# 13 Introduction to Unsupervised Learning Algorithms

## LEARNING OBJECTIVES

- To understand the basics of clustering.
- To understand and appreciate the requirements of clustering.
- To introduce the concept of partitioning-based clustering ($k$-means and $k$-medoids).
- To learn how to create dendrograms using agglomerative-based clustering

## LEARNING OUTCOMES

- Students will be able to understand and appreciate clustering as an unsupervised learning method.
- Students will be able to solve numericals on partitioning-based clustering techniques.
- Students will be able to solve numericals on agglomerative-based clustering using single, complete, and average linkages.

## 13.1 Introduction to Clustering

Clustering is the process of grouping together data objects into multiple sets or clusters, so that objects within a cluster have high similarity as compared to objects outside of it. The similarity is assessed based on the attributes that describe the objects. Similarity is measured by distance metrics. The partitioning of clusters is not done by humans. It is done with the help of algorithms. These algorithms allow us to derive some useful information from the data which was previously unknown. Clustering is also called *data segmentation* because it partitions large datasets into groups according to their similarity.

Clustering can also be used for outlier detection. Outliers are objects which do not fall into any cluster because of too much dissimilarity with other objects. We can utilize them for special applications like credit card fraud detection. In credit card transactions, very expensive and infrequent purchases may be signs of fraudulent cases and we can apply one more level of security to avoid such transactions.

Clustering is known as unsupervised learning because the class label information is not present. You have already seen in supervised learning algorithm that every input has a corresponding output, which helps in designing a model. That is why supervised learning is called learning by example, while unsupervised learning is called learning by observation.

### 13.1.1 Applications of Clustering

Cluster analysis has been widely used in many applications such as business intelligence, pattern recognition, image processing, bioinformatics, web technology, search engines, and text mining.

1. **Business intelligence:** Cluster analysis helps in target marketing, where marketers discover groups and categorize them based on the purchasing patterns. The information retrieved can be used in market segmentation, product positioning (i.e., allocating products to specific areas), new product development, grouping of shopping items, and selecting test markets.
2. **Pattern recognition:** Here, the clustering methods group similar patterns into clusters whose members are more similar to each other. In other words, the similarity of members within a cluster is much higher when compared to the similarity of members outside of it. There is no prior knowledge of patterns of clusters or even how many clusters are appropriate.
3. **Image processing:** Extracting and understanding information from images is very important in image processing. The images are initially segmented and the different objects of interest in them are then identified. This involves division of an image into areas of similar attributes. It is one of the important and most challenging tasks in image processing where clustering can be applied. Image processing has applications in many areas such as analysis of remotely sensed images, traffic system monitoring, and fingerprint recognition.
4. **Bioinformatics:** This is a growing field in terms of research activities and is a part of biotechnology and genetic engineering. In this case, clustering techniques are required to derive plant and animal taxonomies, categorize genes with similar functionalities, and gain insight into structures inherent to populations. Biological systematics is another field which involves study of the diversification of living forms and the relationships among living things through time. The scientific classification of species can be done on the basis of similar characteristics using clustering. This field can give more information about both extinct and extant organisms.
5. **Web technology:** Clustering helps classifying documents on the web for information delivery.
6. **Search engines:** The success of Google as a search engine is because of its intensive searching capabilities. Whenever a query is fired by a user, the search engine provides the result for the searched data according to the nearest similar object which are clustered around the data to be searched. The speed and accuracy of the retrieved resultant is dependent on the use of the clustering algorithm. Better the clustering algorithm used, better are the chances of getting the required result first. Hence the definition of similar object plays a crucial role in getting better search results.
7. **Text mining:** Text mining involves the process of extracting high quality information from text. High quality in text mining means clustering in terms of relevance, novelty, and interestingness. It can be used for sentiment analysis and document summarization.

### 13.1.2 Requirements of Clustering

The requirements of clustering can be enumerated and explained as follows:

1. **Scalability:** A clustering algorithm is considered to be highly scalable if it gives similar results independent of the size of the database. Generally, clustering on a sample dataset may give different results compared to a larger dataset. Poor scalability of clustering algorithms leads to distributed clustering for partitioning large datasets. Some algorithms cluster large-scale datasets without considering the entire dataset at a time. Data can be randomly divided into equal-sized disjoint subsets and clustered using a standard algorithm. The centroids of subsets form an ensemble which can be solved by a centroid correspondence algorithm. The centroids are combined to form a global set of centroids.
2. **Dealing with different types of attributes:** Algorithms are designed to cluster numeric data. However, applications may require clustering other data types like nominal, binary, and ordinal. Nominal data is in alphabetical form and not in integer form. Binary attribute is of two types: symmetric binary and

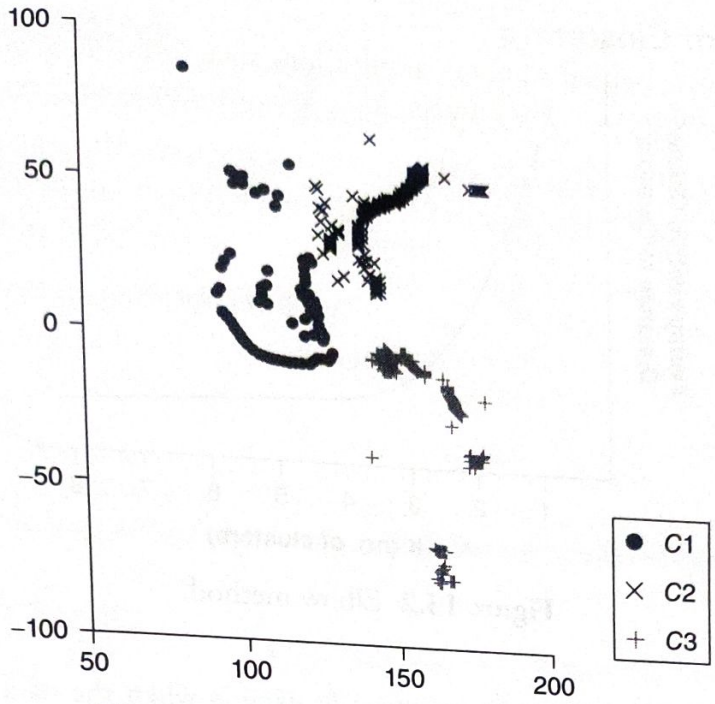


**Figure 13.1** Clusters of arbitrary shapes.

asymmetric binary. In symmetric data, both values are equally important. For example, in gender, it is male and female. In asymmetric data, both values are not equally important. For example, in result, it is pass and fail. The clustering algorithm should also work for complex data types such as graphs, sequences, images, and documents.

3. **Discovery of clusters with arbitrary shape:** Generally, clustering algorithms are to determine spherical clusters. Due to the characteristics and diverse nature of the data used, clusters may be of arbitrary shapes and can be nested within one another. For example, the cluster pattern for active and inactive volcanoes has chain-like patterns as shown in Fig. 13.1.

   Traditional clustering algorithms, such as *k*-means and *k*-medoids, fail to detect non-spherical shapes. Thus, it is important to have clustering algorithms that can detect clusters of any arbitrary shape.

4. **Avoiding domain knowledge to determine input parameters:** Many algorithms require domain knowledge like the desired number of clusters in the form of input. Thus, the clustering results may become sensitive to the input parameters. Such parameters are often hard to determine for high dimensionality data. Domain knowledge requirement affects the quality of clustering and burdens the user.

   For example, in *k*-means algorithm, the metric used to compare results for different values of *k* is the mean distance between data points and their cluster centroid. Increasing the number of clusters will always reduce the distance of data points to the extreme of reaching zero when *k* is the same as the number of data points. Thus, this cannot be used. Instead, to roughly determine *k*, the mean distance to the centroid is plotted as a function of *k* and the "elbow point", where the rate of decrease sharply shifts. This is shown in Fig. 13.2.

5. **Handling noisy data:** Real-world data, which is the input of clustering algorithms, are mostly affected by noise. This results in poor-quality clusters. Noise is an unavoidable problem, which affects the data collection and data preparation processes. Therefore, the algorithms we use should be able to deal with noise. There are two types of noise:
   - Attribute noise includes implicit errors introduced by measurement tools. They are induced by different types of sensors.

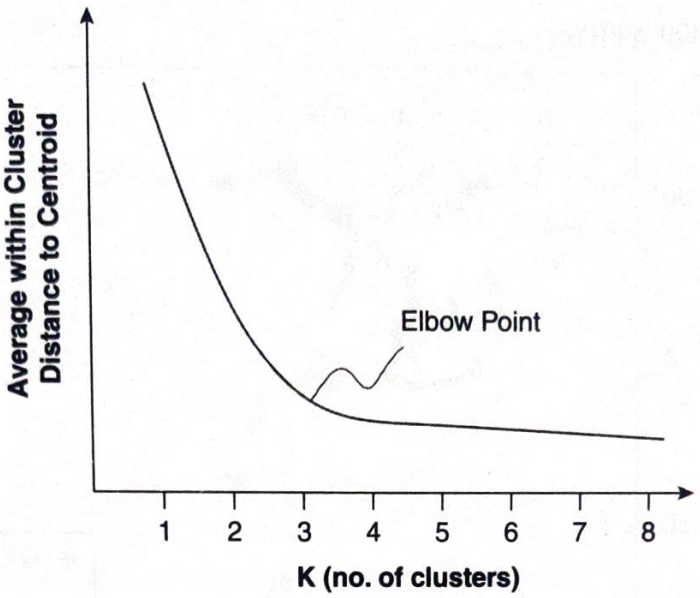


**Figure 13.2** Elbow method.

- Random errors introduced by batch processes or experts when the data is gathered. This can be induced in document digitalization process.

6. **Incremental clustering:** The database used for clustering needs to be updated by adding new data (incremental updates). Some clustering algorithms cannot incorporate incremental updates but have to recompute a new clustering from scratch. The algorithms which can accommodate new data without reconstructing the clusters are called *incremental clustering algorithms*. It is more effective to use incremental clustering algorithms.
7. **Insensitivity to input order:** Some clustering algorithms are sensitive to the order in which data objects are entered. Such algorithms are not ideal as we have little idea about the data objects presented. Clustering algorithms should be insensitive to the input order of data objects.
8. **Handling high-dimensional data:** A dataset can contain numerous dimensions or attributes. Generally, clustering algorithms are good at handling low-dimensional data such as datasets involving only two or three dimensions. Clustering algorithms which can handle high-dimensional space are more effective.
9. **Handling constraints:** Constrained clustering can be considered to contain a set of must-link constraints, cannot-link constraints, or both. In a must-link constraint, two instances in the must-link relation should be included in the same cluster. On the other hand, a cannot-link constraint specifies that the two instances cannot be in the same cluster. These sets of constraints act as guidelines to cluster the entire dataset. Some constrained clustering algorithms cancel the clustering process if they cannot form clusters which satisfy the specified constraints. Others try to minimize the amount of constraint violation if it is impossible to find a clustering which satisfies the constraints. Constraints can be used to select a clustering model to follow among different clustering methods. A challenging task is to find data groups with good clustering behavior that satisfy specified constraints.
10. **Interpretability and usability:** Users require the clustering results to be interpretable, usable, and include all the elements. Clustering is always tied with specific semantic interpretations and applications. The applications should be able to use the information retrieved after clustering in a useful manner.

## 13.2 Types of Clustering

Clustering algorithms can be classified into two main subgroups:

1. **Hard clustering:** Each data point either belongs to a cluster completely or not.
2. **Soft clustering:** Instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned.

Clustering algorithms can also be classified as follows:

1. Partitioning method.
2. Hierarchical method.
3. Density-based method.
4. Grid-based method.

However, the focus in this chapter is on partitioning method and hierarchical-based methods.

### 13.2.1 Partitioning Method

Partitioning means division. Suppose we are given a database of $n$ objects and we need to partition this data into $k$ partitions of data. Within a partition there exists some similarity among the items. So each partition will represent a cluster and $k \leq n$. It means that it will classify the data into $k$ groups, each group contains at least one object and each object must belong to exactly one group. Although this is the general requirement, in soft clustering an object can belong to two clusters also. Most partitioning methods are distance-based.

For a given number of partitions (say $k$), the partitioning method will create an initial partitioning. Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to another. The general criterion of a good partitioning is that objects in the same cluster are close to each other, whereas objects in different clusters are far from each other. Some other criteria can be used for judging the quality of partitions.

Partition-based clustering is often computationally expensive and hence most of the methods apply heuristic methods, including the greedy approach which improves the quality of cluster arriving at a local optimum.

These heuristic clustering methods result in spherical clusters. For complex-shaped clusters and for large datasets, some extensions are required for partition-based methods.

### 13.2.2 Hierarchical Method

Hierarchical clustering is an alternative approach to partitioning clustering for identifying groups in a dataset. It does not require prespecifying the number of clusters to be generated. The result of hierarchical clustering is a tree-based representation of the objects, which is also known as *dendrogram*. Observations can be subdivided into groups by cutting the dendrogram at a desired similarity level. We classify hierarchical methods on the basis of how the hierarchical decomposition is formed. There are two approaches:

1. **Agglomerative approach:** This approach is also known as the *bottom-up approach*. In this approach, we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keeps on doing so until all of the groups are merged into one or until the termination condition holds.
2. **Divisive approach:** This approach is also known as the *top-down approach*. In this approach, 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 done until each object is in one cluster or the termination condition holds. This method is rigid, that is, once a merging or splitting is done, it can never be undone.

### 13.2.3 Density-Based Methods

Density-based clustering algorithm finds nonlinear shapes clusters based on the density. Density-based spatial clustering of applications with noise (DBSCAN) is the most widely used density-based algorithm. It uses the concept of density reachability and density connectivity.

1. **Density reachability:** A point "p" is said to be density reachable from a point "q" if it is within $\varepsilon$ distance from point "q" and "q" has sufficient number of points in its neighbors that are within distance $\varepsilon$.
2. **Density connectivity:** Points "p" and "q" are said to be density-connected if there exists a point "r" which has sufficient number of points in its neighbors and both the points are within $\varepsilon$ distance. This is called *chaining process*. So, if "q" is neighbor of "r", "r" is neighbor of "s", "s" is neighbor of "t", and "t" is neighbor of "p"; this implies that "q" is neighbor of "p".

### 13.2.4 Grid-Based Methods

The grid-based clustering approach differs from the conventional clustering algorithms in that it is concerned not with data points but with the value space that surrounds the data points. In general, a typical grid-based clustering algorithm consists of the following five basic steps:

1. Create the grid structure, i.e., partitioning the data space into a finite number of cells.
2. Calculating the cell density for each cell.
3. Sorting the cells according to their densities.
4. Identifying cluster centers.
5. Traversal of neighbor cells.

## 13.3 Partitioning Methods of Clustering

The most fundamental clustering method is the partitioning method. This method assumes that we already know the number of clusters to be formed, which organizes the objects of a set into several exclusive groups or clusters. If $k$ is the number of clusters to be formed given a dataset $D$ of $n$ objects, the partitioning algorithm organizes the objects into $k$ partitions ($k \leq n$), where each partition represents a cluster. The objective function in this type of partitioning is that the similarity among the data items within a cluster is higher than the elements in a different cluster. In other words, inter-cluster similarity is higher than intra-cluster similarity.

### 13.3.1 k-Means Algorithm

The most well-known clustering algorithm is probably $k$-means. It is taught in a lot of introductory data science and machine learning classes. It is easy to understand and implement.

The main concept is to define $k$ cluster centers. The cluster centers should be kept in such a way that it covers the data points of the entire dataset. The best way to do so is to keep data points as far away from each other as possible. We can then associate each data point to the nearest cluster center. The initial grouping of data is completed when there is no data point remaining. Once the grouping is done, new centroids are computed. These again form clusters based on the new cluster centers. The process is repeated till no more changes are done, which implies the cluter centers do not change any more. This algorithm aims at minimizing an objective function know as squared error function, which is given by

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left( \| x_i - v_j \| \right)^2 \tag{13.1}$$

where

$\| x_i - v_j \|$ is the Euclidean distance between $x_i$ and $v_j$.

$c_i$ is the number of data points in $i$th cluster.
$c$ is the number of cluster centers.

The objective function aims for high intra-cluster similarity and low inter-cluster similarity. This function tries to make the resulting $k$ clusters as compact and as separate as possible. Optimizing the within-cluster variation is computationally challenging since the problem is NP-hard in general even for two clusters (that is, $k = 2$). To overcome the prohibitive computational cost for the exact solution, greedy approaches are often used in practice.

#### 13.3.1.1 Steps in *k*-Means Clustering Algorithm

Let us study the steps in $k$-means clustering algorithm with the following example. Let $X = \{x_1, x_2, x_3, \ldots x_n\}$ be the set of data points and $V = \{v_1, v_2, \ldots v_c\}$ be the set of centers. The steps are:

1. Randomly select $c$ cluster centers.
2. Calculate the distance between each data point and cluster centers.
3. Assign the data point to the cluster having a minimum distance from it and the cluster center.
4. Recalculate the new cluster center using Eq. (13.2).

$$v_i = \left(\frac{1}{c_i}\right)\sum_{j=1}^{c_i} x_i \tag{13.2}$$

where $c_i$ represents the number of data points in the $i$th cluster.

5. Recalculate the distance between each data point and the newly obtained cluster centers.
6. If no data point was reassigned then stop, otherwise repeat steps 3 to 5.

The advantages of $k$-means clustering algorithm are:

1. Fast, robust, and easier to understand.
2. Relatively efficient: The computational complexity of the algorithm is O($tknd$), where $n$ is the number of data objects, $k$ is the number of clusters, $d$ is the number of attributes in each data object, and $t$ is the number of iterations. Normally, $k$, $t$, $d << n$. That is, the number of clusters attributes and iterations will be very small compared to the number of data objects in the dataset.
3. Gives best result when dataset are distinct or well separated from each other.

The disadvantages of $k$-means clustering algorithm are:

1. It requires prior specification of number of clusters.
2. It is not be able to cluster highly overlapping data.
3. It is variant to nonlinear transformations, that is, with different representation of data we get different results (data represented in form of Cartesian coordinates and polar coordinates will give different results).
4. It provides the local optima of the squared error function.
5. Random choosing of the cluster center cannot lead to fruitful result.
6. Applicable only when the mean is defined, that is, fails for categorical data.
7. Unable to handle noisy data and outliers.