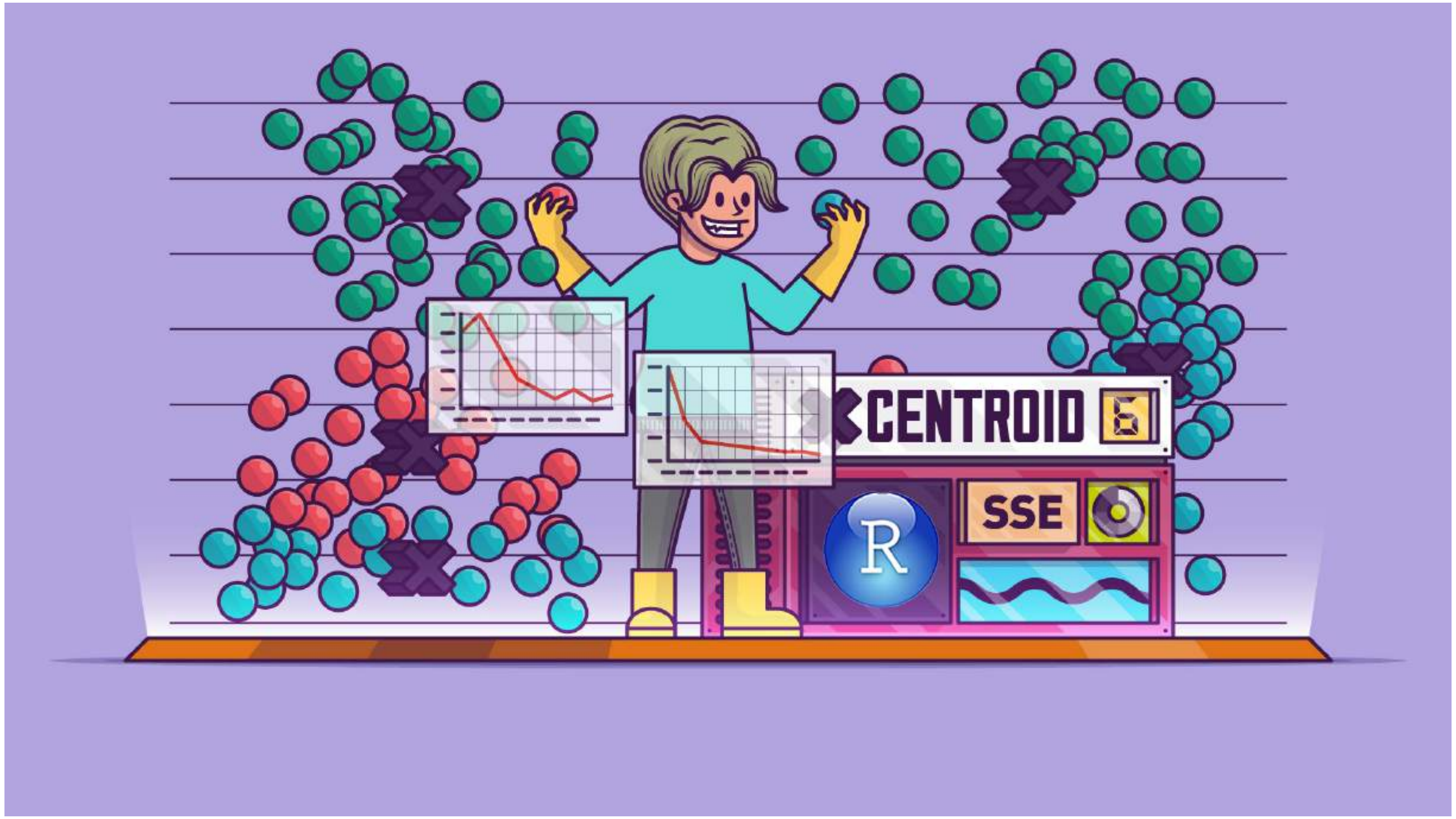


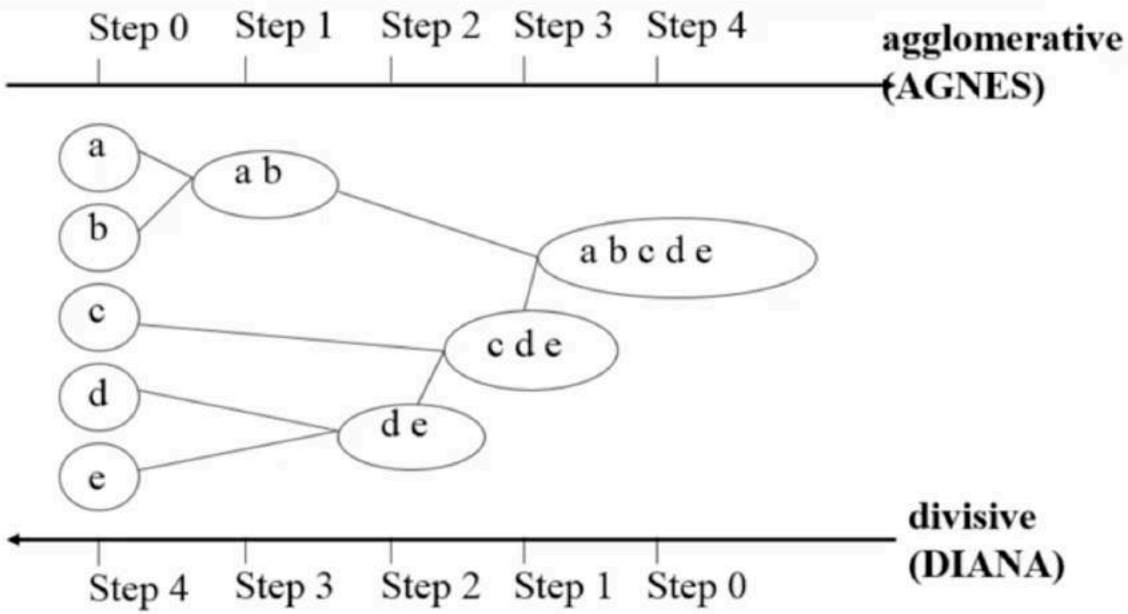
Clustering Analysis

Day 2 : February 26, 2021

Hierarchical Clustering Methods

Hierarchical Clustering

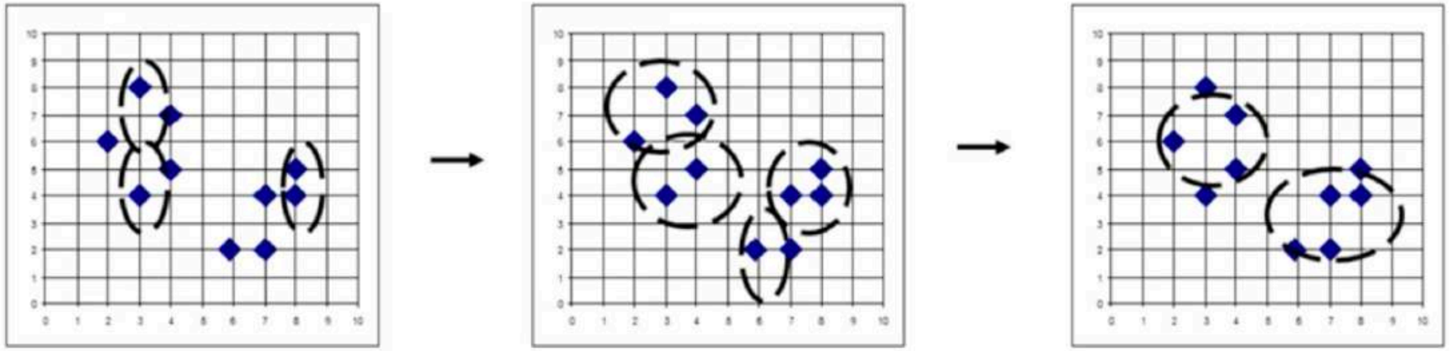
- Hierarchical clustering
  - Generate a clustering hierarchy (drawn as a **dendrogram**)
  - Not required to specify ***K*,** the number of clusters
  - More deterministic
  - No iterative refinement
- Two categories of algorithms:



  - **Agglomerative**: Start with singleton clusters, continuously merge two clusters at a time to build a **bottom-up** hierarchy of clusters
  - **Divisive:** Start with a huge macro-cluster, split it continuously into two groups, generating a **top-down** hierarchy of clusters

Agglomerative Algorithm

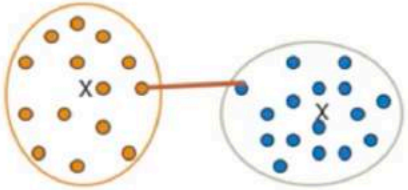
- AGNES (AGglomerative NESting) (Kaufmann and Rousseeuw, 1990)
  - Use the **single-link** method and the dissimilarity matrix
  - Continuously merge nodes that have the least dissimilarity
  - Eventually all nodes belong to the same cluster



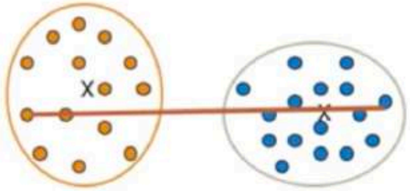
- Agglomerative clustering varies on different similarity measures among clusters
  - Single link (nearest neighbor)
  - Complete link (diameter)
  - Average link (group average)
  - Centroid link (centroid similarity)

Single vs. Complete Link

- Single link (nearest neighbor)
  - The similarity between two clusters is the similarity between their most similar (nearest neighbor) members
  - Local similarity-based: Emphasizing more on close regions, ignoring the overall structure of the cluster
  - Capable of clustering non-elliptical shaped group of objects
  - Sensitive to noise and outliers

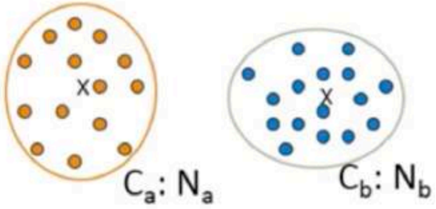


- Complete link (diameter)
  - The similarity between two clusters is the similarity between their most dissimilar members
  - Merge two clusters to form one with the smallest diameter
  - Nonlocal in behavior, obtaining compact shaped clusters
  - Sensitive to outliers

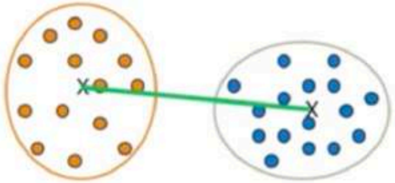


Average vs. Centroid Links

- Agglomerative clustering with **average link**
  - **Average link**: The average distance between an element in one cluster and an element in the other (i.e., all pairs in two clusters)
    - Expensive to compute



- Agglomerative clustering with **centroid link**
  - **Centroid link**: The distance between the centroids of two clusters



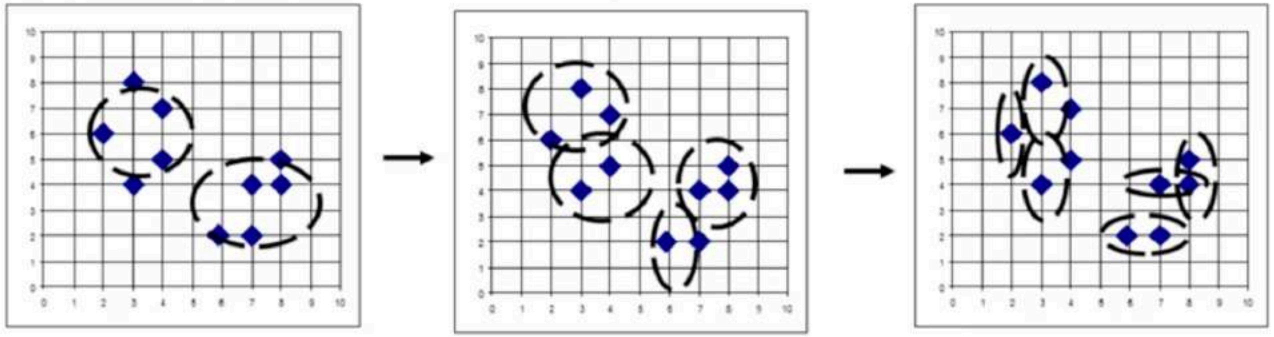
- **Group Averaged Agglomerative Clustering (GAAC)**
  - Let two clusters C_a and C_b be merged into C_{aUb}. The new centroid is:
    - N_a is the cardinality of cluster C_a, and c_a is the centroid of C_a
  - The similarity measure for GAAC is the average of their distances

$$c_{a\cup b} = \frac{N_a c_a + N_b c_b}{N_a + N_b}$$

- Agglomerative clustering with **Ward's criterion**
  - **Ward's criterion:** The increase in the value of the SSE criterion for the clustering obtained by merging them into C_a U C_b: $W(C_{a\cup b}, c_{a\cup b}) - W(C, c) = \frac{N_a N_b}{N_a + N_b} d(c_a, c_b)$

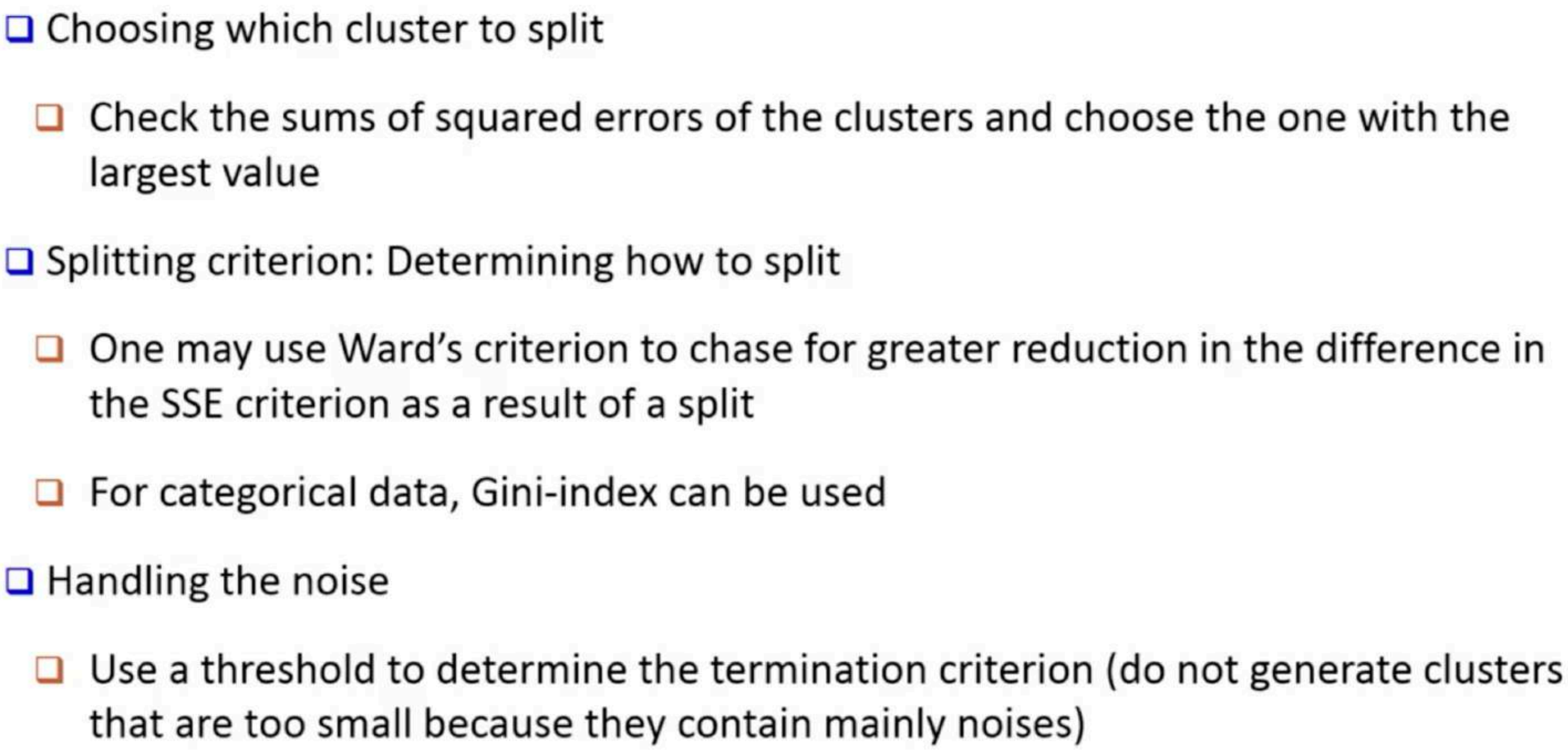
Divisive Algorithm

- DIANA (Divisive Analysis) (Kaufmann and Rousseeuw,1990)
  - Implemented in some statistical analysis packages, e.g., Splus
- Inverse order of AGNES: Eventually each node forms a cluster on its own



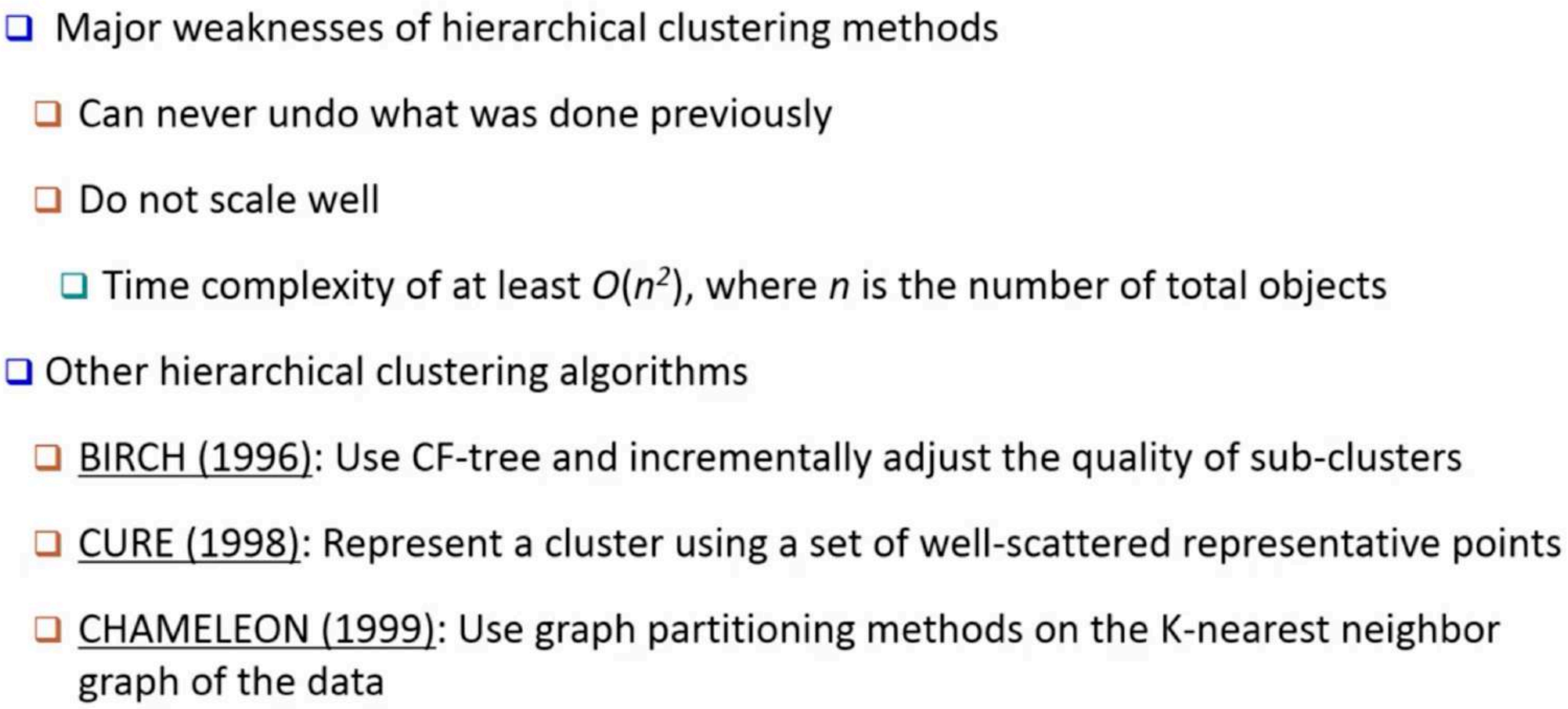
- Divisive clustering is a top-down approach
  - The process starts at the root with all the points as one cluster
  - It recursively splits the higher level clusters to build the dendrogram
  - Can be considered as a global approach
  - More efficient when compared with agglomerative clustering

Divisive Algorithm Discussion



- Choosing which cluster to split
  - Check the sums of squared errors of the clusters and choose the one with the largest value
- Splitting criterion: Determining how to split
  - One may use Ward’s criterion to chase for greater reduction in the difference in the SSE criterion as a result of a split
  - For categorical data, Gini-index can be used
- Handling the noise
  - Use a threshold to determine the termination criterion (do not generate clusters that are too small because they contain mainly noises)

Hierarchical Clustering Extensions



- Major weaknesses of hierarchical clustering methods
  - Can never undo what was done previously
  - Do not scale well
    - Time complexity of at least $O(n^2)$, where n is the number of total objects
- Other hierarchical clustering algorithms
  - BIRCH (1996): Use CF-tree and incrementally adjust the quality of sub-clusters
  - CURE (1998): Represent a cluster using a set of well-scattered representative points
  - CHAMELEON (1999): Use graph partitioning methods on the K-nearest neighbor graph of the data

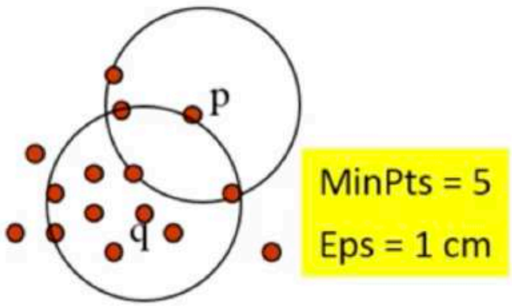
Density-Based and Grid-Based Clustering Methods

Density-Based Clustering

- Clustering based on density (a local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan (only examine the local region to justify density)
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96) To be covered in this lecture
  - OPTICS: Ankerst, et al (SIGMOD'99) To be covered in this lecture
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (also, grid-based) To be covered in this lecture

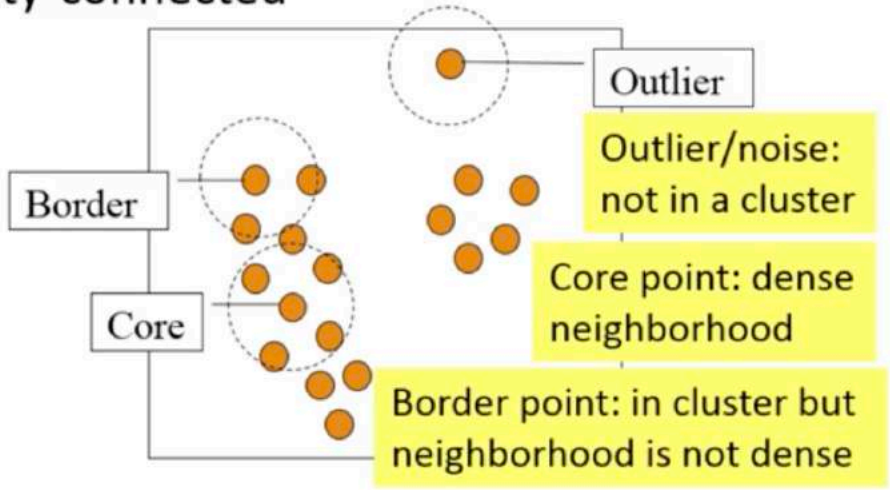
DBSCAN Clustering

- DBSCAN (M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, KDD'96)
  - Discovers clusters of arbitrary shape: Density-Based Spatial Clustering of Applications with Noise
- A *density-based* notion of cluster
  - A *cluster* is defined as a maximal set of density-connected points
- Two parameters:
  - *Eps*: Maximum radius of the neighborhood
  - *MinPts*: Minimum number of points in the Eps-neighborhood of a point
- The Eps-neighborhood of a point *q*:
  - $N_{Eps}(q)$: {p belongs to D | dist(p, q) ≤ Eps}



MinPts = 5

Eps = 1 cm

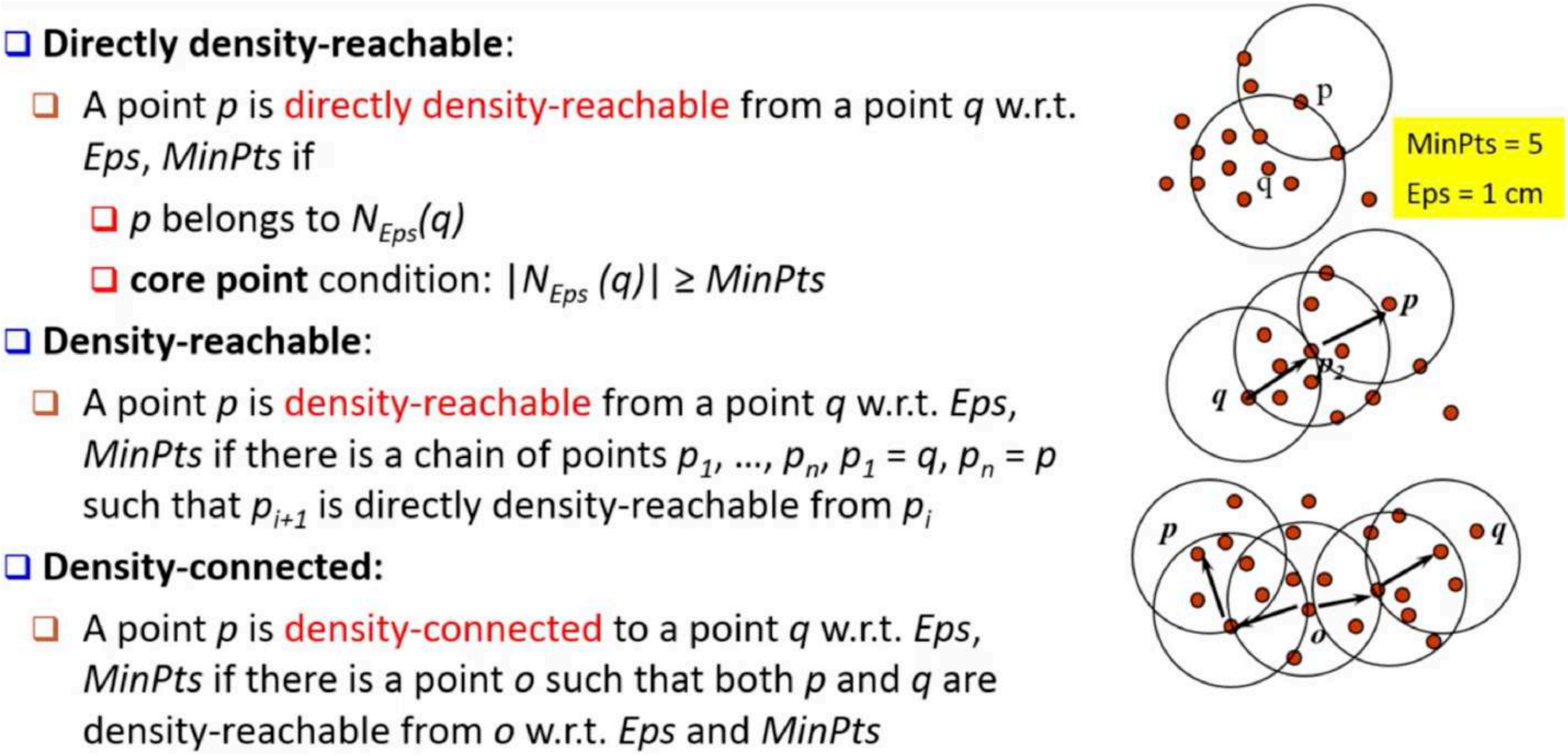


Outlier/noise: not in a cluster

Core point: dense neighborhood

Border point: in cluster but neighborhood is not dense

Density-Reachable & Density-Connected



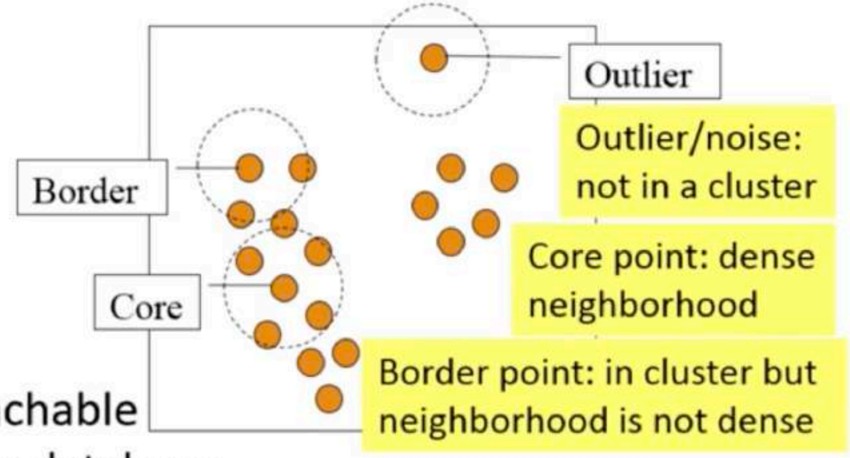
- ❑ **Directly density-reachable**:
  - ❑ A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if
    - ❑ *p* belongs to $N_{Eps}(q)$
    - ❑ **core point** condition: $|N_{Eps}(q)| \geq MinPts$
- ❑ **Density-reachable**:
  - ❑ A point *p* is density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i
- ❑ **Density-connected:**
  - ❑ A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *o* such that both *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*

MinPts = 5

Eps = 1 cm

DBSCAN: The Algorithm

- **Algorithm**
  - Arbitrarily select a point *p*
  - Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*
    - If *p* is a core point, a cluster is formed
    - If *p* is a border point, no points are density-reachable from *p,* and DBSCAN visits the next point of the database
  - Continue the process until all of the points have been processed
- **Computational complexity**
  - If a spatial index is used, the computational complexity of DBSCAN is O(nlogn), where n is the number of database objects
  - Otherwise, the complexity is O(n^2)



Outlier/noise: not in a cluster

Core point: dense neighborhood

Border point: in cluster but neighborhood is not dense

DBSCAN is Sensitive to the Parameter Setting

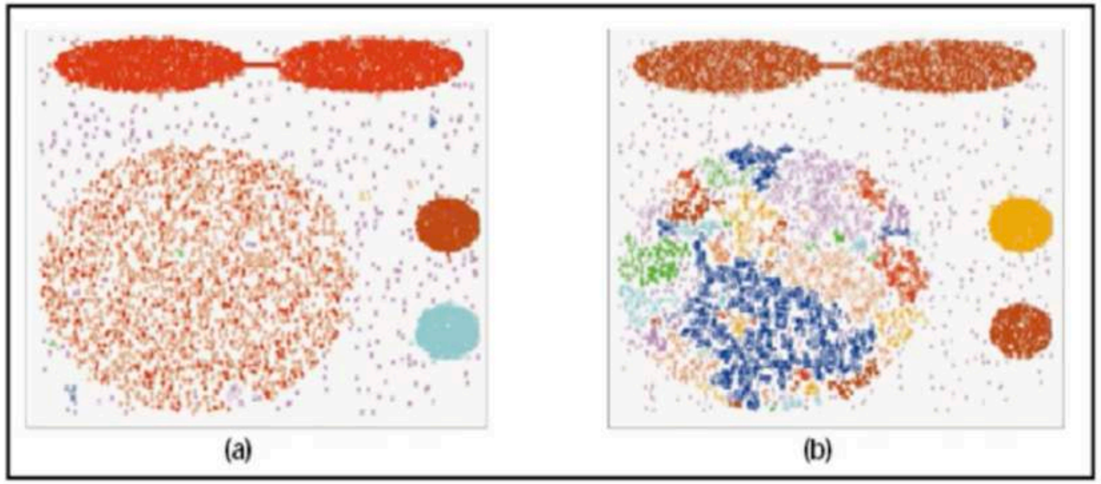


Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

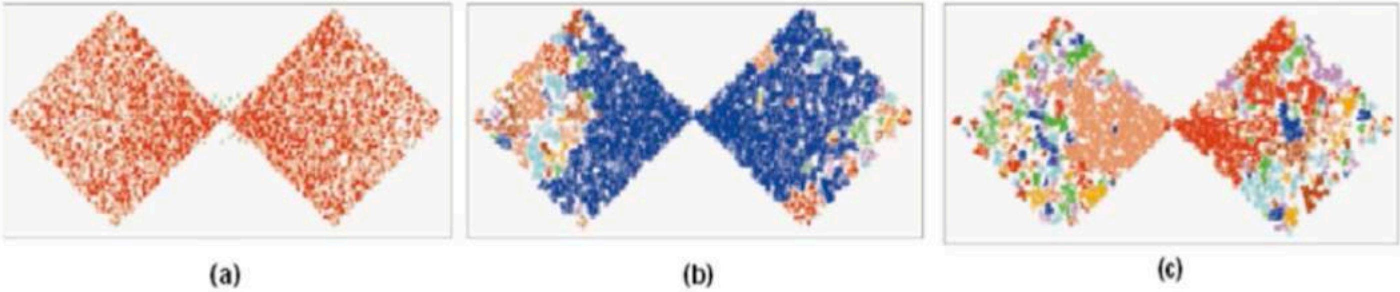
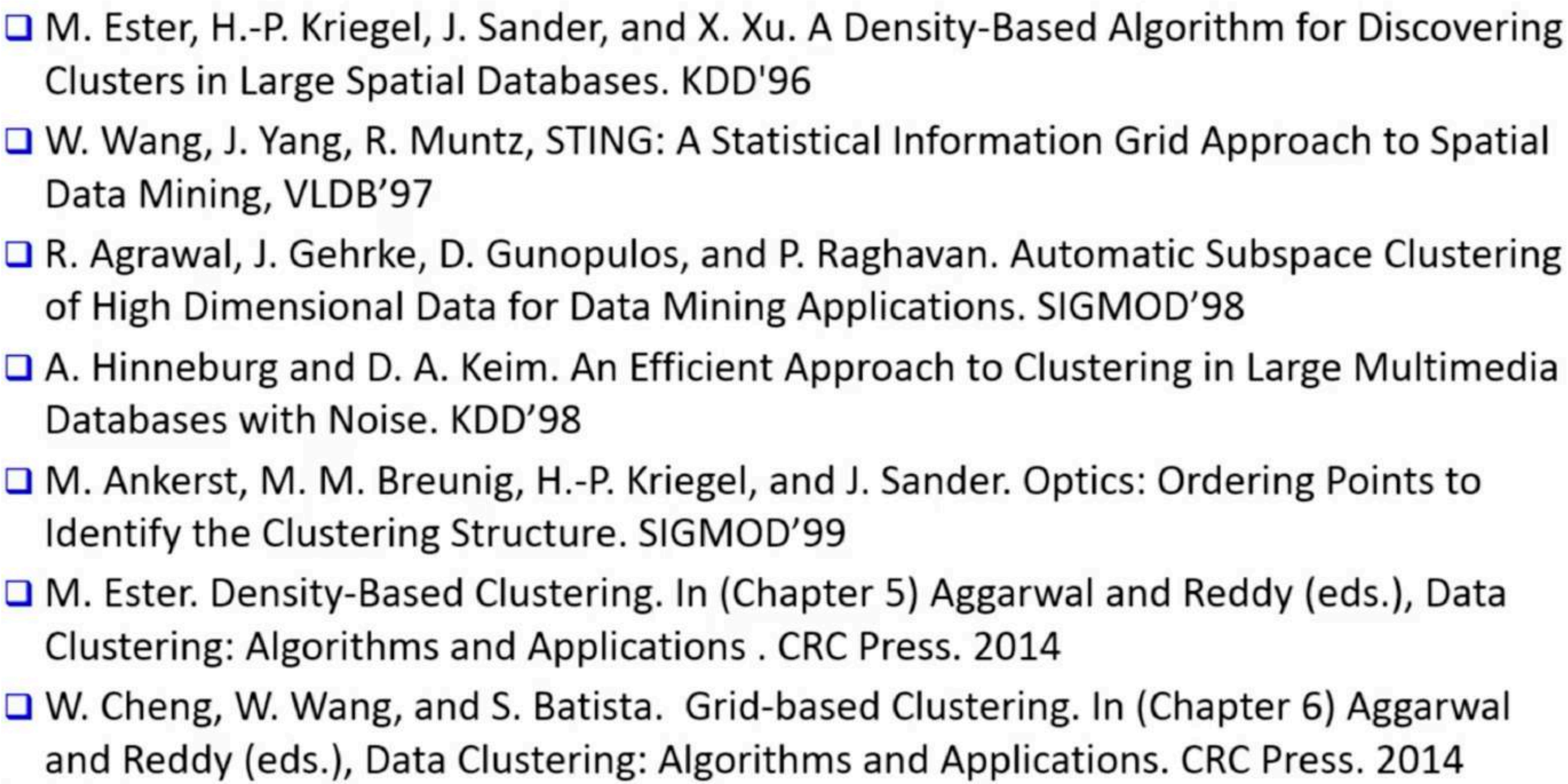


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.

Ack. Figures from G. Karypis, E.-H. Han, and V. Kumar, *COMPUTER*, 32(8), 1999

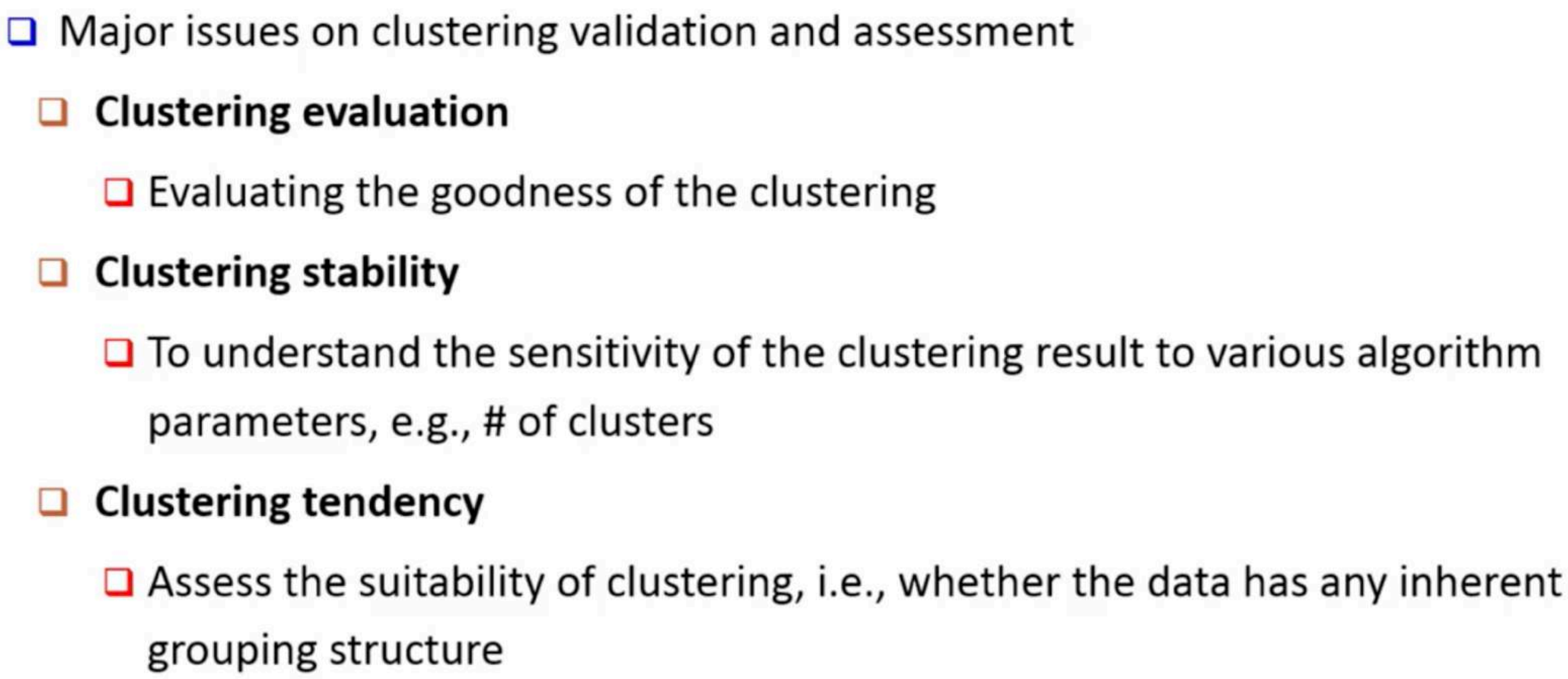
Recommended Readings



- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases. KDD'96
- W. Wang, J. Yang, R. Muntz, STING: A Statistical Information Grid Approach to Spatial Data Mining, VLDB'97
- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. SIGMOD'98
- A. Hinneburg and D. A. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. KDD'98
- M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering Points to Identify the Clustering Structure. SIGMOD'99
- M. Ester. Density-Based Clustering. In (Chapter 5) Aggarwal and Reddy (eds.), Data Clustering: Algorithms and Applications . CRC Press. 2014
- W. Cheng, W. Wang, and S. Batista. Grid-based Clustering. In (Chapter 6) Aggarwal and Reddy (eds.), Data Clustering: Algorithms and Applications. CRC Press. 2014

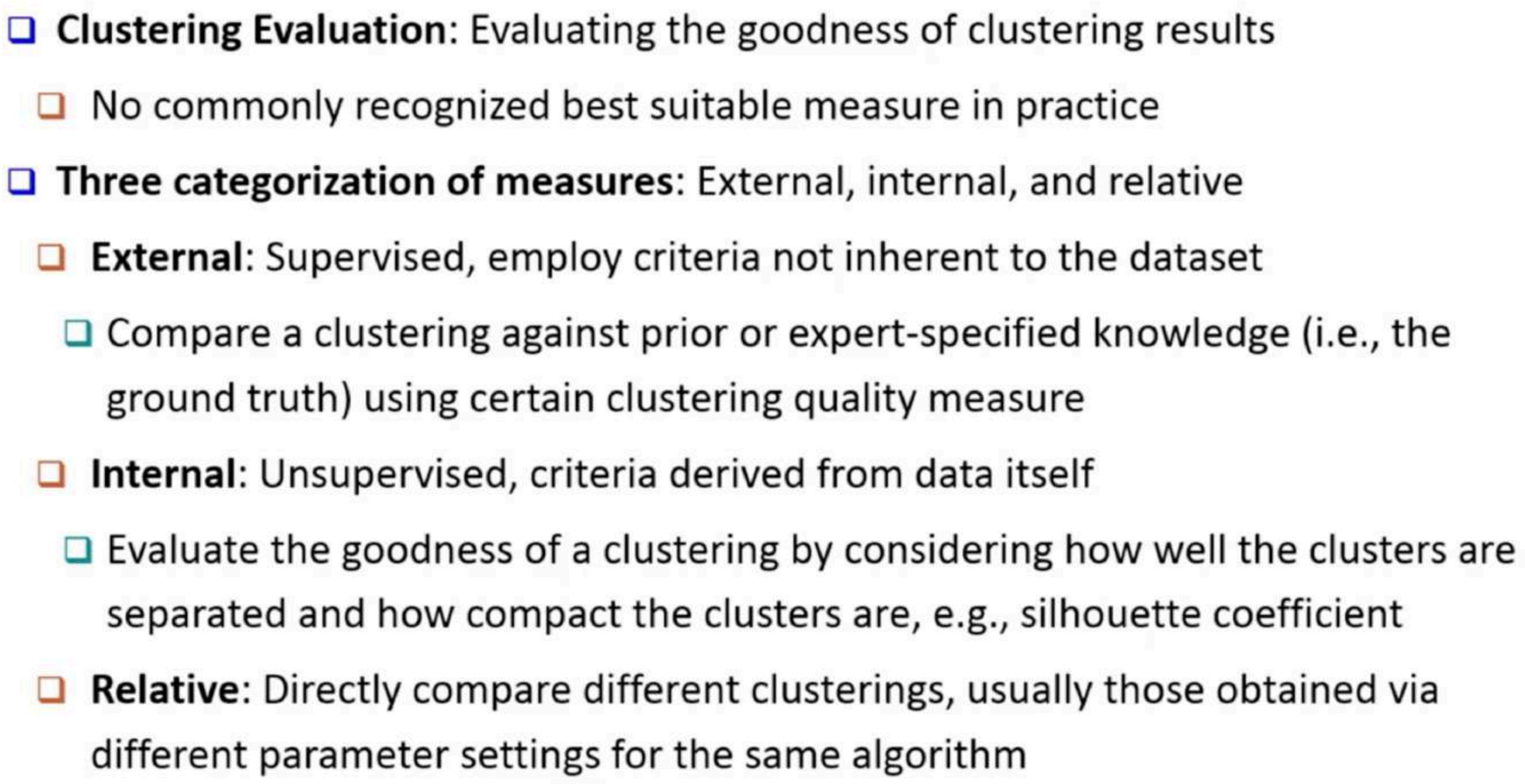
Clustering Validation Methods

Clustering Validation



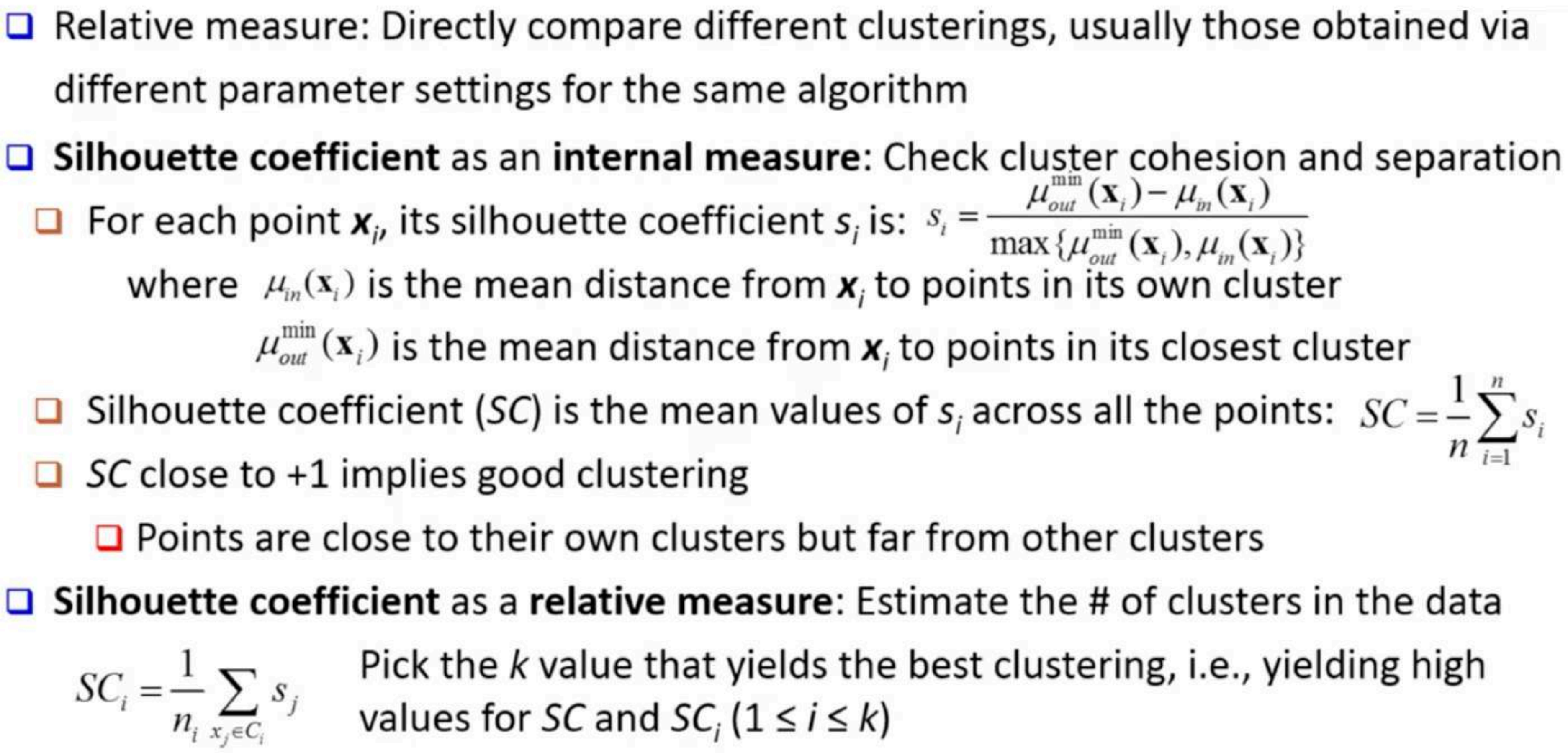
- ❑ Major issues on clustering validation and assessment
  - ❑ **Clustering evaluation**
    - ❑ Evaluating the goodness of the clustering
  - ❑ **Clustering stability**
    - ❑ To understand the sensitivity of the clustering result to various algorithm parameters, e.g., # of clusters
  - ❑ **Clustering tendency**
    - ❑ Assess the suitability of clustering, i.e., whether the data has any inherent grouping structure

Measuring Clustering Quality



- **Clustering Evaluation**: Evaluating the goodness of clustering results
  - No commonly recognized best suitable measure in practice
- **Three categorization of measures**: External, internal, and relative
  - **External**: Supervised, employ criteria not inherent to the dataset
    - Compare a clustering against prior or expert-specified knowledge (i.e., the ground truth) using certain clustering quality measure
  - **Internal**: Unsupervised, criteria derived from data itself
    - Evaluate the goodness of a clustering by considering how well the clusters are separated and how compact the clusters are, e.g., silhouette coefficient
  - **Relative**: Directly compare different clusterings, usually those obtained via different parameter settings for the same algorithm

Relative Measure



- Relative measure: Directly compare different clusterings, usually those obtained via different parameter settings for the same algorithm
- **Silhouette coefficient** as an **internal measure**: Check cluster cohesion and separation
  - For each point ***x**$_i$*, its silhouette coefficient s_i is: $s_i = \dfrac{\mu_{out}^{\min}(\mathbf{x}_i) - \mu_{in}(\mathbf{x}_i)}{\max\{\mu_{out}^{\min}(\mathbf{x}_i), \mu_{in}(\mathbf{x}_i)\}}$

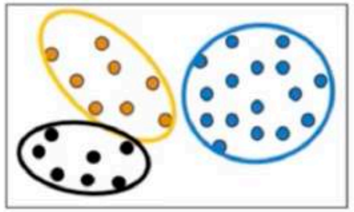
    where $\mu_{in}(\mathbf{x}_i)$ is the mean distance from ***x**$_i$* to points in its own cluster

    $\mu_{out}^{\min}(\mathbf{x}_i)$ is the mean distance from ***x**$_i$* to points in its closest cluster
  - Silhouette coefficient (*SC*) is the mean values of s_i across all the points: $SC = \dfrac{1}{n}\sum_{i=1}^{n} s_i$
  - *SC* close to +1 implies good clustering
    - Points are close to their own clusters but far from other clusters
- **Silhouette coefficient** as a **relative measure**: Estimate the # of clusters in the data

  $SC_i = \dfrac{1}{n_i}\sum_{x_j \in C_i} s_j$ Pick the *k* value that yields the best clustering, i.e., yielding high values for *SC* and SC_i ($1 \le i \le k$)

Cluster Stability

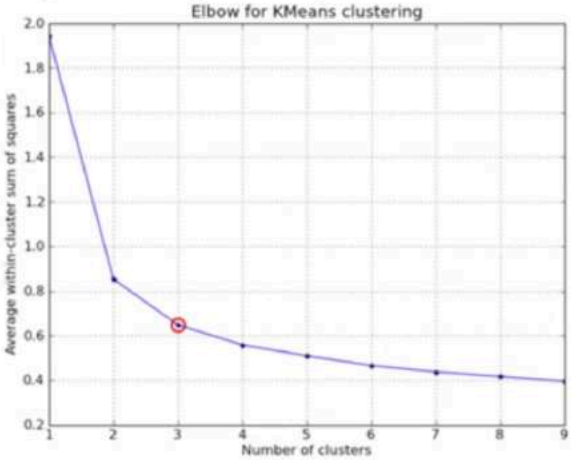
- ❑ Clusterings obtained from several datasets sampled from the same underlying distribution as ***D*** should be similar or “stable”



- ❑ Typical approach:
  - ❑ Find good parameter values for a given clustering algorithm
- ❑ Example: Find a good value of *k*, the correct number of clusters
- ❑ A **bootstrapping approach** to find the best value of *k* (judged on stability)
  - ❑ Generate *t* samples of size *n* by sampling from ***D*** with replacement
  - ❑ For each sample $\boldsymbol{D}_i$, run the same clustering algorithm with *k* values from 2 to k_{max}
  - ❑ Compare the distance between all pairs of clusterings $C_k(\boldsymbol{D}_i)$ and $C_k(\boldsymbol{D}_j)$ via some distance function
    - ❑ Compute the expected pairwise distance for each value of *k*
  - ❑ The value *k** that exhibits the least deviation between the clusterings obtained from the resampled datasets is the best choice for *k* since it exhibits the most stability

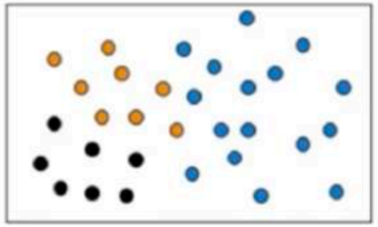
Number of Cluster

- **Empirical method**
  - # of clusters: $k \approx \sqrt{n/2}$ for a dataset of n points (e.g., n = 200, k = 10)
- **Elbow method**: Use the turning point in the curve of the sum of within cluster variance with respect to the # of clusters



- **Cross validation method**
  - Divide a given data set into m parts
  - Use $m - 1$ parts to obtain a clustering model
  - Use the remaining part to test the quality of the clustering
    - For example, for each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and the closest centroids to measure how well the model fits the test set
  - For any $k > 0$, repeat it m times, compare the overall quality measure w.r.t. different k's, and find # of clusters that fits the data the best

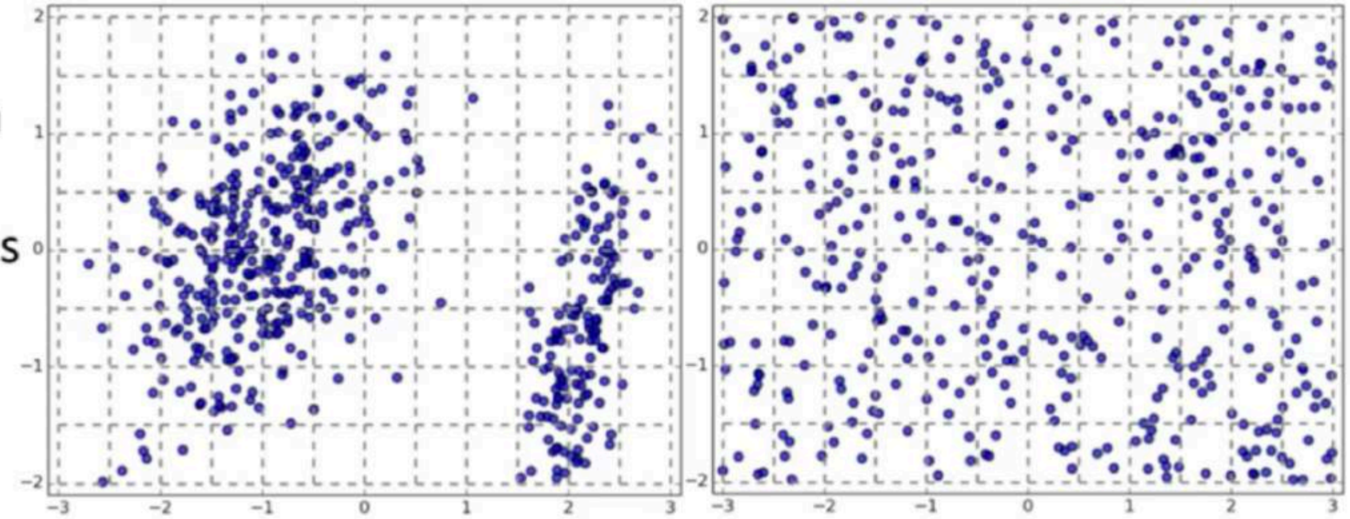
Clustering Tendency



- Assessing the **suitability of clustering**
  - (i.e., whether the data has any inherent grouping structure)
- Determining ***clustering tendency*** or ***clusterability***
  - **A hard task** because there are so many different definitions of clusters
    - E.g., partitioning, hierarchical, density-based, graph-based, etc.
  - Even fixing cluster type, still hard to define an appropriate null model for a data set
- Still, there are some **clusterability assessment methods**, such as
  - **Spatial histogram**: Contrast the histogram of the data with that generated from random samples
  - **Distance distribution**: Compare the pairwise point distance from the data with those from the randomly generated samples
  - **Hopkins Statistic**: A sparse sampling test for spatial randomness

Test Clustering Tendency

- **Spatial Histogram Approach:** Contrast the *d*-dimensional histogram of the input dataset ***D*** with the histogram generated from random samples
  - Dataset D is clusterable if the distributions of two histograms are rather different
- Method outline
  - Divide each dimension into equi-width bins, count how many points lie in each cells, and obtain the empirical joint probability mass function (EPMF)



  - Do the same for the randomly sampled data
  - Compute how much they differ using the *Kullback-Leibler* (*KL*) *divergence* value

Recommended Readings

- M. J. Zaki and W. Meira, Jr.. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press, 2014
- L. Hubert and P. Arabie. Comparing Partitions. *Journal of Classification*, 2:193–218, 1985
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988
- M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On Clustering Validation Techniques. *Journal of Intelligent Info. Systems*, 17(2-3):107–145, 2001
- J. Han, M. Kamber, and J. Pei. Data Mining: Concepts and Techniques. Morgan Kaufmann, 3rd ed. , 2011
- H. Xiong and Z. Li. Clustering Validation Measures. in (Chapter 23) C. Aggarwal and C. K. Reddy (eds.), Data Clustering: Algorithms and Applications. CRC Press, 2014