CS 521
Data Mining Techniques
Instructor: Abdullah Mueen

LECTURE 7: ADVANCED CLUSTERING (FUZZY AND CO-CLUSTERING)

Review: Basic Cluster Analysis Methods (Chap. 10)

Cluster Analysis: Basic Concepts
- Group data so that object similarity is high within clusters but low across clusters

Partitioning Methods
- K-means and k-medoids algorithms and their refinements

Hierarchical Methods
- Agglomerative and divisive method, Birch, Cameleon

Density-Based Methods
- DBScan, Optics and DenCLu

Grid-Based Methods
- STING and CLIQUE (subspace clustering)

Evaluation of Clustering
- Assess clustering tendency, determine # of clusters, and measure clustering quality

Outline of Advanced Clustering Analysis

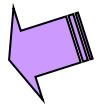
Probability Model-Based Clustering

- Each object may take a probability to belong to a cluster

Clustering Graphs and Network Data

- Similarity measurement and clustering methods for graph and networks

Chapter 11. Cluster Analysis: Advanced Methods



Probability Model-Based Clustering

Clustering High-Dimensional Data

Clustering Graphs and Network Data

Clustering with Constraints

Summary

4

Fuzzy Set and Fuzzy Cluster

Clustering methods discussed so far
- Every data object is assigned to exactly one cluster

Some applications may need for fuzzy or soft cluster assignment
- Ex. An e-game could belong to both entertainment and software

Methods: fuzzy clusters and probabilistic model-based clusters

Fuzzy cluster: A fuzzy set *S:* $F_S : X \rightarrow [0, 1]$ (value between 0 and 1)

Example: Popularity of cameras is defined as a fuzzy mapping

Camera	Sales (units)
A	50
B	1320
C	860
D	270

$$\text{Pop}(o) = \begin{cases} 1 & \text{if } 1,000 \text{ or more units of } o \text{ are sold} \\ \frac{i}{1000} & \text{if } i \ (i < 1000) \text{ units of } o \text{ are sold} \end{cases}$$

Then, *A*(0.05), *B*(1), *C*(0.86), *D*(0.27)

Fuzzy (Soft) Clustering

Review-id	Keywords
R_1	digital camera, lens
R_2	digital camera
R_3	lens
R_4	digital camera, lens, computer
R_5	computer, CPU
R_6	computer, computer game

Example: Let cluster features be
- C_1 :"digital camera" and "lens"
- C_2: "computer"

Fuzzy clustering
- k fuzzy clusters C_1, …,C_k ,represented as a partition matrix M = [w_{ij}]
- P1: for each object o_i and cluster C_j, $0 \le w_{ij} \le 1$ (fuzzy set)
- P2: for each object o_i, $\sum_{i=1}^{k} w_{ij} = 1$, equal participation in the clustering
- P3: for each cluster C_j , $0 < \sum_{i=1}^{n} w_{ij} < n$, ensures there is no empty cluster

$$M = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \frac{2}{3} & \frac{1}{3} \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Let c_1, …, c_k as the center of the k clusters

For an object o_i, sum of the squared error (SSE), p is a parameter: $\mathrm{SSE}(o_i) = \sum_{j=1}^{k} w_{ij}^p dist(o_i, c_j)^2$

For a cluster C_i, SSE: $\mathrm{SSE}(C_j) = \sum_{i=1}^{n} w_{ij}^p dist(o_i, c_j)^2$

Measure how well a clustering fits the data: $\mathrm{SSE}(\mathcal{C}) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ij}^p dist(o_i, c_j)^2$

The EM (Expectation Maximization) Algorithm

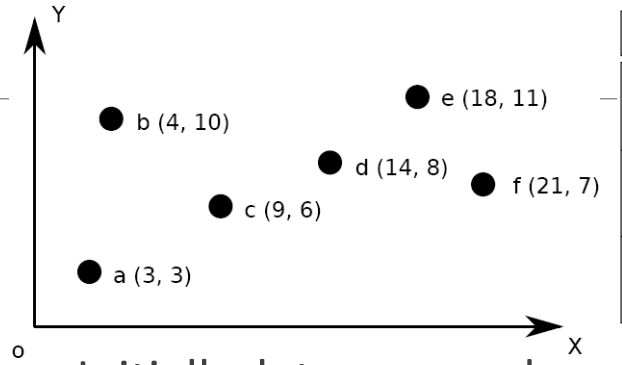
The k-means algorithm has two steps at each iteration:

- **Expectation Step** (E-step): Given the current cluster centers, each object is assigned to the cluster whose center is closest to the object: An object is *expected to belong to the closest cluster*
- **Maximization Step** (M-step): Given the cluster assignment, for each cluster, the algorithm *adjusts the center* so that *the sum of distance* from the objects assigned to this cluster and the new center is minimized

The (EM) algorithm: A framework to approach maximum likelihood or maximum a posteriori estimates of set membership in fuzzy clustering.

- **E-step** assigns objects to clusters according to the current fuzzy clustering or parameters of probabilistic clusters
- **M-step** finds the new clustering or parameters that maximize the sum of squared error (SSE) or the expected likelihood

Fuzzy Clustering Using the EM Algorithm



Iteration	E-step	M-step
1	$M^T = \begin{bmatrix} 1 & 0 & 0.48 & 0.42 & 0.41 & 0.47 \\ 0 & 1 & 0.52 & 0.58 & 0.59 & 0.53 \end{bmatrix}$	$c_1 = (8.47, 5.12)$, $c_2 = (10.42, 8.99)$
2	$M^T = \begin{bmatrix} 0.73 & 0.49 & 0.91 & 0.26 & 0.33 & 0.42 \\ 0.27 & 0.51 & 0.09 & 0.74 & 0.67 & 0.58 \end{bmatrix}$	$c_1 = (8.51, 6.11)$, $c_2 = (14.42, 8.69)$
3	$M^T = \begin{bmatrix} 0.80 & 0.76 & 0.99 & 0.02 & 0.14 & 0.23 \\ 0.20 & 0.24 & 0.01 & 0.98 & 0.86 & 0.77 \end{bmatrix}$	$c_1 = (6.40, 6.24)$, $c_2 = (16.55, 8.64)$

Initially, let c_1 = a and c_2 = b

1st E-step: assign o to c_1,w. wt =

- $w_{c,c_1} = \frac{41}{45+41} = 0.48$

$$\frac{\frac{1}{dist(o,c_1)^2}}{\frac{1}{dist(o,c_1)^2} + \frac{1}{dist(o,c_2)^2}} = \frac{dist(o,c_2)^2}{dist(o,c_1)^2 + dist(o,c_2)^2}$$

- 1st M-step: recalculate the centroids according to the partition matrix, minimizing the sum of squared error (SSE)

$$c_j = \frac{\sum_{\text{each point } o} w_{o,c_j}^2 o}{\sum_{\text{each point } o} w_{o,c_j}^2} \quad c_1 = \left(\frac{1^2\times3+0^2\times4+0.48^2\times9+0.42^2\times14+0.41^2\times18+0.47^2\times21}{1^2+0^2+0.48^2+0.42^2+0.41^2+0.47^2}, \frac{1^2\times3+0^2\times10+0.48^2\times6+0.42^2\times8+0.41^2\times11+0.47^2\times7}{1^2+0^2+0.48^2+0.42^2+0.41^2+0.47^2}\right) = (8.47, 5.12)$$

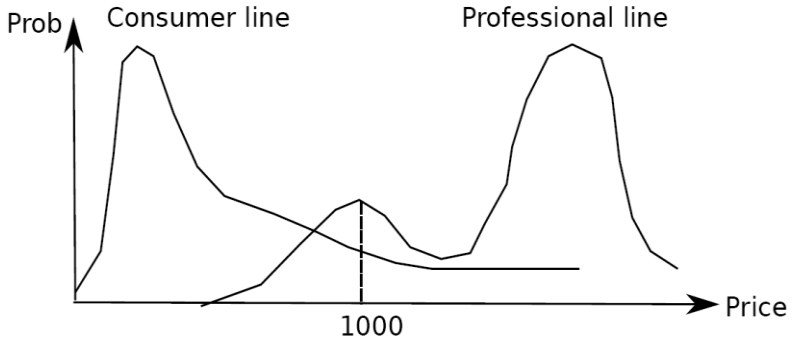
- Iteratively calculate this until the cluster centers converge or the change is small enough

Probabilistic Model-Based Clustering

Cluster analysis is to find hidden categories.

A hidden category (i.e., *probabilistic cluster)* is a distribution over the data space, which can be mathematically represented using a probability density function (or distribution function).

- Ex. 2 categories for digital cameras sold
  - consumer line vs. professional line
  - density functions f_1, f_2 for C_1, C_2
  - obtained by probabilistic clustering



- A **mixture model** assumes that a set of observed objects is a mixture of instances from multiple probabilistic clusters, and conceptually each observed object is generated independently
- **Our task**: infer a set of *k* probabilistic clusters that is mostly likely to generate *D* using the above data generation process

Model-Based Clustering

A set *C* of *k* probabilistic clusters $C_1, \ldots, C_k$ with probability density functions $f_1, \ldots, f_k$, respectively, and their probabilities $\omega_1, \ldots, \omega_k$.

Probability of an object *o* generated by cluster C_j is

$$P(o|C_j) = \omega_j f_j(o)$$

Probability of *o* generated by the set of cluster ***C*** is

$$P(o|\boldsymbol{C}) = \sum_{j=1}^{k} \omega_j f_j(o)$$

- Since objects are assumed to be generated independently, for a data set D = {o_1, …, o_n}, we have,

$$P(D|\boldsymbol{C}) = \prod_{i=1}^{n} P(o_i|\boldsymbol{C}) = \prod_{i=1}^{n} \sum_{j=1}^{k} \omega_j f_j(o_i)$$

- Task: Find a set *C* of *k* probabilistic clusters s.t. $P(D|\boldsymbol{C})$ is maximized
- However, maximizing $P(D|\boldsymbol{C})$ is often intractable since the probability density function of a cluster can take an arbitrarily complicated form
- To make it computationally feasible (as a compromise), assume the probability density functions being some parameterized distributions

Univariate Gaussian Mixture Model

$O = \{o_1, \ldots, o_n\}$ (n observed objects), $\Theta = \{\theta_1, \ldots, \theta_k\}$ (parameters of the k distributions), and $P_j(o_i | \theta_j)$ is the probability that o_i is generated from the j-th distribution using parameter θ_j, we have

$$P(o_i|\Theta) = \sum_{j=1}^{k} \omega_j P_j(o_i|\Theta_j) \qquad P(\mathbf{O}|\Theta) = \prod_{i=1}^{n} \sum_{j=1}^{k} \omega_j P_j(o_i|\Theta_j)$$

- Univariate Gaussian mixture model
  - Assume the probability density function of each cluster follows a 1-d Gaussian distribution. Suppose that there are k clusters.
  - The probability density function of each cluster are centered at μ_j with standard deviation σ_j, θ_j, = (μ_j, σ_j), we have

$$P(o_i|\Theta_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i-\mu_j)^2}{2\sigma^2}} \qquad P(o_i|\Theta) = \sum_{j=1}^{k} \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i-\mu_j)^2}{2\sigma^2}}$$

$$P(\mathbf{O}|\Theta) = \prod_{i=1}^{n} \sum_{j=1}^{k} \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i-\mu_j)^2}{2\sigma^2}}$$

Computing Mixture Models with EM

Given n objects O = {o_1, …, o_n}, we want to mine a set of parameters Θ = {θ_1, …, θ_k} s.t.,P(**O**|**Θ**) is maximized, where $\theta_j = (\mu_j, \sigma_j)$ are the mean and standard deviation of the j-th univariate Gaussian distribution

We initially assign random values to parameters θ_j, then iteratively conduct the E- and M- steps until converge or sufficiently small change

At the E-step, for each object o_i, calculate the probability that o_i belongs to each distribution,

$$P(\Theta_j|o_i, \mathbf{\Theta}) = \frac{P(o_i|\Theta_j)}{\sum_{l=1}^{k} P(o_i|\Theta_l)}$$

- At the M-step, adjust the parameters $\theta_j = (\mu_j, \sigma_j)$ so that the expected likelihood P(**O**|**Θ**) is maximized

$$\mu_j = \sum_{i=1}^{n} o_i \frac{P(\Theta_j|o_i, \mathbf{\Theta})}{\sum_{l=1}^{n} P(\Theta_j|o_l, \mathbf{\Theta})} = \frac{\sum_{i=1}^{n} o_i P(\Theta_j|o_i, \mathbf{\Theta})}{\sum_{i=1}^{n} P(\Theta_j|o_i, \mathbf{\Theta})} \qquad \sigma_j = \sqrt{\frac{\sum_{i=1}^{n} P(\Theta_j|o_i, \mathbf{\Theta})(o_i - u_j)^2}{\sum_{i=1}^{n} P(\Theta_j|o_i, \mathbf{\Theta})}}$$

Advantages and Disadvantages of Mixture Models

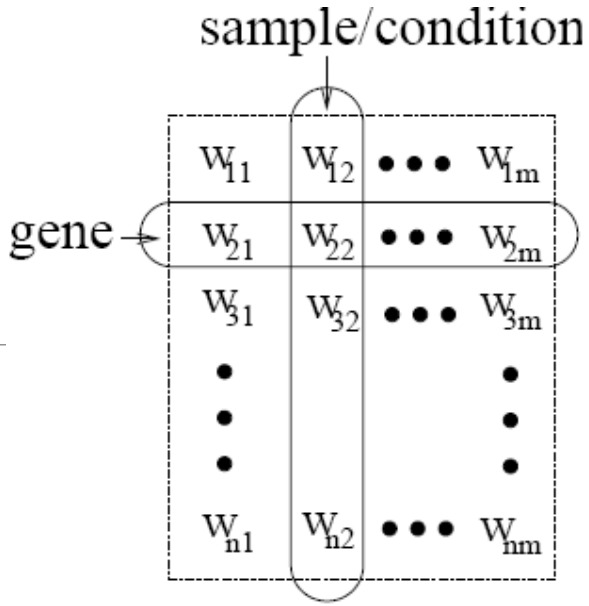
Strength

- Mixture models are more general than partitioning and fuzzy clustering
- Clusters can be characterized by a small number of parameters
- The results may satisfy the statistical assumptions of the generative models

Weakness

- Converge to local optimal (overcome: run multi-times w. random initialization)
- Computationally expensive if the number of distributions is large, or the data set contains very few observed data points
- Need large data sets
- Hard to estimate the number of clusters

Bi-Clustering Methods

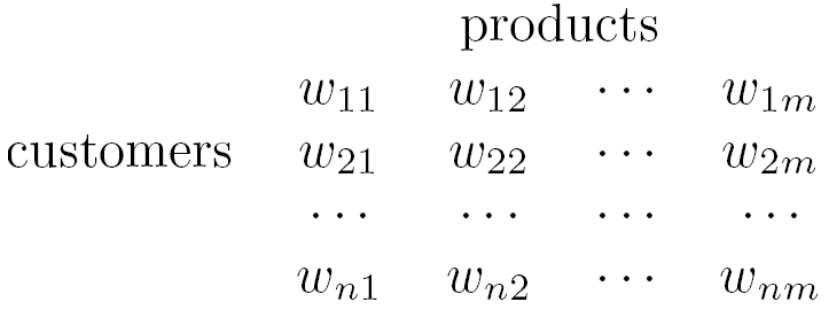


Bi-clustering: Cluster both objects and attributes simultaneously (treat objs and attrs in symmetric way)

Four requirements:

- Only a small set of objects participate in a cluster
- A cluster only involves a small number of attributes
- An object may participate in multiple clusters, or does not participate in any cluster at all
- An attribute may be involved in multiple clusters, or is not involved in any cluster at all

- Ex 1. *Gene expression* or *microarray data:* a gene sample/condition matrix.
  - Each element in the matrix, a real number, records the expression level of a gene under a specific condition
- Ex. 2. Clustering customers and products
  - Another bi-clustering problem



Types of Bi-clusters

Let A = {a_1, ..., a_n} be a set of genes, B = {b_1, …, b_n} a set of conditions

A bi-cluster: A submatrix where genes and conditions follow some consistent patterns

4 types of bi-clusters (ideal cases)

- Bi-clusters with constant values:
  - for any *i* in *I* and *j in J*, $e_{ij} = c$
- Bi-clusters with constant values on rows:
  - $e_{ij} = c + \alpha_i$
  - Also, it can be constant values on columns
- Bi-clusters with *coherent values* (aka. *pattern-based clusters*)
  - $e_{ij} = c + \alpha_i + \beta_j$
- Bi-clusters with *coherent* evolutions on rows
  - $(e_{i1j1} - e_{i1j2})(e_{i2j1} - e_{i2j2}) \geq 0$
  - i.e., only interested in the up- or down- regulated changes across genes or conditions without constraining on the exact values

10	10	10	10	10
20	20	20	20	20
50	50	50	50	50
0	0	0	0	0

10	50	30	70	20
20	60	40	80	30
50	90	70	110	60
0	40	20	60	10

10	50	30	70	20
20	100	50	1000	30
50	100	90	120	80
0	80	20	100	10

Bi-Clustering Methods

Real-world data is noisy: Try to find approximate bi-clusters

Methods: Optimization-based methods vs. enumeration methods

Optimization-based methods

- Try to find a submatrix at a time that achieves the best significance as a bi-cluster
- Due to the cost in computation, greedy search is employed to find local optimal bi-clusters
- Ex. δ-Cluster Algorithm (Cheng and Church, ISMB'2000)

Enumeration methods

- Use a tolerance threshold to specify the degree of noise allowed in the bi-clusters to be mined
- Then try to enumerate all submatrices as bi-clusters that satisfy the requirements
- Ex. δ-pCluster Algorithm (H. Wang et al.' SIGMOD'2002, MaPle: Pei et al., ICDM'2003)

Bi-Clustering for Micro-Array Data Analysis

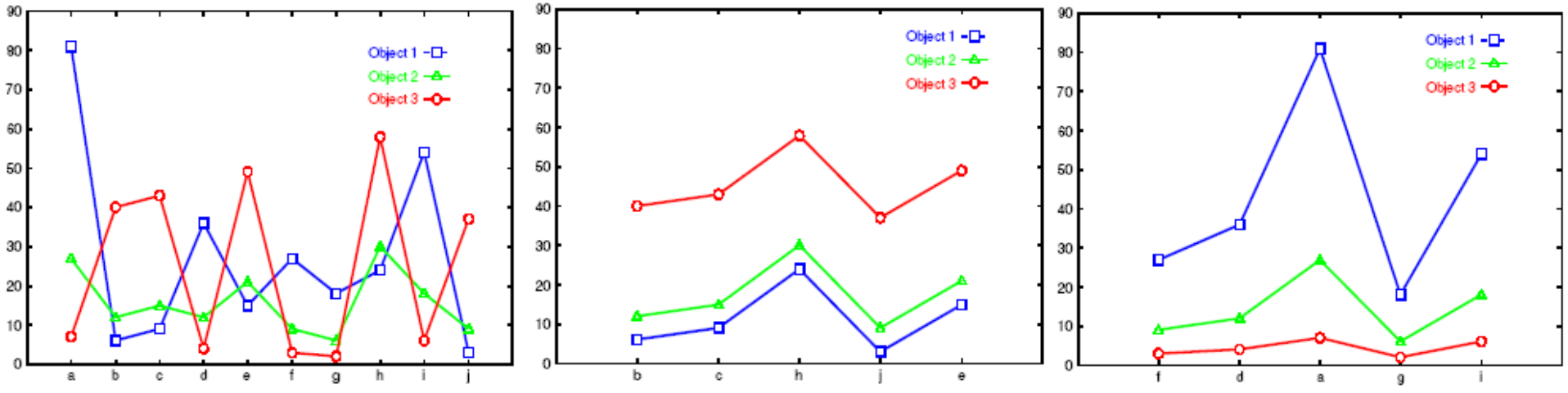
Left figure: Micro-array “raw” data shows 3 genes and their values in a multi-D space: Difficult to find their patterns

Right two: Some subsets of dimensions form nice shift and scaling patterns

No globally defined similarity/distance measure

Clusters may not be exclusive

- An object can appear in multiple clusters



```
clear all
L = diag(sum(M))-M;
[d v] = eig(L);
M = [zeros(16,16) A; A' zeros(16,16)];
L = diag(sum(M))-M;
[d v] = eig(L);
```

```
%shuffle
ind = randperm(16);
for i = 1:16
  for j = 1:16
     MM(i,j) = M(ind(i),ind(j));

  end
end
```

```
%make symmetric
for i = 1:16
   for j = 1:16
      M(i,j) = M(j,i);
   end
end
```