CS229 == Final Project Report

SPEECH & NOISE SEPARATION

Ceyhun Baris Akcay
Stanford University
Department of Electrical Engineering
Stanford/CA
cbakcay@stanford.edu

***Abstract*—In this course project I investigated machine learning approaches on separating speech signals from background noise.**

***Keywords*—MFCC, SVM, noise separation, source separation, spectrogram**

I. INTRODUCTION

Speech processing is a highly popular research subject. As it is a common problem in all signal processing tasks, speech processing is also adversely affected by noise in the environment.

One of the applications that is highly susceptible to noise is indubitably speech recognition. Intuitively thinking about it, the task is to identify words from a speech signal. If this system is going to be useful it has to be invariant to different speakers (different pitch, timbre, etc.) so for example if there are multiple people speaking (i.e. the noise is a superposition of other speech signals) this should obviously deteriorate performance. This in fact is the truth as described in [1].

There is a great deal of literature on this topic. Various methods are used, such as blind source separation [3], utilizing neural networks [4], using probabilistic models [2], using multiple microphones to utilize the spatial difference of the sources, etc. Here I will focus on machine learning approaches.

Speech signals are quasi-periodic signals (whose periods are called pitch) [1]. This is largely due to our vocal cords generating a periodic sound wave which is modulated in our mouth, speech formation can be considered as a periodic signal (the periodic sound waves) being passed through an LTI filter (the mouth) [1]. This nature of speech signals give them unique properties that can be exploited for detecting, isolating even separating them. The fact that different people's voices have different pitch is one major aspect used in separation; in fact such harmonic characteristics of the signal can be very easily seen in a spectrogram [1] an example is in figure 1.

The rest of the report will first present some of the literature on the topic. Followed by a short summary of speech signals that is relevant to the project and finally my method and results will be presented.

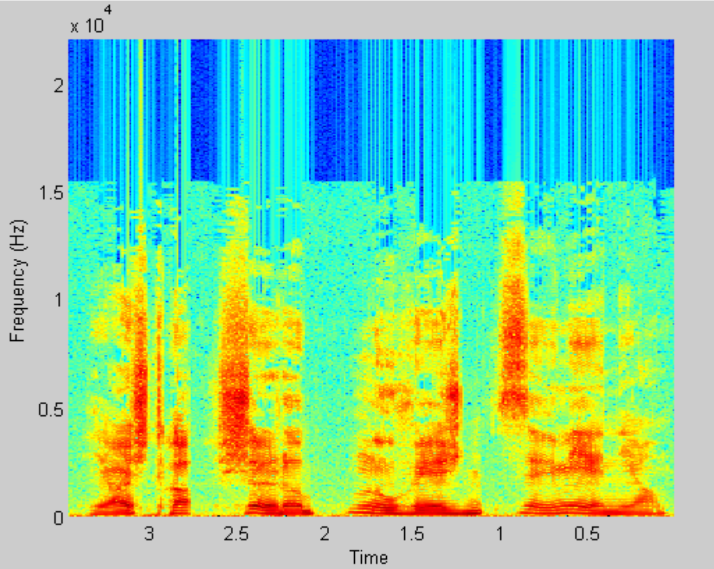

Figure 1: An example spectrogram of me saying out my name. The red parts are the dominant frequencies

II. LITERATURE REVIEW

A. *Spectral Learning for Blind Seperation*

Bach and Jordan, in [3] investigate a method for blind speech separation using a single microphone. Their idea is to segment the spectrogram of the given signal into multiple disjoint segments such that each segment will correspond to the speech of one speaker.

Firstly they construct feature vectors for speech signals that can be used for separation. They use non-harmonic cues such as continuity and common fate, as well as harmonic cues such as pitch and timbre. For timbre they use the spectral envelope and for the pitch cues they have developed a novel algorithm that can identify multiple pitches in a windowed portion of a speech signal and their corresponding dominance within the window (roughly corresponding to amplitude of the pitch.) Obviously for the continuity determination, entire spectrogram needs to be used, as a result the feature space used is very high dimensional. They use an efficient implementation of parameterized affinity matrices to ultimately perform the segmentation.

B. An Approximation to the EM Algorithm

In [2], Frey et. al. propose a different approach for learning noise from a single microphone recording of noisy speech.

They model the speech and noise parts of the signal as independent signals consisting of a mixture of Gaussian distributions (GMM). By using a Mel frequency scale [book] and assuming that the noise and speech have little phase correlation, they approximate the magnitude squared of the signal to be $|Y(f)|^2 \approx |S(f)|^2 + |N(f)|^2$ where 'S' denotes speech and 'N' denotes noise.

They use ALGONQUIN to obtain the posterior probability distribution using a vector Taylor series approximation. They use this approximated probabilistic inference method as the E-step in the EM algorithm (I will not go into every detail here). Finally by approximating a lower bound on the data (and refining it) They do the M-step of the EM algorithm.

C. Neural Networks

Although there are interesting techniques in the literature using neural networks as well. I found that these techniques are not very familiar to me and the aforementioned ideas were more interesting given that we are learning the basic theories behind them in class.

III. Preliminaries

Before I begin with my results I would like to give a brief overview of the speech concepts considered

A. The Logarithmic Frequency Phenomenon

The human auditory system is not uniform in frequencies; the perception of one frequency is usually very different than another (less or more sensitive) and usually is completely insensitive to any frequency below 20 Hz or above 20 KHz [1]. It turns out that the human sensitivity to sound is an approximately logarithmic function of the frequency as well as the amplitude (we are not interested in the amplitude in this project), this is why it is a commonly employed practice to represent the frequency logarithmically [1]. The Mel frequency scale is one such logarithmic frequency scale commonly used in a variety of applications from estimating the vocal tract frequency response to pitch determination [1].

B. Cepstrum and the MFCC

Cepstrum or cepstral analysis is a simple transformation of the frequency spectrum of a signal such that multiplications in the frequency domain (which are convolutions in the time domain) become summations in the quefrency cepstrum, which basically means the logarithm of the frequency spectrum.

MFCC (Mel Frequency Cepstral Coefficients) is especially useful in determining characteristics of a speech signal (e.g. pitch can be very easily determined from it). MFCCs are calculated by first warping the magnitude spectrum of the signal to the Mel scale (which is done by a Mel filter bank) followed by taking the logarithm of the warped magnitude spectrum and ultimately followed by a DCT [1].

The resulting coefficients exhibit great difference between speech and non-speech. For speech signals it has a certain shape in the low quefrencies (corresponding to the vocal tract filter characteristic) and a few impulse like features in the higher quefrencies which correspond to the pitch in the speech.

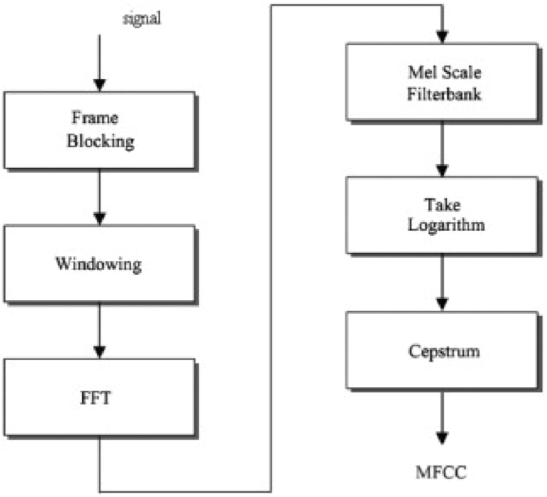

Figure 2: The MFCC pipeline (Figure taken from [5])

IV. The System

A. Identifying parts in the Signal that Contain Speech

The first step is to determine which frames of the signal contain speech. As described above the MFCCs for speech and non-speech signals differ in nature. What I did was to take 10 different speech signals from split them into 25 ms windows and to label these portions of the signal as speech or non-speech. Obviously such a short duration audio signal is not enough for humans to perceive it as speech or non-speech even less when it is not a rectangular but a Hann window. So I labeled these windows as speech or non-speech by looking at the histogram.

After the labeling, using the liblinear package for MATLAB I trained an SVM classifier on them. Ultimately this classifier was used on the speech windows for classification as speech and non-speech. In total there was 512 features for the SVM. Note that even though 25 ms equals to 400 samples at 16 KHz; for efficient DFT computation I preferred a 512-point DFT of the 400-point window.

This classifier achieves approximately 12% testing error and 8% training error. The main cause for the error is my coarse labeling of the training data, unfortunately looking at the histogram it cannot be done much finer. A second cause is speech is not always periodic pulses modulated in the vocal tract; for example "sss" of "shh" kind of sounds are more like white noise passing through the vocal tract so do not have a very distinct MFCC signature as do the voiced syllables [1]. And a second source of error is some windows capture both silence and speech together. Overall these effects caused a more than desired error rate but, still not unreasonably high. So

I decided to continue with this method instead of trying more sophisticated pitch tracking kind of algorithms for this classification.

B. The Denoising Process

For the denoising procedure I preferred to use the approximate EM algorithm described above. Now I will provide some further details on this algorithm called ALGONQUIN [2].

First of all the energy spectrum of the signal plays an important role. Their approximation which follows from the assumption that the noise and speech are uncorrelated is given below:

$$|Y(f)|^2 \approx |S(f)|^2 + |N(f)|^2 \tag{1}$$

Let $\boldsymbol{y}$, $\boldsymbol{s}$ and $\boldsymbol{n}$ be the vector of logarithms of each of the terms (for every value of f) in equation (1). Then we can write:

$$e^y \approx e^s + e^n = e^s(1 + e^{n-s})$$

The multiplication is element-wise. Taking the logarithm of the above expression we end up with the function below:

$$y = g\left(\begin{bmatrix} s \\ n \end{bmatrix}\right) = s + ln(1 + e^{n-s}) \tag{2}$$

assuming the errors in equation (2) to have a Guassian distribution, the observation probability is given by the expression below:

$$p(y \mid s, n) = \boldsymbol{\mathcal{N}}\left(\boldsymbol{y}\,; \boldsymbol{g}\left(\begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{n} \end{bmatrix}\right), \boldsymbol{\Psi}\right) \tag{3}$$

where the $\boldsymbol{\Psi}$ is assumed to be a diagonal covariance matrix for ease of implementation and simplicity.

The prior distributions of the speech and noise are assumed to be independent mixture of Gaussians, which makes the prior probabilities of both as below:

$$p(s) = \textstyle\sum_{c^s} p(c^s)p(s|c^s), \quad p(c^s) = \pi^s_{c^s},$$
$$p(s\,|c^s) = \boldsymbol{\mathcal{N}}(s;\ \mu^s_{c^s}, \Sigma^s_{c^s})$$
$$p(n) = \textstyle\sum_{c^n} p(c^n)p(s|c^n), \quad p(c^n) = \pi^n_{c^n},$$
$$p(s\,|c^n) = \boldsymbol{\mathcal{N}}(s;\ \mu^n_{c^n}, \Sigma^n_{c^n})$$

where $\boldsymbol{c^n}$ and $\boldsymbol{c^s}$ are the noise and clean speech classes respectively and the covariance matrices are both assumed to be diagonal. Combining the priors their independence and equation (3) for the conditional probabilities they arrive at the joint distribution below:

$$p(y, s, c^s, n, c^n) =$$
$$\boldsymbol{\mathcal{N}}\left(y; g\left(\begin{bmatrix} s \\ n \end{bmatrix}\right), \boldsymbol{\Psi}\right) \pi^s_{c^s}\boldsymbol{\mathcal{N}}(s;\ \mu^s_{c^s}, \Sigma^s_{c^s})\pi^n_{c^n}\boldsymbol{\mathcal{N}}(s;\ \mu^n_{c^n}, \Sigma^n_{c^n}) \tag{4}$$

As we can see with this joint distribution $p(s, n \mid c^s, c^n, y)$ is not a mixture of Gaussians and hence to use an EM algorithm needs some further manipulation or simplification of the distribution.

This is done by approximating the posterior probability for the parameters of the current noisy speech windows as below:

$$p(s, c^s, n, c^n|y) \approx q(s, c^s, n, c^n) =$$
$$\boldsymbol{\mathcal{N}}\left(\begin{bmatrix} s \\ s \end{bmatrix}; \begin{bmatrix} \eta^s_{c^s c^n} \\ \eta^n_{c^s c^n} \end{bmatrix}, \begin{bmatrix} \Phi^{ss}_{c^s c^n} & \Phi^{sn}_{c^s c^n} \\ \Phi^{sn}_{c^s c^n} & \Phi^{nn}_{c^s c^n} \end{bmatrix}\right) \tag{5}$$

where $\eta^s_{c^s c^n}$ and $\eta^n_{c^s c^n}$ are the approximated means of the speech and noise for the aforementioned classes. The Φ's are the covariances between the speech and noise according to their superscript and all of them are diagonal; this follows from the assumed priors.

So the approximated posterior for one window is given as below:

$$q(s, n, c^s, c^n) = q(s, n \mid c^s, c^n)q(c^s, c^n)$$

where $q(c^s, c^n)$ is the mixing proportion for the speech and noise classes.

So the goal is to minimize the relative entropy between the p and q distributions aforementioned. Since the relative entropy is in effect a measure of the difference of two probability distributions. Also note that ***ln[p(y)]*** does not depend on any of the variational parameters defined before and minimizing the relative entropy (RL) corresponds to maximizing ***ln[p(y)] – RL*** which is a lower bound on the log-likelihood.

As a result this method called variational inference can be used to approximate an E step in the well-known EM algorithm [6].

Also note that the relative entropy introduces non-Gaussianness into assumed Gaussian distributions. This is prevented by using a second order taylor series approximation to the relative entropy computation. The full update equations are pretty lengthy and won't fit into a double column formatted paper in a readable manner so interested reader is encouraged to check out [2] for the full equations as well as certain derivational details that I omitted here for brevity.

V. Results & Discussion

My data was taken from the PASCAL ChiMe challenge. I corrupted the data by additive white Gaussian noise with varying SNR values. The overall system was able to achieve between 6 dB to 10 dB improvement in the SNR (compared against the pure speech) the improvement does not depend much on the input SNR. Different speech signals with same SNR sometimes produced 5.98 dB and 10.13 dB. The one thing that has to be really worked on is a higher accuracy speech and non-speech classifier, because this misclassification ends up pretty audible at times (you here sudden onset of high white noise). I am not very satisfied with the results, I thought ~20dB would have been possible. In [2] they do not provide a SNR analysis. They just provide the improvement achieved in a sample speech recognition task.

VI. Conclusion

This project firstly gave me the chance to explore a very interesting field in signal processing which is speech processing. Such an extensive literature survey provided a great depth in usages of machine learning as a noise cancellation technique. Also to implement an EM algorithm for the final system gave the opportunity to solidify my understanding of the EM method. All in all I have benefited from the project greatly.

REFERENCES

[1] X. Huang, A. Acero, H-W. Hon, Spoken Language Processing, Prentice Hall, 2001.

[2] B. Frey, T. T. KristJansson, L. Deng and A. Acero, "ALGONQUIN – Learning Dynamic Noise Models from Noisy Speech for Robust Speech Recognition", *Advances in Neural Information Processing System*, 2002, pp. 1165-1172.

[3] F.R. Bach and M. I. Jordan, "Blind One-Microphone Speech Separation: A Spectral Learning Approach", *FRBMI*, 2005, pp. 65-73.

[4] J. Tchorz, M. Kleinschmidt and B. Kollmeier, "Noise Suppression Based on Neurophysiologically-Motivated SNR Estimation for Robust Speech Recognition", *Advances in Neural Information Processing Systems*, 2001 pp. 821-828.

[5] http://www.sciencedirect.com/science/article/pii/S0955598611000847, last visited on 12/14/2012.

[6] R. M.Neal and G. E. Hinton, "A View of the EM Algorithm that Justifies Incremental Sparse and Other Variants", *Learning in Graphical Models*, pp. 355-368, Kluwer Academic Publishers, Norwell MA, 1998.