Unsupervised Learning

Dr. Dinesh Kumar Vishwakarma

Associate Professor,

Department of Information Technological University,
Delhi-110042

Outline

- **Basic concepts**
- **K-means algorithm**
- **Representation of clusters**
- **Hierarchical clustering**
- **Which clustering algorithm to use?**
- **Cluster evaluation**
- **Summary**

Supervised vs. unsupervised learning

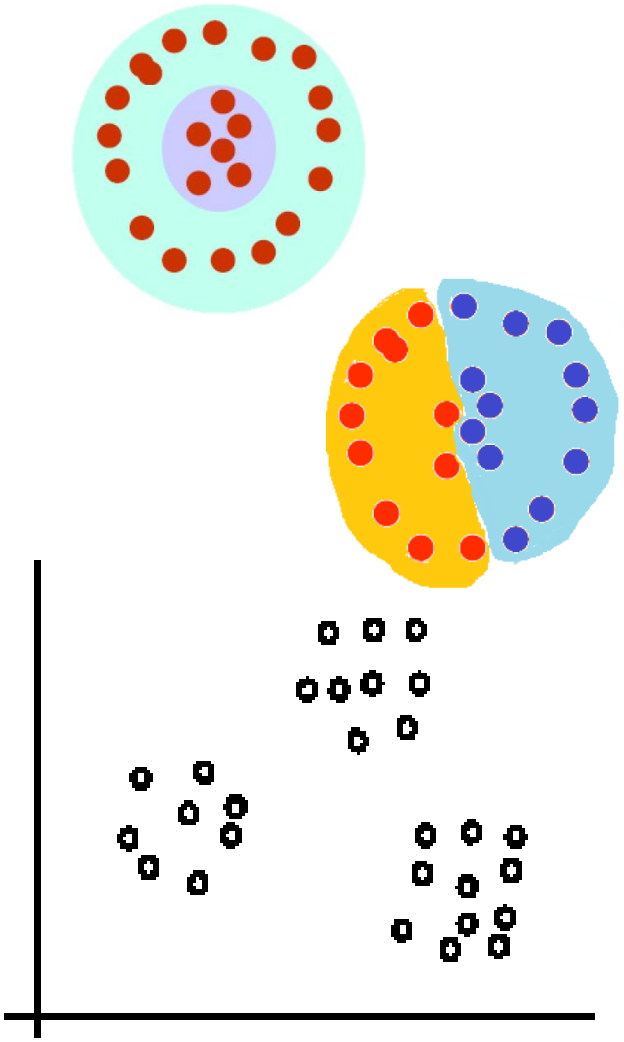
- **Supervised learning: discover patterns in the data that relate data attributes with a target (class) attribute.**
  - ✓ These patterns are then utilized to predict the values of the target attribute in future data instances.
- **Unsupervised learning: The data have no target attribute.**
  - ✓ We want to explore the data to find some intrinsic structures in them.

Clustering

- **Clustering is a technique for finding similarity groups in data, called clusters. i.e.,**
  - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- **Clustering is often called an unsupervised learning task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.**
- **In fact, association rule mining is also unsupervised.**

What is Clusters?

- The organization of unlabeled data into similarity groups called **'clusters'.**
- A cluster is a collection of data items which are **"similar"** between them, & **"dissimilar"** to data items in other clusters.
- The data set has three natural groups of data points, i.e., 3 natural clusters.



What is clustering for?

- **Example 1: groups people of similar sizes together to make "small", "medium" and "large" T-Shirts.**
  - Tailor-made for each person: too expensive
  - One-size-fits-all: does not fit all.
- **Example 2: In marketing, segment customers according to their similarities**
  - To do targeted marketing.
  - Residential Area

What is clustering for? (cont…)

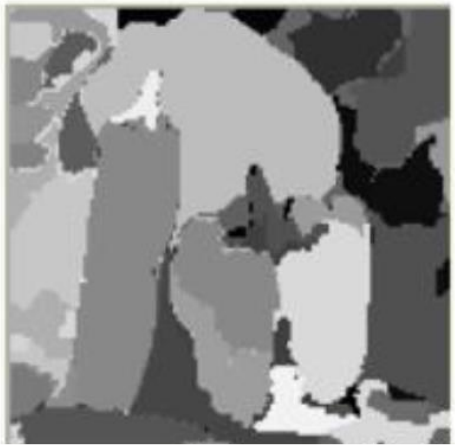
- **Example 3: Given a collection of text documents, we want to organize them according to their content similarities,**
  - To produce a topic hierarchy
- **In fact, clustering is one of the most utilized data mining techniques.**
  - It has a long history, and used in almost every field, e.g., **medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries**, etc.
  - In recent years, due to the rapid increase of **online** documents, **text clustering** becomes important.

What is clustering for? (cont…)

- **Computer Vision**







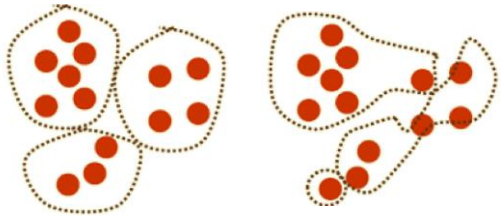
What do we need for Clustering?

- **Proximity measures between two data points (x_i, x_j).**
  - Similarity measures $\mathbf{s}(x_i, x_j)$: large if $x_i, and\ x_j$ are similar.
  - dissimilarity measures (distance) $\mathbf{d}(x_i, x_j)$: small if $x_i, and\ x_j$ are similar

Large **d**, & small **s**

small **d**, & large **s**

- **Criteria Function to evaluate a clustering**



Good

Bad

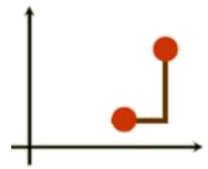
Distance Measurement

- Consider two data points (x_i, x_j)

x_i ——— x_j

- Distance between two points can be defined as **Minkowski Distance.**
  - $d_p(x_i, x_j) = \left[\sum_{k=1}^{m} |x_{ik} - x_{jk}|^p\right]^{\frac{1}{p}}$, where p is positive integer.
- Euclidian Distance when p=2
  - $d(x_i, x_j) = \sqrt{\sum_{k=1}^{m} \left(x_i^{(k)} - x_j^{(k)}\right)^2}$ ***Translation Invariant***

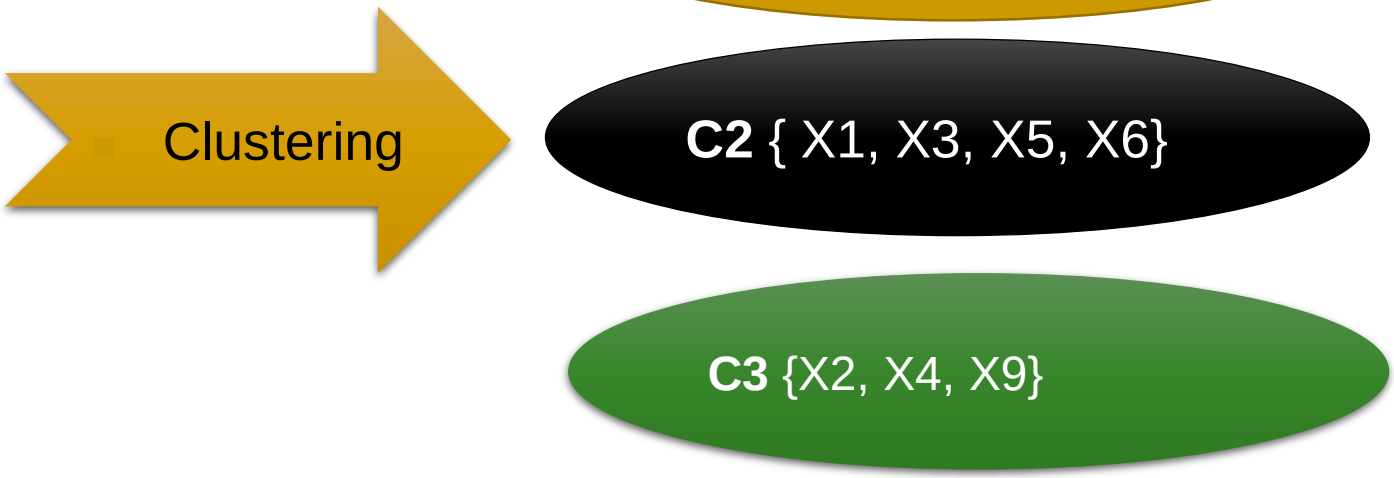
- Manhattan (City Block) Distance when p=1
  - $d(x_i, x_j) = \sum_{k=1}^{m} \left|x_i^{(k)} - x_j^{(k)}\right|$, ***Cheaper to compute.***



Clustering E.g.

- **Consider a data table**

X_1
X_2
X_3
X_4
X_5
X_6
X_7
X_8
X_9



Aspects of Clustering

- **A clustering algorithm**
  - Partitional clustering
  - Hierarchical clustering
- **A distance (similarity, or dissimilarity) function**
- **Clustering quality**
  - Inter-clusters distance $\Rightarrow$ maximized
  - Intra-clusters distance $\Rightarrow$ minimized
- **The quality of a clustering result depends on the algorithm, the distance function, and the application.**

Hierarchical vs Partitional Clustering

- A distinction among different types of clustering is whether the set of clusters is **'nested'** or **'unnested'.**
- A **partitional clustering** a simply a **division of the set of data objects into non-overlapping subsets** (clusters) such that each data object is in exactly one subset.
- **A hierarchical clustering is a set of nested clusters that are organized as a tree.**

K-means clustering

- **K-means is a 'Partitional Clustering algorithm.**
- **Let the set of data points (or instances) *D* be $\{x_1, x_2, \ldots, x_n\}$,**
  - where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in a real-valued space $X \subseteq R^r$, and r is the number of attributes (dimensions) in the data.
- **The *k*-means algorithm partitions the given data into *k* clusters.**
  - Each cluster has a cluster **center**, called **centroid**.
  - k is specified by the user.

K-means algorithm

- **Given *k*, the *k-means* algorithm works as follows:**

  1) Randomly choose *k* data points (seeds) to be the initial centroids, cluster centers

  2) Assign each data point to the closest centroid.

  3) Re-compute the centroids using the current cluster memberships.

  4) If a convergence criterion is not met, go to 2).

K-means algorithm – (cont …)

- **Algorithm** *k-means (k, D)*

1 Choose k-data points as the initials centroids (cluster centres)

2 repeat

3 **for** each data point $\boldsymbol{x} \in \boldsymbol{D}$ do

4 compute the distance from x to each centroid;

5 assign $\boldsymbol{x}$ to the closet centroid // a centroid represents a cluster;

6 **endfor**

7 re-compute the centroid using the current cluster memberships;

8 untill the stopping criterion is met.

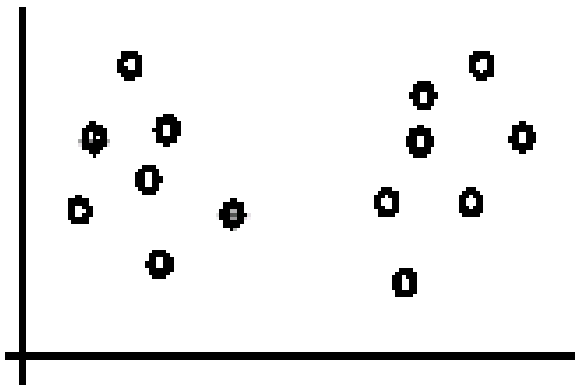
Stopping/convergence criterion

1. No (or minimum) re-assignments of data points to different clusters,
2. No (or minimum) change of centroids, or
3. Minimum decrease in the sum of squared error (SSE),

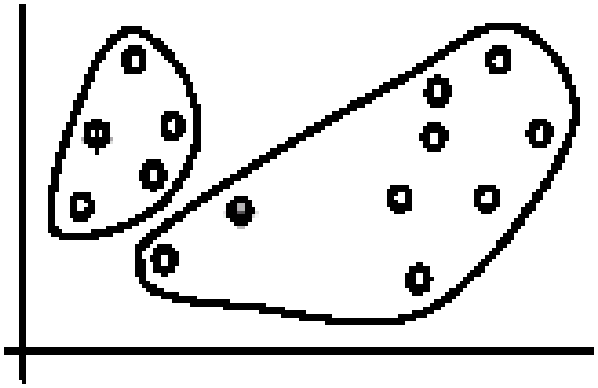
$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2 \qquad (1)$$

- C_j is the j^{th} cluster, m_j is the centroid of cluster C_j (the mean vector of all the data points in C_j), and $dist(x, m_j)$ is the distance between data point x and centroid m_j.

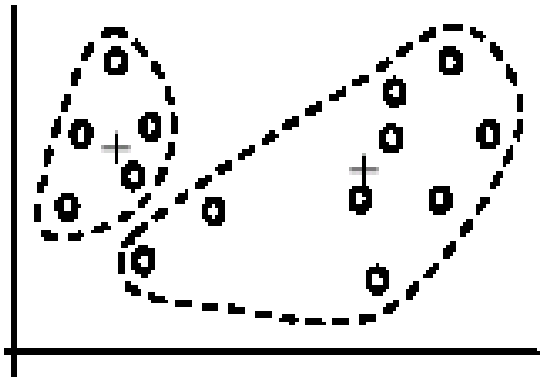
An example



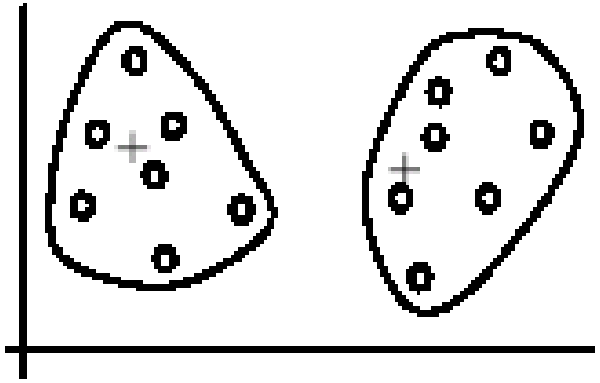
(A) Random selection of k centres



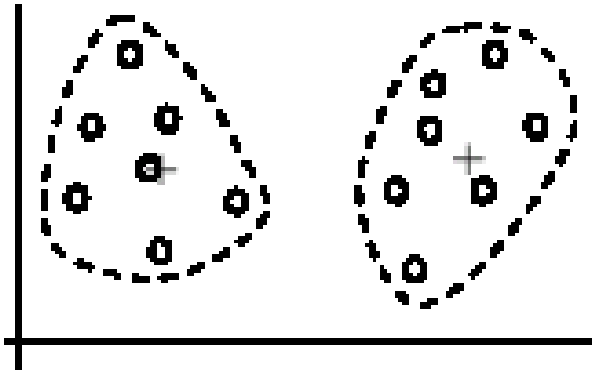
Iteration 1 (B) Cluster assignment



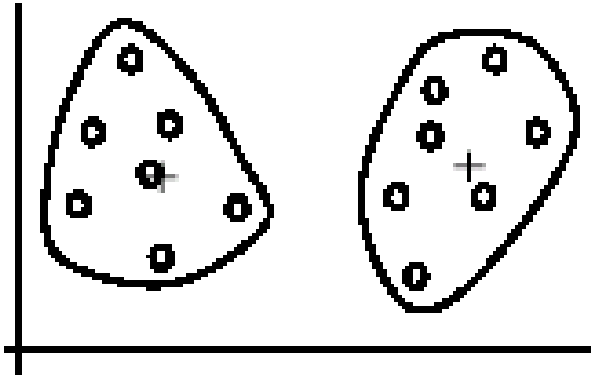
(C) Re-compute centroids



Iteration 2 (D) Cluster assignment



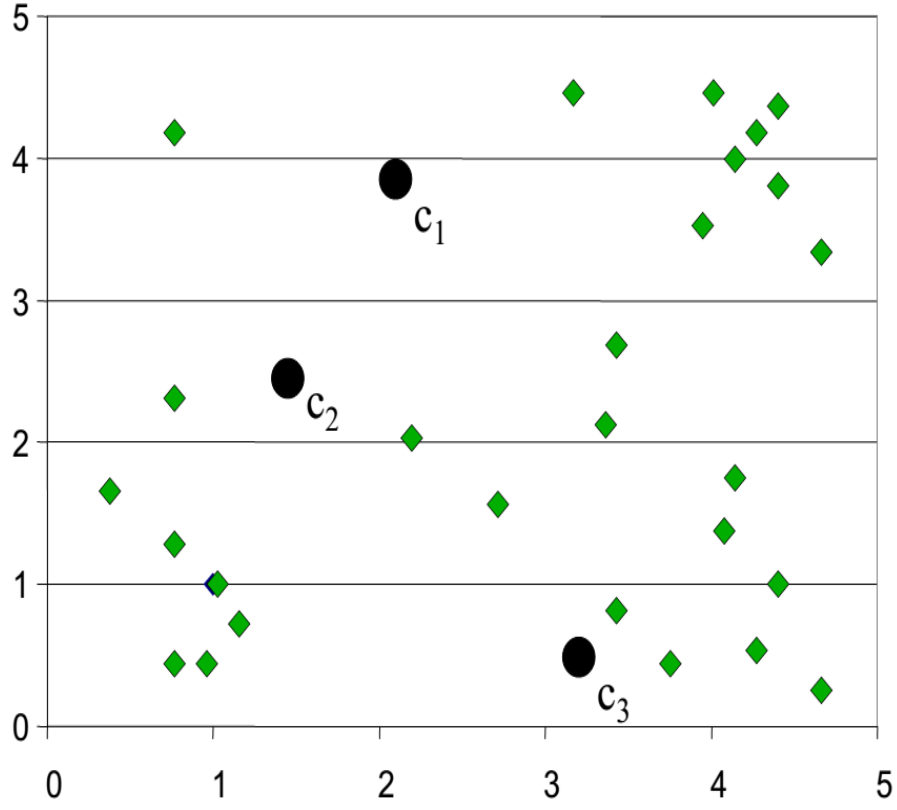
(E) Re-compute centroids



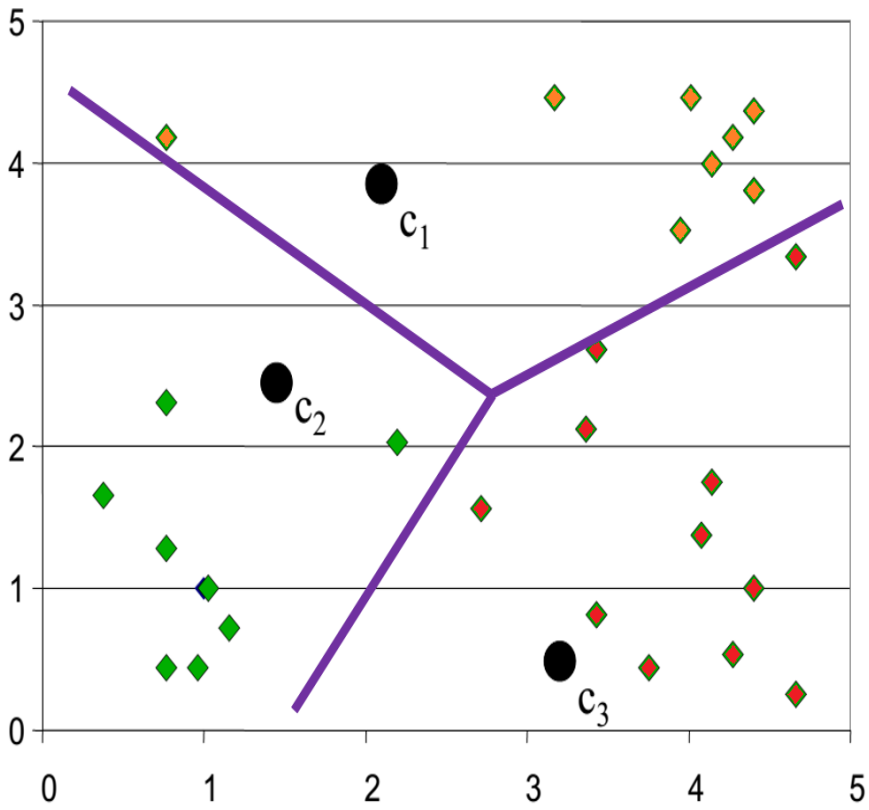
Iteration 3 (F) Cluster assignment

Example

- **Step 1: Randomly initialize cluster centre**

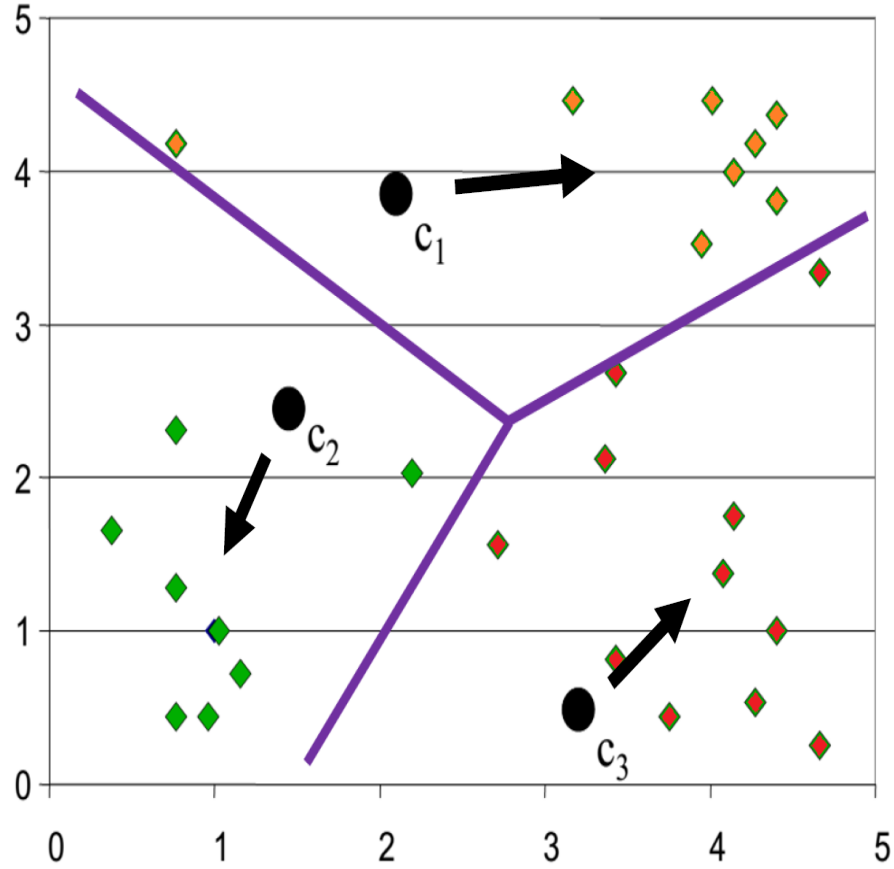


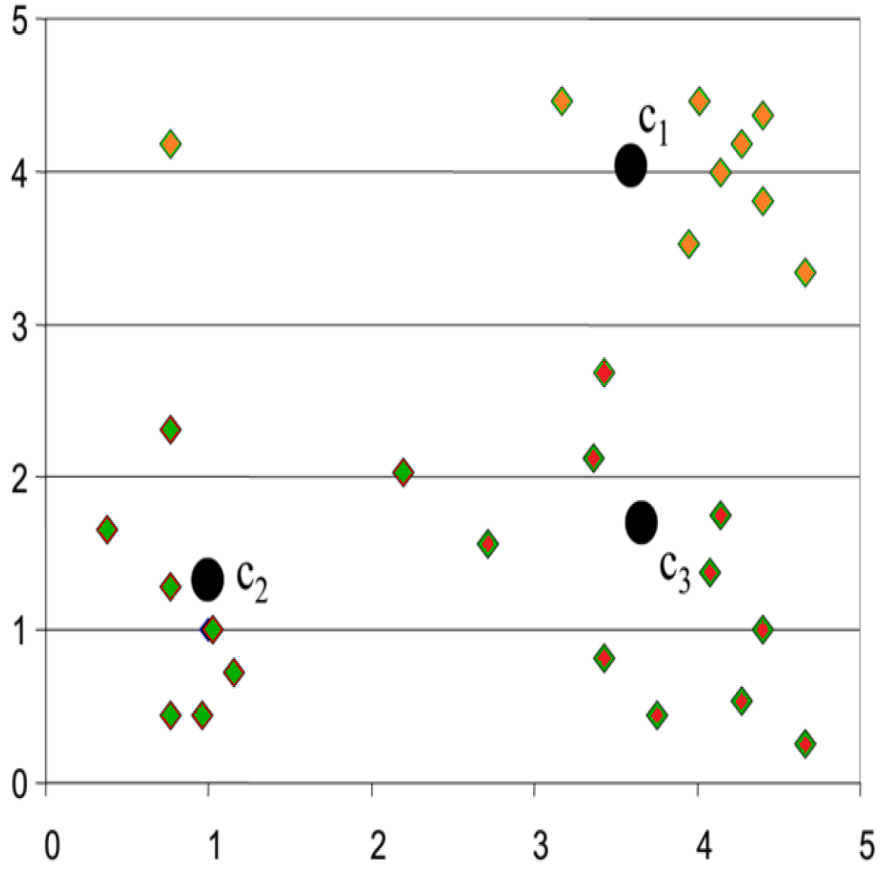
- **Step 2: determine cluster membership to each input**



Example…

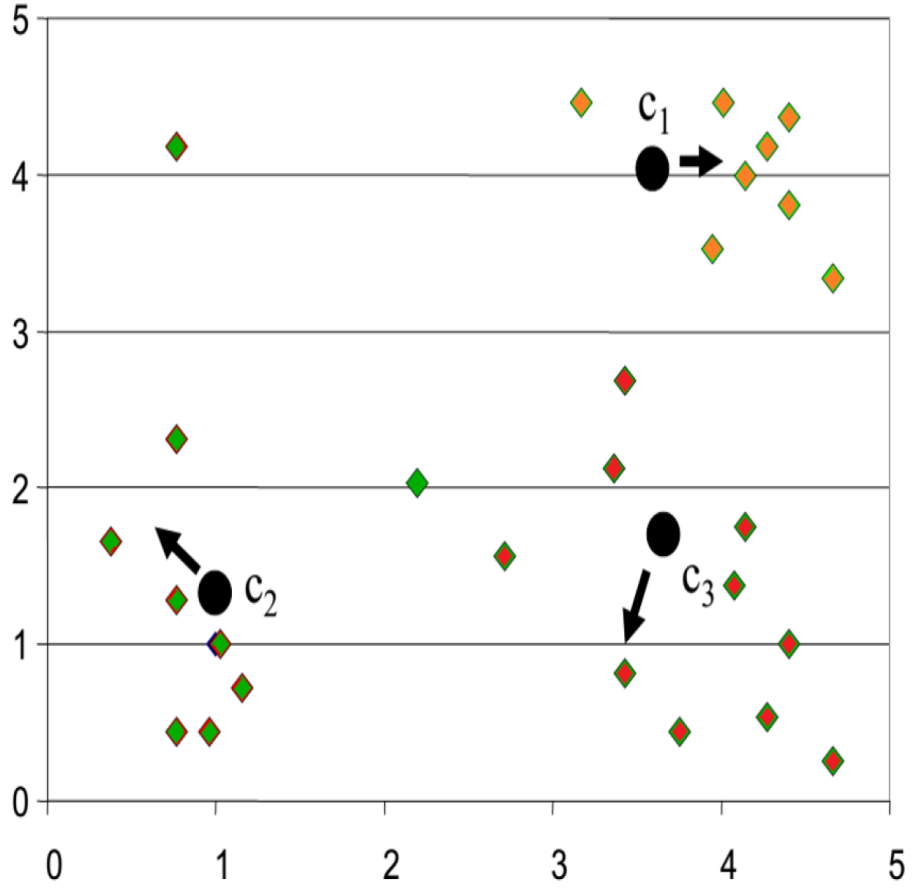
- **Step 3: Re-estimate cluster centre**
- **Result of first iteration**



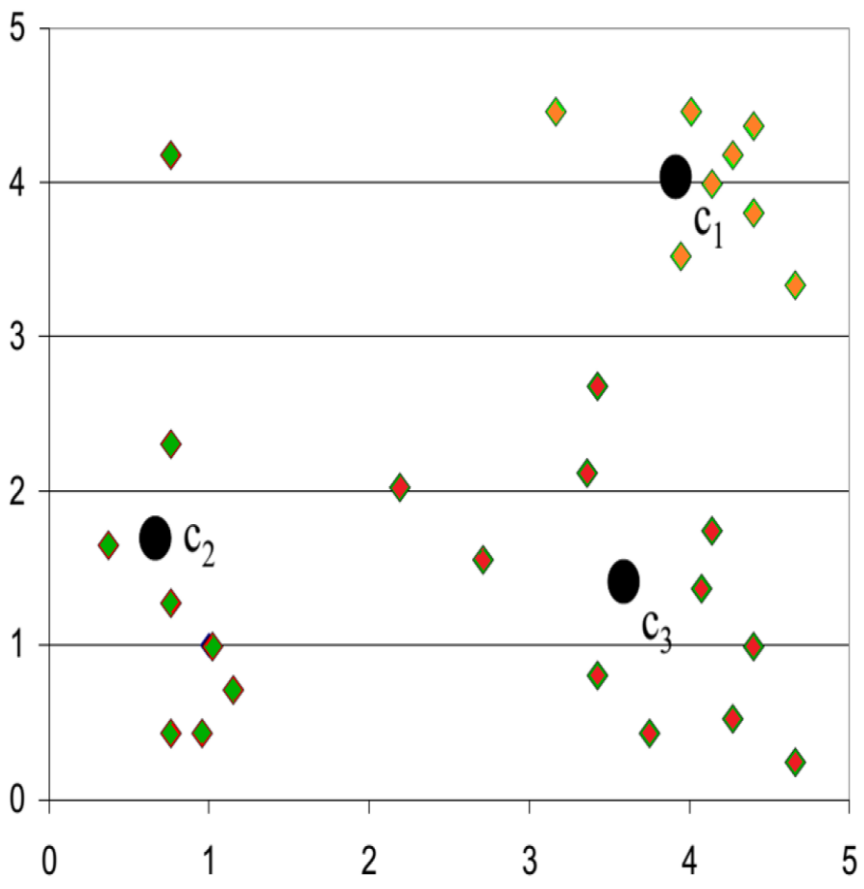


Example…

- **Second iteration**



- **Result of Second iteration**



Example of K-Mean Clustering

- **Consider a data table**

Height (X)	Weight (Y)
185	72
170	56
168	60
179	68
182	72
188	77
180	71
180	70
183	84
180	88
180	67
177	76

- **Consider there are two clusters (K1, K2) and their centroids are (185, 72) & (170, 56)**

K1={185, 72}

K2={170, 56}

- **Euclidian Distance for data point '3'**

$$K1 \rightarrow \sqrt{(168-185)^2+(60-72)^2}=20.80$$

$$K2 \rightarrow \sqrt{(168-170)^2+(60-56)^2}=4.48$$

Hence, data point 3 belongs to K2

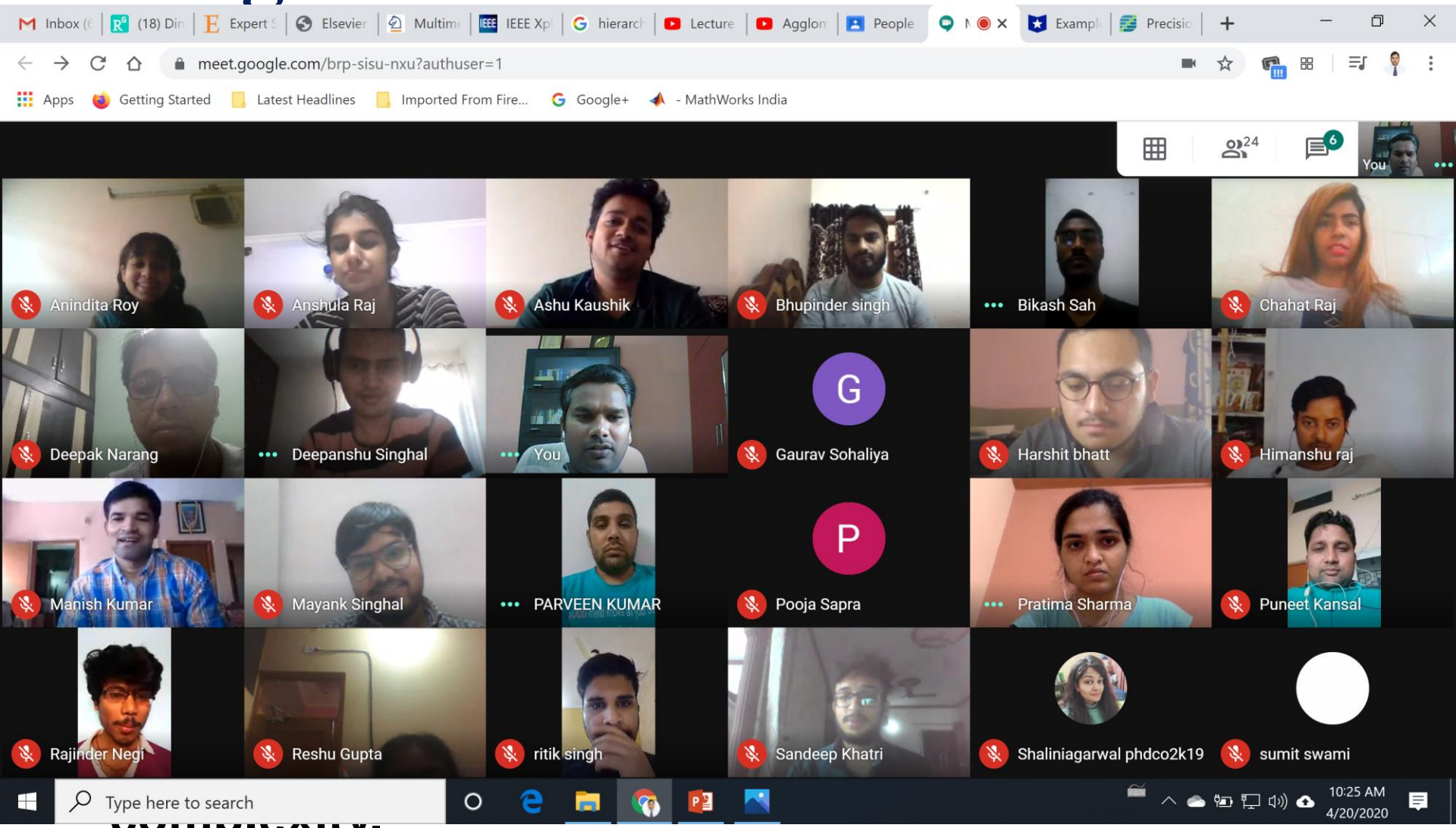
- **New Centroid for K2**

$$K2 \rightarrow \left(\frac{170+168}{2}, \frac{60+56}{2}\right) = (169, 58)$$

$K2\ (2, 3)$

$K1\ (1, 4, 5, 6, 7, 8, 9, 10, 11, 12)$

Strengths of k-means



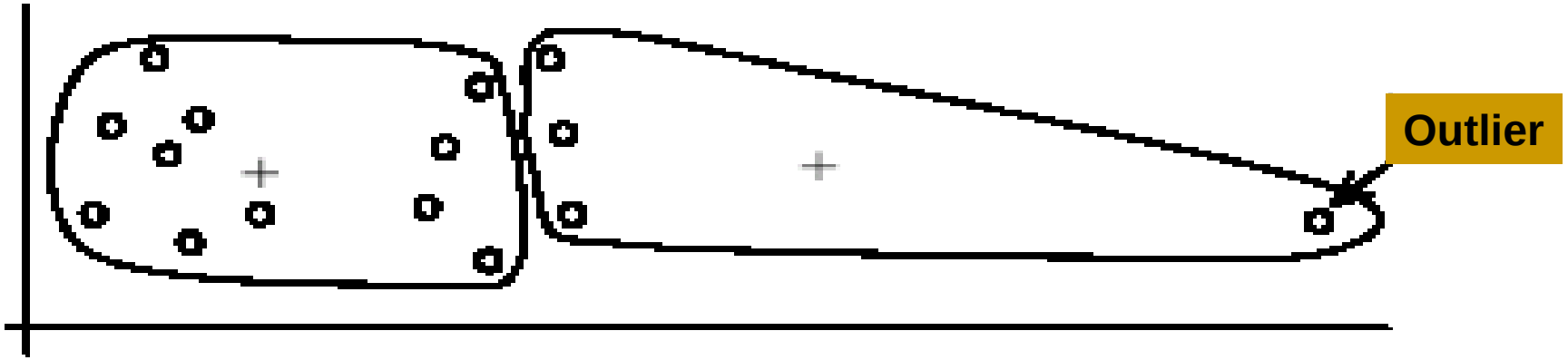
complexity.

Weaknesses of k-means

- **The algorithm is only applicable if the mean is defined.**
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.
- **The user needs to specify *k*.**
- **The algorithm is sensitive to outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Weaknesses of k-means…

- **Problems with outliers**



(A) Undesirable cluster

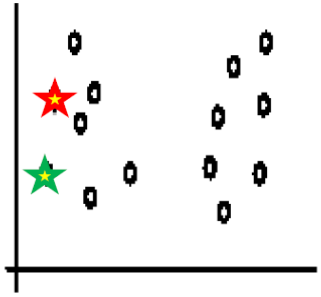
(B) Ideal cluster

Weaknesses of k-means…

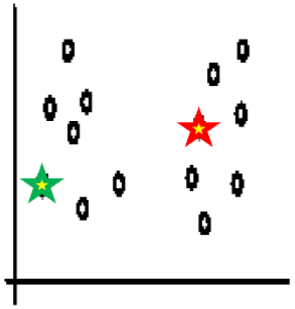
- **To deal with outliers**
- **One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.**
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- **Another method is to perform random sampling. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.**
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

Weaknesses of k-means (cont …)

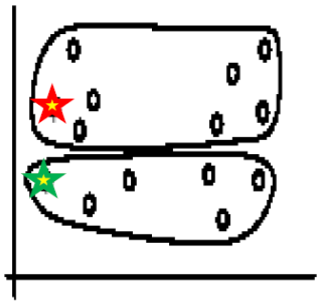
- The algorithm is sensitive to initial seeds.
- If we use different seeds: good results



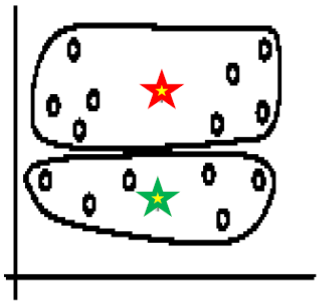
Random selection of seeds (centroids)



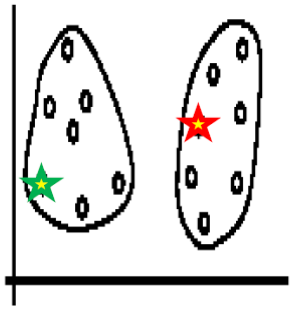
Random selection of seeds (centroids)



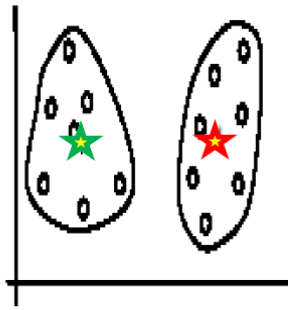
Iteration 1



Iteration 2



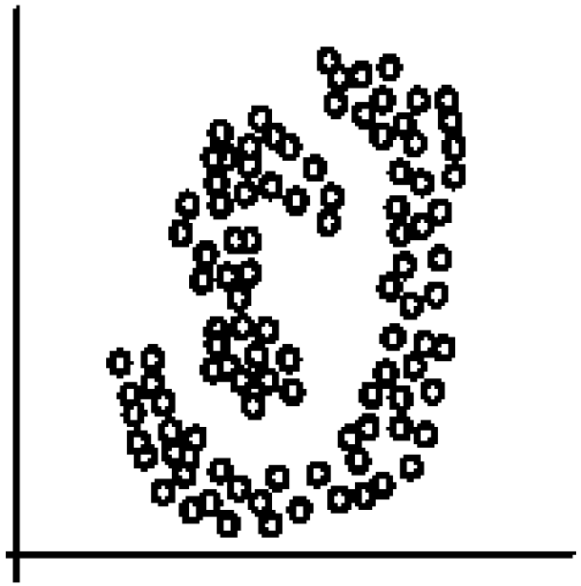
Iteration 1



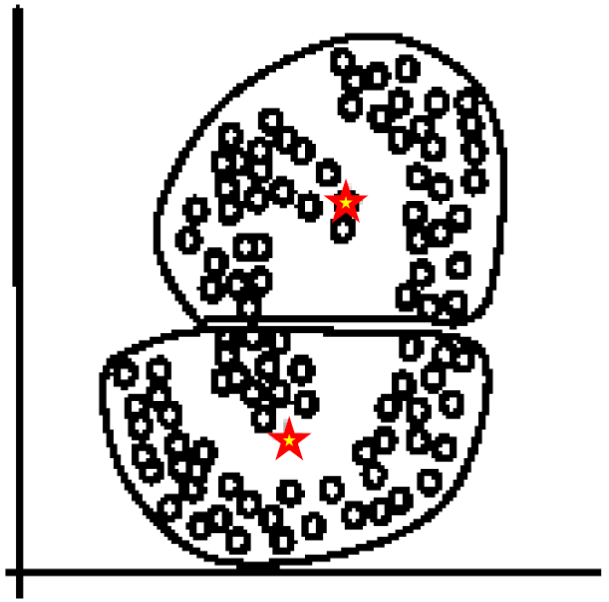
Iteration 2

Weaknesses of k-means (cont …)

- **Special Data Structures:** The k-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



Two natural clusters



K-mean clusters

K-means summary

- **Despite weaknesses, *k*-means is still the most popular algorithm due to its simplicity, efficiency and**
  - other clustering algorithms have their own lists of weaknesses.
- **No clear evidence that any other clustering algorithm performs better in general**
  - although they may be more suitable for some specific types of data or applications.
- **Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!**

Outlines

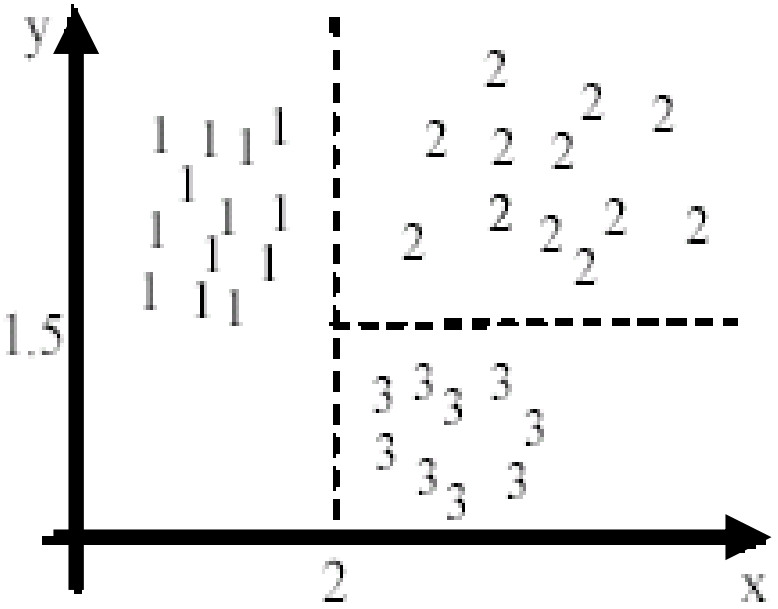
- Basic concepts
- K-means algorithm
- Representation of clusters
- Hierarchical clustering
- Which clustering algorithm to use?
- Cluster evaluation
- Summary

Common ways to represent clusters

- **Use the centroid of each cluster to represent the cluster.**
  - compute the radius and
  - standard deviation of the cluster to determine its spread in each dimension
  - The centroid representation alone works well if the clusters are of the hyper-spherical shape.
  - If clusters are elongated or are of other shapes, centroids are not sufficient

Using classification model

- **All the data points in a cluster are regarded to have the same class label, e.g., the cluster ID.**
  - run a supervised learning algorithm on the data to find a classification model.



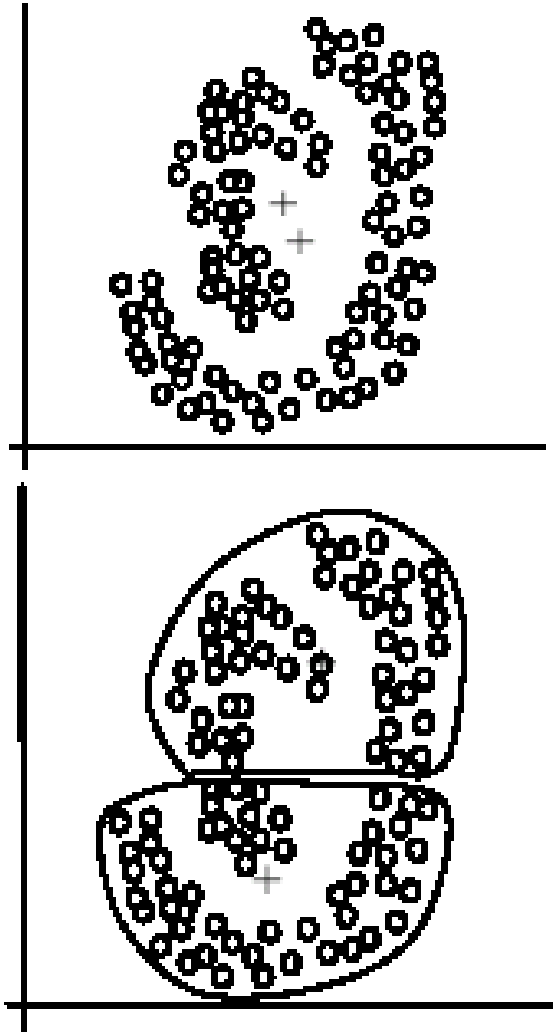
$$x \leq 2 \rightarrow cluster\ 1$$
$$x > 2, y > 1.5 \rightarrow cluster\ 2$$
$$x > 2, y \leq 1.5 \rightarrow cluster\ 3$$

Use frequent values to represent cluster

- This method is mainly for clustering of categorical data (e.g., *k*-**modes** clustering).
- Main method used in text clustering, where a small set of frequent words in each cluster is selected to represent the cluster.

Clusters of arbitrary shapes

- **Hyper-elliptical and hyper-spherical clusters are usually easy to represent, using their centroid together with spreads.**
- **Irregular shape clusters are hard to represent. They may not be useful in some applications.**
  - Using centroids are not suitable (upper figure) in general.
  - K-means clusters may be more useful (lower figure), e.g., for making 2 size T-shirts.

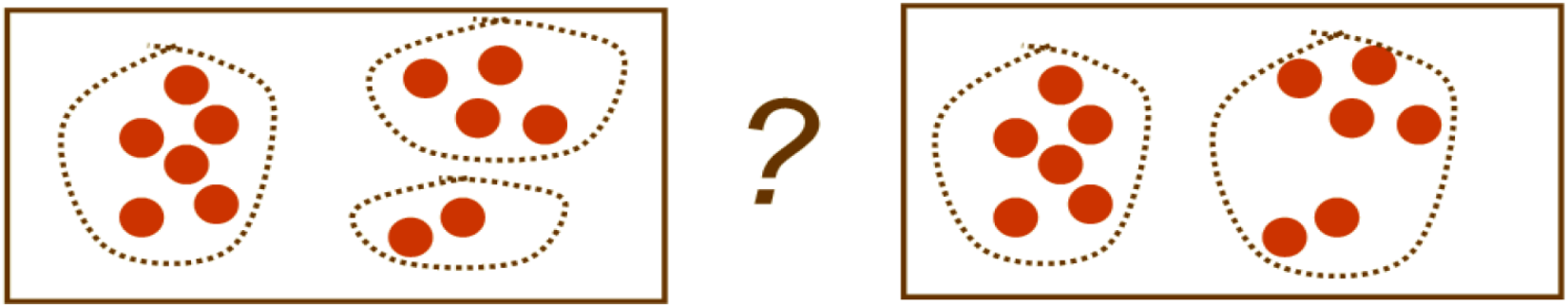


Outlines

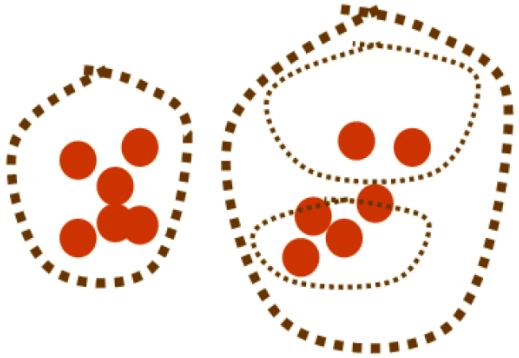
- Basic concepts
- K-means algorithm
- Representation of clusters
- Hierarchical clustering
- Which clustering algorithm to use?
- Cluster evaluation
- Summary

Hierarchical Clustering

- **In previous, a 'flat' clustering is considered.**



- **For some data hierarchical clustering is more appropriate than 'flat'.**

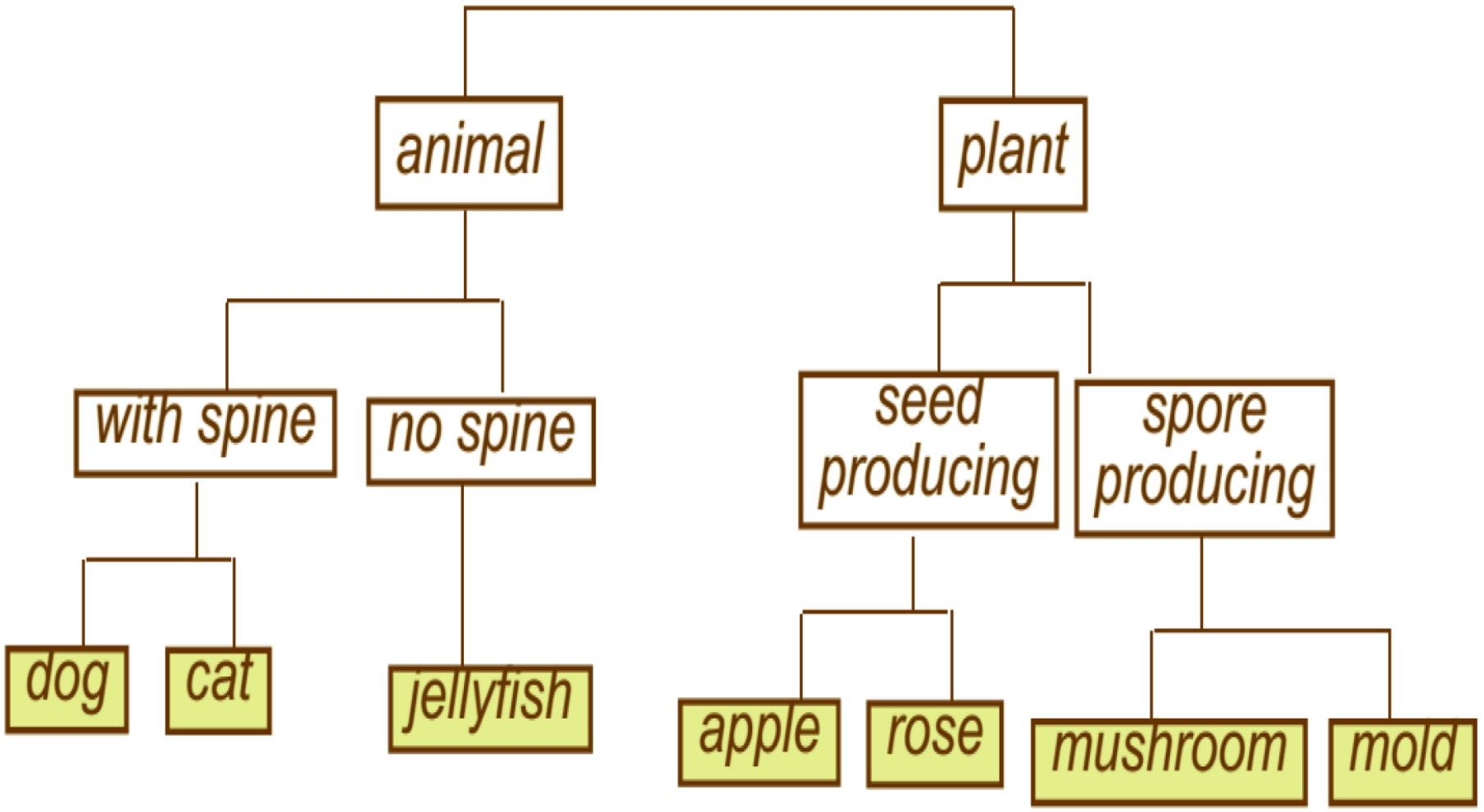


Hierarchical Clustering

Why Hierarchical Clustering?

- It does not assume a particular value of k, as needed by k-means clustering.
- The generated tree may correspond to a meaningful taxonomy.
- Only a distance or **"proximity"** matrix is needed to compute the hierarchical clustering.

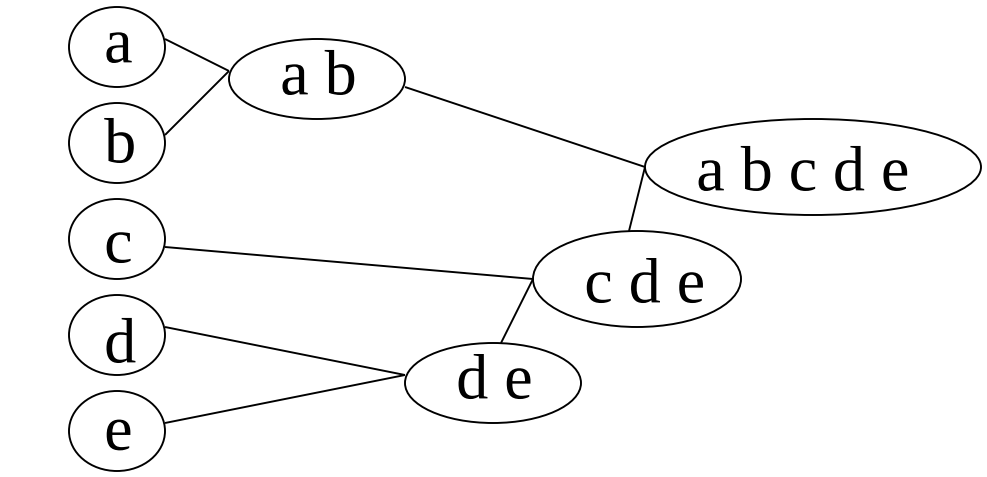
Why Hierarchical Clustering? E.g.



Types of hierarchical clustering

- **Agglomerative (bottom up) clustering: It builds the dendrogram (tree) from the bottom level, and**
  - merges the most similar (or nearest) pair of clusters
  - stops when all the data points are merged into a single cluster (i.e., the root cluster).
- **Divisive (top down) clustering: It starts with all data points in one cluster, the root.**
  - Splits the root into a set of child clusters. Each child cluster is recursively divided further
  - stops when only singleton clusters of individual data points remain, i.e., each cluster with only a single point

Agglomerative approach



Initialization:

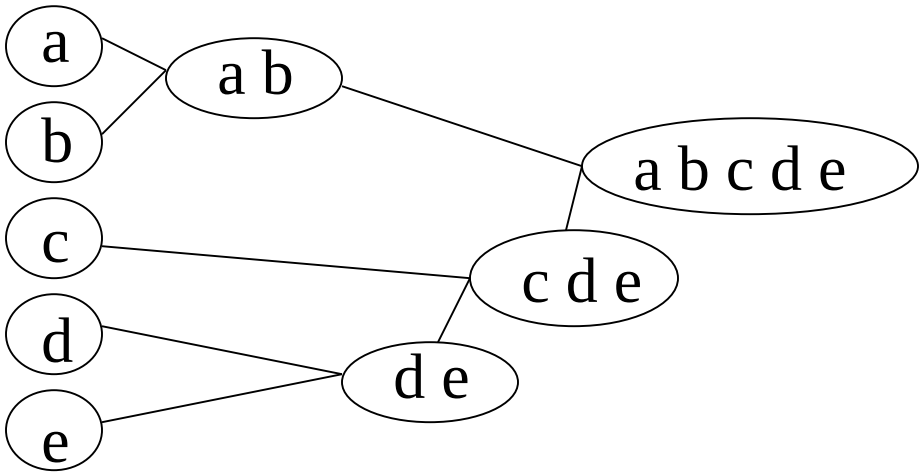
Each object is a cluster

Iteration:

Merge two clusters which are most similar to each other;

Until all objects are merged into a single cluster

Divisive Approaches



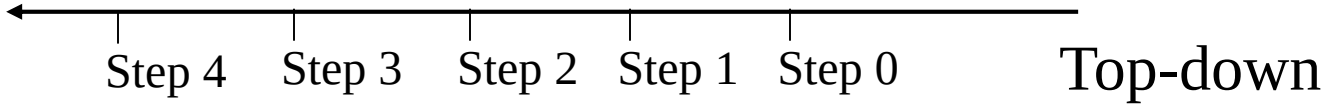
Initialization:

All objects stay in one cluster

Iteration:

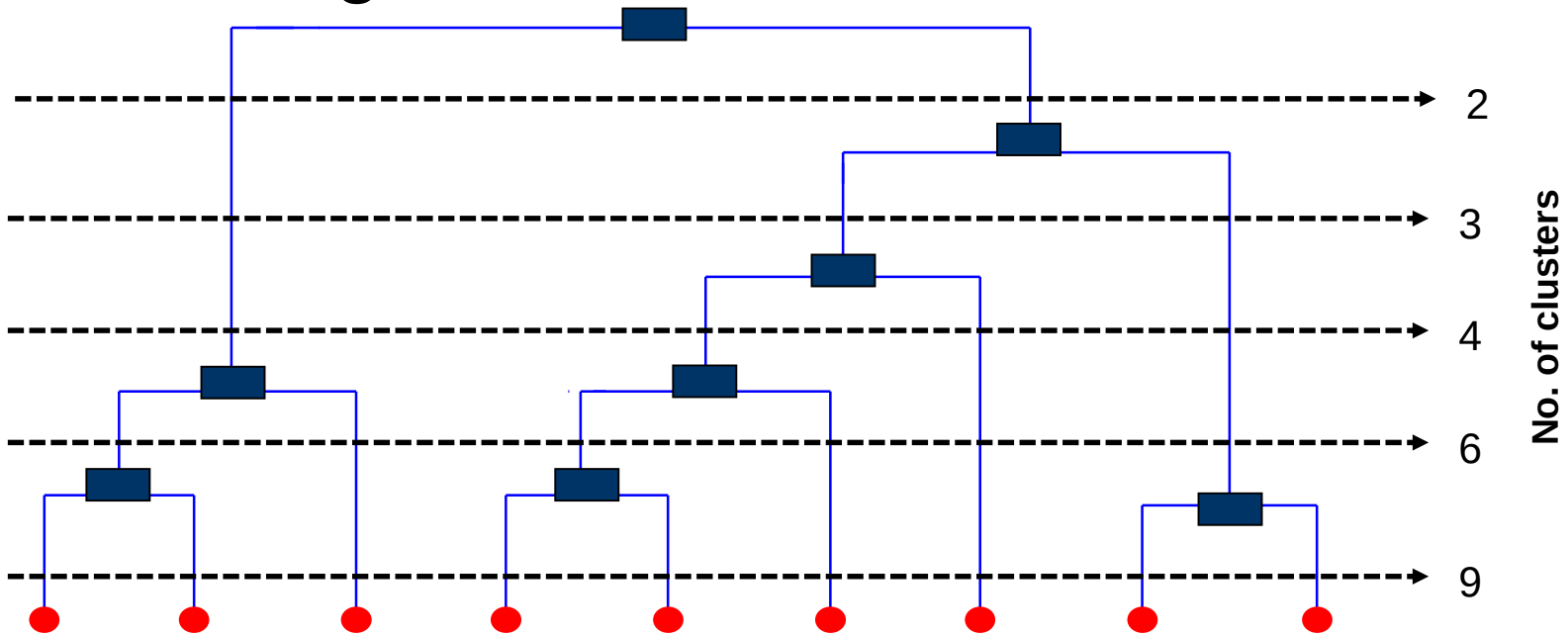
Select a cluster and split it into two sub clusters

Until each leaf cluster contains only one object

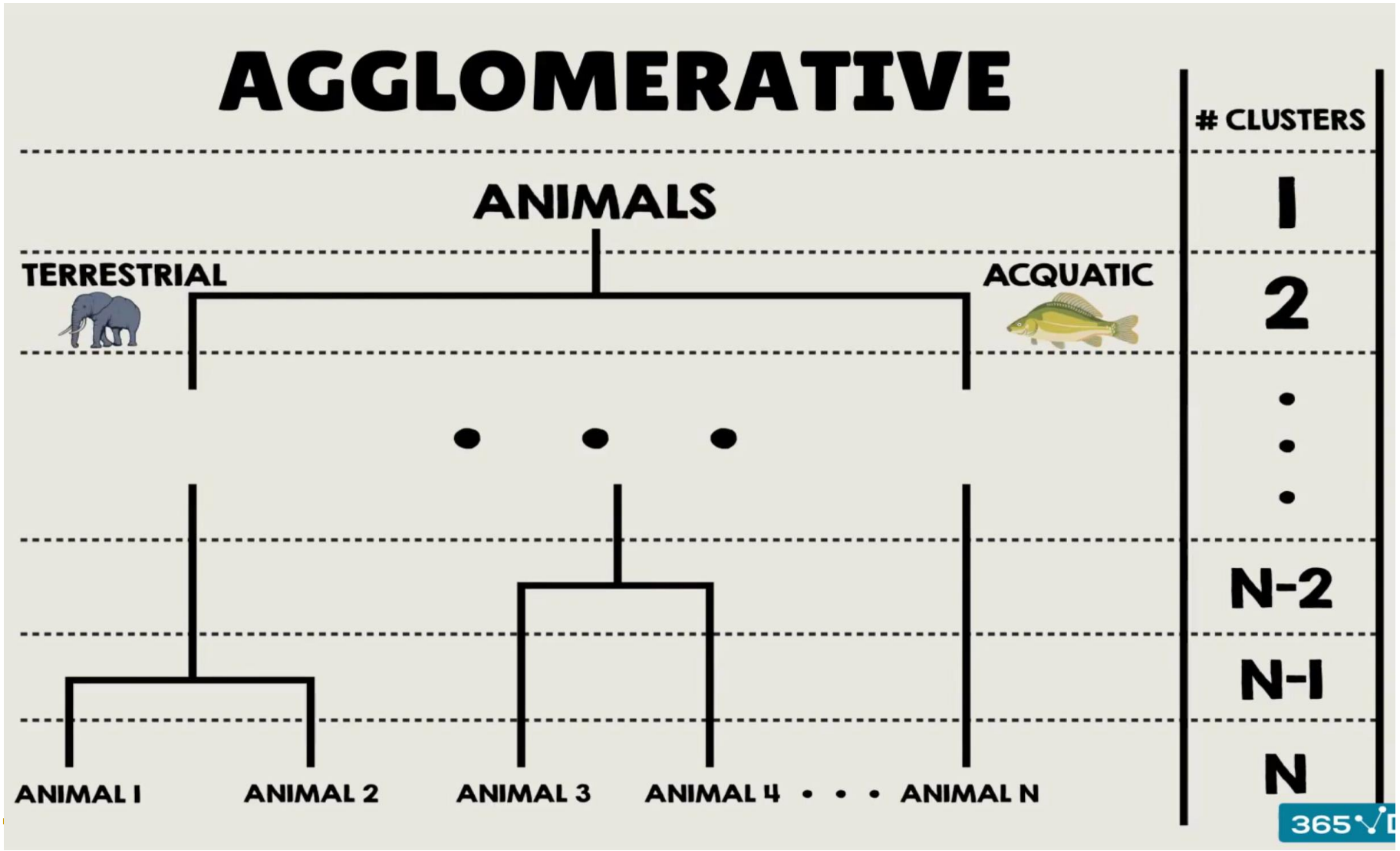


Dendrogram

- A binary tree that shows how clusters are merged/split hierarchically
- Each node on the tree is a cluster; each leaf node is a singleton cluster

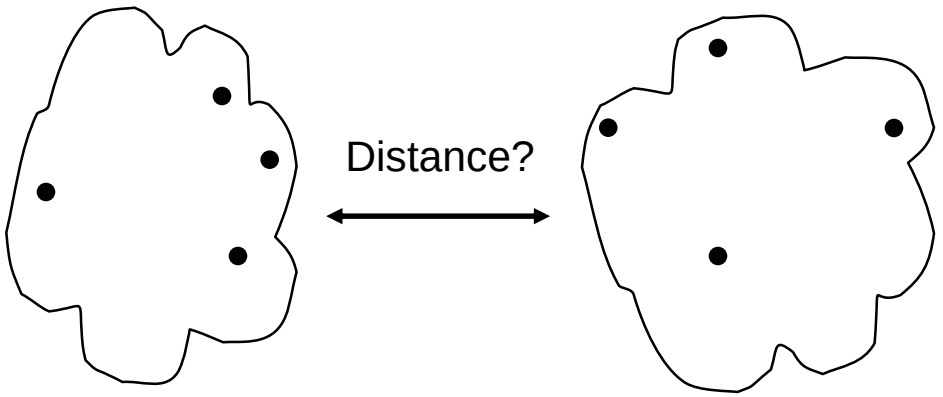


Agglomerative Algorithms



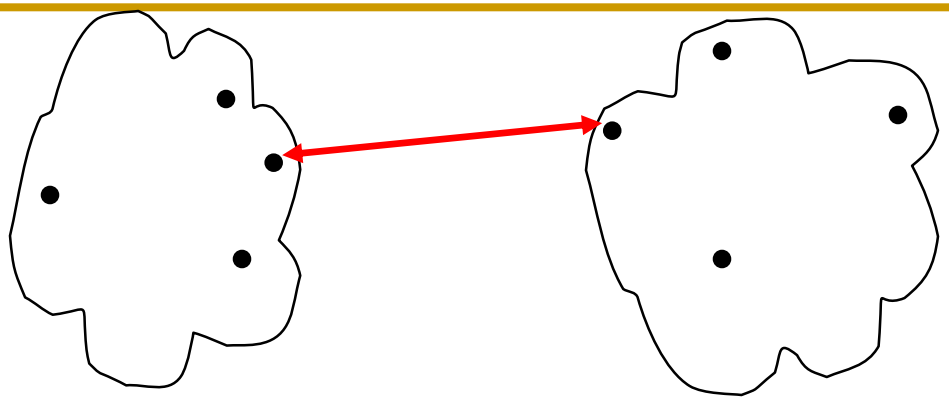
How to Merge Clusters?

- **How to measure the distance between clusters?**
  - **Single-link**
  - **Complete-link**
  - **Average-link**
  - **Centroid distance**



Hint: *Distance between clusters* is usually defined on the basis of *distance between objects.*

How to Define Inter-Cluster Distance

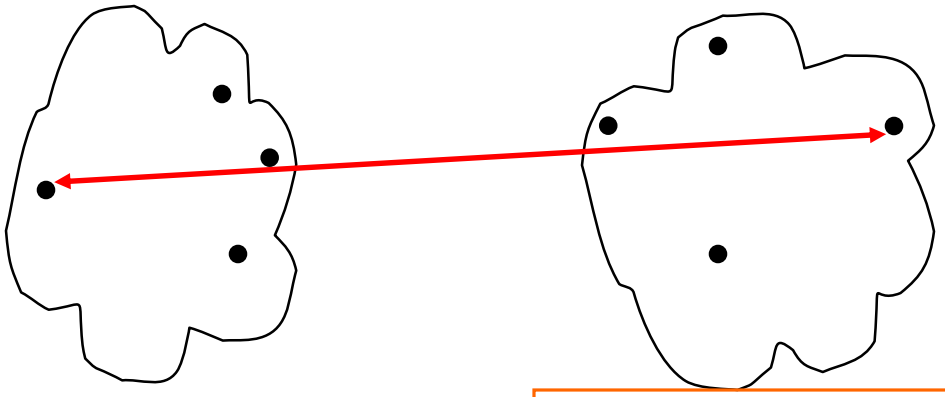


- **Single-link**
- **Complete-link**
- **Average-link**
- **Centroid distance**

$$d_{\min}(C_i, C_j) = \min_{p \in C_i, q \in C_j} d(p, q)$$

The distance between two clusters is represented by the distance of the *closest pair of data objects* belonging to different clusters.

How to Define Inter-Cluster Distance

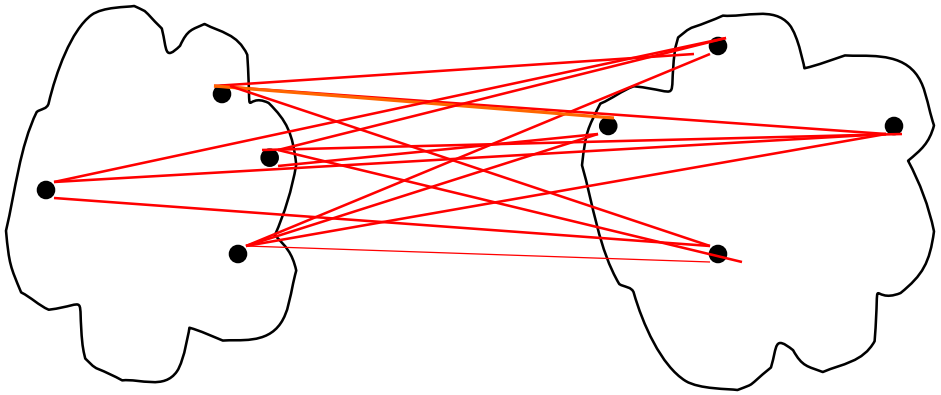


- **Single-link**
- **Complete-link**
- **Average-link**
- **Centroid distance**

$$d_{\min}(C_i, C_j) = \max_{p \in C_i, q \in C_j} d(p, q)$$

The distance between two clusters is represented by the distance of the *farthest pair of data objects* belonging to different clusters.

How to Define Inter-Cluster Distance

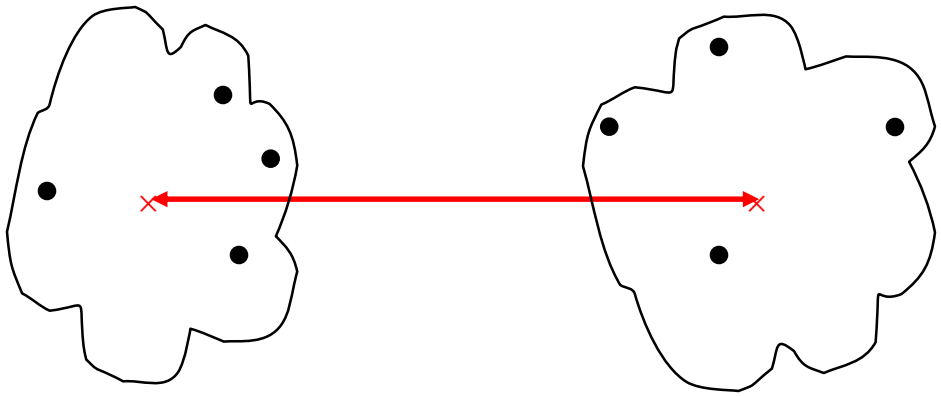


- **Single-link**
- **Complete-link**
- **Average-link**
- **Centroid distance**

$$d_{\min}(C_i, C_j) = \underset{p \in C_i, q \in C_j}{avg} d(p, q)$$

The distance between two clusters is represented by the *average* distance of *all pairs of data objects* belonging to different clusters.

How to Define Inter-Cluster Distance



m_i,m_j are the means of C_i, C_j,

- **Single-link**
- **Complete-link**
- **Average-link**
- **Centroid distance**

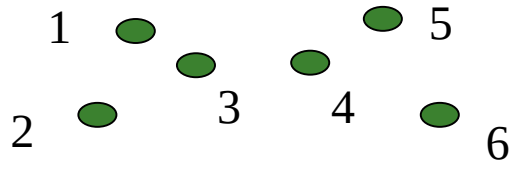
$$d_{mean}(C_i, C_j) = d(m_i, m_j)$$

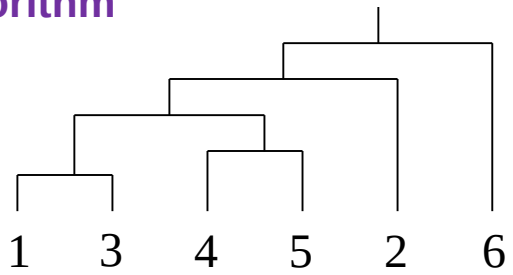
The distance between two clusters is represented by the distance between *the means of the cluters.*

E.g. Agglomerative Hierarchical Clustering

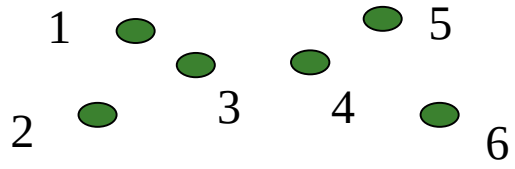
- **For the following data set, we will get different clustering results with the single-link and complete-link algorithms.**

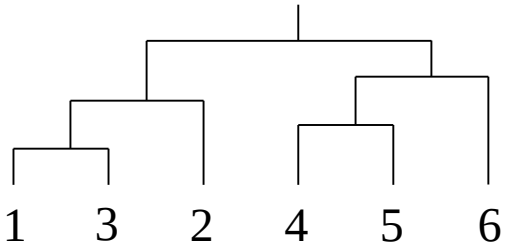
Result of the Single-Link algorithm





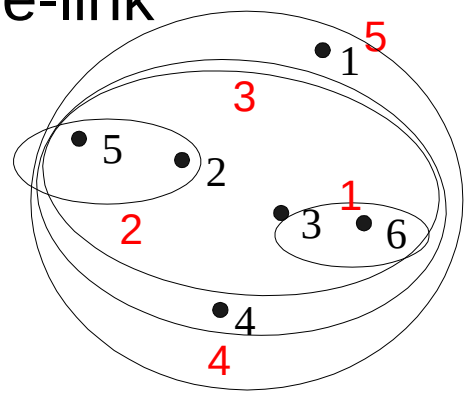
Result of the complete-Link algorithm



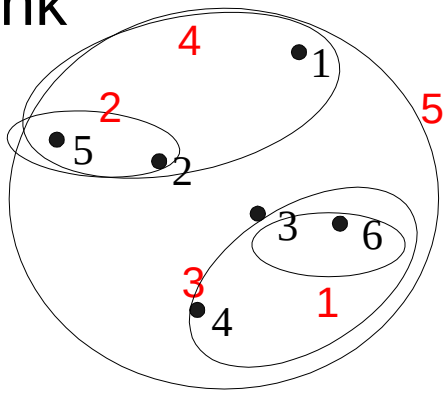


Hierarchical Clustering: Comparison

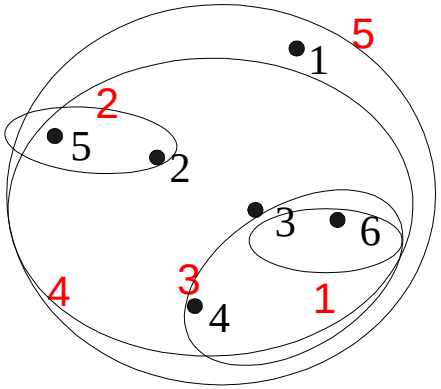
Single-link



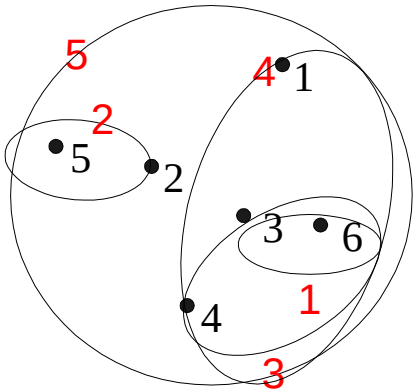
Complete-link



Average-link

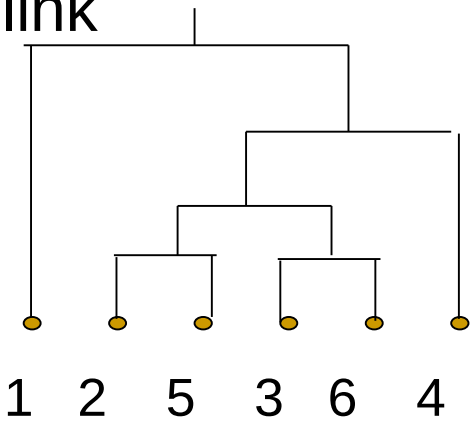


Centroid distance

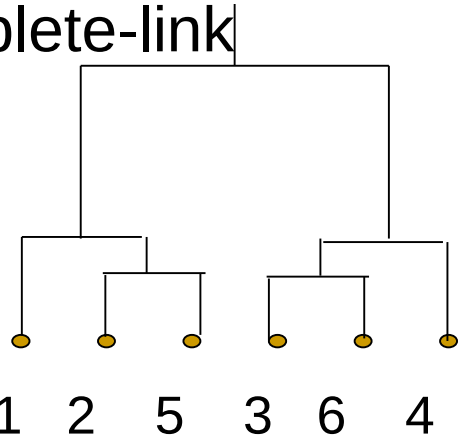


Comparison of Dendrograms

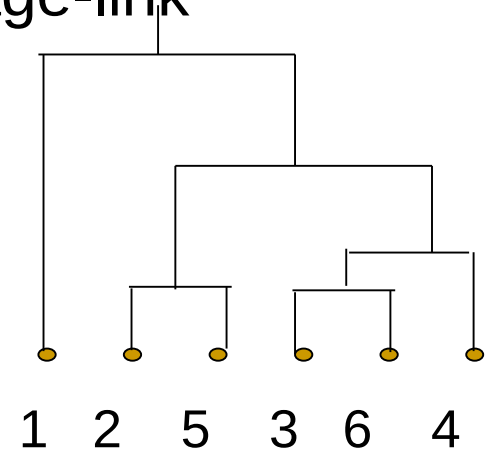
Single-link



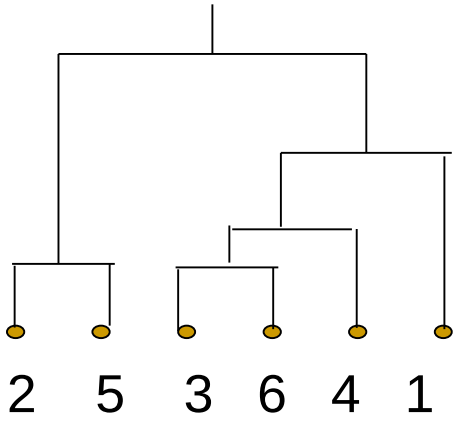
Complete-link



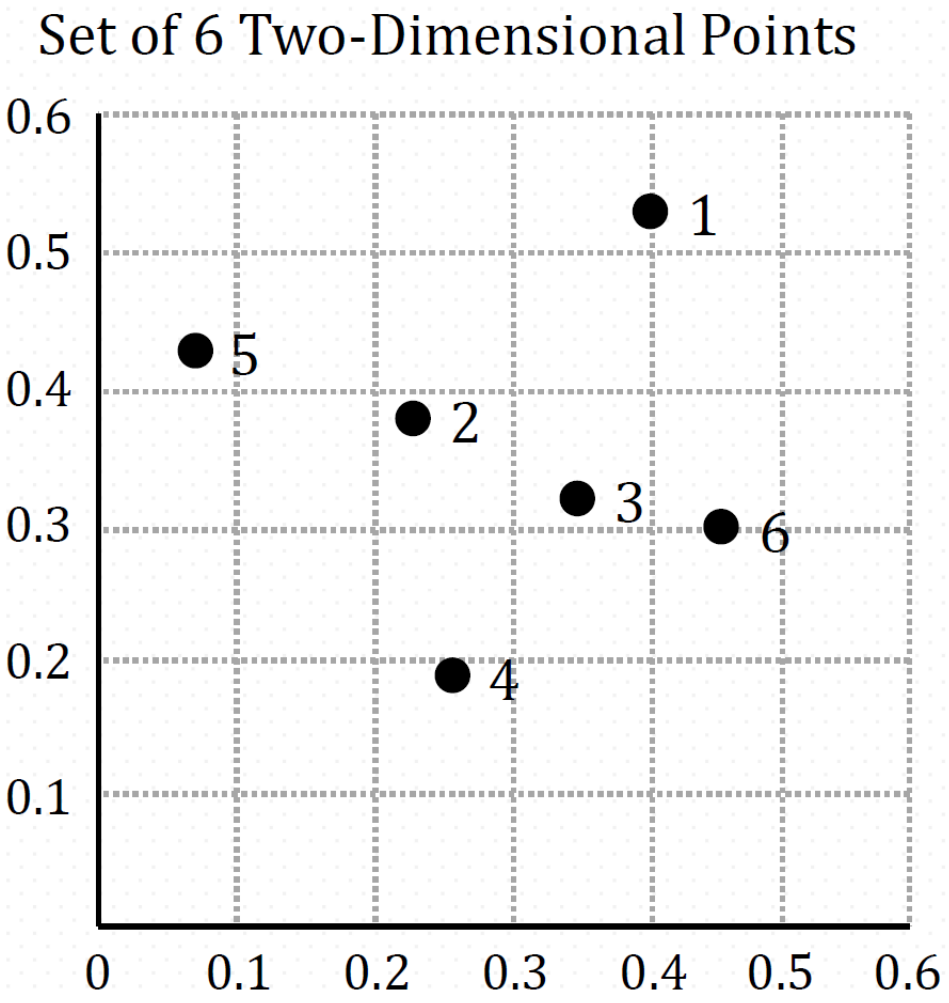
Average-link



Centroid distance



E.G. of Clustering



xy Coordinates of 6 Points

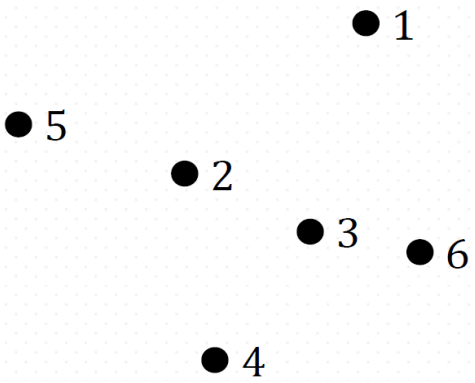
Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

Euclidean Distance Matrix for 6 Points

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

E.G. of Clustering using Single Link

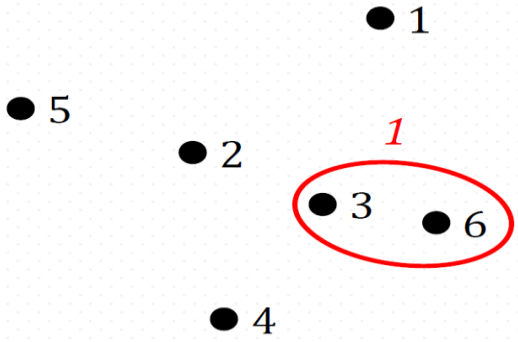
Nested Cluster Diagram



Single Link Distance Matrix

	1	2	3	4	5	6
1	0	0.24	0.22	0.37	0.34	0.23
2		0	0.15	0.20	0.14	0.25
3			0	0.15	0.28	0.11
4				0	0.29	0.22
5					0	0.39
6						0

Nested Cluster Diagram

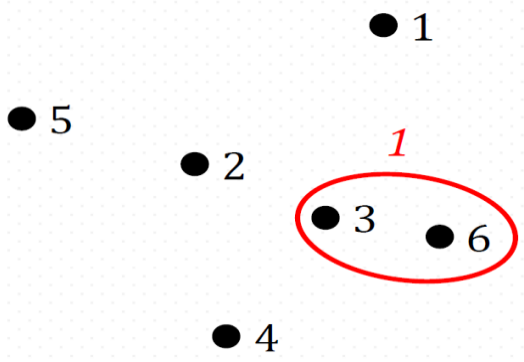


Single Link Distance Matrix

	1	2	3	4	5	6
1	0	0.24	*0.22*	0.37	0.34	0.23
2		0	*0.15*	0.20	0.14	0.25
3			0	*0.15*	*0.28*	0.11
4				0	0.29	0.22
5					0	0.39
6						0

E.G. of Clustering using Single Link

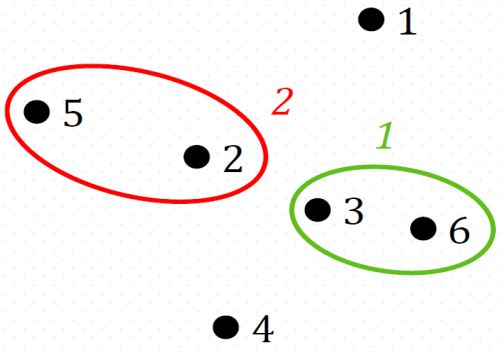
Nested Cluster Diagram



Single Link Distance Matrix

	1	2	3	4	5	6
1	0	0.24	*0.22*	0.37	0.34	0.23
2		0	*0.15*	0.20	0.14	0.25
3			0	*0.15*	*0.28*	0.11
4				0	0.29	0.22
5					0	0.39
6						0

Nested Cluster Diagram

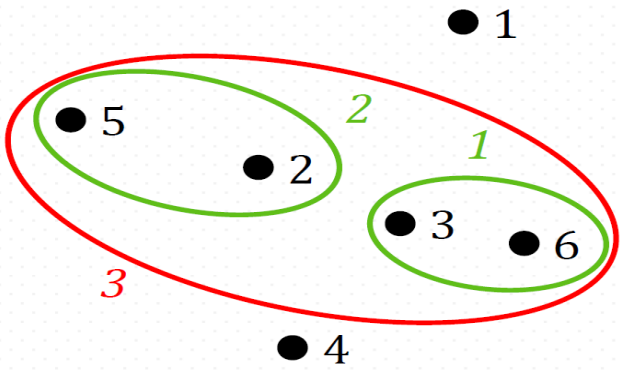


Single Link Distance Matrix

	1	2	4	5	3,6
1	0	*0.24*	0.37	0.34	0.22
2		0	*0.20*	0.14	0.15
4			0	0.29	0.15
5				0	0.28
3,6					0

E.G. of Clustering using Single Link

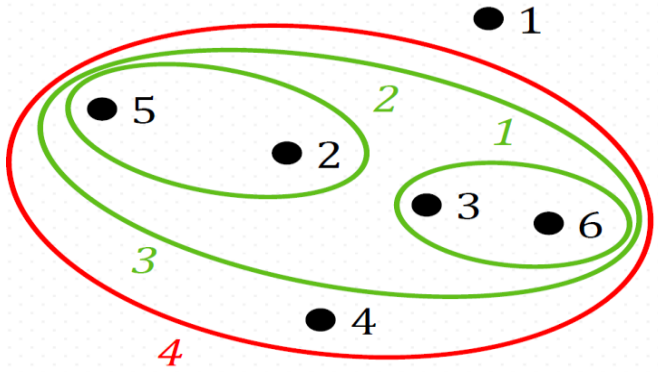
Nested Cluster Diagram



Single Link Distance Matrix

	1	4	2,5	3,6
1	0	0.37	0.24	0.22
4		0	0.20	0.15
2,5			0	0.15
3,6				0

Nested Cluster Diagram

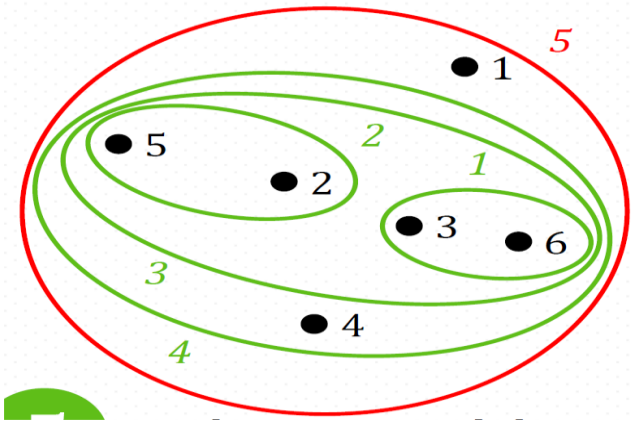


Single Link Distance Matrix

	1	4	2,5,3,6
1	0	0.37	0.22
4		0	0.15
2,5,3,6			0

E.G. of Clustering using Single Link

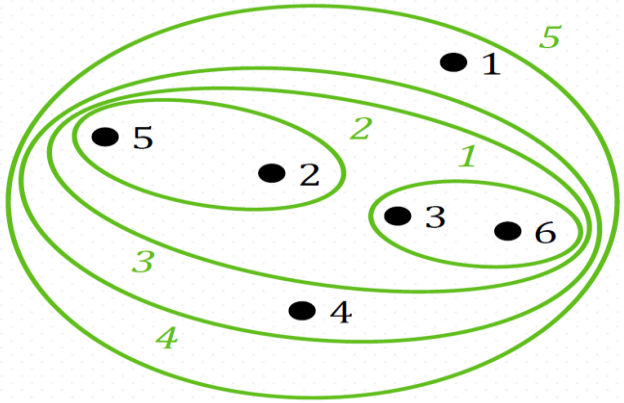
Nested Cluster Diagram



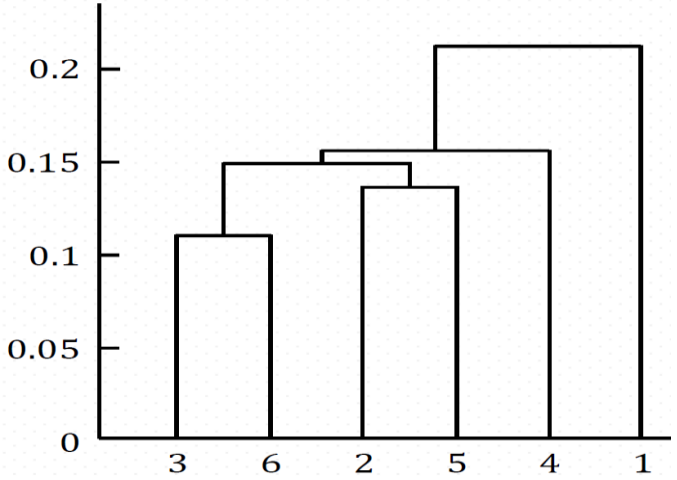
Single Link Distance Matrix

	1	4,2,5,3,6
1	0	0.22
2,5,3,6		0

Nested Cluster Diagram



Hierarchical Tree Diagram



Agglomerative clustering

It is more popular then divisive methods.

- **At the beginning, each data point forms a cluster (also called a node).**
- **Merge nodes/clusters that have the least distance.**
- **Go on merging**
- **Eventually all nodes belong to one cluster**

Agglomerative clustering algorithm

Algorithm Agglomerative(D)

1 Make each data point in the data set D a cluster,
2 Compute all pair-wise distances of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in D$;
2 **repeat**
3 find two clusters that are nearest to each other;
4 merge the two clusters form a new cluster c;
5 compute the distance from c to all other clusters;
12 **until** there is only one cluster left

The complexity

- **All the algorithms are at least $O(n^2)$. n is the number of data points.**
- **Single link can be done in $O(n^2)$.**
- **Complete and average links can be done in $O(n^2 \log n)$.**
- **Due the complexity, hard to use for large data sets.**
  - Sampling
  - Scale-up methods (e.g., BIRCH).

Outlines

- Basic concepts
- K-means algorithm
- Representation of clusters
- Hierarchical clustering
- Which clustering algorithm to use?
- Cluster evaluation
- Summary

How to choose a clustering algorithm

- **Clustering research has a long history. A vast collection of algorithms are available.**
  - We only introduced few key algorithms.
- **Choosing the "best" algorithm is a challenge.**
  - Every algorithm has limitations and works well with certain data distributions.
  - It is very hard, if not impossible, to know what distribution the application data follow. The data may not fully follow any "ideal" structure or distribution required by the algorithms.
  - One also needs to decide how to standardize the data, to choose a suitable distance function and to select other parameter values.

Choose a clustering algorithm (cont …)

- **Due to these complexities, the common practice is to**
  - run several algorithms using different distance functions and parameter settings, and
  - then carefully analyze and compare the results.
- **The interpretation of the results must be based on insight into the meaning of the original data together with knowledge of the algorithms used.**
- **Clustering is highly application dependent and to certain extent subjective (personal preferences).**

Outlines

- Basic concepts
- K-means algorithm
- Representation of clusters
- Hierarchical clustering
- Which clustering algorithm to use?
- Cluster evaluation
- Summary

Cluster Evaluation: hard problem

- **The quality of a clustering is very hard to evaluate because**
  - We do not know the correct clusters
- **Some methods are used:**
  - User inspection
    - Study centroids, and spreads
    - Rules from a decision tree.
    - For text documents, one can read some documents in clusters.

Cluster evaluation: ground truth

- **We use some labeled data (for classification)**
- **Assumption: Each class is a cluster.**
- **After clustering, a confusion matrix is constructed. From the matrix, we compute various measurements, entropy, purity, precision, recall and F-score.**
  - Let the classes in the data D be $C = (c_1, c_2, \ldots, c_k)$. The clustering method produces k clusters, which divides D into k disjoint subsets, $D_1, D_2, \ldots, D_k$.

Evaluation measures: Entropy

Entropy: For each cluster, we can measure its entropy as follows:

$$entropy(D_i) = -\sum_{j=1}^{k} \Pr_i(c_j) \log_2 \Pr_i(c_j), \quad (29)$$

where $\Pr_i(c_j)$ is the proportion of class c_j data points in cluster i or D_i. The total entropy of the whole clustering (which considers all clusters) is

$$entropy_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times entropy(D_i) \quad (30)$$

Evaluation measures: purity

Purity: This again measures the extent that a cluster contains only one class of data. The purity of each cluster is computed with

$$purity(D_i) = \max_j(\Pr_i(c_j)) \tag{31}$$

The total purity of the whole clustering (considering all clusters) is

$$purity_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times purity(D_i) \tag{32}$$

An example

- Assume we have a text collection Dof 900 documents from three topics (or three classes), Science, Sports, and Politics. Each class has 300 documents, and each document in D is labeled with one of the topics (classes). We use this collection to perform clustering to find three clusters. Class/topic labels are not used in clustering. After clustering, we want to measure the effectiveness of the clustering algorithm.

Cluster	Science	Sports	Politics	Entropy	Purity
1	250	20	10	0.589	0.893
2	20	180	80	1.198	0.643
3	30	100	210	1.257	0.617
Total	300	300	300	1.031	0.711

A remark about ground truth evaluation

- **Commonly used to compare different clustering algorithms.**
- **A real-life data set for clustering has no class labels.**
  - Thus although an algorithm may perform very well on some labeled data sets, no guarantee that it will perform well on the actual application data at hand.
- **The fact that it performs well on some label data sets does give us some confidence of the quality of the algorithm.**
- **This evaluation method is said to be based on external data or information.**

Evaluation based on internal information

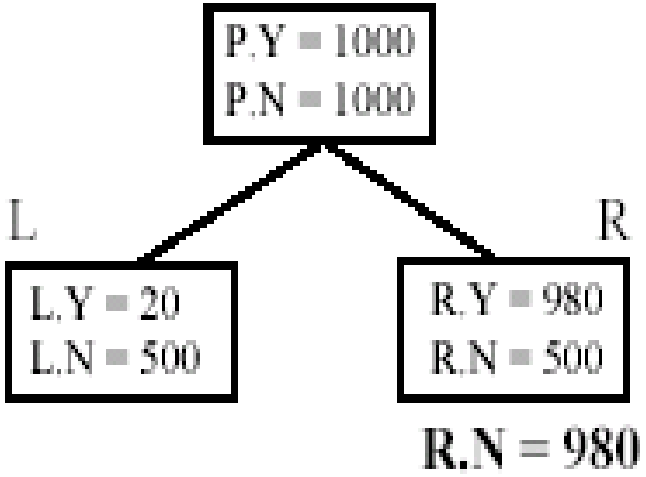
- **Intra-cluster cohesion (compactness):**
  - Cohesion measures how near the data points in a cluster are to the cluster centroid.
  - Sum of squared error (SSE) is a commonly used measure.
- **Inter-cluster separation (isolation):**
  - Separation means that different cluster centroids should be far away from one another.
- **In most applications, expert judgments are still the key.**

Indirect evaluation

- **In some applications, clustering is not the primary task, but used to help perform another task.**
- **We can use the performance on the primary task to compare clustering methods.**
- **For instance, in an application, the primary task is to provide recommendations on book purchasing to online shoppers.**
  - If we can cluster books according to their features, we might be able to provide better recommendations.
  - We can evaluate different clustering algorithms based on how well they help with the recommendation task.
  - Here, we assume that the recommendation can be reliably evaluated.

An example

Example 17: Fig. 20 gives an example. The (parent) node P has two children nodes L and R. Assume P has 1000 Y points and thus 1000 N points, stored in $P.Y$ and $P.N$ respectively. Assume after splitting, L has 20 Y points and 500 N points, and R has 980 Y points and 500 N points. According to the above rule, for subsequent partitioning, we increase the number of N points at R to 980. The number of N points at L is unchanged.



Summary

- **Clustering is has along history and still active**
  - There are a huge number of clustering algorithms
  - More are still coming every year.
- **We only introduced several main algorithms. There are many others, e.g.,**
  - density based algorithm, sub-space clustering, scale-up methods, neural networks based methods, fuzzy clustering, co-clustering, etc.
- **Clustering is hard to evaluate, but very useful in practice. This partially explains why there are still a large number of clustering algorithms being devised every year.**
- **Clustering is highly application dependent and to some extent subjective.**