

732A75 Data Mining Lab-3

Anubhav Dikshit(anudi287) and Nahid Farazmand (nahfa911)

1 March 2019

Clustering

```
Time taken to build model (full training data) : 0 seconds

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

Clustered Instances

0      77 ( 62%)
1      47 ( 38%)

Class attribute: class
Classes to Clusters:

  0  1 <-- assigned to cluster
40 22 | 0
37 25 | 1

Cluster 0 <-- 0
Cluster 1 <-- 1

Incorrectly clustered instances :      59.0      47.5806 %
```

Figure 1: KMeans clustering

For k-Means, we only have the options of Euclidean or Manhattan distance. Both are concepts of distance that are very intuitive for humans, however they seem to fail with the monk1 dataset. For both measurements, it turns out that 47.5806% of the instances are incorrectly clustered. When using MakeDensityBasedClusterer, we get 45.9677%, not much of an improvement. For both algorithms we used the default parameters.

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

When plotting attributes against classification, it can be seen that there is considerable overlap and no well-defined boundary between the two classes. Under these conditions it is understandably quite hard to cluster, since similar points are supposed to lie close to each other in order for the algorithms to succeed. Possible solutions could involve some kind of preprocessing: for instance, to apply a kernel function to the attributes in order to artificially create a separation between the two classes, or similarly to use new attributes calculated as convenient functions of subsets of original attributes.

```

Time taken to build model (full training data) : 0.02 seconds

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

Clustered Instances

0      83 ( 67%)
1      41 ( 33%)

Log likelihood: -6.09856

Class attribute: class
Classes to Clusters:

  0  1 <-- assigned to cluster
44 18 | 0
39 23 | 1

Cluster 0 <-- 0
Cluster 1 <-- 1

Incorrectly clustered instances :      57.0      45.9677 %

```

Figure 2: Density Based clustering

Table 1: Association rules

Parameters	Cluster	Occurences	Confidence
attribute 5 = 1	1	29	1
attribute 1 = 3 attribute 2 = 3	1	17	1
attribute 1 = 2 attribute 2 = 2	1	15	1
attribute 1 = 1 attribute 2 = 1	1	9	1

Would you say that the clustering algorithms fail or perform poorly for the monk1 dataset? Why or why not?

As we noted above it was the characteristics of the data that gave the algorithm a hard time. If we supply the data exactly as is, then another idea to improve the results would be to find a different distance function that is applied within the algorithm itself, taking care of respecting all the constraints for such a function. It may require some amount of domain knowledge in order to engineer such a feature.

Association Analysis

Association analysis does not require points that have the same classification to be close together. Therefore it may be easier to find rules that correctly classify points, so long as there actually is a pattern to be found. We remove those rules where the antecedent is a superset of the antecedent of another rule, because then it is redundant. This is not the case in general but only when the more general rules have 100% confidence. This is because if the confidence was lower, then the rules with a superset-antecedent would provide additional information and therefore it would not be wise to discard them. For cluster 1, we end up with the following

rules. These rules then encompass both clusters since everything that does not belong to cluster 1 must belong to cluster 0. This is a significantly better result than for the clustering algorithm.