Lecture Notes for Chapter 8

Introduction to Data Mining
by
Tan, Steinbach, Kumar
What is Cluster Analysis?

- Finding groups of objects such that:
  - Objects in a group will be similar/related
  - Objects in different groups will be different/unrelated
  - Demo: 1 2

Intra-cluster distances are minimized

Inter-cluster distances are maximized
Clustering Helps Human to Understand

- Clustering as a learning process
- Group related documents for browsing
- Group genes and proteins that have similar functionality
- Group stocks with similar price fluctuations
- Group patients with similar type of depression

Clustering precipitation in Australia
Clustering for utility

- Good clustering has predictive power
- Clusters can be useful for communications (Cluster as compression)
- Efficiently finding nearest neighbors
- Failures of the cluster model highlight interesting objects that deserves special attention: anomaly detection
Road Map

- Clustering Overview
- K-means
- Agglomerative Hierarchical Clustering
- Scalable clustering
- Evaluations
- Other methods
  - Mixture models
  - Spectral clustering
Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

Four Clusters
Types of Clustering

- A clustering is a set of clusters

- Important distinction between **hierarchical** and **partitional** sets of clusters
  - Partitional Clustering
  - Hierarchical clustering
Partitional Clustering

Original Points

A Partitional Clustering
Hierarchical Clustering

Traditional Hierarchical Clustering

Non-traditional Hierarchical Clustering

Traditional Dendrogram

Non-traditional Dendrogram
Other Distinctions Between Sets of Clusters

- Exclusive versus non-exclusive
- Fuzzy versus non-fuzzy
- Partial versus complete
- Heterogeneous versus homogeneous
Types of Clusters

- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters
- Property or Conceptual
- Described by an Objective Function
Types of Clusters: Well-Separated

- Well-Separated Clusters:

3 well-separated clusters
Types of Clusters: Center-Based

- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster

4 center-based clusters
Types of Clusters: Contiguity-Based

- Contiguous Cluster (Nearest neighbor or Transitive)

8 contiguous clusters
Types of Clusters: Density-Based

- Density-based

6 density-based clusters
Types of Clusters: Conceptual Clusters

- Shared Property or Conceptual Clusters

2 Overlapping Circles
Types of Clusters: Objective Function

- Clusters Defined by an Objective Function
  - Finds clusters that minimize or maximize an objective function.
  - Can have global or local objectives.
  - A variation of the global objective function approach is to fit the data to a parameterized model.

- Map the clustering problem to a different domain and solve a related problem in that domain
  - Want to minimize the edge weight between clusters and maximize the edge weight within clusters
Characteristics of the Input Data Are Important

- Type of proximity or density measure
- Sparseness
- Attribute type
- Type of Data
- Dimensionality
- Noise and Outliers
- Type of Distribution
Clustering Algorithms

- K-means and its variants
- Hierarchical clustering
- Density-based clustering
**K-means Clustering**

Initial centroids are often chosen randomly.

‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.

1: Select $K$ points as the initial centroids.
2: repeat
3: Form $K$ clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: until The centroids don’t change

Most of the convergence happens in the first few iterations.

The centroid is (typically) the mean of the points in the cluster.
K-means Clustering – Complexity

- Complexity is $O( n \times K \times I \times d )$
  - $n =$ number of points,
  - $K =$ number of clusters,
  - $I =$ number of iterations,
  - $d =$ number of attributes

- Space complexity: $O((n+K)d)$
Two different K-means Clusterings

Original Points

Optimal Clustering

Sub-optimal Clustering
Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)

\[
SSE = \sum_{i=1}^{K} \sum_{x \in C_i} \text{dist}^2(m_i, x)
\]

- \(x\) is a data point in cluster \(C_i\)
- \(m_i\) is the representative point for cluster \(C_i\)
Importance of Choosing Initial Centroids

Iteration 6
Importance of Choosing Initial Centroids
Importance of Choosing Initial Centroids ...
Importance of Choosing Initial Centroids ...
Problems with Selecting Initial Points

- If there are $K$ ‘real’ clusters then the chance of selecting one centroid from each cluster is small.
  - Chance is relatively small when $K$ is large
  - If clusters are the same size, $n$, then

$$ P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K} $$

- For example, if $K = 10$, then probability $= 10!/10^{10} = 0.00036$
- Sometimes the initial centroids will readjust themselves in ‘right’ way, and sometimes they don’t
Starting with two initial centroids in one cluster of each pair of clusters
10 Clusters Example

Starting with two initial centroids in one cluster of each pair of clusters
10 Clusters Example

Starting with some pairs of clusters having three initial centroids, while other have only one.
Starting with some pairs of clusters having three initial centroids, while other have only one.
Solutions to Initial Centroids Problem

- Multiple runs
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
- Postprocessing
- Bisecting K-means
Handling Empty Clusters

- Choose the point that contributes most to SSE
- Choose a point from the cluster with the highest SSE
Updating Centers Incrementally

- In the basic K-means algorithm, centroids are updated after all points are assigned to a centroid.

- An alternative is to update the centroids after each assignment (incremental approach).
Pre-processing and Post-processing

- **Pre-processing**
  - Normalize the data
  - Eliminate outliers

- **Post-processing**
  - Eliminate small clusters
  - Split ‘loose’ clusters
  - Merge clusters that are ‘close’
  - Can use these steps during the clustering process
Bisecting K-means

- Bisecting K-means algorithm: producing hierarchical clustering

1: Initialize the list of clusters to contain the cluster containing all points.
2: repeat
3: Select a cluster from the list of clusters
4: for \( i = 1 \) to number_of_iterations do
5: Bisect the selected cluster using basic K-means
6: end for
7: Add the two clusters from the bisection with the lowest SSE to the list of clusters.
8: until Until the list of clusters contains \( K \) clusters
Bisecting K-means Example

Iteration 10
Limitations of K-means: Differing Sizes

Original Points

K-means (3 Clusters)
Limitations of K-means: Differing Density

Original Points

K-means (3 Clusters)
Limitations of K-means: Non-globular Shapes

Original Points

K-means (2 Clusters)
Overcoming K-means Limitations

Original Points

K-means Clusters
Overcoming K-means Limitations

Original Points

K-means Clusters
Overcoming K-means Limitations

Original Points

K-means Clusters
Hierarchical Clustering

- Can be visualized as a dendrogram
  - A tree-like diagram that records the sequences of merges or splits
Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)
Types of Hierarchical Clustering

- **Agglomerative:**
  - Start with the points as individual clusters
  - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left

- **Divisive:**
  - Start with one, all-inclusive cluster
  - At each step, split a cluster until each cluster contains a point (or there are k clusters)
Agglomerative Clustering Algorithm

- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. Repeat
  4. Merge the two closest clusters
  5. Update the proximity matrix
  6. Until only a single cluster remains

- Key operation is the computation of the proximity of two clusters
Starting Situation

- Start with clusters of individual points and a proximity matrix.

```
    p1  p2  p3  p4  p5  ...  
p1
p2
p3
p4
p5
...  Proximity Matrix
```
Intermediate Situation

- After some merging steps, we have some clusters

![Proximity Matrix diagram](image-url)
Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.
The question is “How do we update the proximity matrix?”
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity Matrix

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity Matrix

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity Matrix:

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity Matrix

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Cluster Similarity: MIN or Single Link

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

<table>
<thead>
<tr>
<th></th>
<th>I1</th>
<th>I2</th>
<th>I3</th>
<th>I4</th>
<th>I5</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>1.00</td>
<td>0.90</td>
<td>0.10</td>
<td>0.65</td>
<td>0.20</td>
</tr>
<tr>
<td>I2</td>
<td>0.90</td>
<td>1.00</td>
<td>0.70</td>
<td>0.60</td>
<td>0.50</td>
</tr>
<tr>
<td>I3</td>
<td>0.10</td>
<td>0.70</td>
<td>1.00</td>
<td>0.40</td>
<td>0.30</td>
</tr>
<tr>
<td>I4</td>
<td>0.65</td>
<td>0.60</td>
<td>0.40</td>
<td>1.00</td>
<td>0.80</td>
</tr>
<tr>
<td>I5</td>
<td>0.20</td>
<td>0.50</td>
<td>0.30</td>
<td>0.80</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Hierarchical Clustering: MIN

Nested Clusters

Dendrogram
Strength of MIN

Original Points

Two Clusters
Limitations of MIN

Original Points

Two Clusters
Cluster Similarity: MAX or Complete Linkage

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters

```
<table>
<thead>
<tr>
<th></th>
<th>I1</th>
<th>I2</th>
<th>I3</th>
<th>I4</th>
<th>I5</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>1.00</td>
<td>0.90</td>
<td>0.10</td>
<td>0.65</td>
<td>0.20</td>
</tr>
<tr>
<td>I2</td>
<td>0.90</td>
<td>1.00</td>
<td>0.70</td>
<td>0.60</td>
<td>0.50</td>
</tr>
<tr>
<td>I3</td>
<td>0.10</td>
<td>0.70</td>
<td>1.00</td>
<td>0.40</td>
<td>0.30</td>
</tr>
<tr>
<td>I4</td>
<td>0.65</td>
<td>0.60</td>
<td>0.40</td>
<td>1.00</td>
<td>0.80</td>
</tr>
<tr>
<td>I5</td>
<td>0.20</td>
<td>0.50</td>
<td>0.30</td>
<td>0.80</td>
<td>1.00</td>
</tr>
</tbody>
</table>
```
Hierarchical Clustering: MAX

Nested Clusters

Dendrogram
Strength of MAX

• Less susceptible to noise and outliers
Limitations of MAX

Original Points

Two Clusters

• Tends to break large clusters
• Biased towards globular clusters
Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

\[
\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{p_i \in \text{Cluster}_i, p_j \in \text{Cluster}_j} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| \times |\text{Cluster}_j|}
\]

<table>
<thead>
<tr>
<th>I1</th>
<th>I2</th>
<th>I3</th>
<th>I4</th>
<th>I5</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>1.00</td>
<td>0.90</td>
<td>0.10</td>
<td>0.65</td>
</tr>
<tr>
<td>I2</td>
<td>0.90</td>
<td>1.00</td>
<td>0.70</td>
<td>0.60</td>
</tr>
<tr>
<td>I3</td>
<td>0.10</td>
<td>0.70</td>
<td>1.00</td>
<td>0.40</td>
</tr>
<tr>
<td>I4</td>
<td>0.65</td>
<td>0.60</td>
<td>0.40</td>
<td>1.00</td>
</tr>
<tr>
<td>I5</td>
<td>0.20</td>
<td>0.50</td>
<td>0.30</td>
<td>0.80</td>
</tr>
</tbody>
</table>
Hierarchical Clustering: Group Average

Nested Clusters

Dendrogram
Hierarchical Clustering: Group Average

- Compromise between Single and Complete Link

- Strengths
  - Less susceptible to noise and outliers

- Limitations
  - Biased towards globular clusters
Cluster Similarity: Ward’s Method

- Similarity of two clusters is defined as the increase in squared error when two clusters are merged.
- Less susceptible to noise and outliers.
- Biased towards globular clusters.
- Hierarchical analogue of K-means.
Hierarchical Clustering: Comparison

MIN

MAX

Ward’s Method

Group Average
Hierarchical Clustering: Time and Space requirements

- \( O(N^2) \) space since it uses the proximity matrix.
  - \( N \) is the number of points.

- \( O(N^3) \) time in many cases
  - There are \( N \) steps and at each step the size, \( N^2 \), proximity matrix must be updated and searched
  - Complexity can be reduced to \( O(N^2 \log(N)) \) time for some approaches
Hierarchical Clustering: Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone.

- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters
MST: Divisive Hierarchical Clustering

- Build MST (Minimum Spanning Tree)
MST: Divisive Hierarchical Clustering

- Use MST for constructing hierarchy of clusters

**Algorithm 7.5** MST Divisive Hierarchical Clustering Algorithm

1: Compute a minimum spanning tree for the proximity graph.
2: repeat
3: Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).
4: until Only singleton clusters remain
Cluster Validity

- How to evaluate the “goodness” of the resulting clusters?
  - But “clusters are in the eye of the beholder”!
Motivation for Evaluation: Clusters found in Random Data

Random Points

K-means

DBSCAN

Complete Link
Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - Internal /Unsupervised
  - External /Supervised
  - Relative: compare two different clustering results
Measuring Cluster Validity Via Correlation

- Two matrices
  - Proximity Matrix
  - “Incidence” Matrix

- Compute the correlation between the two matrices

- High correlation indicates that points that belong to the same cluster are close to each other.

- Not a good measure for some density or contiguity based clusters.
Measuring Cluster Validity Via Correlation

![Graph 1]

\[ \text{Corr} = -0.9235 \]

![Graph 2]

\[ \text{Corr} = -0.5810 \]
Using Similarity Matrix for Cluster Validation

- Order the similarity matrix with respect to cluster labels and inspect visually.
Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

DBSCAN
Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

K-means
Using Similarity Matrix for Cluster Validation

• Clusters in random data are not so crisp

Complete Link
Internal Measures: Sum of Square Error

- Clusters in more complicated figures aren’t well separated
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of clusters
Internal Measures: Cohesion and Separation

- Cluster Cohesion
- Cluster Separation
- Example: Squared Error
  - Cohesion is measured by the within cluster sum of squares (SSE)
    \[ WSS = \sum_{i} \sum_{x \in C_i} (x - m_i)^2 \]
  - Separation is measured by the between cluster sum of squares
    \[ BSS = \sum_{i} |C_i|(m - m_i)^2 \]
    - Where \(|C_i|\) is the size of cluster \(i\)
Internal Measures: Cohesion and Separation

- A proximity graph based approach can also be used for cohesion and separation.
BSS + WSS = Constant

K=1 cluster: 
\[ WSS = (1 - 3)^2 + (2 - 3)^2 + (4 - 3)^2 + (5 - 3)^2 = 10 \]
\[ BSS = 4 \times (3 - 3)^2 = 0 \]
\[ Total = 10 + 0 = 10 \]

K=2 clusters: 
\[ WSS = (1 - 1.5)^2 + (2 - 1.5)^2 + (4 - 4.5)^2 + (5 - 4.5)^2 = 1 \]
\[ BSS = 2 \times (3 - 1.5)^2 + 2 \times (4.5 - 3)^2 = 9 \]
\[ Total = 1 + 9 = 10 \]
# External Measures of Cluster Validity: Entropy and Purity

## Table 5.9. K-means Clustering Results for LA Document Data Set

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Entertainment</th>
<th>Financial</th>
<th>Foreign</th>
<th>Metro</th>
<th>National</th>
<th>Sports</th>
<th>Entropy</th>
<th>Purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>40</td>
<td>506</td>
<td>96</td>
<td>27</td>
<td>1.2270</td>
<td>0.7474</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>7</td>
<td>280</td>
<td>29</td>
<td>39</td>
<td>2</td>
<td>1.1472</td>
<td>0.7756</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>671</td>
<td>0.1813</td>
<td>0.9796</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>162</td>
<td>3</td>
<td>119</td>
<td>73</td>
<td>2</td>
<td>1.7487</td>
<td>0.4390</td>
</tr>
<tr>
<td>5</td>
<td>331</td>
<td>22</td>
<td>5</td>
<td>70</td>
<td>13</td>
<td>23</td>
<td>1.3976</td>
<td>0.7134</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>358</td>
<td>12</td>
<td>212</td>
<td>48</td>
<td>13</td>
<td>1.5523</td>
<td>0.5525</td>
</tr>
<tr>
<td>Total</td>
<td>354</td>
<td>555</td>
<td>341</td>
<td>943</td>
<td>273</td>
<td>738</td>
<td>1.1450</td>
<td>0.7203</td>
</tr>
</tbody>
</table>

**Entropy** For each cluster, the class distribution of the data is calculated first, i.e., for cluster \( j \) we compute \( p_{ij} \), the ‘probability’ that a member of cluster \( j \) belongs to class \( i \) as follows: 
\[
p_{ij} = \frac{m_{ij}}{m_j}
\]
where \( m_j \) is the number of values in cluster \( j \) and \( m_{ij} \) is the number of values of class \( i \) in cluster \( j \). Then using this class distribution, the entropy of each cluster \( j \) is calculated using the standard formula 
\[
e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}
\]
where the \( L \) is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., 
\[
e = \sum_{i=1}^{K} \frac{m_j}{m} e_j
\]
where \( m_j \) is the size of cluster \( j \), \( K \) is the number of clusters, and \( m \) is the total number of data points.

**Purity** Using the terminology derived for entropy, the purity of cluster \( j \), is given by 
\[
purity_j = \max p_{ij}
\]
and the overall purity of a clustering by 
\[
purity = \sum_{i=1}^{K} \frac{m_j}{m} \text{purity}_j.
\]
Final Comment on Cluster Validity

“The validation of clustering structures is the most difficult and frustrating part of cluster analysis.
Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.”

_Algorithms for Clustering Data_, Jain and Dubes
Machine Learning for Clustering

- Mixture Models
- Spectral Clustering
Mixture Models for Data Clustering

- **Key assumption**
  - The observed data points are generated by a fixed number of statistical models

- **Learning algorithm**
  - Find the set of statistical models that best fit the observed data distribution
  - Often resort to iterative algorithms such as Expectation-Maximization (EM) algorithm
Mixture Model for Doc Clustering

- A set of language models \( \Theta = \{ \theta_1, \theta_2, \ldots, \theta_K \} \)
  \[ \theta_i = \{ p(w_1 | \theta_i), p(w_2 | \theta_i), \ldots, p(w_V | \theta_i) \} \]
Mixture Model for Doc Clustering

- A set of language models \( \Theta = \{ \theta_1, \theta_2, \ldots, \theta_K \} \)
  \[
  \theta_i = \{ p(w_1 | \theta_i), p(w_2 | \theta_i), \ldots, p(w_V | \theta_i) \}
  \]

- Probability \( p(d = d_i) \)
  \[
  p(d = d_i) = \sum_{\theta_j} p(d = d_i, \theta = \theta_j)
  \]
Mixture Model for Doc Clustering

- A set of language models \( \Theta = \{ \theta_1, \theta_2, \ldots, \theta_K \} \)
  - \( \theta_i = \{ p(w_1 | \theta_i), p(w_2 | \theta_i), \ldots, p(w_V | \theta_i) \} \)

- Probability \( p(d = d_i) \)

\[
p(d = d_i) = \sum_{\theta_j} p(d = d_i, \theta = \theta_j) = \sum_{\theta_j} p(\theta = \theta_j) p(d = d_i | \theta = \theta_j)
\]
Mixture Model for Doc Clustering

- A set of language models \( \Theta = \{\theta_1, \theta_2, \ldots, \theta_K\} \)
  \( \theta_i = \{p(w_1 | \theta_i), p(w_2 | \theta_i), \ldots, p(w_V | \theta_i)\} \)

- Probability \( p(d = d_i) \)

\[
p(d = d_i) = \sum_{\theta_j} p(d = d_i, \theta = \theta_j) \\
= \sum_{\theta_j} p(\theta = \theta_j) p(d = d_i | \theta = \theta_j) \\
\propto \sum_{\theta_j} p(\theta = \theta_j) \prod_{k=1}^{V} \left[ p(w_k | \theta_j) \right]^{tf(w_k, d_i)}
\]
The likelihood function for a given document collection

\[ D = (d_1, d_2, \ldots, d_n) \]

\[ l(D; \Theta) = \sum_{i=1}^{n} \log p(d_i) = \sum_{i=1}^{n} \log \left[ \sum_{\theta_j} p(\theta = \theta_j) p(d = d_i \mid \theta = \theta_j) \right] \]

Finding the optimal language models that maximize the likelihood function

\[ \Theta^* = \arg \max_{\Theta} l(D; \Theta) \]
EM: Alternative Optimization

- **E-step**: fix language models $\Theta$, compute $z_{ij}$
- **M-step**: fix $z_{ij}$, compute language models $\Theta$

- $Z_{ij}$: the membership of document $i$ to cluster $j$
Further Issues

- How to decide the number of components?

- How to model documents with multiple topics?
  - Probabilistic latent semantic index (PLSI) (Hofmann, 2001)
  - Latent Dirichlet Allocation (LDA) (Blei et al., 2002)
Clustering Using Models

- Mixture Models
- **Spectral Clustering**
Limitation of Mixture Models

- The mixture models look for compact clustering structures
- Spectral Clustering: clustering structures with high connectivity
Graph Partition

- MinCut: bipartite graphs with minimal number of cut edges

CutSize = 2
2-way Spectral Graph Partitioning

- Weight matrix $W$
  - $w_{i,j}$: the weight between two vertices $i$ and $j$

- Membership vector $q$

$$q_i = \begin{cases} 
1 & i \in \text{Cluster } A \\
-1 & i \in \text{Cluster } B 
\end{cases}$$

$$q = \arg \min_{q \in [-1,1]^n} \text{CutSize}$$

$$\text{CutSize} = J = \frac{1}{4} \sum_{i,j} \left( q_i - q_j \right)^2 w_{i,j}$$
Normalized Cut (Shi & Malik, 1997)

- Minimum cut does not balance the size of bipartite graphs

\[ d_i = \sum_{j \in P} w_{i,j} \]

\[ J = \frac{s(A, B)}{d_A} + \frac{s(A, B)}{d_B} \]

\[ s(A, B) = \sum_{i \in A} \sum_{j \in B} w_{i,j}, \quad d_A = \sum_{i \in A} d_i, \quad d_B = \sum_{i \in B} d_i \]
Constraint Based Clustering
based on Jiawei Han
Constraint-Based Cluster Analysis?

- Need user feedback: Users know their applications the best
- Less parameters but more user-desired constraints, e.g., an ATM allocation problem: obstacle & desired clusters
A Classification of Constraints in Cluster Analysis

- Clustering in applications: desirable to have user-guided (i.e., constrained) cluster analysis
- Different constraints in cluster analysis:
  - Constraints on individual objects (do selection first)
    - Cluster on houses worth over $300K
  - Constraints on distance or similarity functions
    - Weighted functions, obstacles (e.g., rivers, lakes)
  - Constraints on the selection of clustering parameters
    - # of clusters, MinPts, etc.
  - User-specified constraints
    - Contain at least 500 valued customers and 5000 ordinary ones
  - Semi-supervised: giving small training sets
Clustering With Obstacle Objects

- K-medoids is more preferable since k-means may locate the ATM center in the middle of a lake
- Visibility graph and shortest path
- Triangulation and micro-clustering
- Two kinds of join indices (shortest-paths) worth pre-computation
  - VV index: indices for any pair of obstacle vertices
  - MV index: indices for any pair of micro-cluster and obstacle indices
An Example: Clustering With Obstacle Objects

*Not* Taking obstacles into account

Taking obstacles into account
Clustering with User-Specified Constraints

- Example: Locating $k$ delivery centers, each serving at least $m$ valued customers and $n$ ordinary ones.

- Proposed approach:
  - Find an initial “solution” by partitioning the data set into $k$ groups and satisfying user-constraints.
  - Iteratively refine the solution by micro-clustering relocation (e.g., moving $\delta \mu$-clusters from cluster $C_i$ to $C_j$) and “deadlock” handling (break the microclusters when necessary).
  - Efficiency is improved by micro-clustering.

- How to handle more complicated constraints?
  - E.g., having approximately same number of valued customers in each cluster?! — Can you solve it?
Cluster analysis groups objects based on their similarity and has wide applications.

Measure of similarity can be computed for various types of data.

Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, and model-based methods.

There are still lots of research issues on cluster analysis:
- Scalability (considerable progress has been made)
Middle Term Feedback for Instructor

- What you like about this class?
- What you don’t like about this class? Suggestion for the instructor to improve?
- What topics you hope the instructor can cover later (if any)?