Pattern Recognition 42 (2009) 243 – 250

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A new feature selection method for Gaussian mixture clustering

Hong Zeng, Yiu-Ming Cheung*

Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China

ARTICLE INFO

Article history:
Received 7 December 2007
Received in revised form 29 May 2008
Accepted 31 May 2008

Keywords:
Gaussian mixture
Clustering
Feature selection
Relevance
Redundance

ABSTRACT

With the wide applications of Gaussian mixture clustering, e.g., in semantic video classification [H. Luo, J. Fan, J. Xiao, X. Zhu, Semantic principal video shot classification via mixture Gaussian, in: Proceedings of the 2003 International Conference on Multimedia and Expo, vol. 2, 2003, pp. 189–192], it is a nontrivial task to select the useful features in Gaussian mixture clustering without class labels. This paper, therefore, proposes a new feature selection method, through which not only the most relevant features are identified, but the redundant features are also eliminated so that the smallest relevant feature subset can be found. We integrate this method with our recently proposed Gaussian mixture clustering approach, namely rival penalized expectation-maximization (RPEM) algorithm [Y.M. Cheung, A rival penalized EM algorithm towards maximizing weighted likelihood for density mixture clustering with automatic model selection, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004, pp. 633–636; Y.M. Cheung, Maximum weighted likelihood via rival penalized EM for density mixture clustering with automatic model selection, IEEE Trans. Knowl. Data Eng. 17(6) (2005) 750–761], which is able to determine the number of components (i.e., the model order selection) in a Gaussian mixture automatically. Subsequently, the data clustering, model selection, and the feature selection are all performed in a single learning process. Experimental results have shown the efficacy of the proposed approach.

1. Introduction

Gaussian mixture (GM) clustering has been widely applied to a variety of scientific areas such as semantic video classification [1], data mining, microarray data analysis, pattern recognition, signal processing, image processing, and so forth. In general, GM clustering involves the model selection, i.e., to determine the number of components in a mixture (also called *model order* interchangeably), and the estimation of the parameters of each component in a mixture, through the observed data each represented as a vector of features (also referred to as *attributes* or *variables*). Among the existing GM clustering algorithms, most of them assume that the features of an observation vector (called *observation* for short) have the same contribution to the clustering structure, which, however, may not be true from the practical viewpoint. That is, there may be some irrelevant features in the observations. Under the circumstances, the inclusion of such features could hinder the clustering algorithm from detecting the clustering characteristic of observations. In addition, among the relevant features, some might be redundant as they do not carry any additional partitioning information. According to the Occam's Razor principle, those redundant features may deteriorate the generalization ability of the learned model, e.g., see Ref. [2]. Hence, it is always desirable to find the smallest relevant feature subset, which may bring on several other potential benefits such as the reduction of the collection and storage requirements, the comprehensible enhancement of the resulting partition, and so forth. In the supervised learning, discriminant features or the combination of these features could be effectively extracted based on the class label information, e.g., see several recent works [3–6]. However, in the unsupervised learning, it is a nontrivial task to perform the feature selection in the absence of the ground-truth labels that could guide the assessment of the relevance and redundancy for each feature. The problem becomes even more challenging when the true number of clusters is unknown *a priori*. It is known that the optimal feature subset and the optimal number of clusters are inter-related, i.e., different clustering results might be obtained on different feature subsets. This suggests that the feature selection should be taken into account jointly with the clustering and the model selection.

In the literature, there have been several representative methods that address the issue of the feature selection for the clustering. In the approaches [7,8], features are typically chosen prior to a clustering algorithm based on the general characteristics of observed data. Although they significantly reduce the dimensionality, these selected features may not be necessarily well suited to a mining algorithm

* Corresponding author. Tel.: +852 34115155.
E-mail address: ymc@comp.hkbu.edu.hk (Y.-M. Cheung).

0031-3203/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.05.030

[9]. Thus, in order to obtain both optima for the feature subset and the clustering structure, some algorithms, e.g., see Refs. [10,11], wrap the feature selection around the clustering algorithm. Such kind of approaches can effectively improve the partitioning accuracy at a cost of more computations because the feature evaluation phase repeatedly uses the intermediate outputs of the clustering algorithm to evaluate the quality of the feature subset candidates. Recently, the approaches in Refs. [9,12] have been proposed to tackle these two tasks in a single optimization paradigm. Several preliminary experiments in Refs. [9,12] have shown the promising results. Nevertheless, such methods suppose that the explicit parametric density of an irrelevant feature is Gaussian. Although such feature selection methods can work well in some cases, their performance may be degraded when the assumption is violated. Moreover, to the best of our knowledge, the issue of feature redundance has not been studied yet in the unsupervised feature selections, although it has been recently studied in the supervised learning [2,13,14].

In this paper, we propose a new feature selection method, in which the clustering and the feature selection are performed iteratively. A new evaluation index is firstly introduced to identify the most relevant features. This new index does not need to explicitly specify the parametric density form of irrelevant features in contrast to the feature relevance measurements used in Refs. [9,12]. The efficacy of this index has been demonstrated in our preliminary work [15]. Further, our feature selection method eliminates the redundant features, using the Markov Blanket filter [13,16], to find the smallest relevant feature subset that represents the data partitions of interest quite well. We integrate the proposed feature selection method with our recently proposed GM clustering approach, namely, rival penalized expectation-maximization (RPEM) algorithm [17,18], which is able to determine the number of components (i.e., the model order selection) in a GM automatically. Subsequently, the data clustering, model selection, and the feature selection are all performed in a single learning process. Experimental results have shown promising results of the proposed algorithm on both synthetic and real data.

The remainder of the paper is organized as follows. Section 2 overviews the GM clustering and RPEM algorithm. The proposed feature selection method is described in Section 3. Section 4 presents the proposed algorithm in detail, and Section 5 shows the experimental results. Finally, we draw our concluding remarks in Section 6.

2. Overview of GM clustering and RPEM algorithm

2.1. GM clustering

Suppose that N i.i.d. observations, denoted as $\mathbf{X_N}=\{\mathbf{x_1},\mathbf{x_2},\ldots,\mathbf{x_N}\}$, are 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

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

where each observation $\mathbf{x_t}$ $(1\leqslant t\leqslant N)$ is a column vector of d-dimensional features: $x_{1t},\ldots,x_{dt}$. Furthermore, $p(\mathbf{x}_t|\theta_j^*)$ is the jth Gaussian component with the parameter $\theta_j^*=\{\boldsymbol{\mu}_j^*,\boldsymbol{\Sigma}_j^*\}$, where $\boldsymbol{\mu}_j^*$ and $\boldsymbol{\Sigma}_j^*$ are the mean vector (also called *center* vector interchangeably) and covariance matrix of the jth component, respectively. α_j^* is the true mixing coefficient of the jth component in the mixture. The main task of GM 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, where k is an estimate of the true model order k^*. A general approach is to search in the parameter space to find a Θ that reaches a maximum of the fitness in terms of *maximum likelihood* (ML) defined below:

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

The commonly used searching strategy is the EM algorithm [19–21]. However, there is no penalty for the redundant mixture components in the above likelihood, which means that the model order k cannot be automatically determined and has to be assigned in advance. Although some model selection criteria, e.g., see Refs. [22,23], 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.

2.2. The RPEM algorithm

Recently, the RPEM algorithm [18,17] has been proposed to determine the model order automatically together with the estimation of the model parameters. This algorithm introduces unequal weights into the conventional likelihood; thus the weighted likelihood is written below:

$$Q(\Theta,\mathbf{X}_N)=\frac{1}{N}\sum_{t=1}^{N}\log p(\mathbf{x}_t|\Theta)=\frac{1}{N\zeta}\sum_{t=1}^{N}\mathscr{M}(\mathbf{x}_t;\Theta) \tag{2}$$

with

$$\mathscr{M}(\mathbf{x}_t;\Theta)=\sum_{j=1}^{k}g(j|\mathbf{x}_t,\Theta)\log[\alpha_j p(\mathbf{x}_t|\theta_j)]-\sum_{j=1}^{k}g(j|\mathbf{x}_t,\Theta)\log h(j|\mathbf{x}_t,\Theta), \tag{3}$$

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 jth 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:

$$\sum_{j=1}^{k}g(j|\mathbf{x}_t,\Theta)=\zeta \quad \text{for any } 1\leqslant t\leqslant N$$

and

$$\forall j,\quad \lim_{h(j|\mathbf{x}_t,\Theta)\to 0}g(j|\mathbf{x}_t,\Theta)\log h(j|\mathbf{x}_t,\Theta)=0,$$

where ζ is a positive constant. In Ref. [18], they are constructed by the following equation (with $\zeta=1$):

$$g(j|\mathbf{x}_t,\Theta)=(1+\varepsilon_t)I(j|\mathbf{x}_t,\Theta)-\varepsilon_t h(j|\mathbf{x}_t,\Theta) \tag{4}$$

with

$$I(j|\mathbf{x},\Theta)=\begin{cases}1 & \text{if } j=c\equiv\arg\max\limits_{1\leqslant i\leqslant k}h(i|\mathbf{x},\Theta)\\ 0 & \text{if } j=r\neq c\end{cases} \tag{5}$$

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, i.e., $j=c$ with $I(j|x,\Theta)=1$. Accordingly, a positive weight $g(c|\mathbf{x}_t,\Theta)$ will strengthen 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 finally, whereas the "pseudo-clusters" will be gradually faded out from the mixture.

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

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

The more detailed implementation of the RPEM can be found in Ref. [18]. 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, the learning rate η, the maximum number of epochs $epoch_{max}$, initialize Θ as $\Theta^{(0)}$.
output: *The converged* Θ.
1 $epoch\_count \leftarrow 0$, $m \leftarrow 0$;
2 **while** $epoch\_count \leqslant epoch_{max}$ **do**
3 **for** $t \leftarrow 1$ **to** N **do**
4 **Step 1**: *Given* $\Theta^{(m)}$, *calculate* $h(j|\mathbf{x}_t, \Theta^{(m)})$*'s to obtain* $g(j|\mathbf{x}_t, \Theta^{(m)})$*'s using Eq.* (4);
5 **Step 2**: $\Theta^{(m+1)} = \Theta^{(m)} + \Delta\Theta = \Theta^{(m)} + \eta \left.\frac{\partial \mathscr{M}(\mathbf{x}_t;\Theta)}{\partial \Theta}\right|_{\Theta^{(m)}}$
6 where η is a small positive learning rate. Let $m \leftarrow m+1$.
7 **end**
8 $epoch\_count \leftarrow epoch\_count + 1$;
9 **end**

3. Unsupervised feature selection

3.1. Selecting the relevant features

Let us begin with a simple example. We generate the data points from a bivariate two-component GM as shown in Fig. 1. If the two clusters are projected onto the Y-axis, it is unable to distinguish these two clusters by the feature Y because the observations from the two clusters are in the same dense region of this dimension. Hence, the feature Y will not be helpful in finding the clustering structure, i.e., it is irrelevant to the clustering. On the contrary, the projections onto the X-axis can provide the useful information regarding the clustering structure, thus the feature X is relevant to the clustering.

Based on this scenario, we claim that a feature is less relevant if, along this feature, the variance of observations in a cluster is closer to the global variance of observations in all clusters. Subsequently, we propose the following quantitative index to measure the relevance of each feature:

$$SCORE_l = \frac{1}{k}\sum_{j=1}^{k} Score_{l,j} = \frac{1}{k}\sum_{j=1}^{k}\left(1 - \frac{\delta_{l,j}^2}{\delta_l^2}\right), \quad l = 1, \ldots, d, \tag{6}$$

where $\delta_{l,j}^2$ is the variance of the jth cluster projected on the lth dimension:

$$\delta_{l,j}^2 = \frac{1}{N_j - 1}\sum_{t=1}^{N_j}(x_{l,t} - \mu_{l,j})^2, \quad \mathbf{x}_t \in j\text{th } cluster,$$

$N_j = \sum_{t=1}^{N} I(j|\mathbf{x}_t, \Theta)$ is the number of data in the jth cluster, $\mu_{l,j}$ is the mean of the lth feature $x_{l,t}$'s in the jth cluster, and $\sum_{j=1}^{k} N_j = N$. Furthermore, δ_l^2 is the global variance of the whole data on the lth dimension:

$$\delta_l^2 = \frac{1}{N-1}\sum_{t=1}^{N}(x_{l,t} - \mu_l)^2, \quad \mu_l = \frac{1}{N}\sum_{t=1}^{N} x_{l,t}.$$

The $Score_{l,j} = 1 - (\delta_{l,j}^2/\delta_l^2)$ measures the dissimilarity between the variance in the jth cluster and the global variance on the lth feature, which equivalently indicates the relevance of the lth feature for the jth cluster. Thus, $SCORE_l$ represents the average relevance of the lth feature to the clustering structure. When the value of $SCORE_l$ is close to the maximum value (i.e., 1), it represents the case 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, this feature is very useful to detect the grouping structure. Otherwise, the value of $SCORE_l$ will be close to the minimum value, i.e., 0. To prevent the score from being degenerated in a situation where $\delta_{l,j}^2$ is greater than δ_l^2 in Eq. (6),which may be caused by numerical computation in computer, we clip the $Score_{l,j}$ at 0. That is, we update $Score_{l,j}$ in Eq. (6) by

$$Score_{l,j} = \max\left(0, 1 - \frac{\delta_{l,j}^2}{\delta_l^2}\right). \tag{7}$$

According to the score of each feature, we can obtain the selected relevant feature subset R' in the following way:

$$R' = F - \{F_l | SCORE_l < \beta, F_l \in F\},$$

where F is the full feature set and β is a user-defined threshold value. By a rule of thumb, we set $\beta \in [0, 0.5]$. In the next subsection, we will further select the non-redundant features from R'.

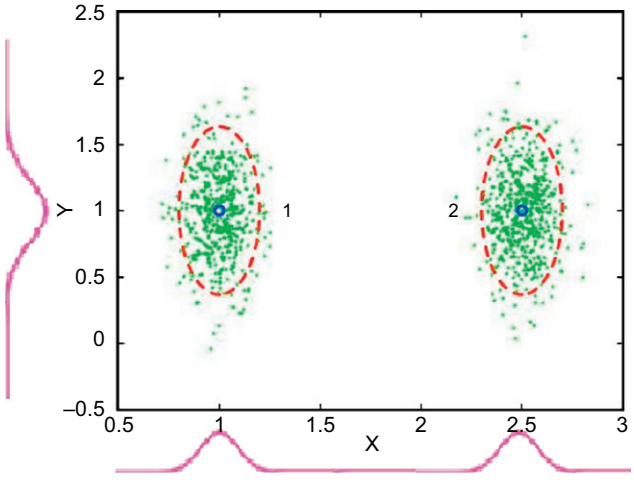

Fig. 1. The feature X is relevant to the partitioning, while the feature Y is irrelevant.

3.2. Selecting the non-redundant features

Based on the information theory, we know that a feature is redundant if it carries no additional partition information provided that the remaining features are presented. Hence, we are able to ignore it without compromising the model performance. Such an idea can be realized by a feature's Markov Blanket [24] as follows, which is originally proposed in the domain of supervised learning.

Definition 1 (*Markov Blanket*). Given a feature F_l, let $M_l \subset F$ ($F_l \notin M_l$), M_l is the Markov Blanket for F_l if the probability

$$P(F - M_l - F_l, C | F_l, M_l) = P(F - M_l - F_l, C | M_l),$$

where C is the class label.

Hence, if a Markov Blanket M_l for F_l can be found in the feature set F, i.e., M_l subsumes the information that F_l has about C, we are

able to eliminate the feature F_l from F without affecting the class prediction accuracy.

Since there might not be a full Markov Blanket for a feature, Koler and Sahami [16] proposed a method that sequentially eliminates such features based on the existence/non-existence of an *approximate* Markov Blanket in the candidate feature subset. Broadly, it iteratively constructs a candidate Markov Blanket M_l for F_l, and measures how close M_l is to a true Markov Blanket for F_l. If M_l is the closest one to a true Markov Blanket for F_l, F_l is eliminated and the process repeats. The closeness of M_l to a true Markov Blanket for F_l is measured by the expected cross entropy:

$$\Delta(F_l|M_l) = \sum_{f_{M_l} f_l} P(M_l = f_{M_l}, F_l = f_l) \times KL(P(C|M_l = f_{M_l}, F_l = f_l) \| P(C|M_l = f_{M_l})), \quad (8)$$

where f_{M_l} denotes the features in the candidate Markov Blanket, $KL(.\|.)$ denotes the Kullback–Leibler divergence with $KL(P\|Q) = \sum_x P(x)\log(P(x)/Q(x))$.

If M_l is a genuine Markov Blanket for F_l, we have $\Delta(F_l|M_l) = 0$. An approximate Markov Blanket is formulated by relaxing this requirement, i.e., let $\Delta(F_l|M_l)$ be a very small value. The candidate Markov Blanket is constructed by picking the top T features that have the highest Pearson correlation to F_l, where T (i.e., the size of Markov Blanket) is often a small integer. The reason for formulating the candidate Markov Blanket in this way is that the features in F_l's Markov Blanket M_l are directly influenced by F_l, while other features are conditionally independent of it as given by M_l. Since the expected cross entropy needs to compute the posterior probability $P(C|.)$, it will be convenient to utilize the binary values of original feature values to save the computational costs. An applicable discretization[1] method can be found in Ref. [13]. The complete Markov Blanket filtering algorithm in Refs. [13,16] is presented in Algorithm 2.

Algorithm 2. The Markov blanket filtering algorithm

Initialize
- Let $m = 1$ and $G^{(m)} = F$;

Iterate
- For $F_l \in G^{(m)}$, let M_l be the set of T features, and $F_i \in G^{(m)} - F_l$ for which the correlation between F_l and F_i is the highest;
- Compute $\Delta(F_l|M_l)$ for each feature l;
- Choose the F_{l_m} that minimizes $\Delta(F_l|M_l)$, and define $G^{(m+1)} = G^{(m)} - F_{l_m}$;
- Let $m = m + 1$;

Until $|G^{(m)}| = T$.

The sequence $\{l_1, l_2, \ldots, l_{|F|-T}\}$, in which the features are removed by this method, corresponds to a feature ranking in ascending order of non-redundancy. Thereby, the feature that appears first in the list (i.e. F_{l_1}, the first one that has been removed from F) is the most redundant among all the features, while the features left after Markov Blanket filtering algorithm has stopped, i.e., $\{F - \{F_{l_1}, F_{l_2}, \ldots, F_{l_{|F|-T}}\}\}$, are the least redundant.

Unfortunately, the Markov Blanket filtering algorithm needs class labels and cannot be straightforwardly applied to the clustering problems. To circumvent this difference, we assume that a set of clusters can be modeled as a set of different classes. Moreover, since the minimum value of expected cross entropy in the mth iteration, written as $\min_{F_l \in G^{(m)}} \Delta(F_l|M_l)$, increases over m, the most non-redundant features can be simply obtained via

$$R'' = \left\{ F_{l_m} | \min_{F_l \in G^{(m)}} \Delta(F_l|M_l) > \gamma \cdot \min_{F_l \in G^{(1)}} \Delta(F_l|M_l) \right\} \cup \{R' - \{F_{l_1}, F_{l_2}, \ldots, F_{l_{|R'|-T}}\}\},$$

where $m = 1, 2, \ldots, |R'| - T$, $F_{l_m} \in R'$, $G^{(1)} = R'$, and γ is a user-defined threshold (e.g., $\gamma = 2$).

4. Iterative feature selections in the RPEM algorithm

Since the optimal number of clusters and the optimal feature subset are inter-related, we integrate the feature selection schemes of Section 3 into the RPEM algorithm so that the feature selection and the clustering process are performed iteratively in a single learning process. Specifically, given a feature subset, we run the RPEM algorithm by scanning all observations once (i.e., one epoch) and obtain a near optimal partition. Then, the proposed feature selection scheme outputs a refined feature subset in terms of the relevance and non-redundance under the current data partition. Subsequently, a more accurate partition will be performed using the selected feature subset in the next epoch. Algorithm 3 presents the details of the proposed algorithm.

Algorithm 3. Iterative feature selection in RPEM clustering algorithm

input: $\mathbf{X}_N, k_{max}, \eta, epoch_{max}, \beta, \gamma, T$
output: *The most relevant and non-redundant feature subset* $\hat{R}$
1 $\hat{R} \leftarrow \{F\}$;
2 $epoch\_count \leftarrow 0$, $m \leftarrow 0$;
3 Initialize Θ as $\Theta^{(0)}$;
4 **while** $epoch\_count \leqslant epoch_{max}$ **do**
5 **for** $t \leftarrow 1$ **to** N **do**
6 **Step 1**: *Given* $\Theta^{(m)}$, *calculate* $h(j|\mathbf{x}_t, \Theta^{(m)})$*'s to obtain* $g(j|\mathbf{x}_t, \Theta^{(m)})$*'s on* $\hat{R}$;
7 **Step 2**: *Update parameters* Θ *on* F *by* $\Theta^{(m+1)} = \Theta^{(m)} + \eta \frac{\partial \mathscr{M}(\mathbf{x}_t;\Theta)}{\partial \Theta}\Big|_{\Theta^{(m)}}$, and let $m \leftarrow m + 1$;
8 **end**
9 $\hat{R} \leftarrow FeatureSelection(F, \beta, \gamma, T)$;
10 $epoch\_count \leftarrow epoch\_count + 1$;
11 **end**

Procedure: `Feature Selection` (F, β, γ, T)
input: F, β, γ, T
output: $\hat{R}$
// `Select the relevant features`
1 *Calculate* $SCORE_l, F_l \in F$;
2 $R' \leftarrow F - \{F_l | SCORE_l < \beta, F_l \in F\}$;
`Select the non-redundant features`
3 *Perform Markov Blanket filtering*;
4 $R'' = \{F_{l_m} | \min_{F_l \in G^{(m)}} \Delta(F_l|M_l) > \gamma \cdot \min_{F_l \in G^{(1)}} \Delta(F_l|M_l)\} \cup \{R' - \{F_{l_1}, F_{l_2}, \ldots, F_{l_{|R'|-T}}\}\}$;
5 $\hat{R} \leftarrow R''$;

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

$$g(j|\mathbf{x}_t, \Theta) = I(j|\mathbf{x}_t, \Theta) + h(j|\mathbf{x}_t, \Theta), \quad j = 1, \ldots, k_{max}. \quad (9)$$

It can be shown that the above design satisfies the required constraints on the $g(j|\mathbf{x}_t, \Theta)$'s. Evidently, at each time step, such a design gives the winning component only, i.e., the cth component, an extra award with the amount of $I(c|\mathbf{x}_t, \Theta) = 1$. This weight design actually penalizes those rival components in an implicit way. Consequently,

[1] The discretized feature is only used for computing the KL divergence, and the Pearson correlation is still calculated with the original feature values.

analogous to the weight design in Eq. (4), it enables the RPEM algorithm to automatically determine an appropriate number of components as well by gradually fading the redundant components out from the mixture.

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

$$SCORE_l = \frac{1}{k_{nz}} \sum_{j=1}^{k_{nz}} Score_{l,j} = \frac{1}{k_{nz}} \sum_{j=1}^{k_{nz}} \max\left(0, \left(1 - \frac{\delta_{l,j}^2}{\delta_l^2}\right)\right),$$

where k_{nz} is the number of the clusters in the current partition:

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

$|K|$ is the cardinality of the set K, which contains the index variables marking the clusters whose weights have been pruned to zero. In implementation, we can find the component whose weight is smaller than $1/N$, where N is the number of the observations. We should not include such components in the calculation of feature relevance score.

5. Experimental results

This section shows the experimental results on two synthetic data sets and four real-world benchmark data sets. In all the experiments, the initial number of components k_{max} should be safely large so that the initialization properly covers the data. We therefore, set $k_{max} = 10$, and the initial mixing coefficients $\alpha_j = 1/k_{max}$ $(j = 1, \ldots, k_{max})$. The initial centers of the clusters $\boldsymbol{\mu}_j$'s were randomly chosen from the data points, and the initial covariance matrices were

$$\boldsymbol{\Sigma}_j^{(0)} = \frac{1}{5d} trace\left(\frac{1}{N}\sum_{t=1}^{N}(\mathbf{x}_t - \boldsymbol{\mu})(\mathbf{x}_t - \boldsymbol{\mu})'\right)\mathbf{I}$$

where $\boldsymbol{\mu} = (1/N)\sum_{t=1}^{N}\mathbf{x}_t$ is the global mean vector of the data, and $\mathbf{I}$ is an identity matrix.

Furthermore, we can set $epoch_{max}$ at a value that is large enough. Alternatively, we can set it at a medium value at first. If the algorithm has not converged yet after $epoch_{max}$ epochs, we may adaptively increase the value of $epoch_{max}$ a little and let the algorithm run more several epochs until the convergence is achieved. The T parameter has been studied in Ref. [16]. It is suggested that T should be set at a very small integer, e.g., 1 or 2. We, therefore, set it at 2. It was found that there was no apparent difference between the performance by using $T = 2$ and 1. As for the two thresholds, β and γ, as a rule of thumb, we found that the algorithm works well when they were set at 0.4 and 2, respectively. In the following, we report the results based on these settings.

5.1. Synthetic data

Firstly, we investigated the capability of the proposed algorithm to select the relevant and non-redundant features while determining the correct number of clusters simultaneously. As shown in Fig. 2(a), 1000 data points were generated by the following bivariate GM:

$$\begin{aligned} p(\mathbf{x}|\Theta^*) = {} & 0.3 * p\left(\mathbf{x}\middle|\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}0.1 & 0.0\\0.0 & 0.1\end{pmatrix}\right) \\ & + 0.4 * p\left(\mathbf{x}\middle|\begin{pmatrix}1\\5\end{pmatrix}, \begin{pmatrix}0.1 & 0.0\\0.0 & 0.1\end{pmatrix}\right) \\ & + 0.3 * p\left(\mathbf{x}\middle|\begin{pmatrix}5\\5\end{pmatrix}, \begin{pmatrix}0.1 & 0.0\\0.0 & 0.1\end{pmatrix}\right). \end{aligned}$$

In Fig. 2(a), we are unable to discriminate the three clusters by a single dimension alone. Actually, both features are relevant to the clustering. We duplicated the two dimensions and formed a four-dimensional data. Each data were further appended with six independent variables that were sampled from a standard normal distribution and finally yielding a 10-dimensional data set to be analyzed.

Apparently, either $\{F_1, F_2\}$ or $\{F_3, F_4\}$ can determine the three clusters, i.e., one of the two pairs of features are redundant. The last six dimensions are unimodal, thus being irrelevant to the clustering. We ran the proposed algorithm (denoted as IRRFS-RPEM) 10 times, the three components and the non-redundant relevant feature subset were always correctly found in all runs. Fig. 2(b) shows the learning curve of the component mixing coefficients in a typical run. Table 1 lists some intermediate outputs. In the column of "ranking", the first row of each epoch is the relevance score ($SCORE_l$) in descending order; the second row is the minimum value of expected cross entropy, with its corresponding feature in the sequential removal order by the Markov Blanket filtering. The two rows under the column of "selected features" list the outputs in the two feature selection stages.

We then compared our algorithm with the one proposed by Law et al. [9], denoted as GMClusFW. GMClusFW 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. Thereby, its performance may be degraded to a certain degree if this assumption is violated. To illustrate this, we appended six variables uniformly distributed between zero and five to the data with the above four relevant dimensions, but the distribution of irrelevant features was still supposed to be the standard Gaussian. It is found that the algorithm of Ref. [9] was unable to give a proper inference about the clusters any more. Instead, it always largely over-fits the data as illustrated in Fig. 3(a). This implies that the algorithm of Ref. [9] is sensitive to the assumption upon the feature distributions.

In contrast, the proposed algorithm has circumvented this drawback. Fig. 3(b) demonstrates its learning curve of the component mixing coefficients. Table 2 presents its intermediate outcomes. As shown in Fig. 3(b), it succeeded to find the true clustering structure in the original feature space. Besides, only two most relevant features (i.e. $\{F_3, F_4\}$ as given in Table 2) were used without redundant features.

5.2. Real-world data

We further investigated the performance of the proposed algorithm on four benchmark real-world data sets [25]. We normalized the mean and variance of each data set to 0 and 1, respectively. For comparison, we also performed the RPEM algorithm, GMClusFW algorithm in Ref. [9], and a variant of the proposed algorithm (denoted as IRFS–RPEM) that carried out the relevancy analysis only in the feature selection phase without considering the feature redundancy. The same weight functions were chosen for RPEM, IRFS–RPEM and IRRFS–RPEM. We evaluated the clustering accuracy using the index of *error rate*. 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 parameter estimates of the GM model, then each data point in the testing set was appended a label of the cluster it belonged to, where the cluster label was determined by the majority vote of training data in that cluster. The error rate was computed by the mismatch degree between the obtained labels of the testing points and their 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 four real-world data sets are summarized in Table 3. The following are some remarks:

Remark 1. It can be seen that both IRFS–RPEM and IRRFS–RPEM have reduced the error rates on all sets compared to the RPEM algorithm because not all features are relevant to the partitioning task.

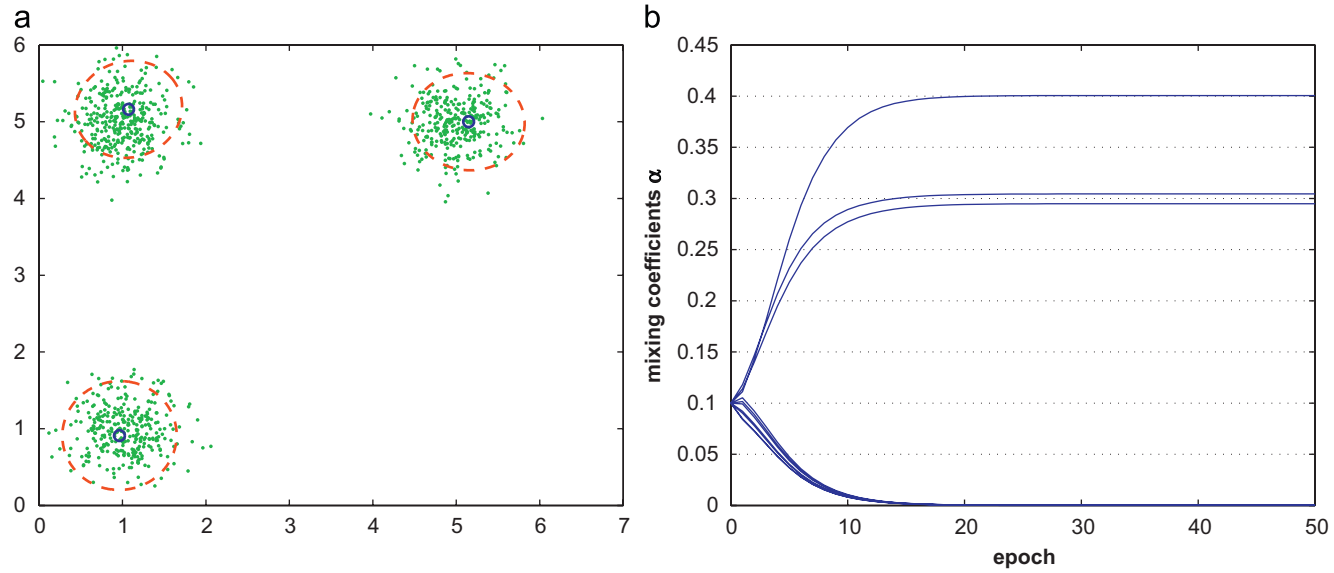

Fig. 2. (a) The bivariate data with three-cluster structure. (b) The learning curve of $(\{\alpha_j\}_{j=1}^{k_{max}})$ by the proposed IRRFS-RPEM algorithm on the first synthetic data.

Table 1
The intermediate outcomes of the proposed IRRFS-RPEM algorithm on the first synthetic data, where the corresponding feature F_l of a score under the column of "ranking" is shown in the parentheses

Epoch	Ranking	Selected features
1	$0.97(F_1)\ 0.97(F_4)\ 0.97(F_3)\ 0.97(F_2)\ 0.35(F_6)\ 0.33(F_7)\ 0.22(F_{10})\ 0.17(F_8)\ 0.11(F_5)\ 0.11(F_9)$	$\{F_1,F_2,F_3,F_4\}$
	$0(F_1)\ 0(F_2)$	$\{F_3,F_4\}$
15	$0.86(F_1)\ 0.86(F_2)\ 0.84(F_4)\ 0.84(F_3)\ 0.29(F_7)\ 0.24(F_8)\ 0.22(F_6)\ 0.21(F_{10})\ 0.20(F_5)\ 0.17(F_9)$	$\{F_1,F_2,F_3,F_4\}$
	$0(F_1)\ 0(F_2)$	$\{F_3,F_4\}$
50	$0.97(F_2)\ 0.97(F_1)\ 0.97(F_4)\ 0.97(F_3)\ 0.04(F_7)\ 0.03(F_5)\ 0.01(F_8)\ 0.01(F_9)\ 0.00(F_{10})\ 0.00(F_6)$	$\{F_1,F_2,F_3,F_4\}$
	$0(F_1)\ 0(F_2)$	$\{F_3,F_4\}$

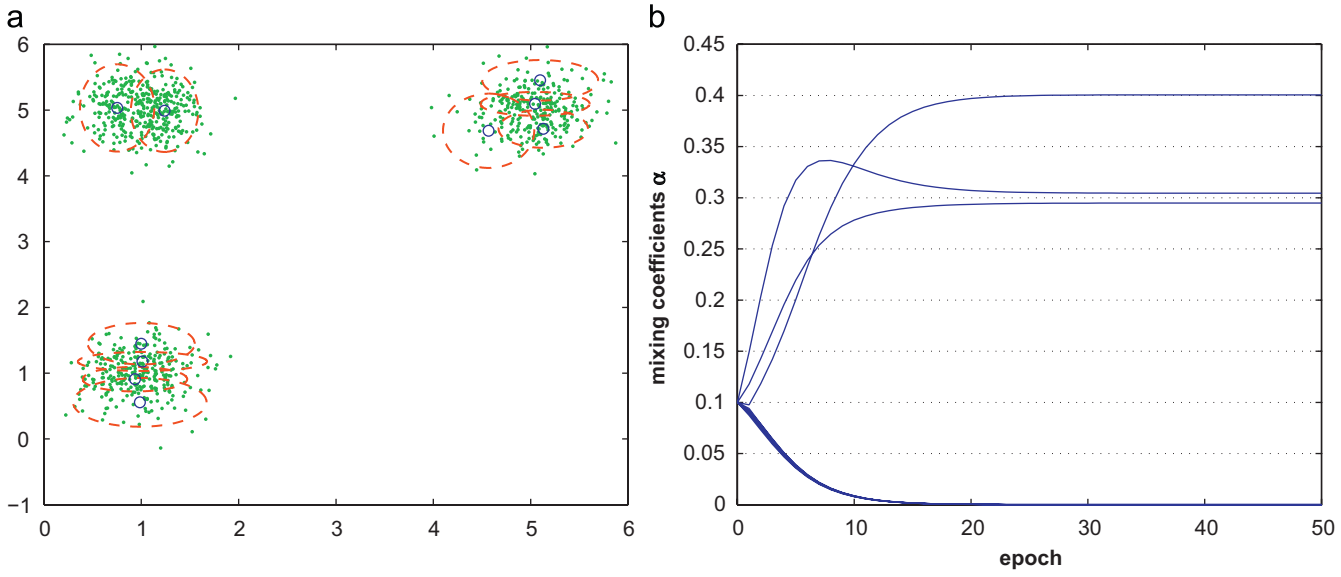

Fig. 3. (a) The clustering results on the second synthetic data set obtained by GMClusFW in Ref. [9] projected on the first two features: $k=10$ (over-fitting). (b) The learning curve of $(\{\alpha_j\}_{j=1}^{k_{max}})$ by the proposed IRRFS-RPEM algorithm on the second synthetic data: $k=3$ (correct).

Remark 2. In Table 3, RPEM gives the higher accuracy of model order. The possible reasons are as follows: Firstly, the number of classes may not be exactly the same as the number of clusters in the data. Secondly, according to our experimental results, we found that the involvement of these less discriminating features could not only lead to the "over-fitting" of cluster model (which has been pointed out by Law et al. [9] and etc.), but also may result in "under-fitting" (see Table 3, the model order results for RPEM algorithm). Therefore,

Table 2
The intermediate outcomes of the proposed IRRFS-RPEM algorithm on the second synthetic data

Epoch	Ranking	Selected features
1	$0.97(F_1)\ 0.97(F_2)\ 0.97(F_3)\ 0.97(F_4)\ 0.39(F_6)\ 0.36(F_5)\ 0.27(F_9)\ 0.24(F_8)\ 0.17(F_{10})\ 0.16(F_7)$	$\{F_1, F_2, F_3, F_4\}$
	$0(F_1)\ 0(F_2)$	$\{F_3, F_4\}$
7	$0.67(F_1)\ 0.66(F_4)\ 0.66(F_2)\ 0.62(F_3)\ 0.21(F_8)\ 0.17(F_6)\ 0.17(F_{10})\ 0.11(F_7)\ 0.06(F_5)\ 0.03(F_9)$	$\{F_1, F_2, F_3, F_4\}$
	$0(F_1)\ 0(F_2)$	$\{F_3, F_4\}$
50	$0.97(F_1)\ 0.97(F_2)\ 0.97(F_3)\ 0.97(F_4)\ 0.04(F_{10})\ 0.01(F_8)\ 0.01(F_9)\ 0.01(F_7)\ 0.00(F_5)\ 0.00(F_6)$	$\{F_1, F_2, F_3, F_4\}$
	$0(F_1)\ 0(F_2)$	$\{F_3, F_4\}$

Table 3
Results of the 10-fold runs on the test sets for each algorithm

Data set	Method	Model order (mean ± std)	Error rate (mean ± std)
wdbc	RPEM	1.7 ± 0.4	0.2610 ± 0.0781
$d = 30$	GMClusFW	5.7 ± 0.3	0.1005 ± 0.0349
$N = 569$	IRFS–RPEM	2.3 ± 0.4	0.1021 ± 0.0546
$k^* = 2$	IRRFS–RPEM	fixed at 2	$\mathbf{0.0897 \pm 0.0308}$
sonar	RPEM	2.3 ± 0.8	0.4651 ± 0.0532
$d = 60$	GMClusFW	1.0 ± 0.0	0.5000 ± 0.0000
$N = 1000$	IRFS–RPEM	2.8 ± 0.6	0.3625 ± 0.0394
$k^* = 2$	IRRFS–RPEM	2.7 ± 0.7	$\mathbf{0.3221 \pm 0.0333}$
wine	RPEM	2.5 ± 0.7	0.0843 ± 0.0261
$d = 13$	GMClusFW	3.3 ± 1.4	0.0673 ± 0.0286
$N = 178$	IRFS–RPEM	4.7 ± 1.7	$\mathbf{0.0492 \pm 0.0182}$
$k^* = 3$	IRRFS–RPEM	3.1 ± 0.5	0.0509 ± 0.0248
ionosphere	RPEM	1.8 ± 0.5	0.4056 ± 0.0121
$d = 32$	GMClusFW	3.2 ± 0.6	0.2268 ± 0.0386
$N = 351$	IRFS–RPEM	2.6 ± 0.8	0.2921 ± 0.0453
$k^* = 2$	IRRFS–RPEM	2.5 ± 0.5	$\mathbf{0.2121 \pm 0.0273}$

Each data set has N data points with d features from k^* classes.

Table 4
The proportions of the average selected features in the 10-fold runs

Data	IRFS–RPEM (%)	IRRFS–RPEM (%)
synthetic1	40	20
synthetic2	40	20
wdbc	51.16	50.33
sonar	57	55.83
wine	83.65	62.31
ionosphere	68.13	34.38

the clustering accuracy may be affected. It is worth to mention that we found the same "under-fitting" phenomenon when applying the algorithm by Law et al. to the sonar data. Their algorithm always determines that there is only one cluster, but actually there are at least two clusters in this data (see Table 3).

Remark 3. Table 4 lists the proportions of the average selected features by IRFS–RPEM and IRRFS–RPEM in the whole feature set for each data sets. For the *wdbc* and the *sonar* data sets, IRFS–RPEM and IRRFS–RPEM have selected approximately the same number of features with the similar predictive performances. This implies that there may not be much redundancy in the selected relevant features for these two data sets. In contrast, the *wine* and *ionosphere* sets both likely present the redundancy in the selected relevant features. In particular, for the *ionosphere* data set, the accuracy even gets improved with much fewer features (nearly $\frac{1}{3}$ of its original size) selected by IRRFS–RPEM, compared to the features selected by IRFS–RPEM. Although IRFS–RPEM seems to give a slightly better predictive accuracy on the *wine* data, the number of components it utilized is greater than the correct one.

Remark 4. The proposed algorithm outperforms GMClusFW in terms of *error rate* as shown in Table 3. Further, IRRFS–RPEM always gives a smaller estimation of the model order than GMClusFW. The latter is more likely to use more components for all the utilized data sets, especially for the *wdbc* data set. This phenomenon is consistent with the results we have demonstrated on the second synthetic data set.

6. Concluding remarks

In this paper, we have proposed a new feature selection method for GM clustering. Firstly, we have introduced a new feature relevance measurement index to identify the most relevant features. This new index does not need to explicitly specify the parametric density form of irrelevant features, thus it is more applicable to the situation without knowing the distribution of the irrelevant features a priori. Secondly, we have extended the usage of the Markov Blanket filter to select the non-redundant relevant features in the unsupervised learning. These feature selection schemes are then integrated into the RPEM clustering algorithm, whereby a new algorithm iterating between the clustering and the feature selection has been developed. Experiments have shown the effectiveness of the proposed algorithm in comparison with the existing methods on both the synthetic and real-world benchmark data sets. Undoubtedly, the techniques proposed in this paper are applicable to semantic multimedia data as well, e.g. semantic image classification, in which the irrelevant and redundant features are common in classifying the semantic images. We shall leave this study elsewhere in the future.

Acknowledgments

The work in this paper is jointly supported by a grant from the Research Grant Council of the Hong Kong SAR (Project no: HKBU 210306) and the Faculty Research Grant of Hong Kong Baptist University under Project FRG/07-08/II-54.

References

[1] H. Luo, J. Fan, J. Xiao, X. Zhu, Semantic principal video shot classification via mixture Gaussian, in: Proceedings of the 2003 International Conference on Multimedia and Expo, vol. 2, 2003, pp. 189–192.
[2] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (7–8) (2003) 1157–1182.
[3] X. Li, S. Lin, S. Yan, D. Xu, Discriminant locally linear embedding with high order tensor data, IEEE Trans. Syst. Man Cybernet. B 38 (2) (2008) 342–352.
[4] D. Tao, X. Li, X. Wu, S.J. Maybank, General tensor discriminant analysis and gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1700–1715.
[5] D. Tao, X. Tang, X. Li, Which components are important for interactive image searching? IEEE Trans. Circuits Syst. Video Technol. 18 (1) (2008) 3–11.
[6] D. Tao, X. Tang, X. Li, Y. Rui, Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm, IEEE Trans. Multimedia 8 (4) (2006) 716–727.
[7] M. Dash, K. Scheuermann, P. Liu, Feature selection for clustering-a filter solution, in: Proceedings of IEEE International Conference on Data Mining, 2002, pp. 115–122.
[8] P. Mitra, C. Murthy, S. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2) (2002) 301–312.
[9] M. Law, M. Figueiredo, A. Jain, Simultaneous feature selection and clustering using mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 26 (9) (2004) 1154–1166.
[10] J. Dy, C. Brodley, Visualization and interactive feature selection for unsupervised data, in: Proceedings of ACM Special Interest Group on Knowledge Discovery in Data, 2000, pp. 360–364.

[11] J. Dy, C. Brodley, Feature selection for unsupervised learning, J. Mach. Learn. Res. 5 (2005) 845–889.
[12] C. Constantinopoulos, M. Titsias, A. Likas, Bayesian feature and model selection for Gaussian mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 28 (6) (2006) 1013–1018.
[13] E. Xing, M. Jordan, R. Karp, Feature selection for high-dimensional genomic microarray data, in: Proceedings of the 18th International Conference on Machine Learning, 2001, pp. 601–608.
[14] L. Yu, H. Liu, Efficiently handling feature redundancy in high-dimensional data, in: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 685–690.
[15] H. Zeng, Y.M. Cheung, Iterative feature selection in Gaussian mixture clustering with automatic model selection, in: Proceedings of the International Joint Conference on Neural Networks, 2007, pp. 2277–2282.
[16] D. Koller, M. Sahami, Toward optimal feature selection, in: Proceedings of International Conference on Machine Learning, 1996, pp. 284–292.
[17] Y.M. Cheung, A rival penalized EM algorithm towards maximizing weighted likelihood for density mixture clustering with automatic model selection, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004, pp. 633–636.
[18] Y.M. Cheung, Maximum weighted likelihood via rival penalized EM for density mixture clustering with automatic model selection, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 750–761.
[19] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Stat. Soc. B 39 (1977) 1–38.
[20] D. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, Cambridge, MA, 2003.
[21] C. Bishop, Pattern Recognition and Machine Learning, Springer, Berlin, 2006.
[22] G. Schwarz, Estimating the dimension of a model, Ann. Stat. 6 (2) (1978) 461–464.
[23] C. Wallace, P. Freeman, Estimation and inference via compact coding, J. Roy. Stat. Soc. B 49 (3) (1987) 241–252.
[24] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, Los Altos, CA, 1988.
[25] D. Newman, S. Hettich, C. Blake, C. Merz, UCI Repository of Machine Learning Databases, 1998. URL: ⟨http://www.ics.uci.edu/~mlearn/MLRepository.html⟩.

About the Author—HONG ZENG received his B.Eng. degree in Telecommunication Engineering from Nanjing University of Science and Technology and M.Eng. degree in Signal and Information Processing from Southeast University. He is currently a Ph.D. student in the Department of Computer Science of the Hong Kong Baptist University. His main research interests include clustering, feature selection, and related applications.

About the Author—YIU-MING CHEUNG received the Ph.D. degree from the Department of Computer Science and Engineering at the Chinese University of Hong Kong in 2000. Currently, he is an Associate Professor at the Department of Computer Science in Hong Kong Baptist University. His research interests include machine learning, information security, signal processing, pattern recognition, and data mining. Dr. Cheung is the founding and present chair of the Computational Intelligence Chapter in IEEE Hong Kong Section. He is a senior member of IEEE and ACM.