CIS6930 Fall 2017: Introduction to Data Mining
Project II: Clustering

By Jay Shah (1461-3930)

11/7/2017

1 Clustering Techniques

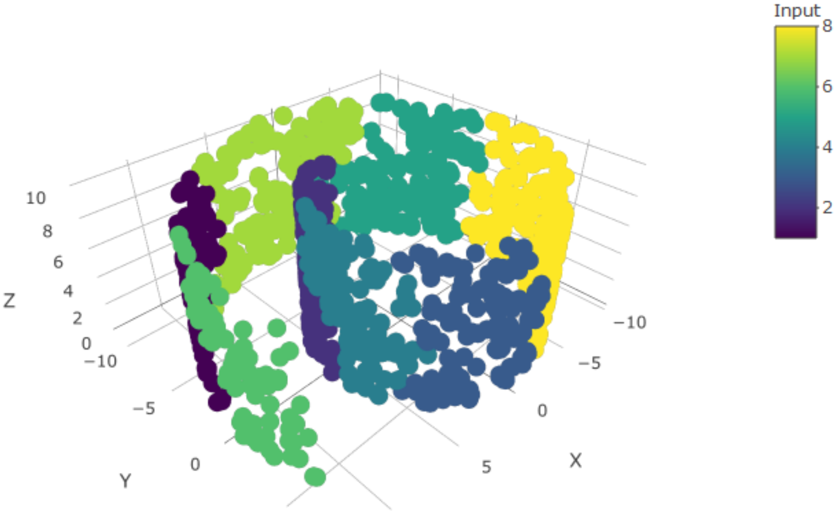
[1] Hierarchical Clustering:

- This type of clustering seeks to build a hierarchy of clusters. These clusters are developed using two methods:

  1) Agglomerative: - starts with each point as cluster and merge them in one cluster like bottom-up fashion.

  2) Divisive: - In this method, all the data points start as a single cluster. A top down approach is used where the single cluster is split recursively.

- 3D plot of clusters is given below.



- Function used: hclust
- Parameters:

  1) Input Data

  2) Method Name (Ward.D, Ward.D2, single, complete, average)

- Here, Ward.D2 method is chosen as it gives the best accuracy among other methods.
- Output: Dendrogram/Tree whose leaf are data point and other nodes are cluster

- Function used to get 8 clusters: Cuttree
- Parameters:

  1)Output of the first function

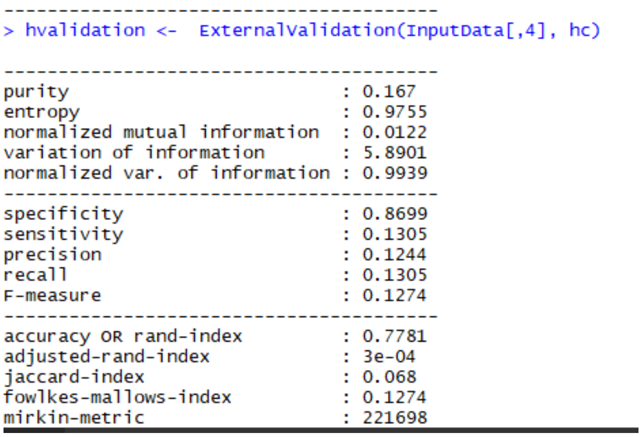
  2)N where n is number of clusters we want

- Accuracy using external validation function

  1)Accuracy or Rand-Index: 0.7781

  2)Precision: 0.1244

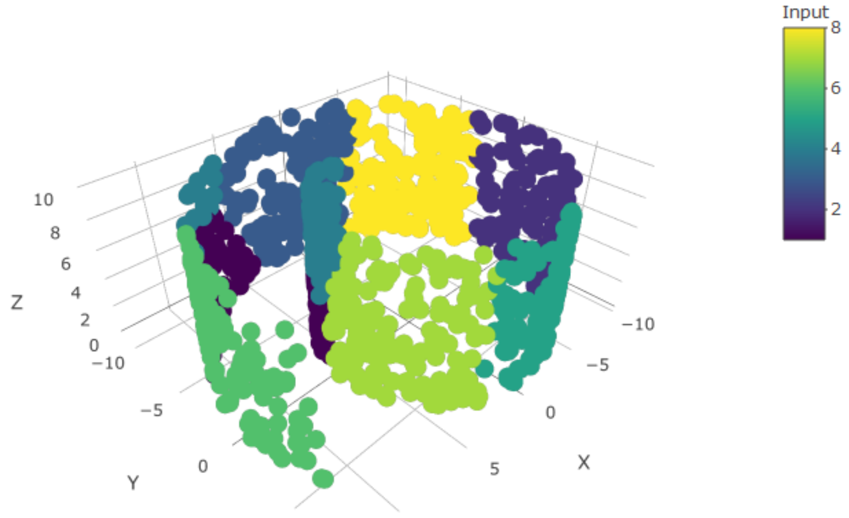
  3)Recall: 0.1305



```
----------------------------------------
> hvalidation <-  ExternalValidation(InputData[,4], hc)

----------------------------------------
purity                              : 0.167
entropy                             : 0.9755
normalized mutual information       : 0.0122
variation of information            : 5.8901
normalized var. of information      : 0.9939
----------------------------------------
specificity                         : 0.8699
sensitivity                         : 0.1305
precision                           : 0.1244
recall                              : 0.1305
F-measure                           : 0.1274
----------------------------------------
accuracy OR rand-index              : 0.7781
adjusted-rand-index                 : 3e-04
jaccard-index                       : 0.068
fowlkes-mallows-index               : 0.1274
mirkin-metric                       : 221698
```

[2] K-means Clustering:

- K-means Clustering is a clustering method which splits the data into ‘k’ clusters. The start-points for these k clusters can be either randomly selected using the ‘Forgy method’ or the ‘Random Partition’ method, or provided by the user.
- 3D plot of clusters is given below.



- Function used: K-means
- Parameters:

  1)Input Data

  2)K where K is number of clusters (8 in our case)

  3)n-start which is random sets chosen as center initially

- Output: K clusters which contains points that are
- In our method n-start = 2 is chosen as it gives the best accuracy among the n-start value ranges from 1 to 25
- As K-means choose centroids randomly, we get different accuracies each time we run the algorithm.

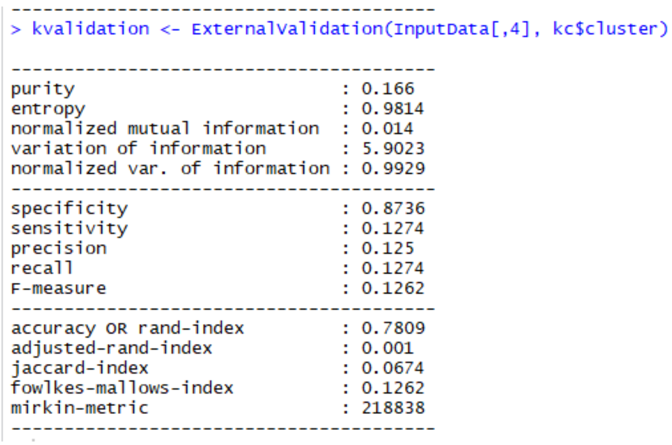
Accuracy	0.777	0.776	0.776	0.776	0.775	0.777	0.778	0.779	0.778	0.780

- Accuracy using external validation is

  1)Average Accuracy / Rand-Index: 0.776

  2)Precision: 0.125

  3)Recall: 0.1274

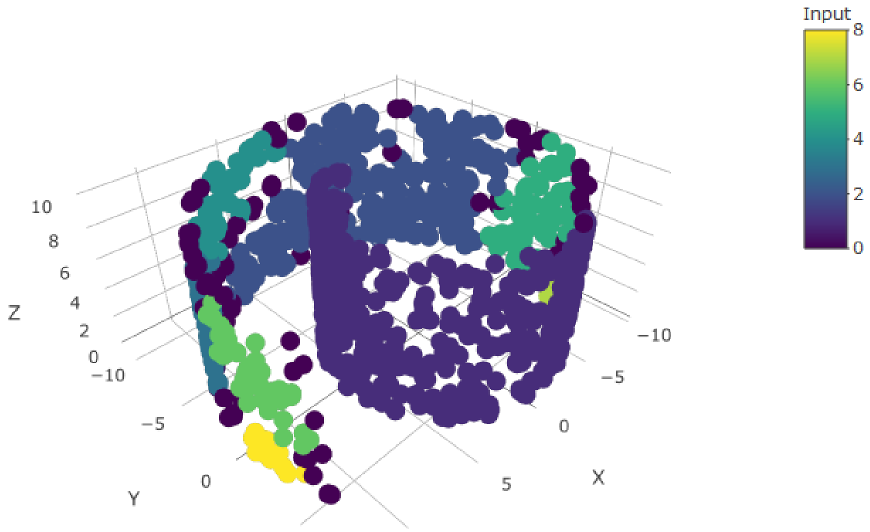


```
----------------------------------------
> kvalidation <- ExternalValidation(InputData[,4], kc$cluster)

----------------------------------------
purity                             : 0.166
entropy                            : 0.9814
normalized mutual information      : 0.014
variation of information           : 5.9023
normalized var. of information     : 0.9929
----------------------------------------
specificity                        : 0.8736
sensitivity                        : 0.1274
precision                          : 0.125
recall                             : 0.1274
F-measure                          : 0.1262
----------------------------------------
accuracy OR rand-index             : 0.7809
adjusted-rand-index                : 0.001
jaccard-index                      : 0.0674
fowlkes-mallows-index              : 0.1262
mirkin-metric                      : 218838
----------------------------------------
```

[3] Density Based Clustering:

- Density-based spatial clustering of applications with noise (DBSCAN) is a density based data clustering algorithm. This algorithm groups together points which are packed together closely i.e. points which have nearby neighbours while marking points which are in low-density regions as outliers or noise points.

- 3D plot of clusters is given below.



- Function used: DBScan

- Parameters:

    1)Input Data

    2)Epsilon: size of the epsilon neighborhood

    3)Min points: Number of minimum points in eps region

- Output: 8 clusters which contains number of records Data

- In our method we must tune the value of epsilon and minimum points to get the 8-total number of clusters.

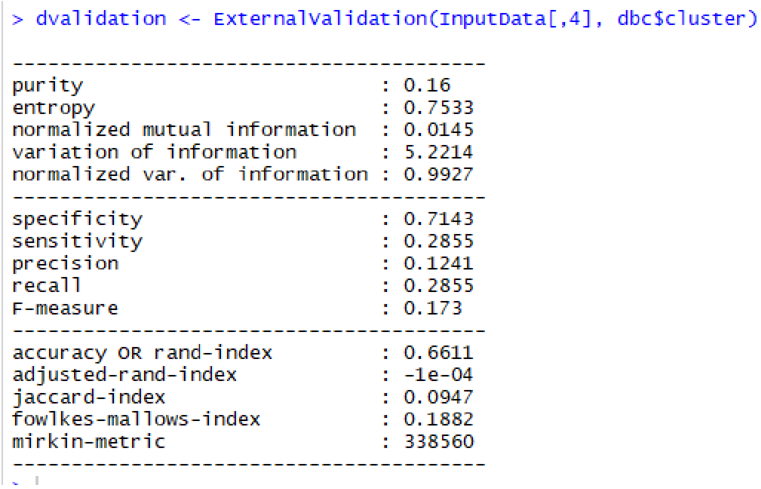
- Epsilon value can be identified using knnplot. Minpts values can be tuned in using trial and error method only.

- Accuracy using external validation is

    1)Accuracy / Rand-Index: 0.6611

    2)Precision: 0.1241

    3)Recall: 0.2855



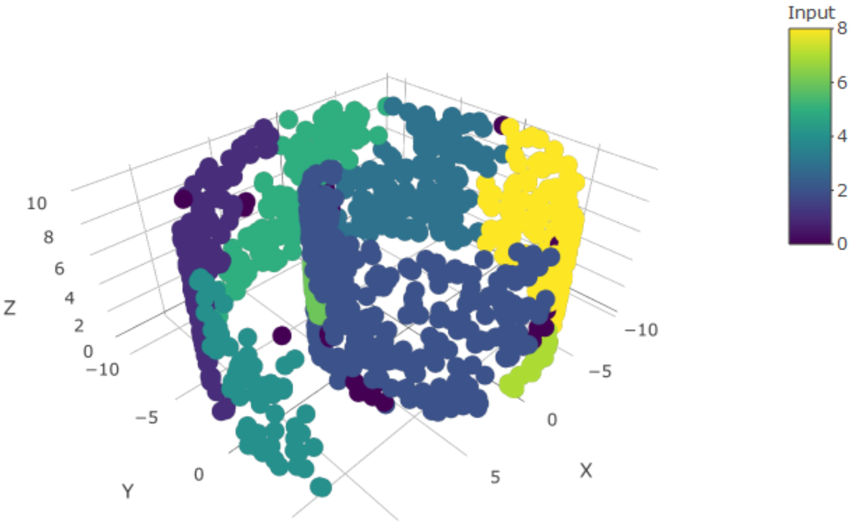
```
> dvalidation <- ExternalValidation(InputData[,4], dbc$cluster)

----------------------------------------
purity                               : 0.16
entropy                              : 0.7533
normalized mutual information        : 0.0145
variation of information             : 5.2214
normalized var. of information       : 0.9927
----------------------------------------
specificity                          : 0.7143
sensitivity                          : 0.2855
precision                            : 0.1241
recall                               : 0.2855
F-measure                            : 0.173
----------------------------------------
accuracy OR rand-index               : 0.6611
adjusted-rand-index                  : -1e-04
jaccard-index                        : 0.0947
fowlkes-mallows-index                : 0.1882
mirkin-metric                        : 338560
----------------------------------------
```

[4] Graph Based Clustering:

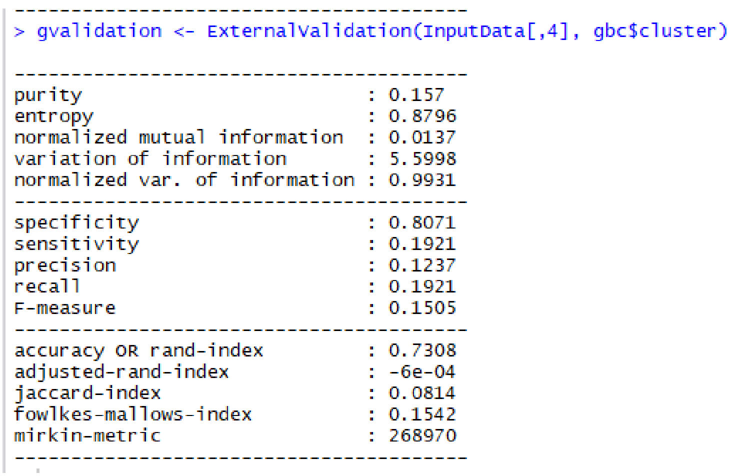
- Graph-Based clustering uses a proximity graph to divide the data points into different clusters. A proximity matrix is created consisting all the data points. Each of these points are considered to be a node in a graph.

- Each edge between the nodes has a weight which is the value of the proximity between the nodes from the matrix. Initially the proximity graph is fully connected with MIN (single link) and MAX (complete-link) that can be viewed as starting with this graph. In the simplest case, clusters are connected components in the graph. Also, the amount of data to be calculated is drastically reduced in this case due to sparsification.

- 3D plot of clusters is given below.



- Function used: sNNclust
- Parameters:
  1) Input Data
  2) K – Neighborhood size for nearest neighbor sparsification to create the shared graph
  3) Eps value – Two objects will only reachable to each other if they share at least eps nearest neighbors
  4) Minpts - Minimum number of points that share at least epsilon nearest neighbors
- Output: 8 clusters which contains number of records in the Data
- Epsilon value can be identified using KNNplot. In this method we must tune the value of K, minimum points to get the 8-total number of clusters.
- Values can be tuned using trial and error method only.
- Accuracy using external validation is
  1) Accuracy / Rand-Index: 0.7308
  2) Precision: 0.1237
  3) Recall: 0.1921



```
----------------------------------------
> gvalidation <- ExternalValidation(InputData[,4], gbc$cluster)

----------------------------------------
purity                              : 0.157
entropy                             : 0.8796
normalized mutual information       : 0.0137
variation of information            : 5.5998
normalized var. of information      : 0.9931
----------------------------------------
specificity                         : 0.8071
sensitivity                         : 0.1921
precision                           : 0.1237
recall                              : 0.1921
F-measure                           : 0.1505
----------------------------------------
accuracy OR rand-index              : 0.7308
adjusted-rand-index                 : -6e-04
jaccard-index                       : 0.0814
fowlkes-mallows-index               : 0.1542
mirkin-metric                       : 268970
----------------------------------------
```

2 Method used for Second Dataset

There are four methods we can think of applying on the dataset.

1. Hierarchical Clustering: - We cannot use this method as our dataset contains 1 million records and this algorithm requires to build a distance matrix which requires a lot of space and time.

2. Graph based Clustering: - We cannot use this method as it also requires distance matrix which will lead to use of large space and time complexity.

3. Density based Clustering: **-** Density based clustering gives very low accuracy for previous dataset and it also requires value of k and min points to set manually. So I have not chose density based clustering method for this dataset.

4. K-means Clustering: **-** K-means method can be used but we have to decide the number of clusters in this method.

    - Let's say we choose n-start value randomly as 23.
    - Now we must decide the value of K.
    - To decide the value of K, I tried to run algorithm for k=1 to 150 and stored the value of inter cluster distance and intra cluster distance. Now we can say that value of k is efficient if we get high inter cluster distance and low intra cluster distance.
    - I took the difference of inter cluster difference and intra cluster difference for each K value and chose one with the maximum difference value.
    - K value for which I got the maximum difference value is 121.
    - Now for different n-start values I got different K value but all were in the range of 120-140.
    - **K = 121 clusters are best for this dataset.**
    - **I think K-means clustering algorithm will give the best result for this dataset.**
    - **Applying Root mean square deviation method, RMSD value = 7.39**
    - **Here Similarity matrix cannot be built as it requires the 1 million*121 size of space which is too big for this project right now.**

3 Conclusion

1. Efficiency, Accuracy of an algorithm depends on the various factors like size of the dataset.

2. On the Dataset1, K-means algorithm gives good accuracy than other algorithms and it remains close to the result of hierarchical algorithm. As the size of dataset increases graph base may give higher accuracy but it may take a lot of space.

4 References

1. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/hclust.html

2. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/kmeans.html

3. https://cran.r-project.org/web/packages/dbscan/dbscan.pdf

4. https://www.rdocumentation.org/packages/dbscan/versions/1.1-1/topics/sNNclust