Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Image Compression using Singular Value Decomposition (SVD)

Arush Chatterjee

Sanskriti School, India

Abstract Images, integral to numerous applications, are encoded as matrices where each element represents a pixel's grayscale intensity. In grayscale images, values range from 0 (representing black) to 1 (indicating white). As image dimensions increase, so does the demand for storage space. Smaller images are easily managed, but larger ones pose challenges. Hence, data compression techniques are applied to mitigate storage consumption. One effective approach involves employing Singular Value Decomposition (SVD) on the image matrix. Through SVD, we create a low-rank approximation for each color channel separately, resulting in a 3-dimensional array that closely approximates the original image. This process achieves image compression while retaining vital image characteristics. This paper illustrates the fundamental concept of SVD and demonstrates its remarkable efficacy in substantially reducing image storage requirements while preserving image quality to a nearly perfect degree. As an illustrative example, we utilized a grayscale image of a bird to showcase how SVD can generate a near-replica of the original image while utilizing only 7.82% of the original image's storage capacity. This underscores the practical importance of SVD in optimizing image storage and transmission.

Keywords SVD, Matrices, Storage

1. Principal of SVD

According to SVD, any n × m matrix, A, can be factorized into $U\Sigma V^T$, where U and V are orthogonal matrices, and Σ is a diagonal matrix comprised of the singular values of A ($\sigma_1 > \cdot\cdot\cdot > \sigma_n > 0$) which appear in descending order along the main diagonal. The eigenvalues of AA^T and A^TA are represented by the numbers $\sigma_1^2 \geq \cdot\cdot\cdot \geq \sigma_n$. The principle of SVD is expressed as a formula below.

$$\mathbf{A = U\Sigma V^T}$$

$$\mathbf{U = [u_1\ \ u_2\ \ldots.u_{n-1}\ \ u_n]} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\sigma_1} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \boldsymbol{\sigma_n} \end{bmatrix} \qquad \mathbf{V^T} = \begin{bmatrix} \mathbf{v_1}^{\mathbf{T}} \\ \mathbf{v_2}^{\mathbf{T}} \\ \ldots \\ \mathbf{v_{n-1}}^{\mathbf{T}} \\ \mathbf{v_n}^{\mathbf{T}} \end{bmatrix}$$

Illustrating SVD using an example –

Let a matrix A = $\begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}$

Then, $A^T = \begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix}$

Calculating AA^T and A^TA we get,

$$AA^T = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 8 & 0 \\ 2 & 0 \end{bmatrix}$$

$$A^TA = \begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix}$$

Now we find the eigenvalues of AA^T and A^TA using the equations -

1. $|AA^T - \lambda I| = 0$
2. $|A^TA - \lambda I| = 0$

It's observed that solving either equation yields two identical eigenvalues (λ). This equivalence arises because a matrix's eigenvalues are derived from its characteristic polynomial, obtained by subtracting the identity matrix from the matrix itself. In the case of square matrices, the determinant remains consistent when calculated for both the matrix and its transpose. Consequently, the characteristic polynomials of A^TA and AA^T are identical if A is square.

Solving both the equations, we get $\lambda_1 = 8$ and $\lambda_2 = 2$

Now that we have λ_1 and λ_2, we can find out σ_1 and σ_2 and hence compute Σ using the relations below –

$\sigma_1 = \sqrt{\lambda_1} = \sqrt{8}$

$\sigma_2 = \sqrt{\lambda_2} = \sqrt{2}$

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} = \begin{bmatrix} \sqrt{8} & 0 \\ 0 & \sqrt{2} \end{bmatrix}$$

Now we find the orthogonal matrix U whose columns are the unit eigenvectors of AA^T. We use the equation $\mathbf{[AA^T - \lambda I]x = 0}$ for the two different eigenvalues ($\lambda_1 = 8$ and $\lambda_2 = 2$) to get the first and second column of U.

Solving the equations –

1. $[AA^T - \lambda_1 I]x_1 = 0$
2. $[AA^T - \lambda_2 I]x_2 = 0$

We get x1 = $\begin{bmatrix} -1 \\ 0 \end{bmatrix}$ and x2 = $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Since U is orthogonal we convert the vectors x_1 and x_2 into their unit vectors by dividing them by their magnitude. i.e.

$u_1 = \frac{x1}{|x1|} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$

$u_2 = \frac{x2}{|x_2|} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Therefore, U = $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$

Now we find the orthonormal matrix V whose columns are the unit eigenvectors of A^TA. We use the equation $\mathbf{[A^TA - \lambda I]x = 0}$ for the two different eigenvalues ($\lambda_1 = 8$ and $\lambda_2 = 2$) to get the first and second column of V.

Repeating the same process we applied to find U, we find that –

$$V = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

Therefore we have found Σ, U, V and can factorize $\mathbf{A = U\Sigma V^T}$ in matrix form as below -

$$\begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{8} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

2. How SVD aids in Image Compression

When we expand on the formula wherein matrix A can be factored into 2 orthogonal matrices (U,V) and 1 diagonal matrix (Σ), we can express our initial matrix A as a combination of simpler rank 1 matrices.

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

$$A = [u_1 \ u_2 \ \cdots \ u_{n-1} \ u_n] \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & \sigma_{n-1} & 0 \\ 0 & 0 & 0 & 0 & \sigma_n \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \ldots \\ v_{n-1}^T \\ v_n^T \end{bmatrix}$$

$$A = u_1\sigma_1 v_1^T + u_2\sigma_2 v_2^T + \cdots + u_{n-1}\sigma_{n-1}v_{n-1}^T + u_n\sigma_n v_n^T$$

$$A = \sum_{i=1}^{n} u_i \sigma_i {v_i}^T$$

We know that the diagonal matrix Σ is arranged from the largest value σ_1 at the top to the smallest value σ_n at the bottom. Therefore, only the first few summations of $A = \sum_{i=1}^{n} u_i \sigma_i {v_i}^T$ will have the largest values. This holds key to data compression since we can literally approximate A by adding only the first few values of the summation

We obtain a new rank-k matrix, where k represents the total number of rank-1 matrices combined to form the approximation A_{approx}.

$$A_{approx} = \sum_{i=1}^{k} u_i \sigma_i {v_i}^T$$

As the value of k rises, image quality improves, but so does the memory requirement. Consequently, smaller k values are favored. Remarkably, near-perfect replication of a geometric image is achievable with rank-1 or rank-2 SVD. Storing just the initial two columns of U and V, along with their singular values, allows exact replication of a square image, utilizing a mere 0.02% of the original storage capacity.

3. Showcasing SVD Compression using an Example

Let's demonstrate how image compression by k rank SVD works using a 512 × 512 pixel colored image of a bird. Given below is the original colored image when k=512.

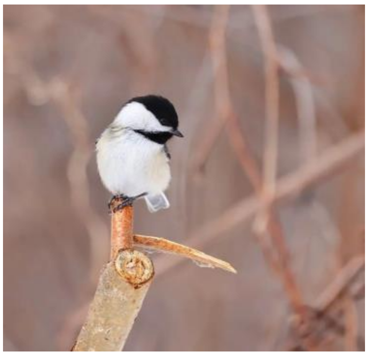

Figure 1: Original, Colored Image of Bird

We separate the grayscale values for each color to create 3 independent 2-dimensional matrices to perform an SVD for each color separately. Since *k*=20 is the lowest k value which provided the closest approximation to the original image, let's use that as our primary example to see the practicality and effectiveness of SVD in Image Compression and also for our calculation of Compression Ratio in this case.

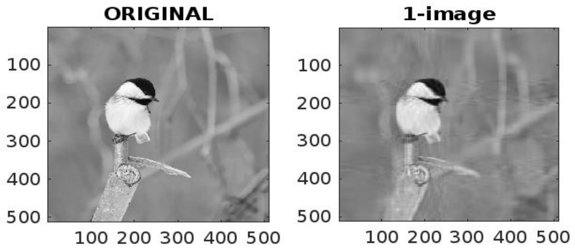

Figure 2: The image on the left is a grayscale representation of the original, colored image. The image on the right is the SVD compressed at k=20

As seen above, we have created an almost perfect replica of the original gray scale image of the bird. Using mathematics we can show that we have created this almost perfect replica using only 7.82% of the original image's memory.

Let the memory a m × n pixel non-compressed image, I_m require = mn

$$I_m = mn$$

However when we apply rank k SVD, we actually apply it on the three matrices, U, Σ, and V.

Let's consider matrix U. Initially, U is an m × m matrix, but we only require its first k columns. This implies that U can be condensed into an m × k matrix. We can denote its storage as U_m, represented by –

$$U_m = mk$$

Likewise, we're interested in only the initial k rows of matrix V^T. Therefore, we can optimize storage by condensing V into an n×k matrix, denoted as V_m, simplifying the representation -

$$V_m = nk$$

Finally, as we are retaining solely the initial k columns of U and V, we require only the first k singular values from the diagonal matrix Σ. This condensed storage, denoted as Σ_m, can be represented by –

$$\Sigma_m = k$$

If A_m is the total storage of the three orthogonal matrices. It can be represented as -

$A_m = U_m + V_m + \Sigma_m$

$A_m = mk + nk + k$

$A_m = k(m + n + 1)$

For the example of the bird we've taken above,

$I_m = 512 \times 512 = 262{,}144$ bytes

$A_m = 20(512 + 512 + 1) = 20{,}500$ bytes

For the example above, the Compression Ratio, which is simply the ratio of the storage of the non-compressed image to the storage of the SVD compressed image, comes out to be 12.78

4. Conclusion

This demonstration of image compression through k-rank SVD highlights the remarkable efficiency of this technique. We achieved an almost perfect replica of the original image and through mathematical analysis, revealed the substantial reduction in memory usage. This efficiency becomes particularly evident when considering the storage requirements of the three orthogonal matrices involved in the SVD process. The total storage significantly diminishes compared to the storage needed for the uncompressed image, leading to a Compression Ratio of 12.78 for the example of the bird image taken above.

As technology advances and the demand for high-quality images in various applications continues to rise, the role of SVD in image compression becomes even more critical. Its capacity to create compact representations of images while retaining essential details makes it indispensable in fields such as medical imaging, satellite imagery, and multimedia communication. In essence, this paper underscores the vital role of SVD as a transformative tool in image compression. By harnessing the power of SVD, we can pave the way for more efficient, accessible, and meaningful handling of visual data, shaping the future landscape of image processing and communication.

References

[1]. Kurics T. (2016). Image Compression Using Singular Value Decomposition. Medium.

[2]. Baumann T. (2023). SVD Image Compression Demo. timbaumann