Recap: SVD and Low-Rank Approximation

DS701 Session 14 — Wed Oct 21, 2026

Today’s plan

Knowledge-Check Review

KC 1: The three factors, and the sum of rank-1 pieces

Write down the singular value decomposition of a matrix $A \in \mathbb{R}^{m \times n}$ (with $m > n$) and say, for each of the three factors, what it contains and what its shape is. Then write the same decomposition as a sum of rank-1 matrices, and say what a singular value σ_i tells you in that form.

KC 2: Full rank, low *effective* rank

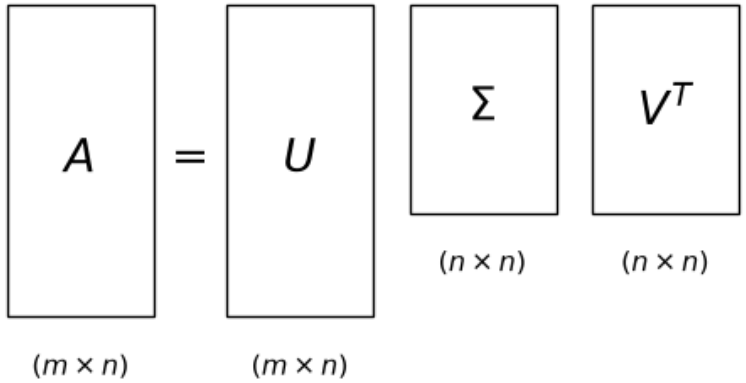
The Abilene traffic matrix is 1008×121 *and has rank 121 — it is full rank — yet the lecture says it has* ***low effective rank*** *and that a rank-20 approximation is within 9% of it. Explain what "low effective rank" means, how the* plot of singular values* reveals it, and how the formula $\|A - A^{(k)}\|_F^2 = \sum_{i>k} \sigma_i^2$ lets you read the approximation error off that plot without ever forming $A^{(k)}$.*

KC 3: Twenty numbers per person

In the Netflix example the ratings matrix is $500{,}000$ *users* $\times$ $18{,}000$ *movies, and the winning approach modelled it as having rank 20–40. Using the factorization* $A^{(k)} \approx \tilde{U}\tilde{V}$ *with* $\tilde{U} \in \mathbb{R}^{m\times k}$ *and* $\tilde{V} \in \mathbb{R}^{k\times n}$*, explain what a* ***row of*** $\tilde{U}$ *and a* ***column of*** $\tilde{V}$ *represent, how a single rating is computed from them, and why this is a* simplification* of the data in Occam's sense — one that could predict a rating the user never gave.*

Highlights

One factorization, three ways to read it

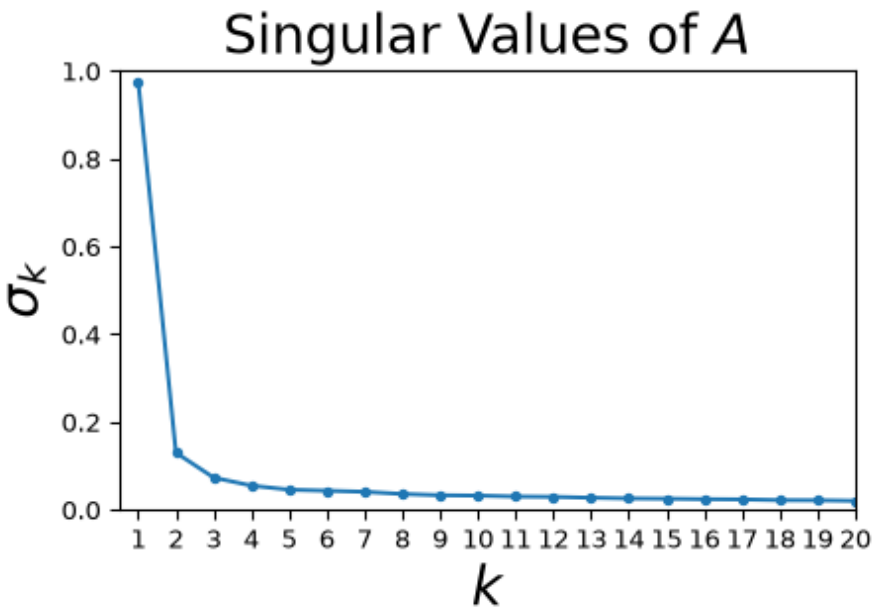


$$A = U\Sigma V^T, \qquad A\mathbf{v}_i = \sigma_i \mathbf{u}_i$$

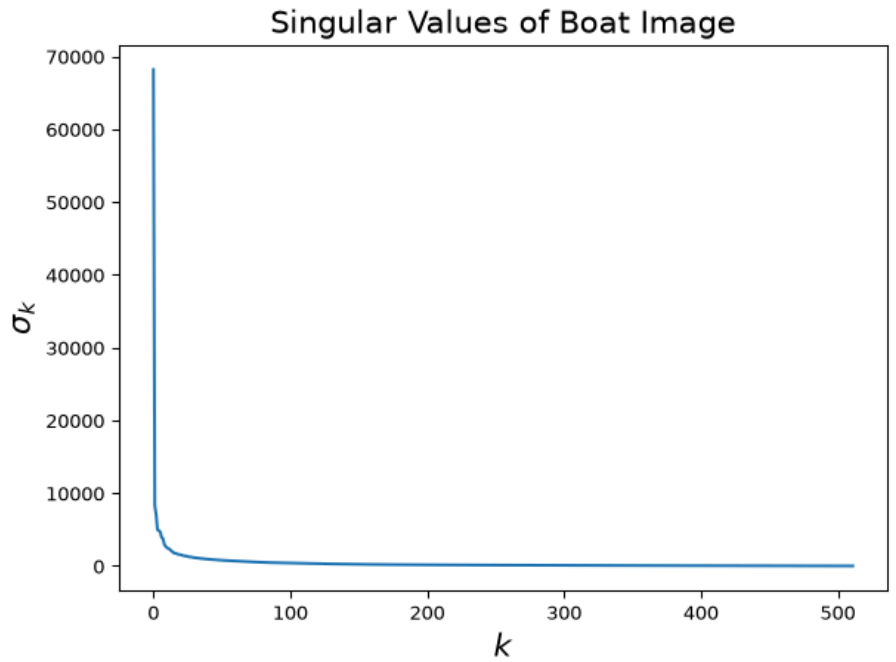
Rank, eigenvalues, singular values

The refresher woven into this session — three facts you will verify numerically today:

The spectrum is an importance ranking



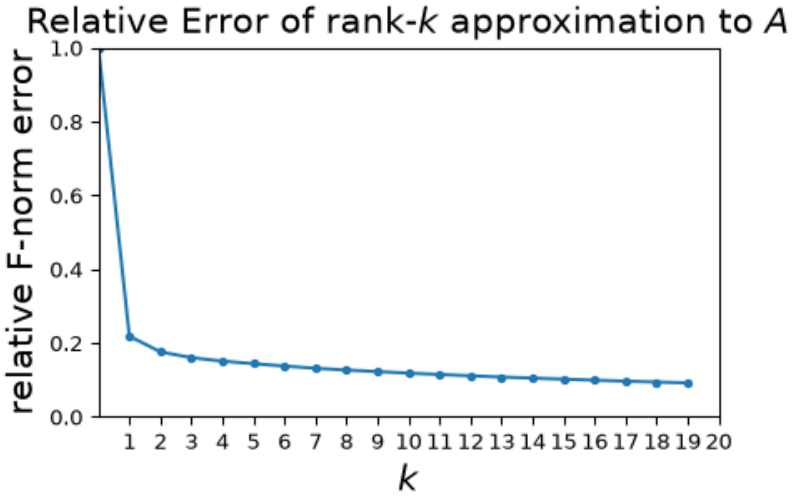
Traffic matrix, 1008×121, rank 121.



Boat photo, 512×512, rank 512.

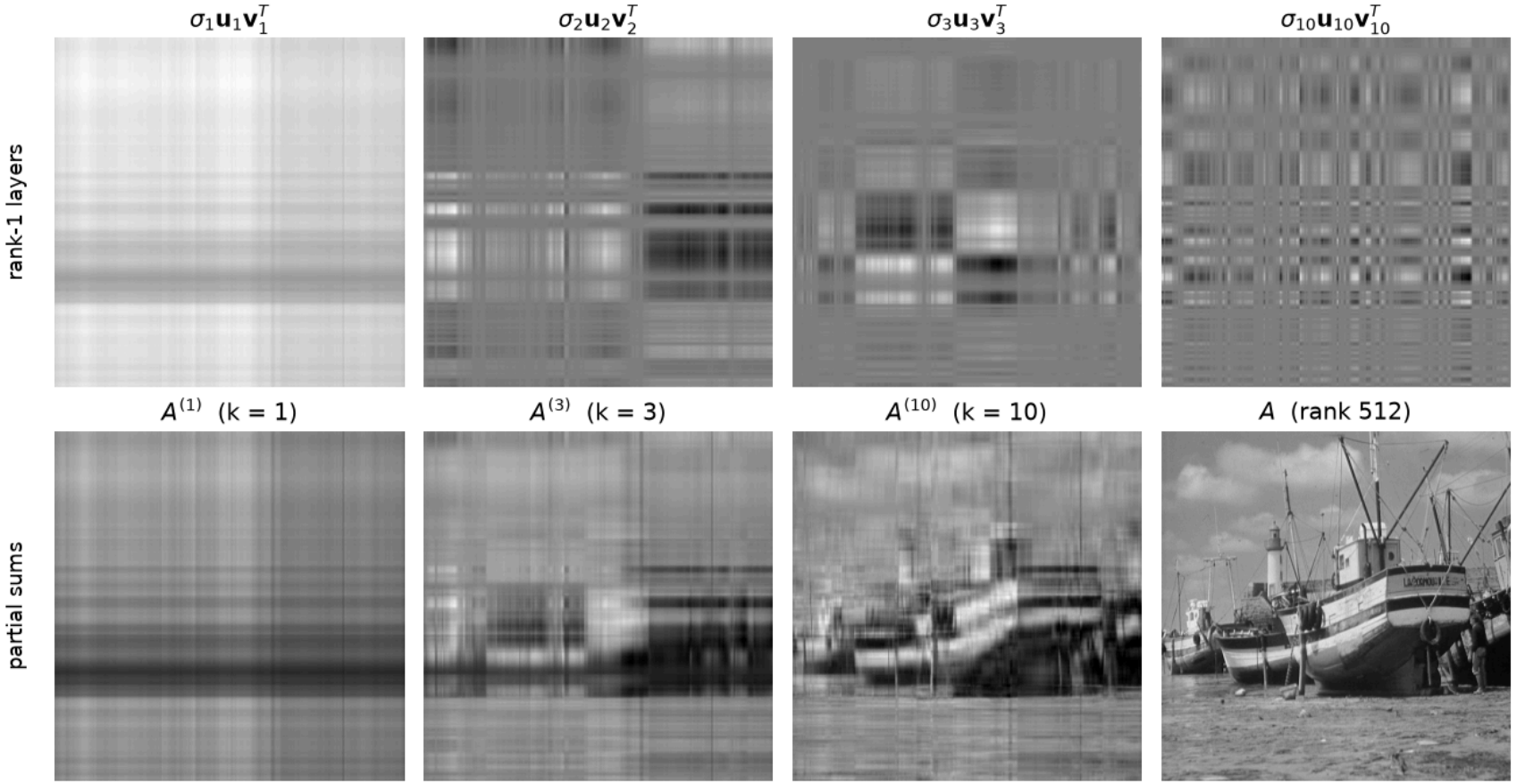
Eckart–Young: the best rank-k approximation

Distance between matrices: Frobenius, $\|A - B\|_F = \sqrt{\sum_{ij} (a_{ij} - b_{ij})^2}$ — Euclidean distance in $\mathbb{R}^{mn}$.



Relative error read straight off the singular values: 9% at $k = 20$.

Layers of a photograph



Compression: what you keep, what you pay

Rank 40 Boat

Rank 512 Boat

Rank 40 of 512:

Two readings of a low-rank matrix

Common patterns — columns of U.

$\mathbf{a}_j \approx \sum_{i \le k} v_{ji}\sigma_i \mathbf{u}_i$: every column of the data is a mix of the same k patterns; $\mathbf{u}_1$ is the strongest one (traffic: the daily rhythm shared by all 121 traces).

Latent factors — rows of $U\Sigma$ and columns of V^T.

$a_{ij} = \tilde{U}_{i,:} \cdot \tilde{V}_{:,j}$: user i and item j are both points in $\mathbb{R}^k$, and the entry is their inner product. Netflix: $k \approx 20$.

Where the SVD goes from here

	matrix	what the SVD gives	session
compression / denoising	image, traffic	$A^{(k)}$: fewest numbers for a given error	today
PCA	centred data X (n samples $\times$ d features)	$\mathbf{v}_i$ = principal directions, $\sigma_i^2/(n-1)$ = variance explained	15
recommenders	users $\times$ items, mostly *missing*	rows of $U\Sigma$ / columns of V^T = latent tastes; fill the blanks	22

In-Class Activity

Activity: compress a photo, then find the hidden factors

Goal: take a grayscale image apart with `np.linalg.svd`, rebuild it at rank 1, 5, 20 and 50, measure Frobenius error, energy and compression, and check Eckart–Young against two rank-20 competitors; verify $\Sigma^2 = \mathrm{eig}(A^T A)$ and rank = number of non-zero singular values; then take the SVD of a synthetic user × item ratings matrix with planted taste factors, *see* the factors in the top-2 latent dimensions, and predict held-out ratings at rank 2 versus full rank.

- Work in groups of 2–3. Open **your section's** notebook — the image and the ratings data differ.
- Parts marked **(autograded)** are submitted to Gradescope; the rest is participation.
- Staff will circulate — be ready to explain any part of your work.
- Dependencies: `numpy`, `pandas`, `matplotlib`, `scikit-learn`.

Section A1 Open in Colab Open on GitHub

Going deeper

Full lecture notes: SVD — Low Rank Approximations

- Singular Vectors and Values and the decomposition — definitions, shapes, properties
- Outer Products — the SVD as a weighted sum of rank-1 matrices
- Matrix Rank and column space — the $k(m+n)$ storage argument
- Low Effective Rank and Finding Rank-k Approximations — Frobenius norm, Eckart–Young, the error formula
- Traffic Data — spectrum, elbow, relative error, storage
- Images — the boat at rank 40
- Common Patterns and Latent Factors — the two readings; the Netflix example
- Refresher: eigenvalues and eigenvectors, eigendecomposition, SVD
- Coming up: PCA (Session 15) and Recommender Systems (Session 22)