Lab 3

Jesper Lindberg & Andreas Stasinakis

4/3/2019

Contents

Association Analysis -2 **2**

1.1 First, cluster the data **2**

1.2 Use the Clusters to class evaluation model **2**

1.3 Use association analysis **3**

1.4 Try to find as few rules predicting class 1 as possible **3**

Association Analysis -2

The goal with this lab is the following: *Appreciate the importance of the distance metric used within the clustering algorithm, e.g. the clusters found by the algorithm may not correspond to the clustering that is most natural for us human beings.*

In this lab we will use the monk dataset, it can be fund here: https://www.ida.liu.se/~732A75/lab/monk1.arff

1.1 First, cluster the data

First, cluster the data with different algorithms and number of clusters.

First, we run SimpleKMeans with 2 numbers of clusters and seed is set to 10. This gives us a squared error of 358 (very high). We then try to use the MakeDensityBasedClusters, we use this first with the default valu of minStDev set to $1.0E-6$. Which results in the exakt same sum of squared errors (SSE).

1.2 Use the Clusters to class evaluation model

Use the Clusters to class evaluation model to see whether the clustering algorithm is able to discover the class division existing in the data. Hopefully, you won't be able to discover it. In this exercise, we want you to answer to the following question: Why can the clustering algorithms not find a clustering that matches the class division in the database?

This is first for the SimpleKMeans and then for the MakeDensityBaedClusters. In the first case, we get an SSE of 305 and in the second case we get an SSE of 305. Both of them is an increase. After playing around with different classes and values, we were not able to find the class division.

1.3 Use association analysis

Use association analysis to find a set of rules that are able to accurately predict the class label from the rest of the attributes. We recommend you to use a 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.

Using the vaues mentionedin the task, we get a "best rules" in Weka. The rules is as follows:

1. *attribute#5=1 29 ==> class=1 29 lift:(2) lev:(0.12) [14] conv:(14.5)*
2. *attribute#1=3 attribute#2=3 17 ==> class=1 17 lift:(2) lev:(0.07) [8] conv:(8.5)*
3. *attribute#3=1 attribute#5=1 17 ==> class=1 17 lift:(2) lev:(0.07) [8] conv:(8.5)*
4. *attribute#5=1 attribute#6=1 16 ==> class=1 16 lift:(2) lev:(0.06) [8] conv:(8)*
5. *attribute#1=2 attribute#2=2 15 ==> class=1 15 lift:(2) lev:(0.06) [7] conv:(7.5)*
6. *attribute#1=3 attribute#5=1 13 ==> class=1 13 lift:(2) lev:(0.05) [6] conv:(6.5)*
7. *attribute#5=1 attribute#6=2 13 ==> class=1 13 lift:(2) lev:(0.05) [6] conv:(6.5)*
8. *attribute#2=3 attribute#5=1 12 ==> class=1 12 lift:(2) lev:(0.05) [6] conv:(6)*
9. *attribute#3=2 attribute#5=1 12 ==> class=1 12 lift:(2) lev:(0.05) [6] conv:(6)*
10. *attribute#1=3 attribute#2=3 attribute#6=2 12 ==> class=1 12 lift:(2) lev:(0.05) [6] conv:(6)*
11. *attribute#4=1 attribute#5=1 11 ==> class=1 11 lift:(2) lev:(0.04) [5] conv:(5.5)*
12. *attribute#1=2 attribute#5=1 10 ==> class=1 10 lift:(2) lev:(0.04) [5] conv:(5)*
13. *attribute#2=2 attribute#5=1 10 ==> class=1 10 lift:(2) lev:(0.04) [5] conv:(5)*
14. *attribute#1=1 attribute#2=1 9 ==> class=1 9 lift:(2) lev:(0.04) [4] conv:(4.5)*
15. *attribute#4=2 attribute#5=1 9 ==> class=1 9 lift:(2) lev:(0.04) [4] conv:(4.5)*
16. *attribute#4=3 attribute#5=1 9 ==> class=1 9 lift:(2) lev:(0.04) [4] conv:(4.5)*
17. *attribute#1=2 attribute#2=2 attribute#3=1 9 ==> class=1 9 lift:(2) lev:(0.04) [4] conv:(4.5)*
18. *attribute#1=3 attribute#2=3 attribute#3=1 9 ==> class=1 9 lift:(2) lev:(0.04) [4] conv:(4.5)*
19. *attribute#3=1 attribute#5=1 attribute#6=1 9 ==> class=1 9 lift:(2) lev:(0.04) [4] conv:(4.5)*

There is too many rules to really say anything about it. However, you can spot some redundent rules.

1.4 Try to find as few rules predicting class 1 as possible

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. Now, try to answer the question above. Finally, would you say that the clustering algorithms fail or perform poorly for the monk1 dataset? Why or why not?

We set the number o frules to 4 and without chaning anything else, the "best rules found" this time is:

1. *attribute#1=1 attribute#5=1 6 ==> class=1 6 lift:(2) lev:(0.02) [3] conv:(3)*
2. *attribute#1=1 attribute#2=2 attribute#5=3 6 ==> class=0 6 lift:(2) lev:(0.02) [3] conv:(3)*
3. *attribute#1=1 attribute#2=3 attribute#5=2 6 ==> class=0 6 lift:(2) lev:(0.02) [3] conv:(3)*

4. attribute#2=1 attribute#5=4 class=0 6 ==> attribute#1=2 6 lift:(2.95) lev:(0.03) [3] conv:(3.97)

Now, with only 4 rules it's easier to analyze. The first rule tells us that if attribute number 1 is set to 1 and attribute number 5 is set to 1, that instance is classified as class 1, 6 times. We can then see the confidence of the rule is 1, which is very high. These 4 rules describe the class 1 perfectly. Also no rules are redundent.

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

We belive that this is that there is too much overlaping in the clusters of the data points.