Singular Value Decomposition - How are images compressed ?

Abhik Biswas

November 25, 2021

1 Why Singular Value Decomposition and not Diagonalization ?

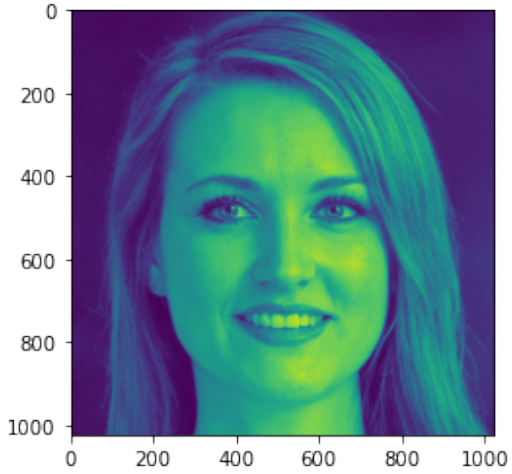
Many of my friends have argued me the reason to study such a complex algorithm, when we can simply diagonalize a matrix and well, I don't need to explain myself further, instead of a typical 256×256 units of space occupation, we come down to only 256 units of space occupation after diagonalization.(For the sake of simplicity, I am only considering that any nonzero number takes up 1 unit of space.) But, is every matrix diagonalizable ?
The answer is no - they are not. Diagonalization of matrices is a separate topic of Linear Algebra and I am not covering it here, but there are many criterions that must be satisfied, and you cannot be sure that every picture you click satisfies each of these criterions. Here is where SVD comes in. Even though every matrix is not diagonalizable, they can be always decomposed into the SVD form,

$$A = Q_1 \Sigma Q_2^T$$

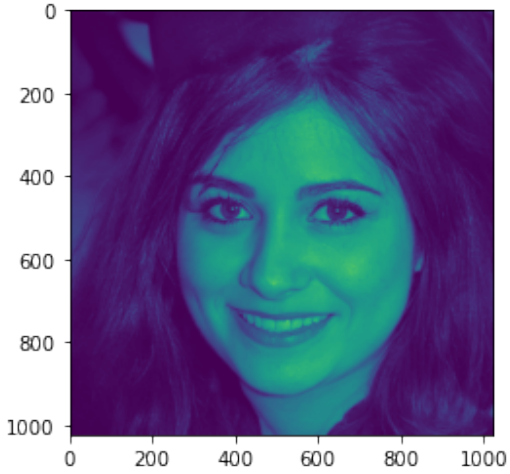
where U contains the eigenvectors of AA^T and V contains the eigenvectors of $A^T A$ and Σ contains the singular values.

2 How does SVD work ?

I will be now continuing with how SVD works, nothing extremely deep into mathematics, but just the basics.



(a) Face 1



(b) Face 2

Figure 1: Images of faces, where the images were first converted to Grayscale and only the first layer was chosen. Consider each of these images as a matrix of size 1024×1024, where each entry $\in \{0, 1, 2, \ldots 255\}$

Consider the matrix A^TA, where size of A is $m \times n$. Then, A^TA is a square matrix of size $n \times n$. A^TA is real & symmetric and hence, all the eigenvalues are real and eigenvectors corresponding to distinct eigenvalues are orthogonal.
Thus, we can easily obtain an orthonormal set of eigenvectors of the matrix A^TA $\{\vec{v_1}, \vec{v_2}, \ldots \vec{v_n}\}$ for eigenvalues $\{\lambda_1, \lambda_2, \ldots \lambda_n\}$

Now, we have

$$A^TAv_i = \lambda_i v_i \quad \forall i = 1, 2, \ldots n$$

and

$$||v_i|| = 1 \quad \forall i = 1, 2, \ldots n$$

since $v_i's$ are orthonormal. So, if we consider $(A^TAv_i) \cdot v_i$, then

$$(A^TAv_i) \cdot v_i = (\lambda_i v_i) \cdot v_i = \lambda_i \tag{1}$$

as $||v_i|| = 1$. Also, $(A^TAv_i) \cdot v_i$ can be written as

$$(A^TAv_i) \cdot v_i = (A^TAv_i)^T v_i = v_i^T A^T Av_i = (Av_i)^T Av_i = ||Av_i||^2 \tag{2}$$

Equating (1) and (2), we get $\lambda_i = ||Av_i||^2$, which means that $\lambda_i \geq 0 \quad \forall i$.
Now, arrange the eigenvalues in their descending orders of magnitude and rewrite as

$$\lambda_{i_1}, \lambda_{i_2} \ldots, \lambda_{i_n}$$

where $\lambda_{i_k} \geq \lambda_{i_j}$ if $i_k < i_j$ and let $\lambda_{i_k} > 0 \quad \forall 0 \leq i \leq r$ and $\lambda_{i_k} = 0 \quad \forall k > r$
Now, we obtain what is known as the singular values σ such that

$$\sigma_i = \sqrt{\lambda_i} \quad \forall i \leq r \tag{3}$$

Hence, we construct the matrix Σ such that

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & \dots 0 \\ 0 & \sigma_2 & 0 & \dots 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \sigma_n \end{bmatrix} \tag{4}$$

Now, let $y_i = \frac{1}{\sigma_i} A v_i$. Then $y_i \in \mathbb{R}^n$ and

$$||y_i|| = \frac{1}{\sigma_i} ||A v_i|| = \frac{\sqrt{\lambda_i}}{\sigma_i} = 1 \tag{5}$$

Also, for $i \neq j$, we have

$$y_i \cdot y_j = \frac{1}{\sigma_i \sigma_j} (A v_i) \cdot (A v_j) = \frac{1}{\sigma_i \sigma_j} (A v_i)^T (A v_j) = \frac{1}{\sigma_i \sigma_j} v_i^T (A^T A) v_j$$

$$\Rightarrow y_i \cdot y_j = \frac{1}{\sigma_i \sigma_j} v_i^T \lambda_j v_j = \frac{1}{\sigma_i \sigma_j} \lambda_j v_i^T v_j = 0$$

Thus,

$$y_i \cdot y_j = 0 \ \ \forall i \neq j \tag{6}$$

Hence, we obtain a set of orthonormal vectors $\{y_1, y_2, \dots y_r\}$, which can be extended to an orthonormal basis of $\mathbb{R}^n$, which is $\{y_1, y_2, \dots y_n\}$.
Let

$$Q1 = \begin{bmatrix} | & | & & | \\ y_1 & y_2 & \dots & y_n \end{bmatrix} \quad y_i \in \mathbb{R}^n \tag{7}$$

and

$$Q_2 = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_n \end{bmatrix} \quad v_i \in \mathbb{R}^n \tag{8}$$

Then,

$$Q_1^T A Q_2 = \begin{bmatrix} — y_1^T — \\ — y_2^T — \\ \vdots \\ — y_n — \end{bmatrix} \begin{bmatrix} | & | & & | \\ A v_1 & A v_2 & \dots & A v_n \end{bmatrix}$$

Therefore, $(Q_1^T A Q_2)_{ij} = y_i^T A v_j$ and while $j \leq r$, $y_j = \frac{1}{\sigma_j} A v_j \Rightarrow A v_j = \sigma_j y_j$

$$\therefore y_i^T A v_j = y_i^T \sigma_j y_j = \sigma_j y_i^T y_j = \begin{cases} \sigma_j & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

and for $j > r$, $||A v_j||^2 = \lambda_j = 0$ (By arrangement of eigenvalues)

$\therefore A v_j = 0 \Rightarrow y_i^T A v_j = 0.$

Hence we obtain, $Q_1^T A Q_2 = \Sigma \Rightarrow A = Q_1 \Sigma Q_2^T$ as Q_1 and Q_2 are orthogonal matrices.

3 Image Size Compression

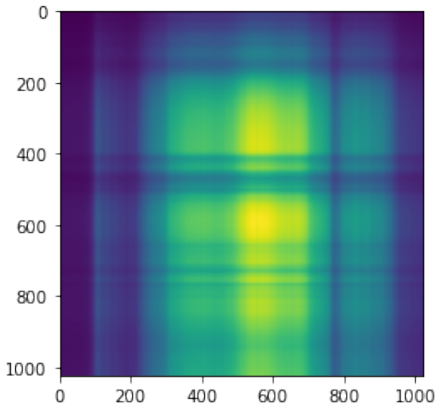
Now that the mathematics is over, let's take a look at how images are compressed. If we include k singular values while compressing an image, then it is said to be a Rank k approximation of the image. For Rank k approximation, let us consider

$$\Sigma_k = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0\dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \sigma_k & \dots 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

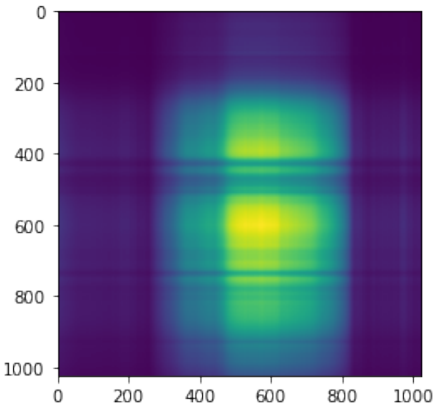
Then the k^{th} approximation of image A is given by

$$A_k = Q_1 \Sigma_K Q_2^T$$

Some approximations of the images shown in Figure(1) are as follows:



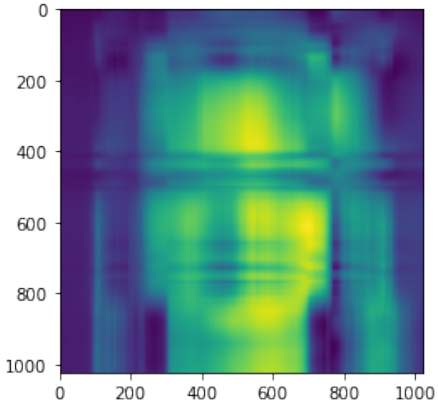
(a) Face 1 Rank 1 Approximation.



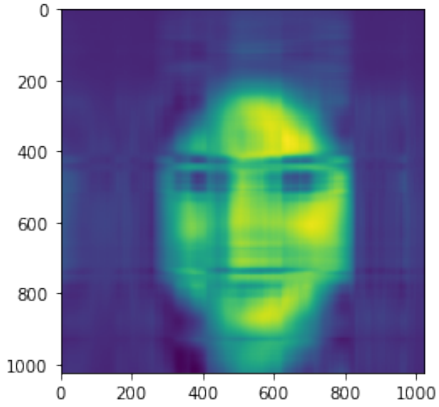
(b) Face 2 Rank 1 Approximation.

Figure 2: Rank 1 Approximation of both the faces

As you will be able to see from the following images, as we keep on increasing the rank of approximation, more details come into the image. Figure(2) is completely uninterpretable, whereas from Figure(3), you can definitely tell that it's a human face! Also, notice that at rank 30 approximation captures most of the facial details and is quite similar to the original images in Figure(1)! Impressive, isn't it? Earlier we required a matrix of size 1024×1024,(space complexity $\mathcal{O}(n^2)$ for a $n \times n$ image.) with each entry being non-zero and it took up $\approx 10^7$ units of space. Now we require only $(2 \times 1024 \times k) + k$ units of storage for a rank k approximation (space complexity $\mathcal{O}(nk)$, $k << n$). Unless k is close to 1024, this is extremely less than what the original image was taking. [1] [2]

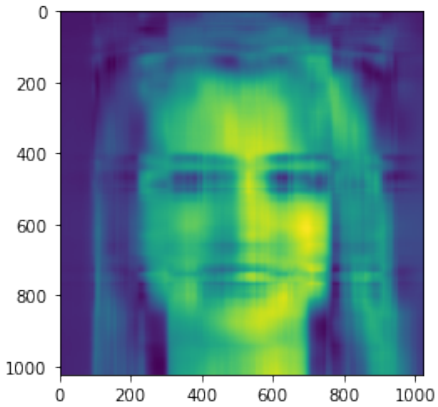


(a) Face 1 Rank 3 Approximation.

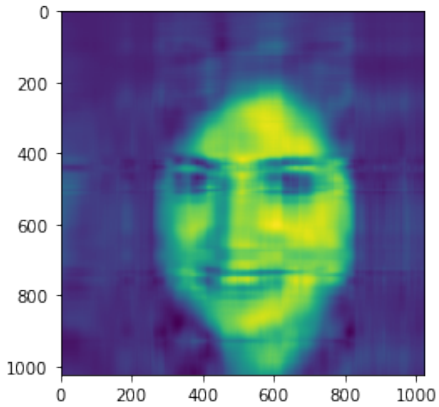


(b) Face 2 Rank 3 Approximation.

Figure 3: Rank 3 Approximation of both the faces

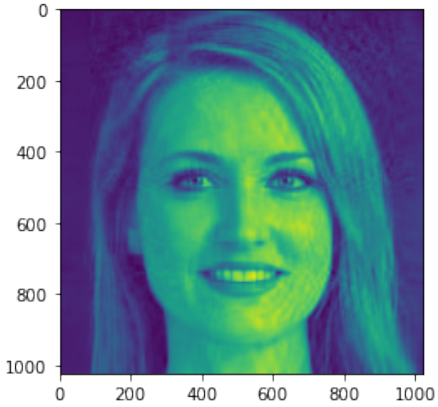


(a) Face 1 Rank 5 Approximation.

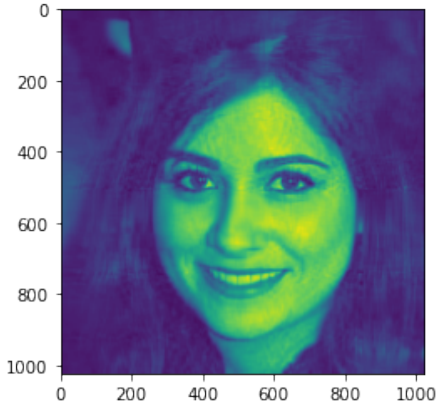


(b) Face 2 Rank 5 Approximation.

Figure 4: Rank 5 Approximation of both the faces



(a) Face 1 Rank 30 Approximation.



(b) Face 2 Rank 30 Approximation.

Figure 5: Rank 30 Approximation of both the faces

[1]All calculations and images are processed using Python 3.8 and NumPy

[2]Facial images are obtained from https://thispersondoesnotexist.com/. People with these faces do not exist and are imagined by GANs.