Content Based Music Recommender System

by

Duggirala Krishna teja

A Project Report Submitted
in
Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in
Computer Science

Supervised by

Dr. Carol J. Romanowski

Department of Computer Science

B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, New York

December 2016
Acknowledgments

I would like to take this opportunity to thank to Dr. Carol J. Romanowski for giving me the opportunity to work under her mentorship and for her guidance for this Capstone Project. A special thanks to Dr. Leon Reznik for his guidance and advice on project deliverables in form of presentations, report and poster presentation during the Colloquium class. Last but not the least, a heartfelt mention for Dr. Hans-Peter Bischof, Rebecca O’Connor, Cindy Wolfer, my fellow classmates and all of the RIT and ISS staff for their help and role played all throughout my masters degree.
Abstract

Content Based Music Recommender System

Duggirala krishna teja

Supervising Professor: Dr. Carol J. Romanowski

With the digitization of music, it has paved the path for many changes. Some of the significant ones being extinction of CDs, online music streaming services, and too many options. With many options, the users are confused to select what they want to hear thus the need for a recommender system had arisen. One of the main factors which makes a user like a song is the subject of the song. Finding the subject/latent topic of a song is a very hard task because musicians play with contrasting or implied emotions thus causing the lyrics to be confusing, but musical interpretations of several users can be very helpful in finding the latent topic. Since manual annotation to a large set of music collections is arduous, Topic Modelling is preferred to automatically identify the underlying topics present in the text sources. In this project, Topic Modelling was performed on the interpretations of songs to produce a better recommendation system.
Contents

Acknowledgments ................................................................. ii

Abstract .................................................................................. iii

1 Introduction ........................................................................... 1

2 Related work ......................................................................... 3

3 Data ....................................................................................... 4
   3.1 Data - Collection ........................................................... 4
   3.2 Data - Cleaning .............................................................. 5

4 Design & Implementation ...................................................... 11
   4.1 Sentiment Analysis: ....................................................... 11
   4.2 Topic Modelling: ........................................................... 12

5 Results .................................................................................... 16

6 Conclusion ............................................................................... 18
   6.1 Current Status ............................................................... 18
   6.2 Future Work ................................................................. 18
   6.3 Lessons Learned ........................................................... 19

Bibliography ............................................................................... 20
List of Tables

5.1 Comparison Table .......................................................... 16
List of Figures

3.1 Word Cloud of lyrics data before cleaning . . . . . . . . . . . . . . . . . . . 5
3.2 Word cloud of Interpretations data before cleaning . . . . . . . . . . . . . . 6
3.3 Interpretations data after cleaning . . . . . . . . . . . . . . . . . . . . . . . . 8
3.4 Lyrics data after cleaning . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
3.5 Word Cloud of lyrics data after cleaning . . . . . . . . . . . . . . . . . . . 9
3.6 Word cloud of Interpretations data after cleaning . . . . . . . . . . . . . . 10
4.1 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
4.2 Sentiment Analysis Result . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
4.3 Topic Model Result . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
4.4 Classification Result . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
4.5 Final Output . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
5.1 Comparison . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
Chapter 1

Introduction

- There are many music recommender systems available, some are based on metadata, some are based on algorithms and some are based on content such as lyrics.

- One of the leading music streaming providers Spotify uses collaborative filtering, one of the main disadvantages of using collaborative filtering is that information about the data is not considered.

- Some disadvantages of using the previously mentioned music recommender systems:
  
  Metadata Recommender systems results are too general. User efforts required to create a profile.

  Collaborative filtering Recommender systems cannot draw inferences about a single user. The results are very subjective.

  Content based Recommender systems use lyrics which can be very confusing.

Many researchers have attempted to extract subject information using several supervised and unsupervised techniques on the song lyrics. While they produced quite good results, the lyrics contain many words which are ambiguous in nature and made it difficult for the machine to understand the context of it.

- The analyzation of the interpretations of songs is an active research topic, hence decided to perform my algorithm on the interpretation of the songs.
Following are the main goals of this project:

- To create a data set of song lyrics and interpretations of each song by several users.
- To collect tags/moods of different albums
- To perform sentiment analysis on the lyrics of the songs.
- To perform topic modelling on the interpretations of the song.
- To classify the tag of the album by using the results of sentiment analysis and topic modelling.
Chapter 2

Related work

Furuya et. al \[4\] focused on finding the mood of the song by classifying the song lyrics. They have proposed an algorithm where they extract all the emotions related words from a song lyrics and then match it with the available feeling word list to classify the song.

Alen Lukic \[6\] a Carnegie Mellon University student worked on several types of topic modelling on song lyrics such as LDA (Latent Dirichlet Allocation) and PAM (Pachinko allocation).

Michael Fell et. al \[3\] took different dimensions of song and applied n-grams algorithm to find the genre of the song, distinguishing best or worst song, and approximate publication date.

Rachael E. Jack et. al \[5\] proposed a paper on finding the basic set of human emotions based on facial expressions.

Adam Poulston et. al \[7\] proposed a paper where they have implemented n-grams and topic modelling on 4 different languages corpora consisting of tweets.
Chapter 3

Data

There were many API’s and data sets for collecting the meta data of songs and lyrics of the song. Some of the famous API’s or data sets researched are

- Million Song Dataset,
- Echonest,
- Allmusic,
- Musicbrainz, and
- Discogs.

While these API’s were great, none of them provided me with interpretations for songs. Then tried to search a lot for an API/dataset which had interpretations of songs but could not find one. So finally decided to create one on my own.

3.1 Data - Collection

Implemented a program with the help of python’s beautiful soup package which parsed the websites and gathered lyrics, interpretations and tags of several albums. The websites used for these purposes are all music[1], and song meanings[2]. Due to the restriction of time, created a small dataset which consisted about 250 songs. Even though its small its good enough to run the selected algorithms and get good results.
Following are the steps taken to collect the data:

- First had to manually select the link of the album/song from song meaning website.
- Import the HTML parser via beautiful soup and start parsing the page.
- Next, using the div and attribute tags find the title of album and create folders with the name of title in the lyrics and interpretations folder.
- Open those files in write mode.
- Then again using the div and attribute tags to find the lyrics and interpretations and write them to the respective folders.

![Figure 3.1: Word Cloud of lyrics data before cleaning](image)

### 3.2 Data - Cleaning

Data cleaning also known as Data scrubbing is one of the most important tasks of any data related project. If the task is not done properly, there might be a drastic change in the
output. So took the maximum time to carefully check what the data is and what steps to be taken care of. First, I have created an R script to create a word cloud of the collected data. Checked each word cloud properly and came to a decision to perform the following techniques.

The lyrics data was much clearer and did not require much cleaning except for the few basic ones but the interpretation’s data was very messy and needed extensive cleaning.

As shown in the figures 3.1 and 3.2, we can easily see many faults with the data such as

- Many stop words are given high importance because of the frequency of the occurrence.

- In figure 3.2 a particular word ”@qrainn” is shown at the bottom of the word cloud which is a user tag name and not so useful for our project.

- Words like ’Flag’, ’Link’, ’error’ can be seen in figure 3.2
• words like '2010', '21st December', 'april' are present in the word cloud as they are the dates when a particular comment has been posted.

**HTML Markups** The first thing I had to do was remove all the HTML and Java script Markups / tags such as `<div>`, `<script>`, and `<link>` present in the data by using beautiful soup functionality.

**Tokenization** It is a process of converting a huge block of text into words, phrases, special characters, and digits called as tokens. These tokens are then added to a list so that further data cleaning and algorithms can be performed on them. Tokenization is performed by using the python split functionality, it basically splits each word or special symbol which is encountered after space.

**Lower case conversion** All the tokens are then converted into lower case so that all the data will be not case sensitive which can again create problems for the algorithm.

**Type checker** It is a process of using a dictionary to check if a particular word is valid or not. This task has been carried out by using the python package PyEnchant and either auto-correct the misspelt words or removed all the words which are not valid.

**Stemming** For grammatical reasons a user might use different forms of a single word, but none of them would be use full for the training the system, hence all the words are reduced to the root/base.

**Stop word removal** Stop Words are words which do not have any important significance to the mining process, such words are filtered out of the corpus. There were few words which were specific to this data set which had to be added to the normal English words stop list such as Link, album, lyrics, song, error, log.
**Punctuation removal** All the punctuations and special characters had to be removed from the corpus using the following regular expression $(\text{^A-}\text{Za-z0-9})$ which basically means any character which is not present between $A - Z, a - z, 0 - 9$ is substituted by a space.

**User Tag names removal** The data set of interpretations had user tags for each comment which were again of no significance to the system. So written a regular expression to remove all words which started with the symbol ‘@’.

**Digits removal** All the numerical data was removed by using the python’s substitute functionality such that all the digits are replaced by a space.

```
only one notice not lyrics flesh which

ya
thought ya
might like go show
feel warm thrill confusion
space cadet glow
tell something eluding sunshine
what you expected see
find out what behind cold eyes
just have claw your way through disguise

lyrics provided are from flesh question mark side four

Kind important detail
```

Figure 3.3: Interpretations data after cleaning
Figure 3.4: Lyrics data after cleaning

Figure 3.5: Word Cloud of lyrics data after cleaning
Figure 3.6: Word cloud of Interpretations data after cleaning
Chapter 4

Design & Implementation

The system is designed in such a way that both lyrics and interpretations are always considered differently. The lyrics undergo different data cleaning process and the interpretations undergo different data cleaning procedure. Lyrics are subjected to sentiment analysis whereas the interpretations are subjected to topic modelling as shown in Figure 4.1.

![Figure 4.1: Methodology](image)

4.1 Sentiment Analysis:

Finding the sentiment of the song / album looked like a good idea since tagged data was not available. The sentiment of the song / album can add the value to the classifying tag. So when a particular album is selected all the songs present in that folder are extracted into a bag of words list to perform sentiment analysis. To train the Naive Bayes classifier we
have used most frequently occurring positive and negative word list. Then each word from
the album lyrics was classified either as positive or negative. If the polarity of the song was
greater than 0 then it is classified as a positive song else if the polarity is less than zero then
it is classified as negative song.

![Figure 4.2: Sentiment Analysis Result](image)

### 4.2 Topic Modelling:

Finding the latent topic in a cluster of documents manually is a hard task. Topic modelling
is a statistical model which has been widely used to discover latent topics in collections
of documents. The main aim of this project is to find the underlying mood of an al-
bum so as to recommend similar albums to users. Hence topic modelling algorithms are
a perfect fit for our system. The chosen topic modelling algorithms are Latent Dirichlet
Allocation(LDA) and N-Grams algorithms.

**Latent Dirichlet Allocation** LDA traverses through each document, and randomly as-
sign each word in the document as one of the topics. Since it’s random, it is not a good
way to find latent topics, so as to refine it, LDA traverse again through each word in the
document and for each topic which were taken randomly before, we compute two things:

- \( p_1 = p(\text{topic } t \mid \text{ document } d) \) = it calculates the percentage of words in document d
  that are currently assigned to a particular topic t, and

- \( p_2 = p(\text{word } w \mid \text{ topic } t) \) = it calculates the percentage of assignments to topic t over
  all documents that come from this word w.
Then we reassign \( w \) to a new topic, where we choose topic \( t \) with the best probability by multiplying \( p_1 \) and \( p_2 \). The previous step is repeated a chosen number of times preferably larger, after a certain point of time the topics stop changing which shows that the topics are chosen are correct.

**Ngrams** It is similar to the LDA except that it uses \( N \) words combined to assign the topics whereas LDA uses a bag of word approach i.e one word at a time.

**Classification** Classification of the topics was one of the major roadblocks faced during the project. I have researched a lot and found out many techniques to classify but all of them required annotated data. Hence finally decided to come up with my own classification algorithm. A paper on human emotions [5] helped in deciding on the 5 basic human emotions such as happy, sad, anger, grief, and romantic.

These are the steps for the algorithm:

- Created 5 lists with the names happy, sad, anger, grief, and romantic.
- Searched for the synonyms of each category by using synset from NLTK wordnet and added to the lists.
- Then took each word in each topic from the topic model result and found out the root or synonym of them and matched them with the 5 lists.
- if there is a match the counter increases.
• Averaged the count for each category.

• Assigned a weight for the sentiment analysis of the album and weights for each category.

• Finally calculated the average of the weights and selected the highest average category and classified the album into that category.

\[
\begin{align*}
\text{Happy: } & 0.21381 \\
\text{sad: } & 0.21614 \\
\text{Anger: } & 0.21226 \\
\text{Grief: } & 0.17889 \\
\text{Romantic: } & 0.17889
\end{align*}
\]

\text{Sentiment: Negative}

\text{The Album's Tag is Sad}

Figure 4.4: Classification Result

As shown in Figure 4.4, Sometimes there might be a clash with the weights between any two moods. If the top mood is clashing with another mood, then to resolve this situation, the weight for the topic modelling result will increase by 1 so as to provide a clear result.
Figure 4.5: Final Output
Chapter 5

Results

Since the data was not tagged or annotated, the results could not be tested in the traditional way. So, Album Moods of the data were parsed from a site called allmusic with the help of beautiful soup package and have been written to a file. Most of the moods were very synonymous i.e Happy = Cheerful, Delighted, Joyful, Chirpy. So had to transform those moods into the chosen classification moods with the help of NLTK synset. Finally checked if the system tags were concurrent with the allmusic moods.

<table>
<thead>
<tr>
<th>Happy</th>
<th>Angry</th>
<th>Sad</th>
<th>Romantic</th>
<th>Grief</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Music</td>
<td>78</td>
<td>44</td>
<td>60</td>
<td>45</td>
</tr>
<tr>
<td>Proposed System</td>
<td>85</td>
<td>48</td>
<td>50</td>
<td>46</td>
</tr>
</tbody>
</table>

As shown in Table 5.1, Our system was on par with allmusic website system, it showed an accuracy of 89.4%.
Figure 5.1: Comparison
Chapter 6

Conclusion

6.1 Current Status

Developed a system which collects data, cleans the data, gives the sentiment of a song or an entire album and finds the latent mood of the album.

6.2 Future Work

The project gives good accurate moods of the album. However, the results can be improved by performing the following tasks.

- Annotate the data with the help of human subjects.
- Increase the data set for a much better system performance.
- Design & Implement an application which when integrated automatically goes through the user playlists and gets similar tracks
- Design & implement a sarcastic method to check how sarcastic the songs are.
- Try to integrate the audio tones into the classification process so that there will be a much better recommendation.
- Increase the number of emotions such as aggressive, Sentimental, Sexy.
6.3 Lessons Learned

- Interpretations are much better than lyrics to find the latent topics/subject of a song.

- You need to have annotated data for a much better system performance.
Bibliography


