732A75 Data Mining Lab-3

Sridhar Adhikarla(sriad858) and Lakshidaa Saigiridharan(laksa656)

6 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 %
```

```
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 %
```

We just have the option of two types of distance functions in K-Means algorithm, Euclidean and Manhatten distance. But these did not seem to work well with this monk1 dataset. Using the default parameters of K-Means for both the measurements, it turned out that 47% of the instances were incorrectly clustered, and using Density Based clustering did not make much improvement as well. 45% of the data was clustered wrong.

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

The clustering algoritm is not working properly for this dataset because, there is no proper seperation between the classes. When plotting the data with the class labels we found that there is an overlap in the clusters, and using K-Means directly on this data will not be useful. Some kind of perprocessing is required that would create a seperation between the classes, and make it easy for the clustering algorithm to work. For a clustering algorithm to work the instances of a class should lie close to each other, and the different classes centroids are well seperated.

A possible solution is to use a kernal function or LDA algorithm to create an artificial seperation between the classes. These algorithms project the data on to a plane that maximizes the seperation between the classes. Using this would make it easy for the clustering algorithm to work.

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

The clustering algorithm fails because the distance functions (Euclidean and Manhatten) is not able to capture all the constraints in the data. Preprocessing the data to create an artificial seperation will help these basic distance functions to work properly. If not, we can create a custom distance functions that captures these complex constraints or we can add new features(made from the existing ones), but that would require some domain knowledge.

Association Analysis

Table 1: Association rules

Parameters	Cluster	Occurences	Confidience
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

The only problem we were facing until now is that the points from the same class should be close to each other. Association analysis does not require that, which makes it easier to find rules and correctly classify points. Association analysis is a means to discover relationships in large data sets. Hidden data relationships will be expressed as a collection of association rules. In the case of this dataset, we end up with a certain set of rules for cluster 1. These general rules have 100% confidence and these rules cover both the datasets as the points that are not covered by cluster 1 belong to cluster 0. This is the reason other rules that had lower confidence level were removed. This algorithm has significantly better results than the clustering algorithms for such datasets.