CS5691-PRML A1

Understanding E.V.D and S.V.D

CS19B004 Akkiraju Dharani Gopal

January 30, 2022

1 Motivation

To understand the working of Eigenvalue Decomposition (E.V.D) and Singular Value Decomposition (S.V.D), we perform them on a gray-scale image. We take the top K Eigen Values/Singular Values to reconstruct the image and compare it to the original image using the Frobenius Norm.

2 E.V.D / S.V.D

2.1 Eigen Value Decomposition

Recall that a nonzero vector v of dimension N is an Eigenvector of a square $n \times n$ matrix A if it satisfies a linear equation,

$$Av = \lambda v$$

E.V.D can only be performed on square matrices.The given matrix A is factorized in the form,

$$A = VWV^{-1}$$

where V is the square $n \times n$ matrix whose i^{th} column is the Eigenvector v_i of A, and W is the diagonal matrix whose i^{th} diagonal element corresponding to the i^{th} Eigenvalue λ_i.

2.2 Singular Value Decomposition

S.V.D generalises E.V.D and can be performed on any $n \times m$ matrix.
An $n \times m$ matrix A is factorized into three matrices as,

$$A = U\Sigma V^T$$

where on possible decompostion is as,

- U can be calculated as an $n \times n$ matrix where the i^{th} column is the i^{th} Eigenvector for the matrix AA^T
- Σ can be calculated as an $n \times m$ rectangular diagonal matrix with the diagonal elements corresponding to the i^{th} ($i \leq min(n, m)$) Singular value which can be calculated by taking the square root of i^{th} Eigenvalue of the matrix AA^T (These values will be real and positive as AA^T in our case is real and symmetric).
- V can be calculated as an $m \times m$ matrix where the i^{th} column is the i^{th} Eigenvector for the matrix A^TA

If m = n, these three matrices perform a geometric equivalent of rotation/reflection, scaling and another rotation/reflection to mimic the effect of the original matrix.

2.3 Taking the top K values

In E.V.D /S.V.D the eigen values/singular values represent the weight/contribution along the corresponding axes set by the change in basis.
Hence the Eigen values/ Singular values with higher magnitude contribute more in the reconstruction of the original matrix. Therefore, by removing the values closer to 0, the matrix (and hence the image) can be reconstructed without losing a lot of data .
We sort both the Eigen values and Singular values and use the top k of them to reconstruct the original matrix. For the $n \times n$ matrix,the reconstructed matrix only requires $(n \times k)+(k \times k)+(k \times n)$ values.

3 Experimentation

The given gray-scale image (of dimensions 256×256) is converted into a 256×256 matrix with each value indicating the brightness of the pixel, 0 being black and 255 being white.

3.1 Some Observations

- Considering an $n \times n$ image, taking k Eigen Values/ Singular Values, the reconstructed image requires $2nk + k^2$ values. To reduce space, we need $(n \times n) > 2nk + k^2$, i.e for a $k < \frac{n}{\sqrt{2}+1}$, we need to have a clear image or else this method uses up more space than the original image. (for $n = 256, k \leq 106$).
- As E.V.D might give complex values, magnitudes are taken for sorting. For each conjugate pair, we should include or exclude both of them in the final matrices

3.2 Results

We perform S.V.D and E.V.D on the given image and reconstruct it using only the top K values. On varying the value of k from 1 to 256, we compute the Frobenius norm to indicate the error with both methods and plot them on a graph.

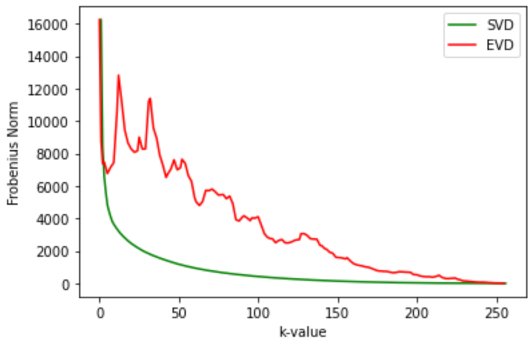
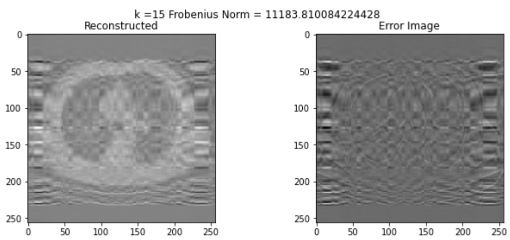
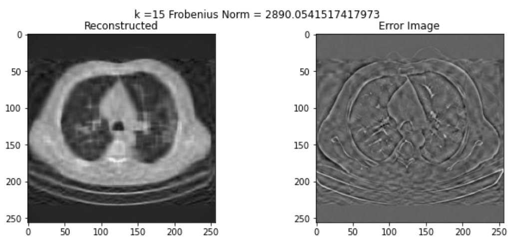


Figure 1: Frobenius norm vs K plot

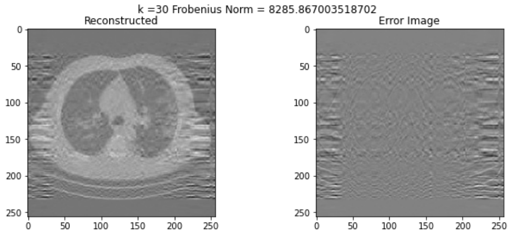
A few example reconstructed images to show the difference between the effectiveness of E.V.D and S.V.D. Here's a link to all the images.



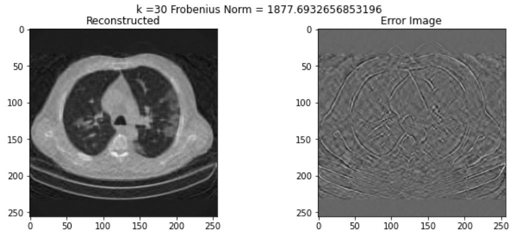
(a) E.V.D with k = 15



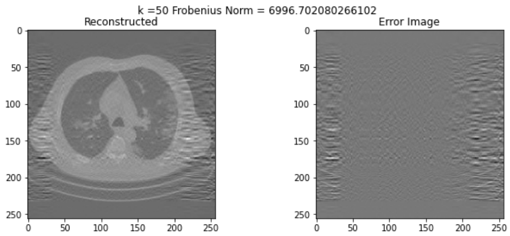
(b) S.V.D with k = 15



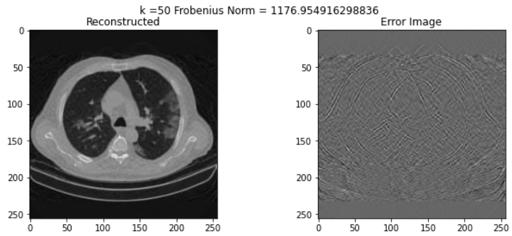
(a) E.V.D with k = 30



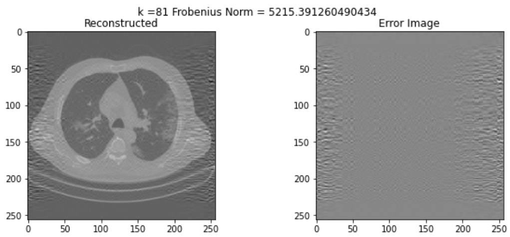
(b) S.V.D with k = 30



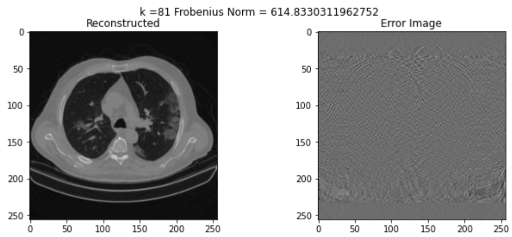
(c) E.V.D with k = 50



(d) S.V.D with k = 50



(e) E.V.D with k = 81



(f) S.V.D with k = 81

Figure 3: With k = 15

4 Inference

- Figure 1 shows the error norm to be more random in the case of E.V.D compared to S.V.D, which is a smooth and continuously decreasing plot.
- S.V.D delivers much lesser losses (as indicated by the norm) for smaller value of k compared to E.V.D.
- The S.V.D appears to have maximum of the information in the largest few singular values and hence the error almost flattens out horizontally after a few k, but the E.V.D appears to still contribute significantly to the image till a much larger k.
- Even with k being as high as 106, the image from E.V.D still has some noise and colours aren't accurate. Whereas for S.V.D by about k = 50, the image is quite crisp and seems closer to the original image.
- On a real matrix, S.V.D, by the nature of the calculation gives only real values whereas E.V.D might gives complex values which result in loss of data when its real component is used in reconstruction.
- S.V.D gives some basic understanding required for Dimensionality reduction and P.C.A(Principle Component Analysis).

⇒ S.V.D is a better technique to preserve information about the image while using lesser space and appears to be more stable and predictable compared to E.V.D