CS550: Massive Data Mining and Learning
Homework 2

Due 11:59pm Monday, March 23, 2020

Only one late period is allowed for this homework (11:59pm Tuesday 3/24)

Submission Instructions

Assignment Submission Include a signed agreement to the Honor Code with this assignment. Assignments are due at 11:59pm. All students must submit their homework via Sakai. Students can typeset or scan their homework. Students also need to include their code in the final submission zip file. Put all the code for a single question into a single file.

Late Day Policy Each student will have a total of *two* free late days, and for each homework only one late day can be used. If a late day is used, the due date is 11:59pm on the next day.

Honor Code Students may discuss and work on homework problems in groups. This is encouraged. However, each student must write down their solutions independently to show they understand the solution well enough in order to reconstruct it by themselves. 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 is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code seriously and expect students to do the same.

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

On-line or hardcopy documents used as part of your answers:

I acknowledge and accept the Honor Code.

*(Signed) Keya Desai*____

If you are not printing this document out, please type your initials above.

Answer to Question 1(a)

Yes, matrices MM^T and M^TM are symmetric, square and real.

- **Symmetric:** Transpose of first matrix is equal to the other: $(MM^T)^T = M^TM$. Thus, both matrices are symmetric.
- **Square:** M is a matrix of size $p \times q$. Hence size of $MM^T = (p \times q) \times (q \times p) = (p \times p)$. Similarly, M^T will be a square matrix of size $(q \times p) \times (p \times q) = (q \times q)$.
- **Real:** Since M is real, M^T will be real, and their multiplication will be real.

Answer to Question 1(b)

Let e be the eigenvector and λ be the corresponding eigenvalue of matrix M^TM.

$$\begin{aligned}\implies M^TM(e) &= \lambda(e) \\ \implies M(M^TM)(e) &= M\lambda(e) \\ \implies MM^T(Me) &= \lambda(Me)\end{aligned} \tag{1}$$

Therefore, eigenvalue of $MM^T = \lambda$ but eigenvector $= Me$. Hence, MM^T and M^TM have the same eigenvalues but different eigenvectors.

Answer to Question 1(c)

Eigen value decomposition of a real, symmetric and square matrix M is written as:

$$B = Q \wedge Q^T$$

Since we proved in 1(a) that M^TM is a real, symmetric and square matrix, it's eigen value decomposition can be written as:

$$M^TM = Q \wedge Q^T \tag{2}$$

Answer to Question 1(d)

$$\begin{aligned} M &= U\Sigma V^T \\ \implies M^TM &= (U\Sigma V^T)^T(U\Sigma V^T) \\ &= (V\Sigma^T U^T)(U\Sigma V^T) \\ &= V\Sigma^T(U^TU)\Sigma V^T \\ &= V\Sigma^T I \Sigma V^T \\ &= V\Sigma^2 V^T \end{aligned} \tag{3}$$

Answer to Question 1(e)(a)

U = $\begin{bmatrix} 0.27854301 & 0.5 \\ 0.27854301 & -0.5 \\ 0.64993368 & 0.5 \\ 0.64993368 & -0.5 \end{bmatrix}$
$\Sigma = \begin{bmatrix} 7.61577311 & 1.41421356 \end{bmatrix}$
$V^T = \begin{bmatrix} 0.70710678 & 0.70710678 \\ -0.70710678 & 0.70710678 \end{bmatrix}$

Answer to Question 1(e)(b)

$evals = \begin{bmatrix} 58. & 2. \end{bmatrix}$
$evecs = \begin{bmatrix} 0.70710678 & -0.70710678 \\ 0.70710678 & 0.70710678 \end{bmatrix}$

Answer to Question 1(e)(c)

From Eq. 2 and Eq. 3,

$$Q \wedge Q^T = U \Sigma V^T \tag{4}$$

we can conclude that eigen value decomposition of $M^T M$ $(= Q \wedge Q^T)$, gives eigen vecotrs (Q) which corresponds to V obtained by singular value decomposition of M ($= U \Sigma V^T$). The values obtained in the experiment, corroborates this observation:

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

Answer to Question 1(e)(d)

From Eq. 4, we can observe that:

$$\wedge = \Sigma^2 \tag{5}$$

That is, eigen values of $M^T M$ correspond to square of singular values of M.

Singular values of M, $\Sigma = \begin{bmatrix} 7.61577311 & 1.41421356 \end{bmatrix}$
Eigen values of $M^T M, \wedge = \begin{bmatrix} 58. & 2. \end{bmatrix} = \begin{bmatrix} 7.61577311 & 1.41421356 \end{bmatrix}^2$

Answer to Question 2(a)

Given that the web has no dead ends, we need to prove that $w(r') = w(r)$.

$$\begin{aligned} w(r') &= \sum_{i=1}^{n} r'_i \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} r_j \\ &= \sum_{j=1}^{n} (\sum_{i=1}^{n} M_{ij}) r_j \end{aligned} \tag{6}$$

The inner summation corresponds to column sum for each column of matrix M. Since there are no dead-ends, the sum of each column will be 1.

$$w(r') = \sum_{j=1}^{n} r_j = w(r) \tag{7}$$

Answer to Question 2(b)

We are given that,

$$r'_i = \beta \sum_{i=1}^{n} M_{ij} r_j + \frac{(1-\beta)}{n} \tag{8}$$

where $1 - \beta$ is the teleportation probability. Hence, $w(r')$ will now be given by:

$$\begin{aligned} w(r') &= \sum_{i=1}^{n} r'_j \\ &= \sum_{i=1}^{n} (\beta \sum_{j=1}^{n} M_{ij} r_j + \frac{(1-\beta)}{n}) \\ &= \beta \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} r_j + \sum_{i=1}^{n} \frac{(1-\beta)}{n} \\ &= \beta \sum_{j=1}^{n} r_j + \frac{(1-\beta)}{n} n \\ &= \beta w(r) + (1-\beta) \end{aligned} \tag{9}$$

For $w(r') = w(r)$,

$$
\begin{aligned}
w(r) &= \beta w(r) + (1-\beta) \\
\implies (1-\beta) w(r) &= (1-\beta) \\
\implies w(r) &= 1
\end{aligned} \tag{10}
$$

Thus, $\boxed{w(r) = w(r') = 1}$

Answer to Question 2(c)(a)

The equation for $r_i^{'}$ in terms of β, M, and r:

$$
\begin{aligned}
r_i' &= \beta \sum_{j=1}^{n} M_{ij} r_j + \frac{(1-\beta)}{n} \sum_{j \in live} r_j + \frac{1}{n} \sum_{j \in dead} r_j \\
&= \beta \sum_{j=1}^{n} M_{ij} r_j + \frac{(1-\beta)}{n} \sum_{j \in live} r_j + \frac{(1-\beta)+\beta}{n} \sum_{j \in dead} r_j \\
&= \beta \sum_{j=1}^{n} M_{ij} r_j + \frac{(1-\beta)}{n} \sum_{j \in live} r_j + \frac{(1-\beta)}{n} \sum_{j \in dead} r_j + \frac{(\beta)}{n} \sum_{j \in dead} r_j \\
&= \beta \sum_{j=1}^{n} M_{ij} r_j + \frac{(1-\beta)}{n} \sum_{j} r_j + \frac{(\beta)}{n} \sum_{j \in dead} r_j
\end{aligned} \tag{11}
$$

Given that w(r) = 1, $\implies \sum_{j=1}^{n} r_j = 1$, hence,

$$
\boxed{r_i^{'} = \beta \sum_{j=1}^{n} M_{ij} r_j + \frac{1-\beta}{n} + \frac{\beta}{n} \sum_{j \in dead} r_j} \tag{12}
$$

Answer to Question 2(c)(b)

Using Eq. 12,

$$
\begin{aligned}
w(r') &= \sum_{i=1}^{n} r'_j \\
&= \sum_{i=1}^{n}(\beta \sum_{j=1}^{n} M_{ij}r_j + \frac{(1-\beta)}{n} + \frac{\beta}{n} \sum_{j \in dead} r_j) \\
&= \beta \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij}r_j + \sum_{i=1}^{n} \frac{(1-\beta)}{n} + \sum_{i=1}^{n} \frac{\beta}{n} \sum_{j \in dead} r_j \\
&= \beta \sum_{j=1}^{n} \sum_{i=1}^{n} M_{ij}r_j + \frac{(1-\beta)}{n} n + \frac{\beta \sum_{j \in dead} r_j}{n} n \\
&= \beta \sum_{j=1}^{n} \sum_{i=1}^{n} M_{ij}r_j + (1-\beta) + \beta \sum_{j \in dead} r_j
\end{aligned}
\tag{13}
$$

$\sum_{i=1}^{n} M_{ij} = 1 \quad \forall j \in live$. And $\sum_{j=1}^{n} M_{ij} = 0 \quad \forall j \in dead$. Therefore,

$$
\begin{aligned}
w(r') &= \beta \textstyle\sum_{j \in live}^{n} (1) r_j + (1-\beta) + \beta \displaystyle\sum_{j \in dead} r_j \\
&= \beta (\textstyle\sum_{j \in live}^{n} r_j + \displaystyle\sum_{j \in dead} r_j) + (1-\beta) \\
&= \beta (\textstyle\sum_{j=1}^{n} r_j) + (1-\beta) \\
&= \beta w(r) + (1-\beta) \\
&= \beta + (1-\beta) \\
&= 1
\end{aligned}
\tag{14}
$$

Thus, $\boxed{w(r') = 1}$

Answer to Question 3(a)

Node IDs with highest page rank:

Node IDs	Page rank
53	0.0379
14	0.0359
1	0.0352
40	0.0339
27	0.0331

Answer to Question 3(b)

Node IDs with lowest page rank:

Node IDs	Page rank
85	0.0033
59	0.0035
81	0.0036
37	0.0037
89	0.0038

Answer to Question 4(a)

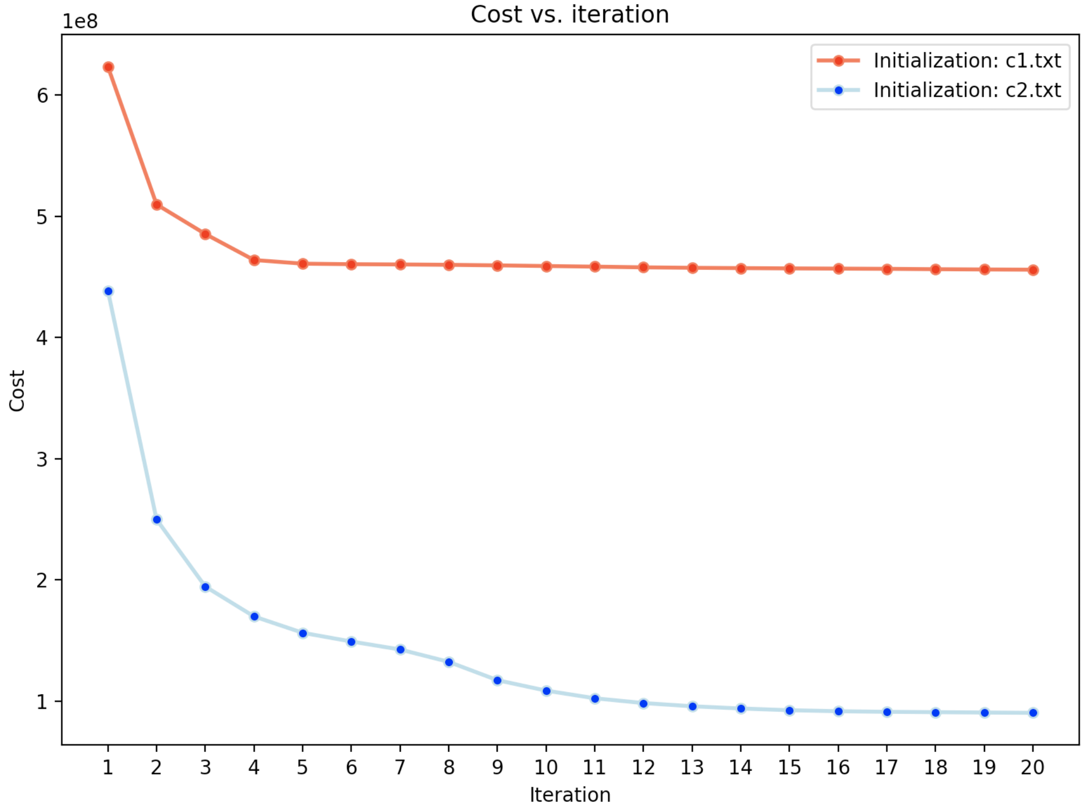

Figure 1: Cost vs. Iteration for two initialization methods

Answer to Question 4(b)

Percentage change in cost after 10 iterations of the k-Means algorithm when the cluster centroids are initialized using:

1. **c1.txt = 26.39%**
2. **c2.txt = 75.25%**

Cluster centroids are chosen randomly in c1.txt whereas c2.txt contains initial cluster centroids as far away as possible. Cluster initialization using the 2nd technique is better because clusters are spread out in vector space. So every point in the data will be mapped to a nearby cluster from the 1st iteration itself, and hence, the algorithm requires lesser number of iterations to converge. Same can be observed using the percent improvement values after 10 iterations. Initialization using c2.txt is converging at a faster rate than initialization using c1.txt