Let us define a real matrix M (of size $p \times q$) and let us assume this matrix corresponds to a dataset with p data points and q dimensions.
We now look at the matrices MM^T and M^TM and explore if they are symmetric, square and real, which all follow from the basics of linear algebra.

- Symmetric. Both matrices are symmetric.
  Proof. A matrix A is symmetric if $A = A^T$, so is is sufficient to show that $(MM^T)^T = MM^T$ and $(M^TM)^T = M^TM$.

  $$\begin{aligned}(MM^T)^T &= (M^T)^T M^T = MM^T \\ (M^TM)^T &= M^T(M^T)^T = M^TM\end{aligned}$$

- Square. Both matrices are square.
  Proof. We know that matrix M is of size $p \times q$, so M^T is of size $q \times p$. Their multiplication schemes would look like this:

  $$MM^T = \quad p\left\{\overbrace{\begin{bmatrix} \quad & \quad \\ \quad & \quad \\ \quad & \quad \end{bmatrix}}^{q}\right. \cdot \quad q\left\{\overbrace{\begin{bmatrix} \quad & \quad \\ \quad & \quad \\ \quad & \quad \end{bmatrix}}^{p}\right. = \quad p\left\{\overbrace{\begin{bmatrix} \quad & \quad \\ \quad & \quad \\ \quad & \quad \end{bmatrix}}^{p}\right.$$

  $$M^TM = \quad q\left\{\overbrace{\begin{bmatrix} \quad & \quad \\ \quad & \quad \\ \quad & \quad \end{bmatrix}}^{p}\right. \cdot \quad p\left\{\overbrace{\begin{bmatrix} \quad & \quad \\ \quad & \quad \\ \quad & \quad \end{bmatrix}}^{q}\right. = \quad q\left\{\overbrace{\begin{bmatrix} \quad & \quad \\ \quad & \quad \\ \quad & \quad \end{bmatrix}}^{q}\right.$$

  So both matrices are square, MM^T is of dimensions $p \times p$ and M^TM of dimensions $q \times q$. Also, a matrix must be square in order to be symmetric.

- Real. Both matrices are real.
  Proof. Because M is real, also M^T is real. So both products MM^T and M^TM are real, since they are multiplied by two real matrices.

Now we are observing nonzero eigenvalues of MM^T and of M^TM.
Let $M = U\Sigma V^T$, where U (of size $p \times k$) and V (of size $q \times k$) are column-orthonormal matrices and Σ (of size $k \times k$) is a diagonal matrix, all given by SVD.
It follows

$$MM^T = (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)((V^T)^T\Sigma^T U^T) = (U\Sigma V^T)(V\Sigma^T U^T) = U\Sigma^2 U^T$$
$$M^TM = (U\Sigma V^T)^T(U\Sigma V^T) = ((V^T)^T\Sigma^T U^T)(U\Sigma V^T) = (V\Sigma^T U^T)(U\Sigma V^T) = V\Sigma^2 V^T$$

We see that in both cases the nonzero eigenvalues are the diagonal entries of Σ^2.
Let x be the eigenvector of MM^T and λ the corresponding eigenvalue, therefore

$$\begin{aligned} MM^Tx &= \lambda x \\ M^TMM^Tx &= M^T\lambda x \\ M^TMM^Tx &= \lambda M^Tx \\ M^TM(M^Tx) &= \lambda(M^Tx) \end{aligned}$$

It follows that the corresponding eigenvector for eigenvalue λ of M^TM is M^Tx.
Therefore, their eigenvectors are not the same.

Let Q be an ortogonal matrix containing the eigenvector as its columns and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_d)$ containing the eigenvalues of $M^T M$.
Since $M^T M$ is symmetric, square and real (by 1(a)), it can be written as

$$M^T M = Q \Lambda Q^T$$

Let $M = U\Sigma V^T$, where U and V are columnorthonormal and Σ is a diagonal matrix, given by SVD.
Then $M^T = (V^T)^T\Sigma U^T = V\Sigma U^T$ and

$$M^TM = V\Sigma U^TU\Sigma V^T = V\Sigma^2V^T$$

The code is available in the attached file `HW2_q1.py`.

- SVD of M:

$$U = \begin{bmatrix} -0.27854301 & 0.5 \\ -0.27854301 & -0.5 \\ -0.64993368 & 0.5 \\ -0.64993368 & -0.5 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 7.61577311 & 0 \\ 0 & 1.41421356 \end{bmatrix}$$

$$V^T = \begin{bmatrix} -0.70710678 & -0.70710678 \\ -0.70710678 & 0.70710678 \end{bmatrix}$$

- Eigenvalue decomposition of M^TM:

$$Evals = \begin{bmatrix} 58.0 & 2.0 \end{bmatrix}$$

$$Evecs = \begin{bmatrix} 0.70710678 & -0.70710678 \\ 0.70710678 & 0.70710678 \end{bmatrix}$$

- Correspondence between V produced by SVD and the matrix $Evecs$:
  Some observations:
  - Matrix V is in this case the same as V^T.
  - We notice that V and $Evecs$ have the same values with a different sign (+/–) in the first column. It seems that they might be equivalent and to show that we must find the matrix Q^{-1} and P such that $V = Q^{-1}\ Evecs\ P$. This is true for

    $$P = I \text{ and } Q^{-1} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

    Matrices V and $Evecs$ are equivalent.
  - Both matrices are orthogonal, therefore the conditions

    $$VV^T = V^TV = I \text{ and } (Evecs)(Evecs)^T = (Evecs)^T(Evecs) = I$$

    are true, also applying that $V^T = V^{-1}$ and $Evecs^T = Evecs^{-1}$.

- Relationship between the $Evals$ and the singular values of M
  The singular value of M are exactly the square roots of the eigenvalues of M^TM.

Implementation for both parts of this question is available in the attached file `HW2_q2.html`.

Let $\mathcal{X}$ be a set of n data points in the d-dimensional space $\mathbb{R}^d$. Given the number of clusters k and the set of k centroids $\mathcal{C}$, we now proceed to define various distance metrics and the corresponding cost functions that they minimize.

1. Using the Euclidean distance as the distance measure, we compute the cost function for every iteration i.

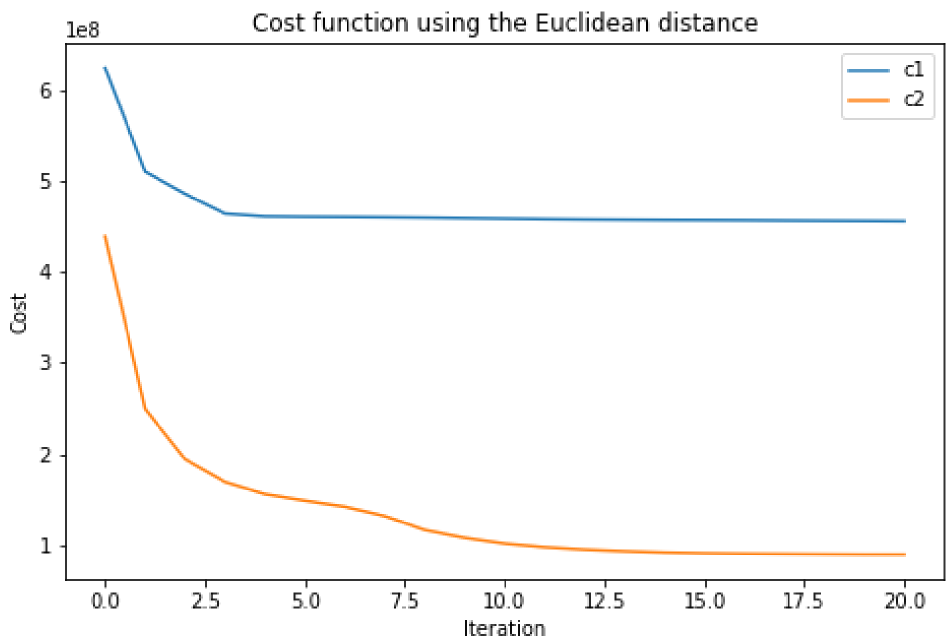

Figure 1: Graph of cost function using the Euclidean distance on initial centroids from files `c1.txt` and `c2.txt`.

2. Percentage change after 10 iterations of the K-Means algorithm:

   - `c1.txt` improves by cca 26.5 % (0.2648391714456047)
   - `c2.txt` improves by cca 76.7 % (0.7669795594605944)

   In this case, `c2.txt` (the lower graph on the figure) is much better, because cluster centroids are as far apart as possible using Euclidean distance. In this case, true clusters are split less often, which leads to a better set of clusters.

1. Using the Manhattan distance as the distance measure, we compute the cost function for every iteration i.

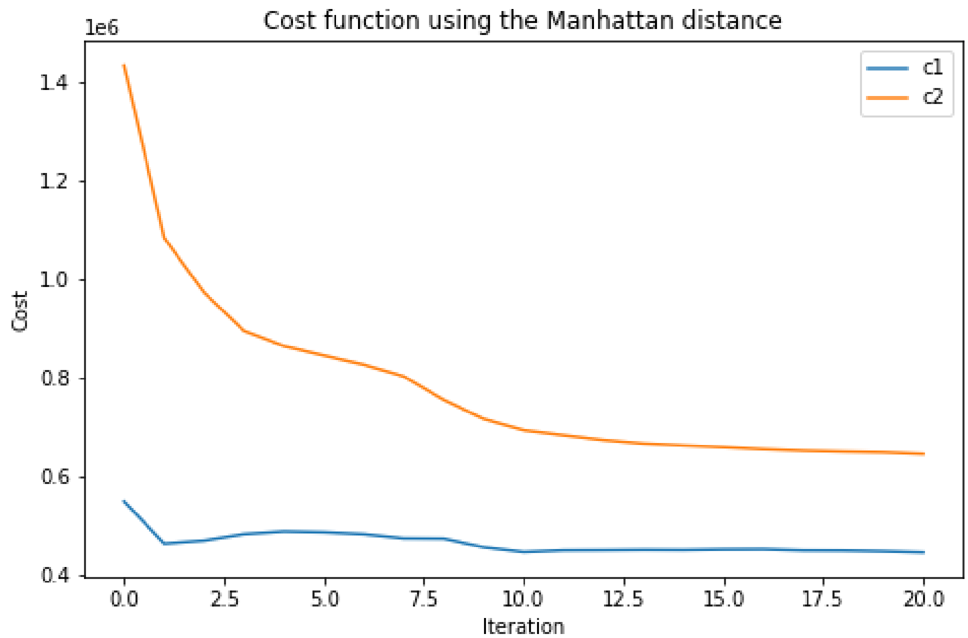

Figure 2: Graph of cost function using the Manhattan distance on initial centroids from files `c1.txt` and `c2.txt`.

2. Percentage change after 10 iterations of the K-Means algorithm:

   - `c1.txt` improves by cca 18.7 % (0.18654926116799223)
   - `c2.txt` improves by cca 51.6 % (0.5155409910238908)

   In this case, `c1.txt` (the lower graph on the picture) is much better, even if the centroids are chosen randomly. That is because points in `c2.txt` are as far apart as possible using Euclidean distance, so it is not usually the case with Manhattan distance. Therefore, the points in `c2.txt` are not necessarily very far apart in Manhattan distance.

Let ϵ_{iu} denote the derivative of the error E with the respect R_{iu}. Then the expression for ϵ_{iu} (for each R_{iu}) is

$$\epsilon_{iu} = 2 \cdot \left(R_{iu} - q_i \cdot p_u^T\right)$$

where we update the equation as

$$q_i \leftarrow q_i + \eta \cdot (\epsilon_{iu} \cdot p_u - 2 \cdot \lambda \cdot q_i)$$
$$p_u \leftarrow p_u + \eta \cdot (\epsilon_{iu} \cdot q_i - 2 \cdot \lambda \cdot p_u)$$

and η is the the learning rate.

Implementation of Stohastic Gradient Descent algorithm is available in the attached file `HW2_q2.html`.

In the algorithm, the value for learning rate μ was set to $\mu = 0.03$.

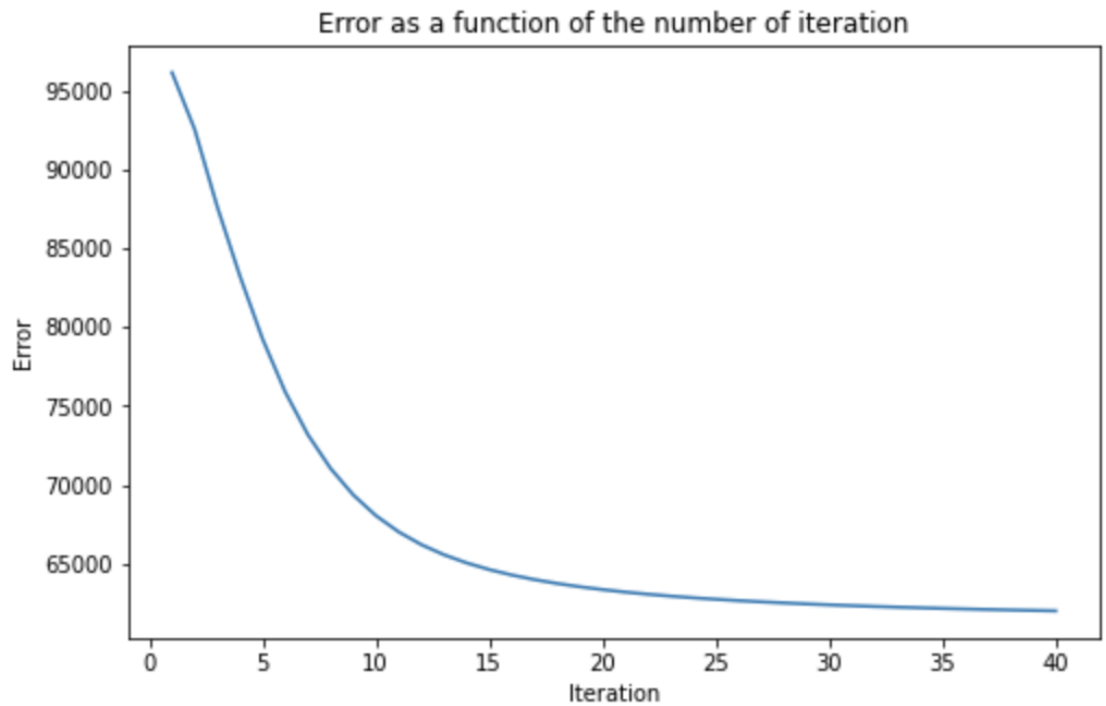

Figure 3: Plot of the function E as the function of the number of iterations with value $\mu = 0.03$.

Let $T = RR^T$. Now we observe the T_{ii} and T_{ij} for $i \neq j$.

$$T_{ii} = \sum_{j=1}^{n} R_{ij} \cdot (R^T)_{ji} = \sum_{j=1}^{n} R_{ij}^2 = \sum_{j=1}^{n} R_{ij}$$

since R_{ij} is either 0 or 1. Thus follows that T_{ii} is equal to the number of items j that user i likes, so it is equal to $degree(user_i)$.

$$T_{ij} = \sum_{k=1}^{n} R_{ik} \cdot (R^T)_{kj} = \sum_{k=1}^{n} R_{ik} \cdot R_{jk}$$

Since R_{ik} and R_{jk} are either 1 or 0, also their product is 1 or 0. Therefore, $R_{ik} \cdot R_{jk} = 1$ iff. both users i and j likes item k. Thus follows that T_{ij} for $i \neq j$ is the number of items that both $user_1$ and $user_2$ likes, so it is equal to the number of paths of length 2 between $user_1$ and $user_2$ (via the common item).

Item similarity matrix S_I. Let us define the item similarity matrix S_I $(n \times n)$, such that the element in row i and column j is the cosine similarity of $item_i$ and $item_j$, which corresponds to column i and column j of the matrix R, denoted as c_i and c_j, so

$$(S_I)_{ij} = \text{cos-sim}(c_i, c_j) = \frac{c_i \cdot c_j}{||c_i|| \cdot ||c_j||}$$

Since we want to prove that $S_I = Q^{-1/2} R^T R Q^{-1/2}$, note that

$$(R^T R)_{ij} = c_i c_j$$

and

$$(Q^{-1/2})_{ii} = \frac{1}{\sqrt{c_i c_i}} = \frac{1}{||c_i||}$$
$$(Q^{-1/2})_{jj} = \frac{1}{\sqrt{c_j c_j}} = \frac{1}{||c_j||}$$

Thus we have, considering Q is diagonal,

$$\begin{aligned}(Q^{-1/2} R^T R Q^{-1/2})_{ij} &= \sum_{k,l,m} Q_{ik}^{-1/2} R_{kl}^T R_{lm} Q_{mj}^{-1/2} \\ &= \sum_{l} Q_{ii}^{-1/2} R_{li} R_{lj} Q_{jj}^{-1/2} \\ &= \frac{c_i \cdot c_j}{||c_i|| \cdot ||c_j||}\end{aligned}$$

which is exactly the same as $(S_I)_{ij}$. So, clearly, it follows

$$S_I = Q^{-1/2} R^T R Q^{-1/2}$$

User similarity matrix S_U. Let us define user similarity matrix S_U where the element in row i and column j is the cosine similarity of $user_i$ and $user_j$, which correspond to row i and row j of the matrix R, denoted as r_i and r_j, so

$$(S_U)_{ij} = \text{cos-sim}(r_i, r_j) = \frac{r_i \cdot r_j}{||r_i|| \cdot ||r_j||}$$

In this case, the matrix P has the same role as Q and R the same as R^T, since the relation ($user_u$ likes $item_i$) is now translated into ($item_i$ is liked by $user_u$), which is exactly the same as transposing the matrix R and taking the matrix P instead of Q. Following the previous case, note that

$$(RR^T)_{ij} = r_i r_j$$

and

$$(P^{-1/2})_{ii} = \frac{1}{\sqrt{r_i r_i}} = \frac{1}{||r_i||}$$
$$(P^{-1/2})_{jj} = \frac{1}{\sqrt{r_j r_j}} = \frac{1}{||r_j||}$$

Thus we have, considering P is diagonal,

$$
\begin{aligned}
(P^{-1/2}RR^TQ^{-1/2})_{ij} &= \sum_{k,l,m} P_{ik}^{-1/2} R_{kl} R_{lm}^T P_{mj}^{-1/2} \\
&= \sum_{l} P_{ii}^{-1/2} R_{li} R_{lj} P_{jj}^{-1/2} \\
&= \frac{r_i \cdot r_j}{||r_i|| \cdot ||r_j||}
\end{aligned}
$$

which is exactly the same as $(S_U)_{ij}$. So clearly it follows

$$S_U = P^{-1/2}RR^TP^{-1/2}$$

The recommendation method using user-user collaborative filterinh for *user* u, can be described as follows: for all *items* s, compute r_{us} as

$$r_{us} = \sum_{x \in users} \text{cos-sim}(x, u) \cdot R_{xs}$$

or

$$r_{us} = \sum_{x \in items} R_{ux} \cdot \text{cos-sim}(x, s)$$

and recommend the k items for which r_{us} is the largest.
Let the recommendation matrix be Γ $(m \times n)$, s. t. $\Gamma(i, j) = r_{ij}$.
We want to express Γ for both item-item and user-user collaborative filtering approaches in term of R, P and Q.

Item-item.
Let $r_{us} := \sum_{x \in users} R_{ux} \cdot \text{cos-sim}(x, s)$. We know that $(S_I)_{us} = \text{cos-sim}(c_u, c_s)$, where c_u denotes u-th and c_s denotes s-th item. We thus have

$$(RS_I)_{us} = \sum_{x \in items} R_{ux} \cdot (S_I)_{xs} = \sum_{x \in items} R_{ux} \cdot \text{cos-sim}(x, s) = r_{us}$$

following that

$$\Gamma = RS_I = RQ^{-1/2} R^T R Q^{-1/2}$$

User-user.
Let $r_{us} = \sum_{x \in users} \text{cos-sim}(x, u) \cdot R_{xs}$. We know that $(S_U)_{us} = \text{cos-sim}(r_u, r_s)$, where r_u denotes u-th and r_s denotes s-th user. We thus have

$$(S_U R)_{us} = \sum_{x \in users} (S_U)_{us} \cdot R_{xs} = \sum_{x \in users} \text{cos-sim}(x, u) \cdot R_{xs} = r_{us}$$

following that

$$\Gamma = S_U R = P^{-1/2} R R^T P^{-1/2} R$$

Implementation is available in the attached file `HW2_q4.html`.

User-user.
Names of five TV shows that have the highest similarity scores for Alex for the user-user collaborative filtering.

- `FOX 28 News at 10pm`
- `Family Guy`
- `2009 NCAA Basketball Tournament`
- `NBC 4 at Eleven`
- `Two and a Half Men`

Item-item.
Names of five TV shows that have the highest similarity scores for Alex for the item-item collaborative filtering.

- `FOX 28 News at 10pm`
- `Family Guy`
- `NBC 4 at Eleven`
- `2009 NCAA Basketball Tournament`
- `Access Hollywood`

Information sheet
CS246: Mining Massive Data Sets

Assignment Submission Fill in and include this information sheet with each of your assignments. This page should be the last page of your submission. Assignments are due at 11:59pm and are always due on a Thursday. All students (SCPD and non-SCPD) must submit their homework via Gradescope (http://www.gradescope.com). Students can typeset or scan their homework. Make sure that you answer each (sub-)question on a separate page. That is, one answer per page regardless of the answer length. Students also need to upload their code on Gradescope. Put all the code for a single question into a single file and upload it.

Late Homework Policy Each student will have a total of *two* late periods. *Homework are due on Thursdays at 11:59pm PT and one late period expires on the following Monday at 11:59pm PT*. Only one late period may be used for an assignment. Any homework received after 11:59pm PT on the Monday following the homework due date will receive no credit. Once these late periods are exhausted, any assignments turned in late will receive no credit.

Honor Code We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down their solutions independently, i.e., each student must understand the solution well enough in order to reconstruct it by him/herself. Students should clearly mention the names of all the other students who were part of their discussion group. Using code or solutions obtained from the web (GitHub/Google/previous year's solutions etc.) is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code very seriously and expect students to do the same.

Your name: Sara Bizjak

Email: sarabizjak97@gmail.com **SUID:** 27202020

Discussion Group: Petra Podlogar

I acknowledge and accept the Honor Code.

(Signed) ____________________