変分ベイズ法に基づく音響モデルのトポロジー学習および混合分布要素の分割

實廣　貴敏[†]　　中村　　哲[†]

[†] ATR 音声言語コミュニケーション研究所
〒619–0288 「けいはんな学研都市」光台二丁目２番地２
E-mail: [†]{takatoshi.jitsuhiro,satoshi.nakamura}@atr.jp

あらまし　本論文では，変分ベイズ法に基づく手法で，非均一な長さの環境依存 HMM 構造を自動で生成する方法および混合分布要素を分割する方法を提案する．ゆう度最大化規準 (ML) が HMM 構造作成に一般に用いられるが，過学習問題が存在する．これを避けるため，近年，変分ベイズ法が音声認識の音響モデルに適用されてきている．我々は変分ベイズ法を逐次状態分割法 (Successive State Splitting, SSS) 法に適用する．この手法は環境方向だけでなく，時間方向のバリエーションを生成できる．実験結果から，提案法は従来法より効率的なモデルを自動で作成できることが分かった．さらに，時間方向の構造も考慮した変分ベイズ法に基づく混合分布数の増加手法についても検討した．変分ベイズ的手法により，従来法と比較して，より少ない混合数でほぼ同様な性能が得られることが分かった．

キーワード　音声認識, 音響モデル, トポロジー学習, SSS 法, 変分ベイズ法

Variational Bayesian Based Topology Training and Mixture Component Splitting for Acoustic Modeling

Takatoshi JITSUHIRO[†] and Satoshi NAKAMURA[†]

[†] ATR Spoken Language Translation Research Laboratories
2–2–2 Hikaridai, "Keihanna Science City" Kyoto 619–0288 Japan
E-mail: [†]{takatoshi.jitsuhiro,satoshi.nakamura}@atr.jp

Abstract　We propose a automatic generation method of non-uniform and context-dependent HMM topology and a splitting method of mixture components based on the Variational Bayesian (VB) approach. Although the Maximum Likelihood (ML) criterion is generally used to create HMM topologies, it has an over-fitting problem. Recently, to avoid this problem, the VB approach has been applied to create acoustic models for speech recognition. We introduce the VB approach to the Successive State Splitting (SSS) algorithm, which can create both contextual and temporal variations for HMMs. Experimental results show that the proposed method can automatically create more efficient models than those by the original method. We employ the VB approach to increase the number of mixture components. The VB approach obtained almost the same performance with the smaller number of mixture components in comparison with that obtained by using ML-based methods.

Key words　speech recognition, acoustic model, topology training, SSS algorithm, variational Bayesian approach

1. Introduction

To create acoustic models, phonetic decision tree clustering[1] is widely used as a method of generating tied-state structures. It can generate contextual variations. the Maximum Likelihood Successive State Splitting (ML-SSS) algorithm has been proposed as a method to create contextual and temporal variations[3]. These methods used the Maximum Likelihood (ML) criterion to choose better splitting or clustering. However, the ML criterion often results in a model that over-fits the training data. Because the likelihood value for training data increases as the number of parameters increases, it is impossible to find the best model by using the ML criterion only.

To solve this problem, information criteria such as the Minimum Description Length (MDL) criterion and the Bayesian Information Criterion (BIC) have been introduced as splitting and stop criteria for creating context-dependent Hidden Markov Models (HMM). There are also some methods using phonetic decision tree clustering [4], or the SSS algorithm (MDL-SSS) [5]. These methods continue to split states as the information criteria are improved. Although they work well in practical terms, conventional information criteria require some assumptions, e.g., asymptotic normality, and they cannot exactly evaluate complicated models like neural networks, or HMMs, which cannot satisfy such assumptions.

In the field of machine learning, the Variational Bayesian (VB) method was proposed to avoid over-fitting by ML estimation [6]. Recently, the VB approach has been applied to speech recogniton. Decision tree clustering with the VB method was proposed [7], and Variational Bayesian GMMs were applied to speech recognition [8].

We propose an automatic topology creation method using the SSS algorithm with the Variational Bayesian method, which we call the VB-SSS algorithm, to estimate topologies more exactly [9]. The SSS algorithm can create contextual and temporal variations. In contrast, decision tree clustering can only create contextual variations. In [7], they describe the general parameter estimation of HMMs based on the VB approach and the topology estimation by tree-clustering based on the VB approach. In the decision tree clustering, the number of states per triphone must be decided before clustering, and it is never changed after clustering. Therefore, our proposed method, the SSS algorithm based on the VB approach, has a higher number of degrees of freedom than that of the decision tree clustering.

We also evaluate a method for increasing the number of mixture components by using the VB approach, based on a topology obtained by the VB-SSS algorithm. In [7], they evaluated two methods for constructing Gaussian mixture models. One sets the same number of Gaussians per state for all states, and selects an appropriate model by a VB objective function. The other determines the number of Gaussians for each state by splitting and merging Gaussians in each state with the objective function. In [8], Valente and Wellekens produced GMMs by decreasing the number of mixture components in each phoneme. Since the VB-SSS algorithm generates HMM structures with temporal structures, our proposed methods consider temporal structures to make mixture models by splitting Gaussians with the VB approach.

In Section 2., we present the VB-SSS algorithm, and in Section 3. explain a method for increasing the mixture components. In Section 4., we evaluate the performance of our proposed methods with experiments. Finally, we provide our conclusion in Section 5..

2. Variational Bayesian Approach for SSS Algorithm

2.1 Overview of VB-SSS

Our proposed method is based on the ML-SSS algorithm [3]. The ML-SSS algorithm assumes that each state has a single Gaussian distribution, and that each category can be represented by one Gaussian distribution when splitting is performed. This algorithm also assumes that suboptimal models can be obtained by increasing the number of mixture components after this topology training even if such models are not optimal for the number of parameters. Therefore, our proposed method, the VB-SSS algorithm, also uses only a single Gaussian model, and after this algorithm, there is a need for a method to increase the number of mixture components.

Figure 1 shows the flow of the VB-SSS algorithm. This section briefly explains the VB-SSS algorithm. First, the topology of an initial model is set and its parameters are estimated. Second, the prior parameters for each state are set, after which the posterior parameters for each state are estimated, and the VB objective function, $\mathcal{F}_m$, (see [6] for details) is calculated as the baseline energy.

After that, each type of splitting is performed in the same manner as with the ML-SSS algorithm. For each splitting, after two new states are created, the posterior parameters are estimated, and the energy gains of both the contextual splitting and the temporal splitting are calculated. Next, the state splitting with the maximum energy gain is selected. If there is no state that can increase its energy, the splitting is stopped. Furthermore, when $\mathcal{F}_m$ decreases or converges, the splitting is stopped. Otherwise, the parameters of HMMs are estimated, and these procedures are repeated. In this paper, all of the posterior parameters are estimated by using all of the data for each test splitting.

2.2 Contextual and temporal splitting

The probability density of the HMM Θ, which has N_s states with one Gaussian distribution and N_a transitions for each state for both contextual and temporal splitting, is

$$p(O|\Theta) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{s_t}, \boldsymbol{\Sigma}_{s_t}) a_{s_t r_{t+1}}, \qquad (1)$$

where $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_t, \ldots, \boldsymbol{o}_T\}$ is a set of training samples, s_t denotes the state number at time t, and r_t represents the transition arc number at time t. In addition, $\boldsymbol{\mu}_{s_t}$ is a mean vector at s_t, $\boldsymbol{\Sigma}_{s_t}$ denotes a covariance matrix at s_t, and $a_{s_t r_{t+1}}$ is a transition probability. We use a diagonal matrix as the covariance matrix. The maximum of N_a is N_s, and N_a in this paper can be replaced by N_s. However, this splitting algorithm can use $N_a = 2$ only.

The probability for the complete data set to which the latent variables are introduced is

$$p(\boldsymbol{O}, Z|\Theta) = \prod_{t=1}^{T} \prod_{i=1}^{N_s} \prod_{j=1}^{N_a} \{\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) a_{ij}\}^{z_{ij}^t}, \qquad (2)$$

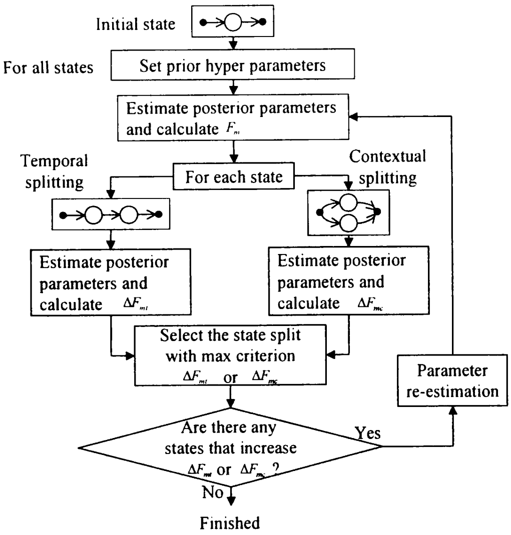
Initial state
For all states
Set prior hyper parameters
Estimate posterior parameters and calculate F_m
Temporal splitting
For each state
Contextual splitting
Estimate posterior parameters and calculate ΔF_{mt}
Estimate posterior parameters and calculate ΔF_{mc}
Select the state split with max criterion ΔF_{mt} or ΔF_{mc}
Parameter re-estimation
Are there any states that increase ΔF_{mt} or ΔF_{mc} ?
Yes
No
Finished

Figure 1 Flow of the Variational Bayesian SSS algorithm.

where $Z = \{z_{ij}^t\}_{i=1,j=1,t=1}^{N_s,N_a,T}$ is the set of latent variables.

The objective function $\mathcal{F}_m$ is defined as a lower bound of a marginal likelihood over all random quantities with a fixed structure m [6]:

$$\mathcal{F}_m = \int q(Z)q(\Theta) \ln \frac{p(O, Z|\Theta)p(\Theta)}{q(Z)q(\Theta)} dZd\Theta, \qquad (3)$$

where $q()$ stands for a variational posterior probability, which approximates a true posterior probability; $q()$ becomes the closest distribution to its true posterior probability when $\mathcal{F}_m$ is maximized. An iterative procedure to find the optimal variational posteriors is defined by the partial derivative of $\mathcal{F}_m$ w.r.t. each $q()$. It is referred to as the Variational Bayesian EM Steps.

When the ith state with the HMM parameter Θ_i is split into the i_1th state and the i_2th state, and the parameter $\hat{\Theta}_i$ is estimated for the current splitting, the splitting criterion can be represented by using the objective function $\mathcal{F}_m$ as follows,

$$\Delta\mathcal{F}_m^{(n+1)} = \mathcal{F}_m^{(n+1)}(\hat{\Theta}_i) - \mathcal{F}_m^{(n)}(\Theta_i), \qquad (4)$$

where n is the iteration number.

2.3 Priors

We assume that the probability of parameters can be factorized as follows.

$$p(\Theta) = p(N_s, N_a)p(\boldsymbol{a}|N_s, N_a)p(\boldsymbol{\Sigma}|N_s)p(\boldsymbol{\mu}|\boldsymbol{\Sigma}, N_s). \qquad (5)$$

We also assume that the prior of $\boldsymbol{a} = \{a_{ij}\}_{i=1,j=1}^{N_s,N_a}, a_{ij} \geqq 0, \sum_{j=1}^{N_a} a_{ij} = 1$ is a *Dirichlet* distribution, and that the prior of $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\} = \{\{\boldsymbol{\mu}_i\}_{i=1}^{N_s}, \{\boldsymbol{S}_i\}_{i=1}^{N_s}\}$ is a *normal-Gamma* distribution,

$$p(\boldsymbol{a}|N_s, N_a) = \prod_{i=1}^{N_s} \mathcal{D}(\{a_{ij}\}_{j=1}^{N_a}; \phi_0) \propto \prod_{i=1}^{N_s}\prod_{j=1}^{N_a} a_{ij}^{\phi_0 - 1}$$

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|N_s)$$

$$= \prod_{i=1}^{N_s}\prod_{k=1}^{D} \mathcal{N}(\mu_{ik}; \nu_{0k}, \xi_0^{-1}\sigma_{ik})\mathcal{G}(\sigma_{ik}^{-1}; \eta_0/2, b_{0k}/2),$$

where D is the order of parameters, μ_{ik} and σ_{ik} are the kth elements of $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, respectively, $\mathcal{N}()$ denotes the Gaussian distribution, $\mathcal{G}()$ represents the Gamma distribution, and ϕ_0, ν_{0k}, ξ_0, η_0, and b_{0k}. are prior parameters. The definition of the Gamma distribution is $\mathcal{G}(s; \eta, \lambda) = \frac{\lambda^\eta}{\Gamma(\eta)} s^{\eta-1}\exp(-\lambda s)$, where $\Gamma()$ is the Gamma function.

2.4 Posteriors

We also assume that the posterior probability of parameters can be factorized as follows.

$$q(\Theta) = q(N_s, N_a)q(\boldsymbol{a}|N_s, N_a)q(\boldsymbol{\Sigma}|N_s)q(\boldsymbol{\mu}|\boldsymbol{\Sigma}, N_s). \qquad (6)$$

The posterior probability can be derived from the Variational Bayesian EM algorithm [6].

$$q(\boldsymbol{a}|\boldsymbol{O}, N_s, N_a) = \prod_{i=1}^{N_s} \mathcal{D}(\{a_{ij}\}_{j=1}^{N_a}; \{\phi_{ij}\}_{j=1}^{N_a}), \qquad (7)$$

$$\phi_{ij} = \phi_0 + \bar{N}_{ij},\ \bar{N}_{ij} = \sum_{t=1}^{T} \bar{z}_{ij}^t,\ \bar{z}_{ij}^t =< z_{ij}^t >_{q(Z)},$$

$$q(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\boldsymbol{O}, N_s)$$

$$= \prod_{i=1}^{N_s}\prod_{k=1}^{D} \mathcal{N}(\mu_{ik}; \nu_{ik}, \xi_i^{-1}\sigma_{ik})\mathcal{G}(\sigma_{ik}^{-1}; \eta_i/2, b_{ik}/2), \qquad (8)$$

$$\bar{N}_i = \sum_{t=1}^{T} \bar{z}_i^t,\ \bar{z}_i^t =< z_i^t >_{q(Z)},$$

$$\nu_{ik} = \frac{\bar{N}_i\bar{o}_{ik} + \xi_0\nu_{0k}}{\bar{N}_i + \xi_0},\ \xi_i = \xi_0 + \bar{N}_i,\ \eta_i = \eta_0 + \bar{N}_i,$$

$$b_{ik} = b_{0k} + \bar{c}_{ik} + \frac{\bar{N}_i\xi_0}{\bar{N}_i + \xi_0}(\bar{o}_{ik} - \nu_{0k})^2,$$

$$\bar{\boldsymbol{o}}_i = \frac{1}{\bar{N}_i}\sum_{t=1}^{T} \bar{z}_i^t \boldsymbol{o}_t,\ \bar{c}_{ik} = \sum_{t=1}^{T} \bar{z}_i^t(o_{tk} - \bar{o}_{ik})^2.$$

Here, $< x >_{f(x)} = \int xf(x)dx$ is the expectation of x for $f(x)$. The variational posterior probability of latent variables is also derived in the same manner as the unknown parameters; $\mathcal{F}_m$ can be derived from these priors and posteriors.

The variational posterior probability of latent variables and the VB objective function were also derived. The detail is omitted here.

3. Increasing Mixture Components Based on the VB Approach

3.1 Splitting mixture method

After topologies are obtained by the VB-SSS algorithm, the number of mixture components is increased by the following algorithm based on the VB approach. We define the

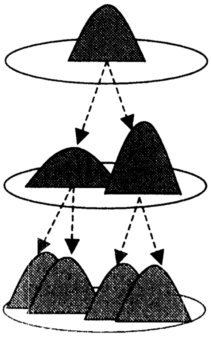

Figure 2 Splitting each distribution into two distributions.

splitting mixture method as follows. **[Splitting mixture method]**

(1) Set an initial model obtained by topology training. $M^{(0)} = 1, n = 0$.

(2) Calculate the objective function $\mathcal{F}_m^{(n)}$.

(3) Iterate the following steps for each phoneme.

(a) Split each distribution into two distributions in each state. $M^{(n+1)} = 2M^{(n)}$. (Fig. 2)

(b) Estimate posterior distributions, and calculate the objective function $\mathcal{F}_m^{(n+1)}$, repeatedly.

(c) Stop splitting when $\Delta\mathcal{F}_m^{(n+1)} = \mathcal{F}_m^{(n+1)} - \mathcal{F}_m^{(n)}$ is a negative number. Otherwise, $n = n + 1$, and go to (a).

This algorithm splits each mixture component to two distributions, as in Fig. 2. In this algorithm, the number of mixture components is estimated for each phoneme. It obtains more suitable models than models with the same number of mixture components for all phonemes.

3.2 VB approach for increasing mixture components

In [7] and [8], the authors estimated the number of mixture components for each state because their methods are the same as those used for GMMs. On the other hand, the VB-SSS algorithm estimates model structures by considering the transition probabilities using the forward-backward algorithm. Therefore, our proposed method estimates the number of mixture components with the forward-backward algorithm for phoneme periods.

Gaussian mixture HMMs can be represented as follows.

$$p(\boldsymbol{O}|\Theta) = \prod_{t=1}^{T} \left\{ \sum_{k=1}^{M_{s_t}^{(n)}} w_{s_t k} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{s_t}, \boldsymbol{\Sigma}_{s_t}) \right\} a_{s_t r_{t+1}}, \quad (9)$$

where s_t denotes the state index at time t, $\{w_{ik}\}_{k=1}^{M_{s_t}^{(n)}}$ is a set of mixture weights for state i, $\boldsymbol{\mu}_{s_t}$ is a mean vector, and $\boldsymbol{\Sigma}_{s_t}$ is a covariance matrix. In addition, r_t is an arc index at time t, $\{a_{ij}\}_{j=1}^{N_a}$ is a set of transition probabilities.

The priors and posteriors for transition probabilities, mean vectors, and precision matrices can be defined just as those of the VB-SSS algorithm. $p(\boldsymbol{a}|N_s, N_a) = \prod_{i=1}^{N_s} \mathcal{D}(\{a_{ij}\}_{j=1}^{N_a}; \phi_0)$ is for transition probabilities, and $p(\boldsymbol{\mu}, \boldsymbol{S}|N_s, \{M_i\}_{i=1}^{N_s}) = \prod_{i=1}^{N_s} \prod_{k=1}^{M_i} \prod_{l=1}^{D} \mathcal{N}(\mu_{ikl}; \nu_{0l}, \xi_0^{-1}\sigma_{ikl}) \mathcal{G}(\sigma_{ikl}^{-1}; \eta_0/2, b_{0l}/2)$ is for mean vectors and precision matrices. For mixture weights, a *Dirichlet* distribution can be used.

$$p(\boldsymbol{w}|N_s, \{M_i\}_{i=1}^{N_s}) = \prod_{i=1}^{N_s} \mathcal{D}(\{w_{ik}\}_{k=1}^{M_i}; \rho_0),$$

where ρ_0 is a prior parameter. The posterior probabilities for these probabilities and the VB objective function, including mixture components, can be derived in the same manner as these in the VB-SSS algorithm.

For recognition, posterior predictive probability is used for the Bayesian approach.

$$p(\boldsymbol{x}|m, \boldsymbol{O}) = \prod_{t=1}^{T} \int p(\boldsymbol{x}_t|\Theta_{s_t s_{t+1}}, m, \boldsymbol{O}) \times p(\Theta_{s_t s_{t+1}}|m, \boldsymbol{O}) d\Theta_{s_t s_{t+1}}. \quad (10)$$

Here, $\boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$ is a set of test data, and m represents a structure indicator, that is, the number of states, transitions, and mixture components in this work. The true posterior probability $p(\Theta_{ij}|m, \boldsymbol{O})$ is approximated by the variational posterior probability $q(\{a_{ij}\}_{j=1}^{N_a}|m) \prod_{k=1}^{M_i} q(\{w_{ik}\}_{k=1}^{M_i}|m) q(\boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}|m)$.

$$p(\boldsymbol{x}|m, \boldsymbol{O}) \simeq \prod_{t=1}^{T} \bar{\phi}_{ij} \sum_{k=1}^{M_i} \bar{\rho}_{ij} \prod_{l=1}^{D} \mathcal{T}(x_{tl}; \nu_{ikl}, \sigma_{ikl}, f_{ik}),$$

$$\bar{\phi}_{ij} = \phi_{ij} / \sum_{j'} \phi_{ij'}, \bar{\rho}_{ij} = \rho_{ik} / \sum_{k'} \rho_{ik'},$$

$$f_{ik} = \eta_{ik}, \sigma_{ikl} = b_{ikl}(\xi_{ik} + 1)/(\xi_{ik} f_{ik}).$$

$\mathcal{T}(\cdot)$ is a *Student-t* distribution.

$$\mathcal{T}(\boldsymbol{x}_t; \boldsymbol{\nu}_{ik}, \Phi_{ik}, f_{ik}) = C_{ik}\{1 + (\boldsymbol{x}_t - \boldsymbol{\nu}_{ik})'(f_{ik}\Phi_{ik})^{-1}(\boldsymbol{x}_t - \boldsymbol{\nu}_{ik})\}^{-\frac{1}{2}(f_{ik}+1)},$$

$$C_{ik} = \frac{\Gamma((f_{ik} + D)/2)}{(f_{ik}\pi)^{D/2}\Gamma(f_{ik}/2)|\Phi_{ik}|^{\frac{1}{2}}}.$$

Here, "′" represents a transpose.

4. Experiments

4.1 Experimental conditions

We compared our proposed method, the VB-SSS, to the ML-SSS and the MDL-SSS algorithms [5]. For the ML-SSS, two models with different maximum state lengths, 3 or 4, were created. These two models are the baseline models.

For the acoustic training set, we used Japanese dialog speech from the ATR travel arrangement task (TRA) database [10] uttered by 166 males. The total length of speech was 2.1 hours.

For testing, we used dialog speech that includes 213 sentences from the TRA database uttered by a different set of 17 males. For topology training, we employed the VB approach only for the splitting and stopping criteria. Multi-class composite bigram models [11] were used, and the vocabulary size

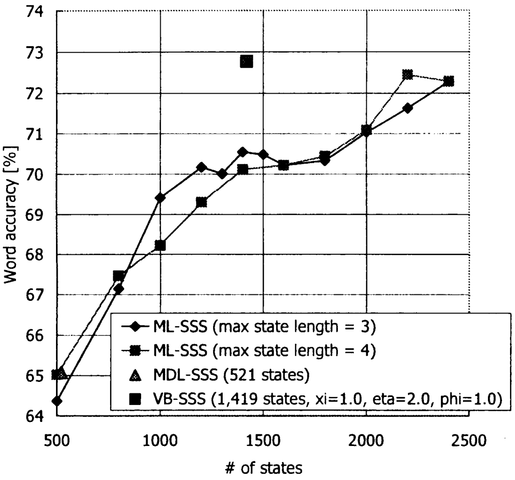

Figure 3 Word accuracy rates by single Gaussian models.

Table 1 Word accuracy rates [%] and # of states in parentheses for several hyperparameters.

$\xi_0 = 0.1$	$\phi_0 = 1.0$	$\phi_0 = 10$	$\phi_0 = 100$
$\eta_0 = 0.2$	68.87 (766)	69.14 (764)	68.76 (775)
$\eta_0 = 2.0$	72.12 (1,361)	71.30 (1,252)	67.73 (1,246)
$\xi_0 = 1.0$	$\phi_0 = 1.0$	$\phi_0 = 10$	$\phi_0 = 100$
$\eta_0 = 0.2$	68.87 (760)	68.92 (761)	68.87 (749)
$\eta_0 = 2.0$	<u>72.77</u> (1,419)	<u>72.55</u> (1,425)	72.17 (1,433)
$\xi_0 = 10$	$\phi_0 = 1.0$	$\phi_0 = 10$	$\phi_0 = 100$
$\eta_0 = 0.2$	70.01 (771)	68.60 (757)	69.09 (780)
$\eta_0 = 2.0$	71.30 (1,315)	72.06 (1,358)	66.00 (1,211)

was 5,000. The sampling frequency was 16 kHz, the frame length was 20 ms, and the frame shift was 10 ms. We used 12-order MFCC, ΔMFCC, and Δ log power as feature parameters. In addition, cepstrum mean subtraction was applied to each utterance. We used 26 kinds of phonemes and one silence. Three states were used as the initial model for each phoneme, and one Gaussian distribution for each state was used during topology training. A silence model with three states was built separately from the phoneme models, and to increase the number of mixture components with the VB approach, the number of Gaussians for the silence model was determined by employing the VB approach. In these experiments, we used $\phi_0 = 1.0$, $\xi_0 = 1.0$, $\eta_0 = 2.0$ for the prior parameters of the VB-SSS. ν_{0k} and b_{0k} were set from the element values of the mean vectors and the covariance matrices.

4.2 Evaluation for topology training

Figure 3 shows the results by using the single Gaussian models. The performance of the MDL-SSS was again worse than the baseline, ML-SSS, due to the small amount of training data. On the other hand, with about 60% of the ML-SSS states, the VB-SSS achieved a comparable recognition rate.

Next, we analyzed the dependencies of the prior hyperparameters. Table 1 shows word accuracy rates and the number of states of several prior parameters for the 5k-CSR task, with the trend of results for segmented phoneme recognition being almost the same. The fluctuation of performance is small when ϕ_0 is changed under almost the optimal values, $\xi_0 = 1.0$ and $\eta_0 = 2.0$. Also, ϕ_0 is a hyperparameter of transition probabilities. Because transition probabilities do not have much effect on recognition performance, the influence of ϕ_0 is smaller than the other parameters.

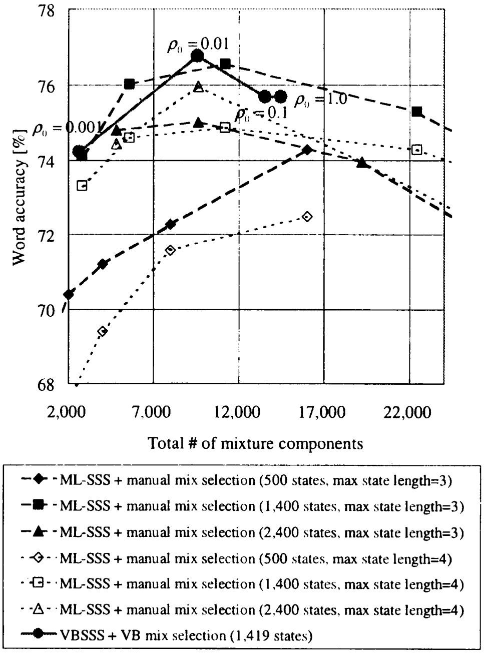

Figure 4 Word accuracy rates by Gaussian mixture models.

4.3 Evaluation of mixture splitting

Figure 4 shows the results by using the splitting mixture method. Furthermore, Table 2 shows the average number of mixture components, the total number of mixture components, and word accuracy rate for the best model of the baseline and the models by using the VB approach with several values of the prior parameter, ρ_0. Posterior predictive probabilities defined by Eq. (10) are used for decoding by Bayesian approach, showing that the VB approach obtained almost the same performance with a 15%-smaller number of Gaussians than that obtained by using the ML based method. These results indicate that recognition performance is dependent on ρ_0. This posterior parameter is updated by $\rho_{ik} = \rho_0 + \bar{N}_{ik}$. The effectiveness of ρ_0 is dependent on the number of samples, $\bar{N}_{ik}$; the larger the number of samples, the smaller the effect. Furthermore, the amount of training data in these experiments is too small for use as conventional training data.

In addition, we evaluated four combinations of topology training methods and mixture selection methods. This ex-

Table 2 The average number of mixture components per state, the total number of mixture components, and word accuracy rate

	ρ_0	#mixtures /state	#mixtures	WA[%]
ML-SSS + manual mix selection (1,400 states)	–	8	11,200	76.56
VB-SSS	0.001	1.87	2,652	74.23
+ VB mix	0.01	6.74	9,564	76.77
selection	0.1	9.53	13,520	75.69
(1,419 states)	1.0	10.19	14,460	75.69

periment can show that criteria both for topology training and mixture selection should be consistent. For topology training, we can select either the ML-SSS or the VB-SSS, while for mixture selection, we can use the ML-based manual selection or the VB-based method. The decoding method is dependent on the parameter estimation method, and for ML-based manual selection, it is the usual ML-based decoding (ML decoding) method. For models trained by VB-based mixture selection, posterior predictive probabilities are used for decoding. This is called "PPP decoding" for short in this section. Therefore, there are four combinations as listed below.

(1) ML-SSS + manual mixture selection + ML decoding

(2) VB-SSS + VB mixture selection + PPP decoding

(3) ML-SSS + VB mixture selection + PPP decoding

(4) VB-SSS + manual mixture selection + ML decoding

In both methods (1) and (2), the criteria for both topology training and mixture selection are the same, and their results are the same as those in Fig. 4 and Table 2.

Figure 5 shows word accuracy rates achieved by these four combinations. The VB approach both for topology training and mixture selection gave the best result among these combined methods.

5. Conclusion

We proposed using the Variational Bayesian approach to automatically create non-uniform, context-dependent HMM topologies. We introduced the VB approach to the SSS algorithm to create contextual and temporal variations for HMMs and then defined posterior probability densities and the VB free energy as split and stop criteria. The VB-SSS automatically achieved comparable performance with about 60% of states generated by the ML-SSS. Furthermore, we evaluated a method for increasing the number of mixture components, employing the VB approach. Experimental results indicated that the VB approach could obtain almost the same performance with a 15%-smaller number of Gaussians than that obtained by using the ML-based method.

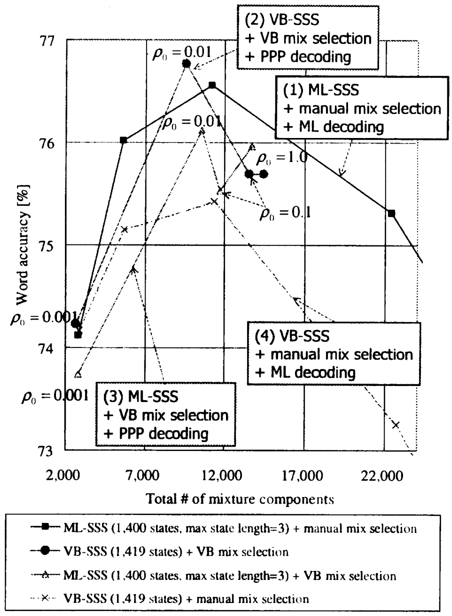

Figure 5 Word accuracy rates by four types of combinations.

Acknowledgment

This research was supported in part by the National Institute of Information and Communications Technology.

Reference

[1] S. J. Young, J. J. Odell, P. C. Woodland, "Tree-based state tying for high accuracy acoustic modeling," Proc. of the ARPA Workshop on Human Language Technology, pp. 307–312, 1994.

[2] J. Takami, S. Sagayama, "A Successive State Splitting Algorithm for Efficient Allophone Modeling," Proc. ICASSP'92, vol. 1, pp. 573–576, 1992.

[3] M. Ostendorf, H. Singer, "HMM topology design using maximum likelihood successive state splitting," Computer Speech and Language, vol. 11, pp. 17–41, 1997.

[4] K. Shinoda, T. Watanabe, "Acoustic modeling based on the MDL principle for speech recognition," Proc. of EUROSPEECH'97, pp. 99–102, 1997.

[5] T. Jitsuhiro, T. Matsui, S. Nakamura, "Automatic Generation of Non-uniform HMM Topologies Based on the MDL Criterion," IEICE Trans. Inf. & Syst., vol. E87-D, no. 8, pp. 2121–2129, 2004.

[6] H. Attias, "A variational Bayesian framework for graphical models," In Advances in Neural Information Processing Systems 12, 2000.

[7] S. Watanabe, Y. Minami, A. Nakamura, N. Ueda, "Variational Bayesian estimation and clustering for speech recognition," IEEE Trans. Speech and Audio Processing, vol. 12, no. 4, pp. 365–381, 2004.

[8] F. Valente, C. Wellekens, "Variational Bayesian GMM for speech recognition," Proc. of EUROSPEECH'03, vol. 4, pp. 441–444, 2003.

[9] T. Jitsuhiro, S. Nakamura, "Automatic Generation of Non-Uniform HMM Structures Based on Variational Bayesian Approach," Proc. of ICASSP2004, vol. I, pp. 805–808, 2004.

[10] T. Takezawa, T. Morimoto, Y. Sagisaka, "Speech and language databases for speech translation research in ATR," Proc. of the 1st International Workshop on East-Asian Language Resources and Evaluation (EALREW'98), 1998.

[11] H. Yamamoto, Y. Sagisaka, "Multi-class composite n-gram based on connection direction," Proc. of ICASSP'99, vol. 1, pp. 533–536, 1999.