STRUCTURAL SEGMENTATION WITH CONVOLUTIONAL NEURAL NETWORKS MIREX SUBMISSION

Thomas Grill **Jan Schlüter**
Austrian Research Institute for Artificial Intelligence, Vienna
{thomas.grill,jan.schlueter}@ofai.at

ABSTRACT

This submission to the MIREX 2015 Music Structural Segmentation task employs a Convolutional Neural Network (CNN) to identify boundaries within a piece of digital audio. The network was trained on a combination of mel-scaled log-magnitude spectrograms (MLSs) and self-similarity lag matrices (SSLMs) with two-level human structural annotations following the SALAMI guidelines. It is based on our work presented in Grill and Schlüter [3]. Apart from detecting boundaries, our submission also attempts to assign labels to the resulting segments using a simple model based on 2D-DCTs and a cosine distance measure.

1. INTRODUCTION

In order to detect structural segment boundaries in digital audio, we use an artificial neural network trained in a supervised fashion on human-annotated data. For this, we formulate boundary prediction as a binary classification problem: Given an excerpt of an audio signal, decide whether there is a structural boundary at its center or not. Once we have a model solving this problem, we can apply it to a sequence of excerpts extracted in a sliding-window fashion to obtain a curve of boundary probabilities. We search for peaks in this curve in order to predict boundaries in the given music piece.

Here, the music excerpts are represented as mel-scaled log-magnitude spectrograms (MLSs) and a pair of self-similarity lag matrices (SSLMs), the classifier is a Convolutional Neural Network (CNN), and the human-annotated data is an excerpt of the public SALAMI dataset [6] plus additional data annotated according to the same guidelines. The training data was carefully selected to be disjoint from the three datasets used in the MIREX evaluation campaign.

In [3], our method achieved results considerably outperforming any submission from MIREX 2012 to 2014 on a subset of the SALAMI dataset, which contains both classical and popular music recorded under studio conditions and in live concerts. For MIREX 2015, we submit the best-performing neural network of [3] tuned for an evaluation time tolerance of ± 0.5 seconds, with a slight modification on feature preprocessing.

2. METHOD

The different components of our method are detailed in [3], with references to [4] and [7]. Here, we will only give an overview and point out what has changed compared to the previously published work.

2.1 Feature Extraction and Preprocessing

From the audio signal, we compute a mel-scaled logarithmic-magnitude spectrogram (MLS) of 80 bands. To be able to train and predict on spectrogram excerpts near the beginning and ending of a music piece, we apply a simple padding strategy for the MLS features. If the first (or last, respectively) non-zero spectrogram frame has a mean volume of $\geq$-40 dBFS, we assume an abrupt boundary and pad the spectrogram with a -100 dBFS constant. Conversely, we pad with repeated copies of this first or last non-zero spectrogram frame. To either padding, we add ± 3 dB of uniform noise. This is different from [3], where a padding with low-volume pink noise was used. The resulting MLS is subjected to a HPSS decomposition, yielding a pair of harmonic and percussive MLS components (see Figure 1). A second feature pair is generated from the unpadded MLS in the form of self-similarity lag matrices (SSLMs) with short range (14 seconds) and long range (88 seconds) lag time, respectively. For the SSLMs, the front and back padding is done in a cyclic (wrap-around) manner, as if the audio is looped.

2.2 Network Architecture and Training

The architecture and training procedure is identical to the one described in [3, Section 3.3]. The software runs in Python, using numpy, scipy [5], Theano [1] and Lasagne [2] packages.

Our networks are trained and validated on a set of 733 music pieces annotated according to the SALAMI guidelines, but disjoint from the three datasets used in the MIREX Music Structural Segmentation evaluation [7, Section 4]. We used 633 pieces for training, and 100 pieces for validation, to find the best-performing configurations both for our study in [7] and for our MIREX submission.

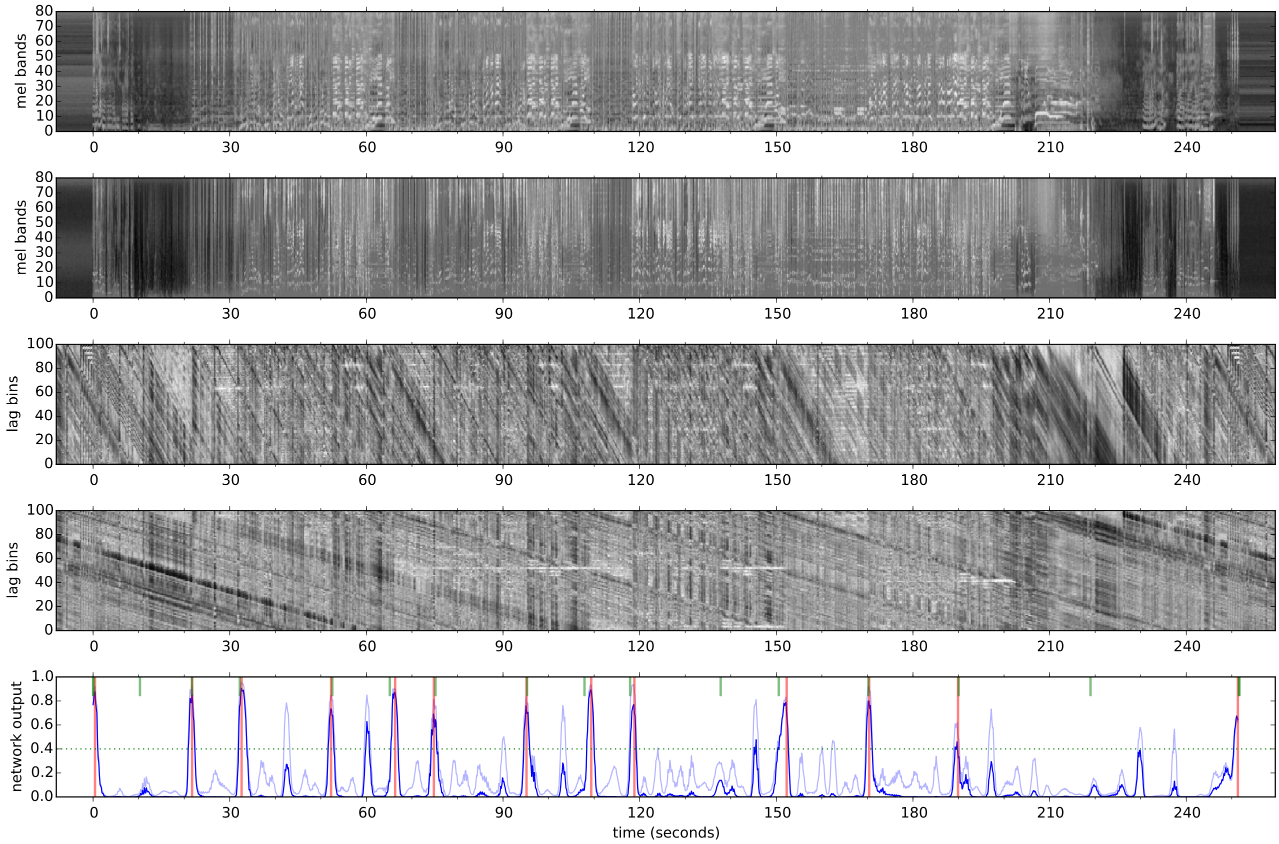

Figure 1. Example features (MLS-HPSS harmonic and percussive components and SSLMs, short and long range) and network output for *The Weight* by *Rachel Weber*, SALAMI id 1304. For every time frame of the time-synchronous features, the network computes an output value. Concatenating all values, we obtain a curve of probabilities (bottom panel, opaque blue) for first-level boundaries for the entire music piece. The probability output for second-level boundaries is shown in faint blue. Local maxima of the probability curve are boundary candidates; thresholding and windowing them selects the boundary predictions (red). Ground-truth annotations are shown as short vertical bars (green).

2.3 Boundary prediction from network output

After training, the networks are applied to pieces of music. For every spectrogram excerpt, the network computes a scalar output between 0 and 1, which can be interpreted as the probability of a boundary occurring at the center of the excerpt. By applying the network to a sequence of excerpts, advancing a single time frame between each, we obtain a curve for the entire music piece (this can be efficiently implemented as a series of convolutions and a final dot product). With peak-picking, windowing and thresholding, we obtain boundary locations from this curve. The peak-picking threshold for a given network is chosen to optimize the boundary retrieval F_1-score on the validation set. See Figure 2 for an illustration of the threshold optimization.

For improved results, we train four identically-parameterized networks (instead of five in [3], for efficiency reasons), starting from different random weight initializations, and average their output before peak-picking. This is a standard technique known as *bagging*.

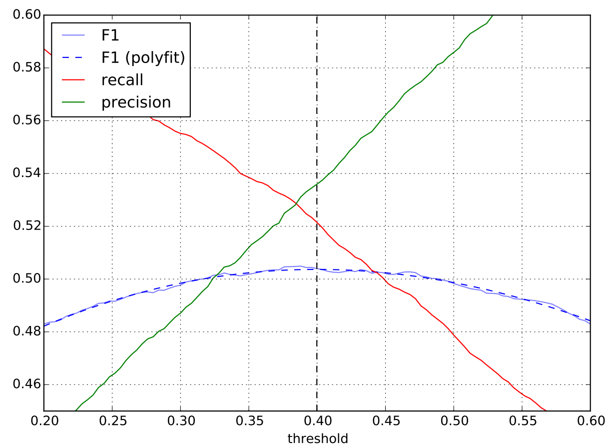

Figure 2. Threshold optimization performed on the validation data set (100 music pieces), with the F_1, precision and recall being the respective mean results over the data set for a specific threshold value. The optimal value is the maximum position of a polynomial curve fitted to the F_1 results.

2.4 Labeling

In order to apply labels to the segments retrieved by the strategy outlined in the above sections, we apply a simplistic model. This is just for the sake of labeling at all, and will be much refined in future contributions.

The mel-scaled log-magnitude spectrogram (before HPSS decomposition) is segmented at the detected boundaries, and each part is subjected to a two-dimensional DCT-II transformation. We keep a fixed number of components for both temporal and spectral dimensions and omit the static components (zero-index DCT bins).

These segment models $\mathbf{x}_i$ are compared in a pairwise manner using a cosine distance measure $\delta_{\cos}(\mathbf{x}, \mathbf{y}) = 1 - \left\langle \frac{\mathbf{x}}{\|\mathbf{x}\|}, \frac{\mathbf{y}}{\|\mathbf{y}\|} \right\rangle$ and a penalty factor (with an adjustable exponent p) for logarithmic differences in segment durations d_i

$$D_{i,j} = \delta_{\cos}(\mathbf{x}_i, \mathbf{x}_j)\, e^{|ln(d_i) - ln(d_j)|p}. \quad (1)$$

The inter-segment distances are grouped using hierarchical clustering [1] with average/UPGMA linkage, and a 'distance' criterion with threshold t.

Experiments on the validation data set have revealed that the best results are obtained when all spectral DCT bins are retained (79 bins), but only 9 bins in the temporal dimension, blurring the representation of temporal evolution. The threshold t has been set to 0.7, and the penalty exponent for duration difference p to 0.525. See Figure 3 for an illustration of the segment models and resulting labels for the same music piece as in Figure 1.

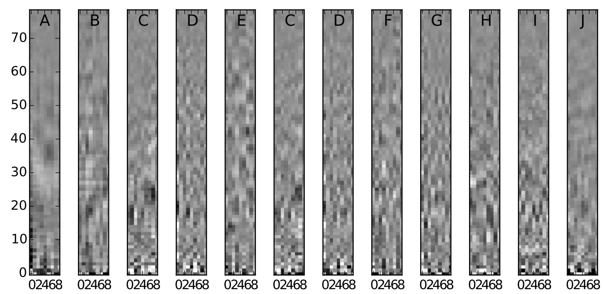

Figure 3. 2D-DCT segment models and resulting labels for the identified segments in *The Weight* by *Rachel Weber*, SALAMI id 1304. Labels C and D denote repeated segments.

3. ACKNOWLEDGMENTS

This research is funded by the Federal Ministry for Transport, Innovation & Technology (BMVIT) and the Austrian Science Fund (FWF) through project TRP 307-N23 and the Vienna Science and Technology Fund (WWTF) through project MA14-018.

[1] `http://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html`, accessed 2015-08-14

4. REFERENCES

[1] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proc. of the Python for Scientific Computing Conference (SciPy)*, June 2010.

[2] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, Jeffrey De Fauw, Michael Heilman, diogo149, Brian McFee, Hendrik Weideman, takacsg84, peterderivaz, Jon, instagibbs, Dr. Kashif Rasul, CongLiu, Britefury, and Jonas Degrave. Lasagne: First release., August 2015. [Online; accessed 2015-08-14].

[3] Thomas Grill and Jan Schlüter. Music Boundary Detection Using Neural Networks on Combined Features and Two-Level Annotations. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, Malaga, Spain, 2015.

[4] Thomas Grill and Jan Schlüter. Music Boundary Detection Using Neural Networks on Spectrograms and Self-Similarity Lag Matrices. In *Proceedings of the 24th European Signal Processing Conference (EUSIPCO 2015)*, Nice, France, 2015.

[5] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2015-08-14].

[6] Jordan Bennett Louis Smith, John Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J Stephen Downie. Design and creation of a large-scale database of structural annotations. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 555–560, 2011.

[7] Karen Ullrich, Jan Schlüter, and Thomas Grill. Boundary Detection in Music Structure Analysis using Convolutional Neural Networks. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, Taipei, Taiwan, 2014.