

Introduction to Clustering

Machine Learning Unit 19

Sudeshna Sarkar

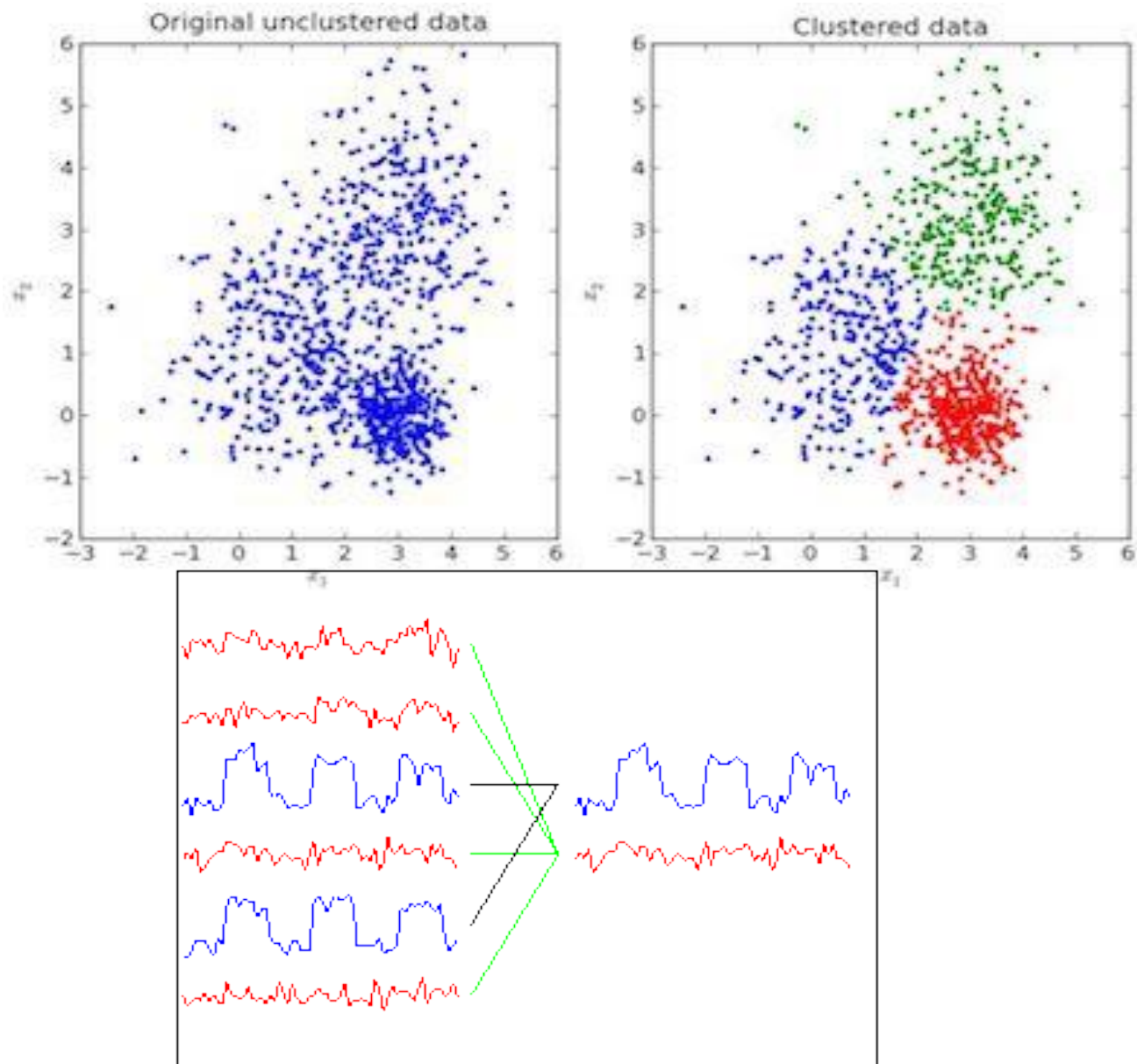
Centre of Excellence in Artificial Intelligence

Indian Institute of Technology Kharagpur

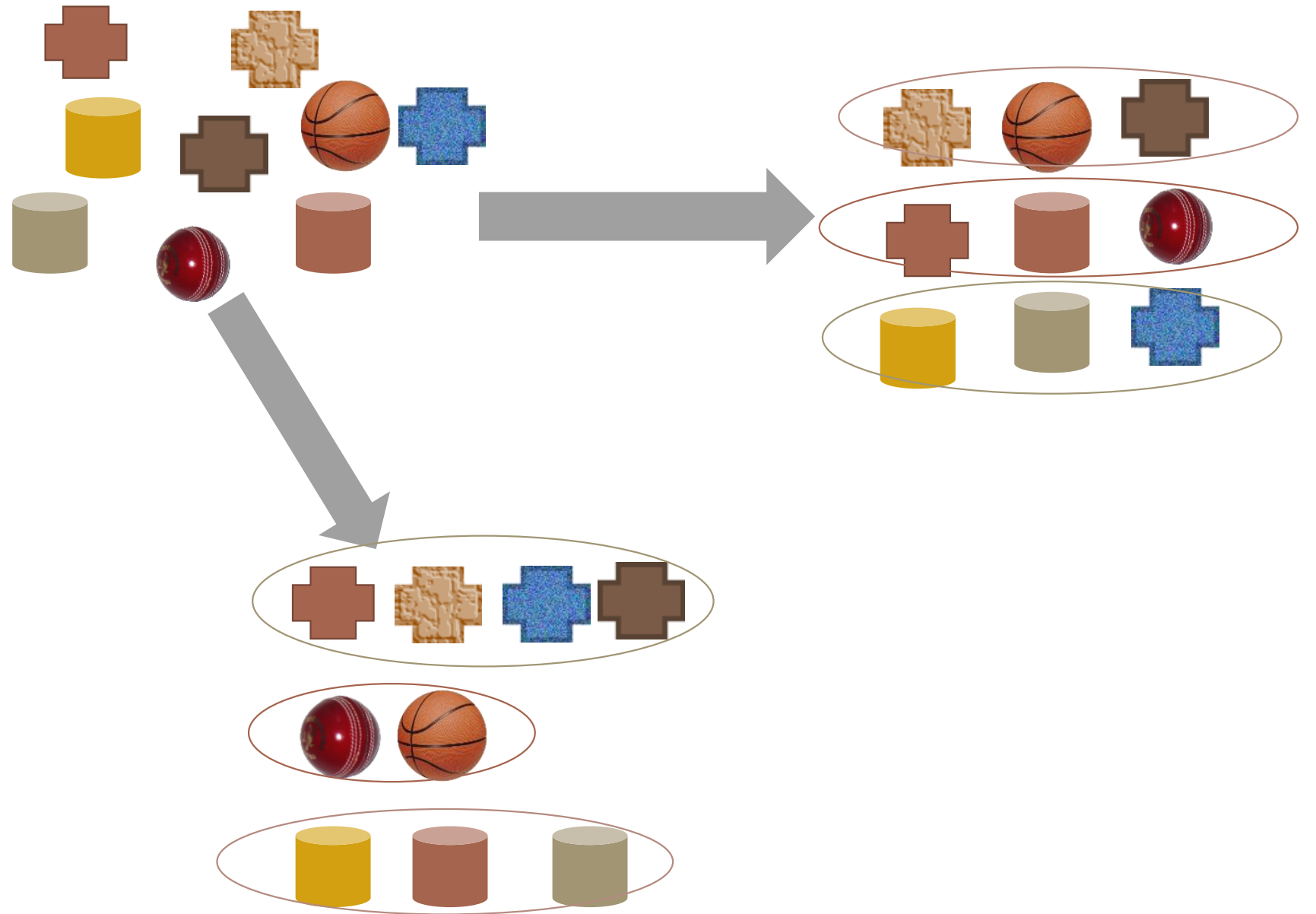
Applications

- Segmenting of customers with similar market characteristics — pricing , loyalty, spending behaviors etc.
- Grouping of products based on their properties
- Identify similar energy use customer profiles
 - <x> = time series of energy usage
- Clustering weblog data to discover groups of similar access patterns.
- Recognize communities in social networks.
- Top 20 topics in Twitter

Clustering




Clustering



Task

Recipe Recommendation

- The repository contains many recipes (ingredient, steps, picture)
- Data: Browsing history (user, recipe)
- Recommend new recipes to user
- Group similar recipes into clusters
- For the test user
 - Identify the clusters from which he has browsed the most recipes
 - Show him the popular recipes from that cluster



Kancha Aam diye Mangsho (Lamb with Raw Mango)


Kancha Aam diye Mangsho. Lamb cooked over a fatigued flame with potatoes and potol. The perfume and subtle tart of green mangoes adds a magical twist to this otherwise ubiquitous Mangshor Jhol. Another classic from the house of the Tagores. Wicked !!!

Course	Main course
Cuisine	Bengali
Prep Time	5 minutes
Cook Time	70 minutes
Servings	

INGREDIENTS

- 500 g mutton shoulder curry cut pieces
- 1/4 cup raw green mango
- 4-5 potol or parwal or pointed gourds peeled and cut into two roundels
- 2 potatoes peeled and cut into quarters
- 1/3 cup yoghurt
- 1/3 cup onion paste
- 2 tsp ginger paste
- 3-4 dry red chillies
- 1 tsp turmeric powder
- 2 tsp red chilli powder
- 1/4 cup ghee
- 1.5 tbsp cooking oil
- salt to taste

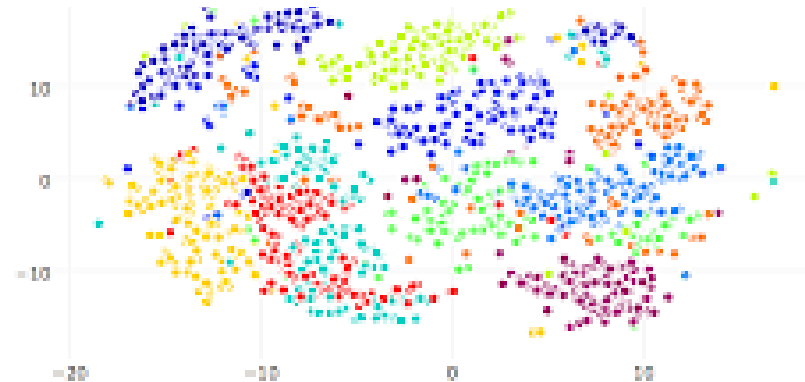
INSTRUCTIONS



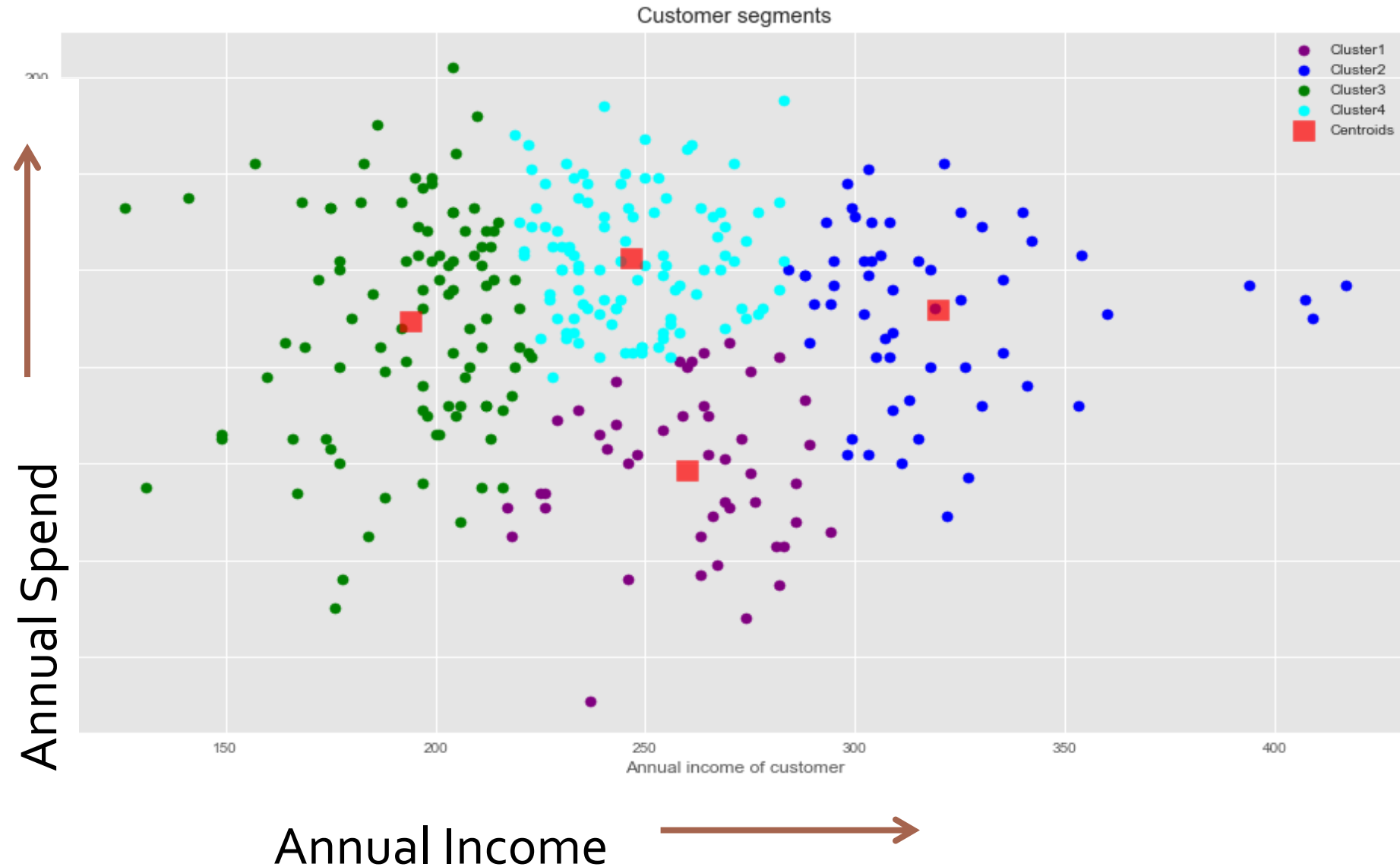
Customer Segmentation

- Group customers based on their demography and activity
 - Purchase History
 - Demographic
 - Content engagement
 - Behaviour
 - Customer Lifecycle Stage

Cater to customer groups for promotion, recommendation, product development strategies



INCOME	SPEND
233	150
250	187
204	172
236	178
354	163
192	148
294	153
263	173
199	162
168	174
239	160
275	139
266	171
211	144



Applications: News Clustering

Top Stories

- Donald Trump
- Istanbul Atatürk Airport
- Nigel Farage
- Subramanian Swamy
- Salman Khan
- European Union
- NASA
- Toyota
- Telangana
- Helium
- Kharagpur, West Bengal
- India
- World
- Business
- Technology
- Entertainment
- Sports
- Science
- Health
- More Top Stories

Top Stories

LIVE :Airport partially reopened after blasts that killed 36

The Hindu - 1 hour ago

Two explosions rocked Istanbul's Ataturk airport, killing 36 people and wounding 147, Turkey's justice minister Bekir Bozdag said on Tuesday.

Istanbul attack: Hrithik trolled for 'took economy' tweet; B-Town mourns loss of lives India Today

Istanbul airport attack: Suicide blasts kill 36, 147 injured; PM blames IS Financial Express

Local Source: At least 36 killed in terror attack on Istanbul's Atatürk Airport Hurriyet Daily News

Opinion: Istanbul airport attack: Turkey's vengeance will be like 'rain from hell' CNN

Wikipedia: Istanbul Atatürk Airport

See realtime coverage

Related: Istanbul Atatürk Airport » Istanbul »

India Today Zee News The Guardian

Bonanza for government staff: Cabinet approves 23.6% overall pay hike

Times of India - 1 hour ago

NEW DELHI: The Cabinet on Wednesday approved a 23.6 per cent increase in government employees' overall pay - basic pay plus allowances - as recommended by the 7th Pay Commission.

Petition Against Ban On Gay Sex To Be Heard By Chief Justice Of India

NDTV - 1 hour ago

India's gay community has been fighting to get a ban on homosexual sex overturned ever since the Supreme Court reinstated a decades-old law in late 2013.

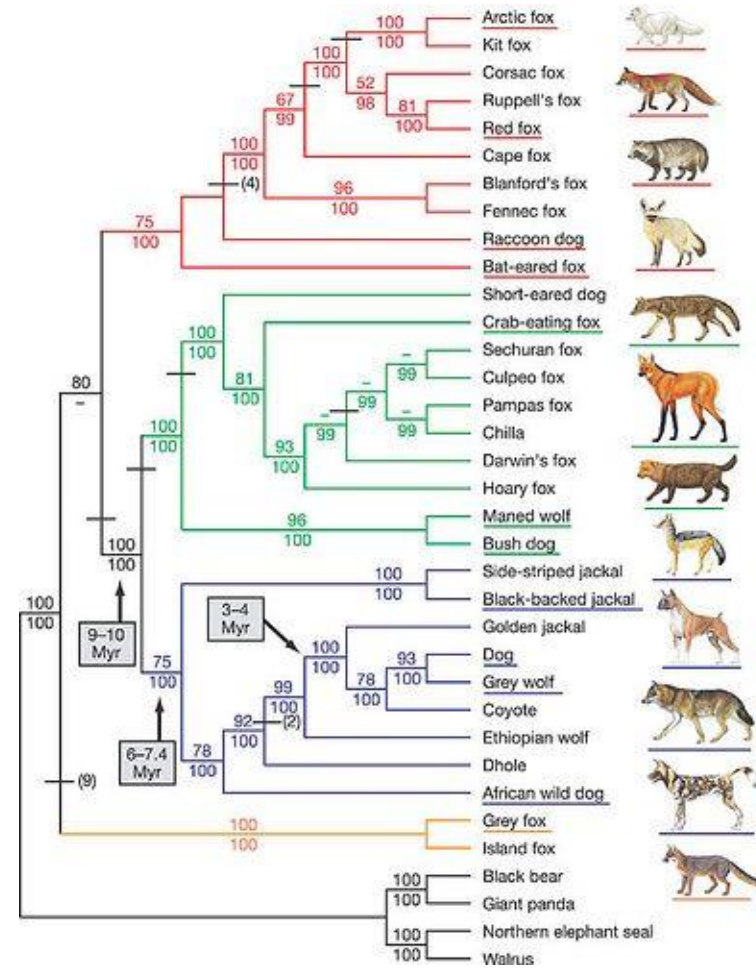
Rajan may have stayed had govt promptly reacted: RBI Governor's parents

Business Standard - 3 hours ago

Almost two weeks after RBI Governor Raghuram Rajan has announced his intention to return to academia after completion of his term in September, controversy surrounding his second term does not seem to be waning.

Monsoon session from July 18, Govt confident of GST passage

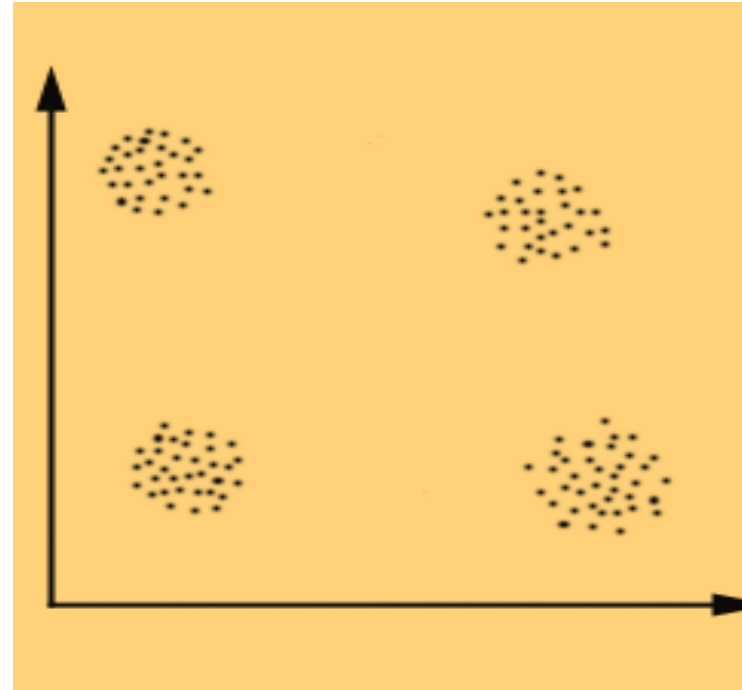
Clustering



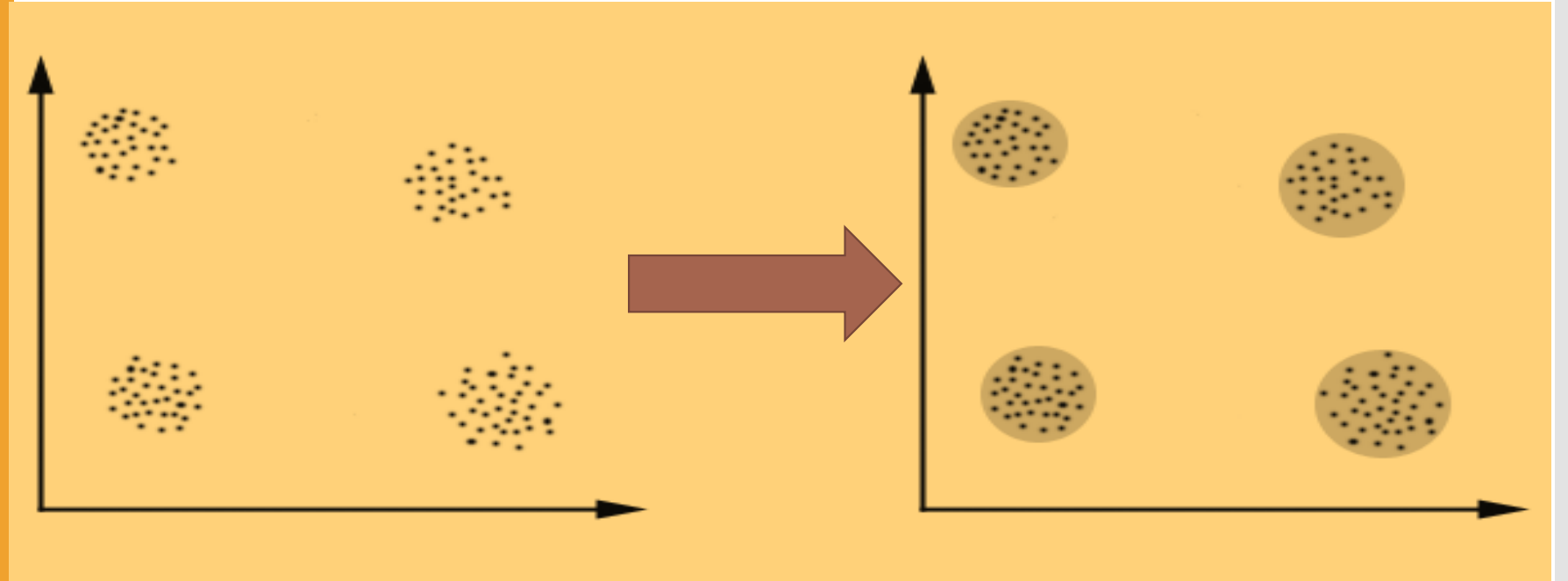
Clustering

- Similarity or distance measures:
 - Euclidean
 - Manhattan distance
 - Cosine similarity
 - Pearson correlation
 - etc.

How many
clusters?

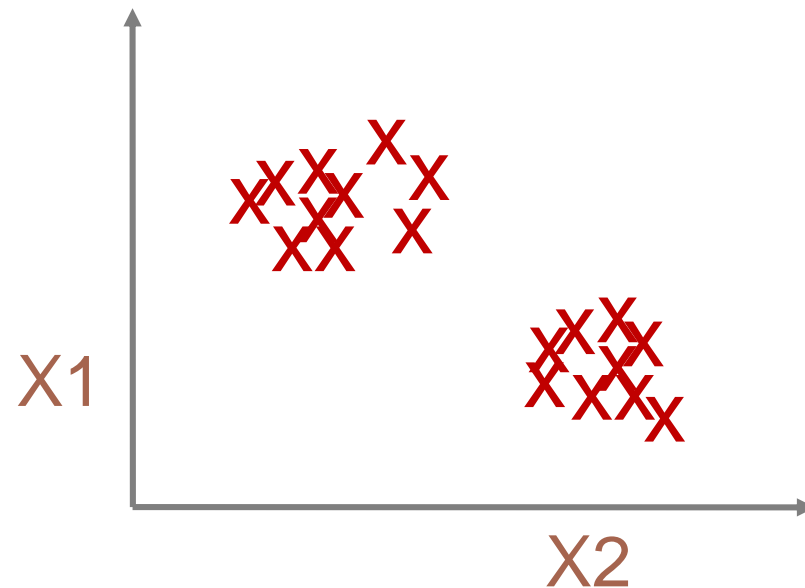


An illustration



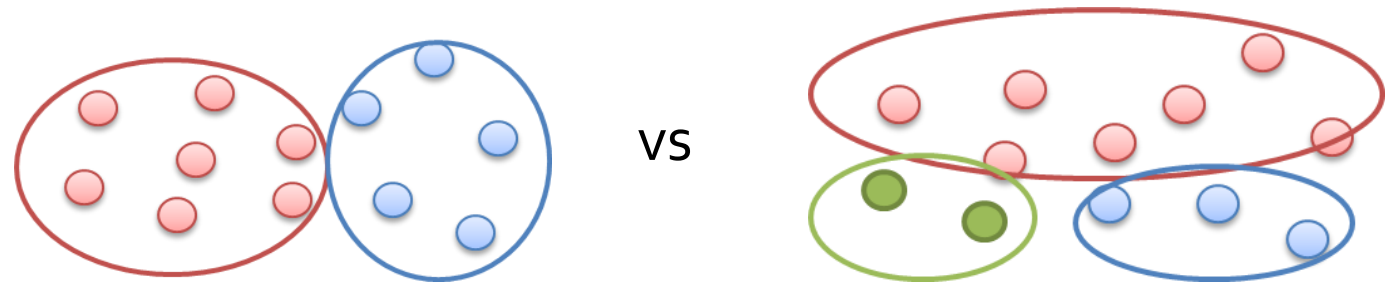
Clustering

- Given examples: $\{X_1, X_2, \dots, X_m\}$, $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$
Find a *natural* grouping of the data such that
 - Inter-cluster **similarity** is maximized
 - Intra-cluster **similarity** is minimized



Aspects of clustering

1. A proximity measure
 - Similarity measure $s(x_i, x_j)$: large if x_i and x_j are similar
 - Distance measure $d(x_i, x_j)$: small if x_i and x_j are similar
2. A clustering algorithm
3. Criterion to evaluate clustering quality



Similarity / Distance Measures



DX 10-24mm f/3.5-4.5G ED	DX 12-24mm f/4G ED	14-24mm f/2.8G ED	24mm f/2.8G	16-35mm f/4G VR	DX 17-55mm f/2.8G IF ED	18-35mm f/3.5-4.5G ED	DX 18-300mm f/3.5-6.3G ED VR
Instant Savings: \$100.00	Instant Savings: \$100.00	Instant Savings: \$200.00	Instant Savings: \$50.00	Instant Savings: \$100.00	Instant Savings: \$140.00	Instant Savings: \$100.00	Instant Savings: \$100.00
28mm f/1.8G	28mm f/2.8 ED	35mm f/1.4G	35mm f/1.8G	24-70mm f/2.8G ED	24-85mm f/3.5-4.5G ED VR	28-300mm f/3.5-5.6G ED VR	DX 55-200mm f/4.5-5.6G ED VR II
Instant Savings: \$50.00	Instant Savings: \$40.00	Instant Savings: \$200.00	Instant Savings: \$50.00	Instant Savings: \$200.00	Instant Savings: \$100.00	Instant Savings: \$250.00	Instant Savings: \$200.00
DX Micro 40mm f/2.8G	50mm f/1.4G	58mm f/1.4G	60mm f/2.8G ED Micro AF	70-200mm f/2.8G IF ED VR II	70-200mm f/4G ED VR	80-400mm f/4.5-5.6G ED VR	DX 55-300mm f/4.5-5.6G ED VR
Instant Savings: \$30.00	Instant Savings: \$50.00	Instant Savings: \$200.00	Instant Savings: \$100.00	Instant Savings: \$300.00	Instant Savings: \$100.00	Instant Savings: \$400.00	Instant Savings: \$150.00
85mm f/1.4G Lens	85mm f/1.8G Lens	Micro 105mm f/2.8G IF-ED VR	DX 16-85mm f/3.5-5.6 ED VR	Nikon SB-500 AF Speedlight	TC-20E III Teleconverter		
Instant Savings: \$150.00	Instant Savings: \$50.00	Instant Savings: \$135.00	Instant Savings: \$100.00	Instant Savings: \$20.00	Instant Savings: \$50.00		

Depends on the problem domain and data type

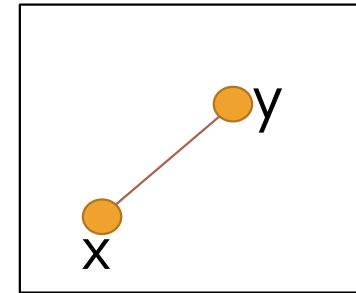
- Customers
- Time series
- Text
- Images

Similarity between items

- Similarity or distance measures:
 - Euclidean
 - Manhattan distance
 - Cosine similarity
 - Pearson correlation
 - ...

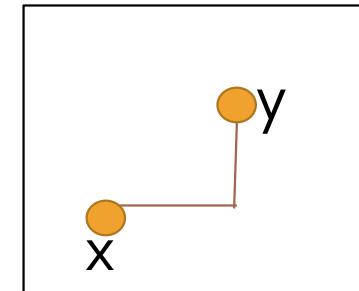
Distance/ Similarity measures

Euclidean distance $d(\mathbf{X}_i, \mathbf{X}_j) =$
$$\sqrt{\sum_{s=1}^d |x_{is} - x_{js}|^2}$$



Manhattan distance

$$d(\mathbf{X}_i, \mathbf{X}_j) = \sum_{s=1}^d |x_{is} - x_{js}|$$



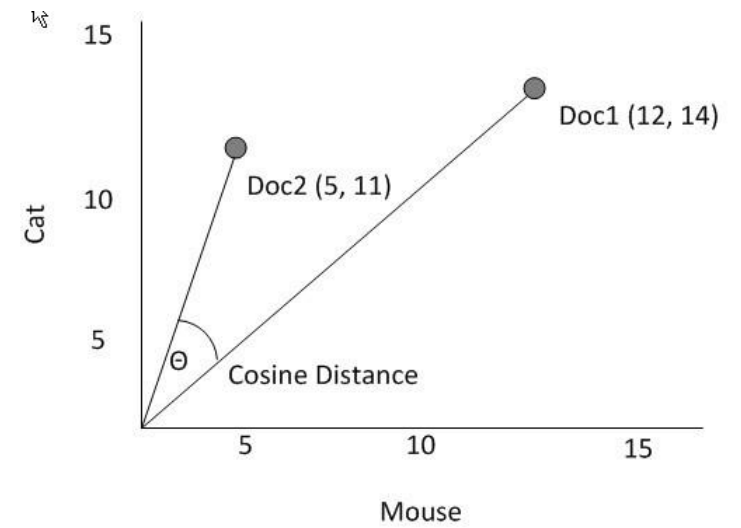
Minkowski family of distance measures

$$d(\mathbf{X}_i, \mathbf{X}_j) = \left(\sum_{s=1}^d |x_{is} - x_{js}|^p \right)^{1/p}$$

Similarity measures

- Text Similarity
- Cosine distance

$$\text{cosine}(X_i, X_j) = \frac{X_i \cdot X_j}{\|X_i\| \cdot \|X_j\|}$$



(Dis)similarity measures

- Correlation coefficients (scale-invariant)
- Mahalanobis distance

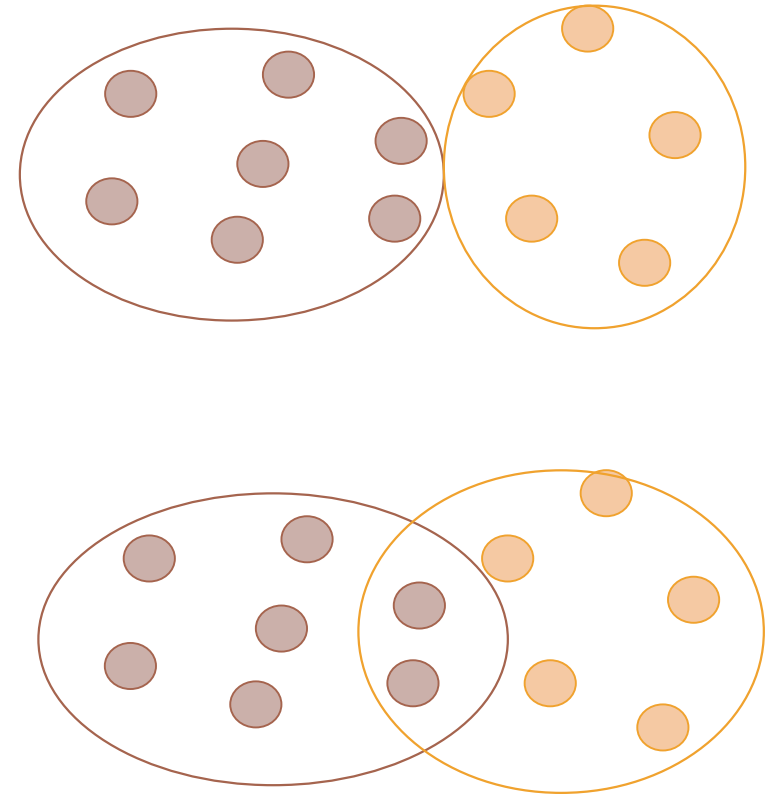
$$d(x_i, x_j) = \sqrt{(x_i - x_j)\Sigma^{-1}(x_i - x_j)}$$

- Pearson correlation

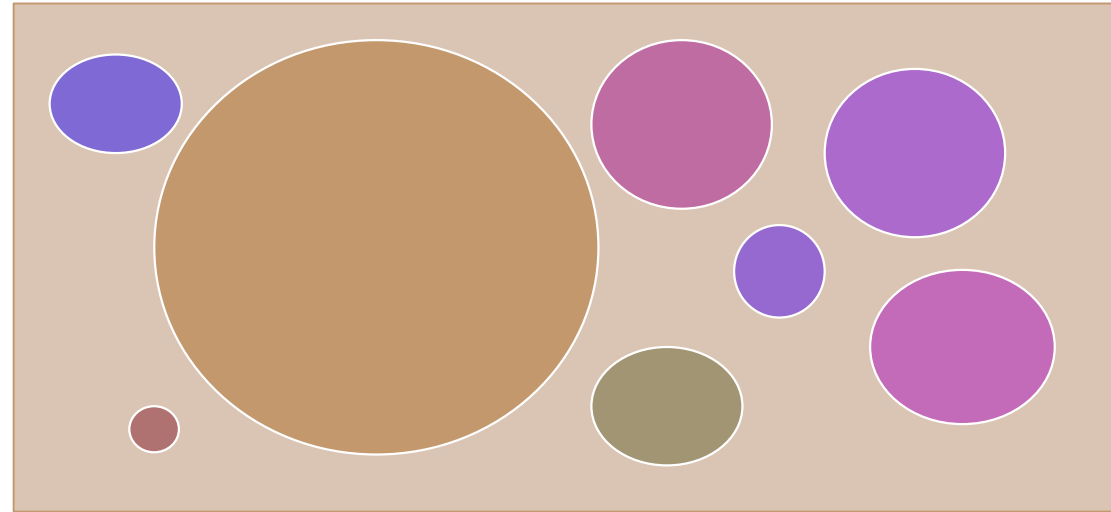
$$r(x_i, x_j) = \frac{\text{Cov}(x_i, x_j)}{\sigma_{x_i}\sigma_{x_j}}$$

Different Types of Clustering

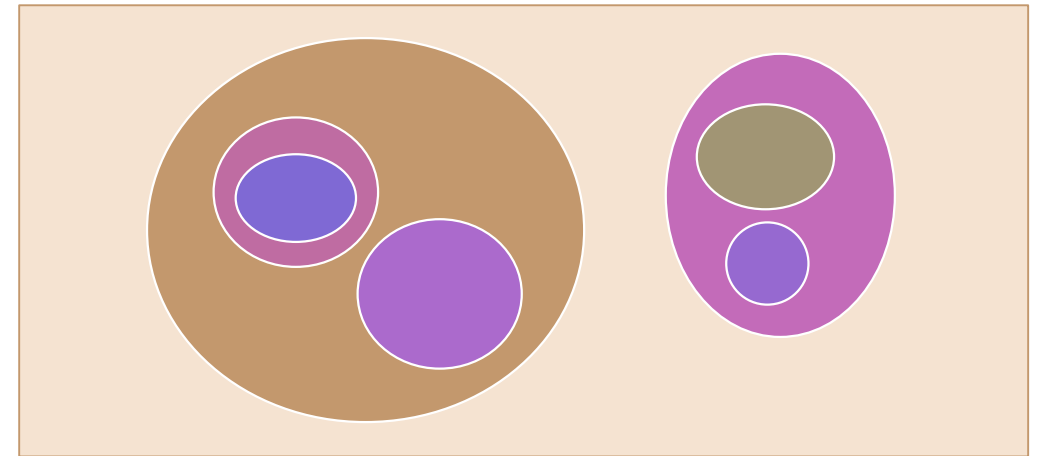
- Exclusive (Hard)
 - Non-overlapping subsets
 - Each item is a member of a single cluster
- Overlapping (Soft)
 - Potentially overlapping subsets
 - A item can simultaneously belong to multiple clusters



Flat Clusters vs Hierarchical Clusters



VS



Challenges in Clustering

- Data is very large
- High dimensional data space.
- Data space is not Euclidean (e.g. NLP problems).

Thank You

Evaluation

- Quality: “goodness” of clusters
- Aspects of validation
 - External Index: Measure the extent to which the clustering results match to ground truth labels
 - Internal Index: without reference to external information
 - Statistical framework: determine reliability
 - To what confidence level, the clusters are not formed by chance

External Evaluation

Evaluated based on data with known class labels e

- Rand Index

RI

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

- Jaccard Index

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

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

- F-measure

Entropy and Purity

- The number of objects in both the k -th cluster and j -th groundtruth: $|C_k \cap P_j|$

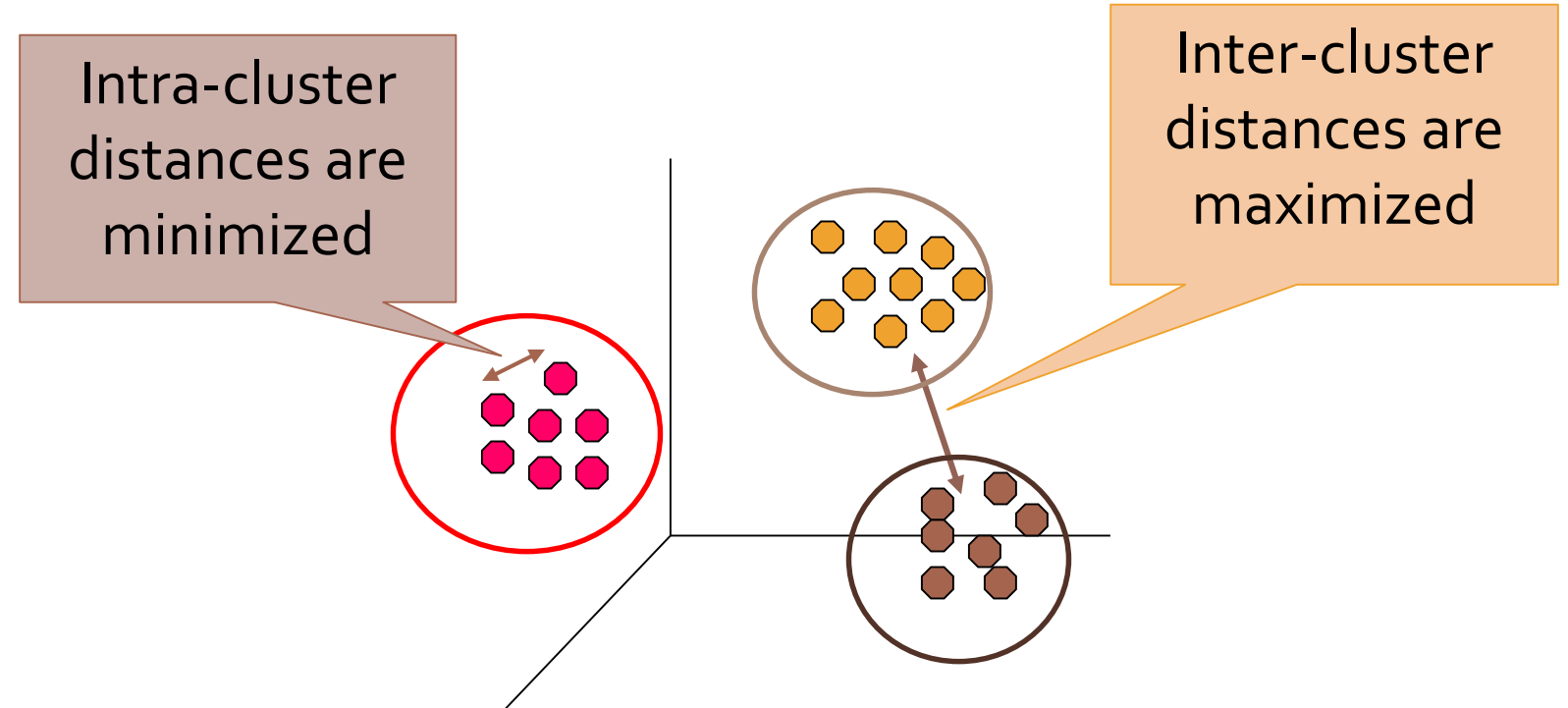
$$\text{Purity} = \frac{1}{n} \sum_k \max_j |C_k \cap P_j|$$

Homogeneity Score:

$$h = 1 - \frac{H(Y_{true} | Y_{predicted})}{H(Y_{true})}$$

Internal Evaluation

- Find groups of objects such that the objects in a group are similar and different from the objects in other groups



Cohesion and Separation

- WCSS/ SSE: Cohesion is measured by within cluster sum of squared distance

$$WCSS = \sum_i \sum_{x \in C_i} dist(x, \mu_i)^2$$

- Separation is measured by the between cluster sum of squares

$$BSS = \sum_i |C_i| dist(\mu, \mu_i)^2$$

$|C_i|$: size of cluster C_i μ : Centroid of whole data set

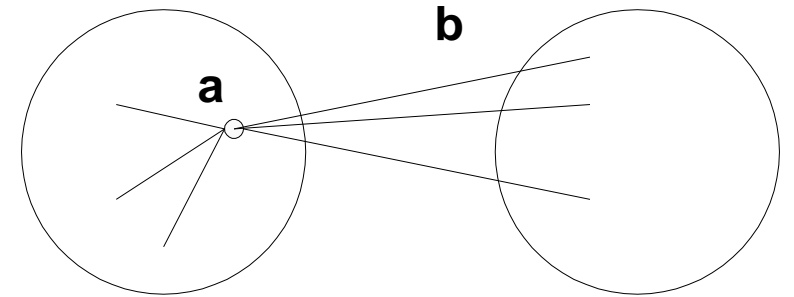
Davies-Bouldin Index

- Assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters

$$DB = \frac{1}{n} \sum_{i=1}^k \max_{j \neq i} \frac{\sigma_i + \sigma_j}{dist(\mu_i, \mu_j)}$$

Silhouette Score

- Combines cohesion and separation scores
- For an instance x
 - a = average distance of x to the points in its cluster
 - b = min (average distance of x to points in another cluster)
 - The **silhouette coefficient** :
 $s = 1 - a/b$ if $a < b$,
($s = b/a - 1$ if $a \geq b$)
 - The closer to 1 the better
- Can calculate the Average Silhouette width for a cluster or a clustering



K-means Clustering

Machine Learning Unit 20

Sudeshna Sarkar

Centre of Excellence in Artificial Intelligence

Indian Institute of Technology Kharagpur

Clustering by Partitioning

Given K

- Construct a partition of m objects

$$X = \{X_1, X_2, \dots, X_m\}$$

$X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ is a vector in a real-valued space
 $X \subseteq \mathbb{R}^n$

into a set of K clusters $C = \{C_1, C_2, \dots, C_K\}$

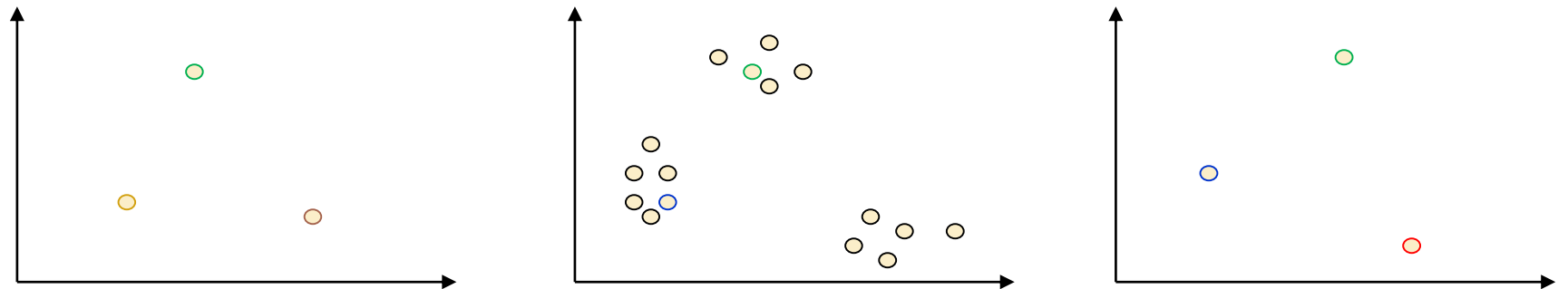
- The cluster mean μ_i serves as a prototype of the cluster C_i

K-means algorithm

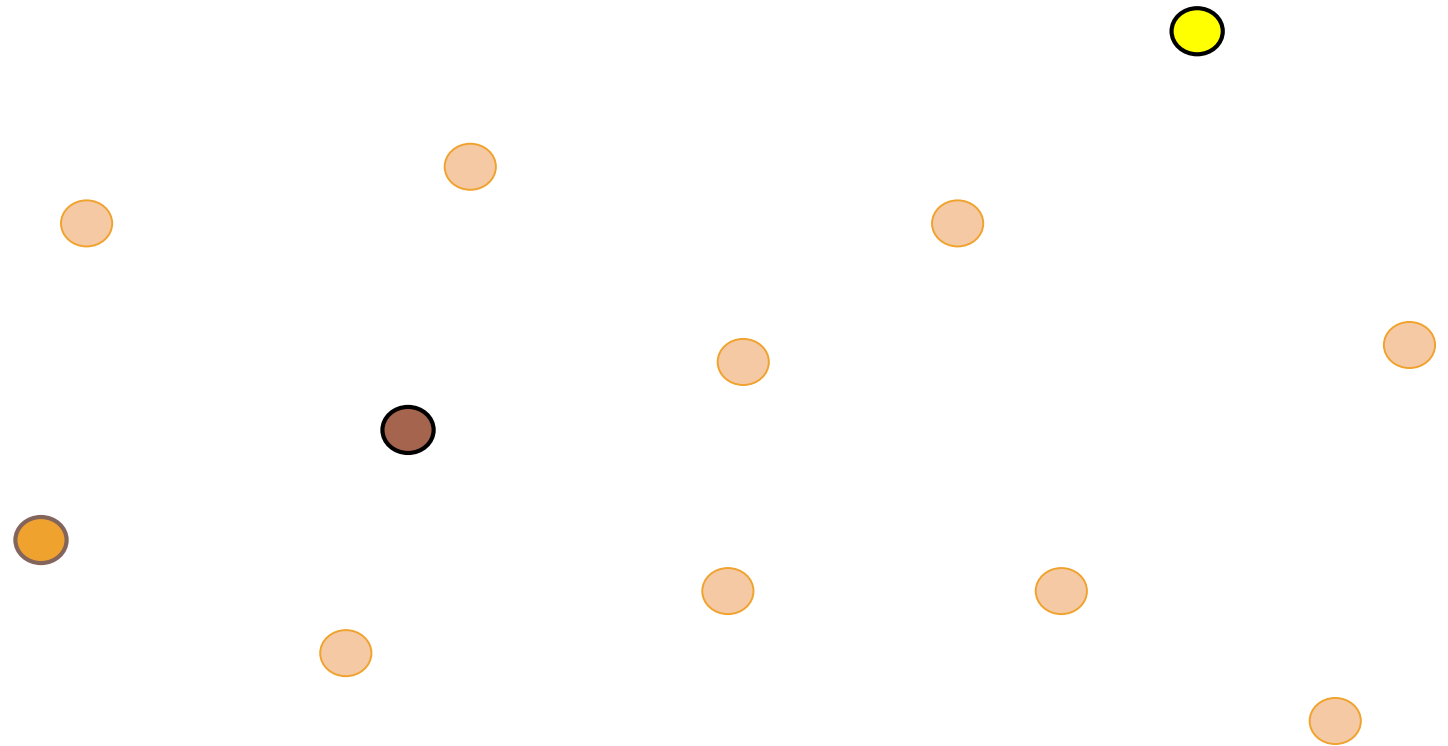
(MacQueen, 1967)

Given K

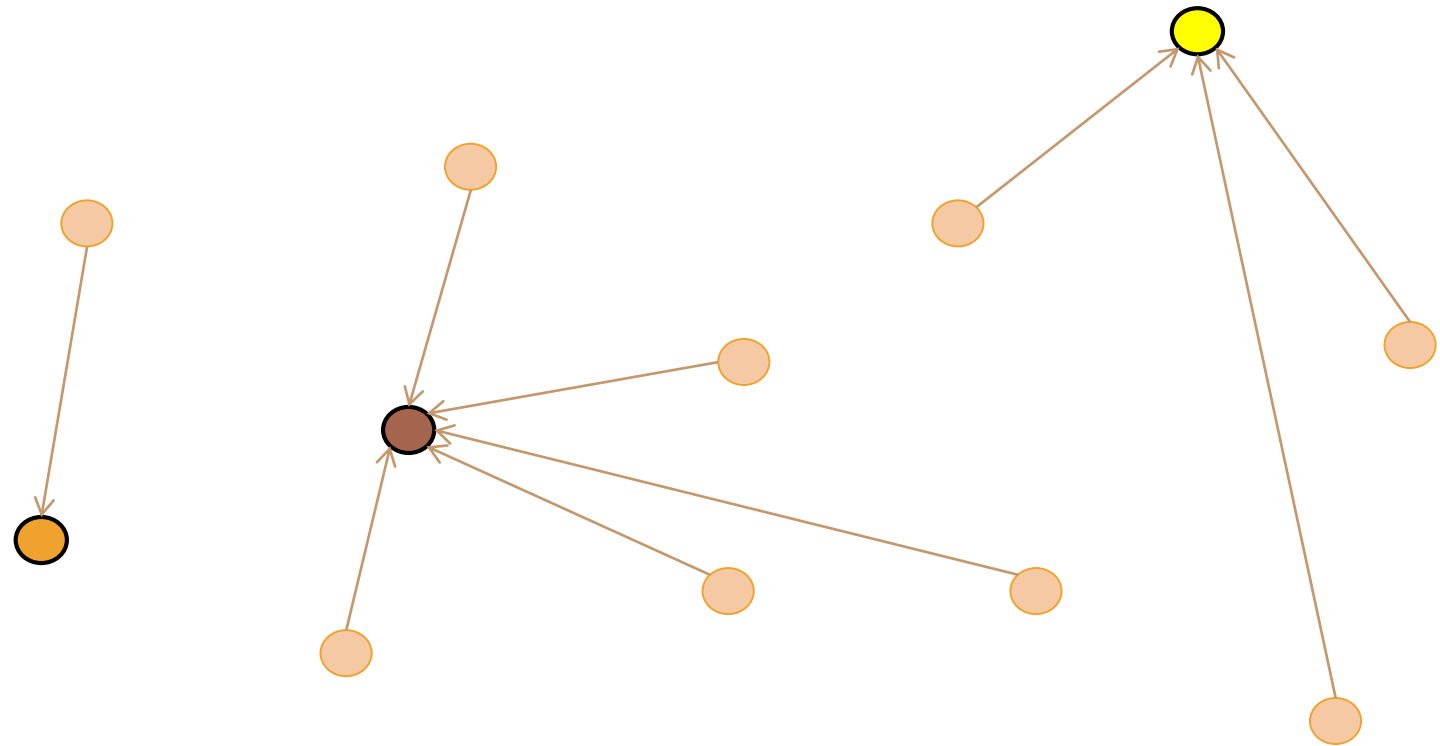
1. **Initialization:** Randomly choose K data points (seeds) to be the initial cluster centres
2. **Cluster Assignment:** Assign each data point to the closest cluster centre
3. **Move Centroid:** Re-compute the cluster centres using the current cluster memberships.
4. If a convergence criterion is not met, go to 2.



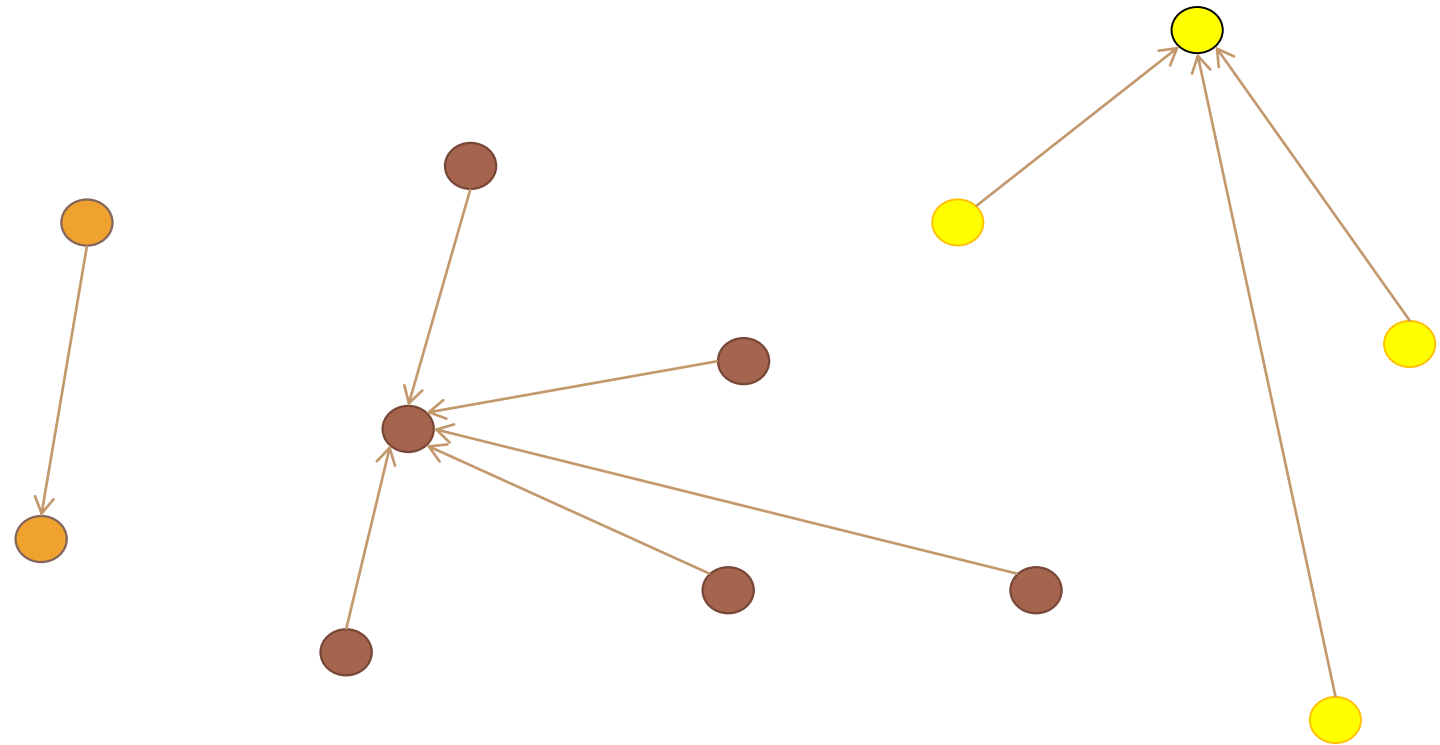
1. Random Initialization



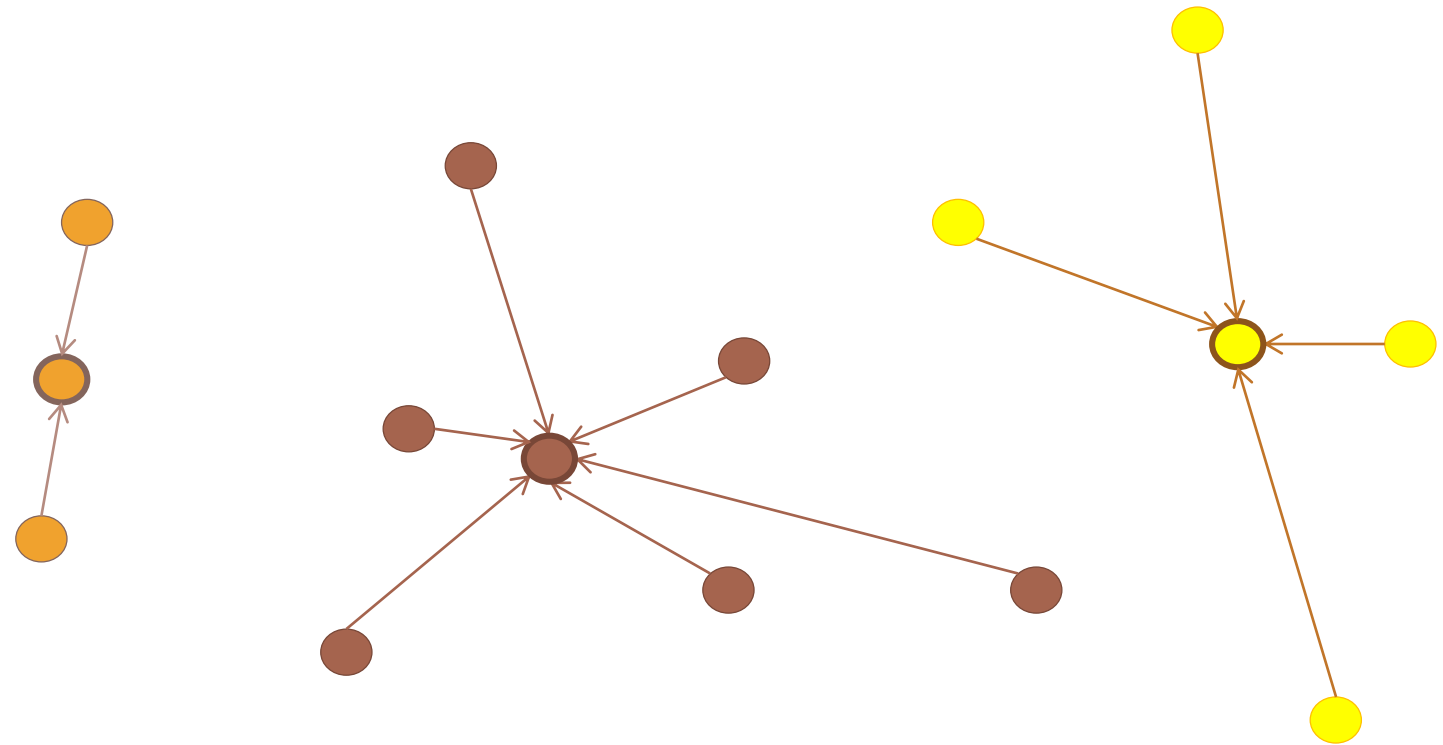
2. Cluster Assignment



2. Cluster Assignment



3. Move Centroid



Stopping criterion

- No re-assignments of data points to different clusters
 - No (or minimum) change in centroids
- or
- Minimum decrease in the *sum of squared error*

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} \|x_i - \mu_i\|^2$$

Optimization Objective

- Good clustering: Within cluster variation is small

$$\underset{C}{\text{minimize}} \left\{ \sum_{i=1}^K WCV(C_i) \right\}$$

- E.g., the within-cluster sum of squares (WCSS) (sum of distance functions of each point in the cluster to the cluster mean)

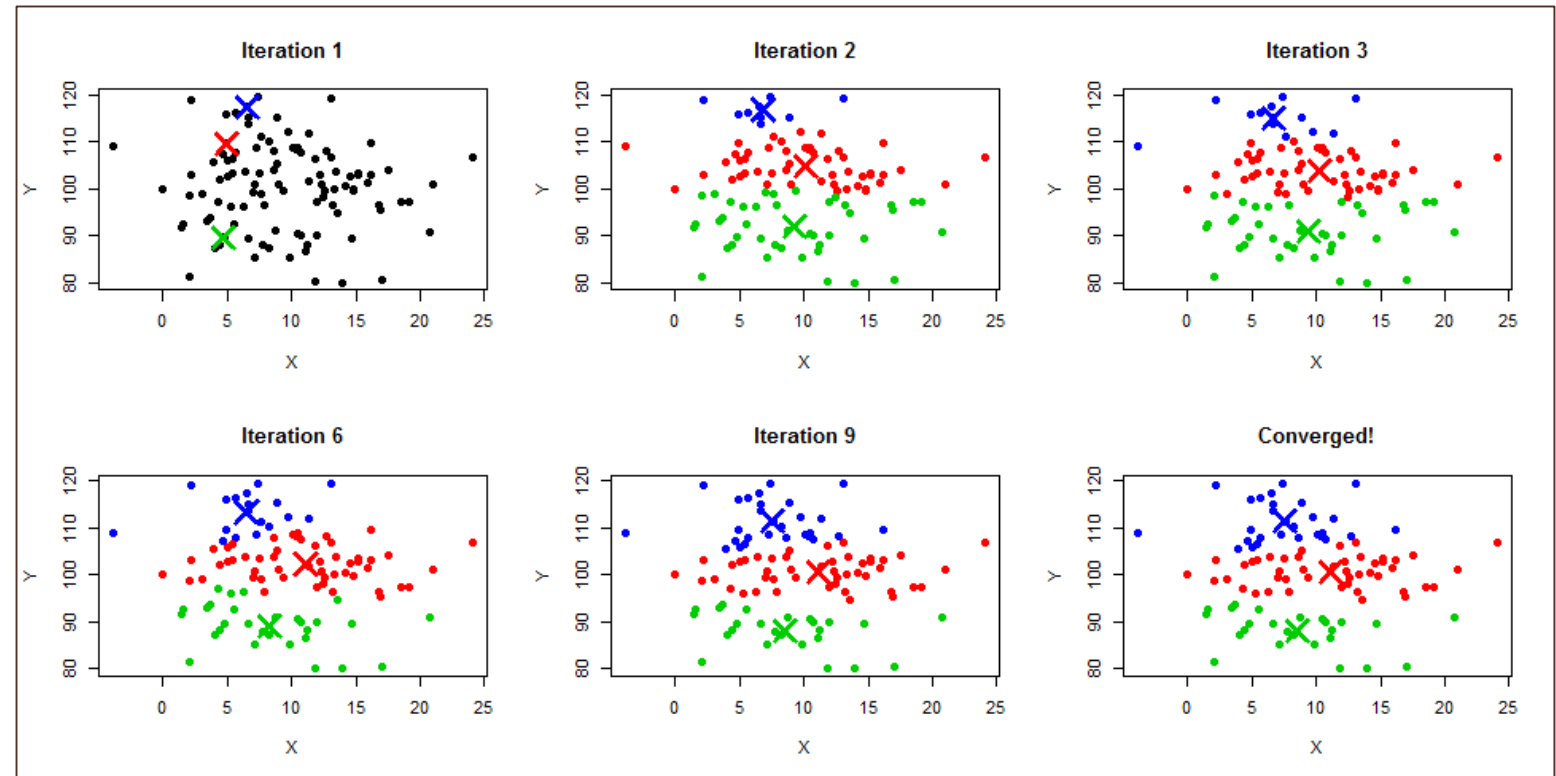
$$\underset{C}{\text{argmin}} \sum_{i=1}^K \sum_{x \in C_k} \|X_i - \mu_i\|^2$$

Convergence Property

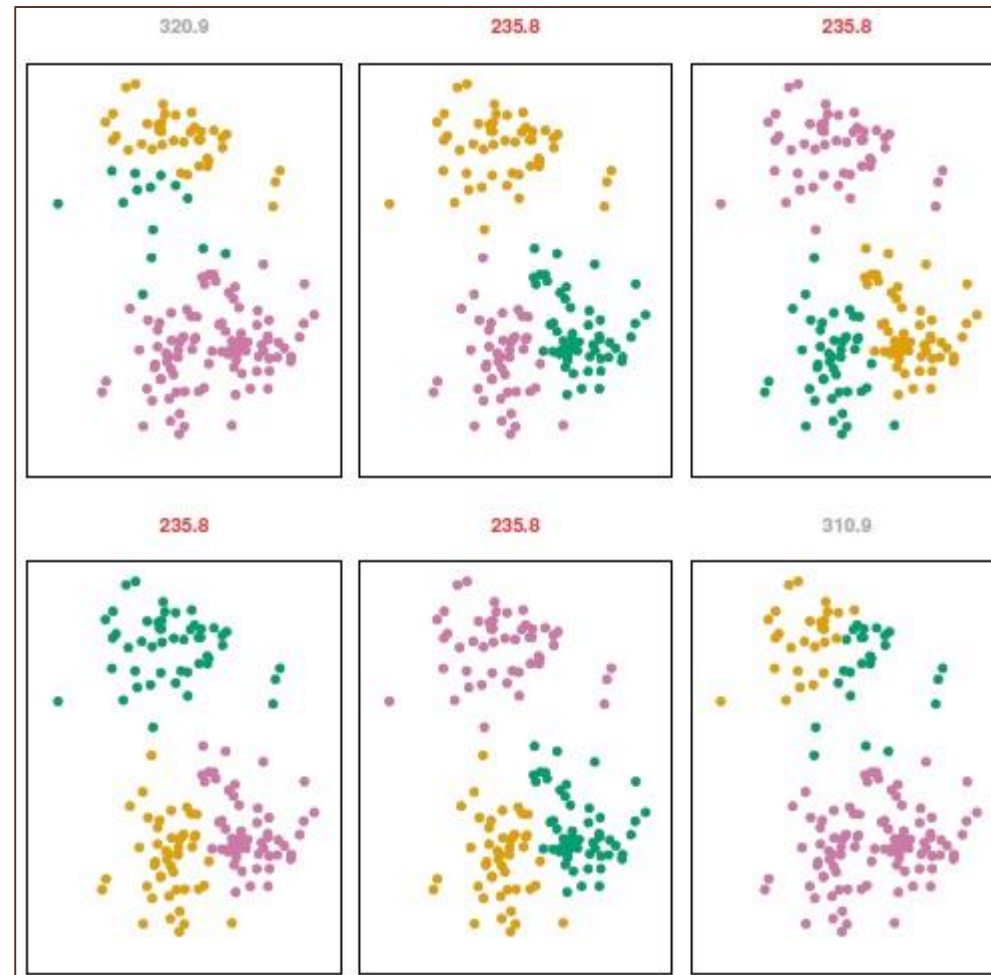
$$SSE = \sum_{i=1}^K \sum_{x \in C_i} \|X_i - \mu_i\|^2$$

- The algorithm is guaranteed to decrease the objective function SSE at every iteration.
- However it is not guaranteed to give the global optimum.

Kmeans illustrated



Different Initial Values



Picking cluster seeds

1. **Lloyd's Method:** Random Initialization
2. **K-Means++ :** Iteratively construct a random sample with good spacing across the dataset.

Picking cluster seeds

Lloyd's Method: Random Initialization

- May converge at a local optimum
1. Perform multiple runs
 - Each run with a different set of randomly chosen seeds
 2. Select that configuration that gives minimum SSE

K-means++

- Choose centers at random from the data points
 - Weight the probability of choosing the centres according to their squared distance from the closest centre already chosen

K-means++ implementation

- Let $D(X)$ be the distance between a point X and its nearest centre.
- Choose the next centre proportional to $D^2(X)$

- Choose c_1 at random.
- For $j = 2$ to K
Pick c_j from the remaining data points $\{X_i\}$
 $\Pr(c_j = X_i) \propto (X_i)$

Convergence of K -Means

- Consider data points in Euclidean space
- Error of each data point = Euclidean distance of the point to its closest centroid
- SSE: total sum of the squared errors for each point
- Re-computation monotonically decreases SSE (finds a local minima)

Convergence of K -Means

Recomputation monotonically decreases each square error

Proof:

(n_j is number of members in cluster j):

$\sum (x_i - a)^2$ reaches minimum for:

$$\sum -2(x_i - a) = 0$$

$$\text{i.e., } \sum x_i = \sum a = n_j a$$

$$\text{i.e., } a = 1/n_j \sum x_i = c_j$$

K -means typically converges quickly

Time Complexity

m items, n dimensions, K clusters, I iterations

- Computing distance between two items is $O(n)$
- Reassigning clusters: $O(Km)$ distance computations
- Total for one iteration $O(Kmn)$
- Computing centroids: Each item gets added once to some centroid: $O(mn)$
- Assume these two steps are each done once for I iterations: $O(IKmn)$

How to select K?

1: Use cross validation to select K

- What should we optimize?

2: Let the domain expert look at the clustering and decide

3: The “knee” solution

Deciding K

- Plot the objective function values for different values of K
- “knee finding” or “elbow finding”.

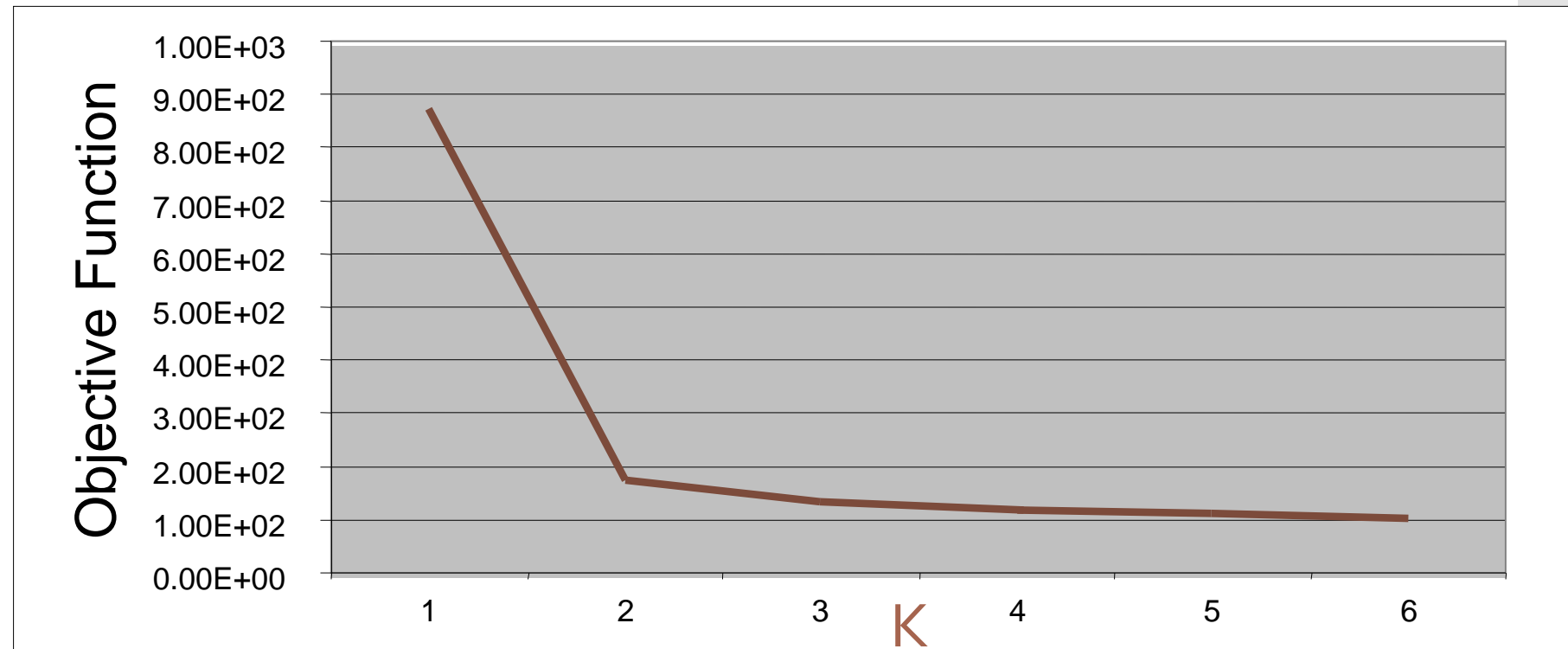


Figure from slide by Eamonn Keogh

Advantages

- Fast, robust easy to understand
- Relatively efficient: $O(IKmn)$
- Normally, $K, I, m \ll n$
- Gives best result when data set are distinct or well separated from each other.

Disadvantages

- Requires apriori specification of the number of cluster centers.
- Hard assignment of data points to clusters
- Euclidean distance measures can unequally weight underlying factors.
- Applicable only when mean is defined
- Only local optima

Thank You

Evaluation of Clustering

Machine Learning Unit 21

Sudeshna Sarkar

Centre of Excellence in Artificial Intelligence

Indian Institute of Technology Kharagpur

Evaluation

- Quality: “goodness” of clusters
- Aspects of validation
 - External Index: Measure the extent to which the clustering results match to ground truth labels
 - Internal Index: without reference to external information
 - Statistical framework: determine reliability
 - To what confidence level, the clusters are not formed by chance

External Evaluation

Evaluated based on data with known class labels e

- Rand Index

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

- Jaccard Index

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}$$

- F-measure

External Evaluation

Entropy and Purity

- The number of objects in both the k -th cluster and j -th groundtruth: $|C_k \cap P_j|$

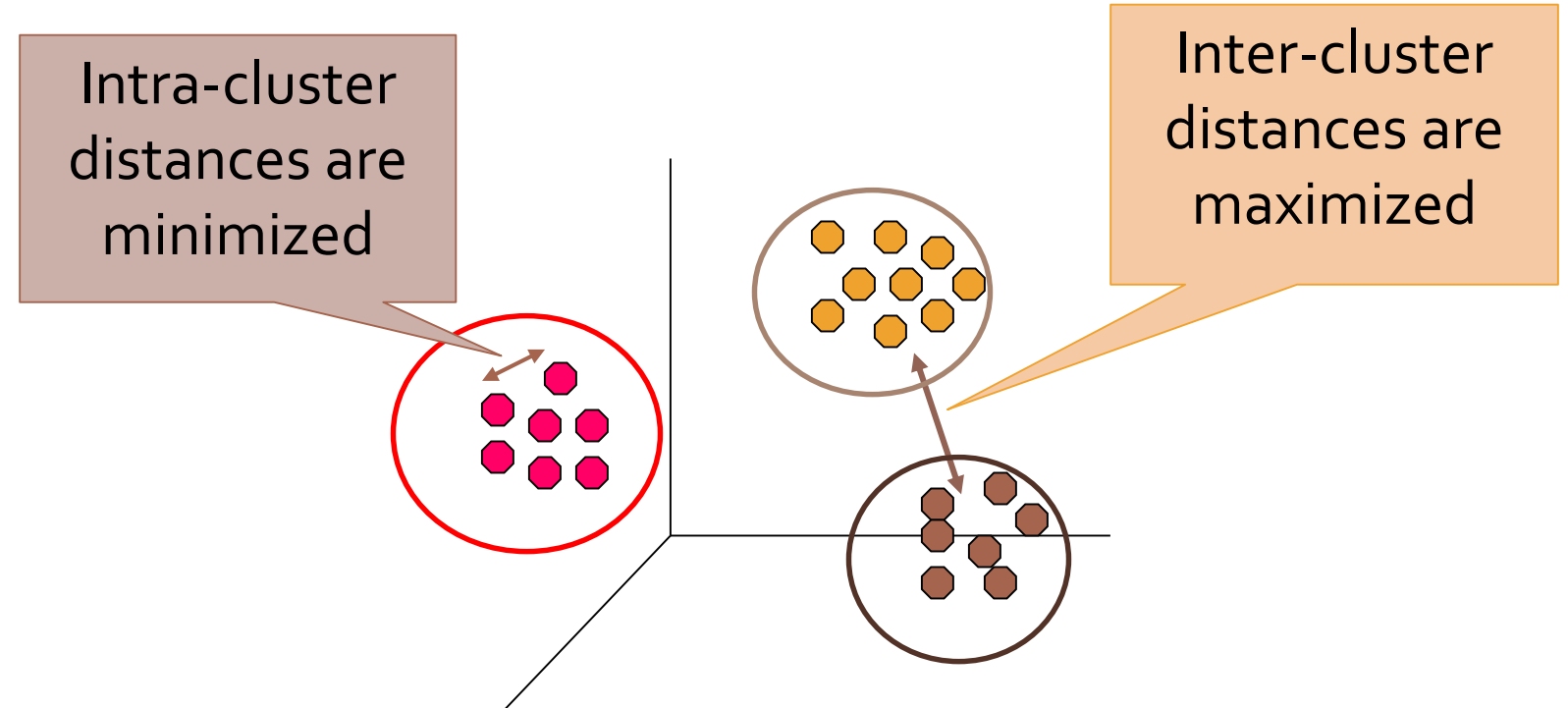
$$\text{Purity} = \frac{1}{n} \sum_k \max_j |C_k \cap P_j|$$

Homogeneity Score:

$$h = 1 - \frac{H(Y_{\text{true}} | Y_{\text{predicted}})}{H(Y_{\text{true}})}$$

Internal Evaluation

- Find groups of objects such that the objects in a group are similar and different from the objects in other groups



Cohesion and Separation

- WCSS/ SSE: Cohesion is measured by within cluster sum of squared distance

$$WCSS = \sum_i \sum_{x \in C_i} dist(x, \mu_i)^2$$

- Separation is measured by the between cluster sum of squares

$$BSS = \sum_i |C_i| dist(\mu, \mu_i)^2$$

$|C_i|$: size of cluster C_i μ : Centroid of whole data set

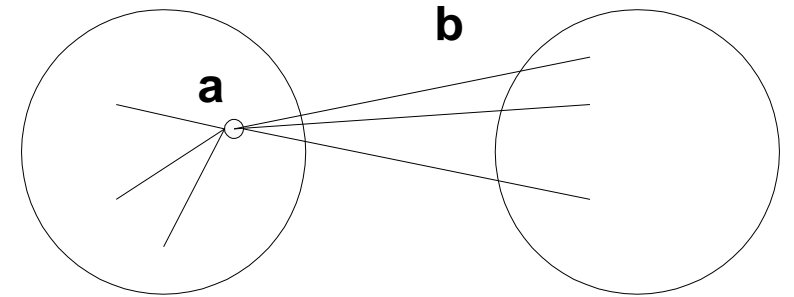
Davies-Bouldin Index

- Assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters

$$DB = \frac{1}{n} \sum_{i=1}^k \max_{j \neq i} \frac{\sigma_i + \sigma_j}{dist(\mu_i, \mu_j)}$$

Silhouette Score

- Combines cohesion and separation scores
- For an instance x
 - a = average distance of x to the points in its cluster
 - b = min (average distance of x to points in another cluster)
 - The **silhouette coefficient** :
 $s = 1 - a/b$ if $a < b$,
($s = b/a - 1$ if $a \geq b$)
 - The closer to 1 the better
- Can calculate the Average Silhouette width for a cluster or a clustering



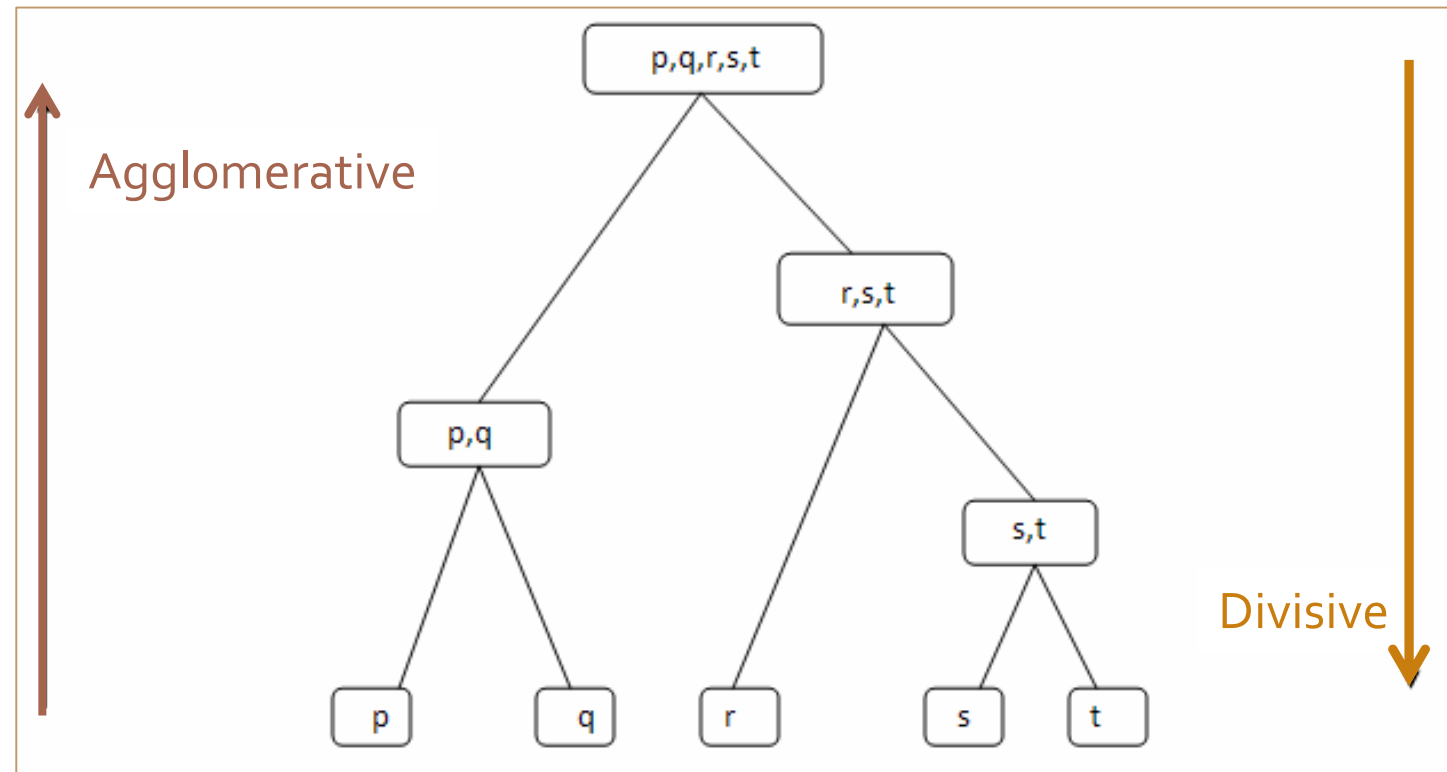
Issues

- Representation
 - Representation of instances (features)
 - Proximity function (similarity / distance measure)
- Hard v Soft clustering
 - Can an instance belong to more than one cluster?
- Clustering algorithm
 - Flat or Hierarchical
 - Density based
 - ...
- Number of clusters
 - Fixed
 - Data driven

Hierarchical Algorithms

Agglomerative (bottom-up): Start with each point as a cluster. Clusters are combined based on their “closeness”.

Divisive (top-down): Start with one cluster including all points and recursively split each cluster.



Hierarchical Clustering

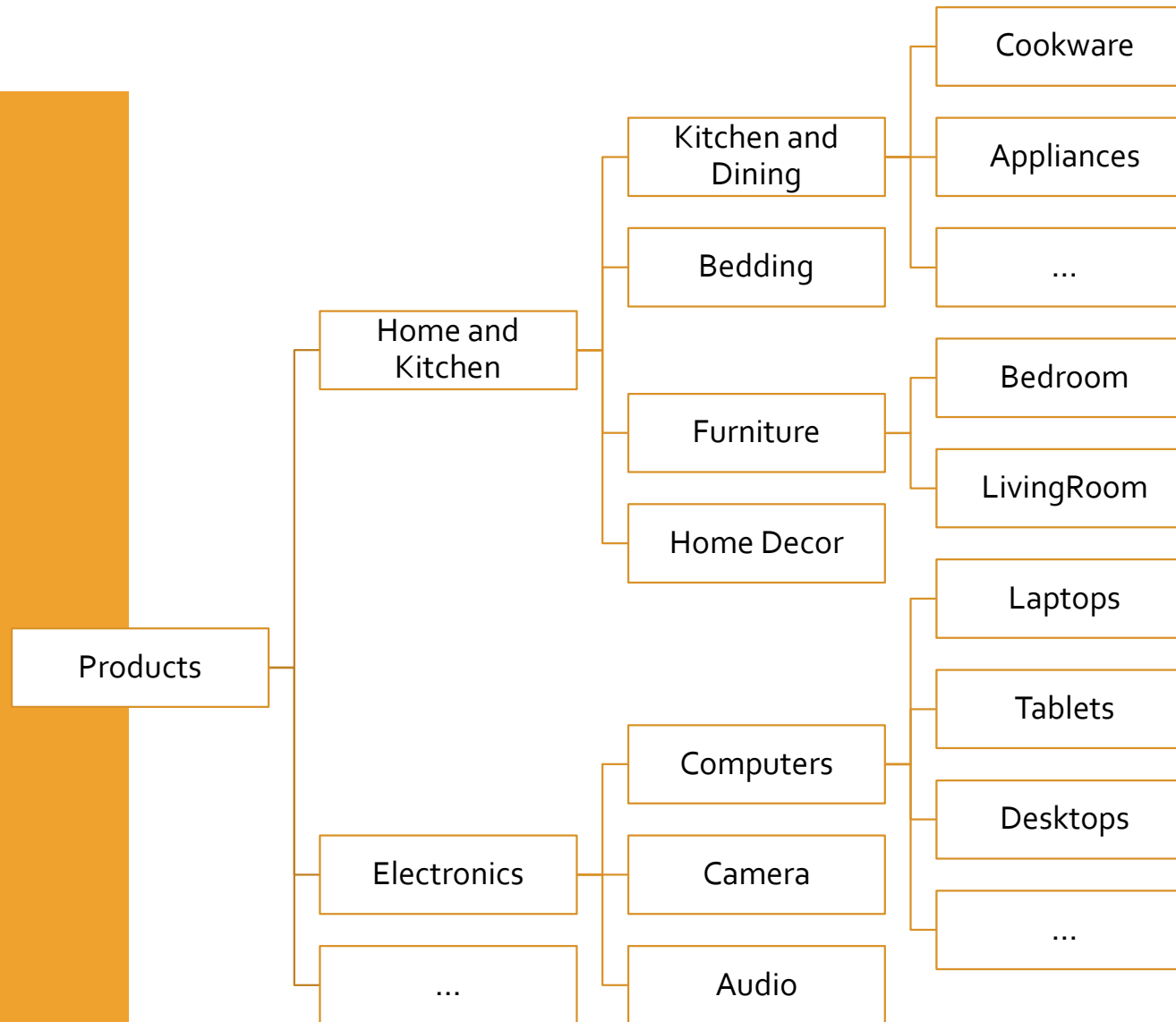
Machine Learning Unit 22

Sudeshna Sarkar

Centre of Excellence in Artificial Intelligence

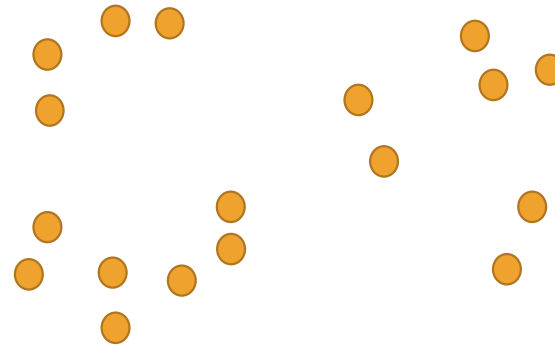
Indian Institute of Technology Kharagpur

Hierarchical Clustering



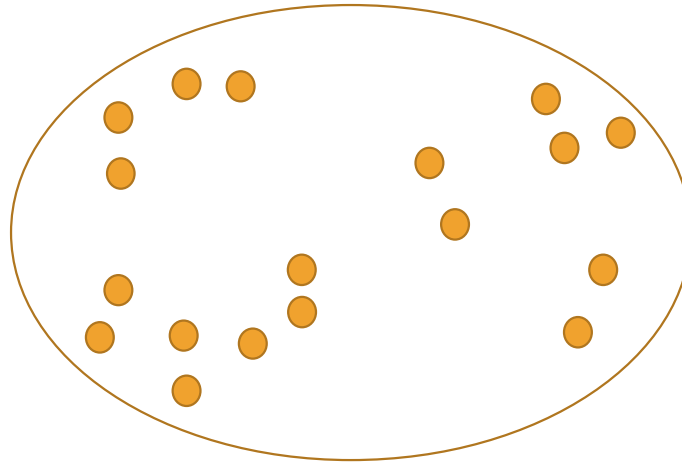
Top-down Hierarchical Clustering (Divisive)

Start with a single cluster of all the samples.
Partition recursively into two clusters
until there is one cluster for each observation.



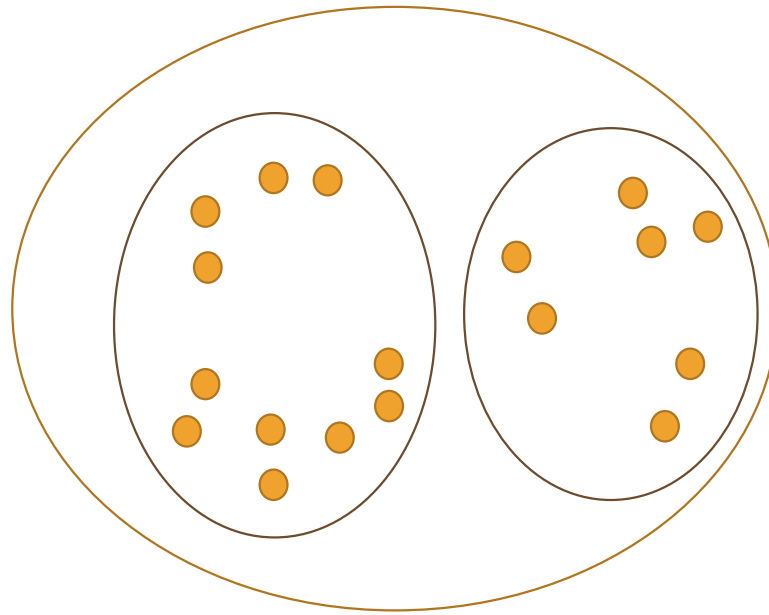
Top-down Hierarchical Clustering (Divisive)

Start with a single cluster of all the samples.
Partition recursively into two clusters
until there is one cluster for each observation.



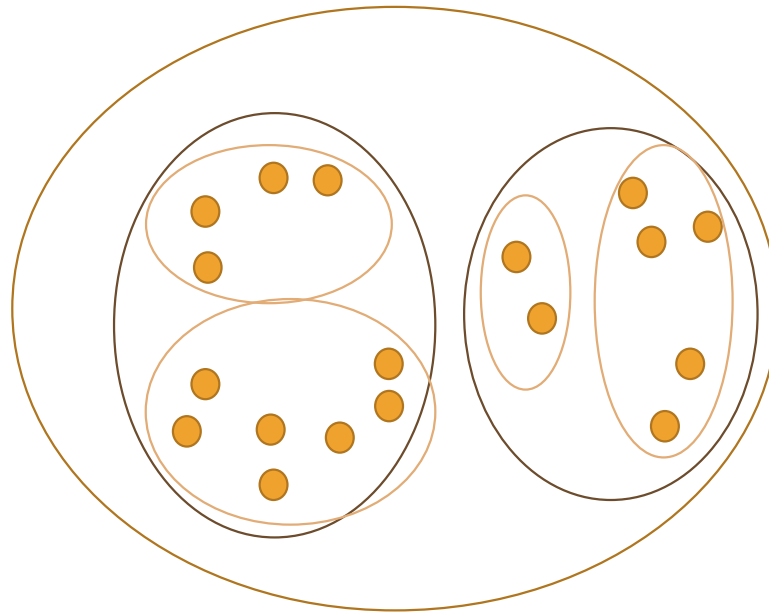
Top-down Hierarchical Clustering (Divisive)

Start with a single cluster of all the samples.
Partition recursively into two clusters
until there is one cluster for each observation.



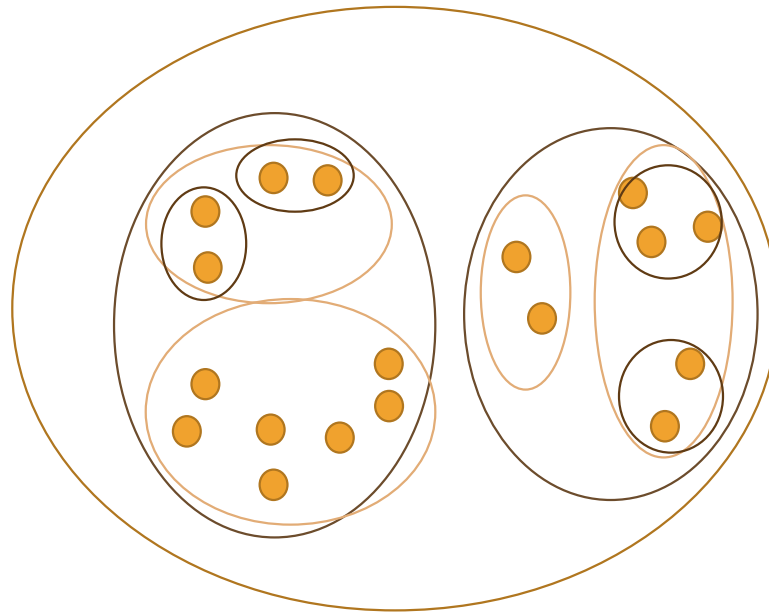
Top-down Hierarchical Clustering (Divisive)

Start with a single cluster of all the samples.
Partition recursively into two clusters
until there is one cluster for each observation.



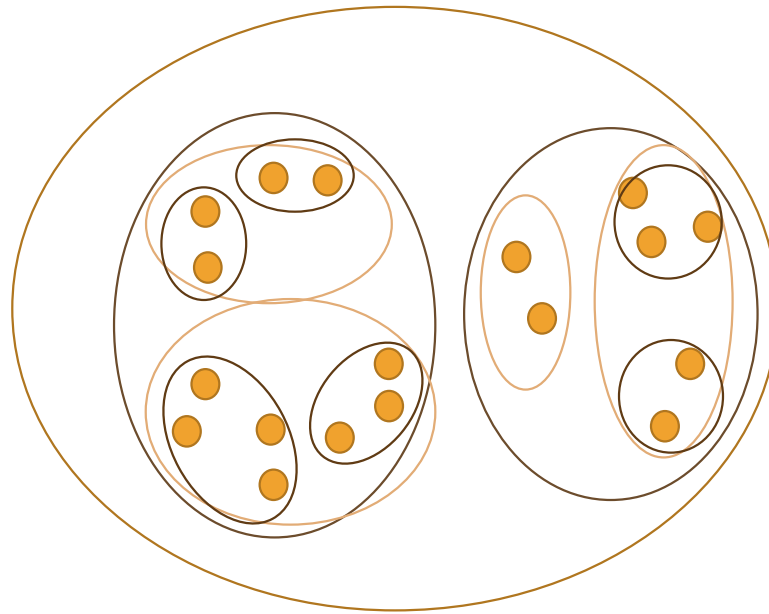
Top-down Hierarchical Clustering (Divisive)

Start with a single cluster of all the samples.
Partition recursively into two clusters
until there is one cluster for each observation.



Top-down Hierarchical Clustering (Divisive)

Start with a single cluster of all the samples.
Partition recursively into two clusters
until there is one cluster for each observation.



Types of hierarchical clustering

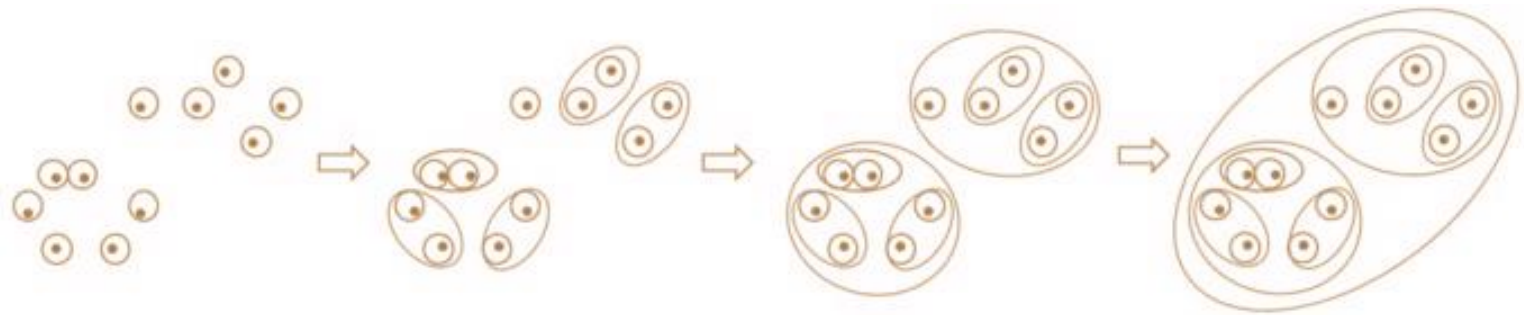
1. Divisive Hierarchical Clustering

Hierarchical k-means

- Start with all data points $\{x_1, x_2, \dots, x_m\}$ in one cluster
- Run k-means on the data and split into k child clusters $\{c_1, c_2, \dots, c_k\}$
- Recursively run *k-means* on each child cluster
- Stop when only singleton clusters of individual data points remain.

Bottom-up Hierarchical Clustering (Agglomerative)

Initially each sample is treated as a single cluster
Successively merge pairs of clusters
until all clusters have been merged into a single cluster.



Types of hierarchical clustering

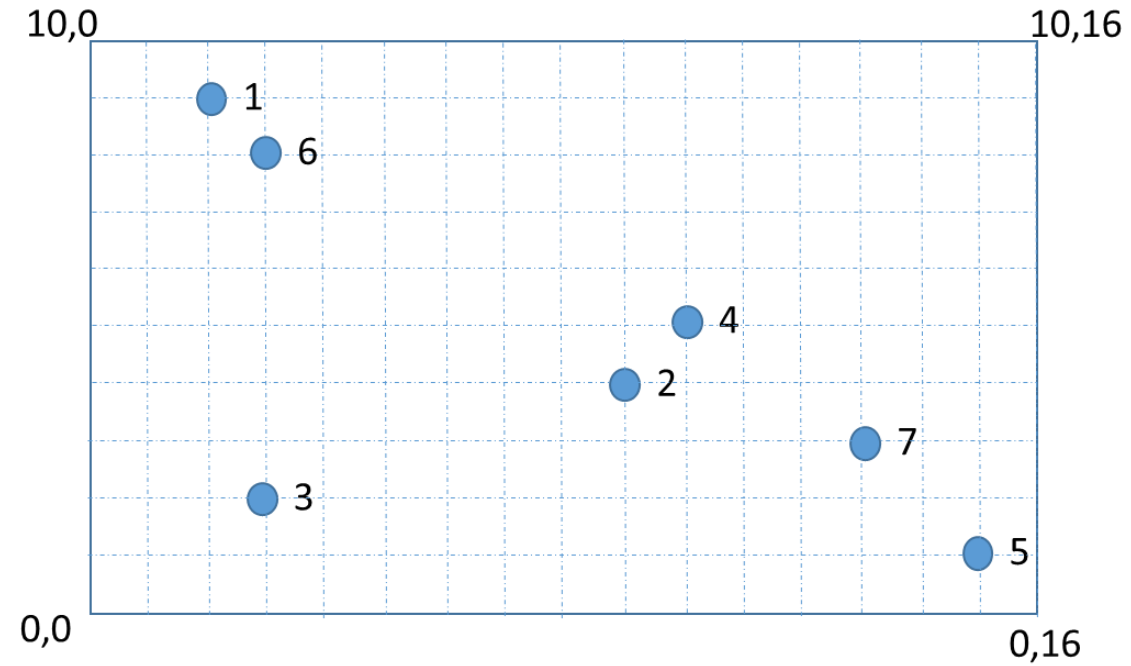
2. Agglomerative (bottom up) clustering

Agglomerative (bottom up) clustering: Builds the hierarchical tree from the bottom level

1. Start with a collection of m singleton clusters
$$c_i = \{x_i\}$$
2. Repeat
 1. Merge the most similar (or nearest) pair of clusters c_i, c_j into a new cluster c_{i+j}
 2. Remove c_i, c_j from the collection and add c_{i+j}
3. Until a single cluster is left

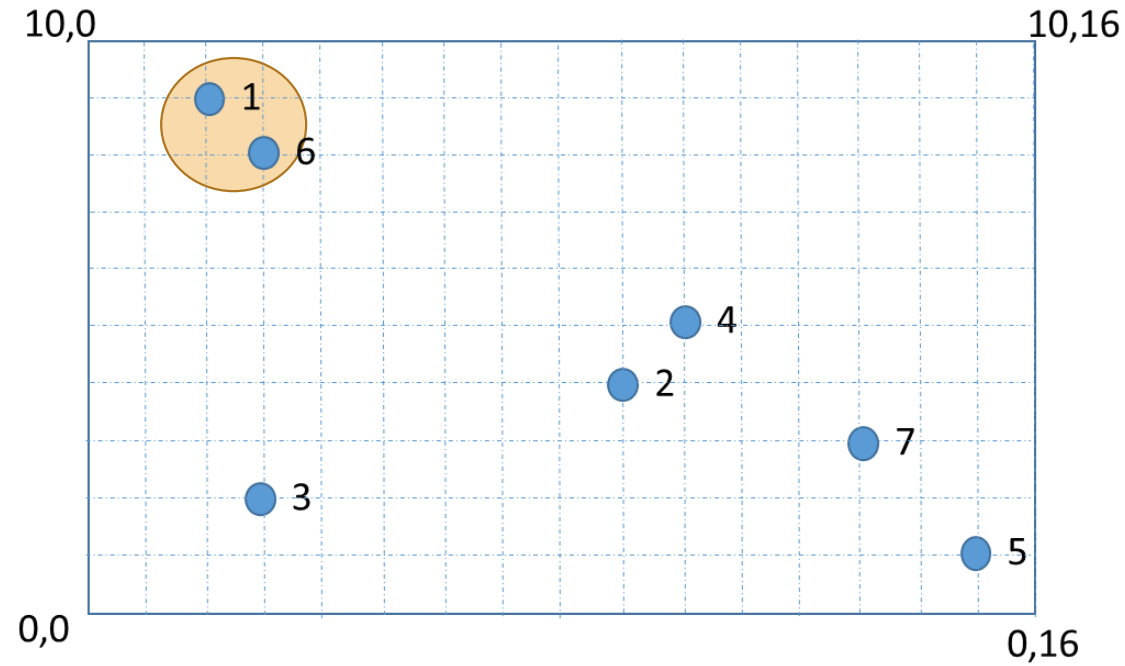
Produces a hierarchical tree of clusters or a dendrogram

Hierarchical Clustering: Example



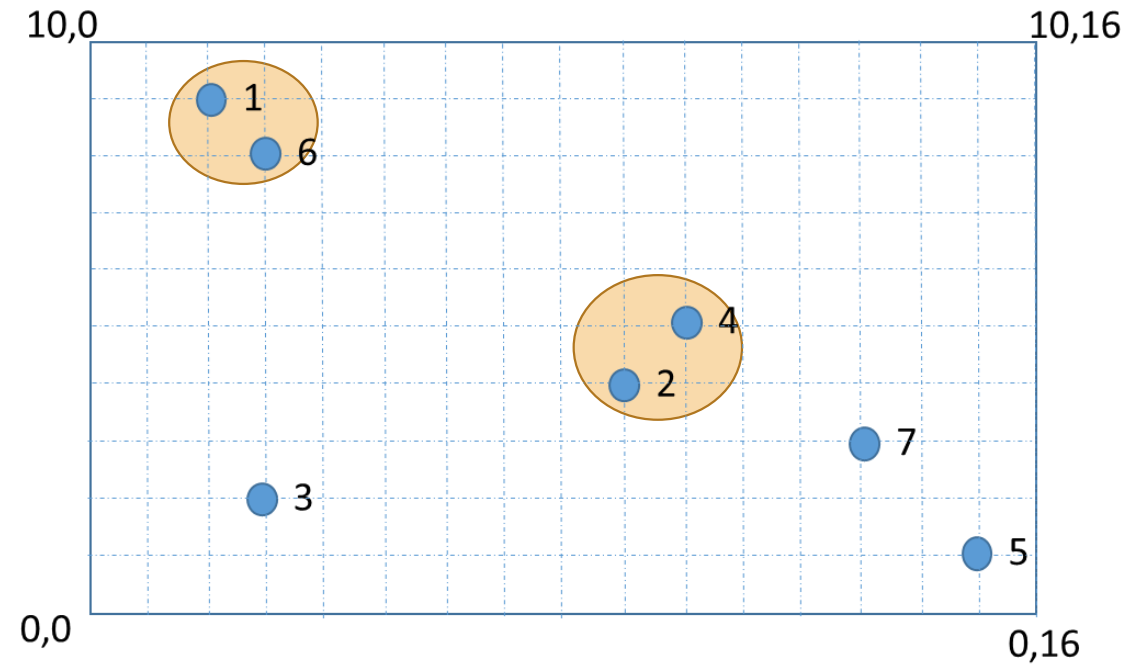
1. $C = \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}$

Hierarchical Clustering: Example



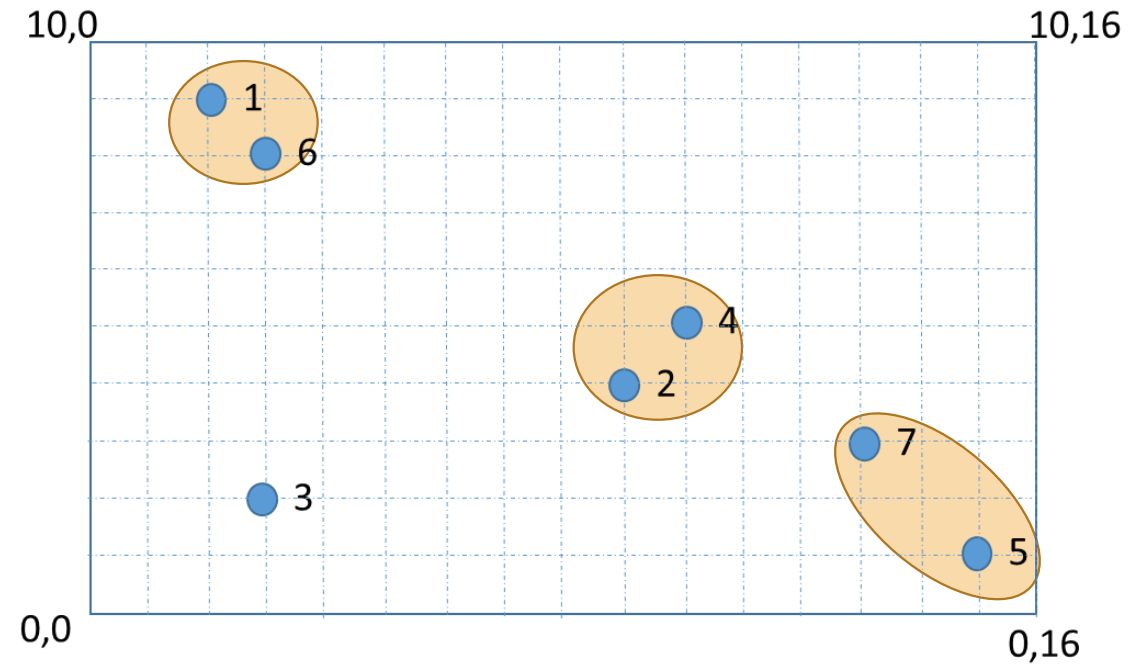
1. $C = \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}$
2. $C = \{1, 6\}, \{2\}, \{3\}, \{4\}, \{5\}, \{7\}$

Hierarchical Clustering: Example



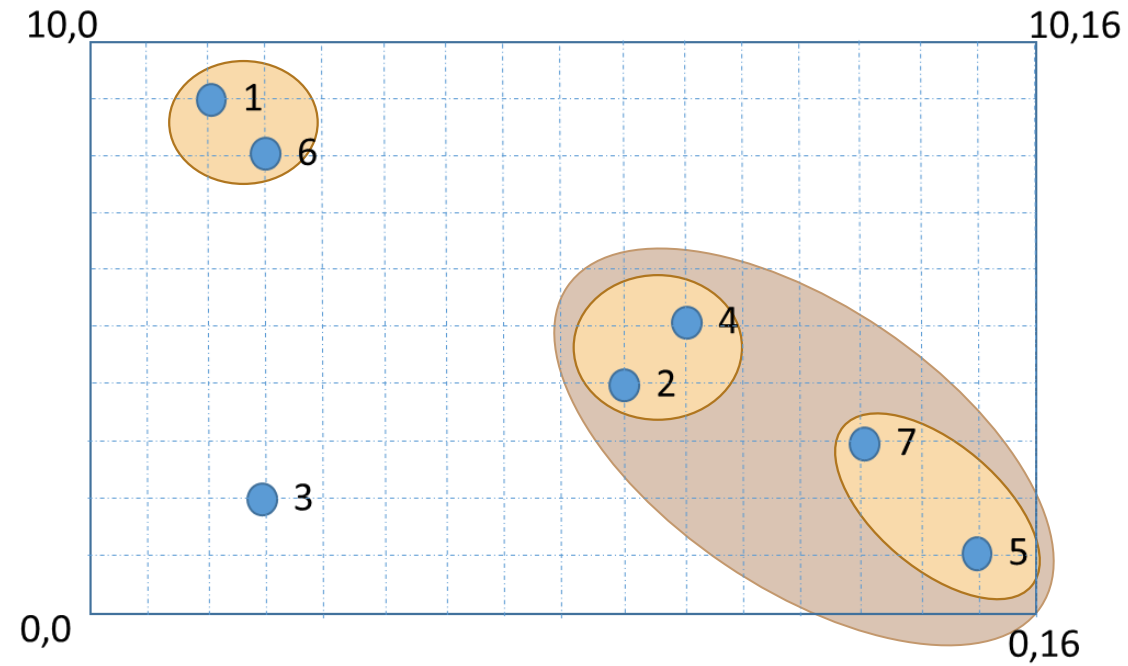
1. $C = \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}$
2. $\{1, 6\}, \{2\}, \{3\}, \{4\}, \{5\}, \{7\}$
3. $\{1, 6\}, \{2, 4\}, \{3\}, \{5\}, \{7\}$

Hierarchical Clustering: Example



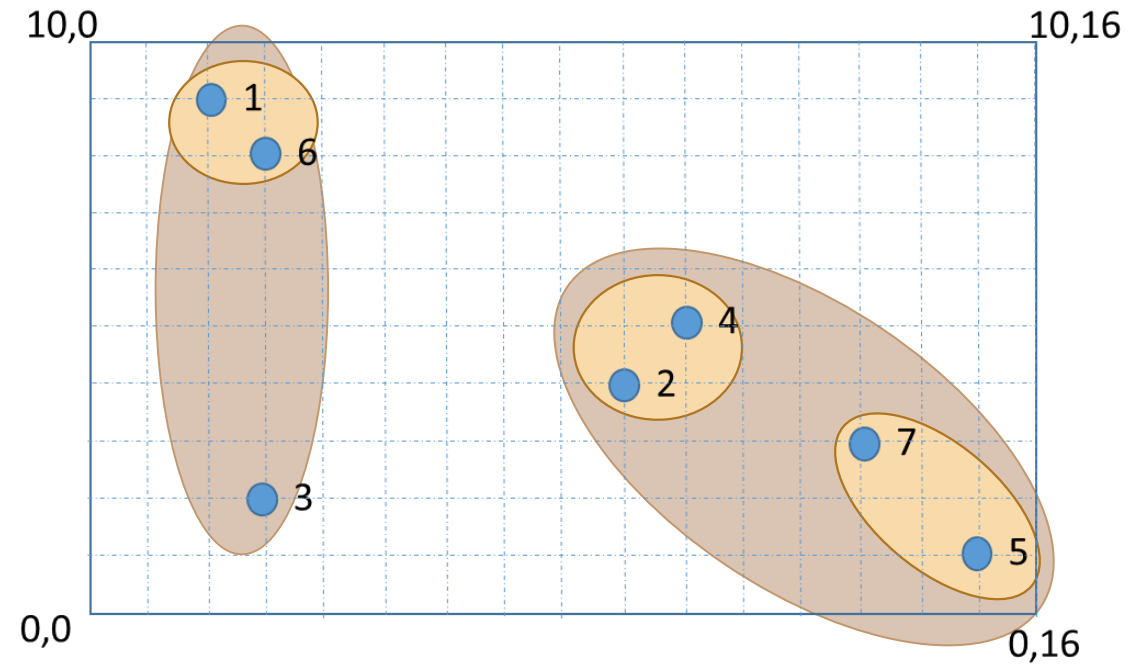
1. $C = \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}$
2. $\{1, 6\}, \{2\}, \{3\}, \{4\}, \{5\}, \{7\}$
3. $\{1, 6\}, \{2, 4\}, \{3\}, \{5\}, \{7\}$
4. $\{1, 6\}, \{2, 4\}, \{3\}, \{5, 7\}$

Hierarchical Clustering: Example



1. $C = \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}$
2. $\{1, 6\}, \{2\}, \{3\}, \{4\}, \{5\}, \{7\}$
3. $\{1, 6\}, \{2, 4\}, \{3\}, \{5\}, \{7\}$
4. $\{1, 6\}, \{2, 4\}, \{3\}, \{5, 7\}$
5. $\{1, 6\}, \{2, 4, 5, 7\}, \{3\}$

Hierarchical Clustering: Example



1. $C = \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}$

2. $\{1, 6\}, \{2\}, \{3\}, \{4\}, \{5\}, \{7\}$

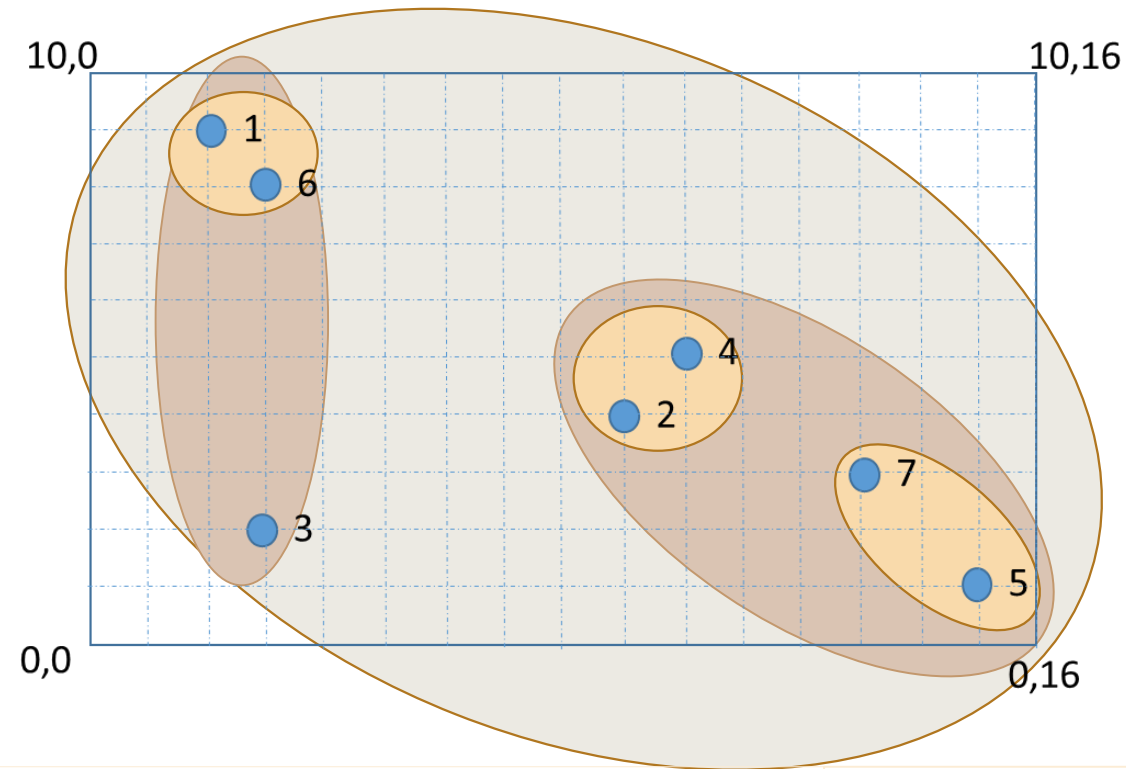
3. $\{1, 6\}, \{2, 4\}, \{3\}, \{5\}, \{7\}$

4. $\{1, 6\}, \{2, 4\}, \{3\}, \{5, 7\}$

5. $\{1, 6\}, \{2, 4, 5, 7\}, \{3\}$

6. $\{1, 6, 3\}, \{2, 4, 5, 7\}$

Hierarchical Clustering: Example



1. $C = \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}$

2. $\{1, 6\}, \{2\}, \{3\}, \{4\}, \{5\}, \{7\}$

3. $\{1, 6\}, \{2, 4\}, \{3\}, \{5\}, \{7\}$

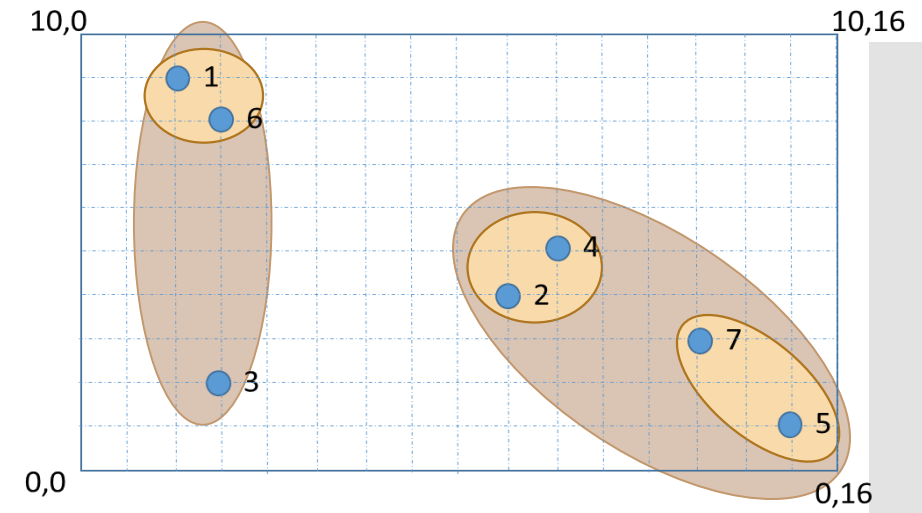
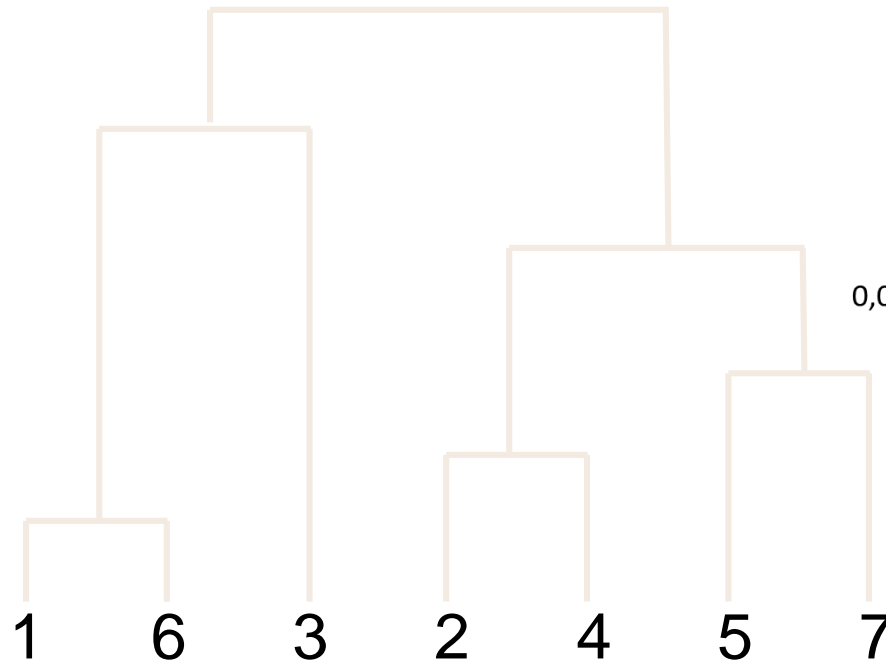
4. $\{1, 6\}, \{2, 4\}, \{3\}, \{5, 7\}$

5. $\{1, 6\}, \{2, 4, 5, 7\}, \{3\}$

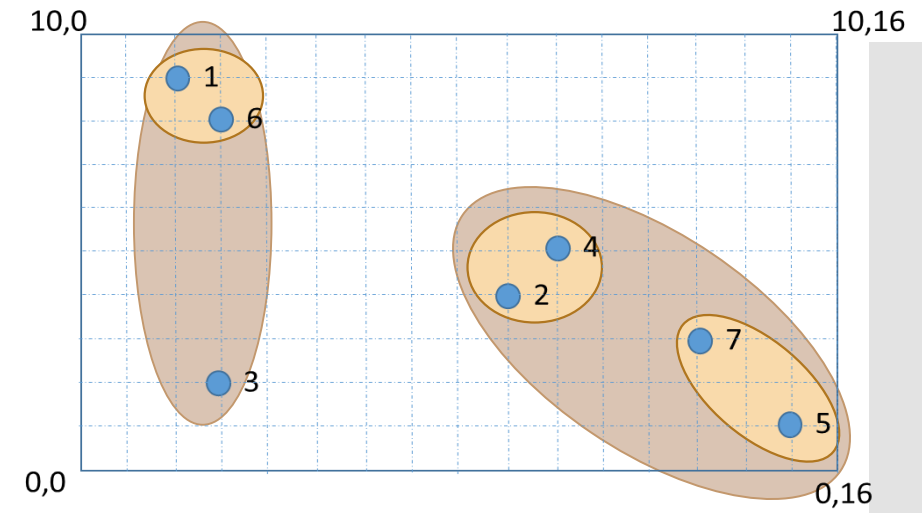
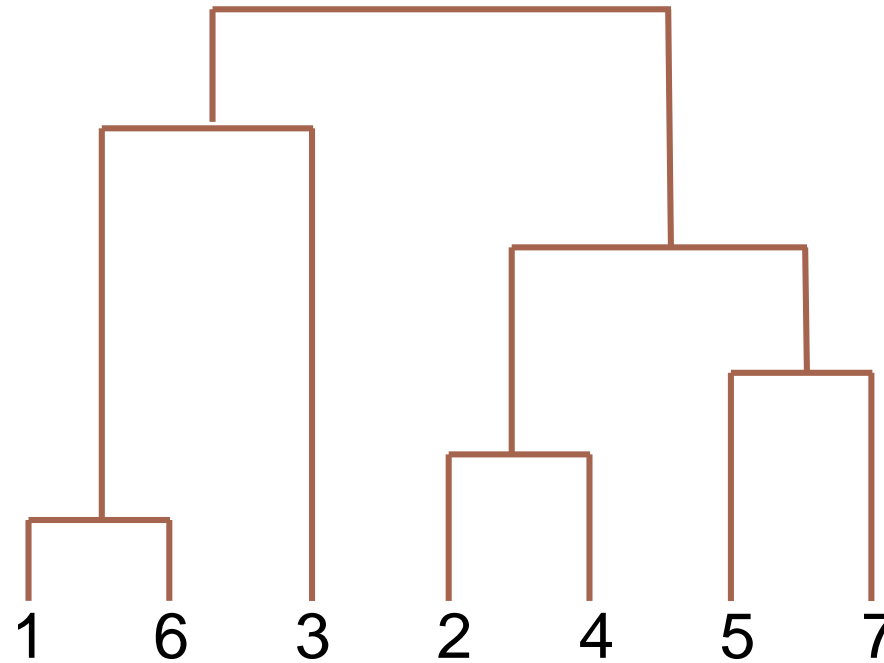
6. $\{1, 6, 3\}, \{2, 4, 5, 7\}$

7. $\{1, 6, 3, 2, 4, 5, 7\}$

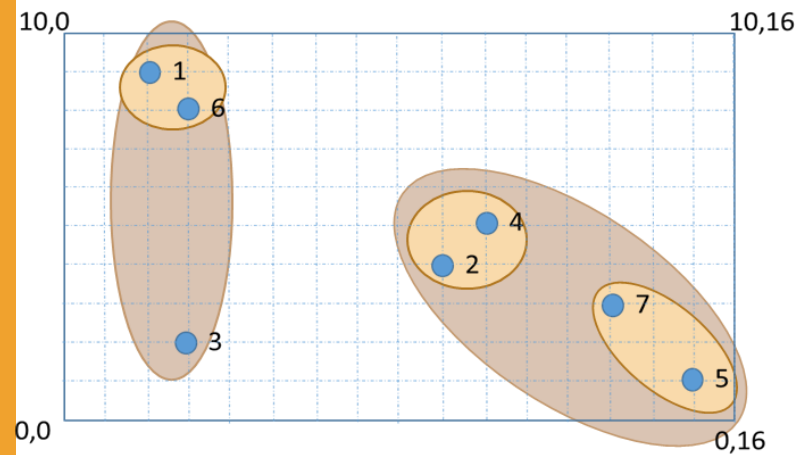
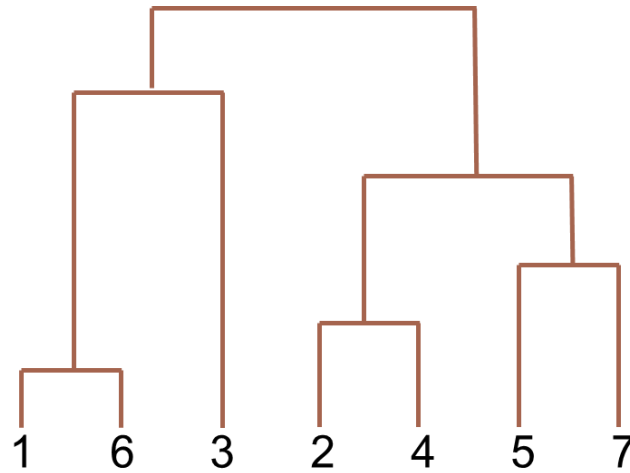
Dendrogram: Hierarchical Clustering



Dendrogram: Hierarchical Clustering



Dendrogram: Hierarchical Clustering



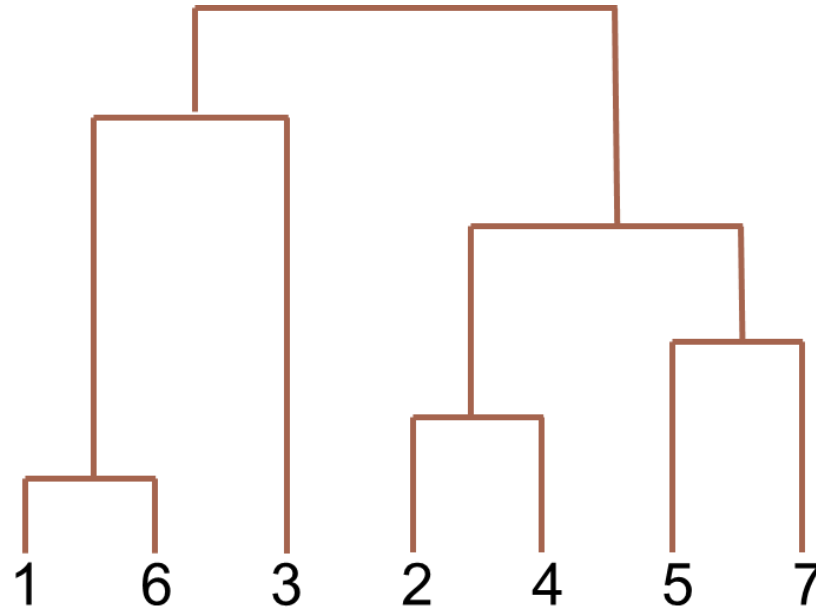
Dendrogram

- Input set S
- Nodes represent subsets of S

Features of the tree

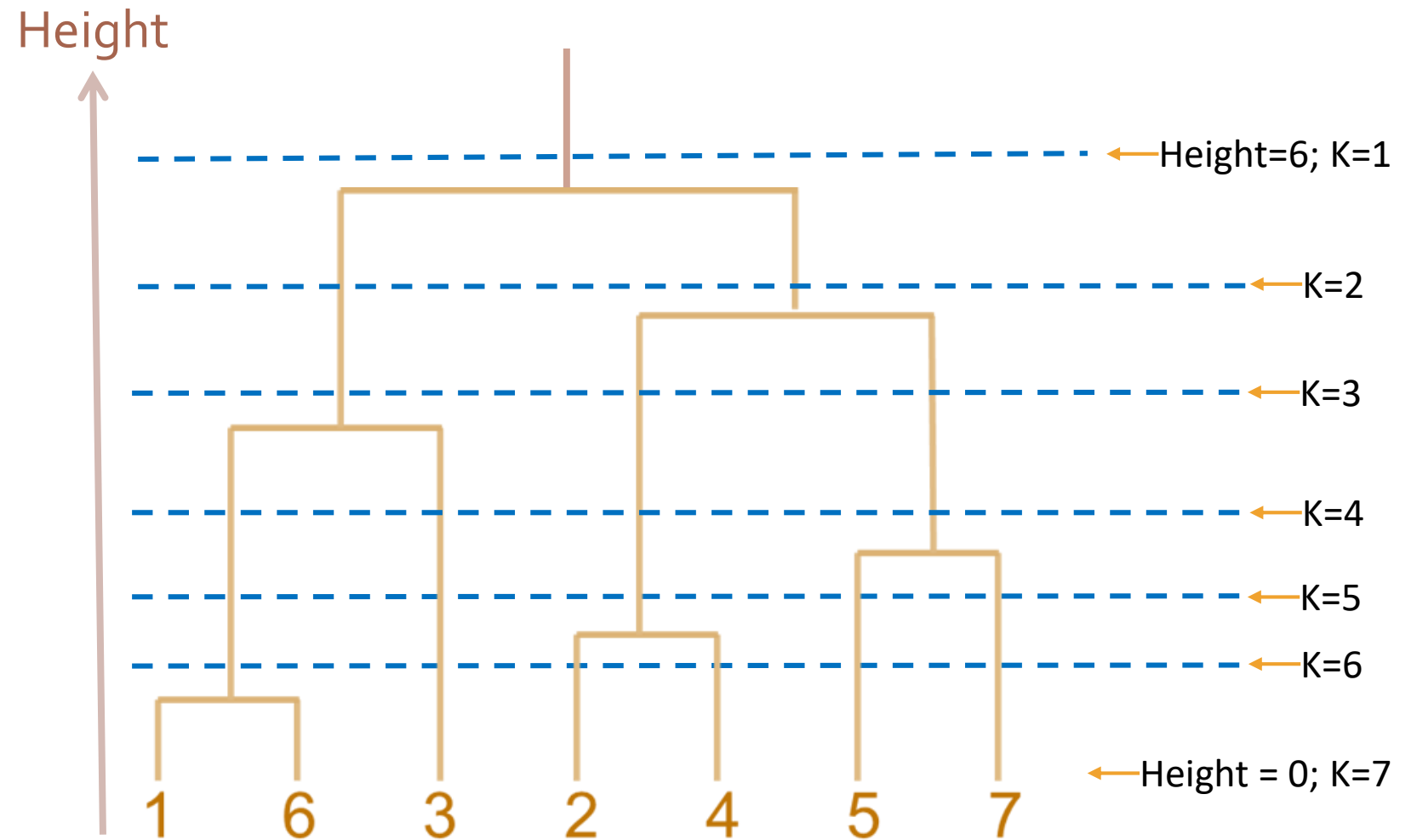
- The root is S
- The leaves are the individual elements of S
- The internal nodes are defined as the union of their children.

Dendrogram: Definition



- Height at leaf: 0
- May be cut at any level: Each connected component forms a cluster.
- Any height represents a horizontal cut and a partition of S into several (nested) clusters.

Hierarchical clustering



Hierarchical Clustering

Machine Learning Unit 23

Sudeshna Sarkar

Centre of Excellence in Artificial Intelligence

Indian Institute of Technology Kharagpur

Hierarchical Agglomerative clustering

- Start with a collection of m singleton clusters $c_i = \{x_i\}$
- Compute the *distance matrix* between the clusters.
- Repeat
 - Merge the two closest clusters c_i, c_j into a new cluster c_{i+j}
 - **Update the distance matrix:** Remove c_i, c_j from the collection and add c_{i+j}

Until only a single cluster remains.

Different definitions of the distance leads to different algorithms.

Distance Measures

Real variables

- **Euclidean**
- **Manhattan**
- **Cosine**
- Minkowski
- Mahalanobis
- ...

Discrete variables

- **Hamming**
- **Jaccard**
- ...

Distance / Similarity Measures

Real Variables

1. **Euclidean or L2** : straight-line distance between two points in Euclidean space.

$$d_{Euclidean}(x_1, x_2) = \|x_1 - x_2\|_2 = \sqrt{\sum_i (x_1^i - x_2^i)^2}$$

2. **Manhattan or L1** : sum of the absolute value of the difference between the dimensions.

$$d_{Manhattan}(x_1, x_2) = \|x_1 - x_2\|_1 = \sum_i |x_1^i - x_2^i|$$

3. **Cosine**

$$d_{Cosine}(x_1, x_2) = 1 - \frac{x_1 \cdot x_2}{\|x_1\|_2 \|x_2\|_2}$$

Distance / Similarity Measures

Discrete Variables

1. Hamming :is the number of positions in which the attribute values differ.
2. Jaccard Similarity:

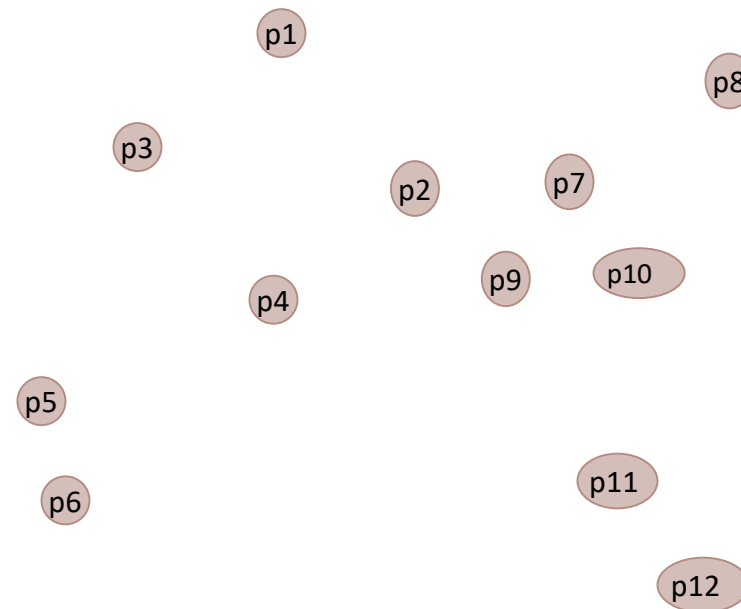
$$J(x_1, x_2) = \frac{|x_1 \cap x_2|}{|x_1 \cup x_2|}$$

Linkage: Definition

- Given distances d_{ij} between data points X_i and X_j
- Given two clusters $c_i = \{\dots\}$ and $c_j = \{\dots\}$
- Compute distance $d(c_i, c_j)$ between c_i and c_j using distances between their contained data points

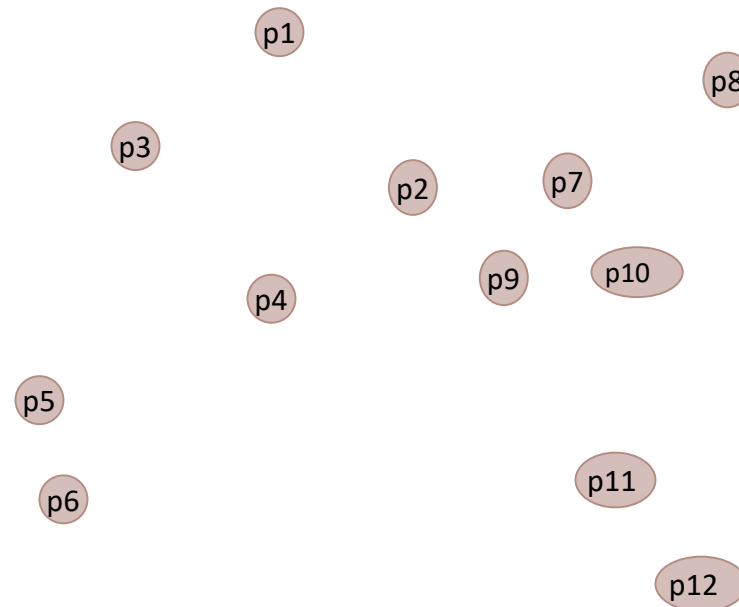
Initialization

- Each individual point is taken as a cluster
- Construct distance/proximity matrix



Initialization

- Each individual point is taken as a cluster
- Construct distance/proximity matrix

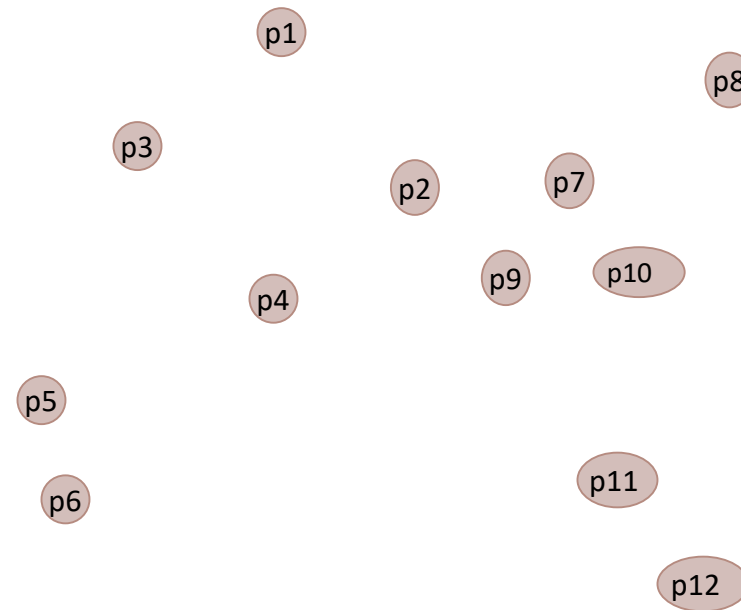


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance/Proximity Matrix

Initialization

- Each individual point is taken as a cluster
- Construct distance/proximity matrix

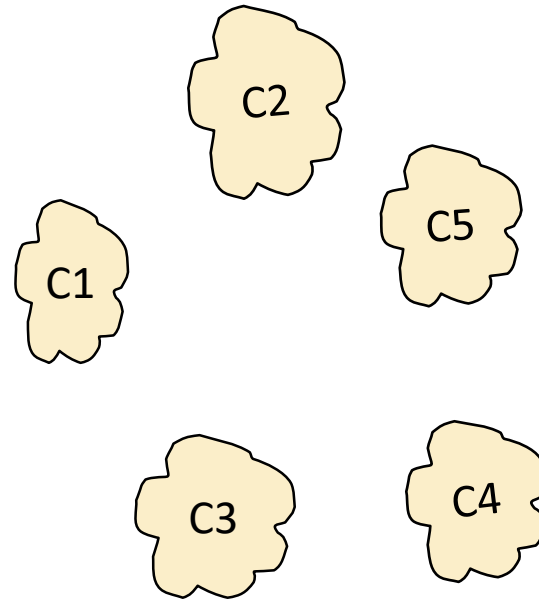


Distance/Proximity Matrix

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

Intermediate State

After some merging steps, we have some clusters

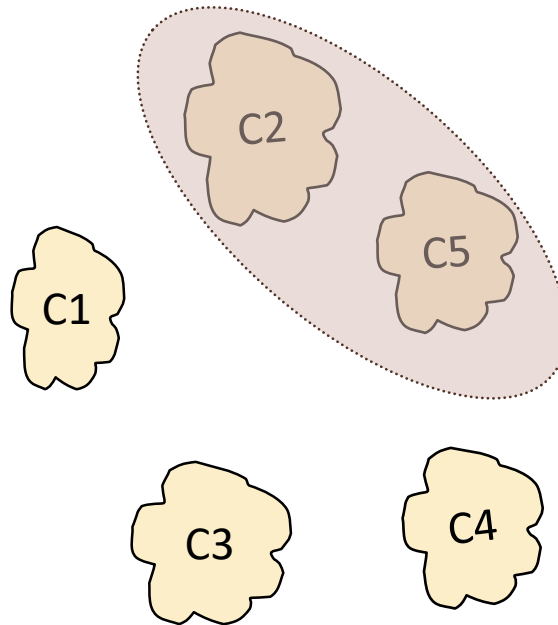


Distance/Proximity Matrix

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Intermediate State

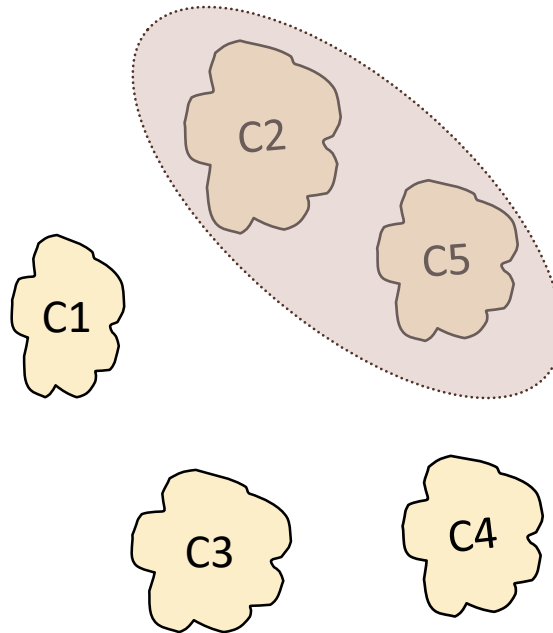
Merge the two closest clusters (C2 and C5) and update the distance matrix.



	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Intermediate State

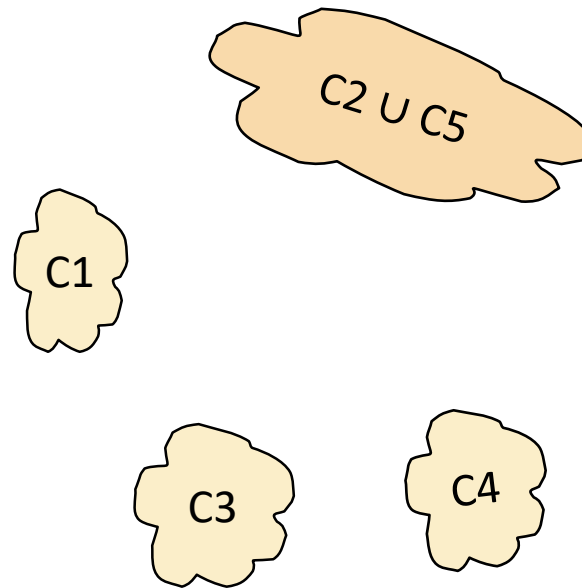
Merge the two closest clusters (C2 and C5) and update the distance matrix.



	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

After Merging

Update the distance matrix



		$C2 \cup C5$			
		C1		C3	C4
C1			?		
$C2 \cup C5$?	?	?	?
C3			?		
C4			?		

Closest Pair

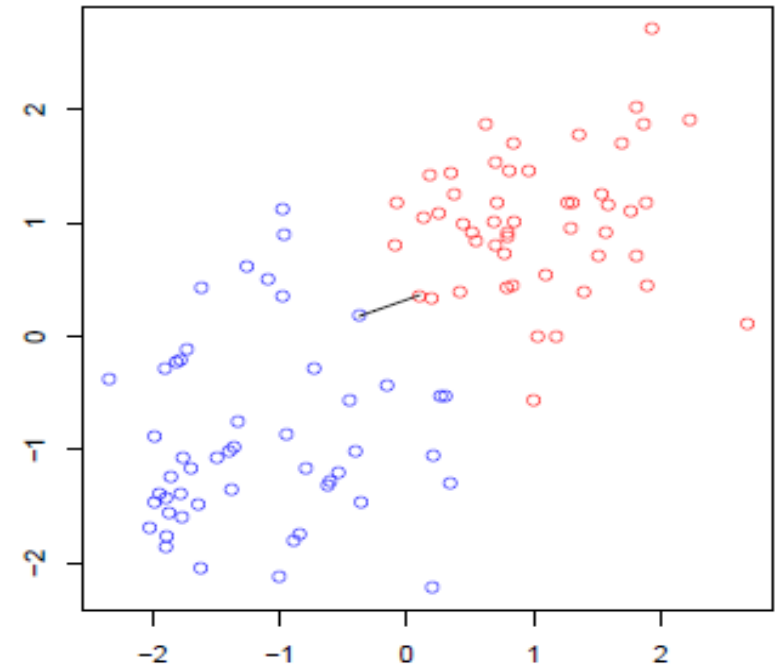
- A few ways to measure distances of two clusters.
- **Single-link**
 - Similarity of the *most* similar (single-link)
- **Complete-link**
 - Similarity of the *least* similar points
- **Centroid**
 - Clusters whose centroids (centers of gravity) are the most similar
- **Average-link**
 - Average cosine between pairs of elements

Distance between two clusters

Single-link distance between clusters C_i and C_j is the *minimum distance* between any object in C_i and any object in C_j

$$\text{sim}(C_i, C_j) = \max_{x \in C_i, y \in C_j} \text{sim}(x, y)$$

It can result in long and thin clusters.

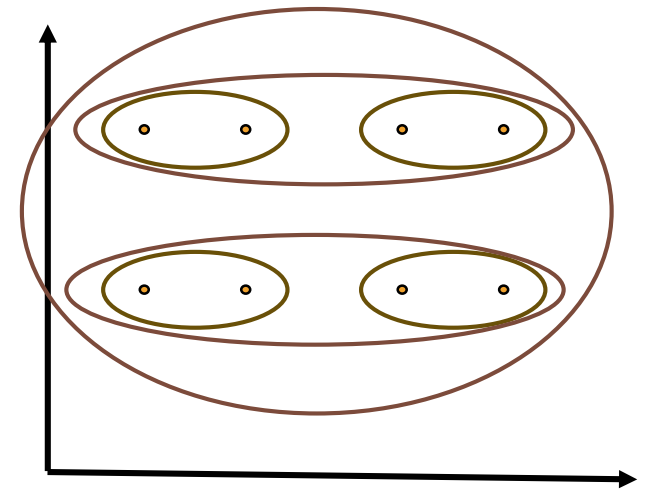


Distance between two clusters

- **Single-link** distance between clusters C_i and C_j is the *minimum distance* between any object in C_i and any object in C_j

$$\text{sim}(C_i, C_j) = \max_{x \in C_i, y \in C_j} \text{sim}(x, y)$$

Determined by one pair of points, i.e., by one link in the proximity graph.



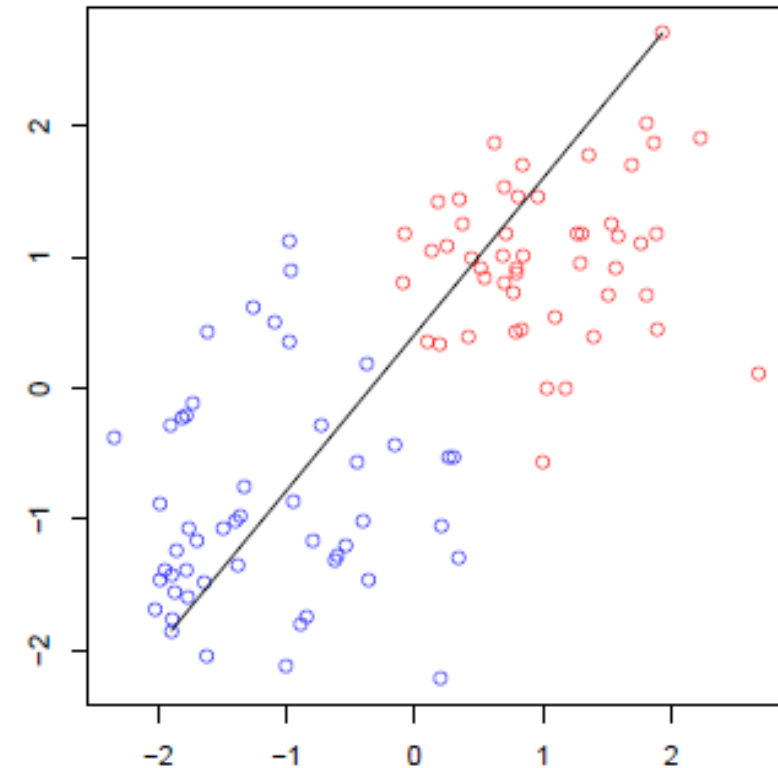
It can result in long and thin clusters.

Complete link method

- The distance between two clusters is the distance of two furthest data points in the two clusters.

$$\text{sim}(C_i, C_j) = \min_{x \in C_i, y \in C_j} \text{sim}(x, y)$$

- Makes “tighter,” spherical clusters that are typically preferable.

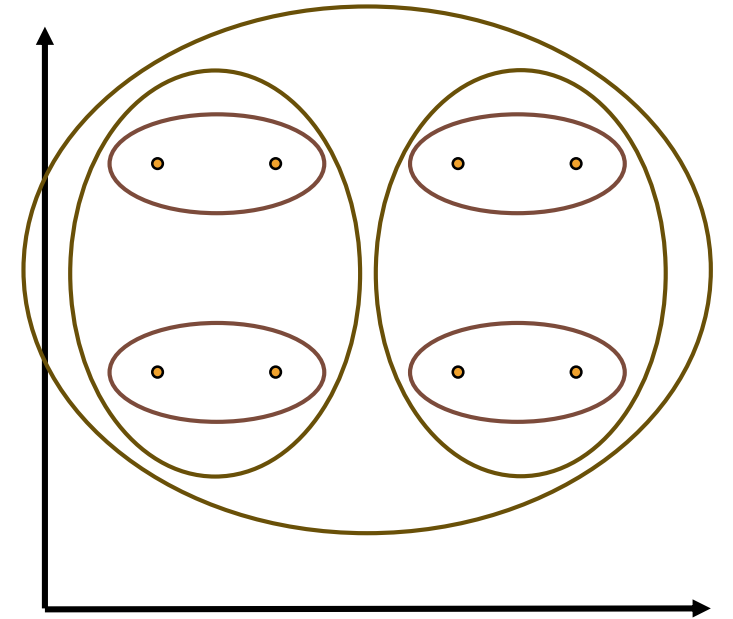


Complete link method

- The distance between two clusters is the distance of two furthest data points in the two clusters.

$$\text{sim}(C_i, C_j) = \min_{x \in C_i, y \in C_j} \text{sim}(x, y)$$

- Makes “tighter,” spherical clusters that are typically preferable.
- Distance between clusters is determined by the two most distant points in the different clusters

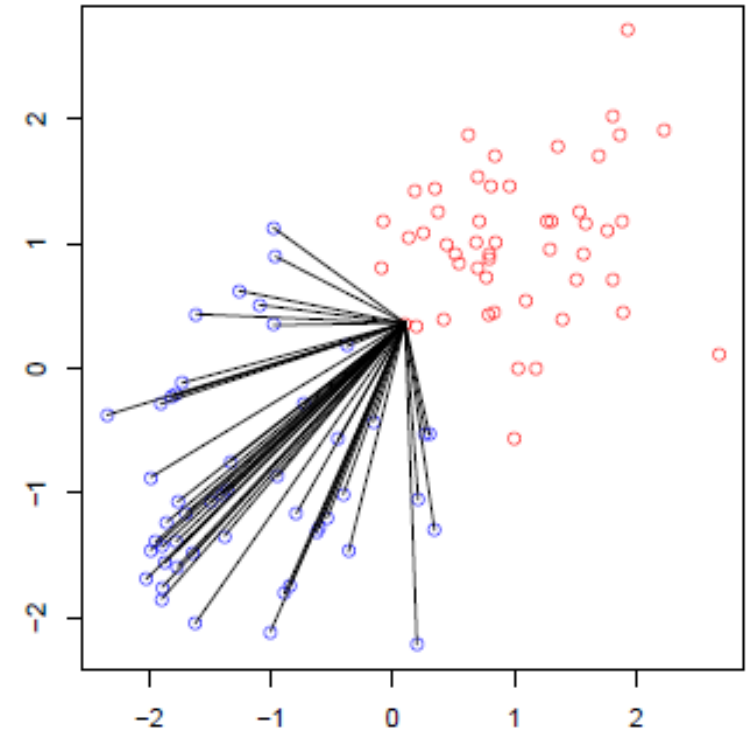


Average Link Clustering

- Similarity of two clusters = average similarity between any object in C_i and any object in C_j

$$\begin{aligned} & \text{sim}(c_i, c_j) \\ &= \frac{1}{|c_i||c_j|} \sum_{x \in c_i} \sum_{y \in c_j} \text{sim}(x, y) \end{aligned}$$

- Two options:
 - Averaged across all ordered pairs in the merged cluster
 - Averaged over all pairs *between* the two original clusters



Compromise between single and complete link. Less susceptible to noise and outliers.

Computational Complexity

- In the first iteration, all HAC methods need to compute similarity of all pairs of m initial instances which is $O(m^2)$
- In $m - 2$ merging iterations, it has to compute the distance between the most recently created cluster and all other existing clusters.
- Single link can be done in $O(m^2)$
- Complete and average links can be done in $O(m^2 \log m)$

Complexity Explained: Single-link clustering:

- While computing distance matrix, find the smallest distance for each data point and keep them in a next-best-merge array.
- In each of the merging steps, find the smallest distance in the next-best-merge array.
- Merge the two identified clusters, and update the distance matrix in $O(m)$.
- Finally, update the next-best-merge array in $O(m)$ in each step. If the best merge partner for k before merging i and j was either i or j , then after merging i and j the best merge partner for k is the merger of i and j .

Complexity: Complete-link clustering

- Compute the distance matrix
- Sort the distances for each data point $O(m^2 \log m)$
- After each merge iteration, update the distances in $O(m)$.
- We pick the next pair to merge by finding the smallest distance that is still eligible for merging. Do this by traversing the m sorted lists of distances using m^2 traversal steps.

Complexity: Average-link clustering

- Compute the distance matrix
- Sort the distances for each data point $O(m^2 \log m)$
- After each merge iteration, update the distances in $O(m)$.
- We pick the next pair to merge by finding the smallest distance that is still eligible for merging. Do this by traversing the m sorted lists of distances using m^2 traversal steps.