Finding Music Friends: Cross-Referencing Listening Trends

SENG 474 / CSC 578D - Project Proposal

Group Members:

- Denholm Scrimshaw
- Sarah Warnock
- Trevor Baker
- Kirin DeVries
- David Luco

Abstract:

Music plays a role in everyone's day-to-day lives. With the constant progression of the internet and an ever increasing population, the variety and availability of music is at an all-time high. As a means to save time and assist in the seemingly aimless navigation of sources, playlist makers attempt to connect tracks based on varying degrees of relevance. Much like the 'Netflix Challenge', the construction of musical playlists is an abstruse problem. In a sense, it is an attempt to extract objective and quantitative properties out of subjective and abstract art form. Many online music providers have been attempting to solve or build upon this problem.

Related Work

Playlists came to fruition as music evolved from live performances to recorded data. Since then, various techniques have been established in response to the revolutions in music storage and sharing. On one side there exists the human connection; the links between producers and consumers that produce a relational web; a beckoning call to the graph theorists. On the other you have contextual analysis of music, which includes textual metadata in the form of: artist names, genre labels, titles of work, album names, indications of moods, etcetera. It also includes audio signal content which indicates timbre, pitch, tempo, and other elements which take considerably more technical analysis.

An advanced method and interface to browse music collections is suggested in [1] which takes into account information about each song as well as social features. When creating a playlist or matching songs by similarity, audio metadata can be used first to restrict the number of possibilities to within a certain genre, and then social tags can be used to match a certain mood. Probabilistic Latent Semantic Analysis (PLSA) is used to develop relationships within a music collection which can then be used to suggest similar songs given a certain track.

Though the quantitative analysis of audio signals would provide an ideal dataset for numerical computation, the algorithms to do so are not only extremely technical, but have been studied for years and are fairly well established. Moreover, services already exist which try to categorize this data into a more relatable context, such as loudness or energy or how acoustic a song sounds. The Echo Nest is a recent acquisition of Spotify which does just that, and goes one step further by using this data to organize music and construct playlists.

The other half of the equation is the human relationships and listening patterns. This is perhaps even more commonplace with companies like SoundCloud who provide track recommendations based on a network of interaction. With the swift emergence of online music suppliers, such as: Apple Music, Slacker, Rdio, Deezer, Pandora, and Google Play Music, competition is racing to come up with a more accurate

(and sophisticated) playlist mechanism. As such, our group's initial intent to combine the human and computer analysis of songs into some new playlist concept seemed anything but new. In fact, Spotify's integration of The Echo Nest into their already developed web of listening patterns was essentially our plan. It's nice to know your ideas are on par with Daniel Ek (the CEO of Spotify, who started his first company at the age of 14) [5], but our realization unfortunately required a shift in direction for this project.

Proposed Project:

How do you come up with something new in a field that has been heavily studied and is currently at the base of several billion dollar companies? You leverage their efforts of course!
Instead of coming up with simpler and less effective algorithms, or worse, a completely new topic, we decided to use these sites to our advantage. The data analysis aspect of this playlisting issue seems to be pretty much handled, so we switched our focus to data collection.

Our idea is based on cross-referencing popular online music distributors for the subsequent purpose of finding similarities among human connections. For instance, suppose a profile on Soundcloud shares 50% of the same 'liked' tracks as one on Spotify. You may draw the conclusion that these two profiles share similar music tastes, and thus the 50% of likes not in common may be of interest to the other party. By utilizing several current services, the resulting data takes advantage of the analysis and playlist algorithms each service has already put in place. Since we are gathering data from several sources, our project will likely focus on the data collection and aggregation into a common, coherent format.

Once we have a reasonable source of data, we can witness the resulting similarity between user profiles, and may be able to construct decision trees using algorithms like ID3 to optimize our gathering 'path' for relevancy. In other words, we may be able to prove that going through a user's followers is a more reliable source than the list of people the user follows. The decisions of the decision tree could come from Bayesian analysis on which attributes of a track are most relevant (eg. artist name over track plays). Although similarity is a simple form of evaluation, it is possible to go further with cross-validation by removing 'liked' tracks from the dataset and testing our list of similar tracks against those removed. Lastly there is human evaluation; using a known user's profile as a training set before having them rate the resulting tracks.

Depending on the time required to collect data from the various sites, the amount and complexity of the analysis performed on the resulting data set may need to be adjusted to fit the project's schedule. If it is not possible to collect useful data from one or more of the sites, it may be necessary to reduce the project's scope to focus on analysis within one site's users.

Project Timeline

Week	Goal	Participants
Jan 25 - 31	**Research** available APIs/datasets, and determine the best candidates based on the **available data** and repercussions on the **complexity/**technology stack required. Take note of potential **connections** for analysis and the associated relevance of the data.	Everyone
Feb 1 - 7	Start data collection by utilizing the highest priority services	Members will be split individually or in partners and directed towards the APIs chosen during the previous week
Feb 8 - 14 Reading Break	Continued collection, and beginning of planning for a standardized way of comparing cross-API data	Collection: Denholm, David, Sarah Analysis:Kirin, Denholm
Feb 15 - 21	Combining data and starting to measure similarity / analysis Write Midterm report	Analysis: Kirin, Evaluation: Denholm, Sarah Written: David, Sarah
Feb 22 - 28	**Midterm Report due** Analysis & Evaluation	Analysis:Kirin, Denholm Evaluation:
Feb 29 - Mar 6	Analysis & Evaluation	Analysis:Kirin, Evaluation: Denholm
Mar 7 - 13	Create Presentation Begin Writing Final report	Presentation: All Written: Denholm, David
Mar 14 - 20	**Presentations begin**	Presentation: All Written: Denholm, David
Mar 21 - 27	Finalize final report	Written: Denholm, David, Sarah
Mar 28 - Apr 3	**Final Report due**	

Data Sources & APIs

SoundCloud

- Retrieve social tags for tracks (eg. “calming”, “energetic”) and user listening patterns, such as favourite artists.
- Python interface: https://github.com/soundcloud/soundcloud-python

Spotify

- Social connections (favourite songs and artists, friends with similar musical tastes)
- Python interfaces: https://pyspotify.mopidy.com/en/latest/ , https://github.com/plamere/spotipy

EchoNest

- Retrieve extended audio features and metadata for songs and artists, including: tempo, key, duration, energy, and many more
- Python interface: http://echonest.github.io/pyechonest/

Last.fm & Libre.fm

- Retrieve social tags for a given track. Of particular interest is the list of top tracks for a user. If a large list of users could be generated, this would be useful in the creation of song recommendations.
- Get similar tracks for a given track. This could be useful to train our algorithm.
- Python interface: https://pypi.python.org/pypi/pylast/
- Used by [1] with their PLSA method

For each of the data sources and APIs listed above, the usefulness and applicability will be assessed with respect to the project’s goals. As the project progresses, data sources and APIs may be added or removed from the above list. It is possible that a given API does not provide access to enough users to perform meaningful analysis. The last.fm API in particular is of concern, since the web site was recently redesigned and some features on the site no longer function, like searching for users by name.

It is important while using these APIs to avoid relying on their built-in suggestion and recommendation features. However, those features will likely be useful in training and developing the data mining algorithms, with potential in establishing evaluation mechanisms.

References

[1] M. Kuhn, R. Wattenhofer and S. Welten, "Social audio features for advanced music retrieval interfaces", *Proceedings of the international conference on Multimedia - MM '10*, 2010.

[2] B. Fields, "Contextualize Your Listening: The Playlist as Recommendation Engine", *Goldsmith University of London*, 2011. [Online]. Available: http://benfields.net/bfields_thesis.pdf. [Accessed: 28-Jan-2016].

[3] SoundCloud Community, "My "recommended" tracks are too much different from the music I do. | SoundCloud Community Forum", 2016. [Online]. Available: https://www.soundcloudcommunity.com/soundcloud/topics/my-recommended-tracks-are-too-much-different-from-the-music-i-do. [Accessed: 28-Jan-2016].

[4] D. Pierce, "Inside Spotify's Hunt for the Perfect Playlist", *WIRED*, 2016. [Online]. Available: http://www.wired.com/2015/07/spotify-perfect-playlist/. [Accessed: 28-Jan-2016].

[5] Wikipedia, "Daniel Ek", 2016. [Online]. Available: https://en.wikipedia.org/wiki/Daniel_Ek. [Accessed: 28-Jan-2016].