Unit - 5

Cluster Analysis

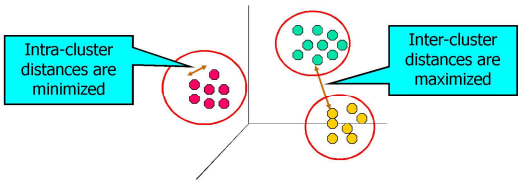
Overview

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Graph-Based Methods
8. Summary

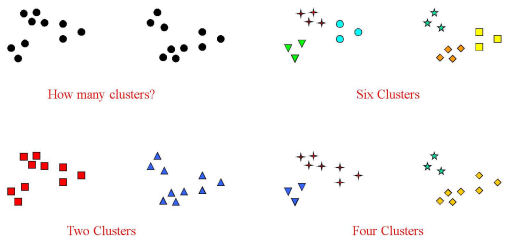
2

What is Cluster Analysis?

- Cluster: a collection of data objects
- Cluster Analysis:
  - An unsupervised learning
  - 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



Notion of a Cluster can be Ambiguous



Clustering: Rich Applications and Multidisciplinary Efforts

- Pattern Recognition
- Spatial Data Analysis
  - Create thematic maps in GIS by clustering feature spaces
  - Detect spatial clusters or for other spatial mining tasks
- Image Processing
- Economic Science (especially market research)
- WWW
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns

5

Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

6

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns

Measure the Quality of Clustering

- Dissimilarity/Similarity metric: Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
- There is a separate "quality" function that measures the "goodness" of a cluster.
- The definitions of distance functions are usually very different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables.
- Weights should be associated with different variables based on applications and data semantics.

Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Ability to handle dynamic data
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Data Structures

- Data matrix (or *object-by-variable* structure)
  - (two modes)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix (or *object-by-object* structure)
  - (one mode)

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

Types of Data in Clustering Analysis

- Interval-scaled variables
- Binary variables
- Nominal, ordinal, and ratio variables
- Variables of mixed types

Interval-Scaled Variables

- Continuous measurements of a roughly linear scale. E.g., weight, height, longitude, latitude, etc.
- Standardize data
  - Calculate the mean absolute deviation:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|)$$

where $m_f = \frac{1}{n}(x_{1f} + x_{2f} + \cdots + x_{nf})$

  - Calculate the standardized measurement (*z-score*)

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

- Using mean absolute deviation is more robust than using standard deviation

Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: *Minkowski distance*:

$$d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+...+|x_{i_p}-x_{j_p}|^q)}$$

where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects, and q is a positive integer

- If $q = 1$, d is Manhattan distance

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}-x_{j_p}|$$

13

Similarity and Dissimilarity Between Objects (Cont.)

- *If* $q = 2$, d is Euclidean distance:

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_p}-x_{j_p}|^2)}$$

  - Properties
    - d(i,j) ≥ 0
    - d(i,i) = 0
    - d(i,j) = d(j,i)
    - d(i,j) ≤ d(i,k) + d(k,j)

- Also, one can use weighted distance, parametric Pearson product moment correlation, or other disimilarity measures

14

Binary Variables

- A contingency table for binary data

Object *i* \ Object *j*	1	0	*sum*
1	*a*	*b*	*a+b*
0	*c*	*d*	*c+d*
sum	*a+c*	*b+d*	*p*

- Distance measure for symmetric binary variables: $d(i,j)=\frac{b+c}{a+b+c+d}$
- Distance measure for asymmetric binary variables: $d(i,j)=\frac{b+c}{a+b+c}$
- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables): $sim_{Jaccard}(i,j)=\frac{a}{a+b+c}$

15

Dissimilarity between Binary Variables

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

Name	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	1	0	1	0	0	0
Mary	1	0	1	0	1	0
Jim	1	1	0	0	0	0

$$d(jack, mary)=\frac{0+1}{2+0+1}=0.33$$

$$d(jack, jim)=\frac{1+1}{1+1+1}=0.67$$

$$d(jim, mary)=\frac{1+2}{1+1+2}=0.75$$

16

Nominal Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching
  - m: # of matches, p: total # of variables

$$d(i,j)=\frac{p-m}{p}$$

- Method 2: use a large number of binary variables
  - creating a new binary variable for each of the M nominal states

17

Ordinal Variables

- An ordinal variable can be discrete or continuous
- Order is important, e.g., rank
- Can be treated like interval-scaled
  - Replace x_{if} by their rank $r_{if} \in \{1, \ldots, M_f\}$
  - Map the range of each variable onto [0, 1] by replacing i^{th} object in the f^{th} variable by

$$z_{if}=\frac{r_{if}-1}{M_f-1}$$

  - Compute the dissimilarity using methods for interval-scaled variables

18

Ratio-Scaled Variables

- Ratio-scaled variable: a positive measurement on a nonlinear scale, approximately at exponential scale, such as Ae^{Bt} or Ae^{-Bt}
- Methods: (Three different approaches)
  - Treat them like interval-scaled variables—*not a good choice!* (why?—the scale can be distorted)
  - Apply logarithmic transformation

    $y_{if} = log(x_{if})$
  - Treat them as continuous ordinal data treat their rank as interval-scaled

19

Variables of Mixed Types

- A database may contain all the six types of variables
  - Interval-scaled, symmetric binary, asymmetric binary, nominal, ordinal, and ratio-scaled
- A weighted formula to combine their effects

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

Where $\delta_{ij}^{(f)} = 0$ if either (i) x_{if} or x_{jf} is missing, or (ii) $x_{if}=x_{jf}=0$ and f is asymmetric binary, otherwise, $\delta_{ij}^{(f)} = 1$

  - *f* is binary or nominal:

    $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ otherwise
  - *f* is interval-based: use the normalized distance
  - *f* is ordinal or ratio-scaled

    compute ranks r_{if} and $z_{if} = \frac{r_{if}-1}{M_f-1}$

    treat z_{if} as interval-scaled

$$d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$$

20

Vector Objects

- Vector objects: keywords in documents, gene features in micro-arrays, etc.
- Broad applications: information retrieval, biological taxonomy, etc.
- Cosine measure:

$$s(x, y) = \frac{x^t \cdot y}{\|x\| \|y\|}$$

Where, x^t is the transposition of vector x and ||x|| is the Euclidean norm of vector x

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$$

Conceptually, it the length of the vector x

21

Mathematics Behind Distance & Angle

Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$ be two m-dimensional vectors given as

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Dot Product The *dot product* between $\mathbf{a}$ and $\mathbf{b}$ is defined as the scalar value

$$\mathbf{a}^T\mathbf{b} = \begin{pmatrix} a_1 & a_2 & \cdots & a_m \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

$$= a_1b_1 + a_2b_2 + \cdots + a_mb_m$$

$$= \sum_{i=1}^{m} a_ib_i$$

May 7, 2019

22

Mathematics... (cont...)

Length The *Euclidean norm* or *length* of a vector $\mathbf{a} \in \mathbb{R}^m$ is defined as

$$\|\mathbf{a}\| = \sqrt{\mathbf{a}^T\mathbf{a}} = \sqrt{a_1^2 + a_2^2 + \cdots + a_m^2} = \sqrt{\sum_{i=1}^{m} a_i^2}$$

The *unit vector* in the direction of $\mathbf{a}$ is given as

$$\mathbf{u} = \frac{\mathbf{a}}{\|\mathbf{a}\|} = \left(\frac{1}{\|\mathbf{a}\|}\right)\mathbf{a}$$

By definition $\mathbf{u}$ has length $\|\mathbf{u}\| = 1$, and it is also called a *normalized* vector, which can be used in lieu of $\mathbf{a}$ in some analysis tasks.

The Euclidean norm is a special case of a general class of norms, known as L_p-*norm*, defined as

$$\|\mathbf{a}\|_p = \left(|a_1|^p + |a_2|^p + \cdots + |a_m|^p\right)^{\frac{1}{p}} = \left(\sum_{i=1}^{m} |a_i|^p\right)^{\frac{1}{p}}$$

for any $p \neq 0$. Thus, the Euclidean norm corresponds to the case when $p = 2$.

Mathematics... (cont...)

Distance From the Euclidean norm we can define the *Euclidean distance* between $\mathbf{a}$ and $\mathbf{b}$, as follows

$$\delta(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\| = \sqrt{(\mathbf{a} - \mathbf{b})^T(\mathbf{a} - \mathbf{b})} = \sqrt{\sum_{i=1}^{m} (a_i - b_i)^2} \qquad (1.1)$$

Thus, the length of a vector is simply its distance from the zero vector $\mathbf{0}$, all of whose elements are 0, i.e., $\|\mathbf{a}\| = \|\mathbf{a} - \mathbf{0}\| = \delta(\mathbf{a}, \mathbf{0})$.

From the general L_p-norm we can define the corresponding L_p-distance function, given as follows

$$\delta_p(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_p \qquad (1.2)$$

May 7, 2019

24

Mathematics... (cont...)

Angle The cosine of the smallest angle between vectors **a** and **b**, also called the *cosine similarity*, is given as

$$\cos\theta = \frac{\mathbf{a}^T\mathbf{b}}{\|\mathbf{a}\|\,\|\mathbf{b}\|} = \left(\frac{\mathbf{a}}{\|\mathbf{a}\|}\right)^T\left(\frac{\mathbf{b}}{\|\mathbf{b}\|}\right) \quad (1.3)$$

Thus, the cosine of the angle between **a** and **b** is given as the dot product of the unit vectors $\frac{\mathbf{a}}{\|\mathbf{a}\|}$ and $\frac{\mathbf{b}}{\|\mathbf{b}\|}$.

The *Cauchy-Schwartz* inequality states that for any vectors **a** and **b** in $\mathbb{R}^m$

$$|\mathbf{a}^T\mathbf{b}| \le \|\mathbf{a}\| \cdot \|\mathbf{b}\|$$

It follows immediately from the Cauchy-Schwartz inequality that

$$-1 \le \cos\theta \le 1$$

Since the smallest angle $\theta \in [0°, 180°]$ and since $\cos\theta \in [-1, 1]$, the cosine similarity value ranges from $+1$ corresponding to an angle of 0°, to -1 corresponding to an angle of 180° (or π radians).

Typical Alternatives to Calculate the Distance between Clusters

- Single link: Smallest distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = min(t_{ip}, t_{jq})$
- Complete link: Largest distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = max(t_{ip}, t_{jq})$
- Average: Average distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = avg(t_{ip}, t_{jq})$
- Centroid: Distance between the centroids of two clusters, i.e., $dis(K_i, K_j) = dis(C_i, C_j)$
- Medoid: Distance between the medoids of two clusters, i.e., $dis(K_i, K_j) = dis(M_i, M_j)$
  - Medoid: one chosen, centrally located object in the cluster

Centroid, Radius and Diameter of a Cluster (for numerical data sets)

- Centroid: The "middle" of a cluster $C_m = \frac{\sum_{i=1}^{N} x_i}{N}$
- Radius: Square root of the average distance from centroid of the cluster to all other points. $R_m = \sqrt{\frac{\sum_{i=1}^{N}(x_i - c_m)^2}{N}}$
- Diameter: Square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(x_i - x_j)^2}{N(N-1)}}$$

Representative-Based Clustering

- Given a dataset with n points in a d-dimensional space $\mathbf{D} = \{\mathbf{x}_i\}_{i=1}^n$, and given the number of desired clusters k, the goal of representative-based clustering is to partition the dataset into *k* groups or clusters C = {C1, C2, …, Ck}
- For each cluster C_i, there exists a representative point (mean or centroid) that summarizes the cluster

$$\boldsymbol{\mu}_i = \frac{1}{n_i}\sum_{\mathbf{x}_j \in C_i}\mathbf{x}_j$$

  Where $n_i = |C_i|$
- **Brute-force approach:** Simply generate all possible partitions of n points into k clusters, evaluate some optimization score for each of them, and retain the clustering that yields the best score

Cont...

- The exact number of ways of partitioning n points into k non-empty and disjoint parts is given by the Stirling numbers of the second kind, given as

$$S(n,k) = \frac{1}{k!}\sum_{t=0}^{k}(-1)^t\binom{k}{t}(k-t)^n$$

- Informally, each point can be assigned to any one of the k clusters, resulting in at most k^n possible clusterings. However, any permutation of the k clusters within a given clustering yields an equivalent clustering, therefore, there are $O(k^n/k!)$ clusterings of n points into k groups.
- Such exhaustive enumeration and scoring of all possible clusterings is not practically feasible

Cont...

- Given a clustering C = {C1,C2,⋯ ,Ck} we need some scoring function that evaluates its quality or goodness. This sum of squared errors scoring function is defined as

$$SSE(\mathcal{C}) = \sum_{i=1}^{k}\sum_{\mathbf{x}_j \in C_i}\|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

- The goal is to find the clustering that minimizes the SSE score

$$\mathcal{C}^* = \arg\min_{\mathcal{C}}\{SSE(\mathcal{C})\}$$

- K-means employs a greedy iterative approach to find a clustering that minimizes the SSE objective.

K-means Algorithm

- Initializes the cluster means by randomly generating k points in the data space.
  - Typically done by generating a value uniformly at random within the range for each dimension.
- Each iteration of K-means consists of two steps:
  - Cluster assignment, and
  - Centroid update

31

K-means Algorithm (cont…)

Cluster assignment

- Given the *k* cluster means, each point $x_j \in D$ is assigned to the closest mean. That is, each point x_j is assigned to cluster C_{j^*}, where

$$j^* = \arg\min_{i=1}^{k} \left\{ \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2 \right\}$$

Centroid update

- Given a set of clusters C_i, $i = 1, 2, \ldots, k$, new mean values are computed for each cluster from the points in C_i

The **cluster assignment** and **centroid update** steps are carried out iteratively until we reach a fixed point or local minima

32

K-means Algorithm (cont…)

- Practically, one can assume that K-means has converged if the centroids do not change from one iteration to the next. For instance, we can stop if

$$\sum_{i=1}^{k} \left\| \boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t-1} \right\|^2 \leq \epsilon$$

- Where $\epsilon > 0$ is the convergence threshold, and *t* denotes the current iteration

33

K-means Algorithm (cont…)

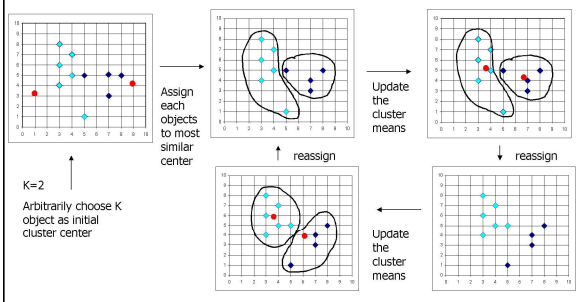
Algorithm 13.1: K-means Algorithm

```
K-MEANS (D, k, ε):
t = 0
Randomly initialize k centroids: μ_1^t, μ_2^t, ..., μ_k^t ∈ R^d
repeat
    t ← t + 1
    // Cluster Assignment Step
    foreach x_j ∈ D do
        j* ← arg min_i { ||x_j − μ_i^t||^2 } // Assign x_j to closest centroid
        C_j* ← C_j* ∪ {x_j}
    // Centroid Update Step
    foreach i = 1 to k do
        μ_i^t ← (1/|C_i|) Σ_{x_j ∈ C_i} x_j
until Σ_{i=1}^{k} ||μ_i^t − μ_i^{t−1}|| ≤ ε
```

34

The K-Means Clustering Method

- Example



35

Illustration of K-Means

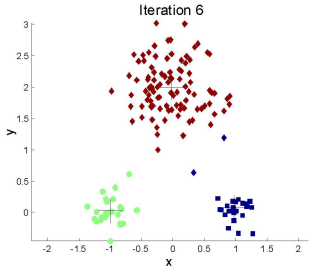
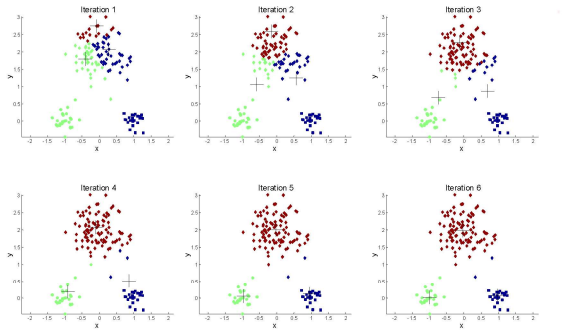


Illustration of K-Means



Comments on the *K-Means* Method

- Strength: *Relatively efficient*: $O(tknd)$, where n is #objects, k is #clusters, t is #iterations, and d is #dimensions in input space. Normally, d, k, $t << n$.
  - **Cluster assignment:** $O(nkd)$, since for each of the n points we have to compute its distance to each of the k clusters, which takes d operations in d dimensions
  - **Centroid re-computation:** $O(nd)$, since we have to add a total of n d-dimensional points
  - **Assuming t iterations, total time for K-means is $O(tnkd)$**

38

Comments on the *K-Means* Method

- Comment: Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- Weakness
  - Applicable only when *mean* is defined, then what about categorical data?
  - Need to specify *k*, the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

39

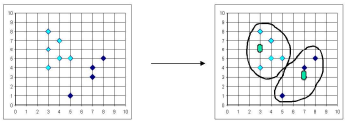
Variations of the K-Means Method

- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang' 98)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters

40

What is the Problem of the K-Means Method?

- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.



41

The *K-Medoids* Clustering Method

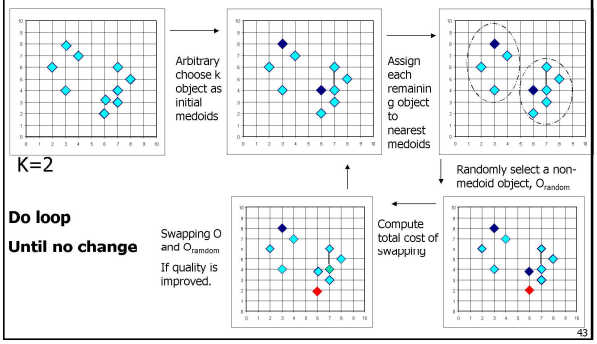
- Find *representative* objects, called medoids, in clusters
- PAM (Partitioning Around Medoids, 1987)
  - Partitioning is performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point. i.e., an absolute-error criterion is used, defined as

$$E = \sum_{j=1}^{k} \sum_{\boldsymbol{p} \in C_j} |\boldsymbol{p} - \boldsymbol{o_j}|$$

  - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering

42

A Typical K-Medoids Algorithm (PAM)



43

PAM (Partitioning Around Medoids) (1987)

- PAM (Kaufman and Rousseeuw, 1987)
- Use real object to represent the cluster
  1. Select ***k*** representative objects arbitrarily
  2. For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost TC_{ih}
  3. For each pair of ***i*** and ***h***,
     1. If $TC_{ih} < 0$, ***i*** is replaced by ***h***
     2. Then assign each non-selected object to the most similar representative object
  4. repeat steps 2-3 until there is no change

44

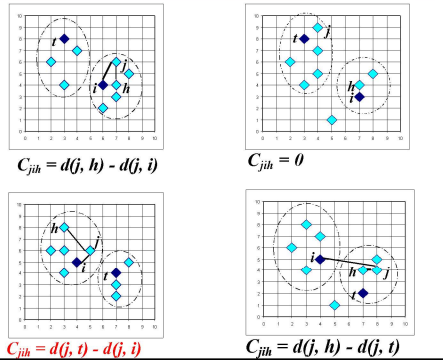
- To determine whether a non-representative object, o_{random}, is a good replacement for a current representative object, o_j, the following four cases are examined for each of the non-representative objects, p:
- **Case 1:** *p* currently belongs to o_j. If o_j is replaced by o_{random} as a representative object and *p* is closest to one of the other representative objects, o_i, $i \neq j$, then p is reassigned to o_i.

45

- **Case 2:** *p* currently belongs to o_j. If o_j is replaced by o_{random} as a representative object and *p* is closest to o_{random}, then *p* is reassigned to o_{random}.
- **Case 3:** *p* currently belongs to representative object, o_i, $i \neq j$. If o_j is replaced by o_{random} as a representative object and *p* is still closest to o_i, then the assignment does not change.
- **Case 4:** *p* currently belongs to representative object, o_i, $i \neq j$. If o_j is replaced by o_{random} as a representative object and p is closest to o_{random}, then *p* is reassigned to o_{random}.

46

PAM Clustering: Total swapping cost $TC_{ih}=\sum_j C_{jih}$



47

What Is the Problem with PAM?

- PAM is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean
- PAM works efficiently for small data sets but does not **scale well** for large data sets.
  - $O(k(n-k)^2)$ for each iteration

    where n is #of data points, k is #of clusters

➔ Sampling based method,

CLARA (Clustering LARge Applications)

48

Hierarchical Clustering

- Given *n* points in a *d*-dimensional space, hierarchical clustering aims to create a sequence of nested partitions, which can be conveniently visualized via a tree or hierarchy of clusters, also called the cluster dendrogram.
- The clusters in the hierarchy range from the fine-grained to the coarse-grained
  - The lowest level of the tree (the leaves) consists of each point in its own cluster
  - The highest level (the root) consists of all points in one cluster
  - Both of these may be considered to be trivial clusters

49

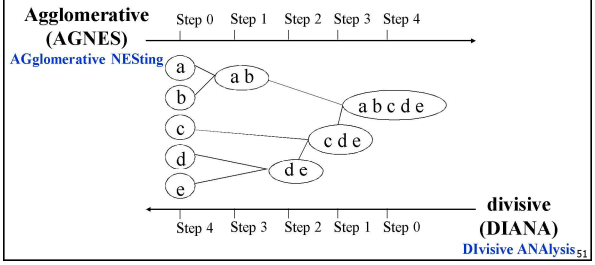
Hierarchical Clustering (cont…)

- Two main algorithmic approaches to mine hierarchical clusters:
  - Agglomerative: Works in a bottom-up manner. Starting with each of the *n* points in a separate cluster, repeatedly merges the most similar pair of clusters until all points are members of a single cluster.
  - Divisive: Do just the opposite; working in a top-down manner. Starting with all the points in the same cluster, recursively splits the clusters until all points are in separate clusters.

50

Hierarchical Clustering (cont…)

- Uses distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



51

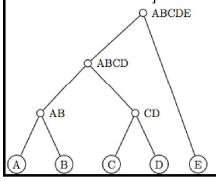
Preliminaries

- Clustering: Given a dataset $D = \{x_1, \cdots, x_n\}$, where $x_i \in R^d$, a clustering $C = \{C_1, \cdots, C_k\}$ is a partition of D, i.e., each cluster is a set of points $C_i \subseteq D$, such that the clusters are pairwise disjoint, i.e., $C_i \cap C_j = \emptyset$ (for all i<>j), and $\cup C_i = D$
- Cluster Nesting: A clustering $A = \{A_1, \cdots, A_r\}$ is said to be nested in another clustering $B = \{B_1, \cdots, B_s\}$ if and only if r > s, and for each cluster $A_i \in A$, there exists a cluster $B_j \in B$, such that $A_i \subseteq B_j$.
- Hierarchical clustering yields a sequence of *n* nested partitions $C_1, \cdots, C_n$, ranging from the trivial clustering $C_1 = \{\{x_1\}, \cdots, \{x_n\}\}$ where each point is in a separate cluster, to the other trivial clustering $C_n = \{\{x_1, \cdots, x_n\}\}$, where all points are in a single cluster.

52

Preliminaries (cont…)

- In general, the clustering C_{t-1} is nested in the clustering C_t.
- The cluster dendrogram is a rooted binary tree that captures the cluster nesting structure, with edges between cluster $C_i \in C_{t-1}$ and cluster $C_j \in C_t$ if C_i is nested in C_j, i.e., if $C_i \subset C_j$.



clustering	clusters
C_1	$\{A\},\{B\},\{C\},\{D\},\{E\}$
C_2	$\{AB\},\{C\},\{D\},\{E\}$
C_3	$\{AB\},\{CD\},\{E\}$
C_4	$\{ABCD\},\{E\}$
C_5	$\{ABCDE\}$

Agglomerative Clustering

- Begins with each of the n points in a separate cluster, and repeatedly merges the two closest clusters until all points are members of the same cluster
- Formally, given a set of clusters $C = \{C_1, C_2, .., C_m\}$, we find the closest pair of clusters C_i and C_j and merge them into a new cluster $C_{ij} = C_i \cup C_j$.
- Next, we update the set of clusters by removing C_i and C_j and adding C_{ij}, as follows $C = C \setminus \{\{C_i\} \cup \{C_j\}\} \cup \{C_{ij}\}$, and repeat the process until C contains only one cluster.
- Since the number of clusters decreases by one in each step, this process results in a sequence of n nested clusterings.
- If specified, we can stop the merging process when there are exactly k clusters remaining.

54

Aggl. Clustering (Pseudo Code

Algorithm 14.1: Agglomerative Hierarchical Clustering Algorithm

```
AGGLOMERATIVECLUSTERING(D, k):
C ← {C_i = {x_i} | x_i ∈ D} // Each point in separate cluster
Δ ← {δ(x_i, x_j): x_i, x_j ∈ D} // Compute distance matrix
repeat
    Find the closest pair of clusters C_i, C_j ∈ C
    C_ij ← C_i ∪ C_j // Merge the clusters
    C ← C \ {{C_i} ∪ {C_j}} ∪ {C_ij} // Update the clustering
    Update distance matrix Δ to reflect new clustering
until |C| = k
```

55

Distance Between Clusters

- The main step in hierarchical clustering algorithm is to determine the closest pair of clusters, which can be computed using various distance measures, such as single link, complete link, group average, and so on.
- The between cluster distances are ultimately based on the distance between two points, which is typically computed using the Euclidean distance or L_2-norm, defined as

$$\delta(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2 = \left(\sum_{i=1}^{d}(x_i - y_i)^2\right)^{1/2}$$

56

Dist. Between Clusters (cont...)

- Single Link/ Min Distance: Given two clusters C_i and C_j , the distance between them, denoted $\delta(C_i , C_j)$ is defined as the minimum distance between a point in C_i and a point in C_j

$$\delta(C_i, C_j) = \min\{\delta(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}$$

- Complete Link/ Max Distance: The distance between two clusters is defined as the maximum distance between a point in C_i and a point in C_j

$$\delta(C_i, C_j) = \max\{\delta(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}$$

57

Dist. Between Clusters (cont...)

- Average Distance: The distance between two clusters is defined as the average pairwise distance between points in C_i and C_j

$$\delta(C_i, C_j) = \frac{\sum_{\mathbf{x} \in C_i}\sum_{\mathbf{y} \in C_j}\delta(\mathbf{x}, \mathbf{y})}{n_i \cdot n_j}$$

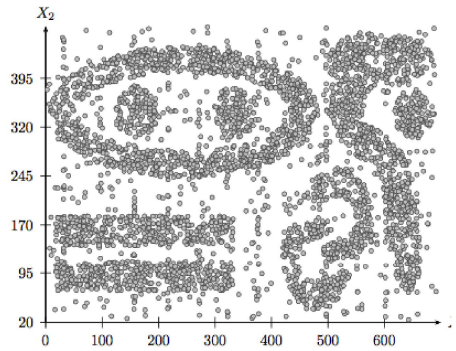
- Mean Distance: The distance between two clusters is defined as the distance between the means or centroids of the two clusters

$$\delta(C_i, C_j) = \delta(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$$

where $\boldsymbol{\mu}_i = \frac{1}{n_i}\sum_{\mathbf{x} \in C_i}\mathbf{x}$.

58

Density-Based Dataset



59

Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points, rather only on distance between points.
- Determines clusters of arbitrary shape (non-convex)
- Uses the concept of ε-neighborhood, which can be defined for a point x as:

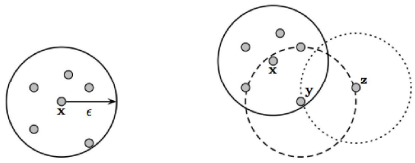
  $N_\varepsilon(x) = \{Y \mid \partial(x, y) \leq \varepsilon\}$

  where, $\partial(x, y)$ represents the distance between x and y
- For any point $x \in D$, we say that x is a core point if

  $|N_\varepsilon(x)| \geq minpts$, where *minpts* is a user-defined local density or frequency threshold.

Density-Based Clustering Methods

- A border point is defined as a point x that does not meet the *minpts* threshold, (i.e., $|N_\varepsilon(x)|$<minpts), but it belongs to the neighborhood of some core point z, i.e., $x \in N_\varepsilon(z)$.
- Finally, if a point is neither a core nor a border point, then it is called a noise point or an outlier.

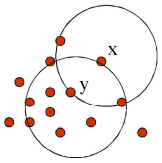


minpts: 6
x: Core point
y: Border point
z: Noise point

Density-Based Clustering Methods

- Directly density-reachable: A point x is directly density reachable from another point y, if $x \in N_\varepsilon(y)$, y is a core point.

$|N_\varepsilon(y)| \geq minpts$



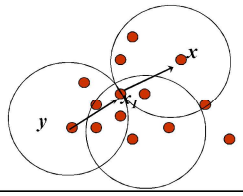
minpts=5

ε = 1

62

Density-Based Clustering Methods

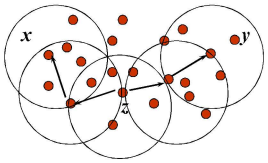
- Density-reachable: A point x is density reachable from another point y, if there exists a chain of points, $x_0, x_1, \cdots, x_l$, such that $x = x_0$ and $y = x_l$, and x_i is directly density reachable from x_{i-1} for all i = 1, 2, ⋯, l.
- In other words, there is set of core points leading from y to x.
- Note that density reachability is an asymmetric or directed relationship



63

Density-Based Clustering Methods

- Density-connected: Any two points x and y are density connected if there exists a core point z, such that both x and y are density reachable from z.



A density-based cluster is defined as a maximal set of density connected points.

64

Density-Based Clustering Methods

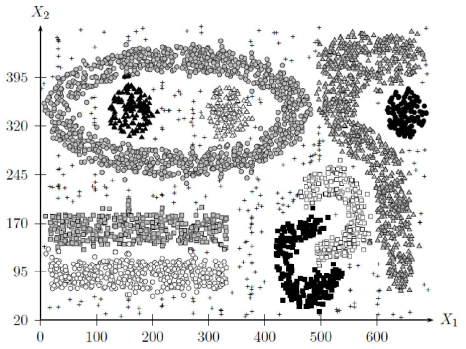
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)
  - OPTICS: Ankerst, et al. (SIGMOD'99)

DBSCAN: Pseudo Code

Algorithm 15.1: Density-based Clustering Algorithm

```
DBSCAN (D, ε, minpts):
Core ← ∅
foreach x_i ∈ D do // Find the core points
    Compute N_ε(x_i)
    id(x_i) ← ∅ // cluster id for x_i
    if N_ε(x_i) ≥ minpts then Cores ← Cores ∪ {x_i}
k ← 0 // cluster id
foreach x_i ∈ Core, such that id(x_i) = ∅ do
    k ← k + 1
    id(x_i) ← k // assign x_i to cluster id k
    DensityConnected (x_i, k)
C ← {C_i}_{i=1}^k, where C_i ← {x ∈ D | id(x) = i}
Noise ← {x ∈ D | id(x) = ∅}
Border ← D \ {Core ∪ Noise}
return C, Core, Border, Noise
DensityConnected (x, k):
foreach y ∈ N_ε(x) do
    id(y) ← k // assign y to cluster id k
    if y ∈ Core then DensityConnected (y, k)
```

Results: DBSCAN (ε =15, minpts: 10)



67

Computational Complexity of DBSCAN

- Main cost: Computing the ε-neighborhood for each point.
  - If the dimensionality is not too high this can be done efficiently using a spatial index structure in O(n log n) time.
  - If dimensionality is high, it takes $O(n^2)$ to compute the neighborhood for each point.
- Once $N_\varepsilon(x)$ has been computed, the algorithm needs only a single pass over all the points to find the density connected clusters.
- Thus, the overall complexity of DBSCAN is $O(n^2)$ in the worst-case.

68

Limitations of DBSCAN

- It is sensitive to the choice of ε, in particular, if clusters have different densities.
- If ε is too small, sparser clusters will be categorized as noise.
- If ε is too large, denser clusters may be merged together.
- In other words, if there are clusters with different local densities, then a single ε value may not suffice.

69

Markov Clustering

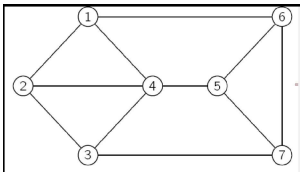
- A graph clustering method based on simulating a random walk on a weighted graph
- Basic intuition: If node transitions reflect the weights on the edges, then transitions from one node to another within a cluster are much more likely than transitions between nodes from different clusters.
  - This is because, nodes within a cluster have higher similarities or weights, and nodes across clusters have lower similarities.

Markov Clustering

- Given the weighted adjacency matrix A for a graph G, the normalized adjacency matrix is given as

  $M = \Delta^{-1}A$, where Δ is the degree matrix
  - The matrix M can be interpreted as the n × n transition matrix where the entry $m_{ij} = a_{ij}/d_i$ can be interpreted as the probability of transitioning or jumping from node i to node j in the graph G.
  - This is because, M is a row stochastic or Markov matrix, that satisfies the following conditions:
    - Elements of M are non-negative, i.e., $m_{ij} \geq 0$, which follows from the fact that A is non-negative, and
    - Rows of M are probability vectors, i.e., row elements add to one, since

$$\sum_{j=1}^{n} m_{ij} = \sum_{j=1}^{n} \frac{a_{ij}}{d_i} = 1$$



Sample Graph

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Adjacency Matrix

$$\boldsymbol{\Delta} = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}$$

Degree Matrix

Markov Matrix

$$M = \Delta^{-1}A = \begin{pmatrix} 0 & 0.33 & 0 & 0.33 & 0 & 0.33 & 0 \\ 0.33 & 0 & 0.33 & 0.33 & 0 & 0 & 0 \\ 0 & 0.33 & 0 & 0.33 & 0 & 0 & 0.33 \\ 0.25 & 0.25 & 0.25 & 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0.33 & 0 & 0.33 & 0.33 \\ 0.33 & 0 & 0 & 0 & 0.33 & 0 & 0.33 \\ 0 & 0 & 0.33 & 0 & 0.33 & 0.33 & 0 \end{pmatrix}$$

Markov Clustering

- M is thus the transition matrix for a Markov chain or a Markov random walk on graph G.
- A Markov chain is a discrete-time stochastic process over a set of states (vertices)
- The Markov chain makes a transition from one node to another at discrete time-steps t = 1, 2, . . ., with the probability of making a transition from node i to node j given as m_{ij}.
- Let the random variable X_t denotes the state at time t. The Markov property means that the probability distribution of X_t over the states at time t depends only on the probability distribution of X_{t-1}, i.e., $P(X_t = i \mid X_0, X_1, \ldots, X_{t-1}) = P(X_t = i \mid X_{t-1})$
- The Markov chain is homogeneous, i.e., the transition probability $P(X_t = j \mid X_{t-1} = i) = m_{ij}$ is independent of the time-step t.

Markov Clustering

- Given node i the transition matrix M specifies the probabilities of reaching any other node j in one time-step.
- Starting from node i at t = 0, let the probability of being at node j at t = 2, i.e., after two steps is $m_{ij}(2)$, which can be computed as follows:

$$m_{ij}(2) = P(X_2 = j | X_0 = i)$$
$$= \sum_{a=1}^{n} P(X_1 = a | X_0 = i) P(X_2 = j | X_1 = a)$$
$$= \sum_{a=1}^{n} m_{ia} m_{aj} = \mathbf{m}_i^T M_j$$

Markov Clustering

$$\mathbf{M}^2 = \mathbf{M} \cdot \mathbf{M} = \begin{pmatrix} — \mathbf{m}_1^T — \\ — \mathbf{m}_2^T — \\ \vdots \\ — \mathbf{m}_n^T — \end{pmatrix} \begin{pmatrix} | & | & & | \\ M_1 & M_2 & \cdots & M_n \\ | & | & & | \end{pmatrix}$$
$$= \left\{ \mathbf{m}_i^T M_j \right\}_{i,j=1}^{n} = \left\{ m_{ij}(2) \right\}_{i,j=1}^{n}$$

- It implies that M^2 is precisely the transition probability matrix for the Markov chain over two time-steps.
- Likewise, the three step transition matrix is $M^2 \cdot M = M^3$
- In general, the transition probability matrix for t time-steps is given as $M^{t-1} \cdot M = M^t$
- A random walk on G thus corresponds to taking successive powers of the transition matrix M.

Transition Probability Inflation

- A variation of the random walk, where the probability of transitioning from node i to j is inflated by taking each element m_{ij} to the power r ≥ 1.
- Given a transition matrix M, the inflation operator is defined as follows:

$$\Upsilon(\mathbf{M}, r) = \left\{ \frac{(m_{ij})^r}{\sum_{a=1}^{n} (m_{ia})^r} \right\}_{i,j=1}^{n}$$

- The inflation operation results in a transformed or inflated transition probability matrix, since the elements remain non-negative, and each row is normalized to sum to 1.
- The net effect of the inflation operator is to increase the higher probability transitions and decrease the lower probability transitions.

Markov Clustering Algorithm (MCL)

- An iterative method that interleaves matrix expansion and inflation steps.
- Matrix expansion corresponds to taking successive powers of the transition matrix, leading to random walks of longer lengths.
- Matrix inflation makes the higher probability transitions even more likely and reduces the lower probability transitions.

Mark

$$\Psi(M, r) = \left\{ \frac{(m_{ij})^r}{\sum_{a=1}^{n} (m_{ia})^r} \right\}_{i,j=1}^{n}$$

Algorithm: M

$t = 0$

Add self-edges

$M_t = \Delta^{-1}A$

Repeat

$t = t + 1$

$M_t = M_{t-1}.M_{t-1}$

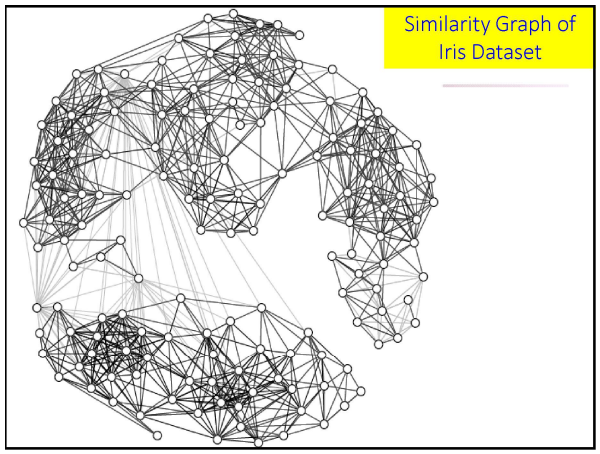
$M_t = \Psi(M_t, r)$

$$\|M_t - M_{t-1}\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}(m_{ij_t} - m_{ij_{t-1}})^2}$$

Untill $||M_t - M_{t-1}||_F \leq \varepsilon$ //*Frobenius norm*

G_t = directed graph induced by M_t

C = {Weakly connected components in G_t}



Clusters Obtained Using MCL