Clustering

Tozammel Hossain

Data Science & Analytics
University of Missouri

What is Clustering Analysis?

- **Aka binning/segmentation/hashing**
- **Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters**
  - Number of clusters is not known ahead of time
- **Cluster: A collection of data objects**
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- **A type of Unsupervised Learning: no predefined classes**

Clustering Applications

- **Typical applications**
  - As a **stand-alone tool** to get insight into data distribution
  - As a **preprocessing step** for other algorithms
- **Biology:**
  - Taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- **Information retrieval:**
  - Document clustering
- **Land use:**
  - Identification of areas of similar land use in an earth observation database

Clustering Applications

- **Marketing:**
  - Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- **City-planning:**
  - Identifying groups of houses according to their house type, value, and geographical location
- **Climate:**
  - Understanding earth climate, find patterns of atmospheric and ocean
- **Economic Science:**
  - market research

Clustering as a Preprocessing Tool

- **Summarization of data**
- **Finding K-nearest Neighbors**
  - Localizing search to one or a small number of clusters
- **Outlier detection**
  - Outliers are often viewed as those “far away” from any cluster
- **Image Processing**
  - Compression: cluster similar colors -> replace all the colors within a cluster with one color

What Is Good Clustering?

- **A good clustering method will produce high quality clusters**
  - high intra-class similarity: **cohesive** within clusters
  - low inter-class similarity: **distinctive** between clusters

Clustering Types

- Representative-based Clustering
- Hierarchical Clustering
- Density-based Clustering
- Spectral and Graph Clustering

Representative-based Clustering

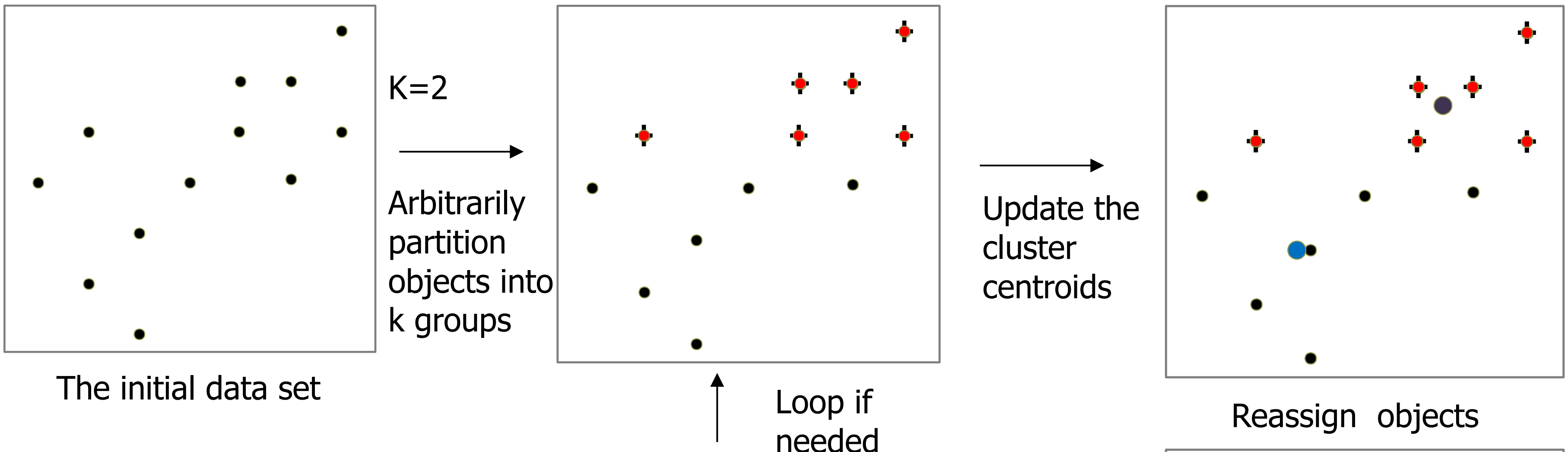
- **aka Prototype based clustering**
- **Given n data points and the number of desired cluster k**
  - partition the dataset into k groups or cluster
- **Data points in cluster are summarized with representative point**
  - Mean (aka centroid) of data points is popular
- **Brute-force/exhaustive approach**
  - generate all possible partitions of n points into k clusters:
    - $k^n/k!$: computationally infeasible with large n
  - evaluate some optimization score for each of them
  - retain the clustering that yields the best score

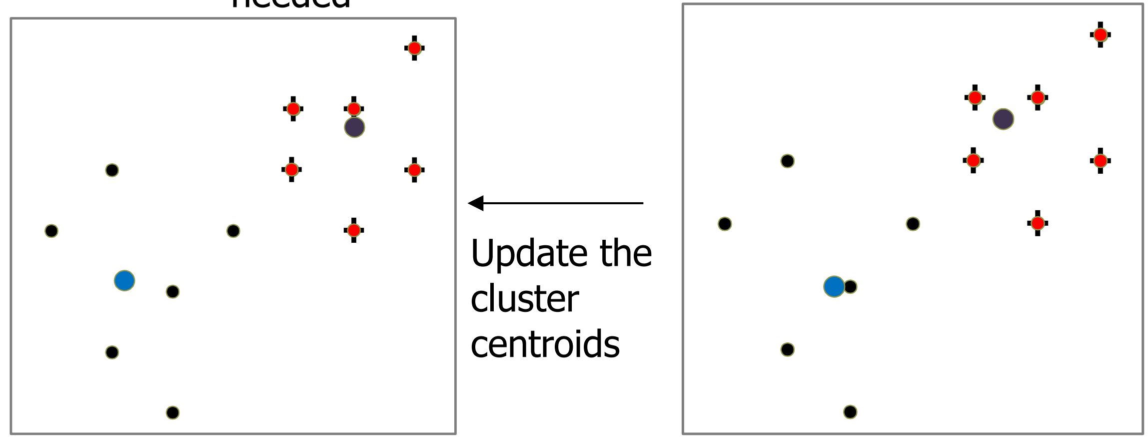
The *K-Means* Clustering Method

- **Given *k*, the *k-means* algorithm is implemented in four steps:**
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., **mean point**, of the cluster)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when the assignment does not change

$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

An Example of K-Means Clustering





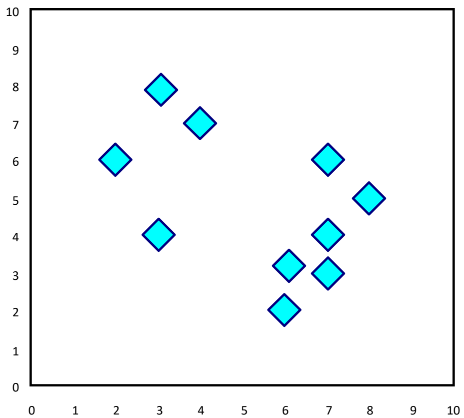
- **Partition objects into *k* nonempty subsets**
- **Repeat**
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- **Until no change**

Comments on K-Means

- **Efficient algorithm: runs very fast**
- **Often terminates at a *local optimal***
- ***Cons:***
  - Applicable only to objects in a continuous n-dimensional space
    - Using the k-modes method for categorical data
    - In comparison, k-medoids can be applied to a wide range of data
  - Need to specify *k,* the *number* of clusters, in advance
    - there are ways to automatically determine the best k
  - Sensitive to noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

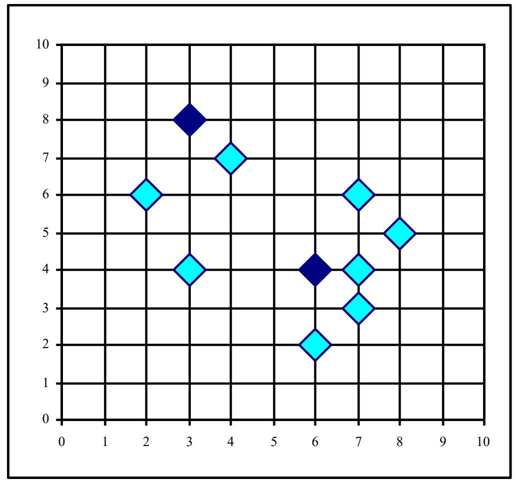
PAM: A Typical K-Medoids Algorithm

Total Cost = 20

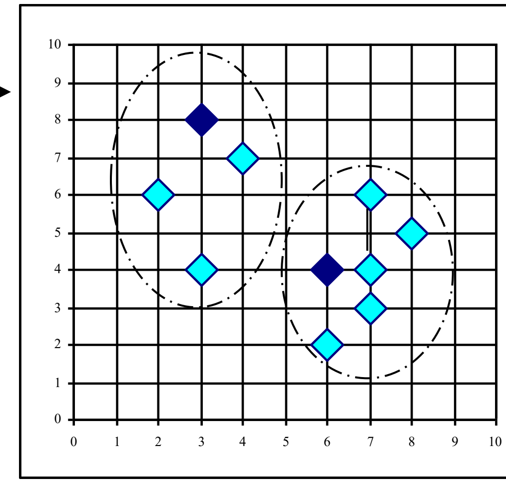


K=2

Arbitrary choose k object as initial medoids



Assign each remaining object to nearest medoids



Randomly select a nonmedoid object,O_{ramdom}

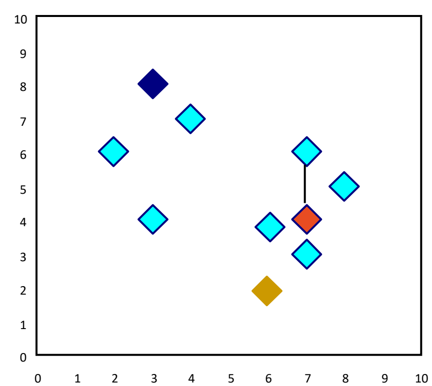
Total Cost = 26

Do loop

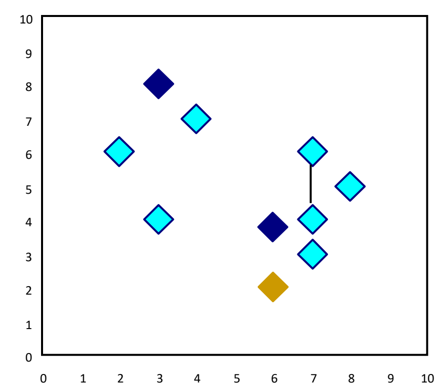
Until no change

Swapping O and O_{ramdom}

If quality is improved.



Compute total cost of swapping



sklearn implementation

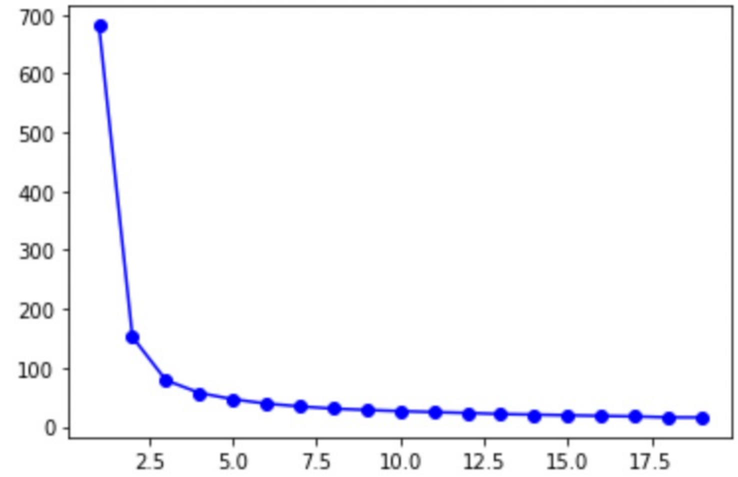
- **n_init = 10:**
  - run 10 times independently with different rando centroids
- **max_iter = 300:**
  - max number of iteration for each run
  - stops if it converges early
- **tol=1e-04**
  - stop if change in center < tol
- cluster_centers_: a signature

```python
from sklearn.cluster import KMeans


km = KMeans(
    n_clusters=3, init='random',
    n_init=10, max_iter=300,
    tol=1e-04, random_state=0
)
y_km = km.fit_predict(X)
```

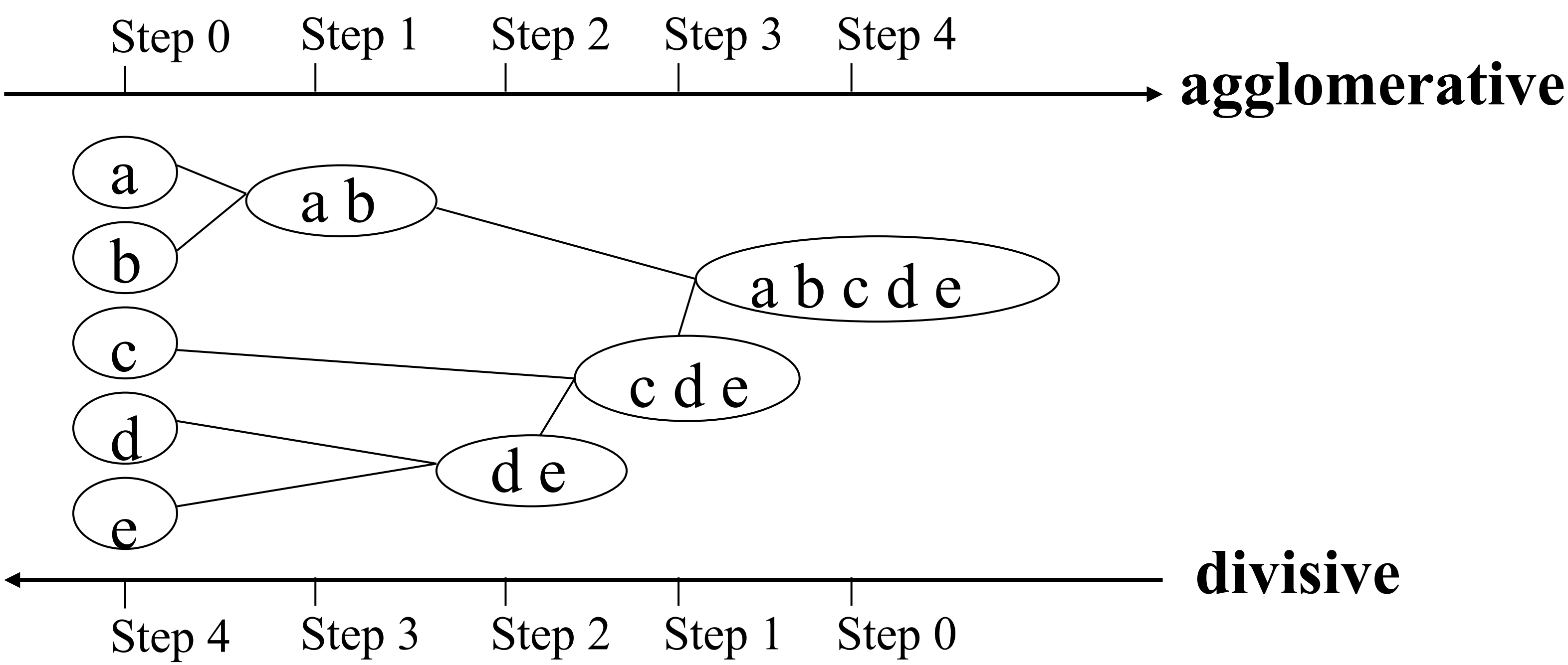
Choosing K

- **Elbow method**
  - Distortion/inertia vs K
  - Distortion: SSE $I = \sum_i (d(i, cr))$
- identify the value of *k* where the distortion begins to decrease most rapidly



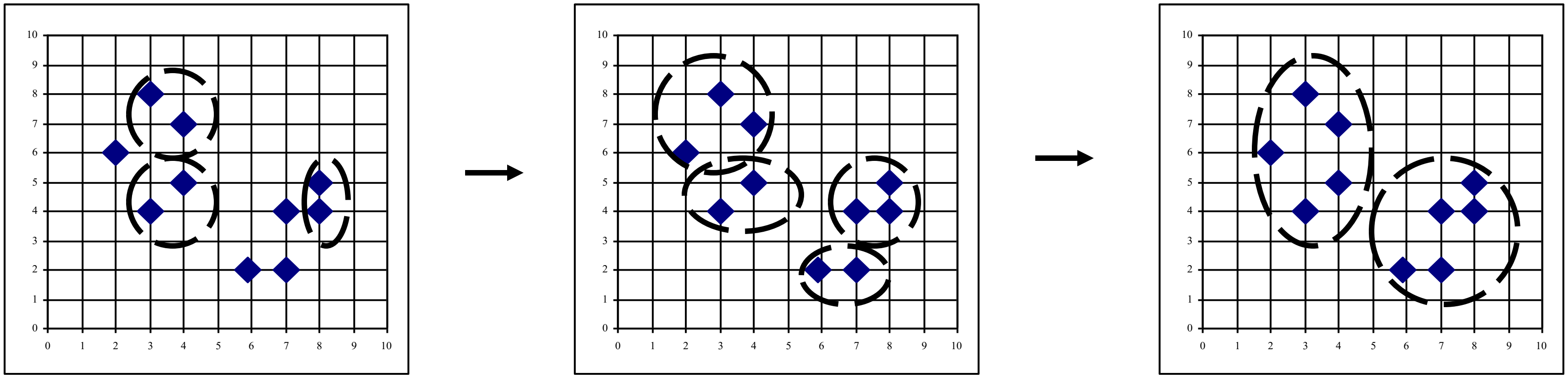
Hierarchical Clustering

- **Use distance matrix as clustering criteria**
  - No need to choose k
  - Need a terminating condition



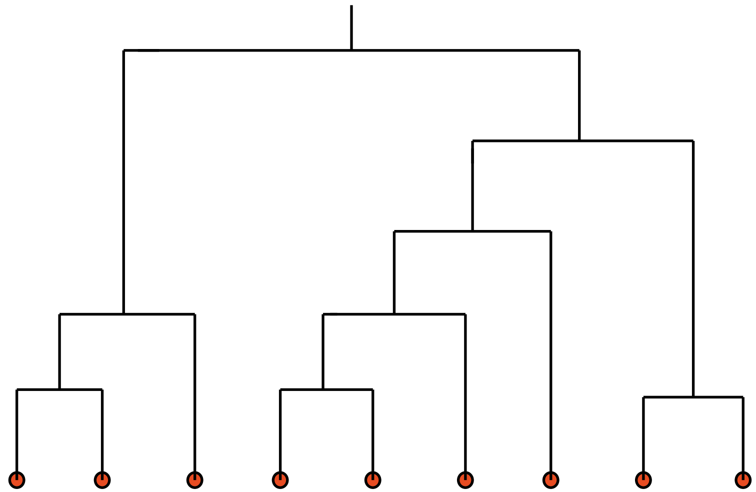
Agglomerative Clustering

- Use a link method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Eventually all nodes belong to the same cluster



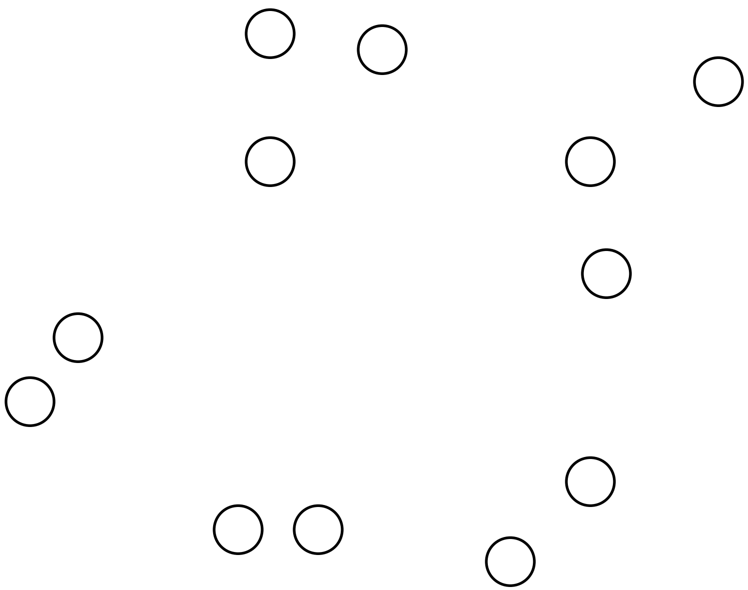
Dendrogram: Shows How Clusters are Merged

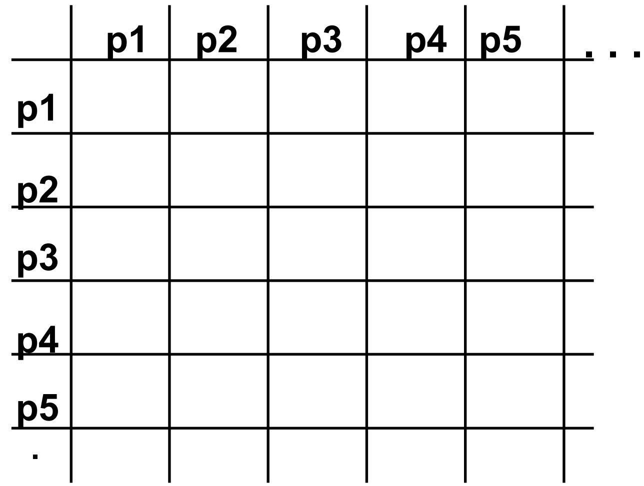
- **Decompose data points to several levels of nested partitioning**
  - Tree of clusters
- **A clustering is obtained by cutting the dendrogram at the desired level**



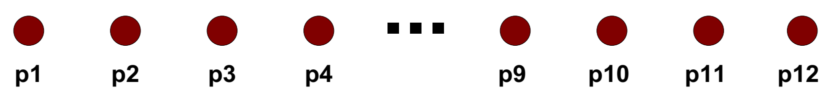
Steps 1 and 2

- **Start with clusters of individual points and a proximity matrix**





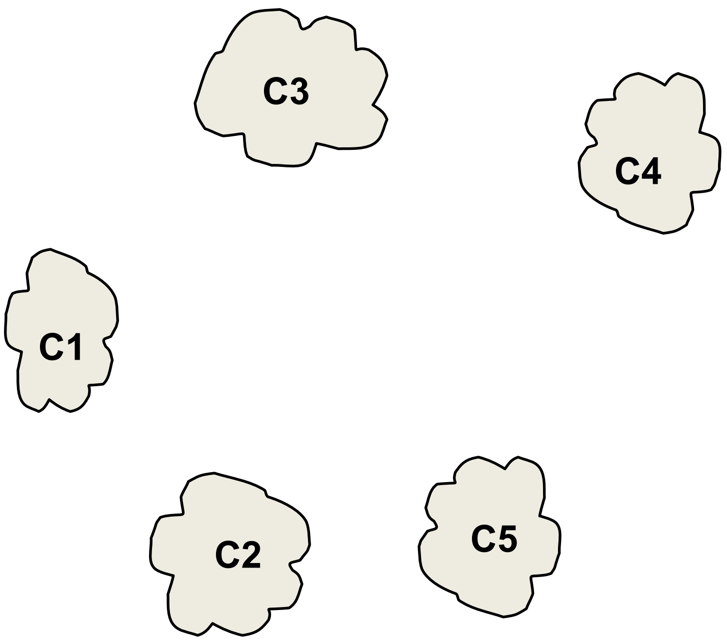
Proximity Matrix



This slide and the following ones are adapted from the book Intro to Data Mining by Tan et al.

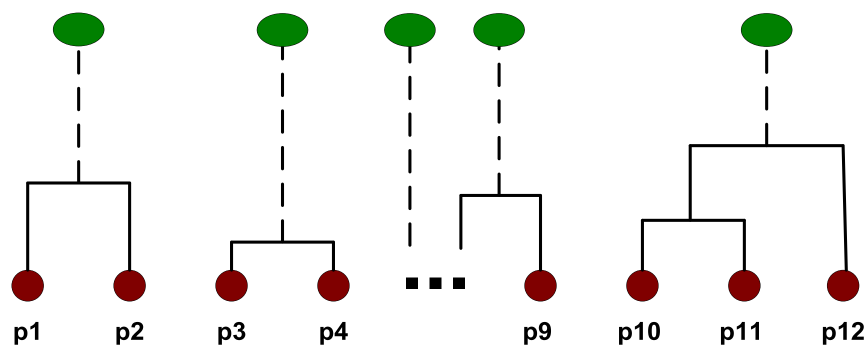
Intermediate Situation

- After some merging steps, we have some clusters



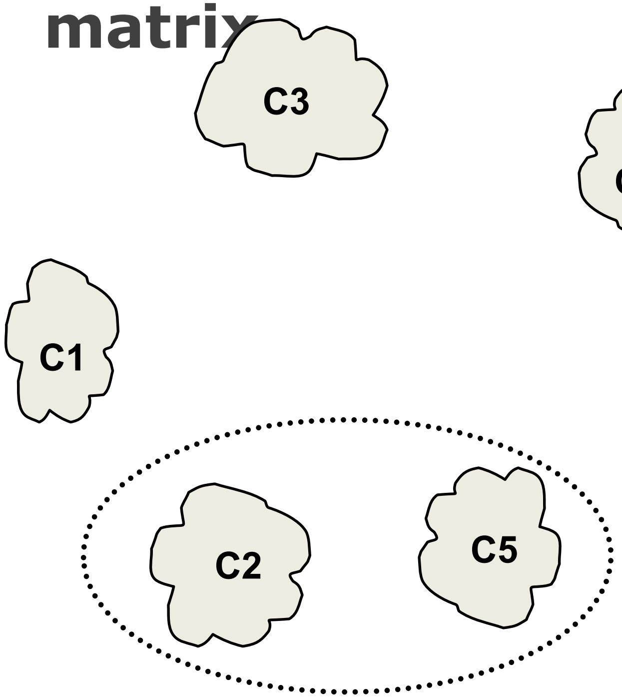
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

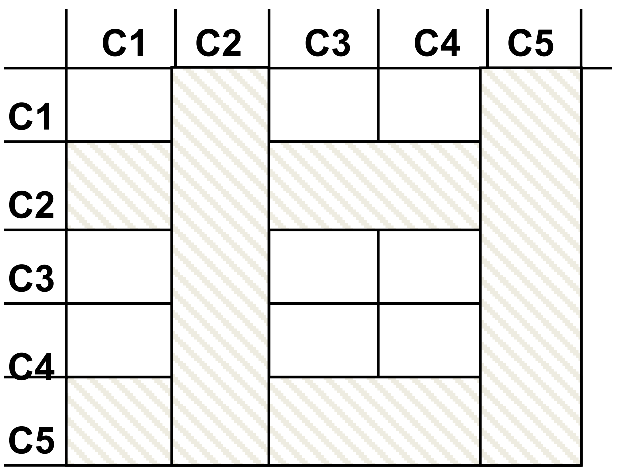
Proximity Matrix



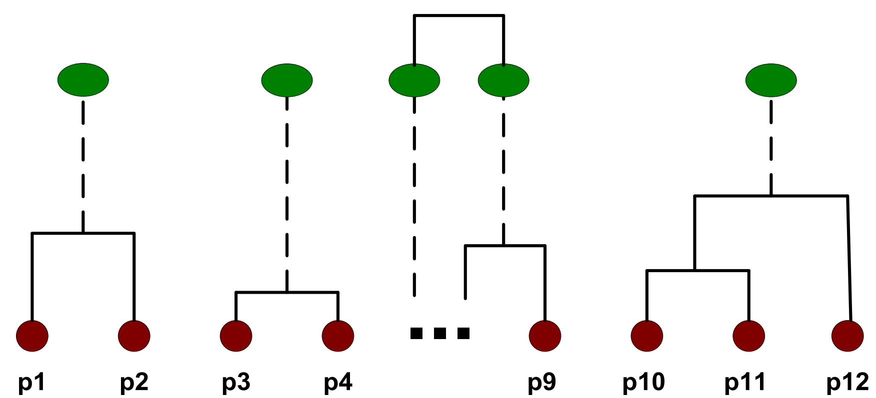
Step 4

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix



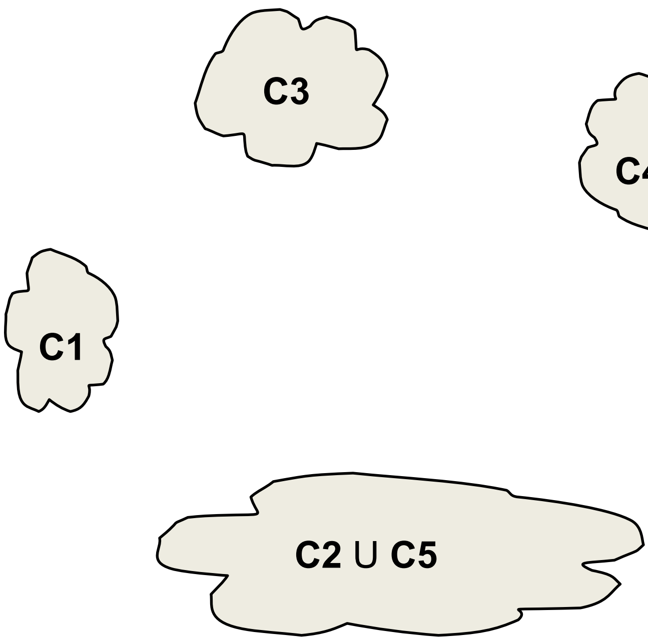


Proximity Matrix



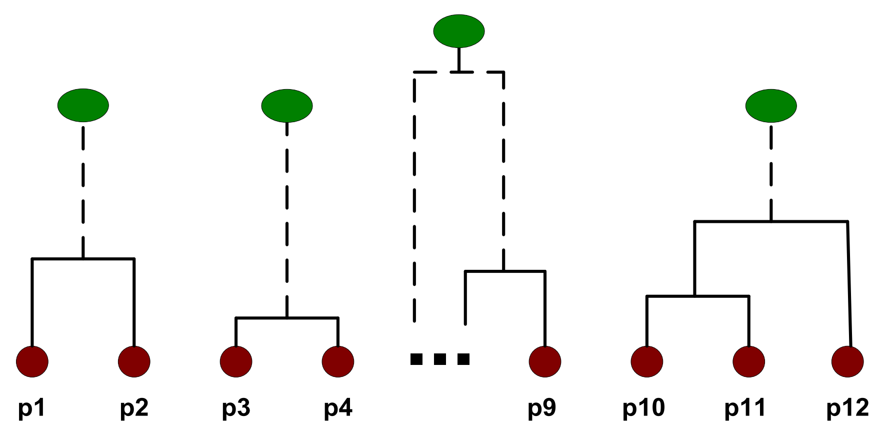
Step 5

- **The question is "How do we update the proximity matrix?"**

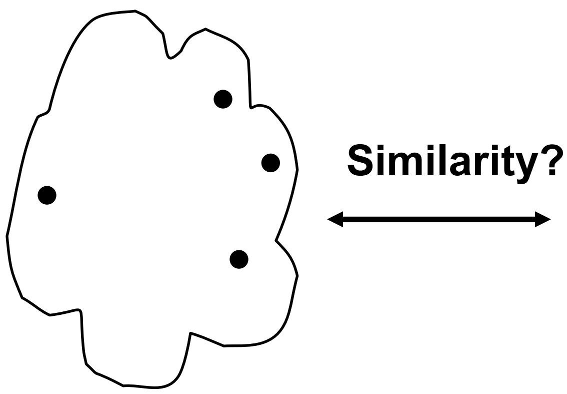


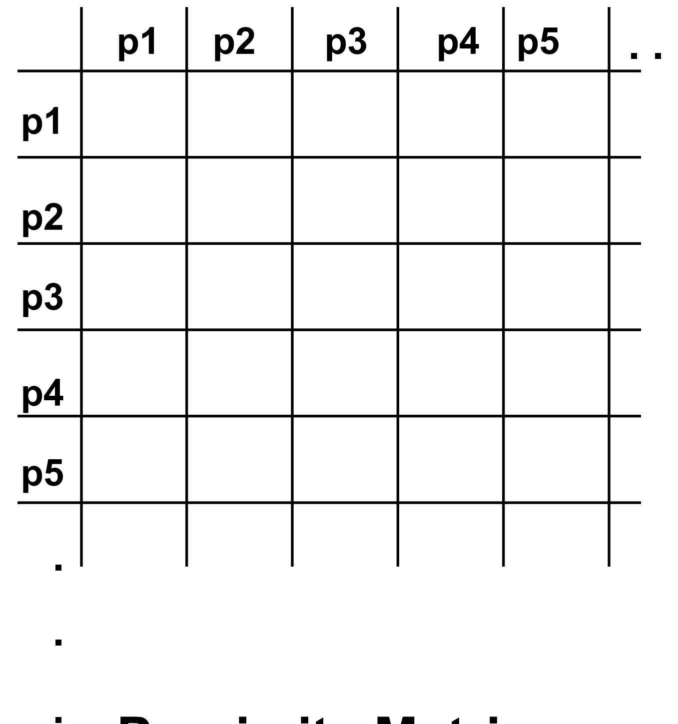
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Distance

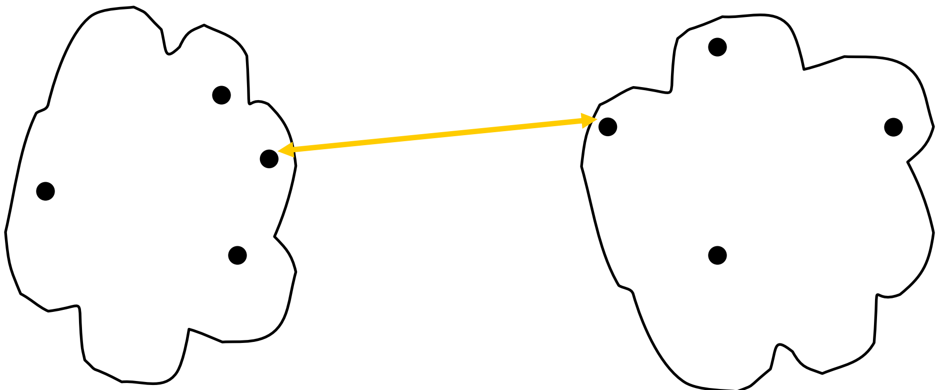




Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

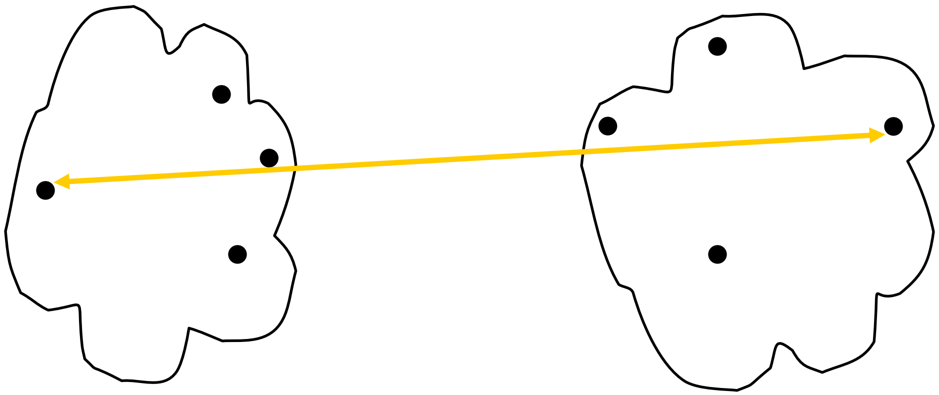


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

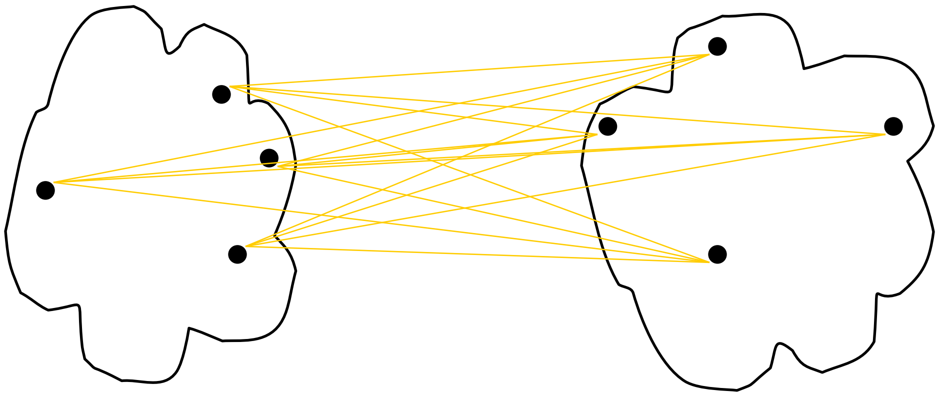


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

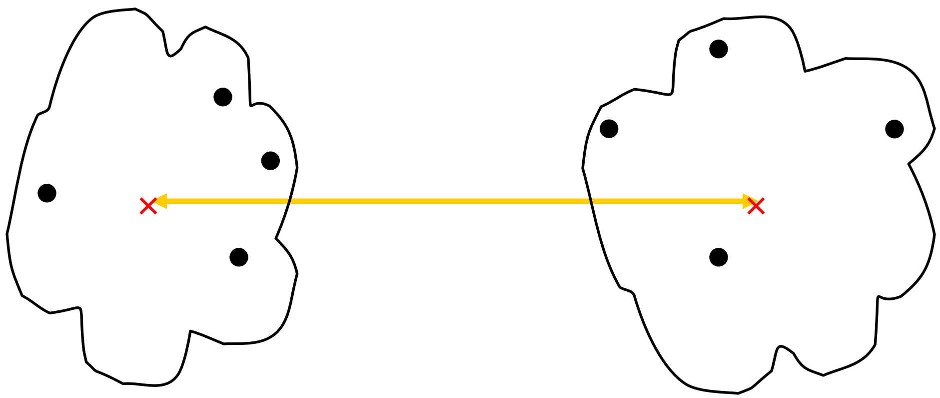


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

How to Define Inter-Cluster Similarity



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

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  – Ward’s Method uses squared error

Distance between Clusters

- **Clusters are merge based on distance**
- **Single link:**
  - smallest distance between an element in one cluster and an element in the other, i.e., dist(Ki, Kj) = min dist(t_ip, t_jq)
- **Complete link:**
  - largest distance between an element in one cluster and an element in the other, i.e., dist(Ki, Kj) = max dist(t_ip, t_jq)
- **Average:**
  - avg distance between an element in one cluster and an element in the other, i.e., dist(Ki, Kj) = avg dist(t_ip, t_jq)
- **Ward:**
  - based on minimizing the variance between clusters (SSE)

Issues with Hierarchical Clustering

- **Can never undo what was done previously**
  - Compare with k-means
- **Do not scale well**
  - time complexity $O(n^2)$

Density-Based Clustering

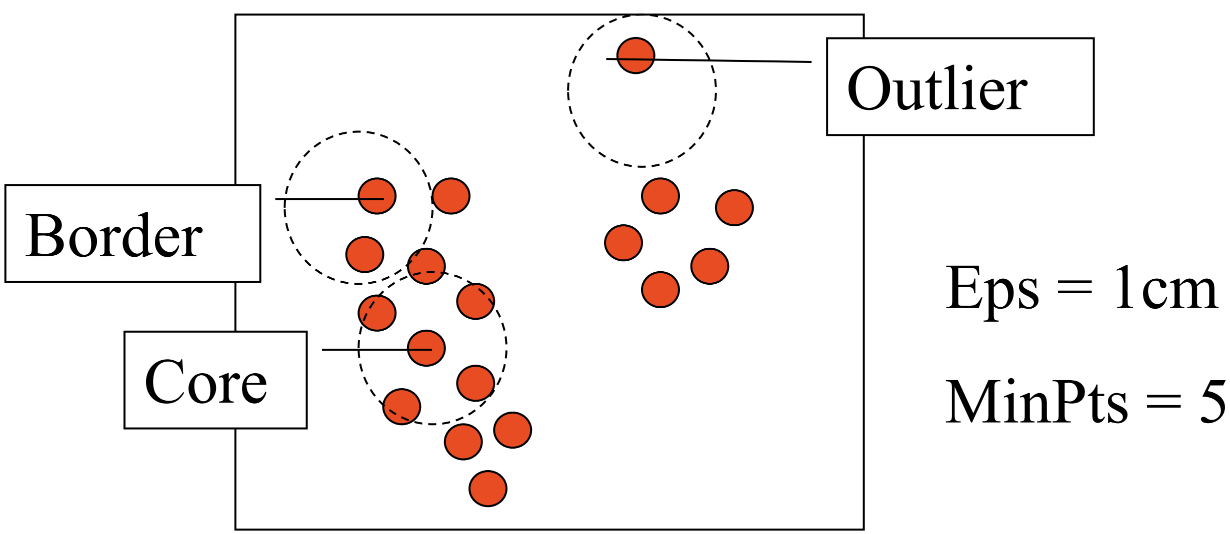
- **Clustering based on density (local cluster criterion), such as density-connected points**
- **Major features:**
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- **Example**
  - DBSCAN, OPTICS, DENCLUE

Density-Based Clustering: Basic Concepts

- Classifying points based on the characteristic of their local neighborhood
- **Two parameters:**
  - **Eps**: Maximum radius of the neighborhood
  - **MinPts**: Minimum number of points in an Eps-neighborhood of that point
- $\boldsymbol{N_{Eps}(p)}$**: {q belongs to D | dist(p,q) ≤ Eps}**

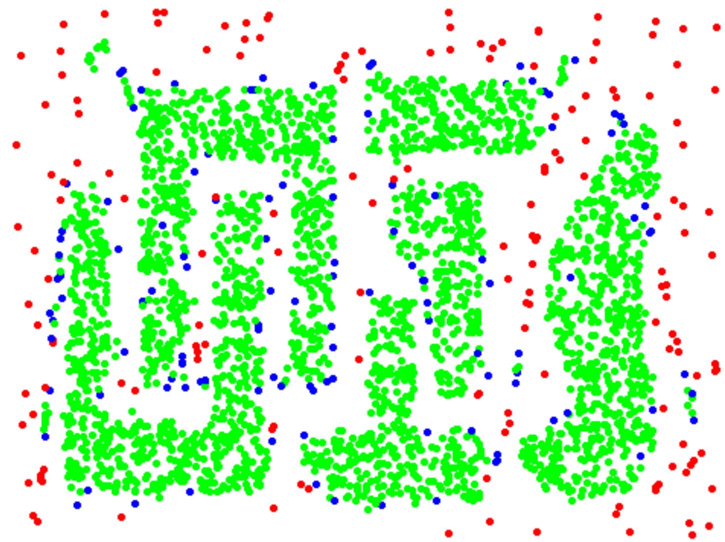
DBSCAN: Density-Based Spatial Clustering of Applications with Noise

- **Relies on a *density-based* notion of cluster:**
  - A *cluster* is defined as a maximal set of density-connected points
- **Discovers clusters of arbitrary shape in spatial databases with noise**



DBSCAN: Core, Border and Noise Points

Original Points

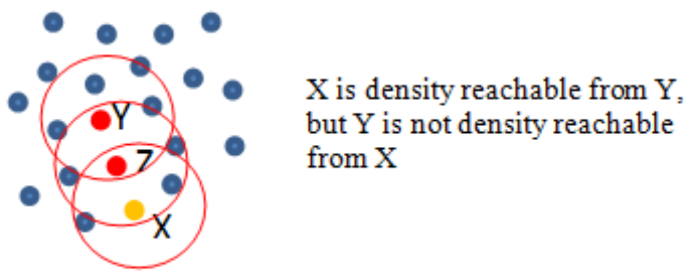


Point types: core, border and noise

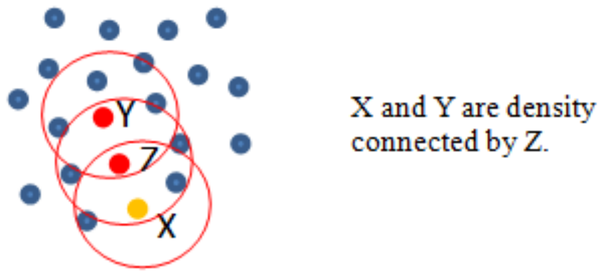
Eps = 10, MinPts = 4

DBSCAN: The Algorithm

- **A point is considered reachable from another point if there is a path consisting of core points between the starting and ending point**
- **Any point that is not reachable is considered an outlier**



a. Density-reachability of points



b. Density connectivity of points

DBSCAN: The Algorithm

- **Arbitrary select a point *p***
- **Retrieve all points density-reachable from *p* w.r.t. *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**

Measures of Cluster Validity

- **Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following two types.**
  - Supervised: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
    - Often called *external indices* because they use information external to the data
  - Unsupervised: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
    - Often called *internal indices* because they only use information in the data

- **You can use supervised or unsupervised measures to compare clusters or clusterings**

Unsupervised Measures: Cohesion and Separation

- **Cluster Cohesion: Measures how closely related are objects in a cluster**
- **Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters**
- **Example:**
  - Silhoutte score
  - Duhn Index