EE-558 | A Network Tour of Data Science

Project: Free Music Archive:

Mood Changing Playlist Generator

Team 52:

Ahmed, Jawwad Imtiaz

Gómez Quintanilla, Miguel

Hosseini, Reza

Jacovides, Emmanouil

Lecturer

Professor Pierre Vandergheynst

Professor Pascal Frossard

Teaching Assistant (FMA dataset)

Michael Defferrard

18th January 2019

1. Motivation & Objective

Scientific literature supports the idea that listening to certain types of songs can improve/change one's mood [1]. The objective of this project is to generate a playlist that can bring the listener from his actual mood (e.g. angry, sad, and so on) to a mood he would like to be in (e.g. relaxed or happy). If the transition from one mood to another happens in a smooth way, then it can unconsciously change the user's mood. This transition should also be smooth in terms of musical transitions while respecting the user's music preferences.

2. Introduction

To achieve the objective, we must find attributes for each song in the database corresponding to each mood. According to Fontaine et al., certain dimensions contain the emotional response of a human to a song [2] and indicators are chosen from our database to quantify these dimensions, as shown in the table below:

Fontaine et al. Dimension	**Corresponding Echonest Audio Feature Dimension**	**Dimension Description**
Pleasantness Valence	Valence	Pleasant emotions as opposed to unpleasant emotions.
Potency Control	Danceability	Feelings of power and proactiveness, making the listener want to express himself, take action.
Activation Arousal	Energy	Emotions such as stress, anger, and anxiety as opposed to disappointment, contentment, and compassion.

*the fourth dimension of Fontaine et al. (Novelty/Unpredictability) was ignored as it was not relevant to the project

In this project, the user will have two options to specify seed and end songs, either by inputting 3 moods indicators or directly choosing the songs.

The similarity network will be built, in order to make sure that the songs selected on the playlist are similar and the musical transitions are smooth. Once the graph is built, several paths of length equal to K will be generated. This length K could theoretically be any number between the shortest path and the diameter of the graph, but we considered k equals the shortest path length plus two, for gaining the maximum efficiency. These generated simple paths will be ranked according to two criteria; the smoothness of the path and the specific user's preference. The best-ranked playlist will be the one returned.

In order to collect data and the develop the application with the desired characteristics, the team required a powerful configured machine. We used a Google Cloud virtual machine with the following configurations: 8 cores (2.6 GHz each) of CPU, 52 GB of memory, and 100GB of SSD drive, running on Debian (Linux).

3. Data Collection and Preprocessing

First, we tried to use the initial dataset provided in the course came from Free Music Archive (FMA). However, since this dataset did not contain user preferences data, we decided to explore other alternatives. For this reason, we used the Million Songs users' history Dataset, provided by Echonest. This dataset only contained (userID-trackID-counts) triplets. Thus, for getting a similar dataset as to FMA, we had to build our own dataset. For doing so, the metadata, audio features and a 30-second preview audio file were collected for each song using Spotify API.

Then, we used the collected 30-second preview audio file and the Librosa library to compute the features, similar to Librosa features in FMA dataset. As the preprocessing step, we split the songs into a train-test split and divide them into 'small-medium' sub-sets.

In total, the project used four datasets, namely:

1. User's History (Million Songs Dataset)
2. Audio Features (Spotify API)
3. MetaData (Spotify API)
4. Librosa Features (Computed)

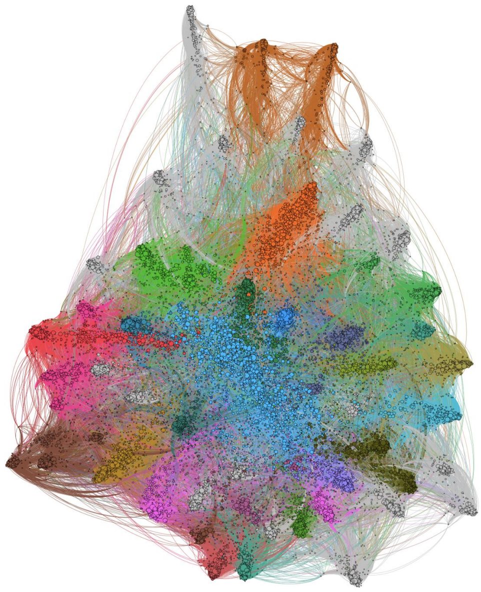
4. Similarity Graph Creation

In order to generate a smooth playlist, the similarity graph has to be based on songs and their connection between other songs based on their audio characteristics [3]. Initially, the similarity was created using Librosa Features (with each of the 518 features that each song contains). However, clustering could not be clearly identified in this graph (Appendix A). Another way of creating the similarity graph has been used, based on user's preferences. In this case, the songs that are more similar will also be closer given the fact that certain users like certain *Concepts* of music.

In order to construct the similarity graph using users' preferences, the Single Value Decomposition (SVD) method has been applied. SVD allows decomposing a matrix into three matrices (Left singular, right singular, and sigma matrices), extracting useful and interesting properties of the original matrix by mapping to lower dimensions [4]. In our case, one of the decomposed matrices contains information about the *Concept/Latent Features* (Appendix B). This matrix was used in the construction of the final similarity graph. The graph is sparsified keeping 30 strongest edges for each node/song, along with setting a threshold to promote the strongest edges.

In order to create the visualized graph with Gephi, the algorithm Modularity was run. This algorithm is used to detect communities in large networks [MIG1]. After running the algorithm, more than 37 communities were found.

In the picture of the similarity graph on the next page, we were able to visualize the underlying structure of the network by visualizing different communities of the songs by with different colors, in Gephi software.



Different Communities of Songs in the Similarity Graph based on User's preferences

5. Playlist Generation

When generating a playlist, the goal is to move from one mood to another mood with the help of changing the moods of songs. The change will be based on transition smoothness and user preference.

a. Seed Song & End Song

The user has two options to select or be assigned the seed song and the end song:

1. **Select Directly the Song:** The user can directly select a seed song or end song which he/she wants to listen to at the beginning of the playlist or at the end of the playlist.
2. **Input Scores for Mood Dimensions:** The user can select the values of the 3 mood dimensions namely Valence, Danceability and Energy for the seed song (given his/her current mood) or end song (given his/her desired mood), and be directly assigned a seed or end song with respect to the closest song corresponding to the values of the dimensions.

b. Smooth Path for Moods

The transition from one song to another song needs to be as smooth as possible. In order to implement the smoothness in the mood change, we compute the L2-Norm between the mood signal and the transpose of the incidence matrix S^T. The mood signal is a matrix containing the audio features from Echonest, which represents the mood of the songs, while the incidence matrix has been computed filtering the adjacency matrix of our graph.

c. Utility Matrix

The Utility Matrix (UM) is created with the help of the users' history database. In the UM, each row represents a user while each column represents a song. The value in each cell corresponds to the number of times the user (row) has listened to that particular song (column). To control the effect of some heavy listeners who had listened a lot to different songs, we mapped the songs' count larger than the 60 percentile of the user's count to number 3 (representing the songs that the user likes very much) and and counts with smaller values to number 1 (representing songs with moderate likability)

From the UM, the team has to extract the *Concepts*. *Concepts* are latent features that are the result of a Low Ranked Matrix Factorization (LRMF). The LRMF helps to describe the user preferences in terms of a small set of latent features. In order to achieve this reduced dimensionality, Collaborative Filtering (CF) has to be performed. In order to execute CF, a "Matrix Factorization" method called "Non-Negative Matrix Factorization" (NMF) is selected by the team. NMF is a linear algebraic model that factors high dimensional vectors into low dimensional vectors [5]. The main advantage for our case using NMF over other factorization techniques is that the approximation is based on adding factors, a property explained as learning parts of the objects, which corresponds to playlists in respect to our case [6].

d. Personalized PageRank

In order to find the most optimal path based on the particular users' preferences, the team needed to rank the importance of each song by an individual user. Page rank itself is the probability of landing on a specific page in the stationary state; the probability that one can move from one node i to j, if it was on node i in the previous step [7]. In the case of Personalized PageRank, you have a teleport probability vector. Thus, one is biasing where to restart, which is more relevant to a given topic or in our case the individual user [8]. Hence, the Personalized PageRank principle was used on the similarity graph with a personalized vector of the individual user from the Utility Matrix as the input.

e. Choosing the Path

A list of paths is obtained from our graph based on the seed and end song, with two ranking mechanisms:

1. **Smoothness Rank:** This ranks all the paths depending on how smooth the paths are given their mood dimensions (section 5.b).
2. **User Preference Rank:** This ranks all the paths based on the Personalized PageRank results obtained for the particular user (section 5.d).

The Final Path is computed based on a combination of both rankings. The team has included a variable weight component λ which is to be multiplied with the User Preference Rank matrix.

6. Demo - Results and Evaluation

The team wanted to evaluate the application created. The seed song was assigned to us based on our 3 mood dimension inputs. The team input low values for Valence, Energy, and Danceability for the seed song, as the aim was to reflect a sad mood. However, high values were input for the three dimensions when selecting a target/end mood, in order to reflect a happy mood. The idea was to imitate a user wishing to go to a happy mood from a sad mood. Input data for seed and end song are shown below:

```
seed_song=song_selection(0.1,0.1,0.1)
seed_song
```

```
end_song=song_selection(0.9,0.9,0.9)
end_song
```

Based on seed and end song, the mood changing playlist was generated as shown below:

	positionID	trackTitle	artistName	releaseDate	topGenre1	audio_Duration_ms
62093	0	She Don't Sleep	Craig Wedren	2005	NaN	225947
57853	1	One More Chance	Nico	2007	Pop	426867
52849	2	Picks And Pans - Live	John Scofield	1990	Jazz	420000
16283	3	Let's Go To Vegas	Paul Gilbert	2008	NaN	142592
1096	4	Hickeys Around My Neck	Audio Two	1988	Hip-hop	232640
31973	5	Wreck Shop	Run–D.M.C.	1993	Hip-hop	194293
7204	6	War	Bounty Killer	2006	Reggae	209227
55808	7	On The Road Again - Pigna People Remix	Telex	2006	NaN	352787

The team listened to the playlist in the assigned sequence, using our personal Spotify accounts to find the songs (Appendix C). It was clear that the first song was a very sad song, as it was very low beat and relaxing song. The next songs that followed had very smooth transitions but where of a higher beat/tempo compared to the previous song. This was exactly what was desired from the playlist generator. The last song on the playlist is a very upbeat and "fun" song, depicting a happy song. The team repeated this process many times, changing both the 3-dimensional values and users and every time a different playlist was obtained but generally followed the trend we wanted to obtain. The team was even successful in going from a happy mood to a sad mood by changing the 3-dimensional values. It was also evident that the user preference was taken into account as it was observed that some playlist were of a single genre such as country or hip-hop, indicating that the particular user likes this dominating genre.

7. Conclusion

The team was successful in the execution of developing a Mood Changing Playlist Generator. There were many complexities that the team encountered during the development, which the team solved by making logical assumptions and limiting the scope where necessary. In terms of future recommendations, the team advises refining the code further to become less computationally extensive and provide a friendlier input interface for the user. The project at hand is beneficial in helping users to improve/change their mood through music. The application created has the potential to become a minimum viable product in the near future.

8. References

[1] Sandra Garrido, Emery Schubert, Daniel Bangert, Musical prescriptions for mood improvement: An experimental study, The Arts in Psychotherapy, Volume 51, 2016, Pages 46-53, ISSN 0197-4556, https://doi.org/10.1016/j.aip.2016.09.002.

[2] Fontaine, J. R. J., Scherer, K. R., Roesch, E. B., & Ellsworth, P. C. (2007). The World of Emotions is not Two-Dimensional. Psychological Science, 18(12), 1050–1057. https://doi.org/10.1111/j.1467-9280.2007.02024.x

[3] Adam Berenzweig, Beth Logan, Daniel P. W. Ellis, and Brian P. W. Whitman. A large scale evaluation of acoustic and subjective music-similarity measures. Comput. Music J., 28(2):63–76, 2004.

[4] Iraj Rahimi Pordanjani, Wilsun Xu, A singular value decomposition-based technique for decoupling and analyzing power networks, International Journal of Electrical Power & Energy Systems, Volume 74, 2016, Pages 265-273, ISSN 0142-0615, https://doi.org/10.1016/j.ijepes.2015.07.028.

[5] Benzi, K., Kalofolias, V., Bresson, X. and Vandergheynst, P. (2019). Song Recommendation with Non-Negative Matrix Factorization and Graph Total Variation. [online] arXiv.org. Available at: https://arxiv.org/abs/1601.01892

[6] Daniel D Lee and H Sebastian Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, vol. 401, no. 6755, pp. 788–791, 1999.

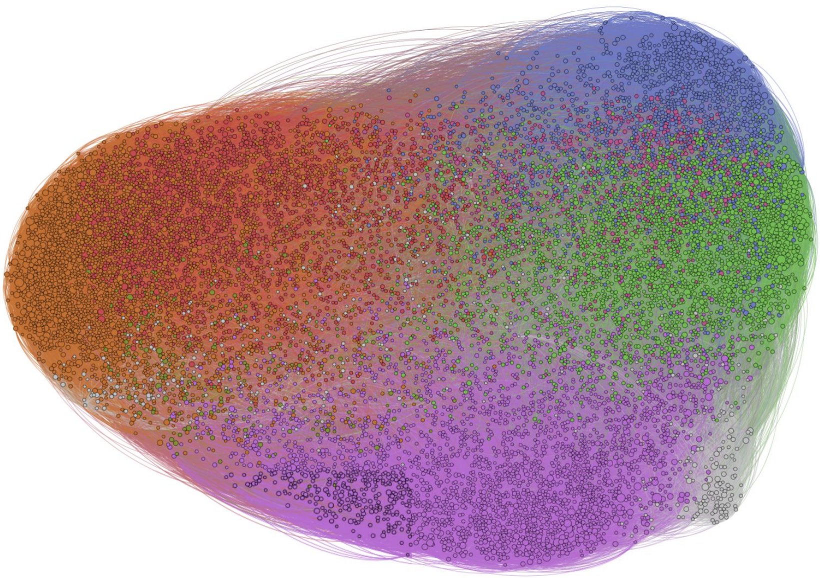
[7] Kumar, K., Abhaya, & Mukoko, F.D. (2013). PageRank algorithm and its variations : A Survey report.

[8] T. H. Haveliwala. Topic-Sensitive PageRank, 11th International World Wide Web Conference, 2002.

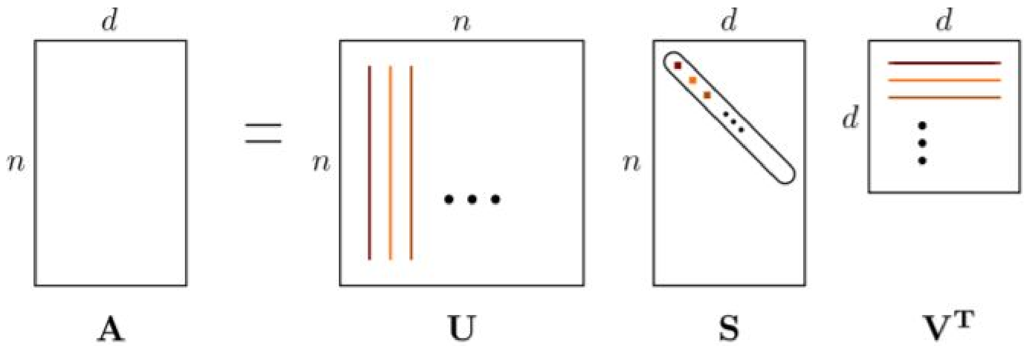
[9] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, Fast unfolding of communities in large networks, in Journal of Statistical Mechanics: Theory and Experiment 2008 (10), P1000

Appendix

A - Similarity Graph Using Librosa Features



B - Singular Value Decomposition



Singular Value Decomposition: Each singular value in S has an associated left singular vector in U, and right singular vector in V.

C - Playlist Generated on Spotify Playlist from the Playlist Generator

Filtrar Descargar

TÍTULO	ARTISTA	ÁLBUM		
She Don't Sleep	Craig Wedren	Lapland	hace 3 minutos	3:46
One More Chance	Nico	Best Of Nico: LIVE	hace 3 minutos	7:07
Picks And Pans - Live	John Scofield	Pick Hits Live	hace 2 minutos	7:00
Let's Go To Vegas	Paul Gilbert	Confessions of a Las Vegas Loser	hace 2 minutos	2:23
Hickeys Around My Neck	Audio Two	What More Can I Say	hace 2 minutos	3:53
Wreck Shop	Run–D.M.C.	Down With The King	hace un minuto	3:14
War	Bounty Killer	Nah No Mercy - The Warlord Scrolls	hace unos se...	3:29
On The Road Again - Pigna People Remix	Telex	On The Road Again	hace unos se...	5:53