HUDM6122 Homework_07

Chenguang Pan

2023-04-12

0.1 Github Address

All my latest homework can be found on Github: https://github.com/cgpan/hudm6122_homeworks . Thanks for checking if interested.

0.2 Ex 7.1

In our class we mentioned the use of correlation-based distance and Eu- clidean distance as dissimilarity measures for hierarchical clustering. It turns out that these two measures are almost equivalent. . . ..

MY SOLUTION:

0.2.1 Solution 1: by pure algebra

First, I prove this proportionality assumption via algebra. Then, I validate it through data.

First, we standardized all the observations (note, not to standardize all the variables!!). Therefore, each observation will X_i follow $X_i \sim N(0,1)$. That is, for any observation, we always have $\sigma_{X_i} = 1$ and $\mu_{X_i} = 0$.

By definition, the squared the Euclidean distance can be writen as

$$d_{ij}^2 = \sum_{k=1}^{q} (x_{ik} - x_{jk})^2. \qquad ...(1)$$

Next, we do some algebra transformations on the correlation equation, like

$$r_{ij} = \frac{Cov(X_i, X_j)}{\sigma_{X_i}\sigma_{X_j}} = Cov(X_i, X_j) = E[(X_i - \mu_{X_i})(X_j - \mu_{X_j})] = E(X_i X_j). \qquad ...(2)$$

According to the definition of the expectation, we can have

$$E(X_i X_j) = \frac{\sum_{k=1}^{q} x_{ik} x_{jk}}{q}. \qquad ...(3)$$

Combine the (2) and (3), we can have

$$1 - r_{ij} = 1 - \frac{1}{q}\sum_{k=1}^{q} x_{ik} x_{jk}. \qquad ...(4)$$

Back to formula (1), after expanding it, we get

$$d_{ij}^2 = \sum_{k=1}^{q} x_{ik}^2 + \sum_{k=1}^{q} x_{jk}^2 - 2\sum_{k=1}^{q} x_{ik} x_{jk}. \qquad ...(5)$$

Since X_i is standardized, we can easily have

$$Var(X_i) = \frac{\sum_{k=1}^{q}(x_{ik} - \mu_{X_i})^2}{q} = 1. \qquad ...(6)$$

From the equation (6), we can prove that $\sum_{k=1}^{q} x_{ik}^2 = q$ and $\sum_{k=1}^{q} x_{jk}^2 = q$. Plug these two rules into the equation (5), we have

$$d_{ij}^2 = 2q - 2\sum_{k=1}^{q} x_{ik}x_{jk}. \qquad ...(7)$$

Comparing the (4) and (7), it is not difficult to find that d_{ij}^2 is $2q$ times that of $1 - r_{ij}$. Note, q is the number of variables!

In addition, if we are analyzing the sample rather than population, we should use the unbiased estimates, which the q should be $q - 1$. Therefore, on sample analysis, the proportion should be $2(q - 1)$.

0.2.2 Solution 2: via data calculation

```
> # import the data
> library("HSAUR2")
> df <- USArrests
> dim(df)
[1] 50  4
> names(df)
[1] "Murder"   "Assault"  "UrbanPop" "Rape"
>
> # standardize the observations
> df_matrix <- as.matrix(df)
> df_mat_obs_std <- t(scale(t(df_matrix)))
>
> # randomly choose two rows
> rand_idx <- sample(1:nrow(df_mat_obs_std),2)
> # calculate the Euclidean distance
> d_ <- dist(df_mat_obs_std[rand_idx,])
> d_sqr <- as.numeric(d_^2)
> # calcualte the correlation
> r_ <- cor(df_mat_obs_std[rand_idx[1],],
+           df_mat_obs_std[rand_idx[2],])
> # q here is 4.
> d_sqr/(1-r_)
[1] 6
```

The number of variables here is 4, since it is the sample analysis, the proportion should be $2(q - 1) = 6$, which is exactly the same to our algebra analysis!

0.3 Ex 7.2

Section 3.3 on page 65 gives a formula for calculating the proportion of the total variation (PTV) explained by the principal components. . . .

MY SOLUTION:

```
> # run pca first
> df_pca <- prcomp(df, scale=TRUE)
> summary(df_pca)
Importance of components:
                          PC1    PC2     PC3     PC4
Standard deviation     1.5749 0.9949 0.59713 0.41645
Proportion of Variance 0.6201 0.2474 0.08914 0.04336
Cumulative Proportion  0.6201 0.8675 0.95664 1.00000
```

```
> # extract the sd of each component
> var_sets <- df_pca$sdev^2
> # calculate the sum of the total variance
> total_var <- sum(var_sets)
> # calculate the PTV for each component
> (ptv_1 <- var_sets[1]/total_var)
[1] 0.6200604
> (ptv_2 <- var_sets[2]/total_var)
[1] 0.2474413
> (ptv_3 <- var_sets[3]/total_var)
[1] 0.0891408
> (ptv_4 <- var_sets[4]/total_var)
[1] 0.04335752
```

The result is exactly the same to the PTV given by the pca results.

0.4 Ex 7.3

We aim at performing hierarchical clustering on the states.

0.4.1 7.3.a

Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states.

MY SOLUTION:

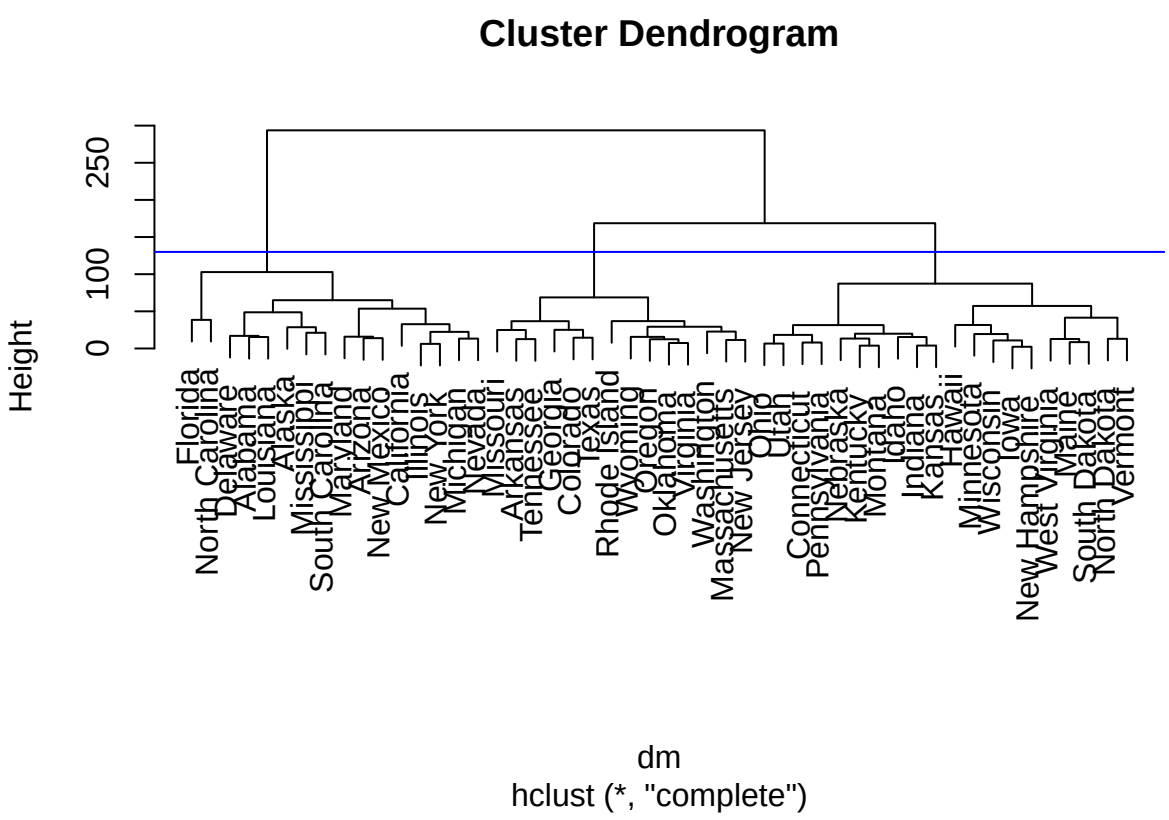
```
> # get the distance matrix
> dm <- dist(df)
> # run the hierarchical clustering
> cc <- hclust(dm,method = "complete")
```

0.4.2 7.3.b

Cut the dendrogram at a height that results in three distinct clusters. Which states belong to which clusters?

MY SOLUTION:

```
> # plot the deprogram
> plot(cc)
> abline(h=130, col="blue")
```

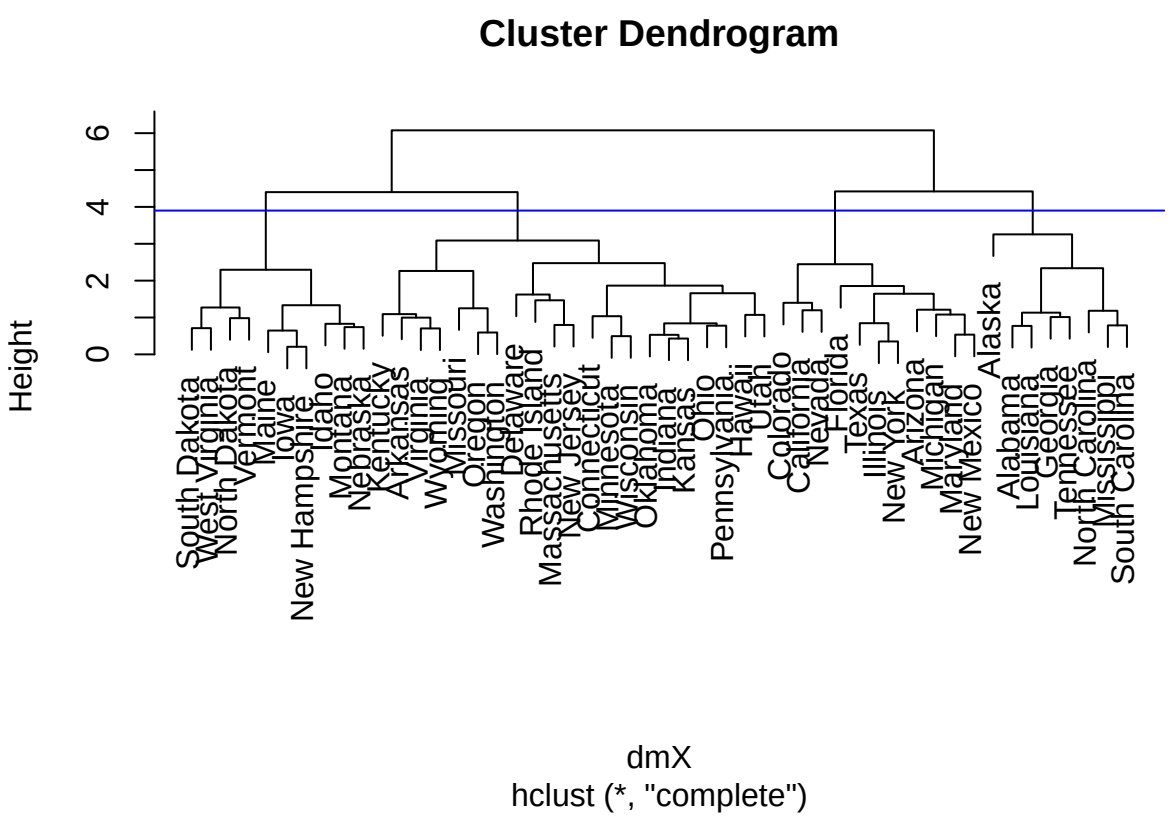


0.4.3 7.3.c

Hierarchically cluster the states using complete linkage and Euclidean distance, after scaling the variables to have standard deviation one.

MY SOLUTION:

```
> # I choose to standardize all columns
> X <- scale(df, center = T, scale = T)
> dmX <- dist(X)
> ccX <- hclust(dmX,method = "complete")
> plot(ccX)
> abline(h=3.9, col="blue")
```



0.4.4 7.3.d

What effect does scaling the variables have on the hierarchical clus- tering obtained?

MY SOLUTION:

Comparing the plots from (b) and (c), one can see that the results are different after running with same method. After checking on the raw data, we can see that the variables are at different scales. One should notice that the Euclidean distance is sensitive to the scaling of variable. That is, a large difference in one dimension can have a disproportionate effect on the overall distance, even if the differences in the other dimensions are relatively small. To avoid this limitation, I suggest that we should standardize the raw data if they are on different scales.