Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 29 (2008) 701–711
Pattern Recognition Letters
www.elsevier.com/locate/patrec

A fast fixed-point BYY harmony learning algorithm on Gaussian mixture with automated model selection

Jinwen Ma *, Xuefeng He

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

Received 1 May 2007; received in revised form 16 October 2007
Available online 8 December 2007

Communicated by F. Roli

Abstract

The Bayesian Ying–Yang (BYY) harmony learning theory has brought about a new mechanism that model selection on Gaussian mixture can be made automatically during parameter learning via maximization of a harmony function on finite mixture defined through a specific bidirectional architecture (BI-architecture) of the BYY learning system. In this paper, we propose a fast fixed-point learning algorithm for efficiently implementing maximization of the harmony function on Gaussian mixture with automated model selection. Several simulation experiments are performed to compare its effectiveness in automated model selection as well as its efficiency in parameter learning with other existing learning algorithms. The experimental results reveal that the performance of the proposed algorithm is superior to its counterparts in these aspects. Moreover, the proposed algorithm is further tested with three typical real-world data sets and successfully applied to unsupervised color image segmentation.

Keywords: Bayesian Ying–Yang (BYY) system; Harmony learning; Gaussian mixture; Automated model selection; Fixed-point

1. Introduction

Gaussian mixture model has been widely applied to data modelling and clustering. Typical statistical methods for learning Gaussian mixture include the EM algorithm (Render and Walker, 1984) and the k-means algorithm (Jain and Dubes, 1988). These approaches share a common limitation in that they have to assume that the number of Gaussians in the mixture is known a priori. However, this assumption is practically unrealistic for many unsupervised learning tasks such as clustering or competitive learning. The critical issue of model selection in terms of the appropriate number of Gaussians must be properly addressed in addition to parameter learning (Hartigan, 1977).

In the literature, two major stream of approaches have been adopted for model selection. The first stream relies on cost-function based selection criteria (Friedman and Rubin, 1967; Scott and Symons, 1971; Akaike, 1974; Milligan and Copper, 1985) to determine best number k^* of Gaussians. However, this stream of approach is not computationally efficient as the process of sequentially evaluating the criterion incurs large computational cost. Moreover, all the existing theoretic selection criteria have their limitations and often result in a wrong result. Another stream, commonly referred to as automated model selection, implements model selection in parallel with parameter learning. The earliest effort of this kind might be due to the rival penalized competitive learning (RPCL) algorithm (Xu et al., 1993) which can automatically determine the appropriate number of clusters for a sample data set by driving extra weight vectors far away from the sample data. Although the RPCL algorithm has been generalized to several versions based on distribution of data or newly found cost function (Xu et al., 1998; Ma and Wang, 2006; Ma and Cao, 2006) with certain mathematical analysis on its

* Corresponding author. Tel.: +86 10 62760609; fax: +86 10 62751801.
E-mail address: jwma@math.pku.edu.cn (J. Ma).

0167-8655/$ - see front matter © 2007 Published by Elsevier B.V.
doi:10.1016/j.patrec.2007.11.014

convergence, it can only provide a rough parameter estimation for Gaussian mixture modelling because there are no mixing proportions of clusters or Gaussians in the algorithm.

Recently, some other gradient based learning algorithms (Ma et al., 2004, 2005; Ma and Wang, 2006) with automated model selection ability have been proposed from the perspective of Bayesian Ying–Yang (BYY) harmony learning principle. The BYY harmony learning principle was proposed in (Xu, 1995) and systematically developed in (Xu, 2001, 2002a,b). It acts as a general statistical learning framework not only for understanding several existing major learning approaches but also for tackling the learning problem with a new learning mechanism that makes model selection automatically during parameter learning. Actually, it has been shown in (Ma et al., 2004) that Gaussian mixture modelling is equivalent to the maximization of the harmony function on a specific BI-architecture of the BYY learning system (related to the Gaussian mixture model) via a gradient learning rule. To improve this BYY gradient learning, the conjugate and natural gradient learning rules (Ma et al., 2005) were further proposed. Furthermore, an adaptive gradient learning algorithm was already proposed and analyzed for the general finite mixture model (Ma and Wang, 2006). On the other hand, an annealing learning algorithm with automated model selection ability was also established on a back architecture (B-architecture) of the BYY learning system related to Gaussian mixture (Ma and Liu, 2007).

Moreover, from the point of view of penalizing the Shannon entropy of the mixing proportions on maximum likelihood estimation (MLE), an entropy penalized MLE iterative algorithm was proposed to make model selection automatically during parameter learning on Gaussian mixture in a similar way (Ma and Wang, 2004). On the other hand, according to a special merge-or-split criterion, a dynamic merge-or-split learning algorithm with automated model selection ability was also proposed for Gaussian mixture modelling in which the initial number of Gaussians can be given arbitrarily (Ma and He, 2005).

In the current paper, we propose a fast fixed-point learning algorithm on Gaussian mixture for efficiently implementing the maximization of the harmony function on the BI-architecture of the BYY learning system related to the Gaussian mixture model. It is derived from the harmony function using Lagrangian multipliers. Simulation experiments show that the fixed-point learning algorithm not only has automated model selection ability in learning, but also is more computationally efficient than the gradient based algorithms. Moreover, it is further tested with three typical real-world data sets and successfully applied to unsupervised color image segmentation.

The rest of this paper is organized as follows. Section 2 outlines the BYY harmony learning principle and derives the fixed-point learning algorithm. Section 3 is devoted to simulation, test and application experimental results and Section 4 concludes the paper.

2. Fixed-point learning algorithm

2.1. BYY learning system and harmony function

A BYY learning system describes each observation $x \in \mathscr{X} \subset R^n$ and its corresponding inner representation $y \in \mathscr{Y} \subset R^m$ via the two types of Bayesian decomposition of the joint density $p(x,y) = p(x)p(y|x)$ and $q(x,y) = q(x|y)q(y)$, called Yang machine and Ying machine, respectively. For purpose of Gaussian mixture modelling, y is limited to an integer variable, i.e., $y \in \mathscr{Y} = \{1,2,\ldots,k\} \subset R$ with $m = 1$. Given a data set $D_x = \{x_t\}_{t=1}^N$, the task of learning on a BYY learning system consists of specifying all the aspects of $p(y|x)$, $p(x)$, $q(x|y)$, $q(y)$ with a harmony learning principle implemented by maximizing the function

$$H(p\|q) = \int p(y|x)p(x)ln[q(x|y)q(y)]\mathrm{d}x\,\mathrm{d}y - \ln z_q, \tag{1}$$

where z_q is a regularization term. Refer to (Xu, 2001) for details.

If both $p(y|x)$ and $q(x|y)$ are parametric, i.e., from a family of probability densities with a parameter θ, the BYY learning system is called to have a bi-directional architecture (BI-architecture). For Gaussian mixture modelling, we use the following specific BI-architecture of the BYY learning system. $q(j) = \alpha_j$ with $\alpha_j \geqslant 0$ and $\sum_{j=1}^k \alpha_j = 1$. Also, we ignore the regularization term z_q (i.e., set $z_q = 1$) and let $p(x)$ be the empirical density $p_0(x) = \frac{1}{N}\sum_{t=1}^N \delta(x - x_t)$, where $x \in \mathscr{X} = R^n$. Moreover, the BI-architecture is constructed with the following parametric form:

$$p(y = j|x) = \frac{\alpha_j q(x|\theta_j)}{q(x|\Theta_k)}, \quad q(x|\Theta_k) = \sum_{j=1}^k \alpha_j q(x|\theta_j), \tag{2}$$

where $q(x|\theta_j) = q(x|y = j)$ with θ_j consisting of all its parameters and $\Theta_k = \{\alpha_j, \theta_j\}_{j=1}^k$. Substituting these component densities into Eq. (1), we have

$$H(p\|q) = J(\Theta_k) = \frac{1}{N}\sum_{t=1}^N \sum_{j=1}^k \frac{\alpha_j q(x_t|\theta_j)}{\sum_{i=1}^k \alpha_i q(x_t|\theta_i)} \ln[\alpha_j q(x_t|\theta_j)]. \tag{3}$$

That is, $H(p\|q)$ becomes a harmony function $J(\Theta_k)$ on the parameters Θ_k. Let $q(x|\theta_j)$ be a Gaussian probability density function given by

$$q(x|\theta_j) = q(x|m_j, \Sigma_j) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-m_j)^{\mathrm{T}}\Sigma_j^{-1}(x-m_j)}, \tag{4}$$

where m_j is the mean vector and Σ_j is the covariance matrix which is assumed to be positive definite. As a result, this BI-architecture of the BYY learning system contains the Gaussian mixture model $q(x,\Theta_k) = \sum_{j=1}^k \alpha_j q(x|m_j,\Sigma_j)$ which tries to model the original distribution of the sample data in D_x.

According to the BYY harmony learning principle (Xu, 2001; Ma et al., 2005), maximization of $J(\Theta_k)$ would auto-

matically lead to model selection on Gaussian mixture. In order to efficiently implement the maximization of $J(\Theta_k)$, we will derive a fixed-point learning algorithm to solve the maximum of $J(\Theta_k)$ in the next subsection.

2.2. Derivation of fixed-point learning algorithm

Since $\sum_{j=1}^{k}\alpha_j = 1$, we introduce the Lagrange multiplier λ and the Lagrange function

$$L(\Theta_k, \lambda) = J(\Theta_k) + \lambda\left(1 - \sum_{j=1}^{k}\alpha_j\right). \tag{5}$$

By matrix differentiation, we have the following set of equations:

$$\frac{\partial L}{\partial \alpha_j} = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{\alpha_j}h_j(t) - \lambda, \tag{6}$$

$$\frac{\partial L}{\partial m_j} = \frac{1}{N}\sum_{t=1}^{N}h_j(t)\Sigma_j^{-1}(x_t - m_j), \tag{7}$$

$$\frac{\partial L}{\partial \Sigma_j} = \frac{1}{2N}\sum_{t=1}^{N}h_j(t)\left[\Sigma_j^{-1}(x_t - m_j)(x_t - m_j)^{\mathrm{T}}\Sigma_j^{-1} - \Sigma_j^{-1}\right], \tag{8}$$

$$\frac{\partial L}{\partial \lambda} = \sum_{i=1}^{k}\alpha_i - 1, \tag{9}$$

where $h_j(t) = p_j(t) + \sum_{i=1}^{k}p_j(t)(\delta_{ij} - p_i(t))\ln[\alpha_i q(x_t|m_i, \Sigma_i)]$, $j = 1, \ldots, k$, δ_{ij} is the Kronecker function and $p_j(t) = p(j|x_t)$. By setting Eqs. (6)–(9)=0 and solving them, we have

$$\lambda = \frac{1}{N}\sum_{i=1}^{k}\sum_{t=1}^{N}h_i(t) \tag{10}$$

and further obtain the following fixed-point (iterative) learning algorithm:

$$\hat{\alpha}_j = \frac{\sum_{t=1}^{N}h_j(t)}{\sum_{i=1}^{k}\sum_{t=1}^{N}h_i(t)}, \tag{11}$$

$$\hat{m}_j = \frac{1}{\sum_{t=1}^{N}h_j(t)}\sum_{t=1}^{N}h_j(t)x_t, \tag{12}$$

$$\hat{\Sigma}_j = \frac{1}{\sum_{t=1}^{N}h_j(t)}\sum_{t=1}^{N}h_j(t)(x_t - \hat{m}_j)(x_t - \hat{m}_j)^{\mathrm{T}}. \tag{13}$$

Clearly, this fixed-point learning algorithm is similar to the EM algorithm for Gaussian mixture. However, it differs from the EM algorithm at $h_j(t)$ which introduces certain rewarding and penalizing mechanism on the mixing proportions so that it has the feature of automated model selection that will be demonstrated in the sequel.

3. Experimental results

3.1. Simulation results and comparisons

In this subsection, various simulation experiments are carried out to demonstrate the performance of the fixed-point learning algorithm for both model selection and parameter estimation on a sample data set from a Gaussian mixture, with being compared with those of the gradient based learning algorithms.

3.1.1. Sample data sets

We conduct 7 Monte Carlo experiments to sample data drawn from a mixture of three or four bivariate Gaussian distributions (i.e., $n = 2$). As shown in Fig. 1, each data set is generated with different degree of overlap among the clusters and with equal or unequal mixing proportions. Below is a detailed description.

(i) The clusters in $\mathscr{S}_1$ and $\mathscr{S}_2$ have equal number of samples, while those in the other five data sets have different numbers of samples;
(ii) The clusters in $\mathscr{S}_1$, $\mathscr{S}_3$ and $\mathscr{S}_6$ are separated, but those in each of the other four data sets are overlapped at certain degree;
(iii) The clusters in $\mathscr{S}_1$ and $\mathscr{S}_2$ are spherical in shape, but those in the other five data sets are elliptic in shape. In particular, the clusters in $\mathscr{S}_5$ and $\mathscr{S}_6$ are rather flat;
(iv) The sample size of the first five data sets is larger as compared with that of $\mathscr{S}_6$ and $\mathscr{S}_7$.

The values of the parameters of the seven data sets are summarized in Table 1 where m_i, $\Sigma_i = [\sigma_{jk}^i]_{2\times 2}$, α_i and N_i denote the mean vector, covariance matrix, mixing proportion and the number of samples of the ith Gaussian cluster respectively.

3.1.2. Automated model selection

We implement the fixed-point learning algorithm on these seven data sets with $k \geqslant k^*$. The parameters of the fixed-point learning algorithm are initialized randomly in some intervals with the constraints. However, it is found by the experiments that if the initial mean vectors of k Gaussians are trained by the rival penalized competitive learning (RPCL) algorithm (Xu et al., 1993) on the sample data with a small number of iterations, the fixed-point learning algorithm converges more quickly. Thus, we always select the initial mean vectors of k Gaussians in the mixture with the aid of a short RPCL process. Learning is stopped once the terminating criterion $|J(\Theta_k^{\mathrm{new}}) - J(\Theta_k^{\mathrm{old}})| < 10^{-7}$ is satisfied. Actually, we find that the fixed-point learning algorithm always converge in all attempts.

The experimental results on $\mathscr{S}_2$ and $\mathscr{S}_4$ are given in Figs. 2 and 3 respectively, with case $k = 8$ and $k^* = 4$. It can be observed that four-clustered data are correctly described by the four Gaussians with the mixing proportions of the other four redundant Gaussians falling to below the threshold value of 0.01 and being consequently discarded. This shows that automated model selection works well to select the correct number of the Gaussians on the given data sets. Similar experimental results are obtained for $\mathscr{S}_5$ with $k = 8$, $k^* = 3$. As shown in Fig. 4,

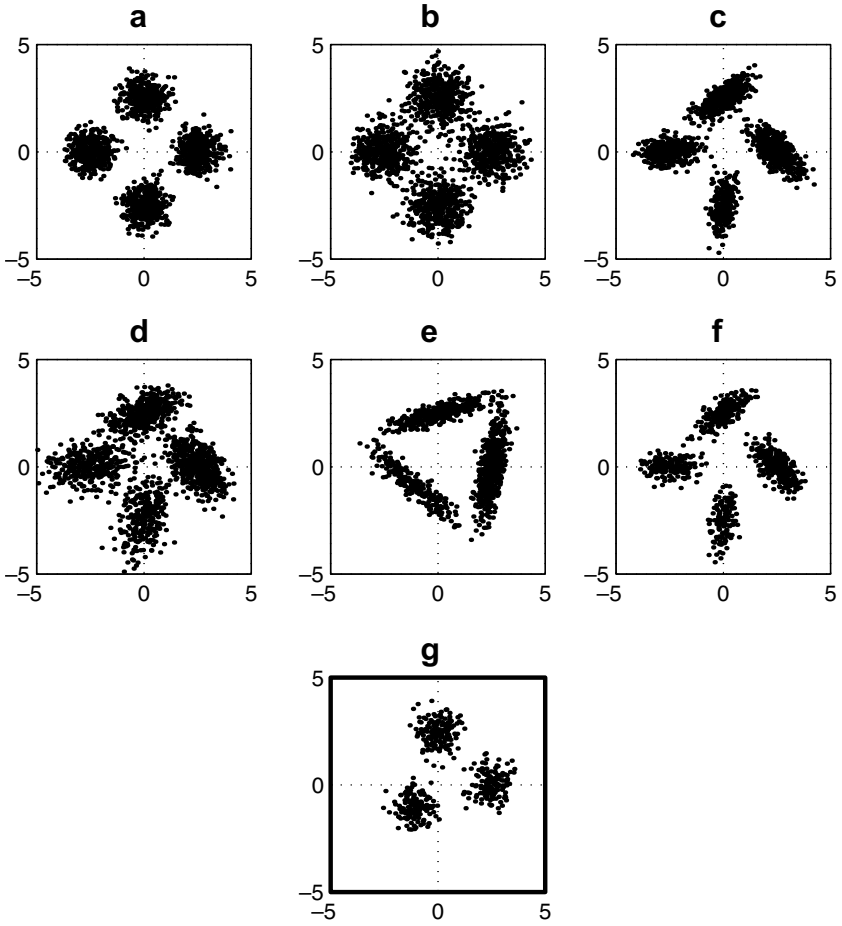

Fig. 1. Seven sets of sample data used in the experiments: (a) Set $\mathscr{S}_1$; (b) Set $\mathscr{S}_2$; (c) Set $\mathscr{S}_3$; (d) Set $\mathscr{S}_4$; (e) Set $\mathscr{S}_5$; (f) Set $\mathscr{S}_6$; (g) Set $\mathscr{S}_7$.

the clusters that constitute the sample data are far from spherical in shape. As shown in Fig. 5, for $\mathscr{S}_6$ with $k = 8$, $k^* = 4$, model selection still works even though the sample size for each cluster is very small.

In fact, automated model selection rarely fails to work when we initialize $k^* \leqslant k \leqslant 3k^*$. However, when $k > 3k^*$, the number of local maxima would increase with k and adversely affect the chance of success.

3.1.3. Parameter estimation

We further compare the converged values of parameters with its actual values. The performance metric of average error is adopted which is the absolute deviation between the estimated parameters and the true parameters in each case. The average errors of the fixed-point learning (FPL) algorithms, the batch gradient learning (BGL) (Ma et al., 2004), the adaptive gradient learning (AGL) (Ma and Wang, 2006) and the EM algorithms are listed in Table 2.[1] We find that the performance of the fixed-point learning algorithm is as good as those of the batch and adaptive gradient algorithms, but is slightly outperformed by the EM algorithm for Gaussian mixture.

3.1.4. Advantages over the gradient learning

From the simulation experiments, we find that the fixed-point learning algorithm has at least three advantages over gradient based algorithms.

(i) There is no additional parameter besides those in Θ_k, which makes it convenient to be implemented. In fact, gradient learning algorithms face a common yet difficult task of having to select an appropriate learning rate which usually varies with each data set.
(ii) The fixed-point learning algorithm is not so sensitive to the initialization of parameters while this is critical for gradient based learning algorithms as improper initialization increases the chance to be trapped in local maxima.
(iii) The fixed-point learning algorithm is more computationally efficient than the two gradient based algorithms. Empirically, as shown in Table 3 we find that the fixed-point learning (FPL) algorithm needs only about one tenth to a quarter of the number of

[1] Since the performances of the natural and conjugate gradient learning algorithms proposed in (Ma et al., 2005) are generally between those of the BGL and AGL algorithms, we neglect the comparison of the FPL algorithm with them.

Table 1
Values of parameters of the seven data sets

The sample set	Gaussian	m_i	σ^i_{11}	σ^i_{12}	σ^i_{22}	α_i	N_i
$\mathscr{S}_1$ ($N = 1600$)	Gaussian 1	(2.5, 0)	0.25	0	0.25	0.25	400
	Gaussian 2	(0, 2.5)	0.25	0	0.25	0.25	400
	Gaussian 3	(−2.5, 0)	0.25	0	0.25	0.25	400
	Gaussian 4	(0, −2.5)	0.25	0	0.25	0.25	400
$\mathscr{S}_2$ ($N = 1600$)	Gaussian 1	(2.5, 0)	0.5	0	0.5	0.25	400
	Gaussian 2	(0, 2.5)	0.5	0	0.5	0.25	400
	Gaussian 3	(−2.5, 0)	0.5	0	0.5	0.25	400
	Gaussian 4	(0, −2.5)	0.5	0	0.5	0.25	400
$\mathscr{S}_3$ ($N = 1600$)	Gaussian 1	(2.5, 0)	0.28	−0.20	0.32	0.34	544
	Gaussian 2	(0, 2.5)	0.34	0.20	0.22	0.28	448
	Gaussian 3	(−2.5, 0)	0.50	0.04	0.12	0.22	352
	Gaussian 4	(0, −2.5)	0.10	0.05	0.50	0.16	256
$\mathscr{S}_4$ ($N = 1600$)	Gaussian 1	(2.5, 0)	0.45	−0.25	0.55	0.34	544
	Gaussian 2	(0, 2.5)	0.65	0.20	0.25	0.28	448
	Gaussian 3	(−2.5, 0)	1.0	0.1	0.35	0.22	352
	Gaussian 4	(0, −2.5)	0.30	0.15	0.80	0.16	256
$\mathscr{S}_5$ ($N = 1200$)	Gaussian 1	(2.5, 0)	0.1	0.2	1.25	0.5	600
	Gaussian 2	(0, 2.5)	1.25	0.35	0.15	0.3	360
	Gaussian 3	(−1, −1)	1.0	−0.8	0.75	0.2	240
$\mathscr{S}_6$ ($N = 800$)	Gaussian 1	(2.5, 0)	0.28	−0.20	0.32	0.34	272
	Gaussian 2	(0, 2.5)	0.34	0.20	0.22	0.28	224
	Gaussian 3	(−2.5, 0)	0.50	0.04	0.12	0.22	176
	Gaussian 4	(0, −2.5)	0.10	0.05	0.50	0.16	128
$\mathscr{S}_7$ ($N = 450$)	Gaussian 1	(2.5, 0)	0.25	0	0.25	0.3333	150
	Gaussian 2	(0, 2.5)	0.25	0	0.25	0.3333	150
	Gaussian 3	(−1, −1)	0.25	0	0.25	0.3333	150

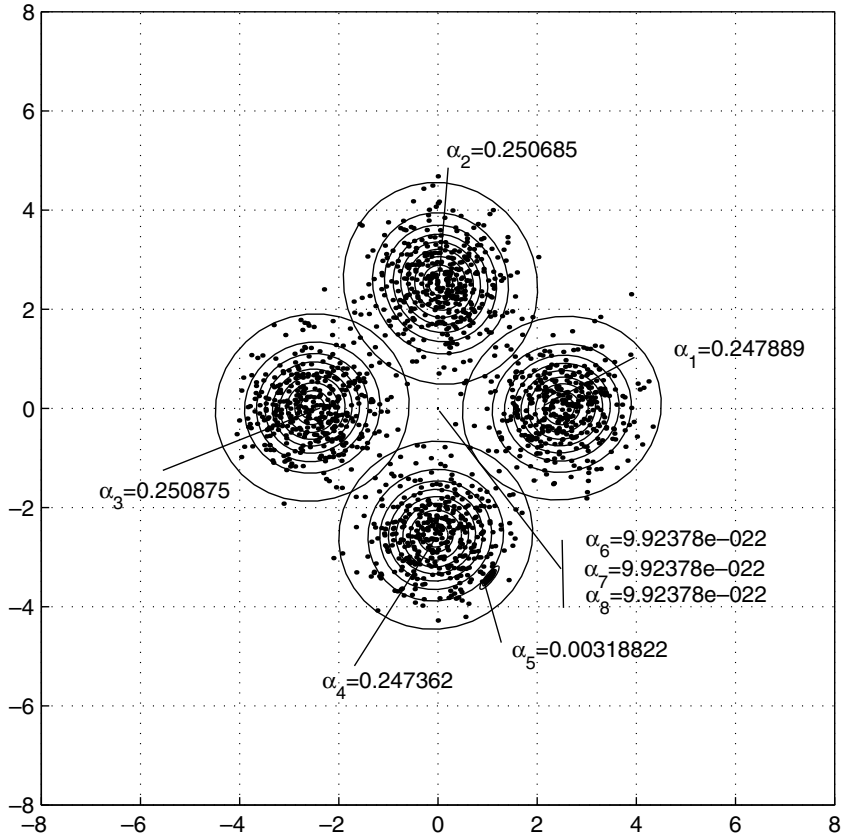

Fig. 2. The experimental result of the fixed-point learning algorithm on the sample set S_2 (stopped after 62 iterations). In this and the following three figures, the contour lines of each Gaussian are retained unless its density is less than e^{-3} (peak).

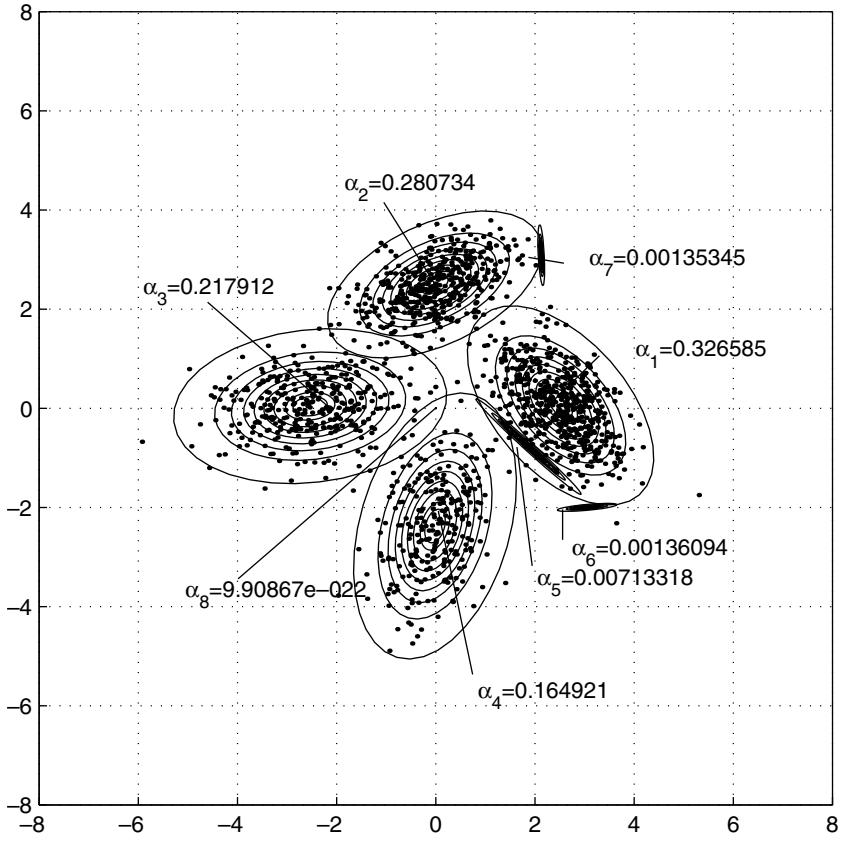

Fig. 3. The experimental result of the fixed-point learning algorithm on $\mathscr{S}_4$ (stopped after 50 iterations).

iterations required by either the batch gradient learning (BGL) or adaptive gradient learning (AGL) algorithm (We neglect the comparisons of the conjugate and natural gradient learning algorithms since they introduce much additional computation in each iteration).

3.1.5. Discussion on the convergence behavior

According to the equations Eqs. (11)–(13), the fixed-point learning algorithm cannot constrain the $\hat{\alpha}_j$ to always stay nonnegative and $\widehat{\Sigma}_j$ positive definite for each iteration. Empirically, we do observe that some $h_j(t)$ becomes negative at certain iteration. However, as the Gaussians in the actual mixture are quite apart, this implies that negative $h_j(t)$ occurs at only a small number of samples. As a result, $\hat{\alpha}_j$ and $\widehat{\Sigma}_j$ can remain nonnegative and positive definite throughout the learning process. Therefore, the fixed-point learning algorithm still works well with certain degree of overlap among the clusters or Gaussians in the sample data.

In a summary, the fixed-point learning algorithm can effectively make model selection automatically during parameter learning. Moreover, it converges much faster than the gradient based BYY learning algorithms, and as accurately as the EM algorithm. As compared with the BYY annealing learning algorithm (Ma and Liu, 2007) and the dynamic merge-or-split learning algorithm (Ma and He, 2005), it still converges much faster. In comparison with the entropy penalized MLE iterative algorithm (Ma and Wang, 2004), it generally converges in the same speed, but leads to a better result.

3.2. Unsupervised classification on real-world data sets

The fixed-point learning algorithm is further tested with three typical real-world data sets: the Iris data, the Wine data and the Waveform data, out of the UCI Machine Learning Repository (Blake and Merz, 1998). Actually, they are well-known benchmark datasets for testing a new proposed clustering learning algorithm. Specifically, the dimensions of the Wine and Waveform data are rather high, while the dimension of the Waveform data is relatively low. On the other hand, the numbers of samples in the Iris and Wine datasets are moderate, while the number of samples in the Waveform dataset is relatively large. Obviously, the fixed-point learning algorithm can only classify these real-world data in each dataset in an unsupervised mode. In the following, the testing results of the fixed-point learning algorithm on the three real-world datasets are described respectively.

3.2.1. On the Iris data

The Iris data set consists of 150 samples of three classes: Iris Versicolor, Iris Virginica and Iris Setosa, with each

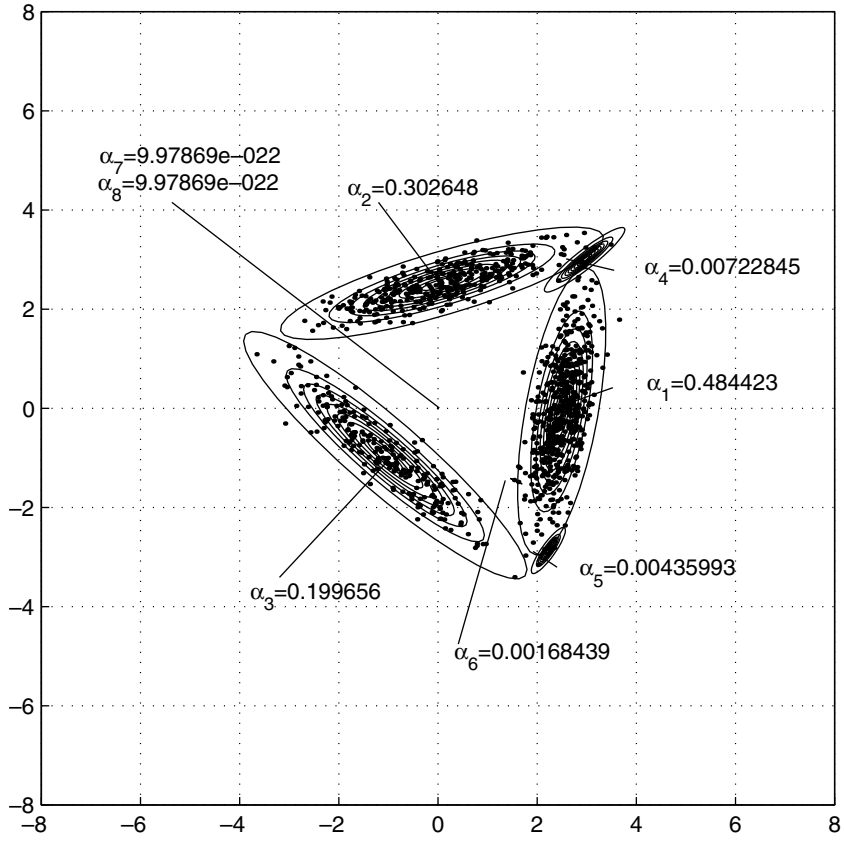

Fig. 4. The experimental result of the fixed-point learning algorithm on $\mathscr{S}_5$ (stopped after 90 iterations).

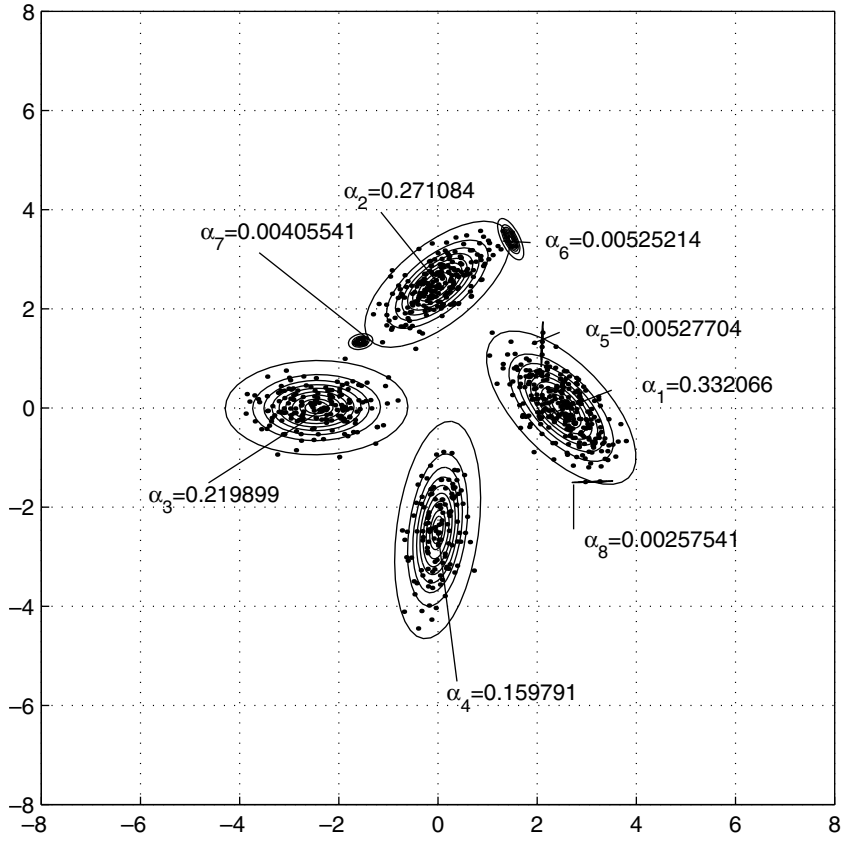

Fig. 5. The experimental result of the fixed-point learning algorithm on $\mathscr{S}_6$ (stopped after 50 iterations).

Table 2
The average errors of the estimated parameters by the four algorithms on the seven data sets

$\mathscr{S}$	$\mathscr{S}_1$	$\mathscr{S}_2$	$\mathscr{S}_3$	$\mathscr{S}_4$	$\mathscr{S}_5$	$\mathscr{S}_6$	$\mathscr{S}_7$
FPL	0.014631	0.023206	0.0201193	0.022816	0.049643	0.009069	0.029247
BGL	0.012281	0.026629	0.0153004	0.024001	0.024168	0.008155	0.026920
AGL	0.015721	0.027539	0.019887	0.025610	0.034658	0.016092	0.028971
EM	0.011799	0.019434	0.015840	0.015928	0.029762	0.008623	0.024167

Table 3
The numbers of iterations required by the three algorithms on the seven sets of sample data

$\mathscr{S}$	$\mathscr{S}_1$	$\mathscr{S}_2$	$\mathscr{S}_3$	$\mathscr{S}_4$	$\mathscr{S}_5$	$\mathscr{S}_6$	$\mathscr{S}_7$
IL	67	69	119	90	246	180	178
BGL	684	308	1021	906	1058	1001	40001
AGL	355	269	817	1105	823	626	1205

For convenience of comparison, the number of iterations of the adaptive gradient learning algorithm is also computed in the batch way as the quotient of the number of adaptive iterations over the number of sample data in the data set.

class containing 50 samples. Each sample or datum is four-dimensional, which represents measures of the plants morphology. Here, the category or class index of each sample in the Iris data set is already known. As the fixed-point learning algorithm is a kind of unsupervised learning algorithm, we can consider that all of 150 samples blindly come from a mixture of three Gaussians, which represent the three Iris classes separately. In such a way, we can implement the fixed-point learning algorithm on these 150 samples to detect the three representative Gaussians and classify them according to the Bayesian classification rule based on posteriori probabilities $p(j|x_t)$ of the final estimated Gaussians. To evaluate the classification performance of the algorithm, we compute the classification accuracy given the real categories of the 150 samples.

By setting $k = 6$ with the initial parameters being selected in the same way as above, we implement the fixed-point learning algorithm on the Iris data. It is found by the experiments that the fixed-point learning algorithm can detect the three classes in the Iris data with an optimal classification accuracy of 96.7% (there are only five errors in the second class) which is slightly less than the best known classification accuracy 98% (there are three errors) of the maximum certainty partitioning with a large number of linear mixing kernels (Gaussian functions) (Roberts et al., 2000).

3.2.2. On the wine data

The Wine data are typical high-dimensional real-world data for testing a classification or clustering algorithm. Actually, the Wine data contain 178 samples of three types of wine. Each datum is 13-dimensional and consists of chemical analysis of a sample from certain type of wine. In the same way, we can use a mixture of three Gaussians to describe the Wine data and implement the fixed-point learning algorithm to estimate the three representative Gaussians for the three types of wine. Also, we just use the class indexes of these wine samples provided in the dataset to check the classification accuracy of the fixed-point learning algorithm on the wine data.

In the experiments, we first regularize these Wine data into a proper interval of $[0, 3]$ for convenience of data processing and set $k = 6$ which is also the two times of the real class number 3. We then run the fixed-point learning algorithm on these regularized Wine data. It is found by the experiments that the fixed-point learning algorithm can detect the three classes in the Wine data with an optimal classification accuracy of 98.32% (there are only three errors in the second class) which is even better than the classification accuracy 97.75% (there are four errors) of the maximum certainty partitioning method (Roberts et al., 2000). However, the Wine data can now be classified completely, i.e., with the classification accuracy 100%, via the adaptive Mahalanobis distance based rival penalized competitive learning (MDRPCL) algorithm with the annealing simulated mechanism proposed recently in (Ma and Cao, 2006).

3.2.3. On the waveform data

The waveform data set is a very etherogeneous real-world dataset for testing a classification or clustering algorithm. Actually, it contains 5000 samples of three types of waveform and each datum is even 21-dimensional and consists of various measures on the waveform. In the same way, we can implement the fixed-point learning algorithm on the Waveform data to detect the three representative Gaussians for the three types of waveform. Again, we just use the class indexes of these waveform samples provided in the dataset to check the classification accuracy of the fixed-point learning algorithm on the waveform data.

In the experiments, for convenience of data processing, we first regularize these waveform data into a proper interval of $[0, 4]$ in the same way as in the case of the Wine data, and then use the Principal Component Analysis (PCA) technique (Duda et al., 2001) to reduce the dimension of the waveform data to 18. Again, k is set to 6, i.e., the two times of the real class number 3. With those preparations and setting, we run the fixed-point learning on the processed Waveform data. It is found by the experiments that the fixed-point learning algorithm can detect the three classes in the Waveform data with an optimal classification accuracy of 82.78% which is slightly less than the classification accuracy of 83.96% of the EM algorithm on the Waveform data with $k = 3$. Although the classification accuracy

of the regularized mixture discriminant analysis method (Halbe and Aladjem, 2007) on the Waveform data may be improved to 86.78%, this method implements three regularized mixtures of Gaussians to learn and represent the three types of waveform separately.

In a summary, the experiment results reveal that the fixed-point learning algorithm can be successfully implemented on some typical real-world datasets for unsupervised classification and its classification accuracy can be as good as those of the other existing methods. However, through the further experiments on the other real-world datasets, we can find that the fixed-point learning algorithm works well only for a set of real-world data that can be modeled accurately or at least approximately by a Gaussian mixture in a certain degree of overlap among the components. Otherwise, it cannot obtain a satisfactory classification accuracy. For example, as the fixed-point learning algorithm is implemented on the Vowel data (Blake and Merz, 1998) that have 11 classes, we can find only 9 classes. Actually, the two pairs of actual classes are heavily overlapped. However, if we only consider the Vowel data on the other separated classes, the fixed-point learning algorithm can obtain a very high classification accuracy. On the other hand, in some cases like the Banana data (Blake and Merz, 1998), actual classes in the real-world data cannot be described by Gaussian distributions and thus the fixed-point learning algorithm cannot be applied efficiently. However, just like the regularized mixture discriminant analysis method, we can use a number of Gaussian mixtures to represent the actual classes, respectively, and implement the fixed-point learning algorithm to learn these Gaussian mixtures separately. In such a way, we can still build a good Bayesian classifier in a hybrid model of both supervised and unsupervised classifications.

3.3. Unsupervised color image segmentation

We finally apply the fixed-point learning algorithm to unsupervised color image segmentation that has been considered as a promising and challenging area in image processing (Boujemaa, 2000). Segmenting a digital color image into homogenous regions corresponding to the objects (including the background) is a fundamental problem in image processing. When the number of objects in an image is not known in advance, the image segmentation problem is in an unsupervised mode and becomes rather difficult in practice. If we consider each object as a Gaussian distribution, the whole color image can be regarded as a Gaussian mixture in the data or color space. Then, the fixed-point learning algorithm provides a new tool for solving this unsupervised color image segmentation problem. In this situation, we set k to be larger than the true number k^* of the actual objects and the pixels in the image are partitioned according to the maximum posteriori probability among $p(j|x_t)$. The extra Gaussians or objects will be discarded as their mixing proportions are less than a threshold value 0.01. In our experiments, we first regularize all the three coordinates of the pixels in each RGB based color image via dividing them by 32 so that the regularized coordinates are within an appropriate interval of $[0, 8]$. The initial parameters for the fixed-point learning algorithm are selected similarly as above and the algorithm ends with a simplified stopping criterion: $\sum_{j=1}^{k} \|m_j^{\text{new}} - m_j^{\text{old}}\| < 10^{-5}$.

The first experiment is made on the color image of two goats which is shown in Fig. 6a. We implement the fixed-point learning algorithm on this color image with $k = 6$ and lead to the segmentation results shown in Fig. 6b. It can be found that two objects are finally located accurately, while the mixing proportions of the other four Gaussians (objects) are reduced to below 0.01, i.e., these objects are extra and discarded from the figure. That is, the correct number of the actual objects have been detected on the color image with an accurate segmentation. Moreover, the second experiment has been made on the color image of house, which is shown in Fig. 7a, with $k = 6$. As shown in Fig. 7b, two objects are located accurately, while the mixing proportions of the other four extra Gaussians (objects) become less than 0.01. That is, the correct number of the objects can still be detected on this color image. Finally, the fixed-point learning algorithm is implemented on the color image of jellies, which is shown in Fig. 8a, with

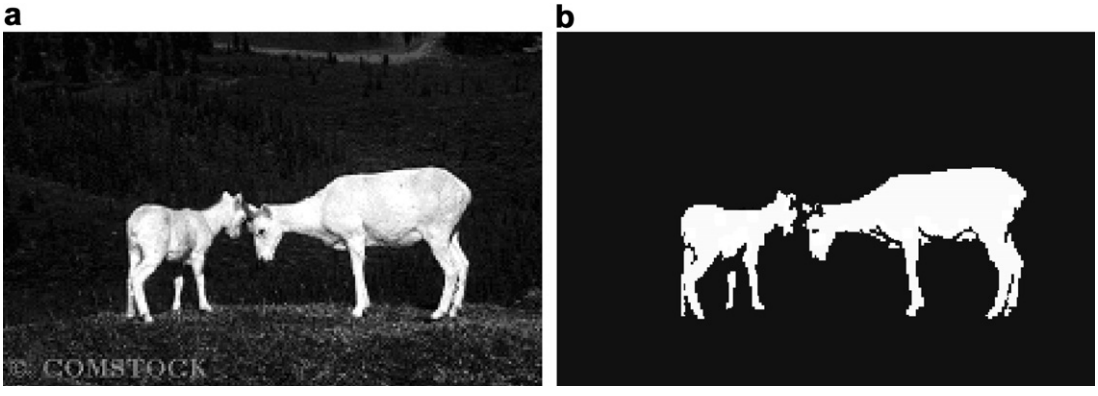

Fig. 6. The segmentation result on the color image of two goats. (a) The original color image of two goats; (b) the segmented image of two goats via the fixed-point learning algorithm.

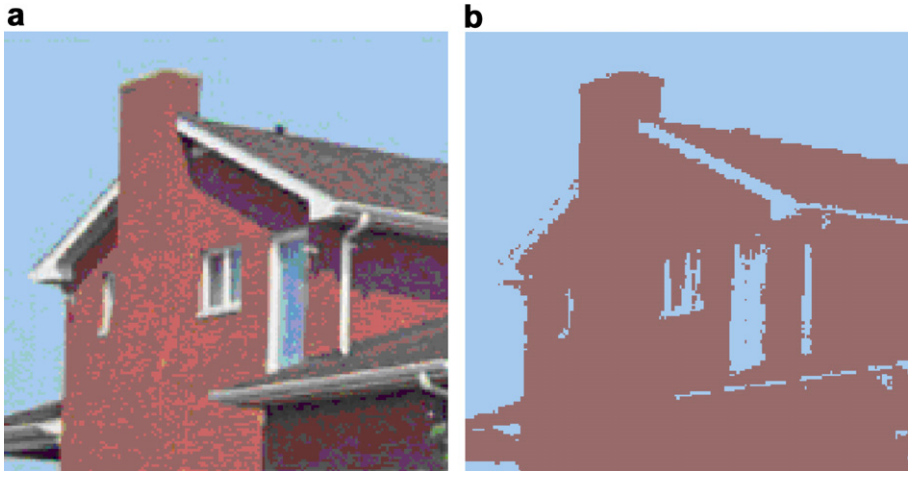

Fig. 7. The segmentation result on the color image of house. (a) The original color image of house; (b) the segmented image of house via the fixed-point learning algorithm.

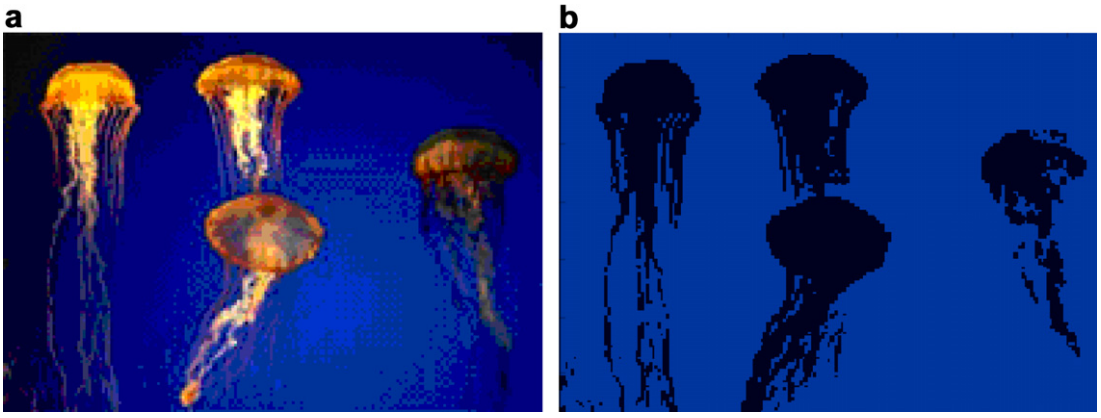

Fig. 8. The segmentation result on the color image of jellies. (a) The original color image of jellies; (b) the segmented image of jellies via the fixed-point learning algorithm.

$k = 8$. As shown in Fig. 8b, the two objects are located accurately, with the mixing proportions of other four extra Gaussians reduced below 0.01.

As a result, the fixed-point learning algorithm can detect the number of actual objects in each of these color images. Moreover, the segmentation results of the fixed-point learning algorithm are better than those of the generalized competitive clustering (GCC) algorithm (Boujemaa, 2000) (based on the fuzzy clustering theory) given in the web http://www-rocq.inria.fr/boujemaa/Partielle2.html. By comparison, we can easily find that the fixed-point learning algorithm leads to a more accurate segmentation on the contours of the objects in each image generally.

4. Conclusions

In this paper, we have constructed a fast fixed-point learning algorithm for the BYY harmony learning on Gaussian mixture with automated model selection. It is derived from a specific bi-architecture of the BYY learning system using Lagrangian optimization method. Simulation results confirm that the fixed-point learning algorithm has automated model selection ability for the number of Gaussians and produce good estimates for the actual parameters. It is also shown that the fixed-point learning algorithm is superior to the gradient based learning algorithms as well as the other automated model selection algorithms. Moreover, the fixed-point learning algorithm is tested with three typical real-world dataset and successfully applied to unsupervised color image segmentation.

Acknowledgements

This work was supported by the Natural Science Foundation of China (NSFC) for Project 60471054. The authors thank Prof. Lei Xu for his strong support and helpful discussions, and Prof. Taijun Wang, Mr. Lei Li, and Miss Hongyan Wang for their simulation and experiment supports. A preliminary version of this work or the algorithm was presented at the International Conference on Neural Networks and Signal Processing (ICNN&SP03), Dec. 14–

17, 2003, Nanjing, China, and published in its proceedings, vol. 1, pp. 7–10.

References

Akaike, H., 1974. A new look at the statistical model identification. IEEE Trans. Automat. Control AC-19, 716–723.

Blake, C.L., Merz, C.J., 1998. UCI Repository of Machine Learning Databases. University of California, Department of Information and Computer Science, Irvine. <http://www.ics.uci.edu/mlearn/MLRepository.html>.

Boujemaa, N., 2000. Generalized competitive clustering for image segmentation. In: Proc. 19th Internat. Conf. of the North American Fuzzy Information Processing Society, pp. 133–137.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern Classification. Wiley.

Friedman, H.P., Rubin, J., 1967. On some invariant criteria for grouping data. J. Amer. Statist. Assoc. 62, 1159–1178.

Halbe, Z., Aladjem, M., 2007. Regularized mixture discriminant analysis. Pattern Recognition Lett. 28 (15), 2104–2115.

Hartigan, J.A., 1977. Distribution problems in clustering. In: Van Ryzin, J. (Ed.), Classification and Clustering. Academic Press, New York, pp. 45–72.

Jain, A.K., Dubes, R.C., 1988. Algorithm for Clustering Data. Prentice Hall, Englewood Cliffs, NJ.

Ma, J., Cao, B., 2006. The Mahalanobis distance based rival penalized competitive learning algorithm. Lecture Notes Comput. Sci. 3971, 442–447.

Ma, J., He, Q., 2005. A dynamic merge-or-split learning algorithm on Gaussian mixture for automated model selection. Lecture Notes Comput. Sci. 3578, 203–210.

Ma, J., Liu, J., 2007. The BYY annealing learning algorithm for Gaussian mixture with automated model selection. Pattern Recognition 40 (7), 2029–2037.

Ma, J., Wang, T., 2004. Entropy penalized automated model selection on Gaussian mixture. Internat. J. Pattern Recognition Artificial Intell. 18 (8), 1501–1512.

Ma, J., Wang, L., 2006. BYY harmony learning on finite mixture: Adaptive gradient implementation and a floating RPCL mechanism. Neural Process. Lett. 24 (1), 19–40.

Ma, J., Wang, T., 2006. A cost-function approach to rival penalized competitive learning (RPCL). IEEE Trans. Systems Man Cybernet. Part B: Cybernet. 36 (4), 722–737.

Ma, J., Wang, T., Xu, L., 2004. A gradient BYY harmony learning rule on Gaussian mixture with automated model selection. Neurocomputing 54, 481–487.

Ma, J., Gao, B., Wang, Y., Cheng, Q., 2005. Conjugate and natural gradient rules for BYY harmony learning on Gaussian mixture with automated model selection. Internat. J. Pattern Recognition Artificial Intell. 19, 701–713.

Millgan, G.W., Copper, M.C., 1985. An examination of procedures for determining the number of clusters in a data set. Psychometrika 46, 187–199.

Render, R.A., Walker, H.F., 1984. Mixture densities, maximum likelihood and the EM algorithm. SIAM Rev. 26 (2), 195–239.

Roberts, S.J., Everson, R., Rezek, I., 2000. Maximum certainty data partitioning. Pattern Recognition 33, 833–839.

Scott, A.J., Symons, M.J., 1971. Clustering methods based on likelihood ratio criteria. Biometrics 27, 387–397.

Xu, L., 1995. Ying–Yang machine: A Bayesian–Kullback scheme for unified learnings and new results on vector quantization. In: Proc. 1995 Internat. Conf. on Neural Information Processing, ICONIP'95, 30 October–3 November, vol. 2, pp. 977–988.

Xu, L., 2001. Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on Gaussian mixtures, three-layer nets and ME-RBF-SVM models. Internat. J. Neural Syst. 11 (1), 43–69.

Xu, L., 2002a. Ying–Yang learning. In: Arbib, Michael A. (Ed.), The Handbook of Brain Theory and Neural Networks, second ed. The MIT Press, Cambridge, MA, pp. 1231–1237.

Xu, L., 2002b. BYY harmony learning, structural RPCL, and topological self-organizing on mixture modes. Neural Networks 15, 1231–1237.

Xu, L., Krzyak, A., Oja, E., 1993. Rival penalized competitive learning for clustering analysis, RBF net, and curve detection. IEEE Trans. Neural Networks 4 (4), 636–649.

Xu, L. 1998. Rival penalized competitive learning, finite mixture, and multisets clustering. In: Proc. 1998 IEEE Int. Joint Conf. On Neural Networks, Anchrorage, Alaska, May 4–9, vol. 3, pp. 251–2530.