

## Module 22: Unsupervised Learning: Part One: Clustering Techniques

### Learning outcomes

1. Analyse real-life applications and limitations of the clustering techniques.
2. Apply similarity and distance measures to compare clusters and justify clustering decisions.
3. Implement clustering techniques to generate outputs and explain the groupings.

### Understanding clustering

- **Purpose:** clustering is an unsupervised learning method that organises unlabelled data into meaningful groups based on similarity or distance.
- **Key idea:** items in the same cluster are more similar to each other than to items in other clusters.
- **Applications:** customer segmentation, anomaly detection, document grouping, gene expression analysis and more.

### Measuring similarity and distance

- **Distance metrics** define how similar (or dissimilar) data points are.
  - *Euclidean distance*: straight-line distance in feature space.
  - *Centroid distance*: distance between cluster centres.
  - *Single linkage (minimum distance)*: closest pair between clusters.
  - *Complete linkage (maximum distance)*: farthest pair between clusters.
  - *Average linkage*: mean of all pairwise distances.
- **Impact:** the chosen measure influences cluster formation and interpretation

### Hierarchical clustering

- **Concept:** builds a hierarchy of clusters – either bottom-up (*agglomerative*) or top-down (*divisive*).
- **Visualisation:** a *dendrogram* shows how clusters merge step by step – the height of joins reflects dissimilarity.
- **Advantages:**
  - No need to pre-specify the number of clusters.
  - Highly interpretable through dendograms.
- **Limitations:**
  - Computationally intensive on large data sets.
  - Myopic (greedy) nature – early merges can't be undone.
  - Sensitive to noise and outliers.
  - Difficult to select the optimal number of clusters.
- **Best used for:** small, clean data sets for which interpretability matters.

## ***k*-means clustering**

- **Concept:** divides data into  $k$  clusters by iteratively assigning points to the nearest centroid and updating centroids until convergence.
- **Key steps:**
  1. Choose  $k$  clusters.
  2. Initialise centroids (e.g. random or *k-means++*).
  3. Assign each point to the closest centroid.
  4. Recompute centroids as cluster means.
  5. Repeat until the assignments stabilise.
- **Elbow method:** plot the sum of squared distances vs the number of clusters – the ‘elbow’ indicates the optimal  $k$ .
- **Advantages:** fast, scalable and effective for spherical clusters.
- **Limitations:** requires  $k$  in advance; sensitive to initialisation and outliers.

## Implementing clustering in Python

- **Hierarchical clustering:**
  - `scipy.cluster.hierarchy` → `linkage()`, `dendrogram()`
  - `sklearn.cluster` → `AgglomerativeClustering()`
- ***k*-means clustering:**
  - `sklearn.cluster.KMeans()`
  - Use normalisation (`MinMaxScaler` or `normalize()`) to scale features before clustering.
- **Visualisation:** Use `matplotlib` or three-dimensional plots to interpret cluster separations and centroids.

## Real-world applications

- **Alibaba Group case study:** used  $k$ -means to optimise locations of self-pickup centres by grouping customers based on proximity.
- **Business and research contexts**
  - Customer segmentation for targeted marketing.
  - Network intrusion detection.
  - Text clustering for legal or product reviews.
  - Gene expression analysis in bioinformatics.

**Key takeaway:** clustering supports decision-making when natural groupings exist but explicit labels do not.

## Practical considerations

- Normalise or standardise data before clustering.
- Remove outliers that distort distance calculations.
- Combine clustering with dimensionality reduction (e.g. PCA) for high-dimensional data.

- Interpret results carefully – cluster boundaries are not always absolute.

## Summary

Having completed this module, you should now be able to:

- Apply similarity and distance measures to compare clusters.
- Implement hierarchical and k-means clustering in Python.
- Interpret dendograms and elbow plots to select cluster structures.
- Analyse real-world applications and limitations of clustering methods.