ANLY 501 Project: Part 1

October 4, 2016

Joo Chung and Jingshi Sun

**The Data Science Problem**

Digital music is a rapidly growing market, with the use of online music streaming services alone saw a near doubling between 2014 and 2015 (Nielsen, 2015). A common source of revenue for these services is the use of targeted advertisements, whether through within-streaming audio advertisements[[1]](#footnote-1) or through music discovery algorithms[[2]](#footnote-2). As such, the ability to offer more relevant and personally engaging advertisements to users is a potential means of increasing revenue (Farahat & Bailey, 2012). Music recommendation algorithms typically take into account user listening behaviors and tastes in addition to standard demographic variables (e.g., streaming devices, approximate geographic location). The current report will investigate the viability of using the user’s emotional state in order to further inform a recommendation algorithm.

The connection between emotional states and music preferences are well documented. Studies have shown that people listen to music to regulate and manage their moods (Lonsdale & North, 2011). Music therapy has been known for some time as a viable treatment for mood disorders (Choi, Lee, & Lim, 2008; Maratos, Crawford, & Procter, 2011). Of course, music and music performance has traditionally been considered an outward communication of the emotions of the performers (Scherer, 1995).

As a device that accurately transmits the emotional states of music listeners at scale does not currently exist, an alternative (and less controversial) methodology is the utilization of nationally aggregated listener behavior in the context of popularized social media. The current report will investigate whether or not the emotional content in popular Twitter posts have an appreciable effect on song popularity during the week following the event. Further, we hypothesize that the song preference of music listeners will respond differently depending on the specific emotions depicted by the popular posts in a fashion that is consistent with social psychological literature (e.g., Lazarus, 1991; Roseman, Antoniou, & Jose, 1996). For example, we believe that if a popular post depicts happiness, listeners will prefer songs that are also happy in theme and tone (i.e., emotion-consistent listening preference). In contrast, we believe that posts that inspire emotions such as fear or disgust will be motivated to listen to songs that are unrelated (or even counter) in theme and tone to the stimulus as a means of distracting oneself from stressors (i.e., emotion-inconsistent listening preference). Among the multitude of possible emotions to explore, the current study will focus on the six so-called “basic” emotions (i.e., anger, disgust, happiness, sadness, surprise, and fear) that have been studied extensively in psychological literature (Ekman, 1992).

**Potential Analyses**

In order establish a link between popular Twitter posts to song popularity, several important datasets will need to be collected. There are several variables that are likely available directly from the music streaming service. Several notable examples are 1) song genre, 2) artist popularity, and 3) the word composition of the song lyrics. Word lyrics, in combination with the title and album names, provide the crucial bridge that connects the emotional themes of a given song to the emotional content of Twitter posts. The Twitter posts themselves will be collected using the Twitter API with a particular focus on the top 10-15 retweeted posts of a given week. The song lyrics and Twitter posts will then be assigned a primary (and possibly secondary) emotional theme through the use of the NRC Word-Emotion Association Lexicon database[[3]](#footnote-3). The degree of emotional cohesion between the song lyrics and Twitter posts will then be used as a predictor in a regression-type analysis controlling for other song-descriptive variables. The data collection could then be repeated weekly over a period of time or in order to detect longitudinal differences in music preferences as new Twitter posts become popular.

**Collecting New Data**

During this first stage of the project, we aimed to collect data pertaining to the songs and their popularity ranking and song lyrics. For song ranking, several data sources were examined (e.g., Pandora, Spotify) and we determined that Last.fm was suitably well-known, open in their data dissemination policies to allow it to be our primary source of song data, and was easy to use for the data of interest (i.e., aggregates for popularity in the United States). In particular, the Last.fm API allowed us to directly retrieve the ranking of the most popular tracks in the United States from the previous week using a single method (i.e., geo.getTopTracks). Further, Last.fm’s lack of a hard limit on the number of API calls allowed us to experiment with the API and collect the data fairly quickly. A list of available song information is shown in Table 1.

|  |  |
| --- | --- |
| Table 1: Attributes from Last.fm | |
| Attribute Name | Description |
| page | Page number of top tracks |
| name | Name of the track |
| songmbid | Track MusicBrainz ID (MBID) |
| artistname | Artist Name |
| artistmbid | Artist MusicBrainz ID (MBID) |
| artisturl | Artist URL on Last.fm |
| duration | Duration of track (in seconds) |
| listeners | Number of listeners for week of interest |
| url | Track URL on Last.fm |
| rank | Track rank for week of interest |
| streamstat | Unknown (No documentation) |
| streamtext | Unknown (No documentation) |
| imagesmallurl | Album image URL (Small) |
| imagemedurl | Album image URL (Medium) |
| imagelargeurl | Album image URL (Large) |
| imagexlargeurl | Album image URL (X-Large) |

Last.fm utilizes a user’s listening behaviors in order to recommend similar artists and songs that the user has not tried yet. Strictly speaking, Last.fm does not offer any services to actually play music through their service, but acts as an optional integrated service with existing internet radio stations (e.g., Pandora, Slacker) and on-demand online music streaming services (e.g., Spotify, Apple Music). Thus, Last.fm provides a substantive ranking of popularity based on a multitude of media sources.

The following is an example Last.fm API call to in Python using the urllib module, which requests the top tracks in the United States from the previous week (with a limit of 2 tracks for each of 2 pages, for a total of 4 tracks):

url = {'method': 'geo.getTopTracks',

'country': 'United States',

'page': 2,

'limit': 2,

'api\_key': api\_key,

'format': 'json'}

response = requests.get("http://ws.audioscrobbler.com/2.0/", url)

txt = response.json()

print(txt)

The output of the above script is shown on Figure 1. In total, a dataset containing the top 5656 tracks was collected for the seven days prior to September 27, 2016.

|  |
| --- |
| Figure 1: Example Last.fm API JSON Response (Top Half) |
|  |

The collection of lyric data was far more involved. Official channels for gathering lyrics for each song involved the use of the Lyricfind API or the Musixmatch API, both of which were paid services. Musixmatch did offer a non-commercial license, but it limited the number of API calls to 2000 per day and only provided incomplete lyrics (specifically, 70% of the lyrics were omitted). Due to these limitations, we decided to utilize the LyricWikia, which crowdsources its lyrics text from its users. An example search result for “Starboy” by The Weeknd is shown on Figure 2.

|  |
| --- |
| Figure 2: LyricWikia Page Layout |
|  |

Searching for a specific track on LyricWikia can be completed by specifying the artist name and song name (separated by a colon and replacing white spaces with underscores) into the URL. (For the example in Figure 2, the URL was: http://lyrics.wikia.com/wiki/The\_Weeknd:Starboy.) Given this simple scheme, we were able to use the song and artist names collected from Last.fm and incorporate them into a URL request through the Python module urllib. The following is an example URL request in Python using the urllib module which corresponds to the page in Figure 2:

url = "http://lyrics.wikia.com/wiki/" + 'The\_Weeknd' + ":" + 'Starboy'

req = urllib.request.Request(url) # create Request object

data = urllib.request.urlopen(req) # open site and pull html

getdata = str(data.read())

print(getdata)

The output of the above script is shown on Figure 3. However, before these requests could be used, the search URLs needed to cleaned in several ways. First, beyond the replacement of white spaces to underscores, the URL also needed to be free of any parentheses present in either the song or artist names. For example, the full credits for the song shown on Figure 2 is actually “Starboy (feat. Daft Punk)” but the final search result did not incorporate the parentheses or its contents. In practice, the removal of parentheses generally improved the rate of correct searches, so this correction was applied to all song titles and artist names while utilizing the search. Secondly, special characters such as accented letters (e.g., “ã”) or language-specific letters (e.g., “ø”) needed to be replaced by their English alphabet analogues (e.g., “ã” becomes “a”, “ø” becomes “o”). This was done in order to prevent a common Python error triggered from a string being unable to be implicitly converted to ASCII (a conversion that occurs automatically within the urllib.request.Request() method). Lastly, the vertical bar character “|” needed to be replaced by “L,” which was the convention used in LyricWikia (e.g., the song “untitled 04 | 08.14.2014.” by Kendrick Lamar returned an empty search unless the URL header was converted to “Kendrick\_Lamar:Untitled\_04\_L\_08.14.2014.”). After completing the cleaning process, the URL was then used to full the resulting page HTML.

In the event of a failed search, LyricWikia typically offered either a “Recommended…” link or a “Did you mean…” link which provided the correct URL to the lyrics of interest. The Python script automatically detected the failed search and utilized regular expressions to transfer to the recommended site. In the event that the Python script failed to find any recommended links, the string value “Not found” was used instead of the lyrics.

If the correct URL was found, the section containing the lyrics was isolated using regular expressions. As shown on Figure 3, the section containing the lyrics had integers in place of each letter (i.e., decimal text), so these were converted back into letters using the Python chr() function. The finalized dataset contains two attributes, as shown on Table 2.

|  |
| --- |
| Figure 3: LyricWikia Page Source |
|  |

|  |  |
| --- | --- |
| Table 2: Attributes from LyricWikia | |
| Attribute Name | Description |
| url | Track URL on Last.fm (for matching) |
| lyrics | Lyrics string |

**Data Issues**

At this stage we noted that there were several notable issues with the collected Last.fm dataset. First, the API occasional exhibited strange repetitious behavior by returning several hundred duplicates with each method call. For our current dataset, the number of duplicates was rather small (286) relative to the total number of rows so this behavior was not deemed to be especially problematic. It should be noted that since the LyricWikia dataset utilized the song name and artist variables from the Last.fm dataset, the same 286 duplicate values were present in this dataset as well. Secondly, as shown on Table 1, there were several variables (streamstat and streamtext) that were in the return JSON files that had ambiguous in their purpose due to the lack of documentation. These attributes will be further investigated to determine their usage. Thirdly, the rank variable was found to restart at zero at every page (e.g., ranking of 0 to 50 for the first page, and ranking of 0 to 50 for second page, etc.). While this is not especially problematic since the order of popularity was retained in the ordering the rows, the creation of a new “global” ranking variable was needed.

For the LyricWikia dataset, there were also several notable issues stemming from the crowdsourced nature of the data source. First, the lyrics found on the website contained inconsistent formatting. For example, certain songs opted to use the text “(x2)” in order to denote that a certain section was repeated twice, while other songs opted to actually repeat the section twice. This will become an issue when determining word prevalence. Also, inconsistent spacing sometimes caused the first letter of the lyric string to be cut off, though this is generally rare. Secondly, there were several songs on the dataset that actually do not have lyrics (i.e., the songs are instrumentals), which have the string “Instrumental” in place of their lyrics. Since these songs are not technically missing values, we will have to determine if their presence is problematic during analyses. Thirdly, despite the song list being limited to the United States, there were a handful of foreign language lyrics. As with the instrumental songs, these are technically not missing values so we will need to further investigate the possibility of keeping these cases in the dataset through the means of translation. Lastly, a theoretical source of additional noise is the possibility of spelling mistakes or the use of similar sounding but different meaning words due to the fact that these lyrics were crowdsourced. This possibility will be further investigated at a future date during the analysis portion of this study.

**Data Cleaning**

This section will detail some preliminary data quality statistics that were generated. There were several variables with a prevalence of missing (or equivalent to missing) data after accounting for the duplicate rows. As shown on Table 3, variables streamstat and streamtext in the Last.fm dataset contained zeroes for all rows, which suggests that it was an attribute that was created but never populated by Last.fm. Variables songmbid and duration have approximately one-third of their respective fields missing (or equivalent to missing). In particular, duration had a value of zero (which is obviously impossible for a song), which likely means that this field is still in the process of being populated by Last.fm. The remaining attributes from the Last.fm dataset contained 4% or less missing values (or equivalent) and were not deemed to be especially problematic. From the LyricWikia dataset, lyrics were not found for 8.5% of the songs after accounting for the duplicate rows. As a measure of overall data quality, an average percentage of missing (or equivalent) values was calculated for both the Last.fm and LyricWikia datasets. The overall data quality of the Last.fm dataset was 17.13% while the overall data quality of the LyricWikia dataset was 4.31%. By this measure, the Last.fm dataset appears to be more problematic overall when compared to the LyricWikia dataset.

**Feature Generation**

For the Last.fm dataset, three additional variables were created. First, a binary variable (called duplicate) was created which simply designated which rows were duplicated based on the variable url. This variable will facilitate data cleaning for the next step of the project. Secondly, a “global” rank variable (called fullrank) was created in response to the discovery that the rank variable provided by the Last.fm API restarted on every page. The creation of this variable took into account the duplicate rows designated by the variable duplicate. Lastly, a preliminary artist popularity rating (called artistfreq) was created by summing the number of occasions where a given artist appeared in the list of top 5000 songs. This variable will designate songs that are popular despite the artist not being widely popular. (For example, “Panda” by Desiigner was ranked 58th despite the artist appearing only once in the top 5000.)

For the LyricWikia dataset, one additional variable was created. The character length of the lyrics variable was calculated after removing white spaces and punctuation characters. This variable identified songs with an unusually large number of characters (which is characteristic of rap music) and several songs with unusually low number of characters. This variable was only populated for songs where the lyrics were found.

|  |  |
| --- | --- |
| Table 3: Data Quality Statistics | |
| Overall Statistics | |
| Total number of rows in Last.fm dataset | 5656 |
| Total number of rows in LyricWikia dataset | 5656 |
| Duplicate rows in Last.fm dataset | 286 |
| Duplicate rows in LyricWikia dataset | 286 |
|  | |
| Last.fm Output | |
| % Missing for page | 0.00% |
| % Missing for name | 0.00% |
| % Missing for songmbid | 33.70% |
| % Missing for artistname | 0.00% |
| % Missing for artistmbid | 4.08% |
| % Missing for artisturl | 0.00% |
| % “0” for duration | 36.19% |
| % Missing for listeners | 0.00% |
| % Missing for url | 0.00% |
| % Missing for rank | 0.00% |
| % “0” for streamstat | 100.00% |
| % “0” for streamtext | 100.00% |
| % Missing for imagesmallurl | 0.02% |
| % Missing for imagemedurl | 0.02% |
| % Missing for imagelargeurl | 0.02% |
| % Missing for imagexlargeurl | 0.02% |
|  | |
| LyricWikia Output | |
| % Missing for url | 0.00% |
| % “Not found” for lyrics | 8.61% |
|  | |
| Overall Data Quality | |
| Last.fm dataset | 17.13% |
| LyricWikia dataset | 4.31% |
| Note: Percentages currently include duplicates. | |

**CITATIONS**

Choi, A.N., Lee, M.S., & Lim, H.J. (2008). Effects of group music intervention on depression, anxiety, and relationships in psychiatric patients: A pilot study. *The Journal of Alternative and Complementary Medicine*. 14(5), 567-570.

Ekman, P. (1992). An argument for basic emotions. *Cognition and Emotion*, 6(3-4).

Farahat, A. & Bailey, M.C. (2012). How effective is targeted advertising? *SSRN*. Retrieved from http://ieeexplore.ieee.org/document/6580780/.

Lazarus, R.S. (1991). *Emotion and Adaptation*. Oxford University Press.

Lonsdale, A.J. & North, A.C. (2011). Why do we listen to music? A uses and gratifications analysis. *British Journal of Psychology*, 102, 108–134.

Maratos, A., Crawford, M.J., & Procter, S. (2011). Music therapy for depression: it seems to work, but how? *The British Journal of Psychiatry*. 199(2), 92-93.

Nielsen (2015). *2015 Nielsen Music U.S. Report*. Retrieved from http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2016-reports/2015-year-end-music-report.pdf.pdf.

Roseman, I.J, Antoniou, A., & Jose, P.E. (1996). Appraisal determinants of emotions: Constructing a more accurate and comprehensive theory. *Cognition and Emotion*, 10(3), 241-277.

Scherer, K.R. (1995). Expression of emotion in voice and music. *Journal of Voice*, 9(3), 235-248.

1. http://www.adweek.com/news/technology/4-ways-spotify-and-pandora-target-millennials-differently-167222 [↑](#footnote-ref-1)
2. http://smallbizlink.monster.com/news/articles/466-google-myspace-facebook-to-launch-new-music-discovery-and-revenue-models [↑](#footnote-ref-2)
3. http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm [↑](#footnote-ref-3)