Collaborative Filtering for Movie Ratings

Team name: better_than_random

Anthony Boulos
Rebecca El Chidiac
Nadezhda Zhukova

Data Science Lab

October 15, 2025

Problem Statement — Movie Recommendation

Goal. Predict the *missing ratings* in a sparse user–item matrix.

Setting.

- Observed ratings are few $\Rightarrow$ strong sparsity, cold users/items.
- Predict $\widehat{r}_{ui}$ for unseen (u, i) pairs.
- Evaluate with **RMSE**.

Fig. 1. Example of a user–item matrix.

Alternating Least Squares (ALS)

Goal: Learn low-dimensional latent representations for users and items by minimizing the reconstruction error of the observed matrix: $\boxed{\mathrm{R} \approx UV^\top}$

Objectif:

$$\min_{U,V} \sum_{(u,i)\in\Omega} (R_{ui} - U_u^\top V_i)^2 + \lambda \left(\|U\|_F^2 + \|V\|_F^2\right)$$

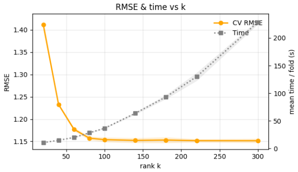
- $R \in \mathbb{R}^{m\times n}$: user–item ratings matrix.
- $U \in \mathbb{R}^{m\times k}$: user latent factors.
- $V \in \mathbb{R}^{n\times k}$: item latent factors.
- λ : regularization coefficient.

Optimization: Alternating between user and item updates via regularized least squares for a fixed number of iterations n_iters:

$$U_u \leftarrow (V_\Omega^\top V_\Omega + \lambda I)^{-1} V_\Omega^\top R_{u,\Omega}, \quad V_i \leftarrow (U_\Omega^\top U_\Omega + \lambda I)^{-1} U_\Omega^\top R_{\Omega,i}$$

Hyperparameters ALS

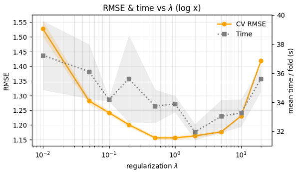
k (latent dimension)



$\uparrow k \rightarrow$ RMSE $\downarrow$ then plateau;

time $\nearrow$.

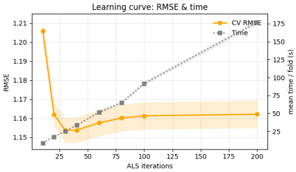
λ (regularization, log-x)



Small $\lambda \rightarrow$ overfit,
large $\lambda \rightarrow$ underfit;

time - stable.

n_iter (ALS iterations)



RMSE improves quickly,

then stalls.

ALS with Genre-Enriched Item Factors

Goal: Enhance the pure ALS model by injecting genre information into the latent representation of items: $\boxed{\mathsf{R} \approx U(V + GW)^\top}$

Objectif:

$$\min_{U,V,W} \sum_{(u,i)\in\Omega} (R_{ui} - U_u^\top (V_i + G_i W))^2 + \lambda(\|U\|_F^2 + \|V\|_F^2) + \gamma(\|W\|_F^2)$$

- $G \in \mathbb{R}^{n\times d}$: one-hot genre matrix.
- $W \in \mathbb{R}^{d\times k}$: learnable projection from genres to latent space.
- Each item factor becomes $V_i + G_i W$.

Optimization: Alternating least-squares updates for U, V, and W:

$$U_u \leftarrow (V_\Omega^\top V_\Omega + \lambda I)^{-1} V_\Omega^\top R_{u,\Omega}$$
$$V_i \leftarrow (U_\Omega^\top U_\Omega + \lambda I)^{-1} U_\Omega^\top (R_{\Omega,i} - U_\Omega^\top G_i W)$$
$$W \leftarrow (G^\top G + \gamma I)^{-1} G^\top (R - UV^\top)$$

Graph-Regularized Matrix Factorization (Items Only)

Goal:
Learn latent factors U, V such that connected items have similar embeddings.

Build item similarity graph:

$$S_{ij} = \cos(\mathsf{genre}_i, \mathsf{genre}_j) = \frac{\mathsf{g}_i \cdot \mathsf{g}_j}{\|\mathsf{g}_i\| \, \|\mathsf{g}_j\|} \;\Rightarrow\; D_{ii} = \sum_j S_{ij}, \; L_v = D_v - S_v$$

Idea: Add a *Laplacian smoothness term* to the standard MF objective.

$$\min_{U,V} \sum_{(u,i)\in\Omega} (R_{ui} - U_u^\top V_i)^2 + \lambda(\|U\|_F^2 + \|V\|_F^2) + \alpha \, \mathrm{Tr}(V^\top L_v V)$$

Interpretation

Penalty:

$$\mathrm{Tr}(V^\top L_v V) = \frac{1}{2}\sum_{i,j} S_{ij}\|V_i - V_j\|^2$$

Interpretation:

L_v is the item graph Laplacian, built from genre-based cosine similarities.
Encourages similar movies to have similar latent representations.

Updated ALS Optimization:

$$V_i \leftarrow \left(U_\Omega^\top U_\Omega + (\lambda + \alpha D_{ii}) I\right)^{-1} \left(U_\Omega^\top R_{\Omega,i} + \alpha \sum_j S_{ij} V_j\right)$$

Results and Interpretation

Hyperparameters choice : RandomSearch then GridSearch on a more restrained area.

Model	RMSE	Time (s)	Comments
Baseline ALS	0.985	30.34	Standard latent factor model
Genre-Enriched ALS	0.861	230.06	Adds semantic information via genres
Graph-Regularized MF	0.946	56.5	Smooths similar items through Laplacian regularization

Table: Evaluation performed on the validation set of the platform.

Observation: Genre information and graph regularization both improve accuracy. The genre-enriched model achieves the best RMSE, while Laplacian regularization offers a good trade-off between accuracy and runtime.

Next steps

1. Add Laplacian regularization on users too.
2. Compare against a combination of LSH for items and users.

Details. Combine user- and item-based neighbors to fill missing ratings: handle cold users via similar items, and cold items via similar users.
The final prediction is a weighted combination:

$$\widehat{r}_{ui} = \alpha\,\widehat{r}_{ui}^{(u)} + (1-\alpha)\,\widehat{r}_{ui}^{(i)}$$

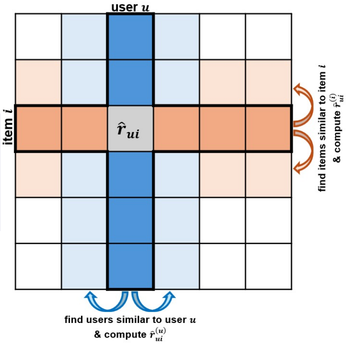


Fig. 2. Computing $\widehat{r}_{ui}$ via UBCF (column) + IBCF (row).