TDDD41 - Lab 3

Martin Estgren, Alice Reinaudo

March 9, 2017

1 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 *monk*1 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.

1.1 Why can the clustering algorithm not find a division that matches the classes?

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.

1.2 Would you say that it goes poorly for monk1? 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 function.

2 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.

```
bla
```

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 that for the clustering algorithm.