Day 8: Singular Value Decomposition (SVD)

The Swiss Army Knife of Matrix Factorizations

1. What is SVD?

The Singular Value Decomposition is one of the most powerful and ubiquitous tools in linear algebra. It provides a complete **geometric** characterization of any linear transformation represented by a matrix $A \in \mathbb{R}^{m\times n}$.

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

- $\mathbf{U} \in \mathbb{R}^{m\times m}$: An **orthogonal matrix** whose columns $\mathbf{u}_i$ are the **left singular vectors**. They form an orthonormal basis for the output space $\mathbb{R}^m$.
- $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$: A **rectangular diagonal matrix** with non-negative entries $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$ ($p = \min(m, n)$). These are the **singular values**. They represent the "gain" or "stretch" factor along each corresponding direction.
- $\mathbf{V} \in \mathbb{R}^{n\times n}$: An **orthogonal matrix** whose columns $\mathbf{v}_i$ are the **right singular vectors**. They form an orthonormal basis for the input space $\mathbb{R}^n$.

2. Why is SVD So Powerful?

Key Properties

- **Universality:** Exists for *any* matrix, square or rectangular, real or complex.
- **Stability:** Numerically robust to compute, unlike the eigenvalue decomposition for non-normal matrices.
- **Reveals Rank:** The number of non-zero singular values σ_i is the **rank** r of the matrix A.
- **Optimal Low-Rank Approximation (Eckart–Young–Mirsky Theorem):** The best rank-k approximation A_k of A (in both the spectral and Frobenius norms) is given by truncating the SVD:

  $$A_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad \text{(where } k < r\text{)}$$

  This is the foundation of data compression and dimensionality reduction.

3. Geometric Intuition

The SVD provides a clear geometric interpretation of the linear transformation $\mathbf{x} \mapsto A\mathbf{x}$:

1. **Rotation/Reflection in $\mathbb{R}^n$:** Multiply by V^T aligns the standard basis with the right singular vectors ($\mathbf{v}_i$).
2. **Scaling:** Multiply by Σ stretches the space along the coordinate axes by the factors σ_i.
3. **Rotation/Reflection in $\mathbb{R}^m$:** Multiply by U maps the scaled result to the correct orientation defined by the left singular vectors ($\mathbf{u}_i$).

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

4. Worked Example: A Simple 2×2 Matrix

Let's decompose the matrix:

$$A = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$$

This matrix simply scales the x-axis by 3 and the y-axis by 2. Its SVD is trivial but instructive.

$$\begin{aligned} A &= U \Sigma V^T \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T \\ &= I \cdot \Sigma \cdot I \end{aligned}$$

- **Singular Values:** $\sigma_1 = 3$, $\sigma_2 = 2$.
- **Interpretation:** The input directions $(\mathbf{v}_1, \mathbf{v}_2)$ are already aligned with the standard axes, and so are the output directions $(\mathbf{u}_1, \mathbf{u}_2)$. The transformation is a pure scale with no rotation.

Now consider a non-trivial matrix:

$$B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Its SVD (computed numerically) is approximately:

$$\begin{aligned} B &= U \Sigma V^T \\ &\approx \begin{bmatrix} -0.85 & -0.53 \\ -0.53 & 0.85 \end{bmatrix} \begin{bmatrix} 1.62 & 0 \\ 0 & 0.62 \end{bmatrix} \begin{bmatrix} -0.53 & -0.85 \\ -0.85 & 0.53 \end{bmatrix}^T \end{aligned}$$

This shows the transformation: 1) Rotate by $\sim 122^\circ$ (V^T), 2) Scale axes by ~ 1.62 and ~ 0.62 (Σ), 3) Rotate by $\sim 148^\circ$ (U).

5. Core Applications in AI/ML

SVD in Practice

- **Principal Component Analysis (PCA):** PCA is performed by taking the SVD of the centered data matrix (or the covariance matrix). The right singular vectors $\mathbf{v}_i$ are the **principal components**, and σ_i^2 is proportional to the variance explained by each component.
- **Latent Semantic Analysis (LSA) / Indexing:** In NLP, a term-document matrix A is decomposed via SVD. U relates to terms, V to documents, and Σ to the strength of "latent concepts". Truncating the SVD ($A \approx A_k$) smoothes the data and reveals hidden semantic relationships.
- **Recommender Systems:** The famous **Netflix Prize** was won using matrix factorization techniques (like SVD++) that are direct descendants of SVD. It factors a user-item rating matrix into lower-dimensional user and item latent factor matrices.
- **Data Compression & Dimensionality Reduction:** Images can be represented as matrices. A rank-k SVD approximation stores only $k(m+n+1)$ numbers instead of mn, achieving compression (e.g., for a 1000×1000 image, a rank-50 approximation uses only $50(1000+1000+1) = 100,050$ values instead of $1,000,000$).
- **Numerical Linear Algebra:** SVD is used to solve linear systems, compute the pseudo-inverse ($A^\dagger = V \Sigma^\dagger U^T$), and find low-rank approximations of Jacobians in deep learning.