Egyptian Informatics Journal (2011) **12**, 19–27

Cairo University

Egyptian Informatics Journal

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

ORIGINAL ARTICLE

An algorithm for unsupervised learning and optimization of finite mixture models

Ahmed R. Abas

Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig, Egypt

Received 4 May 2010; accepted 28 September 2010
Available online 22 March 2011

KEYWORDS

Finite Mixture Models;
Expectation–Maximization;
Unsupervised learning;
Clustering;
Optimization

Abstract In this paper, an algorithm is proposed to integrate the unsupervised learning with the optimization of the Finite Mixture Models (**FMM**). While learning parameters of the **FMM** the proposed algorithm minimizes the mutual information among components of the **FMM** provided that the reduction in the likelihood of the **FMM** to fit the input data is minimized. The performance of the proposed algorithm is compared with the performances of other algorithms in the literature. Results show the superiority of the proposed algorithm over the other algorithms especially with data sets that are sparsely distributed or generated from overlapped clusters.

1. Introduction

Unsupervised learning or 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 groups or clusters. Feature vectors belonging to the same cluster are similar to each other more than to other feature vectors belonging to the other clusters. Several clustering algorithms are proposed in the literature such as the **K-means** algorithm, and the **FMM** [1,2]. The **FMM** is preferred for cluster analysis because it produces a certainty estimate of the membership of each feature vector to each one of the clusters in the input data set. 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 as it 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 i.e., overlapped, or when mixing weights of the mixture components have extreme values i.e., 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

E-mail address: arabas@zu.edu.eg

Peer review under responsibility of Faculty of Computers and Information, Cairo University.
doi:10.1016/j.eij.2011.02.005

Production and hosting by Elsevier

set. Optimization is one of the most difficult problems in cluster analysis [5].

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 as examples the Bayesian Information Criterion (**BIC**) [6], the 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], the Akaike's Information Criterion (**AIC**) [12], the Approximate Weight of Evidence (**AWE**) criterion [13], and the Evidence-Based Bayesian (**EBB**) criterion [14]. Also, a new **MML**-like criterion is proposed [15] and used with the Component-Wise EM (**CEM**) algorithm [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. In other words, the selected model does not necessarily represent well-separated clusters that are clearly associated with the model components [17]. 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 **BIC/MDL** criterion has been shown to produce a good approximation to Bayes factor [18]. However, although the **BIC/MDL** criterion is preferred when data clusters are separated and the data size is large [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 overlapped or when the number of feature vectors in the given data set is small [20]. 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.

Another group of criteria for the estimation of the number of **FMM** components is based on the mutual information. This group includes Data Entropy that is used to evaluate different mixture models with different number of components [21]. However, this criterion may overestimate the number of components in the presence of outliers, as it is biased toward producing separated components. Another criterion in this group based on the Bayesian-Kullback Ying-Yang learning theory [22] is proposed [23]. This criterion is 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]. Also, an algorithm that is based on the mutual information theory is proposed [20]. However, on the opposite of the algorithms that use the penalized-likelihood criteria, this algorithm removes the largest component that is overlapped with many other small components in the **FMM**. This results in bad quality of the cluster structure obtained by 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 the likelihood function to be largely decreased. This algorithm also underestimates the number of mixture components when some clusters are poorly separated in the data space. Finally, the authors used only centers of the mixture components instead of all the data points in their definition of the mutual information between two components in the **FMM**. This may be only valid with data sets that are dense and concentrated around their cluster centers as the examples shown by the authors. In the rest of this paper, this algorithm is referred to as the Mutual Information (**MI**) algorithm. Another algorithm that is based on mutual information theory is proposed [24]. However, this algorithm has initialization problem due to starting with small number of components in the mixture model. In addition, this algorithm has satisfactory results in determining the number of mixture components that is equal to the number of clusters of the input data set only when the size of this data set is large as reported by the authors. With small data sets, especially those data sets that are sparsely distributed and generated from overlapped clusters, this algorithm underestimates the number of mixture components due to the use of the histogram method for density estimation. Recently, a Bayesian Ying–Yang (**BYY**) scale-incremental **EM** algorithm for Gaussian mixture learning for both the parameter estimation and model selection is proposed [25]. 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. In addition, with small data sets, especially those data sets that are sparsely distributed and generated from overlapped 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.

Different criteria for the estimation of the number of **FMM** components include Adaptive Mixtures algorithm that is a recursive form of the **EM** algorithm [26]. Although this algorithm does not require a range of the number of components, it 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. This results from the iterative form of the **EM** algorithm, which may generate an unnecessary component for few outliers in the data set and also may allow many components to be overlapped. In addition, the resulting model depends on the order of presenting the input data patterns to the algorithm due to the recursive nature of the algorithm. Finally, 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 [27]. However, this criterion requires not only a large data set in order to be divided into training and test data but also a sufficient range of the number of components. In addition, the selected model is not necessarily the optimum

one for clustering in terms of cluster separation and model complexity. Therefore, it may overestimate the number of components when the given data set is sparsely distributed. Algorithms that use statistical tests are proposed to estimate the number of components in the **FMM** [28]. However, the output of these algorithms depends on a user-defined threshold that controls the decision of splitting non-Gaussian shape components. In addition, the statistical tests used in these algorithms are sensitive to the outliers in the given data set [28]. Finally, these algorithms do not compromise fitting the mixture model to the given data set with the complexity of this model. Finally, 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 [29]. However, this algorithm may stop at a local minimum of the entropy function resulting in a model that is not the optimum one because of a large potential barrier between this model and the next one that has less data entropy [30]. In addition, this algorithm requires as large number of computations as the Bayesian algorithms (see for example, [31]) due to the use of the MCMC sampling. Therefore, these algorithms are impractical for many pattern recognition applications [30,15].

In this paper, an algorithm is proposed to determine both the number of components in the **FMM** and its parameters for fitting an input data set that may be sparsely distributed or generated from overlapped clusters. As it learns the **FMM** parameters the proposed algorithm minimizes the mutual information among components of the **FMM** while keeping the reduction in the likelihood of this **FMM** to fit the input data minimum. The rest of this paper is organized as follows: Section 2 presents an algorithm that is proposed to integrate the unsupervised learning and the optimization of the **FMM** using data sets that may be sparsely distributed or contain overlapped clusters. Section 3 presents a comparison study of the proposed algorithm and other algorithms such as the **MI**, the **MML**, and the **BIC** algorithms based on their results in clustering the input data and determining the number of **FMM** components. Section 4 presents the conclusions and the future work.

2. The proposed algorithm

The steps of the proposed algorithm, named the **TU**ned **M**utual **I**nformation theory (**TUMI**) algorithm in the rest of this paper, are shown in Fig. 1. The **TUMI** algorithm uses both random parameter initialization and the **CEM** algorithm [15] in order to reduce the effect of obtaining sub-optimal results or approaching the boundary of the parameter space while learning the **FMM** parameters. After the convergence of the **CEM** algorithm the mutual information between each component and the rest of the **FMM** components is computed as explained in Section 2.1. The component that has the smallest mixing weight in the **FMM** and positive mutual information with the rest of the **FMM** components is considered unnecessary. Therefore, this component can be deleted from the **FMM** provided that the rate of decrease in the likelihood function due to this deletion is less than a certain threshold value that is defined in Section 2.2. The threshold value can be used to tune the **TUMI** algorithm to allow some overlap among the **FMM** components. Parameters of the **FMM** components are estimated in each 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.

2.1. Computing the mutual information

To introduce the notation, let **D** be a given data set that consists of n feature vectors that are independently and identically distributed in d-feature space. 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 $\boldsymbol{\theta}_i$ is the set of parameters that define the centre and the covariance matrix of the i-th component in M_k. This density function is redefined as:

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

Program model =TUMI(data)
Step 0. Normalize the values of each input data feature to range from 0 to 1.
Step 1. The mutual information in the **FMM** has to be minimized and the total likelihood has to be minimally reduced during the optimization of the **FMM**.
Step 2. Start with a mixture model M that has a large number of components.
Step 3. Sort the mixture components in an ascending order according to their mixing weights.
Step 4. Use the **CEM** algorithm to estimate the parameters of M.
Step 5. Compute the percentage of change in the likelihood function due to the removal of the last component deleted from M.
Step 6. If the percentage of change in the likelihood function $\geq$ the tuning threshold then select M before deleting the last component and go to Step 11.
Step 7. Compute the quantities $I(f_i; M - f_i)$ for all $f_i \in M$.
Step 8. If $I(f_i; M - f_i) \leq 0$, $\forall f_i \in M$ then select M and go to Step 11.
Step 9. Delete from M the component f_α that has the minimum mixing weight and $I(f_\alpha; M - f_\alpha) > 0$.
Step 10. Adjust the mixing weights of the other components in M such that their summation is unity, and go to Step 3.
Step 11. Assign the selected model to the optimum FMM for the input data set.
Step 12. Stop.

Figure 1 Steps of the **TUMI** algorithm.

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] the mutual information between two sub-density functions f_i and f_j in M_k is defined as:

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

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:

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

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:

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

2.2. Tuning the *TUMI* algorithm

The proposed algorithm can be tuned to allow mixture components to be overlapped to some extent. A heuristic is proposed to tune the proposed algorithm. To define this heuristic let the percentage of change in the likelihood function due to the deletion of a component from the mixture model M_k be $dec(k)$, which is defined as:

$$dec(k) = \frac{\log(p(D|M_k)) - \log(p(D|M_{k-1}))}{\log(p(D|M_k))} \quad (6)$$

After a short burn in stage, in which four components are deleted from the mixture model, four percentage values of the likelihood change will be obtained. Then, until the last component is deleted the next smallest component that has a positive mutual information with the rest of the mixture model M_r can be deleted only if $dec(r) < avg(dec(k{:}r+1)) + 3std(dec(k{:}r+1))$, where ***avg*** and ***std*** denote the average and the standard deviation. Since the **TUMI** algorithm is independent of the number of mixture parameters it is less sensitive to the number of features in the input data set than the algorithms that use penalized-likelihood criteria. Therefore, it can handle sparse data sets more accurately than these algorithms. In addition, tuning the mutual information theory allows the **TUMI** algorithm to fit data sets that are generated from overlapped clusters more accurately than other algorithms that are based on information theory without tuning.

3. Experimental results and discussion

Experiments are carried out to compare the performances of the **TUMI**, the **MI**, the **MML**, and the **BIC** algorithms in clustering and determining the number of the **FMM** components. All algorithms are implemented and experiments are carried out using the MATLAB software package. Data sets used are described in Section 3.1. The method of initialization and the convergence conditions of the **EM** algorithm are described in Section 3.2. The measure used to quantify how good the clustering results obtained by a clustering algorithm is described in Section 3.3. Results of experiments and their discussion are shown in Section 3.4.

3.1. Data sets

The data sets used 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 statistical experiments since it is used in [32]. 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 Second data set

This data set is artificially generated such that it contains 150 feature vectors each of which is a vector in four-feature space. Each feature vector is generated with equal probabilities from three separated Gaussian-shape clusters. The center of these clusters are $\boldsymbol{\mu}_1 = [2222]^T$, $\boldsymbol{\mu}_2 = [2262]^T$ and $\boldsymbol{\mu}_3 = [2226]^T$. The covariance matrices of these clusters are identical and equal to $\sum = 0.5\,\mathbf{I}_4$, where $\mathbf{I}_4$ is the identity matrix of order four. 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 small.

3.1.3. The Third data set

This data set is artificially generated such that it contains 200 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 probabilities 0.5, 0.25, and 0.25, respectively. 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 $\sum = \mathbf{I}_{10}$. 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 (i.e., the data set is sparsely distributed).

3.1.4. The Fourth data set

This data set is artificially generated such that it contains 200 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 centres of these clusters are $\boldsymbol{\mu}_1 = [0,0,0,0,0,0,0,0,0,0]^T$, $\boldsymbol{\mu}_2 = [6,2,2,2,6,6,2,2,6,6]^T$, $\boldsymbol{\mu}_3 = [2,6,6,6,2,2,6,6,2,2]^T$, $\boldsymbol{\mu}_4 = [4,4,4,4,4,4,4,4,4,4]^T$ and $\boldsymbol{\mu}_5 = [6,6,6,6,6,6,6,6,6,6]^T$, while their covariance matrices are identical and equal to $\sum = 0.5\mathbf{I}_{10}$. 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.

3.2. Initialization and convergence of the EM algorithm

In all experiments, the **EM** algorithm is initialized with a mixture model that consists of 30 Gaussian components. These

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

Data	TUMI				MI				MML				BIC			
	NMI		K		NMI		K		NMI		K		NMI		K	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Iris (3)	0.86	0.07	*3.16*	0.75	0.38	0.38	1.52	0.52	*0.87*	0.05	*3.39*	0.79	0.76	0.00	2.00	0.00
Data2 (3)	*0.91*	0.06	*3.20*	0.77	0.27	0.34	1.50	0.64	*0.91*	0.05	*3.29*	0.76	0.79	0.13	2.44	0.50
Data3 (3)	*0.80*	0.37	*2.64*	0.77	0.00	0.00	1.01	0.10	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00
Data4	*0.99*	0.04	*4.94*	0.34	0.05	0.17	2.27	4.31	0.77	0.01	3.00	0.00	0.02	0.09	1.03	0.17

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 $\sum = [(1/10d)trace(\sum_T)]\mathbf{I}_d$, where d is the number of features of the data set, and $\sum_T$ is the covariance matrix of the data features. The condition of convergence is $|LOGLH(t) - LOGLH(t-10)| < 0.001$, 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 [33,34] 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 λ 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 of this paper λ is set to 0.0001.

3.3. The evaluation criterion

The mutual information is a symmetric measure to quantify the statistical information shared between two distributions [35]. Based on this fact this measure is used to quantify how good the clustering results obtained by a clustering algorithm for a certain data set is by comparing it to the true classification of this data set [36]. Let **x** and **y** be two random variables represent the true class labels [1...m] for a certain data set and the cluster labels [1...k] resulting from a clustering algorithm for the same data set respectively. The mutual information between **x** and **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 1.0 is proposed for easier interpretation and comparison [36]. This normalized version is called the Normalized Mutual Information (**NMI**) and is computed as:

$$\mathbf{NMI}(\mathbf{X};\mathbf{Y}) = \frac{I(\mathbf{X};\mathbf{Y})}{\sqrt{H(\mathbf{X})H(\mathbf{Y})}} \tag{7}$$

where $H(\mathbf{X})$ and $H(\mathbf{Y})$ denote the entropy of **X** and **Y**. The **NMI** has the value of 1.0 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 [36,37].

Table 2 A comparison of the **TUMI**, the **MI**, the **MML**, and the **BIC** algorithms using the Student's t-test statistic with different variances.

Data	(TUMI, MI)		(TUMI, MML)		(TUMI, BIC)	
	P	***T***	***P***	***T***	***P***	***T***
Iris	*0.00*	12.43	0.71	−0.37	*0.00*	14.65
Data2	*0.00*	18.25	0.78	−0.29	*0.00*	8.14
Data3	*0.00*	21.22	*0.00*	21.23	*0.00*	21.23
Data4	*0.00*	54.87	*0.00*	55.73	*0.00*	99.01

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 average and the standard deviation of both the **NMI** criterion value and the number of **FMM** components determined by the algorithm 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. The shaded cells in this table represent the maximum values of the average **NMI** among all algorithms and the correct number of mixture components (clusters) after rounding to the nearest integer number with each data set. Table 2 shows comparisons of the **TUMI** algorithm with the other algorithms compared using the Student's t-test statistic with different variances for each one of the data sets used. The ***P*** value is the significance and the ***T*** value is the t-statistic. This test examines the statistical significance of the difference in performance of a pair of algorithms using their **NMI** criterion values obtained from 100 experiments each of which has a different random initialization of the **EM** algorithm. The shaded cells in this table represent the cases in which the difference in performance of a certain pair of algorithms is statistically significant according to the 5% significance level. Figs. 2–5 show representative examples of the **FMM**s obtained from the algorithms compared with each one of the four data sets used. In each figure, the ellipses are isodensity curves of each component in the **FMM**.

Table 1 shows that the **TUMI** and the **MML** algorithms are approximately similar and they are better than the **MI** and the **BIC** algorithms with the Iris and the Second data sets that are non-sparsely distributed. Examples are shown in Figs. 2 and 3. With these data sets the **TUMI** and the **MML** algorithms result in the largest **NMI** criterion values and the correct number of mixture components. The results also show that the **TUMI**

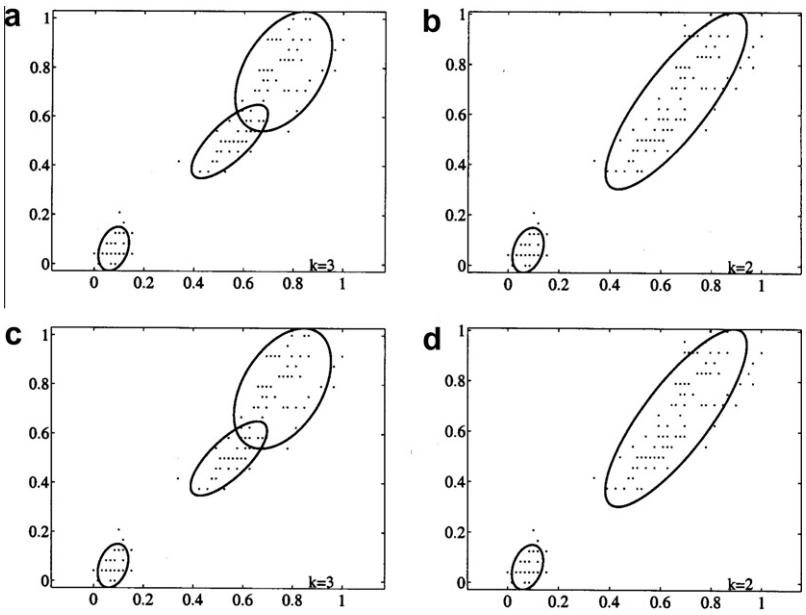

Figure 2 Examples of the **FMM**s obtained from: (a) the **TUMI**; (b) the **MI**; (c) the **MML**; and (d) the **BIC** algorithms for the Iris data set. Results are shown in the subspace that consists of the third and the fourth features of the data set.

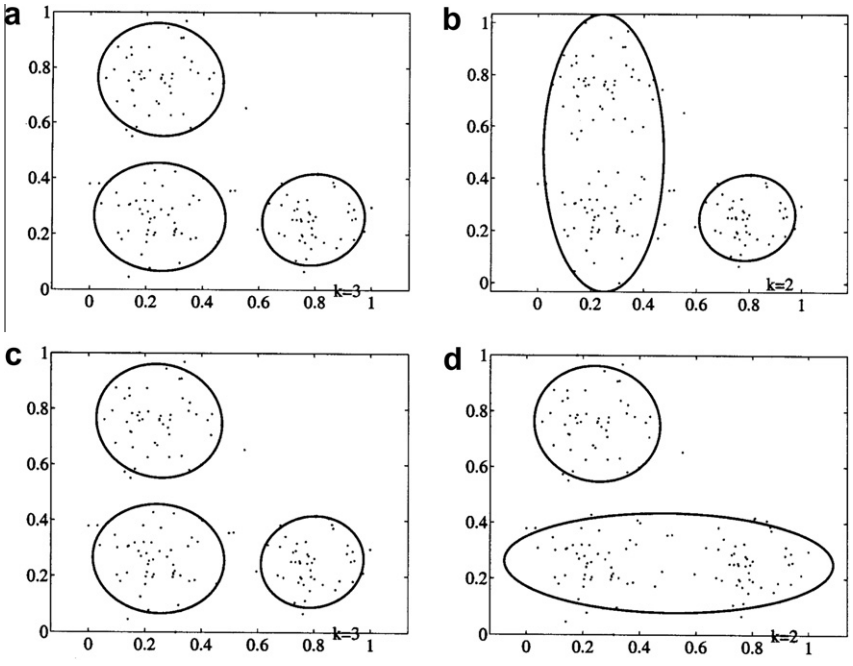

Figure 3 Examples of the **FMM**s obtained from: (a) the **TUMI**; (b) the **MI**; (c) the **MML**; and (d) the **BIC** algorithms for the Second data set. Results are shown in the subspace that consists of the third and the fourth features of the data set.

algorithm is the best with the Third and the Fourth data sets that are sparsely distributed. Examples are shown in Figs. 4 and 5. With these data sets the **TUMI** algorithm results in the largest **NMI** criterion values and the correct number of mixture components. Table 2 shows that the performance of the **TUMI** algorithm outperforms (***T***-values are positive) and it is statistically different from the performances of the **MI** and the **BIC** algorithms in all data sets. The results also show that the performance of the **TUMI** algorithm outperforms and it is statistically different from the performance of the **MML** algorithm in the last two data sets.

These results show that the performance of the **TUMI** algorithm is better than the performances of the **BIC** and the **MML** algorithms with sparsely distributed data sets. This is because it is not as sensitive to the curse of dimensionality as both algorithms do. It is also shown that the performance of the **TUMI**

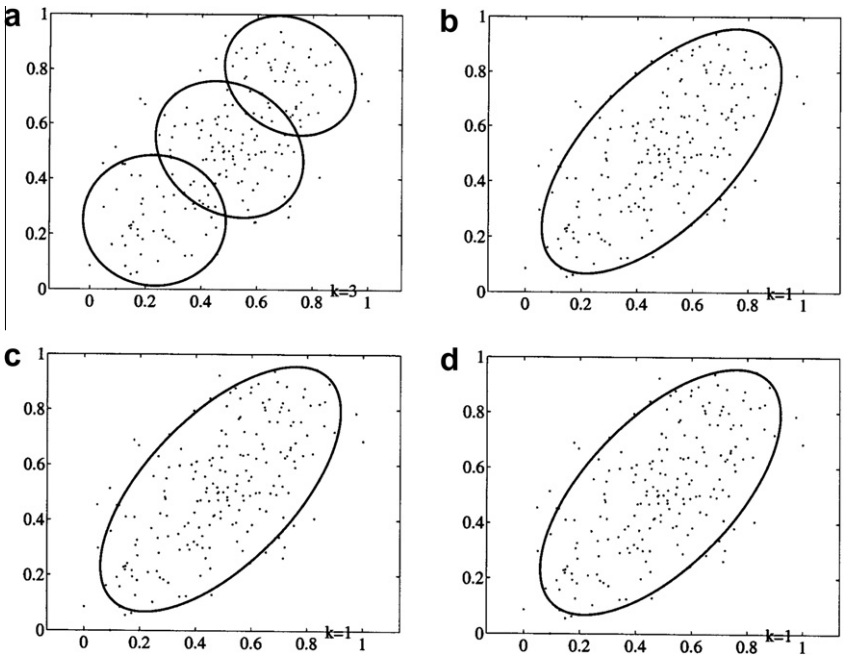

Figure 4 Examples of the **FMM**s obtained from: (a) the **TUMI**; (b) the **MI**; (c) the **MML**; and (d) the **BIC** algorithms for the Third data set. Results are shown in the subspace that consists of the first and the second features of the data set.

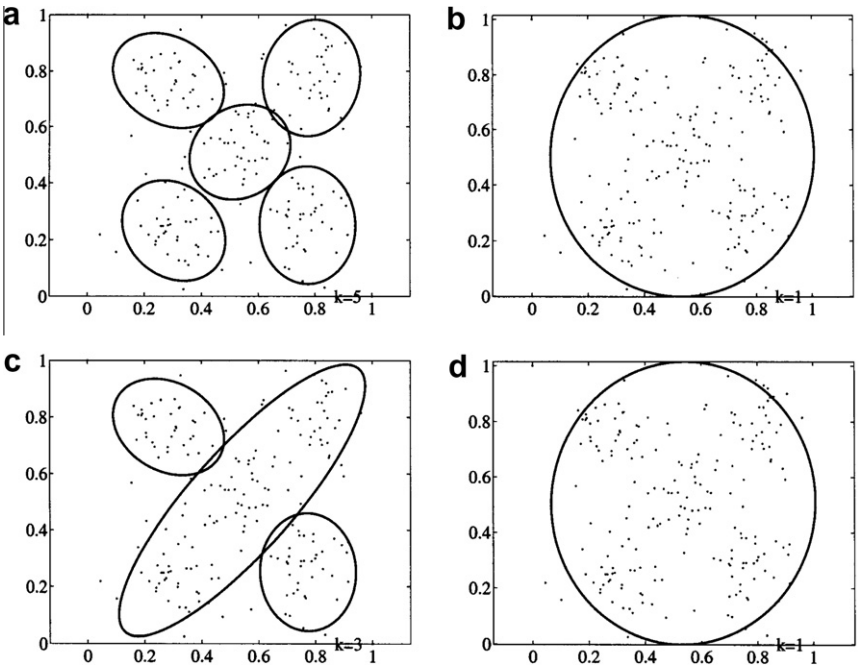

Figure 5 Examples of the resulting **FMM**s obtained from: (a) the **TUMI**; (b) the **MI**; (c) the **MML**; and (d) the **BIC** algorithms for the Fourth data set. Results are shown in the subspace that consists of the first and the second features of the data set.

algorithm is better than the performance of the **MI** algorithm with all data sets. This is because of many reasons; first, deleting from the mixture model the smallest component that has positive mutual information with the rest of the mixture components causes the model fitting to the given data set to be minimally reduced. On the other hand, the **MI** algorithm deletes the component that has the maximum positive mutual information with the rest of the mixture components. This component is always a large component in the **FMM** that is overlapped with many small components and therefore deleting it causes the model fitting to the given data set to be severely reduced. Second, computing the mutual information values using all the data points allows the **TUMI** algorithm to estimate with high accuracy these values with small data sets. On the other hand, the **MI** algorithm computes the mutual information values using the estimated mixture

parameters that are more likely to be biased due to the sparse distribution of the feature vectors in the data space. Although the **MI** algorithm is mathematically more efficient than the **TUMI** algorithm results show that it can be highly inaccurate, and therefore this efficiency gain can be worthless. Third, using the likelihood function to tune the **TUMI** algorithm allows it to estimate the number of mixture components with high accuracy when the given data set is generated from partially overlapped clusters. On the other hand, this sort of tuning is not found in the **MI** algorithm and therefore it tends to underestimate the number of mixture components when the given data set is generated from partially overlapped clusters.

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 Tuned Mutual Information (**TUMI**) algorithm, that is based on the Mutual Information Theory is proposed. This algorithm overcomes problems of the algorithms that use the penalized-likelihood or the mutual information criteria. This algorithm produces a single frame for model estimation and selection. Empirical analysis shows that the proposed algorithm outperforms the **BIC** and the **MI** algorithms. In addition, the proposed algorithm outperforms the **MML** algorithm when the given data set is sparsely distributed. However, the **TUMI** algorithm contains parameters that need empirical adjustment when the input data set is too sparse i.e., the number of data features is too large compared with the number of feature vectors. These parameters are the minimum mixing weight and the regularization constant.

In the future, dimensionality reduction may be used to reduce the sparsity of the input data set, which allows the **TUMI** algorithm to accurately handle too sparse data sets without the need to empirically adjusting its parameters. Also, the **TUMI** algorithm may be used to find out the cluster structure, both the optimum number of clusters and cluster membership for each input feature vector, in more complex and real data sets that may be sparsely distributed or generated from overlapped clusters. For example, the **TUMI** algorithm may be used to determine the Health Inequality structure of the world countries when applied on Health Inequality data sets [38]. These data sets contain a large number of features compared with the number of feature vectors i.e., sparse data.

References

[1] Webb A. Statistical pattern recognition. 2nd ed. UK: John Wiley & Sons; 2002.

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

[3] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm (with discussion). J R 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. 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 R 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 Biometrics 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] Figueiredo M, Jain A. Unsupervised learning of finite mixture models. J IEEE Trans Pattern Anal Mach Intell 2002;24(3):381–96.

[16] Celeux G, Chrétien S, Forbes F, Mkhadri A. A component-wise EM algorithm for mixtures. J Comput Graph Stat 2001;10(4):697–712.

[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] 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.

[19] 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.

[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] Roberts S, Everson R, Rezek I. Maximum certainty data partitioning. J Pattern Recognit 1999;33:833–9.

[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 Recognit Lett 1997;18:1167–78.

[24] Still S, Bialek W. How many clusters? An information-theoretic perspective. J Neural Comput 2004;16(12):2483–506.

[25] Li L, Ma J. A BYY scale-incremental EM algorithm for Gaussian mixture learning. J Appl Math Comput 2008;205:832–40.

[26] Priebe CE. Adaptive mixture density estimation. J Am Stat Assoc 1994;89:796–806.

[27] Smyth P. Model selection for probabilistic clustering using cross-validated likelihood. J Stat Comput 2000;10(1):63–72.

[28] 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 Amsterdam, 2000. http://citeseer.ist.psu.edu/vlassis00multivariate.html.

[29] 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.

[30] 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.

[31] Richardson S, Green P. On Bayesian analysis of mixtures with unknown number of components. J R Stat Soc 1997;B59:731–92.

[32] Fisher RA. The use of multiple measurements in taxonomic problems. J Ann Eugenics 1936;7:179-188. UCI Repository of machine learning databases, Irvine, CA: University of California, Department of Information and Computer Science, August 2010. http://archive.ics.uci.edu/ml/.

[33] 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.

[34] Ueda N, Nakano R, Ghahramani Z, Hinton GE. SMEM algorithm for mixture models. J Neural Comput 2000;12(9): 2109–28.

[35] Cover TM, Thomas JA. Elements of information theory. Wiley; 1991.

[36] Strehl A, Ghosh J. Cluster ensembles – A knowledge reuse framework for combining multiple partitions. J Mach Learn Res 2002;3:583–617.

[37] Fern XZ, Brodley CE. Random projection for high dimensional data clustering: a cluster ensemble approach. Proceeding of The 20th International Conference on Machine Learning (ICML 2003); p. 186–93.

[38] World Health Organization (**WHO**), Data and Statistics, August 2010. http://www.who.int/research/en/.