Data mining: clustering

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

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

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
Data mining: clustering

Hierarchical Clustering

Traditional Hierarchical Clustering

Non-traditional Hierarchical Clustering

Traditional Dendrogram

Non-traditional Dendrogram

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:
  - 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.
Data mining: clustering

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.

![6 density-based clusters](image)

Types of Clusters: Conceptual Clusters

- Shared Property or Conceptual Clusters
  - Finds clusters that share some common property or represent a particular concept.

![2 Overlapping Circles](image)

Clustering Algorithms

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

![K-means Clustering](image)

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 Clustering – Details](image)

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](image)
Data mining: clustering

**Importance of Choosing Initial Centroids**

For each point, the error is the distance to the nearest cluster.

**Evaluating K-means Clusters**

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

\[
SSE = \sum_{i=1}^{n} \sum_{x \in C_i} dist^2 (m_i, x)
\]
- \(x\) is a data point in cluster \(C_i\) and \(m_i\) is the representative point for cluster \(C_i\)
- can show that \(m_i\) 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 \(K\)

**10 Clusters Example**

Starting with two initial centroids in one cluster of each pair of clusters
Data mining: clustering

10 Clusters Example

Iteration 4

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

From: Tan, Steinbach, Kumar, Introduction to Data Mining, McGraw Hill 2006

10 Clusters Example

Iteration 4

Starting with some pairs of clusters having three initial centroids, while other have only one.

From: Tan, Steinbach, Kumar, Introduction to Data Mining, McGraw Hill 2006

10 Clusters Example

Iteration 2

Starting with some pairs of clusters having three initial centroids, while other have only one.

From: Tan, Steinbach, Kumar, Introduction to Data Mining, McGraw Hill 2006

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.

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

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

Elena Baralis, Tania Cerquitelli
Politecnico di Torino
**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 SBC to the list of clusters.
8. until Until the list of clusters contains K clusters

**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)
Data mining: clustering

Overcoming K-means Limitations

One solution is to use many clusters, but need to put together.

From: Tan, Steinbach, Kumar, Introduction to Data Mining, McGraw Hill 2006

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

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

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, ...)

Elena Baralis, Tania Cerquitelli
Politecnico di Torino
**Agglomerative Clustering Algorithm**

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. **Repeat**
     1. Merge the two closest clusters
     2. Update the proximity matrix
  4. **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

**Intermediate Situation**

- After some merging steps, we have some clusters

**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?"

**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
Data mining: clustering

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

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>0.26</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

Dendogram
Data mining: clustering

**Strength of MIN**
- 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

**Hierarchical Clustering: MAX**
- Nested Clusters Dendrogram

**Strength of MAX**
- Less susceptible to noise and outliers

**Limitations of MAX**
- Tends to break large clusters
- Biased towards globular clusters

Elena Baralis, Tania Cerquitelli
Politecnico di Torino
Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.
  \[ \text{proximity(Cluster}_i, \text{Cluster}_j) = \frac{1}{|\text{Cluster}_i| \cdot |\text{Cluster}_j|} \sum_{p \in \text{Cluster}_i} \sum_{p' \in \text{Cluster}_j} \text{proximity}(p, p') \]
- Need to use average connectivity for scalability since total proximity favors large clusters

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</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

- 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: 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
**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.

**DBSCAN Algorithm**

- Eliminate noise points
- Perform clustering on the remaining points

```plaintext
for all core points do
    if the core point has no cluster label then
        Label the current core point with cluster label current_cluster_label
    end if
    for all points in the Eps-neighborhood, except the point itself do
        if the point does not have a cluster label then
            Label the point with cluster label current_cluster_label
        end if
    end for
end for
```

**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
### DBSCAN: Determining EPS and MinPts

- Idea is that for points in a cluster, their $k$th nearest neighbors are at roughly the same distance.
- Noise points have the $k$th nearest neighbor at farther distance.
- So, plot sorted distance of every point to its $k$th nearest neighbor.

### Measures of Cluster Validity

- The validation of clustering structures is the most difficult task.
- To evaluate the "goodness" of the resulting clusters, some numerical measures can be exploited.
- Numerical measures are classified into two main classes:
  - **External Index**: Used to measure the extent to which cluster labels match externally supplied class labels. Depending on this evaluation, different indices are calculated, for example, entropy, purity.
  - **Internal Index**: Used to measure the goodness of a clustering structure without respect to external information. Depending on this evaluation, different indices are calculated, for example, Sum of Squared Error (SSE), cluster cohesion, cluster separation, Rand-Index, adjusted rand-index.

### External Measures of Cluster Validity: Entropy and Purity

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Entropy</th>
<th>Purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>0.3</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>0.2</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>0.4</td>
<td>0.7</td>
</tr>
</tbody>
</table>

External measures for clustering entropy and purity.

### Internal Measures: Cohesion and Separation

#### Cluster Cohesion

Measures how closely related are objects in a cluster:

- Cohesion is measured by the within cluster sum of squares (SSE):
  \[
  WSS = \sum_{i=1}^{C} \sum_{x \in C_i} (x - m_i)^2
  \]

#### Cluster Separation

Measures how distinct or well-separated a cluster is from other clusters:

- Separation is measured by the between cluster sum of squares:
  \[
  BSS = \sum_{i=1}^{K} \sum_{x \notin C_i} \|m - m_i\|^2
  \]

### Internal Measures: Cohesion and Separation

- A proximity graph based approach can also be used for cohesion and separation.
- Cluster cohesion is the sum of the weight of all links within a cluster.
- Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.

### 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."