Chapter 3: Singular Value Decomposition (SVD)

Aymen Negadi

1 Definition

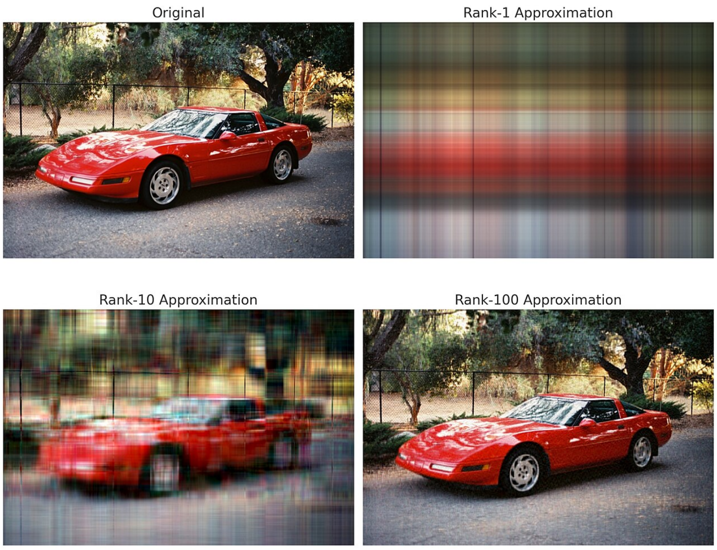
Singular Value Decomposition (SVD) factorizes any real or complex matrix $\mathbf{A}$ of size $m \times n$ into three matrices:

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

Where:

- $\mathbf{U}$: $m \times m$ orthogonal matrix (left singular vectors)
  $\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}_m$
- $\mathbf{\Sigma}$: $m \times n$ diagonal matrix with non-negative singular values σ_i
  $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p)$ where $p = \min(m, n)$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$
- $\mathbf{V}$: $n \times n$ orthogonal matrix (right singular vectors)
  $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}_n$

The values σ_i are called the **singular values** of $\mathbf{A}$. The columns of $\mathbf{U}$ are called the **left singular vectors**, and the columns of $\mathbf{V}$ are called the **right singular vectors**.



2 Practical Examples

2.1 1. Image Compression

- Store only the largest singular values
- Example: Keep top 10% of σ values $\rightarrow$ 90% storage reduction
- Used in: JPEG format, image processing
- Mathematical representation: $\mathbf{A}_{\text{compressed}} = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ where $k \ll \min(m, n)$

2.2 2. Recommendation Systems

- Netflix, Amazon product suggestions
- Finds hidden patterns in user-item ratings
- Example: Users who like action films also like sci-fi
- User-item matrix: $\mathbf{R} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$
- Predicted rating: $\hat{r}_{ij} = \sum_{k=1}^{K} \sigma_k u_{ik} v_{jk}$

2.3 3. Noise Reduction

- Small σ values often represent noise
- Remove small σ's $\rightarrow$ cleaner signal
- Used in: Audio processing, data cleaning
- Filtered matrix: $\mathbf{A}_{\text{clean}} = \sum_{\sigma_i > \tau} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$
- Threshold τ determined by: $\tau = \alpha \cdot \sigma_1$ (typically $\alpha = 0.01$ to 0.1)

2.4 4. Face Recognition (PCA)

- Extract main facial features
- Compare faces using few principal components
- Example: Eigenfaces in security systems
- PCA relation: $\mathbf{X}_{\text{centered}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$
- Principal components = columns of $\mathbf{V}$
- Dimension reduction: Project onto first k columns of $\mathbf{V}$

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

3.1 Step 1: Given matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

3.2 Step 2: Compute $\mathbf{A}^T\mathbf{A}$

$$\mathbf{A}^T = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 5 & 11 \\ 11 & 25 \end{bmatrix}$$

3.3 Step 3: Find eigenvalues of $\mathbf{A}^T\mathbf{A}$

Solve $\det(\mathbf{A}^T\mathbf{A} - \lambda\mathbf{I}) = 0$:

$$\det \begin{bmatrix} 5-\lambda & 11 \\ 11 & 25-\lambda \end{bmatrix} = 0$$

$$(5-\lambda)(25-\lambda) - 121 = 0$$

$$\lambda^2 - 30\lambda + 4 = 0$$

$$\lambda_1 = 15 + \sqrt{221} \approx 29.866, \quad \lambda_2 = 15 - \sqrt{221} \approx 0.134$$

3.4 Step 4: Singular values

$$\sigma_1 = \sqrt{\lambda_1} \approx \sqrt{29.866} \approx 5.465, \quad \sigma_2 = \sqrt{\lambda_2} \approx \sqrt{0.134} \approx 0.366$$

$$\mathbf{\Sigma} = \begin{bmatrix} 5.465 & 0 \\ 0 & 0.366 \end{bmatrix}$$

3.5 Step 5: Find V (eigenvectors of $\mathbf{A}^T\mathbf{A}$)

For $\lambda_1 \approx 29.866$:

$$(\mathbf{A}^T\mathbf{A} - \lambda_1\mathbf{I})\mathbf{v}_1 = 0$$

$$\begin{bmatrix} -24.866 & 11 \\ 11 & -4.866 \end{bmatrix} \mathbf{v}_1 = 0$$

Normalized solution: $\mathbf{v}_1 = \begin{bmatrix} 0.907 \\ 0.421 \end{bmatrix}$

For $\lambda_2 \approx 0.134$:

$$\begin{bmatrix} 4.866 & 11 \\ 11 & 24.866 \end{bmatrix} \mathbf{v}_2 = 0$$

Normalized solution: $\mathbf{v}_2 = \begin{bmatrix} -0.421 \\ 0.907 \end{bmatrix}$

$$\mathbf{V} = \begin{bmatrix} 0.907 & -0.421 \\ 0.421 & 0.907 \end{bmatrix}$$

3.6 Step 6: Find U using $\mathbf{u}_i = \frac{1}{\sigma_i}\mathbf{A}\mathbf{v}_i$

For $i = 1$:

$$\mathbf{u}_1 = \frac{1}{5.465} \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 0.907 \\ 0.421 \end{bmatrix} = \begin{bmatrix} 0.393 \\ 0.640 \end{bmatrix}$$

For $i = 2$:

$$\mathbf{u}_2 = \frac{1}{0.366} \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} -0.421 \\ 0.907 \end{bmatrix} = \begin{bmatrix} -0.580 \\ 0.815 \end{bmatrix}$$

$$\mathbf{U} = \begin{bmatrix} 0.393 & -0.580 \\ 0.640 & 0.815 \end{bmatrix}$$

3.7 Step 7: Final decomposition

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \begin{bmatrix} 0.393 & -0.580 \\ 0.640 & 0.815 \end{bmatrix} \begin{bmatrix} 5.465 & 0 \\ 0 & 0.366 \end{bmatrix} \begin{bmatrix} 0.907 & 0.421 \\ -0.421 & 0.907 \end{bmatrix}$$

3.8 Step 8: Verification

$$\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \approx \begin{bmatrix} 1.000 & 3.000 \\ 2.000 & 4.000 \end{bmatrix} = \mathbf{A} \quad \checkmark$$

4 Interpretation and Applications to Our Example

4.1 1. Data Compression Application

Using only the largest singular value σ_1:

$$\mathbf{A}_{\text{compressed}} = \sigma_1\mathbf{u}_1\mathbf{v}_1^T = 5.465 \begin{bmatrix} 0.393 \\ 0.640 \end{bmatrix} \begin{bmatrix} 0.907 & 0.421 \end{bmatrix}$$

$$\mathbf{A}_{\text{compressed}} = \begin{bmatrix} 1.949 & 0.905 \\ 3.173 & 1.474 \end{bmatrix}$$

Compression ratio: Original: 4 values, Compressed: 1 singular value + 2 vectors = 5 values, but reusable for other matrices.

4.2 2. Noise Reduction Application

Since $\sigma_2 \ll \sigma_1$ ($\sigma_2/\sigma_1 \approx 0.067$), σ_2 could represent noise:

$$\mathbf{A}_{\text{clean}} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T = \begin{bmatrix} 1.949 & 0.905 \\ 3.173 & 1.474 \end{bmatrix}$$

4.3 3. Pattern Recognition

- $\sigma_1 \gg \sigma_2$ indicates one dominant pattern
- First pattern (from $\mathbf{u}_1$ and $\mathbf{v}_1$): Represents the main relationship in the data
- Original columns: $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$
- Reconstructed from first pattern: $\begin{bmatrix} 1.949 \\ 3.173 \end{bmatrix}$ and $\begin{bmatrix} 0.905 \\ 1.474 \end{bmatrix}$
- Shows approximate linear relationship between columns

4.4 4. Matrix Approximation Error

Original matrix norm: $\|\mathbf{A}\|_F = \sqrt{1^2 + 3^2 + 2^2 + 4^2} = \sqrt{30} \approx 5.477$

Singular values contribution:

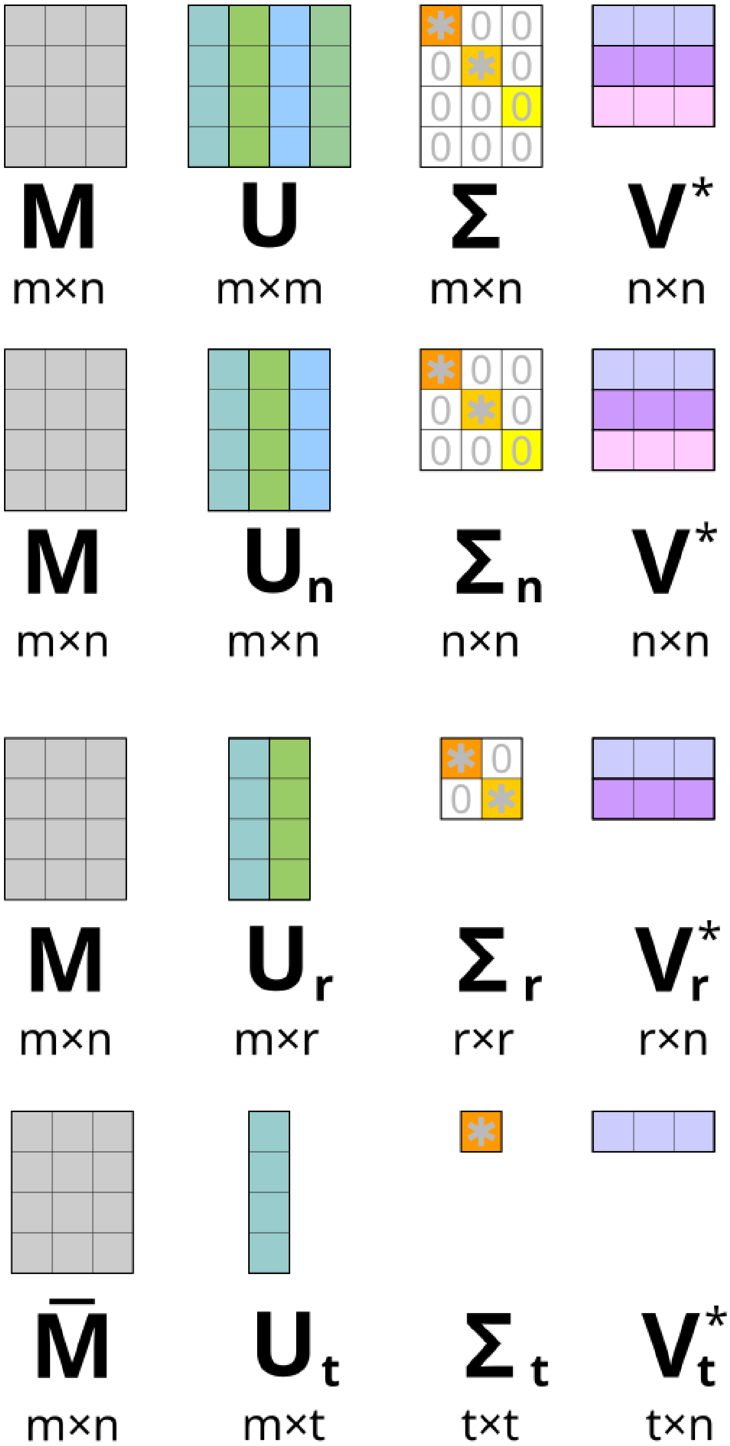
$$\sqrt{\sigma_1^2 + \sigma_2^2} = \sqrt{5.465^2 + 0.366^2} = \sqrt{29.866 + 0.134} = \sqrt{30} = 5.477$$

Using only σ_1 captures:

$$\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} = \frac{29.866}{30} \approx 99.55\% \text{ of the energy}$$

5 Important Properties

1. **Rank**: $\text{rank}(\mathbf{A})$ = number of non-zero singular values
2. **Norms**:
   - Spectral norm: $\|\mathbf{A}\|_2 = \sigma_1$
   - Frobenius norm: $\|\mathbf{A}\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_p^2}$
   - Nuclear norm: $\|\mathbf{A}\|_* = \sigma_1 + \sigma_2 + \cdots + \sigma_p$
3. **Condition number**: $\kappa(\mathbf{A}) = \frac{\sigma_1}{\sigma_p}$ (for $\sigma_p \neq 0$)
4. **Matrix approximation**: Best rank-k approximation: $\mathbf{A}_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^T$
5. **Orthogonality**: $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices


M
m×n
U
m×m
Σ
m×n
V*
n×n
M
m×n
Un
m×n
Σn
n×n
V*
n×n
M
m×n
Ur
m×r
Σr
r×r
V*r
r×n
M̄
m×n
Ut
m×t
Σt
t×t
V*t
t×n