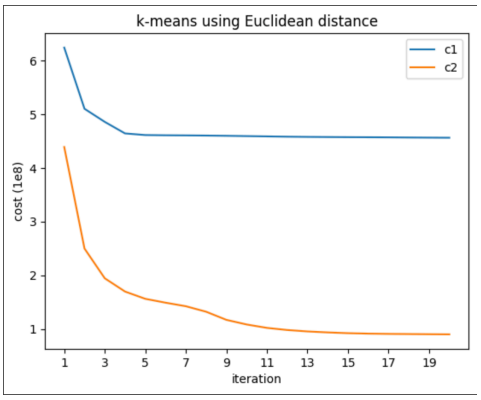
Assignment 4

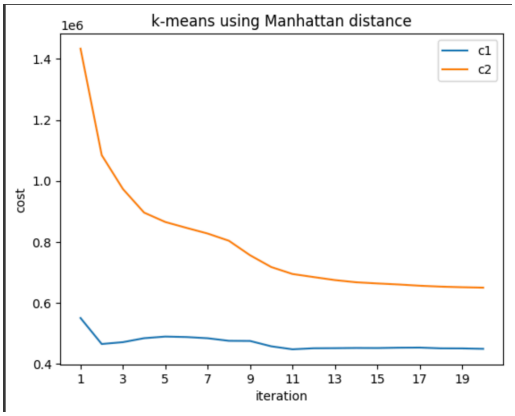
Q1

a)



The percentage change in cost after 10 iterations of c1.txt is: 26.48%. The percentage change in cost after 10 iterations of c2.txt is: 76.70%. c2 is better than c1 since the initial clusters are far apart and less overlap. Hence, the true clusters will be split less often, leading to better final clusters results. Also, K-Means algorithm minimized the Euclidean distance between data and their centroids, so the cost of c2 less than c1 means that c2 is better.

b)



The percentage change in cost after 10 iterations of c1.txt is: 18.65%. The percentage change in cost after 10 iterations of c2.txt is: 51.55%. c1 is better than c2 in terms of cost of Manhattan distance because c1 has lower final cost. Since the initial cluster centroids in c2 are as far as possible by using Euclidean distance, those centroids might not be the furthest in Manhattan distance and weren't necessarily far apart in Manhattan distance.

Q2

a) In the user-item bipartite graph, T_{ii} equals the degree of $user_i$. Since $T_{ii} = \sum_j^n R_{ij} \times (R^T)_{ji} = \sum_j^n R_{ij}^2 = \sum_j^n R_{ij}$. Since R_{ij} is 0 or 1, so $T_{ii} = degree(user_i)$, means the number of items that $user_i$ likes, also equals to the node degree of $user_i$.

T_{ij} $(i \neq j)$ is the number of paths between $user_i$ and $user_j$ with the length of 2, it also represents the number of items that they both like. $T_{ij} = \sum_k^n R_{ik} \times R_{jk}$, $R_{ik} \times R_{jk} (\neq 0)$ means there exists a 2-step path starts from $user_j$ to $user_k$ via $item_k$.

b) $We\ denote\ by\ R_i, the\ ith\ row\ of\ R, and\ by\ R_j, the\ jth\ column.$
$We\ know\ that\ the\ vector\ of\ an\ item\ is\ defined\ by\ one\ column\ is\ R.$
$Furthermore, the\ norm\ of\ say\ item_i\ is\ defined\ by\ \sqrt{\sum_k^m R_{ki}^2}.$

$The\ sum\ is\ in\ fact\ the\ number\ of\ users\ that\ like\ this\ item\ (because\ R_{ki}\ can\ either\ be\ 0\ or\ 1,$ $we\ have\ R_{ki} = R_{ki}^2). So\ the\ norm\ of\ the\ item\ is\ in\ fact\ \sqrt{Q_{ii}}.\ Thus\ we\ have:$

$$\cos - \text{sim}(item_i, item_j) = \frac{R_i \,.\, R_j}{\sqrt{Q_{ii}}\sqrt{Q_{jj}}} = \frac{\sum_k^m R_{ki} R_{kj}}{\sqrt{Q_{ii}}\sqrt{Q_{jj}}}$$

Now, the matrix Q is diagonal, and its diagonal coefficients are all non-zero (otherwise, some items are liked by nobody, and we might as well remove them, because they are useless and because the angle, they form with other items would be ill-defined), so we can denote by Q* the diagonal matrix whose diagonal coefficients are defined by $Q_{ii}^* = Q_{ii}^{-1/2}$.
We then have that:

$$\begin{aligned}(Q^* R^T R Q^*)_{ij} &= \sum_{k,l,m} Q_{ik}^* R_{kl} R_{lm} Q_{mj}^* \\ &= \sum_l Q_{ii}^* R_{li} R_{lm} Q_{mj}^* \\ &= \sum_l \frac{R_{li} R_{lj}}{\sqrt{Q_{ii}}\sqrt{Q_{jj}}} \\ &= \frac{R_i \,.\, R_j}{\sqrt{Q_{ii}}\sqrt{Q_{jj}}}\end{aligned}$$

So, the matrix S_I can be expressed in terms of Q and R:

$$S_u = Q^* R^T R Q^*$$

To compute a similar expression for S_u, we notice that (R, Q, S_I) and (R^T, P, S_u) play similar roles. Indeed, the relation "$user_u$ likes $item_i$" can be put backward into "$item_i$ is liked by $user_u$" which is equivalent to switching users and items, ie to transpose the matrix R. Thus, S_u is given by:

$$S_u = P^* R R^T P^*$$

With P^* being a diagonal matrix whose coefficients are defined $P^*_{ii} = P_{ii}^{-1/2}$.

c) For the user-user collaborative filtering recommendation, we have that:

$$\Gamma_{ij} = r_{ij}$$

$$= \sum_{x \in users} cos - sim(x, i) \times R_{x,j}$$

$$= \sum_{x \in users} cos - sim(i, x) \times R_{x,j}$$

$$= \sum_{x=1}^{m} (S_u)_{i,x} \times R(x, j)$$

$$= (S_u R)_{ij}$$

$$= (P^* R R^T P^* R)_{ij}$$

Thus,

$$\Gamma = P^* R R^T P^* R$$

Similarly, for the item-item collaborative filtering recommendation, we have that:

$$\Gamma_{ij} = r_{ij}$$

$$= \sum_{x \in items} cos - sim(x, j) \times R_{i,x}$$

$$= \sum_{x=1}^{n} R_{i,x} \times (S_I)_{x,j}$$

$$= (R S_I)_{ij}$$

$$= (R Q^* R^T R Q^*)_{ij}$$

Thus,

$$\Gamma = R Q^* R^T R Q^*$$

d)

The names of five TV shows that have the highest similarity scores for Alex for the user-user collaborative filtering are:

1. FOX 28 News at 10pm
2. Family Guy
3. 2009 NCAA Basketball Tournament
4. NBC 4 at Eleven
5. Two and a Half Men

The names of five TV shows that have the highest similarity scores for Alex for the movie-movie collaborative filtering are:

1. FOX 28 News at 10pm
2. Family Guy
3. NBC 4 at Eleven
4. 2009 NCAA Basketball Tournament
5. Access Hollywood