

Chapter 09 패턴인식 시스템 설계

[학습목표]

이 장에서는 지금까지 배운 내용을 활용하여 실제 응용데이터를 위한 인식기를 설계해 본다. 사용하는 데이터는 패턴인식 및 기계학습 분야에서 벤치마크 데이터로 널리 사용하는 얼굴데이터이며, 적절한 전처리를 거친 데이터로부터 특징을 추출하고 베이지 분류기를 이용하여 분류를 수행하는 인식기를 개발할 것이다.

9.1 얼굴인식 문제

9.2 특징추출기의 학습

9.2.1 영상데이터를 위한 주성분분석

9.2.2 PCA에 의한 특징추출 - 아이겐페이스

9.2.3 LDA에 의한 특징추출 - 피셔페이스

9.3 분류기의 설계

9.4 매트랩에 의한 인식기 구현

연습문제

9. 패턴인식 시스템 설계

9.1. 얼굴인식문제

이 절에서는 지금까지 살펴본 내용을 실제 응용데이터에 적용하여 패턴인식 시스템을 설계해 볼 것이다. 사용데이터는 최근 패턴인식 분야에서 활발히 연구되고 있는 얼굴데이터를 사용한다. 얼굴 데이터는 표정, 조명, 각도 등의 다양한 변형이 존재하여 기계학습과 컴퓨터 비전 분야에서 활발히 연구되는 데이터이다. [그림 9-1]에 다양한 변형을 가진 얼굴데이터를 보여주고 있다. 이를 이용하면 얼굴인식 뿐 아니라 포즈인식, 표정인식 등 다양한 목적에 맞추어 데이터를 분석하고 분류를 수행해 볼 수 있다. 여기서는 포즈의 변화가 있는 데이터 집합을 이용하여 얼굴인식을 수행할 것이다.



[그림 9-1] 다양한 변형을 가진 얼굴 영상(왼쪽부터 각도 변화, 표정 변화, 조명 변화)



[그림 9-2] 학습데이터의 예

[그림 9-2]에 학습에 사용된 데이터의 예를 보이고 있다. 이는 얼굴인식 문제의 벤치마크 데이터로 널리 사용되는 FERET 데이터의 일부이다. 각 사람에 대해 -60도부터 60도까지 15도씩 포즈를 변형시킨 9개의 데이터가 존재하여 총 450개의 데이터를 사용하였다. 얼굴인식 문제를 수행하므로 분류대상이 되는 클래스는 50개이며, 한 클래스당 9개의 데이터가 존재한다.

수집된 데이터를 이용하여 인식기를 설계하기 위해 먼저, 전체 데이터를 학습용과 테스트용으로 구분해야 한다. 여기서는 각 사람당 60도, 0도, 그리고 -60도의 세 데이터를 학습을 위해 사용하고, 나머지 6가지 변형 데이터는 테스트로 사용할 것이다. 따라서 학습데이터의 수는 총 150개이며, 테스트 데이터는 총 300개가 존재한다.



[그림 9-3] 크기 조정된 학습데이터

주어진 영상은 흑백영상으로, 조명, 위치 등에 대한 기본적인 전처리는 행해진 상태이다. 그러나 얼굴 아래 부분의 상반신은 얼굴영상에서 불필요한 부분이므로 이 영역을 제거하고 동시에 영상의 사이즈를 줄이기 위하여 좌우 다섯 픽셀씩, 상하 열 픽셀씩 제거하여 얼굴만으로 구성된 영상을 얻는 전처리를 수행하였다. 이 결과 얻어진 영상이 [그림 9-4]에 나타나 있다. 원래 영상의 크기는 70×50 으로 입력 벡터는 3500차원이었으나, 간단한 전처리 후 얻어진 영상의 크기는 50×40 으로 입력벡터는 2000차원이 된다. 따라서 전체 학습 데이터 집합은 150×2000 크기의 행렬 X 로 나타내고, 테스트 데이터 집합은 300×2000 크기의 행렬로 나타낸다.

2000차원의 입력 데이터를 그대로 사용하는 것은 계산량의 증가 및 메모리의 증가를 초래하므로, 여기서는 주성분분석 (PCA)을 통해 차원을 축소하고 특징을 추출하는 과정을 거칠 것이다. 또한 PCA에 의해 찾아진 특징에 대하여 선형판별분석법(LDA)을 적용하여 분류에 적합한 특징을 추출할 것이다. 특징추출과 관련된 내용은 다음절에서 자세히 기술하겠다.

9.2 특징추출기의 학습

9.2.1 영상데이터를 위한 주성분분석

2000차원의 입력 데이터 x 에 대해 주성분분석을 수행하기 위해서는 먼저 150개의 학습데이터 집합을 이용하여 공분산 행렬 Σ 를 계산한 후, 이에 대한 고유치분석을 실시해야 한다. 그런데 2000차원의 데이터에 대한 공분산 행렬은 2000×2000 이 되어, 이에 대한 고유치 분석은 많은 계산량을 요구한다. 따라서 영상 데이터와 같이 입력 차원이 지나치게 큰 데이터에 대해서는 변형된 방식을 통하여 주성분을 추출할 필요가 있다.

150×3500 크기의 데이터 행렬 X 에 대한 공분산 행렬 Σ 를 계산하는 대신, 이 절에서는 그 전치행렬 X^T 를 활용하여 계산량을 줄이면서 원래 데이터 행렬 X 에 대한 주성분을 얻는 방법을 소개한다. 먼저 데이터 행렬 X 와 그 평균 M 을 이용하여 다음과 같은 행렬 S 를 정의한다.

$$S = \frac{1}{N} (X - M) (X - M)^T \quad [\text{식 9-1}]$$

여기서 N 은 데이터의 수이다. 이 행렬은 그 크기가 150×150 으로 고유치 분석이 쉽게 이루어질 수 있다. 고유치 분석을 통하여 찾아진 행렬 S 에 대한 고유벡터행렬과 고유치행렬을 각각 V 와 Λ 라고 두면, 고유치정의를 의하여 다음과 같은 성질을 만족한다.

$$SV = \Lambda V \quad [\text{식 9-2}]$$

[식 9-2]에 [식 9-1]의 정의를 대입하면 다음과 같은 관계식을 얻을 수 있다.

$$\frac{1}{N} (X - M) (X - M)^T V = \Lambda V \quad [\text{식 9-3}]$$

다시 [식 9-3]의 양변에 행렬 $(X - M)^T$ 를 곱해주면 다음과 같은 관계식을 얻게 된다.

$$\frac{1}{N} (X - M)^T (X - M) (X - M)^T V = \Lambda (X - M)^T V \quad [\text{식 9-4}]$$

여기서 원래 데이터의 공분산 행렬 Σ 는 $(X - M)^T (X - M) / N$ 가 됨에 주목하면, 우리가 얻고자 하는 고유벡터는 $(X - M)^T V$ 로 주어지고, 고유치는 Λ 가 됨을 알 수 있다. 결국, 행렬 $(X - M) (X - M)^T / N$ 의 고유벡터를 구한 다음 여기에 $(X - M)^T$ 를 곱해줌으로써 다음과 같이 간단하게 원하는 변환행렬 W 를 얻을 수 있다.

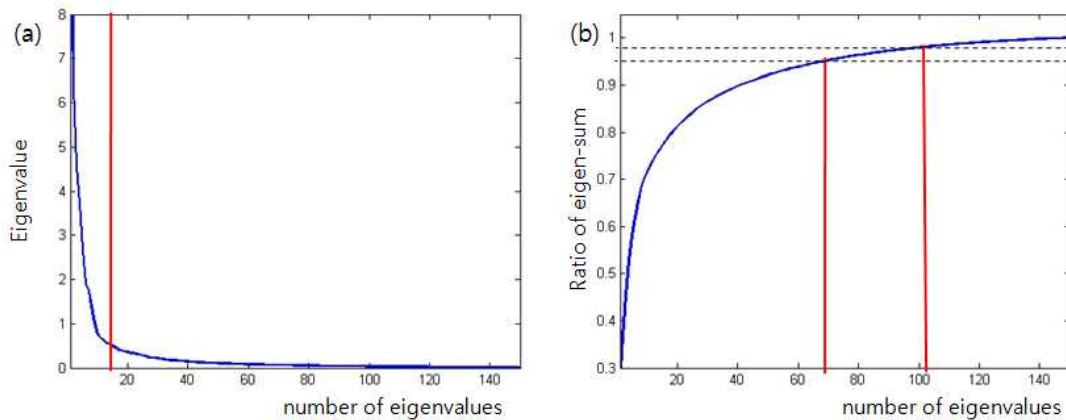
$$W = (X - M)^T V \quad [\text{식 9-5}]$$

마지막으로 변환행렬을 데이터에 곱해줌으로써 새로운 특징을 다음과 같이 얻게 된다.

$$Y = W^T X \quad \text{[식 9-6]}$$

그런데 이렇게 얻어진 행렬 W 는 직교행렬의 조건을 만족하지 못한다는 사실에 유의하자. 추후 W 의 직교성일 필요한 경우에는 W 의 기저벡터를 직교화시키는 연산을 추가적으로 수행하기도 한다. 이에 대해서는 9.4절의 프로그램에서 그 수행 예를 알아보겠다.

변환행렬을 찾는 과정에서 한 가지 더 유의해야 할 것은 W 의 크기이다. 만약 단순한 고유치분해 과정을 거쳐서 W 를 얻었다면, 그 크기는 2000×2000 이 되어, 2000차원의 고유벡터 2000개를 열벡터로 가지는 행렬이 될 것이다. 그런데 여기서는 먼저 V 를 구하고, 이로부터 W 를 얻었으므로, 얻을 수 있는 고유 벡터의 수는 150개로 제한된다. 따라서 W 의 크기는 [식 9-5]로부터 2000×150 이 되어 2000차원의 고유벡터 150개를 열벡터로 가지는 행렬임을 알 수 있다. 결국 찾을 수 있는 고유벡터의 수는 데이터의 수에 제약을 받게 된다.



[그림 9-4] 고유치 값들의 변화

(a) 크기순으로 나열된 고유치 값의 변화 (b) 고유치값 선택을 위한 고유치 합 비율의 변화

그런데 우리는 차원을 축소하는 것이 목적이므로, 이렇게 찾아진 고유벡터들을 모두 사용하는 대신 그 중 일부를 선택하여 사용하게 된다. 따라서 변환을 수행하기 전에 몇 개의 고유벡터를 사용하여 변환행렬 W 를 정의할 것인지를 정해야 한다. 다시 말하면 몇 차원의 특징 벡터 y 를 사용할 것인지를 결정해야 한다. 일반적으로 주성분 분석법에서 특징의 차원을 결정하는 기준으로는 고유치를 활용한다. 고유치 값은 해당 고유벡터 방향으로 사영된 데이터의 분산값을 나타내므로, 고유치 값이 큰 것부터 차례대로 선택한다.

[그림 9-4a]에 고유치 값을 그 크기순으로 나열하여 그래프로 나타내었다. 그래프로부터 고유치 값은 처음 15개 정도까지 어느 정도 큰 값을 가지고, 그 이후에는 값이 급격하게 줄어들음을 알 수 있다. 이와 같이 고유치값 그래프에서 꺾인점 ([그림 9-4a]의 수직선으로 표시된 위치)을 찾아서 그 지점까지의 고유치와 고유벡터만을 사용하는 방법이 경험적 방법으로 사용되고 있다.

다른 방법으로는 8장의 [식 8-20]에서 정의된 $r(m, n) = \frac{\sum_{i=1}^m \lambda_i}{\sum_{i=1}^n \lambda_i}$ 값을 이용

하여 특징 수를 결정할 수 있다. [그림 9-4b]에 m 값에 따른 $r(m,n)$ 값의 변화를 그래프로 나타내었다. [그림 9-4b]에서 첫 번째 수직선은 $r(m,n)$ 값이 0.95보다 큰 경우로 이때의 m 값 69이다. 두 번째 수직선은 $r(m,n)$ 값이 0.98보다 큰 경우로 이때의 m 값 101이다. 이 값의 의미는 m 이 69일 때 정보 손실은 5% 미만인 것을 나타내는 것이다.

9.2.2 주성분분석에 의한 특징추출 결과 - 아이겐페이스

앞 절의 주성분분석을 통해 찾아진 특징을 이용하여 분류를 수행하기에 앞서, 그 특징의 의미에 대하여 조금 더 알아보겠다. [식 9-6]으로부터 새로운 특징벡터들로 이루어진 데이터 행렬 Y 와 원래 데이터 행렬 X , 그리고 변환행렬 W 의 관계에 대해서 생각해 보자. 주성분 분석을 통해 찾은 변환행렬 W 가 직교변환인 경우, [식 9-6]의 양변에 W 를 곱해주면 다음과 같은 관계식을 얻을 수 있다.

$$WY = X \quad \text{[식 9-7]}$$

전체 데이터 행렬에 대해 나타난 [식 9-7]에서 하나의 데이터벡터 x 와 그에 대한 특징벡터 $y = [y_1, y_2, \dots, y_m]$ 의 관계식을 찾으면 다음과 같이 나타낼 수 있다.

$$x = Wy = y_1w_1 + y_2w_2 + \dots + y_mw_m \quad \text{[식 9-8]}$$



[그림 9-5] 주성분 분석에 의해 찾아진 아이겐페이스

여기서 w_1, w_2, \dots, w_m 은 고유치벡터들이다. 결국 이 식으로부터 원래 데이터 x 는 주성분분석으로 찾은 고유치벡터들을 기저벡터로 하고, 그 특징값을 계수로 하는 형태로 다시 나타낼 수 있음을 알 수 있다. 이 때 고유치벡터들은 원래 영상과 같은 크기의 2000차원임에 주목하여, 이것을 하나의 영상데이터로 보고 이미지로 출력한 것이 [그림 9-5]에 나타나 있다. [그림 9-5]는 고유치 값의 순서대로 총 70개의 고유벡터를 영상으로 출력한 것인데, 결국 이 영상들이 일반적인 얼굴데이터를 표현하는 기저 얼굴 영상의 역할을 한다고 볼 수 있다. 이런 의미에서 얼굴인식에 주성분분석을 적용하여 찾아진 고유벡터들을 특별히 <아이겐페이스(Eigenface)>라고 부른다. [그림 9-5]는 [식 9-8]의 관계를 아이겐페이스를 이용하여 개념적으로 표현한 것이다.

[그림 9-6] 아이겐페이스와 특징값의 선형합에 의한 얼굴데이터 표현

그런데, 여기서 한 가지 유의할 점은, 차원축소를 위해 사용하는 W 는 정방행렬이 아니고 n 차원 데이터를 m 차원으로 변환하는 $n \times m$ 행 행렬이므로, 직교행렬의 조건 $WW^T = W^TW = I$ 를 만족하지 못한다는 것이다. 따라서 [식 9-7]과 [식 9-8]에서 나타난 등식은 완벽하게 성립하지는 못하며, [식 9-8]의 우변의 식이 좌변의 원래 입력데이터 x 를 근사하는 값이 된다. 이를 얼굴영상에 대해 적용하면, [그림 9-6]의 가장 왼쪽에 나타난 입력데이터는 오른쪽에 나타난 고유얼굴들의 선형결합에 의해 얻어지는 영상에 의해 근사될 수 있음을 의미한다. 이 때 원래 영상과 근사된 영상의 차, 즉 정보의 손실량은 특징의 차원 수 (즉, 베이스 벡터의 수)에 의존한다.

9.2.3 선형판별분석에 의한 특징추출 결과 - 피셔페이스

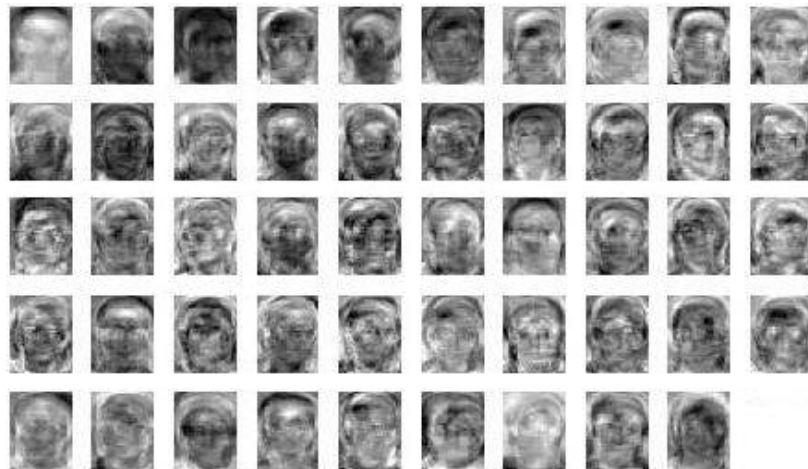
이번에는 특징추출의 또 다른 방법으로 선형판별분석법에 의한 특징추출을 수행해보자. 8장에서 언급한 바와 같이 입력데이터의 차원이 데이터의 수보다 큰 경우에는 클래스내 산점행렬 S_W 가 특이행렬이 되어 역행렬을 얻을 수 없다. 이러한 문제를 해결하기 위해서 이 장에서는 입력데이터를 그대로 사용하는 대신 주성분분석법에 의해 차원축소된 특징을 입력벡터로 사용한다. 특징의 차원은 앞서 고유치합의 비율 $r(m, n)$ 이 0.95보다 커지는 m 의 값인 69를 선택하였다. [식 9-6]로부터 얻어진 69차원의 특징데이터 집합 Y 에 대하여 선형판별분석을 실시한다.

특징데이터 집합 Y 와 클래스라벨 정보를 이용하여 Y 에 대한 클래스내 산점 행렬 S_W 와 클래스간 산점 행렬 S_B 를 계산한 후, $S_W^{-1} S_B$ 의 고유치벡터를 구함으로써 선형판별분석법의 변환행렬 U 를 얻을 수 있다. U 의 크기는 Y 에 의존하여 69×69 의 행렬이 되나, 8장에서 언급한 바와 같이 S_B 의 랭크는 클래스 수에 의존하므로, 고유치값이 0보다 커지는 고유벡터는 49개가 존재한다. 따라서 U 는 69차원의 고유벡터가 49개 존재하는 69×49 크기의 변환행

렬이 된다. 이를 이용하여 특징추출을 수행하는 것은 다음 식에 의해서 가능하다.

$$Z = U^T Y = (WU)^T X \quad [\text{식 9-9}]$$

이 식에서 원래 데이터 집합 X 에 대한 변환행렬은 $(WU)^T$ 로 볼 수 있고, 이것이 선형판별 분석법에 의한 변환행렬이 된다. 그 크기는 2000차원의 벡터 49개로 이루어진 2000×49 가 된다. 주성분분석법의 경우와 마찬가지로 2000차원의 벡터들은 하나의 영상으로 다시 표현할 수 있고, 이것이 [그림 9-7]에 나타나 있다. 주성분분석법의 아이겐페이스(Eigenface)와 마찬가지로 [그림 9-7]에 나타난 영상을 <피셔페이스(Fisherface)>라고 한다. 이 이름은 선형판별분석법의 또 다른 이름인 <피셔의 선형판별 (Fisher's Linear discriminant)>로부터 유래된 것이다.



[그림 9-7] 선형판별분석에 의해 찾아진 피셔페이스

9.3 분류기의 설계

주성분분석에 의해 얻어진 특징데이터 Y 와 선형판별분석에 의해 얻어진 특징데이터 Z 를 이용하여 분류를 수행하기 위해서 지금까지 살펴본 다양한 분류기들을 적용할 수 있을 것이다. 가우시안함수를 가정하는 베이즈 분류기에 대해서는 2장에서 숫자데이터에 대한 분류를 수행해 보았으므로, 여기서는 비모수밀도추정에 기반한 최근접이웃 분류기를 사용한다.

최근접이웃 분류기는 분류대상이 되는 데이터와 저장된 모든 학습데이터와의 거리를 비교하여 가장 가까운 이웃을 찾고, 그 이웃의 클래스로 할당하는 방법이다. 구현이 간단하면서도 비선형 결정경계를 찾을 수 있다는 장점을 가지고 있다. 이 문제에서는 총 300개의 테스트 데이터 각각에 대하여 저장된 150개의 학습데이터들과 거리를 비교한 후, 가장 가까운 이웃을 찾는다. 이때 거리를 계산하는 거리 함수로는 단순한 유클리디안 거리를 사용하였다.

또한 앞 절에서의 논의에서는 특징데이터 Y 와 Z 는 각각 69차원과 49차원으로 정하였으나, 실제 문제에서는 가능한 모든 차원에 대하여 테스트 한 후 가장 적절한 차원을 선택하는 것

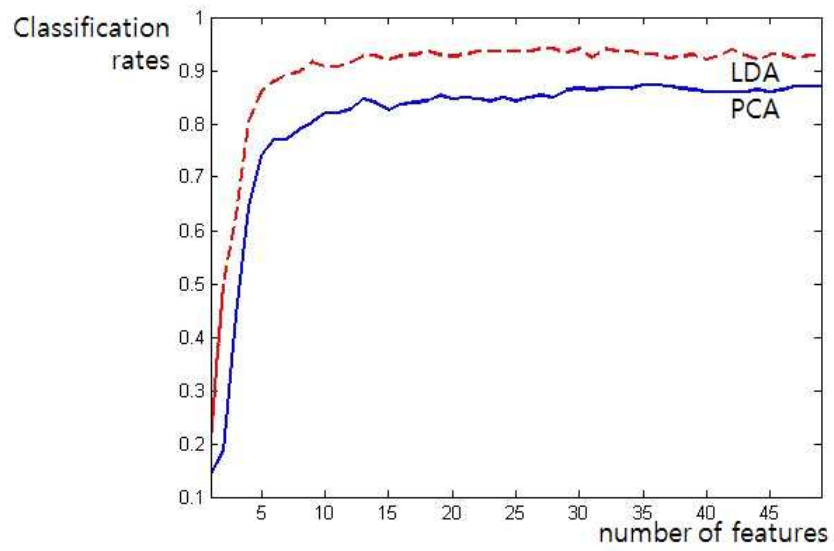
이 바람직하다. 다음 절에서 이를 처리하는 프로그램 코드를 제시하였다.

9.4 프로그램코드 및 수행 결과

[프로그램 9-1]에는 주성분분석과 선형판별분석에 의한 특징추출 프로그램코드를 제시하였다. 저장된 학습데이터와 테스트 데이터를 불러 온 후 학습데이터의 평균을 계산하고, 이어서 학습데이터 행렬의 전치행렬을 이용하여 정의되는 행렬 S 을 얻어서 이에 대한 고유치분석을 통해 고유치벡터 행렬 V 를 얻는다. 다시 이 V 에 학습데이터 행렬을 곱해줌으로써 원하는 변환행렬 W 를 얻을 수 있다. 이렇게 얻어진 변환행렬은 직교행렬의 성질을 만족하지 못하므로, W 가 나타내는 부분공간과 같은 공간을 표현하는 직교기저벡터들로 이루어진 직교행렬로 변환하는 과정을 수행한다. 이 과정을 거치면 주성분 분석법에 의한 특징 추출을 위한 준비가 완료되었다.

이어서 고유치 값의 합의 비율 $r(m,n)$ 이 0.95보다 커지는 특징의 차원을 찾아 주성분분석에 의해 해당 차원의 특징데이터 Y 를 찾는다. 찾아진 특징데이터 집합을 선형판별분석법에 적용한다. 클래스내 산점행렬과 클래스간 산점행렬을 계산하여 고유치분석을 통해 변환행렬과 특징데이터를 얻는다. 이상과 같이 찾아진 변환행렬과 특징데이터 Z 는 파일에 저장해 둔다.

[프로그램 9-2]에서는 저장된 변환행렬과 특징데이터를 읽어와 최근접이웃 분류기에 의해 분류를 수행한다. 1부터 49차원까지 특징의 차원을 변화시키면서 300개 테스트 데이터에 대하여 분류를 수행하여 분류율을 계산하고, 이를 그래프로 나타낸다. [그림 9-8]에 프로그램의 수행으로 얻어지는 특징차원에 따른 분류율의 변화 그래프를 나타내었다. 그림에서 주성분분석에 의해 얻어진 특징에 비해 선형판별분석에 의해 얻어지는 특징을 사용하는 것이 더 높은 분류율을 보임을 알 수 있다. 또한 주성분분석법에 의해 얻어진 특징의 경우 최대 분류율을 87.33%로 이는 원래 입력데이터를 이용한 경우의 분류율 86.33%보다 높았다. 이는 주성분분석법을 이용한 차원축소를 통해 단순히 적은 차원으로 원래 정보를 나타내는 효과 뿐 아니라 인식에 불필요한 잡음을 없애는 효과도 함께 가져올 수 있음을 의미한다. 이상에서는 단순한 최근접이웃 분류기를 이용하여 분류를 수행하였으나, 이 책의 후반부에서 소개되는 다양한 비선형 분류기들을 이용하면 보다 높은 분류율을 얻을 수 있을 것이다.



[그림 9-8] 특징 수에 따른 분류율의 변화

프로그램 9-1 PCA and LDA for Face Recognition		
주성분분석과 선형판별분석에 의한 특징추출		
001	load data_person_trim	% 데이터 불러오기
002	Ntrn=size(train_trim,1);	% 학습데이터의 수
003	Ntst=size(test_trim,1);	% 테스트데이터의 수
004	lsize=size(train_trim,2);	% 입력벡터의 차원
005	Mcls = 50;	% 클래스의 수
006		
007	%%%%%%%% PCA Feature Extraction	
008	X=train_trim;	% 학습데이터 행렬
009	meanfig = mean(X);	% 학습데이터 행렬의 평균
010	M=repmat(meanfig,size(X,1),1);	% 평균벡터를 데이터 수만큼 반복한 행렬
011	S=(X-M)*(X-M)'/Ntrn;	% 고유치분석을 위한 행렬을 계산
012	[V,D,U]=svd(S);	% 고유치분석
013	W=(X-M)'*V;	% 고유벡터 찾기
014		
015	%%%%%%%% LDA Feature Extraction	
016	eval=diag(D);	% PCA로 축소할 차원 결정
017	for i=1:150	
018	if ((sum(eval(1:i))/sum(eval))>0.95) break end	
019	end	
020	Ydim=i;	
021	Wo=orth(W(:,1:Ydim));	% 변환행렬의 직교화
022	Y=(Wo'*(X))';	% PCA에 의한 차원축소 (학습데이터)
023	Yt=(Wo'*(test_trim))';	% PCA에 의한 차원축소 (테스트데이터)
024	Sw=zeros(Ydim);	% Within-Scatter Matrix의 계산
025	for i=1:Mcls	
026	C=Y((i-1)*3+1:i*3,:);	
027	Sw = Sw+3*cov(C);	
028	m(i,:)=mean(C);	
029	end	
030	Sb=Mcls*cov(m);	% Between-Scatter Matrix의 계산
031	[Vf,Df,Uf]=svd(inv(Sw)*Sb);	% LDA 변환행렬 찾기
032	Z=(Vf'*Y)';	% LDA에 의한 특징추출 (학습데이터)
033	Zt=(Vf'*Yt)';	% LDA에 의한 특징추출 (테스트데이터)
034		
035	save feature_person W Vf Y Yt Z Zt;	% 추출된 특징 저장

프로그램 9-2 Face Recognition Using PCA and LDA

주성분분석과 선형판별분석에 의한 특징추출

```

001 load data_person_trim      % 데이터 불러오기
002 load feature_person        % 특징 불러오기
003 Ntrn=size(train_trim,1);    % 학습데이터의 수
004 Ntst=size(test_trim,1);     % 테스트데이터의 수
005 lsize=size(train_trim,2);   % 입력벡터의 차원
006 Mcls = 50;                  % 클래스의 수
007
008 % Classification using PCA features and LDA features
009 for dim=1:Mcls-1             % 가능한 모든 특징 차원에 대하여 반복 수행
010   Wo=orth(W(:,1:dim));        % 정한 차원만큼 주성분벡터행렬 직교화
011   Y=(Wo'*(X'))';              % PCA 특징추출 (학습데이터)
012   Yt=(Wo'*(test_trim'))';     % PCA 특징추출 (테스트데이터)
013   for i=1:Ntst                % 각 테스트 데이터에 대해 분류 시작
014     yt=Yt(i,:);               % 테스트 데이터에 대한 PCA 특징
015     zt=Zt(i,1:dim); %Yt(:,i); % 테스트 데이터에 대한 LDA 특징
016     for j=1:Ntrn              % 학습데이터들과의 거리계산
017       dy(j)=norm(yt-Y(j,1:dim));
018       dz(j)=norm(zt-Z(j,1:dim));
019     end
020     [minvy, miniy]=min(dy);    %최근접이웃 찾기 (PCA 특징)
021     [minvz, miniz]=min(dz);    %최근접이웃 찾기 (LDA 특징)
022     min_labely(i)=train_label(miniy); %최근접이웃의 클래스로 할당 (PCA)
023     min_labelz(i)=train_label(miniz); %최근접이웃의 클래스로 할당 (LDA)
024   end
025   error_labely=find(min_labely~test_label); %분류율 계산 (PCA)
026   correcty=Ntst-size(error_labely,2);
027   classification_ratey(dim)=correcty/Ntst;
028   error_labelz=find(min_labelz~test_label); %분류율 계산 (LDA)
029   correctz=Ntst-size(error_labelz,2);
030   classification_ratez(dim)=correctz/Ntst;
031   [classification_ratey(dim), classification_ratez(dim)]
032 end
033 % 특징 수에 따른 분류율 변화 그래프 그리기
034 figure(1);
035 plot([1:49], classification_ratey); hold on
036 plot([1:49], classification_ratez,'r');

```

연습문제

1. 다양한 패턴 인식 문제들을 생각해 보고, 각 문제에 적합한 방법론을 생각해 보시오.
2. 이 장에서 사용한 선형 특징추출법의 한계와 그 해결방법에 대해 생각해 보시오.
3. 이 장에서 사용한 최근접이웃 분류기의 한계와 그 해결방법에 대해 생각해 보시오.

참고자료

이 장에서는 대표적인 패턴인식 문제의 하나인 얼굴인식을 위한 시스템을 구현해 보았다. 이 장에서 사용한 얼굴인식용 벤치마크 데이터인 FERET 데이터는 아래 제시한 사이트를 참고하기 바란다. 또한 얼굴인식과 관련된 다양한 데이터베이스와 방법론 및 최근 논문들이 링크된 사이트로 Face Recognition 홈페이지가 있다. 이밖에 패턴인식과 기계학습 연구자들을 위한 다양한 벤치마크 데이터들이 제공되는 사이트로 UCI Repository가 있다.

The Facial Recognition Technology (FERET) Database
(http://www.itl.nist.gov/iad/humanid/feret/feret_master.html)
Face Recognition homepage (<http://www.face-rec.org/>)
UCI Repository <http://archive.ics.uci.edu/ml/>