

ANALYSIS OF GENE EXPRESSION DATA USING RPEM ALGORITHM IN NORMAL MIXTURE MODEL WITH DYNAMIC ADJUSTMENT OF LEARNING RATE

XING-MING ZHAO

*Institute of Systems Biology
Shanghai University, Shanghai 200444, P. R. China
xm_zhao@shu.edu.cn*

YIU-MING CHEUNG*

*Department of Computer Science
Hong Kong Baptist University, Hong Kong, P. R. China
ymc@comp.hkbu.edu.hk*

DE-SHUANG HUANG

*Intelligent Computing Lab, Institute of Intelligent Machines
Chinese Academy of Sciences, P. O. Box 1130
Hefei, Anhui 230031, P. R. China*

Microarray technology is a useful tool for monitoring the expression levels of thousands of genes simultaneously. Recently, mixture modeling has been used to extract expression signatures from gene expression profiles. In general, two separate steps are utilized to estimate the number of classes and model parameters, respectively. However, such a method is often time-consuming and leads to suboptimal solutions. In this paper, we therefore apply a one-step approach, namely Rival Penalized Expectation-Maximization (RPEM) algorithm, to analyze the gene expression data. The RPEM algorithm is capable of estimating the parameters of normal mixture model, while determining the number of classes automatically at the same time. Furthermore, we speed up the learning procedure of RPEM by proposing a new mechanism to adjust the learning rate dynamically. The numerical results on real gene expression data demonstrate that our proposed method is indeed effective and efficient.

Keywords: Clustering; dynamic adjustment of learning rate; gene expression; normal mixture model; rival penalized EM algorithm.

1. Introduction

DNA microarray technology is a powerful and efficient tool for studying the expression levels of thousands of genes under various conditions simultaneously. An experiment with a single DNA chip can provide researchers with information on

*Author for correspondence

thousands of genes simultaneously. Using gene expression profiles, we can identify individual genes that perform different biological functions and find differences in distinct cell states or types. By comparing the expression patterns of unknown genes to those of annotated genes, one can predict the functions of unknown genes.^{24,25} Recently, the results obtained by DNA microarray technology have been applied to understanding the subtypes of cancers,¹⁵ detecting tumors,⁹ and so forth.

Nowadays, a large amount of gene expression data has been accumulated, whereas a key problem is to extract expression signatures from these expression profiles. One commonly used method is to group genes with similar expression levels provided that genes with similar expression profiles generally share the same functions. In the literature, a number of clustering methods have been applied to microarray data analysis, including hierarchical clustering,¹¹ k -means,²¹ self organizing maps (SOM),²⁰ and so on. However, most of these methods are heuristically motivated. In particular, the issues of determining the “correct” number of clusters and choosing a good clustering algorithm have not been rigorously handled yet.²³

Recently, model based clustering methods provide an alternative to heuristic-based algorithms.^{1,14,19} In the model based algorithm, the data are assumed to be generated by an underlying finite mixture model,^{16,22} e.g. the typical Gaussian (i.e. normal) mixture model. In general, the model-based method needs to determine the number of classes (also called model selection hereinafter) and estimate the model parameters. In literature, the Expectation-Maximization (EM) algorithm⁸ is usually used to estimate the parameters of the normal mixture model, which however needs to preassign class number. The EM algorithm almost always leads to a poor result if the class number is not appropriately preassigned. Under the circumstances, the approaches with two separate steps are often used in the field of bioinformatics.^{17,23} For example, MCLUST is one model based clustering method developed by Fraley and Raftery,¹² and further used by Yeung *et al.*²³ It first performs model selection by the Bayesian Information Criterion (BIC) scores, where the model with the highest BIC score is selected. After determining the class number, MCLUST utilizes the EM algorithm to estimate the parameters of the model. Finally, the estimated model is used to perform microarray data analysis. Although such a method can work as reported in Ref. 23, it may be time-consuming and lead to a suboptimal solution because model selection and model parameters are determined in two separate steps.

In literature, some one-step approaches have been proposed for model-based clustering methods, such as Gaussian mixtures based on variational Bayes approach⁷ and genetic-based Expectation-Maximization (GA-EM) algorithm for learning Gaussian mixture models.¹⁸ In the one-step methods, model selection and parameter estimation are performed simultaneously. Recently, the second author of this paper has also proposed a new one-step approach, namely Rival Penalized Expectation-Maximization (RPEM) algorithm,^{3,4} which can determine the number of classes automatically by fading out the redundant components from the mixture during the parameter learning process. That is, RPEM can perform model selection and parameter learning together in a single paradigm, but not two steps. Consequently,

RPEM saves the computing time and leads to a better result to a certain degree. In RPEM, the performance of the algorithm depends on the choice of learning rate. Generally, there is a tradeoff between the residue deviation and rate of convergence.⁵ When using a fixed learning rate, it should be small enough for the algorithm to converge. The smaller the learning rate, the smaller the residue deviation, but the slower the convergence speed. It is usually difficult to determine an optimal learning rate in advance because it is problem dependent. Under the circumstances, we propose a new method for adjusting learning rates dynamically in the learning procedure. We denote the RPEM algorithm with the dynamic adjustment of learning rate as RPEM-DLR algorithm. The experimental results have shown the superiority of the RPEM-DLR algorithm to the EM and MCLUST on gene expression data analysis.

2. Normal Mixture Models for Gene Expression Data Analysis

Microarray is a 2D array, typically on a glass, filter, or silicon wafer, upon which genes or gene fragments are deposited or synthesized in a predetermined spatial order allowing them to be made available as probes in a high-throughput, parallel manner. Probes with known identity are used to determine complementary binding, thus allowing massively parallel gene expression and gene discovery studies. An experiment with a single DNA chip can provide researchers with information on thousands of genes simultaneously. Due to the large amount of data, there is a need to develop methods for studying the expressed genes and finding expression patterns. The recent studies have shown that model-based clustering methods in a mixture can successfully perform this task in microarray data analysis, e.g. see Ref. 17.

The mixture model assumes that each group of gene expression profiles is generated by an underlying probability distribution. Suppose the number of classes is k , and the number of samples is N . The likelihood function for a mixture model can be defined as:

$$l(\mathbf{x}; \Theta) = \int \ln p(\mathbf{x}|\Theta) dF(\mathbf{x}) \tag{1}$$

with

$$p(\mathbf{x}|\Theta) = \sum_{j=1}^k \alpha_j p(\mathbf{x}|\theta_j), \tag{2}$$

and

$$\sum_{j=1}^k \alpha_j = 1, \quad \forall 1 \leq j \leq k, \quad \alpha_j > 0, \tag{3}$$

where $\Theta = \{\alpha_j, \theta_j\}_{j=1}^k$ is the set of model parameters, $F(\mathbf{x})$ is the cumulative probability function of \mathbf{x} , $p(\mathbf{x}|\theta_j)$ is a multivariate probability density function (pdf) of the gene expression data \mathbf{x} , and α_j is the proportion of \mathbf{x} that comes from Class j . If the probability density function is Gaussian density function, $p(\mathbf{x}|\theta_j)$ will

become $G(\mathbf{x}|\mu_j, \Sigma_j)$, i.e.

$$p(\mathbf{x}|\theta_j) = \frac{\exp\{-\frac{1}{2}(\mathbf{x} - \mu_j)^T \Sigma_j^{-1}(\mathbf{x} - \mu_j)\}}{(2\pi)^{d/2} |\Sigma_j|^{1/2}},$$

where d is the dimension of \mathbf{x} , μ_j and Σ_j are the mean and covariance matrix for the gene expression profiles from the j th class, respectively. In general, the maximum-likelihood (ML) estimation of Θ can be obtained towards maximizing:

$$L(\mathbf{x}; \Theta) = \sum_{i=1}^N \sum_{j=1}^k h(j|\mathbf{x}_i, \Theta) \ln(\alpha_j p(\mathbf{x}_i|\theta_j)), \tag{4}$$

via the EM algorithm. That is, given the number of classes k , an ML solution of Θ can be obtained via the following iterative E-step and M-step:

- E-step:
Fixing $\Theta^{(old)}$, for each \mathbf{x}_i , calculate the posterior probability density function of class j :

$$h(j|\mathbf{x}_i, \theta_j^{(old)}) = \frac{\alpha_j^{(old)} p(\mathbf{x}_i|\theta_j^{(old)})}{\sum_{r=1}^k \alpha_r^{(old)} p(\mathbf{x}_i|\theta_r^{(old)})}.$$

- M-step:

$$\alpha_j^{(new)} = \frac{1}{N} \sum_{i=1}^N h(j|\mathbf{x}_i, \theta_j^{(old)}), \tag{5}$$

$$\mu_j^{(new)} = \frac{\sum_{i=1}^N \mathbf{x}_i h(j|\mathbf{x}_i, \theta_j^{(old)})}{\sum_{i=1}^N h(j|\mathbf{x}_i, \theta_j^{(old)})}, \tag{6}$$

$$\Sigma_j^{(new)} = \frac{\sum_{i=1}^N h(j|\mathbf{x}_i, \theta_j^{(old)}) (\mathbf{x}_i - \mu_j^{(old)}) (\mathbf{x}_i - \mu_j^{(old)})^T}{\sum_{i=1}^N h(j|\mathbf{x}_i, \theta_j^{(old)})}. \tag{7}$$

After estimating the parameters, the mixture models can be applied to microarray data analysis such as classifying the cancer classes, clustering microarray expression data, and so on. The success of this method depends on the parameter estimation using the EM algorithm. Usually, the EM algorithm can successfully estimate the parameters of the mixture model as long as the number of classes is correctly assigned. Otherwise, the EM algorithm may not work well. Unfortunately, from the practical viewpoint, it is hard to know the true number of classes in advance.

3. The RPEM-DLR Algorithm with Dynamic Adjustment of Learning Rate

Due to the limitations of the EM algorithm, Cheung^{3,4} has proposed an RPEM algorithm, which can determine the number of classes automatically while estimating

the parameters of the mixture model. In the RPEM algorithm, Eq. (1) can be further represented in a weighted form, i.e.

$$l(\mathbf{x}; \Theta) = \int \sum_{j=1}^k g(j|\mathbf{x}, \Theta) \ln p(\mathbf{x}|\Theta) dF(\mathbf{x}) \tag{8}$$

with

$$\sum_{j=1}^k g(j|\mathbf{x}, \Theta) = 1, \tag{9}$$

where $g(j|\mathbf{x}, \Theta)$ s are the designable weights, and $g(j|\mathbf{x}, \Theta) = 0$ if $h(j|\mathbf{x}, \Theta) = 0$ for some j .

The papers^{3,4} have given the design of a class of $g(j|\mathbf{x}, \Theta)$ s, and in particular thoroughly studied the case of

$$g(j|\mathbf{x}_t, \Theta) = 2\varphi(j|\mathbf{x}_t, \Theta) - h(j|\mathbf{x}_t, \Theta) \tag{10}$$

with

$$\varphi(j|\mathbf{x}_t, \Theta) = \begin{cases} 1, & \text{if } j = c = \arg \max_{1 \leq r \leq k} h(r|\mathbf{x}_t, \Theta) \\ 0, & \text{otherwise} \end{cases}.$$

It can be seen that, as N is large enough, the empirical form of Eq. (8) becomes

$$\begin{aligned} Q(X_N; \Theta) &= \frac{1}{N} \sum_{t=1}^N \sum_{j=1}^k [2\varphi(j|\mathbf{x}_t, \Theta) - h(j|\mathbf{x}_t, \Theta)] \ln[\alpha_j p(\mathbf{x}_t|\theta_j)] \\ &\quad - \frac{1}{N} \sum_{t=1}^N \sum_{j=1}^k [2\varphi(j|\mathbf{x}_t, \Theta) - h(j|\mathbf{x}_t, \Theta)] \ln h(j|\mathbf{x}_t, \Theta). \end{aligned} \tag{11}$$

Subsequently, we can estimate Θ via maximizing Eq. (11). The details can be found in Refs. 3 and 4.

The RPEM algorithm in Gaussian density mixture can be summarized as follows:

- **Initialization:** Given a specific k with $k \geq k^*$, where k^* is the true class number, we initialize Θ . Then, at each time step t , we implement the following two steps:
- **Step 1:** Fixing $\Theta^{(old)}$, we calculate

$$h(j|\mathbf{x}_t, \Theta^{(old)}) = \frac{\alpha_j^{(old)} G(\mathbf{x}_t|\mu_j^{(old)}, \Sigma_j^{(old)})}{p(\mathbf{x}_t|\Theta^{(old)})} \tag{12}$$

and

$$g(j|\mathbf{x}_t, \Theta^{(old)}) = 2\varphi(j|\mathbf{x}_t, \Theta^{(old)}) - h(j|\mathbf{x}_t, \Theta^{(old)}), \quad 1 \leq j \leq k$$

with

$$\alpha_j^{(\text{old})} = \frac{\exp(\beta_j^{(\text{old})})}{\sum_{j=1}^k \exp(\beta_j^{(\text{old})})}, \tag{13}$$

where we learn the free variables β_j s rather than α_j s to circumvent the complicated constrained optimization.

- **Step 2:** Fixing $h(j|\mathbf{x}_t, \Theta^{(\text{old})})$ s, we update Θ using gradient ascent method, i.e.

$$\beta_j^{(\text{new})} = \beta_j^{(\text{old})} + \eta_1 [g(j|\mathbf{x}_t, \Theta^{(\text{old})}) - \alpha_j^{(\text{old})}], \tag{14}$$

$$\mu_j^{(\text{new})} = \mu_j^{(\text{old})} + \eta_2 g(j|\mathbf{x}_t, \Theta^{(\text{old})}) \Sigma_j^{-1(\text{old})} (\mathbf{x}_t - \mu_j^{(\text{old})}), \tag{15}$$

$$\begin{aligned} \Sigma_j^{-1(\text{new})} &= [1 + \eta_2 g(j|\mathbf{x}_t, \Theta^{(\text{old})})] \Sigma_j^{-1(\text{old})} \\ &\quad - \eta_2 g(j|\mathbf{x}_t, \Theta^{(\text{old})}) \mathbf{U}_{t,j}, \end{aligned}$$

where $\mathbf{U}_{t,j} = \left[\Sigma_j^{-1(\text{old})} (\mathbf{x}_t - \mu_j^{(\text{old})}) (\mathbf{x}_t - \mu_j^{(\text{old})})^T \Sigma_j^{-1(\text{old})} \right]$, η_1 and η_2 are small positive learning rates with $\eta_1 \ll \eta_2 \leq 1$. The procedure is repeated until Θ converges.

In Eqs. (14) and (15), it can be seen that the learning rate is generally a fixed small positive constant. Actually, the choice of the learning rate can affect the performance of RPEM. In general, there is a tradeoff between the residue deviation and rate of convergence.⁵ When using a fixed learning rate, it should be small enough for the algorithm to converge. The smaller the learning rate, the smaller the residue deviation, but the slower the convergence speed. It is usually difficult to determine an optimal learning rate in advance because it is problem dependent.

It can be seen from **Step 2** that, if the learning rate tends to zeros as the epoch tends to infinity, the convergence of the parameter set $\Theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \beta\}$ will be guaranteed. Hence, we want to design a class of learning rate which will change dynamically in the learning procedure. If the number of data points belonging to certain class j is too large, i.e. α_j is large, the learning rate should decrease and keep those seed points corresponding to larger α_j s from dominating the learning procedure. On the other hand, if α_j is small, the learning rate should increase so that the seed points corresponding to the true ones can get closer to the target cluster center, while those redundant seed points can be pushed away from the cluster center quickly.

According to the condition for the asymptotic convergence provided by a standard theorem¹⁰ from stochastic approximation theory, the learning rate should satisfy:

$$\lim_{it \rightarrow \infty} \eta(it) = 0, \quad \text{and} \quad \sum_{it=1}^{\infty} \eta(it) = \infty, \tag{16}$$

where it is the it th epoch. Under the circumstances, we propose a new method here for adjusting the learning rate dynamically. The learning rate is defined as:

$$\begin{aligned} \eta(j, it) &= \eta(j, it - 1) * \frac{1 - \alpha_j}{1 + \alpha_j}, \\ 1 \leq j \leq k, \quad \text{and} \quad 1 \leq it < \infty, \end{aligned}$$

where $\eta(j, it)$ is the learning rate for the j th component in the it th epoch, and α_j is the proportion of \mathbf{x} that comes from Class j . The initial learning rate η_0 is set at a fixed small positive constant. Therefore, the learning rate will be adjusted dynamically according to α_j in each epoch.

4. Experimental Results and Discussions

To compare the performance of different clustering algorithms, this section utilized the two sets of data, namely fluorescence signal intensity data and yeast cell cycle data, in which the external evaluation criteria were available.

4.1. Fluorescence signal intensity data

In this subsection, the RPEM-DLR algorithm was utilized to discriminate the reliable expression data from the unreliable expression data. In this paper, the two data sets presented by Asyali² were used, which consist of about 2000 cDNA distinct probes and a total of about 4000 elements.¹³ The details of the preparation of the gene expression data can be found in Ref. 2. The data consist of Cy3 and Cy5 channel fluorescence signal intensities. After ground-substraction and normalization, both channels are naturally log-transformed, which makes the distribution of the data closer to normality. Table 1 summarizes the statistical features of the two data sets, where n is the number of samples, SD is the standard deviation, and $\rho_{Cy3,Cy5}$ are the correlations between channels Cy3 and Cy5.²

The initial learning rate η_0 was set at 0.02 and 0.008 for Dataset 1 and Dataset 2, respectively. We first set the number of seed points k at 2, i.e. the true mixture number. The performance of RPEM-DLR algorithm was compared to that of EM and Fuzzy C-means (FCM), where the number of classes for FCM was set at the true number of classes, i.e. 2, through the following experiments. The experimental results are summarized in Fig. 1, where the data points marked as “+” are those classified as reliable by FCM while the ones marked as “o” are those classified as unreliable by FCM, the data points marked by “×” are the learned class centers, and the ellipses were formed by the learned covariance matrices via EM and RPEM-DLR, respectively. It can be seen from Fig. 1 that both EM and RPEM-DLR can successfully locate the cluster centers when the preassigned class number is the true one.

Table 1. Statistics for the two sets of gene expression data.

	Dataset 1		Dataset 2	
	Cy3	Cy5	Cy3	Cy5
n	3080		6498	
Mean	6.28	6.16	6.24	5.90
Median	5.99	5.81	5.97	5.75
SD	1.08	1.16	1.15	1.56
$\rho_{Cy3,Cy5}$	0.93		0.87	

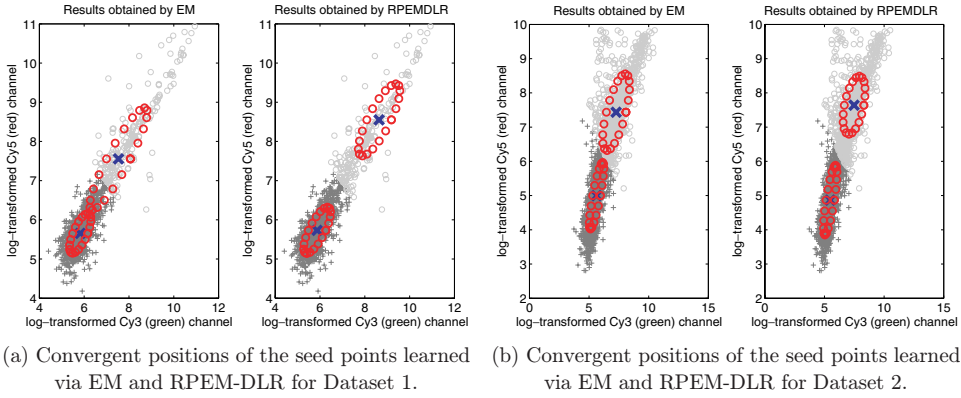


Fig. 1. The convergent positions of seed points, where the number of seed points was the true class number, i.e. 2.

Table 2. The convergent model parameters for the normal mixture model via EM, FCM and RPEM-DLR, respectively, where the number of classes is set at 2.

Dataset	Methods	Component	Mean	Weight	
1	EM	Component 1	5.83	5.66	0.74
		Component 2	7.53	7.56	0.26
	RPEM-DLR	Component 1	5.82	5.67	0.82
		Component 2	8.28	8.34	0.18
	FCM	Component 1	5.83	5.66	0.82
		Component 2	8.14	8.17	0.18
2	EM	Component 1	5.63	4.99	0.63
		Component 2	7.27	7.43	0.37
	RPEM-DLR	Component 1	5.49	4.90	0.60
		Component 2	7.52	7.64	0.40
	FCM	Component 1	5.47	4.77	0.59
		Component 2	7.35	7.48	0.41

In addition, Table 2 summarizes the convergent class centers and weights, i.e. μ_{jS} and α_{jS} , obtained by the EM, FCM and RPEM-DLR for the normal mixture models of the two datasets, where the number of classes is exactly the true class number. It can be seen from Table 2 that both the EM and RPEM-DLR can work when they are assigned the true class number. It can also be seen from Table 2 that the results obtained by the RPEM-DLR algorithm are closer to the FCM rather than EM.

Furthermore, we investigated the performance of RPEM-DLR when the number of seed points is much larger than the true one. We set the number of seed points at 5, i.e. $k = 5$. The experimental results are summarized in Fig. 2. It can be seen that the RPEM-DLR algorithm can significantly locate the class centers by fading out the extra seed points without knowing the number of classes in advance, while EM cannot.

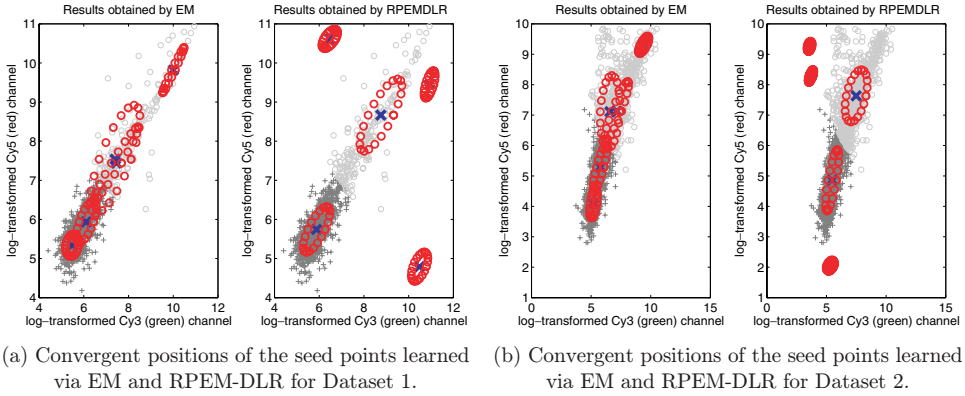


Fig. 2. The convergent positions of seed points, where the number of seed points was set at 5.

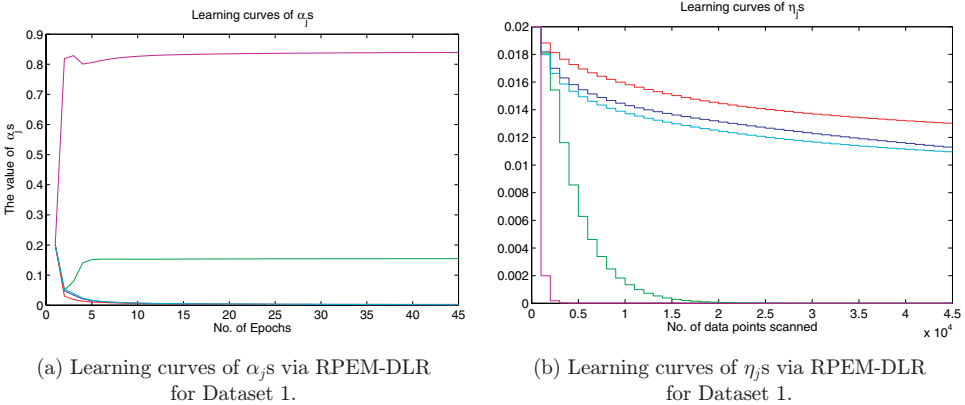


Fig. 3. Learning curves of $\alpha_{j,s}$ and $\eta_{j,s}$ for Dataset 1, where the number of seed points was set at 5.

In addition, we investigated the values of $\alpha_{j,s}$ and $\eta_{j,s}$ learned via RPEM-DLR for the two sets of data. As shown in Figs. 3 and 4, all of those $\alpha_{j,s}$ corresponding to the extra seed points have been approached to zeros after tens of epochs. Further, Table 3 shows $\alpha_{j,s}$ ' values learned via EM and RPEM-DLR for the two datasets. It can be seen that the weights for the extra seed points in RPEM-DLR tend to zeros but those in EM not. That is, the RPEM-DLR algorithm is superior to EM on gene expression data analysis without knowing the number of classes in advance. Such a capability is much useful in gene expression data analysis under normal mixture modeling because it is generally hard to know the correct class number in advance.

4.2. Yeast cell cycle data

The yeast cell cycle dataset⁶ was used to evaluate the performance of the RPEM-DLR algorithm. The yeast cell cycle data demonstrate the fluctuation of expression

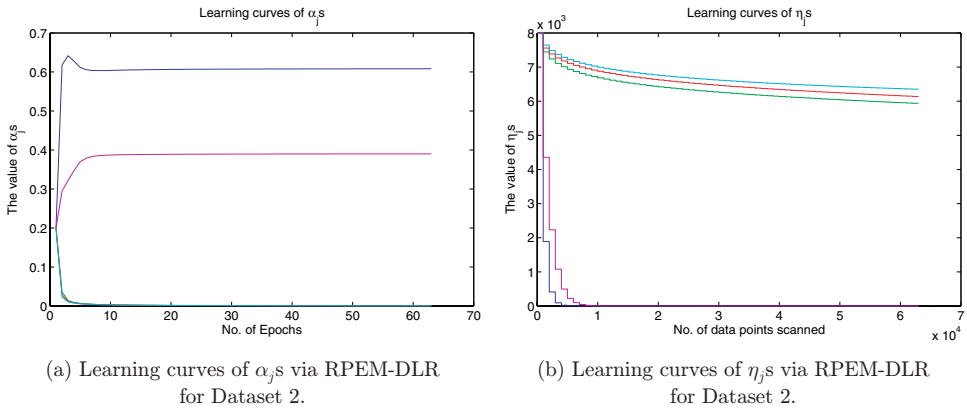


Fig. 4. Learning curves of α_j,s and η_j,s for Dataset 2, where the number of seed points was set at 5.

Table 3. The weights of the four components learned via EM and RPEM-DLR for the two datasets, where the number of classes is set at 5.

	Dataset 1		Dataset 2	
	EM	RPEM-DLR	EM	RPEM-DLR
Component 1	0.022	0.840	0.340	0.608
Component 2	0.131	0.155	0.216	0.390
Component 3	0.445	0.003	0.137	0.00006
Component 4	0.321	0.002	0.282	0.00007
Component 5	0.080	0.001	0.025	0.00005

levels of approximately 6000 genes over two cell cycles, with 17 time points for each gene taken at ten minute intervals. The raw expression profiles can be downloaded from <http://genomics.stanford.edu>. The dataset used here was the same one used by Yeung *et al.*²³ This subset consists of 384 genes whose expression levels peak at different time points corresponding to the five phases of cell cycles (the five-phases criterion), and is available at <http://www.cs.washington.edu/homes/kayee/model>. In literature, it has been known that this set of data can be classified into five classes.²³ Hence, we used them here to evaluate the effectiveness of the RPEM-DLR algorithm.

We compared the RPEM-DLR algorithm with EM and MCLUST, respectively, using the yeast cell cycle dataset. The initial learning rate is set at 0.01. We first set the number of classes $k = 5$, i.e. the true class number. For the EM algorithm, it was directly used to estimate the parameters of the normal mixture model with five components, where the full covariance matrices were adopted. For MCLUST, it first selected the best model by the BIC scores, and then estimated the model parameters using the EM algorithm. In comparison, the RPEM-DLR algorithm selects the model and estimates the model parameters simultaneously without additional computation of BIC scores. Hence, it saves the computing cost. Table 4 shows the results obtained by MCLUST, EM and RPEM-DLR, respectively. It can be seen that both

Table 4. Performance of the MCLUST software, EM and RPEM-DLR algorithms with the correct number of classes preassigned, where α_j is the proportion of observations coming from the j th class.

Cell Division Phase	Methods	α_j	Precision
Early G1 (67 genes)	RPEM-DLR	0.18	82%
	MCLUST	0.19	75%
	EM	0.19	75%
Late G1 (135 genes)	RPEM-DLR	0.40	78%
	MCLUST	0.40	83%
	EM	0.40	83%
S (75 genes)	RPEM-DLR	0.15	68%
	MCLUST	0.10	36%
	EM	0.10	36%
G2 (52 genes)	RPEM-DLR	0.14	62%
	MCLUST	0.12	67%
	EM	0.12	67%
M (55 genes)	RPEM-DLR	0.12	89%
	MCLUST	0.19	95%
	EM	0.19	95%
Overall	RPEM-DLR	—	76%
	MCLUST	—	71%
	EM	—	71%

RPEM-DLR and MCLUST work well when the number of classes k is correctly assigned to the true value, and the performance of RPEM-DLR is a little better than that of MCLUST. Please note that EM is equal to MCLUST in this case.

Then, we suppose the true number of classes is unknown, which is actually true from a practical viewpoint. Under the circumstances, we set the number of classes k at 10 but not 5. For the EM algorithm, the full covariance matrices were adopted. The results obtained by RPEM-DLR, MCLUST and EM are shown in Figs. 5–7, where the number in the bracket represents the number of genes classified correctly. Further, Table 5 shows the comparison of the performances of EM, RPEM-DLR and MCLUST. It can be seen that both the RPEM-DLR algorithm and MCLUST can assign most of the genes to the right classes. The performance of RPEM-DLR algorithm is a little better than that of MCLUST because the RPEM-DLR algorithm determines class number and estimates model parameters in one step, while MCLUST performs model selection and estimates model parameters in two separate steps. For the EM algorithm itself, the performance was the worst with the overall accuracy of 51%.

In addition, we investigated the values of α_j s and η_j s learned via RPEM-DLR. As shown in Fig. 8(a), all of those α_j s corresponding to the extra seed points approached zero after 150 epoches. It can be seen from the above experiments that the RPEM-DLR algorithm can estimate the model parameters and determine the class number automatically without external criteria as long as the preassigned class number is

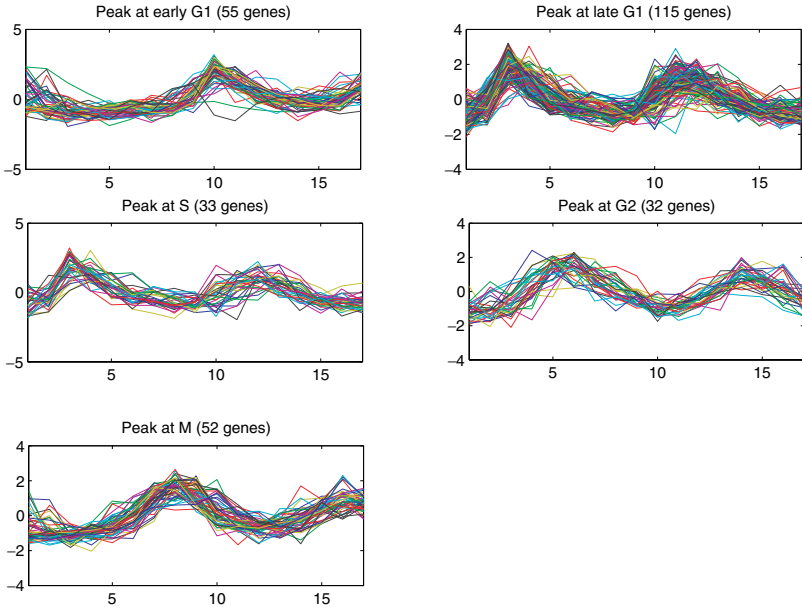


Fig. 5. The genes that were correctly assigned to the five classes by the RPEM-DLR algorithm, where the number of seed points was set at 10.

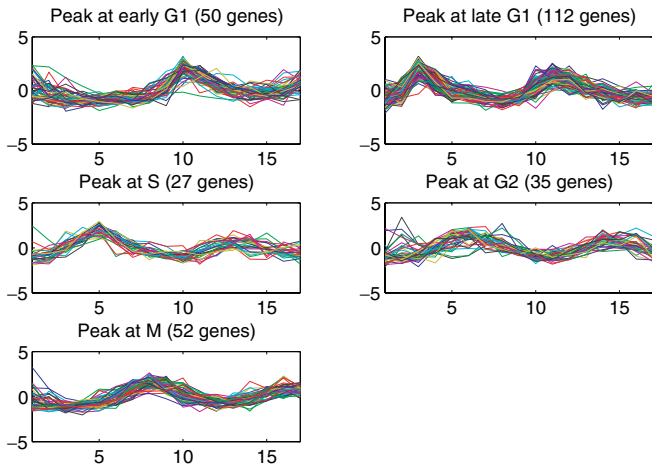


Fig. 6. The genes that were correctly assigned to the five classes by the MCLUST software, where the number of seed points was set at 10.

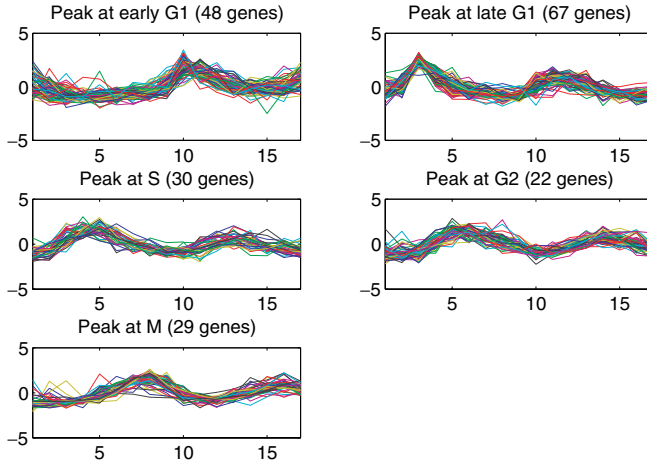


Fig. 7. The genes that were correctly assigned to the five classes by the EM algorithm, where the number of seed points was set at 10.

Table 5. Performance of the EM algorithm, MCLUST software and RPEM-DLR with the number of classes set at 10, where α_j is the proportion of observations coming from the j th class.

Cell Division Phase	Methods	α_j	Precision
Early G1 (67 genes)	RPEM-DLR	0.16	82%
	MCLUST	0.19	75%
	EM	0.09	72%
Late G1 (135 genes)	RPEM-DLR	0.37	85%
	MCLUST	0.40	83%
	EM	0.32	50%
S (75 genes)	RPEM-DLR	0.16	44%
	MCLUST	0.10	36%
	EM	0.09	40%
G2 (52 genes)	RPEM-DLR	0.15	62%
	MCLUST	0.12	67%
	EM	0.12	42%
M (55 genes)	RPEM-DLR	0.13	95%
	MCLUST	0.19	95%
	EM	0.19	53%
Overall	RPEM-DLR	—	74%
	MCLUST	—	71%
	EM	—	51%

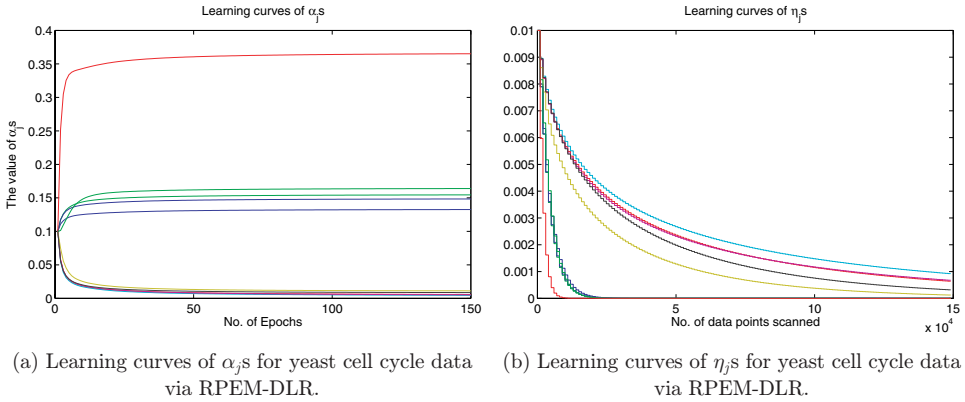


Fig. 8. Learning curves of α_j,s and η_j,s via RPEM-DLR for yeast cell cycle data, where the number of seed points was set at 10.

larger than the true one, which demonstrates the effectiveness and efficiency of our method.

5. Conclusion

High-throughput gene expression data are rich sources to investigate gene functions and biological process in which genes are involved. Although there are a number of clustering methods that have been developed to analyze these data, there is still room for improvement. In this work, an RPEM algorithm with dynamic adjustment of learning rate, namely RPEM-DLR algorithm, has been proposed. Compared with the original RPEM algorithm, RPEM-DLR can adjust learning rate dynamically in the learning procedure, which therefore reduces computation cost and leads to better performance in some cases. Furthermore, the RPEM-DLR algorithm has been utilized to analyze the gene expression data, where the gene expression data are supposed to be generated from a mixture of unknown number of classes. It is shown that the RPEM-DLR is capable of determining the number of classes and the model parameters in a single step, but not two steps like MCLUST. The numerical results have demonstrated the effectiveness of the RPEM-DLR on the gene expression data analysis in comparison with the EM and the MCLUST.

Acknowledgments

This work was supported by the Research Grant Council of the Hong Kong SAR under Project Code HKBU 2103/06E and HKBU 210309, the Faculty Research Grant of Hong Kong Baptist University under Project: FRG2/08-09/122, Shanghai Rising-Star Program (10QA1402700), Innovation Program of Shanghai Municipal Education Commission (10YZ01), and Innovation Funding of Shanghai University.

References

1. R. Alexandridis, S. Lin and M. Irwin, Class discovery and classification of tumor samples using mixture modeling of gene expression data — a unified approach, *Bioinformatics* **20** (2004) 2545–2552.
2. M. H. Asyali and M. Alci, Reliability analysis of microarray data using fuzzy c-means and normal mixture modeling based classification methods, *Bioinformatics* **21**(5) (2005) 644–649.
3. Y. M. Cheung, A rival penalized em algorithm towards maximizing weighted likelihood for density mixture clustering with automatic model selection, *Proc. 17th Int. Conf. Pattern Recognition (ICPR'04)* (Cambridge, United Kingdom, 2004), pp. 633–636.
4. Y. M. Cheung, Maximum weighted likelihood via rival penalized em for density mixture clustering with automatic model selection, *IEEE Trans. Knowl. Data Engin.* **17**(6) (2005) 750–761.
5. C. Chinrungrueng and C. H. Sequin, Optimal adaptive k-means algorithm with dynamic adjustment of learning rate, *IEEE Trans. Neural Networks* **6**(1) (1995) 157–169.
6. R. J. Cho, M. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, D. J. Lockhart and R. W. Davis, A genome-wide transcriptional analysis of the mitotic cell cycle, *Molecular Cell* **2** (1998) 65–73.
7. C. Constantinopoulos and A. Likas, Unsupervised learning of Gaussian mixtures based on variational component splitting, *IEEE Trans. Neural Networks* **18** (2007) 745–755.
8. A. Dempster, N. Laird and D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. Roy. Stat. Soc.* **B39** (1977) 1–38.
9. S. Dudoit, J. Fridlyand and T. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, *J. Am. Stat. Assoc.* **97** (2002) 77–87.
10. A. Dvoretzky, Stochastic approximation revisited, *Adv. Appl. Math.* **7** (1986) 220–227.
11. M. B. Eisen, P. T. Spellman, P. O. Brown and D. Botstein, Cluster analysis and display of genome-wide expression patterns, *Proc. Natl. Acad. Sci. USA* **95** (1998) 14863–14868.
12. C. Fraley and A. E. Raftery, Enhanced software for model-based clustering, discriminant analysis, and density estimation: MCLUST, *J. Classif.* **20** (2003) 263–286.
13. M. A. Frevel, T. Bakheet, A. M. Silva, J. G. Hissong, K. S. Khabar and B. R. Williams, p38 mitogen-activated protein kinase-dependent and -independent signaling of mrna stability of au-rich element-containing transcripts, *Mole. Cell. Biol.* **23** (2003) 425–436.
14. D. Ghosh and A. M. Chinnaiyan, Mixture modeling of gene expression data from microarray experiments, *Bioinformatics* **18** (2002) 275–286.
15. T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield and E. Lander, Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, *Science* **286** (1999) 531–537.
16. G. McLachlan and K. Basford, *Mixture Models: Inference and Applications to Clustering* (Marcel Dekker Inc., Monticello, New York, 1987).
17. G. J. McLachlan, R. W. Bean and D. Peel, A mixture model-based approach to the clustering of microarray expression data, *Bioinformatics* **18** (2002) 413–422.
18. F. Pernkopf and D. Bouchaffra, Genetic-based EM algorithm for learning Gaussian mixture models, *IEEE Trans. Patt. Anal. Mach. Intell.* **27** (2005) 1344–1348.
19. Y. Qu and S. Xu, Supervised cluster analysis for microarray data based on multivariate Gaussian mixture, *Bioinformatics* **20** (2004) 1905–1913.
20. P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander and T. R. Golub, Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation, *Proc. Natl. Acad. Sci. USA* **96** (1999) 2907–2912.

21. S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho and G. M. Church, Systematic determination of genetic network architecture, *Nature Genet.* **2** (1999) 281–285.
 22. J. Wolfe, Pattern clustering by multivariate mixture analysis, *Multivar. Behav. Res.* **5** (1970) 329–350.
 23. K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery and W. L. Ruzzo, Model-based clustering and data transformations for gene expression data, *Bioinformatics* **17** (2001) 977–987.
 24. X.-M. Zhao, Y. Wang, L. Chen and K. Aihara, Gene function prediction using labeled and unlabeled data, *BMC Bioinformatics* **9** (2008) 57.
 25. X.-M. Zhao, L. Chen and K. Aihara, Protein function prediction with high-throughput data, *Amino Acids* **35** (2008) 517–530.
-



King-Ming Zhao received his B.E. and M.E. degrees from Jilin University, China, in 2000 and 2003, respectively. He received his Ph.D. degree from the University of Science and Technology of China, China, in 2005. From May 2006 to May 2008, he is a researcher in

ERATO Aihara Complexity Modelling Project, JST. He is currently an associate professor at the Institute of Systems Biology, Shanghai University, Shanghai, China.



De-Shuang Huang received the B.Sc., M.Sc. and Ph.D. degrees all in electronic engineering from the Institute of Electronic Engineering, Hefei, China, National Defense University of Science and Technology, Changsha, China and Xidian University, Xian,

China, in 1986, 1989 and 1993, respectively. From September 2000, he joined the Hefei Institute of Intelligent Machines, CAS as a Recipient of Hundred Talents Program of CAS.



Yiu-Ming Cheung received his Ph.D. degree from the Department of Computer Science and Engineering at the Chinese University of Hong Kong in 2000. He joined the Department of Computer Science at Hong Kong Baptist University in 2001, and then became an

Associate Professor in 2005. He is a senior member of IEEE and ACM.