IEICE TRANS. INF. & SYST., VOL.E91–D, NO.12 DECEMBER 2008

PAPER

Component Reduction for Gaussian Mixture Models

Kumiko MAEBASHI[†a)], ***Nonmember***, **Nobuo SUEMATSU**[†], ***and*** **Akira HAYASHI**[†], ***Members***

SUMMARY The mixture modeling framework is widely used in many applications. In this paper, we propose a *component reduction* technique, that collapses a Gaussian mixture model into a Gaussian mixture with fewer components. The EM (Expectation-Maximization) algorithm is usually used to fit a mixture model to data. Our algorithm is derived by extending mixture model learning using the EM-algorithm. In this extension, a difficulty arises from the fact that some crucial quantities cannot be evaluated analytically. We overcome this difficulty by introducing an effective approximation. The effectiveness of our algorithm is demonstrated by applying it to a simple synthetic component reduction task and a phoneme clustering problem.

key words: *mixture model, EM-algorithm, maximum likelihood, hierarchical clustering*

1. Introduction

Component reduction is the process whereby a mixture model is collapsed into a mixture with fewer components. Since mixture models are used in a wide variety of applications, component reduction techniques are becoming more important. As an example, consider the case where data is compressed and represented as a mixture model and the original data is lost. We might use a component reduction technique to analyze this data further. Moreover, by iterating the component reduction, hierarchical mixture models can be constructed in a bottom-up manner. The hierarchical mixture model is a useful tool for analyzing data at various levels of granularity [1].

Component reduction can be regarded as the process of fitting a mixture model to another mixture with more components. The EM-algorithm [2], [3] is broadly applied to fit a mixture model to a set of data points [4]. We devise a component reduction algorithm by extending this application of the EM-algorithm to the case in which a Gaussian mixture model is fitted to another Gaussian mixture with more components.

In deriving the algorithm, we first formulate the application of the EM-algorithm to component reduction. Although this formulation provides an EM-procedure, it cannot be performed in practice because some quantities needed in the EM-procedure cannot be calculated analytically. Therefore, we propose an approximated version of the EM-procedure.

The organization of this paper is as follows. Section 2 provides the background and our motivation for this study. The EM-algorithm is described in Sect. 3. In Sect. 4, we formulate the application of the EM-algorithm to component reduction and obtain an EM-procedure. Then, in Sect. 5, we derive an approximation of the EM-procedure. Section 6 discusses three related algorithms. In Sect. 7, we apply our proposed method and the three related methods to synthetic data and a phoneme clustering problem.

2. Background and Motivation

The EM-algorithm alternates between performing an expectation step (E-step) and a maximization step (M-step). The probabilities of assigning the data points to the components of the mixture are calculated in the E-step. These probabilities determine the responsibilities of the components in representing the data points. In the M-step, each of the component parameters is updated so that its likelihood for the data points, weighted by the responsibilities, is maximized.

A straight-forward approach to component reduction is to generate samples from the given mixture model, and then to apply the EM-algorithm to these samples. This is, however, computationally inefficient.

By simply replacing "the data points" with "the components of the original mixture" in the EM-algorithm, we can obtain the outline of a class of algorithms for fitting a mixture model to another mixture model. The existing component reduction algorithms [1], [5], [6] can be regarded as members of this class, and are described in Sect. 6.

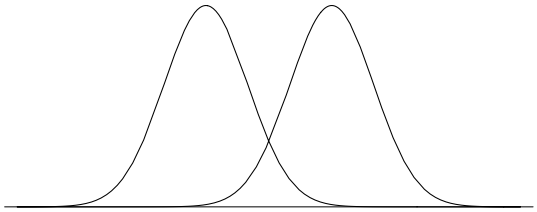
Since each of the components of the original mixture is spatially extended, unlike in the case of data points, the probabilities of properly assigning the original components to the components being fit, should be position dependent.

Existing members of the class of algorithms described above, such as [1], [5] and [6], do not adequately take this into account. To illustrate this problem, we consider the simple component reduction task shown in Fig. 1, in which we try to fit a two component mixture model to the three component mixture. When we consider the assignment of the original component in the middle, we should split it into two parts (illustrated by dashed lines) according to the spatial relationships of the two components of the fitted mixture. Each of the two parts should then be incorporated into its corresponding component. However, this splitting process cannot be done by the current algorithms in the above class.

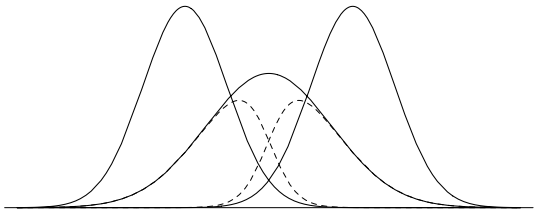
Manuscript received March 17, 2008.
Manuscript revised June 26, 2008.

[†]The authors are with the Graduate School of Information Sciences, Hiroshima City University, Hiroshima-shi, 731–3194 Japan.
a) E-mail: bochin@robotics.im.hiroshima-cu.ac.jp

DOI: 10.1093/ietisy/e91–d.12.2846



(a) Two component mixture model



(b) Three component mixture model

Fig. 1 An example of the fitting problem.

In this paper, we devise a component reduction algorithm which overcomes this limitation.

3. Fitting Mixture Models to Data

We have devised a component reduction algorithm based on the application of the EM-algorithm for fitting mixture models to data. First, we review the application formulated by Dempster [2].

Let us consider approximating a data distribution with the mixture model,

$$f_\Theta(\boldsymbol{x}) = \sum_{j=1}^{C} \pi_j p(\boldsymbol{x}|\theta_j), \tag{1}$$

where C is the number of mixture components, $p(\boldsymbol{x}|\theta_j)$ is the probability density with parameter vector θ_j, π_j is a nonnegative quantity such that for $j = 1, \ldots, C$, $0 \le \pi_j \le 1$ and $\sum_{j=1}^{C} \pi_j = 1$, and $\Theta = \{\pi_1, \ldots, \pi_C, \theta_1, \ldots, \theta_C\}$ is the set of all the parameters in the mixture model.

Given a set of data points, $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, when we apply the EM-algorithm, it is assumed that each data point $\boldsymbol{x}_i$ has been drawn from one of the components of the mixture model. Then, we introduce unobservable vectors $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iC})$ indicating the component from which $\boldsymbol{x}_i$ was drawn: where for every j, y_{ij} is 1 if $\boldsymbol{x}_i$ was drawn from the j-th component and 0 otherwise. Let $\mathcal{Y} = \{y_{ij} | i = 1, \ldots, N, j = 1, \ldots, C\}$. The log-likelihood of Θ for the complete data $(\mathcal{X}, \mathcal{Y})$ is given by

$$L(\Theta|\mathcal{X}, \mathcal{Y}) = \sum_{i=1}^{N} \sum_{j=1}^{C} y_{ij} \log\{\pi_j p(\boldsymbol{x}_i|\theta_j)\}. \tag{2}$$

Since $\mathcal{Y}$ is unobservable, we take the expectation of the log-likelihood with respect to $\mathcal{Y}$ under the given observed data $\mathcal{X}$ and the current estimate $\Theta^{(t)}$. The expected value of the log-likelihood is

$$\begin{aligned} Q(\Theta|\Theta^{(t)}) &= E[L(\Theta|\mathcal{X}, \mathcal{Y}) \mid \mathcal{X}, \Theta^{(t)}] \\ &= \sum_{i=1}^{N} \sum_{j=1}^{C} h_{ij} \log\{\pi_j p_j(\boldsymbol{x}_i|\theta_j)\}, \end{aligned} \tag{3}$$

where $h_{ij} = E[y_{ij} \mid \boldsymbol{x}_i, \Theta^{(t)}]$.

Starting with an initial guess $\Theta^{(0)}$, the EM-algorithm generates successive estimates, $\Theta^{(1)}, \Theta^{(2)}, \ldots$, by iterating the following E- and M-steps:

E-step: Compute $\{h_{ij}^{(t)}\}$, using current estimate $\Theta^{(t)}$.
M-step: Set $\Theta^{(t+1)} = \Theta$ which maximizes $Q(\Theta|\Theta^{(t)})$ given $\{h_{ij}^{(t)}\}$.

The iteration is terminated when the sequence of estimates converges.

4. Fitting a Mixture Model to Another Mixture Model

In this section, we formulate a straight-forward application of the EM-algorithm to fit a mixture model to another mixture. We elucidate that it is difficult to perform the iterative procedure provided by the formulation because it requires the evaluation of integrals which cannot be solved analytically.

The task is described as fitting the U-component mixture model $f_{\Theta_U}(\boldsymbol{x})$ to the given L-component mixture model $f_{\Theta_L}(\boldsymbol{x})$, where $L > U$,

$$f_{\Theta_U}(\boldsymbol{x}) = \sum_{j=1}^{U} \pi_j^U p(\boldsymbol{x}|\theta_j^U), \tag{4}$$

$$f_{\Theta_L}(\boldsymbol{x}) = \sum_{i=1}^{L} \pi_i^L p(\boldsymbol{x}|\theta_i^L). \tag{5}$$

We now introduce a random vector $\boldsymbol{y} = (y_1, \ldots, y_U)$ corresponding to the unobservable vectors $\boldsymbol{y}_i$ in Section 3, where y_j are binary variables drawn according to the conditional probability distributions,

$$\Pr(y_j = 1|\boldsymbol{x}, \Theta_U) = \frac{\pi_j^U p(\boldsymbol{x}|\theta_j^U)}{\sum_{j'=1}^{U} \pi_{j'}^U p(\boldsymbol{x}|\theta_{j'}^U)}. \tag{6}$$

Then, the log-likelihood of Θ_U for $(\boldsymbol{x}, \boldsymbol{y})$ is

$$L(\Theta_U|\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{U} y_j \log\{\pi_j^U p(\boldsymbol{x}|\theta_j^U)\}, \tag{7}$$

and the counterpart of $Q(\Theta|\Theta^{(t)})$ in (3) is defined by taking the expectation of the log-likelihood with respect to $\boldsymbol{x}$ with distribution $f_{\Theta_L}(\boldsymbol{x})$ as

$$\begin{aligned} &Q_{\text{hier}}(\Theta_U|\Theta_U^{(t)}) \\ &\quad = E_{\boldsymbol{x}}\{E_{\boldsymbol{y}}\{L(\Theta_U|\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{x}, \Theta_U^{(t)}\} \mid \Theta_L\}, \\ &\quad = \sum_{j=1}^{U} \sum_{i=1}^{L} \pi_i^L \int p(\boldsymbol{x}|\theta_i^L) h_j(\boldsymbol{x}) \log\{\pi_j p(\boldsymbol{x}|\theta_j^U)\} d\boldsymbol{x}, \end{aligned} \tag{8}$$

where $h_j(\boldsymbol{x}) = \Pr(y_j = 1|\boldsymbol{x}, \Theta_U^{(t)})$.

To derive the E- and M-steps, we introduce another random vector $\boldsymbol{z} = (z_1, \ldots, z_L)$ which indicates the component of the original mixture model from which $\boldsymbol{x}$ is drawn, where z_i are binary variables whose (marginal) probability distributions are given by $\Pr(z_i = 1) = \pi_i^L$. Then, using Bayes' rule, we obtain the following relation:

$$\Pr(\boldsymbol{x}|z_i=1, y_j=1) = \frac{\Pr(y_j=1|\boldsymbol{x}, z_i=1)\Pr(\boldsymbol{x}|z_i=1)}{\Pr(y_j=1|z_i=1)}. \quad (9)$$

From $\Pr(y_j = 1|\boldsymbol{x}, z_i = 1) = \Pr(y_j = 1|\boldsymbol{x}) = h_j(\boldsymbol{x})$ and $\Pr(\boldsymbol{x}|z_i = 1) = p(\boldsymbol{x}|\theta_i^L)$, by denoting $\Pr(\boldsymbol{x}|z_i = 1, y_j = 1)$ as $p(\boldsymbol{x}|i, j)$, (9) can be rewritten as

$$p(\boldsymbol{x}|i, j) = \frac{h_j(\boldsymbol{x})p(\boldsymbol{x}|\theta_i^L)}{h_{ij}}, \quad (10)$$

where $h_{ij} = \Pr(y_j = 1|z_i = 1)$. By substituting (10) into (8), we obtain

$$Q_{\text{hier}}(\Theta_U|\Theta_U^{(t)}) = \sum_{j=1}^{U}\sum_{i=1}^{L}\pi_i^L h_{ij}\int p(\boldsymbol{x}|i, j)\log\{\pi_j p(\boldsymbol{x}|\theta_j^U)\}d\boldsymbol{x}. \quad (11)$$

Although we cannot perform them in practice, we can define the E- and M-steps based simply on (11) as follows:

E-step: Compute $\{p^{(t)}(\boldsymbol{x}|i, j)\}$ and $\{h_{ij}^{(t)}\}$ with the current estimate $\Theta_U^{(t)}$.
M-step: Set $\Theta_U^{(t+1)} = \arg\max_{\Theta_U} Q_{\text{hier}}(\Theta_U|\Theta_U^{(t)})$ given $p^{(t)}(\boldsymbol{x}|i, j)$ and $h_{ij}^{(t)}$.

Since both of these steps involve integrals which cannot be evaluated analytically, we cannot carry them out (without numerical integrations).

5. Component Reduction Algorithm

From now on, we focus our discussion on Gaussian mixture models. Let, $p(\boldsymbol{x}|\theta_i^L)$ and $p(\boldsymbol{x}|\theta_j^U)$ be Gaussians where $\theta_i^L = (\boldsymbol{\mu}_i^L, \Sigma_i^L)$ and $\theta_j^U = (\boldsymbol{\mu}_j^U, \Sigma_j^U)$. Then, we introduce an approximation which enables us to perform the EM-procedure derived in Sect. 4.

5.1 Update Equations in the M-Step

Without any approximation, the parameter set Θ_U which maximizes $Q_{\text{hier}}(\Theta_U|\Theta_U^{(t)})$ given $p^{(t)}(\boldsymbol{x}|i, j)$ and $h_{ij}^{(t)}$ is obtained from

$$\pi_j^U = \sum_{i=1}^{L}\pi_i^L h_{ij}^{(t)}, \quad (12)$$

$$\boldsymbol{\mu}_j^U = \frac{\sum_{i=1}^{L}\pi_i^L h_{ij}^{(t)}\boldsymbol{\mu}_{ij}^{(t)}}{\sum_{i=1}^{L}\pi_i^L h_{ij}^{(t)}}, \quad (13)$$

$$\Sigma_j^U = \frac{\sum_{i=1}^{L}\pi_i^L h_{ij}^{(t)}\{\Sigma_{ij}^{(t)} + (\boldsymbol{\mu}_{ij}^{(t)} - \boldsymbol{\mu}_j^U)(\boldsymbol{\mu}_{ij}^{(t)} - \boldsymbol{\mu}_j^U)^{\mathrm{T}}\}}{\sum_{i=1}^{L}\pi_i^L h_{ij}^{(t)}}, \quad (14)$$

where for every i, j, $\boldsymbol{\mu}_{ij}^{(t)}$ and $\Sigma_{ij}^{(t)}$ are the mean vector and covariance matrix, respectively, of $p^{(t)}(\boldsymbol{x}|i, j)$.

From (10), $p(\boldsymbol{x}|i, j) \propto h_j(\boldsymbol{x})p(\boldsymbol{x}|\theta_i^L)$ holds and we have the analytical forms of $h_j(\boldsymbol{x})$ and $p(\boldsymbol{x}|\theta_i^L)$. Let $q_{ij}(\boldsymbol{x}) = h_j(\boldsymbol{x})p(\boldsymbol{x}|\theta_i^L)$ for convenience. The difficulty stems from the fact that the integrals, $\int q_{ij}(\boldsymbol{x})d\boldsymbol{x}$, $\int \boldsymbol{x}q_{ij}(\boldsymbol{x})d\boldsymbol{x}$, and $\int \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}q_{ij}(\boldsymbol{x})d\boldsymbol{x}$, cannot be solved analytically. Therefore, we cannot calculate the means and covariances of $p(\boldsymbol{x}|i, j)$. So, we introduce an approximation of $p^{(t)}(\boldsymbol{x}|i, j)$ using a Gaussian distribution.

5.2 Approximation

We are now in a position to construct the Gaussian approximation of $p(\boldsymbol{x}|i, j)$, that is, to obtain $\hat{\boldsymbol{\mu}}_{ij}$ and $\hat{\Sigma}_{ij}$ such that $p(\boldsymbol{x}|i, j) \simeq N(\boldsymbol{x}|\hat{\boldsymbol{\mu}}_{ij}, \hat{\Sigma}_{ij})$, where $N(\boldsymbol{x}|\hat{\boldsymbol{\mu}}_{ij}, \hat{\Sigma}_{ij})$ is the Gaussian pdf. The mean and covariance are approximated as follows.

We set $\hat{\boldsymbol{\mu}}_{ij} = \arg\max_{\boldsymbol{x}} q_{ij}(\boldsymbol{x})$. While $\arg\max_{\boldsymbol{x}} q_{ij}(\boldsymbol{x})$ cannot be represented in analytical form, it can effectively be obtained from the solution of

$$\frac{\partial q_{ij}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \boldsymbol{0}, \quad (15)$$

using the Newton-Raphson method starting from a carefully chosen point.

On the other hand, each $\hat{\Sigma}_{ij}$ is estimated using the relation

$$-\frac{1}{N(\boldsymbol{\mu}, \Sigma)}\left.\frac{\partial^2 N(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)}{\partial \boldsymbol{x}^2}\right|_{\boldsymbol{x}=\boldsymbol{\mu}} = \Sigma^{-1}. \quad (16)$$

We are constructing an approximation of $p(\boldsymbol{x}|i, j)$ using the Gaussian distribution $N(\boldsymbol{x}|\hat{\boldsymbol{\mu}}_{ij}, \hat{\Sigma}_{ij})$, and hence a natural choice is

$$\begin{aligned}\hat{\Sigma}_{ij}^{-1} &= -\frac{1}{p(\hat{\boldsymbol{\mu}}_{ij}|i, j)}\left.\frac{\partial^2 p(\boldsymbol{x}|i, j)}{\partial \boldsymbol{x}^2}\right|_{\boldsymbol{x}=\hat{\boldsymbol{\mu}}_{ij}} \\ &= -\frac{1}{q_{ij}(\hat{\boldsymbol{\mu}}_{ij})}\left.\frac{\partial^2 q_{ij}(\boldsymbol{x})}{\partial \boldsymbol{x}^2}\right|_{\boldsymbol{x}=\hat{\boldsymbol{\mu}}_{ij}} \\ &= (\Sigma_i^L)^{-1} + (\Sigma_j^U)^{-1} - \sum_{j'=1}^{U} h_{j'}(\hat{\boldsymbol{\mu}}_{ij})(\Sigma_{j'}^U)^{-1} \\ &\quad + \sum_{k=1}^{U} h_k(\hat{\boldsymbol{\mu}}_{ij})(\Sigma_k^U)^{-1}(\hat{\boldsymbol{\mu}}_{ij} - \boldsymbol{\mu}_k^U)(\hat{\boldsymbol{\mu}}_{ij} - \boldsymbol{\mu}_k^U)^{\mathrm{T}}(\Sigma_k^U)^{-1} \\ &\quad - \sum_{k=1}^{U}\sum_{l=1}^{U} h_k(\hat{\boldsymbol{\mu}}_{ij})h_l(\hat{\boldsymbol{\mu}}_{ij}) \\ &\qquad \cdot (\Sigma_k^U)^{-1}(\hat{\boldsymbol{\mu}}_{ij} - \boldsymbol{\mu}_k^U)(\hat{\boldsymbol{\mu}}_{ij} - \boldsymbol{\mu}_l^U)^{\mathrm{T}}(\Sigma_l^U)^{-1}. \end{aligned} \quad (17)$$

To complete the E-step, we also need to evaluate h_{ij}. From (10), we have

$$h_{ij} = \frac{h_j(\boldsymbol{x})p(\boldsymbol{x}|\theta_i^L)}{p(\boldsymbol{x}|i, j)}, \quad (18)$$

for any $\boldsymbol{x}$. With the approximation, $p(\boldsymbol{x}|i, j) \simeq N(\boldsymbol{x}|\hat{\boldsymbol{\mu}}_{ij}, \hat{\Sigma}_{ij})$,

substituting $\boldsymbol{x} = \hat{\boldsymbol{\mu}}_{ij}$ yields the approximation of h_{ij},

$$\hat{h}_{ij} \propto \frac{h_j(\hat{\boldsymbol{\mu}}_{ij}) p(\hat{\boldsymbol{\mu}}_{ij}|\theta_i^L)}{N(\hat{\boldsymbol{\mu}}_{ij}|\hat{\boldsymbol{\mu}}_{ij}, \hat{\Sigma}_{ij})}. \tag{19}$$

5.3 Approximated EM-Procedure

Here we summarize the EM-procedure including the approximation described in the previous subsection. Setting the number of components U, and starting from some initial estimate $\Theta_U^{(0)}$, the procedure repeatedly alternates the following E- and M-steps:

E-step: With the current estimate $\Theta_U^{(t)}$,

1. Set $\{\hat{\boldsymbol{\mu}}_{ij}^{(t)}\}$ by solving (15) using the Newton-Raphson method.
2. Calculate $\{\hat{\Sigma}_{ij}^{(t)}\}$ using (17).
3. Calculate $\{\hat{h}_{ij}^{(t)}\}$ using (19) and normalize the values such that $\sum_{j=1}^{U} \hat{h}_{ij}^{(t)} = 1$.

M-step: Set $\Theta_U^{(t+1)} = \Theta_U$ where Θ_U is calculated by (12) with $\{\hat{\boldsymbol{\mu}}_{ij}^{(t)}\}$, $\{\hat{\Sigma}_{ij}^{(t)}\}$, and $\{\hat{h}_{ij}^{(t)}\}$.

After a number of iterations, some mixing rates of the components may converge to very small values. When this happens, the components with these small mixing rates are removed from the mixture model. As a result, the number of components can sometimes be less than U.

6. Related Work

6.1 Vasconcelos and Lippman (1999)

A component reduction algorithm has been developed using the notion of virtual samples [1]. The following are the E- and M-steps derived for Gaussian mixture models:

E-step: With the current estimate $\Theta_U^{(t)}$, compute $\{h_{ij}^{(t)}\}$ by

$$h_{ij}^{(t)} = \frac{\pi_j^U [p(\boldsymbol{\mu}_i^L|\theta_j^U) e^{-\frac{1}{2}\mathrm{trace}\{(\Sigma_j^U)^{-1}\Sigma_i^L\}}]^{\pi_i^L N}}{\sum_k \pi_k^U [p(\boldsymbol{\mu}_i^L|\theta_j^U) e^{-\frac{1}{2}\mathrm{trace}\{(\Sigma_k^U)^{-1}\Sigma_i^L\}}]^{\pi_i^L N}}, \tag{20}$$

where N denotes the virtual sample size drawn from the given mixture model.

M-step: Set the next estimate $\Theta_U^{(t+1)} = \Theta_U$ where elements in Θ_U are given by

$$\pi_j^U = \frac{\sum_{i=1}^{L} h_{ij}^{(t)}}{L}, \tag{21}$$

$$\boldsymbol{\mu}_j^U = \frac{\sum_{i=1}^{L} \pi_i^L h_{ij}^{(t)} \boldsymbol{\mu}_i^L}{\sum_{i=1}^{L} \pi_i^L h_{ij}^{(t)}}, \tag{22}$$

$$\Sigma_j^U = \frac{\sum_{i=1}^{L} \pi_i^L h_{ij}^{(t)} \{\Sigma_i^L + (\boldsymbol{\mu}_i^L - \boldsymbol{\mu}_j^U)(\boldsymbol{\mu}_i^L - \boldsymbol{\mu}_j^U)^{\mathrm{T}}\}}{\sum_{i=1}^{L} \pi_i^L h_{ij}^{(t)}}. \tag{23}$$

We can consider h_{ij} to represents the rate of contribution from the i-th component in the given mixture to the j-th component in the fitted mixture. Then, since $\sum_j h_{ij} = 1$, the update rule (21) means that the contribution of each component in the given mixture is the same, despite the expectation that the contributions are proportional to the mixing rates of the components in the given mixture. Thus, it is natural to conclude that the algorithm has an important defect when it is used as a general component reduction technique.

6.2 Goldberger and Roweis (2005)

The algorithm proposed in [5] assigns each component in the given mixture to one of U components in the fitted mixture, such that the KL-divergence between them is minimized. In other words, the algorithm involves hard clustering of the L components into U groups corresponding to components in the fitted mixture. On the contrary, the algorithm proposed in [1] and [6] use soft clustering of the components in the given mixture.

This algorithm introduces a new measure of the differences between two mixture models, defined as

$$d(\Theta_L, \Theta_U, \boldsymbol{\lambda}) = \sum_{i=1}^{L} \pi_i^L \mathrm{KL}[p(\boldsymbol{x}|\theta_i^L) \| p(\boldsymbol{x}|\theta_{\lambda_i}^U)], \tag{24}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_L)$ and λ_i denotes the index of the assigned component in the fitted mixture from the i-th component in the given mixture. Then, Θ^U is fitted by minimizing $d(\Theta_L, \Theta_U, \boldsymbol{\lambda})$ over all possible $(\Theta_U, \boldsymbol{\lambda})$. In addition, a procedure has been defined to find the local minima, which involves repeating the following two steps:

REGROUP: $\boldsymbol{\lambda}^{(t)} = \arg\min_{\boldsymbol{\lambda}} d(\Theta_L, \Theta_U^{(t)}, \boldsymbol{\lambda})$

REFIT: $\Theta_U^{(t+1)} = \arg\min_{\Theta_U} d(\Theta_L, \Theta_U, \boldsymbol{\lambda}^{(t)})$

As has been mentioned in [5], this algorithm can be regarded as a generalization of the k-means algorithm, and uses hard assignment in the same way. Therefore, the flexibility of the resulting models is rather restricted, although the algorithm is efficient in terms of computational cost. In addition, the algorithm might be heavily dependent on initial guesses as is the k-means algorithm.

6.3 Maebashi, Suematsu and Hayashi (2008)

Another component reduction algorithm has previously been proposed by the authors [6]. The algorithm is derived by extending mixture model learning using the EM-algorithm, as in the algorithm proposed in Sect. 5. As mentioned in Sect. 4, we cannot perform the extended EM-algorithm without some approximation. The approximation used in this algorithm is

$$p(\boldsymbol{x}|i, j) \simeq p(\boldsymbol{x}|\theta_i^L). \tag{25}$$

This approximation means that the information concerning

$\boldsymbol{x}$ carried by j is ignored when $\boldsymbol{x}$ generated from the i-th component in the given mixture is assigned to the j-th component in the fitted mixture.

Under this approximation, it is still not possible to calculate the assignment probabilities h_{ij}. To obtain h_{ij} we apply the interpretation of the EM-algorithm proposed by Neal [7], in which the E-step is also a maximization procedure. Thus, h_{ij} are parameters to be determined by solving the maximization problem in the E-step. This interpretation uses an analogy in physics and a temperature parameter is introduced based on this analogy [6]. The following are the E- and M-steps derived for Gaussian mixture models:

E-step: With the current estimate $\Theta_U^{(t)}$, compute $\{h_{ij}^{(t)}\}$ by

$$h_{ij}^{(t)} = \frac{[\pi_j^U p(\boldsymbol{\mu}_i^L|\theta_j^U) e^{-\frac{1}{2}\mathrm{trace}\{(\Sigma_j^U)^{-1}\Sigma_i^L\}}]^\beta}{\sum_k [\pi_k^U p(\boldsymbol{\mu}_i^L|\theta_j^U) e^{-\frac{1}{2}\mathrm{trace}\{(\Sigma_k^U)^{-1}\Sigma_i^L\}}]^\beta}, \tag{26}$$

where β denotes the inverse of the temperature parameter.
M-step: Set the next estimate $\Theta_U^{(t+1)} = \Theta_U$ where elements in Θ_U are given by

$$\pi_j^U = \sum_{i=1}^{L} \pi_i^L h_{ij}^{(t)}, \tag{27}$$

$$\boldsymbol{\mu}_j^U = \frac{\sum_{i=1}^{L} \pi_i^L h_{ij}^{(t)} \boldsymbol{\mu}_i^L}{\sum_{i=1}^{L} \pi_i^L h_{ij}^{(t)}}, \tag{28}$$

$$\Sigma_j^U = \frac{\sum_{i=1}^{L} \pi_i^L h_{ij}^{(t)} \{\Sigma_i^L + (\boldsymbol{\mu}_i^L - \boldsymbol{\mu}_j^U)(\boldsymbol{\mu}_i^L - \boldsymbol{\mu}_j^U)^{\mathrm{T}}\}}{\sum_{i=1}^{L} \pi_i^L h_{ij}^{(t)}}. \tag{29}$$

Since $h_{ij} \in (0, 1)$, the algorithm achieves a soft clustering of components in the given mixture model. However, when the temperature parameter is small, the algorithm behaves almost like a hard clustering algorithm.

6.4 Difference Between the Proposed Algorithm and the Existing Ones

As we have explained in Sect. 2, all of the three algorithms described in Sects. 6.1-6.3 update the j-th component of the fitted mixture so that it optimally represents the weighted components of the original mixture model where the weights of the original components are calculated for the j-th component based on the current estimate. The differences among them lie in the way of calculating the weights.

This common framework shared by the three algorithms restricts what they can achieve. We can rewrite (11) as

$$\begin{aligned} &Q_{\mathrm{hier}}(\Theta_U|\Theta_U^{(t)}) \\ &= \sum_{j=1}^{U} \pi_\cdot^j \int \left[\sum_{i=1}^{L} \pi_i^j p(\boldsymbol{x}|i,j)\right] \log p(\boldsymbol{x}|\theta_j^U) d\boldsymbol{x} \\ &\quad + \sum_{j=1}^{U} \pi_\cdot^j \log \pi_j^U \end{aligned} \tag{30}$$

where $\pi_\cdot^j = \sum_i \pi_i^L h_{ij}$ and $\pi_i^j = \pi_i^L h_{ij}/\pi_\cdot^j$. From the above equation, we can see that when we apply the EM algorithm to the component reduction problem, the j-th component of the fitted mixture has to be chosen so that the KL-divergence of $p(\boldsymbol{x}|\theta_j^U)$ from the mixture model $\sum_i \pi_i^j p(\boldsymbol{x}|i,j)$ is minimized. This fact means that the existing three algorithms, in a sense, approximate $p(\boldsymbol{x}|i,j)$ by $p(\boldsymbol{x}|\theta_i^L)$ for all j, which would result in a very poor approximation as we can see in Fig. 1, in which the dashed lines are plots of the functions proportional to $p(\boldsymbol{x}|i=2, j=1)$ and $p(\boldsymbol{x}|i=2, j=2)$.

On the other hand, our algorithm approximates each $p(\boldsymbol{x}|i,j)$, respectively, and hence we can expect that it can achieve higher accuracy than the existing three algorithms.

7. Experimental Results

To demonstrate the effectiveness of our algorithm, we conducted two experiments. For convenience, we refer to the proposed algorithm as the CREM (Component Reduction based on EM-algorithm) and to our previously-proposed algorithm [6], and the algorithms proposed by Vasconcelos and Lippman [1] and Goldberger and Roweis [5] as CREM0, VL and GR, respectively.

7.1 Synthetic Data

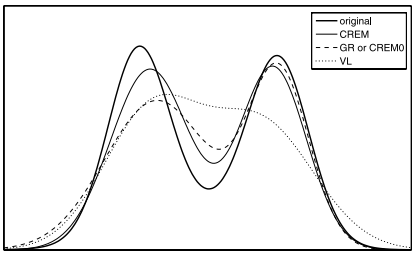
This experiment is intended to verify the effectiveness of our algorithm in component reduction problems similar to the example described in Sect. 2. The experimental procedure is as follows.

1. Draw 500 data points from the 1-dimensional 2-component Gaussian mixture model
$$f_{\Theta_{true}}(x) = \frac{1}{2} \cdot N(x|-2, 1) + \frac{1}{2} \cdot N(x|2, 1). \tag{31}$$
2. Learn a three component model using the standard EM-algorithm, starting from $f(x) = 1/3 \cdot N(x|-2, 1) + 1/3 \cdot N(x|0, 1) + 1/3 \cdot N(x|2, 1)$.
3. Reduce the three-component model obtained in the previous step to a two component mixture using CREM, CREM0, VL, GR and the standard EM, where the initial estimate is determined as
$$\begin{aligned} &f_{\Theta_U}(x) \\ &= \pi_1^U \cdot col\left[\frac{1}{\pi_1^U}\left\{\pi_1^L N(x|\mu_1, \sigma_1^2) + \frac{\pi_2^L}{2} N(x|\mu_2, \sigma_2^2)\right\}\right] \\ &\quad + \pi_2^U \cdot col\left[\frac{1}{\pi_2^U}\left\{\frac{\pi_2^L}{2} N(x|\mu_2, \sigma_2^2) + \pi_3^L N(x|\mu_3, \sigma_3^2)\right\}\right], \end{aligned} \tag{32}$$
where $\pi_1^U = \pi_1^L + \pi_2^L/2$, $\pi_2^U = \pi_2^L/2 + \pi_3^L$ and $col[g]$ denotes the Gaussian that has the minimum KL-divergence from g.

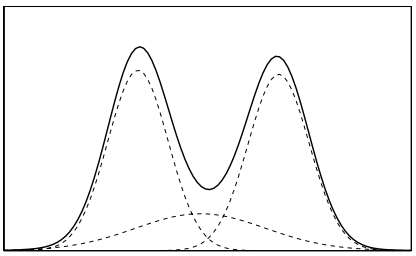
The experiment was repeated 100 times. The parameters, β in CREM0 and N in VL, were empirically tuned to optimize their performances, with actual values of $\beta = 10^5$ and

Table 1 Average KL-divergence and standard deviation.

	$KL(f_{\Theta_L} \| f_{\Theta_U})$	$KL(f_{EM} \| f_{\Theta_U})$	$KL(f_{true} \| f_{\Theta_U})$
CREM	$1.048 \times 10^{-2} (\pm 1.06 \times 10^{-5})$	$1.057 \times 10^{-2} (\pm 1.05 \times 10^{-5})$	$1.057 \times 10^{-2} (\pm 1.05 \times 10^{-5})$
CREM0	$3.585 \times 10^{-2} (\pm 1.27 \times 10^{-4})$	$3.931 \times 10^{-2} (\pm 1.63 \times 10^{-4})$	$4.367 \times 10^{-2} (\pm 8.13 \times 10^{-5})$
GR	$3.587 \times 10^{-2} (\pm 1.29 \times 10^{-4})$	$3.900 \times 10^{-2} (\pm 1.54 \times 10^{-4})$	$4.413 \times 10^{-2} (\pm 1.00 \times 10^{-4})$
VL	$8.052 \times 10^{-2} (\pm 4.15 \times 10^{-4})$	$8.330 \times 10^{-2} (\pm 4.74 \times 10^{-4})$	$8.091 \times 10^{-2} (\pm 5.32 \times 10^{-4})$



(a) Pdf of f_{Θ_U}



(b) Pdf of f_{Θ_L}

Fig. 2 Three and two component mixture model.

$N = 3$. Also, in this experiment, the EM-procedure was terminated when $\max_{i,j}(h_{ij}^{(t)} - h_{ij}^{(t-1)}) < 10^{-5}$. We evaluated the results using the KL-divergence, calculated using numerical integration. Table 1 shows the averages taken over the 100 trials. The results for CREM are the best of all the results. One of the trials is shown in Fig. 2. Figure 2 (a) is a plot of the pdfs obtained from CREM, GR, VL, and CREM0 for the original 3-component mixture shown in Fig. 2 (b). Note that the pdfs obtained from CREM0 and GR are represented by a curve since they are very close to each other and are indistinguishable in the plot. We can see that the pdf obtained from CREM is closest to the original pdf.

7.2 TIMIT Phoneme Recognition

We also applied the four algorithms to clustering the phoneme dataset described in [8]. The dataset contains 5 phoneme classes of 4,509 instances described by log-periodograms of length 256. The dimension of the instance is reduced to 10 dimensions using PCA and 5-layered hierarchical mixture models are constructed according to the structure shown in Fig. 3. The bottom (zero'th) level corresponds to 4,509 data points.

We construct hierarchical mixture models and measure the quality of the mixture model in each layer in terms of clustering quality. If a component reduction is performed properly, the dominant cluster structure that a lower layer has will be inherited by the upper layer, which must have high clustering quality. In each trial of the four algorithms, a 50-component mixture model in the first level is learned using the standard EM-algorithm. The second and higher levels are obtained by applying each component reduction algorithm to the lower levels. To compare these algorithms with the standard EM-algorithm, 20, 10, and 5-component mixtures are learned from the data points using the standard EM-algorithm. Since all four algorithms depend on initial guesses $\Theta_U^{(0)}$, we repeated the experiment 10 times. In this experiment, initial guesses $\Theta_U^{(0)}$ are obtained by picking up the components of the U largest mixing rates from the L components of the lower mixture. The terminal conditions for the four algorithms were empirically tuned to ensure convergence. Consequently, in this experiment, the EM-procedure was terminated when $\max_{i,j}(h_{ij}^{(t)} - h_{ij}^{(t-1)}) < 10^{-5}$. Also, in CREM0 the inverse of the temperature parameter is $\beta = \{1, 10^5\}$ and in VL the number of virtual samples is $N = 4509$, which is the number of instances.

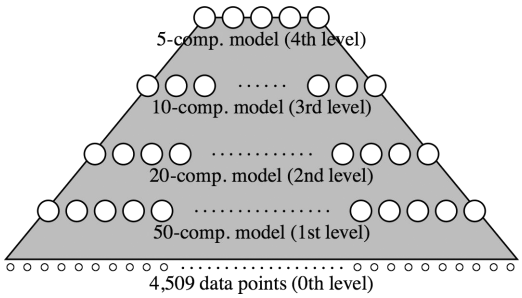


Fig. 3 Structure of constructed hierarchical mixture models in the experiment.

We evaluated the clustering results in terms of error rate and NMI(normalized mutual information) [9]. Let $\boldsymbol{\lambda}^{(c)}$ be the correct class labeling with 5 labels provided in the dataset and $\boldsymbol{\lambda}^{(e)}$ be the cluster labeling with U labels representing a clustering result. For every $n = 1, \ldots, 4059$, the estimated cluster label is defined by

$$\lambda_n^{(e)} = \operatorname*{argmax}_j(\{\pi_j p(\boldsymbol{x}_n|\theta_j) | j = 1, \ldots, U\}). \tag{33}$$

The error rate is defined by

$$\phi^{ER}\left(\boldsymbol{\lambda}^{(c)}, \boldsymbol{\lambda}^{(e)}\right) = 1 - \frac{1}{N}\sum_{j=1}^{U} \max_i n_{c,e}(i, j), \tag{34}$$

where $n_{c,e}(i, j)$ denotes the number of samples that have a class label i according to $\boldsymbol{\lambda}^{(c)}$ as well as a cluster label j

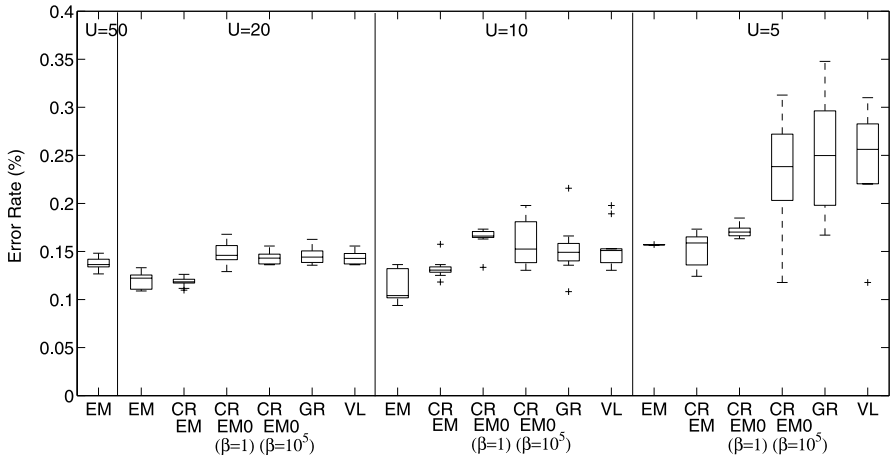


Fig. 4 Boxplot of the error rate for 10 trials.

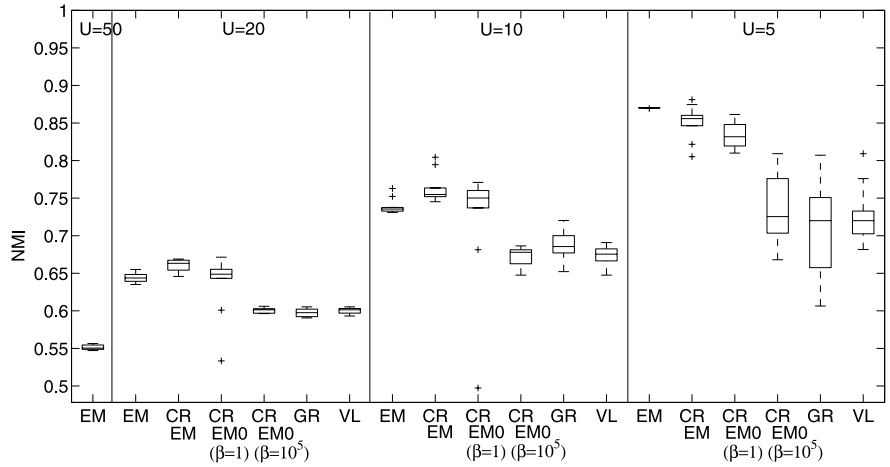


Fig. 5 Boxplot of the NMI for 10 trials.

according to $\boldsymbol{\lambda}^{(e)}$.

Figure 4 shows a boxplot of the error rate. Each box has horizontal lines at the lower quartile, median, and upper quartile. Whiskers extend to the adjacent values within 1.5 times the interquartile range from the ends of the box and + signs indicate outliers.

From Fig. 4, at all the levels, we can confirm that CREM has an advantage at all levels over the other three algorithms in terms of error rate.

The NMI is based on the mutual information and ranges from 0 to 1. A higher NMI indicates that the clustering is more informative. For $\boldsymbol{\lambda}^{(c)}$ and $\boldsymbol{\lambda}^{(e)}$, the NMI is estimated from

$$\phi^{NMI}(\boldsymbol{\lambda}^{(e)}, \boldsymbol{\lambda}^{(c)}) = \frac{\sum_{i=1}^{5}\sum_{j=1}^{U} n_{c,e}(i,j)\log\frac{n_{c,e}(i,j)\cdot N}{n_c(i)\cdot n_e(j)}}{\sqrt{\left(\sum_{i=1}^{5} n_c(i)\log\frac{n_c(i)}{N}\right)\left(\sum_{j=1}^{U} n_e(j)\log\frac{n_e(j)}{N}\right)}}, \quad (35)$$

where $n_c(i)$ denotes the number of samples that have a class label i according to $\boldsymbol{\lambda}^{(c)}$ and $n_e(j)$ denotes the number of samples that have a class label j according to $\boldsymbol{\lambda}^{(e)}$.

Figure 5 shows a boxplot of the NMI. From Fig. 5, CREM has an advantage over CREM0, GR and VL in terms of NMI at all levels. Moreover, at the fourth level($U = 5$), where mixture models have as many components as the classes of the phoneme data, CREM is comparable to the standard EM applied directly to the data.

Interestingly, we can see that CREM outperforms the standard EM at the second and the third levels, while the standard EM is superior in terms of error rate when $U = 10$ as shown in Fig. 4. This phenomenon may be explained by the fact that in most clustering results obtained from the standard EM when $U = 20$ or 10, some clusters involve far fewer data points than the others. We conjecture that the standard EM is more prone to overfitting when the specified number of components is larger than that of the underlying clusters, which is an unfavorable characteristic for hierarchical cluster analysis.

8. Conclusion

We have proposed a component reduction algorithm for Gaussian mixture models that does not suffer from the lim-

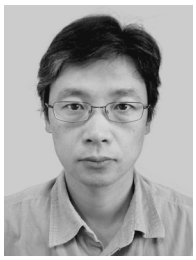
itations of the existing algorithms such as those proposed in [1], [5], [6]. Our algorithm has been derived by applying the EM-algorithm to the component reduction problem and introducing an effective approximation to overcome the difficulty faced in implementing the EM-algorithm.

Our algorithm and the three existing algorithms have been applied to a simple synthetic component reduction task and a phoneme clustering problem. The experimental results strongly support the effectiveness of our algorithm.

References

[1] N. Vasconcelos and A. Lippman, "Learning mixture hierarchies," in Advances in Neural Information Processing Systems 11, ed. M.S. Kearns, S.A. Solla, and D.A. Cohn, pp.606–613, MIT Press, Cambridge, MA, 1999.
[2] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," J. Royal Statistical Society B, vol.39, pp.1–38, 1977.
[3] G.J. McLachlan and T. Krishnan, The EM Algorithm and Extensions, John Wiley & Sons, 1997.
[4] G. McLachlan and D. Peel, Finite Mixture Models, John Wiley & Sons, 2000.
[5] J. Goldberger and S. Roweis, "Hierarchical clustering of a mixture model," in Advances in Neural Information Processing Systems 17, ed. L.K. Saul, Y. Weiss, and L. Bottou, pp.505–512, MIT Press, Cambridge, MA, 2005.
[6] K. Maebashi, N. Suematsu, and A. Hayashi, "Reduction in the number of components in mixture models," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J91-D, no.4, pp.1058–1068, April 2008.
[7] R.M. Neal and G.E. Hinton, "A new view of the EM algorithm that justifies incremental, sparse and other variants," in Learning in Graphical Models, ed. M.I. Jordan, pp.355–368, Kluwer Academic Publishers, 1998.
[8] T. Hastie, R. Tibshirani, and J.H. Friedman, The Elements of Statistical Learning:Data Mining, Inference, and Prediction, Springer-Verlag, 2001.
[9] A. Strehl and J. Ghosh, "Cluster ensembles — A knowledge reuse framework for combining multiple partitions," Machine Learning Research, vol.3, pp.583–417, 2002.

Kumiko Maebashi received the BS and MS degrees in Information Sciences all from Hiroshima City University in 2002 and 2004. She is currently a Ph.D. candidate in the Graduate School of Information Sciences Hiroshima City University. Her research interests include learning probabilistic models.



Nobuo Suematsu received the B.S. degree and the M.S. degree in physics from Kyushu University in 1988 and 1990, respectively. He received his Ph.D. in engineering from Osaka University in 2000. From 1990-1994, he joined Fujitsu Laboratories LTD. In 1994, he joined the faculty member of Hiroshima City University, and currently he is an associate professor of the Graduate School of Information Sciences, Hiroshima City University. His current research interests include machine learning, pattern recognition, etc.

Akira Hayashi received the B.S. degree in mathematics from Kyoto University in 1974. He joined IBM Japan, Ltd. in April, 1974. He received the M.S. degree in computer science from Brown University in 1988, and the Ph.D. degree in computer science from the University of Texas at Austin in 1991. He was a visiting associate professor in Kyushu Institute of Technology, Fukuoka, Japan, until March 1994. Currently, he is a professor in the Faculty of Information Sciences, Hiroshima City University, Hiroshima, Japan. His research interests are in the area of machine learning and pattern recognition.