Lecture 3
Singular Value Decomposition

NLA Chapter 4 & 5

What, why, and how?

- What: Singular Value Decomposition (SVD) is a <u>matrix factorization</u> that make every matrix like a diagonal matrix.
- Why:
  - Conceptual: a powerful device to link many concepts together: rank, 2-norm, F-norm, trace, … In general, a great device for proving and understanding! (When facing a problem, ask yourself "what if I take an SVD of the matrix?)
  - Practical: SVD leads to the most **accurate** algorithm for solving linear system/least square problem; **optimal** low-rank matrix approximation; computing the 2-norm; …
- How: have to defer the computation of SVD to later lectures on eigenvalues (QR algorithms) and iterative methods (Lanczos/Arnoldi iteration).
  - QR algorithm is probably the pinnacle of (classic) NLA! Very reliable, fast, and beautiful algorithm to compute eigen/singular value/vectors.

SVD– The Swiss Army Knife…

Source: https://www.amazon.com/dp/B001DZTJRQ

Geometric Intuition

- The image of the unit sphere (S) under any $m \times n$ matrix is a hyperellipse (AS).
- Hyperellipse in $\mathbb{R}^m$ is a unit sphere stretched by some factors $\sigma_1, \sigma_2, \dots, \sigma_m$ in some **orthogonal** directions $u_1, u_2, \dots, u_m \epsilon \mathbb{R}^m$ (unit vectors).
- The vectors $\{\sigma_1 u_1, \sigma_2 u_2, \dots, \sigma_m u_m\}$ the principal semiaxes of the hyperellipse, with length $\sigma_1, \sigma_2, \dots, \sigma_m$.
- Singular values: the stretch factors $\sigma_1, \sigma_2, \dots, \sigma_m$
- n left singular vectors: $u_1, u_2, \dots, u_m \epsilon \mathbb{R}^m$
- n right singular vectors: preimages of principal semiaxes of AS, $v_1, v_2, \dots, v_m \epsilon \mathbb{R}^m$, i.e. $Au_i = \sigma_i v_i \ (\mathrm{i} = 1, \dots, n)$

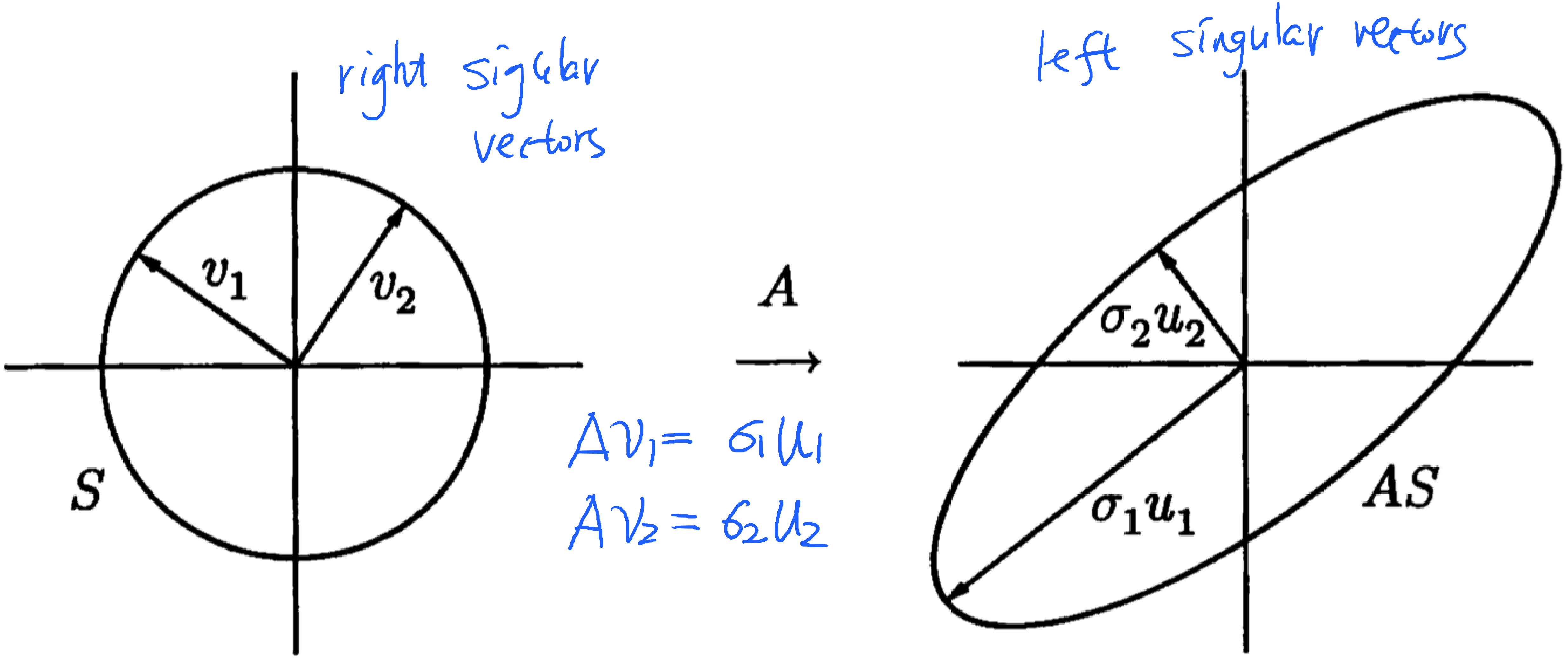


Figure 4.1. *SVD of a* 2×2 *matrix.*

Reduced SVD (1)

- The matrix $A \in \mathbb{R}^{m\times n}$ maps right singular vectors to stretched left singular vectors

$$Av_i = \sigma_i u_i, i = 1,2, \dots, n$$

- Put it in the matrix form (remember the linear combination interpretation of matrix-vector multiplication?)

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \left[\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_n \end{array}\right] = \left[\begin{array}{c|c|c|c} u_1 & u_2 & \cdots & u_n \end{array}\right] \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix},$$

$AV = \hat{U}\hat{\Sigma}$

Reduced SVD (2)

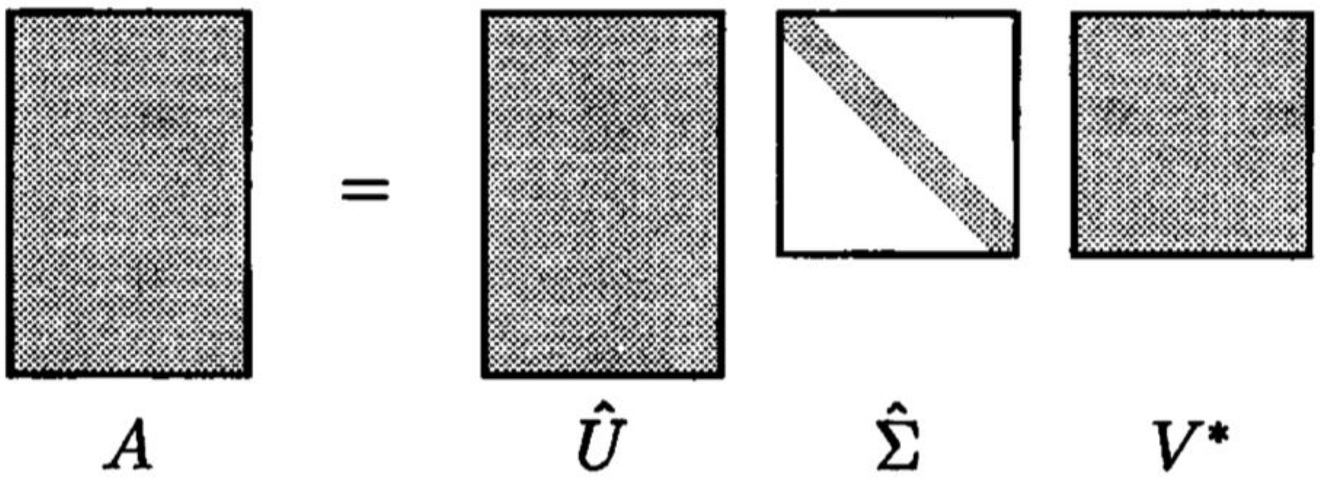
- Matrix form:

$$AV = \hat{U}\hat{\Sigma}$$

- Note that V is unitary, we have

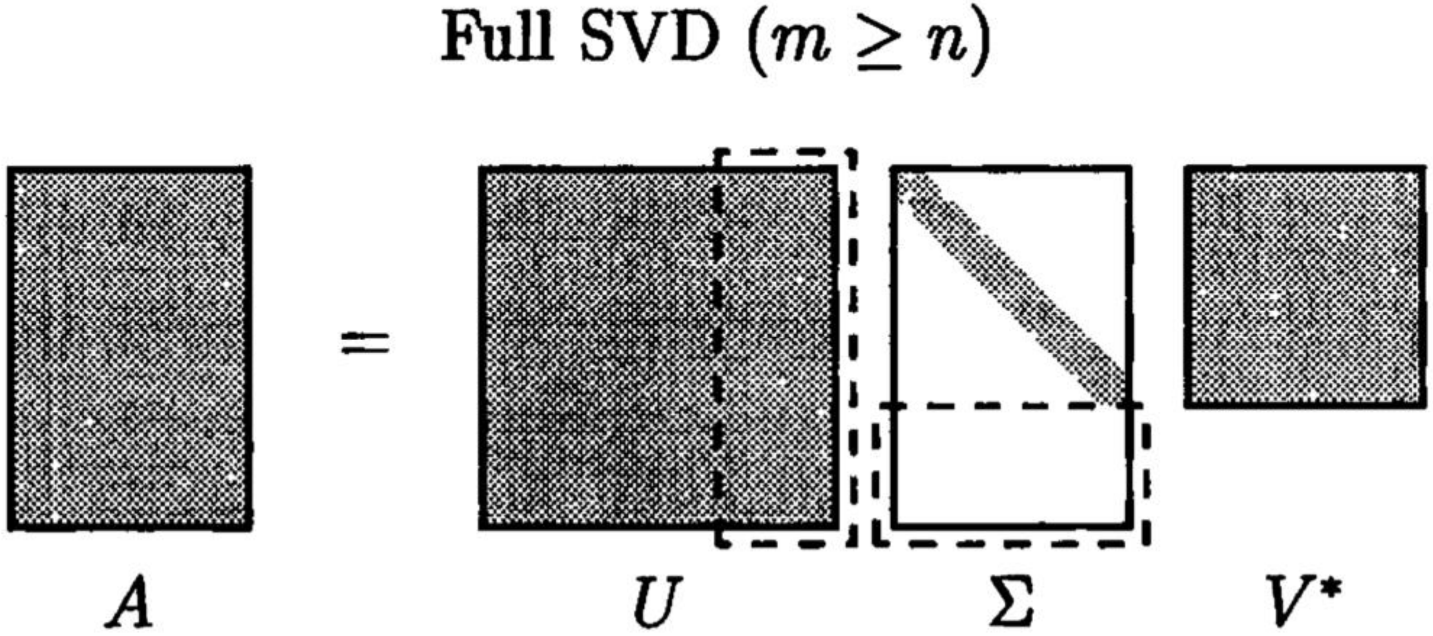
$$A = \hat{U}\hat{\Sigma}V^{\mathrm{T}}$$

Reduced SVD ($m \geq n$)



Full SVD

- Let's make the matrix $\hat{U}$ orthogonal!
  Extend the orthonormal columns of $\hat{U}$ to full $m \times m$ unitary matrix U.
  Extend the $\hat{\Sigma}$ with zeros at the bottom to maintain equality…
- Why bother? (now we allow any rank of A)



Formal definition of SVD

For **any** matrix (we focus on real ones) $A \in \mathbb{R}^{m\times n}$, a singular value decomposition (SVD) of A is a factorization

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

$\boxed{\text{SVD exists and is unique!}}$

where
$U \in \mathbb{R}^{m\times m}$ is orthogonal,
$V \in \mathbb{R}^{n\times n}$ is orthogonal,
$\Sigma \in \mathbb{R}^{m\times n}$ is diagonal.

The diagonal entries of Σ are non-negative and sorted in decreasing order:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0, \quad p = \min\{m, n\}$$

SVD: every matrix is diagonal…

If you look at it with proper basis (singular vectors) for domain/range spaces.

Whenever we have

$$b = Ax$$

We have

$$b' = \Sigma x'$$

If the SVD of A is

$$A = U\Sigma V$$

And we express vectors x, b in the basis of columns of V, U:

$$b' = U^* b, x' = V^* x$$

SVD vs. Eigenvalue Decomposition

Eigenvalue decomposition: if square matrix A is non-singular and non-defective, then it has eigenvalue decomposition:

$$A = X\Lambda X^{-1}$$

Where X matrix has the eigen vectors as columns, and Λ is a diagonal matrix with the eigenvalues on its diagonal.

Differences with SVD:

- SVD always exists, EVD only when A square & non-defective
- SVD have orthogonal bases where EVD not necessarily (but when A is symmetric, then X is orthogonal!)
- SVD is helpful when dealing with A itself or its inverse; EVD is helpful dealing with powers of X (X^n, n=1,2,…)
- Computational of SVD depends on EVD:

$$A = U\Sigma v$$
$$A^*A = V^*\Sigma^2 V$$
$$AA^* = U\Sigma^2 U^*$$

Matrix Properties via SVD

- Assumptions:
  - $A \in \mathbb{R}^{m \times n}$
  - p=min{m,n}
  - r is the number of non-zero singular values of A
  - <x,y,z> means the space spanned by vectors x,y,z
- T5.1: The rank of A is r (the number of non-zero singular values)

- T5.2: range(A) = <$u_1, u_2, \ldots, u_r$>, null(A) = <$v_{r+1}, v_{v+2}, \ldots, v_n$>

(hint: A looks like diagonal matrix under SVD…)

- T5.3:$\|A\|_2 = \sigma_1$ and $\|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2}$
  (hint: again, A looks like diagonal; orthogonal transformations conserve 2/F norm)

- T5.4: The non-zero singular values of A are the square roots of the non-zero eigenvalues of AA^* or A^*A

- T5.5: if A is symmetric, then the singular values of A are the absolute values of eigenvalues of A.

Low rank approximations

- Alternative form of SVD (the sum of rank-1):

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

- Thus, SVD decomposes a matrix A as the sum of r rank-1 matrices.
- What's special about this decomposition is that:

The partial sum captures as much energy of A as possible

SVD gives optimal low-rank approximation

Theorem 5.8. *For any ν with $0 \le \nu \le r$, define*

$$A_\nu = \sum_{j=1}^{\nu} \sigma_j u_j v_j^*;$$

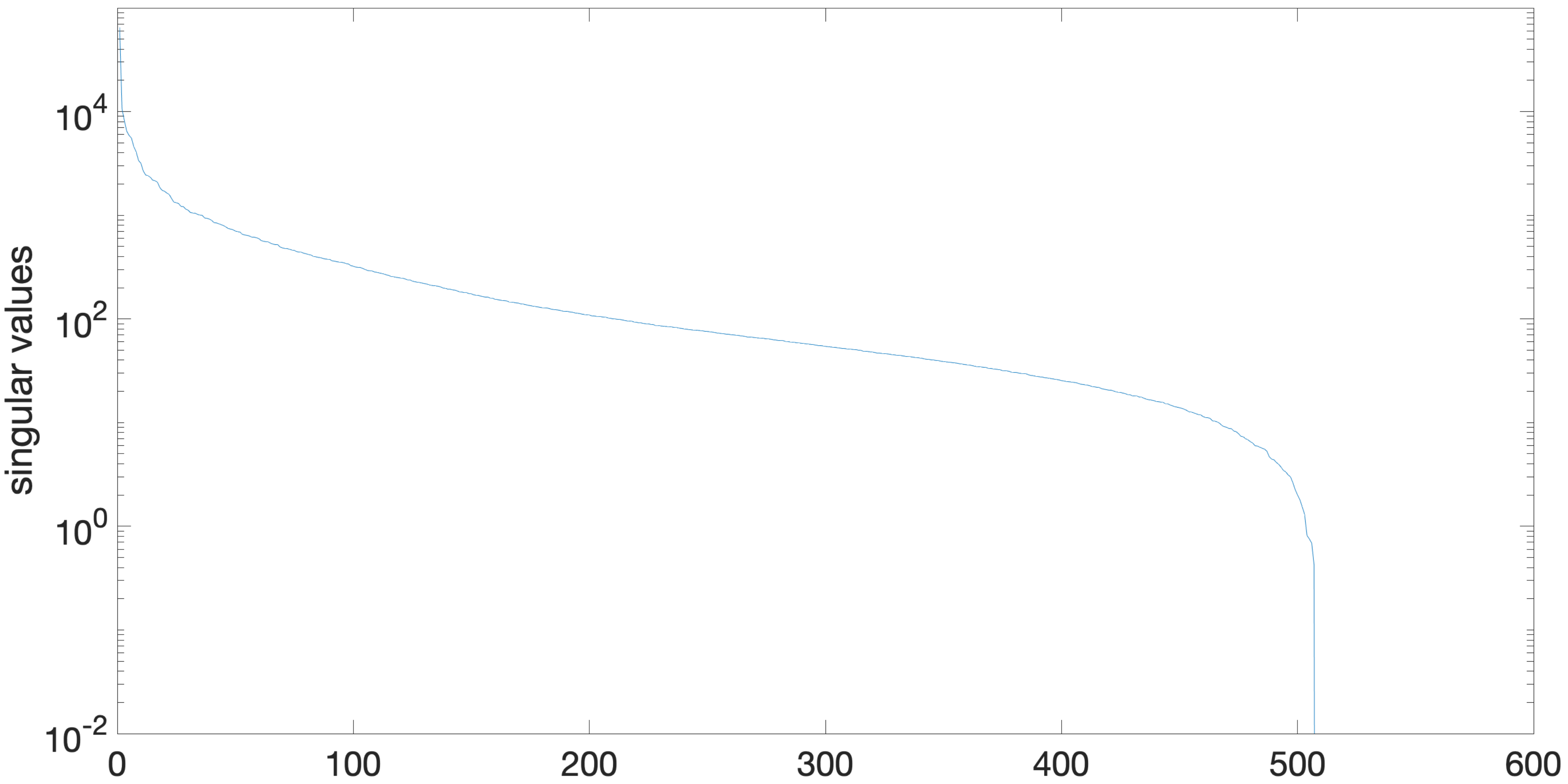
if $\nu = p = \min\{m, n\}$, define $\sigma_{\nu+1} = 0$. Then

$$\|A - A_\nu\|_2 = \inf_{\substack{B \in \mathbb{C}^{m\times n} \\ \mathrm{rank}(B) \le \nu}} \|A - B\|_2 = \sigma_{\nu+1}.$$

Example Application of SVD

- Image compression. We have a 512x512 pixels Lena picture as a 512x512 matrix A


rank=10
rank=20
rank=40
rank=80
rank=160
rank=512


singular values
10^4
10^2
10^0
10^-2
0
100
200
300
400
500
600

Applications in Data Science

- Semantic Analysis
- Collaborative filtering/recommendation (Netflix Prize!)
- Pseudo-inverse (used for least square problem)
- Data compression
- Principal component analysis
- …