Report: LARP Assignment 1

Keval Dodiya (CS19M023) Karan Jivani (CS19M030)

Problem 1

Part 1:

a) Plot the Dataset and Singular Vectors.

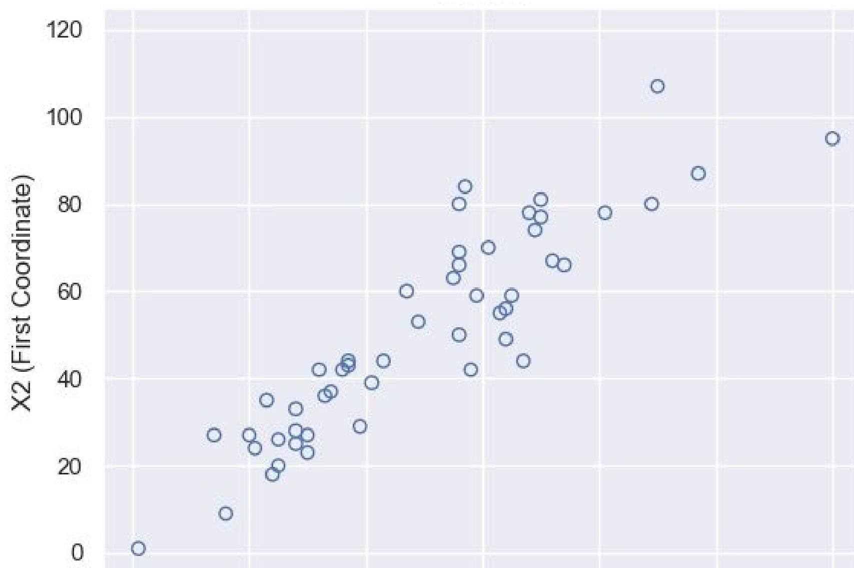


Figure 1.1: Scatter Plot of Dataset1.csv

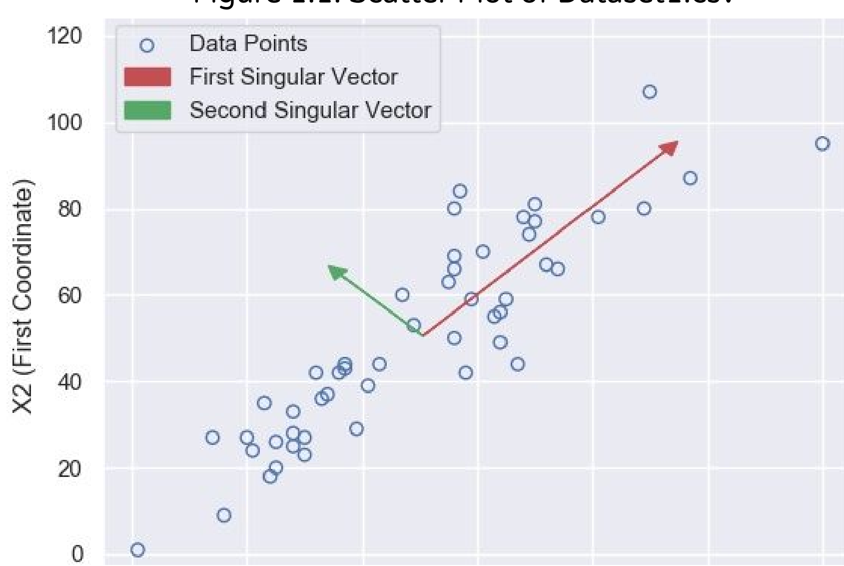


Figure 1.2: Dataset with Singular Vectors

(**Note**: We have used left singular vectors as they have dimension 2, which is plottable)

b) Project the data on top singular vector and plot the projected data along with the original data.

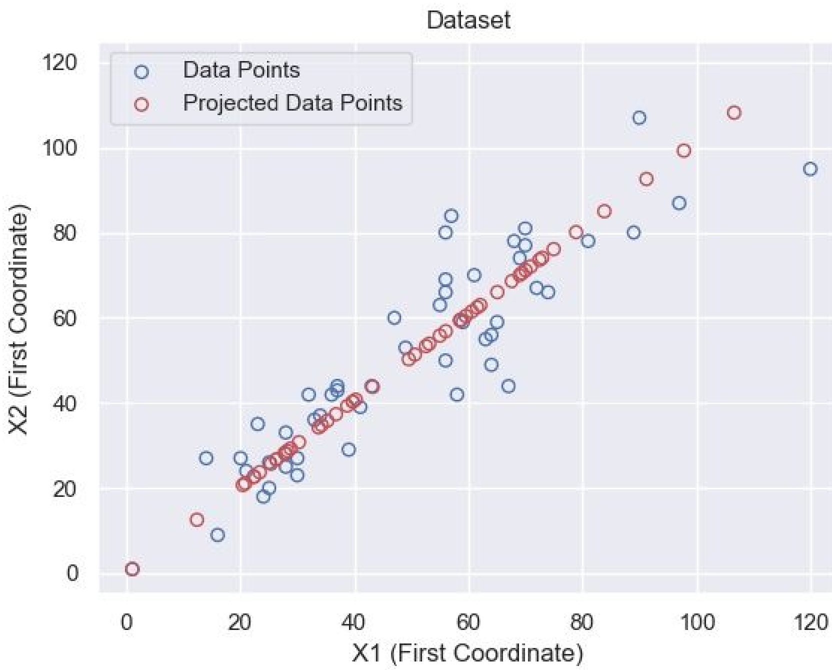


Figure 1.3: Dataset and Projected Data

c) Average error by reducing dimension: 54.862715
Percentage information lost: 8.671393

d) Plot the least squared approximation along with original dataset and top singular vector.

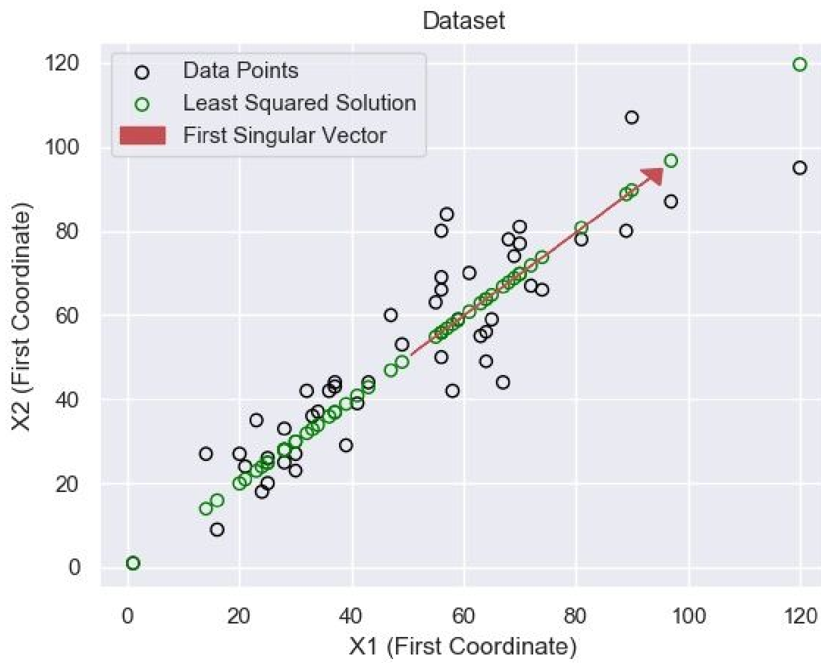


Figure 1.4: Dataset, Least Squares Approximation and Top Singular Vector

- Average Error by Least Squares Approximation: 110.366280
- Top Singular Vector coincides with Least Squares Approximation, but error with least squares approximation is more compared to error with dimension reduction.
- That can be explained as, in Least Squares Approximation, error is due to difference in y-component (since x-component remains the same, green circles, Fig. (1.4)) which is a vertical distance of given point (black circles, Fig. (1.4)) from Approximation Line.
- Whereas in Dimension Reduction, data is projected onto top singular vector, and thus new datapoint (red circles, Fig. (1.3)) which is on singular vector, will be perpendicular from original point(blue circles, Fig.(1.3)), leading to less overall distance.

Part 2:

a) Project the 10 dimensional dataset on top 2 singular vectors and plot the projected data.

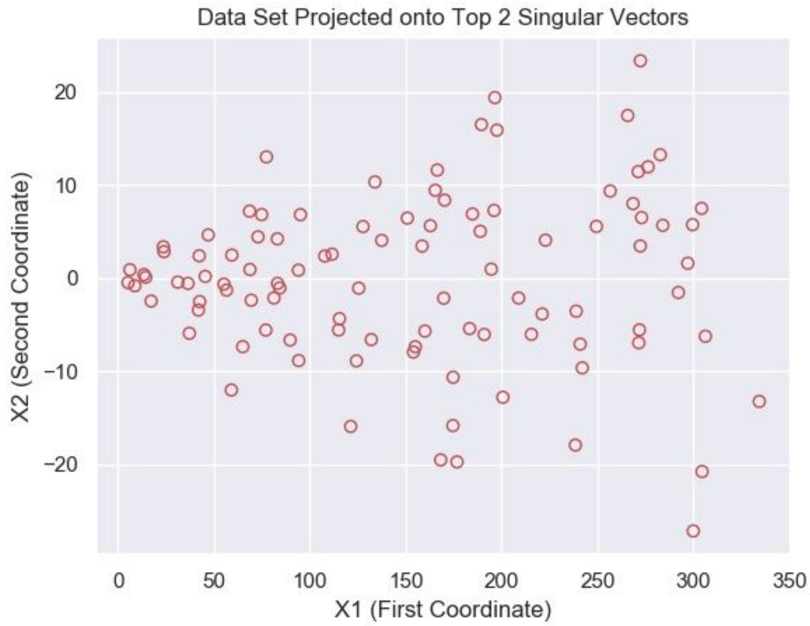


Figure 1.5: 10 dimensional data projected onto top 2 singular vectors

b) Information lost (using only 2-dimension) = 21.1467809

c) Number of singular vectors required to capture at least 90% information = 6

Problem 2

Part 1:

Figure 2.1 Original Images In Grayscale

a) Square images

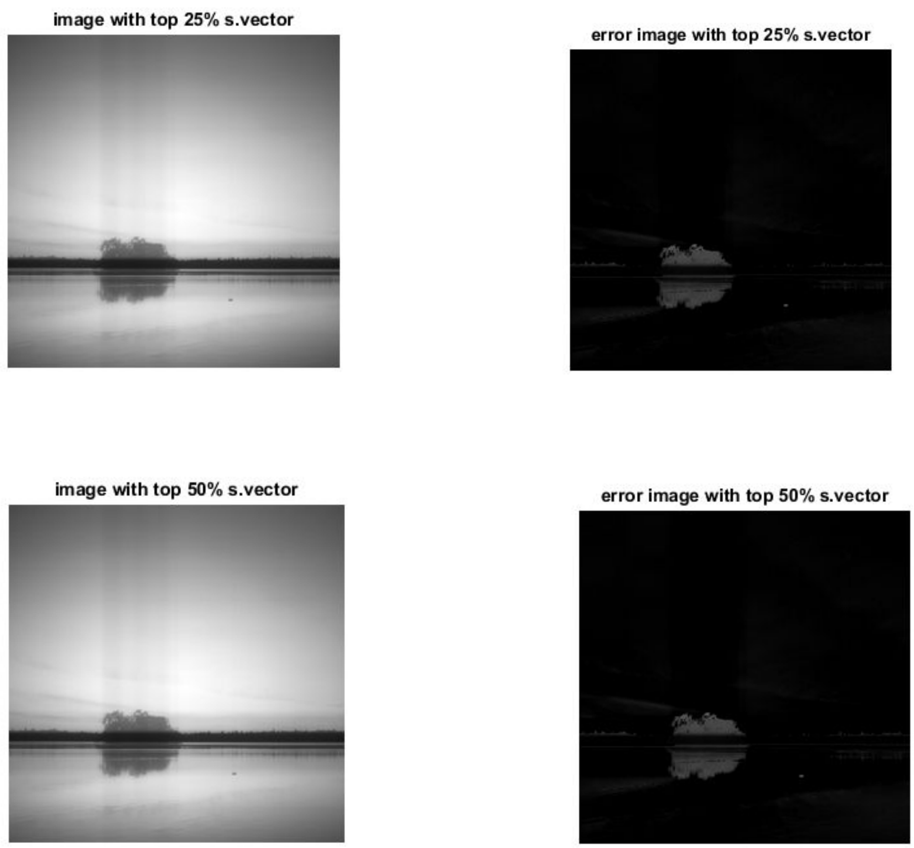


Figure 2.2 Square Images And Corresponding It's Error Image With N = 10%,25%,50% Of Top Singular Vectors.

b) Rectangular Images

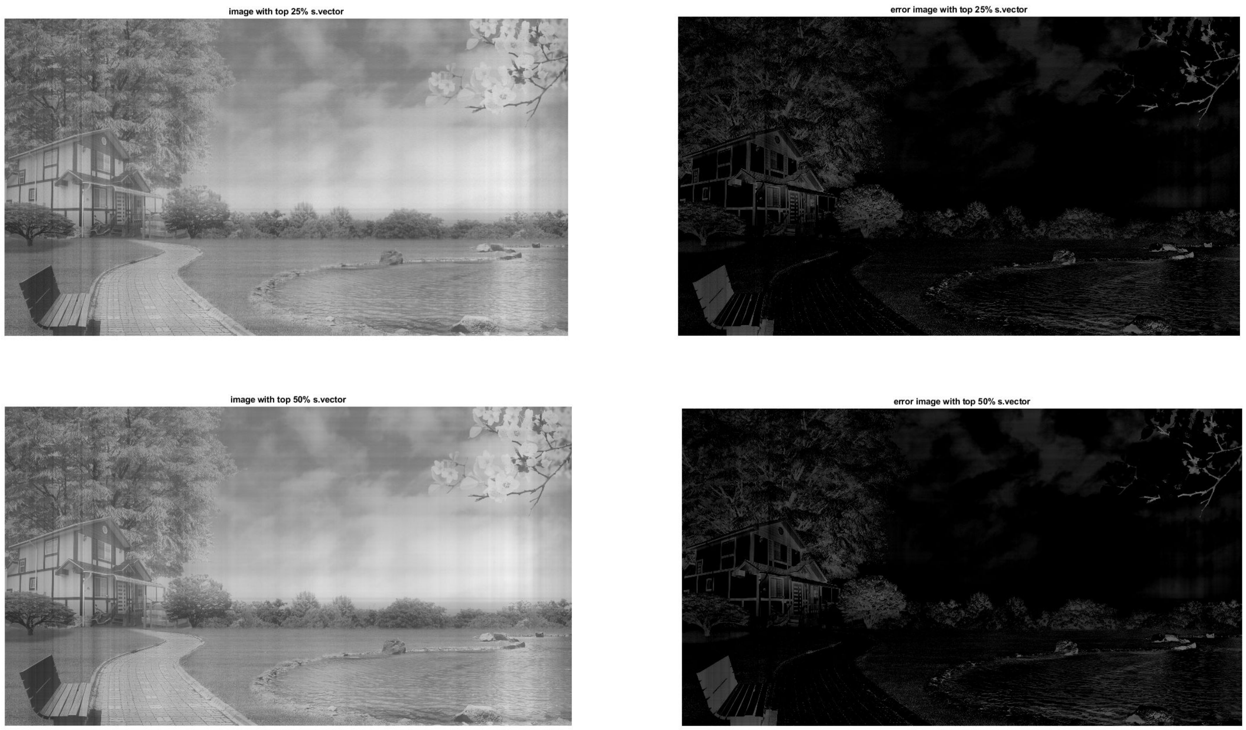


Figure 2.3 Rectangular Images And Corresponding It's Error Image With N = 10%,25%,50% Of Top Singular Vectors.

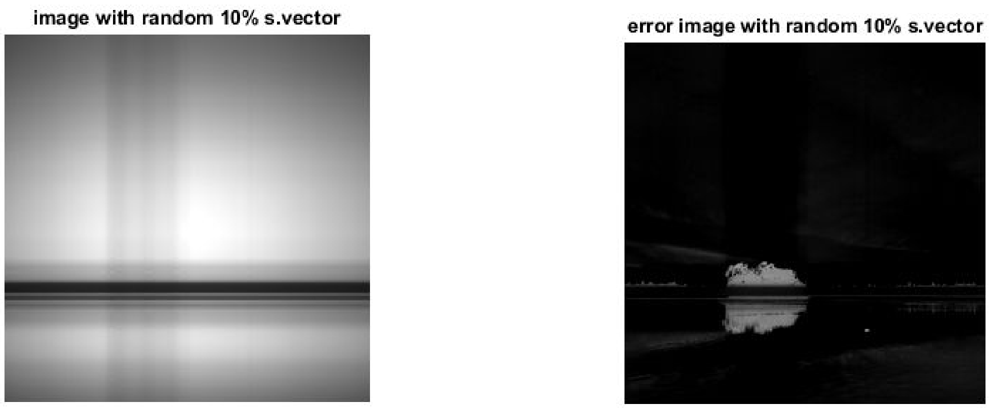
Frobenius norm of Rectangular original image is 894.04617, for square original image is 170.0063, and for different Ns Frobenius norm is given below .

N(%)	10	25	50
Frobenius norm (Quality) of rectangular image	857.8739	858.4640	858.7356
Frobenius norm (Quality) of square image	168.8081	168.8095	168.8097

.

So we can see that as N increases quality of image also increases!!

Part 2: reconstructed image of random 10%(N) singular vectors



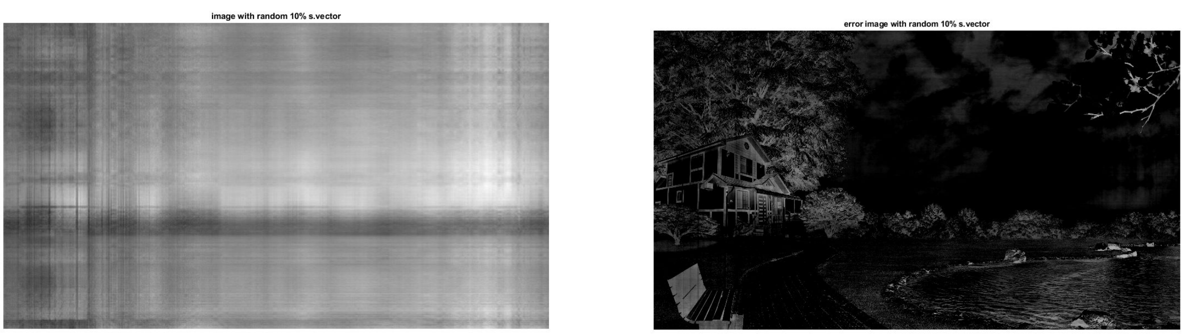


Figure 2.4 Rectangular Images And Corresponding It's Error Image With N = 10%(random) Of Singular Vectors.

Since all singular vectors are equiprobable there is very less chance to select top 10% singular vectors so most of time reconstructed images are blurred. And we post best quality image we found by run of 10 experiment e.x. Frobnenius norm of rect. Image is 852.2026(best) and for square matrix that is 168.4561(best).

Part 3: comparative graph of reconstruction error vs N.

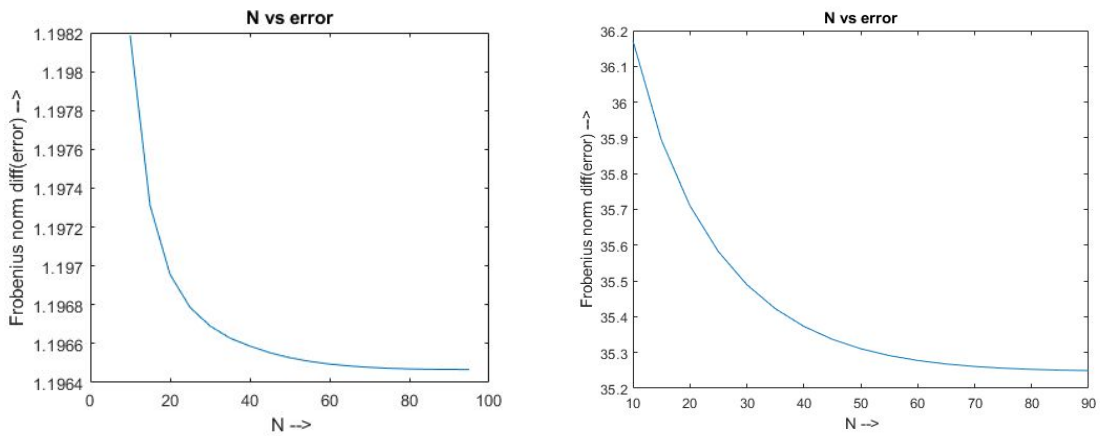


Figure 2.5 Error Vs N Graph Left One Is For Square Image And Right One Is For Rectangular Images.

Problem 3

Part 1 :

a) Given Problem can be converted into Ax = b form, where,

 i) A = Matrix of 0s and 1s. A[i, j] is 1 if i and j are neighbours and if i=j (i.e. they are adjacent and that element itself)
 For example, A = [[1, 1, 1, 0]
 [1, 1, 0, 1]
 [1, 0, 1, 1]
 [0, 1, 1, 1]]

 ii) x = Solution Vector. x[i] = 1 indicates that, toggle the i^{th} tap.

 iii) y = Current state of all taps. That is, if y[i] = 1 then i^{th} tap is open and if y[i] = 0 than i^{th} tap is close. (For our case, y is the vector of all 1s)

b) Solution Sequence x for N = 4 is,
 [0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0]
 That, is close 3^{rd}, 5^{th}, 12^{th}, and 14^{th} tap to close all taps.

c) No, solution sequence is not unique, there are 16 sequences possible

Problem 4

1. Encoding "LINEAR" with,
    a. X1 : [897 1417 3734 3793 229 862]
    b. X2 : [897 1656 2760 6603 229 458]

2. Decoding [927 1345 4006 3913] with,
    a. X1 : "COOL"
    b. X2 : Since X2 is Singular (and X2 Transpose times X2 is also Singular), decoding was possible

3. Decoding [927 1445 3811 3665 708 1081 1778] with,
    a. X1 : "MATHA5k" (Note That, Since Message length is not a multiple of 4, we padded the message with 0 until its length became multiple of 4. Thus, here last three characters of decoded message maybe different from expected as those values depend on what is being added to the original message)
    b. X2 : Since X2 is Singular (and X2 Transpose times X2 is also Singular), decoding was possible