Data Mining & Machine Learning

Yong Zheng

Illinois Institute of Technology
Chicago, IL, 60616, USA

ILLINOIS TECH
College of Computing

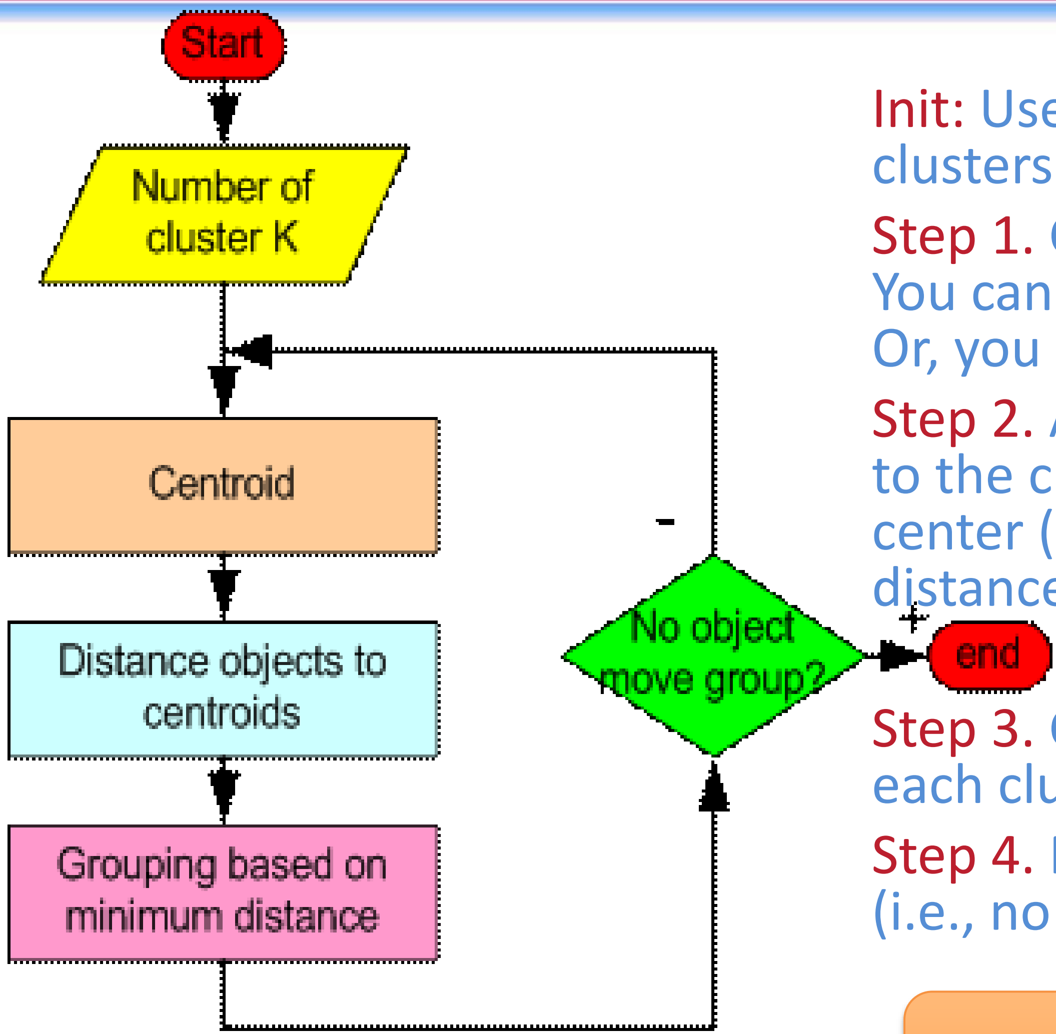
Supervised vs Unsupervised Learning

- **Supervised Learning**
  - Task: Classification (Binary, Multi-Class, Multi-Label)
  - Algorithms: KNN, Naïve Bayes, Tree, SVM, Ensemble
  - Applications: IR, Text Classification
- **Unsupervised Learning**
  - Task: Clustering, Associated Rules, Outlier Detections
  - Algorithms: K-Means, K-Mediods, Hierarchical clustering, Fuzzy clustering, Association Rules, etc

Clustering Tasks

- Partitional Clustering: just group objects to minimize intra-cluster distances and maximize inter-cluster distances
  Example: Document Clustering
- Density-Based Clustering: cluster objects based on the local connectivity and density functions
- Hierarchical Clustering: a clustering process in order to discover the hierarchical structure, like a hierarchical tree
  Example: categories and subcategories; taxonomies

K-Means Clustering Algorithm



Init: Users specifies k – number of clusters used to group the data

Step 1. Create Initial Clusters
You can randomly create k clusters
Or, you can randomly select k objects

Step 2. Assign the remaining data points to the cluster with the nearest cluster center (based on some similarity or distance function)

Step 3. Compute the average point for each cluster -> new cluster center

Step 4. Repeat 2,3 until convergence (i.e., no points move between clusters)

Normalize the features!!

Clustering

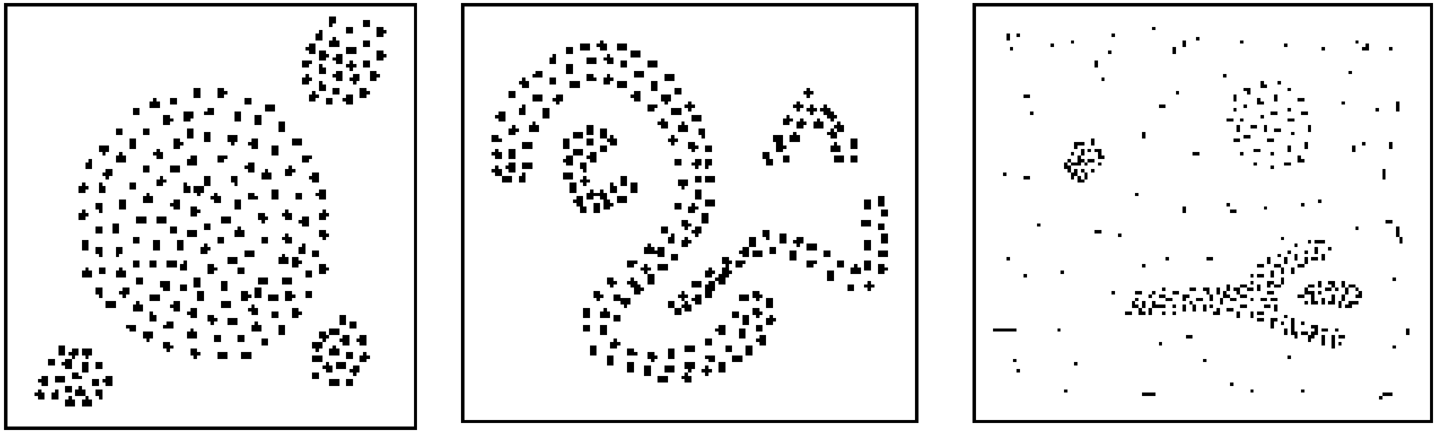
- Intro: Clustering
- Partitional Clustering
- Density-Based Clustering
- Hierarchical Clustering

Clustering

- Intro: Clustering
- Partitional Clustering
- Density-Based Clustering
- Hierarchical Clustering

Density-Based Clustering

- Density-Based Clustering: cluster objects based on the local connectivity and density functions. Each cluster has a considerable higher density of points than outside of the cluster

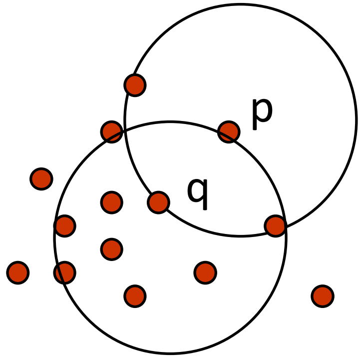


Density-Based Clustering

- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - GDBSCAN: Sander, et al. (KDD'98)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

Density-Based Clustering: Concepts

- Two global parameters***:***
  - ***Eps***: Maximum radius or distance of the neighborhood
  - ***MinPts***: Minimum number of points in the neighborhood of that point
- **Core Object:** its neighborhood has at least MinPts objects
- **Border Object:** object that on the border of a cluster

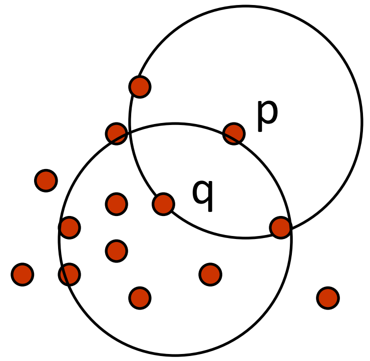


MinPts = 5

Eps = 1 cm

Density-Based Clustering: Concepts

- ***Eps-Neighborhood***
  $N_{Eps}(p)$: $\{q \text{ belongs to } D \mid dist(p,q) \le Eps\}$
- **Directly density-reachable:** A point ***q*** is directly density-reachable from a point ***p*** wrt. ***Eps***, ***MinPts*** if
  - 1) *q* belongs to $N_{Eps}(p)$
  - 2) p is core object

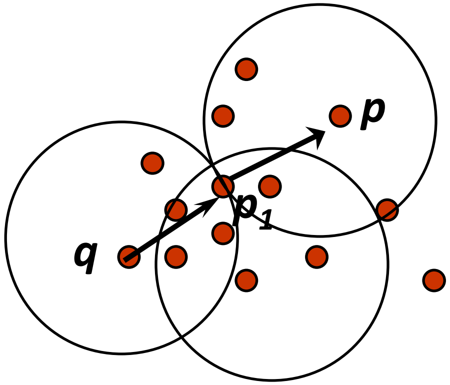


MinPts = 5

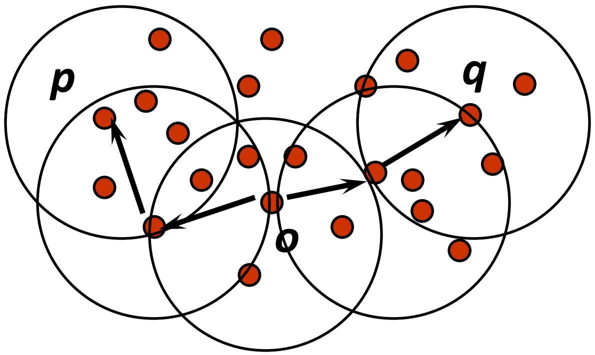
Eps = 1 cm

Density-Based Clustering: Concepts

- **Density-reachable:**
  - A point p is density-reachable from a point q wrt. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i

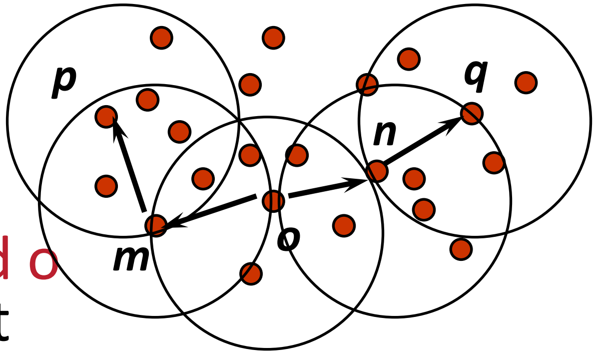


- **Density-connected**
  - A point p is density-connected to a point q wrt. *Eps*, *MinPts* if there is a point o such that both, p and q are density-reachable from o wrt. *Eps* and *MinPts*.



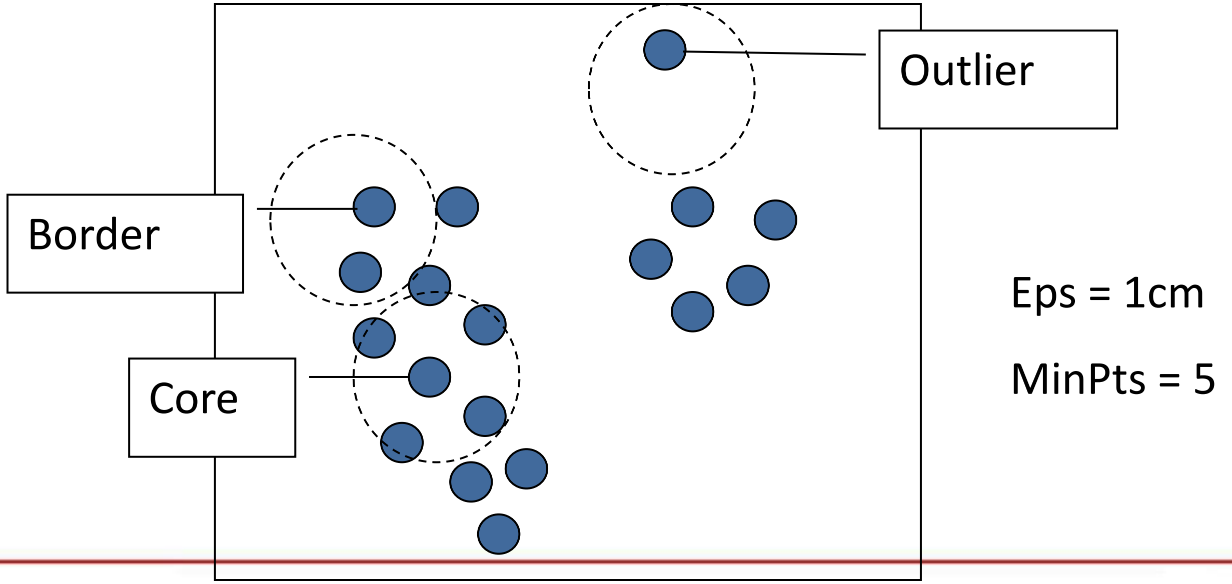
Density-Based Clustering: Concepts

- **Example (Eps, MinPts as parameters)**
  - There are 5 points: o, m, n, p, q
  - Assume o is the core object
  - Directly density-reachable: m and o, n and o
    m and n are in $N_{Eps}(o)$, and o is core object
  - Density-reachable: p and o, q and o
    o->m->p, (o,m) and (m,p) are directly density-reachable
    o->n->q, (o,n) and (n,q) are directly density-reachable
  - Density-connected: p and q
    There is a route, (p, o), (o, q) are density-reachable



DBSCAN

- DBSCAN is a popular density-based clustering method
- It relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- It can discover clusters of arbitrary shape



DBSCAN

– Randomly select a point ***p***

– Retrieve all points density-reachable from ***p*** wrt ***Eps*** and ***MinPts***.

– If ***p*** is a core point, a cluster is formed.

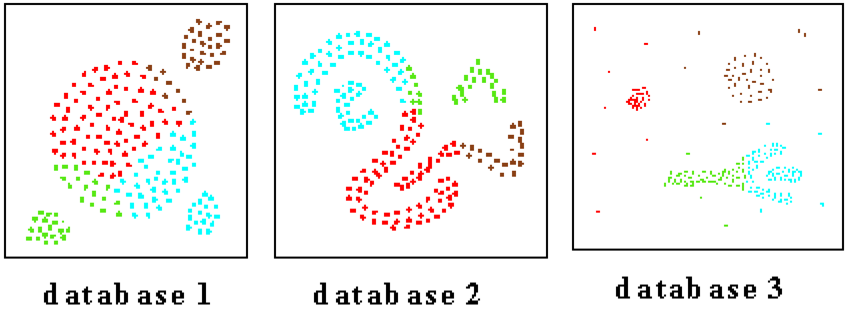
– If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.

– Continue the process until all of the points have been processed.

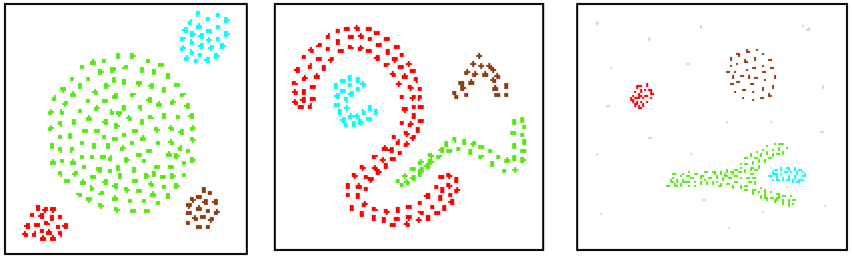
DBSCAN

- CLARANS is an efficient medoid-based clustering algorithm

CLARANS:

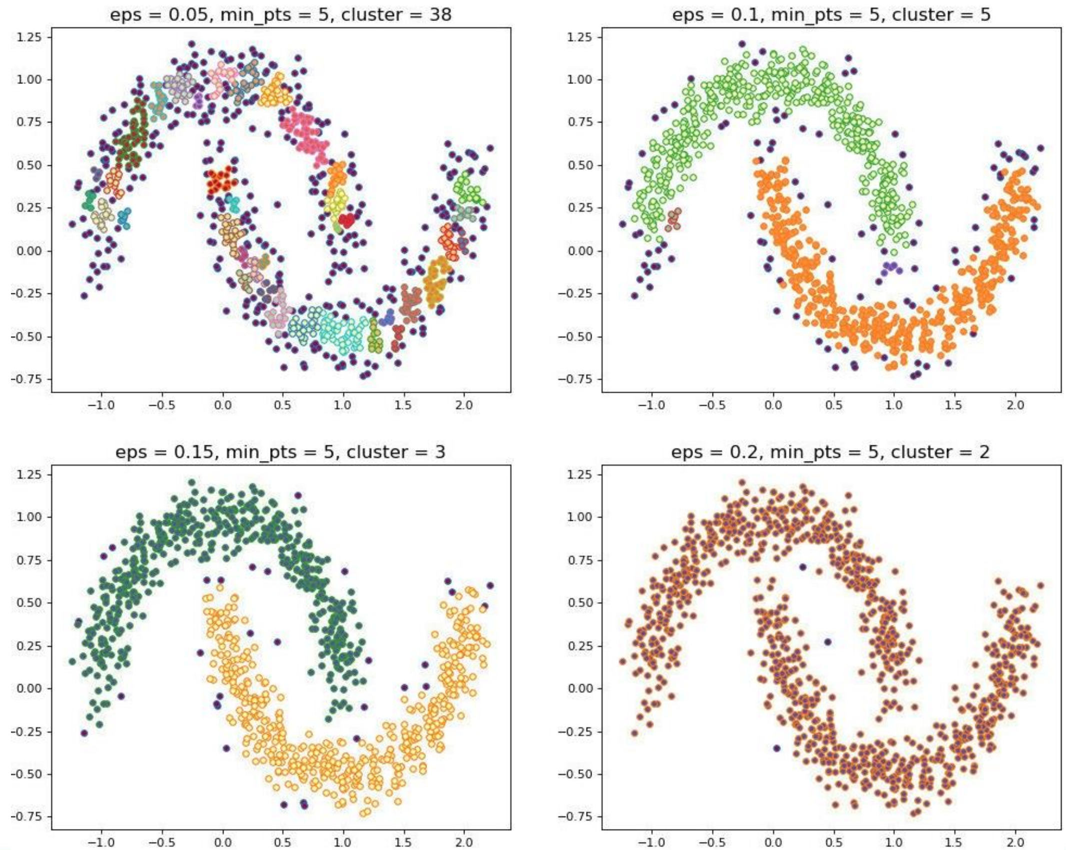


DBSCAN:



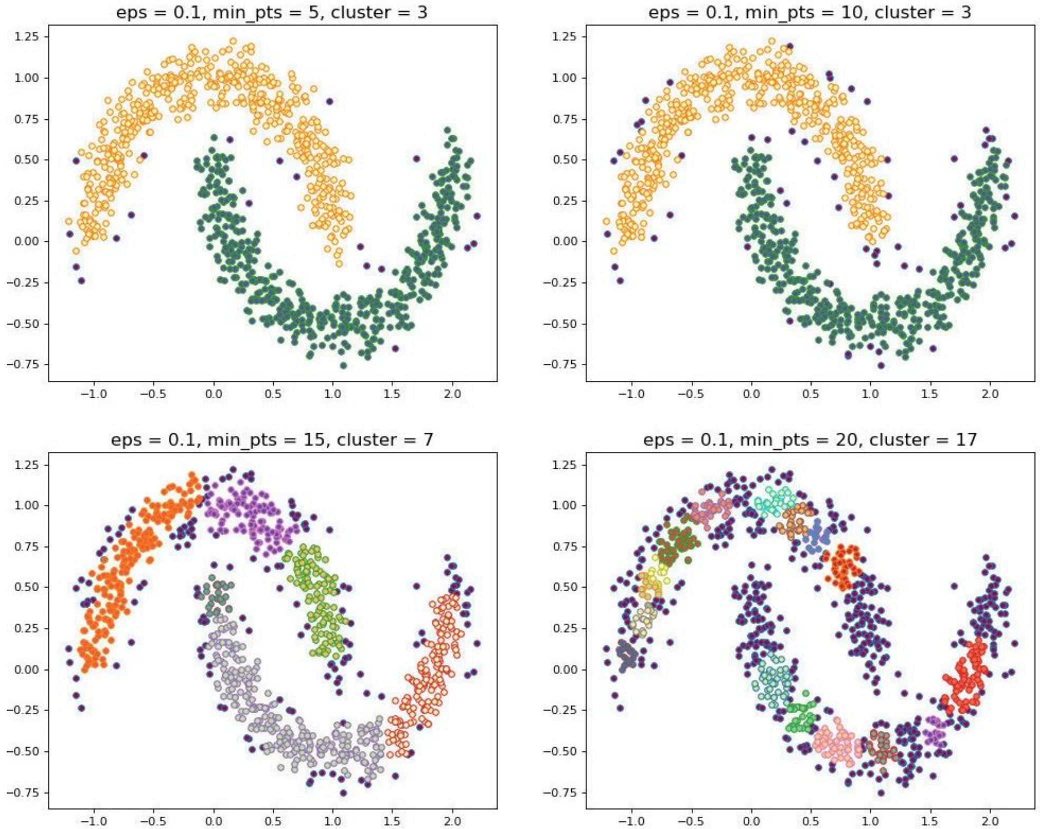
DBSCAN

- The two parameters, Eps and MinPts, need to be carefully tuned up. Otherwise, results may be significantly different
- Example: By using different Eps



DBSCAN

- The two parameters, Eps and MinPts, need to be carefully tuned up. Otherwise, results may be significantly different
- Example: By using different MinPts



K-Means vs DBSCAN

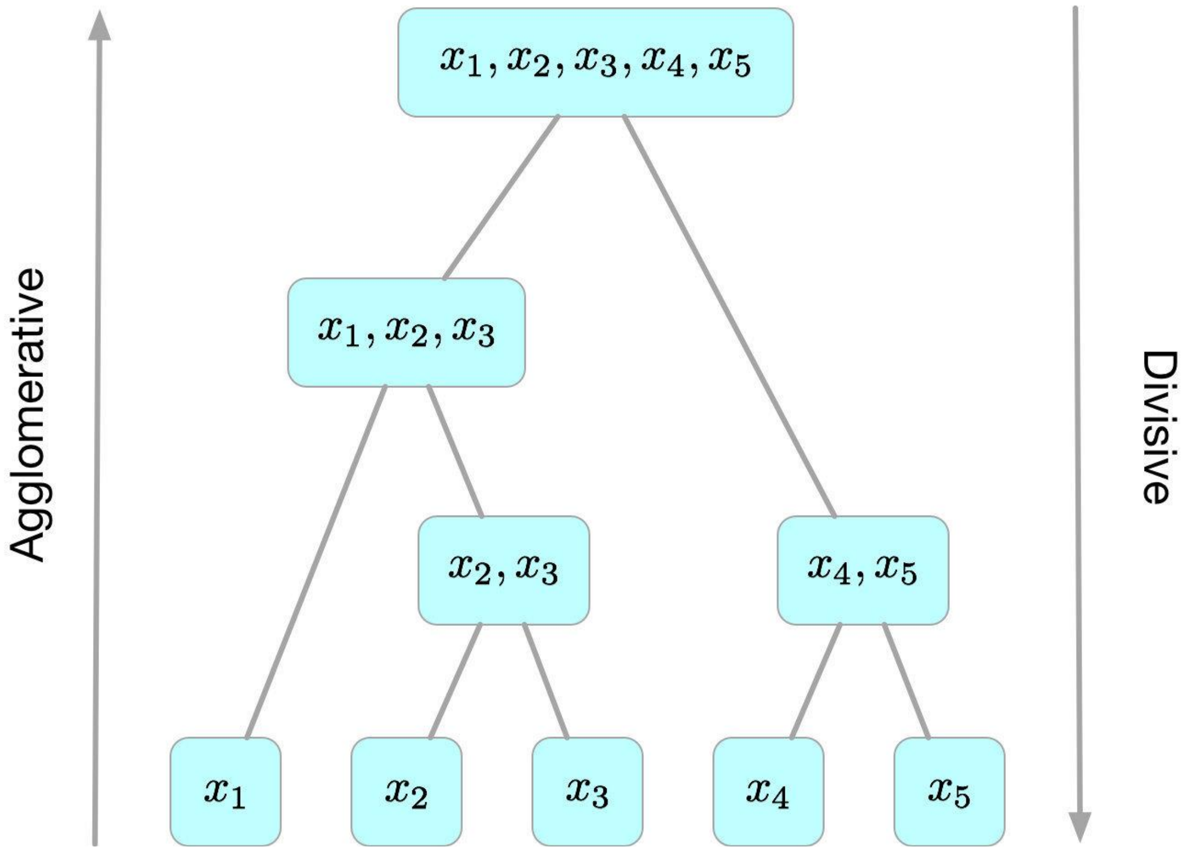
- K-Means
  - Partitional Clustering
  - Need to pre-define the value of K
  - Sensitive to initial settings
  - Sensitive to noise data
- DBSCAN
  - Density-Based Clustering
  - Do not need to pre-define the number of clusters
  - Need to pre-define Eps and MinPTs
  - Sensitive to initial settings
  - Not sensitive to noise data

Clustering

- Intro: Clustering
- Partitional Clustering
- Density-Based Clustering
- Hierarchical Clustering

Hierarchical Algorithms

- Use distance matrix as clustering criteria
  - does not require the no. of clusters as input, but needs a termination condition

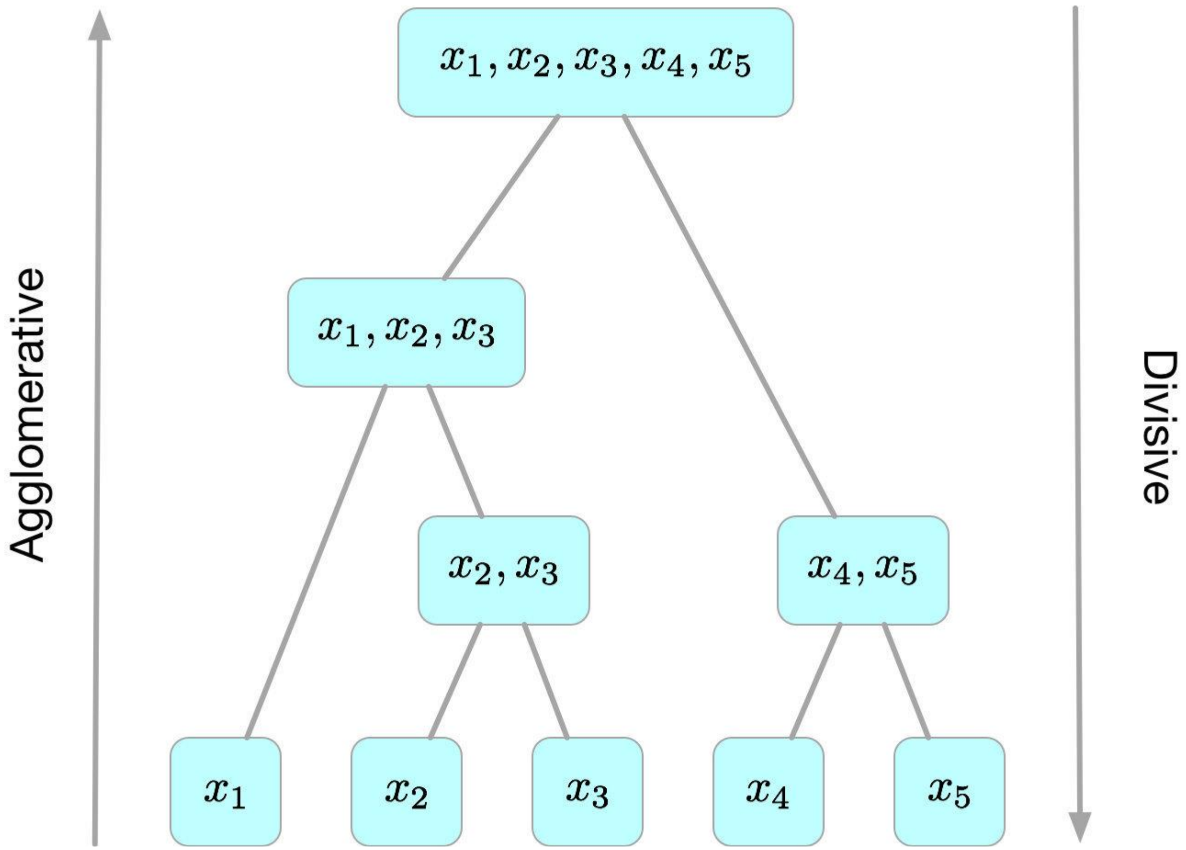


Hierarchical Algorithms

- Agglomerative Method
  - Each individual object is considered as a single cluster at the beginning
  - Choose a way to represent the cluster, such as means-centroid
  - Iterate all clusters, find the two clusters with smallest distance, and merge them to a new cluster
  - Repeat the step above until all objects are grouped to a single cluser

Hierarchical Algorithms

- Agglomerative Method



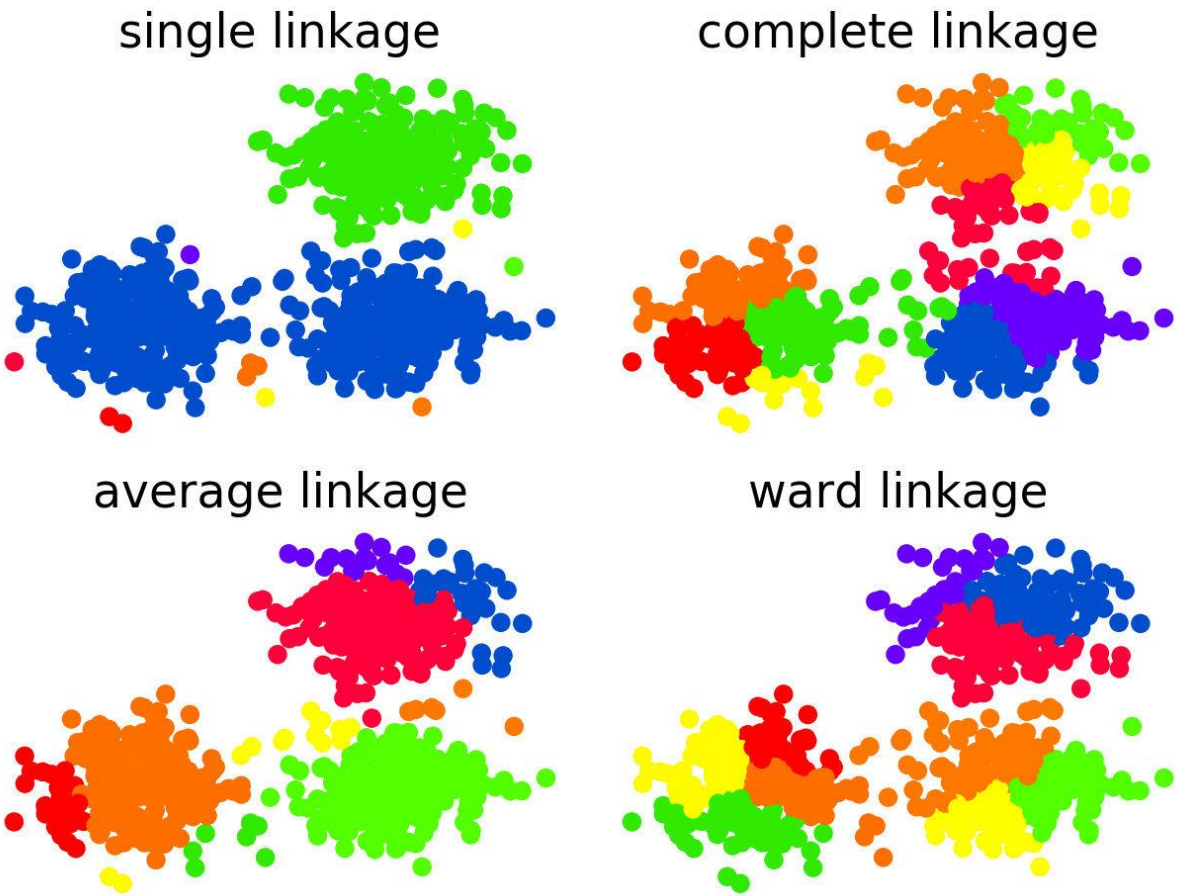
Hierarchical Algorithms

- In K-Means, we need to use similarity or distance metrics to measure the distance between two objects
- In hierarchical clustering, we need to measure the distance between two clusters
- It is more complicated, since there are multiple objects within a cluster

Distance Between Clusters

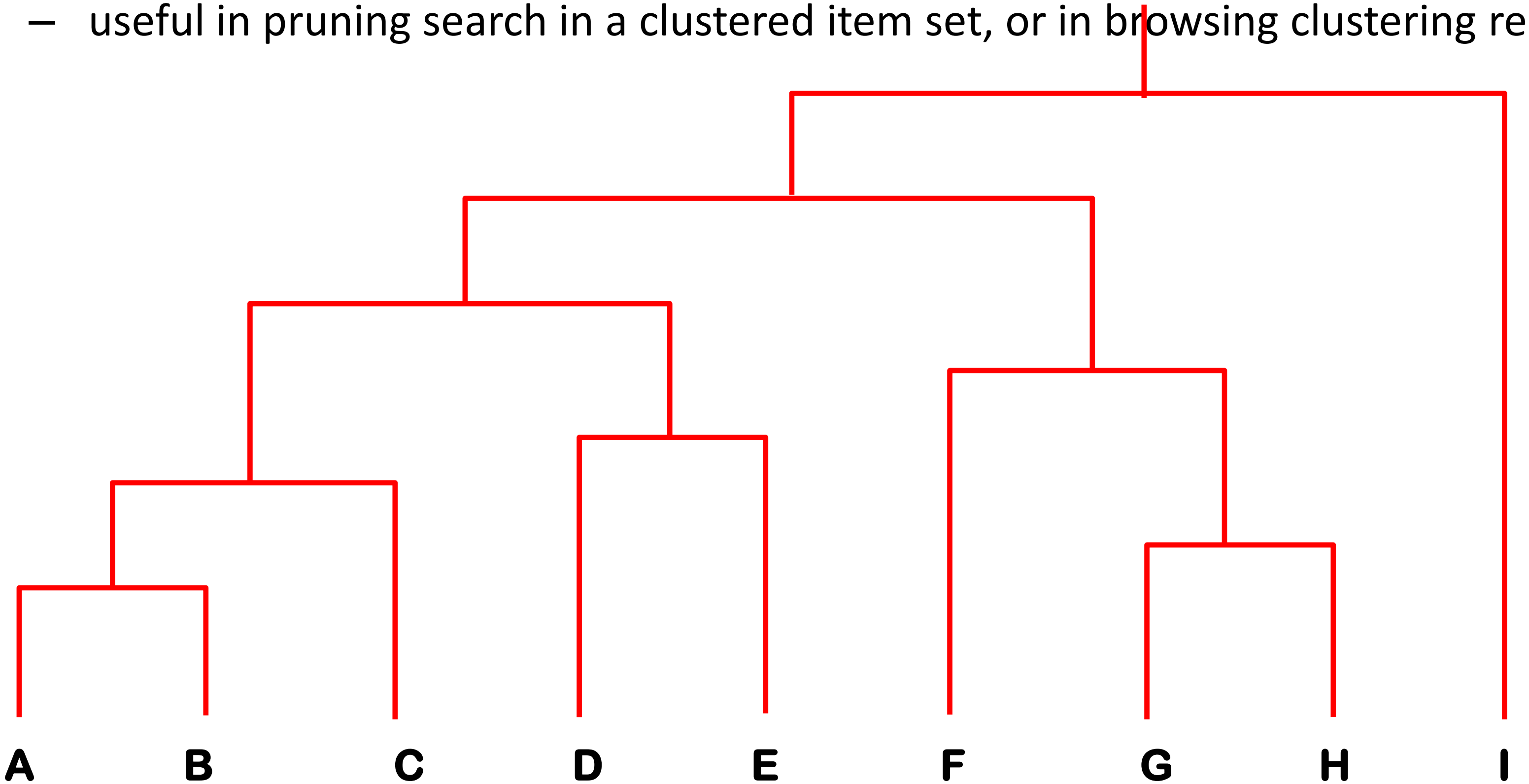
- **Single-Linkage**
  Distance = distance between two closest objects from two cluster
- **Complete-Linkage**
  Distance = distance between two farthest objects from two clusters
- **Ward's Linkage**
  Distance = how much the sum of squares (i.e., within cluster distance) will increase when we merge them
- **UPGMA**
  Distance = average distance of the distance of every two objects in the two clusters
- **Centroid Method**
  Distance = distance between the centroids of the two clusters

Distance Between Clusters



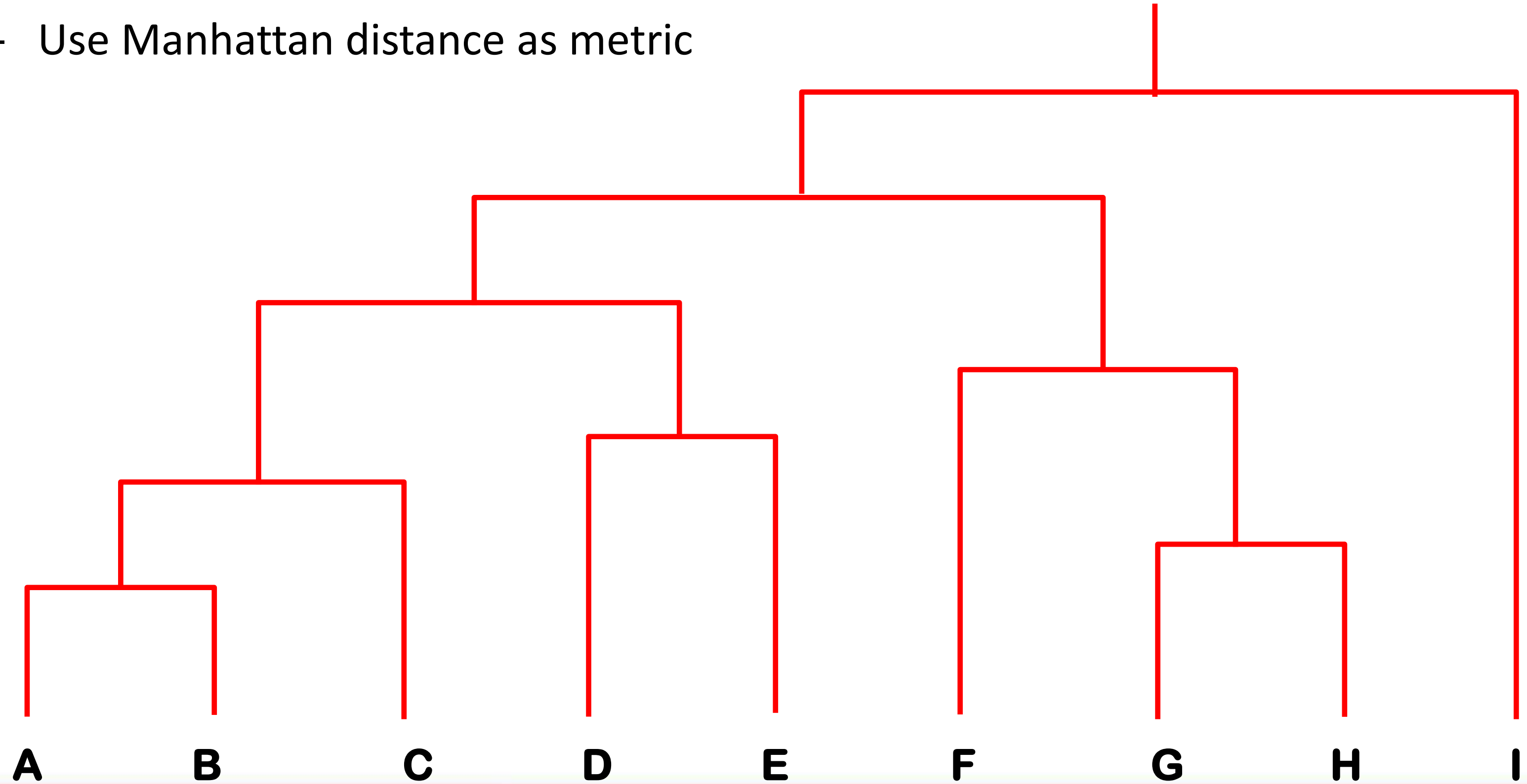
Hierarchical Agglomerative Clustering

- **HAC starts with unclustered data and performs successive pairwise joins among items (or previous clusters) to form larger ones**
  - this results in a hierarchy of clusters which can be viewed as a dendrogram
  - useful in pruning search in a clustered item set, or in browsing clustering results



Hierarchical Clustering

- **Given a list of numbers: 9, 13, 7, 3, 4**
  - Build hierarchical clustering tree structure from bottom to the up
  - Use the mean as the representative of each cluster, Use centroid method to merge clusters
  - Use Manhattan distance as metric



Hierarchical Clustering

- **For example: a list of numbers {9, 13, 7, 3, 4}**
  - At the beginning, each number is an individual cluster

Hierarchical Clustering

- **For example: a list of numbers {9, 13, 7, 3, 4}**
  - At the beginning, each number is an individual cluster
  - We calculate the distance between every two centroids
  - And merge the two clusters with smallest distance

	[3]	[4]	[7]	[9]	[13]
[3]	0				
[4]	1	0			
[7]	4	3	0		
[9]	6	5	2	0	
[13]	10	9	6	4	0

Hierarchical Clustering

- **For example: a list of numbers {9, 13, 7, 3, 4}**
  - At the beginning, each number is an individual cluster
  - We calculate the distance between every two centroids
  - And merge the two clusters with smallest distance
  - Right now, we only have 5 clusters, re-calculate centroids
  - Next: calculate the distance between remaining clusters

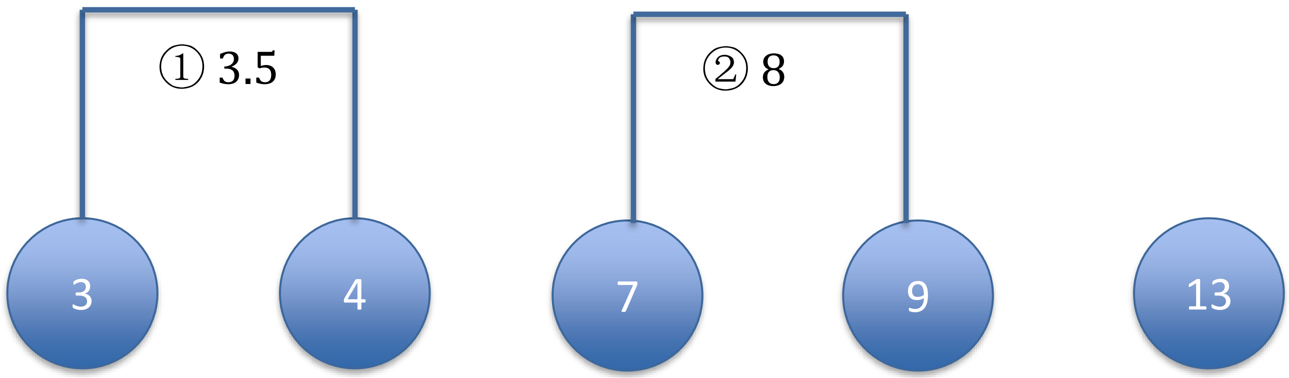
Hierarchical Clustering

- **For example: a list of numbers {9, 13, 7, 3, 4}**
  - calculate the distance between remaining clusters
  - Merge the two clusters with smallest distance

	[3, 4] = 3.5	[7]	[9]	[13]
[3, 4] = 3.5	0			
[7]	3.5	0		
[9]	5.5	2	0	
[13]	9.5	6	4	0

Hierarchical Clustering

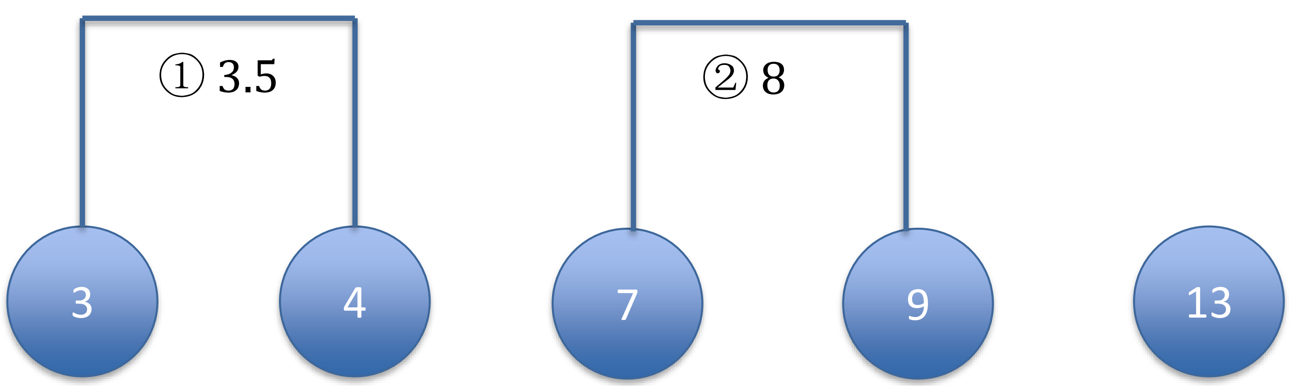
- **For example: a list of numbers {9, 13, 7, 3, 4}**
    - Right now, we only have 3 clusters, re-calculate centroids
    - Next: calculate the distance between remaining clusters



Hierarchical Clustering

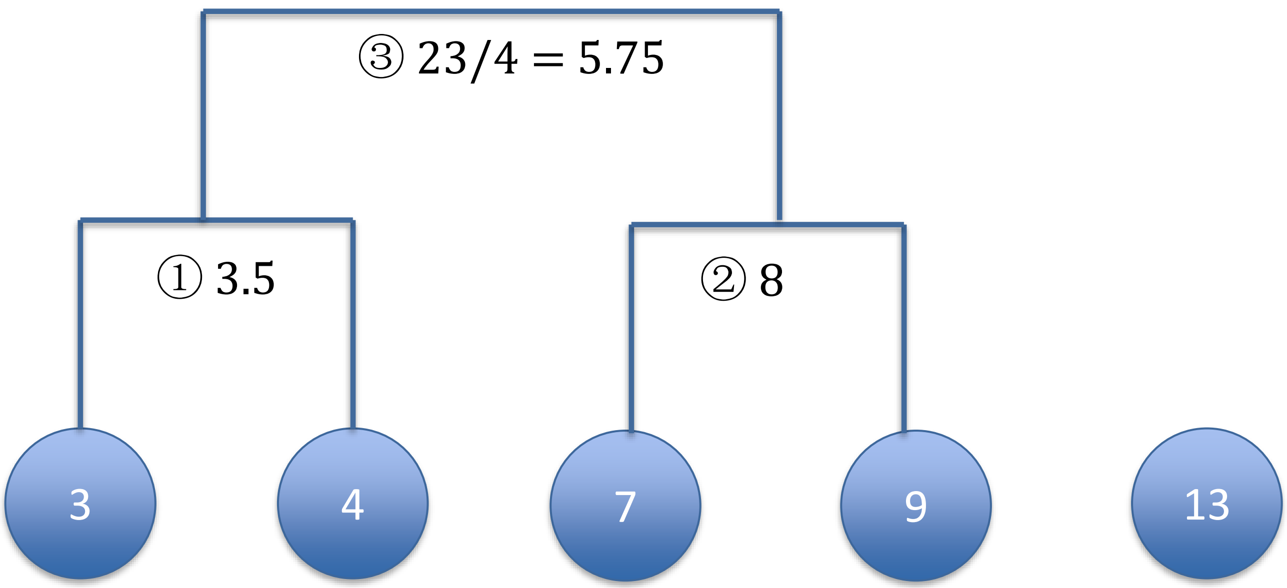
- **For example: a list of numbers {9, 13, 7, 3, 4}**
  - Next: calculate the distance between remaining clusters
  - Merge the two clusters with smallest distance

	[3, 4] = 3.5	[7, 9] = 8	[13]
[3, 4] = 3.5	0		
[7, 9] = 8	4.5	0	
[13]	9.5	5	0

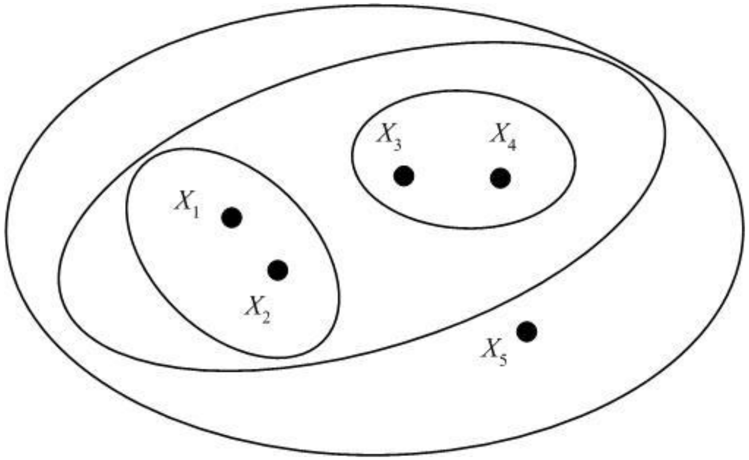


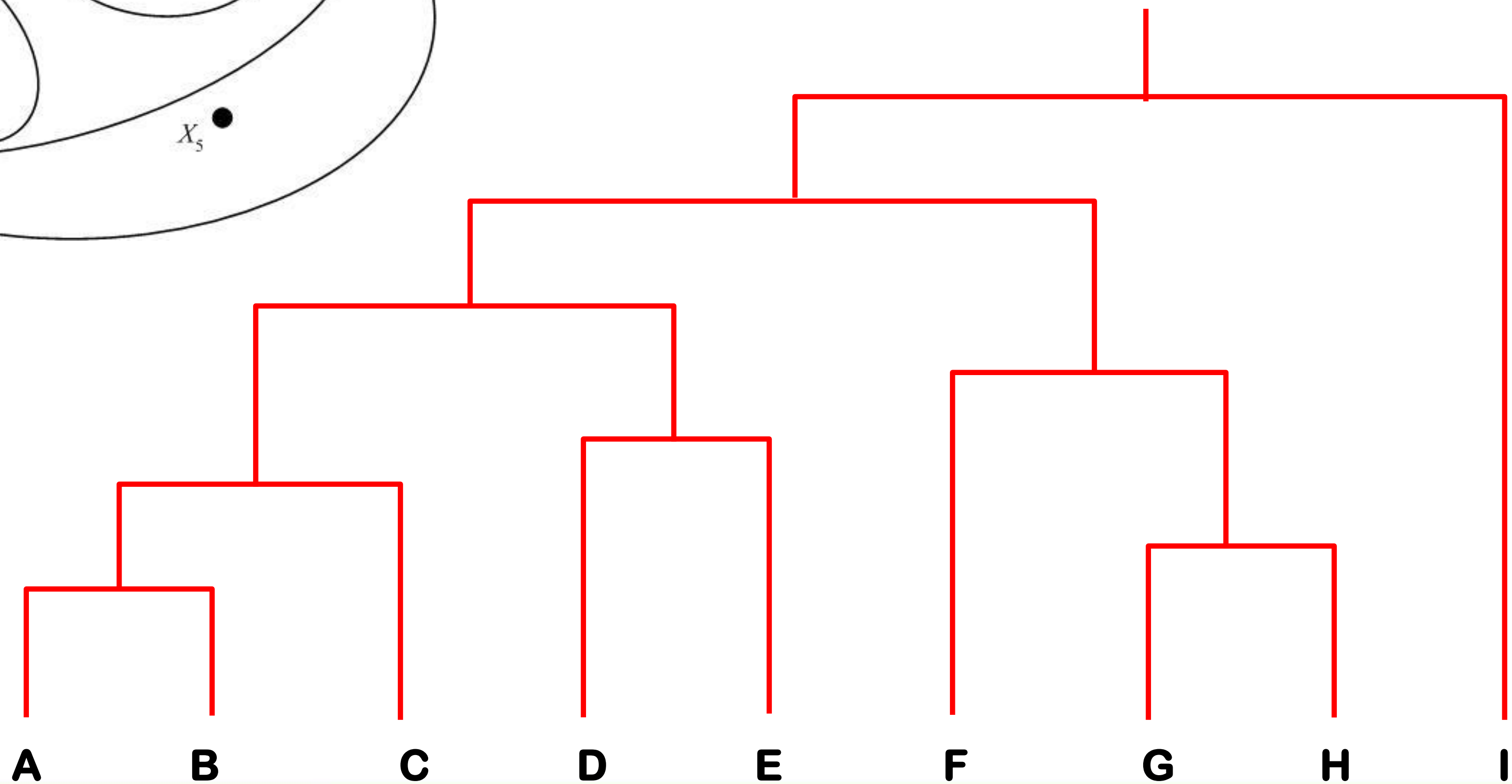
Hierarchical Clustering

- **For example: a list of numbers {9, 13, 7, 3, 4}**
  - Right now, we only have 2 clusters, re-calculate centroids
  - Next: calculate the distance between remaining clusters



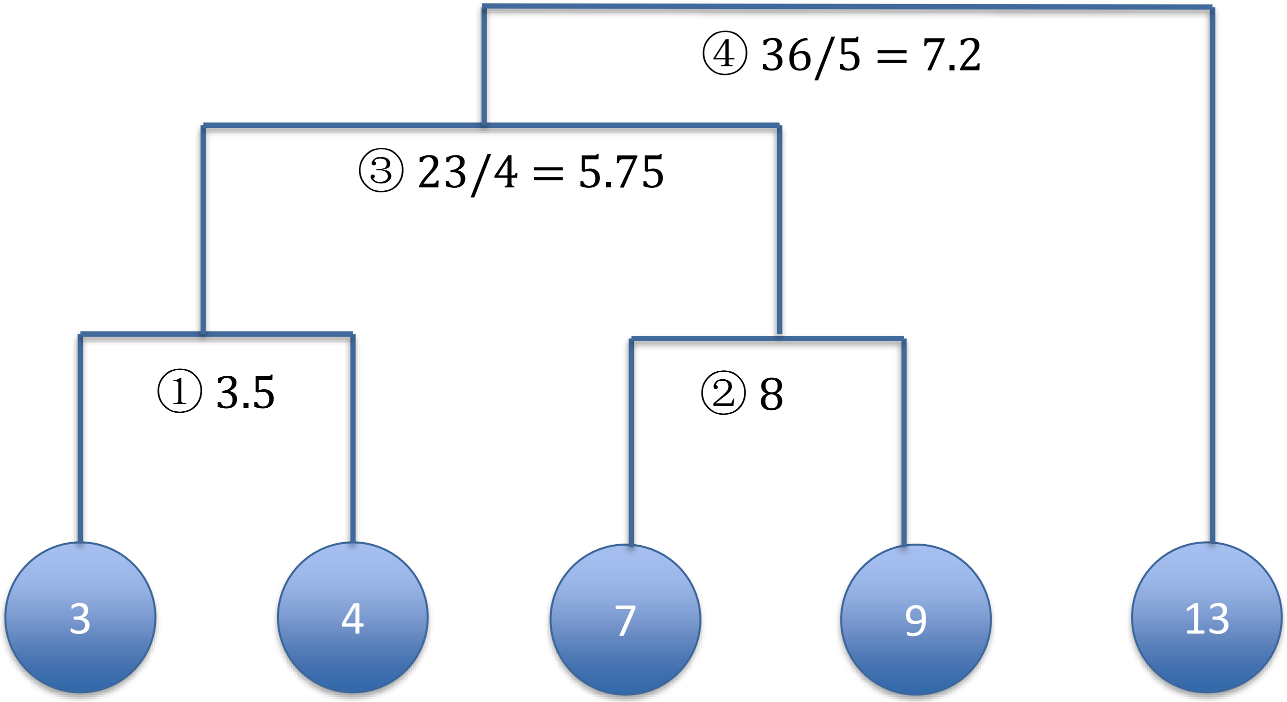
Hierarchical Clustering





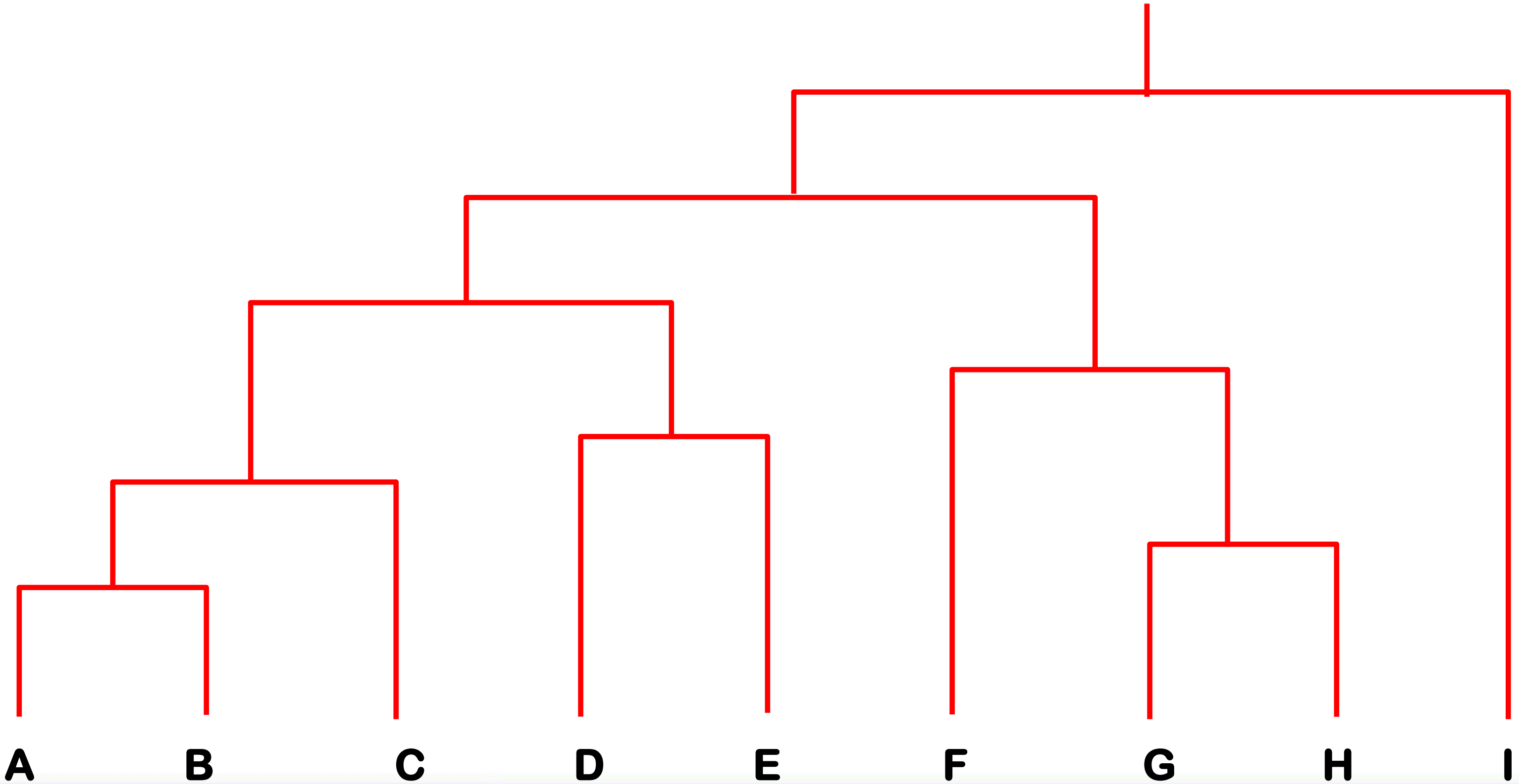
Hierarchical Clustering

- **For example: a list of numbers {9, 13, 7, 3, 4}**
  - Right now, we only have 2 clusters, re-calculate centroids
  - Next: calculate the distance between remaining clusters
  - Repeat until all the members are put into a single cluster



In-Class Practice

- **Given a list of numbers: 1, 2, 4, 9, 13, 22, 29, 34, 45, 66**
  - Build hierarchical clustering tree structure from bottom to the up
  - Use the mean as the representative of each cluster, Use centroid method to merge clusters



Hierarchical Algorithms

- Extensions from the Example
  - Each object is a one-dimension data point in our previous example
  - In general, each object is a multi-dimension vector
  - The hierarchical clustering process is still the same

In-Class Practice

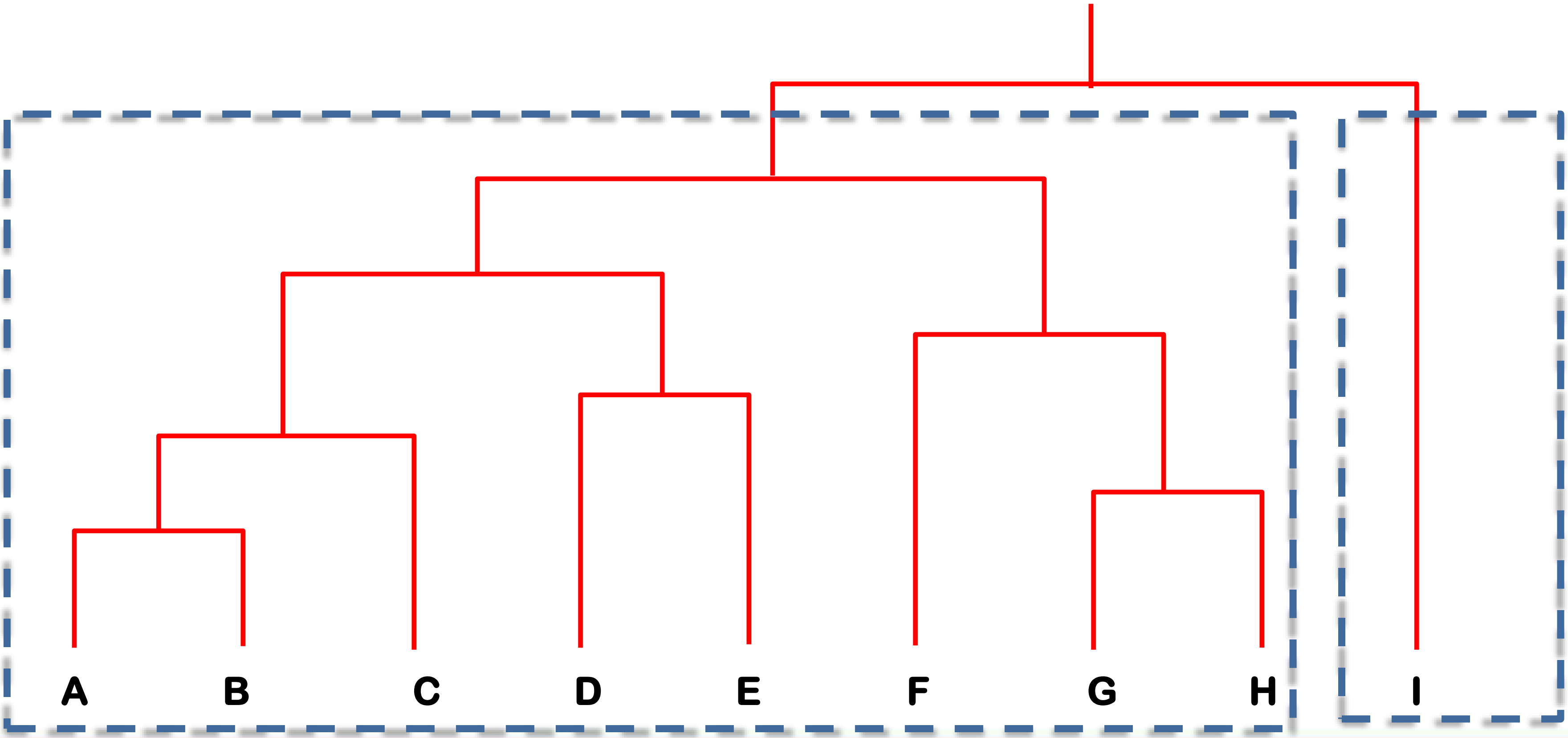
- **Given a list of objects**
  - Objects
    - <1, 1, 3>
    - <1, 2, 5>
    - <2, 3, 3>
    - <4, 1, 2>
    - <2, 2, 4>
    - <1, 5, 3>
  - Build hierarchical clustering tree from bottom to the up
  - Use the mean as the representative of each cluster, Use centroid method to merge clusters
  - Use Manhattan distance as metric

Hierarchical Clustering: How useful it is?

- The hierarchical clustering tree can tell the inner structure or relationships, such as parent/children, category/subcategory. You need to look into the objects after constructing such a hierarchical clustering tree.
- Hierarchical clustering results can also be used to create partitional clusters. You just need to find the appropriate number of the clusters from the top to the bottom levels

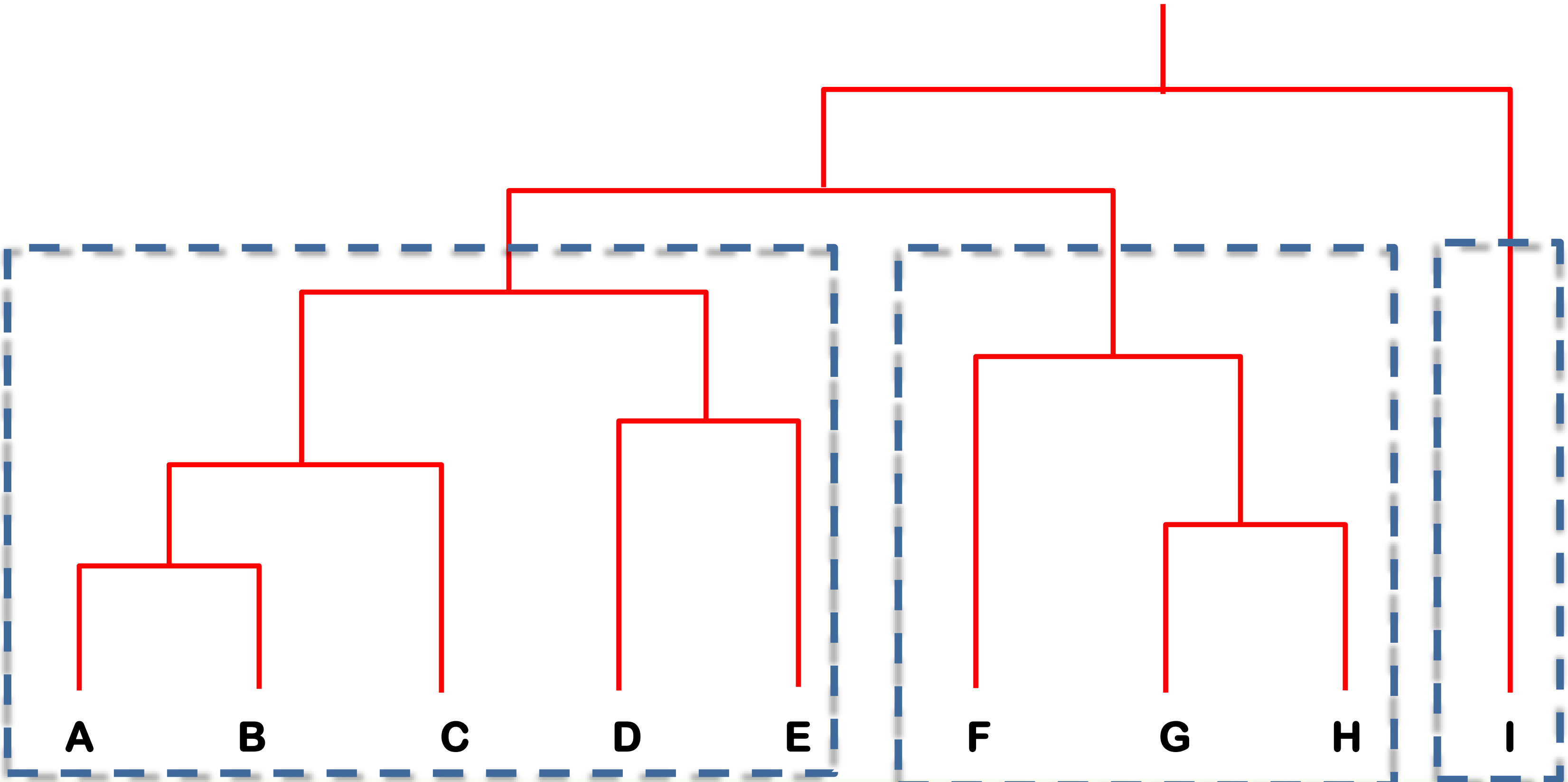
From Hierarchical to Partitional Clustering

- The dendrogram tells u the underlying structure of the data
- We can utilize dendrogram to produce partitional clusters
- **For example, if you need 2 clusters**



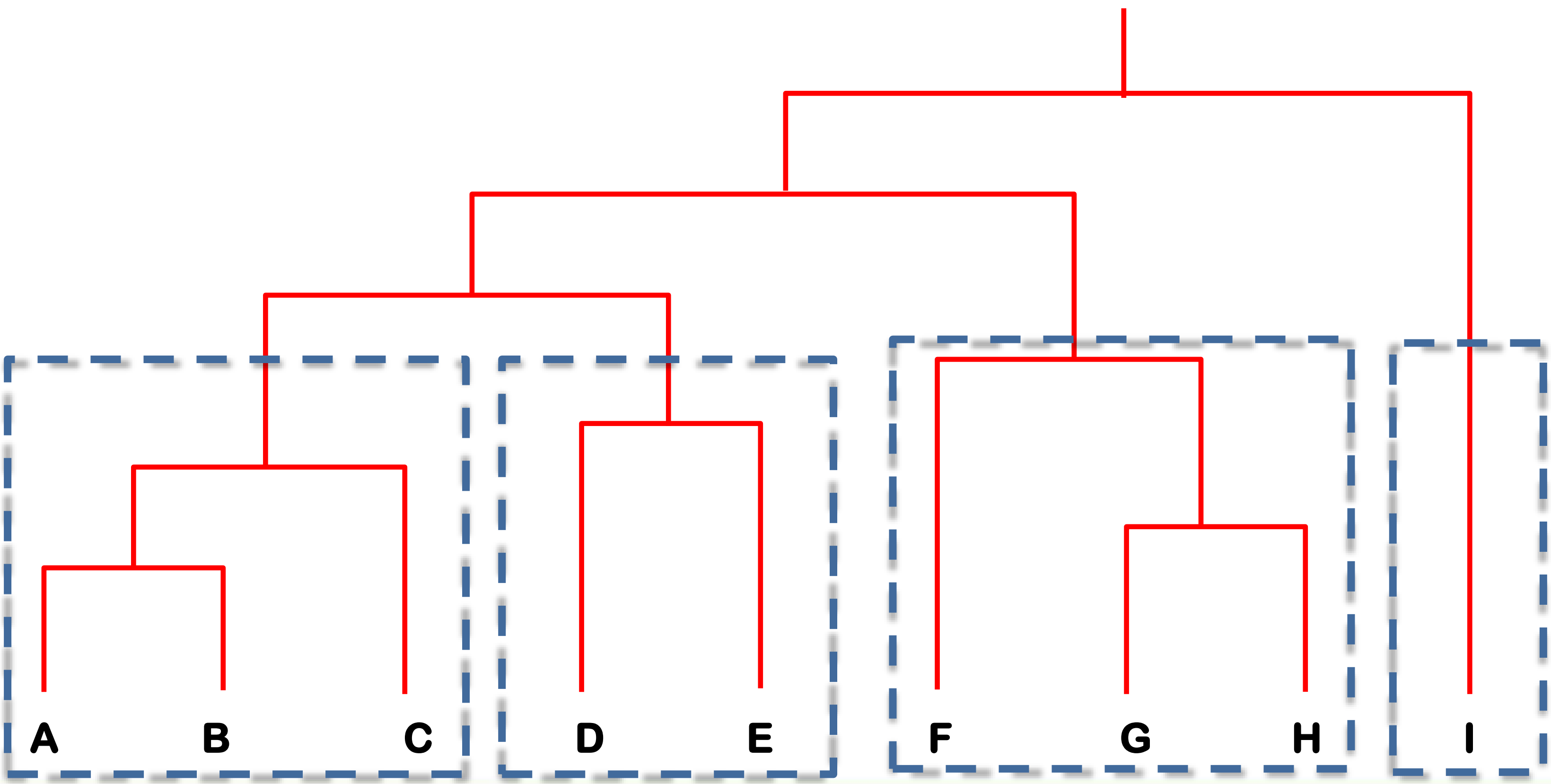
From Hierarchical to Partitional Clustering

- The dendrogram tells u the underlying structure of the data
- We can utilize dendrogram to produce partitional clusters
- **For example, if you need 3 clusters**



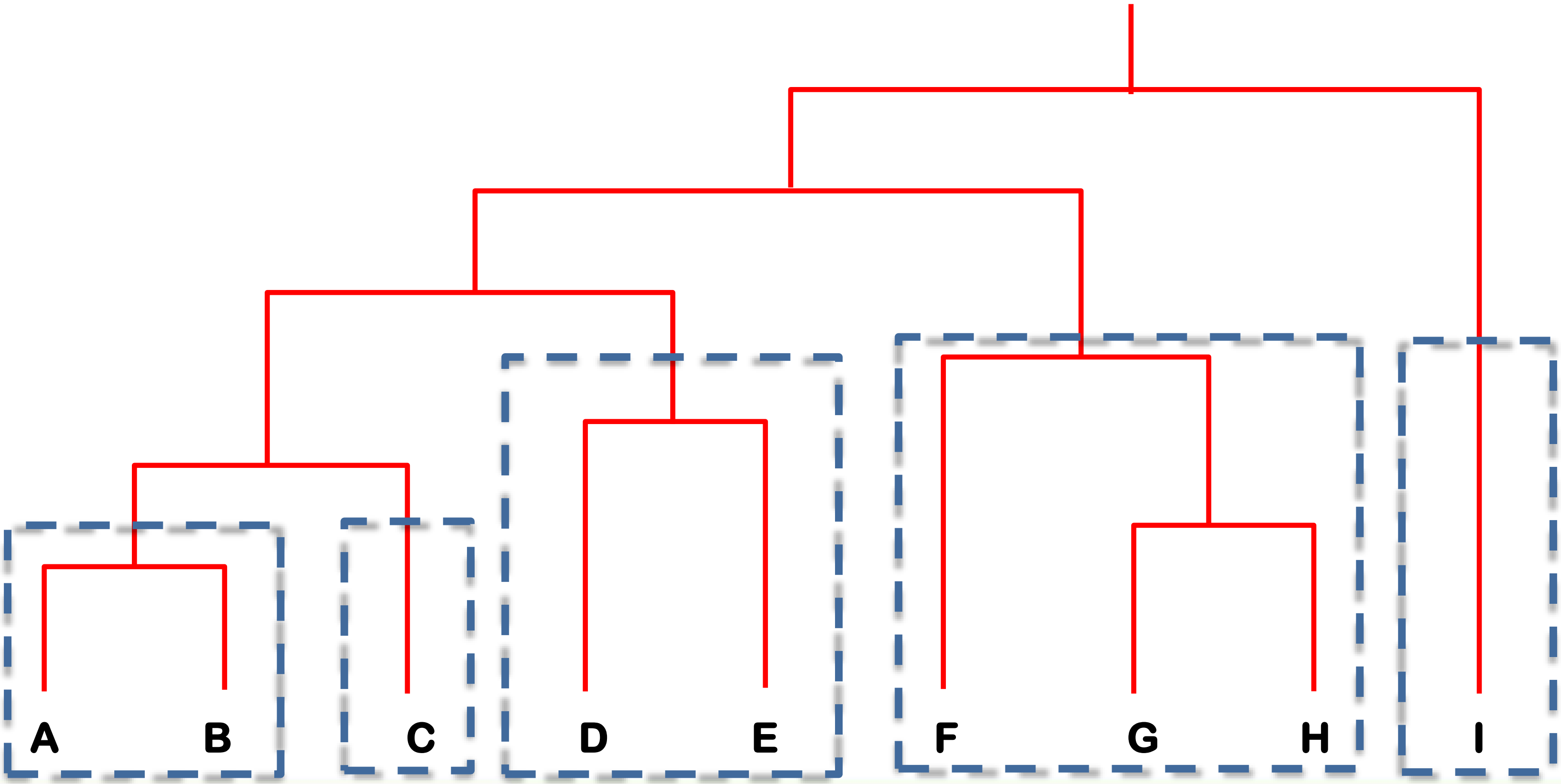
From Hierarchical to Partitional Clustering

- The dendrogram tells u the underlying structure of the data
- We can utilize dendrogram to produce partitional clusters
- **For example, if you need 4 clusters**



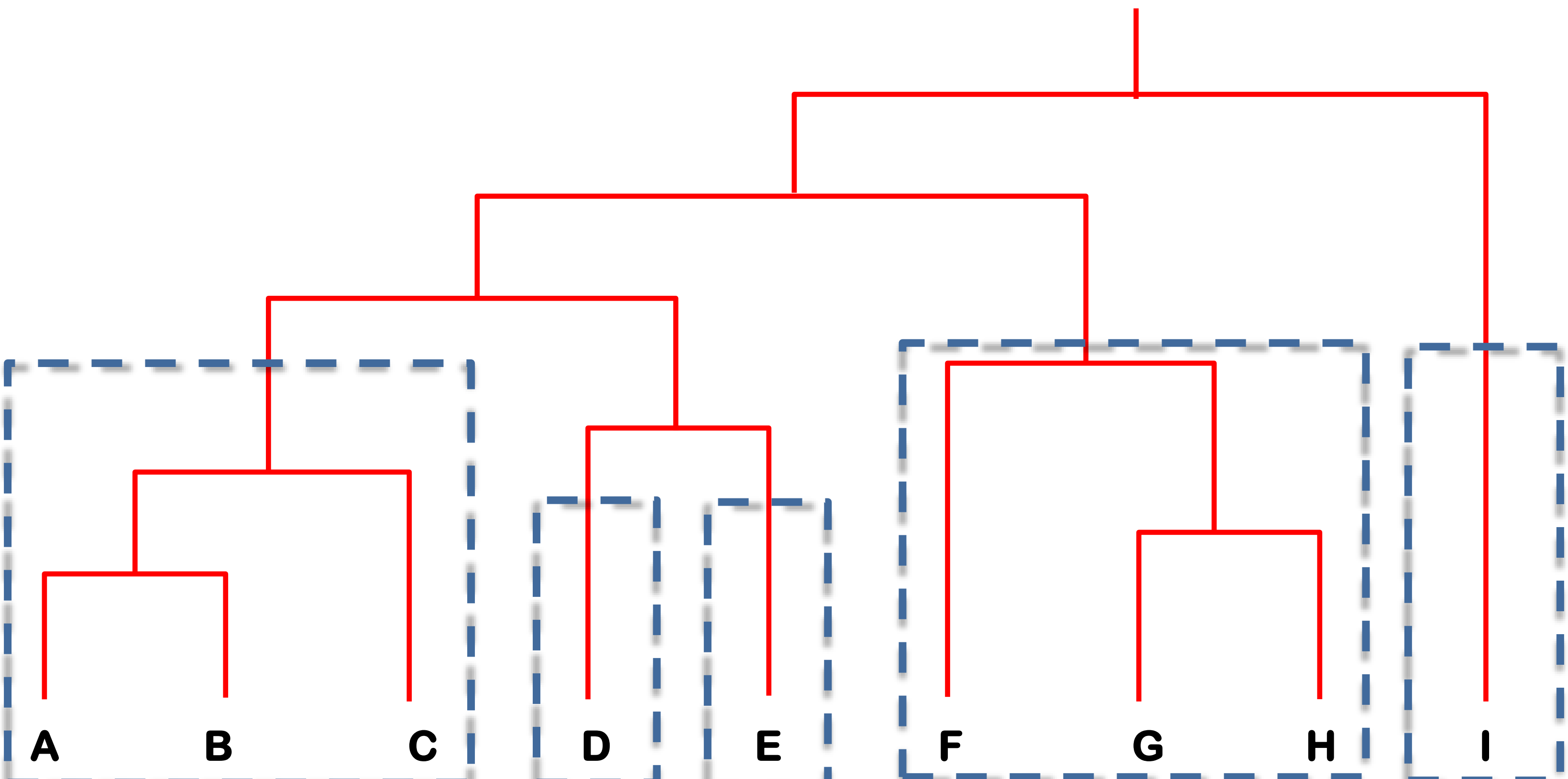
From Hierarchical to Partitional Clustering

- The dendrogram tells u the underlying structure of the data
- We can utilize dendrogram to produce partitional clusters
- **For example, if you need 5 clusters**



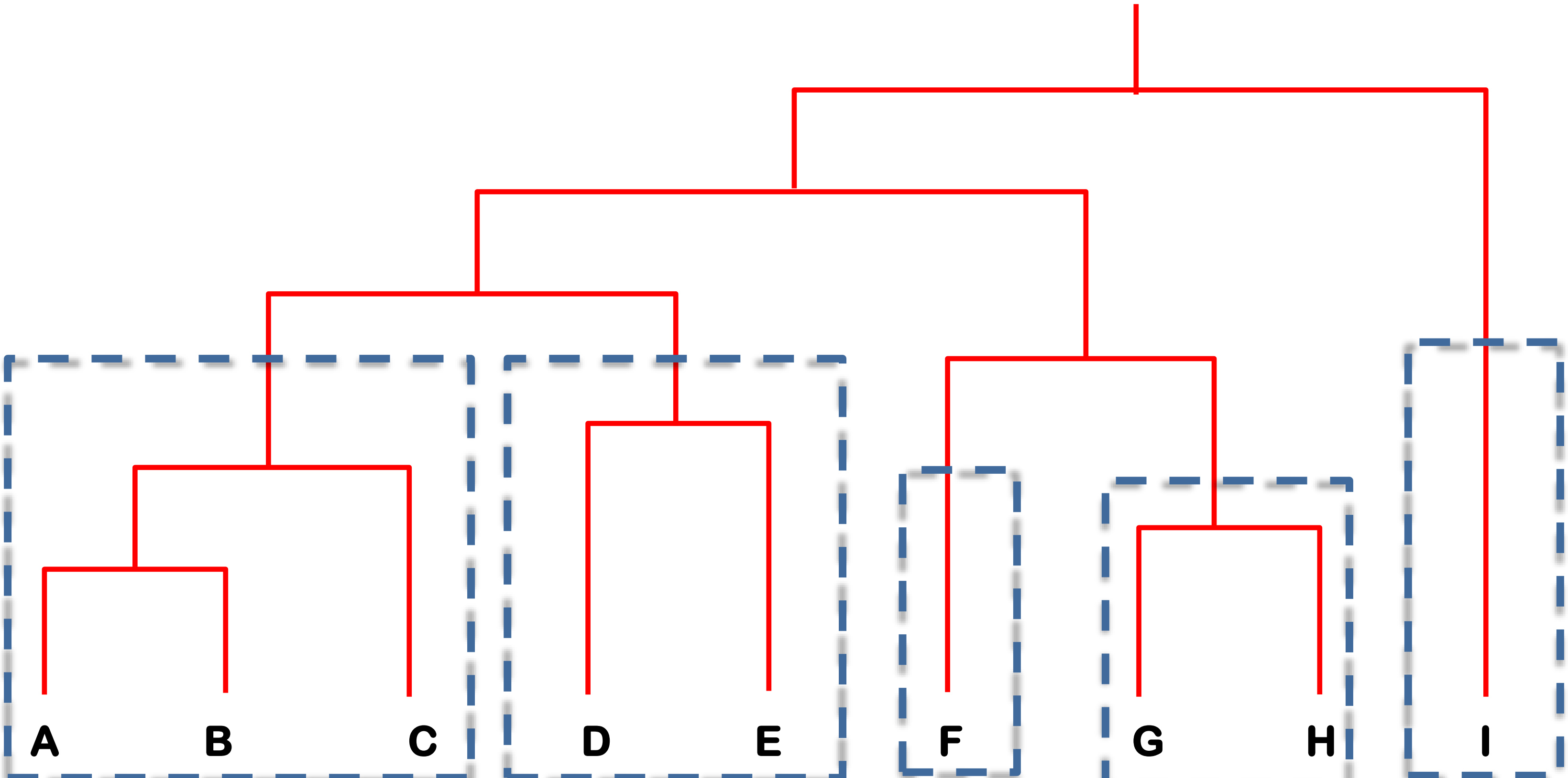
From Hierarchical to Partitional Clustering

- The dendrogram tells u the underlying structure of the data
- We can utilize dendrogram to produce partitional clusters
- **For example, if you need 5 clusters**



From Hierarchical to Partitional Clustering

- The dendrogram tells u the underlying structure of the data
- We can utilize dendrogram to produce partitional clusters
- **For example, if you need 5 clusters**



Hierarchical Clustering: How useful it is?

- The hierarchical clustering tree can tell the inner structure or relationships, such as parent/children, category/subcategory. You need to look into the objects after constructing such a hierarchical clustering tree.
- Hierarchical clustering results can also be used to create partitional clusters. You just need to find the appropriate number of the clusters from the top to the bottom levels
  - You can still use SSE to find the best number of clusters
  - But again, the evaluation process is still the same