Assignment-2 REPORT

200050040-200050075

QUESTION-4

Contents

Principal Component Analysis (PCA)

1 Computing the mean, covariance, eigen vectors

I used the 60000 examples in the training set of "mnist.mat" for the computation.I first casted them into floating point type before performing any computations.

The process is going to be same irrespective of the 'digit' on which we perform the computations.I just looped through all the 10 digits 0-9 and applied the same process.

So for a particular digit, let it be repeated **N** times out of 60000, which positions can be found by the 'labels_train' given in the mnist dataset.

So first I seggregated all the 'N' data samples of a particular digit in a **28x28xN** 3D matrix. For performing computations we need to 'reshape' it to **784xN** 2D matrix, where each column has the 784 coordinates of a particular 'instance' of that digit.

Now we need to caluculate the mean μ, covariance **C** and the principal eigen vector $\mathbf{v_1}$ and it's eigen value λ_1

We need to caluculate the **ML estimate** of mean from the N data samples $\mathbf{d_i}$. But it was derived in the lecture that the **ML estimate** of mean, Covariance are same as the **Sample Mean** and **Sample Covariance**

Caluculating the Mean μ:

So the (empirical)mean is the sample mean of the N samples. i.e, $\mu = \frac{1}{N}\sum_{i=1}^{N}\mathbf{d_i}$, where $\mathbf{d_i}$ is the i'th column in the data.So it is just the average of all the columns and it is straight forward to find mean.

Caluculating the Covariance C:

Let the data matrix **D** be the **784xN** matrix. The (empirical)Covariance is the sample covariance of the N samples.So the(i,j) th element in the Covariance matrix is the covariance of the (i,j) th coordinates in each of the N data samples i.e,

$$\mathbf{C_{ij}} = Cov(x_i, x_j) = \frac{1}{N}\sum_{i=1}^{N}(x_i - mean(x_i))(x_j - mean(x_j))$$

So we can see that if **S = D - μ**, then **C = S*S^T/N**. i,e S is the matrix after subtracting mean from the data. So I used this to caluculate the 784x784 Covariance matrix.

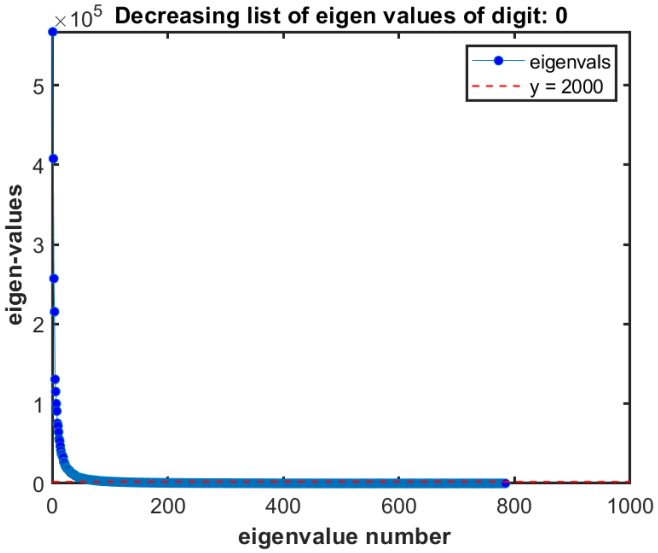
Principal Eigen value and vector: After knowing the covariance matrix **C**, I caluculated the eigen vectors V and eigen values D using [V,D] = eig(C).

Then Sorted the Eigen values using sort(), and picked the largest eigen value λ_1 and it's eigen vector $\mathbf{v}_1$.

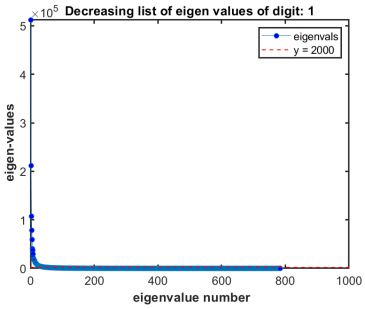
All these values cannot be displayed , so I stored all of those 4 for all the digits as means.mat ,Covariances.mat, "principle_eigvals.mat" "principle_eigenvecs.mat" names are self-explanatory, and each file has data of all the 10 digits.They are stored in the "results" folder .They also get generated if you run the code.**NOTE the covariance data it too large to submit, so I did not submit that data.But It gets saved when Running the program**

2 Plotting eigen values

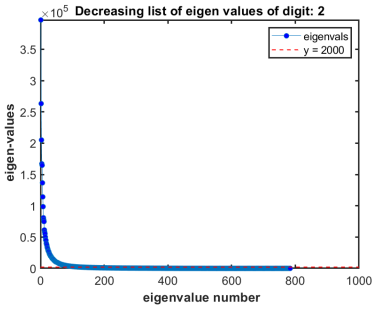
For each digit, we found all the eigen values in a matrix D using [V,D] = eig(C) earlier. Now I sorted those eigen values in descending order using sort() function and plotted those 784 eigen values. The obtained plots for all the 10 digits almost have the same behaviour, so I plotted the eig values of all digits 1-9 in smaller figures and 0 in a bigger figure. But they all can be found in the results folder.



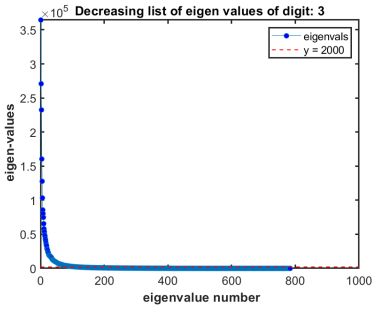
Digit: 0



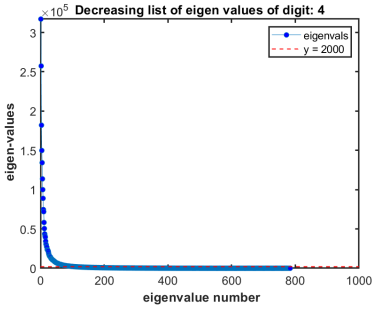
Digit: 1



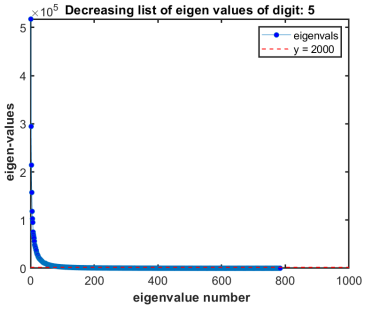
Digit: 2



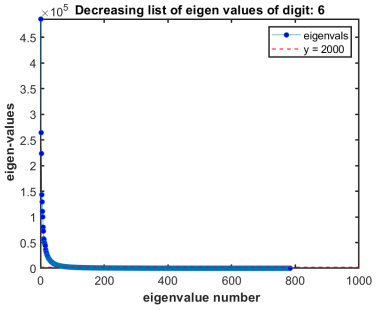
Digit: 3



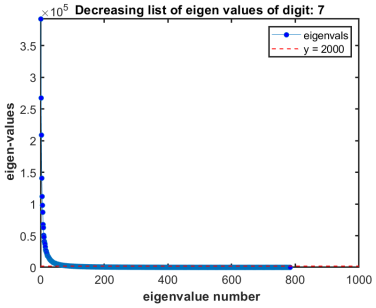
Digit: 4



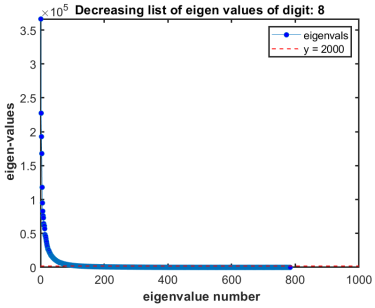
Digit: 5



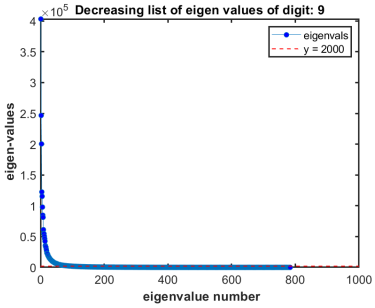
Digit: 6



Digit: 7



Digit: 8



Digit: 9

2.1 Observations

From all the above figures, we can observe that the eigen values fall of very rapidly at start and hence only the first few eig-values are large or significant, all others are very small.i.e, the variation is high in only a few dimensions, and in the variation is significantly less in many other dimensions.I also printed on the console, the number of eigen values more than 1% of the maximum eig-value, It shows that on an avg, there are around **80** eigen values more than the threshold 1% out of 784. and if we increase the threshold to even more like 5%, then it would be even less.

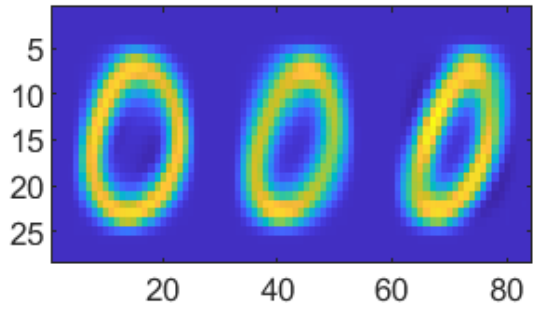
Hence we only have around 80 **significant** modes of variation if we set the threshold to 1%.Which is far less than 784.It is because we can see there is a lot of black around the digits, which does not vary at all and so There is no **variation** in that direction and hence eigen value along that direction is also very less.

3 Plotting Modes of variation

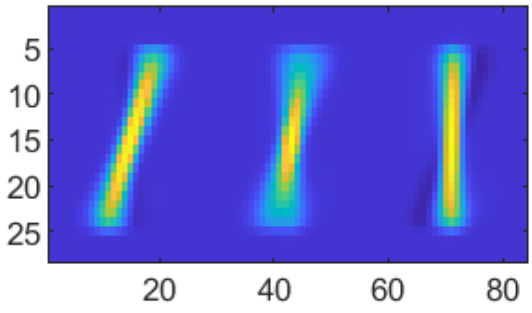
We already found μ, v_1, λ_1 for all digits 0-9

For each digit, I reshaped the 3 column vectors $\mu - \sqrt{\lambda_1}\mathbf{v_1}$, μ and $\mu + \sqrt{\lambda_1}\mathbf{v_1}$ to 28x28 matrix and showed the 3 matrices as images **side-by-side** using the **imagesc()** function.(Using imshow() we are not abl to see any details)

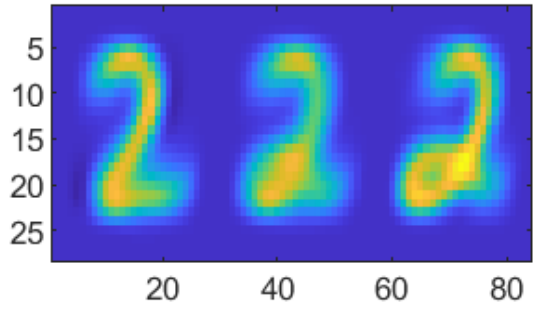
The obtained Results are:($\mu - \sqrt{\lambda_1}\mathbf{v_1}$ left, μ middle and $\mu + \sqrt{\lambda_1}\mathbf{v_1}$ right)



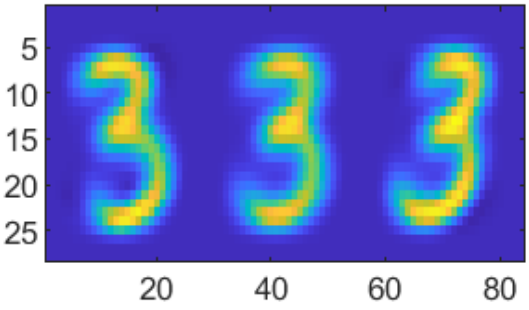
Digit: 0



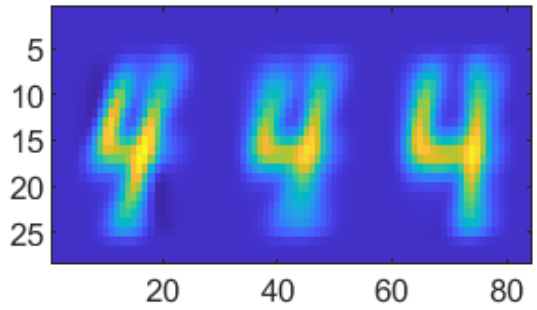
Digit: 1



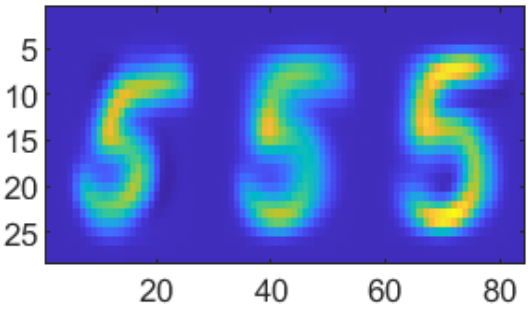
Digit: 2



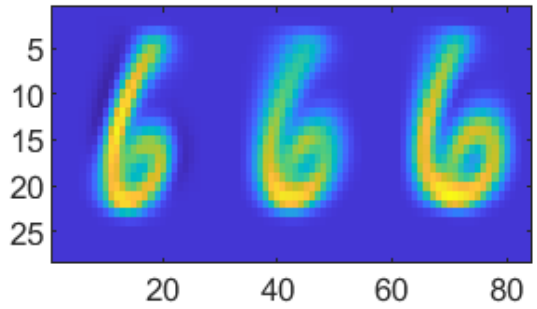
Digit: 3



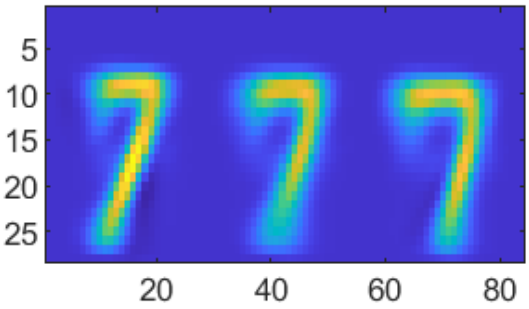
Digit: 4



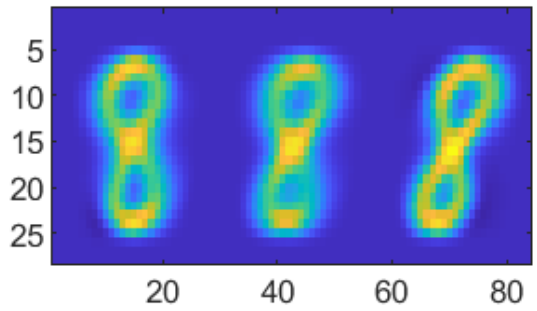
Digit: 5



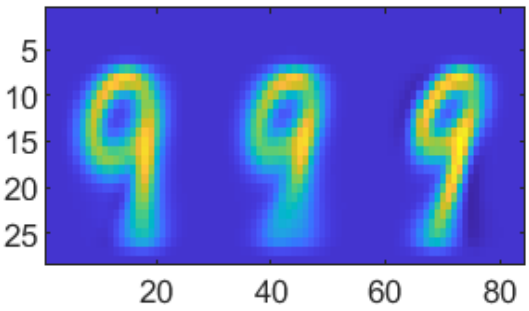
Digit: 6



Digit: 7



Digit: 8



Digit: 9

3.1 Observations

For all the digits we can see the most significant variation in writing the digit by various people,it is because in the next Mode of variations, the **eigen values** drop considerably and hence the **variation** or Variance along those modes would be even lesser.The **Primary mode of variation** captures the most significant differences among people writing the digits.

So let's take the digit-1, then we can see from the plot that most people write it as just a line and the spikes etc they give to the digit are small.Also the primary difference among all the '1's is the orientation of the "line". Also we can see that there are very few people who write '1' with a negative **slope**, i,e, *slanting* back. Most people write it slanting forward.

Say we take digit: **9**. Then the round at the top is almost not varying except a little tilt. And the primary variation is in the tail at the bottom.So people direct the tail in some what wider angle.

Hence using the *Principal modes of variation* we can actually tell how people generally write those digits.

4 Code Running Instructions

`Run the q4.m file in code folder to produce all the 2x10 = 20 result images:` They are also in the 'results' folder.

1. Eigen value plots of each digit, "eigvals**N**.png" for N = 0 -9
2. Modes of variation of each digit, "Images**N**.png" for N = 0 -9

Also I kept the "mnist.mat" data file in the same code folder, Which is read by the program when run.Also the program displays the number of significant eigen values for each digit on console.