Unit 4
Clustering

Basic Concept and Terminologies

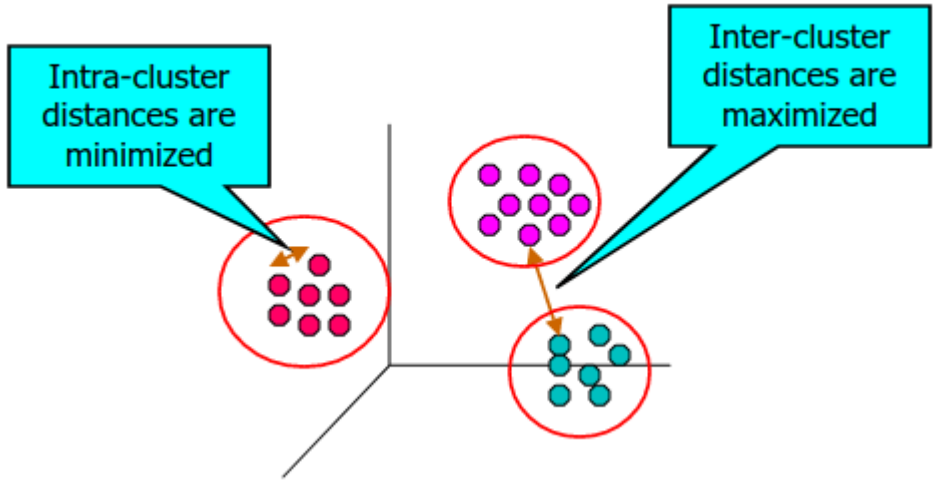
Rupak Raj Ghimire

Objective

- Basic Concept of Clustering

What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



What is Cluster Analysis?

- A cluster of data objects can be treated collectively as one group and so may be considered as a form of data Compression
- The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering.
- 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.

Cluster vs. Classification

- The classification is an effective means for distinguishing groups or classes of objects, it requires the often costly collection and labeling of a large set of training tuples or patterns, which the classifier uses to model each group
- It is often more desirable to proceed in the reverse direction: First partition the set of data into groups based on data similarity (e.g., using clustering), and then assign labels to the relatively small number of groups.
- Additional advantages of such a clustering-based process are that it is adaptable to changes and helps single out useful features that distinguish different groups

Applications of Cluster Analysis

- Land Use detection
- Crop / Forest Type identification
- Data segmentation
- Fault Isolation
- Outlier Detection

What is not Cluster Analysis?

- Supervised classification
  - Have class label information
- Simple segmentation
  - Dividing students into different registration groups alphabetically, by last name
- Results of a query
  - Groupings are a result of an external specification
- Graph partitioning
  - Some mutual relevance and synergy, but areas are not identical

Challenges of Clustering

- Scalability:
  - Many clustering algorithms work well on small data sets containing fewer than several hundred data objects; however, a large database may contain millions of objects.
  - Clustering on a sample of a given large data set may lead to biased results.
  - Highly scalable clustering algorithms are needed.

Challenges of Clustering

- Ability to deal with different types of attributes
  - Many algorithms are designed to cluster interval-based (numerical) data.
  - However, applications may require clustering other types of data, such as binary, categorical (nominal), and ordinal data, or mixtures of these data types.

Challenges of Clustering

- Discovery of clusters with arbitrary shape
  - Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures.
  - Algorithms based on such distance measures tend to find spherical clusters with similar size and density.
  - However, a cluster could be of any shape. It is important to develop algorithms that can detect clusters of arbitrary shape

Challenges of Clustering

- Ability to deal with noisy data
  - Most real-world databases contain outliers or missing, unknown, or erroneous data.
  - Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.

Challenges of Clustering

- Incremental clustering and insensitivity to the order of input records
  - Some clustering algorithms cannot incorporate newly inserted data (i.e., database updates) into existing clustering structures and, instead, must determine a new clustering from scratch.
  - Some clustering algorithms are sensitive to the order of input data.
    - That is, given a set of data objects, such an algorithm may return dramatically different clustering depending on the order of presentation of the input objects.
  - It is important to develop incremental clustering algorithms and algorithms that are insensitive to the order of input.

Challenges of Clustering

- Incremental clustering and insensitivity to the order of input records
  - Some clustering algorithms cannot incorporate newly inserted data (i.e., database updates) into existing clustering structures and, instead, must determine a new clustering from scratch.
  - Some clustering algorithms are sensitive to the order of input data.
    - That is, given a set of data objects, such an algorithm may return dramatically different clustering depending on the order of presentation of the input objects.
  - It is important to develop incremental clustering algorithms and algorithms that are insensitive to the order of input.

Challenges of Clustering

- High dimensionality
  - A database or a data warehouse can contain several dimensions or attributes. Many clustering algorithms are good at handling low-dimensional data, involving only two to three dimensions.
  - Human eyes are good at judging the quality of clustering for up to three dimensions.
  - Finding clusters of data objects in high dimensional space is challenging, especially considering that such data can be sparse and highly skewed.

Challenges of Clustering

- Constraint-based clustering
  - Real-world applications may need to perform clustering under various kinds of constraints. Suppose that your job is to choose the locations for a given number of new automatic banking machines (ATMs) in a city.
  - To decide upon this, you may cluster households while considering constraints such as the city's rivers and highway networks, and the type and number of customers per cluster.
  - A challenging task is to find groups of data with good clustering behavior that satisfy specified constraints.

Challenges of Clustering

- Interpretability and usability
  - Users expect clustering results to be interpretable, comprehensible, and usable.
  - That is, clustering may need to be tied to specific semantic interpretations and applications.
  - It is important to study how an application goal may influence the selection of clustering features and methods.

Types of Data in Cluster Analysis

- Suppose that a data set to be clustered contains *n* objects, which may represent persons, houses, documents, countries, and so on.
- Main memory-based clustering algorithms typically operate on either of the following two data structures
  - Data Matrix: Object-by-variable-structure
  - Dissimilarity Matrix: object-by-object structure

Types of Data in Cluster Analysis

- Suppose that a data set to be clustered contains *n* objects, which may represent persons, houses, documents, countries, and so on.
- Main memory-based clustering algorithms typically operate on either of the following two data structures
  - Data Matrix: Object-by-variable-structure
  - Dissimilarity Matrix: object-by-object structure

Data Matrix

- This represents *n* objects, such as persons, with *p* variables (also called measurements or attributes), such as age, height, weight, gender, and so on.
- The structure is in the form of a relational table, or *n-by-p* matrix (n objects × p variables):

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

Dissimilarity Matrix

- This stores a collection of proximities that are available for all pairs of *n* objects. It is often represented by an *n-by-n* table
  - where d(i, j) is the measured difference or dissimilarity between objects i and j.
  - In general, d(i, j) is a nonnegative number that is close to 0 when objects i and j are highly similar or “near” each other, and becomes larger the more they differ.
  - Since d(i, j) = d(j, i), and d(i, i) = 0, we have the matrix in (7.2).

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Interval-Scaled Variables

- Interval-scaled variables are continuous measurements of a roughly linear scale
- Typical examples include weight and height, latitude and longitude coordinates (e.g., when clustering houses), and weather temperature
- The measurement unit used can affect the clustering analysis. For example, changing measurement units from meters to inches for height, or from kilograms to pounds for weight, may lead to a very different clustering structure

Interval-Scaled Variables

- To help avoid dependence on the choice of measurement units, the data should be standardized.
- Standardizing measurements attempts to give all variables an equal weight.

Standarization

- To standardize measurements, one choice is to convert the original measurements to unitless variables.
- Given measurements for a variable f , this can be performed as follows

1. Calculate the **mean absolute deviation**, s_f:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots + |x_{nf} - m_f|), \tag{7.3}$$

where $x_{1f}, \ldots, x_{nf}$ are n measurements of f, and m_f is the *mean* value of f, that is, $m_f = \frac{1}{n}(x_{1f} + x_{2f} + \cdots + x_{nf})$.

2. Calculate the **standardized measurement**, or **z-score**:

$$z_{if} = \frac{x_{if} - m_f}{s_f}. \tag{7.4}$$

Important Note

- Standardization may or may not be useful in a particular application. Thus the choice of whether and how to perform standardization should be left to the user

Dissimilarity Measure

- Euclidean Distance
- Manhattan (or City Block) Distance
- Minkowski Distance

Euclidean Distance

- The most popular distance measure is Euclidean distance

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{in} - x_{jn})^2}, \tag{7.5}$$

where $i = (x_{i1}, x_{i2}, \ldots, x_{in})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jn})$ are two n-dimensional data objects.

Manhattan Distance

- Another well-known metric is Manhattan (or city block) distance, defined as

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{in} - x_{jn}|. \qquad (7.6)$$

- Both the Euclidean distance and Manhattan distance satisfy the following mathematic requirements of a distance function:
  1. d(i, j) ≥ 0: Distance is a nonnegative number.
  2. d(i, i) = 0: The distance of an object to itself is 0.
  3. d(i, j) = d(j, i): Distance is a symmetric function.
  4. d(i, j) ≤ d(i, h) + d(h, j): Going directly from object i to object j in space is o more than making a detour over any other object h (triangular inequality)

Minkowski Distance

- Minkowski distance is a generalization of both Euclidean distance and Manhattan distance. It is defined as

$$d(i, j) = (|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \cdots + |x_{in} - x_{jn}|^p)^{1/p}, \qquad (7.7)$$

- where *p* is a positive integer. Such a distance is also called *Lp* norm, in some literature.
- It represents the Manhattan distance when *p = 1* (i.e., *L1 norm*) and Euclidean distance when *p = 2 (i.e., L2 norm)*.

Weighted Euclidean distance

- f each variable is assigned a weight according to its perceived importance, the weighted Euclidean distance can be computed as

$$d(i, j) = \sqrt{w_1|x_{i1} - x_{j1}|^2 + w_2|x_{i2} - x_{j2}|^2 + \cdots + w_m|x_{in} - x_{jn}|^2}. \qquad (7.8)$$

- Weighting can also be applied to the Manhattan and Minkowski distances

Categories of the Clustering Methods

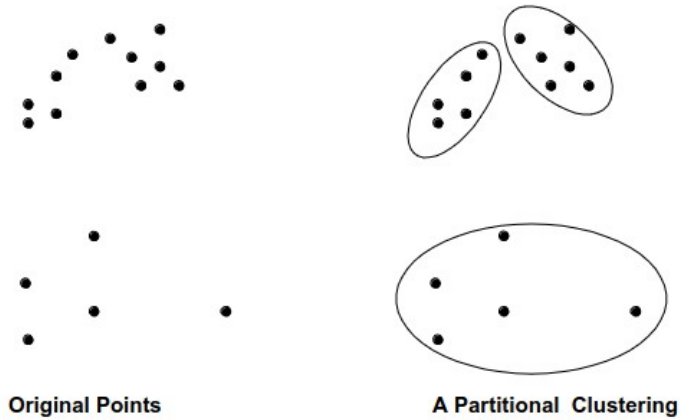
- Many clustering algorithms exist in the literature.
- It is difficult to provide a crisp cate gorization of clustering methods because these categories may overlap, so that a method may have features from several categories
  - Partitioning Methods
  - Hierarchical methods
  - Density-based methods
  - Grid-based methods
  - Model-based methods

Partitioning Methods

- Given a database of *n* objects or data tuples
  - a partitioning method constructs *k* partitions of the data, where each partition represents a cluster and $k \leq n$.
- It classifies the data into k groups, which together satisfy the following requirements:
  - each group must contain at least one object, and
  - Each object must belong to exactly one group.

Partitioning Methods

- Given *k*, the number of partitions to construct, a partitioning method creates an *initial partitioning*.
- It then uses an ***iterative relocation technique*** that attempts to improve the partitioning by moving objects from one group to another

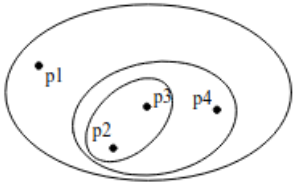


Hierarchical methods

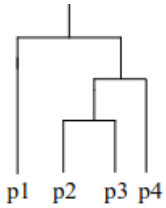
- A hierarchical method creates a hierarchical decomposition of the given set of data objects.
- A hierarchical method can be classified as being either **agglomerative or divisive**, based on how the hierarchical decomposition is formed.
- The agglomerative approach, also called the **bottom-up approach**

Hierarchical methods

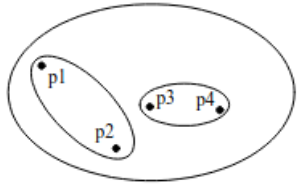
- How it works?
  - Starts with all of the objects in the same cluster.
  - In each successive iteration, a cluster is split up into smaller clusters, until eventually each object is in one cluster, or until a termination condition holds.



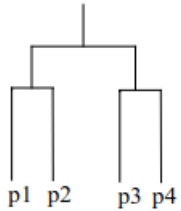
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Density-based methods

- Distance based method unable to cluster the arbitrary shape of the cluster
- Other clustering methods have been developed based on the notion of density.
- Their general idea is to continue growing the given cluster as long as the density (number of objects or datapoints) in the “neighborhood” exceeds some threshold;
  - that is, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points.
- Such a method can be used to **filter out noise (outliers)** and discover clusters of arbitrary shape.

Grid-based methods

- Grid-based methods quantize the object space into a finite number of cells that form a grid structure.
- All of the clustering operations are performed on the grid structure (i.e., on the quantized space).
  - The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space.

Model-based Method

- Model-based methods hypothesize a model for each of the clusters and find the best fit of the data to the given model.
- A model-based algorithm may locate clusters by constructing a density function that reflects the spatial distribution of the data points.
- It also leads to a way of automatically determining the number of clusters based on standard statistics, taking “noise” or outliers into account and thus yielding robust clustering methods.
- Example: EM, COBWEB, SOM (Neural network based)

Partitioning Methods

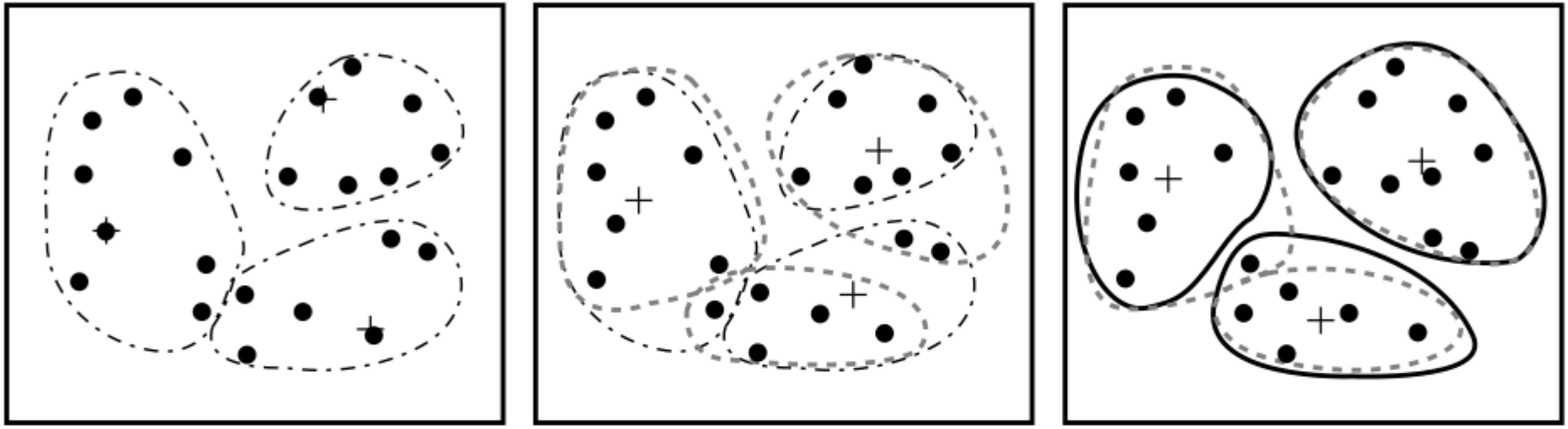
- Given **D**, a data set of n objects, and **k**, the number of clusters to form,
  - a partitioning algorithm organizes the objects into **k** partitions **(k ≤ n)**, where each partition represents a cluster.
  - The clusters are formed to optimize an objective partitioning criterion, such as a dissimilarity function based on distance, so that the objects within a cluster are “similar,” whereas the objects of different clusters are “dissimilar” in terms of the data set attributes.

Centroid-Based Technique

- The k-Means Method

- The k-means algorithm takes the input parameter, **k**, and partitions a set of n objects into **k clusters** so that the resulting intra-cluster similarity is high but the inter-cluster similarity is low.
- Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or center of gravity.

The k-Means Method



Clustering of a set of objects based on the *k*-means method. (The mean of each cluster i
marked by a "+".)

The k-Means Method

Algorithm: *k*-means. The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

- k: the number of clusters,
- D: a data set containing n objects.

Output: A set of k clusters.

Method:

(1) arbitrarily choose k objects from D as the initial cluster centers;
(2) **repeat**
(3) (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;
(4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;
(5) **until** no change;

Limitation of K-Means

- Can be applied only when the mean of a cluster is defined
  - When data has categorical attributes, k means can not be applied.
- The necessity for users to specify k, the number of clusters, in advance can be seen as a disadvantage
- It is sensitive to noise and outlier data points because a small number of such data can substantially influence the mean value.

K-Modes Method

- Another variant to k-means is the k-modes method
- It extends the k-means paradigm to **cluster categorical data** by **replacing the means of clusters with modes**, \
- Using new dissimilarity measures to deal with categorical objects and a frequency-based method to update modes of clusters.

- The k-means and the k-modes methods can be integrated to cluster data with mixed numeric and categorical values.

EM algorithm

- The EM (Expectation-Maximization) algorithm extends the k-means paradigm in a different way.
- Whereas the k-means algorithm assigns each object to a cluster, in EM each object is assigned to each cluster according to a weight representing its probability of membership. In other words, there are no strict boundaries between clusters.
- Therefore, new means are computed based on weighted measure

Thank you