DETERMINISTIC ANNEALING EM ALGORITHM IN PARAMETER ESTIMATION FOR ACOUSTIC MODEL

Y. Itaya[†], *H. Zen*[†], *Y. Nankaku*[†], *C. Miyajima*[‡], *K. Tokuda*[†], *and T. Kitamura*[†]

† Department of Computer Science and Engineering, Graduate School of Engineering,
Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan

‡ Department of Media Science, Graduate School of Information Science,
Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan

ABSTRACT

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 small tasks. In this paper, we applied the DAEM algorithm to 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 ordinary EM algorithm with conventional initialization methods, especially in the flat start training for continuous speech recognition.

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.

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 that 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 enough 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 this problem, the DAEM (Deterministic Annealing EM) algorithm [4] has been proposed. 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. In this paper, we apply the DAEM algorithm to GMMs and HMMs, and investigate the effectiveness of the DAEM algorithm in speaker recognition and continuous speech recognition.

This paper is organized as follows. In section 2, we describe the DAEM algorithm, and apply it to the training of GMMs and HMMs. The section 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

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:

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

where $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_T)$ and $\boldsymbol{q} = (q_1, q_2, ..., q_T)$ 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 probabilities, and it can be computed by the Bayes rule:

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

The log-likelihood for the training data is guaranteed to increase by increasing the value of the Q-function, that is, $Q(\Lambda, \Lambda') \geq$

This work was partially supported by a Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science, Encouragement of Young Scientists (B) (Grant No. 14780274).

$Q(\Lambda, \Lambda) \Rightarrow \mathcal{L}(\boldsymbol{O} \mid \Lambda') \geq \mathcal{L}(\boldsymbol{O} \mid \Lambda)$. Hence the maximization of the Q-function value at each iteration maximizes the likelihood for the training data. The EM algorithm starts with an initial model parameters $\Lambda^{(0)}$, and iterates between the following two steps:

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

where k denotes the iteration number. In this procedure, each step increase the value of $\mathcal{Q}$-function, hence the likelihood of the training data is also guaranteed to increase or leave it unchanged on each iteration.

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 $1/\beta$ 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:

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

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, and the parameter β is a Lagrange multiplier. 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

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

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 β fixed until $\mathcal{F}_\beta$ converged:
$(\text{E step}) \quad : \text{compute } U_\beta(\Lambda, \Lambda^{(k)})$
$(\text{M step}) \quad : \Lambda^{(k+1)} = \arg\min_{\Lambda} U_\beta(\Lambda, \Lambda^{(k)})$

3 Increase β.

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

$$f(q_t|\boldsymbol{o}_t, \Lambda) = \frac{w_m^\beta \mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)^\beta}{\sum_{m=1}^{M} \left\{ w_m^\beta \mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)^\beta \right\}} \tag{10}$$

where w_m is the mixture weight of the m-th mixture component, and $\mathcal{N}(\cdot \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ denotes a Gaussian distribution with the mean vector $\boldsymbol{\mu}_m$ and covariance matrix $\boldsymbol{\Sigma}_m$.

In the case of a 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

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

where $p(\boldsymbol{o}_t|q_t, \Lambda)$ and $p(q_t|q_{t-1}, \Lambda)$ indicate output probabilities and transition probabilities, respectively. The expectations with respect to this distribution can also be calculated by the forward-backward algorithm with using $p(\boldsymbol{o}_t|q_t, \Lambda)^\beta$ and $p(q_t|q_{t-1}, \Lambda)^\beta$ as the obserbation probabitities and the transition probabilities, 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 utters two sets of words: the first set of 216 words were used for training, and the second set of 520 words were used for testing. The speech data was down-sampled from 20kHz to 10kHz, 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.
- "**LBG-EM**" : The initial values of mixture components were computed from each cluster obtained by the LBG algorithm [5]. 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. In this experiment, the temperature parameter β was updated by $\beta^{(i)} = \sqrt{i/I}$. where $\beta^{(i)}$ is the value of β at i-th iteration, and I is the total number of the iterations. From preliminary experiments, we used $I = 20$, and 10 iterations of the EM-steps were conducted at each temperature.

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 1 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 resutls 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. 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, and it can be simply implemented by modifying the code for the ordinary EM algorithm.

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

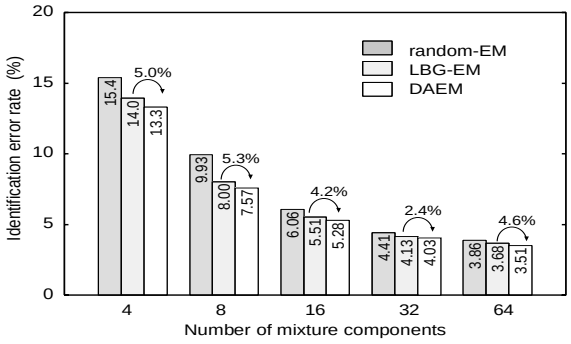

Fig. 1. Results of GMM-based speaker recognition.

phoneme recognition experiments were conducted. For the speaker-dependent experiment, we used phonetically balanced 503 sentences uttered by a male speaker MHT from the ATR Japanese speech database b-set. The 450 sentences were used for training HMMs, and the remaining 53 sentences were used for testing. 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[6] 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 by 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. Notice that the phoneme boundary labels are unavailable in "DAEM" and "flat-start", and they need to be estimated correctly in the training process in order to achieve the same performance as "k-means".

Figure 2 and 3 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 correctly due to the local maxima problem in the EM algorithm. Although the DAEM

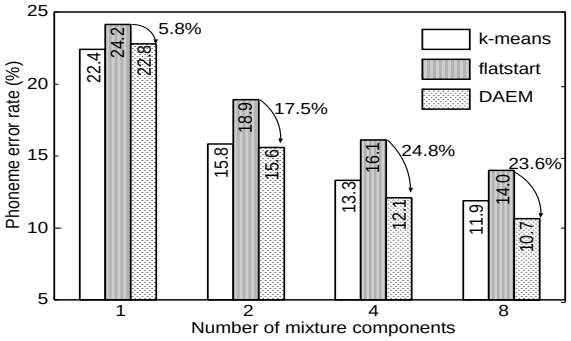

Fig. 2. Results of monophone HMMs (speaker-dependent)

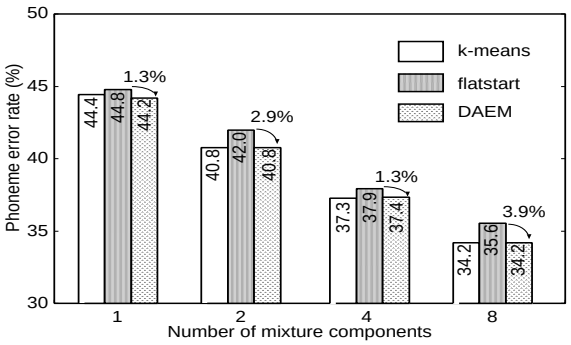

Fig. 3. Results of monophone HMMs (speaker-independent)

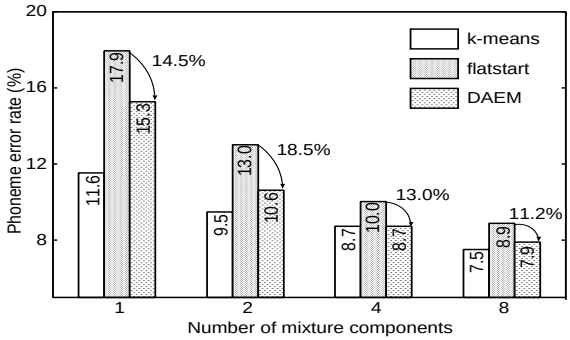

Fig. 4. Results of triphone HMMs (speaker-dependent)

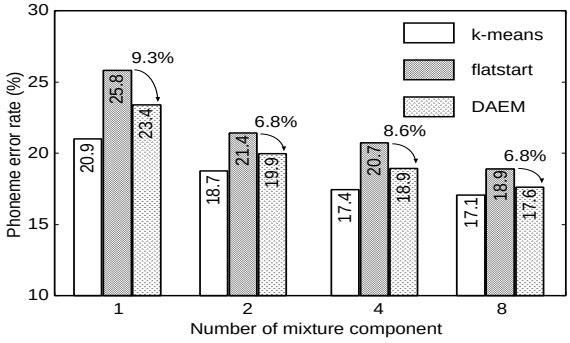

Fig. 5. Results of triphone HMMs (speaker-independent)

algorithm also did not use the phoneme labels, "DAEM" improves the recognition performance significantly than the results of "flat-start" in all the cases of mixtures. The error reduction of 24.8% (4-mixtures) in the speaker-dependent and 3.9% (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.

Figure 4 and 5 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. From the figures, it can be seen that the differences of the error rates between "k-means" and "flat-start" are decreased by increasing the number of mixtures. This means that the number of training data for each state of HMMs was decreased, and the dependency on the initial values was reduced. However, "DAEM" improves the performance than "flat-start" in the all mixture cases, and the error reduction of 18.5% 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 even though we need to determine a proper schedule of decreasing the temperature, the DAEM algorithm provides a simple procedure including the initialization, and it can improve the performance of the flat start training in HMM-based continuous speech recognition.

4. CONCLUSION

In this paper, we investigated the effectiveness of the DAEM algorithm in speaker recognition and continuous speech recognition. The DAEM algorithm is reformulated the EM algorithm as 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 investigate the relation between the amount of training data and the number of model parameters in the decision tree context clustering. We will also carry out the experiments with various update schemes of temperature parameter in the DAEM algorithm.

5. 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," *I*EEE Trans. on Speech and Audio Processing, 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," EDINBURGH UNIVERSITY, pp.119–125, 1990.

[4] N. Uedaand R. Nakano, "Deterministic Annealing EM Algorithm" Neural Networks, (11), pp.271–282, 1998.

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

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