reco.mu: A Music Recommendation System Depending on Listener's Preference by Creating a Branching Playlist

Kosuke Nonaka[1] and Satoshi Nakamura[1]

[1] Meiji University, 4-21-1 Nakano, Nakano-ku, Tokyo, Japan
cs202014@meiji.ac.jp

Abstract. It is not easy to recommend various content, including music, to others. This paper proposes a method that enables people to recommend music depending on listeners' preferences by creating branching playlists. By evaluating the method's effectiveness, we found that branching playlists increased the degree of satisfaction, familiarity, and interest of the listener. We also implemented a Web based music recommendation system called "reco.mu" that incorporates the proposed method. We found that the creator becomes more conscious of the recommender when creating a branching playlist.

Keywords: Recommendation, Music, Playlist, Word-of-mouth, CGM.

1 Introduction

The development of music streaming services has led to a rapid increase in music in circulation, allowing listeners to access any music at any time and any place. For example, Apple Music [1] offers 70 million pieces of music, and Spotify [2] provides more than 50 million pieces of music. It has also become easier to share such content with others online, and the Internet, including social networking services, is filled with recommendations by listeners.

Due to the increase in the number of such contents explosively, popular and well known music is easy to find, music that is not well known is difficult to find. For example, music Websites and CD stores often list popular music by ranking, making such music more visible to the public at the expense of lesser-known music. One shocking study conducted by MIDiA consulting in 2014 [3] reported that 77% of global music revenue in 2013 came from the top 1% of artists. They point out that this is due to the oversupply of music to listeners and that excessive choice interferes with exploration. Therefore, people may not even be aware of music that they might have liked. Furthermore, artists may not earn sufficient income due to this sales bias, which may adversely affect their ability to produce more music. As a result, their fans will lose a chance to listen to their new music because they sometimes give up creating music. To address such issues, we need to make lesser-known (buried) artists and genres more visible to the public and fully understand their appeal. Therefore, we focused on the fans of such artists actively recommend music to others, drawing them into the artist or genre, and increasing new fans.

Since people familiar with specific music have already accumulated knowledge about that music, they can recommend music considering the knowledge level and preferences of a recommendation target (hereafter, listener). When making a recommendation by speaking directly, it is not uncommon to flexibly change the content of the recommendation depending on the response of listeners. Changing such content makes it possible to present more suited to the individual's preferences, which improves recommendation satisfaction.

Based on the idea, we propose a method that enables people to recommend their guess artists' or genres' music effectively by changing the following music depending on the listener's preference. Specifically, the method builds a playlist with a branching structure. Recommendations are made by representing the interactive conversation that people have as conditional branches in a flowchart (hereafter, a branching playlist) (see Fig. 1). By changing the content of the recommendation depending on the listener's response, it is possible to present information that is more suited to the individual's preferences. We believe that the method could improve recommendation satisfaction. Furthermore, we clarify how recommenders create branching playlists and their properties by implementing a prototype Web based music recommendation system called "reco.mu" based on the proposed method.

The contributions of this paper are as follows.

- We proposed a method that enables fans to recommend their favorite music according to the listener's preferences using a branch structure's playlist and clarified its effectiveness.
- We implemented a Web system based on the proposed method and evaluated the effectiveness from the recommenders' and listeners' feedbacks.

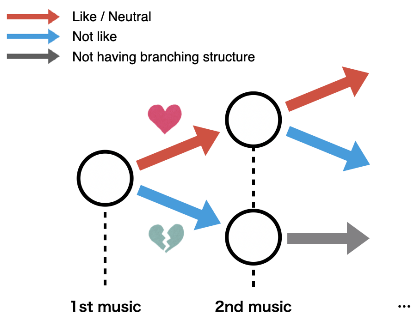

Fig. 1. Overview of branching playlist of proposed method

2 Related Work

Many researchers have researched for recommending music preferred by listeners from a vast collection of music.

Koren et al. [4] researched a method that estimates people with similar music preferences and suggests music using the listener's preferred music. Such a study focused on recommendation accuracy; therefore, the music suggested might be similar. Therefore, it may cause biased music people would listen to.

On the other hand, there are recommendation methods not focusing on accuracy. Herlocker et al. [5] pointed that many people can find items recommended by the usual method sufficiently. They also argued that researchers should evaluate a recommendation system based on indicators such as novelty, indicating that the recommended item is unknown to the listener, and serendipity, suggesting that the item is unexpectedly good for the listener. Actually, some researchers use these indicators and proposed such recommender systems [6][7][8][9], also reported their effectiveness on satisfaction and accuracy so on. With the advent of such recommendation systems, people can listen to a wider variety of music beyond their capabilities. However, it is not easy for listeners to express their preferences and select their favorite music among the many unknown pieces of music. In this research, we aim to eliminate the bias of the listened music by recommending them not by the system but by the people already familiar with them. We expected that the system could recommend music more effectively using people already familiar with the music.

There has also been much research on sales or recommendation methods by the hand of people. Luo et al. [10] conducted a survey that compared with sales income by human and chatbot. As a result, it revealed that customers trust humans more than chatbots. Nielsen Holdings Inc. [11] researched the reliability of advertisements by several media. The result showed that reliability of information from their friend is 90%, most reliable, and reliability of advertisement in video or on the banner is 30%, worst reliable. In addition, there have been studies focusing on general people. Bakshy et al. [12] found that general people are relatively more cost effective than influencers regarding marketing. Regarding this study, Cha et al. [13] suggest that non celebrity people can gain leverage by focusing on a single topic and making creative and insightful posts rather than simply conversing.

From these studies, we expected that information by general people could increase satisfaction than by machine in a recommendation.

3 Proposed Method and Its Effectiveness

3.1 Branching Playlist

We propose a method by creating a branch type of recommendation (see Fig. 1). The method enables people familiar with a particular genre to recommend music depending on the listener's preference by creating a branching playlist. For example, "if you like this music, listen to more maniac music" or "if you do not like this music, listen to more major music." The listener responds to the recommended music by liking it or not liking it, and the method then proceeds to suggest the next piece of music accordingly. Thus, our proposed method will provide listeners with many opportunities to encounter new music/genres and expand the range of music they will listen to.

A recommender creates a playlist with our method by adding a YouTube URL and a specific playback section. In addition, the recommender adds one or two next music continuously and chooses up to two music as the next music. Here, the recommender prepares one music for when a listener likes the previous music and another for when a listener does not like the previous music. If the recommender sets only one music as the next music, the method plays the next music regardless of the listener's preference.

In other words, it may be desirable for recommenders to create divergent playlists by estimating the transition of listeners' preferences. For example, if listeners evaluated the current music as not being to their liking, the recommender might be better set different atmosphere music from the current music.

3.2 Experiment & Results

We evaluated the effectiveness of branching playlists for the recommendation. We recruited two students as recommenders to create playlists (branching and non branching) for four genres. In addition, we recruited twenty four participants (in their 20s) to play the four playlists on a Web browser and rated each music. After playing the four playlists, we asked participants to answer a questionnaire about their level of satisfaction, familiarity, and interest on a 5 point scale (-2 to +2). We used these indicators because we expected the satisfaction could estimate the listener's comprehensive preference. The familiarity can evaluate whether they didn't feel something strange to the

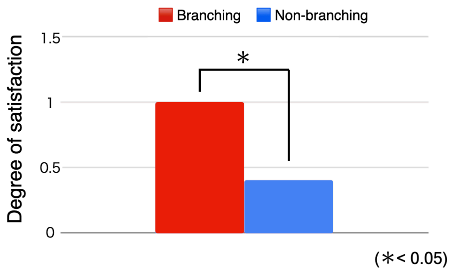

Fig. 2. Satisfaction level of each playlist

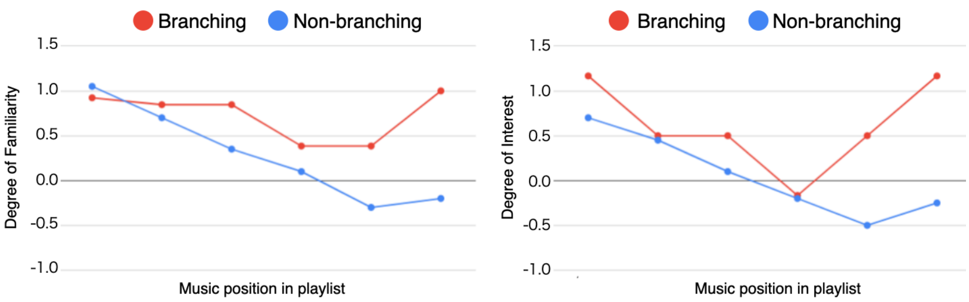

Fig. 3. Transition of evaluation value (left: familiarity, right: interest)

suggested music and the interest can assess whether they will prefer the music or not. We did not tell participants whether the playlist they were playing had a branching or non branching structure.

Fig. 2 shows the satisfaction level of the branching and non branching playlist. The satisfaction level was higher regarding the branching playlists. There was a significant difference between branching and non branching playlists (unpaired t-test, $p<0.05$). Moreover, Fig. 3 shows the familiarity and interest at a certain point in a playlist. For the non branching playlists, the familiarity and interest gradually decreased from the first music to the last. For the branching playlists, the familiarity and interest also gradually decreased but increased in the end.

The results indicate that a branching playlist is useful for increasing recommendation satisfaction, familiarity, and interest. However, the recommenders said it was burdensome to create such a playlist due to its complex structure. A reason may be that we did not prepare a tool for creating a branching playlist in this experiment. We also had only four branching playlists created. Therefore, we could only conduct a limited analysis. To address these issues, we need to develop a tool to help recommenders easily create branching playlists and share them. We also need to analyze more playlists by operating the system.

4 Prototype System

We implemented a prototype music recommendation Web service called "reco.mu" based on the proposed method. In addition, we conducted a detailed analysis of how recommenders create playlists and how listeners play them. In our implementation, we used JavaScript for the client side and MySQL and PHP to store the playlist and music information for the server side. In addition, we used Songle [14] to obtain music information for playback and playlist creation.

Here, we focus on the dialogue between a recommender and a listener. Then, we roughly classify the recommendation strategy into several categories. Therefore, we prepared several branching structures and made the recommender create branching playlists based on these structures. The reason for this is that we thought recommenders would be able to create playlists with more awareness of the order of music and branching structure through trial and error. Also, if they create branching playlists completely freely regardless of structure, it may increase the difficulty of creating because they would have more considerations. Therefore, we prepared eight different playlist structures to accommodate various recommendations in our prototype system (see Fig. 4).

4.1 Design of branching playlist

Fig. 4 shows the branching structures prepared in this study. As mentioned above, by focusing on an interactive dialogue, we determined that recommenders have unique strategies. Each playlist we prepared consisted of ten music to reduce the burden of creating playlists since it enables recommenders to consider only the music order. The following are the details of each playlist shape.

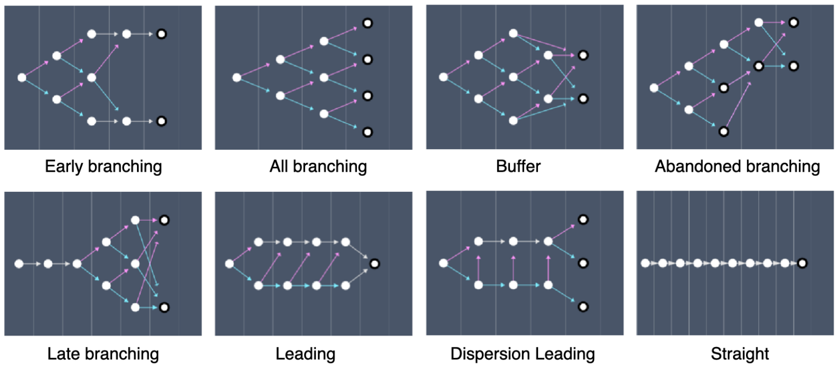

Fig. 4. Playlist structures

- **Early branching:** The first half of the playlist has a branching structure that allows the listener to listen to pieces of music that match the listener's preferences. The second half presents pieces of music that match the listener's preferences, making it easier for the listener to understand the appeal of the music.
- **All branching:** All stages have a branching structure designed to respond to the listener's preferences in more detail.
- **Buffered type:** Up to the third music is the same as in All branching, but the number of music presented in the final stage is narrowed to two so that the pieces of music recommended become more specific. This structure also reduces the number of quitters by providing a stage that serves as a buffer before reaching the last music.
- **Abandoned branching:** If a listener does not like the first music, this playlist will stop playing after a minimum of three pieces of music. In other words, if a listener is not interested in music, this playlist gives up to recommend.
- **Late branching:** The first three pieces of music must be listened to, and the subsequent music have a branching structure so that listener can listen to music adapted to the listener's preferences.
- **Leading type:** We designed this structure so that if the listener likes music at least once, the music played after will be those that the recommender wants to recommend strongly. We have narrowed down the final recommendation to one music so that the playlist plays the target music finally.
- **Dispersion leading:** As in the Leading type, if a listener judged music as a favorite once, the next music after that becomes fixed regardless of the listener's preference. By setting the number of music played in the end to three instead of just one, we aim to bring them closer to the listeners' preferences by considering the preferences of the recommenders while leading the music to be presented.
- **Straight:** This playlist structure is a conventional shape with no branching structure.

4.2 Usage

This system has two main functions: one is creating and editing playlists, and the other is to play those playlists.

The recommender first selects the playlist shape and then enters the playlist title, artist name, and other information. Then the screen moves to where the recommender can edit the playlist (left of Fig. 5). In addition, by selecting a node in the graph representing the shape of the playlist, the recommender can edit the current music, and by entering the URL of YouTube, the recommender can add the music.

The pink arrows point to the next music if a listener liked the current music, and the light blue arrows point to the next music if a listener did not like the current music. The gray arrows indicate that the music has only one next music, so the system played the same music regardless of the preference.

For playback, by selecting the playlist the listener wishes to view from the top page, the system will redirect to the screen where the listener can play the playlist (right of Fig. 5). While playing the playlist, the listener can use the bottom left button to evaluate whether or not the listener likes the current music being listened to, and the next music played will change depending on the listener's input. The listener can also skip tracks by clicking the bottom right button. In this case, we made the system to play the music at the end of the light blue arrow. This is because listeners who like to listen to the music are unlikely to stop playing in the middle of the music, and the skip button is likely to be used when the music is not to their liking.

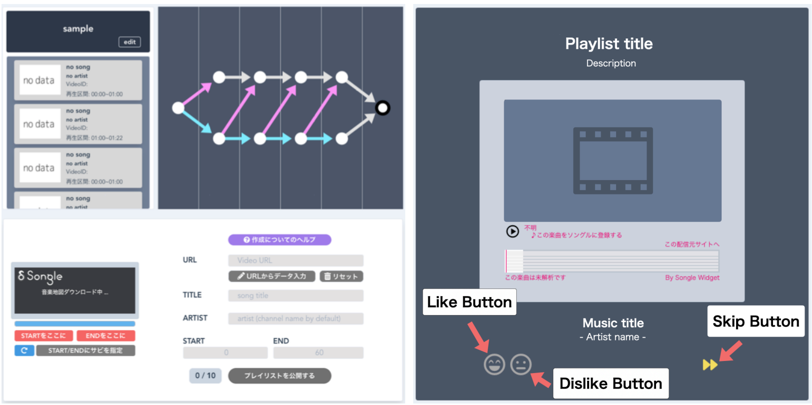

Fig. 5. Screen of prototype system
(left: screen for editing, right: screen for playing)

5 Results and analysis

We published "reco.mu" on July 19, 2020. Therefore, this study analyzed the operational results for about six months, from July 19, 2020, to January 13, 2021.

5.1 Creating playlists

We firstly asked our university students to create branching playlists and to answer several questions soon after the release. Also, we published reco.mu generally and recruited users to create and listen to playlists using Yahoo! Cloud sourcing [15] (conducted from November 17 to 20, 2020). In the request, we instructed participants to "create a playlist of your favorite genre or artist that you would recommend to others." As a result, 19 participants created 52 branching playlists and 32 non branching playlists. The structure breakdown of the branching playlists was 8 Leading, 8 Early branching, 10 Late branching, 11 All branching, 3 Dispersion leading, 7 Abandoned branching, and 5 Buffered.

For example, one participant created a playlist, "80's Western rock music", and its shape is Leading. As its name suggests, this playlist consists of 80's rock music. The first music on the playlist is "Under Pressure (Queen)." According to the survey, the recommender selects this music for the first because this music is famous. Moreover, it also revealed that this recommender selects other music based on the listener's preference.

5.2 Playing playlists

Listeners accessed the created playlists 1,217 times and played music in the playlists 8,374 times for all playlists. The number of playbacks was 5407 of the branching playlists and 2,967 for non branching. During playback, listeners could use buttons to input their preferences. Note that listeners could stop playing if they were not interested in the playlist. They also did not know the structure of the branching playlist before playback.

Fig. 6 shows the average percentage of listeners who played a playlist (30 branching playlists and 17 non branching playlists, excluding those we could not play due to system malfunction) up to a certain point. In this figure, the playback rate of each branch type is gradually decreasing because listeners may lose interest while listening to the

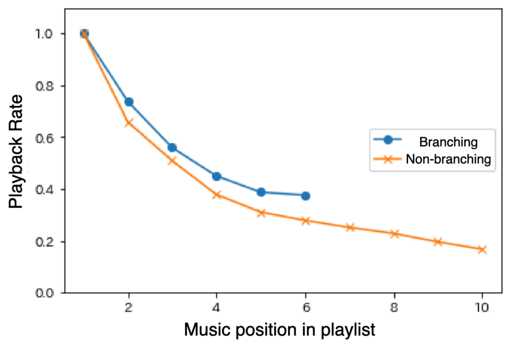

Fig. 6. Play rate for branching and non branching

playlist and end the playback. This figure mentioned about 40% listeners for the branching playlists, and about 20% listeners for the non branching playlists played until the last music.

Here, the length of the playlist differed depending on whether there was a branching structure. The length of branching playlists is from four to six. Therefore, we focused on the playback rate on the fourth, fifth, and sixth position, which is the same as in the branching playlist. We found that the branching playlists had a higher playback rate than the non branching playlists.

Fig. 7 shows examples of visualizing the branching playlists and how the listener played them. The thickness of the arrows indicates the number of times played, and the thicker the arrow, the more listeners played it. The pink arrows point to the next music if the listener liked the music, and the light blue arrows point to the next music if the listener did not like the music. The gray arrows indicate that the music has only one next music, so the system played the same music regardless of the preference. For example, on the left side of Fig. 7, more listeners rated the first music as their favorite since the first pink arrow is thicker than the blue arrow. The subsequent branches also tend to favor the favorite, indicating that the listener traced the playback in this playlist in the way the creator intended. However, on the right side of the figure, the arrows for the first two music are thicker, but the latter half is thinner, suggesting that this playlist may not have sufficiently attracted the listener. Thus, we can see that the listener behavior changes depending on the strategy and intention of the playlist creator.

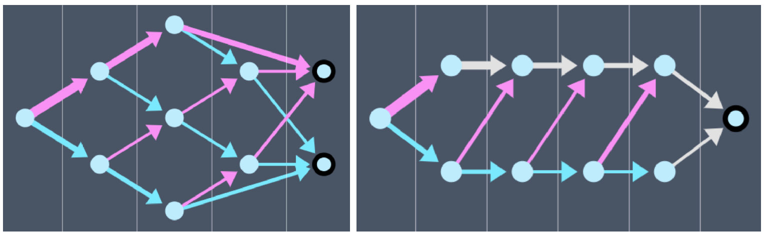

Fig. 7. Examples of how listeners played music

5.3 Results from questionnaires on playlist creation

We conducted a questionnaire survey for recommenders. We asked them how burdensome playlist creation was on a 5 point scale and how they created a playlist from free text. We received responses from recommenders of 36 branching playlists and 24 non branching playlists.

As a result, the average burdensome of creating a branching playlist is 3.08, the average burdensome of creating a non branching playlist is 2.38. Thus, branching playlists are more burdensome to create. We then looked at the descriptive results, which asked about strategies and opinions for creating either type of playlist. Most recommenders made good use of the branching structures to create their branching playlists. We obtained positive feedback, such as, “It was fun to think about the branching structure.” However, some recommenders responded regarding the burden of creating such playlists, such as “I was not used to thinking about how to place the music, and it was difficult.” A few recommenders who created non branching playlists said that they created with more ingenuity, such as “I tried to match the tempo of the music before and after.” However, many recommenders were not concerned with music order, such as “I put them in the order I like or thought of them.”

6 Discussion

6.1 Playlist creation and questionnaires

When creating a playlist to recommend to others, we expected the preferences and familiarity of the other person would be more important than focusing on one’s favorite music. Therefore, we investigated (1) whether there were responses on if the creator considered the preferences of listeners and (2) whether there were responses on if the creator emphasized his or her favorite music.

We found 20 recommenders of (1) for the branching playlists and 6 for the non branching playlists. Example responses are “I decided on two atmospheres of music in accordance with the listener’s preferences and divided playlist path into two.”, “I tried to make the first half of the playlist familiar and the second half more niche.” and so on. For the non branching playlists, all six responses were to the effect that “I first chose music that is popular with everyone then gradually chose less popular music.

We found 8 recommenders of (2) for the branching playlists and 14 for the non branching playlists. In the branching playlists, there were cases in which the respondents selected mainly their favorite music, such as “I selected my favorite music in the order of release date,” and there were few responses about the order and arrangement of the music. On the other hand, many recommenders responded that “I selected music mainly based on my favorites.” for the non branching playlists.

However, several responses such as “I created the playlist with live performance in mind” and “I placed the music in chronological order based on the anime” may not be fully understood without prior knowledge of the content. Standard non branching playlists are suitable for expressing time series so that recommenders may have created their playlists because of this. Alternatively, in music distribution services, playlists of music of a specific genre or artist are sometimes made public, such as “for beginners,” and we predicted that the playlists might have been created by simply compiling music of the same genre without considering the order.

As described above, adding a branching structure to a playlist makes it possible to expand creativity in creating playlists suitable for listeners. In other words, we believe that branching playlists can produce more suitable recommendations for the listener.

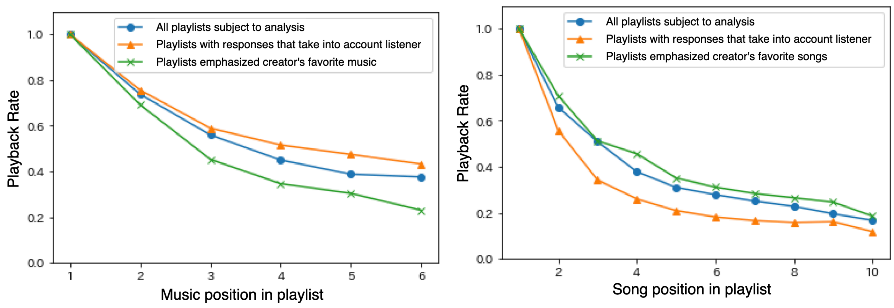

Fig. 8. Playback rate per survey
(left: branching, right: non branching)

6.2 Playlist playback

We analyzed the playback rate of both branching and non branching playlists by focusing on the responses to the questionnaire on playlists. The left of Fig. 8 shows the graphs of the playback rates of all branching playlists, those that considered listener preferences, and those that emphasized the recommenders' favorite music. When the recommenders created a playlist emphasizing their favorite music, the playback rate was higher in the branching playlists when the recommenders created a playlist with the listener in mind and lower. Therefore, it is essential to be aware of the person to whom they are recommending. This result may lead to recommendations that are more interesting to the listeners.

The right of Fig. 8 shows graphs showing the playback rates for all non branching playlists, those considering listener preferences, and those that emphasized the recommender's favorite music. This result indicates that playlists created by emphasizing the recommender's favorite music had high playback rates. In contrast, playlists created by taking into account listener preferences had low playback rates.

As mentioned in the previous subsection, all six responders considered listener preferences when creating their non branching playlists. For example, "I first chose music that was universally accepted then gradually chose lesser-known music." In a non branching playlist, the probability of not liking the next music when the listener moves to more maniac music may be higher than in a branching playlist because the next music to be played is fixed even if the listener evaluates his/her preference during playback. This gap in preference and familiarity between the recommender and listener may have caused many listeners to discontinue playback. On the other hand, playlists created emphasizing the recommender's favorite music had a higher playback rate. However, these responses are "I created with the artist's live performance in mind" or "I placed music in chronological order based on anime." Therefore, while the playlist may be enjoyable for people familiar with the artist or content, it may be unfamiliar and uninteresting for people unfamiliar with the content. These results indicate that creating branching playlists makes recommenders consider listener preferences for more interesting recommendations.

7 Conclusion and Future Work

Only a few pieces of music are known to the public, which causes disparity in popularity and recognition. Therefore, we proposed a method of creating branching playlist that enables recommenders to recommend the next music depending on the listener's preferences. We evaluated the method's effectiveness, and the results indicate that recommendations through branching playlists may improve interest and familiarity. We implemented a prototype Web based music recommendation system called "reco.mu" based on the proposed method and investigated the characteristics of branching playlists and how recommenders create them. We found that although branching playlists are more burdensome to create than non branching playlists, they may enable recommenders to be more conscious of listener preferences. It was also shown that such branching playlists are more likely to attract listeners' interest than those that reflected the recommender's preferences.

In the future, we plan to implement a function that allows multiple recommenders who are fans to collaborate to create a playlist. This collaboration is expected to increase motivation and reduce the psychological burden of the creation process. Furthermore, by incorporating the opinions of others, a recommender can consider the branching structure from multiple perspectives, which may make it possible to make playlists that are easy and familiar to more people.

References

1. Apple Music, https://www.apple.com/jp/apple-music/, last accessed 2021/8/13.
2. Spotify, https://www.spotify.com/jp/, last accessed 2021/8/13.
3. Mulligan, M.: The Death of The Long Tail : The Super Star Music Economy. MIDiA Consulting (2014).
4. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer, vol. 42, no. 8, pp. 30-37 (2009).
5. Herlocker, J.H., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), vol. 22, no. 1, pp. 5-53 (2004).
6. Li, P., Que, M., Jiang, Z., Hu, Y., Tuzhilin, A.: PURS: Personalized Unexpected Recommender System for Improving User Satisfaction. Fourteenth ACM Conference on Recommender Systems, pp. 279-288 (2020).
7. Goto, M., Goto, T.: Musicream: Integrated Music-Listening Interface for Active, Flexible, and Unexpected Encounters with Musical Pieces. Journal of Information Processing, vol. 17, no. 17, pp. 292-305 (2009).
8. Menk, A., Sebastia, L., Ferreira, R.: CURUMIM: A Serendipitous Recommender System for Tourism Based on Human Curiosity, 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 788-795 (2017).
9. Ge, M., Delgado-Battenfeld, C., Jannach, D.: Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity. Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 257-260 (2010).

10. Luo, X., Siliang, T., Zheng, F., Zhe, Q.: Frontiers: Machines vs. Humans: The Impact of Artificial Intelligence Chatbot Disclosure on Customer Purchases. Marketing Science, vol. 38, no. 6, pp. 937-947 (2019).
11. Nielsen Holdings plc: GLOBAL ADVERTISING CONSUMERS TRUST REAL FRIENDS AND VIRTUAL STRANGERS THE MOST, https://www.nielsen.com/us/en/insights/article/2009/global-advertising-consumers-trust-real-friends-and-virtual-strangers-the-most/, last accessed 2021/8/13.
12. Bakshy, E., Hofman, J. M., Mason, W. A., Watts, D. J.: Everyone's an Influencer: Quantifying Influence on Twitter. Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, no. 10, pp. 65-74 (2011).
13. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.: Measuring User Influence in Twitter: The Million Follower Fallacy. Proceedings of the International AAAI Conference on Web and Social Media, vol. 4, no. 1, pp. 10-17 (2010).
14. Goto, M., Yoshii, K., Fujihara, H., Mauch, M., Nakano, T.. Songle: A Web Service for Active Music Listening Improved by User Contributions, Proc. of ISMIR 2011, pp. 311-316 (2011).
15. Yahoo!, https://crowdsourcing.yahoo.co.jp/, last accessed 2021/8/13.