Two Applications of Singular Value Decomposition

Qianyu Zheng

November 19th 2023

1 Singular Value Decomposition

For any linear transformation $T : V^n \to W^m$, there exists orthonormal basis $e_1, e_2, \ldots, e_m$ for W and $f_1, f_2, \ldots, f_n$ for V such that T has a rectangular diagonal matrix with diagonal entries $\sigma_1, \sigma_2, \ldots, \sigma_{\min(n,m)}$ with regard to these basis. [1]

We can rewrite the above statement in terms of matrices: for any matrix $A^{m\times n}$, we can decompose it to:

$$A = U_2 D U_1^*$$

- U_2 is a $m \times m$ matrix with columns that form a orthonormal basis of V^m.
- U_1^* is a $n \times n$ matrix with rows that form a orthonormal basis of W^n.
- D is a rectangular diagonal $n \times m$ matrix with diagonal entries $\sigma_1, \sigma_2, \ldots, \sigma_{\min(n,m)}$

Based on Singular Value Decomposition (abbreviated as SVD), each $m \times n$ matrix $A^{m\times n}$ with equal numbers of rows and columns can be written as:

$$A = \begin{pmatrix} | & & | \\ e_1 & \ldots & e_m \\ | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & \ldots & 0 \\ 0 & \sigma_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_{\min(m,n)} \end{pmatrix} \begin{pmatrix} — f_1^T — \\ — f_2^T — \\ \vdots \\ — f_n^T — \end{pmatrix}$$

The decomposition can also be written in the following way as a sum of rank-1 matrices. $l = \min(m,n)$ in the following expression:

$$A = \sigma_1 \begin{pmatrix} | \\ e_1 \\ | \end{pmatrix} \begin{pmatrix} — f_1^T — \end{pmatrix} + \cdots + \sigma_l \begin{pmatrix} | \\ e_l \\ | \end{pmatrix} \begin{pmatrix} — f_l^T — \end{pmatrix} = \sum_{i=1}^{l} \sigma_i \begin{pmatrix} | \\ e_i \\ | \end{pmatrix} \begin{pmatrix} — f_i^T — \end{pmatrix}$$

Therefore, we can approximate the matrix A with the first k rank-1 matrices since the singular values follow $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_l$. The matrices with larger singular values contribute most to the matrix A, and the last terms with smaller singular values have much lower contributions.

The approximation of A, A' will be:

$$A \approx A' = \sum_{i=1}^{k} \sigma_i \begin{pmatrix} | \\ e_i \\ | \end{pmatrix} \begin{pmatrix} — f_i^T — \end{pmatrix}, k \leq l$$

As proven during the lecture, A' is the best approximation of A with matrices of rank k.

2 SVD in image compression

As we know, any image contains three color channels: red, green, and blue. Each channel can be represented as a $m \times n$ matrix with values ranging from 0 to 255. We will now compress the matrix A by performing SVD on each channel and picking only the first k singular values. [2]

For each channel, the original matrix size is $m \times n$, so the size of the original image is proportional to $3mn$. After performing SVD and selecting only k largest singular values, we have k rank-1 matrix in the form:

$$\sigma_i \begin{pmatrix} | \\ e_i \\ | \end{pmatrix} \begin{pmatrix} — f_i^T — \end{pmatrix}$$

Each matrix needs to store $1 + m + n$ values, since σ_i is one value, each vector e_i has m values, and f_i has n values. Thus, after compression, the size of the image is proportional to $3k(1 + m + n)$, which is significantly smaller than $3mn$.

The following image demonstrates applying different compression factor k to compress an image of the Tech Tower:

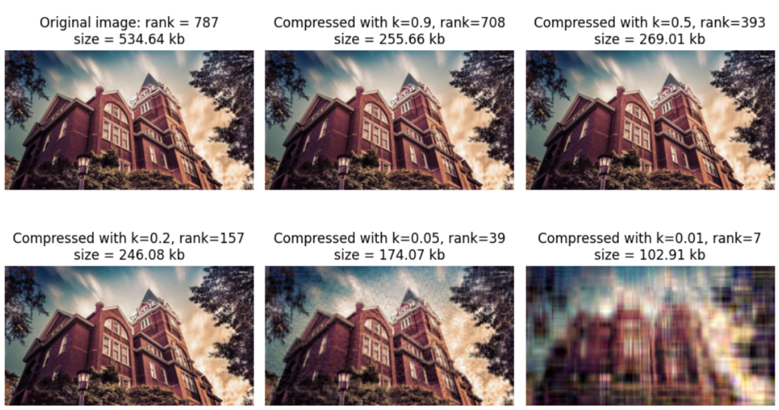

Figure 1: Image compression using SVD

3 SVD in image denoising

Gaussian noise has a probability density function (pdf) that follows a normal distribution, characterized by mean μ and variance σ^2. This gives it an isotropic, uniform effect across the image.

The key theory that enables noise removal is that the singular vectors can be split into two components:

1. Principal components - Corresponding to large singular values, these contain most of the meaningful image content.

2. Noisy components - Corresponding to small singular values, these mostly contain noise.

The noise components arise because noise is spread across all the eigenvectors. However, the eigenvalues associated with the principal components are much larger.

By reconstructing the image only from the principal components (filtering out small singular values), we remove the vectors that mostly contain noise. This filters out a significant portion of noise while retaining key image features and enables SVD to be an effective tunable filter for Gaussian denoising based on mathematically sound matrix decomposition theory.

Notably, the number of principal components that are retained determines how much of the original image is preserved. Using more principal components preserves more information but retains more noise spreading among the components.

Conversely, keeping fewer principal components filters out more noise but also removes some finer details of the original image. This creates a tradeoff between noise reduction and preservation of details. [3]

The following image demonstrates using SVD to denoise a noisy image of the Bobby Dodd Stadium at Georgia Tech.

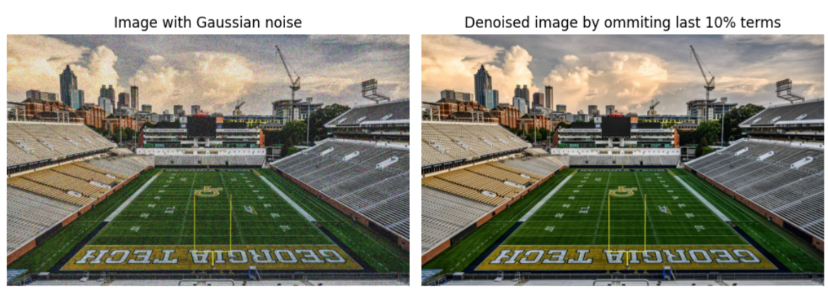

Figure 2: Image denoising using SVD

Code Availability

All the code and images involved in this paper are available on github.

References

[1] Sheldon Axler. *Linear algebra done right.* Springer International Publisher, 2015.

[2] Tim Baumann. Image compression with singular value decomposition, 2023.

[3] Ajit Rajwade. Image denoising using the higher order singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):849–862, 2013.