

CLUSTERING

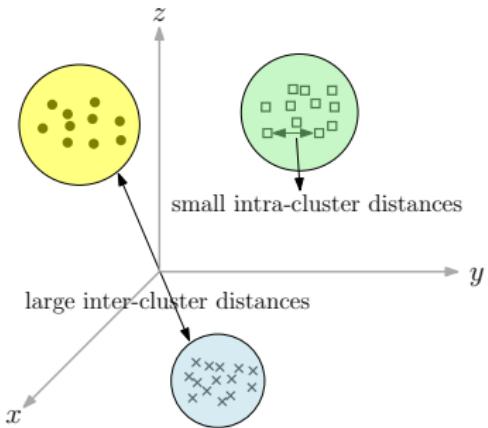
- Imdadullah Khan

Clustering: Definition

Clustering/cluster analysis/data segmentation

Grouping of objects into clusters such that objects in the same cluster are more similar and objects in different clusters are less similar

- Intra-cluster distances (between pairs of points in the same cluster)
- Inter-cluster distances (between pairs of points in different clusters)



Clustering: Definition

- The **clustering hypothesis**: Points in the same cluster behave similarly with respect to information needs
- Clustering is an supervised task, there is no right answer
- There is not even the right number of clusters

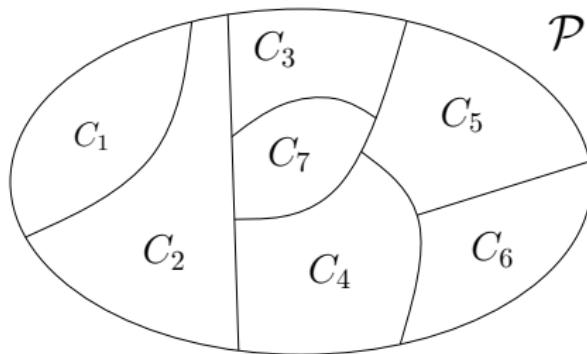
Clustering: Consideration

- Partitioning Level
 - Single level or multi-level (hierarchical partitioning)
 - Some times multiple levels of partitioning are required e.g.
 - Students partitioned at many levels, by schools, by departments, by major/minor, even further by CGPA
 - Books in a library clustered into subject areas, topics, sub-topics
- Exclusive or Non-Exclusive Clustering
 - Can points belong to more than one clusters (are clusters intersecting)
 - In social networks typically we get overlapping communities
- Similarity Measure
 - What is type of data, what similarity measure to be used
 - **It is important that similarity measure reflects inherent grouping in data**
- Clustering Space
 - Are all attributes of data points are to be considered (full space)
 - Clustering based on a subspace, e.g. for clustering students based on performance, gender and address info can be ignored

Clustering: Definition

Generally, clustering produces a partition $[C_1, C_2, \dots, C_k]$ of the dataset \mathcal{P}

- Each $C_i \subseteq \mathcal{P}$
- For $i \neq j$, $C_i \cap C_j = \emptyset$
- $\bigcup_{i=1}^k C_i = \mathcal{P}$



Clustering: Definition

Generally, clustering produces a partition $[C_1, C_2, \dots, C_k]$ of the dataset \mathcal{P}

Two broadly different ways of clustering depending on input

Input: Given a dataset (feature vectors) and a proximity measure

Output: Clusters of the dataset into k clusters

Alternatively,

Input: Given pairwise proximity values for a (abstractly described) dataset
(e.g. distance or similarity matrix)

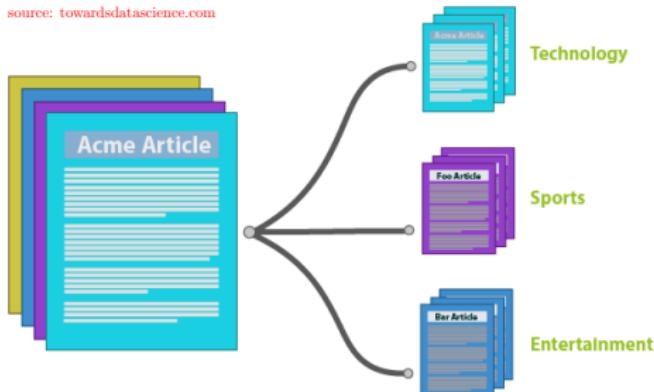
Output: Clusters of the dataset into k clusters

The number of clusters k may or may not be part of the input (fixed)

Clustering: Applications

Documents Clustering

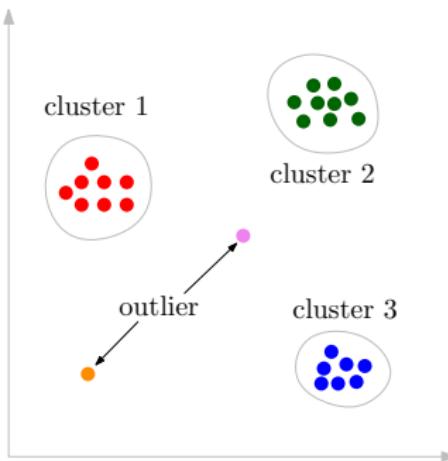
- Group documents based on their similarity with other documents
 - document similarity computed from their TF-IDF vectors
- Documents about same topic or written by same author ideally would form a cluster
 - e.g. sports, politics, entertainment, news
- Benefits: reduces search space, improves search and retrieval cost



Clustering: Applications

Outlier Detection

- Outliers are substantially different from other objects in a dataset
- Identify objects that do not belong to a cluster (or the object itself is a cluster)
- Benefits: fraud detection in financial transactions, data cleaning

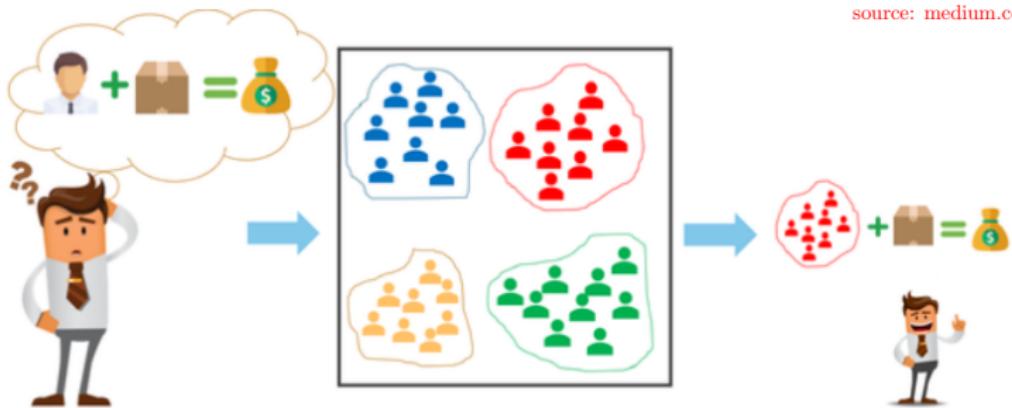


Clustering: Applications

Market Segmentation/ Business Intelligence

- Subdivide customers into distinct subsets
 - Customers in same subset share common characteristics
- Each subset can be target of different marketing campaigns
- Based on the target, use appropriate proximity measures
 - purchasing history, age, salary, nature of job etc.

source: medium.com



Trying to determine the appropriate audience for the product

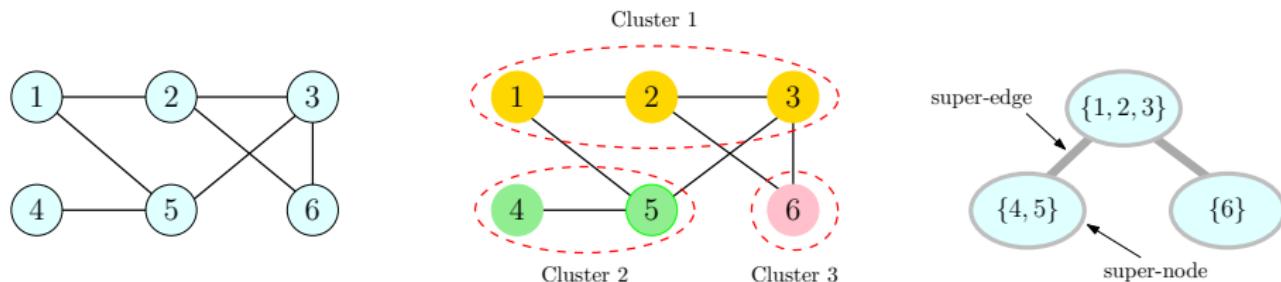
Using clustering algorithms on the customer base

Selling the product to the targeted audience

Clustering: Applications

Data Compression and Graph Summarization

- Make clusters in a graph
 - Each cluster corresponds to a super node in the graph
- Efficient storage, transmission, processing and analysis of graphs



Basic Clustering Methods

- Broadly clustering methods are
 - Distance Based
 - Density and grid-based methods
 - Generative Model based
 - Other methods used for specific data types
 - e.g. for graph data we used connectivity based clustering
- It is possible that different clustering methods generate different clusterings on same data set

Distance based Clustering

Assumes a meaningful proximity measure defined over the dataset \mathcal{P}

Can be categorized into

1 Point assignment based methods

- Require points as feature vectors and the distance measure
- Assume that number of required clusters k is provided
- Produces a single level partition of \mathcal{P} into k parts

2 Hierarchical methods

- Can work with the pairwise distance matrix without explicit points representation or the definition of distance measure
- Produces multi-level partitions of \mathcal{P}
- Does not require number of clusters k as input
- Can be further categorized into
 - Agglomerative methods (Bottom-Up Approach)
 - Divisive methods (Top-Down Approach)

k-Partition Problem

- Given a set of n points, $\mathcal{P} \subset \mathbb{R}^m$ and $k \in \mathbb{Z}$, number of clusters
- Assume Euclidean distance measure over \mathcal{P} ▷ I_p , cosine can be used
- For a subset $C_i \subseteq \mathcal{P}$, denote by \mathbf{c}_i the **centroid** of C_i

$$\mathbf{c}_i := \frac{1}{|C_i|} \sum_{x \in C_i} x$$

- Centroid is the arithmetic mean of m -dim vectors (coordinate-wise mean)
- Goodness of a k -partition $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ is measured by

$$\text{sum of squared error, } SSE(\mathcal{C}) = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mathbf{c}_i\|^2$$

also called Within SSE

- **Problem:** Find a k -partition \mathcal{C}^* of \mathcal{P} with minimum SSE
- Brute force approach (try all $\binom{n}{k}$ partitions) is **not feasible**

k-means Algorithm

- A basic greedy algorithm for the *k*-Partition problem

Algorithm : *k*-means algorithm (\mathcal{P}, k)

Select k random points as initial centroids

▷ Alternatives of centroids can be used

while Stopping criterion is not met **do** ▷ Many choices

 Assign each point $x \in \mathcal{P}$ to the centroid closest to x

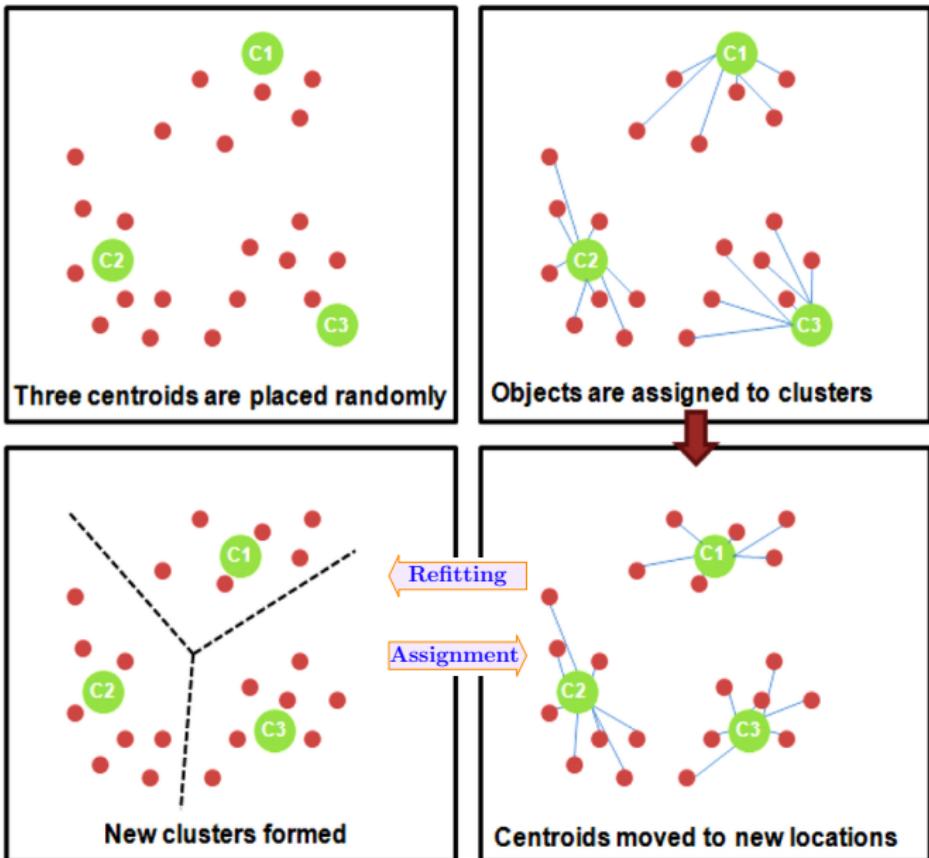
 ▷ closeness w.r.t the similarity measure

 ▷ Assignment Step

 Compute the centroids of (modified) clusters

 ▷ Refitting Step

k-means Algorithm: Illustration



k-means Algorithm: Runtime

Algorithm : k -means algorithm (\mathcal{P}, k)

Select k random points as initial centroids

► Alternatives of centroids can be used

while Stopping criterion is not met **do**

► Many choices

Assign each point $x \in \mathcal{P}$ to the centroid closest to x

▷ closeness w.r.t the similarity measure

Compute the centroids of (modified) clusters

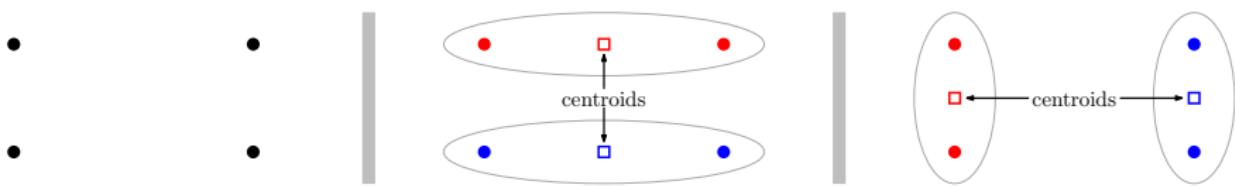
- Each iteration: $O(nk)$ distance computations
 - For each $x \in \mathcal{P}$ compute distances to centroids and find closest
 - Recompute centroids: in total takes $O(n)$ time
 - number of iterations is $t \implies$ total runtime is $O(tkn)$
 - t depends on the stopping rule, but generally, $t, k \ll n$
 - Total time is $O(n)$ distance computations, very efficient

k-means Algorithm: Stopping Criteria

- A clustering with smaller SSE than another is not necessarily better
- SSE depends on the value of k
 - $k = n \implies SSE = 0$
 - In general large $k \implies$ small SSE
- Stopping criterion could be
 - Stop when there is **minimal (less than a threshold)** change to SSE
 - Stop when no change in centroids
 - Stop when few points (less than a threshold) change their centroids

k-means Algorithm: Initial *k* centers

Quality of final clustering critically depends on initial centroids



Different initial centers lead to different clustering, maybe very suboptimal

k-means Algorithm: Initial *k* centers

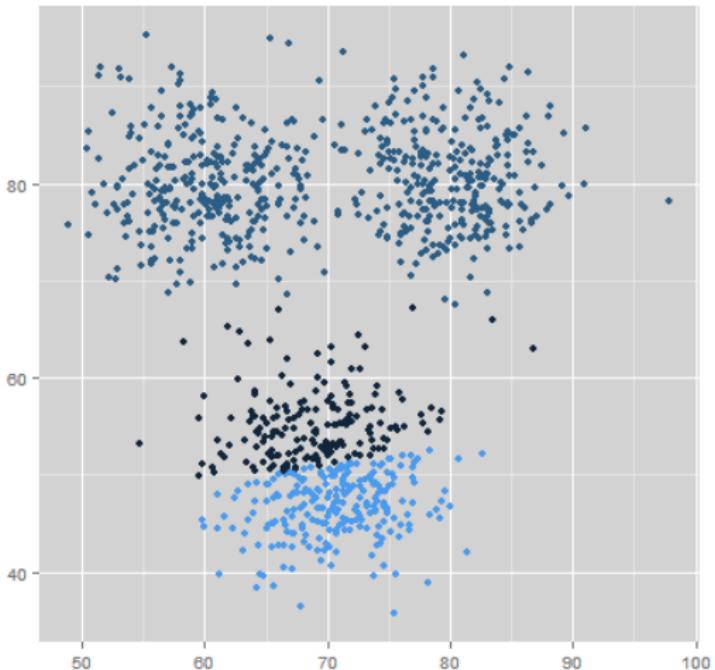
Quality of final clustering critically depends on initial centroids



Different initial centers lead to different clustering, maybe very suboptimal

k -means Algorithm: Initial k centers

Quality of final clustering critically depends on initial centroids



k-means Algorithm: Initial *k* centers

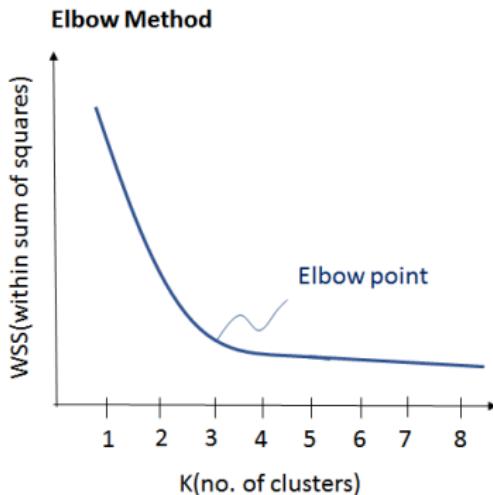
Quality of final clustering critically depends on initial centroids

Some methods used in the literature to choose initial centroids are

- Randomly chose k points in space of \mathcal{P} (e.g. \mathbb{R}^m)
- ***k*-means++:** Choose first point at random, choose the next point farthest from the first chosen point, repeatedly choose the next point that is farthest from the already chosen points
- Sample a subset of points. Run hierarchical clustering to get k clusters, choose the centroids of each cluster as initial centroids

k-means Algorithm: The right value of *k*

- Suppose SSE is the right clustering quality parameter
- Suppose k is the right number of clusters
- If we cluster into $k' < k$ clusters, then SSE will go up
- If we cluster into $k' > k$ clusters, then SSE will go sharply down
- Using this '*concavity*', the right value of k can be found with a binary search



k-means Algorithm: The right value of *k*

k-means algorithm is very sensitive to outliers because mean is an unstable statistic

- Let $\mathcal{P} = \{1, 2, 3, 8, 9, 10, 25\} \subset \mathbb{R}$
- The correct clustering looks like $\{1, 2, 3\}$, $\{8, 9, 10\}$ and 25 is an outlier
 - $SSE(\{1, 2, 3\}, \{8, 9, 10, 25\}) = 196$
 - $SSE(\{1, 2, 3, 8\}, \{9, 10, 25\}) = 189.67$
- *k*-means will select the latter partition
 - Clearly it is not good, as it separates 8 from 9 and 10

k-means Algorithm: dealing with instability

- One way to deal with the instability of centroids is to choose some other representative of each cluster
- Representative of a cluster is called **clusteroid**
- Since clusteroid, generally is somewhat central element of the cluster, it is also called **medoid**
- The goal here is to choose k clusteroids and minimize the sum of distances from each point to its clusteroid

k-Medians algorithm

- Median is less sensitive to outliers than mean
- We use '*median of clusters*' instead of centroids as clusteroids
- Let med_i be the '*median*' of a cluster C_i .
- Goodness of a k -partition $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ is measured by

$$S_{med}(\mathcal{C}) := \sum_{i=1}^k \sum_{x \in C_i} \|x_i - med_i\|^2$$

Problem: Find a k -partition \mathcal{C}^* of \mathcal{P} with minimum $S_{med}(\cdot)$

- Various definitions of median for points in higher dimensions
- Oja Median, Simplicial Median, 1-Median, coordinate-wise median

k-medians Algorithm

- We give a generic version of *k*-medians algorithms

Algorithm : *k*-medians algorithm (\mathcal{P}, k)

Select k points as initial medians ▷ randomly or arbitrarily

while Stopping criterion is not met **do** ▷ many choices

 Assign each point $x \in \mathcal{P}$ to the median closest to x ▷ closeness w.r.t
the similarity measure

 Compute the medians of (modified) clusters

▷ using the adopted definition of median

Partition Around Medoids

A pseudocode of Partition Around Medoids (PAM) is as follows:

Algorithm : Partition Around Medoids (\mathcal{P}, k)

Select k points as initial clustroids (medoids) arbitrarily

while Stopping criterion is not met **do** > many choices

 Choose a non-medoid point p

 Compute change in SSE with replacing a medoid m with p

 If the change in SSE is negative, then swap m with p

Runtime is $O(k(n - k)^2)$ in each iteration

k-modes algorithm: Categorical Data

- *k*-Means or *k*-Medians cannot handle nominal data
- *k*-Modes algorithm is an extension for nominal data
- It just replaces mean with mode of cluster
- Mode of multidimensional data is vector of coordinate wise modes
- Some distance to the clusteroid also needs to be defined
- In the above definition of modes, distance to clusteroid (mode) can be for instance the Hamming distance
- We can use any of the distance measures discussed for nominal data

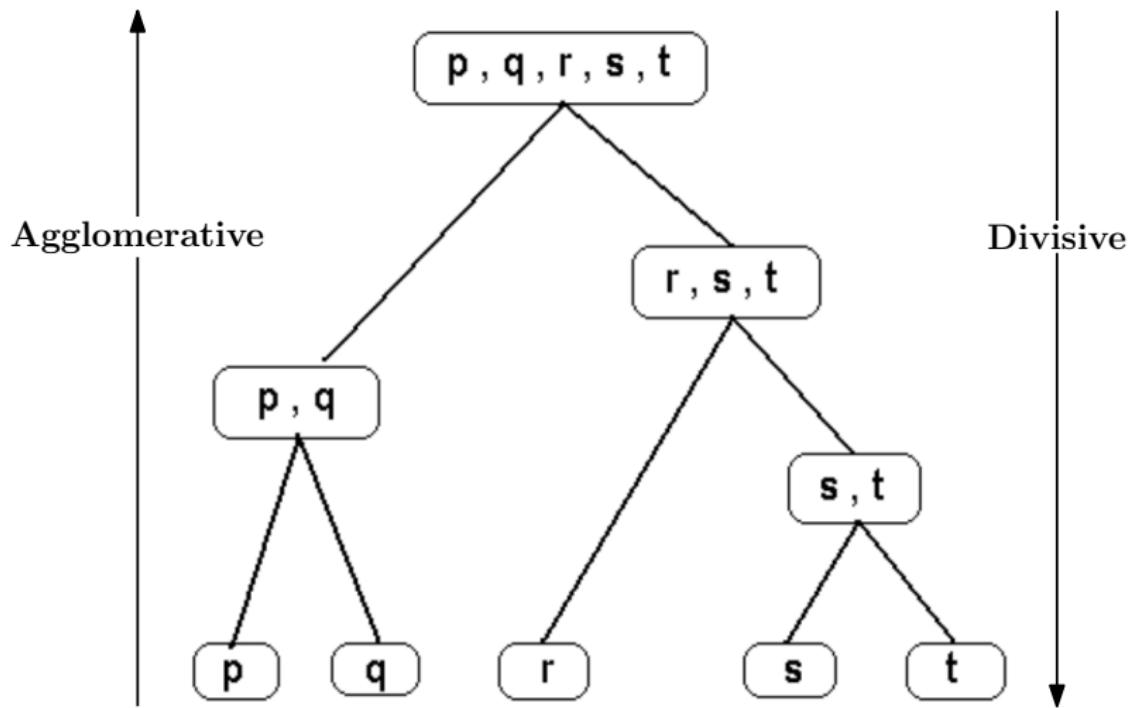
Hierarchical Clustering

Hierarchical Clustering

- Creates a hierarchy of clusters (multi-level partitions)
 - returns a set of nested clusters
- Generally no requirement of a fixed number of k clusters
- Hierarchical method can be
- **Divisive Approach (Top-Down)**
 - Initially all points are in one huge cluster
 - In every step one current cluster is split into two
 - generates a top-down hierarchy of clusters
- **Agglomerative Approach (Bottom-Up)**
 - Initially each point is a cluster itself
 - In every step two clusters are merged into one
 - generates a bottom-up hierarchy of clusters

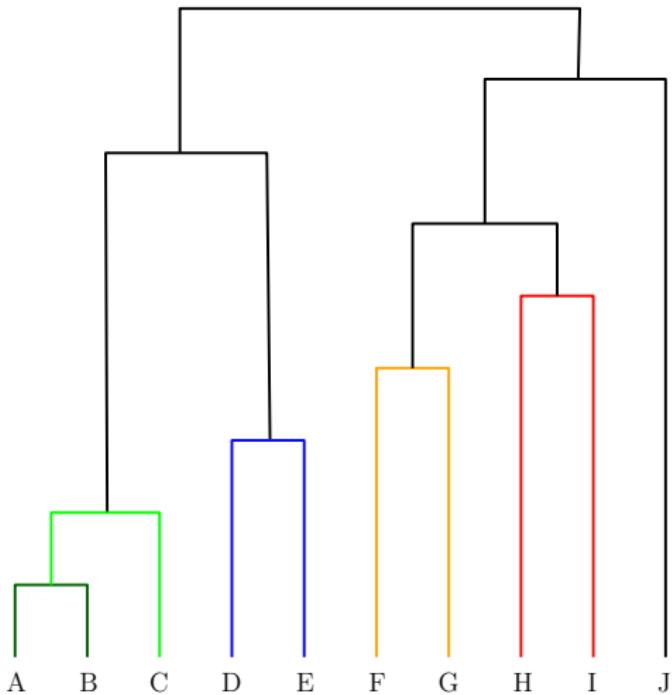
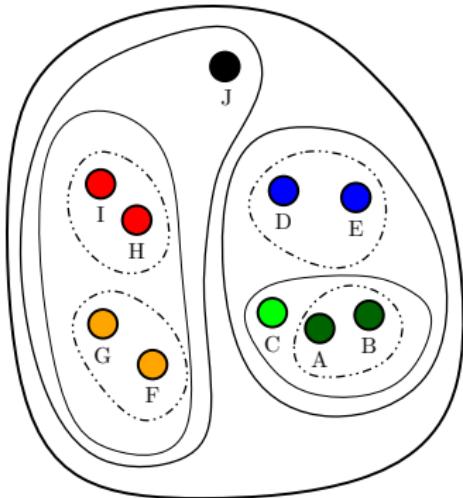
Hierarchical Clustering

Hierarchical Clustering: Agglomerative and Divisive Approach



Hierarchical Clustering

Output of hierarchical clustering is represented by a **dendrogram**
(a tree recording the sequence of merges or splits)



Hierarchical Clustering: Divisive Approach

We will discuss [spectral clustering](#) a divisive clustering approach

Hierarchical Clustering: Agglomerative Approach

Agglomerative Clustering

- Initially each point is a cluster itself
- In every step two '*close by*' clusters are merged into one
- Generates a bottom-up hierarchy of clusters

Key considerations:

- Representation of clusters
- Distance between clusters
- The choice of pairs of clusters to be merged
- A stopping criterion

Agglomerative Clustering

Algorithm : Generic Agglomerative Clustering (\mathcal{P})

Initialize with each point as a cluster in \mathcal{C}

while stopping criterion is not met **do**

 Choose the best pair of clusters $(C_i, C_j) \in \binom{\mathcal{C}}{2}$

$C_m \leftarrow \text{MERGE}(C_i, C_j)$

$\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_i, C_j\} \cup \{C_m\}$

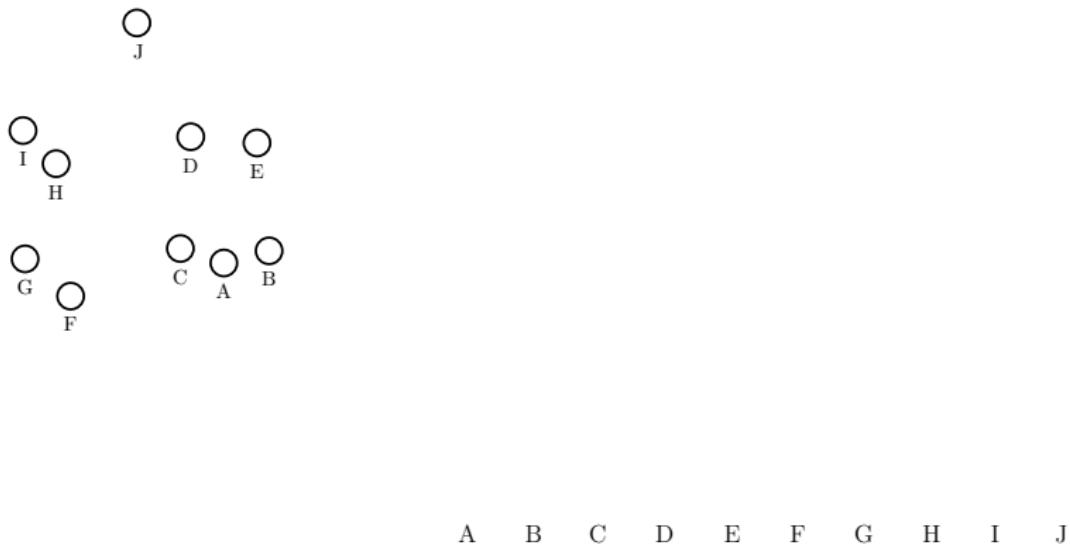
Agglomerative Clustering

Generic Agglomerative Clustering

$$\mathcal{P} = \{A, B, C, D, E, F, G, H, I\}$$

Initialize with each point as a cluster in \mathcal{C}
while stopping criterion is not met do

Choose the best pair of clusters $(C_i, C_j) \in \binom{\mathcal{C}}{2}$
 $C \leftarrow \text{MERGE}(C_i, C_j)$
 $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_i, C_j\} \cup \{C_m\}$



Agglomerative Clustering

Generic Agglomerative Clustering

$$\mathcal{P} = \{A, B, C, D, E, F, G, H, I\}$$

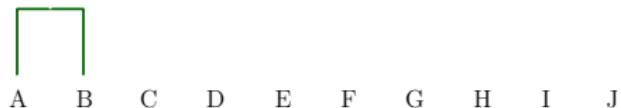
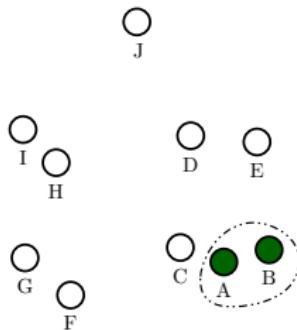
Initialize with each point as a cluster in \mathcal{C}

while stopping criterion is not met **do**

 Choose the best pair of clusters $(C_i, C_j) \in \binom{\mathcal{C}}{2}$

$C \leftarrow \text{MERGE}(C_i, C_j)$

$\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_i, C_j\} \cup \{C_m\}$



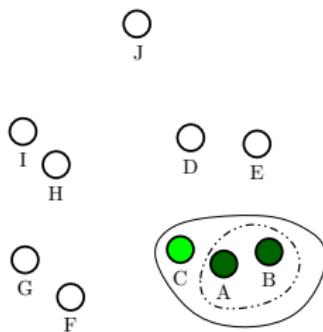
Agglomerative Clustering

Generic Agglomerative Clustering

$$\mathcal{P} = \{A, B, C, D, E, F, G, H, I\}$$

Initialize with each point as a cluster in \mathcal{C}
while stopping criterion is not met do

Choose the best pair of clusters $(C_i, C_j) \in \binom{\mathcal{C}}{2}$
 $C \leftarrow \text{MERGE}(C_i, C_j)$
 $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_i, C_j\} \cup \{C_m\}$



Agglomerative Clustering

Generic Agglomerative Clustering

$$\mathcal{P} = \{A, B, C, D, E, F, G, H, I, J\}$$

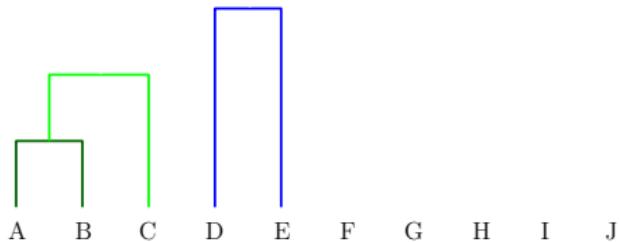
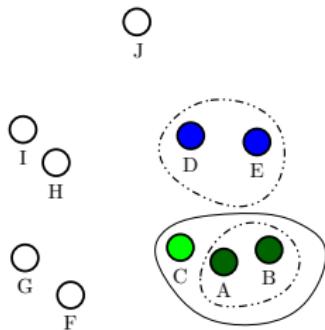
Initialize with each point as a cluster in \mathcal{C}

while stopping criterion is not met **do**

 Choose the best pair of clusters $(C_i, C_j) \in \binom{\mathcal{C}}{2}$

$C \leftarrow \text{MERGE}(C_i, C_j)$

$\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_i, C_j\} \cup \{C_m\}$



Agglomerative Clustering

Generic Agglomerative Clustering

$$\mathcal{P} = \{A, B, C, D, E, F, G, H, I\}$$

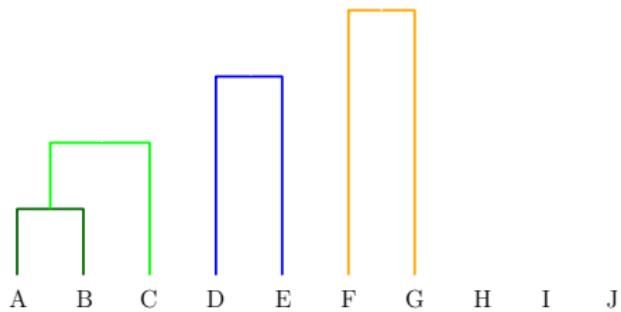
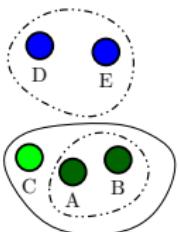
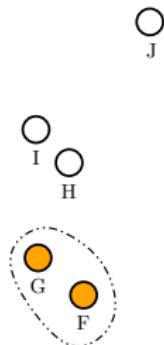
Initialize with each point as a cluster in \mathcal{C}

while stopping criterion is not met **do**

 Choose the best pair of clusters $(C_i, C_j) \in \binom{\mathcal{C}}{2}$

$C \leftarrow \text{MERGE}(C_i, C_j)$

$\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_i, C_j\} \cup \{C_m\}$



Agglomerative Clustering

Generic Agglomerative Clustering

$$\mathcal{P} = \{A, B, C, D, E, F, G, H, I\}$$

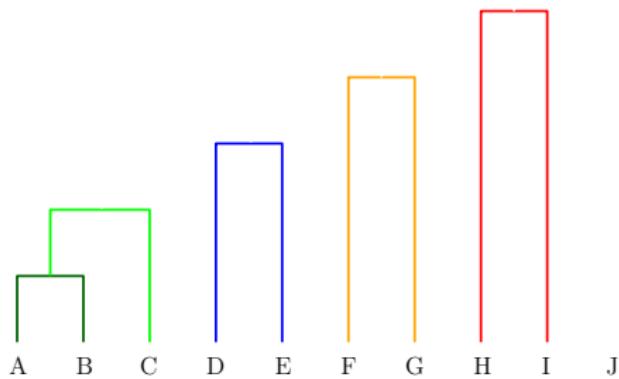
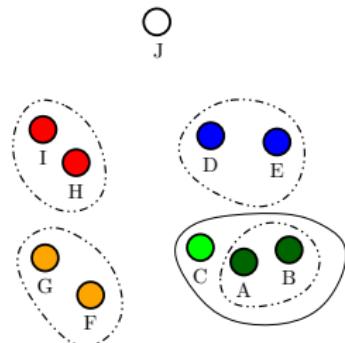
Initialize with each point as a cluster in \mathcal{C}

while stopping criterion is not met **do**

 Choose the best pair of clusters $(C_i, C_j) \in \binom{\mathcal{C}}{2}$

$C \leftarrow \text{MERGE}(C_i, C_j)$

$\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_i, C_j\} \cup \{C_m\}$



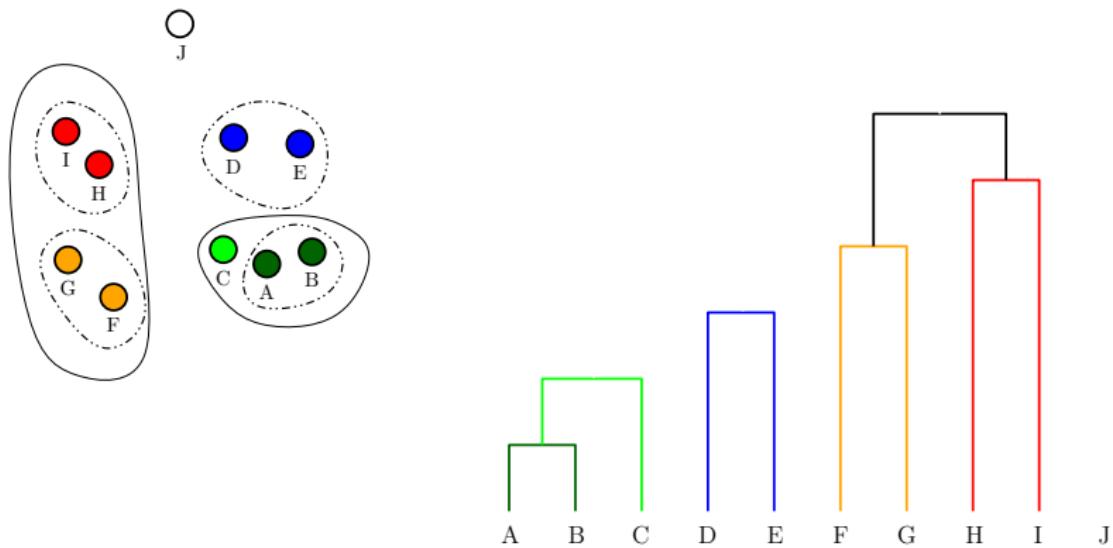
Agglomerative Clustering

Generic Agglomerative Clustering

$$\mathcal{P} = \{A, B, C, D, E, F, G, H, I\}$$

Initialize with each point as a cluster in \mathcal{C}
while stopping criterion is not met do

Choose the best pair of clusters $(C_i, C_j) \in \binom{\mathcal{C}}{2}$
 $C \leftarrow \text{MERGE}(C_i, C_j)$
 $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_i, C_j\} \cup \{C_m\}$



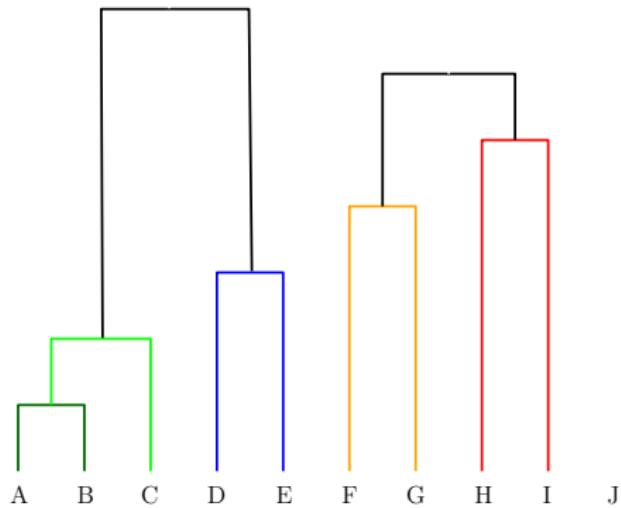
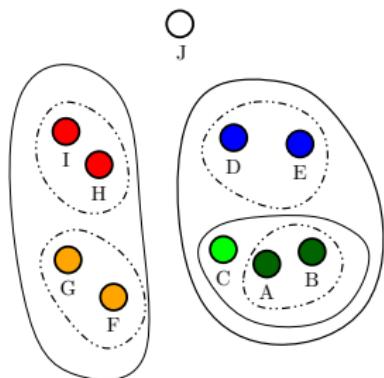
Agglomerative Clustering

Generic Agglomerative Clustering

$$\mathcal{P} = \{A, B, C, D, E, F, G, H, I\}$$

Initialize with each point as a cluster in \mathcal{C}
while stopping criterion is not met do

Choose the best pair of clusters $(C_i, C_j) \in \binom{\mathcal{C}}{2}$
 $C \leftarrow \text{MERGE}(C_i, C_j)$
 $C \leftarrow C \setminus \{C_i, C_j\} \cup \{C_m\}$



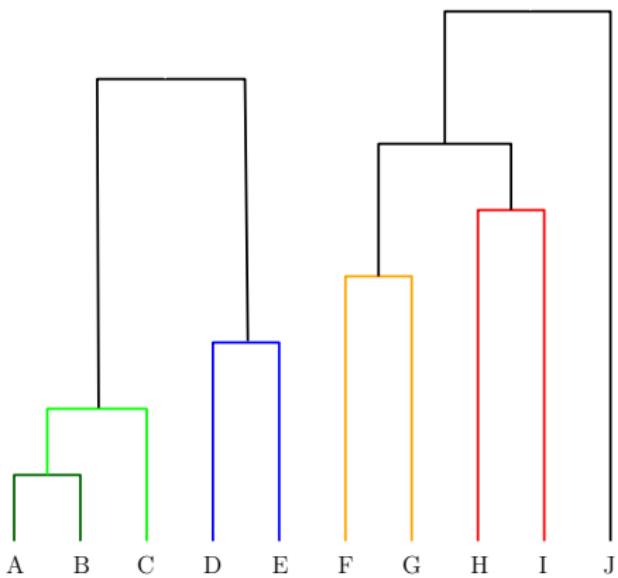
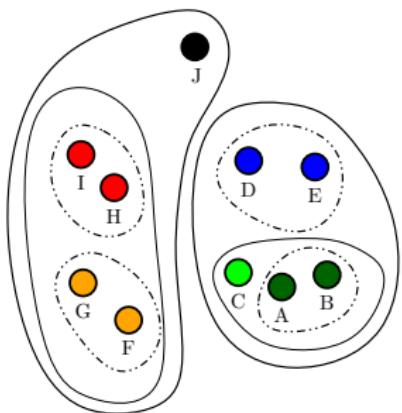
Agglomerative Clustering

Generic Agglomerative Clustering

$$\mathcal{P} = \{A, B, C, D, E, F, G, H, I\}$$

Initialize with each point as a cluster in \mathcal{C}
while stopping criterion is not met do

Choose the best pair of clusters $(C_i, C_j) \in \binom{\mathcal{C}}{2}$
 $C \leftarrow \text{MERGE}(C_i, C_j)$
 $C \leftarrow \mathcal{C} \setminus \{C_i, C_j\} \cup \{C_m\}$



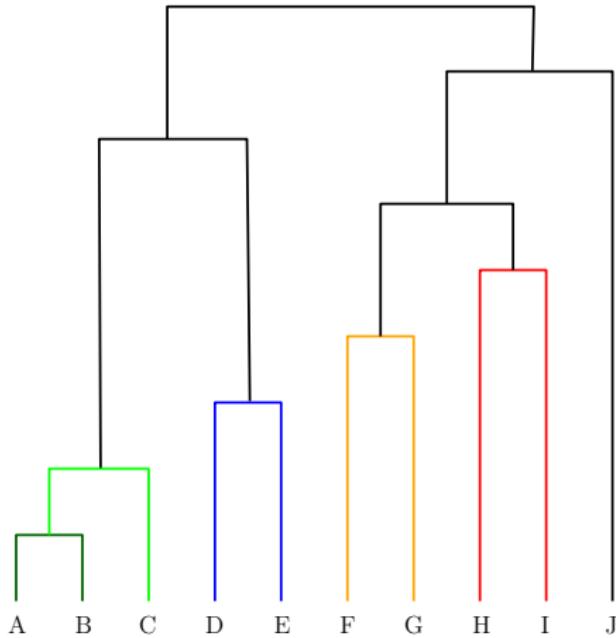
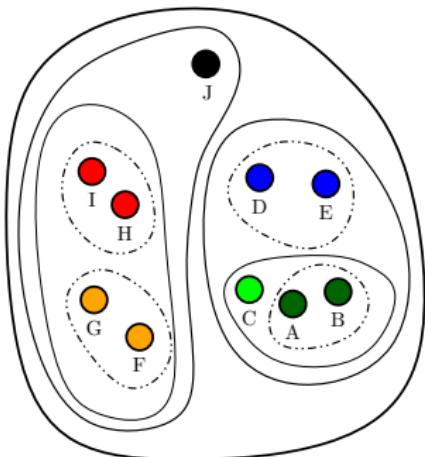
Agglomerative Clustering

Generic Agglomerative Clustering

$$\mathcal{P} = \{A, B, C, D, E, F, G, H, I\}$$

Initialize with each point as a cluster in \mathcal{C}
while stopping criterion is not met do

Choose the best pair of clusters $(C_i, C_j) \in \binom{\mathcal{C}}{2}$
 $C \leftarrow \text{MERGE}(C_i, C_j)$
 $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_i, C_j\} \cup \{C_m\}$



Hierarchical Clustering

Hierarchical Clustering (both divisive and agglomerative)

- is rigid in nature
- once a cluster is made, it cannot be undone
- less chances of improvement
- generally less computational cost
- does not require specific number of clusters
- Can stop when clustering is good enough (stopping criterion)

Agglomerative Clustering

Agglomerative Clustering

- Initially each point is a cluster itself
- In every step two '*close by*' clusters are merged into one
- Generates a bottom-up hierarchy of clusters

Key considerations:

- Representation of clusters
- Distance between clusters
- The choice of pairs of clusters to be merged
- A stopping criterion

Agglomerative Clustering: Euclidean Space

Let $\mathcal{P} = \{x_1, \dots, x_N\}$, each $x_i \in \mathbb{R}^n$

- **Represent** each cluster by its centroid
- **Distance** between two clusters is the distance between their centroids
- **Select** a pair of clusters with minimum (inter-centroid) distance
- **Stop** when the number of clusters is equal to k

Note that this representation requires explicit feature vectors for points

Cluster: Diameter, Radius and Density

- **Diameter** of a cluster C , $dia(C)$ is the max inter-point distance in C

$$dia(C) = \max_{x,y \in C} \{d(x,y)\}$$

- **Radius** of a cluster C with centroid \mathbf{c} , $rad(C)$ is the maximum distance of a point in C from the centroid \mathbf{c}

$$rad(C) = \max_{x \in C} \{d(x, \mathbf{c})\}$$

- **Density** of a cluster C : is mass (number of points) over **volume**
 - What is volume (shape) of the cluster? Use an estimate

$$den(C) = \frac{|C|}{dia(C)^2} \quad \text{or} \quad den(C) = \frac{|C|}{rad(C)^2} \quad \text{or} \quad den(C) = \frac{|C|}{rad(C)^n}$$

The exponent in denominator is usually 1, 2, or n (dimensionality of points)

Agglomerative Clustering: Stopping Criteria

Stop based on '*quality*' of recently merged cluster, e.g. when the

- average inter-point distance of the merged cluster is above a threshold
- diameter of the merged cluster is above a threshold
- radius of the merged cluster is above a threshold
- the average distance from the centroid is above a threshold
- sum of squared distances from the centroid is above a threshold

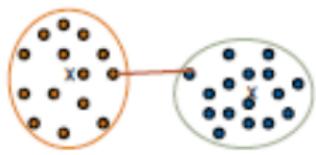
Stop based on a global '*quality*' measure. e.g. when

- the average diameter of all clusters increases above a threshold
 - The idea is as long as we merge cluster that truly should be merged, the average diameter will not significantly increase
 - When we merge a pair that should not be merged, there would be a sudden jump in the average diameter

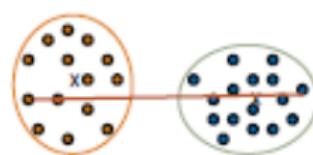
Cluster: Distance Measures between Clusters

Distance Between two cluster C_i and C_j can be defined as

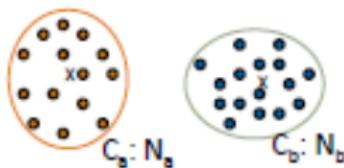
- **Centroid Link:** distance between centroids of C_i and C_j
 - Requires points as numeric vectors
- **Single Link:** Minimum inter-point distance between C_i and C_j
- **Average Link:** Average inter-point distance between C_i and C_j
- **Complete Link:** Maximum Inter-point distance between C_i and C_j



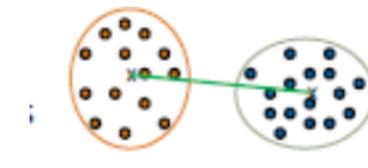
Single Link



Complete Link



Average Link



Centroid Link

Agglomerative Clustering: Pair Selection Criteria

Pair selection can be based on any distance measures between two clusters

e.g. select a pair of clusters with minimum

- Centroid link
- Single Link
- Average Link
- Complete Link

Can also select pair based on '*quality*' of the resulting (merged) cluster

e.g. choose a pair

- resulting in the lowest radius of a merged cluster
- resulting in the lowest diameter of a merged cluster

Agglomerative Clustering: Cluster Representation

We cannot compute centroids if

- points are not real vectors (non-Euclidean space)
- points are only abstractly described (no explicit vectors) and only distance matrix is provided

We can represent clusters by a central element. Any definition of clusteroid of a cluster C can be used, e.g. a point in C

- with minimum sum (average) of distances to other points in C
- with minimum largest distance to a point in C
- with minimum sum of squared distances to other points in C

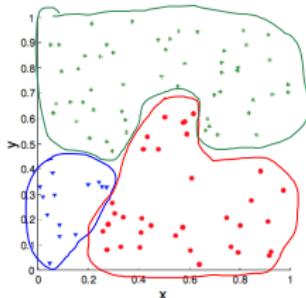
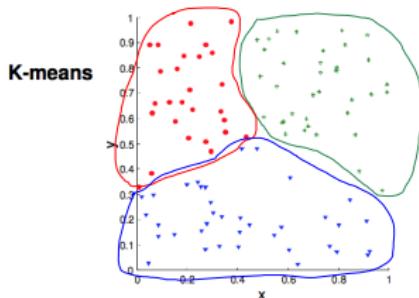
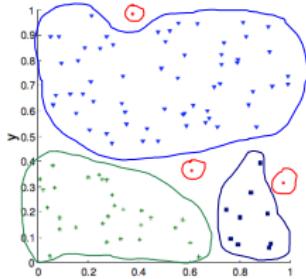
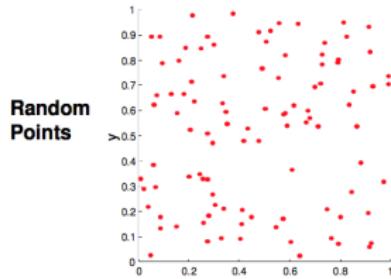
Notions of inter-cluster distances, pair selection and stopping rules can be adapted to this version of problem (replace clusteroid for centroid if needed)

Clustering: Evaluation and Validation

Goals and Aspects of Clustering Quality Assessment

1 Determine cluster tendency of dataset

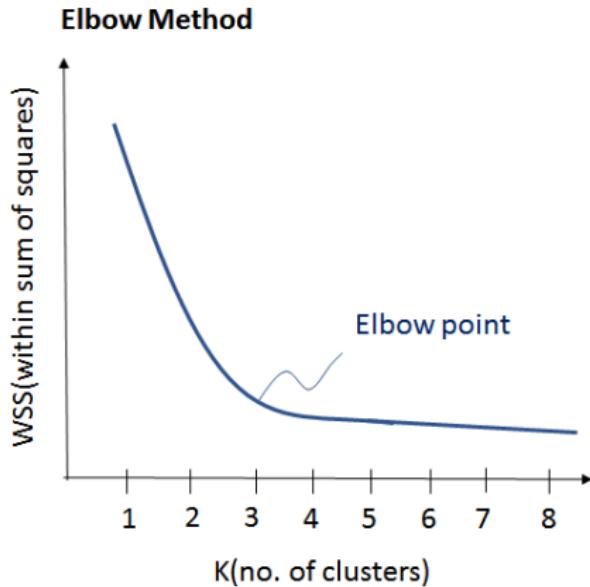
- Are there meaningful groups (non-random structure) in the data
- or clusters represent some patterns in noise



Clustering: Evaluation and Validation

Goals and Aspects of Clustering Quality Assessment

- 1 Determine cluster tendency of dataset
- 2 Find the correct number of clusters
 - Recall the elbow method



Clustering: Evaluation and Validation

Goals and Aspects of Clustering Quality Assessment

- 1 Determine cluster tendency of dataset
- 2 Find the correct number of clusters
- 3 Evaluate Clustering Quality
 - Validate the output clustering by comparing with known results (class labels or manual clustering by experts)
 - Evaluate how well output clustering fit the data without reference to external results
- 4 Compare two clustering algorithms
 - Observe the kind of patterns each try to mine and determine which algorithm is suitable for the task at hand
- 5 Compare two clusters in a clustering

Clustering: Evaluation and Validation

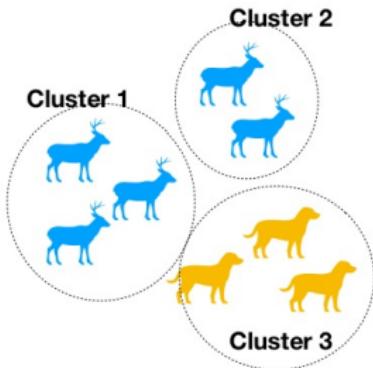
- Clustering is an unsupervised task (cannot use ground truth in the clustering algorithm)
- Cluster Validation against Class Labels: If true class labels are available, we can see how well clusters match with class labels
- Cluster Evaluation with No Class Labels: Assess cluster quality w.r.t proximity measure

Validation of Clustering

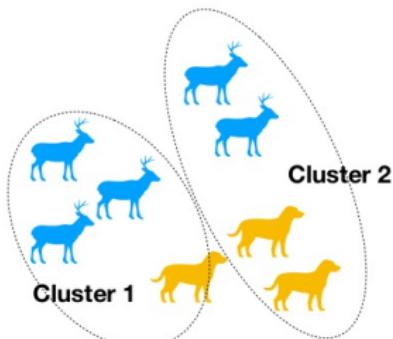
Two basic criteria for **validating clusterings** are:

- **Cluster homogeneity**

- Clusters should contain objects of a single class only
- Such clusters are called **pure**, the purer the clusters the better
- Singleton clusters are the most homogeneous



Good

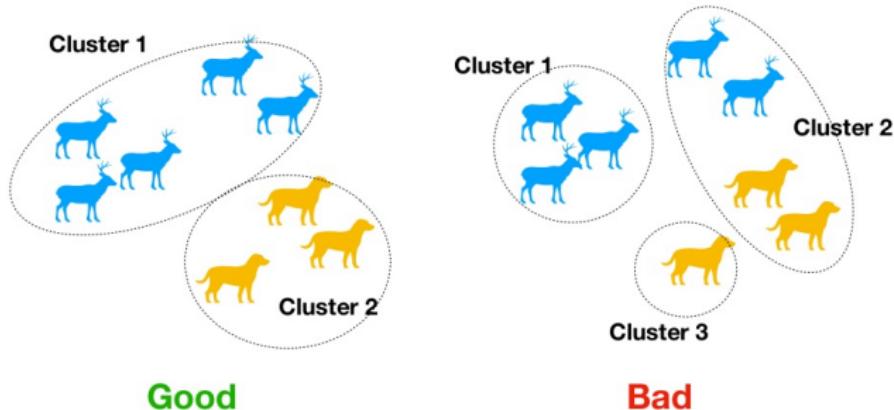


Bad

Validation of Clustering

Two basic criteria for validating clusterings are:

- Cluster homogeneity
- Cluster Completeness
 - Objects in the same class should be contained in a single cluster
 - Classes should not be split into multiple clusters
 - Singleton clusters may not be complete

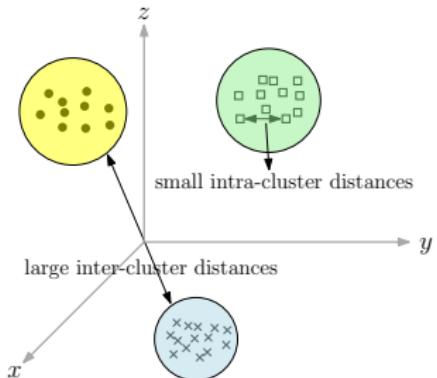


Evaluation of Clustering

Three basic criteria for **evaluating clusterings** are:

■ Cluster Compactness

- Objects in a cluster should be highly similar, Intra-Cluster-Low distances
- generally desired in classification type tasks such as image recognition
- Proximity measure should be meaningful (similarity \sim homogeneity)
- Also called **cluster cohesion**



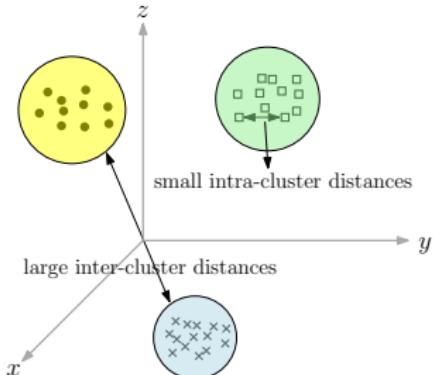
Evaluation of Clustering

Three basic criteria for **evaluating clusterings** are:

- Cluster Compactness

- Cluster Separation

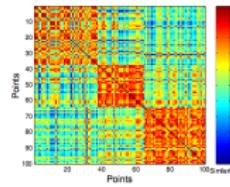
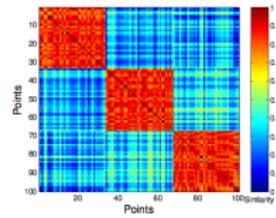
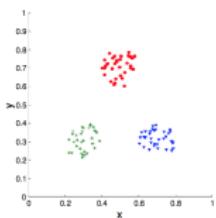
- Clusters should be well-separated (far apart)
- Objects in two different clusters should be highly dissimilar
- Inter-cluster high distances
- Again proximity measure should be meaningful



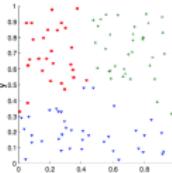
Evaluation of Clustering

Three basic criteria for evaluating clusterings are:

- Cluster Compactness
 - Cluster Separation
 - Agreement of pairwise proximity with clustering-induced metric
 - Clustering should respect the pairwise proximity measure
 - Highly similar objects should be more likely to go to the same cluster and distant pairs should be separated
 - Somewhat encompasses both compactness and separation
-
- Order the similarity matrix with respect to cluster labels and inspect visually.
 - Clusters in random data are not so crisp



K-means



External and Internal Measures of Clustering Quality

Numerical measures for clustering validation and evaluation

- **External or Extrinsic Measures**

- They use class labels
- Some are indexes to measure for a specific criterion
- Different Indexes on a common scale can be combined to measure for combination of criteria

- **Internal or Intrinsic Measures**

- They do not use class labels
- Some are indexes to measure for a specific criterion
- Different Indexes on a common scale can be combined to measure for combination of criteria
- Generally a statistical significance of index values needs to be ascertained

External Measures

Any measures for assessment of quality of classification can be used

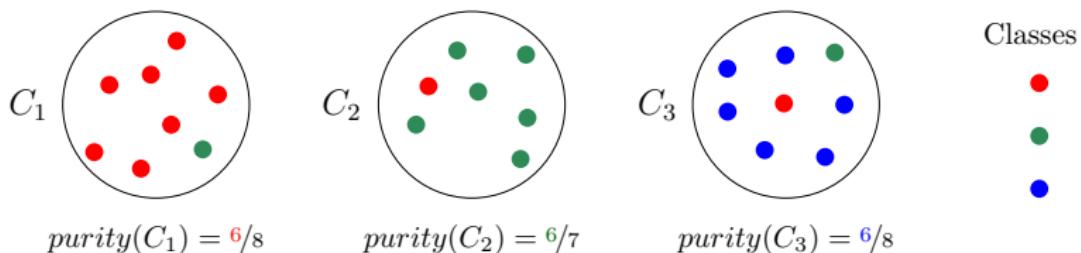
- Accuracy
- Error
- Precision, Recall, F1 measure
- Purity
- Entropy
- Conditional Entropy
- Normalized Mutual Information (NMI)
- Maximum Matching: Match clusters to class and see goodness of matching

External Measures: Purity

- Let $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$, L classes
- Let n_{ij} be objects of class i in cluster C_j
- $p_{ij} = n_{ij}/|C_j|$ (class distribution in clusters)

$$\text{purity}(C_j) = \frac{\max_{1 \leq i \leq L} n_{ij}}{|C_j|}$$

Purity of C_j : ratio of dominant class in C_j to $|C_j|$



Purity of clustering \mathcal{C} :

$$\sum_{j=1}^k \frac{|C_j|}{N} \text{purity}(C_j)$$

- $\text{purity}(\mathcal{C}) = 8/23 \times 7/8 + 7/23 \times 6/7 + 8/23 \times 6/8 = 19/23$

External Measures: Purity

- $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$, L classes
- n_{ij} : objects of class i in cluster C_j
- $p_{ij} = n_{ij}/|C_j|$ (class distribution in clusters)

$$purity(C_j) = \frac{\max_{1 \leq i \leq L} n_{ij}}{|C_j|}$$

Purity of C_j : ratio of dominant class in C_j to $|C_j|$

$$\text{Purity of clustering } \mathcal{C}: \sum_{j=1}^k \frac{|C_j|}{N} purity(C_j)$$

- Purity favors homogeneity only
- Highest purity is 1 when clusters are the purest
- Singleton clusters maximize purity
- Ignores cluster completeness

External Measure: Conditional Entropy

- Let $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ ▷ Clustering into k clusters
- $\mathcal{T} = \{T_1, T_2, \dots, T_L\}$ ▷ True partition into L classes
- Let n_{ij} : objects of class T_i in cluster C_j
- $p_{ij} = n_{ij}/|C_j|$ (class (probability) distribution in C_j)

conditional entropy of \mathcal{T} w.r.t cluster C_j : entropy of class distrib. in C_j

$$H(\mathcal{T}|C_j) = - \sum_{i=1}^L p_{ij} \log_2 p_{ij}$$

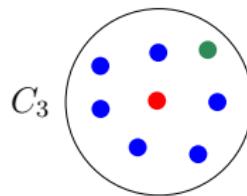
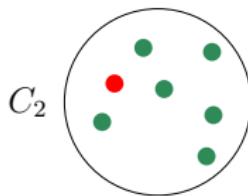
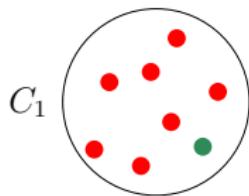
conditional entropy of \mathcal{T} w.r.t clustering \mathcal{C} :

$$H(\mathcal{T}|\mathcal{C}) = - \sum_{j=1}^k \frac{|C_j|}{N} H(\mathcal{T}|C_j) = - \sum_{j=1}^k \frac{|C_j|}{N} \sum_{i=1}^L p_{ij} \log_2 p_{ij}$$

External Measure: Conditional Entropy

- Conditional entropy favors completeness
- Split classes results in higher entropy
- For perfectly complete clusters conditional entropy is 0
- Highest value is $\log L$

Compute values of conditional entropies in this example



$$purity(C_1) = 6/8$$

$$purity(C_2) = 6/7$$

$$purity(C_3) = 6/8$$

External Measure: Rand Index

- Let $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ ▷ Clustering into k clusters
- $\mathcal{T} = \{T_1, T_2, \dots, T_L\}$ ▷ True partition into L classes
- Measures of pairwise agreement of class labels and clustering parts

Clustering		
Number of pairs $\binom{N}{2}$	Same cluster	Different clusters
True classes	True Positive	False Positive
Different classes	False Negative	True Negative

$$\text{Rand Index of } \mathcal{C}: \quad RI(\mathcal{C}) := \frac{TP + TN}{\binom{N}{2}}$$

Compare with standard precision and recall

$$\text{Jaccard Index of } \mathcal{C}: \quad J(\mathcal{C}) := \frac{TP}{TP + FN + FP}$$

Internal Measure: SSE

A measure of compactness of a cluster or clustering is SSE or average SSE

For a k -partition $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ sum of squared error (SSE) or within sum of squared error (WSSE or WSS) is defined as

Let $\mathbf{c}_i = \text{centroid}(C_i)$, then

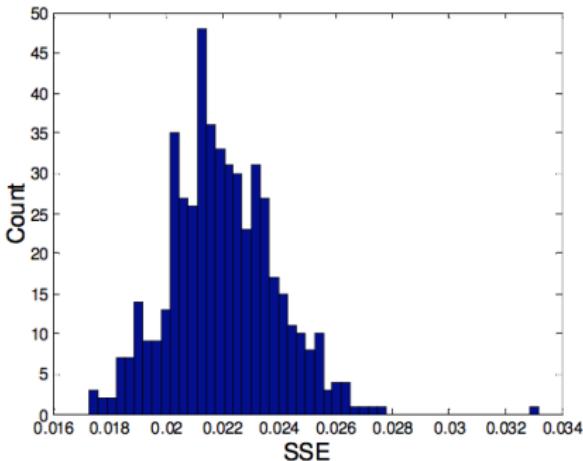
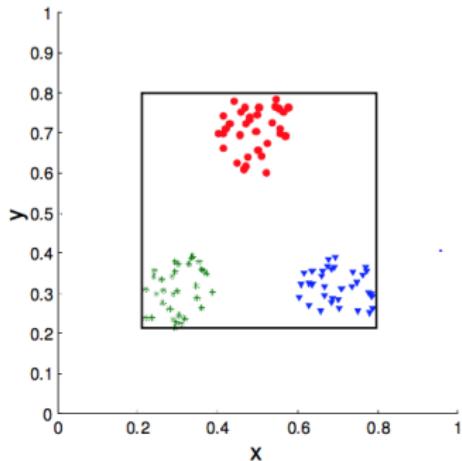
$$SSE(\mathcal{C}) = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mathbf{c}_i\|^2$$

Internal Measure: Statistical Significance

- For SSE and other internal measures need to see whether the values are meaningful or statistically significant (deep statistical theories)
- A rough idea (rule of thumb is) to see if e.g. the obtained SSE for a clustering of N points in a certain space into k clusters is good
- Generate random datasets of N points in the same space (same ranges and dimensions) and then cluster them into k clusters using the same algorithm
- Observe the distribution of the SSE of these trials (say get the mean and st-dev)
- If the mean SSE of these random points is significantly higher than our SSE, then SSE is significant (and clustering is good)

Internal Measure: Statistical Significance

- Compare SSE of 0.005 against three clusters in random data
- Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values



Internal Measure

- Some other internal measures for cluster (or clustering) compactness
 - (weighted) average of intra-cluster pairwise distances
 - Correlation between proximity matrix and cluster co-belonging matrix (see below)
- Smaller values of an index means clusters are compact

Internal Measure

- Clustering Separation can be measured by
 - (weighted) average of inter-cluster pairwise distances
 - the Between Sum of Squares (BSS) given by

Let $\mathbf{c}_i = \text{centroid}(C_i)$ and let $\mathbf{c} = \text{centroid}(\mathcal{P})$ (centroid of the whole dataset), then

$$BSS(\mathcal{C}) = \sum_{i=1}^k |C_i| (\mathbf{c} - \mathbf{c}_i)^2.$$

- Larger values of an index means clusters are well-separated

Internal Measure

Silhouette Coefficient incorporates both cohesion and separation

Let $\mathcal{C} = \{C_1, \dots, C_k\}$. For a point x in C_i

- $a(x) = \frac{1}{|C_i|} \sum_{x \neq y \in C_i} d(x, y)$ ▷ mean distance from x in its cluster
- $b(x) = \min_{j \neq i} \frac{1}{|C_j|} \sum_{y \in C_j} d(x, y)$ ▷ mean distance from x in closest cluster

Silhouette Coefficient of $x \in \mathcal{P}$: $s(x) := \frac{b(x) - a(x)}{\max\{a(x), b(x)\}}$

Silhouette Coefficient of \mathcal{C} is the mean value of $s(x)$ over points in \mathcal{P}

$$SC(\mathcal{C}) := \frac{1}{N} \sum_{x \in \mathcal{P}} s(x)$$

- $SC(\mathcal{C}) \in [-1, 1]$ ▷ the closer to 1 the better clustering

Internal Measure

Correlation between proximity and clustering induced co-belonging matrices

- Let D be the pairwise proximity matrix
- Let C be the co-belonging matrix induced by clustering
 - A row and column for each point, and $C(i,j) = 1$ or 0 depending on whether $x_i \neq x_j \in \mathcal{P}$ belong to the same cluster
- High correlation between these two symmetric matrices means good clustering and vice-versa (if proximity is similarity)
- For distance matrix low correlation indicates good clustering
- Incorporates both compactness and separation, also measures agreement of the clustering-induced metric and the pairwise proximity measure