1. Symmetric:

For $A = MM^T$: $\quad A^T = (MM^T)^T = (M^T)^T M^T = MM^T = A$

For $B = M^T M$: $\quad B^T = (M^T M)^T = M^T (M^T)^T = M^T M = B$

2. Square:

For M : $\quad$ Size $= m \times n$

For MM^T : $\quad$ Size $= m \times m$

For $M^T M$: $\quad$ Size $= n \times n$

3. Real:

If M is real, MM^T and $M^T M$ are real, since the sum or multiplication of real numbers is always real

Let M be an $m \times n$ matrix. Consider the matrices MM^T and M^TM. Suppose λ is a nonzero eigenvalue of M^TM with eigenvector $\mathbf{v}$, i.e., $M^TM\mathbf{v} = \lambda\mathbf{v}$. Multiply both sides by M: $M(M^TM\mathbf{v}) = \lambda M\mathbf{v} \Rightarrow (MM^T)(M\mathbf{v}) = \lambda(M\mathbf{v})$. Let $\mathbf{u} = M\mathbf{v}$, then $MM^T\mathbf{u} = \lambda\mathbf{u}$. Thus, λ is an eigenvalue of MM^T. Similarly, for nonzero eigenvalues of MM^T, we can show they are also eigenvalues of M^TM. Hence, the nonzero eigenvalues of MM^T and M^TM are the same. The eigenvectors, however, are different since MM^T and M^TM are $m \times m$ and $n \times n$ matrices, respectively.

$\mathrm{M}^T M = Q \Lambda Q^T$

The previous definition is fully applicable

Given the SVD decomposition $\mathrm{M} = \mathrm{U}\,\Sigma V^T,$ we want to express $M^T M$ in terms of $V, V^T,$ and Σ.

First, calculate $\mathrm{M}^T:\quad M^T = (U\Sigma V^T)^T = (V^T)^T\Sigma^T U^T = V\Sigma^T U^T.$

Now, find $\mathrm{M}^T M:\quad M^T M = (V\Sigma^T U^T)(U\Sigma V^T) = V\Sigma^T(U^T U)\Sigma V^T.$

Since U is an orthonormal matrix, $\mathrm{U}^T U = I,$ where I is the identity matrix. Thus:

$\mathrm{M}^T M = V\Sigma^T\Sigma V^T.$

Let $\Sigma^T\Sigma$ be denoted as Σ^2 since it results in a diagonal matrix with squared singular values. Hence:

$\mathrm{M}^T M = V\Sigma^2 V^T.$

```
Shape of M: (4, 2)
Shape of U: (4, 2)
Shape of Sigma: (2,)
Shape of Vt: (2, 2)

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]]
```

Figure 1: SVD decomposition. As we can see the vector of singular values is sorted from the most important value to the less important value

```
Shape of MTM: (2, 2)
Shape of Evals: (2,)
Shape of Evecs: (2, 2)
Eigenvalues (sorted): [58.  2.]
Eigenvectors (sorted):
 [[ 0.70710678 -0.70710678]
 [ 0.70710678  0.70710678]]
```

Figure 2: Eigen decomposition. As we can see the vector of eigenvalues is sorted from the most important value to the less important value

```
V (transpose of Vt):
 [[-0.70710678 -0.70710678]
 [-0.70710678  0.70710678]]
Evecs:
 [[ 0.70710678 -0.70710678]
 [ 0.70710678  0.70710678]]
```

Figure 3: The matrix V and the matrix Evecs are equal, a part from the sign of the first column. Anyway, the sign is irrelevant for eigen vectors, so we can say that those 2 matrices represents the same pair of eigenvalues, hence full correspondence

```
L1 norm of the difference between singular values squared and eigenvalues:  2.1760371282653068e-14
```

Figure 4: The L1 norm of the difference vector between the singular values squared and the eigenvectors is basically zero. This means that the square root of the eigenvalues correspond to the singular values

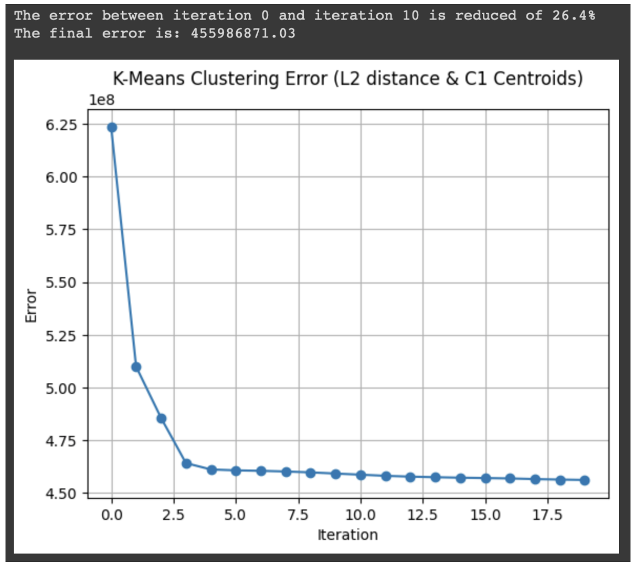

Figure 5: L2 distance and random initialization

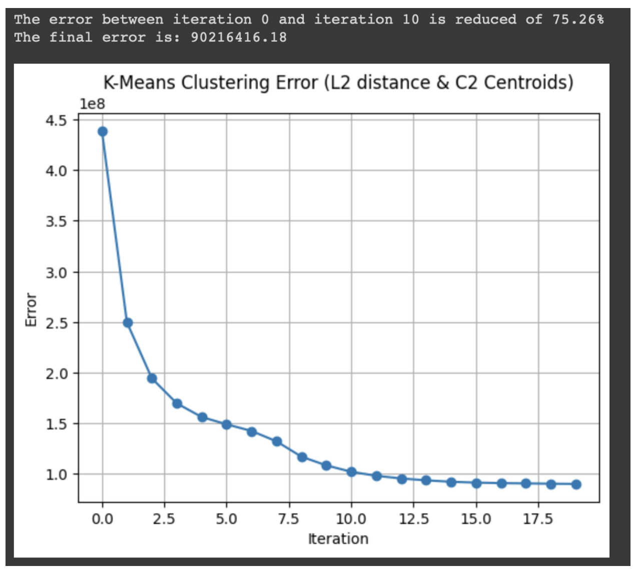

Figure 6: L2 distance and max distance initialization

In this case the error is decreasing more, after 10 iterations, with maximum distance initialization in C2.txt, 75.26%. In general, maximum distance initialization is better in this situation and it provides a smaller error at the end of the algorithm. That is because in K-means with L2 distance, initializing clusters far apart reduces the risk of poor local minima by ensuring that initial centroids cover diverse regions of the data space. This increases the chances of finding the global minimum and improves clustering quality by avoiding empty or overlapping clusters.

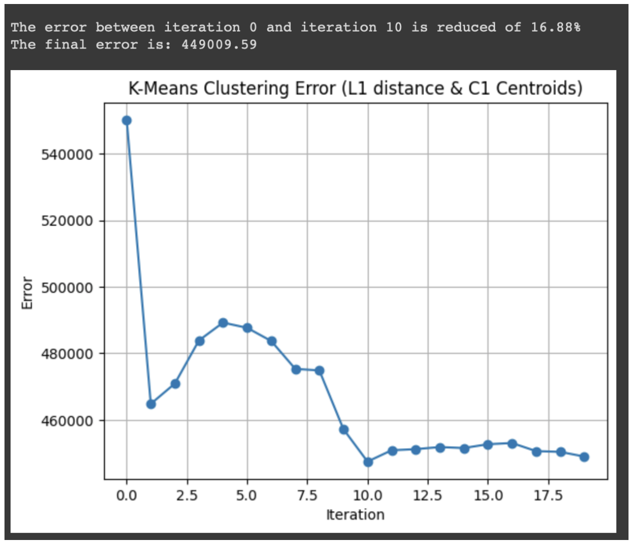

Figure 7: L1 distance and random initialization

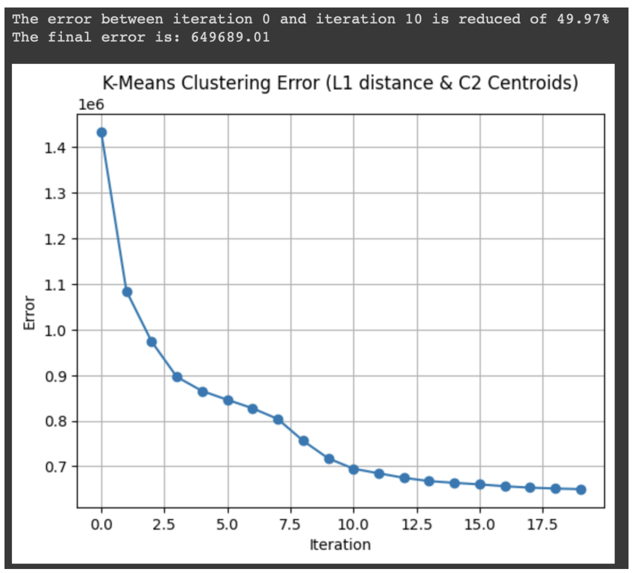

Figure 8: L1 distance and max distance initialization

In this case the error is decreasing more, after 10 iterations, with maximum distance initialization in C2.txt, 49.97%. In general, random initialization is better in this situation and it provides a smaller error at the end of the algorithm. That is because, for L1 distance, clusters tend to form around the median values of points rather than the mean, making them less sensitive to initial centroid placement. This reduces the impact of centroid initialization, so starting centroids far apart is not as crucial for preventing convergence to suboptimal solutions. Moreover, as expected, the error is non-monotonic and sometimes it increases, since K- means only ensures monotonic decrease of cost for squared Euclidean distance.

$$\nabla_{R_{iu}} E = 2\left(R_{iu} - q_i^T p_u\right) = \mathcal{E}_{iu}$$

$$\nabla_{q_{iu}} E = -2\left(R_{iu} - q_i^T p_u\right) \cdot p_u + 2\lambda q_i \implies q_i = q_i + \eta\left(\mathcal{E}_{iu} \cdot p_u - 2\lambda q_i\right)$$

$$\nabla_{p_u} E = -2\left(R_{iu} - q_i^T p_u\right) \cdot q_i + 2\lambda p_u \implies p_u = p_u + \eta\left(\mathcal{E}_{iu} \cdot q_i - 2\lambda p_u\right)$$

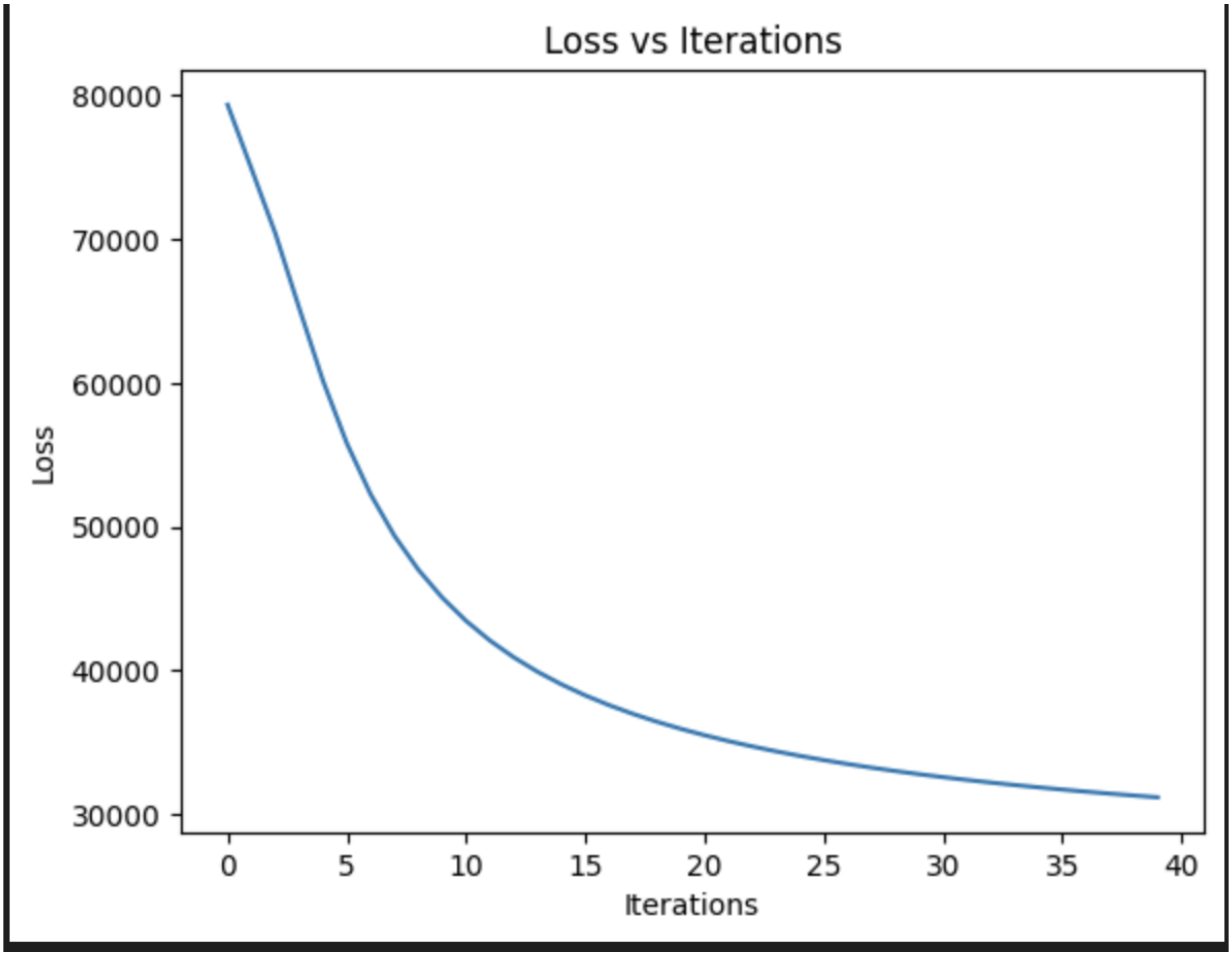

Figure 9: The best learning rate that I have found is $\eta = 0.01$

$\eta = 0.01$ seems to be the best trade-off for reaching a good point of convergence

Let $T = R \times R^T$ be the non-normalized user similarity matrix, where:

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

Interpretation of T_{ii}:

$$T_{ii} = \sum_{k=1}^{n} R_{ik} = \text{Number of items liked by user } i$$

Interpretation of T_{ij} (for $i \neq j$):

$$T_{ij} = \sum_{k=1}^{n} R_{ik} \times R_{jk} = \text{Number of common items liked by both users } i \text{ and } j$$

Bipartite Graph Interpretation:

- T_{ii}: Degree of user node i, i.e., number of items liked by user i.
- T_{ij} (for $i \neq j$): Number of 2-length paths between users i and j through a common item node.

Given the $m \times n$ matrix R and the $n \times n$ diagonal matrix Q, where $Q_{ii} = \sum_{k=1}^{m} R_{ki}^2$ (i.e., the degree of item i), the cosine similarity between items i and j is defined as:

$$\text{cosine similarity}(i, j) = \frac{R_{:i}^T R_{:j}}{\|R_{:i}\| \|R_{:j}\|}$$

where $R_{:i}$ and $R_{:j}$ are the i-th and j-th columns of R, and $\|R_{:i}\| = \sqrt{Q_{ii}}$, $\|R_{:j}\| = \sqrt{Q_{jj}}$. Thus:

$$\text{cosine similarity}(i, j) = \frac{R_{:i}^T R_{:j}}{\sqrt{Q_{ii} Q_{jj}}}$$

Now, define S_I as the item similarity matrix such that:

$$S_I = Q^{-1/2} R^T R Q^{-1/2}$$

where $Q^{-1/2}$ is a diagonal matrix with $Q_{ii}^{-1/2} = \frac{1}{\sqrt{Q_{ii}}}$. Then, the (i, j)-th element of S_I is:

$$(S_I)_{ij} = \left(Q^{-1/2} R^T R Q^{-1/2}\right)_{ij}$$

Given the $m \times n$ matrix R and the $m \times m$ diagonal matrix P, where $P_{ii} = \sum_{k=1}^{n} R_{ik}^2$ (i.e., the degree of user i), the cosine similarity between users i and j is defined as:

$$\text{cosine similarity}(i, j) = \frac{R_{i:} R_{j:}^T}{\|R_{i:}\| \|R_{j:}\|}$$

where $R_{i:}$ and $R_{j:}$ are the i-th and j-th rows of R, and $\|R_{i:}\| = \sqrt{P_{ii}}$, $\|R_{j:}\| = \sqrt{P_{jj}}$. Thus:

$$\text{cosine similarity}(i, j) = \frac{R_{i:} R_{j:}^T}{\sqrt{P_{ii} P_{jj}}}$$

Now, define the user similarity matrix S_U as:

$$S_U = P^{-1/2} R R^T P^{-1/2}$$

where $P^{-1/2}$ is a diagonal matrix with $P_{ii}^{-1/2} = \frac{1}{\sqrt{P_{ii}}}$. This matrix normalizes RR^T to represent the cosine similarity between users.

User-User Collaborative Filtering:

For user-user collaborative filtering, the recommendation score for user u and item s is given by:

$$r_{u,s} = \sum_{x \in \text{users}} \cos\text{-sim}(x, u) \cdot R_{xs}$$

where $\cos\text{-sim}(x, u)$ is the cosine similarity between users x and u, defined as:

$$S_U = P^{-1/2} R R^T P^{-1/2}$$

Thus, the recommendation matrix Γ is:

$$\Gamma = S_U R = \left(P^{-1/2} R R^T P^{-1/2}\right) R = P^{-1/2} R R^T P^{-1/2} R$$

Item-Item Collaborative Filtering:

For item-item collaborative filtering, the recommendation score for user u and item s is given by:

$$r_{u,s} = \sum_{x \in \text{items}} R_{ux} \cdot \cos\text{-sim}(x, s)$$

where $\cos\text{-sim}(x, s)$ is the cosine similarity between items x and s, defined as:

$$S_I = Q^{-1/2} R^T R Q^{-1/2}$$

Thus, the recommendation matrix Γ is:

$$\Gamma = R S_I = R\left(Q^{-1/2} R^T R Q^{-1/2}\right) = R Q^{-1/2} R^T R Q^{-1/2}$$

Final Results:

User-User Collaborative Filtering:

$$\Gamma = P^{-1/2} R R^T P^{-1/2} R$$

Item-Item Collaborative Filtering:

$$\Gamma = R Q^{-1/2} R^T R Q^{-1/2}$$

```
Top 5 shows by user-user collaborative filtering:
        FOX 28 News at 10pm, Family Guy, 2009 NCAA Basketball Tournament, NBC 4 at Eleven, Two and a Half Men

Top 5 shows by item-item collaborative filtering:
        FOX 28 News at 10pm, Family Guy, NBC 4 at Eleven, 2009 NCAA Basketball Tournament, Access Hollywood
```

Figure 10: Top 5 recommendations for the user Alex

As we can see, the top 2 recommendations are equal for both the systems, while the remaining 3 are either different or in a different order of priority

Information sheet
CS246: Mining Massive Data Sets

Assignment Submission Fill in and include this information sheet with each of your assignments. This page should be the last page of your submission. Assignments are due at 11:59pm and are always due on a Thursday. All students (SCPD and non-SCPD) must submit their homework via Gradescope (http://www.gradescope.com). Students can typeset or scan their homework. Make sure that you answer each (sub-)question on a separate page. That is, one answer per page regardless of the answer length. Students also need to upload their code on Gradescope. Put all the code for a single question into a single file and upload it.

Late Homework Policy Each student will have a total of *two* late periods. *Homework are due on Thursdays at 11:59pm PT and one late period expires on the following Monday at 11:59pm PT*. Only one late period may be used for an assignment. Any homework received after 11:59pm PT on the Monday following the homework due date will receive no credit. Once these late periods are exhausted, any assignments turned in late will receive no credit.

Honor Code We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down their solutions independently, i.e., each student must understand the solution well enough in order to reconstruct it by him/herself. 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 (GitHub/Google/previous year's solutions etc.) is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code very seriously and expect students to do the same.

Your name: Davide Ettori ______________________

Email: detto3@uic.edu ______________________ **SUID: 655693998**

Discussion Group: ______________________

I acknowledge and accept the Honor Code.

(Signed) Davide Ettori ______________________