Association Analysis 1

Min-chun shih(shimi077) Saewon Jun(saeju204)

Cluster a given dataset and use association analysis to describe the clusters obtained.

Data set : The dataset consists of 50 samples from each of three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample, they are the length and the width of sepal and petal. As Apriori algorithm in Weka cannot cope with continuous attributes, we should discretize the iris dataset before starting the mining process.

Clustering : Apply SimpleKmeans clusterer to the data with 3 clusters (since there are 3 types of Iris flowers) and seed value 10.

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

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      %
```

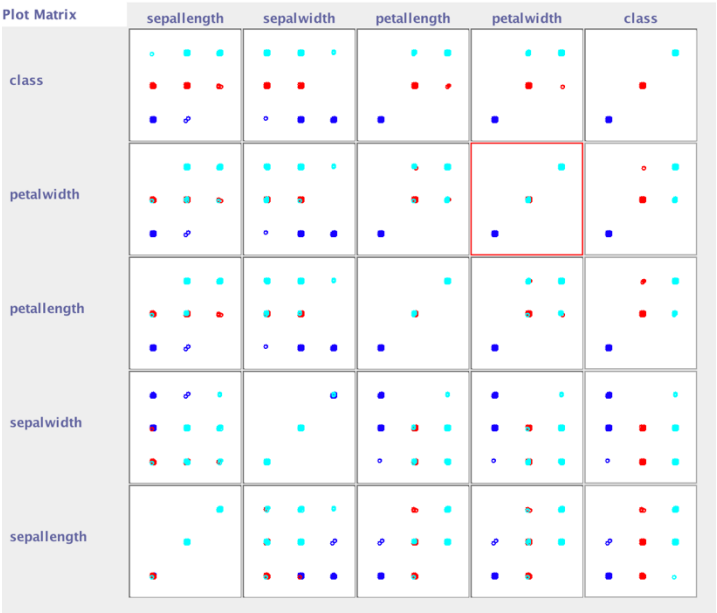
The figure above is the result of K-means clustering algorithm. It seems that the 3 types of iris flowers has classified into 3 cluster quite reasonably, with 9 instances has incorrectly clustered, which is only 6% of entire data.

Association Analysis : We are going to use Apriori algorithm here. Note that after the conjunctions of attribute-value pairs on the right- and left-hand sides of each rule, there is a number. That number indicates the support of the determinant and of the determinant plus the consequent.

```
Apriori
=======

Minimum support: 0.3 (45 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 13

Size of set of large itemsets L(2): 10

Size of set of large itemsets L(3): 5

Size of set of large itemsets L(4): 1

Best rules found:

 1. petalwidth='(-inf-0.9]' 50 ==> petallength='(-inf-2.966667]' 50    conf:(1)
 2. petallength='(-inf-2.966667]' 50 ==> petalwidth='(-inf-0.9]' 50    conf:(1)
 3. class=Iris-setosa 50 ==> petallength='(-inf-2.966667]' 50    conf:(1)
 4. petallength='(-inf-2.966667]' 50 ==> class=Iris-setosa 50    conf:(1)
 5. class=Iris-setosa 50 ==> petalwidth='(-inf-0.9]' 50    conf:(1)
 6. petalwidth='(-inf-0.9]' 50 ==> class=Iris-setosa 50    conf:(1)
 7. petalwidth='(-inf-0.9]' class=Iris-setosa 50 ==> petallength='(-inf-2.966667]' 50    conf:(1)
 8. petallength='(-inf-2.966667]' class=Iris-setosa 50 ==> petalwidth='(-inf-0.9]' 50    conf:(1)
 9. petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 50 ==> class=Iris-setosa 50    conf:(1)
10. class=Iris-setosa 50 ==> petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 50    conf:(1)
```

The figure above is the result of Apriori algorithm. Here the support means the number of instances that satisfy a rule, and confidence mean the proportions of instances that satisfy the left-hand side for which the right-hand side also holds. Mostly what you do is set the minimum value of confidence and seek for the rules with greatest support. confidence level to 1. There are 10 rules found from the data set with confidence level to 1. So we can say that the result is good enough.

Visualize : the data crosstabulated for each pair of attributes.



blue:Setosa, red:Versicolor, skyblue:Virginica

Describing clustering through association analysis : In this report, association analysis is used to describe the clusters found in the Iris dataset. The first thing to do is to create a new attribute that represents the cluster label assigned to each instance. To do this, apply AddCluster filter(with seed:10, number of clusters:3).

- Find rules that are accurate and such that the antecedent does not contain the class attribute and the consequent only contains the cluster attribute.
- Find such rules for the 3 clusters. This should help you to describe the instances grouped in each cluster.
- Repeat the exercise above with a different combination of clustering algorithm, number of clusters and/or number of bins in the discretization filter, in order to see whether you get better or worse results.

To see the results from different variations, we applied different bin number, number of cluster and different clustering algorithm.

With bin size=3, cluster size=3, Kmeans clustering algorithm

From the figure above, we tried to find the rules marked in red box.

```
Apriori
=======

Minimum support: 0.25 (37 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 15

Generated sets of large itemsets:

Size of set of large itemsets L(1): 11

Size of set of large itemsets L(2): 12

Size of set of large itemsets L(3): 6

Size of set of large itemsets L(4): 1

Best rules found:

 1. petallength='(-inf-2.966667]' 50 ==> cluster=cluster3 50    conf:(1)
 2. petalwidth='(-inf-0.9]' 50 ==> cluster=cluster3 50    conf:(1)
 3. class=Iris-setosa 50 ==> cluster=cluster3 50    conf:(1)
 4. petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 50 ==> cluster=cluster3 50    conf:(1)
 5. petallength='(-inf-2.966667]' class=Iris-setosa 50 ==> cluster=cluster3 50    conf:(1)
 6. petalwidth='(-inf-0.9]' class=Iris-setosa 50 ==> cluster=cluster3 50    conf:(1)
 7. petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' class=Iris-setosa 50 ==> cluster=cluster3 50    conf:(1)
 8. petallength='(2.966667-4.933333]' petalwidth='(0.9-1.7]' 48 ==> cluster=cluster1 48    conf:(1)
 9. sepallength='(-inf-5.5]' petallength='(-inf-2.966667]' 47 ==> cluster=cluster3 47    conf:(1)
10. sepallength='(-inf-5.5]' petalwidth='(-inf-0.9]' 47 ==> cluster=cluster3 47    conf:(1)
11. sepallength='(-inf-5.5]' class=Iris-setosa 47 ==> cluster=cluster3 47    conf:(1)
12. sepallength='(-inf-5.5]' petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 47 ==> cluster=cluster3 47    conf:(1)
13. sepallength='(-inf-5.5]' petallength='(-inf-2.966667]' class=Iris-setosa 47 ==> cluster=cluster3 47    conf:(1)
14. sepallength='(-inf-5.5]' petalwidth='(-inf-0.9]' class=Iris-setosa 47 ==> cluster=cluster3 47    conf:(1)
15. petallength='(2.966667-4.933333]' petalwidth='(0.9-1.7]' class=Iris-versicolor 47 ==> cluster=cluster1 47    conf:(1)
16. sepallength='(-inf-5.5]' petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' class=Iris-setosa 47 ==> cluster=cluster3 47    conf:(1)
17. petallength='(4.933333-inf)' petalwidth='(1.7-inf)' 40 ==> cluster=cluster2 40    conf:(1)
18. petallength='(4.933333-inf)' petalwidth='(1.7-inf)' class=Iris-virginica 40 ==> cluster=cluster2 40    conf:(1)
19. petalwidth='(0.9-1.7]' class=Iris-versicolor 49 ==> cluster=cluster1 48    conf:(0.98)
20. petallength='(2.966667-4.933333]' class=Iris-versicolor 48 ==> cluster=cluster1 47    conf:(0.98)
21. petalwidth='(0.9-1.7]' 54 ==> cluster=cluster1 52    conf:(0.96)
22. class=Iris-versicolor 50 ==> cluster=cluster1 48    conf:(0.96)
23. petallength='(2.966667-4.933333]' 54 ==> cluster=cluster1 51    conf:(0.94)
24. petalwidth='(1.7-inf)' 46 ==> cluster=cluster2 43    conf:(0.93)
25. petalwidth='(1.7-inf)' class=Iris-virginica 45 ==> cluster=cluster2 42    conf:(0.93)
```

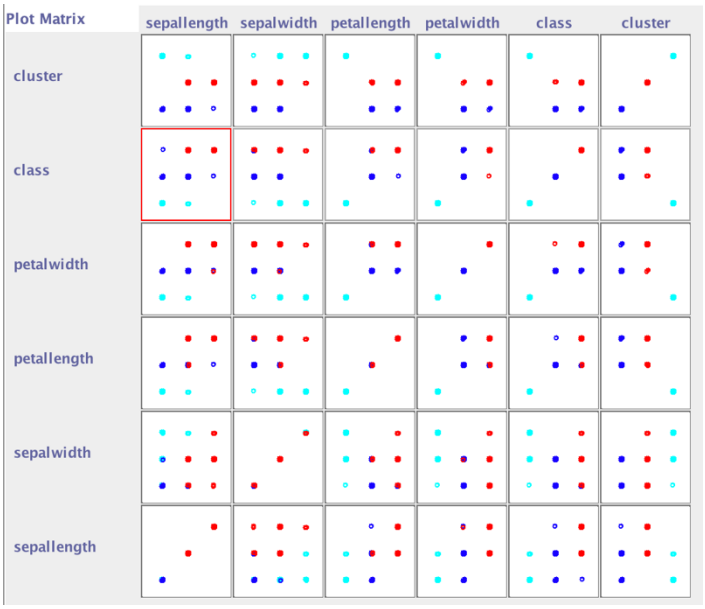
We were able to find the rules for 3 cluster among 25 best rules founded. Bin size equals to 3 seems reasonable for discretizing the data, and of course since there are 3 types of iris, 3 clusters seem reasonable enough.

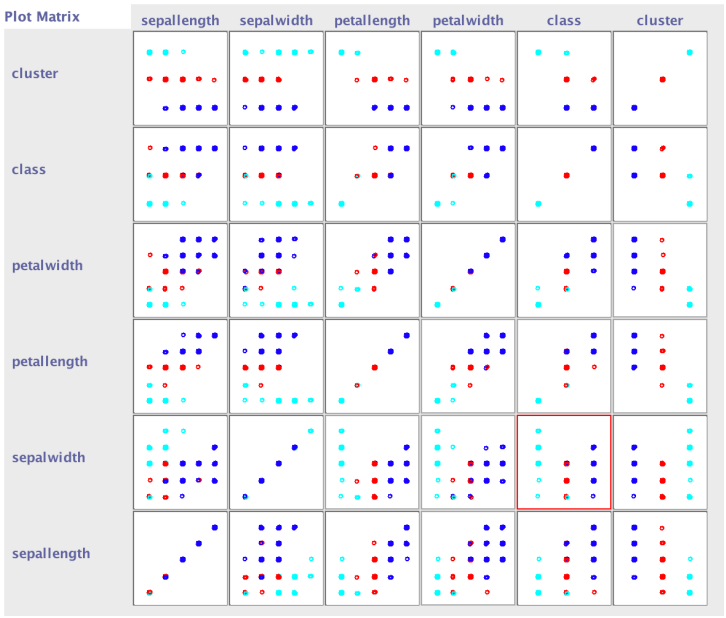
With bin size=5, cluster size=3, Kmeans clustering algorithm

```
Apriori
=======

Minimum support: 0.1 (15 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 18

Generated sets of large itemsets:

Size of set of large itemsets L(1): 22

Size of set of large itemsets L(2): 42

Size of set of large itemsets L(3): 29

Size of set of large itemsets L(4): 9

Size of set of large itemsets L(5): 1

Best rules found:

 1. petallength='(-inf-2.18]' 50 ==> cluster=cluster3 50    conf:(1)
 2. class=Iris-setosa 50 ==> cluster=cluster3 50    conf:(1)
 3. petallength='(-inf-2.18]' class=Iris-setosa 50 ==> cluster=cluster3 50    conf:(1)
 4. petalwidth='(-inf-0.58]' 49 ==> cluster=cluster3 49    conf:(1)
 5. petallength='(-inf-2.18]' petalwidth='(-inf-0.58]' 49 ==> cluster=cluster3 49    conf:(1)
 6. petalwidth='(-inf-0.58]' class=Iris-setosa 49 ==> cluster=cluster3 49    conf:(1)
 7. petallength='(-inf-2.18]' petalwidth='(-inf-0.58]' class=Iris-setosa 49 ==> cluster=cluster3 49    conf:(1)
 8. petallength='(4.54-5.72]' class=Iris-virginica 33 ==> cluster=cluster1 33    conf:(1)
 9. sepallength='(-inf-5.02]' petallength='(-inf-2.18]' 28 ==> cluster=cluster3 28    conf:(1)
10. sepallength='(-inf-5.02]' class=Iris-setosa 28 ==> cluster=cluster3 28    conf:(1)
11. sepallength='(-inf-5.02]' petallength='(-inf-2.18]' class=Iris-setosa 28 ==> cluster=cluster3 28    conf:(1)
12. sepallength='(-inf-5.02]' petalwidth='(-inf-0.58]' 27 ==> cluster=cluster3 27    conf:(1)
13. sepallength='(5.74-6.46]' petallength='(4.54-5.72]' 27 ==> cluster=cluster1 27    conf:(1)
14. sepalwidth='(2.96-3.44]' petallength='(-inf-2.18]' 27 ==> cluster=cluster3 27    conf:(1)
15. sepalwidth='(2.96-3.44]' petalwidth='(-inf-0.58]' 27 ==> cluster=cluster3 27    conf:(1)
16. sepalwidth='(2.96-3.44]' class=Iris-setosa 27 ==> cluster=cluster3 27    conf:(1)
17. sepallength='(-inf-5.02]' petallength='(-inf-2.18]' petalwidth='(-inf-0.58]' 27 ==> cluster=cluster3 27    conf:(1)
18. sepallength='(-inf-5.02]' petalwidth='(-inf-0.58]' class=Iris-setosa 27 ==> cluster=cluster3 27    conf:(1)
19. sepalwidth='(2.96-3.44]' petallength='(-inf-2.18]' petalwidth='(-inf-0.58]' 27 ==> cluster=cluster3 27    conf:(1)
20. sepalwidth='(2.96-3.44]' petallength='(-inf-2.18]' class=Iris-setosa 27 ==> cluster=cluster3 27    conf:(1)
21. sepalwidth='(2.96-3.44]' petalwidth='(-inf-0.58]' class=Iris-setosa 27 ==> cluster=cluster3 27    conf:(1)
22. sepallength='(-inf-5.02]' petallength='(-inf-2.18]' petalwidth='(-inf-0.58]' class=Iris-setosa 27 ==> cluster=cluster3 27    conf:(1)
23. sepalwidth='(2.96-3.44]' petallength='(-inf-2.18]' petalwidth='(-inf-0.58]' class=Iris-setosa 27 ==> cluster=cluster3 27    conf:(1)
24. sepalwidth='(2.96-3.44]' class=Iris-virginica 26 ==> cluster=cluster1 26    conf:(1)
25. sepalwidth='(2.96-3.44]' petallength='(4.54-5.72]' 25 ==> cluster=cluster1 25    conf:(1)
26. sepallength='(-inf-5.02]' sepalwidth='(2.96-3.44]' 22 ==> cluster=cluster3 22    conf:(1)
27. sepallength='(-inf-5.02]' sepalwidth='(2.96-3.44]' petallength='(-inf-2.18]' 22 ==> cluster=cluster3 22    conf:(1)
28. sepallength='(-inf-5.02]' sepalwidth='(2.96-3.44]' petalwidth='(-inf-0.58]' 22 ==> cluster=cluster3 22    conf:(1)
29. sepallength='(-inf-5.02]' sepalwidth='(2.96-3.44]' class=Iris-setosa 22 ==> cluster=cluster3 22    conf:(1)
30. sepallength='(-inf-5.02]' sepalwidth='(2.96-3.44]' petallength='(-inf-2.18]' petalwidth='(-inf-0.58]' 22 ==> cluster=cluster3 22    conf:(1)
31. sepallength='(-inf-5.02]' sepalwidth='(2.96-3.44]' petallength='(-inf-2.18]' class=Iris-setosa 22 ==> cluster=cluster3 22    conf:(1)
32. sepallength='(-inf-5.02]' sepalwidth='(2.96-3.44]' petalwidth='(-inf-0.58]' class=Iris-setosa 22 ==> cluster=cluster3 22    conf:(1)
33. sepallength='(-inf-5.02]' sepalwidth='(2.96-3.44]' petallength='(-inf-2.18]' petalwidth='(-inf-0.58]' class=Iris-setosa 22 ==> cluster=cluster3 22    conf:(1)
34. sepallength='(5.02-5.74]' petallength='(-inf-2.18]' 21 ==> cluster=cluster3 21    conf:(1)
35. sepallength='(5.02-5.74]' petalwidth='(-inf-0.58]' 21 ==> cluster=cluster3 21    conf:(1)
36. sepallength='(5.02-5.74]' class=Iris-setosa 21 ==> cluster=cluster3 21    conf:(1)
37. sepallength='(5.74-6.46]' class=Iris-virginica 21 ==> cluster=cluster1 21    conf:(1)
38. sepallength='(5.02-5.74]' petallength='(-inf-2.18]' petalwidth='(-inf-0.58]' 21 ==> cluster=cluster3 21    conf:(1)
39. sepallength='(5.02-5.74]' petallength='(-inf-2.18]' class=Iris-setosa 21 ==> cluster=cluster3 21    conf:(1)
40. sepallength='(5.02-5.74]' petalwidth='(-inf-0.58]' class=Iris-setosa 21 ==> cluster=cluster3 21    conf:(1)
41. sepallength='(5.02-5.74]' petallength='(-inf-2.18]' petalwidth='(-inf-0.58]' class=Iris-setosa 21 ==> cluster=cluster3 21    conf:(1)
42. petallength='(4.54-5.72]' petalwidth='(1.54-2.02]' 20 ==> cluster=cluster1 20    conf:(1)
43. sepallength='(5.74-6.46]' petallength='(4.54-5.72]' class=Iris-virginica 20 ==> cluster=cluster1 20    conf:(1)
44. sepallength='(5.02-5.74]' class=Iris-versicolor 18 ==> cluster=cluster2 18    conf:(1)
45. sepalwidth='(2.48-2.96]' petallength='(3.36-4.54]' 18 ==> cluster=cluster2 18    conf:(1)
```

As we increased the number of bin size from 3 to 5, we were able to figure out that except cluster 3, the support which is the number of instances that satisfy a rule has decreased for cluster 1 and cluster2. Also, we needed to increase the number of rules to get the rules for every cluster. Also, the visualized plot below indicates that classification is bit more unclear when bin size is 5(on the

right) compared to 3(on the left) - when you increase the jitter option, you can see that when bin size is 5, each point is messier. From this experiment, we were able to figure out that inappropriate size for bin could result in poor performance.





With bin size=3, cluster size=2, Kmeans clustering algorithm

```
Apriori
=======

Minimum support: 0.3 (45 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 10

Size of set of large itemsets L(2): 9

Size of set of large itemsets L(3): 5

Size of set of large itemsets L(4): 1

Best rules found:

 1. sepallength='(-inf-5.5]' 59 ==> cluster=cluster1 59    conf:(1)
 2. petallength='(2.966667-4.933333]' petalwidth='(0.9-1.7]' 48 ==> cluster=cluster1 48    conf:(1)
 3. sepallength='(-inf-5.5]' petallength='(-inf-2.966667]' 47 ==> cluster=cluster1 47    conf:(1)
 4. sepallength='(-inf-5.5]' petalwidth='(-inf-0.9]' 47 ==> cluster=cluster1 47    conf:(1)
 5. sepallength='(-inf-5.5]' class=Iris-setosa 47 ==> cluster=cluster1 47    conf:(1)
 6. sepallength='(-inf-5.5]' petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' 47 ==> cluster=cluster1 47    conf:(1)
 7. sepallength='(-inf-5.5]' petallength='(-inf-2.966667]' class=Iris-setosa 47 ==> cluster=cluster1 47    conf:(1)
 8. sepallength='(-inf-5.5]' petalwidth='(-inf-0.9]' class=Iris-setosa 47 ==> cluster=cluster1 47    conf:(1)
 9. petallength='(2.966667-4.933333]' petalwidth='(0.9-1.7]' class=Iris-versicolor 47 ==> cluster=cluster1 47    conf:(1)
10. sepallength='(-inf-5.5]' petallength='(-inf-2.966667]' petalwidth='(-inf-0.9]' class=Iris-setosa 47 ==> cluster=cluster1 47    conf:(1)
11. petalwidth='(1.7-inf)' 46 ==> cluster=cluster2 46    conf:(1)
12. petallength='(2.966667-4.933333]' class=Iris-versicolor 48 ==> cluster=cluster1 47    conf:(0.98)
13. class=Iris-virginica 50 ==> cluster=cluster2 48    conf:(0.96)
14. petalwidth='(0.9-1.7]' class=Iris-versicolor 49 ==> cluster=cluster1 47    conf:(0.96)
15. petallength='(-inf-2.966667]' 50 ==> cluster=cluster1 47    conf:(0.94)
```

At preprocess tab, you can check the how the data is assigned to each newly created clusters. In this case, 96 data points were assigned to cluster 1, and 54 data points were assigned to cluster2. The first two rule with instance 59 and 48 seem to be not enough to explain the first cluster where 96 data points are assigned. Also, considering that the original data has 3 types of iris, it would be

ideal to have 3 clusters. Inappropriate choice of cluster size might end up underfitting(in this case) or over fitting which will generate poor rules.