Time-Based Clustering for Phonetic Segmentation

Brian Eberman and William Goldenthal
email: bse@crl.dec.com, thal@crl.dec.com

Digital Equipment Corporation
Cambridge Research Lab

ABSTRACT

This paper describes a approach to speech segmentation. Unlike approaches based on spectral measurements, our algorithm iteratively clusters on an LPC representation of time waveform blocks. The algorithm uses a generalized maximum likelihood criterion for deciding when two neighboring pieces of the signal should be joined. This paper describes the algorithm and shows that it yields superior results when compared to metrics based on spectral or cepstral measurements.

1. Background

Segmentation of speech into phonetic components significantly accelerates search in segment-based speech systems. Given a list of possible segment boundaries, the recognizer need only hypothesize a subset of all possible start and end times for any phonetic segment. This leads to a substantial reduction in computation, even with high insertion rates, because the segmental search is $O(n^2)$ in the number of boundaries. However, to realize this gain in performance, the segmentation must be done computationally efficiently and without much prior knowledge.

There are three main techniques that have been explored in the past for segmentation. The first technique applies a sequential likelihood ratio test directly to the time series [1, 2]. Since the test is not symmetric, good performance is only obtained by testing the signal for changes both forward and backward in time. Andre-Obrecht obtained an insertion rate of 120% and a deletion rate of 2.8% on a test set of 1534 French phonemes. This work was later extended in [4] and [5], both of which eliminated the need for a backward component of the test. Unfortunately this approach is very computationally intensive because it works directly on each individual signal measurement.

Another approach to segmentation is to use a smooth derivative operator, such as the Canny edge detector [3], on the signal's spectral representation. However, Glass [6] found that a clustering technique provides superior performance. Glass used a clustering algorithm on the outputs of Seneff's auditory model. The reported insertion and deletion rates of 5% and 3.5% respectively, are computed over the optimal alignment between a *multi-scale* dendrogram and the known phonetic transcription. This optimizing search significantly improves the results that would otherwise be achieved if the clusters obtained at any single level in the dendrogram are used as the segmentation. In section 3. we report on experiments we conducted using this technique that permit a direct comparison.

2. Cluster Based Segmentation

We implemented a single clustering algorithm and experimented with several distance metrics including those used in [6]. The clustering algorithm requires two things:

1. An appropriate representation of the speech signal for clustering.
2. A distance metric for computing the distance between two clusters.

The clustering algorithm works as follows. Let $\{y_i\}$ be a sequence of n observation vectors. These could be the time series measurements, spectral measurement vectors, or cepstral measurement vectors. Divide this sequence of observations into pieces to form an initial collection of observation blocks. The sequence of observation blocks forms the initial cluster sequence $\{C_i\}$.

Let $d(C_i, C_j)$ be a distance function between two clusters C_i and C_j. The distance is calculated using some sufficient statistic computed from the underlying collection of observation vectors. The first step in the algorithm is to compute the distance between successive pairs of clusters C_i and C_j. After this initialization step, clustering proceeds.

For each clustering step, select the smallest distance between two neighboring clusters. Let the first of these clusters be denoted C_i. C_i is then merged with C_{i+1} forming a new cluster C_i'. This new cluster now takes the place of C_i and C_{i+1}. To continue clustering, the distance information is then updated. Clustering proceeds until the minimum distance between two clusters is greater than a threshold T.

For evaluation purposes, we used the 6300 utterances in the TIMIT [7] database. A segmentation boundary was considered to match a hand segmented TIMIT boundary if the hypothesized boundary was within 25 ms of the TIMIT boundary. Unmatched TIMIT boundaries were declared to be deletions, and unmatched segmentation boundaries were insertions.

3. Results for Spectral Techniques

The clustering algorithm was written and tested on mel-frequency cepstral coefficients (MFCC), mel-frequency log power spectral coefficients (MFSC), and the raw time series. All the experiments were performed using the TIMIT database. The TIMIT speech signal is sampled at 16 kHz. We used a frame rate of 5 ms, a FFT length of 256 measurements, a Hamming window of 25.6 ms, 41 mel-frequency spectral filters, and 20 MFCC.

For the MFCC and MFSC measurements two distance metrics were used. The first is a normalized distance and the second is a weighted Euclidean distance. The weighted Euclidean distance between the logarithms' of the MFSC was found to have the best performance.

The normalized distance between two measurement vectors x and y is

$$d(x,y) = \frac{x^T y}{|x||y|}.$$

The weighted Euclidean distance between two vectors is the two norm

$$d_W(x,y) = \sqrt{((x-y)^T W (x-y))}$$

where W is a weighting matrix. We experimented only with a diagonal weighting matrix.

Two clusters are merged by taking the weighted average of the features for each cluster. The features are weighted by the number of measurements of the feature in each cluster.

The results of the two different distance metrics on the MFSC and MFCC are shown in Figure 1. Both percentages are obtained by dividing the insertion and deletion counts by the number of TIMIT segments. Thus an insertion percentage of 100 percent means that twice as may boundaries were generated as actually appeared in the TIMIT database.

As can be seen from Figure 1 the Euclidean distance between the MFSC weighted by their variance performs the best. Although this is only marginally better than the normalized distances between the MFSC. We also performed experiments where the cluster distance was the maximum of the neighboring block distance and the previous cluster distance. Although essential for producing useful dendrograms [6], this refinement had little effect on our segmentation performance.

We believe that all of the metrics give similar results because they all utilize the same underlying information. Therefore,

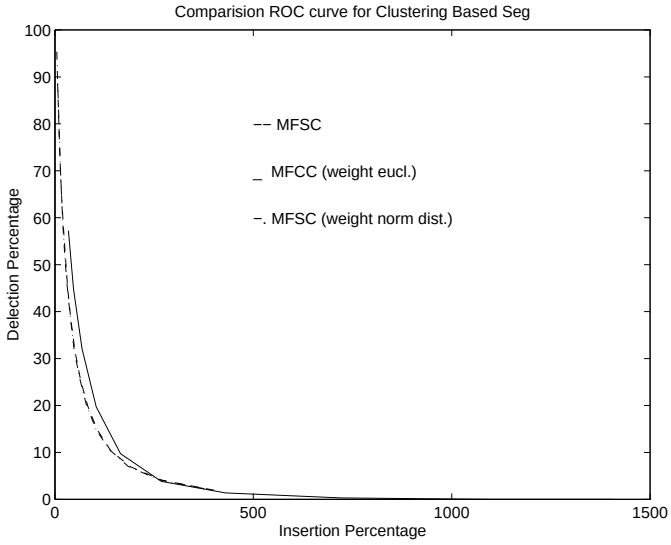

Figure 1: Performance results for MFSC and MFCC input signals using cluster based segmentation.

to achieve better performance more information is needed. This information can be obtained either by using prior information about the expected measurements for each type of phonetic unit, or extracting more information from the signal. The first case involves computationally expensive analysis which is normally conducted during search, and also gets away from the pure segmentation problem. Therefore, we chose to use the original sampled waveform.

4. LPC Based MDL Distance

A second clustering metric was designed based on LPC models of short time-series blocks. This approach combines the clustering algorithm described above with the maximum likelihood metric used by Andre-Obrecht. The speech signal is assembled into 5 ms blocks. The measurements in each block are assumed to come from an LPC model. The cluster distance is the likelihood ratio between the null hypothesis that two adjacent blocks are independent and the hypothesis that the combined blocks form a single stationary unit.

The standard LPC model is

$$y(n) = \sum_{i=1}^{q} a_i y(n-i) + \nu(n)$$

where $\nu(n)$ is a white, zero-mean, Gaussian process with variance V. The likelihood of a sequence of measurements y_1^r is the likelihood of the associated residual given the parameters $\theta = (\{a_i\}, V)$

$$L(y_1^r|\theta) = \prod_{t=1}^{t=r} p(\nu(t)|y_{t-q}^{t-1}, \theta)$$

If the two clusters consist of the measurement blocks $C_1 = y_1^r$ and $C_2 = y_{r+1}^n$ respectively, the cluster distance is the generalized likelihood ratio:

$$d(C_1, C_2) = \max_{\theta', \theta_1, \theta_2} \frac{L(y_1^n|\theta')}{L(y_1^r|\theta_1)L(y_{r+1}^n|\theta_2)}.$$

Since the LPC parameters and residual variance are unknown, they must be estimated. This can be done by maximizing each individual likelihood. In addition, we used a minimum description length (MDL) likelihood to select the optimal number of LPC parameters for each cluster. The cluster distance is then finally the distance computed based on the parameter estimates.

In our implementation, we used PARCOR coefficients instead of LPC's. If we use $\tilde{q}$ unnormalized PARCOR coefficients, the residual energy in the measurements will be

$$E_{\tilde{q}} = E - \sum_{i=1}^{\tilde{q}} \tilde{k}(i)^2,$$

where E is the energy in the block of measurements, and $\tilde{k}(i)$ are the un-normalized PARCOR coefficients. This energy is an estimate of the residual variance $V_{\tilde{q}} = E_{\tilde{q}}$. Using this estimate the log-likelihood of a vector of measurements y_1^r can be shown to be

$$l(y_1^r) = \max_{\tilde{k}} \frac{-r}{2}(\log(2\pi) + \log(V) + 1).$$

Now if we maximize l over the number of PARCOR coefficients, the result will be that the optimal number of parameters is always the maximum number of allowed PARCOR coefficients. This classical problem of over-fitting is prone to producing poor coefficient estimates which results in too large a likelihood for the data. The solution is to add a penalty for the number of coefficients.

It can be shown that the description length is the asymptotically optimal penalty to add [9]. The MDL of y_1^r is a penalized log-likelihood

$$M(y_1^r) = \max_{q, \tilde{k}} l(y_1^r|\tilde{k}) - \frac{bq}{2}\log(r)$$

where q is the number of PARCOR parameters estimated and b is the coding cost of a single parameter. A coding cost of 1 is asymptotically optimal, but for short measurement vectors a higher coding cost gives better results. With the MDL metric, the cluster distance is

$$d(C_1, C_2) = M(y_1^r) + M(y_{r+1}^n) - M(y_1^n).$$

Each MDL calculation involves determining both the optimal number of PARCOR coefficients and their values. Cluster merging is done by weighted averaging of the sufficient statistics used in the PARCOR calculation.

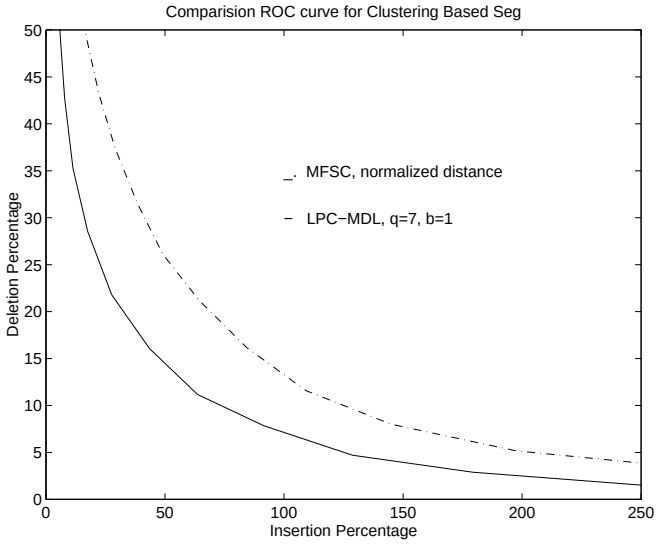

Figure 2: Performance results for MDL time-based clustering versus the normalized distance based clustering for MFSC. The MDL approach yields significantly better results than the spectral approach.

The complexity of the approach is linear in the number of initial blocks and is therefore very fast. The global minimum MDL segmentation can be found using dynamic programming in $O(n^2)$ in the number of blocks, however we found that it did not improve our results.

5. Time Series Clustering: Results

The performance of this approach is compared to our best spectral based results in Figure 2. The figure shows that the technique achieves roughly half the deletion rate for the same insertion rate as the techniques based on spectral measurements. The system achieves a deletion rate of 6.2% at an insertion rate of 100%.

Finally, in order to incorporate the results of this work into a segment based search engine it is important to know which phonetic units and phonetic pairs are most likely to receive insertions and deletions respectively. The set of phonetic units used for evaluation was based on Lee's 39 classes [8]. The performance was evaluated at a deletion rate of 6.2%. The silence category was further broken down into silence (h#, pau, epi), voiced closure (vcl), unvoiced closure (ucl), and glottal stop (q), for a total of 42 categories.

Approximately 33 % of all insertions occur in silence. Table 1 shows the probability of an insertion for each phonetic class. For example, a value of 2 for *sil* implies that a *sil* segment has two insertions on average. A characteristic of the segmenter which we found to be interesting was a tendency to insert an extra boundary in a stop release. The boundary is inserted after the plosive portion and prior to the following

aa	0.264	*ae*	0.345	*ah*	0.358	*aw*	0.407
ay	0.436	*er*	0.466	*eh*	0.487	*ey*	0.497
ih	0.541	*iy*	0.628	*ow*	0.631	*oy*	0.642
uh	0.657	*uw*	0.658	*r*	0.659	*l*	0.665
w	0.771	*y*	0.821	*m*	0.846	*n*	0.877
ng	0.929	*s*	0.948	*z*	0.977	*sh*	0.977
f	1.000	*th*	1.053	*v*	1.203	*dh*	1.269
dx	1.386	*t*	1.392	*d*	1.446	*p*	1.462
b	1.595	*k*	1.638	*g*	1.696	*ch*	1.760
jh	1.780	*h#*	1.817	*sil*	1.979	*vcl*	2.114
ucl	2.141	*q*	2.252				

Table 1: Insertion rate for each phonetic class. Note that the rate of insertions is highest for each element of Lee's original "silence" class, which we broke down to be silence (*h#*, *pau*, *epi*), the voiced (*vcl*) and unvoiced closures (*ucl*), and glottal stop (*q*).

jh	0.001	*ch*	0.001	*sh*	0.002
th	0.002	*s*	0.003	*z*	0.003
f	0.003	*uh*	0.003	*oy*	0.004
ae	0.005	*aw*	0.005	*sil*	0.007
k	0.009	*g*	0.009	*p*	0.009
t	0.009	*h#*	0.010	*ey*	0.011
v	0.012	*ng*	0.014	*ow*	0.018
uw	0.019	*ay*	0.020	*vcl*	0.021
m	0.022	*ucl*	0.023	*dh*	0.023
eh	0.028	*er*	0.030	*b*	0.030
d	0.033	*dx*	0.033	*w*	0.034
ah	0.035	*y*	0.037	*q*	0.038
ih	0.045	*iy*	0.046	*n*	0.072
l	0.085	*aa*	0.088	*r*	0.099

Table 2: Percentage of all deletions accounted for by each phonetic unit in a right context.

release. This effect is attributable to the fact that there is often significant spectral change between these two regions.

During search, the incremental cost of an extra boundary is proportional to $1/n$ (the number of boundaries) which is relatively cheap. However, deletions are more critical because they can cause misrecognition errors extending for several words. The fraction of deletions accounted for when each class is in the "right" context is shown in Table 2. The probability of deletion can be calculated from this table by dividing each class by it's prior probability. Note that the transitions to vowels and semi-vowels (liquids) dominate. This is to be expected, since vowel semi-vowel transitions are spectrally gradual. This problem could be significantly alleviated by creating multi-phonetic units such as [*aa*][*r*], [*ow*][*l*], etc. We noted that the highest 50 left-right transitions for which deletions occur account for 50 percent of the total number of deletions.

6. Conclusions

We found that a maximum likelihood metric for an LPC model of the measurements gave much better results than any of several metrics based on spectral or cepstral measurements. Although not directly confirmed by experiment, the most likely reason for this increase in performance is the elimination of the spectral smoothing introduced in the spectral decomposition to get good spectral estimates. The spectral processing all involves a windowing operation over a relatively long window. This smoothes the spectral estimates producing better measurements, however it also decreases the spectral changes. In contrast, LPC models can be accurately estimated on short signals. Furthermore the LPC statistics of clusters can be combined to exactly produce the LPC statistics of the combined block.

7. REFERENCES

1. R. Andre-Obrecht. Automatic segmentation of continuous speech signals. In *ICASSP 86*, vol. 3, pp 2275–2278, Apr. 1986.
2. R. Andre-Obrecht. A new statistical approach for the automatic segmentation of continuous speech signals. *IEEE ASSP*, 36(1):29–40, Jan. 1988.
3. J. F. Canny. Finding edges and lines in images. TR 720, MIT A.I. Lab., 1983.
4. G. Feng, N. Achab, and P. Combescure. On-line speech segmentation using adaptive models: application to variable rate speech coding. In *EUROSPEECH 91*, vol. 4, pp 705–708, Sept. 1991.
5. R. J. D. Francesco. Real-time speech segmentation using pitch and convexity jump models: application to variable rate speech coding. *IEEE ASSP*, 38(5):741–748, May 1990.
6. J. R. Glass. *Finding Acoustiv Regularities in Speech: Applications to Phonetic Recognition*. PhD, MIT, EECS, May 1988.
7. L. Lamel, R. Kassel, and S. Senef. Speech database development: design and analysis of the acoustic-phonetic corpus. In *DARPA Speech Recognition Workshop SAIC-86/1546*, pp 100–109. Palo Alto, CA, Feb. 1986.
8. K. F. Lee. *Large–Vocabulary Speaker–Independent Continuous Speech Recognition: The SPHINX System*. PhD, CS, CMU, 1988.
9. J. Rissanen. *Stochastic Complexity in Statistical Inquiry*, vol. 15 of *Series in Computer Science*. World Scientific, London, Eng., 1989.