A Music Recommendation System Based on Network

Ryan Deng, Xinyi Zhao, Yue Ma
School of Information
University of Michigan
Ann Arbor, MI 48109

1. Motivation and Research Questions

Popular music is a kind of music with wide appeal that is distributed to large audiences, which is an important cultural expression that captures the listener’s attention for many years (Serrà et al., 2012). We can hear popular music on the radio, on TV, on mobile phones...at any time and any place, as a kind of spiritual “food”, popular music has been integrated into our daily lives. There are also many styles in popular music; in the meanwhile, the listeners’ preference also represents the style and development of popular music in different ages.

In this era of the Internet, popular music has also been greatly developed, with more and more ordinary musicians publishing their works on the Internet platform, and new popular musics born almost every day; besides, the taste of the audience is no longer limited to their favorite singers. Based on this background, we want to set up a music recommendation system based on the song network to help the audience to find more music that suits their taste.

During the process to build the recommendation system, we are curious about 1) what characteristics of the song network are based on the user’s playing history and 2) whether the network can meet the basis to build a recommendation system – they are exactly our research questions.

2. Related Work

In fact, the music recommendation system has been applied in real life. For example, NetEase Cloud Music, a popular mobile app in China, has the function of "Heartbeat Mode", which recommends songs that users may like according to the song currently playing. In addition, many scholars have studied music recommendation systems and tried different models. In research from Vinothini Kasinathan's team, they proposed a music recommendation system based on fuzzy logic, which makes decisions on music recommendation based on users' music listening habits, music genres, and their impact on human beings (Kasinathan et al., 2019). Besides, the research of Byeong-jun Han's team proposed a context-based music recommendation (COMUS) ontology to model users' music preferences and contexts, and to support reasoning on users' expected emotions and preferences (Han et al., 2009). What these two researches have in common is that they put more emphasis on the characteristics of the music itself, which gives us a lot of inspiration.

3. Data

3.1 Dataset

We use the dataset called Million Song Dataset[1], one of the publicly available large-scale collections, which has been used and analyzed by all kinds of music processing technologies (Bertin-Mahieux et al., 2011). It is a collection of audio features and metadata for a million contemporary popular music tracks. It provides a good resource to evaluate a music recommendation system, which includes the full listening history for more than one million users. The two used files in this project are Song Dataset and Taste Profile Dataset.

- Song Dataset (subset-compiled.csv[2]): as the original data from Million Song Dataset directly is very large, approximately 500 GB, we use the subset exposed on GitHub for this project. The subset consists of 10,000 songs, about 1% selected at random from the original data.

- Taste Profile Dataset (train_triplets.txt[3]): Million Song Dataset also provides the taste profile data, which contains 1,019,318 unique users, 384,546 unique songs, and 48,373,586 records of triplets (user id, song id and play count). This is an important basis for building the network.

3.2 Data Collection and Preprocessing

We download the two datasets directly. Firstly, we use *Pandas* library to read two datasets as dataframes. Because the taste profile dataset contains too much data, about 3 GB, we then use the songs in the song dataset as a limiting condition to filter the taste profile dataset, and only keep the songs co-existing in the two datasets. As a result, there are 3,675 songs kept in the song dataframe and 772,661 user-song-play_count triplets kept in the taste profile dataframe. Finally, for convenience and saving time, we export the two filtered dataframes as csv files, called cleaned-subset.csv and train_triplets_sample.csv respectively.

4. Methods

4.1 Exploratory Data Analysis

We use the groupby function and matplotlib library to see the distributions of the number of users for each song, and the number of songs played by each user, in the taste profile dataframe. The song with the largest number of users has 30,117 users; the user who played the largest number of songs played 48 songs, and the distributions are shown in figure 1. Both two

[1] http://millionsongdataset.com

[2] https://github.com/subha5gemini/MillionSongDataset/blob/master/subset-compiled.csv

[3] http://millionsongdataset.com/sites/default/files/challenge/train_triplets.txt.zip

distributions seem to follow the power-law distribution. This result means that the largest connected component of the network would contain the most part of it. Thus, if we want to randomly select a small fraction to do our analysis, the network might be sparse. In addition, it tells us that people prefer to select the most popular songs which only take a very small proportion of all the songs.

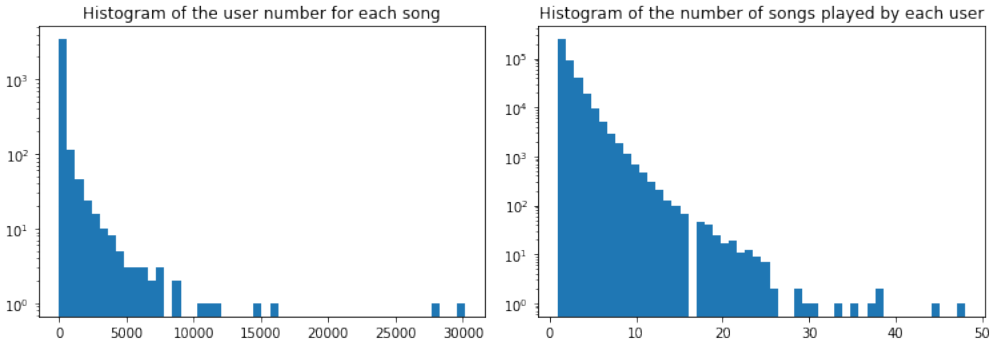

Figure 1

4.1 Network Building

In the taste profile dataset, the relationship between users and songs could be considered as a bipartite network with the users' group and the songs' group. Based on this assumption, we want to build a one-mode projection network graph of the songs. Any two songs in each user's song list can be considered connected, which means that if both songs have been played by at least one person in the dataset, they are considered connected. By using all the song pairs, along with the number of users playing these songs together as weights, we build the song network, and the graph is shown in figure 2.

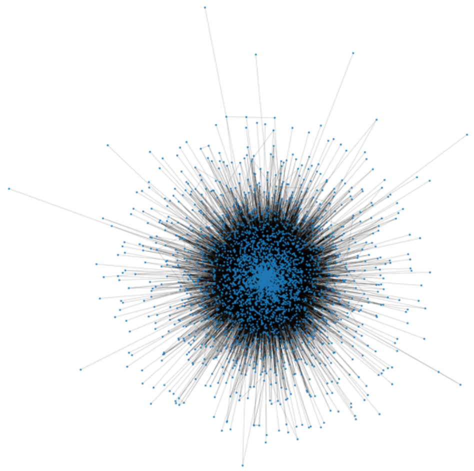

Figure 2

The song network has 3,535 nodes and 201,205 edges, with an average degree of 113.84. Besides, there is only one connected component in the network, so the largest connected component is the network itself. There is a problem here that 3,675 songs are kept in both two data datasets, but there are only 3,535 nodes (songs) in the song network. We find that the user who played these 140 songs has not played other songs. Therefore, these 140 songs cannot be connected to the network.

4.2 Small-World Network Analysis

To do the small world analysis, we have to build a small-world network model first with the same or similar features as our song network. As the average degree of the song network is 113.84, we round it to 114. Then, we build a Watts-Strogatz model graph with 3,535 nodes, 114 nearest neighbors in the ring topology and 0.2 probability of rewiring each edge, and the graph is shown in figure 3.

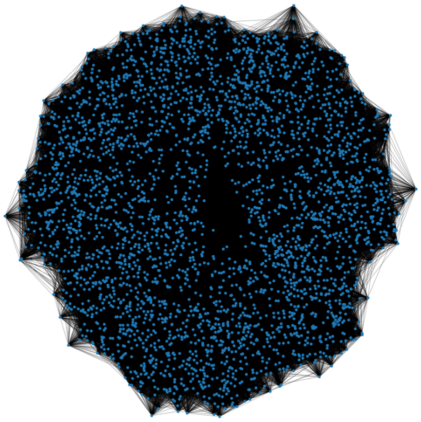

Figure 3

Then, we compare the average shortest path length and average clustering coefficient of two graphs. Watts-Strogatz small-world network model usually has the attributes of a small average shortest path length and a large average clustering coefficient. From figure 4, we can see that our song network has an average shortest path length similar to the Watts-Strogatz model graph, but even has a much larger average clustering coefficient. Thus, we think our song network fulfills the conditions of a small-world network.

	Average Shortest Path Length	Average Clustering Coefficient
Song Network	2.2898	0.6072
Watts-Strogatz Graph	2.2050	0.3888

Figure 4

4.3 HITS & PageRank & Degree Centrality

We use three algorithms to calculate the importance of each node (song) in the song network. Since our song network is undirected, it can be regarded as a two-way directed network in some algorithms

- **Hyperlink Induced Topic Search (HITS):** The HITS algorithm can analyze the importance of nodes by calculating Hub scores and Authority scores. The Hub estimates the node value based on outgoing links, and Authority estimates the node value based on the incoming links.

- **PageRank (PR):** The PageRank algorithm regards the link to the node as a vote, indicating the importance, and ranks the node accordingly.

- **Degree Centrality:** The Degree Centrality algorithm is a metric to analyze the centrality of nodes. The greater the degree of a node, the higher the degree centrality of the node, and the more important the node is in the network.

From the distributions of Hub and Authority, PageRank and Degree Centrality shown in figure 5, we see that all of them follow the power-law distribution, which means that a small fraction of nodes plays much more important roles in the network than others.

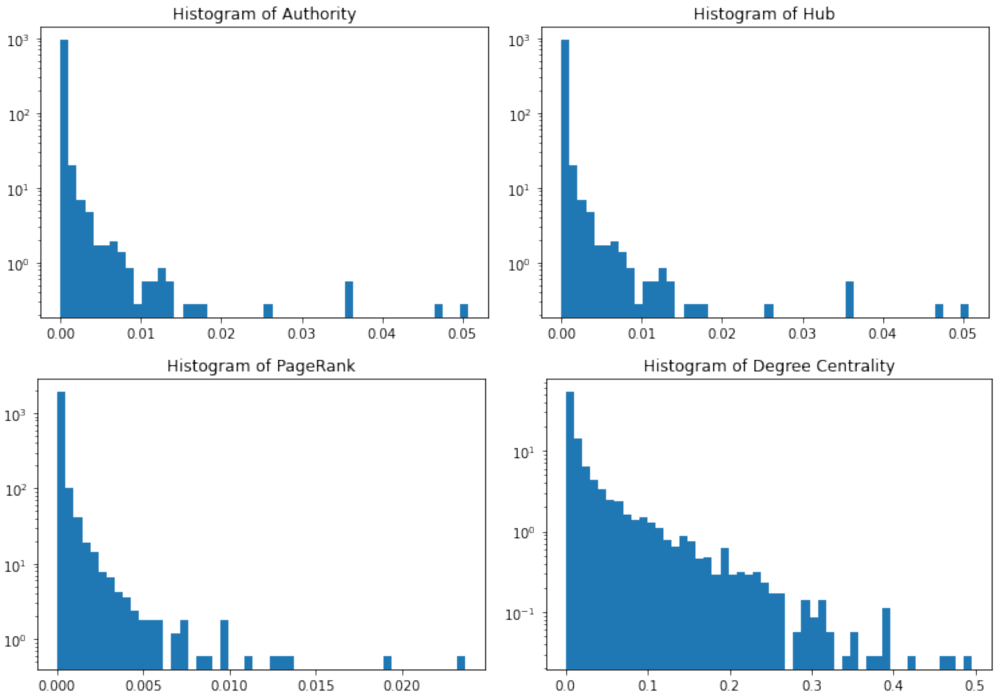

Figure 5

We assume that in real life, the importance of a song is determined by the play counts and popularity. Therefore, we use two features, “song_play_count” from the taste profile dataset and “song_hotness” from the song dataset, as the reference to evaluate the accuracy and performance of building the recommendation system by the HITS algorithm, the PageRank algorithm and the Degree Centrality algorithm.

We first sort the songs by play count, song hotness respectively, and sort the songs according to the Hub score, the Authority score, the PageRank score and the Degree Centrality score respectively. Then we compare the Jaccard Similarity Coefficients between the list sorted by the four scores and the list sorted by play count or song hotness, and the result is shown in figure 6.

The result shows that the PageRank algorithm has the best performance in our project. Thus, we decide to use the PageRank algorithm to build our recommendation system.

Jaccard Similarity	Hub	Authority	PageRank	Degree Centrality
Play Count	0.06	0.06	0.1	0.04
Song Hotness	0.0	0.0	0.02	0.02

Figure 6

5. Results

5.1 Make Recommendation by the Song Network

For each user, we first find the list of songs that have been played by the user, and then find all the songs in the song network that are connected to the user's songs list. This means that we want to find other persons in the dataset who have played at least one song in this user's songs list, to see what else songs they have played, and to recommend these songs to this user. The recommendation preference is based on the PageRank scores. Finally, we select the related top 10 songs with the highest PageRank scores for each user.

5.2 Make recommendations by Collaborative Filtering

To evaluate the quality of the recommendation result by network approach, we implement the commonly used method, Collaborative Filtering, to build the recommendation system, in order to make a comparison with the network method. The original dataset is split into train set and test set and the play count values are considered as rating scores, which are normalized to 0~5. Then the SVDpp model of the *Surprise* library is used to fit the train set, and then the play count of the test set is predicted. The accuracy of this model has been calculated by Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE), and the result is that the RMSE score is 0.5079 and the MAE score is 0.2033, which means that this algorithm works well, and the song recommendation system can be created based on the highest values of the predicted play count for each user.

Additionally, we randomly select 5,000 users and organize the song recommended system to each user by network method, by collaborative filtering, and by the original user history (the baseline), then calculate the mean play count of these songs from the three groups. The T-test is used to analyze the difference of the play count values between the network method and the collaborative filtering or between the network method and the user history.

The result shows that: from user history, the mean play count value of the songs played by each user is 14,664; the mean play count of the songs recommended for each user with the network method is 36,792, and the mean play count of the songs recommended for each user with the collaborative filtering method is 14,447. Based on the p-values of the t-test, we can see that there are significant differences between the network group and the user history group, as well as the collaborative filtering group. However, the difference between the collaborative filtering group and user history group is not significant. Thus, it indicates that the network recommendation method works well for this dataset.

6. Challenges

The most challenging part is that we don't have any ground-truth data to evaluate our recommendation system. Thus, we make some assumptions, like "in real life, the importance of a song is determined by the play counts", to convert the features to the basis to be evaluated. Also, we implemented the Collaborative Filtering method, and compared it with our song network recommendation system to make our results more convincing.

Besides, there are 201 users (0.05% of the sample set) who only played one song, and these 140 songs are only played by these users, so these songs are not in our song network, causing us can't make recommendations for these users. The reason is that our sample set limits a range of songs, so some users may have extra songs that are not on our song list. If we expand the song list, we will have more user-song-play_count triplets, so that this problem can be solved.

7. Conclusion & Further Consideration

In this project, we explore two questions:

1. What are the characteristics of the song network based on the user's playing history?
2. Can the song network meet the basis to build a recommendation system?

After we convert the dataset from a bipartite network to the one-mode projection network of the songs, we compare it with a Watts-Strogatz small-world network model and conclude that the song network conforms to the characteristics of the small-world network. What's more, we calculate the HITS scores (Authority and Hub), PageRank score and Degree Centrality scores for each node of the song network, and we find the PageRank algorithm has the best performance in this project to recommend songs.

We successfully establish the recommendation system based on the song network, by looking for the connected songs in the network graph to the songs that have been played by each user, then the connected songs with the highest PageRank value will be recommended to the user. The play count values of the user history are used to evaluate the network recommendation, and the commonly used collaborative filtering recommendation system is also used to compare with the network method. Our results show that the network recommendation result is better than the baseline as well as the collaborative filtering method in this project.

Only using PageRank score to recommend songs has certain limitations, and some of our assumptions may have certain loopholes, so in the future, we consider contacting Million Song Dataset Website to see if we can get some ground-truth data so that we can explore whether the score with different weights of node importance algorithm can improve the accuracy of the recommendation system.

Reference

Kasinathan, V., Mustapha, A., Tong, T.S., Rani, M.F., & Rahman, N.A. (2019). Heartbeats: music recommendation system with fuzzy inference engine. *Indonesian Journal of Electrical Engineering and Computer Science, 16*, 275-282.

Han, B., Rho, S., Jun, S., & Hwang, E. (2009). Music emotion classification and context-based music recommendation. *Multimedia Tools and Applications, 47*(3), 433–460. https://doi.org/10.1007/s11042-009-0332-6

Serrà, J., Corral, Á., Boguñá, M., Haro, M., & Arcos, J. L. (2012). Measuring the Evolution of Contemporary Western Popular Music. *Scientific Reports, 2*(1), 521. https://doi.org/10.1038/srep00521

Bertin-Mahieux, T., Ellis, D., Whitman, B., & Lamere, P. (2011). The Million Song Dataset. *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*