Clustering and Association Rule Mining Signal Matrix Factorization (CARMS-MF) Recommendation System

Peerapat Tancharoen

Artificial Intelligence for Business Analytics, School of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520, Thailand.

Contributing authors: peerapat.tcr@gmail.com;

Abstract

Recommender systems are essential to digital platforms for delivering personalized content, yet their effectiveness is often limited by severe data sparsity in user–item interactions. Although matrix factorization models are widely adopted, their performance degrades when interaction data are sparse, and many existing hybrid approaches depend on side information that is not always available.

This paper proposes Clustering and Association Rule Mining Signal Matrix Factorization (CARMS-MF), a feature-free hybrid recommendation framework that enhances biased matrix factorization using only the user–item interaction matrix. The proposed method clusters users based on interaction patterns and applies association rule mining to extract meaningful relationships between items-to-item and cluster-to-item. These relationships are transformed into a next-item signal matrix and integrated into a ranking-oriented learning process to improve Top-K recommendation quality.

Experiments on five different categories of the Amazon 5-ratings benchmark datasets demonstrate that CARMS-MF consistently outperforms baseline matrix factorization models across different ranking thresholds. These findings suggest that our proposed method effectively addresses data sparsity and improves recommendation quality in real-world, sparse environments.

Keywords: Recommender systems, Matrix factorization, Bayesian Personalized Ranking, Clustering, Association rule mining, Data sparsity

1 Introduction

Recommender systems are widely used on modern digital platforms to help users discover relevant products and content in the presence of information overload. By leveraging historical user behavior, these systems enable personalized delivery that can improve user experience and support business objectives such as engagement and retention [**?**].

Collaborative filtering, particularly biased matrix factorization, learns latent user and item representations directly from interaction data and has shown strong performance [**?**]. However, these models are highly sensitive to interaction sparsity: when only a small fraction of user–item pairs are observed, learning becomes unreliable and ranking quality degrades [**?**].

To address the sparsity problem, existing research has proposed various hybrid recommender systems that incorporate auxiliary information such as user profiles, item attributes, or contextual features into collaborative filtering models (find others reference). While these approaches often improve recommendation accuracy, they rely on the availability and reliability of side information, which is not always available.

Recent studies leverage interaction data to address sparsity by extracting group-level preference signals via clustering [**?**] and item-to-item relations via co-occurrence [**?**] or association rules [**?**]. While these signals can improve ranking without side information, existing approaches either treat clusters mainly as contextual features with only small gains [**?**], depend partly on side information [**?**], or use co-occurrence only to refine latent factors [**?**]. Based on this review, we extend prior interaction-driven approaches by combining user clustering with association-rule indicators to denoise interaction signals, and by incorporating these signals into a modified learning objective with a re-ranking mechanism to improve Top-K recommendation quality.

In this paper, we propose **Clustering and Association Rule Mining Signal Matrix Factorization (CARMS-MF)**, a feature-free hybrid recommendation framework that enhances biased matrix factorization by integrating additional signals derived solely from the user–item interaction matrix. The proposed approach combines user clustering to model group-level preferences with association rule mining to capture item-level and cluster-level relationships. These signals are incorporated into a ranking-oriented learning process designed to improve Top-K recommendation performance under sparse interaction settings.

The main contributions of this paper are summarized as follows:

- We propose a novel feature-free hybrid recommendation framework that enhances biased matrix factorization using clustering and association rule mining signals extracted directly from interaction data, together with a BPR-based pairwise ranking objective.
- We introduce a next-item signal (CARMS) that jointly captures cluster-to-item and item-to-item preference patterns without relying on side information.
- We conduct comprehensive experiments on multiple public benchmark datasets and show consistent improvements in Top-K ranking performance compared with the biased matrix factorization (Bias MF) baseline across different datasets and cutoff values.

2 Related Work

Collaborative filtering is one of the most widely studied approaches in recommender systems due to its ability to generate personalized recommendations using only historical user–item interactions. Among collaborative filtering methods, matrix factorization has become a dominant technique by modeling users and items in a shared latent space and estimating preferences through their interactions. Biased variants of matrix factorization [**?**] further improve predictive performance by explicitly modeling global, user, and item biases, and are commonly adopted as strong baselines in recommender system research. Despite their effectiveness, matrix factorization–based models are known to suffer from severe performance degradation when interaction data are highly sparse. [**?**]

To solve sparsity, a large body of work has proposed hybrid recommender systems that integrate auxiliary information such as user attributes, item features, or contextual signals into collaborative filtering models [**?**]. While these approaches can significantly improve recommendation accuracy, they rely on the availability and quality of side information, which is not always accessible in practice. Consequently, feature-free hybrid approaches that extract additional signals directly from interaction data have attracted increasing research attention.

One line of research focuses on incorporating clustering techniques into recommendation models. By grouping users or items based on interaction patterns, clustering-based approaches capture group-level preference structures that complement individual-level modeling. For example, cluster-based matrix factorization models leverage user or item clusters to stabilize learning under sparse conditions and improve prediction accuracy [**?**]. However, such approaches often rely primarily on cluster membership information and may overlook fine-grained relationships among items.

Another related direction explores item co-occurrence and association-based methods. These approaches identify relationships between items that frequently appear together in user interaction histories and use them to enhance recommendation performance. Co-occurrence–based matrix factorization models have shown effectiveness in improving recommendations for rare items by incorporating item–item relationships derived from interaction data [**?**]. Association rule mining has also been applied to recommender systems to exploit both positive and negative user preferences, demonstrating that association rules can provide useful complementary information beyond traditional collaborative filtering [**?**].

The proposed CARMS-MF framework clusters users based on interaction patterns and applies association rule mining to extract meaningful item-to-item and cluster-to-item relationships directly from sparse interaction data. These relationships are transformed into a next-item signal matrix, which is then incorporated into a ranking-oriented learning process to improve Top-K recommendation quality under sparse settings.

3 Purposed Method

This section introduces the proposed **Clustering and Association Rule Mining Signal Matrix Factorization (CARMS-MF)** framework, which improves ranking-oriented matrix factorization by incorporating next-item signals extracted solely from the user–item interaction matrix. We describe next-item signal construction via association rule mining, CARMS signal generation, and the final objective that combines biased matrix factorization with a BPR-based ranking loss.

3.1 Clustering and Association Rule Mining Score

Before applying association rule mining, users are first clustered based on their interaction patterns. Let $c_u \in \{0, 1, \ldots, k-1\}$ denote the cluster assignment of user u. This clustering step is used to modify the transaction representation by augmenting each user transaction with cluster-level information, enabling the mining process to capture both item–item and cluster–item association patterns from interaction data.

To filter noisy or unreliable rules, we define a masking term based on the minimum support and confidence thresholds:

$$M_{ab} = \mathbb{I}\big[\mathrm{sup}(a \to b) \geq \sigma \ \wedge \ \mathrm{conf}(a \to b) \geq \kappa\big], \tag{1}$$

where $\mathbb{I}[\cdot]$ is an indicator function, σ and κ are hyperparameters controlling the minimum support and confidence, respectively.

$$W_{ab} = \mathrm{conf}(a \to b) \times \mathrm{lift}(a \to b) \times M_{ab}, \tag{2}$$

where $\mathrm{conf}(a \to b)$ denotes the confidence of the rule, $\mathrm{lift}(a \to b)$ represents the lift (used as the rule importance factor), and M_{ab} is the masking term in Eq. (1) that filters invalid or noisy associations. Here, the rule $a \to b$ can represent either an item-to-item association (item $\to$ item) or a cluster-to-item association (cluster $\to$ item).

3.2 CARMS Signal Generation

Given the clustering and association rule mining score matrix $\mathbf{W}$, a personalized signal vector is generated for each user. When a user u has observed interactions represented by the interaction vector $\mathbf{x}_u$, the corresponding CARMS signal is computed as

$$\mathbf{s}_u = \mathbf{x}_u \mathbf{W} \in \mathbb{R}^{I_a}, \tag{3}$$

where I_a denotes the number of active items. This formulation aggregates association rules contributions from items previously interacted with by the user.

For users with limited or no interaction history within certain clusters, the CARMS signal can alternatively be constructed using cluster-level information:

$$\mathbf{s}_u = \mathbf{W}_{(I_a+c)_u} \in \mathbb{R}^{I_a}, \tag{4}$$

where c_u denotes the cluster assignment of user u. This formulation enables the model to generate meaningful signals even under extreme sparsity by leveraging group-level preference patterns.

3.3 Co-objective Matrix Factorization with BPR

Following the co-objective learning idea in [**?**], the CARMS signal is integrated into a unified objective that combines a pointwise matrix factorization loss and a pairwise ranking loss. The pointwise component is based on biased matrix factorization, which predicts the interaction score between user u and item i as

$$\hat{y}_{ui} = \mu + b_u + b_i + \mathbf{p}_u^\top \mathbf{q}_i, \tag{5}$$

where μ denotes the global bias, b_u and b_i are user and item bias terms, and $\mathbf{p}_u, \mathbf{q}_i \in \mathbb{R}^K$ represent latent factor vectors.

The pointwise term reconstructs observed interaction scores (a regression-style MF objective), while CARMS provides confidence signals that some unobserved items should be ranked higher via association-derived next-item cues. To optimize Top-K ranking, we add Bayesian Personalized Ranking (BPR) as a second objective. The combined loss function is defined as

$$\mathcal{L}_{\mathrm{co}} = \sum_{y_{ui}>0} \left(y_{ui} - \hat{y}_{ui}\right)^2 + \gamma \sum_{y_{ui}>0} \sum_{y_{uj}=0} \left[-w_{ui} \log \sigma \left(\hat{y}_{ui} - \hat{y}_{uj}\right)\right] + \lambda \left(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2\right), \tag{6}$$

where the first term corresponds to squared reconstruction error, the second term is a weighted BPR loss guided by CARMS confidence signals, and the final term is an ℓ_2 regularization applied to the latent factor matrices.

3.4 Positive and Negative Item Definition

To construct meaningful pairwise ranking constraints, candidate items are divided into positive and negative sets based on the CARMS signal. For a user u, the positive item set is defined as

$$i \in \mathcal{P}_u = \{i \mid y_{ui} = 0,\ s_{ui} > 0\}, \tag{7}$$

which includes unobserved items receiving positive support from CARMS. Conversely, the negative item set is defined as

$$j \in \mathcal{N}_u = \{j \mid y_{uj} = 0,\ s_{uj} = 0\}, \tag{8}$$

representing unobserved items without CARMS support.

3.5 Solver

The optimization problem in Eq. (8) is solved using mini-batch stochastic gradient descent (SGD). At each iteration, parameters are updated using gradients computed from a sampled mini-batch:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}, \tag{9}$$

where η denotes the learning rate.

4 Experimental Setup

This section describes the experimental settings used to evaluate the proposed CARMS-MF framework. We introduce the datasets, data splitting protocol, baseline model, evaluation metrics, and hyperparameter search strategy to ensure fair comparison and reproducibility.

4.1 Datasets

Experiments are conducted on five public benchmark datasets from the Amazon product review corpus, covering diverse product categories and sparsity levels. Each dataset consists of user-item interaction records after filtering users and items with fewer than five interactions. Table 1 summarizes the dataset statistics.

Table 1 Statistics of the Amazon benchmark datasets.

Dataset	#Users	#Items	Sparsity (%)
Luxury Beauty	3,819	1,581	0.35
Industry	11,041	5,334	0.01
Pantry	14,180	4,970	0.02
Music	16,566	11,797	0.07
Instruments	27,530	10,620	0.07

All datasets are highly sparse, reflecting realistic recommendation scenarios where only a small fraction of possible user–item interactions are observed. Such characteristics make these datasets suitable for evaluating the effectiveness of sparsity-aware recommendation models.

4.2 Data Splitting Protocol

To evaluate next-item recommendation performance, we adopt a leave-one-out (LOO) splitting strategy. For each user, the most recent interaction is held out as the test instance, while the remaining interactions are used for training. This protocol preserves the temporal ordering of interactions and prevents information leakage, and is widely used in Top-K ranking evaluation.

4.3 Baseline Model

We compare the proposed CARMS-MF framework with a single collaborative filtering baseline, **Biased Matrix Factorization (Bias MF)**. Bias MF predicts user–item preference using latent factors together with global, user, and item bias terms, and serves as a standard benchmark that relies solely on interaction data.

4.4 Evaluation Metrics

Recommendation performance is evaluated using ranking-based metrics that directly reflect Top-K recommendation quality:

- Normalized Discounted Cumulative Gain (NDCG@K)
- Hit Rate (HR@K)

We report results for multiple cutoff values $K \in \{5, 10, 20, 50, 100\}$ to assess performance across different recommendation list sizes. We adopt ranking metrics rather than regression metrics (e.g., RMSE) because the goal of a recommender system is to correctly rank a small set of relevant items at the top of the recommendation list, not to precisely predict absolute interaction scores. In implicit or sparse settings, unobserved entries are not reliable negatives, making error-based measures sensitive to the choice of missing-value treatment, whereas NDCG@K and HR@K directly evaluate the quality of the ranked lists that users actually see.

4.5 Hyperparameter Search Strategy

Model hyperparameters are selected using randomized search to efficiently explore the parameter space. For each dataset, we perform 200 randomized search trials on the training data and select the configuration that yields the best validation performance. The search space for each hyperparameter is summarized in Table 2.

Table 2 Hyperparameter search space used in randomized search.

Hyperparameter	Search Space
Latent dimension	{20, 30, 50, 70, 100}
Learning rate	{0.001, 0.005, 0.01, 0.05, 0.1}
Regularization (λ)	{0.001, 0.005, 0.01, 0.05, 0.1}
Epochs	{20, 30, 50, 70, 100}
User clusters	{1, 3, 5, 7, 10}
Minimum support	{0.0, 0.00001, 0.0001, 0.001, 0.01}
Minimum confidence	{0.0, 0.0001, 0.001, 0.01, 0.1}
Signal weight (γ)	{0.0, 0.1, 1, 10, 100}

5 Results

This section presents the experimental results of the proposed CARMS MF (CARMS bias MF) and compares its performance with the baseline Matrix Factorization (Bias MF). Ranking performance is evaluated using NDCG@K and HR@K, followed by a discussion of dataset-level behavior and selected hyperparameter configurations.

5.1 Overall Ranking Performance

Table 3 reports the Top-K ranking performance of Bias MF and CARMS Bias MF across five Amazon benchmark datasets. Results are evaluated using NDCG@K and HR@K with cutoff values $K \in \{5, 10, 20, 50, 100\}$.

Across all datasets and evaluation metrics, CARMS Bias MF consistently outperforms the baseline Bias MF. The improvements are observed for both ranking quality (NDCG@K) and recommendation coverage (HR@K), indicating that the proposed method improves not only the top-ranked items but also the overall ordering of the recommendation list.

5.2 Dataset-Level Analysis

The magnitude of improvement varies across datasets. On Amazon Pantry, CARMS Bias MF achieves roughly **3**× higher Top-K ranking performance than the Bias MF baseline. This large gain is consistent with the supermarket-like nature of the dataset, where association rule mining effectively captures complementary items frequently purchased together, producing strong next-item signals for sparse recommendation.

5.3 Hyperparameter Analysis

Table **??** reports the optimal hyperparameter configurations obtained via randomized search with 200 iterations, using NDCG@10 as the target metric.

Number of user clusters: The search space includes $K = 1$ to test whether clustering is necessary. For four out of five datasets, the best models select $K > 7$, suggesting heterogeneous user behaviors and that cluster-level rules contribute to accuracy. Amazon Music is an exception where the best configuration uses $K = 1$, indicating limited benefit from clustering.

Minimum support and confidence: We also allow $min\_support = 0$ and $min\_confidence = 0$ to test using unfiltered co-occurrence. None of the best configurations select zero thresholds, implying that filtering rare/noisy rules improves CARMS signal quality.

Gamma (γ): The search range includes $\gamma = 0$ (no BPR) up to $\gamma = 100$. All datasets select $\gamma = 100$, highlighting the importance of the BPR term and confirming that CARMS signals are most effective when optimized with a ranking-oriented objective.

5.4 Computation Time Analysis

Model complexity is a key limitation of CARMS-MF. Due to the additional CARMS signal construction and the co-objective BPR optimization, CARMS-MF requires on average 5.64× longer training time than the Bias MF baseline (under each model's best hyperparameter setting), trading higher computational cost for improved recommendation accuracy.

Table 3 Overall ranking performance comparison between Bias MF and CARMS Bias MF.

		NDCG@K					HR@K				
Dataset	Model	@5	@10	@20	@50	@100	@5	@10	@20	@50	@100
Amazon Luxury	Bias MF	17.8	18.2	18.6	19.3	19.8	19.6	21.1	22.7	26.1	29.5
	CARMS Bias MF	25.7	26.7	27.3	28.2	28.9	29.5	32.4	35.1	39.3	43.9
Amazon Industry	Bias MF	4.4	4.6	4.9	5.3	5.6	5.2	6.1	7.0	9.0	10.9
	CARMS Bias MF	5.6	6.3	7.0	7.8	8.6	7.0	9.0	11.8	16.1	20.8
Amazon Pantry	Bias MF	0.5	0.7	0.9	1.3	1.8	0.8	1.4	2.3	4.4	7.0
	CARMS Bias MF	1.7	2.1	2.7	3.6	4.5	2.6	4.0	6.2	11.0	16.4
Amazon Music	Bias MF	1.9	2.1	2.4	2.8	3.2	2.7	3.4	4.6	6.5	8.8
	CARMS Bias MF	3.0	3.6	4.3	5.3	6.1	4.4	6.3	9.1	14.0	19.2
Amazon Instrument	Bias MF	3.9	4.2	4.4	4.7	4.9	4.6	5.5	6.3	7.8	9.3
	CARMS Bias MF	6.4	7.0	7.6	8.3	8.8	8.0	10.0	12.3	15.7	19.0

Table 4 Best hyperparameter selected by randomized search (200 iterations) using NDCG@10 as the target.

Dataset	Model	Latent	LR	λ	Epoch	K	Support	Confidence	γ
Amazon Luxury	Bias MF	100	0.001	0.1	20	–	–	–	–
	CARMS Bias MF	100	0.01	0.001	50	10	0.001	0.1	100
Amazon Industry	Bias MF	100	0.001	0.05	20	–	–	–	–
	CARMS Bias MF	30	0.001	0.005	100	7	0.0001	0.1	100
Amazon Pantry	Bias MF	20	0.001	0.05	30	–	–	–	–
	CARMS Bias MF	100	0.01	0.005	70	7	0.0001	0.1	100
Amazon Music	Bias MF	100	0.001	0.001	30	–	–	–	–
	CARMS Bias MF	100	0.01	0.01	50	1	0.0001	0.1	100
Amazon Instrument	Bias MF	100	0.001	0.005	20	–	–	–	–
	CARMS Bias MF	100	0.01	0.001	100	7	0.0001	0.1	100

6 Limitations

Although the proposed CARMS-MF model achieves higher ranking accuracy than the baseline model, several limitations should be acknowledged.

Model complexity and hyperparameter explosion: Compared with the baseline Bias MF model, which uses four hyperparameters, CARMS-MF introduces four additional hyperparameters related to clustering and association rule mining, resulting in a total of eight hyperparameters. As described in Section 3, each hyperparameter is selected from five candidate values. Consequently, the total number of possible hyperparameter combinations increases substantially—from $5^4 = 625$ combinations for Bias MF to $5^8 = 390{,}625$ combinations for CARMS-MF. Given this enlarged search space, the randomized search strategy with 200 trials may not sufficiently explore all promising configurations for CARMS-MF, and the reported results may not correspond to the true global optimum.

Computational cost: CARMS-MF requires significantly more computation time than the baseline model. Based on the best-performing hyperparameter settings, the average training time of CARMS-MF is approximately five times higher than that of Bias MF. This increase is primarily due to the additional CARMS signal construction process, as well as the more complex objective function that combines the original SSE loss with an additional BPR-based ranking loss. Although these computations can be performed offline, the higher computational burden may limit scalability in large-scale or time-sensitive applications.

7 Conclusion

This thesis begins by addressing a fundamental challenge in modern recommendation systems, namely data sparsity. In sparse user–item interaction settings, baseline models such as Bias Matrix Factorization struggle to learn reliable latent representations, as their learning mechanisms rely solely on observed rating data. This limitation directly degrades recommendation accuracy and represents a critical research gap in existing recommender system literature.

To address this gap, this thesis proposes a novel matrix factorization framework, namely Clustering Association Rule Mining Signal Matrix Factorization (CARMS-MF). The objective of the proposed method is to extract next-item signals by integrating user clustering and association rule mining, without relying on any additional side information or item/user features. The underlying hypothesis is that incorporating structured signals derived from interaction patterns alone can significantly improve recommendation accuracy compared to conventional baseline models.

A review of related work indicates that only a limited number of studies have attempted to integrate association rules or item co-occurrence information into matrix factorization. Prior approaches either incorporate association rule mining as a separate module while still relying on content-based features, or jointly learn co-occurrence signals alongside user ratings. In contrast, this thesis extends these ideas by introducing user clustering to construct group-level priors and by using association rule indicators to filter noise from sparse interaction data.

The proposed framework augments user transactions through clustering, enabling the formation of user and item groups without additional feature information. Association rules are then mined from the augmented transactions, and the extracted signals are incorporated into the learning process through a co-objective optimization framework that enhances the baseline matrix factorization objective.

Experimental results on multiple Amazon 5-ratings benchmark datasets demonstrate that the proposed CARMS-MF model consistently outperforms baseline matrix factorization models across both NDCG@K and HR@K metrics, for all recommendation list lengths. Notably, the performance gains are most pronounced on highly sparse datasets, such as Amazon Industry and Amazon Pantry, which directly supports the initial hypothesis of this study. These findings confirm that the proposed method is effective for improving recommendation performance in real-world, sparse environments.