MUSIC CHORD RECOGNITION BASED ON MIDI-TRAINED DEEP FEATURE AND BLSTM-CRF HYBIRD DECODING

Yiming Wu, Wei Li

School of Computer Science and Technology, Fudan University, Shanghai, China
yimingwu15@fudan.edu.cn, weili-fudan@fudan.edu.cn

ABSTRACT

In this paper, we design a novel deep learning based hybrid system for automatic chord recognition. Currently, there is a bottleneck in the amount of enough annotated data for training robust acoustic models, as hand annotating time-synchronized chord labels requires professional musical skills and considerable labor. As a solution to this problem, we construct a large set of time synchronized MIDI-audio pairs, and use these data to train a Deep Residual Network (DRN) feature extractor, which can then estimate pitch class activations of real-world music audio recordings. Sequence classification and decoding are then performed with a trained Bidirectional LSTM and Conditional Random Fields (CRF) network. Experiments show that the proposed model is compatible for both regular major/minor triad chord classification and larger vocabulary chord recognition, the performance is good and no less than other state-of-the-art systems. The proposed system also achieved good evaluation score in MIREX 2017 Automatic Chord Estimation task.

Index Terms— Automatic chord recognition, Deep residual network, Bidirectional long short term memory (BLSTM), Conditional random fields (CRF).

1. INTRODUCTION

Audio chord recognition is desired to automatically transcribe the time-synchronized chord label sequence that best describes the harmonic progression of a piece of musical audio data. It is a key factor of music content analysis and has been a long-lasting research theme in the Music Information Retrieval (MIR) community [10].

In the recent decade, with the rapid development of computing resources and training technique, deep learning has made a great success in various research fields like speech recognition, computer vision, and some MIR tasks. Regarding chord recognition, there has been a trend to move from traditional Chroma feature [17, 18] extraction plus shallow-level machine learning models [6, 9, 19] towards deep learning approaches.

Chronologically, the first deep learning-based chord recognition system was proposed in [5], which tries to automatically extract harmonic feature with a trained Convolutional Neural Network (CNN). Following this, several works investigate automatic representation learning via deep learning methods, such as Deep Belief Network (DBN) [4] and Deep Neural Network (DNN) [3]. The latest work [2] designs a deep CNN for feature extraction and frame-wise classification, and uses CRF for post-filtering, which reaches the state-of-the-art chord recognition accuracy on the common datasets.

Rather than learning the mappings between input spectrum and corresponding labels directly, some works have been seeking to obtain more explicit representations with neural networks [1, 16]. The idea of learning Chroma feature extraction with data-driven method was proposed in [1] and named as Deep Chroma Extractor. The neural network-based Chroma extractor learns the mappings between the input spectrum and ideal Chroma vector templates (derived from corresponding chord), with real-world audio recordings and hand-labelled chord information.

A critical issue for all deep learning based methods is to collect and label enough high quality training data. Specific to the chord recognition field, annotating time-synchronized ground-truth chord labels is not only time consuming and tedious, but also needs professional music skills. This bottleneck has obstructed the further development of deep learning based chord recognition methods with big data as a prerequisite.

In this paper, we turn to MIDI (Musical Instrument Digital Interface) formatted music data to overcome the shortcomings. First, we can easily collect much more training data because no additional annotation is needed. Second, the synthesized audio is strictly synchronized to MIDI note information, or in other words, its note-level annotation quality is guaranteed to be perfect.

Based on this idea, we propose a novel chord recognition system with hybrid deep learning architecture. For feature extraction, we use a Deep Residual Network (DRN) to automatically learn and extract deep features characterizing the music content. Unlike other related works which are trained with limited error-prone labels manually marked from real-world music signals, here the DRN is trained with a large set of MIDI files and their synthesized audio signals. For sequence decoding, we employ a combination of Bidirectional Long Short Term Memory (BLSTM) network [7] and Conditional Random Fields (CRF) [13], which is adept in modeling time sequence.

2. SYSTEM OVERVIEW

The proposed chord recognition system includes three subsections, namely feature extractor, pattern matching and optimal label decoding. The acoustic features are first calculated by the DRN from the spectrogram of each music signal. Then the feature vectors are fed into the BLSTM network as a sequence, and a class likelihood vector is

calculated for each frame. Finally, the class likelihood sequence is input to the trained CRF to decode the optimal chord label sequence.

3. FEATURE EXTRACTOR TRAINING

This section describes the architecture of the proposed deep feature extractor and its training procedure.

3.1. Input Preprocessing

Each audio signal (synthesized from MIDI file) is first downsampled to 22,050 Hz and transformed into log-frequency spectrogram representation via Constant-Q Transform [8], which is computed over 6 octaves with 24 bins per octave and 2048 samples of hop size. The magnitude spectrum is transformed into $S_{\log} = \ln(S + \varepsilon)$, where S represents the raw spectrogram and ε is a small number for avoiding zero value in log calculation.

After that, we apply global mean-variance normalization on $S_{\log}$ of a single track, to reduce the variance of overall spectral energy between different music pieces.

Finally, the pre-processed CQT spectrogram is sent to the DRN feature extractor model as input vectors.

3.2. Target Representation

We train the neural network so that it can transform the above spectrogram S_{norm} into an ideal Chroma representation. Concretely, it tries to predict which of the 12 pitch classes are activated at a specific frame, just like what original Chroma vector extractor does. We transform note information of each MIDI file into a Chroma-like 12-dimension binary vector sequence that tells the pitch class activations of corresponding audio frames of the spectrogram. That is, if any MIDI note is active at a specific frame, the value of corresponding pitch class of the target vector of the frame is set to be 1.

To obtain more information, we further add two feature vectors into the Chroma representation, i.e., bass note Chroma and top note Chroma. Each of them is a 12-dimension one-hot (i.e., only one dimension of the vector is 1 and all others are 0) vector that tells the pitch classes of the base note (the lowest active MIDI note) and the top active note (the highest active MIDI note). The lowest and highest notes are excluded in original pitch class activation calculation, so that the compressed vector represents the "middle notes" of the corresponding frame, which are often chord tones. The network is expected to predict the bass note, top note and other pitch class activations of the current frame simultaneously, as shown in Fig. 1.

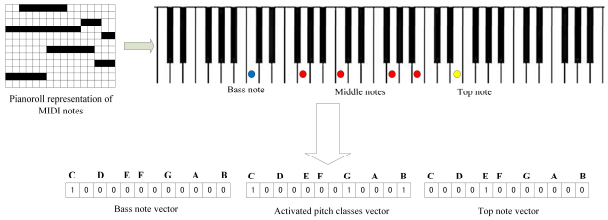

Fig. 1. MIDI note activation of a time frame is represented in three 12-dimension vectors, indicating current bass note, active pitch classes and top note.

In this way, we get a 36-dimension deep acoustic feature for further classification. Compared with the Deep Chroma feature which construct target representation with chord annotations, the proposed feature representation is able to reflect more concrete information about active notes, and the dataset is far less error-prone and much easier to collect as well.

3.3. Deep Residual Network

Deep Residual Network is a feed-forward neural network with shortcut connections [14]. Its basic theory makes the network optimization easier, thus it is possible to construct much deeper neural networks with better performance.

In our work, DRN is used for harmonic feature extraction. The network is constructed by stacking 5 layers, each layer has 512 units with tanh activation function. The output layer, activated with a sigmoid function, is intended to tell if each pitch class is activated (1.0) or not (0.0).

3.4. Network Training

The neural network is trained to minimize the mean-squared error between the network output and the target vectors. Fig. 2 describes the overall structure of feature extractor training. The parameters are optimized using Stochastic Gradient Descent algorithm with learning rate of 0.01.

For network training, we collected 210 MIDI files from RWC Classical, Jazz and Genres dataset [11], plus 900 MIDI files randomly selected from Lakh MIDI dataset [21]. That is 1110 pieces of General MIDI format multitrack music data in total. We synthesized corresponding audio using *Direct MIDI to MP3 Converter* by *Piston Software*, with Chorium soundfont used as the sound source of the General MIDI instruments.

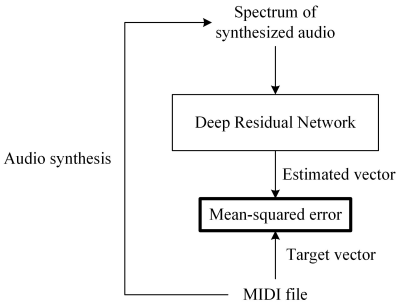

Fig. 2. The training of DRN feature extractor with synthesized MIDI data.

4. BLSTM-CRF SEQUENCE DECODING

This section describes the BLSTM-CRF model for pattern matching and decoding chord sequence, given the feature sequence calculated by the DRN. BLSTM network performs pattern matching, and CRF infers the final label sequence.

4.1. BLSTM Network

We construct a Bi-directional LSTM network with a pair of forward and backward recurrent layers, 512 LSTM units on each layer. Instead of using RNN as "language model" as in previous works [3, 4], the Bi-directional LSTM network acts as a sequence classifier in our proposed model. It receives a

feature vector sequence calculated by the DRN, and outputs another 25-dimension vector sequence that represents the chord class likelihoods on each frame.

To reduce overfitting in the training phase, we apply dropout operation [20] with probability 0.5 to the output of both LSTM layers.

4.2. Conditional Random Fields

Our model employs a linear-chain CRF, which has been widely used in various sequence labelling tasks. Its energy function is defined as,

$$\mathrm{E}(\mathrm{X},\mathrm{Y}) = \prod_i (x_{iy_i} + c_{y_{i-1}y_i}) \quad (1)$$

where for frame *i*, x_{iy_i} is the class likelihood of y_i (calculated by the BLSTM network) and $c_{y_{i-1}y_i}$ is the label transition cost between label y_{i-1} and y_i.

The CRF is trained by optimizing the label transition cost matrix *c*. Given an input sequence, the objective is to minimize the negative log-likelihood of the expected label sequence in light of Eq. (2),

$$\mathrm{L} = -\left(\sum_{\mathrm{i}} x_{iy_i} + \sum_{\mathrm{i}} c_{y_{i-1}y_i} - \ln(\mathrm{Z})\right) \quad (2)$$

where *Z* is the normalizing constant.

At decoding phrase the model finds out the label sequence Y that maximizes the conditional probability P(Y|X) via Viterbi algorithm.

4.3. Network Training

In decoder training phase, the training dataset is composed of pairs of feature sequence (obtained from above feature extraction stage) and time-synchronized chord annotation data. As the decoder does not need to learn the dependency across the whole music, we choose to randomly take a sequence of a fixed length (128 frames, or about 10 seconds) each time to train the BLSTM-CRF model.

The classifier (BLSTM) and the decoder (CRF) component is trained individually. First, the BLSTM network is trained with the output layer activated with softmax function, to classify the feature sequence by itself. After this training is finished, we fix the well-trained parameters of BLSTM and train the parameters of CRF with the same dataset. Parameters of both models are optimized using AdaDelta algorithm.

5. LARGE VOCABULARY CHORD RECOGNITION

In most chord recognition methods, including the proposed neural network architecture, the recognition process is seen as a quantization process that assigns all observations to corresponding one-of-K representations, built on the assumption that the 24 classes (major and minor triads) are mutually independent. However, if we add more complex chords (like seventh and inverted chords) into the vocabulary, this assumption no longer holds, because there exist chords related to each other hierarchically [15]. Some researches on large vocabulary chord recognition try to extract more detailed acoustic feature while follow the conventional flat classification framework. [9, 22].

In practice, we generally estimate each chord in triad level first, and then determine whether it is seventh or more complex chord, based on the prior estimation. To mimic this process, we design a two-stage complex chord recognition method.

Concretely, the proposed system does this by modifying qualities (major or minor triad) and inversion types of each recognized chord signature. Given a chord signature (major or minor triad, which is the estimation result of the BLSTM-CRF network) and the feature sequence of corresponding time frame, we calculate the mathematical mean of the feature value along the dimension of its third, fifth, seventh and major-seventh note, and bass feature value of its root, third and fifth note. Then we determine its true quality and inversions with an explicit thresholding strategy as follows,

1. If the mean value of the seventh or major-seventh note in middle feature is over 0.5, then change the chord quality to seventh (minor seventh) or major-seventh (depending on which value is bigger).
2. If the mean value of the third or fifth note in bass feature is over 0.5 and bigger than that of the root note, then mark the chord as first or second inversion (depending on which value is bigger).

In this way, the chord recognition system is able to support 61 types of chords if considering only the seventh chords, and 181 types of chords if taking chord inversions into account. At the same time, the recognition accuracy of triads is not affected.

6. EXPERIMENTAL RESULTS

In this section, we first describe the training and testing datasets and the evaluation metrics, then perform a series of quantitative experiments to comprehensively investigate the chord recognition accuracy under various conditions.

6.1. Datasets and Evaluation Metrics

We evaluate the proposed system on a compound dataset comprising the following two subsets. **Isophonics:** 180 songs by The Beatles, 19 songs by Queen, and 18 songs by Carole King. **RWC pop**: 100 Japanese and American style pop songs. We perform an 8-fold cross-validation on the dataset (in each fold the training data is used for training the BLSTM-CRF part). Note that this dataset does not contain any song used in the training MIDI dataset.

The commonly used evaluation measure in chord recognition is the Weighed Average Overlap Ratio (WAOR), which is computed in terms of Eq. (3), using the mir_eval library [12]:

$$\mathrm{WAOR} = \frac{\mathrm{t_c}}{\mathrm{t_a}} \quad (3)$$

where t_c is the total overlap time between the annotated and estimated chord label sequences, and t_a is the total duration of annotated chord sequence.

To evaluate the recognition performance in different chord vocabularies, the comparison is performed in the three metrics implemented in the library: **Majmin**, considers only major and minor triads, which is the most conventional comparison metric. **Sevenths**, considers seventh, minor-

seventh, and major-seventh qualities in addition. **Sevenths-inv**, considers triads, sevenths and chord inversions. The three metrics correspond to "majmin", "sevenths", "sevenths-inv" metric in MIREX competition.

6.2. Results and discussions

Below we compare our method with other state-of-the-art baseline systems in terms of major/minor chords and complex chords under the two public datasets. Furthermore, we investigate the influence of deep feature extraction and MIDI training dataset size on the system performance.

6.2.1. Evaluation of Major and Minor Triads Recognition

To evaluate the system performance of recognizing triads, we do the same 8-fold evaluation on different chord recognition systems, i.e., Chordino algorithm [6], the CNN-CRF system [2], the CJK-BLSTM system [22], and the proposed system (denoted as MF-BLSTM-CRF).

Besides the proposed MF-BLSTM-CRF system, we also evaluate its two alternatives to observe the effect of deep feature extraction. One is denoted as BLSTM-CRF, where the DRN is removed and the BLSTM-CRF decoder is trained to directly classify raw CQT spectrum sequences. The other is denoted as DC-BLSTM-CRF, where the DRN extractor is replaced with another neural network of the same architecture, but trained in the way as Deep Chroma Extractor (with raw audio-chord annotation pairs) [1].

Table 1. WAOR under majmin metric

	Iso	RWC
Chordino	76.4%	74.8%
CNN-CRF	83.2%	79.9%
CJK-BLSTM	72.6%	X
BLSTM-CRF	83.1%	79.2%
DC-BLSTM-CRF	83.0%	79.1%
MF-BLSTM-CRF	**84.1%**	**80.8%**

The last three rows of Table 1 indicate that MF-BLSTM-CRF performed better than its two variants BLSTM-CRF and DC-BLSTM-CRF on both the Isophonics dataset and the RWC dataset, the improvements are all over 1%, which again verifies the effectiveness of the proposed 36-dimension deep feature since the decoding modules (BLSTM-CRF part) are the same.

Compared with the three baseline systems, MF-BLSTM-CRF shows slightly better performance than CNN-CRF system. By contrast, it significantly outperforms Chordino and CJKU-BLSTM system. Note that CJKU-BLSTM has no experimental results reported on the RWC dataset.

6.2.2 Evaluation of Complex Chords Recognition

We show the estimation accuracy of complex chords in Table 2. By comparing Table 1 and Table 2, an obvious downward tendency can be observed with the increasing of chords types. In Table 2, MF-BLSTM-CRF is apparently superior to other baseline systems in terms of Sevenths and Sevenths-inv metrics.

Table 2. WAOR under complex chord metrics

	Sevenths		Sevenths-inv	
	Iso	RWC	Iso	RWC
Chordino	53.8%	58.0%	51.0%	53.1%
CNN-CRF	69.7%	53.7%	66.4%	51.1%
CJKU-BLSTM	59.4%	X	57.4%	X
MF-BLSTM-CRF	**71.9%**	**66.5%**	**67.3%**	**63.7%**

The proposed system participated in the MIREX 2017 Audio Chord Estimation task (WL1) and achieved state-of-the-art score in terms of the Sevenths evaluation metric on two Billboard datasets.

6.2.3. Influence of the training dataset size

To examine the influence of training data amount on system performance, we resize the MIDI dataset to 50, 100, 300, 500, 1000, 1500, 2000, 2500 and 3000 tracks, and examine the relationship between the recognition accuracy and different amount of training data.

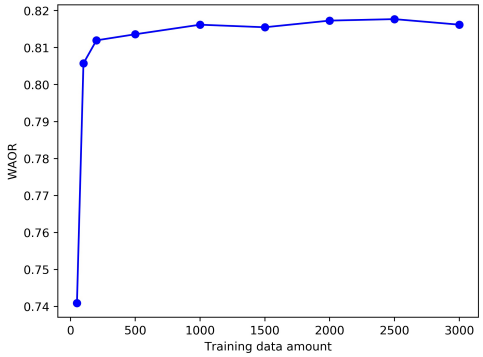

Fig. 3. Relationship between chord recognition accuracy (WAOR) and different size of MIDI dataset.

Fig. 3 shows the calculated relationship curve, where WAOR is tested on the whole dataset (Isophonics+RWC). It is apparently observed that more training data lead to more accurate recognition performance. Especially, when the dataset size is less than 500, the ascending velocity is very fast. When the dataset size goes up from 1000 to 2000 and 3000, the curve tends to be converged, in other words, the overall chord recognition performance is no longer significantly improved.

7. CONCLUSIONS

In this work, we propose a new feature learning procedure and sequence decoder model for automatic chord recognition task. By combining the deep harmonic feature extractor (DRN) and the BLSTM-CRF sequence decoder, the proposed system is fully capable of automatically recognizing chord progressions, whose performance could reach or even outperform the state-of-the-art chord recognition systems. Since feature extraction of the DRN is oriented to note-level pitch class activations, the extracted feature (or the feature learning procedure) can be introduced into other MIR tasks that use Chroma-like features.

REFERENCES

[1] F. Korzeniowski and G. Widmer, Feature Learning for Chord Recognition: The Deep Chroma Extractor, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 37-43, 2016.

[2] F. Korzeniowski and G. Widmer, A Fully Convolutional Deep Auditory Model for Musical Chord Recognition, *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 13-16, 2016.

[3] S. Sigtia, N. Boulanger-Lewandonski and S. Dixon, Audio Chord Recognition with a Hybrid Recurrent Neural Network, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 127-133, 2015.

[4] N. Boulanger-Lewadowski, Y. Bengio and P. Vincent, Audio Chord Recognition with Recurrent Neural Networks, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 335-340, 2013.

[5] E. Humphrey and J. Bello, Rethinking Automatic Chord Recognition with Convolutional Neural Networks, *International Conference on Machine Learning and Applications (ICMLA)*, pp. 357-362, 2012.

[6] M. Mauch and S. Dixon, Approximate Note Transcription for the Improved Identification of Difficult Chords, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 135-140, 2010.

[7] A. Graves, N. Jaitly and A. Moharmed, Hybrid Speech Recognition with Deep Bidirectional LSTM, *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 273-278, 2013.

[8] C. Schörkhuber and A. Klapuri, Constant-Q Transform Toolbox for Music Processing, *Sound and Music Computing Conference (SMC)*, 2010.

[9] T. Cho, Improved Techniques for Automatic Chord Recognition from Music Audio Signals. Ph.D. thesis, New York University, 2014.

[10] M. Mauch and S. Dixon, Simultaneous Estimation of Chords and Musical Context from Audio, *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, 18(6), pp. 1280-1289, 2009.

[11] M. Goto, H. Hashiguchi, T. Nishimura and R. Oka, RWC Music Database: Popular, Classical, and Jazz Music Databases, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 287-288, 2002.

[12] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, Mir_eval: A Transparent Implementation of Common MIR Metrics, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 367-372, 2014.

[13] J. D. Lafferty, A. McCallum and F. C. N. Pereira, Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, *International Conference on Machine Learning (ICML)*, pp. 282-289, 2001.

[14] K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.770-778, 2016.

[15] E. J. Humphrey and J. P. Bello, Four Timely Insights on Automatic Chord Estimation, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 673-679, 2015.

[16] E. J. Humphrey, T. Cho and J. P. Bello, Learning a Robust Tonnetz-Space Transform for Automatic Chord Recognition, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 453-456, 2012.

[17] E. Gómez, Tonal Description of Music Audio Signals, Ph.D. thesis, Department of Information and Communication Technologies, Universitat Pompeu Fabra, 2006.

[18] M. Khadkevich and M. Omologo, Time-frequency Reassigned Features for Automatic Chord Recognition, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 181-184, 2011.

[19] R. Chen, W. Shen, A. Srinivasamurthy and P. Chordia, Chord Recognition using Duration-Explicit Hidden Markov Model, *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 445-450, 2012.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.

[21] C. Raffel, Learning-based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching, Ph.D. Thesis, Columbia University, 2016.

[22] J. Deng and Y. Kwok, A Hybrid Gaussian-HMM-Deep Learning Approach for Automatic Chord Estimation with Very Large Vocabulary, *International Society for Music Information Retrieval Conference (ISMIR)*, pp.812-818, 2016.