

MML 2020

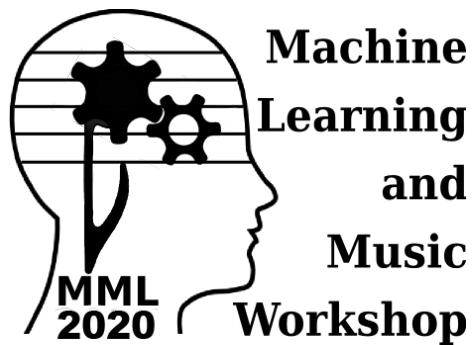
13th International Workshop on Machine Learning and Music

at

ECML/PKDD 2020

18.9.2020 Virtual

Proceedings



Rafael Ramirez
Pompeu Fabra University, Spain

Darrell Conklin
University of the Basque Country UPV/EHU, Spain
IKERBASQUE, Basque Foundation for Science, Spain

José Manuel Iñesta
University of Alicante, Spain

Contents

- 1 Mel-spectrogram Analysis to Identify Patterns in Musical Gestures: a Deep Learning Approach**
David Dalmazzo, Rafael Ramirez
- 5 Deep Learning vs. Traditional MIR: a Case Study on Musical Instrument Playing Technique Detection**
Zehao Wang, Jingru Li, Xiaouu Chen, Zijin Li, Shicheng Zhang, Baoqiang Han, Deshun Yang
- 10 Adapted NMFD update procedure for removing double hits in drum mixture decompositions**
Len Vande Veire, Cedric De Boom, Tijl De Bie
- 15 Modeling Expressive Performance Deviations in Cello**
Tiange Zhu, Rafael Ramirez, Sergio Giraldo
- 20 audioLIME: Listenable Explanations Using Source Separation**
Verena Haunschmid, Ethan Manilow, Gerhard Widmer
- 24 The Impact of Label Noise on a Music Tagger**
Katharina Prinz, Arthur Flexer, Gerhard Widmer
- 28 Geometric Deep Learning for Music Genre Classification**
Manoranjan Sathyamurthy, Xiaowen Dong, M. Pawan Kumar
- 32 Improving Audio Onset Detection for String Instruments by Incorporating Visual Modality**
Grigoris Bastas, Aggelos Gkiokas, Vassilis Katsouros, Petros Maragos
- 36 Audio textures in terms of generative models**
Lonce Wyse, Muhammad Huzaifah
- 41 Evaluation of different symbolic encodings for music generation with LSTM networks**
Manos Plitsis, Kosmas Kritsis, Maximos Kaliakatsos-Papakostas, Vassilis Katsouros
- 46 Medley2K: A Dataset of Medley Transitions**
Lukas Faber, Sandro Luck, Damian Pascual, Andreas Roth, Gino Brunner, Roger Wattenhofer
- 51 A Machine Learning Approach to Cross-cultural Children's Songwriting Classification**
Rafael Ramirez, Kari Saarilahti
- 56 Two-step neural cross-domain experiments for full-page recognition of Mensural documents**
Francisco J. Castellanos, Jorge Calvo-Zaragoza, José M. Iñesta
- 60 Feature Engineering for Genre Characterization in Brazilian Music**
Bruna Wundervald
- 65 A dataset and classification model for Malay, Hindi, Tamil and Chinese music**
Fajilatun Nahar, Kat Agres, Balamurali BT, Dorien Herremans
- 69 Beat Tracking from Onset Streams Using LSTM Neural Networks**
Aggelos Gkiokas

Mel-spectrogram Analysis to Identify Patterns in Musical Gestures: a Deep Learning Approach

David Dalmazzo and Rafael Ramirez

Music and Machine Learning Lab
Music Technology Group
Department of Communication and Information Technology
Pompeu Fabra University,
Barcelona, Spain
david.cabrera@upf.edu, rafael.ramirez@upf.edu

Abstract. We present a machine learning approach to classify music gestures based on motion capture data. In particular, we record professional violinists while performing eight different bow-stroke techniques and apply deep learning to train classifiers to detect the type of bow-stroke performed. We compare three different convolutional neural networks (CNNs) architectures. Results show that the best architecture for the task is a hybrid CNN-LSTM architecture achieving more than 97% accuracy for the eight-class classification problem.

Keywords: Mel-spectrogram · Deep Learning · Musical gestures · Convolutional Neural Networks.

1 Introduction

Gestures understood as music performance techniques, have a direct consequence on sound qualities; by only analysing the sound we should be able to extract enough information to determine if a technical exercise was performed optimally. As an anatomical analogy, the human auditory cortex encodes spectrotemporal modulations (spectrograms), defined in the literature as ‘temporal fine-structure’, in the auditory nerve [11] which are then processed by a population of neurons located in the central auditory cortex (A1) in the superior temporal gyrus [10]. Although the auditory neural processing is still in the shadow, it is confirmed that a very precise system can discriminate temporal dimensions, location or timber characteristics and it is modulated by an attentional drive, which is formed by the contextual information framing its attentional-sound [12]. From this perspective, we intend to research about Mel-spectrogram application implementing deep learning techniques, to define sound ques, similar to an artificial attentional network to identify spoken-words, but in this case, to model gestural performance goals in musical exercises.

Mel-spectrogram based models have been used in the Music Information Retrieval (MIR). This field comprehends a variety of techniques and research topics. To name those that are more relevant to the paper, we can mention music feature extraction, music similarity and music classification (e.g. genre classification, auto-tagging) [3], [2]. In the field of music classification, we can find several studies focused on genre classification, where two techniques are commonly utilised to extract information from audio signals with the help of Convolutional Neural Networks (CNN): a) waveforms analysis, and b) spectrograms analysis. For instance, the *Shazam* system uses a constellation map as an audio fingerprint of coordinates in a 2D chart made of frequency against time, created from

spectrogram analysis of a database composed of 1.8M tracks. As a consequence, the system has produced a second database of ‘fingerprinted’ files mapped with the audio database, being capable of identifying, even with short segments of audio, the correspondent source [13]. Some common MIR techniques found in the literature are based on filtering adaptation to extract temporal features from spectrograms; in the same manner filters (square pooling) in the first neuronal layers are used in computer vision to extract shapes, new approaches are implemented with long y-axis and a short x-axis to extract temporal characteristics from the spectrogram images [9]. Another technique focusing using spectrograms consists of using convolutinal neural networks (CNNs) and convolutional concatenated filters to potentiate the extraction of temporal features from waveforms [15] [8], [1], [7], [4], [14], [6].

2 Materials and Methods

We recorded a database of bow-stroke classical violin techniques of three experts and three high-level students, while playing a G mayor scale (three octaves) covering the four strings of the violin. The eight bow-stroke techniques the performers were instructed to perform comprised: *Martelé*, *Staccato*, *Detaché*, *Ricochet*, *Legato*, *Trémolo*, *Collé* and *Col legno*. During the recording session, the audio was captured using a Zoom H5 recorder using to Max_8 application, recording WAV files with a sample rate of 44.100Hz/16bits. We have explored three convolutional neural network architectures: 1D CNN (Conv1D), 2D CNN (Conv2D) and a hybrid CNN and LSTM network (CNN_LSTM). To transform the audio signals to spectrograms we implemented the **Librosa**¹ 0.8.0 Python library [5]. **Librosa** is a package for music and audio analysis, suitable from rapid prototyping and implementation in the music information retrieval system.

3 Results

After training three Recurrent Neural Networks architectures using the recorded data for classifying the above bowing-techniques, we obtained the following classification accuracies:

- **Conv1D** Accuracy: 95.161% (sd +/-2.321)
- **Conv2D** Accuracy: 84.301% (sd +/-2.079)
- **CNN_LSTM** Accuracy: 97.473% (sd +/-2.012)

As shown in the tables 1, and 2, the highest accuracy is obtained with the CNN_LSTM. Nevertheless, the simple CNN_Conv1D architecture also produced high accuracy for the task. The CNN_LSTM architecture was tested with five filtering configurations (32,64,128,256,512) and the filter size of 512 reported the higher precision as shown in table 2.

4 Discussion

From the three CNN architectures explored the CNN_LSTM architecture obtained the highest classification accuracy using spectrograms for violin bow-stroke gestural detection. Nevertheless, the models proposed are not yet specialised to identify precise gestural executions as they are not mapped as an encoder-decoder supervised learning model. We have to take into account that for a

¹ web: <https://librosa.org/doc/latest/index.html>

Table 1: Classification Report Conv1D.CNN and Conv2D.CNN

class	precision	recall	f1-score	support	class	precision	recall	f1-score	support
0	1.00	0.96	0.98	24	0	0.96	0.96	0.96	24
1	0.92	1.00	0.96	24	1	0.61	0.92	0.73	24
2	0.84	0.88	0.86	24	2	0.85	0.92	0.88	24
3	1.00	0.89	0.94	36	3	1.00	0.61	0.76	36
4	1.00	1.00	1.00	20	4	0.95	0.90	0.92	20
5	0.80	1.00	0.89	12	5	0.55	1.00	0.71	12
6	1.00	0.97	0.98	30	6	0.82	0.60	0.69	30
7	1.00	1.00	1.00	16	7	0.87	0.81	0.84	16
accuracy			0.95	186	accuracy			0.81	186
macro avg	0.95	0.96	0.95	186	macro avg	0.82	0.84	0.81	186
weighted	0.96	0.95	0.95	186	weighted	0.85	0.81	0.81	186
avg					avg				

Table 2: Classification Report CNN_LSTM

class	precision	recall	f1-score	support
0	1.00	1.00	1.00	24
1	0.92	1.00	0.96	24
2	1.00	0.88	0.93	24
3	1.00	1.00	1.00	36
4	1.00	1.00	1.00	20
5	0.86	1.00	0.92	12
6	1.00	0.97	0.98	30
7	1.00	1.00	1.00	16
accuracy			0.98	186
macro avg	0.97	0.98	0.97	186
weighted avg	0.98	0.98	0.98	186

gesture-sound mapping, we still need much more data and also to develop further the architecture proposed in this paper.

We observed that in the case of the CNN_LSTM architecture with different filtering setups, some of the gestures were better recognised by a small filters and some others by bigger filters in the first layers of the CNN. For instance, with a filter of size 512 *Detaché* was poorly recognised, while using a filter of size 256 the same gesture is detected with high accuracy by the system. That means that it would be advisable to implement a two or three-headed CNN filtering with custom sizes to extract small timing characteristics and also extract timbral information that is very closely related to the nature of the gesture.

The Conv2D.CNN architecture results in a lower accuracy. It support the conclusions expressed in the literature that one-dimensional filters are better for mining temporal characteristics from time-sequence audio data, as 2D filters need a more precise custom definition in sizes to match timbral and temporal information. As future work, CNN architectures with custom 2D filters should be tested and evaluated.

We have focused on the implementation of CNN_LSTM, however, a more precise filtering system implementing a simple CNN with one-dimensional convolutional first layer, might be sufficient to constitute a robust architecture to achieve the goals of estimating correctness of bow-stroke gestural

executions. It also has to be noted that the Conv_1DCNN architecture is computationally less expensive to train and the resulting model is faster.

References

1. Chen, N., Wang, S.: High-level music descriptor extraction algorithm based on combination of multi-channel CNNs and LSTM. Proc. 18th Int. Soc. Music Inf. Retr. Conf. ISMIR 2017 pp. 509–514 (2017)
2. Choi, K., Fazekas, G., Sandler, M.: Automatic tagging using deep convolutional neural networks. Proc. 17th Int. Soc. Music Inf. Retr. Conf. ISMIR 2016 pp. 805–811 (2016)
3. Lee, C.H., Shih, J.L., Yu, K.M., Lin, H.S.: Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. IEEE Trans. Multimed. **11**(4), 670–682 (2009). <https://doi.org/10.1109/TMM.2009.2017635>
4. Lee, J., Park, J., Kim, K.L., Nam, J.: SampleCNN: End-to-end deep convolutional neural networks using very small filters for music classification. Appl. Sci. **8**(1) (2018). <https://doi.org/10.3390/APP8010150>
5. McFee, B., Raffel, C., Liang, D., Ellis, D.P., McVicar, M., Battenberg, E., Nieto, O.: librosa: Audio and music signal analysis in python. In: Proceedings of the 14th python in science conference. vol. 8, pp. 18–25 (2015)
6. Pons, J., Serra, X.: Randomly weighted cnns for (music) audio classification. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 336–340. IEEE (2019)
7. Pons, J., Slizovskaia, O., Gong, R., Gómez, E., Serra, X.: Timbre analysis of music audio signals with convolutional neural networks. 25th Eur. Signal Process. Conf. EUSIPCO 2017 **2017-Janua**, 2744–2748 (2017). <https://doi.org/10.23919/EUSIPCO.2017.8081710>
8. Pons, Jordi; Serra, X.: DESIGNING EFFICIENT ARCHITECTURES FOR MODELING TEMPORAL FEATURES WITH CONVOLUTIONAL NEURAL NETWORKS Jordi Pons and Xavier Serra Music Technology Group , Universitat Pompeu Fabra , Barcelona. IEEE Int. Conf. Acoust. Speech, Signal Process. 2017 pp. 2472–2476 (2017)
9. Schlüter, J., Böck, S.: Improved musical onset detection with convolutional neural networks. In: 2014 IEEE international conference on acoustics, speech and signal processing (icassp). pp. 6979–6983. IEEE (2014)
10. Shamma, S., Elhilali, M., Ma, L., Micheyl, C., Oxenham, A.J., Pressnitzer, D., Yin, P., Xu, Y.: Temporal coherence and the streaming of complex sounds. In: Basic Aspects of Hearing, pp. 535–543. Springer (2013)
11. Shamma, S.A.: Speech processing in the auditory system i: The representation of speech sounds in the responses of the auditory nerve. The Journal of the Acoustical Society of America **78**(5), 1612–1621 (1985)
12. Snyder, J.S., Alain, C., Picton, T.W.: Effects of attention on neuroelectric correlates of auditory stream segregation. Journal of cognitive neuroscience **18**(1), 1–13 (2006)
13. Tang, J., Liu, G., Guo, J.: Improved algorithms of music information retrieval based on audio fingerprint. 3rd Int. Symp. Intell. Inf. Technol. Appl. Work. IITAW 2009 pp. 367–371 (2009). <https://doi.org/10.1109/IITAW.2009.110>
14. Wu, Y., Mao, H., Yi, Z.: Audio classification using attention-augmented convolutional neural network. Knowledge-Based Syst. **161**(March), 90–100 (2018). <https://doi.org/10.1016/j.knosys.2018.07.033>
15. Zhu, Z., Enge, J.H., Hannun, A.: Learning multiscale features directly from waveforms. Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH **08-12-Sept**, 1305–1309 (2016). <https://doi.org/10.21437/Interspeech.2016-256>

Deep Learning vs. Traditional MIR: a Case Study on Musical Instrument Playing Technique Detection

Zehao Wang¹, Jingru Li¹, Xiaoou Chen¹, Zijin Li², Shicheng Zhang³, Baoqiang Han², and Deshun Yang¹

¹ Peking University, China ,

² China Conservatory of Music, China

³ University of Illinois at Urbana-Champaign, USA

{water45wzh, li_jingru, chenxiaoou, yangdeshun}@pku.edu.cn
zijin.li@mcgill.ca, sz18@illinois.edu, hundel@126.com

Abstract. Due to the variety of lengths and patterns of playing techniques and the labeled data available are few, musical instrument playing technique detection is challenging. In this work, we use two datasets of Chinese musical instruments, compare the performance of different audio features and neural network structures on this task. Besides, we make a comparison between DL-based model and DSP-based model. Fully Convolutional Network achieves the highest accuracy 89.5% on the test set. We additionally evaluate and visualize the performance of the proposed method on several real-world studio music (produced by midi) and real-world recording tracks.

1 Introduction

Playing technique is a crucial part of musical instruments performance. It conveys essential emotion and personal expression of both the composer and the performer in music signals. The detection of playing techniques in music recordings is beneficial to the research in automatic music transcription, music information retrieval, and performance analysis. For example, authors of [6] proposed a transcription system for electronic guitar, which transcribes both the notes and performance technique symbols.

However, the characteristic of playing techniques brings challenges to the detection task. In the case of Erhu, a famous Chinese bowed-stringed instrument, the playing techniques have the following characteristics: (1) *the duration of different techniques is on a wide range*; (2) *some techniques are similar to each other when listening*; (3) *some playing techniques are combinations of other techniques*.

Prior work formalized playing technique detection as a multi-stage classification task for segmented patterns of music recordings⁴[5,1,6], or a frame-level

⁴ first segmented, then classification

binary classification task for a given technique[8,7]. Liang *et al.*[3] considered playing techniques detection in the audio signal of musical performance as a particular aspect of automatic music transcription. They have done in-depth studies on piano sustain pedal detection and proposed valuable datasets and frameworks to examine the existence of pedal.

In this work, we mainly studied the feasibility and performance of Deep Learning models on this task. We made a comparison between some DL models (such as RNN⁵ and FCN, CNN⁶ and FCN[4]⁷) using different features (CQT and Mel-spectrogram) as input. We further compared the performance between the above DL-based models and DSP-based models[8,7] on this task. To evaluate our approach, we created a new open dataset containing all playing techniques of Erhu (ErhuPT)⁸ based on DCMI[2], then implement our models to the proposed ErhuPT dataset and an existing dedicated dataset CBF[8] of Chinese Bamboo Flute. The experiment results on the above two datasets showed that Mel+FCN model achieved the best performance, and deep learning based models performed better on long-term techniques than other existing signal processing methods.

2 Datasets

Two datasets are used in this paper. One is created by ourselves, named ErhuPT, the other is called CBF proposed by [8].

- **ErhuPT dataset:** This dataset is mainly about Erhu playing techniques. It contains two sets of recordings on 11 techniques by two different players, and several whole pieces of real-world music. Details can be found online.
- **CBF dataset:** The dataset proposed by [8] contains several playing techniques of Chinese Bamboo Flute. It contains two types of recordings by ten different players, named as isolated techniques and performed techniques(full piece), respectively. Detailed information can be found online.

The sample rate of all the audios in the above datasets is 44.1kHz.

For data pre-processing, firstly, we named the individual playing technique recordings as *short clips*. On both training and test set, we randomly generated audio segments of 10 seconds long by concatenating the short clips, and we named these generated audio segments as *long segments*. To ensure that the long segments sound realistic, we made a milliseconds cross-fade in each boundary of adjacent two short clips.

We also labeled the timestamp of playing techniques during the long segments generation process. The format of the label consists of event tags recorded with a frame length of 0.05 second. Due to the impossibility of playing multiple techniques simultaneously on a solo instrument, the above operation is reasonable.

⁵ Recurrent Neural Network

⁶ Convolutional Neural Network

⁷ Fully Convolutional Networks(FCN)

⁸ To be appeared online.

3 Experiments

Because the lengths of some techniques are short (only 0.15-0.20 seconds, 3-4 frames), we decided not to implement post-processing for the output prediction in our proposed model. We calculated the accuracy of one segment of these test data by hamming distance.

3.1 Different network architectures with different features

To figure out what kind of network architectures and audio features are suitable for this task, we used three famous architectures (RNN, CNN & FCN[4]) in Deep Learning and two audio features (CQT & Mel-spectrogram) to experiment. For brevity, detailed information about the experiments setting, network architectures and training strategies can be found online.

First, we trained the models on 2 subsets of ErhuPT, called *4 classes* (slide, staccato, trill & others) and *11 classes* (full dataset). In each experiment, we generated 10s long segments using ErhuPT dataset or a subset of it for training and test.

Comparison among different features: The results are shown in Table 1. We used FCN as the network architecture, and Mel, CQT & Mel+CQT⁹ as three types of input for the model. The result of *11 classes* experiments shows that Mel-spectrogram performs slightly better than other features. Existing research shows that Mel-filters can better adapt to the human auditory system, and CQT can better extract pitch/F0 information. Because the timbre aspect of playing techniques plays an important role, it is not difficult to understand this result.

Comparison among different architectures: We used Mel-spectrogram as the input, then implemented 4 classes experiments on three models and 11 classes on CNN and FCN. The result shows that FCN performs better than other models.

Further analysis and results on real-world music of the best model can be found online.

3.2 Comparison between DL-based method and DSP-based method

In this part, we mainly focus on the comparison between DL-based method and DSP-based method. We choose FCN+Mel trained on CBF dataset as the DL-based experiment, and experiments in [8,7] by the scattering transform approach as the DSP-based experiments.

These DSP-based experiments use scattering transform as the feature extractor, then implement a frame-level binary classification by Support Vector Machine (SVM) for each technique. The proposed DL-based experiment uses FCN to do a multi-classification directly. The DSP-based experiments train a specific SVM for each technique, respectively. The advantage of this method is that the models between different techniques do not interfere with each other.

⁹ They are in different channels of network input

	4 cl.	11 cl.
FCN+Mel	83%	59%
CNN+Mel	64%	34%
RNN+Mel	49%	—
FCN+CQT	—	57%
FCN+Mel+CQT	—	57%

Table 1. result of conditioned experiments

techniques	ST+SVMs			FCN+Mel		
	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
FT	97.8	99.5	98.7	88.1	92.4	90.2
Tremolo	67.6	41.4	50.7	95.2	88.4	91.7
Trill	89.8	76.3	82.3	90.6	98.0	94.1
Vibrato	75.1	64.7	69.3	92.1	82.9	87.2
Acciacatura	84.2	66.9	71.3	73.3	86.9	79.5
Portamento	70.0	51.1	58.6	83.9	86.7	85.3
Glissando	83.9	83.6	83.3	86.2	92.1	89.0

Table 5. Results of 2 models on CBFdataset. P: precision, R: recall, F: F-score.

Actual	Predicted										
	detache	diangong	harmonic	legato etc.	percussive	pizzicato	ricochet	staccato	tremolo	trill	vibrato
detache	76	0.13	0.33	15.9	0.09	0.27	0.24	0.32	1.14	5	0.61
diangong	0.72	67.8	5.41	15.9	0.18	0.33	1.31	2.54	1.08	2.74	1.96
harmonic	58.4	0.15	0.67	10.4	0.25	0.68	0.49	0.57	2.45	23.1	2.87
legato etc.	2.3	1.13	0.38	83.1	0.22	0.81	3.61	1.17	3.06	3.61	0.60
percussive	0.29	0.80	0.08	4.5	22.7	10.1	6.93	3.05	44.9	1.48	0.19
pizzicato	1.25	1.3	0	8.27	7.95	36.6	30.1	2.46	9.94	1.49	0.65
ricochet	0.45	1	0.14	7.14	6.07	6.83	5.5	10.5	10	1.13	0.51
staccato	1.4	2.07	0.12	6.88	8.13	4.78	31.3	33.8	8.47	2.07	0.97
tremolo	0.16	0.25	0.04	1.22	1.17	0.93	0.68	0.31	93.2	1.89	0.17
trill	1.12	1.27	0.54	16.6	0.27	0.85	0.85	1.29	28.7	48	0.59
vibrato	13.3	1.01	0.54	6.01	0.11	0.36	0.20	0.43	0.98	19.7	57.4

Table 3. Confusion matrix of FCN+Mel on ErhuPT

		Predicted						
		FT	Tremolo	Trill	Vibrato	Acciacatura	Portamento	Glissando
Actual	FT	92.4	0.8	0.7	2.1	2.2	1.6	0.2
	Tremolo	0.6	88.4	1.1	2.7	1.5	3.2	2.5
	Trill	0.1	0.2	98.0	0.0	0.5	1.1	0.2
	Vibrato	6.6	3.8	1.3	82.9	0.7	4.6	0.2
	Acciacatura	1.4	3.5	2.2	0.5	86.9	3.4	2.2
	Portamento	0.5	2.0	5.6	0.5	1.9	86.7	2.8
	Glissando	0.2	1.7	0.6	0.1	1.9	3.5	92.1

Table 4. Confusion matrix of FCN+Mel on CBF Dataset

Fig. 1. Due to the limitation of space, the enlarged version can be found online.

However, the model may predict more than one technique in a given frame, which is troublesome and ambiguous. Our multi-classification model FCN can deal with the last problem. Nevertheless, it will be affected by the long-tailed distribution, as we remarked above.

Table 5 shows the results of the comparison between the DL-based model and the DSP-based model. The results further indicate that FCN performs better on long-term and non-percussive techniques than the DSP-based model, but worse on other techniques. This observation inspires us to find a combination of these two approaches.

4 Conclusion

In this work, we focused on musical instrument playing technique detection. We analyzed the characteristics of different features and different models on two datasets of Erhu and Bamboo Flute for this task and made a comparison between DL-based model and DSP-based model. The best model achieves 89.50% accuracy on CBF dataset and 87.31% accuracy on ErhuPT dataset. Furthermore, we demonstrate the visualization of playing technique detection on real-world music, and the highest accuracy is 44.50%. Our experiments inspire further combining the different approaches.

References

1. Chen, Y.P., Su, L., Yang, Y.H., et al.: Electric guitar playing technique detection in real-world recording based on f0 sequence pattern recognition. In: ISMIR. pp. 708–714 (2015)

2. Li, Z., Liang, X., Liu, J., Li, W., Zhu, J., Han, B.: DCMI: A database of chinese musical instruments
3. Liang, B., Fazekas, G., Sandler, M.: Towards the detection of piano pedalling techniques from audio signal
4. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
5. Su, L., Yu, L.F., Yang, Y.H.: Sparse cepstral, phase codes for guitar playing technique classification. In: ISMIR. pp. 9–14 (2014)
6. Su, T.W., Chen, Y.P., Su, L., Yang, y.h.: Tent: Technique-embedded note tracking for real-world guitar solo recordings. Transactions of the International Society for Music Information Retrieval **2**, 15–28 (07 2019). <https://doi.org/10.5334/tismir.23>
7. Wang, C., Lostanlen, V., Benetos, E., Chew, E.: Playing technique recognition by joint timefrequency scattering. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 881–885 (2020)
8. hong Wang, C., Benetos, E., Lostanlen, V., Chew, E.: Adaptive time-frequency scattering for periodic modulation recognition in music signals. In: ISMIR (2019)

Adapted NMFD update procedure for removing double hits in drum mixture decompositions

Len Vande Veire¹, Cedric De Boom, and Tijl De Bie

Ghent University, Belgium
Contact: `len.vandevveire@ugent.be`

Abstract. Non-negative matrix factor deconvolution (NMFD) can be used to decompose a drum solo recording into K time-varying spectral templates (the constituent sounds) with corresponding activation functions. Unfortunately, choosing the template length, an important hyperparameter, is hard: it must be long enough to capture drum hits with a long decay, but when chosen too large, the algorithm often captures multiple drum hits within the same template. We propose to detect the emergence of such ‘double hits’ during optimization, and to replace them with an exponentially decaying extrapolation of the preceding template frames. Experiments demonstrate the effectiveness of this approach.

Keywords: Non-negative matrix factor deconvolution · Automated drum transcription

1 Introduction

The non-negative matrix factor deconvolution (NMFD) algorithm [8] decomposes a spectrogram matrix $X \in \mathbb{R}_{\geq 0}^{N \times T}$ with N frequency bins and T time frames into a dictionary of K time-varying spectral *templates* $W^{(k)} \in \mathbb{R}_{\geq 0}^{N \times L_\tau}$, and an *activation matrix* $H \in \mathbb{R}_{\geq 0}^{K \times T}$. The spectrogram is modeled as the convolution of the templates with the activation matrix:

$$X_{n,t} \approx \hat{X}_{n,t} = \sum_{k=1}^K \sum_{\tau=1}^{L_\tau} W_{n,\tau}^{(k)} H_{k,t-\tau} \quad (1)$$

where $H_{k,t-\tau}$ is zero when $t < \tau$. $W^{(k)}$ and H are updated iteratively using multiplicative updates in order to minimize a divergence measure $\mathcal{L}(X, \hat{X})$. In this paper, we use the KL divergence, \mathcal{L}_{KL} , and the corresponding update rules for $W^{(k)}$ and H [7]:

$$\mathcal{L}_{KL}(X, \hat{X}) = \sum_{n,t} X_{n,t} \log \frac{X_{n,t}}{\hat{X}_{n,t}} - X_{n,t} + \hat{X}_{n,t}, \quad (2)$$

$$W_{n,\tau}^{(k)} \leftarrow W_{n,\tau}^{(k)} \frac{\sum_t H_{k,t-\tau} (X_{n,t} / \hat{X}_{n,t})}{\sum_t H_{k,t-\tau}}, \quad (3)$$

$$H_{k,t} \leftarrow H_{k,t} \frac{\sum_\tau \sum_n W_{n,\tau}^{(k)} (X_{n,t+\tau} / \hat{X}_{n,t+\tau})}{\sum_\tau \sum_n W_{n,\tau}^{(k)}}. \quad (4)$$

The templates $W^{(k)}$ can be interpreted as short spectrograms of length L_τ that model the constituent sounds of the mixture. Ideally, each $W^{(k)}$ would capture an individual drum hit of a particular instrument, e.g. $W^{(0)}$ captures a single kick drum hit, $W^{(1)}$ captures a single snare drum hit and so on. The corresponding activations H_k then describe where in the mixture these sounds occur. NMFD has already been applied successfully for automated drum transcription and drum separation tasks [1,2,4,5,6,9]. These works only consider constrained settings, though, e.g. only optimizing for H and keeping the dictionary W fixed. We note the absence in literature of a successful application of NMFD where both W and H are optimized jointly.

The template length L_τ is an important hyper-parameter in NMFD. Percussive mixtures often contain some instrument(s) with a long decay, e.g. a kick drum; therefore, L_τ needs to be large enough to adequately capture a single drum hit of these instruments. However, percussive mixtures also often contain hits that follow each other in rapid succession, e.g. the hi-hats. In this case, NMFD often captures multiple drum hits within one template, as has been noted before in the context of drum mixture decomposition using NMFD [5]. This is problematic: the discovered templates then no longer contain single drum hits, or they can even contain drum hits of multiple instruments, so that the resulting activations no longer reflect the onsets of the individual instruments, making the decomposition less interpretable and useful. Figure 1(b) illustrates this problem.

2 Detecting emerging double hits during optimization

We propose to solve the ‘double-hit’ problem by checking after each update of $W^{(k)}$ whether a second onset can be detected in the template. If this is the case, then $W^{(k)}$ is modified by overwriting this second onset with an exponentially decaying extension of the preceding template frames. This will initially lead to a worse approximation of the spectrogram, as important information for the decomposition was removed. However, the expected effect of this modification is that, in the next update of the activations H , some activation value(s) will increase to compensate for the removal of the secondary onset in the template; eventually, after a few updates, each $W^{(k)}$ will ideally only contain a single drum hit, and all onsets will be captured in H_k .

The adapted update procedure for $W^{(k)}$ is as follows:

1. Calculate the updated version of $W^{(k)}$, as in Eqn. (3).
2. Calculate the log-envelope $a^{(k)}[\tau]$ of each updated template $W^{(k)}$:

$$\tilde{a}^{(k)}[\tau] = \sum_n \log \left(W_{n,\tau}^{(k)} + \epsilon \right), \quad (5)$$

$$a^{(k)}[\tau] = \tilde{a}^{(k)}[\tau] - \min_\tau \left(\tilde{a}^{(k)}[\tau] \right). \quad (6)$$

3. Calculate $\Delta a^{(k)}[\tau] = a^{(k)}[\tau + \tau_u] - a^{(k)}[\tau]$. When $\Delta a^{(k)}[\tau]$ is large for some τ , then there is an onset at time τ in the template.

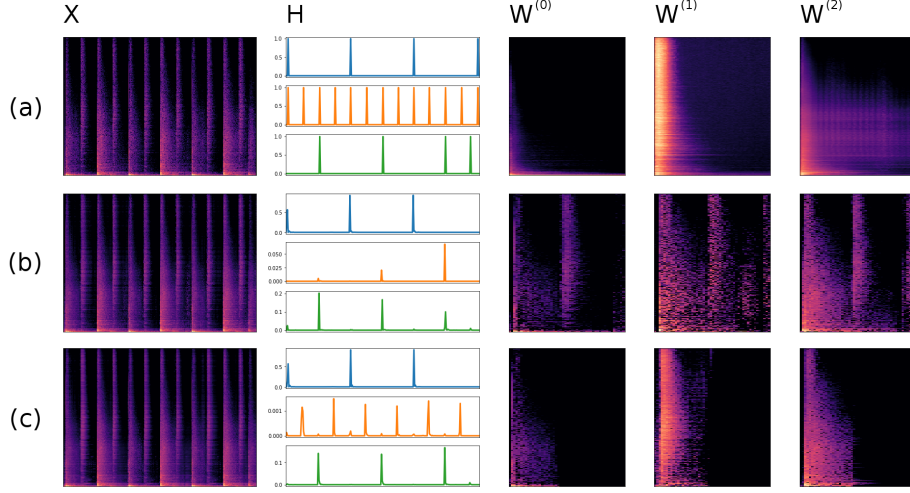


Fig. 1. Illustration of the decomposition of a short drum loop: (a) ground-truth decomposition; (b) decomposition with NMFD; (c) decomposition with the modified NMFD algorithm. Columns: X , the spectrogram; H , the activations; $W^{(0)}$, the first template, capturing the kick drum; $W^{(1)}$, capturing the hi-hats; $W^{(2)}$, capturing the snare drum.

4. Set $a_{\max}^{(k)} = \max(a^{(k)}[\tau])$. Detect onsets in $W^{(k)}$ by determining whether there is an onset larger than some threshold θ_{thr} , $\Delta a^{(k)}[\tau] \geq (\theta_{\text{thr}} a_{\max}^{(k)})$, for some $\tau \geq \tau_{\text{thr}}$. Only peaks that lie past the shift threshold τ_{thr} are considered, in order to not erroneously correct the first (and correct) hit in the template.
5. If there is a second onset in the template at $\tau_{\text{err}} \geq \tau_{\text{thr}}$, then all the frames after this onset are replaced by an exponentially decaying extension of the template frames preceding it:

$$W_{n,\tau}^{(k)} \leftarrow W_{n,\tau_{\text{err}}-\tau_u}^{(k)} \exp(-\gamma(\tau - \tau_{\text{err}})), \tau = \tau_{\text{err}} \dots L_\tau. \quad (7)$$

In our experiments, we use the following settings for the hyper-parameters of this procedure: $\tau_u = 3$, $\theta_{\text{thr}} = 0.05$, $\tau_{\text{thr}} = 10$, $\gamma = 1$, $L_\tau = 50$, $\epsilon = 10^{-18}$, which were empirically found to lead to good results. The STFT spectrogram is calculated with a hop size of 512, and the audio sampling rate is 44.1 kHz.

3 Case study: decomposing a drum loop

As an example, we consider the drum loop in Figure 1(a)¹. It contains three instruments: a kick drum, a snare drum and a hi-hat. The kick drum decays over approximately 50 frames; hence, we set $L_\tau = 50$. We note, however, that the hi-hats occur in rapid succession, i.e. approximately every 25 frames.

¹ This drum loop is a 4 second extract of a solo drum recording from the ENST dataset [3], “062_phrase_rock_simple_medium_sticks.wav”.

When decomposed with the original NMFD algorithm, shown in Figure 1(b), the templates $W^{(k)}$ capture not the individual drum hits, but rather repeating *sub-sequences* of drum hits. The activations consequently are very sparse and are not informative to determine the onset locations of the individual instruments.

When decomposed with NMFD using the proposed modifications, the templates each capture only a single drum hit, as shown in Figure 1(c). Note that the extracted templates very much resemble their ground-truth counterpart, see Figure 1(a). The activations also match the ground-truth onsets quite well; for the hi-hat, i.e. the second component, there is some discrepancy, as only every other onset is clearly captured. The other activations are ‘absorbed’ into the kick drum and snare drum components. This is a consequence of the fact that NMFD cannot distinguish a single-instrument hit from such a consistent layering of multiple instantaneous drum hits (i.e. in this example, each kick/snare drum hit always coincides with a hi-hat hit); an additional mechanism to disentangle such sounds is beyond the scope of this paper.

4 Evaluation on the ENST dataset

We evaluate our approach on all *fast simple* phrases from the ENST dataset [3]. We run the original NMFD algorithm and our adaptation on these extracts, and quantify how many excess drum hits can be detected in each template by counting the number of peaks in $\Delta a^{(k)}[\tau]$, see Section 2. We furthermore measure the spectrogram reconstruction quality using the Mean Absolute Error between X and \hat{X} : $\text{MAE}(X, \hat{X}) = \frac{1}{NT} \sum_{n,t} |X_{n,t} - \hat{X}_{n,t}|$.

For each decomposed mixture, the MAE for the decomposition with the original algorithm and the MAE for the adapted version are nearly identical; furthermore, all spectrograms are approximated well (mean MAE $5.6 \cdot 10^{-5}$ for both the original and the adapted algorithm, stdev. $3.0 \cdot 10^{-5}$ and $3.1 \cdot 10^{-5}$ resp.). The average number of excess peaks detected in $\Delta a^{(k)}[\tau]$ is 2.2 (stdev. 1.0) for default NMFD, and 0 for the adapted procedure². Visual inspection³ of the results shows that in the decompositions with unmodified NMFD, double hits are often present, while these are removed with the proposed procedure.

5 Conclusion

We conclude that the proposed adaptation maintains the same spectrogram reconstruction quality, with the added advantage that NMFD now captures only one drum hit per template. This allows to choose the template length long enough to fully capture drum hits with a long decay, while maintaining a clear and interpretable decomposition even in the presence of rapid successive drum hits.

² Which is an expected result, of course, as we report on the metric that is used in the adapted algorithm to detect double hits in the templates.

³ See the accompanying website for examples: <https://users.ugent.be/~levdvveir/2020MML>

Acknowledgements

Len Vande Veire is supported by a PhD fellowship of the Research Foundation Flanders (FWO).

References

1. Dittmar, C., Müller, M.: Towards transient restoration in score-informed audio decomposition. In: Proc. Int. Conf. Digital Audio Effects. pp. 145–152 (2015)
2. Dittmar, C., Müller, M.: Reverse engineering the amen break: score-informed separation and restoration applied to drum recordings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **24**(9), 1535–1547 (2016)
3. Gillet, O., Richard, G.: Enst-drums: an extensive audio-visual database for drum signals processing. In: Proc of 7th International Conference on Music Information Retrieval, ISMIR 2006 (2006)
4. Laroche, C., Papadopoulos, H., Kowalski, M., Richard, G.: Drum extraction in single channel audio signals using multi-layer non negative matrix factor deconvolution. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 46–50. IEEE (2017)
5. Lindsay-Smith, H., McDonald, S., Sandler, M.: Drumkit transcription via convolutive nmf. In: International Conference on Digital Audio Effects (DAFx-12), York, UK (2012)
6. Roebel, A., Pons, J., Liuni, M., Lagrangey, M.: On automatic drum transcription using non-negative matrix deconvolution and itakura saito divergence. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 414–418. IEEE (2015)
7. Schmidt, M.N., Mørup, M.: Nonnegative matrix factor 2-d deconvolution for blind single channel source separation. In: International Conference on Independent Component Analysis and Signal Separation. pp. 700–707. Springer (2006)
8. Smaragdis, P.: Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In: Int. Conf. on Independent Component Analysis and Signal Separation. pp. 494–499. Springer (2004)
9. Ueda, S., Shibata, K., Wada, Y., Nishikimi, R., Nakamura, E., Yoshii, K.: Bayesian drum transcription based on nonnegative matrix factor decomposition with a deep score prior. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 456–460. IEEE (2019)

Modeling Expressive Performance Deviations in Cello

Tiange Zhu, Rafael Ramirez, and Sergio Giraldo

¹ Music and Machine Learning Lab

² Music Technology Group

³ Department of Communication and Information Technology

Pompeu Fabra University,

Barcelona, Spain

tiangezhu@outlook.com, rafael.ramirez@upf.edu, sergio.giraldo@upf.edu

Abstract. This paper presents a machine learning approach for modeling expressive cello performances by a number of famous musicians. One hundred and eight recordings of Bach's Suite No.1 and corresponding music scores were collected. Recordings were automatically aligned using Dynamic Time Warping algorithm based on chroma features. Several machine learning algorithms were applied to model the timing and dynamics expressive deviations introduced by the musicians. Performance of applied algorithms are discussed; results show that, among the algorithms considered, Random Forest produces the most accurate model for both timing and dynamics expressive deviations.

Keywords: Expressive Performance · automatic transcription · Machine Learning

1 Introduction

Expressive music performance refers to the deviations a professional musician introduces when playing a score, such as timing, dynamics, tempo and articulation alterations, in order to communicate a notated piece. Such musical variations are described as performance actions. The research of expressive music performance aims to identify and quantify key aspects of performance actions that shape expressiveness introduced in musical performance.

Unlike many other music genres, western classical musicians perform a piece strictly according to its original composition without changing any melody content. Here we model variations not in melody but in timing and dynamics using data-driven approaches.

Keyboard instruments such as piano have been the most studied instrument in the field of expressive music performance [1][2][3], partially due to their frequent appearance along the classical music history and high availability of large datasets for relevant research. There are also a number of works focused on expressive performance of wind instruments. For instance, Ramirez et al. [5] presented a genetic rule-based model for jazz saxophone, and Barthet et al. studied

timbre, timing and dynamics in clarinet performance [6] [7]. String instruments have also received some attention over the past years. Giraldo et al. [8] presented a convincing work modeling ornamentation in jazz guitar, and Ortega et al. highlighted Phrase-Level modeling in violin performance [9].

To the best of our knowledge, there are only a few works focused on cello. Igarashi et al. [11] extracted rules of respiration using sensors to measure respiration in cello performance. Hong [10] worked on re-examination of the motor process between expressive timing and dynamics proposed by Todd [4]. However, none of them focused on the topic of expressive performance modeling. Thus, this work is motivated to model the deviation in expressive performances of classical music compositions for cello, particularly focuses on the timing and dynamics aspects.

2 Methodology

A dataset containing 108 recordings of Bach’s Suite No.1 in G Major (BWV 1007) was collected, each of them performed by a different cellist, in WAV format. The symbolic notation of Bach’s Suite No.1 in G Major is obtained in MusicXML format.

Chromagram features were extracted from the STFT spectrogram of the recordings, using Librosa library[12]. The Dynamic Time Warping algorithm was applied to find the optimal path by calculating the Euclidean distance according to obtained chromagrams.

A note segmentation algorithm was developed to segment notes according to detected chroma changes. Small peaks were avoided by setting the minimum threshold of note length to the length of a demisemiquaver.

A set of musical informative descriptors were extracted to depict the characteristics of musical notes, which are presented in Table 1.

Table 1. Neighbour descriptors for individual musical notes

Descriptor	Units	Range
Pitch	Semitones	[1, 127]
Chroma	Semitones	[0, 11]
Onset	Frames	[0, $+\infty$]
Previous onset	Frames	[0, $+\infty$]
Next onset	Frames	[0, $+\infty$]
Previous inter-onset distance	Frames	[0, $+\infty$]
Next inter-onset distance	Frames	[0, $+\infty$]
Previous pitch interval	Semitones	[-60, 60]
Next pitch interval	Semitones	[-60, 60]
Measure	Bars	[0, $+\infty$]
Beat	Beat	[1, 4]

Performance actions were calculated for each note in the recordings. The inter-onset interval(IOI) ratio was measured as deviations in timing, which rep-

resents the ratio of inter-onset interval of a note to the inter-onset interval of its corresponding note in the original notation. In the context of this paper, the dynamics of a piece is defined as the variation in loudness between musical notes. Thus, the root-mean-squared energy in cello performance was measured.

The IOI ratio and RMS energy were further categorized into 3 classes. When the IOI of note i in performance is less than 90% of the IOI of note i in its notation, it is recognised as a shorter note. When the IOI ratio exceeds 110%, the note is recognised as a longer note. Otherwise, it is considered as approximately equal.

The averages of RMS energy within all performances were calculated individually. If the RMS energy of note i is less than 80% of the average energy of the piece, it is defined as a softer note. If it exceeds 120% of the average energy, it is considered as a louder note. Otherwise, it falls into the category of approximately equal.

A classification experiment was conducted, using Support Vector Machine(with a linear kernel), K-nearest Neighbour($k=1$), and Random Forest algorithms. Zero Rule classifier was also applied as baseline to compare with the other algorithms.

A selection of classification models trained by the best performed machine learning algorithm were used to compare with other performances, in order to obtain information on possible correlations between artists from the classification results.

3 Results

Table 2. Classification results

Algorithm	CCI (RMS Energy)	CCI (IOI Ratio)
Zero Rule	42.21%	51.83%
Random Forest	60.30%	57.25%
Support Vector Machine	48.23%	53.29%
k-Nearest Neighbours	51.77%	51.51%

Table 2 presents the arithmetic mean of the Corrected Classified Instances results from the classification of 84 aligned performances, predicting the categories of RMS energy and IOI ratio. The obtained models were evaluated with 10-fold cross validation.

Figure 1 and Figure 2 presented correlation matrices on RMS energy and IOI ratio between 8 selected cello performances, based on the classification results using Random Forest algorithm. The correlation coefficients are in the range of $[-100, 100]$. A correlation coefficient between musician A and musician B presents the difference between the result of training Random Forest model of performance A to predict the classes in performance B and the baseline prediction result of performance A. The negative coefficients were adjusted to 0, suggesting that no potential correlation was found between two artists.

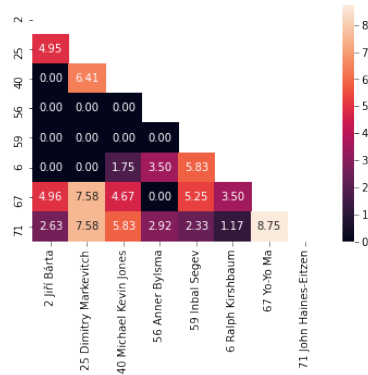


Fig. 1. Correlation Matrix on IOI ratio

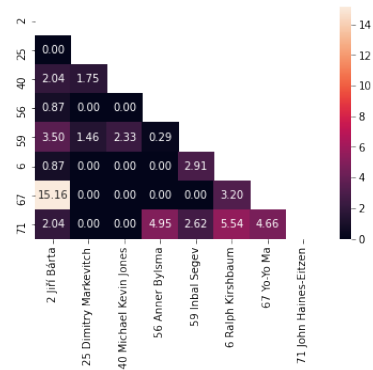


Fig. 2. Correlation Matrix on RMS energy

4 Discussion

A high correlation coefficient between two artists shown in Figure 1 or Figure 2 may suggest that similar techniques were adopted when modulating timing or dynamics in their performances. For instance, the author could conjecture that there is a strong connection between the performance of Yo-Yo Ma and Dmitry Markevitch, on both timing and dynamics aspects. On the contrary, Anner Bylsma’s performance shows little connections with Dmitry Markevitch’s performance. However, such speculation derived from computational results may not be true in the musical perspective, it would require experts such as musicologists to examine.

Table 2 shows that Random Forest, Support Vector Machine and k-Nearest Neighbours all achieved better results than the baseline on the task of dynamics prediction, indicating that all models were able to predict categories of RMS energy in cello performances. As for the results from the prediction on timing, it could be observed that Random Forest reached the highest CCI results, and k-Nearest Neighbours(k=1) failed to give meaningful predictions on IOI ratio categories.

Generally, all models showed better capability on the prediction of dynamics than timing categorized labels. The best performed algorithm was Random Forest, its Correctly Classified Instances reached 60.30% and 57.25% in classifying timing and dynamics, which were comparably better than the baseline performances, showing a strong capability on relevant tasks.

The Support Vector Machine with linear kernel was applied in previous experiments. Such a model could be too simplistic in terms of capturing the expressiveness in cello performances, since it was not a linear problem. It could be interesting to see the results from the experiments using SVM with higher-degree polynomial kernels which allow more flexible decision boundaries, to discover if the changes in settings could improve the prediction accuracy.

As for the undesired performance of k-Nearest Neighbours($k=1$), one of the main reasons could be overfitting. Setting k to a higher number when using k-NN could possibly perform better on modeling the timing and dynamics in expressive performances.

The proposed system may also be improved by adding a feature selection process, since it is still not clear which features are more responsible for generating such results. Filtering out the irrelevant features could reduce the potential of overfitting.

Lastly, a larger dataset would be desirable, especially when the paper tries to address its research questions with machine learning approaches. It is also worth noting that the training data itself was not completely reliable. The audio-symbolic alignment of cello performances was automatically done by the algorithm, and the evaluation method was manually checked on around 10% of the data combined with rough estimates based on plots. It is very likely that there exists cases in the training data that some notes are wrongly aligned. Hence, a systematic and detailed evaluation of the alignment could lead to a better accuracy of modeling.

References

1. Jeong, D., Kwon, T., Kim, Y., Lee, K. & Nam, J. Virtuosonet: A hierarchical rnn-based system for modeling expressive piano performance. In Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR, Delft, The Netherlands, 908–915 (Nov. 4, 2019).
2. Widmer, G. & Tobudic, A. Playing mozart by analogy: Learning multi-level timing and dynamics strategies. *Journal of New Music Research* 32 (3) 259–268 (2003).
3. Oore, S., Simon, I., Dieleman, S., Eck, D. & Simonyan, K. This time with feeling: learning expressive musical performance. *Neural Computing and Applications* 32, 955–967 (2018).
4. Todd, M. & P., N. The dynamics of dynamics: A model of musical expression. *The Journal of the Acoustical Society of America* 91, 3540–3550 (1992).
5. Ramirez, R., Hazan, A., Maestre, E. & Serra, X. A genetic rule-based model of expressive performance for jazz saxophone. *Computer Music Journal* 32, 38–50(2008).
6. Barthelet, M., Depalle, P., Kronland-Martinet, R. & Ystad, S. Acoustical correlates of timbre and expressiveness in clarinet performance. *Music Perception: An Interdisciplinary Journal*. 28. 10.1525/mp.2010.28.2.135. (2010).
7. Barthelet, M., Depalle, P., Kronland-Martinet, R. & Sølvi, Y. Analysis-by synthesis of timbre, timing, and dynamics in expressive clarinet performance. *Music Perception* 28 (2011).
8. Giraldo, S. & Ramirez, R. A machine learning approach to ornamentation modeling and synthesis in jazz guitar. *Journal of Mathematics and Music* 10, 107–126 (2016).
9. Ortega, F., Giraldo, S., Perez, A. & Ramírez, R. Phrase-level modeling of expression in violin performances. *Frontiers in Psychology* 10: 776 (Apr.11, 2019).
10. Igarashi, S., Ozaki, T. & Furukawa, K. Respiration reflecting musical expression: Analysis of respiration during musical performance by inductive logic programming. In *Music and Artificial Intelligence*, 94–106 (Springer Berlin Heidelberg, 2002).
11. Hong, J.-L. Investigating expressive timing and dynamics in recorded cello performances. *Psychology of Music* 31, 340–352 (2003).
12. McFee, B. et al. Librosa 0.5.0 (2017). URL <https://doi.org/10.5281/zenodo.293021>.

audioLIME: Listenable Explanations Using Source Separation

Verena Haunschmid¹, Ethan Manilow³, and Gerhard Widmer^{1,2}

¹ Institute of Computational Perception, Johannes Kepler University, Linz, Austria
verena.haunschmid@jku.at^[0000-0001-5466-7829]

² LIT Artificial Intelligence Lab, Johannes Kepler University, Linz, Austria

³ Interactive Audio Lab, Northwestern University, Evanston, IL, USA

Abstract. Deep neural networks (DNNs) are successfully applied in a wide variety of music information retrieval (MIR) tasks but their predictions are usually not interpretable. We propose *audioLIME*, a method based on Local Interpretable Model-agnostic Explanations (LIME), extended by a musical definition of locality. The perturbations used in LIME are created by switching on/off components extracted by source separation which makes our explanations listenable. We validate audioLIME on two different music tagging systems and show that it produces sensible explanations in situations where a competing method cannot.

1 Introduction

Deep neural networks (DNNs) are used in a wide variety of music information retrieval (MIR) tasks. While they generally achieve great results according to standard metrics, it is hard to interpret how or why they determine their output. This can lead to situations where a network does not learn what its designers intend. One goal of the field of interpretable machine learning is to provide tools for practitioners that push towards making the decisions of opaque models understandable. The field of MIR has many stakeholders—from individual musicians to entire corporations—all of which must be able to trust DNN systems.

A promising approach to this problem is Local Interpretable Model-agnostic Explanations (LIME) [6], which produces explanations of predictions from an arbitrary model *post-hoc* by perturbing interpretable components around an input example and fitting a small, surrogate model to explain the original model’s prediction. Previous attempts at adopting LIME for MIR tasks have used rectangular regions of a spectrogram for explanations [5]. This ignores two defining characteristics of audio data: 1) the lack of occlusion of overlapping sounds and, 2) all parts of a single sound might *not* be contiguous on a spectrogram.

In this work, we introduce *audioLIME*, an extension of LIME that preserves fundamental aspects of audio so explanations are *listenable*. To achieve this we propose a new notion of “locality” based on estimates from source separation algorithms. We evaluate our method on music tagging systems by feeding the explanation back into the tagger and seeing if the prediction changes. Using this technique, we show that our method is able to explain predictions from a waveform-based music tagger, which previous methods cannot do. We also provide illustrative examples of listenable explanations from our system.

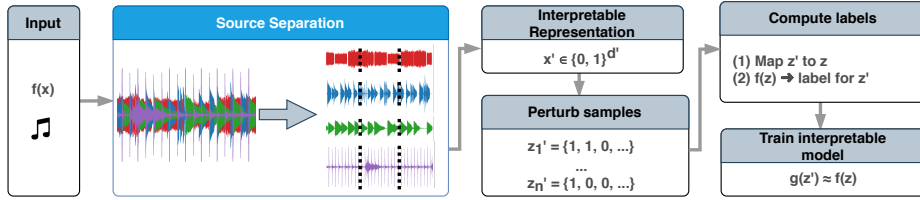


Fig. 1: audioLIME closely follows the general LIME pipeline. The key is the use of source estimates (blue box). A source separation algorithm decomposes input audio into $d' = C \times \tau$ interpretable components (C sources, τ time segments).

2 audioLIME

audioLIME is based on the LIME [6] framework and extends its definition of locality for musical data by defining a new way of deriving an interpretable representation. For an input value x to an arbitrary, black-box model f , LIME first defines a set of d' interpretable features that can be turned on and off, $x' \in \{0, 1\}^{d'}$. These features are perturbed and represented as a set of binary vectors z'_n that an interpretable, surrogate model trains on. This surrogate model matches the performance of the black-box model around x and is able to reveal which of its interpretable features the black-box model relies on.

The key insight of audioLIME is that *interpretability* with respect to audio data should really mean *listenability*. Whereas previous approaches applied techniques from the task of image segmentation to spectrograms, we propose using *source separation estimates* as interpretable representations. This gives audioLIME the ability to train on interpretable *and* listenable features.⁴

The single-channel source separation problem is formulated as estimating a set of C sources, S_1, \dots, S_c , when only given access to the mixture M from which the sources are constituents. We note that this definition, as well as audioLIME, is agnostic to the input representation (e.g., waveform, spectrogram, etc) of the audio. We use these C estimated sources of an input audio as our interpretable components (e.g. $\{\text{piano}, \text{drums}, \text{vocals}, \text{bass}\}$). Mapping $z' \in \{0, 1\}^C$ to z (the input audio) is performed by mixing all present sources. For example $z' = \{0, 1, 0, 1\}$ results in a mixture only containing estimates of drums and bass. The relation of this approach to the notion of *locality* as used in LIME lies in the fact that samples perturbed in this way will in general still be perceptually similar (i.e., recognized by a human as referring to the same audio piece). This system is shown in Figure 1. In addition to source separation, we also segment the audio into τ temporal segments, resulting in $C \times \tau$ interpretable components.

⁴ Python package available at: <https://github.com/CPJKU/audioLIME>

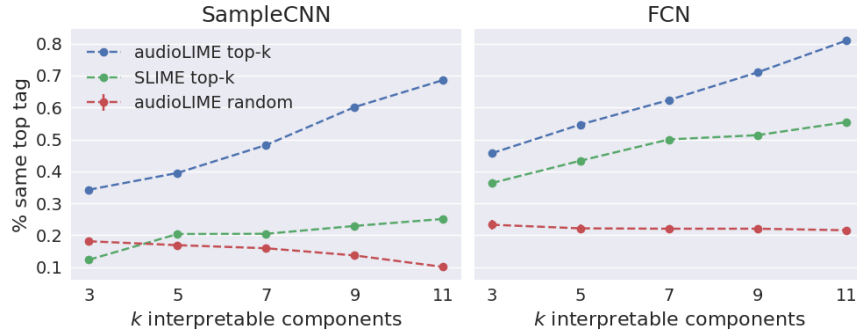


Fig. 2: Percentage of explanations that produced the same tag as the original input using k interpretable components for two music tagging systems. audioLIME (blue) produces better explanations than SLIME [5] (green) and the baseline (red).

3 Experiments

We analyze two music tagging models [7]: FCN [2], which inputs a 29 second spectrogram, and SampleCNN [4], which inputs a 3.69 second waveform. Both models were trained on the MillionSongDatatset (MSD) [1]. The LIME explanation model used is a linear regression model trained with l2 regularization on 2^{14} samples. We use Spleeter [3] as the source separation system.

Quantitative Results To verify that the explanations truly explain the model’s behaviour we perform a simple experiment. If the explanation explains the model’s behaviour we expect the tagger to be able to make the same prediction when only passing the top k selected components, and a different prediction otherwise.⁵

We randomly picked 100 examples from the MSD test set, 20 for each of the 5 most common tags (rock, pop, alternative, indie, electronic). For each example we create several explanations (3/song for FCN, 16/song for SampleCNN) for the top predicted tag. We compare two explanation systems, using the the top k components in each explanation from either audioLIME or SLIME [5]. As a baseline, we compare the prediction each tagger makes on k randomly selected components where audioLIME surrogate models have a positive linear weight.

Figure 2 shows that even when using only a fraction of the components, the tagger makes the same prediction more often with audioLIME than with SLIME or the baseline. Importantly, because audioLIME’s explanations emphasize listenability, they are invariant to the input audio representation of the model, and thus it is able to provide better explanations than SLIME, which does not have the same flexibility. This indicates there is a whole class of waveform-based models that SLIME is unsuited for, but audioLIME still works well.

⁵ Experiment code: <https://github.com/expectopatronum/mml2020-experiments/>

Qualitative Results Because the explanations audioLIME makes are source estimates, it is possible to listen to and make sense of them. To illustrate this, we selected two examples of explanations of a prediction made by FCN.⁶ In the first example, FCN predicted the tag “female vocalist” and, indeed, the top 3 selected audioLIME components are the separated vocals with a female singer. In the second case, FCN predicted the tag “rock”, and in the top audioLIME components we can hear a driving drumset and a distorted guitar, both of which are associated with rock music. In these cases, we can be confident that our music tagging network has learned the correct concepts for these tags, and thus increases our trust in the black-box FCN model.

4 Conclusion

In this work we presented audioLIME, a system that uses source separation to produce *listenable* explanations. We demonstrated an experiment that showed how audioLIME can produce explanations that create trustworthy predictions from music tagging systems that use waveforms or spectrograms as input. We also showed two illustrative examples of explanations from audioLIME. One of the shortcomings of audioLIME is its dependency on a source separation system, which only works with a limited number of source types and may introduce artifacts. However, we note that audioLIME is agnostic to the source separation system, and thus audioLIME is compatible with future work in that space.

References

1. Bertin-Mahieux, T., Ellis, D.P.W., Whitman, B., Lamere, P.: The Million Song Dataset. In: Proc. of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA (2011)
2. Choi, K., Fazekas, G., Sandler, M.B.: Automatic tagging using deep convolutional neural networks. In: Mandel, M.I., Devaney, J., Turnbull, D., Tzanetakis, G. (eds.) Proc. of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States (2016)
3. Hennequin, R., Khelif, A., Voituret, F., Moussallam, M.: Spleeter: A Fast And State-of-the Art Music Source Separation Tool With Pre-trained Models. Late-Breaking/Demo ISMIR 2019 (November 2019), deezer Research
4. Lee, J., Park, J., Kim, K.L., Nam, J.: Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. CoRR **abs/1703.01789** (2017)
5. Mishra, S., Sturm, B.L., Dixon, S.: Local Interpretable Model-Agnostic Explanations for Music Content Analysis. In: Proc. of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, 2017 (2017)
6. Ribeiro, M.T., Singh, S., Guestrin, C.: “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In: Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA (2016)
7. Won, M., Ferraro, A., Bogdanov, D., Serra, X.: Evaluation of CNN-based Automatic Music Tagging Models. In: Proc. of 17th Sound and Music Computing (2020)

⁶ <https://soundcloud.com/veroamilbe/sets/mml2020-explanation-example>

The Impact of Label Noise on a Music Tagger^{*}

Katharina Prinz, Arthur Flexer, and Gerhard Widmer

Institute of Computational Perception, Johannes Kepler University Linz, Austria
{katharina.prinz, arthur.flexer, gerhard.widmer}@jku.at

Abstract. We explore how much can be learned from noisy labels in audio music tagging. Our experiments show that carefully annotated labels result in highest figures of merit, but even high amounts of noisy labels contain enough information for successful learning. Artificial corruption of curated data allows us to quantize this contribution of noisy labels.

Keywords: music tagging · label noise · convolutional neural networks

1 Introduction

The necessity of annotated data for supervised learning often contrasts with the cost of obtaining reliable ground-truth in a manual fashion. Automated methods, on the other hand, simplify the annotation process and result in greater quantities of data with possibly noisy labels, which is why multiple approaches that investigate learning from noisy labels have been proposed [4–7, 11, 12]. In the DCASE2019 Challenge “Audio tagging with noisy labels and minimal supervision”, the question was raised whether noisy labels can also be useful when training a system to perform audio tagging [1]. This work builds upon a submission to the DCASE2019 Challenge [8] and prior work [9], but with a focus particularly on audio samples with musical labels. We investigate the impact of training a music tagger solely on noisy labels, and find that even for a potentially high level of noise we clearly outperform a random baseline at a respectable performance level.

2 Data

We use the musical subset of the data provided for Task 2 of the DCASE2019 Challenge [1]. More precisely, out of the 80 classes in the original setup, our experiments are restricted to samples with one or several out of 12 different classes, containing musical instruments and male/female singing. In total, 3,967 training audio clips remain of which roughly 20% (825) are curated, and 80% (3,142) have noisy labels. In addition we reserve 712 audio clips with curated labels for testing. *Curated* audio clips have been carefully manually annotated [3], whereas *noisy* labels are the result of automated heuristics [1]. As annotations

^{*} This work is supported by the Austrian National Science Foundation (FWF P31988).

are only available at clip-level without time stamps, we speak of weak labels. The average number of labels per curated training clip is 1.03, for curated test clips it is 1.06 and for noisy clips 1.15.

3 Methods

We use a Convolutional Neural Network (CNN) with eight convolutional and three average-pooling layers as proposed in [8]. The inputs of the CNN are 96-bin Mel-spectrograms, transformed to decibel scale. For computing the spectrograms, we use a Fast-Fourier Transform (FFT) size of 2048, a hop-size of 512 and a minimum frequency of 40 Hertz (Hz). Prior to feature computation, raw audio is resampled to 16 kHz. Due to improved results in preliminary experiments, spectrograms are not normalised before being fed into the CNN. In the learning procedure, we use an Adam optimizer with an initial learning rate of 0.001. After 80 epochs, the learning rate is reduced by a factor of 10, and training is continued for 20 more epochs. Batch normalisation and drop-out are used against overfitting. For training, we use one random 3 second snippet per clip; for testing, on the other hand, the full audio is used. In case the raw audio is too short, i.e. shorter than 3 seconds for training samples, circular padding is used.

4 Results

Task 2 of the DCASE2019 Challenge was to maximise the performance of an audio tagging system on weakly multi-labelled data by exploiting both curated and noisy labels. In contrast to this, we focus on the impact these two types of labels have individually. Particularly, we take a closer look at the contribution of noisy labels in the task of music tagging. Before evaluating the performance of our system on different training data, we tune hyper-parameters (cf. section 3) on a separate curated validation set. After this, we train several models; first we use training data with curated labels only, and secondly with noisy labels only. Furthermore, we compare their performance with a random baseline and a variation of the model with intentionally corrupted labels.

To measure the performance of the tagging system, we use the mean average precision (MAP) and the mean area under ROC curves (MAUC) (cf. [10]). We repeat our experiments 5 times for different random CNN initializations, and show mean and standard deviation of MAP and MAUC on the test set over all runs in Table 1. Additionally, we perform paired sample t-tests to determine the significance of different performances. In what follows, differences are denoted as *significant* whenever $|t| > t_{(99, df=4)} = 4.604$.

The first two lines in Table 1 show the performance of our CNN trained on either data with curated labels only, or noisy labels only. Lines 3 and 4 relate to a random baseline and a corrupted version of the curated training set.

More precisely, for line 3 and 4 in Table 1 we train the model on curated data once more, but with two different modifications to the curated labels. First, we create a *random* baseline with an intact label distribution by shuffling the labels

Training Data	MAP	MAUC
Curated Labels	0.767 ± 0.005	0.913 ± 0.004
Noisy Labels	0.665 ± 0.013	0.846 ± 0.010
Random Labels	0.265 ± 0.006	0.502 ± 0.007
Corrupted Labels ($r=70\%$)	0.638 ± 0.019	0.852 ± 0.009

Table 1. Mean \pm std. deviation over 5 runs of models with different training data.

of the training data. Furthermore, we estimate the unknown level of noise present in the noisy dataset by performing the second modification in line 4, for which we intentionally corrupt a certain percentage of curated labels until we reach a similar performance as in line 2. This is done by replacing one single random tag by a random but differing new tag for $r\%$ of curated training data, regardless of the original number of tags for a particular clip. In other words, all clips remain with the same number of tags as before, but with exactly one wrong tag. Note here that due to the low average number of tags per clip (cf. section 2), the resulting level of noise will be closely related to the r value.

Training our CNN with curated audio clips results in a mean performance of 0.767 MAP and 0.913 MAUC as shown in Table 1. This is, for both metrics, a significant difference to the random baseline and to our model trained on data with noisy labels only. Similarly, the difference between the model trained solely on noisy labels (with an average MAP of 0.665 and MAUC of 0.846) and our baseline is statistically significant. If we decrease the number of training samples with noisy labels to correspond to the lower amount of 825 curated training samples, the MAP and MAUC decrease to 0.587 ± 0.012 and 0.795 ± 0.016 , respectively (not shown in Table 1). This benefit of using large quantities of data with noisy labels is in line with previous results on comparable data [2].

For the last line in Table 1, we show the result of training on data with a noise level of $r = 70\%$; this comes close to the performance of training on the actual noisy dataset (not a statistically significant difference). Starting with $r = 0\%$ and increasing this factor with a step-size of 5%, the MAP and MAUC show the first significant decrease when reaching a level of 50% noise. At this point, the average MAP (MAUC) is reduced from 0.767 (0.913) to 0.736 (0.903) (not shown in Table 1). Training the CNN on corrupted labels only, i.e. $r = 100\%$, decreases the two metrics to on average 0.206 and 0.388 respectively, which for the MAUC is a significant difference compared to the random baseline.

5 Discussion and Conclusion

In conclusion, we see that even though training a music tagger on a set of curated audio samples leads to the best performance, a model trained on very noisy labels still outperforms a random baseline significantly, with figures of merit actually much closer to the carefully curated scenario. This is particularly interesting as noisy and curated clips have the same set of classes, but originate from different sources [1]. Being based on Freesound (www.freesound.org) content, curated

audio files often contain isolated sound samples of a class, while noisy files tend to be of a more composite nature, as they consist of Flickr video soundtracks.

To explore the unknown level of noise in the noisy scenario provided for the DCASE2019 challenge [1], we performed additional experiments in which we trained on curated data with intentionally corrupted labels of a certain percentage (cf. [10]). Introducing this controlled amount of noise suggests that the noisy dataset we tried to learn from possibly contains a relatively high amount of 70% wrong labels, although we do not yet have an approximation of how the domain mismatch influences the differences in performance of curated and noisy training data. Nevertheless we were able to show that a weak multi-label audio-tagger trained solely on noisy labels can not only perform significantly better than a random baseline but at a respectable performance level, even in the case of a domain mismatch and a potential high level of noise.

References

1. Fonseca, E., Plakal, M., Font, F., Ellis, D.P.W. and Serra, X.: Audio tagging with noisy labels and minimal supervision. In: Proc. of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop, pp. 69–73. New York University (2019).
2. Fonseca, E., Plakal, M., Ellis, D.P.W., Font, F., Favory, X., Serra, X.: Learning sound event classifiers from web audio with noisy labels. In: Proc. of the IEEE Intern. Conf. on Acoustics, Speech and Signal Processing, pp. 21–25. IEEE (2019).
3. Fonseca, E., Pons, J., Favory, X., Font, F., Bogdanov, D., Ferraro, A., Oramas, S., Porter, A., Serra, X.: Freesound datasets: a platform for the creation of open audio datasets. In: Proc. of the 18th Intern. Society for Music Information Retrieval, pp. 486–493. ISMIR (2017).
4. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.M.: MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: Proc. of the 35th Intern. Conf. on Machine Learning, pp. 2309–2318. PMLR (2018).
5. Li, Y., Yang, J., Song, Y., Cao, L., Luo, J. and Li, L.J.: Learning from noisy labels with distillation. In: Proc. of the IEEE Intern. Conf. on Computer Vision, pp. 1910–1918. IEEE (2017).
6. Liu, T., Tao, D.: Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence* **38**(3), 447–461 (2015).
7. Natarajan, N., Dhillon, I.S., Ravikumar, P.K., Tewari, A.: Learning with noisy labels. In: Advances in neural information processing systems, pp. 1196–1204. Curran Associates (2013).
8. Paischer, F., Prinz, K., Widmer, G.: Audio tagging with convolutional neural networks trained with noisy data. DCASE2019 Challenge (2019).
9. Prinz, K., Flexer, A.: Weak multi-label audio-tagging with class noise. Late-Breaking/Demo ISMIR (2019).
10. Shah, A., Kumar, A., Hauptmann, A.G., Raj, B.: A closer look at weak label learning for audio events. CoRR **abs/1804.09288** (2018).
11. Sukhbaatar, S., Fergus, R.: Learning from noisy labels with deep neural networks. In: Proc. of the 3rd Intern. Conf. on Learning Representations (2015).
12. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. In: Advances in neural information processing systems, pp. 8778–8788. Curran Associates (2018).

Geometric Deep Learning for Music Genre Classification

Manoranjan Sathiyamurthy, Xiaowen Dong, and M. Pawan Kumar

Department of Engineering Science, University of Oxford, Oxford, United Kingdom
manosathya98@gmail.com | xiaowen.dong@eng.ox.ac.uk |
pawan@robots.ox.ac.uk

Abstract. Genres are one of the most commonly used labels to categorise different styles of music. However, no universal definition of a particular genre exists and as such, boundaries between genres can be subjective. In this paper, a multimodal approach within a geometric deep learning setting is explored, using information at an artist level in order to link tracks and establish clearer genre boundaries. Experiments on the MSD-I dataset showed significant improvement when using multimodal graph-based models over methods based on convolutional neural networks alone. The use of domain knowledge also showed noticeable improvement when compared to a purely data-driven approach.

Keywords: Genre Classification · Geometric Deep Learning · Multimodal Fusion.

1 Introduction

In recent years we have seen rapid growth in the use of digital streaming platforms, with large databases of music being made available online. Digital streaming accounted for 58.6% [4] of the revenue generated by the music industry as a whole in 2019, led by streaming services such as Spotify and Apple Music. With this growth it no longer is feasible to manually tag each track with the appropriate metadata and, as a result, research into automated tagging methods is increasing in popularity.

A wide range of work exists in the field of music genre classification [11], with most current machine learning approaches attempting to identify underlying characteristics of audio tracks and utilising these in order to classify songs into appropriate genres. A multimodal approach is suggested in [9] where images and text related to a song, as well as spectrograms of audio samples, are used in a deep learning framework for single-label and multi-label genre classification experiments. However, ignoring potential relationships that can be drawn between tracks of the same, or similar, genre(s) could detract from the quality of classification. Additionally, purely using song level features in a single-label classification task can lead to inconsistencies due to subjective genre label boundaries, with songs potentially overlapping different genres [7].

This paper proposes the use of artist level similarities to form connections between individual songs, allowing song-level features together with relationships between them to be represented in a non-Euclidean form, namely graphs, for use in a geometric deep learning framework [1]. It is thought that since artists generally identify with small numbers of genres, the use of artist level similarities may help better capture relationships between songs hence allowing for improved performance in the genre classification task. We shall also consider the effect of using features from multiple modes [10, 8], following the feature extraction pipeline set in [9].

2 Methodology

2.1 Feature extraction using Convolutional Neural Networks

A baseline is constructed by extracting and classifying features from both audio and visual modalities, by use of convolutional neural networks (CNNs) [2, 3]. Spectrogram representations of the song and the album cover associated with each song form the input to the audio and visual models respectively. The features derived from both modes are embedded in a multimodal space [9], giving rise to feature vectors from a further two modes (mm-audio and mm-visual). Combinations of features from these four modes are used to assess the effects of a multimodal approach. The CNN architectures follow that set out in [9]. Training occurs with Adam as our optimiser and categorical cross entropy as our loss function across all single modal architectures. The visual network utilises the Resnet-101 network with model parameters initialised with those learned in the training of ImageNet. A learning rate of 1×10^{-4} is used with mini batches of 50 samples over 90 epochs with early stopping. The audio network consists of 3 convolutional layers with each layer having 64, 128 and 256 filters respectively. Max pooling is used after each layer, as well as a dropout of 0.5. Mini batches of 32 are used over 100 epochs with early stopping.

2.2 Geometric Deep Learning Approach

A graph based around each mode is formed with nodes representing individual songs and having a graph signal consisting of song-level feature representations from the respective mode, extracted by use of CNNs as described in Section 2.1. An edge between two songs in each graph shall represent the similarity between the artists of said tracks. Each graph is used to train a distinct graph convolutional network (GCN) [6], before the outputs from each graph are combined for classification. Artist similarities can be formed using two distinct methodologies:

(i) *GCN: AGF* Extracting Artist Group Factor (AGF) vectors for each artist [5], from which the cosine similarity between two AGF vectors defines the artist similarity between said artists. The similarity value is binarised by means of a

threshold. AGF vectors are extracted by use of k-means clustering and Latent Dirichlet Allocation with K clusters and R latent groups respectively. The audio, visual, mm-audio and mm-visual configurations that will be used are $K = 200, 200, 500, 500$ and $R = 40, 60, 40, 60$ respectively. A threshold value of 1.0 is found to give optimal performance.

(ii) *GCN: Spotify* Using the Spotify API to identify similar artists based on Spotify users' listening history, in which a similarity value of 1 is given between two related artists.

Table 1. Hyperparameter configuration of the GCN for each mode and each method

Hyperparameter	audio		visual		mm-audio		mm-visual	
	(i)	(ii)	(i)	(ii)	(i)	(ii)	(i)	(ii)
d	0.70	0.40	0.60	0.75	0.70	0.45	0.45	0.40
h	520	20	120	120	1220	620	1120	20
lr ($\times 10^{-4}$)	5.18	1.31	2.42	1.60	3.59	2.75	1.03	1.96

Four different GCNs, one for each mode, shall be trained with each GCN consisting of two convolutional layers followed by a fully connected layer. The outputs from the convolutional layer of each of the four networks shall be combined in various ways to compare single and multimodal performance. Table 1 shows the hyperparameter configuration found through optimisation for each network with d , h and lr being the dropout, hidden layer size and initial learning rate respectively.

3 Experiments

A subset of the Million Song Dataset (MSD), the MSD-I dataset [9], which includes a total of 30,713 tracks spanning 15 unique genres is used. Each track has an associated album cover as well as a single genre label. The dataset contains a total of 9048 artists and 16,753 albums, yielding an average of 3.4 songs per artist and 1.8 songs per album. The dataset split is set to be 70% for training, 15% for validation and 15% for training.

Table 2 shows that, within our single-modal baseline CNN models, visual modes (models **2** and **4**) underperform compared to their audio counterparts (models **1** and **3**). The inclusion of data from multiple modes better performance, with the combination of *mm-audio* and *mm-visual* giving the highest F1 score. A similar pattern is seen in both GCN based models, with visual modes once again achieving the lowest scores. However, GCN based methods are seen to significantly improve on performance when compared to our baseline CNN model, with the inclusion of all four modalities giving the highest F1 score. The use of domain knowledge (*GCN: Spotify*) shows improvement when compared to a purely data-driven approach (*GCN: AGF*) across all combinations of modalities.

Table 2. Macro F1 scores for the classification task of a combination of modes across all three methods.

Model	Mode(s)	Baseline (CNN)	GCN: AGF	GCN: Spotify
1.	Audio	0.346	0.498	0.513
2.	Visual	0.249	0.279	0.341
3.	mm-audio	0.347	0.528	0.615
4.	mm-visual	0.246	0.235	0.353
5.	Audio + mm-audio	0.349	0.676	0.761
6.	Visual + mm-visual	0.241	0.374	0.533
7.	Audio + Visual	0.405	0.597	0.648
8.	mm-audio + mm-visual	0.424	0.521	0.642
9.	All	0.419	0.744	0.788

References

1. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* **34**(4), 18–42 (2017)
2. Choi, K., Fazekas, G., Sandler, M.B.: Automatic Tagging Using Deep Convolutional Neural Networks. In: *Proceedings of the 17th ISMIR Conference*. pp. 805–811 (2016)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770–778 (2016)
4. IFPI: Global music report,(2019). International Federation of the Phonographic Industry (2019)
5. Kim, J., Won, M., Serra, X., Liem, C.C.: Transfer learning of artist group factors to musical genre classification. In: *Companion Proceedings of the The Web Conference 2018*. pp. 1929–1934 (2018)
6. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: *Proceedings of the 5th ICLR* (2016)
7. McKay, C., Fujinaga, I.: Musical genre classification: Is it worth pursuing and how can it be improved? In: *Proceedings of the 7th ISMIR Conference*. pp. 101–106 (2006)
8. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.: Multimodal deep learning. In: *Proceedings of the 28th ICML*. pp. 689–696 (2011)
9. Oramas, S., Barbieri, F., Nieto, O., Serra, X.: Multimodal deep learning for music genre classification. *Transactions of ISMIR*. 2018; 1 (1): 4-21. (2018)
10. Srivastava, N., Salakhutdinov, R.R.: Multimodal learning with deep boltzmann machines. In: *Advances in neural information processing systems*. pp. 2222–2230 (2012)
11. Sturm, B.L.: A survey of evaluation in music genre recognition. In: *International Workshop on Adaptive Multimedia Retrieval*. pp. 29–66. Springer (2012)

Improving Audio Onset Detection for String Instruments by Incorporating Visual Modality

Grigoris Bastas^{1,3}, Aggelos Gkiokas², Vassilis Katsouros¹, and Petros Maragos³

¹ Institute for Language and Speech Processing (ILSP), Athena R.C., Athens, Greece
`{g.bastas, vsk}@athenarc.gr`

² Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
`aggelos.gkiokas@upf.edu`

³ School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece
`maragos@cs.ntua.gr`

Abstract. This paper presents a method for enhancing music audio onset detection in the context of live music performance recordings. As a structural element of our method we utilize a cascade of Temporal Convolutional Neural Networks (TCNs). Conventional frame based spectral representations are used as audio input features, whereas, post-processed body skeletons obtained with OpenPose constitute the visual input source. The network is trained and evaluated on monophonic string recordings from the University of Rochester Multi-Modal Music Performance (URMP) Dataset. Experimental results indicate that our model outperforms audio-based state-of-the-art methods and, additionally, that the visual component enhances detection performance.

Keywords: onset detection · audio-visual analysis · TCN.

1 Introduction

Onset detection is one of the most fundamental problems in the field of Music Information Retrieval (MIR). The state-of-the-art for audio onset detection [8] applies a Convolutional Neural Network (CNN) on spectral representations. However, music is not always experienced by humans solely through the aural modality. For instance, the produced sounds of many musical instruments correspond to certain visible movements and specific positioning of the instrument player’s hands. Regarding the bowed string instruments, bowing motions are comparatively easily detectable and are strongly correlated with note onsets.

In the recent years, deep learning methods for modality fusion have gained increased interest [7]. Several innovative information extraction techniques that rely particularly on fusing audio and visual sources of music have been evolved [2], opening new areas for experimentation and further advancements. Audio-visual analysis focusing on onset detection for string ensembles has been conducted by Li et al. [3] to form a basis for score-informed audio-visual source association. Audio-visual source association has also been handled using vibrato

analysis [6]. In [4], the visual information was reduced to keypoints representing body and finger joints using OpenPose. The vibrato and bow stroke approaches have been combined permitting the generalization of the analysis on woodwind and brass instruments.

In this paper we deploy Temporal Convolutional Neural Networks (TCNs) and we demonstrate that the use of the visual modality can enhance the onset detection method. We focus on bowed string instruments, where the hand and body movement can provide cues on the beginning of the onsets.

2 Method Description

The main architecture employed in this work is a non-causal variant of the TCN model proposed in [1]. The main advantage of TCNs is that, by applying dilated 1D convolutions, they can handle temporal information by conditioning each prediction on an adequately long input, ensuring small added computational burden and large number of trainable parameters at the same time. In this configuration, the dilation factor increases from one layer l to the next by 2^l . At each layer, we apply 150 convolutional filters of size 5 and dropout with probability 0.25. One such network with 6 layers is applied on the visual source (TCN-Visual) and one with 4 layers on the audio (TCN-Audio), both followed by a linear layer with a softmax activation function predicting probabilities of occurring and non-occurring onsets. Our fusion architecture relies on concatenating the outputs of the two models and feeding them to an output network as presented in Fig. 1. Two different output networks were employed: a 4-layer TCN and a 1-layer fully connected network. The predicted onset locations were picked after computing local maxima of the activation function using centered moving maximum with a window size of 5 consecutive frames. Such values were taken under consideration provided that they exceeded a threshold of 0.5.

Our models are trained and tested on monophonic musical performance recordings drawn from the University of Rochester Multi-Modal Music Performance (URMP) Dataset [5] which also provides onset annotations. The raw audio input of 48kHz is further processed and represented in the form of mel spectrograms with 40 frequency bands, hop size of 512 samples and frame length of 2048. As for the visual modality, we chose to use OpenPose for 2D pose estimation and we kept body skeletons comprised of 11 keypoints. Lower body joints, from the knees and below, as well as keypoints corresponding to ears and eyes, were all discarded since they are often occluded and they don't add further musical information. In order to create continuous skeletons that match the audio frame rate, in certain frames, we eliminated specific keypoints that induced unnatural movements, by following the post-processing steps from [4], and we upsampled our data. In frames where certain joints were occluded or eliminated, the keypoints were recreated using linear interpolation between valid frame instances. Standard scaling per feature was applied for each separate performance. Finally, keypoint velocities and accelerations were appended to the feature vectors thus leveraging a 66-dimensional representation.

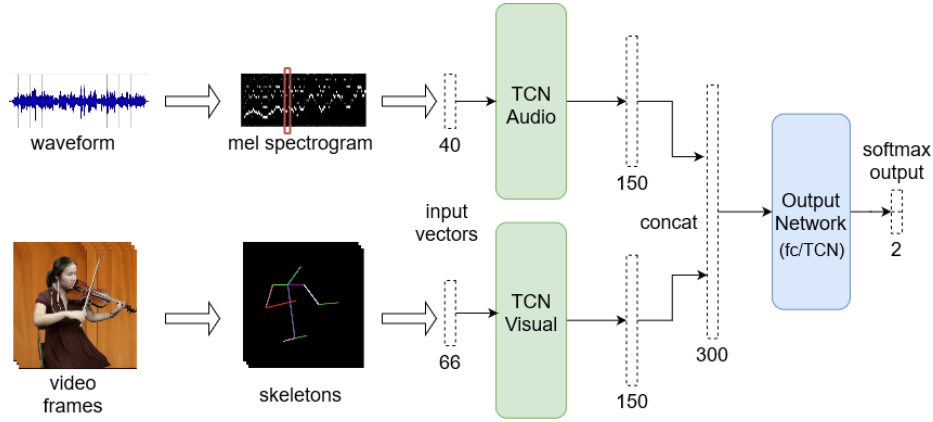


Fig. 1. Fusion model that combines the outputs of the pre-trained visual and audio sub-models by concatenating and feeding them to an output network (fc or TCN).

3 Experimental Setup

We evaluate the dataset using 8-fold cross-validation by computing the F measure for the predictions in each performance, with a tolerance window ± 50 ms around the ground truth values. In the first phase of our experiments, two separate models were tested, one trained on the visual and one on the audio input, using cross-entropy loss. As presented in Table 1, the audio sub-model outperforms the state-of-the-art (CNN-Audio) on URMP dataset. Its visual counterpart naturally yields lower, yet notable results, and exhibits lower stability, as reflected by the relatively high standard deviation among different folds.

In the second phase, the pre-trained sub-models are reloaded and an additional network is fed with the their concatenated output vectors. Separate experiments were conducted in order to test four distinct fusion strategies and their potential to improve the performance in onset detection. The first strategy was based on a cascade of TCN models (TCN-Fusion), where the loaded pre-trained sub-models were rendered free to update their weights while training the whole cascade model. The same arrangement was deployed in a second experiment, this time keeping the sub-model parameters frozen (TCN-Fusion-Frzd). In the alternative layout, with the linear network used in the output instead of the TCN, as previously the sub-model parameters were at first left unfrozen (TCN-LinO-Fusion). However, freezing the pre-trained sub-models (TCN-LinO-Fusion-Frzd) proved slightly more beneficial in this arrangement.

As displayed in Table 2, TCN-Fusion achieves the most notable enhancement (+0.5%) of the onset detector among the tested fusion strategies, in terms of average scores. TCN-LinO-Fusion and TCN-LinO-Fusion-Frzd also outperform the audio sub-model, with little difference from TCN-Fusion which suffered, early on, from over-training. TCN-Fusion-Frzd was the only among the four models to exhibit no enhancement of the detection performance.

Table 1. Performance of models trained on distinct modalities with 8-fold cross-validation.

Models	F measure	
	Mean	Std.
TCN-Visual	0.64	5.82%
TCN-Audio	0.921	1.80%
CNN-Audio[8]	0.886	1.19%

Table 2. Performance of fusion models for 8-fold cross-validation.

Fusion Models	F measure	
	Mean	Std.
TCN-Fusion	0.926	1.99%
TCN-Fusion-Frzd	0.898	3.54%
TCN-LinO-Fusion	0.923	2.22%
TCN-LinO-Fusion-Frzd	0.925	1.66%

4 Conclusions

The audio-visual onset detection exhibited a non negligible improvement over the models which were trained solely on one source. This fact entails that the visual model captured information that the audio model alone couldn't.

As future work, the need to experiment with new fusion strategies is one of our priorities. The same is true about improving the performance of the visual model alone. This can have a positive impact on the fusion model. Experimenting with polyphonic performances is another possible path which could give us the opportunity to push further the limits of audio-visual onset detection analysis.

References

1. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
2. Duan, Z., Essid, S., Liem, C.C., Richard, G., Sharma, G.: Audiovisual analysis of music performances: Overview of an emerging field. *IEEE Signal Processing Magazine* **36**(1), 63–73 (2018)
3. Li, B., Dinesh, K., Duan, Z., Sharma, G.: See and listen: Score-informed association of sound tracks to players in chamber music performance videos. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2906–2910. IEEE (2017)
4. Li, B., Dinesh, K., Xu, C., Sharma, G., Duan, Z.: Online audio-visual source association for chamber music performances. *Transactions of the International Society for Music Information Retrieval* **2**(1) (2019)
5. Li, B., Liu, X., Dinesh, K., Duan, Z., Sharma, G.: Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications. *IEEE Transactions on Multimedia* **21**(2), 522–535 (2018)
6. Li, B., Xu, C., Duan, Z.: Audiovisual source association for string ensembles through multi-modal vibrato analysis. *Proc. Sound and Music Computing (SMC)* (2017)
7. Ramachandram, D., Taylor, G.W.: Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine* **34**(6), 96–108 (2017)
8. Schlüter, J., Böck, S.: Musical onset detection with convolutional neural networks. In: 6th international workshop on machine learning and music (MML), Prague, Czech Republic (2013)

Audio textures in terms of generative models^{*}

Lonce Wyse^[0000–0002–9200–1048] and Muhammad
Huzaifah^{1[0000–0002–7188–3600]}

National University of Singapore, Singapore {lonce.wyse,e0029863}@nus.edu.sg

Abstract. Audio textures, with complex structure spanning different time scales, are a challenge to model and generate synthetically with musical instrument-like interactive control. They are even challenging to define. Deep learning approaches offer new ways to develop generative audio texture models, and they create different demands on training data than traditional modeling approaches. In this paper we briefly review previous modeling approaches, and attempt to rationalize and converge on a definition of textures using modeling concepts. We introduce a new and growing data set along with a system for managing metadata specifically designed for audio textures. Finally, we report on some recent advances in modeling these types of sounds.

Keywords: audio texture · sound database · generative models · audio synthesis.

1 Introduction

1.1 Modeling Objective

The overall goal of this work is to create generative models of audio textures, a class of sounds quite different, and arguable much larger and more complex, than the class of pitched musical instrument sounds. Ideally, the models should be capable of convincingly generating “natural” textures such as rain, crowd murmur, crackling fire, or wind. They should offer control parameters that can be designed to correspond to arbitrary paths through the space of sounds within a model’s range (for example controls that are perceptually, semantically, or musically meaningful). The model should be responsive to parameters in real time, be capable of generating novel sounds between and beyond the sounds used as training examples, and finally, be able to generate sound for any length of time.

1.2 Defining textures

The literature on audio texture modeling is full of discussions and definitions of that grapple with similar concepts, but a definition has proved elusive. Saint-Arnaud and Popat[8] plotted the relationship between time and the information

^{*} This research is supported by a Singapore MOE Tier 2 grant, “Learning Generative Recurrent Neural Networks,” and by an NVIDIA Corporation Academic Programs GPU grant.

content in 3 classes of sound: 1) noise, 2) texture, and 3) speech and music. As time progresses, the information in noise signals plateau early, while for speech and music, information continues to rise with time. The graph of information value for sound textures is in between that of the other two classes. It plateaus, but at a higher level than noise, and at a point further along the time access. This captures one of the key concepts concerning sound textures, that there exists a window of time beyond which the description of the sound remains the same. But it is worth being a little more precise. If the generating model is not known *a priori*, the graph of the information content of a sound texture curve approaches an asymptote but does not ever reach a slope of zero. Each new piece of data can be used to refine a model further whether the model is perceptual or computational.

Some authors have attempted to define textures by qualities inherent in the sound rather than how it is modeled. For example, Schwarz[10], classifies contact sounds from interaction with objects (such as friction and rolling sounds) as not textures. This is because they violate the "wallpaper" premise that fine structure must remain constant over time. Similarly, Strobl et al.[11] rule out the sound of a crying baby as a texture because "the characteristics of the fine structure are not constant enough". These intuitions about how a perceptible "arrow of time" undermines the static nature of audio textures can be clarified in two different ways with the help of some model-based terminology. One is to separate the description of the information that is not constant over shifting windows of time from the rest of the description. That is, we can recognize a layer of "content" upon which the textural part is conditioned. For example, a rolling stone has weight and rate characteristics which determine characteristics of the resulting sound. If those characteristics change over time, then so does the texture. The distinction between content and texture depends on which aspects we are explicitly modeling across time, and which aspects of variation are drawn from a distribution that is constant across time. Thus the content/texture distinction is made in terms of the model used to generate, describe, or perceive a sound, and can be made differently even for one and the same audio example. Another way to contextualize sounds with a strong "arrow of time" as textures is by not considering them in isolation, but *en masse*. For example, if a collection of rolling stones were heard, each having its own (possibly changing) speed chosen from a random distribution, then the resulting sound would be a texture because the description would be constant over different windows of time.

1.3 Previous modeling strategies

Past approaches to synthesizing audio textures have used granular synthesis[7, 9] incorporating a distinction between fine time scale of audio and the larger scale of grains, and wavelet trees that capture statistical relationships across both time and hierarchical scales[2] based on individual sound examples. McDermott and Simoncelli[6] developed a method for matching a set of specific statistics of natural sounds. However, selecting specific time scales or statistics is to make modeling commitments that may not hold for all natural sounds. More recent

deep learning approaches tend to avoid engineering features, and to let the model discover which features are important for a given data set.

2 PSoundSet - an audio texture database

Training neural networks typically requires large amounts of training data. There are good reference data sets for music[1], musical instruments[3], and environmental scenes[4], but they are either not specific to audio textures, or else labeled for training classifiers or unconditional generators rather than for training synthesizers under parametric control[5]. We have started building a new data base, Parameterized Audio Textures Data Sets (PATSets, available online¹) to address this need.

The small but growing PATSet collection consists of both natural and synthetic textures. Each set consists of multiple files that individually or in aggregate sample a parameter space (e.g. engine speed) that can be used for conditional training. To handle the parameter management (labeling, writing and reading files), we developed the paramManager (open sourced²) and a json-like parameter file format. A key feature is that parameters are stored as a pair of time and value arrays so that they can be sampled at much lower rates than the audio files, and rates do not have to be regular. The paramManager code for reading values at specific times (during training, for example) interpolates between the stored values in the file.

Each sound set has its own database entry that includes other metadata about how the sound was recorded or constructed as well as technical (bitrate, channels, coding) specifications. Synthetic sets are stored with the code for creating them so that the integrity of the sound descriptions can be tested and verified, and so that sounds sets other than those already stored on the database can easily be generated. Currently, the PATSet database allows auditioning of all stored sounds, and files are converted to the required sample rates when downloaded. In the future, most synthetic sound sets could be stored and downloaded as code to be constructed only how, when, and where they need to be used.

3 Experiments

We are currently generating and training models mostly with synthetic data sets in order to systematically explore the capabilities and limits of the models for textures (testing long time dependencies in RNNs for example). We are exploring an RNN that we previously developed for modeling musical instrument tones (Wyse[12]). We use an RNN because it can be responsive to control parameter changes within one audio sample in contrast to CNNs (and other

¹ <https://sonicthings.org/9999/>

² <https://github.com/lonce/paramManager>

architectures) that produce an extended duration of output for each parameterized control vector update. The model has been shown to generalize well across a sparsely-trained musical pitch space, but struggles to generate timbres “in between” instruments used in training. Here we report on the ability of this architecture to model sound textures, far more complex than pitch instrument tones, with statistical variation across a continuous range of time scales.

To demonstrate the core competence of the model for modeling textures, we chose one of the synthetic sound sets in the PATSet database, “RegularPops68.” The sound is constructed of a series of regularly spaced “pop” events at rates from 2/sec to 32/sec, with rate as the parameter for conditioning during training and control during synthesis. Each pop consists of 3 random samples of audio followed by a narrow band-pass filter with a center frequency of 415Hz (midi note 68). The 3 random samples give a significantly different timbre and amplitude to every single event. We expect the model to generate the ever-changing but statistically constant variation of each event at the sample rate as well to model the changing regular event rate specified by the conditioning parameter (see Fig. 1, and corresponding audio online³).

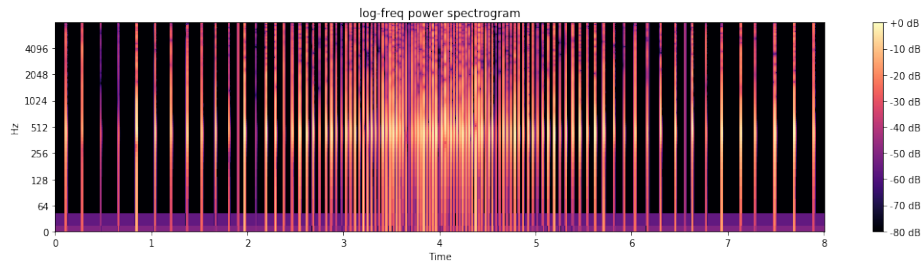


Fig. 1. Audio generated by an RNN trained on regularly spaced filtered random noise “pops” parameterized by an event rate parameter. Both the conditionally specified regular events and the unconditioned constant variation of the timbers are well captured and reproduced.

4 Summary

Recent approaches to modeling provide tools for generating complex audio textures with differently structured information across a continuum of time scales. We discussed the need to recognize the model in how textures are defined. We introduced a new data set of audio and labels appropriate for training texture models, and are exploring the potential of RNNs to model audio textures at different time scales without having to engineer features.

³ <http://animatedsound.com/research/MML2020/RegularPops68.wav>

References

1. Bertin-Mahieux, T., Ellis, D.P., Whitman, B., Lamere, P.: The million song dataset. In: Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011). Miami, FL, USA (2011)
2. Dubnov, S., Bar-Joseph, Z., El-Yaniv, R., Lischinski, D., Werman, M.: Synthesizing sound textures through wavelet tree learning. *IEEE Computer Graphics and Applications* **23**(4), 38–48 (2002)
3. Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D., Simonyan, K.: Neural audio synthesis of musical notes with wavenet autoencoders. In: International Conference on Machine Learning. pp. 1068–1077 (2017)
4. Gemmeke, J., Ellis, D., Freedman, D., Jansen, A., Lawrence, W., Moore, R., Plakal, M., Ritter, M.: Audio set: An ontology and human-labeled dataset for audio events. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 776–780 (2017)
5. Huzaifah, M., Wyse, L.: Deep generative models for musical audio synthesis. arXiv preprint arXiv:2006.06426 (2020)
6. McDermott, J.H., Simoncelli, E.P.: Sound texture perception via statistics of the auditory periphery: evidence from sound synthesis. *Neuron* **71**(5), 926–940 (2011)
7. Roads, C.: Automated granular synthesis of sound. *Computer Music Journal* **2**(2) (1978)
8. Saint-Arnaud, N., Popat, K.: Analysis and synthesis of sound textures. In: Okuno, H.G., Rosenthal, D. (eds.) *Readings in Computational Auditory Scene Analysis*. Erlbaum (1995)
9. Schwarz, D.: Corpus-based concatenative synthesis. *IEEE Signal Processing Magazine* **24**(2), 92–104 (2007)
10. Schwarz, D.: State of the art in sound texture synthesis. In: 14th Int. Conf. Digital Audio Effects. Paris, France (2011)
11. Strobl, G., Eckel, G., Rocchesso, D., Le Grazie, S.: Sound texture modeling: A survey. In: Proceedings of the 2006 Sound and Music Computing (SMC) International Conference. Marseille, France (2006)
12. Wyse, L.: Real-valued parametric conditioning of an RNN for interactive sound synthesis. In: 6th International Workshop on Musical Metacreation (arXiv preprint arXiv:1805.10808). Salamanca, Spain (2018)

Evaluation of Different Symbolic Encodings for Music Generation with LSTM Networks

Manos Plitsis¹, Kosmas Kritsis^{2,3}, Maximos Kaliakatsos-Papakostas², and Vassilis Katsouras²

¹ Computer Science Dept., Sorbonne University, Paris, France
{plitsis}@ircam.fr

² Institute for Language and Speech Processing, Athena R.C., Athens, Greece
{kosmas.kritsis, maximos, vsk}@athenarc.gr

³ Computer Science Dept., University of Piraeus, Piraeus, Greece

Abstract. This paper attempts an empirical study on the effects of symbolic music encoding for music generation with Recurrent Neural Networks. Three distinct music encodings are examined and their characteristics are discussed. Using a simple Long Short-Term Memory recurrent architecture, we generate different music excerpts and we evaluate their output using statistical domain-based measures and human expert knowledge.

Keywords: Symbolic-music representation · LSTM · Music generation.

1 Introduction

Choosing a proper data encoding is complicated, since the same musical piece can be represented in a range of different expressive ways. While data representation/encoding is one of the most important aspects of automatic music generation, to the best of our knowledge, there have been no previous attempts to measure its implications empirically. This paper presents preliminary results of an ongoing research on empirically quantifying the effects of different representations in the latent features learned by Recurrent Neural Networks (RNNs).

We identify two main families of encodings of symbolic music data: event-based [2] and timestep-based encodings [1]. Even though in both encodings a musical piece is represented as a sequence of events, their essential difference is that in the second type each event has a fixed time-length, while in event-based encodings, time is not directly related to the length of the sequence - each 'step' does not move time forward a-priori. Specifically, we study two different timestep-based encodings, which we call *tstep1* and *tstep2* (making a total of three encodings, including the aforementioned event-based, denoted as *event1*).

In *tstep1*, the possible events range between 0-129. Events 0-127 are "note-on" events, corresponding to the 128 MIDI note numbers, 128 is a rest event, while 129 signifies "continue previous event". This encoding is very parsimonious and efficient, with the drawback that the event distribution it produces is greatly skewed towards the 129 event (see Fig. 1b). In *tstep2*, we also have events in the

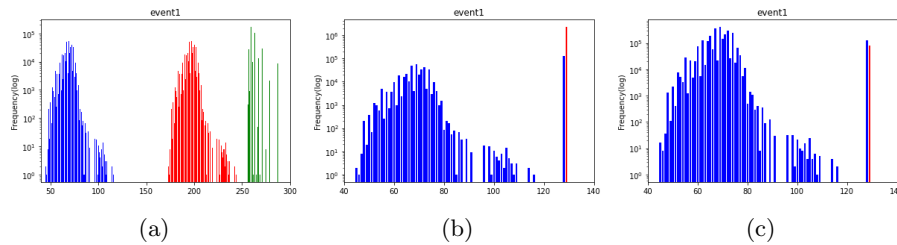


Fig. 1: Distribution of ground truth labels for each representation (in logarithmic scale): a) *event1*, b) *tstep1* and c) *tstep2*. The y axis denotes number of occurrences.

0-129 range, with the main difference that here we assign to 129 a “note-off” event, which stops the previously played note. This has the drawback that a higher resolution than *tstep1* has to be used, in order to model two consecutive same-pitch notes of the minimum duration [1], even though it greatly balances the dataset distribution (see Fig. 1c). Its main advantage is that time is directly given by the length of each sequence, letting the network to learn timing without additional cues, such as meter information [1].

Event1 uses events denoted with integers in the range 0-337, where: 0-127 are “note-on” events, 128 denotes a rest, 128-255 are “note-off” events and 256-337 denote time-shift events, that move time forward by a specific value. This encoding has the advantage of integrating seamlessly other features, such as velocity, by adding special events (as done in [2]). Additionally, by choosing appropriate time resolution for our time-shift events, we can represent longer sparse pieces of music with very short sequences, thus reducing memory and computation cost significantly.

2 Dataset, Preprocessing and Network Architecture

Restricting ourselves to monophonic melodies in the european canon, we chose to use 7264 melody transcriptions in the ***kern* format (a light text-based symbolic music format), from the Essen corpus of the online KernScores library¹. Using the python library *music21*², we extracted the necessary information from the ***kern* files for each encoding and exported as numpy arrays. Then we split each encoding array into batches of 256 sequences, with a sequence length of 64. Additional preprocessing of the input data involves one-hot encoding, to feed into the network. We have chosen to use a very simple architecture, with a single layer of 36 LSTM cells, followed by a dense layer with a softmax activation. A sliding window method is used to auto-regressively produce new sequences. This choice is deliberate; using a simple well-known system, we can focus on the differences caused by the data encodings considered herein.

¹ <https://kern.humdrum.org/>

² <http://web.mit.edu/music21/>

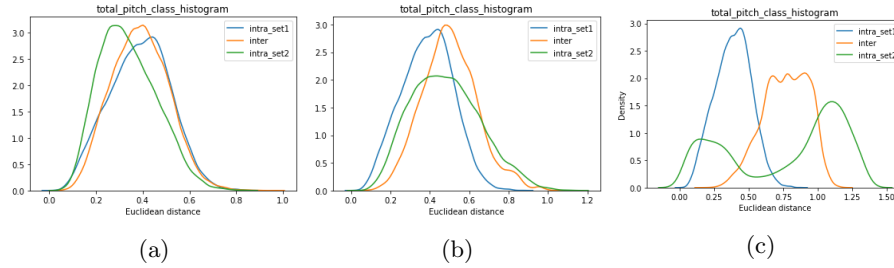


Fig. 2: Pitch Class Histogram intra and inter-set distributions for the three generated sets. Set1 is always the Dataset, while set2 is: a) *event1*, b) *tstep1* and c) *tstep2*.

3 Evaluation and Results

We follow the domain-specific evaluation strategy for music generation networks proposed in [3]. We choose 200 random pieces from our validation set, and by using the first part as a seed melody we generate a set of 200 8-bar melodies (exactly 16 seconds long with a BPM of 120) for each encoding. We then compute 12 music-specific features for each set: Pitch Count (PC), Note Count (NC), Pitch Class Histogram (PCH), Pitch Class Transition Matrix (PCTM), Pitch Range (PR), Average Pitch Shift (PS), Average Inter-onset Interval (IOI), Note Length Histogram (NLH), Note Length Transition Matrix (NLTM).

In order to be able to compare different systems, we perform an exhaustive cross-validation, by comparing each sample from one set to either all other samples of the same set (intra-set distance), or to all samples of another set (inter-set distance). These relative measures give a histogram for each feature, from which we compute a continuous Probability Density Function (PDF) using Kernel Density Estimation. To compare the generated data with the original, we compute the KL Divergence (KLD) and Overlapping Area (OA) between the intra-set PDF of the generated data with the inter-set PDF between the generated and original data. A low KLD indicates similarity in the shape of the compared distributions while a high OA indicates a higher probability density overlap.

We focus towards answering the question: which of the three representations produces pieces that more closely resemble the initial dataset and why? The absolute measures, while giving some insights, they provide only a weak correlation, or fail to give the “bigger” picture.

Looking at the relative measures, we begin to see some trends. In Fig. 2, we plot the computed PDFs for the intra-set and inter-set distances of the Pitch Class Histogram, between the original dataset and each generated set. We notice a repeating tendency when looking at the plots for all features, that *event1* is always closer to the original set. This can also be seen when looking at the similarity measures in Table 1, where *event1* has consistently very high OA and

Table 1: Intra-set similarity measures for the three encodings.

		PC	PC/bar	NC	NC/bar	PCH	PCH/bar	PCTM	PR	PS	IOI	NLH	NLTM
tstep1	KLD	0.738	0.059	0.286	0.162	0.054	0.310	0.112	0.770	0.001	0.038	0.045	0.400
	OA	0.831	0.589	0.569	0.532	0.844	0.847	0.390	-	0.935	0.829	0.763	0.461
tstep2	KLD	0.272	0.403	0.028	0.110	1.094	1.491	0.326	0.020	0.008	0.031	1.525	0.058
	OA	0.819	0.943	0.944	0.847	0.339	0.357	0.638	0.864	0.881	0.847	0.058	0.236
event1	KLD	0.501	0.069	0.017	0.022	0.035	0.132	0.079	0.164	0.004	0.006	0.043	0.061
	OA	0.908	0.925	0.906	0.933	0.815	0.828	0.877	0.892	0.860	0.922	0.909	0.891

small KLD, which suggests a high similarity to the original dataset, while the other two encodings show some "better" and some "weaker" features.

This is all consistent with human expert evaluation. When listening to generated samples, a number of observations are apparent: Samples from *event1* are clearly superior in quality, following both rhythmically and tonally the style of the dataset. The only anomaly (which is not obvious from the statistical analysis) is that the examples fail to follow a metric structure, resulting in syncopations at best or entirely arhythmic at worst (which happens rarely). This is probably due to directly sampling the softmax distribution, resulting in "wrong" time-skip events being chosen (especially given the distribution of the time skip events - see Fig. 1a). Many of these syncopations are also inaudible when listening without a metronome or rhythm accompaniment. *Tstep1*, while being mostly consistent in metric structure, produces many off-key notes and irregularly spaced intervals (this can also partly be due to sampling). Examples from *tstep2* seem to be the weakest of the three, both rhythmically and in pitch, holding many long notes (as expected by its data distribution), followed by flurries of short notes (which can be due to sampling).

All the generated examples of the study at hand can be found at the project's online repository³.

4 Conclusion and Future Work

While the current study is considered successful, there remain many unexplored parameters, such as the importance of resolution or timestep length, the use of more harmonically and rhythmically diverse datasets, and also to polyphonic music, with expressive information such as velocity. Maybe the most important question which has not been tackled, is the apparent superiority of text-based representations, especially for monophonic music generation. Using findings from our research, we hope to produce an exhaustive classification of the most effective ways to model symbolic music. Ongoing work also involves a deeper investigation of the network's inner workings, such as weight activation patterns for each encoding or applying and investigating attention mechanisms and other network architectures.

³ <https://github.com/manosplitsis/MusicRep>

References

1. Eck, D., Schmidhuber, J.: Finding temporal structure in music: Blues improvisation with lstm recurrent networks. vol. 12, pp. 747 – 756 (02 2002). <https://doi.org/10.1109/NNSP.2002.1030094>
2. Oore, S., Simon, I., Dieleman, S., Eck, D., Simonyan, K.: This time with feeling: Learning expressive musical performance. CoRR **abs/1808.03715** (2018), <http://arxiv.org/abs/1808.03715>
3. Yang, L.C., Lerch, A.: On the evaluation of generative models in music. Neural Computing and Applications pp. 1–12 (2018)

Medley2K: A Dataset of Medley Transitions

Lukas Faber*, Sandro Luck*, Damian Pascual*, Andreas Roth*, Gino Brunner,
and Roger Wattenhofer

ETH Zurich, Switzerland

{lfaber,dpascual,brunnegi,wattenhofer}@ethz.ch
{sluck,rothand}@student.ethz.ch

Abstract. The automatic generation of medleys, i.e., musical pieces formed by different songs concatenated via smooth transitions, is not well studied in the current literature. To facilitate research on this topic, we make available a dataset called Medley2K that consists of 2,000 medleys and 7,712 labeled transitions. Our dataset features a rich variety of song transitions across different music genres. We provide a detailed description of this dataset and validate it by training a state-of-the-art generative model in the task of generating transitions between songs.

1 Introduction

Automatic music generation has undergone major development in the last few years, thanks to the progress of deep learning. Indeed, previous studies have demonstrated the ability of deep learning models to generate pleasant music in many different applications Yang et al. (2017); Dong and Yang (2018); van den Oord et al. (2016); Waite et al. (2016); Pati et al. (2019); Boulanger-Lewandowski et al. (2012); Briot et al. (2019); Brunner et al. (2018); Huang et al. (2019); Morgen (2016). To perform well, these models need large amounts of training data and, depending on the application, collecting such data may not be a trivial task. In this work, we contribute to the growing field of automatic music generation by presenting Medley2K, a new dataset for MIDI medley composition.

A medley is a special type of music piece that is formed by connecting different songs through specifically crafted musical transitions. Despite the popularity of medleys, existing literature has not addressed transition generation, yet. One reason for this is the lack of medley-specific datasets, which hampers progress in this field. Collecting such a dataset is challenging since it requires precise annotations of the transitions between individual songs. Our dataset contains machine-readable labeled transitions extracted from 2000 human-curated medleys. In this work, we give a complete description of Medley2K, and we empirically show that it can be used to train deep generative models for medley transition generation.

* Authors in alphabetical order.

2 Related Work

Although a considerably large number of datasets for music modeling are publicly available (a sample collection can be found here¹), none of those datasets is tailored to medley composition. In particular, the name-related MedleyDB dataset (Bittner et al., 2014, 2016) contains polytrack music of single songs rather than medleys. Conversely, our dataset consists of medley pieces with detailed labels on the transition points in order to foster further work on automatic medley composition.

3 Dataset: Medley2K

Medley2K is a new dataset that consists of 2000 human-created medleys crawled from the website `musescore.com` with a total of 10,269 transitions. All medleys are licensed as shareable, while only a subset is available for commercial use. The dataset contains a rich variety of medleys spanning across several paces, musical scales, and genres. The medleys are on average 6 minutes long with 17.47 key changes and 9.99 tempo changes. Furthermore, the medleys in the dataset have rich instrumentation, featuring an average of 7.65 different instruments. Additionally, as seen in Figure 2a, all instruments except for the “Acoustic Grand Piano” occur in less than 10% of medleys, which means that the instrumentation varies largely across samples. The medleys from musecore come as MIDI files, together with PDF scoresheets and a machine-readable MXL file. Typically, composers annotate the point in time where one song in the medley transitions into the next in the scoresheet. These transitions usually start at the beginning of a new bar. We parse the MXL file for annotation indicating such transitions points. To ensure data quality, we filter annotations that do not indicate a transition; in particular, we ignore annotations of numbers, musical symbols (such as ♯), or a manually defined blacklist of musical expressions (such as “vivante”).

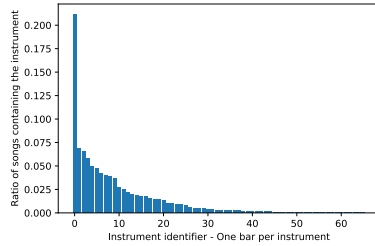
We evaluate the quality of this extraction method on 30 medleys manually labeled, with a total of 205 actual transitions. Table 1 shows the confusion matrix between the actual transition points and the labels given by our extraction method. Overall, the automated extraction achieves a precision of 90.70% and a recall of 57.07%. Note that the high precision value indicates that what we identify as a transition (and will potentially feed into a machine learning model) is very likely a genuine transition. The recall means that we can still extract more transition points, i.e., assuming the method has a similar recall over the whole dataset we could find around 18,000 transitions. Thus, the labeling — while not complete — is of high quality.

¹ <https://ismir.net/resources/datasets>

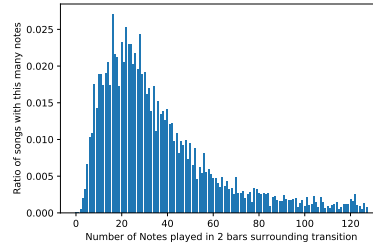
	True Positive	True Negative
Predicted Positive	117	12
Predicted Negative	88	4370

Table 1: Validation of Labeling Process

Next, we examine the notes around the transition point. We observe that for a large fraction of transitions (around 30%) the music around the transition point contains only silence or a single long-held note. These samples cannot be used to learn transitions that consist of more than one note. Since we want to focus on musically pleasant transitions with different notes, we filter the data by looking at the two half bars preceding and the two half bars following the transition point. If a new note starts in either of those four half bars, we keep the transition, otherwise we discard it. This way, each transition consists of at least four played notes. In Figure 2b we show in more detail the number of notes played. The filtered transitions have a high variety of notes ranging from four to more than 60 notes. After this postprocessing, we compose the final dataset with a total of 7,712 labeled transitions.



(a) Instrumentation distribution. Every bar corresponds to one instrument; the first bar is “Acoustic Grand Piano”.



(b) Frequencies of notes played after postprocessing. Every bar corresponds to one number.

4 Experimental Evaluation

In this section, we conduct an experimental study on the validity of the Medley2K dataset for automatic medley composition. To this end, we use a deep neural model that learns to generate Medley transitions as a specialization of the task of filling gaps in music — also called music inpainting. We build on the InpaintNet architecture by Pati et al. (2019) and extend it to support polyphonic music while keeping the same hyperparameters. Given that some internal components of the InpaintNet architecture are tailored to 4/4 beats, we omit in this experiment all transitions with a different beat, resulting in 4,662 transition points. For each transition, we generate a sample by taking the four bars around

the label plus the four bars preceding (past context) and following (future context) the transition point, i.e., 12 bars in total. We encode the data from our Medley2K dataset with a scheme similar to Hadjeres et al. (2017), except that instead of using one symbol for holding the previous note, we use one extra symbol *per note* to denote “Hold”. Although it doubles the number of classes, we found that this encoding reduces class imbalance and improves model performance.

To validate our dataset, we compare two models, one trained with transition data and one trained with arbitrary portions of music from the dataset. We split the transition data into 80/10/10 for training, validation, and test, where the test data is used to evaluate both models. Furthermore, for each model, we consider two training sets, one consisting of 100% of the training transitions (or the equivalent number of samples of arbitrary music), and one with 50% of the samples. The results of these experiments are shown in Figure 3, which shows that given the same amount of data, training on transition data yields better performance on the test set than arbitrary music. In fact, even using only 50% of the transition training is better than using twice as much data of arbitrary music, which demonstrates that training on transitions largely benefits the automatic composition of medleys. This validates our Medley2K dataset as a valuable tool for further work in automatic medley generation.

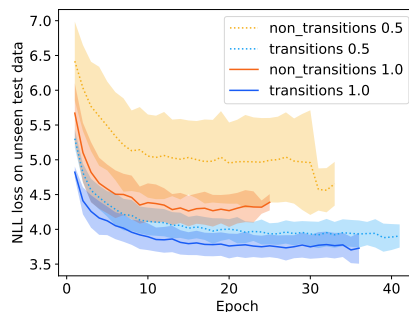


Fig. 3: Performance (NLL loss) of the generative model. Solid lines denote full training sets, dotted lines half training sets. Training on transitions only (bottom lines) achieves better results than training on general music (top lines).

5 Conclusion

We make available² the first dataset for medley composition of MIDI music. The dataset has a rich variety of music pieces, instrumentation, key changes, and tempos. We provide machine-readable labels for 7,712 transition points and validate the dataset by demonstrating its ability to train a state-of-the-art model for music generation. We expect that this dataset will encourage further research in the field of medley generation and automatic medley detection.

² <https://polybox.ethz.ch/index.php/s/STSczoZ2e0IcoVf>

Bibliography

- Bittner, R.M., Salamon, J., Tierney, M., Mauch, M., Cannam, C., Bello, J.P.: Medleydb: A multitrack dataset for annotation-intensive mir research. In: ISMIR. vol. 14, pp. 155–160 (2014)
- Bittner, R.M., Wilkins, J., Yip, H., Bello, J.P.: Medleydb 2.0: New data and a system for sustainable data collection. ISMIR Late Breaking and Demo Papers (2016)
- Boulanger-Lewandowski, N., Bengio, Y., Vincent, P.: Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In: Proceedings of the 29th International Conference on Machine Learning (ICML) (2012)
- Briot, J.P., Hadjeres, G., Pachet, F.: Deep Learning Techniques for Music Generation. Springer (2019)
- Brunner, G., Konrad, A., Wang, Y., Wattenhofer, R.: MIDI-VAE: Modeling Dynamics and Instrumentation of Music with Applications to Style Transfer. In: 19th International Society for Music Information Retrieval Conference (ISMIR), Paris, France (September 2018)
- Dong, H.W., Yang, Y.H.: Convolutional generative adversarial networks with binary neurons for polyphonic music generation. In: 19th International Society for Music Information Retrieval Conference (ISMIR) (2018)
- Hadjeres, G., Pachet, F., Nielsen, F.: DeepBach: a steerable model for Bach chorales generation. In: Proceedings of the 34th International Conference on Machine Learning. pp. 1362–1371 (2017)
- Huang, C.A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., Dai, A.M., Hoffman, M.D., Dinculescu, M., Eck, D.: Music transformer: Generating music with long-term structure. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019 (2019)
- Mogren, O.: C-rnn-gan: A continuous recurrent neural network with adversarial training. In: Constructive Machine Learning Workshop (CML) at NIPS 2016. p. 1 (2016)
- van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. In: Arxiv (2016), <https://arxiv.org/abs/1609.03499>
- Pati, A., Lerch, A., Hadjeres, G.: Learning to traverse latent spaces for musical score inpainting. In: Proc. of the 20th International Society for Music Information Retrieval Conference (ISMIR). Delft, The Netherlands (2019)
- Waite, E., Douglas Eck, A.R., Abolafia, D.: Project magenta: Generating long-term structure in songs and stories (2016), <https://magenta.tensorflow.org/2016/07/15/lookback-rnn-attention-rnn>
- Yang, L.C., Chou, S.Y., Yang, Y.H.: Midinet: A convolutional generative adversarial network for symbolic-domain music generation. In: 18th International Society for Music Information Retrieval Conference (ISMIR) (2017)

A Machine Learning Approach to Cross-cultural Children's Songwriting Classification

Rafael Ramirez¹ and Kari Saarilahti²

¹ Music and Machine Learning Lab
Music Technology Group
Universitat Pompeu Fabra
Roc Boronat 138
Barcelona, Spain
² INTO School
Helsinki, Finland
rafael.ramirez@upf.edu

Abstract. We describe a pilot cross-cultural study involving the examination of music materials composed by children within specific geographical regions. The music materials are audio song-writing compositions during workshops as part of the European Project Future Songwriting in three schools in Germany and Finland. We extract music information retrieval techniques to extract acoustic features from the audio recordings, including spectral centroid, spectral bandwidth, roll-off, zero crossing rate and MFCCs, and apply feature selection to the resulting feature set. We then apply machine learning techniques to classify recordings from different geographical areas. Interestingly, we obtain predictive models capable of classifying the music materials with accuracies well above chance level. This seems to indicate that the features considered provide acoustic information about the pieces and that machine-learning algorithms are capable of use this information to distinguish the compositions of different geographical locations.

Keywords: Songwriting, music composition, machine learning, music information retrieval.

1 Introduction

Music may be characterized by three aspects: sound, behaviour, and concept (Tooby, 1990). Music sound can be defined as a class of auditory signals that are produced by performers, and perceived by listeners, which is composed of melodic, harmonic, rhythmic, timbre, temporal and dynamic components. Music behaviour is associated with activities such as performance, composition, dance, ritual, etc. Music concept has specific functions within any social group (Clayton, 2001; Cross, 2003;

2006; Dissanayake, 2001). The culture concept refers to the set of behaviours, beliefs, social structures, and technologies of a population that are passed down from generation to generation. It includes social conventions related to art, dress, dance, music, religion, etc. It is worth noting that the term culture is not equivalent to ‘country’ or ‘continent’. Moreover, most individuals do not ‘belong’ to a single culture.

A number of cultural influences can act upon particular individuals, merging and manifesting themselves when performing and composing music. In this paper, we describe a pilot cross-cultural study involving the analysis of music materials composed by children in different geographical regions. The music materials are songwriting compositions produced during school workshops as part of the European Project Future Songwriting in nine schools in Germany and Finland. We are sensitive to the fact that it is impossible to characterize nations as singular cultures and compare them with one another. Instead, the current study attempts to investigate if there are common or distinctive compositional patterns in schools in different geographical regions.

2 Materials and Methods

2.1 Music material

Songs composed by children in Finland and Germany were recorded in the context of the Future Songwriting project (www.futuresongwriting.eu). Future Songwriting is a two-year European project focusing on creativity and digital tools in music education funded by the Creative Europe programme of the European Commission. Future Songwriting involves creative school pilot projects for students in Finland, France and Germany. During school pilots, pupils get to compose, write lyrics, arrange, record and produce their own songs. Students compose music by utilizing technology and do not require prior musical knowledge or of music theory, or the ability to play traditional instruments. For this preliminary paper we consider 20 song compositions recordings from 3 schools in different geographic locations, in Finland (2 schools) and Germany (1 school).

2.2 Methods

The following music features were extracted from the audio recordings using Essentia (Bogdanof, 2013), an audio analysis library for music information retrieval developed by the Music Technology Group, Universitat Pompeu Fabra.

- *Chroma*: This feature is useful for analyzing music whose tuning approximates to the equal-tempered scale. It captures harmonic and melodic characteristics of music, while being robust to changes in timbre and instrumentation.
- *RMSE*: The root-mean-square energy (RMSE) is related to the loudness of the signal. It is useful for getting a rough idea about the loudness of a signal.
- *Spectral centroid*: This feature is a measure used in digital signal processing to characterise a spectrum. It indicates where the center of mass of the spectrum is located. Perceptually, it has a robust connection with the impression of brightness of a sound.
- *Zero crossing rate*: It is the number of times that the signal crosses the zero value in the buffer. It helps differentiating between percussive and pitched sounds. Percussive sounds will have a random ZCR across buffers, where pitched sounds will return a more constant value.
- *Spectral spread*: Indicates how spread the frequency content is across the spectrum. Corresponds with the frequency bandwidth. It can be used to differentiate between noisy (high spectral spread) and pitched sounds (low spectral spread).
- *Spectral rolloff*: It is the frequency below which is contained 99% of the energy of the spectrum. It can be used to approximate the maximum frequency in a signal.
- *Mel-Frequency Cepstral Coefficients* (MFCCs): As humans do not interpret pitch in a linear manner, various scales of frequencies were devised to represent the way humans hear the distances between pitches. The mel scale is one of them.

We applied a wrapper feature selection algorithm to select a subset of the original feature set. The resulting features (chroma, spectral rolloff and MFCCs) were used to train classifiers using machine learning algorithms for distinguishing compositions from different geographical locations.

3 Results

The accuracy (i.e. correctly classified instances percentage) obtained by both artificial neural networks (Chauvin, 1995) and decision trees algorithm (Quinlan, 1993) was 70% (baseline = 40%) using stratified 10-fold cross validation evaluation (see Table 1 for details). This result seems to indicate that the reduced number of features considered provide information about the acoustic characteristics of the musical pieces and that machine-learning algorithms are capable of using this information to distinguish the compositions at different geographical locations. It is worth noting that the two Finish Schools are from geographically distant regions with different cultural traditions. Interestingly, the German school compositions are more differentiable from the other two Finish schools than the two Finish schools. Table 2 shows the confusion matrix of the induced classifier obtained by applying the decision trees algorithm.

Table 1. Detailed accuracy by class (Finish School 1 [FS1], Finish School 2 [FS2], German School [GS]) and weighted average (WA)

TP Rate	FP Rate	Precision	Recall	F-Measure	ROCArea	Class
0.571	0.154	0.667	0.571	0.615	0.786	FS1
0.600	0.133	0.600	0.600	0.600	0.873	FS2
0.875	0.167	0.778	0.875	0.824	0.818	GS
WA 0.700	0.154	0.694	0.700	0.695	0.820	

Table 2. Confusion matrix of the Decision tree classifier

	Finish School 1	Finish School 2	German School
Finish School 1	57.4%	28.4%	14.2%
Finish School 2	40%	60%	0%
German School	0%	12.5%	87.5%

Analysis of the feature set showed that the most informative features for the obtained classifiers were MFCCs, spectral rolloff and chroma. In view of this preliminary results, it is worth exploring other acoustic features for training the classifiers and to include more data in the analysis.

4 Conclusions

We have presented a study involving the analysis of music materials composed by children in distinct geographical regions. We extracted acoustic features from the audio recordings, automatically selected a feature subset and then applied neural networks and decision trees algorithms to classify the recordings from different locations. We obtained an accuracy of 70%, well above chance level. This result seems to indicate that the reduced number of features considered provide information about the acoustic characteristics of the musical pieces and that machine-learning algorithms are capable of use this information to distinguish the compositions of different geographical locations. This preliminary results lead the way to extend this work by exploring more features and extending the training data.

References

- Tooby, J., & Cosmides, L. (1990). The past explains the present: Emotional adaptations and the structure of ancestral environments. *Ethology and Sociobiology*, 11, 375–424.
- Clayton, M. (2001). Introduction: towards a theory of musical meaning (in India and elsewhere). *British Journal of Ethnomusicology*, 10, 1–18.
- Cross, I. (2003). Music, cognition, culture and evolution. In I. Peretz & R. J. Zatorre (eds), *The cognitive neuroscience of music* (pp. 42–56). Oxford: Oxford University Press.
- Cross, I. (2006). The origins of music: Some stipulations on theory. *Music Perception*, 24, 79–82.
- Dissanayake, E. (2001). Antecedents of the temporal arts in early mother-infant interaction. In N. L. Wallin, B. Merker, & S. Brown (eds), *The origins of music* (pp. 389–410). Cambridge, MA: MIT Press.
- Dmitry Bogdanov, Nicolas Wack, Emilia Gomez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, Jose Zapata and Xavier Serra (2013). *Essentia: An Audio Analysis Library for Music Information Retrieval*, International Society for Music Information Retrieval Conference.
- Quinlan R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.
- Chauvin, Y. et al., (1995). *Backpropagation: Theory, Architectures and Applications*. Lawrence Erlbaum Assoc.

Two-step neural cross-domain experiments for full-page recognition of Mensural documents

Francisco J. Castellanos, Jorge Calvo-Zaragoza, and José M. Iñesta

Department of Software and Computing Systems
University of Alicante
Spain
`fcastellanos@dlsi.ua.es`

Abstract. Optical Music Recognition (OMR) is the research field that studies the music transcription from music score images into a digital format. Recently, this process has been formulated as a machine-learning problem, obtaining excellent results in controlled scenarios. However, a domain change could seriously affect performance. In this paper, we present a study of performance degradation in cross-manuscript experiments in two relevant steps of the OMR workflow. The results confirm this decrease, leaving room for enhancement with techniques as data augmentation, re-training or domain adaptation.

Keywords: Cross Domain · Music Recognition · Full-page Digitization.

1 Introduction

In the last years, Optical Music Recognition (OMR) [1] has been addressed as a machine-learning problem, being Deep Neural Networks (DNN) one of the well-known promising methods to process music documents through a generalizable strategy. Recently, it has been proved that it is possible to perform a full-page music recognition in only two steps, based on neural networks [4]: staff-region detection combined with end-to-end staff-level recognition. Results have demonstrated the goodness of the method, but it has been evaluated only using the same manuscript that was used to train the models.

A typical drawback of machine-learning methods is the lack of generalisation when the model predicts data from other domains different to those considered in the training process. This issue could be alleviated with straightforward strategies such as data augmentation [9] or re-training, but although they usually provide good results in controlled environments, they could not be enough in practice. In this paper, we contribute with a study on this issue, with a cross-domain evaluation for Mensural documents to determine whether there is room for improvement that could be achieved with strategies such as data augmentation or Domain Adaptation (DA) techniques [6, 8].

2 Methodology

In a recent work [4], a neural approach for full-page music transcription with only two steps was presented. That work shown that it is possible to transcribe music score images into a digital music sequence by combining two processes: a staff-region recognition addressed through a Selectional Auto-Encoder (SAE), which was already successfully used for layout analysis [5], followed by an end-to-end approach based on Convolutional Recurrent Neural Networks (CRNN) [2] to extract the music sequence from each staff detected in the previous step.

The method was evaluated with two corpora, obtaining excellent results when the domain considered is the same throughout the process. In this paper, we extend the experiments performed to study the loss of performance in cross-document evaluation, i.e. training the models with pages of a manuscript and predicting with documents belonging to another collection.

3 Corpora

We consider two datasets, already used in previous works:

- CAPITAN: *Missa* of 97 handwritten pages from the 17th century [3]. It contains 737 staves with 17 112 running symbols of 53 symbol categories.
- SEILS: Symbolically Encoded Il Laurro Secco consists of 150 printed pages from the 16th-century anthology of Italian madrigals *Il Lauro Secco* [7]. The piece contains 1 278 staves with 31 589 symbols within 33 possible categories.

Note that, although both collections are writtten in Mensural notation, CAPITAN is handwritten and SEILS is printed. Examples of each one are shown in Fig. 1.

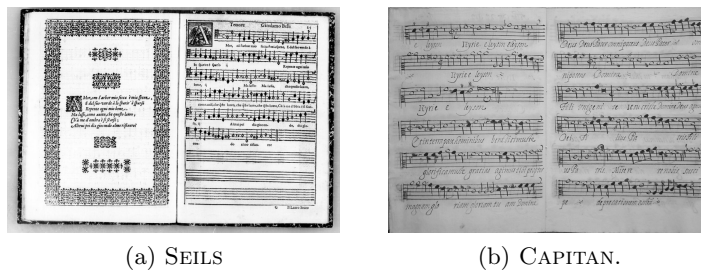


Fig. 1: Page examples of each dataset.

4 Staff-region recognition

As aforementioned, the first process we consider for extending experiments is the staff-region identification. This is performed by a SAE specialized in detecting staff areas. The results, included in Table 1, are presented in terms of F -score (F_1)¹ due to the unbalanced nature of data, precision, recall and Intersection

¹ We consider IoU of 55% or higher as *True Positive*, while lower figures are considered as *False Negative* or *False Positive*.

over Union (IoU), which provides a measure of the overlapping between the predicted staff regions and the ground-truth ones.

Table 1: Staff-retrieval average results in terms of F -score (F_1), precision ($Prec.$), recall ($Rec.$) and IoU represented in %.

Train \ Test	CAPITAN				SEILS			
	F_1	$Prec.$	$Rec.$	IoU	F_1	$Prec.$	$Rec.$	IoU
CAPITAN	99.8	99.8	99.8	81.1	72.3	73.9	73.3	63.4
SEILS	52.3	67.6	46.3	40.3	90.2	91.9	89.0	76.8

The results confirm the reduction in cross-domain situations of all the considered metrics. It could be highlighted the F -score figures from 99.8% to 52.3% for CAPITAN tests, and 90.2% to 72.3% for SEILS. Also, it should be noted the reduction in IoU from 81.1% to 40.3% for the handwritten manuscript, while SEILS is less affected. The engraving of the manuscript could be the reason for this difference. To complement these results, Fig. 2 shows the IoU histogram, whose overlapping in the cross-domain scenario is detrimental compared with the on-domain stage. Therefore, it could be concluded that the quality of the bounding boxes is negatively affected.

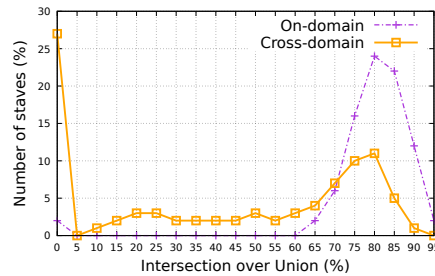


Fig. 2: Average histogram of staves predicted in the staff retrieval stage and ordered by IoU with a granularity of 5%.

5 End-to-end staff-level recognition

Once reported the performance of staff retrieval, in this section we focus in end-to-end staff-level recognition to retrieve the music sequence. These results are shown in terms of Symbol Error Rate (SER), which is computed as the ratio of editing operations to transform the predicted sequence to the expected (ground truth), being the lower, the better.

Table 2: End-to-end average results in terms of SER and represented in %.

Train \ Test	CAPITAN	SEILS
	SER	SER
CAPITAN	13.2	60.1
SEILS	79.3	4.4

As for the experiments, shown in Table 2, the change of domain is also highly disadvantageous, increasing the error figures for CAPITAN from 13.2% to 60.1% of SER, while the error rate in SEILS reaches 79.3% when the on-domain experiment yields 4.4%. The results indicate a high margin of improvement in all cross-domain cases, thereby being interesting as a research challenge.

6 Conclusions

OMR is a research field that has recently been formulated as a machine learning task, obtaining excellent results in controlled environments. However, this promising generalizable strategy, does not provide a solution in practice for cross-domain problems. We report cross-manuscript results for Mensural documents, concluding that the performance is seriously affected by the change of domain, increasing the error rate considerably. The results show a wide room for improvement, being Domain Adaptation a potential strategy to solve this issue.

Acknowledgements

This work was supported by the Spanish Ministry HISPAMUS project TIN2017-86576-R, partially funded by the EU. First author also acknowledges the support from “Programa I+D+i de la Generalitat Valenciana” through grant ACIF/2019/042.

References

1. Calvo-Zaragoza, J., Jr., J.H., Pacha, A.: Understanding optical music recognition. *ACM Comput. Surv.* **53**(4) (Jul 2020). <https://doi.org/10.1145/3397499>
2. Calvo-Zaragoza, J., Toselli, A.H., Vidal, E.: Handwritten music recognition for mensural notation with convolutional recurrent neural networks. *Pattern Recognition Letters* **128**, 115–121 (2019)
3. Calvo-Zaragoza, J., Toselli, A.H., Vidal, E.: Handwritten music recognition for mensural notation: Formulation, data and baseline results. In: 14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9–15. pp. 1081–1086 (2017)
4. Castellanos, F.J., Calvo-Zaragoza, J., Inesta, J.M.: A neural approach for full-page optical music recognition of mensural documents. In: Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, September 23–27, 2020 (2020)
5. Castellanos, F.J., Calvo-Zaragoza, J., Vigliensoni, G., Fujinaga, I.: Document analysis of music score images with selectional auto-encoders. In: Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23–27, 2018. pp. 256–263 (2018)
6. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495* (2014)
7. Parada-Cabaleiro, E., Batliner, A., Schuller, B.W.: A diplomatic edition of il lauro secco: Ground truth for OMR of white mensural notation. In: Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4–8, 2019. pp. 557–564 (2019)

8. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7167–7176 (2017)
9. Wong, S.C., Gatt, A., Stamatescu, V., McDonnell, M.D.: Understanding data augmentation for classification: When to warp? In: 2016 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2016, Gold Coast, Australia, November 30 - December 2, 2016. pp. 1–6 (2016)

Feature Engineering for Genre Characterization in Brazilian Music

Bruna Wundervald¹[0000-0001-8163-220X]

Maynooth University, bruna.wundervald@mu.ie

Abstract. Many factors are involved in the definition of music genres, making it an active area of research. This work focuses on verifying the connection between harmonic information and genre specification in Brazilian music, through the evaluation of feature importance in machine learning models. We construct four different sets of manually engineered harmonic features and assess how they relate to the accuracy of the models, as well as explore the mistakes made by the model in each genre. We identified the most relevant features to be the harmonic ones, followed by external features such as popularity and the proportions of the most common chord transitions in each song.

Keywords: chord features · feature importance · genre characterization.

1 Introduction

Genre is an important form of classifying songs, as they facilitate the search for music, and users even prefer to use genre instead of other metrics when looking for new music [9]. However, many factors are involved in the configuration of a music genre, such as style, historical context, and harmonic structures [3], making the definition of each genre unclear. Inconsistencies and blurriness in the definition of musical genres pose an important problem in various aspects of music studies and is an active area of research in MIR. For such reasons, the focus of this work is towards verifying the connection between harmonic information and genre specification in Brazilian music through the evaluation of feature importance in machine learning models. In addition, as [4] and [6] observed, mid-level music features such as chords configure a rich resource of information regarding genres. The chords sequence of a song fully describes its harmonic progression and it represents a meaningful part of the total music structure. With that, in this work, we also focus on the use of symbolic chords data and in manually extracting harmonically related features for genre classification, representing the chords structures in different and meaningful forms.

Related work has been done all of the usual representations of music data, for example, [14], [18], [1] and [17], which focused in music genre classification using audio extracted features. As for text data related to music, [11] presents a discussion about the characterization of genres through song lyrics. In [5], the authors introduced a vector based representation for chords sequences, bringing

light to an effective way to extract information about symbolic chords data. A similar problem to ours was studied in [15], which focused on harmonic features for genre classification.

2 Definitions

2.1 Data

The data was extracted from the Cifraclub website (<https://www.cifraclub.com.br/>), an online collaborative page of music-sharing, via the `chorrrds` [19] package for R [16]. Though the use of user-inputted (or crowd-sourced) music chords is not very common in MIR due to the possible inconsistencies, recent literature [12, 7] has been showing its value to the community. In total, 8 music genres were used: Reggae, Pop, Forró, Bossa Nova, Sertanejo, MPB, Rock and Samba, all good representatives of the Brazilian music, and from these genres, 106 different artists were available in the online platform, for which the chords and keys for 8339 different songs were collected. Complementary features about the release year and popularity were obtained with the aid of the well-known Spotify *API*.

2.2 Manually Extracted Features

In this work, we emphasized on obtaining various interpretable summary features from the chords, to make use of more information than only the symbolic form of the chords. The engineered features were separated into four thematic groups, organized as the **First set, triads and simple tetrads**: percentage of suspended chords (e.g. Gsus), of chords with the seventh (e.g. C7), of minor chords with the seventh (e.g. Em7, C#m7), of minor (e.g. Em, C#m), of diminished (e.g. B°), and of augmented (e.g. Baug) chords. **Second set, dissonant Tetrads**: percentage of chords with the fourth (e.g. D4), the sixth (e.g. E6), the ninth (e.g. G9), with the major seventh (e.g. F7+, Am7+), with a diminished fifth (e.g. C5- or C5b) and with an augmented fifth (e.g. C5+ ou C5#). **Third set, main chord transitions**: percentage of the first, second, and third most common chord transitions in the song. **Fourth set, miscellany**: popularity, total of non-distinct chords, year of album release, indicator of the key of the song being the same as the most common chord, percentage of chords with varying bass (e.g. C/E, C/G, C/Bb), mean distance of the root note to 'C' in the circle of fifths, mean distance of the root note to 'C' in semitones, absolute number of the most common chord.

2.3 Machine Learning Algorithm

We used the popular Random Forest [2] model, which is mainly characterized by being a tree ensemble that only allows a random subset m of the features to be the candidates for a split, helping to create uncorrelated trees. This bagged ensemble can be written as $\hat{f}(\mathbf{x}) = \sum_{n=1}^{N_{tree}} \frac{1}{N_{tree}} \hat{f}_n(\mathbf{x})$, where \hat{f}_n corresponds to the n -th tree.

3 Results

Table 1. Goodness of fit for the four models: overall accuracy with lower and upper bounds and Kappa statistic with the respective p-value.

Model	Accuracy	L.B.	U.B.	Kappa	P-Value
Model 1	0.53	0.51	0.55	0.37	< 0.0001
Model 2	0.57	0.54	0.59	0.42	< 0.0001
Model 3	0.59	0.56	0.60	0.44	< 0.0001
Model 4	0.62	0.60	0.64	0.49	< 0.0001

Our target variable here is the music genres, and the predictors are the engineered features. There is extensive literature in genre classification and we do not intend to claim that this is a better model than the others, as our primary goal is to observe how the features relate to the accuracy rather than obtaining the best accuracy possible. Four models were fitted in a nested fashion, with each new model being added with one of the features sets described before. Table 1 shows that, for all different models, there is evidence of their accuracy being significantly higher the non-information classification rate. The addition of feature sets progressively increases the accuracy of the models, evidencing that the 4 sets of features are informative to predict the genres. The increase is seemingly uniform: to each new set of variables added, the increase is about 3%.

Table 2. Confusion matrix for the model with all the features.

	Bossa Nova	Forró	MPB	Pop	Reggae	Rock	Samba	Sertanejo
Bossa Nova	0.28	0.00	0.40	0.00	0.00	0.05	0.16	0.12
Forró	0.00	0.00	0.12	0.00	0.00	0.12	0.10	0.65
MPB	0.01	0.00	0.59	0.00	0.00	0.11	0.13	0.15
Pop	0.00	0.00	0.13	0.00	0.00	0.28	0.15	0.44
Reggae	0.00	0.00	0.25	0.00	0.08	0.46	0.08	0.12
Rock	0.00	0.00	0.16	0.00	0.00	0.43	0.05	0.35
Samba	0.01	0.00	0.20	0.00	0.00	0.03	0.66	0.10
Sertanejo	0.00	0.00	0.02	0.00	0.00	0.07	0.02	0.89

Figure 1 shows that the first set of features is the most informative one, meaning that with the basic chords information we can already obtain good results in terms of informing the model about the genres. The external variables, such as the year and popularity, got a high rank in the plot, showing how the Spotify features are also pertinent. The position of the transitions and distances variables strengthen the idea of harmonic characteristics being important to discriminate

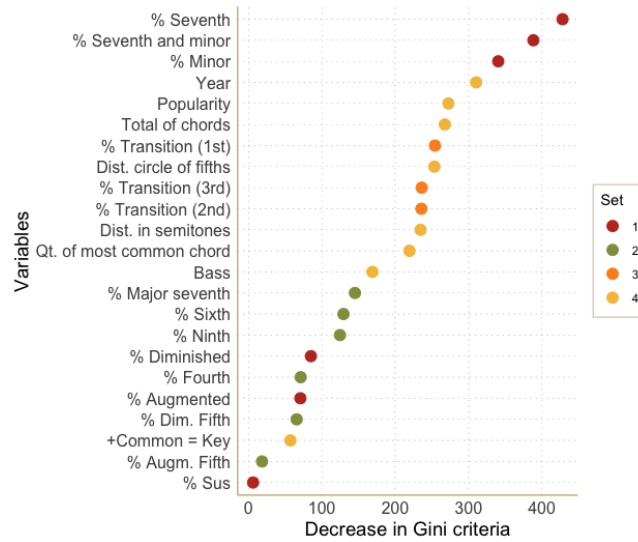


Fig. 1. Importance plot for the fourth model with all the considered features. The top part of the plot is dominated by harmonic features.

music genre. From Table 2, we can see that there is considerable confusion between MPB and Bossa Nova, highlighting their known harmonic similarities. The same happens to Forró, Sertanejo and Pop, which are music genres with a similar origin and, in general, more elementary harmonic structures.

4 Conclusions

With our results, we conclude that manually engineered harmonic features can be useful to characterize Brazilian music genres. More than just predicting music genres, which does not have a consensual utility in the literature, we are interested in inferring which harmonic features are informative for the definition of genres. In our case, the most discriminative features are the percentage of chords with the seventh note, of minor chords with the seventh note, of minor chords, the year of release of the songs, the popularity and the behavior of the most common chord transitions. Apart from that, though our work was limited to one geographic region, we believe that our insights can be extended to other types of music that influenced or were influenced by the genres considered here, such as Jazz, Pop, and Rock music.

The next steps of this work include specially the engineering of the new variables and applying different algorithms, such as deep learning models [8] and naive Bayes models [10], as in [1] and [13], though they might have less interpretable results. In a different sense, we would also like to explore more the use of crowd-sourced data and the relationship between song popularity and the precision of this type of data.

References

1. Bahuleyan, H.: Music genre classification using machine learning techniques. CoRR **abs/1804.01149** (2018), <http://arxiv.org/abs/1804.01149>
2. Breiman, L.: Random forests. Machine Learning (2001). <https://doi.org/10.1023/A:1010933404324>
3. Caldas, W.: Iniciação à Música Popular Brasileira, vol. 1 (2010)
4. Cheng, H.T., Yang, Y.H., Lin, Y.C., Liao, I.B., Chen, H.H.: Automatic chord recognition for music classification and retrieval. In: 2008 IEEE International Conference on Multimedia and Expo. pp. 1505–1508. IEEE (2008)
5. Chuan, C.H., Agres, K., Herremans, D.: From context to concept: exploring semantic relationships in music with word2vec. Neural Computing and Applications **32**(4), 1023–1036 (2020)
6. Corrêa, D.C., Rodrigues, F.A.: A survey on symbolic data-based music genre classification. Expert Systems with Applications **60**, 190–210 (2016)
7. Koops, H.V., de Haas, W.B., Bransen, J., Volk, A.: Automatic chord label personalization through deep learning of shared harmonic interval profiles. Neural Computing and Applications **32**(4), 929–939 (2020)
8. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)
9. Lee, J.H., Downie, J.S.: Survey of music information needs, uses, and seeking behaviours: preliminary findings. In: ISMIR. vol. 2004, p. 5th. Citeseer (2004)
10. Murphy, K.P., et al.: Naive bayes classifiers. University of British Columbia **18**, 60 (2006)
11. Neuman, Y., Perlovsky, L., Cohen, Y., Livshits, D.: The personality of music genres. Psychology of Music (2016). <https://doi.org/10.1177/0305735615608526>
12. Odekerken, D., Koops, H.V., Volk, A.: Decibel: Improving audio chord estimation for popular music by alignment and integration of crowd-sourced symbolic representations. arXiv preprint arXiv:2002.09748 (2020)
13. Oramas, S., Nieto, O., Barbieri, F., Serra, X.: Multi-label music genre classification from audio, text, and images using deep features. CoRR **abs/1707.04916** (2017), <http://arxiv.org/abs/1707.04916>
14. Pampalk, E., Flexer, A., Widmer, G.: Improvements of Audio-Based music similarity and genre classification. In: ISMIR (2005). <https://doi.org/10.1007/s10115-013-0641-y>
15. Pérez-Sancho, C., Rizo, D., Iesta, J.M., De León, P.J., Kersten, S., Ramirez, R.: Genre classification of music by tonal harmony. Intelligent Data Analysis (2010). <https://doi.org/10.3233/IDA-2010-0437>
16. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2018), <https://www.R-project.org/>
17. Scaringella, N., Zoia, G., Mlynek, D.: Automatic genre classification of music content. IEEE Signal Processing Magazine (2006). <https://doi.org/10.1109/MSP.2006.1598089>
18. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing (2002). <https://doi.org/10.1109/TSA.2002.800560>
19. Wundervald, B.: The chorrrds package for extraction of music chords data in r (2018), <https://github.com/r-music/chorrrds>

A dataset and classification model for Malay, Hindi, Tamil and Chinese music

Fajilatun Nahar¹, Kat Agres², Balamurali BT¹, and Dorien Herremans¹

¹ Singapore University of Technology and Design, Singapore

fajilatun.nahar@mymail.sutd.edu.sg

² National University of Singapore, Singapore

Abstract. In this paper we present a new dataset, with musical excerpts from the three main ethnic groups in Singapore: Chinese, Malay and Indian (both Hindi and Tamil). We use this new dataset to train different classification models to distinguish the origin of the music in terms of these ethnic groups. The classification models were optimized by exploring the use of different musical features as the input. Both high level features, i.e., musically meaningful features, as well as low level features, i.e., spectrogram based features, were extracted from the audio files so as to optimize the performance of the different classification models.

Keywords: Music Classification · Ethnic Groups · Machine Learning

1 Introduction

Singapore is a cultural melting pot, with a majority of Chinese, Malay and Indian individuals. It is thus no surprise that Singaporean music is influenced by several different ethnical groups. The earliest form of music in Singapore was traditional Malay music [14], which came from the original settlers of Singapore. They are now the second largest ethnic group in Singapore [16]. Then came the Portuguese influence from the colonial occupation, followed by Chinese and Indian music from the immigrants of those countries [14]. Decades of rich political and cultural history of Singapore has established the current tastes and genres of music in Singapore [10]. In this paper, we create a dataset of music fragments of the three largest ethnical influences in Singapore, namely, Chinese, Malay, and Indian. This allows us to develop machine learning models that can estimate the probability of a song belonging to a certain ethnical group. In future research, these newly developed models will be useful to analyse typical Singaporean songs such as the National Day Songs.

Over the last decade, significant strides have been made regarding audio classification models for mood/emotion [11, 4, 13], genre [15, 6], hit prediction [9] and other topics. Most related to this research is the work on folk tune classification [5, 3]. Here, we focus on contemporary music from different Asian ethnical groups.

In the next section, we will discuss the dataset that we have gathered, followed by the extracted features and developed classification models in Section 3. The performance of our classifiers is presented in Section 4 and the final conclusion is presented in Section 5.

2 Dataset creation

We used the Spotify API³ to retrieve a list of songs for each of our ethnical groups. The songs were manually curated by the first author, using search terms in the Spotify API. General search terms like ‘Hindi songs’, ‘Chinese songs’, ‘Malay songs’ and ‘Tamil songs’ were used, as well as names of popular singers of that specific ethnical group. A total of 15,725 songs were downloaded using the API, of which 3,146 were Chinese songs, 507 Malay songs, 6,729 Hindi songs, and 5,343 Tamil songs. We downloaded the first 30 seconds of the selected songs, some of which are instrumental songs, some contain only vocals, and some are a mix of both. Of these songs, a total of 260 low-level features and 98 high-level features were extracted using Essentia [2] and OpenSMILE [7] respectively. For high-level features, six of the features were categorical features, so those features were one hot encoded, which increased the feature space to a total of 127 features. For low-level features, temporal data was collected in 0.5 seconds frames, totalling 58 frames per song. **These features were averaged for each song.** A detailed description of the features and the dataset itself is available online⁴

Given the large number of extracted features, we do a preliminary exploration of which feature subset is most efficient in the next section.

3 Classification models

There exist many types of classification algorithms that have shown to be effective for audio classification tasks. It is not the intention of this investigation to develop novel architectures or implement complex neural network structures. Instead, we focus on a very influential factor: input features. As per [1], features greatly influence the performance of models. In this research, we hence focus on comparing different input representations (both high and low level music features) in basic, fast, and efficient machine learning models that have proven their efficacy in audio classification: logistic regression, k-nearest neighbours (k-NN), support vector machines (SVM) (with Grid search), and random forest.

The dataset was split into a training and test set with a ratio of 80:20. These models were tested using different feature subsets. These subsets can contain different types of features, and might include a feature selection mechanism, as described in Table 1. This analysis reveals the most effective features for ethnical origin classification on our new dataset.

Two feature selection methods were implemented: 1) A one-way ANOVA test is used to perform the filter method [8] where the p -value is calculated for each feature. Features with a p -value of less than 0.05 are taken into consideration for further analysis. 2) The other technique is the wrapper method [12], where backward elimination was performed by taking subsets of the features to create models using logistic regression. The accuracy of this model was examined and, using an iterative procedure, features were removed. The feature selection process will be stopped when the classifier delivers the best performance.

³ <https://developer.spotify.com/>

⁴ <http://dorienherremans.com/sgmusic>

4 Experiments and results

We set up a preliminary experiment to analyse the influence of different feature representations on the classifier performance. We explored different combinations of high/low level features, with or without feature selection, thus forming Subsets of our data. We should note that these subsets are imbalanced, hence we include the class weighted AUC in the results in Table 1.

Table 1. Subset description and model results

Subset	No of features	Feature type	Feature selection method	Best Model	AUC	Accuracy
1	260	low-level	NA	SVM	0.94	0.79
2	127	high-level	NA	RF	0.88	0.70
3	387	high & low	NA	SVM	0.94	0.79
4	1,820	low-level	NA	SVM	0.93	0.77
5	111	low-level	wrapper	SVM	0.94	0.80
6	82	high-level	wrapper	RF	0.88	0.70
7	182	low-level	filter	SVM	0.95	0.81
8	67	high-level	filter	RF	0.88	0.69
9	92	low-level	filter+wrapper	SVM	0.95	0.81
10	49	high-level	filter+wrapper	RF	0.86	0.69

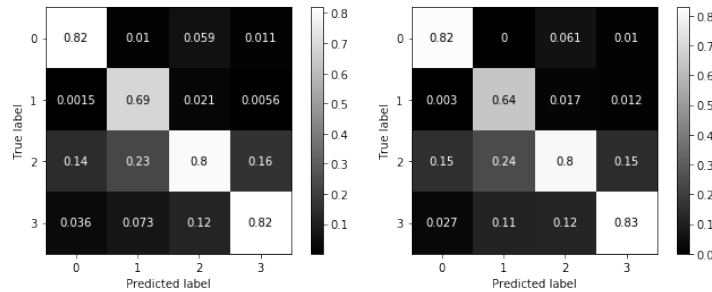


Fig. 1. Confusion matrices of two best performing models; left: Subset 9; right: Subset 7. Label 0 is Chinese; 1 is Malay; 2 is Hindi; 3 is Tamil.

The SVM models using Subset 7 and 9 yielded the best AUC score of 95% and an accuracy of 81% on the test data. Both of these subsets contain only low-level features, and were reduced using feature selection methods. The confusion matrices in Fig. 1 also reveal a very similar performance of these two models. When comparing these two best performing models, we can conclude that Subset 9 is the more desired representation, because it contains less features, and as a result the training time is faster.

5 Conclusions and future work

We have gathered a dataset of 30s musical fragments together with 98 high and 260 low level musical features from four different ethnical origins. We have used this data to train relatively well performing classification algorithms. In an experiment, these classifiers perform best when using low-level audio features with feature selection as the input. In future research, we aim to further expand and visualise the songs of our dataset and make the models more robust, after which we can use them to explore the ethnical origin/influence of typical Singaporean music such as the National Day Songs.

References

- [1] B. Balamurali et al. “Toward robust audio spoofing detection: A detailed comparison of traditional and learned features”. In: *IEEE Access* 7 (2019), pp. 84229–84241.
- [2] D. Bogdanov et al. “ESSENTIA: an open-source library for sound and music analysis”. In: *Proc. of the 21st ACM Int. conf. on Multimedia*. 2013, pp. 855–858.
- [3] W. Chai and B. Vercoe. “Folk music classification using hidden Markov models”. In: *Proc. of Int. conf. on artificial intelligence*. Vol. 6. 6.4. sn. 2001.
- [4] K. Cheuk, K. Agres, and D. Herremans. “The impact of Audio input representations on neural network based music transcription”. In: *Proc. of the Int. Joint conf. on Neural Networks (IJCNN)*. Glasgow, 2020.
- [5] D. Conklin. “Multiple viewpoint systems for music classification”. In: *Journal of New Music Research* 42.1 (2013), pp. 19–26.
- [6] D. C. Corrêa and F. A. Rodrigues. “A survey on symbolic data-based music genre classification”. In: *Expert Syst. Appl.* 60 (2016), pp. 190–210.
- [7] F. Eyben and B. Schuller. “openSMILE: The Munich open-source large-scale multimedia feature extractor”. In: *ACM SIGMultimedia Records* 6.4 (2015), pp. 4–13.
- [8] I. Guyon and A. Elisseeff. “An introduction to variable and feature selection”. In: *Journal of machine learning research* 3.Mar (2003), pp. 1157–1182.
- [9] D. Herremans, D. Martens, and K. Sörensen. “Dance hit song prediction”. In: *Journal of New Music Research* 43.3 (2014), pp. 291–302.
- [10] L. Kong. “The invention of heritage: popular music in Singapore”. In: *Asian Studies Review* 23.1 (1999), pp. 1–25.
- [11] C. Laurier, J. Grivolla, and P. Herrera. “Multimodal music mood classification using audio and lyrics”. In: *Int. conf. on Machine Learning & Appl.* IEEE. 2008, pp. 688–693.
- [12] S. Maldonado and R. Weber. “A wrapper method for feature selection using support vector machines”. In: *Information Sciences* 179.13 (2009), pp. 2208–2217.
- [13] B. G. Patra, D. Das, and S. Bandyopadhyay. “Multimodal mood classification of Hindi and Western songs”. In: *J. Intell. Inf. Syst.* 51.3 (2018), pp. 579–596.
- [14] L. M. Perera and A. Perera. “Music in Singapore: From the 1920s to the 2000s”. In: *National Library Board, Singapore* (2010).
- [15] G. Tzanetakis and P. Cook. “Musical genre classification of audio signals”. In: *IEEE Tran. on speech and audio processing* 10.5 (2002), pp. 293–302.
- [16] *What are the racial proportions among Singapore citizens?* URL: <https://www.gov.sg/article/what-are-the-racial-proportions-among-singapore-citizens>. (accessed: 6.08.2020).

Beat Tracking from Onset Streams Using LSTM Neural Networks

Aggelos Gkiokas

University Pompeu Fabra, Music Technology Group
aggelos.gkiokas@upf.edu

Abstract. In this paper we present a method for detecting music beats from a stream of onsets. Onsets are represented as a time-frequency piano-roll like matrix, that are grouped into frequency bands. To estimate the beats from the onset streams, we utilize a Long Short Term Memory Neural Network, which has been used successfully in tracking the beats from audio. The LSTM takes as input the onset streams and learns a beat activation function, from which the beats are extracted using an HMM-based post processing method. The proposed architecture is trained and evaluated on a classical MIDI collection and an audio dataset that contain both onset and beat annotations, achieving a good beat tracking accuracy.

Keywords: Beat Tracking · Onset Streams · LSTM Networks.

1 Introduction

Automatic estimation of beats is an important task in the Music Information Retrieval (MIR) domain. Early beat tracking methods (late 90's, 00's) were focused on both audio and MIDI input. MIDI files were popular and many researchers chose to use MIDI as the basic representation of the music signals. For example in [14] a probabilistic model for handling MIDI events in an online form is proposed. Dixon [3] presented Beat Root, a method incorporating a probabilistic model for extracting the beats from audio which implements an explicit onset detection step followed by Inter-Onset-Interval (IOI) analysis that makes it capable to be applied to MIDI streams. Raphael [12] proposes a probabilistic model for extracting beats by a "rhythmic parsing" of sequences of times. Similarly, Goto [5] proposed a method for handling audio signals. Similarly to Beat Root, it computes discrete onset times from an onset salience function computed on seven frequency bands as a feature extraction step. Hainsworth and Macleod [7] introduced particle filtering on onset times to estimate the beats of an onset stream. In [15] the authors proposed adaptive oscillators in MIDI signals, while in [13] a beat tracker in the context of performer-sequencer synchronization. A more recent work dealing with MIDI can be found in [6], where a beat tracking method in the context of performance MIDI files is presented.

As the years were passing, MIDI-based rhythm analysis systems were gradually disappearing, and more methods that were dealing directly with audio were

proposed. Currently, the state-of-the-art on beat tracking consists of deep learning methods acting on audio files as in [1, 9, 4, 11]. In contrast to the majority of the literature, this paper deals with the extraction of beats from onsets, using a Long Short Term Memory LSTM network. This choice is justified by the fact that LSTMs have been successful for both onset and beat estimation tasks. Our main contribution is the revision of a problem that has lost attention over the last years despite its importance, and we provide a robust solution based on modern deep based neural architectures.

2 Method Description

Figure 1 illustrates an overview of the proposed method. Onset times which can be either derived from a MIDI file or automatically computed from audio along with the pitch (MIDI number) are represented in a piano-roll binary sparse matrix $X[n, k]$ which is one if there is an onset at time step n and frequency that corresponds to MIDI number k . X is processed by a square filterbank F of M bands and the output $B[n, m]$ represents the number of onset events in time n for the filterbank band m . At next, the onset features B are fed to an bi-directional LSTM network for depth L and size K in a sequence-to-sequence learning schema, that outputs a two-dimensional Beat Activation Function (BAF) over time $O[n, 2]$ indicating the probability that a time instant being a beat/non-beat. Finally, the output of the LSTM network is processed by an HMM-based system to extract an optimal beat sequence from the BAF as proposed in [8].



Fig. 1. Overview of the proposed method.

3 Method Evaluation

The main difficulty in evaluating the proposed method is that there are not publicly available algorithms and datasets making it impossible to reproduce their results or to compare with them. To provide an evaluation as fair and as transparent as possible, we will report results of the proposed method together with the best performing state-of-the-art algorithm that was presented recently in [11]. Their results are not directly comparable, since [11] is an audio-based method trained on different datasets, but can provide an sketch of the potential of the proposed method.

Table 1. Overall results of the proposed method with comparison to best beat tracking method

Dataset	Proposed Method					Madmom [11]				
	F	CML _c	CML _t	AML _c	AML _t	F	CML _c	CML _t	AML _c	AML _t
KDF	69.22	34.34	42.7	53.17	66.14	58.1	32.47	40.23	49.13	60.73
MNet	53.0	14.1	21.4	32.6	48.4	50.78	19.7	27.67	33.16	47.74
4x22	49.7	2.1	14.4	8.9	29.9	52.22	3.43	7.87	13.14	47.63

We consider three datasets, containing mainly classical music and can be considered as difficult datasets, namely **Kunst der Fug** that contains 12141 MIDI files from classical composers, **MusicNet** as presented in [16] that is a collection of 330 classical music recordings annotated with onset times and frequencies and **Vienna 4x22 Corpus**[16] that contains audio with accurate aligned MIDI of performances of 22 professional pianists for four solo piano pieces. For evaluation we use the the standard F-measure together with the continuity based measures CML_c, CML_t, AML_c and AML_t [2]. Onset feature representation is computed on frame a rate of 100 f/s . After experimentation we chose L and size K for the network depth and size and the network was trained following a 8-fold cross validation strategy and by minimizing the Binary Cross Entropy with the Stochastic Gradient Descent with a 0.9 momentum and with an adjustable learning rate with warm restarts [10]. We applied L_2 weight penalization a batch size of 1.

Table 1 presents the performance of the proposed method compared to the CNN-based beat tracking method [11] that gives an overview of how the state-of-the-art performs on these datasets. Here has to be mentioned that these datasets have not been used in the past for beat tracking. For evaluating [11] on the KDF dataset, all MIDI files were synthesized with the TiMidity++¹ software. To ensure fairness of the comparison we compared the performance of [11] on both synthesized MIDI and audio files for the MusicNet and 4x22 datasets and observed no significant differences on the performance.

Overall results indicate that the proposed method achieves results comparable or even better compared to the baseline state-of-the-art method and demonstrate the strong relation between beat positions and onsets. Regarding the KDF dataset the proposed method achieves a good performance of more than 69% based on the F-measure and 66% based on the AML_t measure. A relatively good performance is achieved also on the MusicNet dataset. For the 4x22 Vienna Corpus the performance is poor, but this can be justified by the fact that we used a one-piece-out cross validation approach (4 pieces on the whole dataset).

4 Conclusion

In this paper we proposed the revision of a problem that has lost attention over the last years despite its importance and we presented a method for extracting the beat from onset streams. Evaluation results indicate that the onset times are

¹ <http://timidity.sourceforge.net/>

a dense and rich source of information which can be potentially used to train a model that determines the beats faster and more robustly than using audio. Moreover we provided an experimental protocol that can provided insights of the performance of the proposed method in the absence of other reproducible methods or open datasets for this task. Without being valid to compare directly to audio based state-of-the-art methods, results indicate that there is a lot of potential to continue work in this direction.

References

1. Böck, S., Krebs, F., Widmer, G.: A multi-model approach to beat tracking considering heterogeneous music styles. Citeseer
2. Davies, M.E., Degara, N., Plumbley, M.D.: Evaluation methods for musical audio beat tracking algorithms. Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06 (2009)
3. Dixon, S.: Evaluation of the audio beat tracking system beatroot. *Journal of New Music Research* **36**(1), 39–50 (2007)
4. Gkiokas, A., Katsouros, V.: Convolutional neural networks for real-time beat tracking: A dancing robot application. In: ISMIR. pp. 286–293 (2017)
5. Goto, M.: An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research* **30**(2), 159–171 (2001)
6. Grohganz, H., Clausen, M., Müller, M.: Estimating musical time information from performed midi files. In: ISMIR. pp. 35–40 (2014)
7. Hainsworth, S.W., Macleod, M.D.: Particle filtering applied to musical tempo tracking. *EURASIP Journal on Advances in Signal Processing* **2004**(15), 927847 (2004)
8. Korzeniowski, F., Böck, S., Widmer, G.: Probabilistic extraction of beat positions from a beat activation function.
9. Krebs, F., Böck, S., Widmer, G.: An efficient state-space model for joint tempo and meter tracking.
10. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
11. MatthewDavies, E., Böck, S.: Temporal convolutional networks for musical audio beat tracking. In: 2019 27th European Signal Processing Conference (EUSIPCO). pp. 1–5. IEEE (2019)
12. Raphael, C.: Automated rhythm transcription. In: ISMIR. vol. 2001, pp. 99–107 (2001)
13. Robertson, A., Plumbley, M.: B-keeper: A beat-tracker for live performance. In: Proceedings of the 7th international conference on New interfaces for musical expression. pp. 234–237 (2007)
14. Rosenthal, D., Goto, M., Muraoka, Y.: Rhythm tracking using multiple hypotheses. In: Proceedings of the International Computer Music Conference. pp. 85–85. Citeseer (1994)
15. Sethares, W.A., Arora, R.: Equilibria of adaptive wavetable oscillators with applications to beat tracking. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP’07. vol. 4, pp. IV–1301. IEEE (2007)
16. Thickstun, J., Harchaoui, Z., Kakade, S.: Learning features of music from scratch. arXiv preprint arXiv:1611.09827 (2016)