Knowledge Discovery and Data Mining

Spring 2021

Chap 10. Cluster Analysis: Basic Concepts and Methods

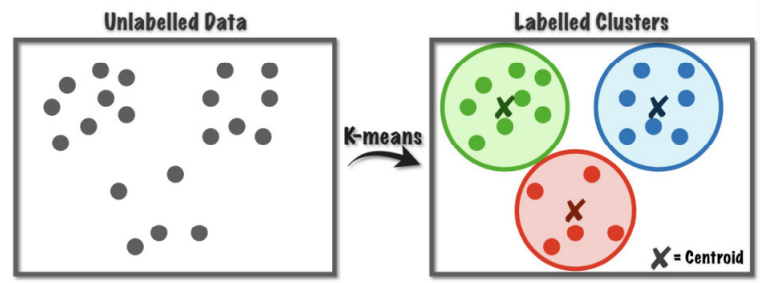
Jiawei Han, Micheline Kamber and Jian Pei, Data Mining: Concepts and Techniques, 3rd ed., The Morgan Kaufmann Series in Data Management Systems Morgan Kaufmann Publishers, July 2011. ISBN 978-0123814791

Tuong Le, PhD

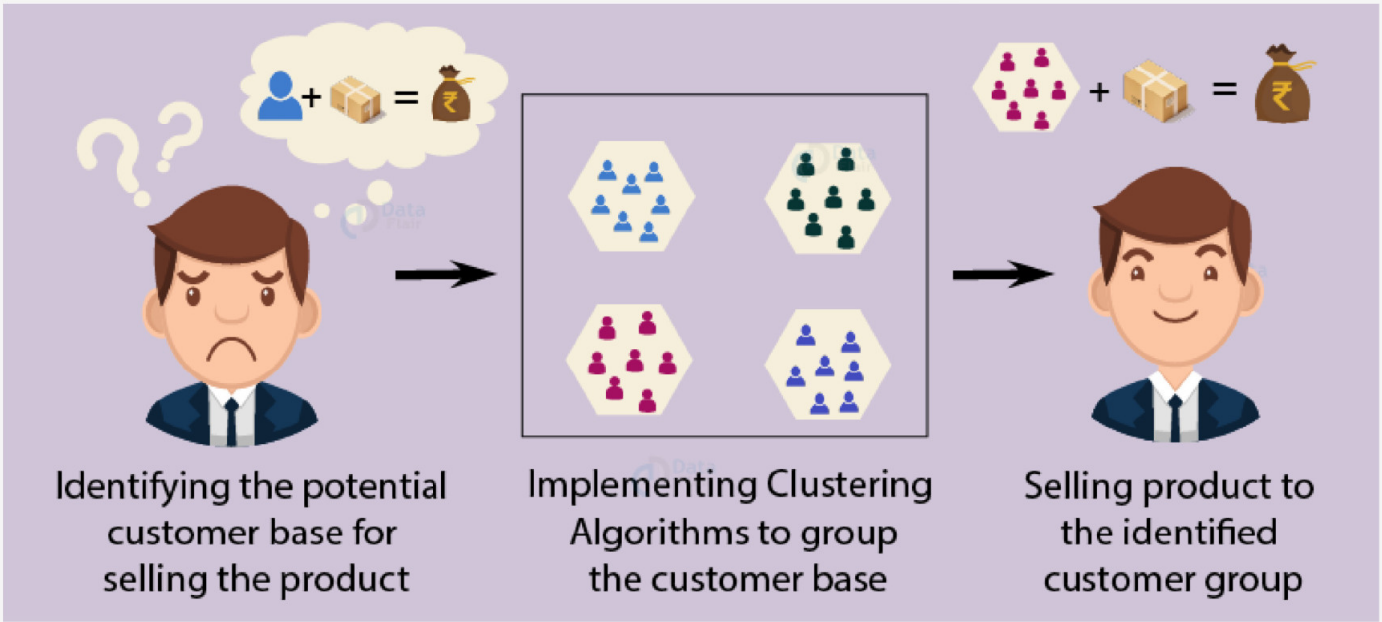
Outline

What Is Cluster Analysis?

- Cluster analysis (clustering, unsupervised learning, unsupervised learning) is the process of partitioning a set of data objects (or observations) into subsets.
  - Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters.
  - The set of clusters resulting from a cluster analysis can be referred to as a clustering
- Cluster analysis has been widely used in many applications such as business intelligence, image pattern recognition, Web search, biology, and security.

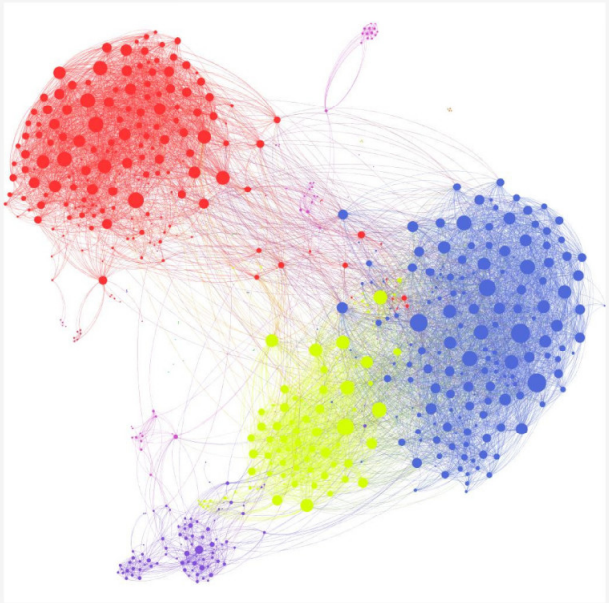


What Is Cluster Analysis: Example of Customer segmentation



Requirements for Cluster Analysis

- ❑ Scalability: Clustering all the data instead of only on samples
- ❑ Ability to deal with different types of attributes: Numerical, binary, categorical, ordinal, linked, and mixture of these
- ❑ Constraint-based clustering
  - User may give inputs on constraints
  - Use domain knowledge to determine input parameters
- ❑ Interpretability and usability
- ❑ Others
  - Discovery of clusters with arbitrary shape
  - Ability to deal with noisy data
  - Incremental clustering and insensitivity to input order
  - High dimensionality

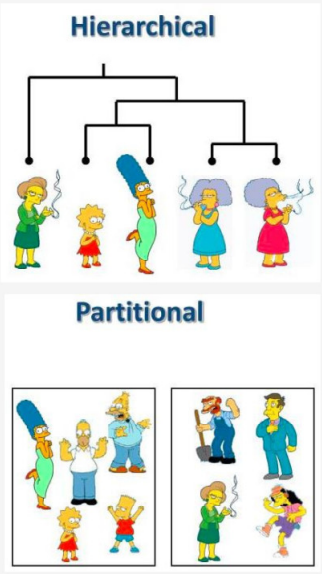


Aspects used for comparing clustering methods

- ❑ **The partitioning criteria**: Single level and multi-level hierarchical partitioning.
- ❑ **Separation of clusters**: Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- ❑ **Similarity measure**: Distance-based (e.g., Euclidian, road network, vector) vs. connectivity-based (e.g., density or contiguity)
- ❑ **Clustering space**: Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering).

Overview of Basic Clustering Methods

- **Partitioning approach** (k-means, k-medoids, CLARANS…)
  - Find mutually exclusive clusters of spherical shape; Distance-based; May use mean or medoid (etc.) to represent cluster center; Effective for small- to medium-size data sets.
- **Hierarchical approach** (Diana, Agnes, BIRCH, CAMELEON…)
  - Effective for small- to medium-size data sets; Cannot correct erroneous merges or splits; May incorporate other techniques like microclustering or consider object "linkages"
- **Density-based approach** (DBSACN, OPTICS, DenClue…)
  - Can find arbitrarily shaped clusters; Clusters are dense regions of objects in space that are separated by low-density regions; Cluster density: Each point must have a minimum number of points within its "neighborhood"; May filter out outliers.
- **Grid-based approach** (STING, WaveCluster, CLIQUE…)
  - Use a multiresolution grid data structure; Fast processing time (typically independent of the number of data objects, yet dependent on grid size)

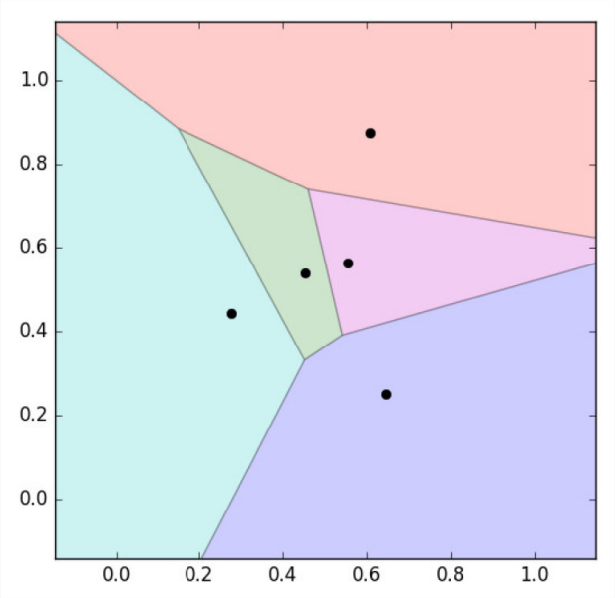


Outline

1. Cluster Analysis: Basic Concepts
2. Partitioning Methods
3. Hierarchical Methods
4. Evaluation of Clustering
5. Summary

Partitioning Methods

- Partitioning approach is the simplest and most fundamental version of cluster analysis.
- Given a data set, D, of n objects, and k, the number of clusters to form, a partitioning algorithm organizes the objects into user-specified (k) partitions ($k \leq n$), $C_1, \ldots, C_k$ that is, $C_i \subset D$ and $C_i \cap C_j = \emptyset$ for $(1 \leq i, j \leq k)$.
- There are many algorithms that come under partitioning method some of the popular ones are **k-means** and **k-medoids**.



k-Means: A Centroid-Based Technique

- A centroid-based partitioning technique uses the centroid of a cluster, C_i, to represent that cluster. The centroid of a cluster can be defined by the mean or medoid of the objects (or points) assigned to the cluster.
- The difference between an object $p \in C_i$ and c_i, the representative of the cluster, is measured by $dist(p, c_i)$, where $dist(x, y)$ is the Euclidean distance between two points x and y.
- The quality of cluster C_i can be measured by the **within-cluster variation**, which is the sum of squared error between all objects in C_i and the centroid c_i, defined as

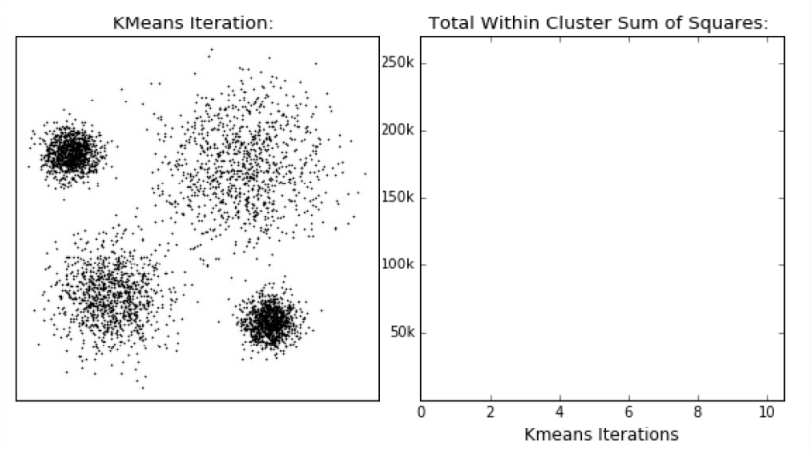
$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, c_i)^2$$

where E is the sum of the squared error for all objects in the data set; p is the point in space representing a given object; and c_i is the centroid of cluster C_i (both p and c_i are multidimensional).

- This objective function tries to make the resulting k clusters as compact and as separate as possible.

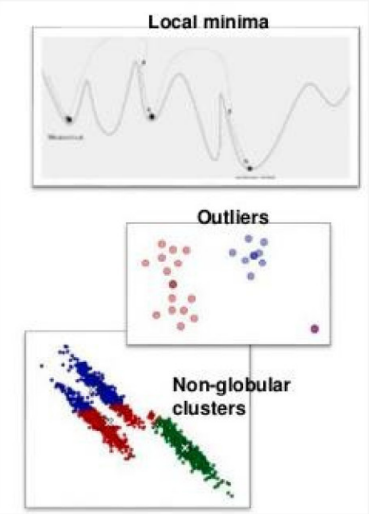
k-Means: Pseudocode

- Given a data set, D, of n objects, and k, the number of clusters, the k-means algorithm is implemented in five steps:
  1. Arbitrarily choose k objects from D as the initial cluster centers;
  2. (Re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
  3. If the assignment does not change then STOP;
  4. Update the cluster means, that is, calculate the mean value of the objects for each cluster;
  5. Go back to Step 2;



k-Means: Pros and Cons

- **Pros**: Efficient: $O(tkn)$, where n is the total number of objects, k is the number of clusters, and t is the number of iterations. Normally, k, $t \ll n$.
  - Comparing: PAM: O(k$(n-k)^2$), CLARA: O(ks^2 + k(n-k))
- **Cons**
  - Applicable only to objects in a continuous n-dimensional space
    - Using the k-modes method for categorical data
    - In comparison, k-medoids can be applied to a wide range of data
  - Need to specify k, the number of clusters. There are ways to determine the best k:
    - Elbow method
    - Silhouette Method
  - Sensitive to noisy data and outliers



k-Means: Variations

- Most of the variants of the k-means which differ in
  - Selection of the initial k means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: k-modes
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: k-prototype method

k-Means: A drawback

- Consider six points in 1-D space having the values 1, 2, 3, 8, 9, 10, and 25, respectively. Intuitively, by visual inspection we may imagine the points partitioned into the clusters {1,2,3} and {8,9,10}, meanwhile point 25 is excluded because it appears to be an outlier. How would k-means partition the values?
- If we apply k-means using k = 2, the partitioning {{1, 2, 3}, {8, 9, 10, 25}} has the within-cluster variation: $(1-2)^2+(2-2)^2+(3-2)^2+(8-13)^2+(9-13)^2+(10-13)^2+(25-13)^2=196$.
- The partitioning {{1, 2, 3, 8}, {9, 10, 25}} has the within-cluster variation: $(1-3.5)^2+(2-3.5)^2+(3-3.5)^2+(8-3.5)^2+(9-14.67)^2+(10-14.67)^2+(25-14.67)^2=189.67$.
- The latter partitioning has the lowest within-cluster variation; therefore, the k-means method assigns the value 8 to a cluster different from that containing 9 and 10 due to the outlier point 25. Moreover, the center of the second cluster, 14.67, is substantially far from all the members in the cluster.

k-Medoids: A Representative Object-Based Technique

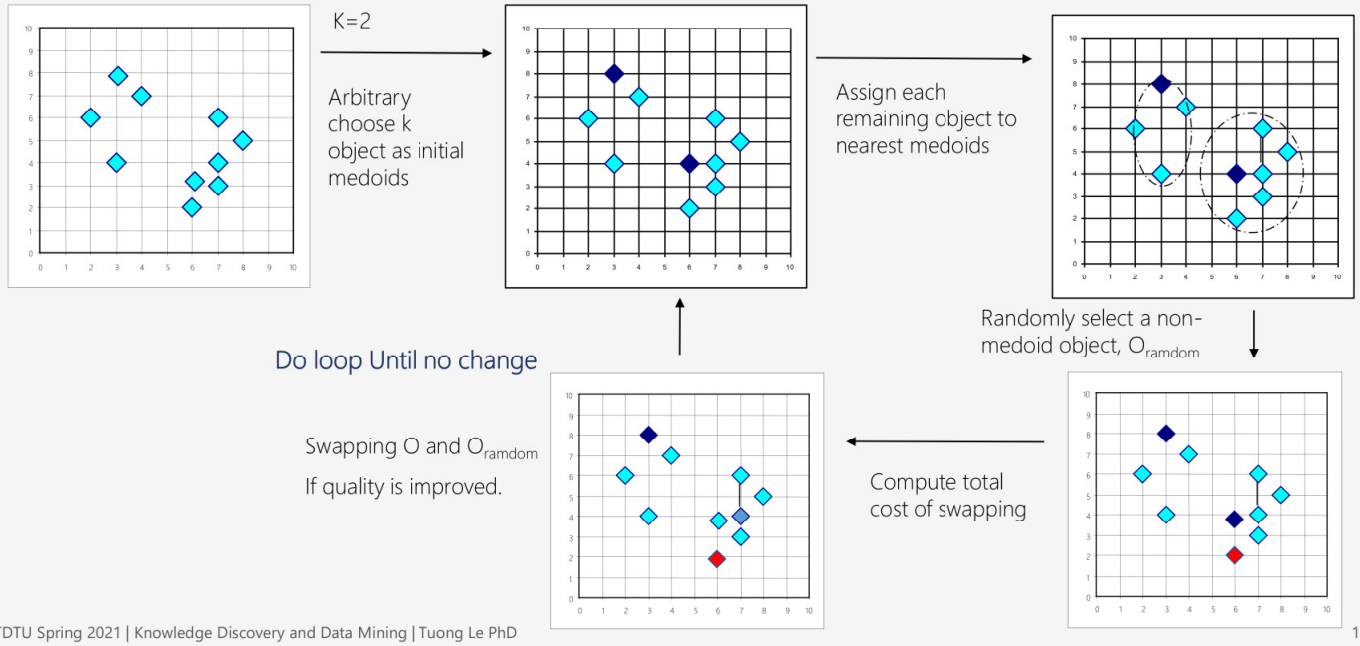
- Instead of taking the mean value of the objects in a cluster as a reference point, this method picks actual objects to represent the clusters, using one representative object per cluster.
- The partitioning method is then performed based on the principle of minimizing the sum of the dissimilarities between each object p and its corresponding representative object. That is, an absolute-error criterion is used, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, o_i)^2$$

where E is the sum of the squared error for all objects in the data set; p is the point in space representing a given object; and o_i is the representative object of cluster C_i.

- This objective function tries to minimize the absolute error.

k-Medoids: Partitioning Around Medoids (PAM) algorithm

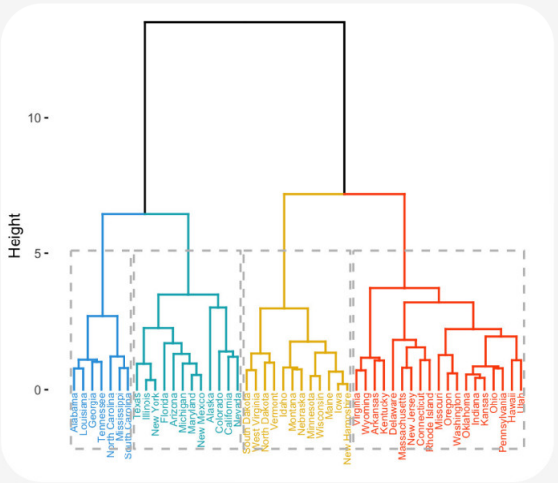


Outline

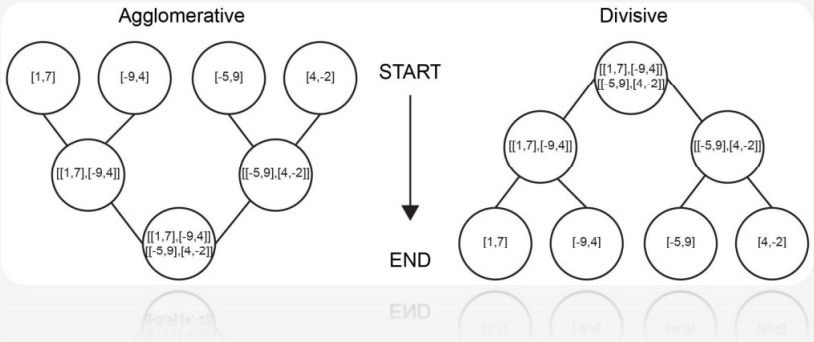
1. Cluster Analysis: Basic Concepts
2. Partitioning Methods
3. **Hierarchical Methods**
4. Evaluation of Clustering
5. Summary

Hierarchical Methods

- A hierarchical clustering method works by grouping data objects into a hierarchy or "tree" of clusters. Representing data objects in the form of a hierarchy is useful for data summarization and visualization.
- Topics for discussion
  - Agglomerative versus Divisive Hierarchical Clustering
  - Distance Measures in Algorithmic Methods
  - BIRCH: Multiphase Hierarchical Clustering Using Clustering Feature Trees
  - Chameleon: Multiphase Hierarchical Clustering Using Dynamic Modeling
  - Probabilistic Hierarchical Clustering

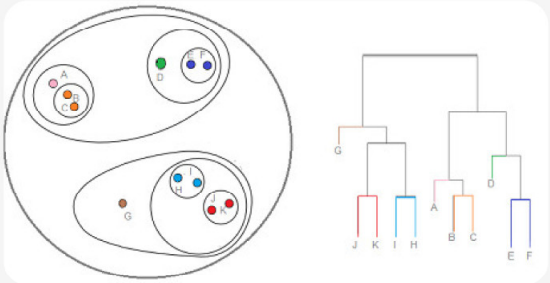


Agglomerative versus Divisive Hierarchical Clustering



- **Agglomerative**: This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- **Divisive**: This is a "top-down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

- A tree structure called a **dendrogram** is commonly used to represent the process of hierarchical clustering.
- It shows how objects are grouped together (in an agglomerative method) or partitioned (in a divisive method) step-by-step.



Distance Measures in Algorithmic Methods

- Whether using an agglomerative method or a divisive method, a core need is to measure the distance between two clusters, where each cluster is generally a set of objects. Four widely used measures for distance:

  - Minimum distance

$$dist_{min}(C_i, C_j) = \min_{p \in C_i,\, p' \in C_j} \{|p - p'|\}$$

A nearest-neighbor clustering algorithm

  - Maximum distance

$$dist_{max}(C_i, C_j) = \max_{p \in C_i,\, p' \in C_j} \{|p - p'|\}$$

A farthest-neighbor clustering algorithm

  - Mean distance

$$dist_{mean}(C_i, C_j) = |m_i - m_j|$$

  - Average distance

$$dist_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i,\, p' \in C_j} |p - p'|$$

The use of **mean or average distance** is a compromise between the minimum and maximum distances and overcomes the outlier sensitivity problem.

where m_i is the mean for C_i; and n_i is the number of objects in C_i.

BIRCH: Multiphase Hierarchical Clustering Using Clustering Feature Trees

- BIRCH is designed for clustering a large amount of numeric data by integrating hierarchical clustering and other clustering methods such as iterative partitioning.
- It overcomes the two difficulties in agglomerative clustering methods: (1) scalability and (2) the inability to undo what was done in the previous step.
- Consider a cluster of n d-dimensional data objects or points. The clustering feature (CF) of the cluster is a 3-D vector summarizing information about clusters of objects. It is defined as

$$CG = \quad n, LS, SS\rangle$$

where LS is the linear sum of the n points (i.e., $\sum_{i=1}^{n} x_i$), and SS is the square sum of the data points (i.e., $\sum_{i=1}^{n} x_i^2$).

- Using a clustering feature, we can easily derive many useful statistics of a cluster. For example,

$$x_0 = \frac{LS}{n}, \qquad R = \sqrt{\frac{nSS - 2LS^2 + nLS}{n^2}}, \qquad D = \sqrt{\frac{2nSS - 2LS^2}{n(n-1)}}$$

- Here, R is the average distance from member objects to the centroid, and D is the average pairwise distance within a cluster. Both R and D reflect the tightness of the cluster around the centroid.

BIRCH algorithm

- Summarizing a cluster using the clustering feature can avoid storing the detailed information about individual objects or points. Instead, we only need a constant size of space to store the clustering feature.
- Moreover, clustering features are additive. That is, for two disjoint clusters, C_1 and C_2, with the clustering features $CF_1 = \langle n_1, LS_1, SS_1\rangle$ and $CF_2 = \langle n_2, LS_2, SS_2\rangle$, respectively, the clustering feature for the cluster that formed by merging C_1 and C_2 is simply

$$CF_1 + CF_2 = \langle n_1 + n_2, LS_1 + LS_2, SS_1 + SS_2\rangle$$

- **Example**: Suppose there are three points, (2,5), (3,2), and (4,3), in a cluster, C_1. The clustering feature of C_1 is

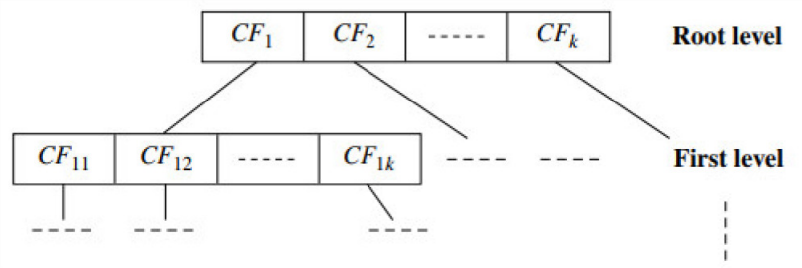
$$CF_1 = \langle 3, (2+3+4, 5+2+3), (2^2+3^2+4^2, 5^2+2^2+3^2)\rangle = \langle 3, (9,10), (29,38)\rangle$$

- Suppose that C_1 is disjoint to a second cluster, C_2, where $CF_2 = \langle 3, (35,36), (417,440)\rangle$. The clustering feature of a new cluster, C_3, that is formed by merging C_1 and C_2, is derived by adding CF_1 and CF_2. That is,

$$CF_3 = \langle 3+3, (9+35, 10+36), (29+417, 38+440)\rangle = \langle 6, (44,46), (446,478)\rangle$$

BIRCH algorithm

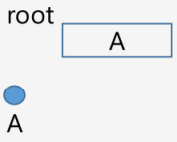
- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering.
- By definition, a nonleaf node in a tree has children.
- The nonleaf nodes store sums of the CFs of their children, and thus summarize clustering information about their children.



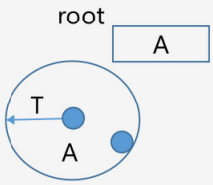
- A CF tree has two parameters:
  - **Branching factor, B**: specifies the maximum number of children per nonleaf node.
  - **Threshold, T**: specifies the maximum diameter of subclusters stored at the leaf nodes of the tree.
- These two parameters influence the size of the resulting tree

BIRCH algorithm: CF Tree

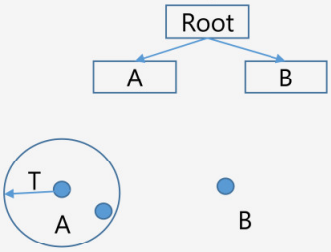
1. Initially, the data points in one cluster.



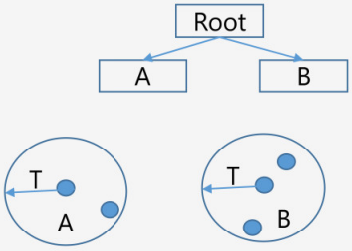
2. The data arrives, and a check is made whether the size of the cluster does not exceed T.



3. If the cluster size grows too big, the cluster is split into two clusters, and the points are redistributed.

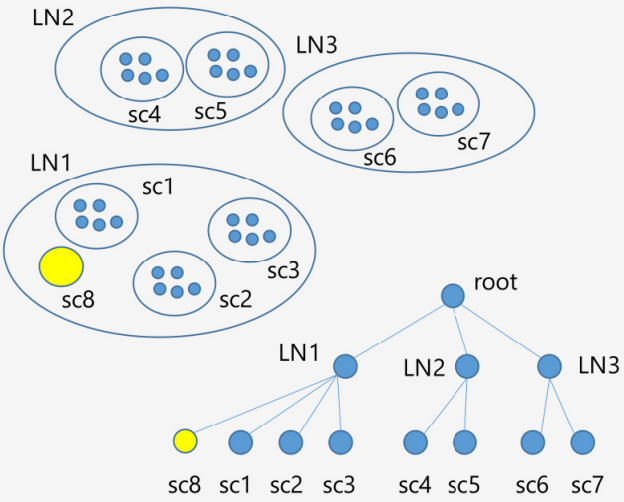


4. At each node of the tree, the CF tree keeps the clustering features (CF)

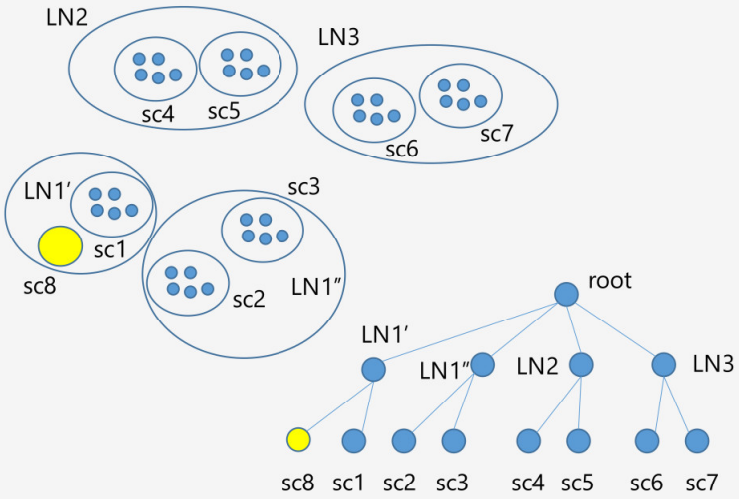


BIRCH algorithm: CF Tree

❑ Another example of the CF Tree Insertion

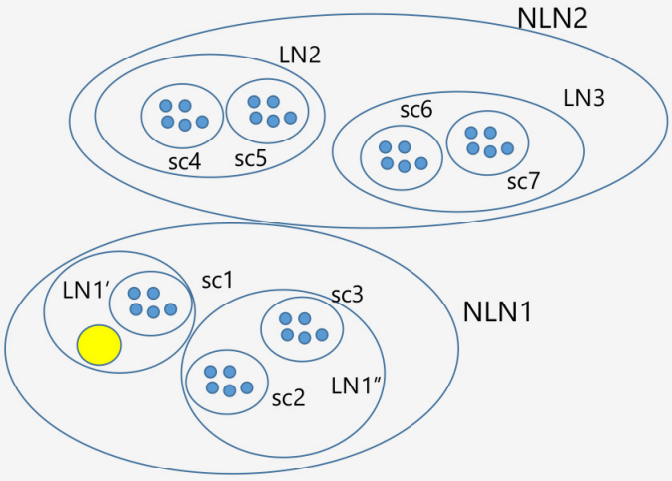


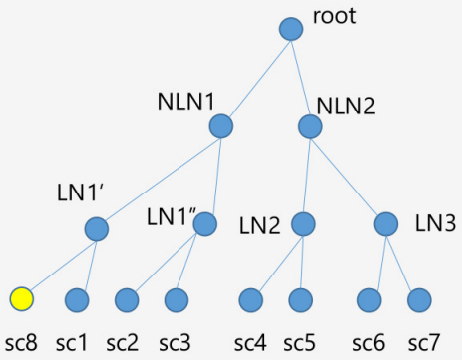
❑ If the branching factor of a leaf node can not exceed 3, then LN1 is split.



BIRCH algorithm: CF Tree

❑ If the branching factor of a non-leaf node can not exceed 3, then the root is split and the height of the CF Tree increases by one.



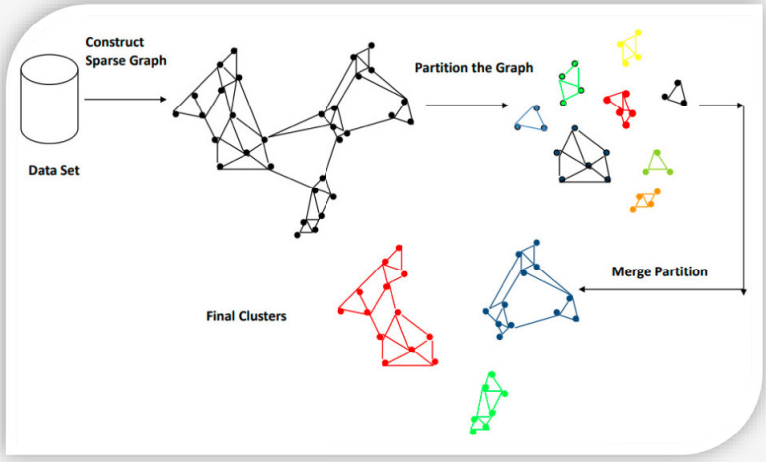


BIRCH algorithm

- The primary phases of BIRCH are:
  - **Phase 1:** BIRCH scans the database to build an initial in-memory CF-tree, which can be viewed as a multilevel compression of the data that tries to preserve the data's inherent clustering structure.
  - **Phase 2:** BIRCH applies a (selected) clustering algorithm to cluster the leaf nodes of the CF-tree, which removes sparse clusters as outliers and groups dense clusters into larger ones.
- The time complexity of BIRCH algorithm is $O(n)$ where n is the number of objects to be clustered.
- Package: sklearn.cluster.Birch (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html)

Chameleon: Multiphase Hierarchical Clustering Using Dynamic Modeling

- Chameleon is a hierarchical clustering algorithm that uses dynamic modeling to determine the similarity between pairs of clusters.
- Chameleon uses a k-nearest-neighbor graph approach to construct a sparse graph, where each vertex of the graph represents a data object, and there exists an edge between two vertices (objects) if one object is among the k-most similar objects to the other.
- Chameleon uses a graph partitioning algorithm to partition the k-nearest-neighbor graph into a large number of relatively small sub-clusters such that it minimizes the edge cut.
- Chameleon then uses an agglomerative hierarchical clustering algorithm that iteratively merges sub-clusters based on their similarity.



Chameleon algorithm

- Chameleon determines the similarity between each pair of clusters C_i and C_j according to their relative interconnectivity, $RI(C_i, C_j)$ and their relative closeness, $RC(C_i, C_j)$.

$$RI(C_i, C_j) = \frac{|EC_{\{C_i,C_j\}}|}{\frac{1}{2}\left(|EC_{C_i}| + |EC_{C_j}|\right)}$$

where $EC_{\{C_i,C_j\}}$ is the edge cut as previously defined for a cluster containing both C_i and C_j. Similarly, EC_{C_i} (or EC_{C_j}) is the minimum sum of the cut edges that partition C_i (or C_j) into two roughly equal parts.

$$RC(C_i, C_j) = \frac{\bar{S}_{EC_{\{C_i,C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\bar{S}_{EC_{C_j}}}$$

where $\bar{S}_{EC_{\{C_i,C_j\}}}$ is the average weight of the edges that connect vertices in C_i to vertices in C_j, and $\bar{S}_{EC_{C_i}}$ (or $\bar{S}_{EC_{C_j}}$) is the average weight of the edges that belong to the min-cut bisector of cluster C_i (or C_j).

- The processing cost for high-dimensional data may require $O(n^2)$ time for n objects in the worst case.

Probabilistic Hierarchical Clustering

- Probabilistic hierarchical clustering methods generally have the same efficiency as algorithmic agglomerative hierarchical clustering methods;
- It uses probabilistic models to measure the distance between clusters

$$dist(C_i, C_j) = -log\frac{P(C_1 \cup C_2)}{P(C_1)P(C_2)}$$

Algorithm: A probabilistic hierarchical clustering algorithm.

Input:

- $D = \{o_1, \ldots, o_n\}$: a data set containing n objects;

Output: A hierarchy of clusters.

Method:

(1) **create** a cluster for each object $C_i = \{o_i\}$, $1 \le i \le n$;
(2) **for** $i = 1$ to n
(3) **find** pair of clusters C_i and C_j such that $C_i, C_j = \arg\max_{i \neq j} \log \frac{P(C_i \cup C_j)}{P(C_i)P(C_j)}$;
(4) **if** $\log \frac{P(C_i \cup C_j)}{P(C_i)P(C_j)} > 0$ then merge C_i and C_j;
(5) **else** stop;

A probabilistic hierarchical clustering algorithm.

Outline

1. Cluster Analysis: Basic Concepts
2. Partitioning Methods
3. Hierarchical Methods
4. Evaluation of Clustering
5. Summary

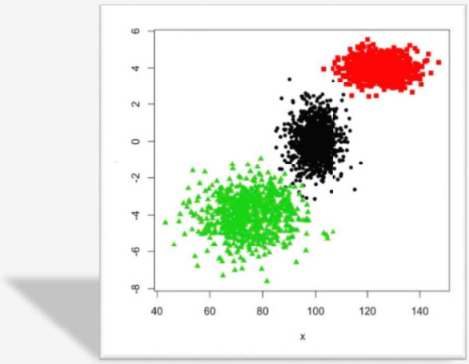
Evaluation of Clustering

The major tasks of clustering evaluation include the following:

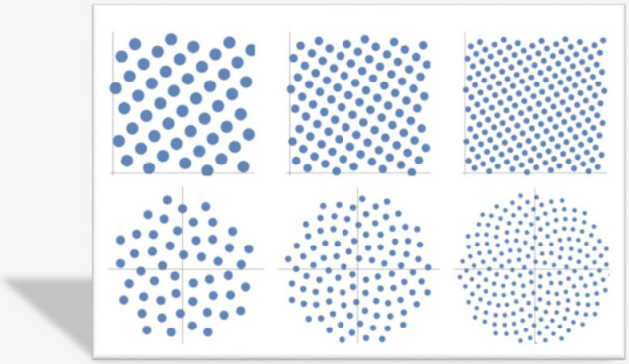
- **Assessing clustering tendency**: Clustering analysis on a data set is meaningful only when there is a nonrandom structure in the data.
- **Determining the number of clusters in a data set**: it is desirable to estimate this number even before a clustering algorithm is used to derive detailed clusters.
- **Measuring clustering quality**: A number of measures can be used. Some methods measure how well the clusters fit the data set, while others measure how well the clusters match the ground truth, if such truth is available.

Assessing Clustering Tendency: Uniform distribution

- Although a clustering algorithm may still artificially partition the points into groups, the groups will unlikely mean anything significant to the application due to the uniform distribution of the data.
- Therefore, clustering requires non-uniform distribution of data.



Non-uniform distribution of data



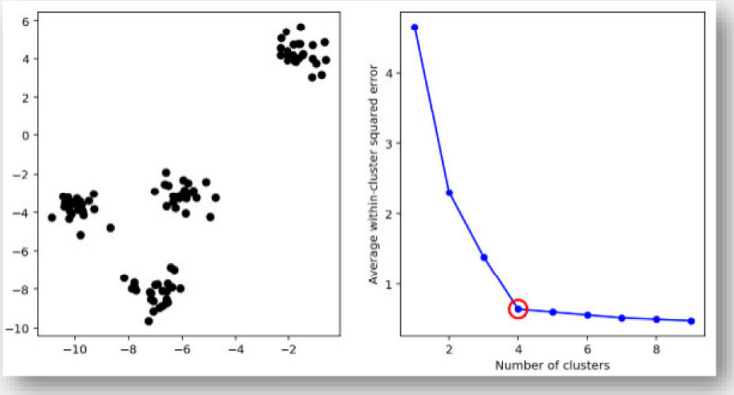
Uniform distribution of data

Assessing Clustering Tendency: Hopkins Statistic

- Try to measure the probability that the data set is generated by a uniform data distribution. This can be done by applying the Hopkins Statistic. Recommended package: https://pypi.org/project/pyclustertend/
- Given a data set, D, which is regarded as a sample of a random variable, o, we want to determine how far away o is from being uniformly distributed in the data space. We calculate the Hopkins Statistic as follows:
  1. Sample n points, $p_1, \ldots, p_n$, uniformly from D. For each point, p_i, we find the nearest neighbor of p_i ($1 \le i \le$ n) in D, and let x_i be the distance between p_i and its nearest neighbor in D. That is,
  $$x_i = \min_{v \in D}\{dist(p_i, v)\}$$
  2. Sample n points, $q_1, \ldots, q_n$, uniformly from D. For each q_i ($1 \le i \le$ n), we find the nearest neighbor of q_i in $D - \{q_i\}$, and let y_i be the distance between q_i and its nearest neighbor in $D - \{q_i\}$. That is,
  $$y_i = \min_{v \in D, v \neq q_i}\{dist(q_i, v)\}$$
  3. Calculate the Hopkins Statistic, H, as
  $$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$
- If D were uniformly distributed, H would be about 0.5 while D were highly skewed H would be close to 0. That is if $H \ge 0.5$, it is unlikely that D has statistically significant clusters.

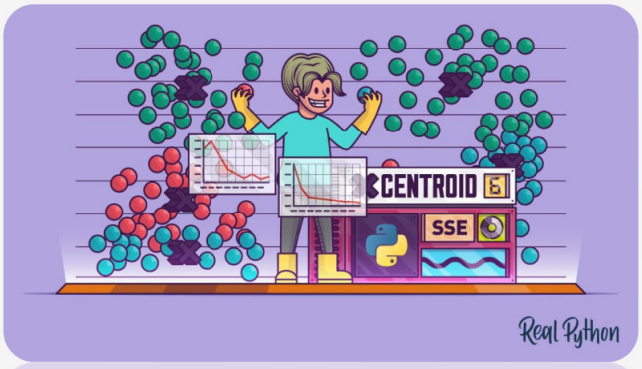
Determining the Number of Clusters

- Determining the "right" number of clusters in a data set is important, not only because some clustering algorithms like k-means require such a parameter, but also because the appropriate number of clusters controls the proper granularity of cluster analysis.
- Elbow method
  1. Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters
  2. For each k, calculate the total within-cluster sum of square (WSS)
  3. Plot the curve of wss according to the number of clusters k.
  4. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.



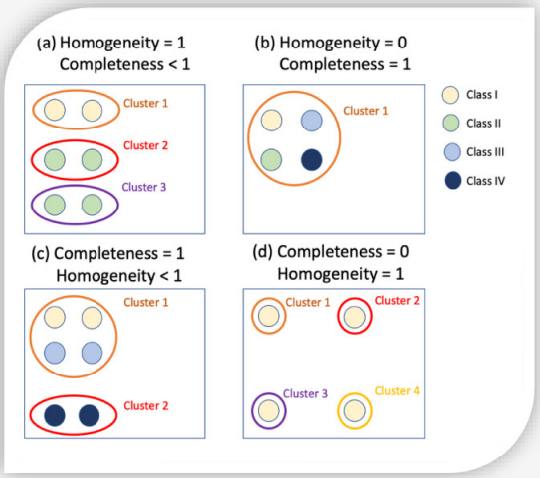
Measuring Clustering Quality

- There are a few methods to choose from for measuring the quality of a clustering which can be categorized into two groups according to whether ground truth is available.
- If ground truth is available, it can be used by **extrinsic methods**, which compare the clustering against the group truth and measure.
- If the ground truth is unavailable, we can use **intrinsic methods**, which evaluate the goodness of a clustering by considering how well the clusters are separated.



Measuring Clustering Quality: Extrinsic methods

- The core task in extrinsic methods is to assign a score, $Q(C, C_g)$, to a clustering, C, given the ground truth, C_g.
- In general, a measure Q on clustering quality is effective if it satisfies the following four essential criteria:
  - **Cluster homogeneity**. A perfectly homogeneous clustering is one where each cluster has data-points belonging to the same class label. Homogeneity describes the closeness of the clustering algorithm to this perfection.
  - **Cluster completeness**. A perfectly complete clustering is one where all data-points belonging to the same class are clustered into the same cluster. Completeness describes the closeness of the clustering algorithm to this perfection.
  - **Rag bag**. A perfectly "Rag bag" clustering has a rag bag that contains noisy objects from various categories according to ground truth.
  - **Small cluster preservation**. A perfectly "small cluster preservation" clustering is one which tend to splits the big category rather than the small category.



Measuring Clustering Quality: Extrinsic methods

- Many clustering quality measures (such as **BCubed precision** and **BCubed recall metrics**) satisfy some of these four criteria.
- Let $D = \{o_1, \dots, o_n\}$ be a set of objects, and C be a clustering on D. Let $L(o_i)(1 \le i \le n)$ be the category of o_i given by ground truth, and $C(o_i)$ be the cluster_ID of o_i in C. Then, for two objects, o_i and o_j, $(1 \le i, j \le n, i \ne j)$ the correctness of the relation between o_i and o_j in clustering C is given by

$$Correctness(o_i, o_j) = \begin{cases} 1, & if\ L(o_i) = L(o_j) \Leftrightarrow C(o_i) = C(o_j) \\ 0, & \text{otherwise} \end{cases}$$

- **BCubed precision** and **BCubed recall** are defined as

$$\text{Precision BCubed} = \frac{\sum_{i=1}^{n} \frac{\sum_{\boldsymbol{o_j}: i \ne j, C(\boldsymbol{o_i}) = C(\boldsymbol{o_j})} \text{Correctness}(\boldsymbol{o_i}, \boldsymbol{o_j})}{\|\{\boldsymbol{o_j} | i \ne j, C(\boldsymbol{o_i}) = C(\boldsymbol{o_j})\}\|}}{n}$$

$$\text{Recall BCubed} = \frac{\sum_{i=1}^{n} \frac{\sum_{\boldsymbol{o_j}: i \ne j, L(\boldsymbol{o_i}) = L(\boldsymbol{o_j})} \text{Correctness}(\boldsymbol{o_i}, \boldsymbol{o_j})}{\|\{\boldsymbol{o_j} | i \ne j, L(\boldsymbol{o_i}) = L(\boldsymbol{o_j})\}\|}}{n}$$

- Package: https://github.com/hhromic/python-bcubed

Measuring Clustering Quality: Intrinsic Methods

- The silhouette coefficient is an intrinsic method that evaluate a clustering by examining how well the clusters are separated and how compact the clusters are.
- For a data set, D, of n objects, suppose D is partitioned into k clusters, $C_1, \ldots, C_k$. For each object $o \in D$, we calculate $a(o)$ as the average distance between o and all other objects in the cluster to which o belongs. Similarly, $b(o)$ is the minimum average distance from o to all clusters to which o does not belong:

$$a(\boldsymbol{o}) = \frac{\sum_{\boldsymbol{o}' \in C_i, \boldsymbol{o} \neq \boldsymbol{o}'} dist(\boldsymbol{o}, \boldsymbol{o}')}{|C_i| - 1} \qquad b(\boldsymbol{o}) = \min_{C_j : 1 \leq j \leq k, j \neq i} \left\{ \frac{\sum_{\boldsymbol{o}' \in C_j} dist(\boldsymbol{o}, \boldsymbol{o}')}{|C_j|} \right\}$$

- The silhouette coefficient of o is then defined as

$$s(o) = \frac{b(o) - a(o)}{\max(a(o), b(o))}$$

- The value of the silhouette coefficient is between -1 and 1. The smaller the value, the more compact the cluster.
- To measure a cluster's fitness within a clustering, we can compute the average silhouette coefficient value of all objects in the cluster. To measure the quality of a clustering, we can use the average silhouette coefficient value of all objects in the data set.
- The silhouette coefficient and other intrinsic measures can also be used in **the elbow method** to heuristically derive the number of clusters in a data set by replacing the sum of within-cluster variances.

Outline

Summary

- **A cluster** is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering.
- **Clustering algorithm categories** are: **partitioning methods**, **hierarchical methods**, **density-based methods**, and **grid-based methods**.
- **A partitioning method** first creates an initial set of k partitions, where parameter k is the number of partitions to construct. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. Typical partitioning methods include k-means, k-medoids.
- **A hierarchical method** creates a hierarchical decomposition of the given set of data objects. The method can be classified as being either agglomerative (bottom-up) or divisive (top-down), based on how the hierarchical decomposition is formed.
- **Clustering evaluation** assesses the feasibility of clustering analysis on a data set and the quality of the results generated by a clustering method. The tasks include assessing clustering tendency, determining the number of clusters, and measuring clustering quality.