A Hybrid Matrix-Factorization Recommender on MovieLens-25M

Authors: Aniket Gulab Khedkar
Contact: aniketgk@umich.edu
Project Duration: June 2025 – July 2025
Date: July 2025

Abstract

We build an end-to-end movie recommendation system using the **MovieLens-25M** dataset. The system covers data ingestion, cleaning, exploratory analysis, problem formulation, algorithm design, offline evaluation, error analysis, and deployment considerations. We compare simple popularity and item-similarity baselines against (i) **Explicit MF with biases** (Koren et al., 2009) and (ii) **Implicit WRMF** (Hu-Koren-Volinsky, 2008) on binarized feedback, then **blend**collaborative scores with **content** features (genres/tags) to mitigate cold-start. We evaluate with ranking metrics (Precision@K, Recall@K, NDCG@K) and rating error (RMSE/MAE), and outline reproducibility and ethical considerations. Dataset facts and licensing follow GroupLens/Kaggle documentation. GroupLensKaggle

Keywords: recommender systems, collaborative filtering, matrix factorization, hybrid models, MovieLens-25M, implicit feedback, ranking metrics.

---

1. Introduction

Recommender systems personalize large catalogs to improve user satisfaction and business outcomes. Matrix-factorization (MF) methods popularized during the Netflix Prize remain strong baselines due to robustness, scalability, and interpretability, while modern practice emphasizes **top-K ranking** metrics and **implicit signals**. We adopt MF with biases for explicit ratings and WRMF for implicit interactions, and combine them with light content signals (genres/tags) in a hybrid. datajobs.comyifanhu.net

---

2. Related Work

Latent factor/biased MF. Models user/item vectors and bias terms, minimizing squared error with L2 regularization; strong results across rating datasets. datajobs.com
Implicit feedback WRMF. Treats observations as positive with confidence weights; optimizes a weighted squared loss via ALS and scales to large data. yifanhu.netchrisvolinsky.com
Evaluation shift to ranking. Precision@K/Recall@K and NDCG@K are standard for top-N evaluation, often preferred over pure RMSE. evidentlyai.com+1Shaped
Temporal leakage caution. Chronological splits reduce leakage when using time-stamped interactions (e.g., MovieLens-25M). arXiv

---

3. Data

3.1 Source & Scope

MovieLens-25M: ~25,000,095 ratings and ~1,093,360 tag applications on ~62,000 movies from ~162,000 users (1995-11-21 through 2019-11-21). We use ratings.csv, movies.csv, tags.csv and optionally genome-scores/tags. [GroupLens](#) Mirror copies on Kaggle provide convenient access for experiments. Always respect GroupLens license/attribution terms. [Kaggle](#)

3.2 Fields

- **ratings.csv:** userId, movieId, rating (0.5–5.0), timestamp
- **movies.csv:** movieId, title, genres (pipe-separated)
- **tags.csv:** userId, movieId, tag, timestamp
- **genome-scores/tags:** dense "tag genome" relevance matrix (optional). [GroupLens](#)

3.3 Pre-processing

- Remove users/items with < **5** interactions to stabilize factors.
- Map IDs to contiguous indices; convert timestamps to datetime; ensure UTC consistency.
- **Implicit view** (for WRMF): label y_ui = 1 if rating ≥ 3.5, else 0; set confidence c_ui = 1 + α·r_ui_cnt, with α tuned. [yifanhu.net](#)

---

4. Problem Formulation

4.1 Rating Prediction (explicit)

Given observed ratings R∈RU×IR∈RU×I, predict r^ui=μ+bu+bi+puTqir^ui=μ+bu+bi+puTqi.
Objective:

min⁡b*,P,Q∑(u,i)∈Ω(rui−μ−bu−bi−puTqi)2+λ(‖P‖F2+‖Q‖F2+‖b‖22)b*,P,Qmin(u,i)∈Ω∑(rui−μ−bu−bi−puTqi)2+λ(‖P‖F2+‖Q‖F2+‖b‖22)

optimized by SGD/ALS. [datajobs.com](#)

4.2 Top-N Ranking (implicit)

For binarized feedback, WRMF solves:

min⁡P,Q∑u,icui(pui−puTqi)2+λ(‖P‖F2+‖Q‖F2),P,Qminu,i∑cui(pui−puTqi)2+λ(‖P‖F2+‖Q‖F2),

where pui∈{0,1}pui∈{0,1} and cui=1+αruicui=1+αrui. Optimized by alternating least squares (ALS). [yifanhu.net](#)

---

5. Methods

5.1 Baselines

- **Global mean / user-mean / item-mean** predictors.
- **Popularity@K:** top-rated or most-rated items globally; useful for cold-start and as a sanity check.

5.2 Item-kNN (Cosine)

Compute item–item cosine similarity on user-centered ratings; score user uu for item ii via weighted sum over kkneighbors.

5.3 Explicit MF with Biases ("MF-Bias")

Implementation via SGD (factors k=64k=64, learning rate 1e−21e−2, λ=1e−2λ=1e−2); early stopping on validation RMSE; clip predictions to [0.5, 5.0]. [datajobs.com](#)

5.4 Implicit WRMF (ALS)

Binarize interactions; set α∈{10,20,40}α∈{10,20,40}, k∈{64,128}k∈{64,128}, regularization $\lambda \in \{0.05, 0.1\}\lambda \in \{0.05, 0.1\}$, **15–20** ALS iterations with user/item coordinate updates. [yifanhu.net]

5.5 Content & Hybrid

- **Content vectors**: TF-IDF over **genres** and lemmatized **tags** (down-weight rare tags).
- **Hybrid blend**: $s_{hyb}(u,i) = \beta \cdot s_{CF}(u,i) + (1-\beta) \cdot s_{CB}(u,i) s_{hyb}(u,i) = \beta \cdot s_{CF}(u,i) + (1-\beta) \cdot s_{CB}(u,i)$, $\beta \in [0,1]\beta \in [0,1]$ tuned on validation NDCG@K.
- **Cold-start**: back-off to content rank + popularity prior when CF evidence is scarce.

---

6. Experimental Protocol

6.1 Splits

We perform **chronological splits per user** to avoid leakage: for each user, sort by timestamp, use **80% train / 10% validation / 10% test**. For ranking, we **filter out** items the user interacted with in train when scoring the candidate set ("all-items except seen"). [arXiv]

6.2 Metrics

- **RMSE/MAE** on ratings (explicit MF only).
- **Precision@K, Recall@K** with $K \in \{5,10,20\} K \in \{5,10,20\}$.
- **NDCG@K** to weight hit positions (log discount). Metric definitions follow standard practice in ranking systems. [evidentlyai.com+1Shaped]

6.3 Hyperparameter Tuning

Grid search on $k, \lambda, \alpha, \beta k, \lambda, \alpha, \beta$ using validation NDCG@10; early stopping on RMSE (explicit) and NDCG@10 (implicit/hybrid).

---

7. Results & Analysis

(Guidance for interpreting your outputs)

1. **Popularity** provides strong P@K on short tails but low personalization; expect lowest NDCG.
2. **Item-kNN** improves on head genres; suffers with sparse users.
3. **MF-Bias** typically **reduces RMSE** vs. baselines and improves NDCG@K through latent structure. [datajobs.com]
4. **WRMF (ALS)** generally **wins on top-K ranking** under implicit binarization, particularly for highly active users; tune $\alpha\alpha$ carefully. [yifanhu.net]
5. **Hybrid** improves **cold-start** and niche items via tag/genre signals; blending weight $\beta\beta$ often lands in **0.6–0.8** when CF is reliable, lower when data is sparse.

Tip: When reporting, include per-segment metrics (e.g., new vs. experienced users; long-tail items) and calibration plots of score quantiles vs. hit-rate.

---

8. Ablations & Diagnostics

- **Sparsity sensitivity:** vary min-interactions (5/10/20).
- **Temporal holdout:** compare random vs. chronological splits; show leakage effect delta. [arXiv]
- **Confidence weight $\alpha\alpha$:** plot NDCG@10 vs. $\alpha\alpha$ for WRMF.

- **Cold-start:** evaluate users/items unseen in train using content/popularity only.

---

9. Deployment Considerations

- **Candidate generation** (WRMF) → **re-ranker** (hybrid/content rules).
- **Freshness:** schedule periodic ALS re-fits; warm-start factors for new data.
- **Explainability:** surface top contributing genres/tags and nearest-neighbor rationales.
- **Monitoring:** track coverage, novelty, and user-side metrics (click-through, dwell) alongside Precision@K drift.

---

10. Ethical & Responsible AI Notes

- **Bias & representation:** popular/franchise bias; consider diversity/serendipity constraints.
- **User controls:** allow opt-outs, content filters, and feedback toggles.
- **Privacy:** MovieLens is de-identified; in production, follow data minimization and retention policies.

---

11. Reproducibility Checklist

- Dataset: **MovieLens-25M** (GroupLens; Kaggle mirror). Include README and license notice in repo. GroupLensKaggle
- Environment: Python ≥3.10; core libs: pandas, scipy, scikit-learn, implicit (for WRMF ALS), lightfm(optional), matplotlib.
- Random seeds fixed; deterministic ALS where possible.
- Publish: config YAML with chosen hyperparameters; log metrics and timings.

---

12. Conclusion

On MovieLens-25M, **latent factor models** remain strong, especially **WRMF** for top-N recommendation. A lightweight **hybrid** that blends collaborative scores with **genres/tags** improves coverage and cold-start without heavy feature engineering. Chronological evaluation and ranking-centric metrics are critical for realistic assessment. Future work includes sequence-aware models and graph-based recommenders.

References

- **MovieLens-25M Dataset** (official description & download). GroupLens
- **Kaggle Mirror – MovieLens-25M** (convenient access & notebooks). Kaggle
- **Koren, Bell, Volinsky (2009)**: *Matrix Factorization Techniques for Recommender Systems.* Foundational biased MF for ratings. datajobs.comHasso-Plattner-Institut
- **Hu, Koren, Volinsky (2008)**: *Collaborative Filtering for Implicit Feedback*

Datasets. WRMF objective & ALS optimization. yifanhu.netchrisvolinsky.com
- **Ranking Metrics Guides**: Precision/Recall@K, NDCG overviews. evidentlyai.com+1Shaped
- **Temporal Leakage Discussion** (use time-aware splits). arXiv

---

Appendix A — Dataset Facts (for your report's "Data" table)

- Ratings: **25,000,095**; Users: **~162,000**; Movies: **~62,000**; Tags: **~1,093,360** applications; Date range: **1995–2019**. GroupLens

Appendix B — Suggested Hyperparameters (starting points)

- **MF-Bias (explicit):** factors=64, lr=1e-2, reg=1e-2, epochs=20–30, early-stop on val RMSE. datajobs.com
- **WRMF (implicit):** factors=128, reg=0.1, $\alpha \in \{10,20,40\}$, iters=15–20; use CG-optimized ALS from implicit.