Recap: Recommender Systems

DS701 Session 22 — Wed Nov 18, 2026

Today's plan

Knowledge-Check Review

KC 1: Predicting a missing rating

In item–item collaborative filtering, how is a missing rating $\hat{r}_{ui}$ *predicted? Name the two ingredients (what is compared to what, and how the pieces are combined), and say why the ratings matrix being 99.98% empty is the central difficulty rather than a detail.*

KC 2: “Something like an SVD”

The lecture said we want “something like an SVD” of the ratings matrix but cannot use the SVD algorithm directly, and instead minimises $\|(R - UV^T)_S\|^2 + \lambda(\|U\|^2 + \|V\|^2)$ *by alternating least squares. Explain (a) why the SVD cannot be applied as-is, (b) what the subscript* S *does to the objective, and (c) why holding* U *fixed turns the problem into one you already know how to solve.*

KC 3: The latent space, again

In Session 14 you took the SVD of a small ratings matrix with planted taste factors and saw users and items land in a shared latent space. Connect that to today: what does a row of U and a row of V mean* in the recommender's factorization, how is a rating formed from them, and what can this shared space do that item–item CF cannot? Then name one situation in which neither CF nor MF has anything to say.*

Highlights

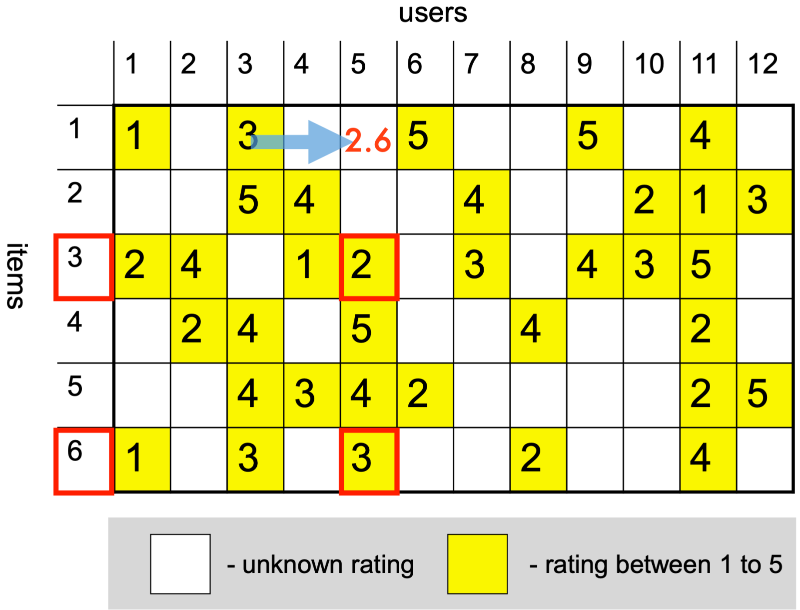
The user–item matrix — and the hole in it

users

items	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3			5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

- unknown rating (white) - rating between 1 to 5 (yellow)

Collaborative filtering: neighbours in rating space



items \ users	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		2.6	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Item–item: to guess r_{ui}, look at items **like** i that u **has rated**, and average.

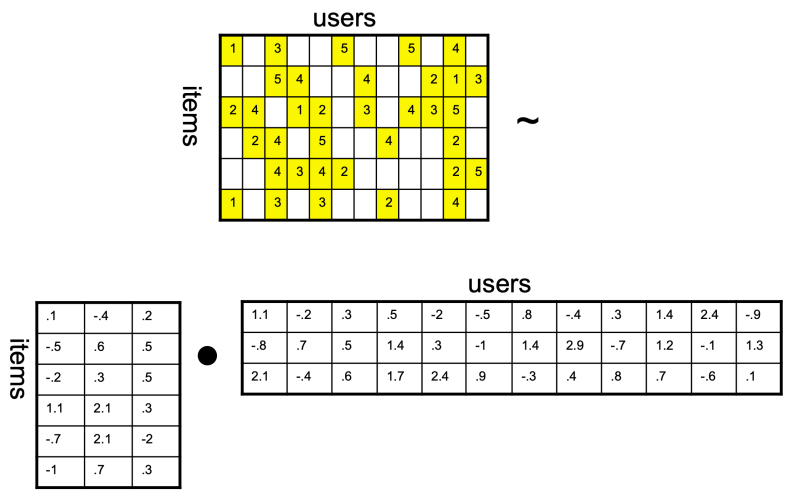
$$\hat{r}_{ui} = \frac{\sum_{j \in n_k(i,u)} s_{ij} \, r_{uj}}{\sum_{j \in n_k(i,u)} s_{ij}} \qquad \frac{0.2 \cdot 2 + 0.3 \cdot 3}{0.2 + 0.3}$$

Correct for bias first

Some users rate everything high; some items are simply loved. Left in, that noise masquerades as similarity.

$$b_{ui} = \mu + \alpha_u + \beta_i \qquad \text{(global mean, user offset, item offset)}$$

Matrix factorization: low rank = latent tastes



CF uses item similarity *or* user similarity. MF puts both in **one latent space**:

ALS in one slide

Jointly convex: fix one factor and the objective is convex in the other.

1. Hold U fixed → solve for V
2. Hold V fixed → solve for U
3. Not converged? Go to 1.

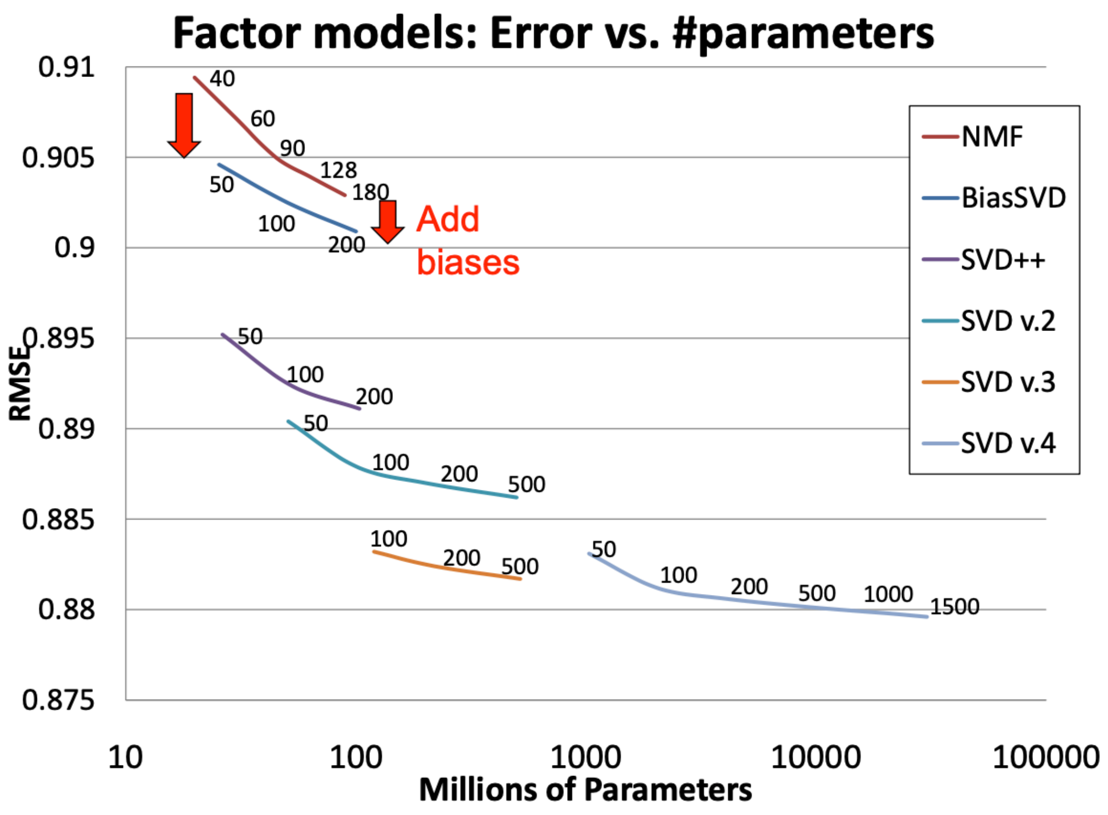
Each half-step is a **ridge regression per row**, restricted to that row's observed entries:

$$\mathbf{u}_u = \left(V_{J_u}^\top V_{J_u} + \lambda I\right)^{-1} V_{J_u}^\top \mathbf{r}_{u,J_u}$$

J_u = the items user u rated, V_{J_u} = their factor rows. Every step lowers the objective → converges.

MF in practice (lifted from Recommender Systems II):

Evaluate on ratings you hid



The standing limitation: cold start

Everything on the previous slides needs a **history**.

In-Class Activity

Activity: collaborative filtering and matrix factorization by hand — then cold start

Goal: on a synthetic streaming service with planted genres and 70% missing ratings, build item–item CF (adjusted-cosine similarity, k-NN prediction), implement ALS for matrix factorization and check the learned factors recover the genres; then experience the cold-start problem — a new user with 0, 1, 3, 10 ratings, a new item, and one hybrid fix.

- Work in groups of 2–3. Open **your section's** notebook — the service differs.
- 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: Recommender Systems

- Reviews are Sparse and Sparseness is Skewed — the Amazon numbers
- Item–Item CF — the worked $\hat{r}_{15}$ example, step by step
- Similarity — Pearson on the common support; binary data
- Improving CF in the presence of bias — the bias model and how it is estimated
- Negative Similarity Scores? and Common Solutions
- Matrix Factorization and Solving Matrix Factorization — the objective and ALS
- ALS in Practice — the dense-block run
- Assessing Matrix Factorization — the Netflix Prize improvements
- The latent-space picture from Session 14: Low Rank Defines Latent Factors
- Where the factors go next: Recommender Systems II — embeddings and DLRM