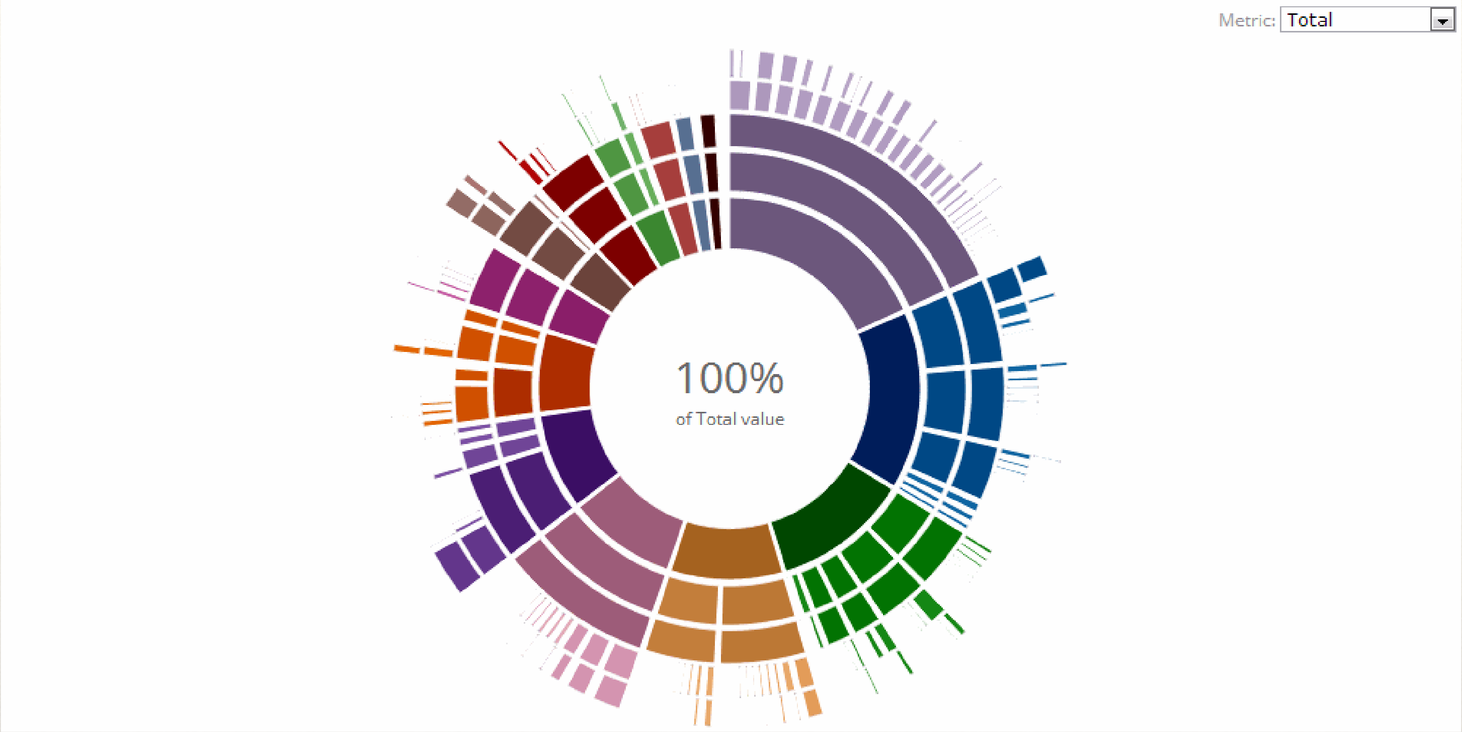
What is the origin of the name Python (programming language)

About me

YANDEX SCHOOL OF DATA SCIENCE



Y-Data: Lecture 2

Clustering methods - Cont.

Guy Shtar
goolig@gmail.com

Slides by:
Segev Arbiv, segevarbiv@gmail.com

Now that you know what clustering is, here is a joke:

Why did the clustering algorithm break up with its significant other?

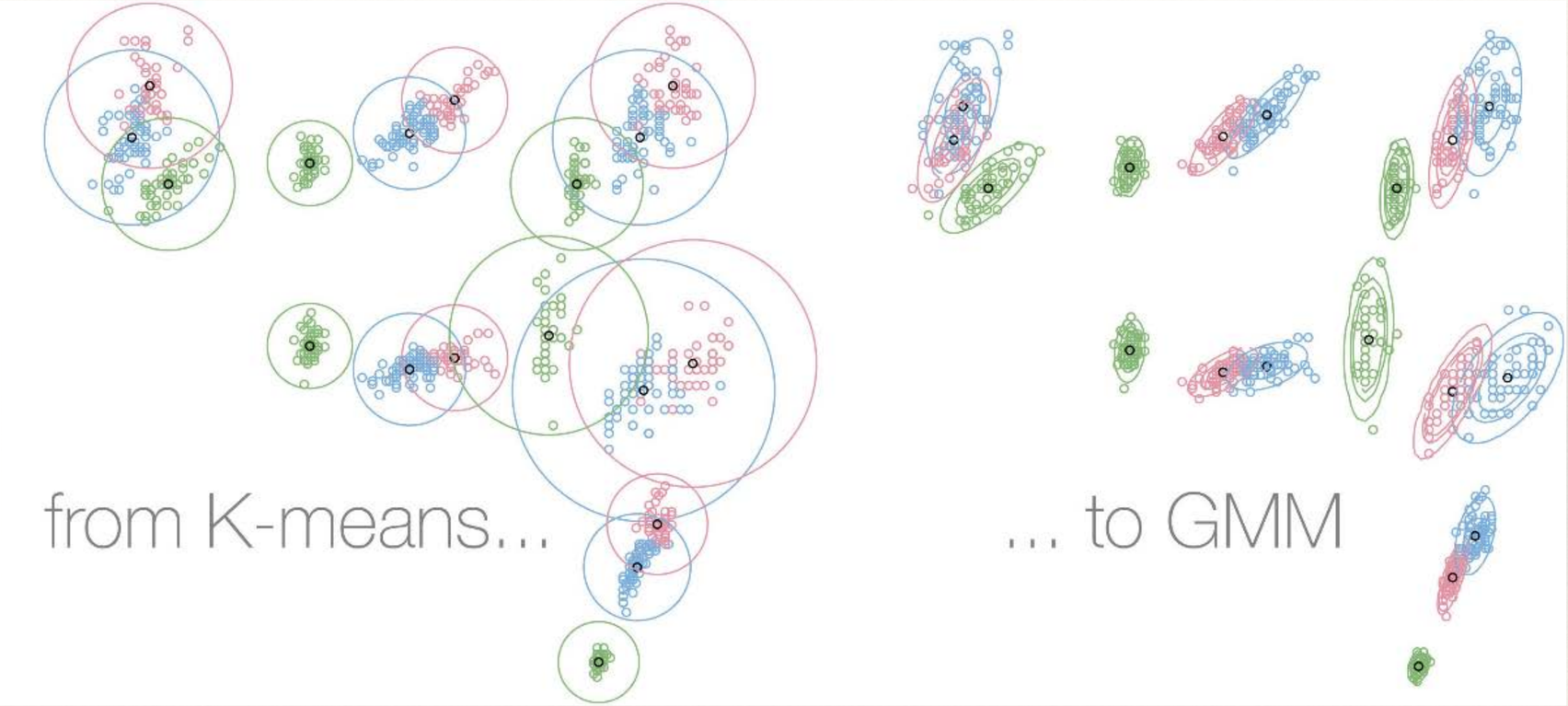
Because it couldn't handle the cluster of emotions.

Agenda

- GMM
- DBSCAN
  - Basic and improvements
- Hierarchical clustering
  - Agglomerative
  - Divisive
- External evaluation

Gaussian Mixture Model

Gaussian Mixture Model



Gaussian Mixture Model

- Gaussian Mixture Model (GMM) is a statistical model which assumes that the data distribution can be approximated with a set of gaussians
- GMM allows for *soft assignment*
- Generative model

Gaussian Mixture Model

A two step process:

1. Select a Gaussian (j)
2. Draw a sample from j's distribution

Namely, we will model the data as follows:

$$p(x;\theta) = \sum_{j=1}^{k} \phi_j N(x;\mu_j,\sigma_j)$$

Where:

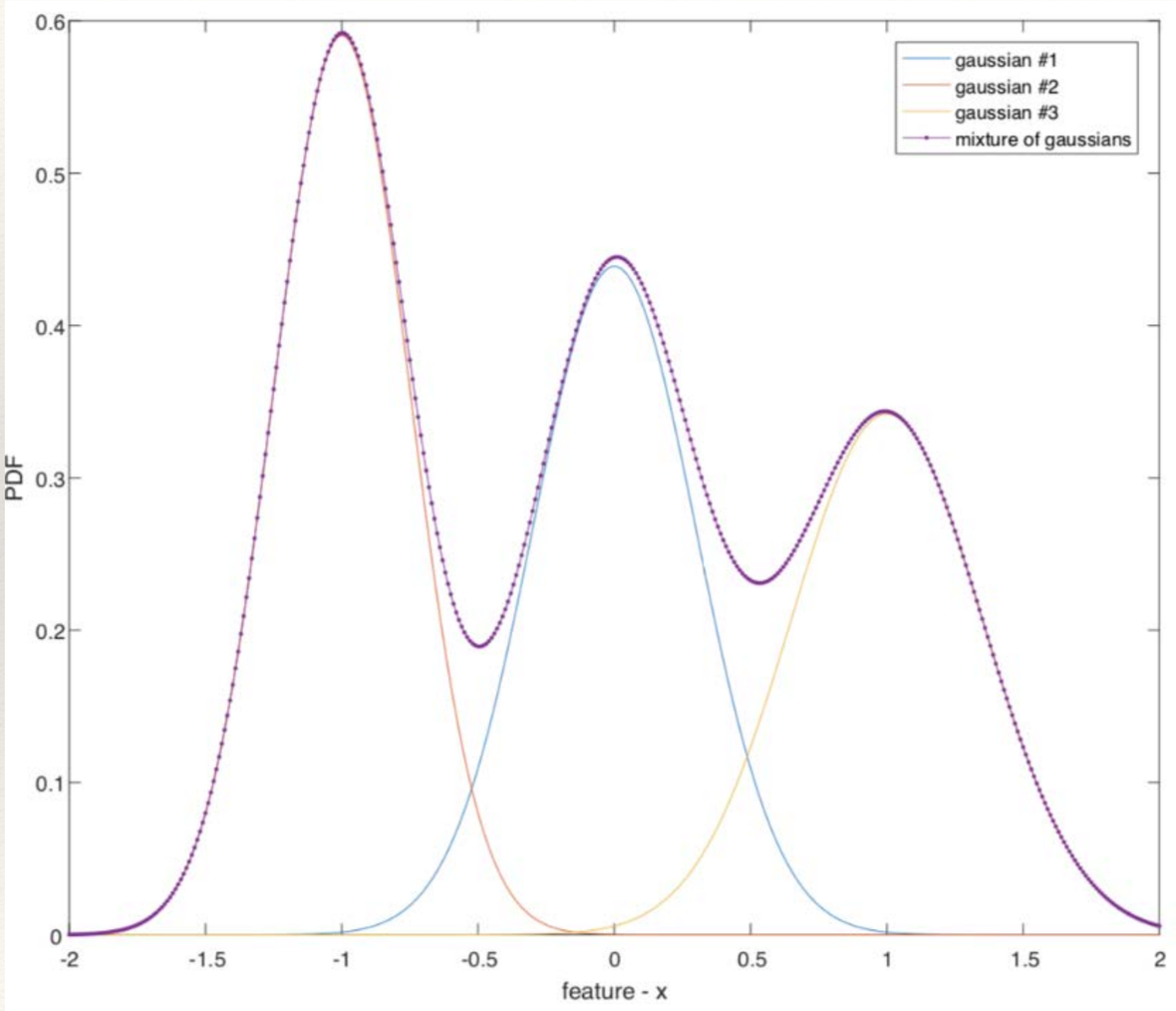
$$N(x;\mu,\sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

$$\phi_j \geq 0, \sum_j \phi_j = 1 \qquad \theta = \{\phi_j, \mu_j, \Sigma_j\}_{j=1}^{k}$$

GMM in short

- Closed from differentiation of mathematical formulation is not possible
- Common approaches to tackle this problem uses an iterative and alternating algorithms, such as EM (Expectation-Maximization)
  - Details in slides for the curious ones

Gaussian Mixture Model



Gaussian Mixture Model
EM

- Pros
  - Often computationally simple
  - Learns the latent structure of the data
  - Deals with non-radial shapes and varying densities
- Cons
  - Converges to a local optimum
  - Need to specify K
  - Not efficient for non-convex shapes

DBSCAN

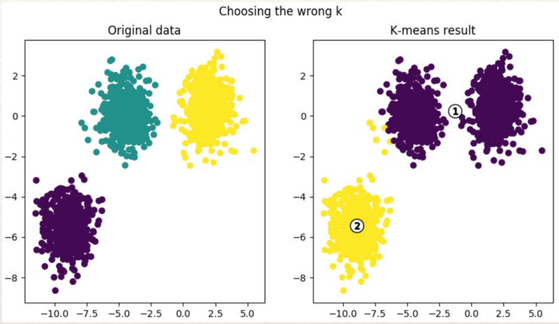
Density-Based Spatial Clustering of Applications with Noise

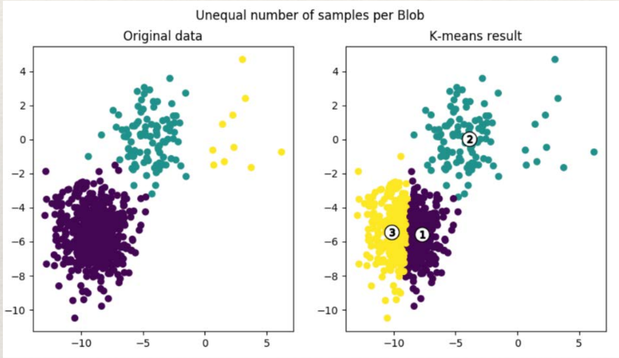
https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf (3 years ago: 16,413 citations)
2 years ago: 19,400. Last year: 22,915.
This year: 26,595

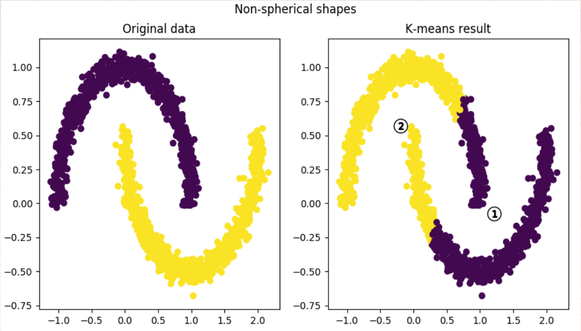
DBSCAN

Disadvantages of "simple" clustering algorithms

- As we spoke, K-Means have some drawbacks:
  - Need to choose k
  - Sensitive to cluster shape
  - Sensitive to starting conditions
  - No robustness to noise
  - Computationally expensive
  - Most are true also for GMM







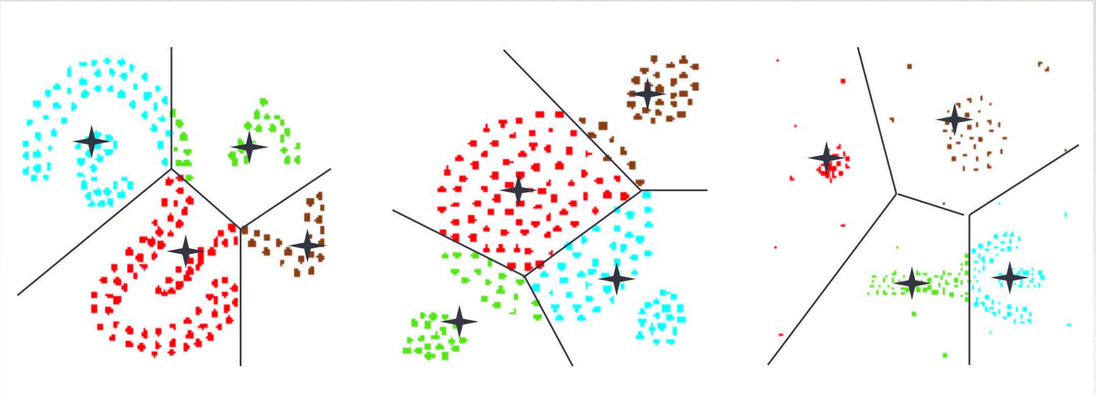
Notebook time

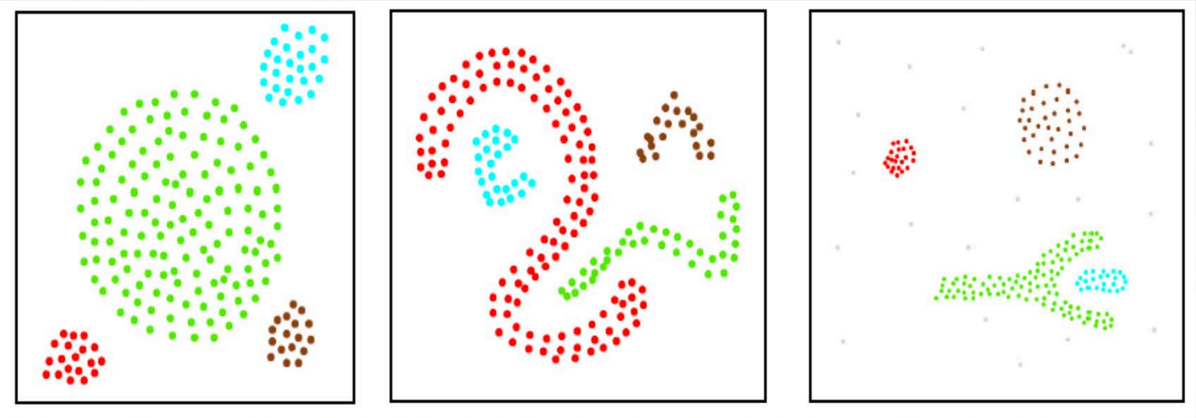
Here

DBSCAN

- Locates regions in a density-based manner
- Can discover clusters of arbitrary shapes
- Non-parametric
- Handles outliers
- Was awarded the “Test of time award” in KDD, 2014

?



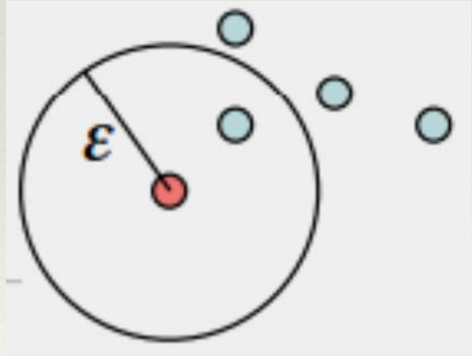


DBSCAN

- Defines density of a point p by the model 2 parameters:
  - ϵ - radius
  - *MinPts* - the desired minimum cluster size
- Neighborhood of p:

d – distance function

$$N_\epsilon(p) = \{q \in D | d(p,q) \leq \epsilon\}$$



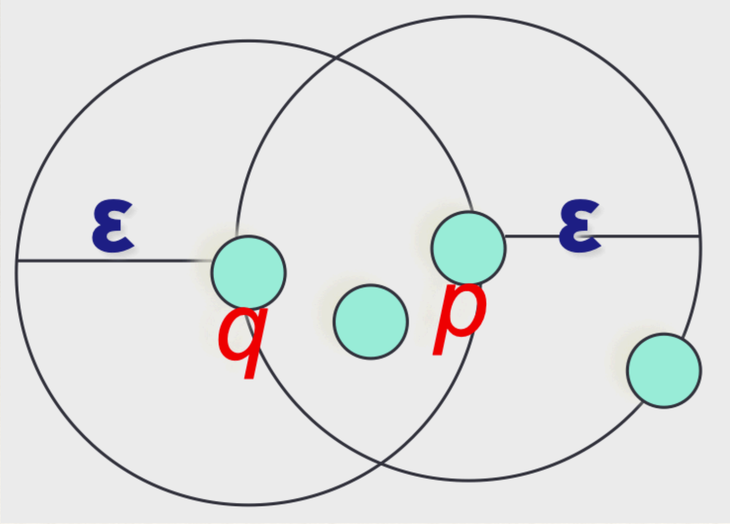
DBSCAN
Definitions

- A **core point** has at least *MinPts* in its neighborhood:

$$|N_\epsilon(p)| \geq MinPts$$

MinPts = 4

?



DBSCAN
Definitions

- A **core point** have at least *MinPts* in its neighbourhood:

$$|N_\epsilon(p)| \geq MinPts$$

- A **border point** has fewer points than *MinPts* in its neighborhood, but it lies in the neighborhood of another core point.
- **Noise point** is neither of the above.

MinPts = 5

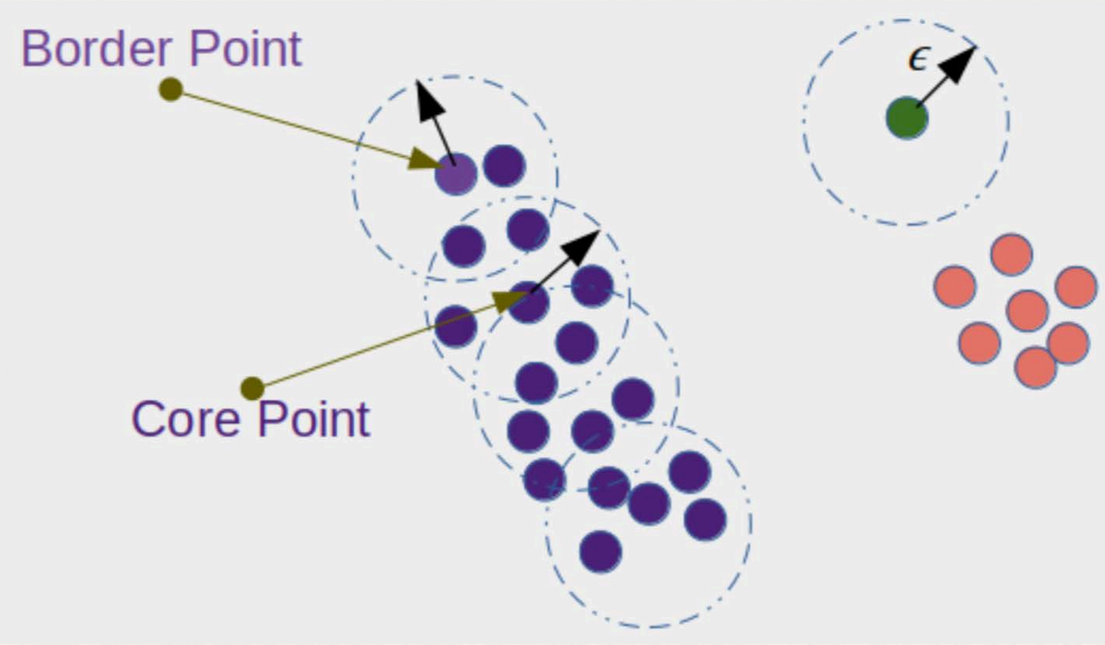
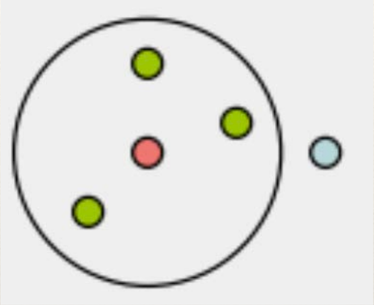


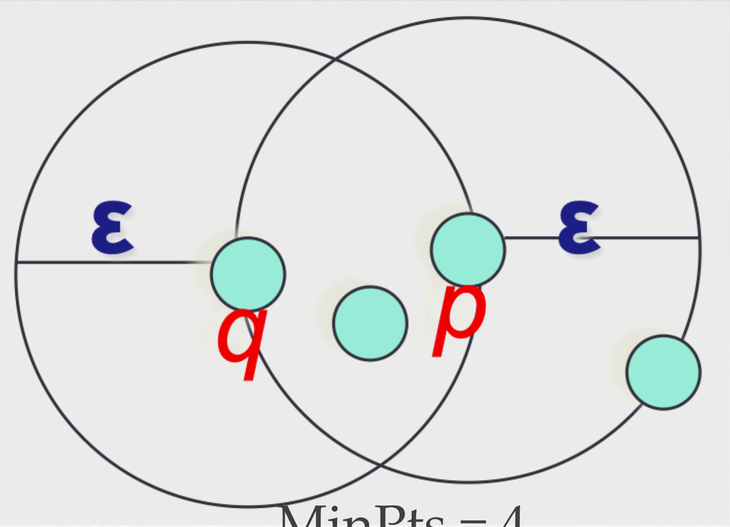
Image: Clustering, Lecture4 - Jing Gao ,SUNY Buffalo

DBSCAN
Definitions

- A point p is **directly density-reachable** from q if:
  - q is a core point
  - $p \in N_\epsilon(q)$
- Symmetric?
  - *Is F(p,q) = F(q,p)?*



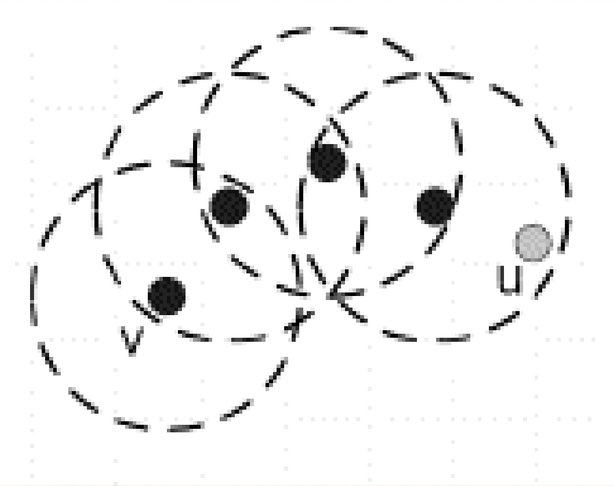
?



MinPts = 4

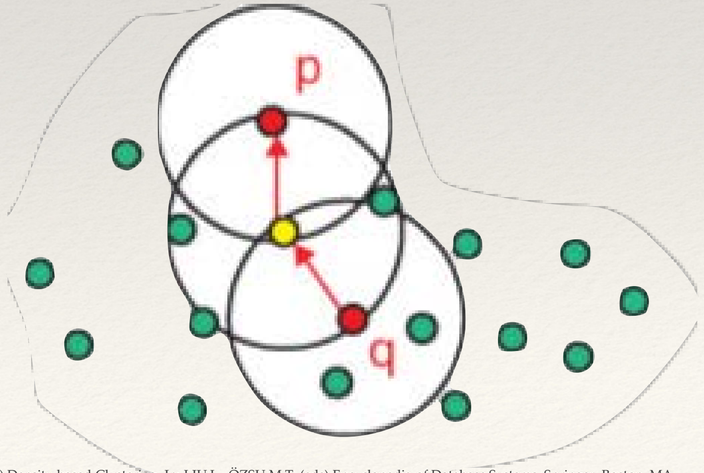
DBSCAN
Definitions

- A point p is **density-reachable** from point q if there is a chain of points $p_1, \ldots, p_n$ where:
  - $p_1 = q,\ p_n = p$
  - p_{i+1} *is directly density−reachable from* p_i
- *Symmetric?*



u is density-reachable from v

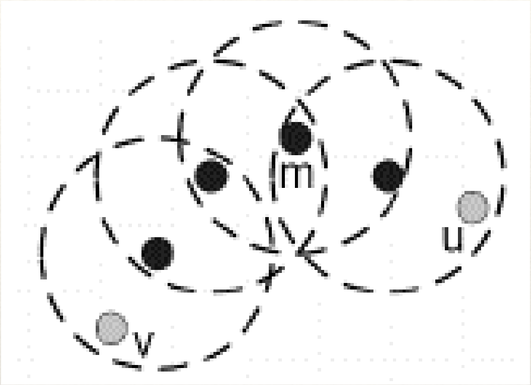
?



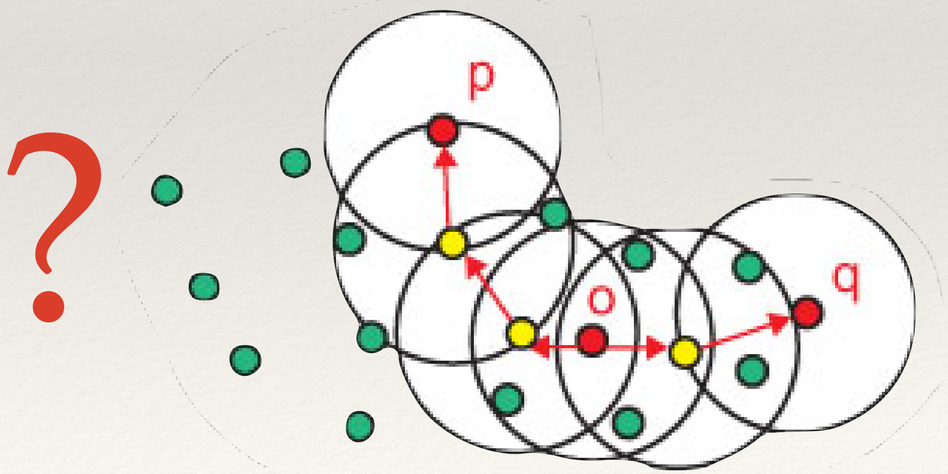
Ester M. (2009) Density-based Clustering. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA

DBSCAN
Definitions

- A point p is **density-connected** to point q if there's an object o such that p and q are **density-reachable** from o.
- Symmetric?



u is density-connected to v, due to existence of m

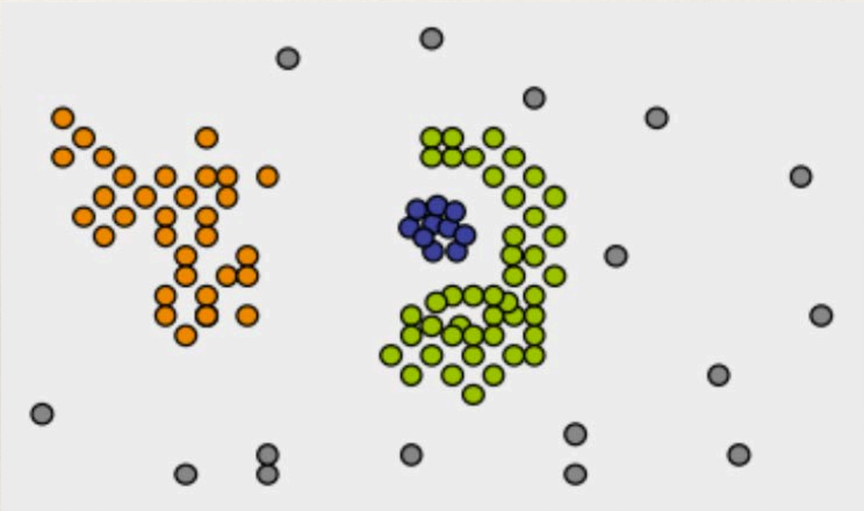


Ester M. (2009) Density-based Clustering. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA

DBSCAN
Definitions

- The main idea:
  - A **cluster** is a set of density-connected points (points that are density-reachable from an arbitrary core point)
  - **Noise** is the set of points which are not contained in any cluster

Now all we have to do is formulate an algorithm for finding that



DBSCAN
Algorithm

```
DBSCAN(DB, distFunc, eps, minPts) {
    C = 0                                                  /* Cluster counter */
    for each point P in database DB {
        if label(P) ≠ undefined then continue              /* Previously processed in inner loop */
        Neighbors N = RangeQuery(DB, distFunc, P, eps)     /* Find neighbors */
        if |N| < minPts then {                             /* Density check */
            label(P) = Noise                               /* Label as Noise */
            continue
        }
        C = C + 1                                          /* next cluster label */
        label(P) = C                                       /* Label initial point */
        Seed set S = N \ {P}                               /* Neighbors to expand */
        for each point Q in S {                            /* Process every seed point */
            if label(Q) = Noise then label(Q) = C          /* Change Noise to [redacted] */
            if label(Q) ≠ undefined then continue          /* Previously processed */
            label(Q) = C                                   /* Label neighbor */
            Neighbors N = RangeQuery(DB, distFunc, Q, eps) /* Find neighbors */
            if |N| ≥ minPts then {                         /* Density check */
                S = S ∪ N                                  /* Add new neighbors to seed set */
            }
        }
    }
}
```

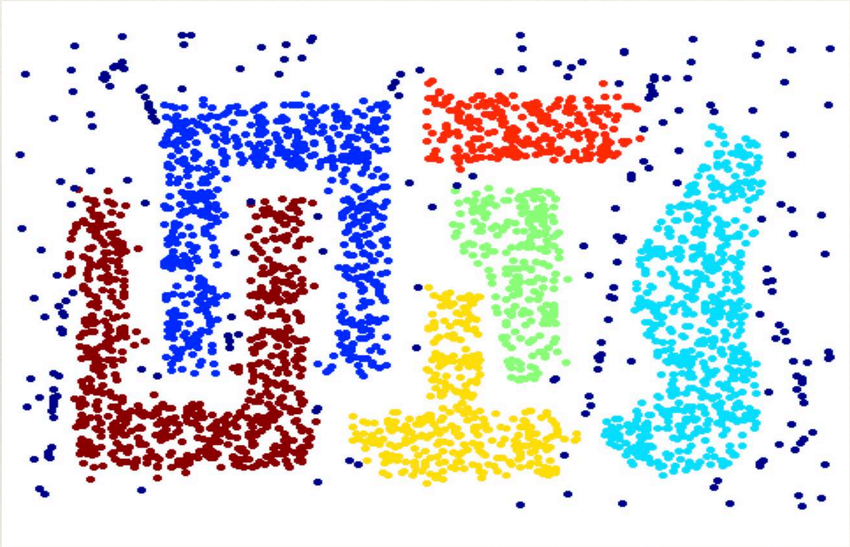
Source: Wikipedia

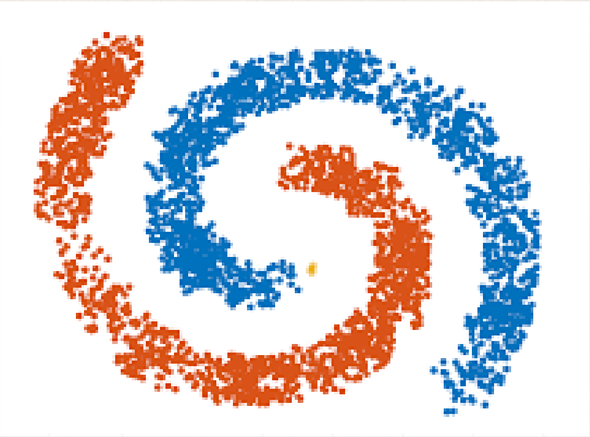
?

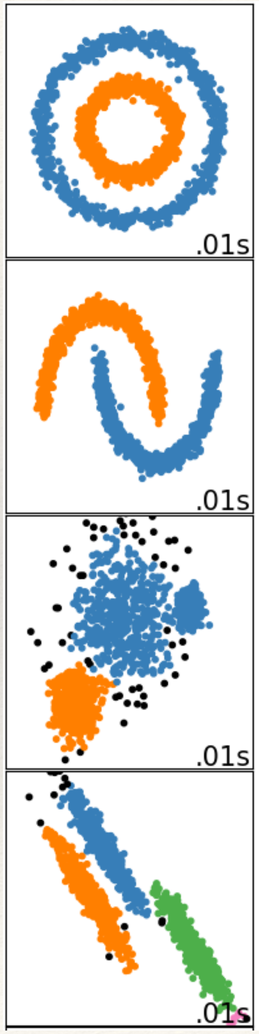
DBSCAN
Complexity

- Typically $\mathcal{O}(N^2)$
- With spatial indexing (e.g. kd-tree) can be reduced to $\mathcal{O}(N \log N)$

DBSCAN
Examples





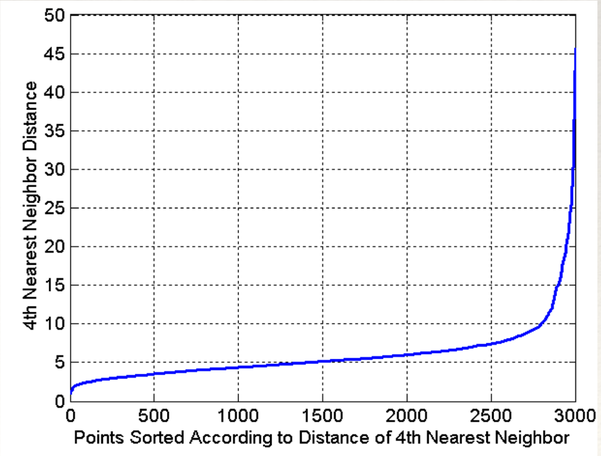


DBSCAN

Demo time!

DBSCAN
Parameters

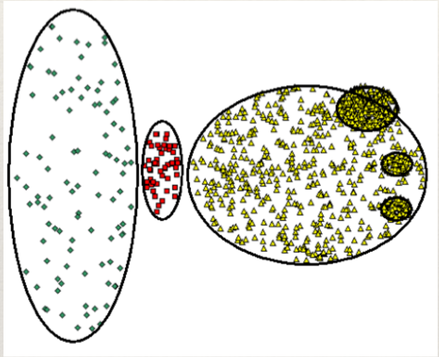
- $MinPts$
  - Can be set by a domain expert
  - According to dimension size: $MinPts \geq D + 1$
  - The larger the dataset - the larger value of $MinPts$ should be chosen
    - Usually governs the min size of cluster. Except for edge cases, border points
- ϵ
  - Too small - majority of the points will remain as noise
  - Too high - expand cluster to unnecessary size
  - Can be chosen using [k-distance] graph

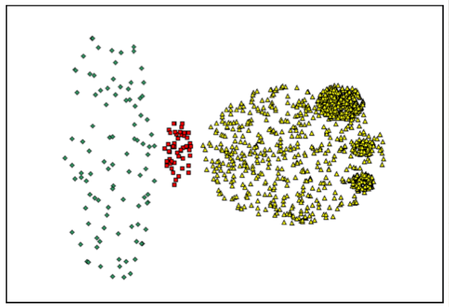


Images: Clustering, Lecture4 - Jing Gao ,SUNY Buffalo

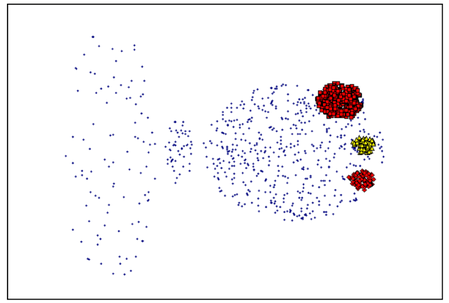
DBSCAN
Main drawback

- DBSCAN is very (!) sensitive to parameters - especially when the clusters are of *varying density*.
- Usually requires expert knowledge in setting those.



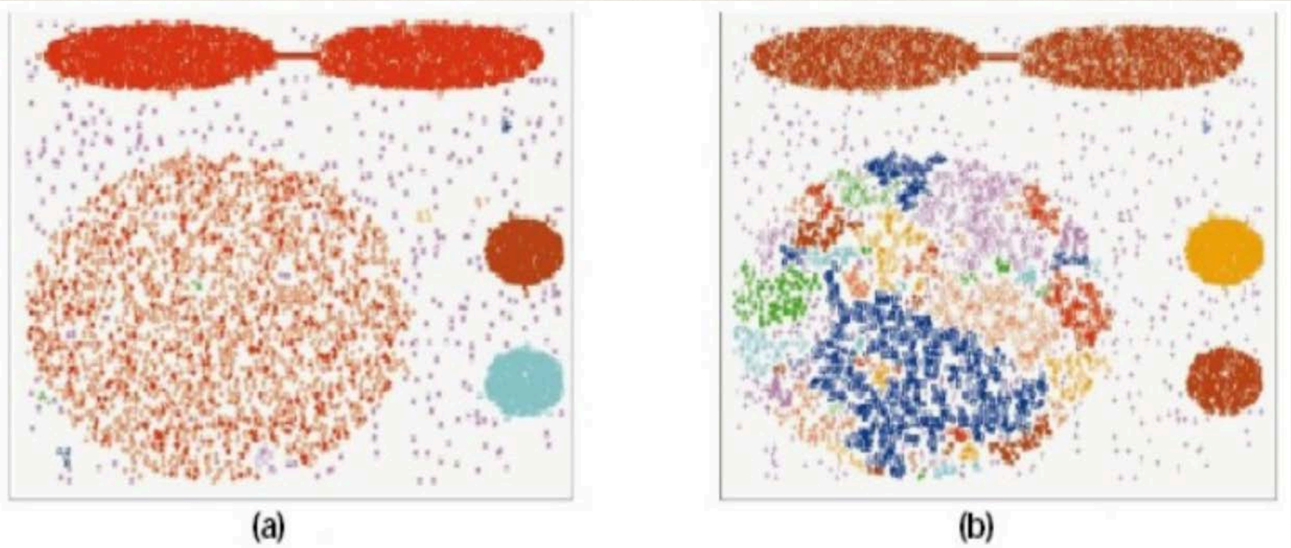


(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

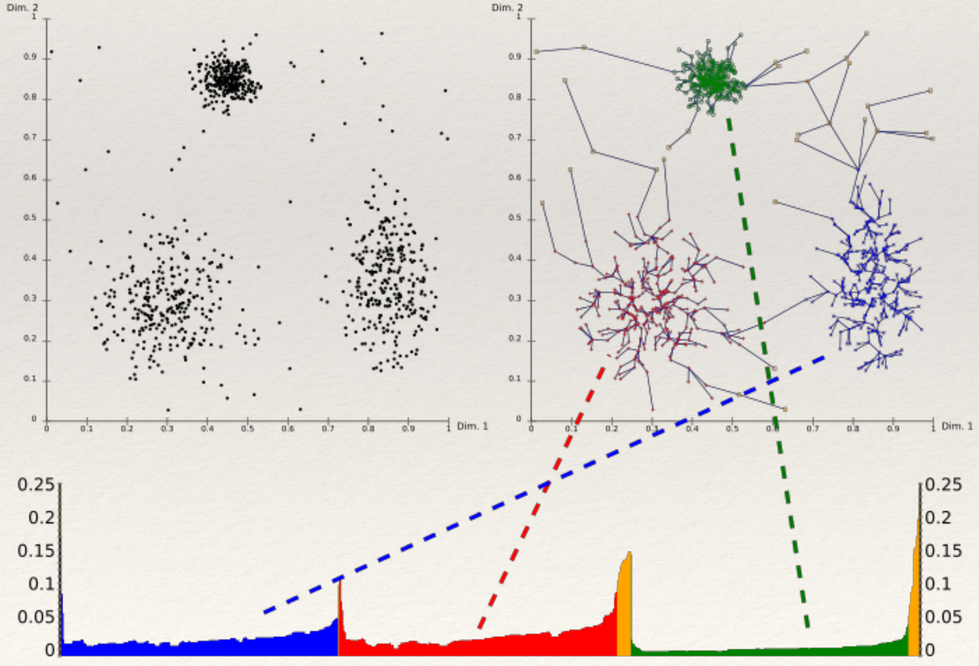
Images: Clustering, Lecture4 - Jing Gao ,SUNY Buffalo



Capter10 cluster basic : Han & Kamber

DBSCAN
Main drawback — Solution

- OPTICS can find clusters of varying density.
  - "OPTICS: Ordering Points To Identify the Clustering Structure", Mihael Ankerst et al
  - Can be used by itself (ignore ϵ) or just auxiliary to select ϵ



reachability-plot

Notebook time

DBSCAN
Improvements

- Can border points change clusters in different runs?

"Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection", Campello, Ricardo J. G. B et al.

?

DBSCAN Improvements

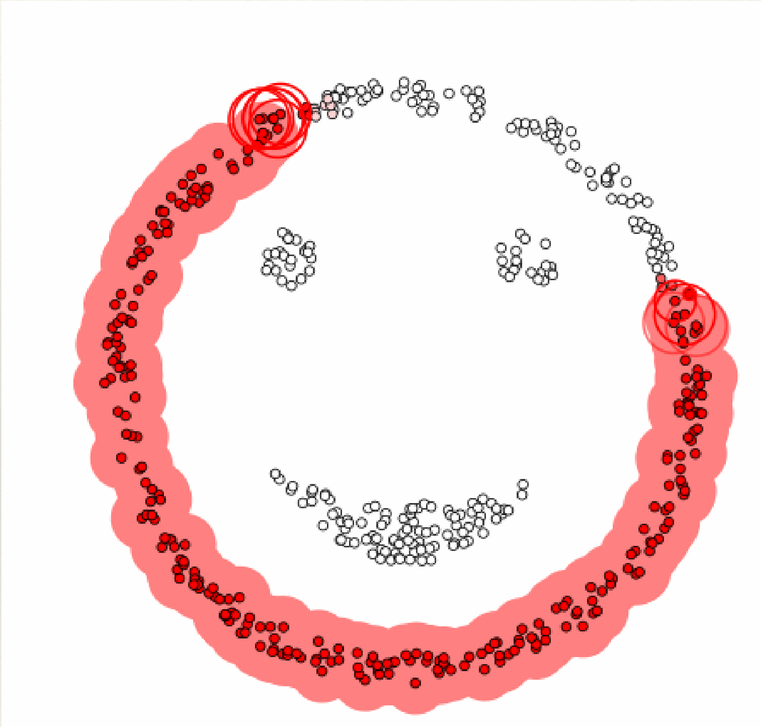
- Treat DBSCAN as a hierarchical problem (HDBSCAN)
  - "Density-Based Clustering Based on Hierarchical Density Estimates", Davoud Moulavi et al.
  - How HDBSCAN works

DBSCAN questions

- Can a cluster have zero core points?
- Can a cluster have less than minPts in it?
- Are DBSCAN results deterministic in terms of random seed?
- Are they always deterministic?
- How distance is defined in DBSCAN?

DBSCAN
Summary

- Pros
  - Can find arbitrary shaped clusters
  - Robust to noise and outliers
  - Doesn't require specific number of clusters
- Cons
  - Semi-deterministic
  - Struggles with different densities in data
  - Choosing ϵ can be difficult



Hierarchical Clustering

Hierarchical Clustering

- Sometimes we want to get more out of our clustering

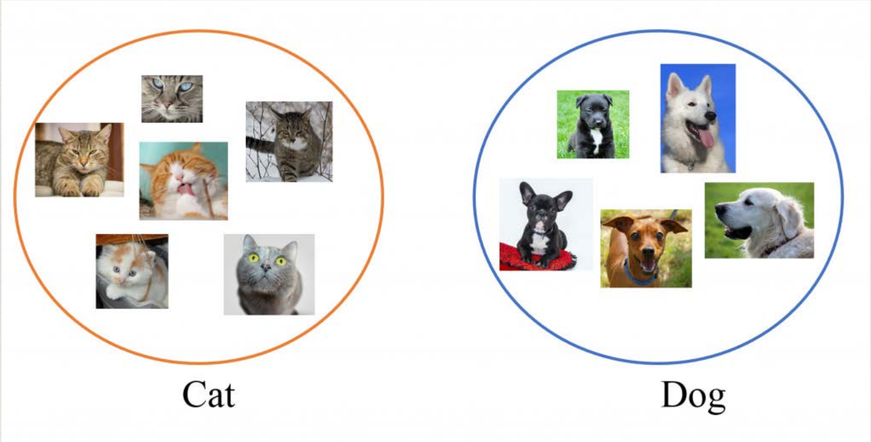


Image: www.medium.com

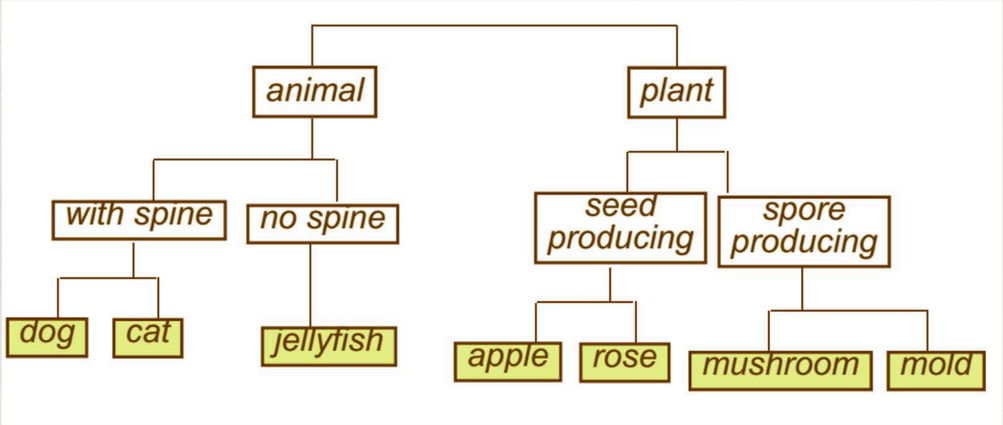


Image: www.medium.com

Hierarchical Clustering

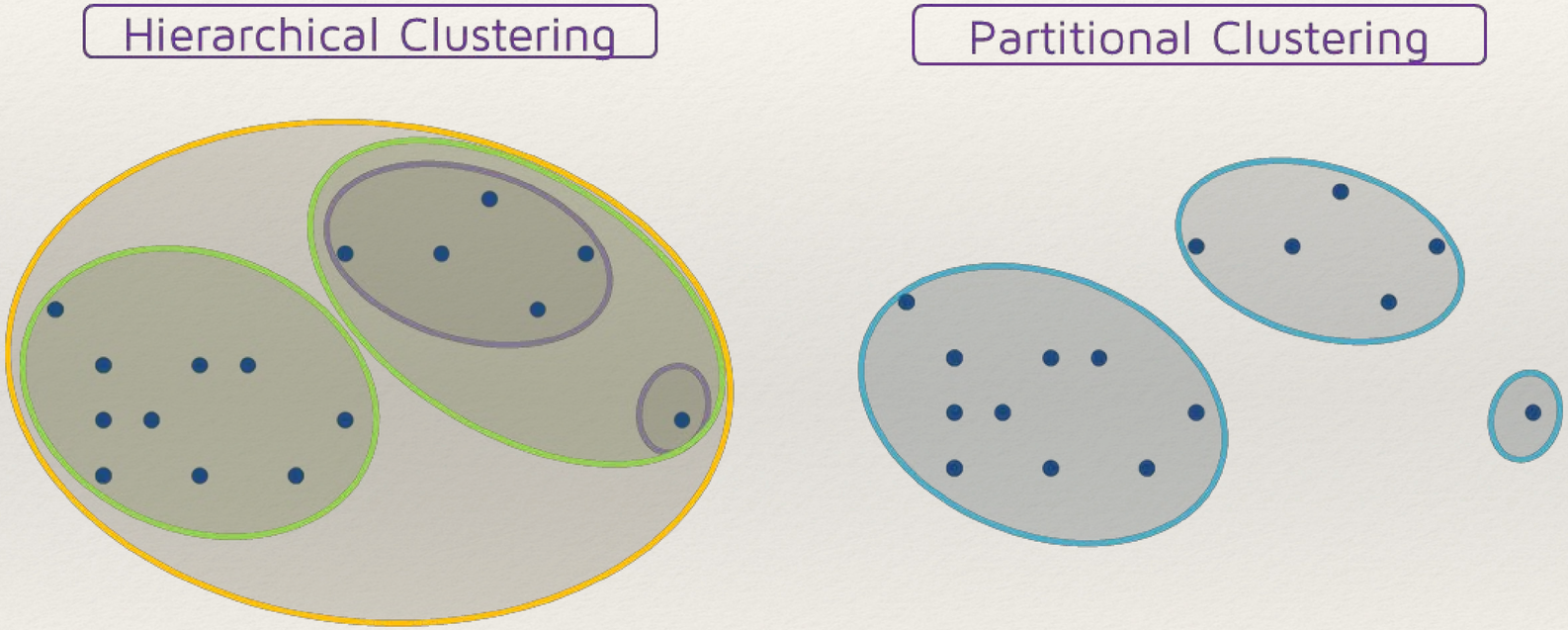


Image: Y-Data 2018 lectures

Hierarchical Clustering

- The idea: build a tree of the data that successively clusters similar group of points.

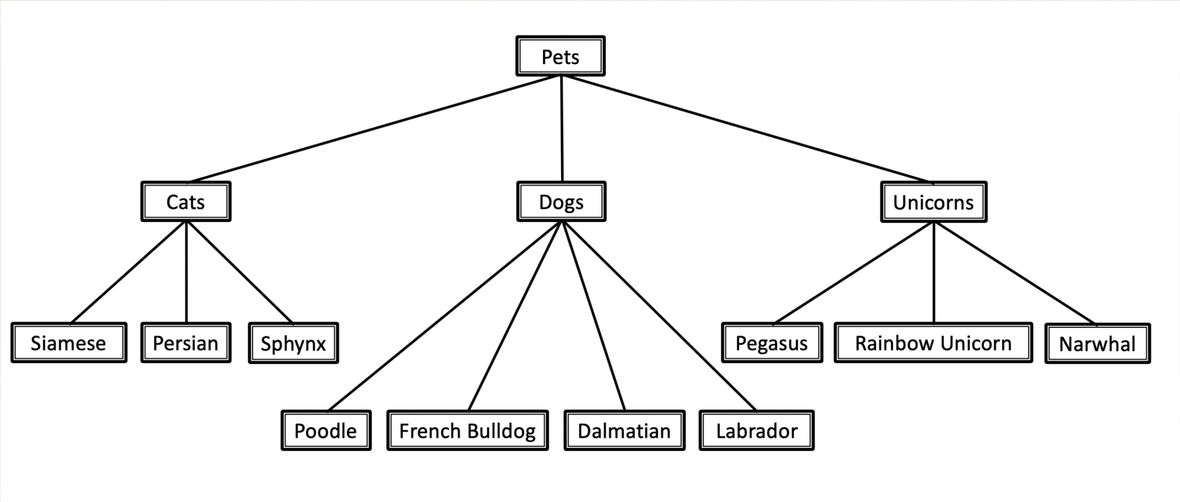


Image: Noa Weiss, https://towardsdatascience.com/https-medium-com-noa-weiss-the-hitchhikers-guide-to-hierarchical-classification-f8428ea1e076

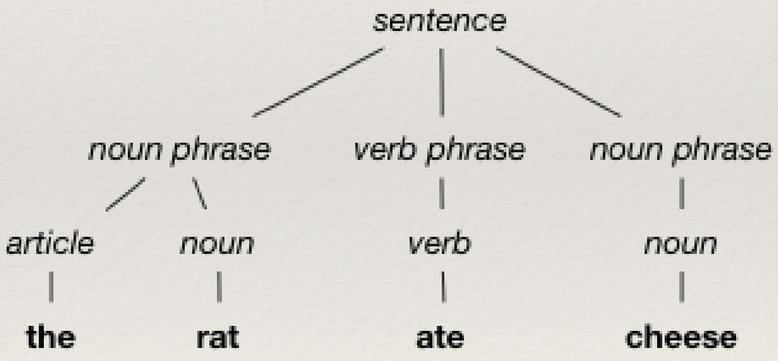


Image: CS 2112, Cornell University

Hierarchical Clustering
Top-down Vs. Bottom-up

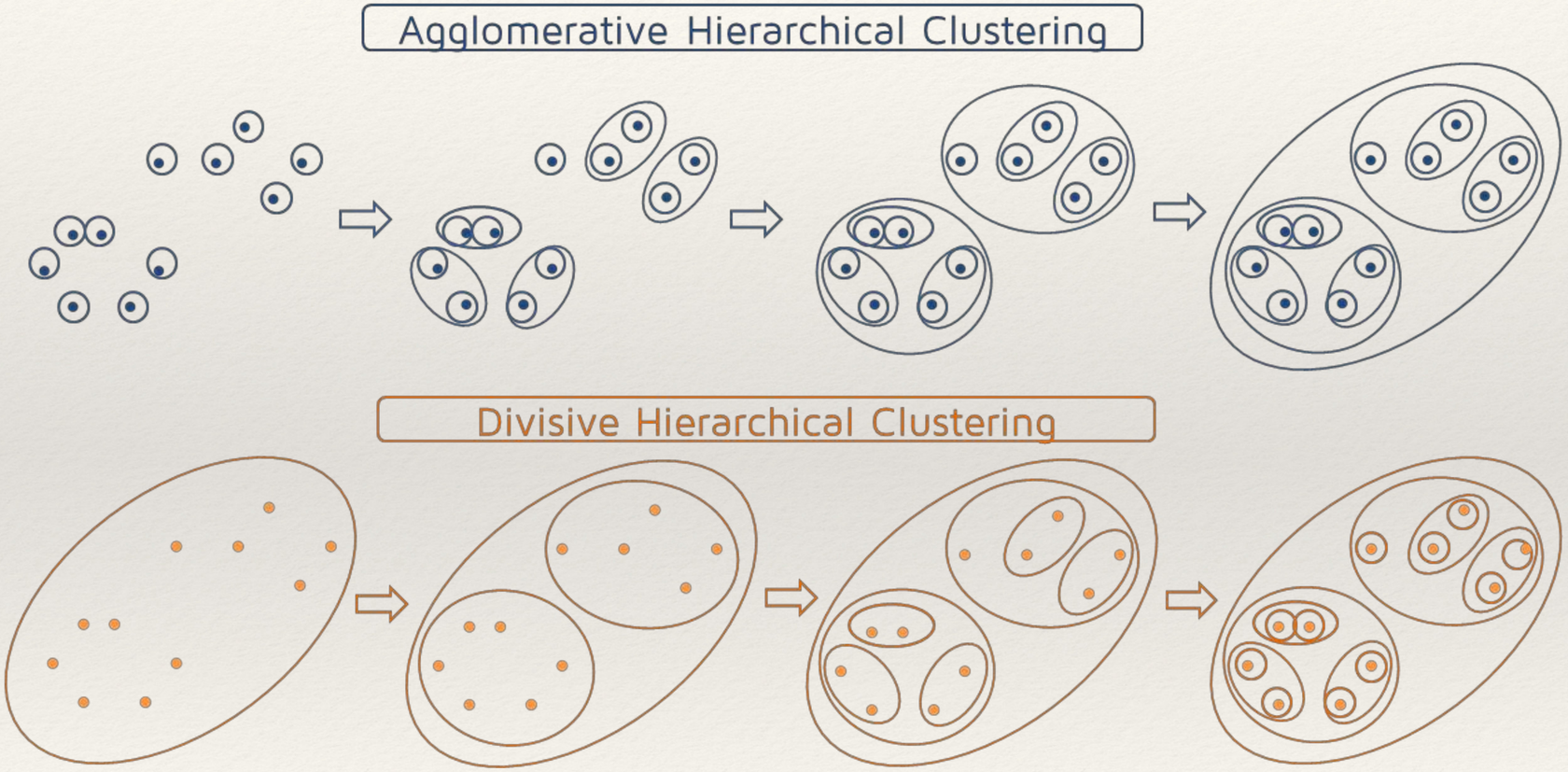


Image: Pawan Jain, https://towardsdatascience.com/https-towardsdatascience-com-hierarchical-clustering-6f3c98c9d0ca

Hierarchical Clustering
Agglomerative

- Init — each point is a cluster
- Repeat:
  - Compute the distance/similarity matrix between all clusters
  - Merge the two closest clusters

Until a single cluster remains

Different definitions of distances can results in different clustering.

Hierarchical Clustering - Agglomerative Dendrogram

- A tree-like diagram that shows the sequences of merges\splits.
- Each node is a cluster

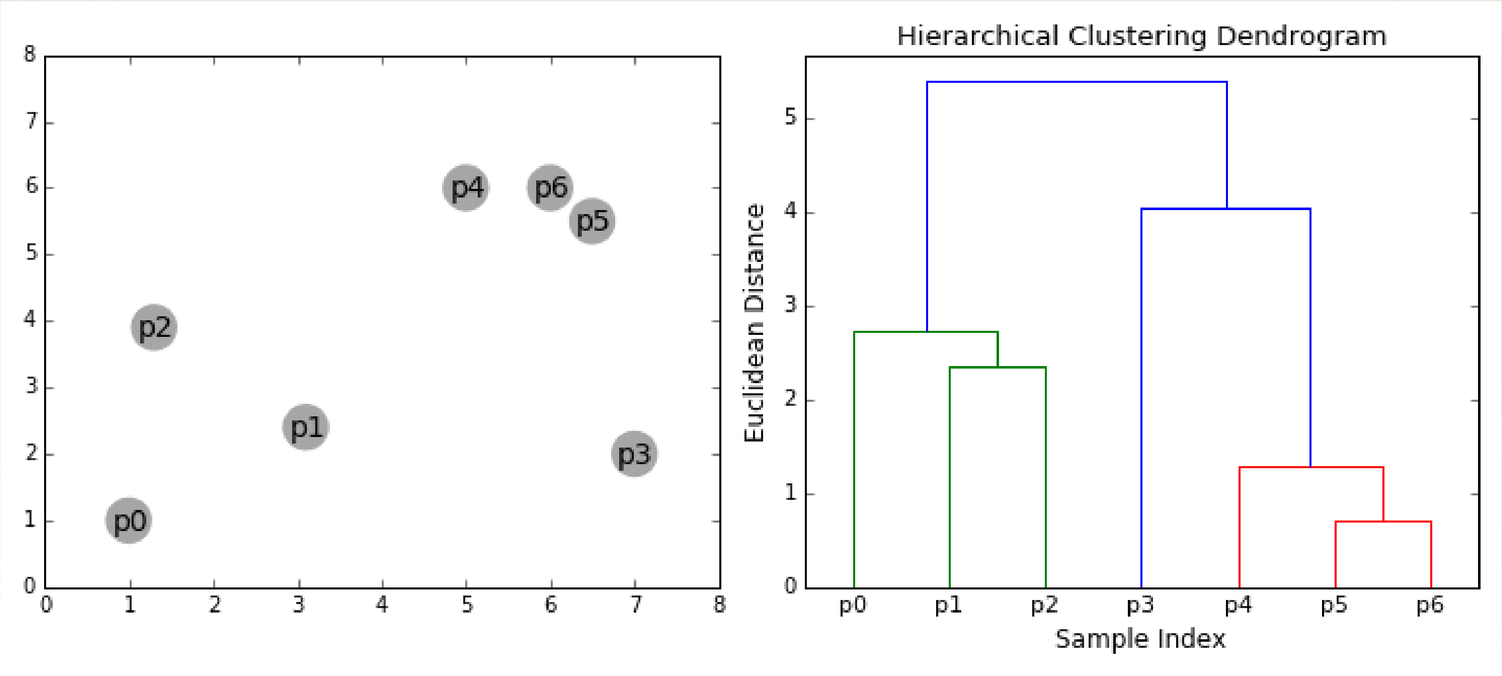


Image: Alex Casanas, https://codingwithalex.com/hierarchical-clustering/

Hierarchical Clustering - Agglomerative Dendrogram

- Clustering is obtained by cutting the dendrogram at a desired level
- Required similarity score
- Largest gap in similarity
- $K = \underset{k}{\operatorname{argmin}}(RSS(k) + \lambda k)$
- Predefined number of clusters, k

?

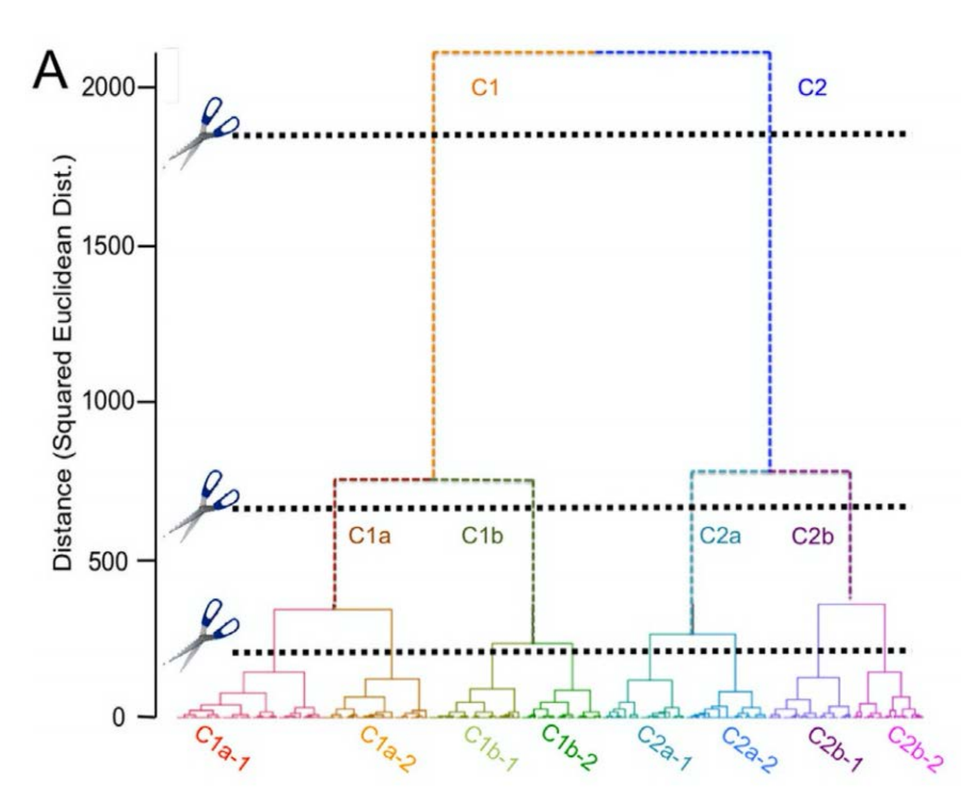
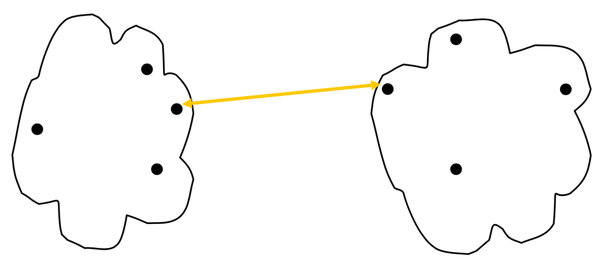


Image: Data-Driven Phenotypic Categorization for Neurobiological Analyses: Beyond DSM-5 Labels
Nicholas T. Van Dam

Hierarchical Clustering - Agglomerative
Linkage type

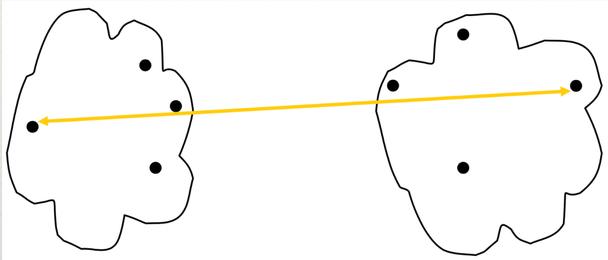
Common cluster distance metrics:

Single Linkage



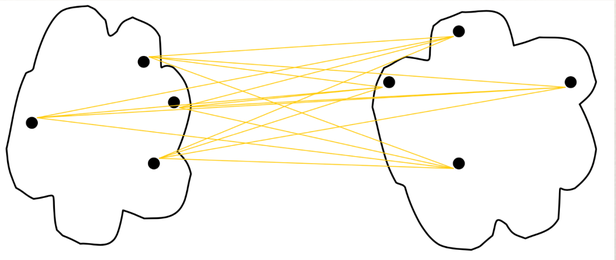
$$d_{\min}(C_i, C_j) = \min_{p \in C_i, q \in C_j} d(p,q)$$

Complete Linkage



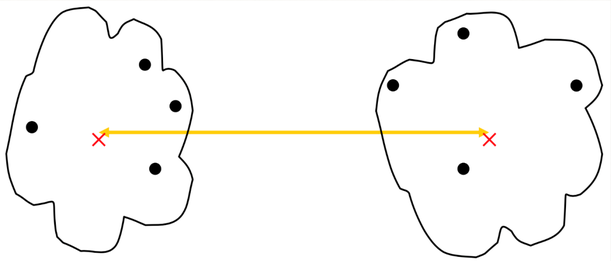
$$d_{\min}(C_i, C_j) = \max_{p \in C_i, q \in C_j} d(p,q)$$

Average Linkage



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

Centroid Linkage



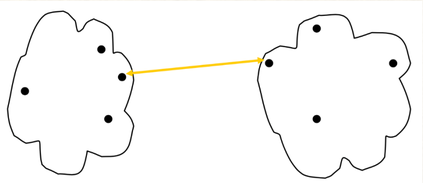
$$d_{mean}(C_i, C_j) = d(m_i, m_j)$$

Images: Clustering, Lecture3 - Jing Gao ,SUNY Buffalo

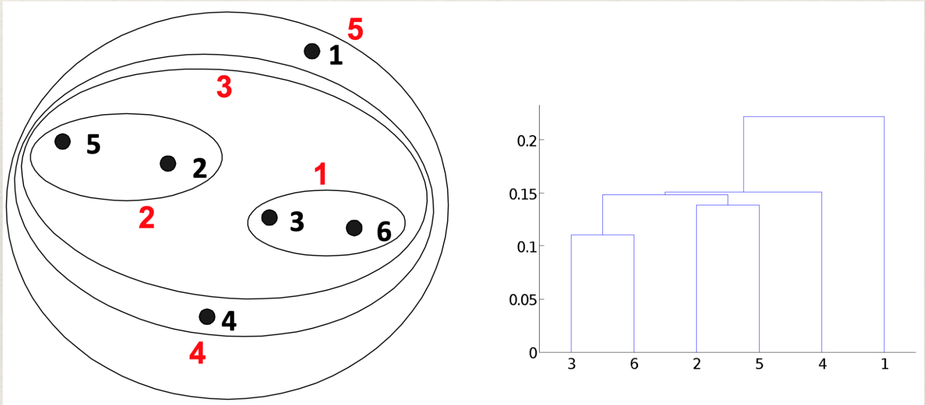
Hierarchical Clustering - Agglomerative
Single Linkage

- Defined by just one pair of points
- Pros:
  - Can handle non elliptic shapes
  - Can improve time complexity
- Cons
  - Encourage elongated clusters - "chaining"
  - Sensitive to noise

Single Linkage



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



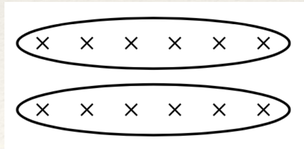


Image: Cambridge University Press

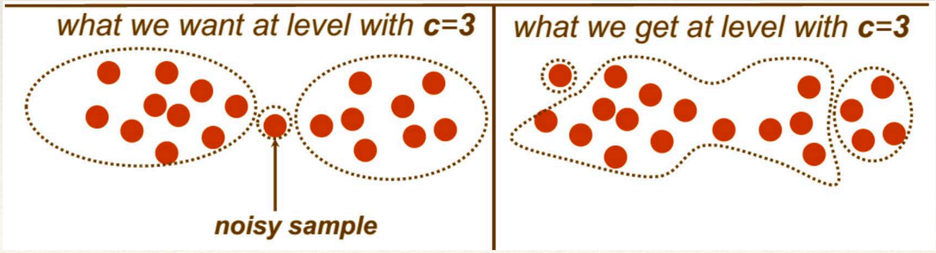
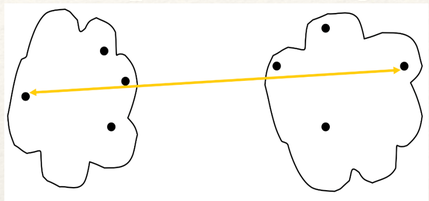


Image: Shimon Ullman, Center for Brains, Minds & Machines

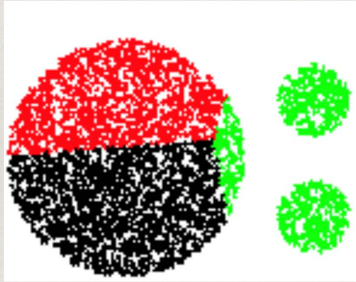
Hierarchical Clustering - Agglomerative
Complete Linkage

- Defined by the most dissimilar set of points
- Pros
  - Encourages compact clusters
  - Less susceptible to noise (not perfect)
- Cons
  - Tends to break large clusters
  - Point can be closer to points in other clusters than to points in its own cluster (crowding)
  - Clusters are compact, but not far enough apart

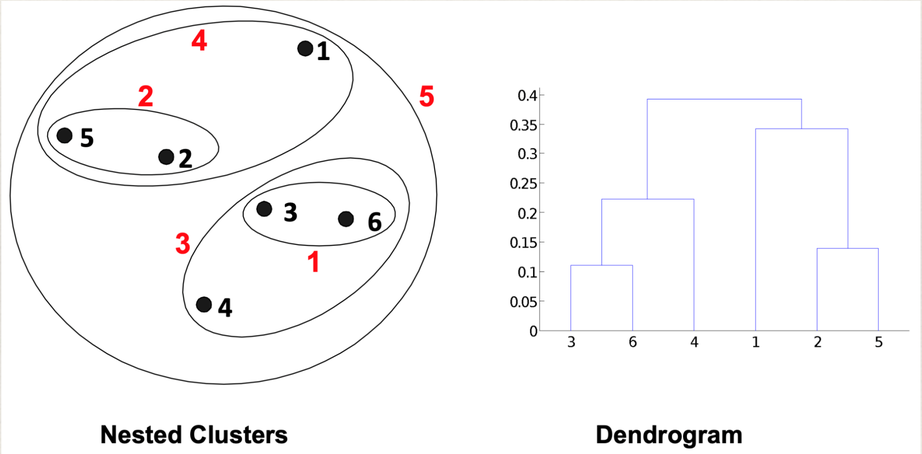
Complete Linkage



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



Images: Clustering, Lecture3 - Jing Gao ,SUNY Buffalo



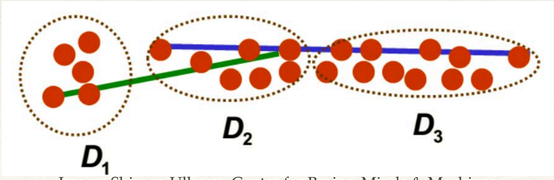
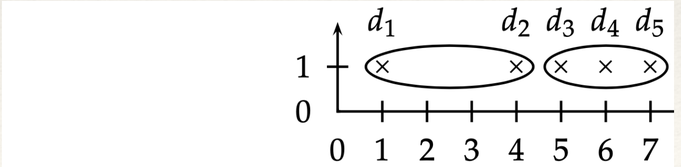


Image: Shimon Ullman, Center for Brains, Minds & Machines



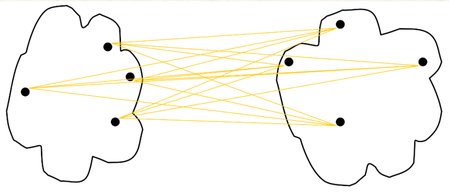
► **Figure 17.7** Outliers in complete-link clustering the x-coordinates $1+2\epsilon, 4, 5+2\epsilon, 6$ and $7-\epsilon$.

Image: Cambridge University Press

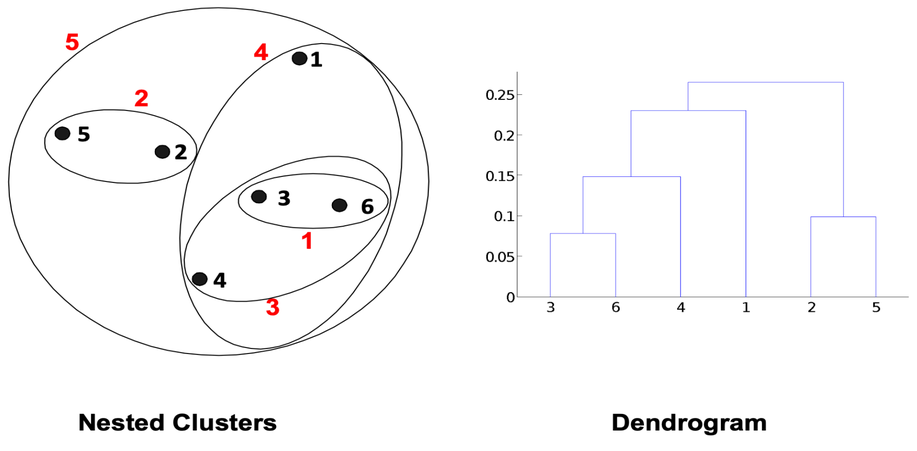
Hierarchical Clustering - Agglomerative
Average Linkage

- Represents a natural compromise between single\complete linkage
- Pros
  - Less susceptible to noise and outliers
- Cons
  - Biased towards global patterns
  - Not clear what properties the resulting clusters. Single and complete linkage each had simple interpretations

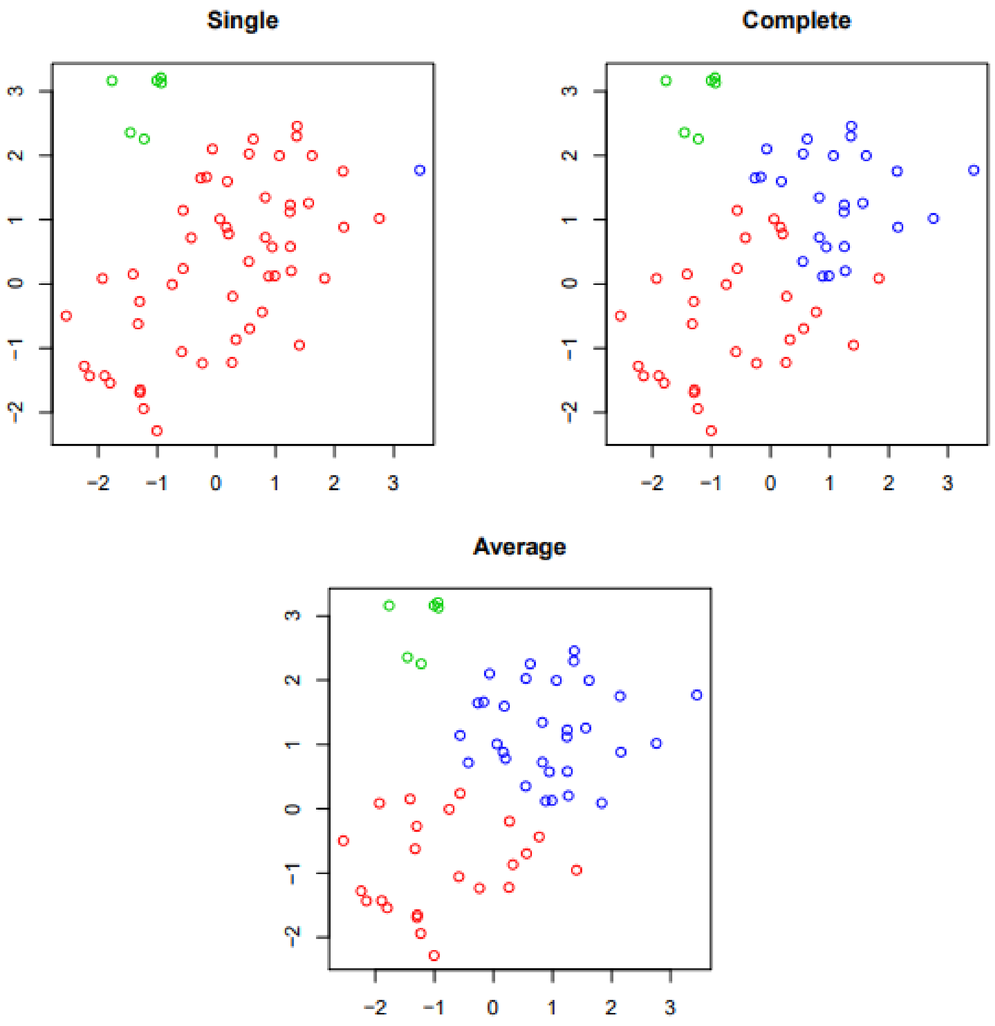
Average Linkage



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



Example of chaining and crowding



Hierarchical Clustering
Ward's method

- Distance between two clusters, A and B, is how much the sum of squares will increase when we merge them (merging cost)

$$d_w(A,B) = \sum_{x \in A \cup B} (x - \mu_{AB})^2 - \sum_{x \in A} (x - \mu_A)^2 - \sum_{x \in B} (x - \mu_B)^2$$

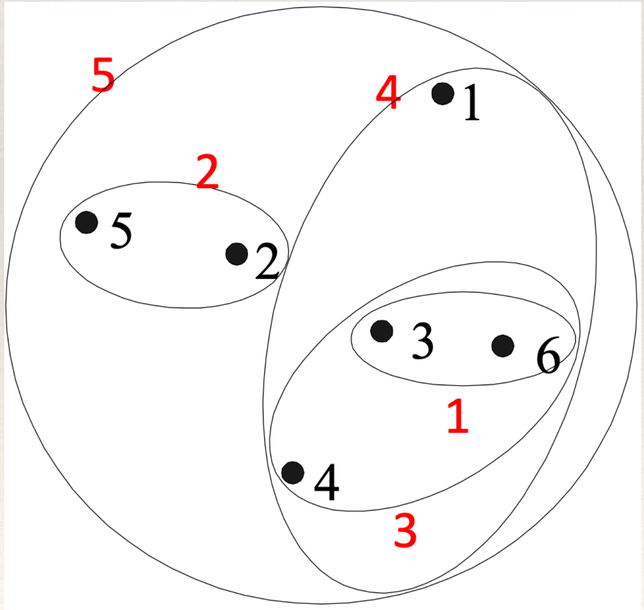
$$= \frac{n_A n_B}{n_A + n_B} (\mu_A - \mu_B)^2$$

Hierarchical Clustering
Ward's method

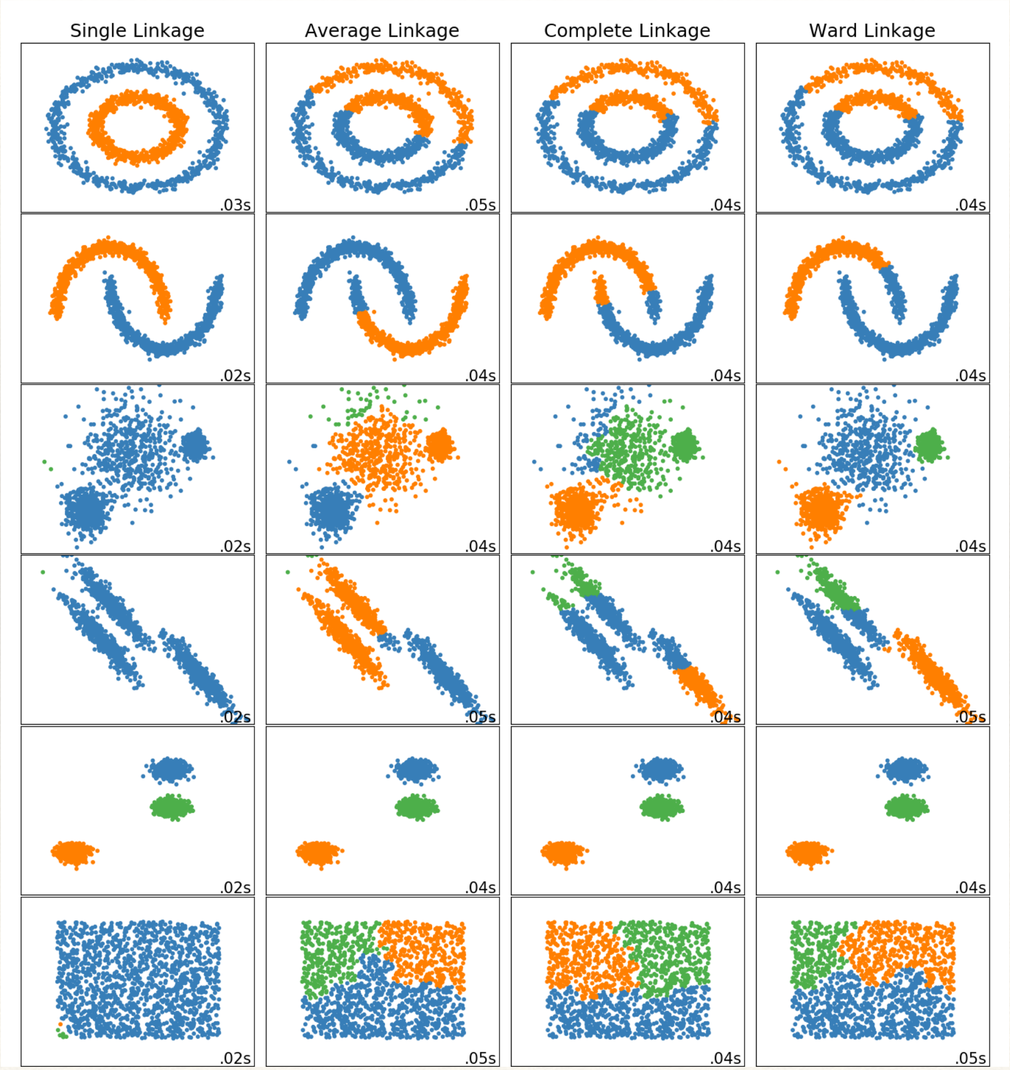
- Similar to average linkage (with L_2)
- In case of a draw - prefer smaller clusters
- Pros
  - Less susceptible to noise and outliers
  - Hierarchical analogue of K-Means
- Cons
  - Biased towards global patterns

$$d_w(A,B) = \sum_{x \in A \cup B} (x - \mu_{AB})^2 - \sum_{x \in A} (x - \mu_A)^2 - \sum_{x \in B} (x - \mu_B)^2$$

$$= \frac{n_A n_B}{n_A + n_B} (\mu_A - \mu_B)^2$$



Hierarchical Clustering
Methods comparison



?

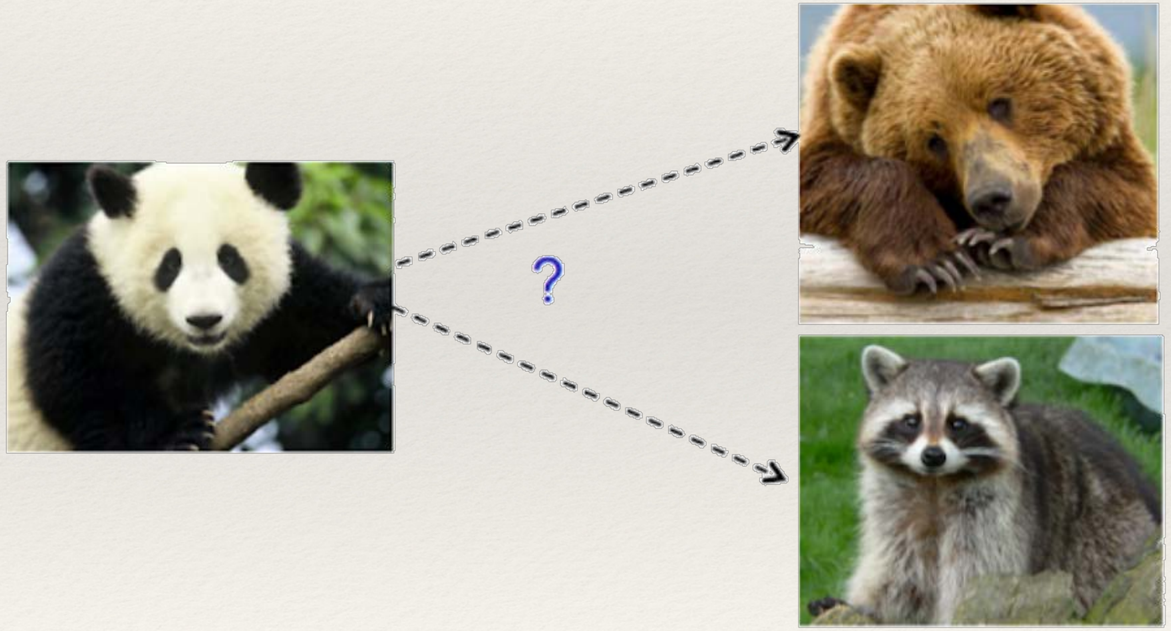
Hierarchical Clustering - Agglomerative
Complexity

- In most cases $\mathcal{O}(N^3)$
- Using priority queue, we can reduce complexity to $\mathcal{O}(N^2 \log N)$
- For single-linkage, we can even achieve $\mathcal{O}(N^2)$

Use cases: Charting Evolution

- Are giant pandas closer to bears or raccoons?

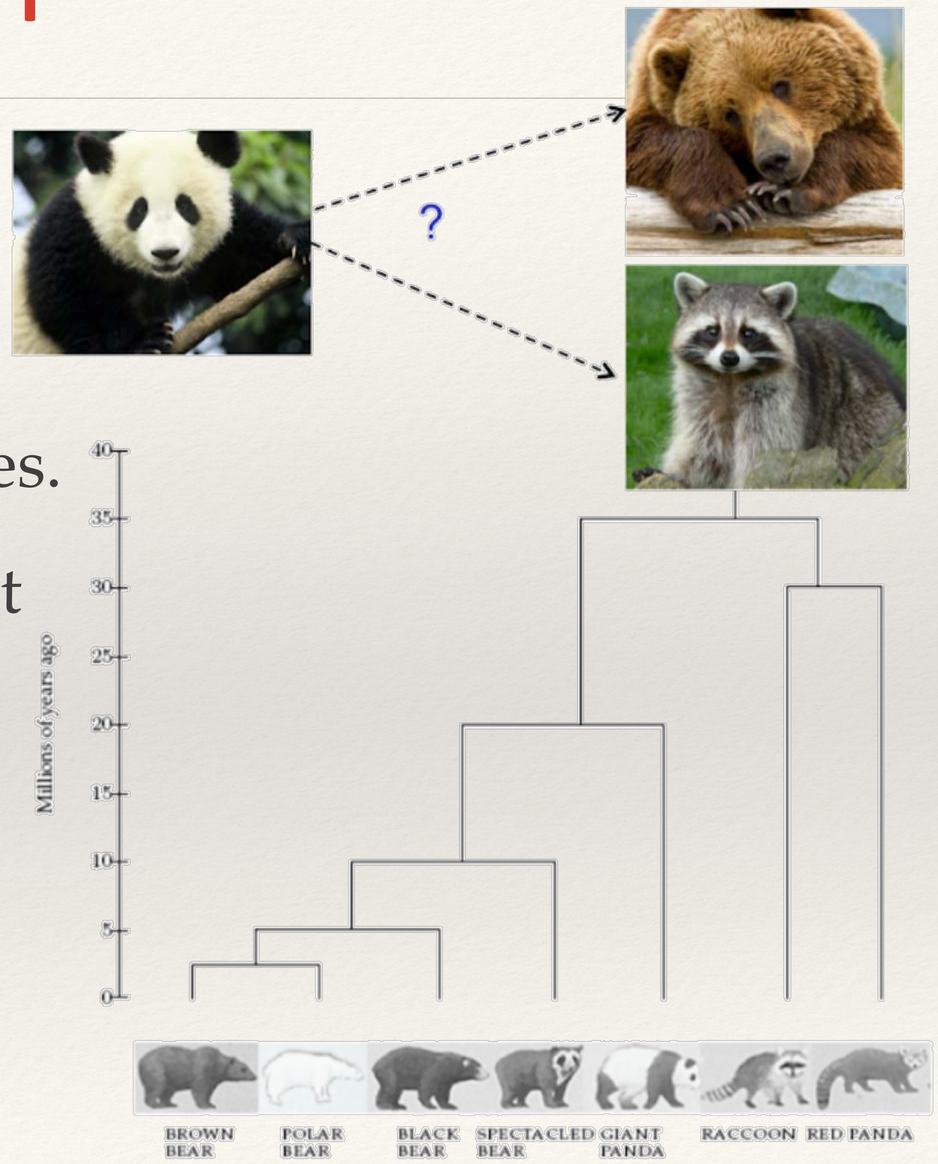
Voting time!



Use cases: Charting Evolution

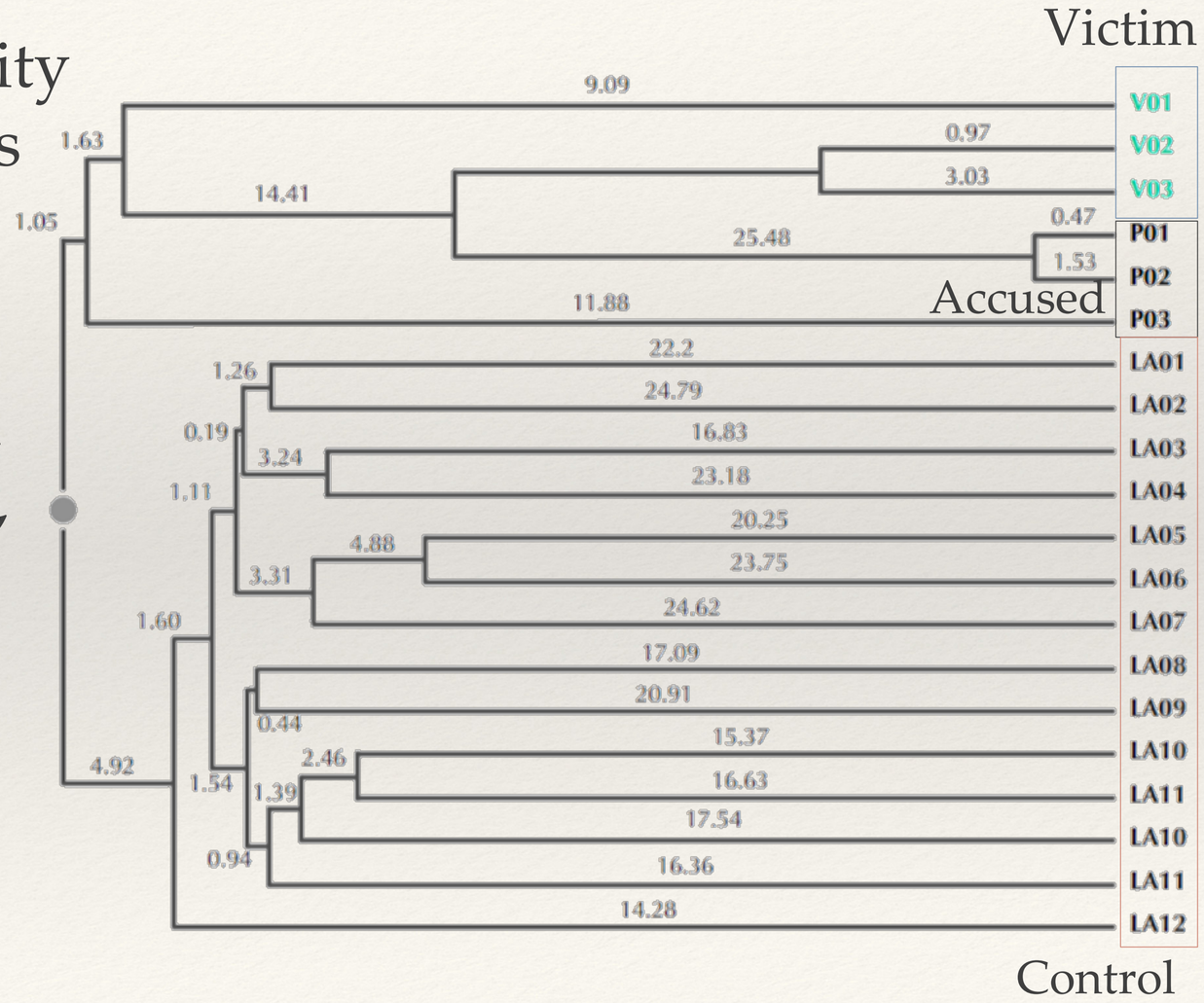
- Are giant pandas closer to bears or raccoons?

1. Generate the DNA sequences
2. Calculate the edit distance between all sequences.
3. Calculate the DNA similarities based on the edit distances.
4. Construct the phylogenetic tree.

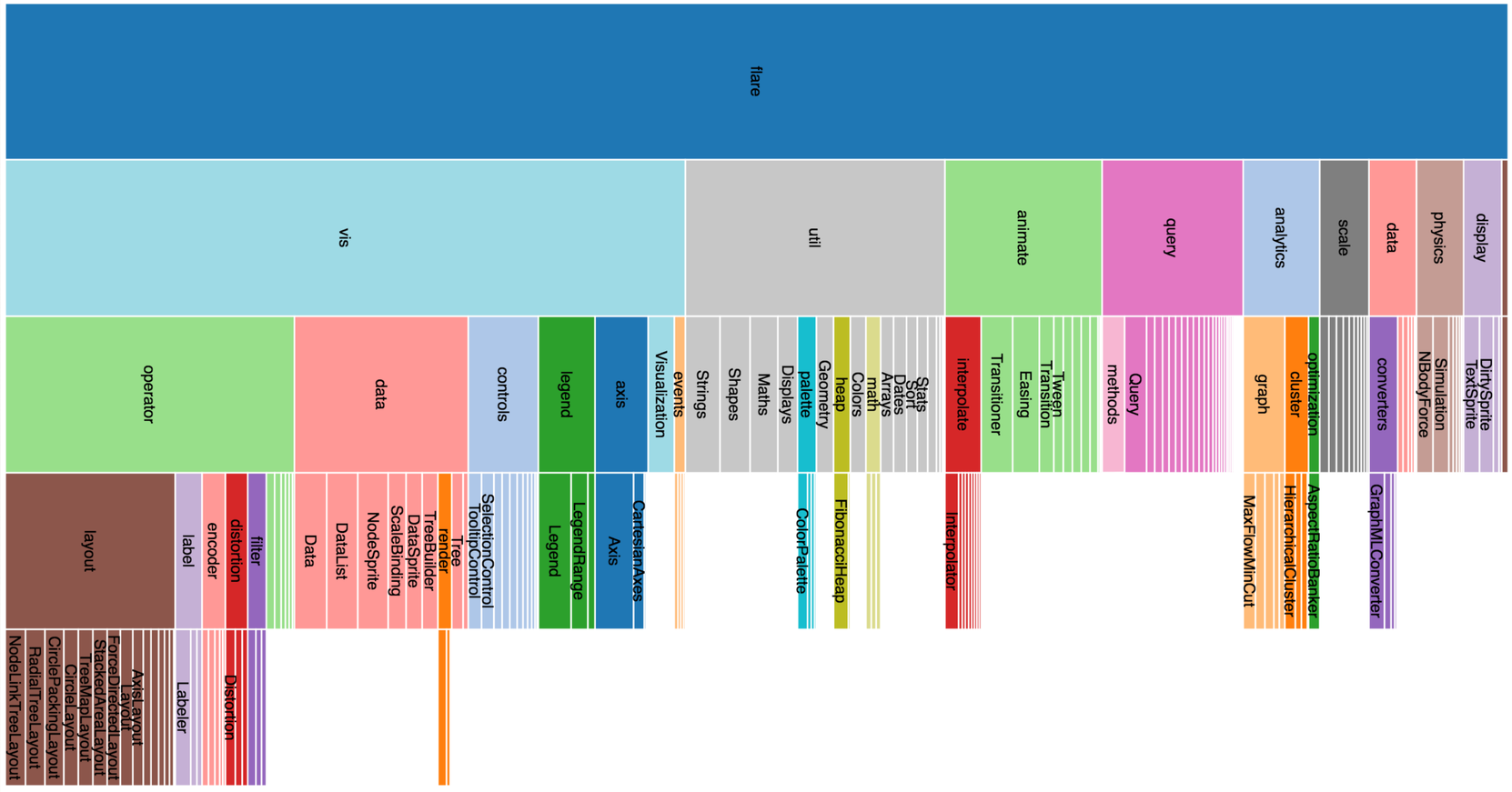


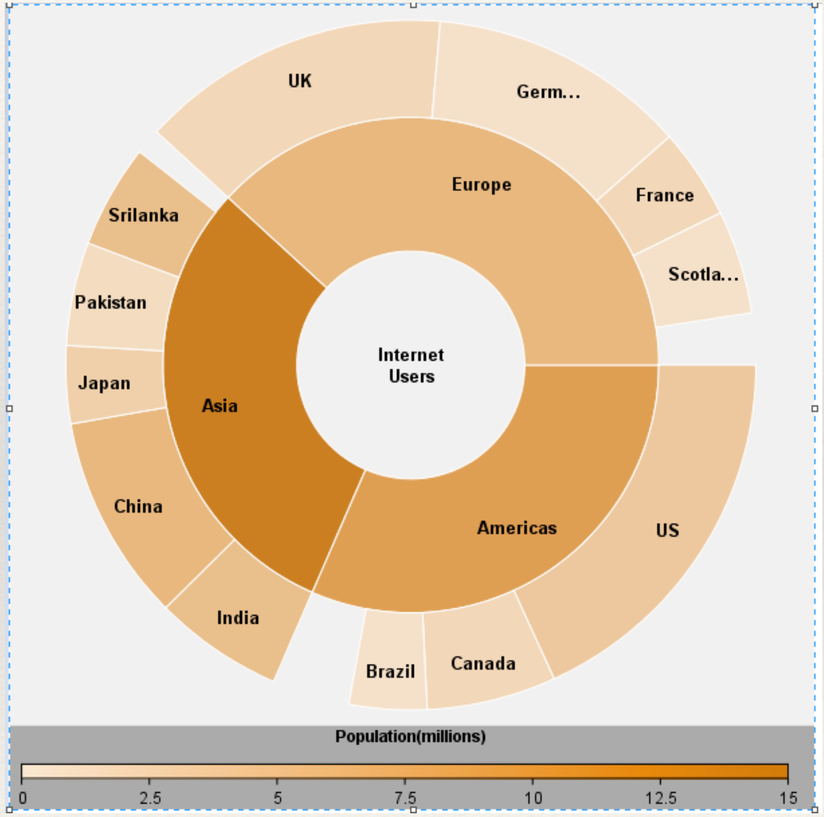
Use cases: Tracking Viruses

- HIV have high mutation rate = similarity of the DNA sequence of the same virus depends on the time since it was transmitted.
- Evidence in **court case**, victim's strand of HIV more similar to accused strand, compared to control group.



Hierarchical Clustering
Icicle





Images: Y-Data 2018 lectures

Hierarchical Clustering
Interactive Icicle

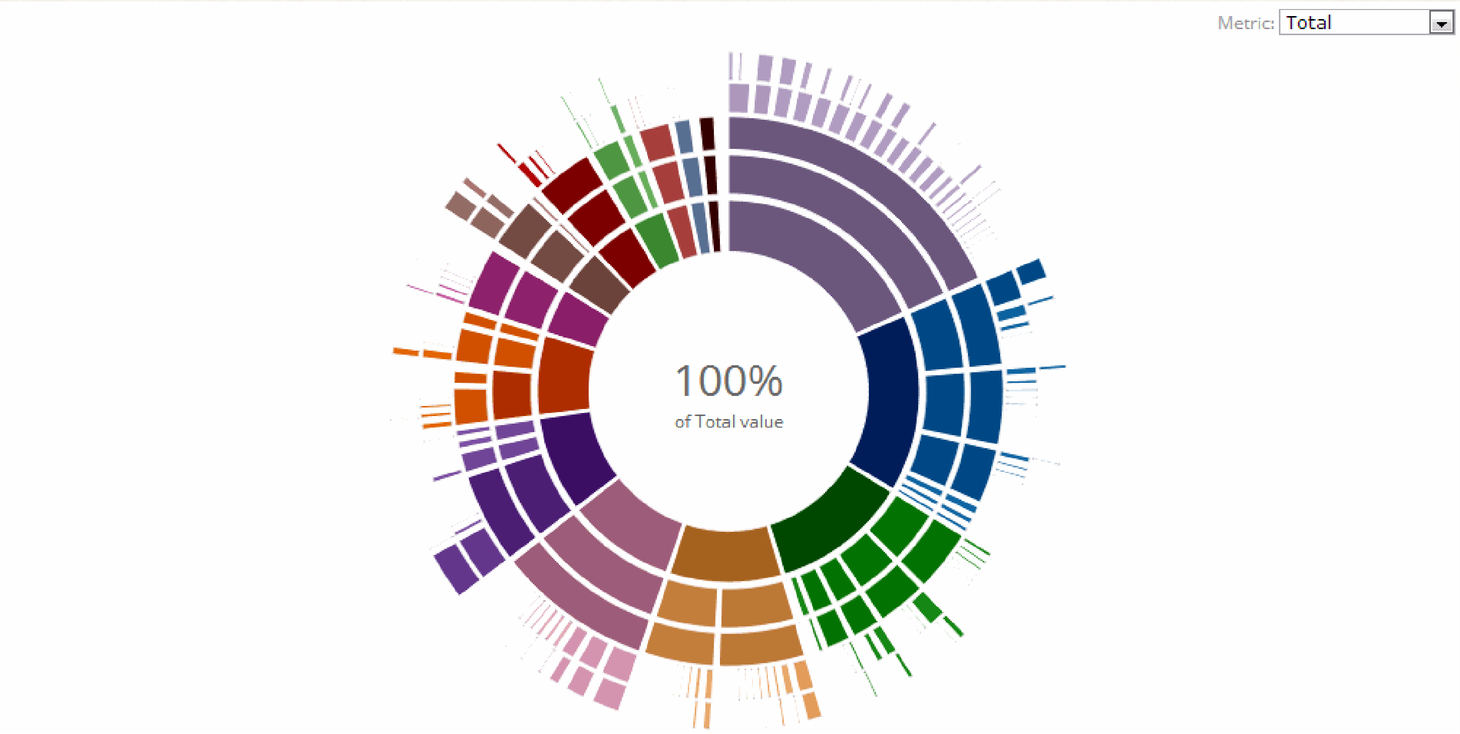
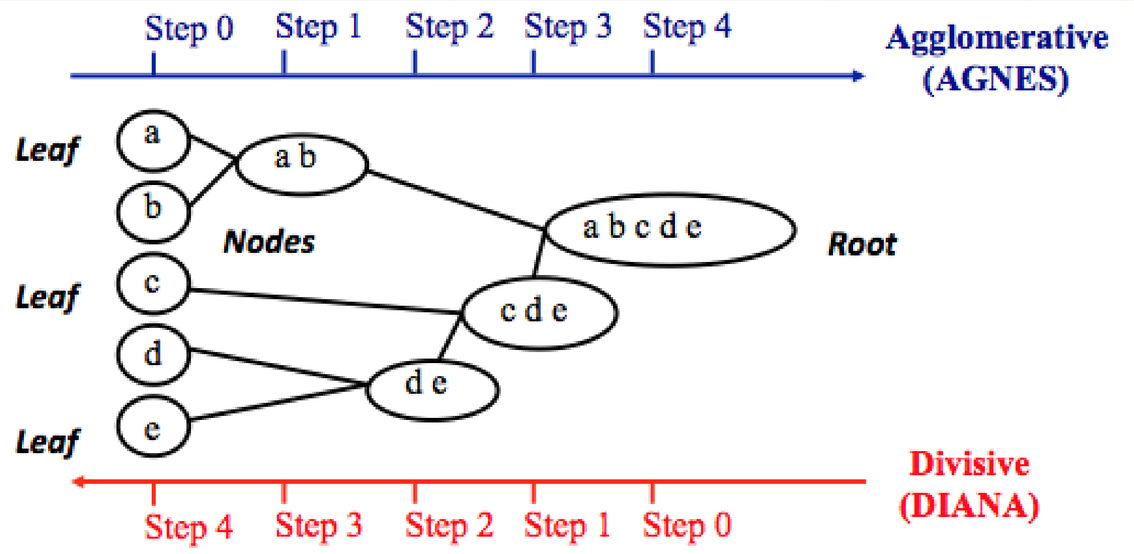


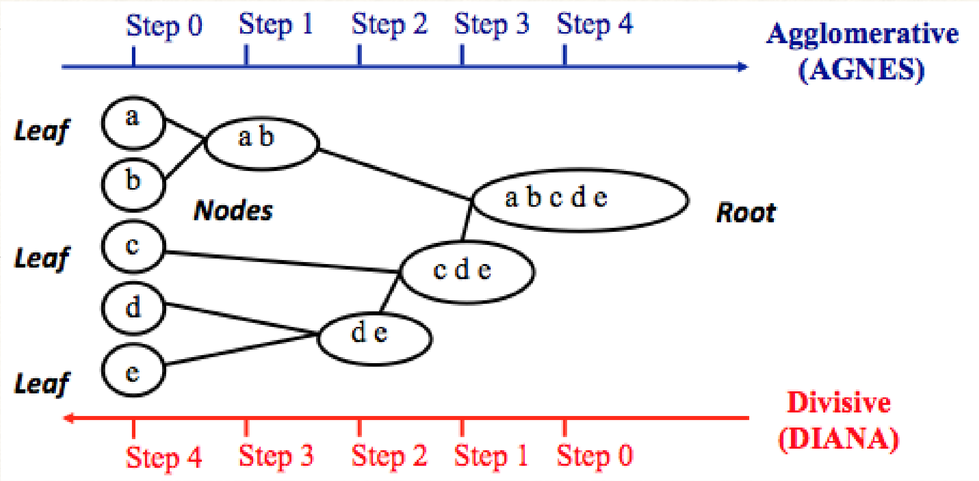
Image: Y-Data 2018 lectures

Divisive hierarchal clustering

- Where all have started: DIANA
  - "Clustering in an Object-Oriented Environement", Anja Struyf et al.

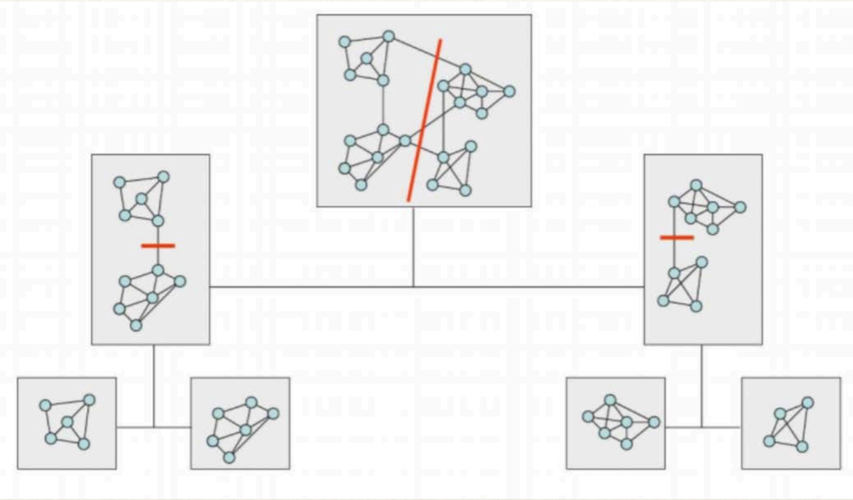


Hierarchical Clustering
Divisive



- DIANA, main idea:
- Starting with one cluster containing all n observations.
- Repeat
  - At each stage, a cluster *C*, with the largest diameter is selected.
  - Find the most disparate observation in *C* and starts a “splinter group”.
  - Reassigns observations closer to the "splinter group" than to the "old party".

Hierarchical Clustering
Divisive



- Laplacian Graph based
  - "Normalized Cuts and Image Segmentation", Jianbo Shi and Jitendra Malik

- Minimum Spanning Tree based
  - "Hierarchical clustering in minimum spanning trees", Meichen Yu et al.

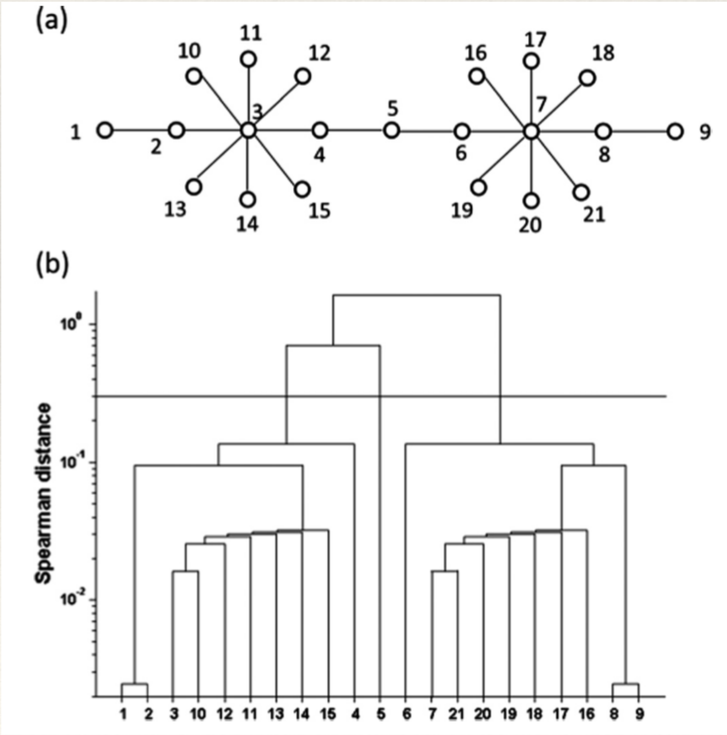


Image: Hierarchical clustering in minimum spanning trees, Meichen Yu et al.

Back to notebook

Hierarchical Clustering
Summary

- Pros:
  - Hierarchical structure is key in some problems
  - Not required to pre specify the number of clusters
  - Clusters correspond to meaningful taxonomies (language, biology etc.)
- Cons:
  - No objective function is directly optimized
  - Different definition of similarity can lead to vastly different dendrograms.
  - The algorithm assumes hierarchical structure on the data - even if such is not appropriate.

Evaluating clustering results

External clustering metrics

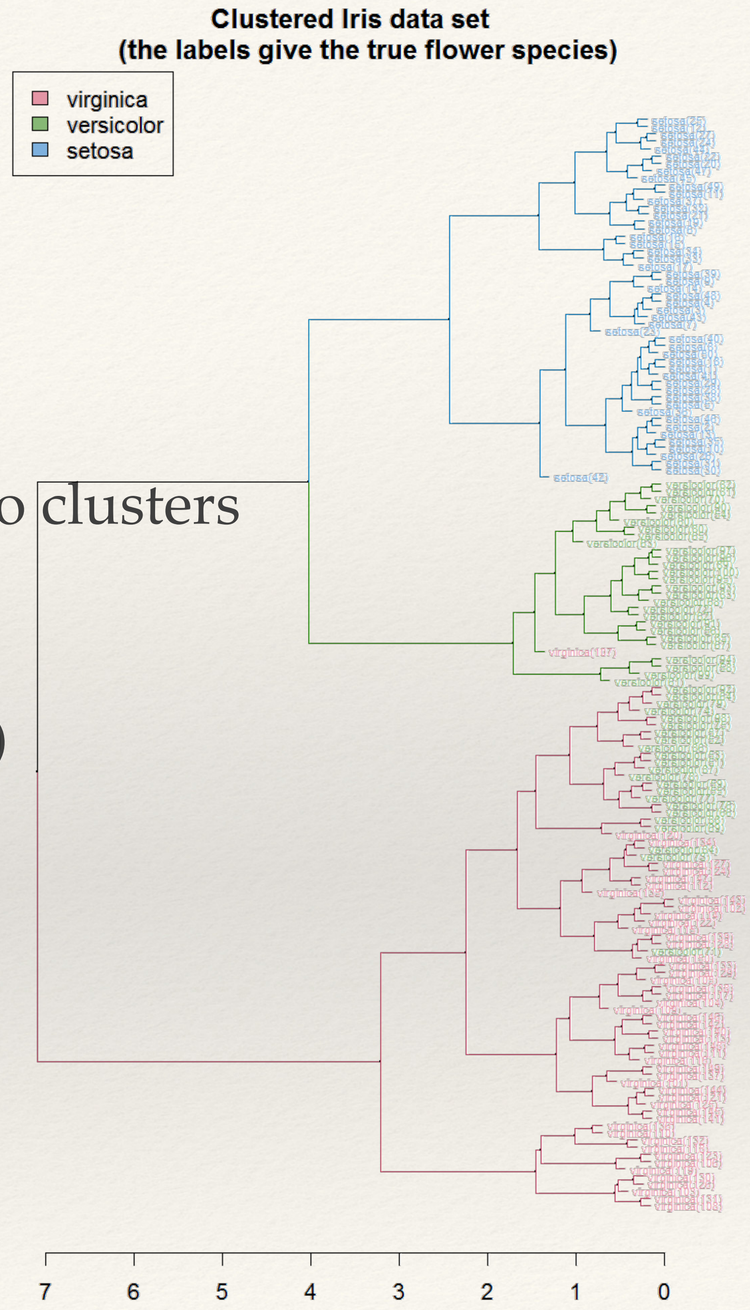
- Silhouette could give us a numerical evaluation of our clustering
- Can help us decide which model is better
- Will Silhouette prefer KMeans or DBSCAN?

?

External clustering metrics

- Silhouette is and *internal measure*
- It uses no information about the true assignment of instances to clusters
- External measures on the other hand use *external information*
- Consist of pre-classified items, often created by experts (labels)

	A	B	C	D	E
1	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
2	5.1	3.5	1.4	0.2	Iris-setosa
3	4.9	3	1.4	0.2	Iris-setosa
4	4.7	3.2	1.3	0.2	Iris-setosa
5	4.6	3.1	1.5	0.2	Iris-setosa
6	5	3.6	1.4	0.2	Iris-setosa



Rand index

Given a set of n elements $S = \{O_1, \dots, O_n\}$, and two clustering results: $X = \{x_1, \dots, x_r\}$ and $Y = \{y_1, \dots y_s\}$:

- a = number of pair of elements from S that are in same cluster in X and in same cluster in Y.
- b = number of pair of elements from S that are in different clusters in X and in different clusters in Y.
- c = number of pair of elements from S that are in same cluster in X and in different clusters in Y.
- d = number of pair of elements from S that are in different clusters in X and in same cluster in Y.

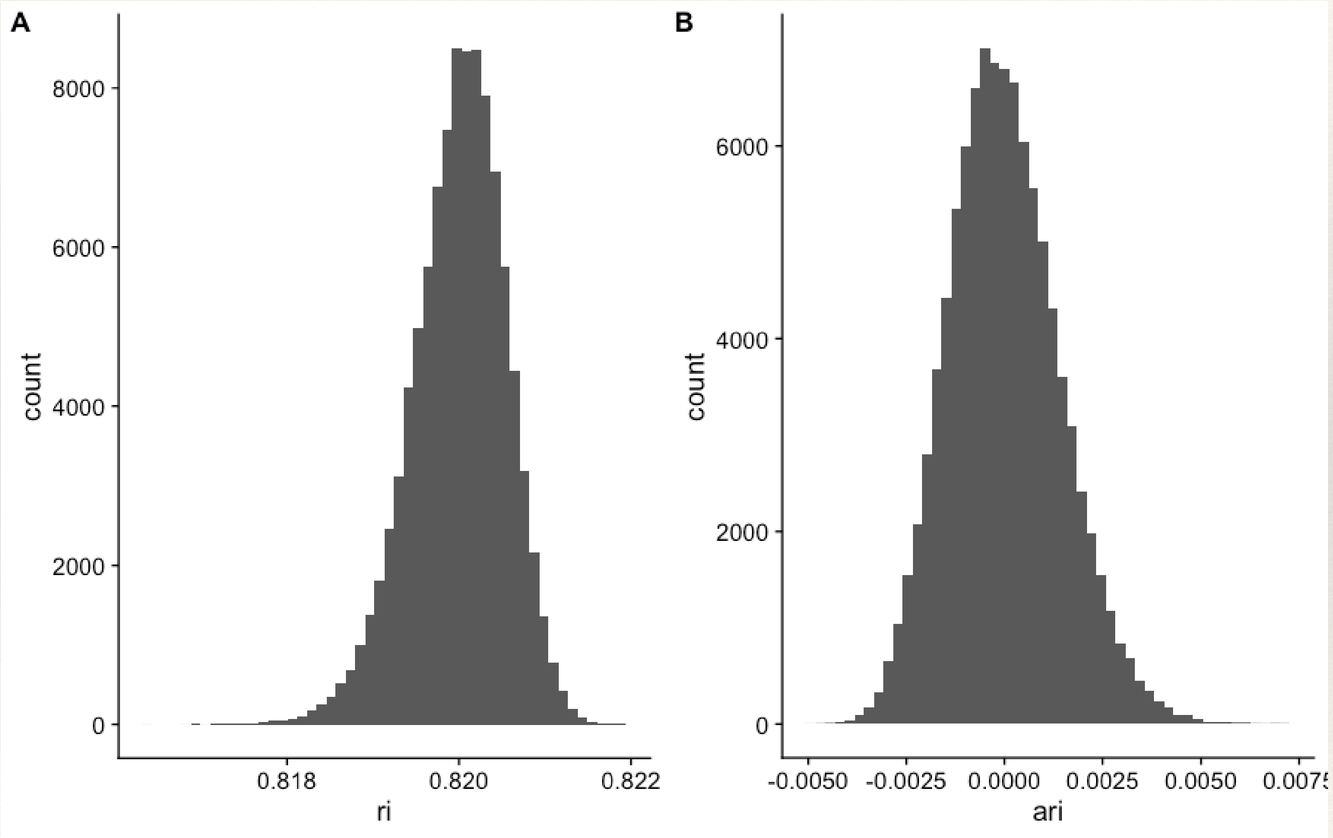
$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$

?

Rand Index

- Rand index has a value between 0 and 1,
- 0 indicating that the two data clusterings do not agree on any pair of points
- 1 indicating that the data clusterings are exactly the same.
- However, even random assignment can lead to high results

Adjusted rand index



Assigning 1,000 points to 10 clusters randomly. 100K repeats.

The average (A) RI is around 0.82, while the (B) ARI is around 0.

[Source (dave tang).]

Adjuster rand index

- The adjusted Rand index is the corrected-for-chance version of the Rand index
- Can have negative values
- Indicating bad assignments

$X \backslash Y$	Y_1	Y_2	$\dots$	Y_s	Sums
X_1	n_{11}	n_{12}	$\dots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\dots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\dots$	n_{rs}	a_r
Sums	b_1	b_2	$\dots$	b_s	

$$\overbrace{ARI}^{\text{Adjusted Index}} = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}$$

Other external measures

Purity

$$\frac{1}{N}\sum_{m\in M}\max_{d\in D}|m\cap d|$$

N = number of samples, M = set of predicted clusters, D = set of true cluster assignment

Back to notebook