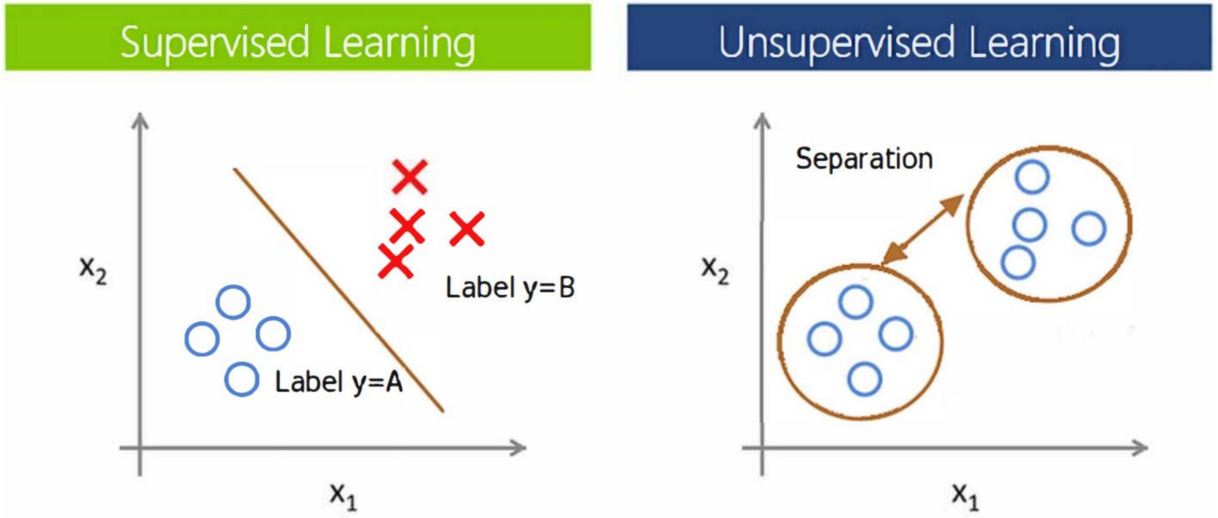
CS3121 - Introduction to Data Science

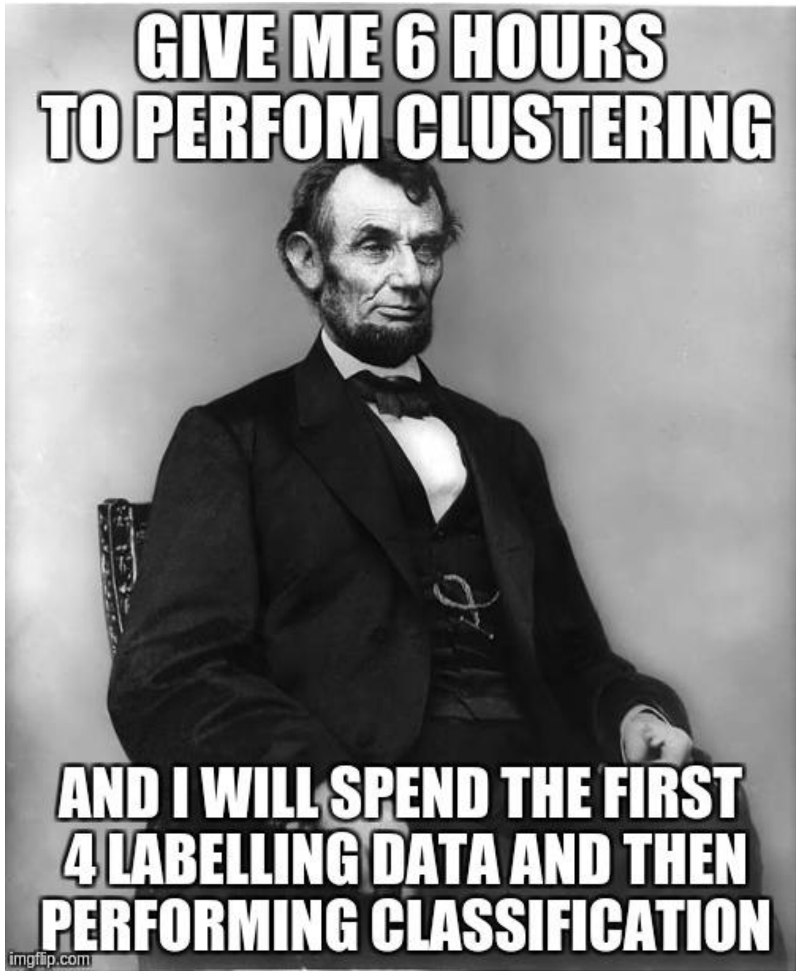
Unsupervised Learning

Dr. Nisansa de Silva,
Department of Computer Science & Engineering
http://nisansads.staff.uom.lk/

Supervised Learning vs. Unsupervised Learning

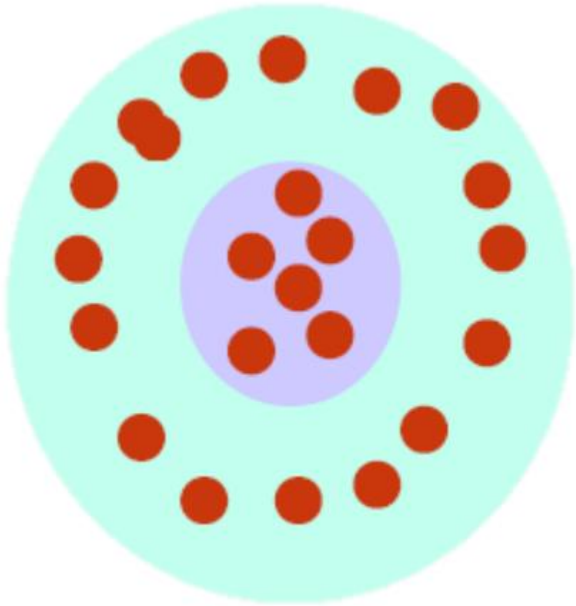
- Supervised learning: discover patterns in the data that relate data attributes with a target (class) attribute.
  - These patterns are then utilized to predict the values of the target attribute in future data instances.
- Unsupervised learning: The data have no target attribute.
  - We want to explore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

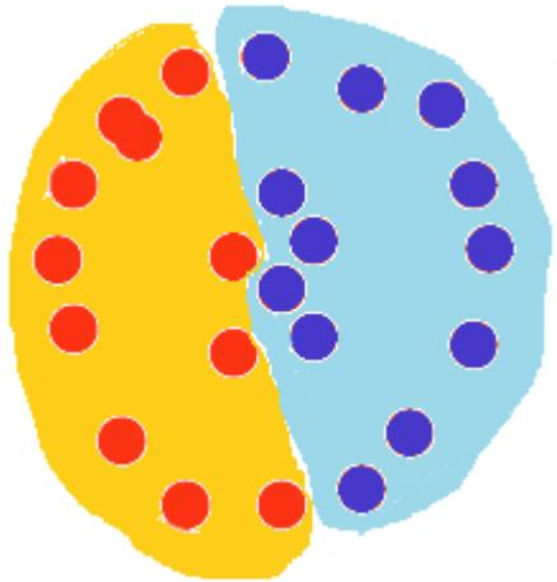




Clustering

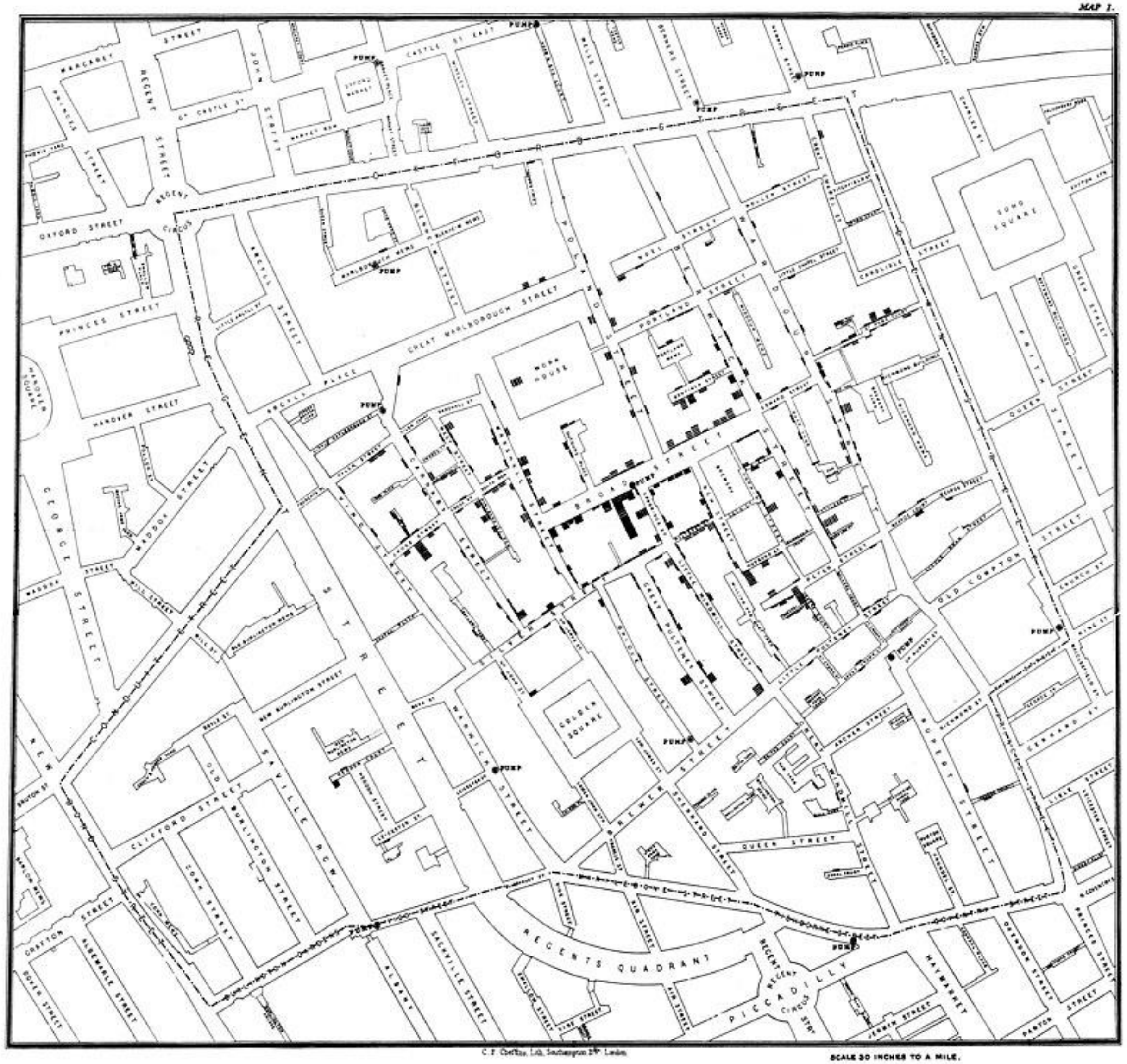
- The organization of unlabeled data into similarity groups called clusters.
  - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.
- Due to historical reasons, clustering is often considered synonymous with unsupervised learning.
  - In fact, association rule mining is also unsupervised
- This lecture focuses on clustering.



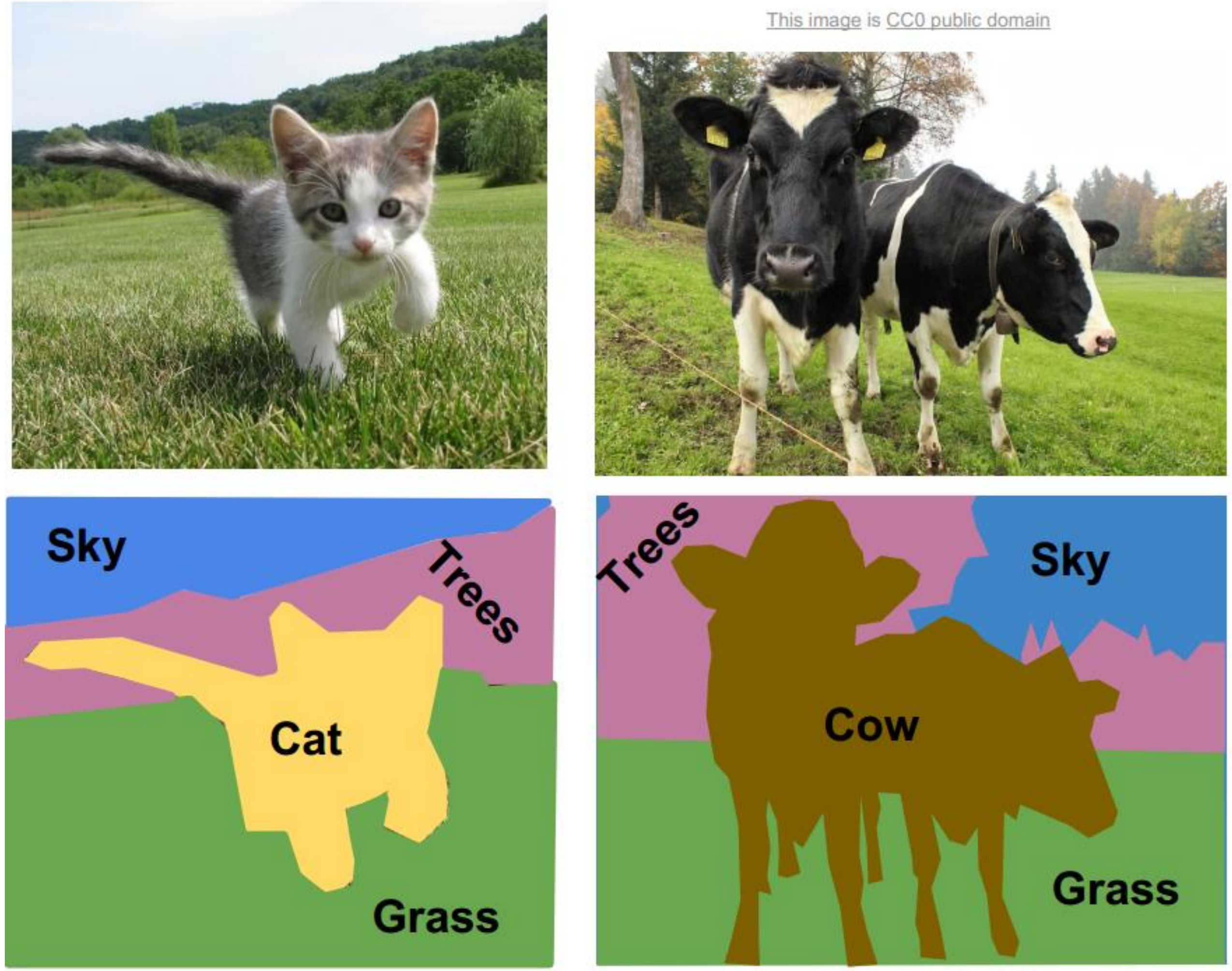


Historic Application of Clustering

- John Snow, London physician plotted the location of cholera deaths on a map during an outbreak in the 1850s.
- The locations indicated that cases were clustered around certain intersections where there were polluted wells, thus exposing both the problem and the solution.



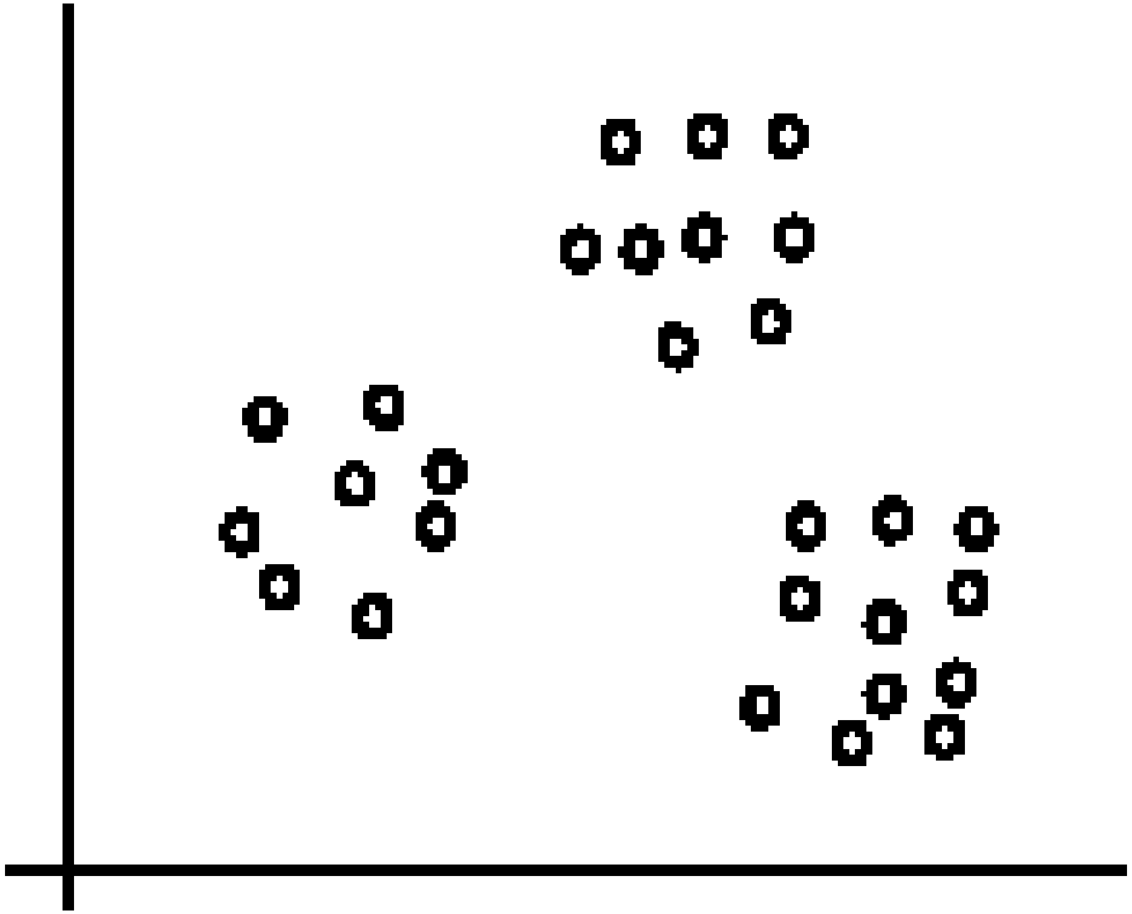
Computer vision application: Image segmentation



https://tariq-hasan.github.io/concepts/computer-vision-semantic-segmentation/

An illustration

- The data set has three natural groups of data points, i.e., 3 natural clusters.



What is clustering for?

- Let us see some real-life examples
- 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.
- 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 do we need for clustering?

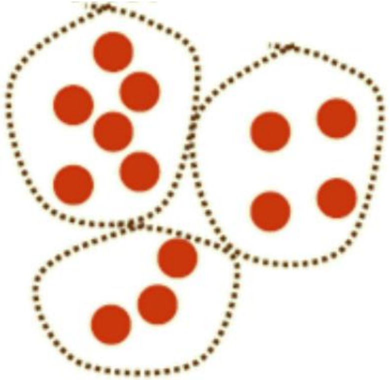
1. Proximity measure. Either,
   - Similarity measure $s(x_i, x_k)$: large if x_i, x_k are similar
   - Dissimilarity (or distance) measure $d(x_i, x_k)$: small if x_i, x_k are similar

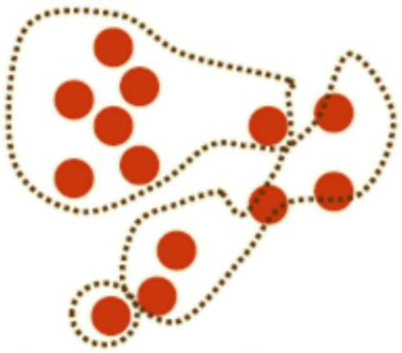
large d, small s

large s, small d

2. Criterion function to evaluate a clustering (Clustering quality)

Good clustering



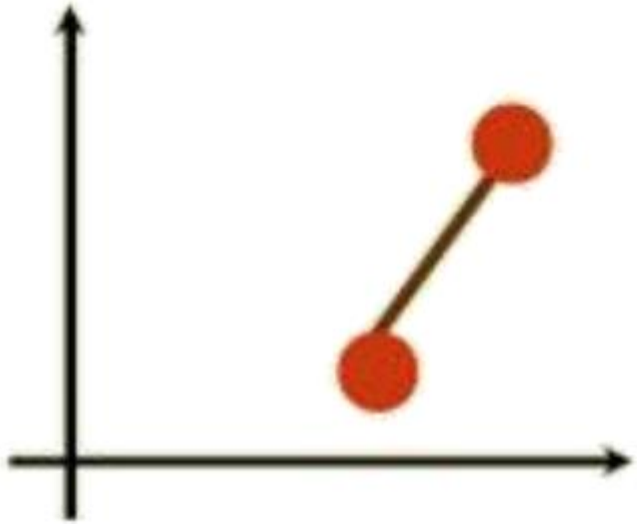


Bad clustering

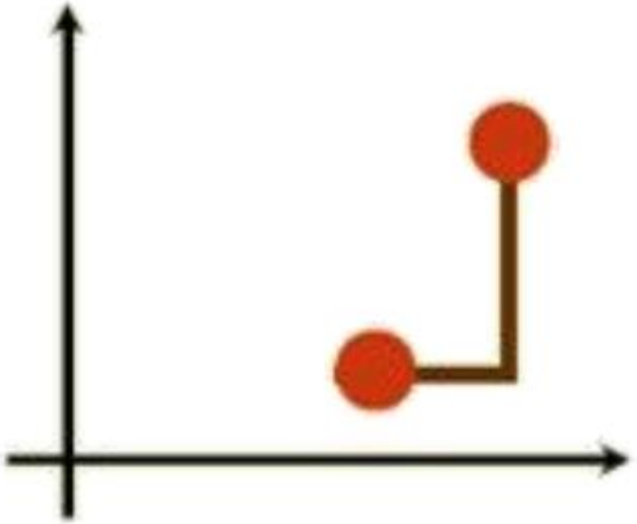
3. Algorithm to compute clustering (Clustering techniques)

Distance (Dissimilarity) Measures

- Euclidian distance
  - $d(x_i, x_j) = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2}$
  - Translation invariant



- Manhattan (city block) distance
  - $d(x_i, x_j) = \sum_{k=1}^{d} |x_{i,k} - x_{j,k}|$
  - Approximation of Euclidian distance



- They are special cases of Minkowski distance:
  - $d_p(x_i, x_j) = \left(\sum_{k=1}^{d} |x_{i,k} - x_{j,k}|^p\right)^{\frac{1}{p}}$
  - Where p is a positive integer

Clustering Quality: Cluster Evaluation (a Hard Problem)

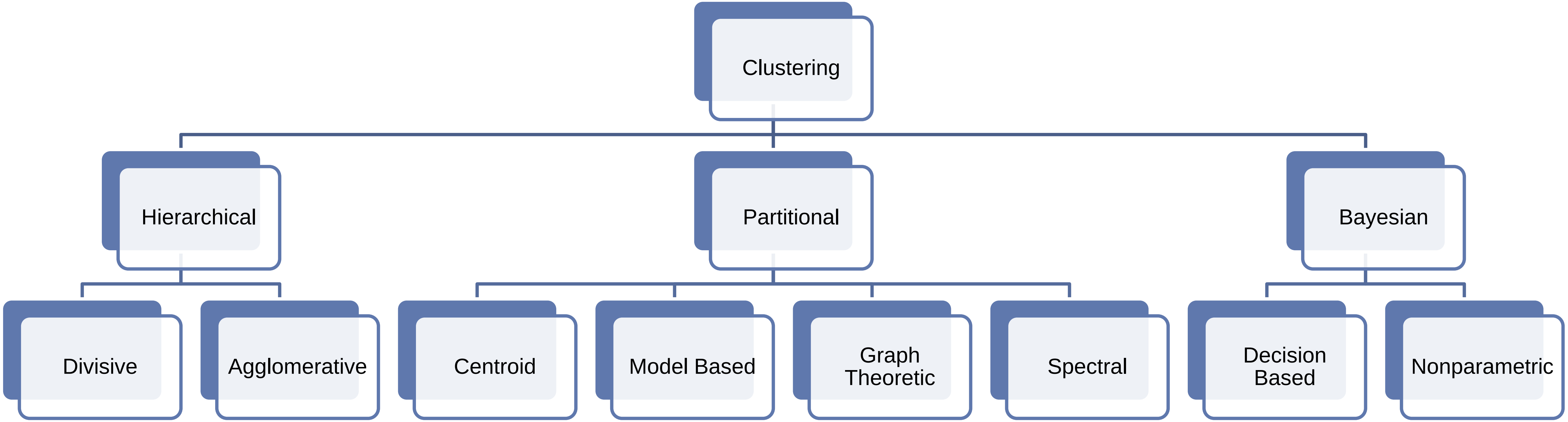
- **Intra-cluster cohesion** (compactness):
  - Maximized
  - Cohesion measures how near the data points in a cluster are to the cluster centroid.
  - Sum of squared error (SSE) is a commonly used measure.
- **Inter-cluster separation** (isolation):
  - Minimized
  - Separation means that different cluster centroids should be far away from one another.
- In most applications, expert judgments are still the key
- However, the overall quality of a clustering result depends on the algorithm, the distance function, and the application.

How Many Clusters?

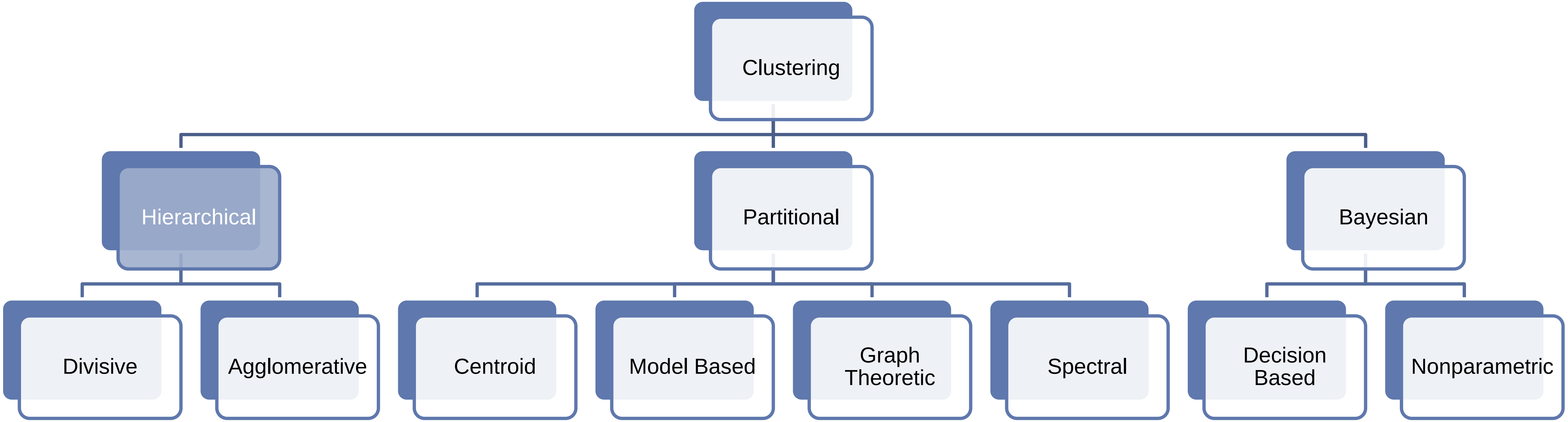
3 clusters or 2 clusters?

- Possible approaches
  1. Fix the number of clusters to k
  2. Find the best clustering according to the criterion function (number of clusters may vary)

Clustering Techniques

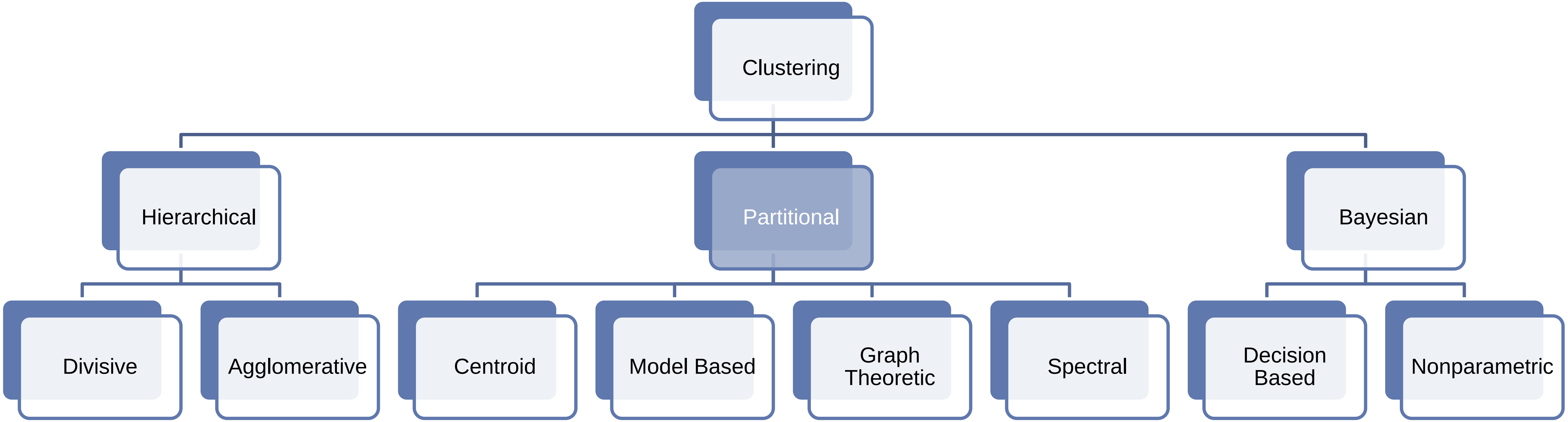


Clustering Techniques



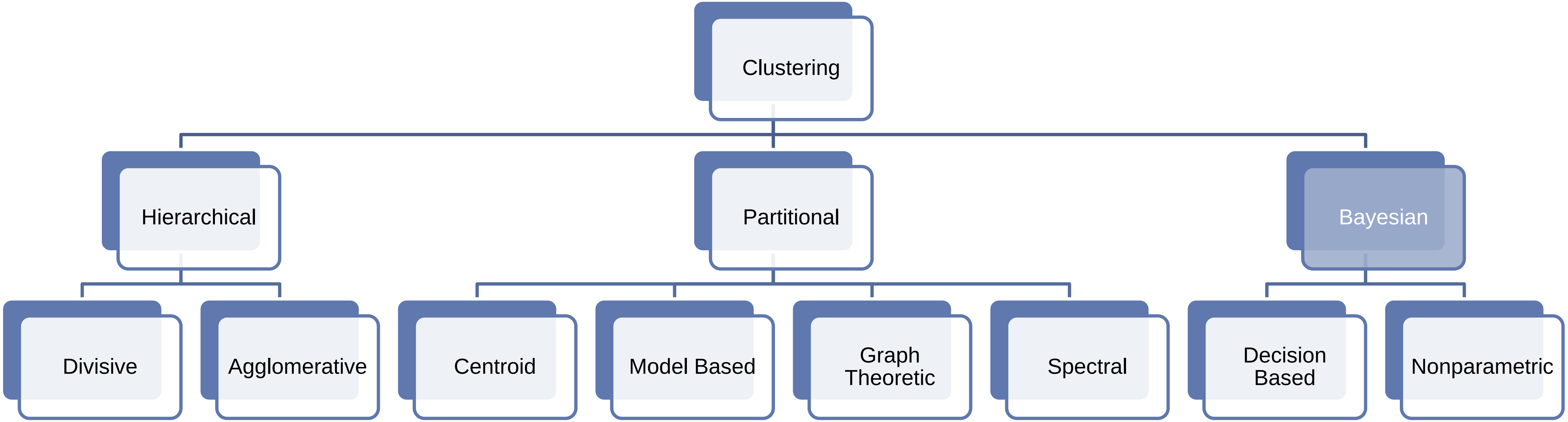
- Hierarchical algorithms find successive clusters using previously established clusters. These algorithms can be either aggolomerative ("bottom-up") or divisive ("top-down")
  - Aggolomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.
  - Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

Clustering Techniques



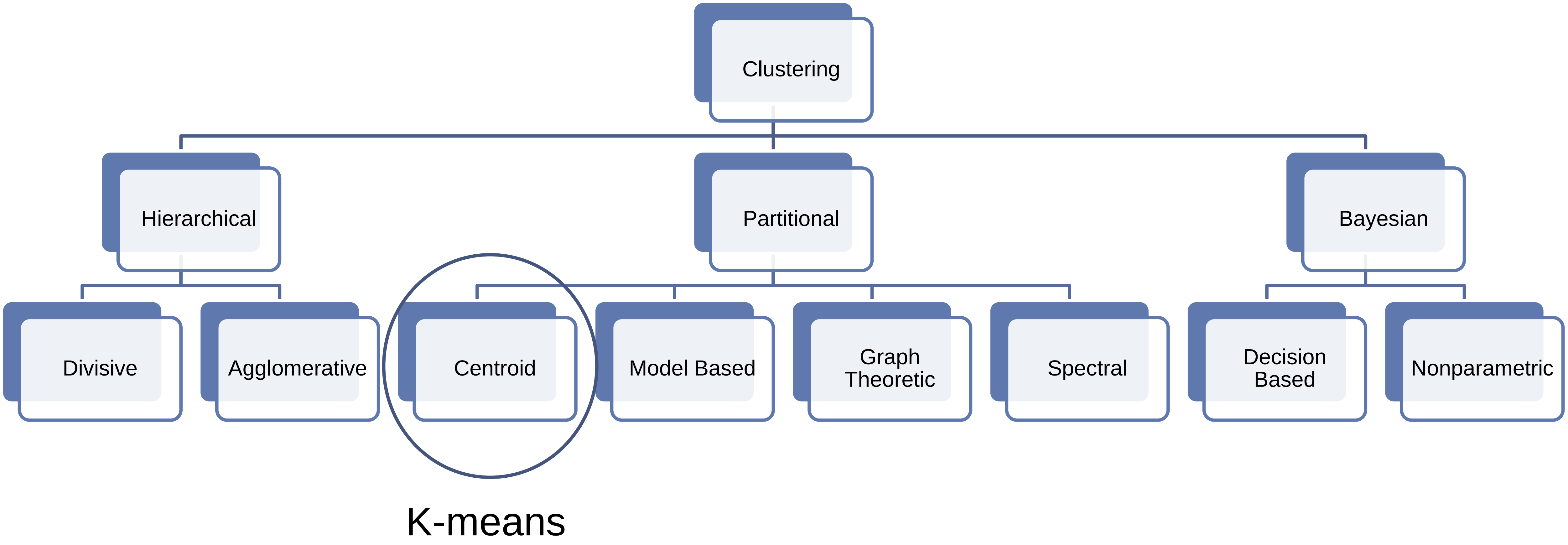
- Partitional algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering

Clustering Techniques



- Bayesian algorithms try to generate a *posteriori distribution* over the collection of all partitions of the data.

Clustering Techniques



K-means algorithm

K-means Clustering

- K-means (MacQueen, 1967) is a partitional clustering algorithm

- Let the set of data points (or instances) D be $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$,

  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).

```
Algorithm k-means(k, D)
  Choose k data points as the initial centroids (cluster centers)
  repeat
      for each data point x ∈ D do
          compute the distance from x to each centroid;
          assign x to the closest centroid        // a centroid represents a cluster
      endfor
      re-compute the centroids using the current cluster memberships
  until the stopping criterion is met
```

Stopping/Convergence Criterion

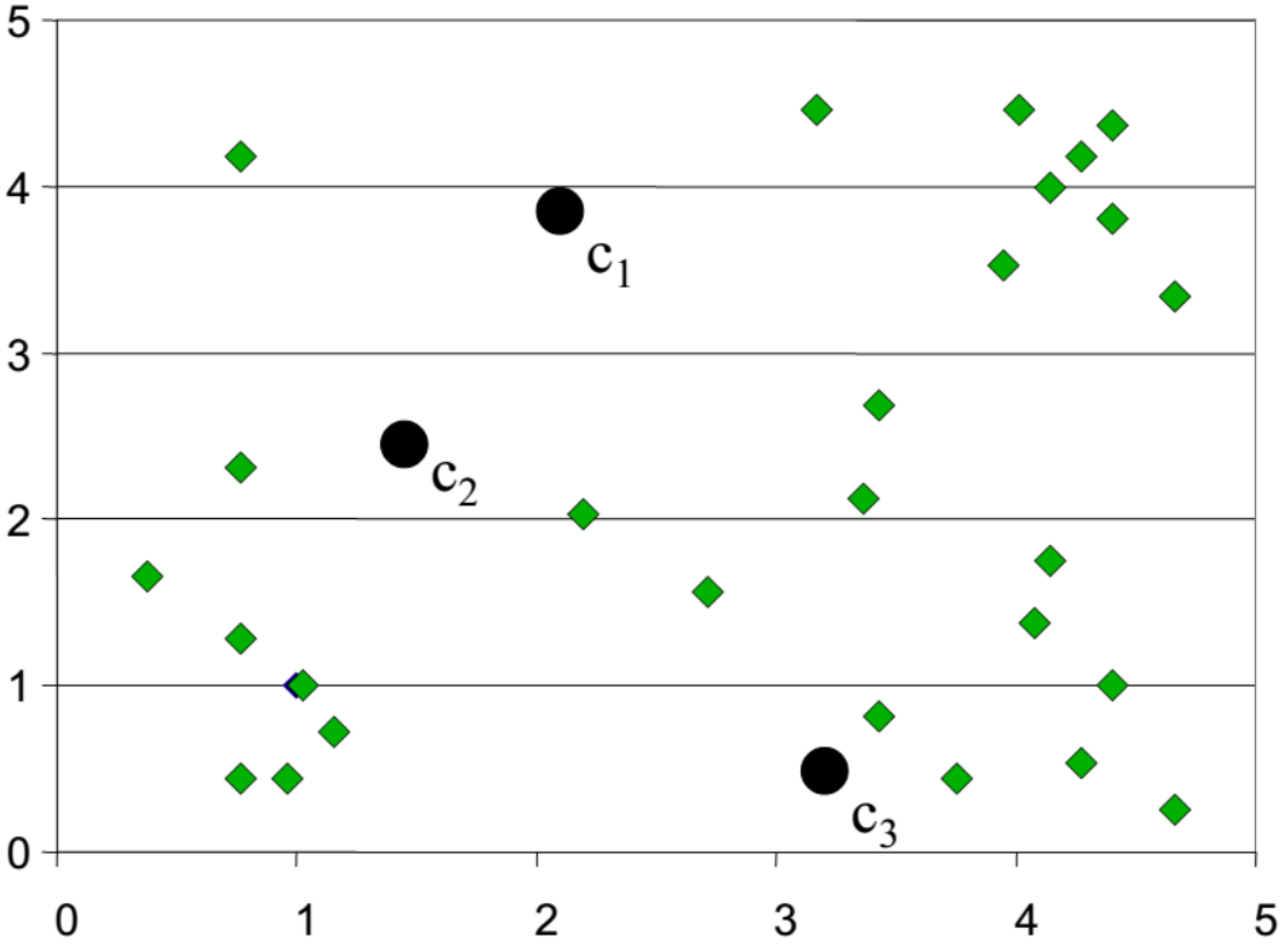
1. no (or minimum) re-assignments of data points to different clusters,
2. no (or minimum) change of centroids, or
3. minimum decrease in the **sum of squared error** (SSE),

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

- C_i is the jth cluster
- $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j)
- $dist(\mathbf{x}, \mathbf{m}_j)$ is the distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_j$.

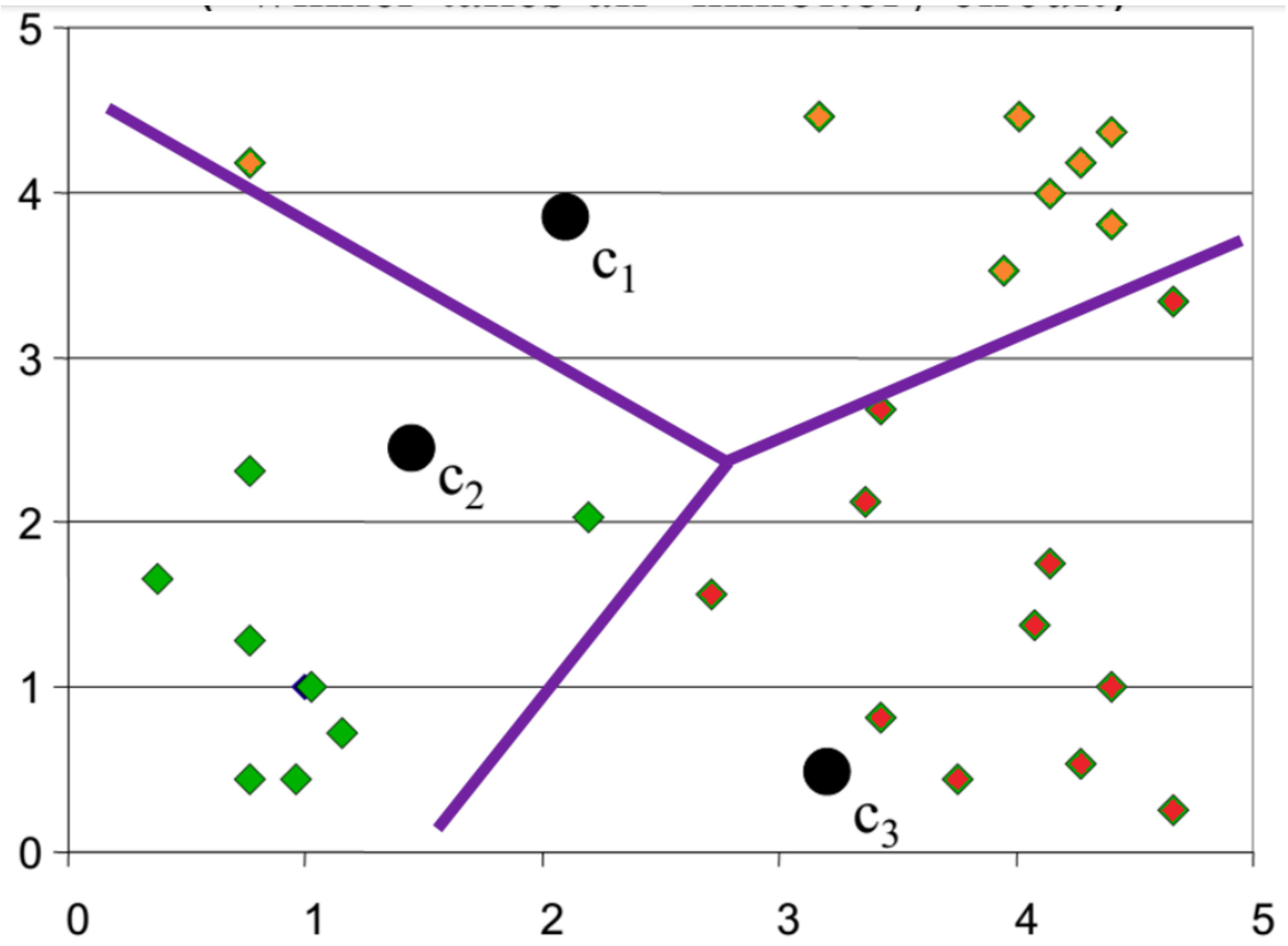
K-means Clustering Example: Iteration 1: Step 1

- Randomly initialize the cluster centers (synaptic weights)



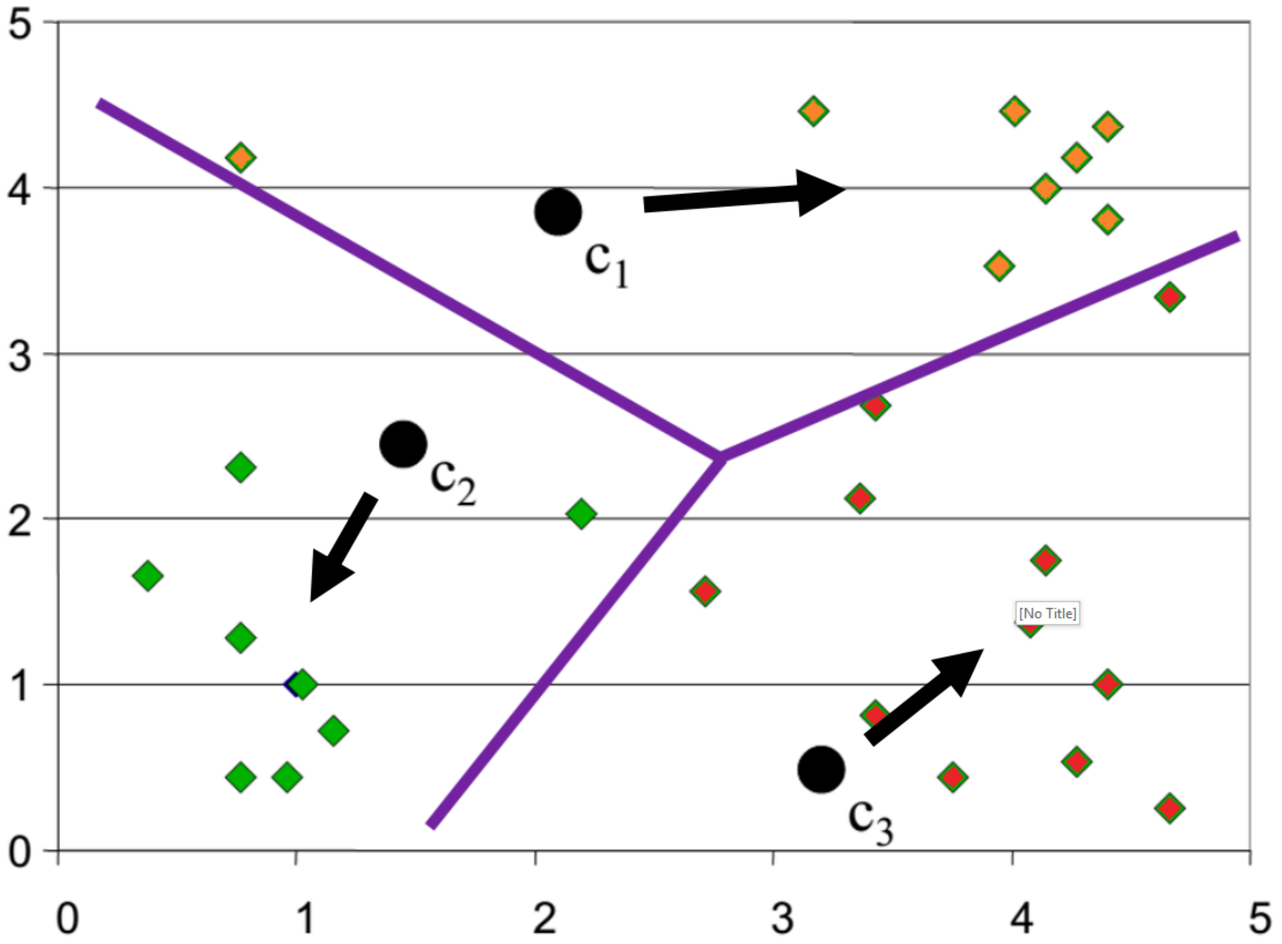
K-means Clustering Example: Iteration 1: Step 2

- Determine cluster membership for each input ("winner-takes-all" inhibitory circuit)

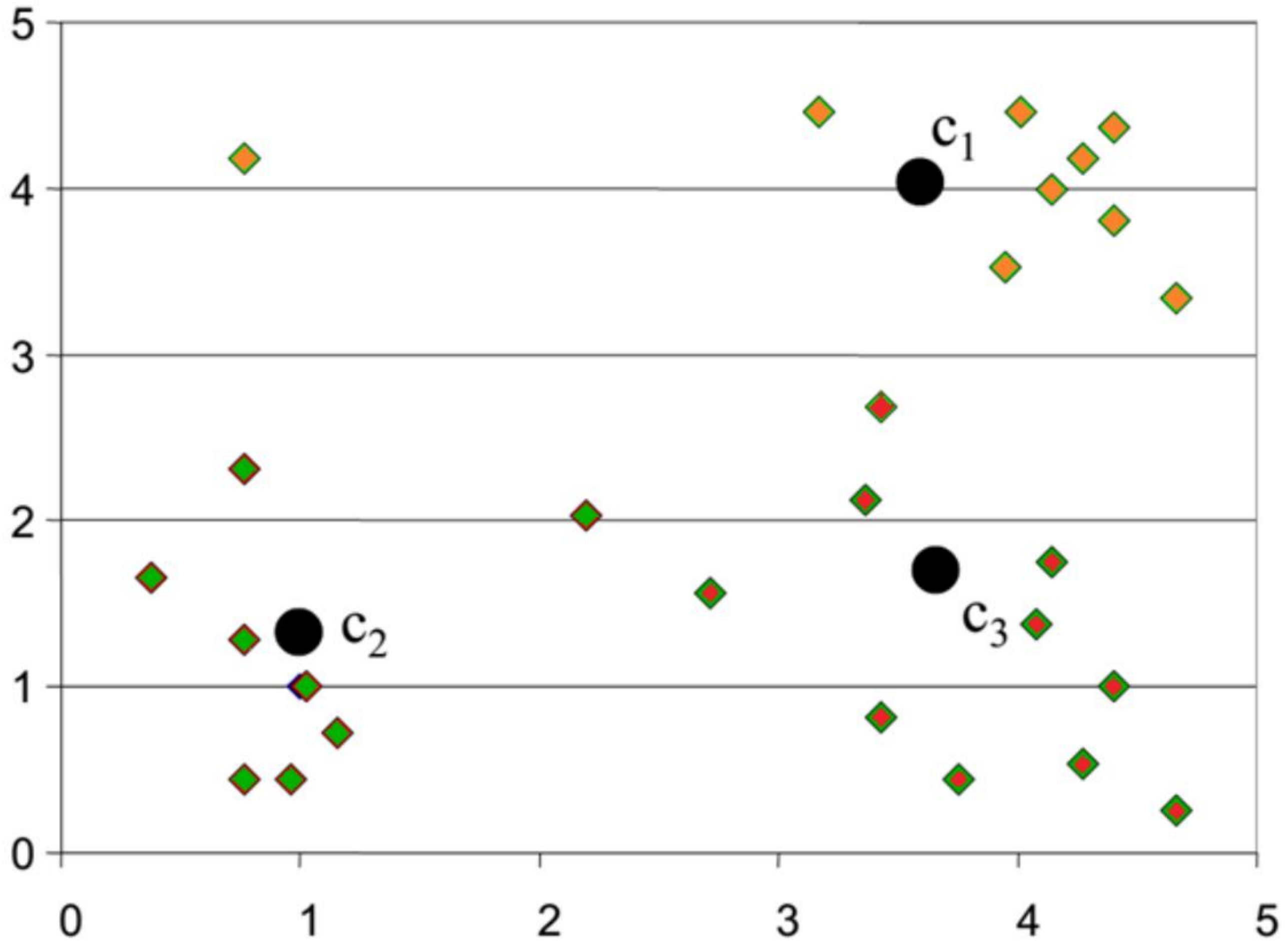


K-means Clustering Example: Iteration 1: Step 3

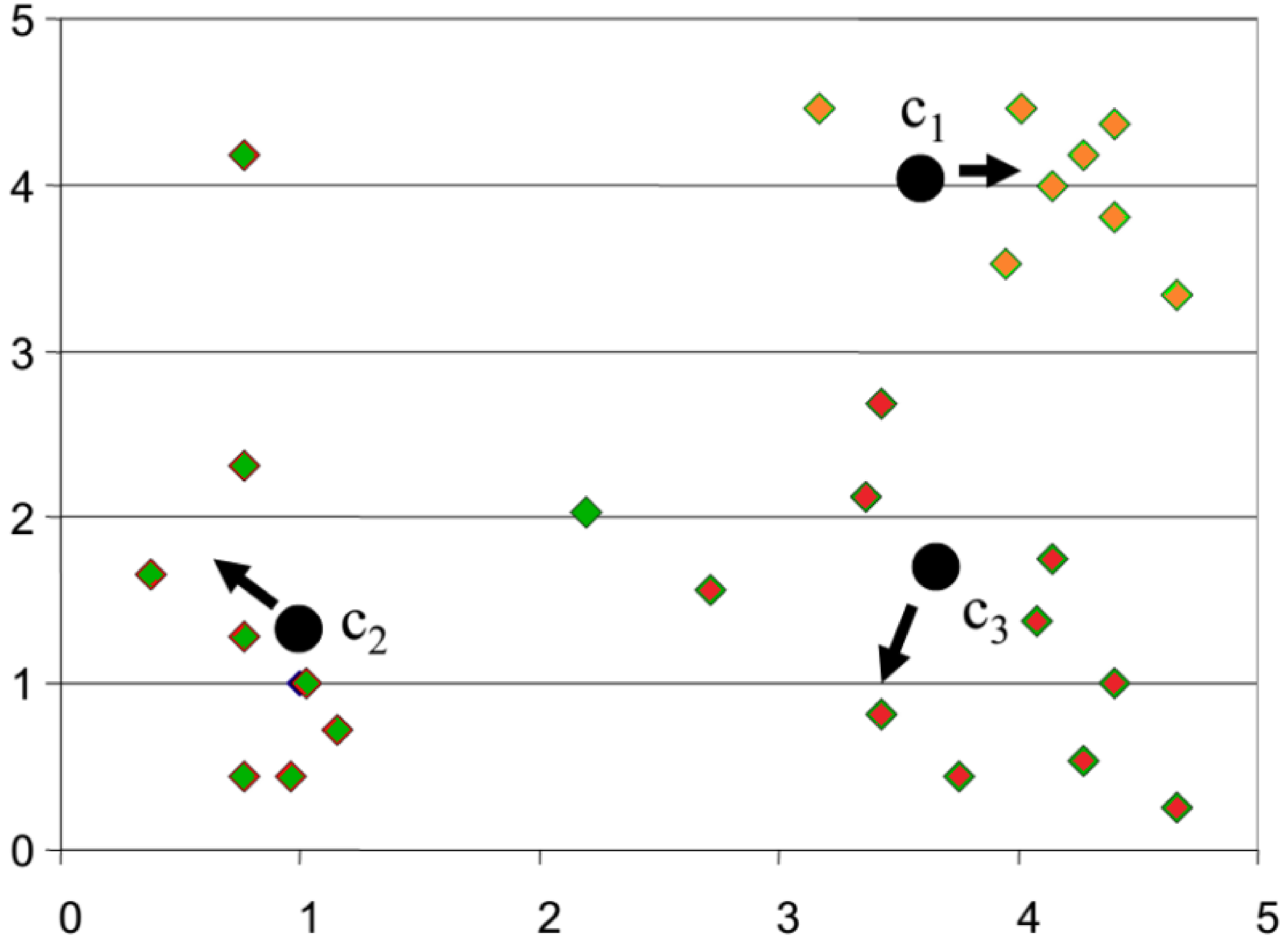
- Re-estimate cluster centers (adapt synaptic weights)



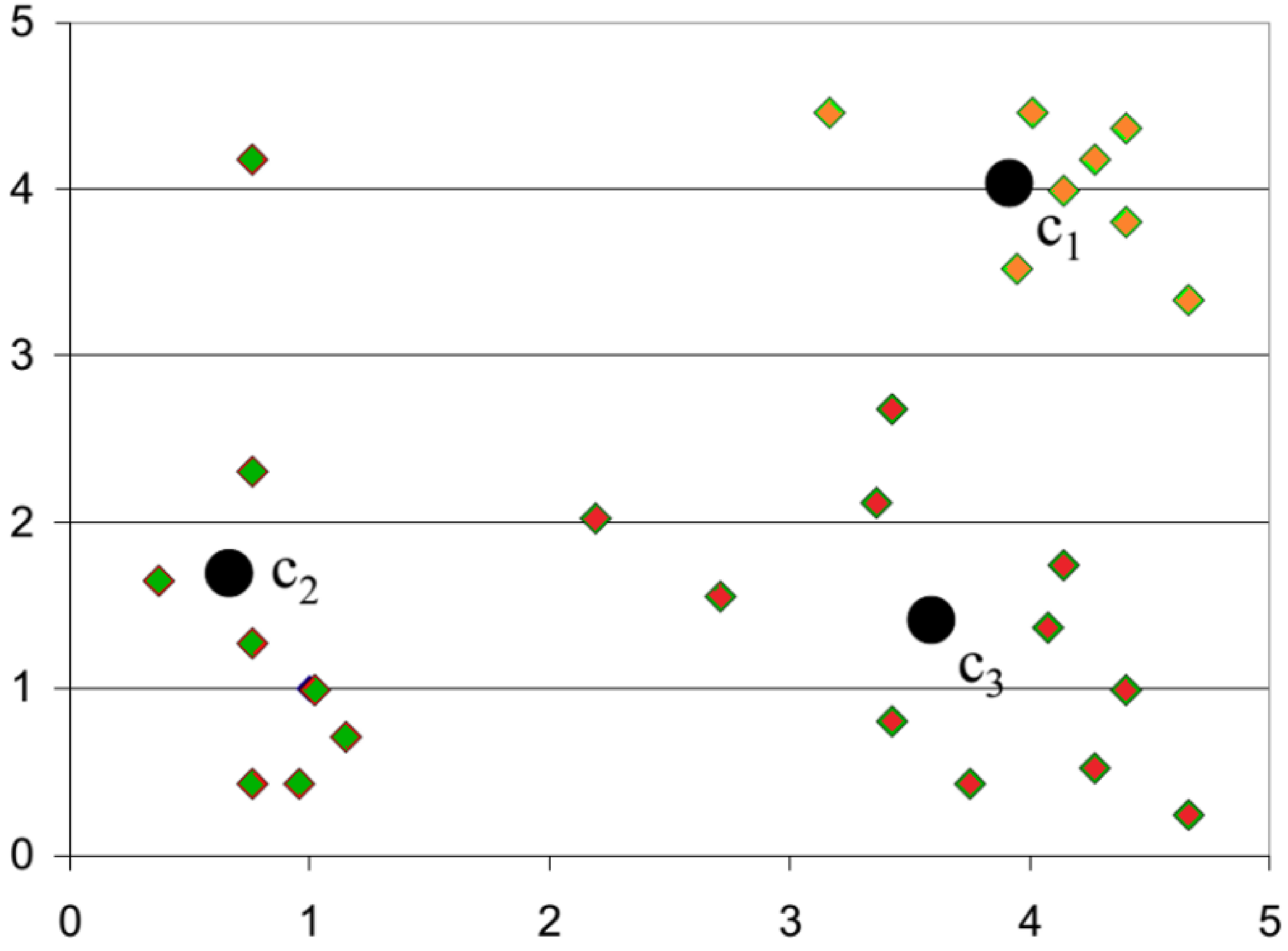
K-means Clustering Example: Iteration 1: Result



K-means Clustering Example: Iteration 2



K-means Clustering Example: Iteration 2: Result



Strengths and Weaknesses of k-means

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tkn)$,

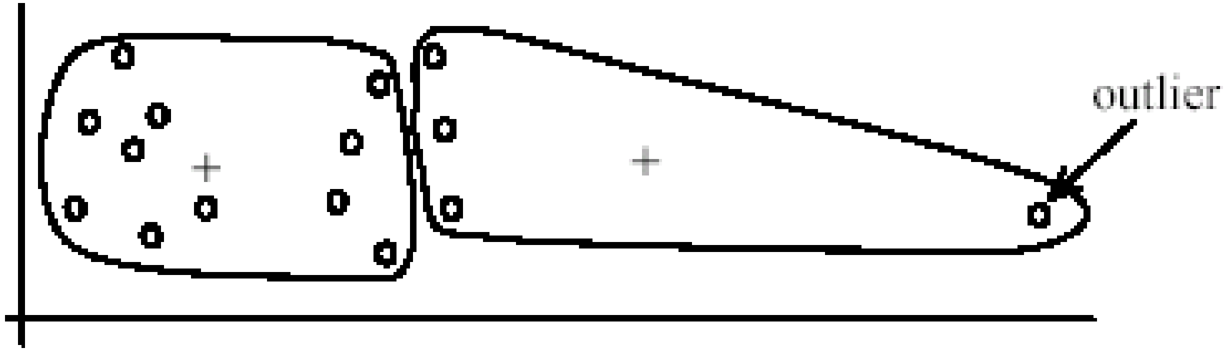
    where n is the number of data points,

    k is the number of clusters, and

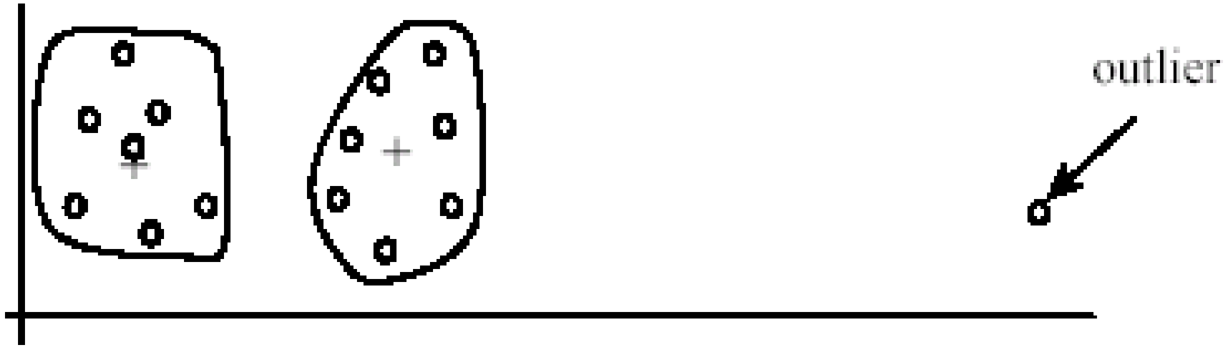
    t is the number of iterations.
  - Since both k and t are small. *k*-means is considered a linear algorithm.

- Weaknesses:
  - 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.

K-means Issues: Difficult to Handle Outliers



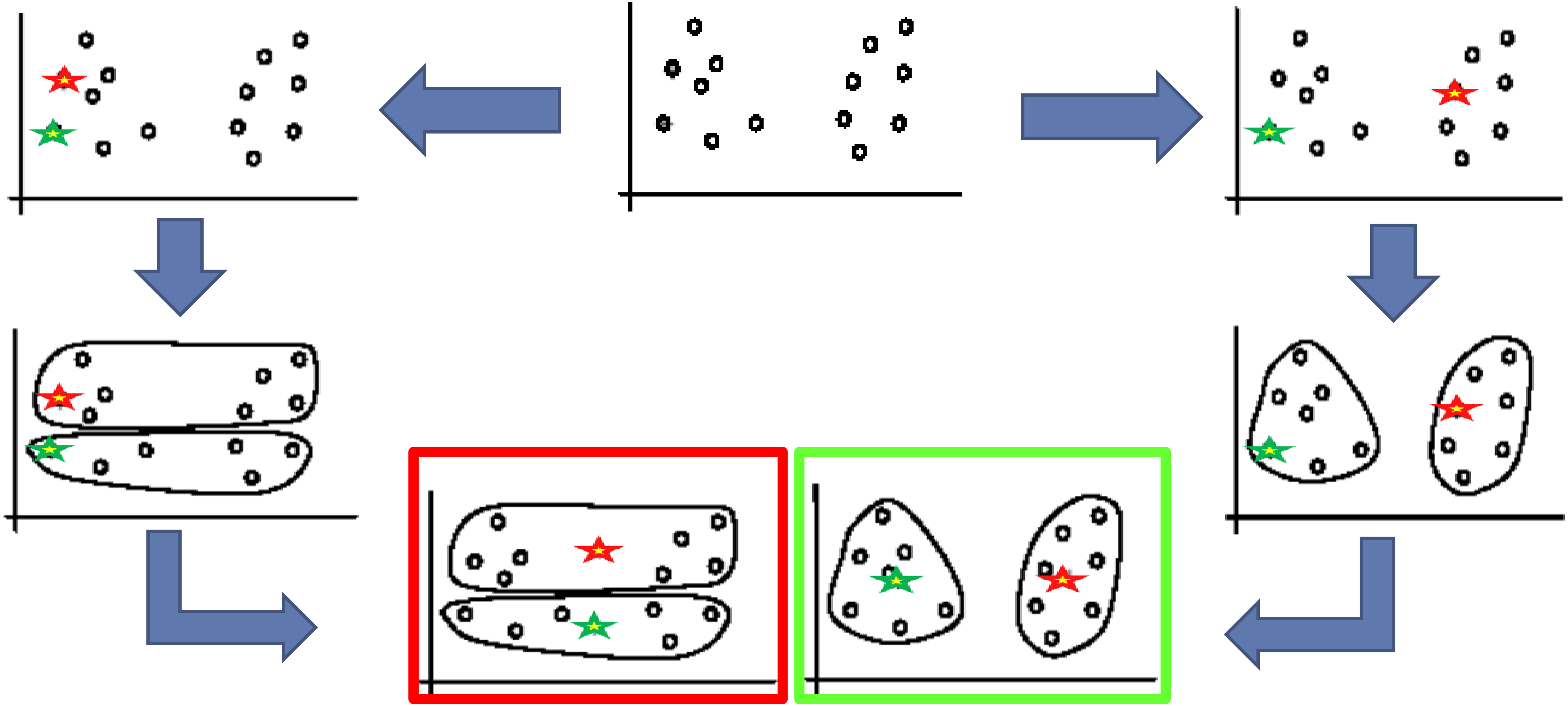
(A): Undesirable clusters



(B): Ideal clusters

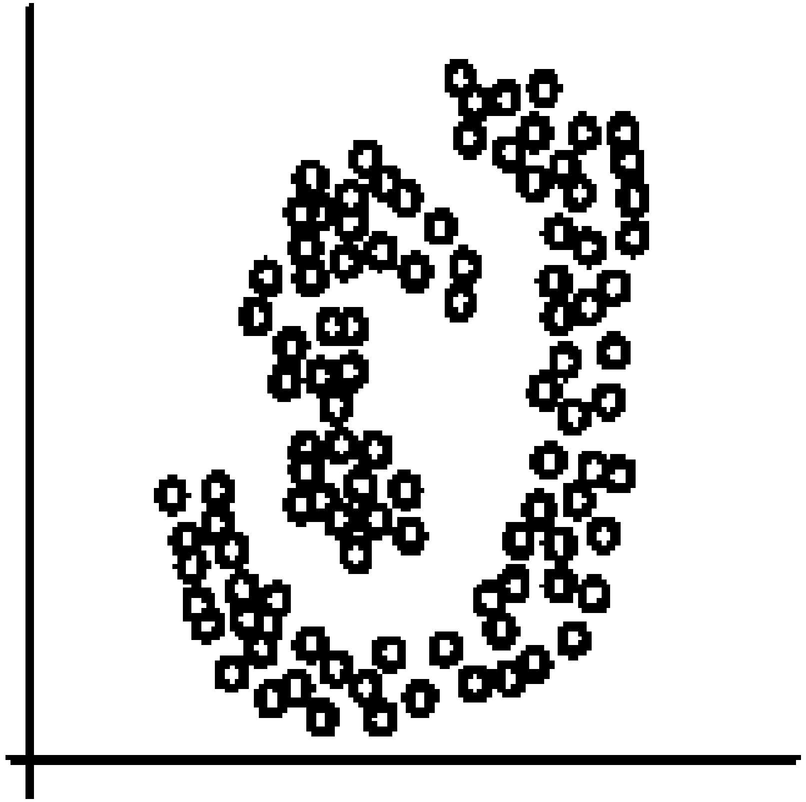
- One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Another method is to perform random sampling. 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

K-means Issues: Sensitivity to Initial Seeds

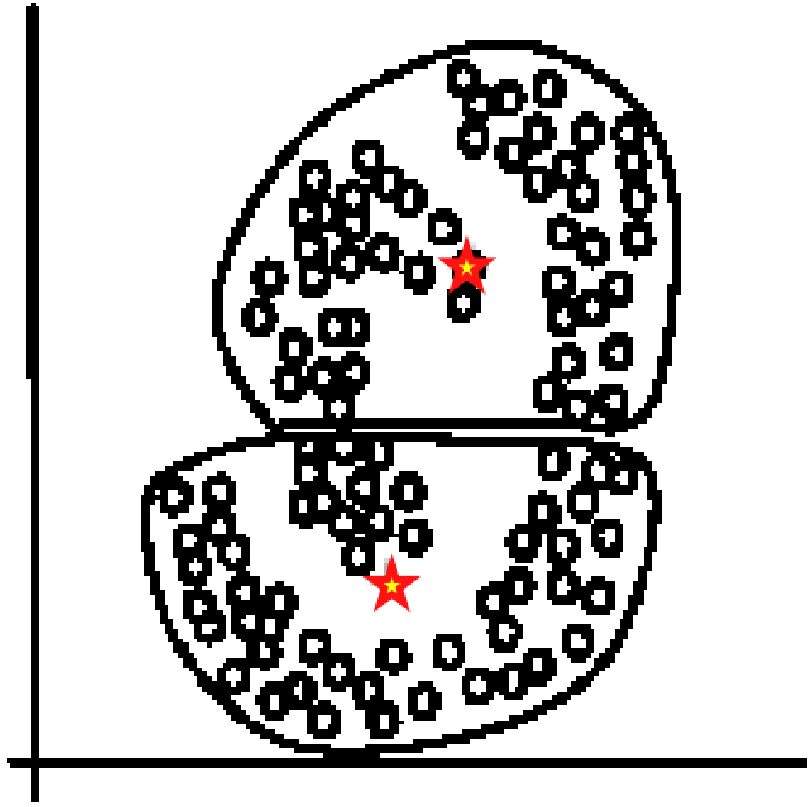


K-means Issues: Special Data Structures

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



Two natural clusters



k-means clusters

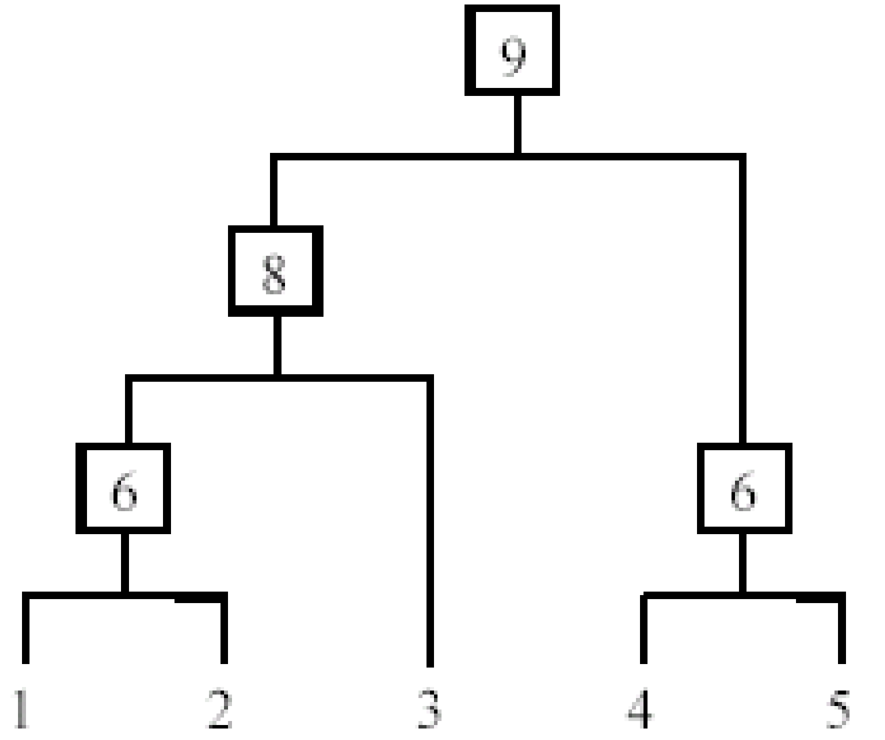
K-means summary

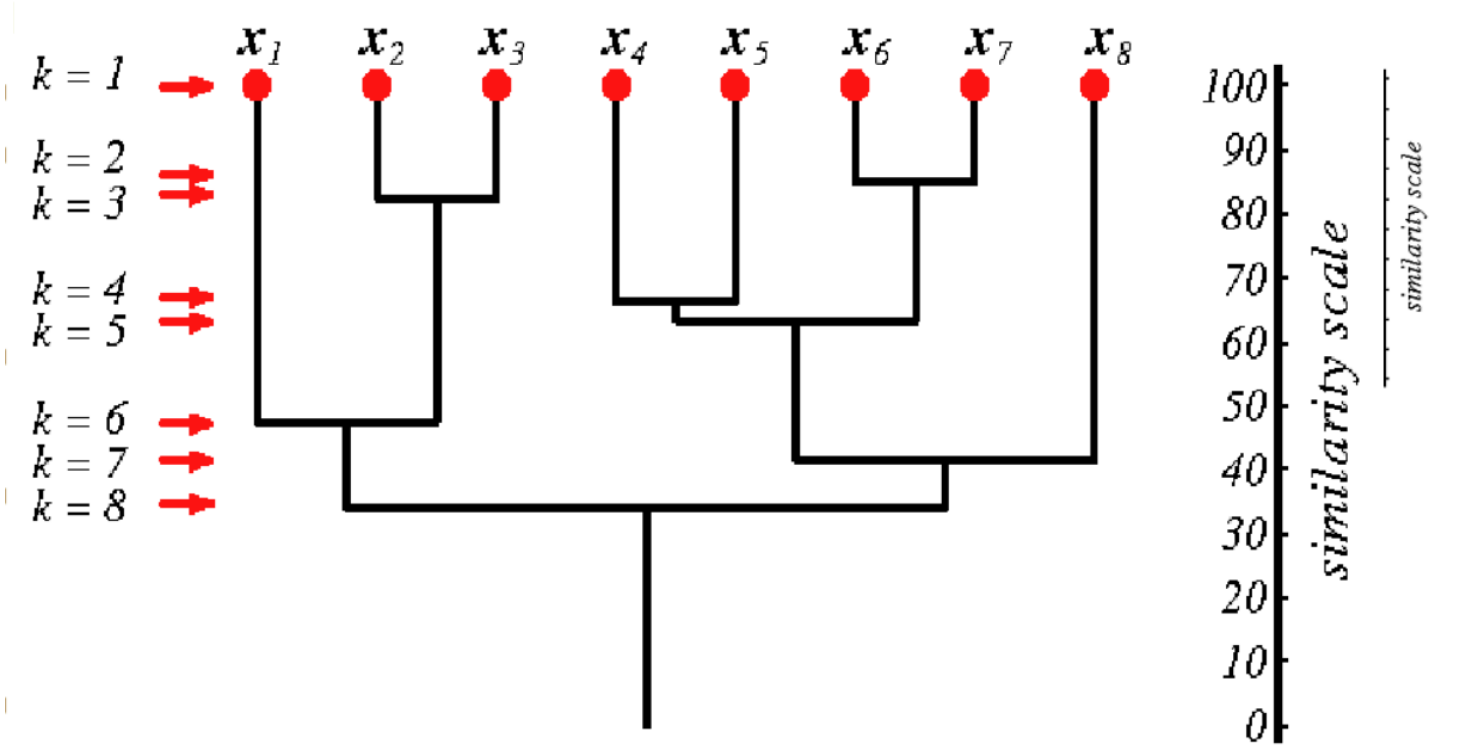
- Despite weaknesses, *k*-means is still the most popular algorithm due to its simplicity, efficiency and
  - other clustering algorithms have their own lists of weaknesses.
- No clear evidence that any other clustering algorithm performs better in general
  - although they may be more suitable for some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

Hierarchical Clustering

Hierarchical Clustering

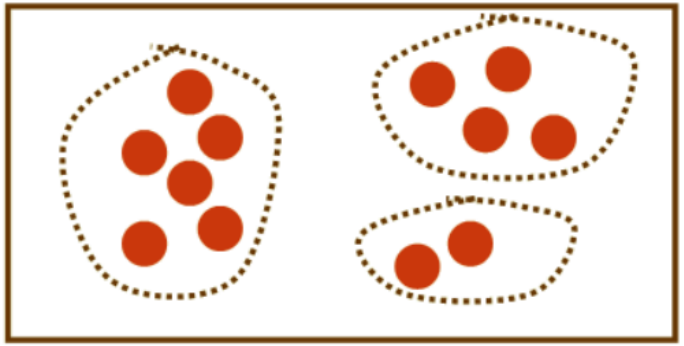
- Produce a nested sequence of clusters, a tree, also called Dendrogram.
- Preferred way to represent a hierarchical clustering.

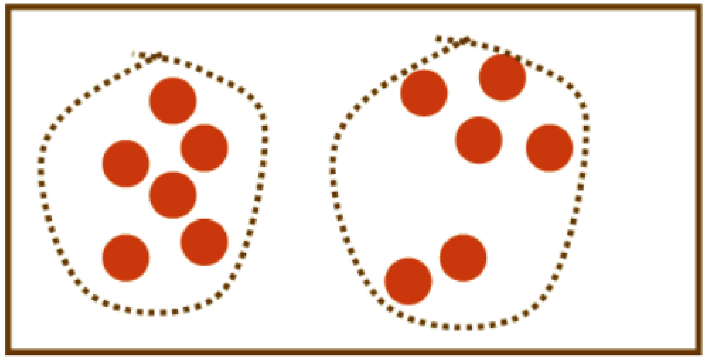




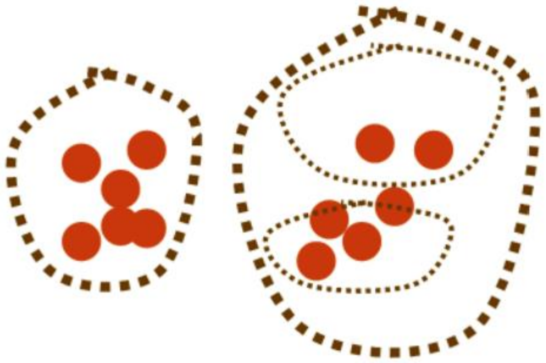
Hierarchical Clustering

- So far we only talked about "flat" clustering.

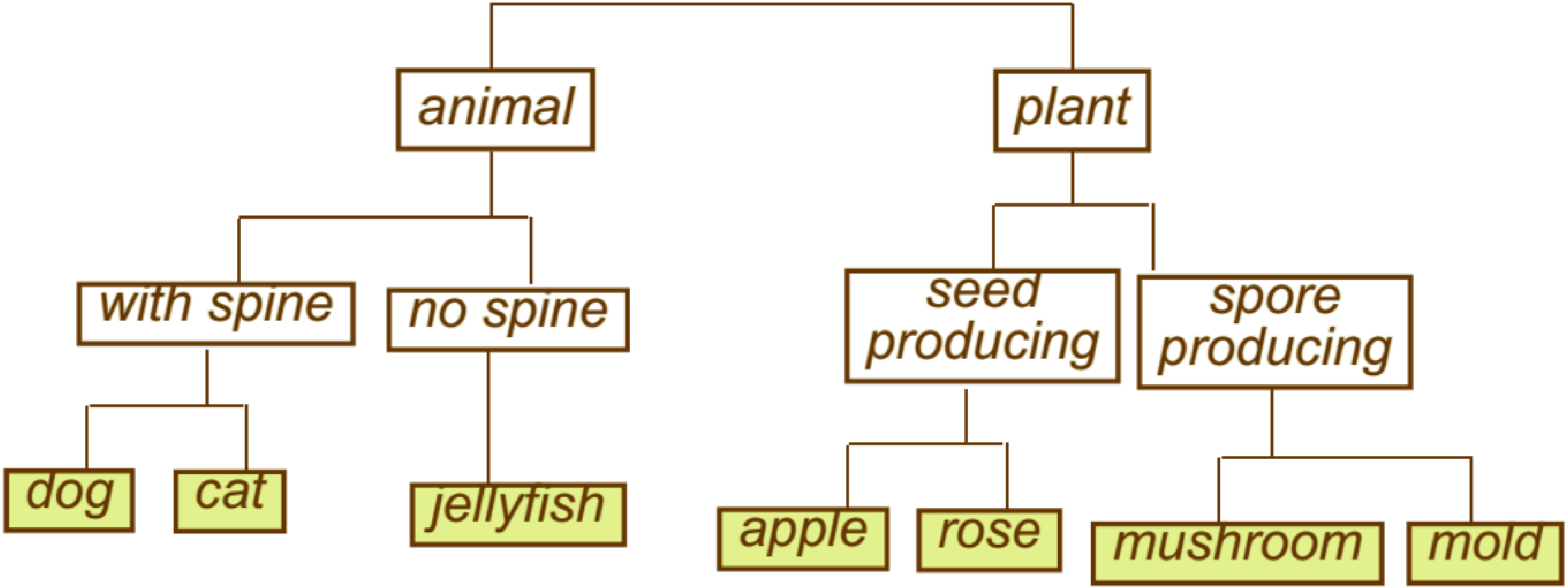




- For some data, hierarchical clustering is more appropriate than "flat" clustering.
- Hierarchical Clustering



Example: Biological Taxonomy

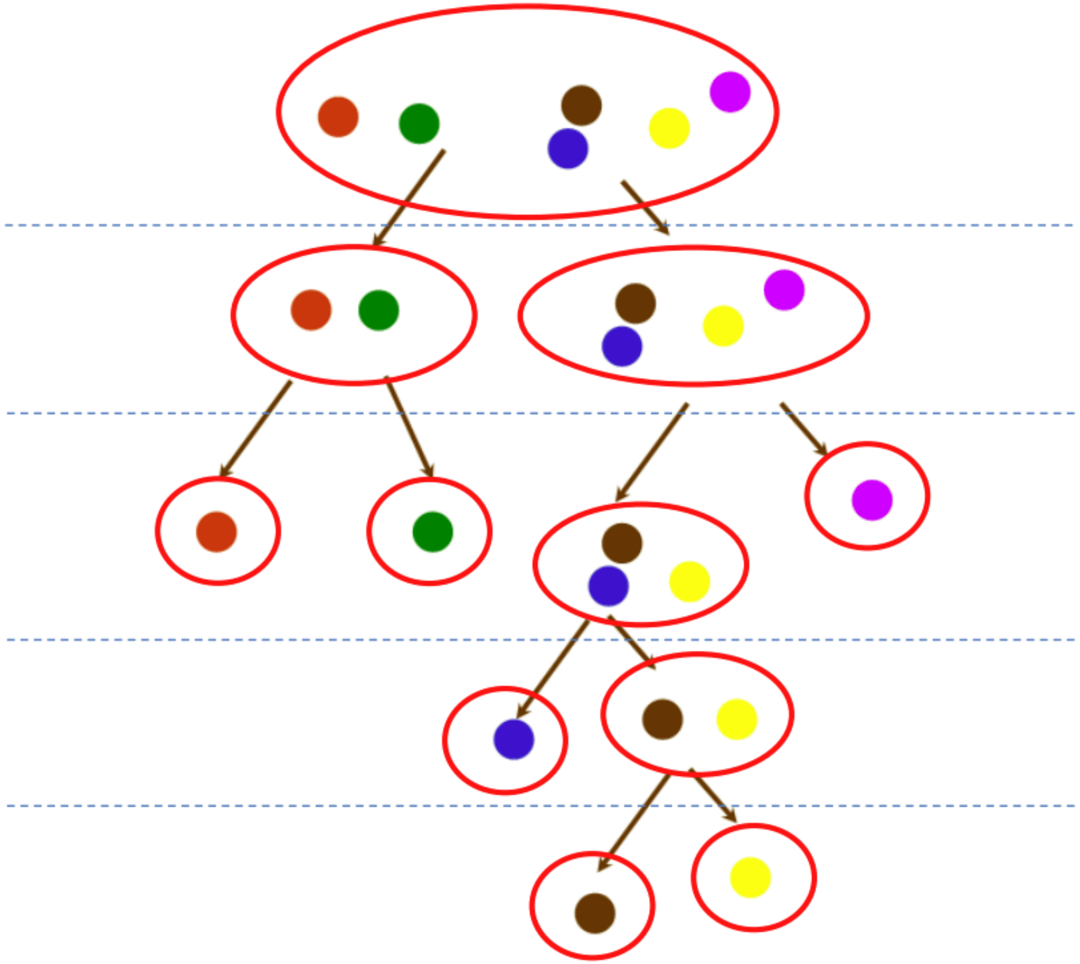


Types of Hierarchical Clustering

- 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

Divisive Hierarchical Clustering

- Any “flat” algorithm which produces a fixed number of clusters can be used.
- Set $c = 2$



Agglomerative Hierarchical Clustering

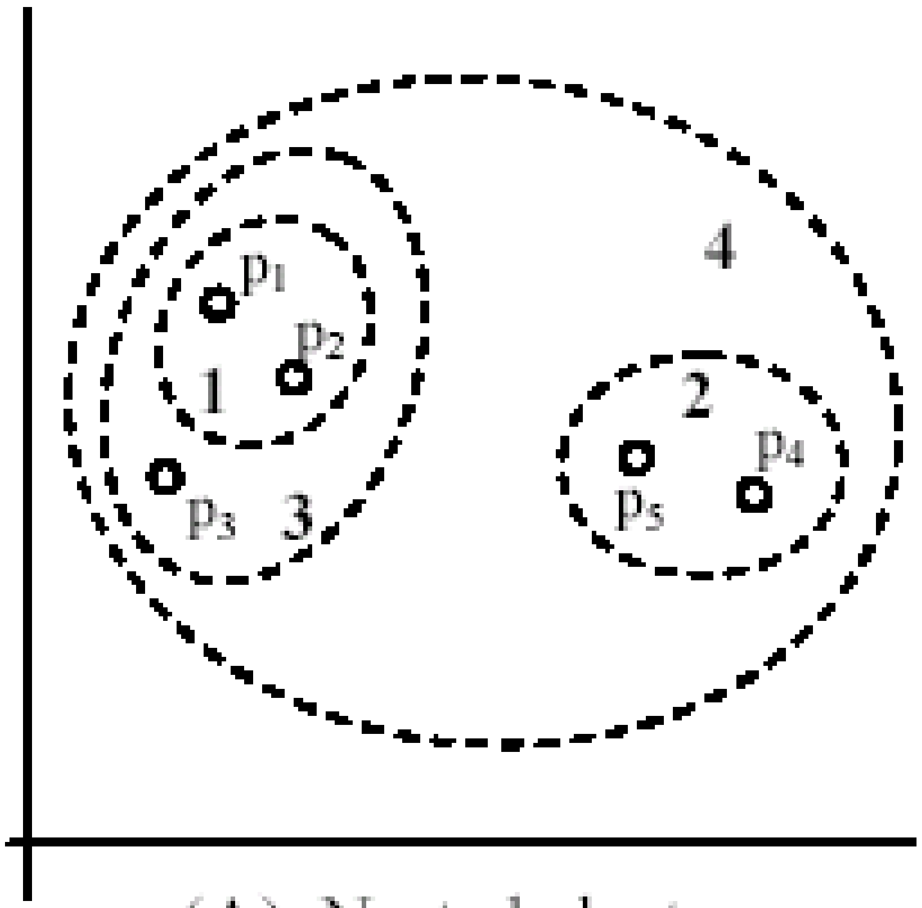
It is more popular then divisive methods.

- 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

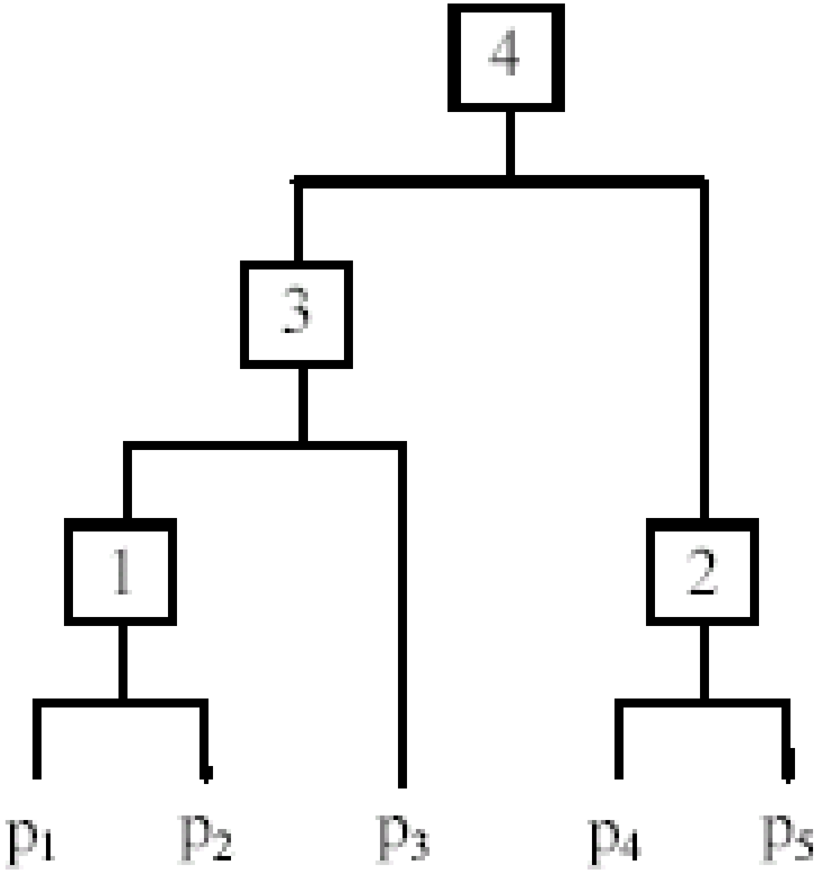
Algorithm Agglomerative(D)

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

An example: Working of the Algorithm



(A). Nested clusters



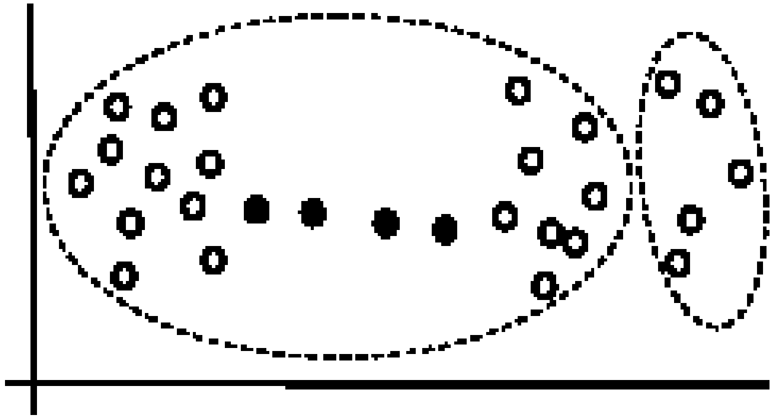
(B) Dendrogram

Measuring the Cluster Distance

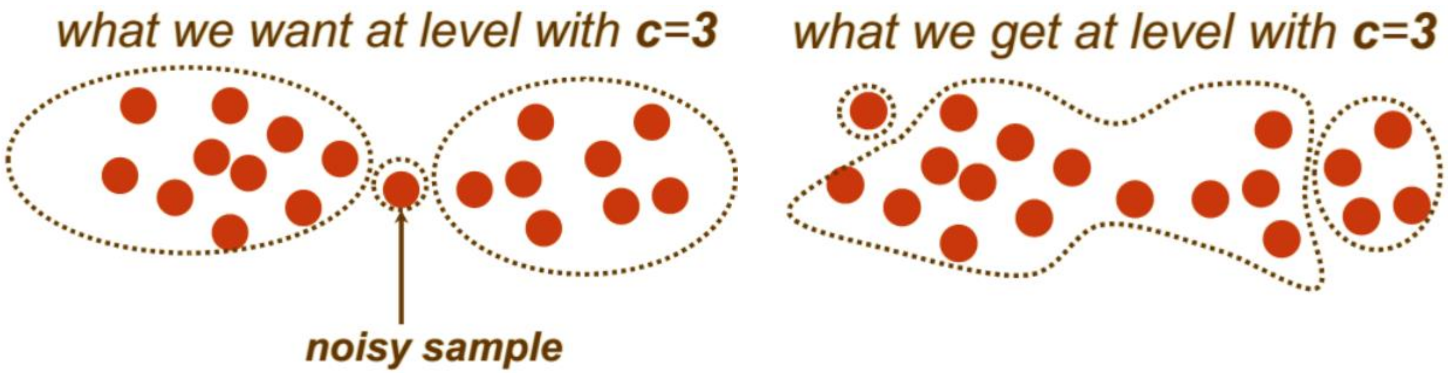
- Four common ways to measure cluster distance
  - Minimum distance $d_{min}(D_i, D_j) = \min_{x \in D_i,\, y \in D_j} \|x - y\|$
  - Maximum distance $d_{max}(D_i, D_j) = \max_{x \in D_i,\, y \in D_j} \|x - y\|$
  - Average distance $d_{avg}(D_i, D_j) = \frac{1}{n_i n_j} \sum_{x \in D_i} \sum_{y \in D_j} \|x - y\|$
  - Mean distance $d_{mean}(D_i, D_j) = \|\mu_i - \mu_j\|$
- A few ways to measure distances of two clusters.
- Results in different variations of the algorithm.
  - Single link
  - Complete link
  - Average link
  - Centroids

Single Link Method (Nearest Neighbor)

- The distance between two clusters is the distance between two closest data points in the two clusters, one data point from each cluster.
- Agglomerative clustering with minimum distance.
- Generates minimum spanning tree.
- Encourages growth of elongated clusters.
- It can find arbitrarily shaped clusters, but
  – It may cause the undesirable "chain effect" by noisy points



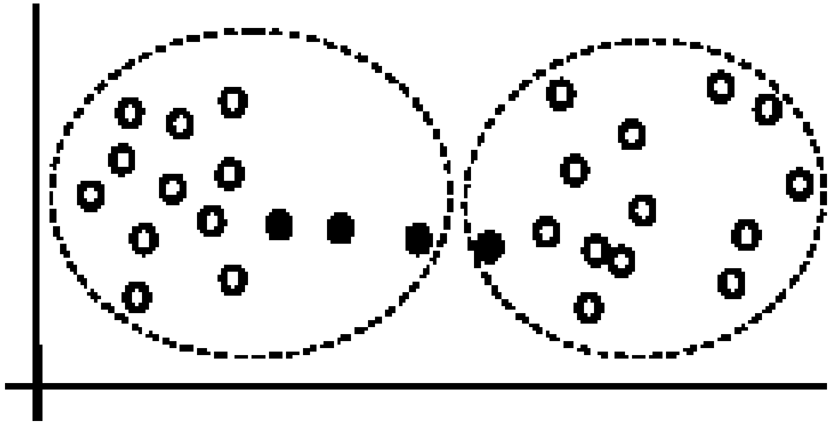
Two natural clusters are split into two

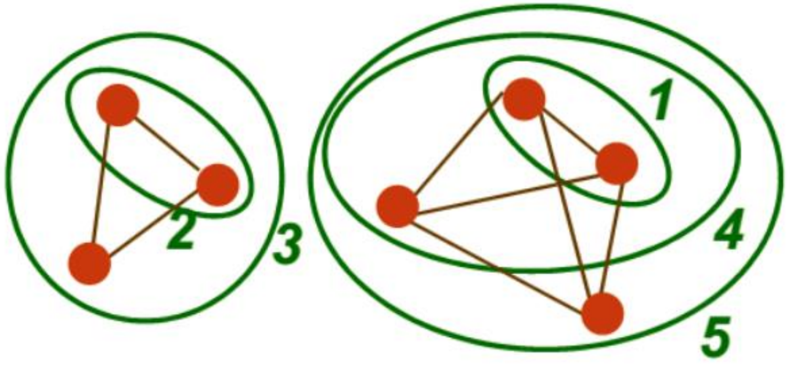


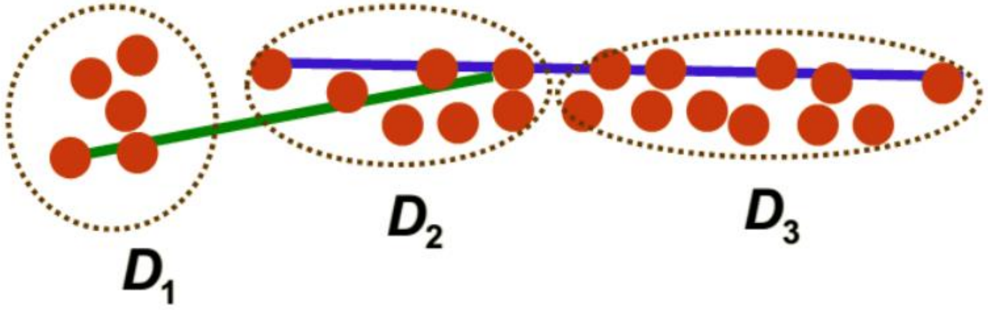
$$d_{min}(D_i, D_j) = \min_{x \in D_i, y \in D_j} \|x - y\|$$

Complete Link Method (Farthest Neighbor)

- The distance between two clusters is the distance of two furthest data points in the two clusters.
- Agglomerative clustering with maximum distance
- Encourages compact clusters
- It is sensitive to outliers because they are far away
- Does not work if elongated clusters are present
  - $d_{max}(D_1, D_2) < d_{max}(D_2, D_3)$
  - Thus D_1 and D_2 are merged instead of D_2 and D_3.







$$d_{max}(D_i, D_j) = \max_{x \in D_i, y \in D_j} \|x - y\|$$

Average Link and Centroid Methods

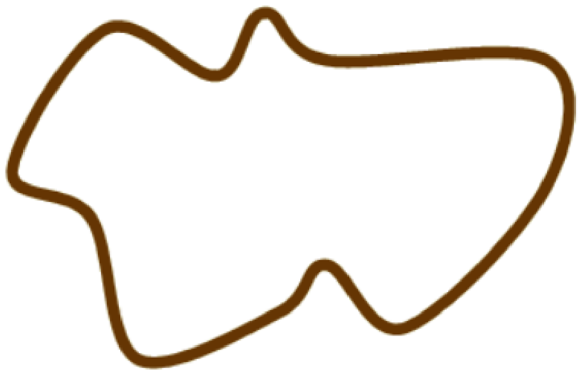
- Average link: A compromise between
  - the sensitivity of complete-link clustering to outliers and
  - the tendency of single-link clustering to form long chains that do not correspond to the intuitive notion of clusters as compact, spherical objects.
  - In this method, the distance between two clusters is the average distance of all pair-wise distances between the data points in two clusters.
- Centroid method: In this method, the distance between two clusters is the distance between their centroids

Divisive vs. Agglomerative

- Agglomerative is faster to compute, in general
- Divisive may be less “blind” to the global structure of the data.

Divisive

When taking the first step (split), have access to all the data; can find the best possible split in 2 parts.



Agglomerative

When taking the first step merging, do not consider the global structure of the data, only look at pairwise structure.

How to choose a clustering algorithm

- Clustering research has a long history. A vast collection of algorithms are available.
  - We only introduced several main 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.
- 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).

Cluster Evaluation

Cluster Evaluation is a 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.

Evaluation Measures: 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:

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

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

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

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

$$purity(D_i) = \max_j(\Pr_i(c_j)) \quad (31)$$

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

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

An example

Example 14: Assume we have a text collection D of 900 documents from three topics (or three classes), Science, Sports, and Politics. Each class has 300 documents. Each document in D is labeled with one of the topics (classes). We use this collection to perform clustering to find three clusters. Note that 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.
- I**nter-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.

Summary

Summary

- Clustering is has along history and still is in active research
  - 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
- 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.
- Competitive learning in neuronal networks performs clustering analysis of the input data

References

- “CS583 - Chapter 4: Unsupervised Learning” by Bing Liu
- “Class 13 - Unsupervised learning – Clustering” by Shimon Ullman, Tomaso Poggio, Danny Harari, Daneil Zysman, and Darren Seibert” by amalalhait
- “Machine Learning” by Andrew Ng