기계학습

이현석 교수

2023-2

기계 학습 개념

- 간단한 기계 학습 예제
  - 가로축은 시간, 세로축은 이동체의 위치
  - 관측한 4개의 점이 데이터
- 예측prediction 문제
  - 임의의 시간이 주어지면 이때 이동체의 위치는?
  - 회귀regression 문제와 분류classification 문제로 나뉨
    - 회귀는 목표치가 실수, 분류는 부류값 ([그림 1-4]는 회귀 문제)

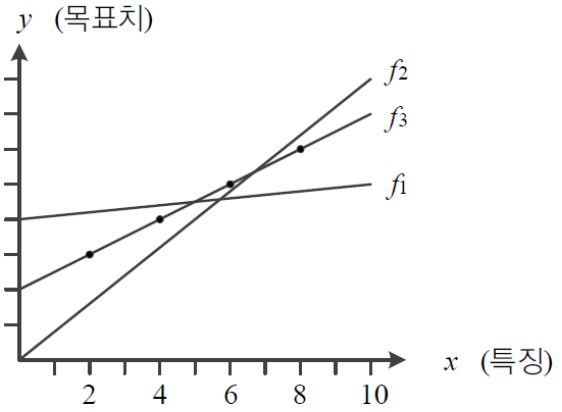


그림 1-4 간단한 기계 학습 예제

기계 학습 개념

훈련집합

- 가로축은 특징, 세로축은 목표치
- 관측한 4개의 점이 훈련집합을 구성함

$$\text{훈련집합: } \mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}, \quad \mathbb{Y} = \{y_1, y_2, \cdots, y_n\} \tag{1.1}$$

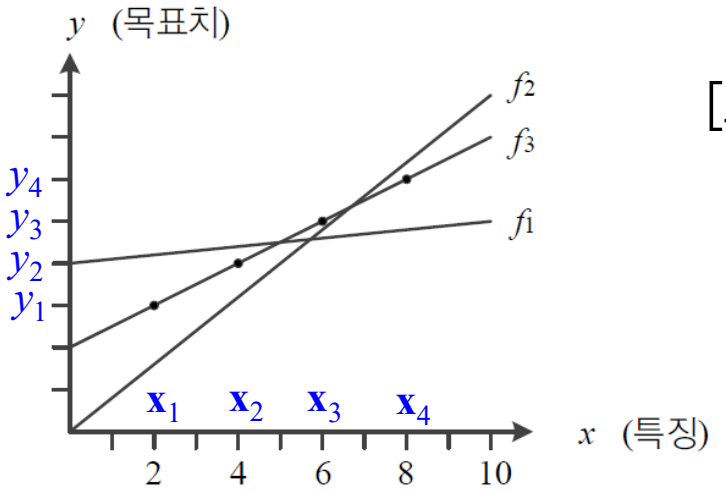


그림 1-4 간단한 기계 학습 예제

[그림 1-4] 예제의 훈련집합

$$\mathbb{X} = \{\mathbf{x}_1 = (2.0), \mathbf{x}_2 = (4.0), \mathbf{x}_3 = (6.0), \mathbf{x}_4 = (8.0)\}$$
$$\mathbb{Y} = \{y_1 = 3.0, y_2 = 4.0, y_3 = 5.0, y_4 = 6.0\}$$

기계 학습 개념

데이터를 어떻게 모델링할 것인가

- 눈대중으로 보면 직선을 이루므로 직선을 선택하자 → 모델로 직선을 선택한 셈
- 직선 모델의 수식
  - 2개의 매개변수 w와 b

$$y = wx + b \qquad (1.2)$$

기계 학습은

- 가장 정확하게 예측할 수 있는, 즉 최적의 매개변수를 찾는 작업
- 처음에는 최적값을 모르므로 임의의 값에서 시작하고, 점점 성능을 개선하여 최적에 도달
- [그림 1-4]의 예에서는 f_1에서 시작하여 $f_1 \rightarrow f_2 \rightarrow f_3$
  - 최적인 f_3은 w=0.5와 b=2.0

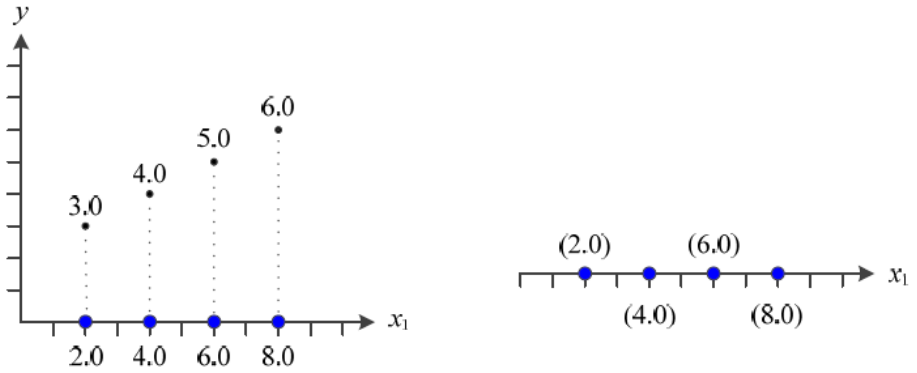
기계 학습 개념

- 학습을 마치면,
  - 예측에 사용
  - 예) 10.0 순간의 이동체 위치를 알고자 하면, $f_3(10.0)=0.5*10.0+2.0=7.0$이라 예측함

- 기계 학습의 궁극적인 목표
  - 훈련집합에 없는 새로운 샘플에 대한 오류를 최소화 (새로운 샘플 집합: 테스트 집합)
  - 테스트 집합에 대한 높은 성능을 일반화generalization 능력이라 부름

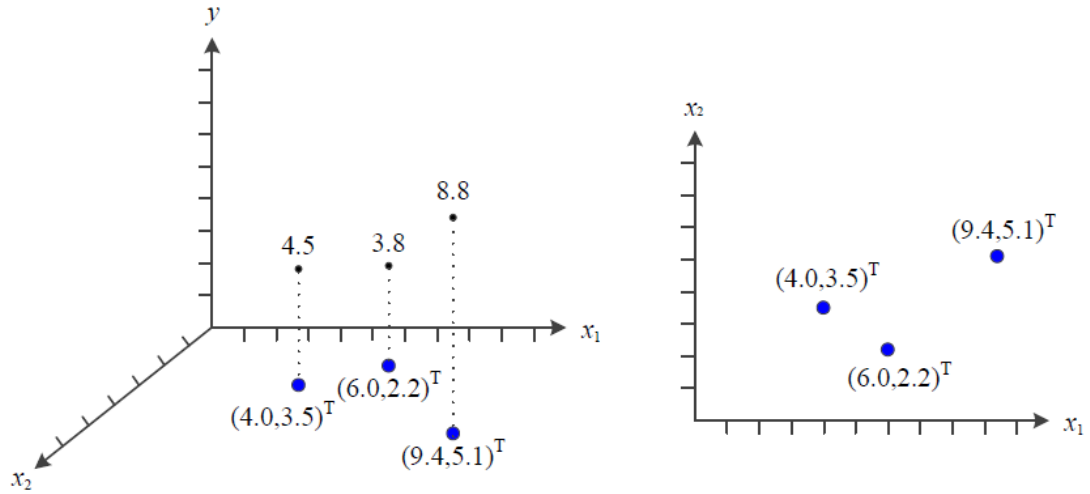
1차원과 2차원 특징 공간

- 1차원 특징 공간



(a) 1차원 특징 공간(왼쪽: 특징과 목푯값을 축으로 표시, 오른쪽: 특징만 축으로 표시)

- 2차원 특징 공간
  - 특징 벡터 표기
    - x=(x1,x2)T
  - 예시
    - x=(몸무게,키)T, y=장타율
    - x=(체온,두통)T, y=감기 여부

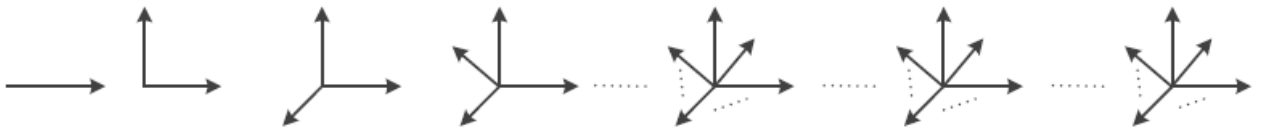


(b) 2차원 특징 공간(왼쪽: 특징 벡터와 목푯값을 축으로 표시, 오른쪽: 특징 벡터만 축으로 표시)

그림 1-5 특징 공간과 데이터의 표현

다차원 특징 공간

- 다차원 특징 공간 예제



Haberman survival: $\mathbf{x} = (\text{나이}, \text{수술년도}, \text{양성 림프샘 개수})^T$

Iris: $\mathbf{x} = (\text{꽃받침 길이}, \text{꽃받침 너비}, \text{꽃잎 길이}, \text{꽃잎 너비})^T$

Wine: $\mathbf{x} = (\text{Alcohol}, \text{Malic acid}, \text{Ash}, \text{Alcalinity of ash}, \text{Magnesium}, \text{Total phenols}, \text{Flavanoids}, \text{Nonflavanoid phenols}$
$\text{Proanthocyanins}, \text{Color intensity}, \text{Hue}, \text{OD280 / OD315 of diluted wines}, \text{Proline})^T$

MNIST: $\mathbf{x} = (\text{화소1}, \text{화소2}, \cdots, \text{화소784})^T$

Farm ads: $\mathbf{x} = (\text{단어1}, \text{단어2}, \cdots, \text{단어54877})^T$

그림 1-6 다차원 특징 공간

다차원 특징 공간

- *d*-차원 데이터
  - 특징 벡터 표기: $\mathbf{x}=(x_1,x_2, \ldots ,x_d)^T$
- *d*-차원 데이터를 위한학습 모델
  - 직선 모델을 사용하는 경우 매개변수 수=$d+1$

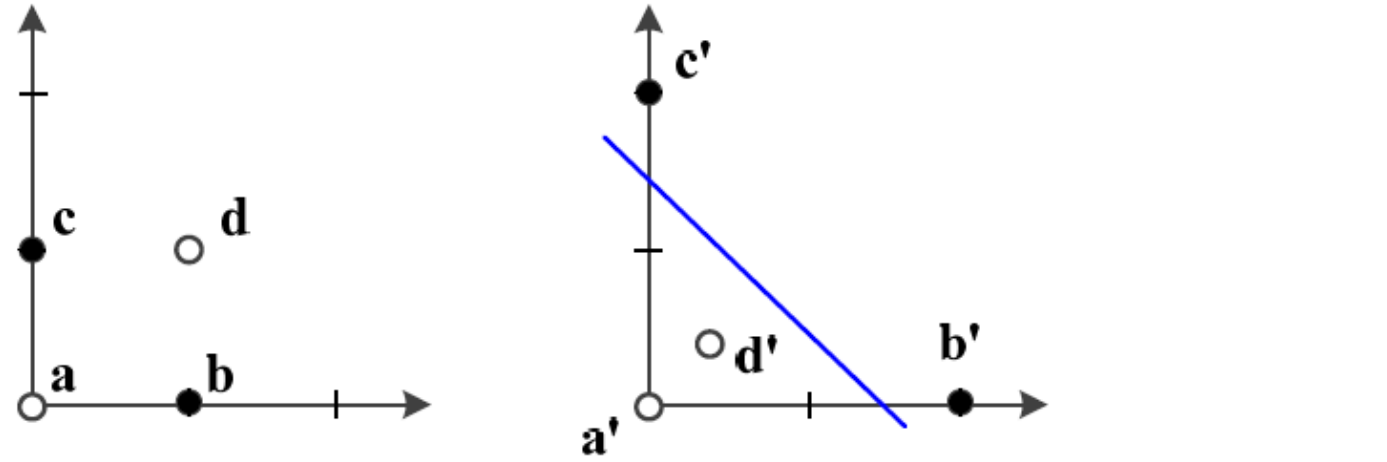
$$y = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b \tag{1.3}$$

  - 2차 곡선 모델을 사용하면 매개변수 수가 크게 증가
    - 매개변수 수=d^2+d+1
    - 예) Iris 데이터: d=4이므로 21개의 매개변수

$$y = w_1 x_1^2 + w_2 x_2^2 + \cdots + w_d x_d^2 + w_{d+1} x_1 x_2 + \cdots + w_{d^2} x_{d-1} x_d + w_{d^2+1} x_1 + \cdots + w_{d^2+d} x_d + b \tag{1.5}$$

특징 공간 변환과 표현 학습

- 선형 분리 불가능linearly non-separable한 원래 특징 공간 ([그림 1-7(a)])
  - 직선 모델을 적용하면 75% 정확률이 한계



(a) 원래 특징 공간 (b) 분류에 더 유리하도록 변환된 새로운 특징 공간

그림 1-7 특징 공간 변환

특징 공간 변환과 표현 학습

식 (1.6)으로 변환된 새로운 특징 공간 ([그림 1-7(b)])

- 직선 모델로 100% 정확률

$$\text{원래 특징 벡터 } \mathbf{x} = (x_1, x_2)^{\mathrm{T}} \rightarrow \text{변환된 특징 벡터 } \mathbf{x}' = \left(\frac{x_1}{2x_1x_2 + 0.5}, \frac{x_2}{2x_1x_2 + 0.5} \right)^{\mathrm{T}} \quad (1.6)$$

$$\mathbf{a} = (0,0)^{\mathrm{T}} \rightarrow \mathbf{a}' = (0,0)^{\mathrm{T}}$$

$$\mathbf{b} = (1,0)^{\mathrm{T}} \rightarrow \mathbf{b}' = (2,0)^{\mathrm{T}}$$

$$\mathbf{c} = (0,1)^{\mathrm{T}} \rightarrow \mathbf{c}' = (0,2)^{\mathrm{T}}$$

$$\mathbf{d} = (1,1)^{\mathrm{T}} \rightarrow \mathbf{d}' = (0.4,0.4)^{\mathrm{T}}$$

표현 학습representation learning

- 좋은 특징 공간을 자동으로 찾는 작업
- 딥러닝은 다수의 은닉층을 가진 신경망을 이용하여 계층적인 특징 공간을 찾아냄
  - 왼쪽 은닉층은 저급 특징(에지, 구석점 등), 오른쪽은 고급 특징(얼굴, 바퀴 등) 추출
- [그림 1-7]은 표현 학습을 사람이 직관으로 수행한 셈

특징 공간 변환과 표현 학습

차원에 대한 몇 가지 설명

- 차원에 무관하게 수식 적용 가능함
  - 예) 두 점 $\mathbf{a}=(a_1,a_2,\ldots,a_d)^T$와 $\mathbf{b}=(b_1,b_2,\ldots,b_d)^T$ 사이의 거리는 모든 d에 대해 성립

$$dist(\mathbf{a},\mathbf{b}) = \sqrt{\sum_{i=1}^{d}(a_i - b_i)^2} \qquad (1.7)$$

  - 보통 2~3차원의 저차원에서 식을 고안한 다음 고차원으로 확장 적용

- 차원의 저주 (curse of dimensionality)
  - 차원이 높아짐에 따라 발생하는 현실적인 문제들
  - 예) d=4인 Iris 데이터에서 축마다 100개 구간으로 나누면 총 100^4=1억 개의 칸
  - 예) d=784인 MNIST 샘플의 화소가 0과 1값을 가진다면 2^{784}개의 칸. 이 거대한 공간에 고작 6만 개의 샘플을 흩뿌린 매우 희소한 분포

데이터에 대한 이해

과학 기술의 발전 과정

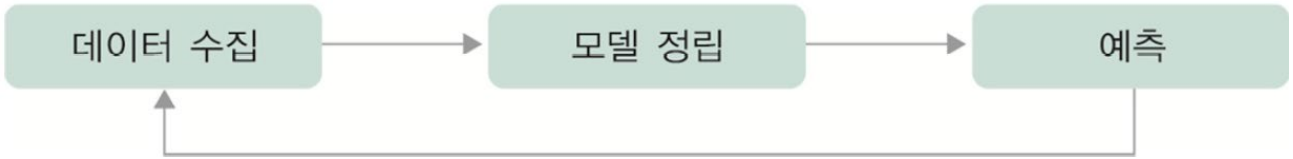


그림 1-8 과학기술의 발전 과정

- 예) 튀코 브라헤는 천동설이라는 틀린 모델을 선택함으로써 자신이 수집한 데이터를 설명하지 못함. 케플러는 지동설 모델을 도입하여 제1, 제2, 제 3법칙을 완성함

기계 학습

- 기계 학습이 푸는 문제는 훨씬 복잡함
- 단순한 수학 공식으로 표현 불가능함
- 자동으로 모델을 찾아내는 과정이 필수

데이터 생성 과정

- 데이터 생성 과정을 완전히 아는 인위적 상황의 예제
  - 예) 두 개 주사위를 던져 나온 눈의 합을 x라 할 때, $y=(x-7)^2+1$ 점을 받는 게임
  - 이런 상황을 '데이터 생성 과정을 완전히 알고 있다'고 말함
    - x를 알면 정확히 y를 예측할 수 있음
      - 실제 주사위를 던져 $\mathbb{X} = \{3,10,8,5\}$를 얻었다면, $\mathbb{Y} = \{17,10,2,5\}$
    - x의 발생 확률 $P(x)$를 정확히 알 수 있음
    - $P(x)$를 알고 있으므로, 새로운 데이터 생성 가능

- [그림 1-6]과 같은 실제 기계 학습 문제
  - 데이터 생성 과정을 알 수 없음
  - 단지 주어진 훈련집합 $\mathbb{X}$, $\mathbb{Y}$로 예측 모델 또는 생성 모델을 근사 추정할 수 있을 뿐

데이터베이스의 중요성

데이터베이스의 품질

- 주어진 응용에 맞는 충분히 다양한 데이터를 충분한 양만큼 수집 → 추정 정확도 높아짐
- 예) 정면 얼굴만 가진 데이터베이스로 학습하고 나면, 기운 얼굴은 매우 낮은 성능

→주어진 응용 환경을 자세히 살핀 다음 그에 맞는 데이터베이스 확보는 아주 중요함

아주 많은 공개 데이터베이스

- 기계 학습의 초파리로 여겨지는 3가지 데이터베이스: Iris, MNIST, ImageNet
- 위키피디아에서 'list of datasets for machine learning research'로 검색
- UCI 리퍼지토리 (2017년11월 기준으로 394개 데이터베이스 제공)

데이터베이스의 중요성

- Iris 데이터베이스는 통계학자인 피셔 교수가 1936년에 캐나다 동부 해안의 가스페 반도에 서식하는 3종의 붓꽃(setosa, versicolor, virginica)을 50송이씩 채취하여 만들었다[Fisher1936]. 150개 샘플 각각에 대해 꽃받침 길이, 꽃받침 너비, 꽃잎 길이, 꽃잎 너비를 측정하여 기록하였다. 따라서 4차원 특징 공간이 형성되며 목푯값은 3종을 숫자로 표시함으로써 1, 2, 3 값 중의 하나이다. http://archive.ics.uci.edu/ml/datasets/Iris에 접속하여 내려받을 수 있다.

Sepal length	Sepal width	Petal length	Petal width	Species
5.2	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
7.0	3.2	4.7	1.4	*I. versicolor*
6.4	3.2	4.5	1.5	*I. versicolor*
6.9	3.1	4.9	1.5	*I. versicolor*
5.5	2.3	4.0	1.3	*I. versicolor*
6.3	3.3	6.0	2.5	*I. virginica*
5.8	2.7	5.1	1.9	*I. virginica*
7.1	3.0	5.9	2.1	*I. virginica*
6.3	2.9	5.6	1.8	*I. virginica*

Setosa

Versicolor

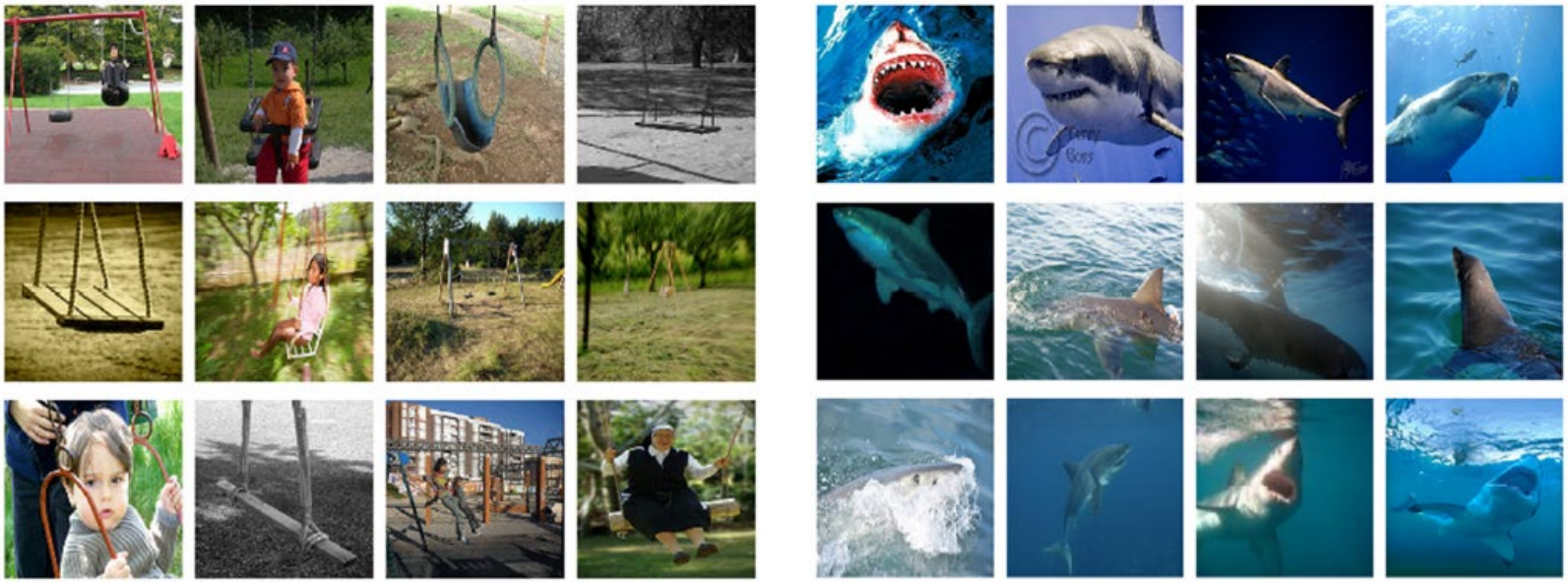
Virginica

데이터베이스의 중요성

- MNIST 데이터베이스는 미국표준국(NIST)에서 수집한 필기 숫자 데이터베이스로, 훈련집합 60,000자, 테스트집합 10,000자를 제공한다. http://yann.lecun.com/exdb/mnist에 접속하면 무료로 내려받을 수 있으며, 1988년부터 시작한 인식률 경쟁 기록도 볼 수 있다. 2017년 8월 기준으로는 [Ciresan2012] 논문이 0.23%의 오류율로 최고 자리를 차지하고 있다. 테스트집합에 있는 10,000개 샘플에서 단지 23개만 틀린 것이다.

데이터베이스의 중요성

- ImageNet 데이터베이스는 정보검색 분야에서 만든 WordNet의 단어 계층 분류를 그대로 따랐고, 부류마다 수백에서 수천 개의 영상을 수집하였다[Deng2009]. 총 21,841개 부류에 대해 총 14,197,122개의 영상을 보유하고 있다. 그중에서 1,000개 부류를 뽑아 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)라는 영상인식 경진대회를 2010년부터 매년 개최하고 있다. 대회 결과에 대한 자세한 내용은 4.4절을 참조하라. http:// image-net.org에서 내려받을 수 있다.



(a) 'swing' 부류 (b) 'Great white shark' 부류

그림 4-20 ImageNet의 예제 영상

데이터베이스 크기와 기계 학습 성능

- 데이터베이스의 왜소한 크기
  - 예) MNIST: 28*28 흑백 비트맵이라면 서로 다른 총 샘플 수는 2^{784}가지이지만, MNIST는 고작 6만 개 샘플

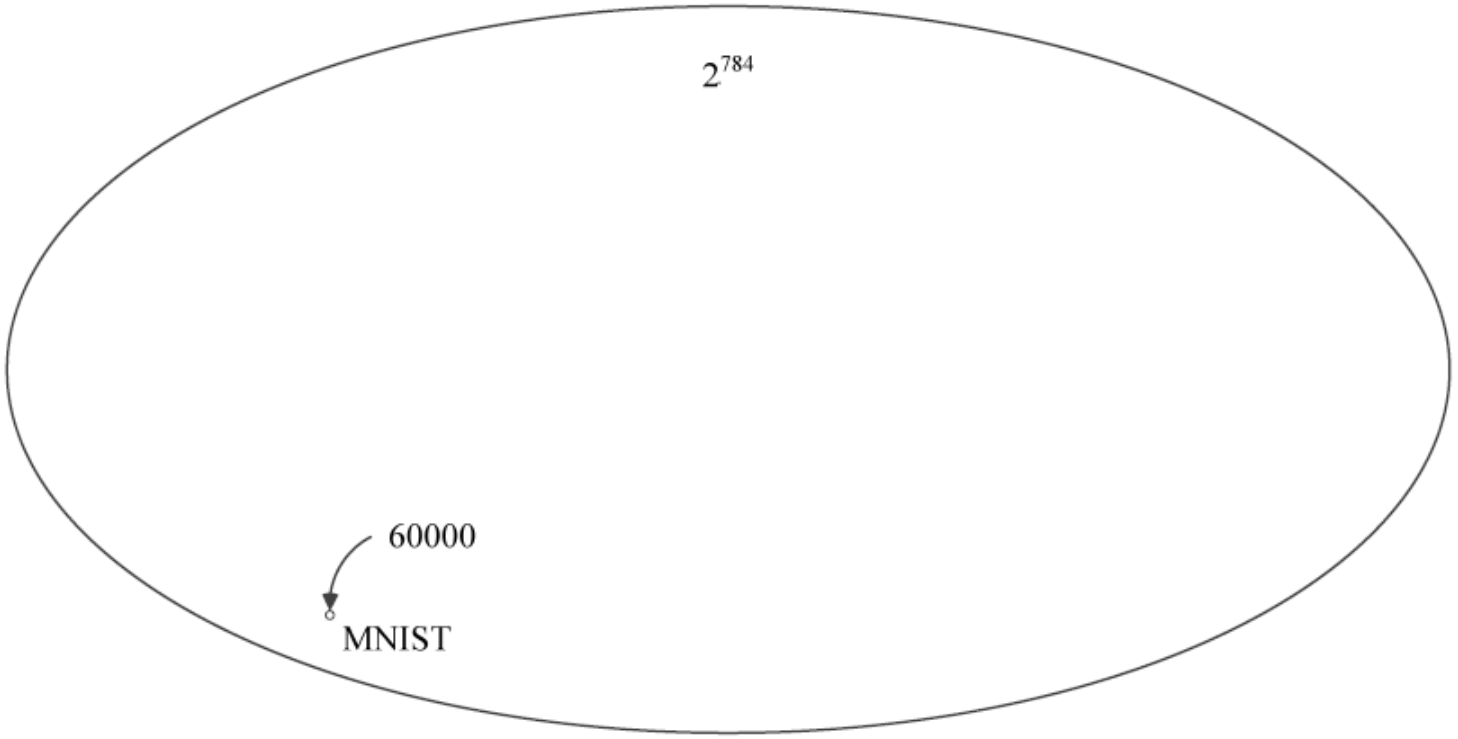
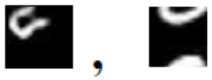


그림 1-9 방대한 특징 공간과 희소한 데이터베이스

데이터베이스 크기와 기계 학습 성능

- 왜소한 데이터베이스로 어떻게 높은 성능을 달성하는가?
  - 방대한 공간에서 실제 데이터가 발생하는 곳은 매우 작은 부분 공간임
    -  와 같은 샘플의 발생 확률은 거의 0
  - 매니폴드 가정
    -  와 같이 일정한 규칙에 따라 매끄럽게 변화

데이터 가시화

- 4차원 이상의 초공간은 한꺼번에 가시화 불가능
- 여러 가지 가시화 기법
  - 2개씩 조합하여 여러 개의 그래프 그림

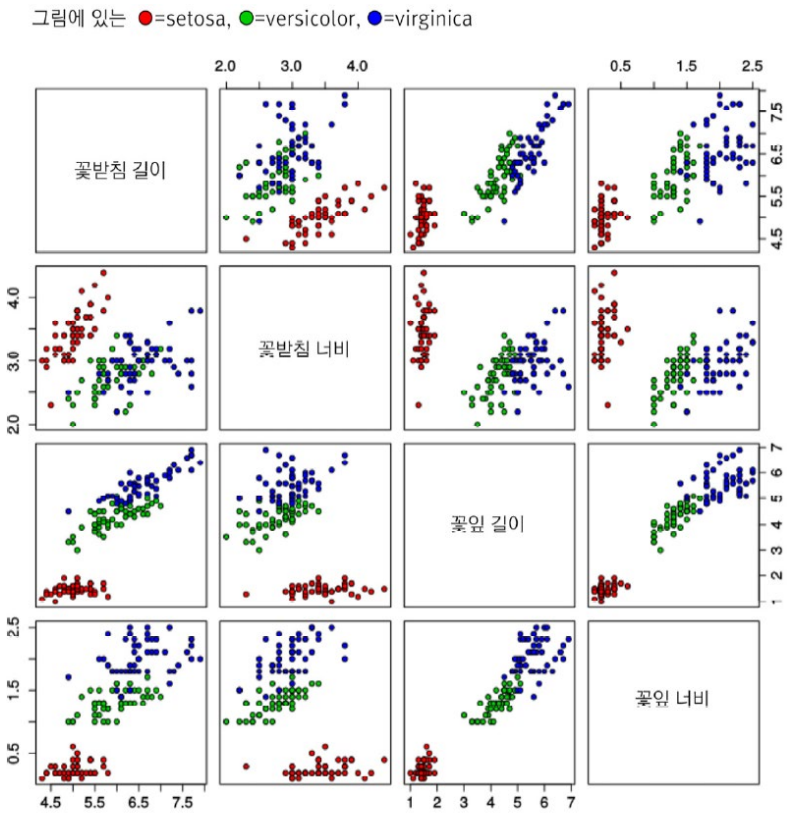


그림 1-10 고차원 특징 공간의 가시화

- 고차원 공간을 저차원으로 변환하는 기법들(주성분분석 6.6.1절)

간단한 기계 학습의 예

선형 회귀 문제

- [그림 1-4]: 식 (1.2)의 직선 모델을 사용하므로 두 개의 매개변수 $\Theta = (w, b)^{\mathrm{T}}$

$$y = wx + b \tag{1.2}$$

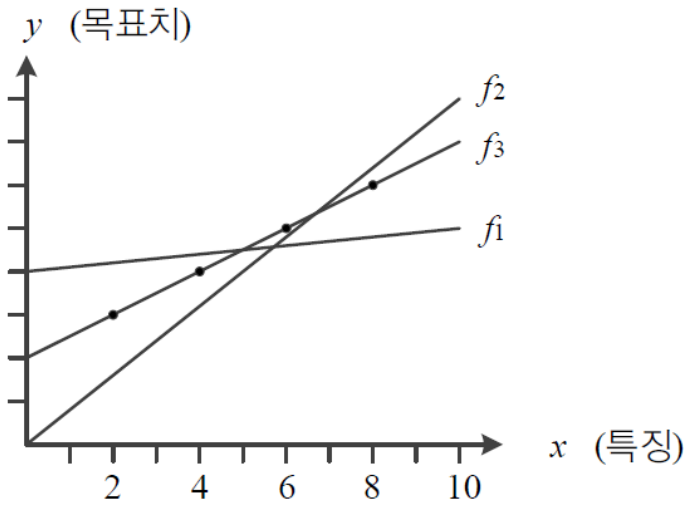


그림 1-4 간단한 기계 학습 예제

간단한 기계 학습의 예

- 목적 함수objective function (또는 비용 함수cost function)
  - 식 (1.8)은 선형 회귀를 위한 목적 함수
    - $f_\Theta(\mathbf{x}_i)$는 예측함수의 출력, y_i는 예측함수가 맞추어야 하는 목푯값이므로 $f_\Theta(\mathbf{x}_i)$-y_i는 오차
    - 식 (1.8)을 평균제곱오차MSE(mean squared error)라 부름

$$J(\Theta) = \frac{1}{n}\sum_{i=1}^{n}(f_\Theta(\mathbf{x}_i) - y_i)^2 \qquad (1.8)$$

  - 처음에는 최적 매개변수 값을 알 수 없으므로 난수로 $\Theta_1 = (w_1, b_1)^T$ 설정 → $\Theta_2 = (w_2, b_2)^T$로 개선 → $\Theta_3 = (w_3, b_3)^T$로 개선 → Θ_3는 최적해 $\hat{\Theta}$
    - 이때 $J(\Theta_1) > J(\Theta_2) > J(\Theta_3)$

간단한 기계 학습의 예

- [예제 1-1]
  - 훈련집합

$$\mathbb{X} = \{x_1 = (2.0), x_2 = (4.0), x_3 = (6.0), x_4 = (8.0)\},$$

$$\mathbb{Y} = \{y_1 = 3.0, y_2 = 4.0, y_3 = 5.0, y_4 = 6.0\}$$

  - 초기 직선의 매개변수 $\Theta_1 = (0.1, 4.0)^T$라 가정

$$\begin{aligned}
\mathbf{x}_1, y_1 &\rightarrow \left(f_{\Theta_1}(2.0) - 3.0\right)^2 = \left((0.1 * 2.0 + 4.0) - 3.0\right)^2 = 1.44 \\
\mathbf{x}_2, y_2 &\rightarrow \left(f_{\Theta_1}(4.0) - 4.0\right)^2 = \left((0.1 * 4.0 + 4.0) - 4.0\right)^2 = 0.16 \\
\mathbf{x}_3, y_3 &\rightarrow \left(f_{\Theta_1}(6.0) - 5.0\right)^2 = \left((0.1 * 6.0 + 4.0) - 5.0\right)^2 = 0.16 \\
\mathbf{x}_4, y_4 &\rightarrow \left(f_{\Theta_1}(8.0) - 6.0\right)^2 = \left((0.1 * 8.0 + 4.0) - 6.0\right)^2 = 1.44
\end{aligned}$$

$$\rightarrow \quad J(\Theta_1) = 0.8$$

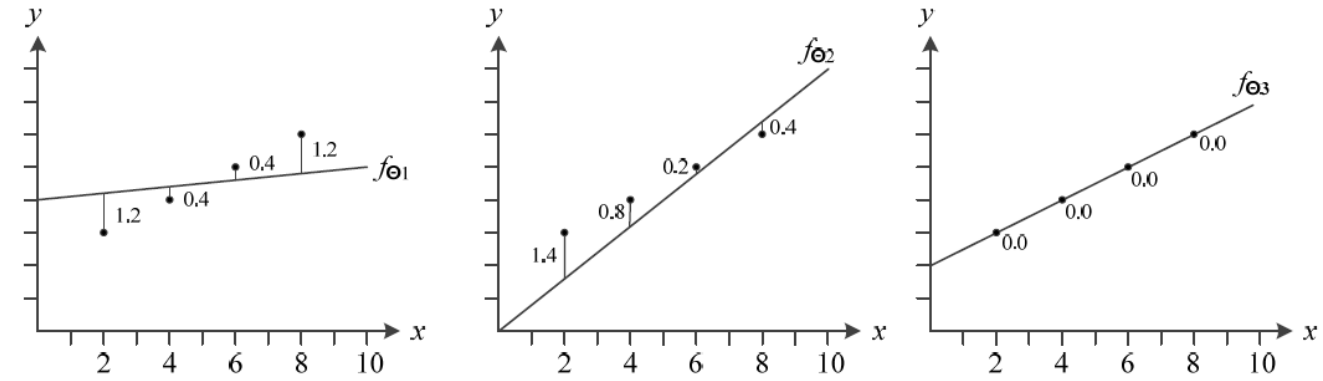
간단한 기계 학습의 예

- [예제 1-1] 훈련집합
  - Θ_1을 개선하여 $\Theta_2 = (0.8, 0.0)^T$가 되었다고 가정

$$\mathbf{x}_1, y_1 \rightarrow \left(f_{\Theta_2}(2.0) - 3.0\right)^2 = \left((0.8 * 2.0 + 0.0) - 3.0\right)^2 = 1.96$$
$$\mathbf{x}_2, y_2 \rightarrow \left(f_{\Theta_2}(4.0) - 4.0\right)^2 = \left((0.8 * 4.0 + 0.0) - 4.0\right)^2 = 0.64$$
$$\mathbf{x}_3, y_3 \rightarrow \left(f_{\Theta_2}(6.0) - 5.0\right)^2 = \left((0.8 * 6.0 + 0.0) - 5.0\right)^2 = 0.04$$
$$\mathbf{x}_4, y_4 \rightarrow \left(f_{\Theta_2}(8.0) - 6.0\right)^2 = \left((0.8 * 8.0 + 0.0) - 6.0\right)^2 = 0.16$$

$$\longrightarrow J(\Theta_2) = 0.7$$

  - Θ_2를 개선하여 $\Theta_3 = (0.5, 2.0)^T$가 되었다고 가정
  - 이때 $J(\Theta_3) = 0.0$이 되어 Θ_3은 최적값 $\hat{\Theta}$ 이 됨



(a) 초기 매개변수 Θ_1 (b) Θ_1을 개선하여 Θ_2가 됨 (c) Θ_2를 개선하여 최적의 Θ_3을 찾음

그림 1-11 기계 학습에서 목적함수의 역할

간단한 기계 학습의 예

- 기계 학습이 할 일을 공식화하면,

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} J(\Theta) \qquad (1.9)$$

  - 기계 학습은 작은 개선을 반복하여 최적해를 찾아가는 수치적 방법으로 식 (1.9)를 품

- 알고리즘 형식으로 쓰면,

알고리즘 1–1 기계 학습 알고리즘

입력 : 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$
출력 : 최적의 매개변수 $\hat{\Theta}$

```
1  난수를 생성하여 초기 해 Θ1을 설정한다.
2  t=1
3  while (J(Θt)가 0.0에 충분히 가깝지 않음)     // 수렴 여부 검사
4      J(Θt)가 작아지는 방향 ΔΘt를 구한다.     // ΔΘt는 주로 미분을 사용하여 구함
5      Θt+1 = Θt + ΔΘt
6      t=t+1
7  Θ̂ = Θt
```

간단한 기계 학습의 예

- 좀 더 현실적인 상황
  - 지금까지는 데이터가 선형을 이루는 아주 단순한 상황을 고려함
  - 실제 세계는 선형이 아니며 잡음이 섞임 → 비선형 모델이 필요

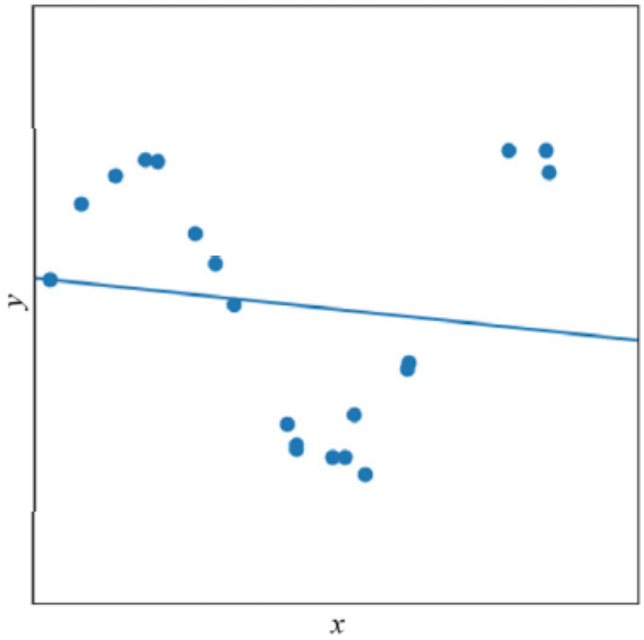


그림 1-12 선형 모델의 한계

모델 선택

- 과소적합과 과잉적합
- 바이어스와 분산
- 검증집합과 교차검증을 이용한 모델 선택 알고리즘
- 모델 선택의 한계와 현실적인 해결책

과소적합과 과잉적합

- [그림 1.13]의 1차 모델은 과소적합underfitting
  - 모델의 '용량이 작아' 오차가 클 수밖에 없는 현상
- 비선형 모델을 사용하는 대안
  - [그림 1-13]의 2차, 3차, 4차, 12차는 다항식 곡선을 선택한 예
  - 1차(선형)에 비해 오차가 크게 감소함

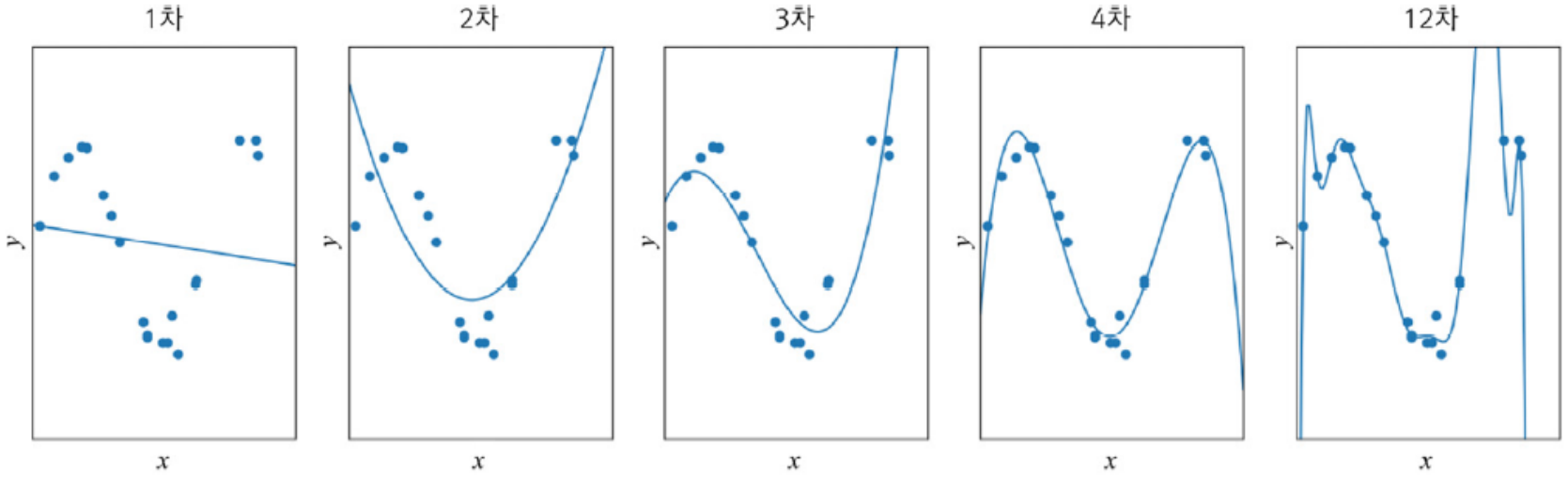


그림 1-13 과소적합과 과잉적합 현상

과소적합과 과잉적합

과잉적합overfitting

- 12차 다항식 곡선을 채택한다면 훈련집합에 대해 거의 완벽하게 근사화함
- 하지만 '새로운' 데이터를 예측한다면 큰 문제 발생
  - x_0에서 빨간 막대 근방을 예측해야 하지만 빨간 점을 예측
- 이유는 '용량이 크기' 때문. 학습 과정에서 잡음까지 수용 → 과잉적합 현상
- 적절한 용량의 모델을 선택하는 모델 선택 작업이 필요함

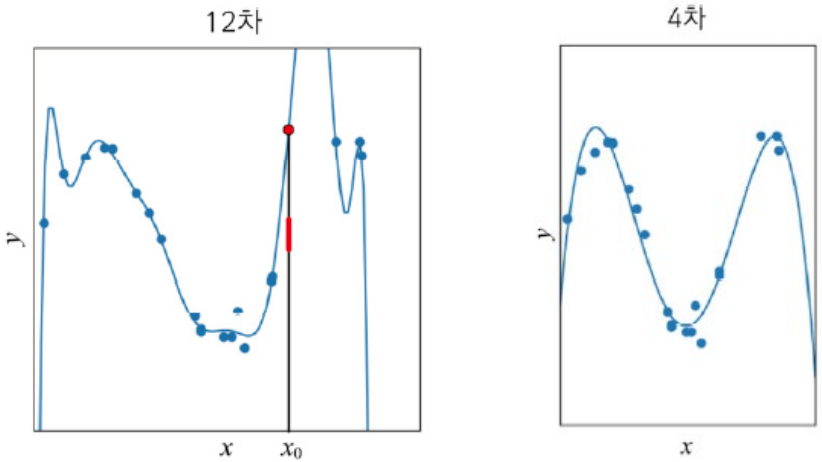


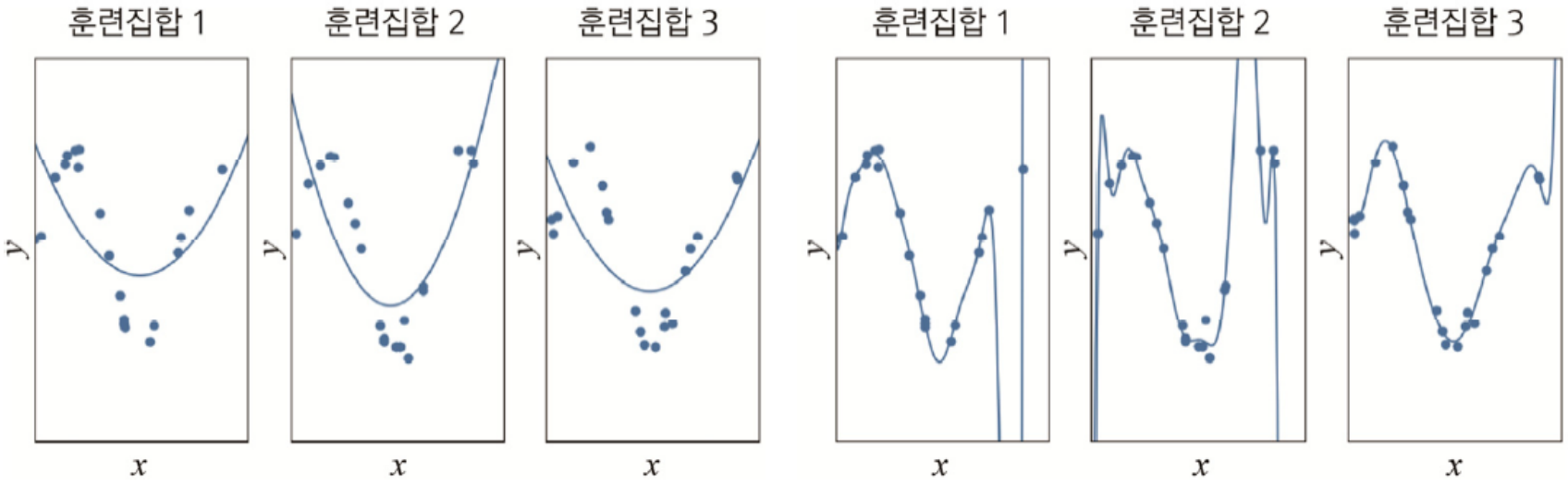
그림 1-14 과잉적합되었을 때 부정확한 예측 현상

바이어스와 분산

- 1차~12차 다항식 모델의 비교 관찰
  - 1~2차는 훈련집합과 테스트집합 모두 낮은 성능
  - 12차는 훈련집합에 높은 성능을 보이나 테스트집합에서는 낮은 성능 → 낮은 일반화 능력
  - 3~4차는 훈련집합에 대해 12차보다 낮겠지만 테스트집합에는 높은 성능 → 높은 일반화 능력

바이어스와 분산

- 훈련집합을 여러 번 수집하여 1차~12차에 적용하는 실험
  - 2차는 매번 큰 오차 → 바이어스가 큼. 하지만 비슷한 모델을 얻음 → 낮은 분산
  - 12차는 매번 작은 오차 → 바이어스가 작음. 하지만 크게 다른 모델을 얻음 → 높은 분산
  - 일반적으로 용량이 작은 모델은 바이어스는 크고 분산은 작음. 복잡한 모델은 바이어스는 작고 분산은 큼
  - 바이어스와 분산은 트레이드오프 관계



(a) 2차 모델(바이어스는 크고, 분산은 작음)

(b) 12차 모델(바이어스는 작고, 분산은 큼)

그림 1-15 모델의 바이어스와 분산 특성

바이어스와 분산

기계 학습의 목표

- 낮은 바이어스와 낮은 분산을 가진 예측기 제작이 목표. 즉 왼쪽 아래 상황

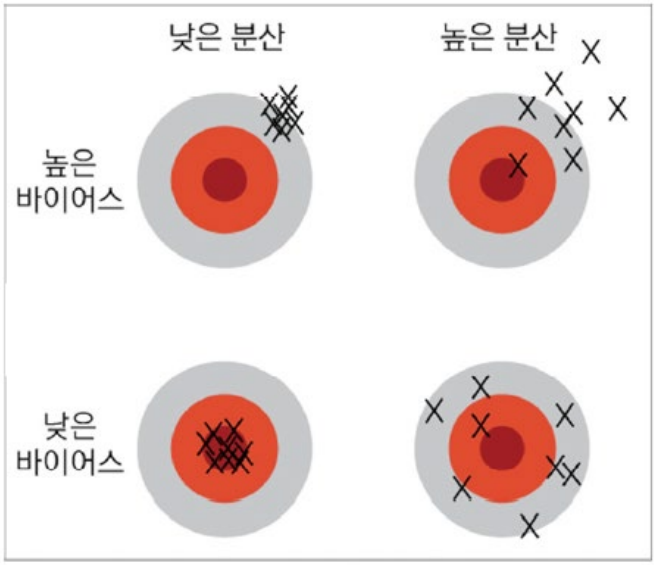


그림 1-16 바이어스와 분산

- 하지만 바이어스와 분산은 트레이드오프 관계
- 따라서 바이어스 희생을 최소로 유지하며 분산을 최대로 낮추는 전략 필요

검증집합과 교차검증을 이용한 모델 선택 알고리즘

검증집합을 이용한 모델 선택

- 훈련집합과 테스트집합과 다른 별도의 검증집합을 가진 상황

알고리즘 1-2 검증집합을 이용한 모델 선택

입력 : 모델집합 Ω, 훈련집합, 검증집합, 테스트집합
출력 : 최적 모델과 성능

```
1  for (Ω에 있는 각각의 모델)
2      모델을 훈련집합으로 학습시킨다.
3      검증집합으로 학습된 모델의 성능을 측정한다.  // 검증 성능 측정
4  가장 높은 성능을 보인 모델을 선택한다.
5  테스트집합으로 선택된 모델의 성능을 측정한다.
```

검증집합과 교차검증을 이용한 모델 선택 알고리즘

교차검증cross validation

- 비용 문제로 별도의 검증집합이 없는 상황에 유용한 모델 선택 기법
- 훈련집합을 등분하여, 학습과 평가 과정을 여러 번 반복한 후 평균 사용

알고리즘 1-3 교차검증에 의한 모델 선택

입력 : 모델집합 Ω, 훈련집합, 테스트집합, 그룹 개수 k
출력 : 최적 모델과 성능

```
1  훈련집합을 k개의 그룹으로 등분한다.
2  for (Ω에 있는 각각의 모델)
3        for (i=1 to k)
4              i번째 그룹을 제외한 k-1개 그룹으로 모델을 학습시킨다.
5              학습된 모델의 성능을 i번째 그룹으로 측정한다.
6        k개 성능을 평균하여 해당 모델의 성능으로 취한다.
7  가장 높은 성능을 보인 모델을 선택한다.
8  테스트집합으로 선택된 모델의 성능을 측정한다.
```

검증집합과 교차검증을 이용한 모델 선택 알고리즘

부트스트랩boot strap

- 난수를 이용한 샘플링 반복

알고리즘 1-4 부트스트랩을 이용한 모델 선택

입력 : 모델집합 Ω, 훈련집합, 테스트집합, 샘플링 비율 $p(0<p\leq 1)$, 반복횟수 T
출력 : 최적 모델과 성능

```
1  for (Ω에 있는 각각의 모델)
2      for (i=1 to T)
3          훈련집합 𝕏에서 pn개 샘플을 뽑아 새로운 훈련집합 𝕏′를 구성한다. 이때 대치를 허용한다.
4          𝕏′로 모델을 학습시킨다.
5          𝕏 − 𝕏′를 이용하여 학습된 모델의 성능을 측정한다.
6      T개 성능을 평균하여 해당 모델의 성능으로 취한다.
7  가장 높은 성능을 보인 모델을 선택한다.
8  테스트집합으로 선택된 모델의 성능을 측정한다.
```

모델 선택의 한계와 현실적인 해결책

- [알고리즘 1-2, 1-3, 1-4]에서 모델 집합 Ω
  - [그림 1-13]에서는 서로 다른 차수의 다항식이 Ω인 셈
  - 현실에서는 아주 다양
    - 신경망(3,4,8장), 강화 학습(9장), 확률 그래피컬 모델(10장), SVM(11장), 트리 분류기 (12장) 등이 선택 대상
    - 신경망을 채택하더라도 MLP(3장), 깊은 MLP(4장), CNN(4장) 등 아주 많음
- 현실에서는 경험으로 큰 틀 선택한 후
  - 모델 선택 알고리즘으로 세부 모델 선택하는 전략 사용
  - 예) CNN을 사용하기로 정한 후, 은닉층 개수, 활성함수, 모멘텀 계수 등을 정하는데 모델 선택 알고리즘을 적용함

모델 선택의 한계와 현실적인 해결책

- 이런 경험적인 접근방법에 대한 『Deep Learning』 책의 비유

> "To some extent, we are always trying to fit a square peg(the data generating process) into a round hole(our model family). 어느 정도 우리가 하는 일은 항상 둥근 홈(우리가 선택한 모델)에 네모 막대기(데이터 생성 과정)를 끼워 넣는 것이라고 말할 수 있다[Goodfellow2016(222쪽)]."

- 현대 기계 학습의 전략
  - 용량이 충분히 큰 모델을 선택 한 후, 선택한 모델이 정상을 벗어나지 않도록 여러 가지 규제regularization 기법을 적용함
  - 예) [그림 1-13]의 경우 12차 다항식을 선택한 후 적절히 규제를 적용

기계 학습 유형

- 지도 방식에 따른 유형
- 다양한 기준에 따른 유형

지도 방식에 따른 유형

- 지도 학습
  - 특징 벡터 $\mathbb{X}$와 목푯값 $\mathbb{Y}$가 모두 주어진 상황
  - 회귀와 분류 문제로 구분
- 비지도 학습
  - 특징 벡터 $\mathbb{X}$는 주어지는데 목푯값 $\mathbb{Y}$ 가 주어지지 않는 상황
  - 군집화 과업 (고객 성향에 따른 맞춤 홍보 응용 등)
  - 밀도 추정, 특징 공간 변환 과업

지도 방식에 따른 유형

강화 학습

- 목푯값이 주어지는데, 지도 학습과 다른 형태임
- 예) 바둑
  - 수를 두는 행위가 샘플인데, 게임이 끝나면 목푯값 하나가 부여됨
    - 이기면 1, 패하면 -1을 부여
  - 게임을 구성한 샘플들 각각에 목푯값을 나누어 주어야 함
- 9장의 주제

준지도 학습

- 일부는 $\mathbb{X}$와 $\mathbb{Y}$를 모두 가지지만, 나머지는 $\mathbb{X}$만 가진 상황
- 인터넷 덕분으로 $\mathbb{X}$의 수집은 쉽지만, $\mathbb{Y}$는 수작업이 필요하여 최근 중요성 부각
- 7장의 주제

다양한 기준에 따른 유형

오프라인 학습과 온라인 학습

- 이 책은 오프라인 학습을 다룸
- 온라인 학습은 인터넷 등에서 추가로 발생하는 샘플을 가지고 점증적 학습

결정론적 학습과 스토캐스틱 학습

- 결정론적에서는 같은 데이터를 가지고 다시 학습하면 같은 예측기가 만들어짐
- 스토캐스틱 학습은 학습 과정에서 난수를 사용하므로 같은 데이터로 다시 학습하면 다른 예측기가 만들어짐. 보통 예측 과정도 난수 사용
- 10.4절의 RBM과 DBN이 스토캐스틱 학습

분별 모델과 생성 모델

- 분별 모델은 부류 예측에만 관심. 즉 $P(y|\mathbf{x})$의 추정에 관심
- 생성 모델은 $P(\mathbf{x})$ 또는 $P(\mathbf{x}|y)$를 추정함
  - 따라서 새로운 샘플을 '생성'할 수 있음
  - 4.5절의 GAN, 10.4절의 RBM은 생성 모델
  - 8.5절의 순환신경망(RNN)을 생성 모델로 활용하는 응용 예제