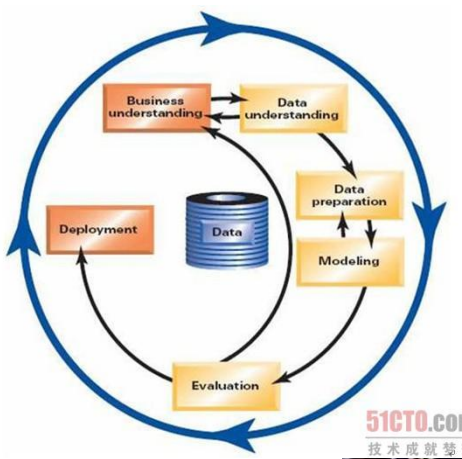
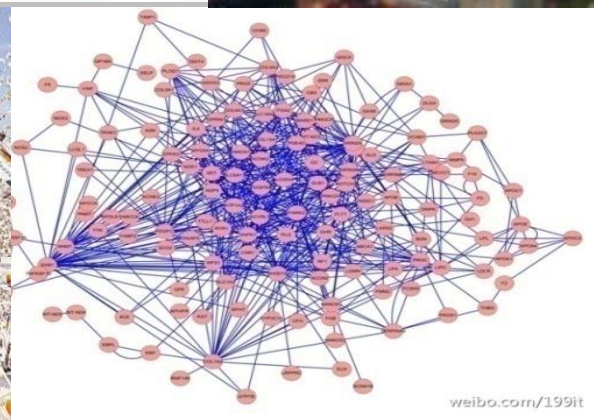




Chapter 10 Hierarchical Algorithm

2025 Autumn

Lei Sun

01 Agglomerative Clustering

02 Linkage

03 Compare between Kmeans and HC

04 Combination

Cluster Dendrogram
Height
0
2
4
6
8
10
12
14
Alabama
Louisiana
Georgia
Tennessee
North Carolina
Mississippi
South Carolina
Texas
Illinois
New York
Florida
Arizona
Michigan
Maryland
New Mexico
Alaska
Colorado
California
Nevada
South Dakota
West Virginia
North Dakota
Vermont
Idaho
Montana
Nebraska
Minnesota
Wisconsin
Maine
Iowa
New Hampshire
Virginia
Wyoming
Arkansas
Kentucky
Delaware
Massachusetts
New Jersey
Connecticut
Rhode Island
Missouri
Oregon
Washington
Oklahoma
Indiana
Kansas
Ohio
Pennsylvania
Hawaii
Utah

Recall: two properties of K-means clustering:

1. It fits exactly K clusters (as specified).
2. Final clustering assignment depends on the chosen initial cluster centers.

Given pairwise dissimilarities d_{ij} between data points, hierarchical clustering produces a consistent result, without the need to choose initial starting positions (number of clusters).

The catch: we need to choose a way to measure the dissimilarity between groups, called the linkage(链).

Given the linkage, hierarchical clustering produces a sequence clustering assignments. At one end, all points are in their own cluster, at the other end, all points are in one cluster.

Basic Idea

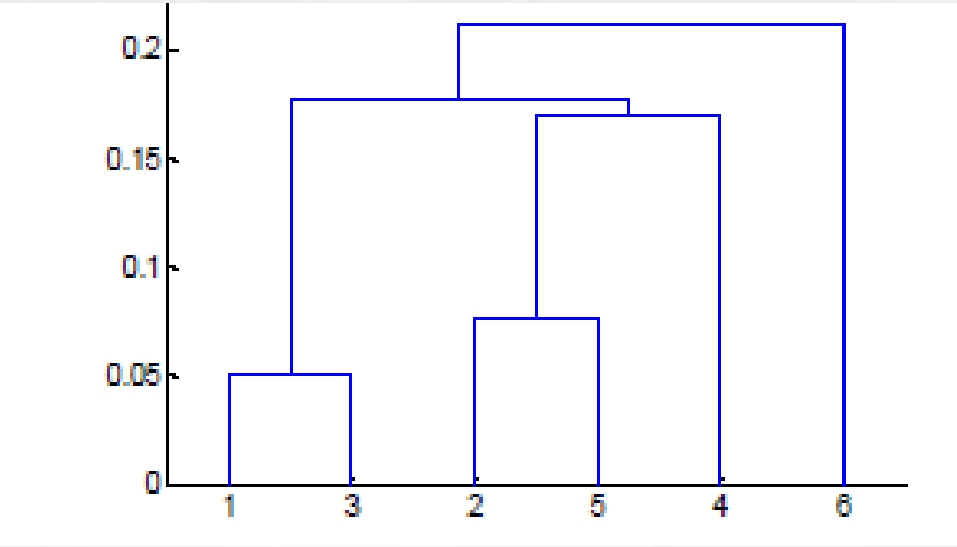
the objective of Hierarchical Clustering

Produce a hierarchical series of nested clusters. A diagram called **Dendrogram** (A Dendrogram is a tree-like diagram that statistics the sequences of merges or splits) graphically represents this hierarchy and is an inverted tree (倒置) that describes the order in which factors are merged (bottom-up view) or clusters are broken up (top-down view).

Basic Idea

Hierarchical clustering methods can be further classified as either agglomerative or divisive, depending on whether the hierarchical decomposition is formed in a buttom-up (merging) or top-down (splitting) fashion.

Can be visualized as a **dendrogram**



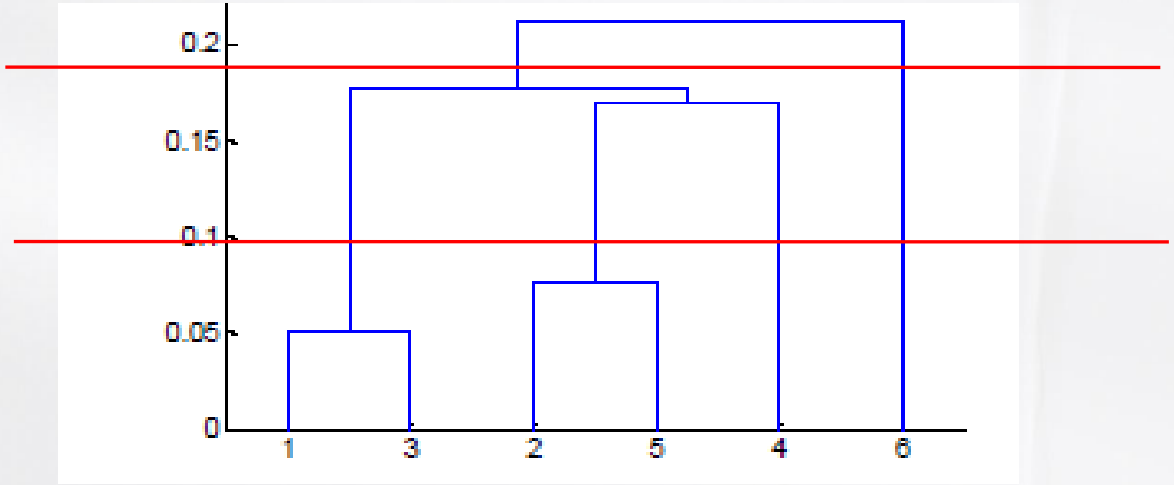
Basic Idea

Dendrogram: convenient graphic to display a hierarchical sequence of clustering assignments. This is simply a tree where

- Each node represents a group
- Each leaf node is a singleton (i.e., a group containing a single data point)
- Root node is the group containing the whole data set
- Each internal node has two daughter nodes (children), representing the groups that were merged to form it

Remember: the choice of linkage determines how we measure dissimilarity between groups of points.

Basic Idea



No assumptions on the number of clusters

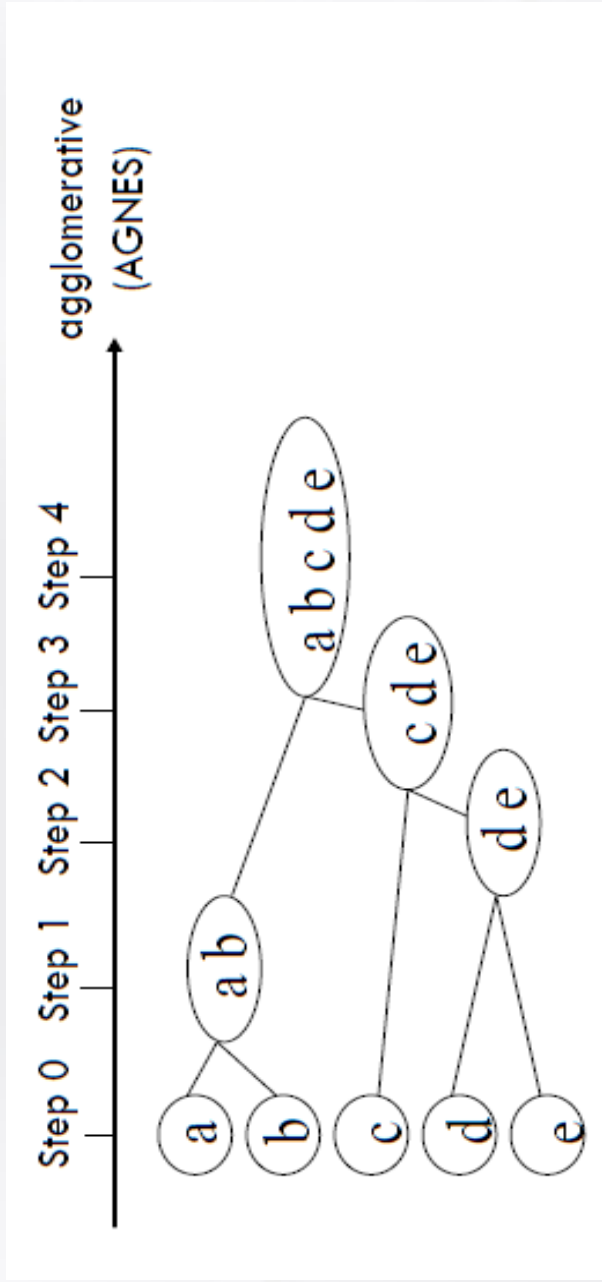
- Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level

Hierarchical clustering may correspond to meaningful taxonomies

- Example in biological sciences (e.g., phylogeny reconstruction, etc), web (e.g., product catalogs)

Agglomerative clustering

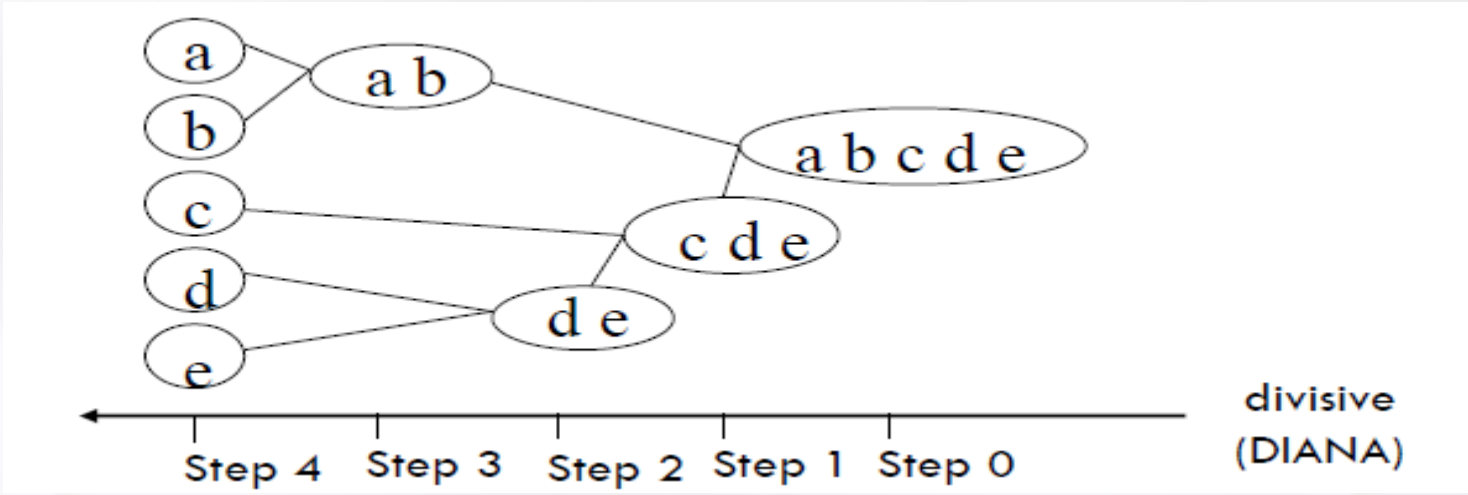
This bottom-up strategy starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied. Most hierarchical clustering methods belong to this category.



Agglomerative clustering

Divisive clustering

This top-down strategy does the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster. It subdivides the clusters into smaller and smaller pieces, until each object form a cluster on its own or until it satisfies certain termination conditions, such as a desired number of cluster or the diameter of each cluster is within a certain threshold.



Agglomerative clustering

Basic algorithm

1. Compute the distance matrix between the input data points
2. Let each data point be a cluster

Repeat

4. Merge the two closest clusters
5. Update the distance matrix
6. **Until only a single cluster remains /certain termination conditions**

Key operation is the computation of the distance between two clusters. Different definitions of the distance between clusters lead to different algorithms.

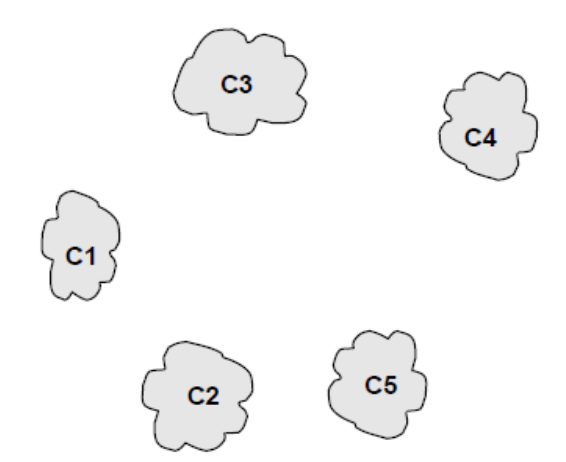
Agglomerative clustering

Start with clusters of individual points and a distance matrix

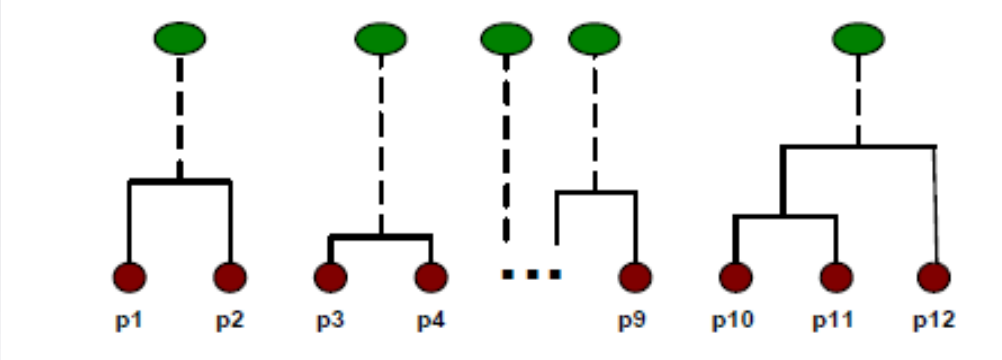
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Agglomerative clustering

After some merging steps, we have some clusters

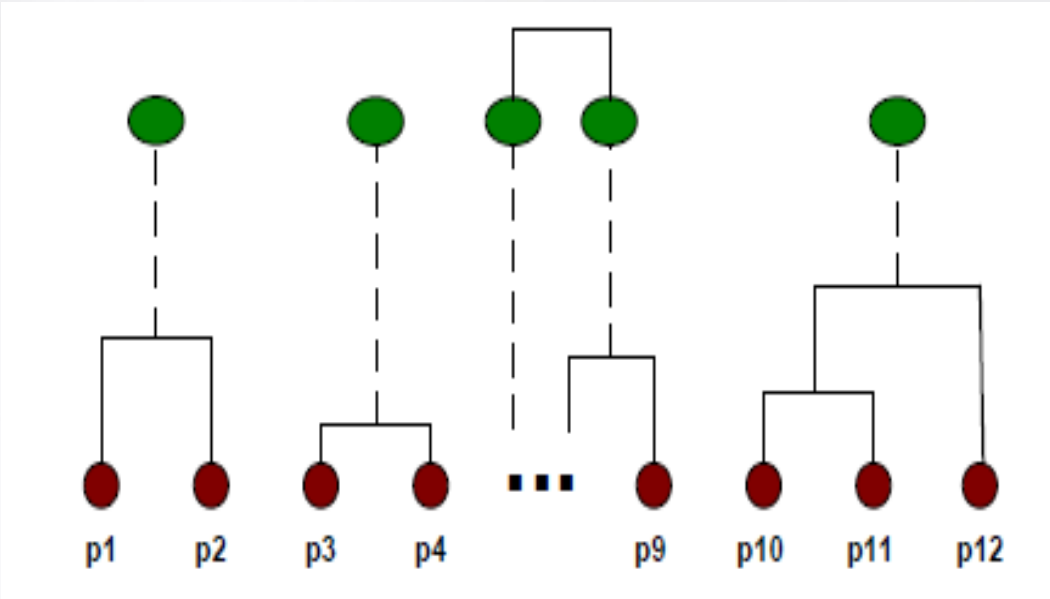


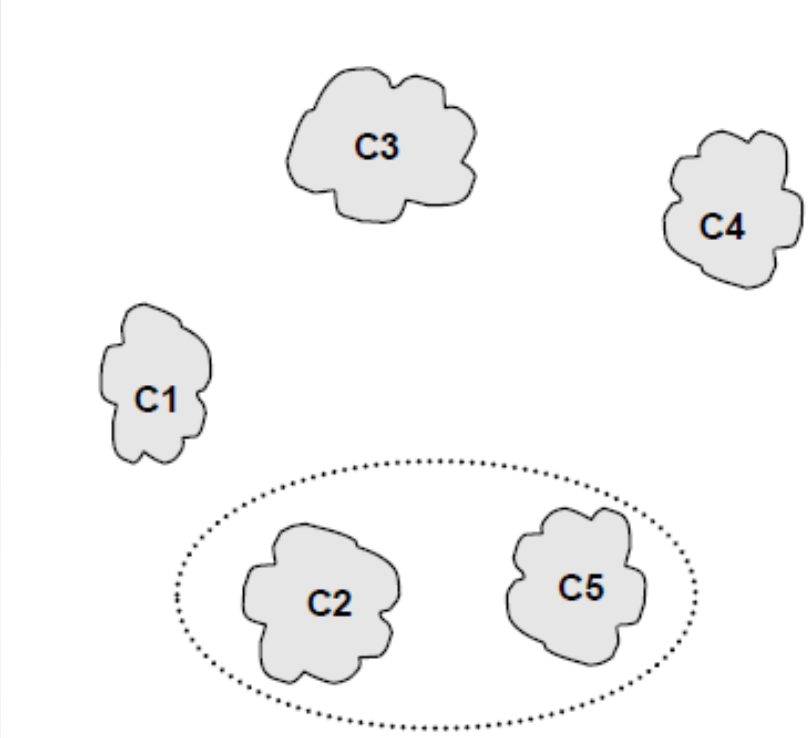
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					



Agglomerative clustering

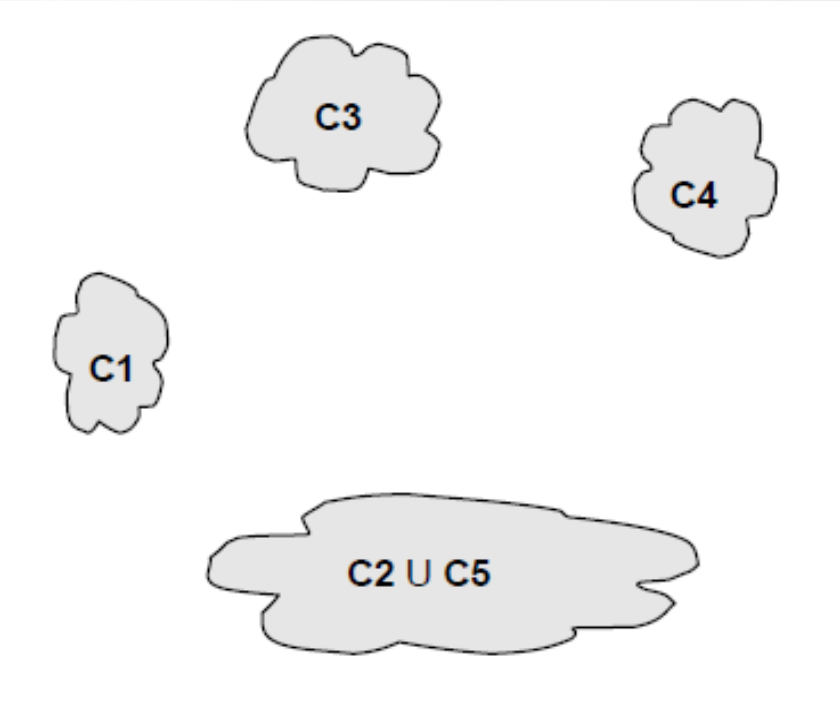
Merge the two closest clusters (C2 and C5) and update the distance matrix.





Agglomerative clustering

Problem: How do we update the distance matrix?



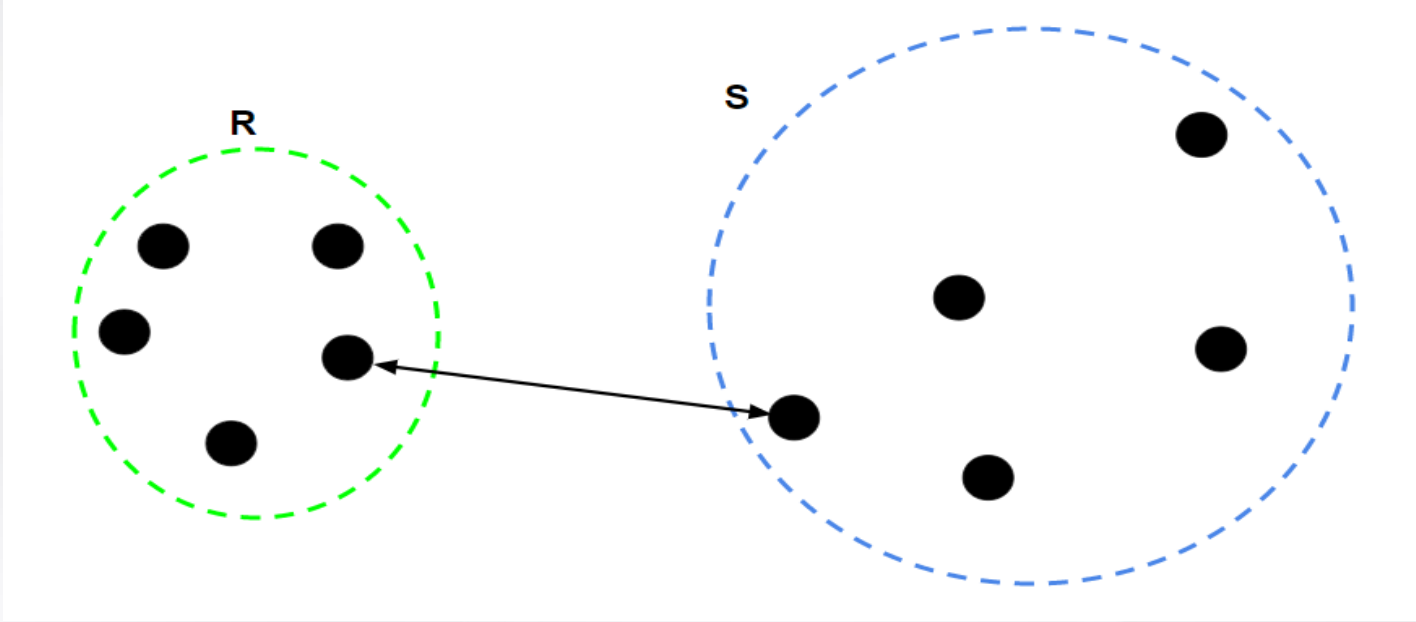
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Linkage---Distance between two clusters

① **Single Linkage**

For two clusters R and S, the single linkage returns the minimum distance between two points **i** and **j** such that **i** belongs to R and **j** belongs to S.

$$L(R,S) = min(D(i,j)), i \epsilon R, j \epsilon S$$

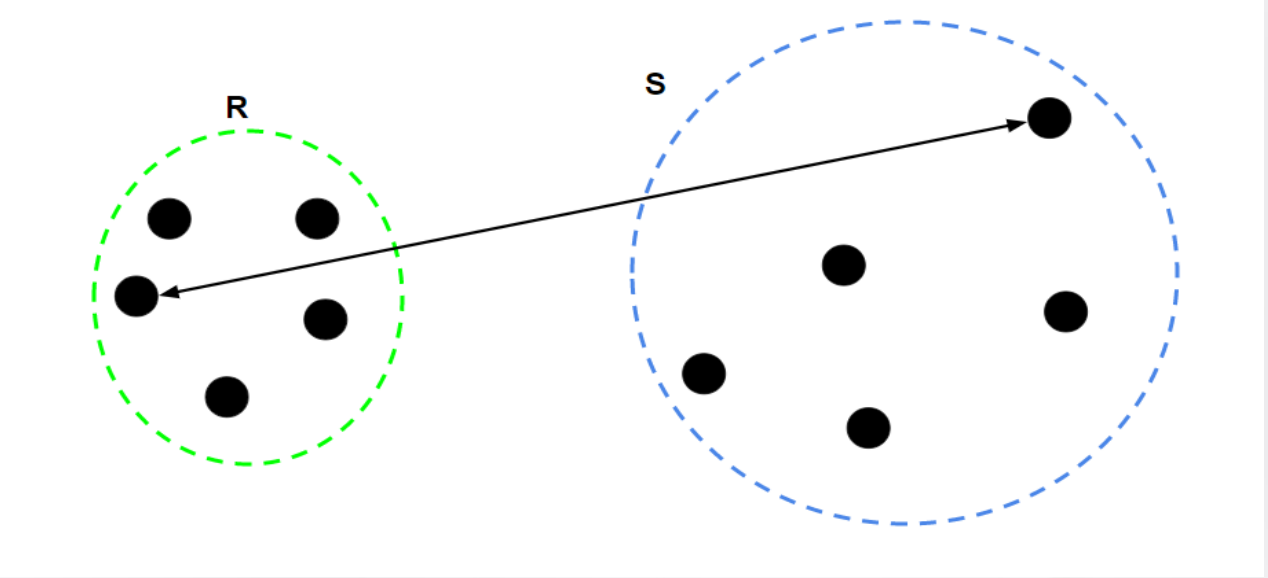


Linkage---Distance between two clusters

② **Complete Linkage:**

For two clusters R and S, the complete linkage returns the maximum distance between two points **i** and **j** such that **i** belongs to R and **j** belongs to S.

$$L(R,S) = max(D(i,j)), i \epsilon R, j \epsilon S$$



Linkage---Distance between two clusters

- ✓ Single linkage suffers from chaining(松散). In order to merge two groups, only need one pair of points to be close, irrespective of all others. Therefore clusters can be too spread out, and not compact enough.
- ✓ Complete linkage avoids chaining, but suffers from crowding. Because its score is based on the worst-case dissimilarity between pairs, a point can be closer to points in other clusters than to points in its own cluster. Clusters are compact, but not far enough apart

Linkage---Distance between two clusters

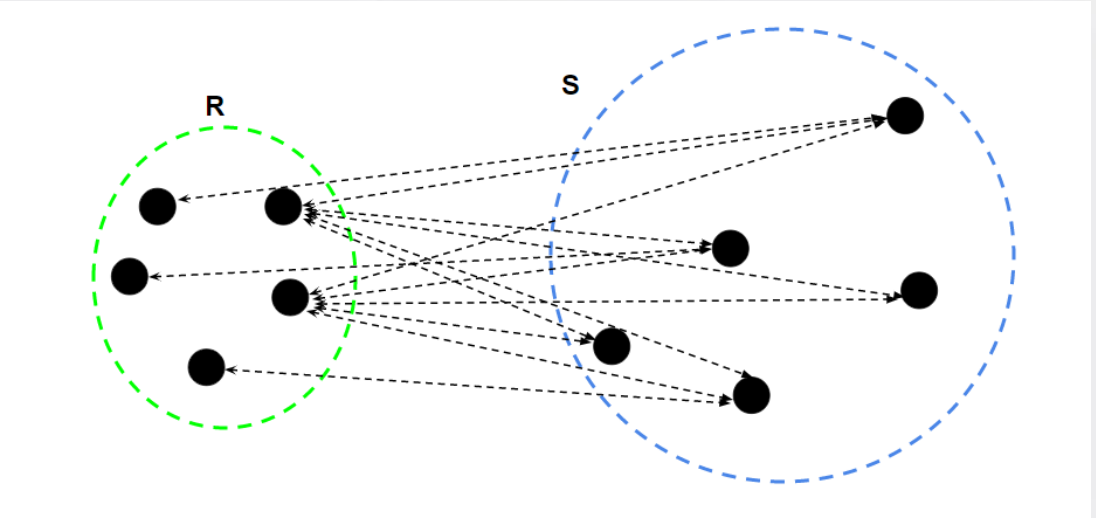
③ Group average Linkage

For two clusters R and S, first for the distance between any data-point i in R and any data-point j in S and then the arithmetic mean of these distances are calculated. Average Linkage returns this value of the arithmetic mean.

$$L(R,S) = \frac{1}{n_R \times n_S} \sum_{i=1}^{n_R} \sum_{j=1}^{n_S} D(i,j), i \in R, j \in S$$

where,

- n_R : Number of data-points in R
- n_S : Number of data-points in S



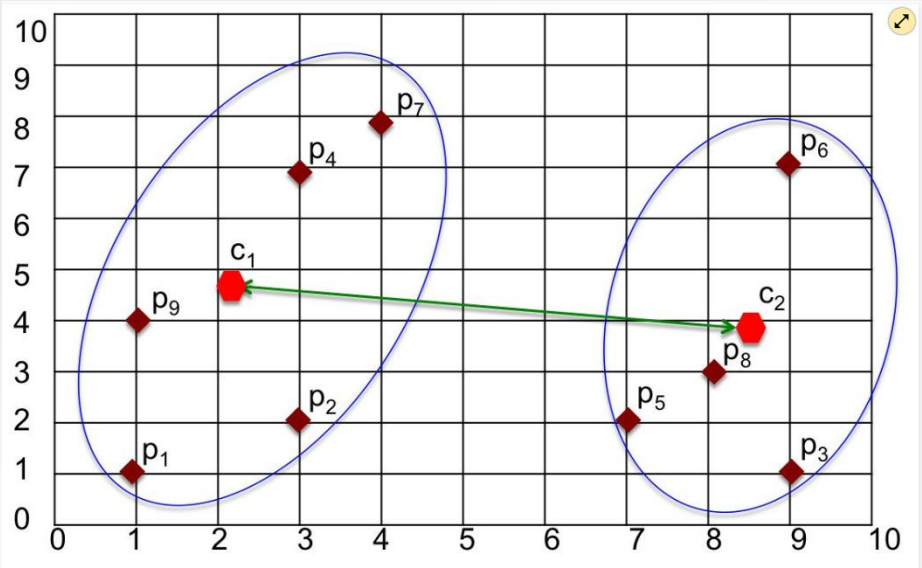
Linkage---Distance between two clusters

④ **Group Mean distance (The Centroid Criterion)**

between clusters C_i and C_j is the distance between the means or centroids of the two clusters.

$$D(C_i,C_j)=d(\mu_i,\mu_j)$$

where μi and μj are the centroids of cluster C_i and C_j respectively.



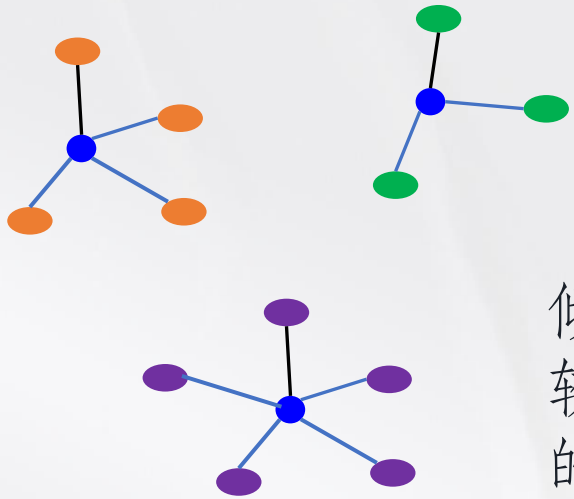
Linkage---Distance between two clusters

⑤ **Ward approach**（新生成的簇的方差增加量是最小的）
analyzes the variance of the clusters rather than measuring distances directly, minimizing the total within-cluster variance.

At each stage, two clusters merge that provide the smallest increase in the combined error sum of squares.

在每一步合并簇时，选择使合并后簇内的“总方差”增加最小的两个簇进行合并。

ESS（合并前）＝ESS（红）+ESS（绿）
+ESS（其它）
ESS（合并后）＝ESS（红绿）+ESS（其它）
ESS（合并后）＝ESS（红紫）+ESS（其它）
ESS（合并后）＝ESS（绿紫）+ESS（其它）

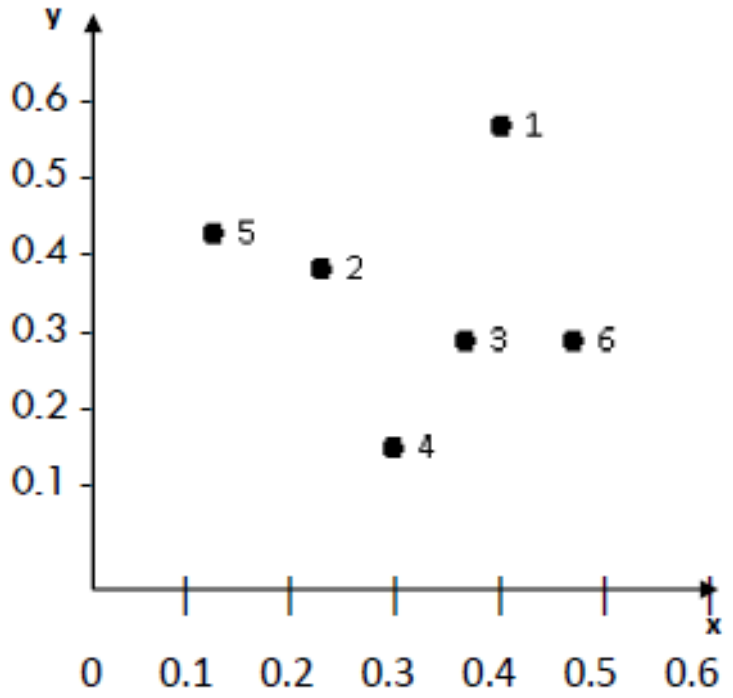


倾向于创建比较均匀（相似）的簇

Example

Assume that the database D is given by the table below. Follow single link technique to find clusters in D. Use Euclidean distance measure.

D	x	y
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

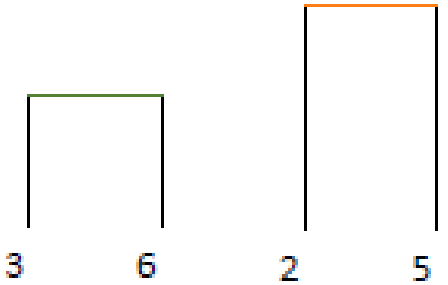


Example

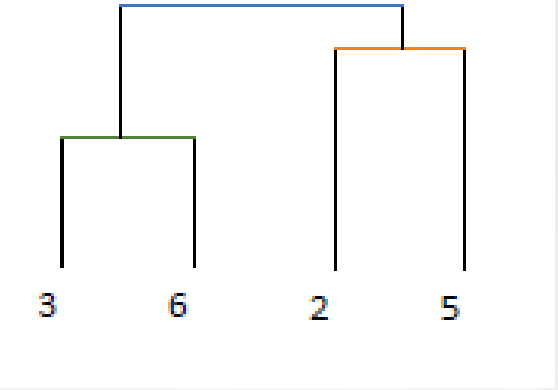
Distance matrix

p1	0					
p2	0.24	0				
p3	0.22	0.15	0			
p4	0.37	0.20	0.15	0		
p5	0.34	0.14	0.28	0.29	0	
p6	0.23	0.25	0.11	0.22	0.39	0
	p1	p2	p3	p4	p5	p6

p1	0				
p2	0.24	0			
(p3, p6)	0.22	0.15	0		
p4	0.37	0.20	0.15	0	
p5	0.34	0.14	0.28	0.29	0
	p1	p2	(p3, p6)	p4	p5

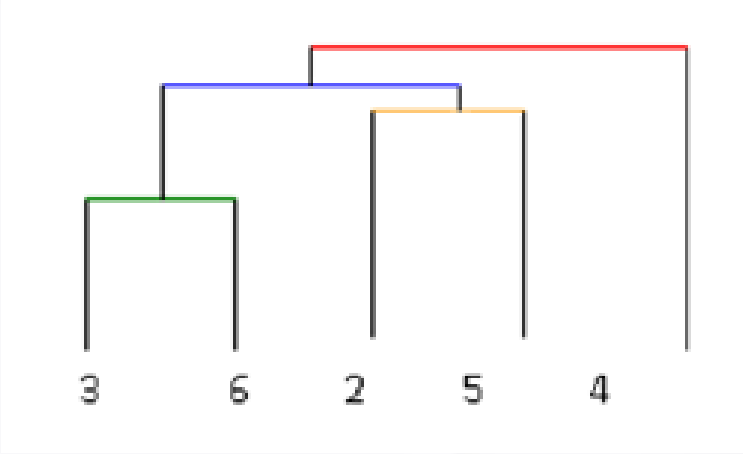


p1	0			
(p2, p5)	0.24	0		
(p3, p6)	0.22	0.15	0	
p4	0.37	0.20	0.15	0
	p1	(p2, p5)	(p3, p6)	p4



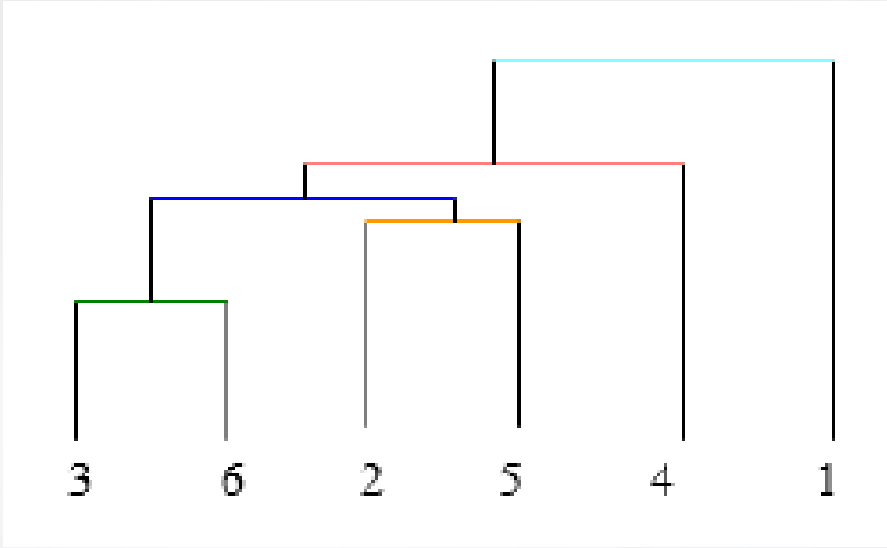
Example

p1	0		
(p2, p5, p3, p6)	0.22	0	
p4	0.37	0.15	0
	p1	(p2, p5, p3, p6)	p4



Distance matrix

p1	0	
(p2, p5, p3, p6, p4)	0.22	0
	p1	(p2, p5, p3, p6,p4)



04 Example

feature	Hierarchical Clustering	K-Means Clustering
Type of clustering	Agglomerative (bottom-up) or divisive (top-down)	Partitional (centroid-based)
Cluster shape	Can handle non-convex shapes and variable cluster sizes	Assumes spherical and equally sized clusters
Distance metric	Can use various distance measures, such as Euclidean, Manhattan, or cosine	Must use Euclidean distance
Scalability	Can be computationally expensive for large datasets or many clusters	Can handle large datasets and many clusters efficiently
Interpretability	Provides a hierarchical structure and dendrogram that can help in interpreting the clustering results	Provides cluster centers and assignments, but no hierarchical structure
outliers	Sensitive	Sensitive

Combining hierarchical clustering and k-means

k-means clustering: a partitioning method used for splitting a dataset into a set of k clusters.

hierarchical clustering: an alternative approach to k-means clustering for identifying clustering in the dataset by using pairwise distance matrix between observations as clustering criteria.

However, each of these two standard clustering methods has its limitations.

K-means clustering requires the user to specify the number of clusters in advance and selects initial centroids randomly.

Agglomerative hierarchical clustering is good at identifying small clusters but not large ones.

Combining hierarchical clustering and k-means

Recall that, in k-means algorithm, a random set of observations are chosen as the initial centers.

The final k-means clustering solution is very sensitive to this initial random selection of cluster centers. The result might be (slightly) different each time you compute k-means.

To avoid this, a solution is to use a **hybrid approach** by combining the **hierarchical clustering** and the **k-means** methods.

This process is named **hybrid hierarchical k-means clustering** (hkmeans).

Combining hierarchical clustering and k-means

The procedure is as follow:

① Compute **hierarchical clustering** and cut the tree into k-clusters

② Compute the center (i.e the mean) of each cluster

③ Compute k-means by using the set of cluster centers (defined in step 3) as the initial cluster centers

k-means algorithm will improve the initial partitioning generated at the step 2 of the algorithm.