A VITERBI ALGORITHM FOR A TRAJECTORY MODEL DERIVED FROM HMM WITH EXPLICIT RELATIONSHIP BETWEEN STATIC AND DYNAMIC FEATURES

Heiga Zen, Keiichi Tokuda, Tadashi Kitamura

Department of Computer Science and Engineering, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466-8555 Japan
{zen,tokuda,kitamura}@ics.nitech.ac.jp

ABSTRACT

This paper introduces a Viterbi algorithm to obtain a sub-optimal state sequence for trajectory-HMM, which is derived from HMM with explicit relationship between static and dynamic features. The trajectory-HMM can alleviate some limitations of HMM, which are i) constant statistics within HMM state and ii) conditional independence of observations given the state sequence, without increasing the number of model parameters. The proposed algorithm was applied to state-boundary optimization for Viterbi training and N-best rescoring. In speaker-dependent continuous speech recognition experiment, trajectory-HMM with the proposed algorithm achieved about 14% error reduction over the standard HMM with the conventional Viterbi algorithm.

1. INTRODUCTION

Speech recognition technology has achieved significant progress with the introduction of the hidden Markov model (HMM). Its tractability and efficient implementations are achieved by some assumptions: i) constant statistics within an HMM state, ii) conditional independence assumption of observations given the state sequence. Although these assumptions make HMM practically useful, they are not realistic for modeling sequences of speech spectra, especially in spontaneous speech. To overcome the shortcomings of HMM, a lot of alternative models, called segment models, have been proposed, e.g., [1–9]. Although these models can improve the speech recognition performance, they generally require an increase in the number of model parameters and computational complexity. Alternatively, the use of the dynamic features (delta and delta-delta features) [10] also improves the performance of HMM-based speech recognizers. It is a simple mechanism for capturing time dependencies, however, it has been thought of as an ad hoc, not an essential solution.

In [11], a trajectory model, called trajectory-HMM, was derived by reformulating the standard HMM whose state output vector includes static and dynamic feature parameters. The standard HMM with static and dynamic features allows inconsistent statistics between the model parameters for static and dynamic features. By imposing the explicit relationship between them, the standard HMM is naturally translated into a trajectory model. The trajectory-HMM can overcome the limitations in the standard HMM framework without any additional parameters. In addition, trajectory-HMM provides a computational model for coarticulation and the dynamics of human speech. A Viterbi-type training algorithm was also derived in [11]. However, the lack of an algorithm to obtain the most likely state sequence does not permit iterating Viterbi training procedure and designing a Viterbi decoder.

In this paper, a Viterbi algorithm to obtain a sub-optimal state sequence for trajectory-HMM is proposed. This is the first step for developing a Viterbi decoder based on trajectory-HMM framework. The proposed algorithm is applied to state-boundary optimization for Viterbi training and N-best rescoring.

The rest of this paper is organized as follows. Section 2 defines the trajectory-HMM. In Section 3 the time-recursive likelihood computation and a Viterbi algorithm for trajectory-HMM are derived. Results of continuous speech recognition experiment are shown in Section 4. Concluding remarks and future plans are presented in the final section.

2. REFORMULATING HMM AS TRAJECTORY-HMM

The output probability of a speech parameter vector sequence $\boldsymbol{o} = \left[\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, \ldots, \boldsymbol{o}_T^\top\right]^\top$ for the standard HMM is given by

$$P(\boldsymbol{o} \mid \lambda) \;=\; \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{o} \mid \boldsymbol{q}, \lambda)\, P(\boldsymbol{q} \mid \lambda)\,, \tag{1}$$

where $\boldsymbol{q} = \{q_1, q_2, \ldots, q_T\}$ is a state sequence. We assume that the speech parameter vector $\boldsymbol{o}_t$ consists of the static feature vector $\boldsymbol{c}_t = [c_t(1), c_t(2), \ldots, c_t(M)]^\top$ (e.g., cepstral coefficients), and dynamic feature vectors $\Delta\boldsymbol{c}_t$, $\Delta^2\boldsymbol{c}_t$ (e.g., delta and delta-delta cepstral coefficients), respectively, that is $\boldsymbol{o}_t = [\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top]^\top$, where the dynamic feature vectors are calculated by

$$\Delta\boldsymbol{c}_t \;=\; \sum_{\tau=-L_-^{(1)}}^{L_+^{(1)}} w^{(1)}(\tau)\boldsymbol{c}_{t+\tau}\,, \qquad \Delta^2\boldsymbol{c}_t \;=\; \sum_{\tau=-L_-^{(2)}}^{L_+^{(2)}} w^{(2)}(\tau)\boldsymbol{c}_{t+\tau}\,. \tag{2}$$

Accordingly, when each state output probability distribution is assumed to be a single Gaussian, $P(\boldsymbol{o} \,|\, \boldsymbol{q}, \lambda)$ is given by

$$P(\boldsymbol{o} \,|\, \boldsymbol{q}, \lambda) \;=\; \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t \,|\, \boldsymbol{\mu}_{q_t}, \boldsymbol{\Sigma}_{q_t}) \;=\; \mathcal{N}(\boldsymbol{o} \,|\, \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}})\,, \tag{3}$$

where $\boldsymbol{\mu}_{q_t}$ and $\boldsymbol{\Sigma}_{q_t}$ are the $3M \times 1$ mean vector and the $3M \times 3M$ covariance matrix, respectively, associated with q_t-th state, and

$$\boldsymbol{\mu}_{\boldsymbol{q}} \;=\; \left[\boldsymbol{\mu}_{q_1}^\top, \boldsymbol{\mu}_{q_2}^\top, ..., \boldsymbol{\mu}_{q_T}^\top\right]^\top \tag{4}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{q}} \;=\; \operatorname{diag}\left[\boldsymbol{\Sigma}_{q_1}, \boldsymbol{\Sigma}_{q_2}, ..., \boldsymbol{\Sigma}_{q_T}\right]\,. \tag{5}$$

Conditions (2) can be arranged in a matrix form:

$$\boldsymbol{o} \;=\; \boldsymbol{W}\boldsymbol{c}, \tag{6}$$

where

$$\begin{aligned} \boldsymbol{c} &= [\boldsymbol{c}_1^\top, \boldsymbol{c}_2^\top, \ldots, \boldsymbol{c}_T^\top]^\top & (7)\\ \boldsymbol{W} &= [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T]^\top & (8)\\ \boldsymbol{w}_t &= \left[\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}\right] & (9)\\ \boldsymbol{w}_t^{(n)} &= [\, \underset{\text{1st}}{\boldsymbol{0}_{M\times M}}, \ldots, \boldsymbol{0}_{M\times M}, \underset{(t-L_-^{(n)})\text{-th}}{w^{(n)}(-L_-^{(n)})\boldsymbol{I}_{M\times M}}, \ldots, \underset{t\text{-th}}{w^{(n)}(0)\boldsymbol{I}_{M\times M}}, \\ &\quad \ldots, \underset{(t+L_+^{(n)})\text{-th}}{w^{(n)}(L_+^{(n)})\boldsymbol{I}_{M\times M}}, \boldsymbol{0}_{M\times M}, \ldots, \underset{T\text{-th}}{\boldsymbol{0}_{M\times M}}\,]^\top, \quad n = 0, 1, 2 & (10) \end{aligned}$$

$L_-^{(0)} = L_+^{(0)} = 0$, and $w^{(0)}(0) = 1$. Thus, $P(\boldsymbol{o} \mid \boldsymbol{q}, \lambda)$ can be viewed as a function of $\boldsymbol{c}$ as follows:

$$\begin{aligned} P(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{q}, \lambda) &= \mathcal{N}(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}) \\ &= K_{\boldsymbol{q}} \cdot \mathcal{N}(\boldsymbol{c} \mid \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}) \,, & (11) \end{aligned}$$

where $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$, $\boldsymbol{P}_{\boldsymbol{q}}$ and $K_{\boldsymbol{q}}$ are given by

$$\begin{aligned} \boldsymbol{R}_{\boldsymbol{q}}\bar{\boldsymbol{c}}_{\boldsymbol{q}} &= \boldsymbol{r}_{\boldsymbol{q}} & (12)\\ \boldsymbol{R}_{\boldsymbol{q}} &= \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W} = \boldsymbol{P}_{\boldsymbol{q}}^{-1} & (13)\\ \boldsymbol{r}_{\boldsymbol{q}} &= \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}} & (14)\\ K_{\boldsymbol{q}} &= \frac{\sqrt{(2\pi)^{MT}|\boldsymbol{P}_{\boldsymbol{q}}|}}{\sqrt{(2\pi)^{3MT}|\boldsymbol{\Sigma}_{\boldsymbol{q}}|}} \cdot \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{\boldsymbol{q}}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}} - \boldsymbol{r}_{\boldsymbol{q}}^\top \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}}\right)\right\} \,. & (15) \end{aligned}$$

From the above expression (11), trajectory-HMM is defined :

$$P(\boldsymbol{c} \mid \lambda) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{c} \mid \boldsymbol{q}, \lambda)\, P(\boldsymbol{q} \mid \lambda) \,, \qquad (16)$$

where

$$P(\boldsymbol{c} \mid \boldsymbol{q}, \lambda) = \mathcal{N}(\boldsymbol{c} \mid \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}) \,. \qquad (17)$$

In this paper, Eq. (17) is referred to as “trajectory likelihood”. Interestingly, the mean $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ is exactly the same as the speech parameter trajectory obtained by the speech parameter generation technique (Case 1 in [12]), that is,

$$\bar{\boldsymbol{c}}_{\boldsymbol{q}} = \arg\max_{\boldsymbol{c}} P(\boldsymbol{o} \mid \boldsymbol{q}, \lambda) = \arg\max_{\boldsymbol{c}} P(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{q}, \lambda) \,. \qquad (18)$$

By assuming $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ is the mean for the spectral parameter vector sequence $\boldsymbol{c}$, corresponding to an utterance, the standard HMM can naturally be translated into a trajectory model. It is also noted that the spectral parameter vector sequence $\boldsymbol{c}$ is modeled by a mixture of Gaussian distributions whose dimensionality is TM, and their covariances $\boldsymbol{P}_{\boldsymbol{q}}$ are generally full. Because of the above characteristics of the mean vector $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ and covariance matrix $\boldsymbol{P}_{\boldsymbol{q}}$, trajectory-HMM can avoid the limitations of HMM.

3. VITERBI ALGORITHM FOR TRAJECTORY-HMM

3.1. Time recursive likelihood computation

To compute “trajectory likelihood” directly, high dimensional linear algebra (e.g., matrix inversion, determinant, etc.) is required. To avoid it, a time-recursive algorithm to compute “trajectory likelihood” is described in this section.

From Eq. (11), “trajectory likelihood” is given by

$$P(\boldsymbol{c} \mid \boldsymbol{q}, \lambda) = K_{\boldsymbol{q}}^{-1} \cdot P(\boldsymbol{o} \mid \boldsymbol{q}, \lambda) \,. \qquad (19)$$

In Eq. (15), although $\left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|$ and $\boldsymbol{\mu}_{\boldsymbol{q}}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}} \boldsymbol{\mu}_{\boldsymbol{q}}$ can be computed time-recursively, it is difficult to compute $\left|\boldsymbol{P}_{\boldsymbol{q}}\right|$ and $\boldsymbol{r}_{\boldsymbol{q}}^\top \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}}$ recursively because of the temporal full-covariance matrix $\boldsymbol{P}_{\boldsymbol{q}}$. However, by using the special structure of $\boldsymbol{P}_{\boldsymbol{q}}$, “trajectory likelihood” can be computed in a time-recursive manner.

First, when $\Delta\boldsymbol{c}_t$ and $\Delta^2\boldsymbol{c}_t$ are computed as regression coefficients from $\{\boldsymbol{c}_{t-L}, \ldots, \boldsymbol{c}_{t+L}\}$, $\boldsymbol{R}_{\boldsymbol{q}}$ becomes a $(4L + 1)$-diagonal symmetric positive define matrix. Accordingly, $\boldsymbol{R}_{\boldsymbol{q}}$ can be decomposed by Cholesky decomposition :

$$\boldsymbol{R}_{\boldsymbol{q}} = \boldsymbol{U}_{\boldsymbol{q}}^\top \boldsymbol{U}_{\boldsymbol{q}} \,, \qquad (20)$$

where $\boldsymbol{U}_{\boldsymbol{q}}$ is an upper $(2L+1)$-band triangular matrix. From Eq. (20), $\left|\boldsymbol{P}_{\boldsymbol{q}}\right|$ can be rewritten as

$$\left|\boldsymbol{P}_{\boldsymbol{q}}\right| = \left|\boldsymbol{R}_{\boldsymbol{q}}\right|^{-1} = \left|\boldsymbol{U}_{\boldsymbol{q}}^\top \boldsymbol{U}_{\boldsymbol{q}}\right|^{-1} = \left|\boldsymbol{U}_{\boldsymbol{q}}\right|^{-2} = \prod_{t=1}^{T} \left|\boldsymbol{U}_{\boldsymbol{q}_{t+L}}^{(t,t)}\right|^{-2} \,, \qquad (21)$$

where $\boldsymbol{q}_{t+L} = \{q_1, \ldots, q_{t+L}\}$. Since $\boldsymbol{U}_{\boldsymbol{q}_{t+L}}^{(t,t)}$ depends only on the state sequence from time 1 to $t + L$, $\left|\boldsymbol{P}_{\boldsymbol{q}}\right|$ can be computed time-recursively.

Secondly, from Eqs. (12), (13), and (20), $\boldsymbol{r}_{\boldsymbol{q}}^\top \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}}$ can be rewritten by

$$\begin{aligned} \boldsymbol{r}_{\boldsymbol{q}}^\top \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}} &= \boldsymbol{r}_{\boldsymbol{q}}^\top \boldsymbol{P}_{\boldsymbol{q}}^\top \boldsymbol{R}_{\boldsymbol{q}} \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}} = \bar{\boldsymbol{c}}_{\boldsymbol{q}}^\top \boldsymbol{U}_{\boldsymbol{q}}^\top \boldsymbol{U}_{\boldsymbol{q}} \bar{\boldsymbol{c}}_{\boldsymbol{q}} & (22)\\ &= \boldsymbol{g}_{\boldsymbol{q}}^\top \boldsymbol{g}_{\boldsymbol{q}} \qquad \left(\boldsymbol{g} = \boldsymbol{U}_{\boldsymbol{q}} \bar{\boldsymbol{c}}_{\boldsymbol{q}} = \boldsymbol{U}_{\boldsymbol{q}}^{-1} \boldsymbol{r}_{\boldsymbol{q}}\right) & (23)\\ &= \sum_{t=1}^{T} \left[\boldsymbol{g}_{\boldsymbol{q}_{t+L}}^{(t)}\right]^\top \cdot \boldsymbol{g}_{\boldsymbol{q}_{t+L}}^{(t)} & (24) \end{aligned}$$

where $\boldsymbol{g}_{\boldsymbol{q}}$ is a vector computed from $\boldsymbol{U}_{\boldsymbol{q}}$ and $\boldsymbol{r}_{\boldsymbol{q}}$ by forward substitution. Since $\boldsymbol{g}_{\boldsymbol{q}_{t+L}}^{(t)}$ depends only on the state sequence from time 1 to $t + L$, $\boldsymbol{r}_{\boldsymbol{q}}^\top \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}}$ can be also computed time-recursively.

As a result, “trajectory likelihood” (Eq. (17)) can be computed time-recursively as follows :

$$P(\boldsymbol{c} \mid \boldsymbol{q}, \lambda) = \prod_{t=1}^{T} \frac{1}{K_{\boldsymbol{q}_{t+L}}^{(t)}} \cdot P(\boldsymbol{o}_t \mid \boldsymbol{q}_t, \lambda) \,, \qquad (25)$$

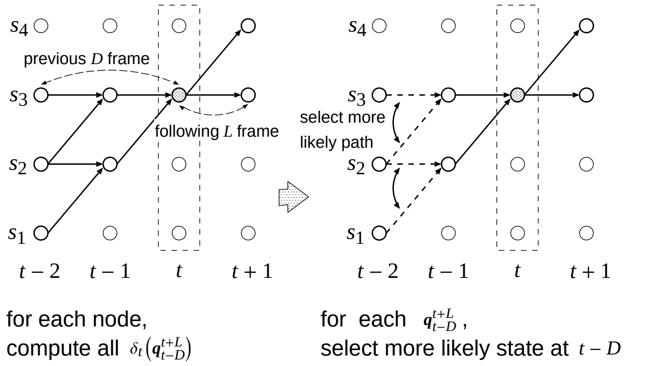
where

$$K_{\boldsymbol{q}_{t+L}}^{(t)} = \frac{\sqrt{(2\pi)^M} \left|\boldsymbol{U}_{\boldsymbol{q}_{t+L}}^{(t,t)}\right|^{-1}}{\sqrt{(2\pi)^{3M} \left|\boldsymbol{\Sigma}_{q_t}\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{q_t}^\top \boldsymbol{\Sigma}_{q_t}^{-1} \boldsymbol{\mu}_{q_t} - \left[\boldsymbol{g}_{\boldsymbol{q}_{t+L}}^{(t)}\right]^\top \boldsymbol{g}_{\boldsymbol{q}_{t+L}}^{(t)}\right)\right\} \,. \qquad (26)$$

3.2. Viterbi algorithm for trajectory-HMMs

In the trajectory-HMM framework, observations and the given state sequence depend on each other because the temporal covariance matrix $\boldsymbol{P}_{\boldsymbol{q}}$ is full. Accordingly, it is difficult to obtain the most likely state sequence for trajectory-HMM by dynamic programming. In this section, we describe an algorithm to obtain a “sub-optimal” state sequence based on dynamic programming. In this algorithm, the state q_{t-D} is determined at time t, according to “trajectory likelihood” for $\boldsymbol{q}_{t+L}$. It can be viewed as a Viterbi algorithm with D-frame delayed decision.

The proposed algorithm illustrated in Fig. 1 involves the followings:



s_4
s_3
s_2
s_1
previous D frame
following L frame
$t-2$
$t-1$
t
$t+1$
for each node,
compute all $\delta_t(\boldsymbol{q}_{t-D}^{t+L})$
select more likely path
for each $\boldsymbol{q}_{t-D}^{t+L}$,
select more likely state at $t-D$

Fig. 1. An overview for proposed Viterbi algorithm ($D = 2, L = 1$)

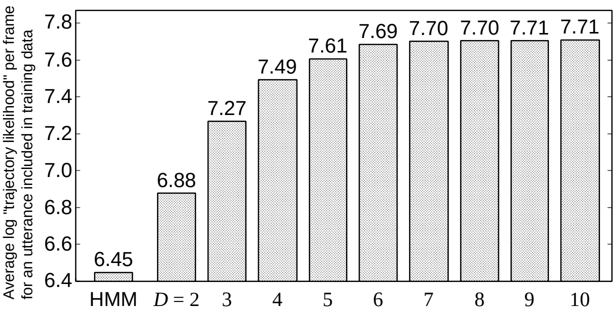


Fig. 2. Average log "trajectory likelihood" per frame for the standard HMM segmentation and state sequences obtained by the proposed algorithm.

0) Initialize : $t = 1$; $\forall \boldsymbol{q}_{1-D}^{1+L}$

$$\begin{aligned} \delta_1\left(\boldsymbol{q}_{1-D}^{1+L}\right) &= \pi_{q_1} b\left(\boldsymbol{q}_{1+L}\right) \\ \psi_1\left(\boldsymbol{q}_{1-D}^{1+L}\right) &= 0 \end{aligned}$$

1) Iterate : $t = 2, \ldots, T$; $\forall \boldsymbol{q}_{t-D}^{t+L}$

$$\begin{aligned} \delta_t\left(\boldsymbol{q}_{t-D}^{t+L}\right) &= \max_{q_{t-D-1}} \left[\delta_{t-1}\left(\boldsymbol{q}_{t-D-1}^{t+L-1}\right) a_{q_{t-1},\, q_t}\right] b(\boldsymbol{q}_{t+L}) \\ \psi_t\left(\boldsymbol{q}_{t-D}^{t+L}\right) &= \arg\max_{q_{t-D-1}} \left[\delta_{t-1}\left(\boldsymbol{q}_{t-D-1}^{t+L-1}\right) a_{q_{t-1},\, q_t}\right] \end{aligned}$$

2) Finish :

$$\begin{aligned} \hat{P} &= \max_{\boldsymbol{q}_{T-D}^{T+L}} \left[\delta_T\left(\boldsymbol{q}_{T-D}^{T+L}\right)\right] \\ \hat{\boldsymbol{q}}_{T-D}^{T+L} &= \arg\max_{\boldsymbol{q}_{T-D}^{T+L}} \left[\delta_T\left(\boldsymbol{q}_{T-D}^{T+L}\right)\right] \\ &= \{\hat{q}_{T-D}, \ldots, \hat{q}_{T+L}\} \end{aligned}$$

3) Back track : $t = T, \ldots, D + 2$

$$\begin{aligned} \hat{q}_{t-D-1} &= \psi_t\left(\hat{\boldsymbol{q}}_{t-D}^{t+L}\right) \\ \hat{\boldsymbol{q}}_{t-D-1}^{t+L-1} &= \{\hat{q}_{t-D-1}, \ldots, \hat{q}_{t+L-1}\} \end{aligned}$$

where π_{q_1} is an initial state probability of state q_1, a_{q_{t-1},q_t} is a state transition probability from q_{t-1} to q_t, $\boldsymbol{q}_{t-D}^{t+L} = \{q_{t-D}, \ldots, q_{t+L}\}$, and $b\left(\boldsymbol{q}_{t+L}\right)$ is given by

$$b(\boldsymbol{q}_{t+L}) = K_{\boldsymbol{q}_{t+L}}^{(t)} \cdot P\left(\boldsymbol{o}_t \mid q_t, \lambda\right) . \tag{27}$$

When D is equal to T, the most likely state sequence can be obtained by the above algorithm. Although the proposed algorithm with larger D can obtain more likely state sequence, it requires huge amount of computational cost. Hence, D have to be set to a proper value balancing its performance and computational complexity.

It is generally considered that the coarticulation affect neighboring frames within 100–200 ms. This indicates that optimal state sequence is probably obtained by the proposed algorithm when D is set to larger than 10 for 10-ms frame shift.

4. EXPERIMENT

4.1. Experimental conditions

We used phonetically balanced 503 sentences uttered by a male speaker MHT from the ATR Japanese speech database b-set. The 450 sentences were used for training monophone standard HMMs and trajectory-HMMs. The remaining 53 sentences were used for testing. Speech signals were sampled at 16 kHz and windowed by a 25.6-ms Blackman window with a 10-ms shift, and then mel-cepstral coefficients were obtained by a mel-cepstral analysis technique . Feature vector consists of 19 mel-cepstral coefficients including the zeroth coefficient, their delta and delta-delta coefficients. We used 3-state left-to-right with no skip HMM structure for modeling 36 Japanese phonemes. Each state output probability distribution is represented by a single Gaussian with diagonal covariance matrix.

4.2. Experimental results

First, we evaluated the performance of the proposed Viterbi algorithm in the likelihood of the obtained state sequence. Figure 2 shows the average log "trajectory likelihood" per frame of the obtained state sequence for an utterance included in the training data. In this figure, "HMM" means that the standard HMM segmentation was used for state sequence $\boldsymbol{q}$, and "$D = 2, \ldots, 10$" mean that the state sequences were obtained by the proposed algorithm. In this experiment, standard HMMs were used for acoustic models. It shows that the proposed algorithm could obtain more likely state sequence than the standard HMM segmentation. Furthermore, as D increased, "trajectory likelihood" was converged. When D was greater than 5, further likelihood improvement was not achieved. It suggests that the proposed algorithm with $D = 5$ can obtain approximately optimal state sequence.

Secondly, the proposed algorithm was applied to state-boundary optimization in Viterbi training procedure. After obtaining the state sequences by the proposed algorithm, model parameters were updated by formulas derived in [11] according to the obtained state sequences. This procedure was iterated several times. Table 1 shows the average log "trajectory likelihood" per frame for whole training data. In Table 1 and 2, "Iteration 0" means that initial models were used. In this experiment, the average log likelihood of the standard HMM segmentation for whole training data was 6.873. These results show that the proposed algorithm could obtain

Table 1. Average log “trajectory likelihood” per frame of the trained trajectory-HMMs for whole training data

	Iteration				
	0	1	2	3	4
$D = 2$	8.112	14.76	15.01	15.08	15.13
$D = 3$	8.501	14.93	15.17	15.28	15.32
$D = 4$	8.642	15.01	15.26	15.36	15.40
$D = 5$	8.699	15.04	15.30	15.39	15.42

Table 2. Phoneme error rate (%) for iteratively trained trajectory-HMMs (the number of *N*-best candidates was 20).

	Iteration					
	0	1	2	3	4	5
$D = 2$	20.1	18.8	18.5	18.8	18.8	18.4
$D = 3$	19.5	18.6	18.7	18.6	18.8	18.8
$D = 4$	19.7	**18.1**	18.2	18.7	18.8	18.5
$D = 5$	19.5	**18.1**	18.5	18.5	18.6	18.6

Table 3. Phoneme error rates (%) for the several number of *N*-best candidates.

	Number of candidates			
	20	30	100	200
$D = 2$	18.8	18.7	18.2	18.6
$D = 3$	18.6	18.4	17.9	17.8
$D = 4$	18.1	17.8	17.6	17.5
$D = 5$	18.1	17.8	17.5	**17.2**

more likely state sequence than the standard HMM segmentation. In addition, iterative training improved the model likelihood for training data.

Thirdly, trained trajectory-HMMs were evaluated in speaker-dependent continuous phoneme recognition experiments. First, an *N*-best list was generated for each test utterance by using the HTK Viterbi decoder with the phonotactic constraints of phoneme sequences in Japanese. For *N*-best list generation, initial models were used. Then, each candidate was re-segmented by the proposed algorithm and rescored with the “trajectory likelihood”. The proposed algorithm with the same D were used for both training and re-segmentation. The phoneme error rate for the baseline system based on the standard HMM was 20.1%.

Table 2 shows the phoneme error rates for iteratively trained trajectory-HMMs. Recognition results were shown in phoneme error rates. When trajectory-HMMs trained by single Viterbi training iteration with the proposed algorithm setting D to 4 or 5 were used, about 10% relative error reduction was achieved over the standard HMM. On the other hand, recognition performance did not improve by iterative training. This suggests that iterative training causes over-fitting to training data.

Table 3 shows the phoneme error rates for the several numbers of *N*-best candidates. In this experiment, trajectory-HMMs trained by Viterbi training with single iteration were used. As the number of *N*-best candidates increased, the recognition performance improved significantly. When 200-best candidates were used, about 14% relative error reduction over the standard HMM was achieved. It indicates that if we search wider space by implementing a Viterbi decoder based on the proposed algorithm, it is expected that larger improvement can be achieved.

5. CONCLUSION

This paper describes a Viterbi algorithm to obtain a sub-optimal state sequence for trajectory-HMM. It can be considered as a Viterbi algorithm with D-frame delayed decision. It was applied to state-boundary optimization for Viterbi training and *N*-best rescoring. In speaker-dependent continuous speech recognition experiment using rescoring scheme, about 14% relative error reduction over the standard HMM with the conventional Viterbi algorithm was achieved.

Future work includes the implementation of Viterbi decoder based on proposed algorithm and derivation of a Baum-Welch-type training algorithm.

6. ACKNOWLEDGEMENT

Authors would like to thank Prof. Takao Kobayashi, Dr. Takashi Masuko, and Mr. Yoshihiko Nankaku for helpful discussions.

7. REFERENCES

[1] M.J.F. Gales and S.J. Young, “Segmental hidden Markov models,” *Proc of Eurospeech’93*, pp.1579–1582, 1993.

[2] H. Gish and K. Ng, “Parametric trajectory models for speech recognition,” *Proc of ICSLP’96*, vol.I, pp.466–469, 1996.

[3] W.J. Holmes and M.J. Russell, “Speech recognition using a linear dynamic segmental HMM,” *Proc of ICASSP’95*, pp.1611–1614, 1995.

[4] M. Ostendorf, V. Digalakis, and O.A. Kimball, “From HMMs to segment models,” *IEEE Trans. on Speech & Audio Process.*, vol.4, no.5, pp.360–378, 1996.

[5] L. Deng, M. Aksmanovic, X. Sun, and J. Wu, “Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states,” *IEEE Trans. on Speech & Audio Process.*, vol.2, no.4, pp.507–520, 1994.

[6] Y. Gong and J.P. Haton, “Stochastic trajectory modeling for speech recognition,” *Proc. of ICASSP’94*, vol.I, pp57–60, 1994.

[7] K.K. Paliwal, “Use of temporal correlation between successive frames in hidden Markov model based Speech recognizer,” *Proc. of ICASSP’93*, pp.215-218, 1993.

[8] M. Ostendorf and S. Roukos, “A stochastic segment model for phoneme-based continuous speech recognition,” *IEEE Trans. Acoust., Speech & Signal Process.*, vol.37, no.12, pp.1857–1869, 1989.

[9] C.J. Wellekens, “Explicit correlation in hidden Markov model for speech recognition,” *Proc. of ICASSP’87*, pp.383–386, 1987.

[10] S. Furui, “Speaker independent isolated word recognition using dynamic features of speech spectrum,” *IEEE Trans. Acoust., Speech, & Signal Process.*, vol.34, pp.52–59, 1986.

[11] K. Tokuda, H. Zen, and T. Kitamura, “Trajectory modeling based on HMMs with the explicit relationship between static and dynamic features,” *Proc of Eurospeech2003*, pp.865–868, 2003.

[12] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, “Speech parameter generation algorithms for HMM-based speech synthesis,” *Proc. of ICASSP2000*, vol.3, pp.1315–1318, 2000.