Above the Din: Identification of Human Voice in Noisy Signals

A. Fried[1], D. Stonestrom[2], M. Stauber[2]
[1]*Department of Physics, Stanford University, Stanford, CA 94305-3030, USA*
[2]*Department of Mechanical Engineering, Stanford University, Stanford, CA 94305-3030, USA*

Abstract: A large amount of machine learning research has been dedicated to understanding human speech, but the ability to identify human speech in the first place can be useful as well. In this paper, an algorithm is described that can identify the presence of a human voice in an audio signal. An investigation into the most relevant features for this process is included in the description. The resultant algorithm applies a Support Vector Machine to the audio features to classify the signal with 90.5% accuracy. The algorithm then filters the classifications to successfully classify the entire audio file correctly.

1 Introduction

The human auditory system is particularly attuned to differentiating human speech from other sounds. The goal of this project is to write an algorithm that can perform live detection of human speech even when partially masked by environmental noise. One envisioned application for such a method is in disaster situations that span a large area or an area that human rescuers cannot enter, robotic "listeners" can be deployed instead and can report if they pick up the voice of survivors in their area. Another application would be for mobile phones that can change their behavior (for example if they ring or vibrate) depending on if they detect the user to be in the middle of a conversation or in a lecture.

In the past, machine-learning research has been focused on "speech recognition", defined as the ability to interpret human speech. Ironically, not as much attention has been paid to the recognition of speech itself versus other forms of noise. For classic speech-recognition there are a variety of features that are used to remove differences between an individual's tone, accent, and other vocal traits. These features effectively produce a signature of human voice regardless of the speaker. The approach of this paper is to utilize a combination of the features developed for classic speech recognition algorithms for the more fundamental identification of the presence of speech problem.

2 Dataset

For this project we compiled a unique dataset. We used single voice, single source recordings of people reading poetry found on www.LibriVox.org as our human voice training and testing data [1]. For our noise data we selected an array of noise recordings found in everyday environments from www.SoundJay.com. The types of environmental noise include industrial noise, nature sounds, and household noises. In total, the dataset contained about 4 hours of recorded sounds, approximately split between human and environmental recordings.

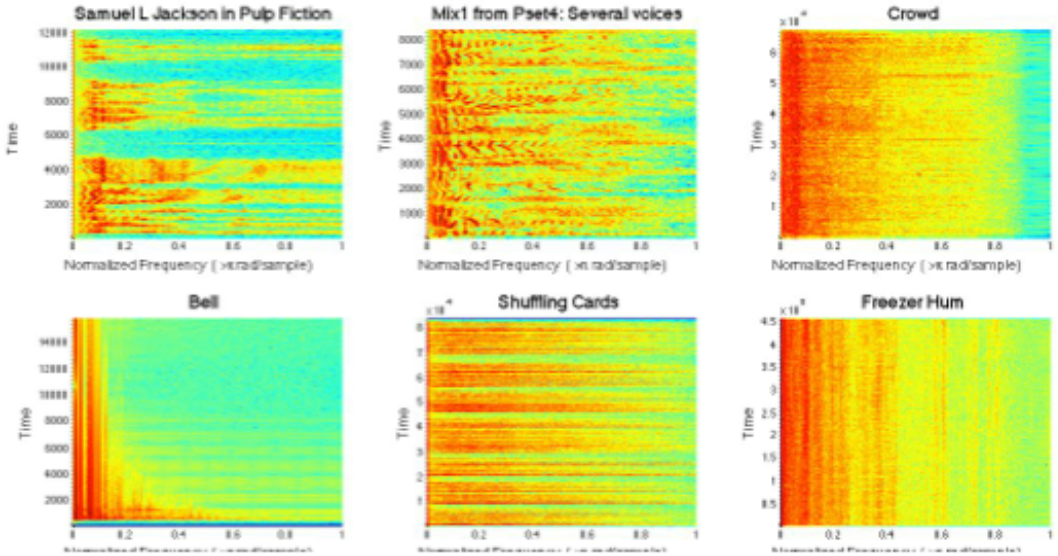

Figure 1: Sample short-time power spectra for different audio signals. There is a clear difference between the spectra of a single human voice and that of Freezer Hum

3 Methods

The development of the algorithm progressed on two main fronts: the features particular to human speech and a classification algorithm that can identify the audio stream in real time with reasonably high accuracy. The dataset audio files were divided up between those that contained a single human voice and those that contained environmental noise. A MATLAB script was created to automatically process the .wav files, record the files' a-priori classification, and extract the features.

3.1 Features

The algorithm employs two classes of spectrum-derived features of the audio file. To extract these two sets of features, we used algorithms available in Reference 2. The first class of features is compiled by comparing the short-time spectra after compressing them in a manner that re-bins and rescales the spectrogram to better mimic how humans perceive sound. The Mel-scale is one such scale; integrating the power spectrum against the bank of filters generates these features. The filters used in the Mel-scale are illustrated below and were designed so that they are perceived as "equally-spaced" in frequency based on human testing. Taking the inverse Fourier transform after this mapping picks out harmonics of voiced signals and generates the Mel Frequency Cepstral Coefficients (MFCC). As noticeable in Figure 1, the covarying harmonics clearly are a unique feature of the spectra. In taking the Fourier transform of the power spectra in this fashion it is hoped that the fundamental frequency of voice box-or sound source might be identified and used as a feature. However, this technique clearly cannot capture all harmonics, as only those that are equally spaced in the Mel scale will be detected, and most sound sources only follow that pattern approximately. Furthermore, harmonics are not always resolvable within the spectrograms.

Voice signals are temporally correlated over short times. Voice recognition algorithms tend to approach the problem by using the "Source-Filter Model" This assumes that voiced sound is comprised of a signal that contains the content of the speech, but then it is filtered as it passes through the vocal tract of the speaker. While the source signal for any given word should be identical regardless of who is saying it, within the model, it is the filter that is unique to every speaker. These filters result in different spectrograms for different people saying the same thing. Most voice-recognition algorithms use a variety of heuristics to re-bin, filter, smooth or rescale any given spectra, so that the effects of different filters are not present in the processed signal. RASTA filtering is another common processing step that smooths noisy spectral features. Machine-learning performed on the processes signal will consequently only be sensitive to the information present in the "source," and will be able to distinguish between spoken words and other sounds.

Perceptual Linear Prediction (PLP) is one technique for extracting features from this processed signal and constitutes the second class of features we examined. In the time-domain, the linear model below is fitted to the processed signal and the 'a' coefficients are extracted.

$$x_n = \sum_{k=1} a_k x_{n-k} \quad C_j = \sum_{k=1} a_k C_{j-k}$$

The fit to the model that optimizes the squared error between x and the signal, yields the equation above for a, expressed in terms of autocorrelation functions, C, which are easily recovered from the Fourier Transform of the processed spectrum.

3.2 Classification

To classify our data we sought an algorithm that will run fast enough to allow for live classification of audio signals. We found that a support vector machine (SVM) with a linear kernel achieved high accuracy as well speed. Our SVM is optimized by L2 regularized norm. In addition, we apply a low pass filter to prevent the output from tracking the misclassification error.

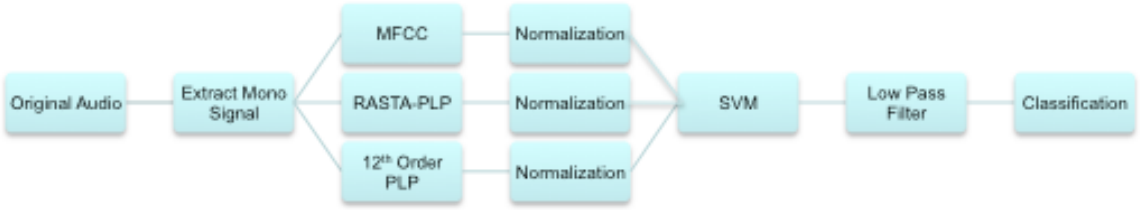

Figure 2: Diagram of Algorithm Scheme

4 Results

A program was developed in MATLAB to apply the algorithm described above to our dataset. The program allows the user to select the percentage of the audio files devoted to training the algorithm and uses the remaining audio to test the classification accuracy of the algorithm.

A

ACTUAL CLASS	PREDICTED CLASS: Noise	PREDICTED CLASS: Human
Noise	90.4%	9.6%
Human	9.8%	90.2%

B

Features	Accuracy (%)
MFCC	69.06
RASTA-PLP	71.09
PLP- 12th order	80.69
ALL	90.52

Figure 3: A. Confusion matrix of the algorithm when using all of the features of the dataset. Note that the algorithm does about as well for classifying both positive and negative data. B. Classification accuracy of the algorithm using various subsets of features. Values were taken when SVM was trained on 85% of our positive and negative data and tested on the remaining portion of data.

4.1 SVM Classification

The SVM step was quite successful at classifying the files in our dataset. As seen in Figure 3A, the percentage of false-negatives and false-positives are about the same. The different feature sets that were employed each produced different levels of accuracy when the SVM was trained off only that feature. This was likely due to the fact that each of the features is designed to pick up on a different aspect of human speech. For example, the RASTA-PLP is very good at picking up the onset of syllables and words, but the MFCC is about the same

across a whole syllable. The combination of these features was able to produce a much higher accuracy than their individual applications [Figure 3B].

4.2 Filtering

The high accuracy of the SVM was over the entire file, but there is still an error of ~10% that causes the classification of a live audio signal to appear choppy. These misclassifications happen most often by the natural breaks in human speech, between syllables and words. By applying a filter, the SVM removes these brief misclassifications and achieves a much higher accuracy for the incoming audio stream as seen in Figure 4.

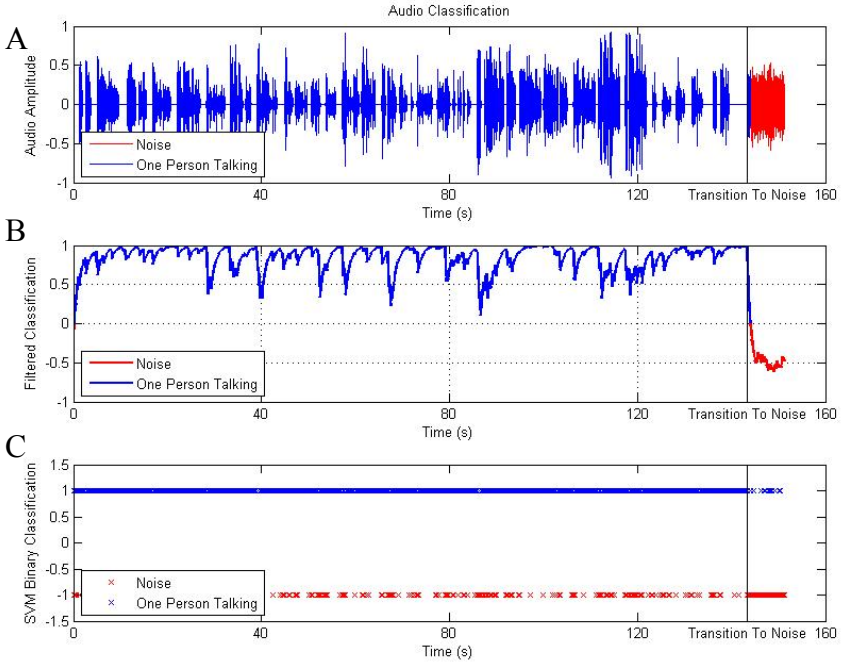

Figure 4: Classification of streaming audio signal with filtering. A. The incoming audio signal starts with a single person talking and transitions to the sound of freezer hum at ~140 seconds. B. Decision values of algorithm after filtering. C. Raw SVM classifications before filtering.

4.3 Classification in Noisy Signals

After training the algorithm on clean recordings of human speakers and environmental sounds, the algorithm was tested on a mixture of these signals. To accomplish this, the volume of a particular environmental noise was held constant and the volume of a speaker was adjusted from zero to one hundred percent of their full volume. When the speaker is at one hundred percent, the two audio signals are about the same volume. The algorithm's accuracy at identifying the speaker's presence was recorded [Figure 5].

When the speaker represents over 20% of the sound in the audio file, the algorithm classifies a majority of the signal as containing a human voice. This result suggests that the algorithm should prove useful to identify human voices in the noisy environments of disaster situations or our everyday lives.

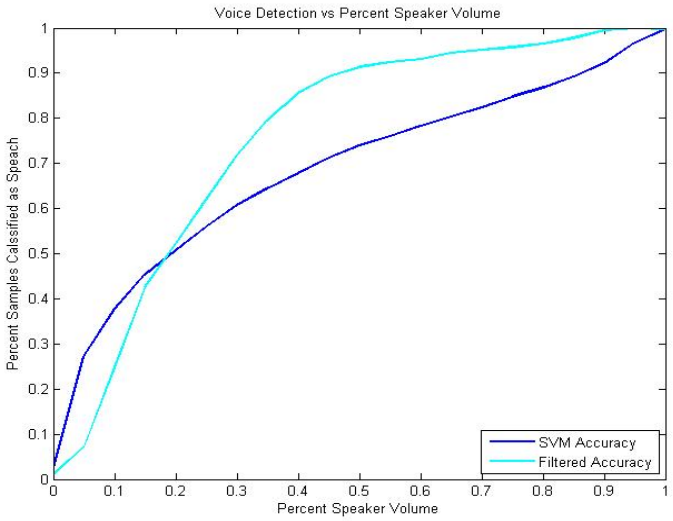

Figure 5: Example relationship between the amount of environmental noise in an audio file and the ability to identify human voice. When the voice is at ~20% the algorithm successfully picks up the voice as the dominant classification.

5 Conclusion and Future Work

This paper presented a strong start to the development of an algorithm to identify human voice in noise environments. The algorithm presented here is currently fast enough to process an incoming signal, but it may prove useful to develop a less computationally intensive set of features. In addition there may be certain sources of environmental noise that were not considered but could be sufficiently similar to the human voice to cause misclassification such as animal noises. This possibility can be tested by compiling an even larger dataset of environmental noise for training and testing.

Another possible extension of the work we have done is counting the number of unique speakers in an audio sample. The features we used, in particular the PLP, are explicitly designed to remove essential differences of the signals of two people saying the same thing. It is not surprising, that when training and testing the SVM off sets of multiple people talking and one person talking, it has a hard time distinguishing between the two categories. Likewise, the SVM cannot distinguish between a crowd of people and a single speaker. Simple tests of nonlinear SVMs have not been effective, though further exploration might prove fruitful. This suggests that the features we chose are not suited for this application. Finer details of the spectrogram might be used instead to count the number of people talking. In particular the correlations between the time-dependence of the harmonics of the voice spectrograms might be a particularly useful set of features.

6 Acknowledgements

Thank you to David Held and Andrew Maas for assisting us during the development of our project. Thank you to Professor Andrew Ng and the wonderful teaching staff for the course. Thanks to our fellow classmates and Piazza contributors!

7 References

[1] Abercrombie, Lascelles. “Short Poetry Collection 091”. LibriVox. http://librivox.org
[2] Ellis, Daniel P. W. “PLP RASTA and MFCC Inversion in MATLAB. http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/
[3] Fan et al LIBLINEAR: A library for large linear classification Journal of Machine Learning Research 9(2008), 1871-1874.
[4] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech", J. Acoust. Soc. Am., vol. 87, no. 4, pp. 1738-1752, Apr. 1990.
[5] H. Hermansky and N. Morgan, "RASTA processing of speech", IEEE Trans. on Speech and Audio Proc., vol. 2, no. 4, pp. 578-589, Oct. 1994.