MIREX 2017 SUBMISSION : AUTOMATIC AUDIO CHORD RECOGNITION WITH MIDI-TRAINED DEEP FEATURE AND BLSTM-CRF SEQUENCE DECODING MODEL

Yiming Wu, Xiangyi Feng, Wei Li
School of Computer Science and Technology,
Fudan University of Shanghai
{yimingwu15, xyfeng15, weili-fudan}@fudan.edu.cn

ABSTRACT

This abstract presents our submission for Automatic Chord Recognition task. The system is constructed with a DRN (Deep Residual Network) trained with a large set of time-synchronized MIDI-audio pairs to precisely estimate active pitch classes at each time frame of real-world music audio recordings. Then a trained BLSTM-CRF (Bidirectional Long Short Term Memory and Conditional Random Fields) architecture performs chord label estimation on the feature sequence. In addition, a simple thresholding decision process is applied at post-processing stage to recognize more complex chord types (seventh chords and inversions).

1. SYSTEM OVERVIEW

The overall network architecture and the calculation flow of our chord recognition system is illustrated in Fig.1, which includes three subsections, namely feature extractor, pattern matcher and optimal label decoder. The acoustic features are first calculated with 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 fed to the trained CRF layer to decode the optimal label sequence. The three parts of the neural network is trained successively. For neural network implementation, we used deep learning framework Chainer[1].

2. FEATURE EXTRACTOR TRAINING

This section describes the architecture of the deep feature extractor and its training procedure. Feature extractor trained with a large set of MIDI data is one of the key features of our system.

2.1. Input Preprocessing

At feature extractor training phase, each audio signal (synthesized from MIDI file) is first downsampled to 22,050Hz and transformed into log-frequency spectrogram representation via Constant-Q Transform [2], which is computed over 6 octaves with 24 bins per octave and 2048 samples of hop size. The magnitude spectrum is transformed to log scale such that

$$\mathrm{S}_{\log} = \ln(S + \varepsilon) \quad (1)$$

where S represents the raw spectrogram and ε is a small number for avoiding zero value in log calculation.

Then global mean-variance normalization is applied to reduce the variance of overall spectral energy between different music pieces.

$$\mathrm{S}_{\mathrm{norm}} = \frac{S_{log} - mean(S_{log})}{var(S_{log})} \quad (2)$$

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

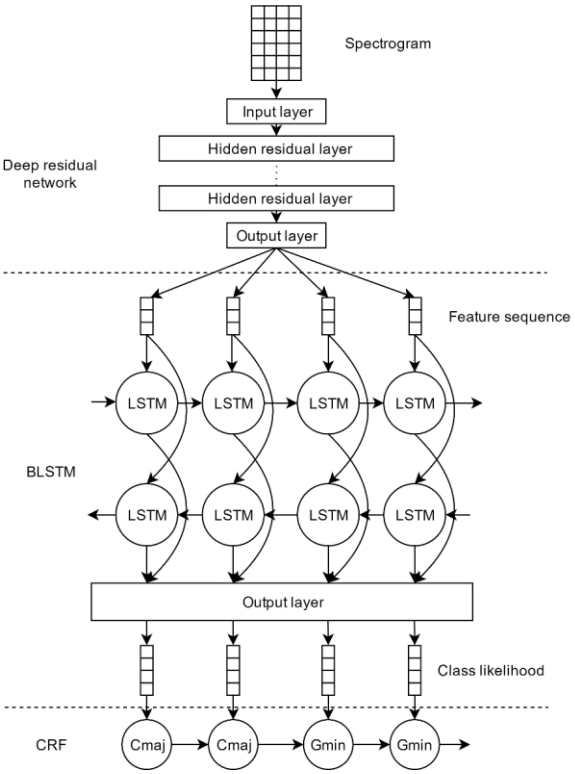

Figure 1. Overview of the presented chord recognition system. The system is composed of a DRN feature extractor trained with MIDI dataset, a BLSTM sequence classifier, and a CRF sequence decoder.

[1] https://chainer.org/

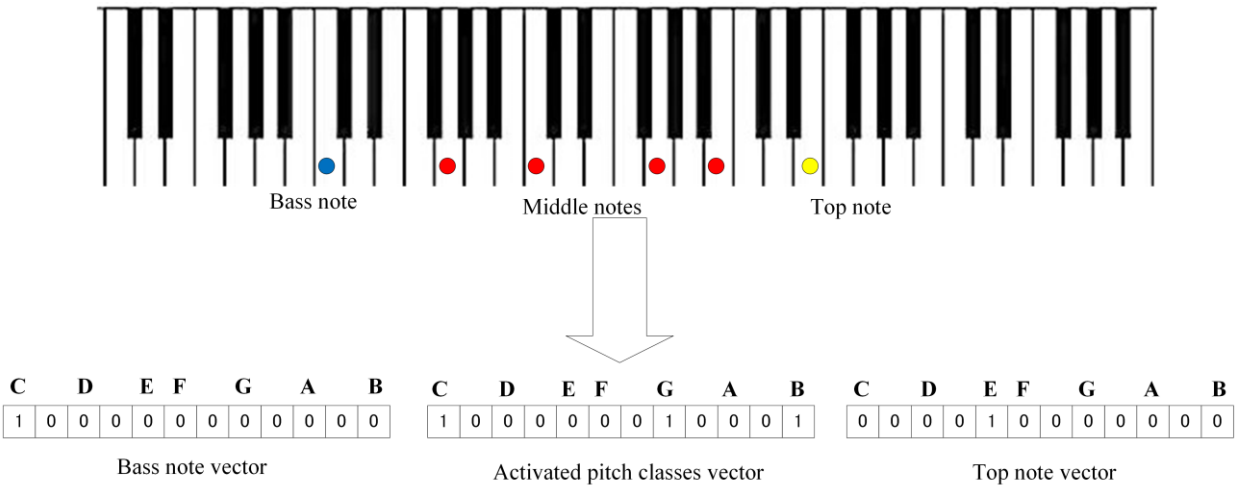

Figure 2. MIDI note activation of a time frame is represented in three 12-dimension vectors, indicating current bass note, active pitch classes and top note respectively.

2.2. Target Representation

We train the neural network so that it can transform the above spectrogram into an ideal harmonic representation. Concretely, it tries to predict which of the 12 pitch classes (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) are activated at a specific time step, just like what original Chroma vector extractor is expected. Instead of obtaining the target vectors from chord annotations(as Deep Chroma extractor did [1]), 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 time step, the value of corresponding pitch class of the target vector of the time frame is set to 1.

Additionally, the representation is expanded to include more aspects of the harmonic information. We further add two feature vectors into the Chroma representation: bass note vector and top note vector. Each of them is a 12-dimension one-hot vector that tells the pitch class of the bass note (the lowest active MIDI note) and the top active note (the highest active MIDI note) in each time frame. The lowest and highest notes are excluded in original pitch class activation calculation, so that the "activated pitch class" vector represents the "middle notes" of the corresponding frame, which are often chord tones. In this way, we get a 36-dimension deep acoustic feature for further classification. The target representation is illustrated in Fig.2. The network is expected to predict the top note as well as bass note and other pitch class activations of the current frame, simultaneously.

2.3. Deep Residual Network

We constructed a deep neural network for harmonic feature extraction. It is made up with stacked fully connected layers, where a shortcut connection is appended between the input and the output of each layer (which becomes a residual block of the Deep Residual Network).

In our system, the network is constructed by stacking 5 such layers. Each layer has 1024 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).

2.4. Network Training

The neural network is trained to minimize the mean-squared error between the network output and the target vectors. Fig.3

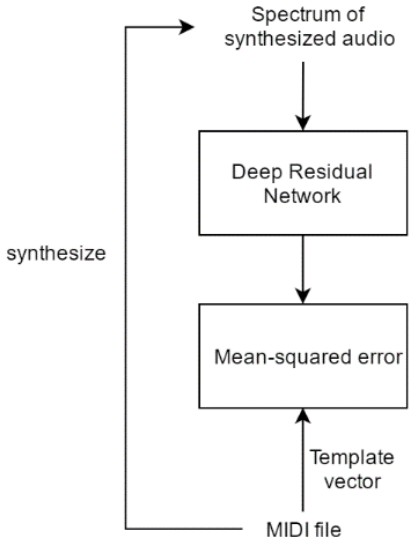

Figure 3. The calculation pipeline of DNN feature extractor training with synthesized MIDI data.

describes the overall structure of feature extractor training. The optimal parameters of the neural network can be estimated using backpropagation algorithm.

For network training, we collected 210 MIDI files from RWC Classical, Jazz and Genres dataset [3], plus 12000 MIDI files randomly selected from Lakh MIDI dataset [6]. We synthesized corresponding audio using *Direct MIDI to MP3 Converter* by *Piston Software*, with *Chorium* soundfont used as the sound source.

5. BLSTM-CRF SEQUENCE DECODING ARCHITECTURE

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. This part is trained after feature extractor training is finished.

25 chord classes are defined for chord classification, including 12 major triads, 12 minor triads and a "Non-chord" label that indicates silent, percussive or monotone areas of music audio.

5.1. BLSTM Network

We construct a Bi-directional LSTM network with a pair of forward and backward recurrent layers with 128 LSTM units on each layer, which 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 of each frame.

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

5.2. Conditional Random Fields

Given an input sequence X, CRF models the conditional probability of output label sequence Y in the following manner:

$$\mathrm{P(Y|X)} = \frac{\exp \mathrm{E(X,Y)}}{\sum_{\mathrm{Y}'} exp E(X,Y)} \qquad (3)$$

where E(X|Y) is the energy function and Y' represents any possible label sequences for sequence X.

In our system, we adopt a linear-chain CRF, which has been widely used in various sequence labelling tasks. In this case, the energy function E is defined as:

$$\mathrm{E(X,Y)} = \sum_{\mathrm{i}} (x_{iy_i} + c_{y_{i-1}y_i}) \qquad (4)$$

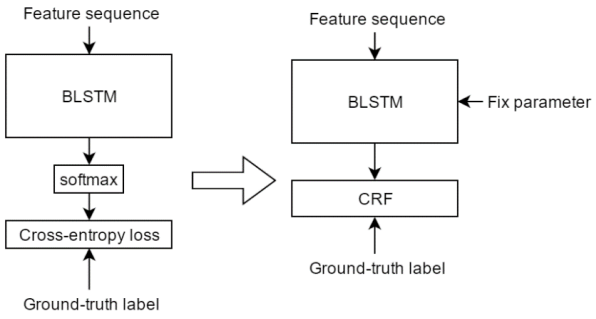

Figure 4. Overview of BLSTM-CRF decoder training. The BLSTM network is first trained as a classifier, then the CRF is trained with the same dataset.

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.

We train CRF by optimizing the label transition cost matrix c. Given an input sequence, the target is to minimize the negative log-likelihood of the expected label sequence:

$$\mathrm{L} = -\left(\sum_{\mathrm{i}} x_{iy_i} + \sum_{\mathrm{i}} c_{y_{i-1}y_i} - \ln(\mathrm{Z})\right) \qquad (5)$$

where Z is the normalizing constant. The parameter can be optimized with gradient descent algorithm.

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

5.3. Network Training

At 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. For the submitted system, the training set is constructed with RWC Popular Music dataset and USPOP Pop music dataset[2].

On each training epoch of BLSTM-CRF model, a fixed length (128 frames, about 10 seconds) are randomly taken from the dataset for loss calculation, for the decoder does not need to learn the dependency across the whole music.

As shown in Figure 4, 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, the well-trained parameters of BLSTM are fixed and the parameters of CRF are optimized with the same dataset.

[2] The annotations provided by Taemin Cho is available at https://github.com/tmc323/Chord-Annotations

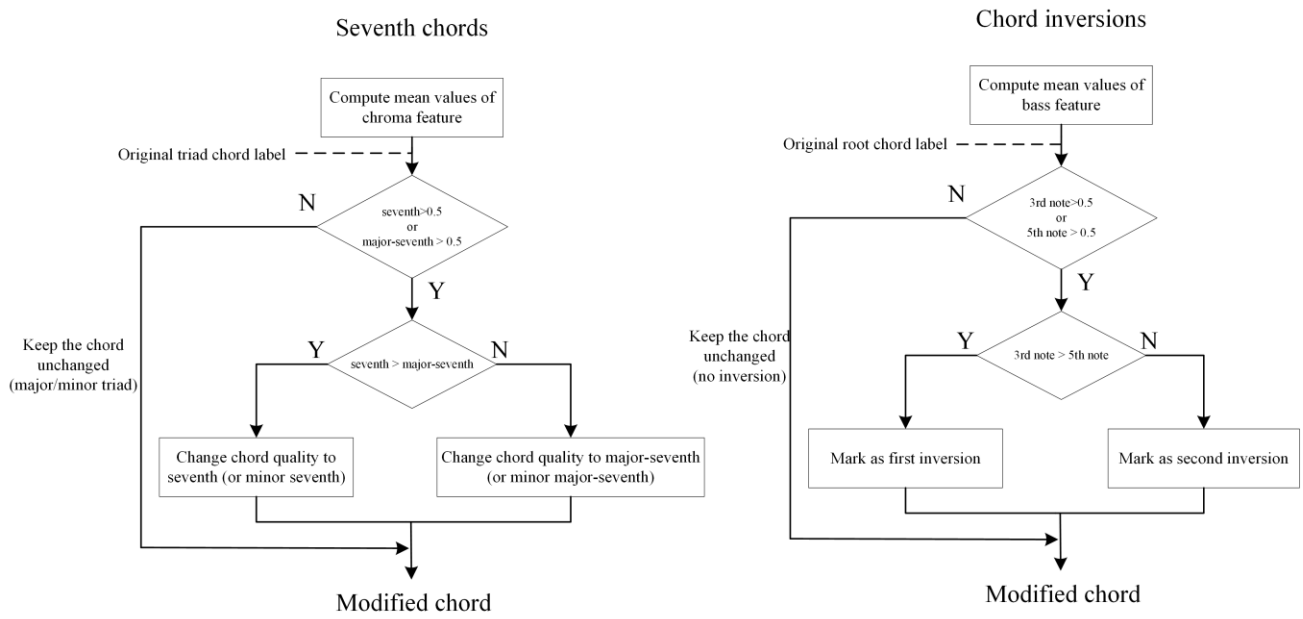

Figure 5.The flow chart of complex chord type decision process, for both seventh and chord inversions. Given the estimated chord triads and corresponding feature sequence, both types of complex chords are determined with simple thresholding rules and comparisons, based on average values of the feature.

6. TOWARDS LARGE VOCABULARY CHORD RECOGNITION

In the presented 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. When the chord vocabulary include seventh chords and chord inversions, this assumption no longer holds [5]. To recognize complex chords in a more reasonable way, in the presented system, we keep the vocabulary of the neural network system unchanged, and determine complex chord types in at post-processing stage.

In practice, the way that a human recognizes complex chords is generally divided into two steps: roughly estimate chords in triad chord level, then determine whether the sevenths or inversion is present in each triad, rather than regarding them as new independent chords. To mimic this process, we design a two-stage complex chord recognition method.

Concretely, the proposed system do this by modifying qualities (major or minor triad) and inversion types of each recognized chord signature. Given a chord signature (in the form of major or minor triad) and the feature sequence (normalized on each frame) of corresponding time region, first the mathematical mean of the feature value along the dimension is calculated on its third, fifth, seventh and major-seventh note, and bass feature value of its root, third and fifth note. Then the true quality and inversion are determined with an explicit thresholding metric. The decision flow of the process is shown in Fig. 5. In this way, the chord recognition system is able to support 61 types of chords if considering only the seventh chords, or 181 types of chords if further taking first and second chord inversions into account. At the same time, the recognition accuracy of triads is not affected.

12. REFERENCES

[1] F. Korzeniowski and G. Widmer, Feature Learning for Chord Recognition: The Deep Chroma Extractor, In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[2] C. Schorkhuber and A. Klapuri, Constant-Q Transform Toolbox for Music Processing. *7th Sound and Music Computing Conference*, Barcelona, Spain. 2010.

[3] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC Music Database: Popular, Classical, and Jazz Music Databases, In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR)*, pp.287-288, 2002.

[4] J.D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, pp.282-289, 2001.

[5] E.J Humphrey and J.P. Bello. Four timely insights on automatic chord estimation. In *Proceedings of 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[6] Colin Raffel. Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching. PhD Thesis, 2016.