shizuka/listen

Dan Hawkins (stacktaxi)

January 11, 2020

Abstract

The `listen` module is an integrated denoiser, VAD, and ASR.

1 VAD

The VAD gets the cepstral coefficients of each frame via the standard algorithm:

$$X_{dct} = \mathrm{dct}\{\log(\mathrm{mel}\{X_n\})\} \tag{1}$$

where mel$\{\ldots\}$ is the output of the mel filterbank, and dct$\{\ldots\}$ is the discrete cosine transform. This yields a frame of width $K = 24$. From here on in this section we will refer to X_{dct} for frame n as X_n. We assume that each X_n fits some gaussian distribution; the set of size M of these distributions forms the gaussian mixture model (GMM) for our VAD. We use naive k-means to initialize each model; we then train the model using maximum likelihood estimation (TODO: elaborate on this.).

1.1 Markov-based signal classification

The output of our GMM over time is g_n, a sequence of frames of width M which give the likelihood that frame n fits each model in the set. For each signal type (i.e. noise, voice, impulse) we can expect g to follow some pattern or set of patterns. To model this, we use a markov model for each signal type. The benefit of this is that markov models are simple to implement and understand; the drawback is that patterns which involve more than one step in time are not accounted for. This model assumes knowledge of just the previous frame is sufficient to classify the signal.

1.1.1 Markov model training

For each signal type we have an M, M matrix, denoted as π_M which tracks transitions between each of the gaussian models. To train the model, we simply run each frame through the GMM and get its best guess m_n as to the value of m for that frame. Then, we increment $\pi_M(m_n, m_{n-1})$ by 1. This continues for every frame in the signal.

1.1.2 Markov model application

If we have P different signal types with a transition matrix for each signal type,

2 Denoiser

Let y be some noisy signal, d be background noise, and s be the desired signal; let Y, D, and S be the Fourier transforms of these signals:

$$y[n] = s[n] + d[n] \tag{2}$$
$$Y = S + D \tag{3}$$

We assume that $|D|$ will take on some sort of average value over time and that the total power of D is more or less constant wrt time. This means that we can approximate $|S|$ with:

$$|\hat{S}| = |Y| - |D| \tag{4}$$
$$\arg \hat{S} \sim \arg Y \tag{5}$$
$$\Rightarrow \hat{S} = (|Y| - |D|) \arg Y \tag{6}$$

We estimate $|D|$ by deciding that some frames are pure noise, and thus their spectral magnitudes represent $|D|$ well. We average the magnitudes of some number of these frames and let this be $|\hat{D}|$. In reality $|D|$ will change over time: as an example, if $|D|$ is dominated by noise from a laptop fan, then the characteristics of $|D|$ will change when the fan speeds up or slows down. Thus $|\hat{D}|$ needs to be resampled every now and then with the following:

$$|\hat{D}| := (1 - \alpha_D)|\hat{D}| + \alpha_D |D_n| \tag{7}$$

where $\alpha_D \in \{0, 1\}$ is the resample factor and n refers to the frame being sampled. Both the total power in $|D_n|$ and the decision of the VAD for frame n may influence the decision to resample. In testing, it was found that resampling based purely on total power caused a large amount of error in $\hat{S}$ to grow over time; this began to influence the VAD's output, so it could not distinguish between white noise and impulse noise. Thus feedback from the VAD is needed to ensure accurate noise resampling.