DATA MINING - 1DL105, 1DI111

Fall 2007

An introductory class in data mining

http://user.it.uu.se/~udbl/dm-ht2007/
alt. http://www.it.uu.se/edu/course/homepage/infoutv/ht07

Kjell Orsborn
Uppsala Database Laboratory
Department of Information Technology, Uppsala University,
Uppsala, Sweden
Data Mining
Cluster analysis: basic concepts and algorithms

(Tan, Steinbach, Kumar ch. 8)

Kjell Orsborn
Department of Information Technology
Uppsala University, Uppsala, Sweden
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
Applications of Cluster analysis

- **Understanding**
  - Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations

- **Summarization**
  - Reduce the size of large data sets

<table>
<thead>
<tr>
<th>Discovered Clusters</th>
<th>Industry Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Technology1 -DOWN</td>
</tr>
<tr>
<td>2</td>
<td>Technology2 -DOWN</td>
</tr>
<tr>
<td>3</td>
<td>Financial -DOWN</td>
</tr>
<tr>
<td>4</td>
<td>Oil-UP</td>
</tr>
</tbody>
</table>


Industry Group:
- Technology1 -DOWN
- Technology2 -DOWN
- Financial -DOWN
- Oil-UP

Clustering precipitation in Australia
What is not Cluster analysis?

• Supervised classification
  – Have class label information

• Simple segmentation
  – Dividing students into different registration groups alphabetically, by last name

• Results of a query
  – Groupings are a result of an external specification

• Graph partitioning
  – Some mutual relevance and synergy, but areas are not identical
Notion of a cluster can be ambiguous

- How many clusters?
- Six Clusters
- Two Clusters
- Four Clusters
Types of clusterings

- 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
Hierarchical clustering

Traditional Hierarchical Clustering

Non-traditional Hierarchical Clustering

Traditional Dendrogram

Non-traditional Dendrogram
Other distinctions between sets of clusters

• Exclusive versus non-exclusive
  – In non-exclusive clusterings, points may belong to multiple clusters.
  – Can represent multiple classes or ‘border’ points

• Fuzzy versus non-fuzzy
  – In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  – Weights must sum to 1
  – Probabilistic clustering has similar characteristics

• Partial versus complete
  – In some cases, we only want to cluster some of the data

• Heterogeneous versus homogeneous
  – Cluster of widely different sizes, shapes, and densities
Types of clusters: well-separated

- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.
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
  - The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most “representative” point of a cluster
Types of clusters: contiguity-based

• Contiguous Cluster (Nearest neighbor or Transitive)
  – A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.

8 contiguous clusters
Types of clusters: density-based

• **Density-based**
  – A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  – Used when the clusters are irregular or intertwined, and when noise and outliers are present.
Types of clusters: conceptual clusters

- Shared Property or Conceptual Clusters
  - Finds clusters that share some common property or represent a particular concept.
Types of clusters: objective function

- Clusters defined by an objective function
  - Finds clusters that minimize or maximize an objective function.
  - Enumerate all possible ways of dividing the points into clusters and evaluate the `goodness' of each potential set of clusters by using the given objective function. (NP Hard)
  - Can have global or local objectives.
    - Hierarchical clustering algorithms typically have local objectives
    - Partitional algorithms typically have global objectives
  - A variation of the global objective function approach is to fit the data to a parameterized model.
    - Parameters for the model are determined from the data.
    - Mixture models assume that the data is a ‘mixture' of a number of statistical distributions.
Types of clusters: objective function ...

- Map the clustering problem to a different domain and solve a related problem in that domain
  - Proximity matrix defines a weighted graph, where the nodes are the points being clustered, and the weighted edges represent the proximities between points
  - Clustering is equivalent to breaking the graph into connected components, one for each cluster.
  - 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
  – This is a derived measure, but central to clustering

• Sparseness
  – Dictates type of similarity
  – Adds to efficiency

• Type of data
  – Dictates type of similarity
  – Other characteristics, e.g., autocorrelation

• Dimensionality
• Noise and outliers
• Type of distribution
Clustering algorithms

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

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, \( K \), must be specified
- The basic algorithm is very simple

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
K-Means example

Given: a 1-dimensional value set \{2, 4, 10, 12, 3, 20, 30, 11, 25\}, \(k=2\)

Randomly assign means: \(m_1=3, m_2=4\)

- \(K_1=\{2, 3\}, K_2=\{4, 10, 12, 20, 30, 11, 25\}\), \(m_1=2.5, m_2=16\)
- \(K_1=\{2, 3, 4\}, K_2=\{10, 12, 20, 30, 11, 25\}\), \(m_1=3, m_2=18\)
- \(K_1=\{2, 3, 4, 10\}, K_2=\{12, 20, 30, 11, 25\}\), \(m_1=4.75, m_2=19.6\)
- \(K_1=\{2, 3, 4, 10, 11, 12\}, K_2=\{20, 30, 25\}\), \(m_1=7, m_2=25\)
- \(K_1=\{2, 3, 4, 10, 11, 12\}, K_2=\{20, 30, 25\}\) (no updates)

Stop as no updates of the clusters will occur in the next iteration resulting in stable means.
K-means clustering – details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to ‘Until relatively few points change clusters’
- 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
Two different K-means clusterings

Original Points

Optimal Clustering

Sub-optimal Clustering
Importance of choosing initial centroids
Importance of choosing initial centroids
Importance of choosing initial centroids …
Importance of choosing initial centroids …
Evaluating K-means clusters

- Most common measure is Sum of Squared Error (SSE) (also called Scatter)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

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

- \( x \) is a data point in cluster \( C_i \) and \( m_i \) is the representative point for cluster \( C_i \)
  - can show that the \( m_i \) that minimizes the SSE corresponds to the center (mean) of the cluster
- Given two clusters, we can choose the one with the smallest error
- One easy way to reduce SSE is to increase \( K \), the number of clusters
  - A good clustering with smaller \( K \) can have a lower SSE than a poor clustering with higher \( K \)
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 for example, if K = 10, then probability = \( \frac{10!}{10^{10}} = 0.00036 \)
  - Sometimes the initial centroids will readjust themselves in ‘right’ way, and sometimes they don’t
  - Consider an example of five pairs of clusters
10 clusters example

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
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**
  - Helps, but probability is not on your side
- **Sample and use hierarchical clustering to determine initial centroids**
- **Select more than k initial centroids and then select among these initial centroids**
  - Select most widely separated
- **Postprocessing**
- **Bisecting K-means**
  - Not as susceptible to initialization issues
Handling empty clusters

• Basic K-means algorithm can yield empty clusters

• Several strategies
  – Choose the point that contributes most to SSE
  – Choose a point from the cluster with the highest SSE
  – If there are several empty clusters, the above can be repeated several times.
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)
  – Each assignment updates zero or two centroids
  – Never get an empty cluster
  – Can use “weights” to change the impact
  – Introduces an order dependency
  – More expensive
Pre-processing and post-processing

• Pre-processing
  – Normalize the data
  – Eliminate outliers

• Post-processing
  – Eliminate small clusters that may represent outliers
  – Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  – Merge clusters that are ‘close’ and that have relatively low SSE
  – Can use these steps during the clustering process
Bisecting K-means

• Bisecting K-means algorithm
  – Variant of K-means that can produce a partitional or a 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

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

• K-means has problems when the data contains outliers.
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

One solution is to use many clusters.
Find parts of clusters, but need to put together.
Overcoming K-means limitations

Original Points

K-means Clusters
Overcoming K-means limitations

Original Points

K-means Clusters
DBSCAN (ch 8.4)

- **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.
DBSCAN: Core, Border, and Noise Points

- Core Point
- Border Point
- Noise Point

Eps = 1
MinPts = 4
DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

\[
\text{current}_\text{cluster}_\text{label} \leftarrow 1 \\
\text{for all core points do} \\
\quad \text{if the core point has no cluster label then} \\
\quad \quad \text{current}_\text{cluster}_\text{label} \leftarrow \text{current}_\text{cluster}_\text{label} + 1 \\
\quad \quad \text{Label the current core point with cluster label current}_\text{cluster}_\text{label} \\
\quad \text{end if} \\
\quad \text{for all points in the } Eps\text{-neighborhood, except } i^{th} \text{ the point itself do} \\
\quad \quad \text{if the point does not have a cluster label then} \\
\quad \quad \quad \text{Label the point with cluster label current}_\text{cluster}_\text{label} \\
\quad \quad \text{end if} \\
\quad \text{end for} \\
\text{end for}
\]
DBSCAN: 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

Original Points

- Varying densities
- High-dimensional data

(MinPts=4, Eps=9.75).

(MinPts=4, Eps=9.92)
DBSCAN: determining EPS and MinPts

- Idea is that for points in a cluster, their kth nearest neighbors are at roughly the same distance
- Noise points have the kth nearest neighbor at farther distance
- So, plot sorted distance of every point to its kth nearest neighbor
Hierarchical clustering (ch 8.3)

- 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 dendogram 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:
    - 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)

- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time
Agglomerative clustering algorithm

• More popular hierarchical clustering technique

• Basic algorithm is straightforward
  – Compute the proximity matrix
  – Let each data point be a cluster
  – Repeat
  – Merge the two closest clusters
  – Update the proximity matrix
  – Until only a single cluster remains

• Key operation is the computation of the proximity of two 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>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>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Proximity Matrix
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>C1</th>
<th>C5</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>C2 U</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td>?</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td>?</td>
<td></td>
<td>?</td>
</tr>
</tbody>
</table>

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

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

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>
</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

<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
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>
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>
Nested Clusters

Hierarchical clustering: MIN

Dendrogram
Strength of MIN

Original Points

Two Clusters

• Can handle non-elliptical shapes
Limitations of MIN

- Sensitive to noise and outliers
Cluster similarity: MAX or Complete linkage

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

Original Points  Two Clusters

• Less susceptible to noise and outliers
Limitations of MAX

- Tends to break large clusters
- Biased towards globular clusters

Original Points

Two Clusters
Cluster similarity: Group average

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

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

- Need to use average connectivity for scalability since total proximity favors large 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: 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 based on the increase in squared error when two clusters are merged
  - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
  - Can be used to initialize 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
- 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 and convex shapes
  - Breaking large clusters
MST: Divisive hierarchical clustering

- Build MST (Minimum Spanning Tree)
  - Start with a tree that consists of any point
  - In successive steps, look for the closest pair of points \((p, q)\) such that one point \(p\) is in the current tree but the other \(q\) is not
  - Add \(q\) to the tree and put an edge between \(p\) and \(q\)
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