International Journal of Semantic Computing

Assessment of Percussive Music Performances with Feature Learning

Chih-Wei Wu
Center for Music Technology
*Georgia Institute of Technology**
cwu307@gatech.edu
http://www.musicinformatics.gatech.edu

Alexander Lerch
Center for Music Technology
Georgia Institute of Technology
alexander.lerch@gatech.edu

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

The automatic assessment of (student) music performance involves the characterization of the audio recordings and the modeling of human judgments. To build a computational model that provides a reliable assessment, the system must take into account various aspects of a performance including technical correctness and aesthetic standards. While some progress has been made in recent years, the search for an effective feature representation remains open-ended. In this study, we explore the possibility of using learned features from sparse coding. Specifically, we investigate three sets of features, namely a baseline set, a set of designed features, and a feature set learned with sparse coding. In addition, we compare the impact of two different input representations on the effectiveness of the learned features. The evaluation is performed on a dataset of annotated recordings of students playing snare exercises. The results imply the general viability of feature learning in the context of automatic assessment of music performances.

Keywords: Music Performance Assessment; Music Information Retrieval; Feature Learning.

1. Introduction

Music performance is a sequence of actions that integrates both cognitive and motor skills. Starting from the musical ideas, this process of converting thoughts into movements is, as pointed out by Palmer [1], among the most skill-intensive actions produced by human beings. To cultivate these skills, the qualitative assessment by peers and teachers is an essential pedagogical component in music education. A systematic assessment that is able to facilitate improvements usually requires careful

*Center for Music Technology, Georgia Institute of Technology, 840 McMillan St, Atlanta, GA 30332

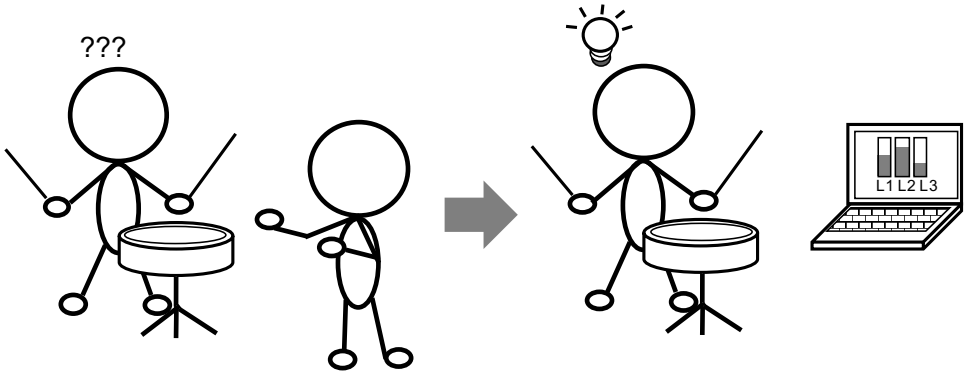

Fig. 1. The concept of automatic music performance assessment systems.

examination of different aspects of a performance, such as technical correctness and alignment with aesthetic ideals. This task, however, is extremely difficult due to its subjective nature. As a result, human raters tend to exhibit a large variance in terms of their severity, rating scale, and the interpretation of rating categories [2]; the bias of the raters and the ill-defined categories, as suggested by Thompson and Williamon [3], could adversely impact both the discriminability and the fairness of assessment. A computational approach that provides consistent and repeatable feedback, as illustrated in Fig. 1, might offer a potential solution to this issue and enhance the students' learning experience.

With recent advances in the field of Music Information Retrieval (MIR) for tasks such as music transcription [4] and source separation [5], the realization of intelligent music systems with reliable functionality became plausible, opening up new possibilities for music education [6]. Commercial software such as SmartMusic[a] and Yousician[b] both showcase how automatic assessment tools could enhance the music learning process with more flexible practice sessions and individualized feedback. These efforts, while providing rudimentary solutions to the users, still fall short in characterizing non-technical aspects of a performance. We hypothesize that this limitation originates from the design of the audio features. In MIR, a set of well-established features has proven to be surprisingly successful in applications such as music genre classification [7], music emotion recognition [8], and drum sound classification [9], however, using the same features for other tasks such as music performance assessment has been shown lead to sub-optimal performance [10, 11].

In this paper, we explore the possibility of applying unsupervised feature learning in the context of music performance assessment. In particular, we compare the effectiveness of a set of baseline features, designed features, and learned features in assessing students' snare drum performances from a large set of recordings of the

[a]http://www.smartmusic.com Last access: 2017/10/15
[b]http://yousician.com Last access: 2017/10/15

Florida all-state auditions. A successful set of features would clearly describe the acoustic events rendered by the performers, providing a good foundation for studying the connection between music performance and its corresponding assessment given by the human judges. Therefore, the goal of this paper is to identify optimal features for characterizing percussive instrument performances.

The contributions of this paper include:

(1) new insights into the viability of applying feature learning to the problem of automatic music performance assessment,
(2) a new input representation that characterizes percussive instrument performances for feature learning purposes, and
(3) the demonstration of potential improvements in predicting judges' ratings using the proposed method.

The remainder of this paper is structured as follows: Section 2 presents the related work in automatic music performance assessment and feature learning. In Sect. 3, we introduce our proposed method. The experiment setup and results are described in Sect. 4. Finally, the conclusion and future directions are presented in Sect. 5.

2. Related Work

Music, as a performing art, requires the rendition of a musical score or idea into a physical acoustical realization [12]. The score can be understood as a blueprint for a performance [13] that contains implicit and ambiguous information that can be difficult to describe and quantify [14, 15, 1, 16]. This ambiguity leads to often significant differences between performances of the same piece. These differences can fall into the categories pitch (vibrato, intonation, etc.), rhythm (tempo, micro-timing, etc.), dynamics (accents, loudness, etc.), and sound quality (playing technique, articulation, etc.) [17].

Music Performance Analysis (MPA) is a research field that focuses on the study of the performance parameters in the acoustic rendition rather than the musical score itself [12]. Instead of analyzing the intentions of the composer from the symbolic representation of a music composition, MPA focuses on interpreting the artistic decisions made by the performers during their music performance. For example, by playing the same score at different tempi or with different dynamics, the performer makes choices that result in performances which convey different information with various levels of expressiveness. One of the main challenges of MPA, therefore, is to associate these expressions with human perception in a musically meaningful way.

To automate the process of MPA, a system must handle the extraction and interpretation of the important parameters of music performances. In the early research, most of the analysis was performed on symbolic data extracted from external sensors or MIDI devices. For example, electronic keyboards have been utilized to investigate timing [18], or pianos with sensors have been used to study

dynamic variations [19] or asynchronies between pianists' hands [20]. More recently, the focus has gradually shifted to the analysis of audio recordings, often focusing on intonation (compare, e.g., [21]) or tempo [22].

As early as the 1930s, Seashore proposed to use objective measurements of performance parameters to support music education [23]. The commercial applications mentioned in the introduction testify to the success of this idea and its commercial viability. The basic difficulty here that the purely descriptive approach to investigating music performance of most of the studies presented above, does not directly result in an overall assessment of quality, which would be helpful at least for certain applications in music education. Instead, they limit themselves to describe parameter variations and discuss differences and commonalities of multiple performances and often ignore perceptive and cognitive aspects of the reception of a performance.

Nevertheless, the assessment of music performance also has been a topic to study. The basic approach to this problem usually involves the careful design of audio features that are capable of extracting the most relevant information from performance data in different contexts. For instance, Nakano presented an automatic system that evaluates the singing skill of the users [24]; by characterizing the performances through pitch interval accuracy and vibrato features, the system was able to classify the performance into the two classes *good* and *poor*. Similarly, Knight et al. attempted to classify the tonal quality of trumpet performances into the categories "good" and "bad" with low-level audio features. Abeßer et al. propose a system that automatically assesses the quality of vocal and instrumental performances of 9th and 10th graders [25]. Features representing the pitch, intonation, and rhythmic correctness are designed to quantify the students' performances. They report that the system is able to predict four different performance qualities with occasional confusions between the adjacent classes. In a system that identifies common mistakes by the flute beginners, Han and Lee propose to use features such as MFCCs with sparse filtering for detecting events such as poor blowing and misfingering [26]. Bozkurt et al. used features extracted from the fundamental frequency contour to classify vocal performances into the categories "pass" and "fail" [27]. More recently, both Wu et al. and Vidwans et al. assess students' instrumental performances using a set of features derived from pitch, amplitude, and inter-onset interval histograms[10, 11]. The evaluation results show some correlation between the model predictions and expert judges' ratings.

All of the above mentioned systems use custom-designed features in the analysis pipeline. This approach, while translating existing music domain knowledge into machine operations, might also strip away important information that resides in the data as it focuses on specific aspects that the designers consider to be important. Feature learning, on the other hand, allows an algorithm to find the most suitable features based on given input representations. Several feature learning methods have been found successful in music related applications, especially for the task of music

genre recognition. For example, Lee et al. applied convolutional Deep Belief Networks (DBNs) to magnitude spectrograms to learn features [28]; the evaluation results show that the learned features outperform conventional audio features in recognizing music genre. Similarly, Hamel and Eck demonstrate that a DBN is able to derive features that achieve state-of-the-art performance in music genre recognition [29]. Henaff et al. apply Sparse Coding (SC) to the log spectrogram and use the learned features to train a SVM classifier. The evaluation results provide further evidence for the effectiveness of the learned features for recognizing music genres [30]. Nam et al. introduce a pre-processing pipeline that improves the SC feature learning [31]; the evaluation results on a music tagging dataset compare favorably against the traditionally used audio features. More recently, SC-derived features have also been shown effective for music genre recognition [32] and music emotion recognition [33].

Based on the findings in the previous work, both DBNs and SC seem to be useful approaches for feature learning in music related tasks. Compared with DBNs, however, SC seems to have a broader range of applications in addition to the MGR task. Therefore, in this study, we propose to explore the viability of applying sparse coding for music performance assessment.

3. Method

3.1. *System Overview*

The proposed system comprises both a training and a testing phase. As shown in Fig. 2, the training phase starts by transforming each audio recording into the input representation; the input representations are collected for the training data and passed to the feature learning block. In the feature learning step, the feature extractor is derived by solving an optimization problem across the entire dataset, and the corresponding feature vector for each recording is calculated. Finally, the features are used to train a regression model that minimizes the loss between the model prediction and the ground truth (judges' ratings) for each recording, along with an outlier removal step to refine the model.

In the testing phase, a similar procedure is followed for each recording to prepare the input representation and extract features using the derived feature extractor. The resulting feature vector is then used to predict the judges' ratings with the pre-trained regression model. In the following sections, more details of each processing step are presented.

3.2. *Input Preparation*

The goal of the input preparation step is to normalize, reduce the amount of data, and to convert the data into meaningful input representations. First, audio recordings are down-mixed to one channel and resampled to a sampling rate of 22.05 kHz after decoding. All files are normalized to a numerical range between -1 and 1. Finally, two different matrix representations are computed, the Short-Time Fourier Transform (STFT) and a Local Histogram Matrix (LHM).

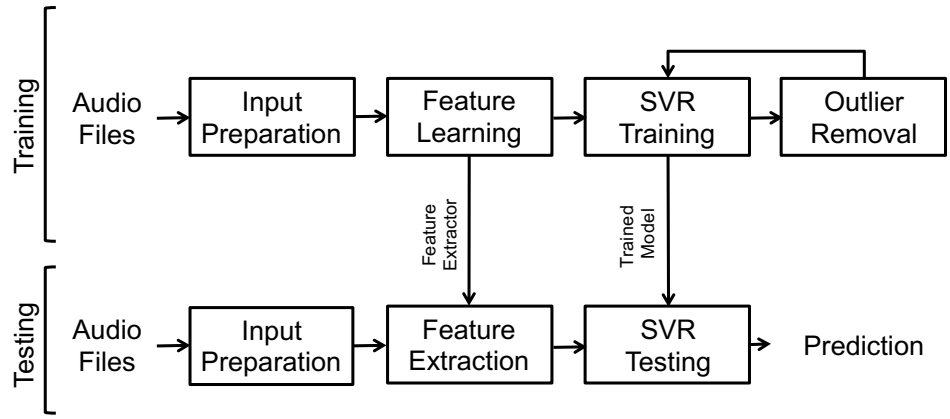

Fig. 2. The flowchart of the proposed method.

3.2.1. *STFT*

The STFT is computed using a block size of 512 audio samples, windowed with a Hann window. Neighboring blocks are overlapping by 75% (384 samples). Only the magnitude spectrogram is used and the phase information is discarded. The resulting spectrogram M_{stft} is a $m \times n$ matrix, in which $m = 257$ for the number of frequency bins and n equals the number of blocks. This input representation has been referred to by some authors as baseline representation [34, 35].

3.2.2. *LHM*

The histogram input representation aims to capture the most important overall characteristics of the percussive instrument performances. The extraction process of the Local Histogram Matrix (LHM) is shown in Fig. 3. In order to extract the histogram, the input time-domain audio signal is first partitioned into non-overlapping local segments of length 10 s. This length enables us to capture information of the higher level structure such as music phrases. Within each segment, the Inter-Onset-Interval (IOI) histogram vector $\mathbf{v}_{\text{ioi}}$, the amplitude histogram vector $\mathbf{v}_{\text{amp}}$, and the averaged Mel Frequency Cepstral Coefficients (MFCC) vector $\mathbf{v}_{\text{mfcc}}$ is extracted. Concatenating these vectors will result in local histogram matrix M_{lhm} that represents each individual recording. The vectors are computed as follows:

(1) **IOI histogram vector ($\mathbf{v}_{\text{ioi}}$)**: First, the onset times, i.e., the start times of individual drum events are estimated within the local segment. This is done following a standard approach to onset detection [17]: a novelty function extraction based on spectral differences (spectral flux) between neighboring STFT blocks is followed by an adaptive median threshold in order to detect and pick peaks in the novelty function; the time of these peaks are the onset times. The result, therefore, is a vector $onset(i)$ of onset times in which i is the onset index.

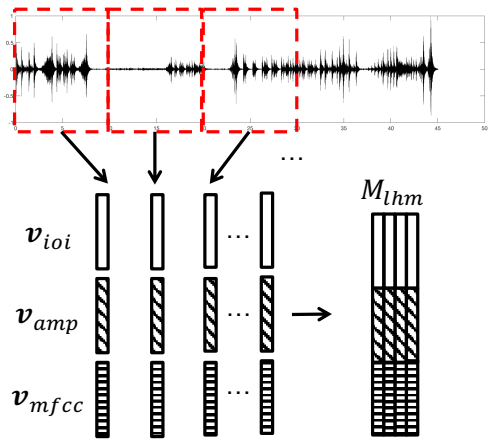

Fig. 3. Illustration of the construction process of local histogram matrix; $\mathbf{v}_{\text{ioi}}$ is the IOI histogram vector, $\mathbf{v}_{\text{amp}}$ is the amplitude histogram vector, $\mathbf{v}_{\text{mfcc}}$ is the averaged MFCCs vector, and M_{lhm} is the local histogram matrix.

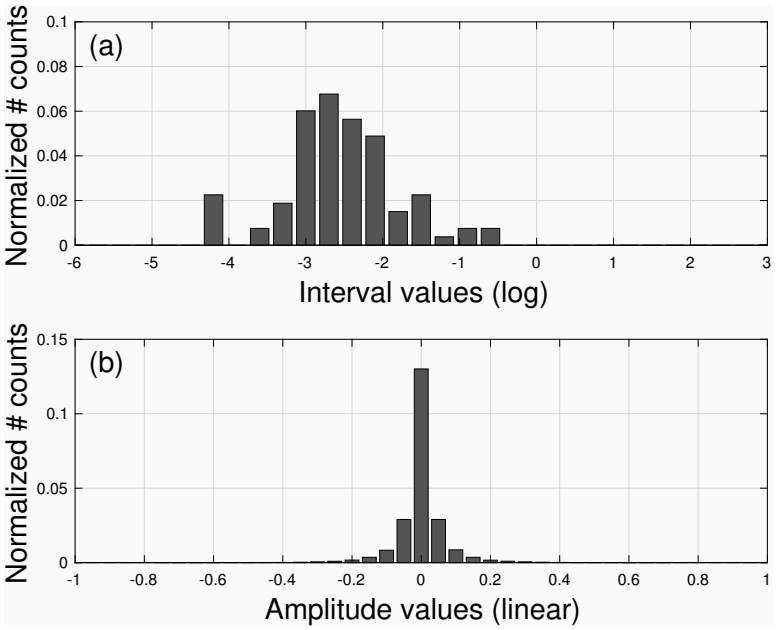

Fig. 4. An example of (a) an IOI histogram vector $\mathbf{v}_{\text{ioi}}$ and (b) an amplitude histogram vector $\mathbf{v}_{\text{amp}}$. All histograms are normalized by the total number of counts and can be interpreted as probability distributions.

Second, the IOI sequence is computed as the difference between neighboring onset times (Matlab code: `IOI = diff(onset(i))`), followed by a non-linear

transformation $IOI_{\ln} = \log(IOI)$. The reason for the non-linear transformation is that the ratios of onset intervals are frequently powers of 2 (for example, an 8th note is twice as long as a 16th note), so that the transformation helps to characterize the rhythmic patterns on a musically more meaningful numerical scale. Finally, the output $\mathbf{v}_{\text{ioi}}$ is computed by estimating the histogram from the $IOI_{\ln}$ with a pre-defined range and resolution, determined heuristically through data observation in pilot experiments. The histogram $\mathbf{v}_{\text{ioi}}$ is normalized to a sum of 1 by dividing it by the total number of onsets in the entire recording. The resulting $\mathbf{v}_{\text{ioi}}$ is a column vector with dimensionality $d_{\text{ioi}} = 31$. An example of $\mathbf{v}_{\text{ioi}}$ is shown in Fig. 4(a).

(2) **Amplitude histogram vector ($\mathbf{v}_{\text{amp}}$)**: Following a procedure similar to the one described above, $\mathbf{v}_{\text{amp}}$ is calculated by estimating the histogram of the amplitude values within the local segment. The range of the histogram is from -1 to 1 with a resolution of 0.05 between the consecutive bins. The $\mathbf{v}_{\text{amp}}$ is normalized by the total number of samples in the entire recording. The resulting vector is a column vector of dimensionality $d_{\text{amp}} = 41$. An example of $\mathbf{v}_{\text{amp}}$ is shown in Fig. 4(b).

(3) **Averaged MFCC vector ($\mathbf{v}_{\text{mfcc}}$)**: Given a local segment, the first 13 Mel Frequency Cepstral Coefficients (MFCCs) [36], a compact and widely used description of the shape of the spectral envelope of the signal are computed. The used STFT parametrization equals the one defined in Sect. 3.2.1. This leads to a $13 \times n_{\text{b}}$ MFCC matrix where n_{b} is the number of blocks within the segment. This matrix $\mathbf{v}_{\text{mfcc}}$ is aggregated across the n_{b} blocks, resulting in an output column vector of dimensionality $d_{\text{mfcc}} = 13$.

The same procedure will repeat for each 10-second local segment, resulting in the final matrix M_{lhm} is a $m' \times n_{\text{s}}$ matrix, in which

$$m' = d_{\text{ioi}} + d_{\text{amp}} + d_{\text{mfcc}} = 85 \tag{1}$$

and n_{s} is the number of local segments.

3.3. *Feature Learning*

3.3.1. *Sparse Coding*

The feature learning algorithm used in this paper is Sparse Coding (SC), which can be expressed as

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \frac{1}{2}\|X - D\alpha\|_2^2 + \lambda\|\alpha\|_1, \tag{2}$$

with $X \in R^{m \times n}$ as the input matrix, i.e., M_{stft} or M_{lhm}. D is the $m \times k$ dictionary matrix, α is the $k \times n$ sparse matrix, λ is the sparsity coefficient, n is the number of local segments, and k is the user-defined dictionary size.

This ℓ_1 regularized Least Absolute Shrinkage and Selection Operator (LASSO) problem can be solved by the Least Angle Regression (LARS) algorithm efficiently[37],

in which the dictionary D and the sparse matrix α can be learned by iteratively minimizing the reconstruction loss. Finally, the resulting dictionary D can be considered as the feature extractor, where the corresponding sparse representation α can be used to compute the features.

To compute the final feature vector that represents each audio recording, our system first learns an universal dictionary D_{all} from the entire dataset. This can be done by concatenating the M_{stft} or M_{lhm} across all the training files, and solve the LASSO optimization problem on the concatenated matrix X_{all}. Next, the $\alpha_{\text{individual}}$ is estimated from each recording by substituting the X in Eq. 2 with the input representation of an individual file $X_{\text{individual}}$ while keeping the $D = D_{\text{all}}$ fixed throughout the optimization process. The resulting $\alpha_{individual}$ is a $k \times n$ sparse matrix. Finally, $\alpha_{\text{individual}}$ is aggregated using mean and standard deviation across n segments, producing a feature vector $\alpha_{\text{final}} = [mean(\alpha_{\text{individual}}); std(\alpha_{\text{individual}})]$ with a dimensionality of $d_{\text{final}} = 2 \times k$. $[;]$ is a vector concatenation operator.

In our experiment, the Matlab implementation for SC from the open source library SPAMS[c] [38] is used. The parametrizations $k = \{32, 64, 128\}$ are tested, and a sparsity coefficient $\lambda = 1/\sqrt{block\ size}$ is applied.

3.3.2. *Convolutional Autoencoder*

In order to investigate other feature learning approaches as well, we included another baseline system for feature learning: the Convolutional Auto-encoder (CAE). The inclusion of this system allows us to compare the Sparse Coding-based system with a neural feature learning system, providing more insights on feature learning in the context of music performance assessment. The architecture of the CAE in this paper is shown in Fig. 5. The CAE feature learning process starts by taking a Mel-spectrogram X of the recording as the input. The network is trained to output X', which is the reconstruction of the input. There are four convolutional layers with {32, 16, 8, 4} channels of 3×3 kernels in the encoder. A batch normalization layer and a max-pooling layer of (2, 1) are added to each convolutional layer. These specific max-pooling parameters are chosen in order to maintain the temporal resolution and extract block-wise features from the input. The bottleneck layer is also a convolutional layer with 4 channels of 3×3 kernels. All of the non-linear units in this CAE are Rectified Linear Units (ReLUs). The decoder has a structure symmetric to the encoder with the max-pooling layers replaced by the up-sampling layers. The selected loss function for the training process is the Mean Squared Error (MSE) between X and X'; this optimization process is achieved using a gradient-descent-based algorithm, and the number of training epochs is 30.

To extract the features from the CAE, a process inspired by Choi et al. [39] is used. As shown in Fig. 5, this process first computes the intermediate outputs from all the layers in the encoder (including the bottleneck layer). Next, these outputs are

[c]http://spams-devel.gforge.inria.fr Last accessed: 2017/10/15

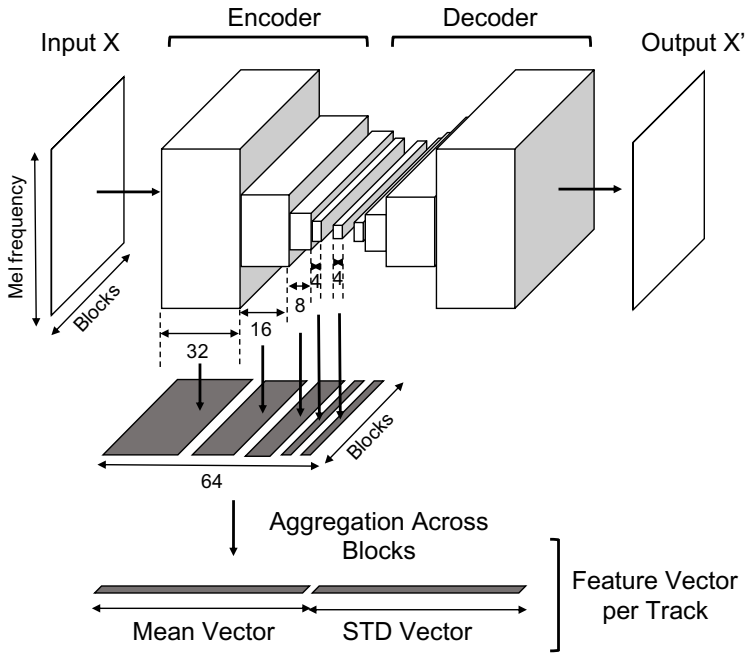

Fig. 5. The architecture of the CAE and the feature extraction process. Note that the mean and standard deviation (STD) vectors are computed across the blocks; these two vectors are concatenated to summarize the entire recording. X is a $64 \times N$ Mel-spectrogram in dB scale. 64 is the number of Mel frequency bins.

aggregated across the Mel-frequency axis through averaging. Finally, the aggregated outputs are stacked into a $64 \times N$ feature matrix, where N is the number of blocks. To derive the final feature vector for each recording, these frame-wise features are further aggregated by computing the mean and standard deviation across the blocks. This is similar to the process described in Sect. 3.3.1. Finally, the concatenation of mean and standard deviation vector is used to represent the entire recording of the student's music performance. The dimensionality of this final vector is $d_{\text{cae}} = 128$.

3.4. *Regression Model*

The regression model used is Support Vector Regression (SVR) with a linear kernel. The Matlab implementation of this algorithm from the open source library libsvm [40], is used with default settings. Due to the limited size of sample pool (see Sect. 4.1 for more details), a Leave One Out (LOO) cross-validation scheme is applied to our evaluation process. The main idea of LOO is to sequentially select one sample from the pool as test data and use the rest of the pool as training data until all the samples have been tested. Additionally, an outlier removal process is implemented by iteratively removing the sample with the largest residual between the prediction and

the actual label until 5% of the entire data is eliminated. This process potentially helps the regressor to better capture the underlying patterns of the data.

4. Experiments

4.1. *Dataset*

The dataset used for this study is provided by the Florida Bandmasters Association (FBA). This dataset contains audio recordings from the Florida all-state auditions of three groups of students: middle school (7th and 8th graders), concert band (9th and 10th graders), and symphonic band (11th and 12th graders) for three consecutive years (2013 to 2015). A total number of 19 types of instruments are included in the auditions. As a result, this dataset can be split into different subsets of recordings by their year, group, and instrument (for instance, 2013, middle school, clarinet). Each subset contains up to 180 recordings of student performances.

In each recording, a student is required to perform several exercises, such as technical etude, lyrical etude, chromatic scale, 12 major scales, and sight-reading. Each exercise is graded by expert judges using assessment categories such as *musicality*, *note accuracy*, *rhythmic accuracy*, *tone quality*, *artistry*, and *articulation*. The number of judges and the grading criteria are not available. The maximum score of these categories vary from 5 to 40. In our experiments, all of the ratings are normalized to a range between 0 and 1 by dividing the score with the maximum allowed value of the corresponding category. More information on the dataset can be found in our Github repository.[d]

In this study, the focus is on assessing percussion performance. For percussion instruments, the audition session includes 5 different exercises, which are mallet etude, snare etude, chromatic scale (xylophone), 12 major scales (xylophone), and sight-reading (snare). To further narrow down the scope of this study, we use only the subset of middle school snare etude from all three years. This particular combination is selected for containing a comparably high number of students. As shown in Table 1, a total number of 274 recordings of snare etude with an averaged duration of 51.3 s is available for analysis. For this particular exercise, the assessment categories are musicality (L1) and rhythmic accuracy (L2).

4.2. *Experiment Setup*

This paper presents two experiments that highlight different characteristics of the proposed feature learning method in the context of assessing student snare drum performances. The goal of Experiment 1 is to compare the effectiveness of two different input representations (see Sect. 3.2) for SC. For Experiment 2, the goal is to find the best combination of feature sets in order to achieve the highest performance.

In Experiment 1, the tested configurations are:

[d]https://github.com/GTCMT/FBA2013, Last access: 2017/10/15

Table 1. Statistics of the middle school snare etude from 2013 to 2015

Middle school/ Percussion/ Snare Etude			
Year	**#Students**	**Total Dur (sec)**	**Average Dur (sec)**
2013	98	4953	50
2014	90	4595	51
2015	86	4608	53

(1) **SC (STFT)**: sparse coding features using STFT as input representations and
(2) **SC (LHM)**: sparse coding features using LHM as input representations.

Both configurations are tested using $k = \{32, 64, 128\}$.

In Experiment 2, the regression model trained on the SC learned features using the proposed LHM with $k = 32$ is compared to two different sets of features, referred to as *Baseline* and *Designed* features, as proposed by Wu et al. [10]. The regression performance of the following feature set combinations is tested:

(1) **Baseline**: the standard spectral and temporal features such as spectral centroid, spectral rolloff, spectral flux, zero-crossing rate, and 13 MFCCs. The dimensionality is $d_{\text{baseline}} = 68$.
(2) **Designed**: the designed rhythmic and dynamic features derived from the IOI and amplitude histograms. These features involve the calculation of various statistics, such as crest, skewness, flatness, kurtosis, etc., directly from the histograms of the entire recording (compare [10] for more details). The dimensionality is $d_{\text{designed}} = 18$.
(3) **CAE**: see Sect. 3.3.2
(4) **SC (LHM)**: see Sect. 3.3.1.
(5) **Designed + Baseline**: a combined feature set consisting of both baseline and designed features.
(6) **SC + Baseline**: a combined feature set consisting of both SC and baseline features.
(7) **CAE + Baseline**: a combined feature set consisting of both CAE and baseline features.
(8) **SC + Designed**: a combined feature set consisting of both SC and designed features.
(9) **CAE + Designed**: a combined feature set consisting of both CAE and designed features.

4.3. *Metrics*

The performance of the models is investigated using the following standard statistical metrics: the Pearson correlation coefficient r and the coefficient of determination, i.e.,

Table 2. Evaluation results for Experiment 1: L1 represents musicality, and L2 represents rhythmic accuracy. The higher is better.

Input Representation		STFT		LHM	
Dictionary Size	Metrics	L1	L2	L1	L2
k = 32	r	0.34	0.19	**0.65**	**0.57**
	R^2	0.08	-0.02	**0.41**	**0.29**
k = 64	r	0.41	0.26	**0.70**	**0.50**
	R^2	0.11	-0.00	**0.45**	**0.06**
k = 128	r	**0.41**	0.28	0.33	**0.34**
	R^2	**0.08**	**-0.07**	-0.08	-0.78

R^2. These metrics are commonly used to evaluate the strength of the relationship between the regression predictions and ground truth. Details of the mathematical formulations can be found in [41].

4.4. *Experiment Results*

In this section, the evaluation results of both experiments are presented and discussed. Since all of the correlation results are significant ($p \ll 0.05$), their p-values are not reported.

The evaluation results of Experiment 1 are shown in Table 2. The following trends can be observed: first, the LHM input representation outperforms the STFT representation in almost every case. This result clearly shows that for this task, LHM is a more effective input representation than STFT. A likely explanation is that this discrepancy originates from the representations' capabilities of capturing temporal information. Since the STFT only represents the spectral content at various single instances, it does not encapsulate any temporal dependencies between meaningful audio events such as consecutive drums hits. This temporal information, while being likely to reside in the sparse matrix α, will be lost after the feature aggregation. The absence of temporal information apparently poses a problem for SC to learn higher level music concepts such as rhythm. LHM, on the other hand, captures some temporal dependencies between the audio events with the IOI histograms. This allows for the rhythmic information to be translated into the SC dictionary and to reflect on the final features, leading towards a better performance. Second, both STFT and LHM perform poorly for the largest dictionary size with $k = 128$. The reason behind this degradation for both representations might be due to the limited size of the sample pool. When $k = 128$, the resulting feature dimensionality, as described in Sect. 3.3, would be $k \times 2 = 256$. This will result in a feature matrix of 274×256 (students $\times$ features), and the model could suffer from the curse of dimensionality [42]. One potential solution to this problem would be to apply dimensionality reduction methods such as Principal Component Analysis (PCA) or feature selection, however,

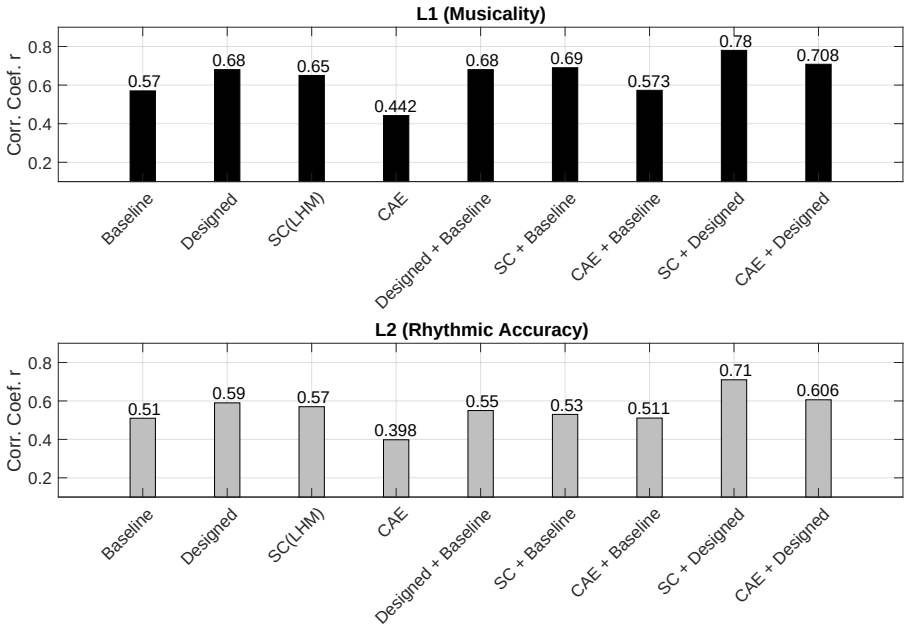

Fig. 6. Evaluation results of Experiment 2 for correlation coefficient r (top) L1 (bottom) L2

this is out of the scope of this study. Third, when $k = 32$, the LHM has the largest improvement over STFT in both L1 and L2. This result might indicate a relationship between the best performing k and the number of samples in the dataset.

The evaluation results of Experiment 2 are shown in Fig. 6 and Fig. 7. The first observation we can make is that both the designed and the SC features outperform the baseline features. This is in line with our expectations, as the baseline features are low-level instantaneous features that are not optimized to the task. Second, the proposed SC features achieve comparable performance to the designed features. The fact that SC features perform similarly to the designed features is encouraging, suggesting the possibility of deriving viable features through feature learning with minimum effort in feature crafting. Third, when combined with baseline features, both designed and SC features exhibit almost no improvements. There exist two possible explanations for this. On the one hand, it could be that these features do not add any information that is not already in the other features. On the other hand, this issue could be similar to the situation in Experiment 1 when having a large k, as adding baseline features might induce the curse of dimensionality by introducing too many features to the current size of sample pool. Four, among all the tests in Experiment 2, the highest performance is achieved with the combination

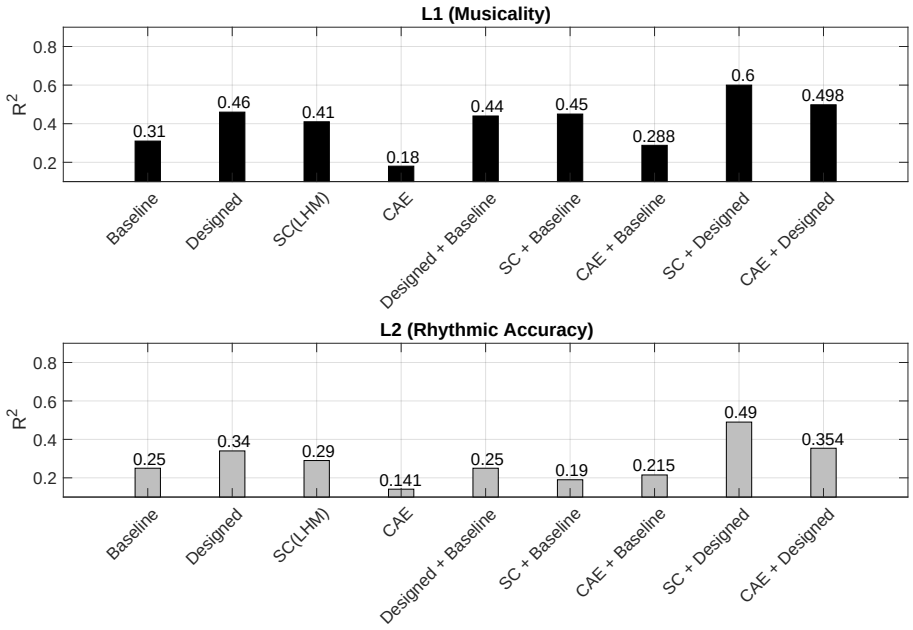

Fig. 7. Evaluation results of Experiment 2 for R^2 (top) L1 (bottom) L2

of designed and SC features. The result implies the effectiveness of SC features in capturing information that is non-redundant to the information provided by the designed features. Five, the CAE generally shows disappointing results. It seems to not be able to learn essential information and apparently encapsulates only limited domain knowledge, resulting in a performance lower than the Baseline. This could be an issue with either the input representation (compare Experiment 1) or maybe the choice of the loss function not focusing the network on the "features" of interest. It is worth noting that the CAE is able to learn something: although the CAE results are always lower than the Baseline results, the combination of CAE and Designed always tends to outperform the combination of Baseline and Designed, hinting at the network learning something that is not properly represented by the designed features. This is the case even when the combination CAE and Baseline is never able to clearly improve the results compared to the baseline only.

This demonstrates the potential of using feature learning methods to complement designed features due to the learned features' ability to capture relevant information not or incompletely modeled by features designed with expert knowledge. It is clear, however, that more work is needed to optimize the training process of feature learning systems. Modifications in input format and training procedure can help

ensuring that the algorithm learns the essential features.

A general observation that can be made from all experiment results is the superior model performance for L1 over L2. As discussed in [10], musicality is an abstract but holistic measure of the performance, and it is therefore more likely to reflect the overall perception of the judges and is more consistent across the years.

5. Conclusion

This article presents unsupervised feature learning to derive features for assessing percussive instrument performances. Specifically, we propose to use a histogram-based input representation to SC in order to allow the sparse coding to take advantage of temporal rhythmic information. We could show that — for the task of performance assessment — this input representation outperforms an STFT-based input representation that is frequently used for feature learning approaches. The learned features perform comparable to expert-designed features for this task, and are able to capture task-relevant information that is not represented in the designed features, indicating the suitability of combining "traditional" feature design methods with feature learning approaches for optimal performance. The work also showcases the dependency of feature learning performance on input representation. The autoencoder results highlight the differences between feature learning algorithm and the necessity for optimized input representations.

In summary, the contributions of this work are: First, it provides insights into the selection of input representations for feature learning in the context of snare drum performance assessment, favoring representations inspired by domain knowledge over standard representations such as STFT. Second, the proposed learned features achieve comparable results with the designed features, demonstrating the viability of deriving competitive features with minimum effort in feature design. Finally, combining the designed features with the SC features, the highest performance can be achieved. This result suggests that learned features might be able to provide complementary information to the features designed with domain knowledge, further optimizing the performance of a given task.

Possible future directions of this work include:

(1) Evaluate and compare the efficiency of other feature learning algorithms. Methods such as DBN[29] and denoising autoencoders [43] may provide new insights into the finding of the best feature learning strategy for automatic performance assessment.
(2) Investigate the reasons for the poor autoencoder performance. A different input format, even disregarding the suggestions of previous research, might enable the autoencoder to learn more task-relevant information. Similarly, an adapted or re-designed loss function could help to force the autoencoder to learn relevant data.
(3) Adapt the proposed method to other types of instruments such as Brass or Woodwind instruments. The evaluation results on melodic instruments could

lead to the refinement of the proposed feature learning scheme.

(4) Try to allocate more data. To further investigate the robustness and generality of this approach, more data is needed. Additionally, the increase in sample pool can also help the investigation into issues such as the relationship between dictionary size k and the number of samples. This might lead to a better strategy for selecting the best performing dictionary size.

(5) Study the relation of performance data and judges' assessments. Can we quantify which properties most influence the grade? What are the qualities of a performance that are most crucial with respect to the assessment? Answers to questions such as these can give us new insights into music performance, its properties, and its assessment.

Music Performance Analysis and assessment is a multifaceted topic, and the traditional custom-designed features might not capture all information that is needed for a detailed analysis. Complementing the set of features with learned features is a promising direction that we intend to explore further.

Acknowledgments

The authors would like to thank the Florida Bandmasters Association for providing the dataset used in this study.

References

[1] C. Palmer, "Music performance," *Annual Review of Psychology*, vol. 48, pp. 115–138, 1997.

[2] B. C. Wesolowski, S. A. Wind, and G. Engelhard Jr., "Examining Rater Percision in Music Performance Assessment: An Analysis of Rating Scale Structure using the Multifaceted Rasch Partial Credit Model," *Music Perception*, vol. 33, no. 5, pp. 662–678, 2016.

[3] S. Thompson and A. Williamon, "Evaluating evaluation: Musical performance assessment as a research tool," *Music Perception*, vol. 21, no. 1, pp. 21–41, 2003.

[4] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions," *Journal of Intelligent Information Systems*, July 2013.

[5] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[6] C. Dittmar, E. Cano, J. Abeßer, and S. Grollmisch, "Music Information Retrieval Meets Music Education," in *Multimodal Music Processing*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012, vol. 3, pp. 95–120.

[7] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[8] Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbull, "Music emotion recognition: A state of the art review," in

Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), 2010.

[9] P. Herrera, A. Dehamel, and F. Gouyon, "Automatic labeling of unpitched percussive sounds," in *Proceedings of Audio Engineering Society (AES) Conference*, 2003.

[10] C.-W. Wu, S. Gururani, C. Laguna, A. Pati, A. Vidwans, and A. Lerch, "Towards the Objective Assessment of Music Performances," in *Proceedings of International Conference on Music Perception and Cognition (ICMPC)*, 2016, pp. 99–102.

[11] A. Vidwans, S. Gururani, C.-W. Wu, V. Subramanian, and R. V. Swaminathan, "Objective descriptors for the assessment of student music performances," in *Proceedings of Audio Engineering Society (AES) Conference on Semantic Audio*, 2017.

[12] A. Lerch, *Software-Based Extraction of Objective Parameters from Music Performances*. München: GRIN Verlag, 2009.

[13] E. F. Clarke, "Understanding the psychology of performance," in *Musical Performance A Guide to Understanding*, J. Rink, Ed. Cambridge: Cambridge University Press, 2002.

[14] F. Dorian, *The History of Music in Performance The Art of Musical Interpretation from the Renaissance to Our Day*. New York: W. W. Norton & Company Inc, 1942.

[15] L. B. Meyer, *Emotion and Meaning in Music*. Chicago: University of Chicago Press, 1956.

[16] J. Beran and G. Mazzola, "Analyzing Musical Structure and Performance A Statistical Approach," *Statistical Science*, vol. 14, no. 1, pp. 47–79, 1999.

[17] A. Lerch, *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. John Wiley & Sons, 2012.

[18] P. Desain and H. Honing, "Does expressive timing in music performance scale proportionally with tempo?" *Psychological Research*, vol. 56, pp. 285–292, 1994.

[19] B. H. Repp, "The dynamics of expressive piano performance: Schumann's 'Trumerei' revisited," *Journal of the Acoustical Society of America (JASA)*, vol. 100, no. 1, pp. 641–650, 1996.

[20] W. Goebl, "Melody lead in piano performance: Expressive device or artifact?" *Journal of the Acoustical Society of America (JASA)*, vol. 110, no. 1, pp. 563–572, 2001.

[21] T. M. Walker, "Instrumental Difference in Characteristics of expressive musical performance," Dissertation, The Ohio State University, Columbus, 2004.

[22] S. Dixon, "Automatic Extraction of Tempo and Beat From Expressive Performances," *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, Mar. 2001.

[23] C. E. Seashore, *Psychology of Music*. New York: McGraw-Hill, 1938.

[24] T. Nakano, M. Goto, and Y. Hiraga, "An automatic singing skill evaluation method for unknown melodies using pitch interval accuracy and vibrato features," in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, 2006, pp. 1706–1709.

[25] J. Abeßer, J. Hasselhorn, C. Dittmar, A. Lehmann, and S. Grollmisch, "Automatic quality assessment of vocal and instrumental performances of ninth-grade and tenth-grade pupils," in *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2013, pp. 975–988.

[26] Y. Han and K. Lee, "Hierarchical approach to detect common mistakes of beginner flute players," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 77–82.

[27] B. Bozkurt, O. Baysal, and D. Yret, "A Dataset and Baseline System for Singing Voice Assessment," in *Proceedings of the International Symposium on CMMR*, Matosinhos, 2017.

[28] H. Lee, Y. Largman, P. Pham, and A. Y. Ng, "Unsupervised feature learning for

audio classification using convolutional deep belief networks," *Advances in Neural Information Processing Systems*, pp. 1–9, 2009.

[29] P. Hamel and D. Eck, "Learning Features from Music Audio with Deep Belief Networks." in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2010, pp. 339–344.

[30] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun, "Unsupervised learning of sparse features for scalable audio classification." in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2011, pp. 681–686.

[31] J. Nam, J. Herrera, M. Slaney, and J. Smith, "Learning sparse feature representation for music annotation and retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2012, pp. 565–570.

[32] K.-C. Hsu, C.-S. Lin, and T.-S. Chi, "Sparse coding based music genre classification using spectro-temporal modulations," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 744–750.

[33] C. O'Brien, "Supervised feature learning via sparse coding for music information rerieval," Master's thesis, Georgia Institute of Technology, 2015.

[34] L. Su, L.-F. Yu, and Y.-H. Yang, "Sparse cepstral and phase codes for guitar playing technique classification," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 9–14.

[35] L. Su, H. M. Lin, and Y. H. Yang, "Sparse modeling of magnitude and phase-derived spectra for playing technique classification," *IEEE/ACM Transactions on Speech and Language Processing*, vol. 22, no. 12, pp. 2122–2132, 2014.

[36] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug. 1980. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1163420

[37] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[38] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2009, pp. 1–8.

[39] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[40] C.-C. Chang and C.-J. Lin, "LIBSVM : a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[41] J. T. McClave and T. Sincich, *Statistics*, 9th ed. Prentice Hall, 2003.

[42] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Sprinter, 2006.

[43] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2008, pp. 1096–1103.