Data Mining (CS 451)

JAN-MAY’ 18

PRAGYA VERMA

RECAP

- A Pattern-Growth Approach for Mining Frequent Itemsets
- Mining Frequent Itemsets Using the Vertical Data Format
- Pattern Evaluation Methods

CONTENTS

- Cluster Analysis: Basic Concepts and Methods
  - What is Cluster Analysis?
  - Requirements for Cluster Analysis
  - Overview of Basic Clustering Methods
  - Partitioning Methods
    - k-Means
    - k-Medoids
  - Hierarchical Methods
  - Density Based Methods
  - Evaluation of Clustering

What is Cluster Analysis?

- Cluster Analysis or simply clustering 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 clustering.
- Clustering is also referred to as automatic classification.
- Also known as data segmentation

Requirements for Cluster Analysis

- The following are the typical requirements of clustering in data mining:
  - Scalability
  - Ability to deal with different types of attributes
  - Ability to deal with noisy data
  - Insensitivity to input order
  - Capability of clustering high-dimensional data
  - Constraint based clustering
  - Interpretability and usability

Overview of Basic Clustering Methods

Method	General Characteristics
Partitioning methods	– 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 methods	– Clustering is a hierarchical decomposition (i.e., multiple levels) – Cannot correct erroneous merges or splits – May incorporate other techniques like microclustering or consider object “linkages”
Density-based methods	– 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

Partitioning Methods

- The simplest and most fundamental version of cluster analysis is partitioning, which organizes the objects of a set into several exclusive groups or cluster.
- Assumption: The number of clusters is given as background knowledge.
- Formally, given a set, D, of n objects, and k, the number of clusters to form, a partitioning algorithm organizes the objects into k partitions ($k \leq n$), where each partition represents a cluster.
- The clusters are formed to optimize an objective criterion such as dissimilarity function based on distance, so that the objects within a cluster are “similar” to one another and “dissimilar” to objects in other clusters.

Partitioning Methods: k-Means

- Suppose a data set, D, contains n objects in Euclidean space.
- Partitioning methods distribute the objects in D into k clusters C_1, C_2,..C_k, that is $C_i \subset D$ and $C_i \cap C_j = \emptyset$ for ($1 \leq i,j \leq k$).
- An objective function is used to assess the partitioning quality so that objects within a cluster are similar to one another but dissimilar to objects in other clusters.
- A centroid based partitioning technique uses the centroid of a cluster, C_i, to represent that cluster.
- The centroid can be defined in various ways such as by the mean or medoid of the objects (or points) assigned to the cluster.

Partitioning Methods: k-Means (Contd.)

- The difference between an object p ϵ C_i and $\mathbf{c_i}$, 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 the 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 $\mathbf{c_i}$, defined as

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

- Where E is the sum of squared error for all objects in the data set; **p** is the point in space representing a given object, and $\mathbf{c_i}$ is the centroid of the cluster C_i

Partitioning Methods: k-Means (Contd.)

- k-Means Algorithm:

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, that is, calculate the mean value of the objects for each cluster;
(5) **until** no change;

Partitioning Methods: k-Medoids

- The k-Means algorithm is sensitive to outliers.
- Thus, instead of taking the mean value of the objects in a cluster as a reference point, we can pick actual objects to represent the clusters, using one representative object per cluster.
- Each remaining object is assigned to the cluster of which the representative object is the most similar.
- An absolute-error criterion is used, defined as

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

- Where E is the sum of absolute error for all objects **p** in the data set, and $\mathbf{o_i}$ is the representative object of C_i

Partitioning Methods: k-Medoids (Contd.)

- k-Medoid Algorithm

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 in D as the initial representative objects or seeds;
(2) **repeat**
(3) assign each remaining object to the cluster with the nearest representative object;
(4) randomly select a nonrepresentative object, o_{random};
(5) compute the total cost, S, of swapping representative object, o_j, with o_{random};
(6) **if** $S < 0$ **then** swap o_j with o_{random} to form the new set of k representative objects;
(7) **until** no change;

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 clusters in useful for data summarization and visualization.
- A hierarchical clustering method can be either *agglomerative* or *divisive*, depending on whether the hierarchical decomposition is formed in bottom-up (merging) or top-down (splitting) fashion.

Hierarchical Methods: Agglomerative Clustering

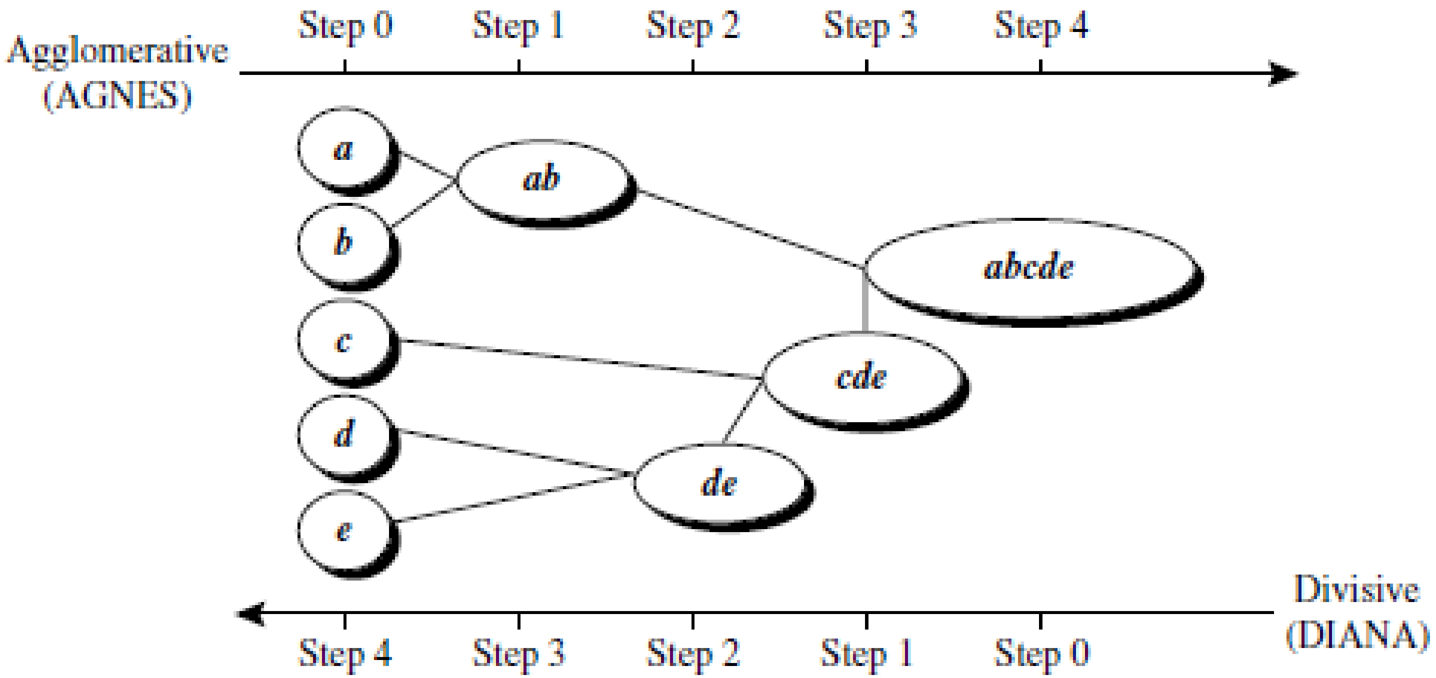
- An agglomerative clustering method uses a bottom-up strategy.
- It typically starts by letting each object form its own cluster and iteratively merges clusters into larger and larger clusters, until all the objects are in a single cluster or certain termination conditions are satisfied.
- The single cluster becomes the hierarchy’s root.
- For the merging step it finds two clusters that are closest to each other, and combines the two to form one cluster.
- An agglomerative clustering requires at most n iterations.

Hierarchical Methods: Divisive Clustering

- A divisive hierarchical clustering method employs a top-down strategy.
- It starts by placing all objects in one cluster, which is the hierarchy’s root.
- It then divides the root into several smaller sub-clusters, and recursively partitions those clusters into smaller ones.
- The partitioning process continues until each cluster at the lowest level is coherent enough – either containing only one object, or the objects within a cluster are sufficiently similar to each other.
- In either agglomerative or divisive hierarchical clustering, a user can specify the desired number of clusters as a termination condition.

Hierarchical Methods: Agglomerative vs. Divisive Clustering

- Agglomerative and Divisive Hierarchical Clustering on data objects {a, b, c, d, e}



Distance Measures

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

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

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

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

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

Density-Based Methods

- Partitioning and Hierarchical methods are designed to find spherical-shaped clusters. They have difficulty finding clusters of arbitrary shape.
- To find clusters of arbitrary shape, we can model clusters as dense regions in data space, separated by sparse regions. This is the main strategy behind density-based clustering methods.
- The density of an object **o** can be measured by the number of objects close to **o**.
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) finds *core* objects, that is, objects that have dense neighborhood.

Density-Based Methods: DBSCAN

- It connects core objects and their neighborhoods to form dense regions as clusters.
- A user specified parameter $\epsilon > 0$ is used to specify the radius of a neighborhood.
- The ϵ-neighborhood of an object **o** is the space within a radius ϵ centered at **o**.
- Due to fixed neighborhood size parameterized by ϵ, the density of a neighborhood can be measured simply by the number of objects in the neighborhood.
- To determine whether a neighborhood is dense or not, DBSCAN uses another user-specified parameter, *MinPts*, which specifies the density threshold of dense regions.

Density-Based Methods: DBSCAN (Contd.)

- An object is core object if the ϵ–neighborhood of the object contains at least *MinPts* objects.
- Given a set, D, of objects, we can identify all core objects with respect to the given parameter ϵ and *MinPts*.
- For a core object **q** and an object **p**, we say that **p** is directly density-reachable from **q** (with respect to ϵ and *MinPts*) if **p** is within the ϵ neighborhood of **q**.
- An object **p** is directly density-reachable from **q** if and only if **q** is a core object and **p** is in the ϵ-neighborhood of **q**.

Density-Based Methods: DBSCAN (Contd.)

- DBSCAN Algorithm

Method:

```
(1)  mark all objects as unvisited;
(2)  do
(3)       randomly select an unvisited object p;
(4)       mark p as visited;
(5)       if the ε-neighborhood of p has at least MinPts objects
(6)            create a new cluster C, and add p to C;
(7)            let N be the set of objects in the ε-neighborhood of p;
(8)            for each point p′ in N
(9)                 if p′ is unvisited
(10)                     mark p′ as visited;
(11)                     if the ε-neighborhood of p′ has at least MinPts points,
                         add those points to N;
(12)                if p′ is not yet a member of any cluster, add p′ to C;
(13)           end for
(14)           output C;
(15)      else mark p as noise;
(16) until no object is unvisited;
```

Evaluation of Clustering

- The major task of clustering evaluation include the following:

1. Assessing the clustering tendency
2. Determining the number of clusters in a data set
3. Measuring clustering quality

Evaluation of Clustering: Assessing Clustering Tendency

- Clustering tendency assessment determines whether a given data set has a non-random structure, which may lead to meaningful clusters.
- Consider a data set that does not have any non-random structure, such as a set of uniformly distributed points in a data space.
- Even though a clustering algorithm may return clusters for the data, those clusters are random and are not meaningful.
- The Hopkin Statistic is a spatial statistic that tests the spatial randomness of a variable as distributed in a space.

Evaluation of Clustering: Assessing Clustering Tendency (Contd.)

- 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.
- The Hopkin Statistic is calculated as follows:

1. Sample n points uniformly from D. For each point $\mathbf{p_i}$, we find the nearest neighbor of $\mathbf{p_i}$ in D. Let x_i be the distance between $\mathbf{p_i}$ and its nearest neighbor in D. That is,

$$x_i = \min_{v \in D}\{dist(p_i, v)\}$$

Evaluation of Clustering: Assessing Clustering Tendency (Contd.)

2. Sample n points uniformly from D For each $\mathbf{q_i}$, we find the nearest neighbor of q_i in D – {$\mathbf{q_i}$}, and let y_i be the distance between $\mathbf{q_i}$ and its nearest neighbor in D – {$\mathbf{q_i}$}. That is,

$$y_i = \min_{v \in D, v \neq q_i} \{dist(q_i, v)\}$$

3. Calculate the Hopkin 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

Evaluation of Clustering: Measuring Clustering Quality

- Ground Truth is the ideal clustering that can exist.
- If ground truth is available, it can be used extrinsic methods, which compare the clustering against the ground truth and measure.
- If ground truth is unavailable, we can use intrinsic methods, which evaluate the goodness of a clustering by considering how well the clusters are separated.
- Ground truth can be considered as supervision in the form of “cluster labels”.
- Hence, extrinsic methods are also known as *supervised methods*, while intrinsic methods are *unsupervised methods*.