PCA 혼합 모형과 클래스 기반 특징에 의한 LDA의 확장

(Extensions of LDA by PCA Mixture Model and Class-wise Features)

김 현 철 † 김 대 진 †† 방 승 양 †††
(Hyun-Chul Kim) (Daijin Kim) (Sung-Yang Bang)

요 약 LDA는 클래스간 퍼진 정도와 클래스내 퍼진 정도의 비를 최대화하는 변환를 구하는 데이타 구분 기술이다. LDA는 여러 가지 응용에 성공적으로 응용되었지만 그 모델의 단순성과 관련된 두 가지 한계를 가지고 있다. 첫째는 각 클래스의 데이타가 가우시안 분포를 가진다고 가정되므로 복잡한 분포를 갖는 데이타를 구분하는데 실패한다는 것이다. 둘째는 LDA가 클래스의 전체 범위에 대해서 단지 하나의 변환만을 주므로 클래스 기반의 정보를 잃게 된다는 것이다. 본 논문은 위의 문제들을 극복하는 세가지 확장들을 제안한다. 첫 번째 확장은 더 복잡한 분포를 표현할 수 있는 PCA 혼합 모형을 이용하여 클래스 내 퍼진 정도를 모델링함으로써 첫째 문제를 극복한다. 두번째 확장은 클래스 기반 특징들을 제공하기 위해서 각 클래스에 대해 다른 변환을 취함으로써 둘째 문제를 극복한다. 셋째 확장은 PCA 혼합 모형의 관점에서 각 클래스를 표현함으로써 앞의 두 확장을 결합하는 것이다. 숫자 인식과 알파벳 인식에 대한 실험에서 LDA의 모든 제안된 확장들이 LDA보다 더 좋은 분류 성능을 보여 주었다.

키워드 : LDA, LDA 확장, PCA 혼합 모형, 클래스 기반 특징

Abstract LDA (Linear Discriminant Analysis) is a data discrimination technique that seeks transformation to maximize the ratio of the between-class scatter and the within-class scatter. While it has been successfully applied to several applications, it has two limitations, both concerning the underfitting problem. First, it fails to discriminate data with complex distributions since all data in each class are assumed to be distributed in the Gaussian manner; and second, it can lose class-wise information, since it produces only one transformation over the entire range of classes. We propose three extensions of LDA to overcome the above problems. The first extension overcomes the first problem by modeling the within-class scatter using a PCA mixture model that can represent more complex distribution. The second extension overcomes the second problem by taking different transformation for each class in order to provide class-wise features. The third extension combines these two modifications by representing each class in terms of the PCA mixture model and taking different transformation for each mixture component. It is shown that all our proposed extensions of LDA outperform LDA concerning classification errors for handwritten digit recognition and alphabet recognition.

Key words : LDA, extensions of LDA, PCA mixture models, class-wise features

1. Introduction

LDA is a well-known classical statistical technique to find the projection that maximizes the ratio of scatter among the data of different classes to scatter within the data of the same class [1]. Features obtained by LDA are useful for pattern classification since they bring the data of the same class more closely, and the data of different classes farther. Recently, LDA has been applied to several problems, such as face recognition [2] and image

† 비 회 원 : 포항공과대학교 컴퓨터공학과
grass@postech.ac.kr
†† 비 회 원 : 포항공과대학교 컴퓨터공학과 교수
dkim@postech.ac.kr
††† 정 회 원 : 포항공과대학교 컴퓨터공학과 교수
sybang@postech.ac.kr
논문접수 : 2004년 12월 6일
심사완료 : 2005년 7월 12일

retrieval [3], and displays a better performance than PCA.

Although LDA usually gives good discrimination performance, two deficiencies are known to exist, as follows [4]. LDA is too flexible in situations where there are many highly correlated variables, and too rigid in situations where the class boundaries are complex or nonlinear. In the former case, LDA can overfit the data, and in the latter case, LDA can underfit the data. To overcome these deficiencies, a number of extensions of LDA has been proposed in the literature [4,5]. Recently, nonlinear extensions of LDA by kernel methods have also been proposed [6].

In this paper, we concentrate on two drawbacks of LDA that can cause an underfitting problem. The first is where LDA assumes that the data of each class have uni-modal Gaussian distribution. LDA does not fit well for distributions other than the uni-modal Gaussian distribution. The second is where LDA finds one projection over all classes, and so loses some class-wise important information for the classification.

To overcome these drawbacks, we introduce three extensions of LDA. The first extension (PM-LDA (LDA with PCA Mixture)) overcomes the first drawback by using the PCA mixture model for modeling the data of each class. This works well for data with more complex distribution than uni-modal Gaussian distribution. This extension is similar to MDA[5], an extension of LDA by Gaussain mixture model, but this extension uses the PCA mixture model. In the second extension (C-LDA (LDA with class-wise features)), we overcome the second drawback by extending LDA using class-wise features. We use different projections for different classes by obtaining an individual transformation matrix which corresponds to each class. In this way we can keep some class-wise important information. Finally, the third extension (PM-S-LDA (LDA with PCA Mixture and sub-classwise features)) combines the above two extensions. We model each class by a PCA mixture model. By obtaining an individual transformation matrix for each subclass (mixture component), we use different projections for different subclasses.

This paper is organized as follows. Section 2 describes the LDA. Section 3 describes the PCA mixture model. Section 4 explains the proposed extensions of LDA. Section 5 shows the simulation results of pattern classification problems using handwritten digit and alphabet data. Finally, we present our conclusions.

2. LDA

The goal of LDA is to find an orientation for which the projected samples are well separated [1]. Specifically, LDA seeks a transformation matrix $\boldsymbol{W}$ that in some sense maximizes the ratio of the between-class scatter to the within-class scatter. Initially, we consider a within-class scatter matrix for the within-class scatter. A within-class scatter matrix $\boldsymbol{S_W}$ is defined as

$$\boldsymbol{S_W} = \sum_{i=1}^{c} \sum_{x \in C_i} (\boldsymbol{x} - \boldsymbol{m_i})(\boldsymbol{x} - \boldsymbol{m_i})^t, \tag{1}$$

where c is the number of classes, C_i is a set of data within the ith class, and $\boldsymbol{m_i}$ is the mean of the ith class.

The within-class scatter matrix represents the degree of scatter within classes as a summation of covariance matrices of each class.

Next, we consider a between-class scatter matrix for a between-class scatter. A between-class scatter matrix $\boldsymbol{S_B}$ is defined as

$$\boldsymbol{S_B} = \sum_{i=1}^{c} n_i (\boldsymbol{m_i} - \boldsymbol{m})(\boldsymbol{m_i} - \boldsymbol{m})^t \tag{2}$$

The between-class scatter matrix represents the degree of scatter between classes as a covariance matrix of means of each class.

We seek a transformation matrix that in some sense maximizes the ratio of the between-class scatter and the within-class scatter. For the scalar measure of scatter, we use the determinant of scatter matrices. The determinant of the scatter matrix is the product of variances in the transformed directions, since the determinant is the product of the eigenvalues. Using this measure, the criterion function $J(\boldsymbol{W})$ can be defined as

$$J(\boldsymbol{W}) = \frac{|\boldsymbol{W}^t \boldsymbol{S_B} \boldsymbol{W}|}{|\boldsymbol{W}^t \boldsymbol{S_W} \boldsymbol{W}|}. \tag{3}$$

We can obtain the transformation matrix $\boldsymbol{W}$ as one that maximizes the criterion function $J(\boldsymbol{W})$.

The columns of optimal $\boldsymbol{W}$ are the generalized eigenvectors $\boldsymbol{w_i}$ that correspond to the largest eigenvalues in

$$\boldsymbol{S_B w_i} = \lambda_i \boldsymbol{S_W w_i}. \tag{4}$$

3. PCA Mixture Model

As mentioned before, LDA assumes that the data of each class have Gaussian distribution. Thus, the classification performance is limited when the data distribution is more complex. The data in a certain class is separated into many subclasses. We can treat such a case effectively by the PCA mixture model. We extend LDA by using this advantage of PCA mixture model in Section 3. Next, we explain PCA mixture model.

We consider a PCA mixture model which combines the ideas of mixture models and PCA. As in mixture models [7], the density of data $\boldsymbol{x}$ is represented as the weighted sum of component densities as

$$P(\boldsymbol{x}) = \sum_{k=1}^{K} P(\boldsymbol{x}|c_k,\theta_k)P(c_k). \tag{5}$$

Each component density $P(\boldsymbol{x}|c_k,\theta_k)$ is modeled in the PCA transformed space as

$$P(\boldsymbol{x}|c_k,\theta_k) = P(\boldsymbol{s_k}|c_k,\theta_k), \tag{6}$$

where $\boldsymbol{s_k} = \boldsymbol{T_k^T}(\boldsymbol{x}-\boldsymbol{\mu_k})$. $\boldsymbol{T_k}$ and μ_k are a PCA transform matrix and a mean for a mixture component k. For a mixture component k, the PCA feature vectors $\boldsymbol{s_k}$ are decorrelated, and so its covariance matrix $\boldsymbol{\Sigma_k^s} = E[\boldsymbol{s_k s_k}^T]$ is a diagonal matrix whose diagonal elements correspond to the principal eigenvalues. Next, the conditional density function $P(\boldsymbol{x}|c_k,\theta_k)$ of the PCA feature vectors for a mixture component k can be simplified as

$$P(\boldsymbol{s_k}|c_k,\theta_k) = \frac{1}{(2\pi)^{\frac{m}{2}}|\Sigma_k^{\boldsymbol{s_k}}|^{\frac{1}{2}}} exp(-\frac{1}{2}\boldsymbol{s_k}^T\Sigma_k^{-1}\boldsymbol{s_k}) \tag{7}$$

$$= \prod_{j=1}^{m} \frac{1}{(2\pi)^{\frac{1}{2}}\lambda_{k,j}^{\frac{1}{2}}} exp(-\frac{s_j^2}{2\lambda_{k,j}}),$$

where $(\lambda_{k,1},\cdots,\lambda_{k,m})$ are the m dominant eigenvalues of the feature covariance matrix $\Sigma_k^{\boldsymbol{s}}$ for a mixture component k. The proposed model,which has no Gaussian error term, can be considered as a simplified form of Tipping and Bishop's model [8].

The parameters of a mixture model can be estimated by an EM algorithm, which maximizes the likelihood [9]. The EM algorithm for the proposed model can be easily derived. Each iteration consists of two steps: an expectation step (E-step) followed by a maximization step (M-step). Each step is run for each mixture component. The EM Algorithm starts its run after the parameters are initialized, and stops when the density undergoes no further changes.

(1) E-step

Given the data set x and the parameters $\theta_k^{(t)}$ of the mixture model at the tth iteration, we estimate the posterior distribution $P(c_k|\boldsymbol{x},\theta_k^{(t)})$ using

$$P(c_k|\boldsymbol{x},\theta_k^{(t)}) = P(c_k|\boldsymbol{s_k},\theta^{(t)}) \tag{8}$$

$$= \frac{P(\boldsymbol{s_k}|c_k,\theta_k^{(t)})P(c_k)}{\sum_{k=1}^{K} P(\boldsymbol{s_k}|c_k,\theta_k^{(t)})P(c_k)},$$

where K is the number of mixture components and $P(\boldsymbol{s_k}|c_k,\theta_k^{(t)})$ is computed by Eq. 7.

(2) M-step

The new means $\boldsymbol{\mu}_k^{(t+1)}$ and the new covariance matrixes $\Sigma_k^{(t+1)}$ of the kth mixture component are obtained by the following update formula.

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{p=1}^{N} P(c_k|\boldsymbol{x}_{p,\theta_k^{(t)}})\boldsymbol{x}_p}{\sum_{p=1}^{N} P(c_k|\boldsymbol{x}_{p,\theta_k^{(t)}})} \tag{9}$$

$$\Sigma_k^{(t+1)} = E[P(c_k|\boldsymbol{x},\theta^{(t)})(\boldsymbol{x}-\boldsymbol{\mu}_k^{(t+1)})^T(\boldsymbol{x}-\boldsymbol{\mu}_k^{(t+1)})]$$

$$= \sum_{p=1}^{N} P(c_p|\boldsymbol{x}_{p,}\theta^{(t)})(\boldsymbol{x}_p-\boldsymbol{\mu}_k^{(t+1)})^T(\boldsymbol{x}_p-\boldsymbol{\mu}_k^{(t+1)}) \sum_{p=1}^{N} P(c_p|\boldsymbol{x}_{p,\theta^{(t)}}). \tag{10}$$

The new variance parameters $\lambda_{k,j}^{(t+1)}$ are obtained by selecting the largest m eigenvalues in the eigenvector computation as

$$\Sigma_k^{(t+1)}\boldsymbol{w}_{k,j} = \lambda_{k,j}^{(t+1)}\boldsymbol{w}_{k,j}. \tag{11}$$

PCA transform matrixes $\boldsymbol{T_k}$ is obtained as $[\boldsymbol{w_{k,1}}\ \boldsymbol{w_{k,2}}\ \cdots\ \boldsymbol{w_{k,m}}]$.

4. Extensions of LDA

4.1 Extension of LDA by PCA Mixture Model (PM-LDA)

LDA uses covariance matrices for the within-class scatter matrices, which means that data in each class are assumed to be Gaussian distributed. When data is not distributed into the uni-modal Gaussian, LDA does not work well. However, there are many cases in which data is not distributed into the uni-modal Gaussian. We extend LDA by the PCA mixture model, called PM-LDA, which is capable of modeling more complex distributions, such as the multi-modal Gaussian.

In PM-LDA, we apply the PCA mixture model to each class C_i where each class is a combination of s mixture components. Next, we obtained means m_{ik}, variances $\boldsymbol{V_{ik}}$, transformation matrixes $\boldsymbol{T_{ik}}$, for the kth mixture component of the ith class. $\boldsymbol{V_{ik}}$ is a diagonal matrix whose diagonal element is eigenvalues $\lambda_{ik,j}$ which is the jth largest eigenvalue of the covariance matrix for the kth mixture component in the ith class. Next, we define the new scatter matrices. The covariance matrix for the kth mixture component in the ith class is $\boldsymbol{T_{ik}V_{ik}T_{ik}^t}$. A new within-class scatter matrix $\boldsymbol{S_W}$ is the summation of covariance matrices for all classes and all mixture components as

$$\boldsymbol{S_W}=\sum_{i=1}^{c}\sum_{k=1}^{s}\boldsymbol{T_{ik}V_{ik}T_{ik}^t}, \tag{12}$$

where c and s are the number of classes and the number of mixture components of each class, respectively. The new within-class scatter matrix $\boldsymbol{S_W}$ represents within-class scatter more accurately because it reflects the distributions of subclasses (mixture components).

A new between-class scatter matrix $\boldsymbol{S_B}$ is defined by considering the concept of subclasses as

$$\boldsymbol{S_B}=\sum_{i=1}^{c}\sum_{k=1}^{s}n_{ik}(\boldsymbol{m}-\boldsymbol{m_{ik}})(\boldsymbol{m}-\boldsymbol{m_{ik}})^t, \tag{13}$$

where n_{ik} is the number of data belonging to the kth mixture component in the ith class. The new between-class scatter matrix $\boldsymbol{S_B}$ also represents the between-class scatter more accurately because it reflects the distributions of subclasses.

From the newly defined scatter matrices $\boldsymbol{S_W}$ and $\boldsymbol{S_B}$, we can obtain the transformation matrix $\boldsymbol{W}$ that maximizes the criterion function

$$J(\boldsymbol{W})=\frac{|\boldsymbol{W^tS_BW}|}{|\boldsymbol{W^tS_WW}|}. \tag{14}$$

The columns of optimal $\boldsymbol{W}$ are the generalized eigenvectors $\boldsymbol{w_i}$ that correspond to the largest eigenvalues in

$$\boldsymbol{S_Bw_i}=\lambda_i\boldsymbol{S_Ww_i}. \tag{15}$$

4.2 Extension of LDA by Class-wise Features (C-LDA)

LDA finds one transformation matrix over the whole data of all the classes. This property makes LDA extract the global features from data. Although it may be the merit of LDA, this property gives the loss of some class-wise information. Therefore, we extend LDA by the class-wise features, called C-LDA, in which they can be used for classification.

We define a within-class scatter matrix for each class. The within-class scatter matrix $\boldsymbol{S_{W_i}}$ for the i th class is defined as

$$\boldsymbol{S_{W_i}}=\sum_{x\in C_i}(\boldsymbol{x}-\boldsymbol{m_i})(\boldsymbol{x}-\boldsymbol{m_i})^t. \tag{16}$$

For the between-class scatter matrix, we use the same between-class scatter matrix $\boldsymbol{S_B}$ as used in LDA such as

$$\boldsymbol{S_B}=\sum_{i=1}^{c}n_i(\boldsymbol{m_i}-\boldsymbol{m})(\boldsymbol{m_i}-\boldsymbol{m})^t. \tag{17}$$

We attempt to find the class-wise transformation matrix $\boldsymbol{W_i}$ using $\boldsymbol{S_{W_i}}$ and $\boldsymbol{S_B}$ for the ith class. We obtain $\boldsymbol{W_i}$ that maximizes the following criterion function

$$J_i(\boldsymbol{W_i})=\frac{|\boldsymbol{W_i^tS_BW_i}|}{|\boldsymbol{W_i^tS_{W_i}W_i}|}. \tag{18}$$

The jth column of optimal $\boldsymbol{W_i}$ is the generalized eigenvector $\boldsymbol{w_{i_j}}$ that correspond to the jth largest eigenvalues in

$$\boldsymbol{S_Bw_{i_j}}=\lambda_{i_j}\boldsymbol{S_{W_i}w_{i_j}}. \tag{19}$$

4.3 Extensions of LDA by PCA Mixture Model and Subclass-wise Features (PM-S-LDA)

Here, we combine the two extensions of LDA mentioned before. The first extension of LDA works well for data with multi-modal Gaussian distribution. The second extension of LDA uses class-wise features. We extend LDA by a combination of two ideas of mixture models and

class-wise features, called PM-S-LDA, in which each class is modeled by the mixture of many components and specific features corresponding to each component are used.

In PM-S-LDA, we apply the PCA mixture model to each class C_i with s mixture components. We then obtained means $\boldsymbol{m}_{ik}$, variances $\boldsymbol{V}_{ik}$, transformation matrixes $\boldsymbol{T}_{ik}$, for the kth mixture component in the ith class. $\boldsymbol{V}_{ik}$ is a diagonal matrix whose diagonal element is eigenvalues $\lambda_{ik,j}$ which is the jth largest eigenvalue of the covariance matrix for the kth mixture component of the ith class. Next, we define the new scatter matrices $\boldsymbol{S}_{\boldsymbol{W}ik}$ and $\boldsymbol{S}_{\boldsymbol{B}}$ as follows.

$$\boldsymbol{S}_{\boldsymbol{W}ik} = \boldsymbol{T}_{ik}\boldsymbol{V}_{ik}\boldsymbol{T}_{ik}^{t} \tag{20}$$

$$\boldsymbol{S}_{\boldsymbol{B}} = \sum_{i=1}^{c}\sum_{k=1}^{s} n_{ik}(\boldsymbol{m}-\boldsymbol{m}_{ik})(\boldsymbol{m}-\boldsymbol{m}_{ik})^{t} \tag{21}$$

where n_{ik} is the number of data belonging to the kth mixture component in the i class.

From the new scatter matrixes $\boldsymbol{S}_{\boldsymbol{W}ik}$ and $\boldsymbol{S}_{\boldsymbol{B}}$, we can obtain the transformation matrix $\boldsymbol{W}_{ik}$ that maximizes the criterion function

$$J_{ik}(\boldsymbol{W}_{ik}) = \frac{|\boldsymbol{W}_{ik}^{t}\boldsymbol{S}_{\boldsymbol{B}}\boldsymbol{W}_{ik}|}{|\boldsymbol{W}_{ik}^{t}\boldsymbol{S}_{\boldsymbol{W}ik}\boldsymbol{W}_{ik}|} \tag{22}$$

The ith column of the optimal $\boldsymbol{W}_{ik}$ is the generalized eigenvector $\boldsymbol{w}_{ik,j}$ that correspond to the largest eigenvalues in

$$\boldsymbol{S}_{\boldsymbol{B}}\boldsymbol{w}_{ik,j} = \lambda_{ik,j}\boldsymbol{S}_{\boldsymbol{W}ik}\boldsymbol{w}_{ik,j} \tag{23}$$

5. Simulation Results and Discussion

5.1 Handwritten digits recognition

We applied LDA and its extensions to handwritten digits recognition. We used UCI handwritten digit data [10]. Some digits in the database are shown in Figure 1. The UCI handwritten digit data have a training set of 3,823 and a test set of 1797. The original image of each digit has the size of 32x32 pixels. It is reduced to the size of 8x8 pixels where each pixel is obtained from the average of the block of 4x4 pixels in the original image. So each digit is represented by a feature vector with the size of 64x1.

We applied LDA, its extensions, and ICA (Independent Component Analysis [12]) to handwritten digit recognition in the following way. For LDA, we transform the data and the means $\boldsymbol{m}_i$ by $\boldsymbol{W}$, and assign the data $\boldsymbol{x}$ to the class C_{LDA} whose corresponding transformed mean is nearest to the transformed data, as

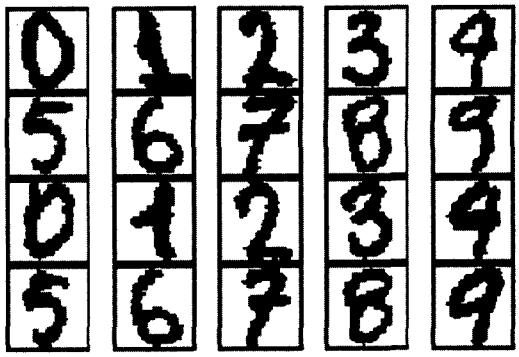

Figure 1 Examples of handwritten digits in UCI database

$$C_{LDA} = \arg\min_i |(\boldsymbol{x}-\boldsymbol{m}_i)\boldsymbol{W}. \tag{24}$$

For PM-LDA, we transform the data x and the means $\boldsymbol{m}_{i,k}$ by $\boldsymbol{W}$ and assign the data to the class C_{PM-LDA} whose corresponding transformed mean is nearest to the transformed data, as

$$C_{PM-LDA} = \arg_i \min_{i,k} |(\boldsymbol{x}-\boldsymbol{m}_{ik})\boldsymbol{W}. \tag{25}$$

For C-LDA, we transform the data x and the means $\boldsymbol{m}_i$ by the $\boldsymbol{W}_i$ of each class, and assign the data to the class C_{C-LDA} whose corresponding transformed mean is nearest to the transformed data by the class-wise transformation matrix $\boldsymbol{W}_i$, as

$$C_{C-LDA} = \arg\min_i |(\boldsymbol{x}-\boldsymbol{m}_i)\boldsymbol{W}_i| \tag{26}$$

For PM-S-LDA, we transform the data $\boldsymbol{x}$ and the means $\boldsymbol{m}_{ik}$ by the corresponding $\boldsymbol{W}_{ik}$. The test data point is assigned to the class $C_{PM-S-LDA}$ whose corresponding transformed mean is nearest to transformed data by the subclass-wise transformation matrix $\boldsymbol{W}_{ik}$ of the kth component in the ith class, as

$$C_{PM-S-LDA} = \arg_i \min_{i,k} |(\boldsymbol{x}-\boldsymbol{m}_{ik})\boldsymbol{W}_{ik}|. \tag{27}$$

For ICA, we transform the data $\boldsymbol{x}$ and the means $\boldsymbol{m}_i$ by $\boldsymbol{Z}$ The test data point is assigned to the class C_{ICA} whose corresponding transformed mean

is nearest to transformed data, as

$$C_{ICA} = L(\text{argmin}_{z_r}|(x - x_r)Z|), \quad (19)$$

Figure 2 plots the classification errors according to a different number of features for the test data using LDA and its extentions. The number of features and classification errors in the best case are shown in Table 1. ICA result is shown only in Table 1 because ICA components do not have any order. Even though more than 2 mixture components have been taken, it does not show any significant improvement in classification peformance. So, we used only two mixture components for learning the PCA mixture model for each class in PM-LDA and PM-S-LDA. PM-LDA and PM-S-LDA have the advantage of extracting more features, since they have the S_B with higher rank. The performances of all the three extensions are better than LDA, and of PM-S-LDA is the best. We ensure that three extensions are better than LDA, but the performance is not so good for handwritten digit recognition (1-NN (1-Nearest Neighbor) (2.00%)), because we use only 10 or 20 means and the number of features is small. In actual experiments, we can obtain at most 9 effective features for LDA and C-LDA, and can obtain at most 19 effective features for PM-LDA and PM-S-LDA, while the orignal number of features is 64. These restrictions make the performance inferior for handwritten digit recognition.

Table 1 The best performance and the corresponding number of features for the handwritten digit recognition

Methods	Number of features	test error
LDA	6	8.68%
PM-LDA	19	7.29%
C-LDA	9	5.84%
PM-S-LDA	15	3.78%
ICA	26	10.96%

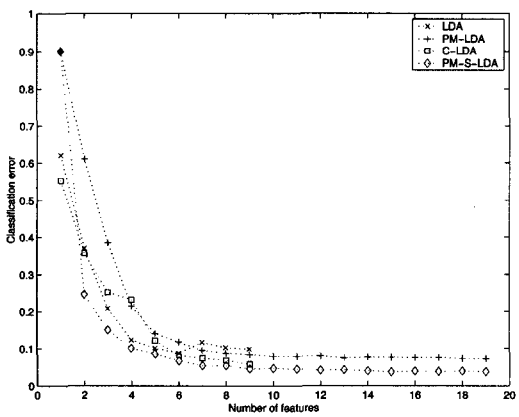

Figure 2 Classification errors vs. the number of features for handwritten digits recognition

5.2 Alphabet recognition

We also applied LDA, its extensions, and ICA to alphabet recognition. Alphabet has 26 classes that is much grater than 10 classes in the digit. Therefore, we expect that more features can be used and that we will obtain better results than the handwritten digit recognition. For each class, 300 data are randomly extracted from ETL-6 database [11]. Some alphabets in ETL-6 database are shown in Figure 3. The whole data of 7800 are subdivided into the set of 5200 for training and the set of 2600 for testing. The original image of each letter has a size of 64x63 pixels. The image is resized into 64x64 by inserting one additional line of white pixels into the last line. It is reduced to the size of 8x8 pixels, where each pixel is obtained from the average of the block of 8x8 pixels in the original image. So each letter is represented by a feature vector with the size of 64x1.

We applied LDA and its extensions to alphabet recognition in the same way as in the classification

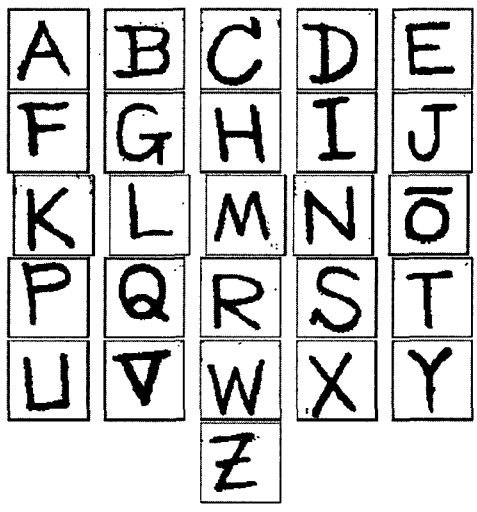

Figure 3 Examples of alphabets in ETL-6 database

of handwritten digit data. Figure 4 plots the classification errors according to different number of features for the testing data using LDA and its extensions. The best classification errors and the corresponding number of features are shown in Table 2. ICA result is shown only in Table 2 because ICA components do not have any order. Even though more than 2 mixture components have been taken, it does not show any significant improvement in classification peformance. Therefore, we used only two mixture components for learning the PCA mixture model for each class in PM-LDA and PM-S-LDA. The performances of all the three extensions are better than that of LDA, and C-LDA is the best. C-LDA seems to show a better performance than PM-S-LDA because the distribution of the alphabet data is relatively simple. In actual experiments, we can obtain at most 25 effective features for LDA and C-LDA, and can obtain at most 51 effective features for PM-LDA and PM-S-LDA, while the orignal number of features is 64. In this case, extensions of LDA perform quite well. C-LDA outperforms 1-NN (1-Nearest Neighbor) (2.92%), where the features of 8x8 pixels were used and all the training set data are used for classification.

Table 2 The best performance and the corresponding number of features for alphabet recognition

Methods	Number of features	test error
LDA	17	10.96%
PM-LDA	19	9.73%
C-LDA	25	1.88%
PM-S-LDA	22	4.19%
ICA	23	29.96%

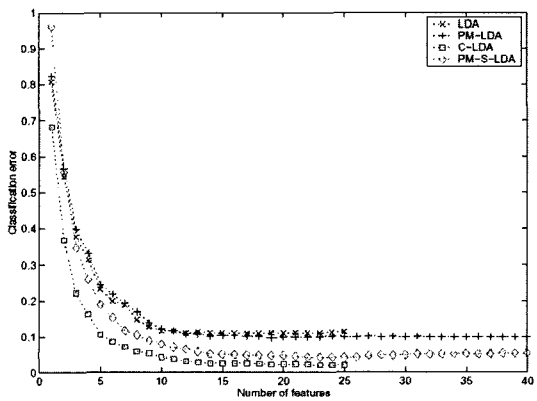

Figure 4 Classification errors vs. the number of features for alphabet recognition

6. Conclusion

We proposed three extensions of LDA by using mixture models and class-wise features in order to overcome the underfitting problem of LDA. The first PM-LDA, models the within-class scatter more accurately by using a PCA mixture model. The second, C-LDA, uses class-wise transformation matrix, and extract features in a class-wise manner. The third, PM-S-LDA combines two ideas; that is, models each class by the PCA mixture model and extract features in the subclass-manner by using the subclass-wise transformation matrix. In the simulation, all the proposed extensions showed better classification performance than LDA.

There are two problems with the proposed extensions. First, the maximum number of effective features are too small for the data with a small number of classes. Second, the extensions may not work well for data in which have a small number of samples in each class. In such cases, there may be a singularity problem for the within-class scatter matrix in C-LDA and PM-S-LDA, and PCA mixture models may not be learned well in PM-LDA and PM-S-LDA. In the future, we will attempt to solve these two problems.

참 고 문 헌

[1] Duda, R., Hart, P., 1974. Pattern classification and scene analysis. Wiley, New York.

[2] Belhumeur, P., Hespanha, J., Kriegman, 1997. D. Eigenfaces vs.Fisherfaces: class specific linear projection, IEEE Transactions on PAMI, 19(7), 711-720.

[3] Swets, D., Weng, J., 1996. Using Discriminant Eigenfeatures for Image Retrieval, IEEE Transactions on PAMI, 18(8), 831-836.

[4] Hastie, T. and Buja, A. and Tibshirani, R., 1995. Penalized discriminant analysis, Annals of Statistics, 23, 73-102.

[5] Hastie, T., Tibshirani, R., 1996. Discriminant Analysis by Gaussian Mixtures, Journal of the Royal Statistical Society: series-B.

[6] Baudat, G. and Anouar, F., 2000, Generalized

discriminant analysis using a kernel approach. Neural Computation, 12(10), 2385-2404.
[7] Jacobs, R., Jordan, M., Nowlan, S., Hinton, G., 1991. Adaptive mixtures of local experts. Neural Computation. 3, 79-87.
[8] Tipping, M., Bishop, C., 1999. Mixtures of probabilistic principal component analyzers. Neural Computation, 11, 443-482.
[9] Dempster, P., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society: series-B. 39(4), 1-38.
[10] Blake, C., Merz, C., 1998. UCI Repository of Machine Learning Databases. Department of Information and Computer Science, University of California, Irvine, CA.
[11] ETL Character Database, Image Understanding Section, Electrotechnical Laboratory, 1-1-4, Umezono, Tsukuba, Ibaraki, 305, Japan.
[12] Hyvrinen A. and Oja E., "A Fast Fixed-Point Algorithm for Independent Component Analysis, Neural Computation, vol. 9, no. 7, pp. 1483-1492, 1997.

김 현 철
1999년 포항공대 컴퓨터공학 학사, 수학 학사(복수전공). 2001년 포항공대 컴퓨터공학 석사. 2001년 이후 포항공대 컴퓨터공학 박사과정 재학. 관심분야는 패턴인식, 기계학습 등

김 대 진
1981년 연세대학교 전자공학과 학사 1983년 KAIST 전기 및 전자공학과에서 석사. 1991년 미국 Syracuse University 컴퓨터공학과에서 박사를 받았음. 한국방송공사 기술연구소, 동아대학교 컴퓨터공학과에 근무하다가 1999년 7월 이후 포항공과대학교 컴퓨터공학과에서 근무. 관심분야는 패턴인식, 지능형 시스템, 생체 인식 등

방 승 양
1966년 일본 Kyoto대학 전기공학에서 학사. 1969년 서울대학교 전기공학에서 석사. 1974년 미국 University of Texas 전산학에서 박사를 받았음. 미국 Wayne State University, NCR, Bell 연구소 등에서 근무하다가 1981년 귀국. 한국전자기술연구소 시스템부 실장, 부장 역임. (주)유니온시스템 전무. 1986년부터 포항공대 컴퓨터공학과 교수. 현재 뇌연구센터 소장. 관심분야는 패턴인식, 신경회로망