Classification for Image

Wenchen Guo

The IMAGE dataset only contains two classes of digits "2" and "6". The matrix images is of size 784-by-1990, i.e., there are totally 1990 images, and each column of the matrix corresponds to one image of size 28-by-28 pixels (the image is vectorized; the original image can be recovered).

We'll use three methods to achieve classfication for this dataset.
1. EM Algorithm.
2. LDA Algorithm.
3. QDA Algorithm.

1. Implementing EM Algorithm

Deriving EM Algorithm for GMM.

Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ be a sample of n independent observations from a mixture of C multivariate normal distributions of dimension p, and let $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ be the cluster variables that determine the component from which the observation clusters. Thus, we can determined $p_{i,c}$ by Bayes theorem:

$$p_{i,c} = P(z_i = c|x_i, \theta) \propto p(z_i = c)p(x_i|z_i = c) = \pi_c \phi(x_i|\mu_c, \Sigma_c)$$

$$\begin{aligned} Q(\theta|\theta^{(k)}) &= \sum_{i=1}^{n}\sum_{c=1}^{C} [p_{i,c}\log\pi_c + p_{i,c}\log\phi(x_i|\mu_c,\Sigma_c)] \\ &= \sum_{i=1}^{n}\sum_{c=1}^{C} p_{i,c}\left[\log\pi_c - \frac{1}{2}\log\Sigma_c - \frac{1}{2}(x_i-\mu_c)^\top\Sigma_c^{-1}(x_i-\mu_c) - \frac{p}{2}\log(2\pi)\right] \end{aligned}$$

The aim is to find the parameters $\theta = \left(\boldsymbol{\pi_c}, \boldsymbol{\mu}_c, \Sigma_c\right)$ that maximize $Q(\theta|\theta^{(k)})$ subject to the constraint $\sum_{c=1}^{C}\pi_c = 1$. We can rewrite the problem as

$$\begin{aligned} l(\theta,\lambda) &= \sum_{i=1}^{n}\sum_{c=1}^{C} [p_{i,c}\log\pi_c + p_{i,c}\log\phi(x_i|\mu_c,\Sigma_c)] + \lambda(\sum_{c=1}^{C}\pi_c - 1) \\ &= \sum_{i=1}^{n}\sum_{c=1}^{C} p_{i,c}\left[\log\pi_c - \frac{1}{2}\log\Sigma_c - \frac{1}{2}(x_i-\mu_c)^\top\Sigma_c^{-1}(x_i-\mu_c) - \frac{p}{2}\log(2\pi)\right] + \lambda(\sum_{c=1}^{C}\pi_c - 1) \\ &= \sum_{i=1}^{n}\sum_{c=1}^{C} p_{i,c}\log\pi_c - \frac{1}{2}\sum_{i=1}^{n}\sum_{c=1}^{C} p_{i,c}\log\Sigma_c - \frac{1}{2}\sum_{i=1}^{n}\sum_{c=1}^{C} p_{i,c}(x_i-\mu_c)^\top\Sigma_c^{-1}(x_i-\mu_c) \\ &\quad - \frac{p}{2}\sum_{i=1}^{n}\sum_{c=1}^{C} p_{i,c}\log(2\pi) + \lambda(\sum_{c=1}^{C}\pi_c - 1) \end{aligned}$$

Take derivative to θ and set it equals 0:

$$\begin{aligned}
\frac{\partial}{\partial \pi_c} l(\theta, \lambda) &= \sum_{i=1}^{n} \frac{p_{i,c}}{\pi_c} + \lambda = 0 \\
\frac{\partial}{\partial \mu_c} l(\theta, \lambda) &= -\frac{1}{2} \sum_{i=1}^{n} p_{i,c} 2\Sigma_c^{-1}(x_i - \mu_c)(-1) \\
&= \sum_{i=1}^{n} p_{i,c} \Sigma_c^{-1}(x_i - \mu_c) = 0 \\
\frac{\partial}{\partial \Sigma_c} l(\theta, \lambda) &= -\frac{1}{2} \sum_{i=1}^{n} p_{i,c} \Sigma_c^{-1} - \frac{1}{2} \sum_{i=1}^{n} p_{i,c} - \Sigma_c^{-1}(x_i - \mu_c)(x_i - \mu_c)^\top \Sigma_c^{-1} = 0
\end{aligned}$$

Then,

$$\begin{aligned}
&\sum_{c=1}^{C} \pi_c = 1 \\
&\pi_c = -\frac{1}{\lambda} \sum_{i=1}^{n} p_{i,c} \\
&\sum_{i=1}^{n} p_{i,c} \Sigma_c^{-1}(x_i - \mu_c) = 0 \\
&\sum_{i=1}^{n} p_{i,c} \Sigma_c^{-1} = \sum_{i=1}^{n} p_{i,c} \Sigma_c^{-1}(x_i - \mu_c)(x_i - \mu_c)^\top \Sigma_c^{-1}
\end{aligned}$$

For π_c:
Since $\sum_{c=1}^{C} p_{i,c} = 1$, then

$$\sum_{c=1}^{C} \pi_c = -\frac{1}{\lambda} \sum_{c=1}^{C} \sum_{i=1}^{n} p_{i,c} = -\frac{n}{\lambda} = 1$$

Thus, $\lambda = -n$ and $\pi_c = \frac{1}{n} \sum_{i=1}^{n} p_{i,c}$.

For μ_c:

$$\begin{aligned}
&\sum_{i=1}^{n} p_{i,c} \Sigma_c^{-1}(x_i - \mu_c) = 0 \\
&\sum_{i=1}^{n} p_{i,c} \Sigma_c^{-1} \mu_c = \sum_{i=1}^{n} p_{i,c} \Sigma_c^{-1} x_i \\
&\mu_c = \frac{\sum_{i=1}^{n} p_{i,c} x_i}{\sum_{i=1}^{n} p_{i,c}}
\end{aligned}$$

For Σ_c:

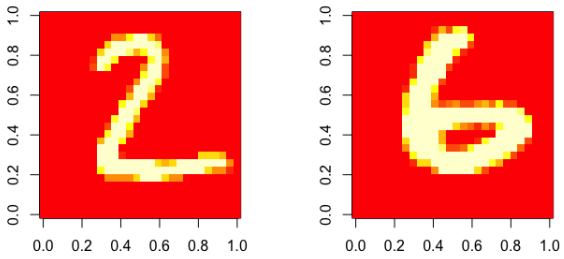
$$\begin{aligned}
&\sum_{i=1}^{n} p_{i,c} \Sigma_c^{-1} = \sum_{i=1}^{n} p_{i,c} \Sigma_c^{-1}(x_i - \mu_c)(x_i - \mu_c)^\top \Sigma_c^{-1} \\
&\sum_{i=1}^{n} p_{i,c} \Sigma_c = \sum_{i=1}^{n} p_{i,c} (x_i - \mu_c)(x_i - \mu_c)^\top \\
&\Sigma_c = \frac{\sum_{i=1}^{n} p_{i,c} (x_i - \mu_c)(x_i - \mu_c)^\top}{\sum_{i=1}^{n} p_{i,c}}
\end{aligned}$$

Therefore:

$$\pi_c^{(k+1)} = \frac{1}{n}\sum_{i=1}^{n} p_{i,c}$$
$$\mu_c^{(k+1)} = \frac{\sum_{i=1}^{n} p_{i,c} x_i}{\sum_{i=1}^{n} p_{i,c}}$$
$$\Sigma_c^{(k+1)} = \frac{\sum_{i=1}^{n} p_{i,c}(x_i - \mu_c^{(k+1)})(x_i - \mu_c^{(k+1)})^\top}{\sum_{i=1}^{n} p_{i,c}}$$

Visualizing Image.

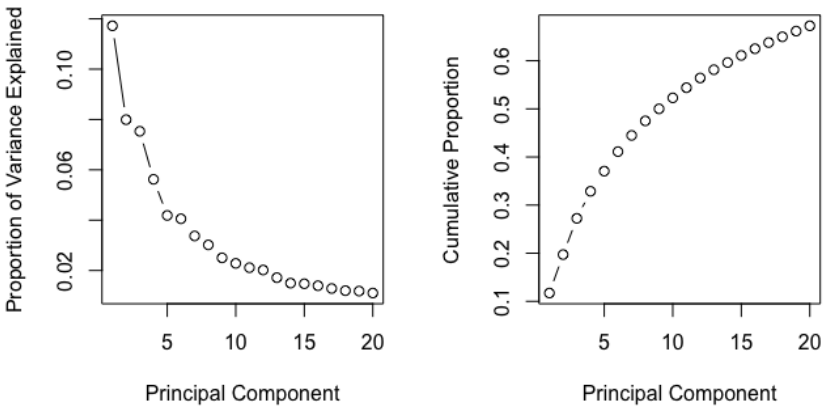
We choose the 1st image of digit "2" and the 1900th image of digit "6" in this dataset.
visualize them, respectively:



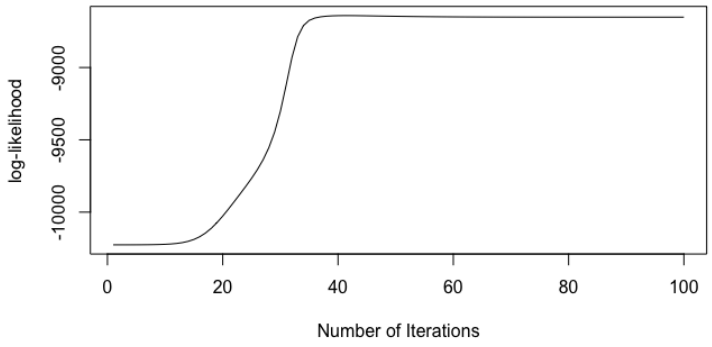
Implementing EM Algorithm.

The dataset of images is of size 784-by-1990, where p=784 and n=1990.

First, we can use PCA to deduce the dimensions of p:
Here are the plots of "Proportion of Variance Explained" and "Cumulative Proportion of Variance Explained" of the first 20th PCs:



We only use the first two Principal Components and reduce our data set to size 2-by-1990.
Then, we can use EM algorithm: Run the E step and M step 100 times. We can plot the log-likelihood function versus the number of iterations. It is obvious that the algorithm is converging.



Results.

The weights for components "2" and "6" are 0.5529 and 0.4471 respectively.

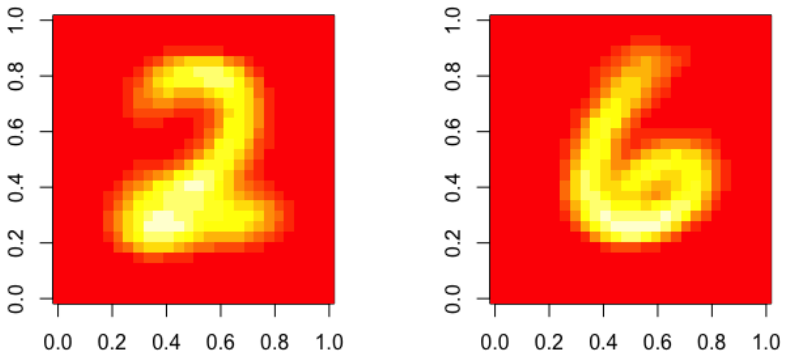
The mean vector $\hat{\mu}$ is

$$\hat{\mu} = [\hat{\mu}_1 \ \hat{\mu}_2] = \begin{bmatrix} -1.827 & 2.259 \\ -0.511 & 0.632 \end{bmatrix}$$

with $\hat{\mu}_1$ for components "2" and $\hat{\mu}_2$ for components "6".

We can use the PCA decomposition matrix and inversely transform the mean vectors into original vcetor space p=784.

Then we can reformat the vectors into 28-by-28 images and show these images:



Accuracy.

Use the $p_{i,c}$ to infer the labels of the images, and compare with the true labels:
For the 1990 image, there are 1032 labeled as component 1 and 958 labeled as component 2.
Our estimates of the labels comtains 1084 of component 1 and 906 of component 2.

By comparing the two sets of labels, we miss classfied 140 out of 1990 images. The miss classification rate is 7.0352%, and the correct classification rate is 92.9648%.

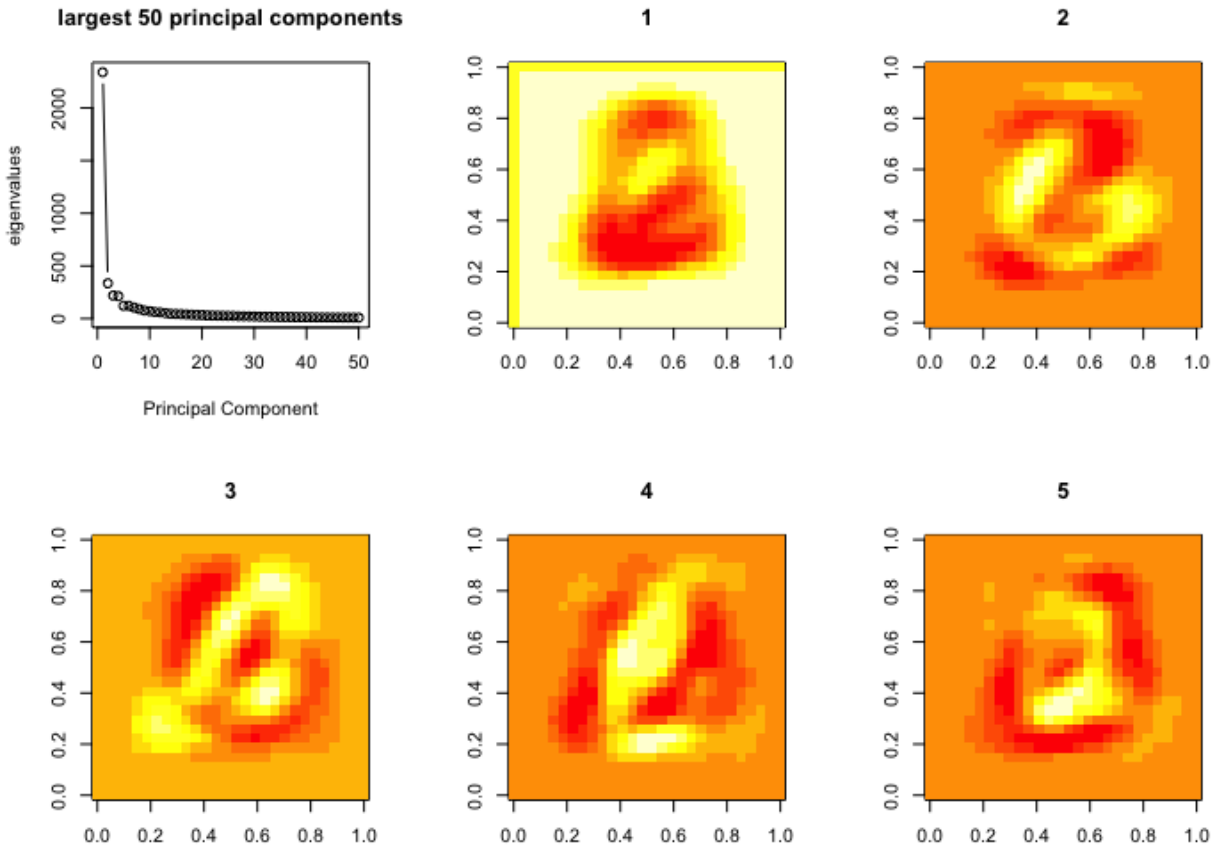
FALSE	140
TRUE	1850

2. Implementing LDA and QDA

This time we will use the label information. We will use 80% of digits "2" and 80% of "6" for training, and 20% of digits "2" and 20% of "6" for testing.

PCA.

As a first step, we will perform dimensionality reduction by PCA. Form the covariance using training data for digits "2" and "6" together. Find the largest eigenvectors associated with the largest 50 principal components. The first plot is the eigenvalues of the largest 50 principal components.
The remaining 5 plots are the first 5 principal components visualization, by reshape them into 28 by 28 images.



Now project the data using the largest 50 principle components, i.e., we will now work with data with 50 dimensions only.

Implement LDA.

Implement LDA using the projected data. Find the means and covariance matrix using training data. Then perform classiffication on test data. The accuracy is 97.49%. The results of the classes are:

		True label	
		2	6
Estimated	2	201	5
label	6	5	187

The miss classiffication rate for class 2 is $\frac{5}{206} = 2.43\%$, and the miss classiffication rate for class 6 is $\frac{5}{192} = 2.43\%$

Implement QDA.

Implement QDA using the projected data. Find the means and covariance matrix using training data. Then perform classiffication on test data. The accuracy is 99.75%. The results of the classes are:

		True label	
		2	6
Estimated	2	206	1
label	6	0	191

The miss classiffication rate for class 2 is 0, and the miss classiffication rate for class 6 is $\frac{1}{192} = 0.52\%$

By comparison of the accuracy, QDA is better.
LDA assumes that the different classes has the same variance or covariance matrix, where QDA does not assume the equality of group covariance matrices.
The covariance matrix of class 2 might not the same as the covariance matrix of class 6, thus QDA tends to be better than LDA.