Denoising and Labeling Hydrophone Data

Philip Robinson

Oregon Health Sciences University

July 20, 2018

Abstract

There is a need for an index of acoustic events and associated metadata for University of Hawaii's Aloha Cabled Observatory (ACO) hydrophone recordings. Currently searching the audio recordings is done at human pace by listening to the audio stream, which is prohibitive at approximately 10 years of content. The proposal is to develop a processing system that creates simple data store of vocals and metadata associated with vocalization events, renders a denoised [1–3] copy of those vocalization events, and provide a simple time based retrieval system. The system is written using `python` and the `scipy` [5] libraries for signal processing. The extended goal of this work is to use whale vocalization as a proxy measurement of migration patterns, and attempt to identify migratory changes against known climate events. In order to accomplish this task, noise in audio recording should be smoothed or removed, vocalizations must be localized/indexed, features of speech must be extracted, species must be identified [6] or clustered.

1 Introduction

University of Hawaii's ACO has approximately 10 years of hydrophone data, in need of indexing, labeling, and cleaning. To enable further research and bootstrap more advanced indexing techniques, a data retrieval, analysis, and labeling tool is Provided is a simple module enabling these three features. The module provided include:

- reading & listen to ACO encoded data
- standard `datetime` units for navigation
- parameterized visualizations for audio units
- simple denoising via spectral subtraction
- center wave by empirical mode decomposition

2 Data Retrieval

The ACO data is gathered at N22°45.110′ W158°00′. The recordings are encoded as variable bit-width raw sensor readings (fixed-width for each chunk of 4096 samples). Each file is saved as a 5 minute chunk, named with it's datetime stamp. The set elected for this project was 24K samples per second (there exists a 96K as well).

```python
from aco import ACOio, datetime, timedelta

loader = ACOio('./basedir/')
target = datetime(
    day=18, month=2, year=2016,
    hour=7, minute=55
)

src = loader.load(datetime)
snip = src[
  timedelta(seconds=7):
  timedelta(seconds=11)
]
snip.View()
```

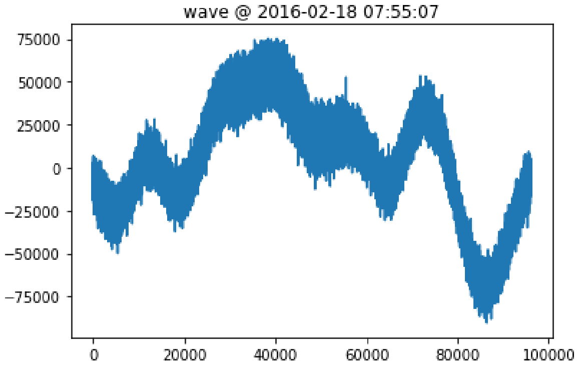

Figure 1: Raw Data

It is visible, from (Figure 1) that the direct current gain is not centered at zero, nor trivially accumulative. This is a consequence of changes in atmospheric pressure, due to the ocean's motion, attenuating the signal. Without these changes in atmospheric pressure the current gain can be corrected by removing the mean of the original signal, however in this case empirical mode decomposition is used to yield more accurate results. This will be addressed in greater detail later.

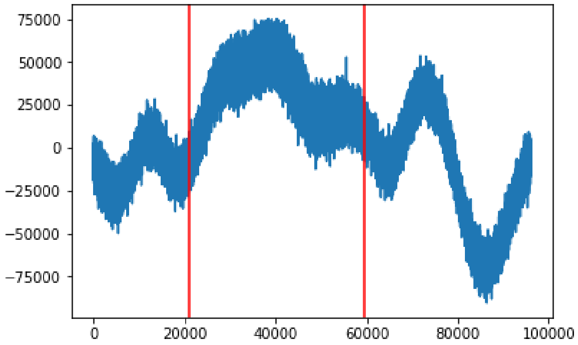

Figure 2: Raw Vocalization

It is also not obvious this track has a vocalization, highlighted in (Figure 2). This pattern is indicative of high amounts of noise. Usually, a better perspective is gained using the `logspectrum`, however as seen (Figure 3) this is mostly non-informative. This is for two reasons. Firstly, the dynamic range of the track is too great due to the direct current gain as mentioned above. Secondly, the sampling rate is too high, resulting in far too much data being expressed in a small amount of space, which this expensive run-time effects on complicated processing algorithms, like empirical mode decomposition.

```
snip.View(
  'logspectrum',
  frame_duration=.08,
  frame_shift=.02
)
```

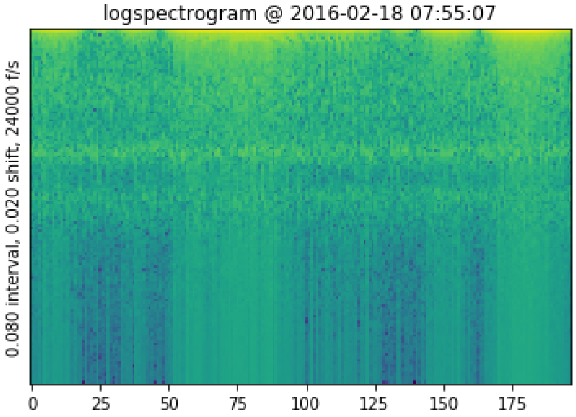

Figure 3: Raw Spectrogram

3 Spectral Subtraction

Spectral subtraction is a technique used to denoise signals with additive white noise (like found in this application). It is also, a very cheap algorithm with many modifications [3], which makes it an ideal first pass on the data.

The sampled signal y is modeled as the desired signal x and background noise b.

$$y[n] = x[n] + b[n]$$

At a single frame at point p and length L,

$$\begin{aligned} Y(pL,\omega) &= X(pL,\omega) + B(pL,\omega) \\ |Y(pL,\omega)|^2 &= |X(pL,\omega)|^2 \\ &\quad + |B(pL,\omega)|^2 + |X(pL,\omega) * B(pL,\omega)| \end{aligned}$$

We can estimate the background noise, given sufficiently small frames, by subtracting the mean power spectrum from a segment of only noise. Using Welch's method [4] we are able to estimate the mean power spectrum.

```python
def _spectral_subtraction(self, other,
    frame_duration=.08, frame_shift=.02,
    wtype='boxcar', alpha=5.0, beta=.01):

  Frames = self._Frame(
    frame_duration, frame_shift).data
  power = other.power(
    frame_duration, frame_shift, wtype)
  window = signal.get_window(
    wtype, self.to_frame_count(frame_duration))

  spectrum = np.fft.fft(Frames * window)
  amplitude = np.abs(spectrum)
  phase = np.angle(spectrum)

  _ = (amplitude ** 2.0) - (power * alpha)
  _ = np.maximum(_, beta * amplitude ** 2.0)
  _ = np.sqrt(_)
  _ = _ * np.exp(phase*1j)

  return _
```

Figure 4: Spectral Subtraction

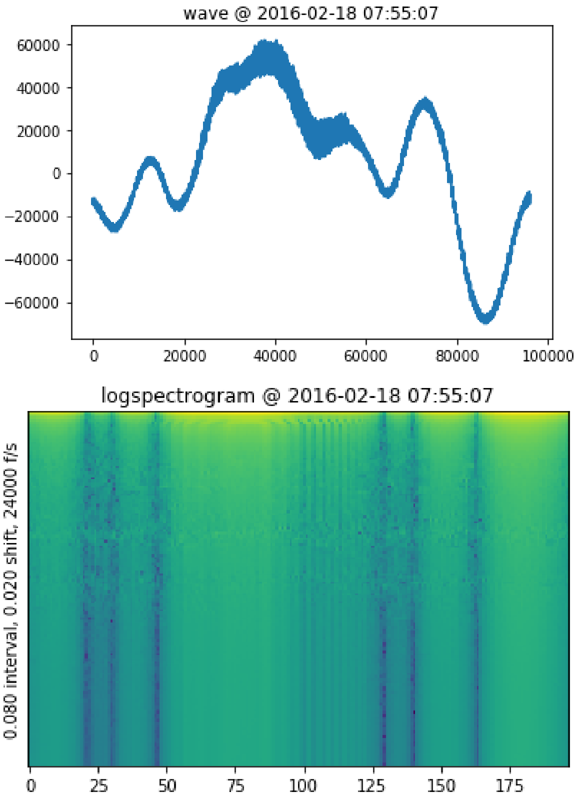

Figure 5: After Subtraction

Rather than the original spectral subtraction algorithm, (`Figure 4`) implements the algorithm found in [2]. This allows for adjustable α and β parameters, used in tuning some audio artifacts consequent of spectral subtraction.

From (`Figure 5`) it is much easier to see the vocalization section in the waveform. It is apparent that the spectrogram has less noise, but not necessarily in a useful way. This is due to the dynamic range and resolution problems mentioned above.

```
noise = snip[
  timedelta(seconds=0):
  timedelta(seconds=.8)
]
clean = snip.subtract(noise)
down = clean.resample(40000)
down.View(
  'logspectrum',
  frame_duration=.08,
  frame_shift=.02
)
```

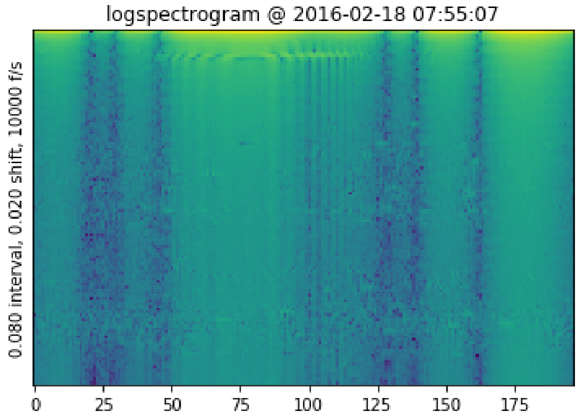

Figure 6: After Resamling

Down sampling the data, as shown (`Figure 6`), gives us the first visible pattern in the spectrogram. It is faint, due to far too high of dynamic range in the waveform.

4 Empirical Mode Decomposition

Empirical Mode Decomposition is used to address the variable gain found in this waveform, and allow us a more informative exploratory data analysis. Empirical mode decomposition expresses the current waveform as a summation of sine functions of learned mode. The results are stored, ordered by highest to lowest frequency. We can sum the first k levels of the yield to capture the majority of data and audible effects, while ignoring the low order functions (that encode the gain).

This is a computationally expensive procedure, that benefits from the down sampling step.

```
centered = down.remove_dc(levels=13)
centered.View()
centered.View(
  'logspectrum',
  frame_duration=.08,
  frame_shift=.02
)
```

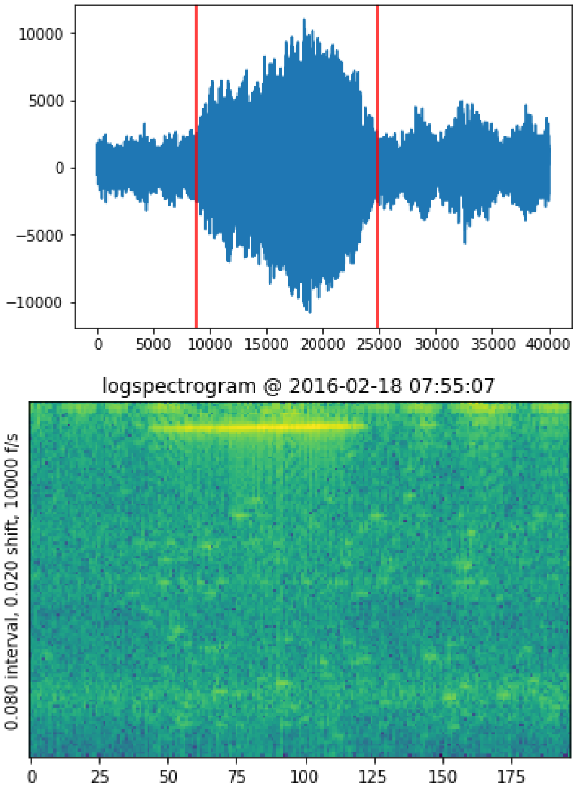

Figure 7: After EMD

5 Labeling

With this level of granularity, we can correctly label our original dataset using both auditory and visual queues.

```
noise.label_src_as_noise()
centered[
  timedelta(seconds=0.88):
  timedelta(seconds=2.48)
].label_src_as_voice()
```

Figure 8: Label Sections of Audio

6 Results

With the right parameters, this works very well to clean the data visually and audibly. Unfortunately, not all theses parameters have obvious or predictable trends. The goal of being able to with high precision label vocalization for later learned methods is definitely hit, and the ability to encode metadata enabling better audio track retrieval for human perception is on its way.

The code in its current form can be found here[1] with documentation coming soon. Once the datastore is setup in a shared location, there will be more opportunities for community contributions.

Finally, some of the parameters elected for this paper are not best fit, but better represent the challenges faced in this project. Actually a level of 5 and a resample of 10000 yield the audio files attached with the report.

References

[1] Venkitasamy Baskar, V. Rajendran, and E. Logashanmugam. Study of different denoising methods for underwater acoustic signal. 2015.

[2] Michael G. Berouti, Richard M. Schwartz, and John Makhoul. Enhancement of speech corrupted by acoustic noise. In *ICASSP*, 1979.

[3] Cliston Cole, Marc Karam, and Heshmat Aglan. Noise removal in speech processing using spectral subtraction. 05:1146–1147, 04 2008.

[4] Peter D. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. 15:70 – 73, 07 1967.

[5] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[6] L. Shamir, C. Yerby, R. Simpson, A. M. von Benda-Beckmann, P. Tyack, F. Samarra, P. Miller, and J. Wallin. Classification of large acoustic datasets using machine learning and crowdsourcing: Application to whale calls. *Acoustical Society of America Journal*, 135:953–962, February 2014.

[1] https://github.com/probinso/Whales