PAPER *Special Section on Corpus-Based Speech Technologies*

Deterministic Annealing EM Algorithm in Acoustic Modeling for Speaker and Speech Recognition

Yohei ITAYA†a), ***Nonmember***, **Heiga ZEN**†b), ***Student Member***, **Yoshihiko NANKAKU**†c), **Chiyomi MIYAJIMA**††d), **Keiichi TOKUDA**†e), ***and*** **Tadashi KITAMURA**†f), ***Members***

SUMMARY This paper investigates the effectiveness of the DAEM (Deterministic Annealing EM) algorithm in acoustic modeling for speaker and speech recognition. Although the EM algorithm has been widely used to approximate the ML estimates, it has the problem of initialization dependence. To relax this problem, the DAEM algorithm has been proposed and confirmed the effectiveness in artificial small tasks. In this paper, we applied the DAEM algorithm to practical speech recognition tasks: speaker recognition based on GMMs and continuous speech recognition based on HMMs. Experimental results show that the DAEM algorithm can improve the recognition performance as compared to the standard EM algorithm with conventional initialization algorithms, especially in the flat start training for continuous speech recognition.

key words: *DAEM algorithm, acoustic modeling, EM algorithm, GMMs, HMMs*

1. Introduction

The EM (Expectation-Maximization) algorithm [1] is widely used for parameter estimation of statistical models with hidden variables. This algorithm provides a simple iterative procedure to obtain approximate ML (maximum likelihood) estimates. However, since the EM algorithm is a hill-climbing approach, it suffers from the local maxima problem, equivalently the initialization dependence problem.

On the other hand, GMMs (Gaussian mixture models) [2] and HMMs (hidden Markov models) [3] have been commonly used in acoustic modeling for speaker and speech recognition, respectively. In conventional approaches, the LBG algorithm for GMMs and the segmental k-means algorithm for HMMs have been employed to obtain initial model parameters before applying the EM algorithm. However, these initial values are not guaranteed to be near the true maximum likelihood point, and the posterior density becomes unreliable at an early stage of training. Especially in continuous speech recognition, it is difficult to obtain accurate phoneme boundaries for all training data. Hence, the embedded training has been used in which phoneme boundaries are also dealt as hidden variables, and estimated based on the EM algorithm. Furthermore, in the worse case where the boundary information is not available, a method called the flat start training is often applied. In this method, initial parameters of HMMs are given by making all states of all models equal, and then carry out the embedded training. In these situations, we do not have sufficient prior knowledge to obtain a good initial values for the EM algorithm, and it would converge to one of the local maxima or saddle points of the likelihood surface caused by a number of possible hidden state sequences.

To overcome the local maximum problem, several modified version of EM have been proposed. The simplest method is the EM algorithm starting from multiple initial values and select the model parameters which achieve the highest likelihood. However, it is difficult to determine how to generate initial values, and a large number of performing the EM algorithm for each initial value is required to avoid local maxima. To avoid this problem, the DAEM (Deterministic Annealing EM) algorithm [4] has been proposed. The algorithm is a variant of the EM algorithm based on DA (Deterministic Annealing) approach. The DA was first proposed for Vector Quantization (VQ) and clustering problem [5], later extended for various pattern classifies [6]. In the DAEM algorithm, the problem of maximizing the log-likelihood is reformulated as minimizing the thermodynamic free energy defined by the principle of maximum entropy and a statistic mechanics analogy. The posterior distribution derived in the DAEM algorithm includes a 'temperature' parameter which controls the influence of unreliable model parameters, and this annealing process can reduce the dependency on initial model parameters. The SMEM (Split and Merge EM) algorithm [7] has been proposed as a technique that possibly overcomes the DAEM algorithm. However the SMEM focuses on mixture distribution models and it cannot be easily applied to HMMs with temporal structure.

In this paper, the DAEM algorithm is applied to acoustic modeling for GMM-based speaker recognition and HMM-based continuous speech recognition. In GMM-based acoustic modeling, even though the SMEM algorithm should be compared with the DAEM algorithm, we focus on the DAEM algorithm in this paper. In addition to the inapplicability of the SMEM algorithm to HMMs, it is also a reason to use the DAEM algorithm that the implementation

Manuscript received July 8, 2004.
Manuscript revised September 19, 2004.

†The authors are with the Department of Computer Science and Engineering, Nagoya Institute of Technology, Nagoya-shi, 466–8555 Japan.

††The author is with the Department of Media Science, Nagoya University, Nagoya-shi, 466–8603 Japan.

a) E-mail: yoheir32@ics.nitech.ac.jp
b) E-mail: zen@ics.nitech.ac.jp
c) E-mail: nankaku@ics.nitech.ac.jp
d) E-mail: miyajima@is.nagoya-u.ac.jp
e) E-mail: tokuda@ics.nitech.ac.jp
f) E-mail: kitamura@ics.nitech.ac.jp

DOI: 10.1093/ietisy/e88–d.3.425

Copyright © 2005 The Institute of Electronics, Information and Communication Engineers

for GMMs and HMMs is performed by some changes of existing programming code which are commonly used in the speech recognition field.

In [6], the DA approach was applied to Minimum Classification Error (MCE) training for HMM-based speech recognizer and compared with ML and GPD training. However, the ML training was performed by the conventional EM algorithm without annealing and they were evaluated in a small spoken English letter recognition experiment. The DA algorithm for HMM design based on the Baum-Welch re-estimation was derived in [8]. However, the update of covariance matrices in the EM algorithm does not included in the annealing process. In this paper, the DAEM algorithm is summarized for deriving the posterior distribution of HMMs which has not been completely derived based on the DAEM algorithm yet, and evaluated on practical tasks of speech recognition.

This paper is organized as follows. In Sect. 2, we describe the DAEM algorithm, and apply it to the training of GMMs and HMMs. The Sect. 3 presents experimental results in speaker recognition and continuous speech recognition tasks. Concluding remarks and our plans for future works are described in the final section.

2. Deterministic Annealing EM Algorithm

In this section, the DAEM algorithm is rederived by means of the variational calculus similarly to "another interpretation of DAEM algorithm" in [4]. According to the derivation, we apply the algorithm to HMMs and show that the expectations of the derived posterior distribution can be calculated by the Forward-Backward algorithm as the standard HMMs.

2.1 EM Algorithm

The objective of the EM algorithm is to estimate a set of model parameters so as to maximize the incomplete log-likelihood function:

$$\mathcal{L}(\Lambda) = \log \int p\left(\boldsymbol{O}, \boldsymbol{q} | \Lambda\right) d\boldsymbol{q} \tag{1}$$

where $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T)$ and $\boldsymbol{q} = (q_1, q_2, \ldots, q_T)$ are observation vectors and hidden variables, respectively, and Λ denotes a set of model parameters. The procedure of the EM algorithm consists of maximizing at each iteration the auxiliary function so called Q-function:

$$Q(\Lambda, \Lambda') = \int p(\boldsymbol{q}|\boldsymbol{O}, \Lambda') \log p(\boldsymbol{O}, \boldsymbol{q}|\Lambda) d\boldsymbol{q} \tag{2}$$

where $p(\boldsymbol{q}|\boldsymbol{O}, \Lambda)$ is the posterior probability, and it can be computed by the Bayes rule:

$$p(\boldsymbol{q}|\boldsymbol{O}, \Lambda) = \frac{p(\boldsymbol{O}, \boldsymbol{q}|\Lambda)}{\displaystyle\int p(\boldsymbol{O}, \boldsymbol{q}'|\Lambda) d\boldsymbol{q}'}. \tag{3}$$

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

$$\begin{aligned} \text{(E step)}: &\quad \text{compute } Q(\Lambda, \Lambda^{(k)}) \\ \text{(M step)}: &\quad \Lambda^{(k+1)} = \arg\max_{\Lambda} Q(\Lambda, \Lambda^{(k)}) \end{aligned}$$

where k denotes the iteration number. This procedure is repeated until convergence of the likelihood.

2.2 Derivation of DAEM Algorithm

In the DAEM algorithm [4], the problem of maximizing the log-likelihood function is reformulated as the problem of minimizing a free energy function:

$$\mathcal{F}_\beta(\Lambda) = -\frac{1}{\beta} \log \int p(\boldsymbol{O}, \boldsymbol{q}|\Lambda)^\beta d\boldsymbol{q} \tag{4}$$

where $1/\beta$ $(0 \le \beta \le 1)$ called the "temperature", and if $\beta = 1$, the negative free energy $-\mathcal{F}_\beta(\Lambda)$ becomes equal to the log-likelihood function $\mathcal{L}(\Lambda)$. To solve this minimization problem, we introduce a new posterior distribution by using Jensen's inequality:

$$\begin{aligned} \mathcal{F}_\beta(\Lambda) &= -\frac{1}{\beta} \log \int f(\boldsymbol{q}|\boldsymbol{O}, \Lambda') \frac{p(\boldsymbol{O}, \boldsymbol{q}|\Lambda)^\beta}{f(\boldsymbol{q}|\boldsymbol{O}, \Lambda')} d\boldsymbol{q} \\ &\le -\frac{1}{\beta} \int f(\boldsymbol{q}|\boldsymbol{O}, \Lambda') \log \frac{p(\boldsymbol{O}, \boldsymbol{q}|\Lambda)^\beta}{f(\boldsymbol{q}|\boldsymbol{O}, \Lambda')} d\boldsymbol{q} \\ &= U_\beta(\Lambda, \Lambda') - \frac{1}{\beta} S_\beta(\Lambda') \end{aligned} \tag{5}$$

where the term $U_\beta(\Lambda, \Lambda')$ is the negative Q-function in which the posterior distribution $p(\boldsymbol{q}, \boldsymbol{O}|\Lambda)$ is replaced by the new function $f(\boldsymbol{q}|\boldsymbol{O}, \Lambda')$, and the term $S_\beta(\Lambda')$ is the entropy of f, i.e.,

$$U_\beta(\Lambda, \Lambda') = -\int f(\boldsymbol{q}|\boldsymbol{O}, \Lambda') \log p(\boldsymbol{O}, \boldsymbol{q}|\Lambda) d\boldsymbol{q} \tag{6}$$

$$S_\beta(\Lambda') = -\int f(\boldsymbol{q}|\boldsymbol{O}, \Lambda') \log f(\boldsymbol{q}|\boldsymbol{O}, \Lambda') d\boldsymbol{q}. \tag{7}$$

It can be seen that the upper bound in Eq. (5) corresponds to the Lagrangian in the principle of maximum entropy. In the deterministic annealing approach, the new posterior distribution f is derived so as to minimize the Lagrangian under the constraint of $\int f d\boldsymbol{q} = 1$. To solve this problem, we can use elementary calculus of variations to take functional derivatives of the upper bound with respect to f, and the optimal distribution can be derived as

$$f(\boldsymbol{q}|\boldsymbol{O}, \Lambda) = \frac{p(\boldsymbol{O}, \boldsymbol{q}|\Lambda)^\beta}{\displaystyle\int p(\boldsymbol{O}, \boldsymbol{q}'|\Lambda)^\beta d\boldsymbol{q}'.} \tag{8}$$

Substituting the derived posterior into Eq. (5), the upper bound agrees with the free energy, i.e.,

$$\mathcal{F}_\beta(\Lambda) = U_\beta(\Lambda, \Lambda') - \frac{1}{\beta} S_\beta(\Lambda'). \tag{9}$$

By inspection, it can be seen that $\mathcal{F}_\beta(\Lambda)$ has the same form as the free energy in statistical physics, and minimizing $\mathcal{F}_\beta(\Lambda)$ with a fixed temperature can be interpreted as the approach to thermodynamic equilibrium.

In the algorithm, the temperature is gradually decreased, and the posterior distribution is deterministically optimized at each temperature. The procedure of the DAEM algorithm is as follows:

```
1 Give an initial model, and set $\beta = \beta^{(0)}$
2 Iterate EM-steps with $\beta$ fixed until $\mathcal{F}_\beta$ converged:
    (E step)   : compute $U_\beta(\Lambda, \Lambda^{(k)})$
    (M step)   : $\Lambda^{(k+1)} = \arg\min_{\Lambda} U_\beta(\Lambda, \Lambda^{(k)})$
3 Increase $\beta$.
4 If $\beta > 1$, stop the procedure. Otherwise go to step 2.
```

where $1/\beta^{(0)}$ is an initial temperature, and should be chosen to a high enough value that the EM-steps can achieve a single global minimum of $\mathcal{F}_\beta$. At the initial temperature, the entropy $S_\beta(\Lambda')$ is intended to be maximized rather than $U_\beta(\Lambda, \Lambda')$, therefore the posterior f takes a form nearly uniform distribution. While the temperature is decreasing, the form of f changes from uniform to the original posterior, and at the final temperature $1/\beta = 1$ the DAEM algorithm agrees with the original EM algorithm. Similarly to the EM algorithm, the DAEM algorithm is also guaranteed to converge at a fixed temperature by decreasing $\mathcal{F}_\beta(\Lambda)$.

2.3 DAEM Algorithm for GMMs and HMMs

In the case of a GMM with M mixtures, the posterior probability of the m-th mixture for the DAEM algorithm is given by

$$f(q_t = m|\boldsymbol{o}_t, \Lambda) = \frac{p(m|\Lambda)^\beta p(\boldsymbol{o}_t|m, \Lambda)^\beta}{\sum_{m'=1}^{M} \left\{p(m'|\Lambda)^\beta \, p(\boldsymbol{o}_t|m', \Lambda)^\beta\right\}} \tag{10}$$

where $p(m|\Lambda)$ is the mixture weight, and $p(\boldsymbol{o}_t|m, \Lambda) = \mathcal{N}(\boldsymbol{o}_t|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ denotes a Gaussian distribution with the mean vector $\boldsymbol{\mu}_m$ and covariance matrix $\boldsymbol{\Sigma}_m$.

In the case of an HMM, the posterior distribution can be calculated by the forward-backward algorithm. The posterior function f of a state sequence $\boldsymbol{q}$ can be written by

$$\begin{aligned} f(\boldsymbol{q}|\boldsymbol{O}, \Lambda) &= \frac{p(\boldsymbol{q}|\Lambda)^\beta p(\boldsymbol{O}|\boldsymbol{q}, \Lambda)^\beta}{\sum_{\boldsymbol{q}'} \left\{p(\boldsymbol{q}'|\Lambda)^\beta p(\boldsymbol{O}|\boldsymbol{q}', \Lambda)^\beta\right\}} \\ &= \frac{\prod_{t=1}^{T} p(q_t|q_{t-1}, \Lambda)^\beta \prod_{t=1}^{T} p(\boldsymbol{o}_t|q_t, \Lambda)^\beta}{\sum_{\boldsymbol{q}'} \left\{\prod_{t=1}^{T} p(q'_t|q'_{t-1}, \Lambda)^\beta \prod_{t=1}^{T} p(\boldsymbol{o}_t|q'_t, \Lambda)^\beta\right\}} \end{aligned} \tag{11}$$

where $p(q_t|q_{t-1}, \Lambda)$ and $p(\boldsymbol{o}_t|q_t, \Lambda)$ indicate transition probability and state output probability, respectively. Note that initial state probability are denoted by $p(q_1|q_0, \Lambda)$. The expectations with respect to this distribution can also be calculated by the forward-backward algorithm with using $p(q_t|q_{t-1}, \Lambda)^\beta$ and $p(\boldsymbol{o}_t|q_t, \Lambda)^\beta$ as the transition probability and the observation probability respectively.

3. Experiments

To evaluate the performance of the DAEM algorithm, text-independent speaker recognition and continuous speech recognition experiments were conducted.

3.1 GMM-Based Speaker Recognition

For speaker recognition experiments, we used the ATR Japanese speech database c-set composed of 80 speakers. Each speaker consists of three sets of words: the first set of 216 words were used for training, and the second set of 260 words were used as development data for determining the number of EM-steps at each temperature and the total number of updating temperature which is denoted by I, and the third set of 260 words is testing set. The speech data was down-sampled from 20 kHz to 10 kHz, and then windowed at a 10-ms frame rate using a 25-ms Blackman window. The 12 mel-cepstral coefficients excluding zero-th coefficients were used as the feature vectors. Each speaker was modeled by one GMM with 4, 8, 16, 32 and 64 mixture components with diagonal covariance matrices.

In this experiment, we compared the following three initialization methods:

- "**random-EM**" : Mixture weights were set equal between all mixtures, and mean vectors of Gaussian components were generated from a normal distribution with mean = 0.0 and variance = 1.0. Diagonal elements of covariance matrices were given by taking the absolute of the generated values from the same distribution. We generated 5 sets of initial values and the model parameters which achieved the highest likelihood was selected.
- "**LBG-EM**" : The initial values of mixture components were computed from each cluster obtained by the LBG algorithm [9]. The mixture weights were given by the proportional values as the number of training data. The codewords were used as the mean vectors, and the diagonal covariances were computed from a set of training data belonging to each centroid.
- "**DAEM**" : The DAEM algorithm was applied. A schedule of decreasing the temperature in the DAEM algorithm should be proceed as slow as possible, particularly at early stage of training. From the result of preliminary experiments, the way of updating temperature parameter β was set to $\beta^{(i)} = \sqrt{i/I}$, $i = 1, 2, \ldots, I$ in this experiment, where $\beta^{(i)}$ is the value of β at i-th iteration, and I is the total number of the iterations. Figure 1 shows the update function of β.

To determine the number of updating β and the number of EM-steps at each temperature for "DAEM", we conducted preliminary experiments. Figure 2 shows the identification error rates while varying the number of update β among **I=** 1, 5, 10, 15, 20 and 30. The number of EM-steps at each temperature was fixed 20. From the figure,

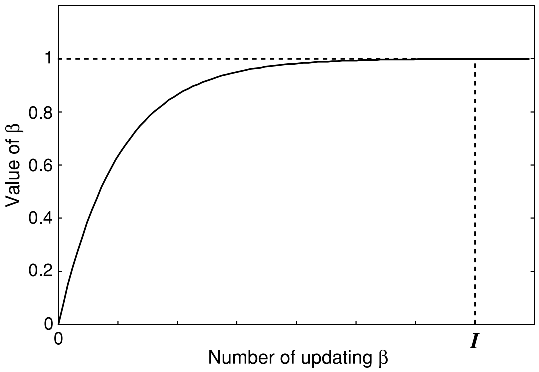

Fig. 1 Update function of β.

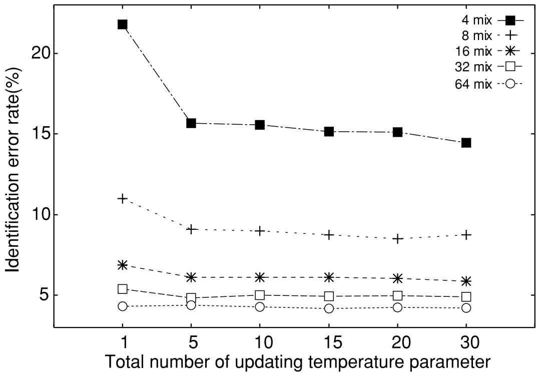

Fig. 2 Comparison of the number of updating temperature parameter.

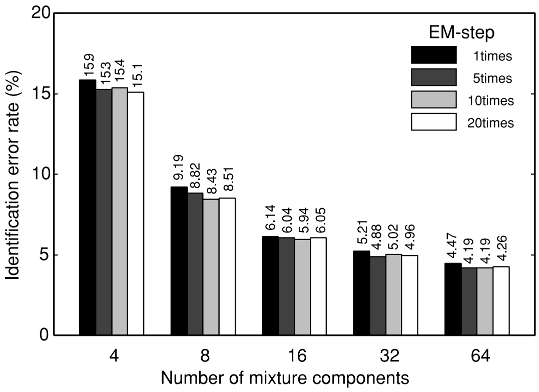

Fig. 3 Comparison of the total number of EM-steps at each temperature.

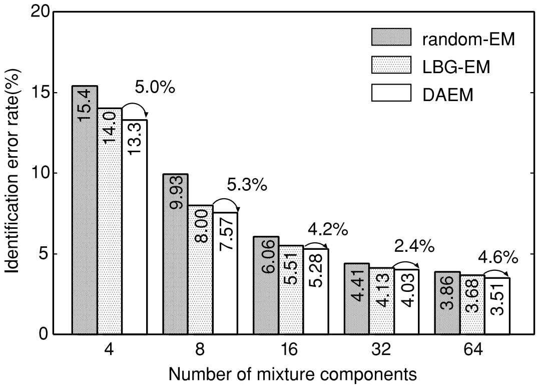

Fig. 4 Results of GMM-based speaker recognition.

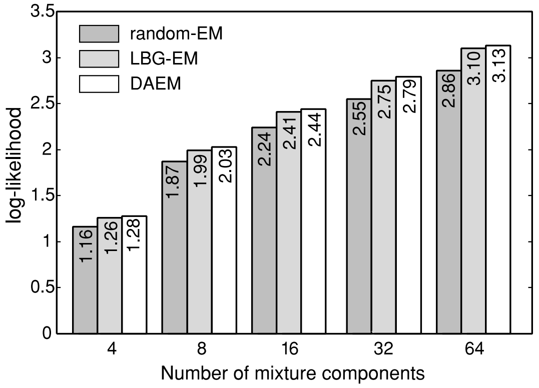

Fig. 5 Average log-likelihood of GMM-based speaker recognition.

the error rates became small when the temperature was decreased slowly. However, no significant difference of error rate was found between 20 and 30 iterations. Figure 3 shows the comparison of the total number of EM-step at each temperature with $I = 20$. At each temperature parameter, EM-step was repeated 1, 5, 10 and 20. It can be seen that 5 and 10 steps achieved good results. Although the temperature parameter should be decreased as slowly as possible, increasing the number of EM-steps at each temperature can be correspond that the change of the temperature becomes relatively large and this might lead to increasing the error rate at 20 EM-steps. From these results, we decided that 20 iterations at updating β and 10 EM-steps at each temperature were used for following experiments. In the both cases of "LBG-EM" and "random-EM", the number of iterations for the EM algorithm was limited by 100 due to the computational cost. Although "DAEM" carried out 200 EM-steps in total, "LBG-EM" and "random-EM" were almost converged within 100 iterations, and a further improvement could not be obtained by taking more than 100 iterations.

Figure 4 shows the results of the GMM-based speaker recognition experiments. It can be seen that the error rates of "random-EM" become higher than the other two cases, and this means that the EM algorithm suffers from the local maxima problem. However, in the case of "LBG-EM", since the initial Gaussian distributions were arranged according to training data, a better final point was achieved, and the error rates were reduced as compared with the results of "random-EM". Moreover, it was confirmed that further improvements were obtained by "DAEM" than "LBG-EM" in the all mixture cases, and the error reduction of 5.3% was obtained in the 8 mixture case. Figure 5 shows the average log-likelihood of training data in the speaker recognition experiment. "DAEM" achieved higher log-likelihood than that of "LBG-EM" in all mixture cases. These results indicate that the DAEM algorithm is effective to relax the problem of initialization dependence for GMM-based speaker recognition. Furthermore, although the training of "LBG-EM" consists of two processes, i.e., the LBG and the EM algorithm, the DAEM algorithm includes the initialization of model parameters.

3.2 HMM-Based Continuous Speech Recognition

To evaluate the performance of the DAEM algorithm for the training of HMMs, speaker- dependent and independent continuous phoneme recognition experiments were conducted. For the speaker-dependent experiment, we used phonetically balanced 503 sentences from the ATR Japanese speech database b-set. 450 sentences from the data set were used for training HMMs, and remaining 53 sentences were used for testing. Speaker-dependent HMMs of four males were constructed and the average results of each speaker experiment were presented. For the speaker-independent experiment, the ASJ-PB database (phonetically balanced) and the ASJ-JNAS database (Japanese newspaper article sentences speech corpus) were used. Gender-dependent monophone and triphone HMMs [10] with 1, 2, 4, and 8 Gaussian mixtures were trained using about 20,000 utterances spoken by 130 speakers, and the IPA-98-Testset (100 sentences) was used for testing.

In this experiment, we compared the following three training procedures:

- “**k-means**” : Using phoneme boundary labels, the segmental k-means algorithm, and the re-estimation based on the EM-algorithm were used for each phoneme HMM. Then, 10 iterations of the embedded training were also conducted.
- “**flat-start**” : The flat start training was performed. Initial parameters of monophone and triphone HMMs are given by making all states of all models equal, and then carry out the embedded training.
- “**DAEM**” : The DAEM algorithm was applied to the embedded training. The value of β was increased in the same manner as GMM (with $I = 10$), and 5 iterations of the EM-steps at each temperature, in total 50 EM-steps were conducted.

The DAEM algorithm with $\beta = 0$ is equivalent to the initial values of the flat start training, i.e., the posterior probabilities of the state sequences have an uniform distribution. However, even though the flat start training updates the model parameters immediately at the first iteration based on unreliable initial parameters (this corresponds the DAEM with $\beta = 0$ at the 1st iteration and $\beta = 1$ at the 2nd iteration), the DAEM algorithm gradually increase the parameter β, and updates the model parameters slowly based on the annealing process.

In practical situations, the performance of triphone HMMs can be improved from “flat-start” by using monophone HMMs as initial parameters. However, since the DAEM algorithm itself is assumed not to have any prior information about hidden variables (this means that an initial point of both “flat-start” and “DAEM” is equal), the experiment of flat-start training is necessary to evaluate the performance of the DAEM algorithm in the embedded training. In preliminary experiments, the results of triphone HMMs initialized by “flat-start” monophone HMMs are similar or worse than “k-means”. Even if the triphone HMMs trained from monophone HMMs achieves the similar performance with “DAEM”, the DAEM algorithm still has an advantage that the algorithm can be performed a simple procedure including the model parameter initialization. Furthermore, starting from non-zero β, the DAEM algorithm can also utilize initial parameters obtained from monophone HMMs. In this case, the degree of influence from an initial model is determined by the temperature parameter at the first iteration. However, we focus on the evaluation of the DAEM algorithm on the simple condition that monophone HMMs are not used as initial parameters for both “flat-start” and “DAEM”.

Figures 6 and 7 show the results of monophone HMMs in the speaker-dependent and speaker-independent experiments, respectively. Comparing the results of “k-means” and “flat-start”, it can be seen that the performance of “flat-start” was worse than “k-means” in the both figures. This is because “k-means” uses the phoneme boundary information as prior knowledge, and “flat-start” could not estimate the phoneme boundaries accurately due to the local maxima problem in the EM algorithm. Although the DAEM algorithm also did not use the phoneme labels, “DAEM”

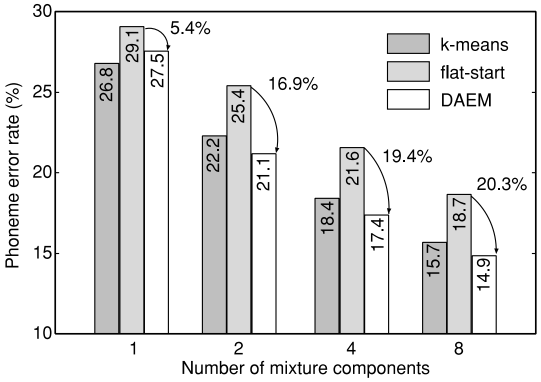

Fig. 6 Results of monophone HMMs (speaker-dependent).

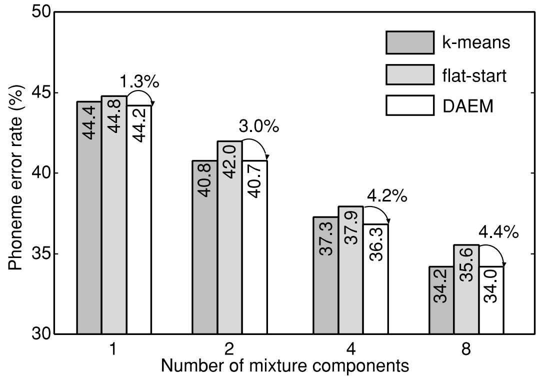

Fig. 7 Results of monophone HMMs (speaker-independent).

improved the recognition performance significantly than the results of "flat-start" in all the cases. The error reduction of 20.3% (8-mixtures) in the speaker-dependent and 4.4% (8-mixtures) in the speaker-independent experiment were obtained. Furthermore, it was confirmed that almost the same recognition rates as "k-means" were achieved by the DAEM algorithm. These results indicate that the influence of initial values was relaxed by the DAEM algorithm in continuous speech recognition.

Figures 8 and 9 show the results of triphone HMMs in the speaker- dependent and independent experiments, respectively. The parameter sharing (state tying) was performed by the decision tree based context clustering, and the number of states was determined by the MDL criterion [11]. Table 1 shows the total number of states in triphone HMMs with single mixture after the context clustering. The number of states of "k-means" become almost half of the other two methods. This means that "k-means" performs an efficient modeling by using the phoneme labels. On the contrary, since the triphone HMMs of "flat-start" and "DAEM" were trained independently of each other without labels, there is an inconsistency of temporal segmentations of acoustic features between states to be shared in the clustering. Even though the similar number of states was obtained between "flat-start" and "DAEM", in the Figure 8, "DAEM" improves the performance than "flat-start" in all cases, and the error reduction of 17.9% was obtained in the speaker-dependent experiment of 2-mixtures. In the speaker independent experiment, "DAEM" could not achieve the same error rates as "k-means", because variations in speaker characteristics affect estimating the phoneme boundaries. However, "DAEM" shows better results than "flat-start", although the both methods start from the same initial model parameters. These results show that the DAEM algorithm provides a simple procedure including the initialization even though we need to determine a proper schedule of decreasing the temperature, and it can improve the performance of the flat start training in HMM-based continuous speech recognition.

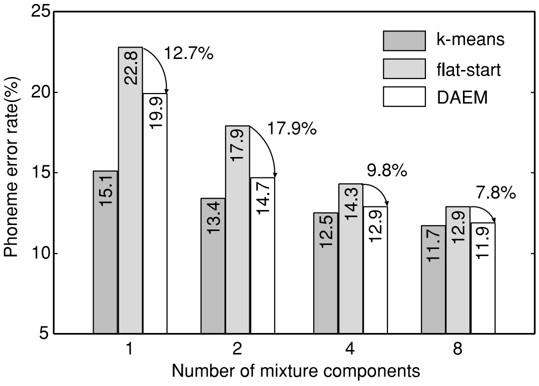

Fig. 8 Results of triphone HMMs (speaker-dependent).

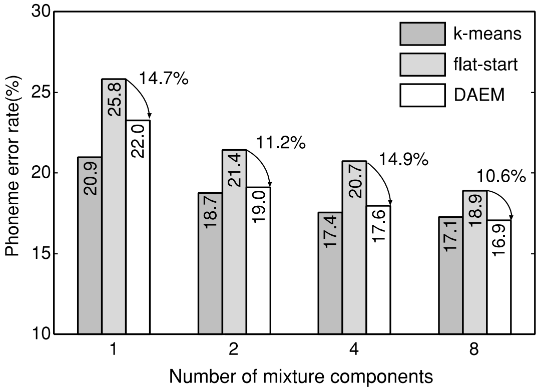

Fig. 9 Results of triphone HMMs (speaker-independent).

Table 1 Total number of distributions (triphone).

	k-means	flat-start	DAEM
speaker-dependent	1425	3244	3283
speaker-independent	7784	16776	14788

4. Conclusion

In this paper, we investigated the effectiveness of the DAEM algorithm in speaker recognition and continuous speech recognition. The DAEM algorithm is a reformulated version of the EM algorithm derived by minimizing the thermodynamic free energy, and can relax the problem of initialization dependence in the EM algorithm. The experimental results show that the DAEM algorithm is effective for acoustic modeling based on GMMs and HMMs, especially in the flat start training of HMM-based continuous speech recognition.

As a future work, we will also carry out the experiments with various update schemes of temperature parameter in the DAEM algorithm. More practical experiments are also future works, e.g., speaker recognition tasks with interval changes and large vocabulary speech recognition. Furthermore, we will investigate the relation among local maxima, amount of training data and model complexity.

References

[1] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. Roy. Statist. Soc., vol.39, pp.1–38, 1977.

[2] D.A. Reynolds and R.C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," IEEE Trans. Speech Audio Process., vol.3, no.1, pp.72–83, Jan. 1995.

[3] X.D. Huang, Y. Ariki, and M.A. Jack, Hidden Markov models for speech recognition, pp.119–125, Edinburgh University, 1990.

[4] N. Ueda and R. Nakano, "Deterministic annealing EM algorithm," Neural Netw., vol.11, pp.271–282, 1998.

[5] K. Rose, E. Gurewitz, and G.C. Fox, "Vector quantization by deterministic annealing," IEEE Trans. Inf. Theory, vol.38, no.4, pp.1249–1258, 1992.

[6] A.V. Rao and K. Rose, "Deterministically annealed design of hidden Markov model speech recognizers," IEEE Trans. Speech Audio Process., vol.9, no.2, pp.111–126, Feb. 2001.

[7] N. Ueda and R. Nakano, "EM algorithm with split and merge operations for mixture models," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J82-D-II, no.5, pp.930–940, May 1999.

[8] D. Miller, K. Rose, and P.A. Chou, "Deterministic annealing for trel-

lis quantizer and HMM design using Baum-Welch re-estimation," Proc. Int. Conf. Acoustic Speech Signal Processing, vol.5, pp.261–264, 1994.

[9] C. Miyajima, Discriminative training for system module integration in speaker and speech recognition, Doctoral Dissertation, Nagoya Institute of Technology, Jan. 2001.

[10] J.J. Odell, The use of context in large vocabulary speech recognition, PhD dissertation, Cambridge University, 1995.

[11] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," J. Acoust. Soc. Jpn. (E), vol.21, no.2, pp.79–86, 2000.

Yohei Itaya was born in 1980. He recieved the B.S. degree in Computer Engineering from the Nagoya Institute of technology, Nagoya, Japan in 2002. He is a master candidate of the Nagoya Institute of technology. His research interests include speaker and speech recognition. He is a student member of the Acoustical Society of Japan.

Heiga Zen received the A.E. degree in electronic and information engineering from Suzuka National College of Technology, Suzuka, Japan, and B.E. degree in computer science and M.E. degree in electrical and computer engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1999, 2001 and 2003, respectively. Currently he is a Ph.D student of the Department of Computer Science at the Nagoya Institute of Technology. His research interests include automatic speech recognition and text-to-speech synthesis. He is a member of ASJ.

Yoshihiko Nankaku received the B.E. degree in Computer Science, and the M.E. and Dr.Eng. degrees in the Department of Electrical and Electronic Engineering from the Nagoya Institute of Technology, Nagoya Japan, in 1999, 2001, and 2004 respectively. He is currently a postdoctoral fellow at the Nagoya Institute of Technology. His research interests include statistical machine learning, image recognition, speech recognition and synthesis and multimodal interface. He is a member of ASJ.

Chiyomi Miyajima received the B.E. degree in computer science and M.E. and Dr.Eng. degrees in electrical and computer engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1996, 1998, and 2001, respectively. From 2001 to 2003, she was a Research Associate of the Department of Computer Science, Nagoya Institute of Technology. Currently she is a Research Associate of the Graduate School of Information Science, Nagoya University, Nagoya, Japan. Her research interests include automatic speaker recognition and multi-modal speech processing. She received the Awaya Award in 2000 from the Acoustical Society of Japan. She is a member of ASJ and JASL.

Keiichi Tokuda received the B.E. degree in electrical and electronic engineering from the Nagoya Institute of Technology, Nagoya, Japan, the M.E. and Dr.Eng. degrees in information processing from the Tokyo Institute of Technology, Tokyo, Japan, in 1984, 1986, and 1989, respectively. From 1989 to 1996 he was a Research Associate at the Department of Electronic and Electric Engineering, Tokyo Institute of Technology. From 1996 to 2004 he was a Associate Professor at the Department of Computer Science, Nagoya Institute of Technology as Associate Professor, and now he is a Professor at the same institute. He is also an Invited Researcher at ATR Spoken Language Translation Research Laboratories, Japan and was a Visiting Researcher at Carnegie Mellon University from 2001 to 2002. He is a co-recipient of the Paper Award and the Inose Award both from the IEICE in 2001, and the TELECOM System Technology Prize from the Telecommunications Advancement Foundation Award, Japan, in 2001. He was a member of the Speech Technical Committee of the IEEE Signal Processing Society. His research interests include speech speech coding, speech synthesis and recognition, and statistical machine learning.

Tadashi Kitamura received the B.E. degree in electronics engineering from Nagoya Institute of Technology, Nagoya, in 1973, and M.E. and Dr.Eng. degrees from Tokyo Institute of Technology, Tokyo, in 1975 and 1978, respectively. In 1978 He joined the Research Laboratory of Precision Machinery and Electronics, Tokyo Institute of Technology, as a Research Associate. In 1983 he joined Nagoya Institute of Technology, as a Assistant Professor. He is currently a Professor of Graduate School of Engineering of Nagoya Institute of Technology. His current interests include speech processing, image processing and multi-modal biometrics. He is a member of IEEE, ISCA, ASJ, IPSJ and ITE.