Egyptian Informatics Journal (2013) **14**, 79–88

Cairo University

**Egyptian Informatics Journal**

www.elsevier.com/locate/eij
www.sciencedirect.com

ORIGINAL ARTICLE

# On determining efficient finite mixture models with compact and essential components for clustering data

**Ahmed R. Abas** *

*Department of Computer Science, College of Computer in Leith, Umm Al-Qura University, Makka Al-Mukarrama, Saudi Arabia*
*Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig, Egypt*

Received 25 June 2012; revised 19 January 2013; accepted 11 February 2013
Available online 13 March 2013

**KEYWORDS**

Finite mixture models;
Clustering;
Model selection;
Mutual information;
Compact components

**Abstract** In this paper, an algorithm is proposed to learn and evaluate different finite mixture models (**FMMs**) for data clustering using a new proposed criterion. The **FMM** corresponds to the minimum value of the proposed criterion is considered the most efficient **FMM** with compact and essential components for clustering an input data. The proposed algorithm is referred to as the **EMCE** algorithm in this paper. The selected **FMM** by the **EMCE** algorithm is efficient, in terms of its complexity and composed of compact and essential components. Essential components have minimum mutual information, that is, redundancy, among them, and therefore, they have minimum overlapping among them. The performance of the **EMCE** algorithm is compared with the performances of other algorithms in the literature. Results show the superiority of the proposed algorithm to other algorithms compared, especially with small data sets that are sparsely distributed or generated from overlapping clusters.



## 1. Introduction

Cluster analysis is an important task in pattern recognition. It is interested in grouping similar feature vectors in an input data set into a number of clusters. Feature vectors in one cluster are similar to each other more than to other feature vectors in the other clusters. Different clustering algorithms are proposed in the literature such as the **K-means** algorithm and the **FMM** [1,2]. The **FMM** produces a certainty estimate of the membership of each feature vector to each one of the clusters in the input data set. This advantage is important for cluster analysis as it helps data analysts in interpreting clustering results. Each component in the **FMM** is usually a Gaussian distribution. Unsupervised learning of the **FMM** parameters is usually achieved via the Expectation–Maximization (**EM**) algorithm [3]. The **EM** algorithm determines the **FMM** parameters that maximize the likelihood of this **FMM** to fit the input data set. However, the **EM** algorithm has some limitations. First, it produces sub-optimal results because it

* Address: Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig, Egypt. Tel.: +966 580016325.
E-mail addresses: armohamed@uqu.edu.sa, arabas@zu.edu.eg.
Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

1110-8665 
http://dx.doi.org/10.1016/j.eij.2013.02.002

converges to the nearest local maximum of the likelihood function to the starting point. Second, it produces biased estimates for the mixture parameters when clusters are poorly separated, that is, overlapped, or when mixing weights of the mixture components have extreme values, that is, data are sparsely distributed [4]. Optimization of a **FMM** is defined as the minimization of the number of components in the **FMM** required for fitting an input data set. Optimization is a difficult problem in cluster analysis [5]. The optimum **FMM** is therefore less complex in terms of the number of its parameters, that is, it is efficient.

Several criteria are proposed in the literature for the estimation of the number of **FMM** components and hence the number of clusters assuming that each cluster is represented by a component in the **FMM**. A group of these criteria is the penalized-likelihood criteria, which include the Bayesian Information Criterion (**BIC**) [6], Bezdek's Partition Coefficient (**PC**) [7], and the Minimum Message Length (**MML**) criterion [8]. Other examples are the Information Theoretic Measure of Complexity (**ICOMP**) [9,10], the Minimum Description Length (**MDL**) criterion [11], Akaike's Information Criterion (**AIC**) [12], the Approximate Weight of Evidence (**AWE**) criterion [13], and the Evidence-Based Bayesian (**EBB**) criterion [14]. It has been shown that the **BIC/MDL** criterion performs comparably with both of the **EBB** and the **MML** criteria, and it outperforms many other criteria in the literature [14]. The Component-Wise EM (**CEM**) algorithm [15] is used with an **MML**-like criterion that is proposed [16] to estimate the number of **FMM** components. The resulting algorithm overcomes problems of the common **EM** algorithm such as obtaining sub-optimal results and approaching the boundary of the parameter space when at least one of the components becomes too small. However, due to the dependency on the **EM** algorithm, the model selected using these criteria is not necessarily the best model for clustering small data sets. The selected model does not necessarily represent well-separated clusters that are clearly associated with the model components [17].

However, although the **BIC/MDL** criterion is preferred when data clusters are separated, and the data size is large [18], and it produces a good approximation to Bayes factor [19]; it tends to overestimate the number of components when cluster shapes are not Gaussian [4]. On the other hand, it tends to underestimate the number of components when clusters are overlapping or when the number of feature vectors in the given data set is small [20]. Also, both of the **BIC** and the **MML** criteria have poor performance with sparsely distributed data [21]. Penalized-likelihood criteria compromise the goodness of fitting of the **FMM** to the input data set with the complexity of that **FMM**. Since the mixture complexity is a quadratic function of the number of features (dimensions) in the input data set, these criteria are sensitive to the increase of the number of features in the input data set. In the rest of this paper, the algorithms that use the **BIC** and the **MML** criteria for determining the number of **FMM** components are referred to as the **BIC** algorithm and the **MML** algorithm respectively.

A different group of criteria for estimating the number of **FMM** components is based on the mutual information theory. Based on the Bayesian–Kullback Ying–Yang learning theory [22], a criterion is proposed [23] and used in determining the number of **FMM** components [5]. However, due to the dependency on the **EM** algorithm for learning mixture model parameters, this criterion has the same drawbacks of the penalized-likelihood criteria. Therefore, this criterion produces inaccurate results with small data sets [5]. It includes Data Entropy that is used to evaluate different mixture models with different number of components [24]. However, this criterion may overestimate the number of components in the presence of outliers because it is biased toward producing separated components. Based on the mutual information theory, another algorithm is proposed [20]. However, this algorithm removes the largest component that is overlapping with other small components in the **FMM**. This produces inaccurate cluster structure that is obtained from the resulting **FMM** because large components in the **FMM** are supported by the data more than small components. In addition, deleting large components in the **FMM** causes high losses in the likelihood function. This algorithm underestimates the number of mixture components when some clusters are poorly separated. A different algorithm based on mutual information theory is proposed [25]. However, this algorithm has initialization problem due to starting with small number of components in the mixture model. This algorithm has satisfactory results only when the size of the input data set is large as reported by the authors. With sparse data sets and other data sets containing overlapping clusters, this algorithm underestimates the number of mixture components due to the use of the histogram method for density estimation. A Bayesian Ying–Yang (**BYY**) scale-incremental **EM** algorithm is proposed [26]. However, this algorithm has initialization problem due to starting with small number of components in the mixture model and using the **BYY** harmony function as a stopping criterion that depends on the estimated values of mixture parameters via the **EM** algorithm. With data sets that are sparsely distributed and generated from overlapping clusters, this algorithm underestimates the number of mixture components because the **BYY** harmony function is biased toward producing well-separated clusters of nearly equal size. Recently, an algorithm based on the mutual information theory, called Tuned Mutual Information (**TUMI**) algorithm, is proposed [21]. This algorithm overcomes problems of the algorithms that use the penalized-likelihood or the mutual information criteria [21]. However, the **TUMI** algorithm contains parameters that need empirical adjustment. Also, it uses a heuristic condition based on the change in the likelihood function in selecting the optimal **FMM**. Finally, the **TUMI** algorithm does not have a criterion to evaluate the resulting **FMM** from a specific initialization point in the data space. Therefore, different results may be obtained using different initialization points in the data space, and the optimal number of components of the **FMM** is considered the average of the number of components of the resulting **FMMs** [21].

Different criteria for estimating the number of **FMM** components include Adaptive Mixtures algorithm that is a recursive form of the **EM** algorithm [27]. This algorithm may overestimate the number of components when the given data set contains sparsely distributed data [20]. Also, it may underestimate the number of components when some clusters in the data space are poorly separated [21]. In addition, this algorithm does not have a measure that compromises the increase in the **FMM** complexity with the goodness of fitting of that model to the given data. A cross-validated likelihood criterion is proposed to estimate the number of components in the **FMM** using large data sets [28]. However, this criterion re-

quires a large data set and a sufficient range of the number of components. In addition, it may overestimate the number of components when the given data set is sparse [21]. Statistical tests are proposed to estimate the number of components in the **FMM** [29]. However, the output of these tests depends on a threshold that controls the decision of splitting non-Gaussian-shape components. In addition, these tests are sensitive to the outliers in the given data set [29]. Finally, these tests do not compromise fitting the mixture model to the given data set with the complexity of this model. An algorithm that uses Markov-Chain Monte Carlo (**MCMC**) sampling to explore the space of different model sizes is proposed to estimate the optimum number of components in the **FMM** according to an entropy-based measure [30]. However, this algorithm may stop at a local minimum of the entropy function resulting in a model that is not the optimal one [31]. In addition, this algorithm requires large number of computations similar to the Bayesian algorithms [32], and therefore, it is not practical [15,31]. A Rival Penalized Expectation–Maximization (**RPEM**) algorithm is proposed to learn the model parameters via maximizing a weighted likelihood [33,34]. This algorithm forces the components in a **FMM** to compete each other such that the parameters of the winner component are updated to adapt to an input feature vector and all rivals' parameters are penalized with the strength proportional to the corresponding posterior probabilities. Therefore, some components in a **FMM** fade out during the learning process. However, determining the optimal number of clusters depends on the number of components with large weights in the resulting **FMM**. This number may change for different runs of the algorithm on the same data set due to the sensitivity of the **EM** algorithm to the initialization and the change in the order of presentation of the input data feature vectors to the algorithm. The algorithm does not have a criterion to evaluate its different results in order to point out the optimal **FMM** for the input data set. In addition, this algorithm assumes that each feature vector in the input data is generated only from one component in the **FMM**, which contradicts the main assumption of the **FMM** that feature vectors of the input data are generated from all its components with different probabilities. Therefore, clustering results of the resulting **FMM** are inaccurate especially when data are generated from partially overlapping clusters.

In this paper, a new algorithm that is referred to as the **EMCE** algorithm is proposed to integrate the unsupervised learning and the optimization of the **FMM**. It learns and evaluates different **FMM**s for clustering an input data set using a new proposed criterion that is referred to as the **EMCE** criterion. The **FMM** corresponds to the minimum value of the **EMCE** criterion is considered the most efficient **FMM**, in terms of its complexity, that is composed of compact and essential components for clustering the input data set. This **FMM** has the minimum number of components, which have the minimum within-component variation and the least mutual information, that is, redundancy among them. The rest of this paper is organized as follows: Section 2 presents the proposed **EMCE** criterion and the proposed **EMCE** algorithm. Section 3 presents a comparison study of the **EMCE** algorithm and other algorithms in the literature such as the **TUMI**, the **MML**, and the **BIC** algorithms in determining the optimal number of **FMM** components and learning their parameters for clustering an input data set. Section 4 presents the conclusions and the future work.

## 2. The proposed EMCE algorithm

The **EMCE** algorithm uses a new proposed **EMCE** criterion. This criterion is based on the theory that the best cluster structure for a given data set should have dense and well-separated clusters that have the minimum number of parameters to be estimated. To realize this theory, the **EMCE** criterion selects the cluster structure that minimizes the within-cluster variations, the mutual information among clusters and the product of the relative weights of clusters in representing the given data. To introduce the notation, let $\mathbf{D} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$ be a given data set that consists of $n$ feature vectors that are independently and identically distributed in $d$-feature space. The values on each feature are scaled such that they range from 0 to 1. This reduces the sparsity of the data and increases the accuracy of estimating its cluster structure [35]. The cluster structure is revealed using the **FMM** that fits the input data set. Each component in this model is assumed to represent a cluster in the input data set. The components of the mixture model have non-restricted Gaussian distributions. Then, using a mixture model $M_k$ that contains $k$ components, the density function of this data set is defined as:

$$p(\mathbf{x}) = \sum_{i=1}^{k} p(\mathbf{x}|\boldsymbol{\theta}_i) P(\boldsymbol{\theta}_i) \tag{1}$$

where $\mathbf{x} \in \mathbf{D}$ and $\boldsymbol{\theta}_i$ are the set of parameters that define the mean, the covariance matrix, and the mixing weight of the $i$th component in $M_k$, that is, $\theta_i = \{\mu_i, \boldsymbol{\Sigma}_i, P(i)\}$, where $i = 1{:}k$. This density function is redefined as:

$$p(\mathbf{x}) = \sum_{i=1}^{k} f_i(\mathbf{x}) \tag{2}$$

where $f_i(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i)$. This equation shows that the mixture model can be regarded as the summation of $k$ sub-density functions. Based on the general definition of the mutual information [2,21], the mutual information between two sub-density functions $f_i$ and $f_j$ in $M_k$ is defined as:

$$I(f_i; f_j) = \sum_{\mathbf{x}\in\mathbf{D}} \sum_{\mathbf{y}\in\mathbf{D}} r(\mathbf{x}, \mathbf{y}) \log_2 \frac{r(\mathbf{x}, \mathbf{y})}{f(\mathbf{x}) f(\mathbf{y})} \tag{3}$$

where $r(\mathbf{x}, \mathbf{y})$ is the joint distribution of finding $\mathbf{x}$ and $\mathbf{y}$ feature vectors. The mutual information measures how much two distributions differ from statistical independence. Since $\mathbf{x}$ and $\mathbf{y}$ are conditionally independent, the value of $r(\mathbf{x}, \mathbf{y})$ can be determined as:

$$r(\mathbf{x}, \mathbf{y}) = [f_i(\mathbf{x}) + f_j(\mathbf{x})][f_i(\mathbf{y}) + f_j(\mathbf{y})] \tag{4}$$

From Eqs. (3) and (4), it is easy to notice that when two sub-density functions represent two statistically independent distributions, the mutual information between them is zero, otherwise it is greater than zero. The mutual information between a certain sub-density function $f_i$ and the rest of the mixture model $M_k$ is then defined as:

$$I(f_i; M_k - f_i) = \sum_{f_j \in M_k - f_i} I(f_i; f_j) \tag{5}$$

Let the total mean of the given data set be m, where $\mathbf{m} = \sum_{i=1}^{k} \boldsymbol{\mu}_i / k$. The covariance matrix $\boldsymbol{\Sigma}_T$ can be decomposed into the within and the between-clusters covariance matrices as follows:

$$\boldsymbol{\Sigma}_T = \boldsymbol{\Sigma}_W + \boldsymbol{\Sigma}_B \tag{6}$$

The within-clusters covariance matrix $\boldsymbol{\Sigma}_W$ is defined as follows:

$$\boldsymbol{\Sigma}_W = \sum_{i=1}^{k} \boldsymbol{\Sigma}_i \tag{7}$$

The between-clusters covariance matrix $\boldsymbol{\Sigma}_B$ is defined as follows:

$$\boldsymbol{\Sigma}_B = \frac{1}{k-1} \sum_{i=1}^{k} (\boldsymbol{\mu}_i - \mathbf{m})(\boldsymbol{\mu}_i - \mathbf{m})^T \tag{8}$$

Then, the proposed **EMCE** criterion for evaluating different mixture models and determining the most **e**fficient **m**odel with compact and **e**ssential components corresponds to the minimum criterion value is as follows:

$$EMCE = \frac{|\Sigma_W|}{|\Sigma_T|} + \frac{\sum_{i=1}^{k} I(f_i; M_k - f_i)}{\sum_{i=1}^{k_{\max}} I(f_i; M_{k_{\max}} - f_i)} + \frac{\left(1/\prod_{i=1}^{k} P(i)\right)}{\left(1/\prod_{i=1}^{k_{\max}} P(i)\right)} \tag{9}$$

Minimizing the first term in the **EMCE** criterion produces the minimum within-component variation, which in turn results in the most compact components of the mixture model. In addition, minimizing the second term produces the minimum mutual information among components, that is, the mixture model is composed of essential and well-separated components. Finally, minimizing the third term in the **EMCE** criterion produces the minimum product of component mixing weights, that is, the minimum number of components in the mixture model that have approximately equal weights. This in turn results in the most efficient model in terms of its complexity. Therefore, minimizing the **EMCE** criterion produces the most efficient mixture model with compact and essential components to represent the given data set. To achieve optimality of the resultant **FMM,** the three terms of the **EMCE** criterion are made percentages to make them comparable and allow for the best compromise between them. Starting with a mixture model with $k_{\max}$ components, the first term of the **EMCE** criterion will be too small while each one of the other terms evaluates to one. As $k$ decreases, the first term increases because the sizes of the components increase, while the other two terms decrease because components become more separated, and their mixing weights become larger. The minimum **EMCE** criterion value assures the best compromise between compactness of the mixture components, mutual information among them, that is, essentiality and their number, that is, mixture complexity.

The **EMCE** algorithm uses both the random parameter initialization and the **CEM** algorithm [16] in order to reduce the effect of obtaining sub-optimal results or approaching the boundary of the parameter space while learning the **FMM** parameters. The algorithm starts with a mixture model with large number components $k_{\max}$ that is twenty in the experiments shown in this paper. The **CEM** algorithm is used to estimate parameters of the mixture model. After convergence of the **CEM** algorithm, the **EMCE** criterion value of the current **FMM** is computed. The component that has the smallest mixing weight in the **FMM** is considered unnecessary. Therefore, this component can be deleted from the **FMM**. Parameters of the new **FMM** are computed by the **CEM** algorithm. This process continues until there is one component in the model. Parameters of the **FMM** components are estimated in every iteration of the **CEM** algorithm in an ascending order according to their mixing weights. This allows small components to survive and reduces the likelihood that a large component absorbs small neighboring ones. Finally, the mixture model that has the minimum **EMCE** criterion value is considered the optimal mixture model for the input data set, and its number of components is considered the optimal number of clusters from which the input data set is generated. This mixture model has the minimum number of components that have the minimum within-component variation and the minimum mutual information among its components. In other words, it is efficient because it has a small number of parameters, that is, small complexity, and its components are compact and essential, that is, not redundant. Finally, the steps of the proposed **EMCE** algorithm are shown as follows.

Program model = EMCE (data)

**Step 0**. Normalize the values of each input data feature to range from 0 to 1.

**Step 1**. The mutual information among components of the **FMM** and the number of these components should be minimized and the within-component variation should be minimally increased during optimization.

**Step 2**. Start with a mixture model $M$ that has a large number of components $k_{\max}$.

**Step 3**. Sort the mixture components in an ascending order according to their mixing weights.

**Step 4**. Use the **CEM** algorithm to learn parameters of $M$.

**Step 5**. Compute the **EMCE** criterion value (see, Eq. (9)) for the mixture model $M$.

**Step 6**. If **Bestmodel** is Empty then:

- Save the current mixture model $M$ as **Bestmodel.**
- Save the current **EMCE** criterion value as **BestEMCE.**

**Step 7**. If the current **EMCE** criterion value < **BestEMCE** then:

- Save the current mixture model $M$ and the current **EMCE** criterion value as **Bestmodel** and **BestEMCE**, respectively.

**Step 8**. If $k == 1$ then Go To **Step 9**. Else:

- Delete from $M$ the component $f_\alpha$ that has the minimum mixing weight.
- Decrement $k$.
- Adjust the mixing weights of the other components in $M$ such that their summation is unity.
- Go To **Step 3**.

**Step 9**. Assign the **Bestmodel** that corresponds to the minimum **EMCE** criterion value to the optimal **FMM** for the input data set.

**Step 10**. Stop.

## 3. Experimental results and discussion

The performance of the **EMCE** algorithm is compared to the performances of the **TUMI**, the **MML**, and the **BIC** algorithms in determining the optimal number of components in a **FMM** that is used for data clustering and learning parameters of these

components using different data sets. All algorithms are implemented, and experiments are carried out using the MATLAB software. Data sets used are described in Section 3.1. The method of initialization and the convergence condition of the **EM** algorithm are described in Section 3.2. The measure used to quantify how good the clustering results obtained from the resulting **FMM** from each algorithm is described in Section 3.3. Results of experiments and their discussion are shown in Section 3.4.

### 3.1. Data sets

Data sets used in the experiments shown in this paper have different types of cluster separation and different numbers of features. These data sets are described as follows:

#### 3.1.1. The Iris data set

This data set is commonly used in classification analysis [36]. It consists of 150 feature vectors each of which is a vector in four-feature space. These feature vectors represent three clusters of equal sizes. Two clusters are overlapped in the data space. The purpose of using this data set is to test the algorithms compared when data clusters are poorly separated and when the number of features is small.

#### 3.1.2. The Wine data set

This data set is also commonly used in classification analysis [36]. It consists of 178 feature vectors each of which is a vector in 13-feature space. These feature vectors represent three clusters whose sizes are 59, 71, and 48 feature vectors. The clusters are separable in the data space. The purpose of using this data set is to test the algorithms compared when data clusters are separated and when the number of features is large compared to the number of feature vectors.

#### 3.1.3. The third data set

This data set is artificially generated such that it consists of 90 feature vectors each of which is a vector in 10-feature space. These feature vectors are generated from three poorly separated Gaussian-shape clusters with equal probabilities. The centers of these clusters are $\boldsymbol{\mu}_1 = [0,0,0,0,0,0,0,0,0,0]^T$, $\boldsymbol{\mu}_2 = [-2,-2,-2,-2,-2,-2,-2,-2,-2,-2]^T$, and $\boldsymbol{\mu}_3 = [2,2,2,2,2,2,2,2,2,2]^T$, while their covariance matrices are identical and equal to $\boldsymbol{\Sigma} = \mathbf{I}_{10}$. The purpose of using this data set is to test the algorithms compared when data clusters are poorly separated and when the number of features is large compared to the number of feature vectors, that is,, the data set is sparsely distributed.

#### 3.1.4. The fourth data set

This data set is artificially generated such that it consists of 150 feature vectors each of which is a vector in 10-feature space. These feature vectors are generated from five separated Gaussian-shape clusters with equal probabilities. The centers of these clusters are $\boldsymbol{\mu}_1 = [2,2,2,2,2,2,2,2,2,2]^T$, $\boldsymbol{\mu}_2 = [6,2,2,2,6,6,2,2,6,6]^T$, $\boldsymbol{\mu}_3 = [2,6,6,6,2,2,6,6,2,2]^T$, $\boldsymbol{\mu}_4 = [4,4,4,4,4,4,4,4,4,4]^T$ and $\boldsymbol{\mu}_5 = [6,6,6,6,6,6,6,6,6,6]^T$, while their covariance matrices are identical and equal to $\boldsymbol{\Sigma} = 0.5\mathbf{I}_{10}$. The purpose of using this data set is to test the algorithms compared when data clusters are separated and when the number of features is large compared to the number of feature vectors, that is, the data set is sparsely distributed.

### 3.2. Initialization and convergence of the EM algorithm

In all experiments, the **EM** algorithm is initialized with a mixture model that consists of 20 Gaussian components. These components are equally weighted and they have non-restricted covariance matrices. The center locations of these components are randomly chosen from the data set. The covariance matrices of these components are initialized similarly as $\boldsymbol{\Sigma} = [(1/10d)trace(\boldsymbol{\Sigma}_T)]\mathbf{I}_d$, where $d$ is the number of features of the data set and $\boldsymbol{\Sigma}_T$ is the covariance matrix of the data set used. The convergence condition used for the **EM** algorithm is $|[LOGLH(t) - LOGLH(t-10)]/LOGLH(t-10)| < 0.01$, where $LOGLH(t)$ and $LOGLH(t-10)$ are the natural logarithm of the likelihood function at iterations $(t)$ and $(t-10)$, respectively. A Bayesian regularization method [37,38] is used to prevent the algorithm from approaching the boundary of the parameter space. This happens when at least one component of the **FMM** collapses onto one data point resulting in a singular covariance matrix for this component. A regularization term $\lambda\mathbf{I}_d$, where $\lambda$ is a regularization constant and $\mathbf{I}_d$ is the identity matrix of order $d$, is added to the update equation of the covariance matrix in the M-step of the **CEM** algorithm. In the experiments shown in this paper, $\lambda$ is set to 0.0001.

### 3.3. The evaluation criterion for FMM clustering results

The mutual information is a symmetric measure to quantify the statistical information shared between two distributions [39]. Therefore, this measure is used to quantify how good the clustering results obtained using a **FMM** for a certain data set is by comparing it to the true classification of this data set [40]. Let $\mathbf{X}$ and $\mathbf{Y}$ be two random variables represent the true class labels $[1 \ldots m]$ for a certain data set and the cluster labels $[1 \ldots k]$ resulting from a **FMM** clustering for the same data set, respectively. The mutual information between $\mathbf{X}$ and $\mathbf{Y}$ is defined as $I(\mathbf{X};\mathbf{Y}) = \sum_{i=1}^{m}\sum_{j=1}^{k} P_{ij}\log_2(P_{ij}/P_iP_j)$, where $P_{ij}$ is the probability that a member of cluster $j$ belongs to class $i$, $P_i$ is the probability of class $i$, and $P_j$ is the probability of cluster $j$. Since this measure is not bounded by the same constant for all data sets, a normalized version that ranges from 0 to 1 is proposed for easier interpretation and comparison [40]. This normalized version is called the normalized mutual information (**NMI**) and is computed as follows:

$$NMI(\mathbf{X};\mathbf{Y}) = \frac{I(\mathbf{X};\mathbf{Y})}{\sqrt{H(\mathbf{X})H(\mathbf{Y})}} \tag{10}$$

**Table 1** A comparison of the EMCE, the TUMI, the MML and the BIC algorithms in determining the number of components (clusters) in the FMM used for clustering data. The number between brackets with the name of each data set is the number of classes of this data set.

| Data | EMCE | | TUMI | | MML | | BIC | |
|---|---|---|---|---|---|---|---|---|
| | NMI | $K$ | NMI | $K$ | NMI | $K$ | NMI | $K$ |
| Iris (3) | 0.90 | 3 | 0.88 | 3 | 0.78 | 5 | 0.76 | 2 |
| Wine (3) | 0.97 | 3 | 0.04 | 1 | 0.54 | 2 | 0.73 | 2 |
| Data3 (3) | 1.00 | 3 | 0.02 | 1 | 0.00 | 1 | 0.00 | 1 |
| Data4 (5) | 1.00 | 5 | 0.00 | 1 | 0.53 | 2 | 0.00 | 1 |

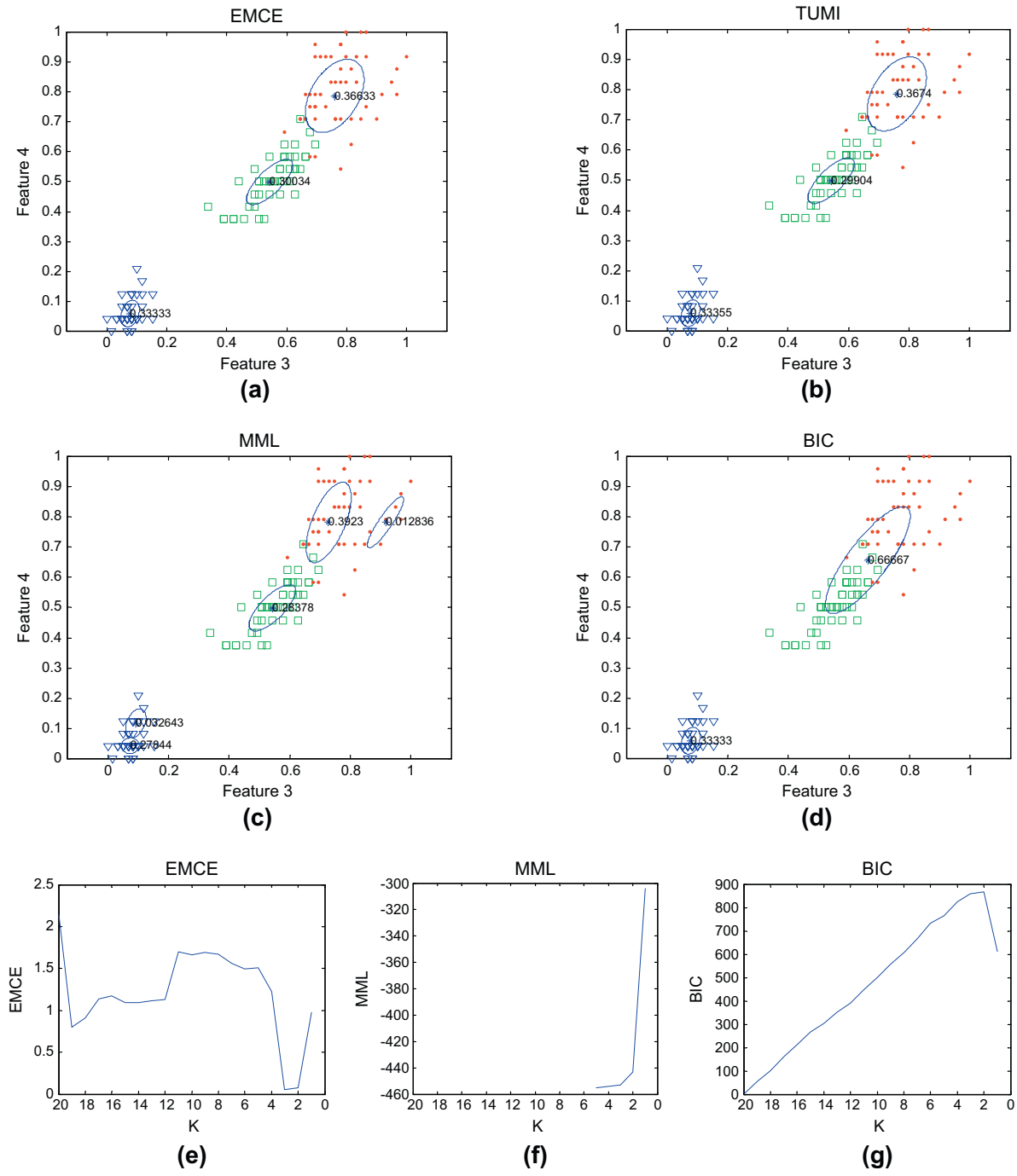


**Figure 1** The FMMs obtained from the algorithms compared and the distribution of the criteria used against the number of mixture components $k$ with the Iris data set.

where $H(\mathbf{X})$ and $H(\mathbf{Y})$ denote the entropy of $\mathbf{X}$ and $\mathbf{Y}$. The **NMI** has the value of 1 when there is a one to one mapping between the clusters obtained and the true classes (i.e., $k = m$) of a given data set. Since this measure is not biased toward large $k$, it is preferred to compare different data partitions [40,41].

### 3.4. Discussion of results

Table 1 shows the performances of the algorithms compared with each one of the data sets used. The performance of each algorithm is evaluated by the values of the **NMI** criterion and the number of **FMM** components corresponding to the optimal value of the criterion used by the algorithm resulting from 100 experiments. Each experiment has different random initialization values of the **EM** algorithm. This repetition of the experiments removes the effect of initialization values of the **EM** algorithm on the results of the algorithms [21]. Since the **TUMI** algorithm has no criterion to evaluate its results, the average values of the **NMI** criterion and the number of **FMM** components rounded to the nearest integer number are used in the comparison as shown in [21]. The shaded cells in this table rep-

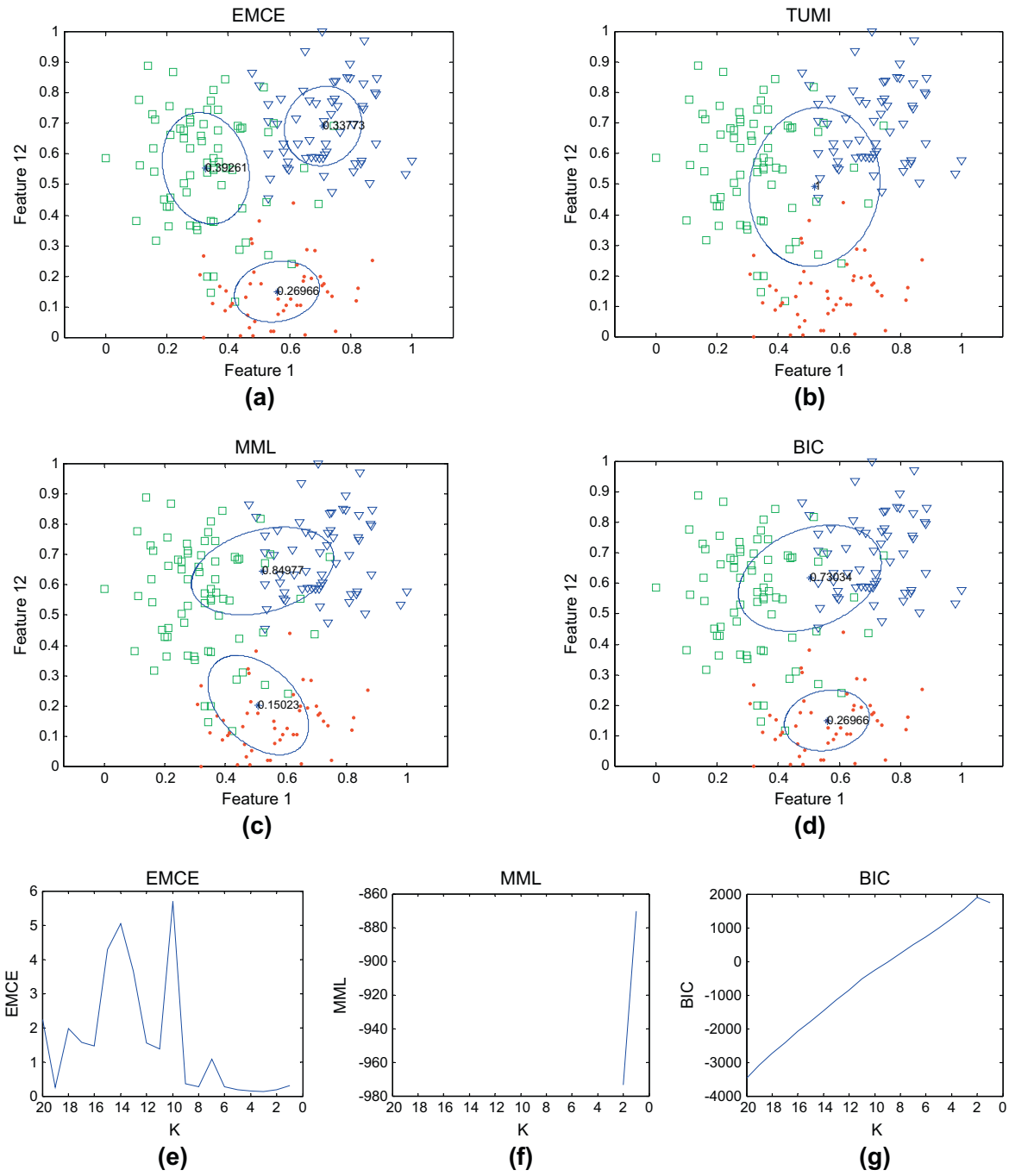


**Figure 2** The FMMs obtained from the algorithms compared and the distribution of the criteria used against the number of mixture components $k$ with the Wine data set.

resent the maximum values of the **NMI** among all algorithms and the correct number of mixture components (clusters) with each data set. Figs. 1–4(a–d) show examples of the **FMMs** obtained from the algorithms compared with each one of the four data sets used. The ellipses in these figures are isodensity curves of each component in the **FMM**. These figures (e–g) also show the distribution of the **EMCE**, **MML** and the **BIC** criteria against the number of components $k$ in the **FMM** through the runtime of the corresponding algorithms.

Table 1 and Figs. 1–4 show the superiority of the **EMCE** algorithm over other algorithms in all data sets. The **EMCE** algorithm results in the largest **NMI** criterion values and the correct number of mixture components with all data sets. These results show that the **EMCE** algorithm is less sensitive to the curse of dimensionality than all other algorithms compared. This is because the proposed **EMCE** criterion only depends on the characteristics of the **FMM** representing the input data set such as the within-component/cluster variation, the mutual information among components/clusters and the relative mixing weights of components/clusters. On the other hand, the criteria used in the other algorithms compared depend explicitly on the dimensionality of the input data set

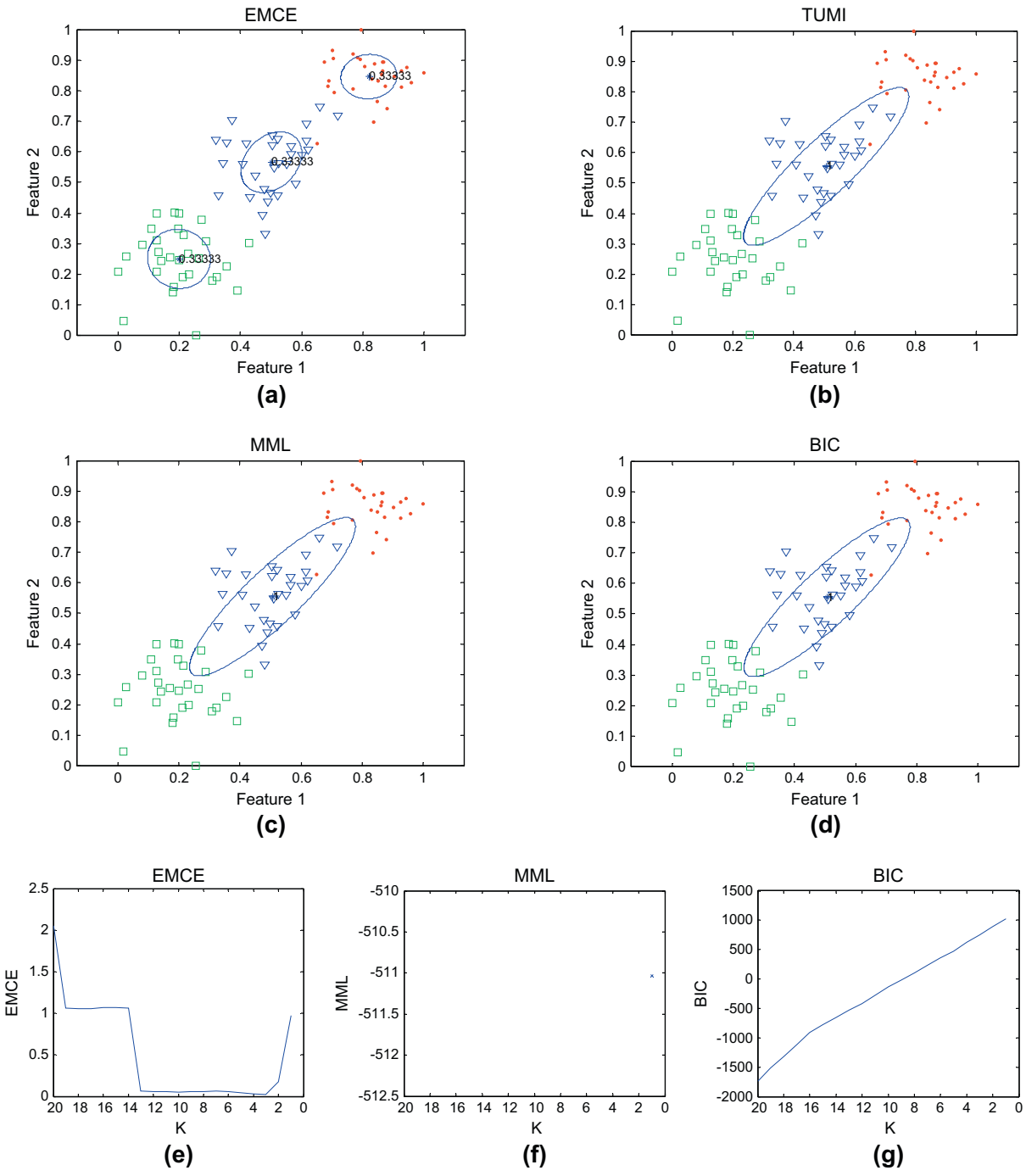


**Figure 3** The FMMs obtained from the algorithms compared and the distribution of the criteria used against the number of mixture components $k$ with the third data set.

either as a penalty to the likelihood of the **FMM** to represent the input data set as in the **MML** and the **BIC** algorithms or as a main factor in determining the density of the feature vectors in the input data set that are used in computing the likelihood of the **FMM** to represent this data set as in the **TUMI** algorithm. In addition, the penalty term in these criteria depends on the size of the data set $n$, and therefore, they underestimate the number of clusters with small-size data sets. In the **MML** algorithm, components of zero mixing weights are deleted through learning of parameters of the **FMM;** therefore, the MML/K curve does not cover the whole range of $K$ in the examples shown in Figs. 1–4. This problem results from poor initialization of **FMM** parameters that lead to the generation of empty components or clusters [43].

## 4. Conclusions and future work

In this paper, the commonly used criteria for determining the number of **FMM** components required to fit an input data set are reviewed. A new algorithm, called the efficient model with

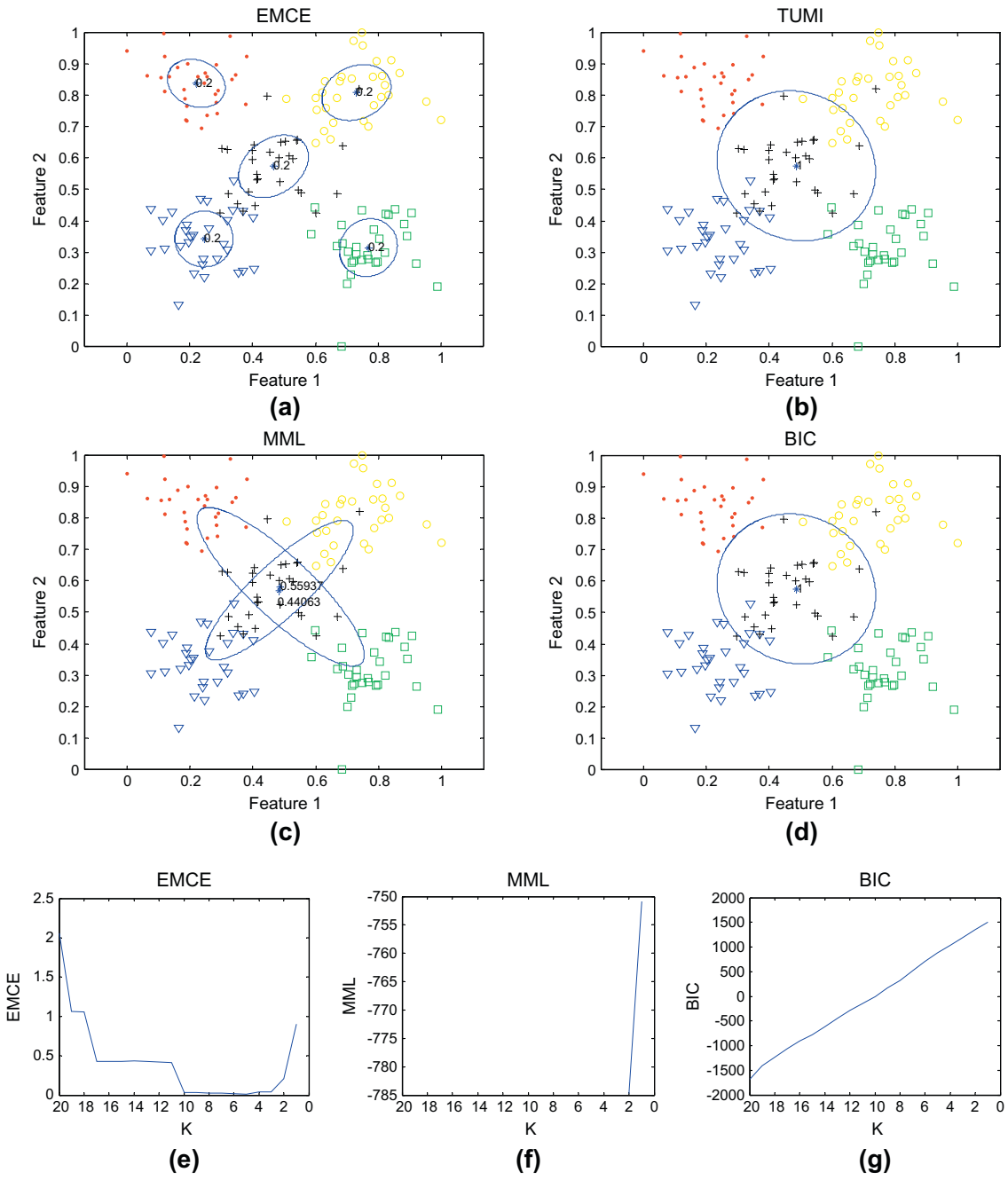


**Figure 4** The FMMs obtained from the algorithms compared and the distribution of the criteria used against the number of mixture components $k$ with the fourth data set.

compact and essential components (**EMCE**) algorithm is proposed. It is based on a new proposed model selection criterion, called the **EMCE** criterion. This algorithm overcomes problems of the algorithms that use the penalized-likelihood or the Mutual Information criteria with small and sparse data. This algorithm produces a single frame for model estimation and selection for data clustering. Empirical analysis shows that the proposed algorithm outperforms the **TUMI**, the **MML**, and the **BIC** algorithms, especially with small and sparse data that may be generated from overlapping clusters.

In the future, feature weighting may be used to reduce the effect of redundant features of the input data set. This will allow the **EMCE** algorithm to accurately handle too sparse data sets to find out their cluster structures, both the optimum numbers of clusters and cluster membership for each input feature vector. For example, the **EMCE** algorithm may be used in determining the Health Inequality structure of the world countries when applied on Health Inequality data sets [42]. These data sets contain a large number of features compared with the number of feature vectors, that is, sparse data.

## References

[1] Duda RO, Hart PE, Stork DG. Pattern classification. 2nd ed. USA: John Wiley & Sons; 2001.

[2] Webb A. Statistical pattern recognition. 2nd ed. UK: John Wiley & Sons; 2002.
[3] Dempster AP, Laird NM, Rubin DB. Maximum Likelihood from incomplete data via the EM algorithm (with discussion). J Roy Stat Soc 1977;B39:1–38.
[4] Biernacki C, Govaert G. Using the classification likelihood to choose the number of clusters. J Comput Sci Stat 1997;29(2):451–7.
[5] Guo P, Chen CLP, Lyu MR. Cluster number selection for a small set of samples using the Bayesian Ying–Yang Model. J IEEE Trans Neural Netw 2002;13(3):757–63.
[6] Schwarz G. Estimating the dimension of a model. J Ann Stat 1978;6:461–4.
[7] Bezdek J. Pattern recognition with fuzzy objective function algorithms. New York: Plenum Press; 1981.
[8] Wallace C, Freeman P. Estimation and inference via compact coding. J Roy Stat Soc 1987;B49(3):241–52.
[9] Bozdogan H. ICOMP: a new model-selection criterion. In: Bock HH, editor. Classification and related methods of data analysis. Amsterdam: North Holland Publishing Company; 1988. p. 599–608.
[10] Bozdogan H. On the information-based measure of covariance complexity and its application to the evolution of multivariate linear models. J Commun Stat – Theory Methods 1990;19(1):221–78.
[11] Rissanen J. Stochastic complexity in statistical inquiry. Singapore: World Scientific; 1989.
[12] Whindham M, Cutler A. Information ratios for validating mixture analysis. J Am Stat Assoc 1992;87:1188–92.
[13] Banfield J, Raftery A. Model-based Gaussian and non-Gaussian clustering. J Biomet 1993;49:803–21.
[14] Roberts SJ, Husmeier D, Rezek I, Penny W. Bayesian approaches to gaussian mixture modelling. J IEEE Trans Pattern Anal Mach Intell 1998;20:1133–42.
[15] Celeux G, Chrétien S, Forbes F, Mkhadri A. A component-wise EM algorithm for mixtures. J Comput Graph Stat 2001;10(4):697–712.
[16] Figueiredo M, Jain A. Unsupervised learning of finite mixture models. J IEEE Trans Pattern Anal Mach Intell 2002;24(3):381–96.
[17] Celeux G, Soromenho G. An entropy criterion for assessing the number of clusters in a mixture model. J Classif 1996;13:195–212.
[18] Cutler A, Windham MP. Information-based validity functionals for mixture analysis. In: Bozdogan H, editor. Proceedings of the first US/Japan conference on the frontiers of statistical modeling: an information approach. Netherlands: Kluwer Academic Publishers; 1994. p. 149–70.
[19] Kass RE, Wasserman L. A reference bayesian test for nested hypotheses and its relationship to the schwarz criterion. J Am Stat Assoc 1995;90(431):928–34.
[20] Yang ZR, Zwolinski M. Mutual information theory for adaptive mixture models. J IEEE Trans Pattern Anal Mach Intell 2001;23(4):396–403.
[21] Abas AR. An algorithm for unsupervised learning and optimization of finite mixture models. Egypt Inform J 2011;12(1):19–27 [March].
[22] Xu L. How many clusters?: A Ying–Yang machine based theory for a classical open problem in pattern recognition. Proc IEEE Int Conf Neural Netw 1996;3:1546–51.
[23] Xu L. Bayesian Ying–Yang machine, clustering and number of clusters. J Pattern Recogn Lett 1997;18:1167–78.
[24] Roberts S, Everson R, Rezek I. Maximum certainity data partitioning. J Pattern Recogn 1999;33:833–9.
[25] Still S, Bialek W. How many clusters? An information-theoretic perspective. J Neural Comput 2004;16(12):2483–506.
[26] Li L, Ma J. A BYY scale-incremental EM algorithm for Gaussian mixture learning. J Appl Math Comput 2008;205:832–40.
[27] Priebe CE. Adaptive mixture density estimation. J Am Stat Assoc 1994;89:796–806.
[28] Smyth P. Model selection for probabilistic clustering using cross-validated likelihood. J Stat Comput 2000;10(1):63–72.
[29] Vlassis N, Likas A, Kröse B. A multivariate kurtosis-based approach to Gaussian mixture modeling, Technical report IAS-UVA-00-04. The Netherlands: Computer Science Institute, University of Amesterdam; 2000. <http://citeseer.ist.psu.edu/vlassis00multivariate.html>.
[30] Roberts S, Holmes C, Denison D. Minimum-entropy data partitioning using reversible jump Markov chain Monte Carlo. J IEEE Trans Pattern Anal Mach Intell 2001;23:909–14.
[31] Figueiredo MAT, Leitao JMN, Jain AK. On fitting mixture models. In: Hancock E, Pellilo M, editors. Energy minimization methods in computer vision and pattern recognition. Berlin: Springer-Verlag; 1999. p. 54–69.
[32] Richardson S, Green P. On Bayesian analysis of mixtures with unknown number of components. J Roy Stat Soc 1997;B59:731–92.
[33] Cheung YM. Maximum weighted likelihood via rival penalized EM for density mixture clustering with automatic model selection. In: IEEE transactions on knowledge and data engineering, vol. 17, no. 6, June 2005. p. 750–61.
[34] Zeng H, Cheung YM. A new feature selection method for Gaussian mixture clustering. Pattern Recogn 2009;42:243–50.
[35] Rafat A. An adaptive approach for clustering incomplete data sets: an application to health inequality analysis. Germany: LAP LAMBERT Academic Publishing AG & Co. KG; 2010.
[36] UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science; March 2012. <http://archive.ics.uci.edu/ml/>.
[37] Ormoneit D, Tresp V. Improved Gaussian mixture density estimates using Bayesian penalty terms and network averaging. In: Touretzky DS, Mozer MC, Hasselmo ME, editors. Advances in neural information processing systems, vol. 8. The MIT Press; 1996. p. 542–8.
[38] Ueda N, Nakano R, Ghahramani Z, Hinton GE. SMEM algorithm for mixture models. J Neural Comput 2000;12(9):2109–28.
[39] Cover TM, Thomas JA. Elements of information theory. Wiley; 1991.
[40] Strehl A, Ghosh J. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 2002;3:583–617.
[41] Fern XZ, Brodley CE. Random projection for high dimensional data clustering: a cluster ensemble approach. In: Proceeding of the 20th international conference on machine learning (ICML 2003); 2003, p. 186–93.
[42] World Health Organization (WHO). Data and statistics, August 2010. <http://www.who.int/research/en/>.
[43] Ossama O, Mokhtar HMO, El-Sharkawi ME. An extended *k*-means technique for clustering moving objects. Egypt Inform J 2011;12:45–51.