TDDD41-732A75 Data Mining - Clustering and Association Analysis and Advanced Data Mining

Pranav Pankaj Chandode (prach896)

Varun Gurupurandar (vargu125)

Lab 1 Clustering

1. Introduction

This lab is to provide hands-on experience with clustering techniques using Weka, a popular data mining toolkit.We will explore different clustering algorithms, including SimpleKMeans and MakeDensityBasedClusterer. The goal is to understand how clustering algorithms work, analyze their outputs, and gain insights into how data can be grouped.

2. SimpleKMeans Clustering

a. Choose a set of attributes for clustering and provide a motivation. (Hint: always ignore the attribute "name". Why does the name attribute need to be ignored?)
Solution: We ignored the name attribute as suggested, as it has no value addition to the clustering algorithm. The name attribute is a string and as the KMeans algorithm uses Euclidean distance, it will impact adversely in this algorithm.
We chose the attributes of fat and protein because we thought that there would be no direct correlation between them so there would be some meaningful clusters. We tried selecting the attributes of fat and energy but there was a very high correlation between them, so the KMeans algorithm was virtually drawing a line and clustering those data points accordingly. We also tried selecting multiple attributes but then without doing any dimensionality reduction, we would lose a lot of variance, so we stuck with fat and protein. The clusters formed using them were based on the fat content in the food items; higher fat one cluster and lower fat other cluster.

b. Experiment with at least two different numbers of clusters, e.g., 2 and 5, but with the same seed value 10.
Solution: We experimented with 2 and 5 clusters. The SSE of the algorithm decreased as the number of clusters increased. This is obvious according to the clustering algorithm; the SSE calculates the error by taking the difference within the cluster. The higher number of clusters signifies that the points which are closer would be classified together, meaning the distance between the cluster centers and the points will decrease, which in turn will decrease the SSE.

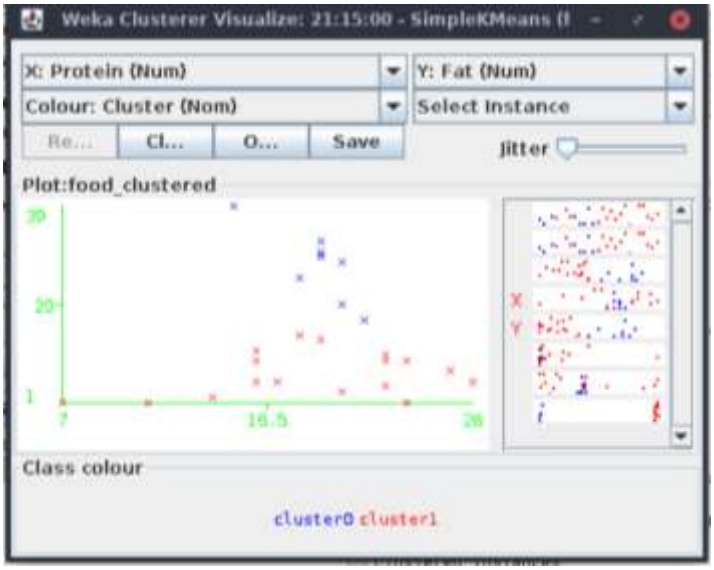
The SSE for **2** clusters was **1.73176** and there was class imbalance as there were 18 points in one cluster and 9 in the other. This seems logical and we can say that there is clustering based on the fat content in the food items.

The SSE for **5** clusters was **0.58006**. There was almost even number of points in all the classes but the distinction between them was not perceivable and arguable.

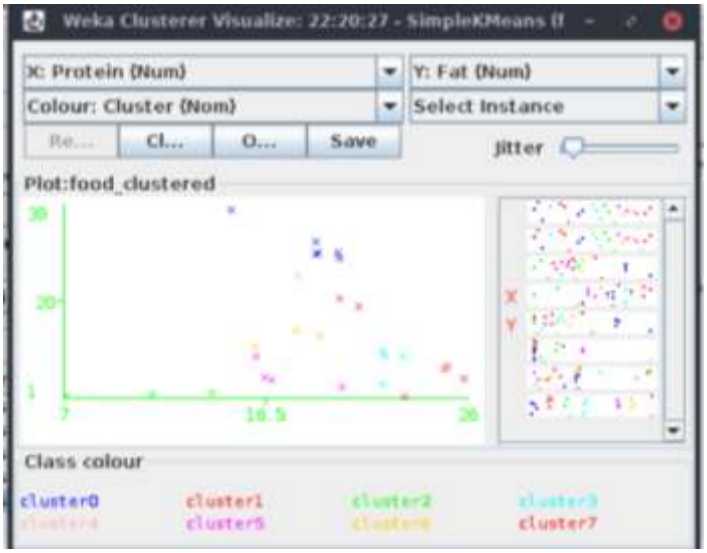
c. Try with a different seed value, i.e., different initial cluster centers. Compare the results with previous results. Explain what the seed value controls.
Solution: The seed value controls the initialization of cluster centroids. As we change the seed value it can affect the cluster formations eventually. If the clusters are less, the probability of formation of same clusters is considerably high but if the cluster number is large, we can get different clusters for different seed values with high probability.

We experimented with 2 and 5 clusters, when the seed value was changed to 20, the clusters were categorized differently than when the seed value was 10. This also affects the SSE, which increased in the case of 2 clusters to **2.36332.** Even when the seed changed for 5 clusters, the clusters were different but this time the SSE reduced to **0.36880.** Hence, the seed value controls the cluster formations as it initializes the cluster centroids which in turn affect the clusters.

d. Do you think the clusters are "good" clusters? (Are all members "similar" to each other? Are members from different clusters dissimilar?)
Solution: We select only 2 attributes and get the centroids of the same since the number of attributes is less, we can clearly see the similarity between the data points and classify the clusters as good clusters. The data points in different clusters are dissimilar. On the other hand, if the number of attributes is increased, then we see that the visualization is difficult and categorizing the clusters as good clusters will also become difficult. This can be visualized in the plots below for 2 and 8 clusters. The 2 clusters are easily distinguishable and we can say that the clusters are dissimilar from one another and the within cluster points are similar. While in 8 clusters, it is difficult to comment on the similarity and dissimilarity between the clusters.



K = 2 clusters and seed 10



K = 8 clusters and seed 10

e. What does each cluster represent? Choose one of the results. Make up labels (words or phrases in English) that characterize each cluster.
Solution: One of the cluster represents high fat food, and the second represents low fat. The clustering is highly dependent on the fat content compared to the protein. This might be because there is very less correlation between both the features. This is evident in the cluster 2, seed 10 case; the centroid of both the clusters have the same protein value but different fat value.

3. **MakeDensityBased Clusters**
Now with MakeDensityBasedClusters, SimpleKMeans is turned into a density-based clusterer. You can set the minimum standard deviation for normal density calculation. Experiment with the algorithm as follows:

1. Use the SimpleKMeans clusterer which gave the result you chose in the previous section.

Solution: We chose k = 2, seed = 10 for this experiment.

2. Experiment with at least two different standard deviations. Compare the results. (Hint: Increasing the standard deviation to higher values will make the differences in different runs more obvious and thus it will be easier to conclude what the parameter does.)

Solution: We experimented with standard deviation values ranging from 1e-6 to 20. Initially, the effect of this parameter was barely noticeable until we reached a minimum standard deviation of around 10. Beyond this point, the blue cluster started to absorb data points from the red cluster, eventually causing them to disappear.

Unlike traditional k-means clustering, density-based clustering generates probability density functions (PDFs) for each cluster. These PDFs are determined by the cluster's mean (centroid) and standard deviation. A data point is assigned to the cluster with the highest probability density.

To prevent numerical instability when the standard deviation becomes very small, a lower limit is enforced—if a cluster's standard deviation falls below this threshold, it is artificially inflated in the PDF. This ensures clusters do not collapse onto a single or very few data points. However, if the standard deviation is set too high, larger clusters tend to absorb smaller ones as iterations progress.