Entropy Penalized Learning for Gaussian Mixture Models

Boyu Wang, Feng Wan, Peng Un Mak, Pui In Mak, and Mang I Vai

Abstract—**In this paper, we propose an entropy penalized approach to address the problem of learning the parameters of Gaussian mixture models (GMMs) with components of small weights. In addition, since the method is based on minimum message length (MML) criterion, it can also determine the number of components of the mixture model. The simulation results demonstrate that our method outperform several other state-of-art model selection algorithms especially for the mixtures with components of very different weights.**

I. Introduction

As a flexible and powerful statistical tool, finite mixture models, in particular Gaussian mixture models (GMMs) [1] have been extensively studied and used in the various domains such as pattern recognition, image analysis, computer version due to their computational tractability, ease to implement, and capability of representing arbitrarily complex probability density function with high accuracy. The general approach to the parameter estimation in mixture models is expectation-maximization (EM) algorithm [2], which converges to a maximum likelihood of mixture parameters and is easy to programming. However, the EM algorithm suffers two well-known difficulties: it may converge to local maxima of the log-likelihood function and thus be sensitive to initial conditions. On the other hand, the number of the components of mixture models (model order) is assumed to be known beforehand, but this prior knowledge is not usually available for many practical applications.

For the first problem, some techniques have been proposed along two lines. The first one focus on finding a good initialization of the parameters in mixture models to avoid the local maxima. The most straightforward approach is to re-run EM algorithm numerous times with different initialized parameters and select the model with maximum likelihood. However, the computation of this procedure is usually laborious. Some improved approaches based on the output of classification EM [3] or moment estimation [4] have also been proposed to choose the initial values for EM algorithm. A comparison study of some initialization methods for EM algorithm can be found in [5]. The other approach to alleviate the local optima problem is to modify learning process of the EM algorithm. (e.g., [6], [7]).

Another limitation of maximum likelihood by EM is that it cannot automatically select the appropriate number of components for mixture models, since the value of likelihood function will always increase when a new component is added. One choice for model selection is the Bayesian method which introduces the prior distributions over the number of the components and parameters of each component in mixtures. One drawback of this method is the intractable integration. A common approach to handle this problem is the Markov chain Monte Carlo (MCMC) based technique [24], which is, however, computational demanding. Alternatively, the use of variational approximation has been proposed to maximize the marginal likelihood of the data so that the values of the mixing coefficients can be optimized [12], [25].

Another approach to choose the model order is to formulate a cost function based on complexity criteria, usually in the form of a likelihood term plus penalty term to control the complexity of the models (i.e., the number of the components), such as the Akaike information criterion (AIC) [8], Bayesian information criterion (BIC) [9], the minimum message length (MML) [10], [11], integrated classification criterion (ICL) [21]. Some criteria based on entropy penalized likelihood have also been proposed. In [22], the likelihood is regularized by the Shannon entropy of mixture coefficients so that the model complexity can be controlled. The regularized EM (REM) algorithm is proposed in [23], which penalizes the likelihood function with the mutual information between the missing data and the incomplete data. The motivation of this method is to reduce the uncertainty of missing data. Therefore, the REM can also determine the model order since the simpler model will reduce the uncertainty. Similarly, by introducing a regularized term defined as the entropy of posterior probability, entropy regularized likelihood (ERL) is proposed in [16] to balance the data fitting and the model complexity.

Given a cost function, the most straightforward approach to select the number of the components is to repeat EM algorithm for mixture models with different order and the best model is obtained by comparing the cost function. However, this method is too computationally demanding, and also suffers from the local optimum problem. To handle this problem, one common approach is to integrate fitting data and model selection simultaneously rather than successively, and one advantage of this strategy is that it can also alleviate the sensitivity to the initial condition. This is mainly because of the gradual optimization procedure by adding or killing the components. In general, this learning strategy can be further divided into two subcategories: the incremental (bottom-up) method (e.g., [12] – [15]), and the decremental (top-down) method (e.g. [11], [16], [17]). Compared with top-down method, one shortcoming of incremental algorithm is that there is no consensus on the choice of appropriate split criteria, nor comprehensive theoretical analysis of different

The authors are with the Department of Electrical and Electronics Engineering, Faculty of Science and Technology, University of Macau, Av. Padre Tomás Pereira,Taipa, Macau. (e-mail:ma76533@umac.mo).

978-1-4244-9637-2/11/$26.00 ©2011 IEEE

split criteria. More important, the split of one component into two components is an ill-posed problem. Although some methods have been introduced, how to split the components is still an open problem.

Based on the considerations discussed above, we prefer the top-down methods to develop the learning algorithm for simultaneously estimate the parameter of mixture models and determining the number of components. Among the proposed algorithms, one elegant and interesting approach is developed in [11], where the estimation and model selection is integrated based on a MML-like criterion. On the other hand, optimizing the proposed criterion, from Bayesian point of view, can be also regarded as searching for the maximum a posteriori (MAP) solution of the parameters, in which the Dirichlet prior is imposed on the mixing coefficients. In addition, this approach has been proved very effective, but there are still two problems: the learning procedure is repeated until only one component left, then the model with smallest value of MML criterion is selected as the output of the algorithm. If the number of the component is too large, this algorithm is relative inefficient. More important, since the component with smallest weight is annihilated compulsively, a component which is already well adjusted to a subset of data produced by a low weight may get forced to zero, instead of an unnecessary heavier component almost overlapping another one [11]. Actually, most of existing methods also suffers from the latter problem (i.e., they cannot detect the components of small weights), and this can be viewed as the third problem of learning GMMs.

To overcome these drawbacks, we propose an entropy penalized learning algorithm, which has the following advantages: 1) The proposed method is able to simultaneously learn the parameter of GMMs and perform the model selection. 2) Compared with proposed algorithm in [11], the number of iteration in the learning procedure is reduced significantly, and the futile operation is also avoided. 3) Our algorithm can also handle small subsets of data.

The reminder of this paper is organized as follows. In Section II, we briefly review the EM algorithm for mixture model and MML criterion. The details of the proposed method are presented in Section III, and experimental results are reported in Section IV. Finally, some discussions and conclusions are provided in Section V.

I. Learning the Mixture Models and MML

Suppose we have a data set $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, consisting of N independent identical distributed (i.i.d.) observations of a random d-dimensional variable $\mathbf{x}$. If it follows a K-component finite mixture distribution, its probability density function (pdf) can be given by

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x} \mid \boldsymbol{\theta}_k), \text{ with } 0 \le \pi_k \le 1 \text{ and } \sum_{k=1}^{K} \pi_k = 1 \quad (1)$$

where π_k is the mixing coefficient, and $\boldsymbol{\theta}_k$ is the set of parameters for the kth component.

Define $\boldsymbol{\Theta} \equiv \{\pi_1, ..., \pi_K, \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K\}$ as the complete set of the parameters needed to specify the mixture model. The optimal set of the parameters is usually estimated by maximizing the log-likelihood of the pdf

$$\log p(\mathbf{X} \mid \boldsymbol{\Theta}) = \sum_{n=1}^{N} \log p(\mathbf{x}_n \mid \Theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k p(\mathbf{x}_n \mid \boldsymbol{\theta}_k) \quad (2)$$

It is well known that the maximum likelihood estimation cannot be obtained in a closed form. Hence, we need to resort the optimization techniques, and one common choice is the EM algorithm, which generates a sequence of the estimation of the set of the parameters by alternately applying the E-step and M-step until convergence.

However, the solution of the objective function (2) estimated by EM algorithm cannot determine the number of components. To overcome this problem, one common approach is to formulate a criterion which balances the data fitting and model complexity. In [11], a criterion based on the approximation of MML is developed

$$L(\boldsymbol{\Theta}, \mathbf{X}) = \frac{M}{2} \sum_{k:\pi_k > 0} \log\left(\frac{N\pi_k}{12}\right) + \frac{K_{nz}}{2} \log\left(\frac{N}{12}\right) + \frac{K_{nz}(M+1)}{2} - \log p(\mathbf{X} \mid \boldsymbol{\Theta}) \quad (3)$$

where M is the number of the parameters specifying each component, K_{nz} is the number of components with non-zero probability; see [11] for more details in derivation of the criterion.

II. Algorithm Description

A. Entropy Penalized Based Learning Approach

Based on the MML-like criterion and the constraints in (1), the update equations for means and covariance matrices are the same as the conventional EM algorithm, and the update of the mixing coefficients in the EM algorithm is given by [11]

$$\pi_k = \frac{\max\left\{0, \left(\sum_{n=1}^{N} p(k \mid \mathbf{x}_n)\right) - \frac{M}{2}\right\}}{\sum_{k=1}^{K} \max\left\{0, \left(\sum_{n=1}^{N} p(k \mid \mathbf{x}_n)\right) - \frac{M}{2}\right\}} \quad (4)$$

Since the component annihilation in (4) does not take into account the additional decrease in $L(\boldsymbol{\Theta}, \mathbf{X})$ caused by the decrease in K_{nz}, the component with smallest weight needs to be forced to zero during the learning procedure. In other words, most of the components are eliminated compulsively rather than removed by (4), which leads to the failure of fitting the data with small subset. In addition, the learning procedure is repeated until only one component left, which is relative inefficient if the number of components is large.

To overcome these drawbacks, we consider the following objective function:

$$\tilde{F}(\boldsymbol{\Theta}, \boldsymbol{\alpha}, \mathbf{X}) = \frac{M}{2} \sum_{k:\tilde{\pi}_k > 0} \log\left(\frac{N\tilde{\pi}_k}{12}\right) + \frac{K_{nz}}{2} \log\left(\frac{N}{12}\right) + \frac{K_{nz}(M+1)}{2} - \log \tilde{p}(\mathbf{X} \mid \boldsymbol{\Theta}) \quad (5)$$

where $\tilde{\pi}_k = \alpha_k \pi_k$, $0 \le \alpha_k \le 1$, and $\log \tilde{p}(\mathbf{X} \mid \boldsymbol{\Theta})$ is given by

$$\log \tilde{p}(\mathbf{X} \mid \boldsymbol{\Theta}) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \tilde{\pi}_k p(\mathbf{x}_n \mid \boldsymbol{\theta}_k) \quad (6)$$

As will be shown in the later experiments, the introducing

of $\{\alpha_k\}$ can not only avoid the futile operation, but also circumvent the problem of premature annihilation.

B. Interpretation

If (6) is a log-likelihood function, we should constrain $\sum_{k=1}^{K}\tilde{\pi}_k = 1$. With $0 \le \pi_k \le 1$, $\sum_{k=1}^{K}\pi_k = 1$, and $0 \le \alpha_k \le 1$, however, $\sum_{k=1}^{K}\tilde{\pi}_k = 1$ is satisfied only when $\alpha_k = 1$ for all $\pi_k > 0$, which gives the trivial solution for $\{\alpha_k\}$. Hence, we do not impose the constraint $\sum_{k=1}^{K}\tilde{\pi}_k = 1$, and this can be justified by introducing an auxiliary component $p(\mathbf{x}\,|\,\boldsymbol{\theta}_{k+1})$ such that $p(\mathbf{x}_n\,|\,\boldsymbol{\theta}_{k+1}) \le \varepsilon,\ \forall\, n = 1,...,N$, with ε an arbitrarily small value, and the corresponding mixing coefficient is $\tilde{\pi}_{K+1}$ such that $\sum_{k=1}^{K+1}\tilde{\pi}_k = 1$ [17].

Suppose that $p(\mathbf{x}\,|\,\boldsymbol{\theta}_{k+1})$ generates N' samples, then the criterion is

$$\tilde{F}'(\boldsymbol{\Theta},\boldsymbol{\alpha},\mathbf{X}) = \frac{M}{2}\sum_{k:\tilde{\pi}_k>0}\log\left(\frac{(N+N')\tilde{\pi}_k}{12}\right) + \frac{K_{nz}}{2}\log\left(\frac{(N+N')}{12}\right) + \frac{K_{nz}(M+1)}{2} - \log\tilde{p}'(\mathbf{X}\,|\,\boldsymbol{\Theta}) \quad (7)$$

where

$$\log\tilde{p}'(\mathbf{X}\,|\,\boldsymbol{\Theta}) = \sum_{n=1}^{N}\log\left(\sum_{k=1}^{K}\tilde{\pi}_k p(\mathbf{x}_n\,|\,\boldsymbol{\theta}_k) + \tilde{\pi}_{K+1}p(\mathbf{x}_n\,|\,\boldsymbol{\theta}_{K+1})\right) + \sum_{n=N+1}^{N+N'}\log\left(\sum_{k=1}^{K}\tilde{\pi}_k p(\mathbf{x}_n\,|\,\boldsymbol{\theta}_k) + \tilde{\pi}_{K+1}p(\mathbf{x}_n\,|\,\boldsymbol{\theta}_{K+1})\right) \approx \sum_{n=1}^{N}\log\sum_{k=1}^{K}\tilde{\pi}_k p(\mathbf{x}_n\,|\,\boldsymbol{\theta}_k) + \sum_{n=N+1}^{N+N'}\log\tilde{\pi}_{K+1}p(\mathbf{x}_n\,|\,\boldsymbol{\theta}_{K+1}) \quad (8)$$

Clearly, $\log\tilde{p}'(\mathbf{X}\,|\,\boldsymbol{\Theta})$ is the log-likelihood function, and the only difference between $\log\tilde{p}(\mathbf{X}\,|\,\boldsymbol{\Theta})$ and $\log\tilde{p}'(\mathbf{X}\,|\,\boldsymbol{\Theta})$ is the term $\sum_{n=N+1}^{N+N'}\log\tilde{\pi}_{K+1}p(\mathbf{x}_n\,|\,\boldsymbol{\theta}_{K+1})$, which is independent of $\tilde{\pi}_k,\ \forall\, k = 1,...,K$. Accordingly, optimizing (8) via the EM, we obtain the similar formulation for means and covariance matrices, except the mixing coefficients, which is given by

$$\tilde{\pi}_k \approx \frac{1}{N+N'}\sum_{n=1}^{N}\tilde{p}(k\,|\,\mathbf{x}_n),\quad \forall\, k = 1,...,K \quad (9)$$

where

$$\tilde{p}(k\,|\,\mathbf{x}_n) \approx \frac{\tilde{\pi}_k N(\mathbf{x}_n\,|\,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K}\tilde{\pi}_j N(\mathbf{x}_n\,|\,\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)},\quad \forall\, n = 1,...,N \quad (10)$$

and

$$\tilde{\pi}_k \approx \frac{N'}{N+N'},\quad k = K+1 \quad (11)$$

Obviously, the estimation of the parameters (the means and the covariance matrices) of mixture model will not be affected by the additional component, and the only modification is given by

$$\tilde{\pi}_k = \frac{N}{N+N'}\pi_k,\quad \forall\, k = 1,...,K \quad (12)$$

which provides a trivial solution for α_k (i.e., $\alpha_k = \frac{N}{N+N'}$, $\forall\, k = 1,...,K$).

To avoid the trivial solution, ignoring the N' samples generated by the auxiliary component, we now consider the problem in another way by imposing the following penalized term on $\{\alpha_k\}$:

$$H(\boldsymbol{\alpha}) = -\sum_{k=1}^{K}\left(\alpha_k\log\alpha_k + (1-\alpha_k)\log(1-\alpha_k)\right) \quad (13)$$

which represent the uncertainty of $\{\alpha_k\}$. H yields the maximum value when $\alpha_k = 1/2,\ \forall\, k = 1,...,K$. On the contrary, H decreases when $\{\alpha_k\}$ are forced to be zeros or ones.

Finally, with the binary entropy function as the penalized term, we have the following objective function

$$\tilde{L}(\boldsymbol{\Theta},\boldsymbol{\alpha},\mathbf{X}) = \frac{M}{2}\sum_{k:\tilde{\pi}_k>0}\log\left(\frac{N\tilde{\pi}_k}{12}\right) + \frac{K_{nz}}{2}\log\left(\frac{N}{12}\right) + \frac{K_{nz}(M+1)}{2} - \log\tilde{p}(\mathbf{X}\,|\,\boldsymbol{\Theta}) - \beta NH(\boldsymbol{\alpha}) \quad (14)$$

with $\tilde{\pi}_k = \alpha_k\pi_k$, $0 \le \alpha_k \le 1$, $0 \le \pi_k \le 1$, and $\sum_{k=1}^{K}\pi_k = 1$, where β is the penalty factor which is determined by experience.

C. The Complete Entropy Penalized EM Algorithm

Given the objective function (14), the parameters can be updated via a modified version of component-wise EM (CEM^2) [11], [18]. The description of the proposed entropy penalized EM algorithm can be summarized in Fig. 1.

Compared with the algorithm in [11], it can be observed that the equations for mixing coefficients, means, and covariance matrices in our algorithm are similar with the algorithm proposed in [11]. The main difference lies in the E-step, in which the posterior probability $p(k\,|\,\mathbf{x}_n)$ is calculated based on $\tilde{\pi}_k = \alpha_k\pi_k$ rather than π_k. In the subsequent M-step, α_k and π_k are updated respectively. The formulas for the parameters are almost same as the algorithm in [11].

For the auxiliary facotors $\{\alpha_k\}$, noting that compared with $\{\pi_k\}$, we do not impose the constraint $\sum_{k=1}^{K}\alpha_k = 1$. Therefore, setting the derivatives of $\tilde{L}$ in (14) with respect to α_k to zero, we obtain

$$\frac{\partial\tilde{L}}{\partial\alpha_k} = \frac{M}{2\alpha_k} - \sum_{n=1}^{N}\frac{\pi_k p(x_n\,|\,\boldsymbol{\theta}_k)}{\sum_{j=1}^{K}\alpha_j\pi_j p(x_n\,|\,\boldsymbol{\theta}_j)} + \beta N\log\frac{\alpha_k}{1-\alpha_k} = 0 \quad (15)$$

which gives

Algorithm: *Entropy Penalized EM Algorithm*

Input: Data Matrix $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, penalty factor β

Output: Optimal mixture model: $K, \boldsymbol{\Theta}(t) \equiv \{\pi_1, ..., \pi_K, \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K\}$, and $\hat{\boldsymbol{\alpha}}(t) = \{\alpha_k\}$

Procedure:

Initialize the parameters of the mixture model $\boldsymbol{\Theta}$ and the accelerating factors $\{\alpha_k\}$, $K_{\text{nz}} = K_{\max}$, $t = 0$.

Repeat

$t = t+1$

for $k = 1$ to $K_{\max}$, do

E-Step:

$$p(k \mid \mathbf{x}_n) = \alpha_k \pi_k p(x_n \mid \boldsymbol{\theta}_k) \Big/ \sum_{j=1}^{K} \alpha_j \pi_j p(x_n \mid \boldsymbol{\theta}_j)$$

M-Step:

Calculate π_k according (4); calculate α_k

if $\pi_k > 0$

$$\hat{\boldsymbol{\theta}}_k = \arg\max_{\boldsymbol{\theta}_k} \log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})$$

else

$K_{\text{nz}} = K_{\text{nz}} - 1$

end if

end for

Update the parameters of the mixture model $\boldsymbol{\Theta}(t)$ and $\hat{\boldsymbol{\alpha}}(t)$. Update $\tilde{L}(t)$ according (14)

Until $\tilde{L}(t-1) - \tilde{L}(t) < \varepsilon \left| \tilde{L}(t-1) \right|$ or some other stop criterion is met

Return the model parameter set: $K, \boldsymbol{\Theta}$

Fig. 1. The entropy penalized EM algorithm

$$\alpha_k \log \frac{\alpha_k}{1-\alpha_k} = \frac{\left(\sum_{n=1}^{N} p(k \mid \mathbf{x}_n) \right) - \frac{M}{2}}{\beta N} \tag{16}$$

Finally, using a softmax function $\alpha_k = e^{\gamma_k} / (1 + e^{\gamma_k})$, we have

$$\left(\gamma_k - \frac{n_k}{\beta N} \right) e^{\gamma_k - \frac{n_k}{\beta N}} = \frac{n_k}{\beta N} e^{-\frac{n_k}{\beta N}} \tag{17}$$

where

$$n_k = \max\left\{0, \left(\sum_{n=1}^{N} p(k \mid \mathbf{x}_n) \right) - \frac{M}{2} \right\} \tag{18}$$

(17) is a transcendental equation, which can be solved by resorting Lambert W function [19], and if $n_k = 0$, we just simply set $\alpha_k = 0$ (the value of α_k cannot affect the result when $n_k = 0$, since $\pi_k = 0$). Since the component annihilation is carried out only according to (4), the component with lowest weight is not compulsively removed during the learning process, which enables the algorithm to fit the data with low weight component, as shown in the following section

III. EXPERIMENTS

In this section, we evaluate the performance of the proposed algorithm on synthetic data sets. Moreover, we also compare our approach (we refer to it as EEM) with Figueiredo-Jain algorithm (FJ-EM) [11], deterministic annealing based model selection method (DAMS) [17], greedy EM method [13], as well as variational component splitting method (VCS) [12].

The first synthetic data set consists of 2000 samples from a mixture of eight two-dimensional Gaussian component (see also [20], [14]), where

$\boldsymbol{\mu}_1 = [1.5,\ 0]^T$ $\boldsymbol{\mu}_2 = [1,\ 1]^T$ $\boldsymbol{\mu}_3 = [0,\ 1.5]^T$ $\boldsymbol{\mu}_4 = [-1,\ 1]^T$
$\boldsymbol{\mu}_5 = [-1.5,\ 0]^T$ $\boldsymbol{\mu}_6 = [-1,\ -1]^T$ $\boldsymbol{\mu}_7 = [0,\ -1.5]^T$ $\boldsymbol{\mu}_8 = [1,\ -1]^T$

and

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_5 = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad \boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_7 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.01 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_4 = \boldsymbol{\Sigma}_6 = \boldsymbol{\Sigma}_8 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

Fig. 2 demonstrates the true mixtures with different mixing coefficients, and the results of EEM. The weights of the components $\boldsymbol{\mu}_4$ and $\boldsymbol{\mu}_8$ are reduced from 0.125 to 0.04, while the other six components have the same weight. It can be observed that even when the weights are small, we still need Gaussians to fit the components. We repeat the experiments 100 times for each set of weights of the components, and the results of each algorithm are shown in Table 1. It can be seen that the EEM can successfully fit the components with small

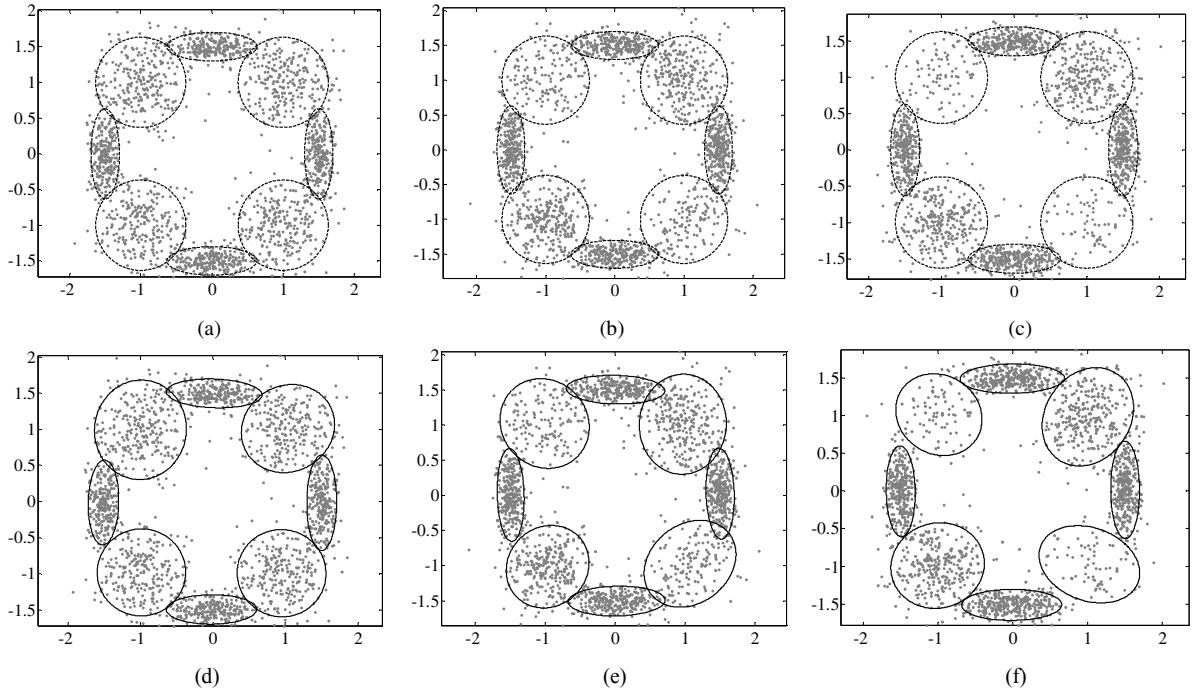

Fig. 2. The mixture models with different weights and the corresponding results of EEM: (a),(d) for μ_4 and μ_8 with the weights of 0.125; (b),(e) for μ_4 and μ_8 with the weights of 0.07; (c),(f) for μ_4 and μ_8 with the weights of 0.0

weights, while the FJ-EM fails when the coefficients of μ_4 and μ_8 are becoming smaller. In addition, when the coefficients are smaller, the number of the components detected by the algorithm is 9 or 10, rather than 6 or 7. In other words, the algorithm cannot interpret the mixtures correctly: the small components lead to redundant clusters, rather than being merged with other components. In [11], such failure case is attributed to premature annihilation. Since the weight of a smaller component are prematurely forced to zero, while heavier components overlap with each other, the algorithm may get trapped in local optima during the learning process. On the other hand, this condition can also be interpreted that if the number of samples of a Gaussian component is too small, the Gaussianity of the component for some algorithms is unobvious so that it may need more Gaussian components to model the non-Gaussian one.

DAMS and greedy EM also achieves good result for this example. However, due to the annealing and search scheme, the execution time of DAMS and greedy EM is much larger than other algorithms. The learning strategy of EEM is not to eliminate the components compulsively, and therefore it can avoid the local optima caused by premature annihilation.

The second example, as shown in Fig.3, consists of 1000 samples from four Gaussian components, which is much more complicated, since two of the four components share a common mean, but have different covariance matrices. Besides, the mixing coefficient of one component is small, and overlaps with another component. The parameters of the mixture are

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = [-4,-4]^T \quad \boldsymbol{\mu}_3 = [2,\ 2]^T \quad \boldsymbol{\mu}_4 = [-1,\ -6]^T$$

and

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 6 & -2 \\ -2 & 6 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \quad \boldsymbol{\Sigma}_4 = \begin{bmatrix} 0.125 & 0 \\ 0 & 0.125 \end{bmatrix}$$

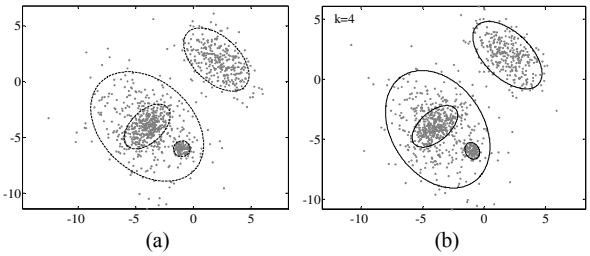

Fig. 3. The four-(overlapped) component mixture models (a) and estimated result of EEM (b)

Fig. 4 illustrates the modeling successful rates of different methods for the mixtures with different mixing coefficients. The weight of the μ_4 is decreased from 0.12 to 0.05, and the other three components have the same weight. We also run the experiment 100 times for each set of weights of the components. It can be seen that the proposed method obtain the best result in term of determining the number of the components. As the weight of μ_4 is becoming smaller, the performances of other methods are deteriorated significantly while EEM achieves robust results. The EEM takes the advantages of MML criterion while circumvent the premature annihilation in FJ-EM algorithm so that the local optima problem is alleviated. Compared with FJ-EM, since the

TABLE I
THE NUMBER OF COMPONENTS ESTIMATED BY DIFFERENT METHODS FOR MIXTURES WITH DIFFERENT WEIGHTS

Methods	FJ-EM				greedy-EM			DAMS			VCS			EEM		
Weights / Components	8	9	10	11	8	9	10	8	9	10	8	9	10	8	9	10
0.125	100				95	5		100			95	4	1	100		
0.1	100				93	6	1	100			94	5	1	100		
0.9	100				95	5		100			94	6		100		
0.8	100				91	7	2	100			89	10	1	100		
0.7	85	14	1		95	5		98	2		84	16		100		
0.6	65	25	9	1	94	6		95	5		86	10	4	100		
0.5	57	30	11	2	92	8		97	3		83	16	1	97	3	
0.4	46	30	16	8	91	9		93	7		82	18		93	6	1

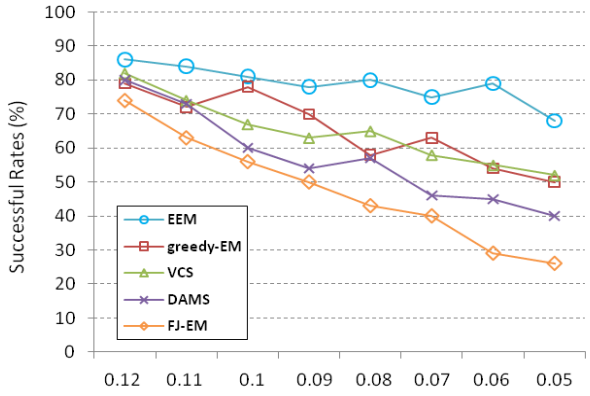

Fig. 4 The successful rates of different methods (the weight of μ_4 is decreased 0.12 to 0.05).

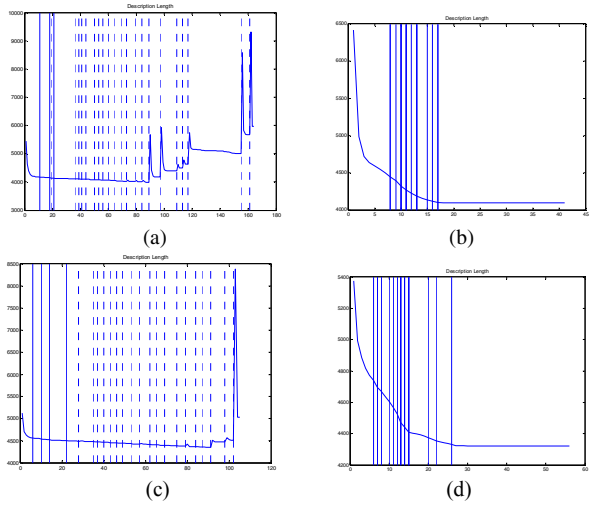

Fig. 5. The evolution of the cost functions of FJ-EM ((a), (c)) and EEM ((b), (d)) for the two examples. The vertical solid lines indicate the annihilation of one or more components by the CEM^2 algorithm, and the vertical dotted the component being eliminated compulsively

component in EEM is forced to zero automatically rather than compulsively, premature annihilation, as well as futile operation, can be avoid, so that EEM can model the components with small weight.

Because of the auxiliary factor α_k, the mixing coefficients are forced to the boundary of the parameter space much faster than the learning algorithm based on MML [11] (i.e., need less iterations). Fig. 5 illustrates the evolution of the cost function of FJ-EM and EEM for the two examples, from which it can be observed that the count of iterations of EEM is much less than FJ-EM. As top-down methods, EEM, FJ-EM, and DAMS are all initialized with 25 components. Due to the deterministic annealing scheme, the computational time of DAMS is one or two order of magnitude slower than EEM and FJ-EM, and it mainly depends on the annealing schedule. Although the update of α_k needs to resort to solving Lambert W function, the execution time of EEM is still faster than FJ-EM (roughly 2 to 5 times), especially when the number of components of mixture models is large, since the count of iterations of EEM is much less than FJ-EM.

IV. DISCUSSIONS AND CONCLUSIONS

This paper proposes an entropy penalized learning approach for modeling mixtures. The motivation of the proposed method is to avoid the compulsive elimination and futile operation. Since the components are automatically faded out by CEM^2 inside EEM, the premature annihilation problem caused by compulsive elimination in [11] can be circumvented. As a result, the proposed algorithm can fit the components with small weight. In addition, compared with other top-down algorithms, the amount of iteration and execution time of the proposed method are also decreased.

From Bayesian point of view, the penalty term $-\beta NH(\alpha)$ can be also viewed as an entropic-like prior of the auxiliary factor α_k, the strength of which is controlled by β:

$$P(\alpha_k) \propto \exp -\eta\left(\alpha_k \log \alpha_k + (1-\alpha_k)\log(1-\alpha_k)\right)$$
$$= \alpha_k^{-\eta\alpha_k}(1-\alpha_k)^{-\eta(1-\alpha_k)} \quad \text{with } \eta = N\beta \qquad (19)$$

The entropic-like priors of the auxiliary factor α_k with different ηs are shown in Fig.6, from which it can be observed that if β is too large, the penalty/prior will be strong enough to lead a trivial result (i.e., $\alpha_k = 1/2, \forall\ k = 1,\ldots,K$). On the other hand, if β is too small, the penalty will be too weak, and the objective function (14) will degenerate to MML criterion. As a result, the corresponding learning process will be similar with the algorithm proposed in [11] but without compulsive annihilation operation. In othr words, both too large and too small penalty factors can cause insufficient component elimination. We further investigated the selection of penalty factor β, and have found that when $\beta \in [0.03, 0.1]$, the EEM can achieve desirable result. The best performances are achieved as the value of β is chosen around 0.05, which is also the value we used in our experiments. It should also be noted that contrary to the priors of the mixture coefficients proposed in [11], [26], the priors favor fair estimated (i.e., $\alpha_k = 1/2$) instead of annihilation operation. At first glance, this

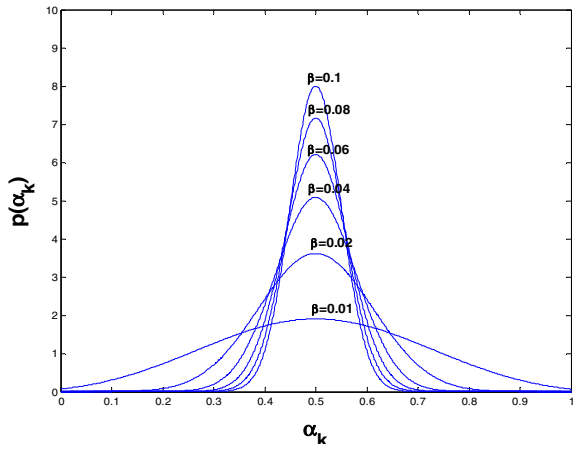

Fig. 6. Plot of the entropic-like priors of α_k in (19).

contradict the principle of top-down model selection algorithms, that is, the prior cannot kill the auxiliary factors $\{\alpha_k\}$. In fact, in the proposed method, our purpose is to eliminate $\{\pi_k\}$ rather than $\{\alpha_k\}$. In the learning process, it just because of $\{\alpha_k\}$ that the components are automatically forced to zero instead of compulsive eliminated, and the premature annihilation is avoided.

Even if the experimental results have demonstrated the superior performance of the proposed algorithm, the further theoretical analysis remains to be clarified, which may also provide some guidelines for the choice of penalty factor β. We also perform cross validation to evaluate performance of the proposed method in term of how well the data are fitted to the trained mixture model, and we investigated the failure cases of FJ-EM and the proposed method, and have found that FJ-EM tends to omit the component with small weight while fit other samples with redundant clusters, which may not affect the error rate much. On the other hand, the components learned by EEM may be totally inconsistent with the true mixtures. This phenomenon was also mentioned in [17], which also needs to be thoroughly analyzed. In addition, the proposed method should also be further evaluated on the real world data set.

ACKNOWLEDGMENT

The authors gratefully acknowledge support from the Macau Science and Technology Department Fund (Grant FDCT/036/2009/A) and the University of Macau Research Fund (Grants RG059/08-09S/FW/FST and RG080/09-10S/WF/FST).

REFERENCES

[1] G. McLachlan, and D. Peel, *Finite Mixture Models*. New Yorks: Wiley, 2000.

[2] G. McLachlan., and T. Krishnan, *The EM Algorithm and Extensions*. New Yorks: Wiley, 1997

[3] C. Biernacki, G. Celeux, and G. Govaert, "Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models," *Comput. Statist. Data Anal.*, vol. 41, pp. 561–575, 2003.

[4] W. D. Furman, and B. G. Lindsay, "Measuring the relative Effectiveness of moment estimators as starting values in maximizing likelihoods," *Comput. Statist. Data. Anal.*, vol. 17, pp. 493–507, 1994.

[5] D. Karlis, and E. Xekalaki, "Choosing initial values for the EM algorithm for finite mixtures," *Comput. Statist. Data Anal.*, vol. 41, pp. 577–590, 2003.

[6] N. Ueda, and R. Nakano, "Deterministic annealing EM algorithm," *Neural Netw.*, vol. 11, no. 2, pp. 271–282, 1998.

[7] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "SMEM algorithm for mixture models," *Neural Comput.*, vol.12, no.9, pp. 2109–2128, 2000.

[8] H. Akaike, "A new look at the statistical model identification," *IEEE Trans Aut. Ctrl.*, vol. 19, no.6, pp. 716-723, 1974.

[9] G. Schwarz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, pp. 461-464, 1978

[10] C. S. Wallace, and D. L. Dowe, "Minimum message length and Kolmogorov complexity," *Comput. J.,* vol. 42, no. 4, pp. 270–283, 1999.

[11] M. A. T. Figueiredo, and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 24, no. 3, pp. 381–396, Mar. 2002.

[12] C. Constantinopoulos, and A. Likas, "Unsupervised learning of Gaussian mixture based on variational component splitting," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 745–755, 2007.

[13] J. Verbeek, M. Vlassis, and B. Krose, "Efficient greedy learning of Gaussian mixture models," *Neural Comput.*, vol. 5, no. 2, pp. 469–485 2003.

[14] D. Ververidis, and C. Kotropoulos, "Gaussian mixture modeling by exploiting the Mahalanobis distance," *IEEE Trans. Signal Process.,* vol. 56, no. 7, pp. 2797–2811, Jul. 2008.

[15] A. P. Benavent, F. E. Ruiz, and J. M. Sáez, "Learning Gaussian mixture models with entropy-based criteria," *IEEE Trans. Neural Netw.,* vol. 20, no. 11, pp. 1756–1771, 2009.

[16] Z. Lu, and H. H. S. Ip, "Generalized competitive learning of Gaussian mixture model," *IEEE Trans. Syst., Man, Cybern. B.,* vol. 39, no. 4, pp. 901–909, Aug. 2009.

[17] Q. Zhao, and D. J. Miller, "A deterministic, annealing-based approach for learning and model selection in finite mixture models," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing.*, 2004, pp. 457–460.

[18] G. Celeux, S. Chrétien, F. Forbes, and A. Mkhadri, "A component-wise EM algorithm for Mixtures," Tech. Rep. 3746, INRIA Rhône-Alpes, France, 1999. Available at http://www.inria.fr/RRRT/RR-3746.html.

[19] R.M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the Lambert W function," *Adv. Comput. Math.,* vol. 5, pp. 329–359, 1996.

[20] B. Zhang, C. Zhang, and X. Yi, "Competitive EM algorithm for finite mixture models," *Pattern Recognit.,* vol. 37, no. 1, pp. 131–144, 2004.

[21] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated classification likelihood," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 22, no. 7, pp. 719–725, July 2000.

[22] J. Ma, T. Wang, "Entropy penalized automated model selection on Gaussian mixtures," *Int. J. Pattern Recognit. Artif. Intell.,* vol. 18, no. 8, pp. 1501–1512, 2004.

[23] H. Li, K. Zhang, and T. Jiang, "The regularized EM algorithm," in *Proc. 20th National Conf. Artificial Intelligence.,* 2005, pp. 807–812.

[24] S. Richardson, and P. Green, "On Bayesian analysis of mixtures with unknown number of components," *J. Roy. Stat. Soc. (B),* vol. 59, no. 4, pp. 731–792, 1997.

[25] A. Corduneanu, and C. M. Bishop, "Variational Bayesian model selection for mixture distributions," in *Artificial Intelligence and Statistics 2001*. San Mateo, CA: Morgan Kaufmann, 2001, pp. 27–34.

[26] M. Brand, "Structure learning in conditional probability models via entropic prior and parameter extinction," *Neural Comput.,* vol. 11, no. 5, pp. 1155–1182, Jul. 1999.