AUTOMATIC CHORD ESTIMATION ON SEVENTHSBASS CHORD VOCABULARY USING DEEP NEURAL NETWORK

Junqi Deng, Yu-Kwong Kwok

Department of Electrical and Electronic Engineering
The University of Hong Kong
{jqdeng,ykwok}@eee.hku.hk

ABSTRACT

This paper proposes an automatic chord estimation (ACE) system with a two-layer architecture. The first layer performs chord smoothing with "GMM + HMM" approach. Then given the results of the first layer, the second layer performs chord estimation using a deep neural network, which is trained on a well chord-type balanced dataset. The system accepts exactly the "SeventhsBass" vocabulary. Three approaches with different configurations of the system are compared with Chordino, which is probably the only both MIREX evaluated and "SeventhsBass" acceptable ACE system. Evaluation results on "The Beatles" dataset show that the best approach outperforms Chordino in the most difficult "SeventhsBass" metric in a significant way.

Index Terms— Automatic Chord Estimation, Deep Neural Network, Deep Belief Network

1. INTRODUCTION

1.1. Evolution in ACE Architectures

Automatic chord estimation has been studied for more than a decade. The "front-end + back-end" system architecture paradigm has been evolving from "pitch-class-profile (PCP) + template based smoothing" [1, 2, 3], to "PCP + hidden-Markov-model (HMM) smoothing" [4, 5, 6], to "PCP + Gaussian mixture model(GMM) + HMM" [7], to "PCP + hypothesis/test" [8], and to "PCP + GMM + dynamic Bayesian network" [9, 10]. Recently, following the success achieved in speech recognition, deep learning methods [11] start to form its shape in ACE, such as convolution neural network (CNN) based system [12], and deep belief network (DBN) - recurrent neural network (RNN) based system [13]. Deep learning enables hierarchical representations be learned from raw features and makes better classifiers using these representations. [14, 15] Obviously more attention is needed in building deep learning based ACE systems [16].

1.2. Evolution in ACE Evaluation

Besides the evolution of system architectures, ACE evaluation methods are also evolving, and it is most reflected in the changing of chord vocabularies in MIREX [17], a yearly Music Information Retrieval (MIR) Evaluation eXchange event. A more complex chord vocabulary leads to a more difficult task. From 2008 to 2012, the most popular evaluation method in ACE is the "majmin24" evaluation, namely, only major and minor triads are evaluated [18]. This leads to a much easier task, in which sevenths, ninths, etc., are all mapped to maj/min, even actually more complicated chord voicings sound very different from their triads. Moreover, chord inversions are not considered, which may have resulted in many ACE systems avoid generating chord inversions [19, 20], despite the fact that chord inversions sound very different from their root positions.

Thanks to a recent work on re-examining ACE evaluations [20], the ACE community is now taking a new evaluation approach. It contains four main categories (namely, "MajMin", "MajMinBass", "Sevenths" and "SeventhsBass") that try to incorporate inversions. Nevertheless, due to the dominant amount of root position chords, systems that only accept root positions can still achieve a high score in all these categories [19] because of their good behavior on root positions. Thus this is still disadvantageous towards systems that accept all chord types in "SeventhsBass" category because they more or less compromise performances on root position chords due to their wide vocabulary acceptability. It is considered a big challenge to both accept large chord vocabulary and achieve high evaluation score under the same vocabulary, such as "SeventhsBass", at the same time. A better comparison between algorithms' performances should also take into account the vocabulary they accept.

1.3. The Proposed ACE System

The proposed ACE system aims at tackling the challenge of "SeventhsBass" evaluation on "SeventhsBass" vocabulary. That is, the system's chord vocabulary is identical to the "SeventhsBass", and it focuses on "SeventhsBass" evaluation

metric. In order to meet this goal, the system uses a new ACE architecture. It starts from a frame-wise "PCP + GMM + HMM" chord segmentation layer [7]. Then on top of that it adds a segment-wise "deep neural network (DNN) + HMM inference" chord estimation layer. The first layer performs a screen of chord estimation for every frame and segment the chords. The second layer, when applied, regards the first layer as a chord segmentation pass and performs another screen of chord estimation for every segment.

The rest of the paper is organized as follows: section 2 and 3 describe engineering details of the two-layer ACE system; section 4 gives experiment results under the standard MIREX evaluation procedure and discusses the results; finally, section 5 sums up the contribution in this paper and deduces some possible directions that may lead to improvement.

2. FIRST LAYER

The first layer is a complete ACE system by itself. The front-end performs various signal processing tasks to compute reliable bass-treble chromagram. The back-end smooths and decodes the chromagram using a probabilistic model. First the input is resampled at 11025 Hz, and a spectrogram of the input is computed using short-time-Fourier-transform (STFT) with 4096-point Hamming window and 512-point hop size. Then each spectrum is up-sampled 80 times and then mapped to a 252-bin log-frequency spectrum with 1/3-semitone per bin step. Tuning is performed as indicated in [9], where the detuning (in semitone) is estimated as:

$$\delta = \frac{wrap(-\varphi - 2\pi/3)}{2\pi} \tag{1}$$

where $wrap$ is a function wrapping its input to $[-\pi, \pi)$, and φ is the phase angle at $2\pi/3$ of the discrete-Fourier-transform of the time averaged log-frequency spectrogram. After tuning, a standardization process is performed to attenuate background noise and enhance harmonic content. It subtracts from the input matrix the running mean of every column and divides the result by the running standard deviation of the same input matrix. The output matrix from the above process is fed to a non-negative-least-square (NNLS) algorithm [7] (Equation 2), which finds for every input spectrum (Y) an optimal non-negative fit (X) of a linear combination of a set of semitone pitch profiles (P). The output is a 84-bin spectrogram with a semitone per bin step.

$$min_{X>0}||P \cdot X - Y||_2^2 \tag{2}$$

2.1. Chromagram Computation

For better chord smoothing results, there is no pre-segment in this step. Specifically, there is no beat-level averaging. To take into account the possibility that bass signal will appear in high pitch range, the system applies a bass profile in the shape of a Rayleigh distribution (Fig.1). Each chroma are computed by weighting the input NNLS spectrum (84-bin) with both profiles, and the saliences belonging to the same pitch class are merged. The normalized result is a 24-dimension bass-treble chromagram.

	μ	σ^2
Bass - Chord Bass	1	0.1
Bass - Not Chord Bass	1	0.5
Treble - Chord Note	1	0.2
No Chord	1	0.2

Table 1. HMM-1 Parameters

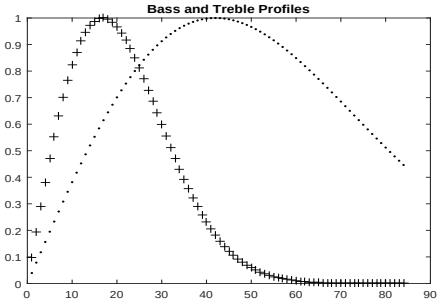

Fig. 1. Bass(+) and Treble(.) Profiles

2.2. Chord Smoothing Model

The bass-treble chromagram are smoothed and decoded by a hidden-Markov-model (the first HMM in the system. Let's call it HMM-1). The number of its hidden states equals to the number of chords. Each hidden node generates a 24-dimension observable node. Each observable node generates a 24-dimension bass-treble chroma. Its language model is unbiased towards any type of chord transition except for having a very high bias on self transitions. Its acoustic model is a 24-dimension multivariate Gaussian model with parameters specified in Table 1. They are tuned for better chord inversion recognition. Since the results in this stage are subject to be corrected in the next stage, thus the model is called "chord smoothing model", because the major contribution of this stage as seen by the next stage is to divide the input into harmonic segments or chord segments. The system's chord vocabulary is identical to "SeventhsBass", including maj, min, maj7, 7, min7, maj/3, maj/5, min/b3, min/5, maj7/3, maj7/5, maj7/7, 7/3, 7/5, 7/b7, min7/b3, min7/5 min7/b7 and the N chord, perfectly conforming with the current MIREX evaluation standard. [20]

3. SECOND LAYER

The second layer is a single chord classifier by itself. When applied on top of the first layer, it re-examines the labels of

every segment and re-labeled them. Two schemes based on DNN have been implemented for comparison. One is without DBN pre-train and the other is with DBN pre-train . Both networks are trained using "JayChou29", a Chinese pop song ground-truth dataset manually labeled by the author. The chord type composition of this dataset is well balanced. It contains 45.6% maj/min chords, 30.5% seventh chords (maj7, min7 and 7), 20.3% inversion chords, and 3.6% no-chord.

3.1. DNN Based Estimation

In this scheme, a 4-layer feed-forward neural network is trained to fit the "JayChou29" dataset. The 24-dimension bass-treble chroma is chosen as input feature. We choose input feature at this level indicates that we tend to believe the prior knowledge we have in traditional ACE practice. Given the limited amount of training data, especially for minority chords, having prior knowledge means the data can be released for more focused use on training higher level representations. The two hidden layers are with size 120 and 544 respectively. The output layer is 277-way softmax, corresponding to the posterior probabilities of chords in "SeventhsBass". All input-output pairs in training data are transposed with 12 different roots to enlarge the number of training cases 12 times. The network is trained with 1000 iterations of full-batch gradient descent, with weight decay factor $\lambda = 1$. The trained neural network is taken as an acoustic model, which outputs probabilities of the 277 chords given a bass-treble chroma. We call a sequence of such chord probabilities a "chordogram", and a single such 277-dimension vector a "chordo". The acoustic model is coupled with a hidden-Markov-model (HMM-2. Fig. 2) with 277 hidden states. In HMM-2, each hidden node generates a 277-dimension observable node. The emission probabilities are modeled as a 277-dimension Gaussian with $\mu = 1$ and $\sigma^2 = 0.1$. Both the transition matrix and the prior probabilities are set to uniform distribution. The new labels generated by HMM-2 will be taken as the chord progression output.

3.2. DBN-DNN Based Estimation

In this scheme, the feed-forward neural network is turned into a deep belief network [21], where the lower three layers are turned into two stacked restricted Boltzmann machines (RBMs), and their connections to the output layer remains the same (Fig. 2). To train this network, first the two stacked RBMs are pre-trained using contrastive divergence (CD) algorithm [21]. After the lower RBM is trained, its hidden units are regarded as visible units to train the second RBM. Each RBM is pre-trained with 10000 iterations of CD-1 with stochastic mini-batch gradient descent of size 1000. The unlabeled data is taken from "JayChou29". After pre-training, the whole network is regarded as a feed-forward DNN, whose initial weights are set by the pre-training process, and it is trained using the same procedure as in the previous section.

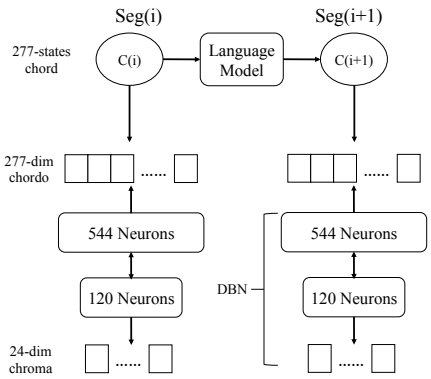

Fig. 2. DBN-DNN based estimation. Note that the decoding is segment-wise.

	Mm	MmB	7	7B	S	NZ
Chordino	74.3	71.4	53.0	50.6	301.2	9
HMM-1	74.2	62.6	66.0	55.5	278.7	13
DNN-HMM-2	71.4	66.5	62.0	57.8	297.1	13
DBN-HMM-2	70.5	65.6	61.3	57.0	288.4	13
NIV	74.3	72.3	66.2	64.4	183.7	8

Table 3. Overall accuracy results on 5 different approaches. M=Maj, m=min, 7=Sevenths, B=Bass, S=sum of individual chord performances, NZ=# of non-zero fields

4. RESULTS AND DISCUSSIONS

Among the many ACE systems submitted to MIREX for open-to-public evaluation between 2013-2015 (that is, after the new evaluation standard [20]), only the "Chordino" system [22] supports chord inversions [19]. It has very good performance despite accepting a large chord vocabulary. Although it does not fully support "SeventhsBass", it is still a high standard baseline approach with a similar vocabulary constraints as the proposed system's. (After submission of this paper, we submit our system and the JayChou29 dataset to MIREX ACE 2015. Now all the results are publicly available in MIREX ACE 2015's website[1]. Note that the good results on Billboard datasets, which we don't have access to, further demonstrates the generality of the proposed approaches)

There are several approaches to compare: Chordino; the proposed system with HMM-1 only; with DNN-HMM-2; and with DBN-HMM-2. And there's a fifth approach based on HMM-1 but not accepting chord inversions. The experiment results on "The Beatles" 180 songs dataset are shown in table 3. Let's focus on the first 4 approaches. In terms of "MajMinBass" and "Sevenths", the results show Chordino wins the former but loses the latter. But when referring to individual chord-type result in Table 2, what actually happens is Chordino performs better at all seventh chords but worse at

[1] www.music-ir.org/mirex/wiki/2015:Audio_Chord_Estimation_Results

%	2.01	0.95	63.31	0.02	0.17	0.27	0.82	0.08	0.06	0.39	8.33	0.61	0.44	14.99	0.01	0.06	0.41	2.37
	M/5	M/3	M	M7/5	M7/3	M7/7	M7	7/5	7/3	7/b7	7	m/5	m/b3	m	m7/5	m7/b3	m7/b7	m7
CH	19.9	17.1	54.4	0.0	0.0	0.0	55.6	0.0	0.0	5.7	41.0	0.0	0.0	54.3	0.0	0.0	0.0	51.0
HM	37.1	17.2	67.3	0.0	0.0	13.6	22.1	0.0	0.0	8.8	3.6	23.1	15.3	56.8	0.0	0.0	0.8	10.7
DN	24.0	25.1	67.9	0.0	0.0	0.0	39.4	0.0	10.1	13.2	4.1	4.4	9.8	58.9	0.0	0.0	0.7	36.2
DB	23.1	26.0	66.7	0.0	0.0	0.0	36.6	0.0	1.9	18.5	4.5	4.4	11.4	58.6	0.0	0.0	0.5	32.5
NI	0.0	0.0	79.3	0.0	0.0	0.0	15.4	0.0	0.0	0.0	2.6	0.0	0.0	73.9	0.0	0.0	0.0	10.2

Table 2. Detail accuracy results on 5 different approaches. The system order is the same as in Table 3. (M=maj, m=min)

all inversions. This can be explained by examining the chord mapping scheme involved in both metrics. In "MajMinBass", chords are mapped to their triads, while in "Sevenths", chord inversions are mapped to their root positions. Thus the seemingly contradictory phenomenon implies that Chordino tends to mis-classify chords to their sevenths or vice versa, but our system tends to mis-classify chords to their inversions or vice versa.

If we focus on "SeventhsBass" metric, which is a weighted average of all the individual chord-type performances in Table 2 , the best approach is DNN-HMM-2. The fifth approach is flawed because it only considers root positions, which is not at a fair bottom line as the other four, but we are not able to deduce this conclusion by looking at the metric score alone. The reason why the fifth approach stands out is because the test dataset contains far more root position chords than inversions. In this case, we need to count on other metrics. One proper metric is the number of non-zero scores. If we compute this on Table 2, the Chordino gets 9, NIV is 8, and others are 13. It shows the vocabulary capability of our system. Another proper metric is simply the sum of individual scores (regardless of the weight). The sum of Chordino is 301.2, the three proposed approaches are 278.7, 297.1 and 288.4 respectively, but NIV has only 183.7. This on one hand strongly indicates the NIV approach is flawed, and on the other hand shows Chordino is still extremely competitive, at least on the chosen dataset.

Notice that the DBN approach performs similar to the DNN approach, which means the initial weights set by pre-training two stacked RBMs doesn't help much in our system. Actually during different trials in training, there's always an over 10% variance between training set and validation set accuracy, no matter how the hyperparameters are chosen (even if using a much lower level 252-bin pitch salience profile as input feature instead). The performance boost in "SeventhsBass" is due to the well balance of the training set, but it still does not reach a welcoming range (such as above 70%). This has at least two reasons. The obvious reason is that the amount of training data is far from enough. Specifically, the amount of minority chord types data is far from enough. This can only be solved by a much larger training dataset, and thus asks for the ACE community to collect more ground-truth data of minority chord types. The second reason is not so obvious. It is a problem of the proposed second layer itself, that a chord is actually not recognized by first averaging the sound in each harmonic segment, but by decoding the segment one slice at a time. For example in an alternating C-G bass scenario of a C:maj, it is recognized as C:maj because one hears the C bass before the G bass. But if averaging the segment, one will confuse whether it should be a C:maj or C:maj/5. There is even more confusion in a walking bass scenario, in which a chord could be recognized as any of its inversion form. Unfortunately in "The Beatles" data set, the above mentioned two scenarios are quite common, so that our system tends to mis-classify a lot of root position chords to their inversions.

5. CONCLUSION

Among the 3 approaches spawned from the proposed system, the DNN-HMM-2 performs the best. Compared with Chordino, our system wins in terms of the most complex metric "SeventhsBass" in a significant way, given a bottom line that both systems accept chord inversions. The performance gain is mainly because the training set is well balanced with a proper amount of various chord types, but the system's architecture itself suffers from a key defect that it is unable to differentiate the time order of a bass line within a harmonic/chord segment. A better DNN-DBN based architecture should be able to accept intra-segment context information into its input layer (note that both [13] and [12] have similar ideas but they lack large vocabulary evaluations). Another way to improve the system is to incorporated a non-uniform language model in HMM-2. But a good language model demands much more labeled data on minority chord transitions. The lack of minority chord data also partly lead to our choice of a 24-dimension input feature instead of a much lower level of raw feature. Thus the learning is not quite "deep" though, since we tend to believe the traditional ACE practice has somewhat figured out a good set of connections and representations in the front-end. Nevertheless, hard-wiring these prior knowledge into the system can ease the limited amount of data for more focused use on a higher level network.

In future deep learning based ACE, we need more sophisticated architectures to incorporate subtle time dependencies; meanwhile we also need a dataset with more abundant minority chords both to do better training on machine learning models and to setup fairer evaluation tasks.

6. REFERENCES

[1] Takuya Fujishima, "Realtime chord recognition of musical sound: A system using common lisp music," in *Proc. ICMC*, 1999, vol. 1999, pp. 464–467.

[2] Juan Pablo Bello and Jeremy Pickens, "A robust mid-level representation for harmonic content in music signals.," in *ISMIR*, 2005, vol. 5, pp. 304–311.

[3] Christopher Harte and Mark Sandler, "Automatic chord identifcation using a quantised chromagram," in *Audio Engineering Society Convention 118*. Audio Engineering Society, 2005.

[4] Alexander Sheh and Daniel PW Ellis, "Chord segmentation and recognition using em-trained hidden markov models," in *Proc. ISMIR*. 2003, vol. 2003, pp. 185–191, International Symposium on Music Information Retrieval.

[5] Maksim Khadkevich and Maurizio Omologo, "Time-frequency reassigned features for automatic chord recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 181–184.

[6] Johan Pauwels and Jean-Pierre Martens, "Integrating musicological knowledge into a probabilistic framework for chord and key extraction," in *Audio Engineering Society Convention 128*. Audio Engineering Society, 2010.

[7] Matthias Mauch and Simon Dixon, "Approximate note transcription for the improved identification of difficult chords.," in *ISMIR*, 2010, pp. 135–140.

[8] Kouhei Sumi, Katsutoshi Itoyama, Kazuyoshi Yoshii, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G Okuno, "Automatic chord recognition based on probabilistic integration of chord transition and bass pitch estimation.," in *ISMIR*, 2008, pp. 39–44.

[9] Matthias Mauch, *Automatic chord transcription from audio using computational models of musical context*, Ph.D. thesis, School of Electronic Engineering and Computer Science Queen Mary, University of London, 2010.

[10] Yizhao Ni, Matt McVicar, Raul Santos-Rodriguez, and Tijl De Bie, "An end-to-end machine learning system for harmonic analysis of music," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 6, pp. 1771–1783, 2012.

[11] Li Deng and Dong Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[12] Eric J Humphrey and Juan P Bello, "Rethinking automatic chord recognition with convolutional neural networks," in *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*. IEEE, 2012, vol. 2, pp. 357–362.

[13] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent, "Audio chord recognition with recurrent neural networks.," in *ISMIR*, 2013, pp. 335–340.

[14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] Yoshua Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[16] Eric J Humphrey, Juan P Bello, and Yann LeCun, "Feature learning and deep architectures: new directions for music informatics," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 461–481, 2013.

[17] J Stephen Downie, "The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research," *Acoustical Science and Technology*, vol. 29, no. 4, pp. 247–255, 2008.

[18] Christopher Harte, *Towards automatic extraction of harmony information from music signals*, Ph.D. thesis, Department of Electronic Engineering, Queen Mary, University of London, 2010.

[19] J Ashley Burgoyne, W Bas de Haas, and Johan Pauwels, "On comparative statistics for labelling tasks: What can we learn from mirex ace 2013," in *Proceedings of the 15th Conference of the International Society for Music Information Retrieval (ISMIR 2014)*, 2014, pp. 525–530.

[20] Johan Pauwels and Geoffroy Peeters, "Evaluating automatically estimated chord sequences," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 749–753.

[21] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[22] Chris Cannam, Matthias Mauch, Matthew EP Davies, Simon Dixon, Christian Landone, Katy Noland, Mark Levy, Massimiliano Zanoni, Dan Stowell, and Luıs A Figueira, "Mirex 2013 entry: Vamp plugins from the centre for digital music," 2013.