Cluster Analysis

Jun Du
The University of Western Ontario
jdu43@uwo.ca
Outline

• Cluster Analysis Overview
• Partitioning Approach: K-means
• Density-based Approach: DBSCAN
• Hierarchical Approach: Agglomerative
• Cluster Evaluation
• Summary
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

Intra-cluster distances are minimized

Inter-cluster distances are maximized
Ambiguous Notion of Cluster

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
  – A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset

• Hierarchical clustering
  – A set of nested clusters organized as a hierarchical tree
Partitional Clustering

Original Points

A Partitional Clustering
Other Types of Clustering

• Exclusive vs non-exclusive
  – Non-exclusive: one point may belong to multiple clusters.

• Fuzzy vs non-fuzzy
  – Fuzzy: a point belongs to every cluster with some weight between 0 and 1
  – Weights must sum to 1 (weights represent probabilities)

• Partial vs complete
  – Partial: only part (not all) of the data are clustered

• .......
Typical Clustering Approaches

• Partitioning approach:
  – Construct various partitions
  – Typical methods: K-means, K-medoids, CLARANS

• Density-based approach:
  – Based on connectivity and density functions
  – Typical methods: DBSACN, OPTICS, DenClue

• Hierarchical approach:
  – Create a hierarchical decomposition of the set of data (or objects) using some criterion
  – Typical methods: Agglomerative, Diana, BIRCH, CAMELEON
Outline

• Cluster Analysis Overview
• Partitioning Approach: K-means
• Density-based Approach: DBSCAN
• Hierarchical Approach: Agglomerative
• Cluster Evaluation
• Summary
K-means Example (1)

Pick $k=3$ initial cluster centers (randomly)
Assign each point to the closest cluster center.
Move each cluster center to the mean of each cluster.
K-means Example (4)

Reassign points to the closest new cluster center

Q: Which points are reassigned?
K-means Example (5)

Three points have been changed
K-means Example (6)

re-compute cluster means
move cluster centers to cluster means
K-means Clustering

1) Pick a number \((K)\) of cluster centers

2) Assign every data point (example) to its nearest cluster center (based on some certain distance function)

3) Move each cluster center to the mean of its assigned data points

4) Repeat steps 2,3 until convergence (cluster centers don’t change any more)
Notes on K-means Clustering

• Initial centers are often chosen randomly.
  – Clusters produced vary from one run to another.
  – Multiple runs with random seeds are commonly applied.

• Distance can be measured by Euclidean distance, cosine similarity, correlation, etc.

• Often the stopping condition is changed to “Until relatively few points change clusters”
Different Initial Centers

Original Points

Optimal Clustering

Sub-optimal Clustering
Choosing Initial Centers (1)

Iteration 1

Iteration 2

Iteration 3

Iteration 4

Iteration 5

Iteration 6
Choosing Initial Centers (2)
Limitations of K-means (1)

• K-means has problems when the data contains outliers. For example:
  • $1, 3, 5, 7, 9$ Mean Center: 5
  • $1, 3, 5, 7, 1009$ Mean Center: 205

• Solutions: variants of K-means
  – K-medians: not affected by extreme values
    • $1, 3, 5, 7, 9$ Median Center: 5
    • $1, 3, 5, 7, 1009$ Median Center: 5
  – K-medoids: most representative data point in the cluster
  – ......
Limitations of K-means (2)

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
Differing Sizes

Original Points

K-means (3 Clusters)
Differing Density

Original Points

K-means (3 Clusters)
Non-globular Shapes

Original Points

K-means (2 Clusters)
Solutions

• One solution is to use many clusters
  — Find parts of clusters, but need to put together

• Or......, we have use another algorithm
Outline

• Cluster Analysis Overview
• Partitioning Approach: K-means
• Density-based Approach: DBSCAN
• Hierarchical Approach: Agglomerative
• Cluster Evaluation
• Summary
DBSCAN

• DBSCAN is a density-based algorithm.
  – Density = number of points within a specified radius (Eps)
  – A point is a core point if it has more than a specified number of points (MinPts) within Eps
    • These are points that are at the interior of a cluster
  – A border point has fewer than MinPts within Eps, 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.
Core, Border, and Noise Points
DBSCAN Algorithm*

1. Label all points as core, border, or noise points.
2. Eliminate noise points.
3. Put an edge between all core points that are within Eps of each other.
4. Make each group of connected core points into a separate cluster.
5. Assign each border point to one of the clusters of its associated core points.

* This algorithm uses the same concepts and finds the same clusters as the original DBSCAN, but is optimized for simplicity.
Core, Border and Noise Points

Original Points

Point types: core, border and noise

Eps = 10, MinPts = 4
When DBSCAN Works Well

- Resistant to Noise
- Can handle clusters of different shapes and sizes
When DBSCAN Does NOT Work Well

- varying densities
- High-dimensional data

Original Points

(MinPts=4, Eps=9.75).

(MinPts=4, Eps=9.92)
Determining $EPS$ and $MinPts$

- Parameters might be less intuitive to set. Given $MinPts = k$, can we properly set $EPS$?
- For the points in the clusters (core points), their $k$th nearest neighbors are at roughly the same distance ($k$-dist).
- Noise points have the $k$th nearest neighbor at farther distance
- So, plot sorted $k$-dist for each data point (ascending order).
Demonstration

- **Demo**
- **SimpleKMeans**
  - Document_Slip (seed=10 vs seed=0)
  - Image_Extract (seed=10 vs seed=0)
- **DBScan** (epsilon=0.08, minPoints=3)
  - Image_Extract
  - Mars
  - Web_Log
  - Document_Slip
Outline

• Cluster Analysis Overview
• Partitioning Approach: K-means
• Density-based Approach: DBSCAN
• Hierarchical Approach: Agglomerative
• Cluster Evaluation
• Summary
Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- 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 dendrogram at the proper level

• They may correspond to meaningful taxonomies
  – Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, ...)
Hierarchical Clustering

- Two main types of hierarchical clustering
  - Agglomerative (bottom up):
    - 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 (top down):
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are \( k \) clusters)

- Algorithms maintain similarity or distance matrix (proximity matrix) to record the distance between clusters
  - Merge or split one cluster at a time
Agglomerative Clustering Algorithm

• More popular hierarchical clustering technique

• Basic algorithm is straightforward
  1. Let each data point be a cluster
  2. Compute the proximity matrix
  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 clusters
  – Different approaches to defining the distance between clusters distinguish the different algorithms
Starting Situation

- Start with clusters of individual points and a 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>

Proximity Matrix
Intermediate Situation

• After some merging steps, we have some clusters

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Proximity Matrix

```
p1  p2  p3  p4  p9  p10 p11 p12
```
Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.
After Merging

- The question is “How do we update the proximity matrix?”

```
<table>
<thead>
<tr>
<th></th>
<th>C2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>?</td>
<td>C5</td>
</tr>
<tr>
<td>C2 U C5</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>C3</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>
```
Inter-Cluster Similarity / Distance

- MIN
- MAX
- Group Average
- Distance Between Centers
- Other methods driven by an objective function

Proximity Matrix
Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centers
- Other methods driven by an objective function

Inter-Cluster Similarity / Distance

<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>

.  Proximity Matrix
Inter-Cluster Similarity / Distance

• MIN
• MAX
• Group Average
• Distance Between Centers
• Other methods driven by an objective function

Proximity Matrix

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

...
Inter-Cluster Similarity / Distance

- MIN
- MAX
- Group Average
- Distance Between Centers
- Other methods driven by an objective function

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>

Proximity Matrix
Inter-Cluster Similarity / Distance

- MIN
- MAX
- Group Average
- Distance Between Centers
- Other methods driven by an objective function

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>
<tr>
<td></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>
<tr>
<td></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>

Proximity Matrix
Hierarchical Clustering: Comparison

MIN

MAX

Group Average
Limitations

• Once a decision is made to combine two clusters, it cannot be undone
• No objective function is directly minimized
• Different schemes have problems with one or more of the following:
  – Sensitivity to noise and outliers
  – Difficulty handling different sized clusters
  – ......
Demonstration

- Demo
- HierarchicalClusterer
  - \Weka\Weather
Outline

• Cluster Analysis Overview
• Partitioning Approach: K-means
• Density-based Approach: DBSCAN
• Hierarchical Approach: Agglomerative
  • Cluster Evaluation
• Summary
Cluster Evaluation

• For supervised classification we have a variety of measures to evaluate how good our model is
  – Accuracy, precision, recall, etc.

• For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?

• But “clusters are in the eye of the beholder”!

• Then why do we want to evaluate them?
  – To avoid finding patterns in noise
  – To compare clustering algorithms
  – To compare two sets of clusters
  – To compare two clusters
Aspects of Cluster Evaluation

1. Assessing clustering tendency
   - Determining if a data set has clusters

2. Measuring clustering quality
   - Unsupervised
   - Supervised

3. Determining the number of clusters
Clustering Tendency

• How do we determine if a given dataset has meaningful clusters?
  – Try to cluster it
• No patterns (clusters) should be discovered if data is randomly distributed.
  – However, ......

K-means

DBSCAN

Random Points
Hopkins Statistic

• Assess if non-random structure exists in the data

• Hopkins Statistic

  – Given a dataset $D$, determine how far away $D$ is from being randomly distributed in the data space

  1. Sample $n$ points, $p_1, \ldots, p_n$, randomly from data space. For each $p_i$, find its nearest neighbor in $D$: $x_i = \min\{\text{dist}(p_i, v)\}$ where $v$ in $D$

  2. Sample $n$ points, $q_1, \ldots, q_n$, from $D$. For each $q_i$, find its nearest neighbor in $D - \{q_i\}$: $y_i = \min\{\text{dist}(q_i, v)\}$ where $v$ in $D$ and $v \neq q_i$

  3. Calculate the Hopkins Statistic:

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

  – If $D$ is randomly distributed, $\Sigma x_i$ and $\Sigma y_i$ will be close to each other and $H$ is close to 0.5. If $D$ is clustered, $H$ is close to 0
Measuring Clustering Quality

• Two methods: Supervised vs. unsupervised
  • Supervised (extrinsic): i.e., the ground truth is available
    – Compare a clustering against the ground truth
    – E.g., precision, recall
  • Unsupervised (intrinsic): i.e., the ground truth is unavailable
    – how well the clusters are separated, and how compact the clusters are
    – E.g., Sum of Squared Error (SSE), Silhouette coefficient
Unsupervised Measuring --- Cohesion

• Basic idea: data points within a cluster are expected to be as close as possible to each other --- cohesion

• Most common measure: Sum of Squared Error (SSE)

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

– \( x \) is a data point in cluster \( C_i \) and \( m_i \) is the center of \( C_i \)
Unsupervised Measuring --- Separation

• Basic idea: data points from different clusters are expected to be as far away as possible from each other --- separation

• Common measure: BSS (between cluster sum of squares)†

\[
BSS = \sum_{i=1}^{K} |C_i| \text{dist}^2 (m, m_i)
\]

– \( m \) is the center of all data points, \( m_i \) is the center of \( C_i \), and \( |C_i| \) is the size of \( C_i \)

† In contrast, SSE is also called WSS (Within cluster Sum of Squares)
Cohesion and Separation: Example

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: \[ WSS = (1 - 1.5)^2 + (2 - 1.5)^2 + (4 - 4.5)^2 + (5 - 4.5)^2 = 1 \]
\[ BSS = 2 \times (3 - 1.5)^2 + 2 \times (4.5 - 3)^2 = 9 \]
\[ Total = 1 + 9 = 10 \]

- BSS + WSS = constant
Silhouette Coefficient

• Silhouette Coefficient combines both cohesion and separation.

• For an individual point, $i$
  
  – Calculate $a =$ average distance of $i$ to the points in its cluster.
  
  – Calculate $b =$ min (average distance of $i$ to points in another cluster).
  
  – The silhouette coefficient for a point is then given by
    
    $s = 1 - \frac{a}{b}$ if $a < b$, (or $s = \frac{b}{a} - 1$ if $a \geq b$, not the usual case).
  
  – Typically between 0 and 1.
  
  – The closer to 1 the better.

• An overall measure of a clustering can be obtained by computing the average silhouette coefficient of all points.
Determine # Clusters

• Empirical method
  – # of clusters $\approx \sqrt{\frac{n}{2}}$ for a data set of $n$ points

• Elbow method
  – Use the turning point in the curve of within cluster sum of squares ($SSE$, $WSS$) w.r.t. the # of clusters
  – Use the turning point in the curve of silhouette coefficient w.r.t the # of clusters
Elbow Method
SSE v.s. Silhouette Coefficient
Note on Cluster Evaluation

• Cluster evaluation can strongly depend on the application domain
  – An intuitive way to evaluate the clustering: check if the produced clusters make sense in the domain.
Summary

• Three clustering algorithms:
  – K-means: most common one
  – DBSCAN: density-based clustering
  – Agglomerative: hierarchical clustering

• Cluster evaluation:
  – Cluster tendency: Hopkins statistic
  – Measuring Clustering Quality:
    • Cohesion: intra-cluster distances
    • Separation: inter-cluster distances
    • Silhouette Coefficient: combination of the two
  – Determine # Clusters: Elbow method