Proceedings of KFIS Spring Conference 2005 Volume 15, Number 1

클래스 정보를 이용한 PCA 기반의 특징 추출

PCA-based Feature Extraction using Class Information

박명수, 나진희, 최진영
서울대학교 전기컴퓨터공학부

Myoung Soo Park, Jin Hee Na, Jin Young Choi
School of Electrical Engineering and Computer Science
Seoul National University
E-mail : mspark@neuro.snu.ac.kr

요 약

영상 데이터와 같은 대용량의 데이터를 분류하고자 할 경우, 입력 데이터의 차원을 줄여서 특징 벡터를 뽑아내는 전처리 과정은 필수적이다. 이 경우 특징 벡터가 입력 데이터의 정보를 최대한 포함하도록 하는 것이 중요하다. 특징 벡터를 뽑는 대표적인 방법으로는 PCA, ICA, LDA, MLP와 같은 특징 추출(feature extraction) 방법을 들 수 있다. PCA와 LDA는 무감독 학습 방식이고, LDA, MLP는 감독 학습 방식에 해당한다. 감독학습 방식의 경우 입력 정보와 함께 클래스 정보를 사용하기 때문에 데이터를 분류하기에 더 좋은 특징들을 뽑아낼 수 있는 장점이 있다. 본 논문에서는 무감독 학습 방식인 PCA에 클래스에 대한 정보를 함께 사용하여 특징을 추출함으로써 데이터 분류에 더욱 적합한 특징들을 뽑는 방법을 제안하였다. 그리고, Yale face database를 사용하여 제안한 알고리즘의 성능을 기존의 알고리즘과 비교, 테스트하였다.

1. 서론

최근에 영상데이터와 같은 대용량의 데이터를 분류하기 위해 특징 선택(feature selection) 기법이나 특징 추출(feature extraction) 기법[1]들이 많이 이용되고 있다. 이 기법들의 주된 목적은 주어진 입력 차원보다 작은 차원을 가진 특징(feature)들을 뽑아내는데 있다. 이러한 입력 차원의 축소에 대한 문제는 데이터 마이닝 등의 분야에서도 중요하게 다루어지는데[2], 이를 통해 입력차원이 클 경우 발생할 수 있는 curse of dimensionality의 문제를 피할 수 있고, 보다 좋은 분류 성능을 기대할 수 있다.

특징 추출에 해당하는 기법으로는 PCA[3], ICA[4], LDA[5] 등의 통계적인 방법과 MLP[6, 7]를 이용하는 방법 등이 있다. PCA, ICA와 같은 방법은 클래스에 대한 정보를 이용하지 않는 무감독(unsupervised) 방식으로, 주어진 입력 데이터로부터 분산이나 정보량을 기반으로 특징들을 얻어내는 일반적인 방법이다. 그런데, 이러한 경우 얻어진 특징들은 분류(classification) 문제에 적합하지 않는 특징들일 수도 있다. 이러한 문제점을 해결하기 위해 클래스에 대한 정보를 이용하는 LDA, MLP 등의 감독(supervised) 방식들이 개발되었다. LDA의 경우는 분류 문제에 적용하기에 적합한 단순하고 강력한 방법이지만, 특정한 경우(예: 클래스 간에 큰 차이가 있을 경우)에만 좋은 특징을 얻을 수 있다는 한계가 있다. MLP를 이용한 경우도 지역 최소점(local minima)에 빠지는 문제 등의 학습상의 한계로 항상 좋은 성능이 보장되는 것은 아니다. 이러한 문제점들을 고려하여, 최근에는 클래스 정보를 이용할 수 있게 ICA를 변형한 ICA-FX[8]와 같은 기법이 제안되었다. ICA-FX는 입력정보와

함께 클래스 정보를 이용하여 feature를 추출하는 기법으로 PCA와 ICA에 비해 좋은 성능을 가지고 있으며, LDA에 비해 보다 일반적인 경우에도 적용가능하다는 장점이 있다.

그런데, ICA-FX를 실제 응용 문제에 적용할 경우에는 LDA나 그밖의 여러 알고리듬들에서와 같이 계산량을 줄이기 위해 PCA를 함께 이용할 필요가 있다[9, 10]. 즉, PCA와 LDA, PCA와 ICA-FX를 결합시켜, PCA를 이용하여 일차적으로 특징들을 추출하여 입력 차원을 줄인 후에, 다시 LDA나 ICA-FX를 이용하는 것이다. 이 경우 입력 데이터의 차원을 줄이는데 있어서 PCA가 큰 역할을 담당하게 되는데, 다음과 같은 문제점들이 발생한다. 첫째로 구현할 때 두 가지 알고리듬을 별도로 구현해야 하는 어려움이 따르며 둘째로 처음 PCA는 뒤의 LDA나 ICA-FX와는 다른 기준(criterion)에 따라 특징을 추출하기 때문에 PCA에 의해 추출된 정보가 ICA-FX나 LDA에 사용하기에 적절한 형태가 아닐 수 있고, 이로 인한 side effect가 발생할 수도 있다.

본 논문에서는 기존의 방법들이 가지는 문제점들을 해결하고, 클래스 정보를 이용하여 좀 더 적합한 특징을 추출할 수 있도록, PCA를 이용한 새로운 특징추출 scheme인 PCA-FX를 제안하고자 한다. 제안한 방법은 실제 문제에 있어서 PCA와 결합하여 이용할 경우에도 PCA를 반복해서 이용하면 되므로 구현상의 부담이 적다. 또한 같은 종류의 기준에 의해 특징이 추출되기 때문에, side effect의 발생 가능성이 낮다. 실험에서는 제안한 알고리듬의 특징 추출 성능을 평가하기 위해 YALE face database를 이용하여 기존의 알고리듬과 비교, 분석하였다.

2. PCA-FX Scheme

PCA는 자료들의 분포를 고려하여 분산이 최대가 되도록 하는 방향성분, 즉 PC(Principal Component)를 찾아내는 방법이다. 찾아낸 PC를 basis로 하고 그 위에 데이터를 투영시키면 데이터들을 구분하기에 적합하도록 하는 분산이 큰 형태로 변환할 수 있다. PC는 데이터의 공분산 행렬(covariance matrix)의 고유치 문제(eigenvalue problem)를 통해 구할 수 있다. 이 경우, 가장 큰 고유치에 해당하는 고유 벡터가 첫 번째 PC이며, 두 번째 PC는 두 번째 고유치에 해당하는 고유 벡터가 된다. 대부분의 경우 큰 고유치에 해당하는 PC일수록 데이터들을 구분하기 위한 정보가 많이 포함되므로, PC를 적절히 선택하면 입력차원에 비해 작은 수의 PC만으로도 데이터의 분포를 작은 오차범위 이내에서 기술할 수 있다. 즉, PC를 이용하면 입력 데이터를 차원이 작은 특징들로 변환할 수 있다.

그러나 PCA를 통해 얻어진 이러한 특징들은 클래스 정보를 이용하지 않고 추출된 것이므로, 경우에 따라서는 데이터를 분류하기에 적합하지 않을 수도 있다. 아래의 그림 1을 살펴보면 주어진 데이터 사이의 분산을 최대로 하는 방향성분인 Z_1은 클래스 간의 분산을 최대로 하는 방향성분인 Z_2와는 방향이 전혀 다름을 알 수 있다. 이와 같은 경우 데이터들을 적절히 분류하기 위해서 클래스 정보를 포함하는 특징 추출 방법이 필요하게 된다.

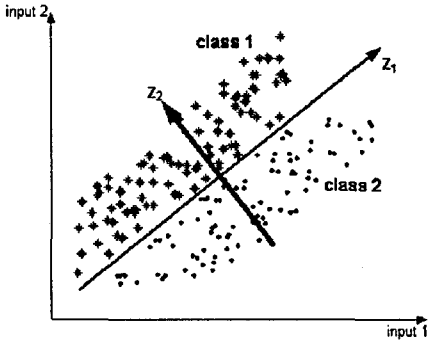

그림 1 클래스 정보가 필요한 예

PCA-FX는 클래스 정보를 이용한 PCA 기반의 특징 추출 scheme이다. 이 방법은 기존의 방법들에 비해 더 일반적인 문제에 적용할 수 있으며, PCA와 같은 criteria를 사용하기 때문에 실제 응용 문제에 PCA와 함께 사용할 경우, ICA-FX나 LDA에 비해 side effect가 일어날 가능성이 적다. PCA-FX는 다음과 같은 세단계로 구성된다.

- 데이터에 class 정보를 추가하는 단계
- PCA를 이용하여 특징을 추출하는 단계
- 변환 행렬 W를 결정하는 단계

첫 번째 단계는 입력 데이터에 클래스에 대한 정보를 추가하는 과정이고, 두 번째 단계는 첫 번째 단계의 결과를 이용하여 PCA를 수행하는 단계로, 공분산 행렬을 구하고 정보를 많이 포함하고 있는 PC를 선택하는 과정을 포함한다. 마지막 단계는 얻어진 PC들로부터 특징을 추출하는 단계이다. 특징추출을 위한 전체적인 구조는 그림 2와 같다.

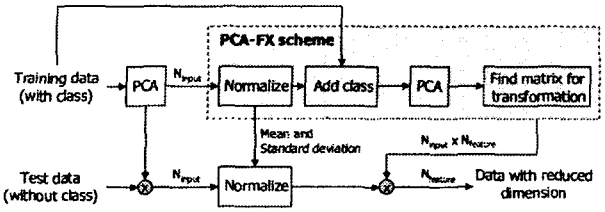

그림 2 PCA + PCA-FX Scheme

2.1 입력 데이터에 클래스 정보를 추가하는 단계

클래스의 수를 N_{class}라고 하고, 입력 데이터에 대한 클래스 정보를 $C(X)=[C_1, C_2, \cdots, C_{N_{class}}]$라고 하자. 이 때, X가 i번째 클래스에 속하면, C_i 값은 P_i를 갖고, C_i가 아닌 나머지는 N_i 값을 갖는다. P_i와 N_i값은 모든 X에 대하여 C_i가 정규분포를 갖도록 결정한다. 예를 들어, X_1, X_2, X_3가 주어지고 이들 각각의 클래스 레벨이 1, 2, 2이면, 클래스 정보 $C(X_1)=[P_1 \quad -N_2]$, $C(X_2)=[-N_1 \quad P_2]$, $C(X_3)=[-N_1 \quad P_2]$와 같이 표현된다. 그리고, C_1과 C_2는 모든 X에 대하여 정규분포를 가져야 하므로, P_1, $-N_1$, $-N_1$의 평균과 $-N_2$, P_2, P_2의 평균이 모두 0이고, 표준편차는 모두 1이 되도록 P_1, P_2, N_1, N_2 값을 결정하면 된다.

X에 C(X)를 추가할 경우, C(X)와 X가 같은 범위의 값을 가지도록 X도 평균이 0이고 표준편차가 1이 되도록 정규화할 필요가 있다. 그리고, X의 각 element들의 평균과 분산값을 구하여, 새로운 X가 들어왔을 때, 이 값들을 통해 정규화시킬 수 있도록 한다. 정규화된 X를 X_{norm}이라고 하고, C(X)를 추가한 X를 X_{aug}라고 하면

$$X_{aug}=[X_{norm} \quad C(X)] \tag{1}$$

와 같이 나타낼 수 있다. 식(1)에서 입력의 차원을 N_{input}, C(X)의 차원을 N_{class}라고 할 때, X_{aug}의 차원은 $1\times(N_{input}+N_{class})$로 주어진다.

2.2 PCA에 기반한 특징 추출 단계

X_{aug}에 [3]에서와 같은 표준 PCA를 적용하면 차원이 $1\times(N_{input}+N_{class})$인 PC들을 구할 수 있다. 그리고, 입력차원 N_{input}을 $N_{feature}$로 줄일려면 큰 고유치를 가진 $N_{feature}$개의 PC를 선택하면 된다. 이러한 PC들로 이루어진 W_{aug}는 다음과 같이 정의된다.

$$W_{aug}=[PC_1^T \; PC_2^T \cdots \; PC_{N_{feature}}^T] \tag{2}$$

$$=\begin{pmatrix} W_{input} \\ W_{class} \end{pmatrix} \tag{3}$$

여기서 PC_i의 차원은 $1\times(N_{input}+N_{class})$이고, W_{aug}의 차원은 $(N_{input}+N_{class})\times N_{feature}$ 이다. 식(3)의 W_{input}과 W_{class}는 각각 입력과 클래스 정보에 해당한다.

2.3 변환 행렬 W를 결정하는 단계

2.2에서 구한 X_{aug}와 W_{aug}를 곱하면 아래의 식(4),(5)에서와 같이 특징 공간으로의 변환된 데이터를 얻을 수 있다.

$$X_{feature}=[X \; C(X)]\times\begin{pmatrix} W_{input} \\ W_{class} \end{pmatrix} \tag{4}$$

$$=X\times W_{input}+C(X)\times W_{class} \tag{5}$$

그런데, 테스트 데이터에는 클래스에 대한 정보 C(X)가 주어지지 않으므로 위의 식을 그대로 적용할 수 없으므로, 근사화된 식(6)을 이용하여 $X_{feature}$를 계산한다.

$$X_{feature}\simeq X\times W_{input} \tag{6}$$

3. 실험 결과

주어진 알고리듬의 성능평가를 위해 Yale database를 이용하여 얼굴인식 실험을 수행하였다. Yale database는 15사람에 대한 165개의 흑백 이미지로 구성되어 있으며, 각 사람마다 11가지의 표정을 포함되어 있다. Yale database에는 얼굴에 맞추어 잘려진 것과 전체 얼굴을 담고 있는 것의 두가지 종류가 있는데, 본 실험에서는 첫 번째 database를 그림 3에서와 같이 21×30로 downsampling 하여 이용하였다. 제안한 알고리듬의 입력으로 사용할 때에는 1×630의 차원을 가진 벡터로 변환하였다.

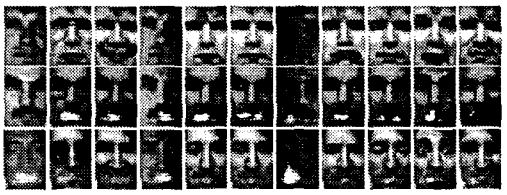

그림 3 Yale database 샘플

제안한 알고리듬의 성능을 평가하기 위해서 leave-one-out 방법을 사용하였다. 이 방법은 전체 데이터가 n개 있을 경우에, n-1개의 데이터는 학습에 사용하고 나머지 한 개의 데이터를 이용하여 테스트 하는 방법이다. 분류를 위해서는 Euclidean distance에 따른 nearest neighborhood classifier를 이용하였고, 실험을 통해 다음과 같은 두 가지 특성을 조사해 보았다.

- 첫 번째 단계의 PCA를 통해 얻은 PC의 수와 PCA-FX에 의해 얻은 특징 수에 따른 분류 오차율
- PCA, LDA, ICA-FX에 의한 특징들의 비교

실험 결과의 구체적인 내용은 다음과 같다.

3.1 분류 오차율 분석

PCA-FX를 통해 얻어진 특징들의 분류 성능은 첫 번째 PCA를 통해 얻은 PC의 수에 따라 다르게 나타났다. 이를 통해 앞에서 언급한 것과 같이 첫 번째 PCA가 두 번째 특징 추출 알고리듬에 영향을 준다는 사실을 확인할 수 있었다. 이를 통해 특징들의 수에 따른 분류 오차율과 분류에 가장 적합한 특징들의 개수를 알아보기 위해서는 첫 번째 PCA를 통해 얻어지는 PC의 수도 적절하게 선택되어야 함을 알 수 있다. 추출해야 할 적절한 PC 및 특징의 수를 결정하기 위해 첫 번째 PCA로부터 얻은 PC의 수와 PCA-FX로부터 얻은 특징의 수를 바꾸어 보면서 실험해 보았다. 그 결과는 표 1에서와 같다. 이 경우 진한 숫자는 같은 수의 PC에서의 최소 오차율을 나타낸다. 그림 4는 표 1의 결과를 그래프로 나타낸 결과이다.

PC 개수	특징의 개수						
	9	10	11	12	13	14	15
18	13.94						
19	13.33	11.15					
20	13.94	**09.70**	10.91	10.91		16.97	16.97
21	10.91	10.91	**09.09**	10.30		10.91	
22	12.73	10.91	11.52	**10.30**	11.52	11.52	
23	11.52	12.73	12.12	11.52	**10.91**	12.73	
24	11.52	09.70		11.52	**06.06**	06.67	09.09
25	10.30	12.73		12.73	**10.30**	12.12	
26	11.52	10.91	09.70	09.70	**08.48**	09.09	09.70
27	10.30	10.91		11.52	**07.27**	07.27	09.70
28	10.30	08.48	09.09	**07.88**	09.70	07.88	09.09
29	12.73	10.30	08.48	**06.67**	09.70	06.67	09.70
30	11.52	11.52		08.48	**07.88**	07.88	10.91
40			09.09	10.30	**08.48**	09.09	09.09

표 1 PCA-FX의 PC, 특징 수에 따른 오차율

이 결과를 통해 PC의 개수가 일정할 경우, 오차율이 특징의 개수가 증가함에 따라 감소하다가 특징의 수가 어떤 임계치에 이르면 다시 증가함을 볼 수 있다. 그리고 오차율이 최소가 되는 특징의 개수는 항상 클래스의 수인 15보다 작게 나타남과 동시에 일정한 경향성을 보이며 나타남을 확인할 수 있다. 본 실험에서는 PC의 수가 24개이고 특징의 수가 13일 때 오차율이 가장 낮게 나타났다.

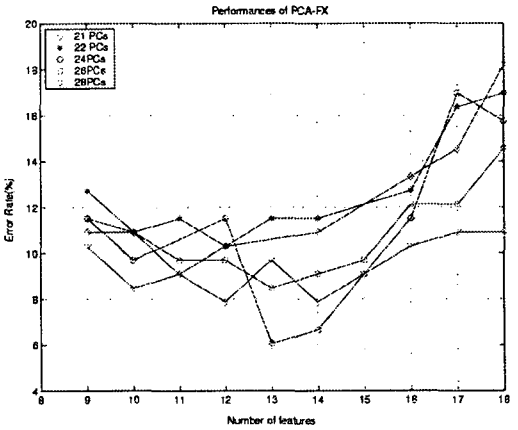

그림 4 특징의 수와 오차율과의 관계

3.2 PCA-FX와 PCA, LDA, ICA-FX의 결과 비교

앞에서 설명한 것 같이 PCA-FX의 성능은 PC의 수에 의존하여 달라졌는데, 실험 결과 이러한 경향은 다른 알고리듬에서도 동일하게 나타났다. 그러나 오차율이 최소가 되는 지점은 알고리듬에 따라 다르게 나타났다. 예를 들어 그림 5에서 보는 바와 같이 ICA-FX는 PCA-FX에 비해 PC나 특징의 개수가 더 큰 지점에서 최소 오차율을 보였다.

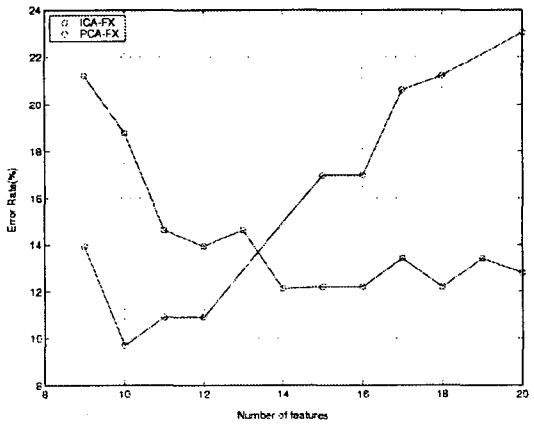

그림 5 PC의 수가 20일 때, PCA-FX와 ICA-FX의 오차율 비교

그림 4에서 나타난 경향성에 주목하여, PCA, LDA, ICA-FX에서도 PC와 특징의 개수를 변화

시켜 가면서 실험해 보았다. 그 결과 각 알고리듬에 있어서의 최소 오차율과 최소 오차율이 나타나는 지점이 표 2에서와 같이 나타났다.

알고리듬	PC 수	특징 수	최소 오차율
PCA	22	9	27.27%
LDA	24	13	12.73%
ICA-FX	30	20	6.06%
PCA-FX	24	13	6.06%

표 2 PC와 특징의 수에 따른 최소 오차율

표 2의 결과를 통해 PCA-FX가 기존의 알고리듬에 비해 적은 수의 PC와 특징들을 가지고 더 좋은 성능을 보임을 확인할 수 있다.

4. 결론

본 논문에서는 기존의 알고리듬들의 문제점을 지적하고, 이러한 문제점들을 극복하기 위해 PCA-FX scheme을 제안하였다. PCA-FX는 PCA를 이용하여 특징 벡터를 뽑을 때, 클래스에 대한 정보를 입력 정보와 함께 사용하여 분류 성능을 향상시키는 방법이다. 실제로 Yale face database를 이용하여 알고리듬의 성능을 평가해 본 결과, 기존의 다른 알고리듬들에 비해 적은 수의 특징과 적은 계산량으로도 낮은 오차율을 얻을 수 있었다.

향후 과제로는 첫 번째로, PCA를 통해 얻어진 특징의 수와 오차율 간의 관계를 이론적으로 분석해 보고자 한다. 이 과정을 통해 PCA가 어떤 식으로 정보를 압축하는지 살펴보고, 데이터 압축에 있어서의 PCA와 ICA의 차이점을 규명해 보고자 한다. 두 번째로, 앞의 그래프를 참조하여 오차율과 특징 수와의 관계를 이론적으로 추정할 수 있는지 살펴보고자 한다. 만약 오차율을 최소로 하기 위해 필요한 특징의 수가 몇 개인지 이론적으로 알 수 있다면, trial-and-error의 과정 없이 최소의 오차율을 갖게 하는 PC와 특징의 수를 구할 수 있을 것이다.

감사의 글 : 본 연구는 산업자원부 차세대 신기술개발사업(수퍼지능칩 및 응용기술 개발 과제)의 지원을 받아 수행되었습니다.

5. 참고문헌

[1] K. J. Cios, W. Pedrycz, and R. W. Swiniarski, Data mining methods for knowledge discovery, chapter 9, Kluwer Academic Publishers, 1998.

[2] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Advances in Knowledge Discovery and Data Mining, AAAI and the MIT Press, 1996.

[3] I. T. Joliffe, Principal Component Analysis, Springer-Verlag, 1986.

[4] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," Neural Computation, vol. 7, no. 6., June 1995.

[5] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 2nd edition, 1990.

[6] H. Lu, R. Setiono, and H. Liu, "Effective data mining using neural networks," IEEE Transaction on Knowledge and Data Engineering, vol. 8, no. 6, Dec. 1996.

[7] R. Setiono, and H. Liu, "A connectionist approach to generating oblique decision trees," IEEE Transaction on Systems, Man, and Cybernetics - Part B: Cybernetics, vol. 29, no. 3, June 1999.

[8] Nojun Kwak and Chong-Ho Choi, "Feature extraction based on ICA for binary classification problems," IEEE Transaction on Knowledge and Data Engineering, Vol. 15, No. 6, pp. 1374-1388, Nov. 2003.

[9] Nojun Kwak, Chong-Ho Choi, and Narendra Ahuja, "Face recognition using feature extraction based on independent component analysis," ICIP2002, Rochester, Sep. 2002.

[10] W. Zhao, R. Chellappa and A. Krishnaswamy, "Discriminant analysis of principal components for face recognition," Proceedings of 3rd IEEE International Conference on Automatic Face and Gesture Recognition, pp.336-341, April 1998.

[11] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711-720, July 1997.