

ENTROPY PENALIZED AUTOMATED MODEL SELECTION ON GAUSSIAN MIXTURE*

JINWEN MA

*Department of Information Science, School of Mathematical Sciences & LMAM,
Peking University, Beijing, 100871, P. R. China
jwmamath.pku.edu.cn*

TAIJUN WANG

*Department of Wireless Engineering, Southeast University,
Nanjing, 210018, P. R. China*

Gaussian mixture modeling is a powerful approach for data analysis and the determination of the number of Gaussians, or clusters, is actually the problem of Gaussian mixture model selection which has been investigated from several respects. This paper proposes a new kind of automated model selection algorithm for Gaussian mixture modeling via an entropy penalized maximum-likelihood estimation. It is demonstrated by the experiments that the proposed algorithm can make model selection automatically during the parameter estimation, with the mixing proportions of the extra Gaussians attenuating to zero. As compared with the BYY automated model selection algorithms, it converges more stably and accurately as the number of samples becomes large.

Keywords: Gaussian mixture; model selection; maximum-likelihood estimation; Shannon entropy; penalty.

1. Introduction

Since Gaussian mixture populations are very common in various fields of practical applications, Gaussian mixture modeling is a powerful approach for data analysis. Actually, considerable effort has been made on Gaussian mixture modeling as well as its applications for clustering analysis on a sample data set.¹⁵ Although there have been various statistical or unsupervised competitive learning methods to do such a task, e.g. EM algorithm¹⁷ for maximum-likelihood, K -means algorithm¹⁴ for the Mean Square Error (MSE) clustering and the self-organizing network for hyperellipsoidal clustering (HEC),⁹ it is usually assumed that the number K of Gaussians, or clusters, in the data set is pre-known. However, in many instances this key information is not available and then the selection of an appropriate number of Gaussians must be made before or during the estimation of the parameters in the

*This work was supported by the Natural Science Foundation of China for Project 60071004.

mixture. Since the number K of Gaussians is just the scale of the mixture model, its determination is actually the problem of model selection for Gaussian mixture modeling. In fact, it is a rather difficult problem.⁸

The earliest possible method for solving this model selection problem may be to choose the optimal number K^* of Gaussians (or clusters) by the Akaike's information criterion² or its extensions.^{4,5,19} Similarly, several indexes of cluster validity have been also proposed to choose the optimal number K^* (e.g. Refs. 3, 16 and 21). But the process of evaluating an information criterion or validity index incurs a large computational cost since we need to repeat the entire parameter estimation process at a number of different values of K , even though such a process is attempted to be organized in a more efficient way, e.g. embedding the checking of the criterion value within clustering as in ISODATA.

From the respect of neural networks, competitive learning (CL) has been developed for clustering analysis and vector quantization.⁶ However, the conventional competitive learning algorithms such as the classical competitive learning algorithm¹⁸ and the frequency sensitive competitive learning algorithm,¹ can be only considered as adaptive versions of K -means algorithm and thus they are unable to solve this model selection problem. In order to do so, Xu, Krzyzak and Oja proposed a new kind of competitive learning algorithm, called rival penalized competitive learning (RPCL) algorithm,²² which has the ability of automatically allocating an appropriate number of weight vectors for a sample data set, with the other extra ones being pushed far away from the sample data. Therefore, the RPCL algorithm can be used to solve the above model selection problem by choosing a number K of the weight vectors which is surely larger than the number of actual clusters in the sample data set. In some extended versions of the RPCL algorithm,²⁴ the weight vectors have been generalized to Gaussian densities, which makes it more suitable for the RPCL algorithm to be applied to solving the model selection problem for Gaussian mixture. Recently, Ma, Wang and Xu proposed a cost function by which a generalized RPCL algorithm, called distance sensitive RPCL algorithm, has been proposed with a more powerful performance and easier implementation.¹⁰

Proposed in 1995²³ and systematically developed in the past years,^{25–27} Bayesian Ying–Yang (BYY) harmony learning acts as a general statistical learning framework not only for understanding several existing major learning approaches but also for tackling the learning problem on a set of finite samples with a new learning mechanism that makes model selection to be implemented automatically during parameter learning via a new class of model selection criteria. Specifically, the BYY harmony learning can be applied to the model selection problem for Gaussian mixture modeling. In fact, Ma, Wang and Xu have already implemented this new mechanism on a BI-architecture and a B-architecture of the BYY system via a gradient, iterative and two annealing algorithms to solve the model selection problem automatically during the parameter learning.^{11–13,20} It has been demonstrated by simulation experiments that the number of Gaussians can be automatically selected for the sample data set, with the mixing proportions of the extra Gaussians attenuating to zero.

In the current paper, based on the fact that reducing the Shannon entropy of the mixing proportions in a Gaussian mixture can lead to some of these proportions to attenuate to zero, i.e. to eliminate the corresponding Gaussians from the mixture, we proposed an entropy penalized maximum-likelihood estimation (MLE) iterative algorithm which can also make the parameter estimation with automated model selection. It is further demonstrated by the simulation experiments.

In the sequel, the entropy penalized maximum-likelihood estimation (MLE) iterative algorithm is derived in Sec. 2. In Sec. 3, several simulation experiments are conducted to demonstrate the proposed algorithm, with a comparison between it and those BYY model selection algorithms. A brief conclusion is given in Sec. 4.

2. Entropy Penalized MLE Iterative Algorithm

We consider the following Gaussian mixture model:

$$P(x|\Theta) = \sum_{j=1}^K \alpha_j P(x|m_j, \Sigma_j), \quad \alpha_j \geq 0, \quad \sum_{j=1}^K \alpha_j = 1 \tag{1}$$

where

$$P(x|m_j, \Sigma_j) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-m_j)^T \Sigma_j^{-1} (x-m_j)} \tag{2}$$

and where K is the number of Gaussians, x denotes a sample vector and d is the dimensionality of x . The parameter vector Θ consists of the mixing proportions α_j , the mean vectors m_j , and the covariance matrices $\Sigma_j = (\sigma_{pq}^{(j)})_{d \times d}$ which are assumed symmetric and positive definite.

In this mixture model, α_j is the proportion or probability of Gaussian (component) j in the mixture population. Then, the Shannon entropy of these mixing proportions, i.e. $I(\alpha) = \sum_{j=1}^K \alpha_j \log \alpha_j$, represents the uncertainty of the distribution of these mixing proportions. Here and hereafter, the base of the logarithm is always the natural number. According to the property of Shannon entropy, as K increases, $I(\alpha)$ generally increases. Oppositely, as $I(\alpha)$ decreases, some mixing proportions may be forced to be zero or a small number. That is, the corresponding Gaussians are essentially eliminated from the mixture. Thus, the number of actual Gaussians is decreased. According to this characteristic, we can penalize the MLE by adding certain minus Shannon entropy of the mixing proportions to the log maximum likelihood (ML) function so that the entropy penalized MLE is able to make the parameter estimation with automated model selection. In the following, we will realize this idea by proposing an entropy penalized ML objective function and establishing an iterative algorithm to maximize it.

Given a sample data set $\mathcal{S} = \{x_t\}_{t=1}^N$ from a Gaussian mixture with K^* components and $K \geq K^*$, the log likelihood function on the mixture model $P(x|\Theta)$ is given by

$$L(\Theta) = \sum_{t=1}^N \log P(x_t|\Theta). \tag{3}$$

The entropy penalized ML objective function can be constructed by

$$f(\Theta) = L(\Theta) - \beta NI(\alpha), \tag{4}$$

where $\beta > 0$ is the penalty factor which is a constant.

According to this objective function $f(\Theta)$, we have an entropy penalized maximum-likelihood estimate:

$$\hat{\Theta} = \arg \max_{\Theta} f(\Theta). \tag{5}$$

By Lagrange’s method of multipliers for $\hat{\Theta}$, we begin to define the λ -function:

$$L(\Theta, \lambda) = f(\Theta) + \lambda \left(1 - \sum_{j=1}^K \alpha_j \right). \tag{6}$$

Using the general methods for matrix derivatives,⁷ we are led to the derivatives of $L(\Theta, \lambda)$ with respect to each α_j, m_j, Σ_j and λ as follows:

$$\frac{\partial L(\Theta, \lambda)}{\partial \alpha_j} = \sum_{t=1}^N \frac{P(x_t|m_j, \Sigma_j)}{P(x_t|\Theta)} + \beta N(1 + \log \alpha_j) - \lambda; \tag{7}$$

$$\frac{\partial L(\Theta, \lambda)}{\partial m_j} = \sum_{t=1}^N \frac{\alpha_j P(x_t|m_j, \Sigma_j)}{P(x_t|\Theta)} \Sigma_j^{-1} (x_t - m_j)(-1); \tag{8}$$

$$\frac{\partial L(\Theta, \lambda)}{\partial \Sigma_j} = \sum_{t=1}^N \frac{\alpha_j P(x_t|m_j, \Sigma_j)}{P(x_t|\Theta)} \frac{1}{2} \Sigma_j^{-1} [(x_t - m_j)(x_t - m_j)^T - \Sigma_j] \Sigma_j^{-1}; \tag{9}$$

$$\frac{\partial L(\Theta, \lambda)}{\partial \lambda} = 1 - \sum_{j=1}^K \alpha_j. \tag{10}$$

Letting these derivatives to zero and introducing the *posteriori* probabilities:

$$h_j(t) = \frac{\alpha_j P(x_t|m_j, \Sigma_j)}{P(x_t|\Theta)}, \tag{11}$$

we have the following equalities that hold at $\hat{\Theta}$:

$$\sum_{t=1}^N h_j(t) + \beta N \alpha_j (1 + \log \alpha_j) - \alpha_j \lambda = 0; \tag{12}$$

$$\sum_{t=1}^N h_j(t) (x_t - m_j) = 0; \tag{13}$$

$$\sum_{t=1}^N h_j(t) [(x_t - m_j)(x_t - m_j)^T - \Sigma_j] = 0; \tag{14}$$

$$\sum_{j=1}^K \alpha_j = 1. \tag{15}$$

According to Eqs. (12) and (15), we further have

$$\begin{aligned} \lambda &= \sum_{j=1}^K \sum_{t=1}^N h_j(t) + \beta N \sum_{j=1}^K (\alpha_j + \alpha_j \log \alpha_j) \\ &= N + \beta N(1 - I(\alpha)) \\ &= N(1 + \beta(1 - I(\alpha))). \end{aligned} \tag{16}$$

Then, from the Eqs. (12)–(14), we obtain the following iterative algorithm for $\hat{\Theta}$:

$$\alpha_j^+ = \frac{1}{N(1 + \beta(1 - I(\alpha)))} \left(\sum_{t=1}^N h_j(t) + \beta N \alpha_j (1 + \log \alpha_j) \right); \tag{17}$$

$$m_j^+ = \frac{1}{\sum_{t=1}^N h_j(t)} \sum_{t=1}^N h_j(t) x_t; \tag{18}$$

$$\Sigma_j^+ = \frac{1}{\sum_{t=1}^N h_j(t)} \sum_{t=1}^N h_j(t) (x_t - m_j)(x_t - m_j)^T. \tag{19}$$

As compared with the EM algorithm for Gaussian mixture, this iterative algorithm implements a penalty mechanism on the mixing proportions during the iterations, which leads to the automated model selection. The penalty factor β can be selected by experience.

3. Simulation Results

In this section, simulation experiments are carried out to demonstrate the entropy penalized MLE iterative algorithm for both model selection and parameter estimation on a set of sample data from a Gaussian mixture. Moreover, this iterative algorithm is compared with the BYY model selection algorithms.

3.1. The sample data

We begin with a description of the seven sets of sample data used for our simulation experiments. We conducted four Monte Carlo experiments in which samples were drawn from a mixture of four or three bivariate Gaussians densities (i.e. $d = 2$).

As shown in Fig. 1, each data set of samples are generated with a certain degree of overlap among the Gaussians (clusters) in the mixture. They are four typical sets of sample data from Gaussian mixtures. The Gaussians in \mathcal{S}_1 are sphere-shaped, with equal number of samples. But those in \mathcal{S}_2 are ellipse-shaped, with different numbers of samples. Moreover, \mathcal{S}_3 consists of three very flat Gaussians and \mathcal{S}_4 has a small number of samples, with similar structure as \mathcal{S}_2 . The detailed parameters for these four sets of sample data are given in Table 1 where $m_i, \Sigma_i = [\sigma_{jk}^i]_{2 \times 2}, \alpha_i$ and N_i denote the mean vector, covariance matrix, mixing proportion and the number of samples of the i th cluster (Gaussian), respectively.

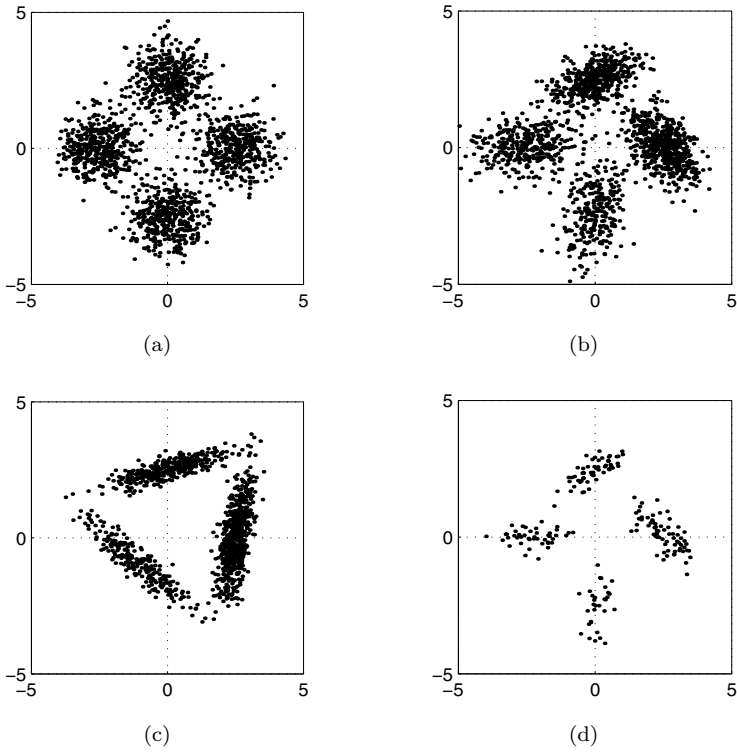


Fig. 1. Four sets of sample data used in the experiments. (a) Set \mathcal{S}_1 ; (b) Set \mathcal{S}_2 ; (c) Set \mathcal{S}_3 ; (d) Set \mathcal{S}_4 .

Table 1. The parameters to the four sets of sample data in the actual Gaussian mixtures.

The Sample Set	Gaussian	m_i	σ_{11}^i	σ_{12}^i	σ_{22}^i	α_i	N_i
\mathcal{S}_1 ($N = 1600$)	Gaussian 1	(2.50, 0)	0.50	0	0.50	0.25	400
	Gaussian 2	(0, 2.50)	0.50	0	0.50	0.25	400
	Gaussian 3	(-2.50, 0)	0.50	0	0.50	0.25	400
	Gaussian 4	(0, -2.50)	0.50	0	0.50	0.25	400
\mathcal{S}_2 ($N = 1600$)	Gaussian 1	(2.50, 0)	0.45	-0.25	0.55	0.34	544
	Gaussian 2	(0, 2.50)	0.65	0.20	0.25	0.28	448
	Gaussian 3	(-2.50, 0)	1	0.10	0.35	0.22	352
	Gaussian 4	(0, -2.50)	0.30	0.15	0.80	0.16	256
\mathcal{S}_3 ($N = 1200$)	Gaussian 1	(2.50, 0)	0.10	-0.20	1.25	0.50	600
	Gaussian 2	(0, 2.50)	1.25	0.35	0.15	0.30	360
	Gaussian 3	(-1, -1)	1	-0.80	0.75	0.20	240
\mathcal{S}_4 ($N = 200$)	Gaussian 1	(2.50, 0)	0.28	-0.20	0.32	0.34	68
	Gaussian 2	(0, 2.50)	0.34	0.20	0.22	0.28	56
	Gaussian 3	(-2.50, 0)	0.50	0.04	0.12	0.22	44
	Gaussian 4	(0, -2.50)	0.10	0.05	0.50	0.16	32

3.2. Automated model selection

We implement the entropy penalized MLE iterative algorithm on those four sample data sets always with $K \geq K^*$, respectively. The parameters are initialized randomly within certain intervals, satisfying the required constraints. The penalty factor β is selected to be 0.15. In all the experiments, the algorithm is stopped when $|f(\Theta_k^{\text{new}}) - f(\Theta_k^{\text{old}})| < 10^{-7}$.

The experimental results on \mathcal{S}_1 and \mathcal{S}_2 are given in Figs. 2 and 3, respectively, with case $K = 8$ and $K^* = 4$. We observe that four Gaussians are finally located accurately, while the mixing proportions of the other four Gaussians were reduced to 0.002 or below, i.e. these Gaussians are extra and can be discarded. That is, the correct number of the Gaussians has been automatically determined on these data sets.

Moreover, the experiment has been made on \mathcal{S}_3 with case $K = 8, K^* = 3$. As shown in Fig. 4, the actual Gaussians are very flat. However, three Gaussians are still located accurately, while the mixing proportions of the other five extra Gaussians are no more than 0.0012. That is, the correct number of the Gaussians can still be determined on such a special data set. Furthermore, the iterative algorithm is also implemented on \mathcal{S}_4 with $K = 8, K^* = 4$. As shown in Fig. 5, each cluster has a small number of samples, the correct number of Gaussians can still be detected, with the mixing proportions of other four extra Gaussians reduced below 0.008.

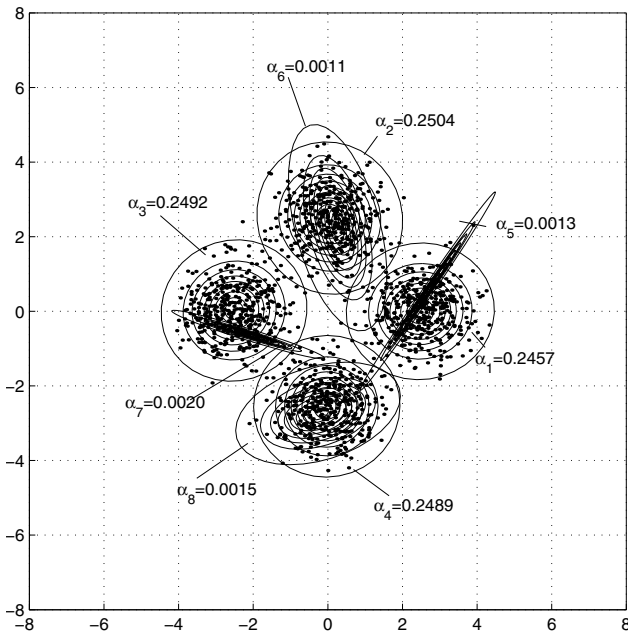


Fig. 2. The experimental result on \mathcal{S}_1 (stopped after 35,015 iterations). In this and the following three figures, the contour lines of each Gaussian are retained unless its density is less than e^{-3} [peak].

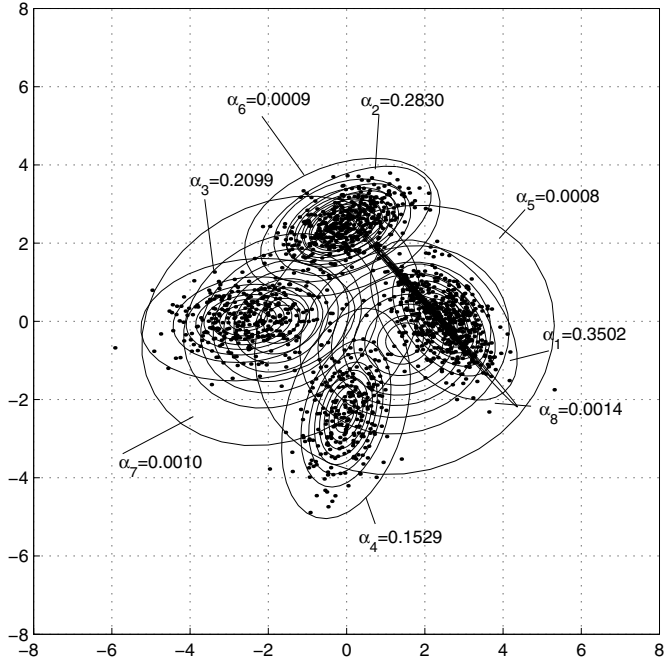


Fig. 3. The experimental result on S_2 (stopped after 33,747 iterations).

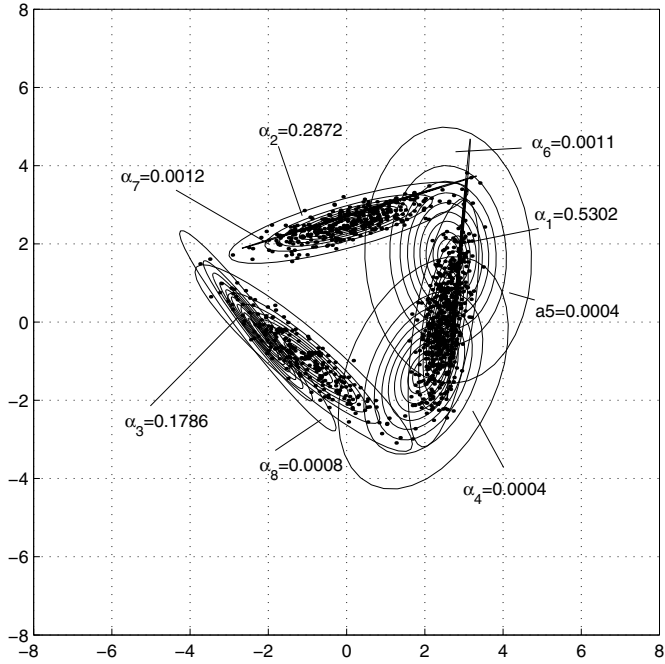


Fig. 4. The experimental result on S_3 (stopped after 5,497 iterations).

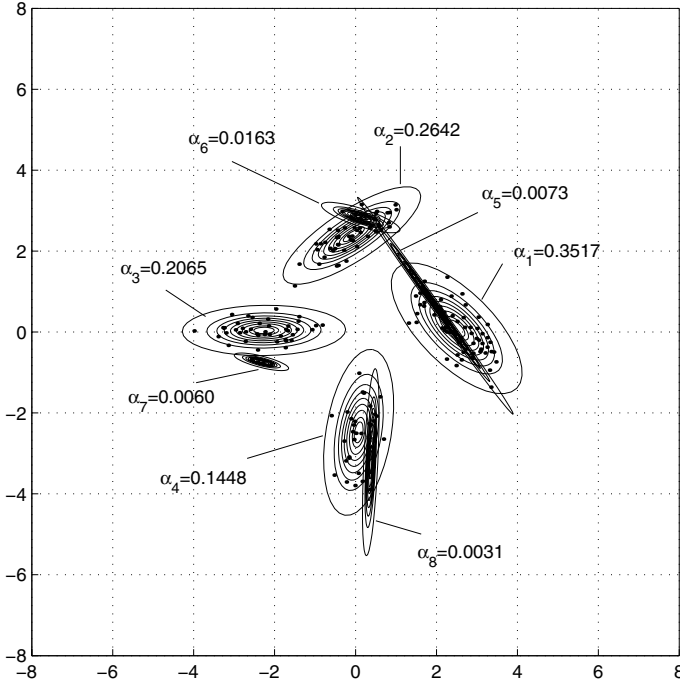


Fig. 5. The experimental result on \mathcal{S}_4 (stopped after 11,339 iterations).

Table 2. The resulting parameters of the iterative algorithm on each of the four sets of sample data.

Sample Set	Gaussian	m_i	σ_{11}^i	σ_{12}^i	σ_{22}^i	α_i
\mathcal{S}_1	Gaussian 1	(2.4785, -0.0010)	0.4924	0.0103	0.4200	0.2457
	Gaussian 2	(0.0616, 2.4988)	0.4779	-0.0236	0.5219	0.2504
	Gaussian 3	(-2.5135, 0.0221)	0.4780	0.0186	0.4498	0.2492
	Gaussian 4	(-0.0285, -2.5475)	0.4814	0.0058	0.4504	0.2489
\mathcal{S}_2	Gaussian 1	(2.4875, 0.0344)	0.4403	-0.2362	0.5255	0.3502
	Gaussian 2	(-0.0166, 2.4829)	0.5951	0.1987	0.2750	0.2830
	Gaussian 3	(-2.5208, 0.043)	0.9377	0.0765	0.3069	0.2099
	Gaussian 4	(-0.0158, -2.3545)	0.3393	0.1771	0.9056	0.1529
\mathcal{S}_3	Gaussian 1	(2.4773, -0.0152)	0.1026	-0.2051	1.2636	0.5302
	Gaussian 2	(0.0527, 2.5376)	1.1736	0.3114	0.1350	0.2872
	Gaussian 3	(-1.0963, -0.9217)	0.9659	-0.7671	0.7138	0.1786
\mathcal{S}_4	Gaussian 1	(2.4999, 0.0911)	0.3269	-0.2234	0.3260	0.3517
	Gaussian 2	(-0.0911, 2.3847)	0.3621	0.2048	0.1817	0.2642
	Gaussian 3	(-2.3639, 0.0381)	0.5092	0.0061	0.0485	0.2065
	Gaussian 4	(0.0769, -2.4808)	0.0938	0.0746	0.5272	0.1448

3.3. *Parameter estimation*

In addition to the correct model selection on K , we further compare the converged values of parameters (discarding the extra Gaussians) with those parameters in the mixture that the samples come from. All the empirical results of the parameters in the experiments on the four sample data sets are put into Table 2 as follows.

Comparing the numbers of Table 2 with those corresponding in Table 1, we have found that the iterative algorithm converges usually with a low average error between the estimated parameters and the true parameters being less than 0.1.

3.4. *The penalty factor and the comparison*

We have further tested the penalty factor β in the range $[0, 1]$ by the simulation experiments on these four sample data sets. By a large number of simulation experiments, we have found that when $\beta \in [0.08, 0.21]$, the entropy penalized MLE iterative algorithm can converge correctly. However, when $\beta < 0.08$, the entropy penalized MLE iterative algorithm cannot always eliminate the extra Gaussians from the mixture, that is, the penalty power is too weak to eliminate all the extra Gaussians. On the other hand, when $\beta > 0.21$, the entropy penalized MLE iterative algorithm cannot always maintain the correct number of Gaussians in the mixture, that is, the penalty power is too strong to maintain the actual Gaussians in the mixture. Therefore, the penalty factor can be selected in a large range. According to the accuracy of the estimation and the universality, the best penalty factor is around 0.15, which is just the value of β we used in the above simulation experiments.

Finally, we have made a comparison between the entropy penalized MLE iterative algorithm and the BYY model selection algorithms^{11–13,20} on these four sample data sets. We have found that the two kinds of algorithms generally have similar results. However, when the number of samples are large such as $\mathcal{S}_1, \mathcal{S}_2$, the entropy penalized MLE iterative algorithm converges more stably and accurately, however it costs more time. But when the number of samples is small such as \mathcal{S}_4 , the BYY model selection algorithm converges more accurately.

4. **Conclusions**

We have investigated the automated model selection on Gaussian mixture modeling from the point of view of penalizing the Shannon entropy of the mixing proportions on maximum-likelihood estimation. An entropy penalized MLE iterative algorithm is derived for automated model selection with parameter estimation. By simulation experiments we have found that as long as the overlap among the Gaussians or clusters in a data set is not too serious, the number of Gaussians can be correctly determined automatically with a good estimation of parameters of each Gaussian, even on a data set of a small sample size. In comparison with the BYY model selection algorithms, the entropy penalized MLE iterative algorithm converges more stably and accurately when the number of samples becomes large.

References

1. S. C. Ahalt, A. K. Krishnamurty, P. Chen and D. E. Melton, Competitive learning algorithm for vector quantization, *Neural Networks* **3**(3) (1990) 277–291.
2. H. Akaike, A new look at the statistical model identification, *IEEE Trans. Autom. Contr.* **A19** (1974) 716–723.
3. J. C. Bezdek and N. R. Pal, Some new indexes of cluster validity, *IEEE Trans. Syst. Man Cybern.* **B28**(3) (1998) 301–315.
4. H. Bozdogan, *Multiple Sample Cluster Analysis and Approaches to Validity Studies in Clustering Individuals*, Doctoral dissertation, University of Illinois at Chicago Circle (1981).
5. H. Bozdogan, Model selection and Akaike's information criterion: the general theory and its analytical extensions, *Psychometrika* **52**(3) (1987) 345–370.
6. P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach* (Prentice-Hall, London, UK, 1982).
7. S. R. Gerald, *Matrix Derivatives*, Lecture Notes in Statistics, Vol. 2 (Marcel Dekker, NY, 1980).
8. J. A. Hartigan, Distribution problems in clustering, in *Classification and Clustering*, ed. J. Van Ryzin (Academic Press, NY, 1977), pp. 45–72.
9. A. K. Jain and R. C. Dubes, *Algorithm for Clustering Data* (Englewood Cliffs, Prentice Hall, NJ, 1988).
10. J. Ma, T. Wang and L. Xu, Convergence analysis of rival penalized competitive learning (RPCL) algorithm, in *Proc. Int. Joint Conf. Neural Networks (IJCNN'02)*, Honolulu, Hawaii, May 12–17, 2002, Vol. 2, pp. 1596–1601.
11. J. Ma, T. Wang and L. Xu, An adaptive BYY harmony learning algorithm and its relation to rewarding and penalizing competitive learning mechanism, in *Proc. 6th Int. Conf. Signal Processing (ICSP '02)*, Beijing, China, August 26–30, 2002, Vol. 2, pp. 1154–1158.
12. J. Ma, T. Wang and L. Xu, An iterative algorithm for BYY learning on Gaussian mixture with automated model selection, in *Proc. 2003 Int. Conf. Neural Networks and Signal Processing (ICNNSP'03)*, Nanjing, China, December 14–17, 2003, Vol. 1, pp. 7–10.
13. J. Ma, T. Wang and L. Xu, An annealing approach to BYY harmony learning on Gaussian mixture with automated model selection, in *Proc. 2003 Int. Conf. Neural Networks and Signal Processing (ICNNSP'03)*, Nanjing, China, December 14–17, 2003, Vol. 1, pp. 23–28.
14. J. Mao and A. K. Jain, A self-organizing network for hyperellipsoidal clustering (HEC), *IEEE Trans. Neural Networks* **7**(1) (1996) 16–29.
15. G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications to Clustering* (Marcel Dekker, NY, 1988).
16. R. M. Ramze, B. P. F. Lelièveveldt and J. H. C. Reiber, A new cluster validity index for the fuzzy c-means, *Patt. Recogn. Lett.* **19** (1998) 273–246.
17. R. A. Render and H. F. Walker, Mixture densities, maximum likelihood and the EM algorithm, *SIAM Rev.* **26**(2) (1984) 195–239.
18. D. E. Rumelhart and D. Zipser, Feature discovery by competitive learning, *Cogn. Sci.* **9**(1) (1985) 75–112.
19. G. Schwarz, Estimating the dimension of a model, *Ann. Statist.* **6**(2) (1978) 461–464.
20. J. Ma, T. Wang and L. Xu, A gradient BYY harmony learning rule on Gaussian mixture with automated model selection, *Neurocomputing* **56** (2004) 481–487.

21. X. L. Xie and G. A. Beni, A validity method for fuzzy clustering, *IEEE Trans. Patt. Anal. Mach. Intell.* **13**(8) (1991) 843–847.
22. L. Xu, A. Krzyzak and E. Oja, Rival penalized competitive learning for clustering analysis, RBF net, and curve detection, *IEEE Trans. Neural Networks* **4** (1993) 636–648.
23. L. Xu *et al.*, A unified learning scheme: Bayesian–Kullback YING–YANG machine, in *Advances in Neural Information Processing Systems*, eds. D. S. Touretzky *et al.* (MIT Press, Cambridge, MA, 1996), Vol. 8, pp. 444–450; A part of its preliminary version on *Proc. 1995 Int. Conf. Neural Information Processing*, pp. 977–988.
24. L. Xu, Rival penalized competitive learning, finite mixture, and multisets clustering, in *Proc. 1998 IEEE Int. Joint Conf. Neural Networks*, Vol. 3, pp. 251–253.
25. L. 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, *Int. J. Neural Syst.* **11**(1) (2002) 43–69.
26. L. Xu, BYY harmony learning, structural RPCL, and topological self-organizing on mixture modes, *Neural Networks* **15** (2002) 1231–1237.
27. L. Xu, Ying–Yang learning, in *The Handbook of Brain Theory and Neural Networks*, ed. M. A. Arbib, 2nd edn. (The MIT Press, 2002), pp. 1231–1237.



Jinwen Ma received the M.S. degree in applied mathematics from Xi'an Jiaotong University in 1988 and the Ph.D. in probability theory and statistics from Nankai University in 1992. From July 1992 to November 1999, he

was a Lecturer and Associate Professor at the Department of Mathematics, Shantou University. He has been a full Professor at the Institute of Mathematics, Shantou University since December, 1999. In September 2001, he was transferred to the Department of Information Science at the School of Mathematical Sciences, Peking University. During 1995 and 2003, he also made several visits to the Department of Computer Science and Engineering, the Chinese University of Hong Kong as a Research Associate or Fellow.

He has published over 40 academic papers on neural networks, pattern recognition, artificial intelligence and information theory.



Taijun Wang received the M.S. degree in signal and information processing from Southeast University, China, in 1982. He has been an Associate Professor at the Department of Radio Engineering of Southeast University

since 1988.

His research interests are related to digital signal processing, pattern recognition, artificial neural networks, statistical learning theory and scientific visualization.