Clustering

Unit 4: Clustering
Advanced Data Mining

Rupak Raj Ghimire

School of Mathematical Sciences
Institute of Science and Technology (IoST), TU

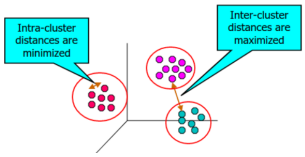
Feb, 2025

Table of contents

What is Cluster Analysis?

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



Cluster Analysis

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

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

Applications of Cluster Analysis

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

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**.
- **Solution:** Highly scalable clustering algorithms are needed

Challenges of Clustering(cont.)

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(cont.)

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(cont.)

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(cont.)

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(cont.)

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(cont.)

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(cont.)

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

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 $\times$ p variables):

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i1} & x_{i2} & \cdots & x_{ip} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

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

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

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.

d(i, j) = d(j, i), and d(i, i) = 0

Dissimilarity Measure

Distance metrics **quantify** the similarity or dissimilarity between data points.

- Used in machine learning, clustering, nearest neighbor search, and other applications.
- Choosing the right distance metric depends on the data characteristics and problem requirements.

Some methods

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

Euclidean Distance

Measures the straight-line distance between two points in an n-dimensional space.

Formula

$$d_E(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Commonly used in clustering (e.g., k-means), classification (e.g., k-NN), and geometry.

Manhattan (City Block) Distance

Measures distance along axes at right angles, like city blocks.

Formula

$$d_M(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

Useful when movement is restricted to grid-like paths (e.g., robotics, urban planning).

Minkowski Distance

Generalization of both Euclidean distance and Manhattan distance.

Formula

$$d_M(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

where, p is a positive integer. Such a distance is also called L_p norm, in some literature

Special cases

- $p = 1 \rightarrow$ Manhattan Distance (L_1 Norm)
- $p = 2 \rightarrow$ Euclidean Distance (L_2 Norm)

Provides flexibility in measuring distance based on parameter p.

Weighted Euclidean Distance

Assigns different **importance** to each feature dimension.

Formula

$$d_W(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2}$$

Used when certain features have more significance than others (e.g., medical diagnosis, image processing)
Weighting can also be applied to the Manhattan and Minkowski distances

When to Use Which Distance?

- **Euclidean Distance** when features are equally important and the space is continuous.
- **Manhattan Distance** when movement is constrained (e.g., grid-based systems).
- **Minkowski Distance** when needing flexibility in distance measurement.
- **Weighted Euclidean Distance** when certain features matter more than others.

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.

Data Standardization

Data standardization is the process of transforming data to have a common scale, ensuring that **each feature contributes equally** to the clustering process.

It is crucial in clustering because algorithms like K-Means use distance-based similarity measures, which are **sensitive to feature magnitudes**.

Idea

Without standardization, features with **large numerical values dominate** the clustering process.

Standardization ensures fair influence by transforming the data into a comparable scale.

Z-Score Standardization

$$X' = \frac{X - \mu}{\sigma}$$

Example: For *Ages* = [25, 50, 40]

- Mean: $\mu = 38.33$
- Std Dev: $\sigma = 12.58$
- Standardized Age 25: $X' = \frac{25-38.33}{12.58} = -1.06$

Interpretation: Now, all feature values have mean = 0 and variance = 1, making them comparable.

Min-Max Scaling

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Example: For salaries [30,000, 90,000, 60,000]

- $X_{\min} = 30,000$, $X_{\max} = 90,000$
- Normalized 60,000: $X' = \frac{60,000-30,000}{90,000-30,000} = 0.5$

Robust Scaling (IQR)

$$X' = \frac{X - \text{Median}}{\text{IQR}}$$

- Uses median and interquartile range (IQR = Q3 - Q1).
- Resistant to outliers.

Unit Vector Scaling

Data is transformed into a unit vector, making it useful for text or high-dimensional clustering

$$X' = \frac{X}{||X||}, \quad ||X|| = \sqrt{X_1^2 + X_2^2 + ... + X_n^2}$$

- Ensures each feature vector has a unit length.
- Useful for cosine similarity-based clustering.

Example

For Age = 25, Salary = 30,000, calculate the norm:
$||X|| = \sqrt{(25)^2 + (30000)^2} = 30000.001$
Now, normalize:
$X'_{Age} = \frac{25}{30000.001} 0.00083$ $X'_{Salary} = \frac{30000}{30000.001} 0.99999$

Log Transformation

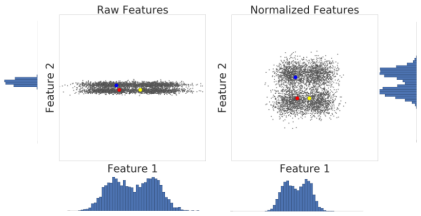
$$X' = \log(X + 1)$$

- Reduces skewness in data.
- Useful for data with large magnitude differences.

Suggested Reading

Book: Data Mining- Concepts and Techniques
Section **7.2 Types of Data in Cluster Analysis**

Clustering: Before and After data standardization



When to Use Which Scaling Method?

Method	When to Use?	Effect
Z-Score	When data follows a normal distribution	Mean = 0, Variance = 1
Min-Max	When we need data in a specific range [0,1]	Retains original feature distribution
Robust Scaling	When data has outliers (e.g., incomes)	Uses median instead of mean

Use Cases

- K-Means, PCA : Z-score Standardization
- Neural Networks : Min-Max Scaling
- Outlier-sensitive data : Robust Scaling (Median-based)

Clustering Methods

Many clustering algorithms exist in the literature.

It is difficult to provide a crisp categorization 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 $k \leq n$.

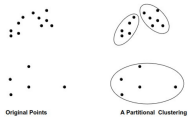
It **classifies** the data into **k groups**, which together satisfy the following requirements:

- Each group must contain at least one object, and
- Each object must belong to exactly one group

Partitioning Methods(cont.)

Given k, the number of partitions to construct, a partitioning method creates an i**nitial partition**ing.

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.

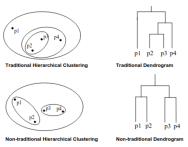
A hierarchical method can be classified as being either **agglomerative** or **divisive**, based on how the hierarchical decomposition is formed.

The **agglomerative** approach, also called the **bottom-up** approach.

Hierarchical methods(cont.)

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.



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 data points) 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.

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 low but the inter-cluster similarity is high**.

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.

Algorithm 1 K-Means Clustering

Input: Dataset $X = \{x_1, x_2, \ldots, x_n\}$, number of clusters k
Output: Cluster centers $C = \{c_1, c_2, \ldots, c_k\}$ and cluster assignments
Initialize k cluster centroids $C = \{c_1, c_2, \ldots, c_k\}$ randomly from X
repeat
  Assign each point to the nearest centroid
  for each data point $x_i \in X$ **do**
    Compute the distance to each centroid: $d(x_i, c_j)$
    Assign x_i to the closest cluster c_j
  end for
  Update centroids
  for each cluster C_j **do**
    Compute new centroid as the mean of all assigned points: $c_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$
  end for
until Centroids do not change (or a maximum number of iterations is reached)
Return final cluster centroids and assignments

K-Means Method(cont.)

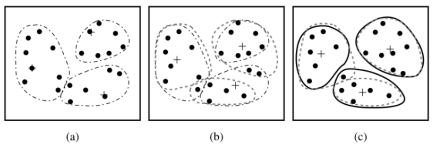


Figure: Building cluster : (a) →(b)→(c)

The k-Means Method(cont.)

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

Hierarchical Clustering

Hierarchical clustering organizes data into a hierarchy of clusters, represented as a **tree-like structure** known as a **dendrogram**

This algorithm **builds a hierarchy** of clusters by **iteratively merging** or **splitting clusters** based on their similarity

Strength

Hierarchical clustering can discover clusters of arbitrary shapes and sizes, and it provides a visual representation of the hierarchical relationships between clusters

Weakness

Hierarchical clustering can be computationally expensive, especially for large datasets. It is also sensitive to the initial ordering of the data points and the choice of the distance metric

Hierarchical Clustering(cont.)

There are two main types of hierarchical clustering

Agglomerative Clustering

This is **bottom-up strategy** starts by **placing each object in its own cluster** and then **merges** these atomic clusters into **larger and larger clusters**, until all of the objects are in a single cluster or until certain termination conditions are satisfied.

Divisive Clustering

This is **top-down strategy** does the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster. It **subdivides the cluster into smaller** and smaller pieces, until each object forms a cluster on its own or until it satisfies certain termination conditions, such as a desired number of clusters is obtained or the diameter of each cluster is within a certain threshold.

Agglomerative Vs. Divisive hierarchical clustering

- AGNES (AGglomerative NESting) an agglomerative hierarchical clustering method
- DIANA (DIvisive ANAlysis), a divisive hierarchical clustering method

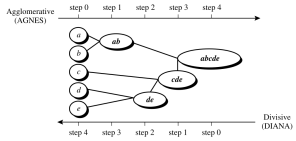


Figure: Agglomerative and divisive hierarchical clustering on data objects a, b, c, d, e

Dendrogram

A tree structure called a **dendrogram** is commonly used to represent the process of hierarchical clustering.

It shows how objects are grouped together step by step

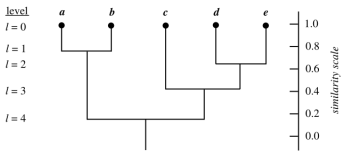


Figure: Dendrogram representation for hierarchical clustering of data objects a, b, c, d, e

Hierarchical Clustering - Types

Single link

When an algorithm uses the *minimum distance*, $d_{min}(C_i, C_j)$, to measure the distance between clusters, it is sometimes called a **nearest-neighbor clustering algorithm**. Moreover, if the clustering process is terminated when the distance between nearest clusters exceeds an arbitrary threshold, it is called a **single-linkage algorithm**.

Complete link

When an algorithm uses the **maximum distance**, $d_{max}(C_i, C_j)$, to measure the distance between clusters, it is sometimes called a **farthest-neighbor clustering algorithm**. If the clustering process is terminated when the maximum distance between nearest clusters exceeds an arbitrary threshold, it is called a **complete-linkage algorithm**.

Hierarchical Agglomerative Clustering(HAC)

Use the distance matrix in table below to perform single link and complete link hierarchical clustering. Show your results by drawing a dendogram

Dataset

p1(0.40. 0.53), p2(0.22, 0.38), p3(0.35,0.32), p4(0.26,0.19), p5(0.08,0.41), p6(0.45,0.30)

Prepare the distance matrix

$$\begin{bmatrix} 0.00 & & & & & \\ 0.23 & 0.00 & & & & \\ 0.22 & 0.14 & 0.00 & & & \\ 0.37 & 0.19 & 0.16 & 0.00 & & \\ 0.34 & 0.14 & 0.28 & 0.28 & 0.00 & \\ 0.24 & 0.24 & 0.11 & 0.22 & 0.39 & 0.00 \end{bmatrix}$$

Identify the min-value element

The minimum value element is $(p_3, p_6)$and value is 0.10

	p_1	p_2	p_3	p_4	p_5	p_6
p_1	0.00					
p_2	0.23	0.00				
p_3	0.22	0.14	0.00			
p_4	0.37	0.19	0.16	0.00		
p_5	0.34	0.14	0.28	0.28	0.00	
p_6	0.24	0.24	0.10	0.22	0.39	0.00

Recalculate or update the distance matrix for cluster

$MIN[dist((p_3, p_6), p_i)]$

Example : p_1

$dist[(p_3, p_6), p_1] = min(dist(p_3, p_1), dist(p_6, p_1))$

$= \min(0.22, 0.23)$

$= 0.22$

	p_1	p_2	$p_{3,6}$	p_4	p_5
p_1	0.00				
p_2	0.23	0.00			
$p_{3,6}$	0.22	0.14	0.00		
p_4	0.37	0.19	0.16	0.00	
p_5	0.34	0.14	0.28	0.28	0.00

Repeat to get another pair

The minimum value element is (p_2, p_5) and value is 0.14

Repeat the process to get the order of the cluster

- 1st cluster (p_3, p_6)
- 2nd cluster (p_2, p_5)
- 3rd cluster (p_2, p_5, p_3, p_6)
- 4th cluster $(p_2, p_5, p_3, p_6, p_4)$
- 5th cluster $(p_2, p_5, p_3, p_6, p_4, p_1)$

for MAX version of HAC

update the distance with $dist[(p_1, p_2), (p_3)] = max[dist(p_1, p_3), dist(p_2, p_3)]$

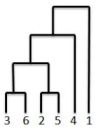


Figure: min version of HAC

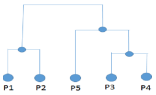


Figure: max version of HAC

Hierarchical Clustering(cont.)

Read the section 7.5 of book.

1. BIRCH: Balanced Iterative Reducing and Clustering Using Hierarchies
2. ROCK(RObust Clustering using linKs): A Hierarchical Clustering Algorithm forCategorical Attributes
3. Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling

Density-based clustering

To discover clusters with **arbitrary shape**, density-based clustering methods have been developed. These typically regard clusters as **dense regions of objects** in the data space that are separated by regions of low density (representing noise).

- DBSCAN: A Density-Based Clustering Method Based on Connected Regions with Sufficiently High Density
- OPTICS: Ordering Points to Identify the Clustering Structure
- DENCLUE: Clustering Based on Density Distribution Functions

DBSCAN

The algorithm grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databases with noise.

It defines a cluster as a maximal set of density-connected points.

- The neighborhood within a radius ϵ of a given object is called the ϵ-**neighborhood** of the object.
- A point is a **core point** if it has more than a specified number of points (MinPts) within ϵ
- A **border point** has fewer than MinPts within , but is in the neighborhood of a core point
- A **noise point** is any point that is not a core point or a border point.
- Every object not contained in any cluster is considered to be noise.

Density-reachability and density connectivity

Given ϵ = Circle radious
MinPts = 3

- Of the labeled points, **m**, **p**, **o**, and **r** are core objects because each is in an ϵ-neighborhood containing at least three points.
- **q** is directly density-reachable from **m**. **m** is directly density-reachable from **p** and vice versa.
- **q** is (indirectly) density-reachable from **p** because **q** is directly density-reachable from **m** and **m** is directly density-reachable from **p**. However, **p** is not density-reachable from **q** because **q** is not a core object. Similarly, **r** and **s** are density-reachable from **o**, and **o** is density-reachable from **r**.
- **o**, **r**, and **s** are all density-connected.

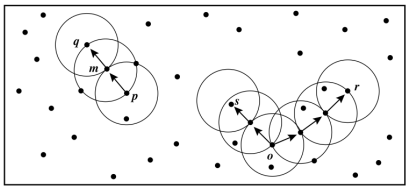


Figure: Density reachability and density connectivity in density-based clustering

Limitation of DBSCAN

DBSCAN can cluster objects given input parameters such as ϵ and *MinPts*, it still leaves the user with the **responsibility of selecting parameter** values that will lead to the discovery of acceptable clusters.

Actually, this is a problem associated with many other clustering algorithms. Such parameter settings are usually empirically set and difficult to determine, especially for real-world, high-dimensional data sets.

Most algorithms are very **sensitive** to such parameter values: **slightly different settings may lead to very different cluster**ings of the data.

Moreover, high-dimensional real data sets often have very **skewed distribution**s, such that their intrinsic clustering structure may not be characterized by global density parameters

OPTICS

Ordering Points to Identify the Clustering Structure
Rather than producing a data set clustering explicitly, **OPTICS** computes an augmented cluster ordering for automatic and interactive cluster analysis.

This ordering represents the density-based clustering structure of the data. It contains information that is equivalent to density-based clustering obtained from a **wide range of parameter settings**.

OPTICS(cont.)

The cluster ordering can be used to extract basic clustering information (such as cluster centers or arbitrary-shaped clusters) as well as provide the intrinsic clustering structure.

Therefore, in order to produce a set or ordering of **density-based** clusters, we can extend the DBSCAN algorithm to process a set of distance parameter values at the same time.

To construct the different clusterings simultaneously, the objects should be processed in a specific order. This order selects an object that is **density-reachable** with respect to the lowest ϵ-value so that clusters with higher density (lower ϵ) will be finished first.

Based on this idea, two values need to be stored for each object:
core-distance, reachability-distance

OPTICS(cont.)

Suppose that $\epsilon = 6$ mm and MinPts $= 5$.

- The core-distance of p is the distance, ϵ' , between p and the fourth closest data object.
- The reachability-distance of q_1 with respect to p is the core-distance of p (i.e., $\epsilon' = 3$ mm) because this is greater than the Euclidean distance from p to q_1 .
- The reachability-distance of q_2 with respect to p is the Euclidean distance from p to q_2 because this is greater than the core-distance of p

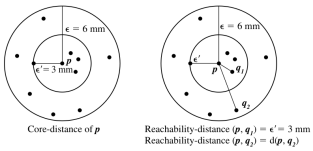


Figure: OPTICS terminology

OPTICS(cont.)

- The OPTICS algorithm **creates an ordering of the objects in a database**, additionally storing the core-distance and a suitable reachability-distance for each object.
- An algorithm was proposed to extract clusters based on the ordering information produced by OPTICS.
- Such information is sufficient for the extraction of all density-based clusterings with respect to any distance ϵ' that is smaller than the distance ϵ used in generating the order.

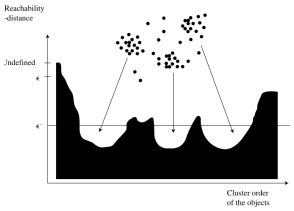


Figure: Cluster ordering in OPTICS

Grid-Based Methods

The grid-based clustering approach uses a **multiresolution grid data structure**.

It **quantizes** the object space into a **finite number of cells** that form a grid structure on which all of the operations for clustering are performed.

The main advantage of the approach is its **fast processing time**, which is typically **independent of the number of data objects**,yet **dependent on only the number of cells** in each dimension in the quantized space.

Grid-Based Methods(cont.)

STING

explores statistical information stored in the grid cells

WaveCluster

clusters objects using a **wavelet transform** method

CLIQUE

represents a **grid and density-based approach** for clustering in high-dimensional data space

STING: STatistical INformation Grid

STING is a grid-based multiresolution clustering technique in which the spatial area is divided into rectangular cells.

There are usually several levels of such rectangular cells corresponding to different levels of resolution, and **these cells form a hierarchical structure**: each cell at a high level is partitioned to form a number of cells at the next lower level.

STING: STatistical INformation Grid

Statistical parameters of higher-level cells can easily be computed from the parameters of the lower-level cells.
These parameters include the following:

- the attribute-independent parameter: count
- the attribute-dependent parameters: mean, stdev (standard deviation), min (minimum), max (maximum)
- the type of distribution that the attribute value in the cell follows: normal, uniform, exponential, or none (if the distribution is unknown)

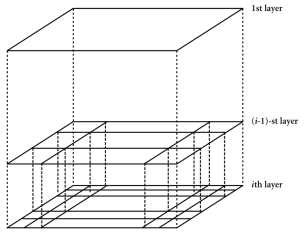


Figure: A hierarchical structure for STING clustering

Model-Based Clustering Methods

Model-based clustering methods attempt to optimize the **fit between the given data and some mathematical model**.

Such methods are often based on the assumption that the data are generated by a mixture of underlying probability distributions

- Expectation-Maximization
- Neural network approach

Expectation-Maximization

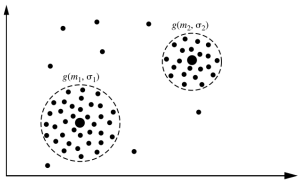
EM starts with an **initial estimate or "guess"** of the parameters of the mixture model (collectively referred to as the parameter vector).

It iteratively **rescores** the objects against the mixture density produced by the parameter vector. The rescored objects are then used to update the parameter estimates.

Each object is assigned a probability that it would possess a certain set of attribute values given that it was a member of a given cluster.

Expectation-Maximization(cont.)

Each cluster can be represented by a probability distribution, centered at a mean, and with a standard deviation.
Here, we have two clusters, corresponding to the Gaussian distributions $g(m_1, \sigma_1)$ and $g(m_2, \sigma_2)$, respectively, where the dashed circles represent the first standard deviation of the distributions.



Expectation-Maximization(cont.)

Step 1: Make an initial guess of the parameter vector

This involves randomly selecting k objects to represent the cluster means or centers (as in k-means partitioning), as well as making guesses for the additional parameters

Step 2: Iteratively refine the parameters (or clusters)

based on the following two steps:

1. Expectation Step
2. Maximization Step

1. Expectation Step

Assign each object x_i to cluster C_k with the probability

$$P(x_i \epsilon C_k) = p(C_k|x_i) = \frac{p(C_k)p(x_i|C_k)}{p(x_i)}$$

where $p(x_i|C_k) = N(m_k, E_k(x_i))$ follows the normal (i.e., Gaussian) distribution around mean, m_k , with expectation, E_k.

In other words, this step calculates the probability of cluster membership of object x_i , for each of the clusters.

These probabilities are the "expected" cluster memberships for object x_i

2. Maximization Step

Use the probability estimates from above to re-estimate (or refine) the model parameters.

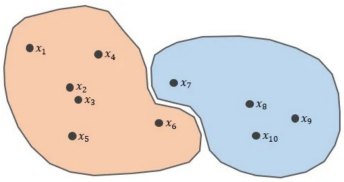
$$m_k = \frac{1}{n}\sum_{i=1}^{n}\frac{x_i P(x_i \epsilon C_k)}{\sum_j P(x_i \epsilon C_j)}$$

This step is the "maximization" of the likelihood of the distributions given the data.

Hard partition

where the data points are strictly allocated as a member of one cluster and are not a member of another cluster, assuming that the number of clusters is known.

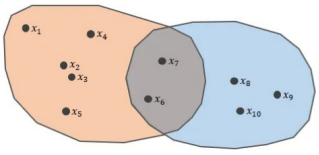
The k-means is one of the algorithms that use a hard partition.



Soft Partition

Every data point is given a probability of closeness $[0, 1]$ for existing clusters, assuming that the number of clusters is known.

One of the algorithms that use fuzzy partition is Fuzzy c-means.



Fuzzy Clustering

With fuzzy clustering, a data point can belong to multiple clusters to varying degrees, providing a richer representation of complex relationships

The key idea behind fuzzy clustering lies in the concept of **membership functions**.

Instead of assigning binary membership values (0 or 1) as in traditional clustering, fuzzy clustering employs membership values ranging from 0 to 1.
These values represent the **degree of belongingness** of each data point to each cluster.

A higher membership value indicates a **stronger association** with a particular cluster, while a lower value signifies a **weaker association**

Fuzzy c-means (FCM)

FCM is the most well-known and widely used fuzzy clustering technique.

It is an iterative algorithm that minimizes the sum of the weighted squared distances between each data point and the centers of the clusters.

The degree of membership of each data point to each cluster is calculated using a membership function, which assigns a probability value between 0 and 1 for each cluster

Fuzzy Clustering - Applications

- **Image Segmentation**: granular delineation of image regions, enabling better object recognition and scene understanding
- **Pattern Recognition**: character recognition or speech recognition
- **Customer Segmentation**:considering the degree of membership to different segments, businesses can tailor personalized marketing strategies
- **Document clustering**: cluster documents based on their content, such as keywords, topics, and themes

Self study

- Graph based Clustering
- Connected Components Clustering (CCC)
- highly connected subgraphs clustering

Evaluation of Clustering

Evaluating the effectiveness of the clustering results, known as clustering evaluation or validation

It can be used to determine which clustering algorithm is best suited for a particular dataset and task, and to tune the hyperparameters of these algorithms

Challenges

Since clustering is an unsupervised learning method, there are no ground truth labels against which the clustering results can be compared.

Determining the correct number of clusters or the best clustering is often a subjective decision, even for domain experts. What one considers as a meaningful cluster, another might dismiss as coincidental.

Evaluation of Clustering

Challenges

In many real-world datasets, the boundaries between clusters are not clear-enough
Some data points might sit at the boundary of two clusters and could be reasonably assigned to both.

Different applications might prioritize different aspects of clustering
For example, in one application, it might be essential to have tight, well-separated clusters, while in another, capturing the overall data structure might be more important.

Types of Evaluation

Two types of clustering evaluation measures (or metrics) Internal measures do not require any ground truth to assess the quality of clusters. They are based solely on the data and the clustering results. External measures compare the clustering results to ground truth labels.

Types of Evaluation(cont.)

Internal Evaluation Measures

Most internal validation measures are based on the following two criteria:

Compactness measures how closely related objects in the same cluster are.
Compactness can be measured in different ways, such as by using the variance of the points within each cluster, or computing the average pairwise distance between them.

Separation measures how distinct or well-separated a cluster is from other clusters.
Examples for measures of separation include pairwise distances between cluster centers or pairwise minimum distances between objects in different

Types of Evaluation(cont.)

External Evaluation Measures

compare the clustering results to ground truth labels

Silhouette Index

The **silhouette index** (or score) **measures the degree of separation** between clusters by comparing each object's similarity to its own cluster against its similarity to objects in other clusters

It measures how similar a point is to its own cluster (cohesion) compared to other clusters (separation).

- Close to 1: Well-clustered
- Close to 0: Overlapping clusters
- Negative: Misclassified points

Silhouette Coefficient

- For each point x_i in a cluster:
  1. Compute the average intra-cluster distance ($a(x_i)$):

$$a(x_i) = \frac{1}{|C| - 1} \sum_{x_j \in C, j \neq i} d(x_i, x_j)$$

We can interpret $a(x_i)$ as a measure of how well point x_i is matched to its own cluster (the smaller the value, the better the match).

Note that $a(x_i)$ is not clearly defined for clusters with size 1, in which case we set $s(x_i) = 0$.

Silhouette Coefficient (cont.)

2. Compute the nearest-cluster average distance ($b(x_i)$):

$$b(x_i) = \min_{C' \neq C} \frac{1}{|C'|} \sum_{j \in C'} d(i, j)$$

$b(x_i)$ is the average distance between x_i and the points in its neighboring cluster, i.e., the cluster whose points have the smallest average distance to xi

3. Compute the silhouette coefficient $S(x_i)$:

$$S(x_i) = \frac{b(x_i - a(x_i)}{\max(a(x_i), b(x_i))}$$

The silhouette coefficient ranges from **-1** to **+1**, where a high value indicates that the point is well matched to its own cluster and poorly matched to neighboring clusters

Silhouette Index(cont.)

Based on the silhouette coefficients of the samples, we now define the silhouette index (SI) as the average of the coefficients over all the data points:

$$SI = \frac{1}{n} \sum_{i=1}^{n} S(x_i)$$

Where, n is the total number of data points

The silhouette index provides an overall measure for the quality of the clustering:

1. An index close to 1 means that the clusters are compact and well separated.
2. An index around 0 indicates overlapping clusters.
3. An index close to -1 means the clustering has either too many or too few clusters.

Silhouette Index(cont.)

Importance

- Provides an intuitive way to assess clustering performance.
- Helps in choosing the optimal number of clusters.
- Can be used to compare different clustering algorithms.

Limitations

- Computationally expensive for large datasets.
- May not work well for non-spherical clusters.
- Sensitive to noise and outliers.

SI for estimating number of clusters

- The red dashed line represents the average silhouette coefficient for each clustering scenario.
- Since $K = 3$ has a higher average silhouette score than $K = 4$, it suggests that $K = 3$ might be a better choice for clustering in this case
- However, the final decision should also consider domain knowledge and the structure of the data

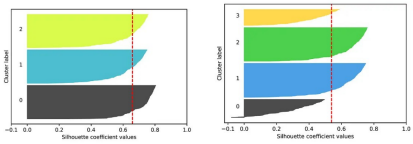


Figure: SI for (left) K=3 and (right) k=4

Calinski-Harabasz Index (CHI)

The Calinski-Harabasz index (CHI) measures the ratio between the between-cluster separation and the within-cluster dispersion

$$CHI = \frac{BCSS/(k-1)}{WCSS/(n-k)}$$

Where,

- k is the number of clusters
- n is the total number of data points
- BCSS (Between-Cluster Sum of Squares)
- WCSS (Within-Cluster Sum of Squares)

Pros: Performs well in identifying the optimal number of clusters
Cons: Assumes spherical clusters

Calinski-Harabasz Index(cont.)

BCSS (Between-Cluster Sum of Squares)

It is the weighted sum of squared Euclidean distances between each cluster centroid (mean) and the overall data centroid (mean)

$$BCSS = \sum_{i=1}^{k} n_i ||c_i - c||^2$$

Where n_i is the number of data points in cluster i, c_i is the centroid (mean) of cluster i, and c is the overall centroid(mean) of all data points

BCSS measures how well the clusters are separated from each other (the higher the better)

Calinski-Harabasz Index(cont.)

WCSS (Within-Cluster Sum of Squares)

It is the sum of squared Euclidean distances between the data points and their respective cluster centroid:

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - c_i||^2$$

WCSS measures the compactness or cohesiveness of the clusters (the smaller the better).

Minimizing WCSS (also known as the inertia) is the objective of centroid-based clustering such as k-means.

Calinski-Harabasz Index(cont.)

Advantages

- Simple to calculate and computationally efficient
- Easy to interpret. Higher values generally indicate better clustering
- Like silhouette score, it can be used to find the optimal number of clusters.

Other Evaluation Scores

Read yourself

- Davies-Bouldin Index - measures the average similarity between each cluster and its most similar cluster
- Dunn Index - Measures the ratio of the minimum inter-cluster distance to the maximum intra-cluster distance. Higher values indicate better clustering.

What to choose?

- Silhouette Index or Dunn Index - If clusters are well-separated and compact
- Davies-Bouldin Index or CH Index - If clusters have varying densities

External Evaluation Measures

External evaluation measures are used when the true labels of the data points are known.

These measures compare the the results of the clustering algorithm against the ground truth labels.

Contingency Matrix

Similar to confusion matrices in classification problems, a contingency matrix (or table) describes the relationship between the ground truth labels and the cluster labels

The rows of the matrix represent the ground-truth classes and its column represent the clusters. Each cell in the matrix, denoted by nij, contains the count of the data points that have a class label i and were assigned to cluster j

Various external evaluation metrics, such as adjusted Rand index and Fowlkes-Mallows index use the contingency matrix as the basis for their calculation

Random Index

The Rand Index (RI), named after William Rand, measures the similarity between the cluster assignments and the true class labels by making pairwise comparisons.

It is calculated as the ratio of the number of agreements between the cluster assignments and the class labels to the total number of pairs of data points:

$$RI = \frac{a+b}{\binom{n}{2}}$$

Where,

- a is the number of pairs of points that have the same class label and also belong to the same cluster.

Random Index (cont.)

- b is the number of pairs of points that have different class labels and belong to different clusters.
- n is the total number of points.

The RI ranges from 0 to 1, where 1 indicates that the cluster assignments and the class labels are exactly the same

Fowlkes-Mallows Index (FMI)

The Fowlkes-Mallows Index (FMI) is defined as the geometric mean of the pairwise precision (the accuracy of grouped pairs of points) and recall (the completeness of correctly grouping pairs that belong together):

$$FMI = \frac{TP}{\sqrt{(TP+FP)(TP+FN)}}$$

where:

- TP (True Positive) is the number of pairs of points that have the same class label and belong to the same cluster.
- FP (False Positive) is the number of pairs of points that have different class labels but are assigned to the same cluster.
- FN (False Negative) is the number of pairs of points that have the same class labels but are assigned to different clusters

Fowlkes-Mallows Index (FMI) (cont.)

The FMI score ranges from 0 to 1, where 0 indicates no correlation between the clustering results and the true labels, and 1 represents a perfect correlation.

Other Evaluation Scores

Read yourself

- Adjusted Rand index (ARI)
- Homogeneity
- Completeness
- V-measure

Parameter Estimation

The number of clusters (KK) affects clustering quality.

- Too few clusters may merge distinct groups
- Too many may overfit noise

Common Methods

- Elbow Method
- Silhouette Score
- Davies-Bouldin Index
- Dunn Index

Elbow Method

A technique used to determine the optimal number of clusters (K) in clustering.

Based on the Sum of Squared Errors (SSE) or Within-Cluster Sum of Squares (WCSS).

The goal is to find a balance between too few and too many clusters.

Working

- Compute clustering for different values of K (e.g., 1 to 10)
- Calculate the WCSS (Within-Cluster Sum of Squares) for each K
- Plot KK vs. WCSS
- Identify the "elbow point" where WCSS starts decreasing at a slower rate

Elbow Method (cont.)

WCSS measures how well the data points are clustered around their respective centroids. It is defined as the sum of the squared distances between each point and its cluster centroid:

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - c_i||^2$$