Applied Machine Learning

Distance, Neighbors & Dimensionality

Computer Science, Fall 2022
Instructor: Xuhong Zhang

Clustering

- ***What is clustering?***
  - Given some unlabeled data, the organization of them into similarity groups called clusters, is called clustering.

- Q: Is clustering supervised or unsupervised?

Historic application of clustering

- *John Snow*, a 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.

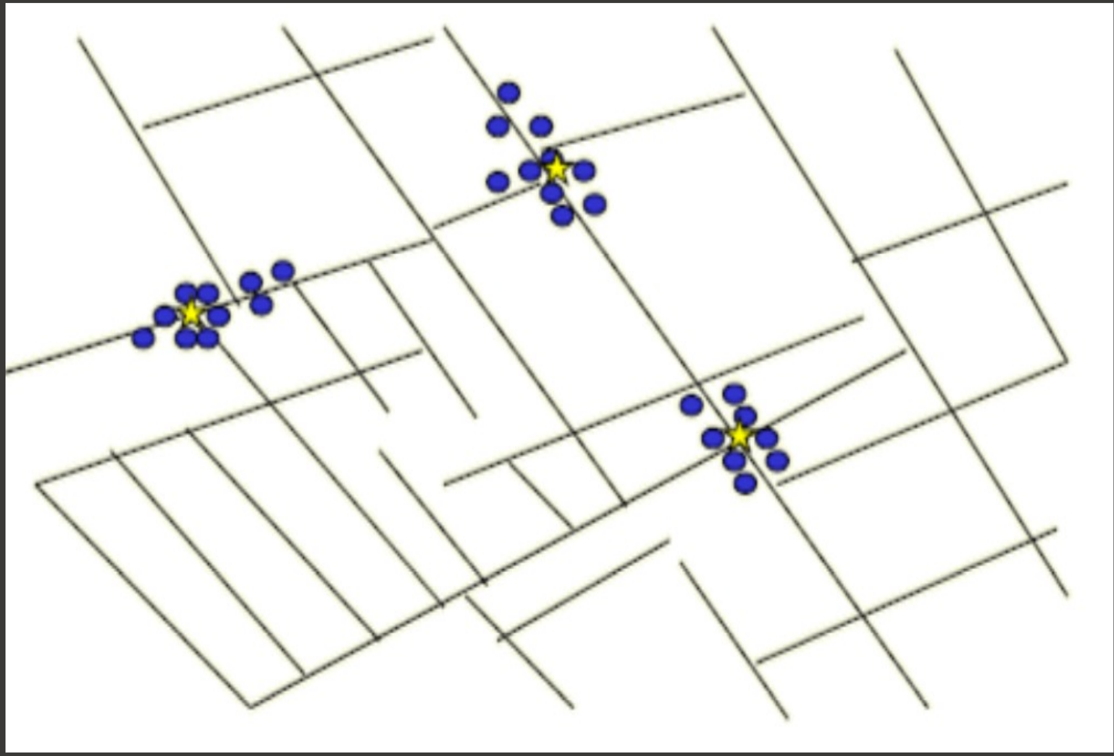
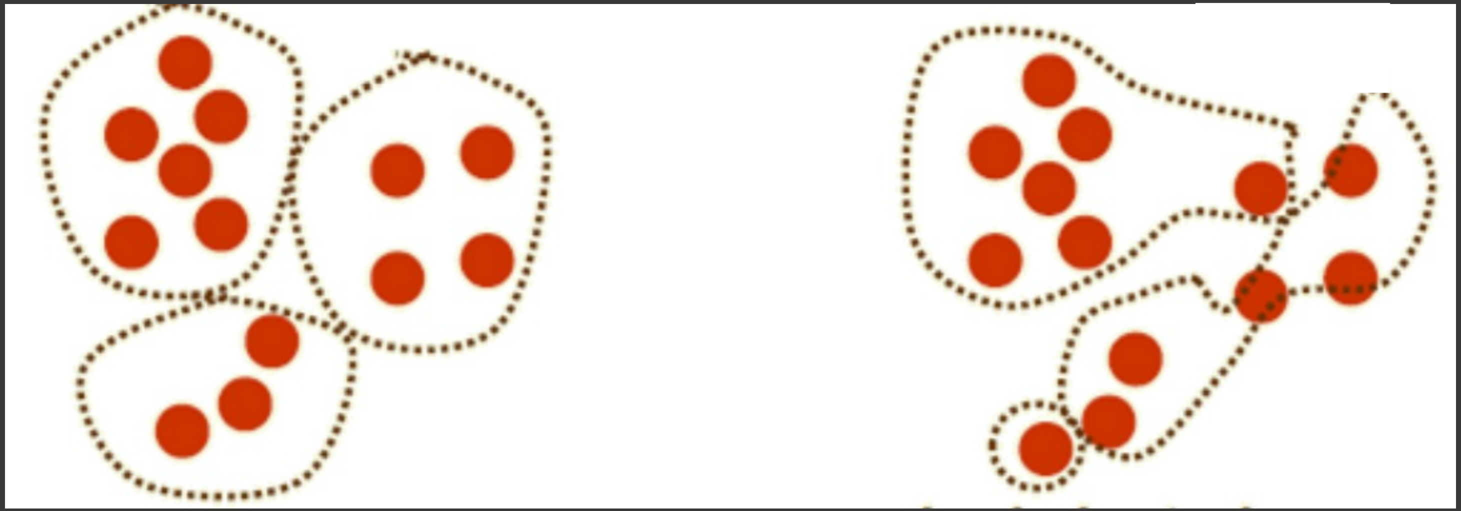


Image Source: Nina Mishra HP Labs

What do we need for clustering

- Similarity/Dissimilarity Measure
  - Similarity measure: large if x_i, x_j are similar
  - Dissimilarity measure (or distance): small if x_i, x_j are similar
- Criterion function to evaluate a clustering



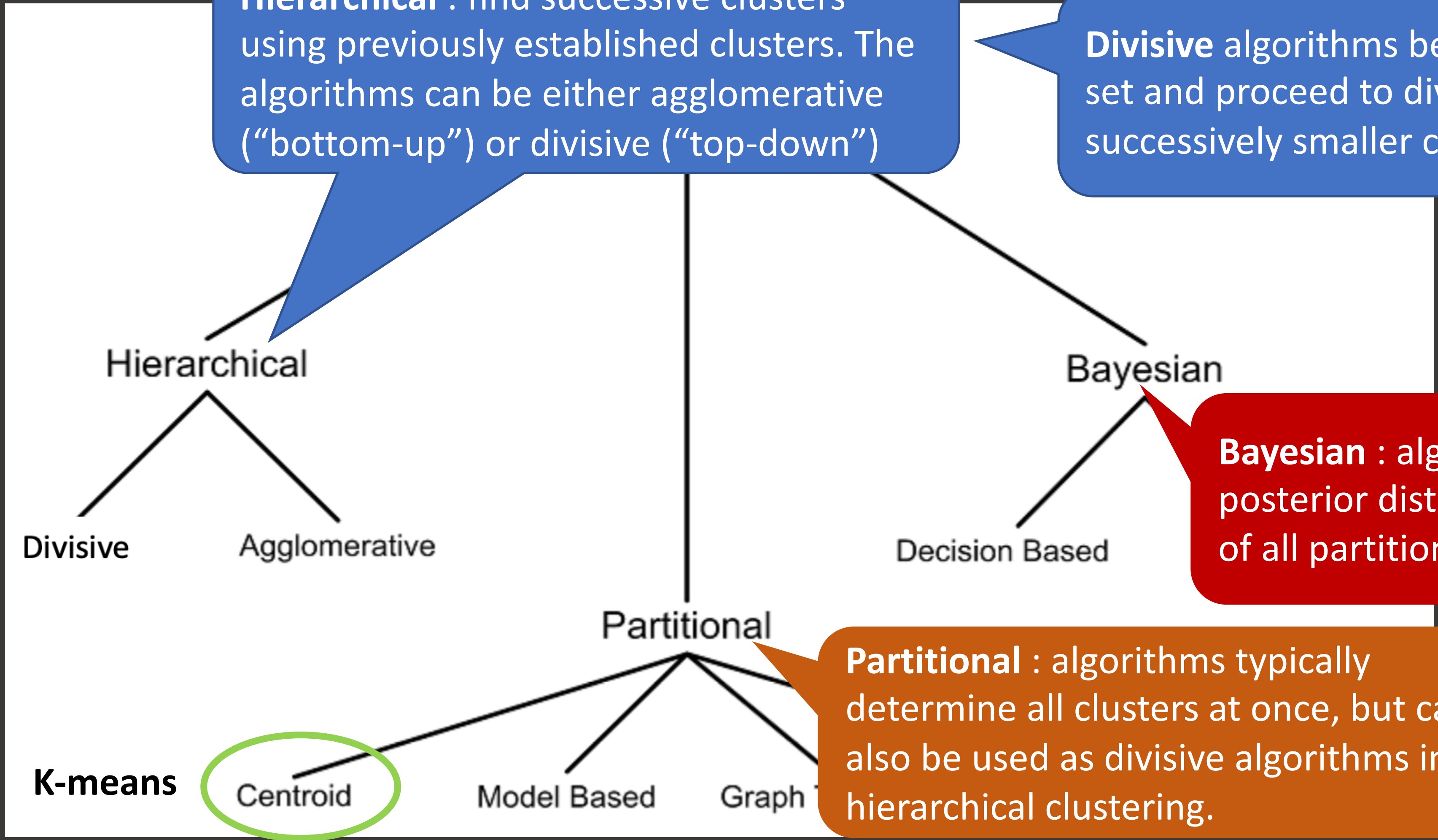
Good ? vs. Bad ?

Clustering techniqu

Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters

Hierarchical : find successive clusters using previously established clusters. The algorithms can be either agglomerative (“bottom-up”) or divisive (“top-down”)

Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.



Bayesian : algorithms try to generate a posterior distribution over the collection of all partitions of the data.

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

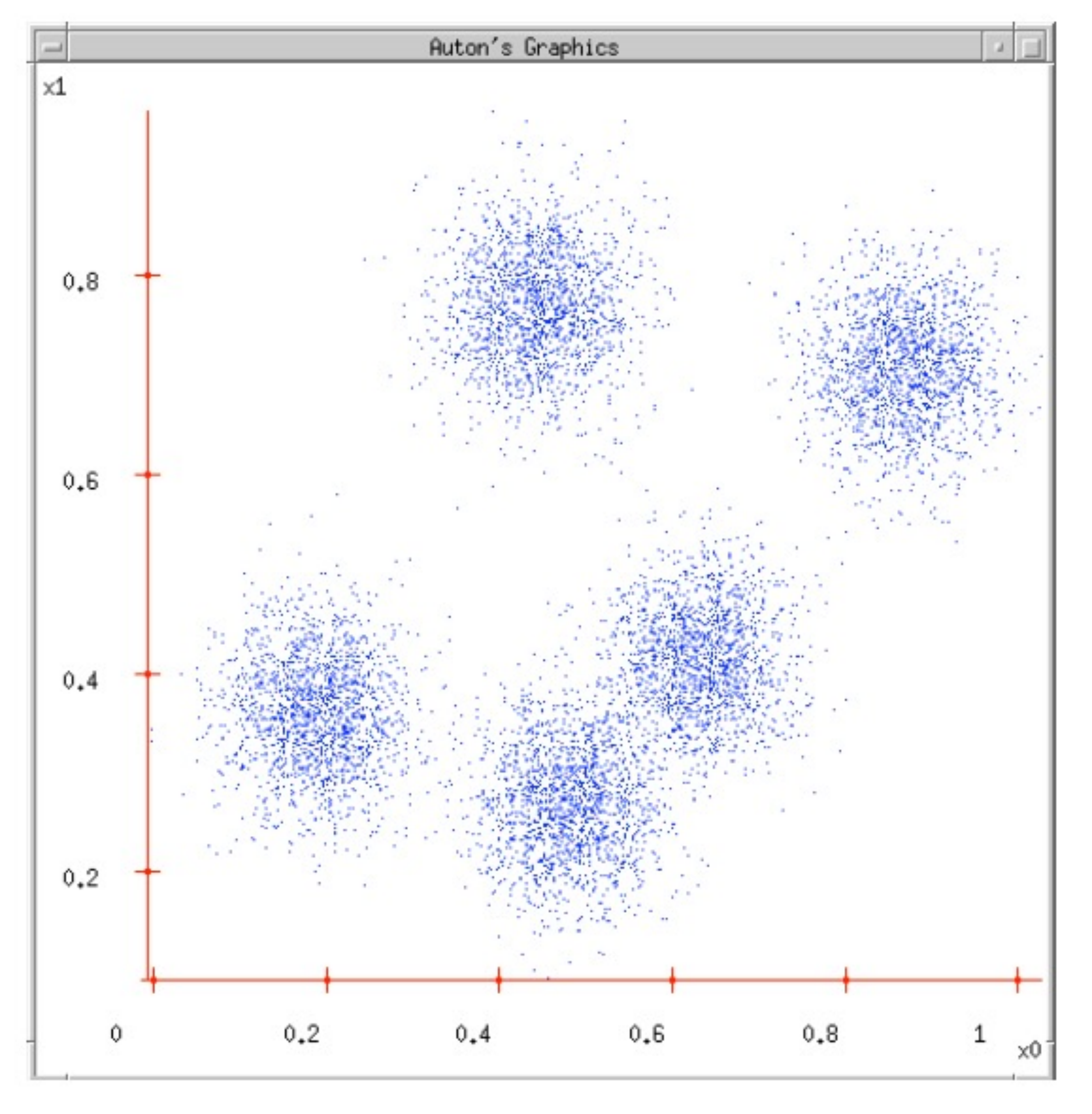
K-Means Clustering

- Supervised learning used labeled data pairs (x, y) to learn a mapping $f: x \rightarrow \mathrm{y}$
    - But, what if we don't have labels?
- No labels : unsupervised learning
    - Only some points are labeled : semi-supervised learning
        - Labels may be expensive to obtain, so we only get a few
    - Clustering is the unsupervised grouping of data points.
        - It can be used for knowledge/pattern discovery

K-Means

- K-Means is a partitional clustering algorithm
- Given a set of data points $\mathcal{D} = \{x_1, \dots, x_n\}$ where $x_i \in \mathbb{R}^d$, and d is the number of dimensions.
- The K-Means algorithm partitions the given data into k clusters with:
  - Each cluster has a cluster center, called centroid.
  - K is specified by the user.

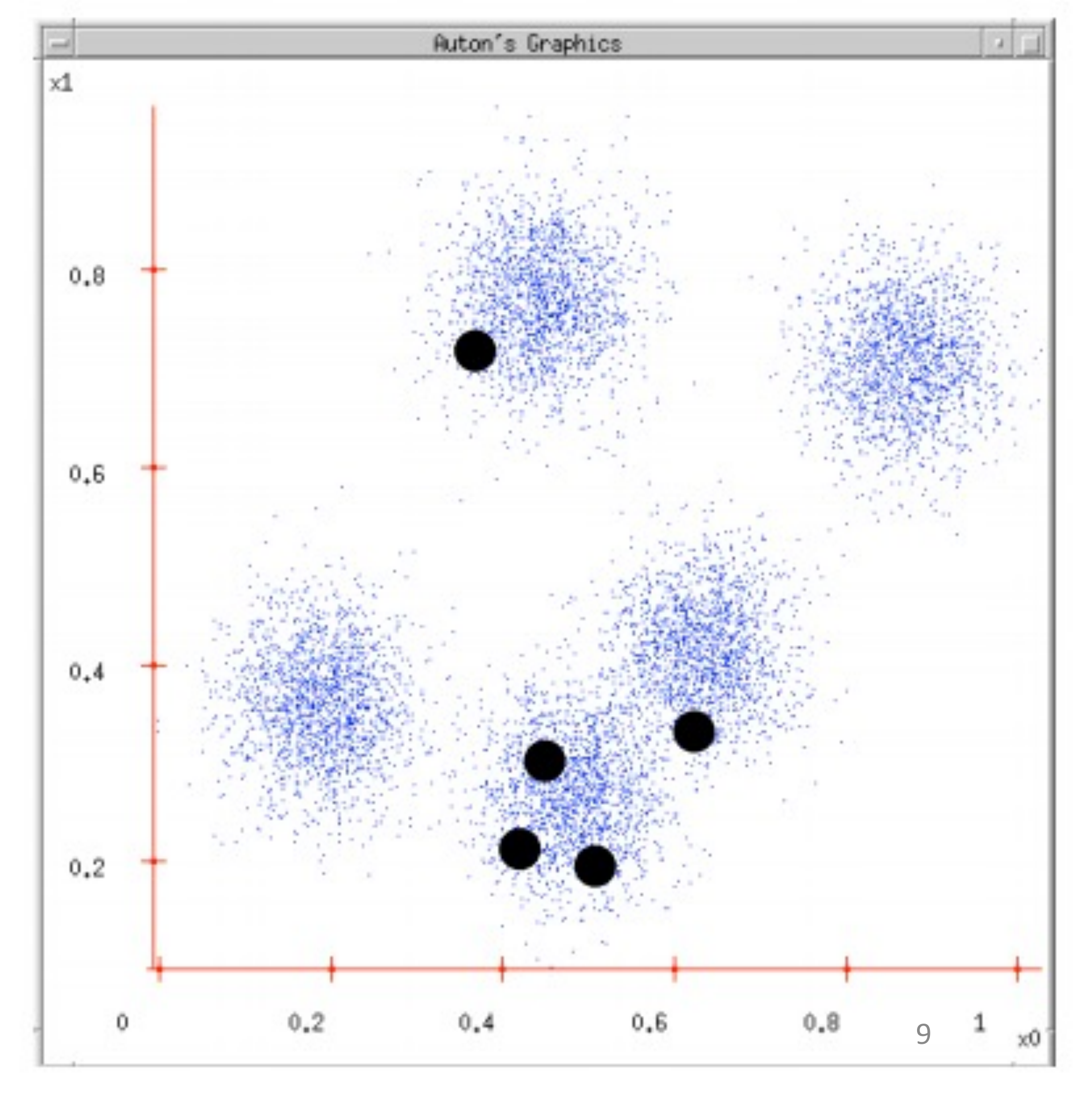
K-Means Clustering



K-Means Clustering

K-Means (k, X)

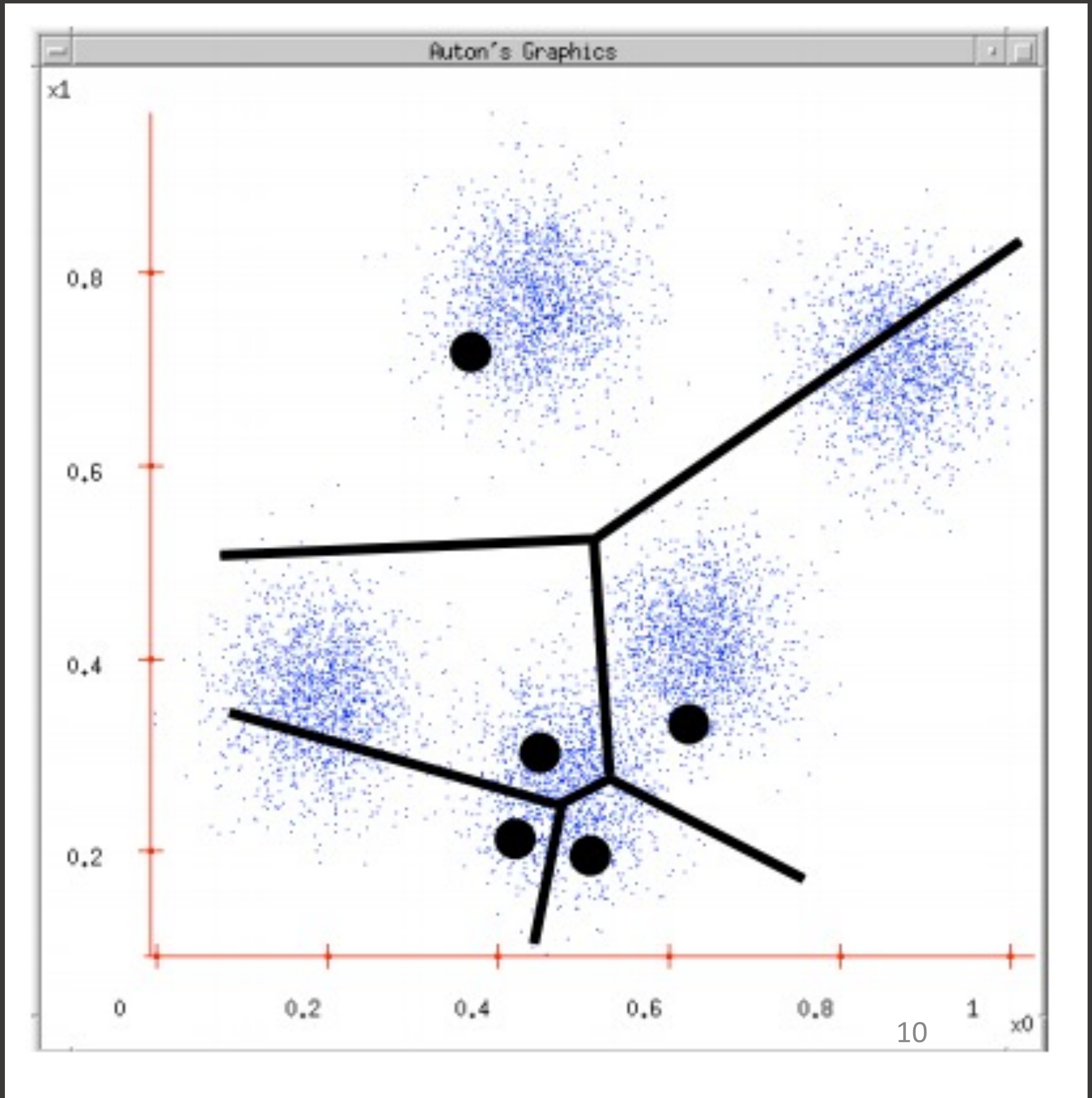
- o Randomly choose k cluster center locations (centroids)
- o Loop until convergence
  - • Assign each point to the cluster Of the closest centroid
  - • Re-estimate the cluster centroids Based on the data assigned to each cluster



K-Means Clustering

K-Means (k, X)

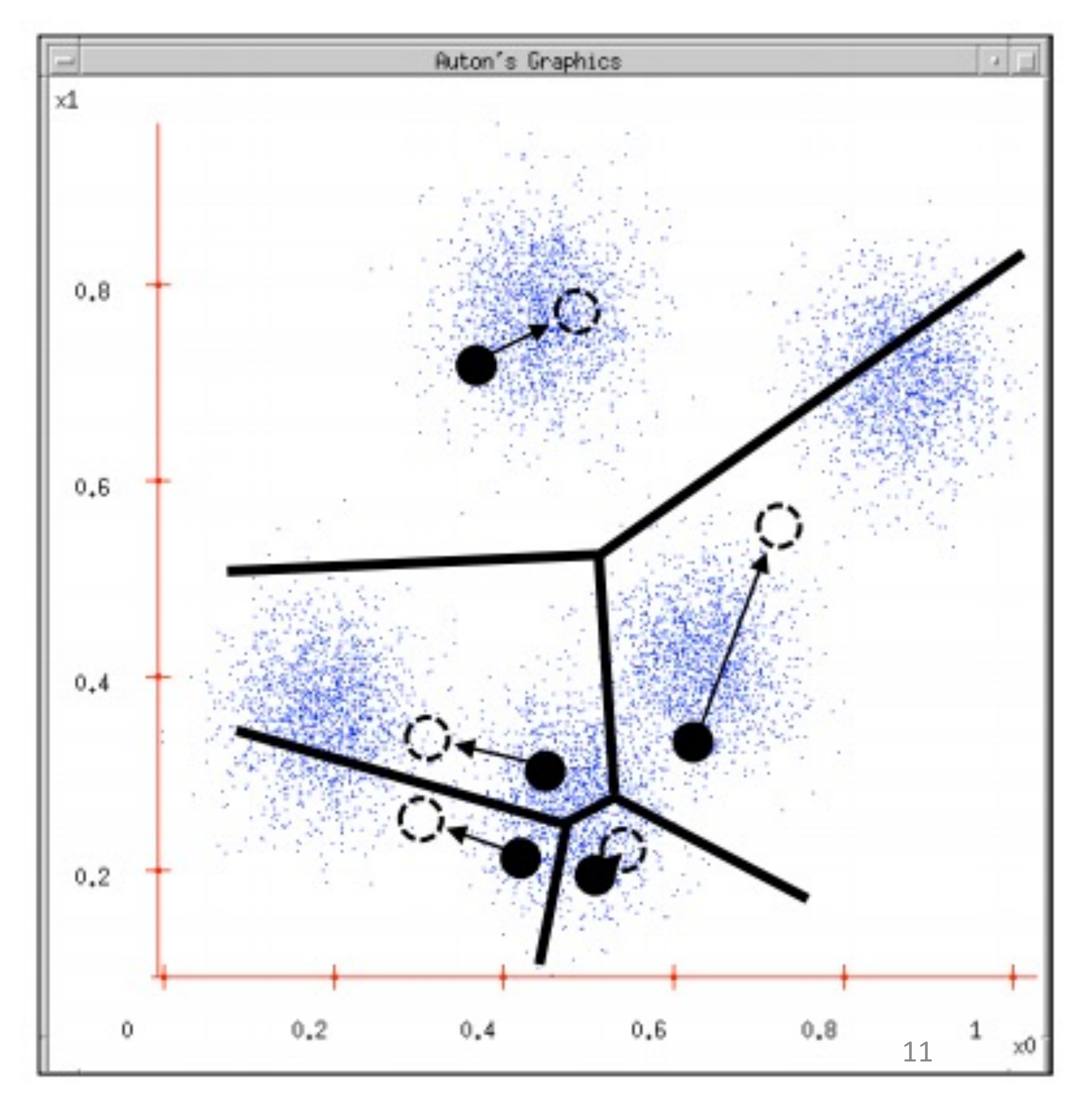
- Randomly choose k cluster center locations (centroids)
- Loop until convergence
  - Assign each point to the cluster Of the closest centroid
  - Re-estimate the cluster centroids Based on the data assigned to each cluster



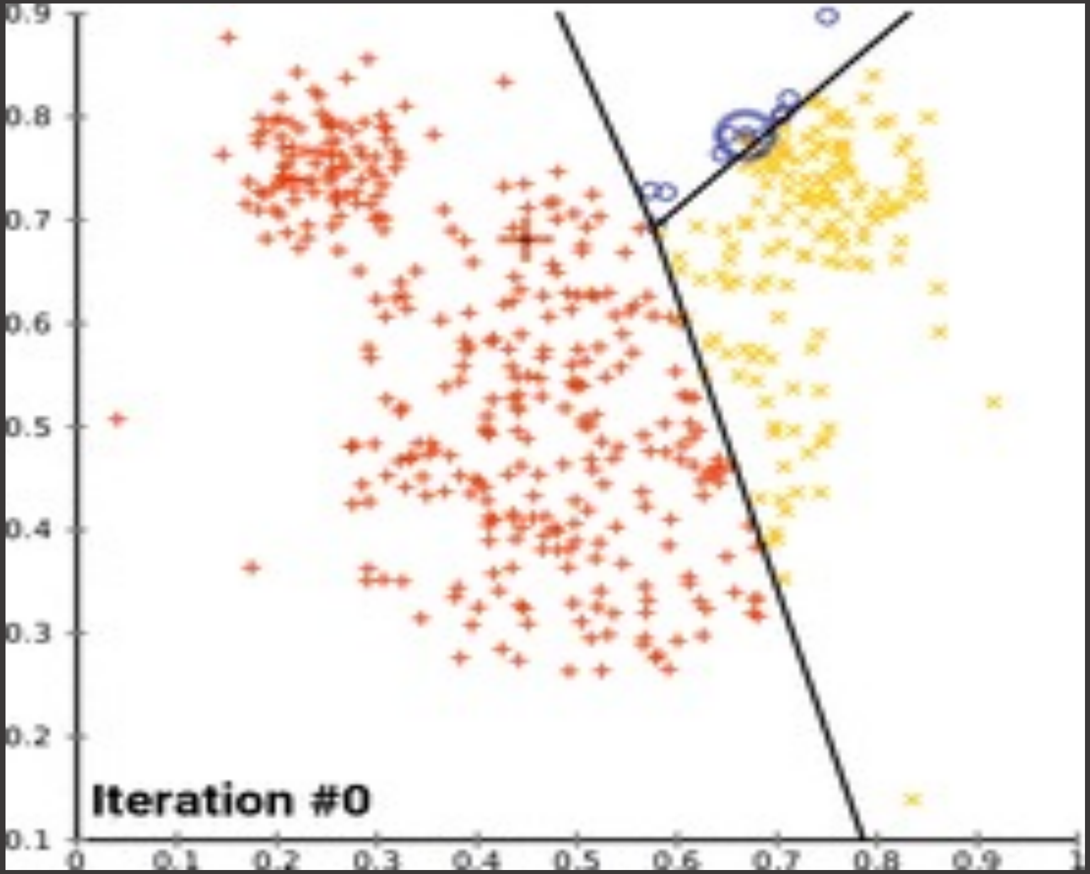
K-Means Clustering

K-Means (k, X)

- Randomly choose k cluster center locations (centroids)
- Loop until convergence
  - Assign each point to the cluster Of the closest centroid
  - Re-estimate the cluster centroids Based on the data assigned to each cluster



K-Means Clustering



Example from Wikipedia

K-Means Clustering

- With initial set of k means $c_1^{(1)}, c_2^{(1)}, \ldots, c_k^{(1)}$
- Do:
  - Assignment step:
    - $S_i^{(t)} = \left\{x_p : ||x_p - c_i^{(t)}||^2 \leq ||x_p - c_j^{(t)}||^2, \forall j, 1 \leq j \leq k\right\}$,
  - Update step:
    - $c_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$
- Until no change

K-Means convergence (stopping) criterion

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

$$SSE = \sum_{j=1}^{k} \sum_{x \in C'_j} d(x, C_j)^2$$

  - C'_j is the j^{th} cluster,
  - C_j is the centroid of cluster C'_j (the mean vector of all the data points in C'_j)
  - $d(x, C_j)$ is the (Euclidean)distance between data point x and centroid C_j

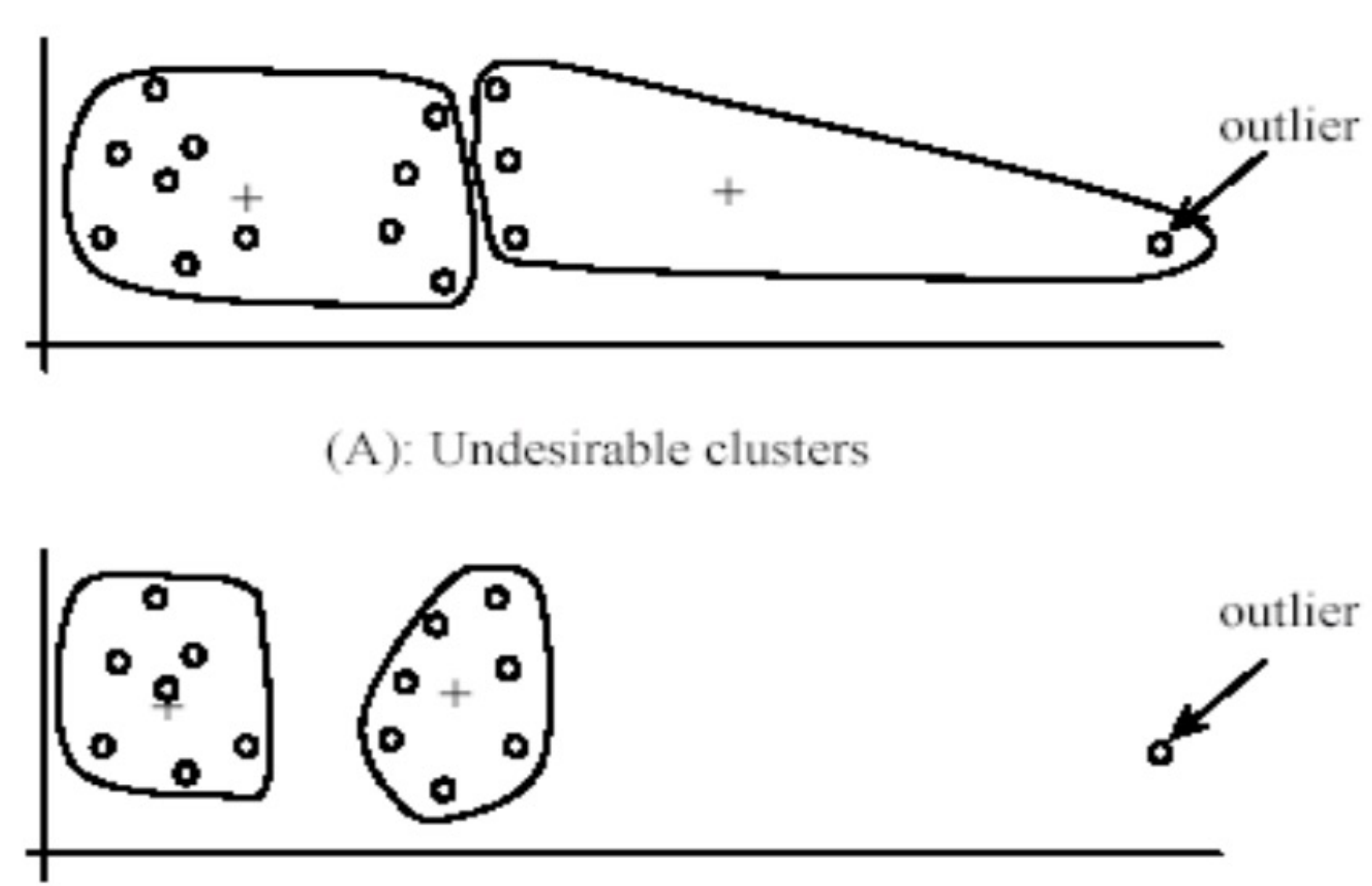
Why use K-Means?

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Question: What is the Time Complexity?
    - ❑ N is the number of data points,
    - ❑ K is the number of clusters,
    - ❑ T is the number of iterations,
    - ❑ Answer: $O(TKN)$.
  - Since both k and t are small, k-means is considered a linear algorithm.
  - K-means is the most popular clustering algorithm.
  - Note: if SSE is used, it terminates at the local optimum. The global optimum is hard to find due to complexity.

Why use K-Means?

- Weakness
  - The algorithm is defined only when the mean is defined.
    - K-mode is defined as the most frequent values for categorical variables.
  - The user needs to specify K, which is arbitrary.
    - Learn the optimal k for the clustering
      - Note that this requires a performance measure
  - The algorithm is sensitive to outliers.
    - Outliers could be errors in the data recording or some special data points with very different values.

Outliers?



(A): Undesirable clusters

(B): Ideal clusters

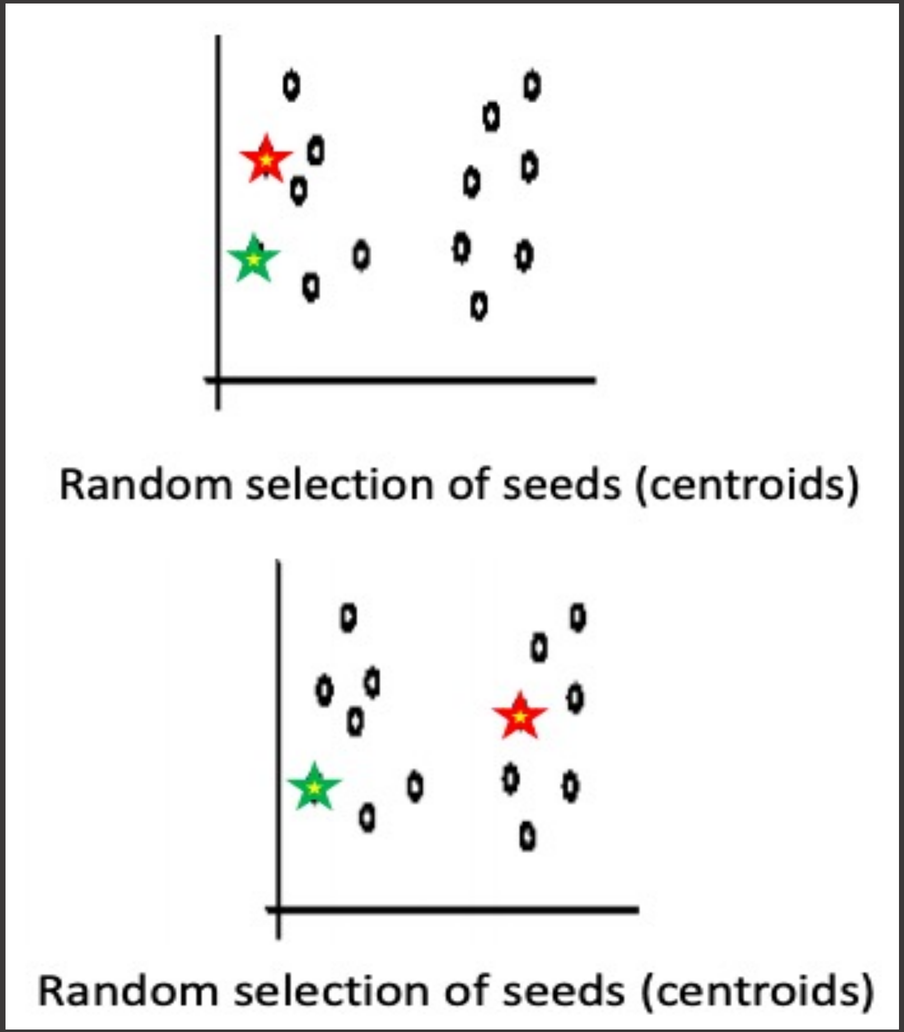
How to deal with outliers?

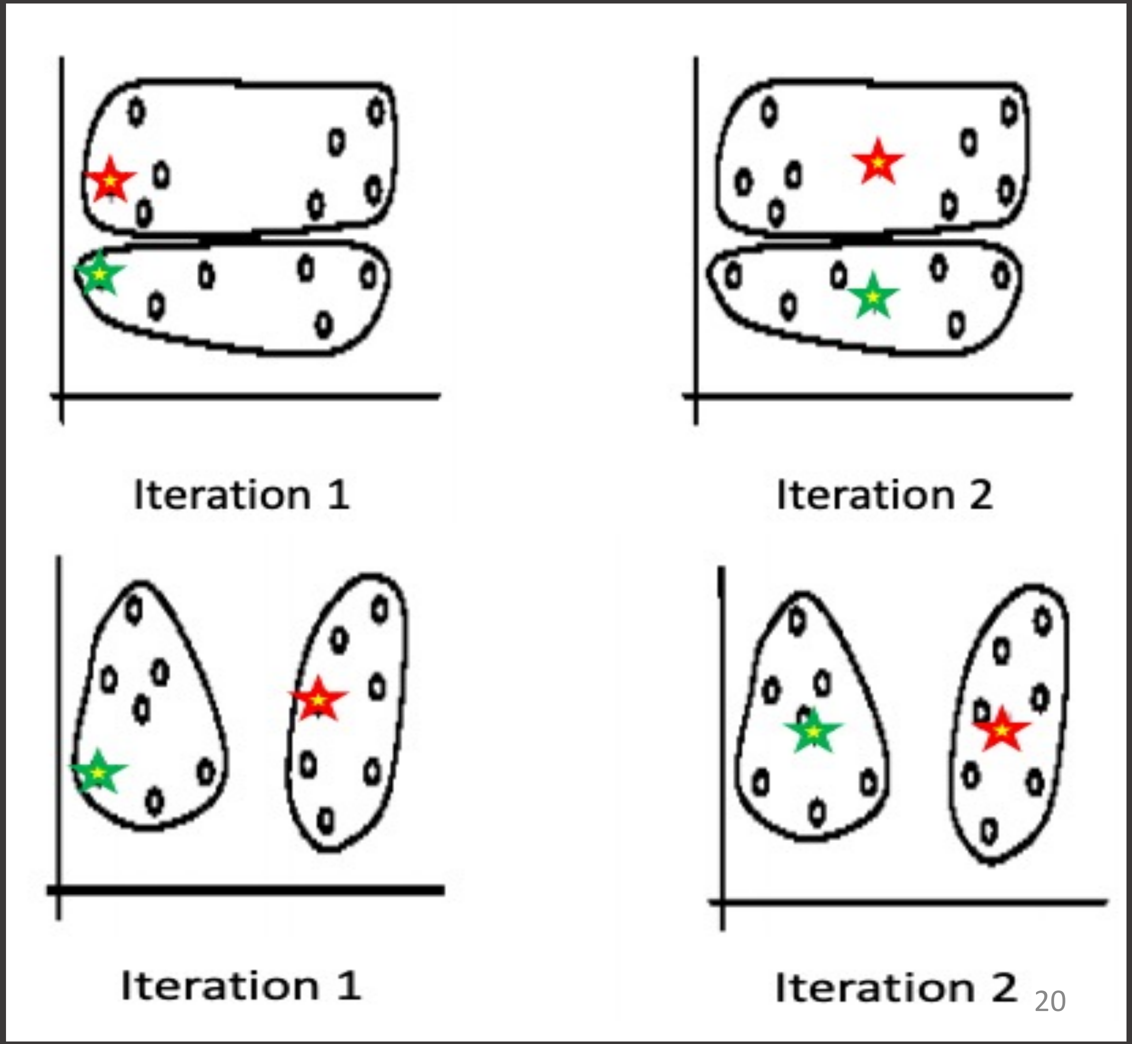
- Remove some data points that are much further away from the centroids than other data points
  - But be careful. It's better to monitor the potential outliers over a few iterations and then decide to remove them or not
- Perform random sampling: by choosing a small subset of the data points, the chance of selecting an outlier is much smaller
  - Then assign the rest of the data points to the clusters by distance or similarity comparison

Initial points

- **Very sensitive** to the initial points
  - Do many runs of K-Means, each with different initial centroids
  - Seed the centroids using a better method than randomly choosing the centroids
    - e.g., Farthest-first sampling

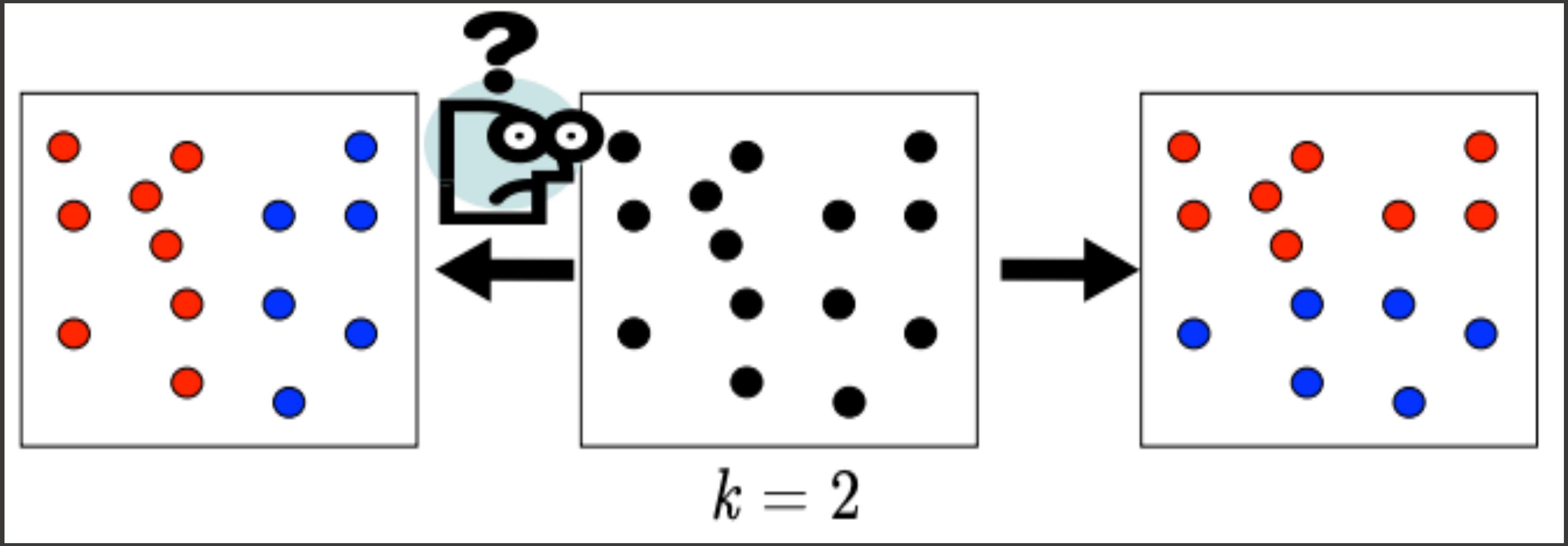
K-Means is sensitive to initial seeds





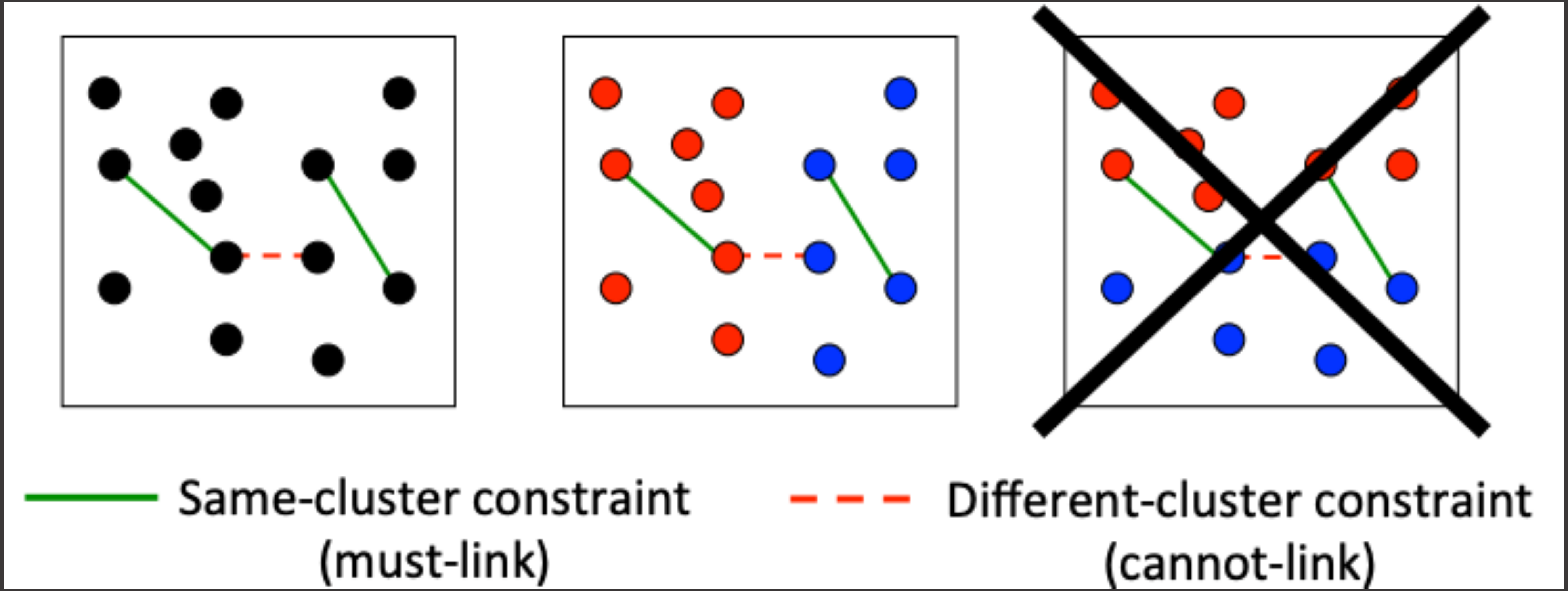
Clustering Analysis

- How can you tell which clustering you want?

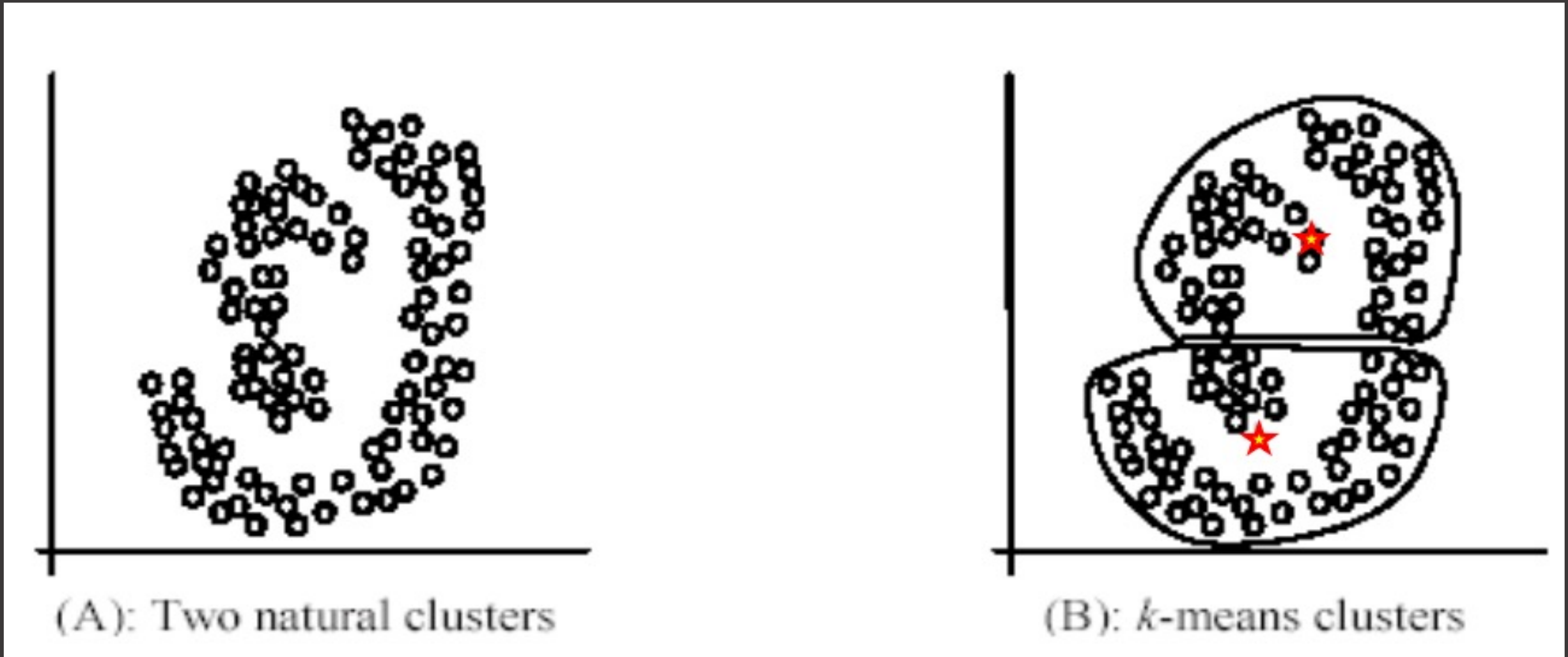


Clustering Analysis

- Constrained clustering techniques (semi-supervised)



Special data structure



Clustering Analysis

- How can you evaluate your clustering result?
  - Within-cluster variances vs. between-cluster variances
  - Which other algorithm we discussed using this criteria?

Summary of K-Means

- Despite weaknesses, k-means is still the most popular algorithm due to its simplicity and efficiency
- No clear evidence that any other clustering algorithm performs better in general
- Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!

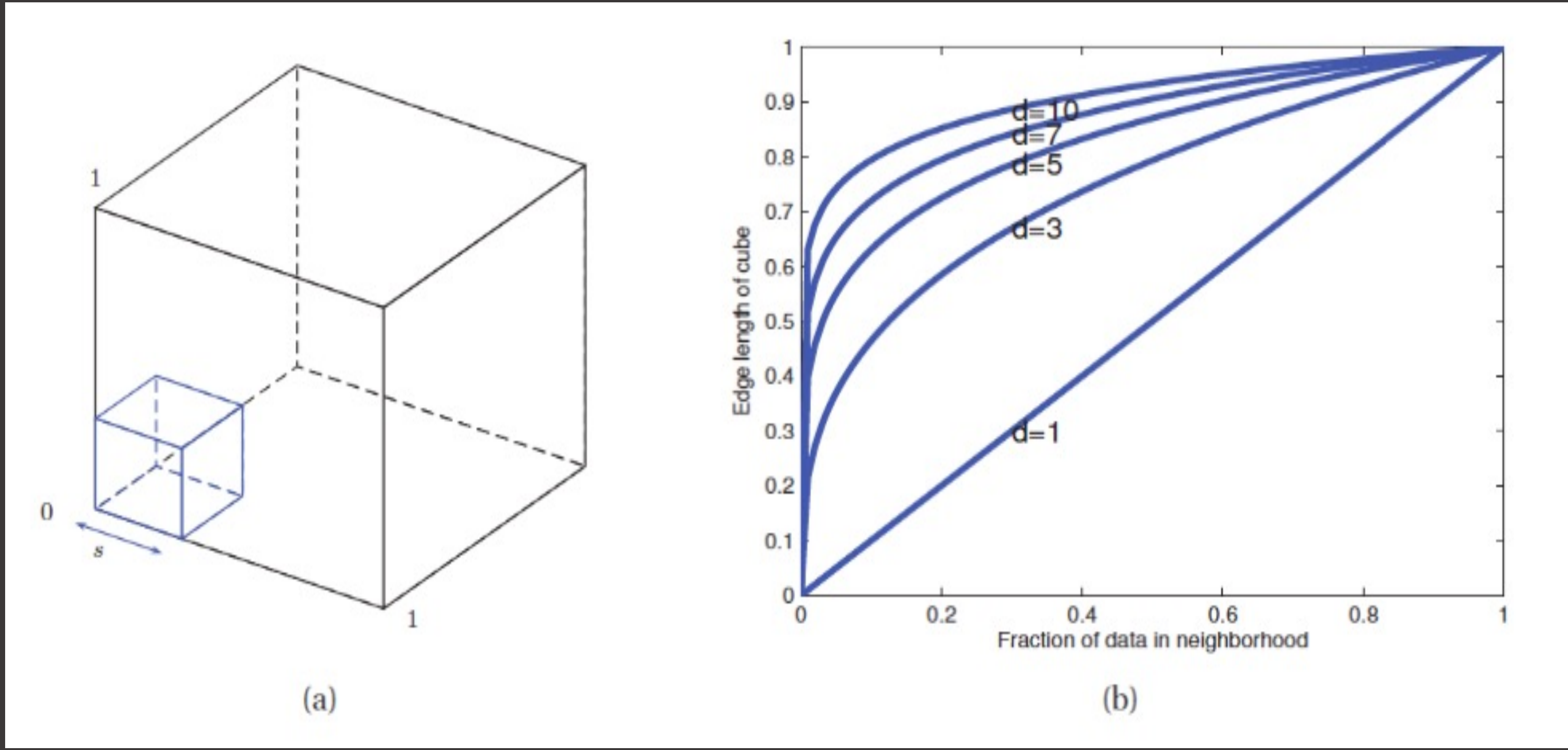
Curse of Dimensionality

- Suppose we have a D-dimensional cube, and a sub-cube with edge length s, f is the proportion of data we have within the sub-cube. Suppose data is uniformly distributed within the D-dimensional cube.

- Question: How long is the edge s if I want my sub-cube contains f of the data?
  - $s = e_D(f) = f^{\frac{1}{D}}$

Curse of Dimensionality

- $s = e_D(f) = f^{\frac{1}{D}}$
  - $e_{10}(0.01) = 0.063$
  - $e_{10}(0.1) = 0.8$



Curse of Dimensionality

- The kNN algorithm is particularly susceptible to the *curse of dimensionality*.

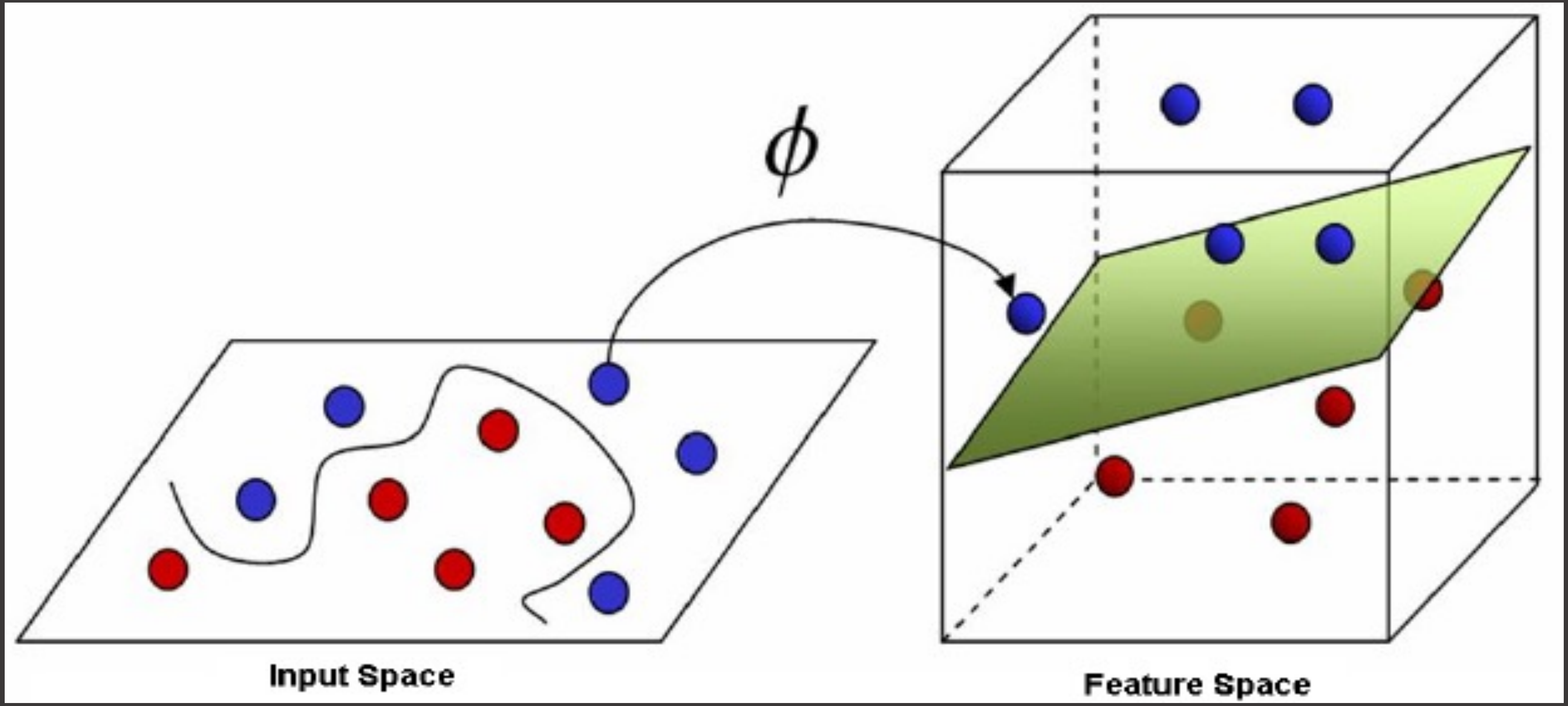
- In machine learning, the curse of dimensionality refers to scenarios with a fixed size of training examples but an increasing number of dimensions and range of feature values in each dimension in a high-dimensional feature space.

Curse of Dimensionality

- In kNN an increasing number of dimensions becomes increasingly *problematic* because the more dimensions we add, the larger the volume in the hyperspace needs to be capture a fixed number of neighbors.
- As the volume grows larger and larger, the "neighbors" become less and less "similar" to the query point as they are now all relatively distant from the query point.

Blessing of dimensionality

- Increase linear separability



Computational Complexity

- The Big-O notation is used to study the asympototic behavior of functions. In the context of algorithms in computer science, the ***Big-O*** notation is most commonly used to measure the ***time complexity*** or runtime of an algorithm for the ***worst case*** scenario (Often, it is also used to measure memory requirements).

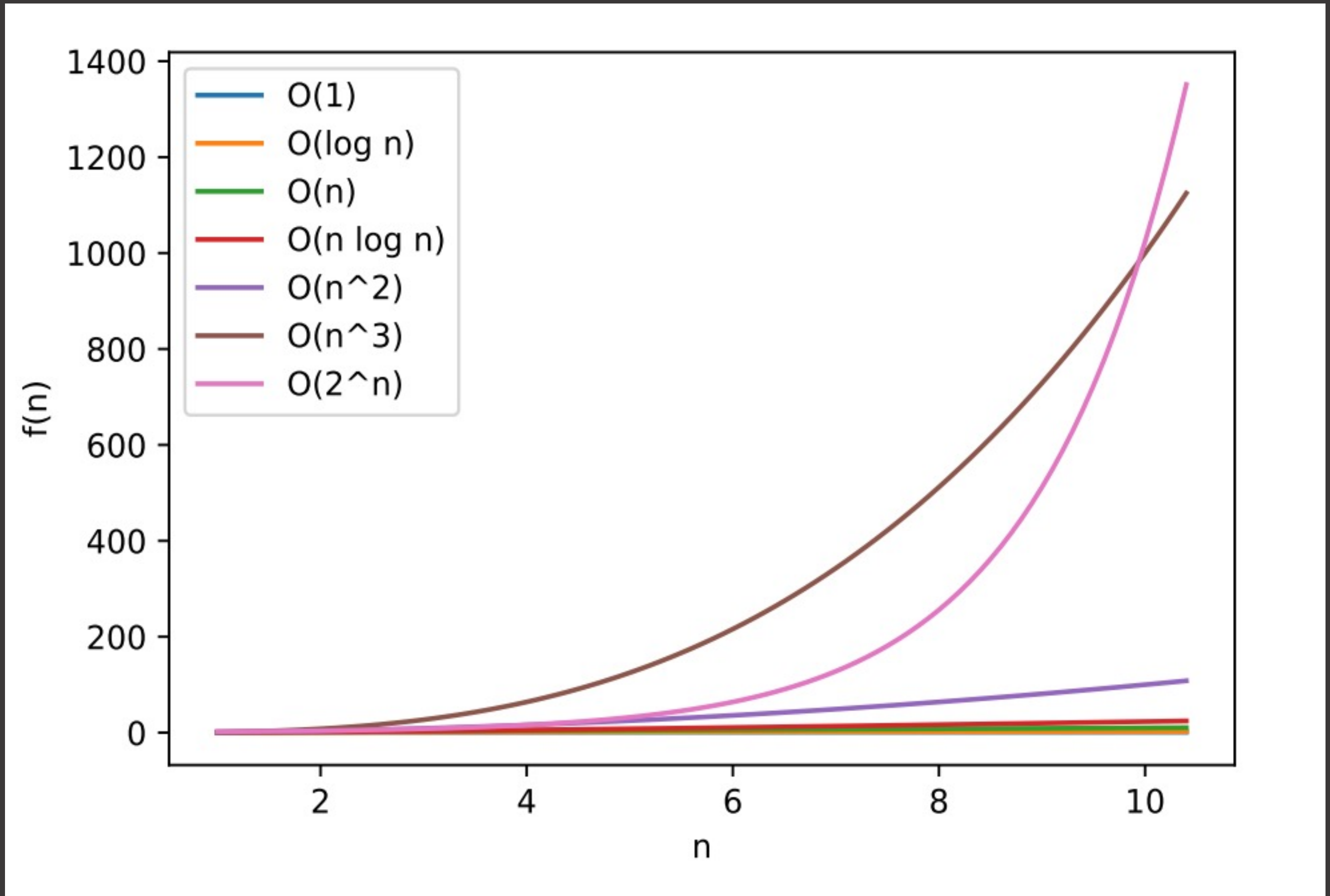
Computational Complexity

- The basic concepts

f(n)	Name
1	Constant
$\log n$	Logarithmic
n	Linear
$n \log n$	Log Linear
n^2	Quadratic
n^3	Cubic
n^c	Higher-level polynomial
2^n	Exponential

Computational Complexity

- The growth rates of common functions

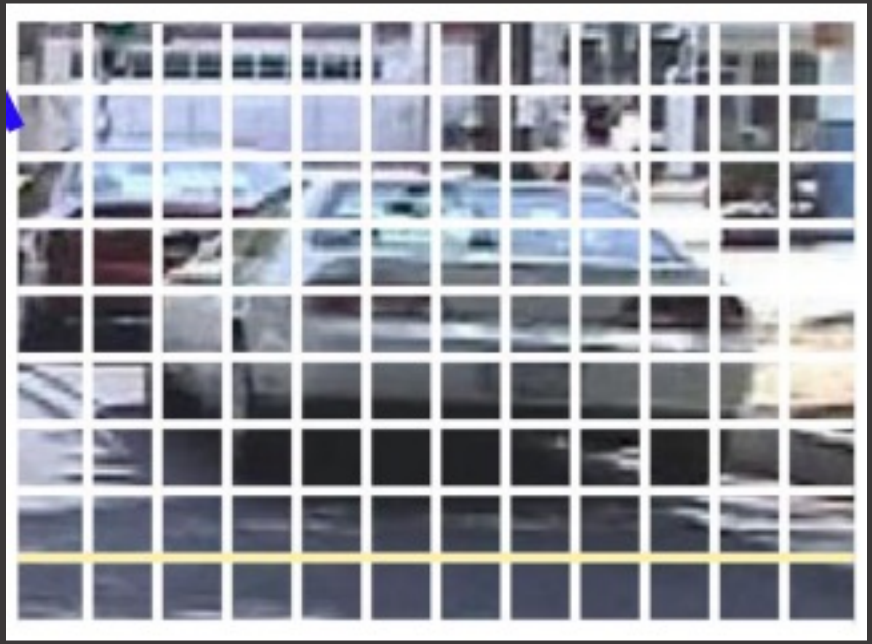


Computational Complexity

- In "Big O" analysis, we only consider the most dominant term, as the other terms and constants become insignificant asymptotically, e.g.,

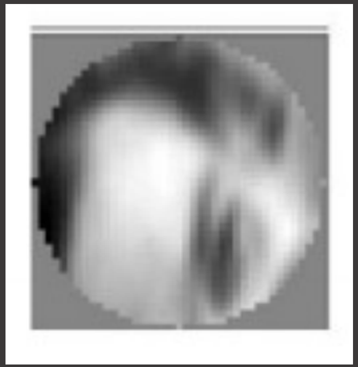
$$f(x) = 14x^2 - 10x + 25 \Longrightarrow O(x^2)$$
$$f(x) = (2x + 8)\log_2(x^2 + 9) \Longrightarrow O(x \log x)$$

Bag of Words: Visual Recognition

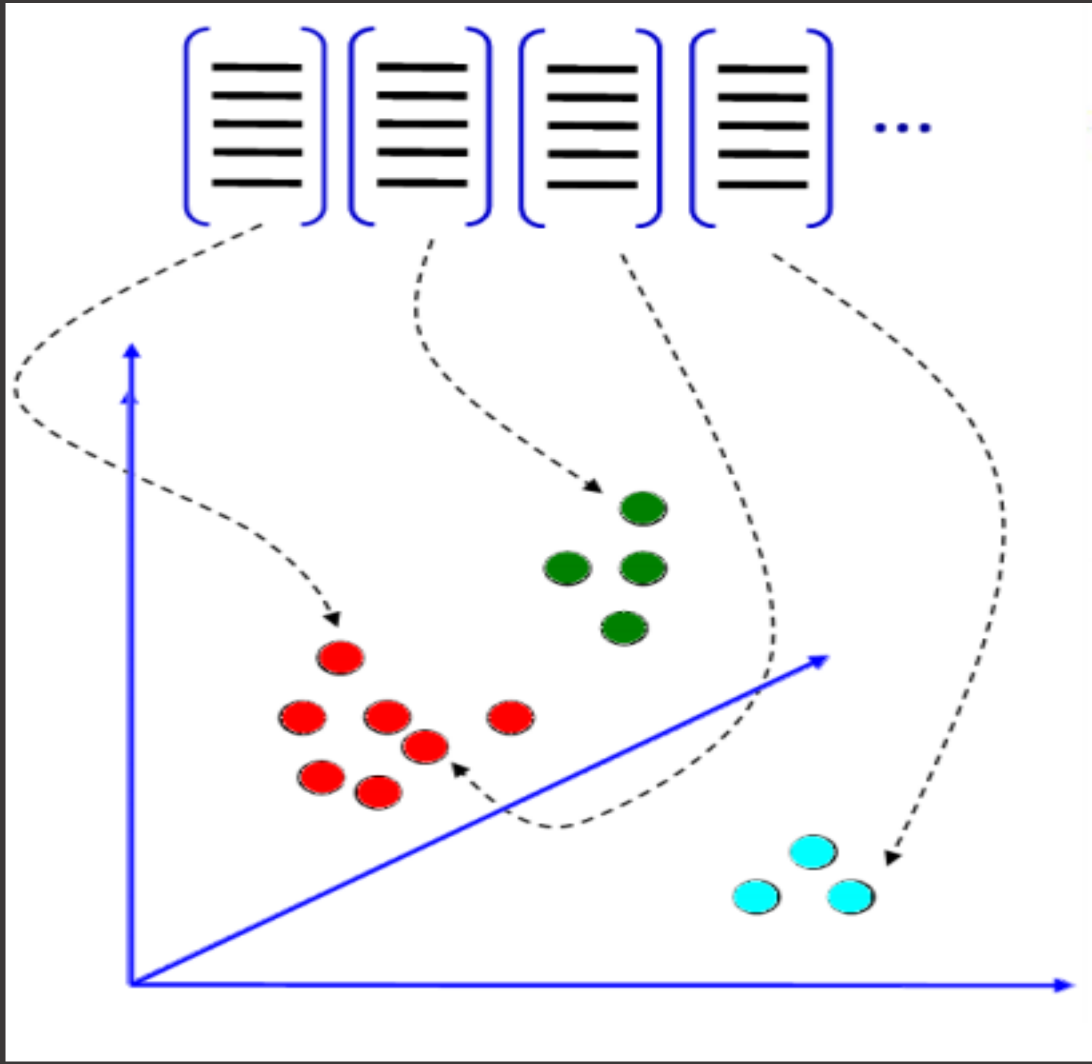


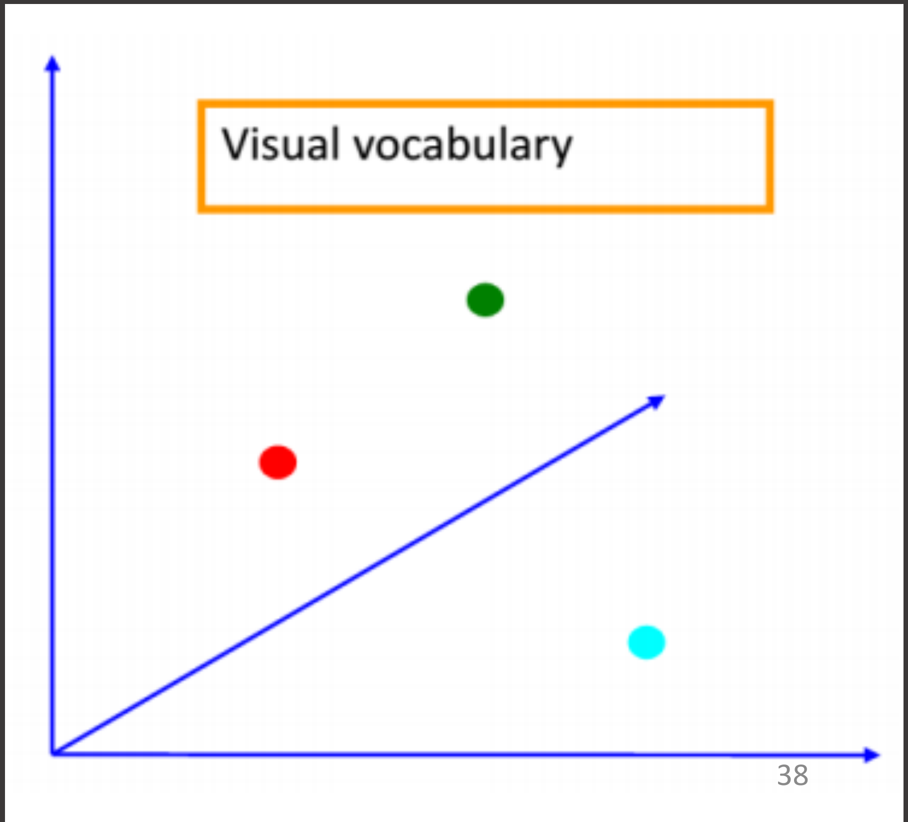
Compute SIFT descriptor [Lowe' 99]

Normalize Patch

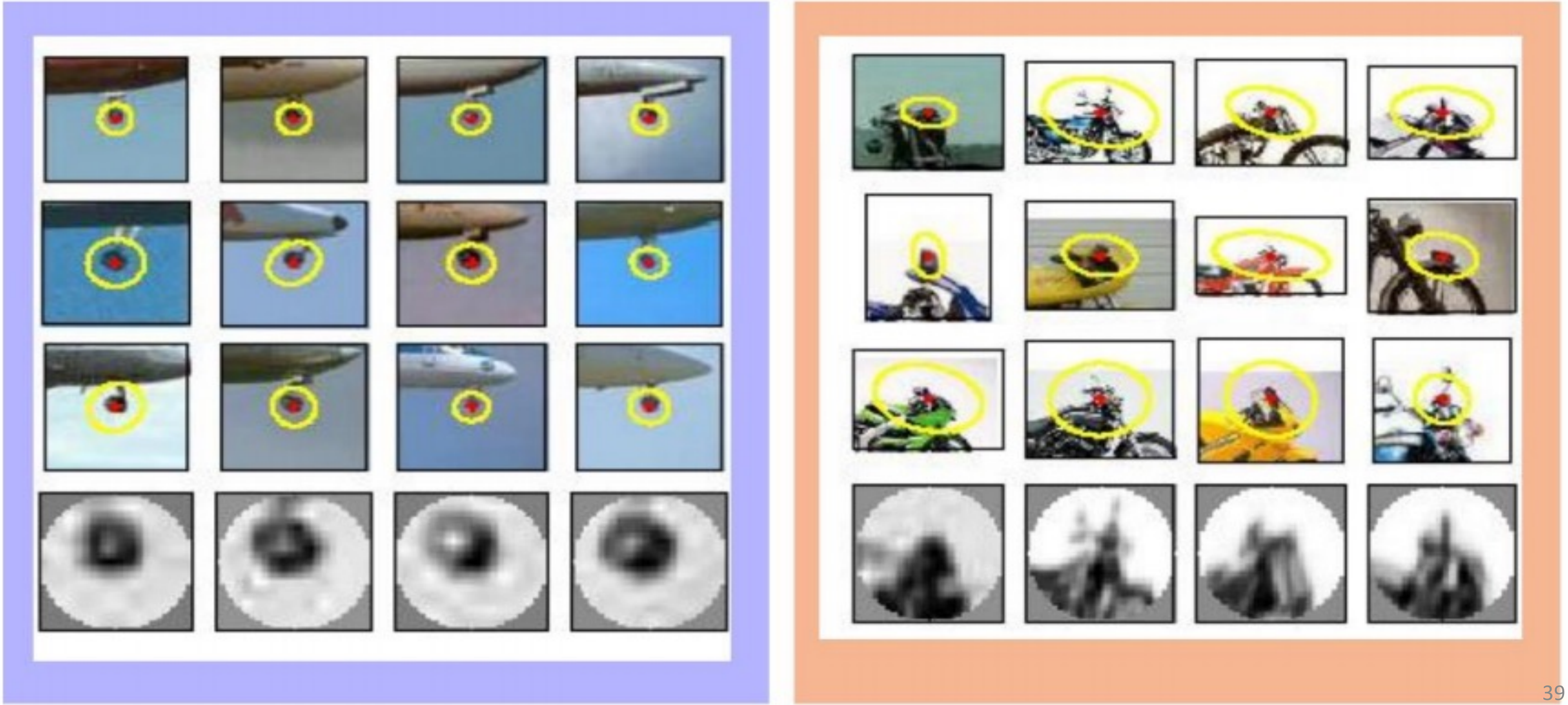


Bag of Words: Visual Recognition



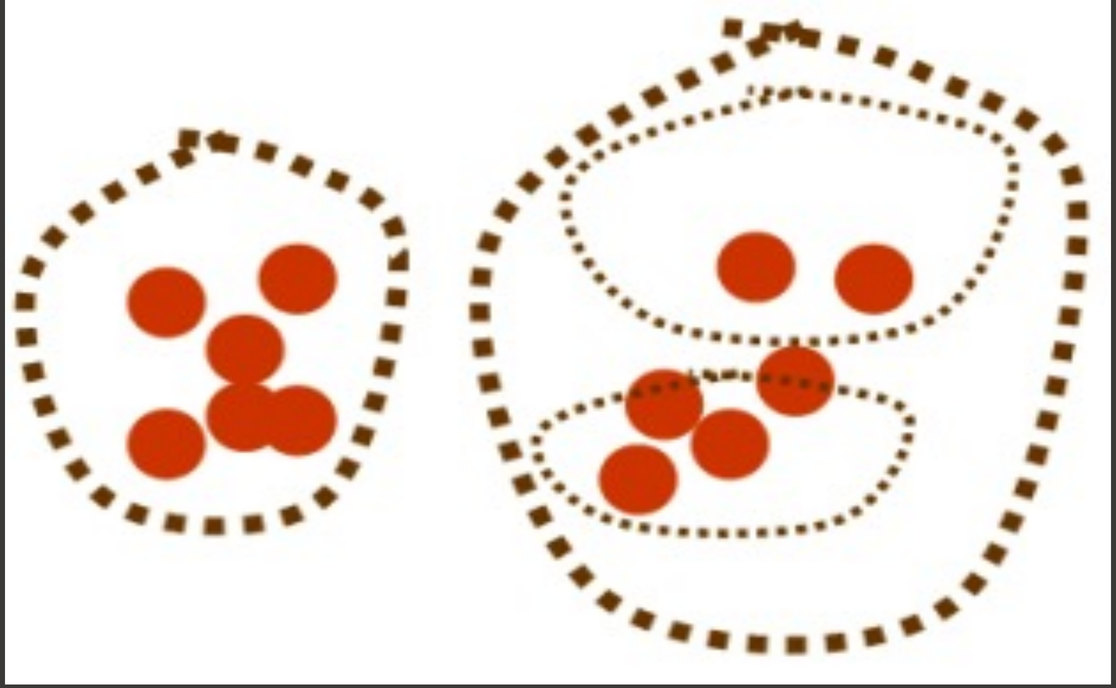


Examples of Words

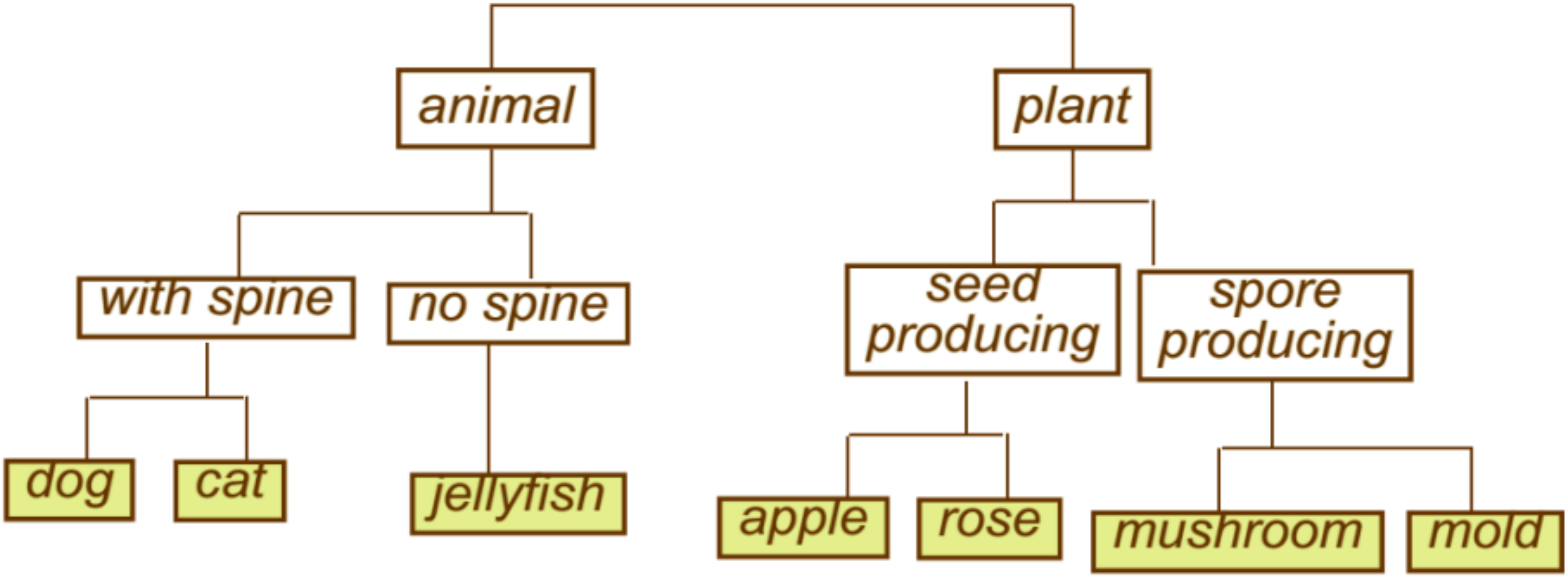


Hierarchical Clustering

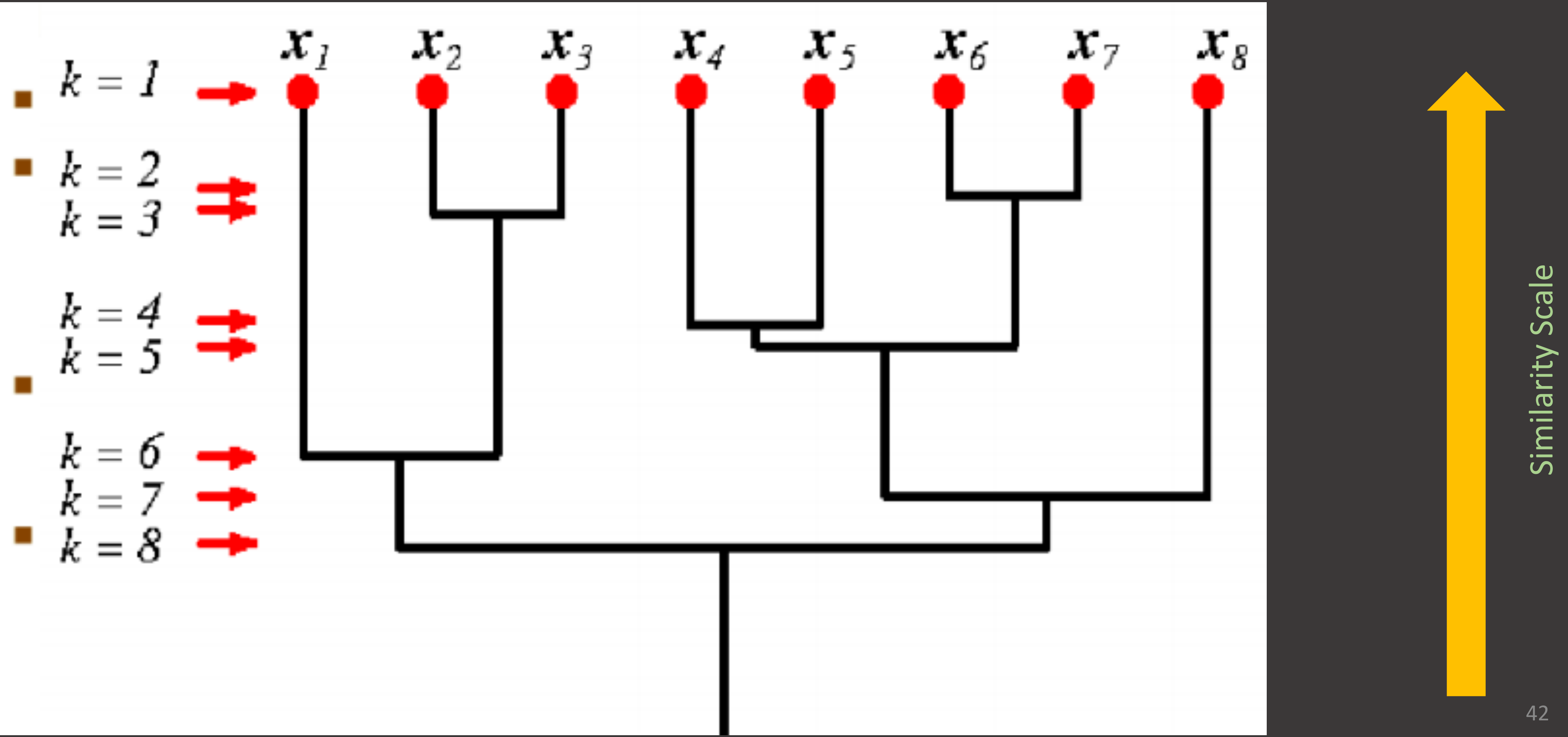
“Flat” Clustering vs. Hierarchical Clustering



Biological Taxonomy



Dendrogram Representation

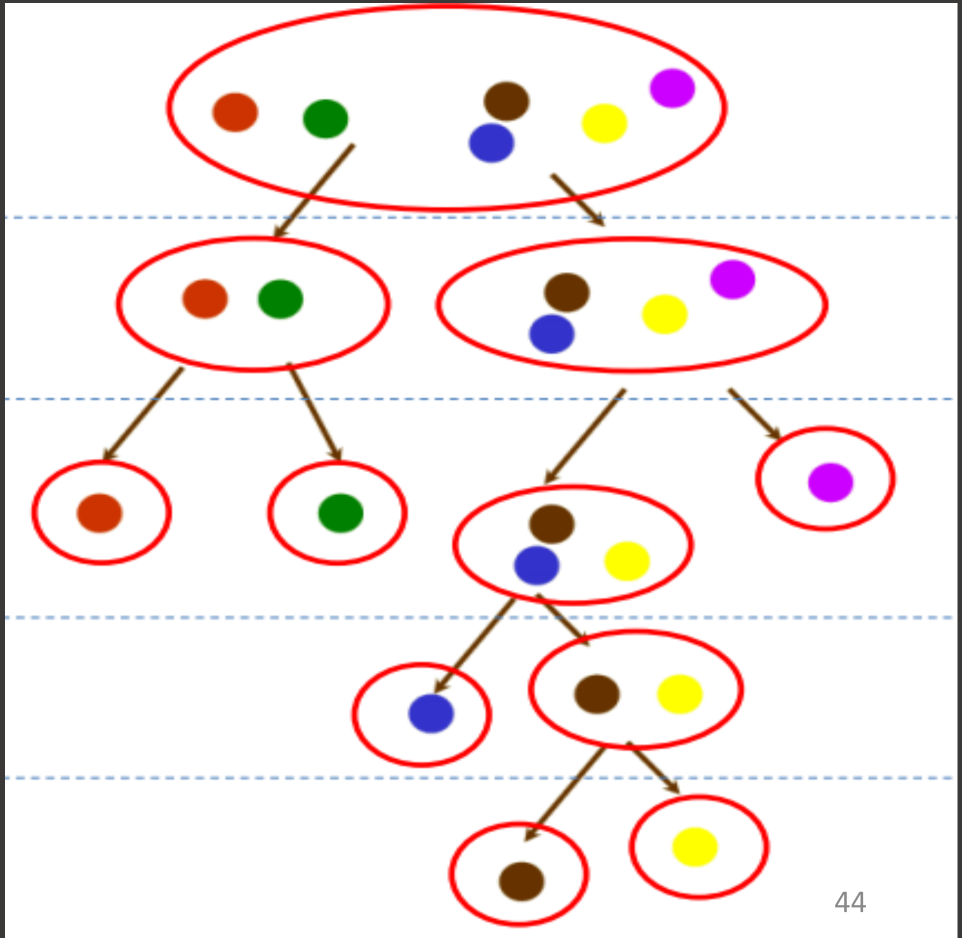


Types of Hierarchical Clustering

- **Divisive** (Top-Down) Clustering
  - Starts with all data points in one cluster, the **root**, then
    - ❑ Splits the root into a set of child clusters, and 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** (bottom up) clustering
  - The dendrogram is built from the bottom level by
    - ❑ Merge the most similar (or nearest) pair of clusters
    - ❑ Stop when all the data points are merged into a single cluster (i.e., the root cluster)

Divisive hierarchical clustering

- Any "flat" algorithm which produces a fixed number of clusters can be used

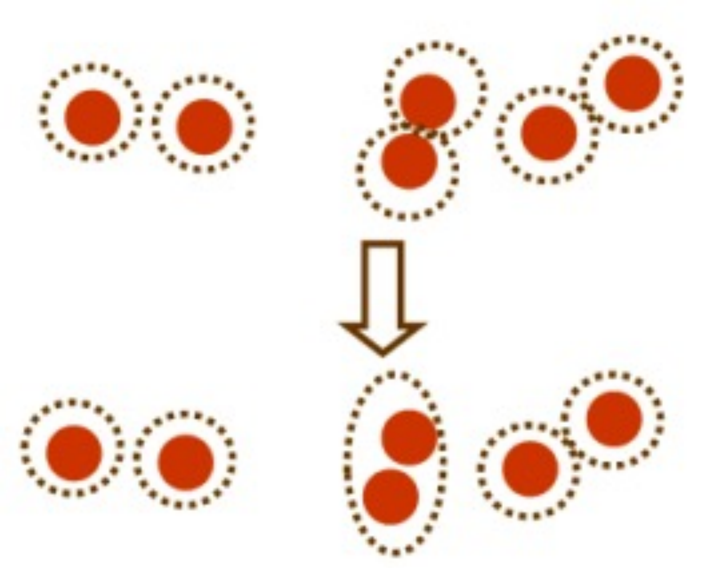


Agglomerative hierarchical clustering

Initialize with each example in singleton cluster.
While there is more than 1 cluster:

1. Find 2 nearest clusters
2. Merge them into 1

Question : How can you measure cluster distance?



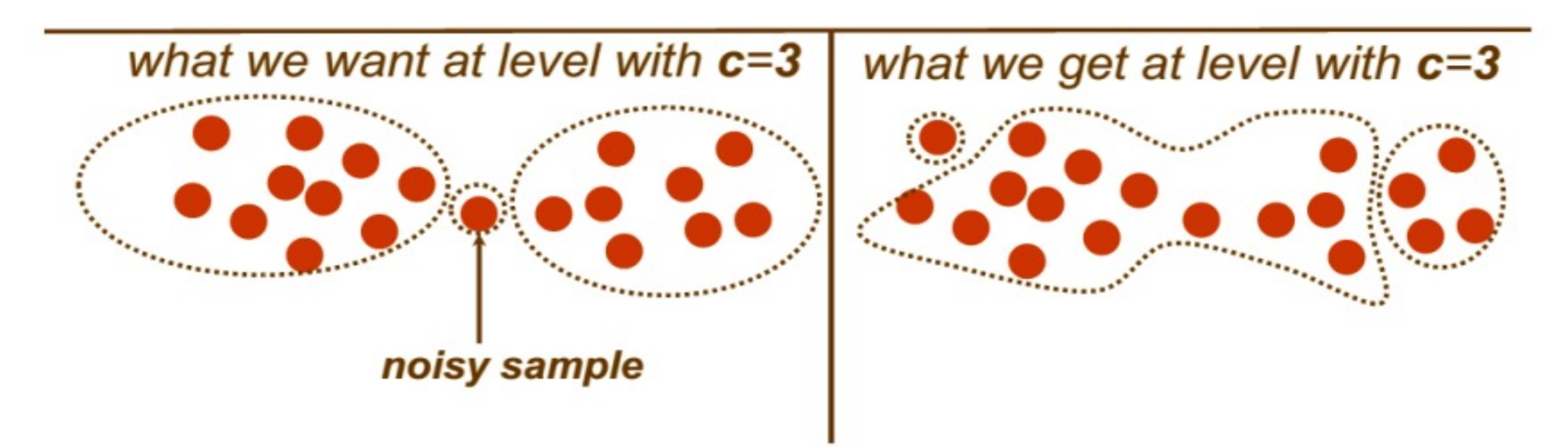
1. minimum distance $d_{min}(D_i, D_j) = \min_{x \in D_i, y \in D_j} \| x - y \|$
2. maximum distance $d_{max}(D_i, D_j) = \max_{x \in D_i, y \in D_j} \| x - y \|$
3. average distance $d_{avg}(D_i, D_j) = \frac{1}{n_i n_j} \sum_{x \in D_i} \sum_{y \in D_j} \| x - y \|$
4. mean distance $d_{mean}(D_i, D_j) = \| \mu_i - \mu_j \|$

Single Linkage or Nearest Neighbor

- Agglomerative clustering with minimum distance

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

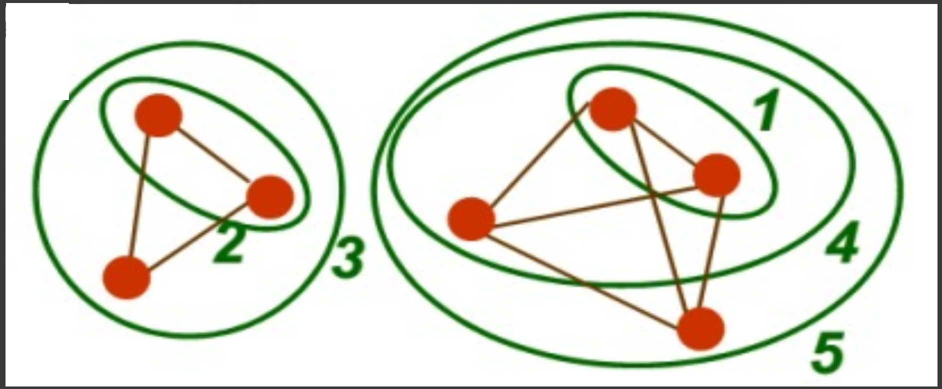
  - Encourages growth of elongated clusters
  - Disadvantage: very sensitive to noise



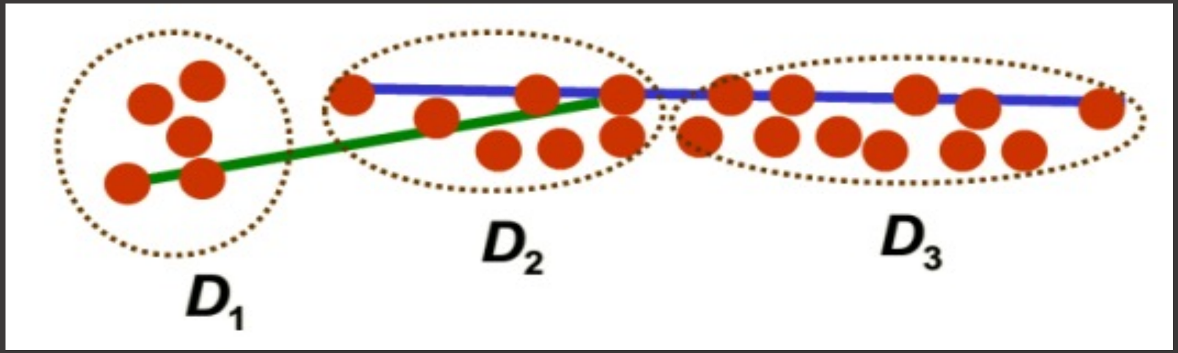
Complete linkage or Farthest neighbor

- Agglomerative clustering with maximum distance

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



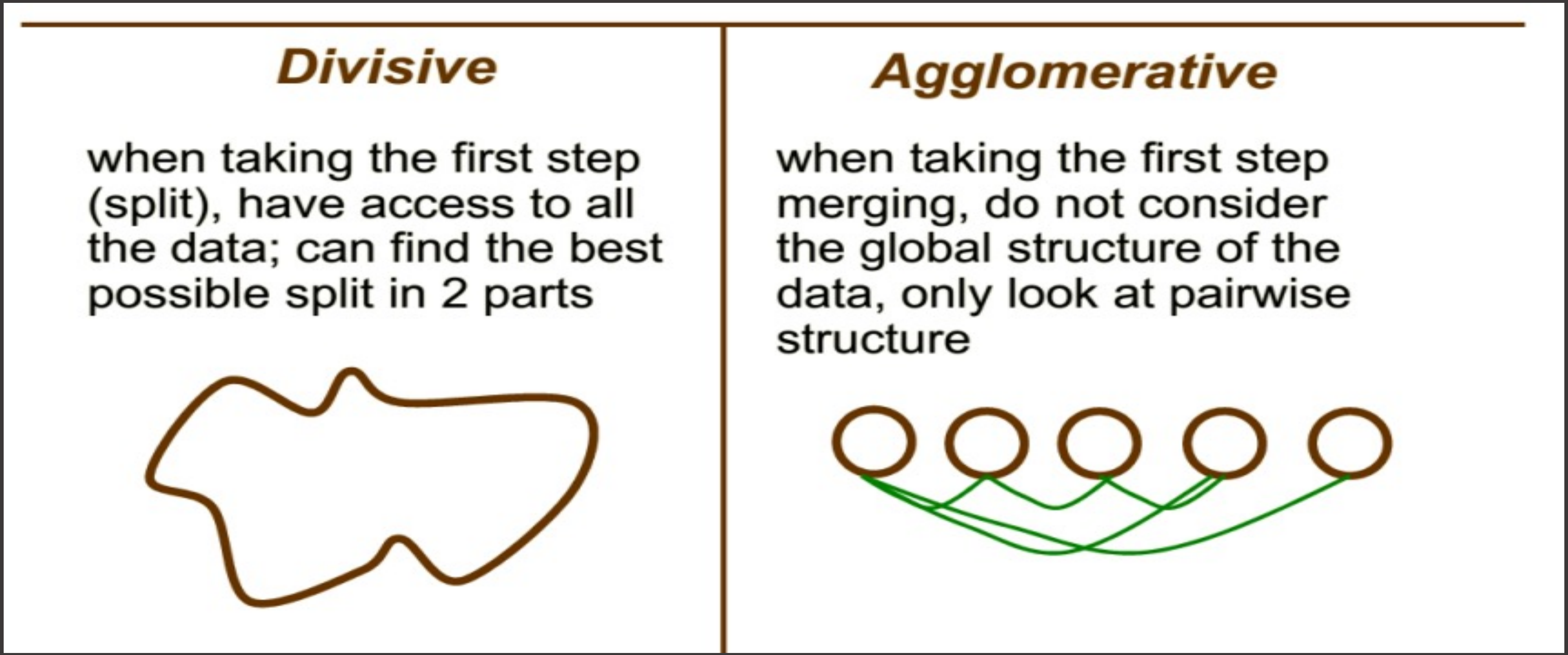
  - Encourages compact clusters
  - Does not work well if elongated clusters present



- $d_{\max}(D_1, D_2) < d_{\max}(D_2, D_3)$
- thus D_1 and D_2 are merged instead of D_2 and D_3

Divisive vs. Agglomerative

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



Summary

- Clustering has a long history and still is in active research
  - There are a huge number of clustering algorithms, among them: Density based algorithm, Sub-space clustering, Scale-up methods, Neural networks based methods, Fuzzy clustering, Co-clustering …
  - More are still coming every year
- Clustering is hard to evaluate, but very useful in practice
- Clustering is highly application dependent (and to some extent subjective)
- Competitive learning in neuronal networks performs clustering analysis of the input data