CSE 547: Machine Learning for Big Data
Homework 2

Academic Integrity We take academic integrity extremely seriously. We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down the solutions and the code independently. In addition, each student should write down the set of people whom they interacted with.

Discussion Group (People with whom you discussed ideas used in your answers):

- *Hwai-Jin Peng*

On-line or hardcopy documents used as part of your answers:

- CSE 547 Lecture Slide (Tim Althoff)
- Collaborative Filtering for Implicit Feedback Datasets (Hu et. al.)
- Analytic solution for matrix factorization using alternating least squares (Alijah Ahmed)
- ALS Implicit Collaborative Filtering (Victor)

I acknowledge and accept the Academic Integrity clause.

(Signed) ***Cheng-Yen Yang***

Answer to Question 1(a)

(1)

For any matrix $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times d}$:

$$Tr(\mathbf{A}\mathbf{B}^T) = \sum_{i=1}^{n} \left(\mathbf{A}\mathbf{B}^T\right)_{ii} = \sum_{i=1}^{n}\sum_{j=1}^{d} \mathbf{A}_{nj}\mathbf{B}^T_{jn} = \sum_{j=1}^{d}\sum_{i=1}^{n} \mathbf{B}^T_{di}\mathbf{A}_{id} = \sum_{j=1}^{d} \left(\mathbf{B}^T\mathbf{A}\right)_{jj} = Tr(\mathbf{B}^T\mathbf{A}) \tag{1}$$

(2)

Since $\mathbf{\Sigma}$ is a symmetric matrix, it has a spectral decomposition as $\mathbf{Ssigma} = \mathbf{PDP}^{-1}$ where $\mathbf{D}$ is a diagonal matrix with eigenvalues of $\mathbf{\Sigma}$. Using the property $Tr(\mathbf{A}\mathbf{B}^T) = Tr(\mathbf{B}^T\mathbf{A})$, we have:

$$Tr(\mathbf{\Sigma}) = Tr(\mathbf{PDP}^{-1}) = Tr(\mathbf{P}^{-1}\mathbf{PD}) = Tr(\mathbf{D}) = \sum_{i=1}^{d} \lambda_i \tag{2}$$

and

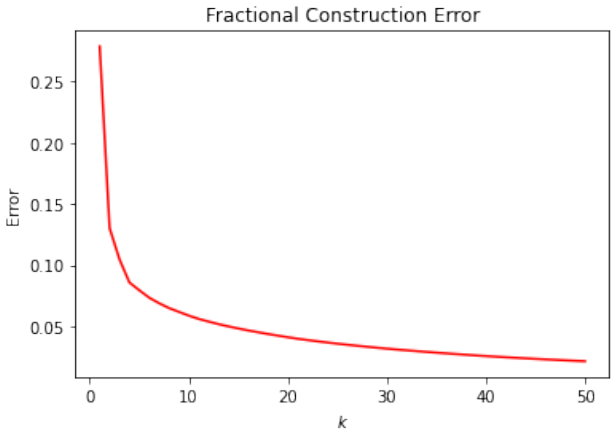
$$Tr(\mathbf{\Sigma}) = \frac{1}{n} Tr(\mathbf{X}^T\mathbf{X}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}^2 = \frac{1}{n}\sum_{i=1}^{n} \left(\sqrt{\sum_{j=1}^{n} |x_{ij}|^2}\right)^2 = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{X}_i||_2^2. \tag{3}$$

Answer to Question 1(b)

(1)

λ_1	λ_2	λ_{10}	λ_{30}	λ_{50}	Total
781.813	161.152	3.340	0.809	0.390	1084.207

(2)



(3)

The first eigenvalue λ_1 is correspond to the first eigenvector or eigenface in this case, i.e. the first principal component, which is the direction along which the data have the most variance. As stated in (1), the value of λ_1 is nearly $5\times$ the value of λ_2, which probably means that the first eigenface capture human face features that are most share between the data images.

Answer to Question 1(c)

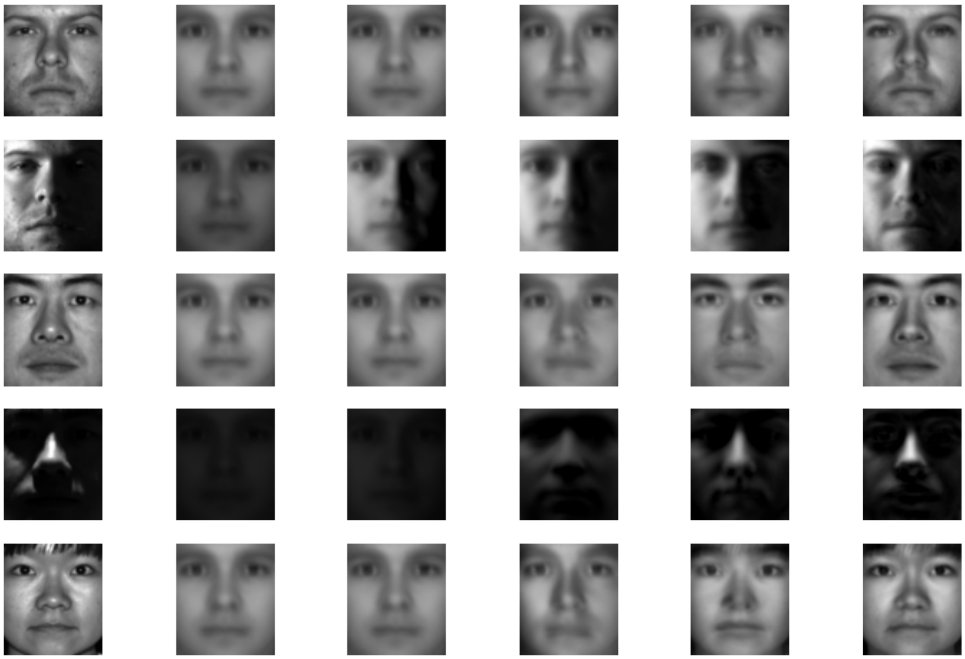
(1)

(2)

These are the first 10 eigenvalues as images, as we mentioned before, the first eigenface capture human face features that are most share between the data images, as the other eigenfaces each highlight a certain type of face features. Since our dataset consists of 38 individuals under 64 different lighting conditions such as lit from the top, front and side, therefore we can observe that the some of the eigenfaces actually represent these types of conditions. For example the second eigenface looks like those data images lit from the side.

Answer to Question 1(d)

(1)

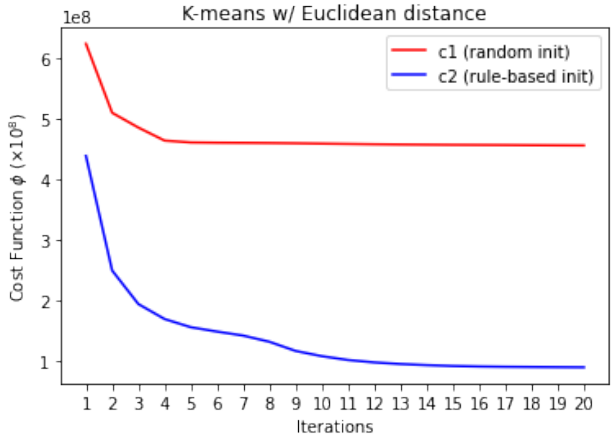


(2)

As we had mentioned, the first eigenface capture human face features that are most share between the data images. As you can see, the second column are the image constructing using $k = 1$, so they share a pretty similair reconstruction across different images. Meanwhile, the second eigenface looks like the main basis of those data images lit from the side. Therefore we can see that in the third column where constructing using $k = 2$, the image on the second row already shows a great reconstruction on the lighting condition as its original image held. Also, the reconstructions preserve more details of the original images as we increase the value of k.

Answer to Question 2(a)

(1)



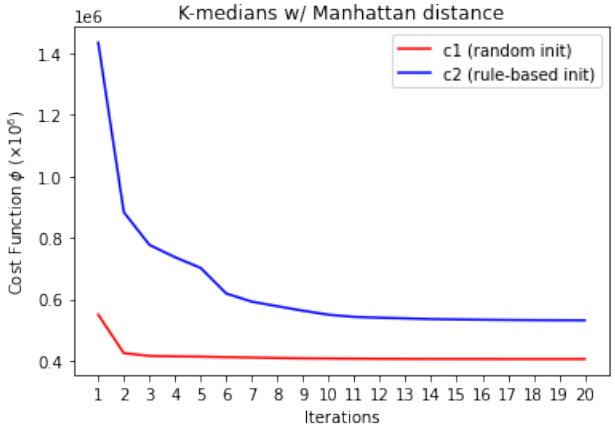
(2)

Initialization Method	Percentage Changes after 10 Iterations
Random Init (c1)	26.5%
Rule-based Init (c2)	76.7%

We can observe that the percentage change of cost function after 10 iterations of $k-means$ using random initialization is lesser than rule-based initialization due to the fact that the final clusters may be more likely to not be that apart. $k-means$ algorithm using rule-based initialization also results in better selections of centroids in comparison to the random initialization as it preserve smaller cost in term of Euclidean distance.

Answer to Question 2(b)

(1)



(2)

Initialization Method	Percentage Changes after 10 Iterations
Random Init (c1)	26.0%
Rule-based Init (c2)	62.1%

We can observe that the percentage change of cost function after 10 iterations of $k-medians$ using random initialization is lesser than rule-based initialization. However, $k-medians$ algorithm using rule-based initialization seems to lead to poorer result in term of cost in comparison to random initialization, which may due to the process of selecting the initial centroids by maximizing the Euclidean distance of the centroids.

Answer to Question 3(a)

According to what Hu et. al., $p_{ui} = 0$ is associated with relatively low confidence as user not taking any positive action on the item. However, for a user to not take action on a certain item can resulted from multiple reasons aside from simply not liking it. Therefore a measure of confidence $c_{ui} = 1 + \alpha r_{ui}$ was designed to represent different levels of confidence among all user-item pairs as well as a minimal value of $c_{ui} = 1$ for any $r_{ui} = 0$.

Answer to Question 3(b)

We want to minimize the cost function:

$$C(\mathbf{X}, \mathbf{Y}) = \sum_{u,i \in \mathcal{U} \times \mathcal{I}} c_{ui} \big(p_{ui} - \mathbf{x}_u^T \mathbf{y}_i\big)^2 + \lambda \Big(\sum_u ||\mathbf{x}_u||^2 + \sum_i ||\mathbf{y}_i||^2 \Big) \tag{4}$$

where the user matrix $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_m]^T \in \mathbb{R}^{m \times f}$ and the item matrix $\mathbf{Y} = [\mathbf{y}_1 \cdots \mathbf{y}_n]^T \in \mathbb{R}^{n \times f}$.

By differentiating with respect to $\mathbf{x}_u$ we obtained:

$$\frac{\partial}{\partial \mathbf{x}_u} C(\mathbf{X}, \mathbf{Y}) = -2 \sum_{i \in \mathcal{I}} c_{ui} \big(p_{ui} - \mathbf{x}_u^T \mathbf{y}_i\big) \mathbf{y}_i + 2\lambda \mathbf{x}_u \tag{5}$$

$$= -2 \sum_{i \in \mathcal{I}} c_{ui} \big(p_{ui} - \mathbf{y}_i^T \mathbf{x}_u\big) \mathbf{y}_i + 2\lambda \mathbf{x}_u \tag{6}$$

since $\mathbf{y}_i^T \mathbf{x}_u = \mathbf{x}_u^T \mathbf{y}_i$ as they are both real vectors which give identical scalar product.

Then since we can treat $\mathbf{y}_i^T$ as the i row in matrix $\mathbf{Y}$, we can derive the equation into:

$$\frac{\partial}{\partial \mathbf{x}_u} C(\mathbf{X}, \mathbf{Y}) = -2\mathbf{Y}^T \Big(\mathbf{C}_u \big(\mathbf{p}_u\big) - \sum_{i \in \mathcal{I}} \mathbf{y}_i^T \mathbf{x}_u \big) \Big) + 2\lambda \mathbf{x}_u \tag{7}$$

$$= -2\mathbf{Y}^T \Big(\mathbf{C}_u \big(\mathbf{p}_u - \mathbf{Y}\mathbf{x}_u\big) \Big) + 2\lambda \mathbf{x}_u \tag{8}$$

$$= -2 \Big(\mathbf{Y}^T \mathbf{C}_u \mathbf{p}_u - \mathbf{Y}^T \mathbf{C}_u \mathbf{Y} \mathbf{x}_u - \lambda \mathbf{x}_u \Big) \tag{9}$$

where $C_u = diag(c_{u1}, \cdots, c_{un}) \in \mathbb{R}^{n \times n}$ and $\mathbf{p}_u = [p_{u1} \cdots p_{un}]^T \in \mathbb{R}^n$.

Finally, by setting the differentiation to 0, we have the optimal $\mathbf{x}_u$:

$$\big(\mathbf{Y}^T \mathbf{C}_u \mathbf{Y} + \lambda \mathbf{I}\big) \mathbf{x}_u = \mathbf{Y}^T \mathbf{C}_u \mathbf{p}_u \tag{10}$$

$$\mathbf{x}_u = \big(\mathbf{Y}^T \mathbf{C}_u \mathbf{Y} + \lambda \mathbf{I}\big)^{-1} \mathbf{Y}^T \mathbf{C}_u \mathbf{p}_u \tag{11}$$

Answer to Question 3(c)

For every $\mathbf{Y}^T(\mathbf{C}^u - \mathbf{I})\mathbf{Y}$, a $f \times f$ matrix, there are only n_u non-zero elements in $\mathbf{C}^u - \mathbf{I}$. Therefore the time complexity will be $O(f^2 n_u)$ for computing $\mathbf{Y}^T(\mathbf{C}^u - \mathbf{I})\mathbf{Y}$ term for a single user u.

Answer to Question 3(d)

Recall that

$$\mathbf{x}_u = \left(\mathbf{Y}^T\mathbf{C}_u\mathbf{Y} + \lambda\mathbf{I}\right)^{-1}\mathbf{Y}^T\mathbf{C}_u\mathbf{p}_u \tag{12}$$

and it takes $O(f^2 n_u)$ for computing $\mathbf{Y}^T(\mathbf{C}^u - \mathbf{I})\mathbf{Y}$ term for a single user u along with $O(f^3)$ for matrix inversion.

Therefore the time complexity for updating $\mathbf{X}$ will be $\sum_{u\in\mathcal{U}} O(f^2 n_u + f^3) = O(f^2 N + f^3 n)$.

Answer to Question 3(e)

(1) Sparsity ratio α

$$\alpha_{small} = 0.0272, \quad \alpha_{synthetic} = 0.2429 \tag{13}$$

(2) Value of $\hat{p}_{30,83}$ for first 10 iterations

Iteration(s)	$\hat{p}_{30,83}^{small}$	$\hat{p}_{30,83}^{synthetic}$
1	0.6554	0.2768
2	0.6406	0.2398
3	1.0613	0.2385
4	0.5469	0.2386
5	0.5189	0.2496
6	0.4885	0.2385
7	0.4395	0.2336
8	0.3807	0.2323
9	0.3301	0.2334
10	0.2950	0.2361

(3) Final Value of $\hat{p}_{30,83}$

$$\hat{p}_{30,83}^{small} = 0.1812, \quad \hat{p}_{30,83}^{synthetic} = 0.3468 \tag{14}$$

(4) Plots of $C_{implicit}$