YCSLA 204
DISK / 22/7/02

ARTICLE IN PRESS

No. of pages: 22
DTD 4.3.1/SPS

Computer Speech and Language (2002) **xx**, x–xx
doi: 10.1016/S0885-2308(02)00028-1
Available online at http://www.idealibrary.com on IDEAL®

Classifier-based non-linear projection for adaptive endpointing of continuous speech

Bhiksha Raj [†§] **and Rita Singh** [‡]

[†] *Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, U.S.A.,*
[‡] *Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.*

Abstract

In this paper we present an algorithm for segmenting or locating the endpoints of speech in a continuous signal stream. The proposed algorithm is based on non-linear likelihood-based projections derived from a Bayesian classifier. It utilizes class distributions in a speech/non-speech classifier to project the signal into a 2-dimensional space where, in the ideal case, optimal classification can be performed with a simple linear discriminant. The projection results in the transformation of diffuse, nebulous classes in high-dimensional space into compact clusters in the low-dimensional space that can be easily separated by simple clustering mechanisms. In this space, decision boundaries for optimal classification can be more easily identified using simple clustering criteria. The segmentation algorithm proposed utilizes this property to determine and update optimal classification thresholds continuously for the signal being segmented. The performance of the proposed algorithm has been evaluated on data recorded under extremely diverse environmental noise conditions. The experiments show that the algorithm performs comparably to manual segmentations even under these diverse conditions.

1. Introduction

Automatic Speech Recognition (ASR) systems of today have little difficulty in generating good recognition hypotheses for large sections of continuously recorded signals containing speech, when they are recorded in controlled, quiet environments. In such environments silence is easily recognized as such and is clearly distinguishable from speech. However, when the signal is noisy the ASR system is no longer able to clearly discern whether a given segment is speech or noise, and often recognizes spurious words in regions where there is no speech at all. This can be avoided if the beginnings and ends of sections of the signal containing speech are identified prior to recognition and recognition is performed only within these boundaries. The process of identification of these boundaries is commonly referred to as endpoint detection or segmentation. While

[§] E-mail: bhiksha@merl.com (B. Raj).

0885-2308/02/$35

there are minor differences in the contexts in which these two terms are used, we will consider them to be synonymous in this paper.

Several methods of endpoint detection that have been proposed in the literature. We can roughly categorize them as rule-based methods and classifier-based methods. Rule-based methods use heuristically derived rules relating to some measurable properties of the signal to discriminate between and non-speech. The most commonly used property is the variation in the energy in the signal. Rules based on energy are usually supplemented by other information such as durations of speech and non-speech events (Lamel *et al.*, 1981), zero crossings (Rabiner and Sambur, 1975), pitch (Hamada *et al.*, 1990), etc. Other notable methods in this category use time-frequency information to locate regions of the signal that can be reliably tagged and then expanded to adjacent regions (Junqua *et al.*, 1994).

Classifier-based methods model speech and non-speech events as separate classes and treat the problem of endpoint detection as one of classification. The class distributions may be modelled by static distributions such as Gaussian mixtures (e.g. Hain and Woodland, 1998) or by dynamic structures such as hidden Markov models (e.g. Acero *et al.*, 1993). More sophisticated versions use the speech recognizer itself as an endpoint detector. Some endpointing algorithms do not clearly belong to either of the two categories, e.g. those that use only the local variations in the statistical properties of the incoming signal to detect endpoints (Siegler *et al.*, 1997; Chen and Gopalakrishnan, 1998).

Rule-based segmentation strategies have two drawbacks. Firstly, the rules are specific to the feature set used for endpoint detection and fresh rules must be generated for every new feature considered. Due to this only a small set of features for which rules are easily derived are used. Secondly, the parameters of the applied rules must be fine tuned to the specific acoustic conditions of the data, and do not easily generalize to other conditions.

Classifier-based segmenters, on the other hand, usually consider parametric representations of the entire spectrum of the signal for endpoint detection. While they typically perform better than rule-based segmenters, they too have some shortcomings. They are specific to the kind of recording environments that they have been trained for, e.g. they perform poorly on noisy speech when trained on clean speech, and vice versa. They must therefore be adapted to the current operating conditions. Since the feature representations usually have many dimensions (typically 12–40 dimensions), adaptation of classifier parameters requires relatively large amounts of data and has not always been observed to result in large improvements in speech/non-speech classification accuracy (Hain and Woodland, 1998). Moreover, when adaptation is to be performed, the segmentation process becomes slower and more complex. This can increase the time lag (or *latency*) between the time at which endpoints occur and the time at which they are identified, which might affect runtime implementations. When classes are modelled by dynamical structures such as HMMs, the decoding strategies used (e.g. Viterbi, 1967) can introduce further latencies. Recognizer-based endpoint detection involves even greater latency since a single pass of recognition rarely results in good segmentation and must be refined by additional passes after adapting the acoustic models used by the recognizer. The problems of high dimensionality and higher latency render classifier-based segmentation less effective in many situations. Consequently, classifier-based segmentation strategies are mainly used only in offline (or *batchmode*) segmentation.

In this paper we propose a classifier-based method of endpoint detection which is based on non-linear likelihood-based projections derived from a Bayesian classifier. In the proposed method, high-dimensional parametrizations of the signal are projected onto a 2-dimensional space using the class distributions in a speech/non-speech classifier. In this 2-dimensional space the separation between classes is further increased by an averaging operation. Rather than adapting classifier distributions, this algorithm continuously updates the estimate of the optimal classification boundary in this 2-dimensional space. The performance of the proposed algorithm has been evaluated on the SPINE (SPINE, 2001) evaluation data, which are recorded under extremely diverse environmental noise conditions. The recognition experiments show the method to be highly effective, resulting in minimal loss of recognition accuracy as compared to manually obtained segment boundaries.

In the rest of this paper we describe the proposed algorithm and the evaluation experiments in detail. In Section 2 we describe the non-linear projections used by the algorithm. In Section 3 we describe how decision boundaries are obtained for the projected features. In Section 4 we describe two implementations of the segmenter. In Section 5 we describe experimental results and in Section 6 we present our conclusions.

2. Segmentation features

In any speech recording, the speech segments differ from non-speech segments in many ways. The energy levels, energy flow patterns, spectral patterns and temporal dynamics of speech are consistently different from those of non-speech. Feature representations used for the purpose of distinguishing speech from non-speech must capture as many of these distinguishing features as possible. For this reason, features used by ASR systems for recognition are particularly suitable. These are typically based on spectral representations derived from the short-term Fourier transform of the signal and are further augmented by difference features that capture the trends in the basic feature (Rabiner and Juang, 1993). Figure 1 shows scatter plots of the first four dimensions of a typical feature vector used to represent signals in ASR systems. The dark regions in the plots represent non-speech events in the signal, and the light regions represent speech events. In both plots in Figure 1, speech and non-speech are observed to have distinctly different distributions.

Such feature representations however tend to have relatively high dimensionality. For example, typical cepstral vectors are 13-dimensional which become 26-dimensional when supplemented by difference vectors. When using high-dimensional features for distinguishing speech from non-speech, Bayesian classifiers are usually more effective than rule-based ones. Bayesian classifiers are however fraught with problems. When the test data do not match the training data used to train the classifiers, they perform poorly. To avoid this problem classifier distributions are typically trained using a large variety of data, so that they generalize to a large number of test conditions. However, it is impossible to predict every kind of test condition that may be encountered and mismatches between the test data and the distributions used by the classifier will always occur. To compensate for this, the class distributions must be adapted to the test data. Commonly used adaptation methods are *maximum a posteriori* (MAP) (Duda *et al.*, 2000) and Extended MAP (Lasry and Stern, 1984) adaptation, and maximum likelihood (ML) adaptation methods such as MLLR (Leggetter and Woodland, 1994). For high-dimensional features both MAP and ML require moderately large amounts of

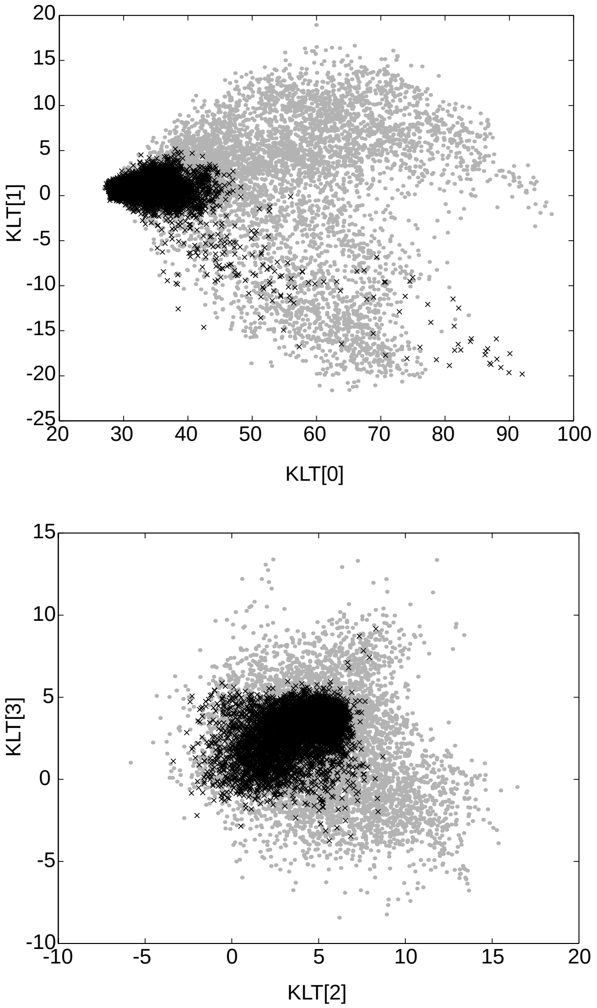

Figure 1. Scatter plot of four components of a typical feature representation of a recorded signal. The feature vectors used in this case were derived by KLT transformation of Mel-frequency log-spectral vectors derived from 25 MS frames of the signal, where adjacent frames overlapped by 15 MS. The left panel shows the scatter of the first two components of the vectors. The right panel shows the scatter of the third and fourth components. In both figures the dark crosses represent vectors from non-speech segments in the signal. The gray points represent vectors from speech segments. The actual scatter of the gray points extends into the black region, but is obscured.

data. In most cases, no labelled samples of the test data are available and the adaptation must therefore be unsupervised. Unsupervised MAP adaptation is generally ineffective (Doh, 2000). Even ML adaptation does not result in large improvements in classification over that given by the original mismatched classifier, for speech/non-speech classification (e.g. Hain and Woodland, 1998). Additionally, for the high-dimensional features considered, MAP and ML adaptation methods require multiple passes over the data and are computationally expensive. This can be a problem since endpoint detection is usually required to be a low computation task.

Many of the problems due to high-dimensional spectral features can be minimized or eliminated by projecting them down to a lower-dimensional space. However, such a projection must retain all classification information from the original space. Linear projections such as the KLT and LDA result in loss of information when the dimensionality of the reduced space is too small. We therefore resort to discriminant analysis for a non-linear dimensionality reducing projection that is guaranteed not to result in any loss in classification performance under ideal conditions (Singh and Raj, 2002). In the following subsection we describe this in greater detail.

2.1. Likelihoods as discriminant projections

Bayesian classification can be viewed as a combination of a non-linear projection and classification with linear discriminants. When attempting to distinguish between N classes, data vectors are non-linearly projected onto an N-dimensional space, where each dimension is a monotonic function, typically the logarithm, of the probability of the vector (or the probability density value at the vector) for one of the classes. An incoming d-dimensional vector X is thus projected onto an N-dimensional vector Y:

$$\begin{aligned} Y &= [\log(P(X|C_1))\log(P(X|C_2))\ldots\log(P(X|C_N))] \\ &= [Y_1 Y_2 \ldots Y_N], \end{aligned} \tag{1}$$

where $\log(P(X|C_i))$ is the log likelihood of the vector X computed using the probability distribution or density of class C_i. This constitutes Y_i, the ith component of Y. Equation (1) defines a *likelihood projection* into a new N-dimensional space of likelihoods. In this space, the optimal classifier between any two classes C_i and C_j is a simple linear discriminant of the form

$$Y_i = Y_j + \varepsilon_{i,j}, \tag{2}$$

where $\varepsilon_{i,j}$ is an additive constant that is specific to the discriminant for classes C_i and C_j. These linear discriminants define hyperplanes that lie at 45° to the axes representing the two classes. In the N-dimensional space, the decision region for any class C_i is the region bounded by the $N-1$ hyperplanes

$$Y_i = Y_j + \varepsilon_{i,j}, \quad j = 1, 2, \ldots, N, \ j \neq i. \tag{3}$$

The classification error expected from the simple optimal linear discriminants in the likelihood space is the same as that expected with the more complicated optimal discriminant in the original space (Singh *et al.*, 2001). Thus, when $N < d$, the likelihood projection constitutes a dimensionality reducing projection that accrues no loss whatsoever of information relating to classification.

For a two-class classifier, such as a speech/non-speech classifier, the likelihood projection reduces the data to only two dimensions. Figure 2 shows an example of the 2-dimensional likelihood projections for the data shown in Figure 1. For a two-class classifier, further dimensionality reduction is possible for no loss of information by projecting the 2-dimensional likelihood space onto the axis defined by

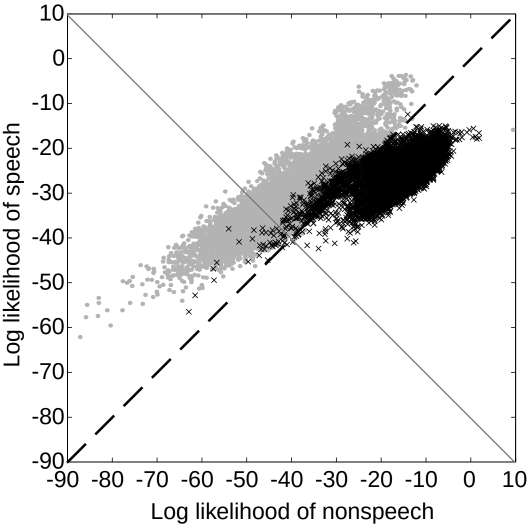

Figure 2. Scatter of the log likelihood of the speech frames computed using the distributions of the non-speech and speech classes. The dotted line shows the optimal linear discriminant between the classes. This discriminant performs exactly as the high-dimensional classifier in the original data space. The solid line shows an axis that is parallel to the one onto which the data are projected to obtain likelihood differences. This is orthogonal to the optimal linear discriminant.

$$Y_1 + Y_2 = 0. \tag{4}$$

This axis is orthogonal to the optimal linear discriminant $Y_1 = Y_2 + \varepsilon_{1,2}$. The unit vector $\hat{u}$ along the axis is $[1/\sqrt{2}, -1/\sqrt{2}]$. The projection Z of any vector $Y = [Y_1, Y_2]$, derived from a high-dimensional vector X, onto this axis can be computed as $\sqrt{2}Y \cdot \hat{u}$, which is given by

$$Z = Y_1 - Y_2 = \log(P(X|C_1)) - \log(P(X|C_2)). \tag{5}$$

The multiplicative factor of $\sqrt{2}$ has been introduced in the projection for simplification and does not affect classification as it merely results in a scaling of the projected features. Figure 3 shows the histograms of such a 1-dimensional projection of the speech and non-speech vectors in the signal used in Figure 1. Figure 4 shows the combined histogram of the speech and non-speech data. From these figures we observe that both the speech and non-speech data have distinctive, largely connected distributions. Further the combined histogram shows a clear inflexion point between the two, the position of which actually defines the optimal classification threshold between speech and non-speech.

The optimal linear discriminant in the 2-dimensional likelihood projection space is guaranteed to perform as well as the optimal classifier in the original multidimensional space only if the likelihoods of the classes are computed using the *true* distribution (or density) of the two classes. When the distributions used for the projection are not the true distributions, the classification performance of the optimal linear discriminant on the projected data is nevertheless no worse than the classification performance obtainable using *these* distributions in the original high-dimensional space (Singh *et al.*,

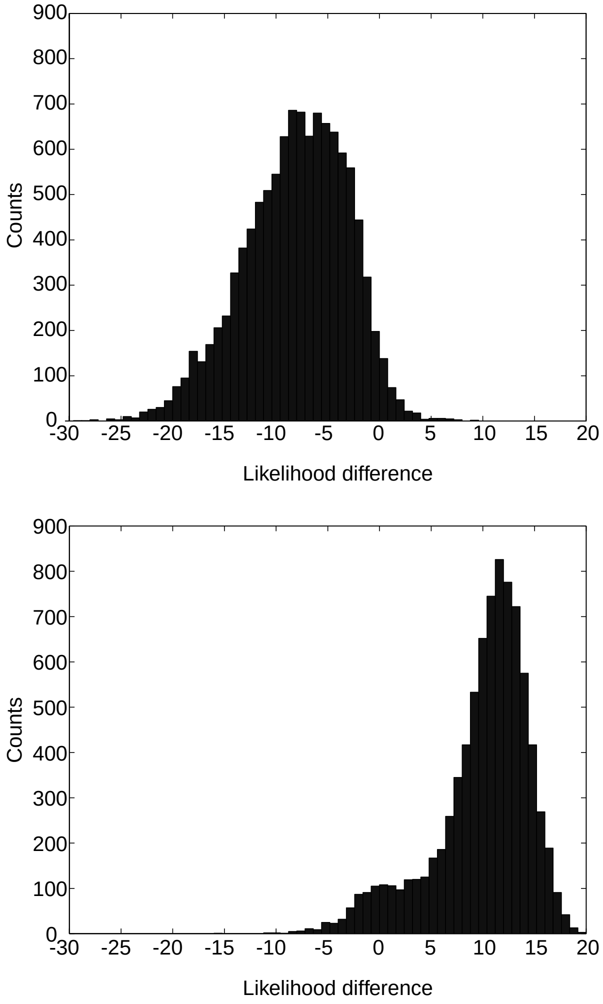

Figure 3. The left panel shows the histogram of the likelihood-difference values of frames of speech. The right panel shows a similar histogram for frames of non-speech signals. The data used for both plots have been sampled from the SPINE1 training corpus.

2001). However, the optimal linear discriminant for the test data may not be easily determinable. Figure 5a illustrates this problem. through an example where the distributions of the likelihood-difference features for speech and non-speech overlap to such a degree that the likelihood-difference histogram exhibits only one clear mode. The threshold value corresponding to the optimal linear discriminant cannot therefore be determined from this distribution. Clearly, the classes need to be separated further in order to improve our chances of locating the optimal decision boundary between them. In the following subsection we describe how the separation between the classes in the space of likelihood differences can be increased by an averaging operation.

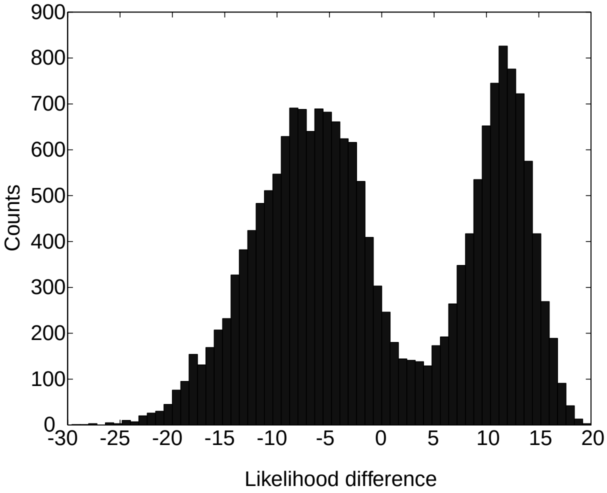

Figure 4. The histogram of likelihood differences for the combined speech and non-speech data used in Figure 3.

2.2. The effect of averaging on the separation between classes

Let us begin by defining a measure of the separation between two classes. Given two classes C_1 and C_2 of a scalar random variable Z, whose means are given by μ_1 and μ_2 and variances by V_1 an V_2, respectively. We can define a function $F(C_1, C_2)$ as

$$F(C_1, C_2) = \frac{(\mu_1 - \mu_2)^2}{c_1 V_1 + c_2 V_2}, \tag{6}$$

where c_1 and c_2 are the fraction of data points in classes C_1 and C_2, respectively. This ratio is analogous to the criterion, sometimes called the *Fischer ratio* or the *F-ratio*, used by the Fischer linear discriminant (Duda *et al.*, 2000) to quantify the separation between two classes. We will therefore also refer to the quantity in Equation (6) as the F-ratio in the rest of this paper. The difference between the Fischer ratio and Equation (6) is that Equation (6) is stated in terms of variances and fractions of data, rather than scatters. Like the Fischer ratio, the F-ratio in Equation (6) is a good measure of the separation between classes – the larger the ratio the greater the separation, and vice versa.

Consider a new random variable $\bar{Z}$ that has been derived from Z by replacing every sample of Z by the weighted average of K samples of Z, all of which are taken from a single class, either C_1 or C_2. The new random variable is given by

$$\bar{Z} = \sum_{i=1}^{K} w_i Z_i, \tag{7}$$

where Z_i is the ith sample of Z used to obtain $\bar{Z}$, $0 \leqslant w_i \leqslant 1$, and the weights w_i sum to 1. Since all the samples of Z that were used to construct any sample of $\bar{Z}$ come from the same class, that sample of $\bar{Z}$ is associated with that class. Thus all samples of $\bar{Z}$ correspond to either C_1 or C_2. The mean of the samples of $\bar{Z}$ that correspond to class C_1 is now given by

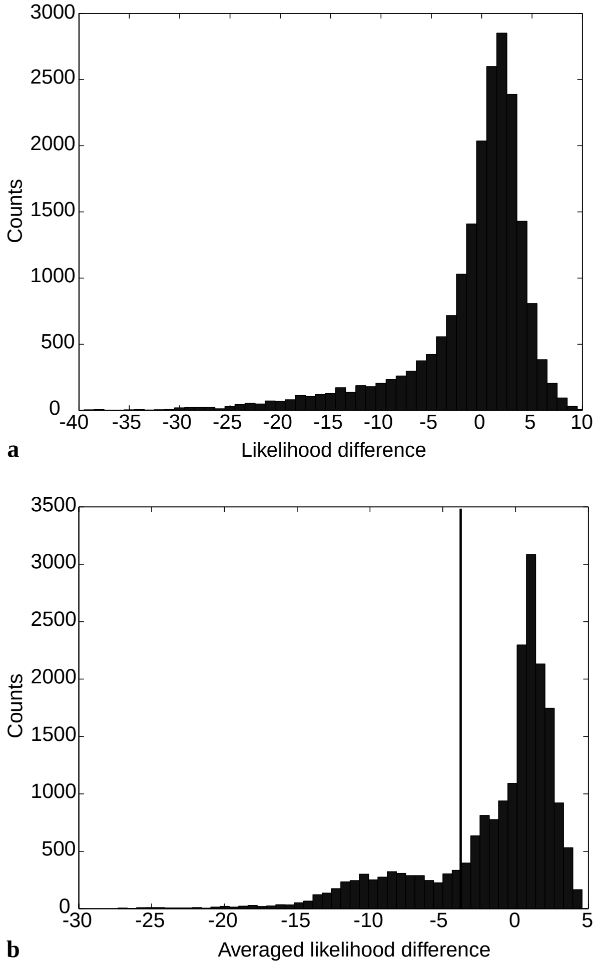

Figure 5. (a) A histogram of likelihood differences for a signal where the speech mode and non-speech mode overlap so the extent that the overall histogram has only one mode. (b) The histogram of the averaged likelihood differences for the data from (a). The speech and non-speech modes are now clearly visible. The vertical line shows the empirically determined optimal classification boundary between the two classes. The optimal threshold is very close to the inflexion point.

$$\bar{\mu}_1 = E(\bar{Z}|C_1) = \sum_{i=1}^{K} w_i E(Z|C_1) = \mu_1. \tag{8}$$

The mean of class C_2 is similarly obtained as $\bar{\mu}_2 = \mu_2$. The variance of the samples of $\bar{Z}$ belonging to class C_1 is given by

$$\begin{aligned} \bar{V}_1 &= E\Big(\big(\textstyle\sum_{i=1}^{K} w_i Z_i - \mu_1\big)^2\Big) = E\Big(\big(\textstyle\sum_{i=1}^{K} w_i (Z_i - \mu_1)\big)^2\Big) \\ &= \textstyle\sum_{i=1}^{K} \sum_{j=1}^{K} w_i w_j E((Z_i - \mu_1)(Z_j - \mu_1)) \\ &= V_1 \textstyle\sum_{i=1}^{K} \sum_{j=1}^{K} w_i w_j r_{ij} = \beta V_1, \end{aligned} \tag{9}$$

where r_{ij} is the relative covariance between Z_i and Z_j and β represents the summation term. It is easy to show from basic arithmetic principles that, since the weights sum to 1,

$$\sum_{i=1}^{K}\sum_{j=1}^{K} w_i w_j = \left(\sum_{j=1}^{K} w_j\right)^2 = 1. \tag{10}$$

Since $0 \leqslant w_j \leqslant 1$ and $|r_{ij}| \leqslant 1$, it follows that

$$\sum_{i=1}^{K}\sum_{j=1}^{K} w_i w_j r_{ij} \leqslant 1. \tag{11}$$

From Equations (9) and (11) it is clear that

$$\bar{V}_1 \leqslant V_1. \tag{12}$$

Thus, the variance of class C_1 for $\bar{Z}$ is no greater than that for Z. Similarly, $\bar{V}_2 \leqslant V_2$. Hence,

$$c_1\bar{V}_1 + c_2\bar{V}_2 = \beta(c_1V_1 + c_2V_2), \tag{13}$$

where $\beta \leqslant 1$. The F-ratio of the classes for the new random variable $\bar{Z}$ is given by

$$\bar{F}(C_1, C_2) = \frac{(\bar{\mu}_1 - \bar{\mu}_2)^2}{c_1\bar{V}_1 + c_2\bar{V}_2} = \frac{(\mu_1 - \mu_2)^2}{\beta(c_1V_1 + c_2V_2)} = \frac{F(C_1, C_2)}{\beta}. \tag{14}$$

If we can ensure that β is less than 1, then the F-ratio of the averaged random variable $\bar{Z}$ is greater than that of the original random variable Z. It is clear from Equation (11) that β is less than 1 if even one of the various r_{ij} values is less than 1.

This fact can be used to improve the separation between speech and non-speech classes in the likelihood space by representing each frame by the weighted average of the likelihood-difference values of a small window of frames around that frame, rather than by the likelihood difference itself. Since the relative covariances between all the frames within the window are not all 1, the β value for the new averaged likelihood-difference feature is also less than 1. If the likelihood-difference value of the ith frame is represented as L_i, the averaged value is given by

$$\bar{L}_i = \sum_{j=-K_1}^{K_2} w_j L_{i+j}. \tag{15}$$

Figure 5b shows the histogram of the averaged likelihood-difference features for the data in Figure 5a. We observe that the speech and non-speech are indeed more separable in Figure 5b than in Figure 5a. In fact, the averaging operation improves the separability between the classes even when applied to the 2-dimensional likelihood space. Figure 6 shows the scatter of the averaged likelihoods for the data used in Figure 2. Comparison of the two figures shows that the averaging has indeed improved the separation between classes greatly even in the 2-dimensional space.

One of the criteria for averaging to improve the F-ratio is that *all* the samples within the window that produces the averaged feature must belong to the same class. For a

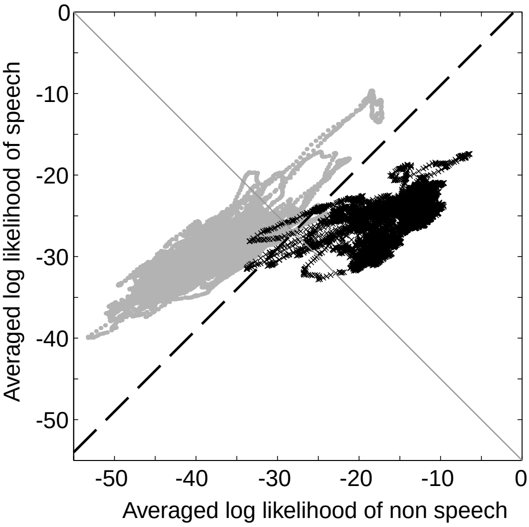

Figure 6. The scatter of averaged likelihoods for the data from Figure 2. The dotted line shows the best linear discriminant with slope 1 between the two classes. The classification error obtained with this discriminant is 1.3%. The optimal classification error for the original (unaveraged) data was 7.4%. Better linear discriminants are possible if the slope is allowed to vary from 1. For the data in this figure, the optimal linear discriminant of any slope, which has not been shown here, has a slope of 0.83 and results in an error rate of only 0.9%.

continuous signal there is no way of ensuring that any window contains only the same class of signal. However in any recording, speech and non-speech frames do not occur randomly. Rather they occur in contiguous blocks. As as result, except for the transition points between speech and non-speech, which are relatively infrequent in comparison to the actual number of speech and non-speech frames, most windows of the signal contain largely one kind of signal, provided they are sufficiently short. Thus, the averaging operation results in an increase in the separation between speech and non-speech classes in most signals. For example, the averaged likelihoods computed for the histogram in Figure 5b were, in fact, computed on the continuous signal without segregating speech and non-speech segments. We observe that the averaging results in an increased separation between speech and non-speech classes even in this case. Note that an averaging operation would not achieve any increase in the separation between classes if speech and non-speech frames were randomly interspersed in the incoming signal.

In this paper we therefore use the averaged likelihood-difference features to represent frames of the signal to be segmented. In the following sections we address the problem of determining which frames represent speech, based on these 1-dimensional features.

3. Threshold identification for endpoint detection

The histograms of averaged likelihood-difference values typically exhibit two distinct modes, with an inflexion point between the two. One of the modes represents the distribution of speech, and the other the distribution of non-speech. An example in shown in Figure 5b. The location of the inflexion point between the two modes approximately represents the optimal decision threshold where the two distributions crossover. This, however, is not easy to locate, since the histogram is, in general, not smooth, and has many minor peaks and valleys as can be seen in Figure 5b. For this reason, the problem of finding the inflexion point is not merely one of finding the minimum. In the following subsections we propose two methods of identifying the location of inflexion points in the histograms: Gaussian mixture fitting and polynomial fitting.

We note here that bimodal distributions are also exhibited by the energy of the speech frames. This has previously been exploited for endpoint detection and noise estimation. For example, Hirsch (1993) and Compernolle (1989), base the estimate of SNR and the presence of the speech signal based on the relative distance between the two modes. Several researchers (e.g. Cohen, 1989) have used the distance between the modes in the histogram of frame energies to estimate SNR. While these approaches look similar to the one suggested in this paper, the similarity between the two is only very superficial.

3.1. Gaussian mixture fitting

In Gaussian mixture fitting we model the distribution of the smoothed likelihood difference features of the signal as a mixture of two Gaussians, one of which is expected to capture the speech mode, and the other the non-speech mode. The mixture weights, means, and variances of the two Gaussians, represented as c_1, μ_1, V_1 and c_2, μ_2, V_2, are computed using the expectation maximization (EM) algorithm (Dempster *et al.*, 1977). The decision threshold is estimated as the point at which the two Gaussians crossover. This point is obtained as the solution to the equation

$$\frac{c_1}{\sqrt{2\pi V_1}} \exp\left(\frac{-(x-\mu_1)^2}{2V_1}\right) = \frac{c_2}{\sqrt{2\pi V_2}} \exp\left(\frac{-(x-\mu_2)^2}{2V_2}\right). \tag{16}$$

By taking logarithms on both sides, this reduces to the quadratic equation

$$\frac{(x-\mu_1)^2}{2V_1} - \log(c_1) + 0.5\log(V_1) = \frac{(x-\mu_2)^2}{2V_2} - \log(c_2) + 0.5\log(V_2) \tag{17}$$

only one of whose two solutions lies between $\mu_1$and μ_2. This is the estimated decision threshold.

The dotted contours in Figure 7 show the Gaussian mixture fit to the histogram in Figure 5b. The thin dotted contours show the individual Gaussians in the fit. The crossover point, marked by the rightmost dotted vertical line, is the estimate of the optimum decision threshold. We observe that the value of the estimated threshold is greater than the true optimum decision threshold, which would result in many more non-speech frames being tagged as speech frames as compared to the optimum decision threshold. This happens when the speech and non-speech modes are well separated. On the other hand, Gaussian mixture fitting is very effective in locating the optimum decision threshold in cases where the inflexion point in the histogram does not represent a local minimum. Figure 8 shows such an example.

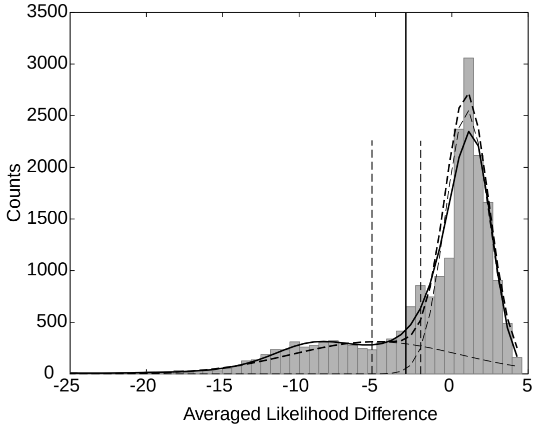

Figure 7. This figure shows the Gaussian and polynomial fits to the histogram in Figure 5b. The thin dotted lines show the individual Gaussians in the Gaussian fit. The thicker dotter line shows the overall Gaussian fit. The solid line shows the polynomial fit to the histogram. The long vertical line shows the empirically determined optimal classification threshold. The shorter vertical dotted lines show the classification thresholds given by the polynomial and Gaussian fits.

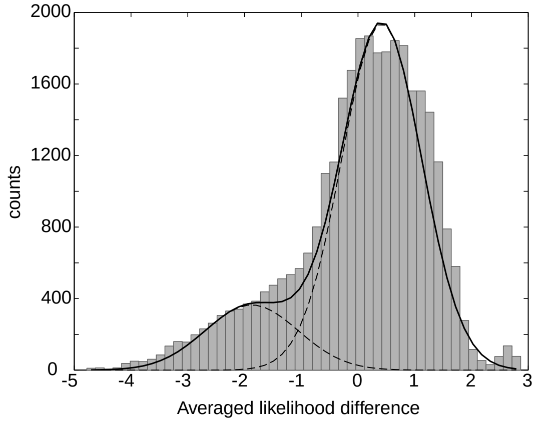

Figure 8. The histogram of the averaged likelihood differences for a signal where the speech and non-speech modes are so close that there is no local minimum between the two. Nevertheless a Gaussian fit to the distribution is successful at locating the classification threshold between the two. The two dotted curves show the two Gaussians in the fit. The solid line shows the overall Gaussian fit. The crossover point between the two Gaussians is the determined decision threshold.

3.2. Polynomial fitting

In polynomial fitting we obtain a smoothed estimate of the contour of the histogram using a polynomial. Due to inherent irregularities, direct modelling of the histogram contour as a polynomial frequently fails to capture the true underlying points of the histogram effectively. We therefore fit a polynomial to the *logarithm* of the histogram, with all bins incremented by 1 prior to the logarithm.

Let h_i be the value of the ith bin in the histogram. We estimate the coefficients of the polynomial

$$H(i) = a_K i^K + a_{K-1} i^{K-1} + \cdots + a_1 i + a_0, \tag{18}$$

where K is the polynomial order and $a_K, a_{K-1}, \ldots, a_0$ are the coefficients that minimize the error

$$E = \sum_i (H(i) - \log(h_i + 1))^2. \tag{19}$$

Optimizing E for the a_j values results in a set of linear equations that can be easily solved. The smoothed fit to the histogram can now be obtained from $H(i)$ by reversing the log and addition by 1 operations as

$$\tilde{H}(i) = \exp(H(i)) - 1 = \exp(a_K i^K + a_{K-1} i^{K-1} + \cdots + a_1 i + a_0) - 1. \tag{20}$$

The thick contour in Figure 7 is the smoothed contour of the histogram obtained using a sixth-order polynomial fit. We see that the polynomial fit models the contours of the histogram very well. Identifying the inflexion point is now merely a question of locating the minimum value of this contour. Note that the exponentiation in Equation (20) is not necessary for locating the inflexion point, which can be located on $H(i)$ itself. Since the polynomial is defined on the indices of the histogram bins, rather than on the centres of the bins, the inflexion point gives us the index of the histogram bin within which the inflexion point lies. The centre of this bin gives us the optimum decision threshold. In histograms where the inflexion point does not represent a local minimum, other simple criteria such as higher order derivatives must be used.

4. Implementation of the segmenter

In this section we describe two implementations of the segmenter: batchmode and runtime. In the former, endpointing is done on pre-recorded signals and real-time constraints do not apply. In the latter, the endpointer must identify beginnings and ends of speech segments with minimal delay in identification, and therefore must have minimal dependence on future samples of the signal.

In both implementations, using a suitable initial feature representation, likelihood difference features are first derived for each frame of the signal. From these, averaged likelihood-difference features are computed using Equation (15). The averaging window can be either symmetric (i.e. $K_1 = K_2$ in Equation (15)) or asymmetric ($K_1 \neq K_2$), depending on the implementation. The window length $(K_1 + K_2 + 1)$ is typically 40–50 frames. Its shape can differ, but we have found rectangular or Banning windows to be particularly effective. Rectangular windows have been observed to be more effective when inter-speech silences are long, whereas the Hanning window is more effective when shorter gaps are expected. The resulting sequence of averaged likelihood differences is used for endpoint detection.

Each frame is now classified as speech or non-speech by comparing its averaged likelihood-difference against a threshold that is specific to the frame. The threshold for any frame is obtained from a histogram computed over a segment of the signal spanning several 1000 frames that includes that frame. The exact placement of this segment is dependent on the type of implementation. Once all frames are tagged as speech or non-speech, contiguous segments of speech that lie within a small number of frames of each other are merged. Speech segments that are shorter than 10 MS are discarded. Finally, all speech segments are padded at the beginning and the end by about half the size of the averaging window.

4.1. Batchmode implementation

In batchmode implementation the entire signal is available for processing. Data from both, the past and the future of any segment of the signal can be used when classifying that segment. In this case the main goal is to extract entire utterances of speech from the continuous signal. Here the window used to obtain the averaged likelihood difference is a symmetric rectangular window, about 50 frames wide. The histogram used to compute the threshold for any frame is derived from a segment of signal *centered* around that frame. The length of this segment is about 50 s when the background noise conditions are expected to be reasonably stationary, and shorter otherwise. We have found segment lengths shorter than 30 s to be inadequate. Merging of adjacent segments and padding of speech segments on either side is performed after the classification as a post-processing step.

4.2. Runtime implementation

Runtime implementation is aimed at applications that require an endpoint detector for continuous listening. In such situations one cannot afford delays of more than a fraction of a second before determining whether the incoming signal is within a speech segment or not. The various parameters of the segmenter must be suitably adapted to the situation. For runtime implementation the averaging window is asymmetric, but remains 40–50 frames wide. The weighting function is also asymmetric. An example of a function that we have found to be effective is one constructed using two unequal Hanning windows. The lead portion of the window, that covers frames to the future of the current frame, is half of an 8 frame wide Hanning window and covers four frames. The lag portion of the window, that applies to frames from the past, is the initial half of a 70–90 frame wide Hanning window, and covers between 35 and 45 frames. We note here that any similar skewed window may be applied.

The histogram used for determining decision thresholds for any frame is computed from the 30 to 50 s long segment of the signal immediately prior to, and including, the current frame. When the first frame that is classified as speech is located, the beginning of a speech segment is marked as having begun half a (averaging) window size number of frames prior to it. The end of a speech segment is marked at the halfway point of the first window size length sequence of non-speech frames following a speech frame.

5. Experimental results

In this section we describe a set of experiments which we conducted to test the endpointing/segmentation strategy proposed in this paper. Both batchmode and runtime-type implementations were evaluated using the CMU SPHINX speech recognition system. We first describe the databases that we chose to use for our experimentation. We then describe the features computed and other implementation details that were specific to the experiments. Finally, we present the experimental results and our conclusions.

5.1. Databases used

The databases chosen for experiments were the "Speech in noisy environments" SPINE1 and SPINE2 databases provided by LDC (Linguistic Data Consortium, 2001). These databases were used by the Naval Research Labs in the years 2000 and 2001 as training, development and test data for the first and second SPINE evaluations (SPINE, 2001). The databases contain speech recorded over a variety of moderate to severe environmental noise conditions of the type encountered in real military environments, such as tanks, helicopters, aircraft carriers, military vehicles, airplanes, etc. The speaking styles are also characteristic of those of military personnel communicating in highly unpredictable and reaction-provoking situations involving tracking and sighting of targets, ambush and evasion. The speech forms are highly spontaneous, with laughter, shouting, screaming, etc. Additionally, in SPINE1 data, several recordings have multiple energy levels introduced due to a combination of continuous recording and push-button activation of the communication channel. When the channel is not open the recording equipment records only equipment noise. When the button is pushed to open the communication channel, channel and background noises, sometimes at high ambient levels, begin to get recorded. Multiple utterances with inter-utterance silences can be recorded within one push-button episode. Energy based segmentation is especially difficult in such cases since the energy level trajectory at the sudden onset of the second level of background noise mimics the trajectories expected when speech sets in (Singh *et al.*, 2001).

The SPINE2 database is in general noisier that the SPINE1 database. The lower panel in Figure 9 shows a typical noisy utterance from the SPINE2 evaluation database. The estimated SNR of this signal is approximately 0 dB. The SNR is low enough to obscure some of the episodes of speech completely. All SPINE1 recordings are 16 KHz sampled wideband speech. The SPINE2 database has two components: 16 KHz wideband speech and coded speech that has been obtained by low pass filtering the signals from the first component to telephone bandwidth and passing them through one of several codecs. Specifically, the MELP, CELP, CVSD and LPC codecs (DDVPC, 2001) have been used in the evaluation data. The codecs introduce high levels of distortion that degrade the signal, making it more difficult both to locate the endpoints of the data and to recognize it. The endpointing algorithm presented in this paper was evaluated against both the wideband and the coded narrowband components of the SPINE2 evaluation data.

5.2. Feature representation and implementation details

All signals were windowed into 25 MS frames, where adjacent frames overlapped by 15 MS. For wideband data a bank of 40 Mel filters covering the frequency range 130–6800 Hz was used to derive a 40-dimensional log-spectral vector for each frame. For coded speech a bank of 32 Mel filters covering the range 200–3400 Hz was used to derive 32-dimensional log-spectral vectors. The Mel-frequency log spectral vectors were then projected down to a 13-dimensional feature vector using the Karhunen Loeve Transform (KLT). The eigenvectors for the KLT were computed from the log-spectral vectors of clean (office and quiet environment) components of the SPINE2 training corpus.

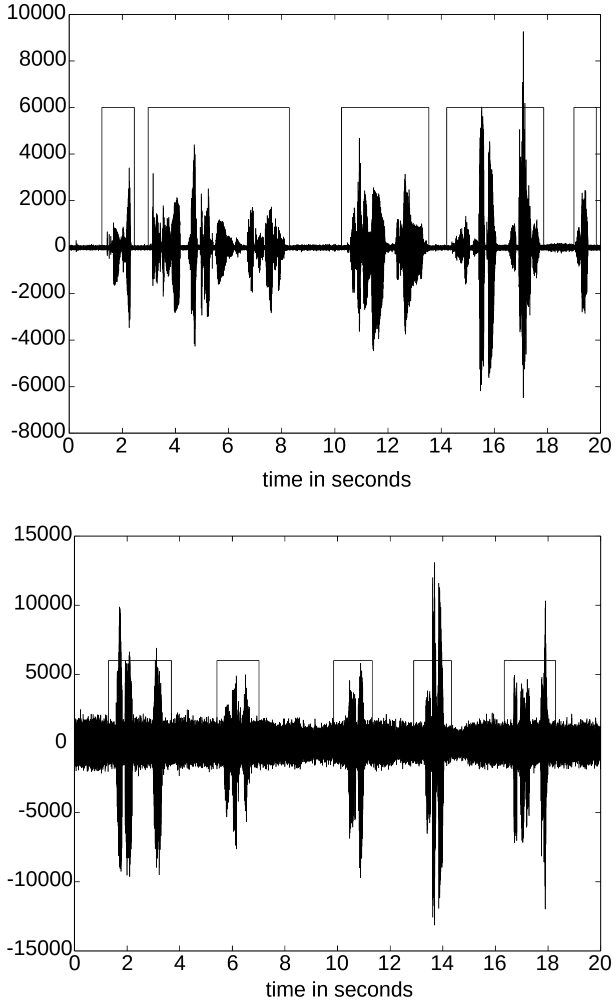

Figure 9. Two examples of segmentations computed by the proposed segmenter. The upper panel shows a relatively clean signal with a global SNR of 20 dB, recorded in an office environment. The lower panel shows a segment of signal with a global SNR of about 0 dB. This signal was recorded in the presence of background operating noise from a Bradley military vehicle.

For the coded data they were computed from low-pass filtered and down-sampled versions of the same dataset (uncoded speech). The 13-dimensional KLT-based feature vector for every frame was augmented by a difference feature vector computed as the difference between the feature vectors of the subsequent and previous frames. The final feature vector for any frame thus had 26 components.

The 26-dimensional features were then projected down to a 2-dimensional likelihood space using distributions of speech and non-speech estimated from the training corpus. For all experiments with wide-band data, the non-speech distribution used for the likelihood-based projection was trained on several different noise types from the SPINE2 training data. The speech distribution was trained with speech segments from clean (office environments) recordings in the SPINE2 training data. It was empirically observed that training the speech distributions on purely speech data (i.e. without within-utterance silences) resulted in the best segmentation performance. The speech used to train the speech distributions was therefore selected by force-aligning the training data to their transcripts using a recognizer and excising all identified within-utterance silences. For experiments with coded data, separate distributions were trained for each type of codec. For each codec the speech and non-speech distributions were trained with clean speech and noise segments from the SPINE2 training corpus that had been coded using the same codec. All distributions were modelled as mixtures of 32 Gaussians, the parameters of which were computed using the EM algorithm. Mixtures of 32 Gaussians were found to be optimal for the task.

Likelihood-difference features were then computed by subtracting the likelihood of non-speech from that of speech and windowing and averaging them using Equation (15) to result in the final averaged likelihood difference feature. For experiments with the batchmode implementation, a symmetric rectangular window 50 frames wide was used for this purpose. For the runtime implementation the asymmetric window describe in Subsection 4.2 was used.

The classification threshold for any frame was estimated from histograms that were computed from 60 s segments of speech centered at that frame, in the batchmode implementation. For frames near the beginning or end of any recording, the first 60 or the last 60 s of the recording were used. For the runtime implementation of the segmenter, the classification threshold for any frame was found using a histogram computed from the 50 s of speech immediately preceding and including the current frame. For frames within 50 s of the beginning of any recording all frames from the beginning until the current frame were used to compute the histogram. For the first 15 s of any recording there were insufficient frames to compute proper histograms, and therefore a default threshold of 0 was used. Figure 9 shows two examples of segmentations obtained using the batchmode segmenters for wideband speech. The segmenter is observed to have accurately captured speech segments in both the noisy and clean signals.

5.3. Results

Table I shows the accuracy with which frames have been identified as speech or non-speech for both, the wideband speech and the coded speech. These accuracies have been measured on a per-frame basis. The reference tags in this case were obtained from the

TABLE I. Classification accuracy of speech and non-speech frames using averaged likelihood difference features and Gaussian-fit based classification threshold estimates

Data type	Segmentation type	Classification accuracy for speech (%)	Classification accuracy for non-speech (%)
Wideband	Batchmode	91.6	90.4
Wideband	Runtime	92.1	86.7
Coded	Batchmode	92.7	82.0

Only raw classification accuracy is reported, and the effect of merging of close segments or deletion of short segments has not been taken into account.

manual endpoints, and all frames within an utterance of speech were tagged as speech. Thus, any within-utterance silence frame, even when correctly identified by the classifier as silence, is counted as an error. The accuracies reported in Table I are therefore lower than the true accuracies. Classification accuracy is seen to be better for batchmode implementation than for runtime implementation, and better for wideband speech than coded speech. The classification accuracy of speech is generally higher than that of non-speech. Many of the classification errors do not result in segmentation errors since they are either misclassifications of isolated or short segments of speech frames within utterances, which get retagged as speech when segments are merged, or similar segments of silence which, although tagged as speech by the classifier, get discarded due to the short duration of the segments. On the whole, frame-level classification accuracy is not fully indicative of the recognition accuracy to be obtained with the segmenter.

The performance of the endpointing algorithm is better measured in terms of the recognition accuracy obtained using its output. Recognition performance is dependent on obtaining complete speech segments, rather than accurate frame-level classification. Table II shows recognition error rates obtained using the various modes of the segmenter on the SPINE1 data. Tables III and IV show similar results for the SPINE2 data. The first row in all tables shows the recognition performance obtained with manually marked endpoints. For the wideband speech, the performances obtained with an energy-based segmenter and a simple classifier-based segmenter are also shown. The energy-based segmenter was based on the algorithm described by Lamel *et al.* (Lamel *et al.*, 1981). The classifier-based segmentation was performed by classifying signal frames directly using the distributions used for the projection and *a posteriori* probabilities for the probabilities that were estimated on a set of held out data. Segments were merged using the same criteria that were used by the proposed likelihood-projection based segmenter.

The column labelled "Mode" in the tables refers to the specific implementation of the segmenter (either batchmode or runtime). The "Threshold" column refers to the method of identifying classification thresholds. Segmentation errors introduce two types of recognition errors. The first, called *gap insertions*, are spurious words hypothesized by the recognizer in non-speech regions, which have been wrongly tagged as speech by the segmenter. The second are errors that occur due to deletions of speech by the segmenter. The final column in Tables II–IV show gap insertions. Not surprisingly, there are no gap insertions when endpoints are manually tagged. Differences between the error rates with manual and automatic endpointing, which are not accounted for by gap insertions, are largely due to deletions by the segmenter. In all test sets, the proposed endpointing

TABLE II. Recognition accuracy obtained on wideband SPINE1 evaluation data that have been segmented using several different methods

Segmenter	Mode	Threshold	Error rate (%)	Gap insertions (%)
Manual endpoints			27.9	0
Proposed algorithm	Batchmode	Gaussian	29.2	0.6
		Polynomial	29.5	0.9
	Runtime		30.0	1.3
Energy-based			35.7	7.2
Classifier-based			37.1	9.1

Gap insertions reflect errors introduced due to spurious non-speech segments that have been identified as speech and recognized.

TABLE III. Recognition accuracy obtained on wideband SPINE2 evaluation data that have been segmented using several different methods

Segmenter	Mode	Threshold	Error rate (%)	Gap insertions (%)
Manual endpoints			47.4	0
Proposed algorithm	Batchmode	Gaussian	48.4	0.7
		Polynomial	48.0	0.7
	Runtime		49.0	0.8
Energy-based			55.7	8.1
Classifier-based			57.9	10.5

Gap insertions reflect errors introduced due to spurious non-speech segments that have been identified as speech and recognized.

TABLE IV. Recognition accuracy obtained on coded SPINE2 evaluation data

Segmenter	Mode	Threshold	Error rate (%)	Gap insertions (%)
Manual endpoints			55.7	0
Proposed algorithm	Batchmode	Polynomial	57.9	1.9

Gap insertions reflect errors introduced due to spurious non-speech segments that have been identified as speech and recognized.

algorithm is observed to outperform both energy-based endpointing and classifier-based endpointing by a large margin. Even on coded speech, the performance of the endpointer is very close to that obtained with manually-tagged endpoints, although the overall recognition accuracy is rather worse than that of wideband speech.

From Tables II and III we observe that while polynomial-based threshold detection performs better on the SPINE2 set, Gaussian-based threshold detection is superior for the SPINE1 set. The reason for this is that the SPINE1 data were less matched to the distributions used for the likelihood projection (which were trained with SPINE2 data) and so the distribution of averaged likelihood-differences frequently did not exhibit clear local minima at the points of inflexion (e.g. Figure 8). Here, as expected, the Gaussian based threshold detection mechanism was better able to locate thresholds. For the SPINE2 data, histograms typically showed clear local minima between modes. Here, (e.g. Figure 7) Gaussian-based threshold detection tended to overestimate the threshold value (thereby classifying more non-speech frames as speech) and the polynomial based method performed better. Gap insertions remained few in all cases. We note that the gap insertion percentage is sometimes larger than the difference between the performances obtained with automatic and manual endpoints. This is because the automatically obtained endpoints sometimes resulted in slightly fewer recognition errors in true speech segments than the manually determined endpoints. This is an artifact of the manner in which recognition is performed in the SPHINX, which expects a short amount of silence at beginnings and ends of utterances.

In Tables II and III, the runtime segmenter is observed to be slightly worse than the batchmode segmenter in all cases. However, the degradation from batchmode to runtime segmentation is not large. During our experiments we also observed that the segmenter performed better under conditions of mismatch between projecting distributions and the data when speech distributions were computed using clean speech, rather than an assortment of noisy speech. Finally, as measured in our experiments with the SPINE data, segmentation with histogram-based threshold estimation takes approximately 0.035 times real time on a 1 GHz Pentium-III processor with 512 megabytes of RAM. This excludes the time taken for computing the KLT-based features, which takes about 0.02 times realtime, but is shared with the recognizer which uses the same features. Segmentation using Gaussian-based threshold detection took about 0.025 times realtime in our experiments. It must be noted that these numbers are, however, functions of the width of the averaging window and the length of the segment used for computing histograms.

6. Discussions and conclusions

In this paper we have proposed an algorithm for endpointing of speech signals in a continuously recorded signal stream. The segmentation is performed using a combination of classification and clustering techniques by using classifier distributions to project data into a space where clustering techniques can be applied effectively to separate speech and non-speech events. In order to enable effective clustering, the separation between classes is improved by an averaging operation. The performance of the algorithm is shown to be almost comparable to that obtained with manually obtained segmentation in moderate and highly noisy speech, as demonstrated by our experiments on the noisy SPINE databases. We note here that a variant of the proposed segmentation algorithm was used by Carnegie Mellon University in both the SPINE1 and SPINE2 evaluations, where its overall performance was best amongst all sites that evaluated on a common platform.

It must be noted here that the performance obtainable with likelihood-based projections is not completely independent of the match between the classifier distributions and the data. As mentioned earlier, optimal discrimination is only possible in the likelihood space if the distributions used are the true distributions of the classes. As the distributions used for the projections deviate from the true distributions, such guarantees are no longer valid. In practice, the result of such mismatches is that the modes in the distribution of the likelihood-differences begin to merge, making it increasingly difficult to classify speech frames accurately. Thus it is important, as far as is possible, to attempt to minimize the mismatch between the distributions used for projection and the data. Adapting the distributions to the data may provide some improvement in the performance. However, it must be noted here that even without adaptation, the endpointer performs extremely well in conditions of small to medium mismatch, such as that between the SPINE2 and SPINE1 data, which differ greatly in the type and level of background noise.

The current implementation of the segmenter does not utilize any *a priori* knowledge of the dynamics of the speech signal. Every frame is classified independently of every other frame. We know, for instance, that the performance of a static classifier-based segmenter can be significantly improved by using dynamic statistical models such as HMMs for the classes instead of static models (e.g. Acero *et al.*, 1993). We expect that similar improvements can be achieved in the proposed segmenter as well by including contextual information through a dynamic model, such as an HMM.

Rita Singh was sponsored by the Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

References

Acero, A., Crespo, C., De la Torre, C. & Torrecilla, J. C. (1993). Robust HMM-based endpoint detector. *Proceedings of Eurospeech'93*, pp. 1551–1554.

Chen, S. & Gopalakrishnan, P. (1998). Speaker, environment and channel change detection and clustering via the Bayesian information criterion. *Proceedings of the Broadcast News Transcription and Understanding Workshop*, pp. 127–132.

Cohen, J. R. (1989). Application of an auditory model to speech recognition. *Journal of the Acoustical Society of America* **85(6)**, 2623–2629.

Compernolle, D. V. (1989). Noise adaptation in hidden Markov model speech recognition systems. *Computer Speech and Language* **3(2)**, 151–168.

DDVPC (2001). DoD digital voice processing consortium. Available as http://www.plh.af.mil/ddvpc/index.html.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of Royal Statistical Society Series B* **39**, 1–38.

Doh, S. -J. (2000). *Enhancements to transformation-based speaker adaptation: principal component and interclass maximum likelihood linear regression*. PhD Thesis, Carnegie Mellon University.

Duda, R. O., Hart, P. E. & Stork, D. G. (2000). *Pattern Classification*, 2nd edition, Wiley, New York.

Hain, T. & Woodland, P. C. (1998). Segmentation and classification of broadcast news audio. *Proceedings of the International Conference on Speech and Language Processing ICSLP98*, pp. 2727–2730.

Hamada, M., Takizawa, Y. & Norimatsu, T. (1990). A noise-robust speech recognition system. *Proceedings of the International Conference on Speech and Language Processing ICSLP90*, pp. 893–896.

Hirsch, H. G. (1993). Estimation of noise spectrum and its application to SNR estimation and speech enhancement. Tech. Report TR-93-012, International Computer Science Institute, Berkeley, CA.

Junqua, J.-C., Mak, B. & Reaves, B. (1994). A robust algorithm for word boundary detection in the presence of noise. *IEEE Transactions on Speech and Audio Processing* **2(3)**, 406–412.

Lamel, L., Rabiner, L. R., Rosenberg, A. & Wilpon, J. (1981). An improved endpoint detector for isolated word recognition. *IEEE ASSP Magazine* **29**, 777–785.

Lasry, M. J. & Stern, R. M. (1984). *A posteriori* estimation of correlated jointly Gaussian mean vectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 530–535.

Leggetter, C. J., & Woodland, P. C. (1994). Speaker adaptation of HMMs using linear regression. Tech. report CUED/F-INFENG/TR. 181, Cambridge University.

Linguistic Data Consortium (2001). Speech in noisy environments (SPINE) audio corpora. LDC Catalog numbers LDC2001S04, LDC2001S06 and LDC2001S99.

Rabiner, L. R. & Sambur, M. R. (1975). An algorithm for determining the endpoints of isolated utterances. *Bell Systems & Technical Journal* **54(2)**, 297–315.

Rabiner, M. R. & Juang, B. H. (1993). *Fundamentals of Speech Recognition. Prentice-Hall Signal Processing Series*, Prentice-Hall, Englewood Cliffs, NJ.

Siegler, M., Jain, U., Raj, B. & Stern, R. M. (1997). Automatic segmentation, classification and clustering of broadcast news audio. *Proceedings of the DARPA ASR Workshop*, pp. 97–99.

Singh, R., & Raj, B. (2002). Likelihood as a non-linear discriminant projection. Under review.

Singh, R., Seltzer, M. L., Raj, B., & Stern, R. M. (2001). Speech in noisy environments: robust automatic segmentation, feature extraction, and hypothesis combination. *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing ICASSP2000*.

SPINE (2001). Available as http://eleazar.itd.nrl.navy.mil/spine/spinel/index.html.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 260–269.

(Received 2 January 2002 and accepted for publication 12 May 2002)