Spafe: Simplified python audio features extraction

Ayoub Malek

1 Youmday GmbH

Summary

This paper describes version 0.3.1 of spafe: a python package for audio features extraction based on the Numpy (Harris et al., 2020) and Scipy (Pauli Virtanen & Vázquez-Baeza, 2019) libraries. Spafe implements various features extraction techniques that can be used to solve a wide variety of recognition and classification tasks (speaker verification, spoken emotion recognition, spoken language identification etc.). The paper provides a brief overview of the library’s structure, theory and functionalities.

Statement of need

In speech processing, features extraction is essentially the estimation of a parametric representation of an input signal. This is a key step in any audio based modeling and recognition process (e.g. speech recognition, sound classification, speaker authentication etc.). There are several speech features to extract, such as the Linear Frequency Cepstral Coefficients (LFCC), Mel Frequency Cepstral Coefficients (MFCC), Linear Predictive Coding (LPC), and Constant-Q Cepstral Coefficients (CQCC) etc. Each type of features has its own advantages and drawbacks (e.g. noise robustness, complexity, inter-components correlation etc.) that can directly affect the researched topic. Unfortunately, existing libraries for extracting these features (e.g. librosa (McFee et al., 2015), python_speech_features (Lyons et al., 2020), SpeechPy (Torfi, 2018) and Bob (A. Anjos & Marcel, 2017)) are limited and mostly focus on one extraction technique (e.g. MFCC), thus it is hard to find reliable implementations of other features extraction algorithms. Consequently, this slows down the research and hinders the possibility of exploring and comparing these different approaches. Hence, the need for spafe, a straightforward solution that unites all these different techniques in one python package.

Introduction

The philosophy of spafe is keeping it simple, flexible and efficient in order to reach a wide range of developers and researchers. Hence, spafe is written in python 3 and only depends on Numpy (Harris et al., 2020) and Scipy (Pauli Virtanen & Vázquez-Baeza, 2019). The library is heavily documented with the help of Sphinx and tested using Pytest. Spafe supports mono signals processing and has been tested with different sampling rates (e.g. 8kHz, 16Khz, 44.1kHz, 48kHz etc.).

Scripts in spafe are divided into four major groups (see Figure 1):

- fbanks filter banks implementations.
- features features extraction implementations.
- frequencies frequencies based features extraction implementations.
- utils helper functions for pre- & post-processing and visualization etc.
Implementation and theory

Filter banks (spafe/fbanks)

A filter bank is defined as an array of band pass filters that splits the input signal into a set of analysis signals, each one carrying a single frequency sub-band of the original signal (Penedo et al., 2019; Sarangi et al., 2020). Each band pass filter is centered at a different frequency, called center frequency. The center frequencies are evenly spaced over a specified scaled frequencies range (e.g. bark scale, erb scale, mel scale etc.). The bandwidths of the filters increase with the frequency, in order to duplicate the human hearing properties, which are characterized by a decreasing sensitivity at higher frequencies. Within this context, spafe provides implementations for the following filter banks: Bark filter banks, Gammatone filter banks, Linear filter banks and the Mel filter banks.

Features (spafe/features)

In an attempt to cover most audio features, spafe provides various frequency and cepstral domain features extraction algorithms, both filter bank-based and auto-regression-based. The
following is a list of the available features extraction routines in the spafe python package:

- Bark Frequency Cepstral Coefficients BFCC
- Constant Q-transform Cepstral Coefficients CQCC
- Gammatone Frequency Cepstral Coefficients GFCC
- Linear Frequency Cepstral Coefficients LFCC
- Linear Prediction Cepstral Coefficients LPCC
- Mel Frequency Cepstral Coefficients MFCC
- Inverse Mel Frequency Cepstral Coefficients IMFCC
- Magnitude based Spectral Root Cepstral Coefficients MSRCC
- Normalized Gammachirp Cepstral Coefficients NGCC
- Power-Normalized Cepstral Coefficients PNCC
- Phase based Spectral Root Cepstral Coefficients PSRCC
- Perceptual Linear Prediction Coefficients PLP
- Rasta Perceptual Linear Prediction Coefficients RPLP

The following figure provides a summary of the included features extraction algorithms and their detailed steps:

![Figure 3: Features extraction algorithms in spafe](image)

In addition to the previously mentioned features, spafe allows for computing the following spectrograms: Bark spectrogram, Cqt spectrogram, Erb spectrogram and Mel spectrogram.

**Frequencies (spafe/frequencies)**

The frequencies modules in spafe focus specifically on the computation of dominant and fundamental frequencies. A dominant frequency is per definition the frequency carrying the maximum energy among all frequencies of the spectrum (Telgarsky, 2013), whereas the fundamental frequency (often noted as $F_0$) is defined as the inverse of the period of a periodic signal (Cheveigné & Kawahara, 2002).

**Utils (spafe/utils)**

The utils scripts, handle most of the input signal pre-processing steps including pre-emphasis, framing and windowing. They also include all the conversion computations needed to convert Hertz frequencies to other frequency scales. On top of that, all feature post-processing routines are in this group. This includes normalization, liftering, deltas computation and visualization.

**Conclusion**

This paper introduced spafe, a python package for audio feature extractions. Spafe provides a unified solution for audio features extraction, that can help simplify and accelerate the research of various audio-based recognition experiments.

**References**


