CSE 547: Machine Learning for Big Data
Homework 2

Answer to Question 1(a)

1. We use the fact that summation and multiplication are commutative operations.

$$Tr(AB^T) = \sum_{i=1}^{n}\sum_{j=1}^{d} a_{ij}b_{ij} = \sum_{j=1}^{d}\sum_{i=1}^{n} a_{ij}b_{ij} = \sum_{j=1}^{d}\sum_{i=1}^{n} b_{ij}a_{ij} = Tr(B^TA)$$

2.(1) Note that X^TX is a symmetric real matrix for any real matrix X, hence $\frac{1}{n}X^TX$ is as well. For symmetric real matrices, we can decompose as follows: $\Sigma = \frac{1}{n}X^TX = V\Lambda V^T$, where $\Lambda \in \mathbb{R}^{d\times d}$ is a diagonal matrix containing the d eigenvalues of $\Sigma \in \mathbb{R}^{d\times d}$. Note that without loss of generality, $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d$. V is an orthogonal matrix, consisting of the eigenvectors of Σ (columnwise). Thus with the use of (a),

$$Tr(\Sigma) = Tr(V\Lambda V^T) = Tr(V^TV\Lambda) \overset{\text{V orthogonal}}{=} Tr(\Lambda) = \sum_{i=1}^{d}\lambda_i$$

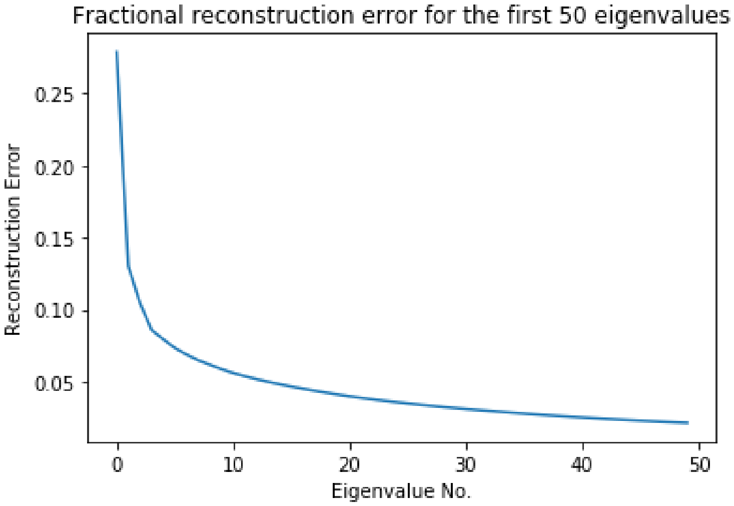
2.(2) Using the first part of this exercise, we get

$$\begin{aligned}Tr(\Sigma) &= Tr(\frac{1}{n}X^TX) = \frac{1}{n}Tr(X^TX) = \frac{1}{n}Tr(XX^T)\\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d} x_{ij}x_{ij} = \frac{1}{n}\sum_{i=1}^{n} < X_i, X_i > = \frac{1}{n}\sum_{i=1}^{n} ||X_i||_2^2\end{aligned}$$

Answer to Question 1(b)

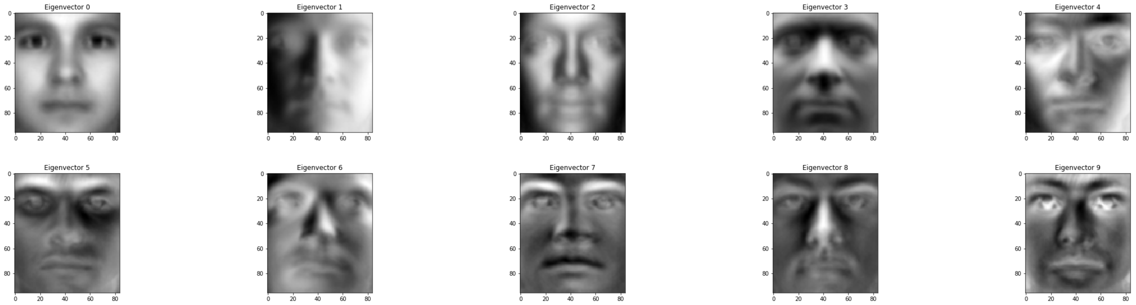
The reason for the first eigenvalue being so much higher than the others is that our data was not demeaned, but rather in the range $[0, 1]^N$ for each sample (picture), where N is the length of the flattened picture. Hence the first eigenvector points towards the centre of mass of X, making it significantly larger than the rest.

- $\lambda_1 = 781.81$
- $\lambda_2 = 161.15$
- $\lambda_{10} = 3.34$
- $\lambda_{30} = 0.81$
- $\lambda_{50} = 0.39$
- $\sum_{i=1}^{d} \lambda_i = 1084.21$



Answer to Question 1(c)

While the Eigenvector 0 seems to capture an average face (which is in line with the assumption we made about the first large eigenvector in 1(b)), the other seem to capture general features of the images like different lightning (Eigenvector 1,4,5,6) or contours of a face (Eigenvector 2, 3,4, 7, 8,9). Some might even be encoding some general features that belong to a face, like the nose or the eyes (EV 8 and 9, respectively). Some are not so easy distinghuisable and might fall into different categories.



Answer to Question 1(d)

It can be seen that the first eigenvector always just represents the average face, with the general brightness already encoded (note that when we do $X\lambda_1$, X will generally have lower values for darker pictures, so the result will be darker). Since the second eigenvector seems to represent lightning from the side, only the representation of the second image changes, since it is the only having light coming from the side. For five eigenvalues, the faces all have gained a little bit of contours and the lightning of image 68 is-apart from the nose-almost entirely encoded. With ten eigenvectors, the faces can be well distinguished. Also, all lightnings (or rather, stark contrasts) are captured at this point. With fifty eigenvectors, there is almost no difference to the original image.

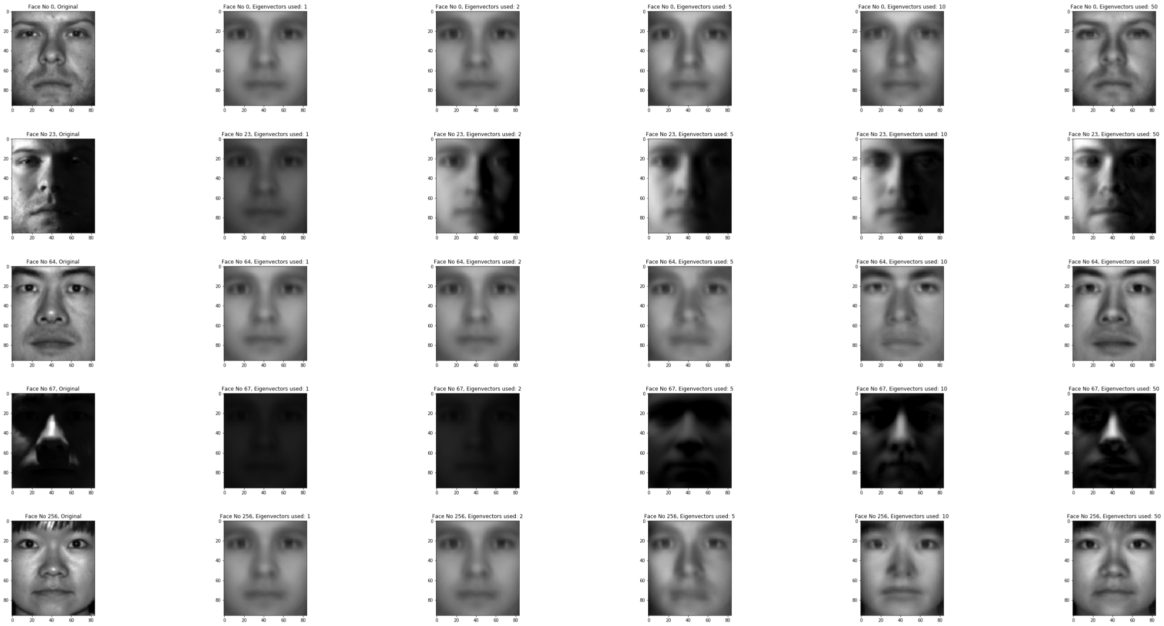

Face No 0, Original
Face No 0, Eigenvectors used: 1
Face No 0, Eigenvectors used: 2
Face No 0, Eigenvectors used: 5
Face No 0, Eigenvectors used: 10
Face No 0, Eigenvectors used: 50
Face No 23, Original
Face No 23, Eigenvectors used: 1
Face No 23, Eigenvectors used: 2
Face No 23, Eigenvectors used: 5
Face No 23, Eigenvectors used: 10
Face No 23, Eigenvectors used: 50
Face No 64, Original
Face No 64, Eigenvectors used: 1
Face No 64, Eigenvectors used: 2
Face No 64, Eigenvectors used: 5
Face No 64, Eigenvectors used: 10
Face No 64, Eigenvectors used: 50
Face No 67, Original
Face No 67, Eigenvectors used: 1
Face No 67, Eigenvectors used: 2
Face No 67, Eigenvectors used: 5
Face No 67, Eigenvectors used: 10
Face No 67, Eigenvectors used: 50
Face No 256, Original
Face No 256, Eigenvectors used: 1
Face No 256, Eigenvectors used: 2
Face No 256, Eigenvectors used: 5
Face No 256, Eigenvectors used: 10
Face No 256, Eigenvectors used: 50
0
20
40
60
80

Figure 1: Original and reconstruction of Images 1,24,64,68,256 (per row). Each column represents a different amount of eigenvectors used for reconstruction (1,2,5,10,50)

Answer to Question 2(a)

For the random initialization, we have an improvement of 15.79% over the initial cost after ten iterations. For the maximum distance initialization, we obtain an improvement of 55.33% over the initial cost after ten iterations.

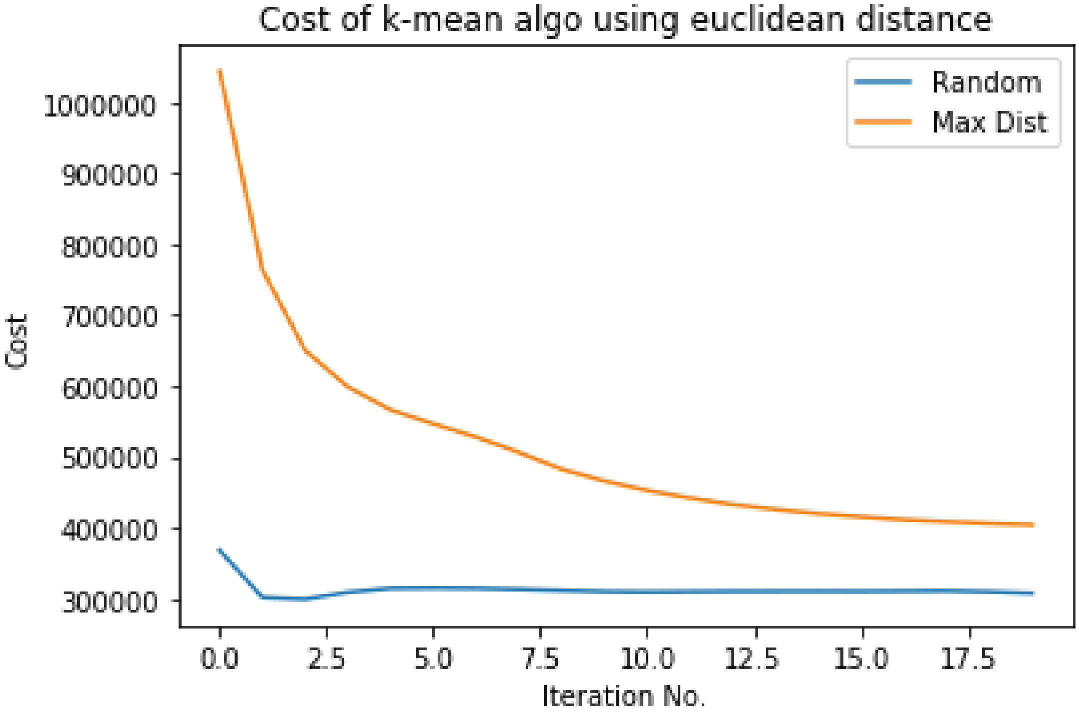


Figure 2: Cost vs. iteration for the two initialization strategies, using Euclidean distance.

Answer to Question 2(b)

For the random initialization, we see an improvement of 18.51% over the initial cost after ten iterations. For the maximum distance initialization, we obtain an improvement of 50.41% over the initial cost after ten iterations. In terms of cost, random initialization seems to work much better than the maximum distance approach, since it only took a few iterations for the random one to plateau out, while the maximum distance one starts out with a much higher cost and is still after 20 iterations a decent margin higher than the initial(!) cost of random initialization. Note that this is true even for the case of Manhattan distance, although c2.txt is distributed in a manner that maximizes L2 distances between initial clusters (in general, L1 and L2 distance do not coincide)

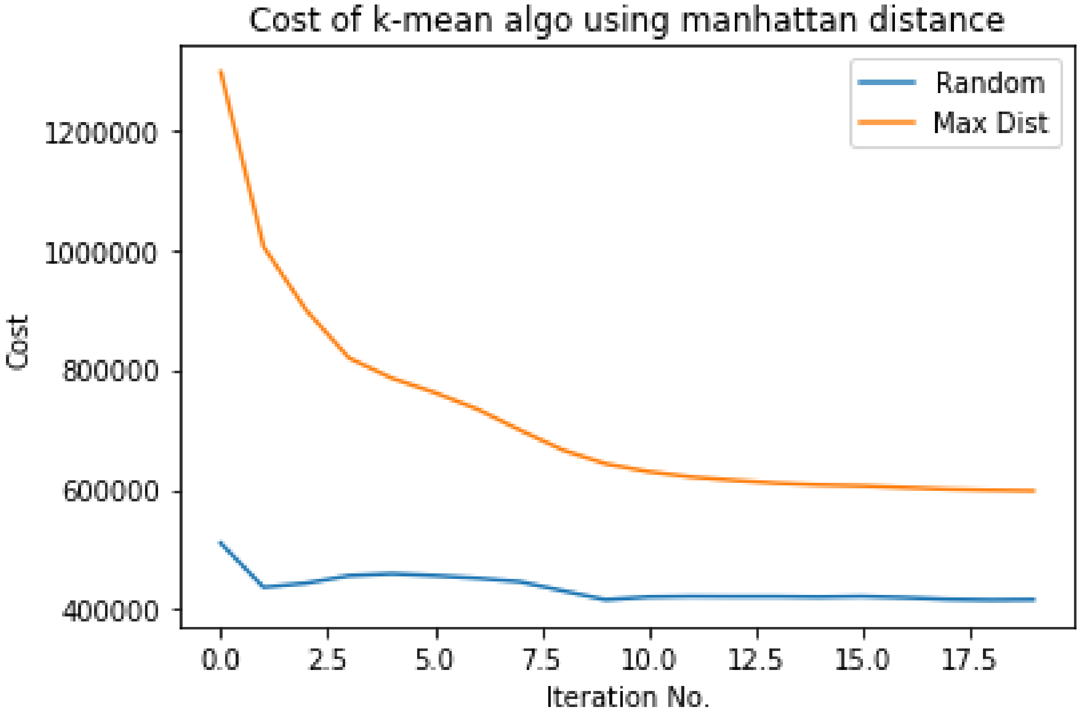


Figure 3: Cost vs. iteration for the two initialization strategies, using Manhattan distance.

Answer to Question 3(a)

Let $r(P, Q)$ be the regularization term of $E(R, P, Q)$, i.e.

$$E(R, P, Q) = \sum_{(i,u) \in \text{ Ratings}} (R_{iu} - q_i p_u)^2 + r(P, Q)$$

Note that the regularization term does not depend on R. Then

$$\begin{aligned}
\epsilon_{iu} &= \frac{\partial E(R, P, Q)}{\partial R_{iu}} = \frac{\partial \sum_{(i,u)\in\text{Ratings}} (R_{iu} - q_i p_u)^2 + r(P, Q)}{\partial R_{iu}} \\
&= \frac{\partial \sum_{(i,u)\in\text{Ratings}} (R_{iu} - q_i p_u)^2}{\partial R_{iu}} \\
&= 2(R_{iu} - q_i p_u)
\end{aligned}$$

For q_i, we get

$$\begin{aligned}
\frac{\partial E(R, P, Q)}{\partial q_i} &= \frac{\partial \sum_{(i,u)\in\text{Ratings}} (R_{iu} - q_i p_u)^2 + r(P, Q)}{\partial q_i} \\
&= \frac{\partial \sum_{(i,u)\in\text{Ratings}} (R_{iu} - q_i p_u)^2}{\partial q_i} + \frac{\partial r(P, Q)}{\partial q_i} \\
&= -2p_u(R_{iu} - q_i p_u) + \frac{\lambda \left(\sum_{u=1}^{n} ||p_u||_2^2 + \sum_{j=1}^{n} ||q_j||_2^2 \right)}{\partial q_i} \\
&= -2p_u(R_{iu} - q_i p_u) + 2\lambda q_j = -p_u \epsilon_{iu} + 2\lambda q_j
\end{aligned}$$

Analogously, we get

$$\frac{\partial E(R, P, Q)}{\partial p_u} = -q_i \epsilon_{iu} + 2\lambda p_u$$

Therefore, our update equations are

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

Answer to Question 3(b)

We chose a learning rate of .025 to obtain an error curve as depicted in Figure 4

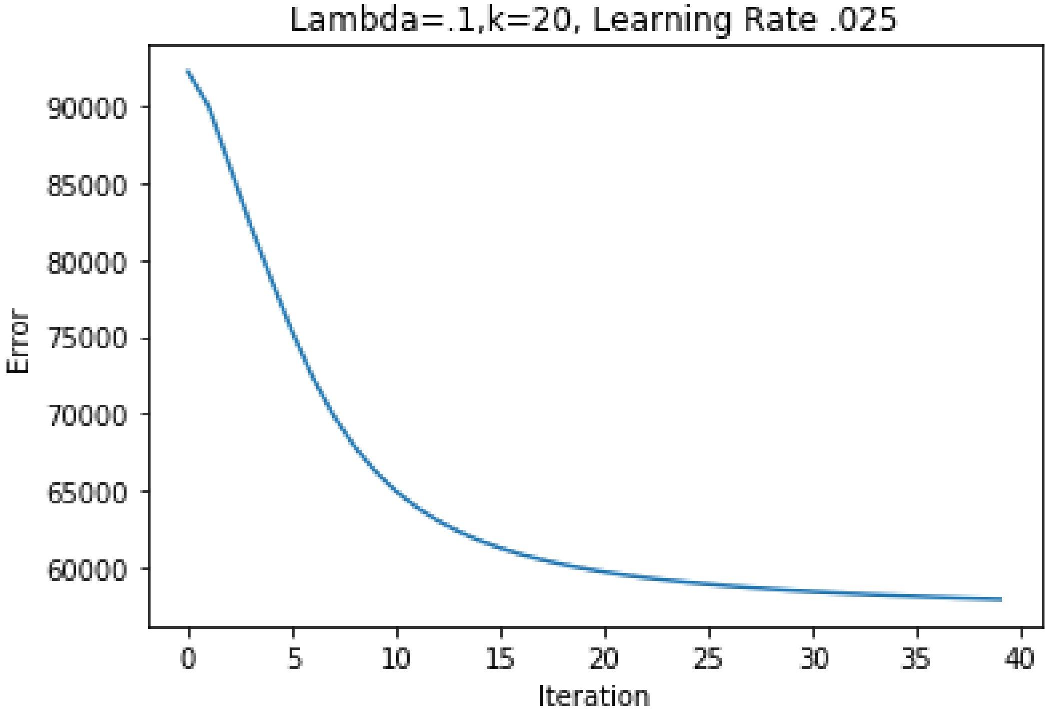


Figure 4: Latent Factor Model Error over 40 Iterations

Answer to Question 4(a)

The bipartite graph can be defined as $G = (V, E)$, with V the set of nodes, E the set of edges. We denote the set of all Items as $\mathcal{I}$ and the set of all users $\mathcal{U}$. Furthermore for a given user u, we denote $V_u := \{i \in \mathcal{I} : (u, i) \in E\}$ as set of all items the user is connected to, or 'likes'. Let R_i be the i-th row of the ratings matrix R. Then

$$T_{ij} = R_i \cdot R_j = \sum_{k=1}^{n} R_{ik} R_{jk}$$

Note that $R_{ik} R_{jk} = 1$ if and only if both user i and j like the item k. In other words, if $k \in V_i \cap V_j$. Hence above sum is equal to the amount of items that are liked by both users:

$$T_{ij} = R_i \cdot R_j = \sum_{k=1}^{n} R_{ik} R_{jk} = |V_i \cap V_j|$$

For the case of $i = j$, this of course is simply the degree of user i, or the amount of items the user i liked, which is P_{ii}. Mathematically, this means

$$P_{ii} = \sum_{k=1}^{n} R_{ik} R_{ik} = \sum_{k=1}^{n} R_{ik}^2 = ||R_i||^2$$

which will be useful in part b) of this exercise. Note that in both cases $i = j$ and $i \neq j$, T_{ij} can be viewed as a kind of similarity: The higher the value, the more items are liked by both i and j.

Answer to Question 4(b)

In order for S_I, S_U to be well-defined, in the following we assume none of the rows and columns of R are zero-vectors. We show that $(S_I)_{ij} = (Q^{-\frac{1}{2}}R^TRQ^{-\frac{1}{2}})_{ij}$ for $i, j \in \{1, \dots, n\}$ arbitray.
Let $R_{.,j}$ denote the j-th column of R. First, note that by definition of S_I,

$$(S_I)_{ij} = \frac{R_{.,i} \cdot R_{.,j}}{||R_{.,i}||||R_{.,j}||} \tag{1}$$

Similar to (a), $(R^TR)_{i,j} = R_{.,i} \cdot R_{.,j}$, where $R_{.,j}$ is the j-th column of R. The entry corresponds to the amount of users that liked both item i and j. Note that for analogous reasoning as in (a), $\sum_{k=1}^m R_{ki}^2 = Q_{ii}$, which is the total amount of users who liked item i. Hence

$$Q_{ii}^{-1/2} = \frac{1}{\sqrt{\sum_{k=1}^n R_{ki}^2}} = \frac{1}{||R_{.,i}||}$$

Using the fact that Q is a diagonal matrix, we get

$$(Q^{-\frac{1}{2}}R^T)_i = \frac{R_{.,i}}{||R_{.,i}||} \quad \text{and} \quad (RQ^{-\frac{1}{2}})_{.,j} = \frac{R_{.,j}}{||R_{.,j}||}$$

Putting everything together, we get

$$(Q^{-\frac{1}{2}}R^TRQ^{-\frac{1}{2}})_{ij} = (Q^{-\frac{1}{2}}R^T)_i(RQ^{-\frac{1}{2}})_{.,j} = \frac{R_{.,i} \cdot R_{.,j}}{||R_{.,i}||||R_{.,j}||} = (S_I)_{ij} \text{ for all } i, j \in \{1, \dots, n\}$$

$$\Leftrightarrow \qquad Q^{-\frac{1}{2}}R^TRQ^{-\frac{1}{2}} = S_I$$

which was to show. We now move on to the user similarity matrix. Analogous to before, we show that $(S_U)_{ij} = (P^{-\frac{1}{2}}RR^TP^{-\frac{1}{2}})_{ij}$ for $i, j \in \{1, \dots, m\}$ arbitray. By definition of the user similarity matrix, it holds

$$(S_U)_{ij} = \frac{R_i \cdot R_j}{||R_i||||R_j||} \tag{2}$$

Note that from (a), we know $P_{ii} = \sum_{k=1}^n R_{ik}R_{ik} = \sum_{k=1}^n R_{ik}^2$, hence

$$P_{ii}^{-1/2} = \frac{1}{\sqrt{\sum_{k=1}^n R_{ik}^2}} = \frac{1}{||R_i||}$$

Therefore

$$(P^{-\frac{1}{2}}R)_i = \frac{R_i}{||R_i||} \quad \text{and} \quad (R^TP^{-\frac{1}{2}})_{.,j} = \frac{R_j}{||R_j||}$$

Hence

$$(P^{-\frac{1}{2}}RR^TP^{-\frac{1}{2}})_{ij} = (P^{-\frac{1}{2}}R)_i(R^TP^{-\frac{1}{2}})_{.,j} = \frac{R_i \cdot R_j}{||R_i||||R_j||} = (S_U)_{ij} \text{ for all } i, j \in \{1, \dots, m\}$$

$$\Leftrightarrow \qquad P^{-\frac{1}{2}}RR^TP^{-\frac{1}{2}} = S_U$$

Answer to Question 4(c)

It is easy to see that S_I, S_U are symmetric matrices. In the case of user-user collaborative filtering, for arbitrary fixed user u and item s we have the following recommendation

$$r_{us} = \sum_{x \in \mathcal{U}} \text{cos-sim}(x, u) R_{x,s} = \sum_{x \in \mathcal{U}} (S_U)_{xu} R_{x,s} = \sum_{x \in \mathcal{U}} (S_U)_{ux} R_{x,s} = (S_U)_u R_{.,s}$$

Hence for the user-user collaborative filtering,

$$\Gamma = S_U R = P^{-\frac{1}{2}} R R^T P^{-\frac{1}{2}} R \tag{3}$$

Similarly, for the item-item collaborative filtering, for an arbitrary user u, item s we get

$$r_{us} = \sum_{x \in \mathcal{I}} R_{u,x} \text{cos-sim}(x, s) = \sum_{x \in \mathcal{I}} R_{u,x} (S_I)_{x,s} = R_u (S_I)_{.,s}$$

Hence for the item-item collaborative filtering,

$$\Gamma = R S_I = R Q^{-\frac{1}{2}} R^T R Q^{-\frac{1}{2}} \tag{4}$$

Answer to Question 4(d)

User-User Collaborative Filtering:

- ”FOX 28 News at 10pm”, Score: 908.48
- ”Family Guy”, Score: 861.18
- ”2009 NCAA Basketball Tournament”, Score: 827.60
- ”NBC 4 at Eleven”, Score: 784.78
- ”Two and a Half Men”, Score: 757.60

Item-Item Collaborative Filtering:

- ”FOX 28 News at 10pm”, Score: 31.36
- ”Family Guy”, Score: 30.00
- ”NBC 4 at Eleven”, Score: 29.40
- ”2009 NCAA Basketball Tournament”, Score: 29.23
- ”Access Hollywood”, Score: 28.97

Submission Instructions

Assignment Submission All students should submit their assignments electronically via GradeScope. Students may typeset or scan their **neatly written** homeworks (points **will** be deducted for illegible submissions). Simply sign up on the Gradescope website and use the course code 97EWEW. Please use your UW NetID if possible.

For the non-coding component of the homework, you should upload a PDF rather than submitting as images. We will use Gradescope for the submission of code as well. Please make sure to tag each part correctly on Gradescope so it is easier for us to grade. There will be a small point deduction for each mistagged page and for each question that includes code. Put all the code for a single question into a single file and upload it. Only files in text format (e.g. .txt, .py, .java) will be accepted. **There will be no credit for coding questions without submitted code on Gradescope, or for submitting it after the deadline**, so please remember to submit your code.

Late Day Policy All students will be given two no-questions-asked late periods, but only one late period can be used per homework and cannot be used for project deliverables. A late-period lasts 48 hours from the original deadline (so if an assignment is due on Thursday at 11:59 pm, the late period goes to the Saturday at 11:59pm Pacific Time).

Academic Integrity We take academic integrity extremely seriously:
(`https://www.cs.washington.edu/academics/misconduct`).
We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down the solutions and the code independently. In addition, each student should write down the set of people whom they interacted with.

Discussion Group (People with whom you discussed ideas used in your answers):

Romain Camilleri, Andrew Wagenmaker

On-line or hardcopy documents used as part of your answers: None

I acknowledge and accept the Academic Integrity clause.

Emil Azadian