KATHMANDU UNIVERSITY

Department of Artificial Intelligence

Dhulikhel, Kavre

Lab Report On: Unsupervised Data Mining Methods

Submitted By:

Name: Aaryan Shakya

Roll No: 20

Subject Code: AICC 301

Submitted To:

Mr. Sunil Regmi

Lecturer, Department of Artificial Intelligence

Submission Date:

6th July 2025

Objectives

1. Understand and apply association rule mining to uncover hidden relationships among items in transactional datasets.
2. Implement clustering algorithms to group unlabeled data based on similarity.
3. Evaluate the performance of unsupervised learning models using internal and external validation methods.

Introduction

1.1. Association Rule Mining

Association rule mining is an unsupervised learning technique that identifies meaningful patterns, correlations, and relationships among a set of items in large databases. It is most commonly used to discover which items frequently appear together in transactions. For example, it can help determine that customers who buy bread and milk often also purchase butter.

How It Works
Association rules are typically written in the form: "If item A is bought, then item B is also likely to be bought." These rules are evaluated using measures like frequency (support), reliability (confidence), and strength of the relationship (lift). The more often a rule appears in the data and the stronger the association, the more useful it is.

1.1.1. Apriori Algorithm

Apriori is one of the most widely used algorithms to generate frequent itemsets and derive association rules. It works by first identifying items that frequently occur together and then expanding those sets into longer sequences, filtering them based on minimum support and confidence levels.

Example
In a grocery store, if many customers who buy bread and eggs also buy milk, the rule could be: "If a customer buys bread and eggs, then they are also likely to buy milk."

Use Cases & Applications

- **Retail:** Market basket analysis to optimize product placement and cross-selling.
- **E-commerce:** Recommending products based on previous purchases.
- **Healthcare:** Identifying common co-occurring symptoms or prescriptions.
- **Web Analytics:** Understanding common navigation paths or click patterns.

Advantages

- Reveals hidden patterns without needing labeled data.
- Generates understandable rules that support decision-making.

- Scales well with large datasets when optimized properly.
- Useful in a wide variety of industries for customer behavior analysis.

1.2. Clustering

Clustering is the process of grouping similar data points into clusters. Unlike classification, clustering does not use predefined labels. Instead, it automatically identifies structure in data based on how similar or different the points are from one another. Clustering is used in exploratory data analysis, customer segmentation, image processing, and more.

1.2.1. K-Means Clustering

K-Means is one of the simplest and most commonly used clustering algorithms. It divides the dataset into a predefined number of clusters, where each cluster is represented by a center point. The algorithm works by assigning each data point to the closest center, then updating the center until the clusters become stable.

Example

A company can use K-Means to group customers based on purchase history into categories like frequent buyers, occasional shoppers, and new customers.

Use Cases & Applications

- Market segmentation
- Image segmentation
- Grouping similar search results
- Customer profiling

Advantages

- Simple and easy to implement
- Works well with large datasets
- Fast and computationally efficient

Limitations

- Requires specifying the number of clusters in advance
- Sensitive to outliers and initial placement of centers
- Works best when clusters are clearly separated

1.2.2. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN is a density-based clustering algorithm that groups together closely packed points and identifies points in sparse regions as noise or outliers. Unlike K-Means, it does not require the number of clusters to be specified in advance.

Example
DBSCAN can be used to identify areas of high customer traffic in a city based on GPS location data, while ignoring isolated or unusual points as outliers.

Use Cases & Applications

- Geospatial data analysis
- Anomaly detection in banking or network systems
- Clustering users on social media based on interaction patterns

Advantages

- Detects clusters of any shape
- Automatically identifies noise or outliers
- No need to specify number of clusters beforehand

Limitations

- May struggle with datasets that have clusters of varying density
- Sensitive to parameter selection

1.2.3. Hierarchical Clustering (Agglomerative and Divisive)

Hierarchical clustering builds a hierarchy of clusters using either a bottom-up approach (agglomerative) or a top-down approach (divisive). The results are often visualized using a dendrogram, which shows how clusters are merged or split at each step.

Agglomerative: Each point starts as its own cluster and clusters are merged step by step.
Divisive: All points start in one cluster and are split recursively into smaller clusters.

Example
In biology, hierarchical clustering can be used to group species by genetic similarity, forming a tree of life structure.

Use Cases & Applications

- Creating taxonomies
- Organizing documents or web pages by topic
- Analyzing social networks

Advantages

- No need to predefine the number of clusters
- Provides a clear visual representation of relationships
- Can handle data with nested structures

Limitations

- Computationally expensive for large datasets
- Not flexible once a decision to merge or split is made

1.3. Clustering Validation Techniques

Evaluating clustering is difficult because there are no predefined labels. Therefore, specialized validation techniques are used to assess the quality of clusters.

1.3.1. Intrinsic (Internal) Validation

Silhouette Score
This metric evaluates how well each data point fits within its own cluster compared to others. A high silhouette score indicates that data points are well matched to their cluster and poorly matched to neighboring clusters.

Use Cases:

- Selecting the best number of clusters
- Comparing different clustering algorithms
- Measuring cluster separation

Advantages:

- Requires no external labels
- Useful in determining how natural or compact the clusters are

1.3.2. Extrinsic (External) Validation

Adjusted Rand Index (ARI)
This metric compares clustering results with actual labels (when available), adjusting for random chance. A high score means that the clustering closely matches the real categories.

Use Cases:

- Benchmarking against labeled datasets like Iris
- Evaluating clustering accuracy
- Comparing clustering algorithms with known classifications

Advantages:

- Provides a fair comparison when ground truth exists
- Useful in academic and benchmark datasets

Experiment

Implementing Apriori, K-Means, and DBSCAN with Visualization and Validation

Python Code

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

# Sample transactional data
dataset = [['Milk', 'Bread', 'Butter'],
        ['Bread', 'Diaper', 'Beer', 'Eggs'],
        ['Milk', 'Diaper', 'Beer', 'Cola'],
        ['Bread', 'Milk', 'Diaper', 'Beer'],
        ['Bread', 'Milk', 'Diaper', 'Cola']]

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

# Apriori to find frequent itemsets
frequent_itemsets = apriori(df, min_support=0.6, use_colnames=True)

# Generate association rules
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)
print(rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']])
```

Step 2: Clustering using K-Means and DBSCAN

- Dataset: Iris dataset (numerical features).
- Libraries: scikit-learn for clustering and validation.

Python Code

```
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans, DBSCAN
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

# Load dataset
iris = load_iris()
X = iris.data

# K-Means clustering
kmeans = KMeans(n_clusters=3, random_state=42)
labels_kmeans = kmeans.fit_predict(X)
silhouette_kmeans = silhouette_score(X, labels_kmeans)
print(f"K-Means Silhouette Score: {silhouette_kmeans:.4f}")

# DBSCAN clustering
dbscan = DBSCAN(eps=0.5, min_samples=5)
labels_dbscan = dbscan.fit_predict(X)
silhouette_dbscan = silhouette_score(X, labels_dbscan) if len(set(labels_dbscan)) > 1 else -1
print(f"DBSCAN Silhouette Score: {silhouette_dbscan:.4f}")

# Visualization of K-Means clusters
plt.scatter(X[:, 0], X[:, 1], c=labels_kmeans, cmap='viridis')
plt.title('K-Means Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()

# Visualization of DBSCAN clusters
plt.scatter(X[:, 0], X[:, 1], c=labels_dbscan, cmap='plasma')
plt.title('DBSCAN Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()
```

Conclusion

In this lab, we explored key unsupervised data mining methods, focusing on association rule mining and clustering techniques. By implementing the Apriori algorithm, we successfully uncovered meaningful patterns and relationships within transactional data, demonstrating how association rules can provide valuable insights without the need for labeled data.

Through the application of clustering algorithms like K-Means and DBSCAN, we grouped unlabeled data based on similarity and density, effectively revealing hidden structures and natural groupings within datasets. The different characteristics of these clustering methods were highlighted — K-Means excels in partitioning data into clear spherical clusters, while DBSCAN is powerful for discovering clusters of arbitrary shape and identifying noise.

Furthermore, we learned the importance of validating unsupervised models using intrinsic metrics such as the silhouette score and extrinsic metrics like the Adjusted Rand Index when ground truth labels are available. These evaluation techniques ensure that the clustering results are meaningful and reliable.

Overall, the lab reinforced the value of unsupervised learning in extracting insights from complex data where labels are not available, with broad applications ranging from market basket analysis and customer segmentation to anomaly detection and beyond.