# A FAST DECODING ALGORITHM BASED ON SEQUENTIAL DETECTION OF THE CHANGES IN DISTRIBUTION

*Qi Li*

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

## ABSTRACT

A fast algorithm for left-to-right HMM decoding is proposed in this paper. The algorithm is developed 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. The scheme is extended to HMM decoding in determining the state segmentations for likelihood or other score computations. As a sequential scheme, it can determine a state boundary in a few time steps after it occurs. The examples in this paper show that the proposed algorithm is 5 to 9 times faster than the Viterbi algorithm while it still can provide the same or similar decoding results. The proposed algorithm can be applied to speaker recognition, audio segmentation, voice/silence detection, and many other applications, where an assumption of the algorithm is usually satisfied.

## 1. INTRODUCTION

Hidden Markov Model (HMM) has been widely used in speaker and speech recognitions. In order to determine HMM state segmentations or compute likelihood scores, a decoding algorithm is needed. The algorithm is important since it takes the majority of the computation in applications. A fast decoding algorithm not only means fast response for recognition but also provides a better performance when computational resource and time are limited. For example, in speaker verification, a fast decoding algorithm means fast response, and more users and channels can be supported given the same, limited hardware.

The Viterbi algorithm is the prevalent HMM decoding algorithm. The concept of the Viterbi algorithm was from graph and network theory, and the HMM decoding problem was solved as the shortest-route problem, which has been well studied, such as Dijkstra's algorithm [1] and many others [2]. The Viterbi algorithm [3] provides an optimum solution to the problem of determining the state segmentation of an HMM in the sense of maximum likelihood [3, 4].

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 a left-to-right HMM. Following the definition of HMM, given a sequence of observations, we can determine the state segmentations by detecting the changes in distribution sequentially. In this paper, we propose an algorithm based on a sequential detection scheme which has an asymptotically optimum property.

Wald [5] introduced the concept of sequential test and formulated *sequential probability ratio test* (SPRT). The test was designed to decide between two simple hypotheses sequentially. Given two constants as the upper and the lower stopping thresholds and the density functions, $p_1$ and $p_2$ of two hypotheses, $H_1$ and $H_2$, respectively, by observing the data and computing the accumulated log likelihood ratio sequentially, SPRT can make a decision on either continuing the observation or stopping the test in accepting $H_1$ or $H_2$.

Using the sequential test to detect a change in distributions was first proposed by Page [6, 7], for memoryless processes. Its asymptotic properties were studied by Lorden [8]. The general form of the test was proposed by Bansal [9] and Bansal et al [10]. They also studied its asymptotic properties for stationary and ergodic process under some general regularity conditions. It has been proved that the test is asymptotically optimum in the sense that it requires the minimum expected sample size for decision, subject to a false alarm constraint [8, 10, 11].

The Page algorithm needs a pre-determined threshold value for decision. It may not be so critical if only the changes between two density functions needs to be determined, but, for HMM decoding, we have to detect the changes between many different density functions and the threshold values are usually not available. To solve the problem, we proposed an algorithm which can make the decision based on a common threshold value with a time constraint for different state pairs, instead of using multiple thresholds.

The proposed algorithm is much faster and can provide the same or similar results as the Viterbi algorithm for the examples tested in the paper. It is especially useful for real-time speaker recognition including speaker verification and identification [12, 13, 14], where the duration of each state is longer enough to meet an assumption of the proposed algorithm. The sequential algorithm is also useful to parallel processing, where state level scores, e.g. likelihood or others [15], can be calculated without waiting to the end of the decoding.

## 2. DETECTING A CHANGE IN DISTRIBUTION

Let $\mathrm{o}_n$ denote an observation vector at time $n$, and $p_1(\mathrm{o}_n)$ and $p_2(\mathrm{o}_n)$ 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, $\mathrm{O} = \{\mathrm{o}_n; n \geq 1\}$, and the density functions $p_1(\mathrm{o}_n)$ and $p_2(\mathrm{o}_n)$, the objective is to detect a possible $p_1$ to $p_2$ change as *reliably and quickly* as possible. Since the change can be happened at any time, we need a sequential detection scheme.

To gain insight, a non-sequential detection scheme was used in [10, 11]. Initialy, the size $n$ of the observation sequence O is fixed. We assume that the change occurrences are equally probable, then the $p_1$ to $p_2$ change occurs right after the data point $\mathrm{o}_j$; $1 \leq j \leq n$, if and only if

$$\sum_{i=j+1}^{n} \log \frac{p_2(\mathrm{o}_i|\mathrm{o}_{j+1}^{i-1})}{p_1(\mathrm{o}_i|\mathrm{o}_1^{i-1})} = \max_{1\leq k\leq n} \left\{ \sum_{i=k+1}^{n} \log \frac{p_2(\mathrm{o}_i|\mathrm{o}_{k+1}^{i-1})}{p_1(\mathrm{o}_i|\mathrm{o}_1^{i-1})} \right\}, \tag{1}$$

where

$$\begin{aligned} p_1(\mathrm{o}_1|\mathrm{o}_1^0) &= p_1(\mathrm{o}_1), \\ p_1(\mathrm{o}_{m+1}|\mathrm{o}_{m+1}^m) &= p_1(\mathrm{o}_{m+1}), \\ \sum_{i=n+1}^{n} \log \frac{p_2(\mathrm{o}_i|\mathrm{o}_{n+1}^{i-1})}{p_1(\mathrm{o}_i|\mathrm{o}_1^{i-1})} &= 0. \end{aligned}$$

Although the above scheme can be implemented by a fast algorithm, a sequential test procedure is still needed and it can be presented as follows.

Given the data block $\{\mathrm{o}^i\}_{i=1}^n$, decide in favor of the change from $p_1$ to $p_2$, iff

$$\sum_{i=1}^{\ell} R_i(\mathrm{o}^i) = \min_{1\leq k\leq n} \left\{ \sum_{i=1}^{k} R_i(\mathrm{o}^i) \right\}, \tag{2}$$

where

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

When data are observed sequentially, Page proposed an algorithm [6, 7] to decide that $p_1$ to $p_2$ change has occurred at the first $n$ such that

$$T(\mathbf{o}^n) = \sum_{i=1}^{n} R_i(\mathrm{o}^i) - \min_{1\leq k\leq n} \left\{ \sum_{i=1}^{k} R_i(\mathrm{o}^i) \right\} \geq \delta \tag{4}$$

where $\delta > 0$ is a pre-determined threshold. A recursive form for the above sequential test is

$$\begin{aligned} T(\mathrm{o}^0) &= 0; & (5) \\ T(\mathrm{o}^n) &= \max\{0, T(\mathrm{o}^{n-1}) + R_n(\mathrm{o}^n)\}, & (6) \end{aligned}$$

where, $p_1$ to $p_2$ change is occured at the first $n$ if $T(\mathrm{o}^n) \geq \delta$.

As pointed by Page [6], the above test breaks up into a repeated Wald sequential test with boundaries at $(0, \delta)$ and a zero initial score. It is asymptotically optimum in the sense that it requires the minimum possible expected sample size for decision, subject to a false alarm constraint. The related theorems and proofs can be found in [8] and [10].

The previous study was interested in detecting the occurrence of the change. We are also need to determine the point of the change for likelihood or other score computations. When $T(\mathrm{o}^n) > \delta$, the last data point of $p_1$ is

$$\ell = \arg\min_{1\leq k\leq n} \left\{ \sum_{i=1}^{k} R_i(\mathrm{o}^i) \right\}. \tag{7}$$

In many applications, it is difficult to determine the threshold value $\delta$. For example, in speech recognition, we may have over one thousand subword HMM's and each HMM has 3 states. Due to different speakers and different spoken contents, it is almost impossible to pre-determine all of the threshold values for every possible combination of connected states or every possible speaker. To apply the sequential scheme in speech applications, we propose a detection scheme as follows, which does not need to pre-determine the threshold value, $\delta$, precisely.

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

$$n - \ell \geq t_\delta, \tag{8}$$

*and*

$$T(\mathbf{o}^n) = \sum_{i=1}^{n} R_i(\mathbf{o}^i) - \min_{1\leq k\leq n} \left\{ \sum_{i=1}^{k} R_i(\mathbf{o}^i) \right\} > \varepsilon, \tag{9}$$

*where $\varepsilon \geq 0$ is a small number or can be just zero as we used in the examples in this paper, $R_i(\mathrm{o}^i)$ is defined as in Eq. (3). The last point $\ell$ of $p_1$ can be calculated using Eq. (7). Here, we assume that the duration of $p_2$ is not less than $t_\delta$.*

An illustration of the proposed scheme is shown in Fig. 1 (a), where $t_\delta$ in Eq. (8) is a time threshold representing a time duration, and $\delta$ in Eq. (4) represents a threshold value of the accumulated log likelihood ratio. It is much easier to determine $t_\delta$ than $\delta$ in speech and speaker recognition, and a common $t_\delta$ can be applied to different HMM's and different states. Generally speaking, a larger $t_\delta$ can give a more reliable change point, but it may delay the decision and cost more in computation. Also, $t_\delta$ should be equal to or less than the duration of $p_2$. The examples in this paper show that the proposed scheme can obtain exactly the same state segmentations as the Viterbi algorithm when $t_\delta \geq 2$.

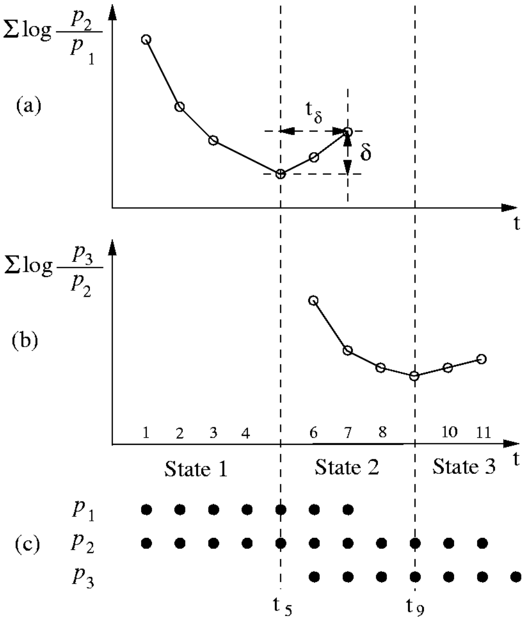

Figure 1: The scheme of the proposed decoding algorithm: (a) the end point detection for state 1, $t_5$; (b) the end point detection for state 2, $t_9$; and (c) the grid points for $p_1$, $p_2$ and $p_3$ computations (dots).

## 3. THE PROPOSED ALGORITHM FOR HMM DECODING

We have introduced the scheme of detecting the change between two stochastic processes. In this session, we apply the proposed scheme to HMM decoding. We focus our discussion on the left-to-right HMM since it is the most popular HMM structure in speech and speaker recognition [16].

For a left-to-right HMM, multiple state segmentations 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. Given $t_\delta$, Eq. (8) and Eq. (9) are evaluated at each step sequentially. At $t = t_7$, we have $t_7 - t_5 \geq t_\delta = 2$ and $T(o^7) > \varepsilon \geq 0$. 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).

We note that the proposed decoding scheme is based on the assumption that the duration of the next state (the number of frames in the next state) is no less than $t_\delta$. Many applications, such as speaker verification, speaker identification, audio segmentation, etc., can normally meet this assumption. When the assumption can not be satisfied as in some speech recognition examples, further evaluation on the next states is necessary. It will be discussed separately.

For the proposed algorithm, the number of additions in HMM decoding is in the order of

$$2\left[T+(N-1)t_\delta\right](C+2) \approx 2C\left[T+(N-1)t_\delta\right], \qquad (10)$$

where $C$ is the number of float point operations at each grid point for log probability, $C+2$ includes the accumulation and the ratio computations, $N$ is the total number of states, $T$ is the total number of frames, and $t_\delta$ is the time threshold. A widely used implementation of a full-search Viterbi algorithm for the left-to-right model needs

$$NT(C+1)+T \approx NTC \qquad (11)$$

additions. Therefore, the speedup of the proposed algorithm is in the order of

$$\frac{NT}{2\left[T+(N-1)t_\delta\right]}. \qquad (12)$$

## 4. EXPERIMENTS

**Example 1:** This is a forced decoding problem from speaker verification [12, 13, 14]. In a training session, a speaker dependent left-to-right HMM is trained with 14 sates and each state has 4 Gaussian mixtures for a pass-phrase "open sesame". In a test session, we need to decode the given test utterance into a sequence of states and calculate likelihood scores. The input is a sequence of 24 dimensional features of cepstral and delta-cepstral coefficients derived from a 10th order LPC analysis over a 30 ms widow updated at 10 ms intervals. For this example, we have 100 cepstral frames and the proposed algorithm gives the exactly same result as the Viterbi algorithm as long as $t_\delta \geq 2$, where $\varepsilon = 0$. The computations in the number of floating point operations (flops) are listed in Table 1. The proposed algorithm is about 5 times faster than the Viterbi algorithm.

Table 1: **Comparisons on Computation**

| | Viterbi Algorithm | Proposed Algorithm | Speedup |
|---|---|---|---|
| Example 1 | 785.5 Flops | 151.5 Flops | 5.2 |
| Example 2 | 29.01 Mflops | 3.05 Mflops | 9.5 |

**Example 2:** This example is to verify the proposed algorithm in the case of each state only has a few frames, e.g. 2 to 10 frames in one state. Now, the given utterance, "open sesame", with 101 frames is decoded into 10 subwords (phonemes). Each of the subword HMM has 3 states. When we do a forced decoding, we concatenate the states of all the subwords as a sequence of 30 states. For this example, the proposed algorithm gave the exactly same result as the Viterbi decoding ($t_\delta = 2$ and $\varepsilon = 0$). The computations for the Viterbi and the proposed algorithms are 27.93 and 2.93 Mflops respectively. Therefore, the proposed algorithm has a speedup of 9.5 approximately, as shown in Table 1.

## 5. CONCLUSIONS

This paper proposed a sequential decoding algorithm based on an asymptotically optimal detection scheme. The algorithm is consistent with the definition of left-to-right HMM. Compared to the Viterbi algorithm for HMM decoding, it has several advantages, although it is not an optimal algorithm in the sense of maximum likelihood. First, it is a sequential algorithm. It can determine a state boundary in a few time steps after it occurs, which is useful to real-time speaker verification, speaker identification, language identification, audio segmentation, silence/voice detection, and other applications. Second, it needs less computation. For example, it can provide faster response or support more channels for speaker verification when the computational resource is limited. Last, the implementation is easier.

We note that this paper presents a preliminary concept for a different decoding approach for speech processing. It still needs further tests before it can be applied to real-world applications.

## 6. ACKNOWLEDGMENT

The author wish to thank Chin-Hui Lee and Wu Chou for reading the draft of this paper and useful discussions.

## 7. REFERENCES

[1] J. R. Evans and E. Minieka, *Optimization algorithms for networks and graphs*. NY: Marcel Dekker, 1992.

[2] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, pp. 269–271, 1959.

[3] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, pp. 260–269, April 1967.

[4] G. D. Forney, "The Viterbi algorithm," *Proceeding of IEEE*, vol. 61, pp. 268–278, March 1973.

[5] A. Wald, *Sequential analysis*. NY: Chapman & Hall, 1947.

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

[7] E. S. Page, "A test for a change in a parameter occuring at an unknown point," *Biometrika*, vol. 42, pp. 523–527, 1955.

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

[9] R. K. Bansal, "An algorithm for detecting a change in stochastic process," Master Thesis, University of Connecticut, EECS Dept., 1983.

[10] 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.

[11] D. Kazakos and P. Papantoni-Kazakos, *Detection and Estimation*. NY: Computer Science Press, 1990.

[12] 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), October 1996.

[13] 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.

[14] 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.

[15] 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.

[16] C.-H. Lee and L. R. Rabiner, "A frame-synchronous network search algorithm for connected word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 1649–1658, November 1989.