CS246: Mining Massive Datasets
Homework 2

Answer to Question 1(a)

Symmetric
Let $A = MM^T$.
By the definition of the transpose of a matrix, for all entries (i, j),

$$M_{ij} = M^T_{ji} \tag{1}$$

We also know that

$$A_{ij} = \sum_{k=1}^{q} M_{ik} * M^T_{kj} \tag{2}$$

Similarly,

$$A_{ji} = \sum_{k=1}^{q} M_{jk} * M^T_{ki} \tag{3}$$

By (1), we know that $M_{ik} = M^T_{ki}$ and $M_{jk} = M^T_{kj}$. Thus, we can substitute into (2) and (3), giving us the following

$$A_{ij} = \sum_{k=1}^{q} M^T_{ki} * M^T_{kj} \tag{4}$$

$$A_{ji} = \sum_{k=1}^{q} M^T_{kj} * M^T_{ki} \tag{5}$$

By the definition of matrix symmetry, matrix A is symmetric if $A_{ij} = A_{ji}$ for all entries (i, j) in A, which we've just proven by demonstrating that (4) and (5) are equal. Thus, MM^T is symmetric. Without loss of generality, $M^T M$ is also symmetric (we can perform the same proof, just adjusting the bounds on the summations and come to the same conclusion). ■

Square
M is given as a matrix of size $p \times q$, and therefore M^T has size $q \times p$. MM^T is the result of multiplying a matrix of size $p \times q$ and a matrix of size $q \times p$, which results in a matrix of size $p \times p$. Likewise, MM^T is the result of multiplying a matrix of size $q \times p$ and a matrix of size $p \times q$, which results in a matrix of size $q \times q$. Thus, both MM^T and $M^T M$ are square matrices. ■

Real
By definition, M is a real matrix. Each entry in MM^T and $M^T M$ is the sum of products of real numbers, which means they're real numbers. Thus, MM^T and $M^T M$ are both real matrices. ■

Answer to Question 1(b)

Proof

Let x be a nonzero eigenvalue of MM^T and let v be the corresponding eigenvector. Then we have

$$MM^Tv = xv \tag{1}$$
$$M^TMM^Tv = M^Txv \tag{2}$$
$$(M^TM)M^Tv = xM^Tv \tag{3}$$
$$M^TMw = xw \tag{4}$$

(1) is given by the definition of eigenvalues. In (2), we introduce matrix multiplication of M^T from the left on both sides of the equation. Then, in (4), we substitute $w = M^Tv$, and we see that x is an eigenvalue of M^TM, paired with a w, its eigenvector. Thus, the nonzero eigenvalues of MM^T are the same as the nonzero eigenvalues of M^TM, but their eigenvectors are not the same. ■

Answer to Question 1(c)

M^TM is a real, symmetric, and square matrix as proven above. Thus the eigenvalue decomposition of M^TM is

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

where $\Lambda = \text{diag}(\lambda_1, ..., \lambda_q)$ contains the eigenvalues of M^TM along its main diagonal and Q is an orthogonal matrix containing the eigenvectors of M^TM as its columns. ■

Answer to Question 1(d)

Let $M = U\Sigma V^T$, where U and V are columnal-orthonormal and Σ is a diagonal matrix. Then,

$$M^T = (U\Sigma V^T)^T \tag{1}$$
$$M^T = (V^T)^T \Sigma^T U^T \tag{2}$$
$$M^T = V\Sigma U^T \tag{3}$$

In (2), the order of the product reverses because of the rule for the transpose of a product. Then, in (3), we simplify $(V^T)^T = V$, because the transpose of the transpose is the original matrix, and $\Sigma^T = \Sigma$ since Σ is a diagonal matrix. Then, we can multiply $M^T M$ as follows,

$$M^T M = (V\Sigma U^T)(U\Sigma V^T) \tag{4}$$
$$M^T M = V\Sigma^2 V^T \tag{5}$$

In (5), $U^T U$ reduces to I since U is orthonormal, so we are left with $M^T M = V\Sigma^2 V^T$. ■

Answer to Question 1(e)

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

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

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

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

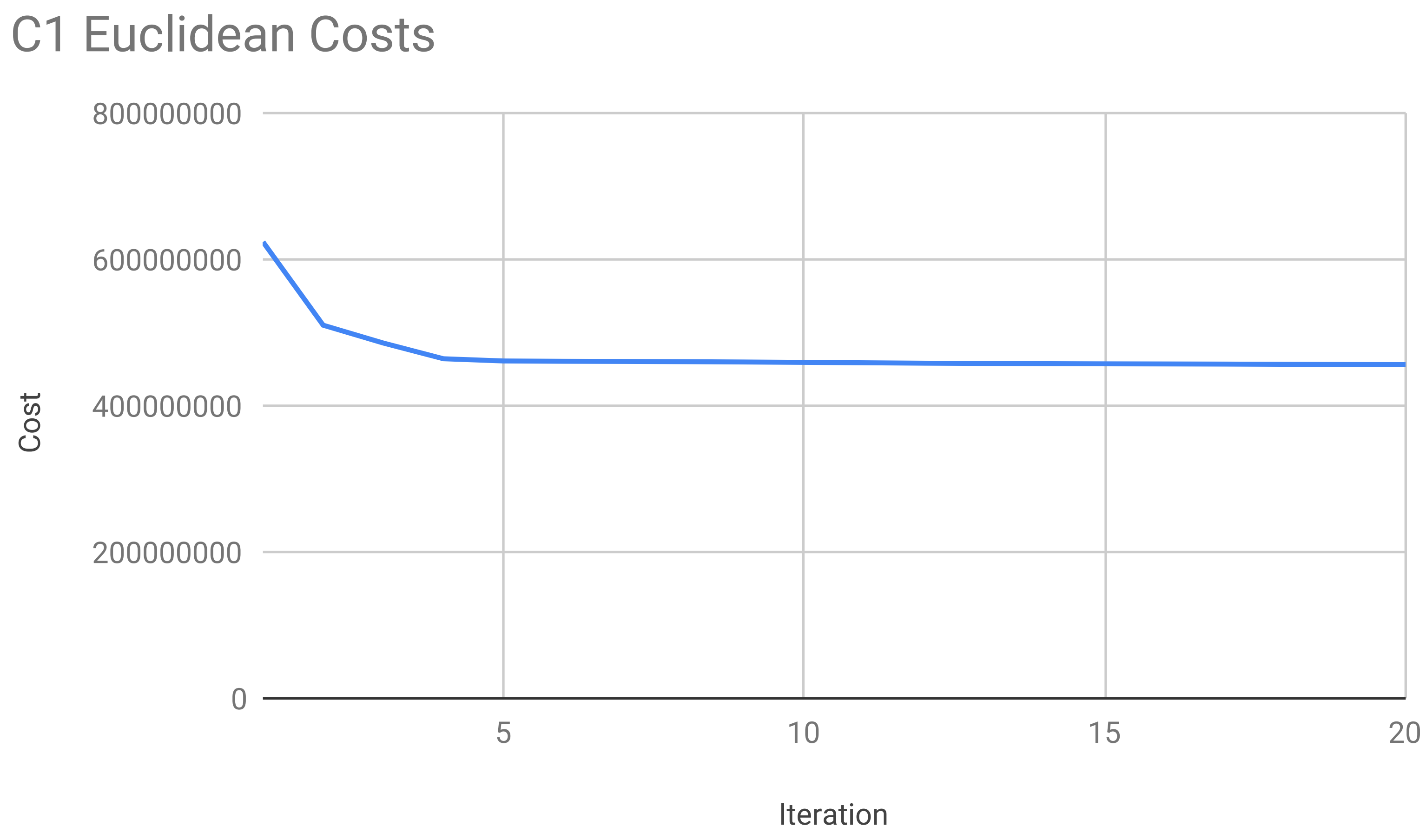
$$Evecs = \begin{bmatrix} 0.70710678 & -0.70710678 \\ 0.70710678 & 0.70710678 \end{bmatrix}$$

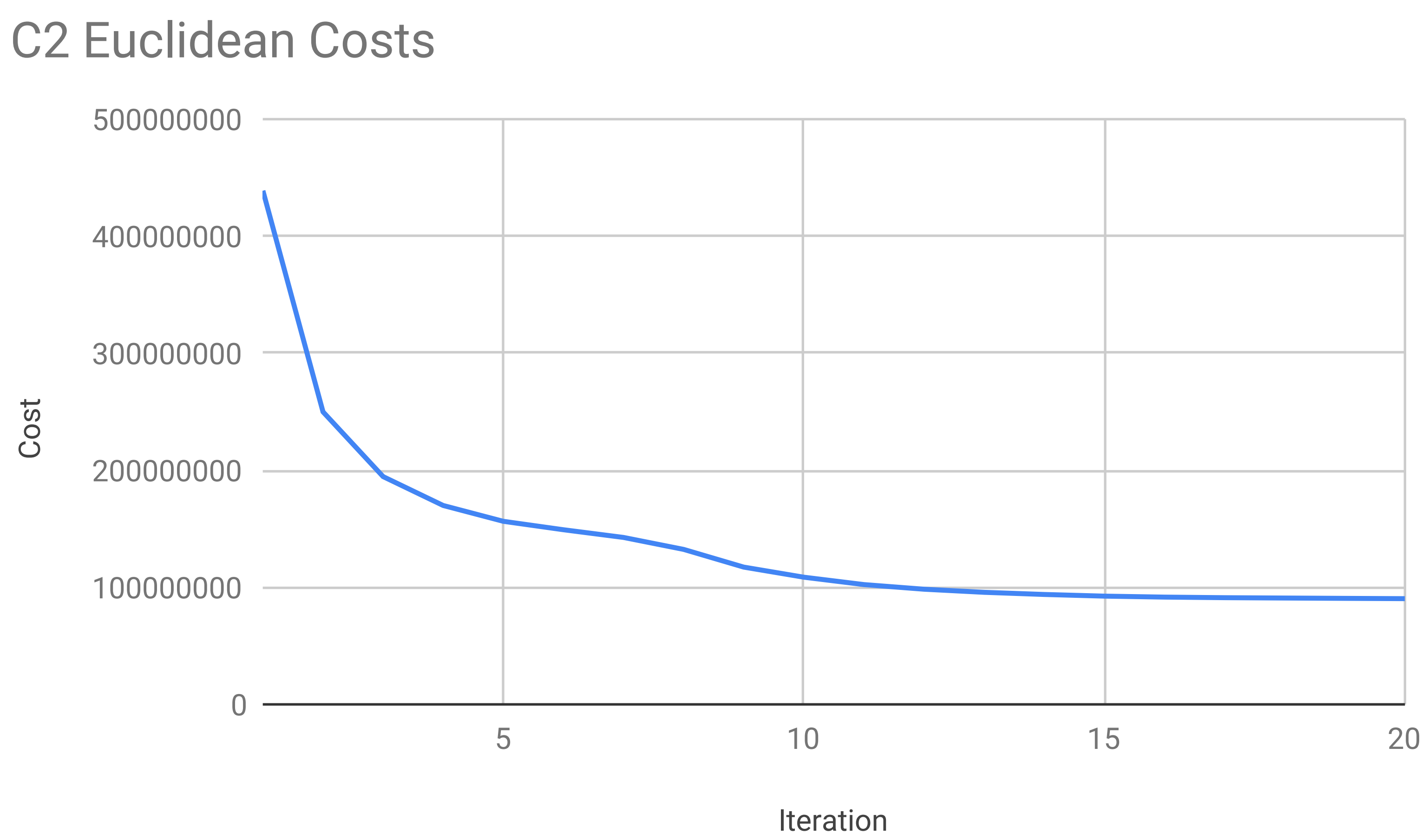
Answer to Question 2(a)

c1.txt + Euclidean Distance Percentage Change = -26.40%
c1.txt + Euclidean Distance decreases from 6.2E8 to 4.5E8 after 20 iterations.

c2.txt + Euclidean Distance Percentage Change = -75.26%
c2.txt + Euclidean distance decreases from 4.3E8 to 9.0E7 after 20 iterations.

Initialization using c2.txt has a greater percentage decrease after 10 iterations, and ends at a lower cost value than random initialization with c1.txt. Since we initialize the centroids as the farthest points from each other, it makes it more likely for distinct clusters to form that are as far from each other as possible, making it less expensive to assign points to them since it's like that for a given point p there will be a centroid that's close to p. With random initialization, it takes longer for us to approach centroids that have that property, giving us a slower percentage decrease and a higher cost at the end of the iterations.


C1 Euclidean Costs
800000000
600000000
400000000
200000000
0
Cost
5
10
15
20
Iteration

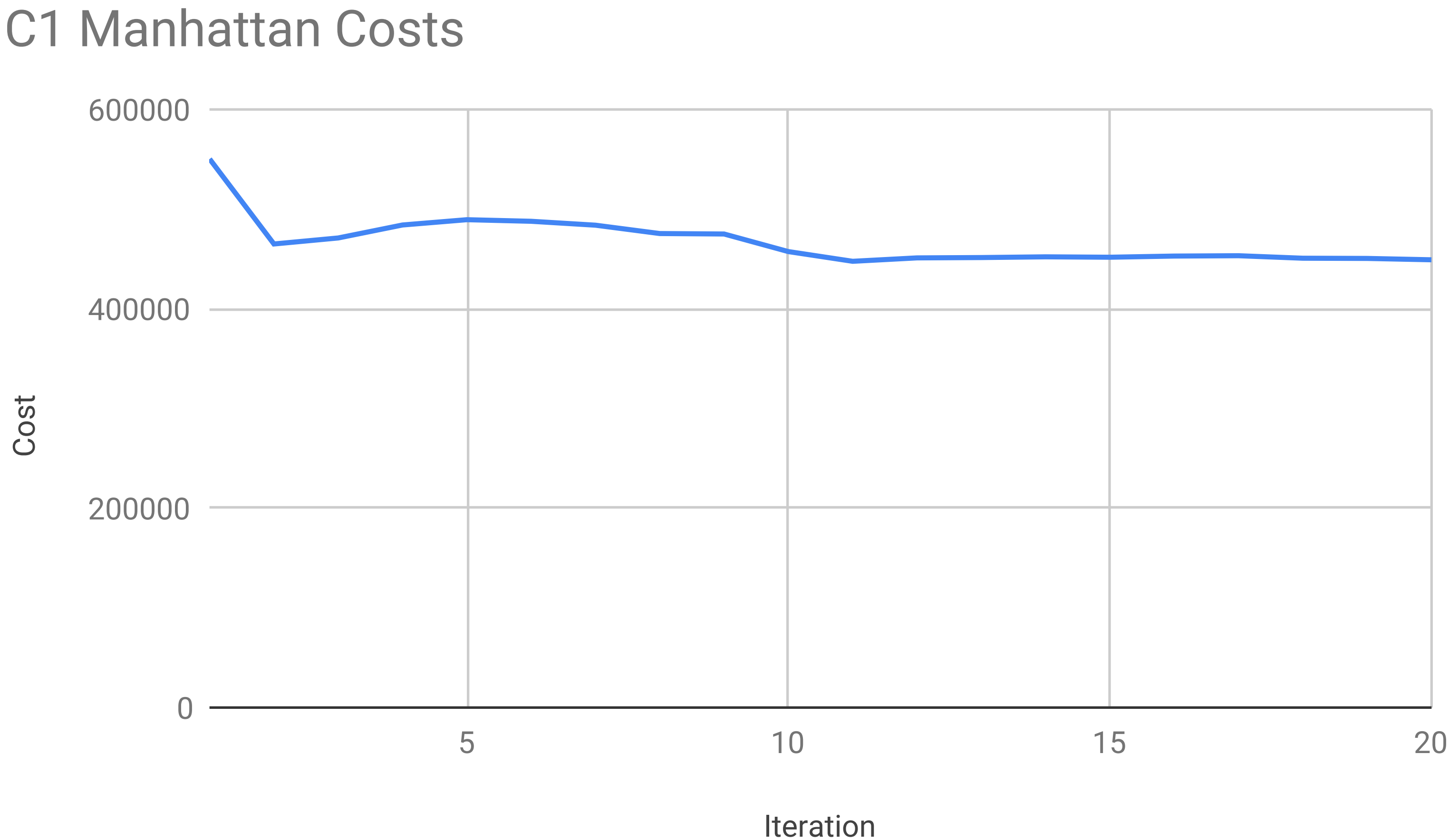

C2 Euclidean Costs
500000000
400000000
300000000
200000000
100000000
0
Cost
5
10
15
20
Iteration

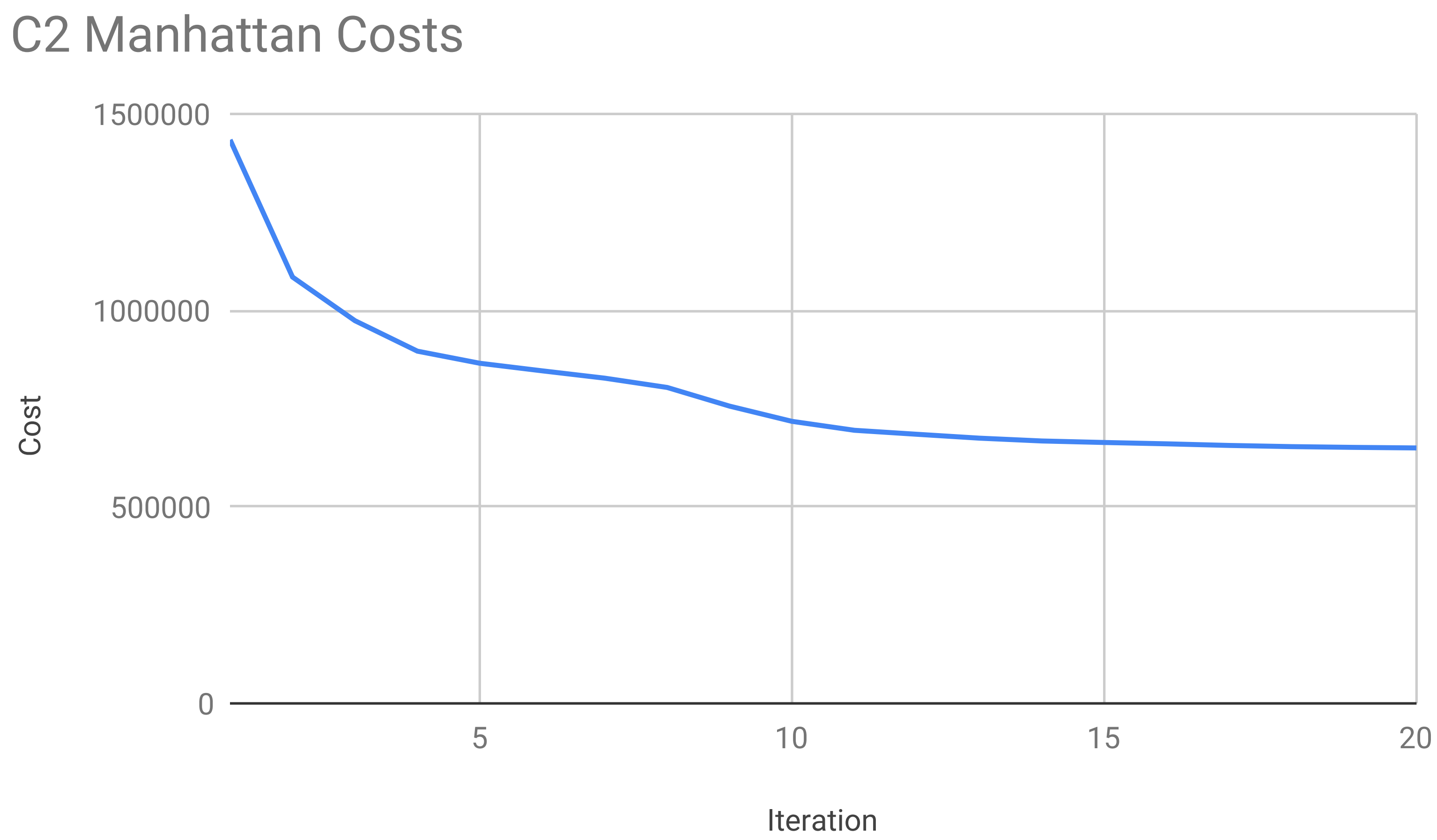
Answer to Question 2(b)

c1.txt + Manhattan Distance Percentage Change = -16.88%
c1.txt + Manhattan Distance decreases from 5.5E5 to 4.4E5 after 20 iterations.

c2.txt + Manhattan Distance Percentage Change = -49.97%
c2.txt + Manhattan Distance decreases from 1.4E6 to 6.4E5 after 20 iterations.

Random initialization works better than initialization with c2.txt in the Manhattan distance case; although its percentage decrease is smaller than that of c2.txt initialization, random initialization achieves a lower cost after 20 iterations. Since we recompute centroids using the mean of the cluster, the cost using c2.txt initialization doesn't end up as low as the cost achieved using random initialization.


C1 Manhattan Costs
600000
400000
200000
0
Cost
5
10
15
20
Iteration



C2 Manhattan Costs
1500000
1000000
500000
0
Cost
5
10
15
20
Iteration

Answer to Question 3(a)

Let $E = \sum_{(i,u)\in \text{ratings}}(R_{iu} - q_i \cdot p_u^T)^2 + \lambda(\sum_u ||p_u||_2^2 + \sum_i ||q_i||_2^2)$, where p_u^T is the uth column vector of P^T.
For each R_{iu} we take the derivative of E with respect to R_{iu},

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

We find the update equations, q_i and p_u by taking the derivative of E with respect to q_i and p_u, respectively, and multiplying by the learning rate η:

$$\begin{aligned}
q_i &\leftarrow q_i - \eta[-2(R_{iu} - q_i \cdot p_u^T)p_u + 2\lambda q_i] \\
q_i &\leftarrow q_i - \eta(-\epsilon_{iu} p_u^T + 2\lambda q_i) \\
q_i &\leftarrow q_i + \eta(\epsilon_{iu} p_u^T - 2\lambda q_i) \\
p_u &\leftarrow p_u - \eta[-2(R_{iu} - q_i \cdot p_u^T)q_i + 2\lambda p_u^T] \\
p_u &\leftarrow p_u - \eta(-\epsilon_{iu} q_i + 2\lambda p_u^T) \\
p_u &\leftarrow p_u + \eta(\epsilon_{iu} q_i - 2\lambda p_u^T)
\end{aligned}$$

So we end up with

$$\begin{aligned}
q_i &\leftarrow q_i + \eta(\epsilon_{iu} p_u^T - 2\lambda q_i) \\
p_u &\leftarrow p_u + \eta(\epsilon_{iu} q_i - 2\lambda p_u^T)
\end{aligned}$$

where η represents the learning rate, and λ represents the regularization constant.

Answer to Question 3(b)

Using a value of $\eta = 0.01$, we achieve $E = 50998.8700418$ after 40 iterations of training.

Answer to Question 4(a)

Define the non-normalized user similarity matrix $T = RR^T$.

T_{ii} signifies the degree of user node i:

$$T_{ii} = R_i \cdot R_i^T$$
$$T_{ii} = R_i \cdot R_i$$

where R_i^T is the i-th column vector of R^T and R_i is the i-th row vector of R, so $R_i^T = R_i$. In (2) we take the dot product of a i-th row vector of R with itself, which just equates to the sum of the elements in R_i, since all the elements in R are just 0's or 1's. This conceptually is the degree of the i-th user node, or how many items they like.

T_{ij}, where $i \neq j$ signifies how many likes users i and j have in common, or how many edges to the same item node they share (how many item nodes are reachable by both user node i and user node j):

$$T_{ij} = R_i \cdot R_j^T$$

where R_i is the i-th row vector of R, and R_j^T is the j-th column vector of R^T. This is equivalent to taking the dot product of the vectors that represent users i and j. Since all the elements in R are 0's or 1's, we are essentially just taking the sum of the elements in common between i and j. Conceptually, this is the number of items that both i and j like.

Answer to Question 4(b)

Let R be an $m \times n$ matrix, representing m users and their preferences over n items. R^T, therefore, is a $n \times m$ matrix, where rows represent items, and columns are users. An entry R^T_{ij} of 0 means that item i is not liked by user j, while a value of 1 means that user j does like item i.

Let Q be a $n \times n$ diagonal matrix, where the i-th diagonal element is the degree of item node i. Thus, we can think of the i-th diagonal element as the sum of the elements of the i-th column vector in R, or equivalently the sum of the elements of the i-th row vector in R^T. Then, $Q^{-1/2}$ is the $n \times n$ diagonal matrix, where element $Q_{ii}^{-1/2} = \frac{1}{\sqrt{Q_{ii}}}$. Since all the elements of R are either 1 or 0, we equivalently have $Q_{ii}^{-1/2} = \frac{1}{||R_i||}$

Let P be a $m \times m$ diagonal matrix, where the j-th diagonal element is the degree of user node j. Thus, we can think of the j-th diagonal element as the sum of the elements of the j-th row vector in R, or equivalently the sum of the elements of the j-th column vector of R^T. Then, $P^{-1/2}$ is the $m \times m$ diagonal matrix, where element $P_{jj}^{-1/2} = \frac{1}{\sqrt{P_{jj}}}$. Similarly to above, since all the elements of R are either 1 or 0, we equivalently have $P_{jj}^{-1/2} = \frac{1}{||R_j||}$

Problem
Let S_I be an $n \times n$ matrix such that the element in row i and column j is the cosine similarity of item i and item j. Thus, we want $S_{I(ij)} = \frac{R_i^T \cdot R_j}{||R_i^T||||R_j||}$, where R_i^T is the i-th row vector of R^T (and also the i-th column vector of R) and R_j is the j-th row vector of R, giving us the cosine similarity of items i and j. We will prove that $S_I = Q^{-1/2} R^T R Q^{-1/2}$.

Proof
First, we construct the numerator of the cosine similarity equation by computing $X = R^T R$. Elements of X are computed as follows:

$$X_{ij} = R_i^T \cdot R_j \tag{1}$$

where R_i^T is the i-th row vector of R^T and R_j is the j-th column vector of R. Now we just have to transform X so that each element includes the denominator of the cosine similarity equation.

Next, we compute $Y = XQ^{-1/2}$, where elements are calculated as follows:

$$Y_{ij} = X_i \cdot Q_j^{-1/2} \tag{2}$$

where X_i is the i-th row vector of X and $Q_j^{-1/2}$ is the j-th column vector of $Q^{-1/2}$. Since $Q^{-1/2}$ is a diagonal matrix, we know that all elements of its j-th column vector will be 0 except for the j-th element. Thus, (2) simplifies to

$$Y_{ij} = X_{ij} * Q_{jj}^{-1/2} \tag{3}$$

$$Y_{ij} = \frac{R_i^T \cdot R_j}{||R_j||} \tag{4}$$

where (4) is true because we showed above that the diagonal elements of $Q^{-1/2}$ are actually just the norms.

Finally, we compute $S_I = Q^{-1/2}Y$, where elements are calculated as follows:

$$S_{I(ij)} = Q_i^{-1/2} \cdot Y_j \tag{5}$$

where $Q_i^{-1/2}$ is the i-th row vector of $Q^{-1/2}$ and Y_j is the j-th column vector of Y. Similarly to above, we can simplifiy because of the properties of $Q^{-1/2}$ we've already discussed:

$$S_{I(ij)} = Q_{ii}^{-1/2} * Y_{ij} \tag{6}$$

$$S_{I(ij)} = \frac{R_i^T \cdot R_j}{||R_i^T||||R_j||} \tag{7}$$

proving that $S_I = Q^{-1/2}R^TRQ^{-1/2}$. ∎

Problem
Let S_U be an $m \times m$ matrix such that the element in row i and column j is the cosine similarity of user i and item j. Thus, we want $S_{U(ij)} = \frac{R_i \cdot R_j^T}{||R_i||||R_j^T||}$, where R_i is the i-th row vector of R and R_j^T is the j-th column vector of R^T, giving us the cosine similarity between users i and j. We will prove that $S_U = P^{-1/2}RR^TP^{-1/2}$.

Proof
First, we construct the numerator of the cosine similarity equation by computing $X = RR^T$. Elements of X are calculated as follows:

$$X_{ij} = R_i \cdot R_j^T \tag{8}$$

where R_i is the i-th row vector of R and R_j^T is the j-th column vector of R^T. Now we just have to transform X so that each element incorporates the denominator of the cosine similarity equation.

Next, we compute $Y = XP^{-1/2}$, where elements are calculated as follows:

$$Y_{ij} = X_i \cdot P_j^{-1/2} \tag{9}$$

where X_i is the i-th row vector of X and $P_j^{-1/2}$ is the j-th column vector of $P^{-1/2}$. We know that because $P^{-1/2}$ is a diagonal matrix, all elements of its j-th column vector are 0 except for the j-th element. Therefore, (9) simplifies to

$$Y_{ij} = X_{ij} * P_{jj}^{-1/2} \tag{10}$$

$$Y_{ij} = \frac{R_i \cdot R_j^T}{||R_j^T||} \tag{11}$$

Last, we compute $S_U = P^{-1/2}Y$, where elements are calculated as follows:

$$S_{U(ij)} = P_i^{-1/2} \cdot Y_j \tag{12}$$

where $P_i^{-1/2}$ is the i-th row vector of $P^{-1/2}$ and Y_j is the j-th column vector of Y. Similarly to above, we can simplify because of the properties of $P^{-1/2}$ that we've already discussed:

$$S_{U(ij)} = P_{ii}^{-1/2} * Y_{ij} \tag{13}$$

$$S_{U(ij)} = \frac{R_i \cdot R_j^T}{||R_i||||R_j^T||} \tag{14}$$

proving that $S_U = P^{-1/2}RR^TP^{-1/2}$. ■

Answer to Question 4(c)

Problem
For the item-item case, show that $\Gamma = RQ^{-1/2}R^TRQ^{-1/2}$, where $\Gamma_{us} = r_{u,s} = \sum_{x \in items} R_{ux} * \text{cos-sim}(x, s)$.

Proof
Let $S_I = Q^{-1/2}R^TRQ^{-1/2}$, as proven in in (4b). This is an $n \times n$ matrix where each entry, $S_{I(xs)}$ corresponds to the cosine similarity of items x and s. R is the original $m \times n$ matrix, where each row represents a user, and each column represents an item. Thus, when we compute $\Gamma = RS_I$ we get an $m \times n$ matrix, where each element is calculated as follows:

$$\Gamma_{us} = R_u \cdot S_{I(s)} \tag{1}$$
$$\Gamma_{us} = \sum_{x \in items} R_{ux} * S_{I(xs)} \tag{2}$$
$$\Gamma_{us} = \sum_{x \in items} R_{ux} * \text{cos-sim}(x, s) \tag{3}$$

In (1), we take the dot product of row vector u of R (where each element represents the user's rating of an item) and column vector s of S_I (where each element represents the cosine similarity of item s and another item). We expand this in (2) and end up with (3), which is exactly what we were seeking to prove. ■

Problem
For the user-user case, show that $\Gamma = P^{-1/2}RR^TP^{-1/2}R$, where $\Gamma_{us} = r_{u,s} = \sum_{x \in users} \text{cos-sim}(u, x) * R_{xs}$.

Proof
Let $S_U = P^{-1/2}RR^TP^{-1/2}$, as proven in (4b). This is an $m \times m$ matrix, where each entry, $S_{U(ux)}$ corresponds to the cosine similarity of users u and x. R is the original $m \times n$ matrix, where each row represents a user, and each column represents an item. Thus, when we compute $\Gamma = S_U R$, we get an $m \times n$ matrix, where each element is calculated as follows:

$$\Gamma_{us} = S_{U(u)} \cdot R_s \tag{4}$$
$$\Gamma_{us} = \sum_{x \in users} S_{U(ux)} * R_{xs} \tag{5}$$
$$\Gamma_{us} = \sum_{x \in users} \text{cos-sim}(u, x) * R_{xs} \tag{6}$$

In (4), we take the dot product of the u-th row vector of S_U (where each element represents the cosine similarity of user u and some other user) and the s-th column vector of R (where each element is the rating that was given to item s by some user). We expand this in (5) and end up with (6), which is what we sought to prove. ■

Answer to Question 4(d)

User-user Filter Recommendations:
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"

Item-item Filter Recommendations:
1. "FOX 28 News at 10pm"
2. "Family Guy"
3. "NBC 4 at Eleven"
4. "2009 NCAA Basketball Tournament"
5. "Access Hollywood"