Music Theory Inspired Policy Gradient Method for Piano Music Transcription

Juncheng Li, Shuhui Qu, Xinya Li, Yun Wang, Xinjian Li; Florian Metze
Language Technology Institute, Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
junchenl@cs.cmu.edu, shuhuiq@stanford.edu, xinyal,xinjianl,fmtze@cs.cmu.edu

Abstract

This paper presents a novel approach to transcribing polyphonic piano music to a symbolic representation by incorporating rule-based reward generalized from classical music theory by Reinforcement Learning (RL). We use convolutional recurrent neural networks (CRNNs) to predict the onset and duration frames of piano notes. We believe that good piano music generally conforms to rules of classical music theory. Thus, by penalizing heavily according to these rules, our RL transcriber is significantly less susceptible to noises that usually exists in audio recordings. As a result, our model results in over a 10% increase of accuracy over the best state-of-the-art methods on the MAPS dataset [9].

1 Introduction

Automatic music transcription is the task of transcribing a raw audio into a symbolic representation such as MIDI or sheet music. A successful approach to this task furthers numerous studies ranging from music information retrieval to musicology: accurate transcription enables direct search on melody, chord progression or short motif. Transcribing music is a difficult task; in this paper, we focus only on piano music. Several factors leading to its major difficulty. First, a piano note is not merely a fixed-duration sine wave at a certain frequency, but a harmony span across the full frequency band with fluctuating energy, and each piano also has its own unique sound signature, so an easy generalization could not be made across different pianos. Another important factor is that multiple notes are usually played simultaneously in piano music, which results in superimposition of notes in recordings: the colliding harmonics of different notes itself naturally creates a difficult source separation problem. Lastly, ambient noises such as background sounds or human speech, singing could severely impair the note transcription.

In our approach, we describe the timbral properties of transcribed piano with a set of rich spectral features. The energy of every hit piano note decays after an its onset; thus, monophonic piano music transcription is solved by peak detection algorithm in amplitude envelopes [17]. However, for polyphonic piano music, this approach fails since the same amplitude envelopes do not contain information of individual frequency regions of the signal, where note onsets and offsets may coincide. Classical studies like [17] also show that implicit onset detection schemes of equaling the onset time of a note to the time of its discovery from signals does not perform well, so we approach the transcriptions problem in two steps: detecting notes onset and predicting the frames.

The challenge in the first step is noise in sample recordings. Effective CRNN models from the past turn out to be rather sensitive to such noises. To provide a decent solution to this challenge, we incorporate in our Transcriber the REINFORCE algorithms by [13], which succeeded in excluding

*Equal contribution.

Preprint. Work in progress.
noise in sequential classification tasks. Note transcription also presents an additional challenge to our second step: the size of action space changes because each predicted chord usually contains multiple notes. To address this problem, we present a model that predicts whether a pitch is on or off, and we introduce classical music theory based reward term on top of the original loss functions to guide the CRNN network not to be ‘fooled’ by the noise. This is effective based on the assumption that good piano musics generally follow the rules of music theory. In this paper, we demonstrate our RL Transcriber could further improve upon the most recent state-of-the-art performance reported in [12] on MAPS dataset [9] for all 3 metrics measuring transcription quality: frame, note, and note with offset.

2 Background

2.1 Policy Gradient Method in Reinforcement Learning

In RL, let \( A \) be a set of action sequences, and \( p_\theta(a) \) be a distribution over action \( a \in A \) which is parameterized by \( \theta \). The objective of the REINFORCE algorithm is as following:We should mention what specific action space we are using in this task

\[
J(\theta) = \sum_{a \in A} p_\theta(a)r(a)
\]

(1)

where \( r(a) \) is the reward signal assigned to each possible action sequence (note transcription), and \( J(\theta) \) is the expected reward under the distribution of possible action sequences. The gradient of the objective \( J \) is as following:

\[
\nabla J(\theta) = \sum_{a \in A} p_\theta(a)\nabla \log p_\theta(a)r(a)
\]

(2)

Due to the high-dimensional sequential action space, the optimization problem is non-trivial. We thus approximate the gradient by sampling. We sample the overall note transcription \( a_k \) from \( p_\theta(a) \). We can calculate the reward function of \( a_k \). The approximate gradient is then computed by averaging the gradient of \( K \) sampled actions.

\[
\nabla J(\theta) \approx \frac{1}{K} \sum_{k=1}^{K} \nabla \log p_\theta(a_k)r(a_k)
\]

(3)

To reduce the variance of the gradient estimate, we introduce a baseline reward \( b \). The gradient function is as following:

\[
\nabla J(\theta) \approx \frac{1}{K} \sum_{k=1}^{K} \nabla \log p_\theta(a_k)(r(a_k) - b)
\]

(4)

In general, REINFORCE algorithm learns model parameters \( \theta \) according to this approximate gradient. The log-probability of actions that lead to high reward are increased, and those lead to low reward are decreased.

2.2 Piano Music Transcription Using Deep Neural Network

Since modern pianos have 88 keys, we can simplify the transcription problem into one of predicting a binary indicator of 88 playable notes for each frame throughout time. Similar to speech recognition systems, such end-to-end piano music transcription systems comprise of acoustic model and a music language model, the former for predicting a fram’s pitch and the later for correlations modelling between notes. Convolutional neural network (CNN) is believed to be the most suitable for acoustic model due to its lighter computational cost (compared to fully connected DNNs) and capability to learn spacial invariant low-level features along both time and frequency axes. Recurrent neural network (RNN) is commonly used in music language models for its ability to model long term correlation. Both kind of predictions are integrated by a probabilistic graphical model [23] similar to HMM, and finally, beam search is used to decode the output. In this paper, we focus on the acoustic model and do not consider the complementary language model. Significant future improvement on our work can definitely be made by heading down this direction.
3 RL Transcriber Design

3.1 Model Architecture & Configuration

We firstly use librosa [19] to compute the log mel-spectrum to generate required spectral features. We adopted the parameters suggested by [14], with a filterbank which keeps 48 bins per octave from the input raw audio, resulting in 229 log-spaced frequency bins with a hop length of 512. Our FFT window size is 2048, and we sampled at 16kHz.

We build both our onset detector and frame detector in a CRNN architecture based on [14]’s. Our onset detector is followed by a bidirectional LSTM with 128 units in both forward and backward directions. The prediction is done by a fully connected sigmoid layer (threshold = 0.5) with 88 probabilities of an onset for each piano key. Our frame activation detector uses the same CRNN architecture, but it takes the onset detector’s output and feed it into the its own bi-directional LSTM layer. It is also followed by fully connected sigmoid layer with 88 probabilities to predict whether the frame is on or off. This architecture is sketched in Fig. 1. According to [12], during our inference, the frame predictor does not start unless the onset predictor predicts positive.

3.2 Reward shaping

Training with the REINFORCE algorithm requires a well-designed reward function. We designed two different types of reward to learn the pitches: 1) metrics-driven reward and 2) music theory reward.

3.2.1 Metrics-driven reward

The metrics-driven reward is the F1 score which both the model’s frames and onsets will be evaluated on. Applying this reward enables the model to directly optimize the evaluation metrics.

\[ r_{\text{onset},f_1}^M(\hat{y}) = F1(\hat{y}_{\text{onset}}, y_{\text{onset}}) \]

\[ r_{\text{frame},f_1}^M(\hat{y}) = F1(\hat{y}_{\text{frame}}, y_{\text{frame}}) \]

where \( \hat{y} = f_\theta(x) \) is the output vector (logits) of the network, \( \hat{y} \) is the onset note and frame note prediction sampled from \( \hat{y} \), and \( y \) the ground truth of notes.

3.2.2 Music theory reward

Ultimately, we do not want a transcription that only optimizes the evaluation metrics; we also hope that it is composed of pleasant-sounding notes that follow rules of basic music theory. Thus, we further summarized several rules based on the counterpoint principles on page 42 of A Practical Approach to Eighteenth-Century Counterpoint [10] and in Constraint Programming in Music [25]. Specifically, we have 7 rules in total and designed rewards accordingly.
• $r_{duration}(a)$: Note duration may only change slowly across a voice, neighbouring notes are either of equal length or differ by 50% at maximum. Notes that don’t follow the rule would be penalized by -0.0001.

• $r_{start-end}(a)$: The first and last note of the entire piece must start and end with the root C. Notes that don’t follow the rule would be penalized by -0.01.

• $r_{pitch}(a)$: The maximum and minimum pitch in a phrase occurs exactly once and it is not the first or last note of the phrase. Here we consider half of a melody a phrase. Notes in a phrase that don’t follow the pitch rule would be penalized by -0.0001.

• $r_{key}(a)$: All notes should belong to the same key. e.g. If the key is C-major, notes in the piece should all be middle C

• $r_{repeat}(a)$: Unless a note is held, a single tone should not be repeated more than four times in a row. Repeating notes that are more than 5 times get a penalty of -0.00001.

• $r_{correlate}(a)$: We penalize the model by -0.01 if the auto-correlation coefficient is greater than .15.

• $r_{interval}(a)$: Good music should move by a mixture of small steps and larger harmonic intervals, large leaps more than a fifth receives negative rewards of 0.001.

From our experience, the music theory might be too specific or restrictive in some cases and might cause the result to fluctuate; the system stability is also very sensitive to the hand crafted penalty amount. The numbers we report here are from the best empirical results we have acquired.

$$r_{MT}(a) = r_{duration}(a) + r_{start-end}(a) + r_{pitch}(a) + r_{key}(a) + r_{repeat}(a) + r_{correlate}(a) + r_{interval}(a) \quad (7)$$

The combination of reward is as follows:

$$r(a) = \gamma r_{onset, f1}^M(\hat{y}) + \gamma r_{frame, f1}^M(\hat{y}) + \delta r_{onset}^{MT}(\hat{y}) + \delta r_{frame}^{MT}(\hat{y}) \quad (8)$$

where $\gamma$ and $\delta$ are the weight parameter of the reward function.

The basic loss function for our RL transcriber are the binary cross-entropy applied frame-wise and element-wise:

$$l_{onset}(y, \hat{y}) = -\sum_{t=1}^{T} (y_t \cdot \log(\hat{y}_t) + (1 - y_t) \cdot \log(1 - \hat{y}_t))$$

$$l_{frame}(y, \hat{y}) = -\sum_{t=1}^{T} (y_t \cdot \log(\hat{y}_t) + (1 - y_t) \cdot \log(1 - \hat{y}_t))$$

where $\hat{y}_t$ is the output vector of the network at time $t$, and $y_t$ the ground truth at time $t$. Thus, the overall objective function is as following:

$$L(\theta) = l_{onset}(y, \hat{y}) + l_{frame}(y, \hat{y}) - J(\theta) \quad (9)$$

4 Experiments

4.1 MAPS Dataset

We use the MAPS dataset\cite{9} that has 31 GB of CD-quality recordings and corresponding annotations of isolated notes, chords, and complete piano pieces. The piano pieces in this dataset consist of synthesized sounds and recorded ones on a Yamaha Disklavier player piano. We use the synthesized pieces (the MUS set: “pieces of piano music”\cite{9}) as the training split and the recorded pieces as the test split, as proposed in\cite{12}, because we often do not have access to well-processed real-word recordings.
4.2 Implementation Detail

We trained our RL transcriber model on the MAPS dataset with processing process described in Section 4.1 using the Adam optimizer, a batch size of 8, a learning rate of 0.0006, and a gradient clipping L2-norm of 3. The same hyperparameters were used to train all models, including those in the ablation study to ensure evaluation consistency.

We compare three different reward combinations for implementing our RL transcriber:

- The reward weight for metrics driven reward is $\gamma = 0.02$
- The reward weight for music theory reward is $\delta = 0.5$
- The reward weight for metrics driven reward is $\gamma = 0.015$, music theory reward is $\delta = 0.3$.

We also re-implement the model described in "onset of frame" [12], Sigta [23], and Kelz [14] using their default hyperparameters. We compare our method to the performance of these approaches to ensure evaluation consistency.

4.3 Metrics

We use both frame-level metric and note-level metric to evaluate our model. We use the MIR eval library to calculate note-based precision, recall, and F1 scores. The note-level metric requires onsets within $\pm 50$ms of ground truth, and offsets resulting in notes of durations within 20% of the ground truth. Frame-based scores are calculated according to the standard metric defined in [12]. Both frame and note scores are calculated per audio; the means of both scores is presented as the final metric for a given collection of audios.

5 Results

Our results are presented in Table 1. Our RL transcriber model not only produces better note-based performance, but also the best frame-level and note-based scores that include offsets. Our overall F1 score is slightly improved over the traditional onset-and-frame methods due to its metric-drive nature. While we do not see a major improvement on the “Frame” or “Note” metric by integrating music theory based rewards, we see the “Note with offset” metric becomes significant boosted, which shows that our hand-crafted rewards might be better at tackling notes’ offset cases.

<table>
<thead>
<tr>
<th></th>
<th>Frame</th>
<th>Note</th>
<th>Note with offset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>Sigtia[23]</td>
<td>71.99</td>
<td>73.32</td>
<td>72.22</td>
</tr>
<tr>
<td>Kelz[14]</td>
<td>81.18</td>
<td>65.07</td>
<td>71.60</td>
</tr>
<tr>
<td>Hawthorne[12]</td>
<td>85.44</td>
<td>71.24</td>
<td>77.23</td>
</tr>
<tr>
<td>$\gamma = 0.02, \delta = 0$</td>
<td>87.82</td>
<td>68.94</td>
<td>77.23</td>
</tr>
<tr>
<td>$\gamma = 0, \delta = 0.5$</td>
<td>86.12</td>
<td>73.33</td>
<td>77.52</td>
</tr>
<tr>
<td>$\gamma = 0.015, \delta = 0.3$</td>
<td>89.24</td>
<td>73.00</td>
<td>80.31</td>
</tr>
</tbody>
</table>

Table 1: 3 Benchmarks of the transcription accuracy, the last 3 rows are results from our RL transcriber with different reward weighting combination.

5.1 Ablation analysis

To understand the individual importance of each piece in our model, we conduct an ablation study. We consider using different combination of reward function, and training with or without baseline:

- we set $\gamma = 0.1, \delta = 0.0$, and train REINFORCE with baseline,
- $\gamma = 0.02, \delta = 0$, w/ baseline;
- $\gamma = 0.1, \delta = 0$, w/o baseline;
- $\gamma = 0.02, \delta = 0$, w/o baseline;
These results show the importance of each component of the reward function. Adding minimum metrics driven reward results in the improvement of both note and note with offset score while maintaining the frame score. Adding music theory driven rewards did not improve Frame or Note performance as expected, this might be due to the fact that the baseline accuracy is already high, and handcrafted rewards might be biased towards only limited musical phenomenon. While we do see the NoteWithOffset metric was improved by the music theory reward with a good margin. This indicates that our handcrafted rewards is effective at detecting the offset of notes. Training the model using REINFORCE with baseline improves the final score by 8%. To our ears, the perceptual decrease in audio quality is best tracked by using both metric driven reward and music theory reward.

<table>
<thead>
<tr>
<th>γ</th>
<th>δ</th>
<th>F1 Frame</th>
<th>F1 Note</th>
<th>F1 Note with offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.015</td>
<td>0.3</td>
<td>80.31</td>
<td>86.77</td>
<td>54.91</td>
</tr>
<tr>
<td>0.015</td>
<td>0.3</td>
<td>77.53</td>
<td>81.19</td>
<td>50.20</td>
</tr>
<tr>
<td>0.02</td>
<td>0.3</td>
<td>74.80</td>
<td>80.15</td>
<td>51.62</td>
</tr>
<tr>
<td>0.02</td>
<td>0.5</td>
<td>77.52</td>
<td>81.15</td>
<td>53.00</td>
</tr>
<tr>
<td>0.1</td>
<td>0.3</td>
<td>74.77</td>
<td>79.94</td>
<td>48.12</td>
</tr>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>73.23</td>
<td>83.81</td>
<td>52.20</td>
</tr>
<tr>
<td>0</td>
<td>0.15</td>
<td>76.89</td>
<td>82.32</td>
<td>51.72</td>
</tr>
<tr>
<td>0</td>
<td>0.3</td>
<td>72.77</td>
<td>73.96</td>
<td>46.33</td>
</tr>
<tr>
<td>0</td>
<td>0.5</td>
<td>72.90</td>
<td>72.15</td>
<td>49.71</td>
</tr>
<tr>
<td>0.015</td>
<td>0.5</td>
<td>80.31</td>
<td>86.77</td>
<td>54.91</td>
</tr>
</tbody>
</table>

Table 2: Ablation test of the systems with and without baseline models

6 Discussions

In this work, we demonstrate a novel approach of transcribing piano music by incorporating generation rules of the music itself, which shows an increased ability to filter out noises, and thus yields significant improvements of a 10% relative improvement compared with other methods.

For potential future improvement, we will need to improve our model on a much larger and more representative piano dataset of various genres and various recording environments. Further improvements on the music theory rules are also certainly possible. Additionally, we hope to explore more deep learning architectures like attention mechanism [26] to improve model’s performance.

7 Related Work

Automatic music transcription (AMT) is a task to transcribe the music audio signal into some form of music notation such as sheet music and MIDI file. Automatic music transcription have several typical subtasks such as pitch detection [11], instrument identification [18], rhythm parsing [21] and onset detection [3]. There are multiple applications that use AMT as an underlying component such as music information retrieval [1] and musicology analysis [2]. While monophonic AMT is considered a solved task, polyphonic AMT still remains open because multiple notes overlap in both time domain and frequency domain.

Traditionally, AMT exploits Non-negative Matrix Factorization (NMF) to decompose music audio into known pitch templates of an instrument [24]. Multiple constraints such as sparseness [7] and temporal continuity [27] and harmonicity [4] were shown to improve the transcription quality. Additionally,
exploiting instrument specific features also proved to be helpful. In the case of piano transcription, models of various note stages: Attack, Decay, Sustain, and Release leads to improvement of the transcription [6] [5].

In recent years, with promising progress in deep learning models, AMT community also tries to propose approaches with deep neural network to tackle the task. For example, Nam proposed a model to use deep belief network to learn representations from spectrum [20]. Sigtia Using RNN as music language model to predict next note [22]. Keltz investigated a glass ceiling problem with convolutional neural network [15]. Most of those works, however, treated the AMT task as a single stack neural network problem, for which a single neural network would generate all necessary music information such as onset, offset and pitch. In contrast to these models, researchers recently proposed a new model to predict onsets and frames with two stacks of neural networks [12]. One stack is to predict onsets, and the other is to classify labels for each frame. As a result, the accuracy of frame classification was improved by conditioning the onset results. Analogous to this work, explicit modeling of onset classification have also been proven to be useful with NMF [6] and CNN [28].

Another typical music related task using deep neural network is the music generation task. With the explosive development of hardware and more readily available computation resources, RNN and LSTM have gained popularity as a model to generate music. The way it generates notes in music is similar to the way it does in text generation task [16]. In text generation task, a character RNN model is trained in a supervised manner to predict the distribution of next character given previous sentences. Similarly, in the case of music generation, a note RNN is used instead to predict next note [8].

One known issue for generating long-step sequences in such a supervised learning is that it fails to generate a globally coherent structure. This had caused the character RNN to fail to generate sentences with a coherent topic and note RNN to generate coherent melodies.

One approach to tackle this problem in note RNN was to tune the note RNN with reinforcement learning [13]. The Google brain team formulate the music generation task as a reinforcement learning task. Instead of optimizing the probability of next note directly with supervised learning, they proposed a reward neural network. The reward neural network aims at making the note RNN to learn a coherent structure by using music theory.

In this work, we proposed a model which exploits both reinforcement learning and music theory to tackle the AMT task.
References


