

Association Analysis -1

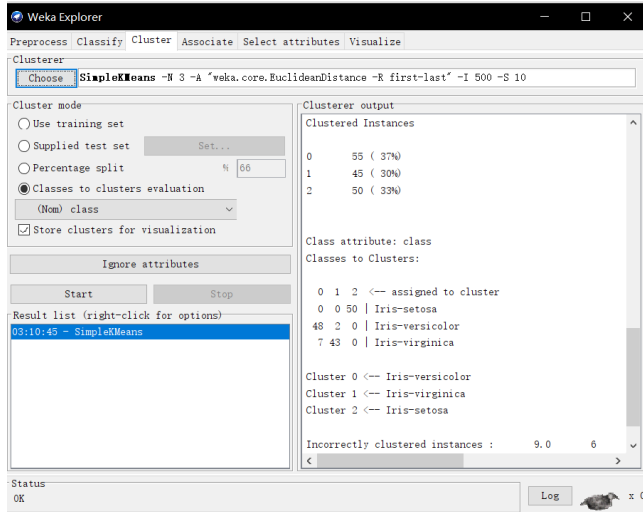
Zijie Feng & Jiawei Wu

2019-3-22

Discretization

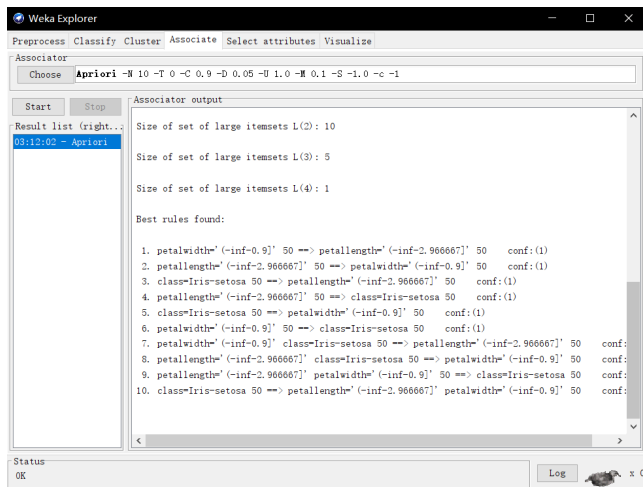
Since the attributes of Iris data are numerical, we have to discretize them into 3 intervals firstly.

Clustering



We use **SampleKMeans** to cluster our data into 3 different new clusters. The clustering result is a little different from the original classes.

Association Analysis



For association analysis, we use the *Apriori* algorithm with default hyper-parameters to analyze our data.

Visualization



The blue points represent Setosa, red points represent Versicolor and cyan represent Virginica.

Describing clustering through association analysis

Result 1: bins=3, SimpleKMeans, K=3

Generated sets of large itemsets:

Size of set of large itemsets L(1): 14

Size of set of large itemsets L(2): 20

Size of set of large itemsets L(3): 15

Size of set of large itemsets L(4): 6

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

```
Associator output
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. sepalwidth'(-inf-5.5]' petallength'(-inf-2.966667]' 47 ==> cluster=cluster3 47 conf:(1)
10. sepalwidth'(-inf-5.5]' petalwidth'(-inf-0.9]' 47 ==> cluster=cluster3 47 conf:(1)
11. sepalwidth'(-inf-5.5]' class=Iris-setosa 47 ==> cluster=cluster3 47 conf:(1)
12. sepalwidth'(-inf-5.5]' petallength'(-inf-2.966667]' petalwidth'(-inf-0.9]' 47 ==> cluster=cluster3 47 conf:(1)
13. sepalwidth'(-inf-5.5]' petallength'(-inf-2.966667]' class=Iris-setosa 47 ==> cluster=cluster3 47 conf:(1)
14. sepalwidth'(-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. sepalwidth'(-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. sepalwidth'(2.8-3.6]' petallength'(-inf-2.966667]' 36 ==> cluster=cluster3 36 conf:(1)
20. sepalwidth'(2.8-3.6]' petalwidth'(-inf-0.9]' 36 ==> cluster=cluster3 36 conf:(1)
21. sepalwidth'(2.8-3.6]' class=Iris-setosa 36 ==> cluster=cluster3 36 conf:(1)
22. sepalwidth'(-inf-5.5]' sepalwidth'(2.8-3.6]' petallength'(-inf-2.966667]' 36 ==> cluster=cluster3 36 conf:(1)
23. sepalwidth'(-inf-5.5]' sepalwidth'(2.8-3.6]' petalwidth'(-inf-0.9]' 36 ==> cluster=cluster3 36 conf:(1)
24. sepalwidth'(-inf-5.5]' sepalwidth'(2.8-3.6]' class=Iris-setosa 36 ==> cluster=cluster3 36 conf:(1)
25. sepalwidth'(2.8-3.6]' petallength'(-inf-2.966667]' petalwidth'(-inf-0.9]' 36 ==> cluster=cluster3 36 conf:(1)
26. sepalwidth'(2.8-3.6]' petallength'(-inf-2.966667]' class=Iris-setosa 36 ==> cluster=cluster3 36 conf:(1)
27. sepalwidth'(2.8-3.6]' petalwidth'(-inf-0.9]' class=Iris-setosa 36 ==> cluster=cluster3 36 conf:(1)
28. sepalwidth'(-inf-5.5]' sepalwidth'(2.8-3.6]' petallength'(-inf-2.966667]' petalwidth'(-inf-0.9]' 36 ==> cluster=cluster3 36 conf:(1)
29. sepalwidth'(-inf-5.5]' sepalwidth'(2.8-3.6]' petallength'(-inf-2.966667]' class=Iris-setosa 36 ==> cluster=cluster3 36 conf:(1)
```

Based on the visualization step, each original class has their own intervals for different attributes. Especially for three clusters, it is reasonable to find their one-one correspondence to original classes quickly. We can find cluster 3 is connected with petallength= $(-\text{inf}-2.966667]$ and also with petalwidth= $(-\text{inf}-0.9]$. Actually, we can find class Setosa and cluster 3 are connected as well by rule 11 & 12.

For cluster 2, it connects with petallength= $(4.933333-\text{inf})$ petalwidth= $(1.7-\text{inf})$. It seems cluster 2 own most parts of class Virginica. Then for cluster 1, it connects with petallength= $(2.966667-4.933333]$ petalwidth= $(0.9-1.7]$, which represents Versicolor.

Result 2: bins=6, SimpleKMeans, K=3

Generated sets of large itemsets:

Size of set of large itemsets L(1): 20

Size of set of large itemsets L(2): 33

Size of set of large itemsets L(3): 24

Size of set of large itemsets L(4): 8

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

Associator output	
1. petallength' (-inf-1.983333]' 50 ==> cluster=cluster1 50	conf:(1)
2. class-Iris-setosa 50 ==> cluster=cluster1 50	conf:(1)
3. petallength' (-inf-1.983333]' class-Iris-setosa 50 ==> cluster=cluster1 50	conf:(1)
4. petalwidth' (-inf-0.5]' 49 ==> cluster=cluster1 49	conf:(1)
5. petallength' (-inf-1.983333]' petalwidth' (-inf-0.5]' 49 ==> cluster=cluster1 49	conf:(1)
6. petalwidth' (-inf-0.5]' class-Iris-setosa 49 ==> cluster=cluster1 49	conf:(1)
7. petallength' (-inf-1.983333]' petalwidth' (-inf-0.5]' class-Iris-setosa 49 ==> cluster=cluster1 49	conf:(1)
8. sepalwidth' (4.9-5.5]' petallength' (-inf-1.983333]' 27 ==> cluster=cluster1 27	conf:(1)
9. sepalwidth' (4.9-5.5]' class-Iris-setosa 27 ==> cluster=cluster1 27	conf:(1)
10. sepalwidth' (4.9-5.5]' petallength' (-inf-1.983333]' class-Iris-setosa 27 ==> cluster=cluster1 27	conf:(1)
11. sepalwidth' (4.9-5.5]' petalwidth' (-inf-0.5]' 26 ==> cluster=cluster1 26	conf:(1)
12. sepalwidth' (4.9-5.5]' petallength' (-inf-1.983333]' petalwidth' (-inf-0.5]' 26 ==> cluster=cluster1 26	conf:(1)
13. sepalwidth' (4.9-5.5]' petalwidth' (-inf-0.5]' class-Iris-setosa 26 ==> cluster=cluster1 26	conf:(1)
14. sepalwidth' (4.9-5.5]' petallength' (-inf-1.983333]' petalwidth' (-inf-0.5]' class-Iris-setosa 26 ==> cluster=cluster1 26	conf:(1)
15. sepalwidth' (5.5-6.1]' class-Iris-versicolor 23 ==> cluster=cluster2 23	conf:(1)
16. sepalwidth' (5.5-6.1]' petallength' (3.95-4.933333]' 21 ==> cluster=cluster2 21	conf:(1)
17. sepalwidth' (-inf-4.9]' petallength' (-inf-1.983333]' 20 ==> cluster=cluster1 20	conf:(1)
18. sepalwidth' (-inf-4.9]' petalwidth' (-inf-0.5]' 20 ==> cluster=cluster1 20	conf:(1)
19. sepalwidth' (-inf-4.9]' class-Iris-setosa 20 ==> cluster=cluster1 20	conf:(1)
20. sepalwidth' (6.1-6.7]' petallength' (4.933333-5.916667]' 20 ==> cluster=cluster3 20	conf:(1)
21. sepalwidth' (-inf-4.9]' petallength' (-inf-1.983333]' petalwidth' (-inf-0.5]' 20 ==> cluster=cluster1 20	conf:(1)
22. sepalwidth' (-inf-4.9]' petallength' (-inf-1.983333]' class-Iris-setosa 20 ==> cluster=cluster1 20	conf:(1)
23. sepalwidth' (-inf-4.9]' petalwidth' (-inf-0.5]' class-Iris-setosa 20 ==> cluster=cluster1 20	conf:(1)
24. sepalwidth' (-inf-4.9]' petallength' (-inf-1.983333]' petalwidth' (-inf-0.5]' class-Iris-setosa 20 ==> cluster=cluster1 20	conf:(1)
25. sepalwidth' (3.2-3.6]' petallength' (-inf-1.983333]' 19 ==> cluster=cluster1 19	conf:(1)
26. sepalwidth' (3.2-3.6]' class-Iris-setosa 19 ==> cluster=cluster1 19	conf:(1)
27. sepalwidth' (6.1-6.7]' petallength' (4.933333-5.916667]' class-Iris-virginica 19 ==> cluster=cluster3 19	conf:(1)
28. sepalwidth' (3.2-3.6]' petallength' (-inf-1.983333]' class-Iris-setosa 19 ==> cluster=cluster1 19	conf:(1)
29. sepalwidth' (2.8-3.2]' petallength' (4.933333-5.916667]' 18 ==> cluster=cluster3 18	conf:(1)
<	

Compared with result 1, some rules in result 1 are separated into several rules in result 2. So it seems that result 2 can give us more detailed rules for each clustering.

However, the growth of bins may increase the difficulty for analysis. Similar to overfitting in classification tree, 'more bins' exponentially increases the number of best rules. This situation might ask people more time for analysis (e.g. find each cluster in best rules list).

Result 3: bins=2, SimpleKMeans, K=2

Generated sets of large itemsets:

Size of set of large itemsets L(1): 14

Size of set of large itemsets L(2): 34

Size of set of large itemsets L(3): 37

Size of set of large itemsets L(4): 18

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

```

Associator output
1. sepalength=(-inf-6.1]' petalwidth=(-inf-1.3]' 74 ==> cluster=cluster1 74 conf:(1)
2. petalength=(3.95-inf)' petalwidth=(1.3-inf)' 71 ==> cluster=cluster2 71 conf:(1)
3. petalength=(-inf-3.95]' 61 ==> cluster=cluster1 61 conf:(1)
4. sepalength=(-inf-6.1]' petalength=(-inf-3.95]' 61 ==> cluster=cluster1 61 conf:(1)
5. sepalwidth=(-inf-3.2]' petalength=(3.95-inf)' petalwidth=(1.3-inf)' 61 ==> cluster=cluster2 61 conf:(1)
6. petalength=(-inf-3.95]' petalwidth=(-inf-1.3]' 60 ==> cluster=cluster1 60 conf:(1)
7. sepalength=(-inf-6.1]' petalength=(-inf-3.95]' petalwidth=(-inf-1.3]' 60 ==> cluster=cluster1 60 conf:(1)
8. sepalength=(6.1-inf)' 55 ==> cluster=cluster2 55 conf:(1)
9. sepalength=(6.1-inf)' petalength=(3.95-inf)' 55 ==> cluster=cluster2 55 conf:(1)
10. sepalength=(6.1-inf)' petalwidth=(1.3-inf)' 51 ==> cluster=cluster2 51 conf:(1)
11. sepalength=(6.1-inf)' petalength=(3.95-inf)' petalwidth=(1.3-inf)' 51 ==> cluster=cluster2 51 conf:(1)
12. class=Iris-setosa 50 ==> cluster=cluster1 50 conf:(1)
13. class=Iris-virginica 50 ==> cluster=cluster2 50 conf:(1)
14. sepalength=(-inf-6.1]' class=Iris-setosa 50 ==> cluster=cluster1 50 conf:(1)
15. petalength=(-inf-3.95]' class=Iris-setosa 50 ==> cluster=cluster1 50 conf:(1)
16. petalength=(3.95-inf)' class=Iris-virginica 50 ==> cluster=cluster2 50 conf:(1)
17. petalwidth=(-inf-1.3]' class=Iris-setosa 50 ==> cluster=cluster1 50 conf:(1)
18. petalwidth=(1.3-inf)' class=Iris-virginica 50 ==> cluster=cluster2 50 conf:(1)
19. sepalength=(-inf-6.1]' petalength=(-inf-3.95]' class=Iris-setosa 50 ==> cluster=cluster1 50 conf:(1)
20. sepalength=(-inf-6.1]' petalwidth=(-inf-1.3]' class=Iris-setosa 50 ==> cluster=cluster1 50 conf:(1)
21. petalength=(-inf-3.95]' petalwidth=(-inf-1.3]' class=Iris-setosa 50 ==> cluster=cluster1 50 conf:(1)
22. petalength=(3.95-inf)' petalwidth=(1.3-inf)' class=Iris-virginica 50 ==> cluster=cluster2 50 conf:(1)
23. sepalength=(-inf-6.1]' petalength=(-inf-3.95]' petalwidth=(-inf-1.3]' class=Iris-setosa 50 ==> cluster=cluster1 50 conf:(1)
24. sepalength=(6.1-inf)' sepalwidth=(-inf-3.2]' 46 ==> cluster=cluster2 46 conf:(1)
25. sepalength=(6.1-inf)' sepalwidth=(-inf-3.2]' petalength=(3.95-inf)' 46 ==> cluster=cluster2 46 conf:(1)
26. sepalwidth=(-inf-3.2]' class=Iris-virginica 42 ==> cluster=cluster2 42 conf:(1)
27. sepalength=(-inf-6.1]' sepalwidth=(-inf-3.2]' petalwidth=(-inf-1.3]' 42 ==> cluster=cluster1 42 conf:(1)
28. sepalength=(6.1-inf)' sepalwidth=(-inf-3.2]' petalwidth=(1.3-inf)' 42 ==> cluster=cluster2 42 conf:(1)
29. sepalwidth=(-inf-3.2]' petalength=(3.95-inf)' class=Iris-virginica 42 ==> cluster=cluster2 42 conf:(1)

```

In the third association analysis, we use bins=2 and K=2 for SimpleKMeans. Cluster 1 and 2 have 75 elements respectively. It is quite good to show the correspondence. The result shows that sepalength='(-inf-6.1]' petalwidth='(-inf-1.3]' has sup=74 with cluster 1, and petalength='(3.95-inf)' petalwidth='(1.3-inf)' has sup=71 with cluster 2.

Additionally, cluster 1 contains all the elements from class Setosa, and cluster 2 contains all from class Virginica. We find the first class Versicolor at 59th line:

```
59. sepalength=(-inf-6.1]' petalwidth=(-inf-1.3]' class=Iris-versicolor 24 ==> cluster=cluster1 24
```

It seems this class Versicolor is separated into 2 new clusters. Such situation is not preferable because we can find such properties is the same as in the 1st line:

```
1. sepalength=(-inf-6.1]' petalwidth=(-inf-1.3]' 74 ==> cluster=cluster1 74
```

This is confused. Thus, the small number of bins or K might results in the ambiguous and underfitting classification. it decreases the accuracy of clustering, especially when the number of real classes is larger than bins and K. It would be more efficient if we can choose appropriate hyper-parameters for clustering and association analysis.