Analysis and Hardware Implementation of Voice Activity Detection Algorithm

Varsha Pendyala
ee13b1022@iith.ac.in

Sumanth Sadhula
ee13b1026@iith.ac.in

***Abstract*— Voice activity detection in the presence of transient noise is a challenging problem since the spectrum of transients changes quickly unlike that of stationary noise. Hence the speech and transients may often appear similar when represented by MFCCs in Euclidean space. In this paper, we use a metric that effectively distinguishes speech from transients through modeling the signal in terms of their latent generating variables. We use this metric to compute the similarity matrix for spectral clustering that divides given frames into two clusters.(i.e., speech presence and speech absence frames). The eigen vectors of the normalized Laplacian of the similarity matrix and the GMM are utilized to compute the likelihood ratio for voice activity detection. In this project, we also implement a simpler algorithm that does online detection of the silent frames on Rasberry Pi.**

I. Introduction

VAD is usually a preprocessing step for several of the speech processing applications. The main uses of VAD are in speech recognition and speech coding. Handling this task of VAD in noisy environments is not straightforward since the statistics of the noise are unknown beforehand. In the presence of stationary noise alone, the spectrum of the signal in short time intervals could be used to measure the spectral flatness in order to decide on the voice activity. The detailed explanation of this approach is discussed in section III. It works well since the spectrum of stationary noise remains flat over a range of frequencies. However, this approach cannot help when the noise spectrum changes quickly, like in the case of transient noises which are abrupt interferences, such as keyboard taps, knocks. Since the spectrum of transients varies in time even quicker than the spectrum of speech, transients could be wrongly detected as speech based on the above approach.

Speech and Transients may often appear similar when represented by MFCCs in the Euclidean space. In this paper, we have used Mahalanobis distance metric to measure the affinity between the feature vectors. When we assume that speech and transients are driven by two independent sets of latent variables controlling their generation, there exists a non-linear mapping between the observable signal and the vector of generating variables. The generation of many signals can be associated with small set of physical constraints controlling their production. For example, the generation of speech is controlled by the position of vocal tract and the movement of lips, tongue and jaw. It is shown in [1] that Mahalanobis distance between the observable signals approximates the Euclidean distance between the underlying generating variables. This metric is used in constructing the similarity matrix for Spectral Clustering, to cluster the MFCCs of the input frames into speech and non-speech clusters. The clustering problem can be done using GMMS. However, fitting a GMM over high dimensional data requires great amount of training data. Also, likelihood ratio obtained with GMMs fitted directly over MFCCs is not reliable, since MFCCs of speech and transients are not effectively distinguished in Euclidean space. The main idea of Spectral clustering method is to use the dominant eigen vectors of the normalized Laplacian of similarity matrix as the new lower dimensional representation of the data.

The implemented algorithm is a supervised learning algorithm. Training is used to estimate the parameters of the GMMs that model the first three dominant eigen vectors of the normalized Laplacian matrix. During testing, likelihood ratio of the new frame w.r.t two GMMs is found. With appropriately chosen threshold, the label for the frame is assigned.

II. Speech Vs Transients

Every clustering problem has three main stages: feature space selection, choosing a metric as notion of similarity between the data points and selecting a clustering algorithm.

We have used 13 dimensional MFCCs as feature vectors. Distance between frames of signal is measured using modified Mahalanobis distance which is given by:

$$||y_n - y_m||_M^2 \triangleq \frac{1}{2}(y_n - y_m)^T (C_n + C_m)(y_n - y_m) \qquad (1)$$

where L is the number of dimensions of feature vectors and C_n and $C_m \in R^{LXL}$ are the covariance matrices of y_n and y_m respectively. The clustering algorithm has a training and testing stage which are described in what follows.

A. Training stage

We have used Timit database for clean speech signals. Transient noises are taken from [2]. Transient noise is added to clean speech such that there could be 3 different kinds of frames in signal sequence: only speech, only transients, both speech and transients. Transient noises are normalized to have the same maximal value as that of speech signal.

During training, we use two sequences s_1, s_2 as the reference signals. s_1 is the speech signal with door knock transient and s_2 is the speech signal with keyboard tap transient. Spectral clustering is performed separately on s_1 and s_2.

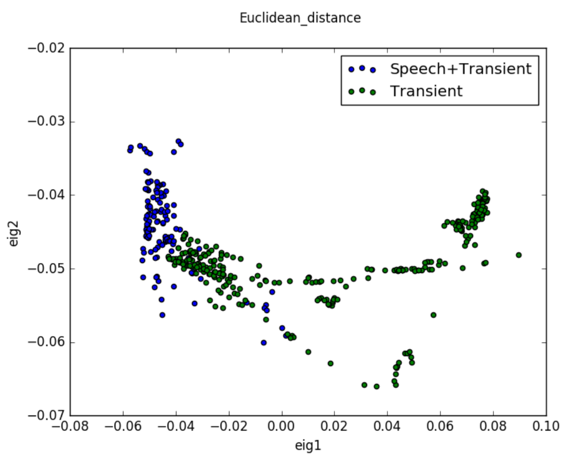

Fig. 1. Scatter plot of the first two eigen vectors, for which the speech signal (reference signal) is contaminated by door-knock transient,Kernel based on Euclidean distance

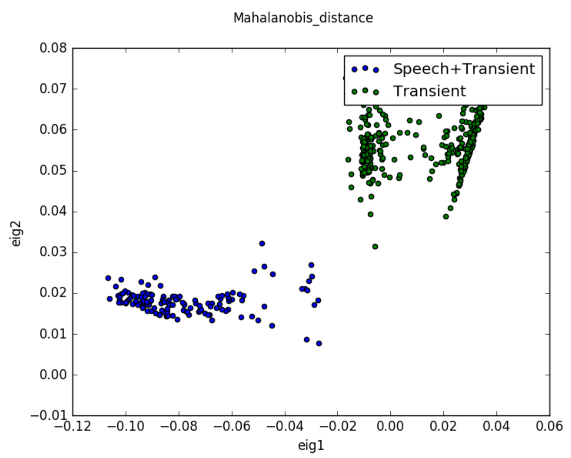

Fig. 2. Scatter plot of the first two eigen vectors, for which the speech signal (reference signal) is contaminated by door-knock transient, Kernel based on modified Mahalanobis distance

Spectral Clustering:

In spectral clustering methods similarity between every pair of data points in a l^{th} training sequence is found using a kernel that is suitably defined for the given application. Let k(y_n,y_m) be the the similarity kernel between y_n and y_m given by:

$$k(y_n,y_m) = e^{-\frac{||y_n-y_m||_M^2}{\varepsilon}} \tag{2}$$

where 1≤n,m≤ N_l (N_l is the number of frames in l^{th} training sequence) and ε is a scaling parameter. Using the kernel in (2), we construct an affinity matrix K $\in R^{N_l X N_l}$ such that its (n,m)th entry, denoted by $K_{n,m}$ is:

$$K_{n,m} = k(y_n,y_m) \tag{3}$$

The normalized Laplacian of the affinity matrix K is found as follows:

$$M = D^{-1}K, \tag{4}$$

where D $\in R^{N_l X N_l}$ is a diagonal matrix with $D_{m,m} = \Sigma_n K_{m,n}$. It implies, rows of M sums up to one. Then we apply Eigen value decomposition to M. We use the eigen vectors to form the low dimensional representation of the frames. Specifically we construct a matrix $U^l \in R^{N_l X J}$ using J< N_l eigen vectors corresponding to J largest eigen values. Because of the sign ambiguity in the computation of eigen vectors, eigen vectors of s_2 is computed such that the mean of each cluster is as close as possible to the mean of the corresponding cluster in s_1. A matrix V is constructed as follows:

$$V = [(U^{s_1})^T \quad (U^{s_2})^T]^T \tag{5}$$

where each row of V is simply the low dimensional representation of corresponding frames in the reference signals.

We use the technique of Nystrome extension with reference signals s_1, s_2 to obtain the low dimensional representation of remaining training sequences.

Nystrome extension:

For every new training sequence we find the similarity matrix between the new frames and the frames of reference signals. If an l^{th} training sequence has N_l frames, we construct a matrix B^l as follows:

$$B^l(i,j) = k(t_i,r_j) \tag{6}$$

-where t_i is the ith frame of l^{th} training sequence and r_j is the j-th frame from the set of reference frames. Once the above matrix is found, the data representation in terms of eigen vectors of the Laplacian is approximated by the following equation:

$$U^l = diag((1B^l_{knn})^{-1})(B^l_{knn})^T V \tag{7}$$

where $U^l \in N^l$X3, 1_{mxn} is mxn matrix with all ones and the ith column of the matrix B^l_{knn} is obtained by setting to zero all elements of ith coulumn of B^l except the k largest elements.The subscript knn stands for k nearest neighbours. This simply means the low dimensional representation of any new frame is the weighted mean of low dimensional representation of k-nearest neighbours of that point in the set of reference frames.

These low dimensional feature vectors are used to estimate the parameters of the two GMMs. One for speech and the other for non-speech. Speech class: only speech or speech+transient, Non-speech class: only transient.

B. Testing stage

Initially, the stationary noise in the test signal has to be attenuated using speech enhancement algorithms. Then, using the same reference signals s_1 and s_2, low dimensional representation of the test frames is found through Nystrome extension. Likelihood of each frame is calculated with respect to two GMMs that have been trained on the dominant eigen vectors in the training stage. Let H_0 and H_1 be speech absence and presence hypotheses, respectively. Let f(.;H_0) and f(.;H_1) be the probability density function of lowd feature vector corresponding to noise only frames and frames containing speech signal, respectively. The likelihood ratio for a new unlabeled frame is given by:

$$\Gamma_t = \frac{f(\phi(t,:);H_1)}{f(\phi(t,:);H_0)} \tag{8}$$

where $\phi(t,:)$ is the low dimension representation of t-th frame. The decision rule for an unlabeled t-th frame is obtained by :

$$\Gamma_t \underset{H_0}{\overset{H_1}{\gtrless}} T_h \tag{9}$$

where T_h is the threshold that controls the trade-off between probability of detection and false alarm. Increasing (decreasing) this parameter leads to a decrease (increase) of both the probability of false alarm and the probability of detection.

III. Silent Frames Detection

We use three different features per frame:

1) Short-time energy (E)
2) Spectral Flatness Measure (SFM) which is calculated using the following equation:

$$SFM_{db} = 10log_{10}(\frac{G_m}{A_m}) \tag{10}$$

where A_m and G_m are arithmatic and geometric means of speech magnitude spectrum respectively.
3) Most dominant frequency component (F).

We use three external parameters: Energy_PrimThresh, F_PrimThresh, SF_PrimThresh that finds their role in the decision rule. Their values are taken from [4]. If first 30 frames are silence, then these are used to find certain statistics of silence admidst any stationary noise. They are:

$$Min\_E = min(E) \tag{11}$$
$$Min\_F = min(F) \tag{12}$$
$$Min\_SF = min(SFM) \tag{13}$$

The above statistics are used to set decision thresholds for all three features.

- Thresh_E = Energy_PrimThresh*log(Min_E)
- Thresh_F = F_PrimThresh
- Thresh_SF = SF_PrimThresh

For every incoming frame i, set Counter=0

- If(E(i)-Min_E)$\geq$ Thresh_E, Counter++
- If(F(i)-Min_F)$\geq$ Thresh_F, Counter++
- If(SFM(i)-Min_SF)geq Thresh_SF. Counter ++

If Counter>1 the frame is marked as speech else it is marked as silence. Whenever we come across a silent frame Min_E is updated as follows thereby updating Thresh_E as previously mentioned:

$$Min\_E = \frac{(Silence\_Count * Min\_E) + E(i)}{Silence\_Count + 1} \tag{14}$$

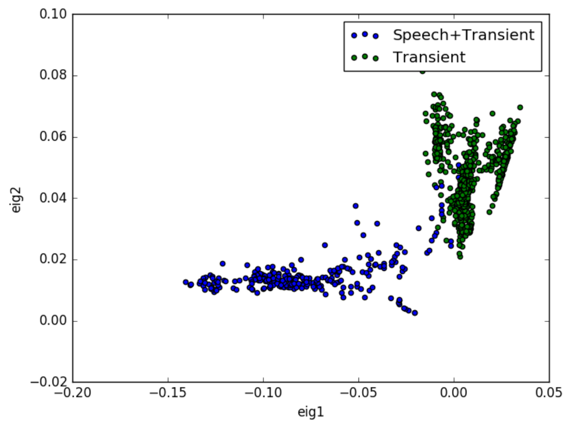

Fig. 3. Scatter plot of the first two eigen vectors of the reference signals after adjusting for the sign ambiguity

IV. Experimental Results

In the training stage we use 50 different speech utterances from Timit database. Two types of transient noise (knock, keyboard tap) is added to them. In the testing stage, we use 50 other different speech utterances corrupted with transient noise. All the signal sequences are sampled at 16kHz. Size of a frame is 32ms with 16ms overlap.The parameter k in computing the matrix B_{knn} is set to 5.

The proposed metric in (1) requires the computation of local covariance matrix of each frame. One approach for their estimation is to use sample covariance as follows:

$$\hat{C_n} = \frac{1}{R+1}\Sigma_{i=0}^{R}(y_{n+i} - \hat{\mu}_n)(y_{n+i} - \hat{\mu}_n)^T \tag{15}$$

where y_{n-R}, y_{n-R+1},... are consecutive frames at temporal neighborhood of frame y_n, and $\hat{\mu}_n = \frac{1}{R+1}\Sigma_{i=0}^{R}y_{n+i}$ is the sample mean. We set R to 15.

We see from Figures 1 and 2 that modified Mahalanobis distance metric effectively distinguishes speeach and transients.

During the training stage we had 18590 frames with both speech and transient, 36660 frames with transient alone. We have used a GMM of 10 mixtures for speech class and another GMM of 20 mixtures for non-speech class. We found that with T_h of $\frac{2}{3}$, probability of detection is around 0.7 and probability of false alarm is close to 0.065.

Implementation on Rasberry Pi:

We have performed online detection of silent frames on Rasberry Pi. As was expected, the algorithm of section III is able to differentiate silence from speech and transients to a great extent without being able to differentiate between speech and transients. We got an accuracy of around 96 percent without stationary noise.

V. Conclusions

We have been able to distinguish speech from transients by exploiting the affinity of data frames in terms of their modified Mahalanobis distance. Once we had the low dimensional representation of the frames using Spectral Clustering and Nystrome extension , GMM parameters with high reliability could be obtained using relatively fewer training data points.

REFERENCES

[1] Kernel Method for Voice Activity Detection in the Presence of Transients,David Dov, Ronen Talmon, Member, IEEE and Israel Cohen, Fellow, IEEE

[2] [Online]. Available: http://www.freesound.org

[3] Voice Activity Detection in Presence of Transient Noise Using Spectral Clustering,Saman Mousazadeh and Israel Cohen, Senior Member, IEEE

[4] A SIMPLE BUT EFFICIENT REAL-TIME VOICE ACTIVITY DETECTION ALGORITHM, M. H. Moattar and M. M. Homayounpour Laboratory for Intelligent Sound and Speech Processing (LISSP), Computer Engineering and Information Technology Dept., Amirkabir University of Technology, Tehran, Iran