Iterative Feature Selection in Gaussian Mixture Clustering with Automatic Model Selection

Hong Zeng and Yiu-ming Cheung

***Abstract*— This paper proposes an algorithm to deal with the feature selection in Gaussian mixture clustering by an iterative way: the algorithm iterates between the clustering and the unsupervised feature selection. First, we propose a quantitative measurement of the feature relevance with respect to the clustering. Then, we design the corresponding feature selection scheme and integrate it into the Rival Penalized EM (RPEM) clustering algorithm (Cheung 2005) that is able to determine the number of clusters automatically. Subsequently, the clustering can be performed in an appropriate feature subset by gradually eliminating the irrelevant features with automatic model selection. Compared to the existing methods, the numerical experiments have shown the efficacy of the proposed algorithm on the synthetic and real world data.**

I. Introduction

Gaussian mixture (GM) clustering has been widely applied to a variety of fields including data mining, time series forecasting, image processing, and so forth. In general, GM clustering needs to make the model selection, i.e. determine the number of components in a mixture (also called *model order* interchangeably), and estimate the parameters of each component in a mixture based on the observations. However, from the practical viewpoint, the elements of each observation (also called *features* hereinafter) may not make the same contribution to the data cluster structure at all. That is, there may be some irrelevant features in the observations. Under the circumstances, the inclusion of such irrelevant features in the clustering will not only drastically increase the computational complexity, but also mask the cluster structure due to the *curse of dimensionality*.

In order to perform an appropriate data partition, a refined subset of most informative features is often expected. However, due to the absence of the ground-truth labels that could guide the assessment of the relevance for each feature with respect to the clustering, it is a nontrivial task to conduct the feature selection in the unsupervised learning. The problem becomes even more challenging when the true number of clusters is unknown *a priori*, as the optimal feature subset and the optimal number of clusters are inter-related: different clustering results might be obtained on different feature subsets. This suggests that the feature selection, which is to identify the features that significantly contribute to the grouping, should be taken into account jointly with the clustering.

In the literature, there have been several representative methods that address the issue of the feature selection for the clustering. In the approaches [1], [2], they ignore this interrelationship between the feature selection and the clustering task and typically choose the features prior to a clustering algorithm. Though it may significantly reduce the dimensionality, these selected features may not be necessarily well suited to the mining algorithm [5]. Thus, in order to obtain both optima for the feature subset and the clustering structure, some approaches, e.g. see [3], [4], wrap the feature selection around the clustering algorithm by first conducting a combinatorial search for candidate subsets in the whole feature space, then evaluating these subsets using the clustering algorithm. Subsequently, the best subset is chosen using a certain criterion during the repeated wrapping around. This kind of approaches may suffer from a heavy computational burden with the time-consuming searching strategy and the repeated execution of the clustering algorithms. Recently, the approaches [5], [6] have managed to tackle these two issues in a single optimization paradigm. The preliminary experiments in [5], [6] have shown the promising results. Nevertheless, such a method supposes that the explicit parametric form of irrelevant feature distribution is known *a priori*, which may be impossible from the practical point of view.

In this paper, we propose an algorithm that deals with the feature selection for the clustering in an iterative way: the algorithm iterates between the clustering and the unsupervised feature selection. First, we will present a quantitative measurement of the feature relevance with respect to the clustering. Then, we will design the corresponding feature selection scheme and integrate it into an efficient clustering algorithm, namely the Rival Penalized EM algorithm [13], which is able to determine the number of clusters automatically. Consequently, our proposed clustering method can perform the model and feature selection in a single paradigm, but without knowing the explicit parametric form of the irrelevant feature distribution in advance. The numerical simulations have shown that the proposed algorithm outperforms the existing ones on both of the synthetic and real world data.

The remainder of the paper is organized as follows. Section II overviews the RPEM algorithm. Section III describes the proposed feature selection procedure. Then, Section IV presents the proposed algorithm and Section V shows the experimental results. Finally, we draw a conclusion in Section VI.

Hong Zeng and Yiu-ming Cheung are with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong SAR, China (email: {hzeng, ymc}@comp.hkbu.edu.hk).

II. The Rival Penalized EM Algorithm for the Gaussian Mixture Clustering

Suppose that the observation data set $\mathbf{X_N} = \{\mathbf{x_1}, \mathbf{x_2} \ldots \mathbf{x_N}\}$ is generated from a mixture of k^* Gaussian components, i.e.,

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

with

$$\sum_{j=1}^{k^*} \alpha_j^* = 1 \quad \text{and} \quad \forall 1 \le j \le k^*, \quad \alpha_j^* > 0,$$

where each observation $\mathbf{x_t}(1 \le t \le N)$ is a vector of d-dimensional features: $[x_{1t}, \ldots, x_{dt}]^T$. Furthermore, $p(\mathbf{x}_t|\theta_j^*)$ is the j^{th} Gaussian component with the parameter θ_j^*, α_j^* represents the true mixing coefficient, or the proportion of the j^{th} component in the mixture. The main purpose of clustering analysis is to find an estimate of $\Theta^* = \{\alpha_j^*, \theta_j^*\}_{j=1}^{k^*}$, denoted as $\Theta = \{\alpha_j, \theta_j\}_{j=1}^{k}$, from N observations. A general approach is to search a set of parameters which could reach a maxima of the fitness in terms of *maximum likelihood* (ML) defined below:

$$\hat{\Theta}_{ML} = \arg\max_{\Theta} \{\log p(\mathbf{X_N}|\Theta)\}.$$

The commonly used search strategy is the *Expectation Maximization* (EM) algorithm [7], [8], [9]. However, there is no penalty in the above likelihood, which means the model order k^* cannot be automatically determined and has to be pre-specified. Although some model selection criteria, e.g. see [10], [11], have been proposed in the literature, they may require users to compare the candidate models for a range of orders to determine the optimal one, whose computation is laborious. Recently, an approach called *Rival Penalized EM* (RPEM for short) [13] has been proposed, by which the order is determined simultaneously with the parameter estimation. It is achieved by introducing unequal weights into the conventional maximum likelihood as the regularization terms. The weighted likelihood is written below:

$$\begin{aligned} Q(\Theta, \mathbf{X}_N) &= \frac{1}{N} \sum_{t=1}^{N} \log p(\mathbf{x}_t|\Theta) \\ &= \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{k} g(j|\mathbf{x}_t, \Theta) \log p(\mathbf{x}_t|\Theta) \\ &= \frac{1}{N} \sum_{t=1}^{N} \mathcal{M}(\Theta, \mathbf{x}_t) \end{aligned} \tag{2}$$

$$\begin{aligned} \mathcal{M}(\Theta, \mathbf{x}_t) &= \sum_{j=1}^{k} g(j|\mathbf{x}_t, \Theta) \log[\alpha_j p(\mathbf{x}_t|\theta_j)] \\ &\quad - \sum_{j=1}^{k} g(j|\mathbf{x}_t, \Theta) \log h(j|\mathbf{x}_t, \Theta) \end{aligned} \tag{3}$$

where

$$h(j|\mathbf{x}_t, \Theta) = \frac{\alpha_j p(\mathbf{x}_t|\theta_j)}{p(\mathbf{x}_t|\Theta)}$$

is the posterior probability that $\mathbf{x}_t$ belongs to the j^{th} component in the mixture, and k is greater than or equal to k^*. $g(j|\mathbf{x}_t, \Theta)$'s are designable weight functions, satisfying the constraints below:

$$\sum_{j=1}^{k} g(j|\mathbf{x}_t, \Theta) = \zeta, 1 \le t \le N,$$

and

$$\forall j, g(j|\mathbf{x}_t, \Theta) = 0 \quad \text{if} \quad h(j|\mathbf{x}_t, \Theta) = 0,$$

where ζ is a positive constant. In [13], they are constructed from the following equation:

$$g(j|\mathbf{x}_t, \Theta) = (1 + \varepsilon_t) I(j|\mathbf{x}_t, \Theta) - \varepsilon_t h(j|\mathbf{x}_t, \Theta)$$

with

$$I(j|\mathbf{x}, \Theta) = \begin{cases} 1 & \text{if } j = c \equiv \arg\max_{1 \le i \le k} h(i|\mathbf{x}, \Theta); \\ 0 & j = r \ne c. \end{cases} \tag{4}$$

and ε_t is a small positive quantity. This construction of weight functions reflects the pruning scheme: when a sample $\mathbf{x}_t$ comes from a component that indeed exists in the mixture, the value of $h(j|\mathbf{x}_t, \Theta)$ is likely to be the greatest, thus this component will be the winner. Accordingly, a positive weight $g(c|\mathbf{x}_t, \Theta)$ will keep it in the temporary model. In contrast, all other components fail in the competition and are treated as the 'pseudo-components". As a result, the negative weights are assigned to them as a penalty. Over the learning process of Θ, only the genuine clusters will survive, whereas the "pseudo-clusters" will gradually faded out from the mixture.

The RPEM gives an estimate of Θ^* via maximizing weighted likelihood (MWL) in (2), i.e.,

$$\hat{\Theta}_{MWL} = \arg\max_{\Theta} \{Q(\Theta, \mathbf{X}_N)\}.$$

The more detailed implementation of the RPEM can be found in [13]. In the following, we summarize its major steps in Algorithm 1.

Algorithm 1: The RPEM algorithm.

input : $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, k, η, $epoch_{max}$, *initial* $\mathbf{\Theta}$
output: *The converged* $\hat{\mathbf{\Theta}}$

$epoch\_count \leftarrow 0$;
while $epoch\_count \le epoch_{max}$ **do**
  for $t \leftarrow 1$ **to** N **do**
    Step 1: *Calculate* $h(j|\mathbf{x}_t, \hat{\Theta})$*'s to obtain* $g(j|\mathbf{x}_t, \hat{\Theta})$*'s;*
    Step 2:
    $\hat{\Theta}^{(new)} = \hat{\Theta}^{(old)} + \Delta\Theta = \hat{\Theta}^{(old)} + \eta \left.\frac{\partial \mathcal{M}(\mathbf{x}_t; \hat{\Theta})}{\partial \Theta}\right|_{\hat{\Theta}^{(old)}}$
    where η is a learning rate.
  end
  $epoch\_count \leftarrow epoch\_count + 1$;
end

III. Unsupervised Feature Selection

It is found that a feature should be irrelevant to the data cluster structure if the clusters are indistinguishable each other when the observations are projected onto this feature. To illustrate this scenario, we show an example using a 2-component bivariate Gaussian mixture in Figure 1. If we project the two clusters onto the Y axis, it is unable to distinguish these two clusters by the feature Y, because the observations from the two clusters are almost projected onto the same dense region of this dimension. Hence, the feature

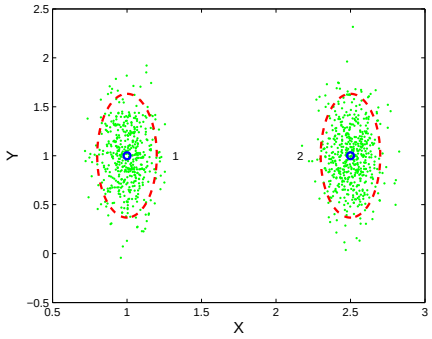


Fig. 1. The feature X is relevant to the partitioning, while the feature Y is irrelevant.

Y will not be helpful in finding the cluster structure, i.e., it is irrelevant for the clustering. On the contrary, the projections onto the X axis can provide the useful information regarding the cluster structure, thus the feature X is relevant for the clustering.

Subsequently, we define the following quantitative measure for the relevance of each feature:

$$SCORE_l = \frac{1}{k}\sum_{j=1}^{k} Score_{l,j} = \frac{1}{k}\sum_{j=1}^{k}(1 - \frac{s_{l,j}^2}{s_l^2})\ l = 1, \ldots, d$$

where k is the number of clusters, $s_{l,j}^2$ is the variance of the j^{th} cluster on the l^{th} dimension, and s_l^2 is the global variance of the whole data on the l^{th} dimension:

$$s_l^2 = \frac{1}{N-1}\sum_{t=1}^{N}(x_{l,t} - m_l)^2, m_l = \frac{1}{N}\sum_{t=1}^{N} x_{l,t}$$

The $Score_{l,j}$ indicates the relevance of the lth feature with respect to the j^{th} cluster. Thus, the average relevance of the lth feature for the clustering is represented by the $SCORE_l$. When the $SCORE_l$ receives a value close to the maximum value (i.e. 1), it approximately indicates that all the local variances of the k clusters on this dimension are considerably small in comparison to the global variance of this dimension, which is tantamount to indicating these clusters far away from each other on this dimension. Hence, these features are very relevant to the partitioning task. Otherwise, the $SCORE_l$ will receive the score close to the minimum value (i.e. 0).

According to the score of each feature, we could obtain the refined relevant feature subset $R^{'}$ in the following way:

$$R^{'} = R - \{l | SCORE_l < \beta \cdot \max_{i \in R}(SCORE_i), l \in R\}$$

where R is the current relevant feature subset, and β is a user-defined threshold value.

IV. The Iterative Feature Selection and Clustering

Since the optimal number of clusters and the optimal features subset are inter-related, we integrate the feature selection scheme of Section III into the RPEM algorithm, thus the clustering and the unsupervised feature selection work in an iterative way. Specifically, at the end of each epoch of the RPEM algorithm, the approximately optimal number of clusters and the corresponding parameters can be estimated on a given feature space. The proposed feature selection method outputs a ranking of each feature in terms of the discriminability with respect to a *reference* partition, i.e., the current data partition. A new partition is performed on the currently chosen feature space in the epoch. Subsequently, the relevant feature subset is refined based on the current *reference* partition. Algorithm 2 presents the details of the algorithm.

Algorithm 2: The RPEM with Iterative Feature Selection.

input : $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, k_{max}, η, $epoch_{max}$, *initial* $\boldsymbol{\Theta}$
output: *The converged* $\hat{\boldsymbol{\Theta}}$ *on* $\hat{R}$, *the relevant feature subset* $\hat{R}$

$\hat{R} \leftarrow \{all\ features\}$;
$epoch\_count \leftarrow 0$;
while $epoch\_count \leq epoch_{max}$ **do**
  for $t \leftarrow 1$ **to** N **do**
    Step 1: *Calculate* $h(j|\mathbf{x}_t, \hat{\Theta})$*'s to obtain* $g(j|\mathbf{x}_t, \hat{\Theta})$*'s on* $\hat{R}$*;*
    Step 2: *Update parameters* $\hat{\Theta}$ *on* $\hat{R}$*;*
    $\hat{\Theta}^{(new)} = \hat{\Theta}^{(old)} + \Delta\Theta = \hat{\Theta}^{(old)} + \eta\ \frac{\partial \mathcal{M}(\mathbf{x}_t;\hat{\Theta})}{\partial \Theta}\Big|_{\hat{\Theta}^{(old)}}$
  end
  $\hat{R}^{'} \leftarrow$ `FeatureSelection`$(\hat{R})$;
  $\hat{R} \leftarrow \hat{R}^{'}$;
  $epoch\_count \leftarrow epoch\_count + 1$;
end

Procedure `FeatureSelection`(R)

input : R
output: $R^{'}$

Calculate $SCORE_l, l \in R$;
$R^{'} \leftarrow R - \{l | SCORE_l < \beta \cdot \max_{i \in R}(SCORE_i), l \in R\}$;

The rationale behind this iterative execution of clustering and the feature selection can be interpreted as follows: Although the optimal feature subset on which an optimal clustering may be obtained is not known *a priori*, we can expect to obtain a potential optimal clustering result on the current given subset. By evaluating on the current reference clustering, it then performs the feature selection procedure

to further refine the relevant feature subset, which may lead to an even better partition in the next epoch.

In the above algorithm, the weight function $g(j|\mathbf{x}_t, \Theta)$'s are designed as:

$$g(j|\mathbf{x}_t, \Theta) = I(j|\mathbf{x}_t, \Theta) + h(j|\mathbf{x}_t, \Theta), j = 1, \ldots, k_{max}$$

where the $I(j|\mathbf{x}_t, \Theta)$ is defined by (4). It is easy to verify that the above design still satisfies the required constraints on the $g(j|\mathbf{x}_t, \Theta)$. Obviously, such a design gives the winning component only, i.e., the c^{th} component, at each time step an extra award whose value is $I(c|\mathbf{x}_t, \Theta) = 1$. This weight design actually penalizes those rival components in an implicit way. Consequently, it is able to automatically determine an appropriate number of components as well.

Since the RPEM algorithm is able to prune the redundant components, the relevance score calculation in each epoch should be therefore adjusted as:

$$SCORE_l = \frac{1}{k_{nz}} \sum_{j=1}^{k_{nz}} Score_{l,j} = \frac{1}{k_{nz}} \sum_{j=1}^{k_{nz}} (1 - \frac{s_{l,j}^2}{s_l^2}) \; l = 1, \ldots, d$$

where k_{nz} is the number of the clusters in the current *reference* partition with

$$k_{nz} = k_{max} - |K|, K = \{j|\alpha_j \equiv 0, j = 1, \ldots, k_{max}\}.$$

$|K|$ is the cardinality of the set K, which contains the index variables marking the clusters whose weights have been pruned towards zero. In general, we should not include such components in the feature relevance score calculation.

V. Experimental Results

This section shows the experimental results on two synthetic data sets and four real world benchmark data sets from the UCI repository [14]. In all the experiments, the initial number of components k_{max} should be large enough so that the initialization properly covers the data. We used the following inequality to estimate the appropriate initial number of components suggested in [12]:

$$k > \frac{\log \sigma}{\log(1 - \alpha_{min})}$$

where $\alpha_{min} = \min\{\alpha_1, \ldots, \alpha_k\}$ is the mixing proportion of the component which is mostly likely to be missed in the initialization. Under the circumstances, if we desire the probability of a successful initialization is no lower than $1 - \sigma = 0.95$, and suppose $\alpha_{min} = 0.2$, k should be greater than 12. We therefore set $k_{max} = 20$, and the initial component weights $\alpha_j = 1/k_{max}$ $(j = 1, \ldots, k_{max})$. The initial centers of each clusters $\mathbf{m}_j$'s were randomly chosen from data points, the initial variances of the clusters on each dimension were set to a fraction (e.g., we arbitrary set it at 1/5) of the global variance on the lth dimension: $s_{l,j}^2 = \frac{1}{5} s_l^2$, and the constant β was set to 0.2. We found that the initialization to these parameters performed reasonably well.

Firstly, we investigated the capability of the proposed algorithm on the model and feature selections using a synthetic data. We appended 8 independent variables, sampled from a standard normal distribution, to each data generated from the following bivariate Gaussian mixture structure, yielding a 10-dimensional data set with 1000 points.

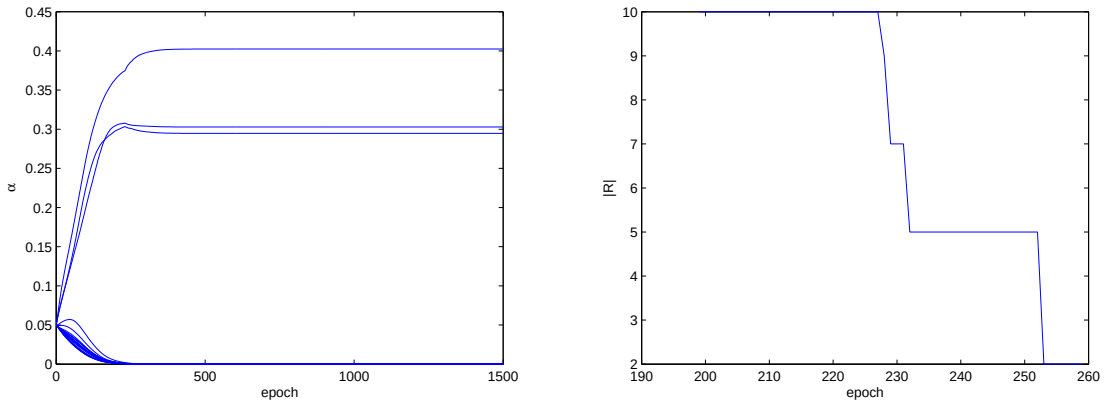
$$0.3 * \mathcal{N}\left[\begin{pmatrix} 1 \\ 1 \end{pmatrix}; \begin{pmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{pmatrix}\right] + 0.4 * \mathcal{N}\left[\begin{pmatrix} 1 \\ 5 \end{pmatrix}; \begin{pmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{pmatrix}\right]$$
$$+0.3 * \mathcal{N}\left[\begin{pmatrix} 5 \\ 5 \end{pmatrix}; \begin{pmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{pmatrix}\right]$$

Apparently, the last 8 dimensions are unimodal and irrelevant to the clustering. The objective is to detect the clustering on the first two dimensions and identify the last 8 irrelevant features. We ran the proposed algorithm 10 times, the 3 components and the irrelevant features were always correctly found in all runs we have tried so far. Figure. 2 shows the learning curves of the component weights and the size of relevant features subset in a typical run.

We then compared our algorithm with the one proposed by Law et, al [5]. The algorithm in [5] makes the *soft* decisions on whether the feature is relevant for the clustering or not, and has to pre-assume the irrelevant features conformed to a Gaussian distribution. Otherwise, its performance would be degraded to a certain degree. For example, we appended 8 variables uniformly distributed between 0 and 5, to the data from the above bivariate mixture structure, provided that the distribution of the irrelevant features is the Gaussian when using the algorithm in [5]. It is found that the algorithm of [5] was unable to give a proper inference about the clusters any more. Instead, it always largely over-fitted the data as illustrated in Figure 3. This implies that the algorithm of [5] is sensitive to the distribution of irrelevant features.

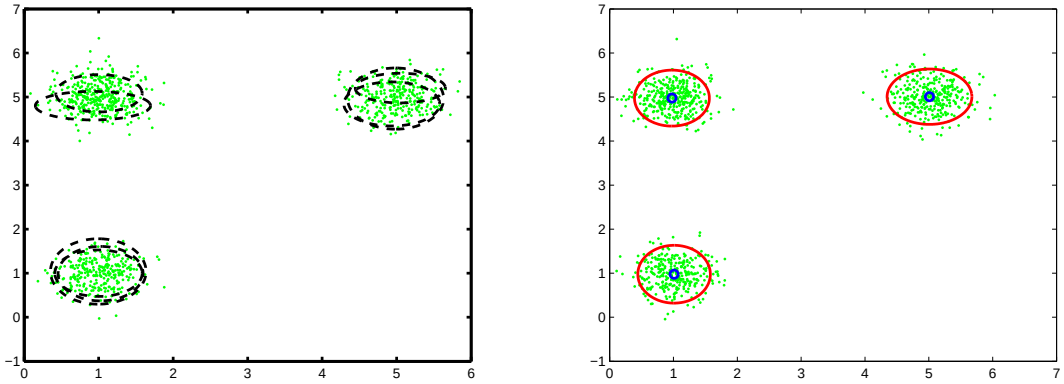
In contrast, the proposed algorithm circumvents this sensitivity. As shown in Figure 3, it has succeeded to infer the clustering structure in the original feature space. The reason is that we assume all the features are relevant at first, and then prune the features from the relevant feature subset, according to the "scoring" derived upon the *reference* clustering in the current epoch. In our method, we need not assume the explicit form of the distribution of the irrelevant features. Figure 4 demonstrates its learning curves of the component weights and the size of the relevant feature subset. It is interesting to note that, both in Figure 2 and 4, the components weights were gradually converged over the epochs when the feature elimination was undergoing, indicating that the feature selection had indeed facilitated the clustering.

Further, we show the proposed algorithm on 4 benchmark real-world data sets [14] in comparison to the RPEM algorithm and the algorithm in [5]. The characters of the data are summarized in Table I. Each data set has N data points with d features from k^* classes. We evaluate the accuracy of the obtained partition with the *error rate* index. After dividing the original data set into the training set and the testing set of the equal size, we executed the above algorithms on the training set to obtain the parameters of the Gaussian mixture model, then each data point in the testing set was classified



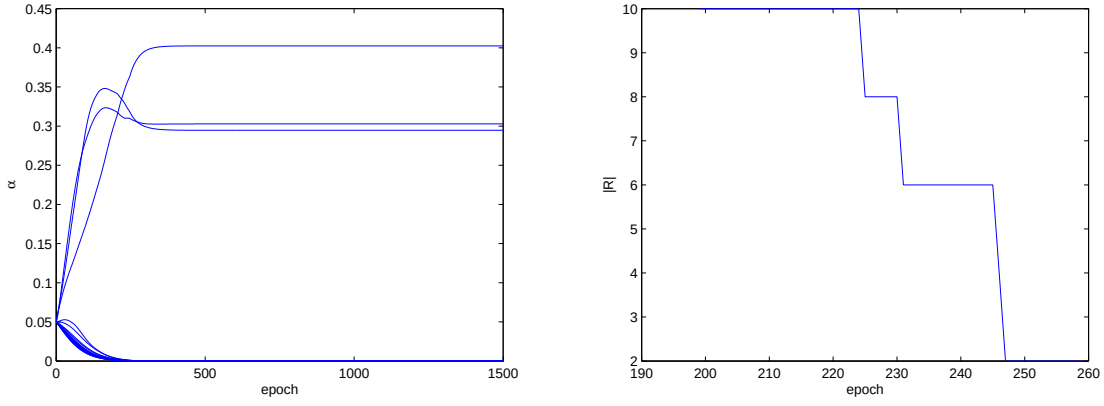
(a) (b)

Fig. 2. The results on the first synthetic data set. (a) the learning curve of $(\{\hat{\alpha}_j\}_{j=1}^{k_{max}})$; (b) the interval when the feature elimination was undergoing, where $|R|$ denotes the size of the relevant feature subset.



(a) (b)

Fig. 3. (a) The clustering results on the second synthetic data set obtained by the algorithm in [5], and (b) the proposed algorithm on the first two features, with the circle marking each cluster and the "o" marking the center of each cluster. In (a), $\hat{k}^* = 8$ (over-fitting), and in (b), $\hat{k}^* = 3$.



(a) (b)

Fig. 4. The results on the second synthetic data set. (a) the learning curve of $(\{\hat{\alpha}_j\}_{j=1}^{k_{max}})$; (b) the interval when the feature elimination was undergoing, where $|R|$ denotes the size of the relevant feature subset.

based on its posterior probability with a class label assigned. The error rate is computed by the mismatch degree between the obtained labels and the ground-truth class labels. The mean and the standard deviation of the *error rate*, along with those of the estimated number of clusters in 10-fold runs on the 4 real world data sets are listed in Table II.

TABLE I
THE REAL WORLD DATA SETS

Data set	d	N	k^*
wine	13	178	3
German	24	1000	2
wdbc	30	569	2
ionosphere	34	351	2

It could be observed from Table II that the proposed method has reduced the error rates on all sets compared to the RPEM algorithm. This is because not all features are relevant with respect to the partitioning task. These features with less discriminating power might confuse the RPEM clustering algorithm. Due to the iterative execution of the clustering and the feature selection, the potential optimal cluster-searching space shrank, thus leading to a better performance. The proportions of the average selected features by our algorithm in the whole feature set for each data sets (denoted as PFS) are reported in Table III.

When comparing the proposed algorithm with the algorithm in [5], although they are comparative in terms of *error rate*, our algorithm seems always given a closer estimation of the model order than algorithm in [5], the latter one is more likely to use more components especially for relatively high dimensional data set.

VI. CONCLUSION

We have presented a feature relevance measurement and integrated it into the RPEM algorithm. Subsequently, the proposed algorithm is able to find an appropriate number of clusters and relevant features for GM clustering. The experimental results have shown that the proposed algorithm outperforms the RPEM and the algorithm of [5].

ACKNOWLEDGMENT

This work was supported by the grants from the Research Grant Council of Hong Kong SAR (Project Codes: HKBU 2156/04E and HKBU 210306), and by the Faculty Research Grants of Hong Kong Baptist University (Project Codes: FRG/05-06/II-42 and FRG/06-07/II-07).

TABLE II
RESULTS OF THE 10-FOLD RUNS ON THE TEST SETS FOR EACH ALGORITHM

Data Set	Method	Model Order $mean \pm std$	Error Rate $mean \pm std$
wine	RPEM	2.5 ± 0.7	0.0843 ± 0.0261
	algorithm in [5]	3.3 ± 1.4	0.0673 ± 0.0286
	proposed method	3.2 ± 0.4	$\mathbf{0.0506 \pm 0.0269}$
German	RPEM	2.1 ± 0.2	0.4620 ± 0.0531
	algorithm in [5]	1.7 ± 0.5	0.3510 ± 0.0716
	proposed method	2.6 ± 1.1	$\mathbf{0.3096 \pm 0.0153}$
wdbc	RPEM	1.7 ± 0.4	0.2610 ± 0.0781
	algorithm in [5]	5.7 ± 0.3	0.1005 ± 0.0349
	proposed method	2.6 ± 0.7	0.1044 ± 0.0217
ionosphere	RPEM	1.8 ± 0.5	0.4056 ± 0.0121
	algorithm in [5]	3.2 ± 0.6	0.2268 ± 0.0386
	proposed method	**fixed at 2**	$\mathbf{0.2198 \pm 0.0761}$

TABLE III
THE PROPORTION OF THE AVERAGE SELECTED FEATURES BY OUR ALGORITHM IN THE WHOLE FEATURE SET IN THE 10-FOLD RUNS

Data	synthetic1	synthetic2	wine	German	wdbc	ionosphere
PSF	20%	20%	93.1%	24.2%	48.7%	85.3%

REFERENCES

[1] M. C. Dash, K. Scheuermann and P. H. Liu, "Feature selection for clustering-A filter solution", *Proc. IEEE International Conference on Data Mining*, Maebashi City, Japan, pp. 115–122, 2002.

[2] P. Mitra, C. A. Murthy and S. K. Pal, "Unsupervised feature selection using feature similarity", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301–312, Mar. 2002.

[3] J. Dy and C. Brodley, "Visualization and interactive feature selection for unsupervised data", *Proc. ACM Special Interest Group on Knowledge Discovery in Data*, Boston, Massachusetts, United States, pp. 360–364, 2000.

[4] J. Dy and C. Brodley, "Feature selection for unsupervised learning", *The Journal of Machine Learning Research*, vol. 5, pp. 845–889, 2004.

[5] M. H. C. Law, M. A. T. Figueiredo and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1154–1166, September 2004.

[6] C. Constantinopoulos, M. K. Titsias and A. Likas, "Bayesian feature and model selection for Gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp. 1013–1018, June 2006.

[7] A. P. Dempster, N. M. Laird and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of Royal Statistical Society (B)*, vol.39, no. 1, pp. 1–38, 1977.

[8] D. MacKay, "*Information Theory, Inference, and Learning Algorithms*", Cambridge Univ. Press, 2003.

[9] C. Bishop, "*Pattern Recognition and Machine Learning*", Springer, 2006.

[10] G. Schwarz,"Estimating the dimension of a model", *Annals of Statistics*, vol.6, no. 2, pp.461–464, 1978.

[11] C. Wallace and P. Freeman,"Estimation and inference via compact coding,"*Journal of Royal Statistical Society (B)*, vol.49, no. 3, pp. 240–265, 1987.

[12] M.A.T. Figueiredo and A.K. Jain,"Unsupervised Learning of Finite Mixture Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381-396, March 2002.

[13] Y. M. Cheung, "Maximum weighted likelihood via rival penalized EM for density mixture clustering with automatic model selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 750–761, June 2005.

[14] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," http://www.ics.uci.edu/mlearn/MLRepository.html, University of California at Irvine, 1998.