732A75 Advanced Data Mining laboratory 2 report

Yuki Washio (timwa902) and Nicolas Taba (nicta839)

April 4, 2021

1 Introduction

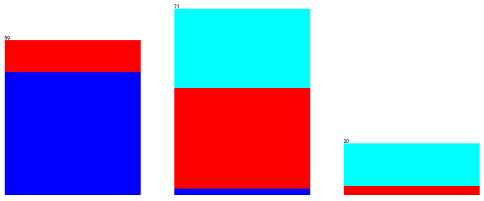
The aim of this laboratory exercise is to use assosciation analysis to describe clusters obtained from a mined dataset.

In this laboratory, we use the iris dataset. This dataset features 50 instances of each 3 classes of flowers (Iris Setosa, Iris Versicolor and Iris Virginica). Each of these data points has data pertaining to the sepal and petal length and width given in cm.

We will be using the seed value 10 whenever prompted. We also set the lower bound of minimum support to 0.2 throughout.

2 Using assosciation analysis with optimal parameters

In this section of the laboratory, we know the make up of this dataset and want to familiarize ourselves with the assosciation analysis tool of Weka. We begin by discretizing the dataset in 3 bins of data since continuous attributes cannot be processed by the software. We perform the simple Kmeans algorithm on the data like in the previous laboratory and crosstabulate the clusters with the class labels.



(a) data separated into 3 bins according to petal-width

```
Clustered Instances

0       55 ( 37%)
1       45 ( 30%)
2       50 ( 33%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0  0 50 | Iris-setosa
 48  2  0 | Iris-versicolor
  7 43  0 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-virginica
Cluster 2 <-- Iris-setosa

Incorrectly clustered instances :       9.0       6      %
```

(b) kmeans clustering using k=3

Figure 1: clustering using Kmeans with k=3 and n=3 bins

Here we see that a few elements of versicolor and virginica are wrongly clustered and account for a 68% error in the clustering. We now perform the apriori algorithm on this clustering to try and ascertain if the rules used to make this clusters have high confidence.

We use the apriori algorithm to find rules that have the clusters as consequent.

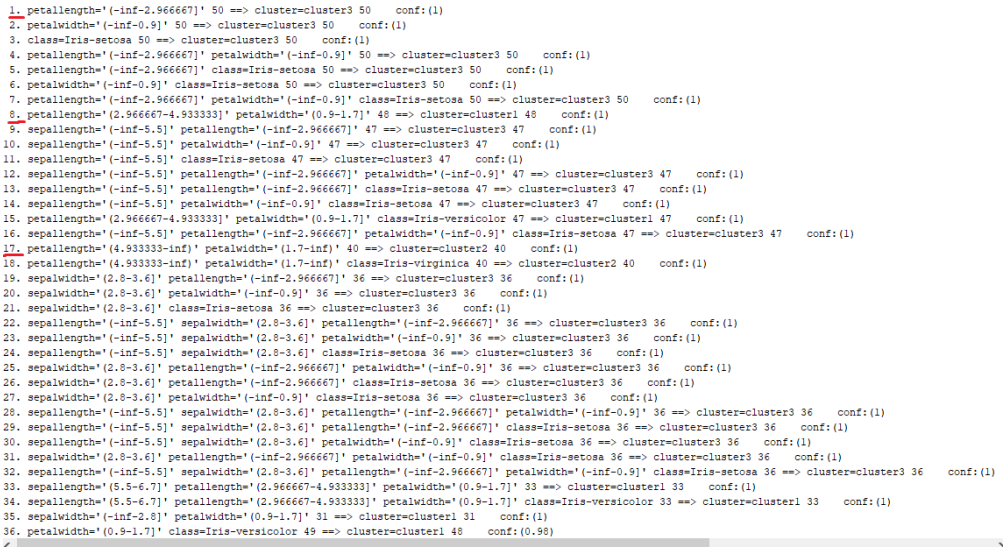


Figure 2: Apriori rules for k=3 and n=3 bins.

In Fig. 2, we have the 36 best rules for this clustering solution. All 35 first rules point to all three clusters with a confidence of 100%, with most rules indicating high confidence in identifying cluster 3 and few rules for clusters 2 and 1. The three best rules concern the petal length for the Third cluster that is completely correctly identified (rule 1). The other rules that allow us to identify clusters 1 and 2 concern petal length and width and allow a good clustering as the support for these rules is the same than for the classes we clustered earlier in ig. 1b.

This is a good indication that our clustering is correct as several rules can be found to cluster the data to one class using one or many determinant with very high support. The 5 first rules in Fig. 2 concern cluster 3 with a support of 50. This indicates that cluster 3 is the setosa class and that this particular class can easily be identified with respect to the others.

The documentation of the dataset indicates that one of the classes is linearly separable from the others whereas the other two are not linearly separable from each other. The result we obtained is thus the expected optimal parameters when we use a very well known dataset. (documentation: https://www.ida.liu.se/~732A75/lab/iris.arff)

3 Investigation of change of bins and cluster numbers

We now wish to investigate how the change in number of bins and clusters affects the clustering algorithm and the apriori algorithm. We know from the previous laboratory that the assignment of the correct number of clusters to the number of classes that we believe

compose the set is an important factor in how the Kmeans algorithm performs, but we do not know yet how the number of bins affects this performance. We will use the same process as previously with the apriori algorithm to assess these effects.

3.1 Changing the number of bins

We will keep the number of clusters fixed to the optimal value while changing the number of bins. We will use the previous result with n=3 and repeat the experiment with n=5 and 10.

```
Clustered Instances

0       63 ( 42%)
1       35 ( 23%)
2       52 ( 35%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0  0 50 | Iris-setosa
 15 33  2 | Iris-versicolor
 48  2  0 | Iris-virginica

Cluster 0 <-- Iris-virginica
Cluster 1 <-- Iris-versicolor
Cluster 2 <-- Iris-setosa

Incorrectly clustered instances :       19.0     12.6667 %
```

(a) Kmeans with 5 bins and k=3

```
Clustered Instances

0       62 ( 41%)
1       41 ( 27%)
2       47 ( 31%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
 26  0 24 | Iris-setosa
 16 27  7 | Iris-versicolor
 20 14 16 | Iris-virginica

Cluster 0 <-- Iris-virginica
Cluster 1 <-- Iris-versicolor
Cluster 2 <-- Iris-setosa

Incorrectly clustered instances :       79.0     52.6667 %
```

(b) Kmeans with 10 bins and k=3

```
Clustered Instances

0       55 ( 37%)
1       45 ( 30%)
2       50 ( 33%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0  0 50 | Iris-setosa
 48  2  0 | Iris-versicolor
  7 43  0 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-virginica
Cluster 2 <-- Iris-setosa

Incorrectly clustered instances :       9.0      6      %
```

(c) Kmeans with 3 bins and k=3

Figure 3: Kmeans algorithm output using different number of bins

From Fig. 3, we can see that increasing the number of bins has decreased the performance of the Kmeans algorithm. When the number of bins is much larger than the optimal (n=10 bins) we find that the performance goes as low as 52% error rate in this case. Most of the errors occur for the versicolor and virginica in the case of n= 5.

We now try to find rules assosciated to each clusters for 5 and 10 bins.

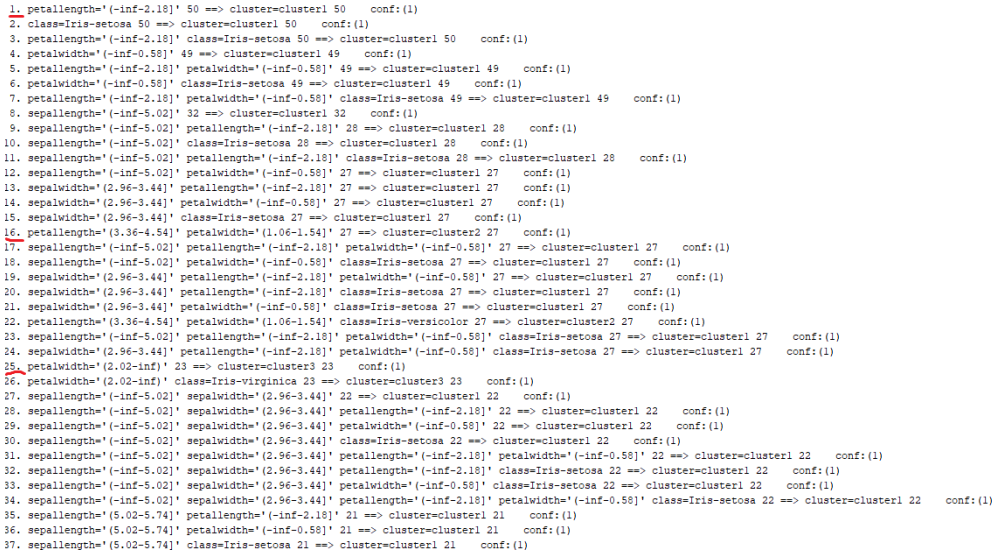


Figure 4: Apriori rules for k=3 and n=5 bins.

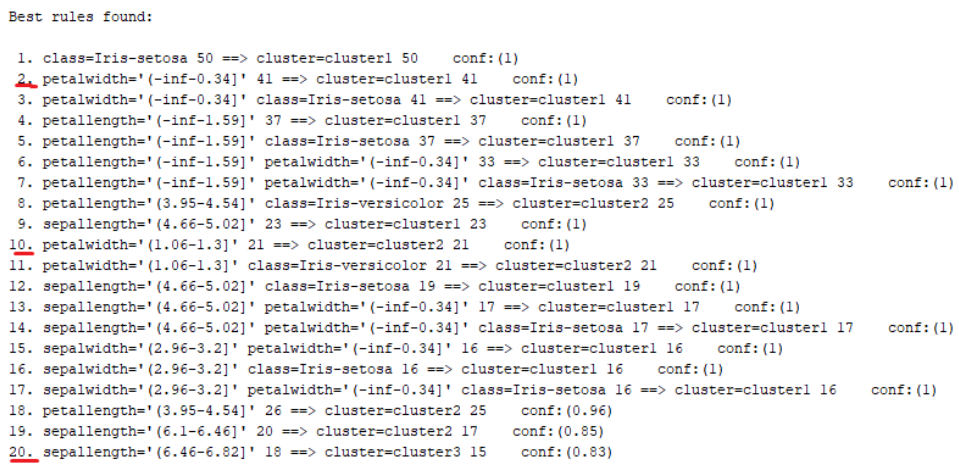


Figure 5: Apriori rules for k=3 and n=10 bins.

In Fig. 4 we see that the rules with highest confidence mostly point to cluster 1. There are some instances of cluster 3 being a consequent (rule 26 and 27). We can think from this that only 1 class can be clearly said to be dissimilar to the others. The best rules for clustering are marked in red.

In Fig 5, we find only 20 rules with high confidence (> 0.8) with rules that have all 3 clusters as consequent. However, we will notice that some of these rules are not "good" rules such as rule 1 that has "belonging to class iris-setosa" yielding cluster 1. This is not a good rule. We can ascertain from these rules that the bin number is not the correct one since the rules that we get are not good with some having low support 15 instances for rule 20 for example. The best rules are marked in red.

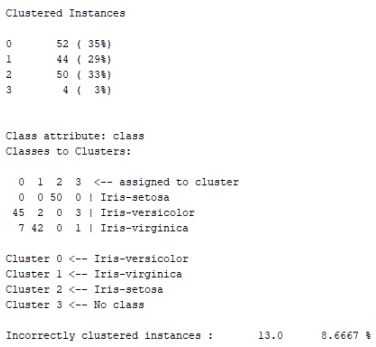
When we change the number of bins, we move the intervals that were used to differentiate when we only had 3 bins. With different intervals, we also notice that there are some

instances where some clusters do not have enough minimum support. For exmaple, in the case of cluster 1, using 5 bins, we have a support of 35/150 which is smaller than the required 0.2. This is why although the confidence in the rules is high, the support is very low and we do not trust these rules.

3.2 Changing the number of clusters

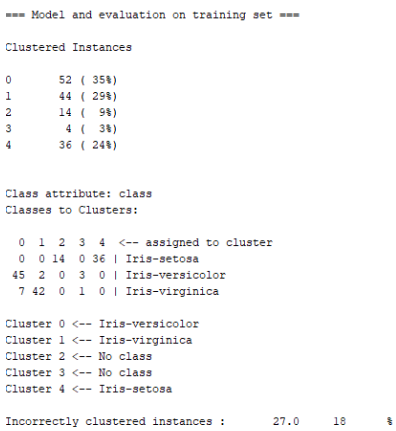
We now will investigate how the number of clusters affects the clustering algorithm and how the effects identified in the previous lab can be also identified using the rules produced by the apriori algorithm.

We will keep n=3 bins constant while varying the number of clusters that are investigated.



```
Clustered Instances

0       52 ( 35%)
1       44 ( 29%)
2       50 ( 33%)
3        4 (  3%)


Class attribute: class
Classes to Clusters:

  0  1  2  3  <-- assigned to cluster
  0  0 50  0 | Iris-setosa
 45  2  0  3 | Iris-versicolor
  7 42  0  1 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-virginica
Cluster 2 <-- Iris-setosa
Cluster 3 <-- No class

Incorrectly clustered instances :       13.0      8.6667 %
```

(a) Kmeans with 3 bins and k=4



```
=== Model and evaluation on training set ===

Clustered Instances

0       52 ( 35%)
1       44 ( 29%)
2       14 (  9%)
3        4 (  3%)
4       36 ( 24%)


Class attribute: class
Classes to Clusters:

  0  1  2  3  4  <-- assigned to cluster
  0  0 14  0 36 | Iris-setosa
 45  2  0  3  0 | Iris-versicolor
  7 42  0  1  0 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-virginica
Cluster 2 <-- No class
Cluster 3 <-- No class
Cluster 4 <-- Iris-setosa

Incorrectly clustered instances :       27.0     18      %
```

(b) Kmeans with 3 bins and k=5

Figure 6: Kmeans algorithm output using different number of clusters and the same number of bins

In Fig 6, we observe that the number of incorrectly clustered instances increases with a rate going from 6% in the optimal case to 18% for 5 clusters. In the case of 4 clusters, all instances of setosa are correctly identified and cluster 3 has very few elements. This could lead us to think already at this point that some of these elements could be clustered elsewhere or are maybe a group of outliers belonging to another class.

We now investigate the rules that are yielded by the apriori algorithm for the previous two cases.

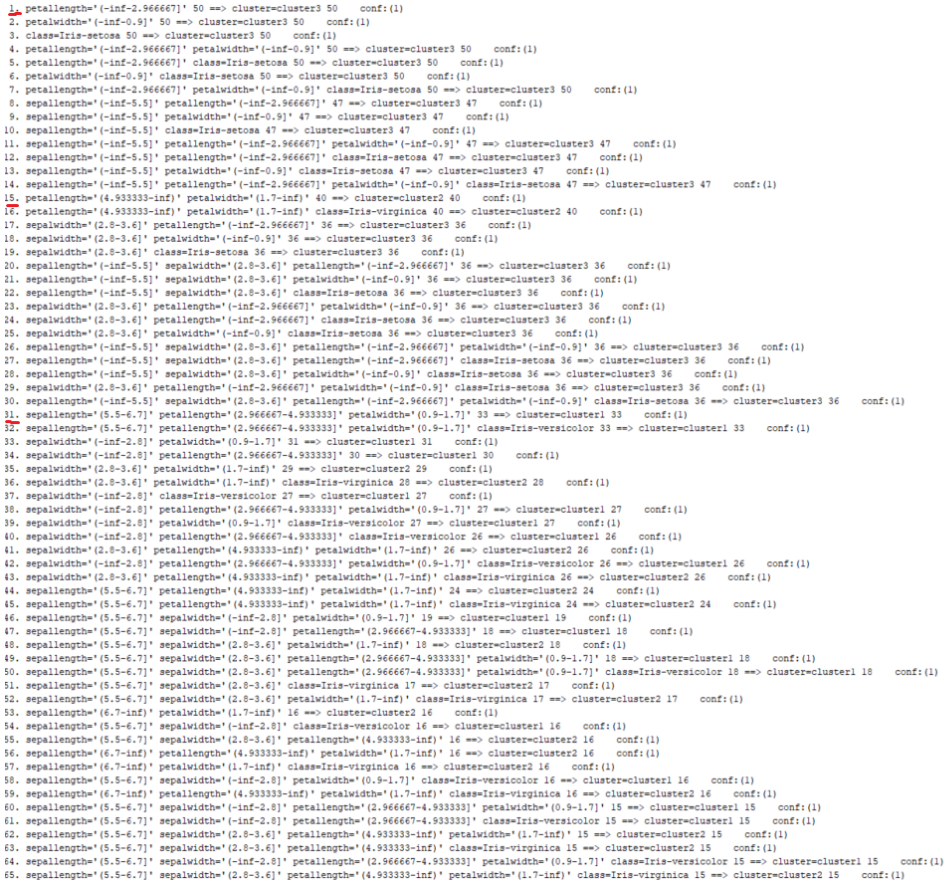


Figure 7: Apriori rules for k=4 and n=3 bins.

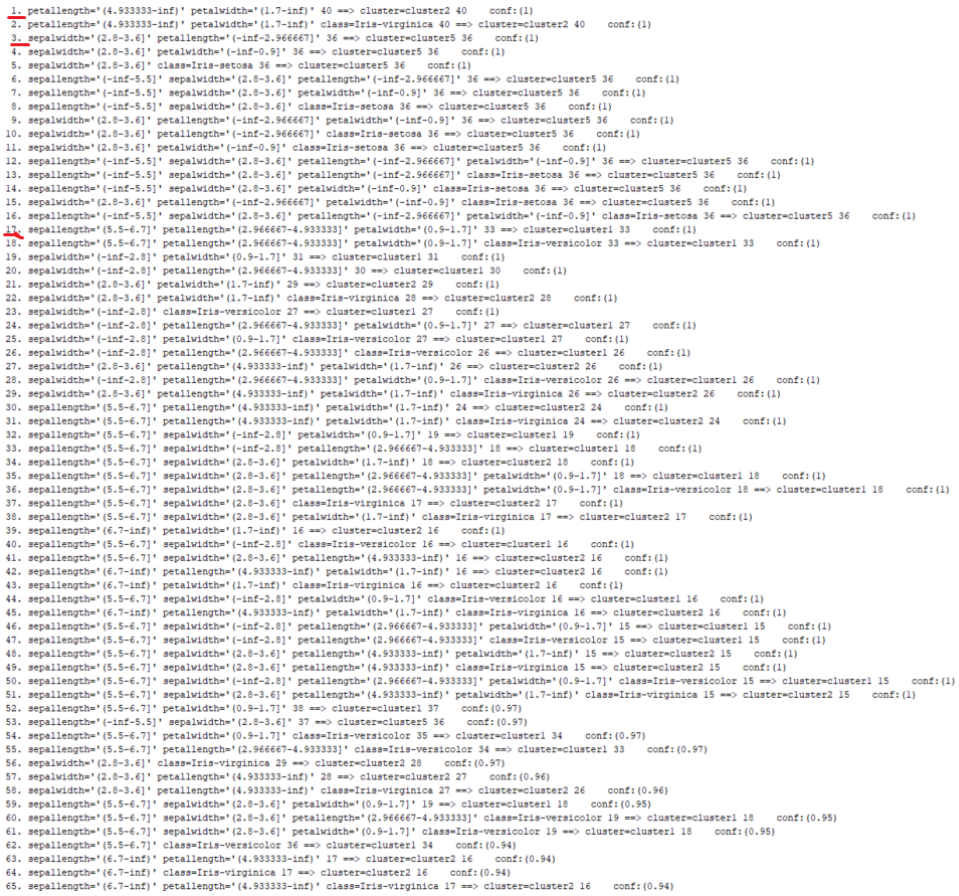


Figure 8: Apriori rules for k=5 and n=3 bins.

As stated in the previous section, the rules produced here are not good rules. They contain determinants that have the class attribute to establish a rule to a cluster consequent. Although they have high confidence, the support for some of these rules is low with many values below 20. Due to these rules being of poor quality, we are lead to think that the clustering is erroneous. The best rules for each clustering are marked in red. Moreover, we will note here that among the best rules, we do not find rules for some clusters. In Fig 7, we do not find rules for the 4th cluster that have high confidence since the clsuter only contains 4 individuals. In 8, similarly, we do not find rules for the thrid and fourth clusters.

When we increase the number of clusters, we find clusters with a very small number of cases. This makes it impossible for some cluster elements to have the required minimum support to establish rules for them using assosication analysis.

4 Discussion

In this work, we established that the apriori algorithm could be used to ascertain the quality of a clustering algorithm by analyzing the rules it yields. A primary glance at the Kmeans algorithm can indicate if there are issues in the number of bins or clusters. Too many clusters or bins will yield poor performance and many errors of clustering.