**Singular Value Decomposition (SVD) and Low-Rank Approximation**

**1. Introduction**

Singular Value Decomposition (SVD) is a powerful technique in linear algebra used to decompose a matrix into three components: UUU, SSS, and VTV^TVT, where UUU and VVV are orthogonal matrices and SSS is a diagonal matrix. SVD is used extensively in data science, signal processing, and machine learning, particularly for dimensionality reduction and matrix approximation. In this report, we demonstrate how to perform SVD on a user-provided matrix, reconstrusct the matrix, and compute a low-rank approximation of the matrix using the top kkk singular values.

**2. Objective**

The main objectives of the implemented code are:

* To perform Singular Value Decomposition (SVD) on a matrix.
* To reconstruct the original matrix from the SVD components.
* To generate a low-rank approximation of the matrix by selecting the top kkk singular values.

**3. Steps Involved**

The code follows these steps:

1. **Matrix Creation**:
   * A user-defined square matrix of size n×nn \times nn×n is created by inputting the values for each element of the matrix.
2. **Singular Value Decomposition**:
   * The matrix is decomposed into three components using SVD:
     + UUU (left singular vectors),
     + SSS (singular values),
     + VTV^TVT (right singular vectors).
   * The decomposition is performed using np.linalg.svd.
3. **Matrix Reconstruction**:
   * Using the SVD components, the original matrix is reconstructed as: Areconstructed=U⋅Σ⋅VTA\_{\text{reconstructed}} = U \cdot \Sigma \cdot V^TAreconstructed​=U⋅Σ⋅VT

where Σ\SigmaΣ is the diagonal matrix formed from the singular values.

1. **Low-Rank Approximation**:
   * A low-rank approximation is generated by selecting the top kkk singular values from the decomposition and reconstructing the matrix with reduced rank: Alow-rank=Uk⋅Σk⋅VkTA\_{\text{low-rank}} = U\_k \cdot \Sigma\_k \cdot V\_k^TAlow-rank​=Uk​⋅Σk​⋅VkT​

where UkU\_kUk​, SkS\_kSk​, and VkV\_kVk​ represent the top kkk singular vectors and singular values.

**4. Code Explanation**

* **Matrix Creation**: The matrix is created using the function create\_matrix(n), where n is the size of the matrix. The user inputs the values for each element, and the matrix is returned as a NumPy array.
* **SVD Decomposition**: The function SVD\_Decomposition(l) uses the NumPy function np.linalg.svd() to decompose the matrix l into UUU, SSS, and VTV^TVT.
* **Matrix Reconstruction**: The function Reconstruction(U, S, VT) reconstructs the matrix by multiplying the matrices UUU, SSS, and VTV^TVT.
* **Low-Rank Approximation**: The function Low\_rankApproximation(U, S, VT, k) performs the low-rank approximation by selecting the top kkk singular values and their corresponding vectors and reconstructing the matrix with reduced rank.

**5. Output**

The code generates the following outputs:

* **Original Matrix**: The matrix input by the user.
* **SVD Components**: The left singular vectors (UUU), singular values (SSS), and right singular vectors (VTV^TVT).
* **Reconstructed Matrix**: The matrix reconstructed using the SVD components.
* **Low-Rank Approximation**: A rank-kkk approximation of the matrix, based on the top kkk singular values.

**6. Example Execution**

For a matrix of size 3x3, the user might input the following values:

Copy code

1 2 3

4 5 6

7 8 9

The SVD decomposition of the matrix will result in:

* UUU (left singular vectors),
* SSS (singular values),
* VTV^TVT (right singular vectors).

After reconstruction, the original matrix will be approximated, and the user will be prompted to enter a rank kkk for the low-rank approximation. For example, if k=2k = 2k=2, a 2-rank approximation of the matrix will be computed and displayed.

**7. Advantages and Applications**

* **Data Compression**: By using a low-rank approximation, significant data compression can be achieved without losing much information, which is especially useful in fields like image processing, natural language processing, and recommender systems.
* **Noise Reduction**: Low-rank approximations can also be used to reduce noise in data, as higher singular values often correspond to noise in the matrix.
* **Dimensionality Reduction**: SVD is commonly used in principal component analysis (PCA) for dimensionality reduction.

**8. Conclusion**

This code demonstrates the application of Singular Value Decomposition (SVD) for matrix decomposition, reconstruction, and low-rank approximation. By retaining only the most significant singular values, it is possible to generate approximations of a matrix with reduced rank. This technique has various applications in data analysis, machine learning, and signal processing, where dimensionality reduction and noise reduction are often required.