

Research Article

Noise Cancellation in Presence of Transient Noise using Spectral Clustering

Ruchita Bari^{#*} and M. R. Bachute[‡]

[#]Dept. of Electronics & Telecommunication G.H. Raisoni institute of Engineering Technology Pune, India

[‡]Dept. of Electronics G.H. Raisoni Collage of engineering Pune, India

Accepted 30 Nov 2016, Available online 07 Dec 2016, Vol.6, No.6 (Dec 2016)

Abstract

The aim of this paper is to remove short term noise. VAD has attracted significant research efforts in the last two decades. In this paper, I develop a novel VAD algorithm based on spectral Clustering methods. I proposed a VAD Technique which is a managed learning algorithm. This algorithm divides the input signal into two part clusters. (i.e., speech presence and speech absence frames). I use labeled data in order to correct the parameters of the kernel used in spectral clustering method for computing the comparison matrix. The parameters obtained in the teaching phase together with the eigenvectors of the normalize Laplacian of the parallel matrix and Gaussian mixture model (GMM) are utilize to compute the likelihood ratio needed for voice activity detection. Simulation results demonstrates the improvement of the proposed method compared to conventional arithmetic model-based VAD algorithms in existence of transient noise.

Keywords: Gaussian mixture model, spectral clustering, transient noise, log like hood algorithm, voice activity detection

1. Introduction

Voice and unvoice classification in an unsolved problem in speech processing and affects divers applications including robust speech recognition discontinuous transmission, Real -Time speech communication on the internet or the combined noise reduction and echo cancellation schemes in the context telephony. Elementary methods for voice activity detection such as G.729 standard, calculate line spectral frequencies, Full-band energy, low-band energy (< 1KHZ), and zero crossing rate. Each frame is then simply classified using a fixed decision boundary in the space defined by these features. Smoothing and adaptive correction can be applied to improve the estimate. Although these methods have acceptable performance when applied to clean signals, their performance essentially degrades in noisy environments even in moderately high signal to noise ratios (SNRs). To overcome this shortcoming, several statistical model-based VAD algorithms have been proposed in the last two decades. The spectral coefficients of the noise and speech signal can be complex Gaussian random variables and developed a VAD algorithm based on the likelihood ratio test (LRT). Following their work, many researchers tried to improve the performance of model-based VAD

algorithms by assuming different statistical models for speech signals, while these methods have superior performances in presence of stationary noise over the elementary Methods, their performances degrade significantly in presence of transient noise such as coughing, Sneezing, keyboard, typing, and door knocking sounds . This means that with high probability, these sounds are detected as speech. VAD is usually a preprocessing step in speech processing. Applications such as speech or speaker recognition.

A straightforward application of VAD would be an automatic camera steering task. Suppose a scenario in which there exist multiple speakers with a camera assigned to each of them (a popular example can be videoconferencing). The camera must be steered to the dominant speaker automatically. While stationary noise can be treated very well using a statistical mode-based method, transient noise could be very annoying . This means that a silent speaker might be identified as a dominant speaker while he/she is just typing or there is a knock on the door. Hence, finding a VAD algorithm which is robust to transient noise would be of practical interest.

VAD can be regarded as an acoustic event Detection (AED) task which detects some acoustical event including transient noise, e.g., door knocking, footsteps, etc. Current most prominent works in AED reflect the aim of bringing the most successful technologies of

*Corresponding author: Ruchita Bari

speech recognition to the field. Implemented a hidden Markov model (HMM)-based AED system with lattice rescoring using a feature set selected by Ada Boost based approach. Improved AED Via audio-visual intermediate integration using generalizable visual features. Using optical flow based spatial pyramid histograms; they planned a method for representing the highly variant visual cues of the acoustic events. Introduced the usage of spectro-temporal fluctuation features in a tandem connectionist approach, modified to generate posterior features separately for each fluctuation scale and then combine the streams to be fed to a classic Gaussian mixture model-hidden Markov model (GMM-HMM) procedure. Voice activity detection can also be regarded as a clustering problem, in which the goal is to classify the input signal into speech absence and speech presence frames. Hence, after choosing an appropriate feature space, one can use a clustering algorithm to obtain a VAD algorithm. Among different clustering methods, spectral clustering has recently become one of the most popular modern clustering algorithms. It is simple to implement, can be solved efficiently by standard linear algebra software, and very often outperforms the habitual clustering algorithms such as the k-means algorithm. Recently, spectral clustering has been utilized by several authors in signal processing applications such as image segmentation, speech separation and clustering of biological sequence data just to name a few.

In this paper, we present a novel voice activity detection algorithm using spectral clustering. In particular, we use a normalized spectral clustering algorithm cluster the Mel-frequency cepstrum coefficients (MFCC) of the received signal into two different clusters, i.e., speech presence and speech absence. The clustering problem can be complete using GMM. However, fitting a GMM to high dimensional data generally require a great amount of training data, and as the number of Gaussian mixture is increased, we need more and more training data to fit the GMM to high dimensional data. The fact that the distribution of natural data, like speech and transient noise is non-uniform and concentrates around low-dimensional structures motivates us to exploit the shape (geometry) of the distribution. for efficient learning. Among different dimensionality reduction techniques, kernel eigen map methods such as local linear embedding Laplacian eigen maps sacking eigen maps and diffusion maps (just to name a few) have recently attracted much research attention. These algorithms exhibit two major advantages over classical dimensionality reduction methods (such as principal constituent analysis or classical multidimensional scaling):

They are nonlinear, and they preserve local structures. The first aspect is essential as most of the time, in their original form, the data points do not lie on linear manifolds. The second point is connected to the fact that in many applications, distances of points that are far apart are meaningless, and therefore need not to be preserved. The main idea of these methods is to

use the dominant eigenvectors of Laplacian of the similarity matrix as the new lower dimension representation of the data. Our proposed algorithm is a supervised learning algorithm. One must train the system before it can be used. Training data is used for estimating the parameters of the kernel used in computation of the similarity matrix. The data is also used in finding two Gaussian mixture models for modeling the first two eigenvectors of the Laplacian of the similarity matrix corresponding to the first two leading eigenvalues of normalized Laplacian matrix. This means that we made the low dimensional representation of the original data (i.e., MFCC) using two different GMMs, one for each cluster. Upon receiving new unlabeled data, the optimum parameters of the kernel are utilized to find the similarity between the new data and the training set in order to find the low dimensional representation of new data. Using the GMMs obtained in the training step, the likelihood ratio is computed, and the final VAD is obtained by comparing that likelihood ratio to a threshold.

The rest of this paper is organized as follows. we formulate our problem and begin a novel VAD for online processing. Simulation results and performance comparison are presented in Finally.

2. Literature Survey

In 2011 S. Mousazadeh and I. Cohen publishing AR-GARCH, parameter estimation, noisy data, non-stationary noise as explain in these paper. Introduced a novel procedure based on the ML estimation method for parameter estimation of the AR-GARCH model in presence of additive noise. And An adaptive version of parameter estimation method, namely, the RML method.

In 2011 Jonathan Kola, Carol Espy-Wilson and Tarun Pruthi publishing Voice Activity Detection using VAD BOX as explain in these paper .Only despite poor performance in music2 noise, the VADs performed well in other periodic noises such as babble noise and music1 noise (music1 noise is instrumental,music2 noise is lyrical), therefore the performance of the VADs was not generally worse in periodic noises, though the worst performance was recorded in a periodic noise.

In 2012 Joon-Hyuk Chang publishing Statistical Model-Based Voice Activity Detection Based on Second-Order Conditional as explain in these paper. Introduced Conventional methods and the proposed method were evaluated in a quantitative comparison under various noise Environments.

In 2012 M. Espi, M. Fujimoto, D. Saito, N. Ono, and S. Sagayama publishing A Tandem Connectionist Model Using Combination Of Multi-scale Spectro-Temporal Fetures For Acoustic Event Detection as explain in these paper. Introduced Compared the performance in AED between traditional GMM-HMM, tandem connectionist with early integration, and tandem connectionist with late integration schemes, in AED of isolated acoustic events.

In 2013 Francois G. Germain, Dennis L. Sun, Gautham J. Mysore publishing Speaker and Noise Independent Voice Activity Detection . Only able to handle a variety of non-stationary noises at low signal-to-noise ratios.

3. Problem Formulation

In this section, we propose our voice activity detection method, which is based on spectral clustering. The basic idea behind spectral clustering method is to use several eigenvectors of the normalized Laplacian of the similarity matrix as a novel low dimension representation of the high dimension data points. Clustering is generally performed on this new representation of the data points using a conventional (weighted) k-means algorithm.

Here we introduce a novel technique for clustering the data based on GMM modeling of the eigenvectors of the normalized Laplacian of the similarity matrix. Every clustering problem consists of the following three main stages: selecting an appropriate feature space, choosing a metric as a notion of similarity between data-points, and selecting the clustering algorithm.

3.1. Feature Selection

Let $x_{sp}(n)$ denote a speech signal and let $x_{tr}(n)$ and $x_{st}(n)$ be the additive contaminating transient and stationary noise signals, respectively. The signal measured by a microphone is given by:

$$y(n) = x_{sp}(n) + x_{tr}(n) + x_{st}(n) \tag{1}$$

The goal is to determine whether there exists speech signal in a given time frame (each approximately 16–20 msec long). Here we choose absolute value of MFCCs and the arithmetic mean of the log-likelihood ratios for the individual frequency bins as our feature space. Morespecifically, let

$Y_m(t, k)(t = 1, 2, \dots, N; k = 1, 2, \dots, k_m)$ And $Y_s(t, k)(t = 1, 2, \dots, N; k = 1, 2, \dots, k_s)$ be the absolute value of the MFCC and the STFT coefficients in a given time frame, respectively. MFCC and the STFT coefficients are computed in k_m and k_s frequency bins, respectively. Then, each frame is represented by a $(k_m + 1)$ dimension column vector defined as follows.

$$Y(:, t) = \begin{bmatrix} Y_{m(:, t)} \\ \Lambda t \end{bmatrix} \tag{2}$$

Where $y_m(:, t)$ is the column of y_m and Λt is the arithmetic mean of the log-likelihood ratios for the individual frequency bands in frame which is given by:

$$\Lambda t = \frac{1}{k_s} \sum_{k=1}^{k_s} \left(\frac{\Gamma_k(t) \varepsilon_k(t)}{1 + \varepsilon_k(t)} - \log(1 + \varepsilon_k(t)) \right) \tag{3}$$

Where $\varepsilon_k(t) = \frac{\lambda_s(t, k)}{\lambda_n(t, k)}$ is called a priori SNR,

which can be estimated using decision-directed

$\lambda_n(t, k)$ is the variance of stationary noise in t -th time and frame k th frequency bin $\Gamma_{K(T)=|Y_s(t,k)|^2 / \lambda_N(t,k)}$ is called posterior SNR, ε is kernel width obtaining during the training phase, $\lambda_N(t, k)$ is the variance of stationary noise in t -th time frame and k -th frequency bin which can be estimated from training data (if there exist sequences consisting of only stationary noise) or by improved minima controlled recursive averaging (IMCRA). The reason behind choosing this feature space for each frame is as follows. The likelihood ratio has been long exploited as a feature for voice activity discovery in presence of stationary noise. The Mel-frequency cepstrum coefficient has been a representation of the short-term power spectrum of a sound, base on a linear cosine transform of a log power spectrum of a nonlinear Mel scale of frequency. MFCCs are commonly used as features in speech recognition systems. Combining these two features appropriately would be a suitable feature space for voice activity detection in presence of transient noise.

3.2. Clustering Algorithm

A popular way for representing the data is to build a similarity graph, which is a weighted graph $G=(V,E)$, where V is the set of the vertices and E is the set of the edges of the graph. Each vertex v_i in this graph represents a data point $Y(:, i)$. Each edge e_{ij} between two vertices v_i and v_j carries a non-negative weight $W(i, j) \geq 0$, which is a measure of similarity between the corresponding points. We assume that the graph G is an undirected one (i.e $W(i, j) = W(j, i)$). A similarity matrix W is matrix whose (j, i) -th element equals to $W(i, j)$. Using the concept of subspace comparison, Bach and Jordan [19] proposed a spectral clustering algorithm using the eigen vectors of matrix $D^{-1/2} W D^{-1/2}$ where D is a diagonal matrix whose i -th diagonal element equals to $\sum_{j=1}^N W(i, j)$ More specifically, let K be the number of clusters and U be a matrix consisting of the first K eigenvectors of $D^{-1/2} W D^{-1/2}$ corresponding to K 's largest eigen values of $D^{-1/2} W D^{-1/2}$. The clustering has done either by running a weighted k-means algorithm on where each point is represented by a row of or by running a k-means algorithm on $V = D^{1/2} U (U^T D U)^{-1}$ where each point is represented by a row of . The most important drawback of this method is that it does not prepare a toll for controlling the tradeoff between probability of false alarm and probability of detection. We will deal with this issue by using GMM modeling of the eigenvectors.

The most important part of a spectral clustering algorithm is the calculation of the similarity matrix. Although the definition of the similarity between points is an application and data dependent, a popular way of defining the similarity matrix is to use a Gaussian kernel as follows:

$$W(i, j) = \exp \left(- \frac{\|Y(:, i) - Y(:, j)\|_2^2}{\sigma} \right) \tag{4}$$

Where $Y(:, i)$ is the i -th data point. The selection of σ is commonly done manually. Selecting σ automatically by running their clustering algorithm repeatedly for a number of values of σ and selecting the one that provides the least distorted clusters. Set the scale by examining a logarithmic scale of the sum of the kernel weights without computing the spectral decomposition of the transition matrix. Suggested calculating a local scaling parameter σ_n for each data point instead of selecting a single scaling parameter σ . The above mentioned methods are somewhat heuristic or hard to implement because of high computational load. The kernel based on minimization of a cost function that characterizes how close the eigen structure of the similarity matrix W is to the true partition.

Our voice activity detection algorithm is a supervised learning one. As a consequence, one must utilize training data in order to adjust the parameters of the algorithm and use those parameters for clustering unlabeled data. In the next two subsections, we illustrate how each of these stages works.

3.3. Learning Algorithm:

In this section, we introduce our learning algorithm based on the method presented in. Suppose that we have a database of clean speech signal, a database of transient noise, and a database of stationary noise. We choose different signals from each database and combine them as follows. Let $x_{sp}^l(n), x_{tr}^l(n), x_{st}^l(n)$ be the l -th speech signal, transient noise, and stationary noise, respectively. Without loss of generality, we assume that all of signals are the same length (i.e. N_l). We built the l -th training sequence, $Y^l \in \mathbb{R}^{k_m+1 \times 3N^l}$, as follow. Let

$$x_1^l(n) = x_{sp}^l(n) + x_{st}^l(n) \tag{5}$$

$$x_2^l(n) = x_{tr}^l(n) + x_{st}^l(n) \tag{6}$$

$$x_3^l(n) = x_{sp}^l(n) + x_{st}^l(n) + x_{tr}^l(n) \tag{7}$$

and let Y_1^l, Y_2^l, Y_3^l be the feature matrix extracted using (2) and (3) from $x_1^l(n), x_2^l(n)$ and $x_3^l(n)$ Then, the l -th training data is obtained by concatenating these matrices as follows:

$$Y^l = [Y_1^l \quad Y_2^l \quad Y_3^l] \tag{8}$$

A typical training sequence is depicted in fig.1. For each of these training sequences, we compute the indicator matrix of the partitions $C^l \in \mathbb{R}^{3N^l \times 4}$ using (9), where C_{ij}^l , is the (i,j) -th element of C^l , $x(\cdot)$ is an indicator function that equals to one if its argument is true and zero otherwise, T_{sp} and T_{tr} are speech and transient noise thresholds and are chosen as the maximum value of threshold such that thresholding the speech or transient noise has no significant effect, \oplus and \otimes are

logical OR and logical AND operators, respectively. $P(\cdot)$ is a power calculation operator defined by.

$$P(x_{sp}^l(:, i)) = \frac{1}{k_s} \sum_{k=1}^{k_s} \|x_{sp}^l(k, i)\|_2^2 \tag{9}$$

$$P(x_{tr}^l(:, i)) = \frac{1}{k_s} \sum_{k=1}^{k_s} \|x_{tr}^l(k, i)\|_2^2 \tag{10}$$

Where $x_{sp}^l(:, i)$ and $x_{tr}^l(:, i)$ are the STFT coefficients of $x_{sp}^l(n)$ and $x_{tr}^l(n)$ in the i -th frame, respectively.

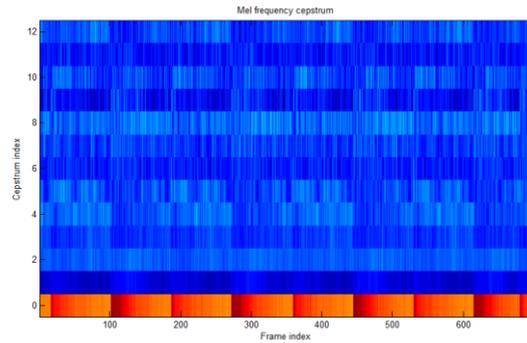


Fig 1 Typical training sequence consisting of speech signal

The last and most important object to be defined, in order to use the training algorithm presented in, is the parametric similarity matrix, i.e., W_θ^l . For our problem, we define this matrix as follows:

$$W_\theta^l(i, j) = \exp(\sum_{p=-\rho}^{\rho} -\alpha p Q(i + p, j + p)) \tag{11}$$

$$Q(i, j) = \left\| Y_m^l(:, i) \left(1 - \exp\left(\frac{-\Lambda_i^l}{\epsilon}\right) \right) - Y_m^l(:, j) \left(1 - \exp\left(\frac{-\Lambda_j^l}{\epsilon}\right) \right) \right\|_2^2 \tag{12}$$

Where $\theta = [\epsilon, \alpha, \rho, \alpha, \rho+1, \dots, \alpha, \rho-1, \alpha, \rho] \in \mathbb{R}^{2\rho+2}$ is the vector of parameter, $Y_m^l(:, i)$ and Λ_i^l are the absolute value of MFCC and the log likelihood ratio of the l -th training sequence in the i -th frame, respectively, and $\|\cdot\|_2$ is the Euclidian norm of a vector. The reason behind choosing this weight function is discussed in the following paragraph.

For designing an appropriate weight matrix, we have taken the following two points into consideration. The first one was the similarity between two individual frames, and the second one was the effect of neighboring frames on deciding whether a specific frame contains speech or transient noise. Combining these two features (i.e., MFCC and likelihood ratio) as in, results in a good metric as a similarity notion between two frames for voice activity detection in presence of transient noise. More specifically, if there exists speech signal or transient noise in a specific frame, the value of likelihood ratio is large (see fig 1(right)); hence, the exponential term in (13) just

about Equals to zero, and the feature for that frame will be approximately the MFCCs. On the other hand, if a specific frame consists of only stationary noise, then the likelihood ratio will be small, and the exponential term in (2) just about equals to one. Consequently, the feature vector will approximately be equal to zero vector for those frames that only contain stationary noise. Considering the MFCCs of a typical training sequence as depicted in fig.1(left), it is apparent that there might exist two frames in the speech part (left part of fig) and the transient noise part (right part of fig.) that are very similar to each other (in the sense that their Euclidean distance is small) but belong to two different clusters. The characteristic that distinguishes the frames containing speech from those frames containing transient noise is that the neighboring frames of a specific speech frame are almost the same, which is not true for transient noise. Choosing the weight function as in(12)-(13), guarantees small similarity between two frames from different classes (speech and transient noise) even if they are very similar to each other (in the Euclidean sense), because of the large distance between neighboring frames. Upon defining the parametric weight function, the parameters can be obtained by solving the following optimization problem.

$$\theta^{opt} = \arg \min_{\theta} \frac{1}{L} \sum_{l=1}^L F(W_{\theta}^l, C^l) \quad (13)$$

$$F(W, C) = \frac{1}{2} \left\| rr^T - D^{1/2} C (C^T D C)^{-1} C^T D^{1/2} \right\|_F^2 \quad (14)$$

Where L is the number of training sequence, $(\cdot)^T$ denotes transpose of a vector or a matrix, and r is an approximate orthonormal basis of the projections on the second principal $D^{-1/2} W D^{-1/2}$ of obtained by classical orthogonal iteration. In practice, we use the gradient method problem.

3.4. Testing Algorithm

A testing algorithm aims to cluster the unlabeled data. The most straightforward way to perform clustering using spectral methods into K disjoint clusters is to use the parameters obtain by the learning algorithm, construct the similarity matrix W , compute K the eigenvectors of $D^{-1/2} W D^{-1/2}$ corresponding to the first largest eigenvalues (denoted by U), and run weighted k-means algorithm on U or k-means algorithm on $V = D^{1/2} U (U^T D U)^{-1}$. This method has two major drawbacks. First, this method can only be used for batch processing (offline giving out) of data. The second and more important one is that, this method does not allow the user to control the tradeoff between the probability of false alarm and the probability of detection. Every detection algorithm must be equipped with a tool such that one can increase the probability of detection (probably) by increasing the probability of

false alarm. In order to overcome these two shortcomings, we utilize the lean-to method proposed in based on the fact that two test points are similar if they see the training data similarly, and the likelihood ratio test as our decision rule. In order to compute the likelihood ratio, we use GMM to model the eigenvectors of normalized Laplacian matrix. In what follows, we discuss these two issues in more detail.

Let $W_{\theta^{opt}}^l$ be the similarity matrix of l -th training sequence and $U^l \in \mathbb{R}^{3N^l \times 2}$ be a matrix consisting of the two eigenvectors of corresponding to the first two largest eigenvalues. Let the column concatenation of U^1 through U^L be

$$U = [(E^1 \odot U^1)^T, \dots, (E^l \odot U^l)^T, \dots, (E_L \odot U^L)^T]^T \quad (15)$$

$$E^l = \sqrt{C^l \text{diag}(1_{1 \times N^l} C^l) 1_{4 \times 2}}; \quad l = 1, 2 \dots L \quad (16)$$

Where \odot is symbol by term multiplication $\text{diag}(a)$, is a diagonal matrix whose diagonal is vector a and $1_{m \times n}$ is an m by n matrix of ones. This normalization of the matrices U^1 through U^L is due to a possible different number of points in the same cluster of different training sequences. Because of sign ambiguity in computation of eigenvectors, each of these eigenvectors is computed such that the mean of each cluster (noise only cluster or speech cluster) is as close as possible to the mean of each cluster of the first training sequence. More specifically, we compute the mean of low dimensional representation of each of the two clusters in the first training sequence and choose the sign of the eigenvectors corresponding to the remaining training series, such that their means are close to the means of the clusters in the first training sequence. We have selected this approach instead of combining all training sequences as a single training sequence because of computational load and memory usage. Combining all training sequence as a single sequence leads to a very large similarity matrix that cannot be handled computationally. This method is in some sense equivalent to ignoring the similarity between each training sequence, which is a widespread approach for sparsifying the similarity matrix.

Once the matrix U , a new representation of the training data, is obtained, we use Gaussian mixture modeling to model each cluster (i.e., speech presence or absence) with a different GMM. A mixture model is a probabilistic model that assumes the underlying data belongs to a mixture division. In a mixture distribution, the density function is a convex combination of other probability concreteness functions. The most common mixture distribution is the Gaussian density function, where each of the mixture components has a Gaussian distribution. This model has been utilized in many machine learning and speech processing applications such as speaker verification, texture retrieval, and handwriting recognition just to name a few. Parameters of the GMM can be estimated by the

Expectation- Maximization (EM) method, and the number of Gaussian component to be used can be selected by the Akaike information criterion (AIC) or Bayesian information criterion (BIC). The procedure to obtain the GMM for each cluster is as follows. For each cluster (i.e., speech presence or absence), we find the rows of the matrix U corresponding to that cluster by using the indicator matrix. Then, by exploiting the EM algorithm and AIC or BIC criterion, we fit a GMM to the new data representation in that cluster. Since the matrix U only depends on the training data, the GMM model for each of the two hypotheses (i.e., speech presence or absence) is obtained during the training phase.

Now suppose we are given T frames of unlabeled data, and we want to decide whether each of these frames belongs to the speech existence or speech absence clusters. For each of these frames, we first extract the feature vector. Using (2) and (3). Let

$$Z(:, t) = \begin{bmatrix} Z_M(:, t) \\ \Lambda_t^Z \end{bmatrix} \quad t = 1, 2 \dots T \quad (17)$$

be the feature vector extracted from unlabeled data, where $Z_M(:, t)$ it is the absolute value of the MFCC of the t -th frame, and Λ_t^Z is the likelihood ratio of t -th unlabeled frame obtained by (3). The similarity matrix between the new data and training data is computed as follows:

$$B = [(B_{\theta^{opt}}^1)^T, (B_{\theta^{opt}}^2)^T, (B_{\theta^{opt}}^L)^T] \quad (18)$$

$$B_{\theta^{opt}}^l(i, j) = \exp(\sum_{p=-p}^p -\alpha_p^{opt} Q^l(i + p, j + p)) \quad (19)$$

$$Q^l(i, j) = \left\| \begin{aligned} & Y_m^l(:, i) \left(1 - \exp\left(\frac{-\Lambda_i^l}{\epsilon^{opt}}\right) \right) \\ & - Z_m(:, j) \left(1 - \exp\left(\frac{-\Lambda_j^Z}{\epsilon^{opt}}\right) \right) \end{aligned} \right\|_2 \quad (20)$$

Where $\theta^{opt} = [\epsilon^{opt}, \alpha_{-p}^{opt}, \alpha_{-p+1}^{opt} \dots \dots \alpha_{p-1}^{opt}, \alpha_p^{opt}]$ is the optimum kernel parameters vector obtain in learning stage by solving the optimization problem in (14), and $B_{\theta^{opt}}^l(i, j) (1 \leq i \leq N^l; 1 \leq j \leq T)$ is the (i, j) -th element of the matrix $B_{\theta^{opt}}^l$. Once the similarity matrix between unlabeled data and training data has been computed, the new data representation in terms of eigenvectors of the Laplacian can be easily approximated by the following equation:

$$\tilde{U} = \text{diag}((1B_{k_{nn}})^{-1}) B_{k_{nn}}^T U \quad (21)$$

where i -th the column of the matrix $B_{k_{nn}}$ is obtained by setting to zero all elements of the i -th column of B , except K the largest elements. The subscript k_{nn} stands for K -nearest neighbor. The last equation means that

the low dimensional representation of a given test point is simply the weighted mean of the low representation k -nearest neighbor of that point in the training set. Using this new illustration of the unlabeled data, the decision rule can be obtained by a likelihood ratio test as follows. Let H_0 and H_1 be speech absence and presence hypotheses, respectively. Let $f(\cdot; H_0)$ and $f(\cdot; H_1)$ be the probability density function of those rows U corresponding to noise only frames and frames containing speech signal, respectively. These two probability density functions were obtained by GMM modeling in the training stage. The likelihood ratio for a new unlabeled frame is given by:

$$\Gamma_t = \frac{f(\tilde{U}(t,:); H_1)}{f(\tilde{U}(t,:); H_0)} \quad (22)$$

Where $\tilde{U}(t, :)$ is the t -th row of the matrix \tilde{U} . Practical evidence shows that using the information supplied by neighboring frames can improve the performance of VAD algorithms. This is because of the fact that frames containing speech signal are usually followed by a frame that also contains speech signal while the transient signals usually last for a single time frame. Using this fact, the decision time frame is obtained by:

$$[ht]VA_t = \sum_{j=-J}^J \Gamma_{t+j} \begin{matrix} H_1 \\ \geq T_h \\ H_0 \end{matrix} \quad t = 1, 2 \dots, T \quad (23)$$

Where T_h is a threshold which controls the tradeoff 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. In a practical implementation, a hangover scheme is required to lower the probability of false rejections. The hangover scheme does this by reducing the risk of a low-energy section of speech at the end of an utterance being falsely old, by arbitrarily declaring a era of speech activity after a period of speech activity has already been detected. This is based on the idea that speech occurrence are highly correlated with time. We use the hangover technique, More specifically, the quantity VA_t is the input of the hangover procedure, and a final VAD decision is obtained from lie over scheme.

4. Simulation Results and Performance Evaluation

In this section, we examine the performance of the proposed method using several simulations. We also compare the performance of our method with that of conventional statistical model-based methods. We perform our simulation for different types of stationary and transient noise for different SNR situations. The SNR is defined as the ratio of the speech energy to the power of stationary noise. The stationary noise energy is computed in persons frames where speech signal is present. All speech and transient noise signals are sampled at 16 kHz (although the same performance was obtained at 8 kHz sampling rate) and normalized to have unity as their greatest. Since the duration of

transient noise is small with respect to speech, defining SNR for transient noise is not useful. Instead, we normalize the transient noise and speech signal to have the same maximum amplitude, which is a very challenging case to treat each signal (speech or transient noise) is approximately 3 sec long. The training and testing sequences are constructed using the procedure introduced in (6) and (7) Speech signals are taken from the TIMIT database.

In the training step, we use $M=50$ different speech utterances (different speakers, half male and half female) and transient noise. In the testing step, we use $M=50$ different speech utterances (different speakers from the training set, half male and half female) and transient noise (different from the training sequences) each approximately 3 sec long (the length of the testing signal is approximately 500 sec, with sixty percent of total frames containing speech). We use windowed STFT with a hamming window of $K_s=512$ samples long and 50% overlap between consecutive frames. We compute the MFCC in $K_m=24$ Mel frequency bands. To solve the optimization problem in the training stage, we use the function in. We solve this optimization problem under the constraint that all estimated parameters are strictly positive. This constraint results in an appropriate similarity matrix. The parameter in computing the matrix is set to 10.

In order to measure up to our method to the conventional statistical based method, we introduce two different kinds of false alarm probabilities. The first type denoted by P_{fa} , is defined as the probability that a speech free frame (i.e., consisting of only stationary noise or stationary noise with transient noise) is detected as a speech frame (i.e., exactly the same as probability of false alarm defined in conventional methods). The second type, denoted by $P_{fa_{tr}}$, is defined as the probability that a frame consisting of stationary and transient noise is detected as a speech frame. We need these two concepts to show the advantage of the projected method over conventional statistical model-based methods.

The number of frames that contain transient noise (which are mostly detected as speech in statistical model-based methods) is little with respect to the total number of frames. Such frames do not affect the probability of false alarm significantly if it is defined as the probability that a noise frame is detected as a speech frame.

Conclusion

I had proposed a novel voice activity detector based on spectral clustering method. My main concern had been dealing with noise cancellation, which is very difficult to handle. Almost all straight methods fail in this situation. Our VAD is a supervised learning algorithm that requires some training data in order to guess the parameter of the kernel used for calculation of a likeness environment. We used GMM to model the eigenvectors of the similarity matrix. In the testing stage, we used eigenvector extension and proposed a VAD which can be used for online processing of the data with a small delay. Simulation results have demonstrated the high performance of the proposed method, particularly its advantage in treating transient noises.

References

- S. Mousazadeh and I. Cohen(2011), AR-GARCH in presence of noise: Parameter estimation and its application to voice activity detection, *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 916–926.
- Jonathan Kola, Carol Espy-Wilson and Tarun Pruthi(2011), , Voice Activity Detection, *MERIT BIEN*
- Joon-Hyuk Chang(2012), , Statistical Model-Based Voice Activity Detection Based on Second-Order Conditional MAP with Soft Decision, *ETRI Journal, Volume 34, Number 2*
- M. Espi, M. Fujimoto, D. Saito, N. Ono, and S. Sagayama(2012), A tandem connectionist model using combination of multi-scale spectro-temporal features for acoustic event detection, in *Proc. ICASSP '12*, pp. 4293–4296.
- Francois G. Germain, Dennis L. Sun, Gautham J. Mysore (2013), Speaker and Noise Independent Voice Activity Detection