DOI: 10.54254/2753-8818/2025.GL24952

Singular Value Decomposition

Xinyi Zhao

Tandon School of Engineering, New York University, Brooklyn, USA
xz3831@nyu.edu

Abstract. Singular Value Decomposition (SVD) is a very important matrix factorization technique in linear algebra which generalizes the eigenvalue decomposition to both non square and non symmetric matrices. This report explains the theoretical foundation of SVD by numerical examples and the comparison of SVD with eigenvalue decomposition on the basis of versatility. Theoretical derivations, including proofs of SVD existence and uniqueness, are presented with practical implementation using Python. Experiments include image compression via truncated SVD and dimensional reduction on the Iris dataset. The results indicate that only top k singular values are enough to retain the essential data features and reduce storage requirements. For example, image compression with 50 singular values results in a MSE of 0.05 and visually clear images, as well as 4D data can be reduced into 2D without losing discriminative patterns. Findings confirm that SVD is computationally stable and efficient, and provides robust solution for rank approximation, noise reduction, and feature extraction. The study demonstrates that SVD's capability to break down data into intelligible components make it to remain relevant in big data analytics, scientific computation and artificial intelligence, with foreseeable future improvements also likely to increase its applications.

Keywords: Matrices, Singular Value Decomposition, Rank, Orthogonal, Eigenvalue

1. Introduction

Singular Value Decomposition (SVD) is a fundamental matrix factorisation technique in linear algebra, generalizing eigenvalue decomposition to non-square and non-symmetric matrices. It decomposes any real or complex matrix into one or more matrices whose size is less than the original one A of size m×n into three matrices, providing deeper insights into structural properties such as rank, nullity, and geometric transformations [1]. SVD has wide applications in different areas, such as signal processing, data compression, machine learning, and computer vision. It is particularly valuable for solving least squares problems, reducing dimensionality, and reducing noise on datasets. It also finds applications in principal component analysis (PCA), search algorithms, and image processing. This paper discusses the mathematical foundations of SVD, its geometric interpretation, existence, and uniqueness [2]. The differences between SVD and eigenvalue decomposition and some practical use of it in python are discussed. The real-world use, the applications of SVD, such as image compression and reductions of data dimensionality are also discussed.

As for basic concepts:

1. Symmetric Matrix: A matrix A is called symmetric when it is equal to its transpose.

DOI: 10.54254/2753-8818/2025.GL24952

Example: $A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$

2. Eigenvalue Decomposition: A symmetric n×n matrix ‘A’ having ‘n’ real eigenvalues and an orthonormal basis of eigenvectors can be expressed as:

$$A = V\Lambda V^{-1} \tag{1}$$

Here, V is the matrix with columns as eigenvectors, and Λ is the diagonal matrix with eigenvalues as diagonal elements. This is called eigenvalue decomposition [3].

3. Singular Value Decomposition: This is a generalisation of eigenvalue decomposition with the requirement that the matrix is not necessarily symmetric or even square. A matrix 'A' can be represented in the form of a matrix factorization of order $m \times n$ [4]. It is a linear algebra method of decomposition of a real or complex matrix into three matrices

Before studying it in detail, an exploration of the components of Singular Value Decomposition (SVD) is presented.

2. Geometric interpretation of SVD

The SVD is geometrically seen as the fact that the image of the unit sphere under any m x n matrix is a hyperellipse [5]. The SVD is applicable to both real and complex matrices. However, in describing the geometric interpretation, it is usually assumed that the matrix is real.

The term "hyperellipse" represents an unfamiliar way of describing m-dimensional ellipses within their generalized format. A hyperellipse in $\mathbb{R}^m$ is formed by stretching the $\mathbb{R}^m$ unit sphere with factors $\sigma_1, \ldots, \sigma_m$ (maybe zero) in the orthogonal directions $u_1, \ldots, u_m \in \mathbb{R}^m$. The vectors u_i are unit-length vectors since $\|u_i\|_2 = 1$. The vectors $\{\sigma_i u_i\}$ function as the principal semiaxes of the hyperellipse, delivering the lengths $\sigma_1, \ldots, \sigma_m$. When A achieves rank r, then exactly r of the σ_i will become non-zero values, and in this context, when m has a minimum value of n, the maximum allowed number of non-zero σ_i will equal n [6]. The unit sphere refers to the Euclidean sphere standard in n-space while using the 2-norm definition; thus, it is denoted as S. Through the transformation A, the image of S contains the shape of a hyperellipse that is defined.

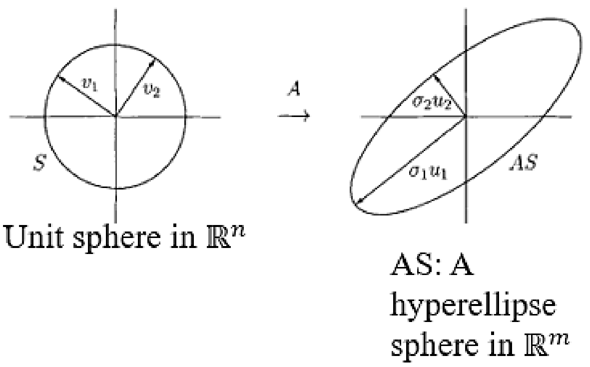

Figure 1: SVD of a 2×2 matrix [3]

Let $A \in \mathbb{R}^{m\times n}$ and consider A maps an input vector in $A \in \mathbb{R}^n$ to an output vector in $\mathbb{R}^m$.
Let $V = \{v_1, \cdots, v_n\}$ be an orthonormal matrix of $\mathbb{R}^n$
Let $U = \{u_1, \cdots, u_m\}$ be an orthonormal matrix of $\mathbb{R}^m$

DOI: 10.54254/2753-8818/2025.GL24952

Let $\{\sigma_1, \cdots, \sigma_m\}$ be a set of m scalars with $\sigma_i \geq 0, i = 1, \cdots m$

Then, $\sigma_i \mu_i$ is the i th principal semiaxis with length σ_i in $\mathbb{R}^m$.

Now, if $\mathrm{rank}(A) = r$, then exactly r of $\{\sigma_1, \cdots, \sigma_m\}$ are nonzero and exactly $m - r$ of σ_i 's we zero.So, if $m \geq n$, then $\mathrm{rank}(A) \leq n$ i.e., at most n of σ_i 's are nonzero.

For simplicity, let's assume $m \geq n$ and $\mathrm{rank}(A) = n$

Definition 1: The singular values are the lengths of the n principal semiaxes of the hyperellipsoid AS (As shown in Figure 1)

The convention is: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$

Definition 2: The n left singular vectors of A are the unit vectors $\{u_1, \cdots, u_n\}$ in $\mathbb{R}^m$ along the principal semiaxes of AS. So, $\sigma_i u_i$ is the i the largest principal semiaxis of AS.

Definition 3: The n right singular vectors of A are the unit vectors $\{v_1, \cdots, v_n\} \in S$ which are the preimages of the principal semiaxes of AS, i.e.,

$A\dot{v}_i = \sigma_i u_i \; i = 1, \cdots, n.$

3. Reduced SVD

Consider $\hat{U} = [u_1 \quad \cdots \quad u_n]_{m \times n}$ and $\hat{\Sigma} = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \end{bmatrix}_{n \times n}$

Consider $[A]_{m \times n}[v_1 \quad \cdots \quad v_n]_{n \times n} = [u_1 \quad \cdots \quad u_n]_{m \times n} \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \end{bmatrix}_{n \times n}$

So,

$$AV = \hat{U}\hat{\Sigma} \tag{2}$$

Now, since V is an orthogonal matrix then:

$$A = \hat{U}\hat{\Sigma}V^T \tag{3}$$

4. Full SVD

It is given that $\hat{U} \in \mathbb{R}^{m \times n}$ in the reduced SVD with $m \geq n$. This implies that the column vectors of $\hat{U}$ do not form an ONB of $\mathbb{R}^m$ unless $m = n$.

$\Rightarrow$ Remedy: Adjoin $m - n$ ON vectors to $\hat{U}$ to form an orthogonal matrix U.

Then $\hat{\Sigma}$ must be changed to $\Sigma \in \mathbb{R}^{m \times n}$

Hence,

$$A = U\Sigma V^\top \tag{4}$$

This is the full SVD of A

For non-full rank matrices, i.e., $\mathrm{rank}(A) = r < \min(m, n)$, only r positive singular values. So,

$\Sigma = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \\ \cdots 0 & 0 & 0 \end{bmatrix}$, in this case $m \geq n$

and $\begin{bmatrix} \sigma_1 & \cdots & 0 & : & 0 \\ \vdots & \ddots & \vdots & : & 0 \\ 0 & \cdots & \sigma_n & : & 0 \end{bmatrix}$, in this case $m \leq n$

when $m = n$,. it's invertible and nonsingular theoretically.

5. Application of SVD

5.1. Image compression

An image is depicted as a matrix where every element corresponds to pixel values.

SVD is applied to decompose the image matrix A into three matrices U, Σ, and V^T [7]. This makes the largest individual singular values and associated vectors can represent the original image with fewer data, making the file smaller but also resulting in some information loss(As shown in Figure 2).

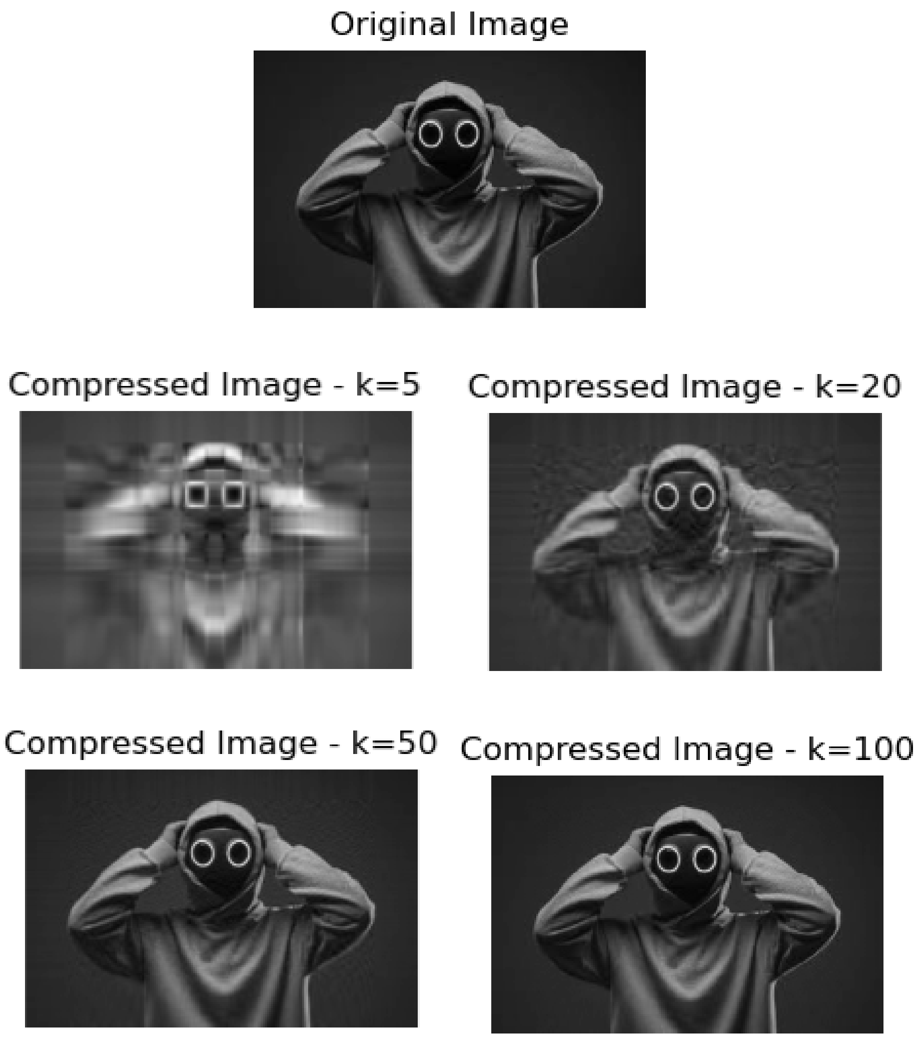

Figure 2: Compressed Images for different values of k

DOI: 10.54254/2753-8818/2025.GL24952

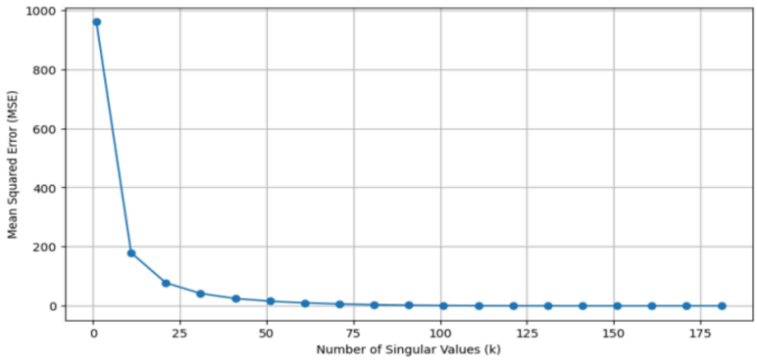

Figure 3: MSE vs number of singular values

Result: The Figure 3 shows the Mean Squared Error of image restorations against the number of singular values (k) employed in Singular Value Decomposition (SVD). It points to a steep drop in MSE as k is raised from a low value, reflecting considerable improvement in image quality (as shown in Figure 4). The MSE stabilizes as k continues to rise, implying that there are diminishing gains in image quality improvement. Hence, an optimal k value is 175, where image quality meets compression efficiency.

5.2. Data dimensionality reduction

Use SVD to perform dimensionality reduction on a dataset for visualization or further analysis.

SVD compressed the 4D data down into 2D while keeping an important structure.

It works:

1. Exploring high-dimensional data: According to Chiu [8], humans do not have the 3D vision and SVD enables to see that.

2. Use how to improve results for machine learning (SVM, k-NN), and reduce noise.

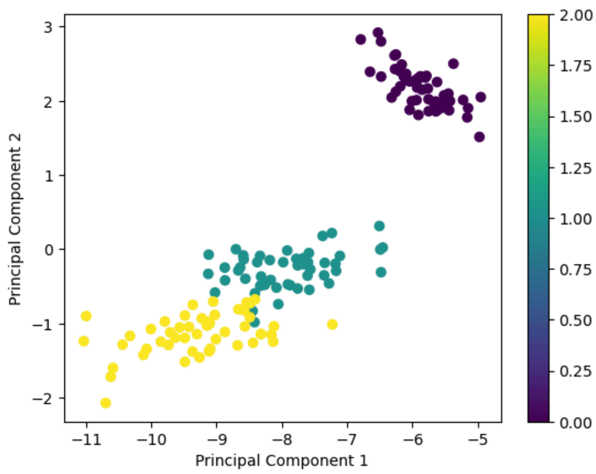

Figure 4: Dimensionality reduction using SVD (iris dataset)

6. Limitations and conclusion

SVD turns out to be a useful and an universal tool from linear algebra, and can provide a sound, robust tool set to analyze and manipulate matrixes. As a result SVD will allow us to decompose a matrix into orthogonal and diagonal components and to give a geom SD interpretation of linear transformations and characterise linear transformations (rank, nullity, principal directions of variation). The application of eigenvalue decomposition is permitted only to square matrices, and their extensions, whereas SVD can be applied to any matrices (square, nonsquare, symmetric or asymmetric) and is therefore very convenient to use in modern computational mathematics. The Singular Value Decomposition (SVD) exhibits remarkable practicality, finding applications in diverse domains such as image compression, noise reduction, data analysis, and dimensionality reduction. Python-based implementations have demonstrated that SVD can effectively reduce the dimensionality of high-dimensional data while preserving a substantial amount of information, minimizing significant information loss. It is concluded that SVD is central to numerical linear algebra and to data science, and provides an efficient way to obtain solutions to difficult engineering, machine learning, and scientific computing problems. This approach is likely to remain relevant in the world, where there's big data processing, as it can identify meaningful patterns of the data and is stable in computing. This demonstrates that the advancements will make the use and efficacy of SVD applicable and efficient in new fields.

Singular Value Decomposition (SVD) is a powerful technique to solve matrix problems but is computationally expensive ($O(\min(mn^2, m^2n))$ complexity), noisy and memory constraint for large datasets[9][10]. In addition, it does not work well with nonlinear structures. Several aspects of future research regarding scalable SVD using parallel and randomized methods and robust variants that are resistant to noise remain for future investigate. Other nonlinear extraction approaches (integration with deep learning) can be combined with adaptive rank selection techniques to determine the superior dimensionality reduction. The applications of quantum SVD include large scale computations and could be federated learning and biomedical signal processing. The appropriate of these limitations will enhance SVD's efficiency and will extend its reach in information technology and data science.

References

[1] Carroll, J. D., & Green, P. E. (1996). Decomposition of Matrix Transformations: Eigenstructures and Quadratic Forms. Mathematical Tools for Applied Multivariate Analysis (Revised Edition), 194-258. https: //doi.org/10.1016/B978-012160954-2/50006-8

[2] Wei, T., Ding, W., & Wei, Y. (2024). Singular value decomposition of dual matrices and its application to traveling wave identification in the brain. SIAM Journal on Matrix Analysis and Applications, 45(1), 634-660. https: //doi.org/10.1137/23M1556642

[3] Hui, J. (2019). Machine Learning — Singular Value Decomposition (SVD) & Principal Component Analysis (PCA). Medium. https: //jonathan-hui.medium.com/machine-learning-singular-value-decomposition-svd-principalcomponent-analysis-pca-1d45e885e491

[4] Kuttler, K. (2021). A First Course in Linear Algebra. Brigham Young University. https: //openlibrary-repo.ecampusontario.ca/jspui/bitstream/123456789/896/2/Kuttler-LinearAlgebra-AFirstCourse-2021A.pdf

[5] Trefethen, L. N., & Bau, D. (2021). Numerical linear algebra. Society for Industrial and Applied Mathematics. https: //www.stat.uchicago.edu/~lekheng/courses/309/books/Trefethen-Bau.pdf

[6] Lloyd, N. T., & David, B. (1997). Numerical linear algebra. Clinical Chemistry, iii edition, UK: Society for Industrial and Applied Mathematics.

[7] Sauer, T. (2011). Numerical analysis. Addison-Wesley Publishing Company.

[8] Chiu, C. H., Koyama, Y., Lai, Y. C., Igarashi, T., & Yue, Y. (2020). Human-in-the-loop differential subspace search in high-dimensional latent space. ACM Transactions on Graphics (TOG), 39(4), 85-1. https: //doi.org/10.1145/3386569.3392409

[9] Schmidt, D. (2020). A Survey of Singular Value Decomposition Methods for Distributed Tall/Skinny Data. arXiv preprint arXiv: 2009.00761v1 [cs.MS]. Retrieved from https: //arxiv.org/abs/2009.00761v1

[10] Golub, G. H., & Van Loan, C. F. (2013). Matrix Computations (4th ed.). Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press. Retrieved from https: //math.ecnu.edu.cn/~jypan/Teaching/books/2013%20Matrix%20Computations%204th.pdf