2014 IEEE International Conference on Systems, Man, and Cybernetics
October 5-8, 2014, San Diego, CA, USA

DYNAMICALLY REGULARIZED HARMONY LEARNING OF GAUSSIAN MIXTURES

Hongyan Wang, Jinwen Ma

Department of Information Science, School of Mathematical Sciences
Peking University, Beijing, 100871, China

ABSTRACT

In this paper, a dynamically regularized harmony learning (DRHL) algorithm is proposed for Gaussian mixture learning with a favourite feature of both adaptive model selection and consistent parameter estimation. Specifically, under the framework of Bayesian Ying-Yang (BYY) harmony learning, we utilize the average Shannon entropy of the posterior probability per sample as a regularization term being controlled by a scale factor to the harmony function on Gaussian mixtures increasing from 0 to 1 dynamically. It is demonstrated by the experiments on both synthetic and real-world datasets that the DRHL algorithm can not only select the correct number of actual Gaussians in the dataset, but also obtain the maximum likelihood (ML) estimators of the parameters in the actual mixture. Moreover, the DRHL algorithm is scalable and can be implemented on a big dataset.

Keywords: Gaussian mixtures; model selection; regularization; maximum likelihood.

1. INTRODUCTION

As a flexible and powerful statistical tool for data analysis and information processing, finite mixture model [1] has found its applications in many problems, such as clustering analysis, image segmentation and speech recognition. Among these applications, Gaussian mixtures are widely used and several statistical learning methods have been proposed to deal with this kind of models, such as the EM algorithm [2] and the method of moments [3]. They usually assumed that the number of Gaussians or clusters in a dataset is pre-known. However, in many instances this key information is not available. Then, the selection of an appropriate number of Gaussians, called model selection, must be made with the parameters learning in mixtures. Thus, the general Gaussian mixture modeling is actually a compound modeling problem of both parameter estimation and model selection, which is a rather complicated and difficult task [4].

The conventional way for solving this compound mixture modeling problem is to select an optimal number k^* of Gaussians as the clusters in the dataset via one of the information, coding and statistical selection criteria such as Akaike's Information Criterion [5], Bayesian Inference Criterion (BIC) [6], Minimum Description Length (MDL) [7], and Minimum Message Length (MML) [8]. However, the validating process of this method is computationally expensive because we need to repeat the entire parameter learning process at a large number of possible k.

Since the 1990s, some statistical learning approaches appeared to solve this problem. Dirichlet processes [9] and reversible jump Markov chain Monte Carlo (RJMCMC) [10] are two typical implementations of the first kind of approaches which uses stochastic simulations. These stochastic simulation methods generally relies on intensive sampling and are very time-consuming. The second kind is the Bayesian model search based on optimizing the variational bounds [11]. The third is unsupervised learning on finite mixtures [12, 13] which introduce certain competitive mechanism into the mixture model such that the model selection can be made adaptively during parameter learning with a simplified MML model selection criterion or using the RPCL mechanism.

Alternatively, the Bayesian Ying-Yang (BYY) harmony learning [14, 15, 16] has also provided a new statistical learning mechanism that makes model selection adaptively during parameter learning. It has already been implemented on Guassian mixture learning and several BYY harmony learning algorithms have also been established for Gaussian mixtures [17, 18, 19]. Although the BYY harmony learning owns the ability of adaptive model selction, its parameter estimation has a notable deviation from the ML estimation which is consistent with true parameters.

In order to solve this deviation problem, we consider the BYY harmony learning as the ML learning with a regularization term of average negative Shannon entropy of the posterior probability per sample using to control the scale or complexity of mixture models. From this point of view, we propose a dynamically regularized harmony learning (DRHL) algorithm for Gaussian mixtures by adding a Shannon entropy regularization term being controlled by a scale factor to the BYY harmony learning. As the scale factor is increased dynamically from 0 to 1, our proposed al-

Jinwen Ma, the corresponding author, Tel: 86-10-62760609, Email: jwma@math.pku.edu.cn.

978-1-4799-3840-7/14/$31.00 ©2014 IEEE

gorithm transforms from the BYY harmony learning with adaptive model selection into the conventional maximum likelihood learning so that the adaptive model selection and the ML estimation are both obtained at last.

The rest of the paper is organized as follows. We begin with a brief description of the BYY harmony learning system for Gaussian mixtures in Section 2. Then, we present the derivation and analysis of the dynamically regularized harmony learning algorithm for Gaussian mixtures in Section 3. Section 4 contains the experimental results on both synthetic and real-world datasets. Finally, we conclude briefly in Section 5.

2. BYY HARMONY LEARNING OF GAUSSIAN MIXTURES

The BYY harmony learning system describes each observation $x \in \mathscr{X} \subset \Re^n$ and its corresponding inner representation $y \in \mathscr{Y} \subset \Re^m$ via the two types of Bayesian decomposition of the joint density: $p(x,y) = p(x)p(y|x)$ and $q(x,y) = q(y)q(x|y)$, which are called Yang machine and Ying machine, respectively. Given a sample dataset $D_x = \{x_t\}_{t=1}^N$ from the Yang or observable space, the BYY harmony learning system is trying to extract the hidden probabilistic structure of x with the help of y from specifying all aspects of $p(y|x)$, $p(x)$, $q(x|y)$ and $q(y)$ by maximizing the following harmony functional:

$$H(p||q) = \int p(y|x)p(x)\ln[q(x|y)q(y)]dxdy. \quad (1)$$

If both $p(y|x)$ and $q(x|y)$ are parametric, the BYY learning system is called to have a Bi-directional Architecture (Bi-Architecture for short). Given a sample dataset $D_x = \{x_t\}_{t=1}^N$, the Bi-architecture of the BYY harmony learning system can be specified as follow. The inner representation y is discrete in $\mathscr{Y} = \{1, 2, \cdots, k\}$ (i.e., with $m = 1$), while the observation x is continuous from a Gaussian mixture distribution. On the Ying space, we let $q(y = j) = \pi_j \geq 0$ with $\sum_{j=1}^k \pi_j = 1$. This is a prior probability distribution for Gaussians or clusters of the mixture. On the Yang space, $p(x)$ is a latent probability density function (pdf) of Gaussian mixture from which D_x are generated. Moreover, in the Ying path, $q(x|y = j) = q(x|m_j, \Sigma_j)$ is assumed to be a Gaussian density function with mean vector m_j and the covariance matrix Σ_j, while in the Yang path, $p(y = j|x)$ is constructed under the Bayesian principle by the following parametric form,

$$p(y = j|x) = \frac{\pi_j q(x|m_j, \Sigma_j)}{q(x|\Theta_k)}, \quad (2)$$

$$q(x|\Theta_k) = \sum_{j=1}^k \pi_j q(x|m_j, \Sigma_j), \quad (3)$$

where $\Theta_k = \{\pi_j, m_j, \Sigma_j\}_{j=1}^k$ and $q(x|\Theta_k)$ is just a Gaussian mixture model that will approximate the latent $p(x)$ via the harmony learning on the BYY learning system.

Put all these components into Eq.(1), we have

$$H(p||q) = E_{p(x)}\left[\sum_{j=1}^k h_j(X)\ln[\pi_j q(X|m_j, \Sigma_j)]\right], \quad (4)$$

where

$$h_j(X) = \frac{\pi_j q(X|m_j, \Sigma_j)}{\sum_{i=1}^k \pi_i q(X|m_j, \Sigma_j)}. \quad (5)$$

That is, $H(p||q)$ is the expectation of a function of the random variable X subject to $p(x)$. With the sample dataset D_x, we get an estimate of $H(p||q)$, called harmony function, as follows:

$$J(\Theta_k) = \frac{1}{N}\sum_{t=1}^N \sum_{j=1}^k h_j(x_t)\ln[\pi_j q(x_t|m_j, \Sigma_j)]. \quad (6)$$

According to the theoretical and experimental results on this Bi-architecture of the BYY harmony learning system for Gaussian mixtures [20, 17, 18, 19], the maximization of $J(\Theta_k)$ is capable of making model selection adaptively during parameter learning when the actual Gaussians or clusters are separated in a certain degree. That is, if we choose k to be larger than the number (k^*) of actual Gausians or clusters in the sample data, the maximization of the harmony function can make k^* Gaussians to match the actual ones and simultaneously eliminate $k-k^*$ extra ones. However, as we mentioned previously, the original BYY harmony learning suffers from inconsistent parameter estimation. So, our work here is to use the regularization mechanism to transform the BYY harmony learning to the ML learning such that adaptive model selection and consistent parameter estimation can be made simultaneously.

3. DYNAMICALLY REGULARIZED HARMONY LEARNING ALGORITHM

3.1. The Dynamic Regularization Mechanism

According to [21], $J(\Theta_k)$ can be divided into two parts,

$$J(\Theta_k) = L(\Theta_k) - O_N(p(y|x)), \quad (7)$$

where the first part is just the log-likelihood function, i.e.,

$$L(\Theta_k) = \frac{1}{N}\sum_{t=1}^N \ln(\sum_{j=1}^k(\pi_j q(x_t|m_j, \Sigma_j))), \quad (8)$$

while the second is the average Shannon entropy of the posterior probability $p(y|x)$ over the sample dataset $\mathcal{D} = \{x_t\}_{t=1}^N$,

$$O_N(p(y|x)) = -\frac{1}{N}\sum_{t=1}^N \sum_{j=1}^k p(j|x_t)\ln p(j|x_t). \quad (9)$$

According to Eq.(7), if $-O_N(p(y|x))$ is viewed as a regularization term, the BYY harmony learning, i.e., maximizing $J(\Theta_k)$, is a regularized ML learning which has already been investigated in [22, 23] by scaling the regularization term with a small positive number. However, since they keep the regularization scale constant just as in the case of the BYY harmony learning, these investigations also suffer from inconsistent parameter estimation.

On the other hand, from Eq.(7) we also have

$$L(\Theta_k) = J(\Theta_k) + O_N(p(y|x)), \tag{10}$$

which indcates that the ML learning is a regularized (BYY) harmony learning with $O_N(p(y|x))$ as the regularization term. To control the regularization,a scale factor $\lambda(\geq 0)$ is introduced,

$$L_\lambda(\Theta_k) = J(\Theta_k) + \lambda O_N(p(y|x)). \tag{11}$$

If $\lambda = 0$, $L_\lambda(\Theta_k) = J(\Theta_k)$ is just BYY harmony function on the Bi-architecture for Gaussian mixtures. If $\lambda = 1$, $L_\lambda(\Theta_k)$ is the log-likelihood function of the Gaussian mixture model. That is, with λ increasing from 0 to 1, maximizing $L_\lambda(\Theta_k)$ changes from the harmony learning to the ML learning. Here we try to control the increasing of λ appropriately to realize adaptive model selection at the previous learning stage and the ML estimation at the final learning stage.

3.2. The Fixed-point Learning Algorithm

At each phase of the dynamically regularized harmony learning with a specific λ, we construct a fixed-point algorithm to maximize $L_\lambda(\Theta_k)$.

For convenience, we utilize the softmax representation for π_j, i.e., $\pi_j = e^{\beta_j}/\sum_{i=1}^K e^{\beta_i}$, $j = 1, \cdots, k$, where $\beta_j \in (-\infty, +\infty), j = 1, \cdots, k$. Letting the derivatives of $L_\lambda(\Theta_k)$ with respect to β_j, m_j and Σ_j, respectively, be zero, we get the following fixed-point (iterative) learning algorithm:

$$\hat{\pi}_j = \frac{\sum_{t=1}^N p(j|x_t)\gamma_j(t)}{\sum_{t=1}^N \sum_{i=1}^k p(i|x_t)\gamma_i(t)}; \tag{12}$$

$$\hat{m}_j = \frac{\sum_{t=1}^N p(j|x_t)\gamma_j(t)x_t}{\sum_{t=1}^N p(j|x_t)\gamma_j(t)}; \tag{13}$$

$$\hat{\Sigma}_j = \frac{\sum_{t=1}^N p(j|x_t)\gamma_j(t)(x_t - \hat{m}_j)(x_t - \hat{m}_j)^T}{\sum_{t=1}^N p(j|x_t)\gamma_j(t)}, \tag{14}$$

where

$$\gamma_i(t) = 1 - \sum_{l=1}^k (p(l|x_t) - \delta_{il}) \ln \pi_l p(x_t|m_l, \Sigma_l) + \lambda \sum_{l=1}^k (p(l|x_t) - \delta_{il}) \ln p(l|x_t), \tag{15}$$

where δ_{ij} is the Kronecker function.

In comparison with the conventional EM algorithm for Gaussian mixtures [2], our proposed fixed-point learning algorithm differs only at the augmenting term $\gamma_j(t)$. It can be easily verified that when $\lambda = 1$, $\gamma_j(t) = 1$, the fixed-point learning algorithm is just the EM algorithm and when $\lambda = 0$, the fixed-point learning algorithm returns to the original fixed-point BYY learning algorithm of maximizing the harmony function $J(\Theta_k)$.

Actually, $\gamma_j(t)$ implements a rival penalized competitive learning (RPCL) mechanism [24]-[25] so that model selection can be made adaptively during parameter learning. At the early learning stage, $\gamma_j(t) < 0$ may happen. According to Eq.(15), the mean vectors of j-th Gaussian will move away from x_t. Otherwise, if $\gamma_j(t) > 0$, the mean vectors of the j-th Gaussian will be attracted to x_t. So, for x_t, Gaussians with $\gamma_j(t) > 0$ are winners while these Gausians with $\gamma_j(t) < 0$ are losers.

However, the fixed-point learning algorithm cannot guarantee the positive definiteness of each covariance matrix during the iteration since $\gamma_j(t)$ may be negative. In order to overcome this problem, we use the EM update rule of the covariance matrixes, i.e., forcing all $\gamma_j(t) = 1$ in Eq.(14), in this specific case. In fact, this simplification is applicable and efficient since the competition for adaptive model selection is mainly among mean vectors and controlled by the mixing proportions.

3.3. The Dynamic Evolution of λ

We further discuss the dynamic evolution of λ with time T during the learning process. According to our regularization mechanism, λ should be very small and increase slowly at the early learning stage to realize adaptive model selection. Then, at the sequent stage, λ can go to 1 at a quicker speed and the algorithm will finally converge to a ML solution. So, it is crucial to check whether the adaptive model selection has accomplished and when to change learning stage.

In order to detect the turning point, we introduce the Shannon entropy of mixing proportions in Gaussian mixtures, $H_\pi = -\sum_{j=1}^k \pi_j \ln \pi_j$. Obviously, H_π is sensitive to the structure of a mixture model. If model selection is not completed, the difference of H_π between two iterations is considerable. Otherwise, the difference should be very small. This motivates us to adopt the absolute change rate of H_π between two iterations, defined by

$$h_\pi(T) = |\frac{H_\pi(T) - H_\pi(T-1)}{H_\pi(T)}|, \tag{16}$$

as an indicator of model selection. T is time, i.e., the number of iterations The whole learning process is divided into two learning stages according to a given threshold $\varepsilon_1(> 0)$ of this indicator. That is, if $h_\pi(T) > \varepsilon_1$, $\lambda(T)$ increases at

low speed; otherwise, it increases at high speed. Since $\lambda(T)$ is assumed to increase exponentially, its dynamic evolution process is given as follow,

$$\lambda(T) = \begin{cases} \lambda_0 * \eta_1^T, & \text{if } h_\pi(T) > \varepsilon_1; \\ \lambda_0 * (\frac{\eta_1}{\eta_2})^{T^*} \eta_2^T, & \text{if } h_\pi(T) \leq \varepsilon_1, \end{cases} \quad (17)$$

where λ_0 (being a very small positive constant) is initial value of λ, η_1, η_2 are two positive constants with constraint that $1 < \eta_1 < \eta_2$, and T^* is the turning point such that $h_\pi(T^*) > h_0$ and $h_\pi(T^*+1) \leq h_0$. When λ reaches 1, we fix it until the iteration converges.

3.4. The Complete DRHL Algorithm

We finally summary our proposed DRHL algorithm. Firstly, we should choose the parameters of the algorithm properly. As mentioned previously, λ_0, η_1, η_2 and ε_1 must be carefully selected to make the evolution of $\lambda(T)$ dynamic. θ_0 is a threshold value to filter out Gaussians with very small mixing proportions during the parameter learning process, while $\varepsilon_2(>0)$ is a threshold value to terminate the iteration. If $\lambda = 1$ and the absolute increment of the log likelihoods is smaller than ε_2, we affirm the convergence of the algorithm. In our learning paradigm, k is flexible. However, it should be larger than the number (k^*) of actual Gaussians or clusters in the dataset. As for the initial setting of the parameters Θ_k, i.e., $\Theta_k^{(0)} = \{\pi_i^0, m_i^0, \Sigma_i^0\}|_{i=1}^k$, some traditional clustering method may be helpful. For example, m_i^0 can be selected through a RPCL procedure [25] and then π_i^0 and Σ_i^0 can be estimated accordingly.

After initializing all the parameters, Θ_k will be updated in each phase of $\lambda(T)$ via the fixed-point learning algorithm given by Eqs (12)-(14). At the end of each learning phase, the Gaussians with the mixing proportions less than λ_0 are annihilated immediately. After $\lambda(T)$ reaches 1, the algorithm goes on until the log likelihood function reaches its maximum value or its absolute increment is less than ε_2.

Since the DRHL algorithm at each learning phase becomes the fixed-point learning algorithm which has the similar update rules as the EM algorithm, we can use the data summarization techniques suggested for the EM algorithm for Gaussian mixtures in [26] to make it scalable and be implemented on a big dataset. Therefore, the DRHL algorithm can be scalable and used on a big dataset.

4. EXPERIMENTAL RESULTS

In this section, various experiments are carried out on both synthetic and real-world datasets to demonstrate the performance of the dynamically regularized harmony learning (DRHL) algorithm for Gaussian mixtures. Moreover, it is compared with some typical existing learning algorithms. In these experiments, we always select $\varepsilon_1 = 1e-5$, $\varepsilon_2 = 1e-5$, $\eta_1 = 1.005$,$\lambda_0 = 0.001$ $\eta_2 = 2$ and $\theta_0 = 0.05$. The other parameters will be specified in the particular experiments.

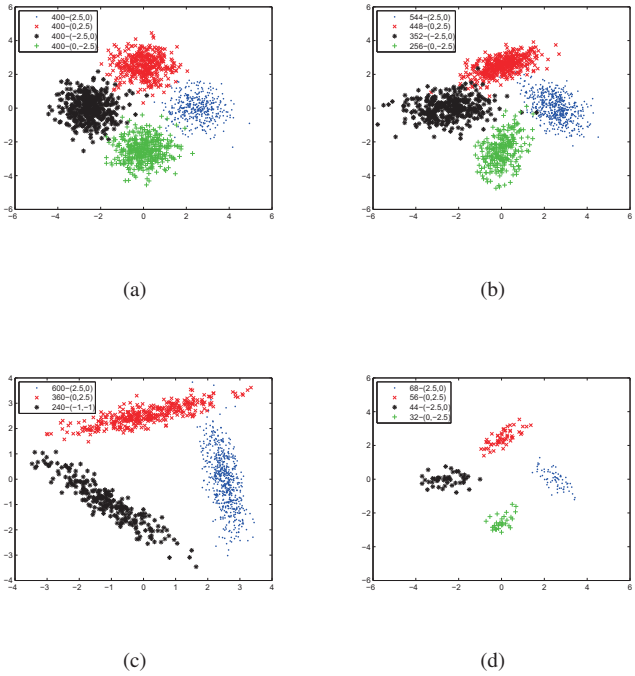


Fig. 1. Four synthetic datasets for simulation experiments. (a). $\mathcal{S}_1$, (b). $\mathcal{S}_2$, (c). $\mathcal{S}_3$, (d). $\mathcal{S}_4$.

4.1. Simulation Experiments

4.1.1. The Synthetic Datasets

We begin to generate four typical synthetic datatsets from mixtures of four or three bivariate Gaussian distributions on the plane coordinate system (i.e., $d = 2$). Clearly, these Gaussian distributions were either sphere-shaped or ellipse-shaped. As shown in Fig.1, the covariance matrices of Gaussian distributions are designed to demonstrate different degrees of overlap among Gaussians (i.e., clusters). Moreover, the four datasets are also generated with equal or unequal mixing proportions. The specific parameters for these four datasets are listed in Table1, where m_i, $\Sigma_i = (\sigma_{jk}^i)_{2\times 2}$, π_i, N_i denote the mean vector, covariance matrix, mixing proportion, and number of samples of the i-th Gaussian, respectively.

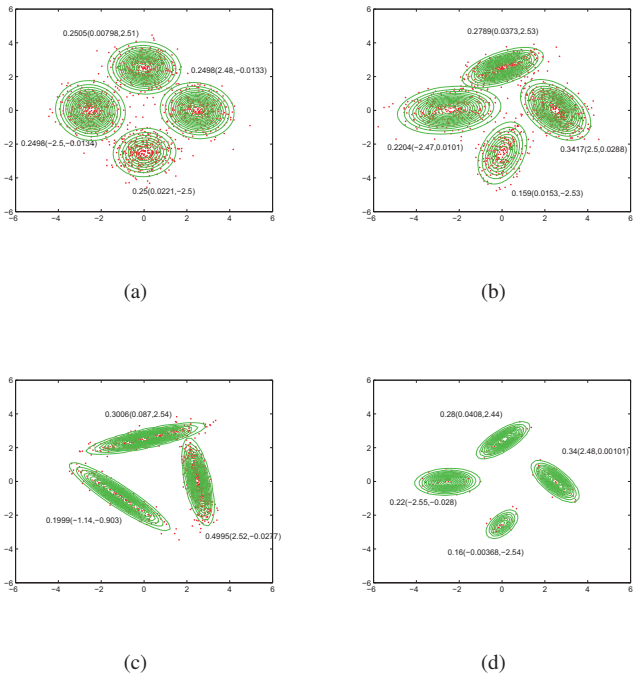
4.1.2. Simulation Results and Comparisons

The DRHL algorithm is conducted on each of these four synthetic datasets with $k = 2k^*$. For illustration, a group of typical experimental results are shown in Fig. 2. These

Table 1. The values of the parameters of the four synthetic datasets.

The dataset	Gaussian	m_i	σ_{11}^i	$\sigma_{12}^i(\sigma_{21}^i)$	σ_{22}^i	π_i	N_i
$\mathcal{S}_1$	G1	(2.50,0)	0.50	0.00	0.50	0.25	400
(N=1600)	G2	(0,2.50)	0.50	0.00	0.50	0.25	400
	G3	(-2.50,0)	0.50	0.00	0.50	0.25	400
	G4	(0,-2.50)	0.50	0.00	0.50	0.25	400
$\mathcal{S}_2$	G1	(2.50,0)	0.45	-0.25	0.55	0.34	544
(N=1600)	G2	(0,2.50)	0.65	0.20	0.25	0.28	448
	G3	(-2.50,0)	1.00	0.10	0.35	0.22	352
	G4	(0,-2.50)	0.30	0.15	0.80	0.16	265
$\mathcal{S}_3$	G1	(2.50,0)	0.10	-0.20	1.25	0.50	600
(N=1200)	G2	(0,2.50)	1.25	0.35	0.15	0.30	360
	G3	(-1,-1)	1.00	-0.80	0.75	0.20	240
$\mathcal{S}_4$	G1	(2.50,0)	0.28	-0.20	0.32	0.34	68
(N=200)	G2	(0,2.50)	0.34	0.20	0.22	0.28	56
	G3	(-2.50,0)	0.50	0.04	0.12	0.22	44
	G4	(0,-2.50)	0.10	0.05	0.50	0.16	32

figures tell that k^* Gaussians (demonstrated by their contour lines) are finally recognized and each estimated Gaussian matches the actual one accurately.



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

Fig. 2. Simulation Results of the DRHL algorithm on the four synthetic datasets, respectively.

We further compared the DRHL algorithm with the MML-based unsupervised learning algorithm particularly for Gaussian mixtures [12], being referred to as CEM^2 for short. Actually, CEM^2 has been considered as a typical and powerful learning algorithm for the Gaussian mixture learning with adaptive model selection in literature. To show the stability and accuracy of converged results, we implement both CEM^2(with the stop criterion $\epsilon = 10^{-6}$) and the DRHL algorithm on each of above four datasets for 50 times with different randomly selected initial parameters. We then compute the frequencies of correct model selection (CMS) and average runtime of these two algorithms over 50 trials on each dataset. The experimental results are listed in Table 2. Obviously, the DRHL algorithm considerably outperforms CEM^2 on both correct model selection and runtime.

In addition to model selection and runtime, we also compare the DRHL algorithm with the CEM^2 algorithm on the accuracy of parameter estimation. For each parameter θ_i, we define $\Delta\theta_i$ as the average absolute error of $|\theta_i - \theta_i^*|$ over 50 trials. For each dataset, we compute the total average absolute error per each parameter called TAE. Actually, the TAEs of the two algorithms on the four datasets are listed in Table 3. It can be found that the DRHL and CEM^2 algorithms have almost the same accuracy on parameter estimation. However, for the fourth type of datasets which demonstrates small sample data, the accuracy of the DRHL algorithm is remarkably better than that of the CEM^2 algorithm.

The DRHL algorithm is further compared with the BYY annealing algorithm (BYY-AEM) [27]. While the DRHL and BYY-AEM algorithms had the similar performance on adaptive model selection, the DRHL algorithm leads to a more accurate parameter estimation. Actually, the TAEs of the BYY-AEM algorithm on the four datasets are respectively 0.0204, 0.0243, 0.0386 and 0.0322, which are slightly

Table 2. The comparison of the DRHL and CEM^2 algorithms on model selection and runtime.

Datasets	DRHL		CEM^2	
	CMS Frequency	runtime(s)	CMS Frequency	runtime(s)
$\mathcal{S}_1$	100%	526	84%	11290
$\mathcal{S}_2$	100%	856	56%	1825
$\mathcal{S}_3$	100%	145	72%	4317
$\mathcal{S}_4$	96%	460	56%	554

Table 3. The comparison of the DRHL and CEM^2 algorithms on parameter estimation accuracy.

Dataset	DRHL	CEM^2
$\mathcal{S}_1$	0.0204	0.0204
$\mathcal{S}_2$	0.0171	0.0172
$\mathcal{S}_3$	0.0363	0.0363
$\mathcal{S}_4$	0.0308	0.0715

higher than those of the DRHL algorithm.

4.2. Unsupervised Classifications of Iris and Wine Data

We further apply the DRHL algorithm to the unsupervised classifications of the Iris and Wine data from UCI Machine Learning Repository [28]. The Iris dataset contains three classes, Iris Versicolor, Iris Virginica and Iris Setosa, and each class consists of 50 samples. Each sample is 4-dimensional vectors measuring the plants morphology. In our experiments on the Iris data, we set the initial value of k as 6 and the initial values of the other parameters as in simulation experiments. Generally, the DRHL algorithm stopped at $k^* = 3$ with the optimal classification accuracy 96.7% (Only five from 150 samples are misclassified). However, it is possible that the DRHL algorithm converges to $k^* = 2$. Since there are two Iris sub-classes which are strongly overlapped, some literatures also accept $k^* = 2$.

The Wine dataset is 13-dimensional and consists of 178 samples of three wines. In this case, we preprocess this dataset by the principal component analysis (PCA) dimension reduction technique [29] and choose only the first three principle components. The DRHL algorithm is conducted on the preprocessed data with the initial value of k as 6. Experimental results demonstrate that the DRHL algorithm always converges with $k^* = 3$ and the accuracy of classification can reach at 98.3% (Only three samples are misclassified).

5. CONCLUSIONS

We have investigated the relationship between the BYY harmony learning and the ML learning and bridged them using a regularization term—the average Shannon entropy of the posterior probability per sample. Based on such a regularization mechanism, we construct the dynamically regularized harmony learning (DRHL) for Gaussian mixtures. By controlling the scale factor of this regularization term to dynamically increase from 0 to 1, the DRHL algorithm starts from the BYY harmony learning with a capability of adaptive model selection, and then gradually transforms to the conventional maximum likelihood learning to obtain a consistent parameter estimation. Moreover, the DRHL algorithm is scalable and can be used on a big dataset with certain data summarization technique. Experimental results demonstrate that, on both synthetic and real-world datasets, the DRHL algorithm can not only select the correct number of actual Gaussians in a dataset, but also obtain the ML estimates of the parameters in the mixture.

Acknowledgments.

This work was supported by the Natural Science Foundation of China for Grants 61171138 and 60771061.

6. REFERENCES

[1] David Peel and G MacLahlan. Finite mixture models, 2000.

[2] Richard A Redner and Homer F Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26(2):195–239, 1984.

[3] Neil E Day. Estimating the components of a mixture of normal distributions. *Biometrika*, 56(3):463–474, 1969.

[4] JA Hartigan. Distribution problems in clustering. *Classification and Clustering*, pages 45–72, 1977.

[5] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[6] Gideon Schwarz et al. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[7] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[8] Chris S. Wallace and David L. Dowe. Minimum message length and kolmogorov complexity. *The Computer Journal*, 42(4):270–283, 1999.

[9] Michael D Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[10] Sylvia Richardson and Peter J Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4):731–792, 1997.

[11] Constantinos Constantinopoulos and Aristidis Likas. Unsupervised learning of gaussian mixtures based on variational component splitting. *IEEE Transactions on Neural Networks*, 18(3):745–755, 2007.

[12] Mario AT Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

[13] Y. M. Cheung. Maximum weighted likelihood via rival penalized em for density mixture clustering with automatic model selection. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):705–761, 2005.

[14] Lei Xu. Ying-yang machine: a bayesian-kullback scheme for unified learnings and new results on vector quantization. In *Proceedings of the 1995 International Conference on Neural Information Processing (ICONIP?5)*, volume 2, pages 977–988, 1995.

[15] Lei Xu. Best harmony, unified rpcl and automated model selection for unsupervised and supervised learning on gaussian mixtures, three-layer nets and me-rbf-svm models. *International Journal of Neural Systems*, 11(01):43–69, 2001.

[16] Lei Xu. Byy harmony learning, structural rpcl, and topological self-organizing on mixture models. *Neural Networks*, 15(8):1125–1151, 2002.

[17] Jinwen Ma, Taijun Wang, and Lei Xu. A gradient byy harmony learning rule on gaussian mixture with automated model selection. *Neurocomputing*, 56:481–487, 2004.

[18] Jinwen Ma and Le Wang. Byy harmony learning on finite mixture: adaptive gradient implementation and a floating rpcl mechanism. *Neural Processing Letters*, 24(1):19–40, 2006.

[19] Jinwen Ma and Xuefeng He. A fast fixed-point byy harmony learning algorithm on gaussian mixture with automated model selection. *Pattern Recognition Letters*, 29(6):701–711, 2008.

[20] Jinwen Ma. Automated model selection (ams) on finite mixtures: a theoretical analysis. In *International Joint Conference on Neural Networks 2006 (IJCNN'06)*, pages 4139–4145. IEEE, 2006.

[21] Lei Xu. Bayesian ying–yang machine, clustering and number of clusters. *Pattern Recognition Letters*, 18(11):1167–1178, 1997.

[22] Zhiwu Lu and Jinwen Ma. A gradient entropy regularized likelihood learning algorithm on gaussian mixture with automatic model selection. In *Advances in Neural Networks-ISNN 2006*, pages 464–469. Springer, 2006.

[23] Zhiwu Lu and Horace Ho-Shing Ip. Generalized competitive learning of gaussian mixture models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(4):901–909, 2009.

[24] Lei Xu, Adam Krzyzak, and Erkki Oja. Rival penalized competitive learning for clustering analysis, rbf net, and curve detection. *Neural Networks, IEEE Transactions on*, 4(4):636–649, 1993.

[25] Jinwen Ma and Taijun Wang. A cost-function approach to rival penalized competitive learning (rpcl). *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(4):722–737, 2006.

[26] Huidong Jin, Man-Leung Wong, and K. S. Leung. Scalable model-based clustering for large databases based on data summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1710–1719, 2005.

[27] Jinwen Ma and Jianfeng Liu. The byy annealing learning algorithm for gaussian mixture with automated model selection. *Pattern Recognition*, 40(7):2029–2037, 2007.

[28] UCI Machine Learning Repository. http://www.ics.uci.edu/∼mlearn/.

[29] Ian Jolliffe. *Principal Component Analysis*. Wiley Online Library, 2005.