
Pergamon

Neural Networks

Neural Networks 11 (1998) 271–282

Contributed article

Deterministic annealing EM algorithm

Naonori Ueda*, Ryohei Nakano

NITT Communication Science Laboratories, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan

Received 24 November 1996; accepted 9 October 1997

Abstract

This paper presents a deterministic annealing EM (DAEM) algorithm for maximum likelihood estimation problems to overcome a local maxima problem associated with the conventional EM algorithm. In our approach, a new posterior parameterized by 'temperature' is derived by using the principle of maximum entropy and is used for controlling the annealing process. In the DAEM algorithm, the EM process is reformulated as the problem of minimizing the thermodynamic free energy by using a statistical mechanics analogy. Since this minimization is deterministically performed at each temperature, the total search is executed far more efficiently than in the simulated annealing. Moreover, the derived DAEM algorithm, unlike the conventional EM algorithm, can obtain better estimates free of the initial parameter values. We also apply the DAEM algorithm to the training of probabilistic neural networks using mixture models to estimate the probability density and demonstrate the performance of the DAEM algorithm.

Keywords: Deterministic annealing; EM algorithm; Maximum likelihood estimation; Maximum entropy principle; Probabilistic neural networks

1. Introduction

The Expectation–Maximization (EM) algorithm (Dempster et al., 1977) is an iterative statistical technique for computing maximum likelihood (ML) estimates from incomplete data. It has generally been employed for a wide variety of parameter estimation problems. Recently, the EM algorithm has also been successfully employed as a learning algorithm for hierarchical mixtures of experts (Jordan and Jacob, 1994) and probabilistic neural networks (PNNs) (Traven, 1991; Streit and Luginbuhl, 1994). In addition, it has been found to have some relationship to the learning of Boltzmann machines (Byrne, 1992).

This algorithm has attractive features such as reliable global convergence, low cost per iteration, economy of storage, and ease of programming, but it is not free from problems in practice. The most serious problem associated with the algorithm is the local maxima problem. This problem makes the performance dependent on the initial parameter value. Indeed, the EM algorithm should be performed from as wide a choice of starting values as possible according to some ad hoc criterion (McLachlan and Basford, 1988). Although the problem is familiar to many, to our knowledge, little effort has been made so far to solve it.

To overcome this problem, we propose a *deterministic annealing* EM (DAEM) algorithm by using the *principle of maximum entropy* and the *statistical mechanics analogy*. In our approach, the ML estimation, that is, maximizing the *log-likelihood function* is reformulated as minimizing the *thermodynamic free energy*, defined as an effective cost function that depends on the *temperature*. Unlike the simulated annealing (Geman and Geman, 1984) where stochastic search is performed on the given energy surface, this cost function is *deterministically* optimized at each temperature.

Such deterministic annealing (DA) approaches have been proposed for vector quantization (VQ) and clustering problems (Rose et al., 1992). Some variants have been presented (Buhmann and Kuhnel, 1993; Wong, 1993) for clustering algorithms. For mixture density estimation problems, Yuille et al. (1994) have recently shown that the EM algorithm can be used in conjunction with DA. In our earlier paper, independent of Yuille's work, we presented a new EM algorithm with DA for the same mixture density estimation problems (Ueda and Nakano, 1994).[1]

* Requests for reprints should be sent to Dr N. Ueda. Tel.: 0081 774 95 1823; Fax: 0081 774 95 1839; E-mail: ueda@cslab.kecl.ntt.co.jp.

[1] In Yuille's paper, the EM as interpreted as one way to solve DA-based optimization problems. On the other hand, in our formulation, the EM process itself is reformulated by a DA approach so that the DA feature is incorporated into the EM algorithm.

0893–6080/98/$19.00
PII S0893-6080(97)00133-0

The aim of this paper is to generalize our earlier work and to derive a DA variant of the general EM algorithm. Since the EM algorithm can be used not only for mixture estimation problems but also for other parameter estimation problems based on the ML method, this generalization is expected to be of value in practice. Although the basic idea has already been presented in our previous paper (Ueda and Nakano, 1995b), we derive the DAEM algorithm more formally with its convergence theorem in this paper. We also show an application of the DAEM algorithm to the training of probabilistic neural networks (PNNs) to estimate the probability density.

This paper is organized as follows. After a brief review of the EM algorithm in Section 2, our deterministic annealing approach is introduced in Section 3. Section 3 also provides the convergence theorem of the DAEM algorithm, some implementation issues, and simulation results to intuitively explain how the DAEM algorithm works. In Section 4, the DAEM algorithm is applied to the training of PNNs for a mixture density estimation problem. Section 5 concludes this paper.

2. General theory of the EM algorithm

2.1. ML estimates from incomplete data

Suppose that a set χ: $\chi = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$, consists of 'observable data' $\chi_{\text{obs}} = \{\boldsymbol{x}_1^{\text{obs}}, \boldsymbol{x}_2^{\text{obs}}, \ldots, \boldsymbol{x}_N^{\text{obs}}\}$ and 'unobservable data' $\chi_{\text{mis}} = \{\boldsymbol{x}_1^{\text{mis}}, \boldsymbol{x}_2^{\text{mis}}, \ldots, \boldsymbol{x}_N^{\text{mis}}\}$. Here, $\boldsymbol{x}_k^{\text{mis}}$ is an unobservable data point corresponding to $\boldsymbol{x}_k^{\text{obs}}$. That is,

$$\chi = (\chi_{\text{obs}}, \chi_{\text{mis}}) = \{(\boldsymbol{x}_k^{\text{obs}}, \boldsymbol{x}_k^{\text{mis}}),\ k = 1, \ldots, N\}$$

χ and χ_{obs} are called complete data and incomplete data, respectively.[2]

Assume that the joint probability density of χ_{obs} and χ_{mis} is parametrically given as $p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta)$, where Θ denotes parameters of the density to be estimated and the dimensionality of Θ is finite and fixed. Provided the data points are i.i.d. sample points, the above distribution can be rewritten as

$$p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta) = \prod_{k=1}^{N} p(\boldsymbol{x}_k^{\text{obs}}, \boldsymbol{x}_k^{\text{mis}}; \Theta).$$

However, in order to make the notation simple, hereafter we use χ_{obs} and χ_{mis} rather than $\boldsymbol{x}_k^{\text{obs}}$ and $\boldsymbol{x}_k^{\text{mis}}$

The ML estimate of Θ is a value of Θ that maximizes the observed data (or incomplete data) log-likelihood function $L(\Theta; \chi_{\text{obs}})$ defined by

$$L(\Theta; \chi_{\text{obs}}) \stackrel{\text{def}}{=} \log p(\chi_{\text{obs}}; \Theta) = \log \int p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta)\, \mathrm{d}\chi_{\text{mis}}. \quad (1)$$

Or equivalently, the ML estimate corresponds to a solution of the following *likelihood equation*:

$$\partial L / \partial \Theta = 0. \quad (2)$$

In many applications, since Eq. (2) becomes nonlinear, a closed-form solution of Eq. (2) cannot be found and therefore some iterative methods should be applied. Although Newton or quasi-Newton algorithms are available for Eq. (2), the EM algorithm is well known as a computationally simpler algorithm for obtaining ML estimates (Dempster et al., 1977).

2.2. EM algorithm

The characteristic of the EM algorithm is to maximize the incomplete data log-likelihood function by iteratively maximizing the expectation of the following complete data log-likelihood function:

$$L_c(\Theta; \chi) = \log p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta). \quad (3)$$

Suppose that $\Theta^{(t)}$ denotes the estimate of Θ obtained after the tth iteration of the algorithm. Then, at the $t+1$th iteration, the E-step computes the expected complete data log-likelihood function denoted by $Q(\Theta|\Theta^{(t)})$, and the M-step finds the Θ maximizing $Q(\Theta|\Theta^{(t)})$. Specifically,

$$\mathbf{E-step}: Q(\Theta|\Theta^{(t)}) \stackrel{\text{def}}{=} \mathrm{E}\{L_c(\Theta; \chi)|\chi_{\text{obs}}; \Theta^{(t)}\} = \int \{\log p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta)\} \times p(\chi_{\text{mis}}|\chi_{\text{obs}}; \Theta^{(t)}) \mathrm{d}\chi_{\text{mis}}$$

$$\mathbf{M-step}: \Theta^{(t+1)} = \operatorname{argmax}_{\Theta} Q(\Theta|\Theta^{(t)})$$

Given an initial value $\Theta^{(0)}$, the EM steps are repeated cyclically until convergence. Here, $p(\chi_{\text{mis}}|\chi_{\text{obs}};\ \Theta^{(t)})$ is the posterior probability density and can be computed by the Bayes rule:

$$p(\chi_{\text{mis}}|\chi_{\text{obs}}; \Theta^{(t)}) = \frac{p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta^{(t)})}{\int p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta^{(t)})\, \mathrm{d}\chi_{\text{mis}}}. \quad (4)$$

The following theorem for the convergence of the EM steps shows the validity of the EM algorithm.

Theorem 1. (Dempster et al., 1977) *Every EM procedure increases $L(\Theta;\chi_{obs})$ at each iteration, that is, $L(\Theta^{(t+1)};\chi_{obs}) \geq L(\Theta^{(t)};\chi_{obs})$, with equality if and only if $Q(\Theta^{(t+1)}|\Theta^{(t)}) = Q(\Theta^{(t)})$.*

Proof. (Dempster et al., 1977) According to Theorem 1, eventually, $L(\Theta^{(t)})$ converges to a local maximum depending on the initial values $\Theta^{(0)}$. Namely, the performance of

[2] In such unsupervised learning as mixture problems, x_k^{mis} reduces to an integer value ($(x_k^{\text{mis}} \in \{1, 2, \ldots, C\}$, where C is the number of components), indicating the component from which an observed data point x_k^{obs} originates.

the EM algorithm highly depends on $\Theta^{(0)}$. In the next section, we will derive a new variant of the EM algorithm to avoid this potential local maxima.

3. Deterministic annealing approach

3.1. DAEM algorithm based on parameterized posterior

In the EM algorithm, the posterior density function plays an important role in the M-step. As mentioned in the previous section, the posterior calculated by Eq. (4) is unreliable at an early stage of the iteration. Thus, instead of Eq. (4), we introduce another posterior $f(\chi_{\text{mis}}|\chi_{\text{obs}})$. Since we do not have any prior knowledge about f, we apply the *principle of maximum entropy* to specify it. Suppose that when χ_{obs} are observed, the *micro state*, i.e., f is unknown, but the *macro information*, i.e., the expectation of the complete data log-likelihood L_c, that is,

$$E_f\{L_c(\Theta;\chi)|\chi_{\text{obs}}\} = \text{const.} \tag{5}$$

Then, f can be obtained as a distribution that maximizes the entropy

$$S = -\int \{\log f(\chi_{\text{mis}}|\chi_{\text{obs}})\} f(\chi_{\text{mis}}|\chi_{\text{obs}})\, \mathrm{d}\chi_{\text{mis}}, \tag{6}$$

under the constraints of Eq. (5) and $\int f \mathrm{d}\chi_{\text{mis}} = 1$. The maximum entropy principle can be interpreted as a method of designing a probabilistic model so that a weight is assigned as equally as possible[3] to each micro state so long as the '*macro information*' is satisfied.[4]

This maximization problem can be easily solved by the method of Lagrange multipliers. Let us now find the extreme of a functional $J[f]$:

$$J[f] \stackrel{\text{def}}{=} S + \beta(E_f\{L_c|\chi_{\text{obs}}\} - \text{const}) + \lambda\left(\int f\, \mathrm{d}\chi_{\text{mis}} - 1\right),$$

where β and λ are Lagrange multipliers. The variation of J, δJ, due to the variation of f, δf, is expressed by

$$\delta J = \int [-1 - \log f(\chi_{\text{mis}}|\chi_{\text{obs}}) + \beta \log p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta) + \lambda] \delta f\, \mathrm{d}\chi_{\text{mis}}.$$

Since $\delta J = 0$ regardless of δf, $[\cdot]$ in the above equation must be zero. Thus,

$$f(\chi_{\text{mis}}|\chi_{\text{obs}}) = \exp\{\beta \log p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta) + \lambda - 1\}. \tag{7}$$

From Eq. (7) and

$$\int f \mathrm{d}\chi_{\text{mis}} = 1$$

we have the following Gibbs distribution:

$$f(\chi_{\text{mis}}|\chi_{\text{obs}}) = \frac{1}{Z}\exp\{-\beta(-L_c(\Theta;\chi))\}. \tag{8}$$

Here, Z is called the *partition function* and is given by

$$Z = \int \exp\{-\beta(-L_c(\Theta;\chi))\}\, \mathrm{d}\chi_{\text{mis}} = \int p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta)^{\beta}\, \mathrm{d}\chi_{\text{mis}}. \tag{9}$$

The parameter β (the Lagrange multiplier) is determined by the value of U. By making an analogy to *annealing*, one can see that $1/\beta$ corresponds to the '*temperature*'. Substituting Eq. (9) into Eq. (8) gives us a new posterior parameterized by β,

$$f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta) = \frac{p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta)^{\beta}}{\int p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta)^{\beta}\, \mathrm{d}\chi_{\text{mis}}}. \tag{10}$$

Since f is expressed by using a p that depends on Θ, we write f as $f(\cdot;\Theta)$ in Eq. (10). There are two special cases; the first is $\beta = 0$ which yields a uniform distribution. The second is $\beta = 1$ where f reduces to the original posterior given by Eq. (4). For $0 < \beta < 1$, an increase of β means a change in the form of f from uniform to the original posterior.

Generally, since an initial value is not guaranteed to be near the true one (i.e., the posterior density is unreliable), the influence of this posterior at the EM-steps should be weakened at an early stage of training. Ideally, as the training proceeds, the effect should be strengthened. To achieve this type of training, by newly adding a β-loop (i.e., *annealing loop*) to the original EM-steps and replacing the posterior given by Eq. (4) with Eq. (10), the following deterministic annealing variant of the EM algorithm can be derived.

[DAEM algorithm]

1. Set $\beta \leftarrow \beta_{\min}(0 < \beta_{\min} \ll 1)$.
2. Set $\Theta^{(0)}$, and $t \leftarrow 0$.
3. Iterate the following EM-steps until convergence:

E − step : $U_\beta(\Theta|\Theta^{(t)})$

$$\stackrel{\text{def}}{=} E_{f^{(t)}}\{-\log p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta)|\chi_{\text{obs}}; \Theta^{(t)}\} = \int \{-\log p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta)\} \frac{p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta^{(t)})^{\beta}}{\int p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta^{(t)})^{\beta}\, \mathrm{d}\chi_{\text{mis}}} \mathrm{d}\chi_{\text{mis}}$$

M − step : $\Theta^{(t+1)}$

$$= \arg\min_{\Theta} U_\beta(\Theta|\Theta^{(t)})$$

Set $t \leftarrow t + 1$.

4. Increase β.
5. If $\beta < 1$, repeat from step 3; otherwise stop.

[3] For example, in the case of mixture models, 'equal weight' means the equal probability with which x_k^{obs} belongs to each component.

[4] Feder (1986) shows that the maximum entropy is a special case of the Minimum Description Length (MDL) criterion (Rissanen, 1978) for some coding scheme, and Hinton et al. (1995) argues that the maximum likelihood estimate can also be formulated based on the MDL criterion in a certain sigmoidal belief network. These results are quite interesting in that the maximum entropy principle can be well motivated by a MDL prior argument in the maximum likelihood estimation problems.

Note that the DAEM algorithm can be viewed as an iterative minimization of U_β at each temperature $(1/\beta)$. An important distinction to keep in mind is that unlike simulated annealing, the optimization in step 3 is *deterministically* performed at each β. This is the reason why we use the term 'deterministic'. Since when $\beta = 1$, $U_1 \equiv -Q$, at that time the DAEM algorithm agrees with the original EM algorithm. In other words, the EM-steps at $\beta = 1$ precisely try to find the ML estimate.

3.2. Theory of DAEM algorithm

In the previous section, we derived the DAEM algorithm in some ad hoc manner. In this section, we will show the theoretical validity of the DAEM algorithm.

Taking the logarithm on both sides of Eq. (10), we have

$$-\frac{1}{\beta}\log\int p(\chi_{\text{obs}},\chi_{\text{mis}};\Theta)^\beta\,\mathrm{d}\chi_{\text{mis}}$$

$$=-\log p(\chi_{\text{obs}},\chi_{\text{mis}};\Theta)+\frac{1}{\beta}\log f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta) \tag{11}$$

Moreover, taking the conditional expectation with respect to the distribution f, we obtain

$$-\frac{1}{\beta}\log\int p(\chi_{\text{obs}},\chi_{\text{mis}};\Theta)^\beta\,\mathrm{d}\chi_{\text{mis}}=U_\beta(\Theta)-\frac{1}{\beta}S_\beta(\Theta), \tag{12}$$

where

$$U_\beta(\Theta)=E_{f^{(t)}}\{-\log p(\chi_{\text{obs}},\chi_{\text{mis}};\Theta)|\chi_{\text{obs}}\}, \tag{13}$$

$$s_\beta(\Theta)E=_{f^{(t)}}\{-\log f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta)|\chi_{\text{obs}}\}. \tag{14}$$

On the other hand, once the partition function is obtained explicitly as in Eq. (9), by using the *statistical mechanics* analogy, we can define the *free energy* as an effective cost function that depends on the temperature:

$$F_\beta(\Theta)$$

$$\stackrel{\text{def}}{=}-\frac{1}{\beta}\log Z$$

$$=-\frac{1}{\beta}\log\int p(\chi_{\text{obs}},\chi_{\text{mis}};\Theta)^\beta\,\mathrm{d}\chi_{\text{mis}}. \tag{15}$$

Therefore, Eq. (12) can be written as

$$F_\beta(\Theta)=U_\beta(\Theta)-\frac{1}{\beta}S_\beta(\Theta). \tag{16}$$

Interestingly, regarding $U\beta$ in Eq. (16) as the *internal energy* (U_β is always positive), Eq. (16) exactly agrees with the expression of *free energy*: $\mathcal{F}=\mathcal{U}-\mathcal{TS}$ which is well known in statistical physics (i.e., $F_\beta \leftrightarrow \mathcal{F}$, $U_\beta \leftrightarrow \mathcal{U}$, $1/\beta \leftrightarrow T$, $S_\beta \leftrightarrow \mathcal{S}$, where T is the temperature and $\mathcal{S}$ is the entropy).

At equilibrium, it is well known that a thermodynamic system settles into a configuration that minimizes its free energy. Hence, we consider the minimization of F with a fixed temperature (β). To solve this, we derive an iterative algorithm. Suppose that $\Theta^{(t)}$ denotes the estimate of Θ obtained after the tth iteration. Then, taking the conditional expectation given χ_{obs} and $\Theta^{(t)}$ on both sides of Eq. (11), we have

$$F_\beta(\Theta)=U_\beta(\Theta|\Theta^{(t)})-\frac{1}{\beta}S_\beta(\Theta|\Theta^{(t)}), \tag{17}$$

where

$$U_\beta(\Theta|\Theta^{(t)})$$

$$=\mathrm{E}_{f^{(t)}}\{-\log p(\chi_{\text{obs}},\chi_{\text{mis}};\Theta)|\chi_{\text{obs}};\Theta^{(t)}\}$$

$$=\int\{-\log p(\chi_{\text{obs}},\chi_{\text{mis}};\Theta)\}f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta^{(t)})\,\mathrm{d}\chi_{\text{mis}},$$

$$S_\beta(\Theta|\Theta^{(t)})$$

$$=\mathrm{E}_{f^{(t)}}\{-\log f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta)|\chi_{\text{obs}};\Theta^{(t)}\}$$

$$\int\{-\log f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta)\}f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta^{(t)})\,\mathrm{d}\chi_{\text{mis}}.$$

Then we can prove the following theorem.

Theorem 2. *If given $\Theta^{(t)}$, we determine $\Theta^{(t+1)}$ as a value of Θ that minimizes $U_\beta(\Theta|\Theta^{(t)})$, then $F_\beta(\Theta^{(t+1)}) \le F_\beta(\Theta^{(t)})$, where equality holds if and only if both $U_\beta(\Theta^{(t+1)}|\Theta^{(t)}) = U_\beta(\Theta^{(t)}|\Theta^{(t)})$ and $f(\chi_{mis}|\chi_{obs};\Theta^{(t+1)}) = f(\chi_{mis}|\chi_{obs};\Theta^{(t)})$.*

Proof. Let

$$\begin{aligned}\Delta F &= F_\beta(\Theta^{(t+1)})-F_\beta(\Theta^{(t)})\\ &=\left(U_\beta(\Theta^{(t+1)}|\Theta^{(t)})-U_\beta(\Theta^{(t)}|\Theta^{(t)})\right)\\ &\quad+\frac{1}{\beta}\left(S_\beta(\Theta^{(t)}|\Theta^{(t)})-S_\beta(\Theta^{(t+1)}|\Theta^{(t)})\right)\end{aligned} \tag{18}$$

Then, the second term on the RHS of Eq. (18) is

$$S_\beta(\Theta^{(t)}|\Theta^{(t)})-S_\beta(\Theta^{(t+1)}|\Theta^{(t)})$$

$$=\int\log\left\{\frac{f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta^{(t+1)})}{f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta^{(t)})}\right\}f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta^{(t)})\,\mathrm{d}\chi_{\text{mis}}$$

$$=\mathrm{E}_{f^{(t)}}\left\{\log\frac{f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta^{(t+1)})}{f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta^{(t)})}\middle|\chi_{\text{obs}};\Theta^{(t)}\right\}$$

$$\le\log\mathrm{E}_{f^{(t)}}\left\{\frac{f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta^{(t+1)})}{f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta^{(t)})}\middle|\chi_{\text{obs}},\Theta^{(t)}\right\}$$

$$=\log\int f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta^{(t+1)})\,\mathrm{d}\chi_{\text{mis}}=\log 1=0. \tag{19}$$

Hence $S_\beta(\Theta^{(t)}|\Theta^{(t)}) \le S_\beta(\Theta^{(t+1)}|\Theta^{(t)})$ holds. The inequality is strict unless $f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta^{(t+1)}) = f(\chi_{\text{mis}}|\chi_{\text{obs}};\Theta^{(t)})$. Note that the inequality in Eq. (19) is due to Jensen's inequality (i.e., $\mathrm{E}\{\log Y\} \le \log \mathrm{E}\{Y\}$ with equality if and only if $Y = 1$).

Therefore, when $\beta > 0$, the second term on the RHS of Eq. (18) is zero or negative.

Consequently, if we set $\Theta^{(t+1)} = \arg\min_\Theta U_\beta(\Theta|\Theta^{(t)})$, then $U_\beta(\Theta^{(t+1)}|\Theta^{(t)}) \leq U_\beta(\Theta^{(t)}|\Theta^{(t)})$ and therefore $\Delta F \leq 0$. The equality holds when $U_\beta(\Theta^{(t+1)}|\Theta^{(t)}) = U_\beta(\Theta^{(t)}|\Theta^{(t)})$ and $S_\beta(\Theta^{(t)}|\Theta^{(t)}) = S_\beta(\Theta^{(t)}|\Theta^{(t)})$, that is, $f(\chi_{\text{mis}}|\chi_{\text{obs}}; \Theta^{(t+1)}) = f(\chi_{\text{mis}}|\chi_{\text{obs}}; \Theta^{(t)})$.

Comparing Eq. (1) with Eq. (15) carefully, one can see that when $\beta = 1$, the free energy agrees with the negative incomplete data log-likelihood, i.e., $F_1(\Theta) \equiv -L(\Theta; \chi_{\text{obs}})$. It follows that the Θ that minimizes $F_1(\Theta)$ is exactly equivalent to the ML estimate that maximizes the incomplete data log-likelihood $L(\Theta;\chi_{\text{obs}})$. In other words, it can be interpreted that the ML estimation, i.e., the problem of maximizing the incomplete data log-likelihood function, is reformulated as the problem of minimizing the free energy. Since the free energy depends on β, by adding another β-loop to the original EM-steps as the annealing process, we arrive at the DAEM algorithm. Clearly, the global convergence of the EM steps in the DAEM algorithm is theoretically guaranteed by Theorem 2.

3.3. Search strategy of the DAEM algorithm

Now let us explain how the DAEM algorithm converges to a suboptimal solution free from the initial parameter values.

As one can see from Eq. (15), the temperature parameter, β, *smooths* the free energy. The rate of smoothing increases with increasing β. For a small enough β, $F\beta(\Theta)$ has only one global minimum and the minimum can easily be found by the EM-steps. Then, by gradually increasing β (decreasing the temperature), the effect of f is gradually strengthened and consequently several local minima of $F_\beta(\Theta)$ appear. Assuming the shape of $F\beta(\Theta)$ gradually changes as β gradually changes, it can be thought that at each step of β, the new global minimum is close to the previous one. Therefore, under the assumption, by executing the EM-steps with the previously obtained minimum as a new initial value, the algorithm can track the new global minimum. By repeatedly performing these at each β while β increases, the algorithm can track the global minimum at each β independently of the initial parameter values. Then, as mentioned before, when $\beta = 1$, the parameterized posterior coincides with the original one (i.e., $F_1(\Theta) = -L(\Theta; \chi_{\text{obs}})$). Therefore, the convergence point of the algorithm at $\beta = 1$ exactly agrees with the desired ML estimate.

However, the above assumption does not always hold in practice. While several new *valleys* may emerge from a *valley* when β increases, the DAEM algorithm tracks the *deepest valley* at this stage. However, when another bifurcation occurs for a larger β, it is possible to develop a situation when a new deeper valley will emerge far from the previously tracked valley. In such a case, the DAEM algorithm will fail to track the global minimum. In this sense, the search strategy of the DAEM algorithm is in general suboptimal.

In other words, the DAEM algorithm guarantees the global optimality under the assumption that a new global optimum point will be close to the previously tracked global optimum point. However, even when the above assumption is not satisfied, the DAEM algorithm can provide much better estimates than methods by the original EM algorithm in the sense that the DAEM algorithm can track at least better local minima.

It is worth comparing the DA approach with the conventional simulated annealing (SA) approach. In SA, a probabilistic search is performed, where the solution may move in a detrimental direction relative to a fixed objective function depending on the temperature. On the other hand, in DA, as mentioned above, the objective function itself is gradually transformed from a global structure into a detailed structure dependent on the temperature, and a deterministic search on each transformed objective function is performed.

3.4. Another interpretation of DAEM algorithm

Recently, Neal and Hinton (1993) presented a new view of the EM algorithm where the EM-steps can be regarded as a grouped version of the *method of coordinate ascent* of the following objective function:

$$J(\tilde{p}, \Theta) \stackrel{\text{def}}{=} \mathrm{E}_{\tilde{p}}\{\log p(\chi_{\text{obs}}, \chi_{\text{mis}}\, \Theta)\} = \mathrm{E}_{\tilde{p}}\{\log \tilde{p}\}. \tag{20}$$

Here, $\mathrm{E}_{\tilde{p}}\{\cdot\}$ denotes the expectation with respect to the distribution over the range of χ_{mis} given by $\tilde{p}$ with fixed Θ under the constraint $\int \tilde{p}\, \mathrm{d}\chi_{\text{mis}} = 1$, while the M-step corresponds to the maximization of J with respect to Θ with fixed $\tilde{p}$. Indeed, one can easily see that given the current estimate $(\tilde{p}^{(t)}, \Theta^{(t)})$, we have $\tilde{p}^{(t+1)} = p(\chi_{\text{mis}}|\chi_{\text{obs}}\, \Theta^{(t)})$ at the E-step, while we have $\Theta^{(t+1)} = \arg\max_\Theta J(\tilde{p}^{(t+1)}, \Theta)$ at the M-step.

It is easy to extend the above discussion to the DAEM algorithm. That is, consider the *coordinate descent algorithm* of the following objective function:

$$J'(f, \Theta) \stackrel{\text{def}}{=} \mathrm{E}_f\{-\log p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta)\} + \frac{1}{\beta}\mathrm{E}_f\{\log f\}. \tag{21}$$

Note that when $\beta = 1$ in Eq. (21), $J' \equiv -J$. Eq. (21) is similar to Eq. (16), but is essentially different in that f in Eq. (21) is unknown, while in Eq. (16), f is known as Eq. (10).

Given the current estimate $(f^{(t)}, \Theta^{(t)})$, the minimization of J' with respect to f with fixed $\Theta^{(t)}$ under the constraint $\int f \mathrm{d}\chi_{\text{mis}} = 1$ must satisfy the following necessary conditions:

$$\int \left[\frac{1}{\beta}(\log f + 1) - \log p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta) + \lambda\right] \delta f\, \mathrm{d}\chi_{\text{mis}} = 0 \text{ and}$$

$$\int f\, \mathrm{d}\chi_{\text{mis}} = 1.$$

Here λ is a Lagrange multiplier. From these we obtain

$$f^{(t+1)} = \frac{p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta^{(t)})^\beta}{\int p(\chi_{\text{obs}}, \chi_{\text{mis}}; \Theta^{(t)})^\beta\, \mathrm{d}\chi_{\text{mis}}}. \tag{22}$$

Eq. (22) is precisely the same as the parametrized posterior given by Eq. (10).

Then, the minimization of J' with respect to Θ with fixed $f^{(t+1)}$ means finding the $\Theta^{(t+1)}$ that minimizes $J'(f^{(t+1)}, \Theta)$. By using Eq. (22), $J'(f^{(t+1)}, \Theta)$ can be written as:

$$J'(f^{(t+1)}, \Theta) = \mathrm{E}_{f^{(t+1)}}\{-\log p(\chi_{\mathrm{obs}}, \chi_{\mathrm{mis}}; \Theta)\} + \frac{1}{\beta}\mathrm{E}_{f^{(t+1)}}\{\log f^{(t+1)}\} = U_\beta(\Theta|\Theta^{(t)}) + \frac{1}{\beta}E_{f^{(t+1)}}\{\log f^{(t+1)}\}. \quad (23)$$

Since the second term on the RHS of Eq. (23) is independent of Θ, the Θ that minimizes $J'(f^{(t+1)}, \Theta)$ is given by

$$\Theta^{(t+1)} = \arg\min_{\Theta} U_\beta(\Theta|\Theta^{(t)}). \quad (24)$$

Clearly, Eqs. (22) and (24) are the same as the EM-steps of the DAEM algorithm. Noting that $1/\beta$ corresponds to the 'temperature', J' can be interpreted as a deterministic annealing variant of $-J$.

3.5. Implementation issues

When the sequence of the EM-steps converges to a *saddle point*[5] (i.e., when the Hessian matrix of $F_\beta(\Theta)$ has at least one negative eigen value) at step 3, the solution may still remain unchanged even if β increases because the stationary point clearly satisfies $\partial F_\beta(\Theta)/\partial\Theta = 0$. In such a case, a local line search in the positive and negative directions of each of the eigen vectors corresponding to the negative eigen values of the Hessian matrix of $F_\beta(\Theta)$ should be performed to escape from the saddle point. In the case the exact calculation of the Hessian is heavy, some approximation of the Hessian or some random search around the saddle point may be done.

The value of $\beta_{\min}$ can be set to a large value as long as $F_\beta(\Theta)$ has only one global minimum at $\beta_{\min}$. Experimentally, we have confirmed that $\beta_{\min}\sim 0.1$ may be enough. As for the temperature scheduling, $\beta_{\mathrm{new}} \leftarrow \beta_{\mathrm{current}} \times \mathrm{const}$, where a const of $1.1\sim 1.5$ may be reasonable. Clearly, at the final β-loop, the value of β should be one.

3.6. Simulation

To visualize how the proposed DAEM algorithm works, we consider a simple one-dimensional, two component normal mixture problem. The mixture is given by

$$p(x;\Theta) = \frac{0.3}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2}(x-m_1)^2\right\} + \frac{0.7}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2}(x-m_2)^2\right\}.$$

In this case, $\Theta = (m_1, m_2)$, and therefore, we can visualize the free energy as a three-dimensional surface defined on the (m_1, m_2)-plane. One hundred sample points in total, i.e., $\chi_{\mathrm{obs}} = \{x_k^{\mathrm{obs}},\ k = 1, \ldots 100\}$, were generated from this mixture with $m_1 = -2$ and $m_2 = 4$. The unobservable data x_k^{mis} corresponding to x_k^{obs} takes an integer 1 or 2. Therefore, the free energy defined by Eq. (15) can be calculated as

$$F_\beta(\Theta) = -\frac{1}{\beta}\log\sum_{\chi_{\mathrm{mis}}} p(\chi_{\mathrm{obs}}, \chi_{\mathrm{mis}}; \Theta)^\beta = -\frac{1}{\beta}\log\sum_{x_1^{\mathrm{mis}}=1}^{2}\cdots\sum_{x_{100}^{\mathrm{mis}}=1}^{2}\prod_{k=1}^{100} p(x_k^{\mathrm{obs}}, x_k^{\mathrm{mis}}; \Theta)^\beta = -\frac{1}{\beta}\sum_{k=1}^{100}\log\sum_{x_k^{\mathrm{mis}}=1}^{2} p(x_k^{\mathrm{obs}}, x_k^{\mathrm{mis}}; \Theta)^\beta.$$

Here the joint distribution is given by the following:

$$p(x_k^{\mathrm{obs}}, x_k^{\mathrm{mis}}; \Theta) = \begin{cases} \frac{0.3}{\sqrt{2\pi}}\exp\{-\frac{1}{2}(x_k^{\mathrm{obs}} - m_1)^2\}, & \text{for } x_k^{\mathrm{mis}} = 1 \\ \frac{0.7}{\sqrt{2\pi}}\exp\{-\frac{1}{2}(x_k^{\mathrm{obs}} - m_2)^2\}, & \text{for } x_k^{\mathrm{mis}} = 2 \end{cases}$$

We set $\Theta^{(0)} = (-2, -4)$. Fig. 1 shows the tracking result of the EM algorithm; the algorithm converges to the local maximum point. Fig. 2 shows the search process by the DAEM algorithm. Although we set $\beta_{\min} = 0.1$ and $\beta_{\mathrm{new}} \leftarrow \beta_{\mathrm{current}} \times 1.1$, we simply show the results at $\beta = 0.1, 0.26, 0.31, 0.35, 0.46$ and 1.0 to save space. Note that the vertical axes are not $F\beta(\Theta)$, but $-F_\beta(\Theta)$ to compare with L in Fig. 1. As mentioned before, when β is small ($\beta = 0.1$), $-F_\beta(\Theta)$ has only one global maximum. Hence, for arbitrary initial parameter values, the maximum can be found by the EM-steps (Fig. 2(a)). One can see that $-F_\beta(\Theta)$ presents a detailed structure of the log-likelihood function $L(\chi_{\mathrm{obs}}; \Theta)$ (shown in Fig. 2(a)–(f)) as β increases, and that the desired tracking is performed.

The final solution by the DAEM algorithm is shown in Fig. 2(f). As shown in Fig. 1, the EM algorithm was trapped by the local maximum point, while the DAEM algorithm successfully converged to near the global optimum point.

4. Application to probabilistic neural networks

4.1. Training using EM and DAEM

The DAEM algorithm can be easily applied to the learning of probabilistic neural networks using mixture models (PNN-MM). We will explain the learning algorithm for a PNN-MM based on the DAEM algorithm (Ueda and Nakano, 1995a). Then, we will compare parameter estimation results obtained by the conventional algorithm and the proposed one.

A PNN-MM is, as shown in Fig. 3, a three-layer

[5] The saddle point appears when $F_\beta(\Theta)$ just begins to bifurcate.

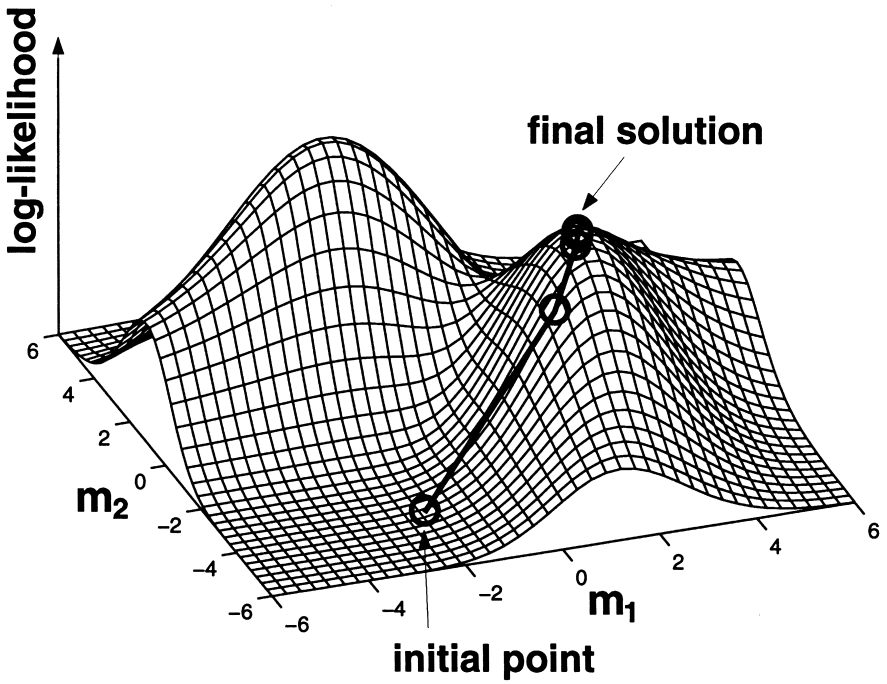

Fig. 1. Parameter estimation process by the EM algorithm with an initial value $(m_1^{(0)}, m_2^{(0)} = (-2, -4))$. The algorithm converges to a local maximum point.

feed-forward network (Specht, 1990; Ghahramani and Jordan, 1994; Jordan and Jacob, 1994). The input units pass the input values to C hidden units. A multivariate Gaussian probability density function (PDF) indexed by an unknown mean vector and covariance matrix is usually utilized as the activation function for each hidden unit. The outputs of the hidden units are weighted by component probabilities. The output layer consists of one unit representing the PDF of a mixture of Gaussians. Specifically, let $g_i(\boldsymbol{x};\ \boldsymbol{\mu}_i,\ \Sigma_i)$ be the Gaussian PDF of the i th hidden unit:

$$g_i(\boldsymbol{x}, \boldsymbol{\mu}_i, \Sigma_i = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}}\exp\left\{ -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)'\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right\},$$

where $\boldsymbol{\mu}_i$, Σ_i denote the mean vector and the covariance matrix of the ith Gaussian, d is the dimensionality of $\boldsymbol{x}$, and $\boldsymbol{x}'$ denotes the transpose of $\boldsymbol{x}$. Let α_i be the weight of the arc connecting the output of the ith hidden unit to the output unit. Then, the output of the PNN-MM can be written in the following Gaussian mixture form:

$$p(\boldsymbol{x}, \Theta) = \sum_{i=1}^{C} \alpha_i g_i(\boldsymbol{x}, \boldsymbol{\mu}_i, \Sigma_i),$$

where $\alpha_i > 0$, $\Sigma_{i=1}^{C}\ \alpha_i = 1$. Θ is a vector of all unknown parameters contained in the Gaussian mixture model:

$$\Theta = (\{\alpha_i, \boldsymbol{\mu}_i, \Sigma_i\},\ i = 1, \ldots, C).$$

Suppose we observe N samples $\chi_{\text{obs}} = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$. Note that for simplicity, we omit 'obs' here. Given the estimate $\Theta^{(t)}$ obtained after the tth iteration of the EM algorithm, at the $t+1$th iteration, the E-step computed the Q function as:

$$Q(\Theta|\Theta^{(t)}) = \sum_{k=1}^{N}\sum_{i=1}^{C}\{\log \alpha_i g_i(\boldsymbol{x}_k; \boldsymbol{\mu}_i, \Sigma_i)\} \times \frac{\alpha_i^{(t)} g_i(\boldsymbol{x}_k; \boldsymbol{\mu}_i^{(t)}, \Sigma_i^{(t)})}{\sum_{j=1}^{C}\alpha_j^{(t)} g_j(\boldsymbol{x}_k; \boldsymbol{\mu}_j^{(t)}, \Sigma_j^{(t)})}$$

and then the M-step finds the $\Theta^{(t+1)}$ that maximizes $Q(\Theta|\Theta^{(t)})$ with respect to Θ. At the M-step, we have the following $t+1$th estimates as (Streit and Luginbuhl, 1994):

For $i = 1, \ldots, C$,

$$\alpha_i^{(t+1)} = \frac{1}{N}\sum_{k=1}^{N} P(\omega_i|\boldsymbol{x}_k; \Theta^{(t)}), \tag{25}$$

$$\boldsymbol{\mu}_i^{(t+1)} = \frac{\sum_{k=1}^{N}\boldsymbol{x}_k P(\omega_i|\boldsymbol{x}_k; \Theta^{(t)})}{\sum_{k=1}^{N} P(\omega_i|\boldsymbol{x}_k; \Theta^{(t)})}, \tag{26}$$

$$\Sigma_i^{(t+1)} = \frac{\sum_{k=1}^{N}(\boldsymbol{x}_k - \boldsymbol{\mu}_i^{(t)})(\boldsymbol{x}_k - \boldsymbol{\mu}_i^{(t)})' P(\omega_i|\boldsymbol{x}_k; \Theta^{(t)})}{\sum_{k=1}^{N} P(\omega_i|\boldsymbol{x}_k; \Theta^{(t)})}. \tag{27}$$

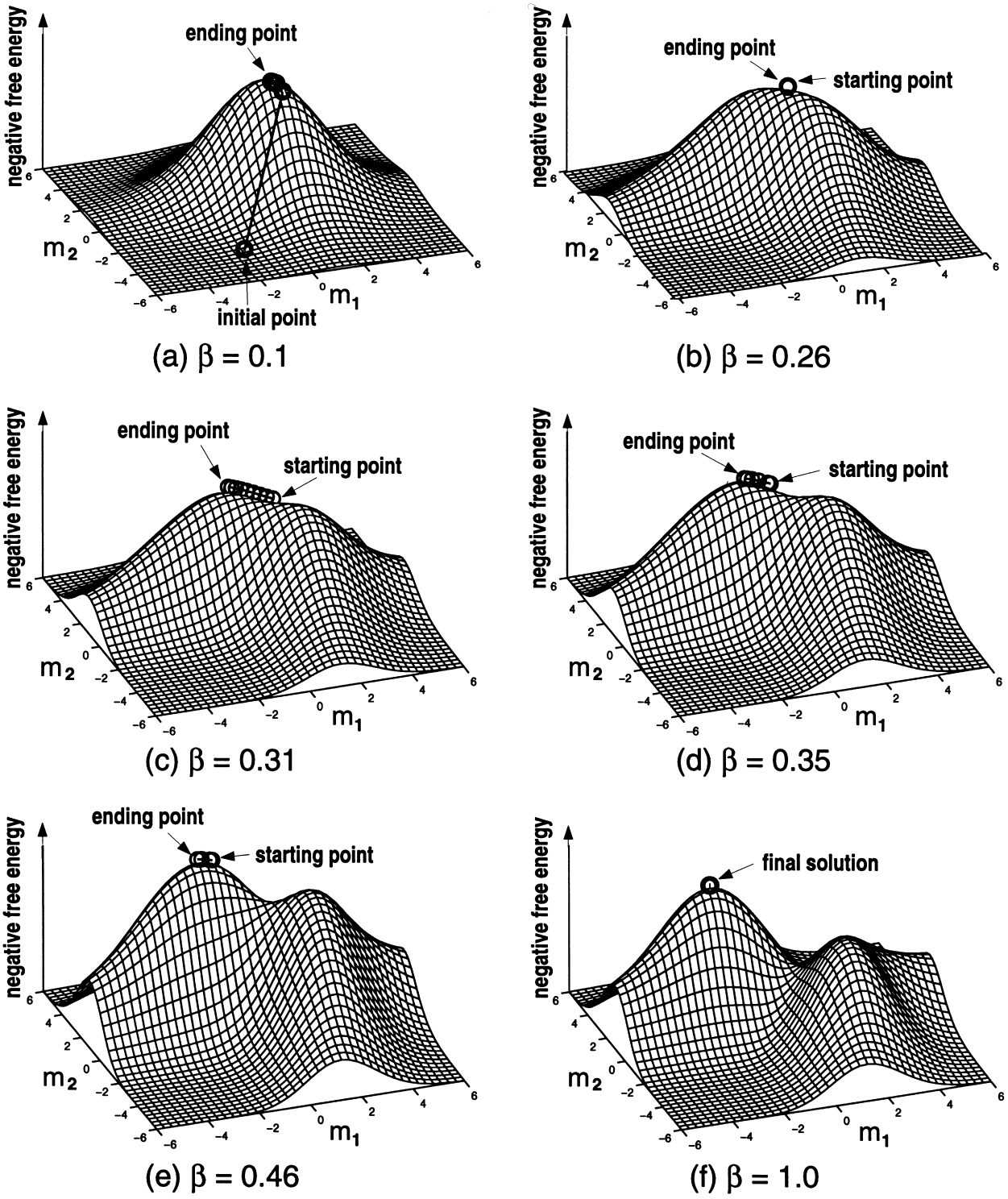

Fig. 2. Parameter estimation process by the DAEM algorithm with an initial value $(m_1^{(0)}, m_2^{(0)} = (-2, -4))$. The algorithm successively tracks the global maximum point at each β and finally finds the maximum likelihood estimate. Each of (a)–(f) shows a negative free energy surface corresponding to a β value.

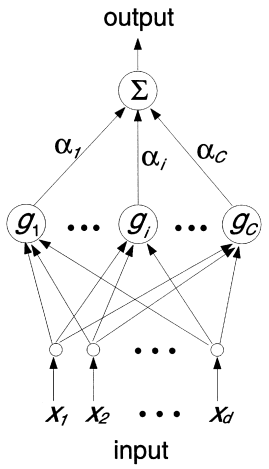

Fig. 3. A probabilistic neural network using a Gaussian mixture.

Here, $P(\omega_i|\boldsymbol{x}_k; \Theta^{(t)})$ means the posterior probability that x_k belongs to the ith component ω_i, and is given by:

$$P(\omega_i|\boldsymbol{x}_k;\Theta^{(t)}) = \frac{\alpha_i^{(t)} g_i(\boldsymbol{x}_k;\boldsymbol{\mu}_i^{(t)},\Sigma_i^{(t)})}{\sum_{j=1}^{C} \alpha_j^{(t)} g_j(\boldsymbol{x}_k;\boldsymbol{\mu}_j^{(t)},\Sigma_j^{(t)})},$$

$$\text{for } i=1,\ldots,C,\ =1,\ldots,N. \tag{28}$$

Therefore, as one can easily see, in the DAEM algorithm, $P(\omega_i|\boldsymbol{x}_k; \Theta^{(t)})$ in Eqs. (25)–(27) is simply replaced with the following posterior parameterized by β:

$$f(\omega_i|\boldsymbol{x}_k;\Theta^{(t)}) = \frac{\{\alpha_i^{(t)} g_i(\boldsymbol{x}_k;\boldsymbol{\mu}_i^{(t)},\Sigma_i^{(t)})\}^{\beta}}{\sum_{j=1}^{C} \{\alpha_j^{(t)} g_j(\boldsymbol{x}_k;\boldsymbol{\mu}_j^{(t)},\Sigma_j^{(t)})\}^{\beta}},$$

$$\text{for } i=1,\ldots,C,\ \ k=1,\ldots,N. \tag{29}$$

Finally, adding a β-loop to the reestimation procedure given by Eqs. (25)–(27), we can obtain the DAEM-based training algorithm for PNN-MM.

4.2. Experimental results

The parameter estimation results by the EM and DAEM algorithms are shown in Fig. 4 and Table 1. The training data generated from a target PDF are shown in Fig. 4(a). The target PDF was a two-dimensional, three-component Gaussian mixture distribution. The true parameters corresponding to this PDF are listed in Table 1 (Θ^*). Each contour shown in Fig. 4(b)–(c) corresponds to the Gaussian boundary of each component. In the DAEM algorithm, we set $\beta_{\text{min}} = 0.5$, $\beta_{\text{new}} \leftarrow \beta_{\text{current}} \times 1.2$. In Fig. 4, t denotes the training time.[6] The initial parameter values were set as shown in Fig. 4(b) and Table 1 ($\Theta^{(0)}$). As shown in Fig. 4(b), given the initial parameter values, the EM algorithm converges to a poor local optimum, while the DAEM algorithm, shown in Fig. 4(c), obtains a value near the global optimum.

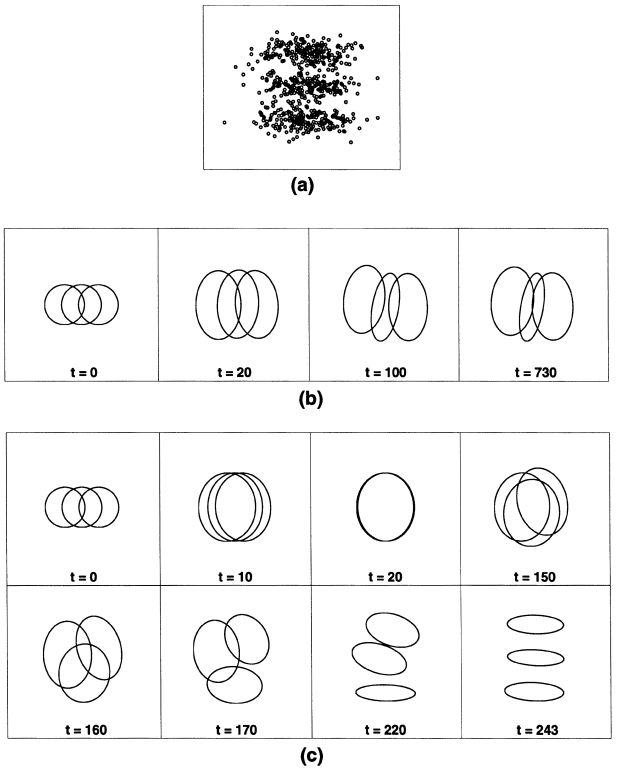

Fig. 4. Parameter estimation results of a Gaussian mixture distribution by the EM and DAEM algorithms. (a) Training data generated from a target probability density function (two-dimensional, three-component Gaussian mixture distribution). (b) Estimation process by the EM algorithm. Each contour corresponds to the Gaussian boundary of each component. t denotes the training time. For the initial values shown at $t=0$, the algorithm converges to a poor local optimum. (c) Estimation process by the DAEM algorithm. The algorithm converges to near the global optimum.

When β is small, as mentioned before, since the posterior probability becomes nearly uniform, each $\boldsymbol{x}_k$ almost equally contributes to each component of the mixture. As a result, all components completely come to overlap as one component (Fig. 4(c), $t=20$). Then, by gradually increasing β, the influence of each $\boldsymbol{x}_k$ is gradually localized and the PDF estimate gradually approaches the true PDF. Consequently, the DAEM algorithm converges to near the optimum result free from the initial parameter values.

We also tested the DAEM algorithm using a high-dimensional real data where the local optima problem is crucial. The data used here were two to six facial images (photographs), each of which was a 50-dimensional feature vector.[7] Given these feature vectors, we tried to fit a C-component Gaussian mixture distribution ($C = 10, 20$) by using each of the EM and the DAEM algorithms. In the experiment, we adopted with diagonal covariance matrix

[6] In the DAEM algorithm, t denotes the total training time, not being reset when β is increased.

[7] Originally, 205-dimensional features were extracted from the image, then these features were reduced to 50-dimensional features by using the KL expansion technique. See Oda et al. (1996) for details.

Table 1
Parameter estimation results by the EM and DAEM algorithm

Parameters: Θ	True values: Θ^*	Initial values: $\Theta^{(0)}$	Estimates by EM: $\tilde{\Theta}$	Estimates by DAEM: $\tilde{\Theta}$
α_1	0.33	0.333	0.381	0.344
α_2	0.333	0.333	0.201	0.342
α_3	0.333	0.333	0.418	0.341
$\boldsymbol{\mu}_1$	$(0\ -2)'$	$(-1\ 0)'$	$(-1.201\ 0.207)'$	$(-0.050\ -1.984)'$
$\boldsymbol{\mu}_2$	$(0\ 0)'$	$(0\ 0)'$	$(-0.054\ -0.147)'$	$(0.057\ 0.032)'$
$\boldsymbol{\mu}_3$	$(0\ 2)'$	$(1\ 0)'$	$(1.162\ -0.100)'$	$(0.050\ 1.994)'$
Σ_1	$\begin{pmatrix} 2 & 0 \\ 0 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1.140 & 0.176 \\ 0.176 & 2.927 \end{pmatrix}$	$\begin{pmatrix} 2.192 & -0.078 \\ -0.078 & 0.214 \end{pmatrix}$
Σ_2	$\begin{pmatrix} 2 & 0 \\ 0 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0.392 & 0.454 \\ 0.454 & 2.898 \end{pmatrix}$	$\begin{pmatrix} 1.984 & -0.088 \\ -0.088 & 0.162 \end{pmatrix}$
Σ_3	$\begin{pmatrix} 2 & 0 \\ 0 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1.035 & -0.041 \\ -0.041 & 2.835 \end{pmatrix}$	$\begin{pmatrix} 1.987 & -0.042 \\ -0.042 & 0.215 \end{pmatrix}$

model for each Gaussian.[8] That is, the covariance matrix of the ith Gaussian is given by

$$\Sigma_i = \mathrm{diag}(\sigma_{i1}^2, \ldots, \sigma_{i50}^2),\ \ i = 1, \ldots, C.$$

In the experiment, initial parameter values for each of the EM and the DAEM algorithms were determined by the following two alternatives:

- *Random initialization*: First, mean vectors, $\boldsymbol{\mu}_i^{(0)}, \ldots, \boldsymbol{\mu}_c^{(0)}$, are initialized to the value of C vectors chosen randomly from the given training data. Next, with these mean vectors, each training vector $\boldsymbol{x}$ is classified into one of C classes based on the *nearest-neighbour rule*:

  Decide $\boldsymbol{x}$ as a class i if $d(\boldsymbol{x}, \boldsymbol{\mu}_i^{(0)}) \leq d(\boldsymbol{x}, \boldsymbol{\mu}_j^{(0)})$, $\forall_j \neq i$, where $d(\boldsymbol{x}, \boldsymbol{\mu}^{(0)})$ is the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{\mu}^{(0)}$. Moreover, the kth component of $\Sigma_i^{(0)}$, $\sigma_{ik}^{2(0)}$, is initialized as $\sigma_{ik}^{2(0)} = \Sigma_{\boldsymbol{x} \in \mathrm{class}_i} (x_k - \mu_{ik}^{(0)})^2 / N_i$. Here x_k is the kth component of $\boldsymbol{x}$ belonging to class i and $\mu_{ik}^{(0)}$ is the kth component of $\boldsymbol{\mu}^{(0)}$. N_i denotes the number of training data belonging to class i. The initial allocation rate of the ith class is given by $\alpha_k^{(0)} = N_i/206$.

- *K-means method*: This alternative initialization using K-means algorithm is usually utilized in Gaussian mixture estimation problems (Huang et al., 1990). The procedure is exactly the same as the previous one except for the mean vectors' initialization. That is, the mean vectors here are further initialized by the K-means algorithm after the random choice. Then, the rest procedures for initializing α_i and Σ_i for $i = 1,...,C$ are the same as those of the above *random initialization*.

Each of the EM and the DAEM algorithms was performed for 20 times from independent initial parameter values and *likelihood* value was calculated for each trial when the EM (DAEM) algorithm converged.

Fig. 5 shows mean and standard deviation (error bar) for calculated likelihood values over 20 trials for each algorithm. In Fig. 5, mean value and standard deviation correspond to the initial parameter value obtained by each initialization method are also shown to compare how each of the EM and the DAEM algorithms increases the likelihood value, starting from the initial likelihood value. From Fig. 5, one can see that in both initializations the DAEM algorithm could obtain bigger and more stable likelihood values than those of the EM algorithm. Clearly, this means that the estimation results by the DAEM algorithm are better than those by the EM algorithm in the maximum likelihood sense.

The superiority of the DAEM algorithm over the EM algorithm was dominant when $C = 20$. This seems to be reasonable because, in general, the number of local maxima of a likelihood function becomes large as the dimensionality of the unknown parameter increases, and therefore the probability that the EM algorithm is trapped by some of poor local maxima can become much higher as the value of C increases than the DAEM algorithm. With respect to computational time, however, the DAEM algorithm was about 15 to 20 times slower than the EM algorithm.

5. Conclusions

We have presented a deterministic annealing variant of the EM algorithm to overcome the local maxima problem associated with the original EM algorithm. In our DA approach, the annealing process begins at a high temperature where the objective function itself is smoothed such that it has only one global optimum point. Then, assuming that the shape of the objective gradually approaches that of the original objective function as the temperature decreases, the DAEM algorithm successively tracks each global

[8] Because the number of training vectors was 206, which was relatively small compared with the dimensionality of the vector, a full-covariance model might be inappropriate.

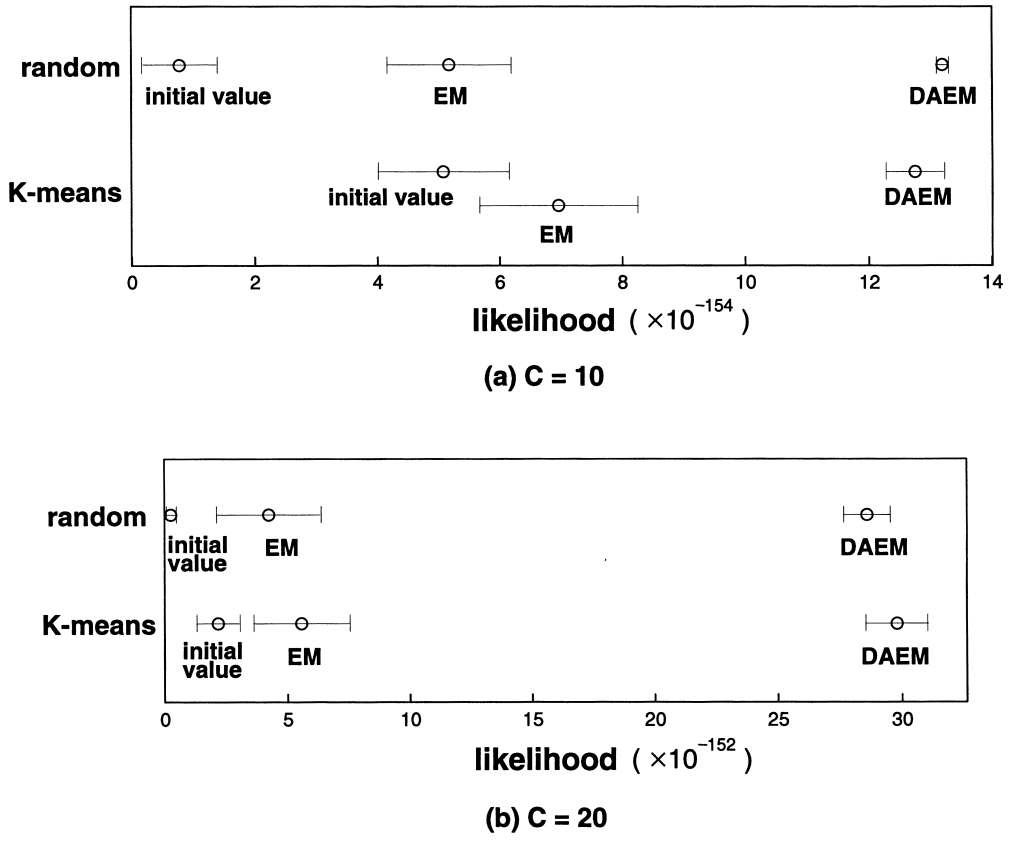

Fig. 5. Likelihood comparison for the EM and the DAEM algorithms for a 50-dimensional, C-component Gaussian mixture estimation problem. The mean value (○) and the standard deviation (error bar) for calculated likelihood values over 20 independent trials are shown for each algorithm. The mean value and the standard deviation corresponding to the initial parameter value obtained by each initialization method ('random' and 'K-means', see text for details) are also shown.

optimum point of the objective functions while the temperature decreases, and finally finds the global optimum point of the original objective function. When the assumption is not satisfied, the DAEM algorithm can obtain better estimates than the EM algorithm in the sense that the DAEM algorithm can converge to better local optima.

An important characteristic of the DAEM algorithm is that, unlike the conventional simulated annealing, a *deterministic* search rather than a *stochastic* search is performed on the smooth objective function at each temperature. Hence, the total search can be executed more efficiently than that by the simulated annealing approach. In our DAEM algorithm, a new posterior parameterized by the temperature makes the algorithm achieve this type of search. Although the DAEM algorithm has been derived using the principle of maximum entropy, the validity of the algorithm has also been formally shown. Computer simulations including an application to the training of probabilistic neural networks have domonstrated the effectiveness of the DAEM algorithm.

We believe that our method can be useful for applications of the EM algorithm. In addition, we expect this type of search to give some new insights into other optimization problems.

References

Buhmann, J., & Kuhnel, H. (1993). Complexity optimized data clustering by competitive neural networks. *Neural Computation*, *5*, 75–88.

Byrne, W. (1992). Alternating minimization and Boltzmann machine learning. *IEEE Trans. Neural Networks*, *3*, 612–620.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum-likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. Ser. B (methodological)*, *39*, 1–38.

Feder, M. (1986). Maximum entropy as a special case of the minimum description length criterion. *IEEE Trans. Inf. Theory*, *32 (6)*, 847–849.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distribution and the Bayesian restoration in images. *IEEE Trans. Pattern Anal. Machine Intell.*, *6 (6)*, 721–741.

Ghahramani, Z. & Jordan, M.I. (1994). Supervised learning from incomplete data via an EM approach. In J. Cowan et al. (Eds.), *Advances in NIPS 6* (pp. 120–127). Morgan Kaufmann.

Hinton, G.E., Dayan, P., Frey, B.J., & Neal, R.M. (1995). The 'wake–sleep' algorithm for unsupervised neural networks. *Science*, *268 (26)*, 1158–1161.

Huang, X.D., Ariki, Y. and Jack, M.A. (1990). *Hidden Markov Models for Speech Recognition*. Edinburgh University Press.

Jordan, M.I., & Jacob, R.A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, *6*, 181–214.

McLachlan, G. & Basford, K. (1988). *Mixture Models: Inference and Application to Clustering*. New York and Basel: Marcel Dekker.

Neal, R.M. and Hinton, G.E. (1993). A new view of the EM algorithm that justifies incremental and other variants. *Biometrika* (submitted).

Oda, M., Akamatsu, S., & Fukamachi, H. (1996). Adaptability of K-L expansion technique for ambiguous face image retrieval. *Electron. Commun. Jpn., 79 (10)*, 35–46.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 465–471.

Rose, K., Gurewitz, E., & Fox, G.C. (1992). Vector quantization by deterministic annealing. *IEEE Trans. Inf. Theory*, *38 (4)*, 1249–1257.

Specht, D.F. (1990). Probabilistic neural networks. *Neural Networks*, *3*, 109–118.

Streit, R.L., & Luginbuhl, T.E. (1994). Maximum likelihood training of probabilistic neural networks. *IEEE Trans. Neural Networks*, *5 (5)*, 764–783.

Traven, H.G.C. (1991). A neural network approach to statistical pattern classification by semiparametric estimation of probability density functions. *IEEE Trans. Neural Networks*, *2 (3)*, 366–377.

Ueda, N. & Nakano, R. (1994). Mixture density estimation via EM algorithm with deterministic annealing. In *Proceedings of the IEEE Neural Networks for Signal Processing* (pp. 69–77). Greece. New York: IEEE.

Ueda, N. & Nakano, R. (1995a). A new maximum likelihood training and application to probabilistic neural networks. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN95)* (pp. 497–502). Paris: EC2&Cíe.

Ueda, N. & Nakano, R. (1995b). Deterministic annealing variant of the EM algorithm. In G. Tesauro et al. (Eds.), *Advances in NIPS 7* (pp. 545–552). Cambridge, MA: MIT Press.

Wong, Y. (1993). Clustering data by melting. *Neural Computation*, *5*, 89–104.

Yuille, A.L., Stolorz, P., & Utans, J. (1994). Statistical physics, mixtures of distributions and the EM algorithm. *Neural Computation*, *6*, 334–340