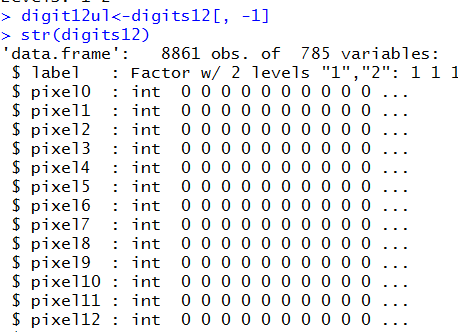
Assignment-2 by Gulsum Oz

1-I get data from MNIST web site than I decided to choose digit 1 and 2 for my homework.

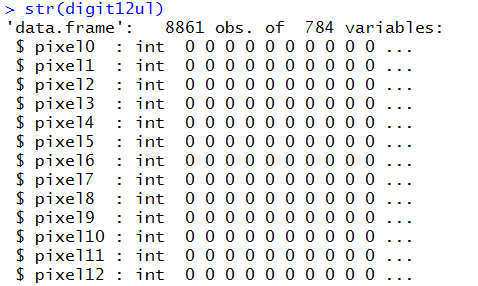
I constructed a data frame from digit 1 and 2

Converted labels of digits to factor for my classification algorithm

Data dimension was 8861 rows and 785 columns, this include 4684 of digit 1’s and 4177 of digit 2’s



2- Next step I took out the labels column named my rest of data as digit12ul. My data frame now has 8861 rows and 784 columns.



3- Next I converted my data frame to matrix named it as mymatrix. Dimension of mymatrix is

dim(mymatrix)

[1] 8861 784

Checking the min and max of the matrix

min(mymatrix)

[1] 0

> max(mymatrix)

[1] 255

4- I used R to displayed some images from the my matrix of combination of digit 1 and 2

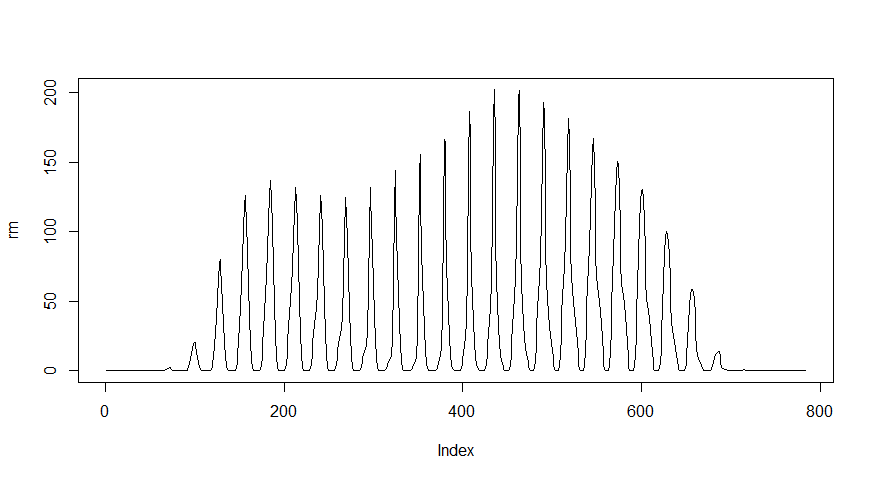
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Row=1111 | Row=2546 | Row=3287 | Row=3345 |
|  |  |  |  |
| Row=5343 | Row=6785 | Row=7865 | Row=8679 |

5-Now, finding and plotting the mean vector 𝝁, then checking the dimension and plotting it.

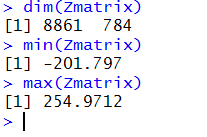
I calculated mean of each row then formed mean vector

Mean vector dimension is [1:784] vector as expected

Mean vector plot looked like this.



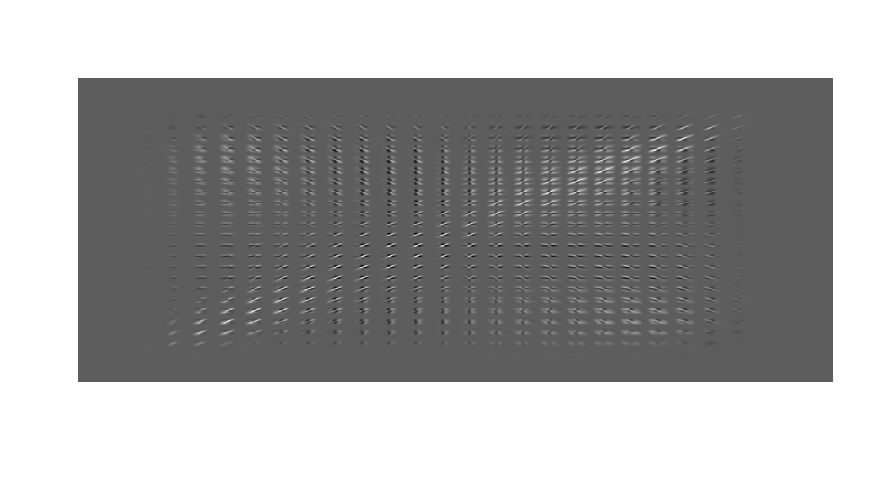
6-Constructing the Z matrix I constructed Z matrix by subtracting mean vector from each corresponding column.



7- Now finding C Covariance matrix from Zmatrix. I used building function of R. cov function does this job.

Cmatrix<-cov(Zmatrix)

Then I displayed the image of Cmatrix below.



8- Finding the Vmatrix

I used R building function to find the eigenvectors and eigenvalues. (Eigen) function does this job in R. It gives two components; one is Eigenvectors the other is eigenvalues.

Eigenvalue<-eigen(Cmatrix, symmetric=TRUE, only.values = FALSE)

Eigenvectors:

Vmatrix<-Eigenvalue$vectors

Eigenvalues :

Values<-Eigenvalue$values

In order to check the orthogonality of Vmatrix, I did multiplication of few pairs of eigenvector matrix.

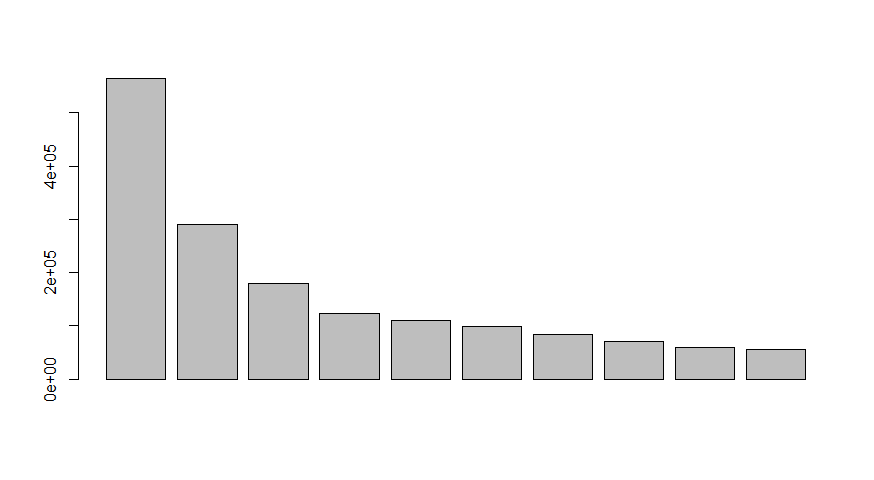
Vmatrix[4:6,1] %\*% Vmatrix[4:6,2]

Result is [,1]

[1,] 0

It is proven that it is orthogonal.

I also can be able to visualize the variance of dimensions, draw the bar plot of the eigenvalues also called screeplot.



9-Finding the Pmatrix with eigenvectors matrix.

First I tried the given equation P=Z\*t(V) but I realized that it doesn’t give correct PC’s

Than after little search I found in R we can get this step by P=Z\*V

Eigen function already gives the transpose of V.

I used R calculation like this Pmatrix<-Zmatrix%\*%Vmatrix

To check mean of the P matrix

mean(Pmatrix)

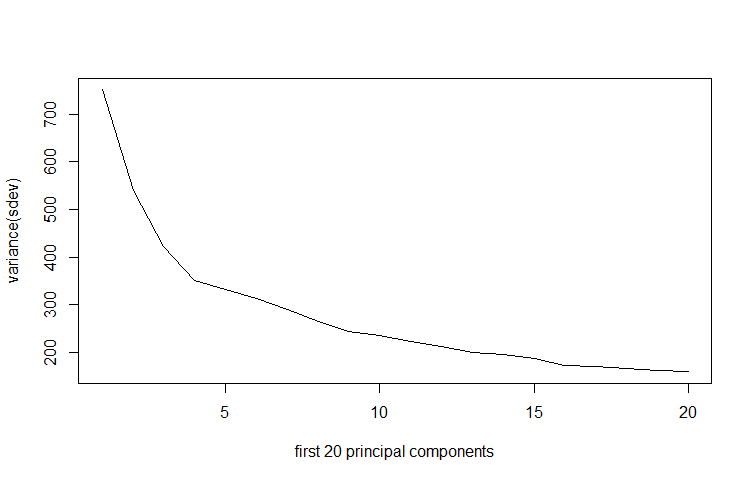
[1] 8.522067e-16

round(mean(Pmatrix))

[1] 0

|  |
| --- |
|  |
|  |
| |  | | --- | |  | |

First 20 principal componentsand its variances is shown in the graph below



10- Reconstructing the image with first two Principle Components. I used the equation

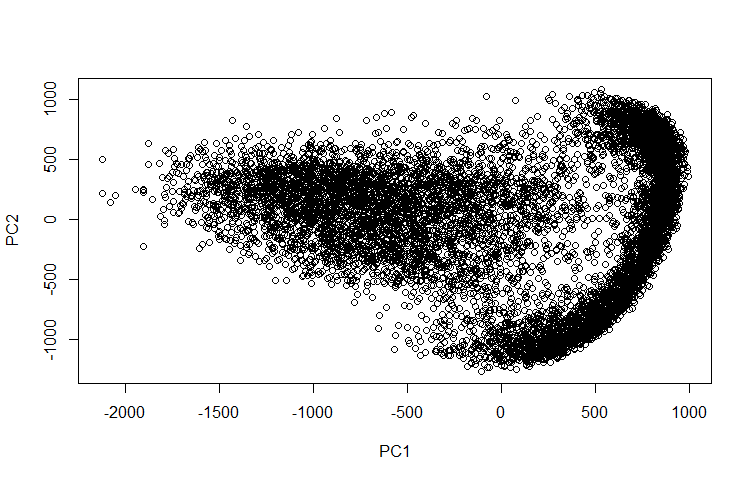
R= PV and for Pmatrix I only used first two PC. Then I tried to display one of the digit by using R.

Here is one of my reconstructed image and its original version.

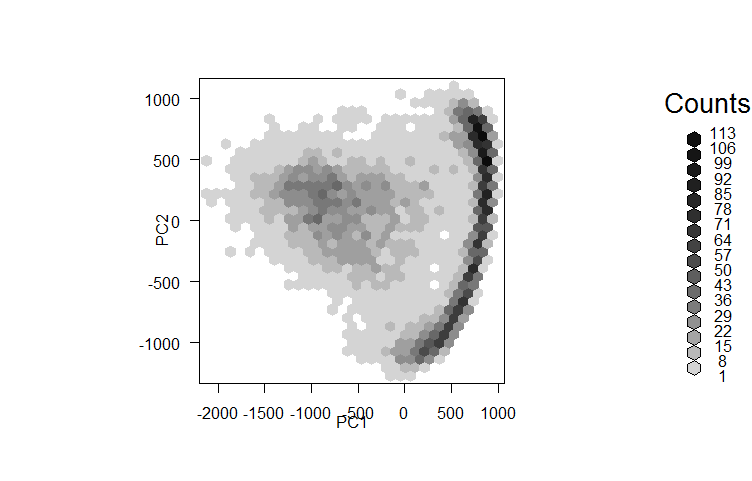
|  |  |
| --- | --- |
| Reconstructed | Original |
|  |  |

11- I plot two principle components for visualization.

Below is the two Principle components.



12- Again by using R I plot 2D-histogram of the above picture



Separate histogram for digit 1 and 2 by using first two PC’s

|  |  |
| --- | --- |
| Digit2-2D Histogram | Digit1-2D histogram |
|  |  |

13-I used Naïve Bayes built-in function in R and made a model to classify digit by using only two principle component that I found above.

Below is my model prediction outcomes confusion table for test data.

Prediction 1 2

1 1383 84

2 22 1169

Classification performance:

Accuracy= 1383+1169/ 1383+1169+22+84 = 2552/2658= 0.9601204

Accuracy= 96%

Sensitivity= 1383/1383+84 =0.9427403

Sensitivity= 94.2%

Specificity= 1169/22+1169= 0.9815281

Specificity= 98.1%

Positive predictive value= 1383/22+1383 =0.9843416

Positive predictive value= 98.4%

14- Then I decided to use K-Nearest Neighbor classification to see the difference

Here is my test data classification for KNN by using first two principle components

For k=3

knnmodel 1 2

1 1376 27

2 29 1226

Accuracy increased little more

Accuracy= 1376+1226/1376+1226+27+29 = 0.9789315

Accuracy for KNN = 97.8%

Then I played around tried different k number I got min misclassification at k=10

Here is the confusion matrix for k=10

knnmodel 1 2

1 1376 17

2 29 1236

Accuracy turned to be (1376+1236)/(1376+1236+17+29)

= 0.9826938

Accuracy = 98.2%