

## Project Report on Modified Spectral Subtraction

For ECE 9305. Applied Probability Theory I

Submitted by: Manan sheel. Student ID: 250938184

### Theory

I have implemented the modification of Spectral Subtraction method for speech enhancement as given in [1] which is based on spectral over-subtraction using the parameter 'alpha' and also introducing a spectral floor, which is a fraction, 'beta' of the estimated noise spectrum. This modification of the original method helps in reducing the 'musical noise' which is a noise introduced after Spectral Subtraction due to the peaks and troughs present in the additive white noise. (we use an additive white noise as the noise for our purpose). Sometimes, performing over-subtraction introduces 'broadband noise' in the enhanced signal. This happens due to the valleys surrounding the peaks in the noise. 'alpha' is a positive value greater than 1 which is based on the SNR (Signal-to-noise ratio) between the signal and noise and it also depends on the value of alpha at 0 dB.

The method adapts to a wide range of signal-to-noise ratios, if a reasonable estimate of the noise spectrum can be obtained. The technique can be described using equation below:

$$\begin{aligned} \text{let } D(w) &= P_s(w) - \alpha P_n(w) \\ P'_s(w) &= \begin{cases} D(w), & \text{if } D(w) > \beta P_n(w) \\ \beta P_n(w), & \text{otherwise} \end{cases} \\ \text{with } \alpha &\geq 1, \text{ and } 0 < \beta \ll 1 \end{aligned}$$

Here  $P'_s(w)$  is the enhanced speech spectrum and  $P_n(w)$  is the estimated noise spectrum obtained during non-speech or 'silent' activity.  $\alpha$  is the over subtraction factor and  $\beta$  is the spectral floor parameter. Parameter  $\beta$  controls the amount of residual broadband noise and the amount of perceived Musical noise. If  $\beta$  is too small, the musical noise will become audible, but the broadband noise will be reduced. If  $\beta$  is too large, then the residual noise will be audible, but the musical noise will be gone. Parameter  $\alpha$  affects the amount of speech intelligibility. If  $\alpha$  is large, the enhanced signal intelligibility will suffer. If  $\alpha$  is too small noise remains in enhanced speech signal. When  $\alpha > 1$ , the subtraction can remove the broadband noise by eliminating most of wide peaks. But the deep valleys surrounding the peaks remain in the spectrum. These valleys between peaks are no longer deep when  $\beta > 0$  compared to when  $\beta = 0$ . We should choose 'alpha' and 'beta' such that we get best performance.

$\alpha$  varies in each frame based on the segmental SNR value and the chosen value of  $\alpha$  at 0 dB.

$\gamma$  parameter creates more spectral change. It is the exponent of the power spectrum and modifies the above equations like this:

$$\text{let } D(\omega) = G[P_s^Y(\omega) - \alpha P_n^Y(\omega)]$$

$$P_s'(\omega) = \begin{cases} D^{1/Y}(\omega), & \text{if } D^{1/Y}(\omega) > \beta P_n(\omega) \\ \beta P_n(\omega), & \text{otherwise} \end{cases}$$

with  $\alpha \geq 1$ , and  $0 < \beta \ll 1$

Here, G is the normalization factor which improves the level of speech in the enhanced signal.

## Implementation

My MATLAB code performs spectral subtraction for speech enhancement as given in [1]. It takes as input the name of the noisy file, the name of the enhanced file, the value of factor alpha at 0 dB as desired (given as alpha\_o) which is either 3 or 4, the value of beta (the spectral floor) which is positive but  $\beta \ll 1$ . It also takes gamma (exponent of power spectrum) as input, thus allowing user to select gamma which is less than 1 and can create more spectral change. Also, it takes the normalization factor as input to improve level of processed signal. By changing values of alpha and beta, we can attenuate musical noise and broadband noise as required while retaining speech intelligibility. These are the steps in my code for implementing the algorithm:

1. We call the function that performs modified Spectral Subtraction which takes the arguments, given in the paragraph above as inputs.
2. Now, we will move into this function.
3. First, it reads the input audio file using 'audioread' command and applies overlapping frames and 'hamming/tukey' windows to the data. First, we divide the signal into overlapping frames using 'buffer' command that takes as input the signal data, the frame size in samples, and the overlap window size in samples.
4. Then, we find the power spectrum of the noisy signal using 'fft' command and squaring the result. We also save the phase of the signal which will be useful while resynthesizing the signal. And we estimate noise spectrum from the first five frames (which may be taken as 'silent' frames).
5. Then, we calculate value of the subtraction parameter 'alpha'. We do this by getting the Signal-to-noise ratio and from our selected value of alpha at 0 dB. We call either one of the 2 functions (based on alpha\_o) that perform the calculation of 'alpha' for us.
6. Then, we perform the modified spectral subtraction by subtracting alpha times the noise estimate from noisy power spectrum. Also, here, we consider the exponent of power spectrum. We apply it after performing subtraction. Also, we apply normalization factor ( $>1$ ) to improve level of processed signal. Then, we apply the spectral floor by not allowing the spectral components (after subtraction) from going below it.
7. After that, we resynthesize signal from the frames by using saved phase of the signal and applying ifft. Then we deframe the result using the function 'deframe\_sig' (taken from internet)
8. Then, at last, we get our enhanced signal.

9. Now, we play both the noisy signal and the enhanced signal (with 5 seconds gap in between) to see the effects of modified spectral subtraction and save the played enhanced signal to a file.
10. Now, we can play around with different values of  $\alpha_o$ ,  $\beta$ ,  $\gamma$ , normalization factor to see its effects on the enhanced signal.

## Results

After running the code, I found that the noise is greatly reduced in the enhanced signal as compared to the noisy signal. The enhanced signal is much cleaner, and the intelligibility is also preserved. I found best results for 50% overlap, i.e. when the overlap window was 50% of the frame size. The best frame size was found to be 25 ms. ' $\alpha_o$ ' was taken either 3 or 4. The best ' $\gamma$ ' was from 0.5 to 1. And I took ' $\gamma$ ' as 5. ' $\beta$ ' was good anywhere between 0.001 to 0.02.

## References:

[1] Berouti, M., Schwartz, M., and Makhoul, J. (1979). Enhancement of speech corrupted by acoustic noise. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, 208-211.

Resources used for writing code:

- [1] Loizou, P.C., 2013. Speech enhancement: theory and practice. CRC press.
- [2] Function `deframe_sig` taken from this source: [https://raw.githubusercontent.com/jameslyons/spl\\_featgen/master/deframe\\_sig.m](https://raw.githubusercontent.com/jameslyons/spl_featgen/master/deframe_sig.m)
- [3] Help taken from this source: [https://gist.github.com/jameslyons/554325efb2c05da15b31#file-spectral\\_subtraction\\_demo-m](https://gist.github.com/jameslyons/554325efb2c05da15b31#file-spectral_subtraction_demo-m)