Component Reduction for Hierarchical Mixture Model Construction

Kumiko Maebashi, Nobuo Suematsu, and Akira Hayashi

Graduate School of Information Sciences,
Hiroshima City University, Hiroshima, 731-3194, Japan
bochin@robotics.im.hiroshima-cu.ac.jp,
{suematsu,akira}@hiroshima-cu.ac.jp

Abstract. The mixture modeling framework is widely used in many applications. In this paper, we propose a *component reduction* technique, that collapses a mixture model into a mixture with fewer components. For fitting a mixture model to data, the EM (Expectation-Maximization) algorithm is usually used. 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.

1 Introduction

Component reduction is the task 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 in 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 granularity levels[1].

Component reduction can be regarded as a task 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 where a mixture model is fitted to another 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 reality, 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.

M. Ishikawa et al. (Eds.): ICONIP 2007, Part II, LNCS 4985, pp. 326–335, 2008.

In Sect. 4, we formulate the application of the EM-algorithm to component reduction and obtain an EM-procedure. Thereafter, in Sect. 5, we derive an approximation of the EM-procedure. In Sect. 6, we apply our method and two 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 assignment probabilities of 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.

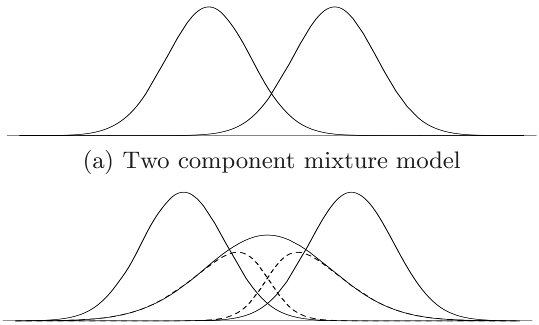
A straight-forward approach to component reduction is to generate samples from the given mixture model, and then to apply the EM-algorithm to these samples. This is, however, computationally inefficient.

By simply replacing "the data points" with "the components of the original mixture" in the above description, 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] can be seen as members of this class.

The algorithm proposed in [1] uses the notion of virtual samples generated from the given mixture. In this algorithm, the assignment probabilities are calculated when the set of virtual samples drawn from a component of the given mixture model is assigned as a whole to the each component of the mixture model being fit. Therefore, the algorithm is regarded as soft clustering of components in the given mixture model.

In [5], another component reduction algorithm is proposed, although the authors considered the case where the component structure of the original model must be preserved. The algorithm assigns each component in the given mixture to one of the components in the fitted mixture, such that the KL-divergence between the mixture models is minimized. In other words, the algorithm involves hard clustering of components in the given mixture into groups corresponding to the components in the fitted mixture.

Since each of the components of the original mixture is spatially extended, unlike in the case of data points, the proper assignment probabilities of the original components to the components being fit should be position dependent. Any member of the aforementioned class of algorithms, such as the above two algorithms, does not take into account this fact adequately. To illustrate this problem, we consider a 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) dependent on the spatial relationships of two components of the fitted mixture. Each of the two parts should then be incorporated into its corresponding component. However, such a splitting process cannot be realized by the algorithms belonging



(a) Two component mixture model

(b) Three component mixture model

Fig. 1. An example of the fitting problem

to the above class. In this paper, we devise a component reduction algorithm which overcomes this limitation.

3 Fitting Mixture Models to Data

We devise a component reduction algorithm based on the application of the EM-algorithm for fitting mixture models to data. We review the application formulated by Dempster[2] here.

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 $j = 1, \ldots, C$, $0 \leq \pi_j \leq 1$ 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 Θ'. The expected value of the log-likelihood is

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

where $h_{ij} = E[y_{ij} \mid \boldsymbol{x}_i, \Theta']$.

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 $\{h_{ij}^{(t)}\}$, under 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 Mixture Models to Another Mixture Model

In this section, we formulate a straight-forward application of the EM-algorithm for fitting mixture models 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 $L > U$,

$$f_{\Theta_U}(\boldsymbol{x}) = \sum_{j=1}^{U}\pi_j^U p(\boldsymbol{x}|\theta_j^U), \qquad \text{and} \qquad f_{\Theta_L}(\boldsymbol{x}) = \sum_{i=1}^{L}\pi_i^L p(\boldsymbol{x}|\theta_i^L).$$

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

$$\Pr(y_j = 1|\boldsymbol{x}, \Theta_U) = \frac{\pi_j^U p(\boldsymbol{x}|\theta_j^U)}{\sum_{j'=1}^{U}\pi_{j'}^U p(\boldsymbol{x}|\theta_{j'}^U)}. \tag{4}$$

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

$$L(\Theta_U|\boldsymbol{x},\boldsymbol{y}) = \sum_{j=1}^{U} y_j \log\{\pi_j^U p(\boldsymbol{x}|\theta_j^U)\}, \tag{5}$$

and the counterpart of $Q(\Theta|\Theta')$ 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

$$\begin{aligned} Q_{\text{hier}}(\Theta_U|\Theta_U') &= E_{\boldsymbol{x}}\{E_{\boldsymbol{y}}\{L(\Theta_U|\boldsymbol{x},\boldsymbol{y}) \mid \boldsymbol{x}, \Theta_U'\} \mid \Theta_L\}, \\ &= \sum_{j=1}^{U}\sum_{i=1}^{L}\pi_i^L \int p(\boldsymbol{x}|\theta_i^L) h_j(\boldsymbol{x}) \log\{\pi_j p(\boldsymbol{x}|\theta_j^U)\} d\boldsymbol{x}, \end{aligned} \tag{6}$$

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

To derive an E-step and an M-step, 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:

$$\Pr(\boldsymbol{x}|z_i = 1, y_j = 1) = \frac{\Pr(y_j = 1|\boldsymbol{x}, z_i = 1)\Pr(\boldsymbol{x}|z_i = 1)}{\Pr(y_j = 1|z_i = 1)}. \tag{7}$$

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 $p(\boldsymbol{x}|i, j)$, (7) can be rewritten as

$$p(\boldsymbol{x}|i, j) = \frac{h_j(\boldsymbol{x})p(\boldsymbol{x}|\theta_i^L)}{h_{ij}}, \tag{8}$$

where $h_{ij} = \Pr(y_j = 1|z_i = 1)$. By substituting (8) into (6), we obtain

$$Q_{\text{hier}}(\Theta_U|\Theta_U') = \sum_{j=1}^{U}\sum_{i=1}^{L}\pi_i^L h_{ij}\int p(\boldsymbol{x}|i, j)\log\{\pi_j p(\boldsymbol{x}|\theta_j^U)\}d\boldsymbol{x}. \tag{9}$$

Although we cannot perform them in reality, we can define the E-step and the M-step simply based on (9) as follows:

E-step: Compute $\{p^{(t)}(\boldsymbol{x}|i, j)\}$ and $\{h_{ij}^{(t)}\}$ under 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 by

$$\begin{aligned}\pi_j^U &= \sum_{i=1}^{L}\pi_i^L h_{ij}^{(t)}, \quad \boldsymbol{\mu}_j^U = \frac{\sum_{i=1}^{L}\pi_i^L h_{ij}^{(t)}\boldsymbol{\mu}_{ij}^{(t)}}{\sum_{i=1}^{L}\pi_i^L h_{ij}^{(t)}}, \\ \Sigma_j^U &= \frac{\sum_{i=1}^{L}\pi_i^L h_{ij}^{(t)}\{\Sigma_{ij}^{(t)} + (\boldsymbol{\mu}_{ij}^{(t)} - \boldsymbol{\mu}_j^U)(\boldsymbol{\mu}_{ij}^{(t)} - \boldsymbol{\mu}_j^U)^{\mathrm{T}}\}}{\sum_{i=1}^{L}\pi_i^L h_{ij}^{(t)}},\end{aligned} \tag{10}$$

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

From (8), $p(\boldsymbol{x}|i,j) \propto h_j(\boldsymbol{x})p(\boldsymbol{x}|\theta_i^L)$ 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

Now, we are 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 be obtained effectively from the solution of

$$\frac{\partial q_{ij}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \mathbf{0}, \tag{11}$$

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

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

$$-\frac{1}{N(\boldsymbol{\mu}|\boldsymbol{\mu},\Sigma)}\left.\frac{\partial^2 N(\boldsymbol{x}|\boldsymbol{\mu},\Sigma)}{\partial \boldsymbol{x}^2}\right|_{\boldsymbol{x}=\boldsymbol{\mu}} = \Sigma^{-1}. \tag{12}$$

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

$$\begin{aligned}\hat{\Sigma}_{ij}^{-1} &= -\frac{1}{p(\hat{\boldsymbol{\mu}}_{\boldsymbol{ij}}|i,j)}\left.\frac{\partial^2 p(\boldsymbol{x}|i,j)}{\partial \boldsymbol{x}^2}\right|_{\boldsymbol{x}=\hat{\boldsymbol{\mu}}_{ij}} = -\frac{1}{q_{ij}(\hat{\boldsymbol{\mu}}_{ij})}\left.\frac{\partial^2 q_{ij}(\boldsymbol{x})}{\partial \boldsymbol{x}^2}\right|_{\boldsymbol{x}=\hat{\boldsymbol{\mu}}_{ij}} \\ &= (\Sigma_i^L)^{-1} + (\Sigma_j^U)^{-1} - \sum_{j'=1}^{U} h_{j'}(\hat{\boldsymbol{\mu}}_{ij})(\Sigma_{j'}^U)^{-1} \\ &+ \sum_{j'=1}^{U} h_{j'}(\hat{\boldsymbol{\mu}}_{ij})(\Sigma_{j'}^U)^{-1}(\hat{\boldsymbol{\mu}}_{ij}-\boldsymbol{\mu}_{j'}^U)(\hat{\boldsymbol{\mu}}_{ij}-\boldsymbol{\mu}_{j'}^U)^{\mathrm{T}}(\Sigma_{j'}^U)^{-1} \\ &- \sum_{j'=1}^{U}\sum_{j''=1}^{U} h_{j'}(\hat{\boldsymbol{\mu}}_{ij})h_{j''}(\hat{\boldsymbol{\mu}}_{ij})(\Sigma_{j'}^U)^{-1}(\hat{\boldsymbol{\mu}}_{ij}-\boldsymbol{\mu}_{j'}^U)(\hat{\boldsymbol{\mu}}_{ij}-\boldsymbol{\mu}_{j''}^U)^{\mathrm{T}}(\Sigma_{j''}^U)^{-1}. \end{aligned} \tag{13}$$

To complete the E-step, we also need to evaluate h_{ij}. From (8), we have

$$h_{ij} = \frac{h_j(\boldsymbol{x})p(\boldsymbol{x}|\theta_i^L)}{p(\boldsymbol{x}|i,j)}, \tag{14}$$

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},

$$\hat{h}_{ij} \propto \frac{h_j(\hat{\boldsymbol{\mu}}_{ij}) p(\hat{\boldsymbol{\mu}}_{ij}|\theta_i^L)}{N(\hat{\boldsymbol{\mu}}_{ij}|\hat{\boldsymbol{\mu}}_{ij}, \hat{\Sigma}_{ij})}. \tag{15}$$

5.3 Approximated EM-Procedure

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

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

1. Set $\{\hat{\boldsymbol{\mu}}_{ij}^{(t)}\}$ by solving (11) using the Newton-Raphson method.
2. Calculate $\{\hat{\Sigma}_{ij}^{(t)}\}$ using (13).
3. Calculate $\{\hat{h}_{ij}^{(t)}\}$ using (15) and normalize them 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 (10) 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 Experimental Results

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

6.1 Synthetic Data

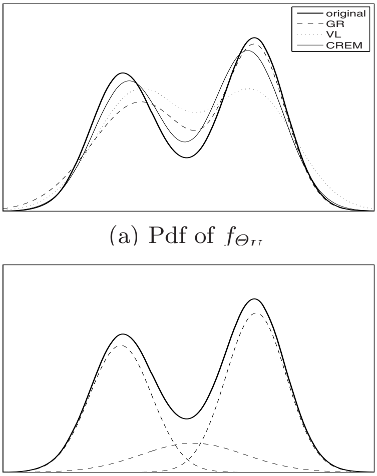
This experiment is intended to verify the effectiveness of our algorithm in component reduction problems similar to the example described in Sect. 2. The experimental procedure is as follows.

1. Draw 500 data points from the 1-dimensional 2-component Gaussian mixture model

$$f_{\Theta_{true}}(x) = \frac{1}{2} \cdot N(x|-2,1) + \frac{1}{2} \cdot N(x|2,1). \tag{16}$$

Table 1. KL-divergence and log-likelihood for data

	$KL(f_{\Theta_L} \| f_{\Theta_U})$	$KL(f_{\Theta_{EM}} \| f_{\Theta_U})$	$KL(f_{\Theta_{true}} \| f_{\Theta_U})$	LL
CREM	0.0120	0.0120	0.0179	−1030.7
GR	0.0347	0.0372	0.0444	−1039.8
VL	0.0780	0.0799	0.0823	−1057.5



(a) Pdf of f_{Θ_U}

(b) Pdf of f_{Θ_L}

Fig. 2. Three and two component mixture model

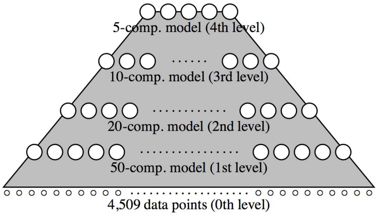

5-comp. model (4th level)
10-comp. model (3rd level)
20-comp. model (2nd level)
50-comp. model (1st level)
4,509 data points (0th level)

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

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, VL, GR and the standard EM, where the initial estimate is determined as

$$f_{\Theta_U}(x) = \pi_1^U \cdot collapsed[\frac{1}{\pi_1^U}\{\pi_1^L N(x|\mu_1,\sigma_1) + 0.5 \cdot \pi_2^L N(x|\mu_2,\sigma_2)\}] + \pi_2^U \cdot collapsed[\frac{1}{\pi_2^U}\{0.5 \cdot \pi_2^L N(x|\mu_2,\sigma_2) + \pi_3^L N(x|\mu_3,\sigma_3)\}], \quad (17)$$

   where $\pi_1^U = \pi_1^L + \pi_2^L/2$, $\pi_2^U = \pi_2^L/2 + \pi_3^L$ and $collapsed[g]$ denotes the Gaussian which has the minimum KL-divergence from g.

The trial was repeated 100 times. We evaluate the results using the KL-divergence, calculated using numerical integration, and the log-likelihood for the generated data. Table 1 shows the averages taken over the 100 trials. The results for CREM show the best value of all the results. We show one of the results in Fig. 2. Fig. 2(a) is a plot of the pdfs obtained by GR, VL, and CREM for the original 3-component mixture shown in Fig. 2(b). We can see that the pdf obtained by CREM is closest to the original pdf.

6.2 TIMIT Phoneme Recognition

We apply the three algorithms to clustering the phoneme dataset described in [6]. The dataset contains 5 phoneme classes of $4,509$ instances described by log-periodograms of length 256. The dimension of the instances 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.

In each trial of the three 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-components mixtures are learned from the data points using the standard EM-algorithm. Since all three algorithms depend on initial guesses $\Theta_U^{(0)}$, we ran the trial 10 times. In the 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 condition of our algorithm was empirically tuned to ensure the convergence of the algorithm. As a result, in this experiment, the EM-procedure was terminated when $\max_{i,j}(h_{ij}^{(t)} - h_{ij}^{(t-1)}) < 10^{-5}$.

We evaluate the clustering results in terms of NMI(normalized mutual information)[7]. 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

$$\lambda_n^{(e)} = \underset{j}{\operatorname{argmax}}(\{\pi_j p(\boldsymbol{x}_n|\theta_j)|j = 1, \ldots, U\}). \tag{18}$$

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

$$\phi^{NMI}(\boldsymbol{\lambda}^{(e)}, \boldsymbol{\lambda}^{(c)}) = \frac{\sum_{h=1}^{5} \sum_{l=1}^{U} n_{h,l} \log \frac{n_{h,l} \cdot N}{n_h \cdot n_l}}{\sqrt{(\sum_{h=1}^{5} n_h \log \frac{n_h}{N}) \cdot (\sum_{l=1}^{U} n_l \log \frac{n_l}{N})}}, \tag{19}$$

where N is the number of samples, $n_{h,l}$ denotes the number of samples that have a class label h according to $\boldsymbol{\lambda}^{(c)}$ as well as a cluster label l according to $\boldsymbol{\lambda}^{(e)}$, $n_h = \sum_l n_{h,l}$, and $n_l = \sum_h n_{h,l}$.

Fig. 4 shows a boxplot of the NMI. 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 the fourth level ($U = 5$), where mixture models have as many components as the classes of the phoneme data, we confirm that CREM has an advantage over GR and VL in terms of NMI. Moreover, CREM is comparable to the standard EM directly applied to the data.

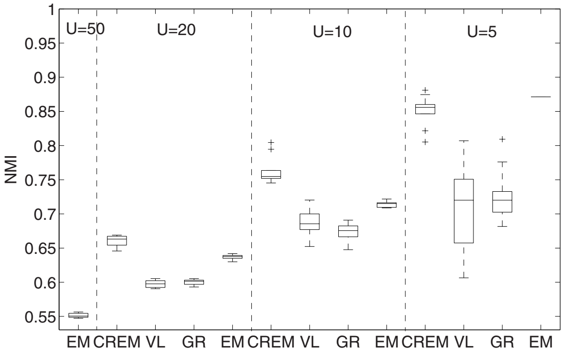


Fig. 4. Boxplot of the NMI for 10 trials

In viewing the results at the second and third levels, we cannot directly compare the results of VL with those of others. This is because the mixtures learned by VL always contained some almost identical components and hence the effective numbers of components were much fewer than the numbers intended. CREM appears to outperform VL and GR at all the levels. In addition, interestingly, we can see that CREM outperforms the standard EM in terms of NMI at the second and third levels. We conjecture that our algorithm is less likely to be trapped by low quality local minima thanks to the coarser descriptions of data. This is a highly preferable behavior for learning algorithms.

7 Conclusion

We have proposed a component reduction algorithm that does not suffer from the limitation of the existing algorithms proposed in [1,5]. Our algorithm was derived by applying the EM-algorithm to the component reduction problem and introducing an effective approximation to overcome the difficulty faced in carrying out the EM-algorithm.

Our algorithm and the two 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. Vasconcelos, N., Lippman, A.: Learning mixture hierarchies. In: Kearns, M.J., Solla, S.A., Cohn, D. (eds.) Advances in Neural Information Processing Systems, vol. 11, pp. 606–612 (1999)
2. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society B 39, 1–38 (1977)
3. McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions. John Wiley and Sons Inc., Chichester (1997)
4. McLachlan, G., Peel, D.: Finite Mixture Models. John Wiley and Sons Inc., Chichester (2000)
5. Goldberger, J., Roweis, S.: Hierarchical clustering of a mixture model. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 17, pp. 505–512. MIT Press, Cambridge (2005)
6. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Heidelberg (2001)
7. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Machine Learning Research 3, 583–617 (2002)