GTDMO - 2nd Assignment

Group 6

November 11, 2020

1 Introduction to the Problem

A survey is submitted to 15 customers of a restaurant asking them if 5 different aspects of the restaurant should be improved, namely

- M1 = variety of starters
- M2 = variety of first dishes
- M3 = variety of cakes
- M4 = speed of service
- M5 = availability of parking

The collected data are stored in the file data6.csv. Higher grades correspond to an advice of bigger improvement, while low grades mean that the customer is satisfied of the present situation.

The collected data are represented in the following table

	M1	M2	M3	M4	M5
1	0	0	1	13	7
2	8	8	2	0	2
3	12	7	14	0	1
4	0	2	1	15	11
5	0	0	0	13	9
6	1	0	1	9	10
7	0	0	0	14	14
8	9	10	10	1	1
9	0	1	0	11	10
10	0	0	0	10	6
11	1	0	0	13	14
12	0	1	0	11	11
13	0	0	2	7	11
14	13	7	8	0	1
15	4	12	15	0	0

1. Can you interpret the results in terms of ”concepts” behind the evaluation?
2. Can you group and classify the customers with respect to their attention to such concepts?

2 Solution

In order to answer the problem we can start running a preliminary analysis drawing an heatmap and looking for clusters in the columns of the dataset to better visualize the data:

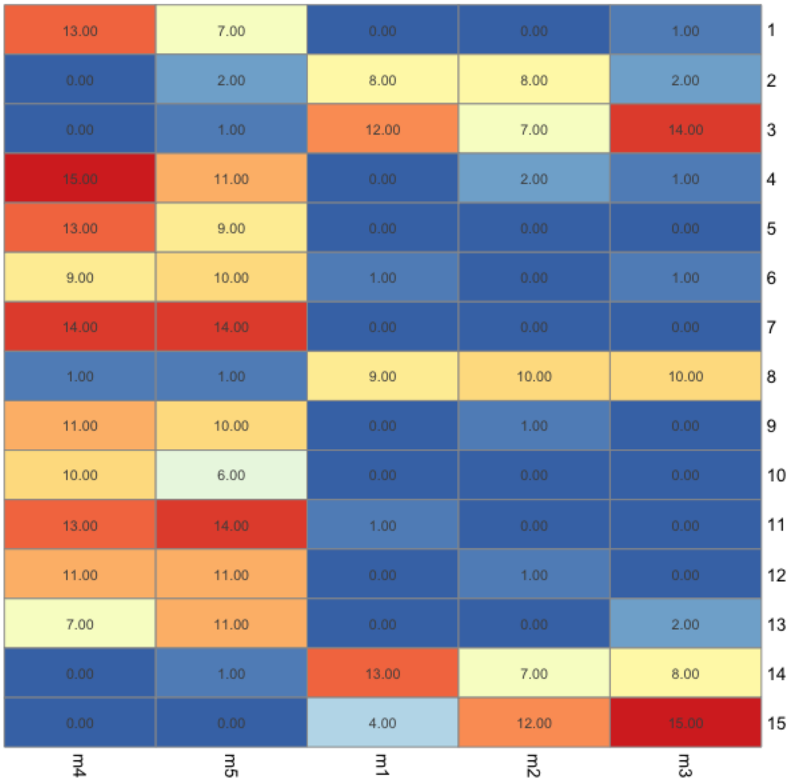


Figure 1: M Matrix heatmap clustered by columns

As the clusterization suggests, we can easily notice that the customers seems to be split in 2 different groups:

- First group strongly suggests to improve the aspects regarding the speed of service and the availability of parking (M4 and M5).
- Second group conveys that is the variety of dishes, that needs more improvement (M1, M2 and M3).

Looking at some descriptive statistics we can highlight that

- on average the suggestion of improvement is of 5 points (median)
- the point that the clients have given more often is 0 (moda)
- the criteria M1 to M3 have lower necessity of improvement compared to the global average.
- Instead, M4 and M5 are the aspects that requires major improvements.

2.1 Part 1

In order to answer the questions we begin by computing Singular Value Decomposition (SVD) of our matrix in *R*. The SVD technique decomposes a matrix M of rank(n) into three components, three new matrices of rank(n):

$$M = U\Sigma V^T \tag{1}$$

where:

- M is a $n \times d$ matrix
- Σ is a diagonal $r \times r$ matrix containing the singular values of M in a decreasing order.
- U is an orthogonal $n \times r$ matrix whose columns contain the left singular vectors of M, present if nu > 0.
- V^T is an orthogonal $r \times d$ matrix whose columns contain the right singular vectors of M, present if nv > 0. The original V matrix has, of course, dimension $d \times r$.

The rank of our original Matrix M is 5 as we can easily discover thanks to a few lines of code in *R*.

We are initially interested just in Σ among the three Matrices we decomposed M in. Σ is a diagonal Matrix containing the Singular Values on its main diagonal and 0 everywhere else. The Singular Values represent the weight of the ”concepts” of the whole information of a Matrix but when these weights are small enough they can be ignored and cut off to both reduce the matrix size and to highlight the remaining ones. From the preview obtained in the heatmap (Figure 1), we can already guess what the results of our Σ will be.

$$\Sigma = \begin{bmatrix} 50.045 & 0 & 0 & 0 & 0 \\ 0 & 36.444 & 0 & 0 & 0 \\ 0 & 0 & 9.879 & 0 & 0 \\ 0 & 0 & 0 & 6.600 & 0 \\ 0 & 0 & 0 & 0 & 6.254 \end{bmatrix} \tag{2}$$

As a matter of fact, we can state that just the first 2 values of them seem to account for more than 96% of the information energy contained in our dataset as it can be better visualized in the figures below

$$\frac{\sum_{j=1}^{2} \sigma_j^2}{\sum_{j=1}^{5} \sigma_j^2} = \frac{50.04^2 + 36.44^2}{50.04^2 + 36.44^2 + 9.87^2 + 6.6^2 + 6.25^2} = 0.96 \tag{3}$$

In Figure 2 it is possible to observe the graph rapidly tending to 0 right after the second singular value, which is exactly what we seek in SVD.

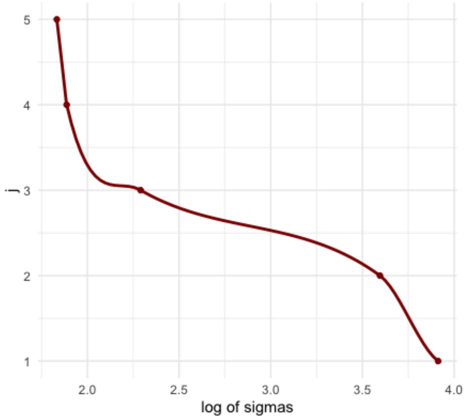


Figure 2: Plot of Log σ_j against its own j

From Figure 3 we can observe how the first singular value alone captures more than half of the total information energy while from the third on, every jump is always ticker than the previous one. Note that in Figure 3, with "Cumulative sum of sigmas over the sum of sigmas" it is actually meant

$$\frac{\sum_{j=1}^{r} \sigma_j}{\sum_{j=1}^{m} \sigma_j} \tag{4}$$

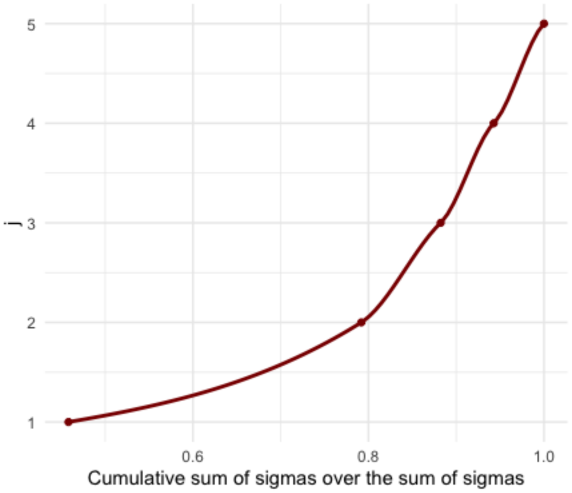


Figure 3: Ratio of the cumulative energy for every additional σ

Truncating the Singular Values and their corresponding columns in U and rows in V^T after 2 positions, we can obtain 3 new matrices $\tilde{U}$, $\tilde{\Sigma}$, $\tilde{V}$ of rank 2 which are the best approximation of M with a rank 2.

This can be formalized in the Eckart-Young Theorem which, in fact, states that the best absolute approximation to the Matrix M that has rank r is defined as:

$$\underset{\tilde{M} s.t. rank(\tilde{M})=r}{\arg\min} \quad ||M - \tilde{M}||_F \tag{5}$$

which it turns out to be exactly $\tilde{U}\tilde{\Sigma}\tilde{V^T}$.

The notation in the previous equation refers to the Frobenius norm, which is the square root of the summation of the squares of the differences between the individual matrix entries. It can be represented by the equation:

$$||A - B||_F = \sqrt{\sum_{ij} (A_{ij} - B_{ij})^2} \tag{6}$$

This is how we are able to decompose a matrix into lower rank matrices without losing much of the relevant data.

$$M \approx \tilde{U}\tilde{\Sigma}\tilde{V}^T \tag{7}$$

We can now conclude that keeping just the first 2 singular values allow us to retain more than 96% of the informational energy of our data set but reducing the rank of M from 5 to 2.

What we are left with, then, are the equations (8) and (9). The $\tilde{U}$ matrix contains the left singular vectors representing the *row to concept* similarity matrix while the $\tilde{V}$ matrix containing the right singular vectors represents the *columns to concept* similarity matrix. $\tilde{\Sigma}$ contains, as we know, the Singular Values representing the weight of the *concepts*.

$$\tilde{M} = \tilde{U}\tilde{\Sigma}\tilde{V}^T \tag{8}$$

$$\tilde{M} = \begin{bmatrix} -0.286 & -0.040 \\ -0.054 & 0.267 \\ -0.066 & 0.524 \\ -0.371 & -0.025 \\ -0.310 & -0.061 \\ -0.268 & -0.016 \\ -0.391 & -0.073 \\ -0.074 & 0.449 \\ -0.296 & -0.041 \\ -0.227 & -0.045 \\ -0.378 & -0.055 \\ -0.309 & -0.043 \\ -0.252 & -0.009 \\ -0.056 & 0.434 \\ -0.051 & 0.494 \end{bmatrix} \begin{bmatrix} 50.045 & 0 \\ 0 & 36.444 \end{bmatrix} \begin{bmatrix} -0.069 & -0.080 & -0.088 & -0.737 & -0.662 \\ 0.549 & 0.525 & 0.635 & -0.126 & -0.065 \end{bmatrix} \tag{9}$$

Now we can finally interpret the results in terms of ”concepts” dividing the 5 initial questions of our survey in 2 main groups:

1. The first group, i.e. the group whose concern is greatest, and therefore the one the restaurant should be most aware of, is mainly interested with what can be conceptualized as *services quality*:
   - M4 = speed of the service
   - M5 = parking availability
2. The second group is composed by the first 3 aspects, respectively:
   - M1 = variety of starters
   - M2 = variety of first dishes
   - M3 = variety of cakes

   These 3 aspects seems instead to have a slightly minor weight in customer’s concerns. They can be summarized in the concept of *courses variety* .

2.2 Part 2

The second question can be addressed just by looking at the $\tilde{U}$ matrix that, as stated above, proposes the strength of the similarity between customers and the concepts.

$$\tilde{U} = \begin{bmatrix} -0.286 & -0.040 \\ -0.054 & 0.267 \\ -0.066 & 0.524 \\ -0.371 & -0.025 \\ -0.310 & -0.061 \\ -0.268 & -0.016 \\ -0.391 & -0.073 \\ -0.074 & 0.449 \\ -0.296 & -0.041 \\ -0.227 & -0.045 \\ -0.378 & -0.055 \\ -0.309 & -0.043 \\ -0.252 & -0.009 \\ -0.056 & 0.434 \\ -0.051 & 0.494 \end{bmatrix} \tag{10}$$

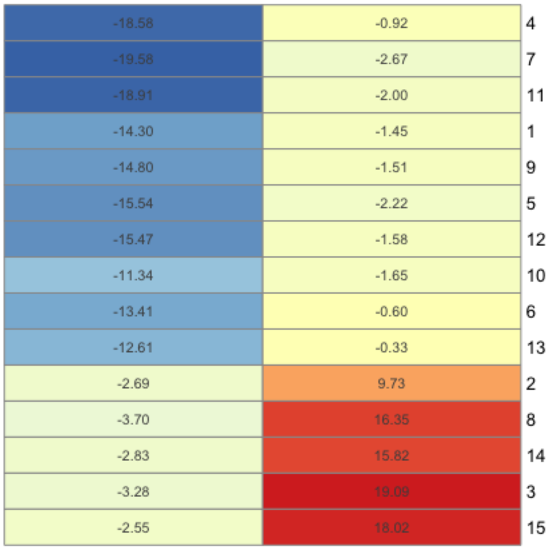
We can further analyze this classification by computing the *"similarity to the concepts" score* of every single user and comparing them together. If we represent the customer with her/his new surveys preferences vector $\vec{m}$, we can compute a score vector as

$$\vec{s}_j = \vec{u}_j \tilde{V} \tag{11}$$

where $\vec{u}_j$ is the vector related to customer j and V is the matrix of our SVD mapping the aspects of the survey to the concepts. Note that we could do this for every new customer and check for their affiliation to the concepts. If we want this for every customer, we can simply multiply

$$S = \tilde{M}\tilde{V} \tag{12}$$

obtaining



-18.58	-0.92	4
-19.58	-2.67	7
-18.91	-2.00	11
-14.30	-1.45	1
-14.80	-1.51	9
-15.54	-2.22	5
-15.47	-1.58	12
-11.34	-1.65	10
-13.41	-0.60	6
-12.61	-0.33	13
-2.69	9.73	2
-3.70	16.35	8
-2.83	15.82	14
-3.28	19.09	3
-2.55	18.02	15

Figure 4: Score Customers Matrix

Where the $j-th$ row represent the score vector of user j.

If we assign a number to every customer we can easily classify them in 2 groups:

1. Customers $\{1, 4, 5, 6, 7, 9, 10, 11, 12, 13\}$, i.e. the 66% of the interviewed sample, are heavy related to the concept represented by the first singular value, therefore They are more concerned with improving *services quality* of the restaurant.
2. Customers $\{2, 3, 8, 14, 15\}$, 33% circa of the sample, are instead highly related to the concept represented by the second singular value and they are indeed more interested in improving the *courses variety* of the restaurant.

2.3 Appendix

For the sake of Reproducibility Research, here I attach the full commented R code used to obtain all Figures, Tables, equations and results.

```
1  # "library" is the command used to upload all the content of each quoted package.
2  library(Matrix)
3  library(readr)
4  library(matlib)
5  library(plm)
6  library(stargazer)
7  library(pheatmap)
8  library(pracma)
9  library(ggplot2)
10 
11 # "file.choose" and "read.csv2" help us to pick up and read our dataset.
12 data6<-file.choose()
13 data6<-read.csv2(data6,header=TRUE,sep=",")
14 
15 # "data[,2:6]" removes the x column, which is in particular the first one, that is useless to our analysis.
16 data6<-data6[,2:6]
17 
18 # "rankMatrix" computes the rank, which is the number of independent columns of the matrix.
19 rankMatrix(data6)
20 
21 # The "svd_notedited" variable is the one that stores the information of the Singular Value Decomposition of matrix,
22 # done with the number of vectors u and v equal to the original rank of the matrix, which is 5.
23 svd_notedited <- svd(data6)
24 
25 # "pheatmap" shows the heatmap of the original matrix so that it's possible to make a first empirical analysis.
26 pheatmap(data6, how_colnames=T, show_colnames=T, cluster_rows = F, cluster_cols = T, display_numbers=T, legend=F)
27 
28 # Then the next commands, from row 28 to 32, define new matrix variables, which are originated by the SVD or computed by it.
29 U_notedited <- svd_notedited$u
30 V_notedited <- svd_notedited$v
31 Vt_notedited <- t(svd_notedited$v) # Vt is the transposed matrix of V
32 D_notedited <- diag(svd_notedited$d, nrow = 5,ncol = 5) # D is the diagonal matrix of d
33 d_notedited <- svd_notedited$d
34 
35 # Again "pheatmap" rappresents the matrix D, so that it's easy to see the optimal number of vectors for the SVD.
36 pheatmap(D_notedited,cluster_rows = F,cluster_cols = F,legend = F,display_numbers = T)
37 
38 # This time "svd_data" is the variable used to collect all the "svd" function output,
39 # computed using the two-dimensional vectors u and v. The best fit will be then a matrix of rank equal to two.
40 svd_data <- svd(data6, nu=2,nv=2)
41 
42 # Again, from row 42 to 45, new variables are created to easily access the SVD data.
43 U <- svd_data$u
44 V <- svd_data$v
45 Vt <- t(svd_data$v) # Vt is the transposed matrix of V
46 D <- diag(svd_data$d, nrow = 2,ncol = 2) # D is the diagonal matrix of d
47 
48 # Here is a command that creates a new variable Mb, that is our best fit made by the matrix product of U times D times V transposed.
49 Mb <- U%*%D%*%Vt
50 
51 # "rankMatrix" to check if the rank of Mb is actually 2.
52 rankMatrix(Mb)
53 
```

```
# A little additional check to see if the rank is 2 is made by computing the echelon form with the function "echelon".
M_ech <- echelon(Mb, reduced = T, verbose= T)

# "len_d" is a vector 1x5.
len_d<- c(1,2,3,4,5)

# "ggplot" it’s used to make the graph of the original d vector and puts it against the number of columns of the initial matrix
ggplot()+
        geom_smooth(col="darkred")+
        theme_minimal()+
        geom_point(col= "darkred")+
        aes(log(d_notedited),len_d, label = ("dfs"))+
        labs(x = "lenght of m", y =’log of sigmas’)

# We can clearly see how in the first two sigmas is concentrated the power of information
# the cumulative sum of sigmas over its actual total sum of sigmas. we can clearly
# see how just the first two sigmas cover more than 80% of the total information.

# Here, instead, it’s used to rappresent the Cumulative sum of sigmas over the sum of sigmas
ggplot()+
        geom_smooth(col="darkred")+
        theme_minimal()+
        geom_point(col= "darkred")+
        aes(log(cumsum(d_notedited)/sum(d_notedited)),len_d)+
        labs(x = "", y="Cumsum(s)/sum(s)",title = ’Cumulative sum of sigmas over the sum of sigmas’)


# The process of Singular Value Decomposition clearly indicates that exists only two main
# "concepts" or groups of customers, the first group is mainly concerned with what we can call Services
# i.e. the services speed and the availability of parking. The second group is mainly concerned with
# the variety of courses.

# Now it’s time to see the result of our best fit and see the possible conclusions
pheatmap(Mb, cluster_rows = T, cluster_cols = F, display_numbers=T, legend=F, show_rownames = T)

#let’s look for the similarity of the other users:
Scores_users<-Mb%*%svd_data$v
Scores_users

# and rappresent them
pheatmap(Scores_users,cluster_rows = T,cluster_cols = F,display_numbers=T, legend=F, show_rownames = T)

# compute cos(theta) all users
# Compute and plot similarity matrix
similarity <- matrix(NA, nrow = 15, ncol = 15)
for (i in c(1:15)){
        for (j in c(1:15)){
                similarity[i, j]<-dot(S[i,],S[j,])/(sqrt(sum(S[i,]^2)*sum((S[j,])^2)))
        }
}
```