Statistic 2

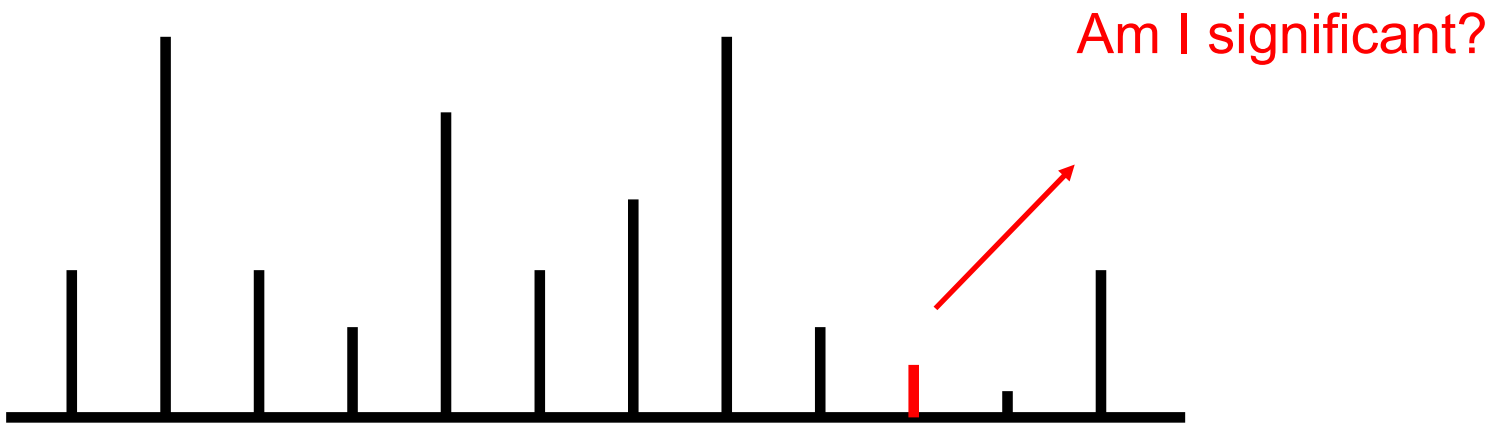
Hypothesis test2

-Hypothesis test2
Empirical p-value
Multiple hypothesis correction
Effect size
Power analysis

-Data processing
Normalization
Clustering: k-mean, hierarchical, DBSCAN, Louvain, leiden, fuzzy (overlap clustering)

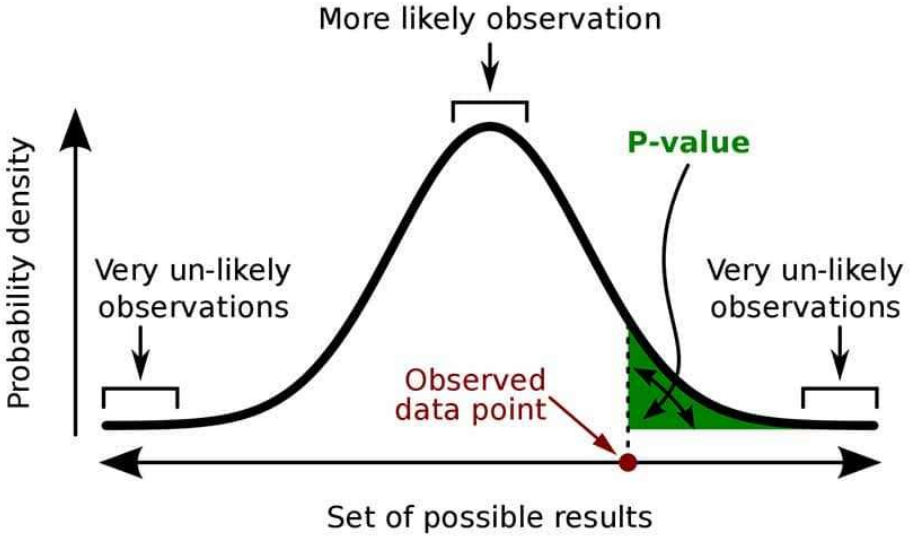
-Clustering evaluation
Silhouette
LISI
NMI(normalized mutual information)
Adjusted rand index (ARI)

Empirical p-value

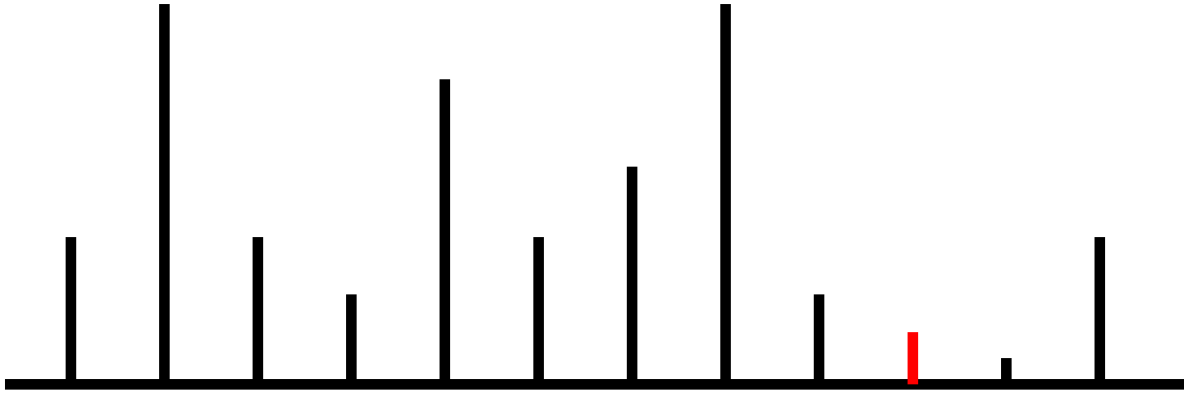


-Sometimes there are no matching distributions
→ But I want to know the p-value!
→ Rank-based approach

Empirical p-value



A **p-value** (shaded green area) is the probability of an observed (or more extreme) result assuming that the null hypothesis is true.



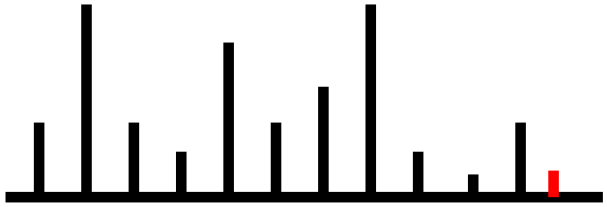
-Measure the cumulative probability for the given sample

→ Same as measuring the p-value from the distribution

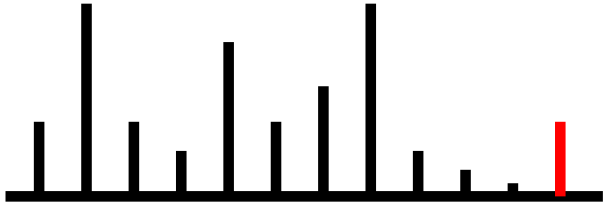
Empirical p-value

*Assume 1000 permutation test
→ Each sample: 0.001 probability

-My data: 1st rank → p value < 0.001

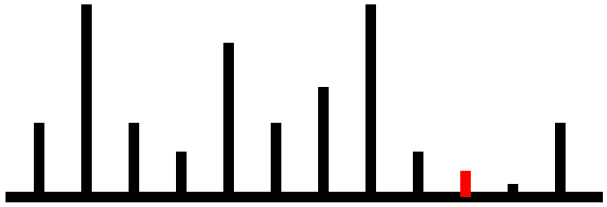


-My data: 1st rank (has tie) → p value == #tie / 1000



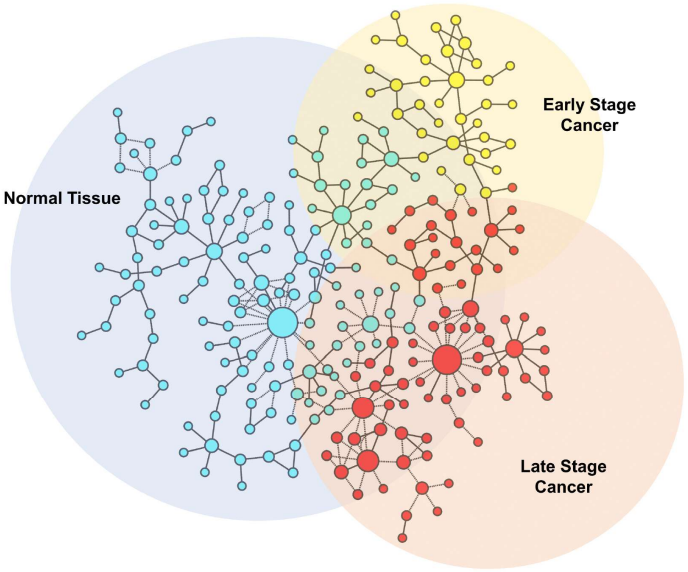
-My data: n rank → p value < rank/1000

-My data: n rank (has tie) → p value < (next rank -1)/1000



→ P value: rank/1000 (or considering self: rank + 1 / 1000 + 1)

Empirical p-value



-Gene network

→ My geneset: significant connectivity?

→ Comparing between random geneset

→ Frequently used in random permutation

Multiple hypothesis correction

Actual truth

Decision	H0 True	H0 False
Fail to reject H0	Correct decision $1 - \alpha$	Incorrect decision type II error (β)
Reject H0	Incorrect decision type I error (α)	Correct decision $1 - \beta$ Power

		Actual: Positive	Actual: Negative
Predicted	Positive	True Positive	False Positive
Predicted	Negative	False Negative	True Negative

Multiple hypothesis correction

P value → Adjusted p value
- Multiple hypothesis correction
- Avoid lucky hits my multiple testing

Actual truth

Decision		H0 True	H0 False
	Fail to reject H0	Correct decision 1 - α	Incorrect decision type II error (β)
	Reject H0	Incorrect decision type I error (α)	Correct decision 1 -β Power

$$p(making\ a\ type\ I\ error) = \alpha$$

$$p(making\ no\ type\ I\ error) = 1 - \alpha$$

$$p(making\ no\ type\ I\ error\ in\ n\ tests) = (1 - \alpha)^n$$

$$p(making\ at\ least\ 1\ type\ I\ error\ in\ n\ tests) = 1 - (1 - \alpha)^n$$

$$p(making\ at\ least\ 1\ type\ I\ error\ in\ 1\ test) = 1 - (1 - \alpha)^1 = 0.05$$

$$p(making\ at\ least\ 1\ type\ I\ error\ in\ 50\ test) = 1 - (1 - \alpha)^{50} = 0.92$$

Multiple hypothesis correction

N multiple hypothesis correction
*Bonferroni method
q-value (adjusted p-value) = p-value * N

*Benjamini-Hochberg (BH) method
q-value: p-value * (N/rank)
→ Sometimes order of p-value changes
→ Collapse the q-value

Effect size

*a value measuring the strength of the relationship between two variables

-Cohen's D

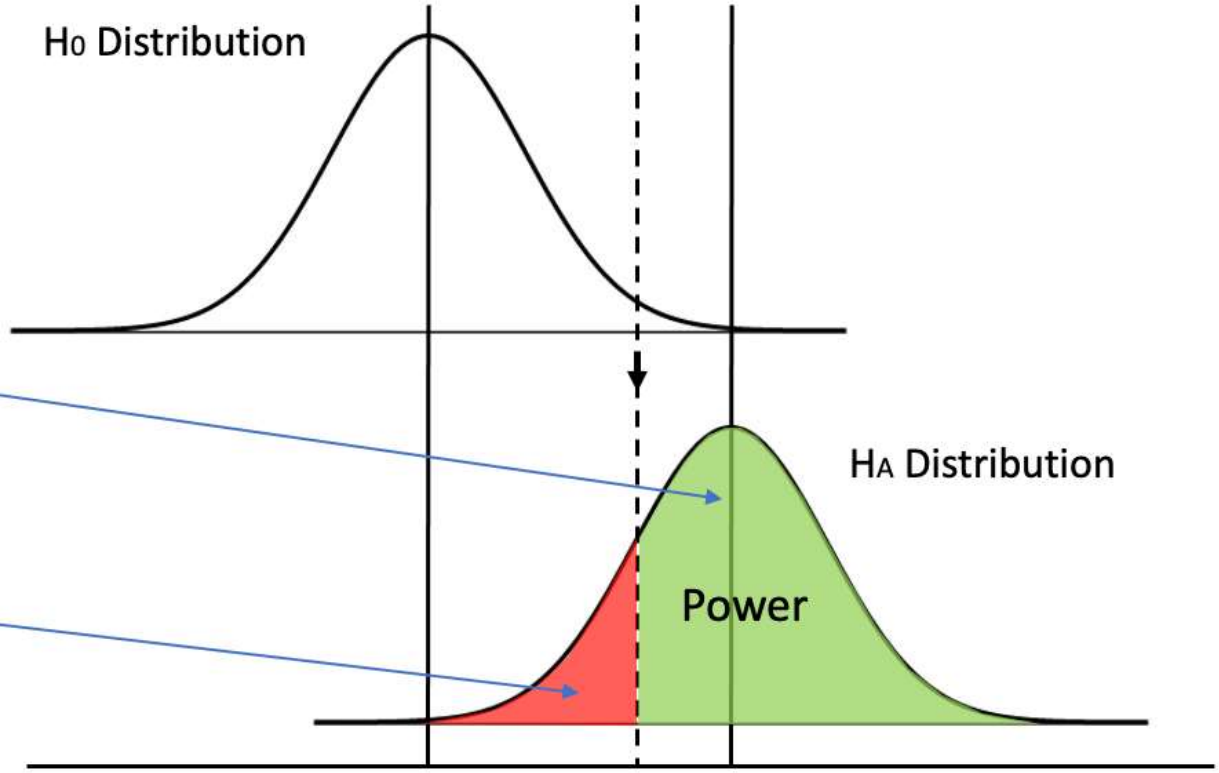
$$d = \frac{\bar{x}_1 - \bar{x}_2}{s}$$

Relative size	Effect size	% of control group below the mean of experimental group
	0.0	50%
Small	0.2	58%
Medium	0.5	69%
Large	0.8	79%
	1.4	92%

Power analysis

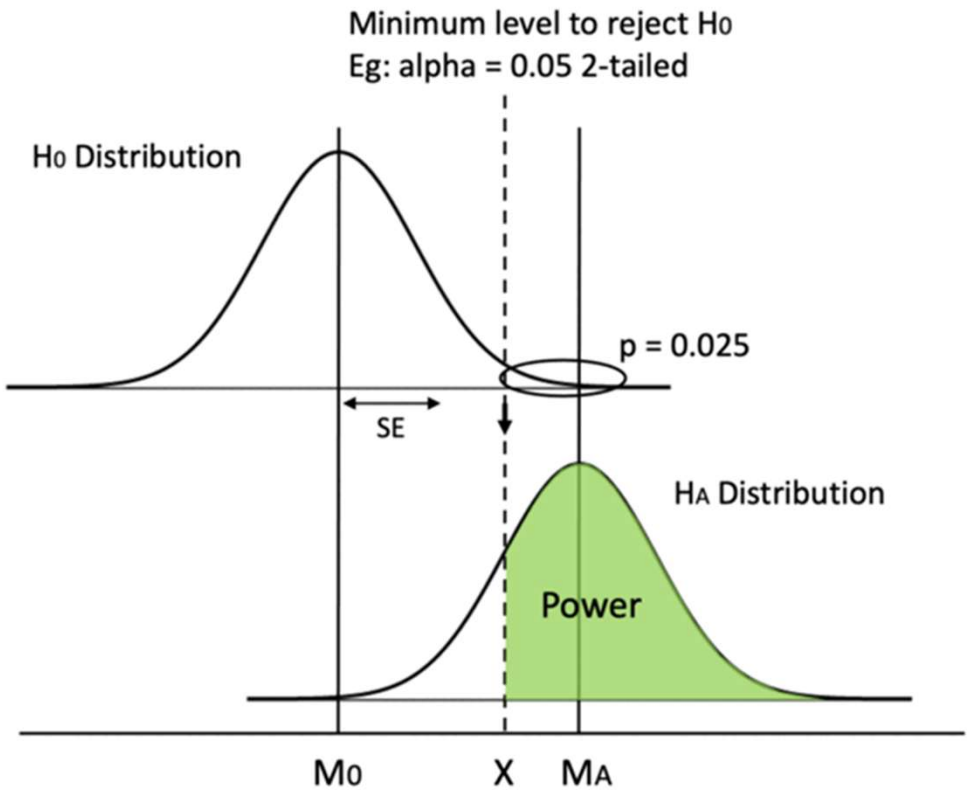
*the calculation used to **estimate** the smallest sample size needed for an experiment, given a required significance level, statistical power, and effect size

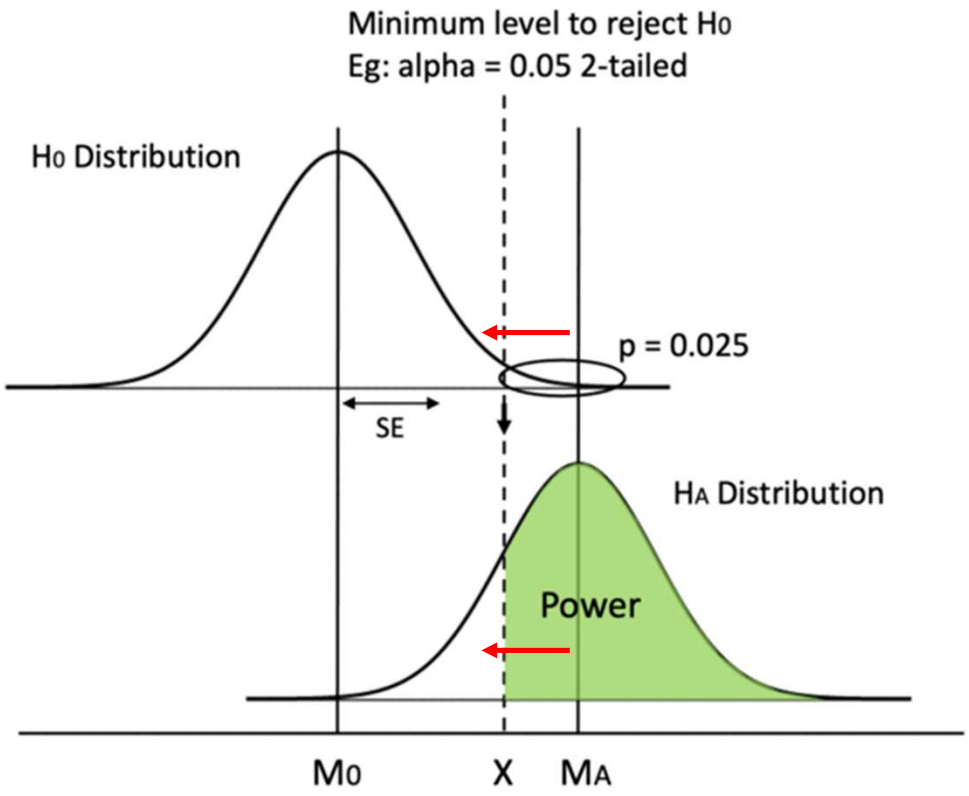
	H_0 is True	H_A is True
Reject H_0	Type I Error False Positive α (alpha)	Correct True Positive 1-β (power)
Fail to reject H_0	Correct True Negative 1-α	Type II Error False Nagative β (beta)



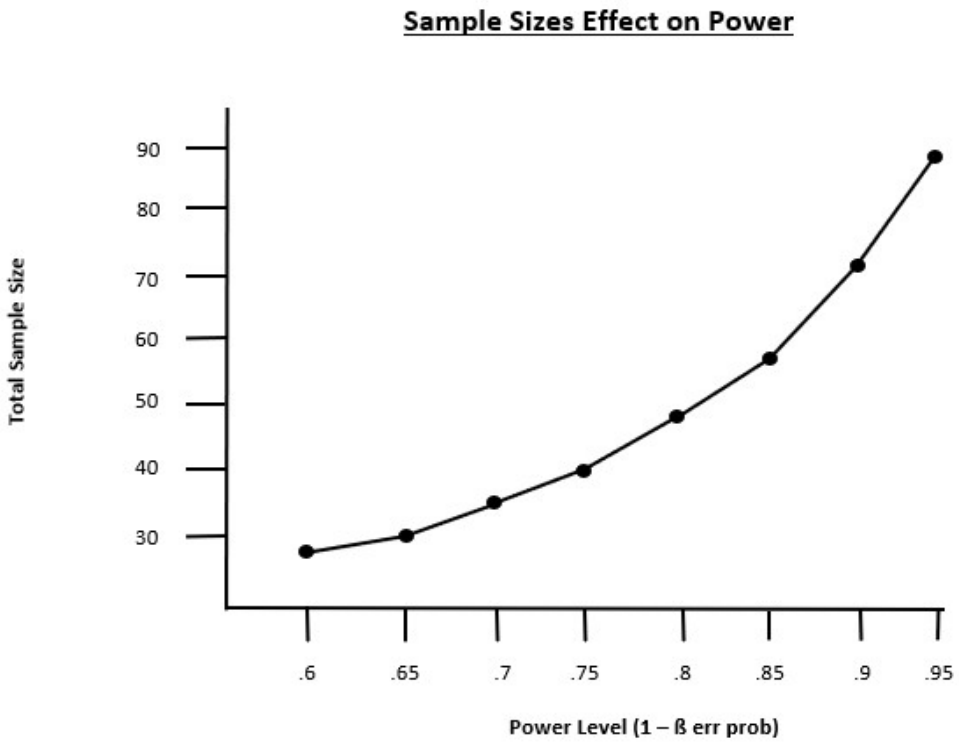
Power analysis

*How to get enough p-value (usually 0.05) → increase sample size N





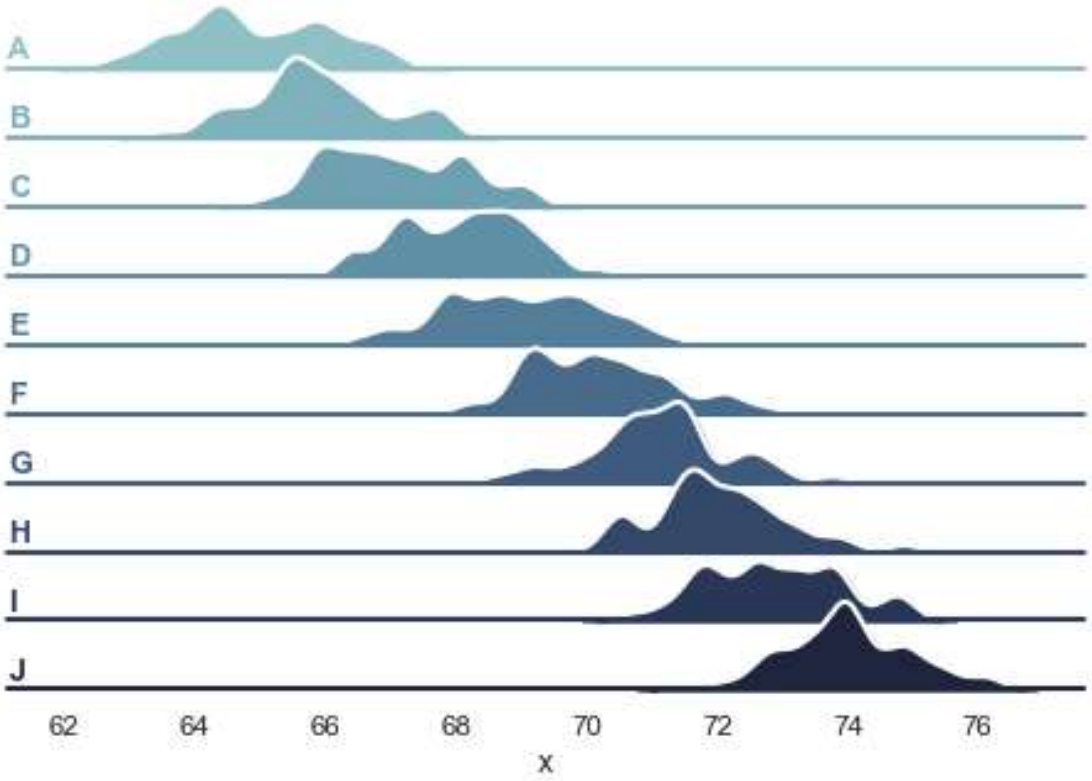
Power analysis



Note: As the sample size increases in the model, so does power.

Data processing

Normalization



*Single data → no need to normalize
*Multiple data → each data may have different space
→ How to analyze them together?
→ Make them into the same “space”

Normalization

*min-max normalization

$$x = \frac{x - x_{min}}{x_{max} - x_{min}}$$

→ Range into 0~1

*Z-score normalization (standardization)

$$x_{new} = \frac{x_i - mean(x)}{stdev(x)}$$

→ mean:0, s.d.:1, gaussian distribution

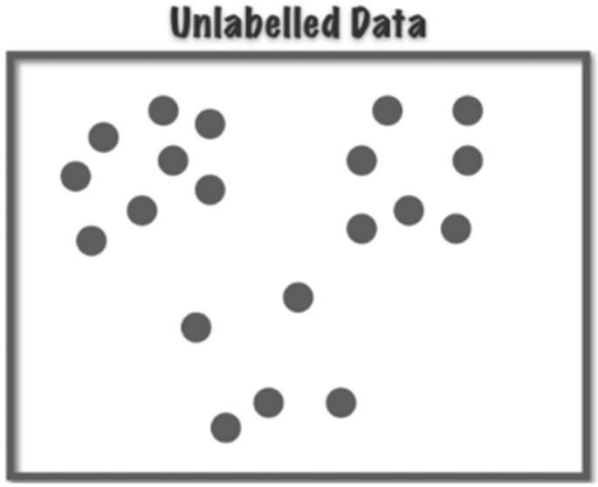
*Robust scaling

$$x_{new} = \frac{x_i - median(x)}{Q3 - Q1}$$

→ Robust to outliers

Clustering

Clustering



Distance measurement for clustering

- **Euclidean distance** (a *straight line* distance in *n*-dimensional space)

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

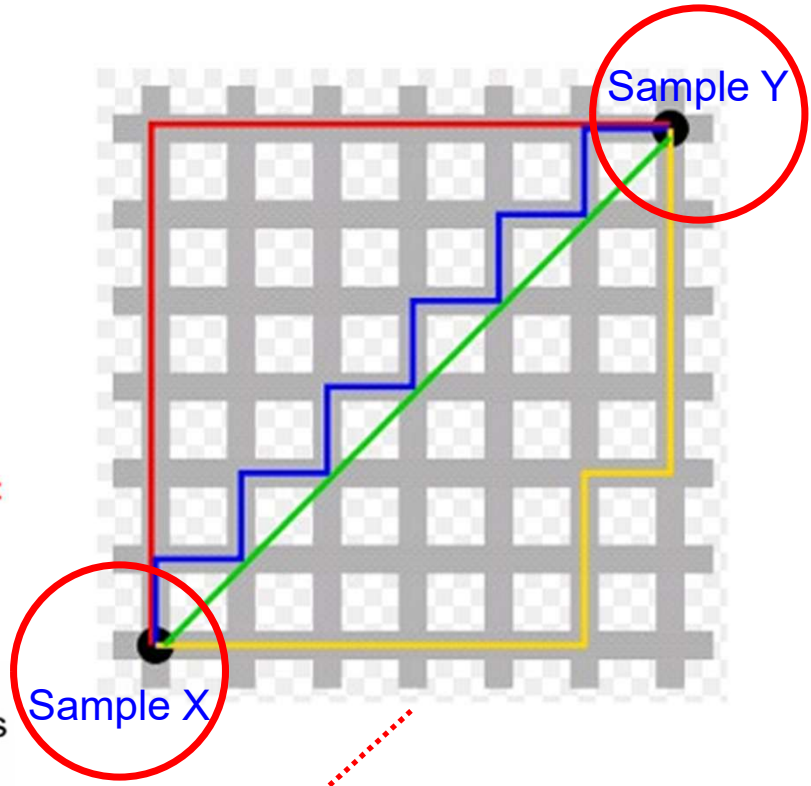
$$d = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}$$,if you don't want to increase the distance with the addition of more dimensions.

- Euclidean distances **may underestimate** join differences such as differences in two correlated expression .
- Therefore, use Euclidean distance **if** you believe that your **dimensions (variables) are not independent**.

- **Manhattan distance** (= city block distance, L1 distance, rectilinear distance, taxicab metric): *distances measured parallel to dimensional axes*. After NY city's grid like street pattern.

$$d = \sum_{i=1}^{n}|x_i - y_i|$$

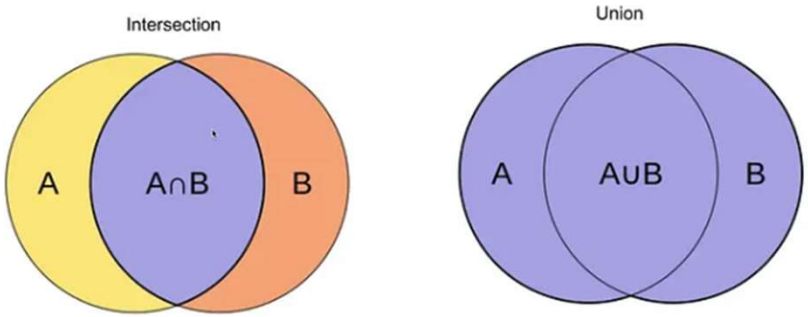
$$d = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i|$$,if you don't want to increase the distance with the addition of more dimensions.



Gene expression space
i: each gene
n: the number of genes

Distance measurement for clustering

*Jaccard coefficient → measure the overlap



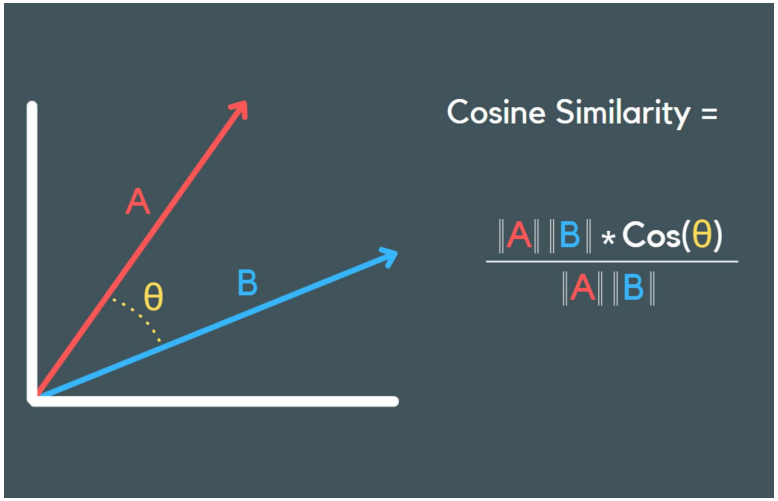
$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

*Cosine similarity

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

$$\text{cosine similarity} = S_C(A,B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

→ Ignore the magnitude



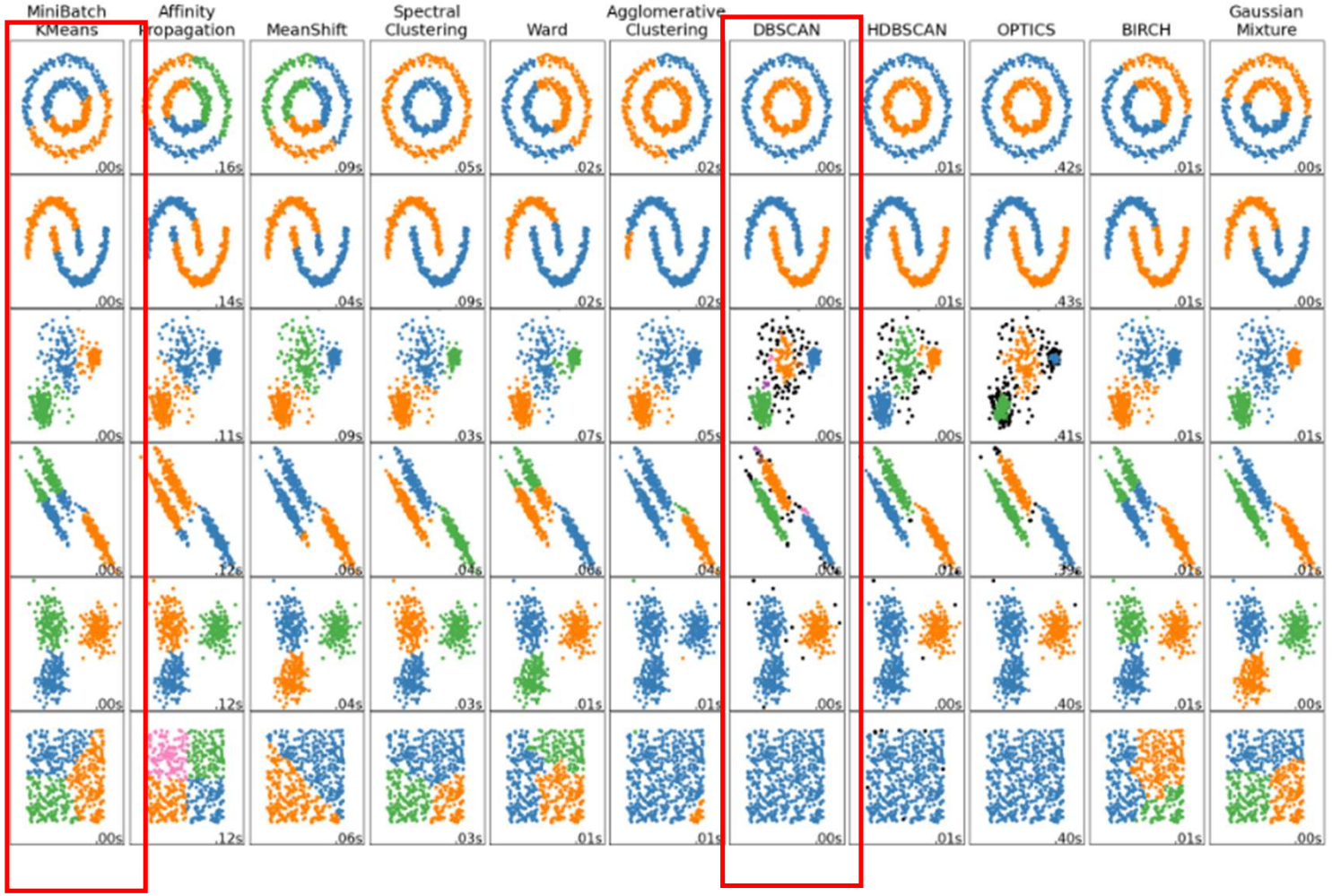
Distance measurement for clustering

*Hamming distance
-binary data

10101, 11110 → distance: 3 (different bit)

10101
11110

Clustering

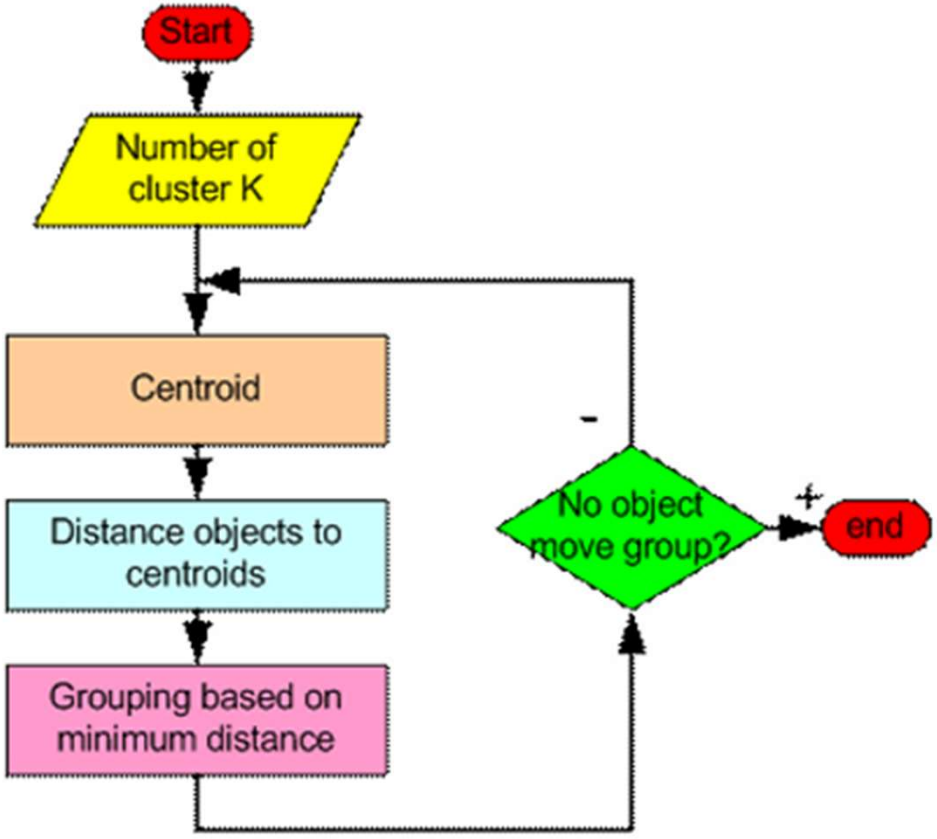

MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
HDBSCAN
OPTICS
BIRCH
Gaussian Mixture

K-mean clustering

K-mean clustering (linear approach)

- **Algorithm**

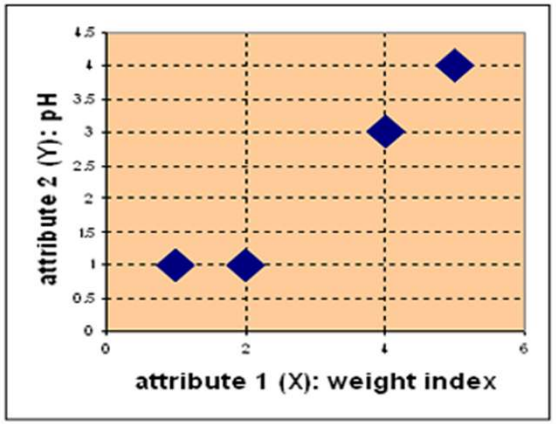
1. ***Initial k***: Decide the number of cluster (*k*)
2. ***Initial value of centroids***: Choose *k* centroids randomly.
3. ***Objects-centroids distance***: Calculate the distance between cluster centroid to each object (with any distance metric).
4. ***Objects clustering***: Assign each object based on minimum distance.
5. ***Determine new centroids***: New centroid moves to mean value of all member objects.
6. ***Iterate steps 3-5***: Until no object change centroid membership.

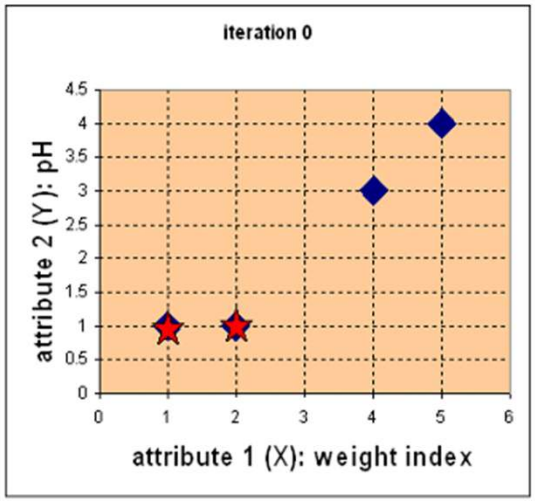


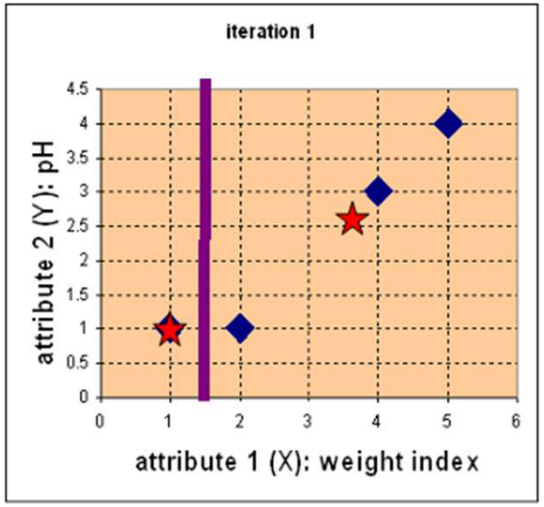
K-mean clustering

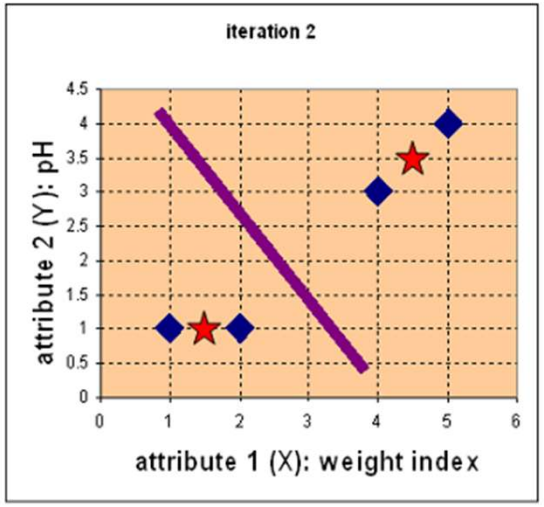
Demonstration

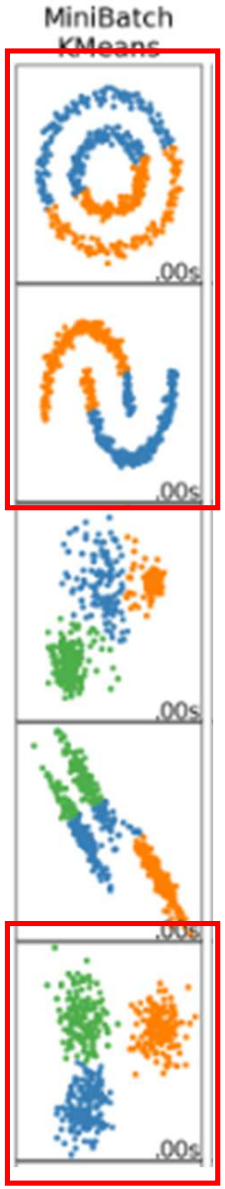
Diamond: object
Star: centroid (k=2)







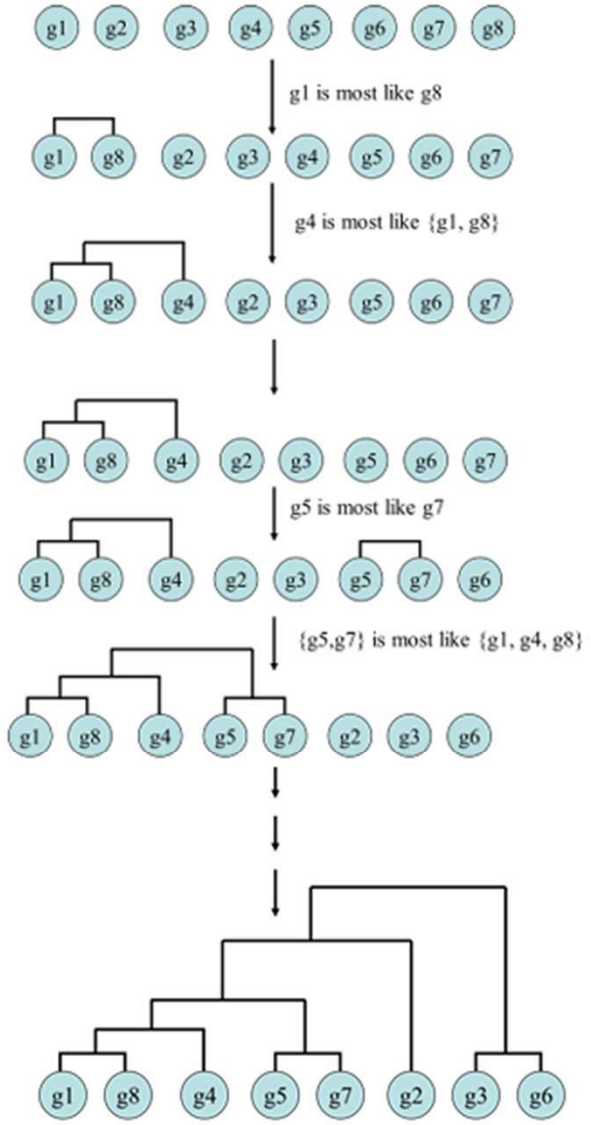




Hierarchical clustering

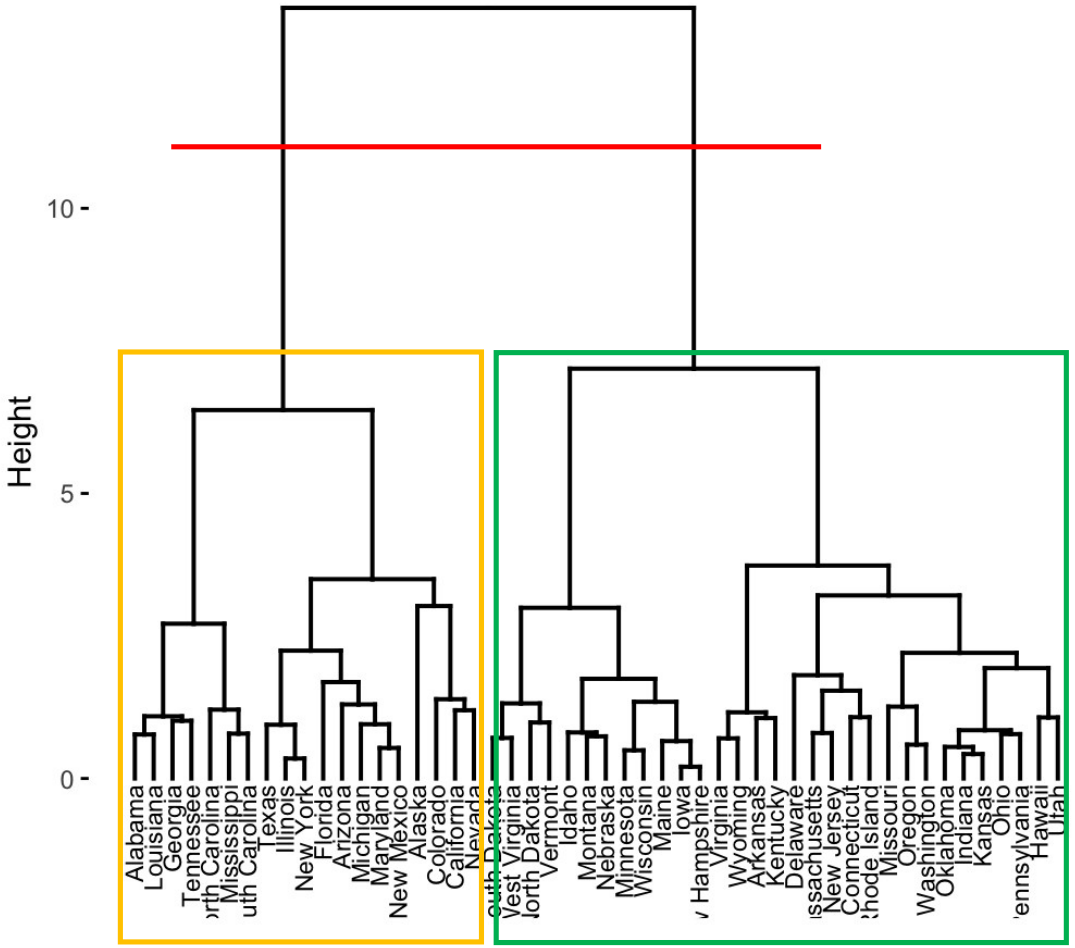
- **Algorithm**

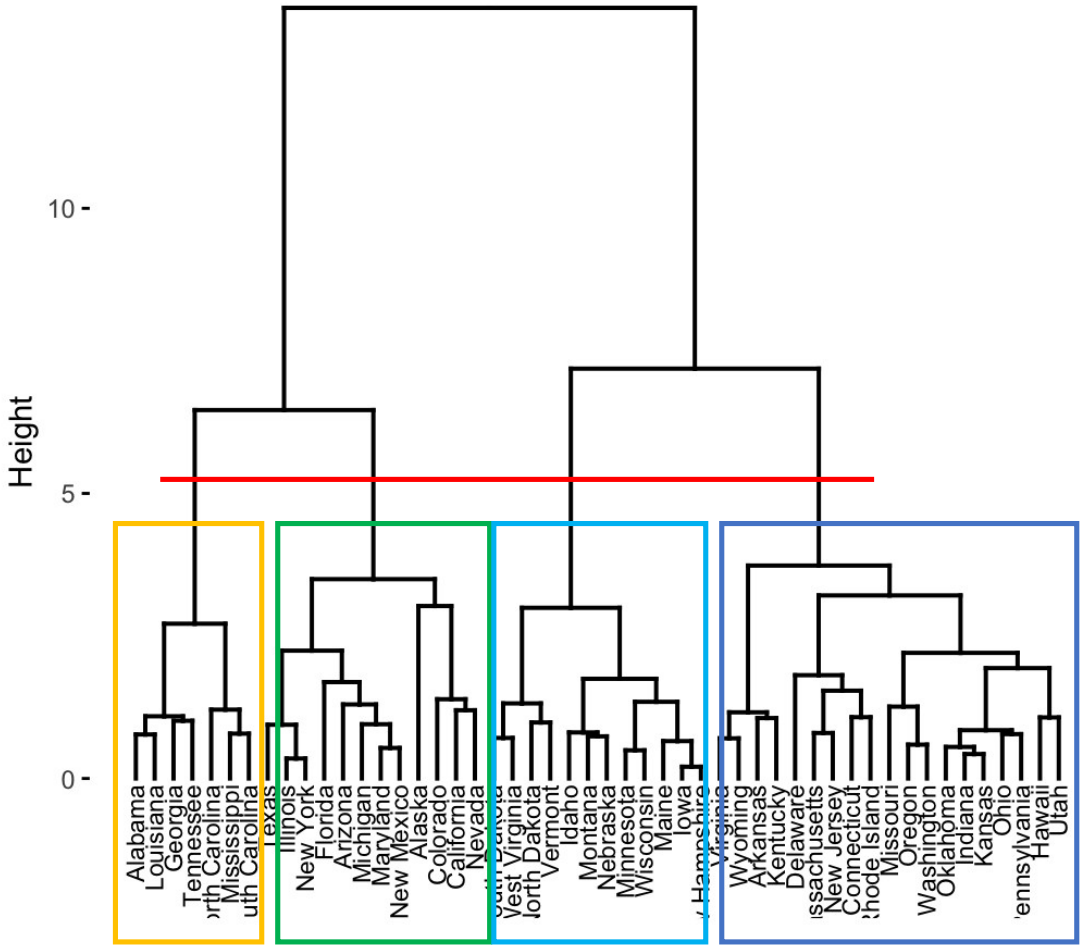
1. Initialize each single gene as a cluster
2. The pairwise distance matrix is calculated for all of the genes to be clustered.
3. The distance matrix is searched for the two most similar genes or clusters.
4. The two selected clusters are merged to produce a new cluster that now contains at least two objects.
5. The distances are calculated between this new cluster and all other clusters. There is no need to calculate all distances as only those involving the new cluster have changed.
6. Steps 3-5 are repeated until all objects are in one cluster.



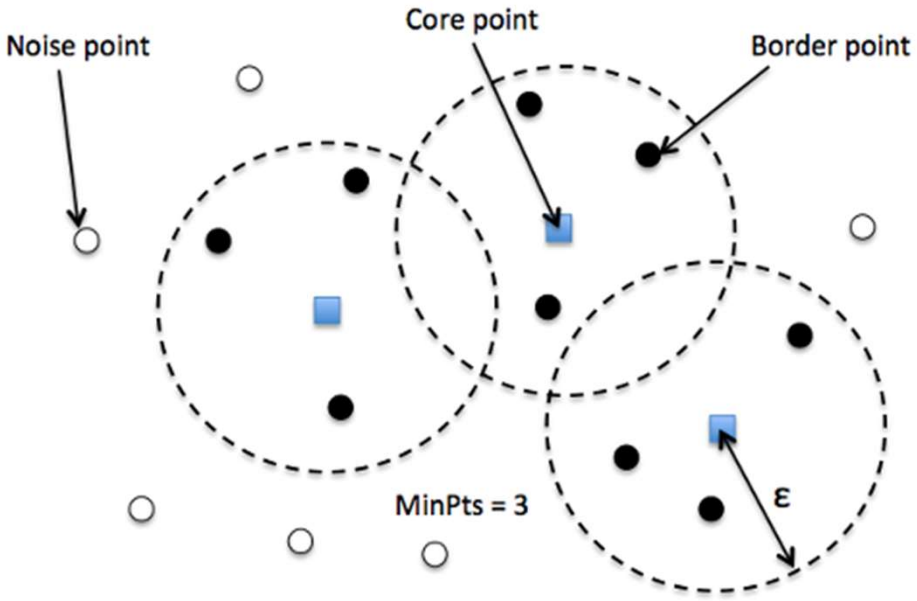
Hierarchical clustering

Tree cutting

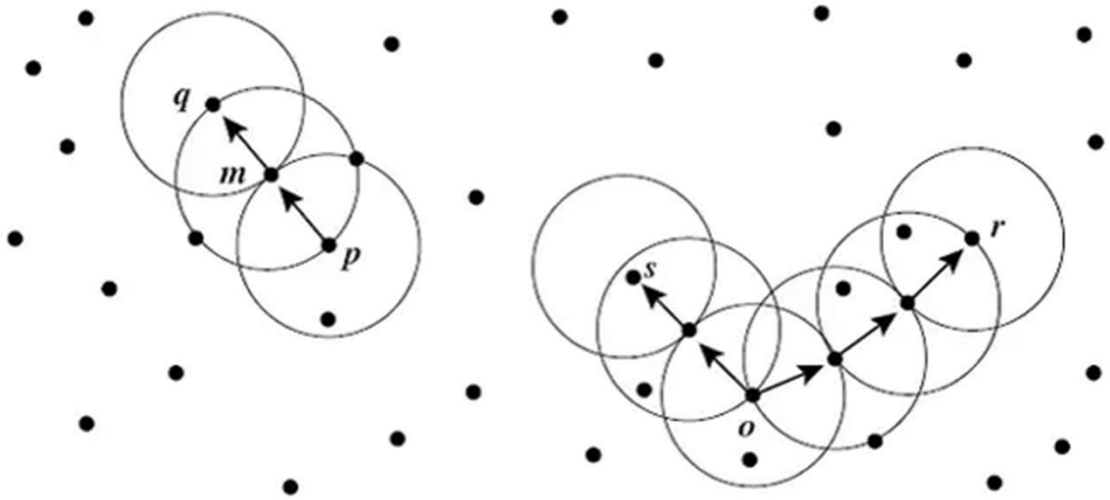




- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
  - Clustering algorithm that divides a dataset into **subgroups of high density regions**.
  - Two parameters required for DBSCAN:
    - **Epsilon (ε)**: a distance parameter that defines the radius to search for nearby neighbors
    - **MinPts:** minimum number of other points required to form a cluster
  - **Core point** – a point that **has at least the minPts of other points within its ε radius**.
  - **Border point** – a point within the **ε** radius of a *core point* BUT has less than the **minPts** within its own **ε** radius
  - **Noise point** – a point that is neither a *core point* or a *border point*



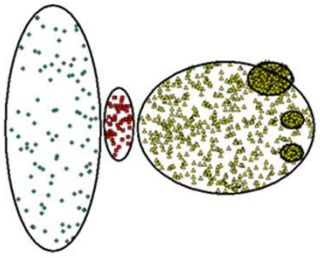
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
  - Each core point forms a cluster together with the points that are reachable within its **ε** radius.
  - **Two points are** considered "***directly density-reachable***" **if one of the points is a core point and the other point is within its ε radius**.
  - **Larger clusters** are formed when directly density-reachable points are **chained together**.
  - In the example image below, there are **two clusters**:



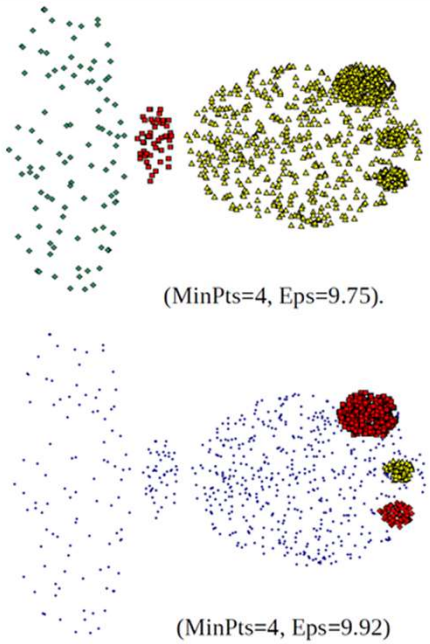
1. If minPts = 3, **p** is directly density-reachable from **m**, which is directly density-reachable from **q**. The sets of points within the **ε** radius of **p** → **m** → **q** form one cluster
2. **r** and **s** are indirectly density-reachable through a path of 4 core points. The set of points within the **ε** radius of this chain forms another cluster.

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

  - The DBSCAN algorithm **repeats the following process until all points have been assigned to a cluster** or are labeled as visited:
    1. **Arbitrarily select a point P**.
    2. **Retrieve all points** directly density-reachable from P **with respect to ε**.
    3. **If P is a core point**, a cluster is formed. Find recursively all its density connected points and **assign them to the same cluster as P**.
    4. **If P is not a core point**, DBSCAN **iterates through the remaining unvisited points** in the dataset.
  - DBSCAN does not require us to specify the number of clusters.
  - It can handle clusters of arbitrarily shapes and sizes.
  - It is robust to noise.

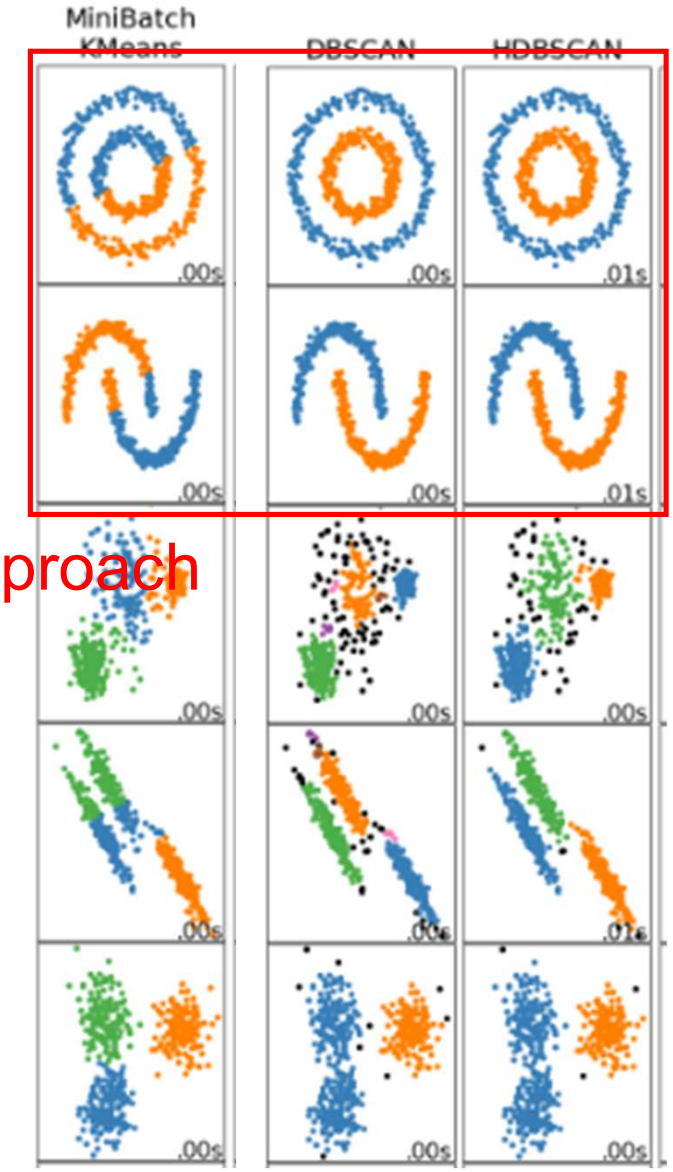


Original Points



(MinPts=4, Eps=9.75).

(MinPts=4, Eps=9.92)

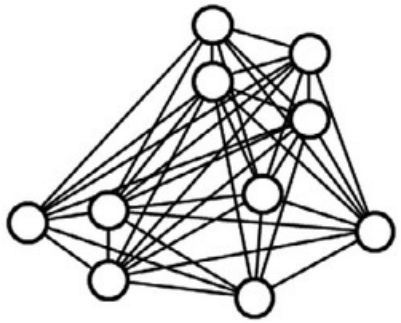


Nonlinear approach

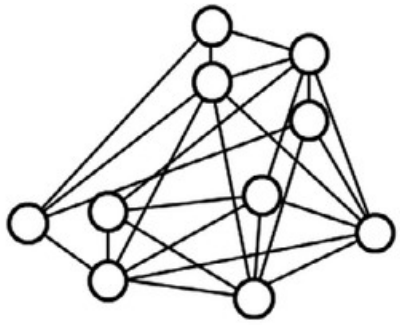
Louvain Clustering

-Network-based clustering
-Modularity (given community)

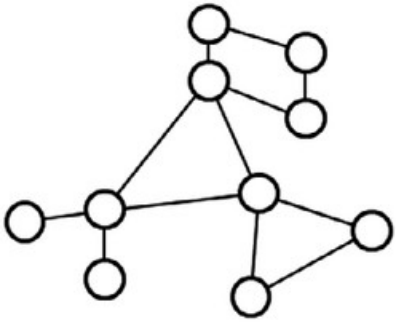
$$Q = \frac{1}{2M}\sum_{i,\ j}^{N}(a_{ij} - \langle t_{ij}\rangle)\delta[C(i),\ C(j)]$$



Modularity ≈ -1
(almost complete network)



Modularity ≈ 0
(random network)

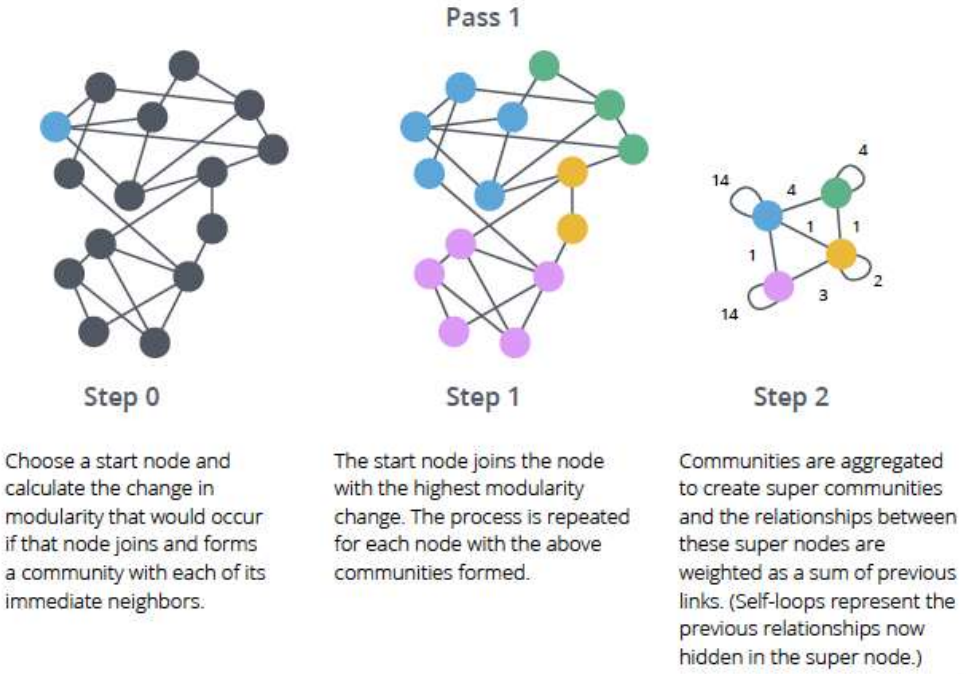


Modularity ≈ 1
(modular network)

M: # total edge
N: # node
a_ij: 0,1
t_ij: random node permutation → <t_ij>: expected value = ki * kj / 2m
C(i): community
δ: community: 0,1

→ Compare between real connection – random connection

Louvain Clustering



Louvain Modularity Algorithm

Pass1
-each node → each community
-each node → change into other communities
→ Higher modularity
-each community → supernode
+ edges within each community: weighted edge
→ Repeat until modularity stops

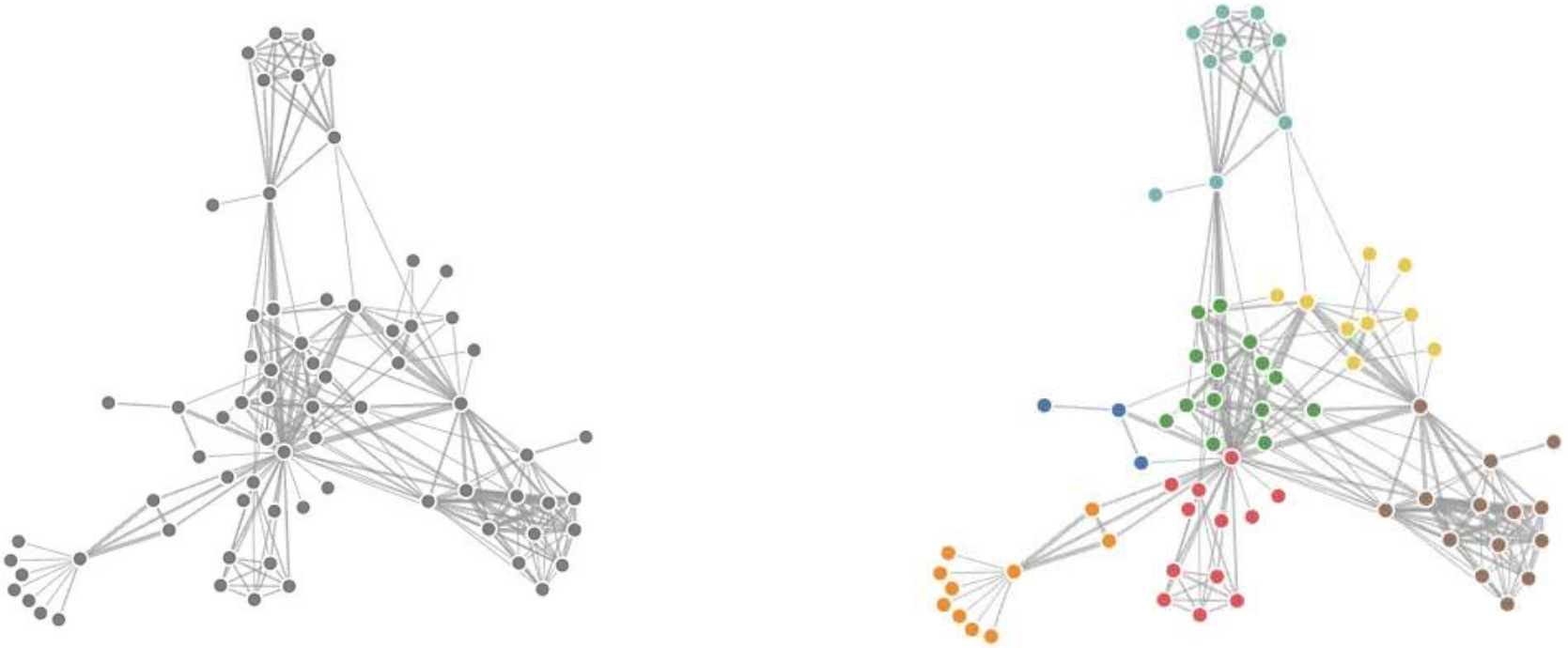
Resolution parameter → control the granularity of the community

High resolution: many clusters

$$\Delta Q = \left[\frac{\Sigma_{in} + 2k_{i,in}}{2m} - \left(\frac{\Sigma_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

$$= \frac{k_{i,in}}{2m} - \gamma\frac{\Sigma_{tot} \cdot k_i}{2m^2}$$

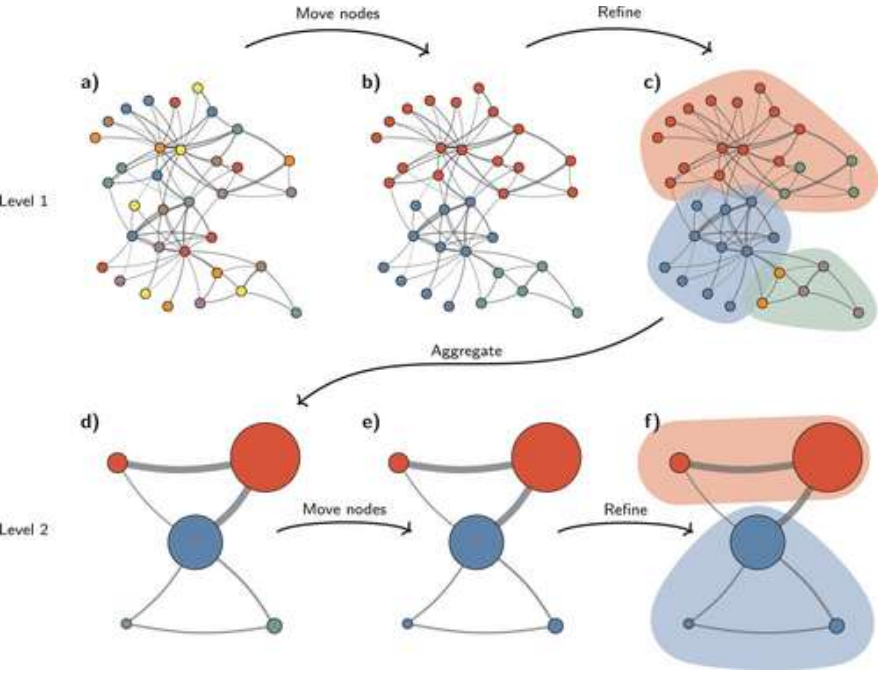
Louvain Clustering



Leiden Clustering

-Improve Louvain clustering having a low modularity community
(because **poorly-connected communities**
→ merging of smaller communities into larger communities)

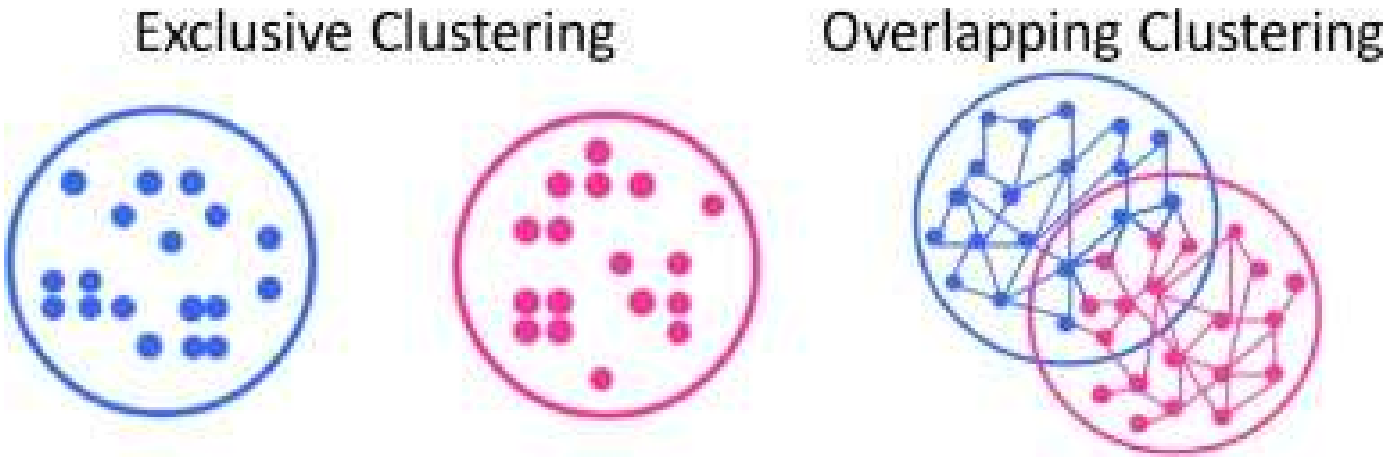
-Improvement: stable, good community, fast



*Iteration
-each node → each community
-each node → change into other communities
→ Higher modularity
→ Refine the lower connectivity community

→ Same as louvain

Overlapping clustering



*Fuzzy c-mean clustering
-k-mean → fuzzy (p>1) → (1: k-mean)
→ Maximize the probability of each k-cluster
→ K-mean algorithm

$$\sum_{j=1}^{k}\sum_{i=1}^{n} u_{ij}^{p}(x_j - c_j)^2$$

-x: each sample
-c: cluster centroid

Clustering evaluation

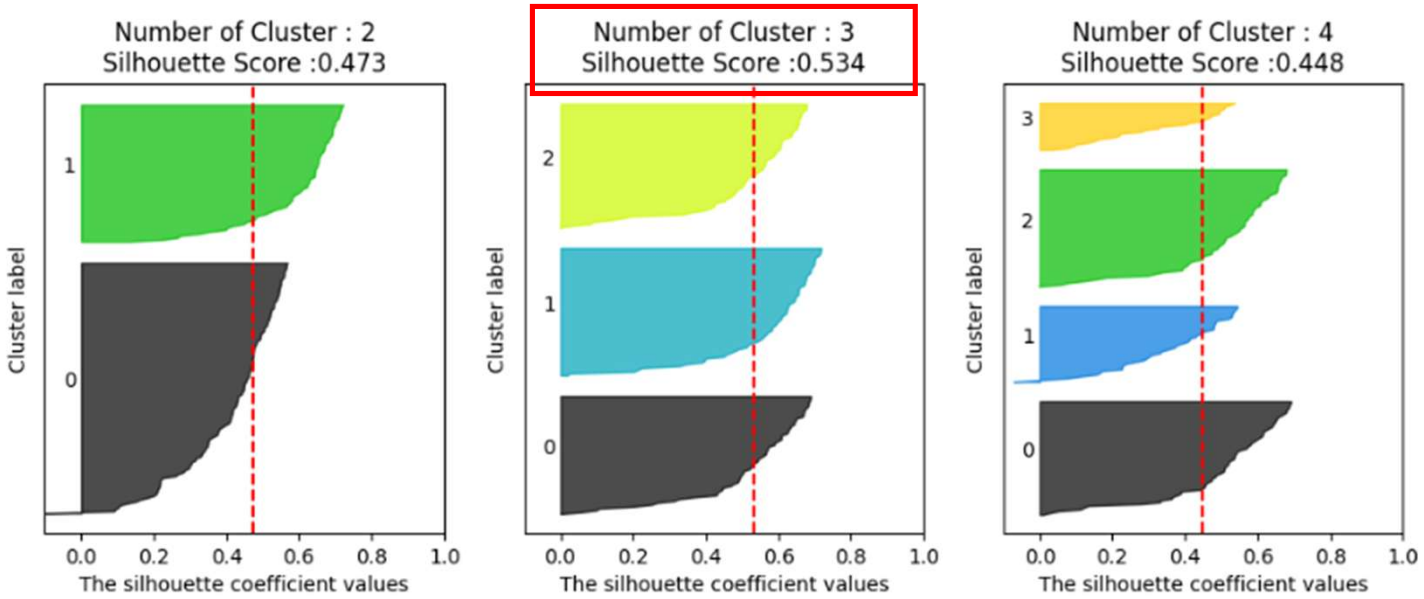
-Silhouette Coefficient

b: distance to closest neighbors

a: distance to self cluster

s → high: far from neighbors → separated

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



Clustering evaluation

-LISI(Local Inverse Simpson's Index)
N(x): k-nearest neighbors
C: label (1~L)

High LISI → mixed
Low LISI → separated

$$p_i = \frac{n_i}{k}$$

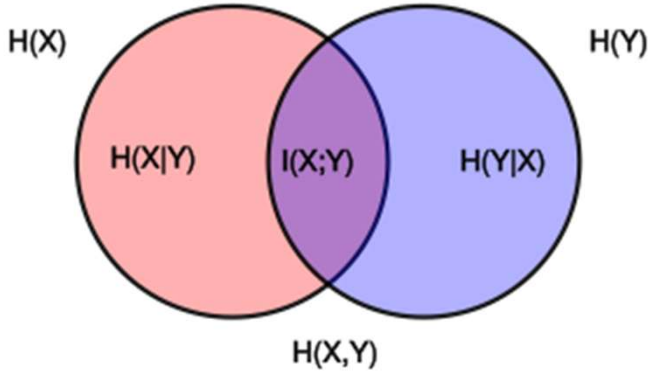
ni: # label i neighbor
k: # total neighbor

$$\mathrm{LISI}(x) = \frac{1}{\sum_{i=1}^{L} p_i^2}$$

Clustering evaluation

-Mutual information:

$$I(X,Y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$



→ 0: two cluster: independent
→ <1: partial overlap
→ 1: equal cluster

-NMI(normalized mutual information): H(x) high → I(x,y) will increase → normalization by magnitude

$$\mathrm{NMI}(X,Y) = \frac{I(X,Y)}{\sqrt{H(X)\,H(Y)}} \qquad H(X) = -\sum_x p(x) \log p(x)$$

Clustering evaluation

Supervised approach: comparing between real membership and clustering

-Rand Index (RI):

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

-Adjusted rand index (ARI):

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathrm{E[RI]}}{\max(\mathrm{RI}) - \mathrm{E[RI]}}$$

→ Vs Expected RI (random)

Expected: true possible pairs * inferred possible pairs / all possible pairs

Max: mean(true possible pairs + inferred possible pairs)

1: perfect clustering

0: random

<0: poor

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}$$

Gender

Political Parties

	Male	Female	Total
Democrat	20	22	42
Republican	21	16	37
Independent	19	32	51
Total	60	70	130

$\binom{22}{2}$

Possible pairs with 22 samples