An audio based piano performance evaluation method using deep neural network based acoustic modeling

Jing Pan[1], *Ming Li*[1], *Zhanmei Song*[2], *Xin Li*[2], *Xiaolin Liu*[2], *Hua Yi*[2], *Manman Zhu*[2]

[1]SYSU-CMU Joint Institute of Engineering, School of Electronics and Information Technology
Sun Yat-sen University, Guangzhou, China
[2]School of Preschool Education, Shandong Yingcai University, Jinan, China

liming46@mail.sysu.edu.cn songzhanmei@126.com

Abstract

In this paper, we propose an annotated piano performance evaluation dataset with 185 audio pieces and a method to evaluate the performance of piano beginners based on their audio recordings. The proposed framework includes three parts: piano key posterior probability extraction, Dynamic Time Warping (DTW) based matching and performance score regression. First, a deep neural network model is trained to extract 88 dimensional piano key features from Constant-Q Transform (CQT) spectrum. The proposed acoustic model shows high robustness to the recording environments. Second, we employ the DTW algorithm on the high-level piano key feature sequences to align the input with the template. Upon the alignment, we extract multiple global matching features that could reflect the similarity between the input and the template. Finally, we apply linear regression upon these matching features with the scores annotated by expertise in training data to estimate performance scores for test audio. Experimental results show that our automatic evaluation method achieves 2.64 average absolute score error in score range from 0 to 100, and 0.73 average correlation coefficient on our in-house collected YCU-MPPE-II dataset.

Index Terms: Piano Performance Evaluation, Music Analysis, Convolutional Neural Network, Dynamic Time Warping, Computer Assisted Piano Learning

1. Introduction

Nowadays, more and more beginners are trying to learn musical instruments by themselves with on-line resources, and learning piano is a popular choice. However, the beginners need a lot of practice and effective practicing needs immediate feedback, for example, a piece of advice from piano instructor. Since manually performance evaluation is both time and labor consuming, we intend to propose an audio based piano performance evaluation system to offer feedbacks to the beginners as a kind of Computer Assisted Piano Learning (CAPL) system. Intuitively, a good performance should have large similarity with the template and vice versa. This system takes the piano audio recording as input, and outputs multiple objective performance evaluation metrics, such as the overall performance score and the mistakes that the performer has made. In this paper, we mainly focus on predicting the expert generated performance score, which is an overall feedback based on the performance.

There has been some efforts made on the automatic piano performance evaluation. Morita, et.al[1] take MIDI sequence generated by the electrical piano when the player is playing on the keyboard as inputs. The MIDI sequence records the onset, velocity and duration of each music note which are used to predict the performance score by spline regression and the average correlation coefficient between system estimated scores and experts evaluated scores is 0.6. Akinaga, et.al[2] also take the MIDI sequences as input and apply Karhunen-Loeve(KL) expansion and K-nearest neighbors (KNN) algorithm on the interval, velocity and duration of each note to predict the performance score. Existing methods mainly focus on the application to electrical pianos with MIDI output function. In real life, many pianos can not generate MIDI files and the extra MIDI collection equipments cost also prevents its large scale usage. In our case, the system input is the audio signal captured by any microphone which makes the application useful for all types of pianos and possible on all mobile devices.

Generally when we try to evaluate an input audio's performance, the music score is a nature solid ground truth. Therefore, we need to transcribe both the music score and the input audio into MIDI sequences to measure their similarity. Transcribing music audio to MIDI sequence itself is a well defined task, called Automatic Music Transcription (AMT)[3]. Currently, most of the proposed AMT method are based on describing the input spectrogram as the weighted combination of basis spectra corresponding to the music pitches, which could be estimated by Non-Negative Matrix Factorization and sparse composition [4] [5]. The unsupervised factorization often leads to small correspondence to the music pitches, causing issues in interrupting the results. The problem is often addressed by applying harmonic constraints in the training stage [6], [7]. The support vector machine is also applied to AMT by classifying the normalized magnitude spectra [8]. Recently, deep learning[9] has been applied to AMT. Dixon and Benetos proposed an End-to-End deep neural network approach with a combined Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) framework, transcribing the spectrogram to the estimated music score [10].

Since our target is to estimate the performance score rather than to transcribe the melody, we adopt the piano key posterior probabilities (PKPP) generated by the acoustic model as our features for the subsequent matching and regression. The low-level features such as spectrum and MFCC contain not only the piano key information but also the environmental noise, reverberation, and channel mismatch. But the AMT system's acoustic model output (such as the PKPP in [10]) can be seen as a better approximation to what the performer really plays or which keys the performer really touched on the piano with less variabilities, which would potentially benefit the performance score estimation. We train a convolutional neural network as an acoustic model of the piano sound. The input of the network is Constant Q Transform(CQT) [11] spectrum and the output is an 88-dimensional PKPP vector indicating the probability of being pressed for every key on the piano keyboard.

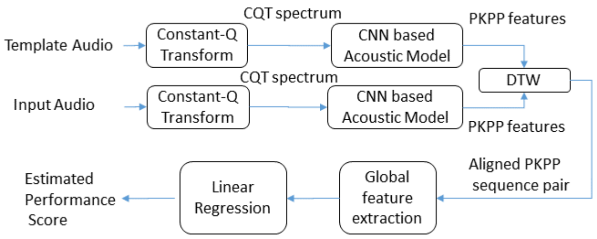

Figure 1: *Overview of the proposed method*

Moreover, after the PKPP feature sequences are extracted, the basic methodology of the proposed performance evaluation method is still the similarity measurement between the input audio and the template audio. The template audio reflects the 'criteria' of annotators in some way. For an input test audio, the more similar it is compared to the template audio, the higher score it might get. So for each song in the database, we select the audio recording piece with the highest performance score as the template audio while the others are used as training and testing data for cross validation. There is a common sense that the template for comparison should be the song score. However, as mentioned above, the song score could not reflect the 'criteria'. We did an experiment for this, and results shows that using the audio with highest score as template is better.

The overview of our method is shown in Figure 1. First, we employ the pre-trained network to extract the PKPP feature sequences from CQT spectrum. Second, DTW algorithm is used for aligning feature sequence pair, then we extract multiple global matching features which contain the differences between the template and the input in terms of similarities and rhythms. Finally, we apply linear regression upon these matching features with the labeled training data to estimate the expert generated performance score for test audio.

The rest of this paper is organized as follows. Our released dataset and acoustic model design are introduced in Section 2 and 3, respectively. The DTW alignment, global matching feature extraction and linear regression are described in Section 4. The experimental results and analysis are presented in Section 5 followed by the conclusion and future works in Section 6.

2. Database Description

The Yingcai Multimodal Piano Performance Evaluation phase II Database (YCU-MPPE-II) [1] was collected at Shandong Yingcai University in 2016. It contains video and audio files that were recorded in multiple examination sessions of a piano course. The three different recording channels are namely video camera microphone, cellphone microphone, and cellphone line in, as shown in Table 1. There are totally 185 audio pieces from 5 songs in this database. Each audio piece corresponds to a single performer. One piano teacher (no prior knowledge about the students) listen the audio recordings and give a performance score for each piece as the ground truth label. There are five different polyphonic songs in the YCU-MPPE-II dataset. The songs are indicated with their numbers since their names are in Chinese. Each of the audio pieces is performed by different players. The amount of audio pieces for each song and each recording channel is shown in Table 1.

Table 1: *Number of audio pieces for each song and each recording channel*

Song Number	Audio pieces	Cellphone Microphone	Cellphone Line In	Video Camera Microphone
1	65	14	51	0
2	25	9	16	0
3	25	19	6	0
4	18	0	0	18
5	52	0	0	52
total	185	42	73	70

3. Acoustic Model Design

3.1. Training Dataset

We train and evaluate our acoustic model on the MAPS dataset [12]. This dataset consists of 270 pieces of piano sound and corresponding MIDI annotations. 210 of these audio are synthesized recordings and the remaining 60 are real recordings. In our work, we train the acoustic model on the synthesized recordings and test it on the real ones.

3.2. Audio Pre-processing

We employ Constant-Q-Transform spectrum, a kind of time-frequency representation of the audio, as the network input. CQT is an important transformation from time domain to frequency domain, similar to the short-time Fourier Transform. CQT could be considered as a series of logarithmically spaced filters which makes it linear in pitch along frequency axis. The details and calculation of Constant-Q Transform could be seen in [11]. CQT is widely applied in music processing due to its log-scale property in frequency domain.

We downsample the audio to 16kHz, and compute CQT with 64 ms frame size, 32 ms frame shift and 36 bins per octave, producing 294 dimensional vectors for each frame. We then perform utterance level mean and variance normalization on each dimension of the spectrum. Furthermore, we sample the MIDI annotation every 32ms to get the corresponding frame label, which is an 88-dimensional binary piano key vector, representing the status (on or off) of the 88 piano keys. Note that, since multiple keys in the keyboard can be touched at the same time, there could be multiple ones in the piano key label vector.

3.3. Network Architecture and Training

For the acoustic model, we adopt the convolutional neural network (CNN)[13]. The motivation is that CNN is proved to be capable of achieving better accuracy in AMT tasks[14, 15]. The chords and harmonics in the piano music contain some certain patterns which may exist in the spectrum context.

The architecture of the network is basically the combination of convolutional layers and fully-connected layers as shown in Figure 2 . The input of the network is a context window of frames and the target is the central frame. We configure the network as shown below: The window size is $w_s = 11$. The number of convolutional layers is $L_c = 2$. The number of kernels for each conv layer is uniformly $n_l = 50$. The kernel size for each conv layer is $k_1 = \{30, 5\}$ and $k_2 = \{10, 3\}$. All of the convolutional layers are activated by hyperbolic tangent function. Each conv layer is followed by a max pooling layer whose pooling size is $\{3, 1\}$. The number of fully connected layers are $L_{fc} = 4$. The numbers of hidden units are

[1] YCU-MPPE-II is freely public open for research purposes. Please send email to liming46@mail.sysu.edu.cn to request the database.

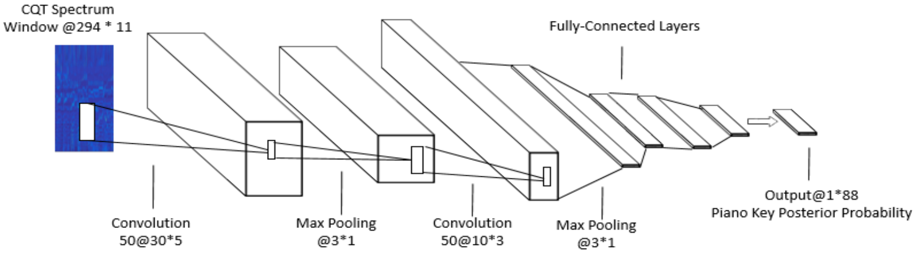

Figure 2: *Architecture of the acoustic model*

respectively $\{500, 250, 250, 125\}$. The activation of fully connected layers is sigmoid function. To avoid overfitting, we set the dropout with rate 0.5 to all layers. After pre-processing, the network is trained on over $1,500,000$ samples and tested on around $490,000$ samples. To monitor the performance while training, we set 20% of the training set as validation set.

3.4. Analysis

It is necessary to evaluate the frame level accuracy before applying the model to our task. To measure the accuracy, a global threshold is set on the output PKPP vector. If the value is larger than the threshold, this particular dimension of piano key vector is simply set to 1, otherwise it is set to 0. We choose F1-score to measure the accuracy of our model. The metric is defined as:

$$P = \frac{1}{T}\sum_{t=1}^{T}\frac{TP[t]}{TP[t] + FP[t]}$$

$$R = \frac{1}{T}\sum_{t=1}^{T}\frac{TP[t]}{TP[t] + FN[t]}$$

$$F = \frac{2 \times P \times R}{P + R}$$

where $TP[t]$, $FP[t]$ and $FN[t]$ respectively indicate the number of true positive, false positive and false negative events at frame t. T is the number of frames in the testing set.

The performance of the proposed network on training set, validation set and testing set is shown in Table 2. Obviously there's no significant difference among the results of these three datasets. It's remarkable that the validation data is synthesized recording and the testing data is recorded from real in-door environment. Though not perfect, the model performs equally on the validation and testing set. Therefore, we can consider that our acoustic model did learn the acoustic features of the piano sound and it can robustly extract the piano key posterior probability features (PKPP) for the subsequent modeling.

Table 2: *The performance of our acoustic model on MAPS data*

Data	Training	Validation	Testing
Frame level F1-score	0.6483	0.6290	0.6203

4. Matching and Regression

4.1. Dynamic Time Warping

After PKPP feature sequences are obtained from the acoustic model mentioned above, we need to measure the similarity between the input and template sequences. Methods to measure sequence similarity vary, such as the end-to-end method applied in NLP, LSTM. However, the sequences contain as long as 7000 thousand frames. It's hard to find a practical method for these long sequences.

Dynamic Time Warping is a popular time series analysis method aiming at aligning two temporal sequences with different duration and speed. The scheme of DTW is essentially to find an optimal alignment through dynamic programming. Suppose $C \in \mathbb{R}^{M\times N}$ is the searching grid matrix, $S \in \mathbb{R}^{D\times M}$ is the feature sequence of template audio and $P \in \mathbb{R}^{D\times N}$ is the feature sequence of input audio, where D is the feature dimension. For each node $node_{i,j}$ in C, corresponding to the matching status for the i^{th} frame of the template sequence and the k^{th} frame of the template sequence, has 5 possible transitions respectively from $node_{x,y}$. The optimal transition for node $node_{i,j}$ is given by:

$$C_{i,j} = min\{C_{x,y} + t_{x,y,i,j} + d_{i,j}\} \tag{1}$$

$$x, y \in \{(i, j-1), (i-1, j), (i-1, j-1), (i-1, j-2), (i-2, j-1)\}$$

where $d_{i,k}$ is the node cost of $node_{i,j}$ given by the cosine distance between input frame i and template frame j, $t_{x,y,i,j}$ is the transition cost from $node_{x,y}$ to $node_{i,j}$ and $C_{i,j}$ is the accumulated cost through the path.

$t_{x,y,i,j}$ is an important parameter for the matching. It would significantly influence the optimal transition. For instance, if the transition from $node_{i-1,j}$ to $node_{i,j}$ is chosen, it means that input frames $i-1$ and i are matched to a single template frame j, and that at least at this moment the player in the input test audio plays more slowly than the player in the template audio, reflecting the difference in rhythms. Transition cost should be assigned to this situation as a punishment factor.

The diagonal transition means that there is no rhythm difference at this point, therefore no transition cost should be assigned to it. In order to make the searching more consistent with the reality, we tune the rest 4 transition costs by grid search.

4.2. Global Matching Features

Through the DTW algorithm, an optimal matching path made up of transitions and the corresponding minimum overall cost could be obtained. As Fig 3 shows, the red line marks the optimal matching path, and the input sequence is shorter than the template sequence. Our DTW algorithm implementation has the function of stopping early which enable the player to play partial song, no need to play to the end. Almost the whole input sequence is matched to about the first 2700 frames of model. The overall cost of the path is 1666.8.

The matching path achieves a satisfying consistency with the ground truth, that the test audio just performs first 70% of the song while the template audio finishes it, and that there are

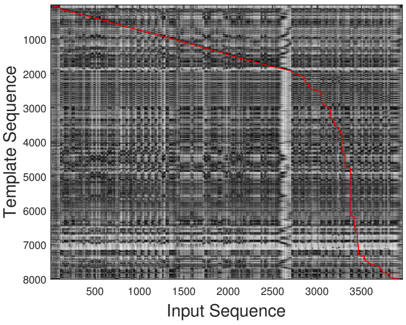

Figure 3: *An example of DTW matching path*

no significantly rhythm and note errors in the input test audio. Evidently, the matching path contains a lot more detail information about the similarity. We tried to extract some matching features from the optimal matching path, to help estimate the performance score.

We adopt three major matching features:

- Average Cost. The average cost is obtained by $C_{avr} = C_{overall}/N_{frames}$, where $C_{overall}$ is the overall cost and N_{frames} is the number of frames of the input feature sequence. Comparing with the overall cost, the average cost is more robust against duration differences.
- Number of halt in path. The term 'halt' is defined as the transition from $node_{i-1,j}$ to $node_{i,j}$, as it reflects that the performance is stalled at that point, compared to the template. This feature tells the local difference of the rhythm.
- Frame Ratio. Suppose the optimal path finally matches m frames of input to n frames of template , then $frameratio = |(m/n) - 1|$. This feature gives the overall difference in rhythm and integrity.

4.3. Linear Regression

Due to the small scale property of our training data and there are only three matching features that need to model, we use linear least square minimization to solve the problem.

In our model, we need to solve the coefficient β_j for the linear regression $y = \beta_1 p_1 + \beta_2 p_2 + \beta_3 p_3 + \beta_4$, where p_1, p_2, p_3 denote the three matching features: average cost, number of stalls and frame ratio, respectively. The least square could easily be solved by pesudo inverse:

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

where Y is the performance score value, $X = [p_1, p_2, p_3, 1]^t$.

5. Experimental Results

We performed our experiments on the YCU-MPPE-II database. Each audio piece has a performance score in the rage of $[0, 100]$. The details of the database is presented in Section 2.

Despite the PKPP feature sequence, we also evaluated the system with CQT spectrum directly as the feature sequence to DTW for comparison. Constrained by the size of the data, we apply Leave-One-Out (LOO) cross validation to evaluate our method. Our method is song-dependent, which means the cross validation is employed among the audios belong to the same song. Recall that for each song, we select the audio with the highest score as the template audio and validation is done upon the rest audio. In each round of Leave-One-Out validation, we select one of them as the input test audio, and remaining ones as training set. Coefficients trained from training set are applied to the input test audio to obtain the estimated score. Rotate and repeat this protocol till all of the pieces are evaluated (despite the one with the highest performance score which is selected as the template).

Table 3: *Mean Absolute Error (MAE) and Correlation Coefficient (CC) of the proposed method with both CQT feature and CNN PKPP feature as the inputs of DTW*

Song Number	Matching with CQT feature		Matching with PKPP feature	
	MAE	CC	MAE	CC
1	2.27	0.70	2.15	0.76
2	4.04	0.76	2.92	0.91
3	3.48	0.44	3.34	0.48
4	3.77	0.47	2.82	0.74
5	2.65	0.53	1.98	0.78
Average	**3.24**	**0.58**	**2.64**	**0.73**

After the LOO iterations, the mean absolute error (MAE) and correlation coefficient (CC) results are shown in Table 3, where error is defined by $|\text{estimated score} - \text{annotation score}|$.

The score is ranged from 0 to 100. We can find out that the mean absolute error which indicates the average difference between the human annotated and system estimated performance score is 2.64 points. Moreover, the MAE and CC of the scores estimated with PKPP features is also significantly better than the one from CQT features (2.64 vs 3.24 and 0.73vs 0.58). Therefore, it's clear that the CNN based acoustic model did improve the performance of the matching, comparing to the raw CQT spectrum.

6. Conclusion

In this paper, we propose an audio based performance evaluation method. Our method is based on the CNN acoustic model, DTW matching, global matching feature extraction and linear regression. The acoustic model is to extract the robust piano key level features. We apply DTW matching on the template and input PKPP feature sequences pair in order to extract some global matching features which can show the similarity between template and input audio. Upon these matching features, we employ linear regression to estimate the performance score for input test audio. Experimental results show that out method perform high correlation and low absolute error with scores annotated by piano experts.

One of our major contributions is that we propose a robust acoustic model that can be used to extract piano key features for piano performance evaluation, which significantly improves the overall performance. Limitations of our method do exist though. The regression coefficient obtained by the training set could only be applied to the input test audio from the same song, which is song-dependent. For each song, there has to be enough training data to achieve good results. Besides, we use linear regression to fit the regression model coefficient, which is different to the approach how human annotators work. Usually the human annotators based on nonlinear mapping of multiple factors. In future, more complicated model can be proposed to address this problem once we collect a large scale database.

7. References

[1] S. Morita and N. Emura, "Evaluation of a scale performance on the piano using spline and regression models," in *Proc. International Symposium on Performance Science*, 2009.

[2] S. Akinaga and M. Miura, "Toward realizing automatic evaluation of playing scales on the piano," in *Proc. International Conference on Music Perception and Cognition*, 2006.

[3] M. Piszczalski and B. A. Galler, "Automatic music transcription," *Computer Music Journal*, vol. 1, no. 4, pp. 24–31, Nov. 1977.

[4] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2003, pp. 177–180.

[5] S. A. Abdallah and M. D. Plumbley, "Polyphonic music transcription by non-negative sparse coding of power spectra," in *Proc. 5th Int. Soc. Music Inf. Retrieval Conf. (ISMIR)*, 2004, pp. 318–325.

[6] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, pp. 528–537, 2010.

[7] ——, "Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 3, pp. 538–549, 2010.

[8] G. E. Poliner and D. P. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, pp. 154–154, 2007.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[10] S. Dixon, S. Sigtia, and E. Benetos, "An end-to-end neural network for polyphonic piano music transcription," *IEEE Trans. Audio Speech Lang. Process.*, vol. 24, no. 5, pp. 927–939, 2016.

[11] J. C. Brown, "Calculation of a constant q spectral transform," *J.Acoust.Soc.Am.*, vol. 89, no. 1, pp. 425–434, Jan. 1991.

[12] V. Emiya, R.Badeau, and B.David, "Multipitch estimation of piano sounds using a new probabilistic spectral with group sparsity," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 6, pp. 1643–16 154, 2010.

[13] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time-series," *The Handbook of Brain Theory and Neural Networks*, 1995.

[14] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio chord recognition with recurrent neural networks," in *Proc. 13th Int.Soc.Music Inf.Retrieval Conf.(ISMIR)*, 2013, pp. 335–340.

[15] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kegl, "Aggregate features and adaboost for music classification," *Machine Learning*, vol. 65, no. 2, pp. 927–939, 2006.