CSE 547: Machine Learning for Big Data
Homework 2

January Shen
April 30, 2019

Answer to Question 1(a)

see scanned part.

Answer to Question 1(b)

1(b)-1

sum of eigenvalues is 1084.2074349947675

$\lambda_1 = 781.8126992600016$
$\lambda_2 = 161.15157496732692$
$\lambda_{10} = 3.339586754887817$
$\lambda_{30} = 0.8090877903777284$
$\lambda_{50} = 0.38957773951814617$

1(b)-2

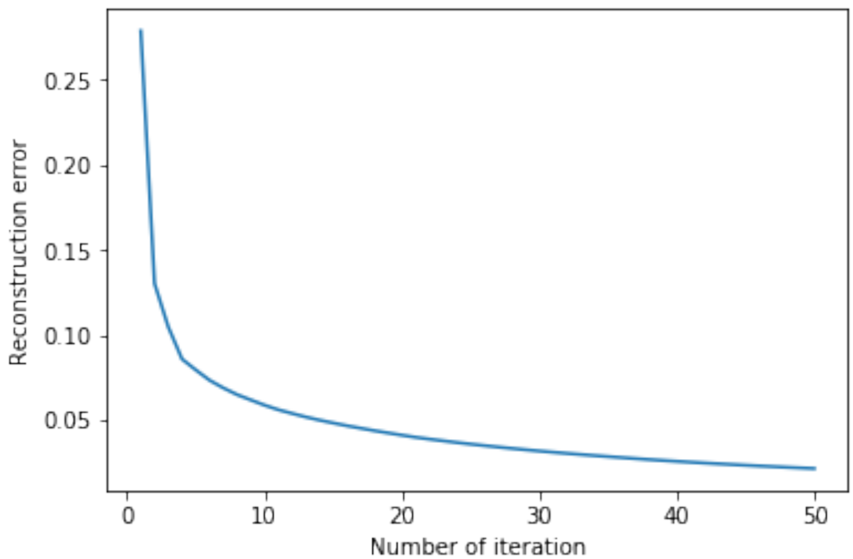

Figure 1: Figure for question 1 (b2)

1(b)-3

The principle eigenvalue captures the major theme of the pictures, which is the contour of a face. This feature is shared by all images. Other features, such as the shape of eyes and eyebrows, are less commonly shared so the eigenvalues are smaller.

Answer to Question 1(c)

1(c)-1

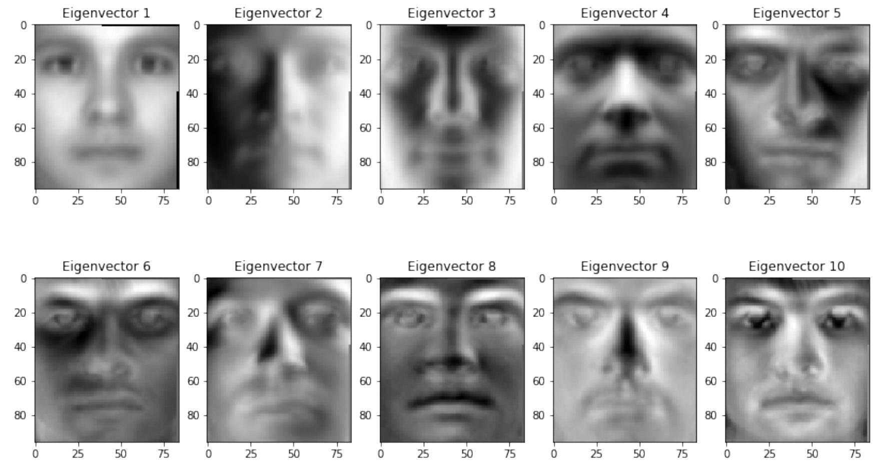

Figure 2: Figure for question 1 (c1)

1(c)-2

1: blurred image of a face
2: contour of a face with light from the right
3: contour of a face with light from the back
4: contour of a face with light from the front
5: contour of a face with light from the left
6: contour of a face with light from the top
7: contour of a face with lighter scale
8: contour of a face with darker scale
9: contour of a face with lighter scale
10: blurred image of a face

Answer to Question 1(d)

1(d)-1

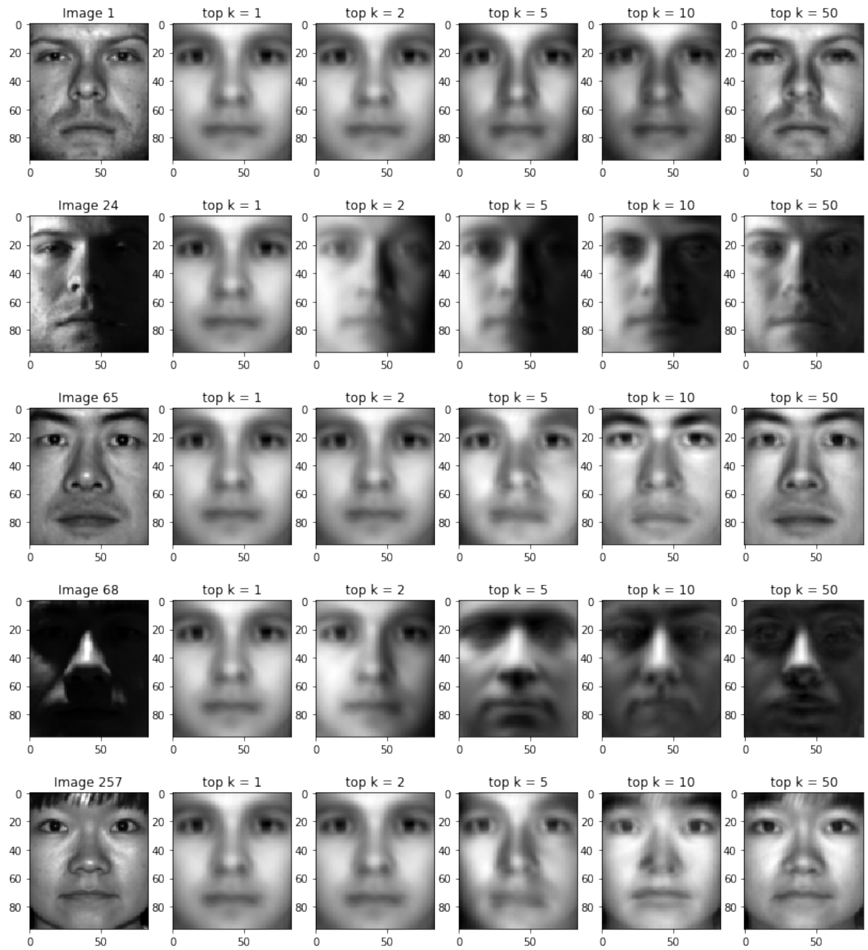

Figure 3: Figure for question 1 (d1)

1(d)-2

The more eigenvectors are applied, the clearer the image is. When we apply for more eigenvectors, the reconstruction error becomes less, so the reconstructed image becomes more like the original image.

Answer to Question 2(a)

2(a)-1

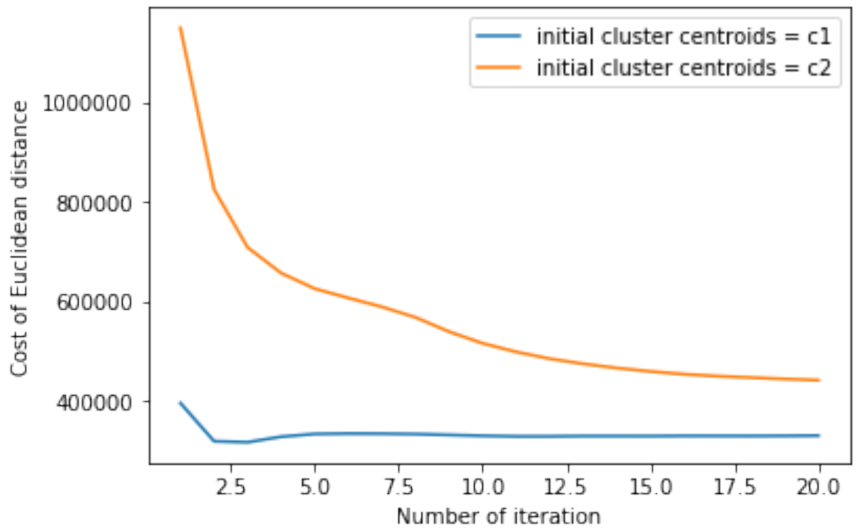

Figure 4: Figure for question 2 (a1)

2(a)-2

The cost change for Euclidean distance after 10 iterations is 20% for c1, and 123% for c2. The randomly chosen c1 has a lower cost at the beginning, but the cost in each iteration bounces back and forth sometimes. c2 has each dot positioned at the farthest distance. The initial setting may be far away from optimal, but it ensures improvement in each iteration.

Answer to Question 2(b)

2(b)-1

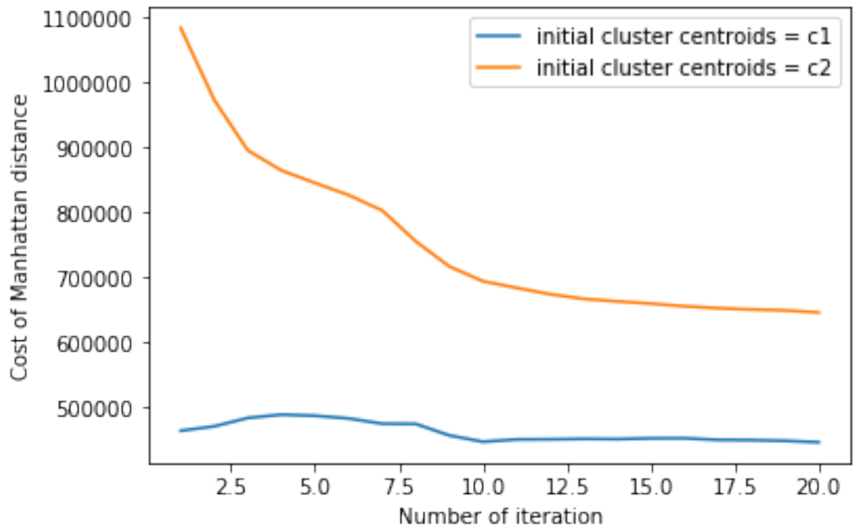

Figure 5: Figure for question 2 (b1)

2(b)-2

The cost change for Manhattan distance after 10 iterations is 5% for c1, and 28% for c2. The randomly chosen c1 has a lower cost at the beginning, but the cost in each iteration bounces back and forth sometimes. c2 has each dot positioned at the farthest distance. The initial setting may be far away from optimal, but it ensures improvement in each iteration.

Answer to Question 3(a)

see scanned part.

Answer to Question 3(b)

The learning rate here is 0.01.

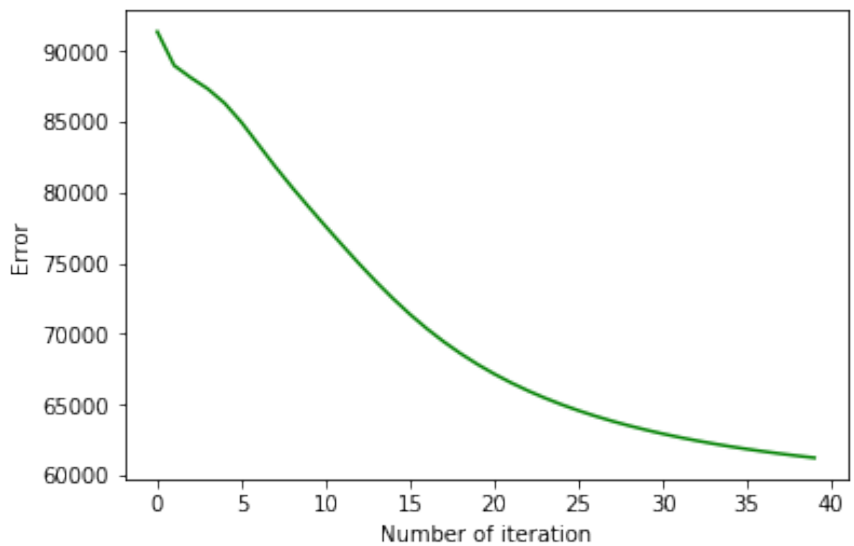

Figure 6: Figure for question 3 (b)

Answer to Question 4(a)

see scanned part.

Answer to Question 4(b)

see scanned part.

Answer to Question 4(c)

see scanned part.

Answer to Question 4(d)

User-User recommendation:
"FOX 28 News at 10pm", Similarity = 908.48
"Family Guy", Similarity = 861.18
"2009 NCAA Basketball Tournament", Similarity = 827.60
"NBC 4 at Eleven", Similarity = 784.78
"Two and a Half Men", Similarity = 757.60

Item-item recommendation:
"FOX 28 News at 10pm", Similarity = 31.36
"Family Guy", Similarity = 30.00
"NBC 4 at Eleven", Similarity = 29.40
"2009 NCAA Basketball Tournament", Similarity = 29.23
"Access Hollywood", Similarity = 28.97

HW2 January Shen

P1a-1. $Tr(AB^T) = \sum_{i=1}^{n} A_i \cdot B_i^T = \sum_{i=1}^{n} B_i^T \cdot A_i = Tr(B^TA)$ *

Let A_i = i-th row of A

P1a-2 $Tr(\Sigma) = Tr(\frac{1}{n} X^T \cdot X) = \frac{1}{n} \sum_{i=1}^{n} X_i^2 = \frac{1}{n} \sum_{i=1}^{n} \|X_i\|_2^2$ *

$= Tr(\frac{1}{n} X \cdot X^T)$

$Tr(\Sigma) = Tr(\frac{1}{n} X \cdot X^T)$ = sum of the eigenvalues.

$\because$ dimension of $Tr(\frac{1}{n} X \cdot X^T) = d \times d$

$\therefore Tr(\frac{1}{n} X \cdot X^T) = \sum_{i=1}^{d} \lambda_i$ *

P3 a

$\varepsilon_{iu} = 2(r_{iu} - q_i \cdot p_u)$

$q_i = q_i + \eta(\varepsilon_{iu} \cdot p_u - 2\lambda \cdot q_i)$

$p_u = p_u + \eta(\varepsilon_{iu} \cdot q_i - 2\lambda \cdot p_u)$

P4 a.

$T = R \cdot R^T$

$T_{ii} = P_{ii}$ = degree of user node i. How many items that user i likes.

$T_{ij}, i \neq j$ = the number of items that both i & j like.

P4 b.

$R^T \cdot R$ = matrix of item i · item j

Let C_i = i-th column of R, $M_{ij} = C_i \cdot C_j$, $i, j \in [1, n]$

Let $R^T \cdot R = M$

$Q_{rc}^{-\frac{1}{2}} = \frac{1}{\sqrt{Q_{rc}}}$, since Q is a diagonal matrix, $\frac{1}{\sqrt{Q_i}} = \frac{1}{\|Q_i\|_2}$.

Let Q_i = i-th column of Q

Let $Q^{-\frac{1}{2}} \cdot R^T \cdot R \cdot Q^{-\frac{1}{2}} = Z$, then $Z_{ij} = \frac{C_i \cdot C_j}{\|C_i\|_2 \|C_j\|_2}$, $Z = S_I$ *

Scanned with CamScanner

HW2-part 2 January Shen

p4b. (cont.)

Let $S_u = U$, $U_{ij} = \frac{User_i \cdot User_j}{\|User_i\| \cdot \|User_j\|}$

$R \cdot R^T$ = matrix of $user_i \cdot user_j \overset{\text{Let}}{=} M$.

Let C_i = ith row of R., $M_{ij} = C_i \cdot C_j$

Since P is a diagonal matrix, $P_i^{-\frac{1}{2}} = \frac{1}{\sqrt{P_i}}$, let P_i = ith row of P

$\Rightarrow$ user similarity matrix $S_u = P^{-\frac{1}{2}} \cdot R \cdot R^T \cdot P^{-\frac{1}{2}}$ ※

P4C.

Let Γ_I = the recommendation matrix for item-item case.

$$\Gamma_I = R \cdot \underbrace{Q^{-\frac{1}{2}} \cdot R^T \cdot R \cdot Q^{-\frac{1}{2}}}_{\substack{\text{cosine similarity} \\ \text{of items}}}$$

cosine similarity of items, where the ith column means every other item's similarity to item-i

Let Γ_{IK} = k-th row of Γ_I.

Γ_{IK} means that k-user's potential preference of each item.

$\Gamma_{IK} = R_K \cdot$ cosine similarity of items.

$\hookrightarrow$ R's kth row

Γ_{Iij} = user i's score of item j based on i's preference in R. ※

Similarly, Let Γ_U = the recomendation matrix for user-user case.

$\Gamma_U = \underbrace{P^{-\frac{1}{2}} \cdot R \cdot R^T \cdot P^{-\frac{1}{2}}}_{\substack{\text{cosine similarity} \\ \text{of users}}} \cdot R$, Γ_{Uij} = score of item j based on everyother user's rating on j and user i's prefercence compared to other users. ※