Time-series segmentation and latent representation of musical instruments

Gregory Szep

King's College London

August 7, 2018

Abstract

Music information retrieval tasks serve as faithful benchmarks for time-series analysis pipelines due to the availability of strongly labelled training data such as MusicNet. Clustering algorithms in spectral sub-spaces, hidden Markov models and causal convolutional neural networks are compared in their ability to transform time-series to a continuous latent space that clusters eleven orchestral instruments. The latent space is evaluated quantitatively with precision-recall metrics obtained by comparing the instrument prediction from a segment of audio to the ground truth obtained from musical scores, and qualitatively by generating samples of audio for given regions in the latent space.

1 Methodology Outline

1.1 Mapping time-series to latent space

The input data are single channel time-series points $\mathcal{D} = \{x(t_1) \ldots x(t_N)\}$ sampled at frequency f from an underlying continuous state-time process $x(t)$, that is the oscillating sound waves emitted by a live orchestra.

1.1.1 Wavelet transforms and independent components

1. get spectrogram using windowed Fourier transform or Gabor transforms, discuss spectral leakage, contrast, normalisation and noise filtering in an unsupervised way
2. perform principal components and independent component dimensionality reduction along frequency dimension, discuss differences between them.

1.1.2 Markov models and expectation-maximisation

1. discuss implementations of random Markov fields in image segmentation and how they can be adapted to audio segmentation
2. random Markov field vs hidden Markov model?
3. expectation-maximisation vs error back-propagation

1.1.3 Feature extraction with causal convolutions

Convolutional architectures have become popular due to their ability to compress spatio-temporal information for discrimination and generation tasks [**?**, **?**]. A causal convolutional network [**?**] — which encodes the arrow of time in its architecture — is trained for the audio segmentation task.

1. dilated causal convolutions as a merge between a feature extractor and a dimensionality reduction technique. This is a supervised method
2. Compare clusters to those obtained in Section 1.1.1

1.2 Clustering in latent space

1. We have a hierarchical clustering problem: there are 11 instruments each of which can play 28-83 notes. The easier problem is to only cluster instruments, the harder problem is to cluster both instruments and notes.

1.2.1 K-means

1. Since we know how many instruments there are, we can apply a naive K-means and see what happens. Here the disadvantage is that time-ordering may be ignored, which can lead to noisy/discontinuous audio segment classifications
2. Discuss de-noising strategies in post-classification: possibly Markov random fields from section 1.1.2?

1.2.2 Fully convolutional networks

1. I shall attempt to adapt the the fully convolutional architecture [**?**] for the audio segmentation task. Discuss advantages of end-to-end trained solution.

1.3 Evaluation methods

1.3.1 Audio segment retrieval

1. outline of object detection / segmentation in image analysis

2. precision-recall metric applied to audio

1.3.2 Instrument generation

1. introduction to encoder / decoder pipelines as generative models which can produce data given activation of input in latent space.
2. attempt to produce sounds that are interpolations between existing instruments, assess qualitatively how realistic they sound

Figure 1: Summary of methodology showing all stages of the audio segmentation task. Each transition between sub-figures can be achieved with appropriate algorithms

2 Dataset Description

2.1 Input and Labels

1. small summary table of the MusicNet dataset [**?**], advantages over EEG and other bio-sensory data for bench-marking signal processing algorithms
2. raw labels are time aligned transcripts of the sheet music. How to we parse that into instrument activations and note activations.
3. cross-validation — if any

2.2 Data Partitioning

1. test, validation and train sets
2. cross-validation — if any

3 Results, Protocols and Conclusions

1. all relevant figures will be presented here
2. all code will be documented on GitHub repo [https://github.com/gszep/sound-detection]

3.1 Learned feature maps

3.2 Clustering performance

3.3 Generating instruments