A Survey of Music Recommender Systems

Aishwarya Sharma

Contents

ABSTRACT

Music Recommender Systems(MRS) has experienced a lot of new developments in the past few years due to the vast majority of online streaming platforms, which have made available all music in the world at the user's fingertip. With such a user base and many choices, the music recommender system is essential for enhancing the user experience. In this paper, different types of approaches to MRS are outlined, along with the challenges that MRS research is facing.

1. INTRODUCTION

Music is an essential aspect of everyday life. People listen to music often while performing other activities such as studying, workout, cooking, etc. Researchers have also found a direct/indirect impact of music on performance. It can not only convey Emotion but also affects the listener's mood. The taste of music, just like any other entertainment category such as movies, varies from person to person and is highly dependent on a multitude of factors.

Building a successful recommender system requires considering the most recent interaction information's implicit, explicit, and contextual aspects. For example, the emotional state of the listener, weather conditions, social surroundings, their activity are all known to influence musical taste and needs.

The recommender systems often use the history of user preferences to provide them with a list of suggestions that they are likely to enjoy. Such a system includes extracting the information and presenting the recommendation to the users based on the pre-existing information.

MRS plays a vital role in enhancing the user experience and suggests content. There are many approaches to the recommender systems, two of which are *Collaborative filtering* and *Content filtering*.

2. THE GOALS OF RECOMMENDER SYSTEMS

With deploying a recommender system to a product, the main goal is to enhance sales and profit. It increases the interactivity and the users' experience. In terms of music, the main goal of a recommender system is:

1. **Relevance**: User is more likely to enjoy the music they find interesting.
2. **Novelty**: The system should recommend new genres and artists to users and help discover new music and artists.
3. **Diversity**: The recommender system should contain a set of different songs, which increases the possibility of the user liking the suggestions presented to him.
4. **Serendipity**: The recommendation should contain the surprise element and should be unexpected. The user might lose interest if the system repeatedly recommends the same songs.

3. CHALLENGES IN MUSIC RECOMMENDER SYSTEMS

3.1 Cold start problem

When a new user signs up in the system or a new song is added to the catalog, the system does not have enough data associated with the song or the user. As a result, the system cannot recommend the new song to the existing users or the existing songs to the new users.

A sub-problem of the cold start problem is *sparsity* when the number of users and the items in the system is vast. *The inverse of the ratio between given and possible ratings is called sparsity.* [9]

The *Content-based approach* does not rely on the ratings from other users other than the target users. Thus, this approach helps in cold-start scenarios as long as the system has user preferences. Both content-based and collaborative filtering suffers from the cold start problem (elaborated in section 4). However, the former faces this problem for both new users and new items, while the latter have cold-start problems for only new users.

3.1.1 Handling cold start problem

For the new item problem, extracting the features can also be leveraged to help predict users' tastes and interests, which can be used in the subsequent stages.

Hybridization is an alternative technique to handle this problem. In [1], the author proposes combining an acoustic CB and item-based CF recommender. It involves extracting acoustic features, including spectral properties, rhythm, and Pitch, which helps the collaborative filter tackle the cold start problem.

Cross-domain recommendation technique aims at improving recommendations in one domain (here music) by making use of information about the user preferences in an auxiliary domain.[2] The user preferences, including choices non-related to music, for example, the personality traits of a user are transferred from an *auxiliary* domain. It results in a more accurate and complete user model.

Active learning can also help deal with the cold-start problem. The system interactively improves its performance based on user feedback.

3.2 Automatic playlist continuation

A playlist is a sequence of songs or tracks intended to listen together. The task of generating such a playlist is attributed as *Automatic Playlist Generation* (APG). Markov chains are often preferred to model the transition between the songs in a playlist [3].

A variation of APG is *Automatic Playlist Continuation* (APC). When a user searches and plays a song, the music recommender system continues to play the songs that match the played song's characteristics, such that listeners can continue enjoying the session beyond the finite length of one song or a playlist.

This task involves inferring the intended purpose of a given playlist. A cold start scenario would be when a new playlist is created, no songs are added, with only a few metadata available (e.g., the playlist's title).

The system can use available data such as the user's profile, previous user-generated playlists, and the listening history.

3.2.1 Handling automatic playlist continuation

For this task, a common approach is to extract the *background knowledge* using machine learning techniques from the manually curated playlists. In the study [4] the authors used the already present playlists on Spotify, and these playlists were analyzed to create contextual clusters, improving recommendations.

Generative machine learning models address the song order. In [5] the authors train various Markov chains to model transitions between songs.

Given many playlists created by users and shared among other users on the online streaming platforms, a possible starting point in the APC task might be the metadata associated with user-generated playlists, such as title or description.

3.3 Evaluating music recommender systems

The accuracy and other related quantitative measures such as precision, recall, or other measures are the most used criteria to evaluate a recommender system. There are other criteria for evaluating recommender systems, such as Novelty, diversity of Serendipity of the recommended song.

Mean Absolute Error (MAE) computes the average absolute deviation between the predicted ratings and the actual ratings provided by users [6].

$$MAE = \frac{1}{|T|} \sum_{r_{u,i \in T}} |r_{u,i} - \hat{r}_{u,i}|$$

Where $r_{u,i}$ and $\hat{r}_{u,i}$ are the actual and the predicted ratings of item i for user u, respectively.

Root-mean-square error (RMSE) is an extension to MAE, where the more significant differences between predicted and accurate ratings are penalized more than smaller ones.

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{r_{u,i \in T}} (r_{u,i} - \hat{r}_{u,i})^2}$$

Novelty measures the ability of a recommender system to recommend new items that the user did not know about before [7].

$$novelty = \frac{1}{|U|} \sum_{u \in U} \sum_{i \in L_u} \frac{-\log_2 pop_i}{N}$$

L_u is the list of recommended songs for the user u, and pop_i is the song's popularity.

Serendipity evaluates MRS based on the surprising and relevant recommendations.

$$serendipity = \frac{|L_u^{unexp} - L_u^{useful}|}{|L_u|}$$

4. APPROACHES TO RECOMMENDER SYSTEMS

4.1 Collaborative Filtering

Collaborative filtering predicts the user's preferences based on the history of their preferences and preferences of similar users. It focuses on the relationship between objects.

In the *User-based* filtering, the ratings provided by similar users to user A are used to make the recommendations for A.

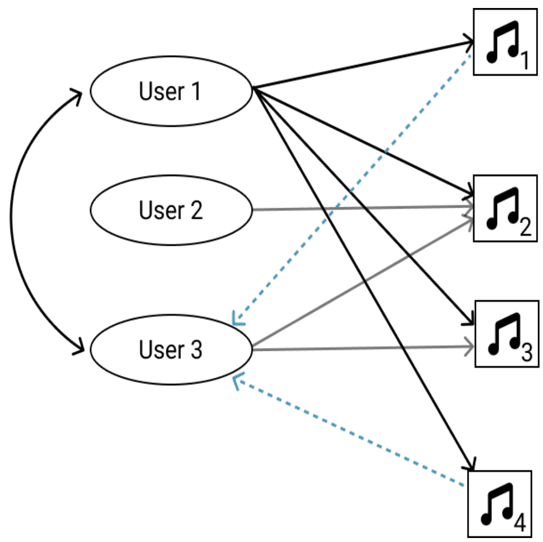

Fig. 1. User based filtering

In the figure 1, user-3 is recommended songs 1 and 4 since user 1 likes the same songs.

The other approach to collaborative filtering is *item-based* filtering. If user A likes some items from a set of items, we recommend more items from this set. For example, if the user likes songs from some artist, we recommend more songs from the same artists.

In the figure 2, user-3 is recommended song-1 since songs 1 and 3 belong to the same set.

4.1.1 Similarity Measure

Let's assume U be the set of n users and I be the set of m songs. We create a matrix $R = \{r_{ui} \in \mathbb{R}^{n\times m}\}$, based on the users' past preferences. This matrix contains information about how frequently user u listen to the song i.

The recommender system uses different similarity measures to compute the similarity between the users and between items: Euclidean distance, cosine metric, Pearson correlation, Manhattan distance, and others. *Cosine metric* can be used for both user-based and item-based collaborative filtering.

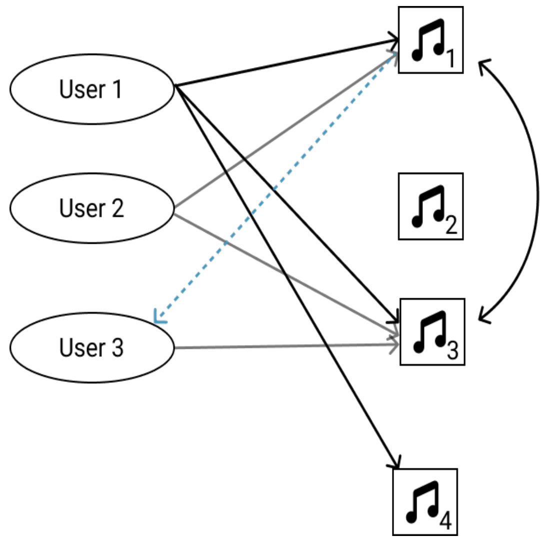

Fig. 2. Item based filtering

Let's assume there are two users u and v, and $I(u)$ be the set of songs rated by user u. Then the cosine similarity between user u and user v is given by:

$$w_{uv} = \frac{I(u) \cap I(v)}{|I(u)|^{\frac{1}{2}}|I(v)|^{\frac{1}{2}}}$$

Similarly, Let there be two songs i and j, and $U(i)$ be the set of users who listened to song i:

$$w_{ij} = \frac{U(i) \cap U(j)}{|U(i)|^{\frac{1}{2}}|U(j)|^{\frac{1}{2}}}$$

4.1.2 Scoring function

A *scoring function* takes user u and item i as the input parameters and outputs a score that quantifies how strongly does a user u likes the item i.

For the user-based filtering the scoring function is computed by:

$$h_{ui}^{U} = \sum_{v \in U} f(w_{uv}) r_{ui} = \sum_{v \in U(i)} w_{uv}$$

The score for the song i depends on the similarities between the target user u and other users v having the item i in their listening history. A piece will have a higher score if similar users often stream it.

Similarly for the item-based collaborative filtering, the score function is computed by:

$$h_{ui}^{S} = \sum_{j \in I} f(w_{ij}) r_{uj} = \sum_{j \in I(i)} w_{ij}$$

The score depends on the similarities between the song i and other users' listening history u.

4.1.3 Ranking Aggregation

For achieving greater efficiency in recommending songs to the users, it is better to use an ensemble of strategies instead of relying on a single strategy to do the work. Each strategy deployed in recommending a song to the user can focus on different aspects of the user. Each strategy is likely to recommend songs different from other set strategies. We can achieve goals to the recommender systems such as Novelty, uncertainty, diversity, and Relevance. Thus, it is better to use an aggregation of the results of these strategies in a recommender system.

4.1.4 Evaluation Metrics

The recommender systems are evaluated based on two methods: *Online methods* and *Offline methods.* In an online way, for the presented recommendations, users' reactions are measured.

Out of the two, the most popular evaluation technique is offline methods. One of the most popular computing accuracy metrics is RMSE or Root Mean Squared Error.

$$RMSE = \sqrt{\frac{1}{|k|} \sum_{(u,i) \in k} (p_{ui} - r_{ui})^2}$$

This error metric is often employed in the regression model. Here, p_{ui} is the predicted rate, k is the number of testing rates. The smaller the RMSE, the better the performance of the recommendation engine.

Mean Average Precision (mAP) is another form of evaluation metric.

$$p_k(u,y) = \frac{1}{k} \sum_{p=1}^{k} r_{uy(p)}$$

4.1.5 Drawbacks:

Collaborative based filtering suffers from a few disadvantages:

- **Scalability:** When the number of users and items becomes vast, it requires high computing resources.
- **Cold Start:** It involves two issues: *new user problem* (lack of user data, when a new user signs up) and *new item problem* (we do not know who to recommend new items to).
- **Sparsity:** When there is a vast number of items as compared to users, the ratings are available to only a subset of the item.

4.2 Content Based Approach

The content-based filtering approach of recommendation involves analyzing the content of the items candidates for recommendation. It is based on the features extracted from the songs in users' listening history, and the recommendations are provided based on the similarities between the songs. The similarities are estimated based on the features extracted from the songs.

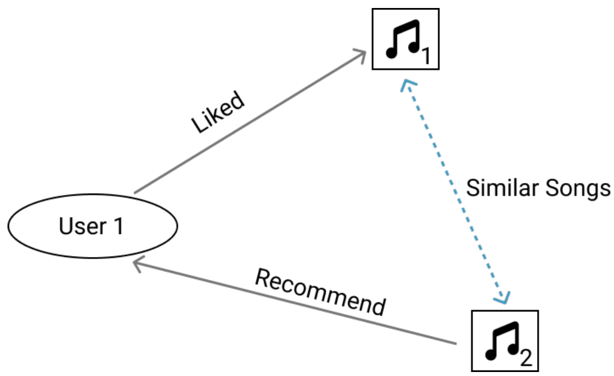

Fig. 3. Content based recommender system

4.2.1 Features for content based filtering

The features extracted from the songs can be classified into three categories:

1. **Low level features:** Some features can be directly extracted from the raw audio file and can be grouped according to their nature.

   Temporal features:

   (a) **Zero-cross rate**: Number of crossings between the signal and the 0-axis for a given time frame.

   (b) **Root Mean Square energy:** It corresponds to the perception of sound intensity. On one frame t

   $$RMS_t = \sqrt{\frac{1}{K} \cdot \sum_{k=t \cdot K}^{(t+1) \cdot K - 1} s(k)^2}$$

   Spectral features:

   (a) **Spectral centroid:** It represents the band where the most energy is. For a frame t

   $$SC_t = \frac{\sum_{n=1}^{N} m_t(n) \cdot n}{\sum_{n=1}^{N} m_t(n)}$$

   (b) **Spectral skewness:** It represents how symmetric the distribution is around its mean value. If the value is 0, it is symmetric. However, a higher value suggests skewed on the left, and a lower value indicates skewness to the right.

$$\gamma_1 = \frac{m_3}{\sigma^3}$$

(c) **Spectral flux:** It represents the change in power between two consecutive frames.

$$F_t = \sum_{n=1}^{N} (D_t(n) - D_{t-1}(n))^2$$

Loudness: The loudness of a song corresponds to the sound volume. The human ear generally focuses on the frequencies 2000-5000 Hz though it varies with age, culture, population, etc.

2. **Middle-level features:** These are the features that are specific to music and are understood by music experts, such as *Pitch, tonality* or the temporal and rhythmic properties such as the duration of the track, metric levels, beat, and rhythm.

3. **High level features:** A non-expert or a common user can understand these features. These features include, *danceability, instrumentals, instruments, artist, lyrics,* etc.

4.2.2 Feature selection:

We have many features that are attributed to a song. However, the next step is to eliminate the irrelevant or less significant features. This step is essential as we want to decrease the complexity of our model and the computation time. Various algorithms select the crucial features from a song that makes more reliable predictions. These algorithms are divided into three categories: **Filter, Wrapper** and **Embedded** model.

4.2.3 Models for recommendation:

The next step is to deploy a machine-learning algorithm to generate recommendations based on the features extracted from the songs. For the recommender systems, supervised learning is used to learn from a set of observations. The most popular algorithms are:

1. **Logistic regression:** Although it is not the most efficient method for recommendation, Logistic regression has an advantage of being fast. The hypothesis function is given by:

$$H(x) = \frac{1}{1+e^{\theta x}}$$

The songs are classified according to the genres using logistic regression, and thus it is a case of multi-class classification. Logistic regression uses **One vs. All** method to classify the songs.

2. **Decision Trees:** A decision tree is composed of nodes, leaves, and edges. The nodes represent the test performed on the parameters; edges are the result of the trials, and leaf nodes represent the final decision or class. The decision tree algorithm uses information gain and entropy to select the parameters at each step.

The parameter is chosen as the node which has highest information gain and lowest entropy. Entropy is calculated by:

$$E = -\sum_{i}^{C} p_i \log_2 p_i$$

3. **K-nearest neighbors :** The algorithm looks for the K closest instances to make a prediction. Based on these instances, the value of the variable is predicted. The value of K is chosen to avoid under-fitting and over-fitting.

   The value of k is selected such that it avoids under-fitting. On the other hand, it should not be too large to prevent over-fitting. There are many criteria to determine the distance between the data-set points, such as *Euclidean distance, Manhattan distance, hamming distance*, etc.

4. **Support vector machines:** It is one of the most widely used methods for music classification. The basic principle is to maximize the margin between the two data groups. The idea is that the data can be linearly separable when represented in high dimensions.

   Thus, the first step is to transform the data to high-dimensional space and then find the data the separates the two groups with maximum margin. The *Kernel functions* is used to get the benefits of high-dimensional space. Some of the commonly user kernel functions are: *linear, polynomial* and *radial* based.

5. **Neural network:** Neural networks are another widely used method for a recommendation.When used with *High-level* features, it gives about 85% accuracy. An artificial neural network is made up of neurons. The signal is transmitted between neurons in the network, and the signal received by a neuron is the weighted sum of different neurons. The output j of the neuron is given by:

$$y_j = f\left(\sum_{i=1}^{N} w_{i,j} x_i\right)$$

   where $w_{i,j}$ is the weight of the inter-connection between neurons i and j and x_i is the signal received by neuron i from j.

4.2.4 Evaluation

There are two stages for the evaluation of the recommendation system. One is based on mathematical and statistical techniques, and the second requires that users listen to the recommended songs.

1. **Evaluation using labels**

   The main goal of the recommender system is to ensure that the predicted elements are relevant, which is evaluated based on *precision.*

$$P_c = \frac{|Rel_c \cap Ret_c|}{|Ret_c|}$$

where, Ret_c is the total number of retrieved items and Rel_c is the total number of relevant items for the class c. Let C be the total number of classes, then the average precision is given by:

$$P = \frac{1}{|C|} \sum_{c \in C} P_c$$

Another parameter for evaluation is **recall**, which gives the number of relevant items retrieved.

$$R_c = \frac{|Rel_c \cap Ret_c|}{|Rel_c|}$$

The average recall is given by

$$R = \frac{1}{|C|} \sum_{c \in C} R_c$$

2. **Evaluation using confusion matrix**

   A confusion matrix can be a helpful evaluation tool for multi-class classification. Ideally, it is expected to have a high accuracy percentage diagonally and other values as low as possible.

3. **Evaluation based on feedback**

   Besides predicting correct genres, it is essential to predict songs relevant to the target user. However, this method of evaluation is expensive in time and is subjective. Users' opinions can differ, and thus we may not generalize the outcome.

   To make this method effective, it is preferable to have diversity; based on demographic, age, gender, personality-based, and geography.

4.2.5 Drawbacks:

1. **Diversity:** The content-based method limits the diversity in the recommendation as it tends to over-specialize.

2. **Cold start:** A new user has to listen and evaluate a certain amount of songs before the system can generate recommendations for them. Thus the integration of a new user cannot be immediate.

4.3 Hybrid approach

It is also possible to combine the complementary recommendation methods to create a hybrid system. This approach eliminates the cold start problem, which is the drawback of content-based and collaborative filtering. These hybrid systems are constructed using four traditional recommendation techniques: content-based, collaborative filtering, knowledge-based, and demographic.

Hybrid techniques outperform conventional recommendation techniques. The knowledge-based recommendation system uses the users' preferences and needs to recommend songs, contrasting to the other conventional approaches, which utilize learning algorithms. One hybridizing approach combines the collaborative and content-based approach or different algorithms such as naive Bayes and K-nearest neighbor content-based models.

5. Future Work in MRS

Studies [8] have shown that the mood, activity, or even the location of the person influences the music they want to listen to. We listen to music in a given moment, in a predefined emotional state, and established circumstances. Further, psychological constructs such as Personality and Emotion could also be integrated with the models to improve the quality of song recommendations.

5.1 Including psychological constructs in music recommendation

Personality based MRS system can predict songs based on the Personality. For example, extravert people enjoy more upbeat and conventional music. Such a recommendation system can predict which songs are likely more enjoyable by extroverted people than others. Such information can also help classify like-minded users and be integrated with a neighborhood-based collaborative filtering approach.

Emotion also has a short-term or long-term effect on musical preferences and tastes. We can also say that music also impacts the emotional state of the user. It can be used as tags in songs, classified based on different emotions.

The psychological aspects can be gathered either implicitly or explicitly. For example, in the case of Personality, the system can ask users to answer questionnaires, or the system can learn Personality by tracking and *observing users' behavioral patterns, for instance, liking behavior on Facebook.* [9]

6. Conclusion

In this survey paper, we discussed the goals of the MRS and the challenges which are usually faced by the conventional MRS approaches. We discussed about the (1) *cold start problem* of items and users, (2) *Automatic playlist continuation* and (3) *challenges in evaluating music recommendation systems.*

We discussed the conventional approaches to MRS, *content based model* and *collaborative filtering* the evaluation criteria used in the approaches and their drawbacks. We also talked about the hybrid approach, which aims at eliminating the two major drawbacks faced by such models *cold start problem* and *sparsity*.

We also discussed how psychological aspects could be included in MRS to recommend a better song particular to a target user.

BIBLIOGRAPHY

[1] Donaldson J (2007) A hybrid social-acoustic recommendation system for popular music. In: Proceedings of the ACM conference on recommender systems (RecSys), Minneapolis, MN, USA

[2] Cantador I, Cremonesi P (2014) Tutorial on cross-domain recommender systems. In: Proceedings of the 8th ACM conference on recommender systems, RecSys'14. ACM, New York, NY, USA, pp 401–402. https://doi.org/10.1145/2645710.2645777

[3] McFee B, Lanckriet G (2011) The natural language of playlists. In: Proceedings of the 12th international society for music information retrieval conference (ISMIR 2011), Miami, FL, USA

[4] Pichl M, Zangerle E, Specht G (2015) Towards a context-aware music recommendation approach: what is hidden in the playlist name? In: 2015 IEEE international conference on data mining workshop (ICDMW). IEEE, pp 1360–1365

[5] McFee B, Lanckriet G (2011) The natural language of playlists. In: Proceedings of the 12th international society for music information retrieval conference (ISMIR 2011), Miami, FL, USA

[6] Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):5–53. https://doi.org/10.1145/963770.963772

[7] Adamopoulos P, Tuzhilin A (2015) On unexpectedness in recommender systems: or how to better expect the unexpected. ACM Trans Intell Syst Technol 5(4):54

[8] Marius Kaminskas and Francesco Ricci. Contextual music information retrieval and recommendation: State of the art and challenges. Computer Science Review, 6(2):89 – 119, 2012.

[9] Schedl, M., Zamani, H., Chen, CW. et al. Current challenges and visions in music recommender systems research. Int J Multimed Info Retr 7, 95–116 (2018). https://doi.org/10.1007/s13735-018-0154-2

[10] https://pure.port.ac.uk/ws/portalfiles/portal/14416496/User_Based_Hybrid_Algorithms_for_Music_Recommendation_Systems_Thesis_Murtadha_Al_Maliki.pdf

[11] MARINE CHEMEQUE-RABEL Content-based music recommendation system: A comparison of supervised Machine Learning models and music features http://www.diva-portal.org/smash/get/diva2:1515217/FULLTEXT01.pdf

[12] E. Shakirova, "Collaborative filtering for music recommender system," 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2017, pp. 548-550, doi: 10.1109/EIConRus.2017.7910613.