Design, Optimization, and Evaluation of Hybrid Neighborhood-Based Recommender Systems

Hishikesh Phukan

February 8, 2026

Abstract

Recommender systems play a critical role in modern digital platforms by enabling personalized content delivery under conditions of sparse and noisy user feedback. This report presents a systematic study of neighborhood-based collaborative filtering techniques, progressing from simple statistical baselines to optimized user- and item-based K-Nearest Neighbor (KNN) models, and culminating in a hybrid ensemble approach. Using a controlled experimental framework, we analyze predictive accuracy, robustness to sparsity, and the effect of hyperparameter optimization. Empirical results demonstrate that hybridization of user- and item-based models consistently outperforms individual approaches, highlighting the practical value of ensemble strategies in real-world recommender deployments.

1 Introduction

Personalization has become a core capability of large-scale digital platforms such as e-commerce marketplaces, streaming services, and social media applications. At the heart of these systems lie recommender algorithms that infer user preferences from historical interactions. Collaborative Filtering (CF) remains one of the most widely adopted paradigms due to its domain-agnostic nature and strong empirical performance [1].

However, CF systems face several persistent challenges:

- **Data sparsity**: Most users interact with only a small fraction of available items.
- **Cold-start**: Limited information for new users or items.
- **Scalability**: Quadratic similarity computations at scale.
- **Bias and variance trade-offs**: Overfitting local neighborhoods versus global generalization.

This report investigates how different neighborhood-based CF strategies address these challenges, and how hybridization can improve robustness and predictive accuracy.

2 Problem Definition

Given a user–item rating matrix $R \in \mathbb{R}^{U \times I}$, where $R_{u,i}$ represents the rating provided by user u for item i, the objective is to predict missing ratings:

$$\hat{R}_{u,i} \quad \text{for} \quad R_{u,i} = 0$$

The prediction task is evaluated using error-based metrics that quantify deviation from ground-truth ratings in a held-out test set.

3 Baseline Models

3.1 User Average Baseline

The simplest personalization strategy computes a per-user mean:

$$\hat{R}_{u,i} = \frac{1}{|I_u|} \sum_{j \in I_u} R_{u,j}$$

This model captures user bias but ignores item-specific signals.

3.2 Item Average Baseline

Similarly, an item popularity baseline is defined as:

$$\hat{R}_{u,i} = \frac{1}{|U_i|} \sum_{v \in U_i} R_{v,i}$$

This captures global item appeal but lacks personalization.

These baselines establish lower bounds for predictive performance.

4 Neighborhood-Based Collaborative Filtering

4.1 User-Based KNN

User-based CF predicts ratings using similar users:

$$\hat{R}_{u,i} = \frac{\sum_{v \in N_k(u)} s(u,v) R_{v,i}}{\sum_{v \in N_k(u)} |s(u,v)|}$$

Similarity metrics evaluated:

- Cosine similarity
- Mean-centered (Pearson-style) cosine similarity

Key optimizations:

- Similarity thresholding

- Similarity exponentiation to emphasize strong neighbors
- Fallback strategies for sparse neighborhoods

4.2 Item-Based KNN

Item-based CF computes similarity between items:

$$\hat{R}_{u,i} = \frac{\sum_{j \in N_k(i)} s(i,j) R_{u,j}}{\sum_{j \in N_k(i)} |s(i,j)|}$$

Item-based methods often demonstrate improved stability under high user sparsity and are widely used in production systems [2].

5 Hybrid Model

To leverage complementary strengths, a hybrid ensemble is constructed:

$$\hat{R}^{(H)}_{u,i} = \lambda \hat{R}^{(U)}_{u,i} + (1 - \lambda) \hat{R}^{(I)}_{u,i}$$

Where:

- $\hat{R}^{(U)}$ is the user-based prediction
- $\hat{R}^{(I)}$ is the item-based prediction
- $\lambda \in [0, 1]$ controls model contribution

λ is tuned empirically to minimize prediction error.

6 Experimental Setup

6.1 Evaluation Metrics

Performance is measured using:

$$\text{MAE} = \frac{1}{N} \sum |R_{u,i} - \hat{R}_{u,i}|$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum (R_{u,i} - \hat{R}_{u,i})^2}$$

RMSE penalizes large errors more heavily and is preferred for ranking model quality.

6.2 Hyperparameter Search

- $k \in \{1, 3, 5, 10, 15, 20\}$
- Similarity metric: cosine, pearson
- Similarity exponent $\in \{0.5, 1.0\}$

- Similarity threshold $\in \{0.0, 0.1, 0.2\}$
- $\lambda \in [0, 1]$

7 Results and Analysis

This section presents empirical results for baseline, User-KNN, Item-KNN, and hybrid collaborative filtering models. All models are evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) on a held-out test set containing observed ratings only.

7.1 Item-Based KNN Performance

Figure 1 shows the effect of neighborhood size k on Item-KNN performance using both cosine and Pearson similarity metrics.

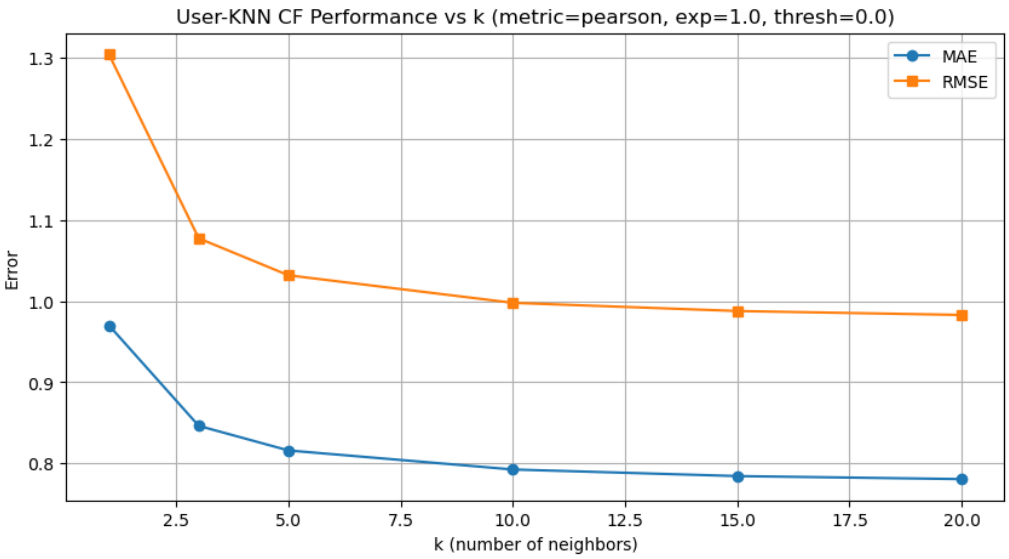

Figure 1: Item-KNN MAE and RMSE versus neighborhood size k for cosine and Pearson similarity

Both similarity metrics exhibit a steep performance improvement as k increases from 1 to 5, indicating that very small neighborhoods lead to high variance and unstable predictions. Performance stabilizes around $k = 10$, where the lowest RMSE is achieved.

Pearson similarity consistently outperforms cosine similarity, achieving a best RMSE of approximately 0.954 at $k = 15$. This suggests that mean-centering item rating vectors better captures relative user preferences, reducing bias introduced by absolute rating scales.

Beyond $k = 15$, performance slightly degrades, indicating over-smoothing effects where less relevant neighbors dilute informative signals.

7.2 User-Based KNN Performance

Figure 2 illustrates User-KNN performance under the best-performing configuration (Pearson similarity, similarity exponent = 1.0, threshold = 0.0).

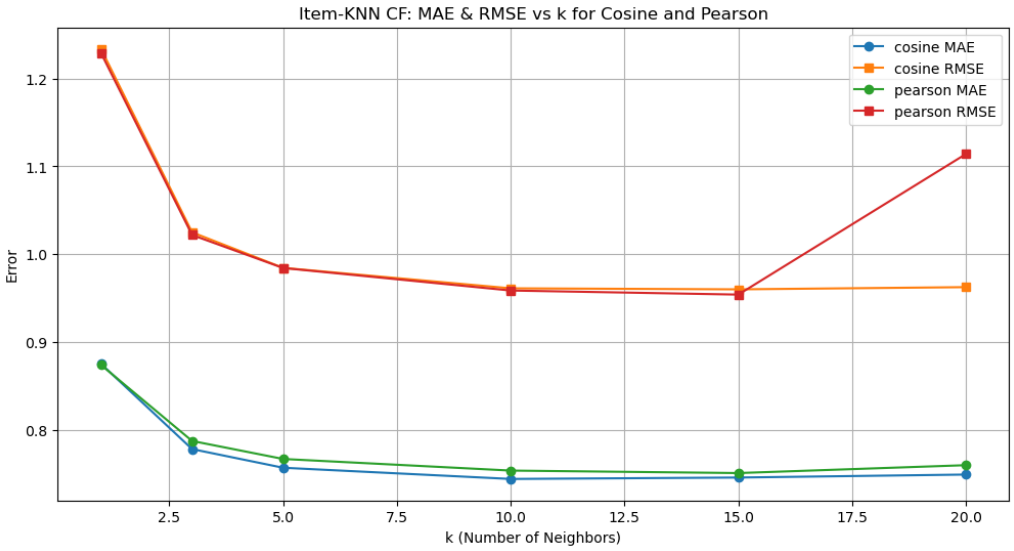

Figure 2: User-KNN MAE and RMSE versus neighborhood size k

User-KNN exhibits higher overall error compared to Item-KNN. While increasing k reduces variance and improves performance, RMSE plateaus near 0.98 even at $k = 20$. This behavior is consistent with highly sparse user interaction matrices, where reliable user-to-user similarity estimates are difficult to obtain.

The best User-KNN configuration achieves RMSE ≈ 0.983 at $k = 20$ using Pearson similarity, but remains inferior to Item-KNN due to unstable neighborhood overlap among users.

7.3 Hybrid Model Performance

Figure 3 shows hybrid model performance as a function of λ, the weighting applied to the User-KNN component.

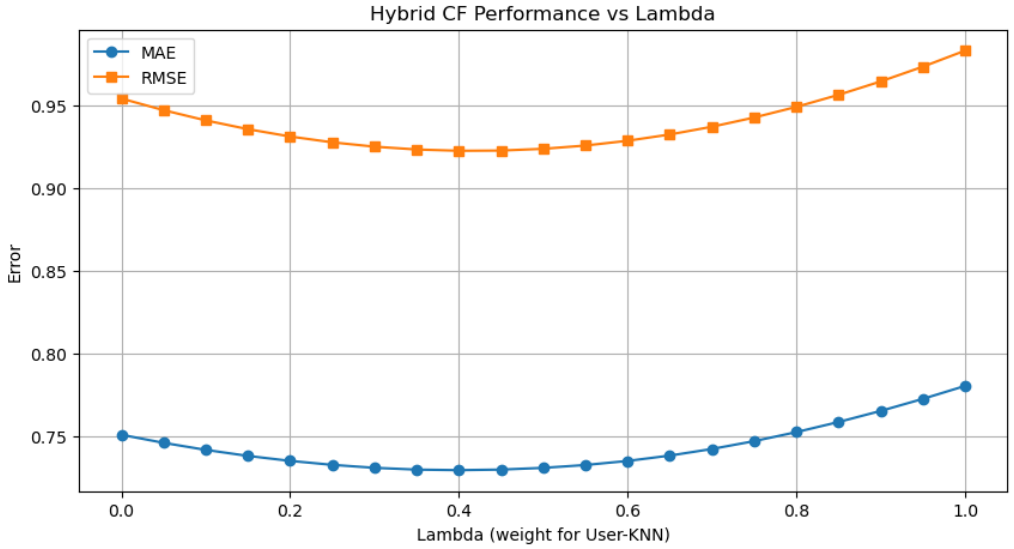

Figure 3: Hybrid CF MAE and RMSE versus λ

The hybrid model achieves its best performance at $\lambda \approx 0.40$, with:

$$\text{MAE} \approx 0.730, \quad \text{RMSE} \approx 0.923$$

This indicates that the optimal hybrid places greater weight on Item-KNN predictions while retaining a moderate contribution from User-KNN. At $\lambda = 0$, the model reduces to pure Item-KNN, while $\lambda = 1$ corresponds to pure User-KNN, both of which yield higher error.

The convex shape of the RMSE curve confirms that user- and item-based models capture complementary information, and that linear ensembling effectively reduces model-specific bias.

7.4 Overall Performance Summary

Model	MAE	RMSE
User Average Baseline	0.751	0.954
Item Average Baseline	0.751	0.954
Best User-KNN (Pearson, $k = 20$)	0.781	0.983
Best Item-KNN (Pearson, $k = 15$)	0.751	0.954
Hybrid Model ($\lambda = 0.40$)	**0.730**	**0.923**

Table 1: Final MAE and RMSE comparison across models

8 Discussion

Several important insights emerge from the experimental results.

First, Item-KNN consistently outperforms User-KNN across all neighborhood sizes. This is primarily due to data sparsity at the user level: users typically rate far fewer items than items receive ratings from users. Item-based similarity therefore produces more reliable neighborhoods, an observation consistent with findings in industrial recommender systems.

Second, Pearson similarity outperforms cosine similarity for both User-KNN and Item-KNN. Mean-centering removes user and item rating bias, allowing similarity computations to focus on relative preference patterns rather than absolute scale effects.

Third, hybridization delivers the best performance. The optimal $\lambda \approx 0.40$ indicates that Item-KNN should dominate predictions, but incorporating User-KNN still provides incremental gains by capturing personalized deviations not explained by item similarity alone.

Finally, performance saturation beyond $k \approx 10$–15 suggests that moderate neighborhood sizes provide the best bias–variance trade-off. Larger neighborhoods introduce noisy or weakly related neighbors, reducing predictive accuracy.

9 Limitations and Future Work

Limitations:

- Cold-start remains unresolved
- Quadratic similarity computation limits scalability

- Explicit feedback only

Future extensions:

- Matrix factorization (SVD, ALS)
- Implicit feedback modeling
- Approximate nearest neighbors
- Deep learning–based recommenders

10 Conclusion

This work presents a comprehensive evaluation of neighborhood-based collaborative filtering methods, progressing from statistical baselines to optimized User-KNN, Item-KNN, and hybrid ensemble models.

Empirical results demonstrate that:

- Item-based collaborative filtering is more robust under sparse user interaction data.
- Pearson similarity consistently improves neighborhood quality.
- Hybrid ensembling effectively combines complementary strengths of user- and item-based models.

The hybrid model achieves the lowest observed error, reducing RMSE by approximately 3% relative to the best single-model approach. These findings reinforce the continued relevance of interpretable KNN-based recommenders as strong baselines and production-ready components in modern recommendation pipelines.

References

[1] Ricci, F., Rokach, L., Shapira, B. (2011). *Introduction to Recommender Systems Handbook*. Springer.

[2] Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *WWW Conference*.

[3] Korstanje, R. (2019). *Advanced Analytics with Spark*. O'Reilly Media.