# SEARCH-SPACE REDUCTION FOR FAST, OPTIMAL HMM DECODING IN SPEAKER VERIFICATION

*Qi Li*

Multimedia Communications Research Laboratory
Bell Labs, Lucent Technologies
600 Mountain Avenue, Murray Hill, NJ 07974, USA

## ABSTRACT

Currently, the most popular algorithm for hidden Markov model (HMM) decoding is the Viterbi algorithm with beam search to reduce search space. However, it is a difficult problem in determining a beam width beforehand. To address this problem, we propose a novel approach on search space reduction. Following the definition of HMM, we first detect the possible change points between HMM states sequentially, then use the change points to locate a subspace for searching. Using a combined forward and backward scheme, we can detect two boundaries consisting of change points to enclose the subspace. The Viterbi algorithm or any other search algorithms can then be applied in the subspace. The experiments on a speaker verification task show that the proposed algorithm is about 4 times faster than a full search algorithm while the accuracy is almost the same. On the same decoding speed, the proposed algorithm provides a better accuracy than a beam-search approach. For an HMM with $S$ states, the upper bound of speedup comparing to the full search approach is approximately $S/3$.

## 1. INTRODUCTION

Hidden Markov Model (HMM) has been widely used in speech and speaker recognition in which the non-stationary speech signal is represented as a sequence of states. In recognition, given an utterance and a set of HMMs, a decoding algorithm is then needed to search for the optimal state path, such that the overall likelihood score of the utterance is maximum. The decoding algorithm is very important to any speech or speaker recognition system since it is the fundamental computation and the algorithm can affect the system performance significantly.

Generally speaking, there are two basic requirements for a decoding algorithm – accuracy and speed. Several search algorithms have been developed. They are Viterbi search, stack decoders, multi-pass search, forward-backward search, state-detection search, etc. The Viterbi algorithm [2] is optimal in the sense of maximum likelihood. Therefore, it meets the first above requirement. However, a full Viterbi search is almost impractical due to the large search space in speech decoding. There are two major approaches to address the speed problem. One way is to change the optimal algorithm to a near optimal one in order to gain the decoding speed (e.g. [1]) but it may lose some accuracy. Another way is to keep the optimal decoding algorithm while trying to reduce the search space. The most popular one is the beam-search algorithm (e.g. [3]). It reduces the search space by pruning the search paths with low likelihood scores using a pre-determined beam width. Obviously, it improves the decoding speed due to the reduced search space, but it is difficult to determine the beam width beforehand. When the value of beam width is too large, the decoder can provide a better accuracy but it slows down the speed; when the beam width value is too small, the decoder is faster but it may give poor accuracy. To address this problem, we propose an algorithm which can determine a subspace from the constraints of HMM without the need of a beam or any other threshold. It would not miss the optimal path in the case that the utterance matches the HMM. In an imposter case, the algorithm can limit the search space therefore it has the potential to decrease the impostor's likelihood scores. Once a subspace is determined, a search algorithm such as the Viterbi algorithm or any other search algorithm based on dynamic programming can be applied to find the optimal path in the subspace.

## 2. HMM STATE CHANGE-POINT DETECTION

An HMM can be completely characterized by a matrix of state transition probabilities, $A$, observation densities, $B$, and initial state probabilities, $\Pi$.

$$\lambda = \{A, B, \Pi\} = \{a_{i,j}, b_i, \pi_i\}, \quad i, j = 1, ..., S, \qquad (1)$$

where $N$ is the total number of states. Given an observation vector $\mathbf{o}_t$ the continuous observation density, for state $j$ is

$$b_j(\mathbf{o}_t) = p_j(\mathbf{o}_t) \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mathbf{o}_t, \mu_{jm}, \Sigma_{jm}), \qquad (2)$$

where $M$ is the total number of the Gaussian components $\mathcal{N}(.)$, $\mu_{jm}$ and $\Sigma_{jm}$ are the mean vector and covariance matrix of the $m$th component at state $j$, respectively

As we proposed in [1], to detect the change point between states is similar to the task of detecting the change point between two data distributions. For a left-to-right HMM, it can be implemented by repeating the following procedure until obtaining the last change point between state $S-1$ and $S$.

Given a time threshold $t_\delta > 0$. we observe data sequentially at time $t$, and decide that the change from state $s$ to $s+1$ occurs, if

$$t - \ell_s \geq t_\delta, \tag{3}$$

and

$$T(\mathbf{o}_t) = \sum_{i=\ell_{s-1}+1}^{t} R(\mathbf{o}_i) - \min_{\ell_{s-1}<k\leq t} \left\{ \sum_{i=\ell_{s-1}+1}^{k} R(\mathbf{o}_i) \right\} > \varepsilon, \tag{4}$$

where $\varepsilon \geq 0$, $\ell_{s-1}$ is the end point of the last state,

$$\ell_s = \arg \min_{\ell_{s-1}<k\leq t} \left\{ \sum_{i=\ell_{s-1}+1}^{k} R(\mathbf{o}_i) \right\}, \tag{5}$$

$$R(\mathbf{o}_i) = \frac{b_{s+1}(\mathbf{o}_i)}{b_s(\mathbf{o}_i)}, \quad s = 1, ..., S-1, \tag{6}$$

and $b_j(.)$ is defined in Eq. (2). It is straightforward to implement the above procedure in a recursive form. For the task of search space detection, we let $t_\delta = 1$ and $\varepsilon = 0$. Thus, the detector would not make any false rejection but with false acceptances. Nevertheless, the false acceptances can be resolved in the algorithm introduced in the next section. Also, the above detection algorithm can be extended to the cases where one or more state skip is allowed in HMM decoding.

## 3. HMM SEARCH SPACE REDUCTION

The entire search space in terms of grid points for HMM decoding can be defined as

$$\mathbf{\Psi} = \{(t, s_t) \mid 1 \leq t \leq T;\ 1 \leq s_t \leq S\}, \tag{7}$$

where $(t, s_t) \in \mathbf{\Psi}$ is a grid point for probability computation, $t$ is the frame number of feature vectors, $s_t$ is the state index at time $t$, $T$ and $S$ are the total numbers of frames and states, respectively. The probability value at each grid point can be computed by Eq. (2). The goal is to detect a subspace $\mathbf{\Omega} \subseteq \mathbf{\Psi}$, which includes the path with the maximum likelihood score under the constraint of the left-to-right HMM.

When applying the above state change-point detection algorithm with $t_\delta = 1$ and $\varepsilon = 0$ in a forward time scheme, i.e. from $t = 1$ to $t = T$, we can detect a sequence of state change points. The grid points along the sequence form a boundary in the search space, called *forward boundary* defined as

$$B^+ = \left\{(t, s_t^+) \mid s_t^+ \leq s_{t+1}^+, t = 1, ..., T\right\} \subset \mathbf{\Psi}, \tag{8}$$

where $s_t^+$ is the state index at time $t$ along the boundary. An example of the forward boundary is shown in Fig. 1 as the solid line. The grid points along the forward dished line and the solid line are involved in the forward detection.

On the other hand, if we detect the state change points in a backward time scheme, i.e. from $t = T$ to $t = 1$, we can detect another sequence of state change points. The grid points along the sequence form another boundary, called *backward boundary* defined as

$$B^- = \left\{(t, s_t^-) \mid s_t^- \leq s_{t+1}^-, t = 1, ..., T\right\} \subset \mathbf{\Psi}, \tag{9}$$

where $s_t^-$ is the state index at time $t$ along the boundary. An example of the backward boundary is again the grid points along the solid line in Fig. 1. The dished line from right to left indicates the direction of the backward sequential detection.

Generally speaking, neither one of the boundaries guarantees the optimal path since both of them may include false acceptances. However, the two boundaries enclose a subspace, consisting of the grid points inside and along the boundaries. If the following constraints hold,

$$\mathbf{\Omega} = \left\{(t, s_t) \mid 1 \leq t \leq T; s_t^- \leq s_t \leq s_t^+\right\}, \tag{10}$$

where $(t, s^+) \in B^+$, $(t, s^-) \in B^-$, and $\mathbf{\Omega} \subseteq \mathbf{\Psi}$, i.e., the grid points of the forward boundary are above the backward boundary at every time frame, the optimal path must be included in $\mathbf{\Omega}$ under the constraint of left-to-right HMM since neither one of the boundaries allows false rejection.

However, the constraint in Eq. (10) may not hold for every grid point, i.e., the grid points of $B^+$ may be located below $B^-$ at some time frames. This may be due to a skipped state, a mismatch between the model and data, or the utterance is from an imposter. In this special case, depending on applications, we can reject the utterance or perform a full search in $\mathbf{\Psi}$ or construct a search space as follows,

$$\mathbf{\Omega}' = \bigcup_{m=1}^{M} \phi_m \bigcup_{n=1}^{N} \psi_n, \tag{11}$$

where $\phi_m$ is a surrounded subspace,

$$\phi_m = \left\{(t, s_t) \mid t_i \leq t \leq t_j; s_{t_i}^- \leq s_t \leq s_{t_j}^+\right\}, \tag{12}$$

and $\psi_n$ is a rectangular subspace in which $B^+$ is under $B^-$,

$$\psi_n = \left\{(t, s_t) \mid t_i \leq t \leq t_j; s_{t_i}^+ \leq s_t \leq s_{t_j}^-\right\}. \tag{13}$$

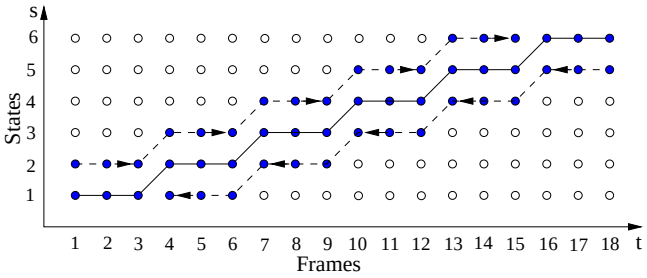


Figure 1: A single path (solid line) is detected from the forward and backward change-point detection.

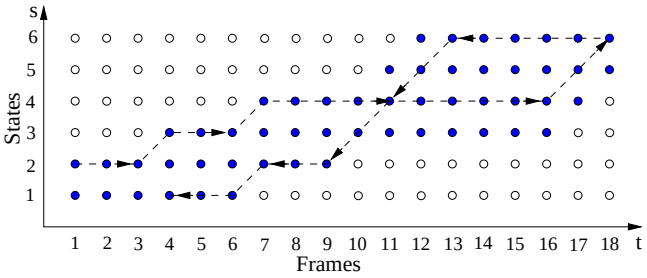


Figure 3: A special case locates between (11,4) and (18,6), where the forward boundary is under the backward one.

A search algorithm can then be employed in $\Omega$ or $\Omega'$ to find the optimal path. There are several typical cases of search space detection. We discuss them as follows. A real search space can be a combination of these cases.

**Case 1: Single Path in the Reduced Space**

If the forward and backward boundaries are identical, it indicates that there exists only a single path in the reduced search space, i.e. $B^+ = B^- = \Omega$. In this case, search is not necessary, and a maximum likelihood score can be computed from the path directly. An example is shown in Fig 1 as the solid line. The points along the dished lines are involved in the change point detection but they are excluded from $\Omega$.

**Case 2: Multiple Paths in a Local Area**

When multiple paths exist in a search space, the forward and backward boundaries may not meet in some local areas in $\Omega$. In this case, search is needed only for those local areas. An example is shown in Fig. 2 as the "hole" of 4 grid points, (8,3), (8,4), (9,3) and (9,4). We only need to search those 4 grid points.

**Case 3: Special Case**

In some special case, the forward boundary may be under the backward boundary in some areas, as shown in Fig. 3 between (11,4) to (18,6). This could be caused by the data which skip one HMM state or the data did not go through all HMM states. Most likely, the data are from an imposter, i.e. there is no match between the testing data and the HMM. Depend on applications, we can either reject the data set, or perform a search on either $\Psi$ or $\Omega'$. For the example in Fig. 3, $\Omega' = \phi \bigcup \psi$, where $\phi = \{(t, s_t) \mid 1 \leq t \leq 11; 1 \leq s_t \leq 4\}$, and $\psi_n$ is a rectangular subspace $\psi_n = \{(t, s_t) \mid 11 \leq t \leq 18; 4 \leq s_t \leq 6\}$.

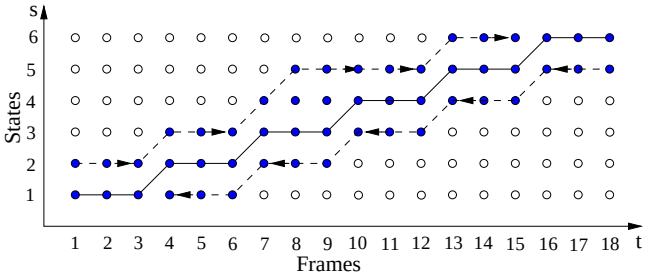


Figure 2: A "hole" is detected from the forward and backward state change-point detection. A search is needed only among 4 grid points, (8,3), (8,4), (9,3) and (9,4).

The algorithm can be summarized as follows:

1. Perform a forward state change-point detection and obtain a forward boundary, $B^+$.
2. Perform a backward state change-point detection and obtain a backward boundary, $B^-$.
3. If $B^+$ is above $B^-$ at all time frames, search for the optimal path in subspace $\Omega$, Eq. (10), if necessary, otherwise, search for the optimal path in $\Omega'$, Eq. (11).
4. Return the accumulated log-likelihood score. Return the state segmentation if necessary.

In the ideal case with only one single path existing in the reduced search space, the upper bound of the speedup of the proposed algorithm for an HMM with $S$ states is approximately $S/3$. In other words, the proposed algorithm can be $S/3$ times faster than a full search one in Case 1.

## 4. EXPERIMENTS

The proposed algorithm was compared with the beam-search and full-search algorithms on a speaker verification task with totally 3,970 utterances from true speakers and 19,608 utterances from impostors. The database with 100 speakers was evaluated on a common pass-phrase, "I pledge allegiance to the flag." Decoding is based on speaker-dependent HMMs with 25 states for the entire pass-phrase. Detail descriptions on the experimental database, features, model structures, and verification system can be found in [4]. The verification decision is based on a log-likelihood ratio test between the decoding score of the target model from this experiment and a given background score [4]. The experimental results are

Table 1: Comparisons on Decoding Performances

| | Pro-posed | Full Search | Beam Widths | | |
|---|---|---|---|---|---|
| | | | 500 | 300 | 200 |
| EER (%) | 2.07 | 2.09 | 2.18 | 2.24 | 2.29 |
| Complexity (MFlops) | 0.76 | 2.74 | 1.43 | 1.14 | 0.92 |
| Overhead (KFlops) | 6.1 | 8.9 | 9.8 | 8.8 | 8.1 |

listed in Tab.1. A summary on the comparison of the system accuracy and decoding speed is shown in Fig. 4. The results showed that the accuracy, in term of equal-error rates, of the proposed algorithm is almost the same as the full search algorithm and much better than the beam search algorithms on different beam widths of 200, 300, and 500, where a typical beam search algorithm was applied. If $(t, s_t^*)$ is the grid point with the best path at time $t$, and the log likelihood score is $L(t, s_t^*)$, the path to any $(t, s_t)$ will be a candidate for extension at frame $t+1$ only if $L(t, s_t) \geq L(t, s_t^*) - \theta$, where $\theta$ is called beam width and usually taken to be a constant. The decoding complexity of the algorithms is compared in million floating point operations (MFlops) and in terms of speedup which is defined as the ratio of MFlops between the full search and other algorithms. The proposed algorithm is about 4 times faster than the full search algorithm. Compared to the beam-search algorithms, the proposed one is either much faster when the accuracy is approximately the same or more accurate when the decoding speed is approximately the same. Decoding overhead was also evaluated in Flops. The proposed algorithm uses less Flops than other algorithms.

The slightly better accuracy compared to the full search is due to the search space reduction. As shown in Table 2, the average target scores of true speakers are almost the same while the average imposter score of the proposed algorithm is reduced due to the mismatch between model and utterances.

## 5. CONCLUSIONS

In this paper, we first introduced the algorithm on sequential HMM state change-point detection, then proposed an algorithm on search space reduction. In the algorithm, the detected state change points are used to form the boundaries of a search space. When two boundaries are detected from a forward and backward scheme, a subspace is determined,

Table 2: Comparisons on Average Target Scores

| Algorithms | Full Search | | Proposed | |
|---|---|---|---|---|
| Speakers | True | Impostor | True | Impostor |
| Scores | 12.12 | 0.92 | 12.10 | 0.83 |

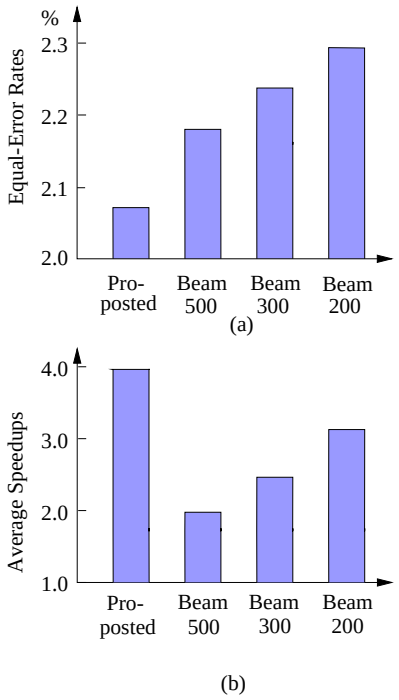


Figure 4: (a) Comparison on system equal-error rates (EER); (b) Comparison to the Viterbi algorithm on speedups.

and a search algorithm can then be applied to find the optimal path. Compared to the beam search algorithm, the proposed one is capable of determining a subspace without any beam width which is difficult to decide beforehand. The experiment in a speaker verification task showed that the proposed algorithm can provide much better accuracy than the beam search algorithm at a similar decoding speed. Compared to a full search algorithm, the proposed one is about 4 times faster with almost the same accuracy.

## 6. ACKNOWLEDGMENT

The author wish to thank Wu Chou for useful discussions.

## 7. 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*, Nov. 1998.

[2] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. on Info. Theory*, vol. IT-13, April 1967.

[3] H. Ney and S. Ortmanns, "Dynamic programming search," *IEEE Signal Processing Magazine*, vol. 16, no. 5, Sept. 1999.

[4] Q. Li and B.-H. Juang, "Speaker verification using verbal information verification for automatic enrollment," in *Proc. of IEEE ICASSP*, May 1998.