CS550: Massive Data Mining and Learning Spring 2022
Problem Set 2
Due 11:59pm Monday, March 21, 2022

Submission Instructions

Submission instructions: These questions require thought but do not require long answers. Please be as concise as possible. You should submit your answers as a writeup in PDF format, for those questions that require coding, write your code for a question in a single source code file, and name the file as the question number (e.g., question_1.java or question_1.py), finally, put your PDF answer file and all the code files in a folder named as your Name and NetID (i.e., Firstname-Lastname-NetID.pdf), compress the folder as a zip file (e.g., Firstname-Lastname-NetID.zip), and submit the zip file via Canvas.

For the answer writeup PDF file, we have provided both a word template and a latex template for you, after you finished the writing, save the file as a PDF file, and submit both the original file (word or latex) and the PDF file.

Late Policy: The homework is due on 3/21 (Monday) at 11:59pm. We will release the solutions of the homework on Canvas on 3/25 (Friday) 11:59pm. If your homework is submitted to Canvas before 3/21 11:59pm, there will no late penalty. If you submit to Canvas after 3/21 11:59pm and before 3/25 11:59pm (i.e., before we release the solution), your score will be penalized by 0.9^k, where k is the number of days of late submission. For example, if you submitted on 3/24, and your original score is 80, then your final score will be $80* 0.9^3=58.32$ for 24-21=3 days of late submission. If you submit to Canvas after 3/25 11:59pm (i.e., after we release the solution), then you will earn no score for the homework.

Honor Code: Students may discuss the homework problems with peers. 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 directly obtained from the web or others 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)___Vaishnavi_Manthena____________________________________

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

Answer to Question 1(a)

- $(AB)^T = B^T * A^T$
- $((A)^T)^T = A$
- If $A^T = A$, then A is symmetric

$(MM^T)^T = ((M)^T)^T * M^T = MM^T$
$(M^TM)^T = M^T * ((M)^T)^T = M^TM$

Therefore, MM^T and M^TM are symmetric.

MM^T is a $(p*q)*(q*p) = (p*p)$ dimensional matrix.
M^TM is a $(q*p)*(p*q) = (q*q)$ dimensional matrix.

Therefore, MM^T and M^TM are square matrices.

The $(i,j)th$ entry of MM^T is obtained by the dot product of the i^{th} row of M and the j^{th} column of M^T, which are real vectors. So, any $(i,j)th$ entry of MM^T is real. Similarly, the $(i,j)th$ entry of M^TM is obtained by the dot product of the i^{th} row of M^T and the j^{th} column of M, which are real vectors. So, any $(i,j)th$ entry of M^TM is real.

Therefore, MM^T and M^TM are real matrices.

Answer to Question 1(b)

Say, λ is an eigen value of MM^T and v is the corresponding eigenvector:
$MM^Tv = \lambda v$
Pre-multiplying both sides by M^T:
$M^TMM^Tv = M^T\lambda v$
$M^TM(M^Tv) = \lambda(M^Tv)$
So, λ, is also an eigen value of M^TM. However, the corresponding eigen vector of M^TM is M^Tv (with dimension q*1) and not v (with dimension p*1).

Every eigen value of MM^T is an eigen value of M^TM. However, the corresponding eigen vectors are not equal.

Answer to Question 1(c)

Since M^TM is square, symmetric, and real, its eigen value decomposition will be $Q\Lambda Q^T$.

Answer to Question 1(d)

$M = U\sum V^T$

$U^T U = I \ and \ V^T V = I$ since U and V are column orthonormal.
$\sum$ is a square diagonal matrix. Therefore, $\sum$ is equal to $\sum^T$.

$$M^T M = (U\sum V^T)^T U\sum V^T = (V\sum^T U^T) U\sum V^T = V\sum^T \sum V^T = \boldsymbol{V\sum^2 V^T}$$

Answer to Question 1(e)(a)

U matrix:
[[-0.27854301 0.5   ]
 [-0.27854301 -0.5   ]
 [-0.64993368 0.5   ]
 [-0.64993368 -0.5   ]]

Sigma matrix:
[7.61577311 1.41421356]

V transpose matrix:
[[-0.70710678 -0.70710678]
 [-0.70710678 0.70710678]]

Answer to Question 1(e)(b)

The sorted eigen values of M^T*M are:
[58.0, 2.0]

The rearranged eigen vectors of M^T*M are:
[[0.70710678 -0.70710678]
 [0.70710678 0.70710678]]

Answer to Question 1(e)(c)

Based on derivations of Part C and D:
$M^T M = Q \Lambda Q^T = V\sum^2 V^T$
Since, the singular value decomposition is unique:

- $Q \ and \ V \ are \ related$
- $\Lambda = \sum^2$

From experiment:

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

$Eigen\ vectors\ of\ M^T M\ in\ decreasing\ order\ of\ eigen\ values$:

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

The corresponding columns of the two matrices are parallel. 2nd columns are exactly the same. The 1st columns are along the same line but in opposite directions.

Answer to Question 1(e)(d)

From parts C and D, we have:
$M^T M = Q\Lambda Q^T = V\sum^2 V^T$
Since, the singular value decomposition is unique:

- $Q\ and\ V$ are related.
- $\Lambda = \sum^2$

Also, from the experiment we can see that:
$Eigen\ values\ of\ M^T M$: $58, 2$
$Singular\ values\ of\ M$: $7.61577311, 1.41421356$

Therefore, the eigen values of $M^T M$ are squares of the singular values of M.

Answer to Question 2(a)

$$w(r') = \sum_{i=1}^{n} r_i' = \sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij} r_j = \sum_{j=1}^{n}\sum_{i=1}^{n} M_{ij} r_j = \sum_{j=1}^{n} r_j \sum_{i=1}^{n} M_{ij}$$

$$= \sum_{j=1}^{n} r_j * \left(\frac{1}{k}\right) * k \quad where\ k\ is\ the\ outdegree\ of\ node\ j$$

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

Answer to Question 2(b)

$$w(r') = \sum_{i=1}^{n} r_i' = \sum_{i=1}^{n}\left(\beta\sum_{j=1}^{n} M_{ij}r_j + \frac{1-\beta}{n}\right) = \beta\sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}r_j + \sum_{i=1}^{n}\frac{1-\beta}{n}$$

$= \beta * w(r) + 1 - \beta$ //From previous proof

So, for $w(r') = w(r)$, the following equation should be satisfied:

$w(r) = \beta * w(r) + 1 - \beta$
$w(r) * (1-\beta) = 1 - \beta$
$\boldsymbol{w(r) = 1}$

Answer to Question 2(c)(a)

$r_i' = \beta\sum_{j=1}^{n} M_{ij}r_j + \sum_{j\epsilon live\ nodes}\frac{(1-\beta)r_j}{n} + \sum_{j\epsilon dead\ nodes}\frac{r_j}{n}$

$= \beta\sum_{j=1}^{n} M_{ij}r_j + \sum_{j\epsilon live\ nodes}\frac{(1-\beta)r_j}{n} + \sum_{j\epsilon dead\ nodes}\frac{(1-\beta+\beta)r_j}{n}$

$= \beta\sum_{j=1}^{n} M_{ij}r_j + \sum_{j\epsilon live\ nodes}\frac{(1-\beta)r_j}{n} + \sum_{j\epsilon dead\ nodes}\left(\frac{(1-\beta)r_j}{n} + \frac{\beta r_j}{n}\right)$

$= \beta\sum_{j=1}^{n} M_{ij}r_j + \sum_{j\epsilon live\ nodes}\frac{(1-\beta)r_j}{n} + \sum_{j\epsilon dead\ nodes}\frac{(1-\beta)r_j}{n} + \sum_{j\epsilon dead\ nodes}\frac{\beta r_j}{n}$

$= \beta\sum_{j=1}^{n} M_{ij}r_j + \frac{(1-\beta)}{n} * \left(\sum_{j\epsilon live\ nodes} r_j + \sum_{j\epsilon dead\ nodes} r_j\right) + \sum_{j\epsilon dead\ nodes}\frac{\beta r_j}{n}$

$= \beta\sum_{j=1}^{n} M_{ij}r_j + \frac{(1-\beta)}{n} * (w(r)) + \sum_{j\epsilon dead\ nodes}\frac{\beta r_j}{n}$

$= \beta\sum_{j=1}^{n} M_{ij}r_j + \frac{(1-\beta)}{n} * 1 + \sum_{j\epsilon dead\ nodes}\frac{\beta r_j}{n}$

$\boldsymbol{r_i' = \beta\sum_{j=1}^{n} M_{ij}r_j + \frac{(1-\beta)}{n} + \frac{\beta}{n}\sum_{j\epsilon dead\ nodes} r_j}$

Answer to Question 2(c)(b)

$w(r') = \sum_{i=1}^{n} r_i' = \sum_{i=1}^{n}\left(\beta\sum_{j=1}^{n} M_{ij}r_j + \frac{(1-\beta)}{n} + \frac{\beta}{n}\sum_{j\epsilon dead\ nodes} r_j\right)$

$= \beta\sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}r_j + \sum_{i=1}^{n}\frac{(1-\beta)}{n} + \frac{\beta}{n}\sum_{i=1}^{n}\sum_{j\epsilon dead\ nodes} r_j$

$= \beta\left(\sum_{j=1}^{n}\sum_{i=1}^{n} M_{ij}r_j\right) + (1-\beta) + \frac{\beta}{n} * n * \sum_{j\epsilon dead\ nodes} r_j$

$$= \beta * \left(\sum_{j=1}^{n} r_j \sum_{i=1}^{n} M_{ij}\right) + (1-\beta) + \beta * \left(\sum_{j\epsilon dead\ nodes} r_j\right)$$

$$= \beta * \left(\sum_{j\epsilon live\ nodes} r_j\right) + (1-\beta) + \beta * \left(\sum_{j\epsilon dead\ nodes} r_j\right)$$

$$= \beta * \left(\sum_{j\epsilon live\ nodes} r_j + \sum_{j\epsilon dead\ nodes} r_j\right) + (1-\beta)$$

$$= \beta * w(r) + 1 - \beta = \beta + 1 - \beta = 1$$

$$\boldsymbol{w(r') = 1}$$

Answer to Question 3(a)

node ids with highest rank in descending order of rank:
[53, 14, 1, 40, 27]

node ids with highest rank: rank
53: 0.037868613328747594
14: 0.035866772133529436
1: 0.03514138301760087
40: 0.03383064398237689
27: 0.03313019554724851

Answer to Question 3(b)

node ids with lowest rank in ascending order of rank:
[85, 59, 81, 37, 89]

node ids with lowest rank: rank
85: 0.003234819143382019
59: 0.003444256201194502
81: 0.003580432413995564
37: 0.003714283971941924
89: 0.0038398576156450873

Question 4 Implementation

I did this question using spark. My result is a java project, which is built with maven using a pom.xml file. So, the whole program is given in the project-template folder. My actual implementation is in Question_4.java which is there as you go into the folders one by one.

I ran my program on iLab1.cs.rutgers.edu.

Instructions to run my program:

1. cd into the project-template directory
2. execute the command “mvn install”
3. execute the command “mvn clean package”
4. execute command:
   spark-submit target/Question_4.jar path_to_data_file path_to_centroid_file

The output would be a list of 20 iteration and ϕ value pairs as shown in answer to 4a.

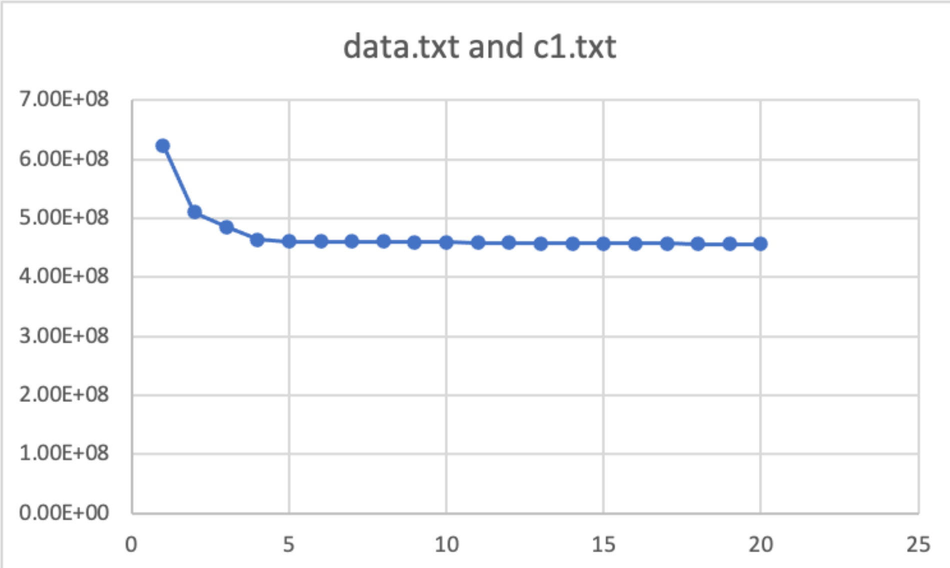
Answer to Question 4(a)

For the graphs, the y-axis is the ϕ values and the x-axis is iteration number.

Output for data.txt and c1.txt:

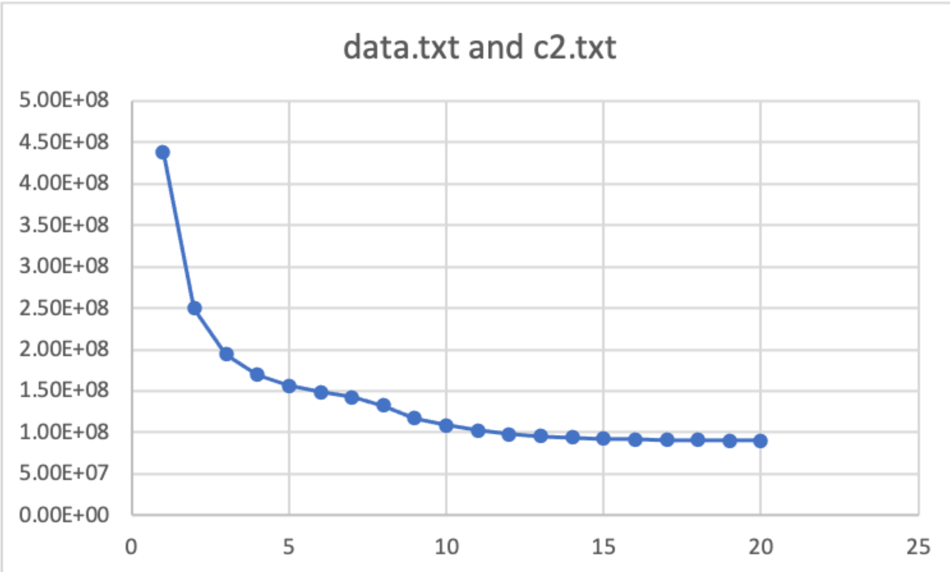
Iteration	phi
1	6.236603453064235E8
2	5.0986290829754597E8
3	4.854806818720084E8
4	4.639970116850107E8
5	4.6096926657299405E8
6	4.6053784798277014E8
7	4.6031309965354246E8
8	4.6000352388940686E8
9	4.595705393177354E8
10	4.590211033422901E8
11	4.584906561919808E8
12	4.579442325879742E8
13	4.575580051986796E8
14	4.572901363523032E8
15	4.570505550595639E8
16	4.5689223561535746E8
17	4.567036307370357E8
18	4.564042030189769E8
19	4.5617780054199505E8
20	4.5598687102734846E8

Graph:



Output for data.txt and c2.txt:

Iteration	phi
1	4.38747790027918E8
2	2.4980393362600294E8
3	1.9449481440631393E8
4	1.6980484145154336E8
5	1.5629574880627596E8
6	1.4909420810896608E8
7	1.4250853161961588E8
8	1.3230386940653005E8
9	1.1717096983719078E8
10	1.0854737717857017E8
11	1.0223720331799614E8
12	9.827801574975717E7
13	9.563022612177444E7
14	9.379331405119292E7
15	9.237713196821108E7
16	9.154160625423913E7
17	9.10455738304242E7
18	9.075224010140836E7
19	9.047017018122767E7
20	9.021641617563146E7



Answer to Question 4(b)

Percentage decrease of cost for c1.txt after 10 iterations:

$$\frac{6.236603453064235\text{E}8 - 4.590211033422901\text{E}8}{6.236603453064235\text{E}8} * 100$$

$$= \frac{6.236603453064235 - 4.590211033422901}{6.236603453064235} * 100$$

$$= \mathbf{26.39886}\%$$

Percentage decrease of cost for c2.txt after 10 iterations:

$$\frac{4.38747790027918\text{E}8 - 1.0854737717857017\text{E}8}{4.38747790027918\text{E}8} * 100$$

$$= \frac{4.38747790027918 - 1.0854737717857017}{4.38747790027918} * 100$$

=75.2597%

Initialization using c2.txt is better for k-means. Intuitively, this is true because in c2.txt centroids are chosen to be points far away from one another and hence the initial cost function value would be lower and its value can effectively improve from here for further iterations.

Also, we can see that within 10 iterations, the percentage improvement in cost when starting with c2.txt (75.26%) is much better than that when starting with c1.txt (26.4%). Also, it can be seen through the graphs, that in general the cost values are less when starting with c2.txt. So, random initialization using c1.txt is not better than initialization using c2.txt.