

Music Recommendation for Individual Music Preference

Hiroto Shinohara and Kazunori Mizuno

Abstract—Over the last several years, music streaming services have come in handy in our lives. Apple Music, Spotify, and Google Play Music is one of the most commonly used music streaming services. There are a number of studies about music recommendation system, one of the functions in music streaming services. Most of studies about music recommendation system express music features using music information extracted from song components. The way to express music features and to come up recommendations out of music features varies.

In this paper, we consider couple of approaches that reflect users music preference including music features and construct our music recommendation model through those approaches. Our proposed method is to recommend music with user preference vector, which has users music preference, referring to the idea of contents-based filtering.

Index Terms—Personalized recommendation, feature extraction, music representations, weighting, affinity discovery.

I. INTRODUCTION

In recent years, development of the Internet makes it easy to upload and access to music on the Internet. Speaking of uploading your music, a few efforts get you to upload your music and there is no need to having to get signed to music labels to release your music like before. If you are in an environment that the Internet is available, there is nothing huge to share your music. Accessing to music is much easier than uploading. All you need to do is make your own account on any music streaming services and the rest is just to enjoy music you want to listen. Since this is easy for creators and listeners to do what they do, the number of songs and streamings is getting huge and there is much data on the Internet. That makes listeners to have a lot of choices of songs to listen.

In order to keep listeners exited and for them to enjoy music that music streaming services provide, music streaming services compete with others by making differences with audio quality, prices, original playlists, availability of lyrics, exclusive radio station, size of music library, and so forth. Additionally, there is a function called music recommendation system as a part of music streaming services. Music recommendation system predicts users preference and recommend songs users would like based on users history.

To construct music recommendation system, there are major methods called collaborative filtering and contents-based filtering. Those 2 filterings are music recommendation models which plays core roles to perform quality music recommendation. Each has their distinct

features to filter songs to accomplish recommendation users prefer. However, there is an advantage and disadvantage on both filterings. At this point, we wonder there must be better way to recommend songs covering those disadvantages.

In this paper, we proposed music recommendation model that reflect users music preference. Our basic strategies are to calculate song vectors and user preference vector defined in Section III-B to compare each other to see the similarity value between 2 vectors to recommend high similarity value songs to users.

For the rest of this paper, we describe types of music recommendation in Section II. Our proposed method is described in Section III with tools used in our method. In Section IV, we explain our experiment settings and result. Finally, we wrap up this paper and describe future work in Section V.

II. MUSIC RECOMMENDATION SYSTEM AND MUSIC FEATURES

There are a lot of ways to construct music recommendation model and to express songs by using features extracted song components [1]-[4].

In this section, we describe music recommendation system in Section II-A and music features in Section II-B by picking up some methods used in music recommendation system and showing how music features are presented. Also, problems on music features in Section II-C that are presented by Lu, Cheng-Che and Tseng, S. Vincent [5] are discussed.

A. Music Recommendation model

Music recommendation system has been used as one of the functions in music streaming services. As an essential part of Music recommendation system, Music recommendation model analyzes songs that users listened to in the past, finds out songs that users might like. And those songs found by recommendation model are recommended by recommendation system, outer layer of recommendation model. Here are major methods used as music recommendation model.

Collaborative filtering and contents-based filtering are one of the major methods to conduct recommendation. Collaborative filtering is based on collecting and analyzing a huge amount of data about user's behaviours, activities, and preference on certain genre of services to predict something they would like to refer to users that have similar behaviours. Different from contents-based filtering, collaborative filtering does not need any understanding of contents themselves. Collaborative filtering on music recommendation services is a method to predict songs that are given good ratings by a user that has similar music preference to the one to recommend songs. The problem on

Manuscript received January 22, 2019; revised March 13, 2019.

H. Shinohara and K. Mizuno are with Takushoku University, Hachioji, Tokyo 193-0985 Japan (e-mail: shinoharahiroto@gmail.com, mizuno@cs.takushoku-u.ac.jp).

collaborative filtering is to have a tendency to predict only popular songs. That is because popular songs are easy to show up on a lot of users song lists.

Contents-based filtering is a method to predict songs that is similar to a song that users liked in the past. To be detailed, contents-based filtering compares a similarity value of a song and user's favorite song directly and predicts a song whose similarity value to user's favorite song is high. Usually, contents-based filtering uses an approach that songs are represented by a set of discrete attributes and features. By representing songs as discrete values, songs are able to be compared to measure their similarity values. However, the problem on contents-based filtering is that contents-based filtering only can predict songs similar to a single favorite song from users.

B. Music Features

Lu, Cheng-Che and Tseng, S. Vincent [5] get music information from musical scores. Music features are represented by a vector by extracting music features from music components and defined in dimensions in a vector. Following 11 types of features are defined as vector dimensions.

- Maximum Interval
- Maximum Pitch
- Minimum Pitch
- Average Relative Pitch
- Average Absolute Pitch
- Average Tempo
- Rhythmic Speed
- Original Key
- Chord Density
- Key Density
- Meter Density

C. Problems on Music Features

There are 2 problems on music features in Section II-B.

The first problem is that how scattered pitches are is not considered. Without knowing the degree of scattering, it is hard to compare the difference of values in Average Relative Pitch and Average Absolute Pitch with the 2 values in other songs.

The second problem is that there are dimensions of music vector whose value is extremely big or small compared to one to another when a song is vectorized. Regardless of users music preference, the influence that dimensions of music vector whose value is extremely big give to choosing recommendation songs is huge.

III. PROPOSED METHOD

In our paper, we conduct music recommendation to which our own method is introduced based on improving problems on contents-based filtering pointed out in Section II-A.

To accomplish our method, we construct 2 types of vectors: vectors that represent music feature and vectors that represent users music preference. By calculating a similarity value between these 2 types of vectors, we improve our music recommendation to users.

In Section III-A, we describe our basic strategies using 2

types of vectors and points that our proposed method improves the problem on contents-based filtering. We define components used in our method in Section III-B. 2 types of vectors, weight, and additional music features are described. Similarity calculation between 2 vectors, song vector and user preference vector, are defined with formulas used in different situations in Section III-C. Section III-D shows workflow of our method.

A. Basic Strategies

Our proposed method is a type of contents-based filtering our idea is introduced to. However, the difference from contents-based filtering is that we have an extra step to conduct recommendation, which is comparison with user preference vector. Our basic strategies are to calculate song vectors and user preference vector to compare each other to see the similarity value between 2 vectors to predict high similarity value songs.

When it comes to regular contents-based filtering, recommended songs are decided by comparing 2 songs, the one is user's favorite song X and the other is one of the bunch of songs Y . If similarity value of X and Y is high, Y would be a recommended song for a user.

As pointed out in Section II-A, contents-based filtering cannot predict songs that include features of multiple favorite songs of users. It only can predict songs from a single favorite song of users. In other words, recommendation songs only have features from a single favorite song from users. To be a solution to this problem, we constructed extra filter called "user preference vector". User preference vector consists with multiple favorite songs from users so that it can recommend different sorts of songs at once. For example, contents-based filtering would simply predict rock songs if user favorite song is rock. Meanwhile, if user 2 different favorite songs are rock and pop, our method would predict songs that have a mixture of rock and pop, which contents-based filtering cannot.

B. Definitions

First of all, we calculate vectors that have each music feature, mentioned in Section II-B, on dimensions of vectors. We call this vector "song vector". Also, we call vectors that reflect users music preference "user preference vector". User preference vector has the same dimensions to song vector and is calculated taking means of values of dimensions in song vectors whose song is highly rated by a user.

"Weight" is a value that represent a correlation between users ratings on songs and each dimension in song vector. Dimensions that each user dignifies and does not are figured out by calculating a correlation between user ratings on songs and each dimension in song vector. We use the advantage of weight whose influence to music recommendation gets bigger as absolute value of correlation coefficient is close to 1 and smaller as absolute value of correlation coefficient is close to 0. Weight is calculated as the following formula:

$$w_{i,r} = \frac{cov(i,r)}{\sigma_i \sigma_r} (-1 \leq w_{i,r} \leq 1) \quad (1)$$

where $w_{i,r}$ denotes a correlation coefficient between the i th

dimension and user's rating r , $cov(i, r)$ denotes a covariance between a value of the i th dimension and user's rating r ; and σ_i denotes a standard deviation of a value of the i th dimension and user's rating r .

There are 2 problems picked in Section II-C. The first problem is that how scattered pitches are is not considered. The second problem is there are dimensions of music vector whose value is extremely big and small compared to one to another. The solutions for these 2 problems are given below. To solve the first problem, we add 4 music features as dimensions that represent standard deviation and variance of pitches.

- Maximum Standard Deviation of Pitch
- Minimum Standard Deviation of Pitch
- Maximum Variance of Pitch
- Minimum Variance of Pitch

These 4 dimensions about pitch have 2 types: Standard Deviation and Variance. Standard Deviation of Pitch has 2 types: Maximum and Minimum. Also, Variance of Pitch has 2 types: Maximum and Minimum. Both Variance and Standard Deviation of Pitch counts on tracks on a song. There are songs that consist of a single track but most of songs consist of multiple tracks. Since each track has the different number of pitch and pitch positions, it is hard to represent a single variance and standard deviation of pitch as an entire song. Therefore, Maximum Standard Deviation of Pitch is the value of a track that has Maximum Standard Deviation of Pitch of all tracks in a song if there are multiple tracks. And Standard Deviation of Pitch, which has the highest value of all tracks, of the track represents Maximum Standard Deviation of Pitch of a song. The same process can be applied to other 3 music features.

To solve the second problem, we introduce song vector normalization. Normalization is one of the methods to rescale data into the range of 0-1. In our method, the data to be rescaled into 0-1 is the values of each vector dimension. We normalize song vector so that the maximum number among values of each dimension in music vector is 1, the minimum number among values of each dimension in music vector is 0.

C. Similarity Calculation between Vectors

To pick up recommendation songs to users, we need to know how to predict recommendation songs. To give every music object in our music dataset [6] similarity value with user preference vector, we compute the similarity value S between song vector and user preference vector using the following formula.

$$S(M, A) = \frac{\sum_{i=0}^{n-1} |m_i - a_i|}{n} \quad (2)$$

where M denotes song vector to be compared to user preference vector; A denotes user preference vector; n denotes the number of dimensions of song vector; m_i denotes the i th dimension of M ; and a_i denotes the i th dimension of A .

If weight is considered in the similarity calculation between song vector and user preference vector, the similarity value is calculated with *weight* (1) applied to the formula (2).

$$S(M, A) = \frac{\sum_{i=0}^{n-1} |w_{i,r}| \times |m_i - a_i|}{n} \quad (3)$$

D. Workflow

Workflow in Fig. 1 shows a procedure of recommending songs to users in our method. First, we gain users music preference. Users start and listen to N songs and rate songs they listen to. User preference vector is calculated based on user ratings. Next, similarity value S between song vector and user preference vector is calculated following formula (2) in Section III-C. Finally, songs that similarity value to user preference vector is high are recommended.

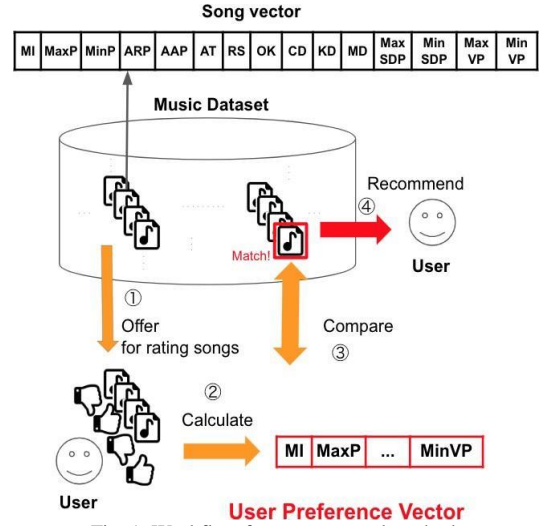


Fig. 1. Workflow for our proposed method.

IV. EXPERIMENTS

As described in Section III-B, 2 parameters, song vector normalization and weight, are introduced in our proposed method. We conducted experiments to figure out which combination is the best by switching on and off the 2 parameters.

TABLE I: COMBINATIONS AND RESULT

| Normalization | Weight | Average Point |
|---------------|--------|---------------|
| O | O | 3.13 |
| O | X | 3.00 |
| X | O | 3.25 |
| X | X | 3.29 |

A. Experimental Settings

To conduct our experiments, users get 10 songs to listen from our music dataset [6] and rate 10 songs with rating range 1-5 to gain users music preference. The bigger the rating number is, the more the song represents users music preference. After getting ratings of 10 songs, user preference vector to be compared to song vector is calculated taking means of values of dimensions in song vectors whose songs are given to users and rated as more than 4 by users.

Next, we recommend songs to users with 4 combinations including song vector normalization and weight shown in

Table I. Users listen to 3 songs that have the highest to third highest similarity value per a recommending combination. Thus, users listen to 12 recommended songs in total. Again, users rate 12 recommended songs with rating range 1-5 in the same way users do with the first 10 songs they are given. The way to decide the best combination is to divide sum of means of ratings per combination with the number of samples. We call this value “average point”. The combination with the biggest average point is the best combination of 4 combinations.

In this experiment, the number of samples are 21.

B. Experimental Result and Discussions

As a result of our experiment in Section IV, we have gotten a result shown in Table I.

Couple of factors why the combination with both weight and vector normalization could not exceed the average point of combination without both of them can be considered. In terms of weight, since this experiment only give users 10 songs to gain their music preference, it can be hard to find music features users dignify. For example, even if there is a strong correlation coefficient between users rating on songs and each dimension in song vector, it might happen to be strong and not accurate enough because of the small number of songs to give to users, which is just 10.

On the other hand, what was wrong on vector normalization is that the way a vector is normalized. Even though there is a huge amount of songs in this world that not even major music streaming services have not licensed to provide as services yet, a vector is normalized in a way that the maximum number among values of each dimension in music vector is 1, the minimum number among values of each dimension in music vector is 0 with the size of our music dataset of about just 110,000. Within about only 110,000 songs, a value of each dimension of a song can be not accurate enough to measure the true value of each of a song. If a vector is normalized with music database bigger than ours in this paper, values of normalized vectors would become different from what we have right now. The reason is because a value of each dimension of a song is set for the certain size of music library. Every time the size of music library changes, a value of each dimension of a song has to be recomputed.

With the result we have gotten, we are going to compare the accuracy of music recommendations between our method with contents-based filtering. Our method to compare with contents-based filtering has no weight or vector normalization as parameters because of the result of average point. Here is a factor that our method can exceed average point of contents-based filtering as a possibility. The advantage our method has and contents-based filtering does not have is user preference vector. User preference vector itself contains music features that users like from multiple songs. The more users have favorite songs in their song list, the closer user preference vector approximates to their desired songs. On the other hand, if users picked up only a song as a favorite song, user preference vector has

the only music features that the single favorite song users picked. This is the same method that contents-based filtering uses. To sum up the difference between our method and user preference vector, our method has an advantage that user preference vector is able to contain music features from multiple favorite songs, which contents-based filtering only contains music features from a single favorite song.

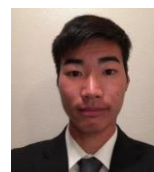
V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed music recommendation that reflect users music preference. Our basic strategies are to calculate song vectors and user preference vector to compare each other to see the similarity value between 2 vectors to recommend high similarity value songs to users.

As our future work, we are going to compare the accuracy of music recommendations between our method with contents-based filtering. And then, we are going to test out the significant of average point. Since our method has an advantage that user preference vector is able to contain music features from multiple favorite songs, there is a possibility that our method can exceed average point of contents-based filtering.

REFERENCES

- [1] M. Kaminskas and F. Ricci, “Contextual music information retrieval and recommendation: State of the art and challenges,” *Computer Science Review*, vol. 6, pp. 89-119, 2012.
- [2] C. D. Corrêa and F. A. Rodrigues, “A survey on symbolic data-based music genre classification,” *Expert Systems with Applications*, vol. 60, pp. 190-210, 2016.
- [3] C. Guan, Y. Fu, X. Lu, E. Chen, X. Li, and H. Xiong, “Efficient karaoke song recommendation via multiple kernel learning approximation,” *Neurocomputing*, vol. 254, pp. 22-32, 2017.
- [4] M. Shan, F. Kuo, M. Chiang, and S. Lee, “Emotion-based music recommendation by affinity discovery from film music,” *Expert Systems with Applications*, vol. 36, pp. 7666-7674, 2009.
- [5] C. Lu, S. V. Tseng, “A novel method for personalized music recommendation,” *Expert Systems with Applications*, vol. 36, pp. 10035-10044, 2009.
- [6] The largest midi collection on the internet. [Online]. Available: https://www.reddit.com/r/WeAreTheMusicMakers/comments/3ajwe4/the_largest_midi_collection_on_the_internet/



H. Shinohara was born in Japan, 1996. He is currently an undergraduate student of Takushoku University, Japan. His major field of study is data mining.



M. Kazunori received the B.Sc, M.Eng and Ph.D in engineering from University of Tsukuba, Japan, in 1996, 1998 and 2001, respectively. Since 2017, he is a professor at Department of Computer Science, Takushoku University, Japan. His research interests include knowledge processing, combinatorial search algorithms, constraint satisfaction, evolutionary algorithms and multi-agent simulation.