Contents

1 Machine Learning Foundations

1.1 SVD

1.1.1 Explanation

The singular value decomposition (SVD) is a matrix decomposition method which decomposes a matrix A into three specific matrices. The matrix A could contain complex numbers, but I will focus on the real numbers because the decomposition is used for machine learning.

Let $A \in \mathbb{R}^{m \times n}$, then there exists a decomposition $A = USV^T$ containing the following matrices:

- $U \in \mathbb{R}^{m \times m}$: an orthogonal matrix ($U^TU = UU^T = I$) with column vectors $\vec{u_1}, ..., \vec{u_m}$.
- $V \in \mathbb{R}^{n \times n}$: an orthogonal matrix with column vectors $\vec{v_1}, ..., \vec{v_n}$.
- $S \in \mathbb{R}^{m \times n}$: a rectangular matrix with only non-zero elements on the main diagonal, $S_{ii} \geq 0$ and $S_{ij} = 0, i \neq j$.

$$A = \begin{pmatrix} | & | & | & | \\ \vec{u_1} & \vec{u_2} & ... & \vec{u_m} \\ | & | & | & | \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} | & | & | & | \\ \vec{v_1} & \vec{v_2} & ... & \vec{v_n} \\ | & | & | & | \end{pmatrix}^T$$

The diagonal elements of S are called the **singular values**, noted as $S_{ii} = \sigma_i$. By convention, they are ordered as $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0$, with r the rank. The columns of U are called the **left singular vectors**, and the columns of V are called the **right singular vectors**. If $m < n$, the matrix S would contain 'padding' of zeros on the right side, instead of underneath. It is needed because the matrix is the same size as the original A. SVD is a very generalized decomposition, so it **exists for any** $A \in \mathbb{R}^{m \times n}$.

1.1.2 Construction of SVD

The SVD is a generalized version of the diagonalization of square matrices. A square matrix $A_{n\times n}$ is called diagonalizable if there exists an invertible matrix $P_{n\times n}$ for which $A = PDP^{-1}$, with D a diagonal matrix containing the eigenvalues of A, and P containing the basis of eigenvectors as columns. Calculating the SVD of this $A_{n\times n}$ matrix will result in exactly the same decomposition as the diagonalization.

Computing the SVD of the rectangular $A_{m\times n}$ matrix is equivalent to finding two orthonormal bases $U = (\vec{u_1}, ..., \vec{u_m})$ and $V = (\vec{v_1}, ..., \vec{v_n})$. Start with constructing the right singular vectors $\vec{v_1}, ..., \vec{v_n}$. Create the matrix $B := A^T A \in \mathbb{R}^{n\times n}$. B is symmetric because $B^T = (A^T A)^T = A^T (A^T)^T = A^T A = B$, and positive semi-definite, which means that for all $x : x^T Bx \geq 0$. We obtain $x^T Bx = x^T A^T Ax = (Ax)^T (Ax) = \langle Ax, Ax\rangle$, which is positive because the dot product computes the sum of squares. Therefore, it is possible to diagonalize B as $B = V DV^T$, with D a diagonal matrix containing the eigenvalues $\theta_1, ..., \theta_n$. Assume that the SVD exists such that:

$$B = A^T A = (USV^T)^T (USV^T) = V S^T U^T USV^T = V S^T SV^T = V S^2 V^T$$

Remember that U is orthogonal so $U^T U = I$ and S is a diagonal matrix so $S^T = S$. We can now see that $B = V DV^T = V S^2 V^T$ thus $\theta_i = \sigma_i^2$. This concludes that V contains the right singular vectors.

To compute the left singular vectors, follow the same principle as above but with the matrix $C := AA^T \in \mathbb{R}^{n\times n}$:

$$C = AA^T = (USV^T)(USV^T)^T = USV^T V S^T U^T = USS^T U^T = US^2 U^T$$

Because C can be diagonalized, we find $AA^T = UDU^T$ with U an orthonormal basis of eigenvectors, representing left singular vectors.

The matrix S can be constructed because we can take the square root of the eigenvalues from the previous diagonalization to find the singular values. The non-zero eigenvalues of AA^T and $A^T A$ are the same, so the entries in S will also be the same.

To connect all the pieces, we have an orthonormal set of right singular vectors in the matrix V, the singular values in the matrix S, and an orthonormal set of left singular vectors in U. There is still a problem because these eigenvectors are not unique. If we first calculate V, normalisation is needed for U based on the vectors in V. This happens with the following calculations:

$$\vec{u_i} := \frac{A\vec{v_i}}{||A\vec{v_i}||} = \frac{1}{\sqrt{\theta_i}} A\vec{v_i} = \frac{1}{\sigma_i} A\vec{v_i}$$

This can be rearranged to find the formula:

$$A\vec{v_i} = \sigma_i u_i \quad \text{or} \quad AV = US$$

What results in the final SVD equation:

$$A = USV^T$$

1.1.3 Implementation details

The `transform` function takes a matrix A as input. This could be a matrix with any possible shape $m \times n$. The implementation first checks the shape of A for computational efficiency:

- When $m \geq n$, calculate $B := A^T A$.
- When $m < n$, calculate $B := AA^T$.

This way, we always work with the smallest matrix product to save computation time. To calculate the first singular vectors, we calculate the eigenvalue decomposition with the `eigh` function because it is specialized for symmetric/Hermitian matrices (which B always is). This is more numerically stable than the general `eig` solver. In theory, these eigenvalues only contain values ≥ 0, but because of floating point arithmetic, small negative values could appear. These are rounded to zero if they occur. A similar problem arises with the singular values because some of them might be very small, resulting in near-zero division. A threshold is set to filter out that numerical noise. The singular values are calculated simply by taking the square root of the eigenvalues of B.

To compute the other singular vectors (depending on which case we are in), we use the formula $U = AV\frac{1}{\sigma}$ or $V = A^T U\frac{1}{\sigma}$, with $\frac{1}{\sigma}$ a diagonal matrix containing the normalized singular values. The function returns the three matrices U, S and V^T as final result.