two clusters is based on the least similar (most distant) pair of objects
  - Determined by one pair of points $dis_{cl}(C_i, C_j) = \max_{x,y}\{d(x,y) | x \in C_i, y \in C_j\}$



Nested clusters

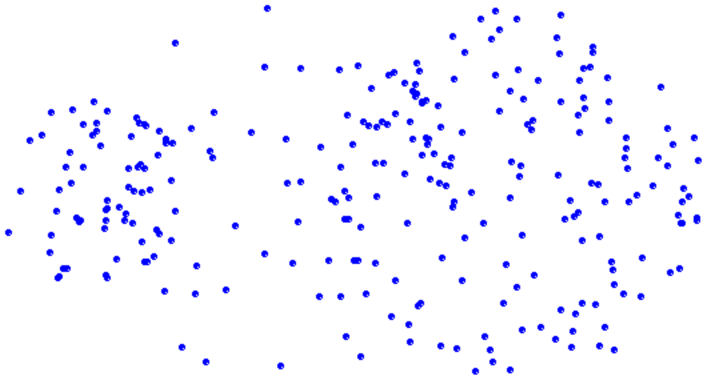
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00



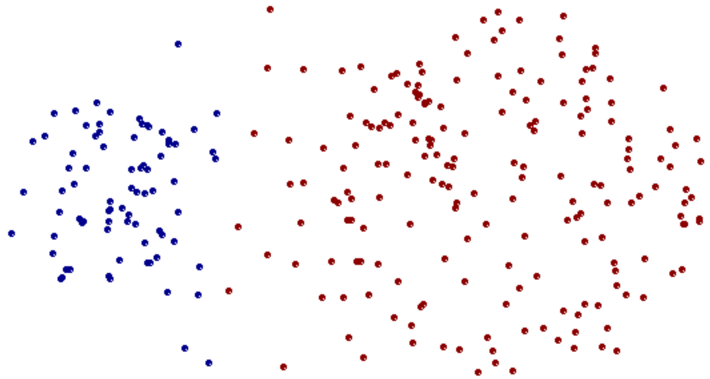
Dendrogram

Complete link distance (MAX): strengths

- Less susceptible to noise and outliers and comparing to MIN



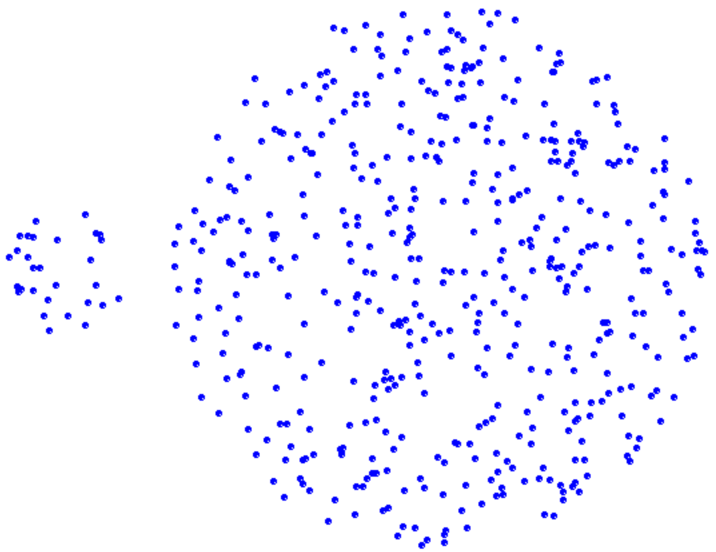
Original points



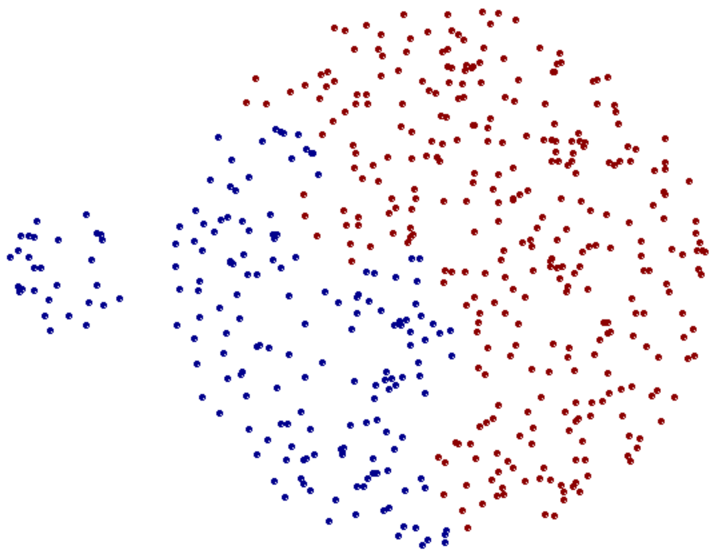
Two clusters

Complete link distance (MAX): limitations

- Because it focuses on minimizing the diameter of the cluster, it will create clusters so that all of them have similar diameter
  - If there are natural larger clusters than others, it tends to break large clusters



Original points



Two clusters

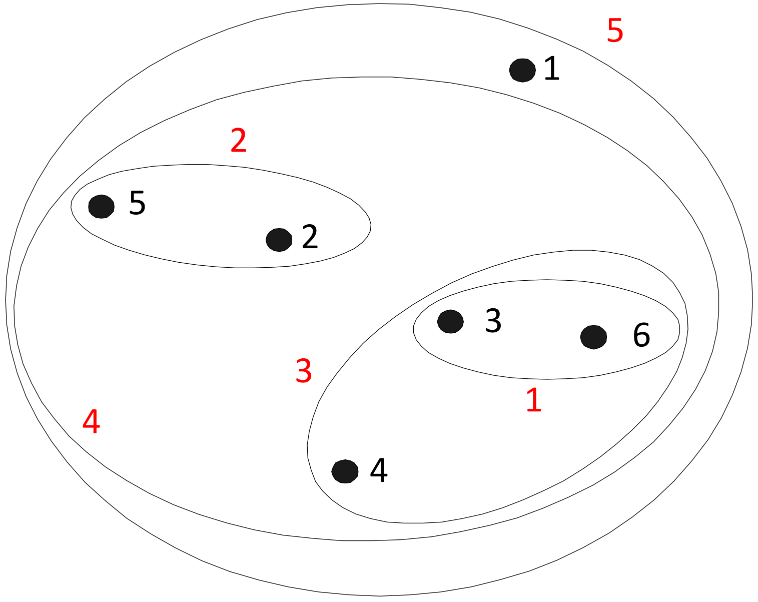
Short break (5’)

- Given the following 1-dimensional dataset, build a hierarchical agglomerative clustering using complete-link distance

(Group) Average-link distance agglomerative clustering algorithm

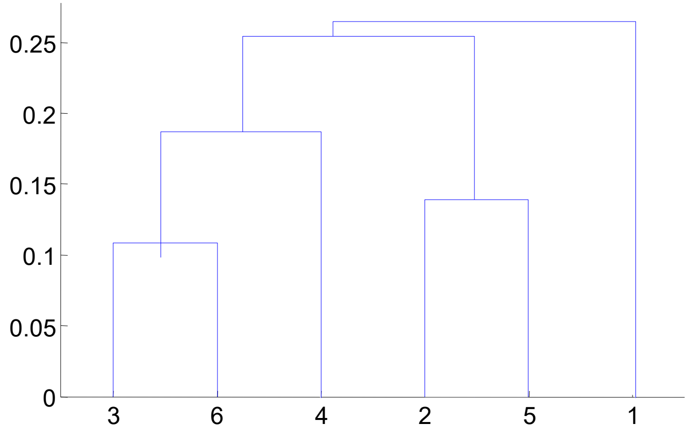
- Proximity of two clusters is the average of pairwise distances between objects in the two clusters.
  - Determined by all pairs of points in the two clusters

$$dis_{avg}(C_i, C_j) = \frac{\sum_{x \in C_i, y \in C_j} d(x,y)}{|C_i||C_j|}$$



Nested clusters

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00



Dendrogram

(Group) Average-link distance: strengths and limitations

- Compromise between Single and Complete Link

- Strengths
  - Less susceptible to noise and outliers

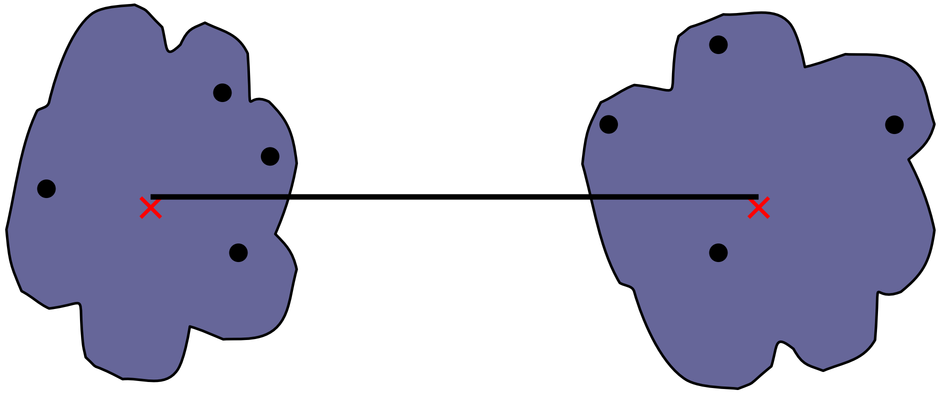
- Limitations
  - Biased towards spherical clusters

Centroid-link distance agglomerative clustering algorithm

- The distance between two clusters is the distance of their corresponding centroids

$$dis_{centroids}\left(C_i, C_j\right) = d(c_i, c_j)$$

- Difference to other measures (often considered bad): the possibility of inversions
  - Two clusters that are merged at step *k* might be more similar than the pair of clusters merged in step *k*-1
  - For the other methods, distance between clusters monotonically increases (or at worst does not increase)



Ward's method

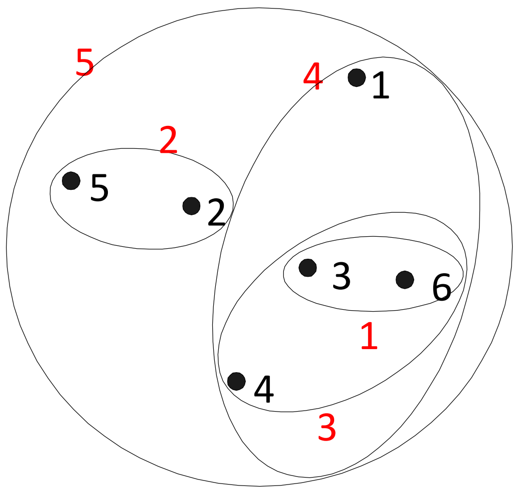
- Ward's method or Ward's minimum variance method
- Clusters are represented by centroids
- The proximity between two clusters is measured in terms of the increase in SSE (sum of squared error) that results from merging the two clusters

$$D_W(C_i, C_j) = \sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 - \sum_{x \in C_{ij}} (x - r_{ij})^2$$

r_i: centroid of C_i
r_j: centroid of C_j
r_{ij}: centroid of C_{ij}

- At each step, merge the pair of clusters that leads to minimum increase in total inter-cluster variance after merging.

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

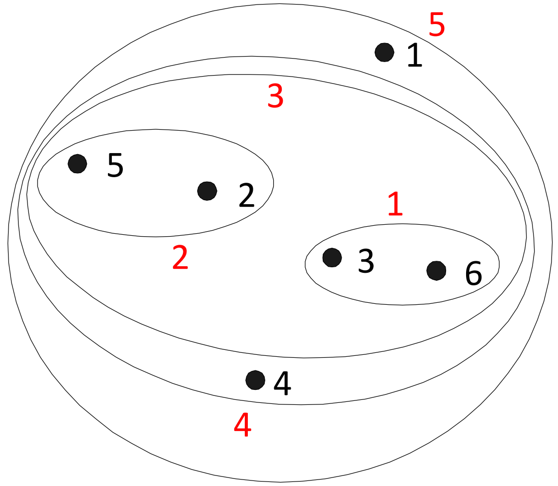


Nested clusters

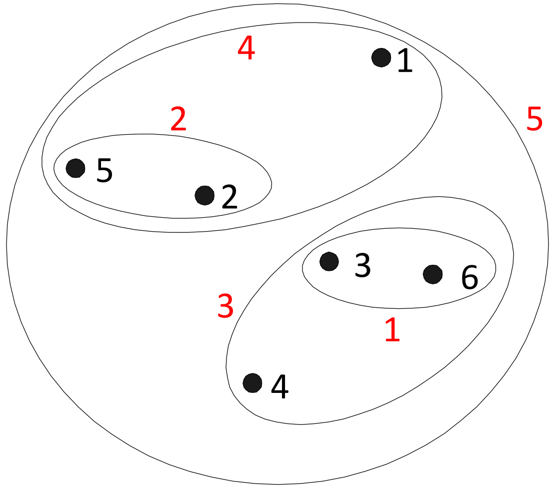
Ward’s method cont’

- Ward’s method seems similarly to *k*-Means: it tries to minimize the sum of square distances of points from their cluster centroids, but not globally
- Less susceptible to noise and outliers
- Biased towards spherical clusters

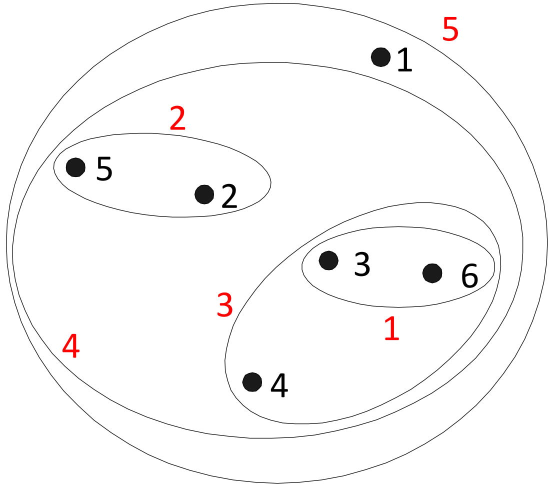
Comparison of the different methods



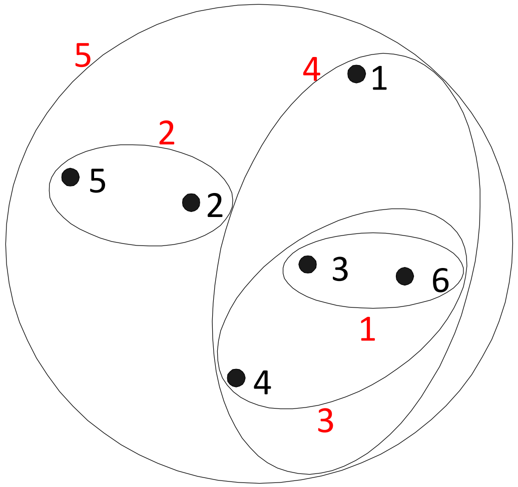
Single link (MIN)



Complete link (MAX)



Group average



Ward’s method

Hierarchical methods: complexity

- $O(n^2)$ space to store the proximity matrix
  - n is the number of points.

- $O(n^3)$ time in most of the cases
  - There are n steps and at each step the size, n^2, proximity matrix must be updated and searched
  - Complexity can be reduced to $O(n^2 \log(n))$ time for some approaches using appropriate data structures

Hierarchical clustering: overview

- No knowledge on the number of clusters
- Produces a hierarchy of clusters, not a flat clustering
  - A single clustering can be obtained from the dendrogram
- No backtracking: Merging decisions are final
  - Once a decision is made to combine two clusters, it cannot be undone
- Lack of a global objective function
  - Decisions are local, at each step
  - No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Breaking large clusters
  - Difficulty handling different sized clusters and convex shapes
- Inefficiency, especially for large datasets

Outline

- Hierarchical clustering basics
- Hierarchical clustering methods
- Bisecting k-Means
- Things you should know from this lecture & reading material

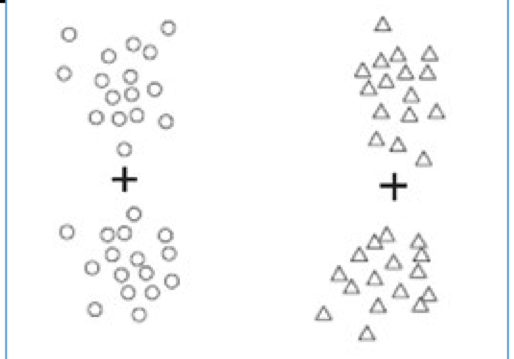
Bisecting k-Means

- Hybrid method, combines *k*-Means and hierarchical clustering
- Idea: first split the set of points into two clusters, select one of these clusters for further splitting, and so on, until *k* clusters remain.
- Pseudocode:

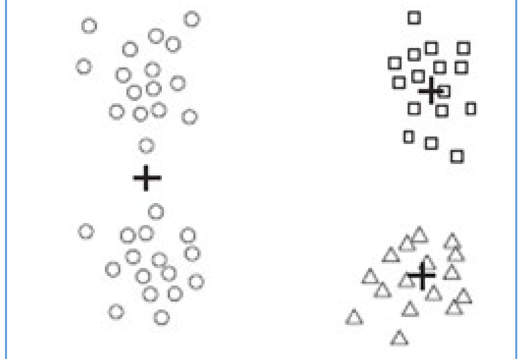
1. All data constitute one cluster ROOT.
2. The ROOT is partitioned in two clusters, its children, using K-Means for K=2.
3. In each subsequent iteration
   2.1. Choose among the leaf clusters the most inhomogeneous one,
   2.2. Partition it into two clusters with K-Means, K=2,
   until K leaf clusters are built.

- Which cluster to split? Different approaches
  - The one with the largest SSE (worse one)
  - Based on SSE and size
  - …

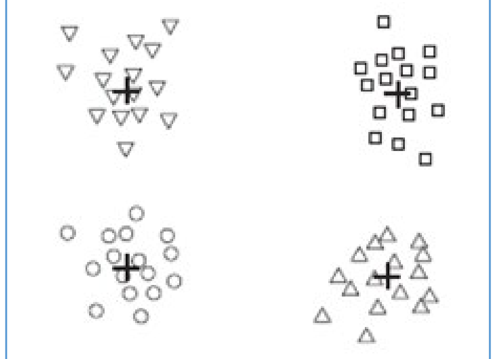
Bisecting k-Means: an example



(a) Iteration 1.



(b) Iteration 2.



(c) Iteration 3.

Outline

- Hierarchical clustering basics
- Hierarchical clustering methods
- Bisecting k-Means
- Things you should know from this lecture & reading material

Overview and Reading

- Overview
  - Hierarchical clustering basics
  - Agglomerative approach
  - Similarity measures between clusters
  - Bisecting kMeans

- Reading
  - Tan P.-N., Steinbach M., Kumar V book, Chapter 8.
  - Data Clustering: A Review, https://www.cs.rutgers.edu/~mlittman/courses/lightai03/jain99data.pdf
  - Nando de Freitas youtube video: https://www.youtube.com/watch?v=voN8omBe2r4

Hands on experience

- Try hierarchical clustering on different datasets, e.g. Iris
  - Some interesting analysis (in R) and datasets can be found at: https://cran.r-project.org/web/packages/dendextend/vignettes/Cluster_Analysis.html
- For the Iris dataset, cut the dendrogram at 3 clusters
  - Is there some mapping between the clusters and the actual species (available as class-labels, not to be used for clustering)?

Thank you

Questions/Feedback/Wishes?

Acknowledgements

- The slides are based on
  - KDD I lecture at LMU Munich (Johannes Aßfalg, Christian Böhm, Karsten Borgwardt, Martin Ester, Eshref Januzaj, Karin Kailing, Peer Kröger, Eirini Ntoutsi, Jörg Sander, Matthias Schubert, Arthur Zimek, Andreas Züfle)
  - Introduction to Data Mining book slides at http://www-users.cs.umn.edu/~kumar/dmbook/
  - Thank you to all TAs contributing to their improvement, namely Vasileios Iosifidis, Damianos Melidis, Tai Le Quy, Han Tran.