2021 Spring

Artificial Intelligence & Deep Learning

Prof. Minsuk Koo

Department of Computer Science & Engineering
Incheon National University

6.4 밀도 추정

- 6.4.1 커널 밀도 추정
- 6.4.2 가우시안 혼합
- 6.4.3 EM 알고리즘

밀도 추정 문제

- 어떤 점 $\mathbf{x}$에서 데이터가 발생할 확률, 즉 확률 분포 $P(\mathbf{x})$를 구하는 문제
- 예를 들어, 그림 6-8에서 $P(\mathbf{x}_1) > P(\mathbf{x}_2) > P(\mathbf{x}_3)$

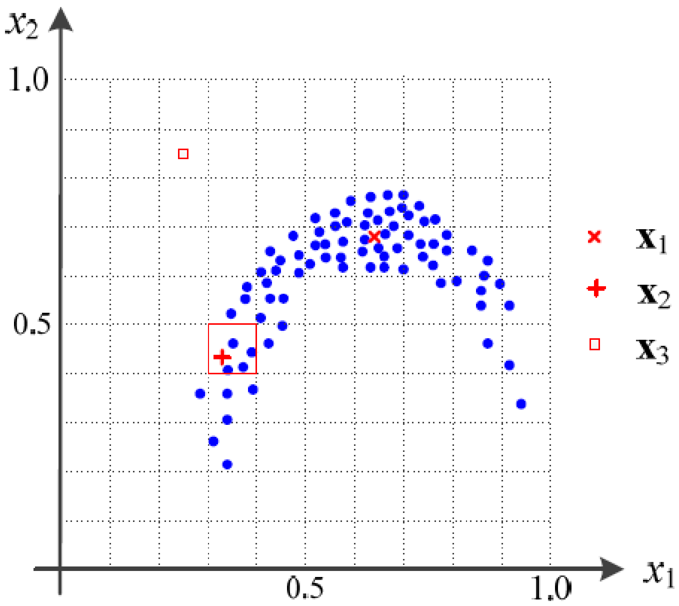


그림 6-8 밀도 추정 문제

6.4.1 커널 밀도 추정

히스토그램 방법

- 특징 공간을 칸의 집합으로 분할한 다음, 칸에 있는 샘플의 빈도를 세어 식 (6.7)로 추정

bin

- 예, $P(\mathbf{x} = +) = \frac{4}{80} = 0.05$

$$P(\mathbf{x}) = \frac{bin(\mathbf{x})}{n} \qquad (6.7)$$

- 여러 문제점
  - 매끄럽지 못하고 계단 모양을 띠는 확률밀도함수가 됨
  - 칸의 크기와 위치에 민감함

6.4.1 커널 밀도 추정

커널 밀도 추정법

- 점 $\mathbf{x}$에 [그림 6-9]가 예시하는 커널을 씌우고 커널 안에 있는 샘플의 가중 합을 이용함
- 대역폭 h의 크기가 중요

$$P_h(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K_h(\mathbf{x}-\mathbf{x}_i) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) \qquad (6.8)$$

여기서 $K_h(\mathbf{x}) = \frac{1}{h^d}K\left(\frac{\mathbf{x}}{h}\right)$

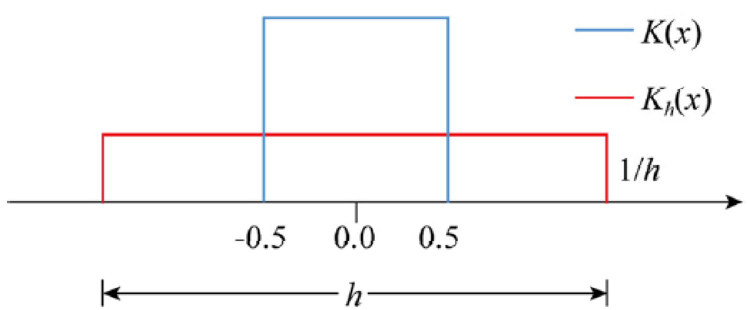


그림 6-9 표준커널함수 K와 크기 변환된 커널함수 K_h

6.4.1 커널 밀도 추정

- 히스토그램 방법과 커널 밀도 추정법의 비교
  - 커널 밀도 추정법은 매끄러운 확률밀도함수를 추정함

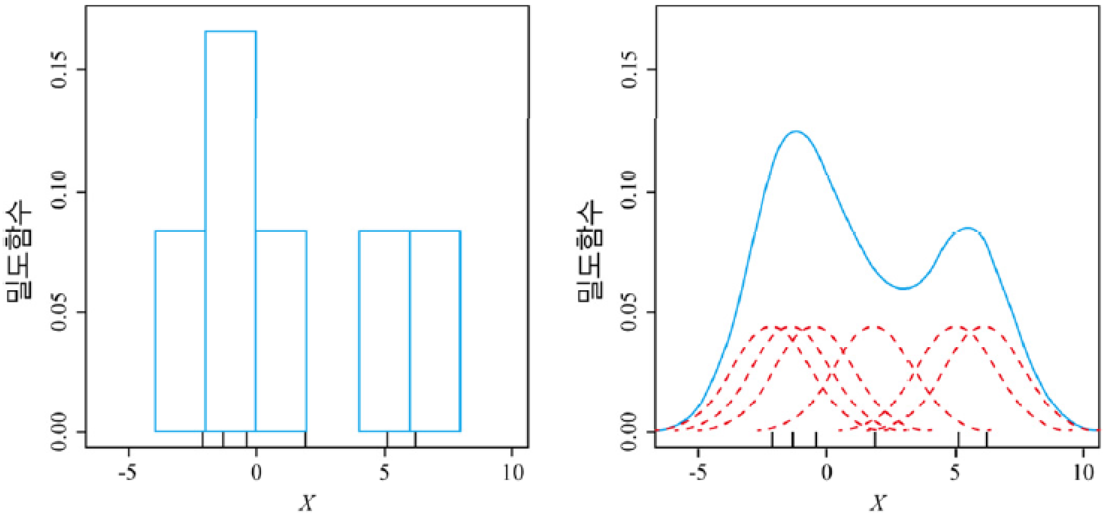


그림 6-10 히스토그램 방법(왼쪽)과 커널 밀도 추정법(오른쪽)의 비교

6.4.1 커널 밀도 추정

- 커널 밀도 추정법에서 대역폭 h의 중요성
  - h가 너무 작으면(빨강) 뾰족뾰족한 모양, h가 너무 크면(녹색) 뭉개짐, 적절하게 설정해야 함(검정)

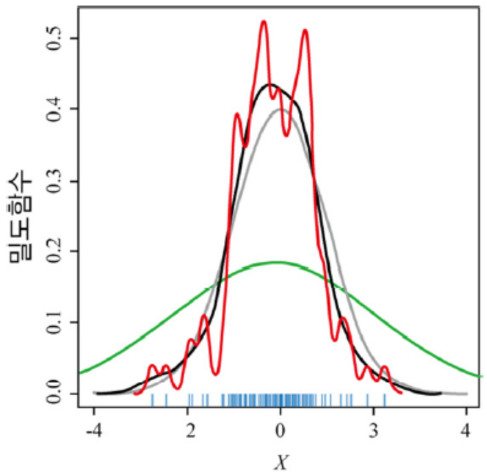


그림 6-11 대역폭이 확률밀도함수 추정에 미치는 영향

대역폭↓ : 분산↓, 높이↑

- 커널 밀도 추정 기법의 근본적 문제점
  - 샘플을 모두 저장하고 있어야 하는 메모리 기반 방법(새로운 샘플이 주어질 때마다 식 (6.8)을 처음부터 다시 계산)
  - 데이터 희소성(차원의 저주)

→ 데이터가 낮은 차원인 경우로 국한하여 활용

6.4.2 가우시안 혼합

- 가우시안을 이용한 방법(모수적 방법)
  - 데이터가 가우시안 분포를 따른다고 가정하고 평균 벡터 $\boldsymbol{\mu}$와 공분산 행렬 $\boldsymbol{\Sigma}$를 추정함

$$\left.\begin{array}{l} P(\mathbf{x}) = N(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \dfrac{1}{\sqrt{|\boldsymbol{\Sigma}|}\sqrt{(2\pi)^d}} \exp\left(-\dfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \\ \text{이때 } \boldsymbol{\mu} = \dfrac{1}{n}\sum_{i=1,n}\mathbf{x}_i, \boldsymbol{\Sigma} = \dfrac{1}{n}\sum_{i=1,n}(\mathbf{x}_i-\boldsymbol{\mu})(\mathbf{x}_i-\boldsymbol{\mu})^{\mathrm{T}} \end{array}\right\} \quad (6.9)$$

- 대부분 데이터가 하나의 가우시안으로 불충분([그림 6-12]의 오른쪽)

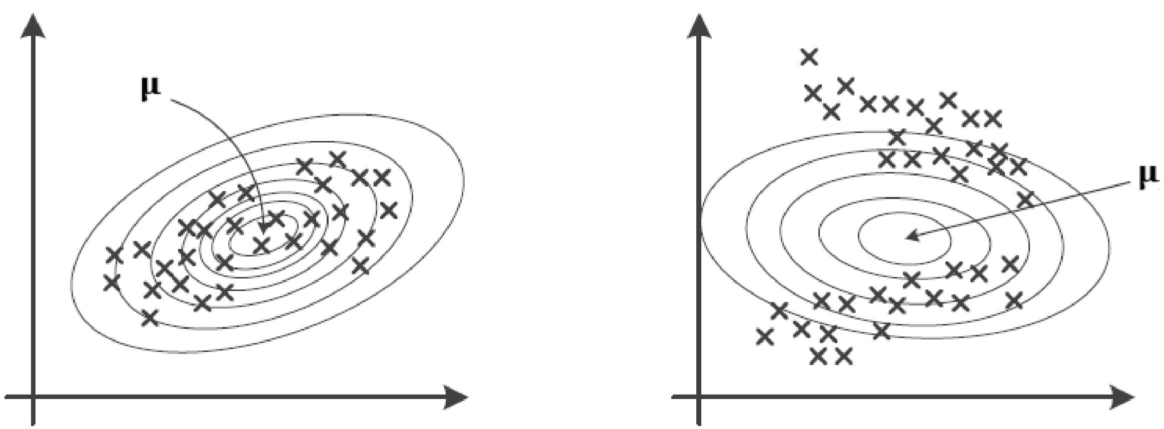
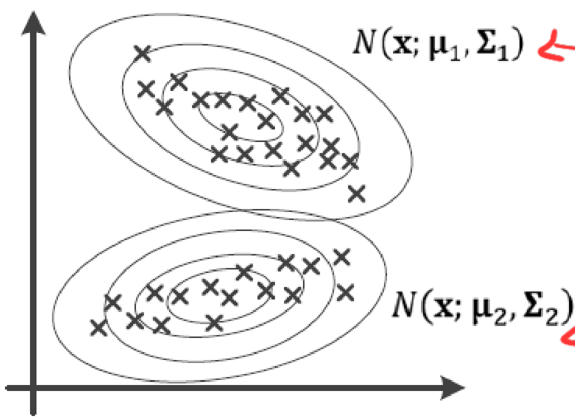


그림 6-12 하나의 가우시안으로 밀도 추정

6.4.2 가우시안 혼합

■ 가우시안 혼합

- [그림 6-13]은 2개의 가우시안을 사용한 예



$$P(\mathbf{x}) = \frac{21}{37} N(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \frac{16}{37} N(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

총 데이터개수

그림 6-13 가우시안 혼합으로 밀도 추정

■ k개의 가우시안으로 일반화하면,

- 확률분포 $P(\mathbf{x})$는 k개 가우시안의 선형 결합으로 표현(식 (6.10))

$$P(\mathbf{x}) = \sum_{j=1}^{k} \pi_j N(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{6.10}$$

6.4.2 가우시안 혼합

- 주어진 데이터와 추정해야 할 매개변수를 정리하면,

주어진 데이터: 훈련집합 $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, 가우시안의 개수 k

추정해야 할 매개변수집합: $\Theta = \{\boldsymbol{\pi} = \underbrace{(\pi_1, \pi_2, \cdots, \pi_k)}_{k개}, (\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), (\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \cdots, (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$

각각의 가우시안 분포에 해당하는 μ 와 공분산 Σ

- 최대 우도를 이용한 최적화 문제로 공식화

$$P(\mathbb{X}|\Theta) = \prod_{i=1}^{n} P(\mathbf{x}_i|\Theta) = \prod_{i=1}^{n}\left(\sum_{j=1}^{k} \pi_j N(\mathbf{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\right) \tag{6.11}$$

$$\log P(\mathbb{X}|\Theta) = \sum_{i=1}^{n} \log\left(\sum_{j=1}^{k} \pi_j N(\mathbf{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\right) \tag{6.12}$$

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \log P(\mathbb{X}|\Theta) \tag{6.13}$$

6.4.3 EM 알고리즘

- EM 알고리즘을 이용한 식 (6.13)의 풀이
  - Θ를 모르므로 난수로 설정하고 출발([그림 6-14]의 예시)

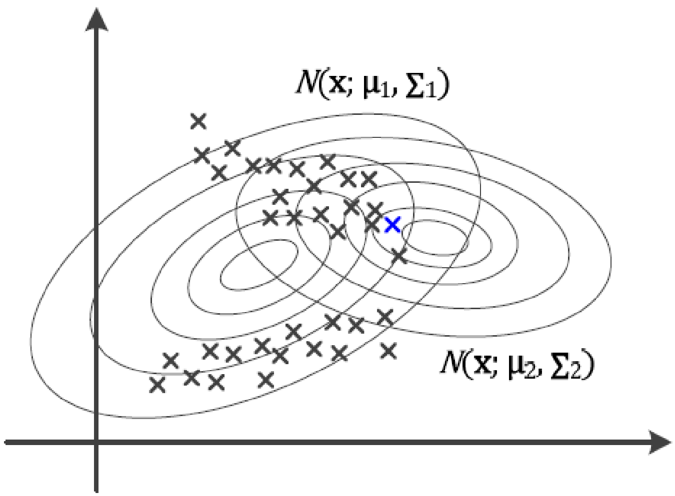


그림 6-14 샘플의 소속 확률을 어떻게 추정할 것인가

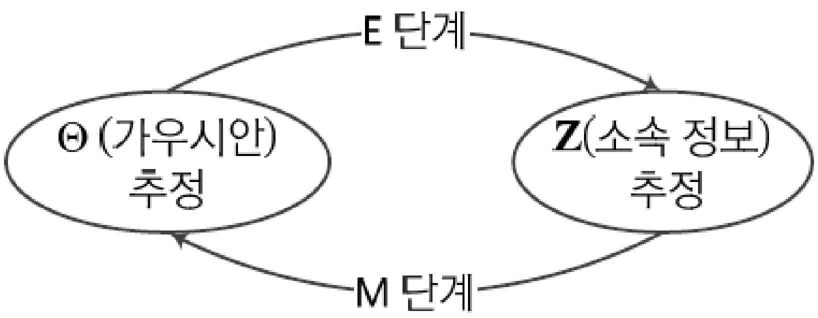


그림 6-15 가우시안 혼합을 위한 EM 알고리즘

  - 가우시안으로 샘플의 소속 정보 개선(E단계) → 샘플의 소속 정보로 가우시안 개선(M단계) → 가우시안으로 샘플의 소속 정보 개선(E단계) → 샘플의 소속 정보로 가우시안 개선(M단계) → .... ([그림 6-15])

6.4.3 EM 알고리즘

가우시안 혼합을 위한 EM 알고리즘

알고리즘 6-4 가우시안 혼합을 위한 EM 알고리즘

입력: 훈련집합 $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, 가우시안의 개수 k

출력: 최적의 가우시안과 혼합 계수 $\Theta = \{\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_k), (\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), (\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \cdots, (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$

```
1  Θ를 초기화한다.
2  while (!멈춤조건)
3     Θ를 이용하여 소속확률 행렬 Z를 추정한다.   // E단계
4     Z를 이용하여 Θ를 추정한다.               // M단계
```

- 라인 3과 라인 4를 위한 수식
- z_{ji}는 $\mathbf{x}_i$가 j번째 가우시안에 속할 확률

$$z_{ji} = \frac{\pi_j N(\mathbf{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{q=1}^{k} \pi_q N(\mathbf{x}_i; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)} \tag{6.14}$$

$$\left.\begin{aligned} \boldsymbol{\mu}_j &= \frac{1}{n_j} \sum_{i=1}^{n} z_{ji} \mathbf{x}_i \\ \boldsymbol{\Sigma}_j &= \frac{1}{n_j} \sum_{i=1}^{n} z_{ji} (\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}} \\ \pi_j &= \frac{n_j}{n} \\ \text{이때 } n_j &= \sum_{i=1}^{n} z_{ji} \end{aligned}\right\} \tag{6.15}$$

6.5 공간 변환의 이해

- 간단한 상황 예시
  - 2개 군집을 가진 [그림 6-16]의 2차원 특징 공간을 극좌표 공간으로 변환하면 1차원만으로 2개 군집 표현 가능

각도와 크기

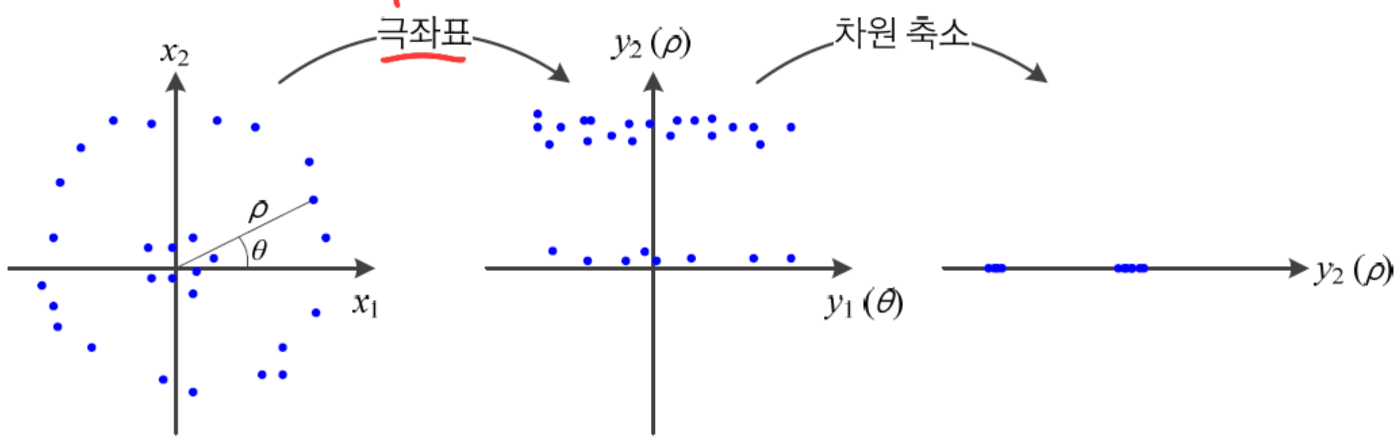


그림 6-16 공간 변환의 예

- 실제 문제에서는 비지도 학습을 이용하여 최적의 공간 변환을 자동으로 알아내야 함

6.5 공간 변환의 이해

인코딩과 디코딩

- 원래 공간을 다른 공간으로 변환하는 인코딩 과정(f), 변환 공간을 원래 공간으로 역변환하는 디코딩 과정(g)

$$\hat{\mathbf{x}} = g(f(\mathbf{x})) \quad (6.16)$$

$x \xrightarrow{\text{인코딩}} f(x) \xrightarrow{\text{디코딩}} g(f(x)) = \hat{x}$

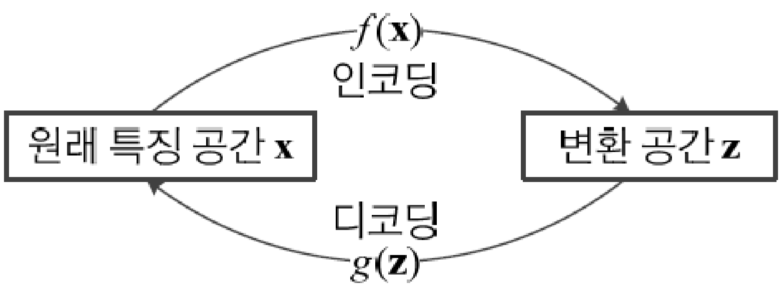


그림 6-17 공간 변환과 역변환

- 예) 데이터 압축의 경우, 역변환으로 얻은 $\hat{\mathbf{x}}$은 원래 신호 $\mathbf{x}$와 가급적 같아야 함
- 예) 데이터 가시화에서는 2차원 또는 3차원의 $\mathbf{z}$ 공간으로 변환. 디코딩은 불필요

6.6 선형 인자 모델

- 6.6.1 주성분 분석
- 6.6.2 독립 성분 분석
- 6.6.3 희소 코딩

6.6 선형 인자 모델

선형 인자 모델

- 선형 연산을 이용한 공간 변환 기법
- 선형 연산을 사용하므로 행렬 곱으로 인코딩(식 (6.17))과 디코딩(식 (6.18)) 과정을 표현

인코더

$$f\colon\ \mathbf{z} = \mathbf{W}_{enc}\mathbf{x} + \boldsymbol{\alpha}_{enc} \qquad (6.17)$$

← encoding

$$g\colon\ \mathbf{x} = \mathbf{W}_{dec}\mathbf{z} + \boldsymbol{\alpha}_{dec} \qquad (6.18)$$

← decoding

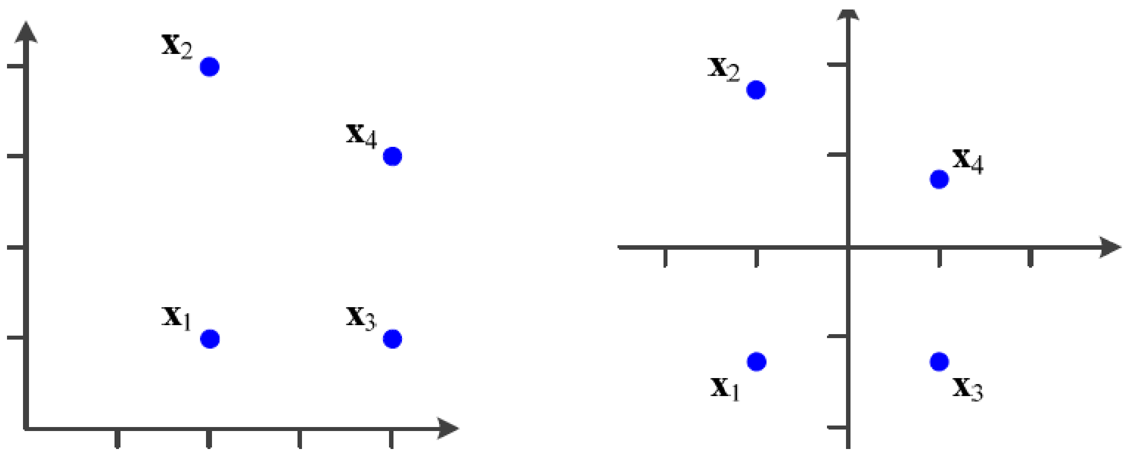
디코더

- $\boldsymbol{\alpha}$는 데이터를 원점으로 이동하거나 잡음을 추가하는 등의 역할
- 인자 $\mathbf{z}$와 추가 항 $\boldsymbol{\alpha}$에 따라 여러 가지 모델이 존재
  - $\mathbf{z}$에 확률 개념이 없고 $\boldsymbol{\alpha}$를 생략하면 PCA(6.6.1절) – 관찰 벡터 $\mathbf{x}$와 인자 $\mathbf{z}$는 결정론적인 1:1 매핑 관계
  - $\mathbf{z}$와 $\boldsymbol{\alpha}$가 가우시안 분포를 따른다고 가정하면 확률 PCA[probabilistic PCA]
  - $\mathbf{z}$가 비가우시안 분포를 따른다고 가정하는 ICA(6.6.2절)

6.6.1 주성분 분석

- 데이터를 원점 중심으로 옮기는 전처리를 먼저 수행

$$\left.\begin{aligned} &\mathbf{x}_i = \mathbf{x}_i - \boldsymbol{\mu}, \qquad i = 1,2,\cdots,n \\ &\text{이때 } \boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \end{aligned}\right\} \qquad (6.19)$$



(a) 원래 훈련집합 $\mathbb{X}$　　(b) $\mathbb{X}$에 식 (6.19)를 적용한 이후

그림 6-18 $\mathbb{X}$의 평균을 0으로 변환

6.6.1 주성분 분석

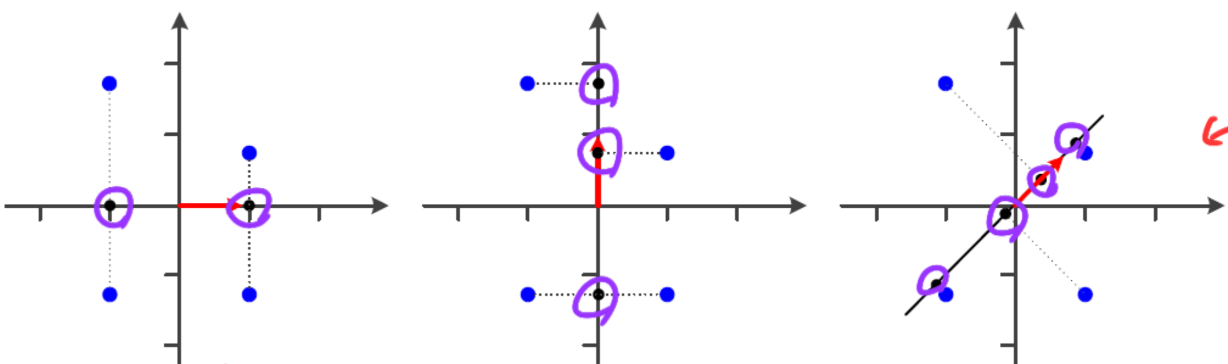
주성분 분석이 사용하는 변환식

- 일반적인 선형 변환식인 식 (6.17)에서 $\mathbf{z}$에 확률 개념이 없고 $\boldsymbol{\alpha}$를 생략하면 주성분 분석
- 변환 행렬 $\mathbf{W}$는 $d * q$로서 주성분 분석은 d차원의 $\mathbf{x}$를 q차원의 $\mathbf{z}$로 변환 $(q < d)$
  matrix
  - $\mathbf{W}$의 j번째 열 벡터와의 내적 $\mathbf{u}_j^{\mathrm{T}}\mathbf{x}$는 $\mathbf{x}$를 $\mathbf{u}_j$가 가리키는 축으로 투영

$$\left.\begin{aligned} &\mathbf{z} = \mathbf{W}^{\mathrm{T}}\mathbf{x} \\ &\text{이때 } \mathbf{W} = (\mathbf{u}_1\ \mathbf{u}_2\ \cdots\ \mathbf{u}_q)\text{이고, } \mathbf{u}_j = (u_{1j}, u_{2j}, \cdots, u_{dj})^{\mathrm{T}} \end{aligned}\right\} \quad (6.20)$$

$$W = \begin{bmatrix} u_1 & u_2 & \cdots & u_q \end{bmatrix}$$

- 예, 2차원을 1차원으로 변환하는 상황$(d = 2, q = 1)$



(a) $\mathbf{u} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ 축으로 투영 — x축으로

(b) $\mathbf{u} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ 축으로 투영 — y축으로

(c) $\mathbf{u} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$ 축으로 투영 — y=x 축

데이터 공간 압축
"
정보유지
but 차원↓

그림 6-19 투영에 의해 2차원을 1차원으로 변환

6.6.1 주성분 분석

주성분 분석의 목적

- 손실을 최소화하면서 저차원으로 변환하는 것
  - [그림 6-19]에서 정보 손실 예
    - [그림 6-19(a)]는 $\mathbf{x}_1$과 $\mathbf{x}_2$ 쌍, $\mathbf{x}_3$과 $\mathbf{x}_4$ 쌍이 같은 점으로 변환되는 정보 손실
    - [그림 6-19(b)]는 $\mathbf{x}_1$과 $\mathbf{x}_3$ 쌍이 같은 점으로 변환되는 정보 손실
    - [그림 6-19(c)]는 4개 점이 모두 다른 점으로 변환되어 정보 손실이 가장 적음
- 주성분 분석은 변환된 훈련집합 $\mathbb{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n\}$의 분산이 클수록 정보 손실이 적다고 판단

6.6.1 주성분 분석

예제 6-2 **[그림 6-19]의 세 가지 경우의 분산**

↳ 축으로

[그림 6-18(a)]의 훈련집합에 식 (6.19)를 적용하기 전과 후는 다음과 같다.

$$\mathbf{x}_1=\begin{pmatrix}2\\1\end{pmatrix},\ \mathbf{x}_2=\begin{pmatrix}2\\4\end{pmatrix},\ \mathbf{x}_3=\begin{pmatrix}4\\1\end{pmatrix},\ \mathbf{x}_4=\begin{pmatrix}4\\3\end{pmatrix}\ \Rightarrow\ \mathbf{x}_1=\begin{pmatrix}-1\\-1.25\end{pmatrix},\ \mathbf{x}_2=\begin{pmatrix}-1\\1.75\end{pmatrix},\ \mathbf{x}_3=\begin{pmatrix}1\\-1.25\end{pmatrix},\ \mathbf{x}_4=\begin{pmatrix}1\\0.75\end{pmatrix}$$

[그림 6-19(a)]의 $\mathbf{u}=(1\ 0)^{\mathrm{T}}$ 축으로 투영된 점은 다음과 같다. $z_1 \sim z_4$의 분산은 1.0이다.

$$z_1=(1\ 0)\begin{pmatrix}-1\\-1.25\end{pmatrix}=-1,\ z_2=(1\ 0)\begin{pmatrix}-1\\1.75\end{pmatrix}=-1,\ z_3=(1\ 0)\begin{pmatrix}1\\-1.25\end{pmatrix}=1,\ z_4=(1\ 0)\begin{pmatrix}1\\0.75\end{pmatrix}=1$$

이제 [그림 6-19(c)]의 $\mathbf{u}=\left(\frac{1}{\sqrt{2}}\ \frac{1}{\sqrt{2}}\right)^{\mathrm{T}}$ 축으로 투영된 점을 구해 보자. $z_1 \sim z_4$의 분산은 1.093이다.

$$z_1=\left(\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}\right)\begin{pmatrix}-1\\-1.25\end{pmatrix}=-1.591,\ z_2=\left(\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}\right)\begin{pmatrix}-1\\1.75\end{pmatrix}=0.530,$$
$$z_3=\left(\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}\right)\begin{pmatrix}1\\-1.25\end{pmatrix}=-0.177,\ z_4=\left(\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}\right)\begin{pmatrix}1\\0.75\end{pmatrix}=1.237$$

따라서 $\mathbf{u}=\left(\frac{1}{\sqrt{2}}\ \frac{1}{\sqrt{2}}\right)^{\mathrm{T}}$ 축이 $\mathbf{u}=(1\quad 0)^{\mathrm{T}}$보다 우수하다고 할 수 있다. 그렇다면 $\mathbf{u}=\left(\frac{1}{\sqrt{2}}\ \frac{1}{\sqrt{2}}\right)^{\mathrm{T}}$ 보다 더 좋은 축이 있을까? 이제부터 최적해를 찾는 방법을 살펴보자.

6.6.1 주성분 분석

PCA의 최적화 문제

문제 6.1 $\mathbb{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n\}$의 분산을 최대화하는 q개의 축, 즉 $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_q$를 찾아라. 이 단위 벡터는 식 (6.20)에 따라 변환 행렬 $\mathbf{W}$를 구성한다.

- $q = 1$로 국한하고 분산을 쓰면,

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2 = \frac{1}{n}\sum_{i=1}^{n}{z_i}^2 = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_i)^2 = \mathbf{u}^{\mathrm{T}}\Sigma\mathbf{u} \tag{6.21}$$

- [문제 6.1]을 바꾸어 쓰면,

max

문제 6.2 식 (6.21)의 분산 σ^2을 최대로 하는 $\mathbf{u}$를 찾아라.

6.6.1 주성분 분석

- PCA의 최적화 문제
  - $\mathbf{u}$가 단위 벡터라는 사실을 적용하여 문제를 다시 쓰면,

라그랑지 승수법

문제 6.3 $L(\mathbf{u}) = \mathbf{u}^{\mathrm{T}}\Sigma\mathbf{u} + \lambda(1 - \mathbf{u}^{\mathrm{T}}\mathbf{u})$를 최대로 하는 $\mathbf{u}$를 찾아라.

  - $L(\mathbf{u})$를 $\mathbf{u}$로 미분하면, $\dfrac{\partial L(\mathbf{u})}{\partial \mathbf{u}} = 2\mathbf{\Sigma}\mathbf{u} - 2\lambda\mathbf{u}$
  - $\dfrac{\partial L}{\partial \mathbf{u}} = 0$을 풀면,

$$\mathbf{\Sigma}\mathbf{u} = \lambda\mathbf{u} \tag{6.22}$$

vector의 크기만 변하게 된다.

- 주성분 분석의 학습 알고리즘

1. 훈련집합으로 공분산 행렬 $\mathbf{\Sigma}$를 계산한다.
2. 식 (6.22)를 풀어 d개의 고윳값과 고유 벡터를 구한다.
3. 고윳값이 큰 순서대로 $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \cdots, \mathbf{u}_d$를 나열한다. (이들을 주성분이라 부름)
4. q개의 주성분 $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_q$를 선택하여 식 (6.20)에 있는 행렬 $\mathbf{W}$에 채운다.

6.6.1 주성분 분석

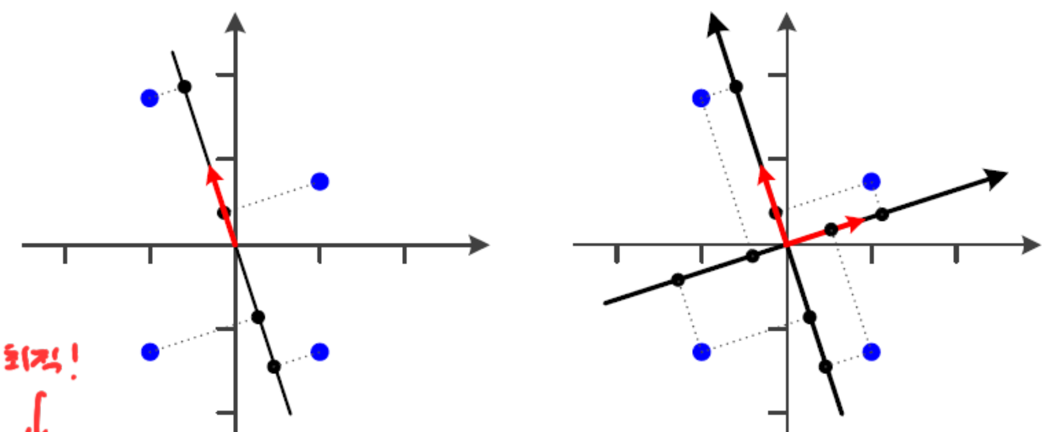
예제 6-3 **PCA 수행**

식 (6.22)를 풀어 [그림 6-18]에 있는 데이터의 최적해를 구해 보자. 먼저 공분산 행렬 $\mathbf{\Sigma}$와 $\mathbf{\Sigma}$의 고윳값과 고유 벡터를 구하면 다음과 같다. 공분산을 구하는 방법은 2장의 식 (2.39)를 참조하라.

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.000 & -0.250 \\ -0.250 & 1.688 \end{pmatrix}$$

$$\lambda_1 = 1.7688, \boldsymbol{u}_1 = \begin{pmatrix} -0.3092 \\ 0.9510 \end{pmatrix}, \quad \lambda_2 = 0.9187, \boldsymbol{u}_2 = \begin{pmatrix} -0.9510 \\ -0.3092 \end{pmatrix}$$

고유 벡터 2개 중 고윳값이 큰 $\mathbf{u}_1$을 선택하고, $\mathbf{u}_1$에 샘플 4개를 투영하면 [그림 6-20(a)]가 된다. 변환된 점의 분산은 1.7688로 [그림 6-19]에 있는 축보다 훨씬 크다는 사실을 확인할 수 있다. $\mathbf{u}_1$은 PCA 알고리즘으로 찾은 최적으로서 더 좋은 축은 없다.



최적!

(a) 축 1개 사용 λ값이 큰 걸 우선으로

(b) 축 2개 사용

그림 6-20 PCA가 찾은 최적 변환

6.6.1 주성분 분석

디코딩 과정

- 역변환은 $\mathbf{x} = (\mathbf{W}^{\mathrm{T}})^{-1}\mathbf{z}$인데, $\mathbf{W}$가 정규직교 행렬이므로 식 (6.23)이 됨

$$\tilde{\mathbf{x}} = \mathbf{W}\mathbf{z} \tag{6.23}$$

- $q = d$로 설정하면 $\mathbf{W}$가 $d * d$이고 $\tilde{\mathbf{x}}$는 원래 샘플 $\mathbf{x}$와 같게 됨([그림 6-20(b)]의 예시)
  - 원래 공간을 단지 일정한 양만큼 회전하는 것에 불과

실제로는 $q < d$로 설정하여 차원 축소를 꾀함

- 많은 응용이 있음
  - 데이터 압축
  - $q = 2$ 또는 $q = 3$으로 설정하여 2차원 또는 3차원으로 축소하여 데이터 가시화
  - 고유얼굴 기법: 256*256 얼굴 영상($d = 65536$)을 $q = 7$차원으로 변환하여 얼굴 인식(정면 얼굴에 대해 96% 정확률) → 상위 몇 개의 고유 벡터가 대부분 정보를 가짐