Module 6 | Python

November 30, 2021

1 Module 6 | Python

1.0.1 Ryan S. Dunn

1.1 Data Science Using Python and R: Chapter 10 - Page 149: Questions #11, 12, 13, & 14

[5]:
```
#import libraries for KMeans algorithm
import pandas as pd
from scipy import stats
from sklearn.cluster import KMeans
```

[4]:
```
#import the white wine training and test
wine_train = pd.read_csv("/Users/ryan_s_dunn/Documents/USD MS-ADS/Applied Data␣
↪Mining 502/Module 6/datasets/white_wine_training", header = 0)
wine_test = pd.read_csv("/Users/ryan_s_dunn/Documents/USD MS-ADS/Applied Data␣
↪Mining 502/Module 6/datasets/white_wine_test", header = 0)
```

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

[14]:
```
#standardize the training and test data sets (z-score)
wine_train_z = pd.DataFrame(stats.zscore(wine_train), columns =␣
↪['alcohol','quality','sugar'])
wine_test_z = pd.DataFrame(stats.zscore(wine_test), columns =␣
↪['alcohol','quality','sugar'])
```

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

[41]:
```
#run the k-means clustering algorithm on the training data
kmeans01 = KMeans(n_clusters = 2).fit(wine_train_z)
```

[10]:
```
#identify the cluster memebership
cluster = kmeans01.labels_
```

[11]:
```
#seperate the records into two groups based on cluster memebership
Cluster1 = wine_train_z.loc[cluster ==0]
Cluster2 = wine_train_z.loc[cluster ==1]
```

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

[12]:
```
#compute summary statistics of cluster 1
Cluster1.describe()
```

[12]:
```
          alcohol     quality       sugar
count  992.000000  992.000000  992.000000
mean    -0.689756   -0.553389    0.424439
std      0.560951    0.778523    1.068893
min     -1.826971   -3.252193   -1.122791
25%     -1.096280   -0.958094   -0.609069
50%     -0.824881   -0.958094    0.396970
75%     -0.323836    0.188956    1.210364
max      1.847359    2.483055    5.512788
```

[29]:
```
Cluster1.mean()
```

[29]:
```
alcohol   -0.689756
quality   -0.553389
sugar      0.424439
dtype: float64
```

Cluster1 is a sweet wine - notice the mean of the sugar attribute.

[13]:
```
#compute summary statistics of cluster 2
Cluster2.describe()
```

[13]:
```
          alcohol     quality       sugar
count  817.000000  817.000000  817.000000
mean     0.837501    0.671924   -0.515353
std      0.744389    0.810249    0.586883
min     -1.075403   -2.105143   -1.101386
25%      0.344224    0.188956   -0.940848
50%      0.761762    0.188956   -0.780310
75%      1.429822    1.336005   -0.223777
max      2.891203    3.630104    2.066568
```

[30]:
```
Cluster2.mean()
```

[30]:
```
alcohol    0.837501
quality    0.671924
sugar     -0.515353
dtype: float64
```

Cluster 2 is a dry wine - notice the mean of the sugar attribute

1.1.4 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.

[15]:
```
#run the k-means clustering algorithm on the test data
kmeans_test = KMeans(n_clusters = 2).fit(wine_test_z)
```

[16]:
```
#identify the cluster memebership
cluster_test = kmeans_test.labels_
```

[17]:
```
#seperate the records into two groups based on cluster memebership
cluster1_test = wine_test_z.loc[cluster_test==0]
cluster2_test = wine_test_z.loc[cluster_test==1]
```

[27]:
```
#compute summary statistics of test cluster 1
cluster1_test.describe() #dry wine - notice sugar mean
```

[27]:

	alcohol	quality	sugar
count	868.000000	868.000000	868.000000
mean	0.756414	0.590031	-0.532449
std	0.777203	0.831249	0.552722
min	-1.432916	-2.063322	-1.068851
25%	0.186001	0.139557	-0.937516
50%	0.671676	0.139557	-0.780428
75%	1.400189	1.240997	-0.244785
max	2.776268	3.443877	1.877186

[31]:
```
cluster1_test.mean()
```

[31]:
```
alcohol    0.756414
quality    0.590031
sugar     -0.532449
dtype: float64
```

cluster1_test is a dry wine - notice the mean of the sugar attribute.

[40]:
```
#compute summary statistics of test cluster 2
cluster2_test.describe() #sweet wine - notice sugar mean
```

[40]:

	alcohol	quality	sugar
count	892.000000	892.000000	892.000000
mean	-0.736062	-0.574156	0.518123
std	0.536420	0.796097	1.064472
min	-2.080483	-3.164762	-1.089453
25%	-1.109133	-0.961882	-0.285988
50%	-0.866295	-0.961882	0.414468
75%	-0.380620	0.139557	1.341542
max	1.643026	2.342437	3.298700

[32]:
```
cluster2_test.mean()
```

[32]:
```
alcohol   -0.736062
quality   -0.574156
sugar      0.518123
dtype: float64
```

cluster2_test is a sweet wine - notice the mean of the sugar attribute.

Module 6 | R

Ryan S Dunn

11/30/2021

Data Science Using Python and R: Chapter 10 - Page 149: Questions #11, 12, 13, & 14

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

```
#import the data into R
wine_train <- read.csv("/Users/ryan_s_dunn/Documents/USD MS-ADS/Applied Data Mining 502/Module 6/dataset
wine_test <- read.csv("/Users/ryan_s_dunn/Documents/USD MS-ADS/Applied Data Mining 502/Module 6/datasets
```

```
#subset and standardize the training and test data sets
train_subset <- subset(wine_train, select = c("alcohol","sugar"))
test_subset <- subset(wine_test, select = c("alcohol","sugar"))

#standardize training
train_z <- as.data.frame(scale(train_subset))
colnames(train_z) <- c("alcohol_z","sugar_z")

#standardize test
test_z <- as.data.frame(scale(test_subset))
colnames(test_z) <- c("alcohol_z", "sugar_z")
```

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

```
#run the k-means clustering algorithm on training data
kmeans01 <- kmeans(train_z, centers = 2)
```

```
#save the cluster memebership of each records as its own variable
cluster <- as.factor(kmeans01$cluster)
```

```
#seperate records into two groups for membership
cluster1 <- train_z[ which(cluster == 1), ]
cluster2 <- train_z[ 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(cluster1)
```

Summary statistics for Cluster 1 (training)

```
##    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
```

```
summary(cluster2)
```

Summary statistics for Cluster 2 (training)

```
##    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
```

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.

```
#run the k-means clustering algorithm on test data
kmeans01_test <- kmeans(train_z, centers =2)
```

```
#save the cluster memebership of each records as its own variable
cluster_test <- as.factor(kmeans01_test$cluster)
```

```
#seperate records into two groups for membership
cluster1_test <- test_z[ which(cluster == 1), ]
cluster2_test <- test_z[ which(cluster == 2), ]
```

```
summary(cluster1_test)
```

Summary statistics for Cluster 1 (test)

```
##    alcohol_z          sugar_z
##  Min.   :-1.5134   Min.   :-1.0686
##  1st Qu.:-0.2187   1st Qu.:-0.9038
##  Median : 0.5096   Median :-0.5743
##  Mean   : 0.4988   Mean   :-0.2454
##  3rd Qu.: 1.2380   3rd Qu.: 0.2084
##  Max.   : 2.7755   Max.   : 3.2978
##  NA's   :48        NA's   :48
```

```
summary(cluster2_test)
```

Summary statistics for Cluster 2 (test)

```
##    alcohol_z          sugar_z
##  Min.   :-2.0799   Min.   :-1.0891
##  1st Qu.:-1.1897   1st Qu.:-0.8420
##  Median :-0.8660   Median : 0.2908
##  Mean   :-0.7360   Mean   : 0.3620
##  3rd Qu.:-0.3805   3rd Qu.: 1.2794
```

```
##  Max.   : 2.2899   Max.   : 3.0506
##  NA's   :1         NA's   :1
```

Introduction to Data Mining:

Exercises 7.7 - Page 603

Questions # 3, 5, 18, 20, 22, 30

3. Many partitional clustering algorithms that automatically determine the number of clusters claim that this is an advantage. List 2 situations in which this is not the case.

(1) When a hierarchy exists, a partional algorithm will have difficulty determining a hierarchical structure.

(2) If you need to reduce the # of dimensions of the data, you must identify the # of clusters you need... this cannot be automatically determined.

5. Identify the clusters in section 7.3 using the center-, contiguity-, and density-based definitions. Also indicate the number of clusters for each case and give a brief explanation of your reasoning.

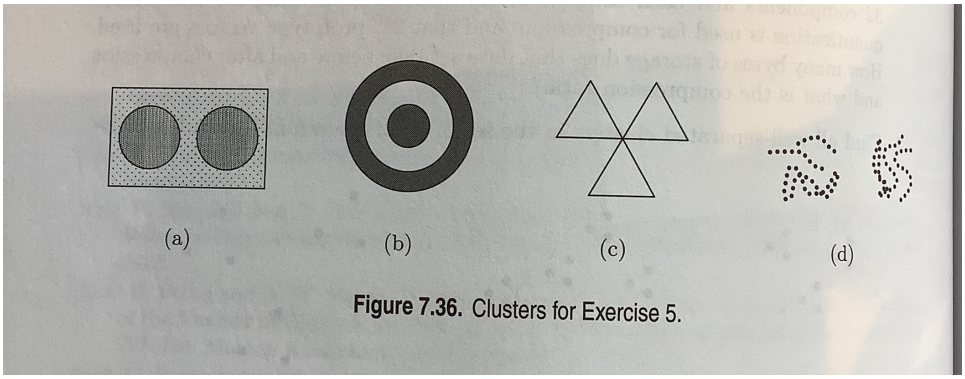

Figure 7.36. Clusters for Exercise 5.

(a) center based - 2 clusters, and the rectangle will be split in half

contiguity - 1 large cluster, the noise connects the clusters

density - 2 clusters, for both dense regions.

(b) center based - 1 cluster, as the center is the same

contiguity - 2 clusters, as contiguity is good for irregular & intertwined data.

density - 2 clusters of both rings

(c) center based - 3 clusters w/ center of triangle being the center

contiguity - 3 clusters, with the joined triangles

density - 3 clusters, with the triangles, since there is "open" space.

(d) center based - 2 clusters, with a center in both squiggle areas

contiguity - 2 clusters, as the squiggle connects

density - 2 clusters, as both squiggles have space inbetween them.

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

Bisecting K-Means uses K-Means "locally" to bisect individual clusters, and so the final set of clusters does not represent a clustering that is a local min with respect to SSE. Also, agglomerative clustering does not produce local min.
Ordinary K-Means however, will produce a local min,

20. Consider the following four faces shown in Figure 7.7.

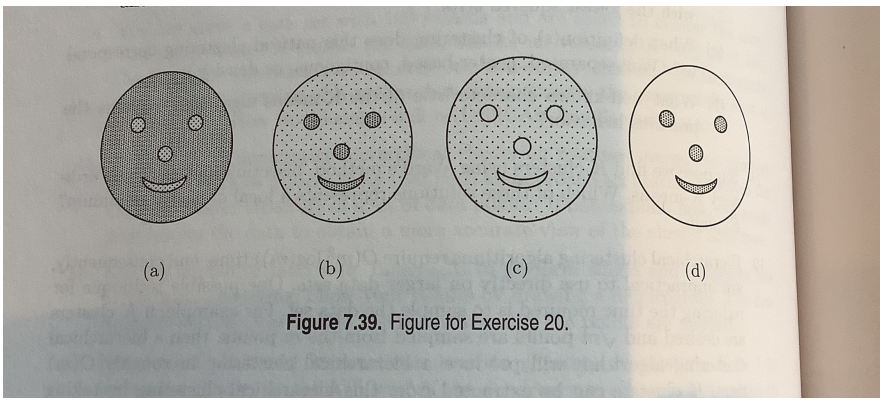

Figure 7.39. Figure for Exercise 20.

(a) for each Figure, could you use single link to find the patterns represented by the eyes, nose, and mouth. Explain.

(b) & (d) have points that are very close together & the other points outside of the regions could be accounted for as noise / don't exist.

(b) for each Figure, could you use the K-means to find the patterns represented by the nose, eyes & mouth. Explain.

Also (b) & (d), for the same reasons as above stated.

(c) What limitation does clustering have in detecting all the patterns formed by the points in Figure 7.7(c)?

The eyes, nose, & mouth are negative space, so the clusters would only find the space where data exists.

22. You are given 2 sets of 100 points that fall within the unit square. One set of points is arranged so 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 2 sets of points?

Yes, as the definition "uniformly spaced" does so, they will have the same space between the points, whereas the other data set will have a more condensed & expanded relationship.

(b) If so, which set of points will typically have a smaller SSE for K=10 clusters?

The randomly distributed set.

(c) What will be the behavior of DBSCAN on the uniform data set? The random data set?

The uniformed data set will be one cluster only, as there is no difference in density.

The random data set will have various clusters, depending on the density of points in the area.

30. Clusters of documents can be summarized by finding the top terms for the documents in the cluster. Suppose that K-means is used to find clusters of both documents and words from the 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 w/ K-Means?

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

The documents that have the highest term frequency would be grouped.