Feature Extraction

This tutorial illustrates two mainstream feature extraction methods applied in speech recognition:
i. Mel-spectrogram and
ii. Mel Frequency Cepstral Features (MFCCs).

In both cases the first step is a Fourier spectrogram computed using the sliding window approach with typically a shift of 10 msec. The phase is discarded and only the log magnitude spectrum is maintained. The further steps in the pipeline try to maintain the key content, but present it slightly different towards the recognition backend.

Note

The above statements need some clarification in the light of modern "end-to-end speech recognition systems". These are trained to map the raw speech waveform to text. However, they require huge amounts of data to be competitive. Sometimes they start from a generic pretrained feature extraction such as 'wav2vec'. Other high end systems use a mel spectrogram as input. In our opinion it has only been shown that starting from the waveform is *possible* and not that it is *advantageous* over starting from a spectral representation. And if not advantageous it must be concluded that starting from the waveform is computationally inefficient as it has been shown that the first layers in such deep neural net learn something very similar to a mel spectrogram.

From Spectrogram to Features

Starting from the log-magnitude spectrum we apply additional FEATURE EXTRACTION. The main purpose of this feature extraction is to help the pattern matching that will follow by:

- compacting the feature vector
- suppress disturbing, noisy side information at the same time
- transforming the feature vector to a domain better suited to the machine learning backend

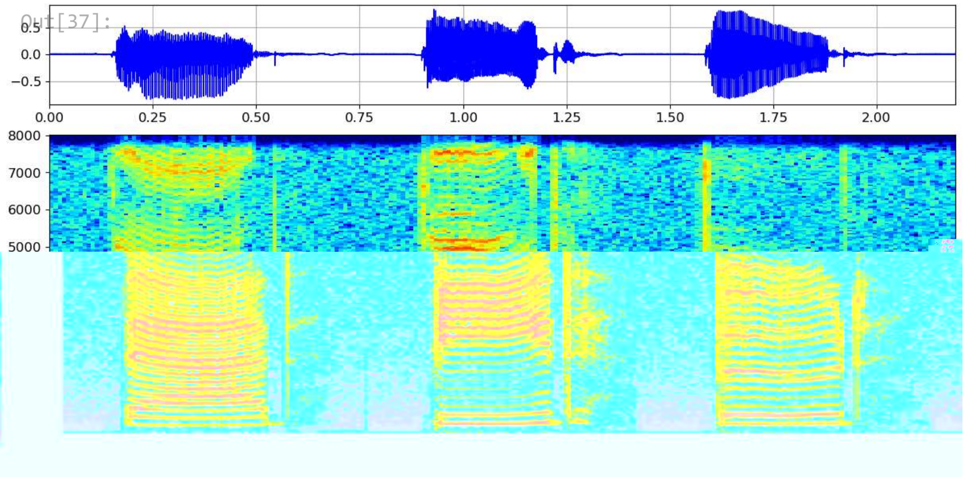
This notebook shows how to do 2 prototypical feature extractions for speech recognition and the last part shows how alternative features can be added

1. Mel Frequency Spectrum with splicing of multiple frames
2. Mel Frequency Cepstral Coefficients including Delta Features
3. Alternative Features

0. Load a Waveform File and compute the Fourier Spectrogram

All feature extraction methods shown here have a Fourier Spectrogram as first step in the pipeline

Out[37]:



1. MEL Spectrogram

When we use mel spectral features for speech recognition, it may not be simply the raw mel spectrogram that is fed to the recognizer. Typically it will be a small pipeline consisting of

Waveform → Fourier Spectrogram → Mel Spectrogam → Mean-Variance Nor[redacted]tacking of frames

STEP 0. Fourier Spectrum
Defines the sliding window parameters

STEP 1. Mel Spectrum

First the Fourier Spectrum is converted to a mel Spectrum. By this the frequency axis is warped as inspired by the human auditory system. The warping corresponds roughly in maintaining the frequency spacing at low frequencies (below 1kHz) and doing a logarithmic compression on the frequency axis as higher frequencies (above 1kHz). The high resolution mel spectrum (+- 80 channels) preserves both spectral envelope and pitch information in a single spectral representation. Within *spchlab* we are by default using 80 channels for 16kHz sampling rate and 64 channels for 8kHz sampling rate.

If you want to use both 8kHz, 11.25kHz and 16kHz files in a single system, you can either upsample the waveform or resample the 16kHz mel filterbank so that it is useful at other sampling frequencies as well.

STEP 2. Mean / Variance Normalization

In neural net and other machine learning systems it is common to *normalize* the feature vectors to zero mean and unit variance. [redacted] etc. Other motivations are valid as well. Mean variance normalization obviously reduces session to session variability. In (mel) spectrogram *mean normalization* is motivated as eliminating an arbitrary gain factor in the recording setup and has been common practice since long.

STEP 3. Stacking of Frames

[redacted] However from a speech recognition point of view, it is better to observe the short time spectrum in its context. This is simply achieved by stacking many adjacent vectors together.

We may e.g. stack N frames on the left and the right using a stride of 2 as shown below. This results in a receptive field of $2(N*s)+1$ frames wide. Using a stride of 2 allows for a wider span receptive field while maintaining a manageble dimension of the feature vector.

$$\underbrace{S_{i-N\times s} \quad \dots \quad S_{i-s} \quad S_i \quad S_{i+s} \quad \dots \quad S_{i+N\times s}}_{F_i}$$

In the example below we stack 5 frames to left and right and use a stride of 2 resulting in a receptive field of 210 msec and a feature vector dimension of 880 (for 80 mel coefficients). Before splicing the mel spectral features are mean and variance normalized.

2. Mel Frequency Cepstral Coefficients (MFCCs)

MFCCs were the feature vector of choice with HMM/GMM systems. They are still used in all kinds of situtations where training data is not abundant, thx to their compactness and decorrelation properties. In modern DNN systems trained on thousands of hours of data, these mathematical properties become highly irrelevant and MFCCs have been replaced by the raw melspectrum or even the raw waveform.

Waveform → Fourier Spectrum → Mel Spectrum → Mel Cepstrum → Add Delta Features

STEP 0. Fourier Spectrum
Defines the sliding window parameters

STEP 1. Mel Spectrum
A low resolution mel spectrum is adequate in the MFCC pipeline. It is common to use 20 channels for narrowband signals (8kHz sampling rate) and 24 channels for wideband signals.

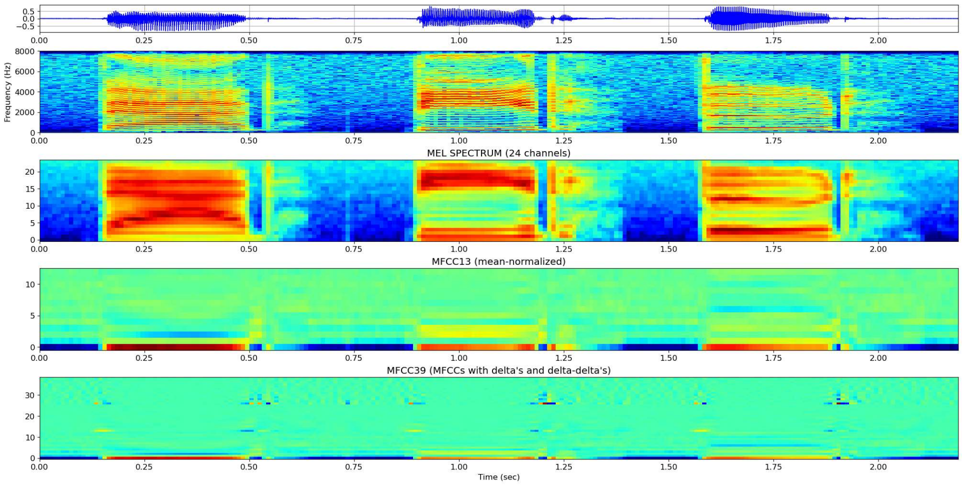
STEP 2. Mel Cepstrum
The mel cepstrum is computed as the IDCT (inverse discrete cosine transform) of the mel spectrum. Typically it is further truncated to 13 coefficients. By virtue of the IDCT the cepstrum contains the same information as the spectrum. However, the "truncated" cepstrum corresponds to the spectral envelope and suppresses the pitch. It has been shown that the cepstral coefficients are highly decorrelated one vs. the other making them suitable for metrics like the Euclidean distance and which makes statistical model much simpler.

STEP 3. Delta's and Delta-Delta's In order to increase the receptive field one could simply splice a few frames of cepstra together. This would, however, reintroduce correlation between successive frames. A beter option is to include so called "Delta"-features that compute the "trend" over time (first order derivatives) and possibly also double delta's computed a second order derivative. In practice the derivatives are approximated by simple regression formulas.

$$\Delta c_i = 2c_{i+2} + c_{i+1} - c_{i-1} - 2c_{i-2} \tag{1}$$
$$\Delta\Delta c_i = \Delta c_{i+1} - \Delta c_{i-1} \tag{2}$$

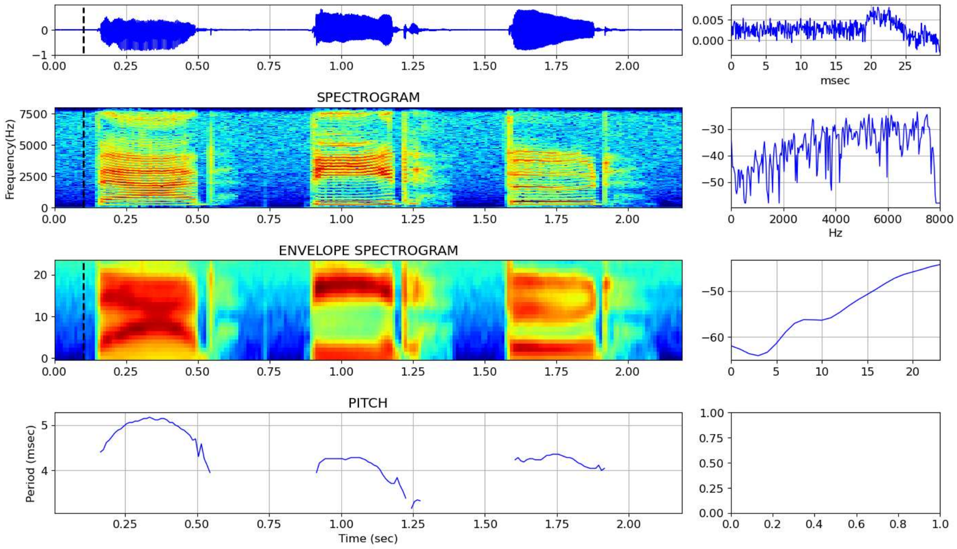
The final feature vector is obtained by stacking instantaneous cepstral features with their delta's and delta-delta's. Thus resulting in a 39-dimensional feature vector derived from a receptive field of 7 frames.

MFCCs are both compact and highly uncorrelated features. This makes them suitable in almost all circumstances and for all backends. Their main drawback is that the lossiness of the transform and the limited receptive field.



3. Some Alternatives

In the following you can see a multitude of spectral and pitch alternatives that could be included in the feature extraction:


msec
SPECTROGRAM
Frequency(Hz)
Hz
ENVELOPE SPECTROGRAM
PITCH
Period (msec)
Time (sec)