David Björelind, davbj395
Philip Kantola, phika529
TDDD41 - Data Mining

Report: Lab 2 - 2

First, we try to identify this rule with two types of clustering algorithms;

SimpleKMeans:
clusters: 2
Incorrectly clustered instances : 59.0 47.5806 %

We tried other number of clusters as 3 but it is gave worse result since the dataset only consists of 2 classes. One of the three clusters was not assigned to a class in the evaluation.
Incorrectly clustered instances : 70.0 56.4516 %

DBSCAN:
Epsilon need to be greater than 1 since there are no point closer to each other than that. DBSCAN can generate outliers which might have given better result.

Epsilon: 1.2
minPoints: 5
Incorrectly clustered instances : 57.0 45.9677 %

Maybe clustering algorithms cannot properly find the correct pattern in the data. This is discussed further at the end of this document.

Association analysis:
The following rules were obtained when running apriori algorithm with minSupport = 0.05 and numRules = 19.

1. **attribute#5=1 29 ==> class=1 29conf:(1)**
2. **attribute#1=3 attribute#2=3 17 ==> class=1 17conf:(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 15conf:(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)**

David Björelind, davbj395
Philip Kantola, phika529
TDDD41 - Data Mining
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 first rule is of most significance, since it has higher support than the rest of the rules. This rule a5 = 1 -> class1 has 100% confidence, which means that zero data points in class0 follows this rule. This also means that following rules containing attribute#5 becomes redundant. Meaning they are “overwritten” by rule 1. Rules 2, 5, 14, 10, 17 and 18 are then the remaining rules, 10,17 and 18 are supersets of 2,5, and 14 and therefore redundant as well. We are left with the following rules; 1, 2, 5 and 14. These rules can be described in the following way;

$(a1=a2) \vee (a5=1)$

Which is the true concept behind monk-1.

These rules form a cluster that is not of a regular shape. If the dataset consisted of 3 attributed, the rules would be apparent as two planes in a three dimensional space. This gives an intuitive idea behind the irregular shape of the cluster that the clustering algorithms we tried were not successful. Clustering algorithms and association algorithms works in different ways, while clustering algs. clusters rows in the dataset, association algs tries to clusters columns, or so to say, tries to find frequent attributes associations. Out of this we draw the conclusion that the monk problems are better suited for being solved with an association algorithm such as Apriori. Awesome.