CS5228: Knowledge Discovery and Data Mining

Lecture 3 — Clustering II

Course Logistics — Update

Quick Recap — Tutorial

ity	state	paren
o-b	PA	Bach
p c	OK	S
o-b	FL	ma
p a	CT	
p c	WV	
p b	IN	asso
p b	CO	
o-b	MP	
o-d	WY	
o-b	MS	

- Alternative encoding of nominal attributes
  - Replace nominal values with 1 or more numerical values
  - Numerical values should reflect underlying assumption of the impact of attribute
  - Example: *"What makes 'state' a potentially useful attribute?"*

Interpretation		Encoding through replacement
Average Political leaning	➜	Percentage of democrats/republicans
State education budget	➜	Dollar-per-student value
School system	➜	Rate of homeschooling
Urbanization	➜	#universities per capita
…	➜	…

Quick Recap — Tutorial

ity	state	paren
o-b	PA	Bach
p c	OK	S
o-b	FL	ma
p a	CT	
p c	WV	
p b	IN	assoc
p b	CO	
o-b	MP	
o-d	WY	
o-b	MS	

- Important: "careless" encoding may imply questionable interpretation
  - Question: *"What is the interpretation of my encoding, and is it meaningful?"*
  - In practice, often very difficult to answer

Encoding		Interpretation
Ordinal values	➜	PA < OK < FL < CT < WV < …
Latitude/Longitude	➜	Geographic location of state matters
#KFC per capita	➜	Proliferation of fast food matters
…	➜	…

It **might** be correct, even if only incidentally!

Quick Recap — Lecture

- Clustering
  - Grouping data points based on their similarities
  - No single definition for cluster or clustering ➜ different meaningful intuitions
  - General-purpose data mining method ("only" distance/similarity measure required)

- Algorithms discussed so far:
  - K-Means (centroid-based, partitional, exclusive, complete)
  - DBSCAN (density-based, partitional, exclusive, partial)

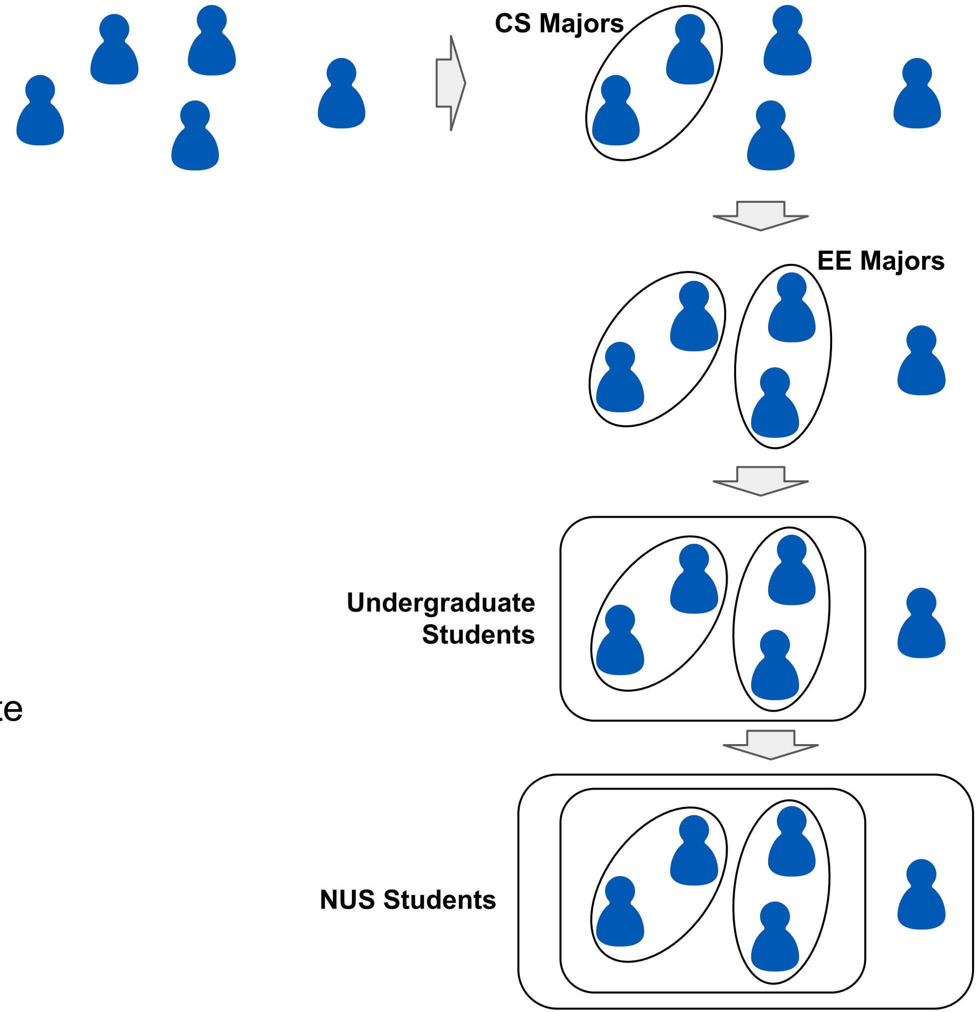
Outline

- Clustering
  - Overview
  - Concepts
  - Applications
- **Clustering algorithms**
  - K-Means
  - DBSCAN
  - **Hierarchical Clustering**
- Cluster Evaluation

Hierarchical Clustering

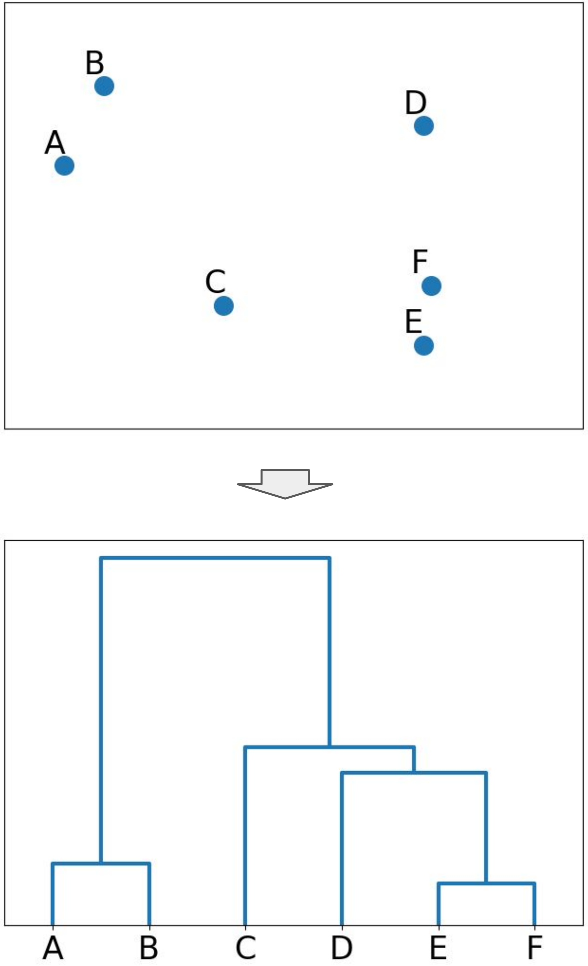
- Basic characteristics
  - Clusters: depends...
  - Clustering: hierarchical (duh!), complete, exclusive (at each level!)

- No parameterization (in principle)
  - In practice, typically number of clusters is specified (similar to K-means)
  - Different choices of measures to calculate distances between clusters



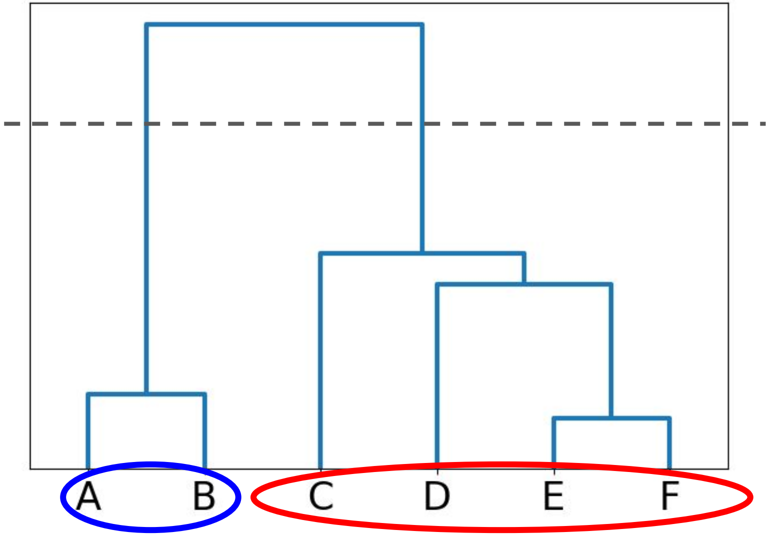
Dendrograms

- Dendrogram: Visualization of hierarchical relationships
  - Binary tree showing how clusters are hierarchically merged/split
  - Each node is a cluster
  - Each leaf is a singleton cluster
  - Height reflects distance between clusters
    (e.g., large distance between A/B and C/D/E/F clusters)

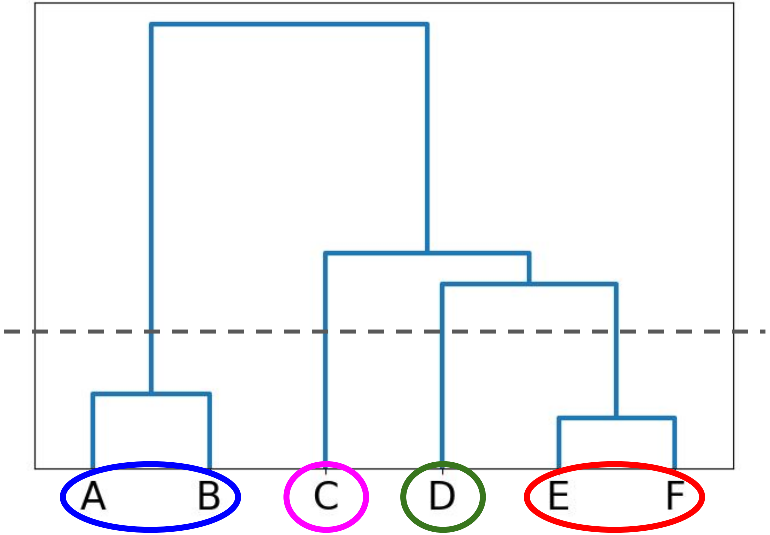


Hierarchical Clustering — Dendrograms

- A clustering can be obtained by cutting a dendrogram at the desired level



2 clusters

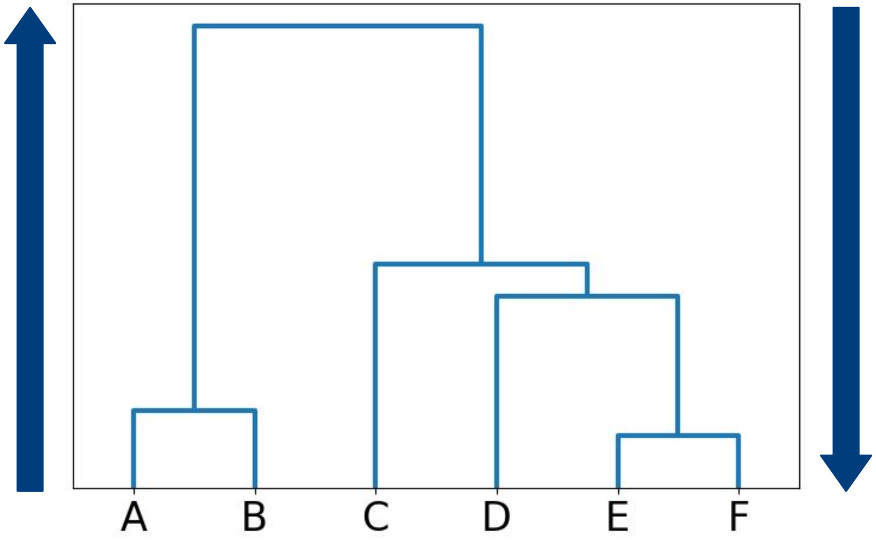


4 clusters

Hierarchical Clustering — 2 Main Types

Agglomerative
(bottom-up)

- Start with each point being its own cluster
- At each step, merge closest pair of clusters
- Stop when only one cluster is left



Divisive
(top-down)

- Start with one cluster containing all points
- At each step, split a cluster
- Stop when each cluster contains a single point

AGNES (AGglomerative NESting)

DIANA (DIvise ANAlysis)

Quick Quiz

AGNES — Basic Algorithm

1. Initialization: Each point forms its own cluster
2. Repeat

    2a) **Merge** the two closest clusters into one

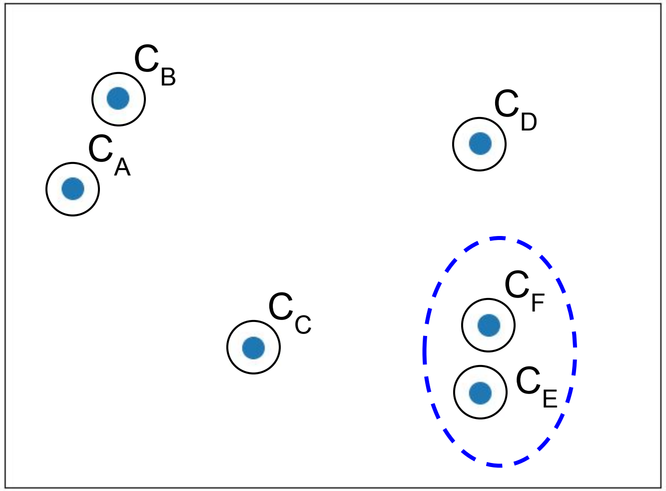
    Until only 1 cluster remains

Download at: https://www.comp.nus.edu.sg/~chris/downloads/images/agglomerative-clustering-visualized.gif

AGNES — Implementation

- Implementation using distance matrix

Initial clustering: each cluster, one point



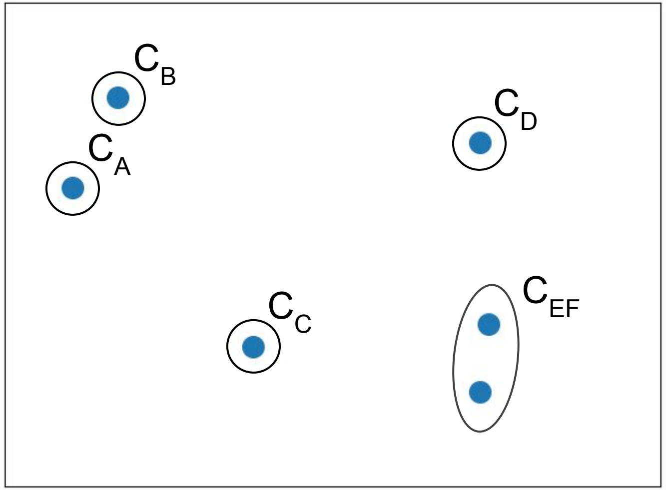
Distance between clusters = distance between points

	C_A	C_B	C_C	C_D	C_E	C_F
C_A	∞	2.25	5.32	9.06	9.79	9.49
C_B		∞	6.08	7.85	9.86	9.21
C_C			∞	6.73	4.81	5.02
C_D				∞	5.51	4.00
C_E					∞	**1.53**
C_F						∞

AGNES — Implementation

- What's the distance between clusters? (beyond containing single points)

Clustering after merging C_E and C_F to C_{EF}

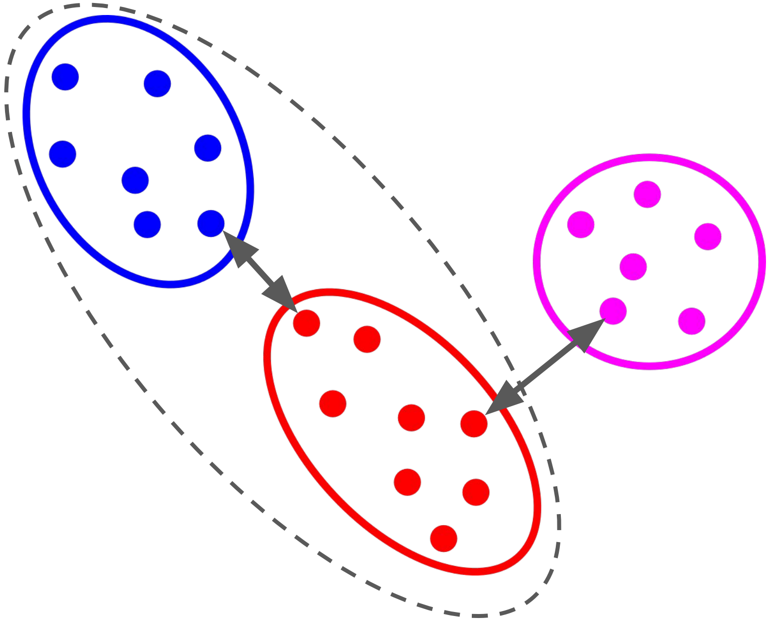


	C_A	C_B	C_C	C_D	C_{EF}
C_A	∞	2.25	5.32	9.06	???
C_B		∞	6.08	7.85	???
C_C			∞	6.73	???
C_D				∞	???
C_{EF}					∞

Quick "Quiz"

AGNES — Single Linkage

- Single Linkage Clustering
  - Distance between clusters = **minimum distance** between two points from each cluster



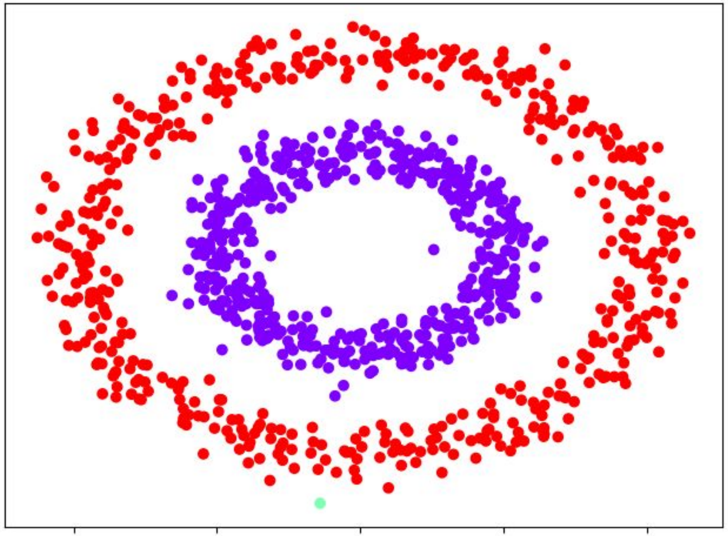
simple pointwise distance

$$d_{single}(C_i, C_j) = \min_{p \in C_i, q \in C_j} d(p, q)$$

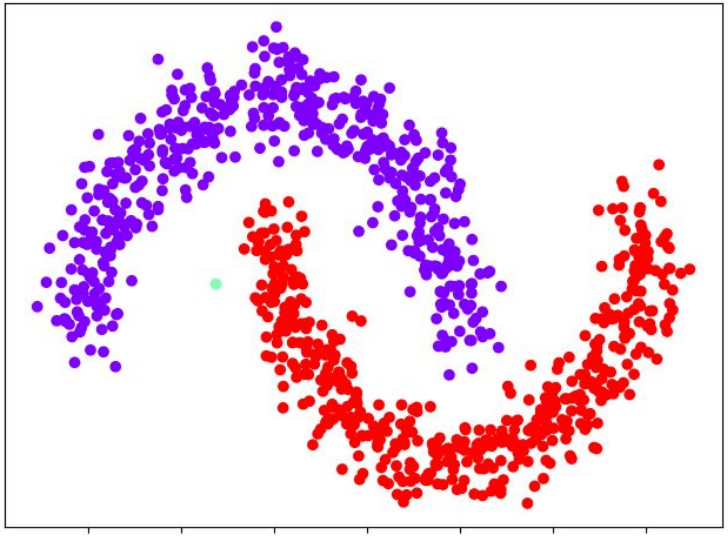
AGNES — Single Linkage

- Strength: Can handle non-globular shapes

#cluster = 3



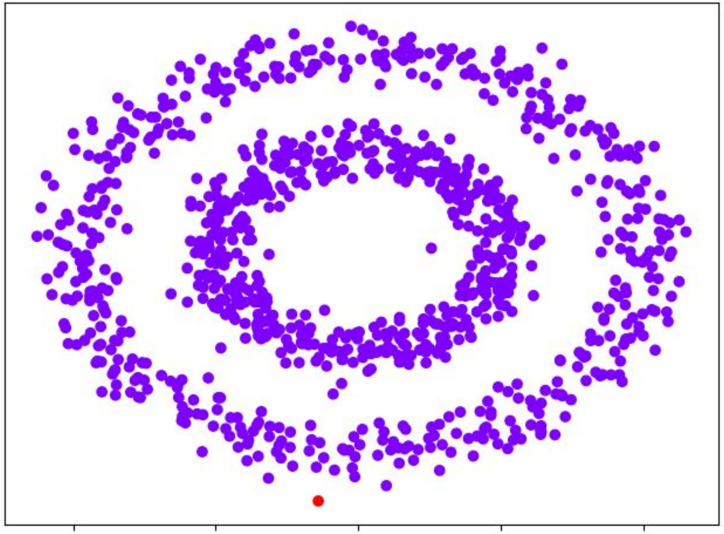
#cluster = 3



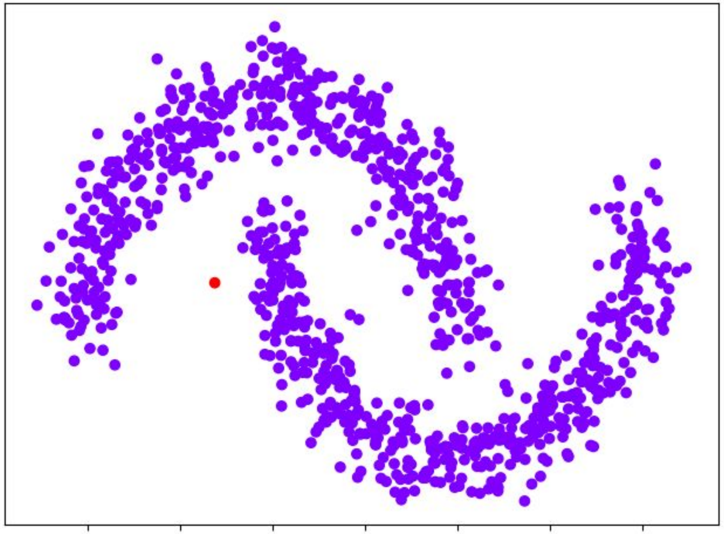
AGNES — Single Linkage

- Weakness: Very susceptible to noise ➜ "Chaining"
  - A single point may cause two clusters get merged

#cluster = 2

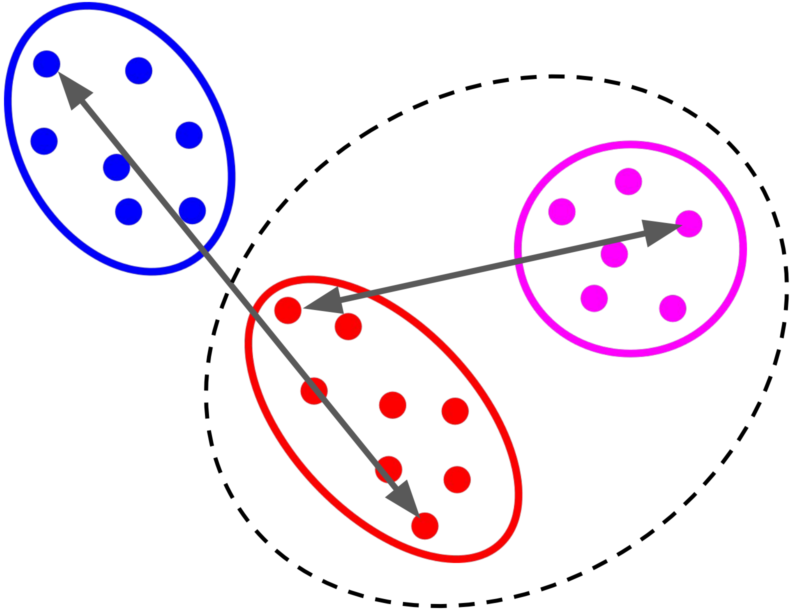


#cluster = 2



AGNES — Complete Linkage

- Complete Linkage Clustering
  - Distance between clusters = **maximum distance** between two points from each cluster

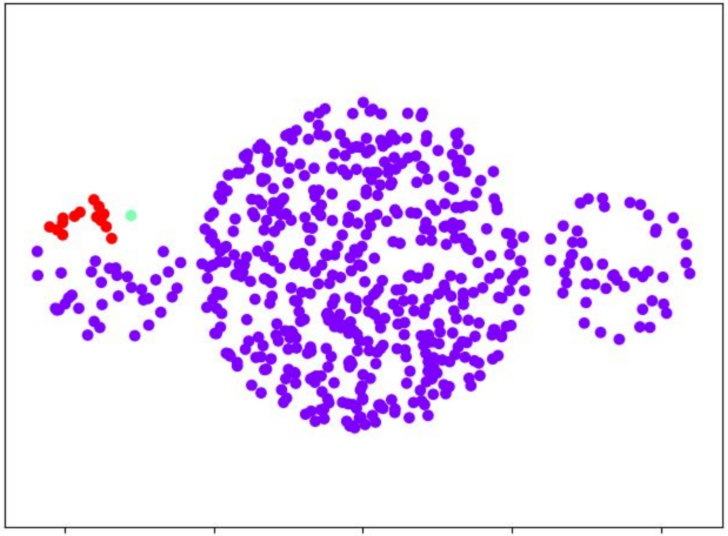


$$d_{complete}(C_i, C_j) = \max_{p \in C_i, q \in C_j} d(p, q)$$

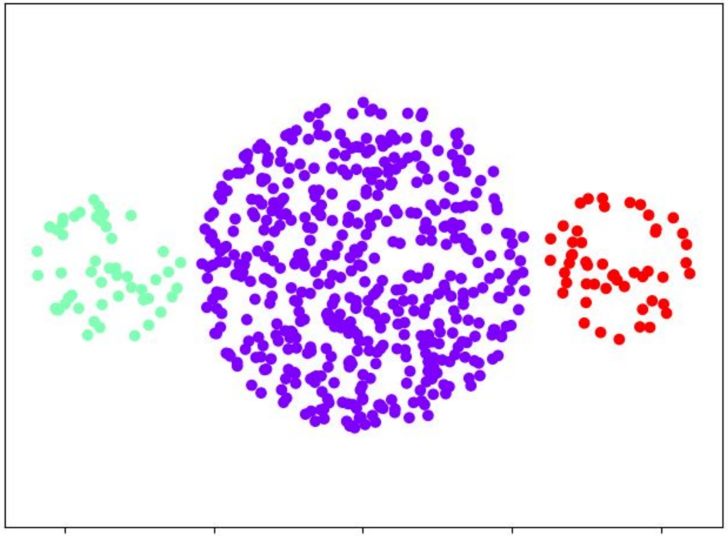
AGNES — Complete Linkage

- Strength: Less susceptible to noise or outliers

Single Linkage, #cluster = 3

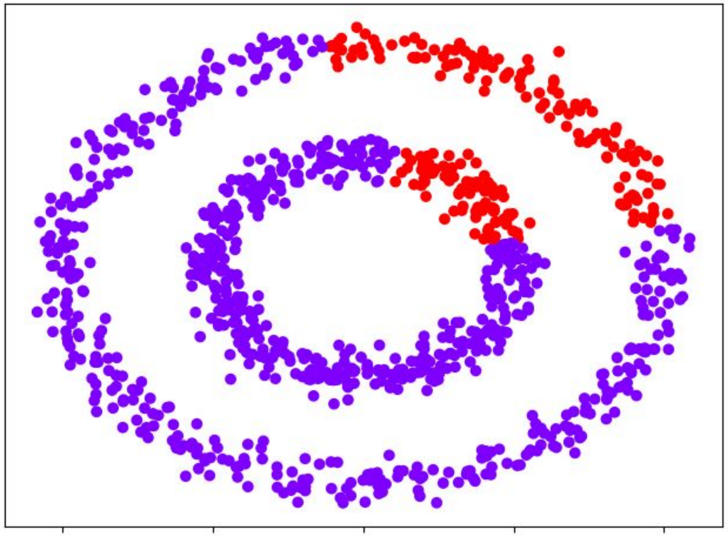


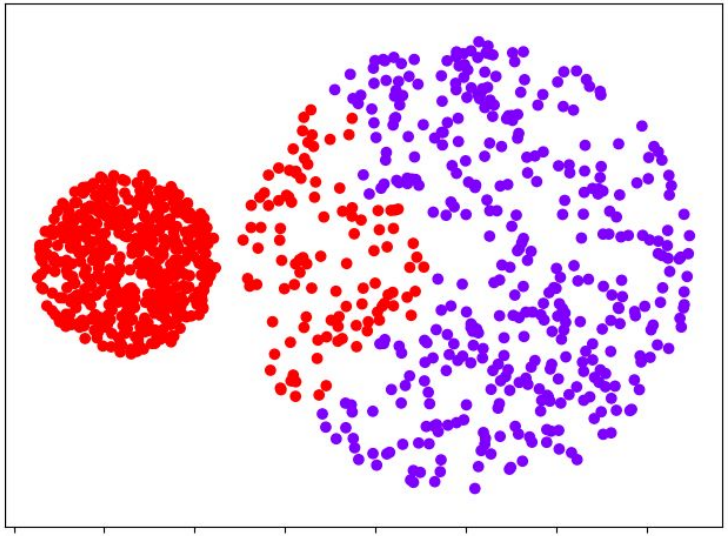
Complete Linkage, #cluster = 3



AGNES — Complete Linkage

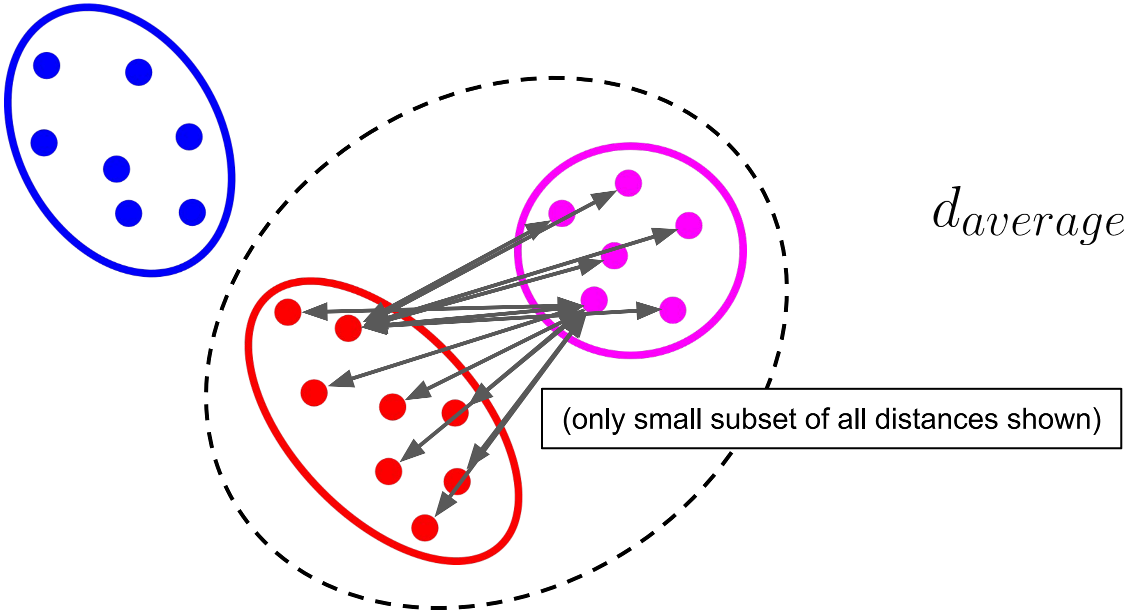
- Weaknesses
  - Bias towards globular clusters
  - Tends to break large clusters





AGNES — Average Linkage

- Complete Linkage Clustering (compromise between single and complete linkage)
  - Distance between clusters = **average distance** between two points from each cluster



$$d_{average}(C_i, C_j) = \underset{p \in C_i, q \in C_j}{avg} d(p,q)$$

AGNES — Linkage Alternatives

- Centroid linkage
  - Distance between clusters = distance between the centroids of each cluster

$$d_{centroid}(C_i, C_j) = d(\underbrace{m_i, m_j}_{\text{centroid of cluster } i \text{ and } j \text{ (m for mean)}})$$

- Ward linkage

$$d_{Ward}(C_i, C_j) = \overbrace{\sum_{k \in C_i \cup C_j} ||x_k - m_{ij}||^2}^{\text{Variance of } C_{ij}} - \overbrace{\sum_{k \in C_i} ||x_k - m_i||^2}^{\text{Variance of } C_i} - \overbrace{\sum_{k \in C_j} ||x_k - m_j||^2}^{\text{Variance of } C_j}$$

$$= \frac{n_i n_j}{n_i + n_j} ||m_i - m_j||^2$$

n_i = #points in cluster C_i

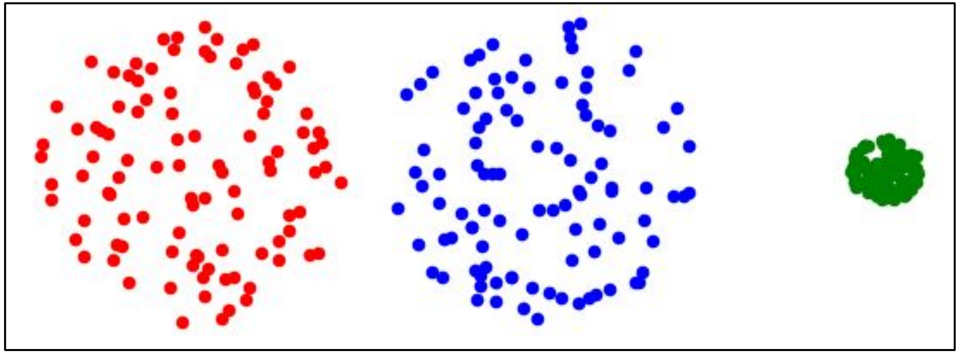
Ward Linkage — Intuition

$$d_{Ward}(C_i, C_j) = \overbrace{\sum_{k \in C_i \cup C_j} ||x_k - m_{ij}||^2}^{\text{Variance of } C_{ij}} - \overbrace{\sum_{k \in C_i} ||x_k - m_i||^2}^{\text{Variance of } C_i} - \overbrace{\sum_{k \in C_j} ||x_k - m_j||^2}^{\text{Variance of } C_j}$$

- Example for Ward Linkage
    - Each blob: 100 data points

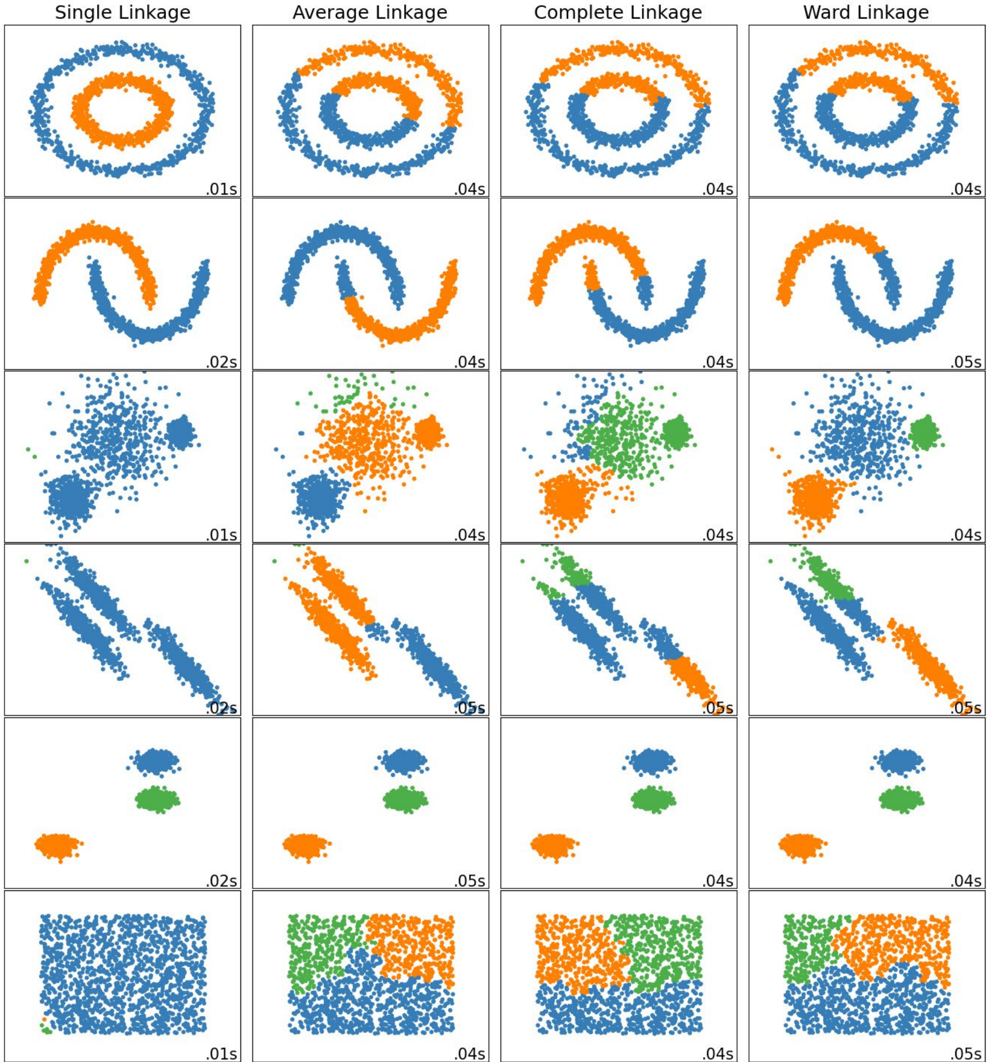
d_{Ward}(Red, Blue) = 1,635 - 195 - 200 = **1,240**

d_{Ward}(Blue, Green) = 1,450 - 200 - 10 = **1,240**



AGNES

- Linkage comparison



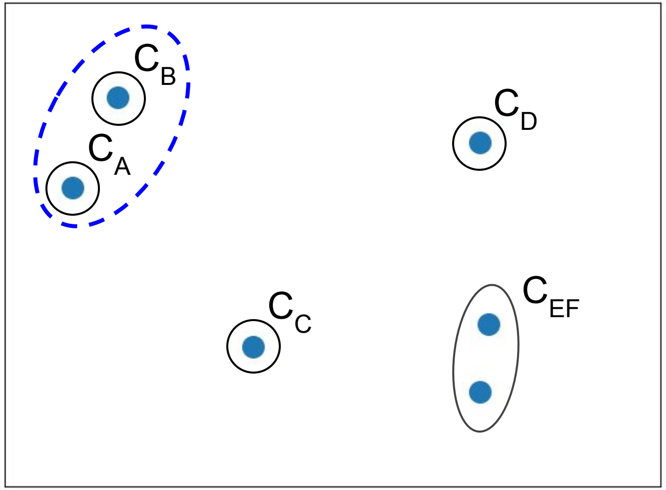
Source:
https://scikit-learn.org/stable/auto_examples/cluster/plot_linkage_comparison.html

Quick Quiz

AGNES — Implementation

- Distance matrix after merging Cluster C_E and C_F + Average Linkage

Clustering after merging C_E and C_F to C_{EF}

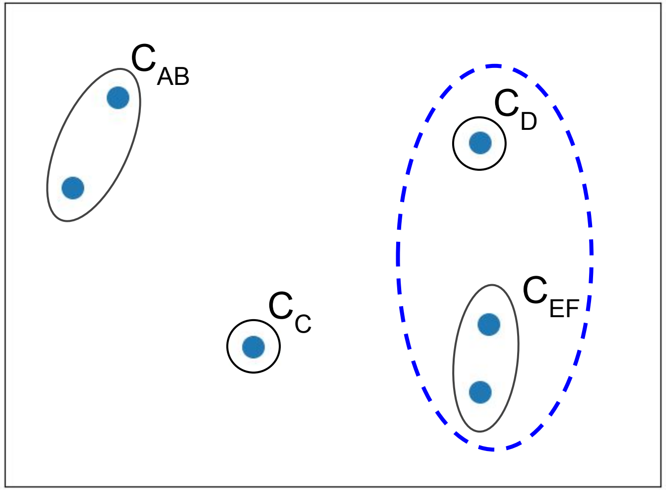


	C_A	C_B	C_C	C_D	C_{EF}
C_A	∞	**2.25**	5.32	9.06	9.64
C_B		∞	6.08	7.85	9.54
C_C			∞	6.73	4.92
C_D				∞	4.76
C_{EF}					∞

AGNES — Implementation

- Distance matrix after merging Cluster C_A and C_B + Average Linkage

Clustering after merging C_A and C_B to C_{AB}

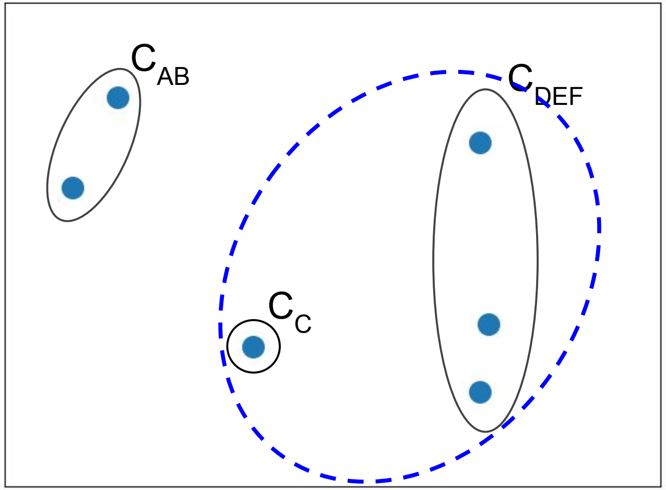


	C_{AB}	C_C	C_D	C_{EF}
C_{AB}	∞	5.70	8.45	9.59
C_C		∞	6.73	4.92
C_D			∞	**4.76**
C_{EF}				∞

AGNES — Implementation

- Distance matrix after merging Cluster C_C and C_{DEF} + Average Linkage

Clustering after merging C_D and C_{EF} to C_{DEF}

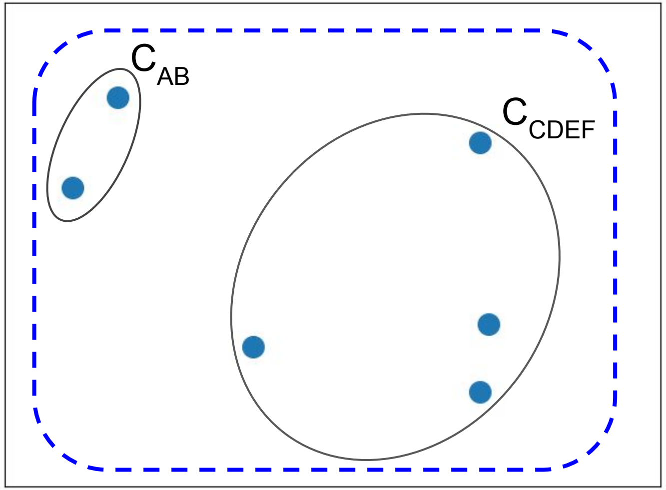


	C_{AB}	C_C	C_{DEF}
C_{AB}	∞	5.70	9.21
C_C		∞	**5.51**
C_{DEF}			∞

AGNES — Implementation

- Distance matrix after merging Cluster C_{AB} and C_{CDEF} + Average Linkage

Clustering after merging C_C and C_{DEF} to C_{CDEF}

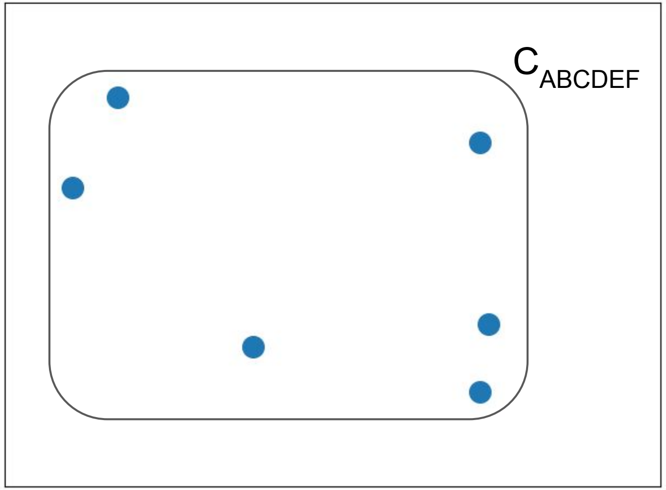


	C_{AB}	C_{CDEF}
C_{CDEF}	∞	**8.33**

AGNES — Implementation

- Distance matrix after merging Cluster C_{AB} and C_{CDEF} + Average Linkage

Clustering after merging C_{AB} and C_{CDEF} to C_{ABCDEF}



	C_{ABCDEF}
C_{ABCDEF}	∞

➜ Done!

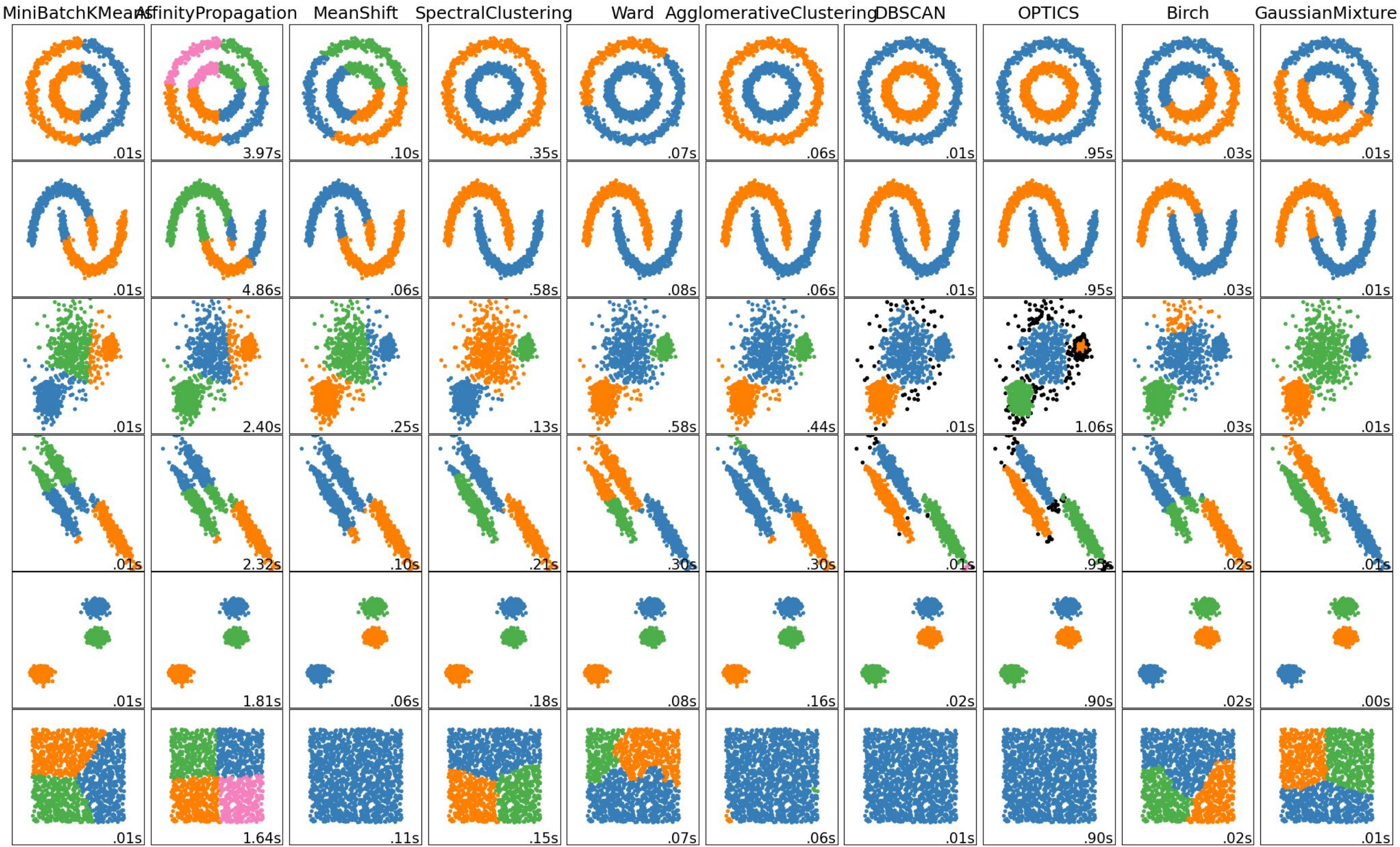
AGNES — Complexity Analysis

- Space Complexity: **$O(N^2)$**
  - Storing distance matrix

- Time Complexity
  - Baseline: **$O(N^3)$** — (N-1) steps, each step $O(N^2)$ to scan distance matrix
  - Using more sophisticated data structures, e.g, heap or priority queue: **$O(N^2 \log N)$**
  - Special optimization for Single Linkage Clustering: **$O(N^2)$**

DIANA — DIvisive ANAlysis

- Top-Down Hierarchical Clustering
  - Start with all points forming one cluster
  - Recursively split one cluster until all cluster have size 1
- Challenge: 2^n ways to split a cluster with *n* points
  - Heuristics needed to restrict search space
  - Generally slower and less common than AGNES
- Cases where DIANA can perform better
  - No complete clustering needed ➜ early stopping
  - Splitting can utilize global knowledge (merging based on local knowledge only)



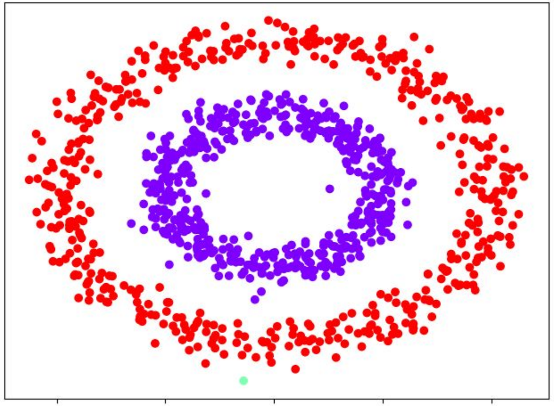
Source: https://scikit-learn.org/stable/modules/clustering.html

Outline

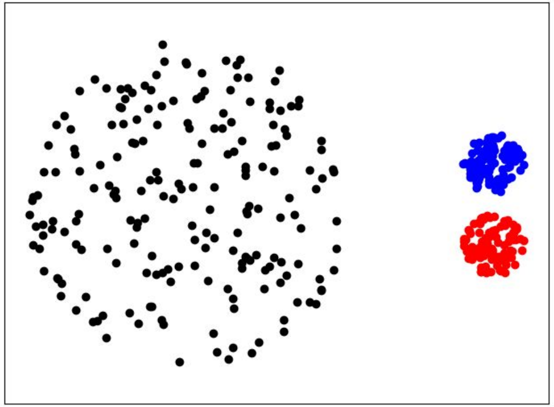
- Clustering
  - Overview
  - Concepts
  - Applications
- Clustering algorithms
  - K-Means
  - DBSCAN
  - Hierarchical Clustering
- **Cluster Evaluation**

Cluster Evaluation

- Problem 1: Just eyeballing the clustering is rarely possible
  - High-dimensional data (≥ 3 dimensions) difficult to impossible to visualize
  - Difficult to assess "nature" of clusters a-priori (e.g., variations in shape, size, density, etc)
  - Presence and distribution of noise or outliers



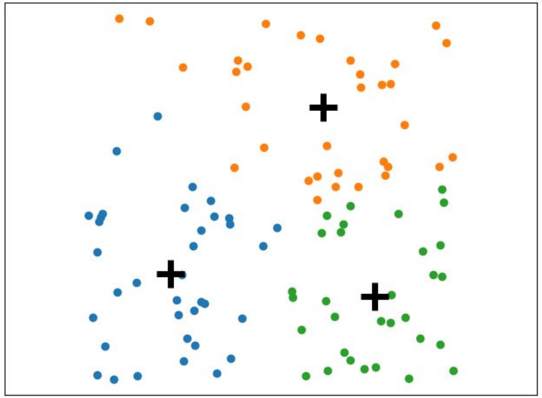
Your data usually does not look like this



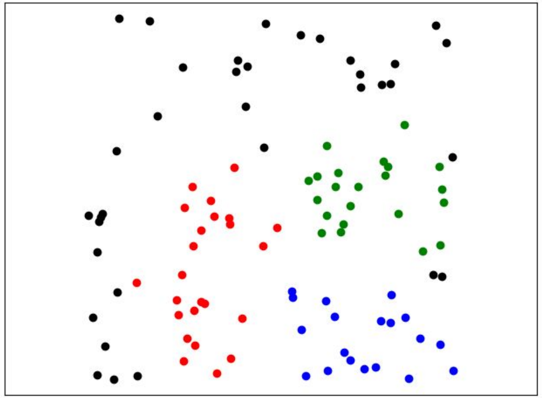
Cluster Evaluation

- Problem 2: Clustering algorithms will always find some cluster
  - Example: K-Means, DBSCAN and AGNES applied to random data

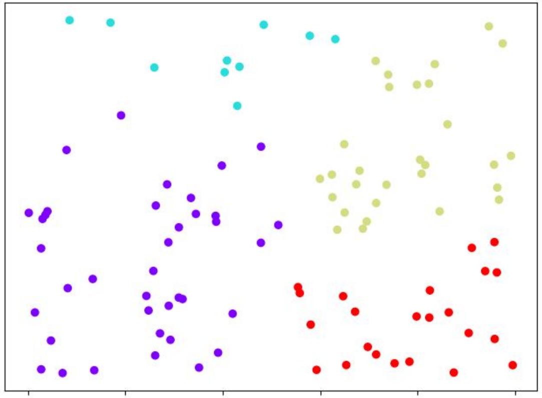
K-Means



DBSCAN



AGNES



Cluster Evaluation

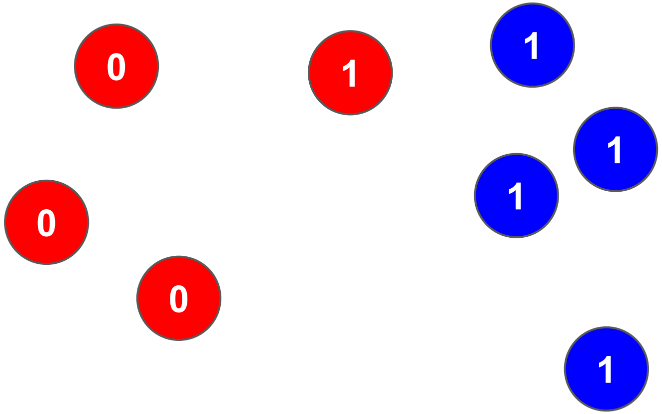
- Purpose of cluster evaluation
  - Comparing the results of different clustering algorithms
  - Comparing the results of a clustering algorithm with different parameters
  - Minimizing the effects of noise on the clustering

➜ Getting a sense of the "goodness" of a clustering

- Two main approaches
  - External quality measures: evaluate a clustering against a ground truth (if available)
  - **Internal quality measures**: evaluate clustering from the data itself

Cluster Evaluation — External Quality Measures

- Ground truth: Labeled data
  - Labels indicate that two points "belong together"
  - If cluster reflect this ➜ good clustering



Cluster	Label
Red	0
Red	0
Red	0
Red	1
Blue	1
Blue	1
Blue	1
Blue	1

External Quality Measures — Cluster Purity

- Cluster purity P
  - N: #points, C: set of cluster, L: set of labels

$$P = \frac{1}{N} \sum_{c \in C} \overbrace{\max_{l \in L} |c \cap l|}^{\text{\#points with most common label } l \text{ in cluster } c}$$

Purity for example:

$$P = \frac{1}{8}(3+4) = 0.875$$

Cluster	Label
Red	0
Red	0
Red	0
Red	1
Blue	1
Blue	1
Blue	1
Blue	1

- Limitations
  - Purity does not penalize having many cluster
    ➜ P=1 easy to achieve with all cluster containing single point

External Quality Measures: Information Retrieval Metrics

- Established metrics from classification tasks
  - **TP** — true positives
    same cluster, same label
    (A/B, A/C, B/C, E/F, …, G/H)
  - **TN** — true negatives
    different clusters, different labels
    (A/E, A/F, A/G, A/H, B/E, …, C/H)
  - **FP** — false positives
    same cluster, different labels
    (A/D, B/D, C/D)
  - **FN** — false negatives
    different cluster, same label
    (D/E, D/F, D/G, D/H)

For the example:
- **TP** = 9
- **TN** = 12
- **FP** = 3
- **FN** = 4

ID	Custer	Label
A	Red	0
B	Red	0
C	Red	0
D	Red	1
E	Blue	1
F	Blue	1
G	Blue	1
H	Blue	1

External Quality Measures — Information Retrieval Metrics

- Rand Index RI
  - Reflects accuracy

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

$$\boxed{RI_{example} = 0.75}$$

- Precision P, Recall R, F1-Score

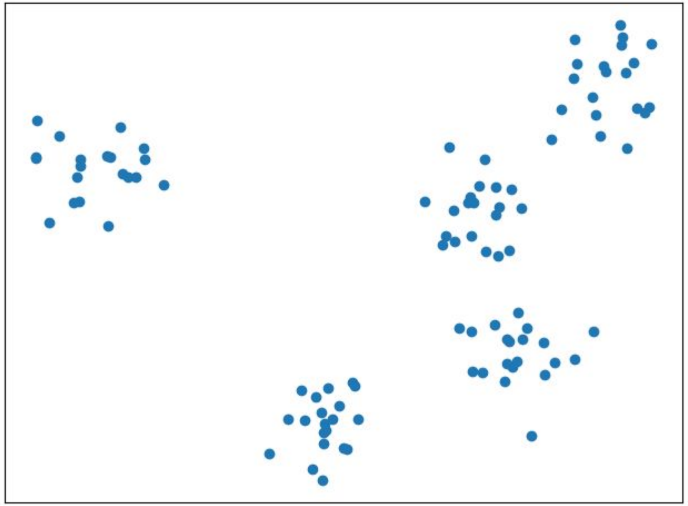
$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \qquad F1 = \frac{2 \cdot P \cdot R}{P + R}$$

$$\boxed{P_{example} = 0.75 \qquad R_{example} = 0.69 \qquad F1_{example} = 0.72}$$

- ...and others using TP, TN, FP, FN

Internal Quality Measures — SSE

- Use SSE to select number of clusters

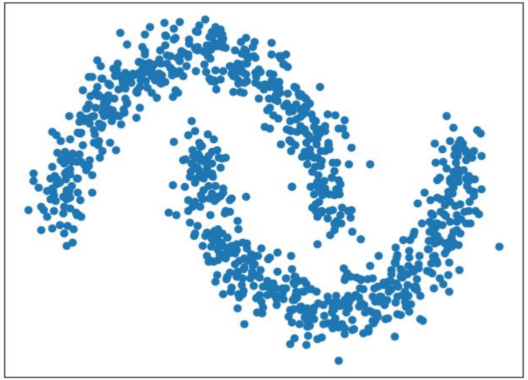


input data

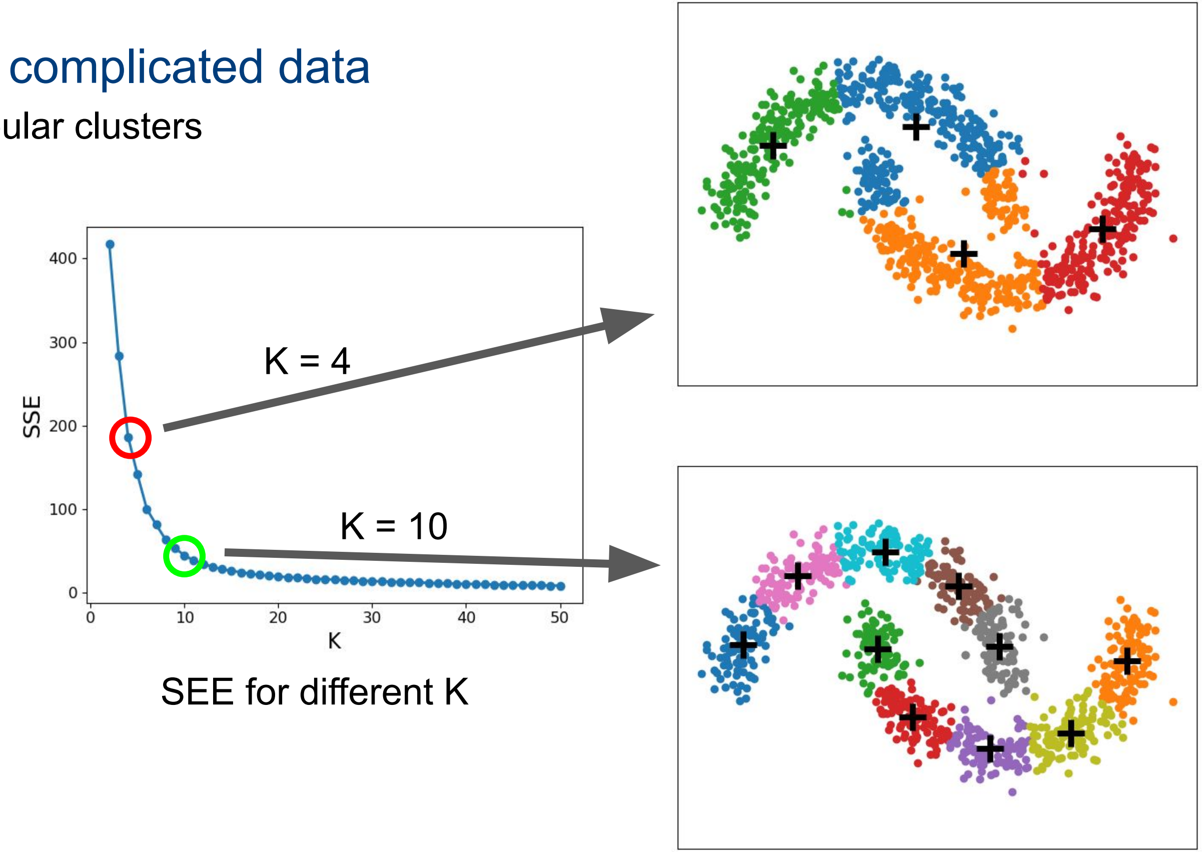
SSE for different K

Internal Quality Measures — SSE

- Also applicable to more complicated data
  - But inherently "favors" globular clusters



input data

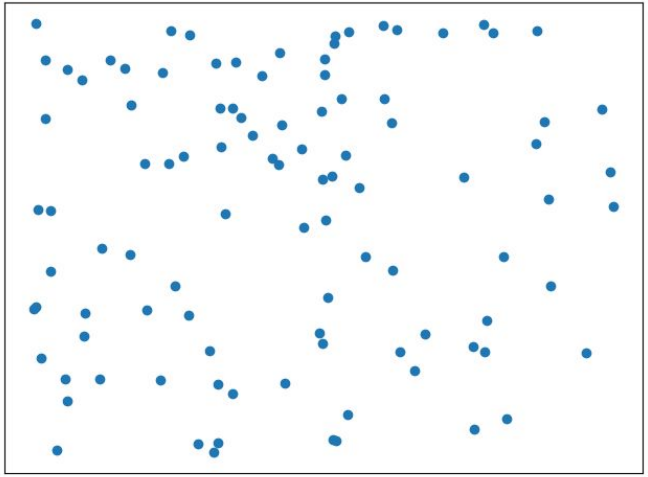


SEE for different K

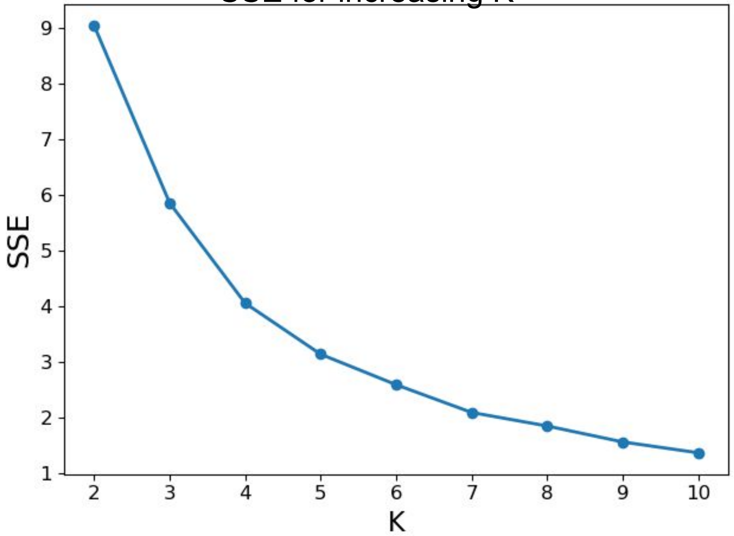
Internal Quality Measures — SSE

- Limitation of SSE as quality measure
  - SSE does not penalize large number of clusters
  - SSE decreases for increasing cluster counts
  - Applicable beyond K-Means, but less intuitive interpretation (in case of non-globular clusters)

random noise



SSE for increasing K



Quick Quiz

Internal Quality Measures — Silhouette Coefficient

- Intuition: A good clustering has
  - High inter-cluster distances
  - Low intra-cluster distances

- For each data point x, define

$x \in C_X$

  - **Cohesion** $a(x)$: average distance to points in the same cluster

$$a(x) = \frac{1}{|C_X - 1|} \sum_{p \in C_X, p \neq x} d(x, p)$$

the smaller, the better

  - **Separation** $b(x)$: minimum average distance to points in a different cluster

$$b(x) = \min_{X \neq K} \frac{1}{|C_K|} \sum_{p \in C_K} d(x, p)$$

the larger, the better

  - **Silhouette**:

$$s(x) = \frac{b(x) - a(x)}{max\{a(x), b(x)\}}, \text{ if } |C_x| > 1$$

$$s(x) = 0, \text{ if } |C_x| = 1$$

Internal Quality Measures — Silhouette Coefficient

$$s(x) = \frac{b(x) - a(x)}{max\{a(x), b(x)\}}$$

- Interpretation

$$-1 \leq s(x) \leq +1$$

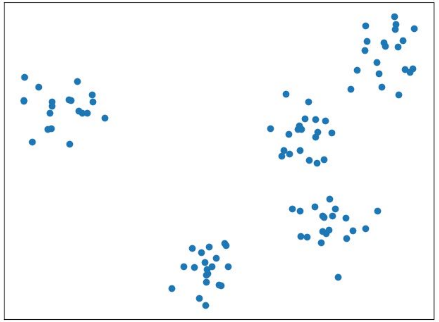
BAD GOOD

- Silhouette Coefficient SC:

$$SC = \frac{1}{N} \sum_{i=1}^{N} s(x_i)$$

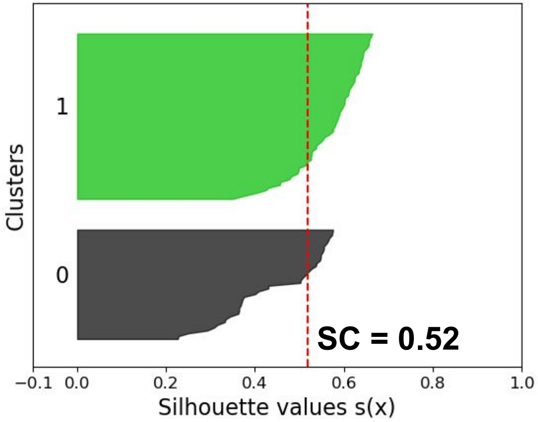
Internal Quality Measures — Silhouette Coefficient

- Example: K-Means

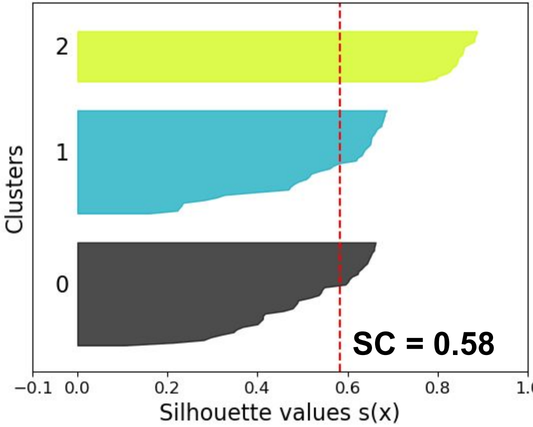


input data
(100 data points)

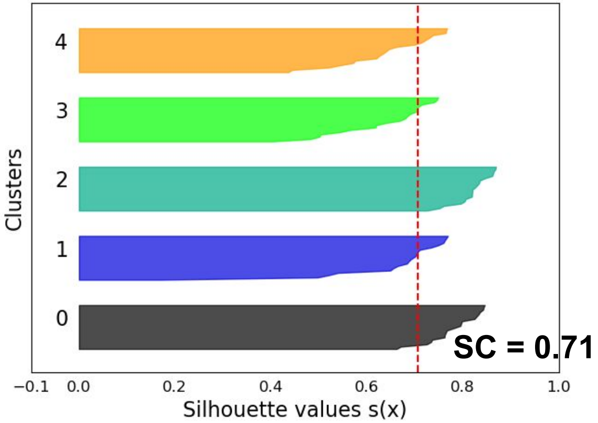
K = 2



K = 3



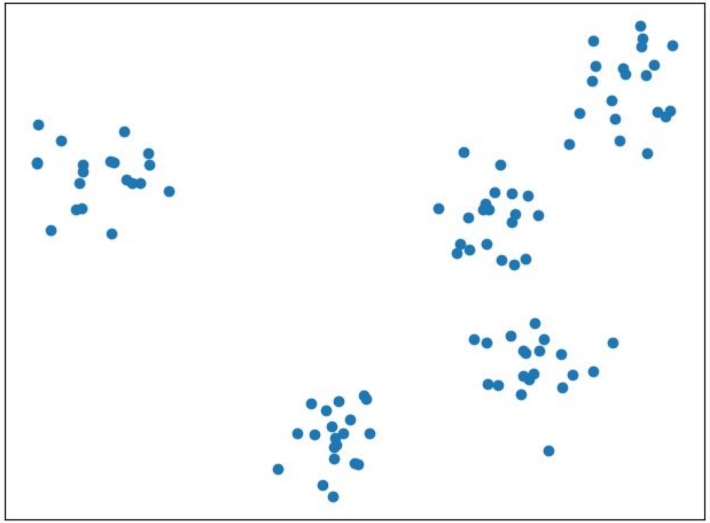
K = 5



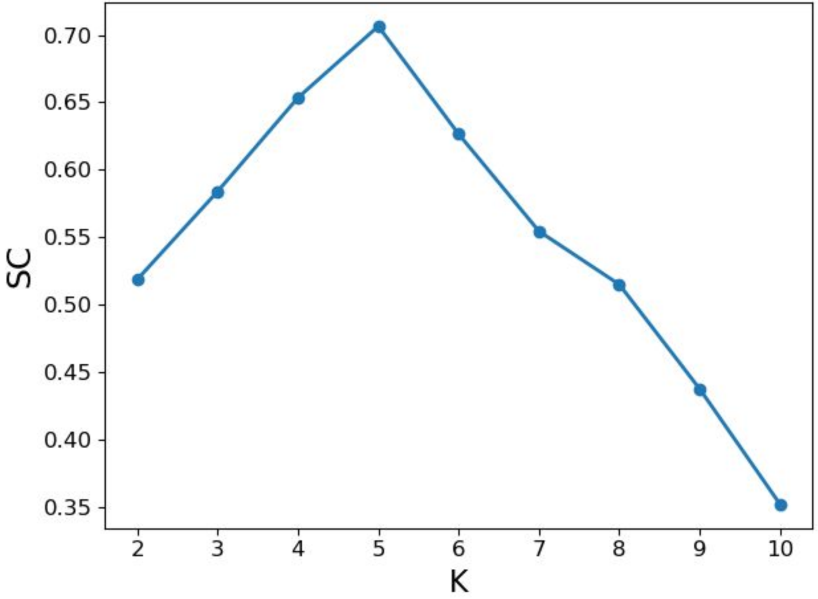
Internal Quality Measures — Silhouette Coefficient

- Example: K-Means

input data



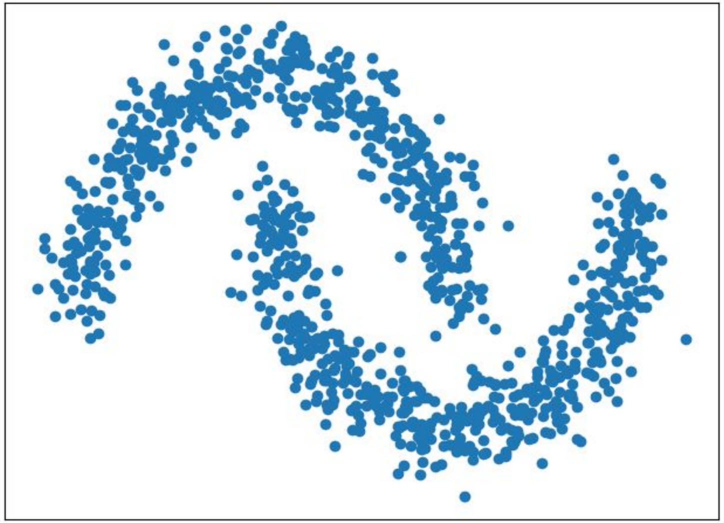
Silhouette Coefficients



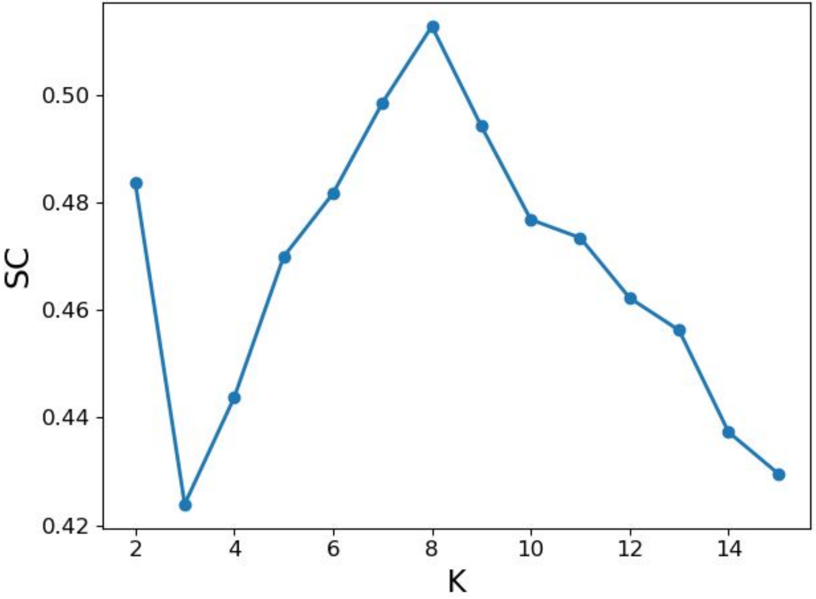
Internal Quality Measures — Silhouette Coefficient

- Example: K-Means

input data



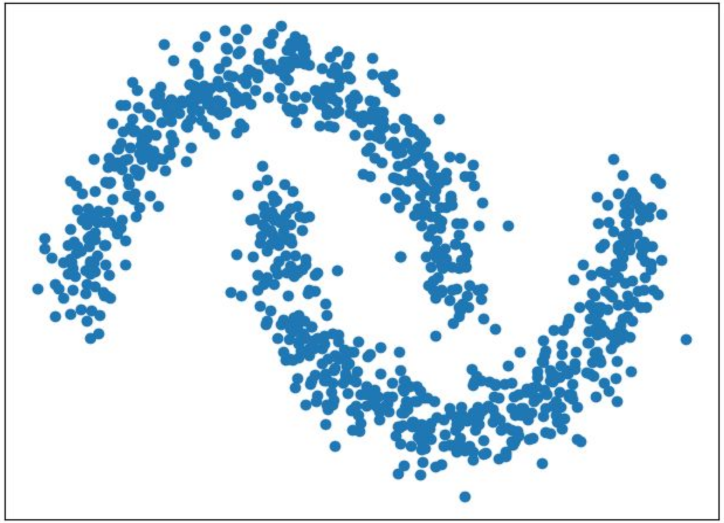
Silhouette Coefficients



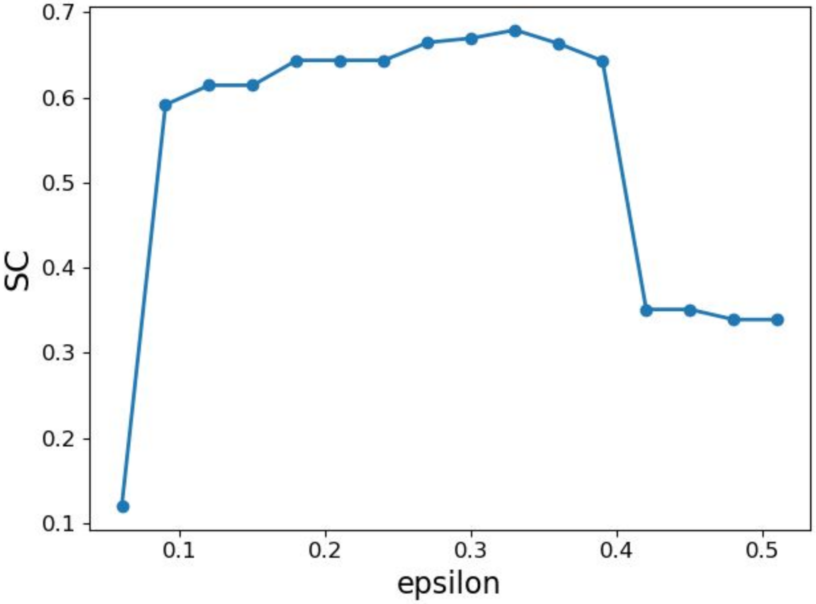
Internal Quality Measures — Silhouette Coefficient

● Example: DBSCAN

input data

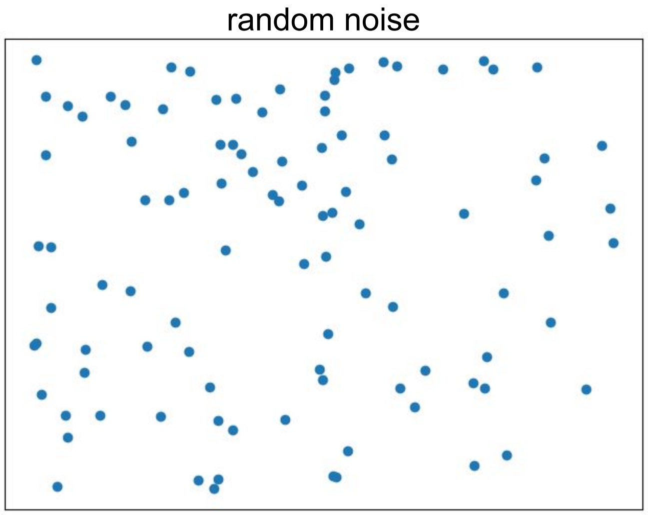


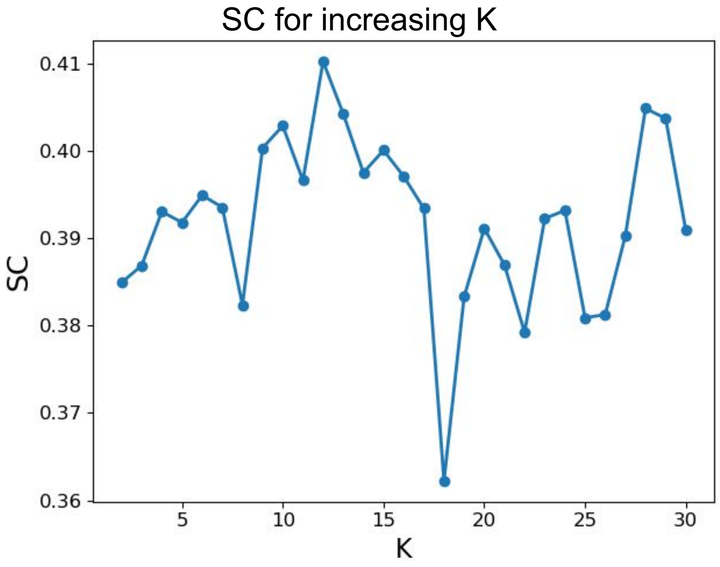
Silhouette Coefficients



Internal Quality Measures — Silhouette Coefficient

- SC for random data (K-Means)





Note: DBSCAN on random data quickly results in 0 or 1 cluster, for which SC is not defined

Cluster Evaluation — Comments

- In practice, choice of "best" clustering often more pragmatic:
  - Fixed number of clusters (problematic for DBSCAN)
  - Parameters defined by tasks
    (e.g., "areas with more than 5 McDonalds within a radius of 500m")
  - Maximum, minimum, or average size of clusters
  - Focus in individual clusters instead of whole clustering
    (e.g., biggest/smallest cluster, cluster that contains certain points)
  - Set *K* "too high" and merge later if needed
  - ...

Outline

- Clustering
  - Overview
  - Concepts
  - Applications
- Clustering algorithms
  - K-Means
  - DBSCAN
  - Hierarchical Clustering
- Cluster Evaluation

Summary — Clustering

- Clustering: Finding patterns (here: cluster/groups) in unlabeled data
  - Very important concept in data mining
  - Wide range of clustering algorithms with varying characteristics (pros & cons)

  ➜ **No "one-size-fits-all" algorithm**

- Discussed algorithms: K-Means, DBSCAN, AGNES
  - Focus on the — arguably intuitive — conceptual inner workings
  - Emphasis on algorithms' strength and weaknesses
  - Many tweaks and optimizations to improve performance

- Major challenge: cluster evaluation
  - No fool-proof method to find the best algorithm or parameters (at least for unlabeled data)

Solutions to Quick Quizzes