EE412 Foundation of Big Data Analytics, Fall 2023
HW2

Name: 권혁태

Student ID: 20180036

Discussion Group (People with whom you discussed ideas used in your answers):

On-line or hardcopy documents used as part of your answers:

Answer to Problem 1

1-a)
If the string sets is given like below:
{ abcde, acfdeg, bcde, abcdf, fdeg }

Minimizing the sum of distance

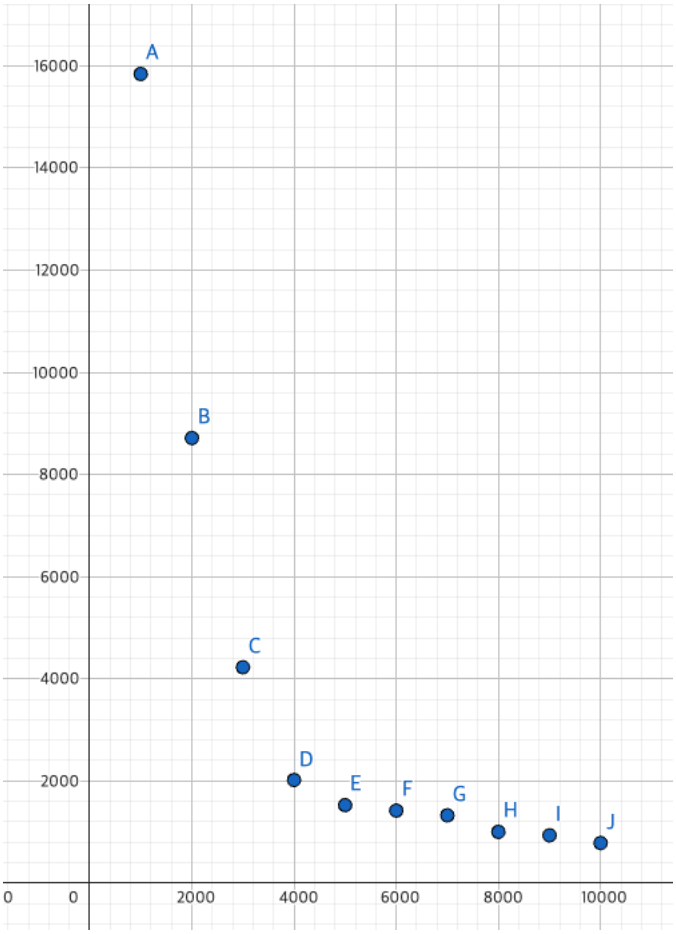
‘abcde’: 3 + 1 + 2 + 5 = 11
‘acfdeg’: 3 + 4 + 3 + 2 = 12
‘bcde’ : 1 + 4 + 3 + 4 = 12
‘abcdf’: 2 + 3 + 3 + 5 = 13
‘fdeg’: 5 + 2 + 4 + 5 = 16

The clustroid is ‘abcde’

Minimizing the maximum distance
The ‘acfdeg’ or ‘bcde’ would be the clustroid which are different from’abcde’

1-b)
(1, 15840.015935162377)
(2, 8712.758000610926)
(3, 4225.773466007263)
(4, 2018.267391109598)
(5, 1526.2137994123182)
(6, 1420.7214130743844)
(7, 1326.9455957023163)
(8, 1003.3225742993732)

(9, 935.1907815482737)
(10, 784.7882449282713)
(11, 620.9149832695052)



Scale 이 안 맞아서, K=1 인 경우를 1000 으로 놓고 plot 을 그렸습니다.

I might think the best number for k is 4, because after 4, the graph's slope is similar

Answer to Problem 2

Exercise 11.1.7

a) [0.19382266 0.47224729 0.8598926]
b) 7.872983346207416
c) [[0.70423389 0.27936833 -0.31216389]
   [0.27936833 0.24418694 -0.19707639]
   [-0.31216389 -0.19707639 0.17859582]]
d) second eigenvector: [0.81649658 0.40824829 -0.40824829]
   second eigenvalue: 1.000000
e) construct new matrix
   [[0.03756722 -0.053965   0.02116944]
   [-0.053965   0.07752027 -0.03040973]
   [0.02116944 -0.03040973 0.01192916]]
   Third eigenvector: [0.54384383 -0.78122713 0.30646053]
   Third eigenvalue: 0.1270166537925833

My python code

```python
import numpy as np
from numpy.linalg import norm

matrix = np.array([
    [1,1,1],
    [1,2,3],
    [1,3,6]
])
def power_iteration(matrix, num_iterations):
    n = matrix.shape[0]
    v = np.ones(n)
    eigenvalue = 0

    for i in range(num_iterations):
        Av = np.dot(matrix, v)
        eigenvalue = norm(Av)
        v = Av / eigenvalue
    return eigenvalue, v

first_eigval, first_eigvec = power_iteration(matrix,50)

print('first eigenvector : ',first_eigvec)
print('first eigenvalue : ',first_eigval)

first_eigvec = first_eigvec.reshape(first_eigvec.shape[0],-1)
first_eigvec_T = first_eigvec.reshape( -1, first_eigvec.shape[0])
```

```
print(first_eigvec * first_eigvec_T)

new_matrix = matrix - first_eigval * first_eigvec * first_eigvec_T
print('second_matrix \n',new_matrix)

second_eigval, second_eigvec = power_iteration(new_matrix, 50)
print('second eigenvector : ' , second_eigvec)
print('second eigenvalue : ',second_eigval)
second_eigvec = second_eigvec.reshape(second_eigvec.shape[0],-1)
second_eigvec_T = second_eigvec.reshape( -1, second_eigvec.shape[0])

last_matrix = new_matrix - second_eigvec * second_eigvec_T
print('third matrix \n',last_matrix)
third_eigval, third_eigvec = power_iteration(last_matrix, 50)
print('third eigenvector : ' , third_eigvec)
print('third eigenvalue : ',third_eigval)
```

Exercise 11.3.1

a) M^T * M
[[36 37 38]
[37 49 61]
[38 61 84]]
M* M^T
[[14 26 22 16 22]
[26 50 46 28 40]
[22 46 50 20 32]
[16 28 20 20 26]
[22 40 32 26 35]]

b) Eigenpairs = (eigen value, eigen vectors)

Eigenpairs for M^T * M
(153.566996, [-0.40928285 -0.56345932 -0.7176358])
(15.4330035, [-0.81597848 -0.12588456 0.56420935])

Eigenpairs for M* M^T
(153.566996, [0.29769568 0.57050856 0.52074297 0.32257847 0.45898491])
(15.4330035, [-0.15906393 0.0332003   0.73585663 -0.5103921 -0.41425998]))

Answer to Problem 3

a) Exercise 9.3.1

	a	b	c	d	e	f	g	h
A	4	5		5	1		3	2
B		3	4	3	1	2	1	
C	2		1	3		4	5	3

Table 1 is a utility matrix, representing the ratings, on a 1–5 star scale, of eight items, a through h, by three users A, B, and C. Compute the following from the data of this matrix.

a) Treating the utility matrix as boolean, compute the Jaccard distance between each pair of users. (Hint: Blank is false, and others are true.)

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**
A	1	1	0	1	1	0	1	1
B	0	1	1	1	1	1	1	0
C	1	0	1	1	0	1	1	1

A ↔ B : 1- 4/8 = 4/8

B ↔ C : 1 - 4/8 = 4/8

C ↔ A : 1 - 3/8 = 3/8

b) Repeat Part (a), but use the cosine distance.

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**
A	1	1	0	1	1	0	1	1
B	0	1	1	1	1	1	1	0
C	1	0	1	1	0	1	1	1

A: (1,1,0,1,1,0,1,1) , B: (0,1,1,1,1,1,1,0), C: (1,0,1,1,0,1,1,1)

A ↔ B : 1- 0.666667 = 0.333333

B ↔ C : 1 - 0.666667 = 0.333333

C ↔ A : 1 - 0.666667 = 0.333333

c) Treat ratings of 3, 4, and 5 as 1 and 1, 2, and blank as 0. Compute the Jaccard distance between each pair of users.

	a	b	c	d	e	f	g	h
A	1	1	0	1	0	0	1	0
B	0	1	1	1	0	0	0	0
C	0	0	0	1	0	1	1	1

A: (1,1,0,1,0,0,1,0)

B: (0,1,1,1,0,0,0,0)

C: (0,0,0,1,0,1,1,1)

A ↔ B : 1- 5/8 = 3/8

B ↔ C : 1 - 3/8 = 5/8

C ↔ A : 1 - 4/8 = 4/8

d) Repeat Part (c), but use the cosine distance.

A ↔ B : 1- 0.57735 = 0.422650

B ↔ C : 1 - 0.288675 = 0.711325

C ↔ A : 1 - 0.5 = 0.500000

e) Normalize the matrix by subtracting from each nonblank entry the average value for its user.

A 평균: 20/6 = 3.3

B 평균: 14/6 = 2.3

C 평균: 18/6 = 3

	a	b	c	d	e	f	g	h
A	4 - 3.3 = 0.7	5- 3.3=1.7		5 - 3.3 = 1.7	1 - 3.3 = -2.3		3 - 3.3 = -0.3	2 - 3.3 = -1.3
B		3- 2.3=0.7	4 - 2.3 = 1.7	3 - 2.3 = 0.7	1 - 2.3 = -1.3	2 - 2.3 = -0.3	1 - 2.3 = -1.3	
C	2 - 3 = -1		1 - 3 = -2	3 - 3 = 0		4 - 3 = 1	5 - 3 = 2	3 - 3 = 0

	a	b	c	d	e	f	g	h
A	0.7	1.7		1.7	-2.3		-0.3	-1.3
B		0.7	1.7	0.7	-1.3	- 0.3	-1.3	
C	-1		-2	0		1	2	0

f) using the normalized matrix from Part (e), compute the cosine distance between each pair of users.

A: (0.7, 1.7, 0, 1.7, -2.3,0,-0.3, -1.3), B: (0,0.7,1.7,0.7,-1.3,-0.3,-1.3), C: (-1,0,-2,0,0,1,2,0) A ↔ B : 1- 0.582099 = 0.417901

B ↔ C : 1 - (- 0.112555) = 0.711325

C ↔ A : 1 - (-0.735347) = 1.735347

Exercise 9.3.2

a) Cluster the eight items hierarchically into four clusters. The following method should be used to cluster. Replace all 3's, 4's, and 5's by 1 and replace 1's, 2's, and blanks by 0. use the Jaccard distance to measure the distance between the resulting column vectors. For clusters of more than one element, take the distance between clusters to be the minimum distance between pairs of elements, one from each cluster. (It is okay that you only choose one possible result with the detailed explanation.)

	a	b	c	d	e	f	g	h
A	1	1	0	1	0	0	1	0
B	0	1	1	1	0	0	0	0
C	0	0	0	1	0	1	1	1

a : (1,0,0), b : (1,1,0), c : (0,1,0), d : (1,1,1) , e : (0,0,0) , f : (0,0,1) , g : (1,0,1), h : (0,0,1)

	a	b	c	d	e	f	g	h
a	x	1/3	2/3	2/3	1/3	2/3	1/3	2/3
b		x	1/3	1/3	2/3	1	2/3	1
c			x	2/3	1/3	2/3	1	2/3
d				x	1	2/3	1/3	2/3
e					x	1/3	2/3	1/3
f						x	1/3	0
g							x	1/3
h								x

(a,b) with distance: 1/3

(c,e) with distance: 1/3

(d,g) with distance: 1/3

(f,h) with distance: 0

b) Then, construct from the original matrix of Table 1 a new matrix whose rows correspond to users, as before, and whose columns correspond to clusters. Compute the entry for a user and cluster of items by averaging the nonblank entries for that user and all the items in the cluster.

	(a,b)	(c,e)	(d,g)	(f,h)
A	4.5	1	4	2
B	3	2.5	2	2
C	2	1	3	3.5

(c) Compute the cosine distance between each pair of users, according to your matrix from Part b

A: (4.5,1,4,2)

B: (3,2.5,2,2)

C: (2,1,3,3.5)

A ↔ B : 1 - 0.904138 = 0.095862

B ↔ C : 1 - 0.870287 = 0.129713

C ↔ A : 1 - 0.881295 = 0.118705

b) Collaborative Filtering

User-based
175 5.0
261 5.0
440 5.0
480 5.0
527 5.0

Item-based
5 5.0
318 5.0
364 5.0
785 5.0
1 4.5

c) Movie Recommendation Challenge

I use collaborative filtering based on user data.
If I can’t get the similar ratings, I just use the average of that movie