Department of Computer Engineering
University of Peradeniya
CO544: Machine Learning and Data Mining

# Lab 04: Clustering and Association Rule Learning

June 24, 2025

## Objectives

To provide students hands-on experience in clustering and Association Rule Learning using Python. At the end of the lab, students should be able to:

- Use the K-Means algorithm for a given dataset.
- Analyze the output of the K-Means algorithm and interpret the results.
- Use the Apriori algorithm to discover association rules.
- Analyze the output of the Apriori algorithm and interpret the results.

## 1 Clustering

### 1.1 Importing Required Modules

```
from sklearn.cluster import KMeans
from sklearn.datasets.samples_generator import make_blobs  # to generate sample
    datasets
```

### 1.2 Creating a Sample Dataset with 4 Clusters

```
X, y = make_blobs(
    n_samples=400,
    centers=4,
    cluster_std=0.90,
    random_state=1
)
```

### 1.3 Determining the Optimum Value of $k$ Using the Elbow Method

```
import matplotlib.pyplot as plt

wcss = []  # within cluster sum of squares
for i in range(1, 11):
    kmeans = KMeans(
        n_clusters=i,
        init='k-means++',
        max_iter=300,
        n_init=10,
        random_state=0
    )
    kmeans.fit(X)
```

```
    wcss.append(kmeans.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

### 1.4 Applying the K-Means Algorithm

```
kmeans = KMeans(n_clusters=4, random_state=0)  # from Elbow method
closest_cluster_index = kmeans.fit_predict(X)
cluster_centers = kmeans.cluster_centers_
```

### 1.5 Visualizing Clusters

```
plt.scatter(X[:, 0], X[:, 1], c='green')
plt.scatter(
    cluster_centers[:, 0],
    cluster_centers[:, 1],
    s=200,
    c='black',
    marker='*'
)
plt.show()
```

#### Exercise 01

1. Import the `iris` dataset from scikit-learn. Convert it into an unlabeled dataset by removing the class attribute.
2. Use the Elbow method to identify the best value for $k$ (minimizing WCSS).
3. Fit the K-Means algorithm with the $k$ found in part (b).
4. Explain the output of:

   ```
   kmeans.cluster_centers_
   ```

   where `kmeans` is your fitted `KMeans` object.
5. Visualize the data points and cluster centers in a 3D plot using the first three features as axes.

## 2 Association Rule Learning

### 2.1 Installing Apriori

```
pip install mlxtend  # installs the mlxtend library
```

### 2.2 Importing Required Modules

```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEncoder
import pandas as pd
```

### 2.3 Input Data

```
dataset = [
    ['Milk', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
    ['Dill', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
    ['Milk', 'Apple', 'Kidney Beans', 'Eggs'],
    ['Milk', 'Unicorn', 'Corn', 'Kidney Beans', 'Yogurt'],
    ['Corn', 'Onion', 'Onion', 'Kidney Beans', 'Ice cream', 'Eggs']
]
```

### 2.4 Creating the DataFrame of Frequent Itemsets

```
te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
```

### 2.5 Applying Apriori Algorithm and Finding Association Rules

```
freq = apriori(df, min_support=0.002, use_colnames=True)
rules = association_rules(freq, metric="lift", min_threshold=1)
```

### Exercise 02

1. Import the provided `groceries.csv` dataset.
2. Explore the dataset and build the frequent-item DataFrame.
3. Apply the Apriori algorithm to find itemsets with support $> 8\%$.
4. Generate association rules using the `lift` metric.
5. Select one rule and interpret it in your own words.
6. How many rules satisfy both `lift` $> 4$ and `confidence` $> 0.8$?

## Submission

Submit:

- Two Python scripts—one for the clustering exercise and one for association-rule learning.
- A single PDF containing answer for both Exercises 01 and 02

Place all files in a folder, compress it, and name it `eyyxxxlab4`, where `yyxxx` is your registration number.