Association Analysis - 2

Yi Yang (yiyan338) Helena Llorens Lluís (hllor282)

March 2025

1 Introduction

In this lab, we use the Weka data mining toolkit to solve association problems.

Through this lab, we apply Weka's data mining algorithms to examine the dataset's structure, assess the limitations of clustering, and demonstrate how association rules provide a more interpretable and effective solution.

2 Dataset

The Monk1 dataset is an artificial dataset commonly used in machine learning research and benchmarking. It is part of the Monk's Problems, a set of three classification tasks designed to evaluate the performance of various learning algorithms. The dataset consists of 124 instances, each described by six discrete attributes and a binary class label.

Unlike natural datasets, the Monk1 dataset does not contain missing values or noisy data, allowing for controlled experimentation. However, due to its artificial nature, clustering algorithms struggle to capture its underlying structure effectively. Instead, association rule learning provides a more suitable method for uncovering patterns within the data.

3 Methodology

This lab follows a structured approach to evaluate the effectiveness of clustering and association rule learning in classifying instances within the Monk1 dataset. The methodology is divided into two main phases:

- Clustering analysis: Apply multiple clustering algorithms with varying numbers of clusters. Use Weka's "Clusters to Class Evaluation" tool to compare the generated clusters with the actual class labels.

- Association Rule Learning: Apply association rule mining using Weka's Apriori algorithm with the following settings (Minimum support: 0.05, Maximum number of rules: 19). Identify a minimal set of rules that can accurately predict the class label based on the other attributes.

4 Experimental Results

4.1 Clustering

First, we applied three different clustering techniques to evaluate whether they could accurately identify the existing class division within the dataset. For algorithms that require a predefined number of clusters, we set K = 2, as the dataset consists of two distinct classes. Additionally, we excluded the class attribute from the clustering process to ensure an unsupervised approach.

To assess the performance of each clustering algorithm, we used Weka's Clusters to Class Evaluation tool to compare the generated clusters with the actual class labels. This allowed us to determine how well the clustering techniques aligned with the true class distribution.

4.1.1 Clustering with SimpleKMeans (K=2)

Table 1: KMeans Classification Accuracy

	Prediction class 0	**Prediction class 1**
Class 0	40	22
Class 1	37	25

Incorrectly clustered instances: 59.0 (47.5806%).

The SimpleKMeans algorithm struggles to separate the classes correctly, misclassifying nearly half of the instances. This indicates that the data distribution does not naturally align with the Euclidean distance-based separation that KMeans relies on. The class labels in the dataset may depend on complex attribute interactions rather than simple feature similarities.

4.1.2 Clustering with Density-Based KMeans (K=2)

Table 2: Density-Based KMeans Classification Accuracy

	Prediction class 0	**Prediction class 1**
Class 0	44	18
Class 1	39	23

Incorrectly clustered instances: 57.0 (45.9677%).

The Density-Based KMeans algorithm slightly improves accuracy compared to SimpleKMeans, but still fails to correctly cluster a large portion of the instances. This suggests that the decision boundary between the two classes is not easily captured by centroid-based clustering.

4.1.3 Clustering with EM Algorithm

Table 3: EM Classification Accuracy

	Prediction class 0
Class 0	62
Class 1	62

Incorrectly clustered instances: 62.0 (50%).

The EM algorithm performs the worst among the clustering methods tested, essentially assigning the same number of instances to each cluster randomly.

4.2 Association analysis

Using Weka's Apriori algorithm, we generated 19 association rules with a minimum support of 0.05 and a minimum confidence of 0.9. The primary goal was to identify a minimal set of rules that can accurately predict class 1, minimizing redundancy.

From the best rules found, we observe that several rules perfectly predict class 1. Among them, we can identify four key rules that are sufficient to describe class 1:

1. attribute5 = 1 $\rightarrow$ class = 1 (Support: 29 instances, Confidence: 1.0)
2. attribute1 = 3 & attribute2 = 3 $\rightarrow$ class = 1 (Support: 17 instances, Confidence: 1.0)
3. attribute1 = 2 & attribute2 = 2 $\rightarrow$ class = 1 (Support: 15 instances, Confidence: 1.0)
4. attribute1 = 1 & attribute2 = 1 $\rightarrow$ class = 1 (Support: 9 instances, Confidence: 1.0)

These four rules can be consolidated into the logical expression (attribute1 = attribute2) OR (attribute5 = 1). This formulation captures the essential pattern for classifying instances as class 1. By retaining only these four rules, we achieve a concise and non-redundant representation of class 1.

Would you say that the clustering algorithms fail or perform poorly for the monk1 dataset? Why or why not?

Clustering algorithms are likely to perform poorly on the MONK-1 dataset because the data structure does not naturally form distinct clusters, making rule-based classification methods like Apriori or decision trees more suitable.