The Stony Brook Glad All Over Machine Project Proposal

Trung Nguyen - 111752939
Anh Quang Do - 110922124
Nam Nguyen - 111171365

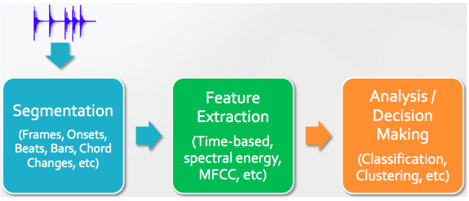
October 23, 2017

1 Background

1.1 Music information retrieval (MIR)

MIR is an interdisciplinary research that lies on the edges of musicology, psychology, signal processing, and machine learning.

Basic of a MIR system, including segmentation, feature extraction, analysis is represented as in the figure below.


Segmentation
(Frames, Onsets, Beats, Bars, Chord Changes, etc)
Feature Extraction
(Time-based, spectral energy, MFCC, etc)
Analysis / Decision Making
(Classification, Clustering, etc)

Basic of a MIR system

1.2 Automatic Evaluation of Karaoke Singing

Many of nowaday karaoke systems have a scoring feature to evaluate singers' performance. However, these rating is poorly constructed and not matched with human rating.

In this project, we focus on the vocal quality of the singing and train the machine to distinguish between good and poor singing. This can be done by recalling features of the singing, such as enthusiasm, emotion, pitch, volume, rythm, melodic similarity measures, etc.

1.3 Feature representation

Music analysis often requires some summarising and is achieved by feature extraction. One common feature extracted is the Mel-Frequency Cepstral Coefficient (MFCC).

1.3.1 Feature extraction

There are many features that can be extracted from music signal. These features can be categorized into: reference features, content-based features and text-based features. Reference features can be those relating to social interactions, e.g. followers, performance rating in, for example, soundcloud. Text-based features includes lyrics, interview, etc. Our approach will be based on content-based features, extracted from the wave signal, e.g. pitch, rythm, etc. and we might use reference features as ground truth for our machine-learned ranking method.

1.3.2 Mel-Frequency Cepstral Coefficient (MFCC)

The content-based features are calculated from low-level signal features (sometime refered as extraction methods), the most important of which is MFCC. MFCC and its derived features (such as "anchor space"[1]) have been shown to give good performance for a variety of audio classification tasks. MFCCs capture the short-time spectral shape, which carries information about instrumental timbres or the quality of a singing.

MFCCs are increasingly finding uses in music information retrieval applications such as genre classification, audio similarity measures, etc.

2 Related work

Music similarity metrics is a research aiming to calculate the similarity between songs or artists, comparing their performance. In [1], authors employ a feature, derived from MFCC, called 'anchor space', which uses musical categories and well-known anchor artists as convenient reference points for describing features of the music. It is inspired by a fold wisdom such as "Jeff Buckley sounds like Van Morrison meets Led Zeppelin, but more folky." Other approaches for song similarity are to embed songs into a Euclidian metric space and do some distance-based analysis and clustering [3].

For evaluation of karaoke singing, there is an approach based on the perception of singing enthusiasm [4]. The authors argue that karaoke is the form of entertainment for amateur so enthusiasm is a good criteria to evaluate them. They identified three acoustic features relevant to such perception: A weighted power, "fall-down", and vibrato extent, developed a system for evaluating singing enthusiasm, and obtained a correlation coefficient of 0.65 between the system output and human evaluation. In our point of view, their method can be considered as score-based ranking. In [5], the authors proposed a score combination from pitch-based, volume-based, and rhythm-based rating, with a reference specified karaoke song to evaluate a piece of singing. This approach is also score-based ranking. In [6], the authors used HMM as a statistical music recognition model for automatic scoring of karaoke computer games. The musical features they employed are Pitch & Pitch Error, Accent, Zero-Crossing Rate, Root-Mean-Squared Energy.

Another effort for analyzing the singing voice is made in [7], in which the authors reported good results of a system for classifying "good" and "poor" singing based on

SVM. In [8], the authors proposed a categorization and segmentation system for singing voice expression using pre-defined rules and HMM. There is another approach for automatic scoring of singing voice based on melodic similarity measures [9]. In [10], a method of evaluating singing skills that does not require score information is represented. The authors used pitch interval accuracy and vibrato as acoustic features to evaluate singing. The approach was then tested by a 2-class (good/poor) classification test with 600 song sequences, and achieved an average classification rate of 83.5%. There is an approach for song classification based on perception of emtion [11].

3 Our Approach

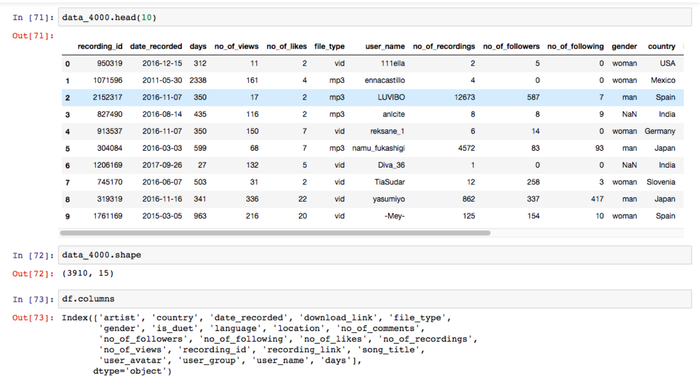
3.1 Dataset

This project requires a large amount of audio data. For this reason, we visited several karaoke websites to look for recording data and among them, *Redkaraoke* was found to be one of the largest sites for online recording worldwide which was created in 2007 and has more than 70000 songs (normally a karaoke website has 15000 songs). Its users are also very diverse in regions. Therefore we think this website can provide good data for our purpose.

We first chose a song, which is *"My heart will go on"* by Celine Dion for our experiment. The reasons behind this choice is that it is one of the most popular and most recorded songs, its level of difficulty is relatively high, and we all like this song so we don't mind hearing it over and over again. The url for this song is: `https://www.redkaraoke.com/karaoke/celine-dion/my-heart-will-go-on/11827`

We then built a scraper using *BeautifulSoup4* library to extract the recordings and users' information. For this *"My heart will go on"* song, we were able to sucessfully extracted around 4000 recordings, around half are .mp3 files and the rest are .mp4 files. We first downloaded all the .mp3 files, which take around 7.5GB of data storage. The scraper we built can extract information for other songs as well, but we think this amount of data is enough for us to experiment and we can always extract information for more songs whenever needed.

Aside from the audio files, text information from the recordings and the users were also extracted and saved as a csv file shown below:



```
In [71]: data_4000.head(10)
Out[71]:
```

	recording_id	date_recorded	days	no_of_views	no_of_likes	file_type	user_name	no_of_recordings	no_of_followers	no_of_following	gender	country
0	950319	2016-12-15	312	11	2	vid	111ella	2	5	0	woman	USA
1	1071596	2011-05-30	2338	161	4	mp3	ennacastillo	4	0	0	woman	Mexico
2	2152317	2016-11-07	350	17	2	mp3	LUVIBO	12673	587	7	man	Spain
3	827490	2016-08-14	435	116	2	mp3	anlcite	8	8	9	NaN	India
4	913537	2016-11-07	350	150	7	vid	reksane_1	6	14	0	woman	Germany
5	304084	2016-03-03	599	68	7	mp3	namu_fukashigi	4572	83	93	man	Japan
6	1206169	2017-09-26	27	132	5	vid	Diva_36	1	0	0	NaN	India
7	745170	2016-06-07	503	31	2	vid	TiaSudar	12	258	3	woman	Slovenia
8	319319	2016-11-16	341	336	22	vid	yasumiyo	862	337	417	man	Japan
9	1761169	2015-03-05	963	216	20	vid	-Mey-	125	154	10	woman	Spain

```
In [72]: data_4000.shape
Out[72]: (3910, 15)

In [73]: df.columns
Out[73]: Index(['artist', 'country', 'date_recorded', 'download_link', 'file_type',
                'gender', 'is_duet', 'language', 'location', 'no_of_comments',
                'no_of_followers', 'no_of_following', 'no_of_likes', 'no_of_recordings',
                'no_of_views', 'recording_id', 'recording_link', 'song_title',
                'user_avatar', 'user_group', 'user_name', 'days'],
               dtype='object')
```

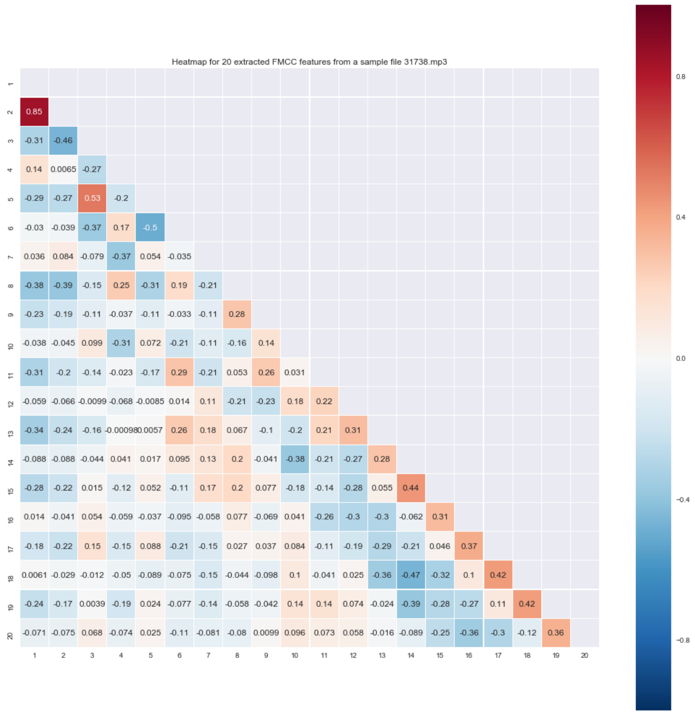
3.2 Tools and libraries

There are many tools available for audio analysis in python, many of which are available to be evaluated in [2]. Among them, we chose the *librosa* library to experiment because of its instruction availability.

3.3 Method and Experiment

Our workflow follows the *Music information retrieval process* presented in section 1.1: First we segment the music (since each audio file has different length) then extract the feature and finally use machine learning methods to evaluate/group the recordings.

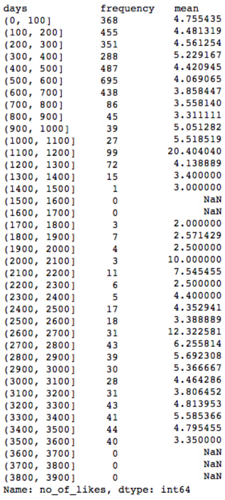
We had experimented extracting the 20 MFCC features from a sample audio file in our dataset. We will continue to find more information about these features to decide which are useful for our problems.



Heat map of 20 MFCC features extracted from a sample audio file in our dataset

Once this step is finished, we may experiment building our model based Metric Learning to Rank [12] to automatically learn the distance metric, and compare it with distance in Euclidian metric space. This distance metric that have been learned will then be used in a learning to rank algorithm, based on Structural SVM.

Besides we want to extract the information not only from the audio files (content-features), but also from the information in the we retrieved (reference feature). These features may act as the groundtruth for our model. One metric which we find interesting is the number of likes. Below is the statistics for the average number of likes in each time frame of 100 days. Except from some outliers (which turns out to be good singers who records several recordings in a short time), the average numbers of like is quite stable. From this analysis, we may conclude that this information may not heavily dependent on the recording's age, and therefore the number of likes may be good good for evaluating whether a recording is good or bad.



```
days            frequency   mean
(0, 100]        368          4.755435
(100, 200]      455          4.481319
(200, 300]      351          4.561254
(300, 400]      288          5.229167
(400, 500]      487          4.420945
(500, 600]      695          4.069065
(600, 700]      438          3.858447
(700, 800]       86          3.558140
(800, 900]       45          3.311111
(900, 1000]      39          5.051282
(1000, 1100]     27          5.518519
(1100, 1200]     99         20.404040
(1200, 1300]     72          4.138889
(1300, 1400]     15          3.400000
(1400, 1500]      1          3.000000
(1500, 1600]      0               NaN
(1600, 1700]      0               NaN
(1700, 1800]      3          2.000000
(1800, 1900]      7          2.571429
(1900, 2000]      4          2.500000
(2000, 2100]      3         10.000000
(2100, 2200]     11          7.545455
(2200, 2300]      6          2.500000
(2300, 2400]      5          4.400000
(2400, 2500]     17          4.352941
(2500, 2600]     18          3.388889
(2600, 2700]     31         12.322581
(2700, 2800]     43          6.255814
(2800, 2900]     39          5.692308
(2900, 3000]     30          5.366667
(3000, 3100]     28          4.464286
(3100, 3200]     31          3.806452
(3200, 3300]     43          4.813953
(3300, 3400]     41          5.585366
(3400, 3500]     44          4.795455
(3500, 3600]     40          3.350000
(3600, 3700]      0               NaN
(3700, 3800]      0               NaN
(3800, 3900]      0               NaN
Name: no_of_likes, dtype: int64
```

Average number of likes for the recordings in each time frame of 100 days

Separating the vocal and non-vocal segments of song as a preprocessing step is also considered. However this turns out to be a hard problem: as oppose to professional multichannel audio recordings, our mp3 files only have 1-2 channels which is very difficult to separate the vocal and non-vocal segments. We will continue to find if there is a solution for this problem.

3.4 Evaluation

References

[1] E.L. Hall, J.B.K. Tio, C.A. McPherson, F.A. Sadjadi Measuring Curved Surfaces for Robot Vision *Computer* Vol.15, no. 12, pp. 42-54 (1982)

[2] Z.Zhang Flexible new technique for camera calibration IEEE Transactions on Pattern Analysis and Machine Intelligenc, vol. 22, no. 11, pp. 1330-1334 (2000)

[3] J.Bouguet Camera calibration toolboxfor MATLAB (2010) http://www.vision.caltech.edu/bouguetj/calib doc/

[4] J.Falcao, N.Hurtos Projector-camera calibration toolbox (2009) http://code.google.com/p/procamcalib/

[5] Paul M.Griffin, Lakshmi S.Narasimhan and Sound R. Yee Generation of uniquely encoded light patterns for range data acquisition *Pattern Recognition* Vol. 25, pp. 609-616 (1992)

[6] Yi-Chih Hsieh Decoding structured light patterns for three-dimensional imaging systems *Pattern Recognition* Vol. 34, pp. 343-349 (2001)