Unsupervised Learning of Finite Mixtures Using Entropy Regularization and Its Application to Image Segmentation

Zhiwu Lu, Yuxin Peng*, and Jianguo Xiao
Institute of Computer Science and Technology, Peking University, Beijing 100871, China
{luzhiwu,pengyuxin,xjg}@icst.pku.edu.cn

Abstract

When fitting finite mixtures to multivariate data, it is crucial to select the appropriate number of components. Under regularization theory, we aim to resolve this "unsupervised" learning problem via regularizing the likelihood by the full entropy of posterior probabilities for finite mixture fitting. Two deterministic annealing implementations are further proposed for this entropy regularized likelihood (ERL) learning. Through some asymptotic analysis of the deterministic annealing ERL (DAERL) learning, we find that the global minimization of the ERL function in an annealing way can lead to automatic model selection on finite mixtures and also make our DAERL algorithms less sensitive to initialization than the standard EM algorithm. The simulation experiments then demonstrate that our algorithms can provide some promising results just as our theoretic analysis. Moreover, our algorithms are evaluated in the application of unsupervised image segmentation and shown to outperform other state-of-the-art methods.

1. Introduction

As a powerful statistical modeling tool for multivariate data, the finite mixture model [7] has been widely used in many applications such as pattern recognition, computer vision, and image analysis. The standard method used to fit finite mixtures to the data is the EM algorithm [9], which converges to a maximum likelihood estimation of the mixture parameters. However, the number k^* of components in the mixture is usually assumed to be fixed and must be provided first. In many instances, this key information is not available, and then we have to select k^* to best fit the data before or during parameter estimation.

The traditional method to solve this model selection problem is to choose the optimal number k^* of components via some statistical criteria such as minimum description length (MDL) [10], minimum message length (MML) [15], and penalty-less information criterion [1]. However, the entire parameter estimation for finite mixtures has to be repeated at different values of k, and the process of evaluating these criteria incurs a large computational cost. Moreover, since at each given k the mixture parameters are often estimated by the EM algorithm, the final model selection will be unavoidably misled by the local convergence problem of it. Other than this deterministic method, there are some even more computationally demanding methods to solve the model selection problem. That is, we can take into account the stochastic simulation [11] and resampling [6] methods to infer the optimal mixture model.

Recently, some more efficient methods have also been developed for automatic model selection on finite mixtures, which follow the idea that an appropriate number of components can be automatically selected during parameter learning by forcing the mixing probabilities of the extra components to tend to zeros. Such component annihilation has been combined with the MML criterion in [4] to speed up the traditional criterion based methods, and then a MML-based component-wise EM algorithm (MML-CEM) is derived which can determine the component number in the optimization procedure. However, the strength of component annihilation can not be controlled and then may be too strong to detect the component number in some cases.

Under regularization theory [12], the above problems can be resolved via regularizing the likelihood by the full entropy of posterior probabilities which can control the model complexity of the mixture. Though the Dirichlet and entropic priors have also been used as regularization terms in [4, 2], these priors defined only by the mixing probabilities can not result in a close-form solution of the M-step. Of course, we can propose some gradient implementations for this entropy regularized likelihood (ERL) learning. However, just as the standard EM algorithm, the gradient-type algorithms have the local convergence problem (e.g., sensitive to initialization). Hence, we follow the deterministic annealing idea of gradually shifting minimum ERL function to minimum negative-likelihood (i.e., maximum likelihood) for global search of the solution, and then propose two de-

*Corresponding author.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

terministic annealing ERL (DAERL) learning algorithms. Other than the deterministic annealing EM (DAEM) algorithm [13] which just considers the local convergence problem, our DAERL algorithms aim to resolve this problem and the model selection problem simultaneously.

The main contribution of this paper is proposing a mechanism of entropy regularization for finite mixture fitting and two algorithms to implement it which are: 1) able to automatically select the number of components (i.e., automatic model selection), and 2) less sensitive to initialization than EM. We also give some asymptotic analysis of our algorithms and show in theory that the global minimization of the ERL function in an annealing way just leads to automatic model selection on finite mixtures. Moreover, the results of unsupervised image segmentation on the Berkeley segmentation database [5] demonstrate that our algorithms outperform other state-of-the-art methods such as the MDL based EM (MDL-EM) algorithm [3] and the MML-CEM algorithm [4] even in this challenging application.

2. Entropy Regularized Likelihood Learning

A random variable $x \in R^n$ is said to follow a k-component finite mixture distribution, if its probability density function (pdf) can be written as:

$$p(x|\Theta_k) = \sum_{l=1}^{k} \alpha_l p(x|\theta_l), \tag{1}$$

where $\{\alpha_l\}_{l=1}^k$ is the set of mixing probabilities satisfying $\alpha_l \geq 0$ and $\sum_{l=1}^{k} \alpha_l = 1$, each θ_l is the set of parameters defining the l-th mixture component, and $\Theta_k = \{\alpha_l, \theta_l\}_{l=1}^k$ is the complete set of mixture parameters.

Given a set of N independent and identically distributed samples $S = \{x_t\}_{t=1}^N$, the negative log-likelihood function corresponding to a k-component finite mixture model $p(x|\Theta_k)$ is

$$L(\Theta_k) = -\frac{1}{N} \sum_{t=1}^{N} \ln(\sum_{l=1}^{k} (p(x_t|\theta_l)\alpha_l)). \tag{2}$$

The well-known EM algorithm for fitting finite mixtures to the data is just an implementation of minimizing $L(\Theta_k)$.

Since we have the posterior probability that x_t arises from the l-th component in the finite mixture model

$$P(l|x_t) = p(x_t|\theta_l)\alpha_l / \sum_{j=1}^{k} p(x_t|\theta_j)\alpha_j, \ l = 1, ..., k, \tag{3}$$

the discrete Shannon entropy of these posterior probabilities for the sample x_t can then be calculated as:

$$E(x_t|\Theta_k) = -\sum_{l=1}^{k} P(l|x_t) \ln P(l|x_t). \tag{4}$$

Note that this entropy has a good property: $E(x_t|\Theta_k)$ is globally minimized at $P(l_0|x_t) = 1, P(l|x_t) = 0$ $(l \neq l_0)$, i.e., the sample x_t is determinedly classified into the l_0-th component in the finite mixture model.

We now consider the average entropy of the finite mixture model over the sample set S:

$$E(\Theta_k) = -\frac{1}{N} \sum_{t=1}^{N} \sum_{l=1}^{k} P(l|x_t) \ln P(l|x_t), \tag{5}$$

and use it to regularize the log-likelihood function by

$$H(\Theta_k) = L(\Theta_k) + \gamma E(\Theta_k), \tag{6}$$

where $\gamma \in [0, \ \gamma_{max}]$ is the regularization factor. That is, $E(\Theta_k)$ is a regularization term to reduce the model complexity such that the finite mixture model can be made as simple as possible by minimizing $H(\Theta_k)$.

3. Two Deterministic Annealing Implementations of ERL Learning

Since the gradient-type algorithms may converge to local minima just as the standard EM algorithm, we further follow the deterministic annealing idea of gradually shifting minimum ERL function $H(\Theta_k)$ to minimum negative-likelihood $L(\Theta_k)$ (i.e., maximum likelihood) with γ gradually reduced to zero, and propose two deterministic annealing implementations for searching the global minimum of the ERL function. Unlike the traditional simulated annealing approach, the ERL function is deterministically minimized at each temperature (controlled by γ), and then this search of the global minimum is much faster.

Note that the two deterministic annealing implementations differ only in the way to estimate the posterior probabilities $P(l|x_t)$ $(l = 1, ..., k)$. On one hand, when the posterior probabilities are considered as functions of mixture parameters Θ_k according to (3), they can then be estimated by $\min_{\Theta_k} H(\Theta_k)$, i.e., first estimate Θ_k and then $P(l|x_t)$. We denote this deterministic annealing implementation of ERL Learning as DAERL1 algorithm in the following. On the other hand, when the posterior probabilities are considered as part of mixture parameters, i.e., $\Theta_k^p = \{P(l|x_t), \ t = 1, ..., N, \ l = 1, ...k\}$, they can then be estimated by $\min_{\Theta_k^p, \Theta_k} H(\Theta_k^p, \Theta_k)$. We denote this deterministic annealing implementation of ERL Learning as DAERL2 algorithm in the following.

In order to derive an explicit expression to update the mixture parameters by minimum ERL function, we focus on a special case of finite mixtures, i.e., the Gaussian mixture model with $p(x|\theta_l)$ given by a Gaussian pdf

$$p(x|\theta_l) = (2\pi)^{-\frac{n}{2}} |\Sigma_l|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-m_l)^T \Sigma_l^{-1}(x-m_l)}, \tag{7}$$

where m_l is the mean vector and Σ_l is the covariance matrix which is assumed to be positive definite. Note that the deterministic annealing implementations of ERL learning presented in the following can be applied to other types of finite mixture models in a similar way.

3.1. DAERL1 Algorithm

We first derive a deterministic annealing implementation of minimum ERL function (i.e., DAERL1) when the posterior probabilities are considered as functions of mixture parameters Θ_k. At each temperature with γ fixed, since $\sum_{l=1}^{k} \alpha_l = 1$, we can compose the following Lagrange function for the task of $\min_{\Theta_k} H(\Theta_k)$:

$$Q(\Theta_k, \lambda) = H(\Theta_k) + \lambda(\sum_{l=1}^{k} \alpha_l - 1). \quad (8)$$

Using the general methods for matrix derivatives, we are then led to the following series of equations:

$$\frac{\partial Q}{\partial \alpha_l} = -\frac{1}{N}\sum_{t=1}^{N} \frac{1}{\alpha_l} U(l|x_t) + \lambda, \quad (9)$$

$$\frac{\partial Q}{\partial m_l} = -\frac{1}{N}\sum_{t=1}^{N} U(l|x_t)\Sigma_l^{-1}(x_t - m_l), \quad (10)$$

$$\frac{\partial Q}{\partial \Sigma_l} = -\frac{1}{2N}\sum_{t=1}^{N} U(l|x_t)\Sigma_l^{-1}(M_{tl} - \Sigma_l)\Sigma_l^{-1}, \quad (11)$$

$$\frac{\partial Q}{\partial \lambda} = \sum_{l=1}^{k} \alpha_l - 1, \quad (12)$$

where $M_{tl} = (x_t - m_l)(x_t - m_l)^T$ and

$$U(l|x_t) = P(l|x_t)(1 + \gamma(\ln P(l|x_t) + E(x_t|\Theta_k))). \quad (13)$$

By setting these derivatives of $Q(\Theta_k, \lambda)$ with respect to α_l, m_l, Σ_l, and λ to be zeros, we then have:

$$\hat{\alpha}_l = \sum_{t=1}^{N} U(l|x_t) / \sum_{j=1}^{k}\sum_{t=1}^{N} U(j|x_t), \quad (14)$$

$$\hat{m}_l = \sum_{t=1}^{N} U(l|x_t)x_t / \sum_{t=1}^{N} U(l|x_t), \quad (15)$$

$$\hat{\Sigma}_l = \sum_{t=1}^{N} U(l|x_t)M_{tl} / \sum_{t=1}^{N} U(l|x_t). \quad (16)$$

These explicit expressions actually give us a deterministic annealing implementation of minimum ERL function: at each temperature, we first update $P(l|x_t)$ according to (3), and then update Θ_k according to (14)-(16). Though similar to the standard EM algorithm, this DAERL1 algorithm makes some important modification of the M-step. That is, during updating Θ_k, the mechanism of entropy regularization enforces a kind of competitive learning among all the components, which then leads to automatic model selection on finite mixtures.

3.2. DAERL2 Algorithm

We further derive another deterministic annealing implementation of minimum ERL function (i.e., DAERL2) when the posterior probabilities are considered as part of mixture parameters $\Theta_k^p = \{P(l|x_t),\ t = 1, ..., N,\ l = 1, ...k\}$. At each temperature with γ fixed, the task of $\min_{\Theta_k^p, \Theta_k} H(\Theta_k^p, \Theta_k)$ can be implemented by a two-step minimization procedure similar to the EM algorithm: (1) E-step: fix Θ_k, $\hat{\Theta}_k^p = \arg\min_{\Theta_k^p} H(\Theta_k^p, \Theta_k)$; (2) M-step: fix $\hat{\Theta}_k^p$, $\hat{\Theta}_k = \arg\min_{\Theta_k} H(\hat{\Theta}_k^p, \Theta_k)$.

For the E-step of ERL learning with Θ_k fixed, we can compose the following Lagrange function using N Lagrange multipliers $\lambda = (\lambda_1, ..., \lambda_N)$:

$$Q(\Theta_k^p, \lambda) = H(\Theta_k^p) + \frac{1}{N}\sum_{t=1}^{N} \lambda_t(\sum_{l=1}^{k} P(l|x_t) - 1). \quad (17)$$

By setting the derivatives of $Q(\Theta_k^p, \lambda)$ with respect to $P(l|x_t)$ and λ_t to be zeros, we have

$$\ln(p(x_t|\theta_l)\alpha_l) + (\gamma - 1)(1 + \ln P(l|x_t)) = \lambda_t, \quad (18)$$

$$\sum_{l=1}^{k} P(l|x_t) = 1. \quad (19)$$

From the above equations, we can then obtain the following fixed-point solution to estimate $P(l|x_t)$:

$$\hat{P}(l|x_t) = (p(x_t|\theta_l)\alpha_l)^{\frac{1}{1-\gamma}} / \sum_{j=1}^{k} (p(x_t|\theta_j)\alpha_j)^{\frac{1}{1-\gamma}}, \quad (20)$$

which is just the Gibbs distribution. If $0 < \gamma < 1$, the Gibbs distribution is more peaked than the estimated posterior according to (3). That is, the belonging component of x_t is forced to become clearer, which just coincides with our motivation to introduce entropy regularization.

For the M-step of ERL learning with $\hat{\Theta}_k^p$ fixed, $\hat{\Theta}_k = \arg\min_{\Theta_k} H(\hat{\Theta}_k^p, \Theta_k)$ is equivalent to $\hat{\Theta}_k = \arg\min_{\Theta_k} L(\Theta_k)$. Since the EM algorithm is just an implementation of minimizing $L(\Theta_k)$, the update rules for Θ_k can then be set the

same form as the M-step of the EM algorithm:

$$\hat{\alpha}_l = \frac{1}{N}\sum_{t=1}^{N}\hat{P}(l|x_t), \tag{21}$$

$$\hat{m}_l = \sum_{t=1}^{N}\hat{P}(l|x_t)x_t / \sum_{t=1}^{N}\hat{P}(l|x_t), \tag{22}$$

$$\hat{\Sigma}_l = \sum_{t=1}^{N}\hat{P}(l|x_t)M_{tl} / \sum_{t=1}^{N}\hat{P}(l|x_t). \tag{23}$$

Through the above two-step minimization procedure, we have actually present another deterministic annealing implementation of minimum ERL function: at each temperature, we first update $P(l|x_t)$ according to (20), and then update Θ_k according to (21)-(23). Though similar to the standard EM algorithm, this DAERL algorithm makes some important modification of the E-step. That is, during updating $P(l|x_t)$, the mechanism of entropy regularization is implemented on these posterior probabilities using the Gibbs distribution, which are more peaked than the estimated posterior according to (3). Hence, the belonging component of each sample x_t is forced to become clearer, which then leads to automatic model selection on finite mixtures.

Note that the above DAERL2 algorithm takes a similar form as the DAEM algorithm [13]. When the regularization factor γ is replaced by the annealing parameter β in the DAEM algorithm using $\gamma = (\beta-1)/\beta$ $(0 < \beta < 1)$, the DAERL2 algorithm just becomes the DAEM algorithm. Hence, for $\gamma < 0$, our method is actually a kind of maximum-entropy (similar to DAEM) to smooth the likelihood $L(\Theta_k)$ for overcoming the local convergence problem of the standard EM algorithm. Moreover, for $\gamma > 0$, our method is actually a kind of minimum-entropy to make automatic model selection for finite mixture fitting. These two seemingly opposite methods are then unified in the framework of entropy regularization.

3.3. Asymptotic Analysis of DAERL Learning

Finally, we try to give an asymptotic analysis of the above DAERL learning, and then prove the promising property of automatic model selection on finite mixtures when there is a certain degree of overlap in the mixture.

Due to the randomness in the sample set, we have to consider the ERL learning asymptotically, i.e., we let $N \to \infty$. The object function $H(\Theta_k)$ of the ERL learning estimated on the sample set in (6) is rewritten as $H_N(\Theta_k)$. Likewise, the estimated functions $L(\Theta_k)$ and $E(\Theta_k)$ in (2) and (5) are also rewritten as $L_N(\Theta_k)$ and $E_N(\Theta_k)$, respectively. According to the probability theory, we then have

$$\begin{aligned} H(\Theta_k) &= \lim_{N\to\infty} H_N(\Theta_k) = \lim_{N\to\infty}(L_N(\Theta_k) + \gamma E_N(\Theta_k)) \\ &= L(\Theta_k) + \gamma E(\Theta_k), \end{aligned} \tag{24}$$

and now $L(\Theta_k)$ and $E(\Theta_k)$ are updated as

$$L(\Theta_k) = \lim_{N\to\infty} L_N(\Theta_k) = -\int p(x|\Theta^*_{k^*}) \ln p(x|\Theta_k) dx,$$

$$E(\Theta_k) = \lim_{N\to\infty} E_N(\Theta_k) = \int E(x|\Theta_k) p(x|\Theta^*_{k^*}) dx,$$

where $\Theta^*_{k^*} = \{\alpha^*_l, \theta^*_l\}_{l=1}^{k^*}$ denotes the set of the true parameters in finite mixtures which the sample data come from. Specifically, k^* is the number of the actual components and $\{\alpha^*_l, \theta^*_l\}$ is the set of true parameters of the l-th component for the actual mixture pdf.

In the following, we give asymptotic analysis of the ERL learning in the case that the finite mixture model $p(x|\Theta^*_{k^*})$ has a certain degree of component overlap. According to information theory, $E(x|\Theta^*_{k^*})$ is high when the belonging component of x is obscure, i.e., the component overlap is large; otherwise, $E(x|\Theta^*_{k^*})$ is low when the belonging component of x is clear, i.e., the component overlap is small. Hence, the average entropy $E(\Theta^*_{k^*})$ can be used to measure the overlap of the finite mixture model. In this paper, we assume that the overlap of the true finite mixture model $p(x|\Theta^*_{k^*})$ should not be too high, i.e., the average entropy $E(\Theta^*_{k^*})$ should be constrained as $|E(\Theta^*_{k^*})| < M \ll \ln k^*$. Note that the true finite mixture model $p(x|\Theta^*_{k^*})$ will tend to the maximum overlap $E(\Theta^*_{k^*}) = \ln k^*$, when $P(l|x) = 1/k^*, l = 1, ..., k^*$ at each data x.

Moreover, the finite mixture model we consider is assumed to be identifiable. That is, in the cases that all the components in the mixture are different, $p(x|\Theta_k) = p(x|\Theta'_{k'})$ if and only if $\Theta_k \supseteq \Theta'_{k'}$ with $k \geq k'$ and the mixing probabilities of the other $k - k'$ extra components in Θ_k being zeros. We now investigate the asymptotic convergence properties of the ERL learning for the finite mixture model and have the following theorem (see the Appendix for the proof).

Theorem 1 *Suppose that the finite mixture model $p(x|\Theta_k)$ is identifiable, and the overlap of the true finite mixtures $p(x|\Theta^*_{k^*})$ is not too high, i.e., $|E(\Theta^*_{k^*})| < M \ll \ln k^*$. If $\Theta^h_{k^h}(\gamma) = \arg\min_{\Theta_k} H(\Theta_k)$, we then have $\Theta^h_{k^h}(\gamma) \supseteq \Theta^*_{k^*}$ with $k^h \geq k^*$ and the mixing probabilities of the other $k^h - k^*$ components in $\Theta^h_{k^h}(\gamma)$ being zeros, when the regularization factor $\gamma \to 0$.*

According to Theorem 1, we can find that the global minimization of the ERL function $H(\Theta_k)$ in an annealing way (i.e., $\gamma \to 0$) leads to automatic model selection on finite mixtures if we let $k > k^*$ and annihilate the components with negligible mixing probabilities. That is, if the model scale is actually defined by the number of positive mixing probabilities in a finite mixture model, it will be equal to k^* via globally minimizing the ERL function $H(\Theta_k)$. Thus, the true model scale can be correctly detected through the global minimization of the ERL function.

Though entropy regularization is originally introduced into the maximum likelihood estimation to resolve the model selection problem, we can also find that minimum ERL function $H(\Theta_k)$ may escape some types of local minima and then avoid the initialization dependence. That is, when local minima of the negative likelihood $L(\Theta_k)$ arise during minimizing $H(\Theta_k)$, the average entropy $E(\Theta_k)$ may still keep large and these local minima may then be avoided. For example, the EM algorithm may not escape one type of local minima when two or more components in the mixture have similar parameters (i.e., the overlap $E(\Theta_k)$ is high) and then share the same data. However, the ERL learning can promote the competition among these components by minimum $H(\Theta_k)$, and then only one of them will survive with the other annihilated.

4. Unsupervised Image Segmentation

Though our DAERL algorithms can be used in many applications, we will focus on unsupervised image segmentation which aims to automatically determine the number of regions (objects) in an image during segmentation. Note that unsupervised image segmentation plays an important role in region-based image retrieval, since the image databases are often huge in this application and the prior setting of region number for each image is no longer feasible.

To resolve this model selection problem in a probabilistic way for unsupervised image segmentation, we will pay our attention to mixture model-based image segmentation (e.g. the spatially variant finite mixture model [8]). Thus, we can take into account our DAERL algorithms which are proposed in the above section to make automatic model selection on finite mixtures.

In the segmentation, we consider an 8-dimensional vector of color, texture, and position features for each pixel of an image, and these features are obtained just as [3]. The three color features are the coordinates in the L*a*b* color space, and we smooth these features to avoid oversegmentation arising from local color variations due to texture. The three texture features are contrast, anisotropy, and polarity, which are extracted at an automatically selected scale. The position features are simply the (x, y) position of the pixel. Once pixels in an image with these combined features are grouped into regions by our DAERL algorithms, we further merge those regions smaller than 1 percent of the image with the adjacent regions if they are similar in the color/texture feature space.

Since our algorithms are based on finite mixtures, we can extend them straightforwardly to use the spatial information just as the mixture model-based image segmentation approach [8]. Additionally, since we can just implement region-based image retrieval if the segmented regions are assigned with the color/texture features, our algorithms can further be evaluated in this application.

5. Experimental Results

Though the two DAERL algorithms can be used for other types of mixture models, we only consider Gaussian mixtures to present their performances. To make comparison with other state-of-the-art methods such as MDL-EM [3] and MML-CEM [4], we first give the initialization details for these algorithms and then make simulation experiments on two sample sets generated from Gaussian mixtures. Moreover, we apply the two DAERL algorithms to unsupervised image segmentation on the Berkeley segmentation database [5], and the four algorithms are evaluated by the probabilistic Rand (PR) index [14].

5.1. Initialization for ERL Learning

The ERL learning is always implemented with $k = k_{max}$ and $\gamma = \gamma_{max}$. That is, the number of components k is initialized a large value k_{max} to make sure $k \geq k^*$ (k^* is the true number of components) and then we annihilate those components with $\hat{\alpha}_l$ reduced below a threshold T (e.g., $T = 0.01$) after certain iterations, while the regularization factor γ is gradually reduced from γ_{max} to zero by $\gamma = \gamma_{max}/(1 + ct)$ where t denotes the number of iterations and we set $c = 0.1$ simply. Moreover, the mean vectors and covariance matrices of the mixture components are initialized by some clustering methods (e.g., k-means). In the experiments, the learning is stopped if $|(H(\hat{\Theta}_k) - H(\Theta_k))/H(\Theta_k)| < 10^{-4}$. Note that the above initialization method and convergence criterion are also used similarly by MDL-EM and MML-CEM.

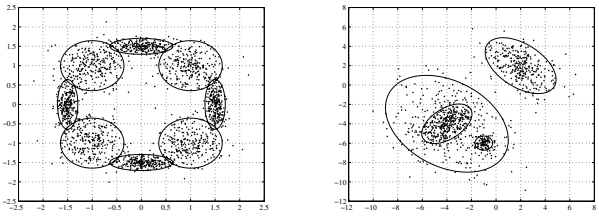
Since the strength of component annihilation during the ERL learning is just controlled by $\gamma = \gamma_{max}$ according to (27) in the Appendix, it is important to select this parameter appropriately for correct model selection. When the mixture overlap is lower, we can select γ_{max} in a large range (e.g., $[0.5,\ 0.8]$) and the component number can be correctly determined after only one run of the ERL learning. However, when the mixture overlap becomes high, the ERL learning may drop some components if γ_{max} is large.

We can solve this problem by introducing the MDL criterion into the ERL learning, which always starts at small γ_{max} (e.g., $\gamma_{max} \in [0.05,\ 0.50)$). If the selected component number $\hat{k} > k_{min}$ after convergence, we can evaluate this candidate model using MDL and then restart the ERL learning with $k = \hat{k} - 1$ by annihilating the least probable component with smallest $\hat{\alpha}_l$ (i.e., in a similar way as [4]). We try to find the optimal model with smallest MDL via repeating the ERL learning until $\hat{k} = k_{min}$. Such ERL learning is different from the traditional approach (e.g., MDL-EM in [3]) in that we only need make a few MDL evaluations since there are some components annihilated during each run of the ERL learning.

5.2. Simulation Results

To present the performance of the ERL learning, we first carry out simulation experiments on the sample data set of $N = 2000$ samples generated from a bivariate 8-component Gaussian mixture as shown in Figure 1(a). The parameters of this Gaussian mixture are:

$$\alpha_l = 1/8,\ l = 1, ..., 8,\ m_1 = [1.5, 0]^T,\ m_2 = [1, 1]^T,$$
$$m_3 = [0, 1.5]^T,\ m_4 = [-1, 1]^T,\ m_5 = [-1.5, 0]^T,$$
$$m_6 = [-1, -1]^T,\ m_7 = [0, -1.5]^T,\ m_8 = [1, -1]^T,$$
$$\Sigma_1 = \Sigma_5 = diag[0.01, 0.1],\ \Sigma_3 = \Sigma_7 = diag[0.1, 0.01],$$
$$\Sigma_2 = \Sigma_4 = \Sigma_6 = \Sigma_8 = diag[0.1, 0.1].$$



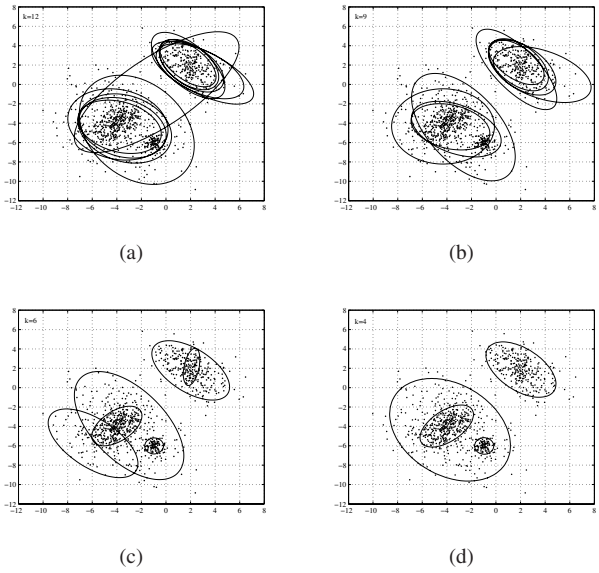
(a) (b)

Figure 1. Two sample data sets generated from Gaussian mixtures: (a) $N = 2000,\ k^* = 8$; (b) $N = 1000,\ k^* = 4$.

In the second example, we use $N = 1000$ samples from a bivariate 4-component Gaussian mixture as shown in Figure 1(b). The detailed parameters of this mixture can be found in [4]. Note that there are two components completely overlapped with the other one in this mixture, while the first mixture as shown in Figure 1(a) has only a low degree of overlap among mixture components.

In the experiments, we just set $k_{min} = 2$ and $k_{max} = 20$ for all the four algorithms. As for γ_{max}, we simply set $\gamma_{max} = 0.2$. Moreover, we run all the algorithms (Matlab code) on a Pentium D 2.8GHz computer with 1.0GB memory. To make an objective evaluation, the success rate of identifying the k^* true components over 250 trials and the average running time for fitting the two mixtures shown in Figure 1 are listed in Table 1 & 2, respectively. Concerning the ability of finding the global optimal solution (i.e., identifying the true number of components and simultaneously avoiding local minima), we can find that the two DAERL algorithms perform generally better than the MDL-EM and MML-CEM algorithms, especially in complicated situations (e.g., the Gaussian mixture of Figure 1(b)). Here, we only evaluate different algorithms by some external criteria since the class labels of all samples are known.

In order to explain in detail the mechanism of automatical detection of the component numbers, we further present the evolution of the ERL learning (only DAERL1 is considered) during fitting the second mixture from one of the successful trials. As shown in Figure 2, we can observe that when two or more components fall into the same data (at a high probability), the ERL learning can promote the competition among them and then make only one of them survive with the other annihilated.



(a) (b) (c) (d)

Figure 2. Fitting the Gaussian mixture of Figure 1(b) by the DAERL1 algorithm: (a), (b), and (c) three intermediate estimates (for $k = 12,\ 9$, and 6); (d) the final estimate (with $k = 4$). Each solid ellipse is the level-curve of a component estimate, and only those components with $\hat{\alpha}_l > T$ are shown.

Algorithm	Success rate	Average time (sec.)
MDL-EM	94.0%	70.6
MML-CEM	**100.0%**	**6.9**
DAERL1	**100.0%**	52.8
DAERL2	97.6%	53.7

Table 1. Experimental results for the mixture of Figure 1(a).

Algorithm	Success rate	Average time (sec.)
MDL-EM	82.0%	33.3
MML-CEM	73.2%	**7.0**
DAERL1	**92.8%**	17.4
DAERL2	92.4%	16.2

Table 2. Experimental results for the mixture of Figure 1(b).

When the computational cost is concerned, we can conclude from Table 1 & 2 that MML-CEM is the lowest, DAERL (DAERL1 or DAERL2) is some more, and MDL-EM is the highest. Though DAERL incurs more computational cost than MML-CEM, it performs much better for identifying the true component number. Hence, DAERL is preferred if we make an overall comparison.

5.3. Segmentation Results

We further apply our DAERL algorithms to unsupervised image segmentation on the Berkeley segmentation database [5]. This benchmark has 300 images along with human (ground truth) segmentations by different individuals. The evaluation of a segmentation algorithm can be achieved by the PR index [14] which takes values between 0 and 1, and a higher PR score indicates that a higher percentage of pixel pairs in the machine segmentation have the same relationship as in each ground truth segmentation.

Algorithm	Average PR index	Average time (sec.)
MDL-EM	0.778	270.0
MML-CEM	0.749	**33.6**
DAERL1	0.797	50.2
DAERL2	**0.808**	88.1

Table 3. The average PR index and running time for all the 300 images in the Berkeley segmentation database.

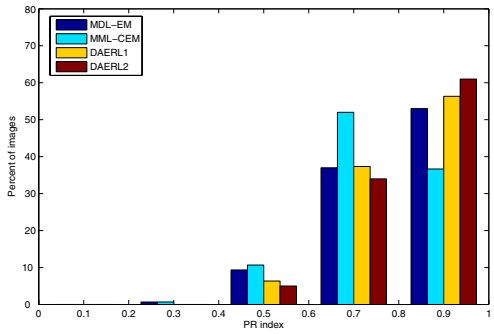


Figure 3. The histogram of the average PR index for all the 300 images in the Berkeley segmentation database.

In the experiments, we simply set $\gamma_{max} = 0.4$ for our DAERL algorithms. Moreover, we just select the region number in $[2,\ 10]$ (i.e, $k_{min} = 2$ and $k_{max} = 10$) for the four algorithms. We then compare these algorithms on the whole database, and their average PR indices for all the 300 images are listed in Table 3. Just as our theoretic analysis in Section 3.3, our algorithms make a global search of solution and then outperform MDL-EM and MML-CEM. Moreover, the histogram shown in Figure 3 gives more details about their PR indices, and we find that our algorithms can create realistic segmentations at a higher probability, if those segmentations with $PR > 0.8$ are considered "realistic".

Moreover, some segmentation samples are also shown in Figure 4, and we can find that our DAERL algorithms successfully detect the object of interest even from the confusing background. However, the MDL-EM algorithm may converge to local minima and then the background may be split into two regions (see the images #134052 and #253036), while the MML-CEM algorithm can not control the strength of component annihilation and the object of interest may be merged with other regions (see the images #134052, #169012, and #249061).

Finally, the average running time taken by the four algorithms on the Berkeley segmentation database is also listed in Table 3. As expected, our algorithms run much faster than MDL-EM which incurs many evaluations of the MDL criterion. As compared with MML-CEM, our algorithms can still be considered computationally comparable in view of the costly annealing procedure. Additionally, if the segmentation results are also taken into account, we can conclude that our algorithms perform generally better.

6. Conclusion

We have investigated the model selection and parameter estimation for finite mixtures through implementing a kind of DAERL learning. Some asymptotic analysis of the DAERL learning then shows that the global minimization of the ERL function in an annealing way can lead to automatic model selection on finite mixtures and also make our algorithms less sensitive to initialization than the standard EM algorithm. The simulation and segmentation experiments then demonstrate that our algorithms can provide some promising results just as our theoretic analysis.

Acknowledgements

The work described in this paper was fully supported by the National Natural Science Foundation of China under Grant No. 60503062, the Beijing Natural Science Foundation of China under Grant No. 4082015, and the Program for New Century Excellent Talents in University under Grant No. NCET-06-0009.

Appendix

This appendix presents the proof of Theorem 1. We begin with $\Theta^h_{k^h}(\gamma) = \arg\min_{\Theta_k} H(\Theta_k)$ at each temperature (controlled by γ), and have $H(\Theta^h_{k^h}(\gamma)) \leq H(\Theta^*_{k^*})$. It then follows from (24) that

$$L(\Theta^h_{k^h}(\gamma)) - L(\Theta^*_{k^*}) \leq \gamma[E(\Theta^*_{k^*}) - E(\Theta^h_{k^h}(\gamma))]. \quad (25)$$

Under information theory, $E(\Theta^*_{k^*}) \geq 0$ and $E(\Theta^h_{k^h}(\gamma)) \geq 0$. Hence, with $|E(\Theta^*_{k^*})| < M$, it follows that $0 \leq E(\Theta^*_{k^*}) < M$. According to (25), we have

$$L(\Theta^h_{k^h}(\gamma)) - L(\Theta^*_{k^*}) \leq \gamma E(\Theta^*_{k^*}) < \gamma M. \quad (26)$$

The difference between $\Theta^h_{k^h}$ and $\Theta^*_{k^*}$ can be measured by $D_{KL}(p(x|\Theta^*_{k^*}), p(x|\Theta^h_{k^h}(\gamma))) = \int p(x|\Theta^*_{k^*}) \ln \frac{p(x|\Theta^*_{k^*})}{p(x|\Theta^h_{k^h}(\gamma))} dx$, where $D_{KL}(\cdot,\cdot)$ is the Kullback-Leibler distance between two probability

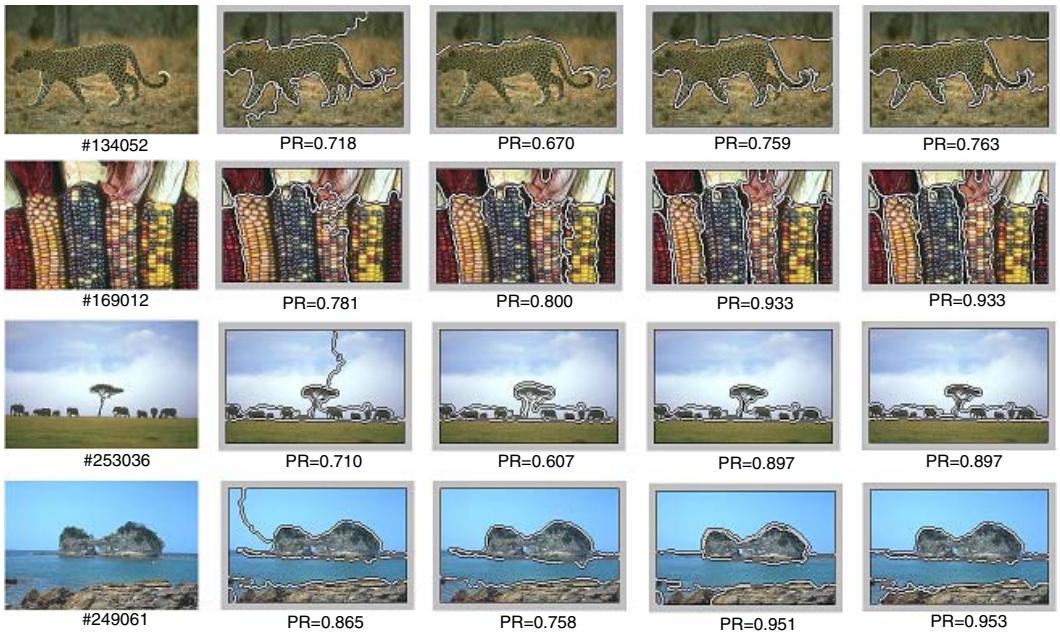


Figure 4. Segmentation results for some sample images by the four segmentation algorithms. The first column is the original images. The second to fifth columns are the results obtained by MDL-EM, MML-CEM, DAERL1, and DAERL2, respectively.

densities and it always keeps $D_{KL}(\cdot,\cdot) \geq 0$. Since $D_{KL}(p(x|\Theta^*_{k^*}), p(x|\Theta^h_{k^h}(\gamma))) = L(\Theta^h_{k^h}(\gamma)) - L(\Theta^*_{k^*})$, according to (26), we have

$$0 \leq D_{KL}(p(x|\Theta^*_{k^*}), p(x|\Theta^h_{k^h}(\gamma))) < \gamma M. \qquad (27)$$

When $\gamma \to 0$, $D_{KL}(p(x|\Theta^*_{k^*}), p(x|\Theta^h_{k^h}(\gamma))) = 0$, i.e., $p(x|\Theta^*_{k^*}) = p(x|\Theta^h_{k^h}(\gamma))$ under information theory. Based on the identifiability of the finite mixture model, we then have $\Theta^h_{k^h}(\gamma) \supseteq \Theta^*_{k^*}$ with $k^h \geq k^*$ and the mixing probabilities of the other $k^h - k^*$ components in $\Theta^h_{k^h}(\gamma)$ being zeros, when the regularization factor $\gamma \to 0$.

References

[1] W. Abd-Almageed and L. S. Davis. Density estimation using mixtures of mixtures of Gaussians. In *Proc. ECCV*, volume 4, pages 410–422, 2006. 1

[2] M. Brand. Structure learning in conditional probability models via entropic prior and parameter extinction. *Neural Computation*, 11:1155–1182, 1999. 1

[3] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002. 2, 5

[4] M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002. 1, 2, 5, 6

[5] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, volume 2, pages 416–423, 2001. 2, 5, 7

[6] G. McLachlan. On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Journal of the Royal Statistical Society Series C*, 36:318–324, 1987. 1

[7] G. McLachlan and D. Peel, editors. *Finite Mixture Models*. John Wiley & Sons, New York, 2000. 1

[8] C. Nikou, N. P. Galatsanos, and A. C. Likas. A class-adaptive spatially variant mixture model for image segmentation. *IEEE Trans. on Image Processing*, 16(4):1121–1130, 2007. 5

[9] R. A. Render and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, 1984. 1

[10] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978. 1

[11] S. Roberts, C. Holmes, and D. Denison. Minimum-entropy data partitioning using reversible jump Markov chain Monte Carlo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(8):909–914, 2001. 1

[12] A. N. Tikhonov and V. Y. Arsenin, editors. *Solutions of Ill-Posed Problems*. Wiley, New York, 1977. 1

[13] N. Ueda and R. Nakano. Deterministic annealing EM algorithm. *Neural Networks*, 11:271–282, 1998. 2, 4

[14] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(6):929–944, 2007. 5, 7

[15] C. S. Wallace and D. L. Dowe. Minimum message length and Kolmogorov complexity. *Computer Journal*, 42(4):270–283, 1999. 1