Unsupervised Dimensionality Reduction for Gaussian Mixture Model

Xi Yang, Kaizhu Huang, and Rui Zhang

Xi'an Jiaotong-Liverpool University,
SIP, Suzhou,215123, China
Xi.Yang07@studnet.xjtlu.edu.cn, {Kaizhu.Huang,Rui.Zhang02}@xjtlu.edu.cn

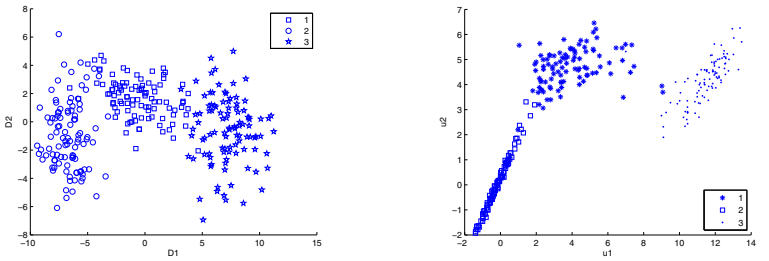
Abstract. Dimensionality reduction is a fundamental yet active research topic in pattern recognition and machine learning. On the other hand, Gaussian Mixture Model (GMM), a famous model, has been widely used in various applications, e.g., clustering and classification. For high-dimensional data, previous research usually performs dimensionality reduction first, and then inputs the reduced features to other available models, e.g., GMM. In particular, there are very few investigations or discussions on how dimensionality reduction could be interactively and systematically conducted together with the important GMM. In this paper, we study the problem how unsupervised dimensionality reduction could be performed together with GMM and if such joint learning could lead to improvement in comparison with the traditional unsupervised method. Specifically, we engage the Mixture of Factor Analyzers with the assumption that a common factor loading exist for all the components. Such setting exactly optimizes a dimensionality reduction together with the parameters of GMM. We compare the joint learning approach and the separate dimensionality reduction plus GMM method on both synthetic data and real data sets. Experimental results show that the joint learning significantly outperforms the comparison method in terms of three criteria for supervised learning.

1 Introduction

Dimensionality Reduction (DR) has been an important and fundamental research topic in pattern recognition and machine learning. Over the last fifty years, there have been many famous proposals in this area. Among them are Principal Component Analysis (PCA), Independent Component Analysis (ICA), Fisher Discriminant Analysis (FDA), Latent Diriclet Analysis (LDA), Maxi-Min Discriminant Analysis (MMDA) [6], and 1-norm based feature selection approach. In the context of classification or regression, DR could be conducted in the supervised style by utilizing certain supervised information (e.g., class labels) so as to find a subspace where different classes of data could be separated as far as possibly. These methods include the above mentioned FDA and MMDA. On the other hand, when the class information is not available, DR is performed in an unsupervised way. This family of approaches includes the famous PCA

C.K. Loo et al. (Eds.): ICONIP 2014, Part II, LNCS 8835, pp. 84–92, 2014.

and ICA. On the other hand, Gaussian Mixture Model (GMM) has achieved big success in both supervised learning, e.g., classification and regression, and unsupervised learning, e.g., clustering.



(a) 2-d subspace by PCA (b) 2-d subspace by joint learning

Fig. 1. Comparison of DR by PCA and the joint learning on simulated data (see Sect. 4.1). Data points with the same shape are supposed to be clustered together.

When GMM is used for practical data, it is usually to perform dimensionality reduction beforehand. The purpose is both to reduce the computational time for high dimensional data and to find a suitable subspace where better clustering or classification performance could be achieved due to the removal of possible noisy features. In this setting, the optimal subspace and the following optimal parameters of GMM are searched separately or independently. Apparently, the optimal subspace obtained by the independent DR may not be appropriate for the following GMM. This is particularly the case in the context of unsupervised learning, e.g., clustering. In supervised learning, class labels could be used for deriving a good subspace, whilst in unsupervised learning, the principles used for DR (e.g., maximization of variance in PCA) may not be appropriate for GMM [4]. Figure 1 (a) illustrates the best 2-dimensional subspace obtained by PCA in one synthetic data. Clearly, the original clustering information among data was less obvious after PCA.

To handle unsupervised dimensionality reduction for GMM, we argue that both the optimal subspace and the parameters for GMM should be jointly learned. This is significantly different from the traditional setting that the two steps are usually conducted separately. Specifically, we engage the Mixture of Factor Analyzers (MFA) [5] where a common factor loading is assumed to exist for all latent factors. Importantly, when this special MFA called MCFA is optimized via the modified EM algorithm, the common factor loading could be regarded as the dimensionality reduction matrix, while the mixture of latent factors can be regarded as the GMM. When GMM is used for unsupervised clustering, its joint learning with the DR subspace would make the clustering properties clearly reserved and even clear. To see the advantages, we also show in Figure 1 (b) the subspace obtained by the joint learning method. Obviously, it could lead to much better clustering performance, especially compared with PCA.

It should be noted that although mixture of factor analysis has been earlier discussed literatures such as [1], it was presented from the viewpoint of data analysis rather than dimensionality reduction. More importantly, the idea of using common loadings, or the joint learning, could also be applied in other mixture models [2]. This presents one important contribution of this paper.

2 Notation

Finite mixture of models are important models and have been widely used in many applications [3]. In the following, we present the notation used in this paper with the focus on introducing GMM. Suppose $\mathbf{y}$ be a p-dimensional vector of feature variables. The density of $\mathbf{y}$ could be modeled by a mixture of g multivariate normal component distributions $P(\mathbf{y};\theta) = \sum_{i=1}^{g} \pi_i \mathcal{N}(\mathbf{y};\mu_i,\sigma_i)$, where each gaussian distribution $\mathcal{N}(\mathbf{y};\mu,\sigma)$ is known as a component of this model and describes the p-variate normal density function with mean μ and covariance matrix σ. The unknown parameter vector θ consists of the mixture weight π_i, the means of component μ_i, and the covariance of component matrices $\sigma_i (i = 1,\ldots,g)$. This vector can be estimated by maximizing the log-likelihood function: $\log L(\theta) = \sum_{j=1}^{n} \log P(\mathbf{y}_j;\theta)$, where $\{\mathbf{y}_j\}$ $(j = 1,\ldots,n)$ is an observed random sample set. By using the Expectation-Maximization (EM) algorithm [1], the local maximizers of log-likelihood function can be obtained as follows:

$$\pi_i^{(k+1)} = \frac{1}{n} \sum_{(i=1)}^{n} P^{(k)}(\omega_i \mid \mathbf{y}_j;\theta), \quad \mu^{(k+1)} = \frac{\sum_{(i=1)}^{n} P^{(k)}(\omega_i \mid \mathbf{y}_j;\theta)\mathbf{y}_j}{\sum_{(i=1)}^{n} P^{(k)}(\omega_i \mid \mathbf{y}_j;\theta)}$$

$$\sigma^{(k+1)} = \frac{\sum_{(i=1)}^{n} P^{(k)}(\omega_i \mid \mathbf{y}_j;\theta)(\mathbf{y}_j - \mu^{(k)})(\mathbf{y}_j - \mu^{(k)})^T}{\sum_{(i=1)}^{n} P^{(k)}(\omega_i \mid \mathbf{y}_j;\theta)} .$$

In the above, ω_i represent the i-th latent component category that each sample $\mathbf{y}_j$ belongs to. With the Bayes theorem, the posterior distribution $P(\omega_i \mid \mathbf{y_j};\theta)$ can be expressed as $P(\omega_i \mid \mathbf{y}_j;\theta) = \frac{\pi_i \mathcal{N}(\mathbf{y}_j;\mu_i,\sigma_i)}{\sum_{h=1}^{g} \pi_h \mathcal{N}(\mathbf{y}_j;\mu_h,\sigma_h)}$, $i = 1,\ldots,g; j = 1,\ldots,n$. Here, as the categories ω_i of each sample $\mathbf{y}_j$ are unknown, the latent variable is the indicator variable ω, $\omega = \{0,1\}, \pi_i = P(\omega_i = 1)$. A data point could be assigned to the component that has the highest estimated posterior probability.

3 Unsupervised Dimensionality Reduction with MCFA

In this section, we will present the mixtures of factor analyzers with common factor loadings (MCFA). This model learns jointly the dimensionality reduction and the parameters of GMM. We will first describe the model definition, and then introduce the involved optimization.

3.1 Model Description

Suppose $Y = (Y_1, \ldots, Y_p)^T$ be generated by linear combination with q-dimensional vector of (unobservable) factors $\mathbf{Z}_{i1}, \ldots, \mathbf{Z}_{in}$. In MFA, the mixture weight $\pi_i, (i = 1, \ldots, g)$ is modeled as

$$\mathbf{Y}_j - \mu_i = \mathbf{\Lambda}\mathbf{Z}_{ij} + e_{ij}, \tag{1}$$

where $\mathbf{\Lambda}$ is called the factor loading vector, the factors $\mathbf{Z}_{ij}$ are distributed independently as $\mathcal{N}(0, I_q)$, e_{ij} is random noise distributed independently under $\mathcal{N}(0, \mathbf{D}_i)$. Here $\mathbf{D}_i$ is a $q \times q$ positive definite symmetric matrix $(i = 1, \ldots, g)$. MCFA further assumes the additional restrictions:

$$\mu_i = \mathbf{A}\xi_i; \quad \sigma_i = \mathbf{A}\mathbf{\Omega}_i\mathbf{A}^T + \mathbf{D}; \quad \mathbf{D}_i = \mathbf{D}; \quad \mathbf{\Lambda}_i = \mathbf{A}\mathbf{K}_i, \tag{2}$$

where $\mathbf{A}$ is a $p \times q$ matrix, ξ_i is a q-dimensional vector $(i = 1, \ldots, g)$, and $\mathbf{D}$ is a diagonal $p \times p$ matrix. Hence the distribution of $\mathbf{Y}_j$ is modeled as

$$\mathbf{Y}_j = \mathbf{A}\mathbf{Z}_{ij} + \mathbf{e}_{ij}, \tag{3}$$

where the (unobservable) factors $\mathbf{Z}_{ij}$ are distributed independently under $\mathcal{N}(\xi_i, \mathbf{\Omega}_i)$, $\mathbf{e}_{ij}$ is random noise distributed independently under $\mathcal{N}(0, \mathbf{D})$, and $\mathbf{D}$ is a diagonal matrix. Here the common loading $\mathbf{A}$ can easily be seen as the transformation matrix, reducing p-dimensional to a latent q-dimensional space.

With the above definitions, the MCFA model can be written as

$$P(\mathbf{y}; \theta) = \sum_{i=1}^{g} \sum_{j=1}^{n} \pi_i \mathcal{N}(\mathbf{y}_j; \mathbf{A}\xi_i, \mathbf{A}\mathbf{\Omega}_i\mathbf{A}^T + \mathbf{D}).$$

Assume we have a mixture of g components by the component-indicator labels ω_i, where ω_i is one or zero depending on whether or not $\mathbf{y}_j$ belongs to the i-th component of the model. The likelihood function can then be written as

$$\mathcal{L}(\mathbf{y}) = \prod_{i=1}^{g} \prod_{j=1}^{n} P(\mathbf{y}_j \mid \mathbf{Z}_{ij}, \omega_i) P(\mathbf{Z}_{ij} \mid \omega_i) P(\omega_i).$$

Since the factors are distributed independently $N(\xi_i, \mathbf{\Omega}_i)$, we have $P(\mathbf{Z}_{ij} \mid \omega_i) = \mathcal{N}(\mathbf{Z}_{ij} \mid \xi_i, \mathbf{\Omega}_i)$. Then, the log-likelihood function is given by

$$\log L_c(\theta) = \sum_{i=1}^{g} \sum_{j=1}^{n} \omega_{ij} \{\log \pi_i + \log \mathcal{N}(\mathbf{y}_j; \mathbf{A}\mathbf{u}_{ij}, \mathbf{D}) + \log \mathcal{N}(\mathbf{Z}_{ij}; \xi_i, \mathbf{\Omega}_i)\}. \tag{4}$$

In the next subsection, we will introduce how to use EM to find the dimensionality reduction matrix $\mathbf{A}$ as well as the parameters of GMM.

3.2 Optimization

Maximization of (4) can be conducted by the famous EM algorithm, or in particularly, the alternating expectation-conditional maximization algorithm (AECM) [5].

E-step. At this step, we need to compute expectations of the hidden variables τ_{ij}, $E(\mathbf{Z} \mid \mathbf{y}_j, \omega_i)$ and $E(\mathbf{Z}\mathbf{Z}' \mid \mathbf{y}_j, \omega_i)$ that appear in the log-likelihood for all data point $j = 1, \ldots, n$ and mixture components $i = 1, \ldots, g$. It is easily verified that

$$E(\mathbf{Z} \mid \mathbf{y}_j, \omega_i) = \xi_i + \gamma_i^T(\mathbf{y}_j - \mathbf{A}\xi_i), \tag{5}$$

$$E(\mathbf{Z}\mathbf{Z}' \mid \mathbf{y}_j, \omega_i) = (I_q - \gamma_i^T A)\mathbf{\Omega}_i + E(\mathbf{Z} \mid \mathbf{y}_j, \omega_i)E(Z \mid \mathbf{y}_j, \omega_i)', \tag{6}$$

where $\gamma_i = (\mathbf{A}\mathbf{\Omega}_i\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{\Omega}_i$.

At each iteration, it is also necessary to compute the conditional expectation of (4) denoted by $\mathbf{Q}(\theta; \theta^{(k)})$. Given the observed data $\mathbf{y}$ and $\theta^{(k)}$, we have

$$\mathbf{Q}(\theta; \theta^{(k)}) := P(Z^k \mid \mathbf{y}^k; \theta). \tag{7}$$

The conditional expectation of the component labels $\omega_{ij} (i = 1, \ldots, g; j = 1, \ldots, n)$ can be written as $\mathbf{E}_\theta\{\omega_{ij} \mid \mathbf{y}_j\} = Pr_\theta\{\omega_{ij} = 1 \mid \mathbf{y}_j\} = \tau_i(\mathbf{y}_j; \theta)$, where $\tau_i(\mathbf{y}_j)$ is the posterior probability that $\mathbf{y}_j$ belongs to the i^{th} component. From (2), it can then be obtained

$$\tau_i(\mathbf{y}_j; \theta) = \frac{\pi_i \phi(\mathbf{y}_j; \mathbf{A}\xi_i, \mathbf{A}\mathbf{\Omega}_i\mathbf{A}^T + D)}{\sum_{h=1}^{g} \pi_h \phi(\mathbf{y}_j; \mathbf{A}\xi_h, \mathbf{A}\mathbf{\Omega}_h\mathbf{A}^T + D)}, \tag{8}$$

Denoting $\tau_{ij}^{(k)} = \tau_i(\mathbf{y}_j; \theta^{(k)})$, we can transform (7) as

$$\begin{aligned} Q(\theta; \theta^{(k)}) = \sum_{i=1}^{g}\sum_{j=1}^{n} \tau_{ij}^{(k)} \{ & \log \pi_i + E_{\theta^{(k)}}\{\log \mathcal{N}(y_j; Az_{ij}, D) | y_j, \omega_{ij} = 1\} \\ & + E_{\theta^{(k)}}\{\log N(z_{ij}; \xi_i, \mathbf{\Omega}_i) | y_j, \omega_{ij} = 1\}\}. \end{aligned}$$

M-step. At the $(k+1)$-th iteration of the EM algorithm, the M-step consists of calculating the updated estimates $\pi_i^{(k+1)}$, $\xi_i^{(k+1)}$, $\mathbf{\Omega}_i^{(k+1)}$, $\mathbf{A}^{(k+1)}$ and $D^{(k+1)}$ by maximization of $Q(\theta; \theta^{(k)})$. The updated estimates of the mixing proportions π_i are derived in the case of the normal mixture model by $\pi_i^{(k+1)} = \frac{1}{n}\sum_{j=1}^{n} \tau_{ij}^{(k)}$, $(i = 1, \ldots . g)$. Concerning the other parameters, we have the following

$$\xi_i^{(k+1)} = \xi_i^{(k)} + \frac{\sum_{j=1}^{n} \tau_{ij}^{(k)} \varphi^{(k)}}{\sum_{j=1}^{n} \tau_{ij}^{(k)}}, \mathbf{\Omega}_i^{(k+1)} = \frac{\sum_{j=1}^{n} \tau_{ij}^{(k)} \varphi^{(k)} \varphi^{(k)^T}}{\sum_{j=1}^{n} \tau_{ij}^{(k)}} + (I_q - \varphi^{(k)})\mathbf{\Omega}_i^{(k)},$$

$$\varphi^{(k)} = (\mathbf{A}^{(k)}\mathbf{\Omega}_i^{(k)}\mathbf{A}^{(k)^T} + D^{(k)})^{-1}\mathbf{A}^{(k)}\mathbf{\Omega}_i^{(k)}(y_j - \mathbf{A}^{(k)}\xi_i^{(k)}).$$

The updated estimates $D^{(k+1)} = diag(D_1^{(k)} + D_2^{(k)})$, where

$$D_1^{(k)} = \frac{\sum_{i=1}^{g}\sum_{j=1}^{n} \tau_{ij}^{(k)} D^{(k)}(I_p - \beta_i^{(k)})}{\sum_{i=1}^{g}\sum_{j=1}^{n} \tau_{ij}^{(k)}}, \beta_i^{(k)} = (\mathbf{A}^{(k)}\mathbf{\Omega}^{(k)}\mathbf{A}^{(k)^T} + D^{(k)})^{-1} D^{(k)},$$

$$D_2^{(k)} = \frac{\sum_{i=1}^{g}\sum_{j=1}^{n} \tau_{ij}^{(k)} \beta_i^{(k)^T}(y_j - \mathbf{A}^{(k)}\xi_i^{(k)})(y_j - \mathbf{A}^{(k)}\xi_i^{(k)})^T \beta_i^{(k)}}{\sum_{i=1}^{g}\sum_{j=1}^{n} \tau_{ij}^{(k)}}.$$

We also have $\mathbf{A}^{(k+1)} = (\sum_{i=1}^{g} \mathbf{A}_{1i}^{(k)})(\sum_{i=1}^{g} \mathbf{A}_{2i}^{(k)})^{-1}$, where $\mathbf{A}_{1i}^{(k)} = \sum_{i=1}^{g} \tau_{ij}^{(k)}\{y_j E^{(k)}(\mathbf{Z} \mid \mathbf{y}_j, \omega_i^{(k)})\}; \mathbf{A}_{2i}^{(k)} = \sum_{i=1}^{g} \tau_{ij}^{(k)}\{E^{(k)}(\mathbf{Z}\mathbf{Z}' \mid \mathbf{y}_j, \omega_i^{(k)})\}$.

4 Experiments

In this section, we evaluate the performance of the joint learning approach MCFA on one simulation and three real data sets (obtained from UCI machine learning repository) in comparison with the PCA followed by GMM. Following previous research, we report the error rate (ERR), the adjust rand index (ARI), and the Bayesian information criterion (BIC) to compare different algorithms. Note that, although we did not use any labeled information in clustering, the clustering result for each sample is known beforehand in the data sets used. Hence we could exploit ERR as the evaluation metric for clustering.

Table 1. Comparison among the MCFA and PCA-GMM on Simulated Data

MCFA					PCA				
Cluster	DIM	ERR	BIC	ARI	Cluster	DIM	ERR	BIC	ARI
2	2	0.3333	4173	0.5600	2	2	0.3333	3153	0.5553
3	2	0.0100	4105	0.9702	3	2	0.0300	3080	0.9126

Simulation Data. To validate the effectiveness of the joint learning approach MCFA, we first performed a simulation experiment. We generated 300 random vectors from each of $g = 3$ different three-dimensional multivariate normal distributions. The three distributions have respectively means $\mu_1 = (0, 0, 0)^T, \mu_2 = (2, 2, 6)^T, \mu_3 = (8, 8, 8)^T$, and covariance matrices

$$\Sigma_1 = \begin{pmatrix} 4 & -1.8 & -1 \\ -1.8 & 2 & 0.9 \\ -1 & 0.9 & 2 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 4 & 1.8 & 0.8 \\ 1.8 & 2 & 0.5 \\ 0.8 & 0.5 & 2 \end{pmatrix}, \Sigma_3 = \begin{pmatrix} 4 & 0 & -1 \\ -1.8 & 2 & 0.9 \\ -1 & 0.9 & 2 \end{pmatrix}.$$

We compared the performance of MCFA with PCA, and plot the unsupervised feature reduction results on Figure 1. It is obvious the joint learning approach leaded to better data separation. To quantitatively evaluate the clustering performance, we compute the ERR, ARI and BIC with the PCA followed by GMM and the joint learning MCFA. These results are shown in Table 1. From the table, the lowest BIC of both approaches are pointed to 3 clusters, indicating that 3 is the best cluster number. Moreover, in case of 3 cluster number, the joint learning MCFA outperformed PCA followed by GMM significantly in terms of the other two criteria.

Table 2. Comparison among MCFA and PCA+GMM on User Knowledge Data

User Knowledge Modeling									
MCFA					PCA				
Cluster	DIM	ERR	BIC	ARI	Cluster	DIM	ERR	BIC	ARI
	2	0.3891	-117	0.4474		2	0.4358	187	0.2469
2	3	0.3891	-87	0.4474	2	3	0.4514	212	0.2896
	4	0.3891	-48	0.4474		4	0.4553	150	0.2442
	2	0.3074	-126	0.4190		2	0.3735	210	0.3001
3	3	0.3035	-121	0.4242	3	3	0.3969	232	0.2924
	4	0.3074	-22	0.4477		4	0.4786	159	0.1781
	2	0.1634	-142	0.6456		2	0.4008	225	0.2771
4	3	0.1868	-92	0.6240	4	3	0.4591	230	0.2791
	4	0.2451	-86	0.5901		4	0.4669	216	0.2593

User Knowledge Modeling Data. This data set consists of $n = 403$ samples and 5 attribute information. The classes are four knowledge levels of the students. As we usually do not know the cluster number, we have compared the joint learning MCFA and PCA+GMM in case of various cluster number and different dimensionality ranged from 2 to the feature number. We report the comparison results in Table 2. Again, it is observed that almost in all the cases, the joint learning demonstrated the better performance than PCA+GMM. Furthermore, the best estimated cluster number of MCFA is 4 and 2 factors according to the lowest BIC. This setting also achieved the lowest ERR, and the highest ARI. It is significantly better than the best case of PCA+GMM.

Table 3. Comparison among the MCFA and PCA+GMM on Physical Data

Physical Data									
MCFA					PCA				
Cluster	DIM	ERR	BIC	ARI	Cluster	DIM	ERR	BIC	ARI
	2	0.3146	7398	0.4717		2	0.3258	4142	0.3963
	3	0.2921	7134	0.5397		3	0.3202	5106	0.4219
	4	0.2921	7010	0.5298		4	0.3202	5945	0.4820
2	5	0.2753	6962	0.5711	2	5	0.3258	6452	0.4088
	6	0.2697	6973	0.5820		6	0.3315	6922	0.3916
	7	0.2697	6986	0.5820		7	0.2865	7255	0.5499
	8	0.2697	7045	0.5820		8	0.2921	7487	0.5397
	2	0.0562	7384	0.8298		2	0.2978	4130	0.3827
	3	0.0225	7096	0.9295		3	0.2697	5109	0.4302
	4	0.0225	6922	0.9309		4	0.1401	5872	0.6170
3	5	0.0169	6935	0.9485	3	5	0.0730	6413	0.7822
	6	0.0056	6881	0.9817		6	0.0562	6905	0.8319
	7	0.0056	6948	0.9817		7	0.0618	7253	0.8185
	8	0.0056	6944	0.9832		8	0.0449	7462	0.8708
	2	0.0618	7411	0.8145		2	0.2978	4165	0.3600
	3	0.0393	7106	0.8792		3	0.2865	5143	0.3977
	4	0.0169	6999	0.9470		4	0.1404	5863	0.6479
4	5	0.0169	6988	0.9470	4	5	0.1124	6445	0.7531
	6	0.0169	7018	0.9551		6	0.1180	6992	0.7436
	7	0.0056	7099	0.9833		7	0.0899	7327	0.8355
	8	0.0056	7121	0.9900		8	0.1011	7505	0.8264

Physical Data. This data set contains results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. It consists of 178 samples with 13 constituents and 3 classes. Again, we have compared the joint learning MCFA and PCA+GMM in Table 3 in cases of various cluster number and different dimensionality ranged from 2 to the feature number. Obviously, in almost all the cases, the joint learning leaded to better performance than PCA+GMM in terms of ERR and ARI. Furthermore, the best estimated cluster number of MCFA is 3 according to the lowest BIC. This also matches the class number in this data set. Such setting again achieved the lowest ERR, and the highest ARI, which outperformed that of PCA+GMM.

Iris Data. The data set contains 3 classes of 150 3-dimensional instances. Using the similar setting in the previous data, we present the performance of the joint learning model MCFA and PCA+GMM approaches in Table 4. Once again, MCFA demonstrated better performance than PCA+GMM. In particular, when cluster number is set to 2, MCFA performed the same as PCA+GMM, while it outperformed significantly in cases of 3 and 4 cluster numbers.

Table 4. Comparison among the MCFA and PCA+GMM on Iris Data

Iris									
MCFA					PCA				
Cluster	DIM	ERR	BIC	ARI	Cluster	DIM	ERR	BIC	ARI
2	2	0.3333	624	0.5681	2	2	0.3333	672	0.5681
	3	0.3333	571	0.5681		3	0.3333	717	0.5681
3	2	0.0200	654	0.9410	3	2	0.0267	672	0.9222
	3	0.0200	571	0.9410		3	0.0267	733	0.9222
4	2	0.0200	692	0.9410	4	2	0.0533	706	0.8700
	3	0.0200	628	0.9410		3	0.0800	755	0.8570

5 Conclusion

This paper mainly introduced a method learning jointly both the optimal subspace and the parameters for GMM. This is significantly different from traditional unsupervised dimensionality reduction for GMM, where the dimensionality reduction and parameter learning are usually conducted independently. A series of experiments on 1 synthetic and 3 real data sets showed that the engaged joint learning approach consistently outperformed the competitive model.

Acknowledgement. The research was partly supported by the National Basic Research Program of China (2012CB316301) and Jiangsu University Natural Science Research Programme (14KJB520037).

References

1. Baek, J., McLachlan, G.J., Flack, L.K.: Mixtures of factor analyzers with common factor loadings: Applications to the clustering and visualization of high-dimensional data. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(7), 1298–1309 (2010)
2. Figueiredo, M., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Transaction on Pattern Analysis and Machine Intelligence 24(3), 381–396 (2002)
3. Huang, K., King, I., Lyu, M.R.: Finite mixture model of bound semi-naive bayesian network classifier. In: Kaynak, O., Alpaydın, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714, pp. 115–122. Springer, Heidelberg (2003)
4. Huang, K., Yang, H., King, I., Lyu, M.R.: Machine Learning: Modeling Data Locally and Gloablly. Springer (2008) ISBN 3-5407-9451-4
5. Mclanchlan, G.J., Peel, D., Bean, R.W.: Modelling high-dimensional data by mixtures of factor analyzers. Computational Statistics & Data Analysis 41, 379–388 (2003)
6. Xu, B., Huang, K., Liu, C.-L.: Maxi-min discriminant analysis via online learning. Neural Networks 34, 56–64 (2012)