CS230

Chord Recognition with Deep learning

Chen Ji
Department of Music
Stanford University
jjjjjc@stanford.edu

Abstract

This project focus on the Automatic Chord Recognition (ACE) with Deep learning. The motivation of this topic mainly generated by two facts. Firstly, a lot of music lovers have no capability to recognize the chords by themselves so that this ACE system could have a huge market. Also, Deep Learning has been a heated top these days with the development of GPU and the computational power, and Deep Learning has achieved great achievements in computer vision. So why not try Deep Learning in computer audio or computer music. In this project, the work-flow of building a AE system is reviewed, which consists of Feature extraction stage and classification stage. Then, according to the Feature extraction stage, the music signal processing approaches such as STFT, CQT and chromagram were studied and computed by using Python, then the chromagram enhancement was also applied. In the classification stage, the theoretical knowledge was studied and also each model was computed by using popular Deep learning packages, such as Tensor Flow, TFlearn and sklearn etc.The abstract should consist of 1 paragraph describing the motivation for your paper and a high-level explanation of the methodology you used/results obtained.

1 Introduction

With the increasing availability of music data and demands of capturing useful information from music contents, recent years have witnessed a burgeoning development of music information retrieval (MIR). Automatic Chord Estimation (ACE) is one of long-standing tasks to be solved and further optimized. The goal of ACE is to segment a music file into several parts and then transcribe each part into its corresponding chord sequence and also system is required to distinguish chord and non-chord (silence, natural sound, environmental noise, etc.). The following Figure 1.1 illustrates the chord recognition process both for human ear and computer ACE system. Admittedly, chord recognition by human ears is the directest and most traditional way, but it requires sophisticated musicianship. People who take systematic music learning and also take large amount of practice can only have ability to have the "ear" to recognize the chord, although sometimes the mistakes may frequently occur when similar chords occur, such as C_{major7}, C_{major9} and C/G . Thus, the automatic estimation for chords are highly required.

2 Related work

The extracted features should be input into the classification system. Before machine-learning and deep-learning methods became popular, templatematching classification was used [16], where the standard PCPs for different chords are created and then by measuring the distance between standard one with the real PCP, the most possible chord can be matched. Later, the probabilistic models of

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

chord transitions were used. Hidden Markov Models (HMMs) [1, 2] is the most used within this type of classification. In terms of deep-learning method, unsupervised learning algorithm, such as deep autoencoder [3] and restricted Boltzmann machine (RBM) [4] can be used for feature learning. If using supervised learning, fully-connected neural network (FCNN), deep belief network (DBN) [4] were used; recurrent neural network (RNN) and convolutional neural network (CNN) were also used for classification.

3 Dataset and Features

The training data is audio files of different songs while the ground truth is the annotation of each chord. Three dataset are used:

- 180 tracks from the TheBeatles180 dataset[1]
- 29 tracks from JayChou dataset(JayChou29)[2]
- 20 tracks from a Chinese pop song dataset(CNPop20)[3]

The first dataset is from Isophonics dataset, where contains many annotated albums from Michael Jackson, Queen, Carole King, and the Beatles and this dataset is contributed by the Center for Digital Music (C4DM) of Queen Mary, University of London; then the second and third datasets are contributed by Junqi Deng from The University of Hong Kong.

Data format explanation
The annotation is in csv form, which can be converted to txt files in the later stages. There are three columns in the annotation file, the first column gives the starting time, the second column shows the ending time and the third column is the chord annotation. Look at the first line, it means that from 0s to 2.6122167s the audio is in "no chords". For the second line, it means that from 2.612267s to 11.459070s the audio is in E chord. By using this method, the whole file records all the chords and their corresponding time period. The significance of recording the starting and ending time is to segment the audio into corresponding parts, so that we can make sure that training data with its ground truth label is precise. Also, it can be noticed that there are some 7th chord ($E:7/3$) and chord inversions ($A:min/b3$), we can classify the $E:7/3$ into E major chord while $A:min/b3$ into A minor chord, because these special chords actually are modified based on major and minor chords. In addition to the "no chord", 25 types of chords should be considered, which can be listed: N, C, Cm, $C\sharp$, $C\sharp m$, D, Dm, $D\sharp$, $D\sharp m$, E, Em, F, Fm, $F\sharp$, $F\sharp m$, G, Gm, A, Am, $A\sharp$, $A\sharp m$, B and Bm.

```
0.000000 2.612267 N
2.612267 11.459070 E
11.459070 12.921927 A
12.921927 17.443474 E
17.443474 20.410362 B
20.410362 21.908049 E
21.908049 23.370907 E:7/3
23.370907 24.856984 A
24.856984 26.343061 A:min/b3
26.343061 27.840748 E
27.840748 29.350045 B
29.350045 35.305963 E
35.305963 36.803650 A
36.803650 41.263102 E
41.263102 44.245646 B
44.245646 45.720113 E
45.720113 47.206190 E:7/3
47.206190 48.692267 A
48.692267 50.155124 A:min/b3
50.155124 51.652811 E
51.652811 53.138888 B
53.138888 56.111043 E
56.111043 65.131995 A
65.131995 68.150589 B
```

Figure 1: Original chord and its inversion

Audio files collection

[1] http://isophonics.net/content/reference-annotations-beatles
[2] http://www.tangkk.net/label/jaychou/
[3] http://www.tangkk.net/label/cnpop/

Because of the issues of copyrights, the Isophonics Beatles dataset, Jaychou dataset and Chinesepop dataset don't provide audio files. So manual work should be done to find the music files and match them with the chord annotations. I obtained the mp3 files by converting the Youtube videos into mp3 files with the free website[4], which can generate 320kbps audio files in mp3 form. This work is time-consuming, because the some of the Youtube videos are not the original released versions or original MV versions with additional scene sounds, where the starting and ending time has some differences with the chord annotation from the dataset. Thus I chose all the music matching with the chord annotations, while I did not include the music, whose starting and ending time has large deviation (exceeds 1 or 2 seconds) with the ground truth. Although the training data will be reduced because of dropping some of unmatched music, it is more important to ensure the quality of the data, instead of quantity. Finally, at this stages, the music files in mp3 form are prepared for the next stages.

4 Methods

There are two parts in chord recognition system: feature extractions and classification. In the feature extraction stage, STFT, CQT and Chromagram are used to extract the features. Then many approaches are used to improve the features : Tuning Compensation, Harmonic Percussive Signal Separation, Recurrence-based smoothing and Median filter. After that, i got the relatively clear features, then i send them into the classification system. Logistic regression, multiple-layer perception NN with different layers are used while deep belief networks are used. The DBN was introduced by Hinton and his team in, where he solved the problem of the training of the multi-hidden layer neural networks proposing a greedy algorithm that trains one layer at the time in a unsupervised manner. This is possible because they are defined as probabilistic generative models made by several layers of Restricted Boltzmann Machines (RBMs). As mentioned before, generative models, in contrast with the discriminative models such as the standard neural networks, are able to provide a joint probability distribution over labels and observable data. They are, thus, able to estimate both P(Label|Observation) and P(Observation|Label), while discriminative models are limited just to the estimation of the former P(Obsevation|Label). Because Restricted Boltzmann Machines (RBMs) are energy based models, in the following subsections, the relevant energy based model, Restricted Boltzmann Machines (RBMs) and how to train the Deep belief network as well as how it works.

5 Experiments/Results/Discussion

5.1 Logistic Regression

The first model i tried is Logistic Regression. For training and testing stages, train set is 80% while test set is 20%. The following table shows the different learning rate. Then, If looking at the influence

Learning Rate	0.2	0.1	0.05	0.02
Accuracy	0.5934	0.6523	0.6934	0.6934

Table 1: Accuracy of Logistic Regression model with different **learning rate**

of chromagram enhancement, then we set learning rate is 0.05 and other parameters the same. we can find that chromagram enhancement can improve the accuracy.

Chromagram enhancement	Yes	No
Accuracy	0.6923	0.4916

Table 2: Accuracy of Logistic Regression model **with and without chromagram enhancement**

Then, because TFlearn provides different kind of the optimizer of **Gradient Descent**, which explained the the theoretical part, such as Stochastic Gradient Descent (SGD), RMSprop, Adam, Momentum and AdaDelta. I tried them out with other parameters keeping the same as the last experiment with chromagram enhancement. I find that Adam optimizer gives the best result and I decide to use it in the latter experiments. Finally, we can get the result that accuracy could reach around 0.7 and the loss

[4]https://www.flvto.biz/youtube-to-mp3/

Optimizer	SGD	RMSprop	Adam	Momemtum	AdaDelta
Accuracy	0.6402	0.6223	0.6923	0.6612	0.6476

Table 3: Accuracy of Logistic Regression model with **different optimizers**

can be reduced to around 1.0. However, the condition is using 7k+ dataset and this is why we cannot get this point because we didn't get sufficient data.

5.2 Multi-Layer Perceptron Neural Network (MLPNN)

Because as previous model, learning rate setting as 0.005 gives good results and number of epochs setting is 800 because after trying this number makes sure the stable accuracy; and also, the chromagram enhancement is included. In this case, we could see that ReLU and Leaky ReLU have better performance of Sigmoid and Tanh. If the data has more distortions, the differences could be bigger. Thus, I decided to use "ReLU" in the experiments of MLPNN. "ReLU" mainly has two benefits: First, ReLU doesn't face gradient vanishing problem as with sigmoid and tanh function. Also, It has been shown that deep networks can be trained efficiently using ReLU even without pre-training. Second, If hard max function is used as activation function, it induces the sparsity in the hidden units.

	Sigmoid	Tanh	Relu	Leaky ReLU
Accuracy	0.6965	0.7234	0.7512	0.7413

Table 4: Accuracies of MLPNN models with **different activation functions**

Then let's use try different number of MLPNN and the other parameters are set as the same as above:

nHidden	3	5	10	15	30	100
Accuracy	0.7265	0.7514	0.7456	0.7510	0.7512	0.7511

Table 5: Accuracies of MLPNN models with **different nHidden**

Initially, I guessed that the larger the number of hidden layers, the higher the accuracy will be. However, 5 is enough for the number of hidden layers. Increasing the number of hidden layers, the accuracy will not be increase but wasting the computation power.

Another question also should be considered: whether the size of training data will influence the accuracies or not? then I did an experiment to test the influence the data size to the model performance. In this case, we say one set of data is in the form of [the integer $\in [0\ 25], a, b, c, d, e, f, g, h, i, j, k, l]$, where the first integer stands for the chord label and the later 12 numbers stand for 12 chromagram features. Let's try to training the model with set size of data in 100, 300, 500, 700, 900, 1100 and 1300. When the data size is very small like 100 sets, the accuracy

Data size (set number)	100	500	1000	1000	1300	1600	1900
Accuracy	0.3611	0.4714	0.6687	0.7125	0.6534	0.7156	0.7427

Table 6: Accuracies of MLPNN models by using **different size of training data**

is pretty low, that's because the model haven't learned the complete features of the chord distribution. Then with the increasing number of data size, the accuracy increased. However, when the data size continues to grow, the accuracy decrease a bit, which can be explained that some the data is deviated from the mainstream so that it makes the model feel confused. Then with the data size increasing, the accuracy becomes stable.

5.3 Deep Belief Network

Generally, there are two training stages of Deep Belief Network, which are pre-training and training. The first stage used unlabeled data, which aims to make the model be familiar with the data it going to learn. Then the second stage is normal training, which is basically the same process as the Logistic Regression and MLPNN models. This method could be similar to the human learning situation, where the students who preview the courses in advance will be easier to learn the class and the learning efficiency will be much better. When we compute the DBN, the fundamental setting is the same but here we need using RBM. Here I used the sklearn, which is also a machine learning package, where provides the DBN model. I used it directly. However, the accuracy is only tested as 60.5678%, which is lower Logistic Regression. Comparing the best performance of three neural networks: Comparing

Logistic Regression	MLPNN	DBN
0.6623	0.7265	0.605678

Table 7: Accuracies of three models in this project

the best performance of three neural networks: The MLPNN gives best result.

5.4 Evaluation in Real Situation

One more thing is that in the last section, deep learning library calculates the accuracy for us, but that calculation is based on "black" and "white", where if the prediction doesn't equal to the ground-truth, it will be counted as false. However, in music, some of chords sound similar and hamonious even if they are not the original. Here I want to define a new method to calculate the "**Accuracy**". If the chord is the same, count 100%; then if then recognize the major into minor or minor to major, count 80%; if root note of the predicted one is away from 1 semitone ,count 60%; if root note of the predicted one is away from 2 semitones ,count 40%; if root note of the predicted one is away from 2 semitones ,count 20%. If the situation mentioned above occurred together, the percentage should be multiplied. Finally, every mark is averaged into a final "Accuracy". I tried three types of music:

Mandarin pop song	English pop song	Classic piano song
75.6%	82%	71.6%

Table 8: Accuracies of three models in this project

6 Conclusion/Future Work

Conclusion 1: If the data is sufficient, the capability of Logistic Regression and MLPNN is similar According to the figures from the Tensorboard, we can see the expected accuracies and loss can reach the same level when data is sufficient. However, when the data is not sufficient, the MLPNN shows the better performance. However, the DBN is a good way to deal with the ACE problem.

Conclusion 2: Deep Belief Network is sensitive to the data, although it is a good model similar to the human pre-viewing
There are two training stages of Deep Belief Network, which are pre-training and training. The first stage used unlabeled data, which aims to make the model be familiar with the data it going to learn. Then the second stage is normal training, which is basically the same process as the Logistic Regression and MLPNN models. This method could be similar to the human learning situation, where the students who preview the courses in advance will be easier to learn the class and the learning efficiency will be much better. However, this method didn't achieve good results in this project. This can be caused by complicated training process of this method, so that subtle influence can be amplified. This model is sensitive to the data quality, and how much distribution for pre-training and post-training. The data type also should be considered. Due to the time limitation, deeper analysis of why this method cannot reach the expectation haven't been unfolded.

Conclusion 3: Pre-processing of data is important
Including basic feature extraction and the chromagram enhancement, they are pretty important in the deep learning system, because the system is sensitive to the data quality and efficiency. I tested the experiment group without chromagram enhancement, the accuracy is much worse than the group without chromagram enhancement. Imagine, if inputting the raw audio files, the model probably even harder to learn how to recognize the chord.

Conclusion 4: Chord recognition can be achieved by regarding it as a classification problem
As we can see from experiment results, even if the results are not good as the one attending music information retrieval, but it do deal with the problem. Thus we can prove that the chord recognition can be seen as a classification problem and deep learning is a good way to deal with the classification problem. However, what we need to notice is that this project only considers the major and minor chords, which indicates that there are only 25 types in classification problem. Then, if we focus on the large vocabulary chord recognition problem, the number of types of chords will dramatically increase (around 100 or even more). At that time, the classification can be really hard to achieve, so other methods should be used, like finding the relationship between the adjacent chords so that the neural net can learn the rules of chord regression, which is more intelligent and it could be close to the AI composition.
Three tasks I can come up as the future work, for the sake of improving the accomplishment of this project.
1. Improve the accuracy
Because the accuracies are still low compared to the results from the latest ISMIR conference results (85%), more work should be done for improving the accuracies. As discussed in the Section 7.1, the reasons can cause undesired accuracies could be 1) Quality of the music data 2) Absent of post-processing 3) Only trying out three types of neural networks. Thus, the music data should be selected more properly and also pre-processed better.
2. Work on the large vocabulary chord recognition
Apart from the improving the accuracies, this system can only work on the major and minor chords, which means it cannot recognize more fancy chords, such as: sus, major7, major9 etc. And these chords are sometimes the cores to make one piece of music attractive. Because the basic major and minor chords are very fundamental, probably could make people feel bored. Thus I plan to work on the large-vocabulary chord recognition as future work, also.

7 Contributions

I did the project individually.

References

[1] A. Sheh and D. P. Ellis, "Chord segmentation and recognition using em-trained hidden markov models," 2003.

[2] Y. Ueda, Y. Uchiyama, T. Nishimoto, N. Ono, and S. Sagayama, "Hmm-based approach for automatic chord detection using refined acoustic features," in Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pp. 5518–5521, IEEE, 2010.

[3] Y. Bengio et al., "Learning deep architectures for ai," Foundations and trends R in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009. [4]G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural computation, vol. 18, no. 7, pp. 1527–1554, 2006.