# Convergence of On-line EM Algorithm

Masa-aki Sato †‡

†ATR Institute International ISD
‡CREST, Japan Science and Techonology Corporation
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
E-mail: masaaki@hip.atr.co.jp

## Abstract

*In this article, an on-line EM algorithm is derived for general Exponential Family models with Hidden variables (EFH models) by introducing a discount factor. It is proven that the on-line EM algorithm is equivalent to a stochastic gradient method with the inverse of the Fisher information matrix as a coefficient matrix. As a result, the stochastic approximation theory guarantees the convergence to a local maximum of the likelihood function.*

*The performance of the on-line EM algorithm is examined by using the mixture of Gaussian model, which is a special type of the EFH model. The simulation results show that the on-line EM algorithm is faster than the batch EM algorithm and the on-line gradient ascent algorithm. It is also shown that the introduction of a discount factor is crucial for fast convergence.*

## 1 Introduction

In recent years, the EM algorithm [3] has been applied to a number of neural network models [1,4,7]. The EM algorithm is a general method for finding a maximum likelihood estimator for a stochastic model with hidden variables. Many neural network models have corresponding stochastic models [1]. The stochastic formulation helps to study the learning problem of neural networks. Hidden variables in the stochastic models correspond to hidden units in the neural networks, and these variables expand the representational capability of the models.

The EM algorithm is a batch algorithm in which all the training data are used to update model parameters at each iteration. It was proven that the likelihood for the training data always increases (or does not change) at each iteration of the EM algorithm [3]. Therefore, the EM algorithm guarantees the convergence to a local maximum of the likelihood function.

In real world problems, it is often desirable to use an on-line learning algorithm in which the training data are supplied one by one and the model parameters are updated each time using the current data. The most popular on-line algorithm is the on-line gradient descent algorithm. This algorithm can be considered as a stochastic approximation method [5] for finding a local minimum of an error function.

On-line EM algorithms have been proposed by several authors [6,10]. Although their convergence have been proved, the convergence speed of these on-line EM algorithms have been shown to be slow compared to the batch EM algorithm. The reason for the slow convergence may be the fact that these algorithms do not include a discount factor to forget the earlier inaccurate estimator contribution. On-line EM algorithms with the discount factor have also been proposed [4,8], and their convergence speed have been shown to be faster than those without the discount factor. However, there have been no convergence proof of these algorithms. Neal and Hinton [7] introduced a free energy formulation for the EM algorithm and proposed a wide variety of incremental EM algorithms whose convergence is guaranteed. One drawback of their study is that their algorithms either need additional storage variables for all of the training data or need to see all of the training data at each iteration. This prevents these algorithms from being applied to real-time on-line settings where new data are indefinitely supplied each time.

In our previous paper, we proved the convergence of the on-line EM algorithm for a normalized Gaussian network [9]. In this paper, we study more general models, namely the Exponential Family models with Hidden variables (EFH models). By using the free energy formulation with the discount factor, we derive an on-line EM algorithm for the EFH models. We prove that the on-line EM algorithm with the discount factor is equivalent to a stochastic gradient method with the inverse of the Fisher information matrix as a coefficient matrix. As a result, the stochastic approximation theory [5] guarantees the convergence to a local maximum of the likelihood function. We also propose a systematic design of the discount factor schedule.

The performance of the on-line EM algorithm is examined by using the mixture of Gaussian model, which is a special type of the EFH model. The simulation results show that the on-line EM algorithm with the discount factor is faster than the batch EM algorithm and the on-line gradient ascent algorithm. It is also shown that the introduction of a discount factor is crucial for fast convergence.

## 2 Model with hidden variables

In this paper, we study the EM algorithm [3] for the general Exponential Family models with Hidden variables (EFH models)[1]. An EFH model for an $N$-dimensional vector variable $\mathbf{x} = (x_1, ..., x_N)^T$ is defined by a probability distribution,

$$\begin{aligned} P(\mathbf{x}|\boldsymbol{\theta}) &= \int d\mathbf{z}\ P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}), \qquad (1) \\ P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) &= \exp\left[\mathbf{r}(\mathbf{x}, \mathbf{z}) \cdot \boldsymbol{\theta} + r_0(\mathbf{x}, \mathbf{z}) - \Psi(\boldsymbol{\theta})\right], \end{aligned}$$

where $\mathbf{z} = (z_1, ..., z_M)^T$ denotes an $M$-dimensional vector hidden variable and $\boldsymbol{\theta} = (\theta_1, ..., \theta_K)^T$ denotes a set of model parameters called the natural parameter. A set of sufficient statistics is denoted by $\mathbf{r}(\mathbf{x}, \mathbf{z}) = (r_1(\mathbf{x}, \mathbf{z}), ..., r_K(\mathbf{x}, \mathbf{z}))^T$. An inner product of two vectors $\mathbf{r}$ and $\boldsymbol{\theta}$ is denoted by $\mathbf{r} \cdot \boldsymbol{\theta} = \sum_{k=1}^K r_k \theta_k$. The normalization factor $\Psi(\boldsymbol{\theta})$ is determined by

$$\exp\left[\Psi(\boldsymbol{\theta})\right] = \int d\mathbf{x} d\mathbf{z}\ \exp\left[\mathbf{r}(\mathbf{x}, \mathbf{z}) \cdot \boldsymbol{\theta} + r_0(\mathbf{x}, \mathbf{z})\right],$$

which is derived from the probability condition $\int d\mathbf{x}\ P(\mathbf{x}|\boldsymbol{\theta}) = 1$. $P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ represents the probability distribution for a complete event $(\mathbf{x}, \mathbf{z})$. If the hidden variable is a discrete variable, the integration $\int d\mathbf{z}$ is replaced by the summation $\sum_{\mathbf{z}}$.

The expectation parameter for the EFH model, $\boldsymbol{\phi} = (\phi_1, ..., \phi_K)^T$, is defined [1] by

$$\begin{aligned} \boldsymbol{\phi} &= \partial \Psi(\boldsymbol{\theta}) / \partial \boldsymbol{\theta} \\ &= E[\mathbf{r}(\mathbf{x}, \mathbf{z})|\boldsymbol{\theta}] = \int d\mathbf{x} d\mathbf{z}\ \mathbf{r}(\mathbf{x}, \mathbf{z}) P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}). \end{aligned}$$

Conversely, the natural parameter $\boldsymbol{\theta}$ can be expressed in terms of the expectation parameter $\boldsymbol{\phi}$ by using the Legendre transformation:

$$\begin{aligned} H(\boldsymbol{\phi}) &= \boldsymbol{\phi} \cdot \boldsymbol{\theta} - \Psi(\boldsymbol{\theta}), \\ \boldsymbol{\theta} &= \boldsymbol{\theta}(\boldsymbol{\phi}) = \partial H(\boldsymbol{\phi}) / \partial \boldsymbol{\phi}. \qquad (2) \end{aligned}$$

For later convenience, let us define the Fisher information matrix $\mathbf{V}(\boldsymbol{\phi})$ for the probability distributions $P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}(\boldsymbol{\phi}))$,

$$\begin{aligned} \mathbf{V}(\boldsymbol{\phi}) &= E\left[\left(\frac{\partial \log P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}(\boldsymbol{\phi}))}{\partial \boldsymbol{\phi}}\right)\right. \\ &\quad \left.\times \left(\frac{\partial \log P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}(\boldsymbol{\phi}))}{\partial \boldsymbol{\phi}}\right) |\boldsymbol{\theta}(\boldsymbol{\phi})\right] \qquad (3) \\ &= \frac{\partial^2 H(\boldsymbol{\phi})}{\partial \boldsymbol{\phi} \partial \boldsymbol{\phi}} = \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\phi}} \end{aligned}$$

## 3 Free energy

In this section, we review the free energy formulation for the EM algorithm [7]. The log-likelihood for a set of observed data, $\mathbf{X}\{T\} = \{\mathbf{x}(t)|t = 1, ..., T\}$, is given by

$$L(\boldsymbol{\theta}|\mathbf{X}\{T\}) = \log P(\mathbf{X}\{T\}|\boldsymbol{\theta}) = \sum_{t=1}^T \log P(\mathbf{x}(t)|\boldsymbol{\theta}).$$

A free energy for the observed data $\mathbf{X}\{T\}$ is defined by

$$\begin{aligned} F(Q\{T\}, \boldsymbol{\theta}|\mathbf{X}\{T\}) &= \sum_{t=1}^T \int d\mathbf{z}(t)\ Q(\mathbf{z}(t)) \qquad (4) \\ &\quad \times \log(P(\mathbf{x}(t), \mathbf{z}(t)|\boldsymbol{\theta}) / Q(\mathbf{z}(t))), \end{aligned}$$

where $Q\{T\} = \{Q(\mathbf{z}(t))|t = 1, ..., T\}$ is a set of variational distributions for the hidden variables $\mathbf{Z}\{T\} = \{\mathbf{z}(t)|t = 1, ..., T\}$. The free energy can be written as

$$\begin{aligned} &F(Q\{T\}, \boldsymbol{\theta}|\mathbf{X}\{T\}) = L(\boldsymbol{\theta}|\mathbf{X}\{T\}) \qquad (5) \\ &- \sum_{t=1}^T \int d\mathbf{z}(t)\ Q(\mathbf{z}(t)) \log(Q(\mathbf{z}(t)) / P(\mathbf{z}(t)|\mathbf{x}(t), \boldsymbol{\theta})). \end{aligned}$$

The second term in the r.h.s of (5) is the Kullback-Leiber(KL) divergence between $Q(\mathbf{z}(t))$ and the posterior probability distribution $P(\mathbf{z}(t)|\mathbf{x}(t), \boldsymbol{\theta})$. Therefore, the maximum of the free energy w.r.t.(:with respect to) $Q\{T\}$ is equal to the log likelihood $L(\boldsymbol{\theta}|\mathbf{X}\{T\})$. Consequently, the maximization of the free energy w.r.t. $Q\{T\}$ and $\boldsymbol{\theta}$ is equivalent to the maximization of the log-likelihood w.r.t. $\boldsymbol{\theta}$, i.e, the maximum likelihood estimation.

The EM algorithm can be derived by alternately maximizing the free energy w.r.t. $Q\{T\}$ and $\boldsymbol{\theta}$. Let $\bar{\boldsymbol{\theta}}$ be the present estimator. In the E(:estimation)-step, the free energy is maximized w.r.t. $Q\{T\}$ while $\boldsymbol{\theta}$ is fixed at $\bar{\boldsymbol{\theta}}$. The maximum solution of $Q\{T\}$ is given by the posterior distribution for the hidden variable:

$$\begin{aligned} Q(\mathbf{z}(t)) &= P(\mathbf{z}(t)|\mathbf{x}(t), \bar{\boldsymbol{\theta}}) \\ &= P(\mathbf{x}(t), \mathbf{z}(t)|\bar{\boldsymbol{\theta}}) / P(\mathbf{x}(t)|\bar{\boldsymbol{\theta}}). \end{aligned}$$

Substituting this solution into the free energy (4), the free energy is expressed as

$$F(Q\{T\}, \boldsymbol{\theta}|\mathbf{X}\{T\}) = L(\boldsymbol{\theta}|\bar{\boldsymbol{\theta}}, \mathbf{X}\{T\}) + (\boldsymbol{\theta} - indep.),$$

where the expected log-likelihood for the complete data $L(\boldsymbol{\theta}|\bar{\boldsymbol{\theta}}, \mathbf{X}\{T\})$ is defined by

$$\begin{aligned} L(\boldsymbol{\theta}|\bar{\boldsymbol{\theta}}, \mathbf{X}\{T\}) &= \sum_{t=1}^T \int d\mathbf{z}(t) P(\mathbf{x}(t), \mathbf{z}(t)|\bar{\boldsymbol{\theta}}) \\ &\quad \times \log P(\mathbf{x}(t), \mathbf{z}(t)|\boldsymbol{\theta}). \end{aligned}$$

Therefore, the maximization of the free energy w.r.t. $\boldsymbol{\theta}$ is equivalent to the maximization of the expected log-likelihood for the complete data w.r.t. $\boldsymbol{\theta}$, which defines the M(:maximization)-step.

## 4 On-line EM Algorithm

In the case of on-line learning, the observed data $\{\mathbf{x}(1), \mathbf{x}(2), ...\}$ are supplied one by one, and the estimator is changed after each observation. Let $\mathbf{X}\{\tau\} = \{\mathbf{x}(t)|t = 1, ..., \tau\}$ be the observed data up to now. A discounted free energy for $\mathbf{X}\{\tau\}$ is defined by

$$F^{\lambda}(Q\{\tau\}, \boldsymbol{\theta}|\mathbf{X}\{\tau\}) = \eta(\tau) \sum_{t=1}^{\tau} \left( \prod_{s=t+1}^{\tau} \lambda(s) \right) \tag{6}$$
$$\times \int d\mathbf{z}(t)\ Q(\mathbf{z}(t)) \log(P(\mathbf{x}(t), \mathbf{z}(t)|\boldsymbol{\theta})/Q(\mathbf{z}(t)))$$

where a time dependent discount factor $\lambda(t)$ $(0 \leq \lambda(t) \leq 1,\ t = 2, 3, ...)$ is introduced for forgetting the earlier inaccurate estimator contributions. The normalization constant $\eta(\tau)$ is given by

$$\eta(\tau) = \left[ \sum_{t=1}^{\tau} \left( \prod_{s=t+1}^{\tau} \lambda(s) \right) \right]^{-1}.$$

The on-line EM algorithm can be derived from the successive maximization of the discounted free energy (6). Let us assume that $Q\{\tau - 1\} = \{Q(\mathbf{z}(t))|t = 1, ..., \tau - 1\}$ and $\boldsymbol{\theta}(\tau - 1)$ have been determined for an observed data set $\mathbf{X}\{\tau - 1\} = \{\mathbf{x}(t)|t = 1, ..., \tau - 1\}$. After observing the current data $\mathbf{x}(\tau)$, the discounted free energy $F^{\lambda}(Q\{\tau\}, \boldsymbol{\theta}|\mathbf{X}\{\tau\})$ is maximized with respect to $Q(\mathbf{z}(\tau))$ while $\boldsymbol{\theta}$ is fixed at $\boldsymbol{\theta}(\tau - 1)$. The solution is given by

$$Q(\mathbf{z}(\tau)) = P(\mathbf{z}(\tau)|\mathbf{x}(\tau), \boldsymbol{\theta}(\tau - 1)). \tag{7}$$

This defines the E-step. In the M-step, the discounted free energy $F^{\lambda}(Q\{\tau\}, \boldsymbol{\theta}|\mathbf{X}\{\tau\})$ is maximized with respect to $\boldsymbol{\theta}$, while $Q\{\tau\}$ is fixed. The solution is given by

$$\boldsymbol{\phi}(\tau) = \langle\langle \mathbf{r}(\mathbf{x}, \mathbf{z}) \rangle\rangle(\tau),$$

where the discounted average $\langle\langle \cdot \rangle\rangle(\tau)$ is defined by

$$\langle\langle \mathbf{r}(\mathbf{x}, \mathbf{z}) \rangle\rangle(\tau) = \eta(\tau) \sum_{t=1}^{\tau} \left( \prod_{s=t+1}^{\tau} \lambda(s) \right) \tag{8}$$
$$\times \int d\mathbf{z}(t)\ Q(\mathbf{z}(t)) \mathbf{r}(\mathbf{x}(t), \mathbf{z}(t)).$$

The discounted average can be calculated by using a step-wise equation:

$$\begin{aligned} \langle\langle \mathbf{r}(\mathbf{x}, \mathbf{z}) \rangle\rangle(\tau) &= (1 - \eta(\tau)) \langle\langle \mathbf{r}(\mathbf{x}, \mathbf{z}) \rangle\rangle(\tau - 1) \\ &\quad + \eta(\tau) E_{\mathbf{z}} \left[ \mathbf{r}(\mathbf{x}, \mathbf{z}) | \boldsymbol{\theta}(\tau - 1) \right], \\ \eta(\tau) &= (1 + \lambda(\tau)/\eta(\tau - 1))^{-1}. \end{aligned}$$

By using (7), the expectation value for the sufficient statistics $E_{\mathbf{z}} \left[ \mathbf{r}(\mathbf{x}, \mathbf{z}) | \boldsymbol{\theta}(\tau - 1) \right]$ is given by

$$E_{\mathbf{z}} \left[ \mathbf{r}(\mathbf{x}, \mathbf{z}) | \boldsymbol{\theta}(\tau - 1) \right] = \int d\mathbf{z}(\tau)$$
$$\times P(\mathbf{z}(\tau)|\mathbf{x}(\tau), \boldsymbol{\theta}(\tau - 1)) \mathbf{r}(\mathbf{x}(\tau), \mathbf{z}(\tau)). \tag{9}$$

The recursive formula for $\boldsymbol{\phi}(\tau)$ is derived from the above equations:

$$\begin{aligned} \Delta\boldsymbol{\phi}(\tau) &= \boldsymbol{\phi}(\tau) - \boldsymbol{\phi}(\tau - 1) \\ &= \eta(\tau) \Big( E_{\mathbf{z}} \left[ \mathbf{r}(\mathbf{x}, \mathbf{z}) | \boldsymbol{\theta}(\tau - 1) \right] - \boldsymbol{\phi}(\tau - 1) \Big). \end{aligned} \tag{10}$$

The new estimator $\boldsymbol{\theta}(\tau)$ is obtained by using (2),

$$\boldsymbol{\theta}(\tau) = \boldsymbol{\theta}(\boldsymbol{\phi}(\tau)) = \frac{\partial H(\boldsymbol{\phi})}{\partial \boldsymbol{\phi}} |_{\boldsymbol{\phi}(\tau)}. \tag{11}$$

## 5 Stochastic approximation

If an infinite number of data, which are drawn independently according to the input data distribution $\rho(\mathbf{x})$, is available, the log-likelihood function is given by

$$L(\boldsymbol{\theta}) = E[\log \left( \int d\mathbf{z} \quad P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) \right)]_{\rho}, \tag{12}$$

where $E[\cdot]_{\rho}$ denotes the expectation value with respect to the unknown input data distribution $\rho(\mathbf{x})$. The maximum likelihood condition $\partial L(\boldsymbol{\theta})/\partial \boldsymbol{\theta} = 0$ turns out to be

$$\begin{aligned} \boldsymbol{\phi} &= E[\int d\mathbf{z} \quad \mathbf{r}(\mathbf{x}, \mathbf{z}) P(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})]_{\rho} \\ \boldsymbol{\theta} &= \boldsymbol{\theta}(\boldsymbol{\phi}) = \partial H(\boldsymbol{\phi})/\partial \boldsymbol{\phi}. \end{aligned}$$

Using the log-likelihood for the current data $\mathbf{x}(\tau)$,

$$L(\boldsymbol{\theta}|\mathbf{x}(\tau)) = \log(\int d\mathbf{z} P(\mathbf{x}(\tau), \mathbf{z}|\boldsymbol{\theta})), \tag{13}$$

the update rules (10) and (11) are further rewritten as

$$\begin{aligned} \Delta\boldsymbol{\phi}(\tau) &= \eta(\tau) \left( \frac{\partial L(\boldsymbol{\theta}|\mathbf{x}(\tau))}{\partial \boldsymbol{\theta}} \right) |_{\boldsymbol{\theta}(\tau-1)} \\ &= \eta(\tau) \left( \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{\theta}} \right) \left( \frac{\partial L(\boldsymbol{\theta}(\boldsymbol{\phi})|\mathbf{x}(\tau))}{\partial \boldsymbol{\phi}} \right) |_{\boldsymbol{\phi}(\tau-1)} \\ &= \eta(\tau) (\mathbf{V}(\boldsymbol{\phi}(\tau - 1)))^{-1} \\ &\quad \times \left( \frac{\partial L(\boldsymbol{\theta}(\boldsymbol{\phi})|\mathbf{x}(\tau))}{\partial \boldsymbol{\phi}} \right) |_{\boldsymbol{\phi}(\tau-1)}, \end{aligned} \tag{14}$$

where the relation (3) is used.

Equation (14) shows that the on-line EM algorithm, i.e., (10) together with (11), is equivalent to a stochastic gradient method with the inverse of the Fisher information matrix $\mathbf{V}^{-1}$ as a coefficient matrix. If the effective learning rate $\eta(\tau) (\geq 0)$ satisfies the condition

$$\sum_{\tau=1}^{\infty} \eta(\tau) = \infty \quad \textit{and} \quad \sum_{\tau=1}^{\infty} \eta^2(\tau) < \infty, \tag{15}$$

the stochastic approximation theory [5] guarantees that the estimator $\boldsymbol{\theta}(\boldsymbol{\phi}(\tau))$ converges to a local maximum of the likelihood function (12).

Equation (14) has a similar form as the natural gradient method proposed by Amari [2], which gives the optimal asymptotic convergence. However, there is an important difference. The Fisher information matrix in (14) is defined for the probability distribution of a complete event $P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}(\boldsymbol{\phi}))$. On the other hand, the Fisher information matrix in the natural gradient method is defined for the probability distribution of an observed event $P(\mathbf{x}|\boldsymbol{\theta}(\boldsymbol{\phi}))$. It is beyond the scope of the current paper to study the relationship between the on-line EM algorithm and the natural gradient method.

When there is no discount factor, i.e., $\lambda(\tau) = 1$, the effective learning rate $\eta(\tau)$ is given by $\eta(\tau) = 1/\tau$. This satisfies the stochastic approximation condition (15). However, the learning speed becomes very slow if this schedule is adopted (see Sec. 6). The reason for this slow convergence is that earlier inaccurate estimator values affect the estimations in later learning stages because there is no discount factor in the sufficient statistics average (8). The introduction of the discount factor is crucial for fast convergence.

We employ the following discount schedule

$$1 - \lambda(\tau) = \frac{1}{(\tau - 2)\kappa + \tau_0}, \tag{16}$$

which can be calculated recursively:

$$\lambda(\tau) = 1 - \frac{1 - \lambda(\tau - 1)}{1 + \kappa(1 - \lambda(\tau - 1))}.$$

The corresponding effective learning rate $\eta(\tau)$ satisfies

$$\eta(\tau) \underset{\tau\to\infty}{\longrightarrow} \left(\frac{\kappa + 1}{\kappa}\right) \frac{1}{\tau}, \tag{17}$$

so that the stochastic approximation condition (15) is satisfied. The constants appearing in (16) have clear physical meanings. $\tau_0$ represents how many samples contribute to the discounted average for the sufficient statistics (8) in the early stage of learning. $\kappa$ controls the asymptotic decreasing ratio for the effective learning constant $\eta(\tau)$ as in (17). The initial condition $\eta(1) = \eta_0$ implies that the initial value for the expectation parameter $\boldsymbol{\phi}(0)$ can be regarded as an average of $(1/\eta_0 - 1)$ data points. Therefore, the value of $\eta_0$ is determined according to the uncertainty of the initial estimator $\boldsymbol{\phi}(0)$. For $0 < \kappa < 1$, there are three phases which are characterized by different $\eta(\tau)$ behaviors. In the first phase, $1 \le \tau \le \tau_0$, $\eta(\tau)$ rapidly decreases to $1/\tau_0$ In this period, a rough estimate of the parameters takes place very quickly. This is the initialization phase. In the second phase, $\tau_0 \le \tau \le \tau_0/\kappa$, $\eta(\tau)$ are nearly equal to $1/\tau_0$. This period is the main part of the learning process, in which a fairly good estimator is obtained. In the third phase, $\tau_0/\kappa \le \tau$, $\eta(\tau)$ behaves like (17). This period stabilizes the estimator value by the $(1/\tau)$- schedule of $\eta(\tau)$.

# 6 Experiment

The performance of the on-line EM algorithm was examined by using two-dimensional Mixture of Gaussian (MG) models. The hidden variable in the MG model corresponds to the unit index of the Gaussians and have discrete values. Two training data sets consisting of 100 and 1000 points were generated from a mixture of four Gaussians. The centers and the variances of the model are shown in Fig.1. The mixing constant was given by $(P(1), ..., P(4)) = (4/9, 2/9, 2/9, 1/9)$. In order to measure the generalization error, the KL-divergence from the data generation model distribution to the trained MG model distribution was calculated. This was evaluated on a 51 x 51 point mesh grid.

The MG model was trained by using four algorithms: the on-line EM with discount factor (OEM), the on-line EM without discount factor (OEM-NO), the batch EM (EM) and the on-line gradient ascent algorithms (GRAD). The on-line gradient ascent algorithm is defined by

$$\Delta\boldsymbol{\theta}(\tau) = \eta(\tau) \left(\frac{\partial L(\boldsymbol{\theta}|\mathbf{x}(\tau))}{\partial\boldsymbol{\theta}}\right)|_{\boldsymbol{\theta}(\tau-1)}. \tag{18}$$

For each learning method and each data set, learning processes starting from 20 different initial conditions were performed. The initial conditions were generated as follows. First, the initial center positions were randomly generated from the region [0 1] x [0 1]. The initial variances for all the units were set to the variance of the training data. The mixing constants were set to the same value. The same set of initial conditions was used for all experiments.

In one epoch, each training algorithm sees all training data once. The batch EM algorithm updates the model parameters once at the end of each epoch, while the on-line algorithms update them for each datum. The computational costs in one epoch for four algorithms are on the same order. The batch EM algorithm can be considered a special type of the on-line EM algorithm, in which the model parameters are only updated at the end of each epoch, and the discount factor is given by $\lambda(\tau) = 0$ at the beginning of each epoch, or $\lambda(\tau) = 1$ otherwise. This fact also supports the assumption that the learning speed of each method can be compared in terms of number of epochs.

The discount schedule for the on-line EM algorithm was given by (16) with $\eta_0 = 0.5$, $\tau_0 = 100$, and $\kappa = 0.05$.

The learning curves for four algorithms are shown in Figs. 2 and 3. The learning speed of the on-line EM algorithm with the discount factor is faster than that of the batch EM algorithm. This is more prominent as the amount of training data increased. The final generalization errors of the both algorithms were almost the same except in some initial conditions, where the batch EM algorithm was trapped to a local maximum of the likelihood function. The reason for this result can be interpreted as follows. In the early stage of the batch EM learning, the same posterior probability employing an inaccurate estimator is used for all training data in one epoch. This may cause an inappropriate assignment of the units for the training data and result in a poor estimation of the new parameter. In the on-line EM algorithm, the posterior probability is calculated with the new parameter, which is updated each time data is presented. Therefore, the unit assignment gradually improves as more data are presented.

The learning speed of the on-line EM algorithm without the discount factor is very slow. This result implies that the introduction of the discount factor is crucial for a good performance of the on-line EM algorithm, as pointed out in Sec. 5. If there is no discount factor, the early inaccurate estimations contribute to the sufficient statistics average even in the later stages of the learning process and degrade the quality of estimations.

The learning performance of the on-line gradient ascent algorithm was very sensitive to the learning rate schedule. If the initial value for the learning rate $\eta_0$ was not smaller than 0.001, unstable oscillatory behavior occurred in the early stage of learning. The best performance was obtained by a learning rate schedule, $\eta(\tau) = 1/\{\gamma(\tau-1)+(1/\eta_0)\}$, with $\eta_0 = 0.0001$ and $\gamma = 0.1$.

The learning curves corresponding to the best performance are shown in Figs. 2 and 3. The figures shows that the performances of the on-line gradient ascent algorithm are worst among the four algorithms.

## 7 Conclusion

In this article, an on-line EM algorithm with a discount factor was derived for general Exponential Family models with Hidden variables (EFH models). It was proven that the on-line EM algorithm is equivalent to a stochastic gradient method with the inverse of the Fisher information matrix as a coefficient matrix. As a result, the stochastic approximation theory guarantees the convergence to a local maximum of the likelihood function.

The performance of the on-line EM algorithm was examined by using the mixture of Gaussian model. The simulation results showed that the on-line EM algorithm with the discount factor was faster than the batch EM algorithm and the on-line gradient ascent algorithm. The efficiency of the on-line EM algorithm over the batch EM algorithm became more prominent as the amount of training data increased. It was also shown that the introduction of the discount factor was crucial for fast convergence.

One of the advantages of our on-line EM algorithm is that the estimator at each time step can be represented by the discounted weighted mean $\ll \mathbf{r}(\mathbf{x}, \mathbf{z}) \gg (\tau)$ defined by Eq. (8). Therefore, the on-line EM algorithm gives a reasonable estimator even in the early stage of learning. Moreover, the learning rate schedule can be systematically designed and the fast learning schedule, in which $\eta_0 \sim o(1)$, is possible. This kind of fast learning schedule is not possible for the on-line gradient ascent algorithm because it causes unstable oscillatory learning behaviors.

## References


[1] Amari, S. (1995). Information Geometry of the EM and em Algorithms for Neural Networks. *Neural Networks*, **9**, 1379-1408.

[2] Amari, S. (1998), Natural Gradient Works Efficiently in Learning, *Neural Computation*, **10**, pp.251-276.

[3] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, **39**, 1-22.

[4] Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, **6**, 181-214.

[5] Kushner, H. J., & Yin, G. G. (1997). *Stochastic Approximation Algorithms and Applications*, New York: Springer-Verlag.

[6] Lowe, D. (1991). On the iterative version of RBF networks: a statistical interpretation. *2nd IEE International Conference on Artificial Neural Networks.* pp. 29-33. Institute of Electrical Engineers.

[7] Neal, R. M., & Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. To appear in M. I. Jordan (Ed.), *Learning in Graphical Models*, Kluwer Academic Press.

[8] Nowlan, S. J. (1991). Soft competitive adaptation: neural network learning algorithms based on fitting statistical mixtures. *CMU Technical Report*, **CS-91-126**, Pittsburgh: Carnegie Mellon University.

[9] Sato, M., & Ishii, S. (2000). On-line EM Algorithm for the Normalized Gaussian Network. *Neural Computation*.**12**, 407-432.

[10] Titterington, D.M. (1984). *Journal of the Royal Statistical Society, Series B*, **46**, 257-267.

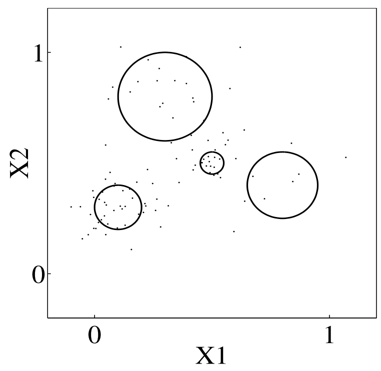


Figure 1: Centers and variances of the data generation model consisting of four Gaussian functions are shown by four circles. The center and radius of a circle correspond to the center and standard deviation of a Gaussian function, respectively. 100 training data points are also plotted.

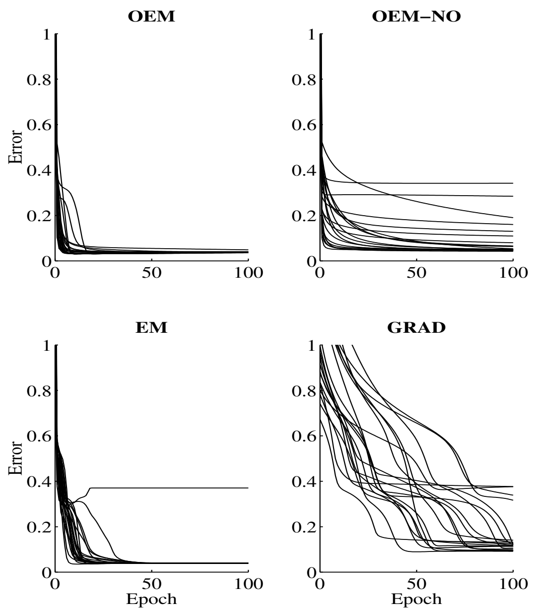


Figure 2: Learning curves for four algorithms are shown. They are the on-line EM with discount factor (OEM), the on-line EM without discount factor (OEM-NO), the batch EM (EM) and the on-line gradient ascent algorithms (GRAD). The numbers of units in the trained MG models are four. Ordinate denotes the generalization error measured by the KL-divergence from the data generation model distribution to the trained MG model probability distribution. Abscissa denotes the epoch number. Each figure plots 20 learning curves starting from 20 different initial conditions. The size of the training data is 100.

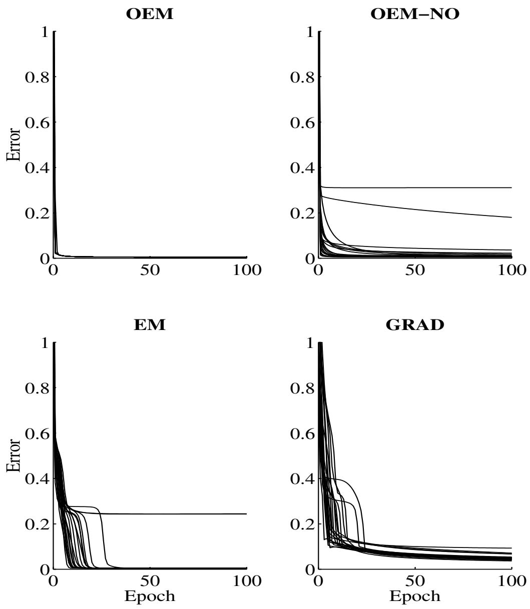


Figure 3: Learning curves for four algorithms are shown. The learning algorithms are the same as in Fig.2. The numbers of units in the trained MG models are four. The size of the training data is 1000.