Machine Learning 101

Rajdeep Chatterjee, Ph.D.
Amygdala AI, Bhubaneswar, India *

February 2025

Singular Value Decomposition (SVD)

1 Introduction

Singular Value Decomposition (SVD) is a fundamental matrix factorization technique in linear algebra. It decomposes a matrix into three simpler matrices, revealing the intrinsic geometric and algebraic properties of the original matrix. SVD is widely used in data compression, noise reduction, and machine learning (e.g., Principal Component Analysis).

2 Variance and Covariance Matrix

2.1 Definition of Variance

Variance is a measure of how much the values in a dataset deviate from the mean. It is mathematically defined as:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{1}$$

where:

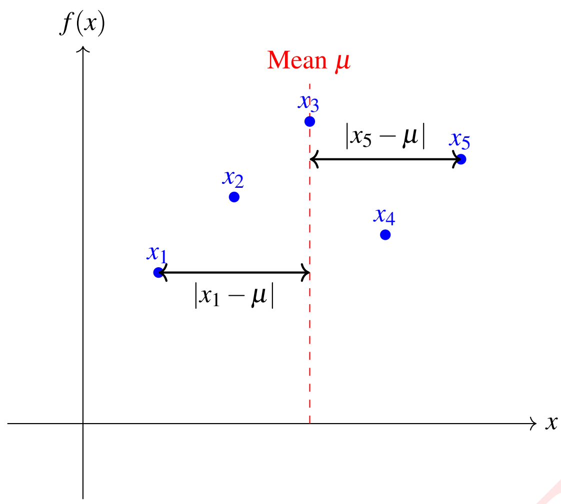
- σ^2 is the variance,
- N is the number of data points,
- x_i are the individual data points,
- μ is the mean of the data, given by $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$.

2.2 Visualization

Below is a simple visualization of variance using TikZ:

*Amygdala AI, is an international volunteer-run research group that advocates for *AI for a better tomorrow* `http://amygdalaai.org/`.

CSERAJDEEP



The variance is higher when points are spread further from the mean, and lower when they are closer together.

2.3 Covariance Matrix

In machine learning and data science, the covariance matrix is a fundamental concept used to measure the relationships between multiple variables in a dataset. It helps in understanding the variance of each feature and how different features vary together.

The covariance matrix Σ of a dataset with n features is an $n \times n$ matrix where each element σ_{ij} represents the covariance between the i^{th} and j^{th} feature:

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix} \tag{2}$$

where:

$$\sigma_{ij} = \text{Cov}(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)] \tag{3}$$

- σ_{ii} (diagonal elements) represent the variance of each feature.
- σ_{ij} (off-diagonal elements) represent the covariance between features X_i and X_j.
- If $\sigma_{ij} > 0$, the features are positively correlated.
- If $\sigma_{ij} < 0$, the features are negatively correlated.
- If $\sigma_{ij} = 0$, the features are uncorrelated (If X and Y are independent, then $\text{Cov}(X,Y) = 0$.).

For a dataset with n random variables, the covariance matrix Σ is given by:

$$\Sigma = \begin{bmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) & \dots & \text{Cov}(X_1, X_n) \\ \text{Cov}(X_2, X_1) & \text{Var}(X_2) & \dots & \text{Cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(X_n, X_1) & \text{Cov}(X_n, X_2) & \dots & \text{Var}(X_n) \end{bmatrix}$$

Copyright ©cserajdeep, 2025

CSERAJDEEP

2.4 Example

Consider a dataset with two variables X and Y, where:

$$X = [1,2,3], \quad Y = [2,4,6]$$

1. Compute the means:
$$\mu_X = \frac{1+2+3}{3} = 2, \quad \mu_Y = \frac{2+4+6}{3} = 4$$
2. Compute variances:
$$\mathrm{Var}(X) = \frac{(1-2)^2+(2-2)^2+(3-2)^2}{3} = \frac{2}{3}$$
$$\mathrm{Var}(Y) = \frac{(2-4)^2+(4-4)^2+(6-4)^2}{3} = \frac{8}{3}$$
3. Compute covariance:
$$\mathrm{Cov}(X,Y) = \frac{(1-2)(2-4)+(2-2)(4-4)+(3-2)(6-4)}{3} = \frac{4}{3}$$

Thus, the covariance matrix is:

$$\begin{bmatrix} \frac{2}{3} & \frac{4}{3} \\ \frac{4}{3} & \frac{8}{3} \end{bmatrix} \tag{4}$$

Usage in Machine Learning

- **Principal Component Analysis (PCA)**: Used to transform correlated features into uncorrelated principal components.
- **Multivariate Gaussian Distribution**: Helps in probabilistic modeling.
- **Feature Selection & Engineering**: Identifies redundant features.
- **Portfolio Optimization**: Used in finance for risk assessment.

Step-by-Step Derivation with Definitions and Numerical Examples

3 Mathematics Behind SVD

Given a matrix $A \in R^{m\times n}$, SVD factorizes it into three matrices:

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

Where:

- $U \in R^{m\times m}$: Orthogonal matrix (left singular vectors).
- $\Sigma \in R^{m\times n}$: Diagonal matrix of singular values.
- $V \in R^{n\times n}$: Orthogonal matrix (right singular vectors).

CSERAJDEEP

Copyright ©cserajdeep, 2025

3.1 Step-by-Step Derivation

1. **Compute $A^T A$ and AA^T:**
   - $A^T A$ is an $n \times n$ symmetric matrix.
   - AA^T is an $m \times m$ symmetric matrix.
2. **Eigenvalue Decomposition:**
   - Compute the eigenvalues λ_i and eigenvectors v_i of $A^T A$.
   - The singular values σ_i are the square roots of the eigenvalues: $\sigma_i = \sqrt{\lambda_i}$.
3. **Construct V:**
   - The columns of V are the eigenvectors v_i of $A^T A$.
4. **Construct Σ:**
   - Σ is a diagonal matrix with singular values σ_i in descending order.
5. **Construct U:**
   - The columns of U are the eigenvectors of AA^T.
   - Alternatively, compute U as $U = AV\Sigma^{-1}$.

3.2 Example

Consider the matrix:

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

1. Compute $A^T A$ and AA^T:

$$A^T A = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 13 & 12 \\ 12 & 13 \end{bmatrix}$$

$$AA^T = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 13 & 12 \\ 12 & 13 \end{bmatrix}$$

2. Compute eigenvalues and eigenvectors of $A^T A$: - Eigenvalues: $\lambda_1 = 25, \lambda_2 = 1$ - Singular values: $\sigma_1 = 5, \sigma_2 = 1$

3. Construct V: - Eigenvectors of $A^T A$:

$$V = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

4. Construct Σ:

$$\Sigma = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}$$

5. Construct U:

$$U = AV\Sigma^{-1} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

Thus, the SVD of A is:

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

Copyright ©cserajdeep, 2025

CSERAJDEEP

Compute SVD with example (complete solution)

1. Compute $A^T A$ and AA^T:

- **Definition:**
  - $A^T A$ is an $n \times n$ symmetric matrix.
  - AA^T is an $m \times m$ symmetric matrix.
- **Example:** Let $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$.
  - $A^T A = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} 35 & 44 \\ 44 & 56 \end{bmatrix}$.
  - $AA^T = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} = \begin{bmatrix} 5 & 11 & 17 \\ 11 & 25 & 39 \\ 17 & 39 & 61 \end{bmatrix}$.

2. Eigenvalue Decomposition:

- **Definition:**
  - Compute the eigenvalues λ_i and eigenvectors v_i of $A^T A$.
  - The singular values σ_i are the square roots of the eigenvalues: $\sigma_i = \sqrt{\lambda_i}$.
- **Example:** For $A^T A = \begin{bmatrix} 35 & 44 \\ 44 & 56 \end{bmatrix}$:
  - Eigenvalues: $\lambda_1 \approx 90.67$, $\lambda_2 \approx 0.33$.
  - Singular values: $\sigma_1 = \sqrt{90.67} \approx 9.52$, $\sigma_2 = \sqrt{0.33} \approx 0.57$.

3. Construct V:

- **Definition:**
  - The columns of V are the eigenvectors v_i of $A^T A$.
- **Example:** For $A^T A$, the eigenvectors are:
  - $v_1 \approx \begin{bmatrix} 0.58 \\ 0.81 \end{bmatrix}$, $v_2 \approx \begin{bmatrix} -0.81 \\ 0.58 \end{bmatrix}$.
  - Thus, $V = \begin{bmatrix} 0.58 & -0.81 \\ 0.81 & 0.58 \end{bmatrix}$.

CSERAJDEEP

Copyright ©cserajdeep, 2025

4. Construct Σ:

- **Definition:**
  - Σ is a diagonal matrix with singular values σ_i in descending order.
- **Example:** For the singular values $\sigma_1 \approx 9.52$ and $\sigma_2 \approx 0.57$:
  - $\Sigma = \begin{bmatrix} 9.52 & 0 \\ 0 & 0.57 \end{bmatrix}$.

5. Construct U:

- **Definition:**
  - The columns of U are the eigenvectors of AA^T.
  - Alternatively, compute U as $U = AV\Sigma^{-1}$.
- **Example:** For AA^T, the eigenvectors are:
  - $u_1 \approx \begin{bmatrix} 0.23 \\ 0.53 \\ 0.82 \end{bmatrix}$, $u_2 \approx \begin{bmatrix} -0.83 \\ -0.24 \\ 0.50 \end{bmatrix}$, $u_3 \approx \begin{bmatrix} 0.50 \\ -0.81 \\ 0.31 \end{bmatrix}$.
  - Thus, $U = \begin{bmatrix} 0.23 & -0.83 & 0.50 \\ 0.53 & -0.24 & -0.81 \\ 0.82 & 0.50 & 0.31 \end{bmatrix}$.

Final SVD Decomposition:

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

For the given example:

$$A = \begin{bmatrix} 0.23 & -0.83 & 0.50 \\ 0.53 & -0.24 & -0.81 \\ 0.82 & 0.50 & 0.31 \end{bmatrix} \begin{bmatrix} 9.52 & 0 \\ 0 & 0.57 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0.58 & 0.81 \\ -0.81 & 0.58 \end{bmatrix}$$

4 SVD as Pseudo-Code

Algorithm 1 Singular Value Decomposition (SVD)

Require: Matrix $A \in R^{m \times n}$
Ensure: Matrices U, Σ, V^T

Compute $A^T A$ and AA^T
Perform eigenvalue decomposition of $A^T A$ to get eigenvalues λ_i and eigenvectors v_i
Compute singular values $\sigma_i = \sqrt{\lambda_i}$
Sort singular values and corresponding eigenvectors in descending order
Construct Σ as a diagonal matrix with singular values σ_i
Compute $U = AV\Sigma^{-1}$
return U, Σ, V^T

CSERAJDEEP

Copyright ©cserajdeep, 2025

5 SVD on a Toy Dataset

Consider a toy dataset represented as a matrix:

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

Step 1: Compute $A^T A$ and AA^T

The Gram matrices are computed as follows:

$$A^T A = \begin{bmatrix} 35 & 44 \\ 44 & 56 \end{bmatrix}, \quad AA^T = \begin{bmatrix} 5 & 11 & 17 \\ 11 & 25 & 39 \\ 17 & 39 & 61 \end{bmatrix} \tag{6}$$

Step 2: Eigenvalue Decomposition

The eigenvalues of $A^T A$ are:

$$\lambda_1 = 90.67, \quad \lambda_2 = 0.33 \tag{7}$$

The singular values, given by $\sigma_i = \sqrt{\lambda_i}$, are:

$$\sigma_1 \approx 9.52, \quad \sigma_2 \approx 0.57 \tag{8}$$

Step 3: Construct V

The eigenvectors of $A^T A$ the form the matrix:

$$V = \begin{bmatrix} -0.62 & -0.78 \\ -0.78 & 0.62 \end{bmatrix} \tag{9}$$

5.1 Step 4: Construct Σ

The diagonal matrix of singular values is:

$$\Sigma = \begin{bmatrix} 9.52 & 0 \\ 0 & 0.57 \\ 0 & 0 \end{bmatrix} \tag{10}$$

Step 5: Construct U

The left singular vectors (eigenvectors of AA^T) form the matrix:

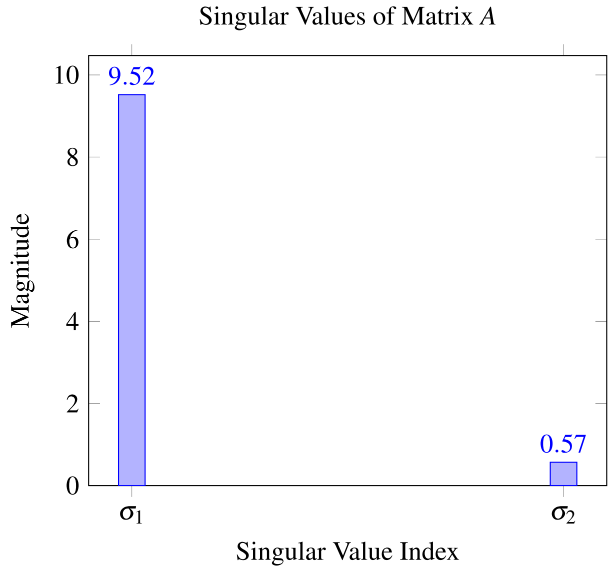
$$U = \begin{bmatrix} -0.23 & -0.88 \\ -0.52 & -0.24 \\ -0.82 & 0.40 \end{bmatrix} \tag{11}$$

CSERAJDEEP

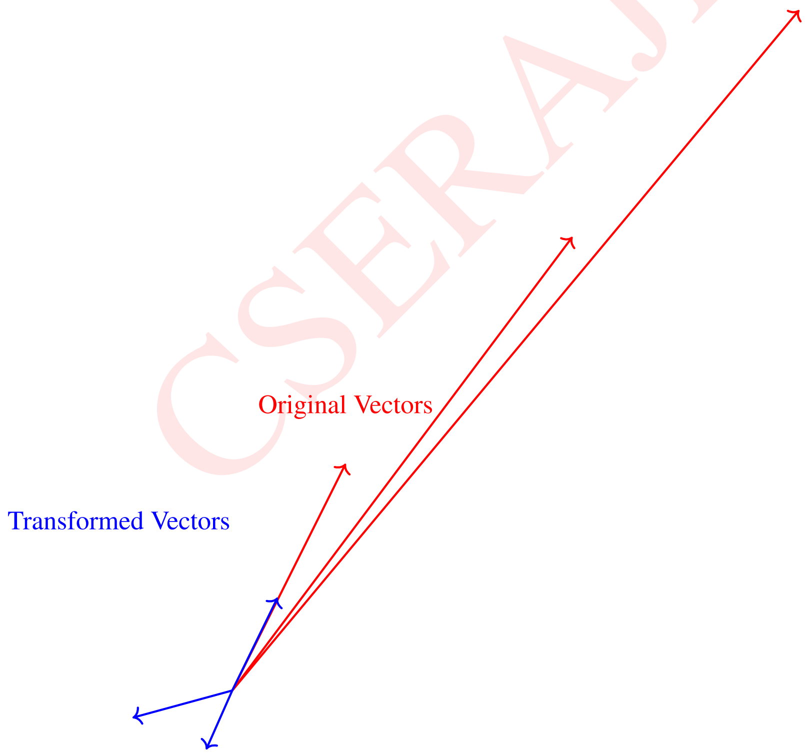
Copyright ©cserajdeep, 2025

6 Visualization Using TikZ

6.1 Singular Values Plot



6.2 Transformation Plot



CSERAJDEEP

Copyright ©cserajdeep, 2025

7 Merits of SVD

- **Robustness:** SVD is numerically stable and provides a well-conditioned factorization of a matrix.
- **Dimensionality Reduction:** It is widely used in Principal Component Analysis (PCA) and Latent Semantic Analysis (LSA) for feature extraction.
- **Noise Reduction:** SVD helps in denoising data by retaining only the significant singular values.
- **Solving Ill-posed Problems:** It is useful for solving systems of linear equations, especially in least-squares solutions.
- **Low-rank Approximation:** SVD can approximate large matrices with fewer components, making it useful for compression.

8 Demerits of SVD

- **Computational Complexity:** The computation of SVD is expensive, with a time complexity of $O(n^3)$ for an $n \times n$ matrix.
- **Storage Requirements:** It requires significant memory for storing the decomposed matrices.
- **Interpretability:** The singular values and singular vectors may not always have a direct physical interpretation.
- **Scalability Issues:** For very large datasets, computing SVD becomes impractical without optimization techniques.

9 Remedies for the Demerits

- **Randomized SVD:** Reduces computational cost by approximating SVD using random projections.
- **Truncated SVD:** Computes only the top k singular values and vectors, reducing memory and computational requirements.
- **Parallel and GPU Computing:** Using distributed or GPU-based implementations speeds up SVD computations.
- **Sparse Matrix Techniques:** Exploiting sparsity in matrices can significantly improve efficiency.
- **Alternative Factorizations:** Methods like QR decomposition or non-negative matrix factorization (NMF) can sometimes be preferable depending on the application.

10 Applications of SVD

Dimensionality Reduction

SVD is widely used in **Principal Component Analysis (PCA)** for reducing the dimensionality of datasets while preserving most of the variance.

CSERAJDEEP

Copyright ©cserajdeep, 2025

Latent Semantic Analysis (LSA)

SVD is applied in **Natural Language Processing (NLP)** to extract hidden relationships between words and documents by decomposing the term-document matrix.

Image Compression

SVD helps in image compression by retaining only the top singular values, significantly reducing the storage space while maintaining image quality.

Noise Reduction

In signal and image processing, SVD is used to filter out noise by removing small singular values, improving data clarity.

Recommendation Systems

SVD is a key component of collaborative filtering techniques, such as in Netflix's recommendation algorithm, to predict user preferences based on past interactions.

Solving Linear Systems

SVD is used to compute the pseudo-inverse of matrices, which is helpful in solving ill-posed linear systems and least-squares problems.

Face Recognition

By applying SVD to face datasets, features can be extracted and used for classification in facial recognition systems.

11 Conclusion

SVD is a versatile tool with numerous applications in scientific computing, data analysis, and machine learning. Its ability to decompose complex data structures into simpler components makes it invaluable in various domains.
To observe an improvement in classification accuracy after applying SVD, it is beneficial to experiment with datasets that have **a large number of features relative to the number of samples** or datasets that are prone to noise and redundancy. Applying SVD in such contexts can lead to more efficient feature representations and better classification outcomes.

CSERAJDEEP

Copyright ©cserajdeep, 2025