Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. A Categorization of Major Clustering Methods
3. Partitioning Methods
4. Hierarchical Methods
5. Density-Based Methods
6. Grid-Based Methods
7. Model-Based Methods
8. Clustering High-Dimensional Data
9. Constraint-Based Clustering
10. Link-based clustering
11. Outlier Analysis
12. Summary
What is Cluster Analysis?

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms
Clustering for Data Understanding and Applications

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market research
Clustering as Preprocessing Tools (Utility)

- **Summarization:**
  - Preprocessing for regression, PCA, classification, and association analysis

- **Compression:**
  - Image processing: vector quantization

- **Finding K-nearest Neighbors**
  - Localizing search to one or a small number of clusters
Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns
Measure the Quality of Clustering

- Dissimilarity/Similarity metric
  - Similarity is expressed in terms of a distance function, typically metric: \( d(i, j) \)
  - The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables
  - Weights should be associated with different variables based on applications and data semantics

- Quality of clustering:
  - There is usually a separate “quality” function that measures the “goodness” of a cluster.
  - It is hard to define “similar enough” or “good enough”
    - The answer is typically highly subjective
Distance Measures for Different Kinds of Data

Discussed in Chapter 2: Data Preprocessing

- Numerical (interval)-based:
  - Minkowski Distance:
    - Special cases: Euclidean ($L_2$-norm), Manhattan ($L_1$-norm)
- Binary variables:
  - symmetric vs. asymmetric (Jaccard coeff.)
- Nominal variables: # of mismatches
- Ordinal variables: treated like interval-based
- Ratio-scaled variables: apply log-transformation first
- Vectors: cosine measure
- Mixed variables: weighted combinations
Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Ability to handle dynamic data
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability
Major Clustering Approaches (I)

- **Partitioning approach:**
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS

- **Hierarchical approach:**
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, ROCK, CAMELEON

- **Density-based approach:**
  - Based on connectivity and density functions
  - Typical methods: DBSCAN, OPTICS, DenClue

- **Grid-based approach:**
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE
Major Clustering Approaches (II)

- **Model-based:**
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, SOM, COBWEB

- **Frequent pattern-based:**
  - Based on the analysis of frequent patterns
  - Typical methods: p-Cluster

- **User-guided or constraint-based:**
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering

- **Link-based clustering:**
  - Objects are often linked together in various ways
  - Massive links can be used to cluster objects: SimRank, LinkClus
Calculation of Distance between Clusters

- **Single link:** smallest distance between an element in one cluster and an element in the other, i.e., \( \text{dist}(K_i, K_j) = \min(t_{ip}, t_{jq}) \)
- **Complete link:** largest distance between an element in one cluster and an element in the other, i.e., \( \text{dist}(K_i, K_j) = \max(t_{ip}, t_{jq}) \)
- **Average:** avg distance between an element in one cluster and an element in the other, i.e., \( \text{dist}(K_i, K_j) = \text{avg}(t_{ip}, t_{jq}) \)
- **Centroid:** distance between the centroids of two clusters, i.e., \( \text{dist}(K_i, K_j) = \text{dist}(C_i, C_j) \)
- **Medoid:** distance between the medoids of two clusters, i.e., \( \text{dist}(K_i, K_j) = \text{dist}(M_i, M_j) \)
  - Medoid: one chosen, centrally located object in the cluster
Centroid, Radius and Diameter of a Cluster (for numerical data sets)

- Centroid: the “middle” of a cluster
  
  \[ C_m = \frac{\sum_{i=1}^{N} (t_{ip})}{N} \]

- Radius: square root of average distance from any point of the cluster to its centroid
  
  \[ R_m = \sqrt{\frac{\sum_{i=1}^{N} (t_{ip} - c_m)^2}{N}} \]

- Diameter: square root of average mean squared distance between all pairs of points in the cluster
  
  \[ D_m = \sqrt{\frac{\sum_{i=1}^{N} \sum_{i'=1}^{N} (t_{ip} - t_{iq})^2}{N(N-1)}} \]
Partitioning Algorithms: Basic Concept

- **Partitioning method:** Construct a partition of a database $D$ of $n$ objects into a set of $k$ clusters, s.t., min sum of squared distance

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} (p - m_i)^2$$

- Given a $k$, find a partition of $k$ clusters that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: $k$-means and $k$-medoids algorithms
    - $k$-means (MacQueen’67): Each cluster is represented by the center of the cluster
    - $k$-medoids or PAM (Partition around medoids) (Kaufman & Rousseeuw’87): Each cluster is represented by one of the objects in the cluster
The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., *mean point*, of the cluster)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when no more new assignment
The *K-Means* Clustering Method

- **Example**

  - Arbitrarily choose *K* objects as initial cluster centers.
  - Assign each object to the most similar cluster center.
  - Update the cluster means.
  - Reassign objects to the closest cluster center.
  - Repeat until convergence.

---

K=2

Arbitrarily choose K objects as initial cluster centers.
Comments on the *K-Means* Method

- **Strength:** Relatively efficient: $O(tkn)$, where $n$ is # objects, $k$ is # clusters, and $t$ is # iterations. Normally, $k, t << n$.
  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$

- **Comment:** Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*

- **Weakness**
  - Applicable only when *mean* is defined, then what about categorical data?
  - Need to specify $k$, the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*
Variations of the *K-Means* Method

- A few 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* (Huang’98)
  - 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
What Is the Problem of the K-Means Method?

- The k-means algorithm is sensitive to outliers!
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the *most centrally located* object in a cluster.
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters $k$ as an input, but needs a termination condition.

Diagram showing the steps of agglomerative (AGNES) and divisive (DIANA) clustering.
AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster
Dendrogram: Shows How the Clusters are Merged

Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.