An Efficient Scoring Algorithm for Gaussian Mixture Model Based Speaker Identification

Bryan L. Pellom, *Student Member, IEEE,* and John H. L. Hansen, *Senior Member, IEEE*

***Abstract*—This letter presents a novel algorithm for reducing the computational complexity of identifying a speaker within a Gaussian mixture speaker model framework. For applications in which the entire observation sequence is known, we illustrate that rapid pruning of unlikely speaker model candidates can be achieved by reordering the time-sequence of observation vectors used to update the accumulated probability of each speaker model. The overall approach is integrated into a beam-search strategy and shown to reduce the time to identify a speaker by a factor of 140 over the standard full-search method, and by a factor of six over the standard beam-search method when identifying speakers from the 138 speaker YOHO corpus.**

I. Introduction

THE ABILITY to recognize a speaker by voice has recently received much attention in the literature. Applications of speaker identification and verification include banking over the telephone, computer security, as well as access to secure documents over the internet. In [1], the use of Gaussian mixture models (GMM's) for speaker identification was shown to provide superior performance compared with several existing techniques. For example, error rates as low as 0.7% have been reported on the 138 speaker, 8 kHz sampled YOHO corpus [2]. However, as the population size and length of test material increases, the computational cost of performing the identification can increase substantially. This letter addresses the problem of reducing the computational complexity of the speaker identification task by applying beam-search pruning in tandem with a novel reordering of the observation sequence.

II. Speaker Identification Based on Gaussian Mixture Models

In GMM-based speaker identification, speech is characterized by frame-synchronous observation vectors, $\mathbf{X} = \{\vec{x}_1, \vec{x}_2, \vec{x}_3, \ldots, \vec{x}_T\}$. Typical frame rates are on the order 10 ms and D-dimensional features are extracted from overlapping analysis windows centered about each frame instant. During identification, the system is presented with a sequence of observations $\mathbf{X}$ produced by one of S modeled speakers. The identity of the speaker producing $\mathbf{X}$ is determined by finding the speaker model λ_s which maximizes the *a posteriori* probability across the speaker set $\lambda_k : k \in (1, S)$

$$\lambda_s = \arg\max_{1 \le k \le S} P(\lambda_k \mid \mathbf{X}). \tag{1}$$

Using Bayes Rule, (1) can be expressed as

$$\lambda_s = \arg\max_{1 \le k \le S} \frac{p(\mathbf{X} \mid \lambda_k) P(\lambda_k)}{p(\mathbf{X})}. \tag{2}$$

Assuming each speaker model is equally likely and noting that $p(\mathbf{X})$ is the same for all models, the identification task can be summarized as finding

$$\begin{aligned} \lambda_s &= \arg\max_{1 \le k \le S} p(\mathbf{X} \mid \lambda_k) \\ &= \arg\max_{1 \le k \le S} \prod_{t=1}^{T} p(\vec{x}_t \mid \lambda_k) \end{aligned} \tag{3}$$

where $p(\vec{x}_t \mid \lambda_k)$ is assumed to be modeled by a mixture of M multivariate Gaussian distributions, $\mathcal{N}(\vec{x}_t; c_m, \vec{\mu}_m, \Sigma_m)$, where c_m, $\vec{\mu}_m$, and Σ_m represent the mixture weight, mean vector, and covariance matrix representing the mth distribution, respectively. In (3), the observations are assumed to be statistically independent, therefore temporal information is not encoded by the model. Furthermore, in order to avoid numerical stability problems, (3) is computed using the log-likelihood

$$\begin{aligned} \lambda_s &= \arg\max_{1 \le k \le S} \sum_{t=1}^{T} \log\{p(\vec{x}_t \mid \lambda_k)\} \\ &= \arg\max_{1 \le k \le S} \sum_{t=1}^{T} \log\left\{ \sum_{m=1}^{M} \frac{c_{m,k}}{(2\pi)^{D/2} |\Sigma_{m,k}|^{1/2}} \right. \\ &\qquad \left. \times \exp\left\{ -\frac{1}{2} (\vec{x}_t - \vec{\mu}_{m,k})' \Sigma_{m,k}^{-1} (\vec{x}_t - \vec{\mu}_{m,k}) \right\} \right\}. \end{aligned} \tag{4}$$

In general, the observations are modeled using diagonal covariance matrices yielding

$$\begin{aligned} \lambda_s = \arg\max_{1 \le k \le S} \sum_{t=1}^{T} \log\left\{ \sum_{m=1}^{M} \frac{c_{m,k}}{(2\pi)^{D/2} \left(\prod_{j=1}^{D} \sigma_{m,k}^2(j) \right)^{1/2}} \right. \\ \left. \times \exp\left\{ -\frac{1}{2} \sum_{j=1}^{D} \frac{(x_t(j) - \mu_{m,k}(j))^2}{\sigma_{m,k}^2(j)} \right\} \right\}. \end{aligned} \tag{5}$$

The complete evaluation of (5) can require significant computational resources. This is especially true if the number of modeled speakers or the duration of the test material is large.

Manuscript received May 5, 1998. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. L. Niles.

The authors are with the Robust Speech Processing Laboratory, Department of Electrical Engineering, Duke University, Durham, NC 27708-0291 USA (e-mail: jhlh@ee.duke.edu).

Publisher Item Identifier S 1070-9908(98)08683-0.

One common method for reducing the computational overhead involves using a "nearest-neighbor" [3] approximation of the likelihood in (5)

$$\lambda_s = \arg\max_{1\le k\le S} \sum_{t=1}^{T} \max_{1\le m\le M} \left\{ C_{m,k} - \frac{1}{2}\sum_{j=1}^{D} \frac{(x_t(j)-\mu_{m,k}(j))^2}{\sigma^2_{m,k}(j)} \right\} \quad (6)$$

where

$$C_{m,k} = \log(c_{m,k}) - \frac{D}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{D}\log\left(\sigma^2_{m,k}(j)\right). \quad (7)$$

Note that the mixture dependent constant $C_{m,k}$ is completely known prior to algorithm run-time and can be precomputed. Other studies have considered applying a beam-search during likelihood calculation. Here, the partial sum of (6) at time τ can be used to update a pruning threshold

$$\Theta_\tau = \left\{ \max_{1\le k\le S(\tau)} \sum_{t=1}^{\tau} \max_{1\le m\le M} \left\{ C_{m,k} - \frac{1}{2}\sum_{j=1}^{D} \frac{(x_t(j)-\mu_{m,k}(j))^2}{\sigma^2_{m,k}(j)} \right\} \right\} - B \quad (8)$$

where $S(\tau)$ denotes the current set of active (i.e., unpruned) models at time τ and B is a constant used to define the user controlled beam-width. During processing, active speaker models whose log-likelihood score falls below Θ_τ are eliminated from the search.

III. Algorithm Formulation

Typical speech processing systems analyze speech by calculating features from overlapping windowed sections of data (on the order of 20 to 30 ms) during which the vocal tract characteristics are assumed stationary. The process of frame overlapping results in neighboring observations which exhibit a high degree of correlation. In the context of speaker identification, correlation among adjacent observations violates the original statistical independence assumption and results in reduced efficiency of the beam-search. This is due to the fact that a limited amount of information is gained from observation $\vec{x}_{t+1}$ versus $\vec{x}_t$ since they sample similar locations in the speaker's acoustic space. Consequently, many observations must be examined before models of unlikely speakers can be pruned during processing.

Intuitively, one might consider approaching this problem using variable frame rates (e.g., sampling the speech observations less often during periods of slow spectral change and more often during periods of fast spectral change). We point out that this approach would throw away data that might be useful in the overall decision. Likewise, one could also consider a method by which the observations are selected based on a spectral distance criterion (e.g., sampling a new observation when the spectral distance between the last sampled observation and the current observation exceeds some threshold). Here, the savings in speed may be outweighed by the cost of the spectral distance computation. In addition, information may be lost just as in the case of the variable frame-rate processing strategy.

The novel approach considered in this letter provides a computationally inexpensive method for improving the information gained from each observation. To achieve this goal, we assume that the entire observation sequence is known and consider reordering the temporal sequence of observations. This is motivated by the fact that the order of the parameter sequence does not affect the final decision given in (6). The reordering process is based on maximizing the *interval distance* or time-interval between successive observations used to update (6). The observation reordering proposed here has two advantages. First, since the observation sequence is reordered, there is no loss of data as in the case of variable frame-rate processing. Second, virtually no computational overhead is required to reorder the observation sequence based on the proposed criterion. One can think of the maximally-distant time interval sampling scheme as providing a mechanism by which observations, drawn from differing phonemes, can be used to rapidly sample the acoustic space of the voice under test. The proposed algorithm is described as follows.

Step 1) Initialize $i = 1$. Form a subset of observations $\mathbf{O}^{(i)}$ containing η observations selected from uniformly spaced intervals across the vectors contained in $\mathbf{X} = \{\vec{x}_1, \vec{x}_2, \vec{x}_3, \cdots, \vec{x}_T\}$.

Step 2) Update the likelihood scores for all unpruned speaker models using observations contained in $\mathbf{O}^{(i)}$. During the update, set a pruning threshold Θ_τ as described in (8). Eliminate all speaker models whose accumulated log-probabilities fall below Θ_τ.

Step 3) Update the total set of scored observations, $\mathbf{Y}^{(i+1)} = \mathbf{Y}^{(i)} \bigcup \mathbf{O}^{(i)}$.

Step 4) Form a subset of observations $\mathbf{O}^{(i+1)}$ by sampling the observations nearest to the midpoints of previously scored elements found in $\mathbf{Y}^{(i+1)}$. For example, if $\vec{x}_1$ and $\vec{x}_5$ are part of $\mathbf{Y}^{(i+1)}$, then $\vec{x}_3$ would be placed in $\mathbf{O}^{(i+1)}$. Increase the pass count: $i = i + 1$.

Step 5) Repeat Steps 2–4 until until only one speaker model remains unpruned, or all observations have been scored and pick highest probable speaker.

For clarity, a graphical illustration of the observation vector reordering procedure is shown in Fig. 1 for an initial uniform sampling of $(\eta = 4)$ frames and a total observation count of $(T = 16)$. Here, observations $\mathbf{O}^{(1)} = \{\vec{x}_1, \vec{x}_5, \vec{x}_9, \vec{x}_{13}\}$ are first used to update the log-probability of each speaker model. Next, the remaining speaker models are updated using observations $\mathbf{O}^{(2)} = \{\vec{x}_3, \vec{x}_7, \vec{x}_{11}, \vec{x}_{15}\}$. Finally, the remaining models are updated using observations $\mathbf{O}^{(3)} = \{\vec{x}_2, \vec{x}_4, \vec{x}_6, \vec{x}_8, \vec{x}_{10}, \vec{x}_{12}, \vec{x}_{14}, \vec{x}_{16}\}$.

IV. Algorithm Evaluation

A. Evaluation Corpus and Speech Features

GMM's were estimated for each of the 138 speakers (106 male, 32 female) of the YOHO speech corpus [4]. To be con-

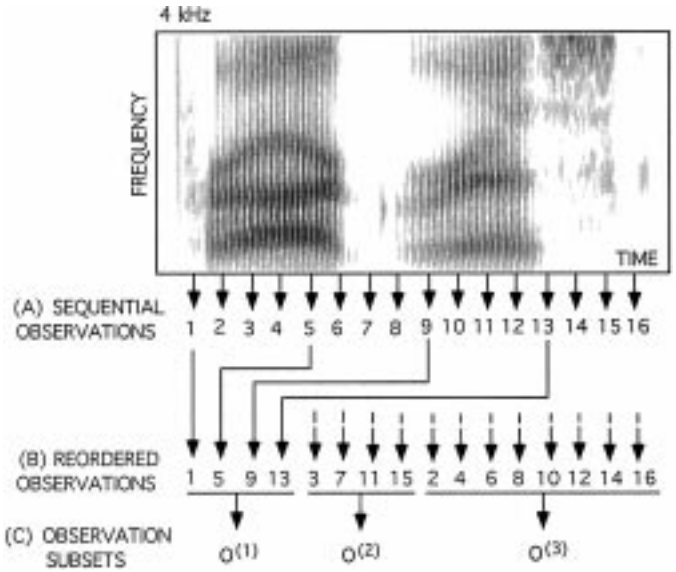

Fig. 1. Example of observation ordering for (A) standard GMM scoring algorithm in which observations are sequentially ordered based on arrival time and (B) reordered observation sequence using proposed algorithm with $\eta = 4$. In (C), speaker identification likelihood scores are estimated using each sequential reordered observation block (with model pruning during likelihood update).

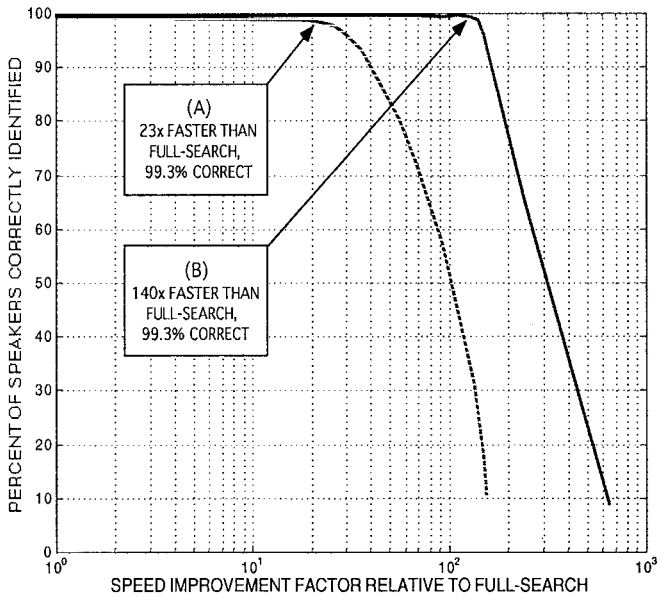

Fig. 2. Plot of speed improvement relative to full-search method versus speaker identification accuracy for the (138 speaker) YOHO corpus. Plots are shown for (A) beam-search with nearest neighbor Gaussian mixture density evaluation and (B) beam-search with nearest neighbor Gaussian mixture density evaluation and proposed observation reordering. In each case, the beamwidth was progressively narrowed to reveal a tradeoff between speaker identification accuracy and algorithm speed improvement.

sistent with previous studies, the training and testing conditions described in [2] were used for algorithm evaluation. Here, the training data for each speaker consisted of approximately 6 min of speech found in the enrollment section of the data base. The evaluation data consisted of ten verification sessions made up of four combination lock phrases (i.e., ten tests per speaker each of approximately 15 s in duration). During model training, the speech was preemphasized using a first-order finite impulse response (FIR) filter of the form $H(z) = 1 - 0.95z^{-1}$. Silence was removed by discarding low-energy frames using an energy based speech activity detection algorithm. During model training, the speech was parameterized every 10 ms from 20 ms overlapping windows. Each frame was parameterized by a vector consisting of 19 mel-frequency cepstral coefficients (MFCC) [5] and normalized log-frame energy. In total, 64 Gaussian mixtures were used to model each speaker.

B. Experimental Procedure

The computational speed of four different algorithm scenarios were compared. Evaluations included 1) full Gaussian mixture density evaluation without beam-search; 2) approximated nearest-neighbor Gaussian mixture density evaluation without beam-search; 3) nearest-neighbor approximation with beam-search; 4) the proposed algorithm consisting of nearest-neighbor approximation, beam-search, and observation reordering. For case 4), the value of η used in the initial uniform sampling was set to 10. For each scenario, the percent of test tokens correctly identified versus time (measured in seconds of the CPU clock) were noted. The CPU time measurement was started from the beginning of the scoring procedure until the identity of the speaker was determined. All simulations were conducted using a Sun Ultra-II workstation. For cases 3) and 4), the beam-search width was adjusted in order to reveal a trade-off between speaker recognition accuracy and computational cost.

C. Experimental Results

The speaker identification accuracy for the baseline system was found to be 99.3%. We point out that this is the same identification accuracy reported in [2]. For the case of complete density evaluation without beam-search [i.e., case 1); full-search], the algorithm required 21 465 s of CPU time on a Sun Ultra-II machine to perform the entire 1380 test scenarios. For case 2), which utilized the nearest-neighbor approximation given in (6), the ID rate remained at 99.3% while improving the speed by a factor of 1.67 (12 823 s of CPU time). Next, we considered nearest-neighbor density evaluation with beam-search. The width of the beam was adjusted in order to reveal the trade-off in algorithm speed versus speaker recognition accuracy. Results of this case 3) evaluation are shown in Fig. 2(a) as a speed improvement factor relative to the full-search versus percent of speakers correctly identified. Here, we see that the speaker ID performance begins to rapidly decline as the speed of the search is increased by more than a factor of 23 (933 s of CPU time) over the baseline full-search condition. However, using the proposed observation reordering method of case 4), we see in Fig. 2(b) that the speaker ID rate remains at 99.3% while delivering a speed improvement of a factor of 140 (153 s of CPU time) over the full-search condition. Beyond a speed-up factor of 140, the speaker ID rate of the proposed method gradually declines. The proposed method provides a factor of six speed improvement (i.e., $933/153 \approx 6$) over conventional sequential sampling with beam-search and comes at virtually no additional resource requirements.

V. Conclusion

In this letter, we have addressed the issue of reducing the computational complexity of identifying a speaker based on the Gaussian mixture speaker model framework. It was illustrated that because observation vectors are computed from overlapping analysis frames, the statistical independence assumption used in Gaussian mixture models is violated. Due to the high degree of correlation between adjacent observation vectors, many observations must be used to update the log-likelihood of each speaker model before unlikely candidates can be pruned using a beam-search mechanism. As a consequence, we have considered an operation that reorders the time-sequence of observation vectors in order to rapidly sample the acoustic space of the voice under test. The information content gained from each observation taken in this manner is significantly greater than that obtained by the traditional sequential log-likelihood update. As a result, unlikely speaker models are rapidly pruned from the search space, greatly reducing the computational complexity of the speaker identification algorithm. The proposed observation reordering was shown to decrease the search time by an additional factor of six over conventional sequential sampling with beam-search. The proposed method is easy to implement, can readily be integrated into existing GMM-based systems, and requires no additional overhead.

References

[1] D. Reynolds and R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 72–83, 1995.

[2] D. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Commun.*, vol. 17, pp. 91–108, 1995.

[3] F. Seide, "Fast likelihood computation for continuous-mixture densities using a tree-based nearest neighbor search," in *Proc. Eurospeech'95*, Madrid, Spain, vol. 2, pp. 1079–1082.

[4] J. Campbell, "Testing with the YOHO CD-ROM voice verification corpus," in *Proc. IEEE ICASSP'95*, Detroit, MI, 1995, vol. 1, pp. 341–344.

[5] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 357–366, 1980.