1 (a)

Both matrices are symmetric, square and real.

Symmetric: $(M^T M)^T = M^T (M^T)^T = M^T M$. $(MM^T)^T = (M^T)^T M^T = MM^T$

Square: $M^T M$ is size q x q. MM^T is size p x p.

Real: Both are real because M is.

1 (b)

For each eigenvalue of MM^T, we can write $MM^T v = v\lambda$. Multiplying both sides by a prefix M^T, we have $M^T M(M^T v) = (M^T v)\lambda$, where the parentheses can be added because of associativity. Therefore λ is also an eigenvalue of $M^T M$, with a corresponding eigenvector being $M^T v$. The eigenvectors are usually not the same. An exception is $M^T = I$.

1 (c)

We can write $M^T M = Q\Lambda Q^T$, because $M^T M$ is symmetric, square and real, by (a).

1 (d)

$M^T M = V\Sigma U^T U\Sigma V^T = V\Sigma^2 V^T$, which matches (c).

1 (e)

Code: 2_1_e.py

Result:

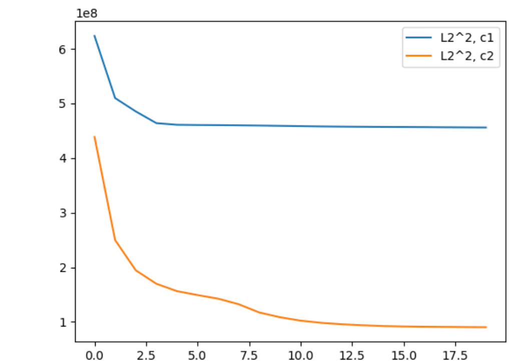
```
U
[[-0.27854301  0.5       ]
 [-0.27854301 -0.5       ]
 [-0.64993368  0.5       ]
 [-0.64993368 -0.5       ]]
Sigma
[7.61577311 1.41421356]
VT
[[-0.70710678 -0.70710678]
 [-0.70710678  0.70710678]]
Evals
[58.  2.]
Evecs
[[ 0.70710678 -0.70710678]
 [ 0.70710678  0.70710678]]
```

Columns of Evecs are the same as those of V, multiplied by ± 1.

Each singular value of M is the square root of an eigenvalues of $M^T M$.

Code: 2_2.py, 2_2_plot.py

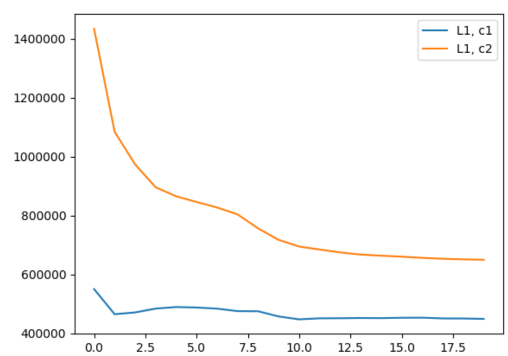
2 (a)



Initialization	**Percentage change in cost after 10 iterations**
c1.txt	26.5%
c2.txt	76.7%

For the Euclidean distance, random initialization is better, because it gives both a good start and a good room for optimization. A deeper reason is that when clusters are initialized far apart, true clusters are split less often, which leads to faster optimization and better results.

2 (b)



Initialization	**Percentage change in cost after 10 iterations**
c1.txt	18.7%
c2.txt	51.6%

For the Manhattan distance, random initialization doesn't appear to be better. It is because points in c2.txt are far apart in the Euclidean distance; they are not necessarily apart in the Manhattan distance.

3 (a)

The answers will be different, depending on where the brackets are drawn. We take

$$E = \left[\sum_{(i,u)\in ratings} (R_{iu} - q_i p_u^T)^2 \right] + \lambda \left(\sum_i \|q_i\|_2^2 + \sum_u \|p_u\|_2^2 \right)$$

so

$$\boxed{\varepsilon_{iu} = \frac{\partial E}{\partial R_{iu}} = 2(R_{iu} - q_i p_u^T)}$$

For SGD, because

$$\frac{\partial E}{\partial q_i} = \left[\sum_{\substack{u \\ (i,u)\in ratings}} 2(R_{iu} - q_i p_u^T)(-p_u) \right] + 2\lambda q_i$$

$$\frac{\partial E}{\partial p_u} = \left[\sum_{\substack{i \\ (i,u)\in ratings}} 2(R_{iu} - q_i p_u^T)(-q_i) \right] + 2\lambda p_u$$

we choose

$$q_i \coloneqq q_i + \eta[2(R_{iu} - q_i p_u^T)p_u - 2\lambda q_i]$$

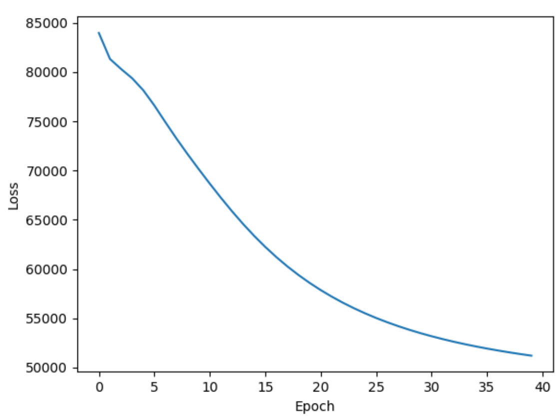
$$p_u \coloneqq p_u + \eta[2(R_{iu} - q_i p_u^T)q_i - 2\lambda p_u]$$

Note this corresponds to enclosing the regularization terms in the summation in the definition of E. But the provided regularization coefficient λ warrants it.

3 (b)

Code: 2_3_b.py

η = 0.01 is a good value. (0.05 and 0.001 give worse results.)



4 (a)

T_{ii}: The number of items that user i likes. I.e. the out-degree of user node i.

T_{ij}: The number of items that user i and j both like. I.e. the number of common neighbors of user i and j.

4 (b)

A diagonal matrix can scale the columns of its left multiplier, so to normalize items, we use $RQ^{-1/2}$. So, similar to (a), the item similarity matrix $S_I = (RQ^{-1/2})^T RQ^{-1/2} = Q^{-1/2}R^T RQ^{-1/2}$.

Similar to S_I, $S_U = (R^T P^{-1/2})^T R^T P^{-1/2} = P^{-1/2}RR^T P^{-1/2}$.

4 (c)

User-user collaborative filtering: $\Gamma_{us} = \sum_{x \in users} cos - sim(x, u) * R_{xs}$

so $\Gamma = S_U R = P^{-1/2}RR^T P^{-1/2}R$.

Item-item collaborative filter: $\Gamma_{us} = \sum_{x \in items} R_{ux} * cos - sim(x, s)$

So $\Gamma = RS_I = RQ^{-1/2}R^T RQ^{-1/2}$.

4 (d)

Code: 2_4_d.py

Take $i_{alex} = 499$ because of 0-indexing.

User-user collaborative filtering:

```
Similarity score    Name of show
908.4800534761278   "FOX 28 News at 10pm"
861.17599928733     "Family Guy"
827.601295474358    "2009 NCAA Basketball Tournament"
784.7819589039742   "NBC 4 at Eleven"
757.601118102423    "Two and a Half Men"
```

Movie-movie collaborative filtering:

```
Similarity score    Name of show
31.3647016783424    "FOX 28 News at 10pm"
30.001141798877754  "Family Guy"
29.396797773402547  "NBC 4 at Eleven"
29.22700156150048   "2009 NCAA Basketball Tournament"
28.971277674055553  "Access Hollywood"
```