Detecting Musical Key of Songs

Yuqing LU
SID 57861450
Department of Electrical Engineering
City University of Hong Kong
yuqinglu4-c@my.cityu.edu.hk

Abstract

This project proposes a method to detect musical key of songs based on the average pitch chroma vector and some metadata, which can be extracted from the Million Song Dataset. However, considering of the music theory and experimental performance, all the data used for training, validation as well as testing has been relabeled and recalculated into another format. Strictly following the class principle and grading criteria, the experiment results are presented in both technical and practical aspects in this report.

1 Related work

2 Methodology

The methodology of this project mainly includes two aspects: musical theory and mathematics. The concept about musical key including chords and tonality, and its relationship to average pitch chroma will be introduced in the musical theory aspect. On the mathematical side, the project applies softmax multi-classification and regression to implement algorithm.

2.1 Musical methodology

Musical Key refers to a specific set of pitches or notes that form the foundation of a piece of music, which determines the overall pitch relationships, harmonies, and melodies within a musical composition. In traditional music theory, there are 12 major and 12 natural minor tonics. Each tonic corresponds to a specific pitch or note within the 12-tone chromatic scale.

(a) C major scale (b) A minor scale

Figure 1: example of Musical keys

There are 2 modes (major or minor) and the 12 tonic (C, Db, D, Eb, E, F, F#, G, Ab, A, Bb, or B). However, the C major and A minor scales, shown in Figure 1 above, contain the same notes, but actually represent different keys due to the different tonic notes and corresponding chords.

Therefore, the complete set of musical keys can be described in the Table 1 provided below. To detect them, the underlying idea is to first classify the input data into two dual-key classes that share eight

Preprint. Under review.

common notes. Based on this initial classification, a more specific classification is then performed to determine the precise key or mode that represents the song as the final musical key.

Table 1: Major and minor keys which contain the same notes

Major	C	Db	D	Eb	E	F	F#	G	Ab	A	Bb	B
Minor	A	Bb	B	C	Db	D	Eb	E	F	F#	G	Ab

Pitch chroma refers to a representation of the relative strength of each pitch class present at each time window in a song. For example in the figure 2 below, I visilize the parameter 'segments_pitches' in dataset of First 100 chroma features generated from the input song "Amor De Cabaret". It is clear that the lighter grids correspond to larger possible pitch values. However, I choose to average the chroma vectors given throughout the length of the song as the extracted feature, which is a vector with length of 12 for one song.

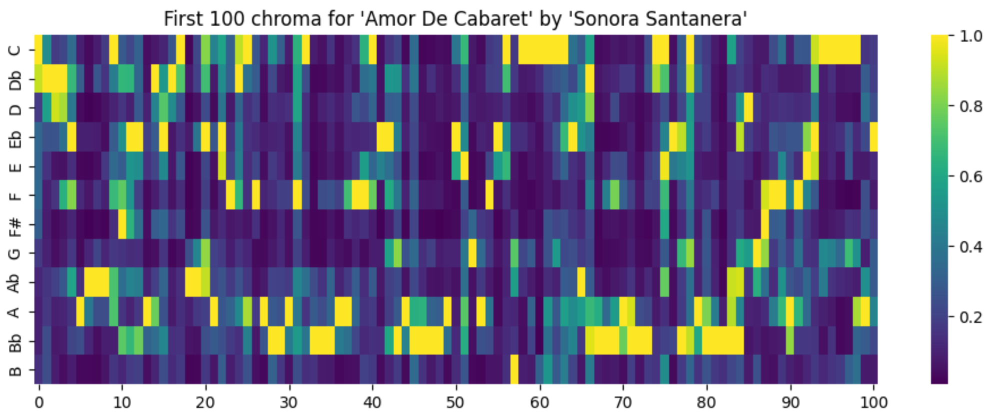

Figure 2: First 100 chroma features generated from the input song "Amor De Cabaret".

2.2 Mathematical methodology

Based on the musical methodology, I designed a softmax regression model for first training on the 12 classes which represent keys containing the same eight dominant notes. Each includes one major and one minor key, mentioned in Table 1. Next, based on each dual-key classes, one logistic regression model (softmax with only two classes) is trained to distinguish the two possible keys within each, which produces a total of 12 new classifier models.

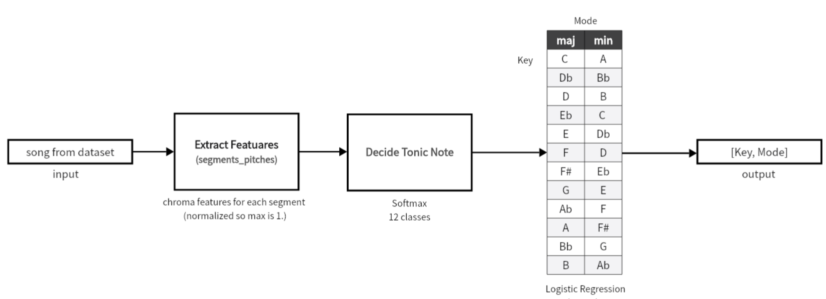

Figure 3: Pipeline of the experiment.

To implement the pipeline as figure 3 shown above and our machine learning class principle, I wrote softmax multi-classification algorithm without library like pyTorch. The mathematical formulas I used are as below:

Linear weighted combination:

$$z_k = w_k^T x + b_k = \left(\sum_{i=1}^{M} w_{k,i} x_i \right) + b_k \tag{1}$$

Softmax activation function:

$$softmax\,(z_k) = a_k = \frac{e^{z_k}}{\sum_{i=1}^{K} e^{z_i}}, \quad k = 0, 1, \ldots, K-1 \tag{2}$$

Cross entropy loss function:

$$L(\hat{y}, y) = -\sum_{k=1}^{K} y_k \log a_k \tag{3}$$

Backpropagation:

$$w_k = w_k - a\frac{\partial J}{\partial w_k} = w_k - a\frac{1}{N}\sum_{i=0}^{N-1} (a_k - y_k)\, x, \quad k = 0, 1, \ldots, K-1 \tag{4}$$

$$b_k = b_k - a\frac{\partial J}{\partial b_k} = b_k - a\frac{1}{N}\sum_{i=0}^{N-1} (a_k - y_k)\,, \quad k = 0, 1, \ldots, K-1 \tag{5}$$

To sum up, considered of the musical key, two parameters are involved: the key itself and the mode. In this experiment, a total of 13 weight parameters are generated. One weight parameter is used to classify the dual-key classes, while the remaining parameters are utilized to make predictions based on the initial classification of the dual-key classes.

3 Experiment

All the codes and data I used are attached to this report, which are strictly following our class project principle. I will introduce it steps by steps.

3.1 Data preparation

Data is divided into three documents, and the paths of them are as below.

```
./MillionSongSubset  original dataset of 10,000 songs
./ValidSongData      training and validation dataset of 3,729 songs
./TestSongData       testing dataset of 31 songs, 1 for my own dataset
./hdf5_getters.py    provided by dataset to open h5 format file
./prepare_data.py    set thresholds to filter data
```

This project makes use of the Million Song Dataset [1], which is a massive collection of features and meta- data for one million songs, provided by The Echo Nest. However, due to the large size of the entire dataset (>300GB), this project was only able to use a 10,000 song subset. Considered of two parameters of 'key_cof' and 'mode_conf' in dataset, I designed a threshold of 0.5 to filter the subset. Therefore, the training data is limited to 3729 songs.

One song in the test dataset should be noticed is that it is a popular song from recently TV series in 2022. In this case, the experiment weights were applied to detect the musical key of the song, serving a real and practical purpose in a composition task. This application demonstrates the practical relevance and potential usefulness of the experiment in the context of contemporary music production.

3.2 Feature extraction

As mentioned in musical methodology 2.1, there are two tasks involved. One is to extract the average pitch chroma from the dataset as our input feature. The other task is to relabel the data into 12 dual-key classes according to the table 1.

```
./extract_feature.py     extract feature and relabel data in csv file
./data_feature.csv       training and validation relabel dataset table
./test_data_feature.csv  testing relabel dataset
```

The relabeled data example is as Table 2 below, each musical key is described as a tuple, where the first number represents the dual-key class, and the second number represents the mode. It is renamed in csv file, which is 'feature_a' and 'feature_b'.

Table 2: Relabel principle table

Major	(0,1)	(1,1)	(2,1)	(3,1)	(4,1)	(5,1)	(6,1)	(7,1)	(8,1)	(9,1)	(10,1)	(11,1)
Minor	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)	(5,0)	(6,0)	(7,0)	(8,0)	(9,0)	(10,0)	(11,0)

3.3 Softmax model creation and training

In this step, I build a class of 'MySoftmaxModel' to implement the multi-classification algorithm, where the functions follow the math formula in mathematical methodology 2.2. The best weight is saved as xlsx format, for 'feature_a' weight is a shape of (12, 12) matrix, and other 'feature_b' weights are shape of (2,12) matrix.

```
./train_validation.py
./weights_feature_a/acc_0.7_seed_3_lr_0.05_iteration_4767.xlsx
./weights_feature_b/feature_a0_C_or_Am/acc_0.97_seed_3_lr_0.05_iteration_283.xlsx
......
./weights_feature_b/feature_a11_B_or_Abm/acc_0.95_seed_3_lr_0.1_iteration_102.xlsx
```

Considered of the data have a parameter of 'confidence' which means the label is not absolutely groundtruth, the training and validation result are as Table 3 and 4 below.

Table 3: feature_a result

dual-keys	**precision**
C/Am	0.73
Db/Bbm	0.7
D/Bm	0.64
Eb/Cm	0.71
E/Dbm	0.7
F/Dm	0.66
F#/Ebm	0.63
G/Em	0.77
Ab/Fm	0.68
A/F#m	0.71
Bb/Gm	0.72
B/Abm	0.5

Table 4: feature_b based on feature_a result

musical keys	**precision**	**musical keys**	**precision**
Am	0.91	Ebm	0.89
C	0.99	F#	0.97
Bbm	0.81	Em	0.91
Db	0.89	G	0.99
Bm	0.87	Fm	0.99
D	0.93	Ab	0.94
Cm	0.9	F#m	0.92
Eb	0.96	A	0.99
Dbm	0.99	Gm	0.82
E	0.95	Bb	0.92
Dm	0.82	Abm	0.99
F	0.98	B	0.95

3.4 Testing result

I applied the weights from the experiment to conduct two different tests. The first test involved detecting the song data from the Million Song Dataset. Another test is to detect the musical key of a popular song in a practical context.

3.4.1 Detecting the song data from the Million Song Dataset

I randomly selected 30 songs from the remaining portion of the Million Song Dataset. However, the 'key_conf' and 'mode_conf' values for these songs were not higher than 0.5. Since there was a scarcity of data that met our training threshold, I decided to set a threshold range between 0.48 and 0.5 for these songs to test. This also allowed me to test the generalization capability of the experiment's weights and assess their performance on songs with slightly lower confidence values.

```
./data_feature.csv      pitch chroma data
```

The Table 5 below showcases the results of the 30 songs. It is important to note that these songs were selected based on a threshold range between 0.48 and 0.5 for 'key_conf' and 'mode_conf' values. This threshold range differs significantly from the training data, where the values were predominantly larger than 0.5.

Table 5: 30 songs detecting result

song_index	**test_result**	**song_index**	**test_result**	**song_index**	**test_result**
0	True	10	False	20	True
1	True	11	True	21	False
2	True	12	False	22	False
3	True	13	False	23	False
4	False	14	False	24	False
5	True	15	True	25	True
6	False	16	True	26	False
7	False	17	True	27	True
8	False	18	False	28	False
9	True	19	False	29	False

3.4.2 Detecting a popular song in practice

This test applies the weight generated in the training to a practical composition case, and the result can be remarkable and inspiring. I randomly chose the song named "Qinghong" from the TV series "Lighter and Princess".

```
./test_qinghong/qinghong.mp3
./test_qinghong/qinghong_chordsequense.rs
./test_qinghong/qinghong_dataset_test.xlsx   pitch chroma data
./test_qinghong/qinghong_musicsheet.mscz     piano composition
```

To conduct this interesting experiment, the initial step involves writing the chord sequence for the song. If you are familiar with musical theory, this task is relatively straightforward. However, you can utilize software to assist you in generating the chord sequence instead of doing it manually. The chord sequence result is same, a clip of it is as Figure 4.

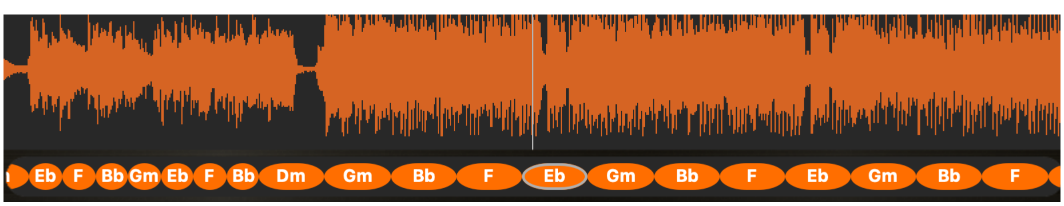

Figure 4: Part of the chord sequence of qinghong.mp3

After obtaining the chord sequence, the next step is to write the pitch chroma and calculate the average number based on this sequence. The approach I used is to assign value of 1 to the pitches that correspond to the chord tones, while assigning a value of 0 to the remaining pitches. The example of this method is as Table 6.

After calculation, the average pitch chroma vector is as Table 7 below.

Table 6: Example of writing pitch chroma based on chord sequence

chord	C	Db	D	Eb	E	F	F#	G	Ab	A	Bb	B
Dm	0	0	1	0	0	1	0	0	0	1	0	0
Bb	0	0	1	0	0	1	0	0	0	0	1	0
Gm	0	0	1	0	0	0	0	1	0	0	1	0

Table 7: average pitch chroma vector of qinghong.mp3

C	Db	D	Eb	E	F	F#	G	Ab	A	Bb	B
0.27	0	0.497	0.24	0	0.51	0	0.45	0	0.29	0.67	0

Input the verage pitch chroma vector into our algorithm, the **result is (10, 1)**, which denotes **Bb major** according to Table 1, 2. And the result is **absolutely correct!**

```
./test.py
```

Then we can write the music sheet with musical key of Bb to cover this song as the Figure 5 below, which ensures that the music sheet accurately reflects the original song while being performed in the desired key.

Figure 5: Music sheet of Bb major in qinghong.mp3

3.5 Source code of project

The documents for this project can be found at

`https://github.com/Aprilsmile1/detect-musical-key.git`

4 Conclusion

This project proposes a method based on supervised learning principles and music theory that can make predictions using a chroma feature vector representation of a song. Its good performance in practice indicates that it may be an effective way to assist us in composing songs.

References

[1] Thierry Bertin-Mahieux et al. "The Million Song Dataset". In: *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*. 2011.