10/20/2024

CS-483 Big Data Mining

Gokhulnath Thirumaran
675086474

1 a) Soln:-

Symmetry:-

If a matrix M is symmetric then

$M = M^T$

So,

MM^T is the product of matrix M and its traspose

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

Thus MM^T is symmetric

For M^TM

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

Thus M^TM is symmetric

Square:- It is meant to be square is

no. of rows = no. of cols.

So M is a $p \times q$ matrix

MM^T involves Multiplying $M_{p \times q} \times M^T_{q \times p}$

ie,

So MM^T is a $p \times p$ matrix

So its a square

For M^TM,

$M^T \Rightarrow q \times p$

$M \Rightarrow p \times q$

$M^TM \Rightarrow q \times q$

Thus its a square

Real

A matrix is real if all its entries are real no.

Assuming M, M^T have real no, So both MM^T & M^TM will be real.

1) b) Let v be the eigenvector of MM^T and $\lambda \neq 0$.

Then we have $MM^Tv = \lambda v$.

multiply M^T on both side,

$$M^TMM^Tv = M^T(\lambda v)$$

$$= \lambda M^Tv$$

$$M^TM(M^Tv) = \lambda(M^Tv)$$

$\Rightarrow M^Tv = u$

$$M^TMu = \lambda u$$

we see that u is the eigenvector of M^TM, $\lambda \neq 0$

Hence MM^T and M^TM have same eigen value.

but with different corresponding eigen vector, when

the eigenvector of $MM^T = D$

$M^TM = M^TD$.

1) c) By eigenvalue decomposition of a real, symmetric, square matrix

$B = \Theta \Lambda \Theta^T$, $\Lambda = \text{diag}(\lambda_1, \dots \lambda_d)$.

From 1a), what

M^TM is real, symmetric, square

so

$M^TM = \Theta \Lambda \Theta^T$, $\Lambda = \text{diag}(\lambda_1, \dots \lambda_d)$ that contains eigenvalue of M^TM along diagonal.

Θ is an ortogonal matrix.

1) d) U, V are column-orthonormal,

$\Rightarrow U^TU = I$

$V^TV = I$

Since Σ is a diagonal matrix

$\Sigma = \Sigma^T$.

Hence

$$M^TM = (U\Sigma V^T)^T (U\Sigma V^T) = V\Sigma^T U^T U \Sigma V^T$$
$$= V\Sigma^T\Sigma V^T$$
$$= V\Sigma^2 V^T$$

1) e)

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

$$\Sigma = \begin{bmatrix} 7.61577311 & 0 \\ 0 & 1.41421356 \end{bmatrix}$$

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

$$E\text{vals} = [58, 2]$$

$$E\text{vecs} = \begin{bmatrix} 0.70710678 & -0.70710678 \\ 0.70710678 & 0.70710678 \end{bmatrix}$$

The first col. of V is -1 times of first eigenvector, second col. of V is equal to second eigenvector

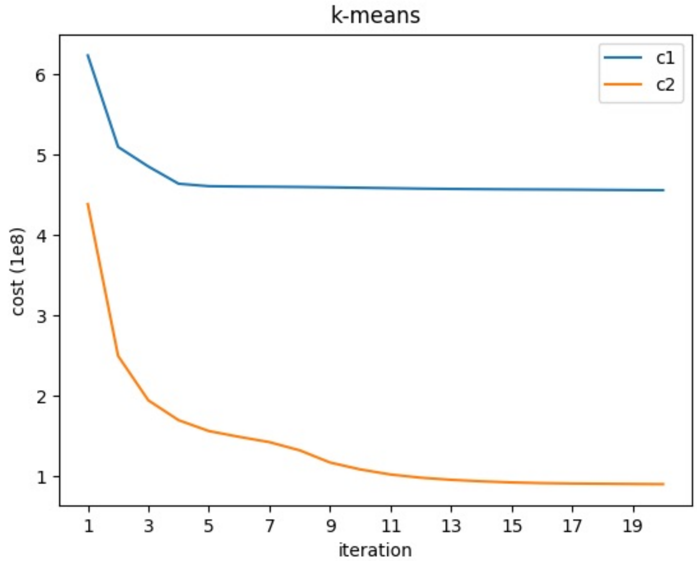
From 1d, we have $M^T M = V \Sigma^2 V^T$

$\Rightarrow M^T M V = V \Sigma^2 V^T V = V \Sigma^2$

So diag(Σ^2) are eigenvector of $M^T M$.

Hence, the singular values of M are square roots of eigenvalues of $M^T M$

2a) 1)



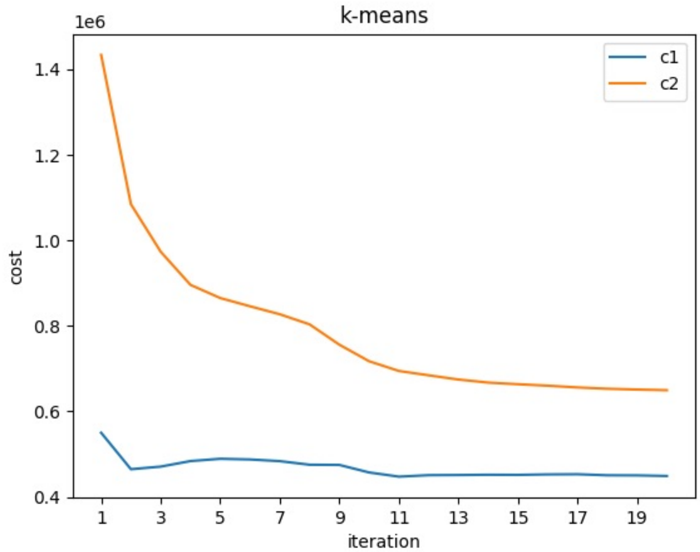
2) The % change of cost after 10 iteration

in C1.txt = 26.48 %.

C2.txt = 76.70 %.

So, c2 is better than c1 since the initial cluster are far apart and less overlap. Hence the cluster will be split less often, leading to better final results. K-means reduced the Euclidean distance between data and centroid, so the cost of C2 less than C1 mean c2 is better.

2b) 1)



2) Similarly, % change in cost after 10 iteration

in c1.txt = 18.65%.

c2.txt = 51.55%.

So c1 is better in terms of cost of Manhattan distance as c1 has lower final cost. Since the initial clusters centroid in c2 are as far as possible by using Euclidean dist, these centroids might not be the furthest in Manhattan distance.

3) a) By computing the derivative E to R_{iu}, q_i, P_u,

$$\frac{\partial E}{\partial R_{iu}} = \mathcal{E}_{iu} = 2\left(R_{iu} - q_i \cdot P_u^T\right) \quad -① $$

$$\frac{\partial E}{\partial q_i} = \nabla q_i = -2 * \left(R_{iu} - q_i \cdot P_u^T\right) * P_u + 2 * \lambda * q_i$$

$$= -\mathcal{E}_{iu} \cdot P_u + 2 * \lambda \cdot q_i \quad - ②$$

$$\frac{\partial E}{\partial P_u} = \nabla P_u = -2 * \left(R_{iu} - q_i \cdot P_u^T\right) * q_i + 2 * \lambda * P_u$$

$$= -\mathcal{E}_{iu} * q_i + 2 * \lambda * P_u \quad - ③$$

$$q_i := q_i - \eta * \nabla q_i \quad - ④$$

$$P_u := P_u - \eta * \nabla P_u \quad - ⑤$$

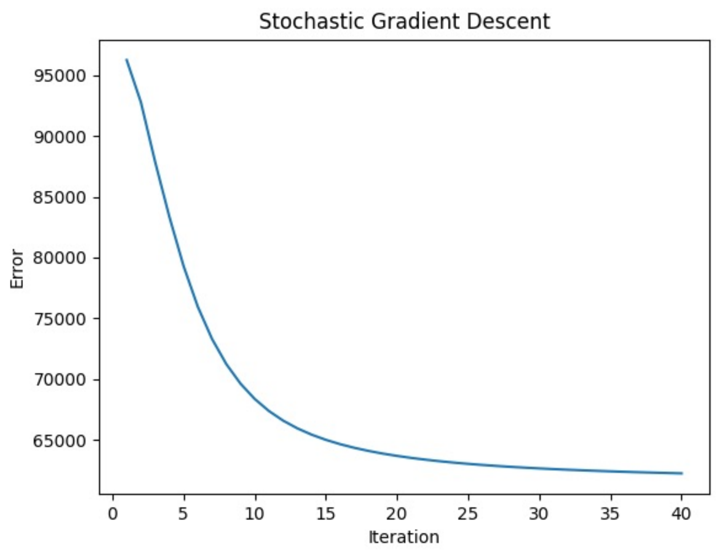
$$\therefore, \quad \mathcal{E}_{iu} = 2 * \left(R_{iu} - q_i * P_u^T\right) \quad - ⑥$$

from 2, 3, 4, 5,

$$q_i := q_i + \eta * \left(\mathcal{E}_{iu} \cdot P_u - 2 \cdot \lambda \cdot q_i\right)$$

$$P_u := P_u + \eta * \left(\mathcal{E}_{iu} * q_i - 2 * \lambda \cdot P_u\right)$$

3) b) $\eta = 0.03$.




```
Iteration 1, error E is: 96255.08133275602
Iteration 2, error E is: 92778.72451785424
Iteration 3, error E is: 87871.98522092946
Iteration 4, error E is: 83352.92883155937
Iteration 5, error E is: 79290.77441858343
Iteration 6, error E is: 75940.97880717664
Iteration 7, error E is: 73294.6033610033
Iteration 8, error E is: 71228.34724191221
Iteration 9, error E is: 69616.23752847518
Iteration 10, error E is: 68354.10253620414
Iteration 11, error E is: 67359.72524078845
Iteration 12, error E is: 66569.11457098607
Iteration 13, error E is: 65933.36103126415
Iteration 14, error E is: 65415.76799199314
Iteration 15, error E is: 64989.10827503597
Iteration 16, error E is: 64633.23071575126
Iteration 17, error E is: 64333.15531857526
Iteration 18, error E is: 64077.64433491214
Iteration 19, error E is: 63858.16589685365
Iteration 20, error E is: 63668.155309747235
Iteration 21, error E is: 63502.4918880767
Iteration 22, error E is: 63357.12799389091
Iteration 23, error E is: 63228.824303818925
Iteration 24, error E is: 63114.95903184191
Iteration 25, error E is: 63013.388816556406
Iteration 26, error E is: 62922.34595510617
Iteration 27, error E is: 62840.36144328099
Iteration 28, error E is: 62766.20653386887
Iteration 29, error E is: 62698.84774062323
Iteration 30, error E is: 62637.41172872021
Iteration 31, error E is: 62581.15757089633
Iteration 32, error E is: 62529.45456418178
Iteration 33, error E is: 62481.76429878175
Iteration 34, error E is: 62437.62601859701
Iteration 35, error E is: 62396.64455934846
Iteration 36, error E is: 62358.480327437435
Iteration 37, error E is: 62322.84091176391
Iteration 38, error E is: 62289.47401601981
Iteration 39, error E is: 62258.16147011504
Iteration 40, error E is: 62228.714132852714
```

4) a) Given $R_{ij} = 0$ or 1

$R^T_{ij} = R_{ji}$. we can compute T_{ii} and T_{ij}

$$T_{ii} = \sum_{k=1}^{n} R_{ik} \times R^T_{ki} = \sum_{k=1}^{n} R_{ik}^2$$

$R_{ik}^2 = 1$, if $R_{ik} = 1$

otherwise $R_{ik}^2 = 0$.

$$T_{ii} = \sum_{k=1}^{n} R_{ik}^2 = \sum_{k=1}^{n} R_{ik}$$

So T_{ii} means the number of item that user i likes, also equale to the node degree of user i

$$T_{ij}, i \neq j = \sum_{k=1}^{n} R_{ik} \times R^T_{kj}$$

$$= \sum_{k=1}^{n} R_{ik} \times R_{jk} \cdot R_{ik} \times R_{jk} = 1$$

$R_{ik} = R_{jk} = 1$
otherwise
$R_{ik} \times R_{jk} = 0$.

So $R_{ik} \times R_{jk} = 1$, if user i and j both like item k.

Thus $T_{ij}, i \neq j$ means the no. of items that both user i and user j like, also means the no. of path between user i and j

h) b) Let $R'^T_{ij} = \dfrac{R^T_{ij}}{\|R_i^T\|}$ $\rightarrow$ R_{ij}^T = R_{ij}^T in R^T. divided by length of its row.

$$\Rightarrow \|R_i^T\| = \sqrt{\sum_{j=1}^{m} (R^T_{ij})^2}$$

$$\frac{\sum_{k=1}^{m} R^T_{ik} R^T_{jk}}{\|R_i^T\| \, \|R_j^T\|} = \frac{\sum_{k=1}^{m} R^T_{ik} R^T_{jk}}{\sqrt{a_{ii}} \sqrt{a_{jj}}} = R_i'^T \cdot (R_j'^T)^T$$

4) c) i)

$$r'_{u,s} = \sum_{x \in users} \text{cos-sim}(x,u) \cdot R_{x,s} = \sum_{x \in users} (P^{-1/2} R R^T P^{-1/2})_{ux} \cdot R_{x,s}$$

$$= (P^{-1/2} R R^T P^{-1/2})_u \cdot R_s .$$

where $(P^{-1/2} R R^T P^{-1/2})_u$ is the u^{th} row vector of $(P^{-1/2} R R^T P^{-1/2})$, and R_s is the s^{th} col. vect of R.

Thus, recom of matrix $\Gamma = P^{-1/2} R R^T P^{-1/2} R$.

2) $$r_{u,s} = \sum_{x \in items} R_{u,s} \cdot \text{cos-sim}(x,s) = \sum R_{ux} \cdot (\Theta^{-1/2} R^T R \Theta^{-1/2})_{xs}$$

$$= R_u \cdot (\Theta^{-1/2} R^T R \Theta^{-1/2})_s$$

where $(\Theta^{-1/2} R^T \Theta^{-1/2})_s$ is the s^{th} col. vect of $(\Theta^{-1/2} R^T R \Theta^{-1/2})$, R_u is the row vector of R.

Thus the recom matrix $\Gamma = R\Theta^{-1/2} R^T R \Theta^{-1/2}$.

4)d

```
The names of five TV shows that have the highest similarity scores for Alex
for the user-user collaborative filtering are:

FOX 28 News at 10pm, with similarity score 908.4800534761279
Family Guy, with similarity score 861.17599928733
2009 NCAA Basketball Tournament, with similarity score 827.6012954743582
NBC 4 at Eleven, with similarity score 784.7819589039738
Two and a Half Men, with similarity score 757.6011181024228

----------------------------------------------------

The names of five TV shows that have the highest similarity scores for Alex
for the item-item collaborative filtering are:

FOX 28 News at 10pm, with similarity score 31.364701678342396
Family Guy, with similarity score 30.001141798877764
NBC 4 at Eleven, with similarity score 29.396797773402543
2009 NCAA Basketball Tournament, with similarity score 29.22700156150048
Access Hollywood, with similarity score 28.971277674055564
```