Improving SVF with DISTBIC for Phoneme Segmentation

Joshua Winebarger, Khalid Daoudi, Hussein Yahia

Improving SVF with DISTBIC for Phoneme Segmentation

Joshua Winebarger, Khalid Daoudi, Hussein Yahia
INRIA (GEOSTAT team)
http://geostat.bordeaux.inria.fr

Abstract

In this paper we examine an application for phoneme segmentation of DISTBIC, a two-pass, text-independent method traditionally used for speaker segmentation. The novelty of this paper is its experimentation with use of the spectral variation function (SVF), a simple non-parametric method for phone segmentation, as a replacement for the distance measure of the first pass of DISTBIC. In doing so we aim to produce a computationally efficient method for text-independent phoneme segmentation that provides good performance. Experiments are carried out on the TIMIT database. We give a performance comparison between the SVF as previously used for segmentation, our DISTBIC-SVF algorithm, and another state-of-the-art algorithm.

1. Introduction

Phoneme segmentation generally describes the task of automatically estimating the location of the boundaries between phonemes in speech. Given a speech signal, the application of such segmentation should produce a series of time indices corresponding to the most likely location of the transitions between phonemes. Such a segmentation is useful for speech coding, training text-to-speech systems, speech indexing or annotating spoken corpora. A well-performing automatic segmentation process is especially needed in the latter example, where corpora can reach very large sizes and the speed of a manual segmentation by expert phoneticians is over 130 times real-time [1].

Segmentation methods can be divided into two broad classes: text-dependent (TD) and text-independent (TI.) Text-dependent methods such as Hidden Markov Models (HMMs) perform segmentation based on both the speech signal and indications of the phonemes present, either through the provision of a phonemic transcriptions or the incorporation of a model trained on a corpus of manually segmented data. This may provide a high-quality segmentation, but is also the source of disadvantages, since the models require a large amount of training and are linguistically constrained in that they presuppose certain phonemes. TI methods on the other hand, avoid extensive training at the expense of performance, using only acoustic information for the detection of transitions. These methods may be useful in multi-lingual applications or situations where phonetic transcriptions are unavailable or inaccurate. [2]

There exist many varieties of TI methods [2] [3]. One classical and easy-to-implement technique, figuring in several papers as a method of finding sub-word transitions, is the use of the spectral variation function (SVF.) Methods of employing the SVF found in the literature vary: [4] used the SVF to find subword transitions with the aid of some heuristic rules using energy and zero-crossing rate. [5] used three variants of the SVF to segment telephone-quality speech. [6] incorporated the SVF as one of the features used in an HMM, whereas it was used by [7] as a means to constrain the transition of the HMM from one phoneme to another. Lastly, [2] and [3] used SVF as a stand-alone method as a reference against other phoneme segmentation methods.

While SVF has the virtue of being computationally inexpensive, it is, as with many non-parametric methods, prone to a high rate of so-called false alarms, leading to unsatisfying overall performance. In this paper we propose to apply the DISTBIC algorithm [8] to the SVF to correct for this deficiency. DISTBIC, which has been used in the past both for speaker turn detection [8] and for phoneme segmentation [3], is

a two-step process which incorporates heuristics and a penalized statistical likelihood ratio test (LLRT) to reduce error rates.

We compare the performance of our algorithm to that another state-of-the art TI method, the Microcanonical Multi-scale Formalism (MMF).

2. SVF

The SVF is a measure of magnitude of overall spectral change from frame to frame, providing a way to quantify the quasi-instantaneous spectral change in a signal with a single value. Traditionally, it is computed as an angle between two normalized cepstral vectors separated by some frame distance, these vectors being the difference between the cepstrum and its average over a multi-frame window.

One form of the SVF is found in [9], where it is employed to estimate an upper limit on the number of phone segments in a speech signal. The expression of this SVF is simply the norm of the delta-cepstral coefficients for the frame *k*:

$$SVF_{\Delta cep}(k) = \sqrt{\sum_{m=1}^{p} [\Delta C_k(m)]^2} \tag{1}$$

with C_k being the mel-frequency cepstrum (MFCC) for frame k and p being the order of the MFCC.

This tendency of the SVF toward over-segmentation, exploited by [9] to give an upper bound on the number of segments, seems to suggest it may be well suited for use with DISTBIC. We hypothesized that applying the heuristic first step of DISTBIC to SVF may provide useful candidate points for the second step of DISTBIC, which uses BIC to reject spurious segmentation candidates found in the first. We chose to use the SVF found in [9] rather than that described in [6] since it is computationally less expensive and has fewer parameters to tune such as the window size or the separation of the left and right contexts.

Applying the SVF to a signal produces a series or curve with the peaks corresponding to areas of rapid, intense spectral change. Traditional approaches to finding a segmentation with this curve identify the phoneme transitions as the minima of its second derivative, sometimes after the application of some smoothing [4]. We followed this approach without smoothing for our implementation of a segmentation with SVF.

3. DISTBIC with SVF

3.1 Distance Curve

The starting point of the DISTBIC algorithm is the production of a distance curve computed from the features extracted from the signal. This distance curve is the basis for a set of candidate transitions selected by heuristics, which are then verified with hypothesis testing. Previous works have proposed several measures for this curve, such as the generalized likelihood ratio (GLR) generated from two fixed windows sliding along the signal, or the Kullback-Leibler divergence measure. In our work, we treat the SVF as the distance curve.

Next, the heuristics examine the maxima of the distance curve and produce a set of candidate transition points. A series of rules are used to choose maxima which are more likely to correspond to phoneme

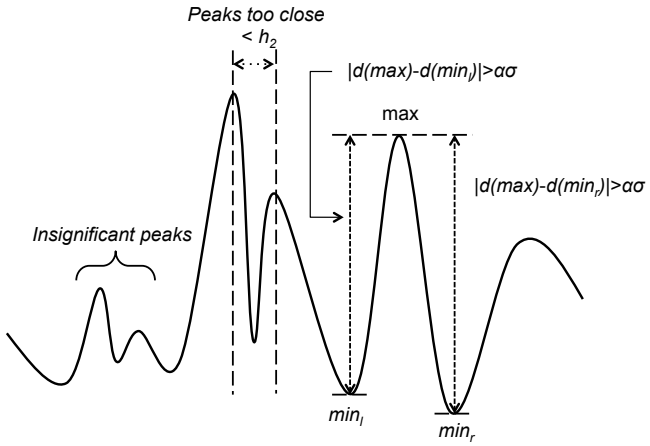

Figure 1: Example of the heuristic measures involved in the heuristics of DISTBIC

transition points. It is recommended in [8] to low-pass filter the distance curve *d(t)* prior to use of the heuristics. We do not include this step since the choice of the filter increases the space of parameters to be tuned, whereas our goal is to maintain simplicity in our system.

First, we ensure a minimum distance between candidate peaks by enforcing a minimum distance h_2 between maxima. If two maxima are separated by less than this distance, they are merged by replacing them with a point located at their averaged position and possessing their averaged magnitude.

To be chosen as a candidate, the difference between a maximum and the minima to its left and right must be greater than the standard deviation of the signal multiplied by tuning factor α:

$$|d(max) - d(min_r)| > \alpha\sigma \text{ and } |d(max) - d(min_l)| > \alpha\sigma$$

In the case of a maximum occurring at the left (resp. right) limit of the signal with no left (resp. right) minimum adjacent, we consider only the difference between the maximum and its right (resp. left) minimum.

3.2 Dynamic Windowing with BIC

The second stage of DISTBIC is a process of dynamic windowing, wherein we verify the candidate transitions of the previous step using an hypothesis test based on the ΔBIC statistic. This hypothesis test is posed as follows. We wish to know if frame number j contains a transition point. We form two subwindows of the feature vectors: $X = (x_{j-N_X}, \ldots, x_j)$ and $Y = (x_j, \ldots, x_{N_Y})$ with N_X and N_Y being the length of each window, and their concatenations denoted simply as Z. Further, we assume that the features within conform to a multi-dimensional Gaussian model. The hypothesis test is then

- H_0: $Z \sim N(\mu_Z, \Sigma_Z)$ meaning that the two windows contain features from the same phoneme
- H_1: $X \sim N(\mu_X, \Sigma_X)$ and $Y \sim N(\mu_Y, \sigma_Y)$ meaning that the windows contain two different phonemes

with μ and σ being the mean vectors and covariance matrices of the windows. The test is formulated as a maximum likelihood ratio:

$$R(j) = \frac{N_Z}{2}\log|\Sigma_Z| - \frac{N_X}{2}\log|\Sigma_X| - \frac{N_Y}{2}\log|\Sigma_Y| \quad (2)$$

The ΔBIC value is the maximum likelihood ratio penalized by model complexity: $\Delta = -R(j) + \lambda P$ where P, the penalty factor, is given as $P = \frac{1}{2}(p + \frac{1}{2}p(p+1))\log(N_Z)$ with p being the dimension of the feature vectors. λ is is a tuning parameter used to control the rejection sensitivity. If $\Delta BIC > 0$, we conclude that a transition exists.

The dynamic windowing algorithm uses this ΔBIC hypothesis test as follows. Define $Q = \{q_0, q_1, \ldots, q_N\}$ as the transitions found from the distance (SVF) and heuristics. To begin, we form three windows, X, Y, and Z as before, with the limits of X being $[q_0, q_1]$, Y being $[q_1, q_2]$, and Z being $[q_0, q_2]$. At this point in our algorithm, we again extract features (with different parameters than those used in the part of the algorithm described in section 3.1) We calculate the covariance matrices for these windows and compute the ΔBIC score. If we verify the candidate, we shift both windows at their default size, with X being $[q_1, q_2]$, Y being $[q_2, q_3]$, and Z being $[q_1, q_3]$. On the other hand, if the candidate is rejected by ΔBIC, we maintain X at $[q_0, q_1]$ and increment the right-hand limit of Y by one transition, such that its bounds are $[q_1, q_3]$, with Z again spanning both X and Y. Then we perform again a ΔBIC test. The process is repeated until we have covered the entire signal. Finally, the candidates that were verified are considered the output of the algorithm.

4. Experimental Results

4.1 Experimental Setup

The TIMIT database was used for evaluation of our algorithm. The SA sentences were excluded since they are the same for each speaker in the database [2] and serve merely to compare accents. We used a subset of three hundred randomly chosen files (approximately 10% of the 'train' portion from the database) for use in selection of the algorithm parameters yielding the best performance. Aiming for the best $F1$ score for the SVF algorithm, we tried 156 variations in the value of the parameters MFCC frame step size and frame length over the three hundred file set.

Choosing the parameters of DISTBIC-SVF for $F1$ proceeded in two steps. We performed trials of the first stage of the algorithm (SVF with heuristics), with 160 different configurations of the parameters α, MFCC frame step, and MFCC frame length. Among those which produced segmentations with HR $> 75\%$, we selected the four combinations of parameters giving the highest ratio of HR to FAR.

Parameters for the second stage of the algorithm (dynamic windowing with ΔBIC) were found in two sub-steps, with parameters λ, MFCC frame step, and MFCC frame length being tuned. We used a 20-file subset of the 300-file set to search for the 25% of parameter configurations giving the best $F1$ score. These candidates configurations were tried on the 300-file subset, with the candidate giving the best $F1$ being selected as the final configuration.

Last, the SVF and DISTBIC-SVF algorithms were evaluated on the full TIMIT database excluding the 300-file subset.

The features used were the first 12 MFCCs computed using the default settings of the Voicebox toolbox for Matlab, with the MFCC frame size and step that gave the optimal performance for each method given in section 4.3.

4.2 Performance Measures

We measured the discrepancy between the manual and automatic segmentations through several measures. Partial performance measures were used: the Hit Rate (HR), which is the ratio of correctly detected

boundaries to the number of reference transcriptions, the Oversegmentation (OS), which is the difference between the number of detected transitions and the number of reference transcriptions over the number of reference transcriptions, and the False Alarm Rate (FAR), which is the difference between the number of detections and hits over the number of detections.

$$HR = \frac{\text{hits}}{\text{transcriptions}} \quad OS = \frac{\text{detections - transcriptions}}{\text{transcriptions}} \quad FAR = \frac{\text{detections - hits}}{\text{detections}}$$

We further define the Percent Correct (PCR) as $1 - FAR$.

The composite performance measure the most often used in the literature is the $F1$ measure, a harmonic mean of PCR and HR:

$$F1 = \frac{2 \cdot PCR \cdot HR}{PCR + HR} \tag{3}$$

An automatic segmentation point was counted as a hit if it fell within 20 milliseconds of a manual segmentation point not already associated with another automatic segmentation point. In the case of two or more automatic segmentations falling within this interval, we count only the earliest one as a hit, the rest being counted as insertions.

4.3 Results

We compared the performance of our DISTBIC-SVF method to SVF as described in section 2, as well as another method based on the microcanonical multiscale formalism [10]. This latter method is a sample-based segmentation which employs computation of local geometrical parameters to yield a function termed the *ACC*, from which candidate transition points are derived. Dynamic windowing with an LLRT is then used to verify these candidate points.

The results of testing each method on the TIMIT database are presented in table 1. An MFCC frame step of 10 ms and frame length of 20 ms were found to give the best $F1$ performance for the SVF algorithm. The parameters giving the best $F1$ value for DISTBIC-SVF are given in table 2.

Table 1: Comparative table of segmentation results. Scores are percentages.

score	**MMF-LLRT**	**SVF**	**DISTBIC-SVF**
HR	72.59	66.47	75.09
FAR	28.58	38.83	30.99
OS	1.64	8.66	8.81
F1	72	63.71	71.92

The gains from the DISTBIC method over SVF are immediately apparent, with the overall score $F1$ being increased significantly compared to SVF, at the expense of a slight increase in oversegmentation. We motivate our explanation for this increase in performance by noting the 12% higher hit rate of DISTBIC-SVF compared to the default SVF algorithm. This shows that the strength of DISTBIC-SVF lies in permitting a configuration of SVF which yields many more segmentations. The 20% lower false alarm rate, on the other hand shows that the ΔBIC eliminates more bad segmentations than the newly-configured SVF produces. The result being is an increase in performance as measured by $F1$ of almost 13%.

A comparison to the MMF-LLRT method shows that DISTBIC-SVF closely approaches its overall performance, with a higher hit hate at the expense of a slightly higher false alarm rate and a higher oversegmentation rate.

Table 2: Parameter configuration of DISTBIC-SVF corresponding to the scores shown in Table 1

SVF + Heuristics		Dynamic Windowing + BIC	
Feature	Value	Feature	Value
MFCC frame step	5 ms	MFCC frame step	1 ms
MFCC frame length	7.5 ms	MFCC frame length	20 ms
α	0.5	λ	1.5
h_2	2		

Conclusion

With the introduction of DISTBIC-SVF we sought to improve upon SVF in order to provide a simple, text-independent phoneme segmentation algorithm having good performance. By evaluating the algorithm on the TIMIT database, we have shown that this technique significantly improves upon the baseline SVF segmentation. Through comparison of our algorithm's scores versus those of the baseline, we demonstrated a synergism of the advantages of SVF and DISTBIC SVF. Namely, whereas SVF may yield a high hit rate with relatively high error, DISTBIC permits a reduction of these errors, giving an overall better segmentation. Further, our implementation shows that we can obtain performance comparable to that of a state-of-the-art method using measures of spectral variation as a basis for phoneme segmentation.

It would be worthwhile for future work to study other feature bases for spectral variation than MFCC, or different frequency scaling. Also, because our system maintains a relatively high false alarm rate even after the steps of DISTBIC, further performance gains may be obtained through error analysis of the predominant causes of insertions.

References

[1] H. Kawai, T. Toda, "An evaluation of automatic phone segmentation for concatenative speech synthesis," in *Proceedings IEEE Int. Conf. Acoust., Speech, Signal Process.,* 1994, vol. 1, pp. 677680.

[2] A. Esposito, G. Aversano, "Text Independent Methods for Speech Segmentation," in *Nonlinear Speech Modeling and Applications,* LNAI 3445, 2005, pp. 261-290

[3] G. Almpanidis, M. Kotti, C. Kotropoulos, "Robust detection of phoneme boundaries using model selection criteria with few observations," in *IEEE Transactions on Audio, Speech, and Language Processing,* Vol. 17, Issue 2, February 2009

[4] M. Artimy, W. Robertson, W. Phillips, "Automatic detection of acoustic sub-word boundaries for single digit recognition," in *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering,* 1999, pp. 751 - 754

[5] B. Petek, O. Andersen, P. Dalsgaard, "On the robust automatic segmentation of spontaneous speech," in *Proceedings of ICSLP 1996*, Oct. 1996, pp. 913-916

[6] F. Brugnara, R. Mori, D. Giuliani, M. Omologo, "Improved connected digit recognition uisng spectral variation functions," in *International Conference on Spoken Language Processing*, October 1992, pp. 627-63

[7] C. Mitchell, M. Harper, L. Jamieson, "Using explicit segmentation to improve HMM phone recognition," in *Proceedings IEEE Int. Conf. Acoust., Speech, Signal Process.,* vol. I, 1995, pp. 229-232

[8] P. Delacourt, C.J. Wellekens,"DISTBIC: A speaker-based segmentation for audio data indexing," in *Speech Communication,* Vol. 32 N1-2, 2000, pp. 111-126

[9] M. Sharma, R. Mammone, "Blind speech segmentation: automatic segmentation of speech without linguistic knowledge," in *Proceedings of ICSLP 1996*, Oct. 1996, pp. 1237-1240

[10] V. Khanagha, K. Daoudi, O. Pont, H. Yahia, "Improving text-independent phonetic segmentation based on the microcanonical multiscale formalism," in *Proceedings IEEE Int. Conf. Acoust., Speech, Signal Process.,*, 2011