AUTO-SEGMENTATION BASED PARTITIONING AND CLUSTERING APPROACH TO ROBUST ENDPOINTING

Yu SHI, Frank K. Soong, Jian-lai ZHOU

Microsoft Research Asia, Beijing, China
{yushi,frankkps,jlzhou}@microsoft.com

ABSTRACT

An auto segmentation based partitioning and clustering approach to robust Voice Activity Detection (VAD) is proposed. It is done in two successive steps: homogeneous frame partitioning and segment clustering. The first step, due to its auto segmentation nature, does not need a noise model, and is applicable to different noise types and SNR's. The algorithm is a dynamic programming based procedure and provides a graceful performance in finding segmentation thresholds. Multiple parameters like energy, pitch and voicing information can be easily incorporated into the procedure. The algorithm is evaluated on the test sets in the Aurora2 database. The algorithm shows its robustness at low SNR operating environments; the endpoint estimate errors are shown to have small variance.

1. INTRODUCTION

Endpointing is a key component in speech recognition systems. There is a strong need of robust VAD technique with more and more speech recognition used in real applications.

Threshold based VAD algorithms extract some measured features from the input signal and compare to thresholds. If the features exceed the threshold, a decision that voice is active can be made [11]. How to achieve accurate thresholds, however, is a big issue in endpointing. Robust estimation method is necessary. Given a background noise interval where the threshold is estimated or updated, mean-value based algorithms depend on whether the interval contains voice or not, while histogram-based algorithms [9] relay on the percentage of voice and background noise in the interval [13]. Histogram-based algorithms also need robust pick detection and lots of data to obtain accurate Probability Distribution Function (PDF). Frame-clustering based methods [13] organize frame features into different clusters without considering the continuity of voice. On the other hand, multiple parameters must be incorporated to be compared with the threshold. Such algorithms like in [14] [10] [12] need *a prior* information or parameter training.

In this paper we propose a new endpointing algorithm based on auto segmentation which is also named as optimal segmentation [4] or optimal partitioning [7] by others. Auto segmentation is an image processing based technique. Its goal is to divide a time series into homogeneous blocks. A variety of signal processing and related problems such as signal detection and characterization, density estimation, cluster analysis, and classification can be viewed as the search for an optimal partition of data given on a time interval. Auto segmentation has already been successfully used in image de-noising [4], text segmentation in information retrieval [5] [3], DNA/RNA sequence analysis [1] [15], etc. In the auto segmentation algorithms, quite a lot try to minimize the segmentation cost via Dynamic Programming (DP) that is often employed in alignment and model-fitting sequence segmentation algorithms.

In the proposed method, we also use DP to do the search. The segmentation score function is defined as a homogeneity criterion penalized by segmentation complexity. We first perform auto segmentation to partition feature vectors of all frames in the time interval into several segments. A purpose of this step is to limit speech-noise transfer times in the interval. Relaxing the dependence on the percentage of noise in the interval is another consideration. Due to its auto segmentation nature, this step does not need a noise model, and is applicable to different noise types and Signal-to-Noise Ratios (SNR's). The produced segment centroids are then ordered according to a sorting factor, after which auto segmentation is performed again to separate the sorted segment centroids into 2 clusters, one is for speech and the other is for background noise. Finally endpoints in the time interval are determined according to the start and end time of the first and last speech segments, respectively. Since both auto segmentations are DP based procedure, the algorithm provides a graceful performance in finding segmentation boundaries and classification threshold. Multiple parameters like energy, pitch and voicing information can be easily incorporated into the procedure.

The algorithm is evaluated on the test sets in the Aurora2 database. Variance of endpoint estimate errors is considered as a primary guideline to evaluate the system performance. In the experiments, we first investigate the speech-noise discrimination of different types of parameters like energy, pitch and voicing information. Then robustness of the proposed approach is tested. From the experimental results we can see that at low SNR operating environments, the endpoint estimate errors are shown to have small variance.

2. AUTO SEGMENTATION AND HOMOGENEOUS FRAME PARTITIONING

For a given time interval I which contains N frames and a predefined parameter K ($1 \leq K \leq N$) which represents the total number of segments to be produced, segmentation $\mathbf{S}(I,K)$ is defined as a set of K blocks

$$\mathbf{S}(I,K) = \{S_k, 1 \leq k \leq K\}$$

where the blocks are sets of frames defined by consecutive indexes $\mathcal{N}_k = \{n_{k-1}+1, \ldots, n_k\}$

$$S_k = \{\vec{\mathbf{x}}_n, n \in \mathcal{N}_k\}$$

satisfying the usual conditions $\bigcup_k S_k = I$ and $S_k \bigcap S_{k'} = \emptyset$ if $k \neq k'$. Here $\vec{\mathbf{x}}_n$ is the d-dimensional feature vector associated with frame n, and n_k is the end frame of segment S_k. The segmentation score function, similarly with Bayesian Information Criterion (BIC) [2], is defined as a homogeneity criterion penalized by segmentation complexity: the number of parameters in the segmentation. The formulation is

$$\begin{aligned} F[\mathbf{S}(I,K)] &= H[\mathbf{S}(I,K)] + P[\mathbf{S}(I,K)] \\ &= \sum_{k=1}^{K} D_k + \lambda_p \#[\mathbf{S}(I,K)] \log(N) \end{aligned}$$

where $H[\mathbf{S}(I,K)]$ is the homogeneity criterion of segmentation $\mathbf{S}(I,K)$ and $P[\mathbf{S}(I,K)]$ is the penalty item. $D_k = D(n_{k-1}+1, n_k)$ is a measure function of homogeneity associated with segment k positioned from frame $n_{k-1}+1$ to n_k. λ_p is the penalty weight. $\#[\mathbf{S}(I,K)]$ is the number of parameters in the segmentation $\mathbf{S}(I,K)$.

In this paper, $D(n_1,n_2)$ is a within-segment distortion which is a function of its boundaries:

$$D(n_1,n_2) = \sum_{n=n_1}^{n_2} [\vec{\mathbf{x}}_n - \vec{\mathbf{C}}(n_1,n_2)]^T [\vec{\mathbf{x}}_n - \vec{\mathbf{C}}(n_1,n_2)]$$

where

$$\vec{\mathbf{C}}(n_1,n_2) = \frac{1}{n_2-n_1+1} \sum_{n=n_1}^{n_2} \vec{\mathbf{x}}_n$$

is the centroid of the segment. Thus the number of parameters in the segmentation $\mathbf{S}(I,K)$ is $K \times d$. An optimal segmentation $\mathbf{S}^*(I)$ can be obtained by minimizing $F[\mathbf{S}(I,K)]$ over all segment numbers and segment boundaries:

$$\mathbf{S}^*(I) = \arg\min_{K,|\mathbf{S}|=K} F[\mathbf{S}(I,K)]$$

Since the segmentation complexity is independent on the positions of segment boundaries when the number of segments is fixed, we can separate the minimization into two successive procedures: first minimize $H[\mathbf{S}(I,K)]$ for each K and then find the minimum value of $F[\mathbf{S}(I,K)]$ over all K.

The minimum of $H[\mathbf{S}(I,K)]$ can be found through a DP procedure which can be implemented in level building manner, i.e., the lth level has l segments. So given a K, there are total $L = K$ levels in DP search. The algorithm derives the optimal partition of the first n frames at level l using previously obtained optimal partitions, i.e., those of the first $1, 2, \ldots, n-1$ frames at level $l-1$. At each level we must consider all possible ending locations $j, l-1 \le j < n$ of the next-to-last segment of the optimal partition. For each putative j, the distortion function is — by the principle of optimality — the distortion of the optimal subpartition prior to j plus the distortion of the last segment itself. The former was stored at previous level, and the later is a simple evaluation of D. The desired new optimal segmentation corresponds to the minimum over all j.

In the implementation of auto segmentation in frame partitioning, a constrained DP algorithm is adopted. The number of frames in produced segments is limited in the range $[n_a, n_b]$. The lower bound is the shortest duration that a phone should occupy, and the upper bound is used to save computing resources. Two boundary functions on the values of frame index n for a given value of level l are defined as $B_a(l) = n_a l$ and $B_b(l) = n_b l$. They are used to restrict the range of the optimal segmentation of the first n frames at level l to fall within a reasonable set of the (n,l) plane. And the putative ending locations of the next-to-last segment of the optimal path, j, are bounded by $(n-n_b)$ and $(n-n_a)$. Due to the limitation of minimum length of the segments, there should be at most $\lfloor N/n_a \rfloor$ segments or levels to be produced. (The symbol $\lfloor x \rfloor$ means the largest integer not greater than x.) More precisely, define $H^*(n,l)$ to be the value of the distortion of the optimal segmentation $\mathbf{S}^*(n,l)$ of the first n frames at level l, for $1 \le n \le N$. The DP algorithm shown in Fig. 1 finds the optimal segmentation $\mathbf{S}^*(N,l^*)$, i.e. $\mathbf{S}^*(I)$.

To further speed up the algorithm, frame partitioning is performed in every 0.5 second.

There are two obvious advantages of frame partitioning before clustering. It is capable of limiting speech-noise transfer times in the time interval and relaxing algorithm's dependence on the percentage of noise.

Auto Segmentation Algorithm

1. Start level ($l = 1$)

$$H^*(n,l) = \begin{cases} D(1,n), & B_a(1) \le n \le B_b(1) \\ \infty, & \text{else} \end{cases}$$

2. For $l = 2, \ldots, (L = \lfloor N/n_a \rfloor)$, do
   - For $n = B_a(l), \ldots, B_b(l)$, do
     - Compute
     $$H^*(n,l) = \min_j \{H^*(j,l-1) + D(j+1,n)\},$$
     for $n - n_b \le j \le n - n_a$.
     - The value of j where this minimum occurs is stored as $p(n,l)$.
3. Select $K^* = l^* = \arg\min_l [H^*(N,l) + \lambda_p l d \log(N)]$ as the optimal number of segments.
4. Backtrack using p to identify the end locations of individual blocks of the optimal segmentation $\mathbf{S}^*(N,l^*)$ in the following way. Let $n_{K^*} = N$, $n_{K^*-1} = p(n_{K^*}, K^*)$, $n_{K^*-2} = p(n_{K^*-1}, K^*-1)$, etc. Then the last block in $\mathbf{S}^*(N,l^*)$ contains frames $n_{K^*-1}+1, \ldots, n_{K^*} = N$, the next-to-last block in $\mathbf{S}^*(N,l^*)$ contains frames $n_{K^*-2}+1, \ldots, n_{K^*-1}$, and so on.
5. Compute centroid $\vec{\mathbf{C}}_k = \vec{\mathbf{C}}(n_{k-1}+1, n_k)$ for each segment.

Fig. 1. Auto segmentation algorithm.

3. SEGMENT CLUSTERING AND ENDPOINT IDENTIFICATION

In this section, we propose a way to organize the segments produced in last section into 2 classes, i.e., speech and noise.

First we represent each segment by its centroid to form a new series $\{\vec{\mathbf{C}}_k, k = 1, 2, \ldots, K^*\}$. Second we sort the centroid series according to a factor related to the variance normalized time-domain LOG Energy (LOGE) and Cross Correlation corresponding to Pitch (CCP). To say more precisely, CCP is obtained from the output of pitch tracker in the Entropic Signal Processing System (ESPS) [8]. For voiced region, it is the peak normalized cross-correlation value that was found to determine the output F0, while for unvoiced region, it is the largest cross-correlation value found at any lag. After the average LOGE (E_k) and CCP (P_k) of each segment being calculated, the sorting factor can be obtained by adding them together, $Q_k = E_k + P_k$. The segment indexes $\{1, 2, \ldots, K^*\}$ are then ordered according to the sorting factor to generate the ordered segment centroids $\{\vec{\mathbf{C}}_{k_1}, \vec{\mathbf{C}}_{k_2}, \ldots, \vec{\mathbf{C}}_{k_{K^*}}\}$ where $\{k_i, 1 \le i \le K^*\}$ satisfies $Q_{k_1} \le Q_{k_2} \le \cdots \le Q_{k_{K^*}}$.

After ordering the segment centroids in the way described in last paragraph, we want to find a boundary to separate speech segments from noise segments. This is obviously another auto segmentation procedure which has only 2 levels. In this step, we do not use the segmentation penalty, i.e., $\lambda_p = 0$, since we suppose there are both speech and noise in the interval. If this kind of clustering technique directly acts on the sorted frame-level feature vectors rather than segment-level, the results must depend on the percentage of noise in the interval. It is obvious since the distortion function is associated with the number of frames in each segment which is hidden in segment-level clustering. Because it is a DP based procedure, this step provides a graceful performance in finding the classification threshold.

Table 1. Endpoint estimate errors (in msec): test set A, averaged over 0-20 dB. (m: mean, σ^2: variance)

ID	parameter	d	Start point		End point	
			m	σ	m	σ
1	LOGE	1	67	102	-117	136
2	RMS	1	98	78	-229	217
3	CCP	1	42	146	-76	175
4	MFCC's	12	34	142	-71	170
5	FBANK's	23	63	114	-120	157
6	1+3	2	74	98	-116	129
7	1+4	13	61	109	-99	140
8	1+5	24	65	111	-121	151
9	2+3	2	90	85	-145	145
10	2+4	13	69	105	-119	156
11	2+5	24	69	108	-128	155
12	1+2+3	3	86	85	-141	137
13	1+3+4	14	66	104	-104	136
14	1+3+5	25	67	109	-119	144
15	2+3+4	14	73	100	-118	145
16	2+3+5	25	70	106	-126	148
17	1+2+3+4	15	77	91	-122	132
18	1+2+3+5	26	71	104	-126	145
19	F0	1	40	137	-96	177

Final decision of endpoints in the interval is identified as the start and end time of the first and last speech segments, respectively.

4. EXPERIMENTAL RESULTS

The experimental database used in this study is the Aurora2 test sets [6]. Endpoint references are obtained by aligning clean test data to a set of HMM models trained on clean data in both training and test sets. In this study, VAD performance is evaluated via the time errors of estimated endpoints to references. Positive value denotes behind while negative value means before. Parameters in frame partitioning are set as follows. Segment length is limited to 3-25 frames in each half second interval. The penalty weight λ_p is set to 0.2;

In the experiment, we first want to investigate the speech-noise discrimination of different types of parameters. The parameters examined here are Mel-Frequency Cepstral Coefficients (MFCC's), log Mel-scale Filter BANK energies (FBANK's), LOGE, local Root Mean Squared measurement (RMS), CCP, and their combinations. MFCC's and FBANK's only contain static features. RMS is also produced by ESPS pitch tracker and is approximately a linear time-domain energy. LOGE and CCP have already been described in Section 3. The reason why we look upon linear and log energies as different parameters is that they give different contribution to start and end points as we noticed. All parameters mentioned here are variance normalized. Table 1 shows the comparison results on test set A at SNR's 0-20 dB. Pitch based results are also given in the table (the last line): the start point denotes the start time of the first frame where pitch is detected and end point denotes the end time of the last voiced frame. Utterances where no pitch detected have not been considered for the last line.

Start points in Table 1 consistently have positive mean errors compared with model-based references, while end points have negative ones. Thus we need compensate the errors by negative and positive constants, respectively. In fact, we want to pay more attention to the variance of the estimate errors since it determines how efficient the compensation could be. From the comparison results, we can conclude that ("$>$" means "better than" here):

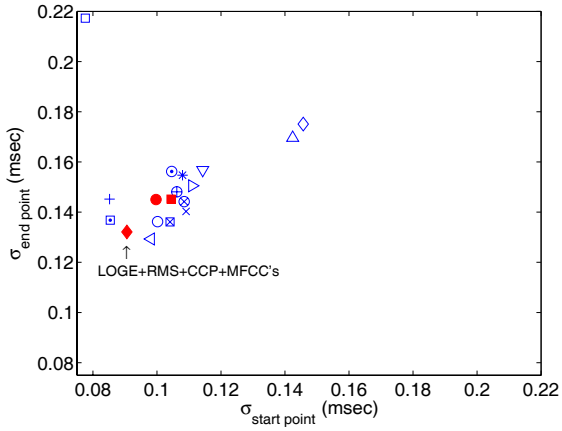

Fig. 2. Scatter diagram of standard deviations of endpoint estimate errors: test set A, averaged over 0-20 dB. ('○':LOGE, '□':RMS, '◇':CCP, '△':MFCC's, '▽':FBANK's, '◁':LOGE+CCP, '×':LOGE+MFCC's, '▷':LOGE+FBANK's, '+':RMS+CCP, '⊙': RMS+MFCC, '∗':RMS+FBANK, '⊡': LOGE+RMS+CCP, '⊠': LOGE+CCP+MFCC's, '⊗':LOGE+CCP+FBANK's, '•': RMS+CCP+MFCC's, '⊕': RMS+CCP+FBANK's, '♦': LOGE+RMS+CCP+MFCC's, '■': LOGE+RMS+CCP+FBANK's)

- Variances of estimate errors of end points are always larger than that of start points.
- For start points the individual parameters are ranked as RMS $>$ LOGE$>$FBANK's$>$MFCC's$>$CCP.
- For end points they are LOGE$>$FBANK's$>$MFCC's$>$CCP$>$ RMS.
- When combined with other parameters, MFCC's show good performance than FBANK's.
- Though CCP alone is not good, it can help others in most cases.

Fig. 2 shows the scatter diagram of the standard deviations of estimate errors for both start points and end points. From the figure, we can see that parameter group 17, i.e. LOGE+RMS+CCP+MFCC's is a better choice. We select it as the feature vector for the following experiments.

Besides investigating the ability to obtain consistent endpoint estimate errors of different parameters, we also want to examine the robustness of the proposed approach. Fig. 3 gives an example on file FAK_3Z82A in subway noise at different SNR's. In each sub-figure, vertical bars on the horizontal line below the waveform denote the segment boundaries in frame partitioning. Below it is the segment clustering result. Higher-level parts denote speech and lower-level parts indicate background noise. The endpoints are marked as longer vertical bars on the waveform and labeled by "start point" and "end point". When SNR's are high, the algorithm can accurately detect voice activity. Even at low SNR's, the algorithm can also obtain reasonable speech-noise classification and endpoint positions.

Fig. 4 plots the standard deviation (std) of endpoint estimate errors of each test set in different noise at different SNR's. The curves are flat and almost less than 0.1 second in the SNR range of clean to 10 dB. Even in 0 dB noise, the standard deviations can be controlled within 0.3 second.

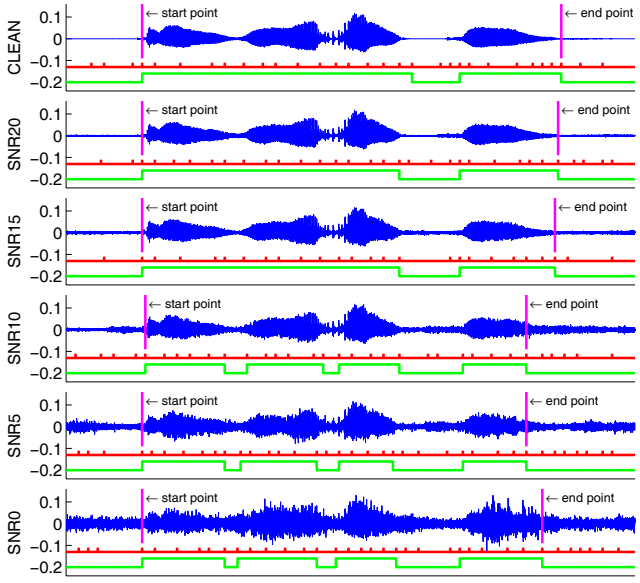

Fig. 3. Endpointing example: test set A, subway noise, file FAK_3Z82A.

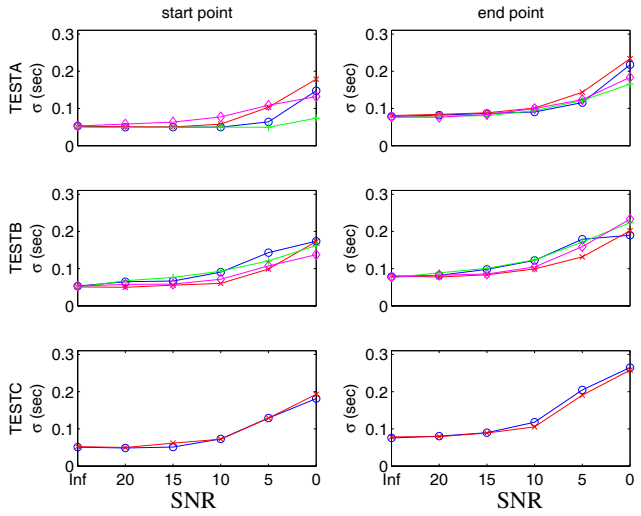

Fig. 4. Standard deviation of endpoint estimate errors (test sets A-C, 'o': N1, '×': N2, '+': N3, '◇': N4)

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a robust endpoint detection algorithm based on auto segmentation. It was implemented in two steps: homogeneous frame partitioning and segment clustering. The first step partitioned all frames in the time interval into several segments. This process is capable of limiting speech-noise transfer times and relaxing algorithm's dependence on the percentage of noise in the interval. Due to its auto segmentation nature, this step does not need a noise model, and is applicable to different noise types and SNR's. The second step clustered the produced segment centroids into speech and noise. Since DP based procedure is used in both steps, the algorithm provided a graceful performance in finding segmentation boundaries and classification threshold. Multiple parameters like energy, pitch and voicing information can be easily incorporated into the procedure. The proposed algorithm was evaluated on the test sets in the Aurora2 database. Variance of endpoint estimate errors was considered as a primary guideline to evaluate the system performance. Experiments on both parameter comparing and robustness testing were carried out. In the first experiment, speech-noise discrimination of several parameters, like energy, pitch, voicing information, and their combinations were compared. In the second experiment, the algorithm showed its robustness at low SNR operating environments. The endpoint estimate errors are shown to have small variances. In the future, we would like to take into account the time intervals where either speech or noise exists. We would also like to automatically compensate the consistent endpoint estimate errors in the algorithm.

6. REFERENCES

[1] J. V. Braun and H.-G. Müller, "Statistical methods for DNA sequence segmentation," *Statistical Science*, vol. 13, no. 2, pp. 142-162, 1998.

[2] S. S. Chen, P. S. Gopalakrishnan, "Clustering via the Bayesian information criterion with applications in voice recognition," in *Proc. ICASSP'98*, vol 1, pp. 645-648, 1998.

[3] P. Fragkou, V. Petridis, and A. Kehagias, "A dynamic programming algorithm for linear text segmentation," *J. Intell. Inform. Syst.*, vol. 23, no. 2, pp. 179-197, Sep. 2004.

[4] T. X. Han. S. Kay. T. S. Huang, "Optimal segmentation of signals and its application to image denoising and boundary feature extraction," in *Proc. ICIP'2004*.

[5] O. Heinonen, "Optimal multi-paragraph text segmentation by dynamic programming," in *Proc. COLING-ACL'98*, pp. 1484-1486, 1998.

[6] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW ASR2000*, pp. 181-188, Sept. 2000.

[7] B. Jackson, J. D. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and T. T. Tsai, "An algorithm for optimal partitioning of data on an interval," *IEEE Signal Processing Letter*, vol. 12, no. 2, pp. 105-108, February 2005.

[8] ESPS, http://www.speech.kth.se

[9] L. F. Lamel, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilson, "An improved endpoint detector for isolated word recognition," *IEEE Transaction on Acoustics, Speech, and Signal Processing*, vol. 29, pp. 777-785, August 1981.

[10] A. Martin, D. Charlet, and L. Mauuary, "Robust speech/non-speech detection using LDA applied to MFCC," in *Proc. ICASSP'2001*, vol. 1, pp. 237-240, 2001.

[11] S. G. Tanyer and H. Ozer, "Voice activity detection in non-stationary noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 8, pp. 478-482, July 2000.

[12] Y. Tian, Z. Wang, and D. Lu, "Robust speech detection with heteroscedastic discriminant analysis applied to the time-frequency energy, " in *Proc. ISCSLP'2002* .

[13] Y. Tian, J. Wu, Z. Wang, and D. Lu, "Fuzzy clustering and bayesian information criterion based threshold estimation for robust voice activity detection," in *Proc. ICASSP2003* .

[14] G.-D. Wu and C.-T. Lin, "Word boundary detection with mel-scale frequency bank in noisy environment," *IEEE Transaction on Speech and Audio Processing*, vol. 8, no. 5, pp. 541-554, September 2000.

[15] P. Xing, C. A. Kulikowski, I. B. Muchnik, I. Dubchak, D. M. Wolf, S. Spengler, and M. Zorn, "Analysis of ribosomal RNA sequences by combinatorial clustering," in *Proc. ISMB'99*, pp. 287-296.