A FAST, SEQUENTIAL DECODING ALGORITHM WITH APPLICATION TO SPEAKER VERIFICATION

Qi Li

Multimedia Communications Research Laboratory
Bell Labs, Lucent Technologies
600 Mountain Avenue, Murray Hill, NJ 07974, USA
qli@research.bell-labs.com

ABSTRACT

To implement speaker verification (SV) technology for real-world applications with a large user population, the system cost becomes an important issue. One needs a fast algorithm which can support more users in a central telephone switch given the limited hardware, or can reduce the hardware requirement on a wireless handset. In [1], a fast, sequential decoding algorithm for left-to-right HMM was proposed. The algorithm is based on a sequential detection scheme which is asymptotically optimal in the sense of detecting a possible change in distribution as reliably and quickly as possible. In this paper, the algorithm is evaluated in a fixed-phrase SV system on a database with 23,578 utterances recorded from 100 speakers. The experimental results show that the decoding speed of the proposed algorithm is about 7 to 10 times faster than the Viterbi algorithm while the accuracy is in an acceptable level. The results indicate that the proposed algorithm can also be applied to speaker identification, utterance verification, audio segmentation, voice/silence detection and many other applications.

1. INTRODUCTION

When applying speaker verification (SV) technology to real-world applications, the system performance, including accuracy and response time, and cost have to be considered in implementation. From an application point of view, we want a system with acceptable performance while keeping the system cost as low as possible. Among different SV algorithms, the hidden Markov model (HMM) based approach gave us the best verification accuracy [2, 3, 4]. In the HMM approach, most of the computation is on HMM decoding. Therefore, in this paper, a fast HMM decoding algorithm with low complexity is evaluated for SV.

For an SV system implemented in a central telephone switch to support a large user population, given the limited hardware, e.g. a fixed number of speech processing boards, a fast, low complexity algorithm means the same hardware can support more telephone channels with the same response time, thus, the cost per channel can be lower. On the other hand, for a wireless handset or other equipment in which SV needs to be processed locally, a fast, low complexity algorithm can provide faster response, or reduce the cost on hardware while keeping the same response time.

Recently, a fast decoding algorithm for left-to-right HMM was proposed [1] based on a sequential detection scheme [5, 6]. The fast and low complexity property of the algorithm has the potential to meet the above requirements. Therefore, in this paper, we evaluate the algorithm on a fixed-phrase SV system and compare its performance with the Viterbi decoding algorithm in the terms of equal-error rate and decoding speed.

As is well known, HMM is a parametric statistical model with a set of states which characterize the evolution of a non-stationary process in speech through a set of short-time stationary events. Within each state, the distribution of the stochastic process is usually modeled by Gaussian mixtures, and the distribution changes from state to state sequentially in left-to-right HMM. The Viterbi algorithm, from graph and network theory, has been widely used for HMM decoding. It is optimal in the sense of maximum likelihood, but it is slow and complex, especially in a full-search implementation which has been used in SV. The proposed algorithm is to determine the changes in the distribution between different states, then finds the state boundary and computes likelihood scores or other kinds of scores sequentially [1].

2. FAST HMM DECODING ALGORITHM

Let $\mathbf{o}_n$ denote an observation vector at time n, and $p_1(\mathbf{o}_n)$ and $p_2(\mathbf{o}_n)$ be the density functions of well known, distinct, discrete, and mutually independent stochastic processes. In the case of HMM decoding, they are the density functions of two connected states, e.g. state 1 and state 2 respectively, and the observed vector sequence is initialy generated in state 1. Given the observation vector sequence, $\mathbf{O} = \{\mathbf{o}_n; n \geq 1\}$,

and the density functions $p_1(\mathbf{o}_n)$ and $p_2(\mathbf{o}_n)$, the objective is to detect a possible p_1 to p_2 change as *reliably and quickly* as possible. A sequential detection scheme was proposed by Page [5, 6]. It is asymptotically optimum in the sense that it requires the minimum possible expected sample size for decision, subject to a false alarm constraint [6, 7]. In [1], a fast HMM decoding algorithm based on the scheme was proposed as follows.

Select a time threshold $t_\delta > 0$. Observe data sequentially, and decide that the p_1 to p_2 change occurs, if

$$n - \ell \geq t_\delta, \tag{1}$$

and

$$T(\mathbf{o}^n) = \sum_{i=1}^{n} R_i(\mathbf{o}^i) - \min_{1 \leq k \leq n} \left\{ \sum_{i=1}^{k} R_i(\mathbf{o}^i) \right\} > \varepsilon, \tag{2}$$

where $\varepsilon \geq 0$ is a small number or can be just zero as we used in the experiments in this paper, $R_i(\mathbf{o}^i)$ is defined as

$$R_i(\mathbf{o}^i) = \log \frac{p_2(\mathbf{o}_i|\mathbf{o}_1^{i-1})}{p_1(\mathbf{o}_i|\mathbf{o}_1^{i-1})}, \tag{3}$$

and the last point ℓ of p_1 can be determined by

$$\ell = \arg \min_{1 \leq k \leq n} \left\{ \sum_{i=1}^{k} R_i(\mathbf{o}^i) \right\}. \tag{4}$$

Here, we assume that the duration of p_2 is not less than t_δ, and $p_1 \neq p_2$.

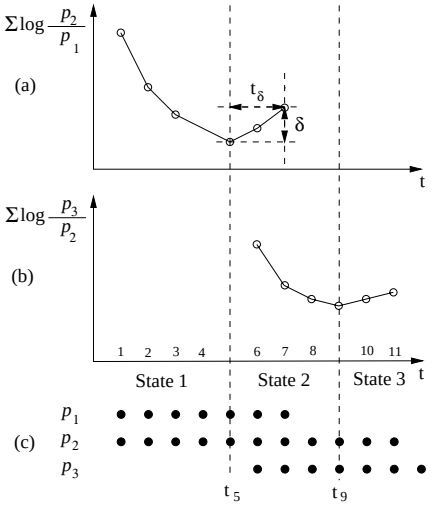

Figure 1: The scheme of the proposed decoding algorithm: (a) the end-point detection for state 1, t_5; (b) the end-point detection for state 2, t_9; and (c) the grid points for p_1, p_2 and p_3 computations (dots).

Multiple state segmentations in a left-to-right HMM can be realized by repeating the above procedure, i.e., to determine the changes of density functions from p_1 of state 1 to p_2 of state 2, from p_2 to p_3, and so on, sequentially. We use Fig. 1 to illustrate the concept. Fig. 1 (a) shows the scheme to determine the end point of state 1. The circles are the accumulated ratio values. Let $t_\delta = 2$, Eq. (1) and Eq. (2) are evaluated at each step sequentially. At $t = t_7$, we have $t_7 - t_5 \geq t_\delta = 2$ and $T(\mathbf{o}^7) > \varepsilon \geq 0$. Thus, the end point of state 1 is t_5. As shown in Fig. 1 (c), so far, only p_1 and p_2 are involved in the computation, where each dot represents one probability computation. The test continues from $t = t_6$ for state 2 as shown in Fig. 1(b). Following the same procedure as above, the determined end point for state 2 is t_9. It involves the computation from t_6 to t_{11} for p_2 and p_3 as shown in Fig. 1 (c).

As analyzed in [1], the speedup of the proposed algorithm compared with a widely used implementation of a full-search Viterbi algorithm is in the order of

$$S = \frac{NT(C+1)+T}{2\left[T+(N-1)t_\delta\right](C+2)} \approx \frac{NT}{2\left[T+(N-1)t_\delta\right]}, \tag{5}$$

where C is the number of floating point operations at each grid point for log probability as shown in Fig. 1 (c), N is the total number of states, T is the total number of frames, and t_δ is the time threshold. This formula has been verified numerically by speech examples in term of floating point operations (Flops) in [1].

When t_δ can not be determined precisely, a practical approach is to run the proposed algorithm more than once in the order of large to small $t_\delta(i)$, e.g. $t_\delta(i) = \{6, 4, 2\}$, then select the largest score. Assuming the the log probability in each grid point is saved from the first decoding with the largest $t_\delta(i)$, the number of additions is approximately in the order of

$$\begin{aligned} & 2\left[T+(N-1)t_\delta(1)\right](C+2) \\ & \quad +2\sum_{i=2}^{m}\left[T+(N-1)t_\delta(i)\right], \\ < \;& 2\left[T+(N-1)t_\delta(1)\right]\left[C+t_\delta(1)\right], \end{aligned} \tag{6}$$

where $t_\delta(1) = \max\{t_\delta(i)\}_{i=1}^{m}$. Since $C \gg t_\delta(1)$, the speedup is approximately the same as in Eq. (5), where $t_\delta = t_\delta(1)$.

3. FIXED-PHRASE SV SYSTEM

We selected a fixed-phrase SV system [2, 3, 4] to evaluate the proposed algorithm because the system is attractive to real-world applications.

The feature vector in this experiment is composed of 12 cepstrum and 12 delta cepstrum coefficients. The cepstrum is derived from a 10th order LPC analysis over a 30 ms window. The feature vectors are updated at 10 ms intervals.

During enrollment, assume five tokens of a true speaker's pass-phrases are collected and verified through a verbal information verification (VIV) procedure [4, 8] from five different sessions. A speaker-dependent (SD) target model, Λ_t, is then trained for the whole phrase. The model is a left-to-right HMM and the number of the states is about 1.5 times the total number of phonemes in the pass-phrase. There are 4 Gaussian mixtures associated with each state [2].

A block diagram of the test session is shown in Fig. 2. After the speaker claims the identity, the system expects the same phrase obtained in the training session. First, the input pass-phrase is segmented into silence and voice by forced decoding using SI phone models and the phoneme transcription saved in the user's profile. If a significantly different phrase is given, the phrase could be rejected by the SI phoneme decoding at this stage. Cepstral mean subtraction (CMS) is then conducted based on the segmentation for channel equalization. The voice portion, **O**, of the pass-phrase is used to compute two log-likelihood (LL) scores by force decoding: a target score, $L(\mathbf{O}, \Lambda_t)$, using the SD target model Λ_t, and a background score, $L(\mathbf{O}, \Lambda_b)$, using the background model Λ_b which is concatenated SI phone HMM's trained on a telephone speech database from different speakers and texts [2]. Each phone HMM has 3 states with 32 Gaussian components associated with each state. Due to unreliable variance estimates from limited amount of training data, a global variance estimate is used as a common variance to all Gaussian components [2] in the target models. A log-likelihood-ratio (LLR) score,

$$L_R(\mathbf{O}; \Lambda_t; \Lambda_b) = L(\mathbf{O}, \Lambda_t) - L(\mathbf{O}, \Lambda_b), \qquad (7)$$

is then calculated for the finial decision on rejection or acceptance by comparing the score with a threshold value.

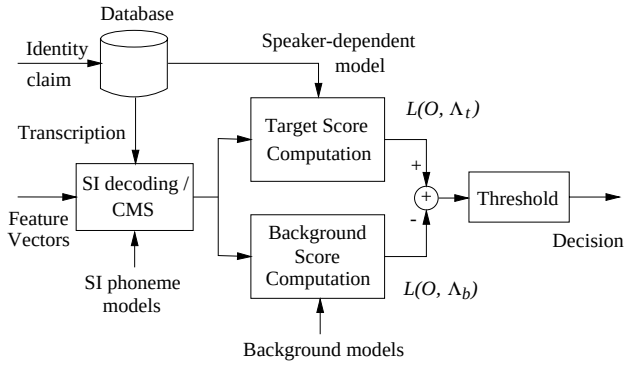

Figure 2: A fixed-phrase speaker verification system

3.1. SV Experiments

The experimental database consists of fixed phrase utterances recorded over the long distance telephone network by 100 speakers, 51 male and 49 female. The fixed phrase, common to all speakers, is "I pledge allegiance to the flag" with an average length of 2.2 seconds and a 25-states left-to-right HMM is used to model the whole phrase. For testing, we used 40 utterances recorded from a true speaker in different sessions (different telephone handsets and channels at different times), and 192 utterances recorded from 49 or 51 impostors of the same gender in different sessions.

In order to focus the evaluation on comparing the proposed algorithm with the Viterbi algorithm, we only test the algorithm on the target score computation, $L(\mathbf{O}, \Lambda_t)$, and leave other modules of the system as before. The proposed algorithm computes the log likelihood scores at $t_\delta(i) = \{2, 4, 6\}_{i=1}^{3}$, and selects the largest score as the decoding result.

The experimental results are listed in Table 1 for the 100 speakers. The performance is measured by the average individual equal-error rates (EER's) with SD thresholds and by the speedup in computing the target score. The second column is the system EER's when the final decision is made on the target LL score, $L(\mathbf{O}, \Lambda_t)$. The third column is the EER's when the decision is made on the LLR score as defined in Eq. (7). The last column is the estimated speedups on computing the target score using Eq. (5). The results show that the proposed algorithm is about 7.6 times faster than the full-search Viterbi algorithm on this task while the EER's are still acceptable for many applications. By applying the proposed decoding algorithm, the EER's on the LL scores only increased slightly while the EER's on the LLR scores increased a lot although it is still in an acceptable range. This can be improved by applying different t_δ values to different states during the decoding. An algorithm which can determine t_δ precisely is needed in the future research.

Table 1: Results in Average Individual EER's

Algorithms	LL Scores	LLR Scores	Estimated Speedups
Full-search Viterbi	4.84%	2.09%	1.0
Proposed algorithm	5.26%	3.07%	7.6

Since this experiment is to evaluate the proposed algorithm, we did not include model adaptation in this experiments. Also, in the target score computation, we used the given voice/silence end-points from SI decoding and compute the score on the detected voice portion of the pass-phrase. The actual EER's can be lower on both algorithms if the target score is computed by force decoding on the entire pass-phase using the SD target model and a silence model as reported in [4]. The length of a typical pass-phrase in the database is about 3 seconds, i.e. 300 frames, including silence and voice, when the target model has 25 states and the silence model has 3 states as in the above experiment, the estimated speedup for target score computation is 9.7.

For the background model consisted of 21 phonemes and 2 silence models with 3 states on each of the models, the estimated speedup is 14.6. From our experiments, the beam search algorithm for HMM decoding can not provide such large speedup.

3.2. Sequential End-Point Detection

In the above section, we discussed the proposed algorithm on an HMM application. In this section, we investigate the feasibility on other applications, such as speaker identification (SID), audio segmentation, silence/voice segmentation, etc. These applications can be considered as a detection problem between two models or among more than two models. In SID, one speaker is usually modeled by one Gaussian mixture model (GMM). In silence/voice segmentation, one GMM can be used to model silence, another one can be the fist or the last state of a voice HMM. Evaluating the detection performance is similar to evaluate the end-point detection between two states in one HMM since each HMM state can be considered as one GMM. Here, we use the end-point between the first and the second states in the above experiment as an example.

Table 2: Comparison in End-Point Detection with the Viterbi Algorithm

Differences	0 frames	1 frames	> 1 frames
True speakers	93.90%	3.38%	2.72%
Impostors	87.28%	6.65%	6.07%

Assuming the detected end-point form the Viterbi algorithm and the proposed sequential algorithm is $\mathbf{E}_{\text{Viterbi}}$ and $\mathbf{E}_{\text{Seq}}$ respectively. The end-point difference is defined as $|\mathbf{E}_{\text{Viterbi}} - \mathbf{E}_{\text{Seq}}|$. The experimental results are shown in Table 2. Among 3970 tested pass-phrases from true speakers, the proposed algorithm provides the exactly the same end points as the Viterbi algorithm on 93.90% of the tested pass-phrases, and 3.38% with one frame difference. Totally, 19608 pass-phrases from the impostors are evaluated. The difference on the impostor's pass-phrase is larger due to the mismatched model. Actually, this can be considered as an advantage to the system performance. It implies that the impostors' scores on the target model will be lower since the end point from the Viterbi algorithm has the highest LL score. The true speaker's experiment is similar to the applications in SID, silence/voice segmentation, etc. because the two GMM's are known. In which, 97.28% of the detected end-points are within one frame difference.

4. DISCUSSIONS AND CONCLUSIONS

The proposed decoding algorithm has more advantages than what we can show from the above experiment. For example, in verbal information verification [8], an utterance level decision score is a combination of a sequence of subword scores. By applying the sequential algorithm, the subword scores can be obtained and evaluated with minimal time delay. Therefore, an impostor's utterance might be rejected before the decoding process reaches the end of the utterance.

In conclusions, the fast, sequential decoding algorithm proposed in [1] was evaluated on a fixed-phrase SV system using a large database. The algorithm is based on a sequential detection scheme which is asymptotically optimum and is consistent with the definition of left-to-right HMM. The experimental results show that the proposed algorithm can provide acceptable EER's while the decoding speech is about 7 to 10 times faster than the Viterbi algorithm. Therefore, it has the potential to reduce the system cost to a factor of 7 to 10. The proposed algorithm can also be applied to SID, voice/silence detection, and many other applications.

5. ACKNOWLEDGMENT

The author wish to thank Chin-Hui Lee, Wu Chou, and Olivier Siohan for useful discussions.

6. REFERENCES

[1] Q. Li, "A fast decoding algorithm based on sequential detection of the changes in distribution," in *Proc. Int'l Conf. on Spoken Language Processing*, (Sydney), Nov. 1998.

[2] S. Parthasarathy and A. E. Rosenberg, "General phrase speaker verification using sub-word background models and likelihood-ratio scoring," in *Proceedings of ICSLP-96*, (Philadelphia), Oct. 1996.

[3] Q. Li, S. Parthasarathy, and A. E. Rosenberg, "A fast algorithm for stochastic matching with application to robust speaker verification," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Munich), pp. 1543–1547, April 1997.

[4] Q. Li and B.-H. Juang, "Speaker verification using verbal information verification for automatic enrollment," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Seattle), May 1998.

[5] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, pp. 100–115, 1954.

[6] R. K. Bansal and P. Papantoni-Kazakos, "An algorithm for detecting a change in stochastic process," *IEEE Trans. Information Theory*, vol. IT-32, pp. 227–235, March 1986.

[7] G. Lorden, "Procedures for reacting to a change in distribution," *The Annals of Mathematical Statistics*, vol. 42, no. 6, pp. 1897–1908, 1971.

[8] Q. Li, B.-H. Juang, Q. Zhou, and C.-H. Lee, "Verbal information verification," in *Proceedings of EUROSPEECH*, (Ghode, Greece), pp. 839–842, Sept. 22-25 1997.