International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 19, No. 5 (2005) 701–713

World Scientific
www.worldscientific.com

CONJUGATE AND NATURAL GRADIENT RULES FOR BYY HARMONY LEARNING ON GAUSSIAN MIXTURE WITH AUTOMATED MODEL SELECTION*

JINWEN MA†, BIN GAO, YANG WANG and QIANSHENG CHENG
Department of Information Science, School of Mathematical Sciences and LMAM
Peking University, Beijing 100871, China
†*jwma@math.pku.edu.cn*

Under the Bayesian Ying–Yang (BYY) harmony learning theory, a harmony function has been developed on a BI-directional architecture of the BYY system for Gaussian mixture with an important feature that, via its maximization through a general gradient rule, a model selection can be made automatically during parameter learning on a set of sample data from a Gaussian mixture. This paper further proposes the conjugate and natural gradient rules to efficiently implement the maximization of the harmony function, i.e. the BYY harmony learning, on Gaussian mixture. It is demonstrated by simulation experiments that these two new gradient rules not only work well, but also converge more quickly than the general gradient ones.

Keywords: Bayesian Ying–Yang learning; Gaussian mixture; automated model selection; conjugate gradient; natural gradient.

1. Introduction

As a powerful statistical model, Gaussian mixture has been widely applied to data analysis and there have been several statistical methods for its modeling (e.g. the method of moments,[3] the maximum likelihood estimation[4] and the expectation-maximization (EM) algorithm[12]). But it is usually assumed that the number of Gaussians in the mixture is pre-known. However, in many instances this key information is not available and the selection of an appropriate number of Gaussians must be made with the estimation of the parameters, which is a rather difficult task.[7]

The traditional approach to this task is to choose a best number k^* of Gaussians via some selection criterion. Actually, there have been many heuristic criteria in the statistical literature (e.g. Refs. 1, 5, 11, 13 and 14). However, the process of evaluating a criterion incurs large computational cost since the entire parameter

*This work was supported by the Natural Science Foundation of China for Projects 60071004 and 60471054.
†Author for correspondence.

estimating process is to be repeated at a number of different values of k. On the other hand, some heuristic learning algorithms (e.g. the greedy EM algorithm[15] and the competitive EM algorithm[20]) have also been constructed to apply a mechanism of split and merge on the estimated Gaussians to certain typical estimation methods like the EM algorithm at each iteration to search the best number of Gaussians in the data set. Obviously, these methods are also time consuming.

Recently, a new approach has been developed from the Bayesian Ying–Yang (BYY) harmony learning theory[16–19] with the feature that model selection can be made automatically during the parameter learning. In fact, it was already shown in Ref. 10 that the Gaussian mixture modeling problem in which the number of Gaussians is unknown can be equivalent to the maximization of a harmony function on a specific BI-directional architecture (BI-architecture) of the BYY system for the Gaussian mixture model and a gradient rule for maximization of this harmony function was also established. The simulation experiments showed that an appropriate number of Gaussians can be automatically allocated for the sample data set, with the mixing proportions of the extra Gaussians attenuating to zero. Moreover, an adaptive gradient rule was further proposed and analyzed for the general finite mixture model, and demonstrated well on a sample data set from Gaussian mixture.[9] On the other hand, 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 also proposed to make model selection automatically with parameter estimation on Gaussian mixture.[8]

In this paper, we propose two further gradient rules to efficiently implement the maximization of the harmony function in a Gaussian mixture setting. The first rule is constructed from the conjugate gradient of the harmony function, while the second rule is derived from Amari and Nagaoka's natural gradient theory.[2] It is demonstrated by simulation experiments that these two new gradient rules not only work well for automated model selection, but also converge more quickly than the general gradient ones.

In the sequel, the BYY harmony learning system and the harmony function are introduced in Sec. 2. The conjugate and natural gradient rules are then derived in Sec. 3. In Sec. 4, they are both demonstrated by simulation experiments, and finally a brief conclusion is made in Sec. 4.

2. BYY System and Harmony Function

A BYY system describes each observation $x \in \mathcal{X} \subset R^n$ and its corresponding inner representation $y \in \mathcal{Y} \subset R^m$ 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 and Ying machines, respectively. In this paper, y is only limited to be an integer variable, i.e. $y \in \mathcal{Y} = \{1, 2, \ldots, k\} \subset R$ with $m = 1$. Given a data set $D_x = \{x_t\}_{t=1}^N$, the task of learning on a BYY system consists of specifying all the aspects of $p(y|x), p(x), q(x|y), q(y)$ via a harmony learning principle implemented

by maximizing the functional

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

where z_q is a regularization term. The details of the derivation can be found in Ref. 17.

If both $p(y|x)$ and $q(x|y)$ are parametric, i.e. from a family of probability densities parametrized by θ, the BYY system is said to have a Bi-directional Architecture (BI-Architecture). For the Gaussian mixture modeling, we use the following specific BI-architecture of the BYY system. $q(j) = \alpha_j$ with $\alpha_j \geq 0$ 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 $x \in \mathcal{X} = R^n$. Moreover, the BI-architecture is constructed in 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, originally introduced in Ref. 16 as $J(k)$ and developed into this form in Ref. 17 being used as a selection criterion of the number k. Furthermore, we let $q(x|\theta_j)$ be a Gaussian density given by

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

where m_j is the mean vector and Σ_j is the covariance matrix which is assumed positive definite. As a result, this BI-architecture of the BYY 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 represent the probability distribution of the sample data in D_x.

According to the best harmony learning principle of the BYY system[18] as well as the experimental results of the general gradient rules obtained in Refs. 9 and 10, the maximization of $J(\Theta_k)$ can realize the automated model selection during parameter learning on a sample data set from a Gaussian mixture. That is, when we set k to be larger than the number k^* of actual Gaussians in the sample data set, it can cause k^* Gaussians in the estimated mixture match the actual Gaussians, respectively, and force the mixing proportions of the other $k - k^*$ extra Gaussians to attenuate to zero, i.e. eliminate them from the mixture. Here, in order to efficiently implement the maximization of $J(\Theta_k)$, we further derive two gradient rules with a better convergence behavior to solve the maximum solution of $J(\Theta_k)$ in the next section.

3. Conjugate and Natural Gradient Rules

For convenience of derivation, we let

$$\alpha_j = \frac{e^{\beta_j}}{\sum_{i=1}^{k} e^{\beta_i}}, \quad \Sigma_j = B_j B_j^T, \qquad j = 1, \ldots, k,$$

where $-\infty < \beta_1, \ldots, \beta_k < +\infty$ and B_j is a nonsingular square matrix. Under these transformations, the parameters in $J(\Theta_k)$ turn into $\{\beta_j, m_j, B_j\}_{j=1}^{k}$. Furthermore, we have the derivatives of $J(\Theta_k)$ with respect to β_j, m_j and B_j as follows. (See Ref. 6 for the derivation.)

$$\frac{\partial J(\Theta_k)}{\partial \beta_j} = \frac{\alpha_j}{N} \sum_{i=1}^{k} \sum_{t=1}^{N} h(i|x_t) U(i|x_t)(\delta_{ij} - \alpha_i), \tag{5}$$

$$\frac{\partial J(\Theta_k)}{\partial m_j} = \frac{\alpha_j}{N} \sum_{t=1}^{N} h(j|x_t) U(j|x_t) \Sigma_j^{-1}(x_t - m_j), \tag{6}$$

$$\operatorname{vec}\left[\frac{\partial J(\Theta_k)}{\partial B_j}\right] = \frac{\partial (B_j B_j^T)}{\partial B_j} \operatorname{vec}\left[\frac{\partial J(\Theta_k)}{\partial \Sigma_j}\right], \tag{7}$$

where δ_{ij} is the Kronecker function, vec$[A]$ denotes the vector obtained by stacking the column vectors of the matrix A, and

$$U(i|x_t) = \sum_{r=1}^{k} (\delta_{ri} - p(r|x_t)) \ln[\alpha_r q(x_t|m_r, \Sigma_r)] + 1,$$

$$h(i|x_t) = \frac{q(x_t|m_i, \Sigma_i)}{\sum_{r=1}^{k} \alpha_r q(x_t|m_r, \Sigma_r)}, \quad p(i|x_t) = \alpha_i h(i|x_t),$$

$$\frac{\partial J(\Theta_k)}{\partial \Sigma_j} = \frac{\alpha_j}{N} \sum_{t=1}^{N} h(j|x_t) U(j|x_t) \Sigma_j^{-1} \big[(x_t - m_j)(x_t - m_j)^T - \Sigma_j\big] \Sigma_j^{-1},$$

and

$$\frac{\partial (BB^T)}{\partial B} = I_{n\times n} \otimes B_{n\times n}^T + E_{n^2\times n^2} \cdot B_{n\times n}^T \otimes I_{n\times n},$$

where $\otimes$ denotes the Kronecker product (or tensor product), and

$$E_{n^2\times n^2} = \frac{\partial B^T}{\partial B} = (\Gamma_{ij})_{n^2\times n^2} = \begin{pmatrix} \Gamma_{11} & \Gamma_{12} & \cdots & \Gamma_{1n} \\ \Gamma_{21} & \Gamma_{22} & \cdots & \Gamma_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \Gamma_{n1} & \Gamma_{n2} & \cdots & \Gamma_{nn} \end{pmatrix}_{n^2\times n^2},$$

where Γ_{ij} is an $n \times n$ matrix whose (j, i)th element is just 1, with all the other elements being zero. With the above expression of $\frac{\partial (B_j B_j^T)}{\partial B_j}$, we have

$$\operatorname{vec}\left[\frac{\partial J(\Theta_k)}{\partial B_j}\right] = \frac{\alpha_j}{2N} \sum_{t=1}^{N} h(j|x_t) U(j|x_t)(I_{n\times n} \otimes B_{n\times n}^T + E_{n^2\times n^2} \cdot B_{n\times n}^T \otimes I_{n\times n}) \times \operatorname{vec}\big[\Sigma_j^{-1}(x_t - m_j)(x_t - m_j)^T \Sigma_j^{-1} - \Sigma_j^{-1}\big]. \tag{8}$$

Based on the above preparation, we can derive the conjugate and natural gradient rules as follows.

Combining these β_j, m_j, and B_j into a vector Φ_k, we can construct the conjugate gradient rule by

$$\Phi_k^{i+1} = \Phi_k^i + \eta \hat{S}_i, \tag{9}$$

where $\eta > 0$ is the learning rate, and the searching direction $\hat{S}_i$ is obtained from the following recursive iterations of the conjugate vectors:

$$S_1 = \nabla J(\Phi_k^1), \quad \hat{S}_1 = \frac{\nabla J(\Phi_k^1)}{\|\nabla J(\Phi_k^1)\|}$$

$$S_i = \nabla J(\Phi_k^i) + V_{i-1} S_{i-1}, \quad \hat{S}_i = \frac{S_i}{\|S_i\|}, \quad V_{i-1} = \frac{\|\nabla J(\Phi_k^i)\|^2}{\|\nabla J(\Phi_k^{i-1})\|^2},$$

where $\nabla J(\Phi_k)$ is just the general gradient vector of $J(\Phi_k) = J(\Theta_k)$ and $\|\cdot\|$ is the Euclidean norm.

As for the natural gradient rule, we further consider Φ_k as a point in the Riemann space. Then, we can construct a $k(n^2 + n + 1)$-dimensional statistical model $\mathcal{F} = \{P(x, \Phi_k) = q(x_t, \Theta_k) : \Phi_k \in \Xi\}$. The Fisher information matrix of the statistical model at a point Φ_k is defined as $G(\Phi_k) = [g_{ij}(\Phi_k)]$, where $g_{ij}(\Phi_k)$ is given by

$$g_{ij}(\Phi_k) = \int \partial_i l(x, \Phi_k) \partial_j l(x, \Phi_k) P(x, \Phi_k) dx, \tag{10}$$

where $\partial_i = \frac{\partial}{\partial \Phi_{ki}}$, i.e. the derivative with respect to the ith component of Φ_k, and $l(x, \Phi_k) = \ln P(x, \Phi_k)$. By the derivatives of $P(x_t, \Phi_k)$ with respect to β_j, m_j, B_j (See Ref. 2 for details):

$$\frac{\partial P(x_t, \Phi_k)}{\partial \beta_j} = \alpha_j q(x_t | m_j, \Sigma_j) \sum_{i=1}^{k} (\delta_{ij} - \alpha_i), \tag{11}$$

$$\frac{\partial P(x_t, \Phi_k)}{\partial m_j} = \alpha_j q(x_t | m_j, \Sigma_j) \Sigma_j^{-1} (x_t - m_j), \tag{12}$$

$$\mathrm{vec}\left[\frac{\partial P(x_t, \Phi_k)}{\partial B_j}\right] = \alpha_j q(x_t | m_j, \Sigma_j) \frac{\partial B_j^T B_j}{\partial B_j} \mathrm{vec}[\Sigma_j^{-1} [(x_t - m_j)(x_t - m_j)^T \Sigma_j^{-1} - \Sigma_j^{-1}], \tag{13}$$

we can easily get an estimate of $G(\Phi_k)$ on a sample data set via Eq. (10) under the law of large numbers. According to Amari and Nagaoka's natural gradient theory,[2] we have the following natural gradient rule:

$$\Phi_k^{i+1} = \Phi_k^i - \eta G^{-1}(\Phi_k^i) \nabla J(\Phi_k^i), \tag{14}$$

where $\eta > 0$ is the learning rate.

According to the theories of optimization and information geometry, the conjugate and natural gradient rules generally have a better convergence behavior than the general gradient ones, especially on the convergence rate, which will be further demonstrated by the simulation experiments in the next section.

4. Experimental Results

In this section, several simulation experiments are carried out to demonstrate the conjugate and natural gradient rules for the automated model selection as well as the parameter estimation on seven data sets from Gaussian mixtures. We also compare them with the batch and adaptive gradient learning algorithms.[9,10]

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

Using k^* to denote the number of Gaussians in the original mixture, i.e. the number of actual clusters in the sample data set, we implemented the conjugate and natural gradient rules on those seven sample data sets by setting a larger k

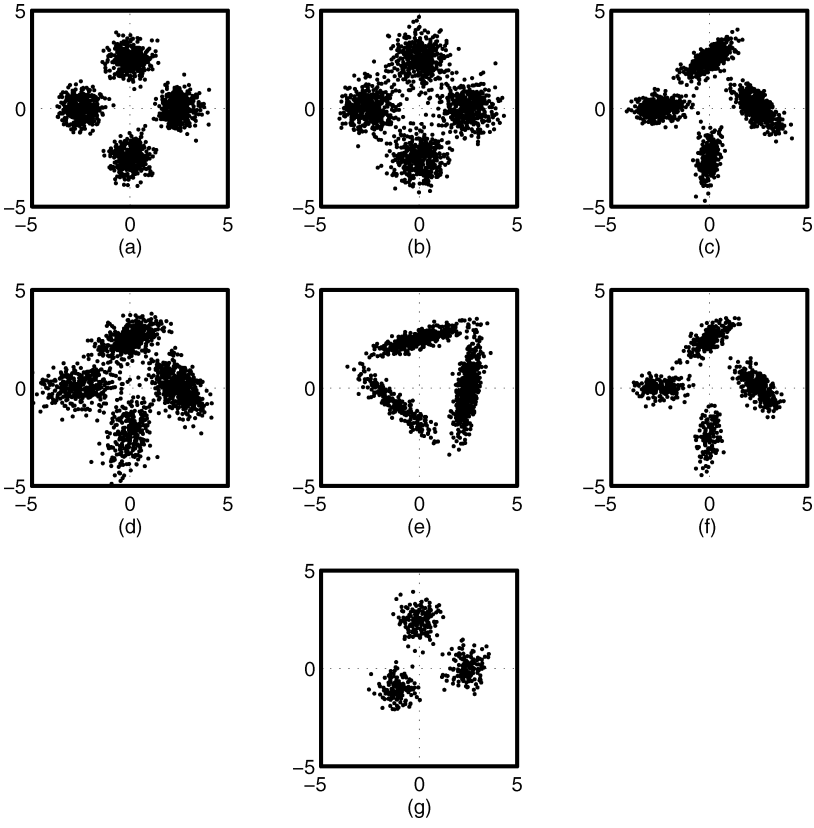


Fig. 1. Seven sets of sample data used in the experiments. (a). Set $\mathcal{S}_1$; (b). Set $\mathcal{S}_2$; (c). Set $\mathcal{S}_3$; (d). Set $\mathcal{S}_4$; (e). Set $\mathcal{S}_5$; (f). Set $\mathcal{S}_6$; (g). Set $\mathcal{S}_7$.

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
$\mathcal{S}_1$	Gaussian 1	(2.5, 0)	0.25	0	0.25	0.25	400
($N = 1600$)	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
$\mathcal{S}_2$	Gaussian 1	(2.5, 0)	0.5	0	0.5	0.25	400
($N = 1600$)	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
$\mathcal{S}_3$	Gaussian 1	(2.5, 0)	0.28	−0.20	0.32	0.34	544
($N = 1600$)	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
$\mathcal{S}_4$	Gaussian 1	(2.5, 0)	0.45	−0.25	0.55	0.34	544
($N = 1600$)	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
$\mathcal{S}_5$	Gaussian 1	(2.5, 0)	0.1	0.2	1.25	0.5	600
($N = 1200$)	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
$\mathcal{S}_6$	Gaussian 1	(2.5, 0)	0.28	−0.20	0.32	0.34	272
($N = 800$)	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
$\mathcal{S}_7$	Gaussian 1	(2.5, 0)	0.25	0	0.25	0.3333	150
($N = 450$)	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

($k \geq k^*$) and $\eta = 0.1$. Moreover, the other parameters were initialized randomly within certain intervals. In all the experiments, the learning was stopped when $|J(\Phi_k^{\text{new}}) - J(\Phi_k^{\text{old}})| < 10^{-5}$.

The experimental results of the conjugate and natural gradient rules on the data set $\mathcal{S}_2$ are given in Figs. 2 and 3, respectively, with case $k = 8$ and $k^* = 4$. We can observe that four Gaussians were finally located accurately, while the mixing proportions of the other four Gaussians were reduced to below 0.01, i.e. these Gaussians are extra and can be discarded. That is, the correct number of the clusters were detected on these data sets. Moreover, the experiments of the two gradient rules have been made on $\mathcal{S}_4$ with $k = 8, k^* = 4$. As shown in Figs. 4 and 5, clusters are typically elliptical. Again, four Gaussians are located accurately, while the mixing proportions of the other four extra Gaussians become less than 0.01. That is, the correct number of the clusters can still be detected on a general data set. Furthermore, the two gradient rules were also implemented on $\mathcal{S}_6$ with $k = 8, k^* = 4$. As shown in Figs. 6 and 7, each cluster has a small number of samples, the correct number of clusters can still be detected, with the mixing proportions of other four extra Gaussians reduced below 0.01.

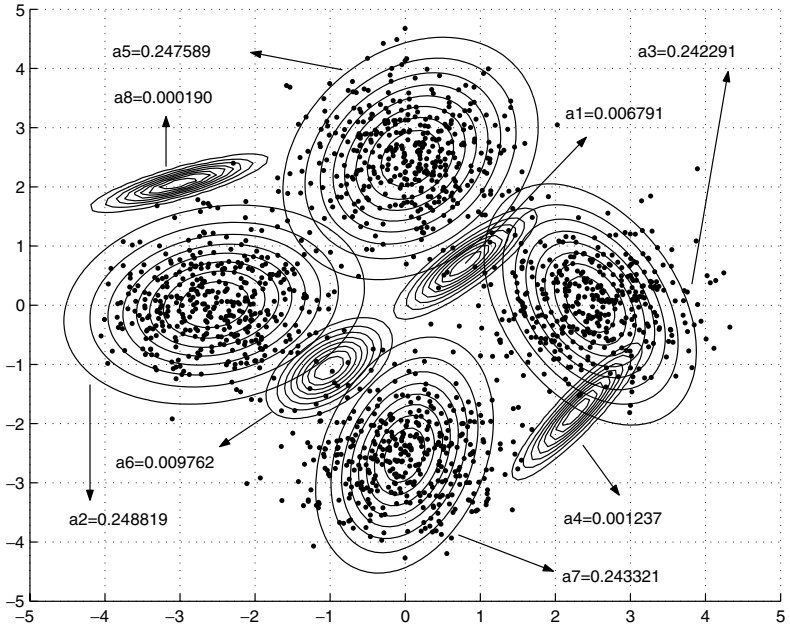


Fig. 2. The experimental result of the conjugate gradient rule (or algorithm) on the sample set S_2 (stopped after 63 iterations). In this and the following three figures, the contour lines of each Gaussian are retained unless its density is less than e^{-3} (peak).

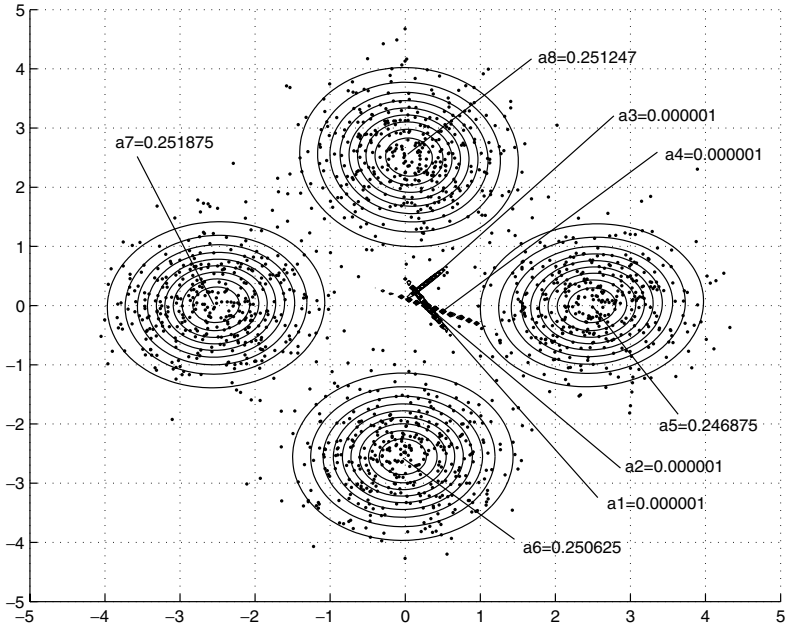


Fig. 3. The experimental result of the natural gradient rule on the sample set S_2 (stopped after 126 iterations).

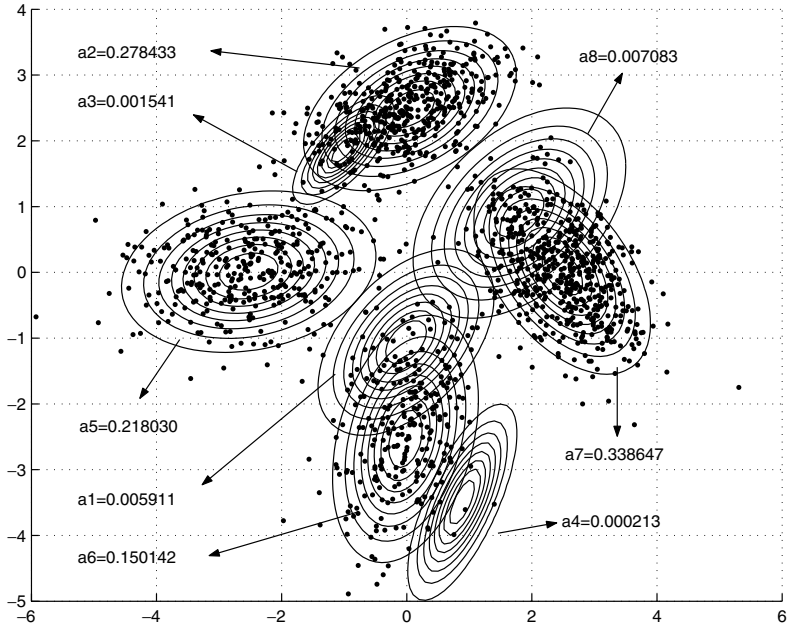


Fig. 4. The experimental result of the conjugate gradient rule on the sample set S_4 (stopped after 112 iterations).

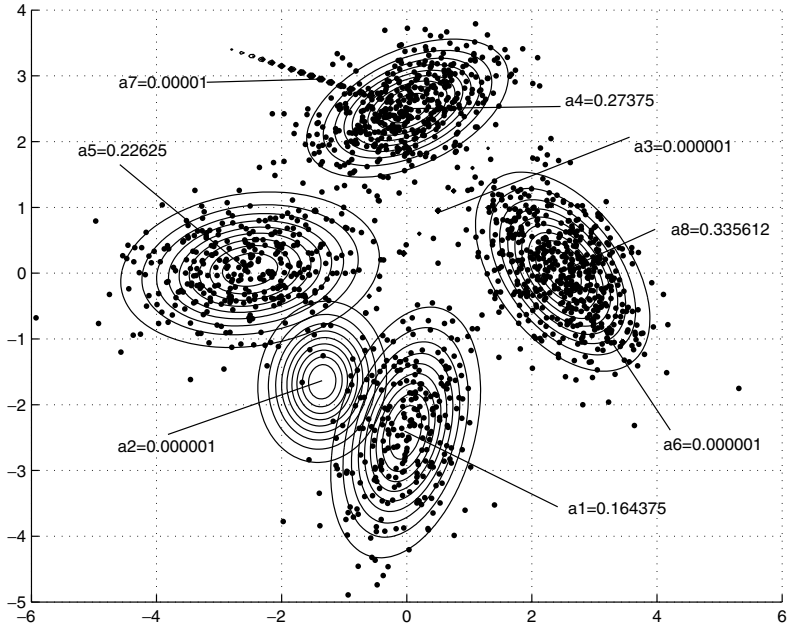


Fig. 5. The experimental result of the natural gradient rule on the sample set S_4 (stopped after 149 iterations).

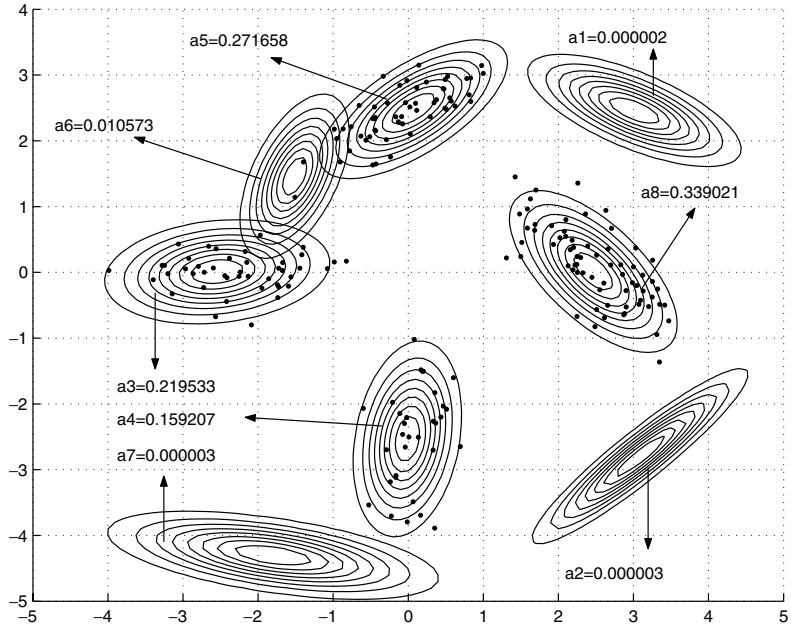


Fig. 6. The experimental result of the conjugate gradient rule on the sample set S_6 (stopped after 153 iterations).

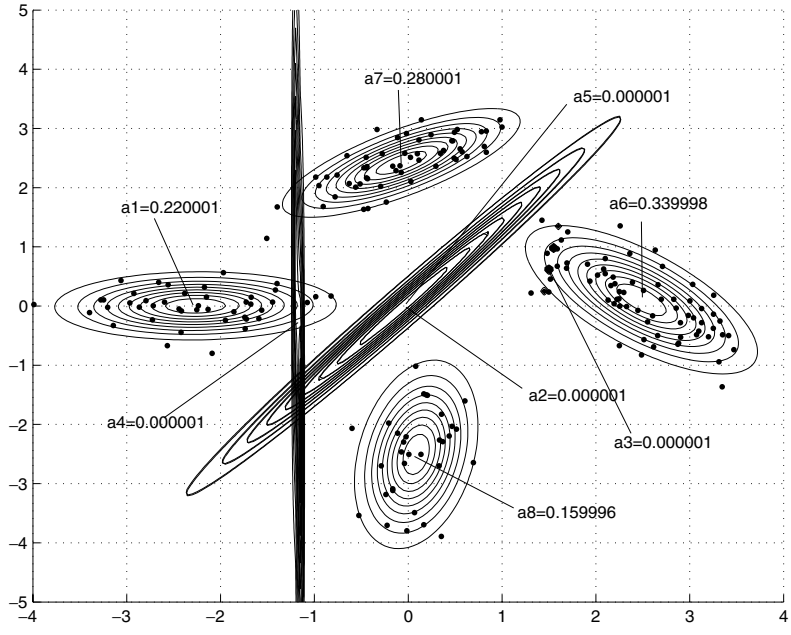


Fig. 7. The experimental result of the natural gradient rule on the sample set S_6 (stopped after 194 iterations).

Further experiments of the two gradient rules on the other sample sets were also made successfully for the correct number detection in similar cases. Actually, in many experiments, a failure rarely occurred for the correct number detection when we set k with $k^* \leq k \leq 3k^*$. However, they may lead to a wrong detection when $k > 3k^*$.

In addition to the correct number detection, we further compared the converged values of parameters (discarding the extra Gaussians) with those parameters in the original mixture from which the samples came from. We checked the results in these experiments and found that the conjugate and natural gradient rules converge with a lower average error between the estimated parameters and the true parameters. Actually, the average error of the parameter estimation with each rule was generally as good as that of the EM algorithm on the same data set with $k = k^*$.

In comparison with the simulation results of the batch and adaptive gradient rules[9,10] on these seven sets of sample data, we found that the conjugate and natural gradient rules converge more quickly than the two general gradient ones. Actually, for the most cases it had been demonstrated by simulation experiments that the number of iterations required by each of these two rules is only about one quarter to a half of the number of iterations required by either the batch or adaptive gradient rule.

As compared with each other, the conjugate gradient rule converges more quickly than the natural gradient rule, but the natural gradient rule obtains a more accurate solution on the parameter estimation.

5. Conclusion

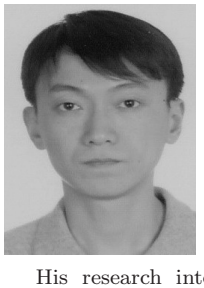
We have proposed the conjugate and natural gradient rules for the BYY harmony learning on Gaussian mixture with automated model selection. They are derived from the conjugate gradient method and Amari and Nagaoka's natural gradient theory for the maximization of the harmony function defined on Gaussian mixture model. The simulation experiments have demonstrated that both the conjugate and natural rules lead to the correct selection of the number of actual Gaussians as well as a good estimate for the parameters of the original Gaussian mixture. Moreover, they converge more quickly than the general gradient ones.

References

1. H. Akaike, A new look at the statistical model identification, *IEEE Trans. Autom. Contr.* **AC-19** (1974) 716–723.
2. S. Amari and H. Nagaoka, *Methods of Information Geometry* (American Mathematical Society, Providence, RI, 2000).
3. N. E. Day, Estimating the components of a mixture of normal distributions, *Biometrika* **56** (1969) 463–474.
4. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (John Wiley, NY, 1973).

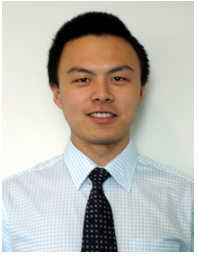
5. H. P. Friedman and J. Rubin, On some invariant criteria for grouping data, *J. Amer. Stat. Assoc.* **62** (1967) 1159–1178.
6. S. R. Gerald, *Matrix Derivatives* (Marcel Dekker, NY, 1980).
7. J. A. Hartigan, Distribution problems in clustering, *Classification and Clustering*, ed. J. Van Ryzin (Academic Press, NY, 1977), pp. 45–72.
8. J. Ma and T. Wang, Entropy penalized automated model selection on Gaussian mixture, *Int. J. Pattern Recognition and Artificial Intellegence* **18** (2004) 1501–1512.
9. 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. ICSP'02* **2** (2002) 1154–1158.
10. 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.
11. G. W. Millgan and M. C. Copper, An examination of procedures for determining the number of clusters in a data set, *Psychometrika* **46** (1985) 187–199.
12. R. A. Render and H. F. Walker, Mixture densities, maximum likelihood and the EM algorithm, *SIAM Rev.* **26** (1984) 195–239.
13. S. J. Robert, R. Everson and I. Rezek, Maximum certainty data partitioning, *Patt. Recogn.* **33** (2000) 833–839.
14. A. J. Scott and M. J. Symons, Clustering methods based on likelihood ratio criteria, *Biometrics* **27** (1971) 387–397.
15. N. Vlassis and A. Likas, A greedy EM algorithm for Gaussian mixture learning, *Neural Process. Lett.* **15** (2002) 77–87.
16. L. Xu, Ying-Yang machine: A Bayesian-Kullback scheme for unified learnings and new results on vector quantization, *Proc. 1995 Int. Conf. Neural Information Processing (ICONIP'95)* **2** (1995), pp. 977–988.
17. 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** (2001) 43–69.
18. L. Xu, Ying-Yang learning, *The Handbook of Brain Theory and Neural Networks*, 2nd ed., ed. M. A. Arbib (The MIT Press, Cambridge, MA, 2002), pp. 1231–1237.
19. L. Xu, BYY harmony learning, structural RPCL, and topological self-organzing on mixture modes, *Neural Networks* **15** (2002) 1231–1237.
20. B. Zhang, C. Zhang and X. Yi, Competitive EM algorithm for finite mixture model, *Patt. Recogn.* **37** (2004) 131–144.

Jinwen Ma received the M.S. degree in applied mathematics from Xi'an Jiaotong University in 1988 and the Ph.D. degree in probability theory and statistics from Nankai University in 1992. From July 1992 to November 1999, he was a Lecturer or Associate Professor at the Department of Mathematics, Shantou University. He has been a full professor at 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 visited the Chinese University of Hong Kong as a Research Associate or Fellow. He has published over 60 academic papers on neural networks, pattern recognition, artificial intelligence and information theory.



Bin Gao received the B.S. degree in mathematics from Shandong University, Jinan, China, in 2001. He is currently a Ph.D. candidate at the School of Mathematical Sciences, Peking University, Beijing, China.

His research interests are in the areas of pattern recognition, machine learning, data mining, graph theory and corresponding applications on text categorization and image processing.



Yang Wang received the M.S. degree from the School of Mathematical Sciences, Peking University in 2004. He is now working in an insurance company.

His research interests are in the areas of statistical learning, pattern recognition and statistical forecasting.

Qiangsheng Cheng received the B.S. degree in mathematics from Peking University, Beijing, China, in 1963. He is a Professor at the Department of Information Science, School of Mathematical Sciences, Peking University. He was the Vice Director of the Institute of Mathematics, Peking University, from 1988 to 1996.

His current research interests include signal processing, time series analysis, system identification and pattern recognition. Prof. Cheng serves as the vice chairman of Chinese Signal Processing Society and has won the Chinese National Natural Science Award.