TDDD41 DATA MINING - CLUSTERING AND ASSOCIATION
ANALYSIS

Lab 2 - Group 4 Report

Lawrence Thanakumar Rajappa (lawra776)
Kyriakos Domanos (kyrdo817)

March 4, 2020

# Association Analysis

Data Preparation

In this lab assignment, we used iris dataset which contains 50 samples of data which corresponds to three species of iris. As a starting process, we discretize the data since association analysis in weka cannot be done on continuous data. This is done by applying discretize filter available in Weka on only first 4 columns and now, the entire data is converted discrete format.

Clustering - SimpleKmeans

Now, we tried to apply **SimpleKmeans** clustering algorithm with the following specifications;

* **Seed value** : 10

* **No. of Clusters** : 3

and igonred **class** attribute and selected Classes to clusters evaluation to crosstabulate the clustering. We tried to visualize the result of clustering and found that with 3 bins, we could see that sepal width and length have lower impact when compared with petal width and length which has higher impact in naming the flower types. This is because, sepal width and length try to overlap more than petal width and length. Please refer the below images,
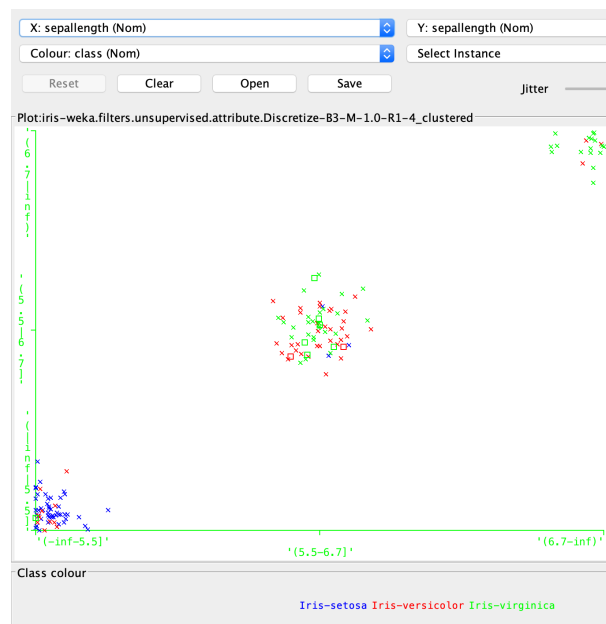

Figure 1: Sepal width and length tends to have more overlap in the data

1
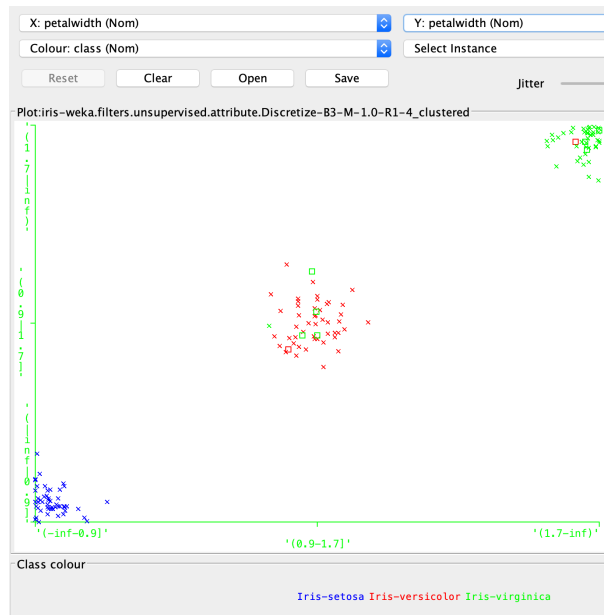
Figure 2: Petal width and length tends to have less overlap in the data

and also having less number of bins, we could lose some information as well. From the clustering result, we could see that clustering has been done well with 3 bins and 3 clusters, *iris setosa* has been clustered properly and belongs to *Cluster 2*, while there is a swap in case of *iris versicolor* and *iris virginica* between *Cluster 0* and *Cluster 1*. Please refer to the confusion matrix below,
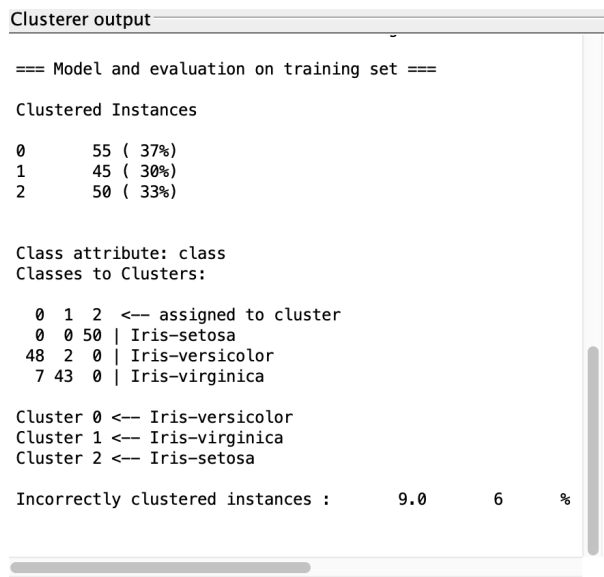
Figure 3: SimpleKmeans clustering with 3 bins

Also, if we try to increase the number of bins and discretize, the results are getting worse, please refer the image below,

Figure 4: SimpleKmeans clustering with 5 bins

Association Analysis - Apriori

Here we have used the following specifications for performing Apriori algorithm on 3 clustered dataset;

* **Minimum Support** = 0.1

* **Metric type** = Confidence

* **Minimum Confidence** = 0.9

with the above properties, we have performed apriori algorithm on the iris dataset and got the following association rules which are given below,

Figure 5: Apriori - Association Rules with Confidence

In this frequent itemset are the attribute-value pair and transaction is a single data point. Moreover, with default properties we weren't able to get proper association rules, hence we changed our properties a bit;

- ∗ **Number of Rules** = 1000

- ∗ **Metric type** = Confidence

- ∗ **Minimum Confidence** = 0.9

In the above properties, **Number of Rules** which specifies how many rules are needed and we get 1000 rules from this property and we keep the rule with 90% confidence.

- ∗ sepalwidth='(-inf-2.8]' petallength='(2.966667-4.933333]' ==> Cluster=cluster1, Occurrence = 30, Confidence:(1)

- ∗ petallength='(2.966667-4.933333]' petalwidth='(0.9-1.7]' ==> Cluster=cluster1, Occurrence = 48, Confidence:(1)

- ∗ petallength='(4.933333-inf)' petalwidth='(1.7-inf)' ==> Cluster=cluster2, Occurrence = 40, Confidence:(1)

- ∗ sepallength='(6.7-inf)' petallength='(4.933333-inf)' ==> Cluster=cluster2, Occurrence = 17, Confidence:(1)

- ∗ sepallength='(-inf-5.5]' petallength='(-inf-2.966667]' ==> Cluster = cluster3, Occurrence = 47, confidence:(1)

- ∗ sepalwidth='(2.8-3.6]' petallength='(-inf-2.966667]' ==> Cluster=cluster3, Occurrence = 36, Confidence:(1)

From the above association rules, we could see that we want class to be in consequent rather than in antecedent, because we need to predict the class of the flower and also when a new data comes into the system, we need to predict its species instead of having the class at beginning and attributes are placed in consequent because it is not to be predicted.

Experiments

Different Number of Bins

By increasing number of bins, we could yield more information and thus we can calculate distance between data points more precisely in k-means algorithm. On the other hand, having to many clusters which would yield less information and possibly it could make the algorithm ambiguous.

We tried to discretize the data with *6 bins* and found that the amount of incorrectly clustered data points are more when compared with the clustering result with 3 bins, which makes the above line true. We then applied *addCluster* filter on the 6 bins data and found the following association rules;

* petallength='(-inf-1.983333]' ==> cluster=cluster1 occurrence = 50 conf:(1)

* petalwidth='(-inf-0.5]' ==> cluster=cluster1 occurrence = 49 conf:(1)

* sepallength='(5.5-6.1]' petallength='(3.95-4.933333]' ==> cluster=cluster2 occurrence = 21 conf:(1)

* petallength='(3.95-4.933333]' petalwidth='(0.9-1.3]' ==> cluster=cluster2 occurrence = 18 conf:(1)

* sepallength='(6.1-6.7]' petallength='(4.933333-5.916667]' ==> cluster=cluster3 occurrence = 20 conf:(1)

* sepalwidth='(2.8-3.2]' petallength='(4.933333-5.916667]' ==> cluster=cluster3 occurrence = 18 conf:(1)

From the association rules above, we infer that cluster1 is characterized by only petal length and petal width, while cluster2 and cluster3 are characterized by mix of attribute values. This clustering is not good because there is 20.6667% of incorrectly clustered instances. This is because some of the values don't reach the minimum support and don't give rise to correct rules. From this we can conclude that instead of increasing the bins, we can perform discretize in an efficient way.

Different Number of Bins