MML 2017

10th International Workshop on Machine Learning and Music

6.10.2017, Barcelona

Proceedings

Rafael Ramirez
Pompeu Fabra University, Spain

Darrell Conklin
University of the Basque Country, Spain

José Manuel Iñesta
University of Alicante, Spain

Contents

Neural Correlates of bow learning technique

Angel Blanco[1] and Rafael Ramírez[1]

[1] Music and Machine Learning Lab, Universitat Pompeu Fabra, Barcelona
lncs@springer.com

Abstract. In this work we want to study the process of learning a musical instrument through the use of audio descriptors and EEG. Twelve subjects participated in our experiment. Subjects were divided into two groups: a group of people who has never played the violin before (six subjects) and a group of experts (more than six years playing the violin). Participants were asked to perform a violin exercise during eighteen trials while the corresponding audio to each trial was recorded together with their EEG activity. Beginners showed significant differences between the beginning of the session and the end corresponding to an improve in the quality of the sound recorded while experts maintained their results. On the other hand, beginners showed more power in the High Beta frequency band (21-35Hz) than experts although the power values decreased during the session correlated with an improvement in the scores of the exercise.

Keywords: EEG, Learning, Music.

1 Introduction

Previous research has investigated the presence of biomarkers during human sensorimotor learning using EEG. For instance, it has been observed that linear and bilateral EEG alpha, as well as high theta increases in power, correlated with enhanced kinematics in participants during the performance of a visuomotor task which required learning and adaptation, while the control group did not show variations in kinematic and electrophysiological parameters [1].

The aim of this work is to find EEG biomarkers associated to different cognitive states during the process of learning a musical instrument, taking the violin as a case study. For that purpose we have used audio descriptors to track the quality of the sound generated by beginners during their learning process and at the same time we record their EEG activity. We also recruited a group of experts violinists (experience > 3 years) to compare both results.

2 Materials & Methods

2.1 Participants

Twelve adults participated in the study. Participants granted their written consent and procedures were positively evaluated by the CIREP, Barcelona, Spain, under the reference number X. Participants were asked to fill a form with questions regarding their level of musical studies and their primary instruments. Only participants who had never played the violin will conform the beginners group (BG), while participants with a high-level profile in violin playing conformed the expert group (EG). Participants were recruited in person at the university campus. Before starting, participants received a written consent form and were informed about their task, the experimental procedures and their right to withdraw from the experiment at any moment. To proceed with the study each participant must have agreed to participate and signed the corresponding consent form.

2.2 Materials

The Emotiv EPOC EEG system [2] was used for acquiring the EEG data. It consists of 16 wet saline electrodes, providing 14 EEG channels, and a wireless amplifier. The electrodes were located at the positions AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4 according to the international 10–20 system. The data were digitized using the embedded 16-bit ADC with 128 Hz sampling frequency per channel and sent to the computer via Bluetooth. The impedance of the electrode contact to the scalp was visually monitored using Emotiv Control Panel software. We collected the data using the OpenViBE platform [3]. The data was processed in EEGLAB [10] under the Matlab environment [4].

The sound of each violin trial was recorded using a Zoom recorder [5] and processed later in Matlab using the yin algorithm [6], which is a commonly used pitch detection algorithm based upon autocorrelation, to extract audio features in order to assess the quality of the audio produced by the subjects.

2.3 Method

Participants will be divided in two different groups: six total beginners in violin playing and six experts. First, participants filled a questionnaire with questions related to their musical ability. Then, participants belonging to the beginner group were shown a ten minutes instructional video on stance and violin position found in the web (reference web), and were asked to play eighteen trials consisting of four up and down bowing movements (playing the A open string), making clear to the participants that their main objective was achieving a stable tone and dynamics. Participants also watched a video reference of an expert executing the same exercise. Then, participants were asked to initiate a trial and to start producing sound with the instrument after the indication of the experimenter. Participants were free to take as many breaks as they wanted through the experiment although, every five trials participants had the opportunity to review as many times as they wanted the reference expert video.

EEG power computation. Before starting, participants imitated the movement and gestures of the exercise, holding the violin and moving their right arm without using a the bow and therefore without generating any kind of sound. The EEG activity was recorded from each one of these three trials in order to be used as a baseline reference to compute later the ERD/ERS for each real trial.

For each subject and each single-trial, the power spectral density (PSD) is computed from each electrode using Welch's overlapped segment averaging estimator. The power of eleven frequency bands were extracted corresponding to Delta (1-4Hz), Theta(4-8Hz), Alpha(8-13Hz), Beta(13-24Hz) and Gamma(30-50Hz), and the low Theta(4-5Hz), Alpha(8-10Hz), Beta(14-20Hz) and the high Theta(6-8Hz), Alpha(11-13Hz), Beta(21-35Hz) components of the bands. Each power value was standardized using the ERD/ERS equation (1). We also computed, as extra descriptors, the average power values of the right (AF4,F4,F8,FC6) and left (AF3,F3,F7,FC5) frontal lobes together with the average values of both frontal hemispheres and another one with the average value of all the electrodes for each band.

$$ERD/ERS(\%)=\frac{baseline\ interval\ band\ power-test\ interval\ band\ power}{baseline\ interval\ band\ power}*100 \quad (1)$$

Extraction of audio features. Generated violin sound was recorded for each trial and processed independently. First, we extracted sound descriptors from the audio signal of each trial using the Yin algorithm implementation in Matlab [7]. This Matlab implementation first windows the signal, using a windows size which depends in the sample rate and the minimum frequency (30Hz by default), and for each windows it computes three different parameters which are: the fundamental frequency in octaves (reference: 440), the aperiodicity measure (the ratio of aperiodic to total power), and the period-smoothed instantaneous power. With those parameters we can compute sound descriptors as Dynamic Stability or Pitch Stability which can be used to assess the quality of an instrument sound as reported by Romani et al [8]. Finally, The inverse values of Dynamic Stability and Pitch Stability together with Aperiodicity were normalized by subtracting the mean. The average value of the three descriptors for each trial conforms a unique descriptor called Sound Instability.

3 Results

3.1 Audio Analysis

The results of Sound Quality along trials of the beginners compared with experts can be seen in Figure 1. The number of trials were divided into three time periods and averged: Early (between trial one and six), Middle (between trial six and twelve) and Late (between trial thirteen and eighteen). One-way Anova for the three time periods was performed for each group in order to see differences between means reflecting the average learning process of the participants. Significant differences were obtained for the beginner group (p<0.25, p=0.00003) but not for the expert group (p<0.25, p=8.844).

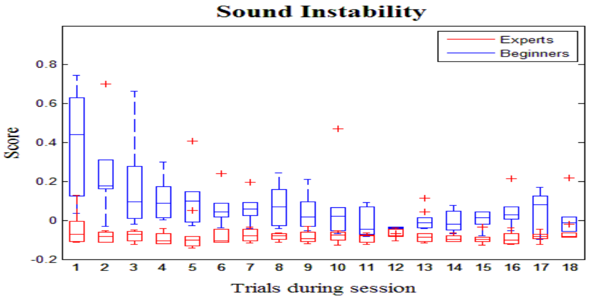


Fig.1. Box plot with the Sound Instability scores of each subject for each trial. We can see the scores of the beginner group in blue colour and the scores of the Expert group in red.

3.2 EEG Analysis

EEG data was first analyzed to see differences between beginners and experts using Information Gain feature selection in Weka [9] for all and each one of the three periods of time. Thirty five features over one hundred ninety eight were chosen by the ranker method. These features included: Gamma, Beta (Low and High), Alpha and High Theta in the frontal cortex, both parietal and temporal hemispheres, and in the right occipital hemisphere.

Seven machine learning algorithms were evaluated using a meta classifier computing the result first, for the best feature selected and later for the best feature selected plus the second one until ten. This method will allow us to see when the results stop increasing thus avoiding overfitting. The learning algorithms chosen were: J48, SMO(c=1.0), SMO(c=2.0), IBk(k=1), IBk(k=3), IBk(k=5) and Multilayer Perceptron. The results were obtained using a ten fold cross-validation.

Best results during the three periods were achieved by IBk(k=1) with a classification accuracy of 87.09% in the early period using five features before the results stabilized: High Beta in both frontal lobes and Gamma in the right parietal and occipital lobe. A classification accuracy of 94.84% in the middle period using also a number of five features: Gamma in left temporal lobe, the average Alpha value of all the electrodes, and High Beta and Alpha in both frontal lobes. And finally, a classification accuracy of 94.38% in the late period using the following five features: High Alpha in both frontal lobes, Gamma in left temporal lobe, High Beta in both frontal lobes and Low Beta in right occipital lobe.

High Beta together with Alpha in the frontal cortex seemed to be the most relevant frequency bands when used to classify between beginners and experts for each time period. In Figure 2 we can see the mean and standard deviation of High Beta compared between beginners and experts.

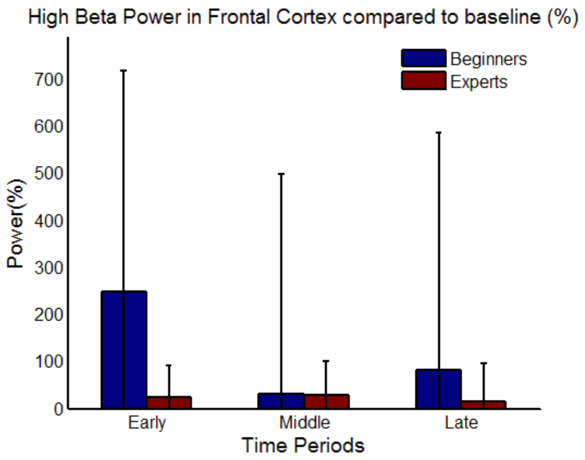


Fig.2. High Beta power in the Frontal Cortex compared to baseline. As we can see, specially at the beginning of the session, High Beta values are bigger in beginners rather than experts.

However when comparing alpha values in the frontal lobe between Beginners and Experts (Figure 3) we can see how Experts showed higher amplitude than beginners in this specific frequency band while beginners didn't showed big variations.

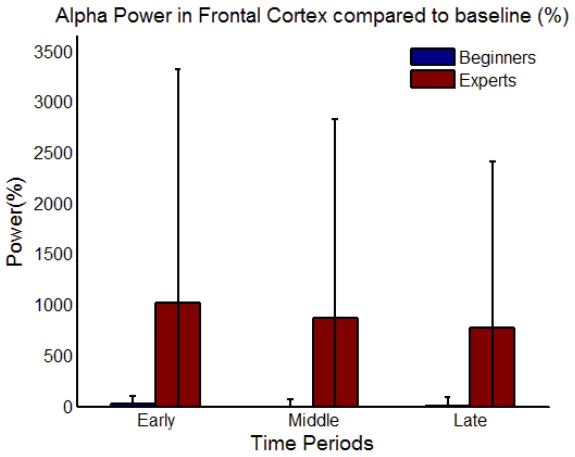


Fig.2. Alpha Power in Frontal Cortex compared to Baseline. As we can see, Experts showed higher values of Alpha power than beginners while performing the trials.

4 Conclusions

Results of the audio analysis shows how while beginners improved the quality of the generated songs along trials experts maintained stable results, thus allowing us to interpret differences between the brain signals of beginners and experts which are not time dependent (like fatigue or tiredness) and may reveal the effects of levels of expertise and skill acquisition.

We have seen how the High Beta band together with Alpha in the frontal cortex seemed to be the best feature that allowed us to classify the EEG data between beginners and experts using IBk (k=1) machine learning algorithm. Experts showed greater Alpha amplitudes than beginners while on the other hand beginners showed greater High Beta amplitudes. Nevertheless High Beta values decreased near to the expert levels between the middle and late period while Alpha remained stable in beginners throughout the session.

References

1. Gentili, R. J., Bradberry, T. J., Hatfield, B. D., & Contreras-vidal, J. L. (2008). A new generation of non-invasive biomarkers of cognitive-motor states with application to smart brain-computer interfaces, (Eusipco).

2.Emotiv Systems Inc. Researchers. (2014). Available online at: http://www.emotiv.com/researchers/

3. Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., et al. (2010). An open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. *MIT Press J. Presence* 19, 35–53. doi: 10.1162/pres.19.1.35

4. MATLAB and DSP Toolbox Release 2013a, The MathWorks, Inc., Natick, Massachusetts, United States

5. Zoom. Available online at: https://www.zoom-na.com/es/products/field-video-recording/field-recording/zoom-h4n-handy-recorder

6. Alain de Cheveigne & Hideki Kawahara (2002) YIN, a fundamental frequency estimator for speech and music. Acoustical Society of America. [DOI: 10.1121/1.1458024]

7. Quim Llimona. YIN pitch estimation toolbox (2015). GitHub repository, https://github.com/lemonzi/matlab/tree/master/yin

8. Romaní Picas O., Parra Rodriguez H., Dabiri D., Tokuda H., Hariya W., Oishi K., & Serra X."A real-time system for measuring sound goodness in instrumental sounds", 138th Audio Engineering Society Convention (2015).

9. Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

10. A Delorme & S Makeig (2004) EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics (pdf, 0.7 MB) Journal of Neuroscience Methods 134:9-21

What were you expecting? Using Expectancy Features to Predict Expressive Performances of Classical Piano Music

Carlos Cancino-Chacón[1,2], Maarten Grachten[2], David R. W. Sears[2], and Gerhard Widmer[1,2]

[1] Austrian Research Institute for Artificial Intelligence
carlos.cancino@ofai.at
[2] Department of Computational Perception, Johannes Kepler University Linz
{maarten.grachten,david.sears,gerhard.widmer}@jku.at

Abstract. In this paper we present preliminary work examining the relationship between the formation of expectations and the realization of musical performances, paying particular attention to expressive tempo and dynamics. To compute features that reflect what a listener is expecting to hear, we employ a computational model of auditory expectation called the *Information Dynamics of Music* model (IDyOM). We then explore how well these expectancy features – when combined with score descriptors using the Basis-Function modeling approach – can predict expressive tempo and dynamics in a dataset of Mozart piano sonata performances. Our results suggest that using expectancy features significantly improves the predictions for tempo.

Keywords: Musical expression, Information theoretic features, IDyOM, RNNs

1 Introduction

Computational models of musical expression can be used to explain the way certain properties of a musical score relate to an expressive rendering of the music [12]. However, existing models tend to use a combination of high- and low-level *hand-crafted features* reflecting structural aspects of the score that might not necessarily serve as perceptually relevant features. An example of such a model is the Basis-Function modeling approach (BM) [7].

To examine the relationship between the formation of expectations during music listening on the one hand, and the realization of musical performances on the other, Gingras et al. [4] employed the *Information Dynamics of Music* model (or IDyOM) [10], a probabilistic model of auditory expectation that computes information-theoretic features relating to the prediction of future events. In their study, these information-theoretic features were shown to correspond closely with temporal characteristics of the expressive performance, which suggests that the performer attempts to decrease the processing burden on listeners during perception by slowing down at unexpected/uncertain moments and speeding up at expected/certain ones.

Here we present preliminary work to support the claim that expectancy measures can inform predictions of expressive parameters related to tempo and dynamics. We extend the work in [4] in two ways. First, rather than simply demonstrating that expectancy measures are related to expressive performances, we show that the use of expectancy features improves the predictive quality of models using other score descriptors, thus providing a more comprehensive framework for the modeling of expressive performances in music of the common-practice period. Second, as opposed to *fitting* the expectancy features to each performance (i.e. training and testing the model on the same performance), the models presented in this paper are evaluated by measuring their prediction error on unseen pieces.

The rest of this paper is organized as follows: Section 2 presents our formalization of expressive parameters, describes the score and expectancy features employed in this study, and finally outlines the regression model used to predict the expressive parameters. Section 3 describes the empirical evaluation of the proposed approach, the results of which are discussed in Section 4. Finally, conclusions are stated in Section 5.

2 Modeling expressive performances

In this section we provide a brief description of the proposed framework. First we describe how expressive dynamics and tempo are encoded. Second, we describe the expectancy and score features. Finally we describe the recurrent neural network (RNN) models used to connect the input features to the expressive targets.

2.1 Targets: Expressive Parameters

An *expressive parameter* is a numerical descriptor that corresponds to common concepts involved in expressive piano performance. We take the local *beat period ratio* (*BPR*) as a proxy for *musical tempo*. We average the performed onset times of all notes occurring at the same score onset and then compute the *BPR* by taking the slope of the averaged onset times (in seconds) with respect to the score onsets (in beats) and dividing the resulting series by its average beat period. For *dynamics*, we treat the performed MIDI velocity as a proxy for loudness. We take the maximal performed MIDI velocity per score onset, divided by 127. This expressive parameter will be denoted *VEL*. To explore how well the expectancy and score features describe the *relative* changes in *BPR* and *VEL*, we also calculate their first derivatives, denoted by BPR_d and VEL_d, respectively. Furthermore, including the derivative time series allows us to compare our findings with the results obtained in [4].

2.2 Features: Multiple Viewpoints

Expectancy Features IDyOM provides a conditional probability distribution of a musical event, given a preceding sequence of events, i.e. $p(v_n \mid v_{n-1}, v_{n-2}, \ldots)$.

Following [4], we use IDyOM to estimate two information-theoretic measures representing musical expectations:

1. **Information content** (IC). The IC measures the unexpectedness of a musical event, and is computed as $IC(v_n) = -\log p(v_n \mid v_{n-1}, v_{n-2}, \ldots)$.
   (a) IC_m. The information content for each melody note. This value is computed using a model that is trained to predict the next chromatic melody pitch using a selection of melodic viewpoints, such as pitch interval (i.e. the arithmetic difference between two consecutive chromatic pitches, measured in MIDI note values), and contour (whether the chromatic pitch sequence rises, falls or remains the same). IDyOM performs a stepwise selection procedure that combines viewpoint models if they minimize model uncertainty as measured by corpus cross entropy [11].
   (b) IC_c. Estimation of the IC computed for the combination of pitch events (a proxy for harmony) at each score onset. IDyOM predicts the next combination of vertical interval classes above the bass (see Score Features 1b).
2. **Entropy** is a measure of the degree of choice or uncertainty associated with a predicted outcome. The entropy can be computed as $H(v_n) = \mathbb{E}\{-\log p(v_n \mid v_{n-1}, v_{n-2}, \ldots)\}$.
   (a) H_m. Entropy computed for each chromatic pitch in the melody.
   (b) H_c. Entropy computed for the combined pitch events at each score onset.

Score Features Following [7], we include low-level descriptors of the musical score that have been shown to predict characteristics of expressive performance.

1. **Pitch.**
   (a) $(pitch_h, pitch_l, pitch_m)$. Three features representing the chromatic pitch (as MIDI note numbers divided by 127) of the highest note, the lowest note, and the melody note at each onset.
   (b) (vic_1, vic_2, vic_3). Three features describing up to three vertical interval classes above the bass, i.e. the intervals between the notes of a chord and the lowest pitch, excluding pitch class repetition and octaves. For example, a C major triad (C, E, G), starting at C_4 would be represented as $(\,pitch_l\ vic_1\ vic_2\ vic_3\,) = (\,\frac{60}{127}\ \frac{4}{11}\ \frac{7}{11}\ 0\,)$, where 0 denotes the absence of a third interval above C_4, i.e. the absence of a fourth note in the chord.
2. **Metrical position.**
   (a) $b_{\phi,t}$. The relative location of an onset within the bar, computed as $b_{\phi,t} = \frac{t \mod B}{B}$, where t is the temporal position of the onset measured in beats from the beginning of the score, and B is the length of the bar in beats.
   (b) (b_d, b_s, b_w). Three binary features (taking values in $\{0, 1\}$) encoding the metrical strength of the t-th onset. b_d is nonzero at the downbeat (i.e. whenever $b_{\phi,t} = 0$); b_s is nonzero at the secondary strong beat in duple meters (e.g. quarter-note 3 in $\frac{4}{4}$, and eighth-note 4 in $\frac{6}{8}$), and b_w is nonzero at weak metrical positions (i.e. whenever b_d and b_s are both zero).

2.3 Recurrent Neural Networks

RNNs are a state-of-the-art family of neural architectures for modeling sequential data. Following [1, 6], we use bidirectional RNNs as non-linear regression models to assess how well the features described above predict expressive dynamics and tempo. In this work, we use an architecture with a composite bidirectional hidden layer with 5 units, consisting of a forwards and backwards long short-term memory layer (LSTMs).

3 Experiments

We perform a 5-fold cross-validation to test the accuracy of the predictions of three models trained on different feature sets for each expressive parameter: a model trained only using expectancy features (**E**), a model trained only using score features (**S**), and a model trained on both expectancy and score features (**E+S**). Each model is trained/tested on 5 different partitions (folds) of a dataset, which is organized into training and test sets, such that each piece in the corpus occurs exactly once in the test set.

For this study we use the Batik/Mozart corpus, which consists of recordings of 13 Mozart piano sonatas (39 movements) by Austrian pianist Roland Batik performed on a computer controlled Bösendorfer SE [2]. Melody voices were identified and annotated manually in this dataset. For each fold, we use 80% of the pieces for training and 20% for testing. The parameters of the models are learned by minimizing the mean squared error on the training set[3]. We evaluate model accuracy with the coefficient of determination R^2 and Pearson's r.

4 Results and Discussion

The results of the 5-fold cross-validation are shown in Table 1. To examine the differences between the R^2 values of the **E**, **S**, and **E+S** feature sets we performed a separate one-way ANOVA for each expressive parameter (BPR, BPR_d, VEL and VEL_d). These differences were statistically significant in all cases at the $p < 0.05$ level as measured by Fisher's F ratio. The same trend emerged for most expressive parameters, with **E+S** outperforming the other models, although post-hoc pairwise comparisons using Tukey's HSD only revealed a significant difference for BPR_d. These results therefore suggest that the models including both expectancy and score features better predict expressive tempo than expressive dynamics. Furthermore, although not directly comparable, the values for R^2 and r in Table 1 seem to be on par with those reported on Chopin piano music using the BM approach [6].

The fact that the use of expectancy features improves model performance for expressive tempo but not for dynamics might be due to the relation of expressive tempo to structural properties of the music, such as phrase-final lengthening, such as the final *ritardando* at the end of a piece [8]. Since expectation features

[3] A repository containing the code is available online: `https://github.com/neosatrapahereje/mml2017_expression_expectation`.

	Tempo				**Dynamics**			
Feature	BPR		BPR_d		VEL		VEL_d	
Set	R^2	r	R^2	r	R^2	r	R^2	r
E	0.038	0.201	0.067	0.259	0.234	0.496	0.185	0.429
S	0.065	0.289	0.105	0.326	0.299	0.569	0.244	0.494
E + S	0.072	0.288	0.124	0.351	0.312	0.574	0.230	0.477

Table 1. Predictive results for expressive tempo and dynamics, averaged over all pieces on the Batik/Mozart corpus. A larger R^2 and r means better performance.

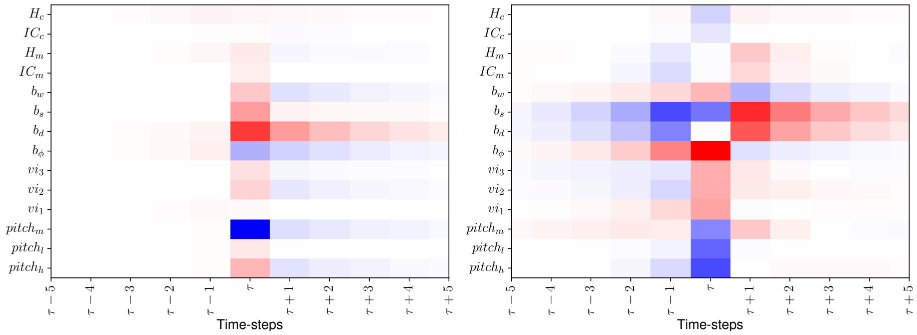


Fig. 1. Sensitivity plots for BPR (left) and BPR_d (right). Each row in the plot corresponds to an input feature and each column to the contribution of its value at that time-step to the output of the model at τ (the center of each plot). Red and blue indicate a positive and negative contribution, respectively.

also relate to music structure in the sense that music tends to be more unpredictable at boundaries between musical segments than within segments [9], this may in part explain why the models are better at predicting changes in expressive tempo BPR_d.

Figure 1 shows 2D differential sensitivity maps that examine the contribution of each feature to the output of the model trained on all features (**E+S**). Although these plots show that the score features have a more prominent role in predicting expressive tempo, as suggested by the results in Table 1, we will focus here on the contribution of the expectancy features. On the one hand, the plots suggest a tendency for the performer to slow down if the next melodic events are unexpected or uncertain (see the reddish hue in H_m and IC_m for time-steps $> \tau$ in the right plot), and to speed up if the previous melodic events were unexpected or uncertain (the bluish hue in H_m and IC_m for time-steps $< \tau$ in the right plot), which is consistent with the findings reported in [4]. On the other hand, while a passage consisting of uncertain harmonic events contributes to an overall slower tempo (the reddish hue in row H_c in the left plot), there is a tendency to speed up if the current harmonic event is unexpected or uncertain (the bluish hue in H_c and IC_c at τ in the right plot).

5 Conclusions

In this paper we presented a model for predicting expressive tempo and dynamics using a combination of expectancy and score features. Our results support the view that expectancy features, as reflecting what a listener is expecting to hear, can be used to predict the way pianists perform a piece. The sensitivity analysis also found some evidence relating to well-known rules/guidelines for performance [3, 4]. Future work may include the use of expectancy features in combination with larger sets of score descriptors (such as those in [5, 1]), and derive expectancy features from deep probabilistic models trained directly on (polyphonic) piano-roll representations.

Acknowledgements This work has been funded by the European Research Council (ERC) under the EUs Horizon 2020 Framework Programme (ERC Grant Agreement No. 670035, project CON ESPRESSIONE).

References

1. Cancino-Chacón, C.E., Gadermaier, T., Widmer, G., Grachten, M.: An evaluation of linear and non-linear models of expressive dynamics in classical piano and symphonic music. Machine Learning 106(6), 887–909 (2017)
2. Flossmann, S., Grachten, M., Widmer, G.: Expressive Performance with Bayesian Networks and Linear Basis Models. In: Rencon Workshop Musical Performance Rendering competition for Computer Systems. pp. 1–2 (Mar 2011)
3. Friberg, A., Bresin, R., Sundberg, J.: Overview of the KTH rule system for musical performance. Advances in Cognitive Psychology 2(2-3), 145–161 (2006)
4. Gingras, B., Pearce, M.T., Goodchild, M., Dean, R.T., Wiggins, G., McAdams, S.: Linking melodic expectation to expressive performance timing and perceived musical tension. Journal of Experimental Psychology: Human Perception and Performance 42(4), 594–609 (2016)
5. Giraldo, S.I., Ramírez, R.: A Machine Learning Approach to Discover Rules for Expressive Performance Actions in Jazz Guitar Music. Frontiers in Psychology 7, 194–13 (Dec 2016)
6. Grachten, M., Cancino-Chacón, C.E.: Temporal dependencies in the expressive timing of classical piano performances. In: Lessafre, M., Maes, P.J., Leman, M. (eds.) The Routledge Companion to Embodied Music Interaction, pp. 360–369. New York, NY (2017)
7. Grachten, M., Widmer, G.: Linear Basis Models for Prediction and Analysis of Musical Expression. Journal of New Music Research 41(4), 311–322 (Dec 2012)
8. Honing, H.: Computational Modeling of Music Cognition: A Case Study on Model Selection. Music Perception 23(5), 365–376 (Jun 2006)
9. Pearce, M., Müllensiefen, D., Wiggins, G.A.: A Comparison of Statistical and Rule-Based Models of Melodic Segmentation. In: Proceedings of the Ninth International Conference on Music Information Retrieval. Philadelphia, PA, USA (2008)
10. Pearce, M.T.: The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition. Ph.D. thesis, City University, London (2005)
11. Sears, D.R.W.: The Classical Cadence as a Closing Schema: Learning, Memory, & Perception. Ph.D. thesis, McGill University, Montreal, Canada (Sep 2016)
12. Widmer, G., Goebl, W.: Computational Models of Expressive Music Performance: The State of the Art. Journal of New Music Research 33(3), 203–216 (Sep 2004)

Modelling and sampling jazz chord sequences

Darrell Conklin

Department of Computer Science and Artificial Intelligence,
University of the Basque Country UPV/EHU, San Sebastián, Spain
IKERBASQUE: Basque Foundation for Science, Bilbao, Spain

Abstract. This paper reports on jazz chord sequence generation with semiotic patterns. Semiotic patterns represent coherence within the sequence and can be derived directly from a template piece using the method of inverse substitution. Instantiations of semiotic patterns are sampled using a statistical model trained on a corpus. Iterated random walk and Gibbs sampling are applied and properties of the sampling methods are discussed.

Keywords: machine learning, music generation, jazz chord sequences, statistical models, semiotic patterns

1 Introduction

All music generation methods must contend with the problem of long term structure and repetition. It is generally accepted that context models (finite state, Markov and Hidden Markov Models) are insufficient, and that intra-opus repetition or *coherence* and inter-opus recurrence are distinct phenomena requiring different computational modelling techniques. Early work [1] interpolates short and long term models, where the former are trained on the particular sequence, and the latter on a corpus of stylistically related music. A more recent approach [2, 3] is to extract intra-opus repetition from a template piece, describing it using a semiotic pattern, then respecting that pattern during generation from a statistical model of a corpus.

With a statistical model of music, one can apply *optimization* methods to propose a few high probability sequences (e.g. [4, 5]) or can apply *sampling methods* to explore the broad probability distribution of a model, followed by drawing a few sequences from desired ranges [6, 5, 7, 2]. In this paper the semiotic pattern method is extended from trance [2] to jazz chord sequences, and the properties of iterated random walk and Gibbs sampling are discussed with application to a jazz song template.

2 Methods

2.1 Data

In this study a statistical model is trained (Section 2.2) on the Weimar Jazz Database (WJazzD) [8], a curated collection of 456 jazz standards, containing a

chord	Fmaj7	G7	Gm7	C7	Am7b5	D7	Gm7
dur	𝅜	𝅜	𝅝	𝅝	𝅝	𝅝	𝅝
root	F	G	G	C	A	D	G
triad	M	M	m	M	d	M	m
seventh	7	b7	b7	b7	b7	b7	b7
int(root)	⊥	M2	P1	P4	M6	P4	P4
int(triad)	⊥	MM	Mm	mM	Md	dM	Mm
int(seventh)	⊥	7b7	b7b7	b7b7	b7b7	b7b7	b7b7

Fig. 1. The first 7 chords of the jazz standard *Desafinado* (Jobim and Mendonça; solo by Art Pepper as WJazzD record #3) with basic viewpoints (top), three derived viewpoints (middle) and three constructed interval viewpoints (bottom).

total of $\approx 29,000$ chords. Pickup measures are ignored and where the form (e.g. AABA) is annotated in the WJazzD only the first iteration of the chorus is used in building a statistical model, thereby removing chord sequence duplications introduced by intra-opus repetitions of the same chorus. These reductions lead to a training set of $\approx 17,000$ chords.

Figure 1 shows a short chord sequence. From each chord symbol is derived a root, triad, and seventh. Note that pitches and intervals are represented by spelled pitches and diatonic intervals. Due to the sparsity of the WJazzD for chords above the seventh (ninths and above, altered chords, etc.), these are not considered in this study. Slash chords are ignored and since chord roots sharper/flatter than A$\sharp$/C$\flat$ are absent from the corpus, they are not considered as part of chord types. Thus all chords are collapsed into a root, a triad (major, minor, augmented, diminished, and sus4) and a seventh (none, major seventh, minor seventh, and diminished seventh), giving a total of 306 chord types. Contiguous self-self transitions resulting from collapsing chords are merged and component durations are added.

2.2 Statistical model for chords

To describe the pieces in the corpus on different levels of abstraction, and deal with sparse data, a viewpoint model is used. A viewpoint τ is a function that maps an event sequence $e_1, \dots, e_\ell$ to a more abstract derived sequence $\tau(e_1), \dots, \tau(e_\ell)$, comprising elements in the codomain $[\tau]$ of the function τ.

To create a statistical viewpoint model, consider two successive chords a and b, and let $v = (\mathsf{int(root)} \otimes \mathsf{int(triad)} \otimes \mathsf{int(seventh)})(b \mid a)$. Following the derivation in [2], the conditional probability $\mathbb{P}(b \mid a)$ can be written in the form

$$\mathbb{P}(b \mid a) = \mathbb{P}(v) \times \mathbb{P}(b \mid v, a) \tag{1}$$

which, if b is fully determined by a and v, simplifies to $\mathbb{P}(b \mid a) = \mathbb{P}(v)$.

Following the methods described in [2], a viewpoint model for the viewpoint $\mathsf{int(root)} \otimes \mathsf{int(triad)} \otimes \mathsf{int(seventh)}$ was learned from the WJazzD corpus. Smoothing is applied so that all possible elements of $[\mathsf{int(root)} \otimes \mathsf{int(triad)} \otimes \mathsf{int(seventh)}]$ have non-zero probabilities, ensuring that Gibbs sampling (Section 2.4) converges to the target sequence distribution. A transition matrix over all chords is compiled according to (1). The *information content* (IC) of a sequence $\boldsymbol{e}$ is $-\log_2 \mathbb{P}(\boldsymbol{e})$. Note that for a pattern Φ the conditional probability $\mathbb{P}(\boldsymbol{e} \mid \Phi)$ is simply a scaling of $\mathbb{P}(\boldsymbol{e})$ by the normalization constant $\mathbb{P}(\Phi)$.

2.3 Semiotic patterns

The semiotic pattern representation developed in [2] is used to represent intra-opus coherence. A *semiotic pattern* is a sequence of *features*, each being a viewpoint and value or variable. In this paper only the chord viewpoint will be used in patterns, thus a pattern can be viewed as a sequence of variables and concrete chord values.

A *substitution* μ is a mapping from all variables in a pattern to chords, and an *instance* $\boldsymbol{e}$ of a pattern Φ is an event sequence given by the application of a substitution: $\boldsymbol{e} = \Phi\mu$. In some situations it is desirable to partially specify a substitution by locking some variables, then μ will be an extension of the initial substitution. Substitutions here are required to be *injective*, that is, no two variables can map to exactly the same chord.

An *inverse substitution* μ^{-1} has the property that $\Phi\mu\mu^{-1} = \Phi$. An inverse substitution creates a pattern $\Phi = \boldsymbol{e}\mu^{-1}$ from one specific piece $\boldsymbol{e}$, and is a standard generalization operator in machine learning [9]. A *least general* inverse substitution can be created by using a fresh variable for every distinct chord appearing in the sequence.

2.4 Sampling methods

Statistical models are used to rank the space of sequences instantiating a given pattern by increasing IC. Since this space is too large to enumerate, it must be sampled, and three methods are applied here:

- *Iterative random walk* (IRW): perform random walk while at each position assuring that the sequence up to that point is a partial instantiation of Φ [2]. Iterate this process many times to generate a sample of solutions.
- *Gibbs sampling*: given a current substitution μ, choose a variable $\rho \in \mathrm{dom}(\mu)$, and select a new substitution $\mu' = \mu[\rho \mapsto a]$ and sequence $\boldsymbol{e}' = \Phi\mu'$ with probability
$$\frac{\mathbb{P}(\boldsymbol{e}')}{\sum_a \mathbb{P}(\Phi\mu[\rho \mapsto a])}$$
and set $\mu \leftarrow \mu'$ in preparation for the next iteration. The process is initiated with some solution $\boldsymbol{e} = \Phi\mu$ (generated by IRW), and iterated to generate a sample of solutions.

– *Gibbs sampling starting from a template*: given the template $\boldsymbol{e}$, pattern Φ and a least general inverse substitution μ^{-1}, μ determines one instantiation of the pattern, and therefore may also be used as the first instantiation of the pattern during Gibbs sampling. This focuses the early steps of sampling towards sequences similar to the template.

3 Results

To study the chord sequence generation method, least general inverse substitutions are used to create semiotic patterns from template pieces in the WJazzD. For illustration, the piece *Anthropology* (Charlie Parker and Dizzy Gillespie; solo by Art Pepper as WJazzD record #1) is chosen as a template.

Figure 2 (top) shows the distribution of IC produced by running 10000 iterations of each method described in Section 2.4. The horizontal red line shows the IC of the template sequence itself and the number of unique sequences produced is shown at the mean of each method. It is apparent that Gibbs sampling reaches a smaller fraction of the space touched by IRW. The diversity (number of unique sequences) produced by Gibbs sampling ($n = 2453$) is lower than IRW, which produced a unique sequence at every iteration. As expected, Gibbs sampling from the template reaches the lowest IC sequences, though with even smaller diversity than regular Gibbs. Interestingly both variants of Gibbs sampling visit sequences with lower information content than the template itself.

The next experiment evaluates diversity of the top k (lowest IC) solutions produced, simulating a real situation where a composer may be interested in, say, just the top 10 sequences. 10000 iterations of each method are run 100 times, and statistics of the top 10 sequences are compiled: the information content distribution, the average similarity between sequences in the set (measured as the proportion of identically instantiated variables), and average similarity to the template sequence. Figure 2 (bottom) shows the results. For IRW the intra-10 set similarity is low and varies little over the 100 runs. For Gibbs the mean intra-10 similarity is high, and also Gibbs with the template as the starting sequence consistently produces a very low diversity set. As expected (blue boxes), where Gibbs sampling starting from a template produces sequences with high similarity to the template.

Finally some fragments of generated chord sequences, corresponding to the first phrase of the template *Desafinado*, are shown in Figure 3 (durations are retained from the template but not shown here). The Fmaj7 chord, which articulates the tonality of the piece, has been locked.

4 Conclusions

This paper has applied viewpoint modelling, semiotic patterns, and sampling to the task of jazz chord sequence generation. Presented for one template in this paper, some behaviours of the different sampling algorithms emerge: IRW provides a better spread of the sequence space while Gibbs tends to sample fewer

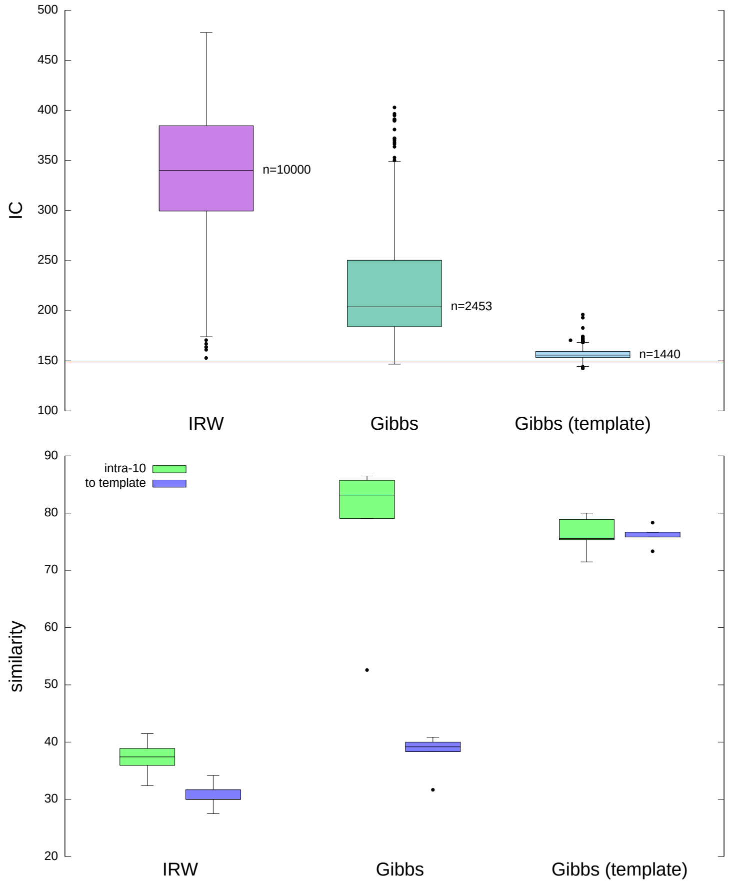


Fig. 2. IC distribution (top): n is the number of unique sequences sampled in 10000 iterations; diversity (bottom) for the template *Anthropology*.

sequences though at better IC ranges, with a lower median IC. The tentative conclusion is that if resources are available, IRW can be a good sampling method for chord sequences while Gibbs will reach good solutions in resource-bounded situations.

Acknowledgements

This research is supported by the project I2C8 (Inspiring to Create) which is funded by the European Union's Horizon 2020 Research and Innovation programme under grant agreement number 754401. Thanks to Klaus Frieler for correspondences regarding the Weimar Jazz Database and to Kerstin Neubarth for helpful discussions on the research.

	🔒											🔒
	𝅜	𝅜	𝅝	𝅝	𝅝	𝅝	𝅝	𝅝	𝅜	𝅜	𝅜	𝅜
template (129)	Fmaj7	G7	Gm7	C7	Am7b5	D7	Gm7	A7	D7	G7	Gbmaj7	Fmaj7
IRW (96)	Fmaj7	Fm7	Bb7	F7	Bbm7	Eb7	Bb7	Ab7	Eb7	Fm7	C7	Fmaj7
Gibbs (80)	Fmaj7	Fm7	Bb7	EbM7	Cm7	F7	Bb7	Eb	F7	Fm7	C7	Fmaj7
GibbsT (81)	Fmaj7	Dm7	G7	CM7	Am7	D7	G7	CM	D7	Dm7	C7	Fmaj7

Fig. 3. Fragment (first phrase) of the lowest IC generated sequence for each method (10000 iterations), template *Desafinado*. One chord of the following phrase is shown. In brackets is the IC of the entire sequence.

References

[1] D. Conklin and I. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.

[2] D. Conklin. Chord sequence generation with semiotic patterns. *Journal of Mathematics and Music*, 10(2):92–106, 2016.

[3] T. Collins, R. Laney, A. Willis, and P. H. Garthwaite. Developing and evaluating computational models of musical style. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 30(1):16–43, 2016.

[4] P. J. Ponce de León, J-M. Iñesta, J. Calvo-Zaragoza, and D. Rizo. Data-based melody generation through multi-objective evolutionary computation. *Journal of Mathematics and Music*, 10(2):173–192, 2016.

[5] D. Herremans, K. Sörensen, and D. Conklin. Sampling the extrema from statistical models of music with variable neighbourhood search. In *Proceedings of the 11th Sound and Music Computing Conference (SMC 2014)*, pages 1096–1103, Athens, Greece, 2014.

[6] D. Conklin. Music generation from statistical models. In *Proceedings of the AISB Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pages 30–35, Aberystwyth, Wales, 2003.

[7] R. P. Whorley and D. Conklin. Improved iterative random walk for four-part harmonisation. In T. Collins, D. Meredith, and A. Volk, editors, *Mathematics and Computation in Music*, pages 64–70. Springer International Publishing, 2015.

[8] K. Frieler, J. Abeßer, W.-G. Zaddach, and M. Pfleiderer. Introducing the Jazzomat Project and the Melo(S)py Library. In *Proceedings of the 3rd International Workshop on Folk Music Analysis (FMA 2013)*, pages 76–78, Amsterdam, Netherlands, 2013.

[9] L. De Raedt. *Logical and Relational Learning.* Springer-Verlag New York, Inc., 2008.

A Real-time Feedback Learning Tool to Visualize Sound Quality in Violin Performances

Sergio Giraldo, Rafael Ramirez, George Waddell, and Aaron Williamon

Universitat Pompeu Fabra (UPF), Barcelona, Spain
Centre for Performance Science, Royal College of Music, UK
{sergio.giraldo, rafael.ramirez}@upf.edu
{george.waddell, aaron.williamon}@rcm.ac.uk

Abstract. The assessment of the sound properties of a performed musical note has been widely studied in the past. Although a consensus exist on what is a good or a bad musical performance, there is not a formal definition of performance tone quality due to its subjectivity. In this study we present a computational approach for the automatic assessment of violin sound production. We investigate the correlations among extracted features from audio performances and the perceptual quality of violin sounds rated by listeners using machine learning techniques. The obtained models are used for implementing a real-time feedback learning system.

Keywords: Machine learning, Violin sound quality, Automatic assessment, Timbre dimensions, Audio features

1 Introduction

The quality of a performed sound is assumed to be a contribution of several parameters of sound such as pitch, loudness and timber. Eerola et al. (2012) identify 26 acoustic parameters of timbre among several instrument groups, that combined produce a particular sound quality, which might reflect a particular instrument performance technique, and/or the expressive intentions of the performer. Automatic characterization of dynamics and articulation from low level audio features has been studied by Maestre & Gómez (2005) in the context of expressive music performance. Knight et al. (2011) study the automatic assessment of tone quality in trumpet sounds using machine learning techniques. Romani Picas et al. (2015) make use of machine learning techniques to identify good and poor quality notes given training data consisting of low and high level audio features extracted from performed musical sounds. However, whereas pitch and dynamic measurements can be easily obtained from a computational perspective, a measure for timbre quality involves significant complications given that the exact formulation of timbre dimensions are still a matter of debate.

In this study we present an approach to automatically assess tone quality. Our aim is twofold: firstly, to understand the correlations between the proposed tone qualities, the ones previously used in the literature (e.g. Romani Picas et al. (2015)) and the features extracted from the audio signal, and secondly, to generate machine learning models to predict the different proposed quality dimensions of the performance from the audio features. We have investigated the relationship between the terms that musicians use for quality assessment (e.g. clarity, warmth, depth, brilliance, resonance, richness, power) and low-level audio features (e.g. spectral centroid, spread, skewness, kurtosis, slope, decrease, roll-off point, flatness, spectral variation, spectral complexity, spectral crest MFCCs, and the energy in specific frequency bands), using machine learning techniques, based on the recordings and evaluations by music experts.

The predictive models were implemented and incorporated into a real-time feedback learning system, able to give automatic feedback about the timbral properties (Timbral dimensions) of exercises/notes being performed.

2 Methodology

2.1 Feature extraction and feature selection for real-time audio analysis.

Low and high-level audio features were extracted from the audio signals in both temporal and spectral domains using the Essentia library (Bogdanov et al. (2013)), using a frame size of 23ms, with a hop size of 11.5ms. On the other hand, perceptual tests to assess the quality of performed notes was conducted, in which 30 participants (with at least one year of musical training) were asked to mark sound quality in terms of predefined dimensions: dynamic, pitch and timbre stability, pitch accuracy and timbre richness, on a 7-point Likert scale. 27 Violin sounds were obtained from the public available data base by (Romani Picas et al. 2015), and selected in order to cover an homogeneous range of the violin's tessitura. Similarly, a proposed list of tone qualities (see Table 1), defined by music educators, was presented in pairs to the listener (e.g. Bright/Dark) to grade the sounds along a 7-point Likert scale.

Table 1: Proposed list of tone qualities by music experts

Tone Qualities	
Dark	Bright
Cold	Warm
Harsh	Sweet
Dry	Resonant
Light	Heavy
Grainy	Pure
Coarse	Smooth
Closed	Open
Restricted	Free
Narrow	Broad

2.2 Dynamics and intonation dimensions

Dynamics and pitch values were extracted from the audio by extracting the energy of the signal based on a frame-based calculation of the Root Mean Square (RMS), as well as, by obtaining frame based pitch values.

Pitch Accuracy (PA) . Pitch accuracy was measured in terms of the deviation in cents of the measured pitch to the closest tempered semitone. The actual pitch value was calculated in real-time on a frame basis (at 66 fps) using the Essentia Pitch Detection library. The obtained pitch values were smoothed using a 10 point average filter. Pitch accuracy was then obtained by obtaining the absolute difference between the pitch value and the closest semitone, and dividing by 50 cents (i.e. half of a semitone size). Thus, pitch accuracy ranges from 0 to 1, where the maximum deviation allowed is a semitone.

Pitch Stability (PS) . Pitch stability was calculated based on the standard deviation of the obtained frame-based value over a 300 ms historic window. Firstly, pitch frame-based values obtained with the Essentia Pitch Detection library were smoothed by applying a 10 point average filter. Standard deviation was calculated over a historical 300ms window. Low standard deviation values were assumed to indicate high pitch stability and vice versa.

Dynamic Stability (DS) . Dynamic stability was obtained by calculating the standard deviation of the energy over a 600 ms historic window. Firstly, we calculated a frame-based RMS (Root Mean Square) values. RMS values are later converted to decibel (dB) values and smoothed using a 10 point average filter. The standard deviation of the filtered RMS values was calculated over a 600 ms historical window. Similarly to pitch stability calculation, low standard deviation values were assumed to correspond to high dynamic stability and vice versa.

2.3 Timbral Dimensions Calculation

Timbral dimensions were calculated by training models which combined several of the audio features extracted with the Essentia library (Bogdanov et al. (2013)). Feature selection was performed over spectral descriptors, known to be close related to timbral characteristics of sound (see Peeters et al. (2011) for an overview), to obtain a subset of tonal descriptors that best predict each of the studied timbral dimension. The selected features include pitch, energy, spectral time-varying descriptors (centroid, spread, skewness, kurtosis, slope, decrease, rolloff, flatness, crest), spectro-harmonic (tristimulus 1, tristimulus 2, tristimulus 3, harmonic energy, noise energy). Mean and standard deviation over a 300 ms window was considered as well, for all the set of descriptors.

Timbre Stability (TS) and Timbre Richness (TR) The spectrum was obtained from the audio frame by means of the Fast Fourier Transform (FFT) and peak detection was performed on the spectrum afterwards. Based on the actual pitch value detected, the harmonic peaks were selected, allowing a 20% deviation from the ideal harmonic series. Later, spectral harmonic features, (e.g. tristimulus 1, 2 and 3) as well as time varying spectral features (e.g. kurtosis, skewness) were calculated.

2.4 Sound Dimensions Modelling.

Machine learning techniques were used to generate models to predict the different quality dimensions from the extracted features. Feature selection techniques were applied in order to obtain the subset of low level (frame-based) descriptors that best predict each of the studied sounds dimensions. Several machine learning schemes were compared, i.e., Linear Regression, M5-trees, Artificial Neural Networks, and Support Vector Machines.

Stability of pitch energy and timber Models were trained, to map the calculated standard deviations of the highest pitch stability rated sounds to 1 (good pitch stability) and, conversely, the bad examples to 0 (bad pitch stability). Correspondingly,models were trained to map the standard deviation values calculated in good/bad dynamic stability examples with a corresponding 0 to 1 dynamic stability value.

Timbre richness and stability Previously selected features were used to train models to order to best predict the ratings obtained on the surveys for timbral properties. For both Timbre Stability and Timbre Richness logistic regression models were obtained, using combinations of spectral features explained in Section 2.1.

3 Results

3.1 Tone survey

Consistency among participants ratings was assessed using Cronbach's coefficient (alpha). An acceptable degree of reliability was obtained (alpha>80, MCGraw and Wong, 1996) for all the sound examples. On the other hand, higher correlations (i.e. CC>0.8) were obtained between the overall quality of the sound and pitch stability/timbre richness.

3.2 Models accuracy

In Table 2 we present the Correlation Coefficient Index (CCI) obtained by the different models studied for the prediction of the rating on each of the dimensions considered. The obtained CCI of the models is presented as calculated in both the Train Set (TS) and on a 10-Cross Fold validation scheme (CV), as an indicator of over-fitting. Consideration was also taken in terms of the feasibility of implementation of the models in a real-time application, giving priority to the ones less computationally expensive. For all the dimensions studied linear regression models were selected because of its overall good performance in terms of accuracy, low computational cost, and simplicity for implementation.

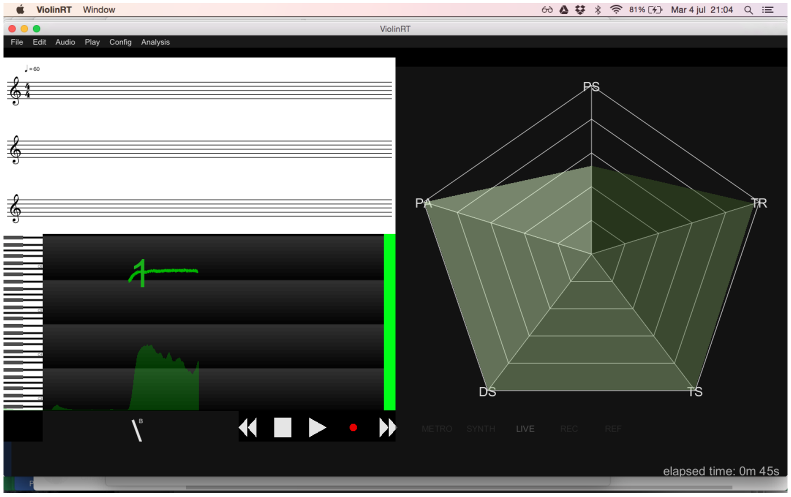
Table 2: Accuracies (CCI) for different sound's dimensions quality

Sound Dimension	Lin.Reg	Reg-Trees	SVMreg	ANN
cv / train	cv / train	cv / train	cv/ train	
Pitch Accuracy	0.89 / 0.80	0.60 / 0.88	0.79 / 0.86	0.64 / 0.72
Pitch Stability	0.80 / 0.91	0.82 / 0.98	0.81 / 0.88	0.68 / 0.68
Dynamic Stability	0.82 / 0.84	0.67 / 0.87	0.78 / 0.85	0.69 / 0.65
Timbre Stability	0.80 / 0.89	0.63 / 0.91	0.86 / 0.81	0.60 / 0.75
timbre Richness	0.78 / 0.86	0.71 / 0.97	0.85 / 0.80	0.60 / 0.66

3.3 Real-time feedback learning widget

The aforementioned sound dimensions models for measuring the goodness in terms of the intonation, dynamics and tone, were presented in a intuitive graphic user interface, on the Violin RT app, as illustrated in Figure 1. Each sound dimension is presented on each axis of a spider chart, aiming at an intuitive user interaction in which the best sound quality is obtained when the chart is full filled.

Fig. 1: Real-time feedback system learning system screen shoot



4 Conclusions

In this paper a computational approach to automatically assess the quality of performed violin sounds was proposed. We conducted perceptual tests on the quality of recorded sounds based on previous defined quality dimensions, and studied the correlation among the different quality dimensions. Energy and spectral descriptors were extracted from the audio signal and machine learning models were obtained to predict the different quality dimensions from the audio features. Results indicate consistency among users responses, and the obtained models accuracy suggests that the extracted audio features contain sufficient information for characterizing the proposed tonal dimensions. Ongoing work includes extending the recording data, as well as modelling other tonal dimensions.

Acknowledgements

This work has been partly sponsored by the Spanish TIN project TIMUL (TIN2013-48152-C2-2-R), the European Union Horizon 2020 research and innovation programme under grant agreement No. 688269 (TELMI project), and the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

References

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J. R., Serra, X. et al. (2013), Essentia: An audio analysis library for music information retrieval., *in* 'ISMIR', pp. 493–498.

Eerola, T., Ferrer, R. & Alluri, V. (2012), 'Timbre and affect dimensions: evidence from affect and similarity ratings and acoustic correlates of isolated instrument sounds', *Music Perception: An Interdisciplinary Journal* **30**(1), 49–70.

Knight, T., Upham, F. & Fujinaga, I. (2011), The potential for automatic assessment of trumpet tone quality., *in* 'ISMIR', pp. 573–578.

Maestre, E. & Gómez, E. (2005), Automatic characterization of dynamics and articulation of expressive monophonic recordings, *in* 'In Proceedings of the 118th Audio Engineering Society Convention', Citeseer.

Peeters, G., Giordano, B. L., Susini, P., Misdariis, N. & McAdams, S. (2011), 'The timbre toolbox: Extracting audio descriptors from musical signals', *The Journal of the Acoustical Society of America* **130**(5), 2902–2916.

Romani Picas, O., Parra Rodriguez, H., Dabiri, D., Tokuda, H., Hariya, W., Oishi, K. & Serra, X. (2015), A real-time system for measuring sound goodness in instrumental sounds, *in* 'Audio Engineering Society Convention 138', Audio Engineering Society.

Album Cover Generation from Genre Tags

Alexander Hepburn, Ryan McConville, and Raúl Santos-Rodríguez

University of Bristol,
United Kindgom
{ah13558,ryan.mcconville,enrsr}@bristol.ac.uk

Abstract. This paper presents a method for generating album cover art by including side information regarding the music content. In this preliminary work, using state of the art Generative Adversarial Networks (GAN), album cover arts are generated given a genre tag. In order to have a sufficient dataset containing both the album cover and genre, the Spotify API was used to create a dataset of 50,000 images separated into 5 genres. The main network was pre-trained using the One Million Audio Cover Images for Research (OMACIR) dataset and then trained on the Spotify dataset. This is shown to be successful as the images generated have distinct characteristics for each genre and minimal repeated textures. The network can also distinguish which genre a generated image comes from with an accuracy of 35%.

Keywords: Cover art generation, Generative Adversarial Networks, Genre classification

1 Introduction

Music and visual effects are often linked together and can provide a multi-sensual experience to the user. Given a small set of songs, images could be chosen or created by hand to fit a song or album. However, this becomes unrealistic as the size of the music collection increases. For instance, consider popular website platforms in which users upload their own music. For instance, `SoundCloud`[1] has 12 hours of music and audio uploaded every minute, from over 150 million independent users. Also, a great deal of effort has been recently devoted to the automatic generation of music using deep learning [3]. For example, `Jukedeck`[2] is a platform that uses deep neural networks to generate unique songs from a user specified style and feeling. Stock images are currently displayed but it is more aesthetically pleasing to have a unique image for each song that also reflects some characteristics found within the music. This paper aims to address this by automatically generating an image at the same time as the music content, where the image is unique and reflects some of the characteristics of music. As musical genres are common proxies to categorise and describe music, we use genre labels as a first abstraction of music properties.

[1] `https://soundcloud.com/`

[2] `https://www.jukedeck.com/`

2 Image generation using Generative Adversarial Networks

State-of-the-art approaches in image generation include those based on Generative Adversarial Networks (GANs)[2, 1, 4]. The framework is based on two complementary networks, namely a discriminative (D) network which tries to classify data into sets and a generative (G) network which is used to create new data from a prior distribution. In GANs a generative and discriminative neural network is pitted against each other, posing a minimax problem. New images are generated using G, using noise samples as a seed. Then the newly generated images are used as an input along with real images to D. D then tries to classify which data is real or generated. The variables in D are optimised to be able to distinguish between real and generated data, whilst the variables in G are optimised to fool D into classifying the generated data as real. As such, G learns how to create real looking data simultaneously as D learns to discriminate between generates images and images from the dataset.

In a similar fashion, a Deep Convolutional Generative Adversarial Network (DCGAN) is a GAN which makes use of convolution layers [6]. This can either be in just the generator or both the generator and discriminator. DCGANs achieve better results when generating complex images. Using a combination of up-sampling and transpose convolution layers in the generator produces higher resolution images that look more lifelike.

Finally, the recent Auxiliary Classifier Generative Adversarial Network (AC-GAN)[5] code some descriptive variables into the noise which is used as an input to the generator network. The discriminator then tries to predict these descriptive variables resulting in more consistent training of both the networks as well as being able to specify classes of images. Additionally, they also introduce the use of latent variables in order to make training GANs more consistent. These are random variables that are generated for every generated image and used within the noise vector as input to the generator network. The discriminator then predicts what the random variables used to generate the image are. The use of these latent variables as well as class labels to conditionally generate examples lead to more realistic images as well as being able to generate any class from the pre-specified set of classes.

3 Experiments

Our first goal in this work is to empirically show that it is possible to automatically generate album covers using GANs. As compared to standard image and computer vision datasets, album covers have a huge variety of objects in them as well as different art styles. The limited availability of labelled training data is also a challenge. Finally, we will show how to use AC-GANs to incorporate the genre information into the generation process.

3.1 Genre agnostic generation: One Million Audio Cover dataset

The One Million Audio Cover Images for Research (OMACIR) is a dataset constructed from a variety of sources containing over one million album cover arts[3]. These images are a mixture of greyscale and RGB images, all of different sizes. There are also a large number of repeated images throughout the dataset which would strongly affect any image generation algorithm. A hash based technique was used to detect and remove 798982 duplicate images. All images were resized to 64x64 and standardised so that values lie in the region (-1,1) with a mean of 0. To generate images from an AC-GAN network trained on the OMACIR dataset which lacks classes, we had to modify the cost function to only optimise w.r.t. generating realistic images and predicting latent variables.[4]

Table 1. Network architectures used in the AC-GAN network when generating album covers from both the One Million Cover Images for Research dataset and Spotify dataset, both using 2 latent variables. In the discriminator fully connected 1 is responsible for predicting whether an image is generated or from the dataset, fully connected 2 is responsible for predicting the class label and fully connected 3 is responsible for predicting the latent variables. Transposed convolution is often referred to as deconvolution.

Generator

Layer	Input	Filter	Output	Upsampling	Activation
Fully connected 1	1x100	100x16384	1x16384	0	Linear
Reshape	1x16384		4x4x1024	0	
Transpose Convolution 1	4x4x1024	4x4x512	8x8x512	2	ReLU
Transpose Convolution 2	8x8x512	4x4x256	16x16x256	2	ReLU
Transpose Convolution 3	16x16x256	4x4x128	32x32x128	2	ReLU
Transpose Convolution 4	32x32x128	4x4x3	64x64x3	2	Tanh

Discriminator

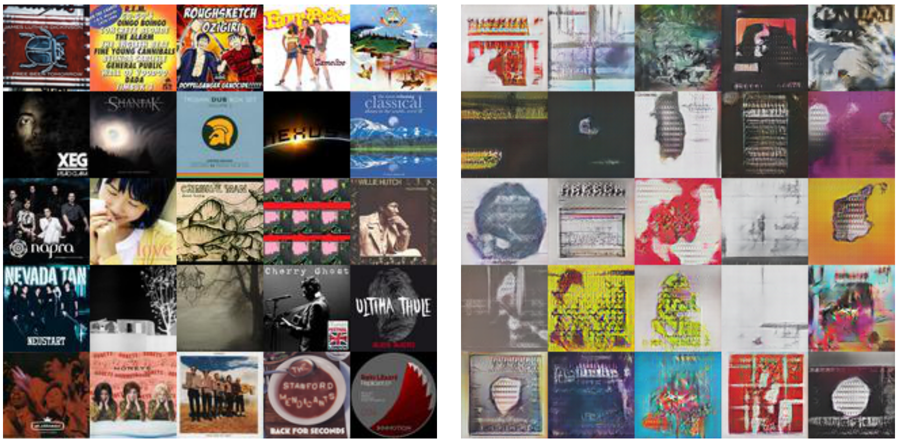
Layer	Input	Filter	Output	Stride	Activation
Convolution 1	64x64x3	4x4x128	32x32x128	2	Leaky ReLU
Convolution 2	32x32x128	4x4x256	16x16x256	2	Leaky ReLU
Convolution 2	16x16x256	4x4x512	8x8x512	2	Leaky ReLU
Convolution 2	8x8x512	4x4x1024	4x4x1024	2	Leaky ReLU
Reshape	4x4x1024		1x16384	0	
Fully connected 1	1x16384	16384x1	1x1	0	Linear
Fully connected 2	1x16384	16384x5	1x5	0	Linear
Fully connected 3	1x16384	16384x2	1x2	0	Linear

The network architecture used is detailed in Table 1. The best network parameters, found via a grid search, include a generative learning rate of 0.002, a discriminative learning rate of 0.001 and a batch size of 128. The input noise

[3] `https://archive.org/details/audio-covers`

[4] Code can be found at `https://github.com/alexhepburn/cover-art-generation`.

is taken from a uniform distribution in the region (-1, 1). Overall the resulting images in Fig. 1 are of good visual quality with minimal repeated textures and have properties which are indicative of album covers.



(a) Original album covers.

(b) Generated album covers.

Fig. 1. AC-GAN trained on the OMACIR dataset.

3.2 Genre aware generation: Spotify dataset

Although OMACIR is extremely useful due to the amount of images, it contains no metadata of artists, genres or album names. To compile a dataset that contains such metadata, the Spotify API[5] was queried with a variety of genres (Jazz, Dance, Rock, Rap and Metal) and the first 10,000 unique album names were recorded for each genre. While it has been established that a deep learning network can generate realistic looking album cover art from the OMACIR dataset, our objective is to generate album covers given prior knowledge about the album itself. To do so requires the use of an AC-GAN network whereby the genre is the descriptive variable used. In order to decrease overfitting an AC-GAN network was first pre-trained using OMACIR and then trained using the Spotify dataset. A discriminative learning rate of $2 \cdot 10^{-5}$, a generative learning rate of $1 \cdot 10^{-5}$ and a batch size of 128 were found to be optimal.

One major flaw when training AC-GANs is that the generator may collapse and always output the same image. One popular method of tracking diversity amongst classes is the use of multi-scale structural similarity (MS-SSIM) [7]. MS-SSIM is an extension of the well known structural similarity index. A high MS-SSIM index for a generated class indicates that there is little variation amongst generated images and as such the generator has collapsed. The MS-SSIM scores

[5] `https://developer.spotify.com/web-api/`

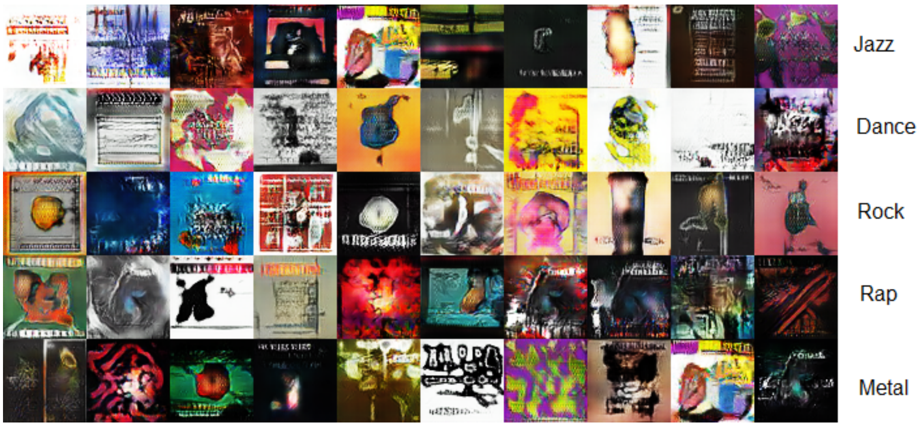
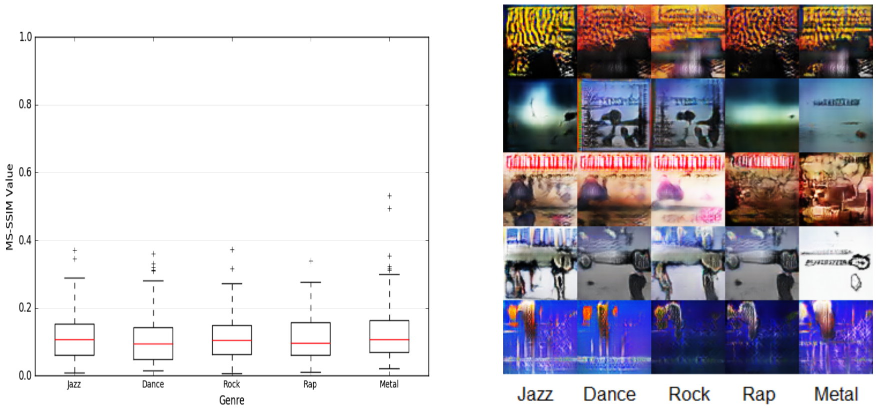


Fig. 2. Images generated from the final AC-GAN network. The network was pre-trained using the OMACIR dataset and then trained using the Spotify dataset.

between real and generated images within the same genre have a similar distribution to the MS-SSIM scores between just the real images, as shown in Fig. 3(a). This means that in terms of MS-SSIM, the real and generated images are interchangeable without affecting the MS-SSIM distribution. Although variance within classes is important, perhaps more important is being able to distinguish which class an image is generated from. Given a generated image, the cross-validated discriminator accuracy for genre classification is $35 \pm 2\%$. For images from the Spotify dataset, the network is able to correctly predict the genre with an accuracy of $47 \pm 4\%$. To establish a baseline for predicting genres of an album cover, a separate network was trained to predict which genre a real album cover belonged to. The network has the same architecture as the discriminator detailed in Table (1) and has a cross-validated accuracy of $59 \pm 4\%$. This implies that there can be improvements in combining both classifying genres and generating images into one network. To explore the visual characteristics of each class, images were generated using the same random and latent variables but with different genres. Fig. 3(b) shows that changing the genre has a different effect depending on the image, although general trends can be spotted. For example, rap covers are noticeably darker while jazz albums are overall lighter. Jazz and rap have more soft light colours whereas the rest have more black harsh shapes, however they all have a similar colour palette. This means the image structure or colour palette is represented in the latent and random variables whereas the style is specified by the genre. This is a positive result as different genres can use the same objects on their album covers but they each have an distinguishable style to them.



(a) MS-SSIM for each genre between 1000 real and 1000 generated examples.

(b) Effect of using same noise and latent variables but different genre.

Fig. 3. Genre diversity of images generated from the AC-GAN network.

4 Conclusions

We have explored the conditional generation of album cover art using AC-GAN architectures, using genre labels in the process. Overall the conditional generation of 64x64 album covers given a genre is possible, although there are still repeated textures in the new images. Using AC-GANs opens up opportunities to include more information about albums when generating cover art although larger images will need to be generated for this to become feasible for a platform such as SoundCloud.

References

1. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In: NIPS (2015)
2. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS (2014)
3. Huang, A., Wu, R.: Deep learning for music. arXiv:1606.04930 (2016)
4. Im, D.J., Kim, C.D., Jiang, H., Memisevic, R.: Generating images with recurrent adversarial networks. arXiv:1602.05110 (2016)
5. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. arXiv:1610.09585 (2016)
6. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR (2016)
7. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: Asilomar Conference on Signals, Systems and Computers. vol. 2, pp. 1398–1402 (2003)

Music style recognition with language models – beyond statistical results

María Hontanilla, Carlos Pérez-Sancho, and Jose M. Iñesta

University of Alicante, Spain
mariahontanilla@yahoo.es, {cperez,inesta}@dlsi.ua.es
http://grfia.dlsi.ua.es

Abstract. In previous works we have shown that $n-$gram language models may be successfully applied to style recognition tasks. Now we present a different study, beyond the purely numerical results, examining more closely some of those results. We have found that some of the compositions might be considered as 'outliers' from a musical point of view, and newer experiments allow us to confirm those musical analyses, showing that our models' musical features are useful for this task.

Keywords: $n-$gram language models, style recognition, composer recognition

1 Introduction

Music style recognition has been an increasing research field in the last years. Up to now, most of the studies done aim to perform a better classification task, according to the success rate.

Sometimes it is difficult, even to expert musicians, to distinguish between composers when their styles are quite similar, often because the compositions belong to the same stylistic context or time, but also because sometimes composers wrote some of their pieces as an homage to other composers, writing in their style. Apparently, the classification should be easier between composers that are further in time than between those temporally closer. Particularly difficult is the known task of distinguishing between Mozart and Haydn: both composers not only shared the same style, but they even dedicated some of their string quartets to the other. A quiz with human listeners performed by the CCARH center at Stanford University [4] shows that non-experts where capable of properly identifying the composer of a 51% of the pieces, whereas this figure raises to only a 66% for self-reported expert listeners. In another recent work [5], the authors reached 80.4% success rate for this task, using a visual representation of musical scores and support vector machines.

When doing classification, maximizing the success rate is usually the target. In this paper we take a second view over the results, trying to find, if possible, a musical explanation to them. Starting from previous works [1], we take a closer look at some of the individual musical pieces, finding that, in some cases, the classifier grounds its decision on very small differences among several models. So the question is: should those results be rejected? On the other hand, we have observed that some of the results seem not to correspond with the general style of the composer, i.e., there are some compositions whose numerical results are far from the mean of the rest of pieces in the training corpus.

These observations have led us to another important question: are those pieces that seem to be quite different from the composer's style different from a musical point of view too? A simple musical comparison has shown that, in some cases, some of those pieces are atypical within the whole musical production of the composer. Then, we have performed new $n-$gram experiments in order to support this musical analysis. The results allow us to verify that, indeed, they agree with this musical analysis.

As an example, we will show two of the examples observed: the first movement from Mozart's string quartet KV 158 as an outlier, and the fourth movement from quartet KV 168 as a tie.

2 Corpora and methodology

2.1 Methodology

From MIDI files, simple musical features (relative pitch intervals and duration ratios) are extracted and converted to ASCII characters, following [2], so that every MIDI file becomes a character sequence. For details on the method, the reader is referred to [1]. Although the MIDI files considered are polyphonic, they are structured in different tracks per voice, so for this particular study, as explained below, only the soprano track (upper voice) will be considered and encoded. Then, an $n-$gram model is built with all the text sequences of every composer, following a leave-one-out scheme. The smoothing method applied is a simple linear interpolation with models of lower $n-$order:

$$\begin{aligned} p_{\mathrm{I}}(w_i|w_{i-n+1}\cdots w_{i-1}) = {} & \lambda_n p_{\mathrm{V}}(w_i|w_{i-n+1}\cdots w_{i-1}) \\ & + \lambda_{n-1} p_{\mathrm{V}}(w_i|w_{i-n+2}\cdots w_{i-1}) \\ & + \cdots \\ & + \lambda_1 p_{\mathrm{V}}(w_i) \\ & + \lambda_0 p_{\mathrm{U}}(w_i) \end{aligned} \tag{1}$$

where p_{I} stands for the interpolated probability of the $n-$gram $(w_{i-n+1}\cdots w_i)$, being w_i a word of the music word sequence $w = w_1 \cdots w_k$, p_{V} is the maximum likelihood estimator, and p_{U} is the uniform probability distribution. The weights $\lambda_n \cdots \lambda_0$ are adjusted using a validation set. For that, the whole data set for a particular author is divided in 5 parts: 4 parts are used for training and 1 part for validation. This is done 5 folds, obtaining 5 different models.

For a given new target work w, its perplexity against each model c is computed as:

$$PP_c(w) = \sqrt[k]{\frac{1}{\prod_{i=1}^{k} p_I(w_i|w_{i-n+1}\cdots w_{i-1})}} \tag{2}$$

These results are averaged, obtaining a mean perplexity and a deviation for the work.

2.2 Corpora

In our previous work [1] five composers were studied and their works compared. The works used were: 52 from G. Ph. Telemann (1681–1767), 51 from G. F. Händel (1685–1759), and 75 from J. S. Bach (1685–1750) from the Baroque style; and 46 from W. A. Mozart (1756–1791) and 49 from F. J. Haydn (1732–1809) from the Classical style. This corpus, which is varied in instrumentation, musical form, and number of pieces, is the same as in [3]. For the present analysis we have focused in the results obtained for individual works, trying to identify those that deserve a closer study.

As shown in [1], the upper voice seems to be useful enough to perform a good classification task, though its results are lower than those achieved when using all the instrument's information. For simplicity in the music analysis, we have done the new experiments using information from the upper voices only.

3 Results

Figure 1 shows the results for every piece of one of the models for the same composer. As an example, we show the results of Mozart using a *decoupled* representation for pitch and duration (both properties for each note are represented independently) and $n-$gram length $n = 3$. In this graph, the horizontal lines represent the mean perplexity and the vertical ones represent the range of the standard deviation. The result labeled `MODEL` is the result of the whole model, followed by the individual results of every test piece in the model.

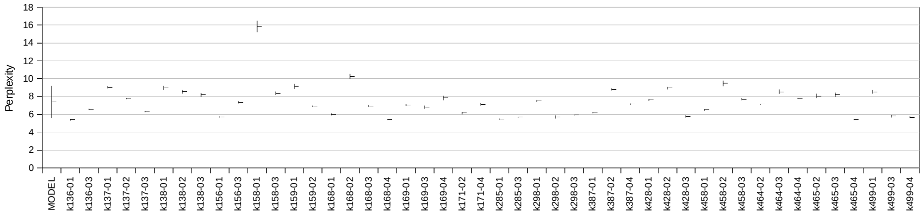


Fig. 1. Graphical comparison of the perplexity of every Mozart's quartet in the training set against a model built with the rest of Mozart's works in it, using independent representations for pitch and duration, and $n = 3$.

3.1 Mozart's string quartet KV 158, 1st movement

Here we can see that there is a piece (KV 158-01) whose result is significantly appart from the whole model (we consider to be significantly distant those pieces whose mean is more than 2 × *stdev* from the mean of the model). Therefore, the question arises as to whether it is actually a musically atypical piece in Mozart's style or it has some particular features not present in the rest of this particular corpus. We have observed that this is indeed a non typical Mozart's piece, particularly in the rhythm of the main melodic motif (Figure 2), which was not an usual rhythm in that period, and that it would rather seem to be a sort of musical innovation or a kind of surprise tried by Mozart in this first movement of the string quartet.

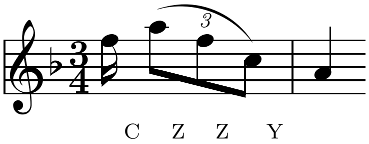


Fig. 2. Main melodic motif from the beginning of the first movement of Mozart's string quartet KV 158 and its rhythmic representation. The characters are the encoding [2] of the inter-onset ratios of every pair of consecutive notes (e.g. 'Z' encodes the ratio = 1 of a pair of equal duration notes).

In order to verify that the anomaly is found in the rhythm, we have performed two additional experiments using both the melodic and rhythmic information alone, and we have compared their results with the previous experiment using them combined with the decoupled representation. These results are shown in Table 1, where it can be observed that the rhythmic feature is the discordant one, as its perplexity is much further from the whole model than when using just pitch intervals.

Table 1. Mean and standard deviations of the perplexity obtained by the model and the first movement of the string quartet KV 158, using intervals only, durations only, and both combined.

	3−grams		
	Intervals	Duration ratios	Intervals and duration ratios
Model	10.6 ± 1.9	**3.8 ± 1.3**	7.4 ± 1.8
KV 158-01	14.2 ± 0.4	**10.0 ± 0.6**	15.9 ± 0.6

We have additionally done a ranking of the rhythmic 3-grams generated from this piece. We have found that, besides the expected most frequent 3-gram (`Z Z`

Z, representing three consecutive notes of the same duration), the two other most frequent 3-grams in this movement belong to this rhythmic motif, whereas they are extremely rare in the whole model. These frequencies are shown in Table 2 and allow us to state that the peculiarity of this piece lies in the rhythm, which is not a usual rhythm in Mozart's language.

Table 2. Frecuencies for the most seen rhythmic 3-grams in the 1st movement of Mozart's quartet KV 158 and for the same 3-grams in the whole Mozart corpus.

	Frequency (%)	
3-gram	KV 158-01	All
Z Z Z	9.9	37.1
Z Z Y	6.8	1.8
C Z Z	6.5	0.1

3.2 Mozart's string quartet KV 168, 4th movement

On the other hand, when comparing every piece against every model, we can see that some of the classification decisions are taken by a very small difference between the models. We wonder whether these decisions should actually be taken into account, or whether they should be addressed in a somehow different way. As an example, we show the results of the fourth movement of Mozart's string quartet KV 168 (Figure 3) against all the five composers' models.

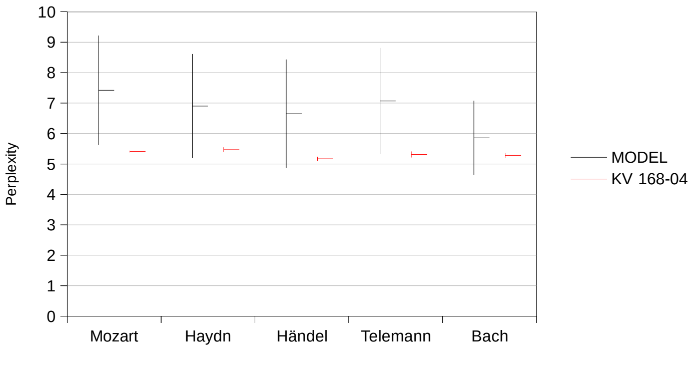


Fig. 3. Comparison of 4th movement from Mozart's quartet KV 168 against every model, using pitch and duration independent representations and $n = 3$. Model perplexities are the averages and deviations for all the Mozart pieces in the dataset.

This movement is actually a fugue, and this might be the reason why the system founds itself in trouble to distinguish one of the composers as the actual author. This is probably due to, although the fugue was a musical form specially preferred in the Baroque period, other composers from different periods have used it as well in their compositions. In this case, the system succeeds in recognizing a style which is closer to the Baroque period than to Mozart's and Haydn's Classical style, assigning a lower perplexity to these composers, but fails in selecting its actual author.

4 Conclusions and future work

From the results shown in this paper, we think that it is desirable going beyond the simple statistical analysis when trying to classify musical compositions in the style of the studied composers. We have realized that the classification of some of the pieces is done with a very little difference from one model to another, and also that there are some pieces that do not fit well in the model built from their author. This shows that it is quite difficult to build a model that is general enough to capture the style of a composer, while being able to identify the subtleties of each individual piece at the same time.

However, a musical analysis on a few pieces has shown that the numerical results for them agree with these analysis indeed, so we think that our system is still able to capture some of the musical features characterizing music styles. Nevertheless, deeper research with other musical compositions needs to be done in this way, in order to study whether these results might be generalized and, if so, to open a discussion about what to do with this kind of pieces when doing classification tasks. Another open question to be addressed is whether or not these models are able to capture higher level stylistic traits, such as musical form, in other to answer some issues that have arised during this study.

References

1. Hontanilla, M., Pérez-Sancho, C., and Iñesta, J. M. (2011) Composer recognition using language models. In: Proc. of Signal Processing, Pattern Recognition, and Applications (SPPRA 2011), pp. 76–83.
2. Doraisamy, S., Rüger, S. (2003) Robust polyphonic music retrieval with n-grams. Journal of Intelligent Information Systems, **21**(1): 53–70.
3. van Kranenburg, P., Backer, E. (2004) Musical style recognition - a quantitative approach. In: Proc. of the Conference on Interdisciplinary Musicology (CIM 2004), pp. 106–107.
4. The Haydn/Mozart String Quartet Quiz. [online] Available at: http://qq.themefinder.org/ [Accessed Sep. 2017].
5. Velarde, G., Weyde, T., Cancino Chacón, C., Meredith, D., Grachten, M. (2016) Composer recognition based on 2D–filtered piano–rolls.In. Proc. of the 17th International Sociecy for Music Information Retrieval Conference (ISMIR 2016), pp. 115–121.

Towards Predicting the Popularity of Music Artists

Fabian Jetzinger, Florian Huemer, Markus Schedl

Johannes Kepler University
Department of Computational Perception
Linz, Austria
markus.schedl@jku.at,
http://www.cp.jku.at/people/schedl

Abstract. This paper explores the possibility of predicting a music artist's future popularity, quantified by how often their tracks are listened to in the past, on a daily basis. Using the LFM-1b dataset of listening histories by Last.fm users, we investigated three regression techniques to predict the amount of listening events an artist will generate per day. To this end, we adopt linear regression, support vector machines, and neural networks to create, analyze, and optimize predictions, which we finally visualize for easy exploration.

Keywords: music, regression, popularity prediction

1 Introduction

How will a music artist's popularity evolve over the next month? This question lays at the basis for our work. The creation of accurate predictions of an artist's future popularity offers a multitude of possible applications, such as in music recommendation systems or as a decision guidance for investors or music labels. To the best of our knowledge, such popularity prediction experiments in the music domain have only been conducted on rather small datasets, exploiting only content features or peer-to-peer networks, the latter having faced a substantial decrease in usage during the last few years, due to the emergence of streaming services like Spotify, Apple Music, or Last.fm. In this paper, in contrast, we exploit a large-scale dataset (LFM-1b) of more than a billion user-generated listening events. Our goal was the generation of predictions and the subsequent continuous optimization of the predictive algorithms to increase accuracy. In this comparative study, we relate the results achieved with different regression approaches to determine which method generates the most accurate predictions.

2 Related Work

Similar work has already been undertaken by various researchers. Staying within the topic of music, one highly relevant work is [8], in which Pachet and Roy use content-based features for popularity prediction of music items, with limited success though. Dhanaraj et al. [3] try to predict hit songs based on extracted acoustic and lyrical information. They ultimately conclude that their lyrics-based features produced slightly more accurate results. Similarly, Herremans et al. [2] explore the prediction of dance hit songs based on several different classifiers and a database of dance hit songs from 1985 to 2013. Furthermore, Ni et al. [7] investigate the prediction of hit songs based on the UK top 40 charts of the last 50 years, with the aim to distinguish songs with their peak positions within the top 5 from songs which peak in the top 30 to 40. Using web sources instead of audio, Schedl et al. [10] determine country-specific popularity of music artists. They investigate search engine playcounts, popularity derived from Twitter, from shared folders in the peer-to-peer network Gnutella, and from Last.fm playcounts. Their conclusion is that these sources are largely inhomogeneous and yield to different popularity scores. Koenigstein and Shavitt [6] try to forecast the Billboard charts based on search queries issued within Gnutella. They show that a songs popularity in the network highly correlates with its ranking in the Billboard charts.

In the multimedia domain, Bandari et al. [1] predict the popularity of news items prior to their release to the public, achieving an overall accuracy of 84%. Yu et al. [11] explore the effect that Twitter contributions have on the amount of views a YouTube video receives over a certain time span, differentiating between sudden increases in viewcount, named “Jumps”, and the initial viewcount a video receives shortly after its upload, named “Early”.

3 Experiments and Results

3.1 Dataset

The LFM-1b dataset [9] used in our work contains information on users, artists, tracks, and listening events. The dataset contains more than 1 billion listening events for more than 3 million individual artists. Listening events, which constitute the main building block for our experiments, are defined by a specific date and time and the corresponding information about track and user. We considered in our experiments the top 100 artists according to number of total listening events to ensure a sufficient amount of data.

Before we were able to start working on the actual predictions, we first had to aggregate the LFM-1b data and transfer it into a suitable database structure (using SQLite[1]) as we were interested in the total number of listening events per artist per day, rather than the raw data contained in the LFM-1b dataset’s [9] listening events file.

3.2 Experimental Setup

To generate a prediction, we use a certain number of past days, which can be specified individually for each experiment. Each value in the feature vector

[1] https://www.sqlite.org

constitutes the number of listening events the specified artist accumulated that day, i.e., over all the artist's tracks. Based on these feature vectors, the goal of the algorithm is to calculate a single value representing the amount of listening events the artist would receive a certain number of days after the last known value. More formally, we use a time series $LE^a_{t_0 \cdots t_N}$ of listening events for artist a, starting at day 0 up to day N, where N is the same number for all artists. We then train different regressors to predict $LE^a_{t_{N+1} \cdots t_{N+M}}$, where M is the time period to forecast, in days.

We investigate variants of linear regression, support vector machines, and neural networks, as provided in the scikit-learn[2] Python package, for our regression task and measure accuracy in terms of the R^2 metric.

Linear Regression As a fairly simple but efficient algorithm, linear regression represented our first approach to create predictions. We were not expecting this method to generate accurate results, instead viewing it as a first step towards further optimization. We did, however, quickly realize that with fairly little optimization, the results achieved with linear regression already appeared to be promisingly accurate, as fairly early tests already achieved an average R^2 value of 84%.

Support Vector Machines We next investigated epsilon-support vector regression [4], which is based on a more sophisticated methodology than linear regression and allows for a more complex range of options concerning the optimization of the algorithm to the specific task at hand. In typical classification problems, support vector machines perform a non-linear transformation on the data, allowing the model to separate the classes more easily. In regression use-cases, such as ours, a line of best fit is calculated instead and the parameter ϵ is introduced as a tolerance range, hence the name epsilon-support vector regression. The algorithm's behavior is strongly dependent on the specified kernel, which is represented by different mathematical functions. For our purposes, we assessed linear, radial basis function (rbf), and polynomial kernels (poly). Overall, using the linear kernel yielded similar results to linear regression, with average R^2 scores of around 83%. When using the kernels rbf and poly, further fine-tuning can be made via the parameters ϵ, C, and γ. Epsilon determines the size of the tolerance range for data that significantly deviates from the calculated model. The tolerance penalty C specifies how harsh data outside this tolerance range should be penalized. Finally, γ determines the intensity of the influence a single data point can have on the overall model. We continuously tweaked these parameters by hand, constantly analyzing the results and comparing them to previously achieved ones. We achieved the best results when using the rbf kernel with $\gamma = 0.00001$, $\epsilon = 1.0$, and $C = 625$, which accomplished an average R^2 value of over 86%.

[2] http://scikit-learn.org

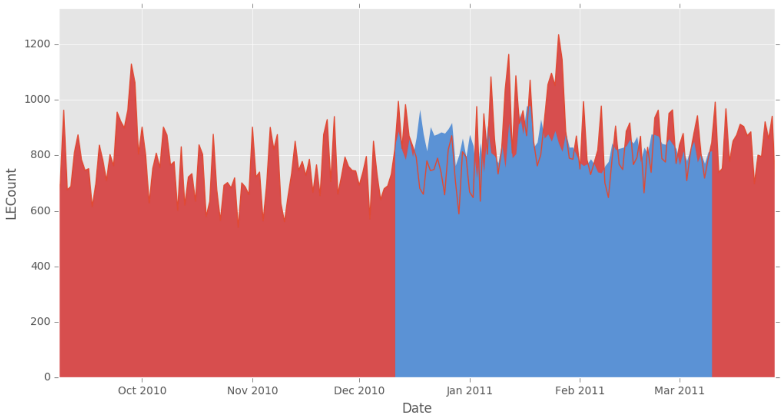


Fig. 1. Prediction over 90 days for Metallica using *linear regression*, December 2010

Neural Networks Artificial neural networks are an advanced machine learning methodology that tries to solve problems based on a layered structure of nodes. We used feed-forward neural networks [5], where each node of a layer is connected to each node in the layer above and below it. As the main purpose of our work with neural networks was getting the most accurate predictions based on a specified number of listening events, the most essential part was the configuration of the network itself. We used a sliding window approach to train our neural network where the window size is 120 days. To determined the best solver, we compared the accuracy of all solvers provided by scikit-learn (sgd, adam, lbfgs) and eventually determined that the lbfgs solver was best suited for the amount of data available in our dataset.The second configuration step was to choose the activation function, for which we used a linear model due to accuracy and consistency of the achieved results. Lastly, we determined the amount of layers and nodes. We chose one hidden layer and increased the number of nodes until further change produced no noticeable differences in results and ended up with 120 input nodes, 16 hidden nodes and one output node. Our best results achieved with neural networks in terms of the R^2 score were around 91%.

3.3 Results and Discussion

As illustrated in Figures 1 and 2 for Metallica, using respectively linear regression and neural networks, the achieved results all appear to be fairly plausible predictions, regardless of the applied algorithm. Red areas represent the true evolution of listening events, blue areas the predictions. Naturally, the longer the predicted time span, the less accurate the achieved results are. Additionally, the amount of available data is a strong limiting factor, meaning that predictions for a well-known artist are usually significantly more accurate than those generated

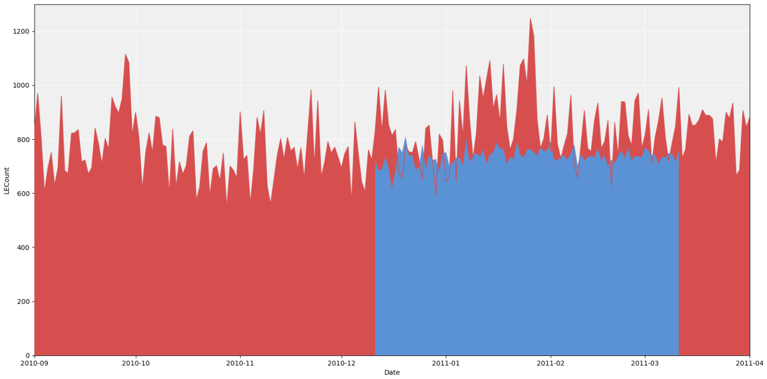


Fig. 2. Prediction over 90 days for Metallica using *neural networks*, December 2010

for an underground band. Achieved results did, however, also strongly depend on the chosen time span. Approaching average R^2 scores of slightly over 93%, some of our best results using linear regression were attained with The Beatles in 2012. Overall, we achieved the highest R^2 scores for artists like The Beatles, Metallica, and Pink Floyd, which we ascribe to the fact that these artists were already well established and fairly popular throughout the time span our data covered. For these artists, all of our applied methods reached average R^2 values of 89% to 94%, with the best scoring predictions lying within a time span of 2012 to 2014. Naturally, we found that artists which exhibit significant jumps or spikes in popularity were much harder to create accurate predictions for. For example, when trying to predict the popularity of Daft Punk in 2013, average R^2 scores of our support vector machine algorithm dropped to around 43%, while linear regression scores sank to 39%.

4 Conclusions

In conclusion, we find that all three regression techniques generate surprisingly accurate results when predicting well established artists, e.g., The Beatles (R^2 of 89% using linear regression) or Metallica (94% using support vector machines). Each technique does, however, possess certain advantages and disadvantages. Linear regression is fairly simple and quick to implement and understand and exceeded our expectations in regards to its accuracy, but is most likely still not the best suited option for real life applications of such problems due to its simplicity. Support vector machines offered slightly higher accuracy and more consistency over artists than linear regression, but performance quickly became a limiting factor when using a larger number of features or predicting a longer time span. Neural networks, on the other hand, probably constitute the best option in our eyes as they allow to use a large number of features (preceding days), which boosted the achieved accuracy, and were also able to generate adequate predictions further into the future.

For future work, we contemplate many ways in which the predictive algorithms could be improved. One of the most obvious and probably also most effective approaches would be to take recent album releases into account when creating predictions. Another idea would be observing social media activity pertaining to specific artists.

Acknowledgments

This research is supported by the Austrian Science Fund (FWF): P25655.

References

1. R. Bandari, S. Asur, and B. A. Huberman. The Pulse of News in Social Media: Forecasting Popularity. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (ICWSM)*, Dublin, Ireland, June 2012.
2. K. S. D. Herremans, D. Martens. Dance hit song prediction. *Journal of New Music Research*, 43(3):291–302, 2014.
3. R. Dhanaraj and B. Logan. Automatic prediction of hit songs. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, London, UK, September 2005.
4. H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. N. Vapnik. Support Vector Regression Machines. In *Advances in Neural Information Processing Systems 9*, pages 155–161, Cambridge, MA, USA, 1996. MIT Press.
5. G. E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40:185–234, 1989.
6. N. Koenigstein and Y. Shavitt. Song Ranking Based on Piracy in Peer-to-Peer Networks. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, October 2009.
7. Y. Ni, R. Santos-Rodríguez, M. McVicar, and T. D. Bie. Hit song science once again a science? In *4th International Workshop on Machine Learning and Music*, Sierra Nevada, Spain, December 2011.
8. F. Pachet and P. Roy. Hit Song Science is Not Yet a Science. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, Philadelphia, PA, USA, September 2008.
9. M. Schedl. The LFM-1b Dataset for Music Retrieval and Recommendation. In *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR)*, New York, USA, June 2016.
10. M. Schedl, T. Pohle, N. Koenigstein, and P. Knees. What's Hot? Estimating Country-Specific Artist Popularity. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, the Netherlands, August 2010.
11. H. Yu, L. Xie, and S. Sanner. Twitter-driven YouTube Views: Beyond Individual Influencers. In *Proceedings of the ACM International Conference on Multimedia*, Orlando, Florida, USA, November 2014.

Data augmentation for deep learning source separation of HipHop songs

Hector Martel and Marius Miron

Universitat Pompeu Fabra, Music Technology Group, Barcelona
hector.martel01@estudiant.upf.edu

Abstract. Training deep learning source separation methods involves computationally intensive procedures relying on large multi-track datasets. In this paper we use data augmentation to improve hip hop source separation using small training datasets. We analyze different training strategies and data augmentation techniques with respect to their generalization capabilities. Moreover, we propose a hip hop multi-track dataset and we implemented a web demo to demonstrate our use scenario. The evaluation is done on a part of the dataset and hip-hop songs from an external dataset.

Keywords: Music Source Separation, Deep Learning, Hip Hop

1 Introduction

Audio Source Separation involves recovering individual components from an audio mixture [8]. This task is related to auditory scene analysis, however it is difficult for the current algorithms to match human ability of segregating audio streams. Matrix decomposition techniques such as Non-Negative Matrix Factorization (NMF) [3], were traditionally used for audio source separation. NMF is particularly popular in this field because of its additive reconstruction properties. However, the NMF iterative procedure is computationally expensive in contrast to newer approaches using deep learning [2]. Furthermore, frameworks as [3] rely on a pitch detection stage and assume a voiced source.

Deep Neural Networks model source separation as a regression problem, taking as an input time-frequency representations such as Short-term Fourier Transform (STFT) magnitude spectrograms, and estimating a continuous output, the magnitude spectrograms for the sources [2,5,7]. Because the estimation assumes a single feed-forward pass through the network, deep learning frameworks are less computationally intensive than NMF [2]. However, deep learning models are expensive to train and require large datasets with isolated instruments [7], which are difficult to obtain. Furthermore, data driven methods can often overfit and fail for a particular test case, which might represent a different problem in itself. To that extent, data augmentation [1] is a regularization technique that increases the robustness of an already trained model and boosts its performance on unseen data.

In this paper we study the use of data augmentation to retrain a general purpose music source separation model for hip hop music, on very small datasets. We are interested in assessing the generalization capabilities of the models trained with such data.

For the experiments we use the Convolutional Neural Network (CNN) autoencoder [1] in [2] which separates pop-rock music with low latency. The baseline architecture comprises an encoding and a decoding phase. At the encoding phase we have a vertical convolution which models timbre characteristics, a horizontal convolution which models temporal evolution, and a dense layer with a low number of units which acts as a bottleneck. The decoding phase assumes performing the inverse operations of the layers in the reverse order, namely another dense layer and two deconvolutions.

We follow the research reproducibility principles and publish the dataset, code, and a web demo.

The remainder of this paper is structured as follows. In Section 2 we present the use scenario, followed by the proposed dataset in Section 3. In Section 4 we evaluate the use scenario and discuss the results. The conclusions are presented in Section 5.

2 Use scenario

HipHop music is an interesting scenario for source separation. The most noticeable characteristic is that the voice is not sung. Thus, pitch-based methods [3] would not work properly to extract the vocals. Furthermore, the drums and the bass can be acoustic, synthesized or sampled from vinyl records, making the timbre variability of the sources very high.

Our use scenario is remixing or upmixing recordings [4] in the same production style, where the instrumentals are created by a single producer and the voices come from different musicians. Furthermore, we are interested in live remixing, where latency plays a crucial role in the overall performance, and it is advantageous to use a deep learning system. Such a system can be used by a music producer or DJ to manipulate songs within a certain genre or production style to play them live.

3 Dataset

3.1 Proposed dataset

We propose a compilation of Hip Hop songs, referred to as HHDS [1], which can be used to train a neural network. The structure of HHDS follows the convention of DSD100 [2] (Demixing Secrets Dataset). HHDS contains the separated tracks for the categories of bass, drums, vocals and others in monophonic WAV files

[1] HHDS, on Zenodo: `http://doi.org/10.5281/zenodo.823037`

[2] Demixing Secrets Dataset (DSD100), SiSEC2016: `http://liutkus.net/DSD100.zip`

with a sampling rate of 44100Hz. The mixture is calculated by normalizing the sum of the tracks. The main difference with respect to DSD100 is that in HHDS there are HipHop songs only, instead of many different genres. The total number of songs is 18, from which 13 are used for training and 5 are used for evaluation. The songs are mixed by one producer and contain vocals in Spanish from 12 different musicians to maximize timbre diversity for voice. More details about the dataset can be found at the repository page.

3.2 Data Augmentation techniques

A deep learning model can become more robust through data augmentation techniques which create more training instances [6]. To that extent, we choose transformations which are relevant for source separation and are applied to the audio signal, rather than the STFT magnitude spectrogram. Thus, similarly to [6], we discard other popular transformations such as pitch shifting and we analyze the following augmentation techniques:
a) Instrument Augmentation (IA) [7]. More renditions of the same song can be created by muting one of the instrument tracks. This transformation is useful modeling hip hop cases, e.g. an instrument does not play in certain sections.
b) Mix Augmentation (MA) [7]. We sum instrument tracks from different songs to create a new mix. The tracks are combined and picked randomly. This transformation de-correlates the harmonic relation between the instruments within a mix, however it provides more training examples of different timbre combinations.
c) Circular Shift (CS) [5,6]. The audio signals corresponding to instrument tracks are shifted between each other with a fixed number of time frames. With this transformation we introduce small temporal deviations of $0.1, 0.2$ seconds which make the network more robust to various time patterns. While temporal alignment of the instrument tracks is slightly modified, the structure of the song does not change.

4 Evaluation

4.1 Experimental setup

a) Parameters To train the network, the spectrograms are passed through it iteratively for 40 epochs using the parameters of the baseline method [2] with mini-batch stochastic gradient descent.
b) Evaluation metrics We use the objective measures proposed in [8]: Source to Distortion Ratio (SDR) as a global quality measure, Source to Interference Ratio (SIR) related to the interferences from other sources, Source to Artifacts Ratio (SAR) related to the presence of artifacts. All measures are expressed in decibels (dB).

4.2 Experiments

The experiments evaluate 8 models, described in Table 1. A generic model trained with DSD100 [2] is used as a reference. 2 models are retrained from the reference using HHDS, and 5 models are trained with HHDS and data augmentation techniques.

Table 1. Models generated during the training phases.

DSD	Trained with DSD100 songs only, used as a reference.
DSD_HH	Retrained from DSD100 with HHDS in a new training.
DSD_HH_2	Retrained from DSD100 with HHDS in a new training, with a lower learning rate for 10 epochs.
HH	Trained with HHDS songs only, used as a specialized case.
HH_COMBI	Trained with HHDS and all the Augmentations combined.
HH_CS	Trained with HHDS and Circular Shift Augmentation.
HH_IA	Trained with HHDS and Instrument Augmentation.
HH_MA	Trained with HHDS and Mix Augmentation.

4.3 Results

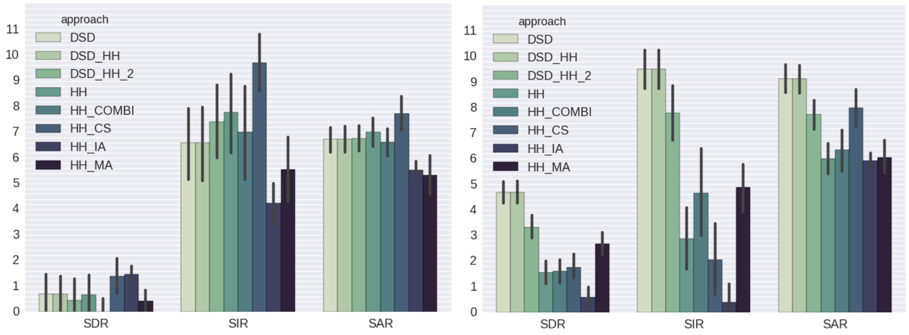


Fig. 1. Results in terms of SDR, SIR, SAR for HHDS Test (left) and the HipHop songs from DSD100 (right).

We evaluate the models for two different contexts involving different production styles, first, on the test set from HHDS (Figure 1 left), and, second, on the 3 songs labeled as HipHop from test set of DSD100 (Figure 1 right) which are not used to train any of the models. In the corresponding figures, error bars are drawn for a confidence interval of 95%. Note that the songs from HHDS comprise one production style, with mostly synthetic drums, while the ones in DSD100 have mostly acoustic drums and contain a more variety of production styles.

As seen in Figure 1 left, the generic model trained on the DSD set, DSD, has lower performance than the models trained on HHDS dataset comprising tracks with similar production style. As expected, the generic model performs better on the DSD100 hip hop set (Figure 1 right) because it models better the acoustic drums and it was trained with larger variety of timbres.

Training the model from scratch on HHDS, HH, improved 0.7dB SDR over the generic model, DSD, while decreasing 3dB over the 3 DSD100 songs which are created in a different production style. Further improvements of 3dB over the generic model and 1.5dB over the HH, are obtained with Circular Shift (CS) augmentation in HH_CS. This augmentation makes the model more robust for unseen songs the same production style (Figure 1 left), however not for the 3 songs in DSD100 (Figure 1 right). This type of augmentation is helpful in our use scenario: remixing of recordings in the same production style.

The other two augmentation techniques do not improve the results on HHDS dataset, however the Mix Augmentation model (HH_MA) obtained more robust performance on the DSD100 songs: 2dB higher. Thus, creating more combinations between different not correlated tracks, keeps the performance stable on the target context and makes it more robust on songs from different production styles. The Instrument Augmentation model (HH_IA) did not improve the baseline method because the combinations created were not realistic.

The combination of all the augmentation techniques in HH_COMBI did not result in a significant improvement in any of the two contexts. It obtained 0.5dB lower performance respect to HHDS, achieving 0.2dB over the baseline method for the same production style. Similarly to HH_IA, the combinations generated are not realistic.

Surprisingly, DSD_HH, which involved initializing the model with weights from DSD, and then re-training with HHDS, did not improve over the generic model and over the model trained from scratch. Also in DSD_HH_2, trained with 10 epochs and a lower learning rate, the improvement is not significant. Further experiments are needed to assess these problems.

5 Conclusions

We have presented a source separation scenario that is well suited for the use of small datasets under the genre-specific assumption. A producer or DJ can use this system to train a model with very few songs and separate songs of the same style. From the experiments it can be extracted that the context-specific methods outperform the general purpose one for test cases similar to the training examples. For songs that differ from the training data the performance can be improved with data augmentation, achieving a more general representation. Thus, we found that there is a trade-off between the specialization of model and its performance under unknown test data.

Future research on this topic can be focused on the development of applications such as upmixing or remixing, as well as exploring more data augmentation techniques. It would also be interesting to expand HHDS with songs from other producers.

The reader is encouraged to check a web-based demo of this paper[3].

6 Acknowledgments

The TITANX used for this research was donated by the NVIDIA Corporation. This work is partially supported by the Spanish Ministry of Economy and Competitiveness under CASAS project (TIN2015-70816-R) and by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

References

1. Y. Bengio et al. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.
2. P. Chandna, M. Miron, J. Janer, and E. Gómez. Monoaural audio source separation using deep convolutional neural networks. *International Conference on Latent Variable Analysis and Signal Separation*, 2017.
3. J.-L. Durrieu, A. Ozerov, C. Févotte, G. Richard, and B. David. Main instrument separation from stereophonic audio signals using a source/filter model. In *Signal Processing Conference, 2009 17th European*, pages 15–19. IEEE, 2009.
4. D. Fitzgerald. Upmixing from mono-a source separation approach. In *Digital Signal Processing (DSP), 2011 17th International Conference on*, pages 1–7. IEEE, 2011.
5. P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Deep Learning for Monaural Speech Separation. *Acoustics, Speech and Signal Processing (ICASSP)*, pages 1562–1566, 2014.
6. M. Miron, J. Janer, and E. Gómez. Generating data to train convolutional neural networks for classical music source separation. In *Sound and Music Computing*, 2017.
7. S. Uhlich, F. Giron, and Y. Mitsufuji. Deep neural network based instrument extraction from music. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 2135–2139. IEEE, 2015.
8. E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech and Language Processing*, 14(4):1462–1469, jul 2006.

[3] `https://hiphopss.github.io/`

Phrase-Level modeling of expression in violin performances

Fábio J. M. Ortega, Sergio I. Giraldo, and Rafael Ramírez

Music Technology Group, Machine Learning and Music Lab, Department of Communication and Information Technology, Pompeu Fabra University, Barcelona, Spain
fabiojose.muneratti@upf.edu

Abstract. A model is proposed for predicting expressive variations in dynamics for violin performances with the purpose of facilitating expressive performance learning by students. The model uses phrases rather than single notes as units of analysis in a lazy learning approach: each phrase in a new score is matched to phrases from expressive performances by experts, adapting the experts' transformations to render an expressive performance of the new score. In preliminary tests, the model approximates the dynamics of actual performances better than an unexpressive baseline model whenever the reference dataset contains melodies similar to those being predicted.

Keywords: expressive music performance, machine learning, music information retrieval

1 Introduction

Expression in music can be understood as the variations in timing, dynamics, pitch, timbre, and other features introduced by musicians as they play. Teaching musical expression traditionally relies on the continuous feedback that a face–to–face setting can provide [1]. When practicing an instrument on one's own, the absence of expert supervision makes acquiring this skill much harder, leading to frustration and high abandonment rates among students [2,3]. If, however, the information about how to play expressively could be generalized by a model based on some large set of recordings by professional musicians, a system could be devised that would be able to provide real–time feedback to students practicing any piece, even if no sample performance of it exists.

In this paper we propose a model for predicting dynamics for violin performances which mimics the process by which a musician would choose to interpret a melody based on their memory of a similar one. Our aim is to determine whether an automatic recognition of phrasing and melodic content present in a score can be used for selecting adequate examples of performance, and, if so, whether having these examples is enough to generate a plausible rendition of a piece.

2 Background

Several models of musical expression have been proposed for various instruments and purposes [4]. We highlight the most relevant comparisons. The *DISTALL* system [5], though designed for piano, can also produce dynamics predictions based on phrase–level analysis of performances, though they define phrasing hierarchically whereas we only focus on short *motivs.* Also, since the musicological analysis of scores is reportedly manual, it does not fit our requirement for being used in an expressive tutor system. Ramirez *et al.* [6] design a model for jazz saxophone that produces performance rules based on data via genetic algorithms. Besides focusing on note–level instead of phrase–level predictions, their approach is different by being rooted in classification (e.g. *piano*, *mezzo–forte*, *fortissimo*), with numerical values for synthesizing audio resulting from an *a posteriori* approximation. These same remarks apply to the model by Giraldo and Ramirez [7] for jazz guitar. Lastly, the basis–functions approach by Grachten and Widmer [8] relies on expressive markings in scores and their interpretation, whereas our model does not require annotated scores and applies very little musicological knowledge. Furthermore, all the discussed models assume a previous training step often very time–consuming before producing performances, whereas we are interested in taking a lazy learning approach that can be used to our favor for selecting the most relevant references for each performance prediction.

3 Materials and Methods

All data used in development come from recordings which were made as part of experiments on ensemble expressive performance [9,10]. A dynamics curve was calculated from the audio extracted from the pickup of the first violin in a performance of the fourth movement of Beethoven's String Quartet no. 4, Op. 18, purposely exaggerated in its expressiveness.

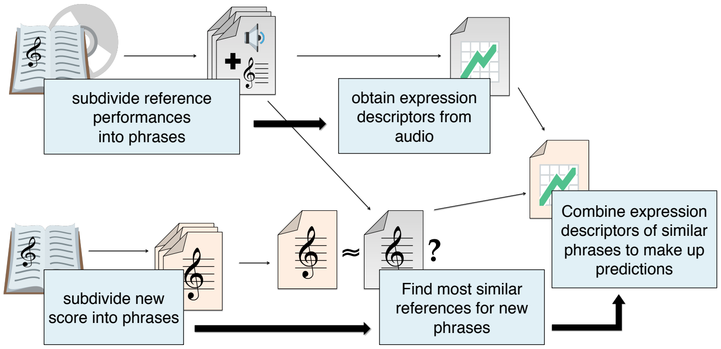

subdivide reference performances into phrases
obtain expression descriptors from audio
Combine expression descriptors of similar phrases to make up predictions
subdivide new score into phrases
≈
?
Find most similar references for new phrases

Fig. 1. Diagram of the steps taken to predict the dynamics for a new score.

The method for generating a prediction is depicted in figure 1. First, the score of the target piece is automatically segmented into phrases in a top–down approach based on the method by Cambouropoulos [11]. Then, ratings of melodic similarity are computed between each phrase from the score and every phrase in the database of references. A dynamic time warping algorithm is used for determining the degree of similarity between two phrases as proposed by Stammen and Pennycook [12]. The warping cost between phrases is interpreted as the distance between them, and the cost function takes pitch contour and note duration ratios into consideration. Predicted dynamics for each phrase may then be computed based on its closest matches.

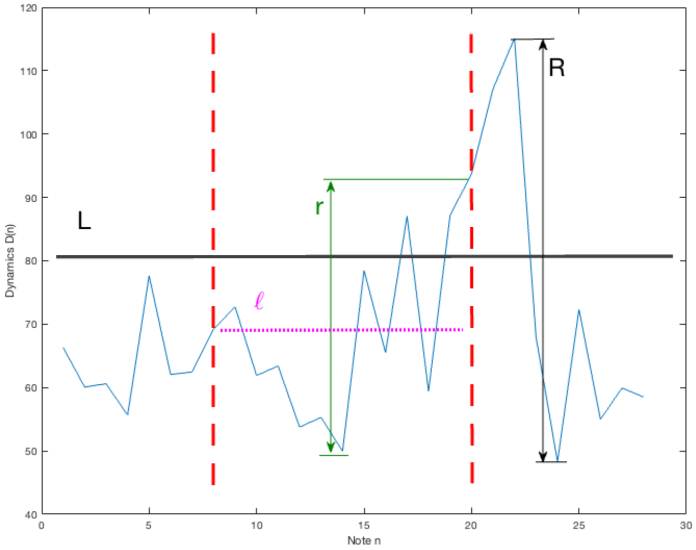


Fig. 2. Performed dynamics for a section of a piece and some key measurements.

Figure 2 represents a dynamics curve plot note by note where the dynamics at each note n is $D(n)$. Dynamics values are obtained as the logarithm of RMS values of audio samples within the (manually segmented) duration of each note, adjusted to the $0 - 127$ scale commonly applied to MIDI velocities. Between dashed lines is the section of a particular phrase in that piece. L is the mean level of the piece, whereas ℓ is the mean level of the phrase. The dynamic range is given by R for the piece and r for the phrase and we use the standard deviation of loudness values as their measure. Considering that pieces may be performed at widely different mean levels and dynamic ranges, if we intend to use phrases from multiple pieces as references for prediction it makes sense to measure their values relative to L and R and allow these to be set by the user for the predicted rendition. Therefore, the characterization of each phrase p in our model is given by three components:

$$\alpha_p = \frac{\ell_p - L}{R} \quad \beta_p = \frac{r_p}{R} \quad \mathit{\Gamma}_p(n) = \frac{D(n) - \ell_p}{r_p}$$

Where α_p represents the overall salience of p in relation to the piece, β_p is the relative dynamic range of the phrase, and $\mathit{\Gamma}_p$ represents the relative dynamics contour, that is, a function which describes how each note in a phrase contributes to its dynamics. Consequently, the dynamics at each note $k \in p$ can always be written as:

$$D(k) = L + R \cdot (\alpha_p + \beta_p \cdot \mathit{\Gamma}_p(k))$$

These three components are measured for each reference phrase and make up the target variables for our learning step. By predicting α, β and $\mathit{\Gamma}$ for all phrases of a target score, the above equation gives us the output prediction for freely chosen values of L and R.

4 Results

A preliminary analysis was conducted using a leave–one–phrase–out setting on the data from the Beethoven recording. Figure 3 shows the distributions of mean absolute error for each note using k–NN ($k = 1$), k–NN ($k = 1$) predicting $\mathit{\Gamma}$ as a quadratic polynomial, and k–NN ($k = 3$). For this last case, target variables take the mean values of the three nearest neighbors. The baseline is a mechanical, unexpressive prediction. From the plot it is visible that the quadratic polynomial was effective as an aproximation, and that k–NN ($k = 3$) is successful in reducing the instances with larger error values while maintaining mean absolute error MAE = 19.6 (15.44% of full–scale), which is lower than the baseline ($p = 1.56 \times 10^{-6}$ for one–sided t–test). Though it performs better than baseline, it should be noted that this is an advantageous case for the model, since the available reference phrases were part of the same performance.

In order to validate the hypothesis that phrases rated as melodically similar share similar dynamic profiles, we split the phrases predicted using k–NN ($k = 1$) in half, separating phrases with *closest* nearest neighbors in the training set from phrases with *farthest* ones. The mean absolute error in dynamics prediction for the phrases with closest neighbors is 8.83% of full–scale on average, whereas phrases with farthest neighbors show 15.77 %FS error on average, meaning predictions were more accurate for phrases which had instances in training data with a higher melodic similarity to them. This is an indication that the adopted measure of melodic similarity can be used as a predictor for performance dynamics, and also that given a larger number of performance examples, the model has a good margin for improving the quality of its predictions, since a wider range of melodies would be available as references.

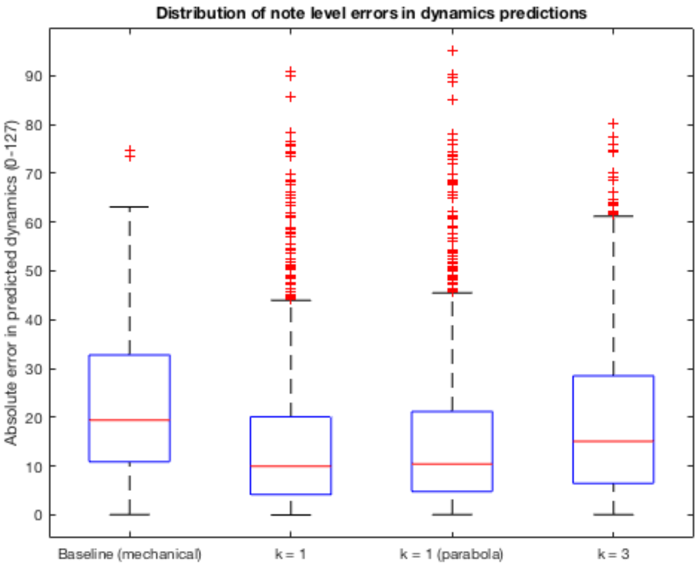


Fig. 3. Boxplot of the prediction errors in a leave–a–phrase–out approach vs. baseline.

5 Conclusions

We have devised a model for expressive dynamics prediction of solo violin performances based solely on the adaptation of performances of similar melodies. Both the proposed approach to characterizing expressive deviations in dynamics and the adopted measure of melodic similarity have been evaluated favourably, though with a limited dataset. Given the model's instance–based nature and the generality of its musicological assumptions, it should also be applicable to performances of other instruments. Furthermore, our data treatment for interpreting the contributions of each phrase to the expression of a musical piece may benefit other models as well. As previous models have shown, including other musical aspects deducible from the scores such as metrical strength and harmonic content should improve the quality of results. Also, an indirect dependency exists between generated predictions and phrasing boundaries, so including multiple phrasing interpretations could be an advantageous trait. As a logical next step, the authors are now preparing a perceptual evaluation of the predictions made by the model to verify if they sound pleasant to listeners.

Acknowledgements

This work has been partly sponsored by the Spanish TIN project TIMUL (TIN 2013-48152-C2-2-R), the European Union Horizon 2020 research and innovation programme under grant agreement No. 688269 (TELMI project), and the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

References

1. Woody, R.H.: The effect of various instructional conditions on expressive music performance. J. Res. Music Educ. 54(1), 21–36 (2006), http://jrm.sagepub.com/cgi/doi/10.1177/002242940605400103
2. Covington, M.V.: The self-worth theory of achievement motivation: Findings and implications. Elem. Sch. J. 85, 4–20 (1984)
3. Juslin, P.N., Karlsson, J., Lindstrm, E., Friberg, A., Schoonderwaldt, E.: Play it again with feeling: computer feedback in musical communication of emotions. J. Exp. Psychol. Appl. 12, 79–95 (2006)
4. Kirke, A., Miranda, E.R.: A survey of computer systems for expressive music performance. ACM Computing Surveys 42, 1–41 (2009)
5. Tobudic, A., Widmer, G.: Relational IBL in music with a new structural similarity measure. In: Proceedings of the 13th International Conference on Inductive Logic Programming. pp. 365–382 (2003)
6. Ramirez, R., Hazan, A., Maestre, E., Serra, X.: A genetic rule-based model of expressive performance for jazz saxophone. Comput. Music J. 32, 38–50 (2008)
7. Giraldo, S.I., Ramirez, R.: A machine learning approach to discover rules for expressive performance actions in jazz guitar music. Front. Psychol. 7, 1965 (2016)
8. Grachten, M., Widmer, G.: Linear basis models for prediction and analysis of musical expression. J. New Music Res. 41, 311–322 (2012)
9. Marchini, M., Ramirez, R., Papiotis, P., Maestre, E.: The sense of ensemble: a machine learning approach to expressive performance modelling in string quartets. J. New Music Res. 43, 303–317 (2014)
10. Papiotis, P., Marchini, M., Perez-Carrillo, A., Maestre, E.: Measuring ensemble interdependence in a string quartet through analysis of multidimensional performance data. Front. Psychol. 5, 963 (2014)
11. Cambouropoulos, E.: The local boundary detection model (LBDM) and its application in the study of expressive timing. In: Proceedings of the International Computer Music Conference ICMC. pp. 17–22 (2001)
12. Stammen, D.R., Pennycook, B.: Real-time recognition of melodic fragments using the dynamic timewarp algorithm. In: Proceedings of the International Computer Music Conference ICMC. pp. 232–5 (1993)

Discovery of statistically interesting global-feature patterns

Kerstin Neubarth[1] and Darrell Conklin[2,3]

[1] Canterbury Christ Church University, United Kingdom
[2] University of the Basque Country UPV/EHU, San Sebastián, Spain
[3] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

Abstract. Descriptive pattern mining supports corpus-level music analysis by identifying interesting patterns in music repertoires. Analyses using global-feature representations usually employ a supervised mining approach, discovering patterns which distinguish between a corpus and an empirical background. Supervised techniques thus rely on availability of a background corpus. This paper presents a method for unsupervised discovery of global-feature patterns, without a background corpus, evaluating pattern candidates against a statistical background model. The method is illustrated in a case study on musical traits of Native American music.

Keywords: pattern discovery, itemset mining, computational music analysis, corpus analysis, Native American music

1 Introduction

In the computational analysis of symbolic music corpora, inter-song patterns capture musical characteristics which recur across several songs in a corpus [4]. Sequential patterns describe sequences of event-level features, while global-feature patterns represent sets of song-level features. Patterns are considered interesting if they occur in a corpus more frequently than expected. For sequential patterns, expected pattern frequencies have been calculated from empirical probabilities according to an explicit anti-corpus [4, 2] or from analytic probabilities according to a statistical background model [6, 3]. Discovery of global-feature patterns, on the other hand, has solely evaluated patterns using empirical probabilities, thus requiring a background or anti-corpus [11]. This paper presents a method for discovering interesting global-feature patterns based on analytic probabilities. The method is applied to a large collection of Native American songs.

2 Statistically interesting global-feature patterns

The pattern discovery method presented in this paper builds on techniques developed in the well-established field of itemset mining [9]. While traditional itemset mining is mainly concerned with discovering frequent itemsets, the method developed here evaluates patterns primarily by a statistical interest criterion.

A *global-feature pattern* is a set of features: $P = \{f_1, ..., f_k\}$. Each feature f_i represents an attribute–value pair, e.g. metreChange : yes. The feature set can be interpreted as logical conjunction: a song satisfies a pattern if all features in the set are true for the song. The number of songs satisfying the pattern gives the *support* of the pattern. To assess the interest I of a pattern P, the observed support of the pattern $O(P)$ is compared against its expected support $E(P)$:

$$I(P) = \frac{O(P)}{E(P)} \tag{1}$$

The expected support can be computed from an analytic background model as

$$E(P) = N \times \prod_{i=1}^{k} p(f_i) \tag{2}$$

where N is the number of songs in the corpus and $p(f_i) = O(f_i)/N$ is the probability of feature f_i. That is, $E(P)$ measures the pattern support which would be expected if the features in the set were independent of each other. A pattern is considered *interesting* if $I(P) > 1$: if the features of the set co-occur more frequently than expected under an assumption of independence [1]. To avoid spurious patterns, additionally a minimum support threshold is applied; a pattern whose support is above the minimum support threshold is called a *frequent* pattern. Finally, to reduce redundancy among discovered patterns, the current study focuses on *maximally general* interesting patterns, i.e. those interesting patterns which do not contain any subsets that are also interesting.

The search space of global-feature sets in a music corpus is exhaustively represented by a set-enumeration tree (e.g. [15]). The number of candidate patterns to be tested can be restricted by employing several pruning strategies. (1) *Semantic pruning*: features derived from the same attribute cannot co-occur in a song, and thus only sets of features derived from different attributes need to be evaluated [15]. (2) *Support-based pruning*: any superset of an infrequent global-feature set is also infrequent, and thus a branch of the search tree can be terminated when an infrequent pattern is reached. (3) *Interest-based pruning*: a pattern is a maximally general interesting pattern if none of its subsets are interesting, and thus a branch of the search tree can be pruned if at least one subset of the candidate is interesting [4].

3 Data and Results

Discovery of interesting global-feature patterns is applied to the Densmore collection of Native American music: songs collected, transcribed and published by Frances Densmore from the 1900s to the late 1950s [14]. Native American music has been studied extensively by ethnomusicologists; in his seminal survey of North-American native music, Bruno Nettl identified features shared by native music repertoires across the continent [10]. Densmore's and Nettl's writings provide a convenient reference to discuss global-feature patterns discovered by

Attribute	Description
tonality	tonality according to third above keynote
firstReKey	first tone relative to keynote
lastReKey	last tone relative to keynote
lastReCompass	last tone relative to compass of song
compass	number of tones comprising compass of song
material	tone material, scale
accidentals	chromatic alterations of tones
structure	relation between contiguous accented tones
firstDir	direction of first melodic progression
firstMetr	metrical position of first tone
initMetre	metre of first measure
metreChange	change of metre (measure-lengths)
rhythmUnit	rhythmic unit(s) in song

Table 1. Music content attributes, based on analyses by Frances Densmore.

	Number of songs	Serial Nos. of songs
Songs containing		
rhythmic unit	6	168, 172, 173, 174, 175, 180
no rhythmic unit	9	51, 52, 53, 169, 170, 176, 177, 178, 179
Total	15	

Table 2. Densmore's analysis of rhythmic units in a subset of songs [7, p. 308]. "For the purpose of this analysis a rhythmic unit is defined as 'a group of tones of various lengths, comprising more than one count of a measure, occurring at least twice in a song, and having an evident influence on the rhythm of the entire song.' " [7, p. 31]

computational data mining. In the current study, we consider 1770 Native American songs. Songs are represented by global features, which cover melodic aspects of songs, tonal material and rhythmic-metric aspects (Table 1). These features have been collated from Densmore's own analyses [12] (see example in Table 2). The resulting vocabulary consists of 52 features, derived from 13 attributes.

With a minimum support threshold of 70 songs (4% of the dataset) and an interest threshold of 3 (thresholds successfully applied in previous work, [5]), the discovery method returns 25 interesting patterns. Table 3 lists example patterns, ranked by pattern interest. To assess the statistical significance of patterns, a p-value was calculated according to the empirically approximated probability distribution of pattern candidates (fitted to ≈171,000 patterns generated at a minimum support of 1, minimum interest of 1.1 and including patterns which are not maximally general). The p-value indicates the probability of finding the number of observed patterns by chance alone and thus the risk of reporting false positives.

	pattern	$E(P)$	$O(P)$	$I(P)$	p-value
P1	{tonality : irregular, firstReKey : irregular}	4	79	22.00	2.7e-15
P2	{tonality : irregular, lastReKey : irregular}	4	79	20.21	1.3e-14
P3	{firstReKey : irregular, lastReKey : irregular}	4	79	19.60	2.3e-14
P4	{material : 4th_5toned_scale, tonality : major, firstReKey : triad_within_octave, lastReKey : fifth}	20	81	4.09	1.3e-5
P5	{firstReKey : keynote, firstDir : up, lastReKey : keynote, lastReCompass : notlowest}	19	78	4.02	1.9e-5
P6	{firstReKey : triad_above_octave, firstDir : down, lastReKey : keynote, lastReCompass : lowest}	42	136	3.28	5.9e-6
P7	{material : 2nd_5toned_scale, tonality : minor, lastReKey : keynote}	27	89	3.25	6.8e-5
P8	{material : 4th_5toned_scale, tonality : major, accidentals : no, lastReKey : fifth, rhythmUnit : yes, metreChange : yes}	23	71	3.12	2.7e-4

Table 3. Interesting global-feature patterns discovered in the Densmore collection of Native American songs. The listed patterns are statistically significant at $\alpha = 0.01$ ($p < 4.0e$-4 with Bonferroni correction).

The results support several observations presented by Densmore and Nettl. According to Densmore's analyses, "with few exceptions, the sequence of tones [in a melody] suggests a keynote" [8, p. 19], and the beginnings and endings of songs show a preference for the keynote or the third, fifth, octave, tenth or twelfth above the keynote [8, p. 21] (Table 3, patterns P4, P5, P6, P7 and P8). The final keynotes "are mostly the lowest tones of the individual songs" [10, p. 49] (pattern P6, but see pattern P5). Figure 1 shows a war song of the Chippewa satisfying pattern P6: the song starts on the twelfth above the keynote, i.e. on a triad tone above the octave, and moves downward until the final lowest tone on the keynote; the melodic motion forms a terrace-type contour [10] of two-bar phrases, each phrase starting on a tone lower or equal to the beginning of the previous phrase and descending through a fifth or occasionally a sixth to reach a plateau of repeated tones at the end of the phrase. In the songs surveyed by Nettl, the "majority (ca. 60 per cent) of the scales are pentatonic" [10, p. 49]; in the Densmore corpus "the largest percentage [is] major in tonality and based upon the upper partials of a fundamental" [8, p. 21] (pattern P4). Those songs "whose tones are not referable to a keynote are classified [...] as irregular in tonality" [8, p. 20]; in consequence the first and last tones cannot be related to a keynote and are also described as irregular (patterns P1, P2 and P3). Similarly, in some cases tonality and material features are semantically related: the fourth five-toned scale is also known as major pentatonic scale (patterns P4 and P8), and the second five-toned scale refers to the minor pentatonic scale (pattern P7). Rhythmically, "North American Indian music is organized heterometrically; isometric construction is very rare" [10, p. 50]: many songs thus contain changes of metre. Figure 2 presents an example illustrating pattern P8: the song contains

Fig. 1. War song of the Chippewa, illustrating pattern P6 of Table 3. Transcribed by Frances Densmore, Densmore catalogue number 346 [7, p. 72].

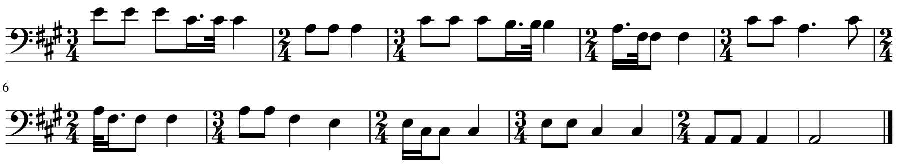
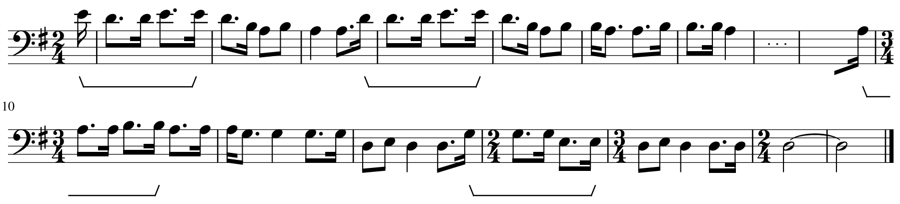


Fig. 2. Song of the Sioux drum presentation ceremony (excerpt), illustrating pattern P8 of Table 3. Transcribed by Frances Densmore, Densmore catalogue number S.20 [7, p. 178]. Brackets below the staffs indicate rhythmic units.



tones of the major pentatonic scale on G, ending on the fifth; the metre changes between duple and triple time; interestingly, the rhythmic unit of the song occurs in both duple- and triple-time measures [7, pp. 178-179].

4 Conclusions

One of the challenges in pattern discovery, especially in the context of exploratory corpus analysis, is the identification of potentially interesting patterns. Existing work on global features tends to adopt a supervised mining approach [13], extracting features which distinguish between different classes of songs. For describing characteristics of a corpus, these methods are thus restricted to situations where the corpus can be contrasted against an anti-corpus. Here we have presented a method for discovering interesting global-feature patterns when no anti-corpus is available, evaluating pattern candidates against an analytic statistical background. The method has been applied to the Densmore collection of Native American songs, revealing global-feature patterns which capture ethnomusicologically interesting traits of North-American Native music: the discovered global-feature sets reflect both inherently musical relations between features and statistically interesting feature combinations which, in the context of existing musicological observations, suggest style-specific musical characteristics of the analysed music corpus.

References

1. S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: generalizing association rules to correlations. *ACM SIGMOD Record*, 2:265–276, 1997.
2. T. Collins, A. Arzt, H. Frostel, and G. Widmer. Using geometric symbolic fingerprinting to discover distinctive patterns in polyphonic music corpora. In D. Meredith, editor, *Computational Music Analysis*, pages 445–474. Springer International, 2016.
3. T. Collins, R. Laney, A. Willis, and P. Garthwaite. Modeling pattern importance in Chopin's mazurkas. *Music Perception*, 4:387–414, 2011.
4. D. Conklin. Discovery of distinctive patterns in music. *Intelligent Data Analysis*, 14:547–554, 2010.
5. D. Conklin and C. Anagnostopoulou. Comparative pattern analysis of Cretan folk songs. *Journal of New Music Research*, 40(2):119–125, 2011.
6. D. Conklin and M. Bergeron. Feature set patterns. *Computer Music Journal*, 32(1):60–70, 2008.
7. F. Densmore. *Chippewa Music II.* Smithsonian Institution, Bureau of American Ethnology, Bulletin 53, Washington, DC, 1913.
8. F. Densmore. *Menominee Music.* Smithsonian Institution, Bureau of American Ethnology, Bulletin 102, Washington, DC, 1932.
9. P. Fournier-Viger, J.-W. Lin, B. Vo, T. Chi, J. Zhang, and H. Lee. A survey of itemset mining. *Wiley Interdisciplinary Reviews: Knowledge Discovery and Data Mining*, 7(4):e1207, 2017.
10. B. Nettl. North American Indian musical styles. *The Journal of American Folklore*, 67(263,265,266):44–56, 297–307, 351–368, 1954.
11. K. Neubarth and D. Conklin. Contrast pattern mining in folk music analysis. In D. Meredith, editor, *Computational Music Analysis*, pages 393–424. Springer International, 2016.
12. K. Neubarth, D. Shanahan, and D. Conklin. Supervised descriptive pattern discovery in Native American music. *Journal of New Music Research*, in press, http://dx.doi.org/10.1080/09298215.2017.1353637.
13. P. K. Novak, N. Lavrač, and G. I. Webb. Supervised descriptive rule induction. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning*, pages 938–941. Springer US, 2010.
14. D. Shanahan and E. Shanahan. The Densmore collection of Native American songs: a new corpus for studies of effects of geography and social function in music. In *Proceedings of the 13th International Conference for Music Perception and Cognition (ICMPC 2014)*, pages 206–209, Seoul, South Korea, 2014.
15. X. Zhang, G. Dong, and R. Kotagiri. Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. In *Proceedings of the 6th ACM SIGKDD International Conference for Knowledge Discovery and Data Mining (KDD 2000)*, pages 310–314, Boston, USA, 2000.

Computational modelling of expressive music performance in hexaphonic guitar

Marc Siquier, Sergio Giraldo, and Rafael Ramirez

Universitat Pompeu Fabra (UPF), Barcelona, Spain
marc.siquier01@estudiant.upf.edu
sergio.giraldo@upf.edu
rafael.ramirez@upf.edu

Abstract. Computational modelling of expressive music performance has been widely studied in the past. While previous work in this area has been mainly focused on classical piano music, there has been very little work on guitar music, and such work has focused on monophonic guitar playing. In this work, we present a machine learning approach to automatically generate expressive performances from non expressive music scores for polyphonic guitar. We treated guitar as an hexaphonic instrument, obtaining a polyphonic transcription of performed musical pieces. Features were extracted from the scores and performance actions were calculated from the deviations of the score and the performance. Machine learning techniques were used to train computational models to predict the aforementioned performance actions. Qualitative and quantitative evaluations of the models and the predicted pieces were performed.

Keywords: Machine learning, Computational models, Expressive music performance, Hexaphonic guitar

1 Introduction

Computational modelling of expressive music performance deals with the study of the deviations from the score that musicians introduce when performing a musical piece (aka. *Performance Actions (PAs)*). Previous studies have mainly focused on monophonic piano classical music. Some exceptions include guitar and saxofone jazz music, in which only monophonic performances have been considered.

In this work we present an approach to computationally model polyphonic guitar performances from hexaphonic guitar recordings. The present approach is an extension of previous work on monophonic expressive performance on jazz guitar [1]. Hexaphonic guitar recordings of musical pieces recorded by a professional guitarist, were obtained using a Roland GK-3 divided pickup. A new set of features was defined aiming to capture not only the melodic (monophonic/horizontal) context of the score, but also the harmonic (polyphonic/vertical) context, depicting the harmonic progression or simultaneity between notes. Score alignment using *Dynamic Time Warping* (DTW) was performed to extract PAs defined as *Onset Deviation* and *Energy Ratio* Later, machine learning models were trained in order to predict the aforementioned PAs. Quantitavie evaluation of the models was performed by means of accuracy measures over both *train* and *cross-validation* schemes, as well as qualitative evaluation was assessed from perceptual tests of the synthesized predicted pieces.

2 Background

In the past, music expression has been mostly studied in the context of classical music, and most research focus on studying timing deviations (onset nuances [2]) and dynamics (energy [3]). There are several expert-based systems, such as the *director musices* [4] by the KTH group, studying this field from different perspectives. On the other hand, Machine-learning-based systems try to automatically obtain the set of rules to predict the PAs. For an overview of theses methods see Goebl 2005 [5]. Kirke et al [6] model polyphonic piano showing that multiple polyphonic expressive actions can be found in human expressive performances.

Previous work on guitar expressive performance modelling has been done by Giraldo et al. [1] and [7], who model ornamentation and PAs in monophonic jazz guitar performances, using machine learning techniques. Bantula [8] models expressive performance for a jazz ensemble of guitar and piano extracting features for chords such as *density, weight* or *range.*

3 Methodology

In Figure 1 we present a block diagram of the whole system, where four main stages are depicted: data acquisition (guitar recording), audio to MIDI transcription, feature extraction and model computation, and finally MIDI synthesis. Expressive hexaphonic guitar recordings were obtained using the Roland GK-3 divided pick-up, which is able to separate the sound from each string [9].

The main output of this first stage was a new dataset consisting of hexaphonic recordings recorded by a guitar player with different performance intentions. This dataset consists of 3 audio recordings (one recording of *Darn that dream* a jazz standard by Jimmy Van Heusen and Eddie De.Lange and two recordings of *Suite en la* a classical piece by Manuel M. Ponce.) resulting in a total of 1414 recorded notes and their corresponding music scores saved as XML files.

Transcription of each individual string was computed using the YIN [10] algorithm and envelope-based note segmentation. The transcription of each string was added into a single MIDI file by having each string in a different channel. After doing performance to score alignment with the original score and the transcription of the expressive guitar performance using Dynamic Time Warping (DTW), feature extraction and PAs computation was performed. As the player was told to follow strictly the score, we can assume there are no structural differences between the music scores and the expressive music performance, so DTW can be applied directly.

Feature extraction was performed following an approach in which each note was characterized by its *nominal*, *neighbouring*, and *contextual* properties. *Nominal* descriptors refer to the intrinsic or intra-note properties of score notes. *Neighbouring* descriptors or inter-note descriptors refer to the relations of the note with its neighbouring or simultaneous notes. *Contextual* descriptors refer to the context of the song in which the note appears in (e.g. chords, key, mode, etc). A total amount of 34 features where extracted for each score note, plus two PAs representing onset and energy deviation from each score note to its matching performance note.

Several machine learning such as K-Nearest Neighbours, Decision Trees, Supervector Machines and Artificial Neural Networks were applied to model *Onset Deviation* (difference in time between performance onset of a note and its corresponding onset in the score) and *Energy Ratio* (ratio between performance note energy and its corresponding energy in the score).

$$Onset\_dev_i = Onset\_perf_j - Onset\_score_i$$

$$Energy\_rat_i = \frac{Velocity\_perf_j}{Velocity\_score_i}$$

Also, feature selection has been computed and analyzed in order to retrieve the subset of descriptors that better predict the studied PAs.

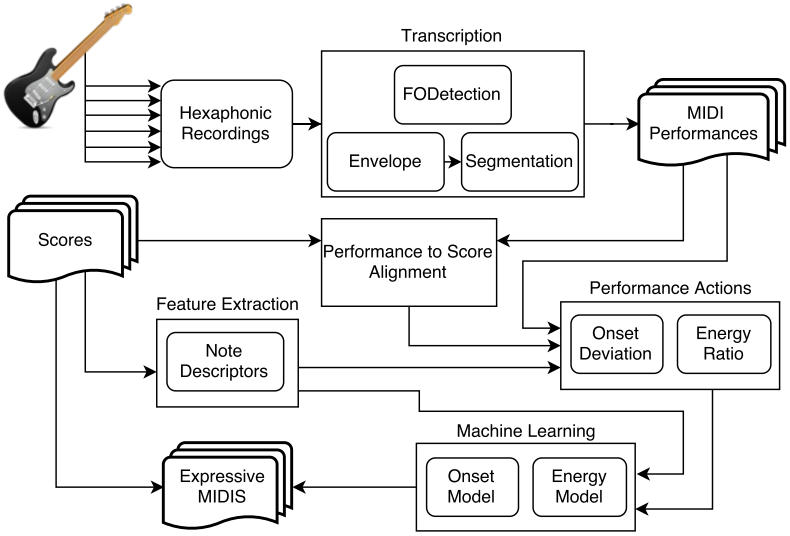


Fig. 1: Block diagram of the whole system.

4 Results

The proposed approach was quantitatively evaluated by measuring *Correlation Coefficient* (CC) obtained with the models studied, qualitatively evaluated by asking listeners to compare predicted and real performances. In Figure 2 we present the obtained CC for *Onset Deviation* and *Energy Ratio*. In red we show the accuracy for the whole training dataset and in blue the results applying 10-fold Cross-Validation (CV). The best accuracy (using CV) was obtained with the set containing the first 5 best ranked features. In Table 1 we show the results comparing different Machine Learning algorithms, both with CV and with the whole training set. We present the CC for *Energy Ratio* and *Onset Deviation* for the whole dataset, and using the best 5 features subset. The best results were achieved with Decision Trees where the obtained subset of 5 features outperforms the rest.

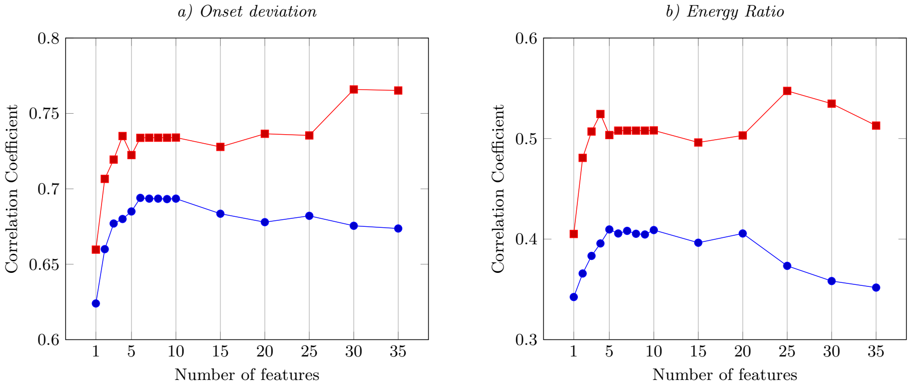


Fig. 2: Accuracies on increasing the number of features. Algorithm used: Decision Tree. Shown values correspond to Correlation Coefficients, in red for Train Dataset and in blue for 10 fold Cross-Validation.

Predicted feature	D.Tree	k_1NN	k_2NN	SVM	ANN
	cv/train	cv/train	cv/ train	cv/train	cv/train
Energy Ratio	0.35/0.51	0.22/1	0.26/0.78	0.21/0.33	0.23/0.63
Onset Deviation	0.67/0.77	0.30/1	0.36/0.81	0.39/0.45	0.29/0.67
Energy Ratio $_{5features}$	0.41/0.50	0.30/1	0.37/0.80	0.14/0.21	0.14/0.36
Onset Deviation $_{5features}$	0.69/0.72	0.38/1	0.61/0.82	0.30/0.31	0.44/0.43
Energy Ratio $_{bestsubset}$	0.41/0.51	0.30/1	0.37/0.79	0.16/0.21	0.15/0.39
Onset Deviation $_{bestsubset}$	0.69/0.73	0.37/1	0.58/0.82	0.30/0.32	0.48/0.48

Table 1: Results comparing different ML models (10 fold Cross-Validation). Shown values correspond to Correlation Coefficients.

For the qualitative survey, several synthesized pieces obtained by the models were compared to both the score (dead pan synthesis) and the performed (synthesized version) piece. Participants were asked to to guess how ”human” they sounded by comparing among them through an on-line survey [1]. Results from 15 participants (Figure 3) show that participants perceived the score more ”human” than the actual performance and predicted score. However, we obtained similar results among the performed piece and the predicted one, which might indicate that our models predictions are close to actual human performances.

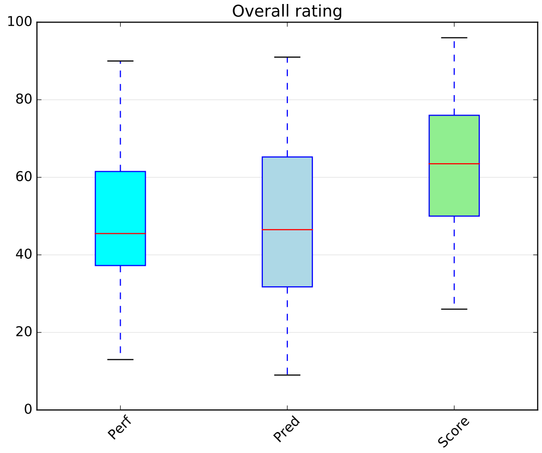


Fig. 3: Results of the on-line survey with performance, predicted and straight score synthesized midis.

5 Conclusions

In this work we have applied machine learning techniques in order to generate models for musical expression in polyphonic guitar music, by training different models for *Onset Deviation* and *Energy Ratio*. We treated polyphonic guitar as an hexaphonic instrument by capturing and transcribing each string separately. We extracted descriptors from the scores in terms of the melodic (Horizontal) as well from the harmonic (Vertical) context. We computed PAs from the aligned transcribed performance and the scores. We trained different models using machine learning techniques. Models were used to predict PAs that later were applied to the scores to be synthesized. Feature selection analysis and accuracy tests were performed to assess models performance. Perceptual tests were conducted on the predicted pieces to rate how close they sound to a human

[1] You can find the survey here: `https://marcsiq2.github.io/`

performance. Results indicate that descriptors contain sufficient information to generate our models able to predict performances close to human ones.

Acknowledgements

This work has been partly sponsored by the Spanish TIN project TIMUL (TIN2013-48152-C2-2-R), the European Union Horizon 2020 research and innovation programme under grant agreement No. 688269 (TELMI project), and the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

References

1. Giraldo, S., Ramírez, R.: A machine learning approach to ornamentation modeling and synthesis in jazz guitar. Journal of Mathematics and Music **10**(2) (may 2016) 107–126
2. Sundberg, J., Friberg, A., Bresin, R.: Attempts to reproduce a pianist's expressive timing with Director Musices performance rules. Journal of New Music Research **32**(3) (2003) 317–325
3. Bresin, R., Friberg, A.: Emotional Coloring of Computer-Controlled Music Performances. Computer Music Journal **24**(4) (2000) 44–63
4. Friberg, A., Bresin, R., Sundberg, J.: Overview of the KTH rule system for musical performance. Advances in Cognitive Psychology **2**(2) (2009) 145–161
5. Goebl, W., Dixon, S., Poli, G.D., Friberg, A., Bresin, R., Widmer, G.: ' Sense ' in Expressive Music Performance : Data Acquisition , Computational Studies , and Models. Artificial Intelligence (2005) 1–36
6. Kirke, Alexis, Miranda, E.R.: An Overview of Computer Systems for Expressive Music Performance. In: Guide to Computing for Expressive Music Performance. (2013) 1–47
7. Giraldo, S.I., Ramirez, R.: A machine learning approach to discover rules for expressive performance actions in jazz guitar music. Frontiers in Psychology **7** (2016) 1965
8. Bantula, H., Giraldo, S., Ramírez, R.: Jazz Ensemble Expressive Performance Modeling. Proc. 17th International Society for Music Information Retrieval Conference (2016) 674–680
9. Angulo, I., Giraldo, S., Ramirez, R.: Hexaphonic guitar transcription and visualization. In: TENOR 2016, International Conference on Technologies for Music Notation and Representation. (2016) 187 – 192
10. Cheveigne, A.D., Kawahara, H.: YIN, a fundamental frequency estimator for speech and music. **111**(April) (2002)

Multimodal Recognition for Music Document Transcription

Javier Sober-Mira, Jorge Calvo-Zaragoza, David Rizo, and José M. Iñesta

Department of Software and Computing Systems
University of Alicante, Alicante, Spain
{jsober,jcalvo,drizo,inesta}@dlsi.ua.es

Abstract. Converting sheet music scores into symbolic format is a necessary step to use computational tools for music indexing and analysis. We present an interactive framework in which user and computer collaborate to complete this transcription task. Our scenario assumes that the user traces the information found in the image using an electronic pen, and the system automatically recognizes the music symbols. The multimodal nature of the signal (both pen strokes and source image) can be used to improve the automatic recognition with Convolutional Neural Networks. Our experiments show that exploiting adequately this multimodality leads to a lower error rate.

1 Introduction

A large number of sheet music sources are available for musicological study. An interesting option is to use computational tools for large-scale indexing and analysis of the music. However, for this task to be feasible it is necessary to have that content transcribed into a machine-readable format.

An efficient way of digitizing sheet music is to resort to automatic transcription tools, usually referred to as Optical Music Recognition. These systems try to automatically extract the meaningful information contained in a music document from an image of its source. Nevertheless, these systems are still far from solving the problem accurately [4], which finally makes this option be discarded in most of the cases. Then, manual transcription is the only option left.

It is important to emphasize the role of the user as part of this process. In such case, the user is the most valuable resource and the system must be focused on minimizing the effort needed to complete the task [5]. It is, therefore, necessary to develop tools that allow an intuitive and efficient interface. In spite of several efforts to develop light and friendly software for music score edition, the process is still considered tedious by most users.

We focus on the human-machine interaction for tasks related to music document transcription. Conventional channels such as keyboard or mouse are not easily applicable here, and so there is a need to introduce new ways of interaction. Handwriting is a natural way of communication for humans, and so it would be interesting to use this kind of interaction for music document transcription. This can be done by means of electronic pen (e-pen) technologies.

The scenario stated produces multimodal signals that can be used to improve the recognition of the music symbols. In this work, we extend the first step presented in [1] by considering Convolutional Neural Networks for the multimodal classification. Within this paradigm, several ways of combining the different modalities produced are proposed, so that the performance can be improved as far as possible.

2 Multimodal data

This section describes the nature of the multimodal data considered in this work. The corpus of our case of study consists of 60 scores from a music archive dated between centuries 16th to 18th, handwritten in mensural notation [2].

We assume a framework in which the user traces symbols on the score using a digital surface, with the aim of automatically recognizing the music symbols. The system therefore receives a multimodal signal: on the one hand, the piece of the original image below the traced shape, referred to as *offline* modality (Fig. 1); on the other hand, the rendered image of the sequence of 2D points followed by the e-pen on the surface, referred to as *online* modality (Fig. 1). The challenge here is how to achieve an adequate synergy that eventually allows taking maximum advantage of all the modalities involved.

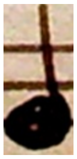


Fig. 1: Offline modality

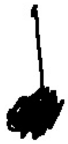


Fig. 2: Online modality

The considered dataset consists of 10 150 samples, each of which is represented by both *offline* and *online* modalities. Data was collected by five different users tracing symbols on the aforementioned archive.

The samples are spread over 30 classes. The number of symbols of each class is not balanced but it depicts the same distribution found in the documents.

2.1 Data overview

The distribution of symbols in the data set is shown in Fig. 3. In each cross-validation folder, a similar number of representatives from each class is found. For example, there are about 2600 samples of the *Minima* symbol, so in each cross-validation set there are approximately 500 samples of it, and the same applies to all classes. In any case, the maximum difference among the sizes of the cross-validation folders for any class was of 21 samples.

Some of the most important symbols considered are shown in Table 1.

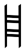
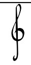
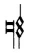
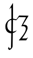
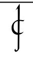
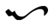
Group	Symbol			
Note	Semibrevis	Minima	Col. Minima	Semiminima
Rest	Longa	Brevis	Semibrevis	Semiminima
Clef	C Clef	G Clef	F Clef (I)	F Clef (II)
				
Signature	Major	Minor	Common	Cut
				
Others	Flat	Sharp	Dot	Custos
				

Table 1: A representative subset of the elementary symbols of the mensural notation archive considered in this work.

Also it is important to point out that those who helped to create the on-line mode data were writers with a basic knowledge of music, not specialists in early music notations. They were not instructed in how the specific mensural symbols should be interpreted.

The online sequences created by the different writers were shuffled in a single set and the cross-validation folders were taken randomly from it, in such a way that the different folds were not conditioned by a particular writing style.

3 Classification framework

We base our classification on Convolutional Neural Networks (CNN), given their great success in a range of tasks related to computer vision [3]. These networks take advantage of local filters, pooling, and many connected layers for learning a data representation to successfully solve classification tasks.

The topology of these networks can be very varied. We selected four architectures that are described below. We denote by $Conv(k, c)$ a spatial convolutional layer with kernel size $k \times k$ and number of filters c, with Rectifier Linear Unit activation. Similarly, we denote by $MaxPool(k)$ a max-pooling layer with kernel size $k \times k$. $Dropout(r)$ is a dropout procedure with a ratio of dropped units r. Then, our network architectures are defined sequentially as:

1. Conv(32,3) → Conv(32, 3) → MaxPool(2) → Conv(32, 3) → Conv(32, 3) → MaxPool(2)
2. Conv(32, 3) → Conv (32, 3) → MaxPool(2) → Conv(32, 3) → Conv(32, 3) → MaxPool(2) → Dropout(0.1)

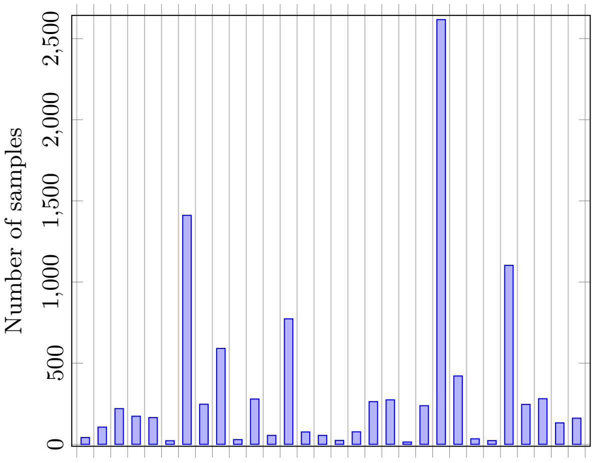


Fig. 3: Distribution of the 10 150 samples among the 30 different classes considered. The most repeated symbols are *Minima* (2617), *Col. Minima* (1411), *Semibrevis* (1103), *Col. Semiminima* (591), and *Dot* (773).

3. Conv(64, 3) → Conv (64, 3) → MaxPool(2) → Conv(32, 3) → Conv(32, 3) → MaxPool(2) → Conv(16, 3) → MaxPool(2) → Conv(16, 3) → MaxPool(2)
4. Conv(64, 3) → Conv (64, 3) → MaxPool(2) → Conv(32, 3) → Conv(32, 3) → MaxPool(2) → Conv(16, 3) → MaxPool(2) → Conv(16, 3) → MaxPool(2) → Dropout(0.1)

In all cases, a fully connected layer with 30 units and *SoftMax* activation is placed on top of the CNN in order to obtain a probability for each of the considered categories. Also, the input layers always expect images of an equal size of 36×36.

3.1 Multimodal classification

There might be several multimodal classification strategies depending on where the modality fusion is actually performed. Next lines describe each of the strategies considered.

Single mode It is interesting to consider how well the single modalities considered behave in order to assess the goodness of the fusion modalities. To this end, we consider the *single mode* classification strategy, which means that just a single modality is considered for the classification.

Late fusion *Late fusion* tries to merge the classification decisions obtained for each modality in order to obtain a more robust decision that takes into account both sources of information at the same time.

Due to the *SoftMax* layer, the output of the CNN corresponds to values between 0 and 1, indicating the confidence that the network gives to each possible category. Therefore, decisions of independent networks can be merged by a linear combination.

Let Ω denote the set of categories considered. Given images x and y from the *offline* and *online* modality, respectively, this fusion emits the label $\hat{\omega}$ such that

$$\hat{\omega} = \arg\max_{\omega \in \Omega} \alpha \, P_{\text{off}}(\omega|x) + (1-\alpha) \, P_{\text{on}}(\omega|y)$$

Where $P_{\text{off}}(\omega|x)$ and $P_{\text{on}}(\omega|y)$ are the probabilities given by the network used for the corresponding modality. Note that α is a parameter that tunes the weight given to each single modality. This parameter has to be fixed empirically.

Intermediate fusion Taking into account the internal operation of the CNN, an *intermediate fusion* can be considered. That is, the combination is performed in the intermediate layers of the network.

In this case, the intermediate union is achieved by the concatenation of the CNN used for each modality. Afterwards, a new fully connected layer with *SoftMax* activation is added to output a new probability value for each category. A graphical illustration is given in Fig. 4,

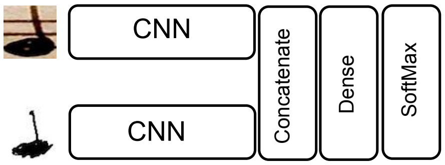


Fig. 4: Structure of the whole multimodal-model classification scheme.

4 Experimentation

Experimentation followed a 5-fold cross-validation scheme. The independent folds were randomly created with the sole constraint of having the same number of samples per class (when possible) in each of them.

Table 2 shows the error rate achieved by each combination of network model and classification scheme. Several $\alpha \in [0, 1]$ were tested for the late fusion strategy, and the best results were obtained for $\alpha = 0.5$.

Our experiments report that the information fusion (both late and intermediate) behave better than using single modalities, satisfying our initial hypothesis.

	Model 1	Model 2	Modal 3	Modal 4
Single mode (offline)	6.3 ± 0.7	6.1 ± 0.7	7.6 ± 0.2	6.6 ± 1.0
Single mode (online)	7.3 ± 0.2	7.3 ± 0.3	10 ± 2.1	7.7 ± 0.8
Late fusion ($\alpha = 0.5$)	3.6 ± 0.5	**3.2 ± 0.2**	4.3 ± 0.8	4.4 ± 0.4
Intermediate fusion	3.5 ± 0.7	3.5 ± 0.6	4.2 ± 0.8	3.6 ± 0.6

Table 2: Error rate (average ± std. deviation) obtained for a 5-fold cross validation experiments with respect to the classification scheme and CNN model.

The best results, on average, are reported by the late fusion with the network model 2. However, the difference with other models does not seem to be really significant.

5 Conclusion

This paper presents a new approach to transcribe sheet music into a computer by using e-pen technologies. This scenario produces a multimodal signal with which to improve the music symbol classification.

Results with this particular dataset was presented, considering CNN and several multimodal classification schemes. Results support that it is worth to consider both modalities in the classification process, as accuracy is noticeably improved with a combination of them than that achieved by each single modality.

This is a first step to achieve a complete system for music document transcription. More factors are still of interest, such as detecting the position of the symbols in the staff. For this task, an initial approach is using other CNN with images that have more top and bottom margins, so that it is able to discriminate by position. Those margins should be enough to see the whole staff, and thus be able to detect the vertical position in the same way a human could do.

References

1. Jorge Calvo-Zaragoza, David Rizo, and José Manuel Iñesta Quereda. Two (note) heads are better than one: Pen-based multimodal interaction with music scores. In *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, pages 509–514, 2016.
2. Antonio Ezquerro Esteban. Música de la catedral de barcelona a la biblioteca de catalunya. *Biblioteca de Catalunya, Barcelona*, 2001.
3. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
4. Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, André R. S. Marçal, Carlos Guedes, and Jaime S. Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190, 2012.
5. Alejandro H. Toselli, Verónica Romero, Moisés Pastor, and Enrique Vidal. Multimodal interactive transcription of text images. *Pattern Recogn.*, 43(5):1814–1825, May 2010.

A data-driven approach for Carnatic percussion music generation

Konstantinos Trochidis[1], Carlos Guedes[1] and Akshay Anantapadmanabhan[2]

[1] New York University Abu Dhabi, Abu Dhabi, UAE
[2] Independent Musician, Chennai, India
kt70@nyu.edu, carlos.guedes@nyu.edu, akshaylaya@gmail.com

Abstract. In this paper, we present a data-driven approach for automatically generating South Indian style rhythmic patterns. The method uses a set of annotated Carnatic percussion performances to generate new rhythmic patterns. All excerpts were manually annotated with beats, downbeats, and stroke registers. To model the rhythmic structure and the generation process of the talas, we use different partition templates that form the durations of the talas. We employed a modified version of the mutual nearest neighbor grouping algorithm to segment the rhythm sequences into meaningful grouping patterns that takes into consideration the proximity and the distance between each stroke inter-onset-interval (IOI) and their adjacent strokes. Finally, we use the *K*-means clustering approach to cluster the rhythmic groupings in terms of similarity.

Keywords: Carnatic music, music generation, rhythmic patterns.

1 Introduction

There is an increasing interest in developing computational strategies for the analysis and understanding of non-western music. Work by [1], [2], [3] and [4] in non-western music constitute some of the earlier examples in this area. Our work tries to develop generative models of Carnatic music percussion using a data-driven approach that departs from earlier work such as the one just mentioned. The goal of this paper is to develop expert systems that can reliably generate music in this style of Indian Classical music, envisioning a contribution on two levels: 1) the creation of tools for lay audiences to interact with musical styles beyond the Western ones; and 2) the automatic generation of unlimited amounts of data for training machine learning algorithms. By building applications that can recreate these musical styles we hope to create innovative tools for interaction with musical heritage that go beyond passively listening to the music. Generative music systems, video games, and virtual worlds are increasingly regarded as powerful tools for music education and performance [5]. We intend to continue developing musical applications that will allow their users to produce non-western rhythms through interaction with generative music algorithms. Using these generative systems to train machine learning algorithms would constitute a major contribution towards the creation of more robust computational systems for the analysis of the region's musical styles.

1.1 Rhythmic structure in Carnatic Music

The rhythmic framework of Carnatic music is based on the tala, which provides a structure for repetition, grouping and improvisation. The tala consists of a fixed time length cycle called avartana, which is also called the tala cycle. The avartana is divided into equidistant basic time units called aksaras, and the first aksara of each avartana is called the sama [6]. Two primary percussion accompaniments in Carnatic music are the Mridangam and Kanjira. All training excerpts used in the proposed generation method were performed on the Mridangam and Kanjira drum in the context of separate solo improvisations.

2 Approach

The approach we adopt in this study is to model the aditala cycle as a series of strokes forming a partition. Each partition is formed using different durations of groupings and sequences of strokes. In our study we used 6 templates of partitions of groups of pulses (fig.1), all adding to 32 pulses. The templates of partitions have been validated in terms of the grammar and theory of this music idiom by direct discussion with Carnatic music expert musicians. Given an audio recording, first we obtain an automatic transcription of a sequence of time-aligned events of all stroke types, their durations (IOIs) and velocities. All the recordings are merged into a text corpus of sequences of strokes. We used a grouping algorithm based on the mutual nearest neighbor that takes into consideration the proximity and the distance between each stroke IOI and their adjacent strokes to group them in meaningful rhythmic patterns. Next, all the patterns are indexed in terms of their duration and those patterns with durations between 2-8 secs are kept and used to form the partition variations of the tala. We transform all textual representation of the groupings into vector feature representations by using the bags of words approach and all grouping patterns are clustered based on similarity using the K-means algorithm.

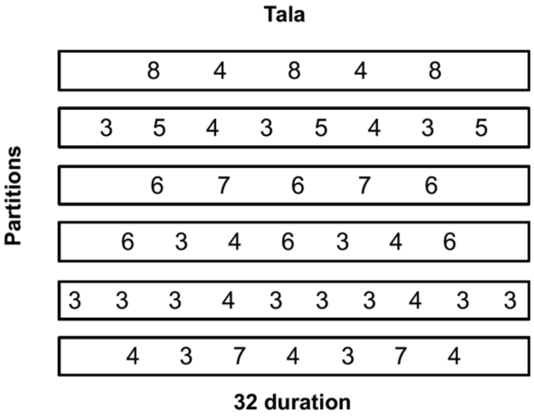


Fig. 1. Partition templates of aditala cycles.

2.1 Dataset

The training corpus consisted of 23 percussion solo compositions and groove patterns in aditala (8 beat-cycle) in three different tempo levels: slow (70bpm), moderate (85bpm) and fast (105bpm). The duration of each composition is around 2.5 minutes. The compositions were performed by professional Carnatic percussionist Akshay Anantapadmanabhan with the Mridangam and Kanjira drum. All excerpts were recorded using a metronome and were manually annotated including the sama and the other beats comprising the tala.

2.2 Encoding the strokes

In the Mridangam dataset each stroke event was encoded as a string based on five registers (Lo/Mid1-2-3/Hi), the hand (left (L) or right (R)) used for initiating the stroke, the inter-onset interval (IOI) between strokes and a value (V) indicating the velocity of the stroke. In the Kanjira drum data we used three register values (Lo/Mid/Hi), the inter-onset interval (IOI) between strokes and the velocity (V) of the stroke. For the Mridangam strokes, we also coded composite strokes played simultaneously with left and right hands. Although the Mridangam and Kanjira have a richer variety of registers and strokes, the reduction to three registers for the Kanjira and five for the Mridangam was a step to compromise the different stroke definition. This reduction was validated by Anantapadmanabhan as a process to accurately encode the different strokes in both percussion instruments. The normalized velocity values of the strokes were obtained by computing an onset detection function by combining energy and phase information in the complex frequency domain, and estimating its amplitude level with a value between 0.2 and 1 according to the strength of the stroke. For example, LoLV2T4, indicates a stroke in the Low register (Lo) using the left hand (L) on the mridangam with velocity 0.5 (V2) and duration of a dotted quarter note (T4).

3 Grouping of strokes

The encoded strokes are parsed using a grouping algorithm that groups them into meaningful rhythmic patterns according to the 6 partition templates shown in figure 2. We used a modified version of the mutual nearest neighbor algorithm [7]. It works based on the proximity of the strokes by measuring the distance between adjacent strokes. Strokes are grouped together if they are nearest neighbors to each other.

A constraint of the algorithm is that every grouping has a minimum number of 2 strokes. We adopted this constraint to avoid very small groupings of individual strokes. The algorithm stops parsing and form new groupings using a threshold that represents the largest duration that a grouping pattern can take. Figure 2 illustrates an example of grouping. The algorithm starts grouping the first two strokes in the rhythmic sequence based on the initial constraint and then compares the distances D1 and D2 between the last stroke in the grouping (blue dot) with the the next two strokes (green and red dots). D2 is less than D1 so the algorithm finds a boundary, saves the current grouping, and creates a new one. The algorithm is iterative and works hierarchically, e.g. when it finishes parsing the strokes and comparing the distances we get two layers of groupings four groupings for the first and two groupings for the second layer.

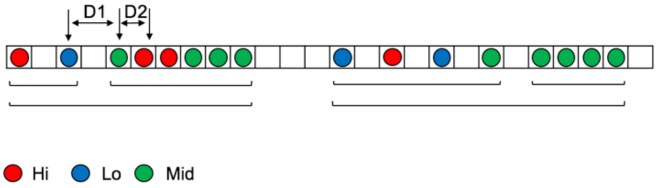


Fig. 2. Grouping of strokes based on the mutual nearest algorithm where brackets indicate groupings of strokes

4 Clustering analysis

4.1 Feature representation

All the groupings of strokes were represented initially as symbolic notation, i.e. a string of text. To transform the textual symbolic information into meaningful feature vectors we used the bag of words approach and extracted all bigrams of the groupings. We generated feature vectors of the groupings by counting the times each bigram two stroke events occur in a grouping. This leaded to a feature matrix, which could be further used for clustering analysis. Finally, the K-means clustering approach was used to cluster the groupings of strokes in terms of similarity.

4.2 Visualizing data using t-SNE

The next step was to convert the high-dimensional data set representing the center of the clusters from the clustering analysis into a matrix of pairwise similarities to enable the visualization of the resulting data. Traditional dimensionality reduction techniques such as Principal Components Analysis are linear techniques that focus on keeping the low-dimensional representations of dissimilar data points far apart.

In our analysis, we used the so called t-distributed stochastic neighbor embedding (t-SNE), for visualizing the resulting similarity data [8]. Compared to methods discussed previously, t-SNE is capable of capturing much of the local structure of the high-dimensional data, while also revealing global structure such as the presence of clusters at several scales. Figure 3 illustrates the result of the t-SNE transformation on the clusters of the groupings. A 2-dimensional axis compromising two components were used for the t-SNE analysis and the pairwise distances between the cluster centers of the groupings are plotted in the axis.

5 Generation process

To synthesize and generate the talas, we modelled the 8-beat aditala cycle into a series of partitions of 32 timepoints/cycle, assuming a beat subdivision in 4 parts, and used the partition templates shown in Figure 1. By analyzing and clustering the different groupings of strokes based on their duration and similarity we were able to have an index of different grouping durations for different clusters. In order to fill the duration

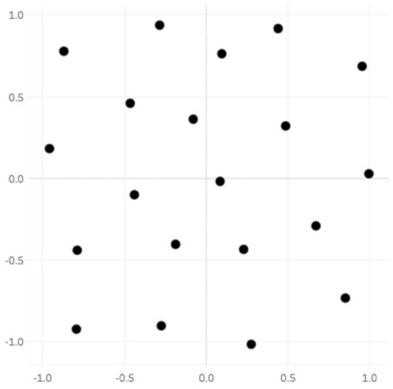


Fig. 3. 2D map of proximity distances between the pattern clusters of groupings

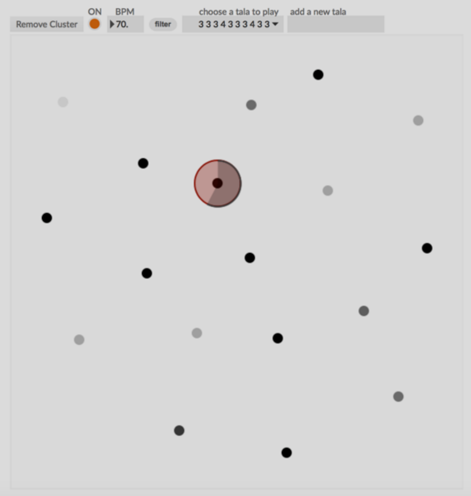
of the tala using a template of partition we used similar groupings in a cluster of different durations given by the partition templates. By clustering the patterns based on similarity we could generate different variations of a specific partition template of a tala using different groupings of the same duration.

5.1 Carnatic music generation application

The results from the analysis were used to develop a generative model that creates rhythmic grooves based on the groupings of strokes. The model was implemented as a Max patch that used as inputs the partition templates, the clusters of the groupings, the durations of the groupings and the coordinates of the cluster centers after the t-SNE data visualization analysis. This tool not only synthesizes the results from the analyses but it can be also used as a computational application for creative and learning exploration of these rhythms. This latter aspect is of particular interest as it provides the gateway to develop software applications for automatic rhythm generation in non-western music styles. Figure 4 depicts a screenshot of the Max patch. The user can interact with the clusters of groupings by travelling in the 2D space and generate talas of preference based on a set of template partitions in various tempo of choice. He can also filter smaller rhythmic values, or create variations by having the program probabilistically choose between different stroke collections of the same duration in the cluster.

6 Discussion and Future work

This work presents a method for automatically generating new Carnatic style rhythmic patterns based on a set of training examples. The approach we adopt in this study is to model the aditala cycle using a series of partition templates. Each partition is formed using different durations of groupings and sequences of strokes. To improve the current methodology of rhythmic grouping we aim to adopt a new approach based on a dictionary method of pre-recorded Carnatic phrases.



BPM 70.

choose a tala to play 3 3 3 4 3 3 3 4 3 3

Fig. 4. Max patch of Carnatic music generation application

This method will use a dataset of well-formed Carnatic grouping dictionary of phrases performed with different variations and durations. These phrases will be later used as groupings to form the duration of the partition templates and generate the talas. Future work will also test the method on a larger dataset of recordings and evaluate the effectiveness of the method by conducting a perceptual study using a group of professional Carnatic musicians in Chennai.

References

1. Serra, X.: A Multicultural Approach in Music Information Research. In Proceedings of the 5th International Society for Music Information Retrieval Conference (ISMIR), Miami, pp. 151–156, 2011
2. Srinivasamurthy, A., Holzapfel, A. & Serra, X.: In search of automatic rhythm analysis methods for turkish and indian art music. Journal of New Music Research, 43:94–114, 2013
3. Bozkurt, B., Ayangil, R., & Holzapfel, A.: Computational analysis of Turkish makam music: Review of state of-the-art and challenges. Journal of New Music Research, 43(1), 3–23, 2014
4. Herremans, D., Weisser, S., Sörensen, K., Conklin, D.: Generating structured music for bagana using quality metrics based on Markov models. Expert Systems With Applications. 42(21):424–7435, 2015
5. Tobias, E.: Let's play! Learning music through video games and virtual worlds. In G. McPherson (Ed.), Oxford handbook of music education, Volume 2. (pp. 531-548). New York: Oxford UP, 2012
6. Sambamoorthy, P.: South Indian Music Vol. I-VI, The Indian Music Publishing House, 1998
7. Tousaint, G.: Measuring the Perceptual Similarity of Middle-Eastern Rhythms: A Cross-Cultural Empirical Study. In: Proceedings of the Fourth International Conference on Analytical Approaches to World Music. New York, 2016
8. Van der Maaten, L.J.P., Hinton, G.E.: Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research, 9(Nov):2579-2605, 2008

Informing bowing and violin learning using movement analysis and machine learning

Erica Volta, Paolo Alborno and Gualtiero Volpe

Casa Paganini – InfoMus, DIBRIS, University of Genova, Genova (Italy)
erica.volta@edu.unige.it, paoloalborno@gmail.com, gualtiero.volpe@unige.it

Abstract. Violin performance is characterized by an intimate connection between the player and her instrument that allows her a continuous control of sound through a sophisticated bowing technique. A great importance in violin pedagogy is, then, given to techniques of the right hand, responsible of most of the sound produced. This study analyses the bowing trajectory in three different classical violin exercises from audio and motion capture recordings to classify, using machine learning techniques, the different kinds of bowing techniques used. Our results show that a clustering algorithm is able to appropriately group together the different shapes produced by the bow trajectories.

Keywords: Movement Analysis, Music Performance, Music Learning, Machine Learning, K-Means, Multimodal Interactive Systems

1 Introduction

1.1 Background

One of the central elements in violin pedagogy is the bowing technique. Methods introducing violin pedagogy commonly begin illustrating how to hold properly violin and bow, as well as the proper posture to support body movements without impeding bow arm (Galamian, 1962). A great importance in the violin pedagogy is given to techniques of the right hand that is responsible of most of the sound produced.

Several studies investigated the movement of the bow, finding a strict relationship between motion characteristics and quality of the performance, e.g., the bowing motion should be fluid (Galamian, 1962) and circular (Starker, 1979).

One of the first and most influential technical approaches to the study of bow movement is Hodgson's *Motion Study and Violin Bowing,* published in 1934. In his famous work, Hodgson used early methods of photographic motion tracking to study the circular nature of bowing technique in cyclographs (see Figure 1).

The controversial insights of Hodgson's work, showing that the bow's trajectory is always curved, has caused an animated pedagogical debate, but the knowledge of the curved nature of bowing has influenced the pedagogy of the last century, giving to violinists an explanation and a metaphor to understand the correctness of their movements, since it is generally not common for students or teachers to see their own playing movements represented in a visual way.

The body of scientific research related to music education has grown significantly in recent decades, e.g., see (Rauscher, Shaw, & al, 1997), as well as the trend in developing sensors-based systems to use bow gestures in interactive performance (Machover, 1992), (Nichols, 2002), (Overholt, 2005). Despite this growth, since Hodgson's work, technology has been rarely applied to music pedagogy and usually restricted to other domains, such as to audio and video recording and playing.

This study is carried out in the framework of the EU-H2020-ICT Project TELMI, having the purpose of enhancing music learning through the development of multimodal systems for real-time and off-line feedback to students.

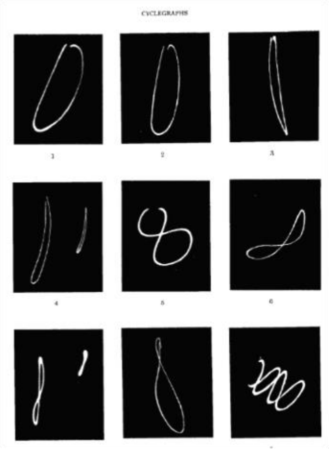


Figure 1: Hodgson's cyclographs

The aim of the present preliminary work is to explore whether multimodal systems and machine learning techniques can be used for analyzing bow trajectories as a means of contributing to music performance pedagogy, by working on selected recordings of renowned performers and teachers recruited by the Royal College of Music in London.

1.2 Bowing techniques

Bow control is a central musician's skill, giving the violinist the ability to direct bow's motion during playing.

In his work, Hodgson divided bowing movements in three categories of motion:

- Movements across the string, influenced by tilt, speed and contact point techniques;
- Rotation movement around the string, that allows changing across strings and changing in direction;
- Movement towards and away from string, variating the weight and that is responsible of particular articulation effects.

For this work, we chose to focus our attention on articulation techniques, as one of the most delicate parts of violin education and as one of the elements that a multimodal system can help to analyze.

From the entire TELMI archive (see Section 1.3) we selected the *Martelé* (from Kreutzer, Op.7), *Spiccato*, and *Sautillé* (from Ševčík, Op.3) exercises recorded by four internationally renowned and esteemed professional violinists involved in the TELMI project. The choice of these exercises was made by considering the importance of these three different bowing techniques and the differences between them that are often confused and difficulty master by students.

One of the difficulties of these studies is related to speed, because there is a common ground where one should be able to make the change from *Spiccato* to *Sautillé* and vice versa without changing any character of the sound profile. Furthermore, the mechanic of these two bowings are completely different. In *Spiccato* every single note is played actively, whereas in *Sautillé* the jumping activity is left quite exclusively to the resiliency of the stick. A further difference lies in the hand's motion: according to Hodgson, in fact, during *Spiccato* the bow designs an eight in the air and when *Spiccato* is quickened to *Sautillé*, the movement of the hand changes to an ellipse, but the bow continues to draw an eight in the air. *Martelé* represents, finally, a third type of fundamental movement of bowing, since it is at the basis of essential bowing techniques, such as *Staccato*, where the pressure is released between each stroke, and the bow speed has to be quite fast, yet light.

1.3 The TELMI archive of multimodal recordings

One of the milestones of TELMI is to build a corpus of multimodal data for informing the development of a multimodal interactive system for technology-enhanced violin learning and teaching (Volpe, Kolykhalova, Volta, & al., 2017).

The archive is organized in a structured collection of exercises that follow the learning path of classical violin students.

It includes several sources of data, such as motion capture of the performer, of the violin and of the bow, ambient and instrument audio, video, physiological data, (electromyography) and Kinect data.

The corpus of material consists of 41 exercises, concerning handling the instrument, techniques of the right and left hands, articulation, and some expressive works (such as Elgar, *Salut d'amour, Op. 12*).

All the recorded data were synchronized and played back using the EyesWeb XMI[1] platform (Camurri, Hashimoto, Ricchetti, & al., 2000), (Volpe, Alborno, & al., 2016).

2 Recordings and segmentation

We recorded 4 players performing the three selected exercises (*Martelé*, *Spiccato* and *Sautillé*) during the recording sessions for the TELMI archive. The violinists received the entire list of exercises in advance and the use of music sheets was allowed.

A Qualysis motion capture system endowed with 13 video cameras was used to record each performance. MoCap data was recorded at 100 Hz and synchronized with two video streams at 50 fps and with a pickup microphone stuck on the violin. The bow was endowed with 4 lightweight reflective markers. 2 more markers defined as virtual markers, together with 6 rigid bodies were included to enhance tracking robustness and reliability. On the violin, we attached 6 further markers.

After the recordings, data was segmented, by considering the musical structure, to isolate single bowing movements for each music phrase. We extracted 869 segments in total from 12 recordings we considered.

Using EyesWeb we computed the 3D-trajectories of the tip of the bow, to check the presence of Hodgson's bowing shapes.

[1] www.casapaganini.org/eyesweb

We then computed six different features on each obtained segment, in particular:

- *Acceleration*
- *Trajectory length*
- *Kinetic energy*: an approximation of the overall energy spent while performing a movement with the bow. It is computed as the total amount of displacement in all of the tracked points.
- *Curvature*: The derivative of the position vector over the curve of the trajectory provides a *tangent vector* to the curve. The curvature is the derivative of such a vector, and describes how the tangent vector changes. As an example, a trajectory following the contour of a geometric shape, such as a square, will bend sharply in some points, so its curvature will have high values.
- *Directness*: this is a measure of the extent to which a given trajectory is direct or flexible. It is computed as the ratio between the Euclidean distance calculated between the starting and the ending point of the considered trajectory, and its length.
- *Smoothness*: this corresponds to the third derivative of the position and it has often been used as a descriptor to evaluate how a motion trajectory varies "slowly" over time (Flash and Hogan, 1985).

3 Clustering

To obtain the same number of features for each segment, we extracted a cumulative histogram with 25 bins for each considered feature, resulting into a 150-dimension feature vector data.

K-means was applied to all the feature vectors, to figure out whether the different kinds of articulation exercises we were studying can be distinguished from the characteristics of the bow motion.

We then estimated the best number of clusters (i.e., the k parameter to seed the k-means algorithm), that is shown in the Elbow curve in Figure 2.

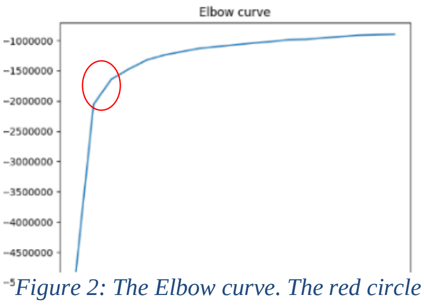


Figure 2: The Elbow curve. The red circle identifies the number of clusters (k =3) we choose for this study.

A value of k=3 was detected as the most appropriate value. Finally, we applied a PCA reduction to lower the dimension to the most representative 2 and 3 features, and visualized the clusters. Resulting data clusters are shown in Figure 3 with different colors. As the figure shows, the considered segments were split mainly in three clusters. We verified a clear separation between three particular bow motion trajectories, i.e., segments belonging to the same cluster present similar trajectories.

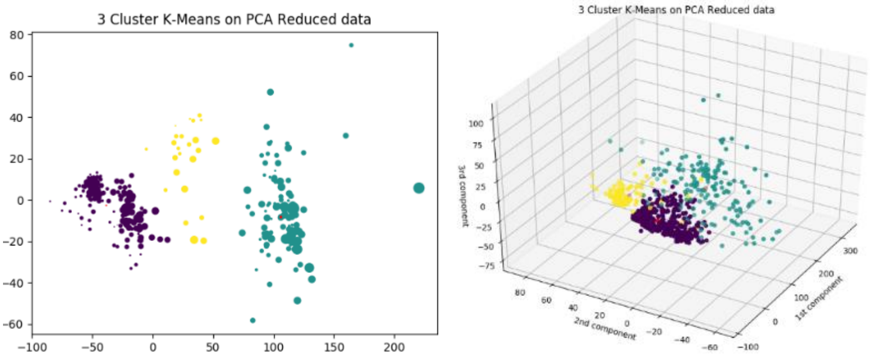


Figure 3: 3 Clusters obtained by applying K-Means on PCA reduced data

We observed that most of the segments composing the purple cluster have been extracted from *Sautillé* performances, and segments belonging to the yellow cluster are mostly related to the *Spiccato* pieces. The most spread cluster, the light green one, is mostly composed by segments belonging to *Martelé* recordings.
Our hypothesis, that needs to be investigated in the future with an extension of the presented work, is that the set of features we computed on the trajectories can effectively be used to distinguish different bowing exercises and articulation techniques.

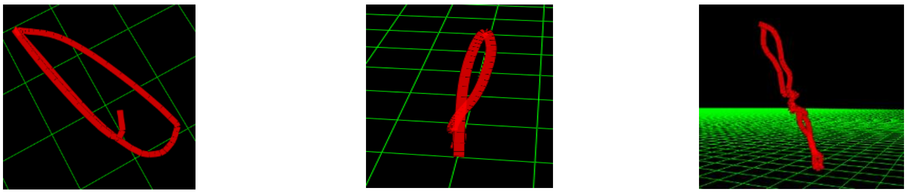


Figure 4: 3D visualization of the trajectories "drawn" by the bow while performing three different exercises. From left to right: a) Martelè: characterized by a circular trajectory, b) Sautillè: characterized by an "8"- trajectory, c) Spiccato: characterized by a lace trajectory.

4 Conclusions

In this paper, we presented some preliminary results of our analysis of the TELMI multimodal corpus of data. Results need to be confirmed by a deeper analysis of the clustering we obtained. We aim at developing more sophisticated techniques to realize an adaptive system, able to understand the type of bow movement and violin exercise starting from movement features.

The use of machine learning techniques in a music educational project aims to further develop algorithms able to automatically assess the quality and precision of the music performance to help students to enhance their musical skills.

5 Acknowledgments

We thank violinists Madeleine Mitchell, John H. Gilbert, Eulalie Charland, and Berent Korfker for participating in the recordings. We thank luthier Alberto Giordano for providing and preparing the violin and the bow.

This work is partially supported by the EU-H2020-ICT Project TELMI.

TELMI has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 688269.

References

Brandfonbrener, A. (2004). Healthier music students: Can medicine and music prescrobe in concert? *Medical Problems of Performing Artists*, 19(1). 1-2.

Camurri, A., Hashimoto, S., Ricchetti, M., et al., (2000). EyesWeb: Toward Gesture and Affect Recognition in Interactive Dance and Music Systems. *Computer Music Journal*, 57-69.

Galamian, I. (1962). *Principles of Violin Playing and Teaching.* London: Faber & Faber.

Kolykhalova, K., Volta, E., Ghisio, S., et al., (2017). A multimodal corpus for technology-enhanced learning of violin playing. *Proceedings of the 12th biannual Conference of the Italian ACM SIGCHI Chapter.* CHItaly.

Machover, T. (1992). *Hyperinstruments – a progress report 1987-1991.* Technical report, MIT Media Laboratory.

Nichols, C. (2002). The vBow: A virtual violin bow controller for mapping gesture to synthesis with haptic feedback. *Organized Sound*, 7(2):215-220.

Overholt, D. (2005). The overtone violin. *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05)*, (pp. 34-37). Vancouver, BC, Canada.

Rauscher, F. H., Shaw, G. L., et al, (1997). Music causes long-term enhancement of preschool children's spatial-temporal reasoning. *Neurological Reserch*, 19, 2-8.

Starker, J. (1979). An organized method of string playing. In M. A. Grodner, *Concepts in String Playing: Reflections by Artisti-Teachers at the Indiana University School of Music* (pp. 133-55). Bloomington: Indiana University Press.

Volpe, G., Alborno, P., et al., (2016). Designing Multimodal Interactive Systems using EyesWeb XMI. *Proceedings of the First International Workshop on Smart Ecosystems AVI 2016*, (pp. 49-56). Bari.