Association Analysis 2

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

Appreciate the importance of the distance metric used within the clustering algorithm

Data set : This is an artificial dataset with 124 instances, each described by 6 discrete attributes and a binary class attribute. There are 3 versions of the so-called monk problem: We are using the first version in this exercise.

Target Concepts associated to the MONK's problem
MONK-1: (a1 = a2) or (a5 = 1)

Clusters may not correspond to classes :

- First, cluster the data with different algorithms and number of clusters. Use the Clusters to class evaluation model to see whether the clustering algorithm is able to discover the class division existing in the data.

Here, we tried to discover the class division based on the 6 other attributes by applying EM algorithm and Kmeans algorithm with different number of clusters. We selected classes to cluster evaluation to evaluate the quality of classification.

EM with 3 clusters

```
Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
 30  0 32 | 0
 20 21 21 | 1

Cluster 0 <-- No class
Cluster 1 <-- 1
Cluster 2 <-- 0

Incorrectly clustered instances :       71.0     57.2581 %
```

EM with 6 clusters

```
Class attribute: class
Classes to Clusters:

  0  1  2  3  4  5  <-- assigned to cluster
 13 22  5 10 10  2 | 0
 18  4 16  6  4 14 | 1

Cluster 0 <-- 1
Cluster 1 <-- 0
Cluster 2 <-- No class
Cluster 3 <-- No class
Cluster 4 <-- No class
Cluster 5 <-- No class

Incorrectly clustered instances :       84.0     67.7419 %
```

simpleKmeans with 3 clusters

```
Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
 33 17 12 | 0
 26 21 15 | 1

Cluster 0 <-- 0
Cluster 1 <-- 1
Cluster 2 <-- No class

Incorrectly clustered instances :       70.0     56.4516 %
```

simpleKmeans with 6 clusters

```
Class attribute: class
Classes to Clusters:

  0  1  2  3  4  5  <-- assigned to cluster
 11 13 12  5 10 11 | 0
 15 16 11 10  7  3 | 1

Cluster 0 <-- No class
Cluster 1 <-- 1
Cluster 2 <-- 0
Cluster 3 <-- No class
Cluster 4 <-- No class
Cluster 5 <-- No class

Incorrectly clustered instances :       96.0     77.4194 %
```

As you can see from the figures above, the percentage of incorrectly clustered instances are very high. We can conclude that clustering algorithm is not enough to discover the existing class division in the data

- Why can the clustering algorithms not find a clustering that matches the class division in the database?

When you check from the preprocess tab , you can easily see that each attributes are sharing each classes in similar portion, so it is hard to find any matching or classifying class division in the database.

- Use association analysis to find a set of rules that are able to accurately predict the class label from the rest of the attributes. (minimum support of 0.05 and a maximum number of rules of 19)
  - ✓ Note that the class attribute is binary, so it suffices to find rules that accurately predict class 1, i.e. an instance is assigned to class 0 if it is not assigned to class 1.
  - ✓ Try to find as few rules predicting class 1 as possible, i.e. try to remove redundant rules. Hopefully, you will be able to perfectly describe class 1 with only 4 rules.

Apriori algorithm

```
Best rules found:

 1. attribute#5=1 29 ==> class=1 29    conf:(1)
 2. attribute#1=3 attribute#2=3 17 ==> class=1 17    conf:(1)
 3. attribute#3=1 attribute#5=1 17 ==> class=1 17    conf:(1)
 4. attribute#5=1 attribute#6=1 16 ==> class=1 16    conf:(1)
 5. attribute#1=2 attribute#2=2 15 ==> class=1 15    conf:(1)
 6. attribute#1=3 attribute#5=1 13 ==> class=1 13    conf:(1)
 7. attribute#5=1 attribute#6=2 13 ==> class=1 13    conf:(1)
 8. attribute#2=3 attribute#5=1 12 ==> class=1 12    conf:(1)
 9. attribute#3=2 attribute#5=1 12 ==> class=1 12    conf:(1)
10. attribute#1=3 attribute#2=3 attribute#6=2 12 ==> class=1 12    conf:(1)
11. attribute#4=1 attribute#5=1 11 ==> class=1 11    conf:(1)
12. attribute#1=2 attribute#5=1 10 ==> class=1 10    conf:(1)
13. attribute#2=2 attribute#5=1 10 ==> class=1 10    conf:(1)
14. attribute#1=1 attribute#2=1 9 ==> class=1 9    conf:(1)
15. attribute#4=2 attribute#5=1 9 ==> class=1 9    conf:(1)
16. attribute#4=3 attribute#5=1 9 ==> class=1 9    conf:(1)
17. attribute#1=2 attribute#2=2 attribute#3=1 9 ==> class=1 9    conf:(1)
18. attribute#1=3 attribute#2=3 attribute#3=1 9 ==> class=1 9    conf:(1)
19. attribute#3=1 attribute#5=1 attribute#6=1 9 ==> class=1 9    conf:(1)
```

The red box indicates the 4 rules which meet *MONK-1: (a1 = a2) or (a5 = 1)*

- Finally, would you say that the clustering algorithms fail or perform poorly for the monk1 dataset? Why or why not?

We would say clustering algorithms fails in this case, and the reason is that the data we have is binary. When we are applying clustering algorithm, we use distance method to measure how they are different from each other and how they can be classified. However, when data is binary, distance metric would not work.