

A Critical Analysis of Music Recommendation Systems and New Perspectives

Dushani Perera¹, Maneesha Rajaratne¹, Shiromi Arunathilake¹, Kasun Karunanayake¹,
Buddy Liyanage¹

¹University of Colombo School of Computing, Colombo, SRI LANKA

¹{2015cs102.stu, 2015cs107.stu, sda, ktk, buddy}@ucsc.cmb.ac.lk

Abstract. Many businesses enhance on-line user experience using various recommender systems which have a growing innovation and research interest. Recommender systems in music streaming applications proactively suggest new selections to users by attempting to predict user preferences. While current music recommendation systems help users to efficiently discover fascinating music, challenges remain in this research area. This paper presents a critical analysis of current music recommender systems and proposes a new hybrid recommender system with efficient and enhanced prediction capabilities.

Keywords: Human-Computer Interaction · Music Recommender System · User-centered Design · Usability and User Experience · User Customization

1 Introduction

With many music items available on the Internet, the task of finding acceptable and preferred music has become difficult. Music recommendation systems help to solve this problem. They help users to filter from the many items available and discover songs of their choice. Music recommendation systems are much more complex than other recommendation systems due to the following facts [1]:

Duration of the music item - Unlike movies or books where the duration is much longer, a song usually lasts 3 to 5 minutes which makes it more forgotten.

Vastness of items lists - A music catalogue has millions of songs compared to the number of movies in a movie catalogue. Hence, challenges such as scalability faced in building a music recommendation system are higher than any other type of recommendation system.

Continuous serial access - Unlike movies, music pieces are consumed more frequently in a session. The right arrangement of songs is a challenge.

Recommending similar music items more than once - Users prefer to listen to the same song twice, which is not the case in a movie recommendation system. Thus, the user may prefer the recommendation of a music item more than once.

Listening behavior - Unlike in situations such as watching a movie or reading a book, users do not usually pay attention when listening to a song; it mostly happens passively. This paper presents a summary of music recommendation systems with the goal of finding out how it could be improved.

2 Characteristics of Music Recommendation Systems

A music recommendation system mainly has three components; a User, a music item and an algorithm to match users with music items. The first component, User, addresses the variations in users' profiles. Preferences of users change due to many criteria. For instance, the difference in age, profession, gender, interests and many more aspects may cause differences in music preference. User preference variations can be captured using two factors. (1) User's personal profile - that is based on user's demography, psychography, and geography. (2) Music access pattern - defines how often a user listens to music [2]. Based on this information, a user's preferences are considered when designing a music recommendation system. Access patterns can also be used to improve dynamic optimization; to find the optimal recommendations for a user in the end. The second component, which is a music item, defines various information regarding the song. Information regarding a music item can be generated using two sub-factors. (1) Editorial data: metadata of the song (artist, composer, title, album name, date of release, genre, etc.) (2) Acoustic data: features of the item [2]. The third and final component is the user-item matching algorithm. This generates the link between the users and music items by matching users' preferences with features of music items. In music recommendation systems this is usually done by tracking initial details submitted, listening history, feedback, etc.

3 Approaches in Music Recommendation

There are mainly two ways in which recommendations can be generated: Collaborative Filtering and Content-Based Filtering. The following discussion explains these approaches.

3.1 Collaborative Filtering (CF)

Collaborative Filtering systems are built on the concept that users who rate items alike will continue to do so in the future. It recommends music items through the preferences provided by other similar users. They generate recommendations by considering ratings of music items between similar users. Last.fm is an example of a collaborative system. Thereby it creates a matrix of user-item ratings. This assists in identifying which users rate alike. This utilizes the nearest neighbor algorithm. However, there are advantages and disadvantages of using a user-item matrix. The center of recommendation systems is user data. This data can be obtained in two methods: explicit and implicit feedback.

(a) Explicit feedback: This data can be obtained when the user is taking direct action and that indicates his preference. For instance, one-to-five-star ratings or thumbs-up and thumbs-down feedback.

(b) Implicit feedback: Looking at the user behavior on the application. For instance, play counts can be used to infer an implicit rating.

There are three main subdivisions of Collaborative Filtering: memory-based, model-based, and hybrid collaborative filtering [1].

Memory-based Collaborative Filtering. Memory-based CF provides recommendations based on previous ratings. This is motivated by the observation that users usually trust the recommendations from like-minded neighbors. This became very popular because they are easy-to-implemented, very intuitive, avoid the need for training and tuning many parameters, and the user can easily understand the rationale behind each recommendation [3][4].

Model-based Collaborative Filtering. This approach uses data mining and machine learning algorithms for the system to train and provide recommendations. User preferences are represented by a set of rating scores and construct special prediction learned complex patterns based on training data [2]. Matrix factorization (MF) is the most popular CF technique used in music streaming platforms. A proven method used in addressing scalability and sparsity is Matrix factorization-based CF algorithms. MF is a linear technique [7]. Alternatives to address limitations of MF are Probabilistic MF and Weighted MF. The three types of Collaborative Filtering techniques have advantages and disadvantages which have been discussed in Table 1. CF methods face certain challenges due to the way and means of its data-gathering methodologies, implementation, design, etc. which have been addressed in Table 2.

Table 1. Comparison of collaborative filtering technique

Collaborative Filtering (CF) Approach	Main advantages	Main shortcomings
Memory-based CF	<ul style="list-style-type: none"> • Easy implementation. • New data can be added easily and incrementally. • Need not consider the content of the songs being recommended. 	<ul style="list-style-type: none"> • Dependent on human ratings. • Performance decrease when data are sparse. • Have limited scalability for large datasets. • Startup problem prevails.
Model-based CF	<ul style="list-style-type: none"> • Address the sparsity, scalability. • Improve prediction performance. • Give an intuitive rationale for recommendations. 	<ul style="list-style-type: none"> • Expensive model building. • Lose important details due to dimensionality reduction techniques.
Hybrid recommenders	<ul style="list-style-type: none"> • Overcome limitations of CF approaches such as sparsity and gray sheep. 	<ul style="list-style-type: none"> • Increased complexity. • Expensive implementation. • Need external information that is usually not available.

Table 2. Limitations of collaborative filtering

Limitation	Cause and Explanation
Data Sparsity	Recommender systems are used to evaluate very large item sets. Therefore, the user-item matrix used for collaborative filtering will be extremely sparse.
Human Effort	User's effort to rate songs using explicit data [18]. Human effort is needed to fill user preference surveys [12].
Popularity Bias	Songs with higher ratings tend to be recommended more than the less rated songs [12].
Cold start Problem	New songs cannot be recommended until some user rates it, new user isn't likely to get recommendations until his profile is evaluated [5].
Scalability	When the existing number of users and songs grow, CF algorithms suffer from scalability problems [5].

3.2 Content-Based Filtering (CBF)

This approach analyses song tracks to predict user preferences. There are two methods to this; low-level filtering and high-level filtering. Low-level filtering takes a music item's metadata such as artist, album, year of release, genre, etc. as inputs. High-level filtering considers tempo, pitch, loudness, and instruments etc. as inputs.

Challenges of Content-based Filtering. 'Startup problem' is the limitation created by features explicitly associated with the songs recommended [5]. 'Overspecialization problem' recommends items that score highly against a user's profile. This approach relies on the details of the music items thus, the system is limited. This is called the "glass-ceiling effect" [2]. For instance, this approach doesn't have the ability to recognize a rock song with melodic lyrics and a rock song with non-melodic lyrics. Thus, this system may predict screamed lyrics songs to users who prefer melodic lyrics.

3.3 Other Methods in Music Recommendation

Hybrid Filtering. This method of filtering combines the advantages of both CF and CBF and can avoid their individual limitations [14][15].

Emotion-Based Model. Emotion-based methods use two factors to perceive emotions: how positive or negative, the emotion is and how exciting or calming the emotion is. One challenge is data collection since in order to accurately model the system, a large amount of data is needed [16]. Ambiguity and granularity is a significant issue since emotion itself is hard to define and describe [17].

Context-Based Model. This approach uses public opinion and information to collect details regarding music items such as genre, artist, etc. in providing recommendations. Challenge in Emotion-Based Model is 'Popularity Bias'.

4 Proposed Recommendation System

4.1 Dynamic Playlist Generation

The proposed system will address three main components. First component is Dynamic Playlist Generation. Users are often required to remember or search through an entire collection of media items to locate a song. When the collection of media grows, the required effort also increases. Generating and recommending a playlist for users is a great method to deal with this issue [8][9]. There are several techniques:

(a) Neighborhood Recommendations - KNN approaches with binary cosine similarities between playlists provide sufficiently accurate recommendations.

- I. KNN + Sequential patterns
- II. Bayesian Personalized Ranking (BPR)
- III. Content-based approach (using metadata of songs) with KNN.

(b) Content-based Approaches - Using content information of tracks (metadata) may lead to the successful tackling of the cold-start problem of CF methods and add diversity and novelty to recommendations. [8].

4.2 Next-track Recommendation

Next-track Recommendation is a specific form of music recommendation that relies mainly on the user's recently played tracks to create a list of tracks to be played next [10]. A particular challenge in this context is that the recommended tracks should not only match the general taste of the listener but should also match the characteristics of the most recently played tracks [6]. Session-based approaches [13] are a good way to predict the user's immediate next actions [5][6].

4.3 Automatic Playlist Continuation

Automatic Playlist Continuation is more challenging and newest research area in the Music Streaming industry. The basic concept is to give a user a playlist containing some seed tracks, thereafter a list of tracks should be recommended for continuing the playlist. Selected method is hybrid session-based recommendations [11].

5. Future Research

This research is also developing an App, 'HelaBeat', for Android devices. User preferences are captured at first installation through the beta-version available on the Google Play Store. Further analysis and research will be undertaken to incorporate recommendations and new music fingerprinting and identification system within the same app.

6. Conclusion

This paper presents a survey of state-of-the-art music recommendation systems, and a proposed hybrid recommendation system that will perform better than other published methods. The outcome of the complete research work, including the full implementation of the new proposed approach within the Hela-beat app, will provide users with seamless streaming of songs and more effective recommendations.

7. References

1. Xiaoyuan, Su, Taghi, M., Khoshgoftaar.: A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, (Section 3):1–19 (2009)
2. O'bryant, J.: A survey of music recommendation and possible improvements. no.April (2017)
3. Hu, Yifan, Koren, Yehuda, Volinsky, Chris.: Collaborative Filtering for Implicit Feedback Datasets. *Proceedings. In: IEEE International Conference on Data Mining, ICDM. 263-272. 10.1109/ICDM.2008.22 (2008)*
4. Karypis, George.: Evaluation of Item-Based Top-N Recommendation Algorithms. 10.1145/502585.502627 (2000)
5. Ludewig, Malte, Jannach, Dietmar.: Evaluation of Session-based Recommendation Algorithms. In: *User Modeling and User-Adapted Interaction*, 10.1007/s11257-018-9209-6 (2018)
6. Iman, K.: *Advances in Next-Track Music Recommendation* (2017)
7. He, Xiangnan, Liao, Lizi, Zhang, Hanwang.: Neural Collaborative Filtering. In: *Proceedings of the 26th International Conference on World Wide Web* (2017)
8. Hariri, Negar, Mobasher, Bamshad, Burke, Robin.: Context-Aware Music Recommendation Based on Latent Topic Sequential Patterns. 10.1145/2365952.2365979 (2014)
9. Celma, Óscar.: Music Recommendation. 10.1007/978-3-642-13287-2_3 (2010)
10. Quadrana, Massimo, Cremonesi, Paolo, Jannach, Dietmar.: Sequence-Aware Recommender Systems. In: *ACM Computing Surveys*, 51. 10.1145/3190616 (2018)
11. Ludewig, Malte, Kamehkhosh, Iman, Landia, Nick, Jannach, Dietmar.: Effective Nearest-Neighbor Music Recommendations. 1-6. 10.1145/3267471.3267474 (2018)
12. Srebro, Nathan, Rennie, Jason, Jaakkola, Tommi.: Maximum-Margin Matrix Factorization. In: *Adv. Neural Inf. Proces, Syst.* 17 (2004)
13. Zhang, Shuai, Yao, Lina, Tay, Yi, Xu, Xiwei, Zhang, Xiang, Zhu, Liming.: Metric Factorization: Recommendation beyond Matrix Factorization (2018)
14. Lops, Pasquale, Gemmis, Marco, de., Semeraro, Giovanni.: Content-based Recommender Systems: State of the Art and Trends. In: *Recommender Systems Handbook*. Boston, MA: Springer US, pp. 73–105 (2011)
15. Will, H., Larry, S., Mark, R., George, F., South S.: Recommending and Evaluating Choices in a Virtual Community of Use. *Mosaic: A Journal for The Interdisciplinary Study Of Literature*, pages. 5–12 (1995)
16. Barragáns-Martí, A., B., Costa-Montenegro, E., Burguillo, J., C., Rey-López, M., Mikic-Fonte, F., A., Peleteiro, A.: A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Information Sciences*, 180 (22) (2010)
17. Burke, Robin.: *Hybrid Recommender Systems: Survey and Experiments* (2002)
18. Yang, B., Lee, S., Park, S.: Exploiting various implicit feedback for collaborative filtering. In: *Proceedings of the 21st International Conference on World Wide Web, ACM*, pp. 639–640 (2012)