Speech recognition based on statistical models including multiple phonetic decision trees

Sayaka Shiota*, Kei Hashimoto†, Heiga Zen‡, Yoshihiko Nankaku§, Akinobu Lee* and Keiichi Tokuda§

Department of Computer Science, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, 466–8555 Japan

(Received 27 October 2010, Accepted for publication 15 April 2011)

Abstract: We propose a speech recognition technique using multiple model structures. In the use of context-dependent models, decision-tree-based context clustering is applied to find an appropriate parameter tying structure. However, context clustering is usually performed on the basis of unreliable statistics of hidden Markov model (HMM) state sequences because the estimation of reliable state sequences requires an appropriate model structures, that cannot be obtained prior to context clustering. Therefore, context clustering and the estimation of state sequences essentially cannot be performed independently. To overcome this problem, we propose an optimization technique of state sequences based on an annealing process using multiple decision trees. In this technique, a new likelihood function is defined in order to treat multiple model structures, and the deterministic annealing expectation maximization algorithm is used as the training algorithm. Experimental continuous phoneme recognition results show that the proposed method of using only two decision trees achieved about an 11.1% relative error reduction over the conventional method.

Keywords: Continuous speech recognition, Acoustic modeling, Context clustering, Phonetic decision tree, Deterministic annealing

PACS number: 43.72.Ne [doi:10.1250/ast.32.236]

1. INTRODUCTION

In hidden Markov model (HMM) based speech recognition, the expectation maximization (EM) algorithm [1] is widely used for parameter estimation. The EM algorithm provides a simple iterative procedure for obtaining approximate maximum likelihood (ML) estimates. However, since the EM algorithm is a hill-climbing approach, it sometimes suffers from the local maxima problem. To relax this problem, the deterministic annealing EM (DAEM) algorithm has been proposed [2]. In the DAEM algorithm, the problem of maximizing the log-likelihood is reformulated as the problem of minimizing the thermodynamic free energy. The posterior distribution derived in the DAEM algorithm includes a “temperature” parameter that controls the influence of unreliable model parameters. It has been reported that the DAEM algorithm is effective for HMM-based speech recognition [3].

In large vocabulary continuous speech recognition systems, context-dependent models (e.g., triphone HMMs) are widely used. Although a large number of triphones can capture variations in speech data, too many parameters cause overfitting. Therefore, maintaining a good balance between the model complexity and robustness is important for achieving good recognition performance. The phonetic decision-tree-based context clustering technique is one of the good solutions to this problem [4]. This technique yields the parameter tying structure that can assign a sufficient amount of training data to each HMM state. The embedded training followed by the context clustering can provide an estimate of reliable model parameters in accordance with the appropriate model structure. However, the context clustering requires statistics of HMM state sequences obtained from model parameters. That is, although we need reliable model parameters to construct the appropriate parameter tying structure, estimating
reliable model parameters requires the appropriate parameter tying structure. Hence, model parameters and a parameter tying structure should be jointly optimized. However, the exact solution of this optimization is computationally intractable. To overcome this problem, we reformulate this optimization problem as the maximization of a newly defined likelihood function that includes the parameter tying structures as a hidden variable. Then, the variational approximation and the DAEM algorithm are adopted to derive a tractable algorithm for achieving a good suboptimal solution. Furthermore, we propose not only a training algorithm using multiple decision trees, but also a speech recognition technique using multiple decision trees in decoding within the same annealing framework.

The rest of this paper is organized as follows. Section 2 describes the DAEM algorithm, and Section 3 describes speech recognition based on multiple decision trees. Experimental results are presented in Section 4, and concluding remarks and future work are presented in the final section.

2. DETERMINISTIC ANNEALING EM ALGORITHM IN PARAMETER ESTIMATION

2.1. EM Algorithm

The objective of the EM algorithm is to estimate a set of model parameters that maximizes the incomplete log-likelihood function:

$$\mathcal{L}(\Lambda) = \log \sum_q P(o, q \mid \Lambda),$$

(1)

where $o = (o_1, o_2, \ldots, o_T)$ and $q = (q_1, q_2, \ldots, q_T)$ are the observation and state sequences, respectively, and $\Lambda$ denotes a set of model parameters. The EM algorithm iteratively maximizes the auxiliary function, the $Q$-function:

$$Q(\Lambda, \Lambda') = \sum_q P(q \mid o, \Lambda) \log P(o, q \mid \Lambda'),$$

(2)

where $P(q \mid o, \Lambda)$ is the posterior probability of $q$. It can be obtained by applying the Bayes rule as follows:

$$P(q \mid o, \Lambda) = \frac{P(o, q \mid \Lambda)}{\sum_q P(o, q \mid \Lambda)}.$$  

(3)

The EM algorithm starts with an initial model parameter $\Lambda^{(0)}$, and iterates between the following two steps.

E step: compute $Q(\Lambda, \Lambda^{(k)})$

M step: $\Lambda^{(k+1)} = \operatorname{arg \ max}_\Lambda Q(\Lambda, \Lambda^{(k)})$

Here, $k$ denotes the iteration number. This procedure is repeated until the convergence of the likelihood. However, since the EM algorithm is a hill-climbing approach, it sometimes suffers from the local maxima problem.

2.2. Deterministic Annealing EM Algorithm

In the DAEM algorithm [2], the problem of maximizing the log-likelihood function is reformulated as the problem of minimizing the following free energy function:

$$\mathcal{F}_\beta(\Lambda) = -\frac{1}{\beta} \log \sum_q P^\beta(o, q \mid \Lambda)$$

$$= -\sum_q f(q \mid o, \Lambda) \log P(o, q \mid \Lambda)$$

$$- \frac{1}{\beta} I[f(q \mid o, \Lambda)],$$

(4)

where $I[x]$ denotes the entropy of $x$ and $1/\beta$ is called “temperature.” If $\beta = 1$, the negative free energy $-\mathcal{F}_\beta(\Lambda)$ becomes equal to the log-likelihood function $\mathcal{L}(\Lambda)$. In the deterministic annealing approach, the new posterior distribution $f$ is derived so as to minimize the free energy under the constraint of $\sum_q f = 1$. To solve this problem, we can use the elementary calculus of variations to take functional derivatives of Eq. (4) with respect to $f$, and the optimal distribution can be derived as

$$f(q \mid o, \Lambda) = \frac{P^\beta(o, q \mid \Lambda)}{\sum_q P^\beta(o, q \mid \Lambda)}.$$  

(5)

In the DAEM algorithm, the temperature parameter $\beta$ is gradually increased while iterating the EM-steps at each temperature. When $1/\beta$ is set to the initial temperature $\beta^{(0)} \simeq 0$, the EM-steps may achieve a single global minimum of $\mathcal{F}_\beta(\Lambda)$. At the initial temperature, the posterior distribution $f$ takes a form of nearly uniform distribution. While the temperature is decreasing, the form of $f$ changes from uniform to the original posterior distribution. Finally, at the temperature $1/\beta = 1$, the DAEM algorithm is identical to the original EM algorithm.

2.3. Optimization of State Sequences

In the HMM case, the DAEM posterior distribution $f$ can be calculated by the forward-backward algorithm. The numerator of the posterior distribution in Eq. (5) is written as

$$P^\beta(o, q \mid \Lambda) = P^\beta(o \mid q, \Lambda) P^\beta(q \mid \Lambda)$$

$$= \prod_{t=1}^T P^\beta(o_t \mid q_t, \Lambda) \prod_{t=1}^T P^\beta(q_t \mid q_{t-1}, \Lambda),$$

(6)

where $P(o_t \mid q_t, \Lambda)$ and $P(q_t \mid q_{t-1}, \Lambda)$ indicate state output and transition probabilities, respectively. It can be observed that Eq. (6) has the same form as the likelihood function of HMMs. Therefore, the expectations with respect to the DAEM posterior distribution $f$ can be calculated by replacing the state output and transition probabilities with $P^\beta(o_t \mid q_t, \Lambda)$ and $P^\beta(q_t \mid q_{t-1}, \Lambda)$, respectively.
3. SPEECH RECOGNITION BASED ON MULTIPLE PHONETIC DECISION TREES

To optimize state sequences, the EM and DAEM algorithms require a parameter tying structure. However, the parameter tying structure is usually constructed from unreliable model parameters, because an appropriate model structure has not yet been constructed for estimating model parameters. This means that the estimation of state sequences and the construction of model structures depend on each other. Hence, they should be optimized simultaneously. However, the exact solution of this optimization is computationally intractable. Consequently, we reformulate this optimization problem as a maximization of a newly defined likelihood function that includes multiple model structures.

3.1. Acoustic Modeling Based on Model Structure Annealing

To derive the algorithm of model structure annealing, we define a new likelihood function that includes parameter tying structures as a hidden variable as follows:

$$ P(o \mid \Lambda) = \sum_q \sum_m P(o, q, m \mid \Lambda), \quad (7) $$

$$ P(o, q, m \mid \Lambda) = P(m)P(q \mid \Lambda)P(o \mid q, m, \Lambda), \quad (8) $$

where $m \in \{1, \ldots, M\}$ are indexes of parameter tying structures and $\Lambda \in \{A_1, \ldots, A_M\}$ denotes a set of model parameters. We assume each parameter tying structure is represented by a phonetic decision tree. In the EM algorithm, the ML estimation of the model parameters is obtained using the posterior distribution of hidden variables estimated in the E-step. Therefore, the ML solution for the newly defined model is regarded as the simultaneous optimization of state sequences and a parameter tying structure. The free energy function including the multiple decision trees for the DAEM algorithm also can be written as

$$ F_{\beta}(\Lambda) = -\frac{1}{\beta} \log \sum_q \sum_m P^0(o, q, m \mid \Lambda). \quad (9) $$

However, estimating the DAEM posterior distribution $f(q, m \mid o, \Lambda)$ is intractable owing to the combination of hidden variables. To solve this problem, we apply the variational EM algorithm [5]. The objective of the algorithm is to minimize the upper bound of the free energy function. The upper bound of the free energy function $\tilde{F}_{\beta}(\Lambda)$ is defined as

$$ F_{\beta}(\Lambda) = -\frac{1}{\beta} \log \sum_q \sum_m P^0(o, q, m \mid \Lambda) \frac{Q(q, m)}{Q(q, m)} $$

$$ \leq -\frac{1}{\beta} \sum_q \sum_m Q(q, m) \log \frac{P^0(o, q, m \mid \Lambda)}{Q(q, m)} $$

$$ = \tilde{F}_{\beta}(\Lambda), \quad (10) $$

where $Q(q, m)$ is an arbitrary distribution. The upper bound $\tilde{F}_{\beta}(\Lambda)$ can be transformed as follows:

$$ \tilde{F}_{\beta}(\Lambda) = \frac{1}{\beta} KL(Q \parallel f) - \log P(o \mid \Lambda) + \text{const}, \quad (11) $$

where $KL(\parallel)$ denotes the Kullback-Leibler (KL) divergence. The above equation shows that minimizing $\tilde{F}_{\beta}(\Lambda)$ with respect to $Q(q, m)$ is equivalent to minimizing the KL-divergence between $Q$ and $f$. If there is no constraint with distribution $Q$, minimizing $\tilde{F}_{\beta}(\Lambda)$ results in $f = Q$. Assuming a constraint to reduce the complexity, the distribution $Q$ that minimizes $\tilde{F}_{\beta}(\Lambda)$ becomes an approximate distribution of $f$. Hence, we assume the following constraint:

$$ Q(q, m) = Q(q)Q(m), \quad (12) $$

where $\sum_q Q(q) = 1$ and $\sum_m Q(m) = 1$. Using these factorized distributions, the upper bound $\tilde{F}_{\beta}(\Lambda)$ can be rewritten as

$$ \tilde{F}_{\beta}(\Lambda) = -\sum_q \sum_m Q(q)Q(m) \log P(o, q, m \mid \Lambda) $$

$$ = -\frac{1}{\beta} I(Q(q)) - \frac{1}{\beta} I(Q(m)). \quad (13) $$

It can be seen that the temperature parameter $\beta$ changes the ratio between the value of the $Q$-function and the entropy of hidden variables in $\tilde{F}_{\beta}(\Lambda)$. Extending this interpretation, we can control the annealing process of decision trees and state sequences individually. By introducing $\beta_q$ and $\beta_m$, $\tilde{F}_{\beta}(\Lambda)$ is rewritten as

$$ \tilde{F}_{\beta}(\Lambda) = -\sum_q \sum_m Q(q)Q(m) \log P(o, q, m \mid \Lambda) $$

$$ = -\frac{1}{\beta_q} I(Q(q)) - \frac{1}{\beta_m} I(Q(m)). \quad (14) $$

The optimal variational posterior distributions $Q(q)$ and $Q(m)$ are derived by minimizing $\tilde{F}_{\beta}(\Lambda)$. This functional optimization can be solved by the variational method, and the following formulae are obtained:

$$ Q(q) \propto p^0_q(o \mid \Lambda) \exp(\log p^0_q(o \mid q, m, \Lambda))Q(q), \quad (15) $$

$$ Q(m) \propto p^0_m(o \mid \Lambda) \exp(\log p^0_q(o \mid q, m, \Lambda))Q(m), \quad (16) $$

where $(\cdot)_Q$ denotes the expectation with respect to the distribution $Q(\cdot)$. Since Eqs. (15) and (16) are dependent on each other, these updates should be iterated in the E-step. Figure 1 illustrates the joint optimization process based on the DAEM algorithm. At the initial temperature $(\beta_q, \beta_m) = (0, 1)$, the variational posterior distributions $Q(q)$ and $Q(m)$ take forms with nearly uniform distribution. While the temperature is decreasing, the forms of $Q(q)$ and $Q(m)$ change from uniform to each original posterior distribution, and at the final temperature $(\beta_q, \beta_m) = (1)$, $Q(q)$ and $Q(m)$ have different original posterior distributions.
Then, the posterior probability of each model structure is in proportion to the likelihood of each model structure. This process represents the approximation of the joint optimization of the state sequences and the model structures.

3.2. Speech Decoding Based on Multiple Model Structures

In the proposed method, multiple decision trees are used in decoding process. However, the multiple decision trees are inapplicable to standard decoders. Therefore, we propose two types of decoding procedures. One is that a single model structure is chosen by setting the temperature $\beta_q$ to 1 (the DAEM algorithm with $\beta_q = 1$ becomes the Viterbi training. However, the final temperature is fixed as $\beta_q = 1$ in this paper). Although the model structure with the largest decision tree is selected at $\beta_q = 1$ in most cases, reliable state sequences can be obtained by using multiple model structures in the early stage of the training procedure. The other is to use multiple model structures not only in the training process but also in the decoding process. Although there are many approaches to using multiple model structures in decoding [6,7], we use Eq. (7) to control the degree of use of multiple decision trees for training and decoding processes. In preliminary experiments, there was a tendency to select only the largest decision tree at the final stage of the training process. This is because the range of the likelihood was very different among the differently sized decision trees, i.e., the largest decision tree had a significantly higher likelihood than the other trees. Consequently, the posterior probability $Q(m)$ of the largest decision tree became almost 1, and the other trees were not used. Therefore, to use multiple decision trees in the decoding process, we adopt a method in which the annealing is stopped in the early stage of the training process. In this method, the decoding is performed so as to minimize the upper bound $F_g$. Using $Q(m)$, the criterion for decoding can be written as

$$\max_q P(q \mid \Lambda) \prod_m p(Q(m) \mid o, q, m, \Lambda).$$  \hspace{1cm} (17)$$

By inspection, this criterion can be calculated by the output probabilities of a multistream HMM where $Q(m)$ becomes the weight of each stream.

4. EXPERIMENTS

To evaluate the effectiveness of the proposed method, a speaker-dependent continuous phoneme recognition experiment and a speaker-independent continuous word recognition experiment were conducted.

4.1. Speaker-Dependent Phoneme Recognition Experiments

4.1.1. Experimental conditions

In this experiment, we used 503 phonetically balanced sentences uttered by a single male speaker MHT from the ATR Japanese speech database b-set [8]. For training, 450 sentences were used and the remaining 53 sentences were used for testing. The speech data was down-sampled from 20 kHz to 16 kHz, windowed at a 25 ms Blackman window, and parameterized into 19 mel-cepstral coefficients by the mel-cepstral analysis technique [9]. Static coefficients including the zeroth coefficients and their first and second derivatives were used as feature parameters. Three-state left-to-right HMMs were used to model 37 Japanese phonemes, and 144 questions were prepared for decision tree clustering. Each state output probability distribution was modeled by a Gaussian distribution with a diagonal covariance matrix. As a decoder, HVite in HTK [10] was used.

In this experiment, the following five training methods were compared.

- “flat-start”: HMMs were initialized with equal mean and variance for all states using no phoneme boundary labels, and re-estimated using the EM algorithm.
- “$k$-means”: HMMs were initialized by the segmental $k$-means algorithm using phoneme boundary labels and re-estimated using the EM algorithm.
- “DAEM-state”: The DAEM algorithm was applied only to the estimation of state sequences. A single decision tree was used.
"DAEM-tree": The DAEM algorithm was applied only to decision trees. The estimation process of state sequences is equivalent to "flat-start.

"DAEM-joint": The DAEM algorithm was applied to both state sequences and decision trees.

In the methods using a single decision tree ("flat-start," "k-means," and "DAEM-state"), a decision tree is obtained by context clustering based on the minimum description length (MDL) criterion [11]. In addition to this model structure, "DAEM-tree" and "DAEM-joint" use a decision tree representing monophone HMMs. It is desirable to use multiple decision trees. However, when several decision trees are used, we must determine many conditions (e.g., the size and structure of trees, and the number of trees). Although how to determine the number of decision trees and how to construct multiple decision trees are essential problems in the proposed method, in this experiment, we only focus on the evaluation of the integration part of multiple decision trees. Therefore, in this experiment, we simply use only two decision trees for model structure annealing ($m = 1$: monophone, $m = 2$: MDL). In the two-decision-tree case, determining the temperature parameter $\beta_m$ is equivalent to setting the variational posterior probabilities $Q(m)$ directly, because the update equation of $Q(m)$ includes $\beta_m$ in Eq. (16), and the ratio between $Q(1)$ and $Q(2)$ is determined by $\beta_m$. Therefore, $Q(m)$ can also be arbitrarily determined instead of $\beta_m$, and at the start and end of the temperature update, $Q(m)$ should be fixed as follows. When $\beta_m$ is set to 0, all decision trees have the same posterior probabilities ($Q(1) = 0.5$ and $Q(2) = 0.5$). When $\beta_m$ is set to 1, $Q(m)$ is in proportion to the likelihood of each decision tree. However, since MDL has a much higher likelihood than a monophone, the posterior probabilities should be $Q(1) = 0$ and $Q(2) = 1$. Therefore, it was assumed that $Q(m)$ was updated by the following linear functions:

$$Q(\text{monophone}) = 0.5\left(1 - \frac{i}{I}\right),$$

$$Q(\text{MDL}) = 0.5\left(1 + \frac{i}{I}\right).$$

The temperature parameter $\beta_q$ was updated by

$$\beta_q(i) = \left(\frac{i}{I}\right)^\alpha, \quad (i = 0, \ldots, I),$$

where $i$ denotes the iteration number of temperature updates, and $\alpha$ was varied as $\alpha = 2^n$ ($n = -7, \ldots, 7$). Figures 2 and 3 show plots of the schedules of the temperature parameters $\beta_q$ and $\beta_m$, respectively. In the DAEM algorithm ("DAEM-state," "DAEM-tree" and "DAEM-joint"), the number of temperature update steps was set to 20 ($I = 20$), and 10 EM-steps were conducted at each temperature. To evenly compare the proposed method with the conventional method, the number of EM-steps was set to 200 for the standard EM algorithm ("flat-start" and "k-means").

### 4.1.2. Experimental results using a single decision tree in decoding

Figure 4 shows the log-likelihood of the training data. It can be seen that the likelihood of "flat-start" was lower than that of "k-means." This is because "flat-start" uses no phoneme boundary information for initializing HMMs and inappropriate initial model parameters cause the local maxima problem. Although the "DAEM-state" also uses no phoneme boundaries, the likelihood of the "DAEM-state" was close to that of "k-means" when an appropriate temperature schedule was used. This result confirmed that the local maxima problem can be relaxed by using the DAEM algorithm. Comparisons of the proposed structure annealing with the conventional methods reveals that "DAEM-tree" yielded similar likelihoods of "k-means" and the "DAEM-state." Furthermore, "DAEM-joint" exhibited the highest likelihood at $\alpha = 2^2$. These results show that structure annealing can yield reliable estimates of state sequences with the use of multiple decision trees.
Figure 5 shows the phoneme accuracy of each method. It is noted that only one decision tree (MDL) is used for decoding. Similar to the likelihood, the phoneme accuracy of “flat-start” was worse than those of the other methods because of the local maxima problem. It can also be seen that the methods using the DAEM algorithm outperformed “k-means,” even though phoneme boundary information was not used in the DAEM algorithm. Moreover, “DAEM-tree” and “DAEM-joint” had improved performance compared with the conventional “DAEM-state,” and an 11.1% relative error reduction was achieved for “DAEM-joint” over “k-means” at $C_11 = 20$. This result indicates that the reliable HMM parameters estimated using structure annealing are effective for improving the speech recognition performance.

4.1.3. Experimental results using multiple decision trees in decoding

For decoding using multiple decision trees, the models obtained at $i < 20$ were used and the performance was evaluated at each iteration. The schedule of temperature parameter $\alpha$ was updated by Eq. (20) with $\alpha = 2^0$. Figures 6 and 7 show the phoneme accuracy without and with an insertion penalty, respectively, while varying the temperature in decoding. (In these figures, the phoneme accuracy of “flat-start” and “k-means” is plotted every 10 EM-steps.) The results at $i = 20$ of “DAEM-tree” and “DAEM-joint” indicate the use of a single decision tree in decoding, because the likelihood of the MDL decision trees is much higher than the monophone decision tree. Therefore, $Q(m = 1) \geq 0$ and $Q(m = 2) \geq 1$ were obtained as the training result. It can be seen from Fig. 6 that decoding using multiple decision trees ($i < 20$) had improved accuracy after setting the appropriate temperature. “DAEM-state” was also evaluated at $i < 20$, but no improvement was obtained because only one decision tree was used. In Fig. 7, although the amount of improvement decreased upon using the insertion penalty, the decoding using multiple decision trees still improved the speech recognition performance compared with that at $i = 20$. These results indicate that not only in the training process but also in the decoding process, the use of multiple decision trees is effective for improving the performance of speech recognition.

4.2. Speaker-Independent Word Recognition Experiments

4.2.1. Experimental conditions

To train speaker-independent HMM sets, we used
37,618 sentences uttered by 122 male and 122 female speakers from Japanese Newspaper Article Sentences (JNAS) [12] as the training data. Two hundred sentences uttered by 23 male and 23 female speakers from JNAS were used for testing. The speech data was windowed at a 25 ms Hamming window, and parameterized into 13 mel-cepstral coefficients by the mel-cepstral analysis technique. Static coefficients including the zeroth coefficients and their first and second derivatives were used as feature parameters. Three-state left-to-right HMMs were used to model 43 Japanese phonemes, and 261 questions were prepared for decision tree clustering. Each state output probability distribution was modeled by a Gaussian distribution with a diagonal covariance matrix. In this experiment, Julius [13] was used as the decoder and a word forward 2-gram and a backward 3-gram were used as the language models.

The compared methods and the conditions for decision trees were the same as those in the speaker-dependent experiment. The number of EM-steps was set to 100 for the standard EM algorithm (“flat-start” and “k-means”). In the DAEM algorithm (“DAEM-state,” “DAEM-tree” and “DAEM-joint”), the number of temperature update steps was set to 20 (I = 20), and 5 EM-steps were conducted at each temperature. The word insertion penalty and the language weight were adjusted to yield the best performance for each method.

4.2.2. Experimental results

Figure 8 shows the log-likelihood of the training data. In the speaker-independent experiment, the estimation of acoustic models is more difficult than that in the speaker-dependent experiment. Therefore, the local maxima problem becomes more serious. However, the likelihood of the DAEM methods (“DAEM-state,” “DAEM-tree” and “DAEM-joint”) were higher than that of “k-means,” and “DAEM-joint” obtained the best value of the likelihood. This result indicates that not only using the DAEM algorithm but also using multiple decision trees is more effective for the local maxima problem in speaker-independent tasks.

Figure 9 shows the word accuracy of each method using a single decision tree (MDL) for decoding. It can be seen that the DAEM methods achieved higher accuracy than “flat-start” when the appropriate temperature schedule was used. Although “DAEM-joint” did not outperform “k-means,” the proposed method is still effective because “DAEM-joint” uses no phoneme boundary information.

Figure 10 illustrates the word accuracy of each method in which multiple decision trees are used for decoding. From the figure, it can be seen that “DAEM-tree” and “DAEM-joint” achieved higher word accuracy than “k-means.” This result suggests that even in the speaker-independent word recognition tasks, the proposed method can improve the performance of speech recognition.

5. CONCLUSIONS

In this paper, we proposed a speech recognition technique using multiple decision trees. In the proposed method, speech recognition was performed by ML estimation of the newly defined statistical model that includes multiple decision trees as a hidden variable. Applying the DAEM algorithm and using multiple decision trees in the

Fig. 8 Log-likelihood of training data (Speaker independent).

Fig. 9 Word accuracy for each temperature schedule.

Fig. 10 Word accuracy using multiple trees in decoding.
early stage of the training process, reliable state sequences can be obtained. In continuous phoneme recognition experiments, the proposed technique improved the performance of speech recognition even when using only two decision trees. As future work, we will consider the optimization of the temperature schedules, investigate the effect of increasing the number of decision trees, and develop an approach for preparing multiple decision trees.

ACKNOWLEDGEMENTS

The research leading to these results was partly funded from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement 213845 (the EMIME project), and the Strategic Information and Communications R&D Promotion Programme (SCOPE), Ministry of Internal Affairs and Communication, Japan.

REFERENCES


