# CS550: Massive Data Mining and Learning
# Homework 2

Due 11:59pm Saturday, March 23, 2019

Only one late period is allowed for this homework (11:59pm Sunday 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)*Akhilesh Harish Mahajan________________________________________

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

## Answer to Question 1(a)

- $(MM^T)^T = (M^T)^T M^T = MM^T$
  $\therefore MM^T$ is symmetric.
  $(M^T M)^T = M^T (M^T)^T = M^T M$
  $\therefore M^T M$ is symmetric.

- Size of Matrix M is (p,q). $\therefore$ size of Matrix $MM^T$ is (p,q)(q,p), which is (p,p). Hence $MM^T$ is a square matrix.
  Size of Matrix M is (p,q). $\therefore$ size of Matrix $M^T M$ is (q,p)(p,q), which is (q,q). Hence $M^T M$ is a square matrix.

- Since we have that $M$ is real, $M^T$ is also real. Product of 2 real matrices is always real.

## Answer to Question 1(b)

Let $MM^T u_1 = \lambda_1 u_1$ where $\lambda_1$ and $u_1$ is the eigenvalue and eigenvector of $MM^T$ respectively.
Multiplying both sides by $M^T$ on the left hand side, we have:
$M^T MM^T u_1 = M^T \lambda_1 u_1$.
We can write this as:
$M^T M(M^T u_1) = \lambda_1 (M^T u_1)$
$\therefore$ We see that both $MM^T$ and $M^T M$ have same eigenvalues, but different eigenvectors.

## Answer to Question 1(c)

Since $M^T M$ is symmetric, square and real, we can write the eigenvalue decomposition of $M^T M$ as:
$M^T M = Q \Lambda Q^T$

## Answer to Question 1(d)

We have $M = U \Sigma V^T$. Hence,
$M^T M = (U \Sigma V^T)^T U \Sigma V^T = V \Sigma^T U^T U \Sigma V^T$
Now, since $U$ is column orthonormal, $U^T U = I$. Also, since $\Sigma$ is a diagonal matrix, $\Sigma^T = \Sigma$.
$\therefore$ We have:
$M^T M = V \Sigma^2 V^T$

## Answer to Question 1(e)(a)

See question1.ipynb for this answer.

## Answer to Question 1(e)(b)

See question1.ipynb for this answer.

## Answer to Question 1(e)(c)

See question1.ipynb for this answer.

## Answer to Question 1(e)(d)

See question1.ipynb for this answer.

## Answer to Question 2(a)

We want to prove that: $w(r) = w(r')$, that is:
$\sum_{i=1}^{n} r_i = \sum_{i=1}^{n} r'_i$
$\sum_{i=1}^{n} r'_i = \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} r_j$
Since we know that there are no dead ends in Matrix M, all the columns of M sum upto 1.
Hence, we have:
$\sum_{i=1}^{n} r'_i = \sum_{j=1}^{n} r_j \sum_{i=1}^{n} M_{ij} = \sum_{j=1}^{n} r_j$
Hence proved!

## Answer to Question 2(b)

We have:
$w(r') = \sum_{i=1}^{n} r'_i = \sum_{i=1}^{n}[\beta \sum_{j=1}^{n} M_{ij} r_j + \frac{1-\beta}{n}] = \sum_{i=1}^{n} \sum_{j=1}^{n} \beta M_{ij} r_j + (1-\beta)$
$= \beta \sum_{j=1}^{n} r_j \sum_{i=1}^{n} M_{ij} + (1-\beta)$
Since we know that there are no dead ends in Matrix M, all the columns of M sum upto 1.
Hence, we have:
$w(r') = \beta \sum_{j=1}^{n} r_j + (1-\beta) = \beta w(r) + (1-\beta)$
Since we want $w(r') = w(r)$, let's say that their value equals $x$.
$\therefore x = \beta x + (1-\beta)$
$\therefore x = \frac{1-\beta}{1-\beta} = 1$
Hence, when $w(r) = 1$, we shall have $w(r') = w(r)$

## Answer to Question 2(c)(a)

We have:
$r'_i = \sum_{j \in live}[\beta M_{ij} + \frac{1-\beta}{n}] r_j + \sum_{j \in dead} \frac{\beta}{n} r_j = \sum_{j \in live} \beta M_{ij} r_j + \frac{1-\beta}{n} \sum_{j \in live} r_j + \sum_{j \in dead} \frac{\beta}{n} r_j$
Since we have $w(r) = 1$
$r'_i = \sum_{j \in live} \beta M_{ij} r_j + \frac{1-\beta}{n} + \sum_{j \in dead} \frac{\beta}{n} r_j$

## Answer to Question 2(c)(b)

$w(r') = \sum_{i=1}^{n}[\sum_{j \in live} \beta M_{ij} r_j + \frac{1-\beta}{n} + \sum_{j \in dead} \frac{\beta}{n} r_j] =$
$\sum_{i=1}^{n} \beta \sum_{j \in live} M_{ij} r_j + \sum_{i=1}^{n} \frac{1-\beta}{n} + \sum_{i=1}^{n} \sum_{j \in dead} \frac{\beta}{n} r_j$
$= \sum_{j \in live} \beta r_j \sum_{i=1}^{n} M_{ij} + (1-\beta) + \sum_{j \in dead} \sum_{i=1}^{n} \frac{\beta}{n} r_j$
Since for $j \in live$, we have the sum of column on Matrix M as 1, we have:
$w(r') = \sum_{j \in live} \beta r_j + (1-\beta) + \sum_{j \in dead} \beta r_j$
$= \beta \sum_{j \in live} r_j + \beta \sum_{j \in dead} r_j + (1-\beta)$
Since $w(r) = 1$, we have $\sum_{j \in live} r_j + \sum_{j \in dead} r_j = 1$. Hence,
$w(r') = \beta + 1 - \beta = 1$
Hence proved.

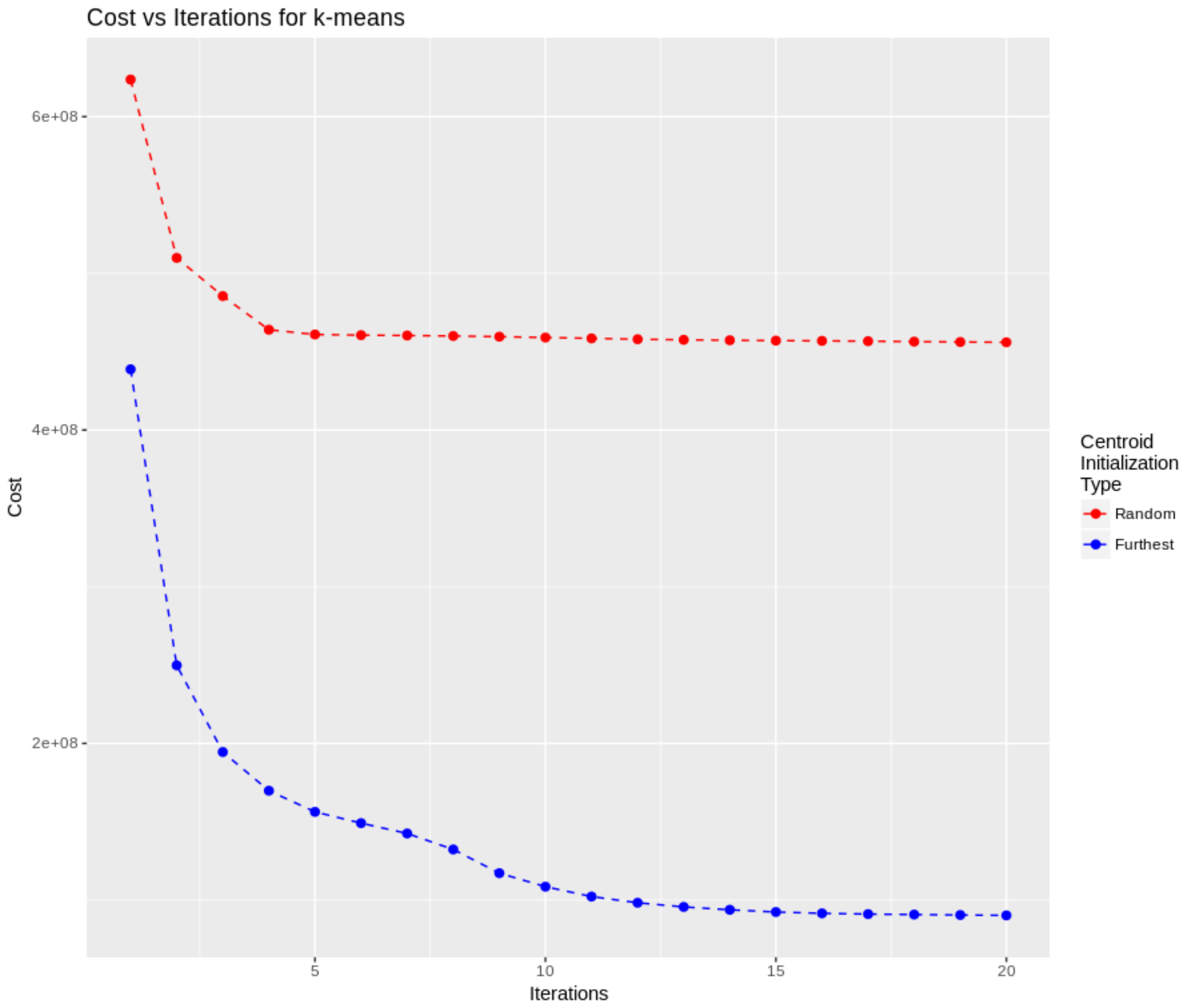
## Answer to Question 3(a)

Top 5 node IDs with highest Pagerank scores: 53, 14, 1, 40, 27
For code, see file question3.ipynb.

## Answer to Question 3(b)

Top 5 node IDs with lowest Pagerank scores: 85, 59, 81, 37, 89
For code, see file question3.ipynb.

## Answer to Question 4(a)



## Answer to Question 4(b)

For c1.txt, the % change after 10 iterations = 26.4
For c2.txt, the % change after 10 iterations = 75.25
c2 is better than c1, as we know that c2 centroids that are as far apart as possible. Hence, the chances of cluster overlap is minimized. The points that are not near center, are now distributed to these furthest centroids, so the cost function is minimized.
In c1, the centroids are randomly initialized, hence, there might be cases where actual clusters are distributed between 2 different clusters in our algorithm, hence increasing the cost in our cost function.
For code, see file WordCount.java.