Hidden Markov Model Estimation Based on Alpha-EM Algorithm:
Discrete and Continuous Alpha-HMMs

Yasuo Matsuyama

Abstract—**Fast estimation algorithms for Hidden Markov models (HMMs) for given data are presented. These algorithms start from the alpha-EM algorithm which includes the traditional log-EM as its proper subset. Since existing or traditional HMMs are the outcome of the log-EM, it had been expected that the alpha-HMM would exist. In this paper, it is shown that this foresight is true by using methods of the iteration index shift and likelihood ratio expansion. In each iteration, new update equations utilize one-step past terms which are computed and stored during the previous maximization step. Therefore, iteration speedup directly appears as that of CPU time. Since the new method is theoretically based on the alpha-EM, all of its properties are inherited. There are eight types of alpha-HMMs derived. They are discrete, continuous, semi-continuous and discrete-continuous alpha-HMMs, and both for single and multiple sequences. Using the properties of the alpha-EM algorithm, the speedup property is theoretically analyzed. Experimental results including real world data are given.**

I. Introduction

SOURCE modeling by a Markov process, i.e., the estimation of a Hidden Markov Model (HMM) has been utilized in various areas. This is because

(a) HMM shows a good performance to approximate the source structure, and

(b) There is an intelligible algorithm called Baum-Welch re-estimation algorithm [1], [2].

Since the appearance of this algorithm, many practical improvements were presented as was summarized in tutorials and monographs [3], [4], [5], [6]. But, recent growing demands on IT to be soft for humans require more efficient HMM estimation methods (hereafter, *re-estimation* will be simply expressed by *estimation*).

After the original presentation of the Baum-Welch algorithm, a general structure called EM-algorithm (Expectation- Maximization algorithm) was presented [7]. Since then, the Baum-Welch algorithm has been regarded as a special case of the EM-algorithm. This is because the derivation from the EM-algorithm is elegant and easier to understand what the missing information in source data is.

After the EM-algorithm was presented, it had been believed that this algorithm would be the supreme which unifies various statistical modeling. In this century, however, the alpha-EM algorithm [8] appeared as a general class which includes the traditional EM algorithm (more precisely, the log-EM algorithm) as its proper subset. The alpha-EM showed faster convergence than the log-EM because of the usage of the alpha-logarithm. Since then, it had been anticipated that a general HMM estimation algorithm would exist. But, this attempt was not successful until recently. This was because simply derived equalities contain non-causality and have exponential complexity. However, new methods described in this paper can cope with such difficulty to all of basic HMM estimation methods. The rest of the text is organized as follows.

In Section II, preliminaries to the alpha-logarithmic information measures are given. In Section III, the new HMM estimation algorithm called alpha-HMM is derived as an abstract form starting from the alpha-EM algorithm. In Section IV, a concrete version of the alpha-HMM which is software-implementable is derived. The idea is a probabilistic environment change and a series expansion. Section V gives experimental results. Section VI is provided for concluding remarks.

II. Complete and Incomplete Data for Hidden Markov Model Estimation

A. Source Data and Markov Model

In the modeling by HMM, a series of source data is given.

$$\mathbf{y} = \{y_t\}_{t=1}^{T} \tag{1}$$

Each y_t may be either a scalar or a vector. If there are multiple sequences provided, they are expressed as follows.

$$\mathbf{y}^{(n)} = \{y_t^{(n)}\}_{t=1}^{T}, \quad n = 1, \cdots, M \tag{2}$$

We will discuss the case of a single sequence first.

Given the observed data $\mathbf{y}$, the problem of HMM is to find the best model in the sense of the Maximum Likelihood Estimation (MLE):

$$P(\mathbf{s}, \mathbf{y} \mid \theta) = \pi_{s_0} \Pi_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(y_t) \tag{3}$$

Here,

$$\mathbf{s} = \left\{ s_0, \{s_t\}_{t=1}^{T} \right\} \tag{4}$$

means a state transition sequence. For both random variable and their values, lower case letters $\mathbf{y}$ and $\mathbf{s}$ will be used identically (usually, random variables are denoted by capital letters).

Probability measures are as follows.

The author is with the Department of Computer Science and Engineering, Waseda University, Tokyo, 169-8555, Japan. E-mail: yasuo2 @waseda.jp

This work was supported in part by the Grant-in-Aid for Scientific Research #22656088, Kayamori Foundation, and Ambient G-COE at Waseda University.

978-1-4244-9637-2/11/$26.00 ©2011 IEEE

(a) Initial state probabilities:

$$\mathbf{\Pi} = \{\pi_i\}, \quad \pi_i = P(s_0 = i), \quad i = 1, \cdots N. \tag{5}$$

(b) State transition probabilities:

$$\mathbf{A} = \{a_{ij}\}, \ a_{ij} = P(s_t = j \mid s_{t-1} = i) \tag{6}$$

Note that if there is no connection from state s_i to state s_j, then $a_{ij} = 0$. This reflects a prior topology for the state transition.

(c) Output probabilities:

$$\mathbf{B} = \{b_j(k)\} \tag{7}$$

Here, each element is as follows.

$$b_j(k) = P(y_t = k \mid s_t = j) \equiv b_{jk} \tag{8}$$

We denote the above set of probabilities by

$$\theta = (\mathbf{\Pi}, \mathbf{A}, \mathbf{B}). \tag{9}$$

Then, we have the following interpretation on the probabilistic data structure. The incomplete data is $\mathbf{y}$, the missing data to be estimated is $\mathbf{s}$, and the complete data is

$$\mathbf{x} \equiv \{\mathbf{s}, \mathbf{y}\} \tag{10}$$

which has the probability of (3) with parameters (9).

B. Alpha-EM Algorithm

The interpretation of HMM by incomplete data, missing data and complete data matches to the EM-algorithm. Since this paper's stance is to find new HMM estimation algorithms, we show a path starting from the alpha-EM algorithm.

Let $P_{\mathbf{y}|\psi}(\mathbf{y} \mid \psi)$ be the probability density or a probability mass for the observed data $\mathbf{y}$ parameterized by ψ. Let $\mathbf{x} \in \mathcal{X}$ be the complete or augmented data which is an ideal observation comprising the missing data. Then, the incomplete data probability density function (pdf) or probability mass function (pmf) is expressed as follows.

$$P_{\mathbf{y}|\psi}(\mathbf{y} \mid \psi) = \int_{\mathcal{X}(\mathbf{y})} P_{\mathbf{x}|\psi}(\mathbf{x} \mid \psi)\, d\mathbf{x} \tag{11}$$

Here, the region of integration is

$$\mathcal{X}(\mathbf{y}) = \{\mathbf{x} \mid \mathbf{Y}(\mathbf{x}) = \mathbf{y}\}. \tag{12}$$

The integration for the pdf becomes a summation for a pmf. Then, the conditional pdf or pmf is as follows.

$$P_{\mathbf{x}|\mathbf{y},\psi}(\mathbf{x} \mid \mathbf{y}, \psi) = \frac{P_{\mathbf{x}|\psi}(\mathbf{x} \mid \psi)}{P_{\mathbf{y}|\psi}(\mathbf{y} \mid \psi)} \tag{13}$$

In the alpha-EM algorithm, the alpha-logarithm is used.

$$L^{(\alpha)}(r) = \frac{2}{1+\alpha}\left(r^{\frac{1+\alpha}{2}} - 1\right) \tag{14}$$

Here, the case of $\alpha = -1$ in the limit is the logarithm, i.e.,

$$L^{(-1)}(r) = \log r. \tag{15}$$

When the alpha-EM algorithm is addressed, it is necessary to consider the incomplete data likelihood ratio in terms of the alpha-logarithm.

$$L_{\mathbf{y}}^{(\alpha)}(\psi \mid \varphi) \equiv L^{(\alpha)}\left(\frac{P_{\mathbf{y}|\psi}(\mathbf{y} \mid \psi)}{P_{\mathbf{y}|\varphi}(\mathbf{y} \mid \varphi)}\right) \tag{16}$$

Here, φ and ψ denote old and new models for (9) in iterative maximization steps. Then, the basic equation of the alpha-EM algorithm is obtained [8].

$$L_{\mathbf{y}}^{(\alpha)}(\psi \mid \varphi) = Q_{\mathbf{x}|\mathbf{y},\varphi}^{(\alpha)}(\psi \mid \varphi) + \frac{1-\alpha}{2}\left\{\frac{P_{\mathbf{y}|\psi}(\mathbf{y} \mid \psi)}{P_{\mathbf{y}|\varphi}(\mathbf{y} \mid \varphi)}\right\}^{\frac{1+\alpha}{2}} D_{\mathbf{x}|\mathbf{y}}^{(\alpha)}(\varphi \| \psi) \tag{17}$$

Here, $D^{(\alpha)}$ is the alpha-divergence between two conditional probabilities $P_{\mathbf{x}|\mathbf{y},\varphi}(\mathbf{x} \mid \mathbf{y}, \varphi)$ and $P_{\mathbf{x}|\mathbf{y},\psi}(\mathbf{x} \mid \mathbf{y}, \psi)$ which is always non-negative. The important term in (17) is the Q-function.

$$Q_{\mathbf{x}|\mathbf{y},\varphi}^{(\alpha)}(\psi \mid \varphi) \equiv E_{P_{\mathbf{x}|\mathbf{y},\varphi}}[L_{\mathbf{x}}^{(\alpha)}(\psi \mid \varphi)] \tag{18}$$

Here, $L_{\mathbf{x}}^{(\alpha)}(\psi \mid \varphi)$ is the alpha-log likelihood ratio for the complete data on $P_{\mathbf{x}|\psi}(\mathbf{x} \mid \psi)$ and $P_{\mathbf{x}|\varphi}(\mathbf{x} \mid \varphi)$.

Due to (17), if the Q-function is positive, so is the incomplete data alpha-log likelihood ratio which is the left hand side of (17) for α<1. Therefore, the alpha-EM algorithm and its variant, the alpha-GEM algorithm, are described as follows.

[Alpha-EM Algorithm]
Initialization: Choose initial values for (9) and use it as φ.
E-Step: Compute Equation (18).
M-step: Compute the update parameter by

$$\psi^* = \arg\max_{\psi} \ Q_{\mathbf{x}|\mathbf{y},\varphi}^{(\alpha)}(\psi \mid \varphi). \tag{19}$$

U-step: Replace φ by ψ^* and check to see the convergence. If the convergence is not achieved, the iteration is repeated by going back to the E-step.

[Alpha-GEM Algorithm]
This is an algorithm where the above M-step is replaced by computing ψ^+.

$$Q_{\mathbf{x}|\mathbf{y},\varphi}^{(\alpha)}(\psi^+ \mid \varphi) \geq 0 \tag{20}$$

It is worth noting in advance that approximated versions of the alpha-HMM algorithms can be regarded as alpha-GEM algorithms. By virtue of the alpha-EM algorithm which possess the relationship (17), we have the following proposition which assures the basic property of the alpha-HMM estimation algorithm.

[Proposition 1]

Let the complete data be $\mathbf{x}=(\mathbf{s}, \mathbf{y})$ where $\mathbf{s}$ is the missing data and $\mathbf{y}$ is the incomplete data, then the expression (18) is equivalent to

$$Q^{(\alpha)}_{\mathbf{s}|\mathbf{y},\varphi}(\psi \mid \varphi) = E_{p_{\mathbf{s}|\mathbf{y},\varphi}}[L^{(\alpha)}_{\mathbf{s},\mathbf{y}}(\psi \mid \varphi)]. \tag{21}$$

If (21) is non-negative, the following inequality hold.

$$P(\mathbf{y} \mid \psi) \geq P(\mathbf{y} \mid \varphi) \tag{22}$$

Note that this is an abstract version of the alpha-HMM.

III. Single Sequence Alpha-HMMs

A. Non-Causal Update Equations

In the case of discrete alphabet series $\mathbf{y}$, the Q-function to be maximized is expressed as follows.

$$Q^{(\alpha)}(\theta_{l+1} \mid \theta_l) = \frac{2}{1+\alpha}\left[\sum_{\mathbf{s}\in\mathbf{S}} P(\mathbf{s} \mid \mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s},\mathbf{y} \mid \theta_{l+1})}{P(\mathbf{s},\mathbf{y} \mid \theta_l)}\right\}^{\frac{1+\alpha}{2}} - 1\right] \tag{23}$$

Here, l is the index for the iteration.

First, we explain the update equation for the state transition probability a_{ij}. Since a_{ij} needs to be a probability mass after the update, it is necessary to use a Lagrange multiplier. Therefore,

$$\frac{\partial}{\partial a_{ij|\theta_{l+1}}}\left\{Q^{(\alpha)}(\theta_{l+1} \mid \theta_l) + \lambda\left(\sum_j a_{ij|\theta_{l+1}} - 1\right)\right\} = 0 \tag{24}$$

is the differentiation for the maximization. This gives

$$a_{ij|\theta_{l+1}} = \frac{\sum_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s},\mathbf{y} \mid \theta_{l+1})}{P(\mathbf{s},\mathbf{y} \mid \theta_l)}\right\}^{\frac{1+\alpha}{2}} N_{ij}(\mathbf{s})}{\sum_j \sum_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s},\mathbf{y} \mid \theta_{l+1})}{P(\mathbf{s},\mathbf{y} \mid \theta_l)}\right\}^{\frac{1+\alpha}{2}} N_{ij}(\mathbf{s})} \tag{25}$$

Here, $N_{ij}(\mathbf{s})$ is the number and positions of the state transition from i to j. As the next step towards a software-implementable algorithm, we need to resolve the following problems.

(a) Non-causality exists: The right hand side contains θ_{l+1}.
(b) The computation of the right hand side requires $O(N^T)$ operations.

The above problems will be settled in the next subsection.

Before moving to the next subsection, we list two more update equations for the output probability and the initial state probability.

$$b_{jk|\theta_{l+1}} = \frac{\sum_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s},\mathbf{y} \mid \theta_{l+1})}{P(\mathbf{s},\mathbf{y} \mid \theta_l)}\right\}^{\frac{1+\alpha}{2}} N_{jk}(\mathbf{s})}{\sum_k \sum_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s},\mathbf{y} \mid \theta_{l+1})}{P(\mathbf{s},\mathbf{y} \mid \theta_l)}\right\}^{\frac{1+\alpha}{2}} N_{jk}(\mathbf{s})} \tag{26}$$

where $N_{jk}(\mathbf{s})$ is the number of occurrence of the output $y_t=k$ at the state $s_t=j$. For the update of the initial state probability, we have

$$\pi_{i|\theta_{l+1}} = \frac{\sum_{s_0=i} P(\mathbf{s} \mid \mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s},\mathbf{y} \mid \theta_{l+1})}{P(\mathbf{s},\mathbf{y} \mid \theta_l)}\right\}^{\frac{1+\alpha}{2}}}{\sum_i \sum_{s_0=i} P(\mathbf{s} \mid \mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s},\mathbf{y} \mid \theta_{l+1})}{P(\mathbf{s},\mathbf{y} \mid \theta_l)}\right\}^{\frac{1+\alpha}{2}}}. \tag{27}$$

B. Causal Approximation and Series Expansion: Discrete Output Case

The core part of the update equations (25)–(27) can be transformed as follows.

[Causal approximation]

$$\begin{aligned} &P(\mathbf{s} \mid \mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s},\mathbf{y} \mid \theta_{l+1})}{P(\mathbf{s},\mathbf{y} \mid \theta_l)}\right\}^{\frac{1+\alpha}{2}} \\ &= P(\mathbf{s} \mid \mathbf{y}, \theta_{l+1})\left\{\frac{P(\mathbf{s},\mathbf{y} \mid \theta_{l+1})}{P(\mathbf{s},\mathbf{y} \mid \theta_l)}\right\}^{-\frac{1-\alpha}{2}} \times \left\{\frac{P(\mathbf{y} \mid \theta_{l+1})}{P(\mathbf{y} \mid \theta_l)}\right\} \\ &\approx P(\mathbf{s} \mid \mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s},\mathbf{y} \mid \theta_l)}{P(\mathbf{s},\mathbf{y} \mid \theta_{l-1})}\right\}^{-\frac{1-\alpha_{\text{causal}}}{2}} \times \left\{\frac{P(\mathbf{y} \mid \theta_l)}{P(\mathbf{y} \mid \theta_{l-1})}\right\} \end{aligned} \tag{28}$$

Here, the last term is the causal approximation by the iteration index shift. Therefore, one obtains the correspondence of

$$\alpha_{\text{causal}} = \alpha + 2 \equiv \beta \tag{29}$$

around the region of $P(\mathbf{y} \mid \theta_l) = P(\mathbf{y} \mid \theta_{l-1}) + o(1)$. We use α_{causal} so that resulting algorithms reflect the expectation by the current probabilistic environment appearing in the last line of Equation (28). In the update equations appearing in this and later sections, a simpler notation β will be used for α_{causal} so that spaces for equations can be saved.

Computation of (28) is now possible, however, another approximation is necessary in view of computational complexity. For this purpose, we use a series expansion.

[Series expansion]

$$\begin{aligned} &P(\mathbf{s} \mid \mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s},\mathbf{y} \mid \theta_l)}{P(\mathbf{s},\mathbf{y} \mid \theta_{l-1})}\right\}^{-\frac{1-\alpha}{2}} \\ &= \frac{1+\alpha}{2} P(\mathbf{s} \mid \mathbf{y}, \theta_l)\frac{P(\mathbf{y} \mid \theta_l)}{P(\mathbf{y} \mid \theta_{l+1})} + \frac{1-\alpha}{2} P(\mathbf{s} \mid \mathbf{y}, \theta_{l-1}) + o(1) \end{aligned} \tag{30}$$

Then, the application of the causal approximation (28) and the series expansion of (30) give the following update equation for the transition probability.

[Transition pr. for single sequence discrete alpha-HMM]

$$a_{ij|\theta_{l+1}} = \frac{(1+\beta)P(\mathbf{y}\mid\theta_l)N_{a_{ij}|\theta_l} + (1-\beta)P(\mathbf{y}\mid\theta_{l-1})N_{a_{ij}|\theta_{l-1}}}{(1+\beta)P(\mathbf{y}\mid\theta_l)\sum_j N_{a_{ij}|\theta_l} + (1-\beta)P(\mathbf{y}\mid\theta_{l-1})\sum_j N_{a_{ij}|\theta_{l-1}}}$$
$$= \frac{(1+\beta)\sum_{t=1}^{T} P(s_{t-1}=i, s_t=j, \mathbf{y}\mid\theta_l) + (1-\beta)\sum_{t=1}^{T} P(s_{t-1}=i, s_t=j, \mathbf{y}\mid\theta_{l-1})}{(1+\beta)\sum_{t=1}^{T} P(s_{t-1}=i, \mathbf{y}\mid\theta_l) + (1-\beta)\sum_{t=1}^{T} P(s_{t-1}=i, \mathbf{y}\mid\theta_{l-1})} \quad (31)$$

It is important here to understand the following.

[Property 1]

The case of $\beta=\alpha_{causal}=1$ is reduced to the traditional log-HMM method.

[Property 2]

The numerator of (31) is a weighted summation of the present and the past update terms. So is the form of the denominator of (31).

[Property 3]

Both the second and third lines of (31) match to the traditional forward-backward method [1], [3] which save the complexity on computing these probabilities.

[Property 4]

The only additional necessity for the alpha-HMM is to memorize the update term at θ_{l-1}. This means that save of iterations directly appears as that of the CPU time. Experiments support this anticipation.

The rest two update equations are as follows.

[Output pr. for single sequence discrete alpha-HMM]

$$b_{jk|\theta_{l+1}} = \frac{(1+\beta)P(\mathbf{y}\mid\theta_l)N_{b_{jk}|\theta_l} + (1-\beta)P(\mathbf{y}\mid\theta_{l-1})N_{b_{jk}|\theta_{l-1}}}{(1+\beta)P(\mathbf{y}\mid\theta_l)\sum_k N_{b_{jk}|\theta_l} + (1-\beta)P(\mathbf{y}\mid\theta_{l-1})\sum_k N_{b_{jk}|\theta_{l-1}}}$$
$$= \frac{(1+\beta)\sum_{t:y_t=k} P(s_{t-1}=i, s_t=j, \mathbf{y}\mid\theta_l) + (1-\beta)\sum_{t:y_t=k} P(s_{t-1}=i, s_t=j, \mathbf{y}\mid\theta_{l-1})}{(1+\beta)\sum_{t:y_t=k} P(s_t=j, \mathbf{y}\mid\theta_l) + (1-\beta)\sum_{t:y_t=k} P(s_t=j, \mathbf{y}\mid\theta_{l-1})} \quad (32)$$

[Initial state pr. for single sequence discrete alpha-HMM]

$$\pi_{i|\theta_{l+1}} = \frac{(1+\beta)P(s_0=i, \mathbf{y}\mid\theta_l) + (1-\beta)P(s_0=i, \mathbf{y}\mid\theta_{l-1})}{(1+\beta)P(\mathbf{y}\mid\theta_l) + (1-\beta)P(\mathbf{y}\mid\theta_{l-1})} \quad (33)$$

C. Continuous Output Series

If the output series $\mathbf{y} = \{y_t\}_{t=1}^T$ appear as continuous multivariate observations, a similar but a different set of update equations from (31) – (33) is obtained. In this case, y_t is a vector in an appropriate dimensional Euclidian space although it is not denoted by a bold font.

The MLE problem for such a continuous alphabet case is to maximize the following likelihood.

$$P(\mathbf{s}, \mathbf{c}, \mathbf{y}\mid\theta) = \pi_{s_0}\prod_{t=1}^{T} a_{s_{t-1}s_t} c_{s_t k_t} b_{s_t k_t}(y_t). \quad (34)$$

Here, $c_{s_t k_t}$ specifies the probability of the transition to the k_t-th branch at the state s_t. $b_{s_t k_t}(y_t)$ is a probability density function for y_t. We assume it to be a Gaussian density.

$$b_{jk}(y_t) = \mathcal{N}(y_t; \mu_{jk}, \Sigma_{jk}) \quad (35)$$

Here, μ_{jk} is the mean vector and Σ_{jk} is the covariance matrix which should not be mixed up with the summation symbol. Then, the output probability density function at state j is

$$b_j(y_t) = \sum_{k=1}^{M} c_{jk}\mathcal{N}(y_t; \mu_{jk}, \Sigma_{jk}) \quad (36)$$

so that $b_j(y_t)$ is a pdf. Practically, such a Gaussian mixture model is the most general case even for the log-HMM although there is a slightly wider class which contains this model [9], [10]. Fig. 1 is a graphical expression for the discrete alphabet case (top) and the Gaussian mixture case (bottom). By reviewing this figure, one finds that c_{jk} of (34) corresponds to b_{jk} of (3).

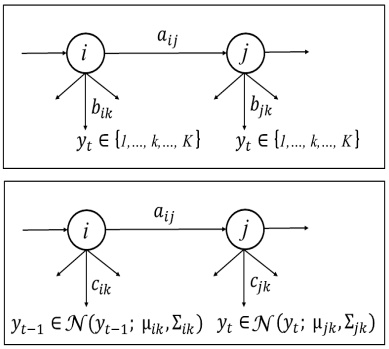

Fig.1. Graphical model comparison of discrete and continuous alphabet cases.

For the case of the mixture probability, missing data are **s** and **c**. Therefore, the Q-function is as follows.

$$Q^{(\alpha)}(\theta_{l+1}\mid\theta_l) = \frac{2}{1+\alpha}\left[\sum_{\mathbf{s}}\sum_{\mathbf{c}} P(\mathbf{s}, \mathbf{c}\mid\mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s}, \mathbf{c}, \mathbf{y}\mid\theta_{l+1})}{P(\mathbf{s}, \mathbf{c}, \mathbf{y}\mid\theta_l)}\right\}^{\frac{1+\alpha}{2}} - 1\right] \quad (37)$$

As can be understood from (34) and (37), update equations for the initial probability and the state transition probability are the same as (31) and (33), respectively. The update equation for c_{jk} is obtained in a similar way to that of a_{ij} by using a Lagrange multiplier.

$$c_{jk|\theta_{l+1}} = \frac{\sum_{\mathbf{s}}\sum_{\mathbf{c}} P(\mathbf{s}, \mathbf{c}\mid\mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s}, \mathbf{c}, \mathbf{y}\mid\theta_{l+1})}{P(\mathbf{s}, \mathbf{c}, \mathbf{y}\mid\theta_l)}\right\}^{\frac{1+\alpha}{2}} N_{jk}(\mathbf{c})}{\sum_{k}\sum_{\mathbf{s}}\sum_{\mathbf{c}} P(\mathbf{s}, \mathbf{c}\mid\mathbf{y}, \theta_l)\left\{\frac{P(\mathbf{s}, \mathbf{c}, \mathbf{y}\mid\theta_{l+1})}{P(\mathbf{s}, \mathbf{c}, \mathbf{y}\mid\theta_l)}\right\}^{\frac{1+\alpha}{2}} N_{jk}(\mathbf{c})} \quad (38)$$

Then, the right hand side is made causal and computable in the same way to the case of the state transition probability.

$$c_{jk|\theta_{l+1}} = \frac{(1+\beta)P(\mathbf{y}\mid\theta_l)N_{c_{jk}|\theta_l} + (1-\beta)P(\mathbf{y}\mid\theta_{l-1})N_{c_{jk}|\theta_{l-1}}}{(1+\beta)P(\mathbf{y}\mid\theta_l)\sum_k N_{c_{jk}|\theta_l} + (1-\beta)P(\mathbf{y}\mid\theta_{l-1})\sum_k N_{c_{jk}|\theta_{l-1}}}$$
$$= \frac{(1+\beta)\sum_{t=1}^{T} P(s_t=j,c_t=k,\mathbf{y}\mid\theta_l) + (1-\beta)\sum_{t=1}^{T} P(s_t=j,c_t=k,\mathbf{y}\mid\theta_{l-1})}{(1+\beta)\sum_{t=1}^{T} P(s_t=j,\mathbf{y}\mid\theta_l) + (1-\beta)\sum_{t=1}^{T} P(s_t=j,\mathbf{y}\mid\theta_{l-1})} \tag{39}$$

The next update equation is on the mean vector μ_{jk}. A direct differentiation of (37) with respect to μ_{jk} gives the following non-causal equality.

$$\mu_{jk|\theta_{l+1}} = \frac{\sum_{\mathbf{s}}\sum_{\mathbf{c}} P(\mathbf{s},\mathbf{c}\mid\mathbf{y},\theta_l)\left\{\frac{P(\mathbf{s},\mathbf{c},\mathbf{y}\mid\theta_{l+1})}{P(\mathbf{s},\mathbf{c},\mathbf{y}\mid\theta_l)}\right\}^{\frac{1+\alpha}{2}}\delta(s_t=j,k_t=k)y_t}{\sum_{\mathbf{s}}\sum_{\mathbf{c}} P(\mathbf{s},\mathbf{c}\mid\mathbf{y},\theta_l)\left\{\frac{P(\mathbf{s},\mathbf{c},\mathbf{y}\mid\theta_{l+1})}{P(\mathbf{s},\mathbf{c},\mathbf{y}\mid\theta_l)}\right\}^{\frac{1+\alpha}{2}}\delta(s_t=j,k_t=k)} \tag{40}$$

Then, by shifting the iteration index, expanding to series, and changing the summation, one obtains the following update equation.

$$\mu_{jk|\theta_{l+1}} = \frac{\left\{(1+\beta)\sum_{t=1}^{T} P(s_t=j,k_t=k,\mathbf{y}\mid\theta_l) + (1-\beta)\sum_{t=1}^{T} P(s_t=j,k_t=k,\mathbf{y}\mid\theta_{l-1})\right\}y_t}{(1+\beta)\sum_{t=1}^{T} P(s_t=j,k_t=k,\mathbf{y}\mid\theta_l) + (1-\beta)\sum_{t=1}^{T} P(s_t=j,k_t=k,\mathbf{y}\mid\theta_{l-1})} \tag{41}$$

Note that (41) is an expression where the past information of θ_{l-1} can be fully utilized.

On the update of the covariance matrix, a matrix differentiation [11] is necessary. One obtains the following non-causal equation by differentiating the Q-function (37) with respect to the inverse of the covariance matrix Σ_{jk}^{-1}.

$$\Sigma_{jk|\theta_{l+1}} = \frac{\sum_{\mathbf{s}}\sum_{\mathbf{c}} P(\mathbf{s},\mathbf{c}\mid\mathbf{y},\theta_l)\left\{\frac{P(\mathbf{s},\mathbf{c},\mathbf{y}\mid\theta_{l+1})}{P(\mathbf{s},\mathbf{c},\mathbf{y}\mid\theta_l)}\right\}^{\frac{1+\alpha}{2}}\delta(s_t=j,k_t=k)(y_t-\mu_{\theta_{l+1}})(y_t-\mu_{\theta_{l+1}})^T}{\sum_{\mathbf{s}}\sum_{\mathbf{c}} P(\mathbf{s},\mathbf{c}\mid\mathbf{y},\theta_l)\left\{\frac{P(\mathbf{s},\mathbf{c},\mathbf{y}\mid\theta_{l+1})}{P(\mathbf{s},\mathbf{c},\mathbf{y}\mid\theta_l)}\right\}^{\frac{1+\alpha}{2}}\delta(s_t=j,k_t=k)} \tag{42}$$

Then, by shifting the iteration index, expanding to series, and changing the summation, one obtains the following update equation.

$$\Sigma_{jk|\theta_{l+1}} = \frac{(1+\beta)\times num_{\theta_l} + (1-\beta)\times num_{\theta_{l-1}}}{(1+\beta)\times denom_{\theta_l} + (1-\beta)\times denom_{\theta_{l-1}}} \tag{43}$$

Here, each term is as follows.

$$denom_{\theta_l} = \sum_{t=1}^{T} P(s_t=j,k_t=k,\mathbf{y}\mid\theta_l) \tag{44}$$

and

$$num_{\theta_l} = \left\{\sum_{t=1}^{T} P(s_t=j,k_t=k,\mathbf{y}\mid\theta_l)\right\}(y_t-\mu_{jk|\theta_l})(y_t-\mu_{jk|\theta_l})^T \tag{45}$$

Note that the update of the covariance matrix (43) has the form where the past information can be utilized effectively. Before moving to the next subsection, we summarize the update method for the Gaussian mixture alpha-HMM, i.e., single sequence continuous alpha-HMM.

[Initial state pr. for single sequence continuous alpha-HMM]
The update equation is (33).
[State trans. pr. for single sequence continuous alpha-HMM]
The update equation is (31).
[Branch pr. for single sequence continuous alpha-HMM]
The update equation is (39).
[Mean vector for single sequence continuous alpha-HMM]
The update equation is (41).
[Covariance matrix for single sequence continuous alpha-HMM]
The update equation is (43), whose components are (44) and (45).

It is important to emphasize here again that all information on θ_{l-1} is just stored in a memory. The computation on the terms indexed by θ_l is equivalent to the log-HMM.

D. Semi-Continuous Alpha-HMM

By reviewing the graphical structure of the Gaussian mixture alpha-HMM in Fig. 1 (log-HMM too), we realize the following.

(a) In the Gaussian mixture HMM, each Gaussian pdf depends upon the arrival state j. Learning of all $N\times K$ Gaussian densities requires a variety of long training sequences.

(b) Assign the role of b_{jk} for the discrete case to c_{jk} of the continuous mode. Also, consider the case that mean vectors and covariance matrices do not depend on the transition state j. Then, the structure becomes a straightforward extension of the discrete case of Fig. 1 (top). This is called semi-continuous HMM [12]. Another interpretation for this structure is MLE-VQ HMM (Maximum Likelihood Vector Quantization HMM).

The model for the semi-continuous alpha-HMM becomes as follows by changing (34) to

$$P(\mathbf{s},\mathbf{c},\mathbf{y}\mid\theta) = \pi_{s_0}\prod_{t=1}^{T} a_{s_{t-1}s_t}c_{s_tk_t}\mathcal{N}(y_t;\mu_{k_t},\Sigma_{k_t}). \tag{46}$$

Therefore, update equations for the semi-continuous alpha-HMM becomes as follows.

[Initial state pr. for single seq. semi-conti. alpha-HMM]
The update equation is (33).
[State trans. pr. for single seq. semi-conti. alpha-HMM]
The update equation is (31).
[Branch pr. for single seq. semi-conti. alpha-HMM]
The update equation is (39).
[Mean vector for single seq. semi-cont. alpha-HMM]

In Equation (41), change $\mu_{jk|\theta_{l+1}}$ to $\mu_{j|\theta_{l+1}}$. In addition to this, remove the term $k_t=k$ in the right hand side.

[Covariance matrix for single seq. semi-conti. alpha-HMM]

In Equations (43) – (45), change $\Sigma_{jk|\theta_{l+1}}$ to $\Sigma_{j|\theta_{l+1}}$. In addition to this, remove $k_t=k$ in the right hand side.

IV. Multiple Sequence Alpha-HMMs

If a pre-designed topology of HMM is an ergodic one, a single long training sequence **y** is enough. If a selected topology has an absorbing state, it is better to use multiple numbers of training sequences. Therefore, we derive update equations for multiple sequence alpha-HMM estimations.

A. Discrete Symbol Case

Let

$$\mathbf{S}=(\mathbf{s}^{(1)},\cdots,\mathbf{s}^{(n)},\cdots,\mathbf{s}^{(M)}) \tag{47}$$

be the collection of M state transition sequences. Then, the Q-function for the multiple sequence is as follows.

$$\begin{aligned}Q^{(\alpha)}(\theta_{l+1}\mid\theta_l)&=\frac{2}{1+\alpha}\left[\sum\nolimits_{\mathbf{S}}P(\mathbf{S}\mid\mathbf{y},\theta_l)\left\{\frac{P(\mathbf{S},\mathbf{y}\mid\theta_{l+1})}{P(\mathbf{S},\mathbf{y}\mid\theta_l)}\right\}^{\frac{1+\alpha}{2}}-1\right]\\&=\frac{2}{1+\alpha}\left[\sum_{\mathbf{S}}\prod_{n=1}^{M}P(\mathbf{s}^{(n)}\mid\mathbf{y}^{(n)},\theta_l)\left\{\frac{P(\mathbf{s}^{(n)},\mathbf{y}^{(n)}\mid\theta_{l+1})}{P(\mathbf{s}^{(n)},\mathbf{y}^{(n)}\mid\theta_l)}\right\}^{\frac{1+\alpha}{2}}-1\right]\\&=\frac{2}{1+\alpha}\left[\prod_{n=1}^{M}\sum_{\mathbf{s}^{(n)}}P(\mathbf{s}^{(n)}\mid\mathbf{y}^{(n)},\theta_l)\left\{\frac{P(\mathbf{s}^{(n)},\mathbf{y}^{(n)}\mid\theta_{l+1})}{P(\mathbf{s}^{(n)},\mathbf{y}^{(n)}\mid\theta_l)}\right\}^{\frac{1+\alpha}{2}}-1\right]\end{aligned} \tag{48}$$

Here, P is a probability for a Markov process.

$$P(\mathbf{s}^{(n)},\mathbf{y}^{(n)}\mid\theta)=\pi_{s_0^{(n)}}\prod_{t=1}^{T}a_{s_{t-1}^{(n)}s_t^{(n)}}b_{s_t^{(n)}}(y_t^{(n)}) \tag{49}$$

Note that forms of the initial state probability, state transition probability and output probability are independent of the sequence index n.

The derivation of the update equation for the initial state starts from the differentiation of the Q-function (48) by $\pi_{i|\theta_{l+1}}$. The difference from the single sequence of subsection III.A is that derivatives appear n times which can be understood from (49). Then, the non-causal equality is obtained as follows.

$$\pi_{i|\theta_{l+1}}=\frac{\sum_{n=1}^{M}g_{\pi_i}(n)}{\sum_{n=1}^{M}\sum_i g_{\pi_i}(n)} \tag{50}$$

Here,

$$g_{\pi_i}(n)=\sum_{s_0^{(n)}=i}P(\mathbf{s}^{(n)}\mid\mathbf{y}^{(n)},\theta_l)\left\{\frac{P(\mathbf{s}^{(n)},\mathbf{y}^{(n)}\mid\theta_{l+1})}{P(\mathbf{s}^{(n)},\mathbf{y}^{(n)}\mid\theta_l)}\right\}^{\frac{1+\alpha}{2}}\Big/f(n) \tag{51}$$

and

$$f(n)=\sum_{\mathbf{s}^{(n)}}P(\mathbf{s}^{(n)}\mid\mathbf{y}^{(n)},\theta_l)\left\{\frac{P(\mathbf{s}^{(n)},\mathbf{y}^{(n)}\mid\theta_{l+1})}{P(\mathbf{s}^{(n)},\mathbf{y}^{(n)}\mid\theta_l)}\right\}^{\frac{1+\alpha}{2}}. \tag{52}$$

Then, the iteration index shift for the causality and the series expansion gives the following update equation.

$$\pi_{i|\theta_{l+1}}=\frac{1}{M}\sum_{n=1}^{M}\frac{(1+\beta)P(s_0^{(n)}=i,\mathbf{y}^{(n)}\mid\theta_l)+(1-\beta)P(s_0^{(n)}=i,\mathbf{y}^{(n)}\mid\theta_{l-1})}{(1+\beta)P(\mathbf{y}^{(n)}\mid\theta_l)+(1-\beta)P(\mathbf{y}^{(n)}\mid\theta_{l-1})} \tag{53}$$

Similarly, one obtains update equations for the state transition and the output.

$$a_{ij|\theta_{l+1}}=\frac{\sum_{n=1}^{M}\frac{(1+\beta)A_{\theta_l}^{(n)}+(1-\beta)A_{\theta_{l-1}}^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)}\mid\theta_l)+(1-\beta)P(\mathbf{y}^{(n)}\mid\theta_{l-1})}}{\sum_{n=1}^{M}\frac{(1+\beta)B_{\theta_l}^{(n)}+(1-\beta)B_{\theta_{l-1}}^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)}\mid\theta_l)+(1-\beta)P(\mathbf{y}^{(n)}\mid\theta_{l-1})}} \tag{54}$$

where

$$A_{\theta_l}^{(n)}=\sum_{t=1}^{T}P(s_{t-1}^{(n)}=i,s_t^{(n)}=j,\mathbf{y}^{(n)}\mid\theta_l) \tag{55}$$

and

$$B_{\theta_l}^{(n)}=\sum_{t=1}^{T}P(s_{t-1}^{(n)}=i,\mathbf{y}^{(n)}\mid\theta_l). \tag{56}$$

For the output probability, one obtains below.

$$b_{jk|\theta_{l+1}}=\frac{\sum_{n=1}^{M}\frac{(1+\beta)C_{\theta_l}^{(n)}+(1-\beta)C_{\theta_{l-1}}^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)}\mid\theta_l)+(1-\beta)P(\mathbf{y}^{(n)}\mid\theta_{l-1})}}{\sum_{n=1}^{M}\frac{(1+\beta)D_{\theta_l}^{(n)}+(1-\beta)D_{\theta_{l-1}}^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)}\mid\theta_l)+(1-\beta)P(\mathbf{y}^{(n)}\mid\theta_{l-1})}} \tag{57}$$

where

$$C_{\theta_l}^{(n)}=\sum_{t:y_t=k}P(s_t^{(n)}=j,\mathbf{y}^{(n)}\mid\theta_l) \tag{58}$$

and

$$D_{\theta_l}^{(n)}=\sum_{t=1}^{T}P(s_t^{(n)}=j,\mathbf{y}^{(n)}\mid\theta_l). \tag{59}$$

B. Continuous Symbol Case

Update equations for the continuous symbol alpha-HMM can be obtained too. The update equations for the initial state probability and state transition probability are the same as (53) and (54), respectively. But, a set of update equations for the output is different form the discrete symbol case. We need update equations for the branch probability, the mean vector and the covariance matrix.

For the branch probability, a method similar to the cases of the initial state probability and the state transition probability is possible by using Lagrange multipliers. This gives the update equation as follows.

$$c_{jk|\theta_{l+1}} = \frac{\sum_{n=1}^{M} \frac{(1+\beta)C_{\theta_l}^{(n)} + (1-\beta)C_{\theta_{l-1}}^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)} \mid \theta_l) + (1-\beta)P(\mathbf{y}^{(n)} \mid \theta_{l-1})}}{\sum_{n=1}^{M} \frac{(1+\beta)D_{\theta_l}^{(n)} + (1-\beta)D_{\theta_{l-1}}^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)} \mid \theta_l) + (1-\beta)P(\mathbf{y}^{(n)} \mid \theta_{l-1})}} \quad (60)$$

where

$$C_{\theta_l}^{(n)} = \sum_{t:c_t^{(n)}=k} P(s_t^{(n)} = j, \mathbf{y}^{(n)} \mid \theta_l) \quad (61)$$

and

$$D_{\theta_l}^{(n)} = \sum_{t=1}^{T} P(s_t^{(n)} = j, \mathbf{y}^{(n)} \mid \theta_l). \quad (62)$$

For $\mu_{jk|\theta_{l+1}}$, a direct vector differentiation on the Q-function (48) is applied. Then, the update equation is describes as follows.

$$\mu_{jk|\theta_{l+1}} = \frac{\sum_{n=1}^{M} \frac{\left\{(1+\beta)F_{\theta_l}^{(n)} + (1-\beta)F_{\theta_{l-1}}^{(n)}\right\} y_t^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)} \mid \theta_l) + (1-\beta)P(\mathbf{y}^{(n)} \mid \theta_{l-1})}}{\sum_{n=1}^{M} \frac{(1+\beta)F_{\theta_l}^{(n)} + (1-\beta)F_{\theta_{l-1}}^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)} \mid \theta_l) + (1-\beta)P(\mathbf{y}^{(n)} \mid \theta_{l-1})}} \quad (63)$$

where

$$F_{\theta_l}^{(n)} = \sum_{t=1}^{T} P(s_t^{(n)} = j, k_t = k, \mathbf{y}^{(n)} \mid \theta_l). \quad (64)$$

Similarly to (63), the update equation for the covariance matrix is obtained by using a matrix differentiation with respect to $\Sigma_{jk|\theta_l}^{-1}$.

$$\Sigma_{jk|\theta_{l+1}} = \frac{\sum_{n=1}^{M} \frac{(1+\beta)G_{\theta_l}^{(n)} + (1-\beta)G_{\theta_{l-1}}^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)} \mid \theta_l) + (1-\beta)P(\mathbf{y}^{(n)} \mid \theta_{l-1})}}{\sum_{n=1}^{M} \frac{(1+\beta)F_{\theta_l}^{(n)} + (1-\beta)F_{\theta_{l-1}}^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)} \mid \theta_l) + (1-\beta)P(\mathbf{y}^{(n)} \mid \theta_{l-1})}} \quad (65)$$

Here, $G_{\theta_l}^{(n)}$ is as follows.

$$G_{\theta_l}^{(n)} = \sum_{t=1}^{T} P(s_t^{(n)} = j, k_t = k, \mathbf{y}^{(n)} \mid \theta_l)(y_t^{(n)} - \mu_{jk|\theta_l})(y_t^{(n)} - \mu_{jk|\theta_l})^T \quad (66)$$

C. Semi-Continuous Case

Update equations for multiple sequences of this case can be obtained by restricting the state-dependence of the mean vector and the covariance.

$$\overline{\mu}_{k|\theta_{l+1}} = \frac{\sum_{n=1}^{M} \frac{\left\{(1+\beta)\overline{F}_{\theta_l}^{(n)} + (1-\beta)\overline{F}_{\theta_{l-1}}^{(n)}\right\} y_t^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)} \mid \theta_l) + (1-\beta)P(\mathbf{y}^{(n)} \mid \theta_{l-1})}}{\sum_{n=1}^{M} \frac{(1+\beta)\overline{F}_{\theta_l}^{(n)} + (1-\beta)\overline{F}_{\theta_{l-1}}^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)} \mid \theta_l) + (1-\beta)P(\mathbf{y}^{(n)} \mid \theta_{l-1})}} \quad (67)$$

where

$$\overline{F}_{\theta_l}^{(n)} = \sum_{t=1}^{T} P(k_t = k, \mathbf{y}^{(n)} \mid \theta_l). \quad (68)$$

Similarly to (67), the update equation for the covariance matrix is obtained by deleting the dependency on the state.

$$\overline{\Sigma}_{k|\theta_{l+1}} = \frac{\sum_{n=1}^{M} \frac{(1+\beta)\overline{G}_{\theta_l}^{(n)} + (1-\beta)\overline{G}_{\theta_{l-1}}^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)} \mid \theta_l) + (1-\beta)P(\mathbf{y}^{(n)} \mid \theta_{l-1})}}{\sum_{n=1}^{M} \frac{(1+\beta)\overline{F}_{\theta_l}^{(n)} + (1-\beta)\overline{F}_{\theta_{l-1}}^{(n)}}{(1+\beta)P(\mathbf{y}^{(n)} \mid \theta_l) + (1-\beta)P(\mathbf{y}^{(n)} \mid \theta_{l-1})}} \quad (69)$$

Here, $\overline{G}_{\theta_l}^{(n)}$ is as follows.

$$\overline{G}_{\theta_l}^{(n)} = \sum_{t=1}^{T} P(k_t = k, \mathbf{y}^{(n)} \mid \theta_l)(y_t^{(n)} - \mu_{jk|\theta_l})(y_t^{(n)} - \mu_{jk|\theta_l})^T \quad (70)$$

D. Discrete and Continuous Case

So far, we presented six types of alpha-HMMs, i.e., {discrete, continuous, semi-continuous}×{single sequence, multiple sequences}. The rest is the case of the partitioned continuous alphabet [13]. This has an interpretation of a mixture of discrete and continuous letters. Update equations for the mixture of discrete and continuous letters are obtained by changing (39) and (60) - (62) to double summations and by allowing null outputs. For instance, the summation of the denominator of (39) is divided into a partition.

$$\sum_{t=1}^{T}(\cdot) = \sum_{g=1}^{G}\sum_{t\in D_g}(\cdot)$$

Here, D_g is a non-overlapping set in the partition. G is the cardinality of the partition. If D_g corresponds to a specific sub-class of continuous alphabet, we can regard that the alphabet is accompanied with a discrete symbol. This is the case of the discrete and continuous alphabet. There are cases of single and multiple sequences. Therefore, we had eight basic types of alpha-HMMs.

V. Theoretical Analysis of Fast Properties

A. Fast Property as an Alpha-EM Algorithm

As was observed in Section II.B, the alpha-HMM is a sub-class of the alpha-EM algorithm. Therefore, fast properties of the alpha-EM which outperform the log-EM in the convergence speed are inherited. The most relevant properties are as follows.

Denote the derivative of a function as follows.

$$\partial^{ij} f(\psi \mid \varphi) \equiv \frac{\partial^{i+j} f(\psi \mid \varphi)}{\partial^i \psi \, \partial^j \varphi} \quad (71)$$

Then, the following proposition which is an interpretation of Theorem 7 of [8] holds.

[Proposition 2]

Let θ^* be a stationary point and let θ_{l+1} be the updated value of the M-step toward this stationary point. Then, the following expansion holds.

$$\theta_{l+1} - \theta^* = J^{(\alpha)}(\theta^*)(\theta_l - \theta^*) + o(1) \tag{72}$$

Here, $J^{(\alpha)}(\theta^*)$ is the Jacobian matrix such that

$$J^{(\alpha)}(\theta^*) = \left[-\partial^{20} Q^{(\alpha)}_{X|Y,\theta^*}(\theta^* \mid \theta^*)\right]^{-1} \times \left[\partial^{11} Q^{(\alpha)}_{X|Y,\theta^*}(\theta^* \mid \theta^*)\right] \tag{73}$$

This proposition tells that it is important to analyze the Jacobian matrix (73) in order to evaluate the convergence speed. On this evaluation, the following proposition holds.

[Proposition 3]

If

$$E_{P(x|y,\theta^*)}[-\partial^2 \log P(y \mid \theta^*) > 0] \tag{74}$$

holds, then the following inequality on the Jacobian matrices holds for $\alpha \in (-1,1)$ or $\beta \in (1,3)$.

$$J^{(\alpha)}(\theta^*) < J^{(-1)}(\theta^*) \tag{75}$$

There are comments on this proposition as follows.

(a) A sufficient condition for (74) is

$$-\partial^2 \log P(y \mid \theta^*) > 0. \tag{76}$$

This is the log-concavity of the incomplete data likelihood. If data comes from an exponential family, this inequality is satisfied.

(b) A Gaussian mixture with probabilistic weights is a member of the exponential family. Therefore, (76) holds, and hence (75) is satisfied. This assures the speedup theoretically. Note that the case of $\alpha = -1$ or $\beta = 1$ is the log-EM.

(c) The case of the causal approximation is $\alpha_{\text{causal}} \equiv \beta = \alpha + 2$.

B. Performance on Data

The previous sub-section tells that the speedup by the alpha-HMM is possible for the range of $\alpha \in (-1,1)$, i.e., $\alpha_{\text{causal}} \equiv \beta \in (1,3)$. This is for the situation that the source data has a Markov process pdf or pmf. Modeling by an exponential family may not coincide with the nature of given data (this holds even for the log-HMM). Therefore, the choice of α_{causal} depends on the nature of the source.

[EXP1: Discrete symbol case]

Fig. 2 shows a convergence trend of a discrete alphabet HMM estimation. The horizontal line stands for the number of iterations. The vertical axis is a convergence performance measured by a log-likelihood. The topology is a two-state ergodic one. Even for such a simple case, the speedup by the alpha-HMM is thus clear. But, there are bumpy trends on the improvement of the speed. Such phenomena can be found as the parameter approaches to the limit of the convergence $\beta \le 3$. Can this undesirable property be improved? The answer is yes.

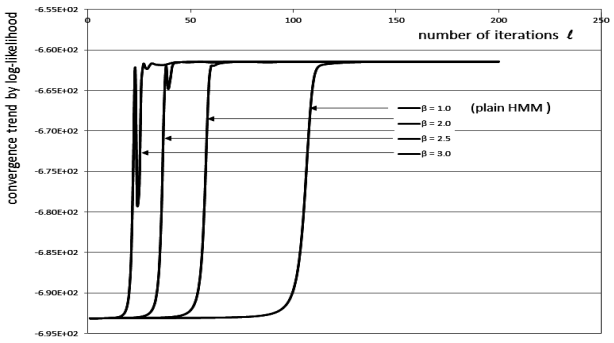

Fig.2. Convergence trend of a discrete alphabet case.

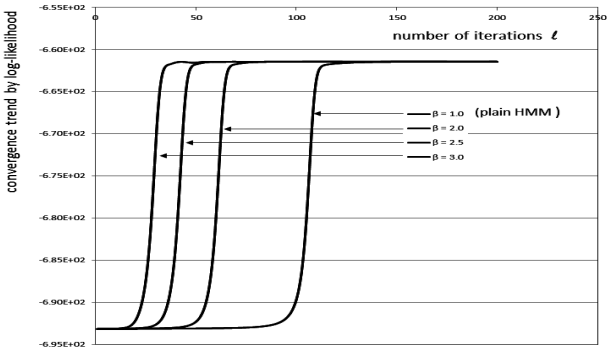

Fig.3. Convergence trend of a discrete alphabet case with a likelihood ratio weighting

Fig. 3 shows the trend of the convergence where the incomplete-data likelihood ratio $P(\mathbf{y} \mid \theta_{l+1}) / P(\mathbf{y} \mid \theta_l)$ is further multiplied to the numerator and the denominator of the update term. Comparing Fig. 2 and Fig. 3, one finds that the speedup of Fig. 3 is slightly less than that of Fig. 2. But, both Fig. 2 and Fig. 3 show that the alpha-HMM beats the traditional HMM estimation clearly.

[EXP2: Single Gaussian]

It can be conceived easily that the case of discrete symbol and that of continuous one may differ. This is true, however, the speedup of the model estimation by the alpha-HMM holds.

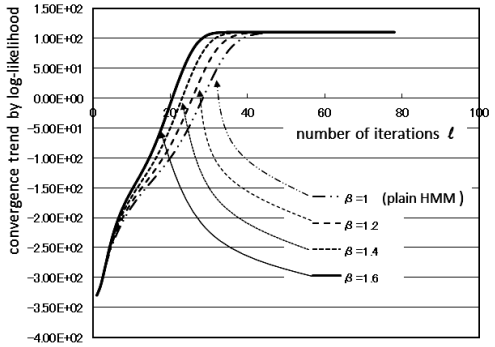

Fig.4. Convergence trend of a continuous alphabet case.

Fig. 4 illustrates a case of an un-mixed Gaussian pdfs where the topology is ergodic. Test data are artificially generated. The speedup of the convergence is clearly

understood from this figure. But, the continuity of symbols causes smoother rising-up than those of discrete cases.

[EXP3: Gaussian mixtures for simulated data]

If we use Gaussian mixture models, graphical rising-ups look similar over the choice of β since the HMM criterion is merely a weak convergence on the likelihood. But, parameters show ripples near the convergence region. Therefore, a convergence criterion reflecting both the likelihood and parameter norms was used. We generated twin-peak triangular pdfs. In this case, the convergence for β={0.8, 0.9, **1.0**, 11, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 21., 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, **3.0**} required iterations of {58, 57, **56**, 55, 55, 54, 53, 52, 52, 51, 50, 49, 49, 48, 48, 47, 47, 46, 46, 45, 45, 45, **45**}, respectively. Thus, the speedup ratio in this case was 56/45=1.22.

[EXP4: Gaussian mixtures for brain signals]

We made experiments on brain signals which are used for the control of humanoids [15]. These signals were measured by NIRS to form a vector time series of a de-oxy-hemoglobin change (ΔHHb) and a normalized tissue hemoglobin index (nTHI) while a subject reads a book. This real-world data is *less stochastic* than the case of EXP3. Therefore, wrong assignments of initial conditions cause NAN (not a number) during the learning iterations. This is inevitable even for the traditional log-HMM. In this experiment, the convergence for β={0.8, 0.9, **1.0**, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, **1.8**, 1.9} required iterations of {17, 17, **16**, 16, 15, 15, 14, 14, 13, 13, **12**, diverge}. Thus, the speedup ratio in this case was 16/12=1.33 even for such a less stochastic case. We note that the divergence was caused by numerical underflows giving NANs.

VI. Concluding Remarks

Following comments by referees, this section summarizes and gives remarks on the significance of the results in this paper and know-hows connecting the theory of the alpha-HMM to its software by using this extra page. They are summarized below.

[Main contributions]

Since the presentation of [14], it has been conceived that further important cases of the HMM including the multiple sequence version, continuous alphabet version and the continuous-discrete version could be possible to derive. This paper covers such cases. Although these algorithms are derived from the alpha-EM, their forms of missing data are versatile. Therefore, different derivations are required as was given in the main text.

[Guidelines connecting the theory with software]

(a) Difficulties caused by ill-conditioning such as numerical underflow need to be avoided using the scaling [3]. Ill-conditioned data cannot be avoided even by the traditional log-HMM. For instance, source data with a univariate Cauchy distribution cause essential instability [1], [9].

(b) A comparison of Figs. 2 and 3 gives there will be many variants to the eight basic cases given in this paper depending on the nature of source data. Since there are so many types of data sources, so are branched versions of the alpha-HMM, each of which will be a good contribution to the field. Such classes include contemporary human-aware applications of HMMs such as BMI (Brain Machine Interface) [15] (EXP4), bioinformatics [6], and adaptable speech processing [3], [4], [5], [10], [12], [13]. The speedup by the alpha-HMM will be credited there.

(c) If a target data is plain which can be modeled by a simple log-HMM converging within only a few iterations, the alpha-HMM will have a reduced opportunity. Even for such a case, however, the traditional or log-HMM cannot beat the alpha-HMM in the speed of the convergence.

(d) Continuous HMMs are significant since they do not require vector quantization in advance. But, such HMMs are often subject to numerical instability more than the discrete case. Readers are requested to understand that this is an essential problem in the HMM even for the traditional log version.

Acknowledgement

The author is grateful to his former students Messrs. Takayuki Ikeda and Ryunoske Hayashi for their valuable assistance. Mr. Ryota Yokote gets credits for his help on experiments.

References

[1] L. E. Baum, T. Petrie, G. Soules and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Annals of Math. Statistics*, vol. 41, pp. 164-171, 1970.

[2] L. R. Welch, "Hidden Markov models and the Baum-Welch algorithm," The Shannon Lecture, IEEE Information Theory Society News Letter, vol. 53, No. 4, pp. 1 and 10-13, 2003.

[3] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, pp. 257-286, 1989.

[4] X. D. Huang, Y. Ariki, and M.A. Jack, *Hidden Markov Models for Speech Recognition*, Edinburgh University Press, Edinburgh, UK, 1990.

[5] L. R. Rabiner and B-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey: Prentice-Hall, 1993.

[6] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of proteins and Nucleic Acids.* Cambridge University Press, Cambridge, UK. 1998

[7] A. P. Dempster, N. M. laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm (with discussions)," *Journal of Royal Statistical Society*, *B*, vol. 39, pp. 1-38, 1977.

[8] Y. Matsuyama, "The alpha-EM algorithm: Surrogate likelihood maximization using alpha-logarithmic information measures," *IEEE Trans. on Inform. Theory*, vol. 49, pp. 692-706, 2003.

[9] L. A Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources" IEEE Trans. IT, vol. 28, pp. 729-734, 1982.

[10] B.-H. Juang, "Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains," AT & T Tech. J., vol. 64, pp. 1235-1245, 1985.

[11] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*, http://matrixcookbook.com, 2008.

[12] X. D. Huang, "Phoneme classification using semicontinuous hidden Markov models," IEEE Trans. SP, 40, pp. 1062-1067, 1992.

[13] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM," Trans. IEICE, D-II, vol. J83-D-II, pp. 1579-1589, 2000.

[14] Y. Matsuyama and R. Hayashi, "Alpha-EM gives fast hidden Markov model estimation: Derivation and evaluation of alpha-HMM," Proc. IJCNN, pp. 663-670, July, 2010.

[15] Y. Matsuyama, K. Noguchi, T. Hatakeyama, No. Ochiai, and T. Hori, "Brain signal recognition and conversion towards symbiosis with ambulatory humanoids," Lecture Notes in Artificial Intelligence, No. 6334, pp. 101-111, 2010.