Chapter VIII.3: Hierarchical Clustering

1. Basic idea
   1.1. Dendrograms
   1.2. Agglomerative and divisive
2. Cluster distances
   2.1. Single link
   2.2. Complete link
   2.3. Group average and Mean distance
   2.4. Ward's method
3. Discussion

Basic idea

- Create clustering for each number of clusters $k = 1, 2, \ldots, n$
- The clusterings must be **hierarchical**
  - Every cluster of a k-clustering is a union of some clusters in an l-clustering for all $l < k$
  - I.e. for all l, and for all $k > l$, every cluster in an l-clustering is a subset of some cluster in k-clustering
- Example:

Basic idea

- Create clustering for each number of clusters $k = 1, 2, ..., n$
- The clusterings must be **hierarchical**
  - Every cluster of a k-clustering is a union of some clusters in an l-clustering for all $l < k$
  - I.e. for all l, and for all $k > l$, every cluster in an l-clustering is a subset of some cluster in k-clustering
- Example:

Basic idea

- Create clustering for each number of clusters $k = 1, 2, \ldots, n$
- The clusterings must be **hierarchical**
  - Every cluster of a k-clustering is a union of some clusters in an l-clustering for all $l < k$
  - I.e. for all l, and for all $k > l$, every cluster in an l-clustering is a subset of some cluster in k-clustering
- Example:

$k = 6$

Basic idea

- Create clustering for each number of clusters $k = 1, 2, \ldots, n$
- The clusterings must be **hierarchical**
  - Every cluster of a k-clustering is a union of some clusters in an l-clustering for all $l < k$
  - I.e. for all l, and for all $k > l$, every cluster in an l-clustering is a subset of some cluster in k-clustering
- Example:

$k = 5$

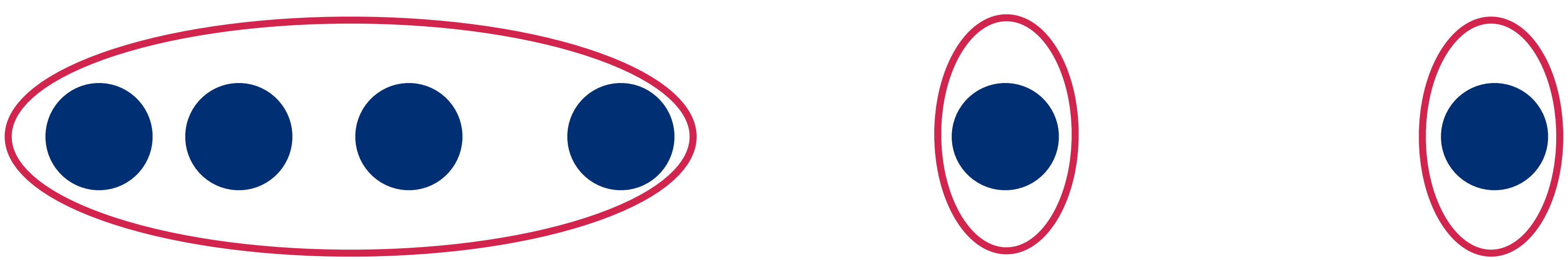
Basic idea

- Create clustering for each number of clusters $k = 1, 2, ..., n$
- The clusterings must be **hierarchical**
  - Every cluster of a k-clustering is a union of some clusters in an l-clustering for all $l < k$
  - I.e. for all l, and for all $k > l$, every cluster in an l-clustering is a subset of some cluster in k-clustering
- Example:

$k = 4$

Basic idea

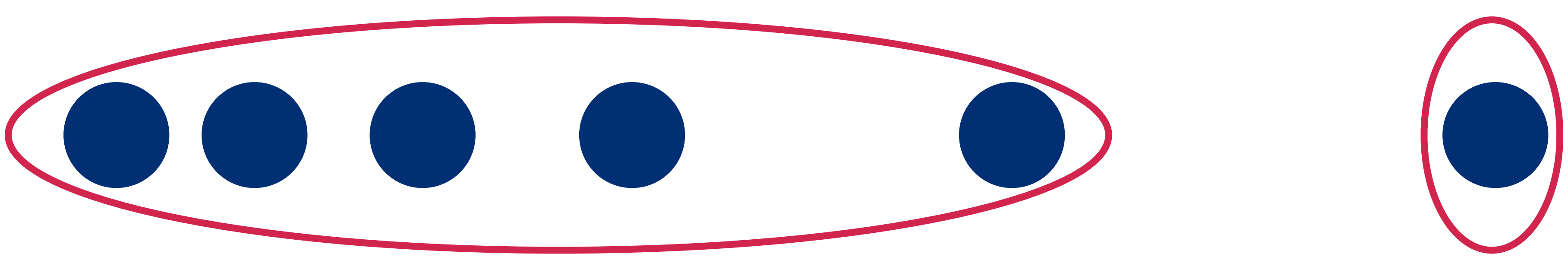
- Create clustering for each number of clusters $k = 1, 2, ..., n$
- The clusterings must be **hierarchical**
  - Every cluster of a k-clustering is a union of some clusters in an l-clustering for all $l < k$
  - I.e. for all l, and for all $k > l$, every cluster in an l-clustering is a subset of some cluster in k-clustering
- Example:



$k = 3$

Basic idea

- Create clustering for each number of clusters $k = 1, 2, ..., n$
- The clusterings must be **hierarchical**
  - Every cluster of a k-clustering is a union of some clusters in an l-clustering for all $l < k$
  - I.e. for all l, and for all $k > l$, every cluster in an l-clustering is a subset of some cluster in k-clustering
- Example:



$k = 2$

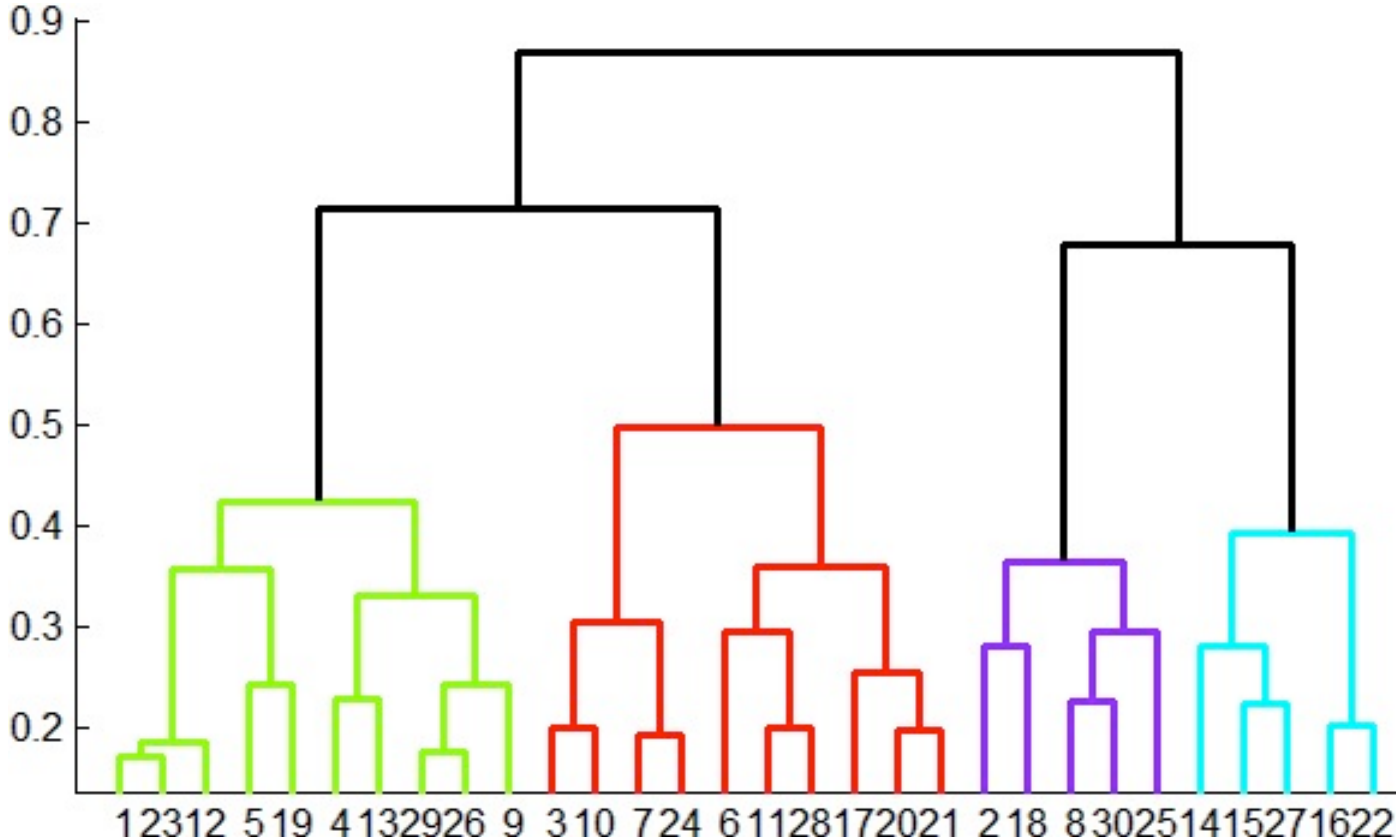
Basic idea

- Create clustering for each number of clusters $k = 1, 2, \ldots, n$
- The clusterings must be **hierarchical**
  - Every cluster of a k-clustering is a union of some clusters in an l-clustering for all $l < k$
  - I.e. for all l, and for all $k > l$, every cluster in an l-clustering is a subset of some cluster in k-clustering
- Example:

$k = 1$

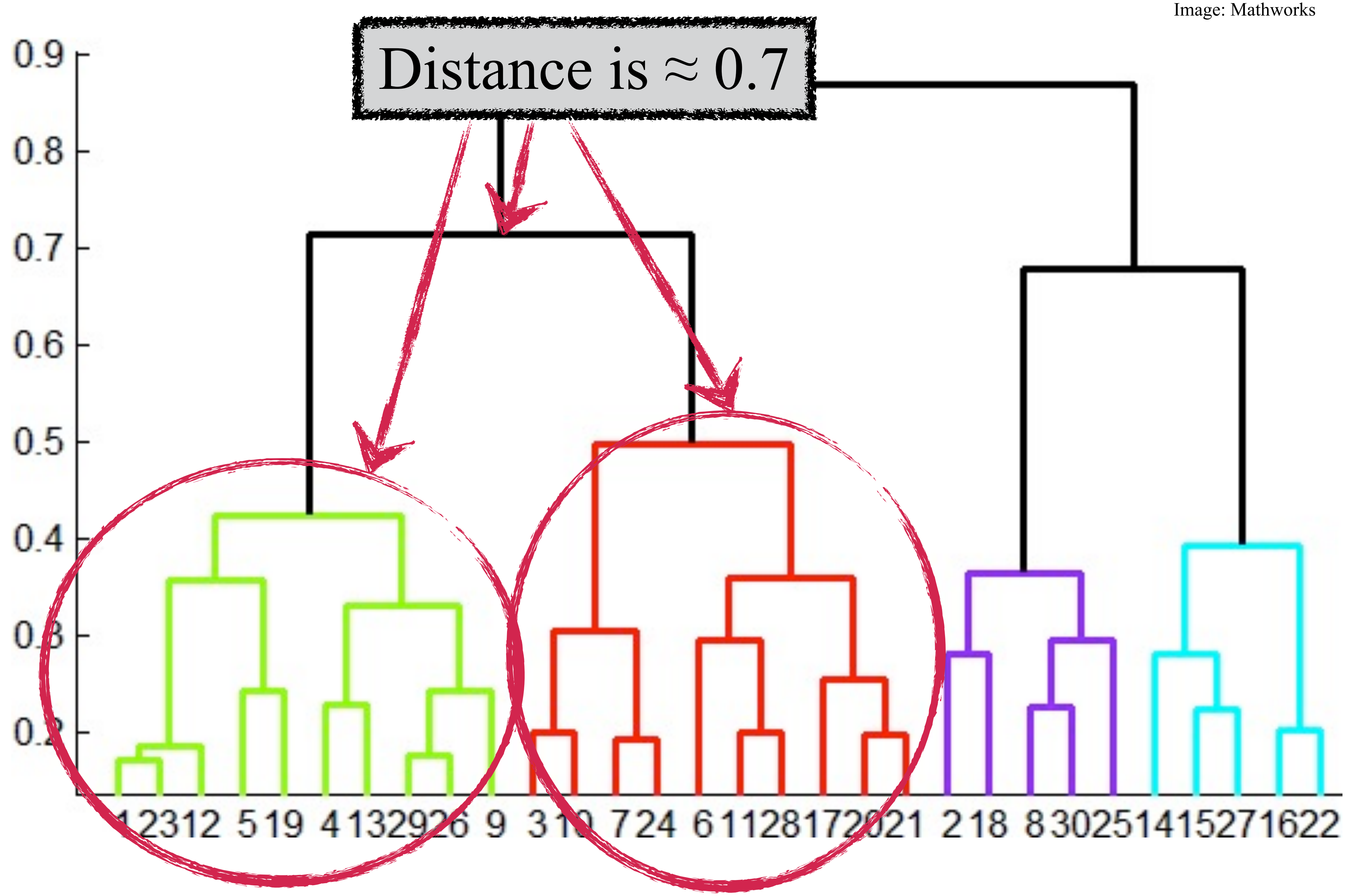
Dendrograms

Image: Mathworks



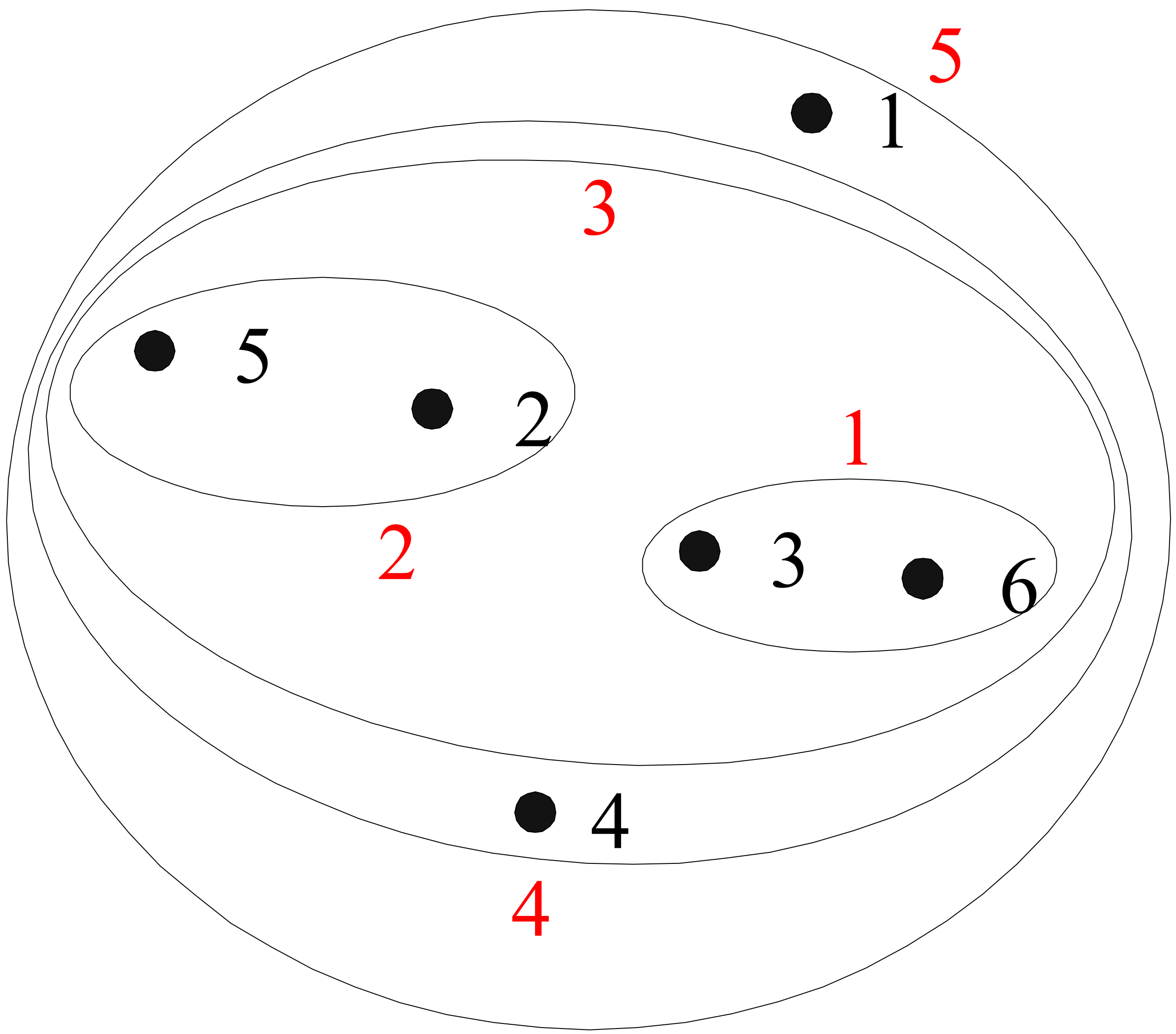
The height of the subtree tree shows the distance between the two branches

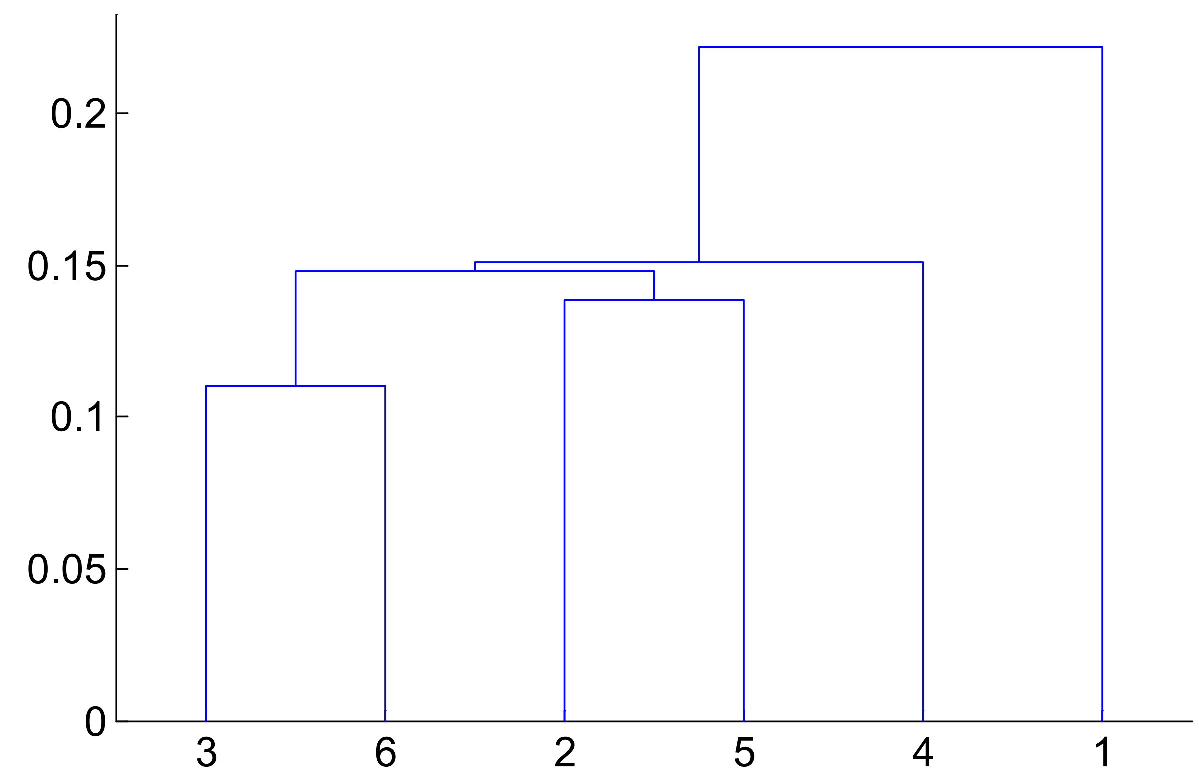
Dendrograms



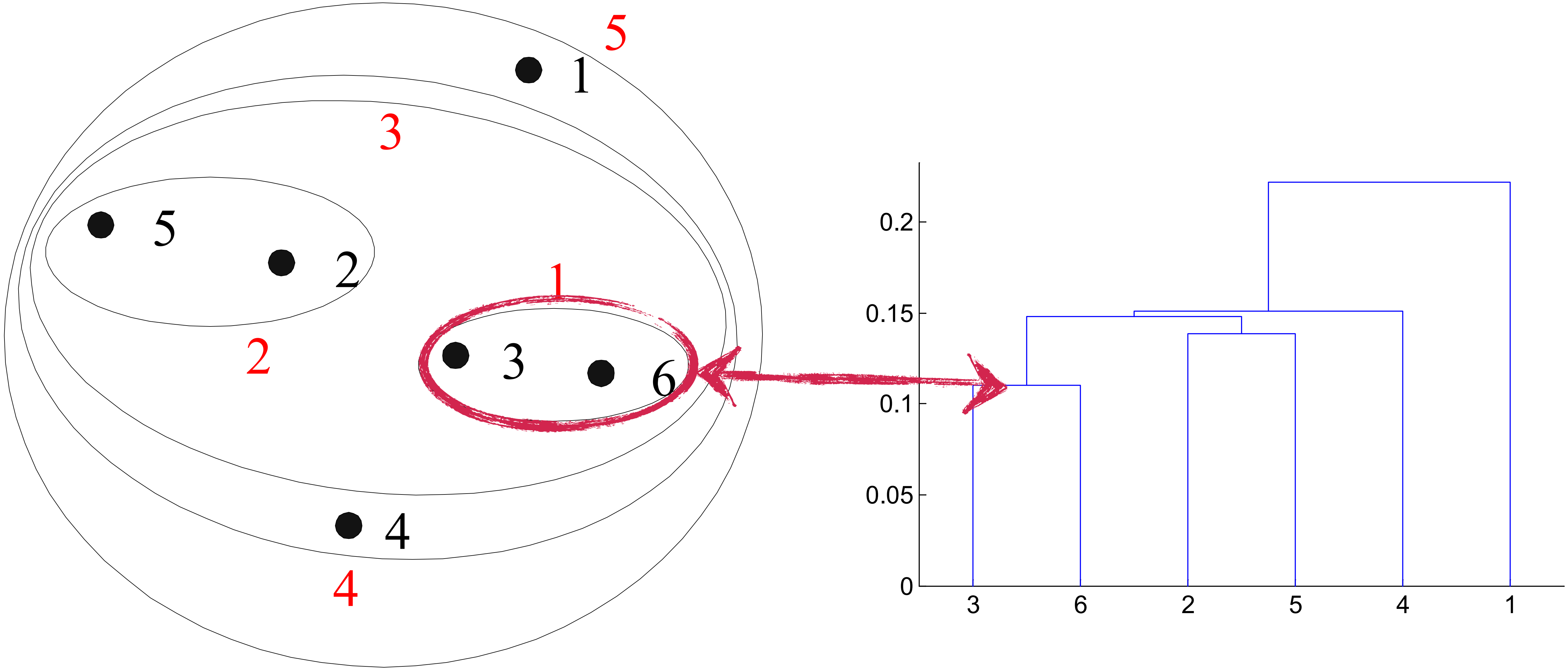
The height of the subtree tree shows the distance between the two branches

Dendrograms and clusters

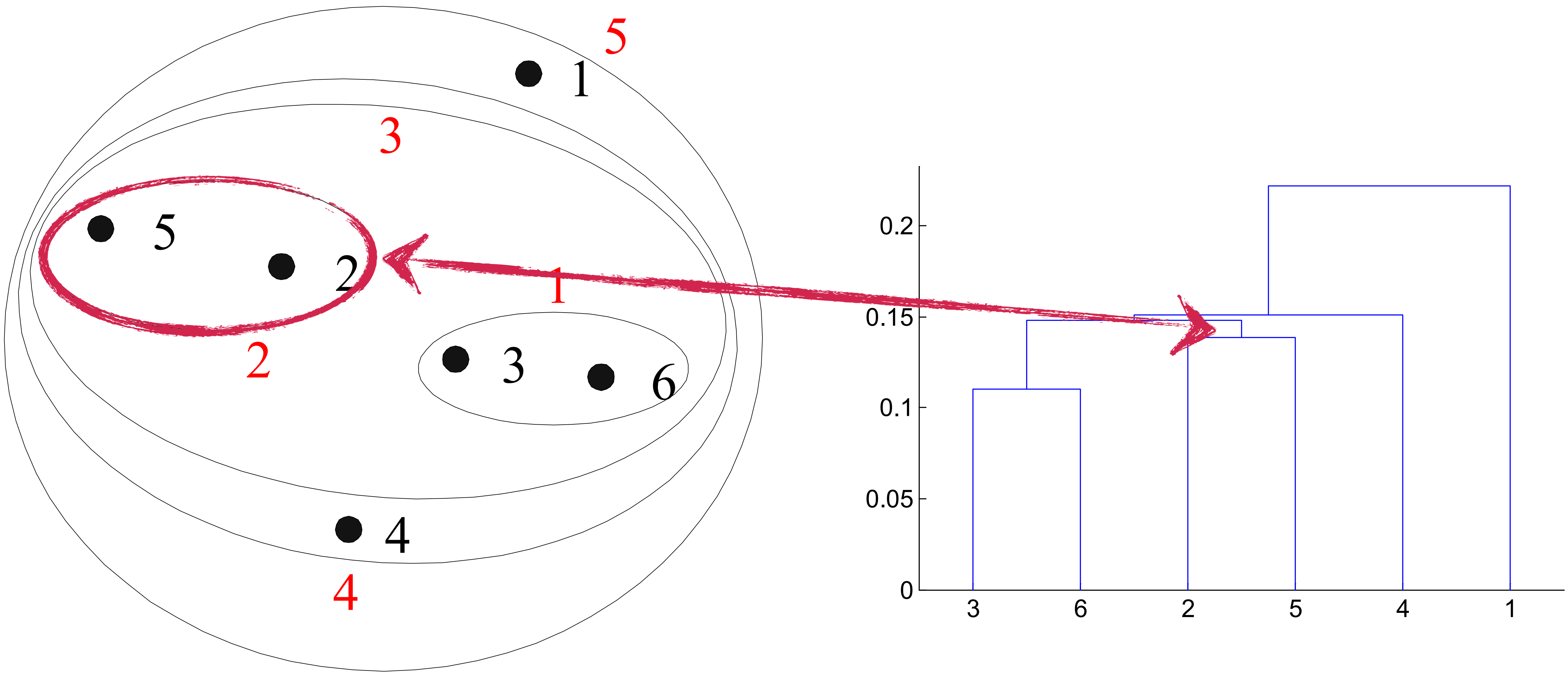




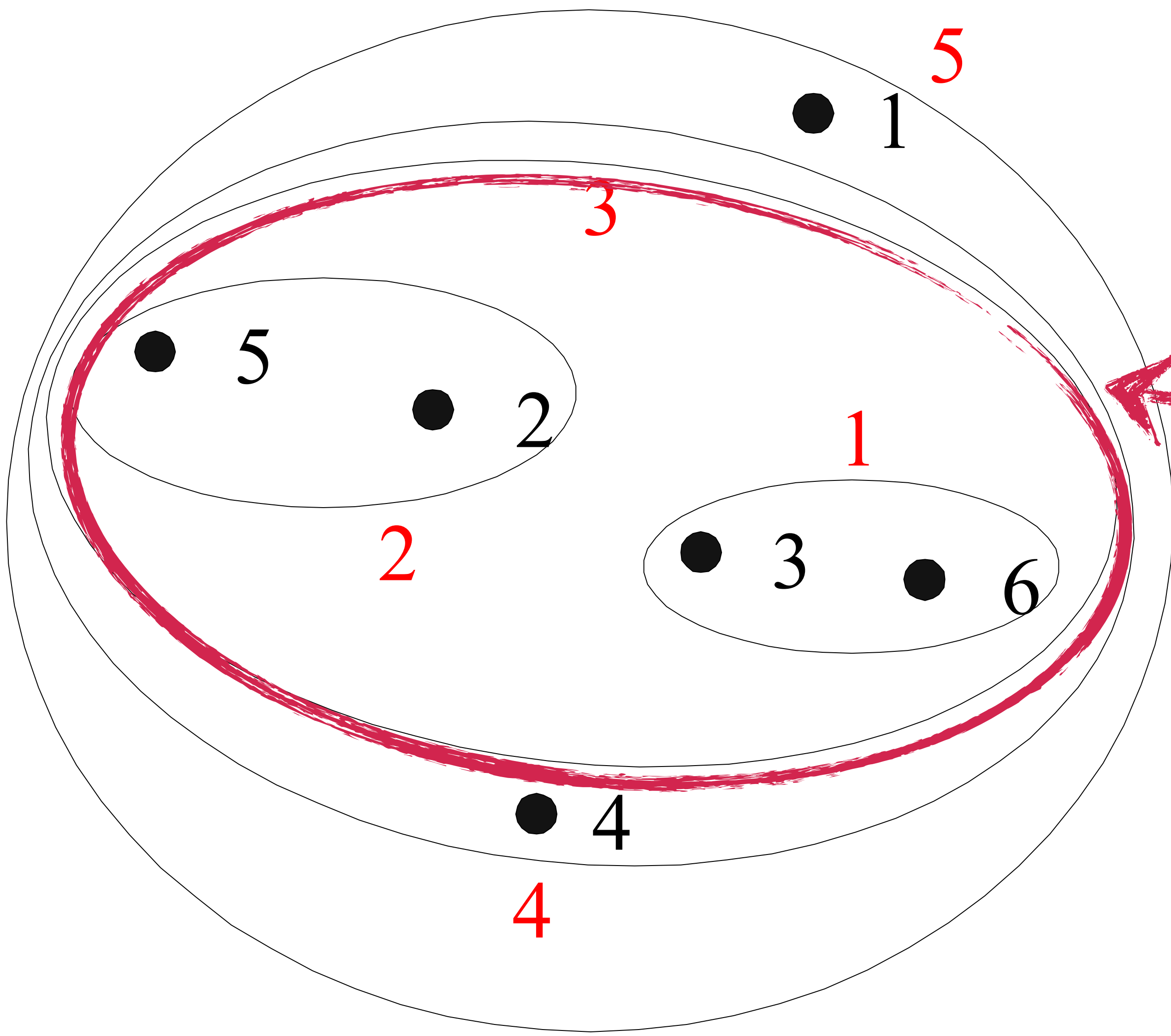
Dendrograms and clusters

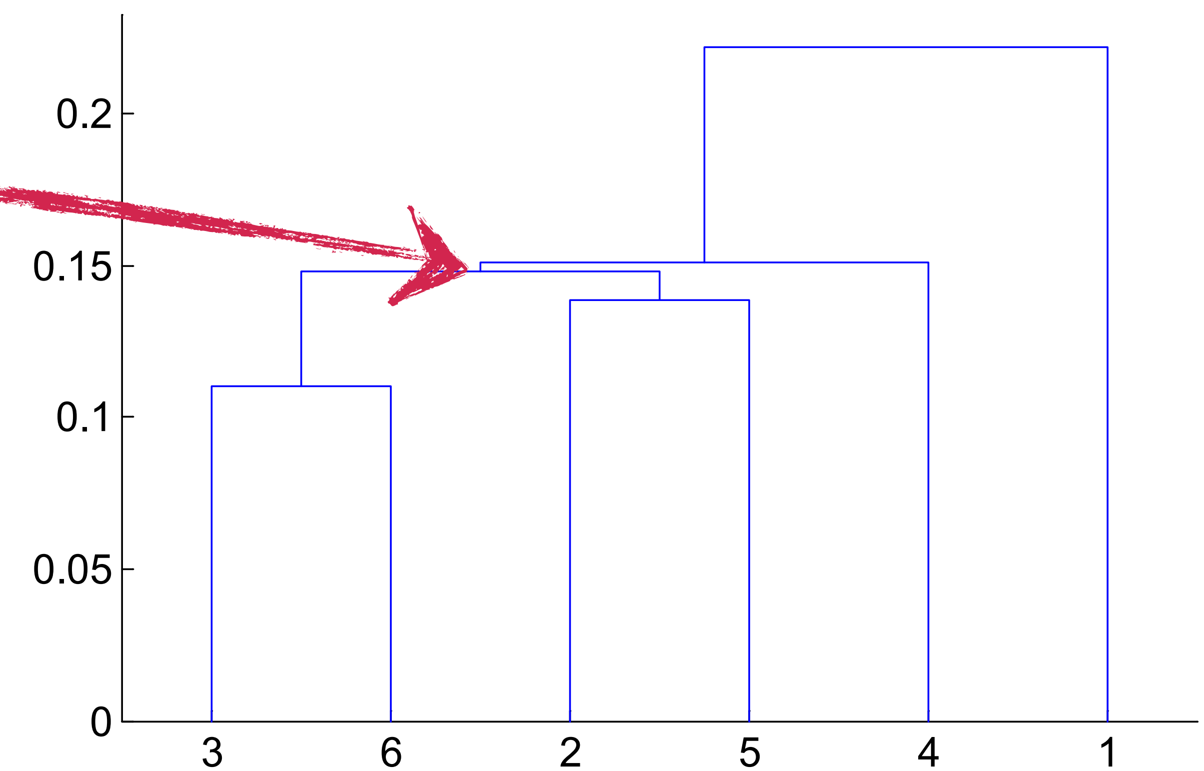


Dendrograms and clusters

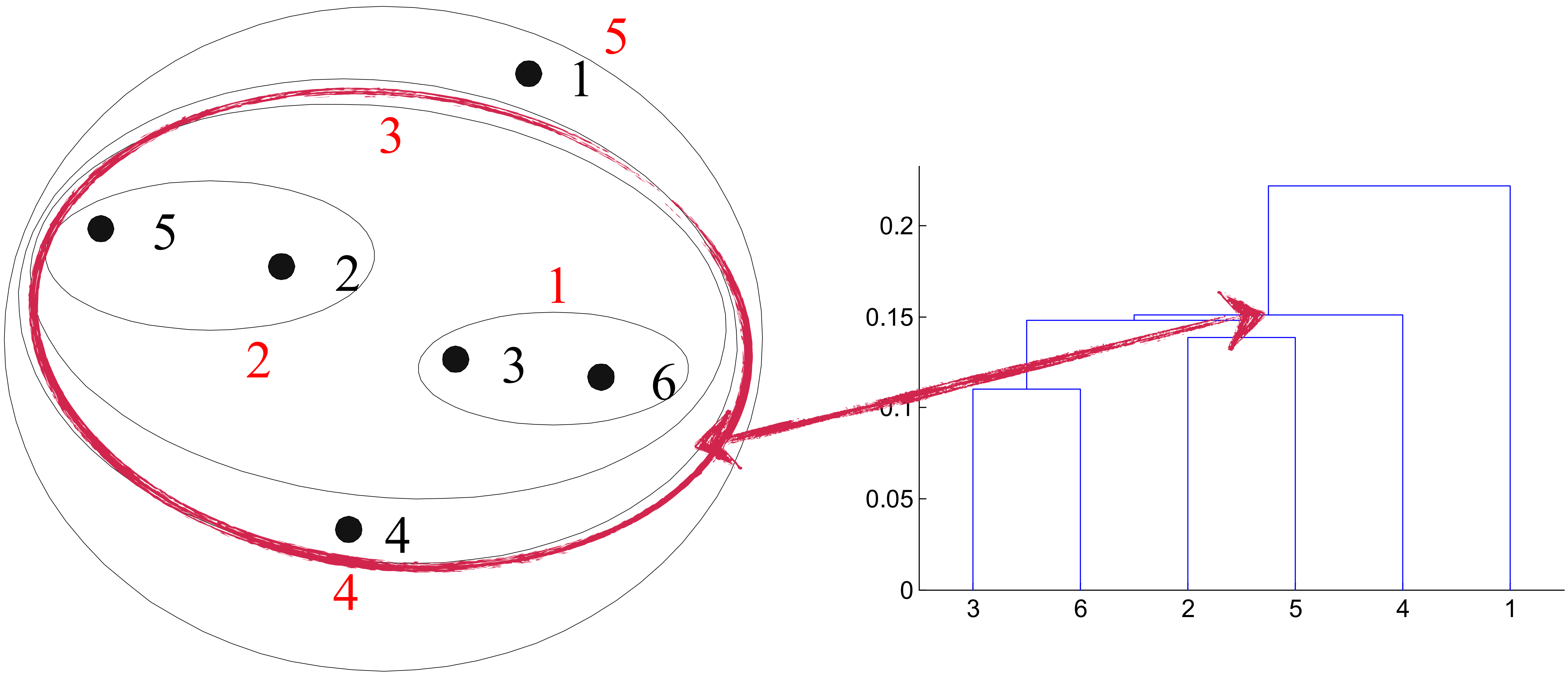


Dendrograms and clusters

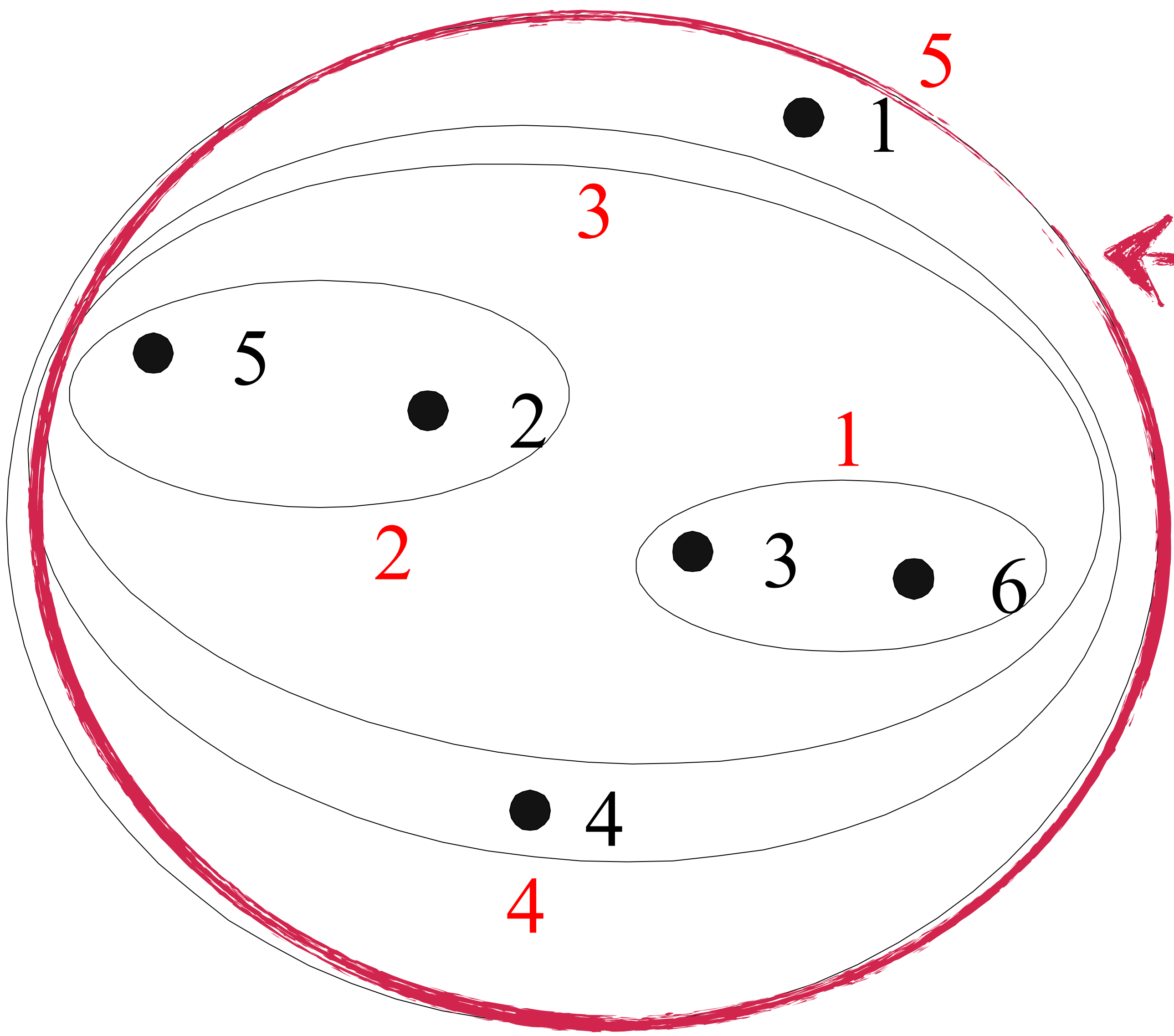


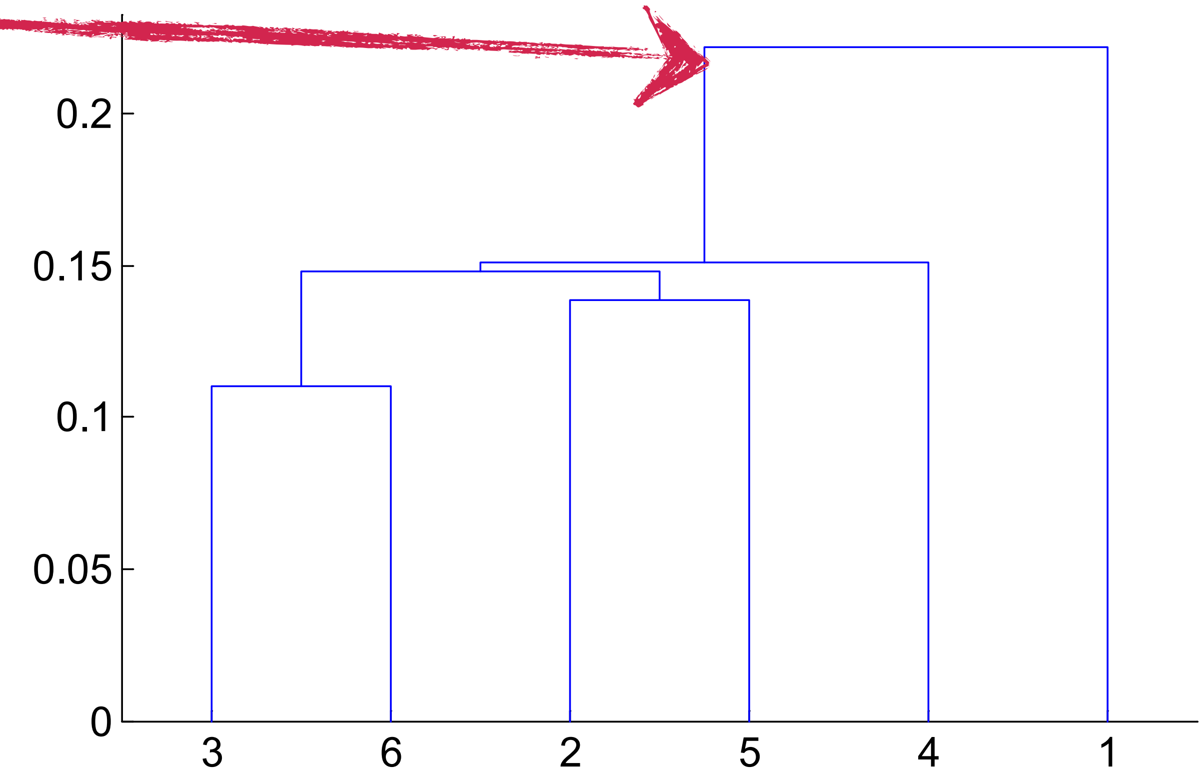


Dendrograms and clusters

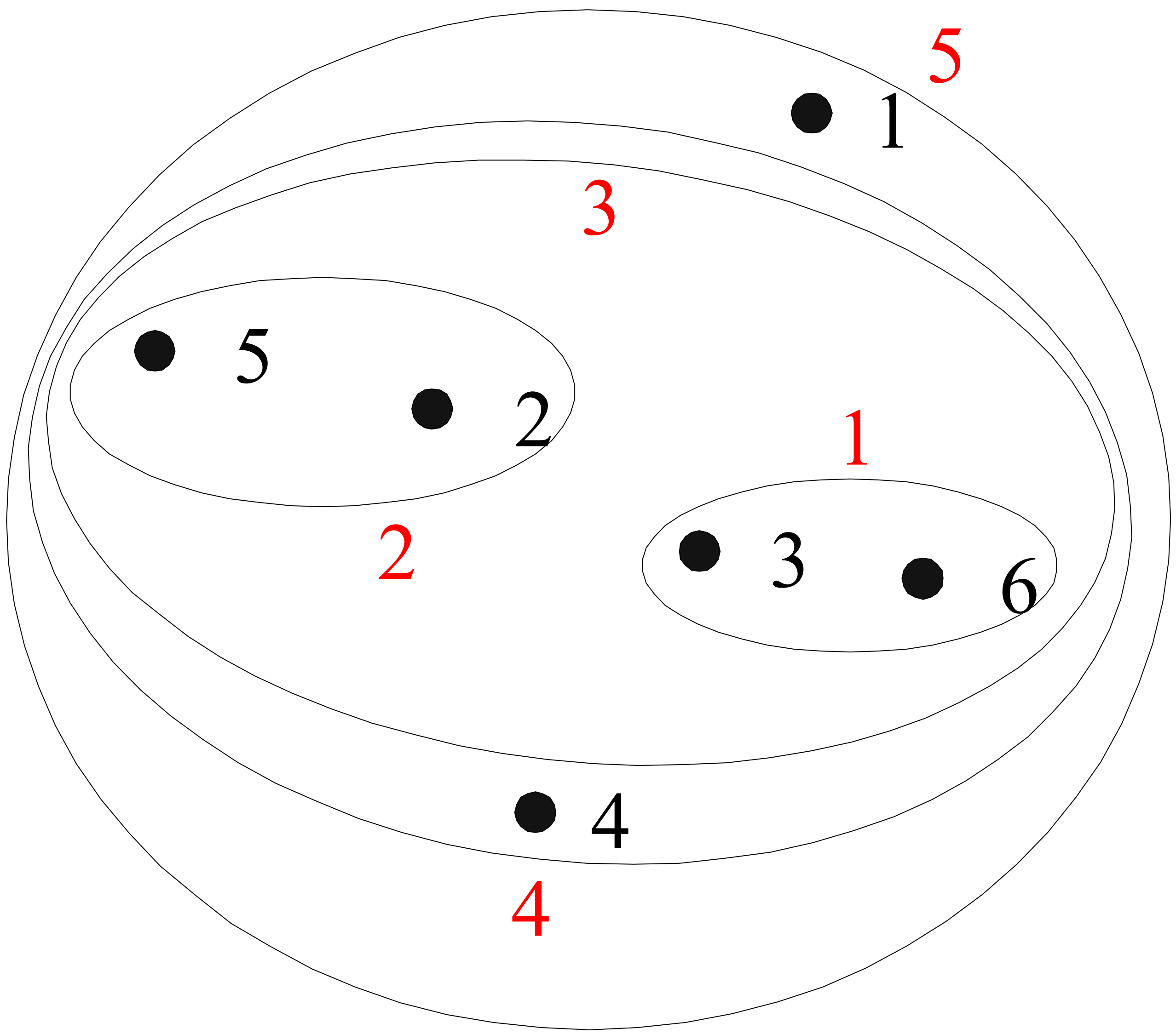


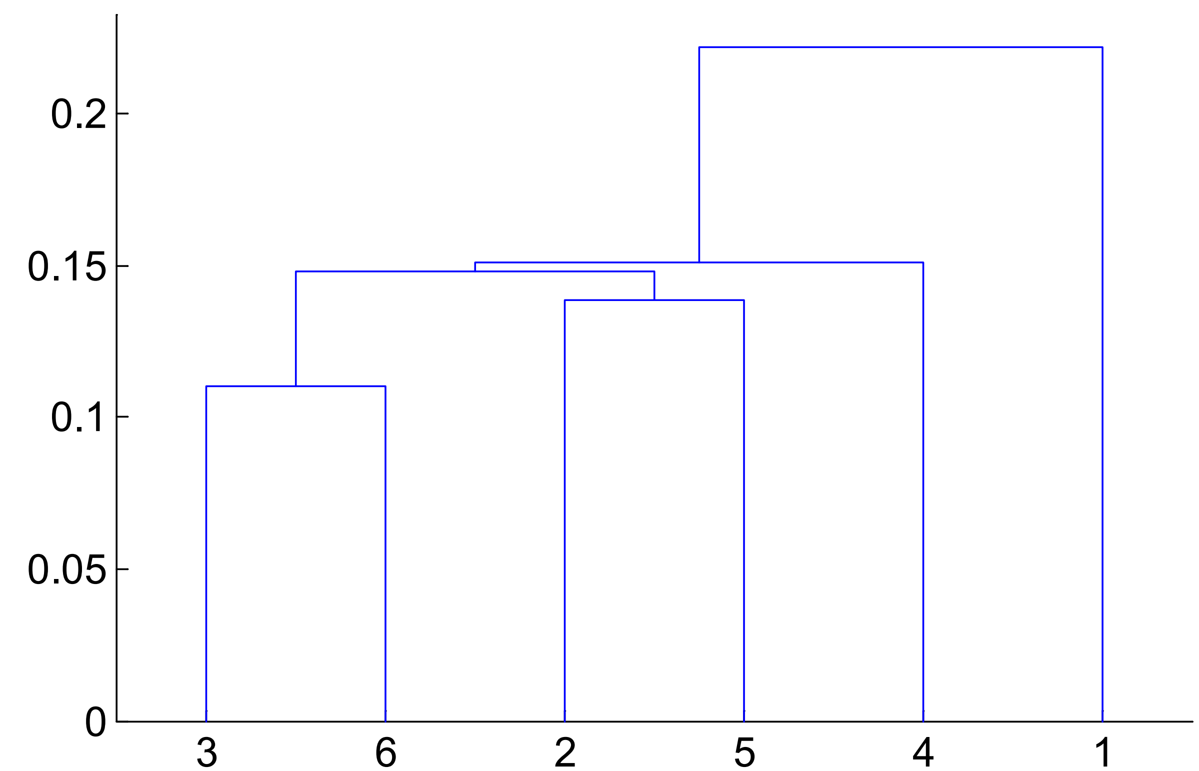
Dendrograms and clusters





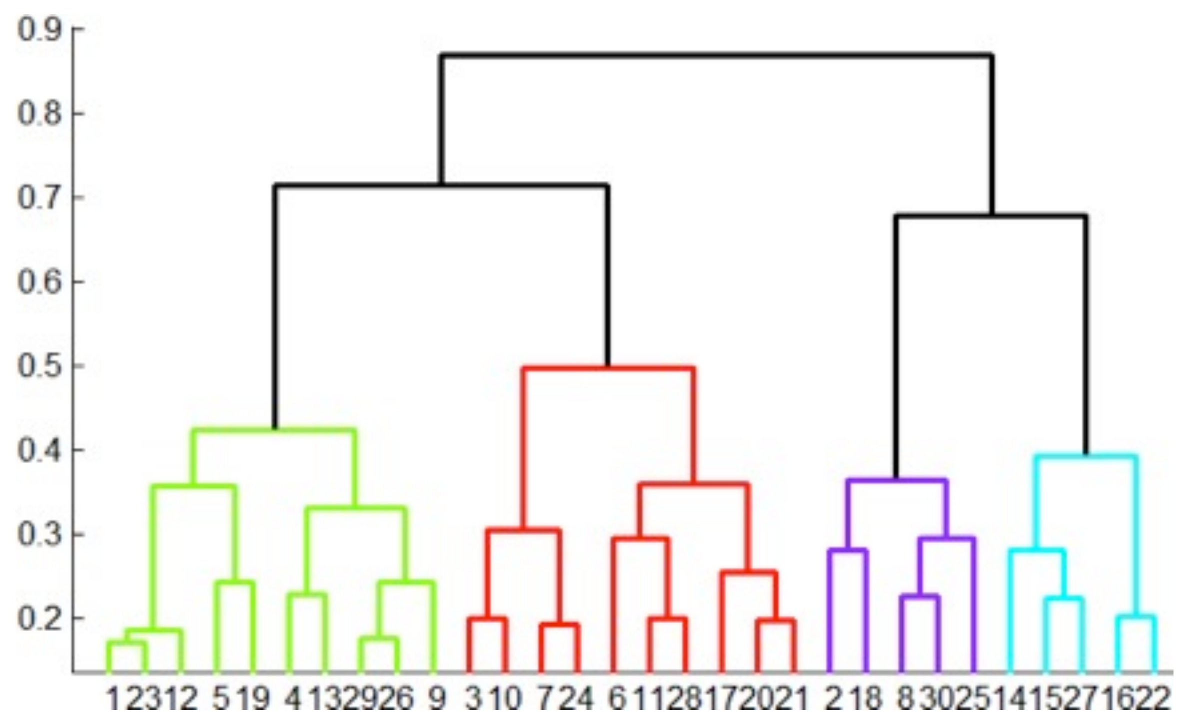
Dendrograms and clusters





Dendrograms

- Dendrograms show the hierarchy of the clustering
- The number of clusters can be deduced from dendrogram
  - Higher branches
- Outliers can be detected from dendrograms
  - Single points that are far from others



Agglomerative and divisive

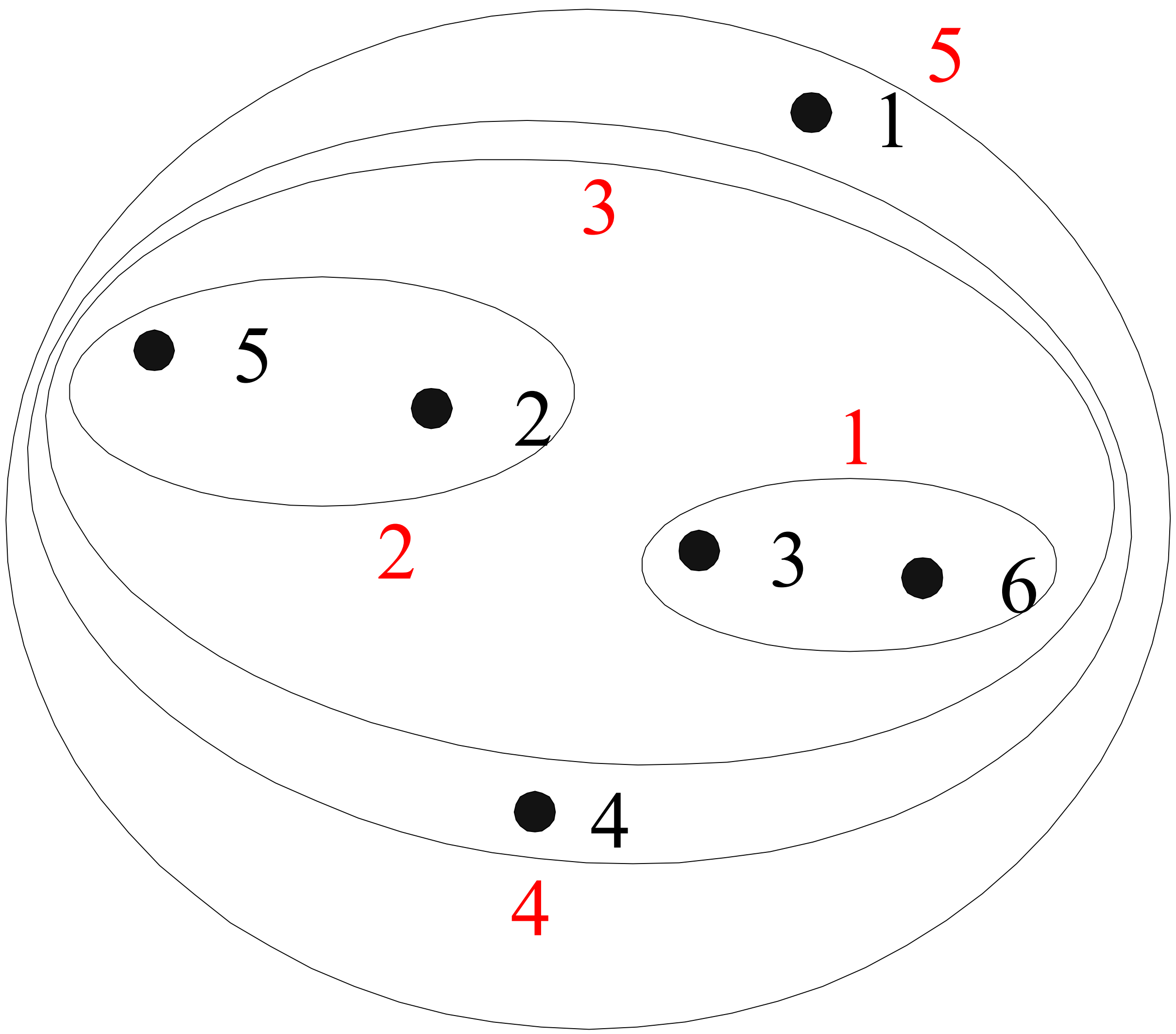
- Agglomerative: bottom-up
  - Start with n clusters
  - Combine two closest points into a cluster of two elements
    - Combine two closest clusters into one bigger cluster
- Divisive: top-down
  - Start with 1 cluster
  - Divide the cluster into two
    - Divide the largest (per diameter) cluster into two smaller

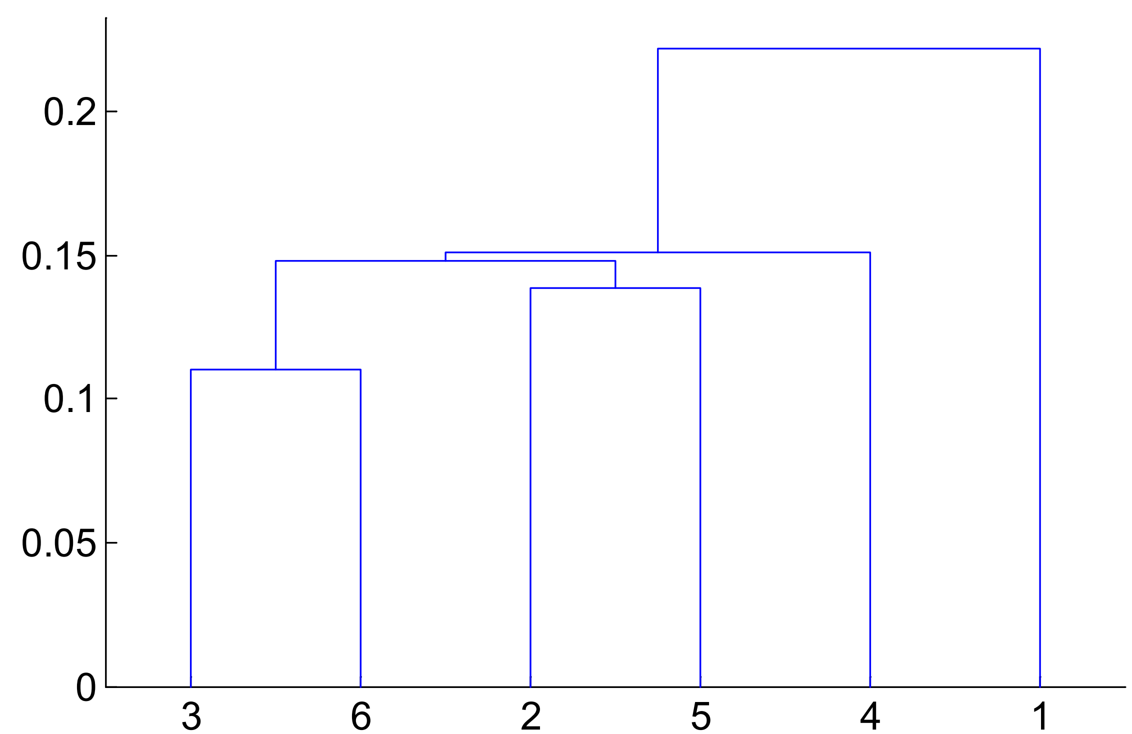
Cluster distances

- The distance between two points x and y is $d(x,y)$
- But what is the distance between two clusters?
- Many intuitive definitions – no universal truth
  - Different cluster distances yield different clusterings
  - The selection of cluster distance depends on application
- Some distances between clusters B and C:
  - minimum distance $d(B,C) = \min\{d(x,y) : x \in B \text{ and } y \in C\}$
  - maximum distance $d(B,C) = \max\{d(x,y) : x \in B \text{ and } y \in C\}$
  - average distance $d(B,C) = \operatorname{avg}\{d(x,y) : x \in B \text{ and } y \in C\}$
  - distance of centroids $d(B,C) = d(\mu_B, \mu_C)$,
    where μ_B is the centroid of B and μ_C is the centroid of C

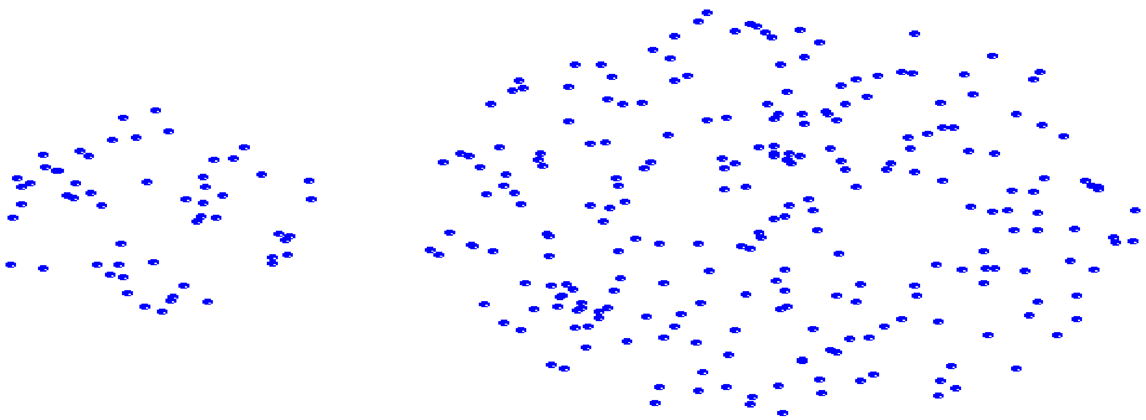
Single link

- The distance between two clusters is the distance between the closest points
  - $d(B,C) = \min\{d(x,y) : x \in B \text{ and } y \in C\}$

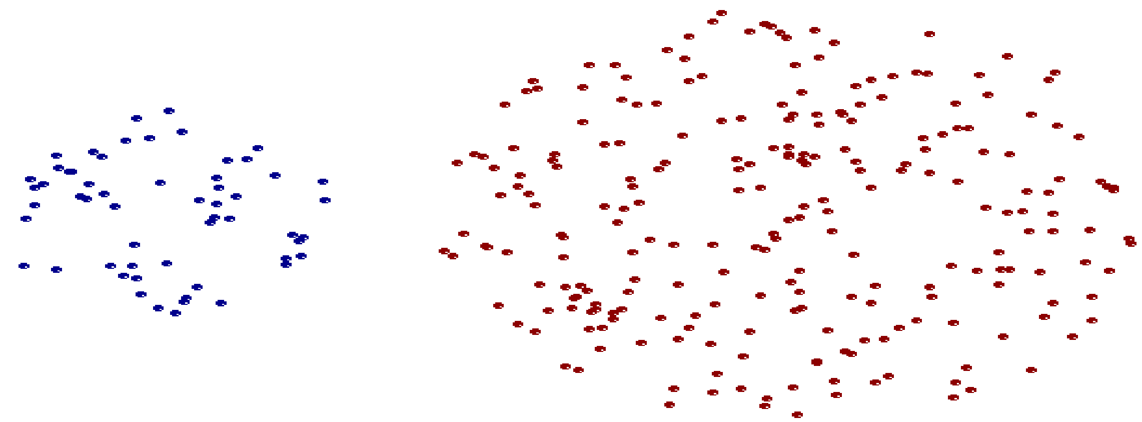




Strengths of single-link



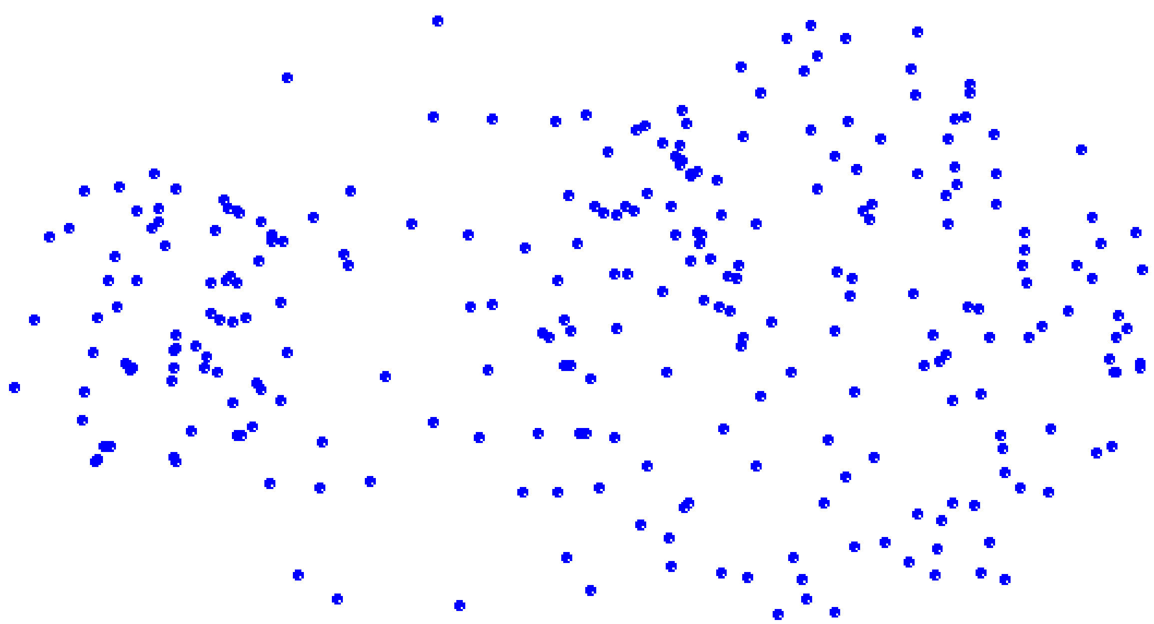
Original Points



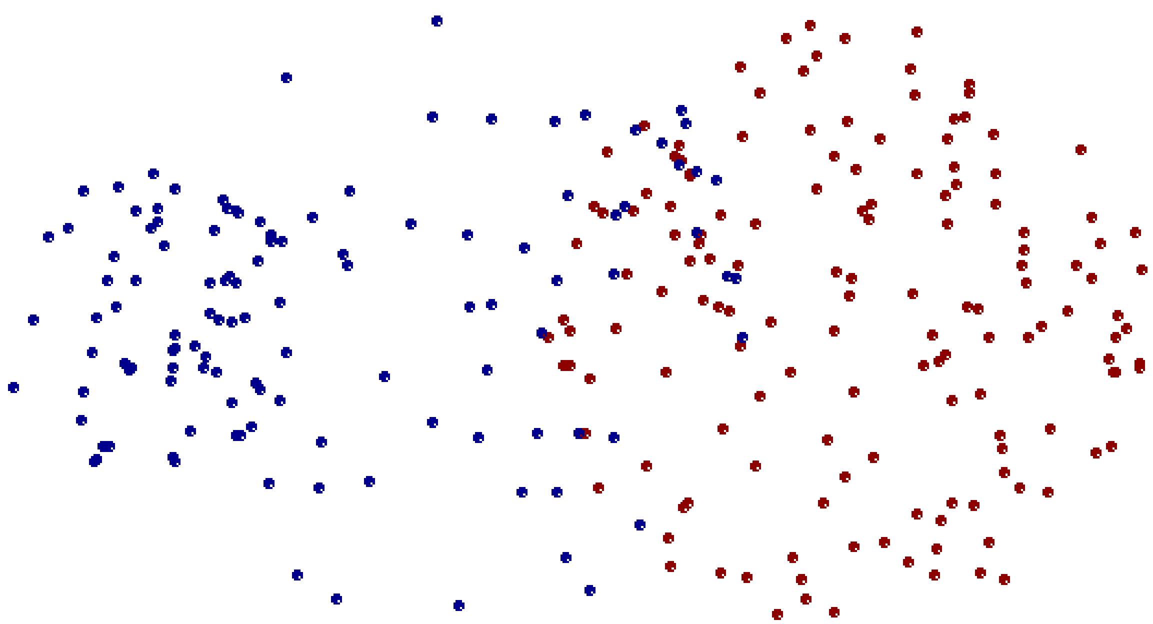
Two Clusters

Can handle non-spherical clusters of unequal size

Weaknesses of single-link



Original Points

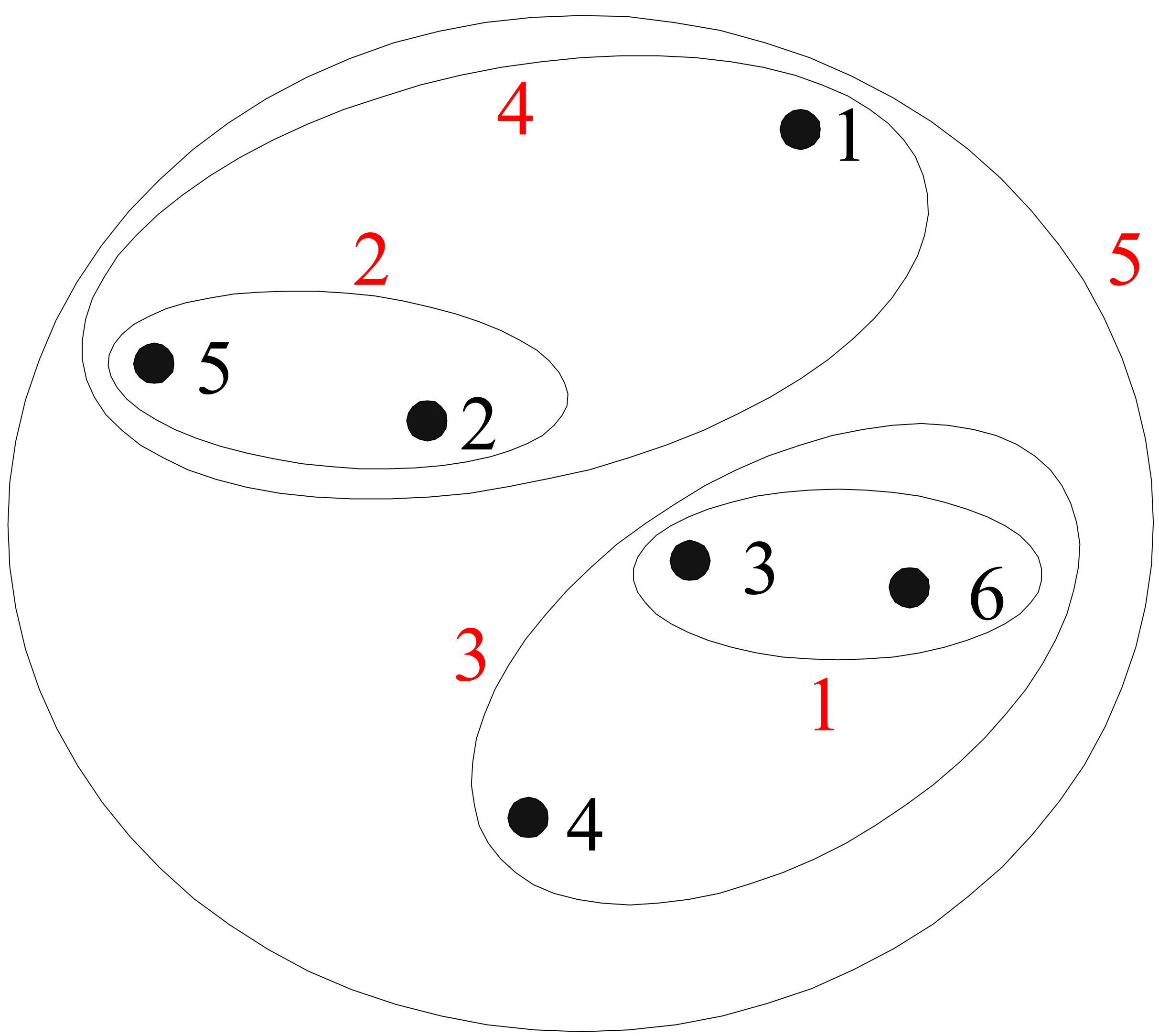


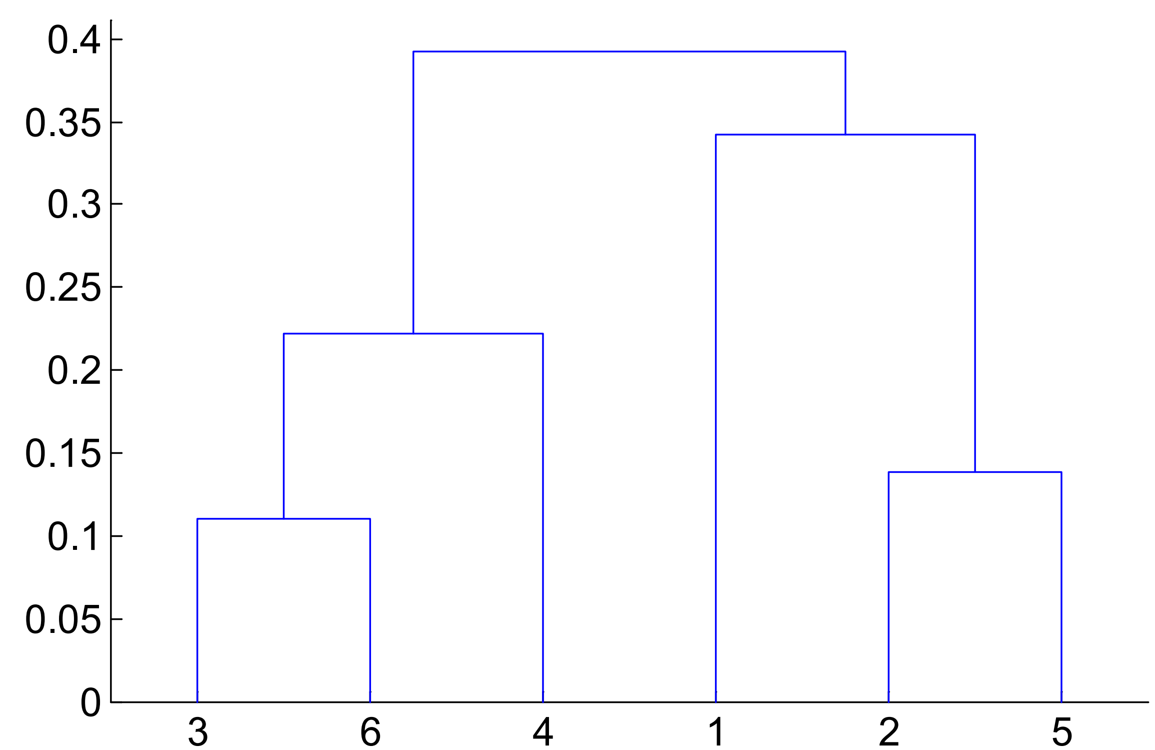
Two Clusters

- Sensitive to noise and outliers
- Produces elongated clusters

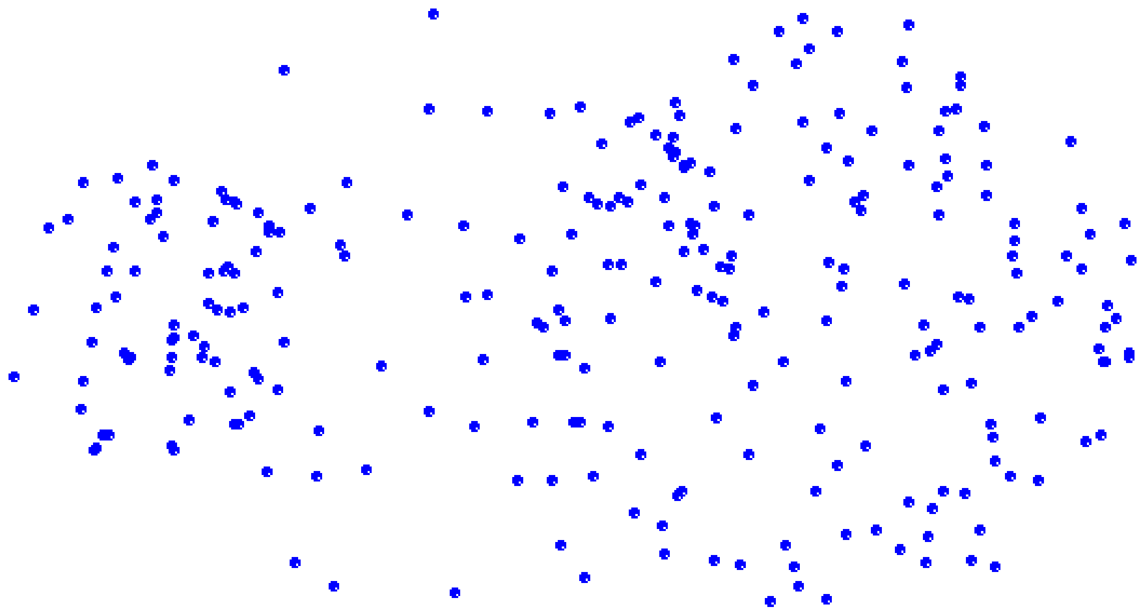
Complete link

- The distance between the clusters is the distance between the furthest points
  - $d(B,C) = \max\{d(x,y) : x \in B \text{ and } y \in C\}$

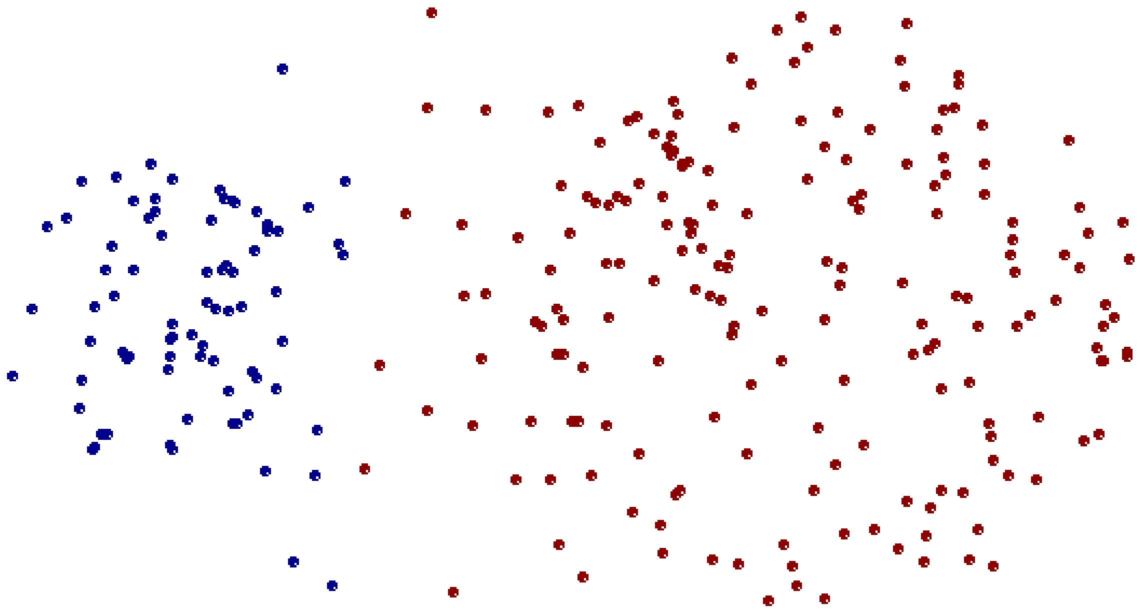




Strengths of complete link



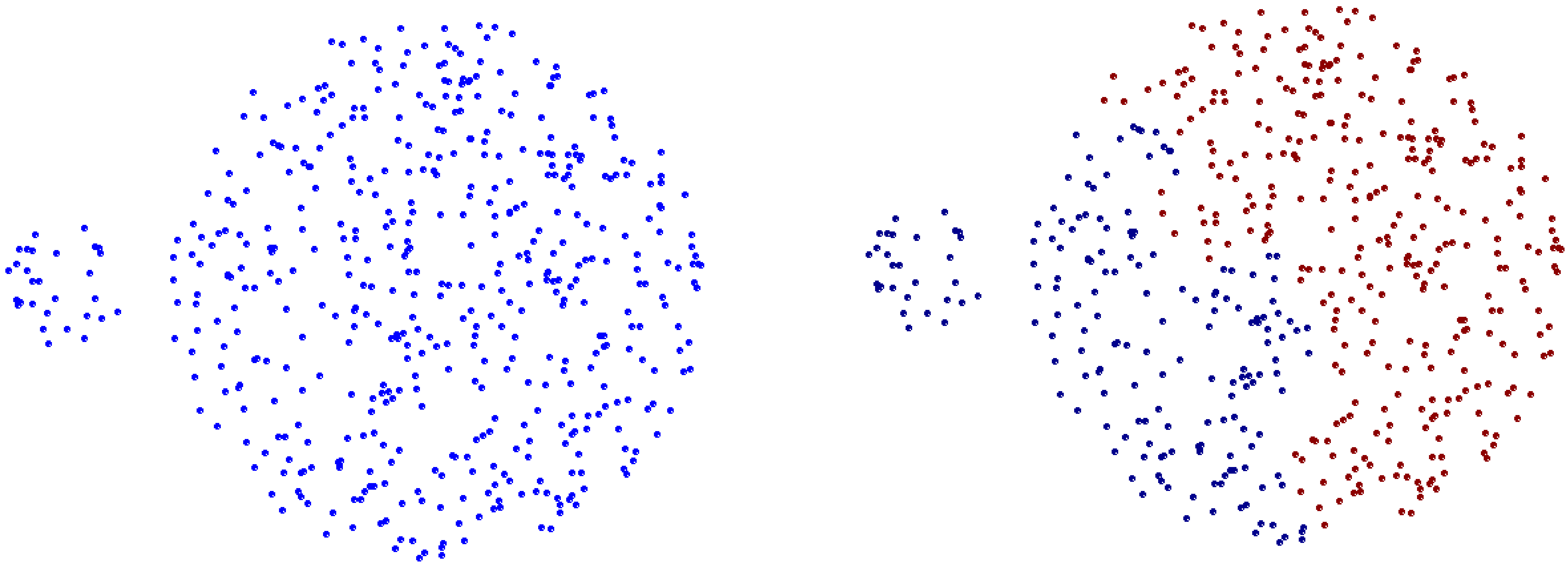
Original Points



Two Clusters

- Less susceptible to noise and outliers

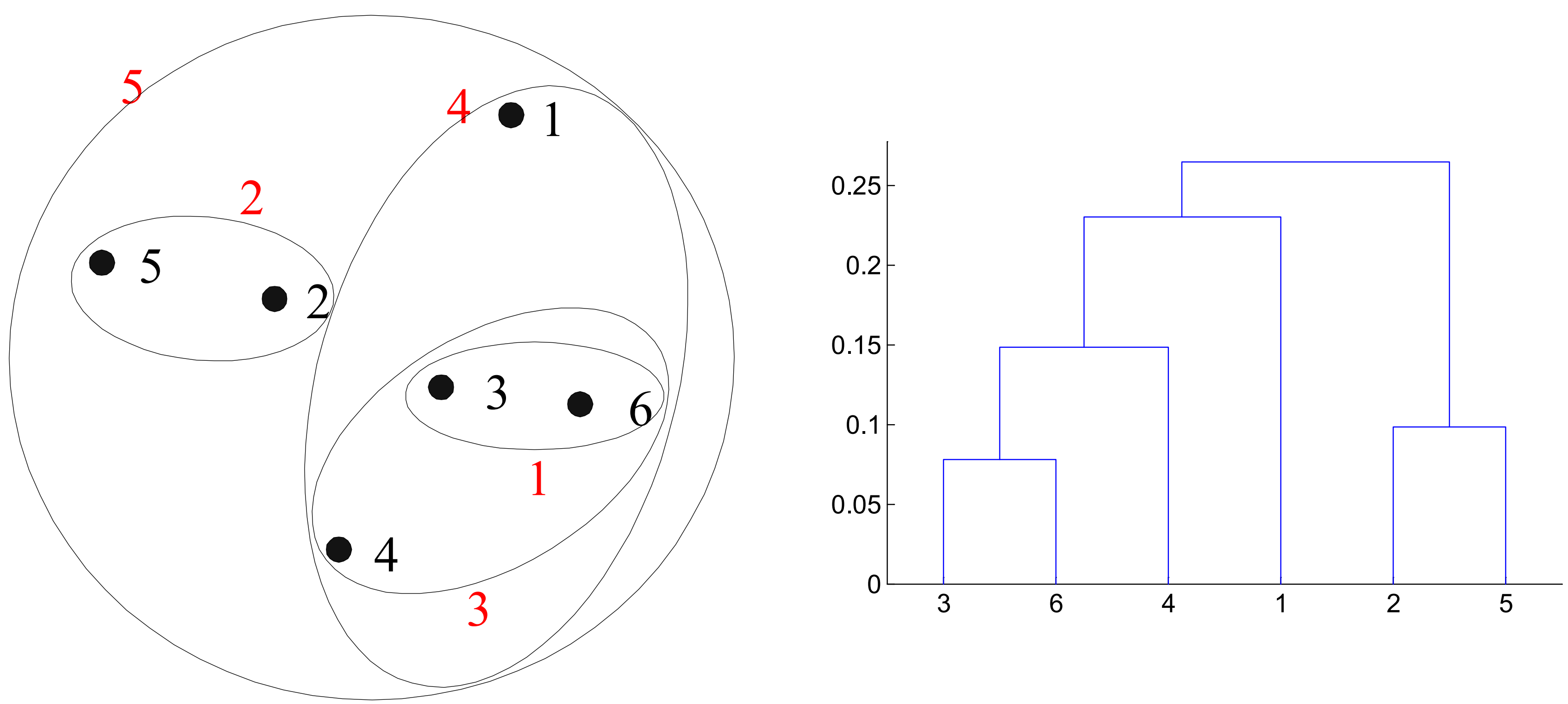
Weaknesses of complete link



- Breaks largest clusters
- Biased towards spherical clusters

Group average and Mean distance

- *Group average* is the average of pairwise distances
  - $d(B,C) = \text{avg}\{d(x,y) : x \in B \text{ and } y \in C\}$
    $= \sum_{x \in B, y \in C} d(x,y)/(|B|\,|C|)$
- *Mean distance* is the distance of the cluster centroids
  - $d(B,C) = d(\mu_B, \mu_C)$



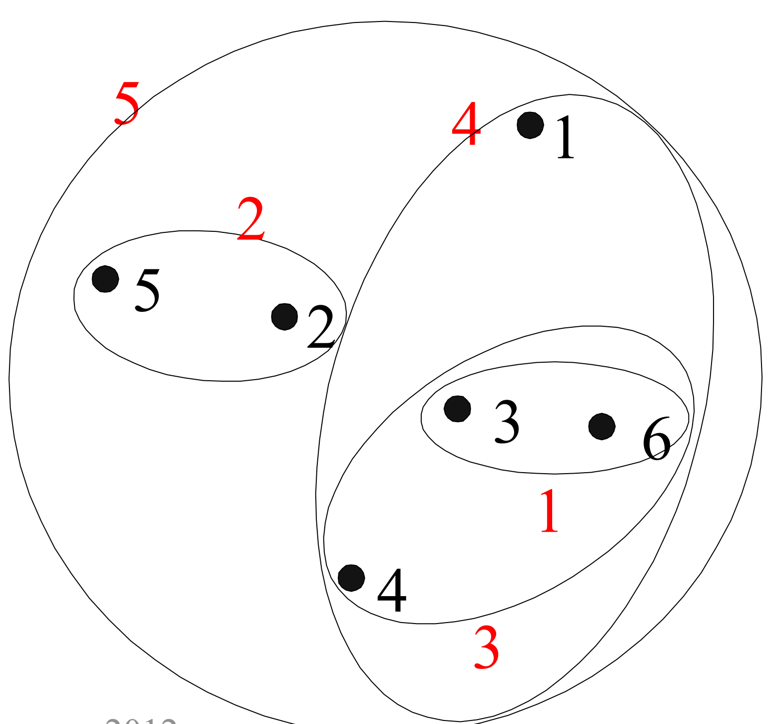
Group average

Properties of group average

- A compromise between single and complete link
- Less susceptible to noise and outliers
  - Similar to complete link
- Biased towards spherical clusters
  - Similar to complete link

Ward's method

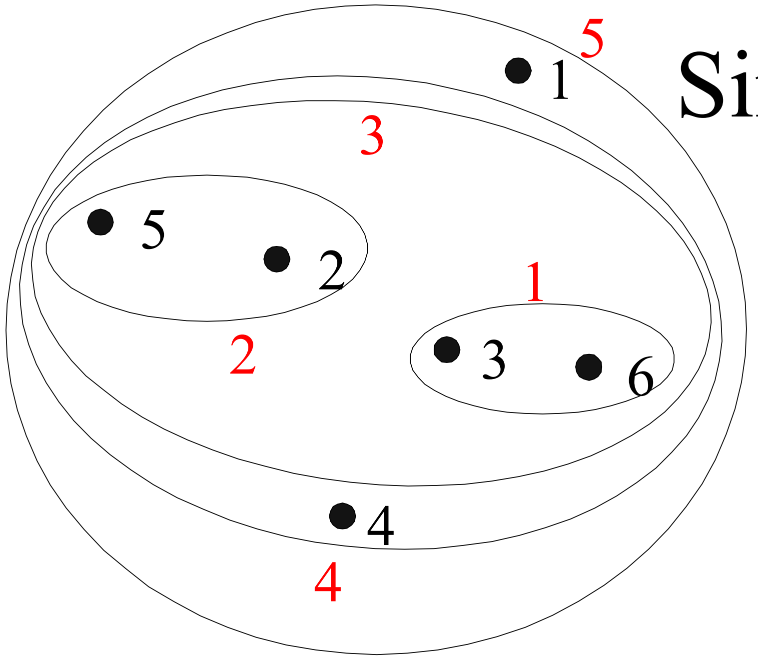
- **Ward's distance** between clusters A and B is the increase in sum of squared errors (SSE) when the two clusters are merged
  - SSE for cluster A is $\mathrm{SSE}_A = \sum_{x \in A} ||x - \mu_A||^2$
  - Difference on merging clusters A and B to cluster C is then $d(A, B) = \Delta\mathrm{SSE}_C = \mathrm{SSE}_C - \mathrm{SSE}_A - \mathrm{SSE}_B$
  - Equivalently, $d(A,B) = |A||B|/(|A|+|B|)||\mu_A - \mu_B||^2$
    - Weighted mean distance



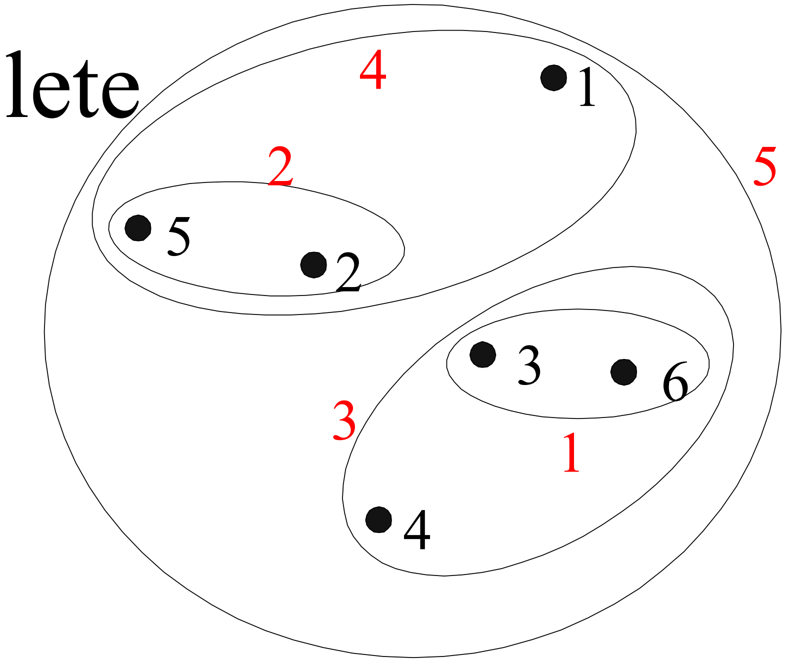
Discussion on Ward's method

- Less susceptible to noise and outliers
- Biased towards spherical clusters
- Hierarchical analogue of *k*-means
  - Hence many shared pros and cons
  - Can be used to initialize *k*-means

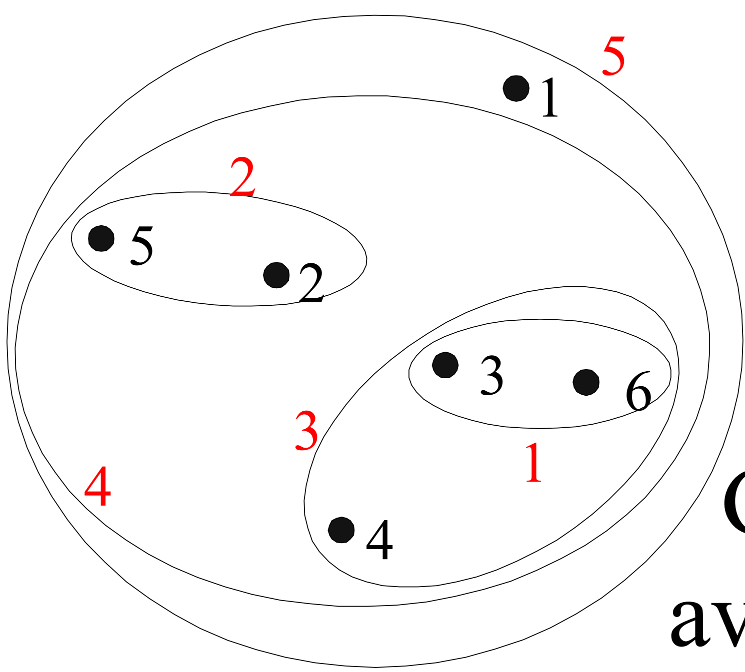
Comparison



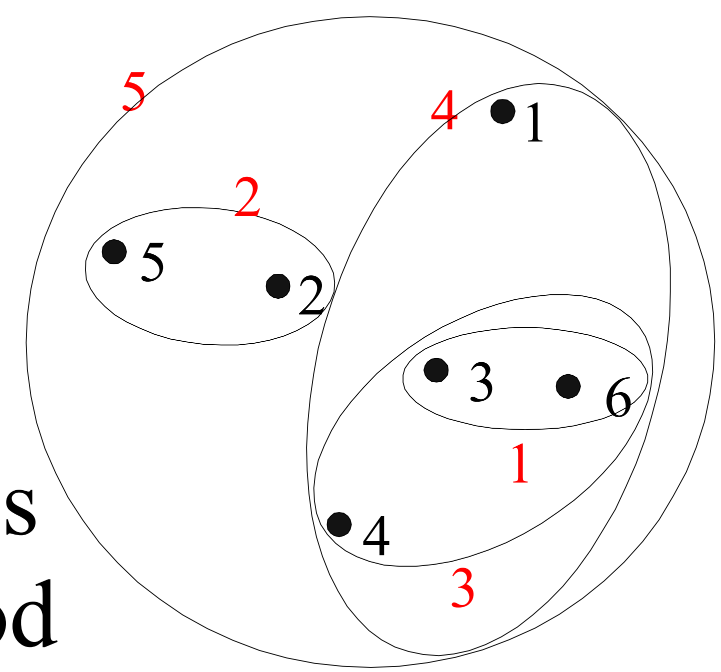
Single link



Complete link

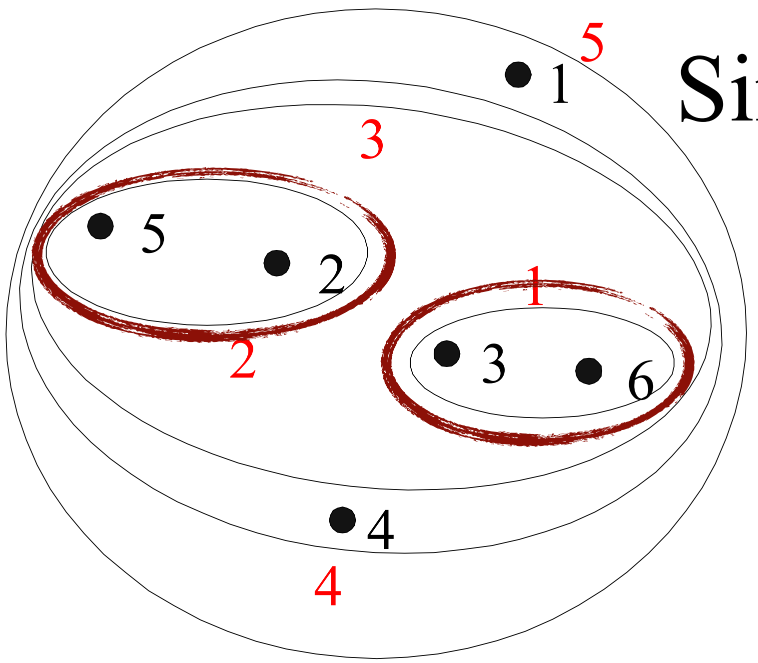


Group average



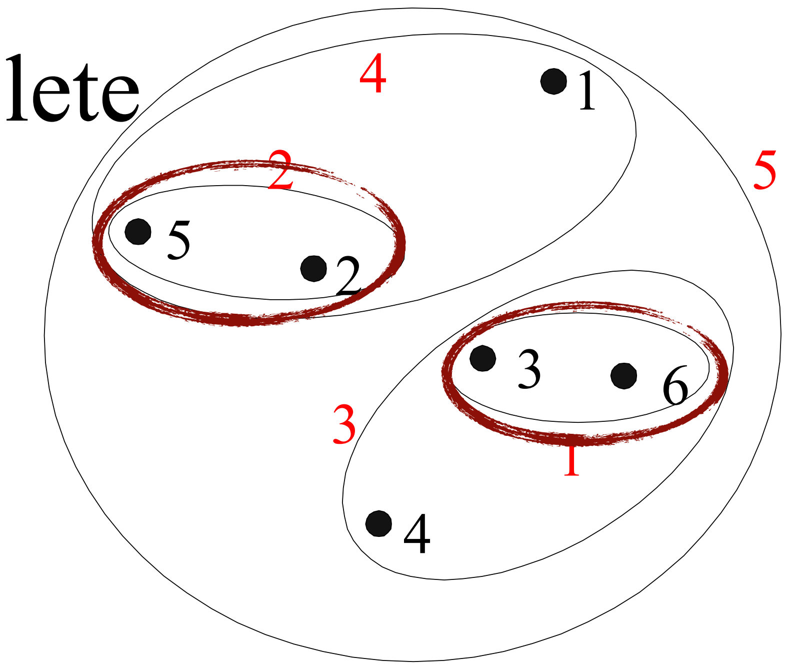
Ward's method

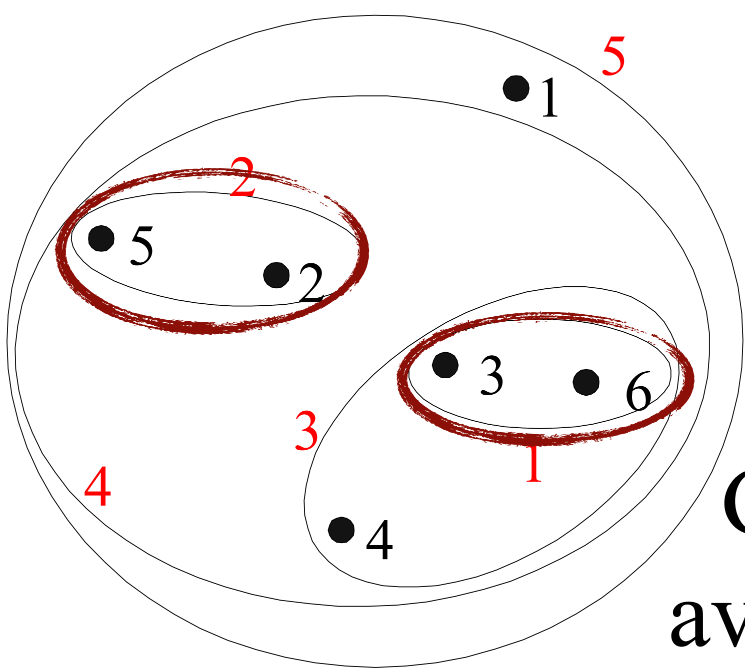
Comparison



Single link

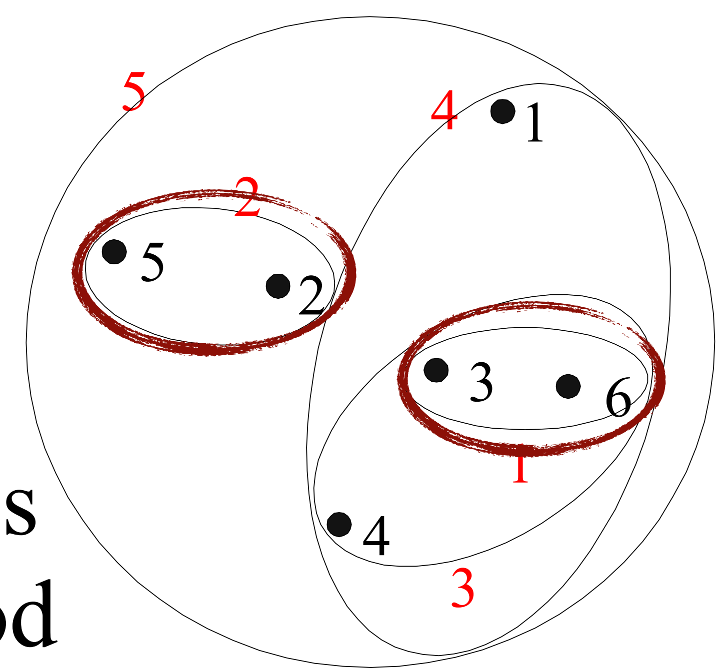
Complete link



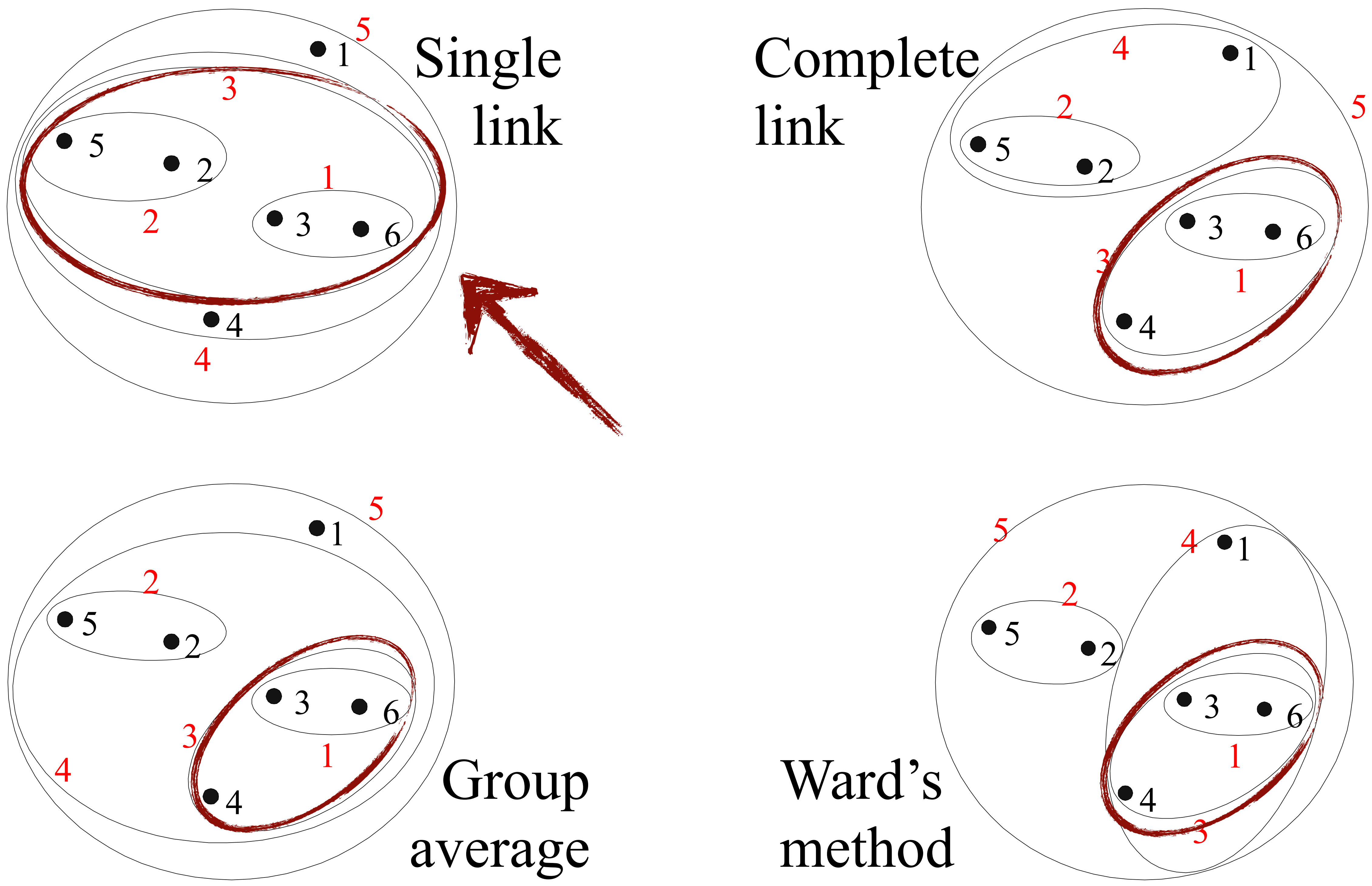


Group average

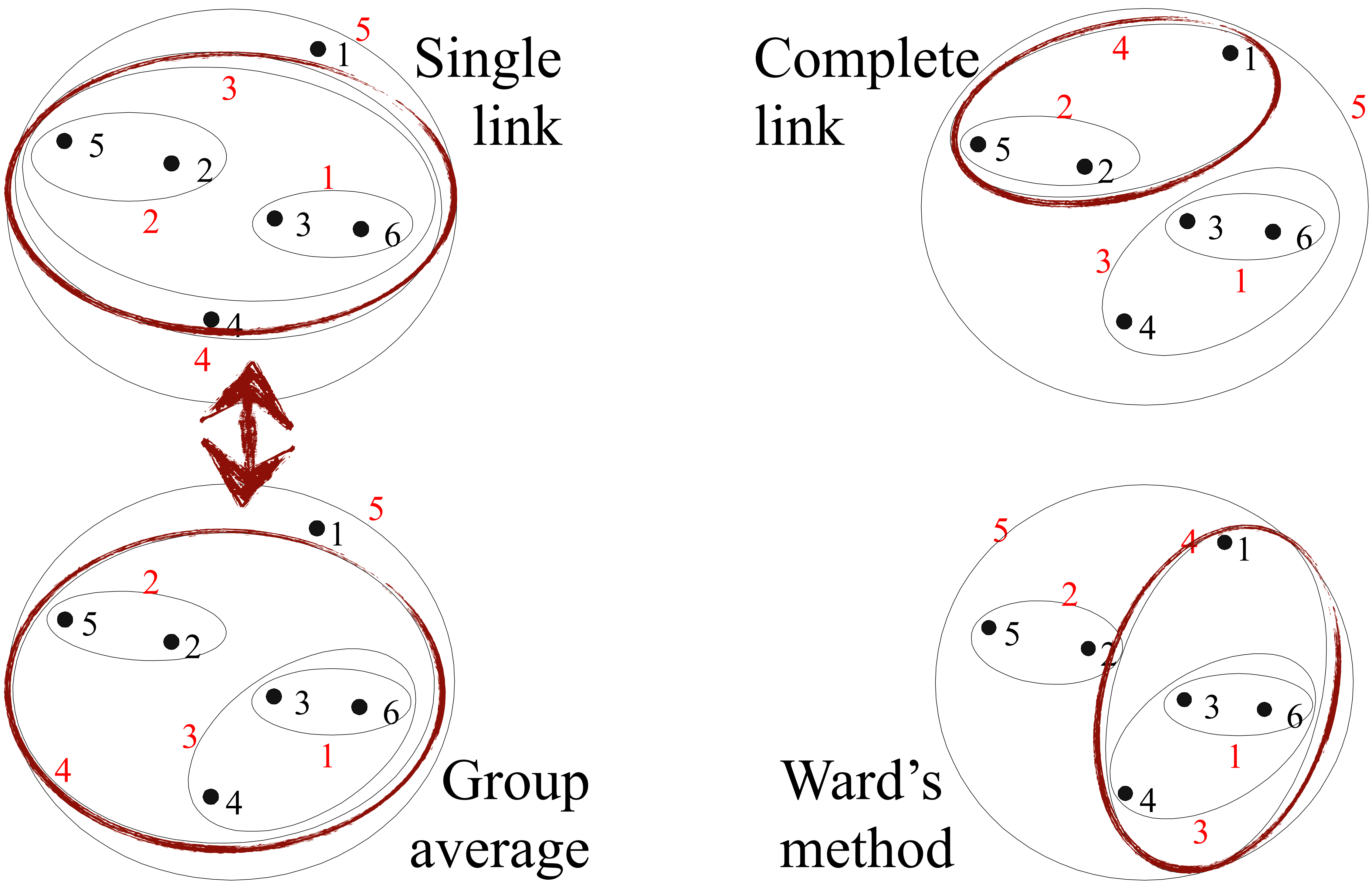
Ward's method



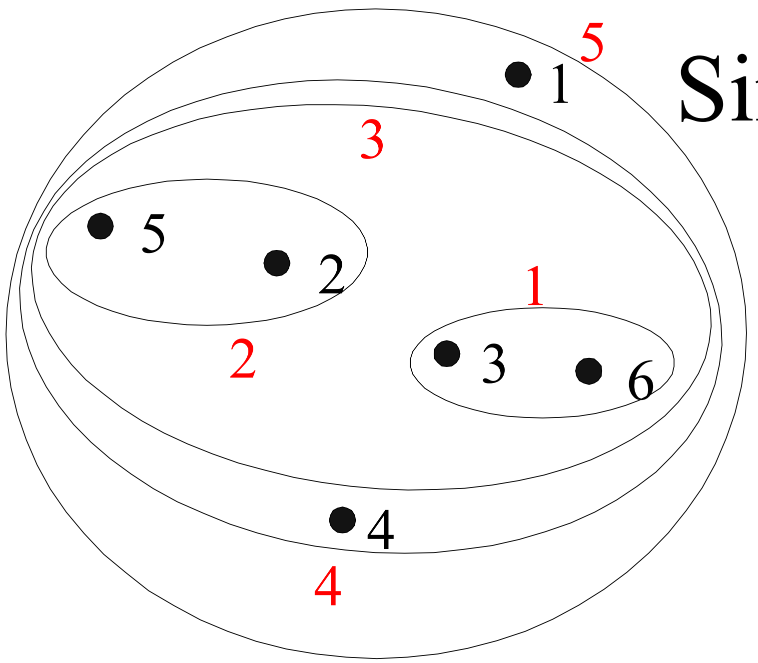
Comparison



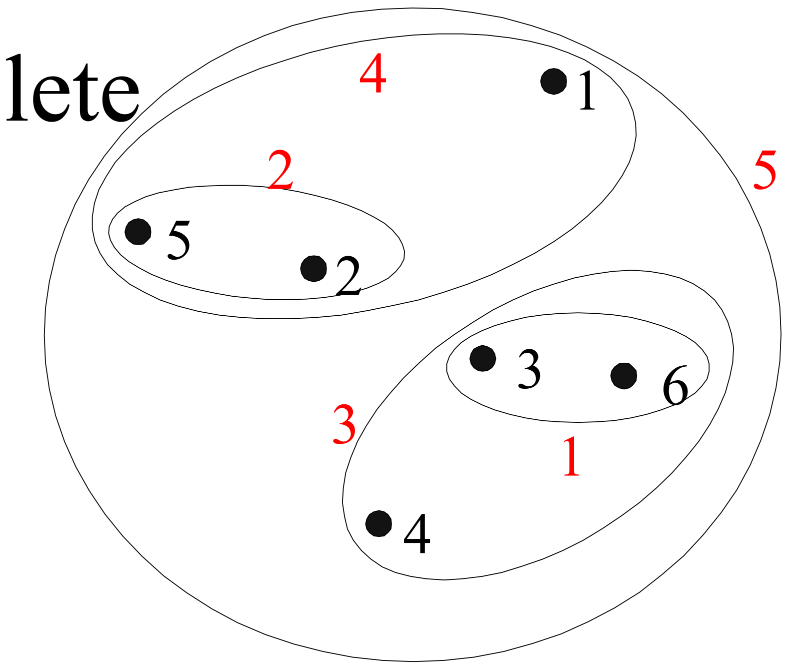
Comparison



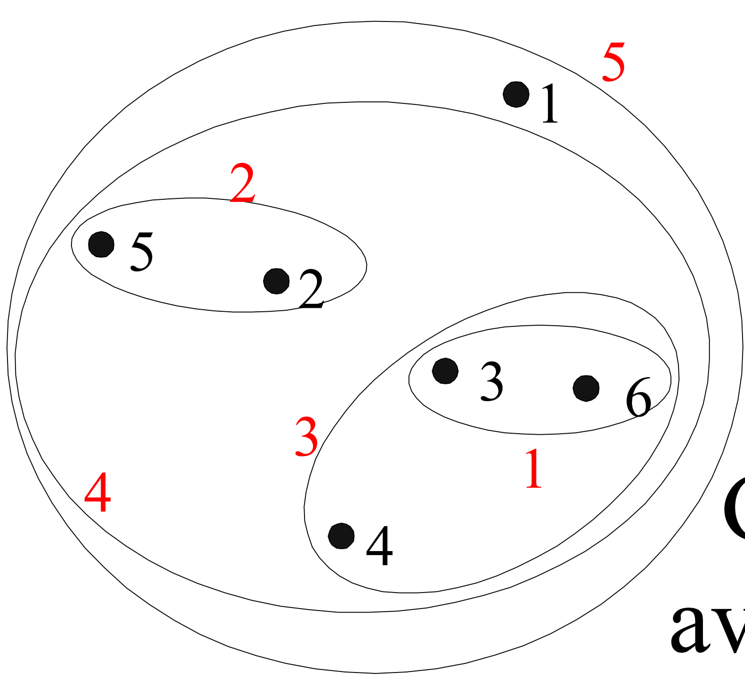
Comparison



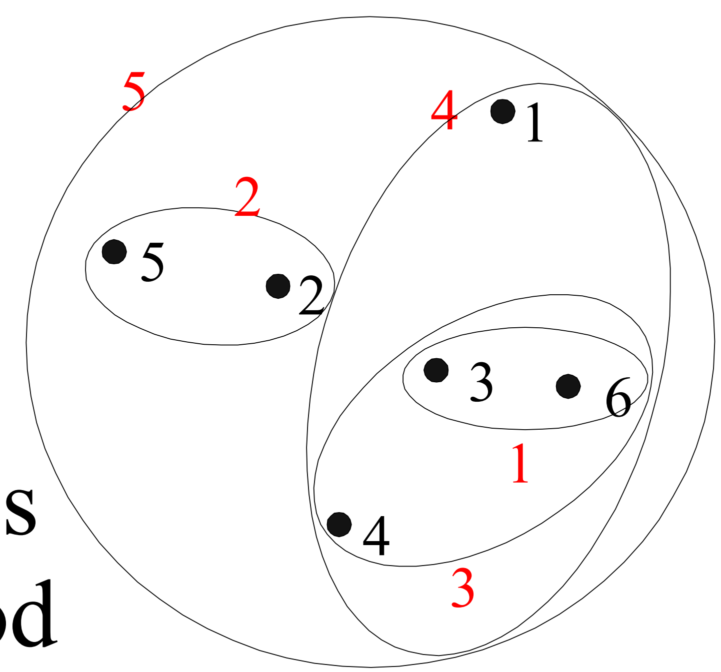
Single link



Complete link



Group average



Ward's method

Lance–Williams formula

- After merging clusters A and B into cluster C, we need to compute C's distance to other clusters Z
- Lance–Williams formula provides a general equation for this

$$d(C,Z) = \alpha_A d(A,Z) + \alpha_B d(B,Z) + \beta d(A,B) + \gamma \left| d(A,Z) - d(B,Z) \right|$$

	α_A	α_B	β	γ
Single link	1/2	1/2	0	–1/2
Complete link	1/2	1/2	0	1/2
Group average	$\|A\|/(\|A\| + \|B\|)$	$\|B\|/(\|A\| + \|B\|)$	0	0
Mean distance	$\|A\|/(\|A\| + \|B\|)$	$\|B\|/(\|A\| + \|B\|)$	$-\|A\|\|B\|/(\|A\|+\|B\|)^2$	0
Ward's method	$(\|A\|+\|Z\|)/(\|A\|+\|B\|+\|Z\|)$	$(\|B\|+\|Z\|)/(\|A\|+\|B\|+\|Z\|)$	$-\|Z\|/(\|A\|+\|B\|+\|Z\|)$	0

Computational complexity

- Takes $O(n^3)$ time in most cases
  - n steps
  - In each step, n^2 distance matrix must be updated and searched
- $O(n^2 \log(n))$ time for some approaches using appropriate data structures
  - Keep distances in a heap
  - Each step takes $O(n \log n)$ time
- $O(n^2)$ space complexity
  - Have to store the distance matrix

Chapter VIII.4: Co-clustering

1. Clustering written with matrices
2. Co-clustering definition
3. Algorithms

Clustering written with matrices

- Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ be the m-dimensional vectors (data points) we want to cluster
- Write these as an n-by-m matrix $\boldsymbol{X}$
  - Each data point is one row of $\boldsymbol{X}$
- The exclusive representative clustering can be re-written using two matrices
  - Matrix $\boldsymbol{C}$ (cluster assignment matrix) has n rows and k columns
  - Each row of $\boldsymbol{C}$ has *exactly* one element 1 while others are 0
  - Matrix $\boldsymbol{M}$ (mean matrix) has k rows and m columns
  - Each row of $\boldsymbol{M}$ corresponds to a centroid of a cluster
- Loss function (SSE) is now $\|\mathbf{X} - \mathbf{CM}\|_2^2$

Example

x_1	1	3
x_2	2	2
x_3	3	4
x_4	2	1
x_5	4	3

Example

x₁	1	3
x₂	2	2
x₃	3	4
x₄	2	1
x₅	4	3

$$\mathbf{X} = \begin{pmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 4 \\ 2 & 1 \\ 4 & 3 \end{pmatrix}$$

Example

x_1	1	3
x_2	2	2
x_3	3	4
x_4	2	1
x_5	4	3

$$\mathbf{X} = \begin{pmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 4 \\ 2 & 1 \\ 4 & 3 \end{pmatrix}$$

$C_1 = \{x_1, x_2, x_4\}$
$C_2 = \{x_3, x_5\}$

Example

x_1	1	3
x_2	2	2
x_3	3	4
x_4	2	1
x_5	4	3

$$\mathbf{X} = \begin{pmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 4 \\ 2 & 1 \\ 4 & 3 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$C_1 = \{x_1, x_2, x_4\}$
$C_2 = \{x_3, x_5\}$

Example

x_1	1	3
x_2	2	2
x_3	3	4
x_4	2	1
x_5	4	3

$$\mathbf{X} = \begin{pmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 4 \\ 2 & 1 \\ 4 & 3 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$C_1 = \{x_1, x_2, x_4\}$
$C_2 = \{x_3, x_5\}$

$\boldsymbol{\mu}1 = (1.66, 2)$
$\boldsymbol{\mu}2 = (3.5, 3.5)$

Example

x_1	1	3
x_2	2	2
x_3	3	4
x_4	2	1
x_5	4	3

$$\mathbf{X} = \begin{pmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 4 \\ 2 & 1 \\ 4 & 3 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$C_1 = \{x_1, x_2, x_4\}$
$C_2 = \{x_3, x_5\}$

$\boldsymbol{\mu} 1 = (1.66, 2)$
$\boldsymbol{\mu} 2 = (3.5, 3.5)$

$$\mathbf{M} = \begin{pmatrix} 1.66 & 2 \\ 3.5 & 3.5 \end{pmatrix}$$

Example

x_1	1	3
x_2	2	2
x_3	3	4
x_4	2	1
x_5	4	3

$$\mathbf{X} = \begin{pmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 4 \\ 2 & 1 \\ 4 & 3 \end{pmatrix}$$

$$C_1 = \{x_1, x_2, x_4\}$$
$$C_2 = \{x_3, x_5\}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\boldsymbol{\mu}1 = (1.66, 2)$$
$$\boldsymbol{\mu}2 = (3.5, 3.5)$$

$$\mathbf{M} = \begin{pmatrix} 1.66 & 2 \\ 3.5 & 3.5 \end{pmatrix}$$

$$\mathbf{CM} = \begin{pmatrix} 1.66 & 2 \\ 1.66 & 2 \\ 3.5 & 3.5 \\ 1.66 & 2 \\ 3.5 & 3.5 \end{pmatrix}$$

Example

x_1	1	3
x_2	2	2
x_3	3	4
x_4	2	1
x_5	4	3

$$\mathbf{X} = \begin{pmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 4 \\ 2 & 1 \\ 4 & 3 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$C_1 = \{x_1, x_2, x_4\}$
$C_2 = \{x_3, x_5\}$

$\boldsymbol{\mu} 1 = (1.66, 2)$
$\boldsymbol{\mu} 2 = (3.5, 3.5)$

$$\mathbf{M} = \begin{pmatrix} 1.66 & 2 \\ 3.5 & 3.5 \end{pmatrix}$$

$$\mathbf{X} - \mathbf{CM} = \begin{pmatrix} -0.66 & 1 \\ 0.33 & 0 \\ -0.5 & 0.5 \\ 0.33 & -1 \\ 0.5 & -0.5 \end{pmatrix}$$

Co-clustering definition

- The same way we clustered $\boldsymbol{X}$, we can also cluster $\boldsymbol{X}^T$
  - This clusters the dimensions, not the data points
- An **(k,l)-co-clustering** of $\boldsymbol{X}$ is partitioning of rows of $\boldsymbol{X}$ into k clusters and columns of $\boldsymbol{X}$ into l clusters
  - Row cluster I and column cluster J define a (combinatorial) **sub-matrix** $\boldsymbol{X}_{IJ}$
    - Element x_{ij} belongs to this sub-matrix if $i \in I$ and $j \in J$
  - Each sub-matrix $\boldsymbol{X}_{IJ}$ is represented by *single value* μ_{ij}
- Let $\boldsymbol{R}$ be the n-by-k row cluster assignment matrix and $\boldsymbol{C}$ the m-by-l column cluster assignment matrix and $\boldsymbol{M} = (\mu_{ij})$ the k-by-l mean matrix
  - The *loss function* is $\|\mathbf{X} - \mathbf{RMC}^{\mathrm{T}}\|_2^2$

Example (3,2)-co-clustering

$$\mathsf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix}$$

Example (3,2)-co-clustering

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{C}^\mathsf{T} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Example (3,2)-co-clustering

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} 1.5 & 2.5 \\ 0 & 1 \\ 4.5 & 3 \end{pmatrix} \qquad \mathbf{C}^T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Example (3,2)-co-clustering

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} 1.5 & 2.5 \\ 0 & 1 \\ 4.5 & 3 \end{pmatrix} \qquad \mathbf{C}^T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Example (3,2)-co-clustering

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} 1.5 & 2.5 \\ 0 & 1 \\ 4.5 & 3 \end{pmatrix} \qquad \mathbf{C}^T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Example (3,2)-co-clustering

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} 1.5 & 2.5 \\ 0 & 1 \\ 4.5 & 3 \end{pmatrix} \qquad \mathbf{C}^T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Example (3,2)-co-clustering

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{M} = \begin{pmatrix} 1.5 & 2.5 \\ 0 & 1 \\ 4.5 & 3 \end{pmatrix} \quad \mathbf{C}^{\mathsf{T}} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Example (3,2)-co-clustering

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} 1.5 & 2.5 \\ 0 & 1 \\ 4.5 & 3 \end{pmatrix} \qquad \mathbf{C}^{\mathsf{T}} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Example (3,2)-co-clustering

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} 1.5 & 2.5 \\ 0 & 1 \\ 4.5 & 3 \end{pmatrix} \qquad \mathbf{C}^{\mathsf{T}} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Example (3,2)-co-clustering

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} 1.5 & 2.5 \\ 0 & 1 \\ 4.5 & 3 \end{pmatrix} \qquad \mathbf{C}^{\mathsf{T}} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Example (3,2)-co-clustering

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix} \quad \mathbf{RMC}^T = \begin{pmatrix} 1.5 & 2.5 & 1.5 \\ 1.5 & 2.5 & 1.5 \\ 0 & 1 & 0 \\ 4.5 & 3 & 4.5 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{M} = \begin{pmatrix} 1.5 & 2.5 \\ 0 & 1 \\ 4.5 & 3 \end{pmatrix} \quad \mathbf{C}^T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Example (3,2)-co-clustering

$$\left| \mathbf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix} - \mathbf{RMC}^{\mathsf{T}} = \begin{pmatrix} 1.5 & 2.5 & 1.5 \\ 1.5 & 2.5 & 1.5 \\ 0 & 1 & 0 \\ 4.5 & 3 & 4.5 \end{pmatrix} \right| = \begin{pmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0 & 0 & 0 \\ 0.5 & 0 & 0.5 \end{pmatrix}$$

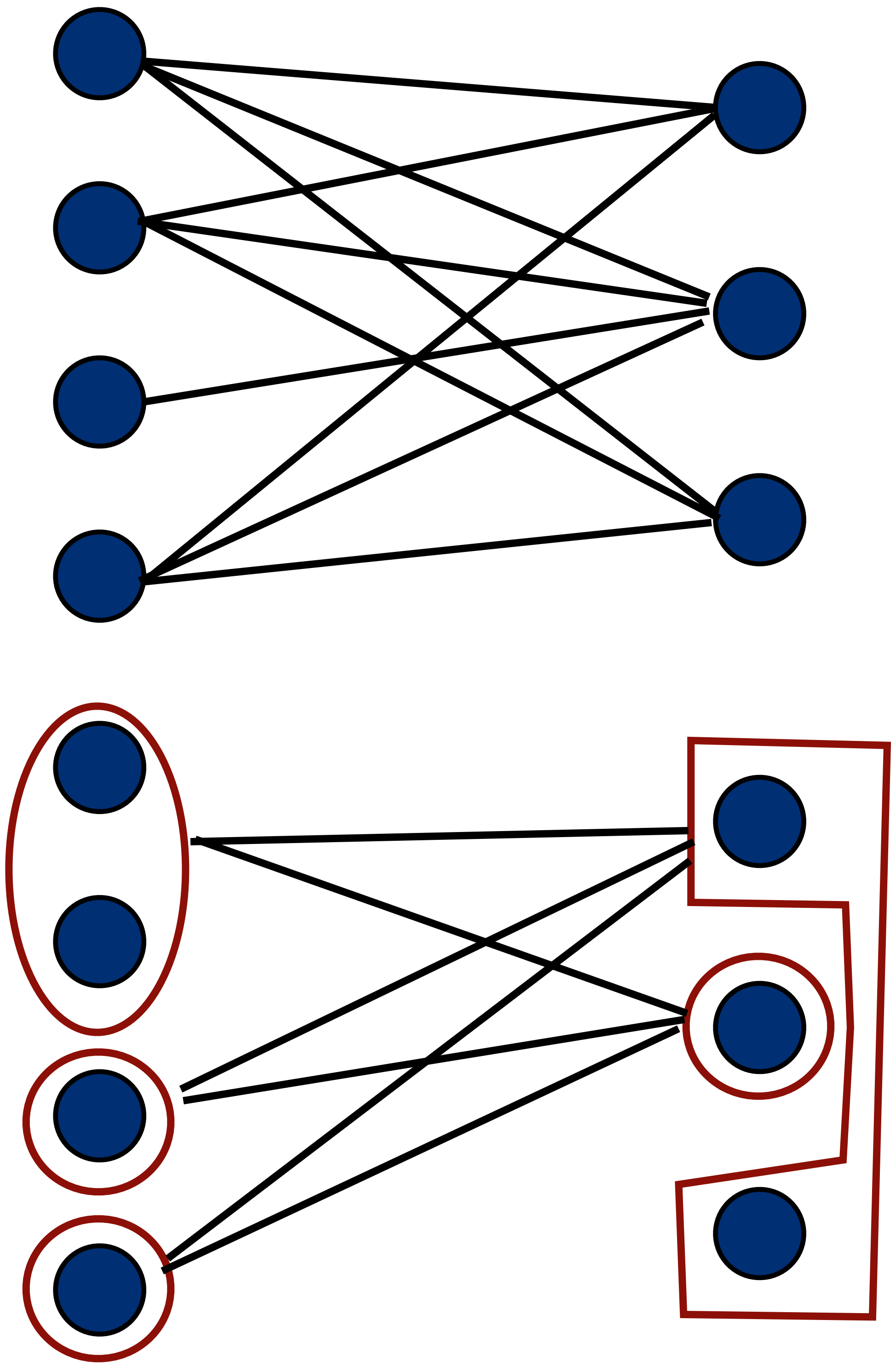
$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} 1.5 & 2.5 \\ 0 & 1 \\ 4.5 & 3 \end{pmatrix} \qquad \mathbf{C}^{\mathsf{T}} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Co-clustering and bipartite graphs

- A *graph* $G=(V,E)$ is *bipartite* if its set of vertices can be partitioned into two sets, L and R, such that all edges in E have one end in L and other in R
- Any n-by-m matrix can be considered as a *weighted bipartite graph*
  - Rows correspond to vertices in L
  - Columns correspond to vertices in R
  - Edge (i,j) has weight x_{ij}
- A co-clustering now clusters vertices in L and vertices in R and replaces edges in E with edges between the clusters having weights μ_{IJ}

Example

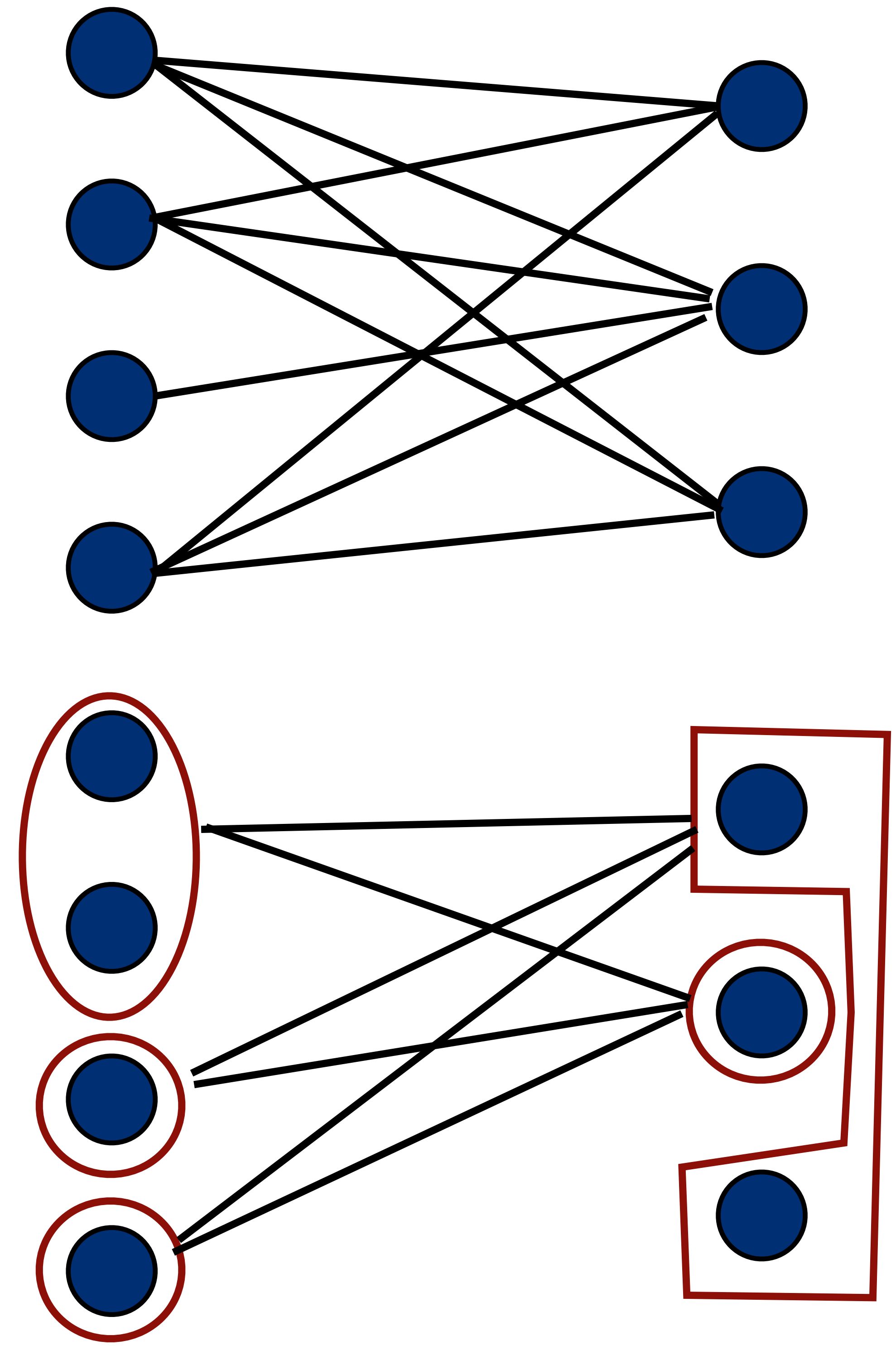
$$\mathsf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix}$$



Example

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \\ 4 & 3 & 5 \end{pmatrix}$$

$$\mathbf{RMC}^{T} = \begin{pmatrix} 1.5 & 2.5 & 1.5 \\ 1.5 & 2.5 & 1.5 \\ 0 & 1 & 0 \\ 4.5 & 3 & 4.5 \end{pmatrix}$$



Algorithm

1. **input** data matrix $\boldsymbol{X}$ and two integers k and l
2. Cluster the rows of $\boldsymbol{X}$ to $\boldsymbol{R}$ (using e.g. k-means)
3. Cluster the columns of $\boldsymbol{X}$ to $\boldsymbol{C}$
4. Let $\boldsymbol{M} = (\mu_{IJ})$, $\mu_{IJ} = (|I|\,|J|)^{-1} \sum_{i \in I, j \in J} x_{ij}$
5. **return** $\boldsymbol{R}$, $\boldsymbol{C}$, and $\boldsymbol{M}$

Chapter VIII.5: Discussion and clustering applications

1. Local and global patterns
2. Kleinberg's impossibility theorem
3. Example clustering applications

Local and global patterns

- The quality of an association rule depends only on the rule itself
- The quality of a cluster depends on all the clusters in the clustering
  - Singleton clusters have the least SSE, but having k–1 singletons and one big cluster typically gives high total SSE
- Association rules are *local* patterns
  - Their goodness depends only on the local part of the data
- Clusters are *global* patterns
  - The overall quality depends also on points not in the cluster

Kleinberg's impossibility theorem

- A *clustering function* is a function f that takes a distance matrix D and returns a partition Γ
  - We expect nothing on the type of points
  - Distance is given using an implicit distance matrix
  - The number of clusters is defined somehow by the clustering function (build-in constant or something else)
  - For example, an algorithm returning a k-means clustering to $k=10$ clusters could be one clustering function
- Idea: list some properties any clustering function should satisfy and show that none can satisfy them all

Three properties

- *Scale-invariance*
  - Clustering does not change if we multiply the distances
  - $f(D) = f(\alpha D)$ for any $\alpha > 0$
- *Richness*
  - For any partition Γ, there is a distance matrix D such that $f(D) = \Gamma$
- *Consistency*
  - The clustering does not change if we move points in the same cluster closer to each other and points in different clusters further away from each other

Impossibility result

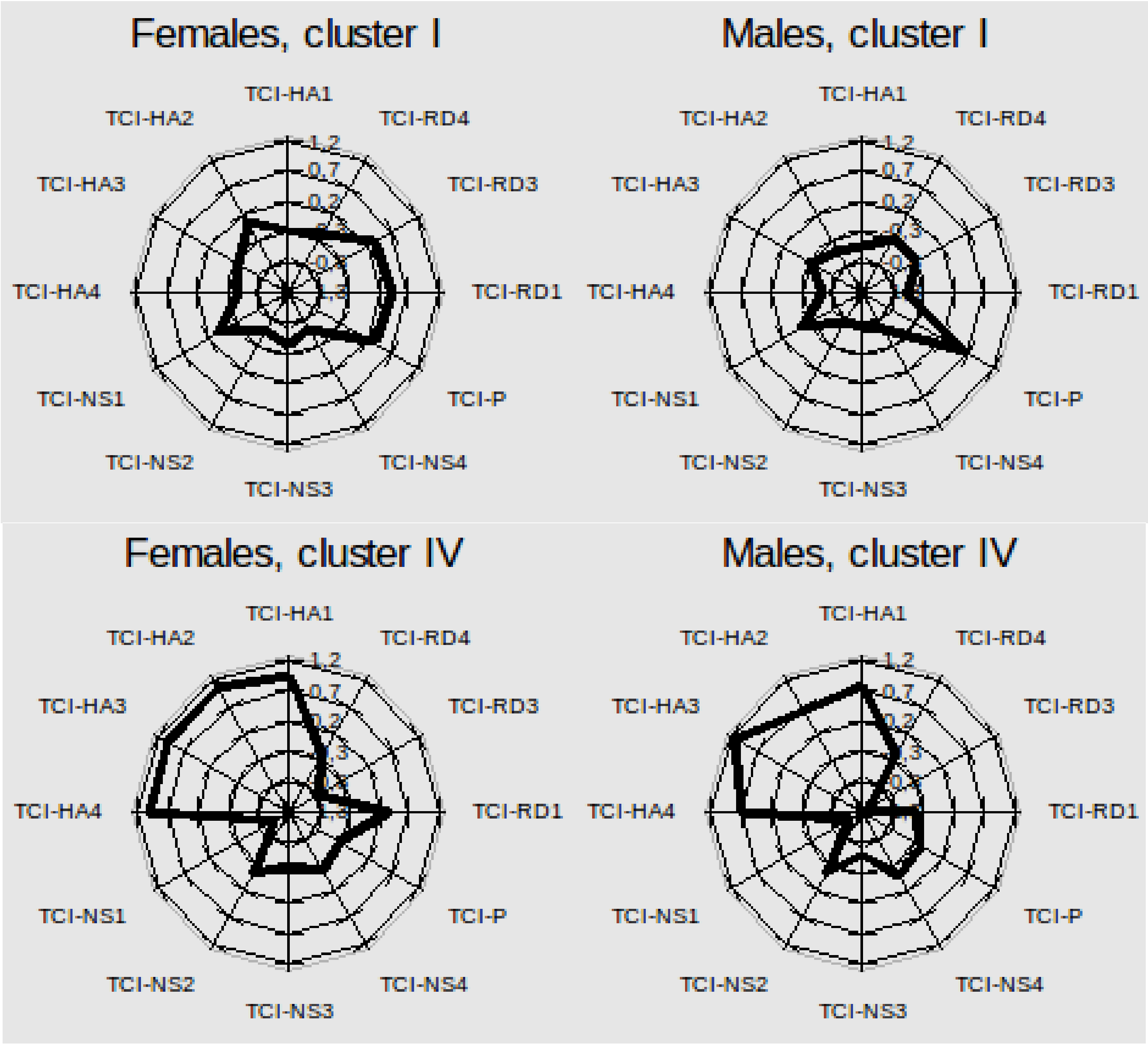
Theorem [Kleinberg '02]. There does not exist any clustering function *f* that satisfies all three properties.

- Single-link hierarchical clustering that stops at $k < n$ clusters satisfies scale-invariance and consistency
- Single-link clustering that stops when the link length is some predefined fraction of maximum pairwise distance satisfies scale-invariance and richness
- Single-link that stops when the link length is longer than some predefined length satisfies richness and consistency

Some clustering applications

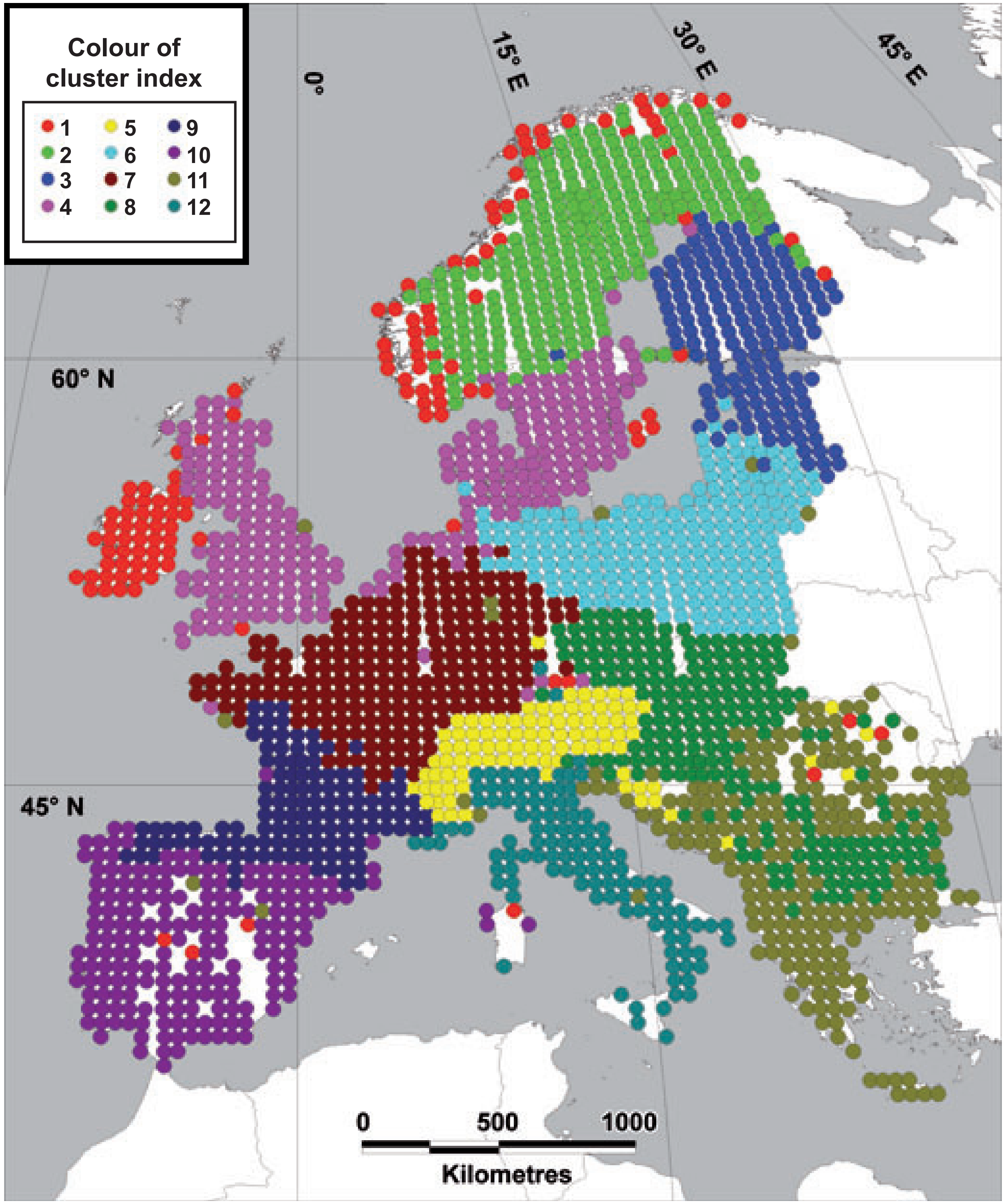
- Biology
  - Creation of phylogenies (relations between organisms)
  - Inferring population structures from clusterings of DNA data
  - Analysis of genes and cellular processes (co-clustering)
- Business
  - Grouping of consumers into market segments
- Computer science
  - Pre-processing step to reduce computation (representative-based methods)
  - Automatic discovery of similar items

More clustering applications



Wessman: Clustering methods in the analysis of complex diseases

Even more clustering applications



Heikinheimo et al.: Clustering of European mammals, 2007