Sound Object Labeling

Hugo Flores García

Fall 2021
Sound object labeling

Bongjun Kim (Winter 2019)
Organizing large sample libraries

Grouping tracks in a DAW

Navigating large music recordings by content
Goal

- Building a system that automatically labels an audio event

Bongjun Kim (Winter 2019)
Machine Learning: Classification
Overview of general classification tasks

Input data → Feature representation → Classifier → Label

A vector of numbers
\[ \vec{x} = \langle a_1, a_2, \ldots, a_n \rangle \]
...that represent attributes of the example.

- Decision Tree
- Nearest Neighbor
- Neural Networks

“Cat”
“Piano”

Bongjun Kim and Hugo Flores García
Classification Tasks

Input data $\rightarrow$ Feature representation $\rightarrow$ Classifier $\rightarrow$ Label

- Feature space should easily discriminate between classes

Classifier “draws” decision boundary

Dog barking

Door knock

Bongjun Kim and Hugo Flores García
Feature selection is important

- how points in the feature space cluster is important
The same feature space could be meaningful for different ways of classifying data.
K-Nearest Neighbor (KNN) Classifier

• When you see a new instance $x$ to classify, find the most similar training example(s) and assign their label to the instance.

• How do you tell what things are similar?
  1. Extract proper features.
  2. Measure distance / similarity in the feature space.

Bongjun Kim (Winter 2019)
K-Nearest Neighbor (KNN) Classifier

Considering 4 nearest neighbors (k=4), X is probably a Class-1.
Now that we know..

Input data → Feature representation → Classifier → Label

KNN

How do we extract meaningful representations from waveforms?

Bongjun Kim and Hugo Flores García
Audio event classification

\[ \mathbf{x} = \langle a_1, a_2, \ldots, a_n \rangle \]

Input data → Feature representation

Bongjun Kim and Hugo Flores García

How do we extract meaningful representations from waveforms?
Some audio recording basics
A diagram illustrating the internal components of a microphone, including wires carrying an audio signal, a magnet, a coil, and a diaphragm.
Voltage over time

Bryan Pardo
Why not use the waveform as a feature representation?
Why not use the waveform as a feature representation?

1 second of audio at 44.1kHZ → 44,100 values!

It's hard to find meaningful patterns!

van den Oord et al. 2016

Need a very powerful model (like a deep neural net) which requires millions of training examples.
Why not use the waveform as a feature representation?

van den Oord et al. 2016

How do we preprocess the audio waveform to obtain meaningful representations?

Bongjun Kim and Hugo Flores García
Commonly used audio features

- **Zero-crossing rate**
  - Time-domain feature
  - Rate of sign changes in a signal
  - Low for harmonic sounds, high for noisy sounds

* Figure: https://en.wikipedia.org/wiki/Zero_crossing
Commonly used audio features

- Zero-crossing rate

Guitar

Snare drum

White noise

Bongjun Kim (Winter 2019)
Commonly used audio features

• Spectral centroid
  • Frequency domain feature
  • The weighted mean of the frequencies in the signal
  • Known as a predictor of the “brightness” of a sound

* figure: https://librosa.github.io/librosa/generated/librosa.feature.spectral_centroid.html
Commonly used audio features

• Spectral centroid
Example: Drum Transcription
Automatic drum transcription

- Let’s build a drum transcription machine only using spectral centroid features
Automatic drum transcription

Onset detection:
librosa.onset.onset_detect
Automatic drum transcription

- Segmentation
- Cutting the recording every \(<onset-2048\) samples>
Automatic drum transcription

- Extracting spectral centroid from each segment

Bongjun Kim (Winter 2019)
Automatic drum transcription

Bongjun Kim (Winter 2019)
Automatic drum transcription-2

• More challenging example
Automatic drum transcription-2

- Onset detection might not work that well on this example, but let’s assume we have perfect onset info
Automatic drum transcription-2

- Segmentation and feature extraction

- The previous example
Automatic drum transcription-2

• More challenging example

You can find more feature extraction functions in the Librosa package
Commonly used audio features

• Spectrogram
  • Plots the magnitude of the frequency spectrum as a function of time.

Lo-res image
(Usually 256x199 for 1 second of audio)

Lower dimensionality than a pure waveform, but it is still high dimensional!
Commonly used audio features

- Mel Frequency Cepstral Coefficients (MFCCs)

- Plots the envelope of the spectrum with just a few coefficients (usually 13)

- Approximately 10 times smaller than a spectrogram!

- The standard for speech recognition before deep learning!
Deep Embeddings

- Can we use a neural net to generate meaningful features?

1s audio (16kHz) -> log-Mel spectrogram -> Convolutional Neural Net -> 128-dim embedding vector
Deep Embeddings: Transfer Learning

Train on a “pretext” task to learn a meaningful internal representation! (aka transfer learning)
Deep Embeddings: VGGish (Simonyan et al. 2015)

The original “deep audio embedding”

Trained on an Audio Tagging task on Audioset (subset of YouTube)

https://medium.com/@yxu71/freesound-tagging-by-vggish-with-knn-731dc3e1dc5a
Deep Embeddings: OpenL3

- **L3-Net (aka OpenL3)**

  Predict whether an audio clip and an image correspond to each other (audiovisual correspondence)

  Train on LOTS of data (all of YouTube if you want!)

  pretext task — only for learning a meaningful representation

  No labels needed! (Self-supervised)

  512-dim or 6144-dim embedding vector

R. Arandjelovic et al., 2017 and Cramer et al., 2019

Update Parameters

Corresponds? (Yes/No)

Compute Loss
Deep Embeddings: TriCycle (Cartwright et al. 2019)

Given an audio clip, predict temporal cycles!

self-supervised: all you need are the timestamps!

512-dim embedding vector
Dimensionality Reduction

Aka how can we visualize high-dimensionality embedding spaces?

1024-dim embedding vector

2-dim projection of embedding space
**Goal:** find a linear projection of your dataset, such that you can keep the axes with the **most** variation

**Dimensions with most variation**

===

“Principal components”

### PCA (Principal Component Analysis)

- **Principal Component #1**
  - Direction of most variation
- **Principal Component #2**
  - Direction of second most variation

t-SNE
(T-distributed Stochastic Neighbor Embedding)

**Goal:** find a *nonlinear* projection of your dataset, such that the *local relationships* between points are preserved.

*Iterative algorithm (slow)!*
Musical Instrument ID (MIID)
Philharmonia Dataset

- 14,000 sound samples of the Philharmonia Orchestra
- Mostly single notes of isolated instruments, 1-5s in length
- 19 melodic instruments + many percussion instruments

https://philharmonia.co.uk/resources/sound-samples/
Some Links I Shared

Google’s infinite drum machine:
https://experiments.withgoogle.com/ai/drum-machine/view/

VQGAN + CLIP:
https://colab.research.google.com/drive/1L8oL-vLJXVcRzCFbPwOoMkPKJ8-aYdPN#scrollTo=ix4T6qkRqZgi

huggingface spaces
https://huggingface.co/spaces