# Chapter 08 특징추출

**[학습목표]**
패턴인식에 있어서 인식기의 성능을 좌우하는 중요한 요소 중 하나가 특징추출이다. 이 장에서는 선형변환에 의한 특징추출 방법으로 주성분분석법과 선형판별분석법에 대하여 알아본다.

# 8. 특징추출

## 8.1. 선형변환에 의한 특징추출

특징추출은 지금까지 주로 논의한 분류기나 군집분석기를 수행함에 앞서서, 데이터를 보다 다루기 쉬운 형태로 변환하는 과정이라고 생각할 수 있다. 원래 데이터를 그대로 쓰는 대신, 인식에 핵심이 되는 정보만을 추출하거나, 데이터에 포함된 잡음을 제거 하는 등 다양한 형태의 특징 추출 과정이 존재한다. 1장에서 설명한 바와 같이 특징추출 방법들은 특정한 데이터 집합에 특화된 경우가 많다. 그러나 이 절에서는 일반적인 데이터에 적용할 수 있는 선형 변환에 의한 특징 추출 방법에 대하여 알아본다.

선형변환에 의한 특징 추출 방법의 가장 큰 목적 중의 하나는 차원 축소라고 할 수 있다. 즉, $n$차원 입력 데이터를 $m$차원 특징 벡터로 매핑하는 선형 변환을 수행할 때 특히 $m \ll n$이 되도록 하는 저차원의 특징 공간을 찾게 된다. 특징 추출에 있어서 차원 축소가 중요한 첫 번째 이유는, 영상데이터와 같은 고차원의 데이터의 경우 계산량이 급격히 증가한다는 것이다. 간단한 예로, 3장에서 배운 밀도함수의 추정에 대해 생각해 보자. 모수적 추정방법에서 가우시안 확률분포를 모델로 사용한 경우, 입력공간의 차원 수 $n$에 따라 추정해야 하는 파라미터의 수가 정해지는데, 그 개수는 $n^2$에 비례하여 늘어나게 된다. 비모수적 추정방법의 경우는 그 의존도가 더욱 심각해진다. 예를 들어 히스토그램법을 사용하는 경우, $n$차원 실수 공간 $[0,1]^n$을 입력공간으로 가지고, 각 구간의 간격을 $h$로 두었을 때, 빈도수를 측정해야 하는 영역의 개수는 $(1/h)^n$개가 되어 기하급수적으로 늘어나게 된다. 이와 같이 입력 차원이 늘어남에 따라 데이터 처리에 많은 비용이 들 뿐 아니라 정확한 추정의 정확도도 저하된다. 이러한 현상을 <차원의 저주 (curse of dimensionality)>라고 하며, 패턴인식 분야에서 중요한 문제 중의 하나이다. 이에 덧붙여, 원래 주어진 입력데이터를 그대로 활용하는 경우에는 인식에 불필요한 정보까지도 함께 사용하게 되어 결과적으로 인식률 저하의 원인이 되기도 한다. 따라서 특징추출 과정에서는 원래 데이터의 차원을 줄이면서 인식에 핵심이 되는 정보만을 뽑은 것이 주된 목적이 된다.

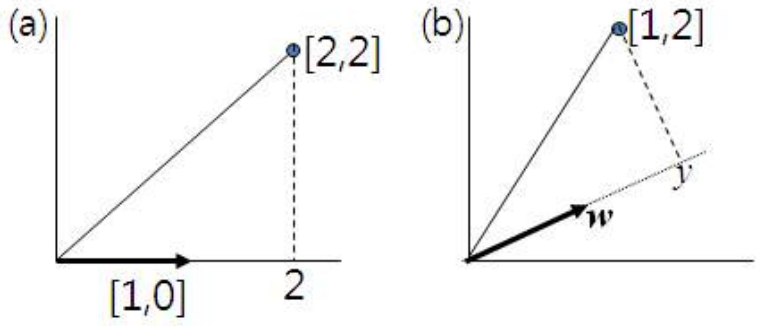
차원축소의 가장 대표적인 방법이 선형변환에 의해 저차원 특징공간으로 매핑하는 <부분공간분석 (Subspace analysis)>이다. 선형변환을 대상으로 하므로, 변환함수는 행렬 $\mathbb{W}$로 정의될 수 있다. 즉, $n$차원 입력벡터 $\boldsymbol{x}$는 $n \times m$ 크기의 변환 행렬 $\mathbb{W}$에 의해 $m$차원 특징 벡터 $\boldsymbol{y}$로 변환될 수 있다. 이를 식으로 나타내면 다음과 같다.

$$\boldsymbol{y} = \mathbb{W}^T \boldsymbol{x} \qquad \text{[식 8-1]}$$

이를 풀어서 쓰면 다음 [식 8-2]와 같이 나타낼 수 있다.

$$\boldsymbol{y} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_m]^T \boldsymbol{x} = \begin{bmatrix} \boldsymbol{w}_1^T \\ \boldsymbol{w}_2^T \\ \vdots \\ \boldsymbol{w}_m^T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_1^T \boldsymbol{x} \\ \boldsymbol{w}_2^T \boldsymbol{x} \\ \vdots \\ \boldsymbol{w}_m^T \boldsymbol{x} \end{bmatrix} = \mathbb{W}^T \boldsymbol{x} \qquad \text{[식 8-2]}$$

따라서 선형변환 행렬 $\mathrm{W}$의 각 열은 입력 벡터 $\boldsymbol{x}$를 사영시킬 벡터가 되고, 사영을 통해 얻어지는 값들이 특징벡터 $\boldsymbol{y}$를 이루게 됨을 알 수 있다. 2차원 데이터를 1차원으로 변환하는 예를 [그림 8-1]에 나타내었다. [그림 8-1a]에서 선형변환 행렬은 크기가 2×1인 벡터로 $\mathrm{W}=[1,0]^T$으로 둘 수 있다. 이를 통해 얻어지는 특징값은 결국 $\boldsymbol{x}$를 $\mathrm{W}$방향 (즉, 가로축)으로 사영한 경우의 좌표값이 된다. 보다 일반적인 변환행렬을 사용한 경우가 [그림 8-1b]에 나타나 있다. 하나의 열벡터 $\boldsymbol{w}$로 이루어진 변환행렬 $\mathrm{W}=\boldsymbol{w}$에 의한 변환 결과로 얻어지는 특징값 $y$는 해당 변환행렬(이 경우에는 벡터 $\boldsymbol{w}$ )로 사영하여 얻어지는 크기 값이 된다.

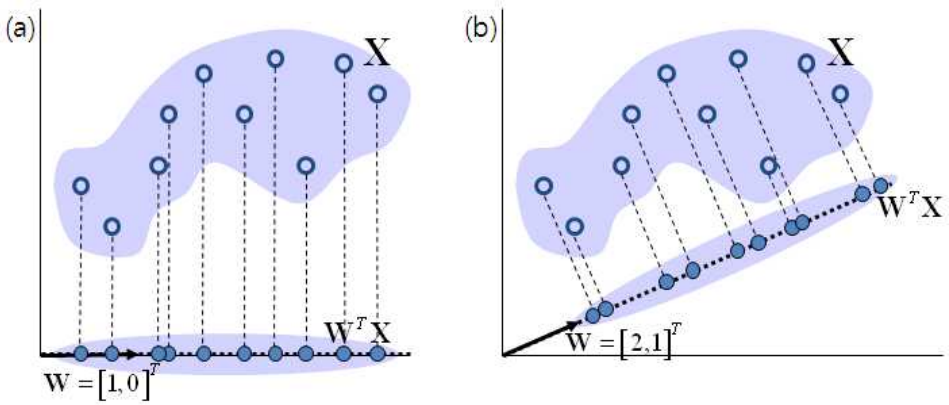



[그림 8-1] 2차원 입력벡터의 1차원 특징벡터로의 변환

지금까지는 하나의 데이터 $\boldsymbol{x}$에 대한 변환을 생각하였는데, 이를 전체 데이터 집합으로 확장하면 $n\times N$ 크기의 행렬 $\mathrm{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_N]$ 로 나타나는 데이터 집합에 대하여 변환행렬 $\mathrm{W}$에 의한 특징추출과정은 다음과 같이 식으로 나타낼 수 있다.

$$\mathrm{Y} = \mathrm{W}^T\mathrm{X} = [\mathrm{W}^T\boldsymbol{x}_1, \mathrm{W}^T\boldsymbol{x}_2, \ldots, \mathrm{W}^T\boldsymbol{x}_N] \qquad \text{[식 8-3]}$$

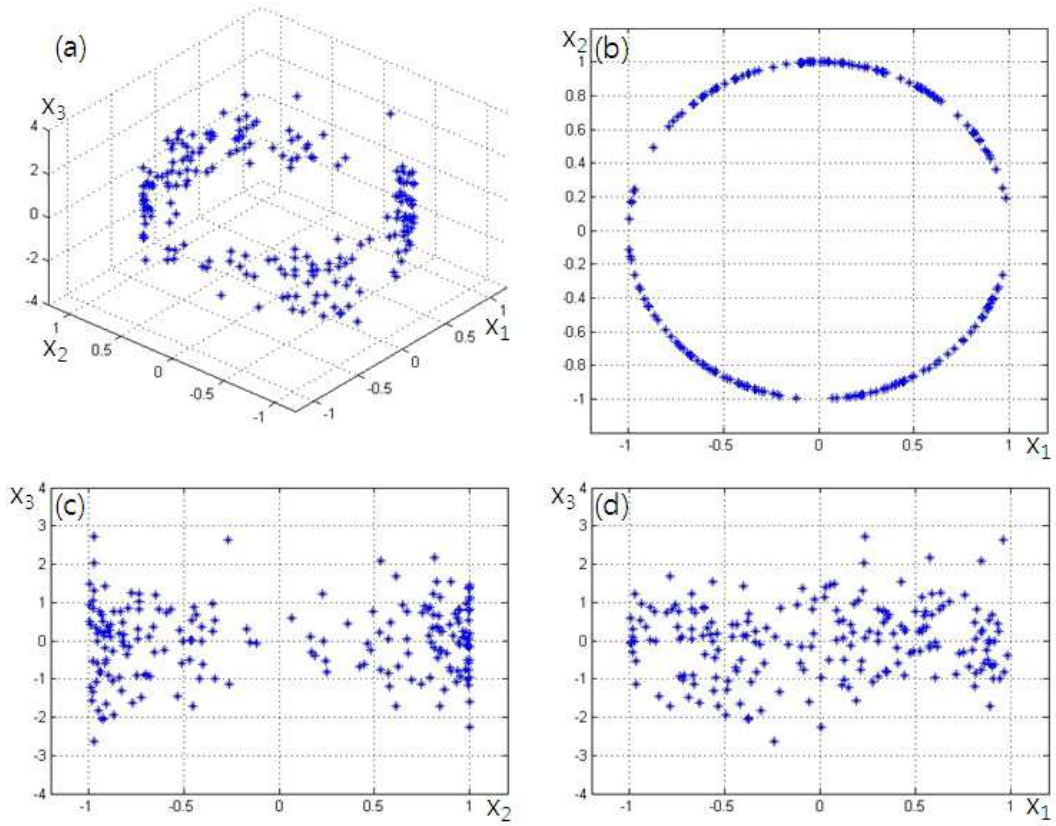
여기서 $\mathrm{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_N]$ 는 특징추출에 의해 얻어진 $m\times N$크기의 특징행렬이다. 2차원 데이터 집합에 대한 특징 추출의 예를 [그림 8-2]에 나타내었다. [그림 8-2a]에서는 '**o**' 기호로 나타난 2차원 데이터에 대해, 변환 행렬 $\mathrm{W}=[1,0]^T$로 선형변환하여 얻어진 1차원 특징값들을 가로축에 표시하였다. [그림 8-2b]에서는 변환행렬 $\mathrm{W}=[2,1]^T$를 이용하여 얻어진 특징값을 $\mathrm{W}$ 방향의 직선상에 표시하였다.




[그림 8-2] 2차원 데이터 집합의 선형변환에 의한 특징추출

이상에서 살펴본 바와 같이 선형변환에 의한 특징추출이란, 주어진 데이터를 변환행렬 $\mathbf{W}$에 의해 정해지는 방향으로 사영함으로써 저차원 특징값을 얻는 것을 말한다. 이때 변환 행렬 $\mathbf{W}$의 각 열은 사영할 저차원 부분공간의 기저를 이루어 해당 부분공간을 정의하고 있다. [그림 8-3]에서는 3차원 데이터의 2차원 부분공간으로의 사영에 의한 특징추출의 예를 보여주고 있다. (a) 나타난 3차원 데이터로부터 서로 다른 기저 행렬 $\mathbf{W}_a, \mathbf{W}_b, \mathbf{W}_c$를 이용하여 각각 2차원으로 선형변환한 결과가 (b), (c), (d)에 나타나 있다. 이 때 사용한 기저벡터는 다음과 같이 정의된다.

$$\mathbf{W}_a = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{W}_b = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{W}_c = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \qquad \text{[식 8-4]}$$




[그림 8-3] 3차원 데이터 집합의 특징추출

[그림 8-3]에서 알 수 있듯이. 어떤 변환행렬을 사용하느냐에 따라 얻어지는 특징의 분포형태가 확연하게 달라진다. 결국 좋은 특징추출이란, 목적에 맞는 부분공간을 찾는 것이며, 이것은 다시 말해서 변환행렬 $\mathbf{W}$ (혹은 변환행렬의 각 열로 나타나는 부분공간의 기저)를 적절히 조정함으로써 분석의 목적에 맞는 특징을 추출하는 것이 필요함을 의미한다. $\mathbf{W}$를 찾는 방법과 찾아진 $\mathbf{W}$는 분석의 목적에 맞추어 변하게 되는데, 이 장에서는 가장 대표적인 특징추출 방법으로 주성분분석법과 선형판별분석법에 대하여 알아보겠다.

# 8.2 주성분분석법

## 8.2.1 공분산 분석에 의한 선형변환

이 절에서 알아보고자 하는 주성분분석법은 기본적으로 데이터의 공분산에 대해 분석하여 그 성질을 바탕으로 선형변환을 수행한다. 따라서 주성분 분석법의 구체적인 알고리즘을 알아보기에 앞서, 원래데이터의 공분산과 선형변환으로 얻어진 특징데이터의 공분산의 관계에 대하여 알아보고, 이를 바탕으로 한 두 가지 선형변환을 소개한다.
입력데이터 집합에 대한 행렬 $\mathrm{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_N]$ 과 선형 변환 행렬 $\mathrm{W}$가 주어졌을 때, 선형변환을 통해 얻어지는 특징 행렬은 $\mathrm{Y} = \mathrm{W}^T\mathrm{X}$와 같이 계산될 수 있다 ([식 8-3] 참조). $\mathrm{X}$의 평균이 벡터 $\boldsymbol{\mu}_x$로 주어지고, 공분산이 행렬 $\boldsymbol{\Sigma}_x$로 주어졌을 때 특징데이터 집합 $\mathrm{Y}$에 대한 평균 $\boldsymbol{\mu}_y$는 다음과 같이 얻을 수 있다.

$$\boldsymbol{\mu}_y = \frac{1}{N}\sum_{i=1}^{N}\mathrm{W}^{\mathrm{T}}\boldsymbol{x}_i = \mathrm{W}^T\boldsymbol{\mu}_x \qquad \text{[식 8-5]}$$

여기서는 공분산에 대한 분석을 수행할 것이므로, $\boldsymbol{\mu}_x$=0 으로 가정하고 논의를 시작한다. 실제 데이터를 다루는 경우에는 입력데이터 $\mathrm{X}$를 사용하는 대신 각 데이터 $\boldsymbol{x}_i$에서 평균을 뺀 행렬 $\mathrm{X}-\boldsymbol{\mu}_x\mathbf{1}^T$ ($\mathbf{1}$은 모든 원소의 값이 1인 $n$차원 열벡터)를 계산하여 사용하는 간단한 전처리를 수행하여 이 가정을 만족시킬 수 있다. 이러한 과정을 센터링(centering)이라고 한다. 센터링된 데이터가 주어졌을 때 ($\boldsymbol{\mu}_x$=0 일 때), 특징데이터 집합 $\mathrm{Y}$에 대한 공분산은 다음과 같이 얻을 수 있다.

$$\boldsymbol{\Sigma}_y = \frac{1}{N}\mathrm{Y}\mathrm{Y}^T = \frac{1}{N}(\mathrm{W}^T\mathrm{X})(\mathrm{W}^T\mathrm{X})^T = \frac{1}{N}\mathrm{W}^T(\mathrm{X}\mathrm{X}^T)\mathrm{W} = \mathrm{W}^T\boldsymbol{\Sigma}_x\mathrm{W} \qquad \text{[식 8-6]}$$

이로부터, 특징데이터 집합의 평균과 공분산은 입력데이터 집합의 공분산을 $\mathrm{W}$에 의해 변환함으로써 얻어짐을 알 수 있다. 바꾸어 생각하면, 새롭게 얻어지는 특징데이터 집합이 인식기가 다루기 쉬운 특정 형태의 공분산 행렬을 가지도록 선형변환 $\mathrm{W}$를 정의하여 입력데이터를 변환시키는 것이 가능함을 의미한다.
그렇다면, 인식기가 다루기 쉬운 공분산이란 어떤 것이 있을까? 앞서 4장에서 우리는 여러 가지 공분산 형태에 따라 베이즈 분류기의 결정경계와 결정규칙에 어떻게 달라지는지 살펴보았다. 인식 대상이 되는 데이터 집합이 단위행렬을 공분산으로 가질 때 결정규칙은 최단거리분류기와 동일해져서 단순히 평균과의 유클리디안 거리를 이용하여 분류를 수행함으로써 최소분류오차를 얻을 수 있었다. 또한 만약 공분산 행렬이 대각행렬이라면, 정규화된 유클리디안 거리함수를 사용하여 분류기를 설계할 수 있어서, 전체 공분산 행렬을 이용하여 정의되는 마할라노비스 거리 함수를 사용하는 일반적인 경우에 비해 계산이 간단해 졌다. 따라서 만약 주어진 입력데이터를 변환하여 대각행렬이나 단위행렬을 공분산으로 가지는 특징벡터를 얻을 수 있다면, 이를 입력데이터 대신 사용함으로써 간단한 분류기를 이용하여

보다 좋은 분류성능을 얻는 것을 기대할 수 있게 된다.
먼저 대각행렬 형태의 공분산 $\Sigma_y$를 얻기 위한 선형변환을 생각해보자. 입력데이터에 대한 공분산 행렬 $\Sigma_x$의 고유벡터들을 열벡터로 가지는 행렬을 $\Phi$라고 하고 고유치를 대각원소로 가지는 행렬을 $\Lambda$라고 할 때, 고유치와 고유벡터의 정의에 의해 다음과 같은 식이 성립한다.

$$\Sigma_x \Phi = \Phi \Lambda \qquad \text{[식 8-7]}$$

이때 고유치 벡터들이 서로 직교하면서 그 크기가 1이라고 하면 $\Phi$는 $\Phi^T \Phi = \mathrm{I}$를 만족하는 직교행렬이 되므로, [식 8-7]은 다음과 같이 다시 쓸 수 있다.

$$\Sigma_x = \Phi \Lambda \Phi^T \qquad \text{[식 8-8]}$$

[식 8-6]과 [식 8-8]로부터, $\mathrm{W} = \Phi$일 때 다음 식에서 보이는 바와 같이 특징벡터의 공분산은 고유치를 대각원소로 가지는 대각 행렬이 된다.

$$\Sigma_y = \mathrm{W}^T \Sigma_x \mathrm{W} = \mathrm{W}^T (\Phi \Lambda \Phi^T) \mathrm{W} = \Lambda \qquad \text{[식 8-9]}$$

정리하면, 입력 데이터 $\mathrm{X}$에 대한 공분산행렬의 고유치분석을 통하여 서로 직교이면서 크기가 1인 고유벡터들을 얻었을 때, 이들을 열벡터로 가지는 행렬 $\Phi$를 이용하여 선형변환함으로써 얻어지는 특징벡터 $\mathrm{Y} = \Phi^T \mathrm{X}$는 대각행렬을 공분산으로 가지는 분포를 이루게 된다. 이 변환은 <대각화 (diagonalization)> 변환이라고 한다.
나아가 $\mathrm{Y}$를 추가적으로 선형변환하여 공분산행렬이 정방행렬이 될 수 있도록 만드는 것이 가능하다. $\mathrm{Y}$의 공분산 행렬이 대각행렬인 $\Lambda$로 주어져 있으므로, 각 대각원소의 제곱근의 역수값을 취하여 얻어지는 행렬 $\Lambda^{-1/2}$을 이용한 선형변환을 통해 새로운 특징데이터를 다음과 같이 얻는다.

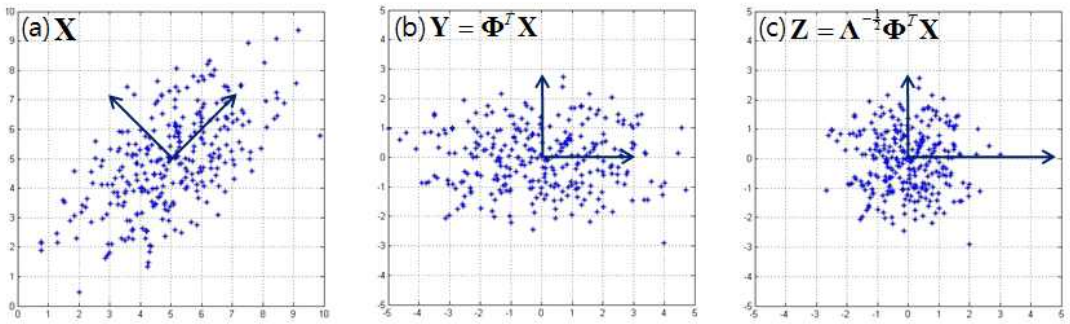
$$\mathrm{Z} = \Lambda^{-1/2} \mathrm{Y} = \Lambda^{-1/2} \Phi^T \mathrm{X} \qquad \text{[식 8-10]}$$

이렇게 얻어지는 특징데이터 $\mathrm{Z}$의 공분산을 [식 8-9]와 같은 방식으로 계산하면 쉽게 다음과 같은 사실을 얻을 수 있다.

$$\Sigma_z = (\Lambda^{-1/2})^T \Sigma_y \Lambda^{-1/2} = \mathrm{I} \qquad \text{[식 8-9]}$$

따라서 입력데이터 $\mathrm{X}$에 대한 고유치행렬 $\Lambda$과 고유벡터행렬 $\Phi$을 이용하여 정의되는 선형변환행렬 $\mathrm{W} = \Phi \Lambda^{-1/2}$은 단위행렬을 공분산으로 가지는 특징데이터 집합을 찾는 변환이 된다. 이렇게 얻어지는 특징데이터는 단위행렬을 공분산으로 가지므로, 각 요소들 간의 상관관계나 분산의 차이 등을 고려하지 않고 단순히 유클리디안 거리에 의한 최단거리분류기를 이용함으로써 최소분류율을 얻는 베이즈 분류를 수행할 수 있다. 이밖에도 이렇게 얻어지는 특징벡터는 각 요소들끼리 상관관계가 없으며, 특히 가우시안 분포를 따를 경우 독립이 되

므로 이후에 수행하는 데이터 분석이 용이해 지는 장점을 가지게 된다. 따라서 패턴 인식 뿐 아니라 신호처리 분야에서 이 변환을 이용한 데이터에 대한 전처리가 널리 사용된다. 이렇게 변환된 데이터는 기존의 복잡한 분포특성이 제거되었다는 의미에서 이 변환을 <화이트닝(whitening)> 변환이라고 부른다.




[그림 8-4] 대각화 변환 및 화이트닝 변환

[그림 8-4]에 2차원 데이터를 이용하여 대각화 변환과 화이트닝 변환을 수행한 예를 나타내었다. [그림 8-4a]에 주어진 입력데이터는 다음과 같은 일반적인 형태의 공분산을 가지는 가우시안 분포로부터 생성되었다.

$$\mathbf{\Sigma}_{\mathbf{x}} = \begin{bmatrix} 2.5 & 1.5 \\ 1.5 & 2.5 \end{bmatrix} \qquad \text{[식 8-10]}$$

이 행렬의 고유치행렬과 고유벡터행렬을 찾으면 각각 다음과 같이 얻을 수 있다.

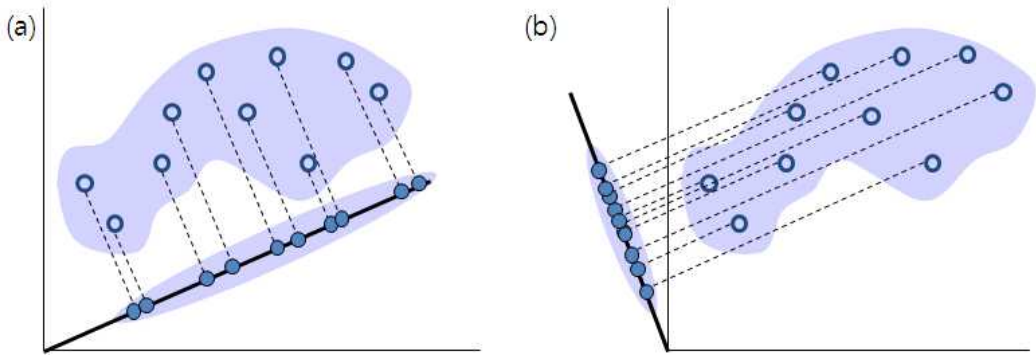
$$\mathbf{\Phi} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}, \quad \mathbf{\Lambda} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \qquad \text{[식 8-11]}$$

얻어진 고유벡터를 [그림 8-4a]에 벡터로 나타내었다. 그림에서 알 수 있듯이 첫 번째 고유벡터는 입력데이터의 두 요소간의 상관관계를 나타내는 방향, 즉 데이터의 분산이 가장 커지는 방향을 가리키고 있으며, 두 번째 고유벡터는 이에 수직인 벡터가 된다. 또한 각 고유벡터 방향으로의 데이터의 분산은 해당 고유치의 값으로 나타난다.
[그림 8-4b]에는 대각화 변환을 통하여 얻어진 특징데이터 $\mathbf{Y}$를 나타내었다. 그림에서 두 요소간의 상관관계가 사라진 것을 확인할 수 있다. 또한 특징데이터 $\mathbf{Y}$의 고유벡터행렬은 $\mathbf{\Lambda}$가 되므로, 결국 입력데이터 $\mathbf{X}$의 두 고유벡터가 각각 수평축과 수직축 방향이 되도록 회전변환이 일어났음을 알 수 있다. [그림 8-4c]에는 화이트닝 변환을 통하여 얻어진 특징데이터 $\mathbf{Z}$를 나타내었다. 특징데이터 $\mathbf{Y}$에서 나타나는 각 요소들 간의 분산차도 사라며, 수평축과 수직축 모두가 분산 1을 가지는 형태로 데이터가 변환되었음을 알 수 있다.
지금까지 살펴본 선형변환은, 입력데이터의 차원에 대한 변화는 고려하지 않으며, 같은 차원의 공간에서 단순히 데이터를 표현하기 위한 기저벡터만을 변형하는 것으로 볼 수 있다. 즉, 특징데이터 $\mathbf{Y}$는 입력데이터 $\mathbf{X}$의 고유벡터를 기저벡터로 사용하여 표현한 것이며, 특징데

이터 $\mathbf{Z}$는 고유벡터에 고유치의 제곱근 값을 곱하여 얻어지는 벡터들을 기저벡터로 사용하여 표현한 값이 된다. 그런데 패턴인식을 위한 특징추출에서는 단순한 데이터 변환 뿐 아니라 차원을 축소하는 목적도 함께 가지므로, 기저벡터의 수를 줄여서 특징벡터의 차원을 줄여주어야 한다. 주성분 분석은 지금까지 살펴본 공분산에 기반한 선형변환을 수행하면서 동시에 기저벡터의 선택을 통하여 축소된 차원을 얻는 과정도 함께 고려한다.

## 8.2.2 주성분분석 알고리즘

<주성분분석 (Principal Component Analysis)>은 대표적인 선형변환에 의한 특징추출 방법으로, 그 기본 형태는 [식 8-2]와 동일하며, 얻어진 특징데이터의 차원 $m$이 입력데이터의 차원 $n$보다 작은 값이 되어 저차원의 특징을 추출하는 것을 기본 전제로 한다. 주성분분석법의 가장 큰 목적은 변환전의 데이터 $\mathbf{X}$가 가지고 있는 정보를 차원 축소 후에도 최대한 유지하도록 하는 것이다. 그렇다면 이러한 목적을 달성할 수 있는 변환행렬 $\mathbf{W}$는 어떻게 찾을 수 있을 것인가? 이에 대한 수학적 유도도 가능하나, 먼저 앞 절에서 살펴본 공분산분석에 기반한 선형변환과의 관계를 중심으로 직관적으로 접근해 보겠다.
예를 들어 [그림 8-2]에서 주어진 데이터를 1차원으로 축소하는 경우를 생각해 보자. 2차원 데이터가 1차원으로 축소되므로 분명히 정보의 손실이 발생하게 된다. 우리의 목적은 이 손실되는 정보를 최소한으로 줄이는 행렬 $\mathbf{W}$를 찾는 것이다. 예제의 경우에 행렬 $\mathbf{W}$는 2차원 열벡터로 표현되며, 이것은 그림에서 원래데이터를 사영시키는 방향을 가리키게 된다. 따라서 결국 주성분분석법은 데이터 손실량을 최소로 하는 사영벡터를 찾는 것이 된다.




[그림 8-5] 변환행렬의 변화에 따른 추출된 특징의 차이

[그림 8-5]에 두 가지 서로 다른 벡터로의 사영 결과를 나타내었다. [그림 8-5b]의 경우를 보면, 2차원 공간상에서 대각선으로 넓게 퍼져있던 데이터들이 1차원으로의 사영 결과 좁은 영역 안에 밀집하게 되어 결과적으로 데이터들 간의 구분이 어려워진다. 즉, 정보의 손실 크다고 할 수 있다. 이에 반해 [그림 8-5a]의 경우를 보면 원래의 2차원 데이터가 가지는 정보들을 거의 유지한 형태의 1차원 데이터를 사영을 통해 얻어냈음을 확인할 수 있다. 이로부터, 선형변환을 통한 특징추출에서 원래 정보의 손실을 최소화하기 위해서는 데이터 집합이 가능한 넓게 퍼질 수 있는 방향으로 사영을 수행하는 것이 필요함을 알 수 있다. 다시 말하면, 데이터의 분산이 가장 큰 방향으로의 선형 변환을 수행함으로써 주성분분석의 목적을

달성할 수 있다.
데이터 집합의 분산이 가장 큰 방향을 찾는다는 것을 결국 8.2.1절에서 소개한 공분산에 기반한 선형변환으로 귀결된다. [그림 8-4]에서 살펴본 바와 같이 공분산 행렬의 최대 고유치와 고유치벡터는 데이터 집합의 분산이 가장 커지는 방향과, 그때의 분산을 나타낸다. 따라서 정보손실이 가장 적어지는 방향을 찾기 위해서는 데이터의 공분산 행렬의 고유치와 고유벡터를 찾아 그 고유치가 큰 것부터 순서대로 찾아서 행렬 $\mathrm{W}$를 구성하면 된다. 이러한 주성분분석 알고리즘을 단계별로 정리하면 다음과 같다. 여기서는 8.2.1의 논의에서 사용한 입력데이터 $\mathrm{X}$의 평균이 0이라는 가정을 사용하지 않고, 일반적인 경우에 대한 처리과정을 기술하였다.

**[주성분분석(PCA) 알고리즘의 수행 단계]**

① 입력데이터 $\mathrm{X}$의 평균 $\boldsymbol{\mu}_x$와 공분산 $\boldsymbol{\Sigma}_x$를 계산한다.

$$\boldsymbol{\mu}_x = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i$$

$$\boldsymbol{\Sigma}_x = \frac{1}{N}(\mathrm{X}-\boldsymbol{M}_x)(\mathrm{X}-\boldsymbol{M}_x)^T \quad , \quad \boldsymbol{M}_x = \boldsymbol{\mu}_x \mathbf{1}^T$$

($\mathbf{1}$은 모든 원소의 값이 1인 $n$차원 열벡터)

② 고유치 분석을 통해 공분산 $\boldsymbol{\Sigma}_x$의 고유치행렬$\boldsymbol{\Lambda}$과 고유벡터행렬$\mathrm{U}$을 계산한다.

$$\boldsymbol{\Sigma}_x = \mathrm{U}\boldsymbol{\Lambda}\mathrm{U}^{\mathrm{T}} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n] \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & \lambda_n \end{bmatrix} [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n]^T$$

③ 고유치 값이 큰 것부터 순서대로 $m$개의 고유치 $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ 를 선택한다.

④ 선택한 고유치에 대응되는 고유벡터를 열벡터로 가지는 변환행렬 $\mathrm{W}$를 생성한다.

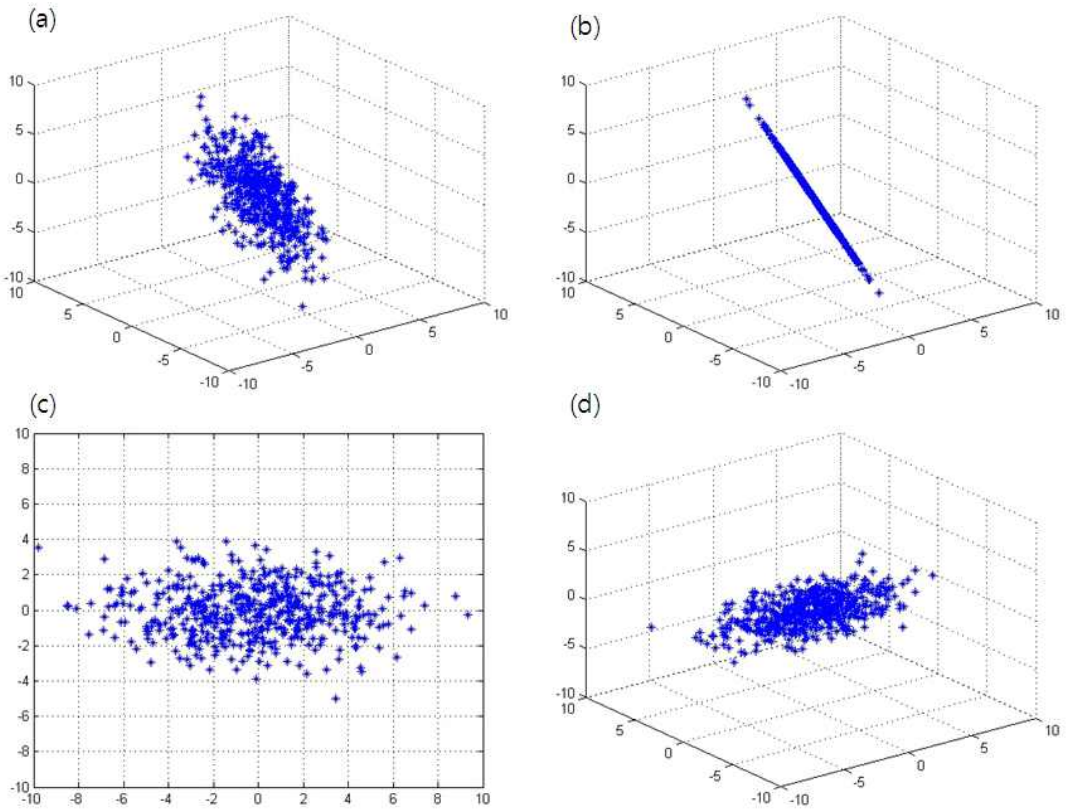
$$\mathrm{W} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m]$$

⑤ $\mathrm{W}$에 의한 선형변환에 의해 특징데이터 $\mathrm{Y}$를 얻는다.

$$\mathrm{Y} = \mathrm{W}^T\mathrm{X}$$

[그림 8-6]에 3차원 데이터에 대한 주성분 분석의 예를 나타내었다. [그림 8-6(a)]에 나타난 3차원 입력데이터들에 대하여 주성분분석을 수행하여 첫 번째 주성분 벡터를 찾아 그 방향을 [그림 8-6(b)]에 나타내었다. 이와 함께 3차원 데이터를 해당 주성분 벡터로 사영하여 얻어지는 점들도 함께 그림에 표시하였다. 각 데이터들의 주성분 벡터로 사영하여 얻어지는 실수값이 주성분 분성에 의해 추출되는 1차원 특징값이 된다. [그림 8-6(c)]에는 첫 번째와 두 번째의 주성분 벡터를 이용하여 선형변환하여 얻어지는 특징값을 나타내었다. 두 개의

벡터를 사용하였으므로 얻어지는 특징은 2차원 벡터로 나타난다. 마지막으로 [그림 8-6(d)]에는 3개의 주성분 벡터를 모두 사용하여 얻어지는 선형변환 행렬 $\mathbf{W}$를 이용하여 변환을 수행한 것이다. 모든 고유벡터를 사용하였으므로, 차원축소의 효과는 발생하지 않으며, 단지 앞의 8.2.1에서 설명한 바와 같이 각 요소간의 상관관계가 사라진 형태의 분포를 가진 3차원 특징데이터를 얻게 된다.




[그림 8-6] 3차원 데이터에 대한 주성분 분석. (a) 입력데이터 (b) 1차 주성분 벡터와 그 벡터로 사영된 1차원 데이터 집합 (c) 1, 2차 주성분 벡터가 이루는 평면과 추출된 2차원 특징값 (d) 1,2,3차 주성분 벡터를 모두 사용한 데이터의 선형변환 결과

## 8.2.3 주성분분석법의 수학적 유도

앞 절에서 직관에 의존하여 살펴본 바와 같이, 주성분 분석법은 차원축소에 의해 발생하는 데이터의 오차 (혹은 정보의 오차)를 최소화 하는 기저벡터를 찾는 것을 목적으로 하였으며, 이는 공분산 행렬의 고유치분석을 통해 얻어질 수 있었다. 이 절에서는 직관적으로 이해한 사실에 대하여 수학적으로 유도해 보겠다.

이를 위해 먼저 우리가 찾아야 하는 값을 파라미터로 정의해야 할 것이다. 선형특징추출에 있어서 파라미터란, 특정 목적을 만족하도록 하는 기저벡터를 의미하므로, $n$차원 입력공간을 나타내는 임의의 기저벡터집합을 $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n\}$ 으로 두어 파라미터를 나타낸다. 이 기저

벡터를 이용하여 임의의 데이터 $\boldsymbol{x}$를 나타내면 다음과 같이 쓸 수 있다.

$$\boldsymbol{x}=\sum_{j=1}^{n}(\boldsymbol{x}^T\boldsymbol{u}_j)\boldsymbol{u}_j \qquad \text{[식 8-12]}$$

이 때 기저벡터는 서로 직교이면서 크기가 1인 직교단위기저라고 가정한다. 만약 $n$차원 입력공간의 데이터를 나타내기 위해 $n$개의 기저벡터를 모두 사용한다면 데이터의 정보의 손실은 없을 것이다. 그러나 우리는 저차원의 특징을 추출하는 것을 목적으로 하므로, $n$개의 기저벡터들 중 $m$ $(m<n)$개만 선택하여 사용한다. 이 때 선택된 $m$개의 기저벡터만을 이용하여 입력벡터 $\boldsymbol{x}$를 근사하여 나타내면 다음 식과 같다.

$$\tilde{\boldsymbol{x}}=\sum_{j=1}^{m}(\boldsymbol{x}^T\boldsymbol{u}_j)\boldsymbol{u}_j \qquad \text{[식 8-13]}$$

원래 입력벡터 $\boldsymbol{x}$와 적은 수의 기저벡터를 이용하여 근사하여 나타낸 벡터 $\tilde{\boldsymbol{x}}$의 차이가 손실되는 정보가 되므로, 손실되는 정보량은 다음과 같이 두 벡터 차의 크기를 이용하여 정의할 수 있다.

$$J=\frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{x}_i-\tilde{\boldsymbol{x}_i}\|^2=\frac{1}{N}\sum_{i=1}^{N}\sum_{j=m+1}^{n}(\boldsymbol{x}_i^T\boldsymbol{u}_j)^2=\sum_{j=m+1}^{n}\boldsymbol{u}_j^T\mathrm{S}\boldsymbol{u}_j \qquad \text{[식 8-14]}$$

여기서 S는 다음 [식 8-15]로 정의되어 데이터로부터 계산되는 통계량이다. 여기서도 앞 절에서와 마찬가지로 데이터들의 평균이 0이 되도록 센터링 되었다고 가정하므로, S는 공분산 행렬이 된다.

$$\mathrm{S}=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}\boldsymbol{x}^T \qquad \text{[식 8-15]}$$

결국 제곱오차에 의해 정의되는 정보손실량 $J$를 최소화 하는 것은 각 기저벡터로 사영하여 얻어지는 특징값들의 분산이 최소가 되는 기저벡터들을 찾아 제거하는 것을 의미한다. 달리 말하면, 데이터들을 각 기저벡터로 사영하여 얻어지는 특징값들의 분산을 최대로 하는 것부터 순서대로 $m$개를 선택하여 사용하고, 나머지 $n-m$개를 버리는 것을 의미한다.
지금까지 우리는 차원축소에 의한 정보손실을 최소화 하는 기저벡터를 선택하는 것은 분산을 최대로 하는 기저벡터를 찾는 것과 동일함을 수학적 분석을 통해 알아보았다. 그렇다면, 주성분분석법에서 찾는 공분산행렬의 고유벡터가 분산을 최대로 하는 기저벡터가 되는지 살펴보아야 하겠다. 이를 위해 먼저 목적함수는 [8-14]를 최소화 하는 기저벡터를 찾아보자. 그런데, 이와 함께 기저벡터가 단위벡터가 되어야 한다는 가정을 함께 만족해야 하므로, 라그랑제 승수 (Lagrange Multiplier)를 이용하여 이 조건을 결합한 변형된 목적함수를 다음 식과 같이 정의할 수 있다.

$$\widetilde{J}(\boldsymbol{u}) = \boldsymbol{u}^T \mathrm{S}\boldsymbol{u} - \lambda(1-\boldsymbol{u}^T\boldsymbol{u}) \qquad \text{[식 8-16]}$$

이 목적함수를 $\boldsymbol{u}$에 대해 미분하여 극소값(혹은 극대값)을 가지는 지점을 찾으면 다음과 같은 관계식을 얻는다.

$$\mathrm{S}\boldsymbol{u} = \lambda\boldsymbol{u} \qquad \text{[식 8-17]}$$

이 식으로부터 목적함수를 최소화 하는 기저벡터 $\boldsymbol{u}$와 라그랑제 승수 $\lambda$는 각각 행렬 S의 고유치와 고유벡터가 됨을 알 수 있다. 또한 [식 8-17]의 양변에 $\boldsymbol{u}^T$를 곱하면 다음 식을 얻을 수 있다.

$$\boldsymbol{u}^T \mathrm{S}\boldsymbol{u} = \lambda \qquad \text{[식 8-18]}$$

결국 공분산행렬의 고유치 값은 해당 고유벡터를 이용하여 데이터를 사영함으로써 얻어지는 특징값들의 분산값을 나타낸다. 또한, 이를 이용하여 목적함수 식을 다시 쓰면 [식 8-19]와 같이 되어, 결국 목적함수의 최소화는 공분산행렬의 고유치값이 가장 큰 것부터 순서대로 $m$개의 기저벡터를 선택하고 나머지를 버림으로써 가능함을 알 수 있다.

$$J = \sum_{j=m+1}^{n} \boldsymbol{u}_j^T \mathrm{S}\boldsymbol{u}_j = \sum_{j=m+1}^{n} \lambda_j \qquad \text{[식 8-19]}$$

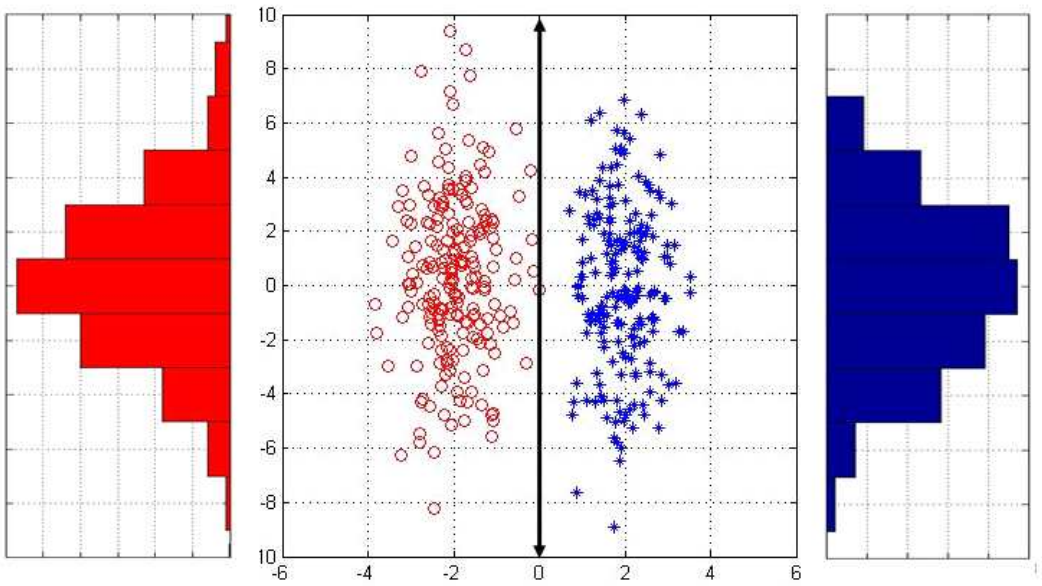
공분산행렬의 고유치 분석을 통해 쉽게 얻을 수 있는 정보 손실량에 대한 [식 8-19]는 주성분분석에 의해 몇 차원의 특징을 추출할 것인지를 선택하는 기준을 제시해 줄 수 있다. 즉, [식 8-19]를 전체 고유치 벡터들의 합으로 나누어 주면, 손실되는 정보량의 비중을 알 수 있다. 이를 바꾸어 말하면, $m$개의 특징을 사용하여 데이터를 표현할 때, 표현 가능한 정보의 비율은 다음 식과 같이 쓸 수 있다.

$$r(n,m) = \frac{\sum_{i=1}^{m}\lambda_i}{\sum_{i=1}^{n}\lambda_i} \qquad \text{[식 8-20]}$$

이 값을 이용하여, 사전에 사용자에 정의된 역치값 $\theta$(예를 들어 $\theta$=0.98)을 이용하여 $\sum_{i=1}^{m}\lambda_i / \sum_{i=1}^{n}\lambda_i > \theta$ 가 되도록 $m$의 값을 결정해 줄 수 있다. 이때 $\theta$=0.98이라는 것은 정보의 손실량이 전체의 2%이하임을 의미한다. 이와 같이 손실을 허용할 정보의 비율을 미리 정하는 방법 이외에도 고유치값의 변화를 이용하여 $m$의 값을 결정하는 방법들이 존재하는데, 이에 대해서는 9장에서 실제 데이터를 이용하여 설명하겠다.

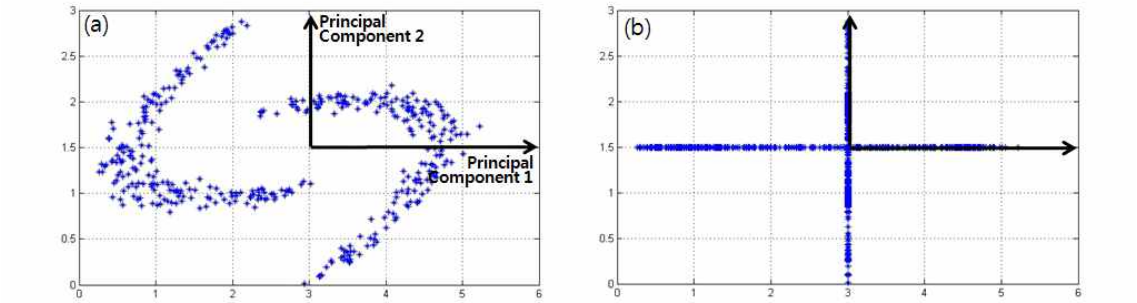
## 8.2.4 주성분분석법의 특성과 문제점

이상에서 살펴본 바와 같이 주성분분석법은 입력으로 주어진 고차원 데이터의 정보를 최대한 손실하지 않는 방향으로의 차원 축소를 수행한다. 데이터 분석에 대한 특별한 목적이 없는 경우에는 이 방법을 취하는 것이 가장 합리적인 차원 축소의 기준이 될 것이다. 그러나 패턴 인식의 문제에 있어서는 주성분분석법이 적절하지 못한 경우가 있다.
주성분분석법은 기본적으로 입력데이터 전체의 2차통계량 (공분산)을 그 분석의 기준으로 둔다. 따라서 각 데이터의 클래스 라벨 정보를 활용하지 않는 비교사 학습에 해당한다. 이와 같이 클래스 정보를 활용하지 않음으로 인하여 결과적으로 분류에 핵심이 되는 정보를 손실하는 결과를 초래할 수 있다.




[그림 8-7] 주성분분석이 적합하지 않은 경우

[그림 8-7]에 주어진 데이터에 대하여 주성분 분석을 수행하는 경우를 생각해 보자. 이 경우에 클래스의 구분 없이 전체 데이터에 대한 주성분을 찾으면 수직축에 가까운 벡터를 찾을 수 있다. 이 방향으로 사영한 특징에 대하여 각 클래스별로 데이터의 분포를 각각 그림의 양쪽에 히스토그램으로 나타내었다. 이렇게 얻어진 특징정보만을 사용하여 분류를 수행한다면, 히스토그램으로부터 알 수 있듯이 두 클래스가 거의 겹치는 분포를 가지게 되어 결과적으로 클래스 정보를 모두 손실한 상태로 분류를 수행하게 된다. 이러한 문제를 해결하기 위하여 다음절에서 소개하는 선형판별분석법에서는 클래스 정보를 활용하여 분류에 핵심적인 정보를 추출하는 방향을 찾는다.



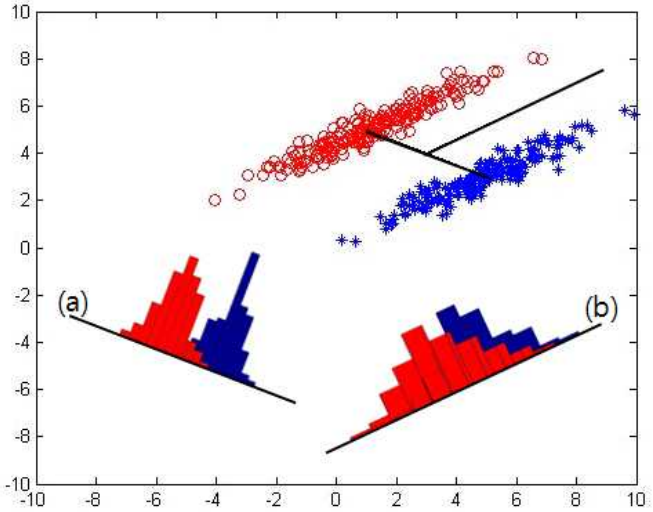

[그림 8-8] 비선형 구조를 가진 데이터(a)와 주성분분석에 의해 얻어진 1차원 특징(b)

이 밖에도 주성분분석은 기본적으로 선형변환을 가정하므로, [그림 8-8]에서처럼 데이터 자체가 비선형 구조를 가지고 분포되어 있는 경우에는 이를 반영하는 저차원 특징을 찾는 것은 불가능하다. [그림 8-8b]에서 보이는 바와 같이 어떤 방향으로의 사영도 데이터가 가지는 2차원 구조를 제대로 표현하지 못한다. 이러한 문제를 해결하기 위하여 커널함수를 사용하는 방법이나 비선형 매니폴드학습법 등이 개발되었다. 이에 대해서는 14장에서 소개하겠다.

# 8.3 선형판별분석법

## 8.3.1 선형판별분석 알고리즘

선형판별분석도 주성분분석법과 마찬가지로 [식 8-2]와 같은 선형변환에 의해 특징을 추출한다. 그러나 주성분분석법과는 다른 점은, 변환행렬 $\mathbf{W}$를 최적화하는 목적함수를 정의함에 있어서 클래스 정보를 적극적으로 활용한다는 것이다.




[그림 8-9] 클래스 라벨을 고려한 선형 변환
(a) 클래스 간 차이를 유지하는 선형변환
(b) 클래스 간의 오버랩을 발생시키는 선형변환

[그림 8-9]에 주어진 데이터에 대하여 분류에 적합한 1차원 특징을 찾는다면 어떤 방향으로 사영을 해야 할 지 생각해 보자. [그림 8-9]의 (b)와 같은 방향으로 사영을 취한 경우, 두 클래스간의 오버랩이 발생하여 결과적으로 분류성능을 저하시키는 반면, (a)그림과 같은 방향을 선택한 경우 1차원 정보만으로도 어느 정도 좋은 분류 성능을 내는 특징을 얻을 수 있다. 선형판별분석법은 이와 같이 분류에 적합한 선형변환을 찾는 방법이다.
그렇다면 [그림 8-9]의 (a)와 같은 방향은 어떻게 정의할 수 있는지 알아보자. 선형변환에 의해 찾아진 특징이 분류에 효과적이 되기 위해서는 각 클래스들이 가능한 서로 멀리 떨어질 수 있도록 그 거리를 유지하는 것이 중요하다. 선형 판별 분석법에서는 클래스 간의 거리를 각 클래스의 평균을 이용하여 [식 8-21]과 같은 목적함수 $J$에 의해 측정한다.

$$J = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \qquad \text{[식 8-21]}$$

$$m_k = \frac{1}{|C_k|} \sum_{\boldsymbol{x}_i \in C_k} \boldsymbol{w}^T \boldsymbol{x}_i = \boldsymbol{w}^T \boldsymbol{m}_k \qquad \text{[식 8-22]}$$

$$s_k = \sum_{\boldsymbol{x}_i \in C_k} (\boldsymbol{w}^T \boldsymbol{x}_i - m_k) = \sum_{\boldsymbol{x}_i \in C_k} \boldsymbol{w}^T (\boldsymbol{x}_i - \boldsymbol{m}_k)(\boldsymbol{x}_i - \boldsymbol{m}_k)^T \boldsymbol{w} \qquad \text{[식 8-23]}$$

[식 8-21]의 분자에 나타난 $m_k$는 [식 8-22]에 정의된 것과 같이 클래스 $C_k$에 속하는 데이터들을 벡터 $\boldsymbol{w}$로 사영하여 얻어지는 1차원 특징값들의 평균을 나타낸다. 이 값은 또한 클래스 $C_k$에 속하는 입력데이터들의 평균벡터 $\boldsymbol{m_k}$를 $\boldsymbol{w}$ 방향으로 사영한 값과도 동일하다. 목적함수에서 분자의 값은 결국 각 클래스에 속하는 특징값들이 평균 간의 거리가 되므로, 분자의 값이 커질수록 두 클래스의 평균이 멀리 떨어져서 분류가 쉬워지게 된다. 한편, 분모에

나타난 $s_i$는 [식 8-23]에서 정의된 바와 같이 각 변환 후에 생기는 특징 데이터 집합에서 각 클래스별 분산에 비례하는 값이다. 따라서 분모의 값이 작아진다는 것은 각 클래스 내의 분산이 작아져서 결과적으로 같은 클래스 내에서는 서로 결집되어진 상황을 나타낸다. 이를 종합하면 목적함수 $J$를 최대로 하는 벡터 $\boldsymbol{w}$는, 클래스 간의 거리는 멀어지게 하고, 같은 클래스 내에서는 결집되게 하여 분류에 적합한 특징으로의 변환을 유도한다.
목적함수 $J$를 최대로 하는 벡터 $\boldsymbol{w}$를 찾기 위하여 $J$를 변환 전의 원래 데이터 $\boldsymbol{x}$를 이용하여 나타내면 다음과 같다.

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T(\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T\boldsymbol{w}}{\boldsymbol{w}^T\sum_{k=1}^{2}\sum_{\boldsymbol{x}_i \in C_k}(\boldsymbol{x}_i - \boldsymbol{m}_k)(\boldsymbol{x}_i - \boldsymbol{m}_k)^T\boldsymbol{w}} = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}} \qquad \text{[식 8-24]}$$

이 식에서 $S_B$는 두 클래스 간의 흩어진 정도를 나타내므로 <클래스간 산점행렬(Between Scatter Matrix)>라고 하며, $S_W$는 각 클래스 내에서 데이터가 흩어진 정도들을 계산하여 모두 합한 것으로 <클래스내 산점행렬(Within Scatter Matrix)> 라고 한다. 이 목적함수를 최대로 하는 $\boldsymbol{w}$는 분자를 최대로 하고 분모를 최소로 하는 방향이므로, 다음과 같이 얻어질 수 있다.

$$\boldsymbol{w} \propto S_W^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2) \qquad \text{[식 8-25]}$$

만약 [식 8-25]에서 분모에 대한 값 $S_W$를 고려하지 않는다면 벡터 $\boldsymbol{w}$는 두 평균의 차벡터에 비례하는 형태를 가지게 되는데, [그림 8-9]의 (a)에서 보이는 방향과 일치한다.
지금까지는 클래스가 두 개인 이진분류의 경우로, 이를 다중클래스 분류로 확장하고, 벡터 $\boldsymbol{w}$도 $\mathbb{W}$로 확장하면 다음과 같이 목적함수를 정의할 수 있다.

$$J(\mathbb{W}) = Tr\left\{(\mathbb{W}S_W\mathbb{W}^T)^{-1}(\mathbb{W}S_B\mathbb{W}^T)\right\} \qquad \text{[식 8-26]}$$

$$S_W = \sum_{k=1}^{M} S_k = \sum_{k=1}^{M}\sum_{\boldsymbol{x}_i \in C_k}(\boldsymbol{x}_i - \boldsymbol{m}_k)(\boldsymbol{x}_i - \boldsymbol{m}_k)^T \qquad \text{[식 8-27]}$$

$$S_B = \sum_{k=1}^{M} N_k(\boldsymbol{m}_k - \boldsymbol{m})(\boldsymbol{m}_k - \boldsymbol{m})^T$$

이 목적함수를 최대로 하는 변환행렬 $\mathbb{W}$는 $S_W^{-1}S_B$의 고유벡터들을 열벡터로 가지는 행렬이 된다. 아래에 선형판별분석법을 단계별로 정리하였다.

**[선형판별분석(LDA) 알고리즘의 수행 단계]**

① 입력데이터 $\mathbf{X}$를 각 클래스 라벨에 따라 $M$개의 클래스로 나누어 각각 평균 $\boldsymbol{m}_k$와 클래스간 산점행렬 $S_B$, 그리고 클래스내 산점행렬 $S_W$를 계산한다.

$$\boldsymbol{m}_k = \frac{1}{|C_k|}\sum_{\boldsymbol{x}_i \in C_k} \boldsymbol{x}_i \qquad (k = 1,2,\ldots,M)$$

$$S_W = \sum_{k=1}^{M} S_k = \sum_{k=1}^{M}\sum_{\boldsymbol{x}_i \in C_k} (\boldsymbol{x}_i - \boldsymbol{m}_k)(\boldsymbol{x}_i - \boldsymbol{m}_k)^T$$

$$S_B = \sum_{k=1}^{M} N_k (\boldsymbol{m}_k - \boldsymbol{m})(\boldsymbol{m}_k - \boldsymbol{m})^T$$

② 고유치 분석을 통해 행렬 $S_W^{-1}S_B$의 고유치행렬$\boldsymbol{\Lambda}$과 고유벡터행렬$\mathbf{U}$을 계산한다.

$$\boldsymbol{S}_W^{-1}\boldsymbol{S}_B = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\boldsymbol{T}} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n] \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & \lambda_n \end{bmatrix} [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n]^T$$

③ 고유치 값이 큰 것부터 순서대로 $m$개의 고유치 $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ 를 선택한다.

④ 선택한 고유치에 대응되는 고유벡터를 열벡터로 가지는 변환행렬 $\mathbf{W}$를 생성

$$\mathbf{W} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m]$$

⑤ $\mathbf{W}$에 의한 선형변환에 의해 특징데이터 $\mathbf{Y}$를 얻는다.

$$\mathbf{Y} = \mathbf{W}^{\boldsymbol{T}}\mathbf{X}$$

## 8.3.2 선형판별분석법의 특성과 문제점

선형판별분석법과 주성분분석법은 두 방법 모두 고유치 분석을 통하여 변환행렬을 찾고 이것을 이용한 선형변환에 의해 특징 추출을 수행한다. 따라서 기본적인 처리 단계가 유사하여 주성분분석법에서 고려해야할 점이나 문제점들이 선형판별분석법에서도 동일하게 존재한다. 첫째로, 선형판별분석법 역시 선형변환에 의한 특징추출을 수행하므로 데이터 집합 전체, 혹은 각 클래스 집합들이 복잡한 비선형 구조를 가진 경우에는 적절한 변환을 수행해 주지 못한다. 이에 대한 해결책으로는 주성분분석법과 마찬가지로 커널방법을 이용하거나 비선형 매니폴드학습법을 이용하는 등의 접근이 필요하다. 이에 대해서는 14장에서 소개하겠다.

둘째로 주성분분석에서와 마찬가지로 선택하는 고유벡터의 개수 (즉, 축소된 차원) $m$의 값을 결정해 주어야 한다. 선형판별분석법의 경우, 주성분분석법에서 제시한 [식 8-20]과 같이 명시적인 값을 찾기는 힘들다. 선형판별분석의 주된 목적이 분류율을 높이는 특징을 찾는 것이므로, 데이터 집합에 대하여 직접 분류를 수행하여 얻어지는 분류율을 기준으로 선택하

는 것이 일반적이다. 이에 대해서는 9장에서 실제 데이터에 적용된 예를 보이겠다. 그런데 그에 앞서 한 가지 유의할 점은 고유치 분석의 대상이 되는 행렬 $S_W^{-1}S_B$에 의해 찾아지는 고유벡터의 개수가 제한된다는 점이다. 먼저 클래스간 산점행렬 $S_B$는 클래스 개수만큼 주어지는 평균벡터 $m_k$로 정의되는 행렬로, 그 랭크가 클래스 개수에 한정된다. 즉, 클래스 개수가 $M$인 경우에 $S_B$의 랭크는 $M-1$이 된다. 따라서 행렬 $S_W^{-1}S_B$의 랭크도 최대 $M-1$이 되어 고유치 값이 0이 아닌 고유벡터는 최대로 $M-1$개만 찾을 수 있다. 결국, 선형판별 분석법에 의해 찾아지는 특징벡터는 그 차원이 최대 $M-1$로 제한된다. 특별히 이분류 문제의 경우는 결국 하나의 특징값만을 찾을 수 있고, 선형판별 분석에 의한 분류는 찾아진 특징값의 범위에 따라 클래스가 정해지게 되어 특징추출 과정이 판별함수를 계산하는 것과 같아진다. "선형판별분석법"이라는 이름은 이러한 관점에 따라 붙여진 것이다.

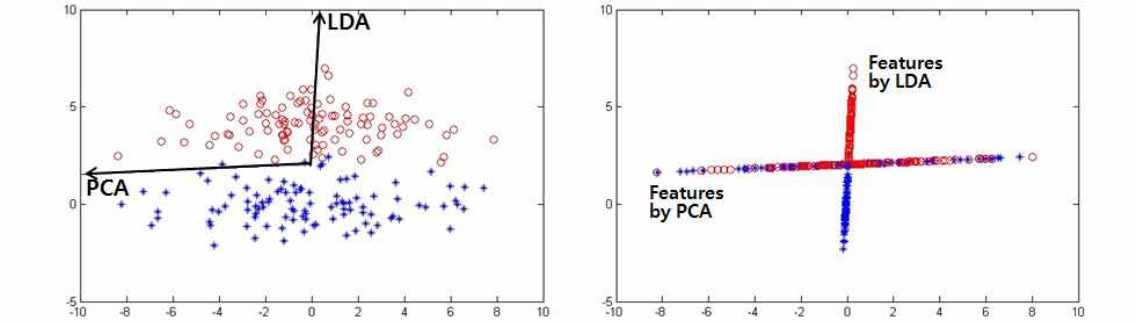
고유치 분석과 관련하여 한 가지 더 유의할 점은 클래스내 산점행렬 $S_W$에 관한 것이다. 고유치분석을 위해서는 이 행렬의 역행렬을 계산해 주어야 하는데, 만약 데이터의 수가 입력차원보다 크지 않게 되면 특이행렬이 되어 역행렬을 찾을 수 없는 문제가 발생하게 된다. 이러한 문제는 영상데이터와 같이 입력차원이 큰 데이터에서 자주 발생하는 문제로, 작은 샘플집합의 문제 (Small Sample Set problem)이라고 불린다. 이를 해결하기 위해서 실제 응용에서는 선형판별분석을 입력데이터에 대하여 직접 수행하는 대신, 먼저 주성분분석에 의해 차원 축소된 특징을 얻고, 이 특징 데이터에 대하여 선형판별분석을 수행하는 방법을 취한다. 이러한 예를 9장에서 직접 살펴볼 것이다.

이상과 같이 여러 가지 계산상의 제약점과 불안정성이 존재함에도 불구하고, 선형판별분석은 기존의 주성분분석이나 요인분석 등의 방법이 가지지 못하는 교사학습 능력을 가지고 있으므로 주성분분석을 통해 만족할 만한 결과를 얻지 못하는 경우에 그 대안으로 널리 사용되고 있다. 두 방법의 차이에 대해서는 다음 절에서 실험을 통해 알아볼 것이다.

## 8.4 매트랩을 이용한 실험

마지막으로 간단한 이차원 데이터에 대해 주성분분석과 선형판별분석을 적용하여 특징을 추출해 보자. 실험에 사용될 데이터가 [그림 8-10]에 나타나 있다. 데이터는 두 가지 클래스로 구분되어 클래스 정보도 함께 주어지지만, 주성분분석법에서는 이를 사용하지 않고 데이터 전체에 대한 주성분을 찾게 된다.

특징추출을 위한 프로그램이 [프로그램 8-1]에 나타나 있다. 생성된 데이터를 2차원 평면에 나타낸 뒤, 각각 PCA와 LDA를 적용시키고, 그 결과 얻은 1차원 특징값을 [그림 8-9]의 오른쪽에 나타내었다. 그림에서 주성분분석법에 의해 찾아진 특징은 클래스 정보를 거의 소실한 반면, 선형판별분석에 의해 찾아진 특징은 분류에 매우 효율적임을 확인 할 수 있다.




[그림 8-10] 실험데이터와 추출된 1차원 특징

**프로그램 8-1 PCA and LDA**

주성분분석법과 선형판별분석법

```
001 % 데이터의 생성 ------------------------------------
002 N=100;
003 m1=[0 0];  s1=[9 0;0 1];
004 m2=[0 4];  s2=[9 0;0 1];
005 X1=randn(N,2)*sqrtm(s1)+repmat(m1,N,1);      % 클래스1 데이터 생성
006 X2=randn(N,2)*sqrtm(s2)+repmat(m2,N,1);      % 클래스2 데이터 생성
007 figure(1);
008 plot(X2(:,1),X2(:,2),'ro'); hold on
009 plot(X1(:,1),X1(:,2),'*');
010 axis([-10 10 -5 10]);
011 save data8_9 X1 X2
012 % PCA에 의한 분석 ----------------------------------
013 X=[X1;X2];
014 M=mean(X); S=cov(X);              % 평균과 공분산 계산
015 [V,D]=eig(S);                     % 고유치분석(U:고유벡터행렬, D: 고유치행렬)
016 w1=V(:,2);                       % 첫 번째 주성분벡터
017 figure(1);                       % 첫 번째 주성분벡터를 공간에 표시
018 line([0 w1(1)*D(2,2)]+M(1),[0 w1(2)*D(2,2)]+M(2));
019 YX1=w1'*X1';  YX2=w1'*X2';      % 첫 번째 주성분벡터로 사영
020 pYX1=w1*YX1; pYX2=w1*YX2;    % 사영된 데이터를 2차원 공간으로 환원
021 figure(2);                    % 사영된 데이터를 2차원 공간으로 환원하여 표시
022 plot(pYX1(1,:), pYX1(2,:)+M(2),'*');
023 hold on axis([-10 10 -5 10]);
024 plot(pYX2(1,:), pYX2(2,:)+M(2),'ro');
025
026 % LDA에 의한 분석 ----------------------------------
027 m1=mean(X1); m2=mean(X2);
028 Sw=N*cov(X1)+N*cov(X2);           % within scatter 계산
029 Sb=(m1-m2)'*(m1-m2);              % between scatter 계산
030 [V,D]=eig(Sb*inv(Sw));            % 고유치 분석
031 w=V(:,2);                        % 찾아진 벡터
032 figure(1);                       % 찾아진 벡터를 공간에 표시
033 line([0 w(1)*-8]+M(1),[0 w(2)*-8]+M(2));
034 YX1=w'*X1';  YX2=w'*X2';         % 벡터w 방향으로 사영
035 pYX1=w*YX1; pYX2=w*YX2;          % 사영된 데이터를 2차원 공간으로 환원
036 figure(2);                    % 사영된 데이터를 2차원 공간으로 환원하여 표시
037 plot(pYX1(1,:)+M(1), pYX1(2,:),'*');
038 plot(pYX2(1,:)+M(1), pYX2(2,:),'ro');
```

# 연습문제

1. [그림 8-9]에 나타난 데이터를 생성한 후, 주성분분석법과 선형판별분석법을 적용해 보시오.
   (1) 다음과 같은 분포에 따라 두 클래스의 데이터를 생성하시오.
      - 클래스 1 : $\mu_1 = \begin{bmatrix}5\\3\end{bmatrix}$, $\Sigma_1 = \begin{bmatrix}3 & 1.7\\1.7 & 1\end{bmatrix}$
      - 클래스 2 : $\mu_2 = \begin{bmatrix}1\\5\end{bmatrix}$, $\Sigma_2 = \begin{bmatrix}3 & 1.7\\1.7 & 1\end{bmatrix}$
   (2) 두 클래스를 하나로 합한 전체 데이터에 대해 PCA를 적용하여 첫 번째 주성분 벡터를 찾으시오.
   (3) 찾아진 주성분 벡터 방향으로 사영한 1차원 특징값들을 찾아서 2차원 평면상에 그려보시오.
   (4) 찾아진 벡터 방향으로 사영한 1차원 특징값들을 찾아서 2차원 평면상에 그려보시오.
   (5) 찾아진 벡터 방향으로 사영한 1차원 특징값들을 찾아서 2차원 평면상에 그려보시오.
   (6) PCA의 결과와 LDA의 결과를 비교해 보시오.
   (7) 같은 데이터에 대해, LDA를 적용하여 하나의 고유벡터를 찾아보고, [그림 8-9a]에서 사용된 것과 비교해 보시오.

2. 1번 문제와 같은 분포를 가진 테스트 데이터집합을 만들고, 최근접이웃 분류기를 이용하여 분류를 수행해 보시오.
   (1) 1에서 주성분분석법에 의해 찾아진 특징값을 이용하여 테스트 데이터에 대한 분류를 수행하고 분류율을 계산하시오.
   (2) 1에서 선형판별분석법에 의해 찾아진 특징값을 이용하여 테스트 데이터에 대한 분류를 수행하고 분류율을 계산하시오.

3. 3개의 클래스를 분류하는 문제에 대하여 다음 단계에 따라 선형판별분석법을 적용해 보시오.
   (1) 다음과 같은 분포에 따라 세 클래스의 데이터를 각각 100개씩 생성하고, 그래프로 나타내시오.
      - 클래스 1 : $\mu_1 = \begin{bmatrix}5\\3\end{bmatrix}$, $\Sigma_1 = \begin{bmatrix}3 & 1.7\\1.7 & 1\end{bmatrix}$
      - 클래스 2 : $\mu_2 = \begin{bmatrix}1\\5\end{bmatrix}$, $\Sigma_2 = \begin{bmatrix}3 & 1.7\\1.7 & 1\end{bmatrix}$
      - 클래스 3 : $\mu_2 = \begin{bmatrix}-2\\-2\end{bmatrix}$, $\Sigma_2 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$
   (2) LDA를 이용하여 변환행렬을 찾아보시오.
   (3) 찾아진 변환행렬을 이용하여 얻어진 2차원 특징데이터를 찾고, 2차원 평면상에 나타내 보시오.
   (4) (1)과 같은 분포를 가지는 테스트 데이터 집합을 각 클래스당 100개씩 생성하시오.