Assignment-2 REPORT

200050040–200050075

QUESTION-5

Contents

PCA for Dimensionality Reduction

We are given the image data in the Euclidean Space of $28^2 = 784$ dimensions.Now we want to re-represent the images using only 84 coordinates in a **84-dimensional basis** for some 84-dimensional **hyperplane** within the original Euclidean space, such that the chosen 84-dimensional hyperplane **maximizes the total dispersion of the original data** (for the chosen digit) within the hyperplane.

Lets say for a digit, we have the data of **N instances** of the digit and using that data we caluculate the **ML estimate** or the **empirical mean** μ **and the Covariance matrix C**, using the same process that was used and explained in Question.4 of the assignment. I am briefly explaining the same again in next few lines.

So first I seggregated all the 'N' data samples of a particular digit in a **28x28xN** matrix and 'reshaped' it to **784xN** 2D matrix, where each column has the 784 coordinates of a particular 'instance' of that digit.

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.

Let the data matrix **D** be the **784xN** matrix.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,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.

After finding the covariance matrix, I found the **Eigen vectors,values** by using [V,D] = eig(**C**). **V** is the matrix whose 784 columns are the 784 eigen vectors and **D** is a diagonal matrix of eigen values.

Our data is in a 784 dimensional basis, also let the **unit** eigen vectors be $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \ldots\ldots, \mathbf{v_{784}}$ and their eigen values be $\lambda_1, \lambda_2, \ldots$. Let **x** be an instance of the particular digit which is a 784x1 vector. As the 784 eigen vectors are **orthogonal** (independent too) they represent a **basis** of the 784 Ecuclidean space.So the variation **around mean** μ can be written as a linear combination of the **eigen vectors**

$$x = \mu + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots\cdots + a_{784}\mathbf{v_{784}}$$

The variance along a particular principle mode of variation(i,e, along a eigen vector) is = its eigen value.But in Q4, we saw that only a few eigen values are **significant** so the variation along other directions is **very less** and it could be thought as **perturbation** in the measurement.So we can reduce the data into a lower dimension and still capture most of its variation.

1 Reducing the Dimensions

It was proved in the lecture that the lower **M** dimensional subspace or **hyperplane** that maximises the **dispersion** is represented by the **top M** principal modes of variation or the **top M eigen vectors** as its basis.

First we need to find the **top M eigen vectors** and then **project** our original data onto the hyperplane.

The projection of vector **x** on a M dimensional plane is given be proj($\mathbf{x}$) $= \langle \mathbf{x}, \mathbf{v_1} \rangle \mathrm{v}_1 + \langle \mathbf{x}, \mathbf{v_2} \rangle \mathrm{v}_2 + \cdots + \cdots + \langle \mathbf{x}, \mathbf{v_M} \rangle \mathrm{v}_M$ [assuming mean 0]

Including the mean μ

proj($\mathbf{x}$) $= \mu + \langle \mathbf{x} - \mu, \mathbf{v_1} \rangle \mathrm{v}_1 + \langle \mathbf{x} - \mu, \mathbf{v_2} \rangle \mathrm{v}_2 + \cdots + \cdots + \langle \mathbf{x} - \mu, \mathbf{v_M} \rangle \mathrm{v}_M$ [any general case]

Now to represent a point **on the hyperplane**, we only need the **M** coordinates, which are nothing but the **coefficients** of the **M** eigen vectors, and they are nothing but the dot product of **x** with the **corresponding eigen vector** [as $\langle \mathbf{x}, \mathbf{v} \rangle = \mathbf{x} \cdot \mathbf{v}$ in real space]

So we transform $\mathbf{x}_{784\times1}$ to 84 coordinates by using the above equation.

$\mathbf{x} \rightarrow [\langle \mathbf{x} - \mu, \mathbf{v_1} \rangle, \langle \mathbf{x} - \mu, \mathbf{v_2} \rangle, \ldots \langle \mathbf{x} - \mu, \mathbf{v_{84}} \rangle]^T_{84\times1}$

In the code, I wrote a function reduce(X,mu,E,D,M) which takes 5 inputs (1) the vector **x** , (2) the mean vector μ (3,4) Eigen vectors and values, (5)The dimension to which we want to reduce which in our case is 84.

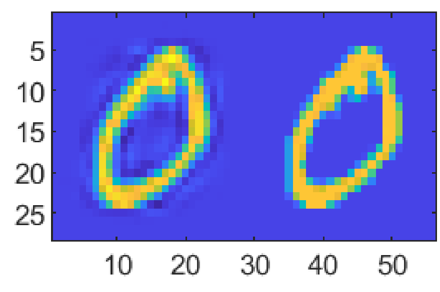
It first finds the **first M eigen vectors** from the given input vectors.Subtracts mean from the given input vectors and computes the 84 coefficients required by dot product $(\mathbf{x} - \mu) \cdot \mathbf{v}$, returns the vector **C** with the 84 coordinates

2 Reconstructing to original dimensions

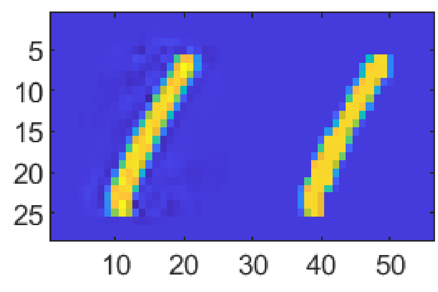
The algorithm to regenerate is same as th eequation wrote above, but we do the reverse. So now as we know the coefiicients of the **top M eigen vectors**, we again multiply the coefficients with the 784 dimension eigen vectors and finally add the mean to regenrate the **projection of x**. We dont actually get **x**, what we get is it's projection on the hyperplane.But it is a good enough approximation of **x** because the variation along the other dimensions is **very less**.

I wrote the function regen(C,mu,E,D,M) which takes 5 inputs (1) the vector **C** which is to be reconstructed , (2) the mean vector μ (3,4) Eigen vectors and values, (5)The dimension to which we want to reduce which in our case is 84.

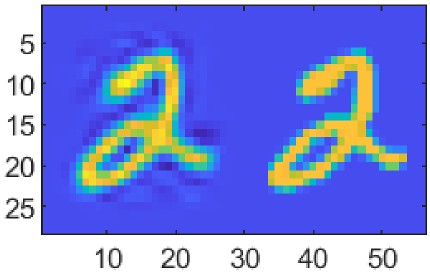
It first finds the **first M eigen vectors** from the given input vectors and multiplies the 84 coefficients with the eigen vectors and finally adds the mean μ, returns the vector **X** with the 784 coordinates
After reconstructing, I visualised the images of the re-constructed and the original side-by side using the imagesc() function and got the following results:I also saved them using the saveas() function.**Reconstructed: LEFT Original: RIGHT**



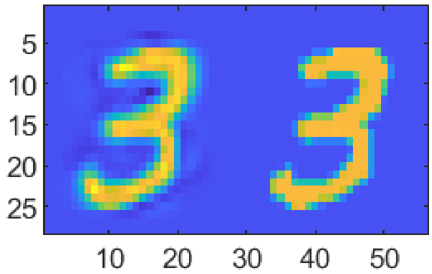
Digit: 0



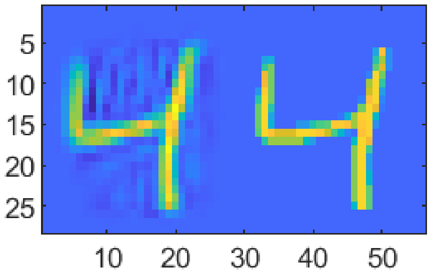
Digit: 1



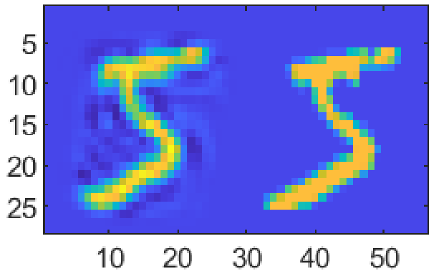
Digit: 2



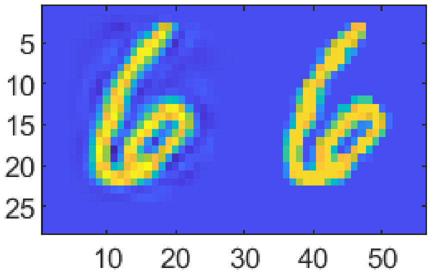
Digit: 3



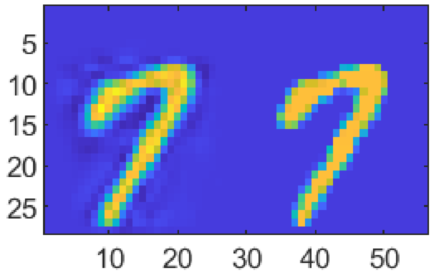
Digit: 4



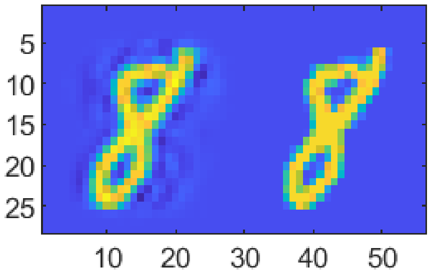
Digit: 5



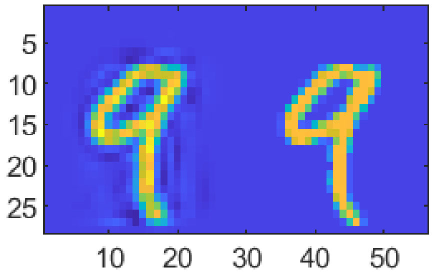
Digit: 6



Digit: 7



Digit: 8



Digit: 9

So we can observe that all the **main data** in the data is NOT LOST. All the primary modes of variationa are captured very well.The difference is just the **perturbation** around the digits.

3 Code Running Instructions

`Run the Q5_dimred.m file in code folder to produce all the 10 result images:` They are also in the 'results' folder.

- Comparison for 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.