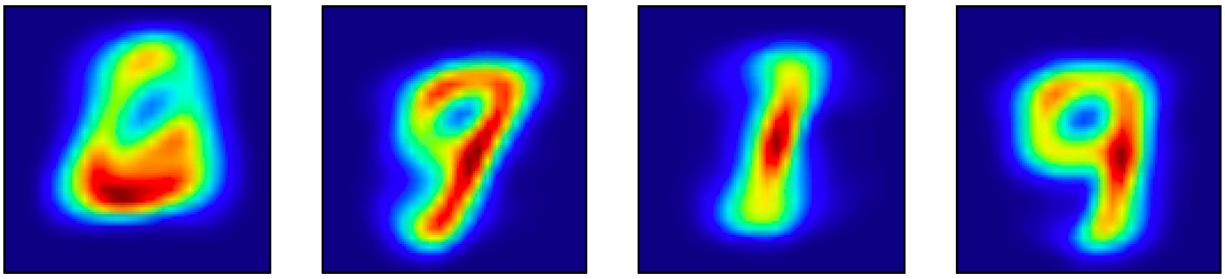
CS 189/289 Introduction to Machine Learning
Homework 7: Unsupervised
Shuhui Huang SID:3032129712

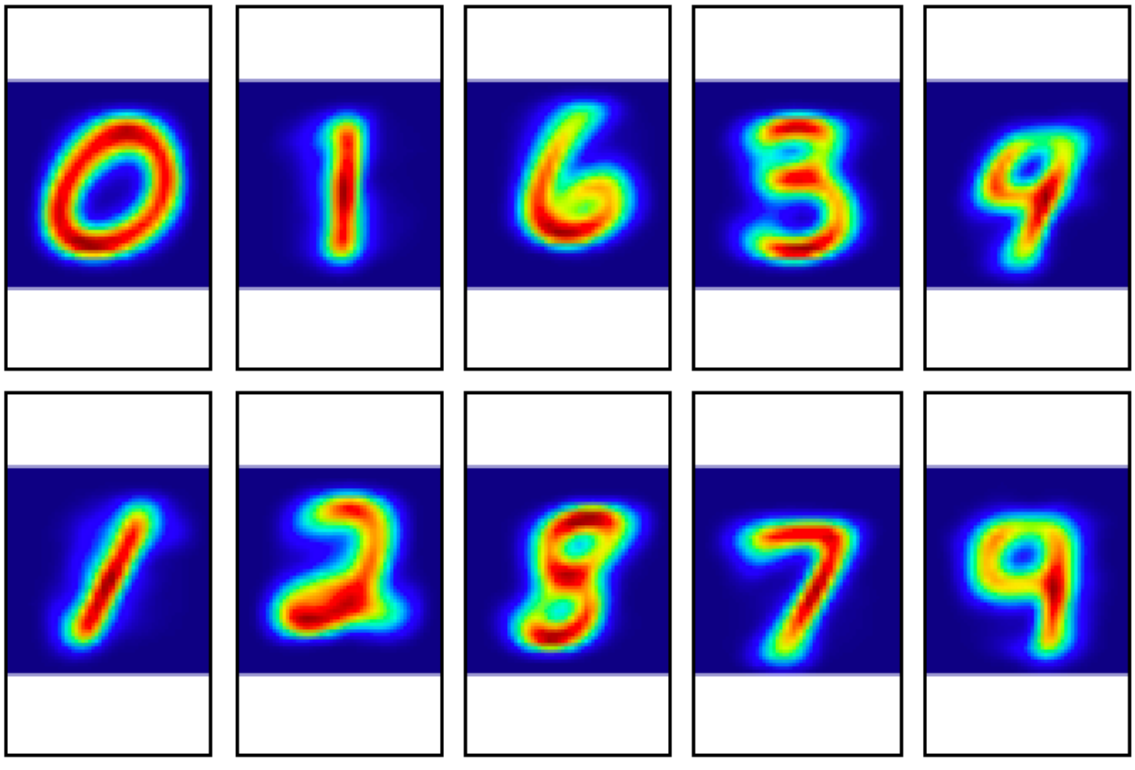
1. k-means clustering

(a) I implement Lloyd's algorithm for solving the k-means objective and the code has been attached in appendix. In my implementation, I initialize the K clusters' centers in "kmeans++ initialization scheme" with K=5, 10, 20 are plotted as follows:

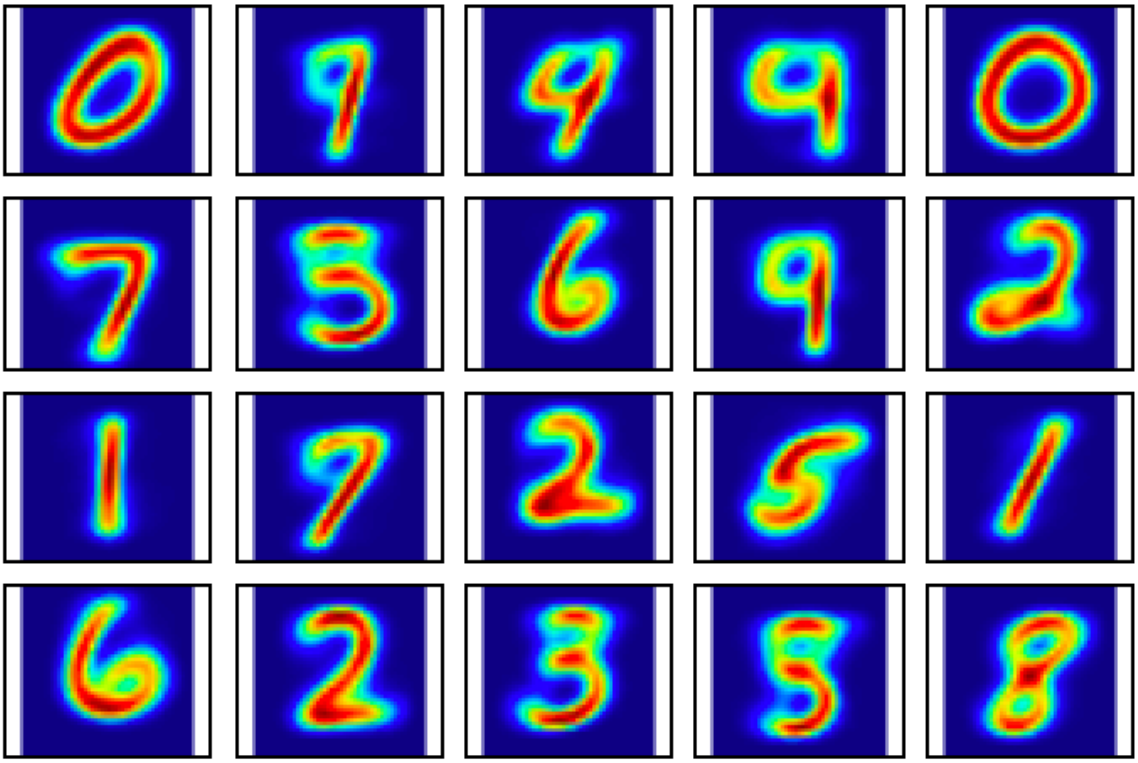
K=5:



K=10:



K=20:



The differences between results with different numbers of cluster centers is that (1) number of pictures in results will be different (2) with cluster centers increase, the same digit is more likely to repeat.

2. Low-Rank Approximation

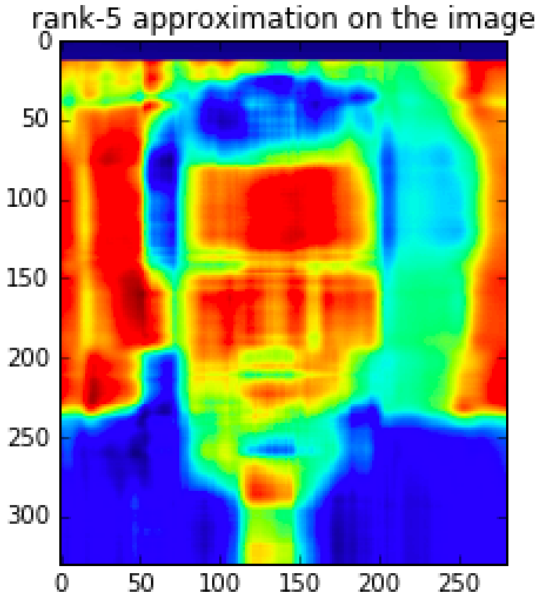
(a).

Let D=U $\sum V^T$, $\sum = diag(\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_m)$. So we want to get

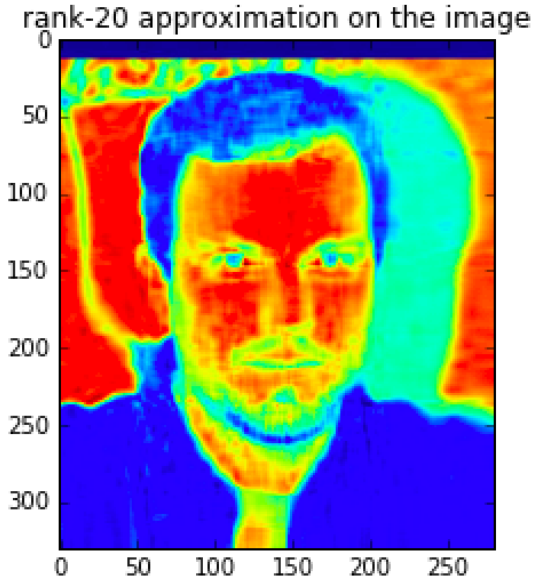
$\min(\text{rank} \leq \text{r})||\text{D} - \widehat{\text{D}}||_F = \sqrt{\sigma_r + \sigma_{r+1} + \cdots + \sigma_m}$

My code has been attached in the appendix, and rank-5, rank-20, and rank-100 approximation on the image I got is:

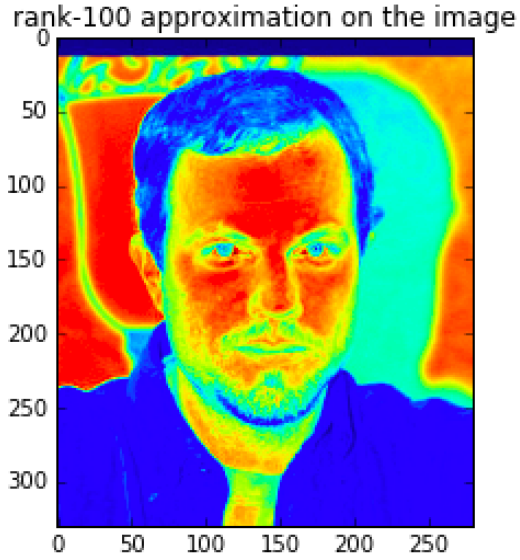
Rank=5:



Rank=20:

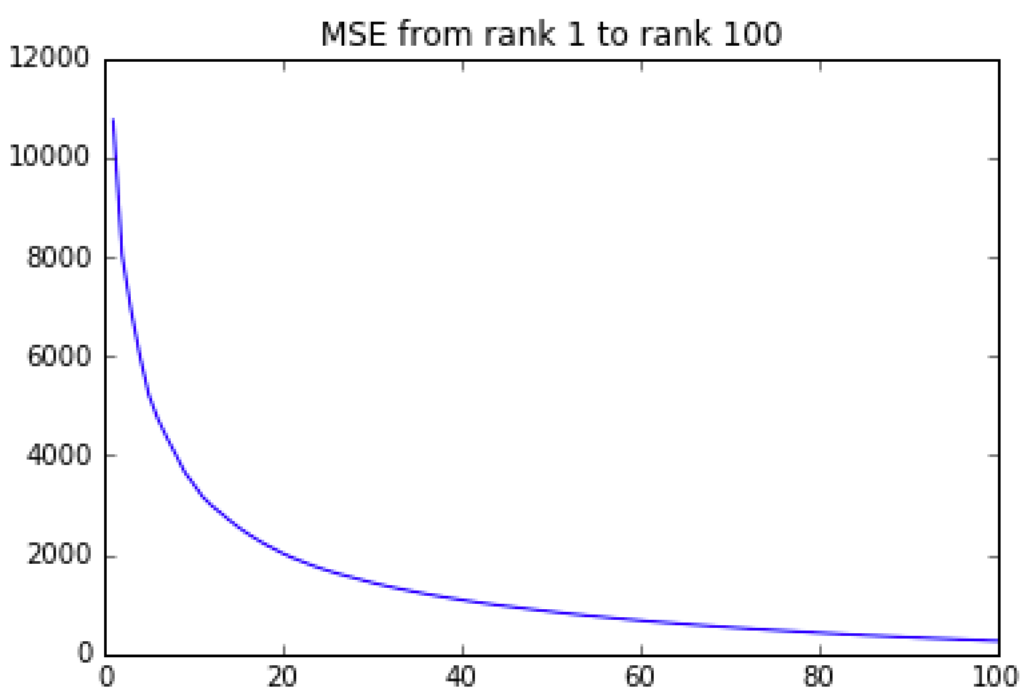


Rank=100:



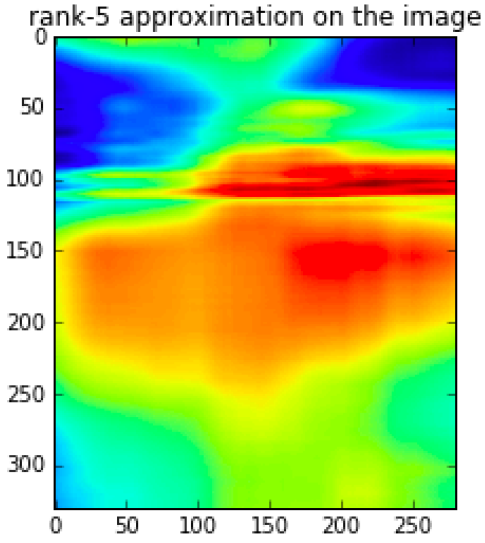
(b).

I use sum of squares of all entries in the matrix followed by a square root as mse, and my code has been attached in the appendix. The plot I got is :

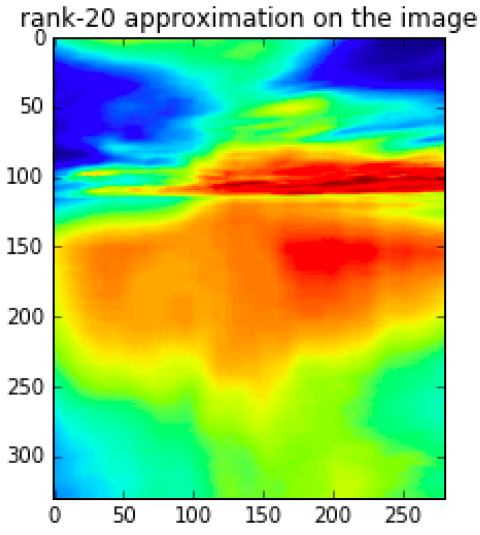


(c) The plot for sky is:

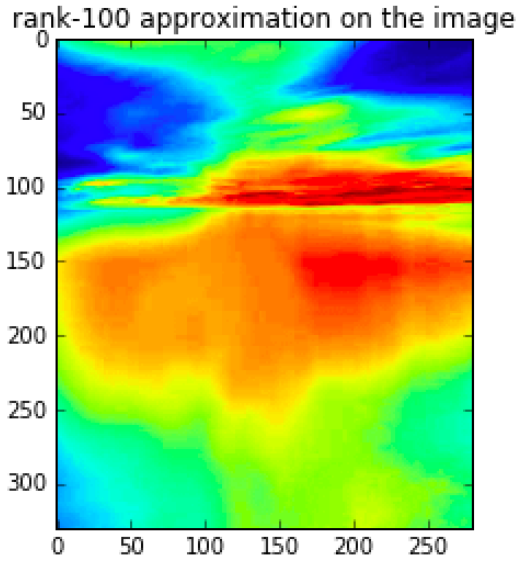
rank=5:



rank=20:



Rank=100:



(d) The lowest rank approximation for face is 50, for sky is 15. The possible reason for such difference is that there are higher differences among pixels in face image, or say, when choose the same rank, face image will lose more information compared with sky image, so the lowest rank for sky is smaller than face. And in matrix, we can say the difference among eigenvalues is higher in sky, so the first 15 biggest eigenvalues and its eigenvectors contains most information, while in face, the first 50 biggest eigenvalues and its eigenvectors contains most information.

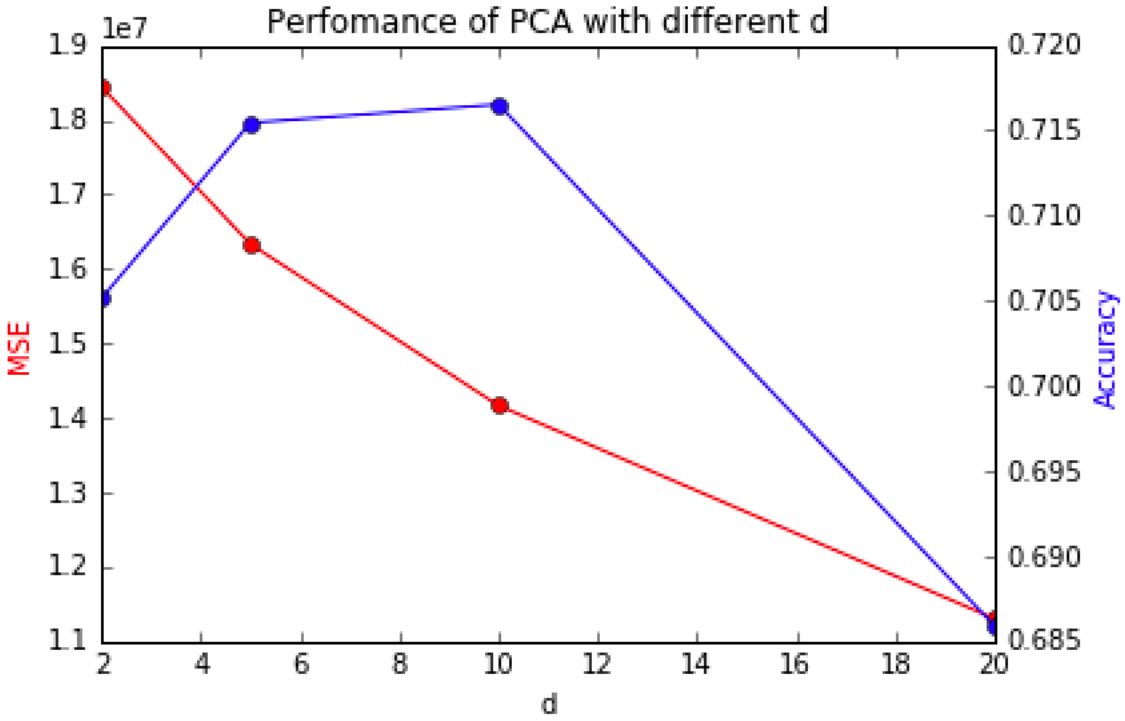
3. Joke Recommender System

3.2 Latent Factor Model

(a). After replacing all missing values with 0 and using PCA to learn the vector representation, I get the following MSEs and prediction accuracies with different value of d (2,5,10,20):

MSE:
[18441623.01788158, 16333384.42019682, 14165432.75799964, 11304007.439729325]

Prediction accuracies:
[0.70514905149051488, 0.71544715447154472, 0.71653116531165306, 0.68590785907859075]



with larger d, MSE will become smaller and smaller, but prediction accuracy might decrease since if d is too large we may over fit the model.

(b) minimize the MSE only on rated joke

If we minimize MSE only on rated joke:

If V_j is fixed, let $\frac{\partial L}{\partial u_i} = 0$, so we can have:

$$u_i = (\sum_{(i,j)\in s} R_{ij} V_j^T)(\sum_{(i,j)\in s} V_j V_j^T + \lambda I)^{-1}$$

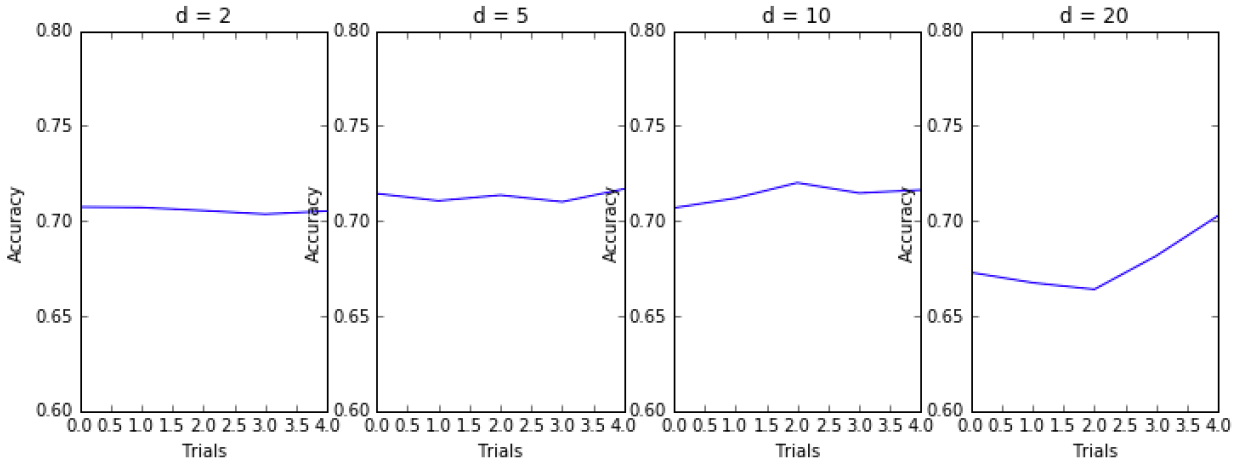
If u_i is fixed, let $\frac{\partial L}{\partial v_j} = 0$, so we can have:

$$v_i = (\sum_{(i,j)\in s} R_{ij} u_i^T)(\sum_{(i,j)\in s} u_i u_i^T + \lambda I)^{-1}$$

So the alternating minimization algorithm is:

1. Random initialize $\{u_i\}$ and $\{v_j\}$.
2. Fix $\{v_j\}$, update $\{u_i\}$ using (1).
3. Fix $\{u_i\}$, update $\{v_j\}$ using (2).
4. If converge, then stop. Otherwise, go to step 2 and repeat.

I use grid search to find the optimal combination of d and λ, the search range for d is [2,5,10,20], and the search range for λ is [0.01,0.1,1,10,100]. The result is as follows:



d=5 and λ = 1 gives the highest validation accuracy.

And the MSE and accuracy with λ = 1 is:

we can see compared with (a), MSE is smaller and accuracy is a little higher.

3.3 Recommending Jokes

I use d=5 and λ = 1 as my final parameters and train on the training data. My code has been attached in the appendix.

My kaggle score is:0.72680