LEARNING PITCH INVARIANTS FOR INSTRUMENT RECOGNITION

Names should be omitted for double-blind reviewing
Affiliations should be omitted for double-blind reviewing

ABSTRACT

Musical instruments offer a wide range of pitches, nuances, and expressive techniques. Identifying instruments within an audio stream We show that Mel-frequency cepstral coefficients (MFCC), classically used in audio signal analysis, lack invariance with respect to realistic pitch shifts. We introduce a representation learning system, based on convolutional neural networks, that outperform engineered features in a supervised task of single-label instrument classification. We further improve the We extend our method to the recognition of multiple instruments playing simultaneously.

1. INTRODUCTION

Among the cognitive attributes of musical tones, pitch is distinguished by a combination of three properties. First, it is relative: ordering pitches from low to high gives rise to intervals and melodic patterns. Secondly, it is intensive: multiple pitches heard simultaneously produce a chord, not a single unified tone – contrary to loudness, which adds up with the number of sources. Thirdly, it is invariant to instrumentation: this makes possible the transcription of polyphonic music under a single symbolic system.

Besides this invariance property, understanding the influence of pitch in audio streams is paramount to the design of an efficient system for automated classification, tagging, and similarity retrieval in music.

Section 2 demonstrates that pitch is the major factor of variability among musical notes of a given instrument, if described by their Mel-frequency cepstra. Section 3

Time-frequency representations, such as the constant-Q wavelet scalogram, are a useful first step for the construction of pitch-adaptive features.

2. HOW INVARIANT IS THE MEL CEPSTRUM ?

The MFCCs were extracted from a filterbank of 40 Mel-frequency bands and 13 discrete cosine transform coefficients.

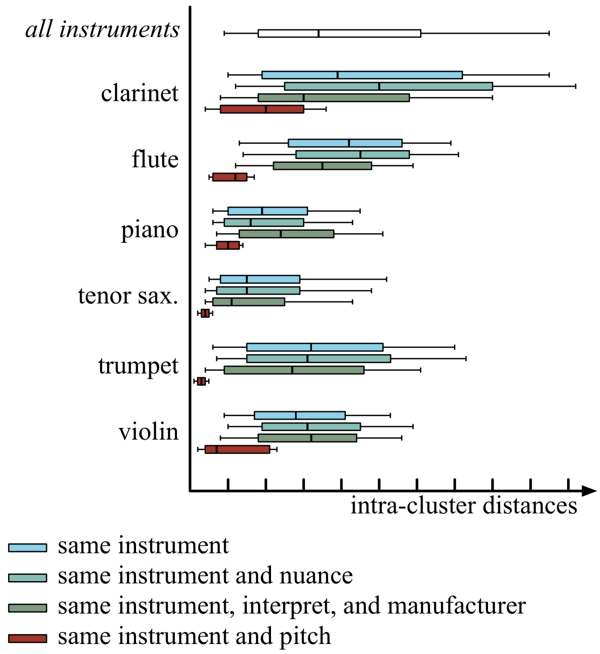

Figure 1: Distributions of squared Euclidean distances among various clusters in the RWC dataset. Whisker ends correspond to lower and upper deciles. See text for details.

3. DEEP CONVOLUTIONAL NETWORKS

3.1 Time-frequency representation

We used the implementation from the librosa package [4] with $Q = 12$ filters per octave, center frequencies ranging from 55 Hz to 14 kHz (8 octaves from A1 to A9), and a hop size of 23 ms. Furthermore, we applied nonlinear perceptual weighting of loudness in order to reduce the dynamic range between the fundamental partial and its upper harmonics. A 3-second sound excerpt $\boldsymbol{x}[t]$ is represented by a time-frequency matrix $\boldsymbol{x_1}[t, k_1]$ of width $T = 128$ samples and height $K_1 = 96$ MIDI indices, i.e. 8 octaves.

3.2 Architecture

First of all, we apply a family $\boldsymbol{W_2}[\tau, \kappa_1, k_2]$ of $K_2 = 50$ learned time-frequency convolutional operators, whose supports are constrained to have width Δt and height Δk_1.

$$\boldsymbol{W_2} \overset{t,k_1}{*} \boldsymbol{x_1} = \sum_{\substack{0 \leq \tau < \Delta t \\ 0 \leq \kappa_1 < \Delta k_1}} \boldsymbol{W_2}(\tau, \kappa_1, k_2) \boldsymbol{x_1}[t-\tau, k_1-\kappa_1] \quad (1)$$

Furthermore, element-wise biases $\boldsymbol{b_2}[k_2]$ are added to the convolutions, resulting in the tensor

$$\boldsymbol{y_2}[t, k_1, k_2] = \boldsymbol{b_2}[k_2] + (\boldsymbol{W_2} \overset{t,k_1}{*} \boldsymbol{x_1})[t, k_1, k_2]. \quad (2)$$

The second step is the application of a pointwise nonlinearity. We have chosen the *rectified linear unit* [ReLU] because of its popularity in computer vision and its computational efficiency.

$$\boldsymbol{y_2^+}[t, k_1, k_2] = \max(\boldsymbol{y_2}[t, k_1, k_2], 0) \quad (3)$$

To achieve invariance to translation as well as frequency transposition, we pool neighboring units in the time-frequency domain (t, k_1) over non-overlapping rectangles of width Δt and height Δk_1.

$$\boldsymbol{x_2}[t, k_1, k_2] = \max_{\substack{0 \leq \tau < \Delta t \\ 0 \leq \kappa_1 < \Delta k_1}} \left\{ \boldsymbol{y_2^+}[t-\tau, k_1-\kappa_1, k_2] \right\} \quad (4)$$

We apply a family $\boldsymbol{W_3}[\tau, \kappa_1, k_2, k_3]$ of K_3 convolutional operators that perform a linear combination of time-frequency feature maps in $\boldsymbol{x_2}$ along the channel variable k_2.

$$\boldsymbol{y_3}[t, k_1, k_3] = \sum_{k_2} \boldsymbol{b_3}[k_2, k_3] + \boldsymbol{W_3}[t, k_1, k_3] \overset{t,k_1}{*} \boldsymbol{x_2}[t, k_1, k_2] \quad (5)$$

After nonlinear rectification and max-pooling, the layer $\boldsymbol{y_3}$ turns into a non-negative tensor $\boldsymbol{x_3}[t, k_1, k_3]$.

$$\boldsymbol{y_4}[k_4] = \sum_{t, k_1, k_3} \boldsymbol{W_4}[t, k_1, k_3, k_4] \boldsymbol{x_3}[t, k_1, k_3] \quad (6)$$

We apply nonlinear rectification, yielding $\boldsymbol{x_4}[k_4] = \boldsymbol{y_4^+}[k_4]$. $\boldsymbol{y_5}[k_5] = \sum_{k_5} \boldsymbol{W_5}[k_4, k_5] \boldsymbol{x_4}[k_4]$.

$$\boldsymbol{x_5}[k_5] = \frac{\exp \boldsymbol{y_5}[k_5]}{\|\exp \boldsymbol{y_5}\|_1} \quad (7)$$

The above ensures that the coefficients of $\boldsymbol{x_5}$ are non-negative and sum to one, hence can be fit to a probability distribution. We define the categorical cross-entropy as

$$\mathscr{L}(\boldsymbol{x_5}, \mathcal{I}) = - \sum_{k_5 \in \mathcal{I}} \log \boldsymbol{x_5}[k_5]. \quad (8)$$

The goal is to minimize the average loss $\mathscr{L}(\boldsymbol{x_5}, \mathcal{I})$ for across all pairs $(\boldsymbol{x_5}, \mathcal{I})$ in the training set.

3.3 Training

The network is trained on categorical cross-entropy over shuffled mini-batches of size 512 with uniform class distribution. The learning rate policy for each scalar weight in the network is *Adam* [3], a state-of-the-art online optimizer for gradient-based learning.

4. DEEP SUPERVISION OF MELODIC CONTOUR

4.1 Disentangling pitch from timbre

$$\mathscr{L}[\boldsymbol{x_2}, \mathcal{P}] = - \sum_{(t, k_1) \in \mathcal{P}} \log \sigma \left(\sum_{k_2} \boldsymbol{x_2}[t, k_1, k_2] \right) \quad (9)$$

4.2 Joint supervision

5. SINGLE-INSTRUMENT CLASSIFICATION

5.1 Experimental design

In order to train the proposed algorithms, we used MedleyDB v1.1. [1], a dataset of 122 multitracks annotated with instrument activations as well as melodic f_0 curves when present. We extracted the monophonic stems corresponding to a selection of eight pitched instruments [see Figure 1]. Stems with leaking instruments in the background were discarded. The evaluation set consists of 120 recordings of solo music collected by Joder et al. [2]. We discarded recordings with extended instrumental techniques, since they are under-represented in MedleyDB.

5.2 Results

6. POLYPHONIC CLASSIFICATION

6.1 Experimental design

6.2 Results

7. CONCLUSIONS

8. REFERENCES

[1] Rachel Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Bello. Medleydb: a multitrack dataset for annotation-intensive mir research. *International Society for Music Information Retrieval Conference*, 2014.

[2] Cyril Joder, Slim Essid, and Gaël Richard. Temporal integration for audio classification with application to musical instrument classification. *IEEE Transactions on Audio, Speech and Language Processing*, 17(1):174–186, 2009.

[3] Diederik P. Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations*, pages 1–13, 2015.

[4] Brian McFee, Matt McVicar, Colin Raffel, Dawen Liang, Oriol Nieto, Eric Battenberg, Josh Moore, Dan Ellis, Ryuichi Yamamoto, Rachel Bittner, Douglas Repetto, Petr Viktorin, Joo Felipe Santos, and Adrian Holovaty. librosa: 0.4.1. zenodo. 10.5281/zenodo.18369, October 2015.