



Singular Value Decomposition

JING HU

ALI RASIM KOCAL

Gliederung

1. Grundbegriffe
2. SVD Definition
3. Dimensionsreduktion
4. Rechenbeispiel mit Numpy
5. Anwendungsmöglichkeiten
6. Literaturverzeichnis

Grundbegriffe

Diagonalmatrix $\forall i \neq j: D_{i,j} = 0$

$$D = \begin{pmatrix} d_{11} & 0 & \cdots & 0 \\ 0 & d_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_{nn} \end{pmatrix}$$

Grundbegriffe

Diagonalmatrix $\forall i \neq j: D_{i,j} = 0$

$$D = \begin{pmatrix} d_{11} & 0 & \dots & 0 \\ 0 & d_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & d_{nn} \end{pmatrix}$$

Transponierte Matrix $\forall i, j: A_{i,j} = A_{j,i}^T$

A

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

Grundbegriffe

Inverse Matrix $A \cdot A^{-1} = A^{-1} \cdot A = E$

$$A \quad \cdot \quad A^{-1} \quad = \quad E$$

$$\begin{pmatrix} 1 & 2 & -1 \\ 0 & 1 & -1 \\ 2 & 2 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

https://www.youtube.com/watch?v=LmiF_iCV-Fk

Grundbegriffe

Orthogonalmatrix

$$Q^T Q = \begin{bmatrix} q_1^T \\ q_2^T \\ q_3^T \\ \vdots \\ q_n^T \end{bmatrix} [q_1 \ q_2 \ q_3 \ \dots \ q_n] = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} = I$$

Beispiel:

$$Q^T Q = \begin{pmatrix} 1/3 & 2/3 & -2/3 \\ -2/3 & 2/3 & 1/3 \\ 2/3 & 1/3 & 2/3 \end{pmatrix} \begin{pmatrix} 1/3 & -2/3 & 2/3 \\ 2/3 & 2/3 & 1/3 \\ -2/3 & 1/3 & 2/3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

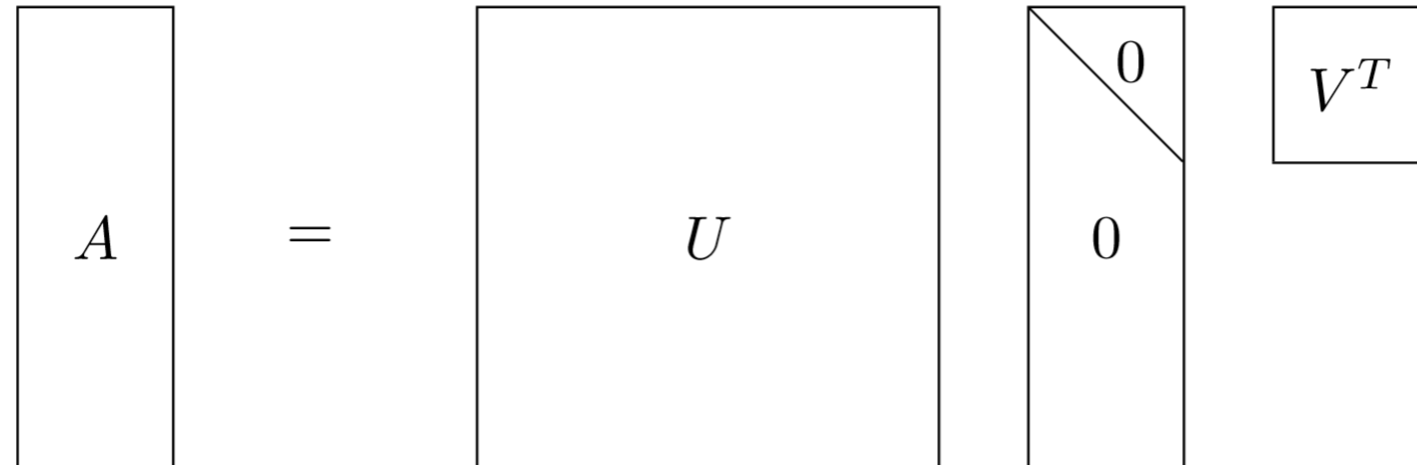
SVD Definition

Jede Matrix $A \in \mathbb{R}^{m \times n}$, mit $m \geq n$, kann zerlegt werden als:

$$A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T$$

Wobei $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ orthogonal, und $\Sigma \in \mathbb{R}^{n \times n} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ so dass $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$

SVD Definition



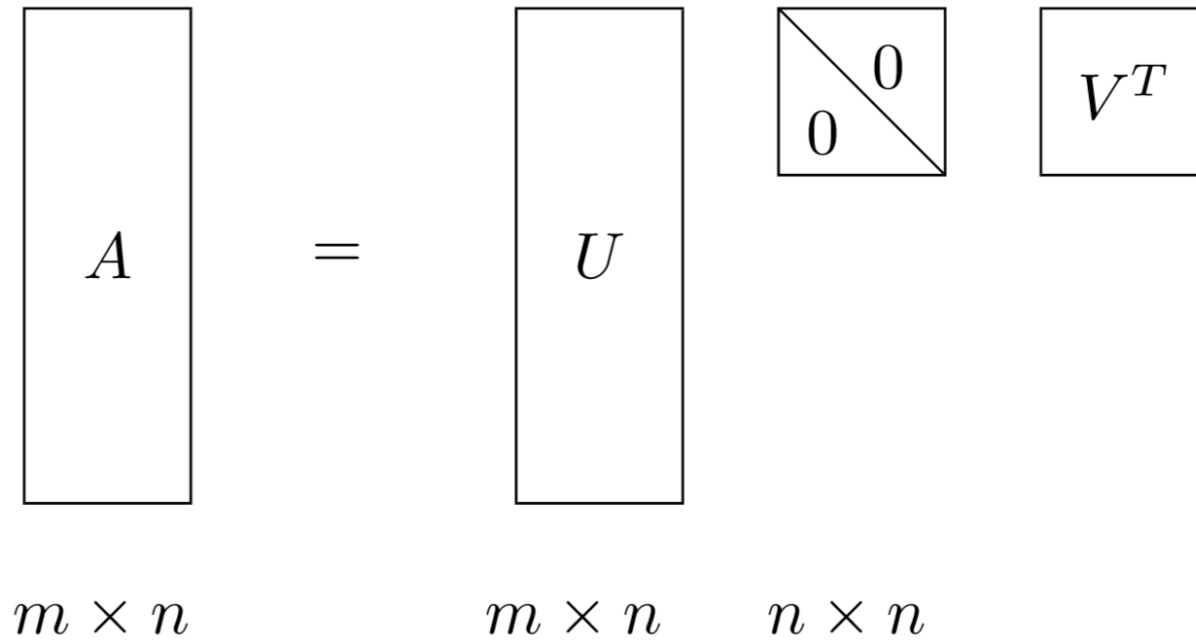
The diagram illustrates the SVD decomposition of a matrix A . It shows a vertical rectangle labeled A on the left, followed by an equals sign. To the right of the equals sign are three components: a large square labeled U , a vertical rectangle with a diagonal line from the top-right to the bottom-left, with a '0' in the top-right triangle and another '0' in the bottom-left triangle, and a small square labeled V^T .

$$\begin{matrix} \boxed{A} & = & \boxed{U} & \begin{matrix} \diagdown \\ 0 \\ \diagup \\ 0 \end{matrix} & \boxed{V^T} \end{matrix}$$

$m \times n$ $m \times m$ $m \times n$

$$U = (U_1 \ U_2)$$

Dimensionsreduktion



The diagram illustrates the SVD decomposition of a matrix A into three matrices: U , Σ , and V^T . Matrix A is represented by a tall rectangle with dimensions $m \times n$ below it. An equals sign follows. Matrix U is another tall rectangle with dimensions $m \times n$ below it. This is followed by matrix Σ , a square with a diagonal line from the top-left to the bottom-right, with '0' in the top-right and bottom-left corners, and dimensions $n \times n$ below it. Finally, matrix V^T is a square with dimensions $n \times n$ below it.

$$\begin{matrix} \boxed{A} & = & \boxed{U} & \boxed{\Sigma} & \boxed{V^T} \\ m \times n & & m \times n & n \times n & \end{matrix}$$

$$A = U_1 \Sigma V^T$$

Rechenbeispiel mit Numpy

<https://github.com/arkocal/SVD-Notebook>

Anwendungen

- Wortähnlichkeit berechnen
- Sentimentanalyse
- Latent Semantic Analysis, LSA
- Recommender system
- How does Netflix recommend movies? Matrix Factorization

<https://www.youtube.com/watch?v=ZspR5PZemcs>

Literaturverzeichnis

- Eldén, Lars. Matrix Methods in Data Mining and Pattern Recognition. Linköping University, Linköping, Sweden. 2007.
- Roth, Benjamin. Schütze Hinrich. Embeddings Learned By Matrix Factorization. Center for Information and Language Processing, LMU Munich. 2018
- How does Netflix recommend movies? Matrix Factorization
<https://www.youtube.com/watch?v=ZspR5PZemcs>