Csci5521 hw3

Kexin Feng 5165462
fengx463@umn.edu

November 9, 2018

Problem 1: I used home made PCA to preprocess the data and do dimension reduction. The home made PCA is implemented with svd. As it turns out, in order to get 90 percentage variance, it's truncated at dimension equal to 73.

Problem 2: For this question, the data set is the handwritten digits. The DoKmeans.m is the core function, whose format is aligned with the requirement.

$$\mathbf{function[assignments, centers, StepCount] = DoKmeans(data, InitialCenters)};$$

InitialCenters was chosen equal to the elements No. 1, 1000, 1001, 2000, 2001, 3000 as required. The number of cluster is set to be k = 6. The termination condition is when the centroids changes are smaller than 1e-6.

Then we can get 6×3 confusion matrix:

$$\begin{pmatrix} 16 & 339 & 11 \\ 41 & 405 & 11 \\ 5 & 16 & 336 \\ 5 & 25 & 429 \\ 406 & 7 & 5 \\ 327 & 8 & 8 \end{pmatrix} \tag{1}$$

If each cluster were assigned a class based on the majority label among members of the cluster, error rate would be

error rate: 0.0658

StepCount = 16

Problem 3 Now we use only 2 principle components and repeat the analysis above.

Confusion matrix:

$$\begin{pmatrix} 24 & 310 & 120 \\ 28 & 214 & 286 \\ 6 & 223 & 370 \\ 160 & 39 & 8 \\ 273 & 0 & 0 \\ 309 & 14 & 16 \end{pmatrix} \tag{2}$$

error rate: 0.2883

StepCount = 29

Problem 4 For this problem, there is no preprocessing. The initial centroids were chosen randomly among all the pixels. For each picture, I tried k = 3, 4, 7, three different cluster numbers. And as k increases, the kmeans picture does have more colors and more detailed structure. The termination condition is when the centroids changes are smaller than 1e-6.

The pictures are shown at the end.

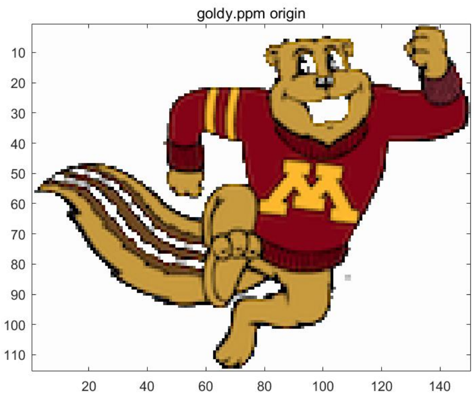
Problem 5: the extra This part was done collaborating with

Name: Miao Yang ID: yang4379

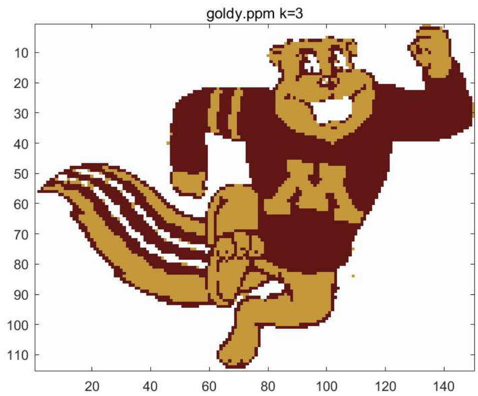
The draft of derivation of theoretical equations are shown at the end of the document.

We implement the EM algorithm on Gaussian dist.. The initial mu, sigma, prior prob. are taken from the K-means process in the previous question. And as required, we used k=4 clusters.

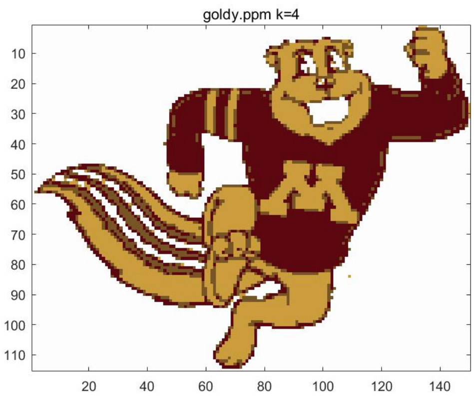
Applying it on the 'Digits089.csv' data, we abtain the following result:



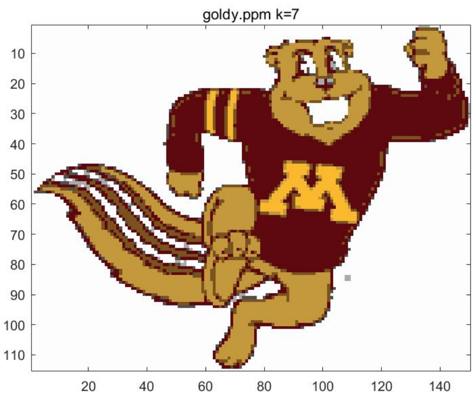
(a) pic1.



(b) pic2.

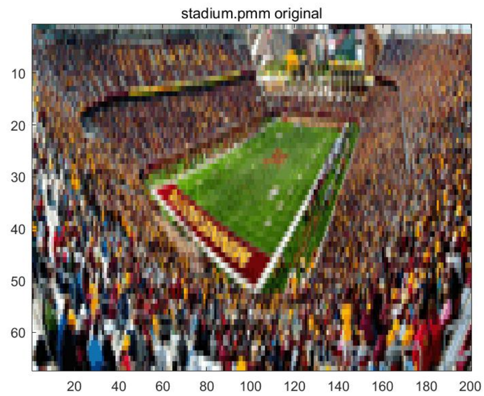


(c) pic3.

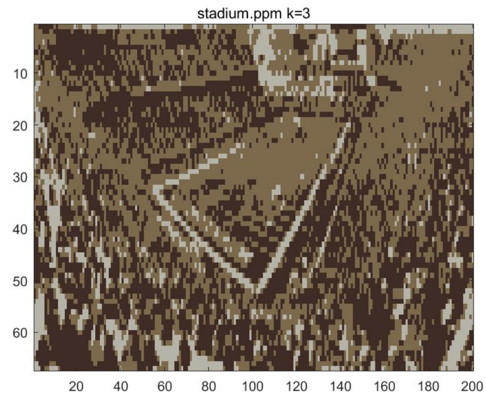


(d) pic4.

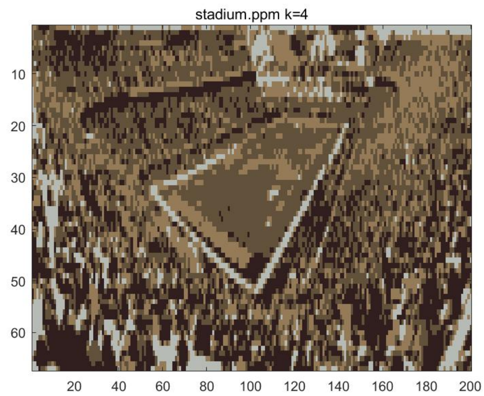
Figure 1: pics



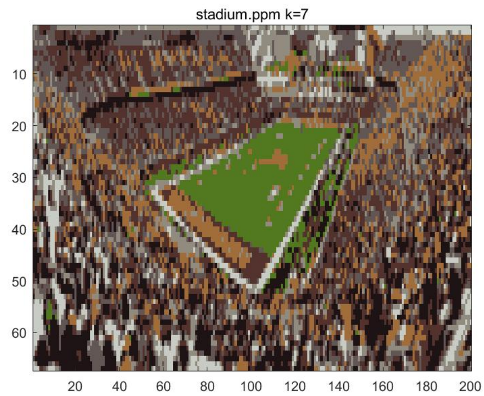
(a) pic1.



(b) pic2.



(c) pic3.



(d) pic4.

Figure 2: pics

Confusion matrix:

$$\begin{pmatrix} 39 & 698 & 31 \\ 350 & 10 & 15 \\ 11 & 85 & 751 \\ 400 & 7 & 3 \end{pmatrix} \tag{3}$$

error rate: 0.0837
And the first several rows of JointProbs and Posteriors look like the following.
JointProbs:

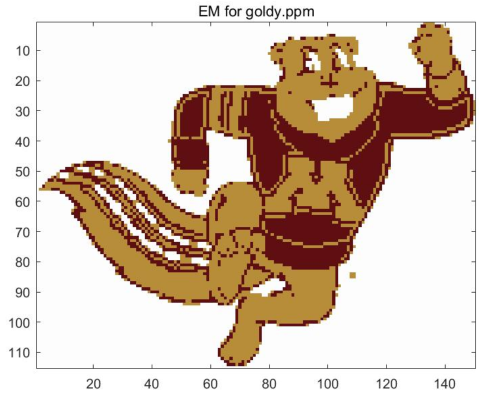
$$unit: \ 1e30 \tag{4}$$

$$\begin{pmatrix}
0 & 0.0056 & 0 & 0 \\
0 & 4.1244 & 0 & 0 \\
0 & 0.8753 & 0 & 0 \\
0 & 9.0033 & 0 & 0 \\
0 & 0.2500 & 0 & 0 \\
0 & 0.2884 & 0 & 0 \\
0 & 0.0101 & 0 & 0 \\
0 & 422.6213 & 0.0012 & 0 \\
0 & 6744641.2281 & 0 & 0.0002 \\
0 & 0.0031 & 0 & 0 \\
0 & 1.1847 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 24.5043 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0.0001 & 0 & 0 \\
0 & 0.1650 & 0 & 0 \\
0 & 20420.3347 & 0.0029 & 23.9168 \\
0 & 0.8461 & 0 & 0 \\
0 & 0.0009 & 0 & 0 \\
0 & 91.2625 & 0 & 0 \\
0.0001 & 4293193250496.6982 & 44384.6517 & 97288.4963 \\
0 & 0.1254 & 0 & 0 \\
0 & 0.0001 & 0 & 0 \\
0 & 0.2375 & 0 & 0 \\
0 & 304.6625 & 0 & 0.0001 \\
0 & 0.0020 & 0 & 0 \\
0 & 0.0020 & 0 & 0 \\
0 & 0.2234 & 0 & 0 \\
0 & 51.7895 & 0 & 0 \\
12589.0655 & 14509453679763.0605 & 94.1731 & 27824176454936.4492 \\
\vdots & & &
\end{pmatrix} \tag{5}$$

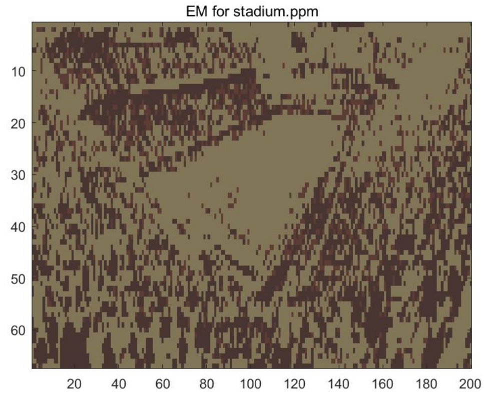
Posteriors:

$$
\begin{pmatrix}
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0.0001 & 0 & 0.9999 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0.9795 & 0 & 0.0005 & 0.0200 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0.9999 & 0 & 0.0001 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0.0070 & 0 & 0.9930 & 0 \\
\vdots & & &
\end{pmatrix} \tag{6}
$$

Applying it to pictures, we got the resulting picutres below.



(a) pic1.



(b) pic2.

Figure 3: pics

READ ME The required functions DoKmeans.m and GetConfusionMatrix.m are used in 'Digits problem' file.

'Picture problem' has DoKmeans.m, too.

'EM problem' has DoEM.m and other necessary files.

Also in each file, it includes main.m which calls the corresponding core function. It serves as a demo of how those function works.

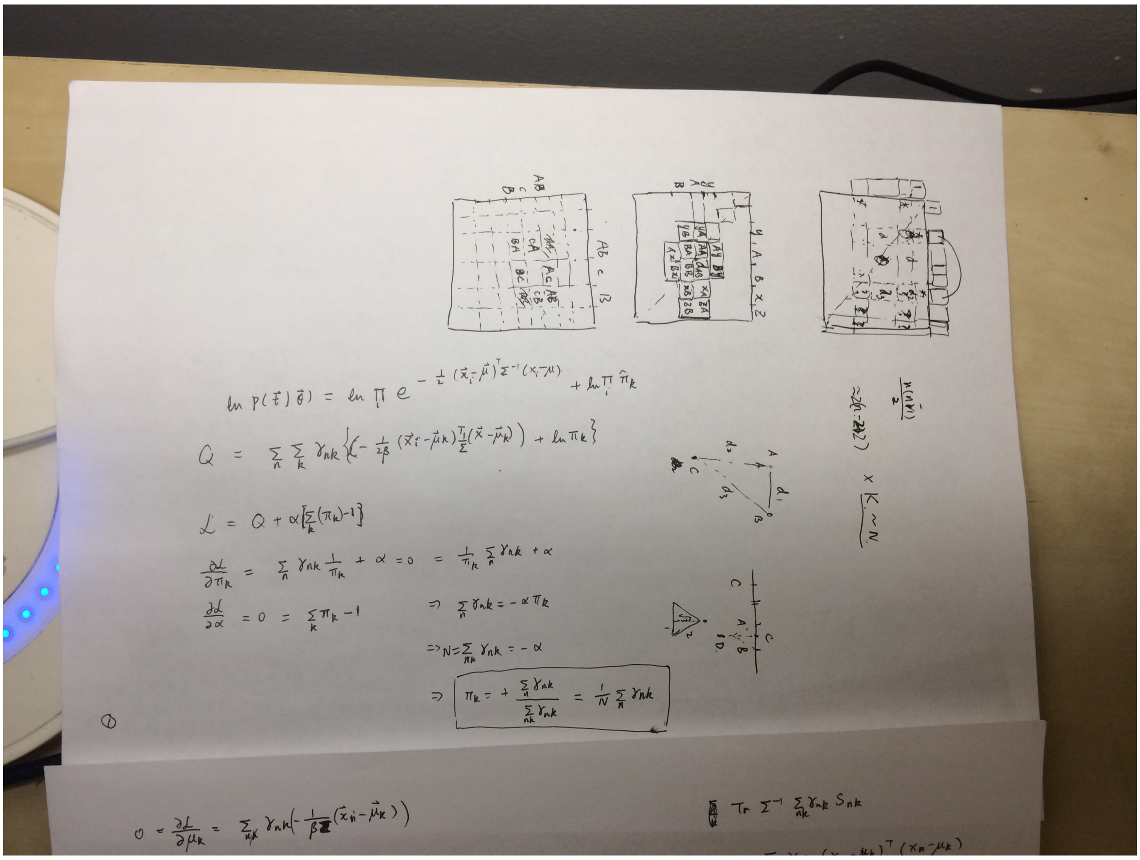


Figure 4: Page 1

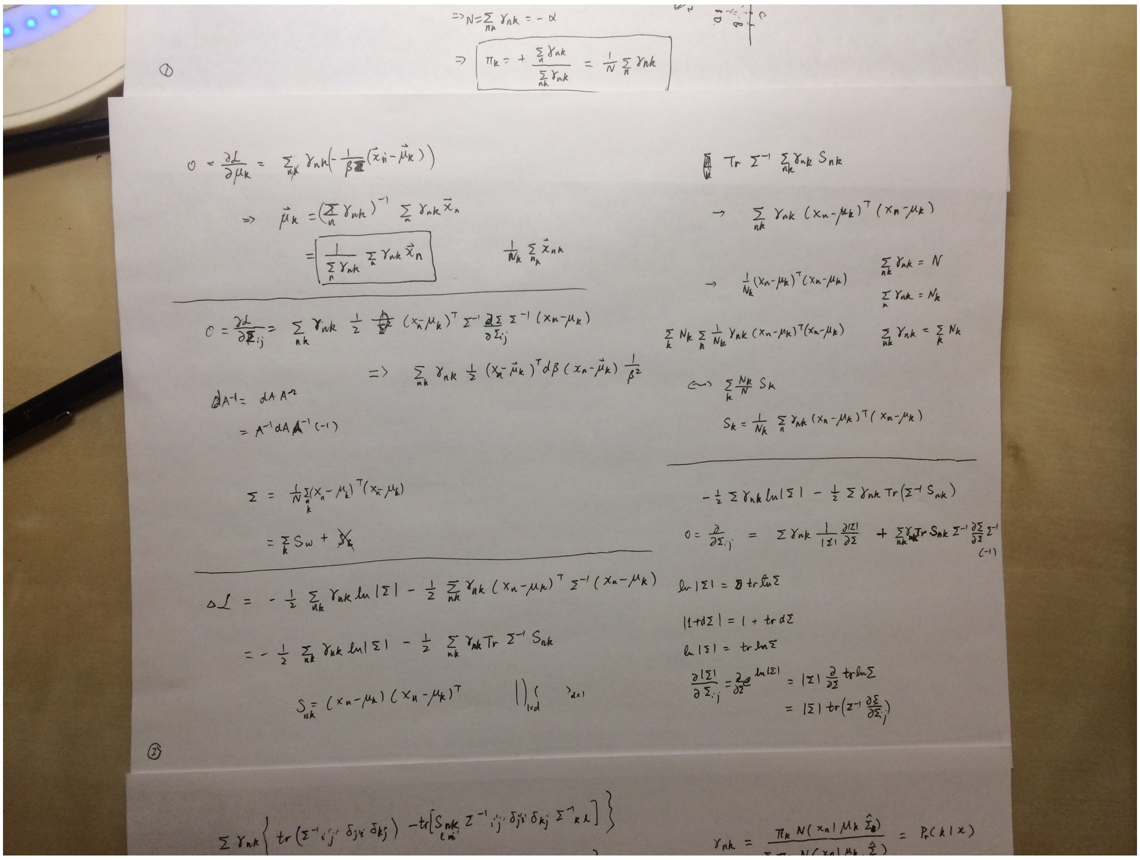


Figure 5: Page 2

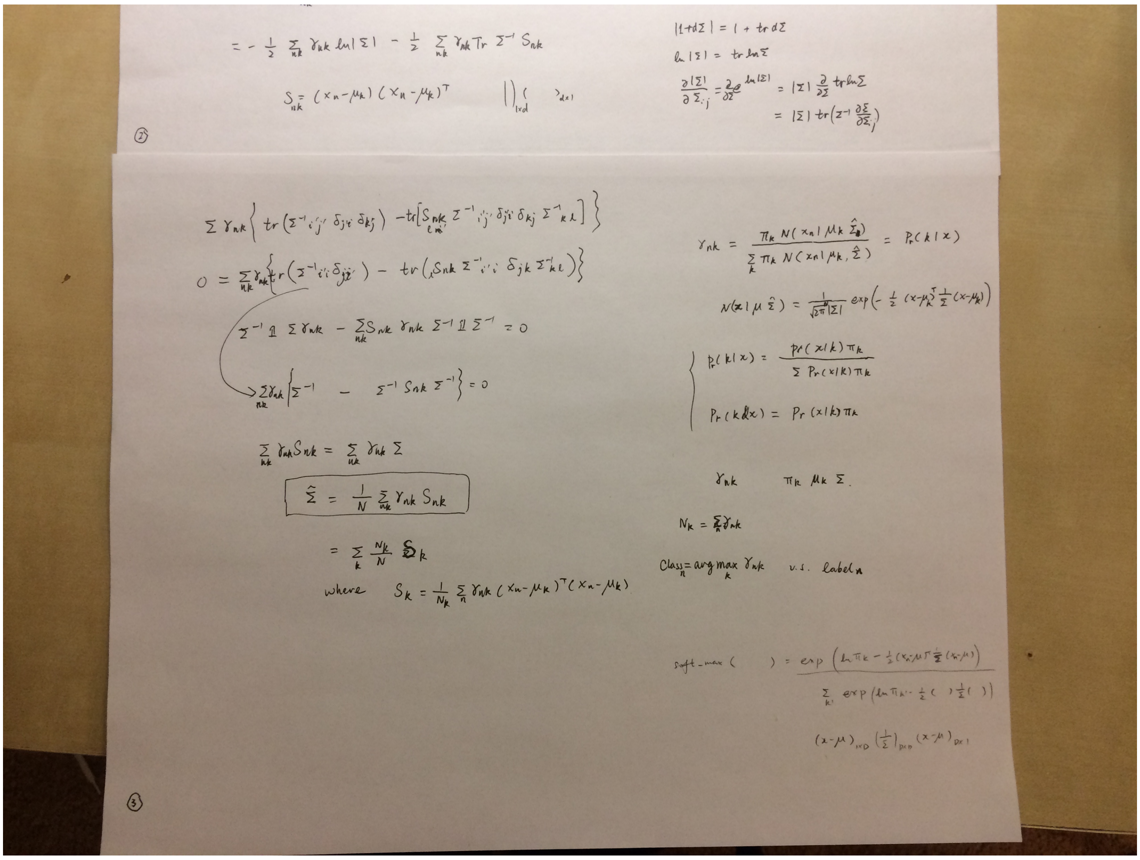


Figure 6: Page 3