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.

Unsupervised Learning ✔

Supervised Learning = classification ✗
## Classification vs. Clustering

<table>
<thead>
<tr>
<th>Classification</th>
<th>Clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Supervised</td>
<td>- Unsupervised</td>
</tr>
<tr>
<td>- Human expert provides examples—assignment of documents to labels</td>
<td>- No human expert involved</td>
</tr>
<tr>
<td>- Have labels</td>
<td>- No labels</td>
</tr>
<tr>
<td>- Want “rules” to assign documents to labels</td>
<td>- Group documents into clusters using “distance”</td>
</tr>
</tbody>
</table>
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>

---

Clustering precipitation in Australia
For better navigation of search results

- For grouping search results thematically
  - clusty.com / Vivisimo
Google News: automatic clustering gives an effective news presentation metaphor
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

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

- **Heterogeneous versus homogeneous**
  - Cluster of widely different sizes, shapes, and densities
Types of Clusters

- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters
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.

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.
  - 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.

6 density-based clusters
More Formal Clustering Definition

- **Input**
  1. A set of documents: \( D = \{d_1, \ldots, d_N\} \)
  2. A desired number of clusters, \( K \)
  3. An objective function, \( f \), that evaluates the quality of a clustering (e.g., similarity/distance function)

- **Output**
  - Assignment of each document to
    - Hard Clustering: **One** of clusters
    - Soft Clustering: **One or many** clusters
  - \( D \rightarrow \{1, \ldots, K\} \)
Clustering Algorithms

- Partitional clustering
  - K-means and its variants

- Hierarchical 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
- Globally optimal

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
Centroid Example

- 2 dimensional space
  - p1: (5, 10)
  - p2: (10, 5)
  - p3: (10, 15)
- Centroid of p1, p2, and p3
  - \( \frac{1}{3} (5+10+10, 10+5+15) = (\frac{25}{3}, \frac{30}{3}) \)
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
K-Means Algorithm

- Randomly Initialize Clusters
- Assign data points to nearest clusters

Slide credit: Serafim Batzoglou
K-Means Algorithm

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate Clusters

Slide credit: Serafim Batzoglou
K-Means Algorithm

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate Clusters

Slide credit: Serafim Batzoglou
K-Means Algorithm

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate Clusters
- Repeat...

Slide credit: Serafim Batzoglou
K-Means Algorithm

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate Clusters
- Repeat…

Slide credit: Serafim Batzoglou
K-Means Algorithm

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate Clusters
- Repeat...until convergence

Slide credit: Serafim Batzoglou
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.

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

- \(x\) is a data point in cluster \(C_i\) and \(m_i\) is the representative point for cluster \(C_i\)
  - can show that \(m_i\) corresponds to the center (mean) of the cluster
- Given two clusters, we can choose the one with the smallest error
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
Seed Choice

- Results can vary based on random seed selection
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings
  - Select good seeds using a heuristic
  - Try out multiple starting points
  - Initialize with the results of another method.

In the above, if you start with B and E as centroids you converge to \{A,B,C\} and \{D,E,F\}
If you start with D and F you converge to \{A,B,D,E\} \{C,F\}
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.
  - More expensive.
  - Introduces an order dependency.
  - Never get an empty cluster.
  - Can use “weights” to change the impact.
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
Bisecting K-means

- **Bisecting K-means algorithm**
  - Variant of K-means that can produce a partitional or a hierarchical clustering

```plaintext
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
```
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)
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 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: HAC
    - 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)

- 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
  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
  - Different approaches to defining the distance between clusters distinguish the different algorithms
Closest pair of clusters

- Many variants to defining closest pair of clusters
- Eg, for document clustering, using “cosine” similarity
- Single-link
  - Similarity of the most cosine-similar (single-link)
- Complete-link
  - Similarity of the “furthest” points, the least cosine-similar
- Average-link
  - Average cosine between all pairs of elements
- Centroid
  - Clusters whose centroids (centers of gravity) are the most cosine-similar
Illustration: Single-Link

Max Similarity = Min pairwise Distance
Illustration: Complete-Link

Min Similarity = Max pairwise distance
Illustration: Average-Link

Average of all Similarities
Illustration: Centroid

C1

C2

C3

Average Inter-Similarity
Single Link Agglomerative Clustering

- Use maximum similarity of pairs:
  \[ \text{sim}(c_i, c_j) = \max_{x \in c_i, y \in c_j} \text{sim}(x, y) \]

- Can result in “straggly” (long and thin) clusters due to chaining effect.

- After merging \( c_i \) and \( c_j \), the similarity of the resulting cluster to another cluster, \( c_k \), is:
  \[ \text{sim}((c_i \cup c_j), c_k) = \max(\text{sim}(c_i, c_k), \text{sim}(c_j, c_k)) \]
Single Link Example
Complete Link

- Use minimum similarity of pairs:

$$sim(c_i, c_j) = \min_{x \in c_i \cup c_j, y \in c_j} sim(x, y)$$

- Makes “tighter,” spherical clusters that are typically preferable.

- After merging $c_i$ and $c_j$, the similarity of the resulting cluster to another cluster, $c_k$, is:

$$sim((c_i \cup c_j), c_k) = \min(sim(c_i, c_k), sim(c_j, c_k))$$
Complete Link Example
Group Average

- Similarity of two clusters = average similarity of all pairs within merged cluster.

\[
sim(c_i, c_j) = \frac{1}{|c_i \cup c_j| (|c_i \cup c_j| - 1)} \sum_{\bar{x} \in (c_i \cup c_j)} \sum_{\bar{y} \in (c_i \cup c_j): \bar{y} \neq \bar{x}} \text{sim}(\bar{x}, \bar{y})
\]

- Compromise between single and complete link.
- Two options:
  - Averaged across all ordered pairs in the merged cluster
  - Averaged over all pairs between the two original clusters
- No clear difference in efficacy
Measuring Cluster Validity Via Correlation

- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.

Corr = -0.9235

Corr = -0.5810
Using Similarity Matrix for Cluster Validation

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

DBSCAN
External Evaluation of Cluster Quality

- Simple measure: \textit{purity}, the ratio between the dominant class in the cluster $\pi_i$ and the size of cluster $\omega_i$

\[
Purity(\omega_i) = \frac{1}{n_i} \max_{j \in C} (n_{ij})
\]

- Biased because having $n$ clusters maximizes purity
Purity example

Cluster I: Purity = 1/6 (max(5, 1, 0)) = 5/6

Cluster II: Purity = 1/6 (max(1, 4, 1)) = 4/6

Cluster III: Purity = 1/5 (max(2, 0, 3)) = 3/5
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: $p_{ij} = m_{ij}/m_j$, 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., $\varepsilon = \sum_{j=1}^{K} \frac{m_j}{m} e_j$, 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 $\text{purity}_j = \max p_{ij}$ and the overall purity of a clustering by $\text{purity} = \sum_{j=1}^{K} \frac{m_j}{m} \text{purity}_j$. 
Bulk of slides on “clustering” are from the original ones by the authors of the DM textbook:
- http://www-users.cs.umn.edu/~kumar/dmbook/

A few slides are by the authors of the IR textbook: