Hierarchical Clustering

Ke Chen

Department of Computer Science, The University of Manchester

Ke.Chen@manchester.ac.uk

Outline

Introduction

Hierarchical clustering overview and illustration

Cluster Distance Metric

Single-link, complete-link and group average measures

Agglomerative Algorithm

Algorithmic description of Agglomerative clustering

Illustrative Example

Step-by-step Agglomerative clustering demo on synthetic dataset and SARS virus tracking

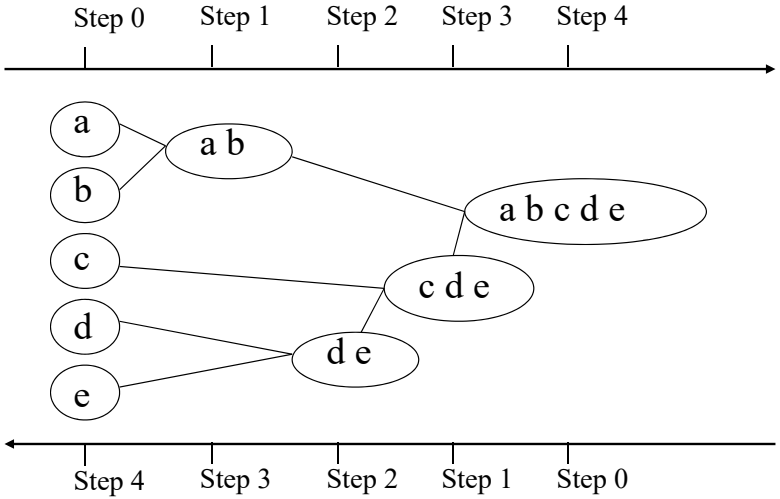
Relevant Issue

Strength versus weakness of cluster distance metrics and limitations

- Hierarchical clustering
  - partition dataset sequentially by constructing nested partitions layer by layer represented by a tree of clusters
  - Unlike K-means, no need to know the number of clusters
  - require extended cluster distance metrics for clustering
- Hierarchical clustering strategy
  - Agglomerative: bottom-up strategy: initially treat every data points as its own atomic cluster, then merge atomic clusters into larger and larger clusters
  - Divisive: top-down strategy: initially treat all data points as one single cluster, then divide the largest cluster into smaller and smaller clusters

- Illustration of hierarchical clustering strategies

  A simple dataset of 5 data points: {a, b, c, d, e}



Cluster Distance Metric

- Single-link: smallest distance between a point in one cluster and a point in the other, i.e.,

$$d_{SL}(C_i, C_j) = \min_{\boldsymbol{a} \in C_i, \boldsymbol{b} \in C_j} d(\boldsymbol{a}, \boldsymbol{b}).$$

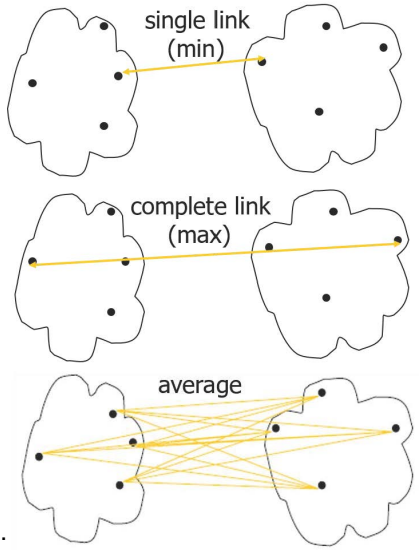
- Complete-link: largest distance between a point in one cluster and a point in the other, i.e.,

$$d_{CL}(C_i, C_j) = \max_{\boldsymbol{a} \in C_i, \boldsymbol{b} \in C_j} d(\boldsymbol{a}, \boldsymbol{b}).$$

- Group-average: averaged distance between a point in one cluster and a point in the other, i.e.,

$$d_{GA}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{\boldsymbol{a} \in C_i, \boldsymbol{b} \in C_j} d(\boldsymbol{a}, \boldsymbol{b}).$$

Stipulation: $d_{SL}(C_i, C_i) = d_{CL}(C_i, C_i) = d_{GA}(C_i, C_i) = 0$.



Example: Given a data set of five objects characterised by a single continuous feature, assume that there are two clusters: C_1: {a, b} and C_2: {c, d, e}. (Minkowski distance for distance matrix)

	a	b	c	d	e
Feature	1	2	4	5	6

1. Calculate the distance matrix .

	a	b	c	d	e
a	0	1	3	4	5
b	1	0	2	3	4
c	3	2	0	1	2
d	4	3	1	0	1
e	5	4	2	1	0

2. Calculate three cluster distances between C_1 and C_2.

Single-link

$$\begin{aligned}\text{dist}(C_1, C_2) &= \min\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\} \\ &= \min\{3, 4, 5, 2, 3, 4\} = 2\end{aligned}$$

Complete-link

$$\begin{aligned}\text{dist}(C_1, C_2) &= \max\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\} \\ &= \max\{3, 4, 5, 2, 3, 4\} = 5\end{aligned}$$

Group-average

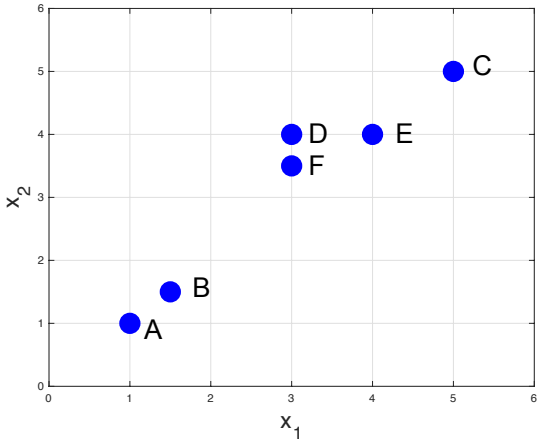
$$\text{dist}(C_1, C_2) = \frac{d(a,c) + d(a,d) + d(a,e) + d(b,c) + d(b,d) + d(b,e)}{6}$$

$$\frac{3+4+5+2+3+4}{6} = \frac{21}{6} = 3.5$$

Input: Distance matrix given or calculated from dataset
Choose one appropriate cluster distance from d_{SL}, d_{CL} and d_{GA}

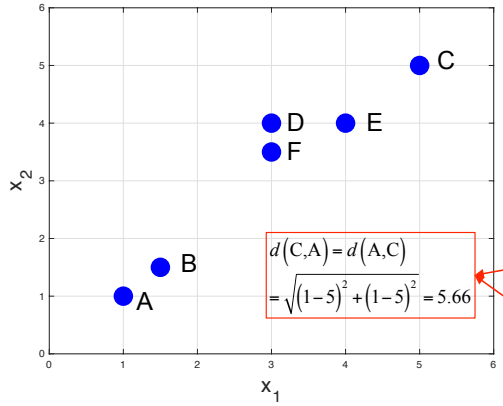
- **Initialisation**: set each data point in dataset as an atomic cluster
- **Step 1**: merge two or more (if equally) closest clusters
- **Step 2**: update the (extended) distance matrix
- **Step 3**: repeat **Step 1** and **Step 2** until only a single cluster remains or K clusters appear (when the number of clusters, K, known in advance)

Example 1: Apply Agglomerative algorithm to synthetic data



A: (1, 1) B: (1.5, 1.5)
C: (5, 5) D: (3, 4)
E: (4, 4) F: (3, 3.5)

Example 1: Apply Agglomerative algorithm to synthetic data



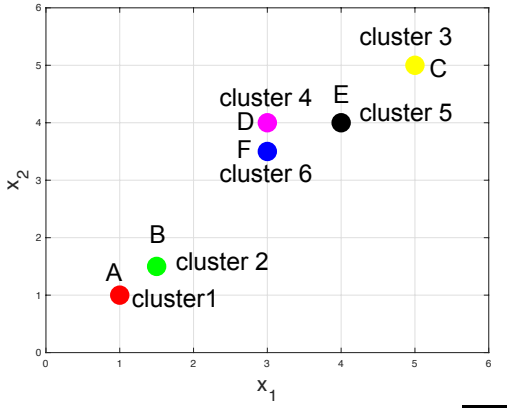
A: (1, 1) B: (1.5, 1.5)
C: (5, 5) D: (3, 4)
E: (4, 4) F: (3, 3.5)

- Step 1: Calculate the distance matrix **D** between all the data points. Use Euclidean in this example.

D matrix

	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Example 1: Apply Agglomerative algorithm to synthetic data



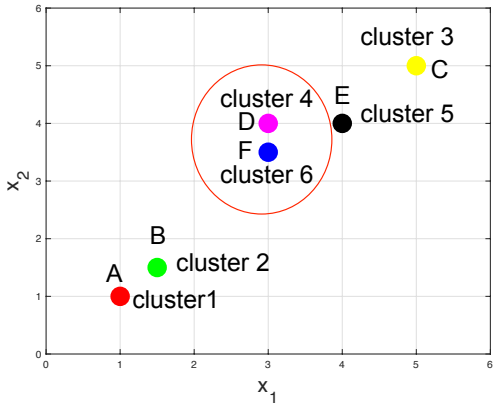
A: (1, 1) B: (1.5, 1.5)
C: (5, 5) D: (3, 4)
E: (4, 4) F: (3, 3.5)

- Step 2: Set each data point as a cluster. The between-cluster matrix **M** is equal to **D**.

M matrix

	{A} C1	{B} C2	{C} C3	{D} C4	{E} C5	{F} C6
C1	0.00	0.71	5.66	3.61	4.24	3.20
C2	0.71	0.00	4.95	2.92	3.54	2.50
C3	5.66	4.95	0.00	2.24	1.41	2.50
C4	3.61	2.92	2.24	0.00	1.00	0.50
C5	4.24	3.54	1.41	1.00	0.00	1.12
C6	3.20	2.50	2.50	0.50	1.12	0.00

Example 1: Apply Agglomerative algorithm to synthetic data



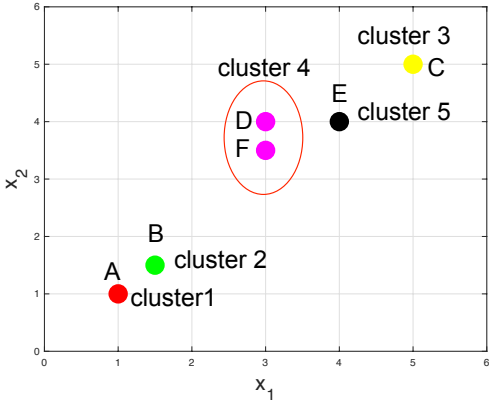
A: (1, 1) B: (1.5, 1.5)
C: (5, 5) D: (3, 4)
E: (4, 4) F: (3, 3.5)

- Step 3: Merge the two closest clusters, and update the between-cluster distance matrix M. Use single-link in this example.

M matrix

	{A} C1	{B} C2	{C} C3	{D} **C4**	{E} C5	{F} **C6**
C1	0.00	0.71	5.66	3.61	4.24	3.20
C2	0.71	0.00	4.95	2.92	3.54	2.50
C3	5.66	4.95	0.00	2.24	1.41	2.50
C4	3.61	2.92	2.24	0.00	1.00	0.50
C5	4.24	3.54	1.41	1.00	0.00	1.12
C6	3.20	2.50	2.50	0.50	1.12	0.00

Example 1: Apply Agglomerative algorithm to synthetic data



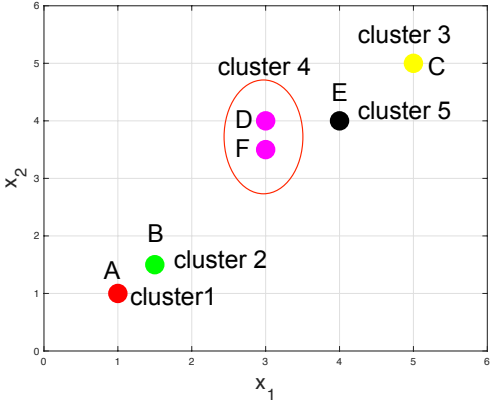
A: (1, 1) B: (1.5, 1.5)
C: (5, 5) D: (3, 4)
E: (4, 4) F: (3, 3.5)

- Step 3: Merge the two closest clusters, and update the between-cluster distance matrix M. Use single-link in this example.

M matrix

		{A} C1	{B} C2	{C} C3	{D,F} **C4**	{E} C5
{A}	C1	0.00	0.71	5.66	?	4.24
{B}	C2	0.71	0.00	4.95	?	3.54
{C}	C3	5.66	4.95	0.00	?	1.41
{D,F}	**C4**	?	?	?	0.00	?
{E}	C5	4.24	3.54	1.41	?	0.00

Example 1: Apply Agglomerative algorithm to synthetic data



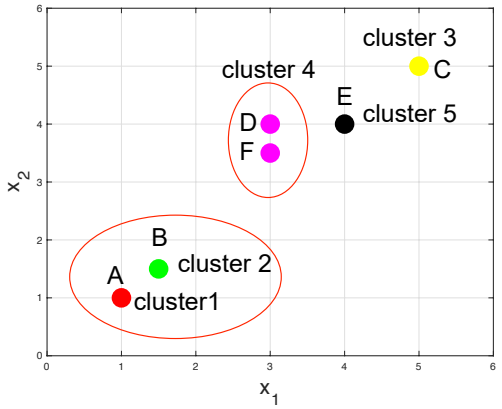
A: (1, 1) B: (1.5, 1.5)
C: (5, 5) D: (3, 4)
E: (4, 4) F: (3, 3.5)

- Step 3: Merge the two closest clusters, and update the between-cluster distance matrix M. Use single-link in this example.

M matrix

		{A} C1	{B} C2	{C} C3	{D,F} C4	{E} C5
{A}	C1	0.00	0.71	5.66	3.20	4.24
{B}	C2	0.71	0.00	4.95	2.50	3.54
{C}	C3	5.66	4.95	0.00	2.24	1.41
{D,F}	C4	3.20	2.50	2.24	0.00	1.00
{E}	C5	4.24	3.54	1.41	1.00	0.00

Example 1: Apply Agglomerative algorithm to synthetic data



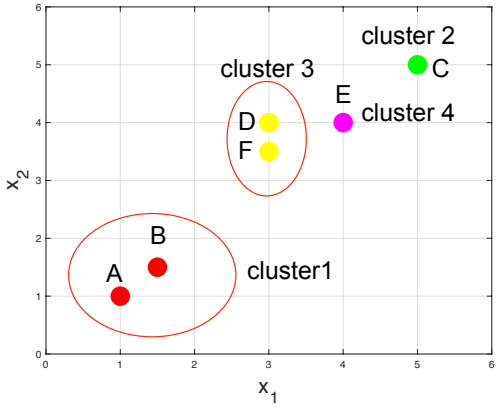
A: (1, 1) B: (1.5, 1.5)
C: (5, 5) D: (3, 4)
E: (4, 4) F: (3, 3.5)

M matrix

- Repeat Step 3: Merge the two closest clusters, and update the between-cluster distance matrix M. Use single-link in this example.

		{A} C1	{B} C2	{C} C3	{D,F} C4	{E} C5
{A}	C1	0.00	0.71	5.66	3.20	4.24
{B}	C2	0.71	0.00	4.95	2.50	3.54
{C}	C3	5.66	4.95	0.00	2.24	1.41
{D,F}	C4	3.20	2.50	2.24	0.00	1.00
{E}	C5	4.24	3.54	1.41	1.00	0.00

Example 1: Apply Agglomerative algorithm to synthetic data



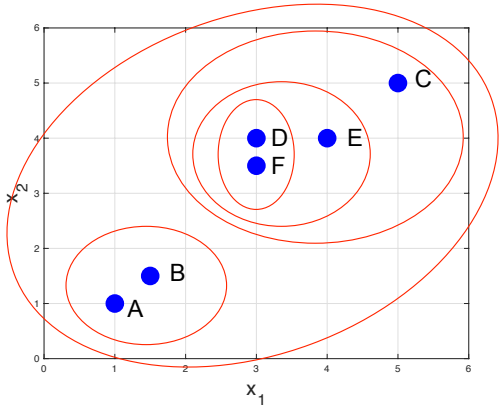
A: (1, 1) B: (1.5, 1.5)
C: (5, 5) D: (3, 4)
E: (4, 4) F: (3, 3.5)

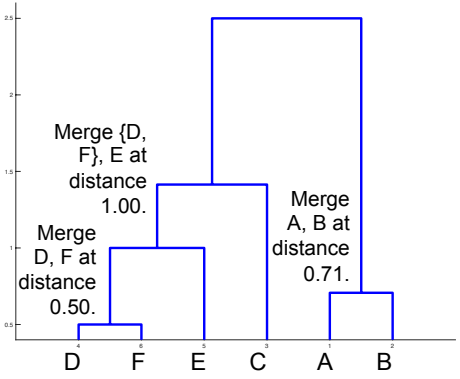
- There are 3 more steps left to reach one single cluster. Complete those steps by yourself.

M matrix

		{A,B} C1	{C} C2	{D,F} C3	{E} C4
{A,B}	C1	0.00	4.95	2.50	3.54
{C}	C2	4.95	0.00	2.24	1.41
{D,F}	C3	2.50	2.24	0.00	1.00
{E}	C4	3.54	1.41	1.00	0.00

Example 1: Apply Agglomerative algorithm to synthetic data

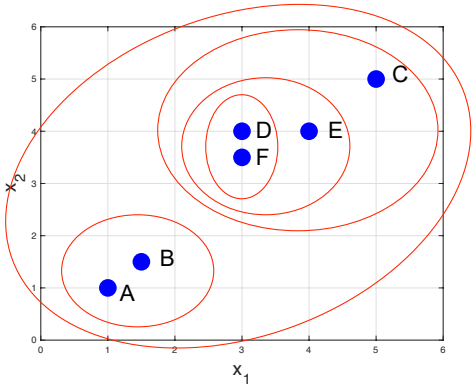


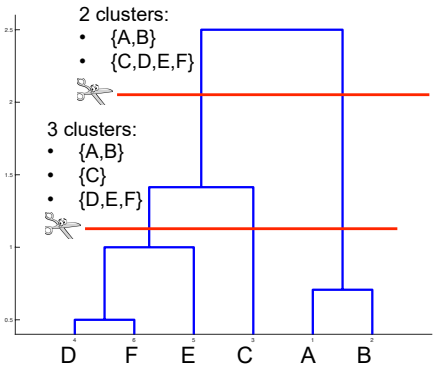


A: (1, 1) B: (1.5, 1.5)
C: (5, 5) D: (3, 4)
E: (4, 4) F: (3, 3.5)

Lifetime of a cluster is the distance difference between that it is created and merged. E.g., Lifetime of {D, F} = 1.00-0.50=0.50

Example 1: Apply Agglomerative algorithm to synthetic data





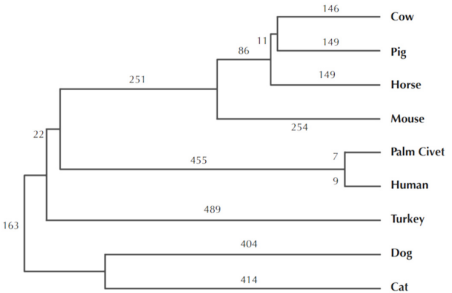
A: (1, 1) B: (1.5, 1.5)
C: (5, 5) D: (3, 4)
E: (4, 4) F: (3, 3.5)

K-cluster lifetime is the distance difference between that *K* clusters created and merged. The longest *K*-cluster lifetime used to decide the number of clusters, *K*, for this dataset.

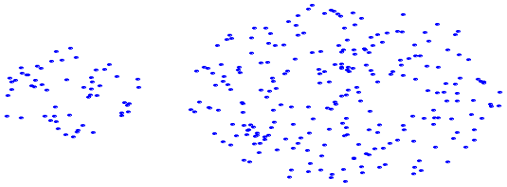
Example 2: SARS virus tracking

- A virus of high mutation rates leads to the similarity of the DNA sequence of the same virus depending on the time since it was transmitted, which can trace paths of transmission.
- With DNA sequences collected from human and animals, Agglomerative algorithm has been applied to assist tracking SARS virus.
- "With the data at hand, we see how the virus used different hosts, moving from bat to human to civet, in that order. So the civets actually got SARS from humans."
  — ***Science Daily***

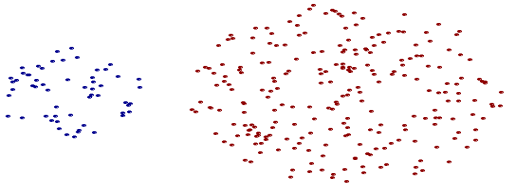
Palm Civet



- Single-link strength: able to handle data of non-Gaussian distribution

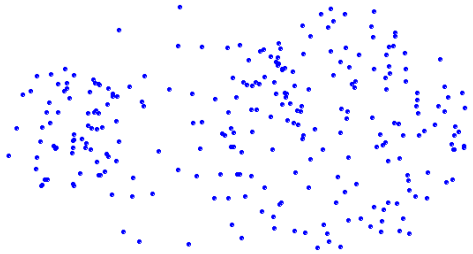


Original Points

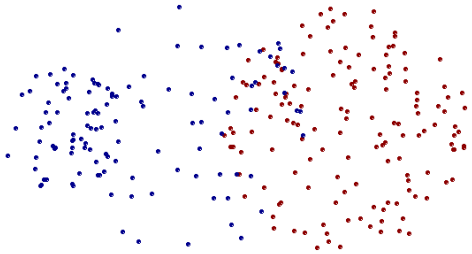


Two Clusters

- Single-link limitation: sensitive to noise and outliers

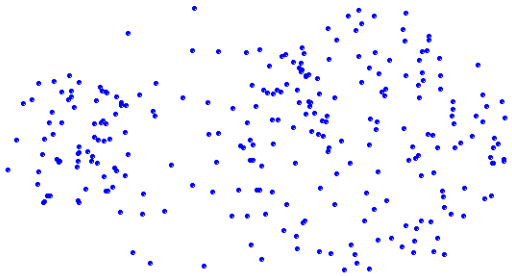


Original Points

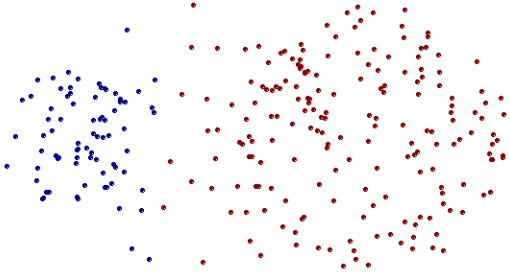


Two Clusters

- Complete-link strength: less sensitive to noise and outliers

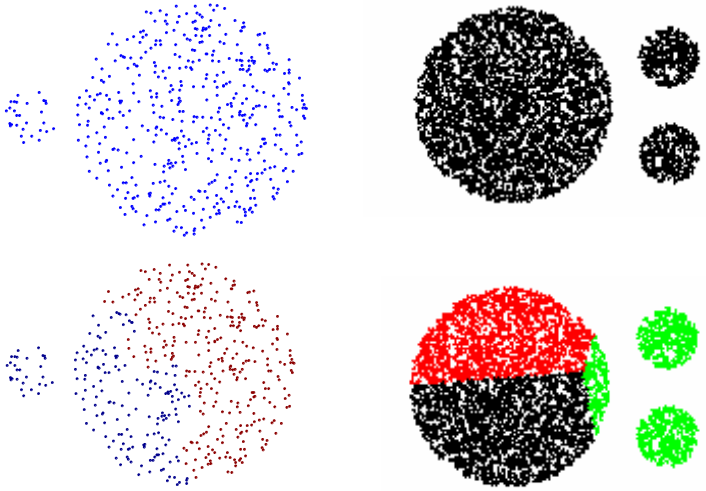


Original Points

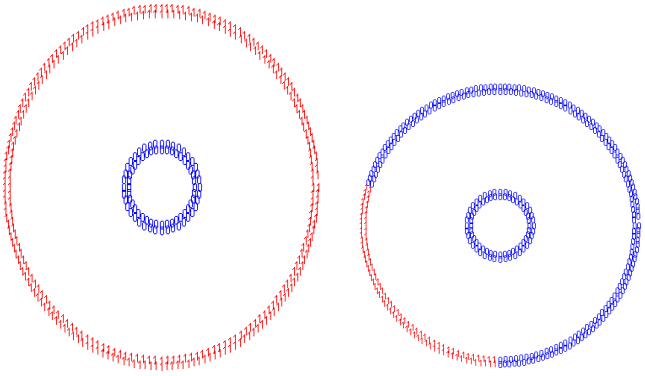


Two Clusters

- Complete-link limitation-1: tend to break large clusters



- Complete-link limitation-2: biased towards globular clusters



MIN (2 clusters) MAX (2 clusters)

- Group-average: compromise between single and complete link
  - Strength: less sensitive to noise and outliers
  - Limitation: biased towards globular clusters

- Limitation of Agglomerative algorithm
  - unable to undo after two clusters merged
  - no loss or objective function to be optimised directly
  - various problems arising from different cluster distance metrics
  - high computational complexity, $O(N^2 \log N)$, where N is the number of data points in a dataset
- Agglomerative algorithm variants
  - BIRCH: scalable to a large dataset for big data
  - ROCK: able to conduct clustering analysis on categorical data
  - CHAMELEON: hierarchical clustering using dynamic modelling

Reference

If you want to deepen your understanding and learn something beyond this lecture, you can self-study the optional references below.

[Alpaydin, 2014] Alpaydin E. (2014): *Introduction to Machine Learning* (3rd Ed.), MIT Press. (Section 7.9)

[Jain et al., 1999] Jain A.K., Murty M.N. and Flynn P.J. (1999): Data clustering: A review. *ACM Computing Survey*, Vol. 31, No. 3, pp. 264-323.

[Xu & Wunsch II, 2005] Xu R. and Wunsch II D. (2005): Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, Vol. 16, No. 3, pp. 645-678.