Toghrul Tahirov: G47609664

Exercise 8

1.0 Given the following Document Term matrix calculate the Item/Item matrix:

	Term1	Term2	Term3	Term4	Term5	Term6	Term7	Term8
Item 1	3	0	2	2	0	0	4	3
Item 2	0	0	4	3	2	0	0	2
Item 3	2	2	0	2	2	1	0	0
Item 4	0	1	0	2	2	0	1	0
Item 5	0	0	0	0	0	2	0	0
Item 6	2	1	3	4	2	2	0	2

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
Item 1		20	10	8	0	26
Item 2	20		10	10	0	32
Item 3	10	10		10	2	20
Item 4	8	10	10		0	13
Item 5	0	0	2	0		4
Item 6	26	32	20	13	4	

a. Determine the Item Relationship matrix using a threshold of 8 or higher

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
Item 1	0	1	1	1	0	1
Item 2	1	0	1	1	0	1
Item 3	1	1	0	1	0	1
Item 4	1	1	1	0	0	1
Item 5	0	0	0	0	0	0
Item 6	1	1	1	1	0	0

b. Determine the clusters using the clique technique (the below deduction are made by taking item 1 as the primary point of comparison)

Class 1: 1, 2, 3, 4, 6
Class 2: 5

c. Determine the clusters using the single link technique. The same results as above apply in this case as well.
Class 1: 1, 2, 3, 4, 6
Class 2: 5

d. Determine the clusters using the star technique where the item selected for the new seed for the next star is the smallest number item nor already part of a class.
Class 1: 1, 2, 3, 4, 6
Class 2: 5

e. If for question a.if you used a threshold of 11 or higher discuss how it would changes the results fromclique, single link, star from above.Generate a new Item relationship matrix based upon 11 as threshold.

The item relationship matrix would look like:

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
Item 1		1	0	0	0	1
Item 2	1		0	0	0	1
Item 3	0	0		0	0	1
Item 4	0	0	0		0	1
Item 5	0	0	0	0		0
Item 6	1	1	1	1	0	

Clique – now is:

- Class 1: 1, 2, 6
- Class 2: 3, 6
- Class 3: 4, 6
- Class 4: 5

Single link:

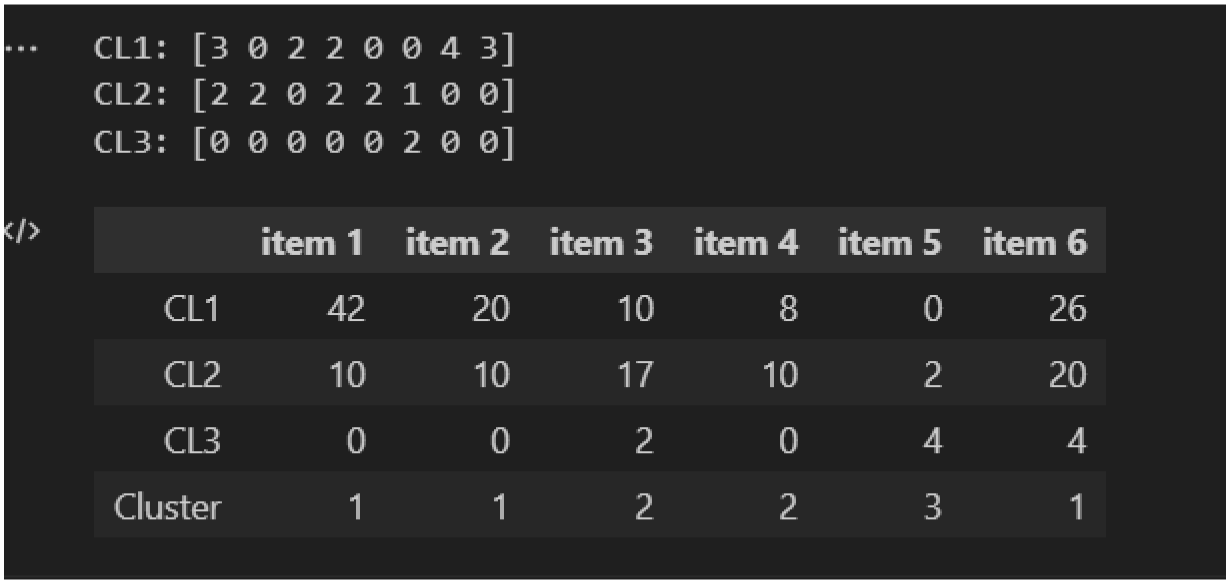
- Class 1: 1, 2, 6, 3, 4
- Class 2: 5

Star :

- Class 1: 1, 2, 6
- Class 2: 3, 6
- Class 3: 4, 6
- Class 4: 5

2. K-means clustering

a. Using the data from problem 1.0 above, assuming K=3 for K-means clustering approach (using existing clusters) – with CL1 = Item1; CL2 = Item3, CL3 = Item5 calculate the new clusters. If there are multiple classes with same number select the class it was in if it's an option.



```
CL1: [3 0 2 2 0 0 4 3]
CL2: [2 2 0 2 2 1 0 0]
CL3: [0 0 0 0 0 2 0 0]
```

	item 1	item 2	item 3	item 4	item 5	item 6
CL1	42	20	10	8	0	26
CL2	10	10	17	10	2	20
CL3	0	0	2	0	4	4
Cluster	1	1	2	2	3	1

Centroid of Class 1 =
Centroid of Class 2 =
Centroid of Class 3 =

SIM(CL1, I1) = 42
SIM(CL2, I1) = 10
SIM(CL3, I1) = 0
I1 belongs to CL1.

SIM(CL1, I2) = 20
SIM(CL2, I2) = 10
SIM(CL3, I2) = 0
I2 belongs to CL1

SIM(CL1, I3) = 10
SIM(CL2, I3) = 17
SIM(CL3, I3) = 2
I3 belongs to Cl2

SIM(CL1, I4) = 8
SIM(CL2, I4) = 10
SIM(CL3, I4) = 0
I4 belongs to CL2

SIM(CL1, I5) = 0
SIM(CL2, I5) = 2

SIM(CL3, I5) = 4
I5 belongs to CL3

SIM(CL1, I6) = 26
SIM(CL2, I6) = 20
SIM(CL3, I6) = 4
I6 belongs to CL1

New Classes are:

Class 1= I1, I2, I6
Class 2 = I3, 4
Class 3 = I5

REPEAT above until items do not move between classes
If we iterate over the calculation at the 3rd iteration the clusters stabilize and there is no longer any redistribution of the clusters. The resultant distribution of items per cluster is as follows:
Class 1= I1, I2, I4, I6
Class 2 = I3
Class 3 = I5

Results can be verified from the code output in the appendices section.

b. Compare the results to the clique cluster from a. Why are they different.

Clique:
CL 1: Items 1, 2, 3, 4, 6
CL 2: Item 5

K-means:
Class 1= Items 1, 2, 4, 6
Class 2 = 3
Class 3 = 5

It is lucid that the results from part a (using the clique clustering technique) are different from part b (using K-means clustering with existing clusters) because the two techniques use different approaches for grouping items.

The clique clustering technique groups items based on their mutual similarities, while K-means clustering partitions the items into K clusters based on their distances from the cluster centers.

In part b, we are using existing clusters as the initial cluster centers for K-means clustering. This approach may lead to different results compared to clique clustering as it depends on the initial selection of cluster centers and the distance measure used.

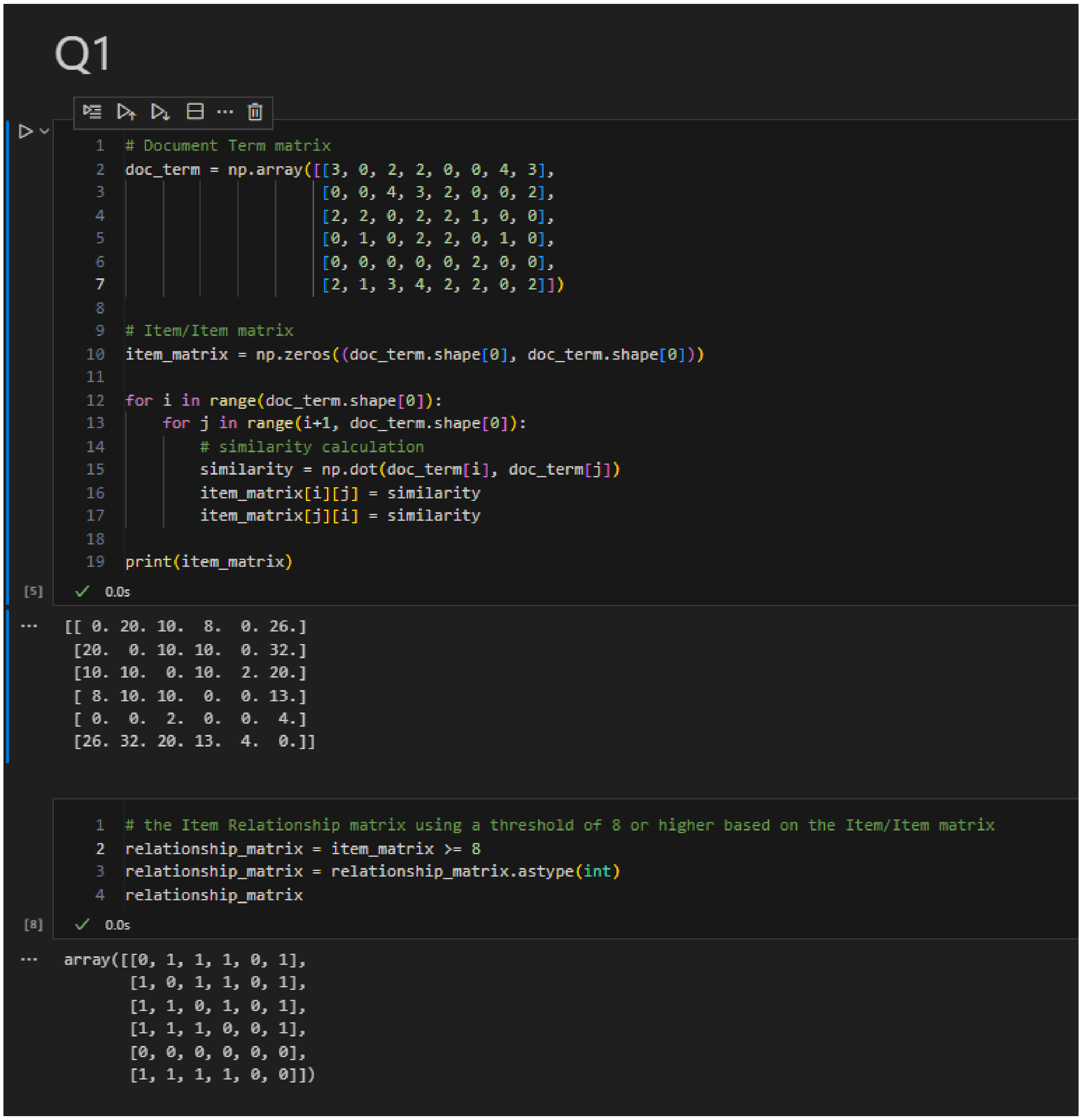
Moreover, when computing the cluster distribution using via the clique clustering method, we are manually setting the threshold. As we have already observed in Task 1, the threshold plays a major role in the number of clusters and final item combinations.

Therefore, it is not surprising to see differences in the results between the two techniques.

Appendices

This section holds the python code snippets used for the tasks

Toghrul Tahirov: G47609664



Q1

```
# Document Term matrix
doc_term = np.array([[3, 0, 2, 2, 0, 0, 4, 3],
                     [0, 0, 4, 3, 2, 0, 0, 2],
                     [2, 2, 0, 2, 2, 1, 0, 0],
                     [0, 1, 0, 2, 2, 0, 1, 0],
                     [0, 0, 0, 0, 0, 2, 0, 0],
                     [2, 1, 3, 4, 2, 2, 0, 2]])

# Item/Item matrix
item_matrix = np.zeros((doc_term.shape[0], doc_term.shape[0]))

for i in range(doc_term.shape[0]):
    for j in range(i+1, doc_term.shape[0]):
        # similarity calculation
        similarity = np.dot(doc_term[i], doc_term[j])
        item_matrix[i][j] = similarity
        item_matrix[j][i] = similarity

print(item_matrix)
```

[5] 0.0s

```
[[ 0. 20. 10.  8.  0. 26.]
 [20.  0. 10. 10.  0. 32.]
 [10. 10.  0. 10.  2. 20.]
 [ 8. 10. 10.  0.  0. 13.]
 [ 0.  0.  2.  0.  0.  4.]
 [26. 32. 20. 13.  4.  0.]]
```

```
# the Item Relationship matrix using a threshold of 8 or higher based on the Item/Item matrix
relationship_matrix = item_matrix >= 8
relationship_matrix = relationship_matrix.astype(int)
relationship_matrix
```

[8] 0.0s

```
array([[0, 1, 1, 1, 0, 1],
       [1, 0, 1, 1, 0, 1],
       [1, 1, 0, 1, 0, 1],
       [1, 1, 1, 0, 0, 1],
       [0, 0, 0, 0, 0, 0],
       [1, 1, 1, 1, 0, 0]])
```

Toghrul Tahirov: G47609664

```
# Question 2

# Document Term matrix
doc_term = np.array([[3, 0, 2, 2, 0, 0, 4, 3],
                     [0, 0, 4, 3, 2, 0, 0, 2],
                     [2, 2, 0, 2, 2, 1, 0, 0],
                     [0, 1, 0, 2, 2, 0, 1, 0],
                     [0, 0, 0, 0, 0, 2, 0, 0],
                     [2, 1, 3, 4, 2, 2, 0, 2]])

# Existing clusters
clusters = {"CL1": [0], "CL2": [2], "CL3": [4]}
CL1, CL2, CL3 = [], [], []


[CL1.append(doc_term[item]) for item in clusters["CL1"]]
CL1 = np.array(CL1).sum(axis=0) / len(clusters["CL1"])
[CL2.append(doc_term[item]) for item in clusters["CL2"]]
CL2 = np.sum(np.array(CL2), axis=0) / len(clusters["CL2"])
[CL3.append(doc_term[item]) for item in clusters["CL3"]]
CL3 = np.sum(np.array(CL3), axis=0) / len(clusters["CL3"])


print(">>> Cluster Centroids before the start of the iterations:")
print(f"CL1: {CL1}")
print(f"CL2: {CL2}")
print(f"CL3: {CL3}")

print(f">>> Clusters before the start of the iterations:")
print(f"CL1: {[i+1 for i in clusters['CL1']]}")
print(f"CL2: {[i+1 for i in clusters['CL2']]}")
print(f"CL3: {[i+1 for i in clusters['CL3']]}")

print('\n' + '-'*50 + '\n')

for i in range(5):
    CL1, CL2, CL3 = [], [], []

    [CL1.append(doc_term[item]) for item in clusters["CL1"]]
    CL1 = np.array(CL1).sum(axis=0) / len(clusters["CL1"])
    [CL2.append(doc_term[item]) for item in clusters["CL2"]]
    CL2 = np.sum(np.array(CL2), axis=0) / len(clusters["CL2"])
    [CL3.append(doc_term[item]) for item in clusters["CL3"]]
    CL3 = np.sum(np.array(CL3), axis=0) / len(clusters["CL3"])
```

Toghrul Tahirov: G47609664

Code output for Question 2

```
>>> Cluster Centroids before the start of the iterations:

CL1: [3. 0. 2. 2. 0. 0. 4. 3.]

CL2: [2. 2. 0. 2. 2. 1. 0. 0.]

CL3: [0. 0. 0. 0. 0. 2. 0. 0.]

>>> Clusters before the start of the iterations:

CL1: [1]

CL2: [3]

CL3: [5]


--------------------------------------------------


>>> Cluster Centroids at the start of iteration 1

CL1: [3. 0. 2. 2. 0. 0. 4. 3.]

CL2: [2. 2. 0. 2. 2. 1. 0. 0.]

CL3: [0. 0. 0. 0. 0. 2. 0. 0.]


>>> Clusters at the end of iteration 1:

CL1: [1, 2, 6]

CL2: [3, 4]

CL3: [5]


--------------------------------------------------


>>> Cluster Centroids at the start of iteration 2

CL1: [1.66666667 0.33333333 3.   3.   1.33333333 0.66666667 1.33333333  2.33333333]

CL2: [1.  1.5 0.  2.  2.  0.5 0.5 0. ]

CL3: [0. 0. 0. 0. 0. 2. 0. 0.]
```

Toghrul Tahirov: G47609664

```
>>> Clusters at the end of iteration 2:

CL1: [1, 2, 4, 6]

CL2: [3]

CL3: [5]

--------------------------------------------------

>>> Cluster Centroids at the start of iteration 3

CL1: [1.25 0.5  2.25 2.75 1.5  0.5  1.25 1.75]

CL2: [2. 2. 0. 2. 2. 1. 0. 0.]

CL3: [0. 0. 0. 0. 0. 2. 0. 0.]

>>> Clusters at the end of iteration 3:

CL1: [1, 2, 4, 6]

CL2: [3]

CL3: [5]

--------------------------------------------------

>>> Cluster Centroids at the start of iteration 4

CL1: [1.25 0.5  2.25 2.75 1.5  0.5  1.25 1.75]

CL2: [2. 2. 0. 2. 2. 1. 0. 0.]

CL3: [0. 0. 0. 0. 0. 2. 0. 0.]

>>> Clusters at the end of iteration 4:

CL1: [1, 2, 4, 6]

CL2: [3]

CL3: [5]
```

Toghrul Tahirov: G47609664

```
--------------------------------------------------

>>> Cluster Centroids at the start of iteration 5
CL1: [1.25 0.5  2.25 2.75 1.5  0.5  1.25 1.75]
CL2: [2. 2. 0. 2. 2. 1. 0. 0.]
CL3: [0. 0. 0. 0. 0. 2. 0. 0.]

>>> Clusters at the end of iteration 5:
CL1: [1, 2, 4, 6]
CL2: [3]
CL3: [5]

--------------------------------------------------
```