e-PGPathshala
Subject : Computer Science
Paper: Data Analytics
Module No 26: CS/DA/26 -Clustering - I
Quadrant 1 – e-text

1.1 Introduction

Cluster analysis divides data into groups (clusters) for the purposes of summarization or improved understanding. Cluster analysis groups objects (observations, events) based on the information found in the data describing the objects or their relationships. The goal is that the objects in a group should be similar (or related) to one another and different from (or unrelated to) the objects in other groups. The greater is the similarity (or homogeneity) within a group and the greater the difference between groups, the better the clustering.

The definition of what constitutes a cluster is not well defined, and in many applications, clusters are not well separated from one another. Nonetheless, most cluster analysis seeks, as a result, a crisp classification of the data into non-overlapping groups. Cluster analysis is a classification of objects from the data, where by "classification" we mean a labeling of objects with class (group) labels. As such, clustering does not use previously assigned class labels, except perhaps for verification of how well the clustering worked. Thus, cluster analysis is sometimes referred to as "unsupervised classification" and is distinct from "supervised classification," or more commonly just "classification," which seeks to find rules for classifying objects given a set of pre-classified objects.

Classification is an important part of data mining, pattern recognition, machine learning, and statistics (discriminate analysis and decision analysis). While cluster analysis can be very useful, either directly or as a preliminary means of finding classes, there is more to data analysis than cluster analysis. For example, the decision of what features to use when representing objects is a key activity of fields such as data mining, statistics, and pattern recognition. Cluster analysis typically takes the features as given

and proceeds from there. Thus, cluster analysis, while a useful tool in many areas, is normally only part of a solution to a larger problem that typically involves other steps and techniques.

1.2 Learning Outcomes

- Understand the challenges with high dimensional data
- Know the challenges in clustering
- Learn Hierarchical Clustering Techniques

1.3 High Dimensional Data

Clustering in high-dimensional spaces is a difficult problem which is recurrent in many domains, for example in image analysis. The difficulty is due to the fact that high-dimensional data usually live in different low-dimensional subspaces hidden in the original space.For example, given a cloud of data points as shown in figure 1 and we want to understand its structure, it may be very difficult even for a very advance level tool. The only think which we can visually find is the organization of points in a two dimensional space.

Figure 1. Cloud of data points

1.4 Clustering High-Dimensional Data

Clustering high-dimensional data has many applications such as text documents, DNA micro-array data and so on. Major challenges are:

1. Many irrelevant dimensions may mask clusters and hence the cluster properties can be mined exactly.
2. Distance measure becomes meaningless, due to equi-distance.
3. Clusters may exist only in some subspaces and hence identification and representation suitable subspaces become very much essential.

1.4.1 Clustering Methods for High-Dimensional Data

Clustering of high dimensional data needs feature transformation, feature selection and subspace clustering. Feature transformation will be effective only if most dimensions are relevantand methods like PCA & SVD are useful only when features are highly correlated/redundant.

Feature selection is done using wrapper or filter approaches for finding a subspace where the data have nice clusters. Subspace-clustering is essential for finding clusters in all the possible subspaces. Methods like CLIQUE, ProClus, and frequent pattern-based clustering are common among them.

1.4.2 Challenges with High Dimensions

- Data in only one dimension is relatively packed
- Adding a dimension “stretch” the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measure becomes meaningless—due to equi-distance

1.4.3 The Problems of Clusteringsparsity in high dimensional data

Given a set of points, with a notion of distance between points, group the points into some number of clusters, so that Members of a cluster are close/similar to each other, Members of different clusters are dissimilar.Usually, for the Points in a high-dimensional space, Similarity is defined using a distance measure (Euclidean, Cosine, Jaccard, edit distance, etc).

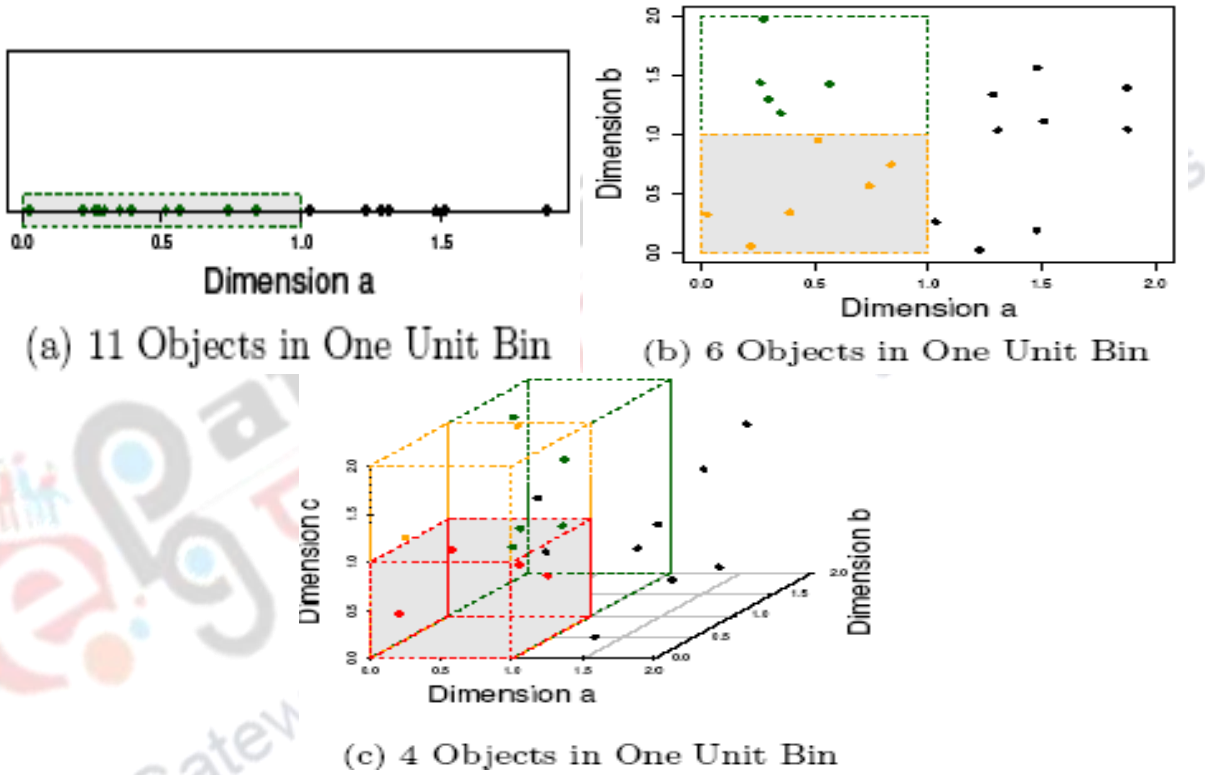


Figure 2 (a), (b), (c). Clusters of data visualized in 1, 2 and 3 Dimensions

The representation of data points in figure 2 (a), (b) and (c) are in 1, 2 and 3 dimensions respectively. From the illustration it can be understood that as the dimensions increases the data becomes more and more sparse.

Clustering is a hard problem: Clustering in two dimensions looks easy. Clustering small amounts of data looks easy and in most cases, looks are not deceiving. In many

applications involve not 2, but 10 or 10,000 dimensions. High-dimensional spaces look different. Almost all pairs of points are at about the same distance.The human eye can pretty easily separate these data into three groups, but the clustering algorithms fails pretty hard.

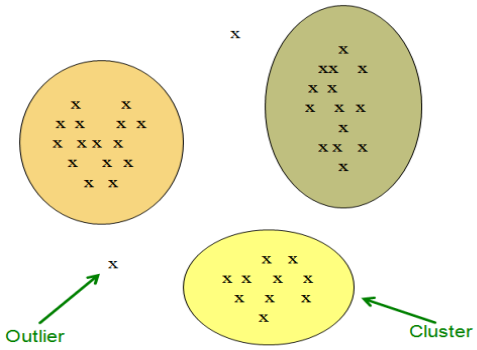


Figure 3. Sample Clusters & Outliers

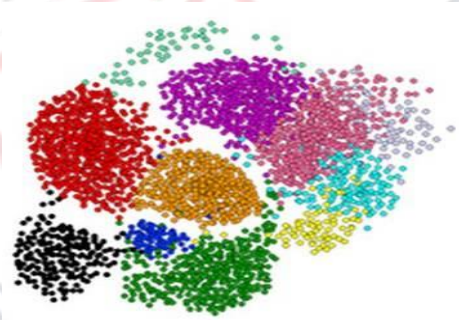


Figure 4. High Dimensional Cluster

Every clustering algorithm makes structural assumptions about the dataset that need to be considered. This can produce undesirable results when the clusters are elongated in certain directions — particularly when the between-cluster distance is smaller than the maximum within-cluster distance. Single-linkage clustering, in contrast, can perform well in these cases, since points are clustered together based on their nearest neighbor, which facilitates clustering along ‘paths’ in the dataset

1.5 Clustering Problem:

1.5.1 Galaxies: A catalog of 2 billion “sky objects” represents objects by their radiation in 7 dimensions (frequency bands). The problem is Cluster into similar objects, e.g., galaxies, nearby stars, quasars, etc.

Figure 4. Sloan Digital Sky Survey.

1.5.2 Music CDs: Intuitively, Music divides into categories, and customers prefer a few categories. Represent a CD by a set of customers who bought it and similar CDs have similar sets of customers, and vice-versa.Space of all CDs can be visualized as a space with one dimension for each customer. The values in a dimension may be 0 or 1 only and a CD might be considered as a point in this space (x_1, x_2,…,x_k), where x_i = 1 iff the i^{th} customer bought the CD. For Amazon, the dimension is tens of millions and the task isto find the clusters of similar CDs.

Distance metrics: As with CDs we have a choice when we think of documents as sets of words or shingles:

- Sets as vectors: Measure similarity by the cosine distance.
- Sets as sets: Measure similarity by the Jaccard distance.
- Sets as points: Measure similarity by Euclidean distance.

Distance metrics will be dealt in next module

1.5.3 Documents: Finding topics, it represent a document by a vector (x_1, x_2,…,x_k), where x_i = 1 iff the i^{th} word (in some order) appears in the document. It actually doesn't matter if k is infinite; i.e., we don't limit the set of words. Documents with similar sets of words may be about the same topic.

1.6 *Major Clustering Approaches*

The notion of a "cluster" cannot be precisely defined, which is one of the reasons why there are so many clustering algorithms. Different researchers employ different cluster models, and for each of these cluster models again different algorithms can be given. Understanding these "cluster models" is key to understanding the differences between the various algorithms.Clustering algorithms can be categorized based on their cluster modelinto the following categories:

- Partitioning Method
- Hierarchical Method
- Density-based Method
- Grid-Based Method
- Model-Based Method
- Constraint-based Method

1.6.1 Partitioning algorithms:

Construct various partitions and then evaluate them by some criterion. Suppose we are given a database of 'n' objects and the partitioning method constructs 'k' partition of data. Each partition will represent a cluster and k ≤ n. It means that it will classify the data into k groups, which satisfy the following requirements:

- Each group contains at least one object.
- Each object must belong to exactly one group.

1.6.2 Hierarchy algorithms:

Create a hierarchical decomposition of the set of data (or objects) using some criterion. We can classify hierarchical methods on the basis of how the hierarchical decomposition is formed. There are two approaches here −

- Agglomerative Approach
- Divisive Approach

Agglomerative Approach

This approach is also known as the bottom-up approach. In this, we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keep on doing so until all of the groups are merged into one or until the termination condition holds.

Divisive Approach

This approach is also known as the top-down approach. In this, we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds. This method is rigid, i.e., once a merging or splitting is done, it can never be undone.

1.6.3 Density-based:

Based on connectivity and density functions. The basic idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold, i.e., for each data point within a given cluster, the radius of a given cluster has to contain at least a minimum number of points.

1.6.4 Grid-based:

Based on a multiple-level granularity structure. In this, the objects together form a grid. The object space is quantized into finite number of cells that form a grid structure. The major advantage of this method is fast processing time. It is dependent only on the number of cells in each dimension in the quantized space.

1.6.5 Model-based:

A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other. This method locates the clusters by clustering the density function. It reflects spatial distribution of the data points.

This method also provides a way to automatically determine the number of clusters based on standard statistics, taking outlier or noise into account. It therefore yields robust clustering methods.

1.6.6 Constraint-based Method

In this method, the clustering is performed by the incorporation of user or application-oriented constraints. A constraint refers to the user expectation or the properties of desired clustering results. Constraints provide us with an interactive way of communication with the clustering process. Constraints can be specified by the user or the application requirement.

Among all these clustering methods, partitioning and hierarchical clustering are the most commonly and widely used clustering algorithm. In this module hierarchical based method is dealt in detail and the partitioning method will be explained in next module. Any text on data mining will narrate these algorithms as it is easy to understand even to the novice.

Approaches to Improve Quality of Hierarchical Clustering

Here are the two approaches that are used to improve the quality of hierarchical clustering

- Perform careful analysis of object linkages at each hierarchical partitioning.
- Integrate hierarchical agglomeration by first using a hierarchical agglomerative algorithm to group objects into micro-clusters, and then performing macro-clustering on the micro-clusters.

Point Assignment:

- Maintain a set of clusters

- Points belongs to “nearest” cluster

Key operation:

Repeatedly combine two nearest clusters

Three important questions:

- How do you represent a cluster of more than one point?
- How do you determine the “nearness” of clusters?
- When to stop combining clusters?

How to represent a cluster of many points?

- **Key problem:**

  As you merge clusters, how do you represent the “location” of each cluster, to tell which pair of clusters is closest?

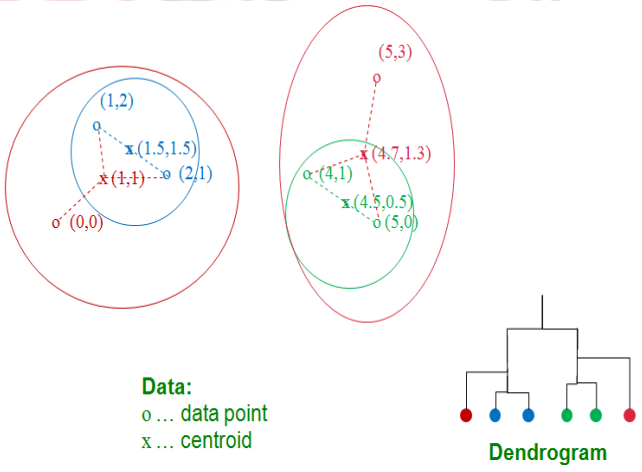
- **Euclidean case:**

  Each cluster has a ***centroid*** = average of its (data)points

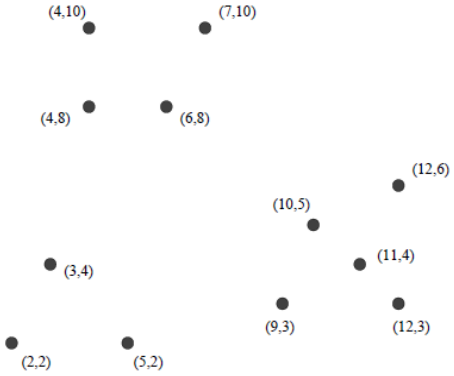
How to determine “nearness” of clusters?

- Measure cluster distances by distances of centroids

Example: Hierarchical clustering



Twelve points to be clustered hierarchically

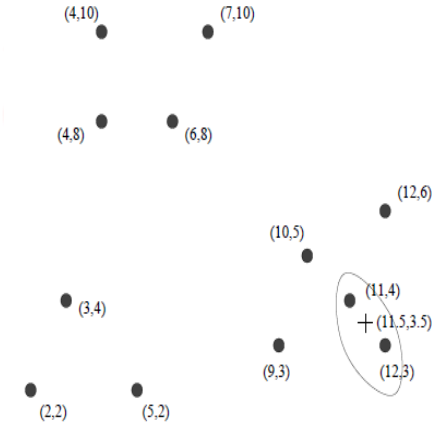


Among all the pairs of points, there are two pairs that are closest:

(10,5) and (11,4) or (11,4) and (12,3).

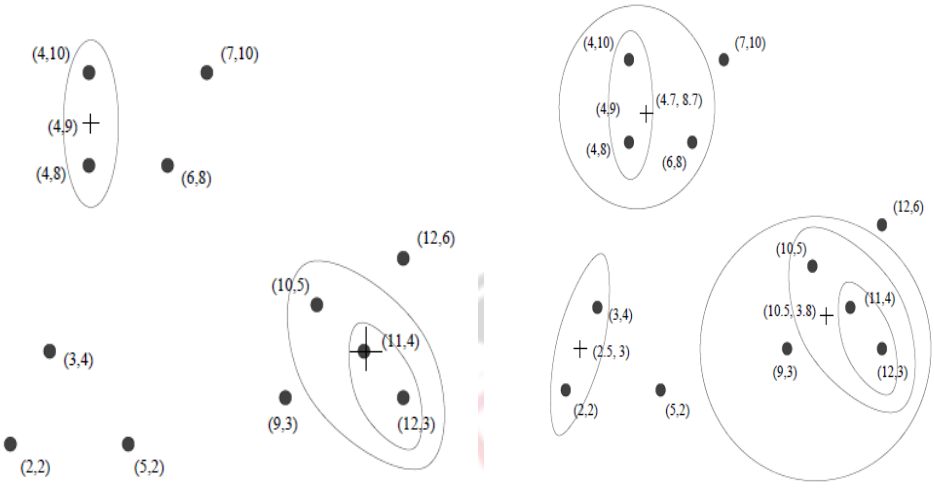
Each is at distance √2.

- Let us break ties arbitrarily
- decide to combine (11,4) with (12,3).



- The result is including the centroid of the new cluster, which is at (11.5, 3.5).
- Next we may think that (10,5) gets combined with the new cluster next, since it is so close to (11,4).

- But our distance rule requires us to compare only cluster centroids, and the distance from (10,5) to the centroid of the new cluster is 1.5√2, which is slightly greater than 2.
- Thus, now the two closest clusters are those of the points (4,8) and (4,10).
- We combine them into one cluster with centroid (4,9).



Summary:

Clustering:

- Given a set of points, with a notion of distance between points, group the points into some number of *clusters*
- *Different distance metrics discussed and the same will be explained in detail in next module.*
- *Classification of major clustering methods*

Algorithms:

- Agglomerative hierarchical clustering explained with example
- Partitioning K-means will be dealt in the up coming module