Singular Value Decomposition

Practical Linear Algebra | Lecture 14

Singular Value Decomposition

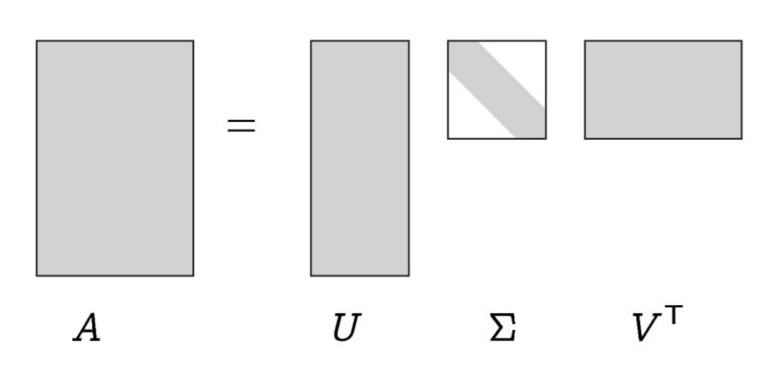
- Any arbitrary matrix $A \in \mathbb{R}^{m \times n}$ can be decomposed as

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

where

- U has orthonormal columns called the left singular vectors of A
- V has orthonormal columns called the right singular vectors of A
- Σ is a diagonal matrix with positive real numbers σ_i, called the singular values of A, along the main diagonal

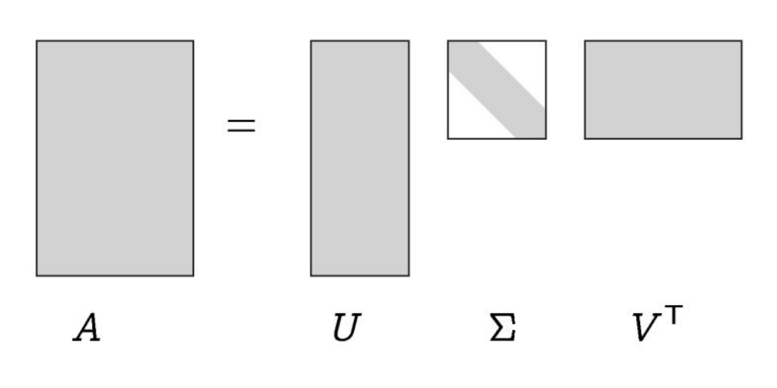
Singular Value Decomposition



- U and V are not necessarily orthogonal matrices
- $U^T U = I$ and $V^T V = I$, but in general $UU^T \neq I$ and $VV^T \neq I$
- Can arrange singular values in descending order without loss of generality

From https://ee263.stanford.edu/lectures/svd.pdf

Singular Value Decomposition



- Matrix sizes: $A \in \mathbb{R}^{m \times n},\ U \in \mathbb{R}^{m \times r},\ \Sigma \in \mathbb{R}^{r \times r},\ V \in \mathbb{R}^{n \times r}$
- This implies $r \leq \mathbf{min}(m, n)$

From https://ee263.stanford.edu/lectures/svd.pdf

SVD in Numpy

```python
import numpy as np

# generate a random matrix
A = np.random.randint(low=-99, high=99, size=(4, 3))
print('A:\n', A, '\n')

# compute singular value decomposition
U, S, V = np.linalg.svd(A)

# S is list of singular values, so use np.diag() to create matrix
SMat = np.diag(S)

# round for ease of reading
print('U:\n', np.round(U, 1), '\n')
print('S:\n', np.round(SMat, 1), '\n')
print('V^T:\n', np.round(V.transpose(), 1), '\n')
```

```
A:
 [[ 20  59  63]
 [ 95 -15 -79]
 [ 20 -40  26]
 [ 19 -66  25]]
```

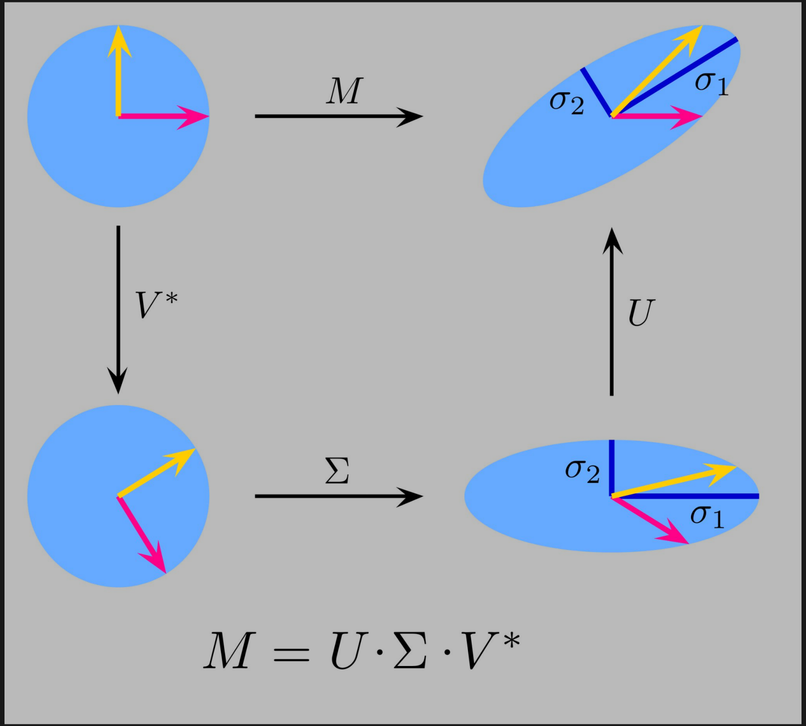
=

```
U:
 [[ 0.4  0.4 -0.8  0.1]
 [-0.9  0.3 -0.3  0. ]
 [-0.1 -0.5 -0.4 -0.8]
 [-0.1 -0.7 -0.3  0.6]]
```

```
S:
 [[132.1   0.    0. ]
 [  0.   91.6   0. ]
 [  0.    0.   74.4]]
```

```
V^T:
 [[-0.6  0.2 -0.8]
 [ 0.4  0.9 -0.1]
 [ 0.7 -0.3 -0.6]]
```

Geometric Interpretation



1. *Rotate or Reflect*
2. *Stretch or Compress*
3. *Rotate or Reflect*

From https://commons.wikimedia.org/wiki/File:Singular-Value-Decomposition.svg

Singular Values

- What exactly *are* singular values?
- They're the generalization of eigenvalues
- They're actually square roots of the eigenvalues of $A^T A$
  - $\sigma_i = \sqrt{\lambda_i(A^T A)}$
- They're also square roots of the eigenvalues of AA^T
  - $\sigma_i = \sqrt{\lambda_i(AA^T)}$
- $A^T A$ and AA^T have the same eigenvalues
- The matrix norm of A is just the greatest singular value
  - $\|A\| = \sigma_1$

Condition Number

- The condition number of a matrix A is

$$\kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

  where $\sigma_{\max}(A)$ is the greatest singular value and $\sigma_{\min}(A)$ is the least
- Measures the sensitivity of a matrix and is useful in error analysis
- Matrix is well-conditioned if κ is small, ill-conditioned if κ is large

Dyads

$$A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

- Any matrix is just the sum of dyads (rank-one matrices)
- The number of nonzero singular values (r) is just $\mathbf{rank}(A)$
- The SVD generalizes the spectral decomposition to arbitrary matrices

Dyads

$$A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

- What if we only added some of the dyads together...?

Low-Rank Approximations

- Consider the problem of finding the best rank k approximation of a matrix
- "Best" in the sense of minimizing the matrix norm of the difference matrix
- In symbols: find the rank k matrix A_k that minimizes $||A - A_k||$
- Solution:

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

- This is known as the Eckart–Young–Mirsky theorem
- Turns out this minimizes both the matrix norm *and* the Frobenius norm!

Low-Rank Approximations

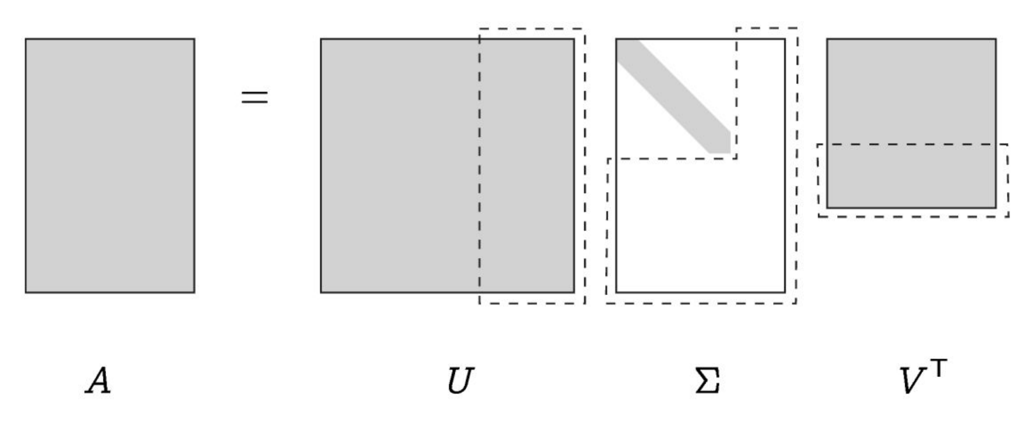
- The error of the optimal low-rank approximation has a simple closed form:

$$\|A - A_k\| = \left\| \sum_{i=k+1}^{r} \sigma_i u_i v_i^T \right\| = \sigma_{k+1}$$

- The error is exactly the largest singular value of the matrix we *left out of* A

$$A = A_k + A_{\text{leftover}} = \sum_{i=1}^{k} i^{\text{th}}\text{dyad} + \sum_{i=k+1}^{r} i^{\text{th}}\text{dyad}$$

Full SVD



- If we extend U and V to be orthogonal (hence square) matrices, and append zeros to Σ, we get the full SVD
- The SVD on previous slides is called the thin or compact SVD

From https://ee263.stanford.edu/lectures/svd.pdf

Proof of Existence of SVD

Outline of Proof

1. Get the eigenvectors and eigenvalues of $A^T A$
    - *These are the right singular vectors and singular values (squared) of* A
2. Use those to define the left singular vectors of A
    - *These are also the eigenvectors of* AA^T
3. Rewrite step 2 in matrix notation
    - *...and out pops the SVD*

Proof of Existence of SVD

Step 1: Get v_i and σ_i

- Note that $A^T A$ is symmetric and PSD ($A^T A \geq 0$)
- Apply spectral decomposition and note that $\lambda_i \geq 0$

$$\begin{aligned} A^T A &= V \Lambda V^T \\ &= \sum_{i=1}^{n} \lambda_i v_i v_i^T \\ &= \sum_{i=1}^{n} \sigma_i^2 v_i v_i^T \end{aligned}$$

- For any particular v_i, we have $A^T A v_i = \lambda_i v_i = \sigma_i^2 v_i$

Proof of Existence of SVD

Step 2: Define u_i

- Define u_i such that $\sigma_i u_i = Av_i$
- Note that $\sigma_i u_i$ is an eigenvector of AA^T (so u_i is too)

$$\begin{aligned} AA^T(\sigma_i u_i) &= AA^T(Av_i) \\ &= A(\sigma_i^2 v_i) \\ &= \sigma_i^2 Av_i \\ &= \sigma_i^2(\sigma_i u_i) \end{aligned}$$

Proof of Existence of SVD

Step 2: Define u_i

- Also note that u_i is a unit vector

$$
\begin{aligned}
(\sigma_i u_i)^T(\sigma_i u_i) &= (Av_i)^T(Av_i) \\
\sigma_i^2 \|u_i\|^2 &= v_i^T A^T A v_i \\
\sigma_i^2 \|u_i\|^2 &= v_i^T \lambda_i v_i \\
\sigma_i^2 \|u_i\|^2 &= \lambda_i v_i^T v_i \\
\sigma_i^2 \|u_i\|^2 &= \lambda_i \\
\sigma_i^2 \|u_i\|^2 &= \sigma_i^2 \\
\|u_i\| &= 1
\end{aligned}
$$

Proof of Existence of SVD

Step 3: Rearrange

$$\begin{bmatrix} \sigma_1 u_1 & \sigma_2 u_2 & \cdots & \sigma_n u_n \end{bmatrix} = \begin{bmatrix} Av_1 & Av_2 & \cdots & Av_n \end{bmatrix}$$

$$\begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix} = A \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$$

$$U\Sigma = AV$$

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

Proof of Existence of SVD

- Here's the insight that inspired this proof
- Given the full SVD of A, we can find U and V like this:

$$
\begin{aligned}
A^T A &= (U\Sigma V^T)^T (U\Sigma V^T) \\
&= V\Sigma^T U^T U \Sigma V^T \\
&= V\Sigma^T \Sigma V^T \\
&= V\Sigma^2 V^T
\end{aligned}
$$

$$
\begin{aligned}
AA^T &= (U\Sigma V^T)(U\Sigma V^T)^T \\
&= U\Sigma V^T V \Sigma^T U^T \\
&= U\Sigma\Sigma^T U^T \\
&= U\Sigma^2 U^T
\end{aligned}
$$

Least Squares and SVD

- SVD is deeply connected to the least squares and least norm solutions we studied earlier

Least Squares and SVD

$$
\begin{aligned}
A^\dagger &= (A^T A)^{-1} A^T \\
&= ((U\Sigma V^T)^T U\Sigma V^T)^{-1} (U\Sigma V^T)^T \\
&= (V\Sigma^T U^T U\Sigma V^T)^{-1} V\Sigma^T U^T \\
&= (V\Sigma^T \Sigma V^T)^{-1} V\Sigma^T U^T \\
&= (V\Sigma\Sigma V^T)^{-1} V\Sigma U^T \\
&= (V\Sigma^2 V^T)^{-1} V\Sigma U^T \\
&= V\Sigma^{-2} V^T V\Sigma U^T \\
&= V\Sigma^{-2} \Sigma U^T \\
&= V\Sigma^{-1} U^T
\end{aligned}
$$

Least Norm and SVD

$$\begin{aligned}
A^\dagger &= A^T(AA^T)^{-1} \\
&= (U\Sigma V^T)^T(U\Sigma V^T(U\Sigma V^T)^T)^{-1} \\
&= V\Sigma^T U^T(U\Sigma V^T V\Sigma^T U^T)^{-1} \\
&= V\Sigma U^T(U\Sigma\Sigma U^T)^{-1} \\
&= V\Sigma U^T(U\Sigma^2 U^T)^{-1} \\
&= V\Sigma U^T U\Sigma^{-2} U^T \\
&= V\Sigma\Sigma^{-2} U^T \\
&= V\Sigma^{-1} U^T
\end{aligned}$$

We get the same expression!

Generalized Pseudoinverse

- The generalized pseudoinverse of a matrix $A = U\Sigma V^T$ is

$$A^\dagger = V\Sigma^{-1}U^T$$

- *Least squares and least norm are really the same thing!*
- We don't need to worry about whether A is skinny or fat – the generalized pseudoinverse will take care of that

Next Time

- Applications of SVD