Gabi Rivera

ADS501-01

03Dec2022

Introduction to Data Mining: Exercises 7.7

3. Many partitional clustering algorithms that automatically determine the number of clusters claim that this is an advantage. List two situations in which this is not the case.

Answer:

Two situations where partitional clustering's automatic numbered cluster is not an advantage is when the clusters need to be specified to a value because the dataset will have to be reduced as needed later on in the process, and when the data has a hierarchical structure.

5. Identify the clusters in Figure 7.36 using the center-, contiguity-, and density-based definitions. Also indicate the number of clusters for each case and give a brief indication of your reasoning. Note that darkness or the number of dots indicates density. If it helps, assume center-based means Kmeans, contiguity-based means single link, and density-based means DBSCAN.

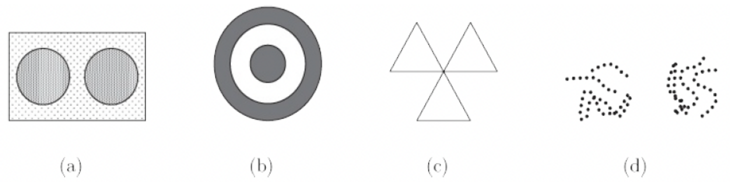

Figure 7.36.
Clusters for **Exercise 5**.

Answer:

a.) Center-based: The number of the cluster is 2 since there are two dark circles inside the rectangle box.

Contiguity-based: The number of clusters is 1 because the noise will merge the two circles.

Density-based: The number of clusters is 2 as the low density will be eliminated.

b.) Center-based: The number of clusters is 1. The two circles will be counted as having one middle center.

Contiguity-based: The number of clusters is 2 because there are two circular dark clusters.

Density-based: The number of clusters is 2 because there are two circular dark clusters.

c.) Center-based: The number of clusters is 3 because there are three triangles.

Contiguity-based: The number of clusters is 1 because all will be merged into one.

Density-based: The number of clusters is 3 because there are three triangles.

d.) Center-based: The number of clusters is 2 because there are two separate groups of centers.

Contiguity-based: The number of clusters is 5 because there are 5 strings of lines in each group.

Density-based: The number of clusters is 2 because there are two groups that are highly dense.

18. Suppose we find K clusters using Ward's method, bisecting K-means, and ordinary K-means. Which of these solutions represents a local or global minimum? Explain.

Answer: Ordinary K-means can represent a local minimum because of its refinement step which both Ward's method and Bisecting K-means lack but none of these solutions are sure to produce a global minimum.

20. Consider the following four faces shown in Figure 7.39 . Again, darkness or number of dots represents density. Lines are used only to distinguish regions and do not represent points.

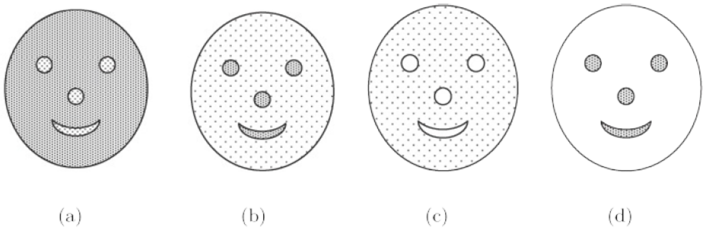

Figure 7.39.
Figure for **Exercise 20**.

a. For each figure, could you use single link to find the patterns represented by the nose, eyes, and mouth? Explain.

   Answer: b and d. For b, the dark clusters are closer to each other compared to the noise. For d, there are only dark clusters and no noise.

b. For each figure, could you use K-means to find the patterns represented by the nose, eyes, and mouth? Explain.

   Answer: b and d. For b, the dark clusters will be identified together with the noise. For d, the dark clusters will be readily identified.

c. What limitation does clustering have in detecting all the patterns formed by the points in Figure 7.39(c) ?

   Answer: For c, there are only noise. There are empty spaces but these will not be detected.

22. You are given two sets of 100 points that fall within the unit square. One set of points is arranged so that the points are uniformly spaced. The other set of points is generated from a uniform distribution over the unit square.

a. Is there a difference between the two sets of points?

Answer: There is a difference between the two sets of points as one will have an even uniform distribution of points compared to the other which will have variable density with uniform spacing.

b. If so, which set of points will typically have a smaller SSE for clusters?

Answer: The one with variable densities will have a smaller SSE for clusters.

d. What will be the behavior of DBSCAN on the uniform data set? The random data set?

Answer: DBSCAN will merge or remove the uniform data set. With the random dataset, DBSCAN will identify clusters of higher densities and remove the lower densities.

30. Clusters of documents can be summarized by finding the top terms (words) for the documents in the cluster, e.g., by taking the most frequent k terms, where k is a constant, say 10, or by taking all terms that occur more frequently than a specified threshold. Suppose that K-means is used to find clusters of both documents and words for a document data set.

a. How might a set of term clusters defined by the top terms in a document cluster differ from the word clusters found by clustering the terms with Kmeans?

Answer: Top term clusters have the potential to overlap and misrepresent other frequent terms that are not captured in the top selection. In K-means, represent all most frequent terms without overlapping.

b. How could term clustering be used to define clusters of documents?

Answer: Term clustering can be used to define clusters of documents by taking the top documents having the most frequent terms.

Module6

Gabi Rivera

2022-12-04

Data Science Using Python and R: Chapter 10

For the following exercises, work with the white_wine_training and white_wine_test data sets. Use either Python or R to solve each problem.

11. Input and standardize both the training and test data sets.

```
wine_train = read.csv('white_wine_training', sep = ',')
wine_test = read.csv('white_wine_test', sep = ',')

#Subset predictors:
X = subset(wine_train, select = c("alcohol", "sugar"))
X_test = subset(wine_test, select = c("alcohol", "sugar"))

#Standardize predictors:
Xs = as.data.frame(scale(X))
colnames(Xs) = c("alcohol_z", "sugar_z")

Xs_test = as.data.frame(scale(X_test))
colnames(Xs_test) = c("alcohol_z", "sugar_z")
```

12. Run k-means clustering on the training data set, using two clusters.

```
kmeans01 = kmeans(Xs, centers = 2)

#Save cluster membership of each record as it's own variable
cluster = as.factor(kmeans01$cluster)

#Description of each cluster
Cluster1 = Xs[ which(cluster == 1), ]
Cluster2 = Xs[ which(cluster == 2), ]
```

13. Give the mean of each variable within each cluster and use the means to identify a "Dry wines" and a "Sweet wines" cluster.

```
#Summary
summary(Cluster1)
```

```
##    alcohol_z          sugar_z
##  Min.   :-1.8265   Min.   :-0.9085
##  1st Qu.:-1.1586   1st Qu.: 0.3541
##  Median :-0.9081   Median : 0.8676
##  Mean   :-0.7552   Mean   : 0.9608  
##  3rd Qu.:-0.4072   3rd Qu.: 1.4882  
##  Max.   : 2.0138   Max.   : 5.5113
```

```
summary(Cluster2)
```

```
##    alcohol_z          sugar_z       
##  Min.   :-1.5760   Min.   :-1.1225  
##  1st Qu.:-0.1568   1st Qu.:-0.9513  
##  Median : 0.4276   Median :-0.8443  
##  Mean   : 0.4902   Mean   :-0.6236  
##  3rd Qu.: 1.1790   3rd Qu.:-0.3521  
##  Max.   : 2.8904   Max.   : 1.4775
```

Cluster 1 = positive value for Sweet wine

- Dry wine, means: -0.76
- Sweet wine, means: 0.96

Cluster 2 = positive value for Dry wine

- Dry wine, means: 0.49
- Sweet wine, means: -0.62

14. Validate the clustering results by running k-means clustering on the test data set, using two clusters, and identifying a "Dry wines" and a "Sweet wines" cluster.

```
#Validate test dataset
kmeans01_test = kmeans(Xs_test, centers = 2)
cluster_test = as.factor(kmeans01_test$cluster)
Cluster1_test = Xs[ which(cluster_test == 1), ]
Cluster2_test = Xs[ which(cluster_test == 2), ]

summary(Cluster1_test);
```

```
##    alcohol_z          sugar_z       
##  Min.   :-1.7430   Min.   :-1.1225  
##  1st Qu.:-0.2403   1st Qu.:-0.9245  
##  Median : 0.3441   Median :-0.5019  
##  Mean   : 0.3697   Mean   :-0.1869  
##  3rd Qu.: 1.0120   3rd Qu.: 0.3327  
##  Max.   : 1.9303   Max.   : 3.5853
```

```
summary(Cluster2_test)
```

```
##    alcohol_z          sugar_z       
##  Min.   :-1.8265   Min.   :-1.1011  
##  1st Qu.:-1.2421   1st Qu.:-0.7801  
##  Median :-0.9081   Median : 0.3327  
##  Mean   :-0.8267   Mean   : 0.3606  
##  3rd Qu.:-0.5742   3rd Qu.: 1.2100  
##  Max.   : 1.4294   Max.   : 5.5113
```

Cluster 1 = positive value for Sweet wine

- Dry wine, means: -0.83
- Sweet wine, means: 0.36

Cluster 2 = positive value for Dry wine

- Dry wine, means: 0.37
- Sweet wine, means: -0.19

Module 6 Assignment

Data Science Using Python and R: Chapter 10

For the following exercises, work with the white_wine_training and white_wine_test data sets. Use either Python or R to solve each problem.

In [1]:

```
import pandas as pd
from scipy import stats
from sklearn.cluster import KMeans
```

11 Input and standardize both the training and test data sets.

In [3]:

```
#Import both datasets:
wine_train = pd.read_csv("white_wine_training", sep = ',')
wine_test = pd.read_csv("white_wine_test", sep = ',')


#Isolate predictors:
X = wine_train[['alcohol', 'sugar']]
X_test = wine_test[['alcohol', 'sugar']]


#Standardize predictors using z-scores:
Xz = pd.DataFrame(stats.zscore(X), columns=['alcohol', 'sugar'])
Xz_test = pd.DataFrame(stats.zscore(X_test), columns=['alcohol', 'sugar'])
```

12 Run k-means clustering on the training data set, using two clusters.

In [4]:

```
#Kmeans model:
kmeans01 = KMeans(n_clusters = 2).fit(Xz)

#Save clustering membership to it's own variable:
cluster = kmeans01.labels_

#Separate records into two groups based on cluster membership:
Cluster1 = Xz.loc[cluster == 0]
Cluster2 = Xz.loc[cluster == 1]
```

13 Give the mean of each variable within each cluster and use the means to identify a “Dry wines” and a “Sweet wines” cluster.

In [6]:

```
#Summary1:
Cluster1.describe()
```

Out[6]:

	alcohol	sugar
count	712.000000	712.000000
mean	-0.755428	0.961034
std	0.580989	0.818726
min	-1.826971	-0.908740
25%	-1.158911	0.354160
50%	-0.908388	0.867883
75%	-0.407343	1.488630

max	2.014374	5.512788

In [7]:

```
#Summary2:
Cluster2.describe()
```

Out[7]:

	alcohol	sugar
count	1097.000000	1097.000000
mean	0.490305	-0.623752
std	0.905663	0.475694
min	-1.576448	-1.122791
25%	-0.156821	-0.951551
50%	0.427732	-0.844525
75%	1.179299	-0.352208
max	2.891203	1.477928

14 Validate the clustering results by running k-means clustering on the test data set, using two clusters, and identifying a “Dry wines” and a “Sweet wines” cluster.

In [9]:

```
kmeans_test = KMeans(n_clusters = 2).fit(Xz_test)
cluster_test = kmeans_test.labels_
Cluster1_test = Xz_test.loc[cluster_test == 0]
Cluster2_test = Xz_test.loc[cluster_test == 1]
```

In [11]:

```
#Summary1:
Cluster1_test.describe()
```

Out[11]:

	alcohol	sugar
count	638.000000	638.000000
mean	-0.802630	1.065341
std	0.561207	0.779670
min	-2.080483	-1.037949
25%	-1.190079	0.396441
50%	-0.947241	1.032518
75%	-0.542512	1.583612
max	1.562080	3.298700

In [12]:

```
#Summary2:
Cluster2_test.describe()
```

Out[12]:

	alcohol	sugar
count	1122.000000	1122.000000
mean	0.456397	-0.605782
std	0.903287	0.459740
min	-1.675754	-1.089453
25%	-0.218729	-0.945241
50%	0.368129	-0.821632

75%	1.157351	-0.285988
max	2.776268	1.423949