Lab 3

Stefano Toffol (steto820), Mim Tekin Kemal (mimte666)

13 March, 2019

Dataset description

In this lab we are asked to test on the `MONK1` dataset a series of clustering algorithm and then run an association analysis in order to find reevant links between each observation. This dataset was in fact artificially created with the purpose of having a series of observations on which to test the efficacy of different algorithms.

The MONK's problems rely on the an artificial robot domain, in which robots are described by six different attributes [Wnek et al. (1990)]. These are:

$$
\begin{aligned}
x_1&: \text{head\_shape} && \in && \text{round, square, octagon}\\
x_2&: \text{body\_shape} && \in && \text{round, square, octagon}\\
x_3&: \text{is\_smiling} && \in && \text{yes, no}\\
x_4&: \text{holding} && \in && \text{sword, baloon, flag}\\
x_5&: \text{jacket\_color} && \in && \text{red, yellow, green, blue}\\
x_6&: \text{has\_tie} && \in && \text{yes, no}
\end{aligned}
$$

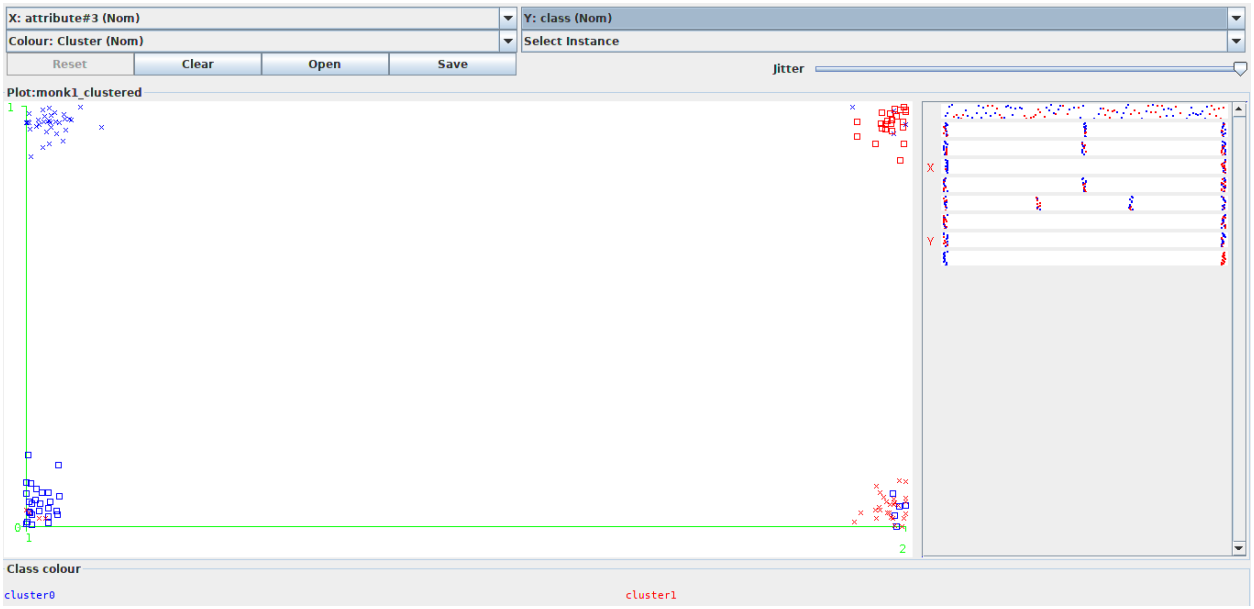
The researchers have generated three binary classification problems useing various subset of the original dataset, which consists of 432 total observations, and combining some of the features together in a logical expression to create a class variable, the response variable of the classification problem.
We are dealing with the first `MONK` problem, a dataset of 126 observations where the *class* column is given by the following logical exppression: `(head_shape == body_shape) or (jacket_color == "red")`. For this reason we expect the clustering algorithms to poorly perform, since the domain of the data is exclusively discrete, the class is given by a non-linear combination of the variables and, most importantly, half of the variables are actually useless, which may lead the algorithms used to try modelling uninteresting behaviours. The purpose of this analysis is to distinguish the observations according to this logical rule but without specifying it to our algorithms, pretending we do not know it, and using exclusively the data provided.

Clustering algorithms

First of all we start by applying some of the knowns clustering algorithms to the data. Having a dataset made exclusively by categorical variables, the *simple k-mean* algorithm has to be discarded. The same happens to all the other partitioning methods known, such as *PAM*, *CLARA* ecc...
We are therefore forced to use some of the hierarchical methods in order to try clustering the dataset in two classes. Of all the possible types of link, using the *euclidean distance* the one that gives the best result is the **average** linkage. The performances of the algorithm are summarized in the image below.

```
Clustered Instances

0       70 ( 56%)
1       54 ( 44%)


Class attribute: class
Classes to Clusters:

  0  1  <-- assigned to cluster
 31 31 | 0
 39 23 | 1

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

Incorrectly clustered instances :       54.0     43.5484 %
```

As we can see, the algorithm performs poorly, and is just slightly better than randomly guessing the class of the observation (only $\approx 56.45\%$ are correctly classified). In the next figure we can actually understand the reason behind the problem: there is almost only one variable, `attribute#3` (variable *is_smiling*), which determines the belonging of each observation to a certain class. This is a behaviour we may have expected, since with categorical variables the division in just two clusters may collide with the division given by a present variable.



Increasing the numbers of cluster may seem counterproductive, since the number of classes is just two, but this strategy allows us to unveal some characteristics of the data and of the algorithms applied. Of all the possible linkage used, none manages to improve the performances, except for one: the *single linkage*. In fact with this criteria one of the cluster will only be composed by a single observation (classified as outlier), while the other two will be balanced and manage to correctly classify $\approx 61.29\%$ of the total observations (see image below). In this way the single linkage hierarchical clustering algorithm becomes the top performing method of the ones tried. The results however are still of low accuracy, and we will need to run a more detailed association analysis in order to discover the underlying structure of the data.

```
Clustered Instances

0       67 ( 54%)
1       56 ( 45%)
2        1 (  1%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
 41 21  0 | 0
 26 35  1 | 1

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

Incorrectly clustered instances :       48.0     38.7097 %
```

Association analysis

Since the clustering methods failed (as expected) in detecting the real structure of the data, we now proceed fitting the *apriori* algorithm to the dataset. The instructions suggest us to use a maximum number of rules equal to 19 and setting a minimum support of 0.05; the remaining options were left on their default values (confidence 0.9).
After the computations are made, `Weka` selects for use the set of 19 best rules to describe the dataset (see image below). The set, in the image below, is of course too big for our purposes; however from the rules displayed we can find a smaller subset that does not contain redundance and perfectly describes the data available.

```
Generated sets of large itemsets:

Size of set of large itemsets L(1): 19

Size of set of large itemsets L(2): 151

Size of set of large itemsets L(3): 378

Size of set of large itemsets L(4): 125

Size of set of large itemsets L(5): 6

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)
```

From the rules above, we can select the following first four, **non redundant**, rules:

- *attribute#5=1 29 ==> class=1 29 conf:(1)* (1^{st} rule, color of the jacket is red)
- *attribute#1=3 attribute#2=3 17 ==> class=1 17 conf:(1)* (2^{nd} rule, same shape of body and head, case 3)
- *attribute#1=2 attribute#2=2 15 ==> class=1 15 conf:(1)* (5^{th} rule, same shape of body and head, case 2)
- *attribute#1=1 attribute#2=1 9 ==> class=1 9 conf:(1)* (14^{th} rule, same shape of body and head, case 1)

As we can see, the four rules selected here, all with confidence 1, correspond to the actual logic rule that define our class variable. We can therefore conclude that the a-priori algorithm was capable to detect the real structure underlying the data, ignoring the noise variables present in the data.

Bibliography

Wnek, J., Sarma, J., Wahab, A., and Michalski, R. S. (1990). Comparing learning paradigms via diagrammatic visualization: A case study in concept learning using symbolic, neural net and genetic algorithm methods. In *Proceedings of the 5th International Symposium on Methodologies for Intelligent Systems – ISMIS'90*, pages 428–437, Knoxville, TN.