Big Data Mining

Unsupervised Learning: Cluster Analysis

Min-Yuh Day, Ph.D.
Assistant Professor
Department of Information Management
Tamkang University

http://mail.tku.edu.tw/myday

2018-10-29
<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Subject/Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2018/09/10</td>
<td>Course Orientation for Big Data Mining</td>
</tr>
<tr>
<td>2</td>
<td>2018/09/17</td>
<td>ABC: AI, Big Data, Cloud Computing</td>
</tr>
<tr>
<td>3</td>
<td>2018/09/24</td>
<td>Mid-Autumn Festival (Day off)</td>
</tr>
<tr>
<td>4</td>
<td>2018/10/01</td>
<td>Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data</td>
</tr>
<tr>
<td>5</td>
<td>2018/10/08</td>
<td>Fundamental Big Data: MapReduce Paradigm, Hadoop and Spark Ecosystem</td>
</tr>
<tr>
<td>6</td>
<td>2018/10/15</td>
<td>Foundations of Big Data Mining in Python</td>
</tr>
<tr>
<td>7</td>
<td>2018/10/22</td>
<td>Supervised Learning: Classification and Prediction</td>
</tr>
<tr>
<td>8</td>
<td>2018/10/29</td>
<td>Unsupervised Learning: Cluster Analysis</td>
</tr>
<tr>
<td>9</td>
<td>2018/11/05</td>
<td>Unsupervised Learning: Association Analysis</td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Subject/Topics</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>10</td>
<td>2018/11/12</td>
<td>Midterm Project Report</td>
</tr>
<tr>
<td>11</td>
<td>2018/11/19</td>
<td>Machine Learning with Scikit-Learn in Python</td>
</tr>
<tr>
<td>12</td>
<td>2018/11/26</td>
<td>Deep Learning for Finance Big Data with TensorFlow</td>
</tr>
<tr>
<td>13</td>
<td>2018/12/03</td>
<td>Convolutional Neural Networks (CNN)</td>
</tr>
<tr>
<td>14</td>
<td>2018/12/10</td>
<td>Recurrent Neural Networks (RNN)</td>
</tr>
<tr>
<td>15</td>
<td>2018/12/17</td>
<td>Reinforcement Learning (RL)</td>
</tr>
<tr>
<td>16</td>
<td>2018/12/24</td>
<td>Social Network Analysis (SNA)</td>
</tr>
<tr>
<td>17</td>
<td>2018/12/31</td>
<td>Bridge Holiday (Extra Day Off)</td>
</tr>
<tr>
<td>18</td>
<td>2019/01/07</td>
<td>Final Project Presentation</td>
</tr>
</tbody>
</table>
Unsupervised Learning: Cluster Analysis
Outline

• Unsupervised Learning
• Cluster Analysis
• K-Means Clustering
# Data Mining Tasks and Machine Learning

<table>
<thead>
<tr>
<th>Data Mining Tasks &amp; Methods</th>
<th>Data Mining Algorithms</th>
<th>Learning Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prediction</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classification</td>
<td>Decision Trees, Neural Networks, Support Vector Machines, kNN, Naive Bayes, GA</td>
<td>Supervised</td>
</tr>
<tr>
<td>Regression</td>
<td>Linear/Nonlinear Regression, ANN, Regression Trees, SVM, kNN, GA</td>
<td>Supervised</td>
</tr>
<tr>
<td>Time Series</td>
<td>Autoregressive Methods, Averaging Methods, Exponential Smoothing, ARIMA</td>
<td>Supervised</td>
</tr>
<tr>
<td><strong>Association</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link analysis</td>
<td>Expectation Maximization, Apriori Algorithm, Graph-Based Matching</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>Sequence analysis</td>
<td>Apriori Algorithm, FP-Growth, Graph-Based Matching</td>
<td>Unsupervised</td>
</tr>
<tr>
<td><strong>Segmentation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clustering</td>
<td>k-means, Expectation Maximization (EM)</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>Outlier analysis</td>
<td>k-means, Expectation Maximization (EM)</td>
<td>Unsupervised</td>
</tr>
</tbody>
</table>

Unsupervised Learning: Cluster Analysis

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
Scikit-Learn Machine Learning Map

classification

- SVC
- Ensemble Classifiers
- Naive Bayes Classifier
- KNeighbors Classifier
- SGD Classifier
- Text Data
- Linear SVC

<100K samples

<10K samples

<10K categories known

>50 samples

>10K samples

Predicting a category

Do you have labeled data?

Predicting a quantity

Just looking

Predicting structure

Tough luck

Regression

- SGD Regressor
- Lasso
- ElasticNet
- RidgeRegression
- SVR(kernel='rbf')
- SVR(kernel='linear')

Clustering

- Spectral Clustering
- GMM
- KMeans
- MiniBatch KMeans
- MeanShift
- VBGMM

<10K samples

<10K samples

<10K samples

<10K samples

<10K samples

Scikit-Learn

scikit-learn
Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification
Identifying to which category an object belongs to.
Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ...

Regression
Predicting a continuous-valued attribute associated with an object.
Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ...

Clustering
Automatic grouping of similar objects into sets.
Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ...

Dimensionality reduction
Reducing the number of random variables to consider.
Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization.

Model selection
Comparing, validating and choosing parameters and models.
Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics.

Preprocessing
Feature extraction and normalization.
Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction.

Source: http://scikit-learn.org/
## Example of Cluster Analysis

<table>
<thead>
<tr>
<th>Point</th>
<th>P</th>
<th>P(x, y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>a</td>
<td>(3, 4)</td>
</tr>
<tr>
<td>p02</td>
<td>b</td>
<td>(3, 6)</td>
</tr>
<tr>
<td>p03</td>
<td>c</td>
<td>(3, 8)</td>
</tr>
<tr>
<td>p04</td>
<td>d</td>
<td>(4, 5)</td>
</tr>
<tr>
<td>p05</td>
<td>e</td>
<td>(4, 7)</td>
</tr>
<tr>
<td>p06</td>
<td>f</td>
<td>(5, 1)</td>
</tr>
<tr>
<td>p07</td>
<td>g</td>
<td>(5, 5)</td>
</tr>
<tr>
<td>p08</td>
<td>h</td>
<td>(7, 3)</td>
</tr>
<tr>
<td>p09</td>
<td>i</td>
<td>(7, 5)</td>
</tr>
<tr>
<td>p10</td>
<td>j</td>
<td>(8, 5)</td>
</tr>
</tbody>
</table>
### K-Means Clustering

<table>
<thead>
<tr>
<th>Point</th>
<th>P</th>
<th>P(x,y)</th>
<th>Dist to m1</th>
<th>Dist to m2</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>a</td>
<td>(3, 4)</td>
<td>1.95</td>
<td>3.78</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p02</td>
<td>b</td>
<td>(3, 6)</td>
<td>0.69</td>
<td>4.51</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p03</td>
<td>c</td>
<td>(3, 8)</td>
<td>2.27</td>
<td>5.86</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p04</td>
<td>d</td>
<td>(4, 5)</td>
<td>0.89</td>
<td>3.13</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p05</td>
<td>e</td>
<td>(4, 7)</td>
<td>1.22</td>
<td>4.45</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p06</td>
<td>f</td>
<td>(5, 1)</td>
<td>5.01</td>
<td>3.05</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p07</td>
<td>g</td>
<td>(5, 5)</td>
<td>1.57</td>
<td>2.30</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p08</td>
<td>h</td>
<td>(7, 3)</td>
<td>4.37</td>
<td>0.56</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p09</td>
<td>i</td>
<td>(7, 5)</td>
<td>3.43</td>
<td>1.52</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p10</td>
<td>j</td>
<td>(8, 5)</td>
<td>4.41</td>
<td>1.95</td>
<td>Cluster2</td>
</tr>
</tbody>
</table>

m1: (3.67, 5.83)  
m2: (6.75, 3.50)
Cluster Analysis
Cluster Analysis

• Used for automatic identification of **natural groupings** of things
• Part of the machine-learning family
• Employ **unsupervised learning**
• Learns the clusters of things from past data, then assigns new instances
• There is not an output variable
• Also known as **segmentation**

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems
Cluster Analysis

Clustering of a set of objects based on the *k-means method*. *(The mean of each cluster is marked by a “+”)*.

Source: Han & Kamber (2006)
Cluster Analysis

• Clustering results may be used to
  – Identify natural **groupings of customers**
  – Identify rules for assigning new cases to classes for targeting/diagnostic purposes
  – Provide characterization, definition, labeling of populations
  – Decrease the size and complexity of problems for other data mining methods
  – Identify **outliers** in a specific domain (e.g., rare-event detection)

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems
Example of Cluster Analysis

Point  P(in)  P(x,y)

p01  a  (3, 4)
p02  b  (3, 6)
p03  c  (3, 8)
p04  d  (4, 5)
p05  e  (4, 7)
p06  f  (5, 1)
p07  g  (5, 5)
p08  h  (7, 3)
p09  i  (7, 5)
p10  j  (8, 5)
Cluster Analysis for Data Mining

- Analysis methods
  - Statistical methods (including both hierarchical and nonhierarchical), such as $k$-means, $k$-modes, and so on
  - Neural networks (adaptive resonance theory [ART], self-organizing map [SOM])
  - Fuzzy logic (e.g., fuzzy c-means algorithm)
  - Genetic algorithms

- Divisive versus Agglomerative methods

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems
Cluster Analysis for Data Mining

• How many clusters?
  – There is not a “truly optimal” way to calculate it
  – Heuristics are often used
    1. Look at the sparseness of clusters
    2. Number of clusters = \((n/2)^{1/2}\) (n: no of data points)
    3. Use Akaike information criterion (AIC)
    4. Use Bayesian information criterion (BIC)

• Most cluster analysis methods involve the use of a distance measure to calculate the closeness between pairs of items
  – Euclidian versus Manhattan (rectilinear) distance

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems
**k-Means Clustering Algorithm**

- *k*: pre-determined number of clusters
- Algorithm *(Step 0: determine value of *k*)

**Step 1:** Randomly generate *k* random points as initial cluster centers

**Step 2:** Assign each point to the nearest cluster center

**Step 3:** Re-compute the new cluster centers

**Repetition step:** Repeat steps 2 and 3 until some convergence criterion is met (usually that the assignment of points to clusters becomes stable)

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems
Similarity

Distance
Similarity and Dissimilarity Between Objects

- **Distances** are normally used to measure the **similarity** or **dissimilarity** between two data objects.

- Some popular ones include: **Minkowski distance**:

  \[ d(i, j) = \sqrt[|q|]{|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \ldots + |x_{ip} - x_{jp}|^q} \]

  where \( i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \) and \( j = (x_{j1}, x_{j2}, \ldots, x_{jp}) \) are two \( p \)-dimensional data objects, and \( q \) is a positive integer.

- If \( q = 1 \), \( d \) is **Manhattan distance**

  \[ d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ip} - x_{jp}| \]

Source: Han & Kamber (2006)
• If $q = 2$, $d$ is Euclidean distance:

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)}$$

— Properties

• $d(i,j) \geq 0$
• $d(i,i) = 0$
• $d(i,j) = d(j,i)$
• $d(i,j) \leq d(i,k) + d(k,j)$

• Also, one can use weighted distance, parametric Pearson product moment correlation, or other disimilarity measures

Source: Han & Kamber (2006)
Euclidean distance vs Manhattan distance

• Distance of two points \( x_1 = (1, 2) \) and \( x_2 = (3, 5) \)

Euclidean distance:
\[
= \sqrt{(3-1)^2 + (5-2)^2} \\
= \sqrt{2^2 + 3^2} \\
= \sqrt{4 + 9} \\
= \sqrt{13} \\
= 3.61
\]

Manhattan distance:
\[
= |3-1| + |5-2| \\
= 2 + 3 \\
= 5
\]
The **K-Means Clustering Method**

- **Example**

  1. Arbitrarily choose \( K \) objects as initial cluster centers.
  2. Assign each object to the most similar center.
  3. Update the cluster means.
  4. Reassign objects to the nearest mean.
  5. Reassign objects to the nearest mean again.

Source: Han & Kamber (2006)
K-Means Clustering
## Example of Cluster Analysis

<table>
<thead>
<tr>
<th>Point</th>
<th>P</th>
<th>P(x,y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>a</td>
<td>(3, 4)</td>
</tr>
<tr>
<td>p02</td>
<td>b</td>
<td>(3, 6)</td>
</tr>
<tr>
<td>p03</td>
<td>c</td>
<td>(3, 8)</td>
</tr>
<tr>
<td>p04</td>
<td>d</td>
<td>(4, 5)</td>
</tr>
<tr>
<td>p05</td>
<td>e</td>
<td>(4, 7)</td>
</tr>
<tr>
<td>p06</td>
<td>f</td>
<td>(5, 1)</td>
</tr>
<tr>
<td>p07</td>
<td>g</td>
<td>(5, 5)</td>
</tr>
<tr>
<td>p08</td>
<td>h</td>
<td>(7, 3)</td>
</tr>
<tr>
<td>p09</td>
<td>i</td>
<td>(7, 5)</td>
</tr>
<tr>
<td>p10</td>
<td>j</td>
<td>(8, 5)</td>
</tr>
</tbody>
</table>
**K-Means Clustering**

**Step by Step**

<table>
<thead>
<tr>
<th>Point</th>
<th>P</th>
<th>P(x,y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>a</td>
<td>(3, 4)</td>
</tr>
<tr>
<td>p02</td>
<td>b</td>
<td>(3, 6)</td>
</tr>
<tr>
<td>p03</td>
<td>c</td>
<td>(3, 8)</td>
</tr>
<tr>
<td>p04</td>
<td>d</td>
<td>(4, 5)</td>
</tr>
<tr>
<td>p05</td>
<td>e</td>
<td>(4, 7)</td>
</tr>
<tr>
<td>p06</td>
<td>f</td>
<td>(5, 1)</td>
</tr>
<tr>
<td>p07</td>
<td>g</td>
<td>(5, 5)</td>
</tr>
<tr>
<td>p08</td>
<td>h</td>
<td>(7, 3)</td>
</tr>
<tr>
<td>p09</td>
<td>i</td>
<td>(7, 5)</td>
</tr>
<tr>
<td>p10</td>
<td>j</td>
<td>(8, 5)</td>
</tr>
</tbody>
</table>
**K-Means Clustering**

**Step 1: K=2, Arbitrarily choose K object as initial cluster center**

- **Initial** $m_1 = (3, 4)$
- **Initial** $m_2 = (8, 5)$

<table>
<thead>
<tr>
<th>Point</th>
<th>P</th>
<th>P(x,y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>a</td>
<td>(3, 4)</td>
</tr>
<tr>
<td>p02</td>
<td>b</td>
<td>(3, 6)</td>
</tr>
<tr>
<td>p03</td>
<td>c</td>
<td>(3, 8)</td>
</tr>
<tr>
<td>p04</td>
<td>d</td>
<td>(4, 5)</td>
</tr>
<tr>
<td>p05</td>
<td>e</td>
<td>(4, 7)</td>
</tr>
<tr>
<td>p06</td>
<td>f</td>
<td>(5, 1)</td>
</tr>
<tr>
<td>p07</td>
<td>g</td>
<td>(5, 5)</td>
</tr>
<tr>
<td>p08</td>
<td>h</td>
<td>(7, 3)</td>
</tr>
<tr>
<td>p09</td>
<td>i</td>
<td>(7, 5)</td>
</tr>
<tr>
<td>p10</td>
<td>j</td>
<td>(8, 5)</td>
</tr>
</tbody>
</table>
Step 2: Compute seed points as the centroids of the clusters of the current partition

Step 3: Assign each objects to most similar center

<table>
<thead>
<tr>
<th>Point</th>
<th>P</th>
<th>P(x, y)</th>
<th>m1 distance</th>
<th>m2 distance</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>a</td>
<td>(3, 4)</td>
<td>0.00</td>
<td>5.10</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p02</td>
<td>b</td>
<td>(3, 6)</td>
<td>2.00</td>
<td>5.10</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p03</td>
<td>c</td>
<td>(3, 8)</td>
<td>4.00</td>
<td>5.83</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p04</td>
<td>d</td>
<td>(4, 5)</td>
<td>1.41</td>
<td>4.00</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p05</td>
<td>e</td>
<td>(4, 7)</td>
<td>3.16</td>
<td>4.47</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p06</td>
<td>f</td>
<td>(5, 1)</td>
<td>3.61</td>
<td>5.00</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p07</td>
<td>g</td>
<td>(5, 5)</td>
<td>2.24</td>
<td>3.00</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p08</td>
<td>h</td>
<td>(7, 3)</td>
<td>4.12</td>
<td>2.24</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p09</td>
<td>i</td>
<td>(7, 5)</td>
<td>4.12</td>
<td>1.00</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p10</td>
<td>j</td>
<td>(8, 5)</td>
<td>5.10</td>
<td>0.00</td>
<td>Cluster2</td>
</tr>
</tbody>
</table>

**K-Means Clustering**

Initial: m1 (3, 4)
Initial: m2 (8, 5)
Step 2: Compute seed points as the centroids of the clusters of the current partition

Step 3: Assign each objects to most similar center

Euclidean distance
\[ b(3,6) \leftrightarrow m1(3,4) = ((3-3)^2 + (4-6)^2)^{1/2} = 0^2 + (-2)^2^{1/2} = (0 + 4)^{1/2} = (4)^{1/2} = 2.00 \]

Euclidean distance
\[ b(3,6) \leftrightarrow m2(8,5) = ((8-3)^2 + (5-6)^2)^{1/2} = (5^2 + (-1)^2)^{1/2} = (25 + 1)^{1/2} = (26)^{1/2} = 5.10 \]
**Step 4: Update the cluster means,**

Repeat Step 2, 3,

stop when no more new assignment

---

**K-Means Clustering**

- Point | P | P(x,y) | m1 distance | m2 distance | Cluster
- p01   | a  | (3, 4) | 1.43        | 4.34        | Cluster1
- p02   | b  | (3, 6) | 1.22        | 4.64        | Cluster1
- p03   | c  | (3, 8) | 2.99        | 5.68        | Cluster1
- p04   | d  | (4, 5) | 0.20        | 3.40        | Cluster1
- p05   | e  | (4, 7) | 1.87        | 4.27        | Cluster1
- p06   | f  | (5, 1) | 4.29        | 4.06        | Cluster2
- p07   | g  | (5, 5) | 1.15        | 2.42        | Cluster1
- p08   | h  | (7, 3) | 3.80        | 1.37        | Cluster2
- p09   | i  | (7, 5) | 3.14        | 0.75        | Cluster2
- p10   | j  | (8, 5) | 4.14        | 0.95        | Cluster2

m1 = (3.86, 5.14)

m2 = (7.33, 4.33)
Step 4: Update the cluster means, Repeat Step 2, 3, stop when no more new assignment

\[ m_1 = (3.67, 5.83) \]
\[ M_2 = (6.75, 3.50) \]

<table>
<thead>
<tr>
<th>Point</th>
<th>P</th>
<th>P(x,y)</th>
<th>( m_1 ) distance</th>
<th>( m_2 ) distance</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>a</td>
<td>(3, 4)</td>
<td>1.95</td>
<td>3.78</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p02</td>
<td>b</td>
<td>(3, 6)</td>
<td>0.69</td>
<td>4.51</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p03</td>
<td>c</td>
<td>(3, 8)</td>
<td>2.27</td>
<td>5.86</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p04</td>
<td>d</td>
<td>(4, 5)</td>
<td>0.89</td>
<td>3.13</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p05</td>
<td>e</td>
<td>(4, 7)</td>
<td>1.22</td>
<td>4.45</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p06</td>
<td>f</td>
<td>(5, 1)</td>
<td>5.01</td>
<td>3.05</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p07</td>
<td>g</td>
<td>(5, 5)</td>
<td>1.57</td>
<td>2.30</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p08</td>
<td>h</td>
<td>(7, 3)</td>
<td>4.37</td>
<td>0.56</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p09</td>
<td>i</td>
<td>(7, 5)</td>
<td>3.43</td>
<td>1.52</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p10</td>
<td>j</td>
<td>(8, 5)</td>
<td>4.41</td>
<td>1.95</td>
<td>Cluster2</td>
</tr>
</tbody>
</table>

\( K \)-Means Clustering
### K-Means Clustering

Stop when no more new assignment

<table>
<thead>
<tr>
<th>Point</th>
<th>P</th>
<th>P(x,y)</th>
<th>m1 distance</th>
<th>m2 distance</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>a</td>
<td>(3, 4)</td>
<td>1.95</td>
<td>3.78</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p02</td>
<td>b</td>
<td>(3, 6)</td>
<td>0.69</td>
<td>4.51</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p03</td>
<td>c</td>
<td>(3, 8)</td>
<td>2.27</td>
<td>5.86</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p04</td>
<td>d</td>
<td>(4, 5)</td>
<td>0.89</td>
<td>3.13</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p05</td>
<td>e</td>
<td>(4, 7)</td>
<td>1.22</td>
<td>4.45</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p06</td>
<td>f</td>
<td>(5, 1)</td>
<td>5.01</td>
<td>3.05</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p07</td>
<td>g</td>
<td>(5, 5)</td>
<td>1.57</td>
<td>2.30</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p08</td>
<td>h</td>
<td>(7, 3)</td>
<td>4.37</td>
<td>0.56</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p09</td>
<td>i</td>
<td>(7, 5)</td>
<td>3.43</td>
<td>1.52</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p10</td>
<td>j</td>
<td>(8, 5)</td>
<td>4.41</td>
<td>1.95</td>
<td>Cluster2</td>
</tr>
</tbody>
</table>

m1 (3.67, 5.83)
m2 (6.75, 3.50)
**K-Means Clustering** *(K=2, two clusters)*

*stop when no more new assignment*

<table>
<thead>
<tr>
<th>Point</th>
<th>P</th>
<th>P(x,y)</th>
<th>m1 distance</th>
<th>m2 distance</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>a</td>
<td>(3, 4)</td>
<td>1.95</td>
<td>3.78</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p02</td>
<td>b</td>
<td>(3, 6)</td>
<td>0.69</td>
<td>4.51</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p03</td>
<td>c</td>
<td>(3, 8)</td>
<td>2.27</td>
<td>5.86</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p04</td>
<td>d</td>
<td>(4, 5)</td>
<td>0.89</td>
<td>3.13</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p05</td>
<td>e</td>
<td>(4, 7)</td>
<td>1.22</td>
<td>4.45</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p06</td>
<td>f</td>
<td>(5, 1)</td>
<td>5.01</td>
<td>3.05</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p07</td>
<td>g</td>
<td>(5, 5)</td>
<td>1.57</td>
<td>2.30</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p08</td>
<td>h</td>
<td>(7, 3)</td>
<td>4.37</td>
<td>0.56</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p09</td>
<td>i</td>
<td>(7, 5)</td>
<td>3.43</td>
<td>1.52</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p10</td>
<td>j</td>
<td>(8, 5)</td>
<td>4.41</td>
<td>1.95</td>
<td>Cluster2</td>
</tr>
</tbody>
</table>

**K-Means Clustering**

m1 (3.67, 5.83)
m2 (6.75, 3.50)
### K-Means Clustering

<table>
<thead>
<tr>
<th>Point</th>
<th>P</th>
<th>P(x,y)</th>
<th>m1 distance</th>
<th>m2 distance</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>a</td>
<td>(3, 4)</td>
<td>1.95</td>
<td>3.78</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p02</td>
<td>b</td>
<td>(3, 6)</td>
<td>0.69</td>
<td>4.51</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p03</td>
<td>c</td>
<td>(3, 8)</td>
<td>2.27</td>
<td>5.86</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p04</td>
<td>d</td>
<td>(4, 5)</td>
<td>0.89</td>
<td>3.13</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p05</td>
<td>e</td>
<td>(4, 7)</td>
<td>1.22</td>
<td>4.45</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p06</td>
<td>f</td>
<td>(5, 1)</td>
<td>5.01</td>
<td>3.05</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p07</td>
<td>g</td>
<td>(5, 5)</td>
<td>1.57</td>
<td>2.30</td>
<td>Cluster1</td>
</tr>
<tr>
<td>p08</td>
<td>h</td>
<td>(7, 3)</td>
<td>4.37</td>
<td>0.56</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p09</td>
<td>i</td>
<td>(7, 5)</td>
<td>3.43</td>
<td>1.52</td>
<td>Cluster2</td>
</tr>
<tr>
<td>p10</td>
<td>j</td>
<td>(8, 5)</td>
<td>4.41</td>
<td>1.95</td>
<td>Cluster2</td>
</tr>
</tbody>
</table>

m1 \((3.67, 5.83)\)

m2 \((6.75, 3.50)\)
Classification and Prediction

Data Mining and Machine Learning in Google Colab

```python
# Import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from pandas.plotting import scatter_matrix

# Import sklearn
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier

print("Imported")

# Load dataset
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
df = pd.read_csv(url, names=names)

print(df.head(10))
print(df.tail(10))
print(df.describe())
print(df.info())
print(df.shape)
print(df.groupby('class').size())

plt.rcParams["figure.figsize"] = (10,8)
df.plot(kind='box', subplots=True, layout=(2,2), sharex=False, sharey=False)
plt.show()

df.hist()
plt.show()
```

https://colab.research.google.com/drive/1QE7fR2OxHiQ0_p6l1nnZDIFF354Nf_Lw
# importing the libraries
import numpy as np
import matplotlib.pyplot as plt
#matplotlib inline
import pandas as pd

# importing the Iris dataset with pandas
# Load dataset
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
df = pd.read_csv(url, names=names)

array = df.values
X = array[:,0:4]
Y = array[:,4]

# Finding the optimum number of clusters for k-means classification
from sklearn.cluster import KMeans
wcss = []

for i in range(1, 8):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

# Plotting the results onto a line graph, allowing us to observe 'The elbow'
plt.rcParams["figure.figsize"] = (10,8)
plt.plot(range(1, 8), wcss)
plt.title('The elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS') # within cluster sum of squares
plt.show()
#importing the libraries
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd

#importing the Iris dataset with pandas
# Load dataset
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
df = pd.read_csv(url, names=names)

array = df.values
X = array[:,0:4]
Y = array[:,4]
# Finding the optimum number of clusters for k-means classification

```python
from sklearn.cluster import KMeans
wcss = []

for i in range(1, 8):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

# Plotting the results onto a line graph, allowing us to observe 'The elbow'
plt.rcParams["figure.figsize"] = (10,8)
plt.plot(range(1, 8), wcss)
plt.title('The elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')  # within cluster sum of squares
plt.show()
```

https://colab.research.google.com/drive/1QE7fR2OxHiQ0_p6I1nnZDIFF354Nf_Lw
**K-Means Clustering**

The elbow method \((k=3)\)

![Image of The elbow method graph]

https://colab.research.google.com/drive/1QE7fR2OxHiQ0_p6l1nnZDlFF354Nf_Lw
```python
kmeans = KMeans(n_clusters = 3, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
y_kmeans = kmeans.fit_predict(X)
```

#Applying kmeans to the dataset / Creating the kmeans classifier
1 kmeans = KMeans(n_clusters = 3, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
2 y_kmeans = kmeans.fit_predict(X).
# Visualising the clusters
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Iris-setosa')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Iris-versicolour')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Iris-virginica')

# Plotting the centroids of the clusters
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 100, c = 'yellow', label = 'Centroids')

plt.legend()
#Applying kmeans to the dataset / Creating the kmeans classifier
kmeans = KMeans(n_clusters = 3, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
y_kmeans = kmeans.fit_predict(X).

#Visualising the clusters
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Iris-setosa')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Iris-versicolour')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Iris-virginica')

#Plotting the centroids of the clusters
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 100, c = 'yellow', label = 'Centroids')
plt.legend()
Outline

• Unsupervised Learning
• Cluster Analysis
• K-Means Clustering
References

- Jiawei Han, Micheline Kamber and Jian Pei (2011), Data Mining: Concepts and Techniques, Third Edition, Morgan Kaufmann 2011.