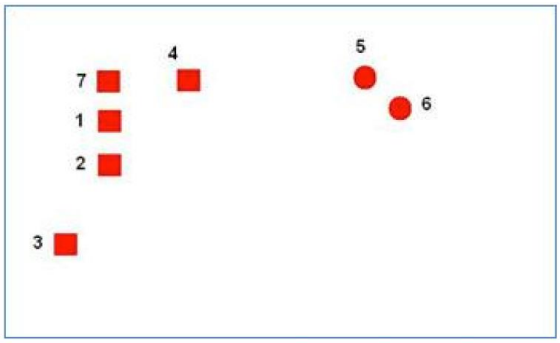
CS 422 FALL 2018

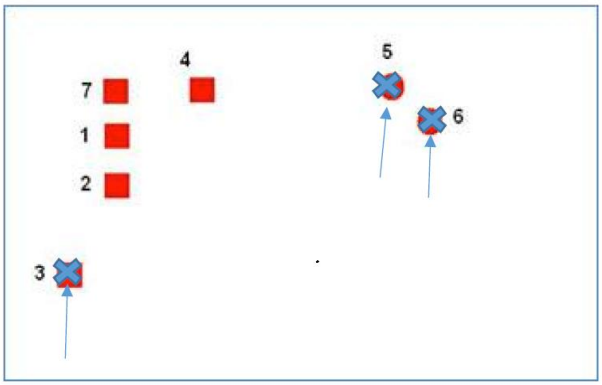
HOMEWORK 6 TA SOLUTION

I Clustering

1)



- In K-means clustering algorithm, a centroid governs each cluster and the points closest to the centroid are allocated to that particular cluster.
- Given for k=3, initial centroids as 3,5,6 we will assign remaining points to the nearest centroids and thus cluster. Centroids are marked in cross signs as shown below.



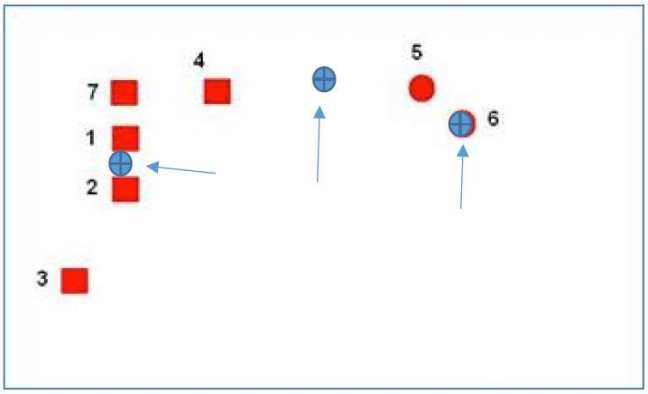
- After 1st iteration,

  Cluster 1(Centroid-3) – 3, 2, 1, 7

  Cluster 2(Centroid-5)- 4, 5

  Cluster 3(Centroid-6)- 6

  - Euclidean distance for each point with the centroids are calculated, which will decide about the points belong to which cluster based on smallest distance. According to the rule, above points are assigned to 3 clusters. Centroids are recomputed after allocation and new positions are shown with '+' signs as below.
  - As shown by the blue arrow marks, its evident that the cluster centroid for points 3, 2, 1 and 7 has gone closer to point 4 because of the presence of point 2, 1 and 7. Also, the cluster centroid of points 4 and 5 moved far from 5 because of the presence of point 4.



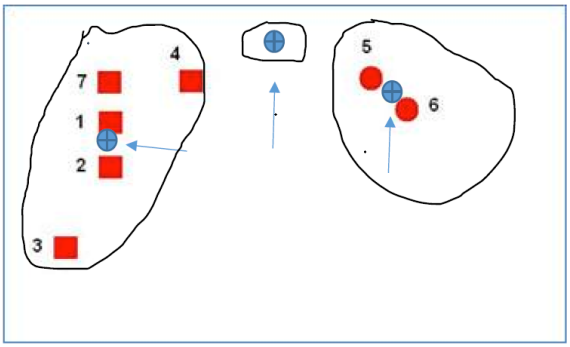
- After 2nd iteration,

  Cluster 1 - 3, 2, 1, 7, 4

  Cluster 2 – { }

  Cluster 3 – 5, 6

  - Point 4 falls in the cluster 1 as now it is closer to the centroid of data points 3,2,1 and 7 with point 5 being closer to the cluster 3 containing point 6. Thus, cluster 2(in the middle) becomes empty.
  - As shown below, points 4 and 5 moved to cluster 1 and cluster 3 respectively and thereby the cluster 2(in the middle) has now become an empty cluster as it does not contain any data points because the euclidean distance of any data point from the centroid of this empty cluster will always be greater than the distance from the centroid of the other two clusters formed.



Analysis:

- K-means clustering selects k random centroids and the points are assigned to the initial centroids, which are all in the larger group of points. After the points are assigned to the centroid, the centroid is then updated.
- When the K-means algorithm terminates, because no more changes occur, the centroids have identified the natural groupings of points.
- Choosing the proper initial centroid is the key step of the basic k-means procedure. A common approach is to choose the initial centroids randomly, but the resulting clusters are often poor and may also result in empty cluster as illustrated above.

2)

- If there is a chance of an empty clustering to occur, then the clustering cannot be the global minimum solution based on RSS. RSS (Residual Sum of Squares) is given by squared distance of each vector from its centroid summed over all vectors. It shows how good the centroids represent their members of their clusters.

$$\mathrm{RSS}_k = \sum_{\vec{x} \in \omega_k} |\vec{x} - \vec{\mu}(\omega_k)|^2 \qquad \mathrm{RSS} = \sum_{k=1}^{K} \mathrm{RSS}_k$$

- RSS is a measure to decide cluster quality. For an empty cluster C_k, its RSS_k value will be 0 and with above equations. Recall that the aim is to minimize the RSS value to prove K-means convergence.
- The solution is to choose a replacement centroid, this can be done, for example, by choosing a data point which is farthest away from any current centroid or choosing the point that contributes most to RSS. If there are several empty clusters, the above can be repeated several times.

  When we replace the empty cluster with a new cluster based on the chosen centroid, some points will be assigned to that new cluster. They will be assigned to it because the new centroid will be closer to them than their current centroid. Therefore, the contribution of those points to RSS will decrease.

  This means that if we replace the empty cluster with a new non-empty one, the RSS will decrease. Therefore a clustering with an empty cluster cannot be optimal. Therefore we always replace the empty cluster.

II Recommender System :

	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

→ We have to treat ratings of 3, 4, 5 as 1 and 1, 2, blank as 0. We will use this notation to find updated utility matrix as shown below:

	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

→ Jaccard similarity is given by,

$$\text{Jaccard similarity} = \frac{f_{11} + f_{00}}{f_{10} + f_{01} + f_{00} + f_{11}}$$

Jaccard distance = 1 − Jaccard similarity

* For users - A, B :

$$\text{Jaccard similarity}(A, B) = \frac{3+2}{2+1+3+2} = \frac{5}{8} = 0.625$$

$$\text{Jaccard distance}(A, B) = 1 - 0.625 = \underline{0.375}$$

* For users - B, C :

$$\text{Jaccard similarity}(B, C) = \frac{2+1}{2+3+2+1} = \frac{3}{8} = 0.375$$

$$\text{Jaccard distance}(B, C) = 1 - 0.375 = \underline{0.625}$$

* For users - A, C:

$$\text{Jaccard similarity}(A,C) = \frac{2+2}{2+2+2+2} = 0.5$$

$$\text{Jaccard distance}(A,C) = 1 - 0.5 = 0.5$$

→ Cosine Distance:

* For users - A, B:

$$\cos(A,B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{2}{\sqrt{4} \times \sqrt{3}} = 0.577$$

* For users - B, C:

$$\cos(B,C) = \frac{B \cdot C}{\|B\| \cdot \|C\|} = \frac{1}{\sqrt{3} \times \sqrt{4}} = 0.289$$

* For users - A, C:

$$\cos(A,C) = \frac{A \cdot C}{\|A\| \cdot \|C\|} = \frac{2}{\sqrt{4} \times \sqrt{4}} = 0.5$$

* Conclusion:

→ Jaccard distance is better than cosine distance, as a similarity measure because recommendation system for movies are either based on user-user or item-item collaborative filtering.

→ Here, the blank ratings are considered as 0, we use it to count cosine similarity/distance, which can lead to wrong results than expected.

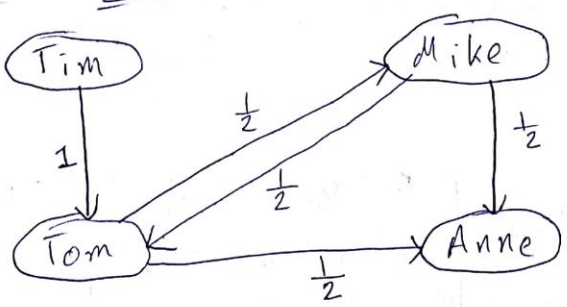
→ Jaccard similarity takes into account, as to whether the item was rated or not i.e. rating is present or not.

→ Jaccard similarity can handle binary values ② for any attribute. Hence, for users' ratings for movies recommendation, Jaccard similarity/distance is better.

III Page Rank

III Page Rank :

Directed graph:



→ Users are represented by nodes.

→ If a user A mentions about user B, then we draw an edge between them.

→ So, there are 4 users and 5 edges as per given tweets.

Transition Matrix:

	Tim	Mike	Tom	Anne
Tim	0	0	0	0
Mike	0	0	1/2	0
Tom	1	1/2	0	0
Anne	0	1/2	1/2	0

→ To make it stochastic, we need to have summation of each column equal to 1.

Updated transition matrix:

	Tim	Mike	Tom	Anne
Tim	0	0	0	1/4
Mike	0	0	1/2	1/4
Tom	1	1/2	0	1/4
Anne	0	1/2	1/2	1/4

→ For last column, we divide 1 to 4 users with 1/4 value.

→ Remaining all columns sum up to 1.

→ We use below equation recursively to compute the page rank.

$$A = \beta M v + (1-\beta)\frac{e}{n}$$

; A = Rank matrix
$1-\beta$ = Probability of teleporting = 0.1 (given)
n = No. of nodes = 4
e = Eigenvector matrix of $n * 1$ size

* Iteration-1:

$$A_1 = 0.1 * \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix} + 0.9 \times \begin{bmatrix} 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 1/2 & 1/4 \\ 1 & 1/2 & 0 & 1/4 \\ 0 & 1/2 & 1/2 & 1/4 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}$$

$$= \begin{bmatrix} 0.025 \\ 0.025 \\ 0.025 \\ 0.025 \end{bmatrix} + \begin{bmatrix} 9/160 \\ 27/160 \\ 63/160 \\ 45/160 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 0.0813 \\ 0.1938 \\ 0.4188 \\ 0.3063 \end{bmatrix}$$

* Iteration-2:

→ We use rank matrix calculated in iteration-1.

$$A = \beta M A_1 + (1-\beta)\frac{e}{n}$$

$$A_2 = 0.9 \times \begin{bmatrix} 0 & 0 & 0 & 1/4 \\ 0 & 0 & 1/2 & 1/4 \\ 1 & 1/2 & 0 & 1/4 \\ 0 & 1/2 & 1/2 & 1/4 \end{bmatrix} \times \begin{bmatrix} 0.0813 \\ 0.1938 \\ 0.4188 \\ 0.3063 \end{bmatrix} + 0.1 \times \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}$$

$$= \begin{bmatrix} 0.069 \\ 0.258 \\ 0.229 \\ 0.345 \end{bmatrix} + \begin{bmatrix} 0.025 \\ 0.025 \\ 0.025 \\ 0.025 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0.094 \\ 0.283 \\ 0.254 \\ 0.37 \end{bmatrix}$$

* Iteration-3:

→ We use rank matrix calculated in iteration-2.

$$A = \beta M A_2 + (1-\beta)\frac{e}{n}$$

$$A_3 = 0.9 \times \begin{bmatrix} 0 & 0 & 0 & 1/4 \\ 0 & 0 & 1/2 & 1/4 \\ 1 & 1/2 & 0 & 1/4 \\ 0 & 1/2 & 1/2 & 1/4 \end{bmatrix} \times \begin{bmatrix} 0.094 \\ 0.283 \\ 0.254 \\ 0.37 \end{bmatrix} + 0.1 \times \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0.083 \\ 0.197 \\ 0.295 \\ 0.324 \end{bmatrix} + \begin{bmatrix} 0.025 \\ 0.025 \\ 0.025 \\ 0.025 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0.108 \\ 0.222 \\ 0.320 \\ 0.349 \end{bmatrix}$$

→ So, after iteration-3, we have:

$$\begin{bmatrix} \text{Tim} \\ \text{Mike} \\ \text{Tom} \\ \text{Anne} \end{bmatrix} = \begin{bmatrix} 0.108 \\ 0.222 \\ 0.320 \\ 0.349 \end{bmatrix}$$

∴ Ranks can be given as Anne-1, Tom-2, Mike-3, Tim-4

∴ Importance of user is given as

Anne > Tom > Mike > Tim

→ We consider rank matrix after all iterations and sort values in descending order and ranks users.

→ Higher the value, more important is the user. It is considered as an important user as it is @mention by many other users. Here, Anne is @mention by more no. of times by other users, hence she is important user.

→ In Twitter, recommendation for a user to follow an important user works in this concept only, as the important user will have @mention by many times by other users.