Collaborative Filtering on Implicit Feedback

Project Report

Federico Vaona

VR495585

University of Verona – Data Mining Exam

1 Introduction

Recommender systems have become a fundamental tool in a wide range of online services, from e-commerce platforms to streaming applications. Their main goal is to assist users in navigating extremely large catalogs of items by providing personalized suggestions, thereby improving user experience and engagement. Traditionally, many of these systems rely on explicit feedback, such as ratings or reviews, where users directly express their preferences. However, in real-world scenarios explicit feedback is often sparse, costly to collect, and sometimes unreliable, since users may be unwilling or unable to provide explicit ratings consistently.

For these reasons, implicit feedback is of great practical interest. Implicit data, such as listening logs, browsing history, or purchase records, is abundant and can be collected passively, without requiring active user input. At the same time, working with implicit feedback presents unique challenges: the absence of explicit negative signals, the noisy nature of interactions, and the difficulty of distinguishing between true preferences and casual or accidental consumption.

This project focuses on these challenges by reimplementing the method proposed in the paper *"Collaborative Filtering for Implicit Feedback Datasets"* by Hu, Koren, and Volinsky (2008). The model introduced in this work is specifically designed to handle implicit feedback through a confidence-weighted matrix factorization approach. The method introduces the key idea of separating binary preferences from confidence levels, thus capturing both the presence of user interactions and their varying reliability.

The implementation has been tested on the Last.fm 1K dataset, which provides real listening histories of nearly one thousand users. In order to make the experiments computationally feasible within the scope of the exam project, the dataset was reduced to the first 500 users, still maintaining a significant amount of interactions. The system was then compared with simpler baselines such as popularity ranking and item–item collaborative filtering, in order to highlight the added value of the latent factor approach.

The broader objective of this work is not only to reproduce the results of the original paper, but also to critically evaluate the behavior of the algorithm on a different dataset, discuss its strengths and limitations, and provide a comparative analysis against alternative methods. In this way, the project demonstrates how research methods can be reimplemented and assessed on a real dataset, highlighting both their strengths and their practical limitations.

2 Dataset

The experiments are based on the **Last.fm 1K** music listening dataset, which contains complete listening histories for one thousand users. For feasibility and to align with the project constraints, we created a custom reduced version of the dataset by selecting the first 500 users from the original collection. This filtering step produced a smaller but still representative dataset, with approximately 2.5 million interactions, which makes the computational experiments manageable while preserving enough data to evaluate recommendation quality. The key characteristics are:

- **Source**: Last.fm 1K dataset (2009 snapshot)
- **Content**: User listening history (user, timestamp, artist, track)
- **Profiles**: User demographics (gender, age, country, signup date)
- **Subset**: First 500 users, approximately 2.5 million interactions

3 Data Loading and Preprocessing

The preprocessing phase was crucial to transform the reduced dataset into a form suitable for recommendation experiments.

The listening logs were then aggregated by user and artist, resulting in a user–item matrix where each entry represented the number of times a given user had listened to a particular artist. This step converted the sequence of events into an implicit feedback signal, consistent with the paper's framework.

Since recommendation is essentially a prediction of future behavior, the data was split temporally. The earlier interactions were used as training data, while the most recent ones were set aside for testing. In particular, the test set corresponds to the last month of listening events recorded in the dataset, measured from the global timestamp of the final interaction. This setup reflects real-world conditions, where recommendations must be generated from past user activity and evaluated on future events.

Finally, repeated listens to the same artist that appeared both in the training and in the test set were removed from the test portion. The motivation was to avoid overestimating the models' accuracy through "easy" predictions of repeated consumption, and instead to focus on the discovery of new artists for each user. This design choice is consistent with the evaluation protocol described in the original paper, where re-watches of the same program were excluded to better capture the system's ability to recommend novel items.

4 Models

Three different recommendation models were implemented and compared. The first two act as simple baselines, while the third corresponds to the implicit feedback factorization proposed in the paper by Hu, Koren and Volinsky. In this way, we can appreciate the advantages of the more sophisticated approach against intuitive but limited methods.

Popularity

The first and simplest model is **Popularity**, which recommends the artists with the highest overall number of listens in the dataset. This method completely ignores personalization: all users receive the same list of recommendations. Although extremely simple, it provides a useful reference baseline to evaluate whether more complex algorithms are truly adding value.

Item–Item Collaborative Filtering

The second approach is **Item–Item Collaborative Filtering**, based on cosine similarity between artists. For each candidate artist, the model computes a score as a weighted sum of similarities with the artists already listened to by the user. This method is intuitive and explainable, since recommendations can be traced back to similar items in the user's history. However, its performance is limited in sparse datasets, where similarity values may be unreliable due to the lack of co-listening data.

Implicit ALS

We adopt the confidence-weighted matrix factorization of Hu et al. (2008). Let r_{ui} be the number of listening events of item i by user u in the training window. From r_{ui} we derive:

- **Preference**:
$$p_{ui} = \mathbb{I}[\, r_{ui} > 0 \,],$$
a binary signal indicating whether an interaction was observed.

- **Confidence** (two variants):
$$\underbrace{c_{ui} = 1 + \alpha\, r_{ui}}_{\text{linear}} \qquad \text{and} \qquad \underbrace{c_{ui} = 1 + \alpha\, \log\!\big(1 + \tfrac{r_{ui}}{c_0}\big)}_{\text{logarithmic (with data–driven scale } c_0)} \quad .$$

  For the logarithmic confidence, we set the scale parameter as
$$c_0 = \max\big(1.0,\ \text{mean}(\texttt{counts})\big),$$

where `counts` denotes the positive interaction counts $r_{ui} > 0$ aggregated over the training data. c_0 normalizes the magnitude of r_{ui} so that confidence grows smoothly and remains comparable across users and items.

Objective. Implicit ALS learns user factors $\{x_u\}$ and item factors $\{y_i\}$ by minimizing the weighted square loss over *all* user–item pairs:

$$\min_{\{x_u\},\{y_i\}} \sum_{u,i} c_{ui} \left(p_{ui} - x_u^\top y_i\right)^2 \;+\; \lambda \left(\sum_u \|x_u\|_2^2 + \sum_i \|y_i\|_2^2 \right),$$

with $\lambda > 0$ for regularization. The problem is solved by alternating least squares (ALS), iterating closed-form updates for user and item factors. By combining binary preferences with confidence weights, the model captures latent dimensions of user taste and item characteristics, generalizing beyond direct co-occurrence and producing more robust recommendations.

5 Evaluation Methodology

To evaluate the models, we use metrics suited to implicit feedback. Traditional error-based measures such as RMSE are not meaningful in this setting, since there are no explicit ratings to compare against and a zero denotes "unobserved," not a negative judgment. We therefore adopt ranking-based metrics that assess whether relevant items are placed near the top of the list, in line with the evaluation used in Hu et al. (2008). The main indicators were:

The main indicators were:

- **Recall@K**, which measures how many of the actually consumed items appear in the top-K recommendations.
- **MAP@K**, which also considers the position of relevant items, rewarding correct rankings earlier in the list.
- **Expected Percentile Ranking (EPR)**, which measures the average position of relevant items within the full recommendation list. Lower values indicate a better overall ranking.

This evaluation framework aligns with the paper and focuses on the quality of the recommendation ranking rather than predicting exact ratings.

6 Sensitivity Analysis for ALS models

We varied the confidence strength $\alpha \in \{10, 40, 100\}$ and the number of latent factors $n_{\text{factors}} \in \{16, 32\}$ for both linear and logarithmic confidence. Three clear trends emerged.

(i) Larger α consistently hurts performance. For ALS–linear, Recall@10 drops from 0.0151 (α=10, 32 factors) to 0.0019 (α=100, 16 factors), while EPR deteriorates from 16.20 (α=10, 16) to 21.57 (α=100, 32).
For ALS–log with 16 factors, EPR remains stable (15.33 $\rightarrow$ 15.45) but Recall@10 decreases (0.0182 $\rightarrow$ 0.0126) as α grows.

(ii) Increasing the number of factors from 16 to 32 does not help on this small, sparse dataset and typically worsens global ranking. For ALS–log at α=10, EPR degrades from 15.33 (16 factors) to 16.75 (32 factors) with no gain in Recall@10 (0.0182 $\rightarrow$ 0.0182). For ALS–linear at α=10, Recall@10 increases (0.0088 $\rightarrow$ 0.0151) but EPR worsens (16.20 $\rightarrow$ 17.55), indicating a trade-off toward overfitting and less consistent catalog-wide ordering.

(iii) Logarithmic scaling is more robust than linear. Across α, ALS–log with 16 factors keeps EPR around 15.3–15.5, whereas ALS–linear drifts up to 19–22.

Overall, the best trade-off is obtained with ALS–log at $\alpha = 10$ and $n_{\text{factors}} = 16$, suggesting that on this reduced, sparse split smaller α and a modest latent dimensionality are preferable; this contrasts with the original paper, which favors larger α and higher dimensionality on large, denser datasets, and further supports the role of log scaling in damping heavy-user repetitions and preserving stable rankings.

Table 1: ALS sensitivity to confidence strength α and latent dimensionality n_{factors}.

Confidence	$\boldsymbol{\alpha}$	**Factors**	**Recall@10**	**MAP@10**	**EPR**
Linear	10	16	0.0088	0.0072	16.20
Linear	10	32	0.0151	0.0211	17.55
Linear	40	16	0.0044	0.0050	17.73
Linear	40	32	0.0107	0.0082	19.61
Linear	100	16	0.0019	0.0008	19.77
Linear	100	32	0.0038	0.0033	21.57
Log	10	16	0.0182	0.0214	15.33
Log	10	32	0.0182	0.0218	16.75
Log	40	16	0.0157	0.0219	15.47
Log	40	32	0.0157	0.0161	17.05
Log	100	16	0.0126	0.0146	15.45
Log	100	32	0.0126	0.0153	17.47

7 Results

The table below summarizes the performance of the models on the reduced Last.fm dataset. For the linear and logarithmic ALS variants, the reported scores correspond to the best configuration found ($\alpha = 10$, $n_{\text{factors}} = 16$).

Model	**Recall@10**	**MAP@10**	**EPR (%)**
Popularity	0.0063	0.0056	17.48
Item–Item CF	0.0094	0.0128	15.68
ALS (linear conf.)	0.0088	0.0072	16.20
ALS (log conf.)	**0.0182**	**0.0214**	**15.33**

Table 2: Performance comparison across models.

On this reduced Last.fm split, absolute top-K values are low due to high sparsity, but relative gaps are clear. ALS with log-confidence achieves the best overall performance, yielding a more consistent catalog-wide ranking. Item–Item remains a strong baseline under sparsity, while ALS with linear confidence tends to overweight frequent interactions and therefore performs worse than both Item–Item and the ALS variant with logarithmic confidence.
Overall, the log-confidence variant offers the best trade-off between global ranking quality (lower EPR) and top-K accuracy (higher Recall/MAP) in this setting.

8 Strengths and Limitations

Strengths

- **ALS with log confidence** achieves the best trade-off between global ranking quality (low EPR) and top-K accuracy.
- Latent factors allow generalization beyond direct co-occurrence, while remaining computationally feasible with modest dimensionality.
- **Item–Item CF** remains a strong baseline: simple, explainable, and effective under high sparsity.

Limitations

- **Popularity bias**, with frequent artists dominating recommendations.
- **Dataset reduction and per-user cap**: necessary for computational feasibility but removing part of the long-tail behavior and reducing the richness of interactions.
- **Sensitivity to hyperparameters**: performance varies significantly with α and the number of factors; large values tend to overweight repeated interactions and worsen EPR, showing that careful tuning is required on sparse data.

9 Conclusion

This project reimplemented the implicit-feedback ALS model of Hu, Koren, and Volinsky and benchmarked it against simple baselines (Popularity and Item–Item) on a reduced Last.fm split (500 users, capped at 5000 events per user) with a temporal hold-out (train on earlier interactions; test on the last month). Using ranking metrics (Recall@10, MAP@10, EPR), we confirm that confidence-weighted matrix factorization is effective for implicit data: the logarithmic-confidence variant delivers the best overall trade-off, achieving lower EPR and stronger top-K accuracy. On this reduced split, the best setting is $\alpha = 10$ and $n_{\text{factors}} = 16$. At the same time, Item–Item remains competitive under high sparsity, underscoring the value of neighborhood methods on small datasets.

Taken together, these results illustrate *why* implicit ALS is strong in this setting: by separating binary preferences from confidence weights, the model can learn latent structure that goes beyond simple audience overlap and thereby improves catalog-wide ranking (lower EPR). The logarithmic confidence further mitigates the impact of heavy-user repetitions, which would otherwise dominate the loss and skew the factors toward the most frequent artists. At the same time, performance is not unconditional: on a reduced, sparse split, larger α and higher dimensionality tend to overemphasize repeated interactions and increase variance, which explains why smaller α and a modest number of factors work better here even though larger values can be beneficial on larger, denser datasets.

This also clarifies the role of strong baselines. Item–Item remains competitive because, under sparsity, local similarities derived from overlapping users are easier to estimate reliably than high-dimensional embeddings; comparing against it prevents us from

attributing to ALS gains that are actually due to popularity effects. The main limitations we observe follow directly from the nature of implicit data and from scale: popularity bias arises because many users interact with the same popular items, the reduced sample size limits the expressiveness of latent factors, and the models show **sensitivity to hyperparameters** (e.g., large α or excessive dimensionality degrade EPR), which calls for careful tuning on sparse data.

For these reasons, future work should *target the failure modes we observe*: popularity-aware reweighting or regularization (or different choice of c_0 and small α) to reduce the influence of heavy users; broader hyperparameter search; and experiments on larger, denser splits to test whether the beneficial regime for larger α and dimensionality observed in the original paper reappears when more signal is available.

References

- Hu, Y., Koren, Y., & Volinsky, C. (2008). *Collaborative Filtering for Implicit Feedback Datasets.* IEEE ICDM.
- Leskovec, J., Rajaraman, A., & Ullman, J. (2014). *Mining of Massive Datasets.*
- Course slides: *Recommender Systems* (Damiano Carra, Univ. Verona).
- Last.fm 1K dataset documentation.