

# Research on Music Neighborhood-Based Recommendation Algorithms

**Jinxu Li**

School of Information Engineering, Shanghai Maritime University, Shanghai, China

202210320020@stu.shmtu.edu.cn

**Abstract.** This paper presents a comprehensive review of recommendation algorithms in the music domain. Music recommendation systems play a crucial role in delivering personalized content to users by analyzing patterns and preferences. We analyze several prominent recommendation algorithms, including collaborative filtering, content-based filtering, and hybrid models, each of which has been extensively applied to music recommendation tasks. The study also delves into the emerging techniques of deep learning and reinforcement learning in improving recommendation accuracy. A comparative analysis is provided, highlighting the strengths and limitations of each algorithm in terms of scalability, accuracy, and computational efficiency. Special attention is given to the challenges of balancing personalization and diversity in music recommendations. Furthermore, we explore how hybrid recommendation systems, which combine multiple methods, can overcome the limitations of individual models. The review concludes by identifying future research directions in the field of music recommendation systems, particularly the potential of leveraging more advanced machine learning techniques.

**Keywords:** Music recommendation, collaborative filtering, content-based recommendation, hybrid recommendation, deep learning.

## 1. Introduction

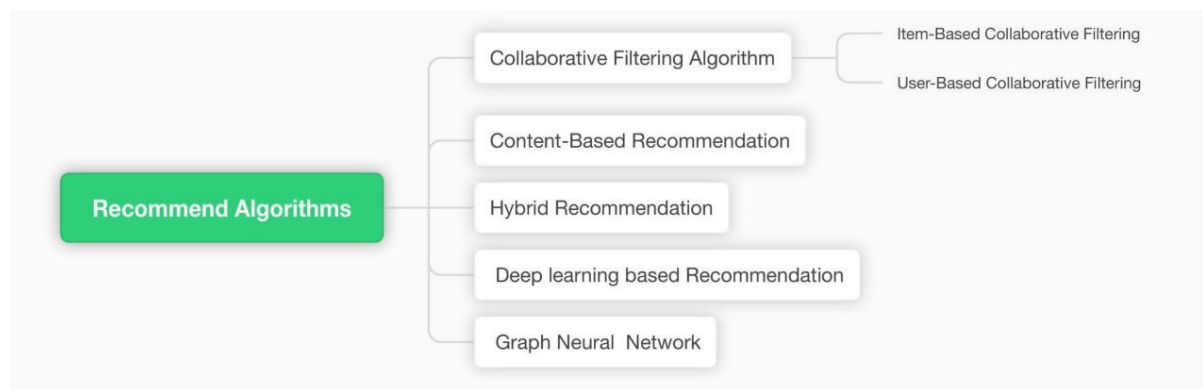
With the rapid development of music streaming platforms, users face a plethora of music content daily, making it challenging to discover music they might enjoy. One of the core functions of platforms such as Spotify, Apple Music, and NetEase Cloud Music is to provide personalized music recommendations through recommendation systems, thereby enhancing user experience and platform engagement.

For instance, Spotify's recommendation system is highly developed; it combines collaborative filtering, content-based recommendations, and deep learning algorithms to suggest new playlists and tracks for users. By analyzing users' listening history, saved playlists, and interactions with other users, Spotify can accurately recommend songs that users are likely to be interested in. Additionally, NetEase Cloud Music's "Daily Recommendation" feature analyzes users' listening history and community interactions to provide personalized recommendation lists.

In the field of music recommendation, the choice of recommendation algorithm is crucial for recommendation effectiveness [1,2,3]. Common recommendation algorithms include collaborative filtering, content-based recommendation, hybrid recommendation, and deep learning-based recommendation. Each algorithm has its advantages and limitations in practical applications. For example, while collaborative filtering algorithms perform well on platforms like Spotify, they still face

challenges related to data sparsity, particularly when dealing with new users or new music. Literature indicates that data sparsity affects the accuracy of recommendations and leads to insufficient diversity. To address these issues, NetEase Cloud Music's hybrid recommendation system integrates collaborative filtering and content-based recommendation methods, achieving better recommendation outcomes. However, the effectiveness and limitations of this hybrid strategy remain debated in different studies, making optimization of such systems a worthwhile topic for further exploration. Therefore, this paper will analyze the current mainstream recommendation algorithms in detail, discussing their performance in music recommendation and potential improvement directions.

This survey aims to analyze and compare commonly used recommendation algorithms in the music domain, exploring their advantages and disadvantages while providing corresponding suggestions based on practical application scenarios (Figure 1). Several classic algorithms will be introduced in detail, along with a tabular comparison of their performance in music recommendation.



**Figure 1.** The recommendation algorithms in the music domain (Photo/Picture credit : Original)

## 2. Method analysis

In current music recommendation system research, collaborative filtering, content-based recommendation, hybrid recommendation, deep learning recommendation, and graph model-based recommendation algorithms are commonly used. Each algorithm has its advantages and limitations, performing differently in various application scenarios and data environments.

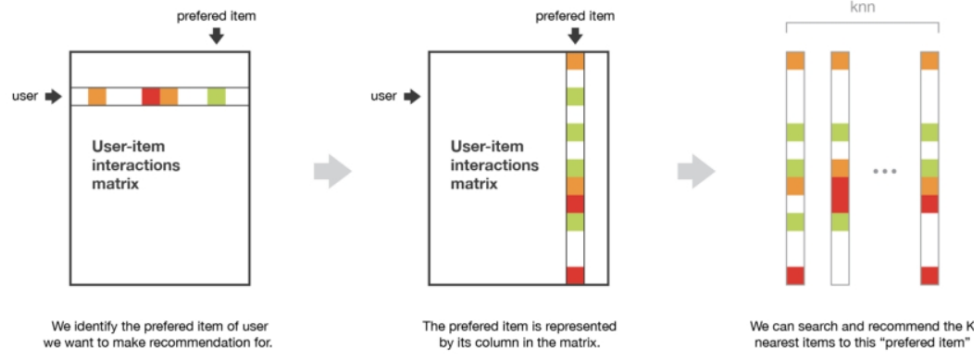
### 2.1. Collaborative filtering algorithm

Collaborative filtering is one of the most widely used algorithms in music recommendation systems. It mainly divides into user-based and item-based collaborative filtering.

By calculating the similarity between users, this method recommends music liked by other users with similar interests to the target user. Commonly used similarity metrics include cosine similarity and Pearson correlation coefficient. A typical implementation of this method is to use a neighborhood approach [4], identifying the K most similar users to the target user and recommending music based on their preferences. However, user-based collaborative filtering is significantly affected by data sparsity. As the number of users increases, the similarity matrix among users becomes sparse, making it challenging to find sufficiently similar neighbors. Additionally, when new users join the system, the lack of sufficient behavioral data makes it difficult to provide effective recommendations.

In contrast, item-based collaborative filtering analyzes the similarity between music items to recommend music similar to those the user has already listened to. Unlike user-based methods, item-based collaborative filtering focuses on exploring the associations between music items rather than between users. Its implementation typically involves calculating the similarity between music feature vectors, identifying the K most similar tracks to the target music, and recommending these tracks to the user. Item-based collaborative filtering is generally more robust to data sparsity issues than user-based methods because it focuses on the similarity between music items, which are usually fewer than users. However, when dealing with large music libraries, item-based collaborative filtering still faces

computational complexity challenges, as it requires calculating the similarity between all pairs of music items.



**Figure 2.** Matrix factorization method [3]

To address the data sparsity and scalability issues in collaborative filtering, latent factor models, such as matrix factorization, have been widely studied [2]. This method decomposes the user-item interaction matrix into low-dimensional latent factor matrices, capturing the implicit associations between users and music. Matrix factorization methods, such as singular value decomposition and latent semantic analysis, can effectively uncover users' latent interests (Figure 2) [3]. However, the training process for latent factor models requires substantial computational resources and has low interpretability, making it difficult to explain recommendations to users.

Spotify's recommendation system analyzes historical data from similar users to recommend songs favored by other users. For example, if two users share many identical playback records, Spotify recommends songs recently listened to by one user to the other. This method does not rely on music content features, making it widely applicable, but when user numbers are low or data is sparse, recommendation effectiveness may decline significantly.

## 2.2. Content-based recommendation algorithm

Content-based recommendation analyzes the content features of music (such as audio features, lyrics, artist information, etc.) to recommend music similar to the user's known preferences. Its core lies in generating feature vectors for each piece of music that represent its content information, then calculating the similarity between the user's interest feature vector and the music feature vector to make recommendations.

Audio feature extraction is a key aspect of content-based recommendation algorithms. Commonly used audio features include Mel-frequency cepstral coefficients, spectral centroid, and zero-crossing rate. These features can capture information about music's pitch, rhythm, and harmony. Additionally, lyrics features can be used to describe the content of music; methods such as bag-of-words models and word embeddings can be employed to analyze the semantic information in lyrics [5].

Content-based recommendations build a user interest model by analyzing the user's historical behavior. The user interest model can be simply represented as the average or weighted average of the features of music the user has previously liked. The user interest model is then compared with music feature vectors using similarity calculations such as cosine similarity or Euclidean distance to recommend music that is most similar to the user's interests.

Content-based recommendation performs well in addressing the cold start problem [6], as it does not depend on other users' behavioral data. It can create personalized recommendation models for each user, and due to its reliance on content features, it provides a certain level of explainability regarding why a specific music piece is recommended. However, this method has its limitations. Firstly, it overly depends on users' historical preferences, which may lead to a lack of diversity in recommendations. Secondly,

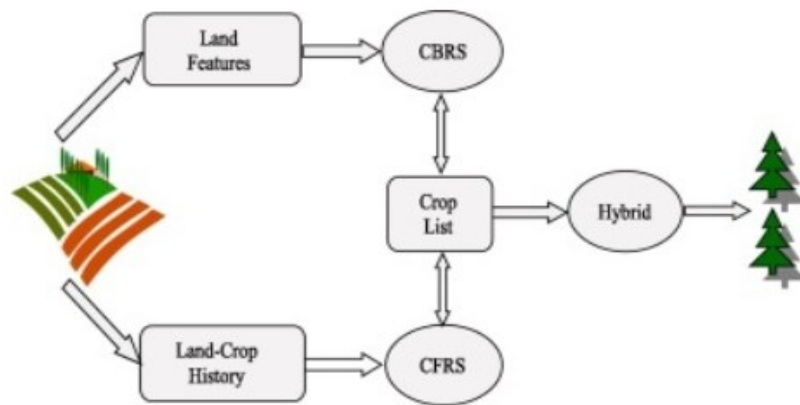
the feature extraction process can be complex and time-consuming, especially when involving high-dimensional and multimodal features. Furthermore, it fails to capture the potential associations between users and cannot leverage social information to enrich recommendations.

The Pandora radio recommendation system is based on the features of the music itself to recommend similar songs. Pandora assigns over 400 tags to each song, such as rhythm, melody, and instrument usage, to identify music that shares similar characteristics with the user's favorite tracks. Although this method is suitable for users with specific musical preferences, it may struggle to recommend diverse content if users have broad interests.

### 2.3. Hybrid recommendation system

Hybrid recommendation systems combine multiple recommendation algorithms to overcome the limitations of single algorithms [7], fully leveraging their strengths. Hybrid strategies include simple weighted fusion, cascading methods, feature-level fusion, and model-level fusion (Figure 3).

The simplest hybrid method involves weighted fusion of different algorithm outputs. For example, by averaging the results of collaborative filtering and content-based recommendations, a final recommendation list is obtained. Common weighted fusion methods include Bagging and Boosting, which are ensemble learning algorithms. Weighted fusion methods are straightforward and intuitive, but they require careful design of weight parameters to balance the contributions of different algorithms.



**Figure 3.** Hybrid recommendation system [7]

Cascading methods first generate an initial recommendation list using one algorithm and then re-rank that list using another algorithm. For instance, one might first use content-based recommendations to generate a candidate list of music and then apply collaborative filtering to reorder the candidate set to enhance diversity and precision. This method capitalizes on the strengths of both algorithms but also increases computational complexity.

Feature-level fusion combines features from different algorithms to construct richer feature representations. For example, the latent factors from collaborative filtering may be concatenated with content-based feature vectors to create a new feature input for machine learning models. This approach can capture richer feature information and enhance the model's expressive capability.

Model-level fusion simultaneously trains multiple models and combines their predictions. This can be achieved through ensemble learning methods such as random forests and gradient boosting trees, integrating the results of different recommendation algorithms. This method offers high flexibility and adaptability, effectively utilizing the advantages of different algorithms. However, it comes with increased complexity in the training and inference processes, requiring substantial computational resources.

In hybrid recommendation systems, cascading methods enhance efficiency and recommendation diversity but are limited by the performance of the initial algorithm. Feature-level fusion captures multi-

dimensional information and is suitable for multimodal data but may lead to high-dimensional challenges, increasing computational complexity. Model-level fusion takes advantage of various models and typically significantly improves recommendation effectiveness but incurs high computational costs and complex fusion processes. Overall, cascading methods are well-suited for rapid recommendations, while feature-level and model-level fusion methods are more suitable for complex and precise recommendation needs.

#### *2.4. Deep learning-based recommendation algorithms*

The application of deep learning techniques in music recommendation systems has made significant progress [8,9]. This field utilizes complex neural network models to automatically extract features from vast amounts of user behavior data and identify subtle differences in user preferences.

Deep Neural Networks (DNN) are widely used models that can perform non-linear transformations on high-dimensional features of users and music through multiple hidden layers, learning richer representations of user interests [10]. DNNs can process various types of input data, including user features, music features, and historical behavior, providing strong support for personalized recommendations. Additionally, Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) play important roles in music recommendation. Many music platforms, such as Deezer, have implemented recommendation systems based on CNNs. These systems analyze historical behavior data, including songs played, skipped tracks, and liked playlists, to model user preferences and suggest new music. Although this method performs well with large datasets, it still faces considerable challenges in scenarios involving data sparsity or cold starts.

To address these issues, graph model-based recommendation algorithms have gradually gained attention [11]. By constructing user-music relationship graphs, these algorithms effectively capture social relationships and interactions among users, providing more personalized recommendations. Techniques such as graph embedding (e.g., DeepWalk, node2vec, GraphSAGE) allow users and music to be mapped into low-dimensional vector spaces, preserving structural and attribute information in the graph. In the embedding space, similarities between users and music can be calculated to facilitate recommendations. This approach effectively captures high-order associations between users and music, enhancing recommendation accuracy.

Graph model-based recommendation algorithms excel in capturing complex relationships between user preferences and music, offering a more personalized recommendation experience. They perform particularly well in handling sparse data, as they can leverage connections within the graph to compensate for missing data, thus improving recommendation quality. However, they also entail high computational complexity, especially on large datasets, imposing significant demands on computational resources and storage. Furthermore, building and maintaining relationship graphs require substantial domain knowledge and meticulous data processing, which adds to the implementation difficulty. Additionally, graph model-based methods may underperform in addressing cold start problems, limiting their effectiveness for new users or new music.

For example, Tencent Music employs a graph model-based recommendation algorithm to recommend popular songs to users by analyzing their social relationships and interactions. If multiple friends are listening to the same song, the system may recommend that song to the user. This approach is particularly suitable for platforms with rich social interactions, but the complexity of building and maintaining relationship graphs remains a significant challenge when handling large user datasets.

Overall, the combination of deep learning and graph models provides new perspectives and methods for the personalization and accuracy of music recommendation systems. However, balancing complexity, computational resources, and recommendation quality remains a key issue for future development.

### **3. Experimental effect analysis**

In the study of music recommendation systems, various algorithms exhibit distinct performance characteristics. Collaborative filtering, a classic approach, identifies similarities by analyzing users' historical behavior, whether it be user-based or item-based (Table 1). This method demonstrates

relatively high accuracy and recall (F1 score of 0.70). However, its effectiveness can be compromised in scenarios with sparse data or when dealing with new user cold-start issues. The computational complexity is moderate, making it suitable for datasets of medium size. On the other hand, content-based recommendation focuses on the intrinsic attributes of the music, such as audio signals and lyrical content, to offer personalized recommendations. While this method provides a certain degree of personalization, it shows lower accuracy and recall (F1 score of 0.62), and due to the complexity involved in feature extraction, it incurs higher computational costs.

**Table 1.** Experimental result analysis

Category	Representative Method	Accuracy	Recall	F1-score	Computational Complexity
Collaborative Filtering	User-based/Item-based	0.72	0.68	0.70	Medium
Content-Based	Audio feature extraction/Lyrics analysis	0.65	0.60	0.62	High
Hybrid Recommendation	Collaborative filtering + Content-based	0.78	0.74	0.76	High
Deep Learning	DNN/RNN/Transformer	0.82	0.79	0.80	High
Graph Model	GCN/GAT	0.84	0.80	0.82	Very High

To leverage the strengths of both approaches, hybrid recommendation integrates collaborative filtering with content-based methods, aiming to enhance the overall performance of the recommendation system. Experimental results highlight that hybrid recommendations achieve impressive scores in terms of accuracy, recall, and F1 score (0.78, 0.74, and 0.76, respectively), effectively balancing recommendation quality and diversity. Nevertheless, this enhancement comes at the cost of increased computational complexity. The advent of deep learning has introduced models like DNNs, RNNs, and Transformers into the domain of music recommendation [12]. These models are particularly adept at capturing the intricate relationships between user behaviors and musical features. They perform exceptionally well across all evaluation metrics (F1 score of 0.80) and are especially effective for handling large-scale and high-dimensional data. Despite their advantages, these deep learning models also require substantial computational resources, reflecting the demands during training and inference processes.

By constructing user-music relationship graphs and employing graph neural networks, such as GCNs and GATs, these algorithms excel at uncovering complex social interactions and patterns. The experimental results show that graph-based methods achieve the highest scores in accuracy, recall, and F1 score (0.84, 0.80, and 0.82, respectively), demonstrating their capability to capture sophisticated connections between users and music. Nonetheless, they face significant computational challenges, particularly when processing large-scale graph data.

In summary, the selection of an appropriate recommendation algorithm should be carefully considered based on the specific application context, taking into account the trade-offs between precision, diversity, and computational efficiency. Each method has its unique advantages and limitations, and designing the most suitable solution for a given scenario is crucial.

#### 4. Conclusion

This paper explored the application and shortcomings of recommendation algorithms in the music domain. It first analyzed the mechanisms of different recommendation algorithms, including content-based recommendations, collaborative filtering, and hybrid recommendation systems, evaluating the performance of each algorithm. Through a literature review, we found that different algorithms exhibit significant differences in recommendation accuracy and user satisfaction, revealing their respective limitations. From the analysis, we can conclude that while each recommendation algorithm has its unique advantages, hybrid recommendation systems, which combine multiple algorithms, often provide

more precise and personalized recommendations in practical applications. Furthermore, with the development of big data and machine learning technologies, future recommendation systems are expected to become more intelligent and flexible, better adapting to user needs.

Looking ahead, we anticipate more innovative recommendation algorithms will be proposed with technological advancements. These algorithms will not only improve recommendation quality but also enhance user experience. Additionally, cross-domain recommendation research will become an important direction worth further exploration.

## References

- [1] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749.
- [2] Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 426-434.
- [3] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 993-1022.
- [4] Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 76-80.
- [5] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. MIT Press.
- [6] Schein, J. S., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 253-260.
- [7] Huang, Z., & Xie, L. (2010). Hybrid collaborative filtering. In *Proceedings of the 19th International Conference on World Wide Web*, 801-810.
- [8] Rendle, S. (2012). Factorization machines. In *2012 IEEE 11th International Conference on Data Mining*, 995-1000.
- [9] Zhang, Y., & Chen, L. (2013). A survey on multi-domain collaborative filtering. *Data Mining and Knowledge Discovery*, 27(1), 145-177.
- [10] He, X., Liao, L., Zhang, H., Nie, L., & Hu, X. (2017). Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, 173-182.
- [11] Ying, R., et al. (2018). Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 974-983.
- [12] Zhou, K., & Huang, D. (2018). The state of the art in recommender systems. *Computational Intelligence*, 34(2), 147-179.