기계 학습
-Machine Learning-

20221291 조영서

순천향대학교 AI·빅데이터학과

pandajr@sch.ac.kr

목차

Chapter1.1 기계 학습이란

Chapter1.1 – 기계 학습이란

- 학습이란? <표준국어대사전>

  "경험의 결과로 나타나는, 비교적 지속적인 행동의 변화나 그 잠재력의 변화. 또는 지식을 습득하는 과정[국립국어원2017]"

- 기계 학습이란?
  - 인공지능 초창기 사무엘의 정의

  "Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort. 컴퓨터가 경험을 통해 학습할 수 있도록 프로그래밍할 수 있다면, 세세하게 프로그래밍해야 하는 번거로움에서 벗어날 수 있다[Samuel1959]."

Chapter1.1 – 기계 학습이란

- 기계 학습이란?
  - 현대적 정의

"A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.

어떤 컴퓨터 프로그램이 *T*라는 작업을 수행한다. 이 프로그램의 성능을 *P*라는 척도로 평가했을 때 경험 *E*를 통해 성능이 개선된다면 이 프로그램은 학습을 한다고 말할 수 있다[Mitchell1997(2쪽)]."

"Programming computers to optimize a performance criterion using example data or past experience
사례 데이터, 즉 과거 경험을 이용하여 성능 기준을 최적화하도록 프로그래밍하는 작업[Alpaydin2010]"

"Computational methods using experience to improve performance or to make accurate predictions
성능을 개선하거나 정확하게 예측하기 위해 경험을 이용하는 계산학 방법들[Mohri2012]"

Chapter1.1 – 기계 학습이란

- 인공지능의 탄생
    - 컴퓨터의 뛰어난 능력
        - 사람이 어려워하는 일을 아주 쉽게 함
        - 80932.46789076*0.39001324와 같은 곱셈을 고속으로 수행(현재는 초당 수십억개)
        - 복잡한 함수의 미분과 적분 척척
    - 컴퓨터에 대한 기대감 (컴퓨터의 능력 과신)
        - 사람이 쉽게 하는 일, 예를 들어 고양이/개 구별하는 일도 잘 하지 않을까
        - 1950년대에 인공지능이라는 분야 등장
- 초창기는 지식기반 방식이 주류
    - 예) “구멍이 2개이고 중간 부분이 홀쭉하며, 맨 위와 아래가 둥근 모양이라면 8이다”

Chapter1.1 – 기계 학습이란

- 큰 깨달음
  - 지식기반의 한계
  - 단추를 “가운데 구멍이 몇 개 있는 물체”라고 규정하면 많은 오류 발생

그림 1–2 인식 시스템이 대처해야 하는 심한 변화 양상(8과 단추라는 패턴을 어떻게 기술할 것인가?)

  - 사람은 변화가 심한 장면을 아주 쉽게 인식하지만, 왜 그렇게 인식하는지 서술하지는 못함

Chapter1.1 – 기계 학습이란

- 인공지능의 주도권 전환
  - 지식기반 → 기계 학습
  - 기계 학습: 데이터 중심 접근방식



그림 1-3 기계 학습으로 만든 최첨단 인공지능 제품들

Chapter1.1 – 기계 학습이란

- 간단한 기계 학습 예제
  - 가로축은 시간, 세로축은 이동체의 위치
  - 관측한 4개의 점이 데이터

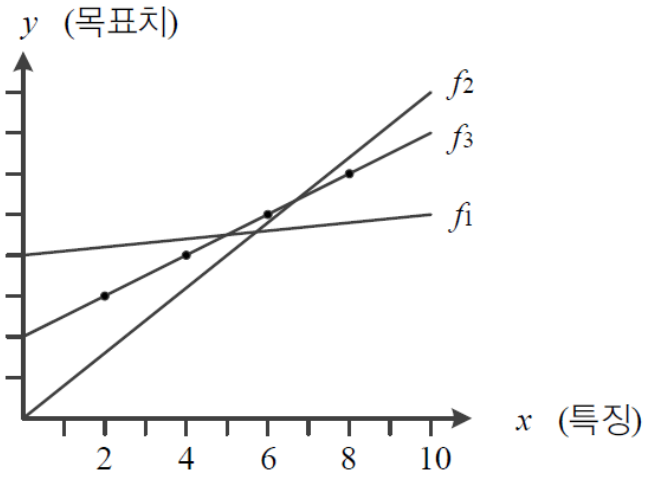


그림 1–4 간단한 기계 학습 예제

- 예측prediction 문제
  - 임의의 시간이 주어지면 이때 이동체의 위치는?
  - 회귀regression 문제와 분류classification 문제로 나뉨
    - 회귀는 목표치가 실수형(연속형), 분류는 범주형(이산형) ([그림 1-4]는 회귀 문제)

Chapter1.1 – 기계 학습이란

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

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

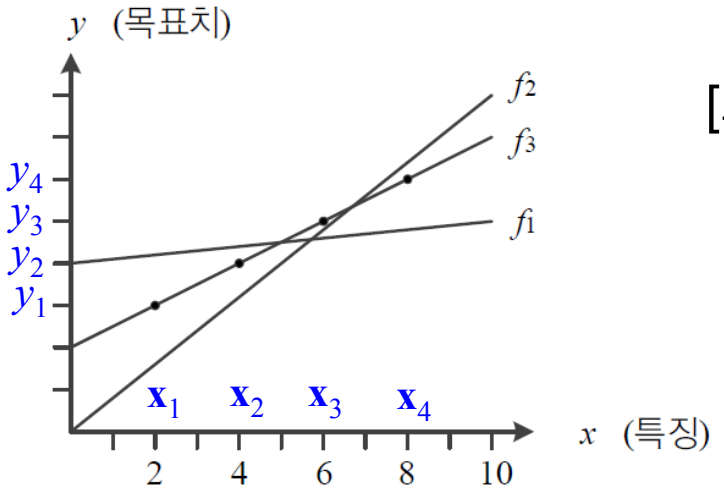


그림 1-4 간단한 기계 학습 예제

[그림 1-4] 예제의 훈련집합

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

Chapter1.1 – 기계 학습이란

- 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 $w = 0.5$와 $b = 2.0$

Chapter1.1 – 기계 학습이란

- 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

Chapter1.1 – 기계 학습이란

- 기계 학습: 가장 정확하게 예측할 수 있는 최적의 매개변수 값을 찾는 작업
- 테스트: 훈련 집합에 없는 ‘새로운‘ 샘플에 대한 목푯값을 예측하는 과정
- 일반화: 테스트 집합에 대해 높은 성능을 가진 성질

Chapter1.1 – 기계 학습이란

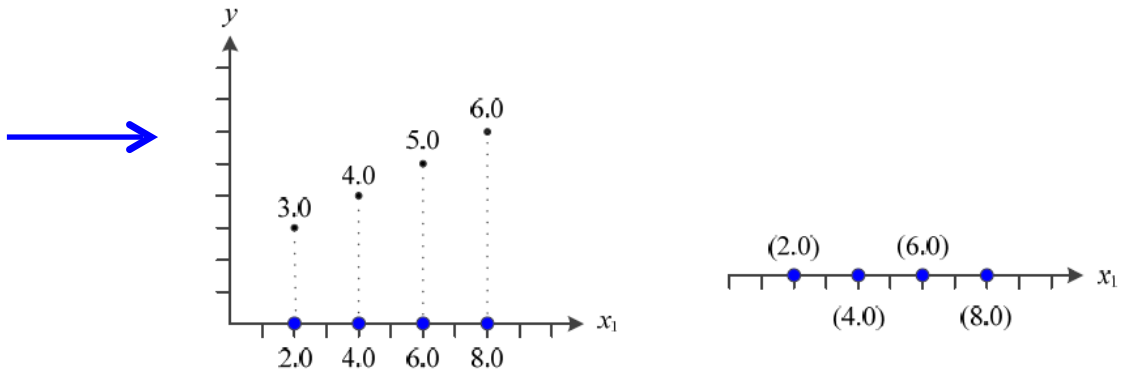
표 1-1 사람의 학습과 기계 학습의 비교

기준	사람의 학습	기계 학습
학습 과정	능동적	수동적
데이터 형식	자연에 존재하는 그대로	일정한 형식에 맞추어 사람이 준비함
동시에 학습 가능한 과업 수	자연스럽게 여러 과업을 학습	하나의 과업만 가능
학습 원리에 대한 지식	매우 제한적으로 알려져 있음	모든 과정이 밝혀져 있음
수학 의존도	매우 낮음	매우 높음
성능 평가	경우에 따라 객관적이거나 주관적	객관적(수치로 평가, 예를 들어 정확률 99.8%)
역사	수백만 년	60년 가량

Chapter1.2 특징 공간에 대한 이해

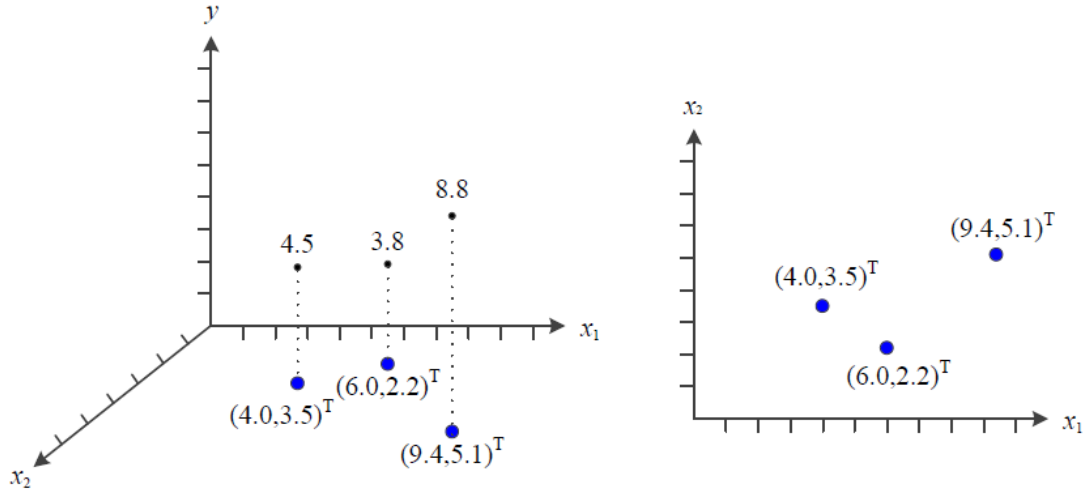
Chapter1.2 – 특징 공간에 대한 이해

- 1차원 특징 공간



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

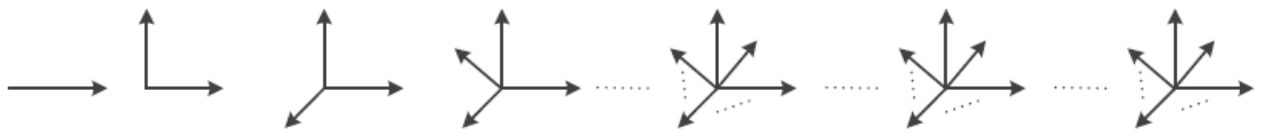
- 2차원 특징 공간
  - 특징 벡터 표기
    - $\mathbf{x} = (x_1, x_2)^T$
  - 예시
    - $\mathbf{x}$ = (몸무게, 키)^T, y = 장타율
    - $\mathbf{x}$ = (체온, 두통)^T, y = 감기 여부



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

그림 1-5 특징 공간과 데이터의 표현

Chapter1.2 – 특징 공간에 대한 이해



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

Chapter1.2 – 특징 공간에 대한 이해

- 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 = $d + 1$

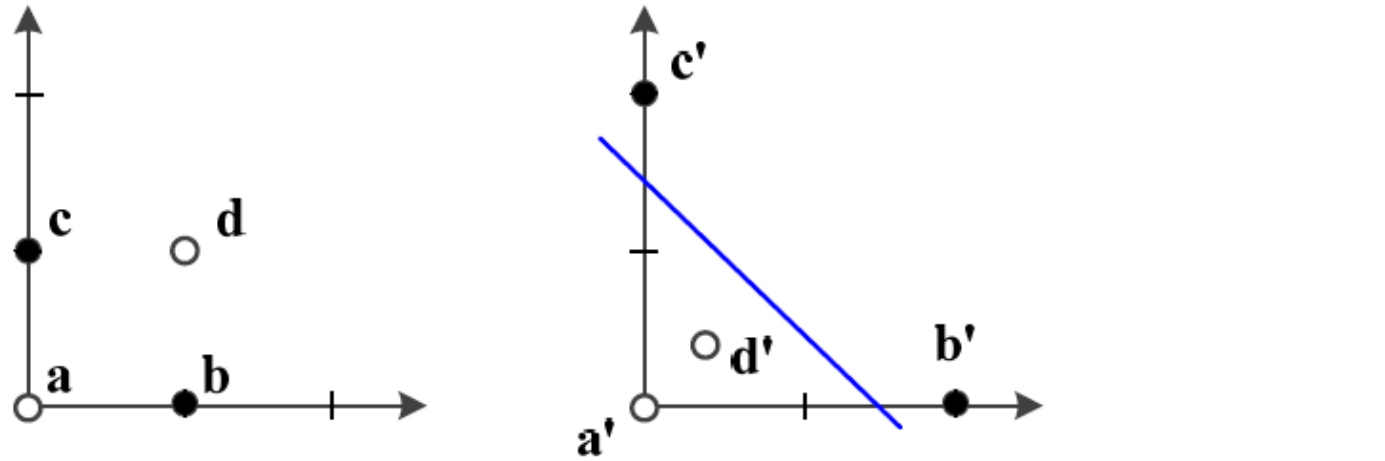
$$y = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b \tag{1.3}$$

  - 2차 곡선 모델을 사용하면 매개변수 수가 크게 증가
    - 매개변수 수 = $d^2 + d + 1$
    - 예) Iris 데이터: d = 4이므로 21개의 매개변수
    - 예) MNIST 데이터: d=784이므로 615,441개의 매개변수

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

Chapter1.2 – 특징 공간에 대한 이해

- 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

Chapter1.2 – 특징 공간에 대한 이해

- 식 (1.6)으로 변환된 새로운 특징 공간 ([그림 1-7(b)])
  - 직선 모델로 100% 정확률

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

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

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

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

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

- 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

Chapter1.2 – 특징 공간에 대한 이해

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

Chapter1.3 데이터에 대한 이해

Chapter1.3 – 데이터에 대한 이해

- 과학 기술의 발전 과정

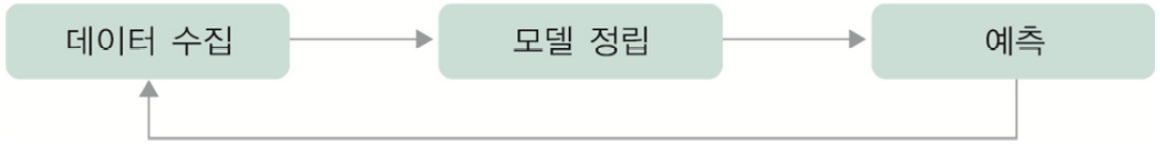


그림 1-8 과학기술의 발전 과정

  - 예) 티코 브라헤는 천동설이라는 틀린 모델을 선택함으로써 자신이 수집한 데이터를 설명하지 못함. 케플러는 지동설 모델을 도입하여 제 1, 제 2, 제 3법칙을 완성함

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

Chapter1.3 – 데이터에 대한 이해

- 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
- 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

Chapter1.3 – 데이터에 대한 이해

- 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)

Chapter1.3 – 데이터에 대한 이해

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

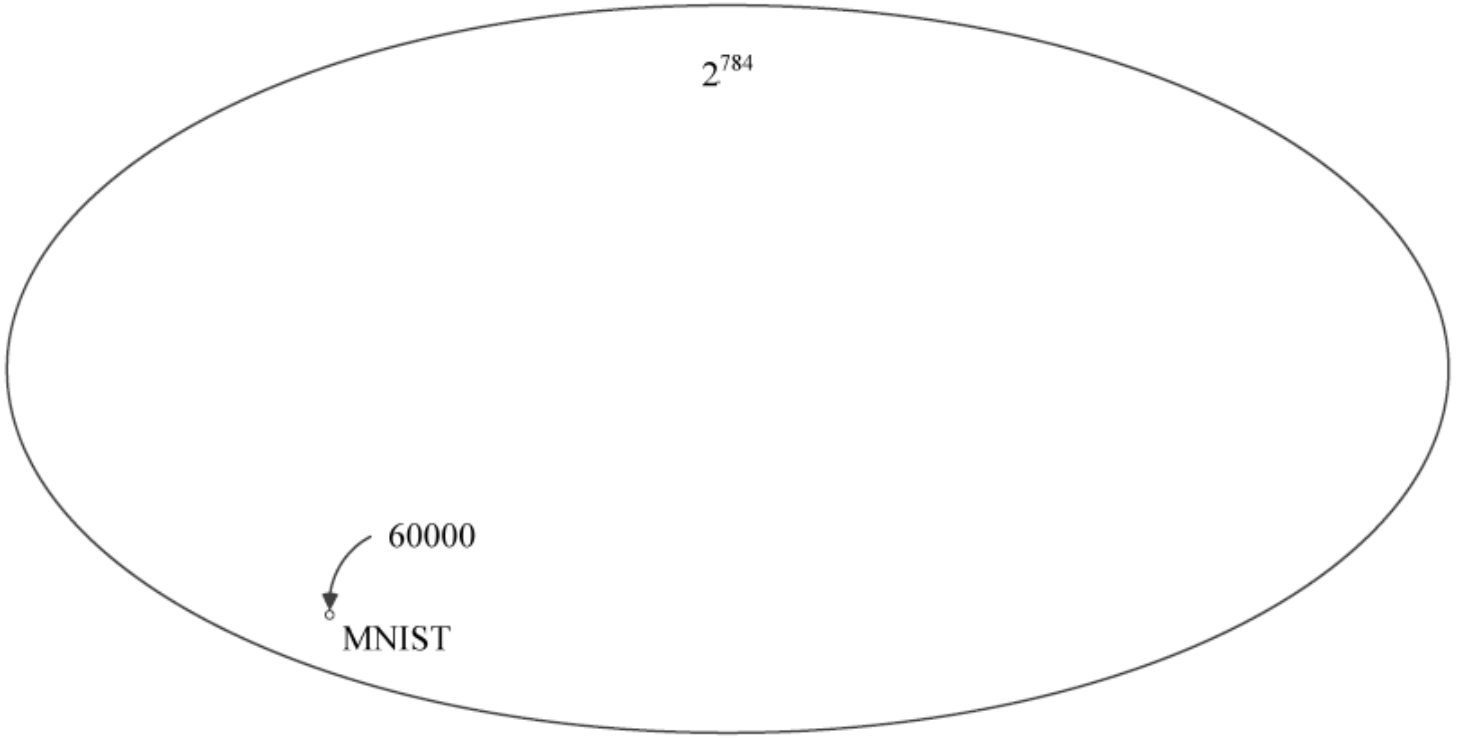


그림 1–9 방대한 특징 공간과 희소한 데이터베이스

Chapter1.3 – 데이터에 대한 이해

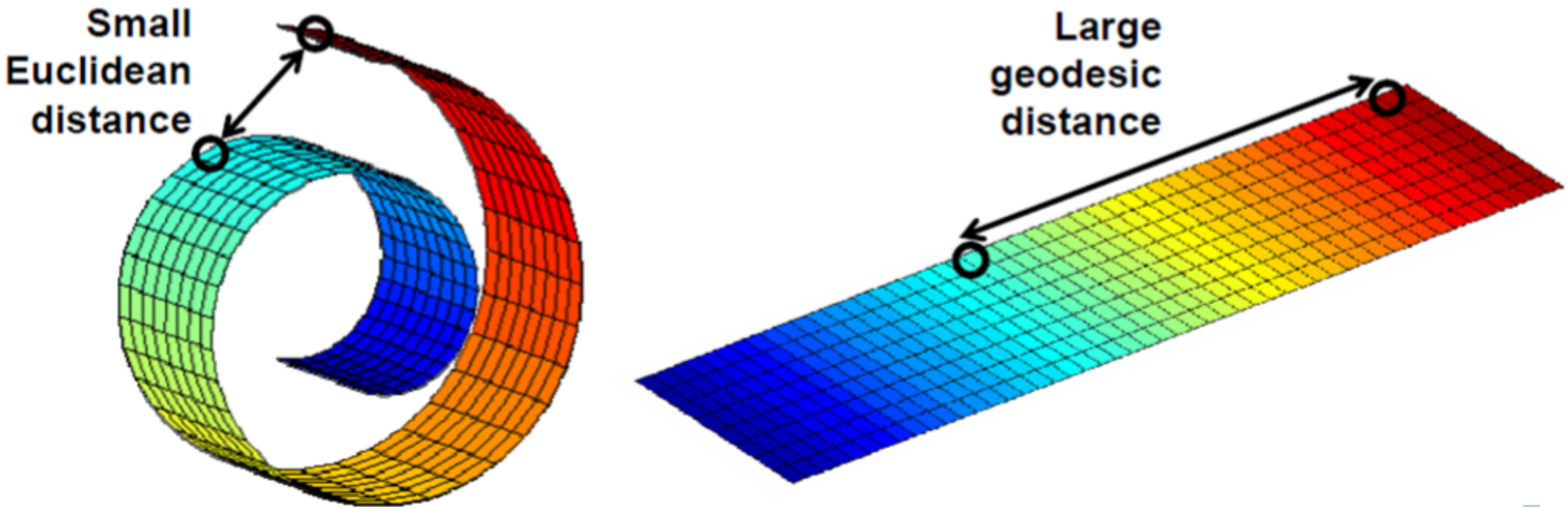
- 왜소한 데이터베이스로 어떻게 높은 성능을 달성하는가?
  - 방대한 공간에서 실제 데이터가 발생하는 곳은 매우 작은 부분 공간임
    -  같은 샘플의 발생 확률은 거의 0
  - 매니폴드 가정
    -  이 일정한 규칙에 따라 매끄럽게 변화

Chapter1.3 – 데이터에 대한 이해

- 매니폴드(manifold): 데이터가 있는 공간

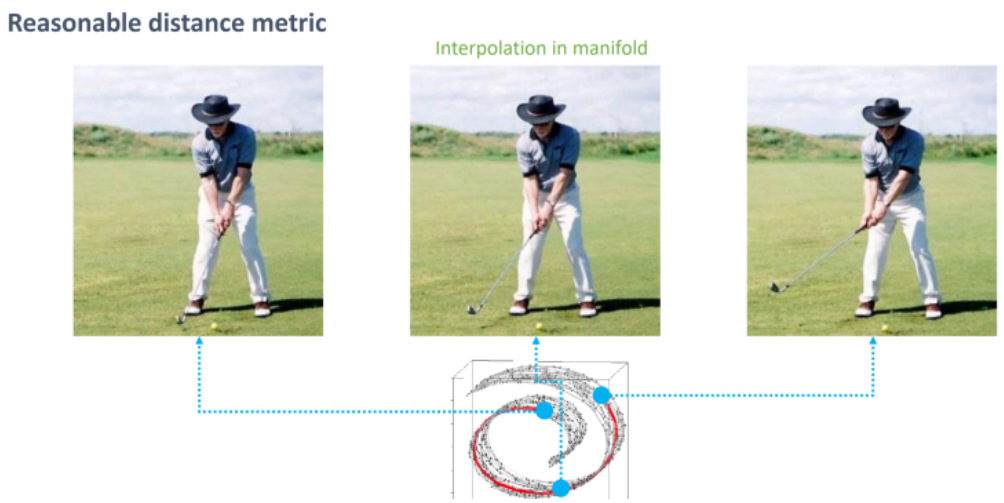
- 매니폴드 가정(manifold hypothesis): 데이터가 고차원이라도 이 집합을 포함하는 저차원의 매니폴드가 있다. 즉, 데이터가 고차원이라도 저차원의 매니폴드 상에 위치할 수 있으며, 낮은 차원의 매니폴드를 벗어나면 밀도가 작아진다

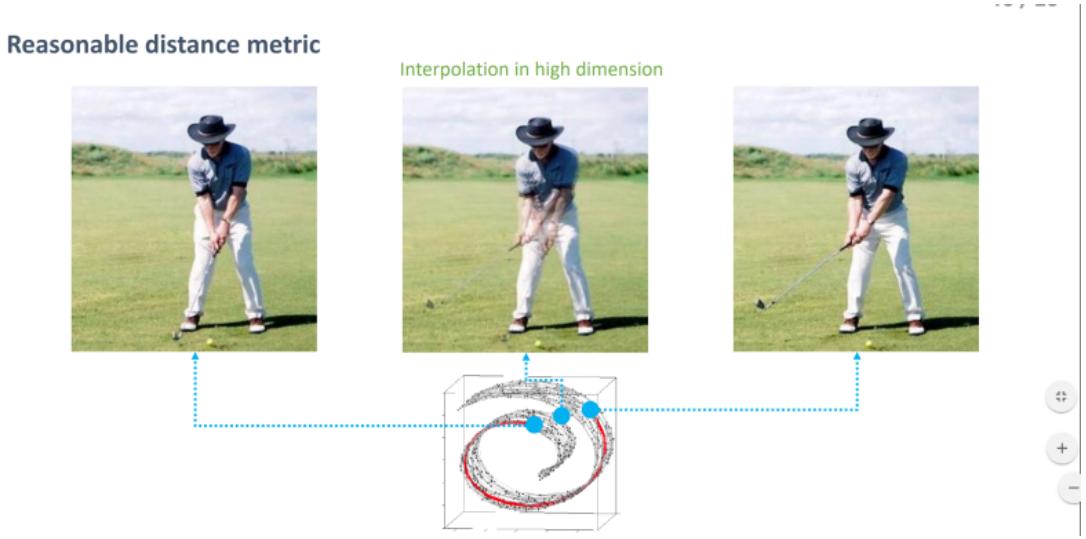
Chapter1.3 – 데이터에 대한 이해



출처: https://roytravel.tistory.com/105

Chapter1.3 – 데이터에 대한 이해





출처: https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=coniecual&logNo=221417921857

Chapter1.3 – 데이터에 대한 이해

- 4차원 이상의 초공간은 한꺼번에 가시화 불가능

- 여러 가지 가시화 기법
  - 2개씩 조합하여 여러 개의 그래프 그림

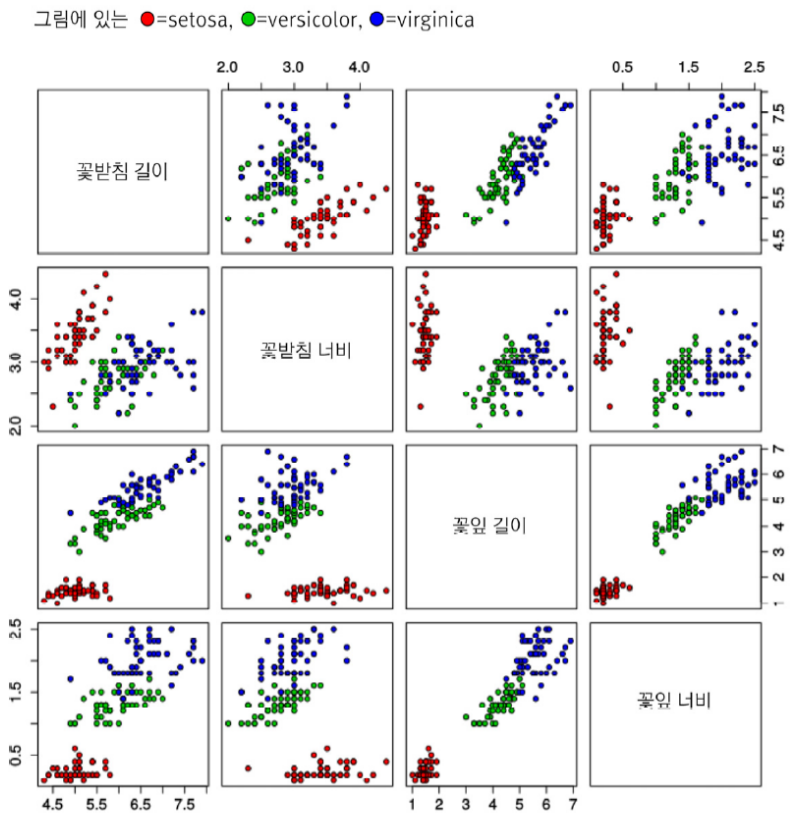


그림 1–10 고차원 특징 공간의 가시화

Chapter1.4 간단한 기계 학습의 예

Chapter1.4 – 간단한 기계 학습의 예

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

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

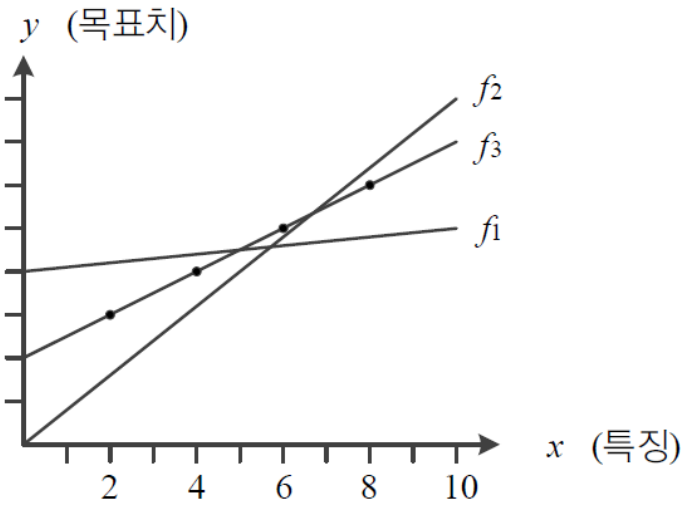


그림 1–4 간단한 기계 학습 예제

Chapter1.4 – 간단한 기계 학습의 예

- 목적 함수objective function 또는 비용함수 (cost 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 $\Theta_1 = (w_1,\ b_1)^{\mathrm{T}}$ 설정 → $\Theta_2 = (w_2, b_2)^{\mathrm{T}}$ 로 개선 → $\Theta_3 = (w_3, b_3)^{\mathrm{T}}$ 로 개선 → Θ_3는 최적해 $\hat{\Theta}$
    - 이때 $J(\Theta_1) > J(\Theta_2) > J(\Theta_3)$

Chapter1.4 – 간단한 기계 학습의 예

- 훈련집합

$$\mathbb{X} = \{x_1 = (2.0),\ \ x_2 = (4.0),\ x_3 = \left(6.0\right),\ x_4 = (8.0)\},$$
$$\mathbb{Y} = \{y_1 = (3.0),\ y_2 = (4.0),\ y_3 = (5.0),\ y_4 = (6.0)\}$$

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

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

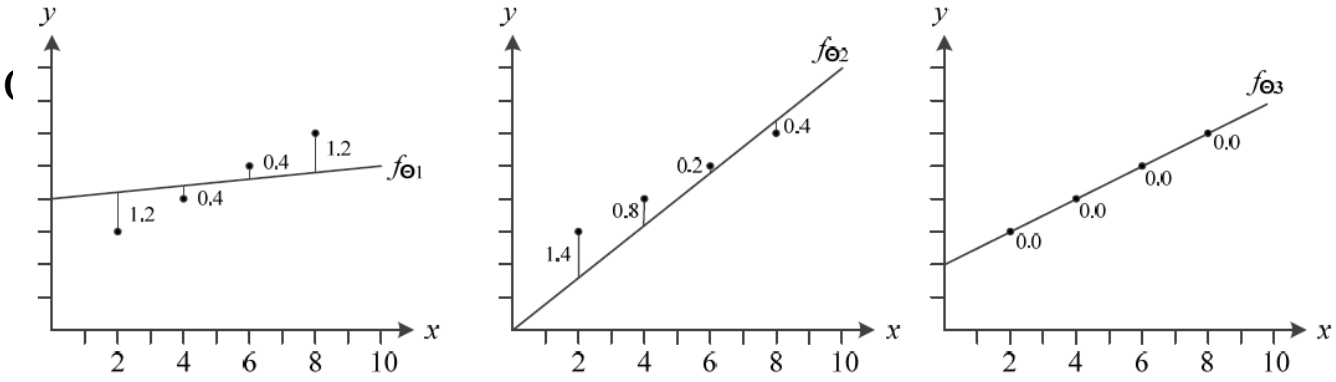
$$J\left(\Theta_1\right) = 0.8$$

Chapter1.4 – 간단한 기계 학습의 예

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

$$
\begin{aligned}
\mathbf{x}_1, y_1 &\rightarrow \left(f_{\Theta_2}(2.0) - 3.0\right)^2 = \left((0.8 * 2.0 + 0.0) - 3.0\right)^2 = 1.96 \\
\mathbf{x}_2, y_2 &\rightarrow \left(f_{\Theta_2}(4.0) - 4.0\right)^2 = \left((0.8 * 4.0 + 0.0) - 4.0\right)^2 = 0.64 \\
\mathbf{x}_3, y_3 &\rightarrow \left(f_{\Theta_2}(6.0) - 5.0\right)^2 = \left((0.8 * 6.0 + 0.0) - 5.0\right)^2 = 0.04 \\
\mathbf{x}_4, y_4 &\rightarrow \left(f_{\Theta_2}(8.0) - 6.0\right)^2 = \left((0.8 * 8.0 + 0.0) - 6.0\right)^2 = 0.16
\end{aligned}
\quad \longrightarrow \quad J(\Theta_2) = 0.7
$$

  - Θ_2를 개선하여 $\Theta_3 = (0.5,\ 2.0)\mathrm{T}$가 되었다고 가정
  - 이때 $J($



(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

Chapter1.4 – 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 $\hat{\Theta}$

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

Chapter1.4 – 간단한 기계 학습의 예

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

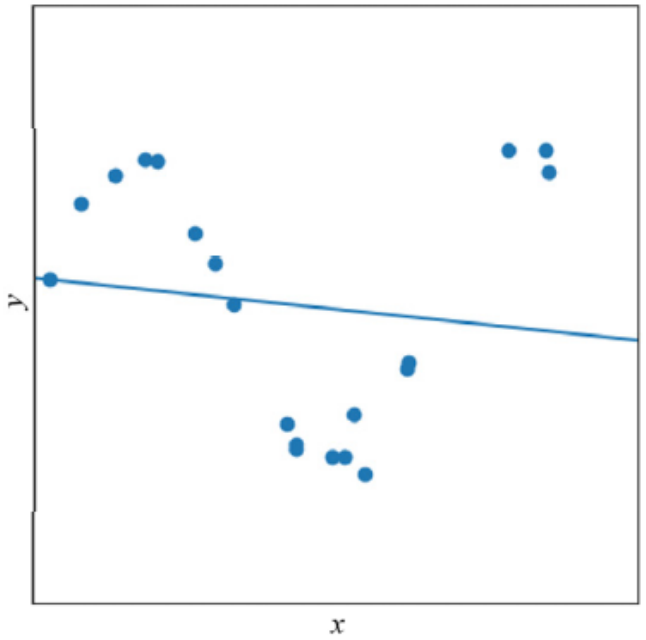


그림 1–12 선형 모델의 한계

Chapter1.5 모델 선택

Chapter1.5 – 모델 선택

- [그림 1.13]의 1차 모델은 과소적합
  - 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

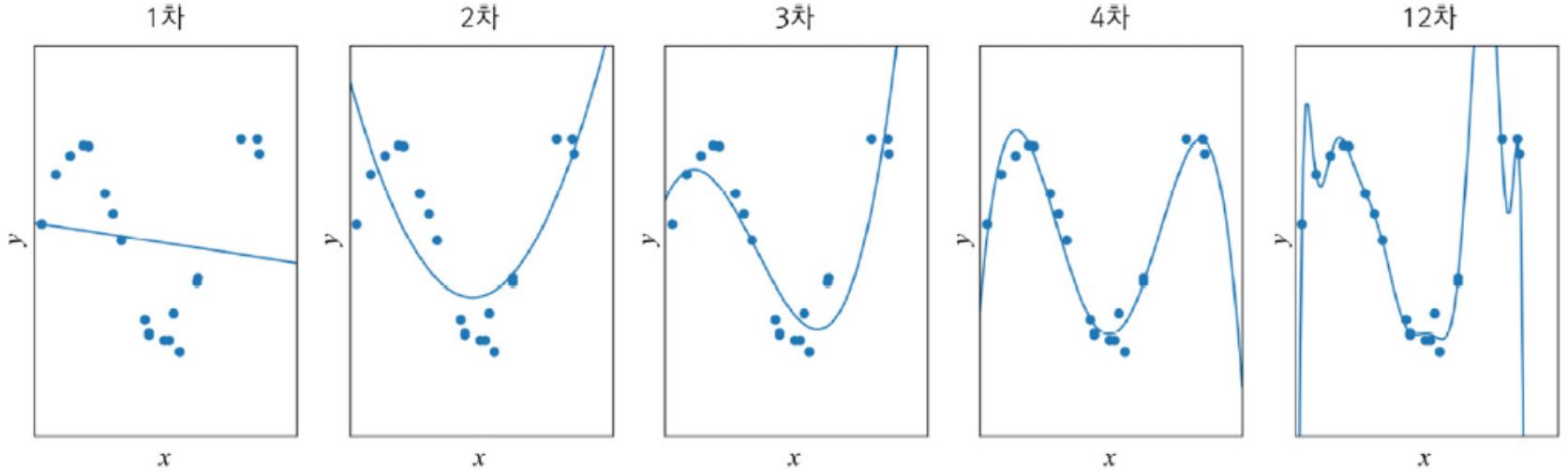


그림 1–13 과소적합과 과잉적합 현상

Chapter1.5 – 모델 선택

- 과잉적합
  - 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

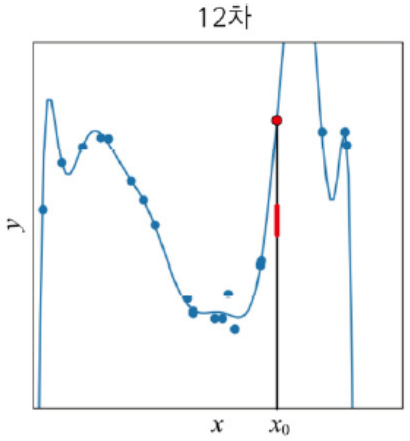


그림 1-14 과잉적합되었을 때 부정확한 예측 현상

Chapter1.5 – 모델 선택

- 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

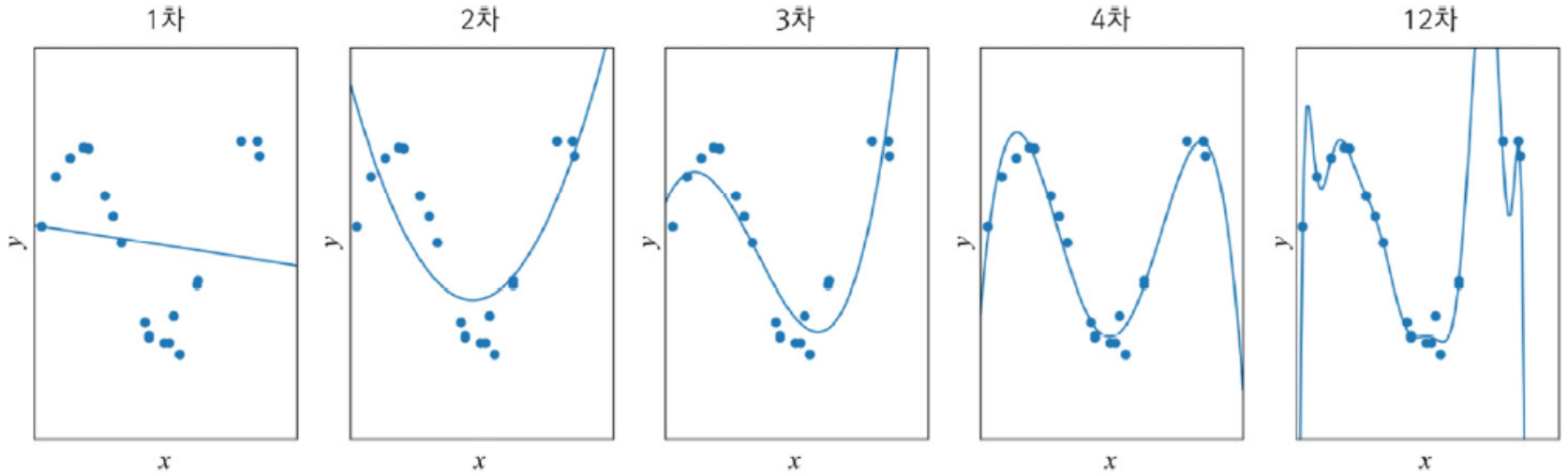
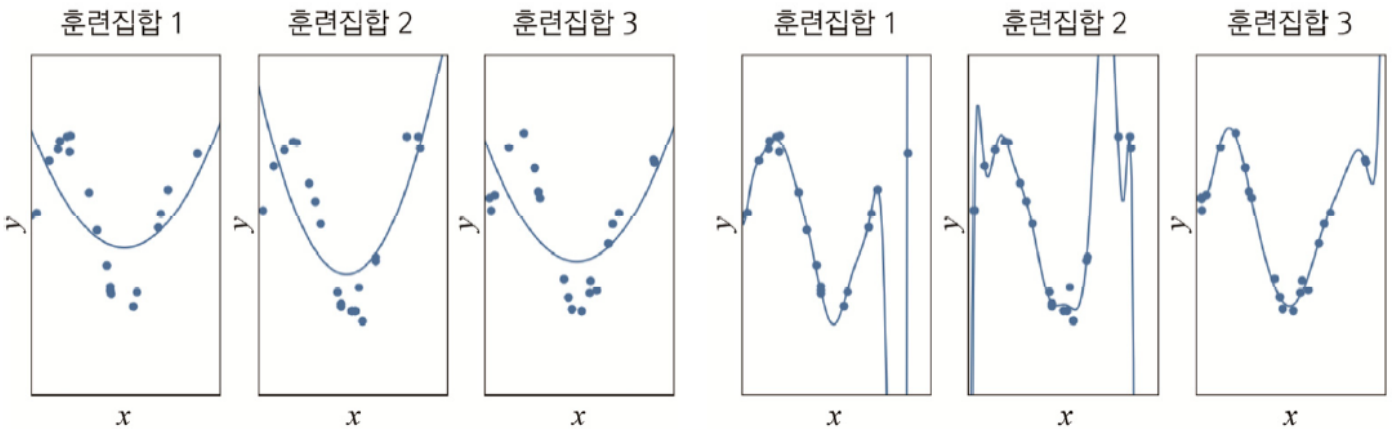


그림 1-13 과소적합과 과잉적합 현상

Chapter1.5 – 모델 선택

- 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

Chapter1.5 – 모델 선택

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

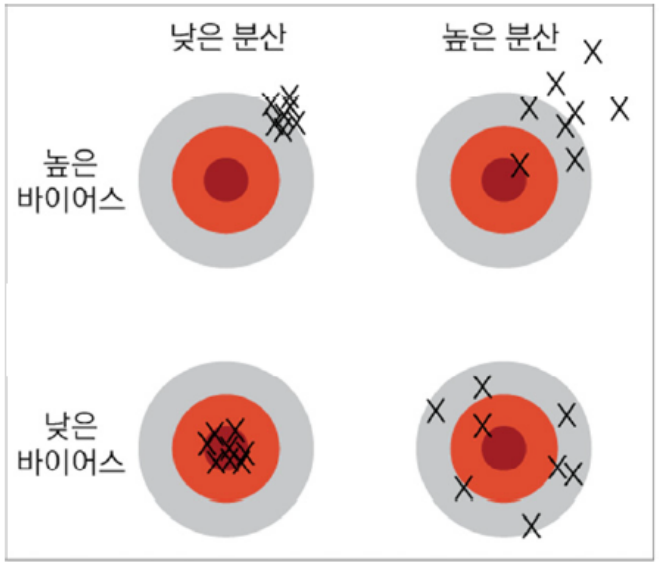


그림 1-16 바이어스와 분산

  - 하 계
  - 따라서 바이어스 희생을 최소로 유지하며 분산을 최대로 낮추는 전략 필요

Chapter1.5 – 모델 선택

- 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.
```

Chapter1.5 – 모델 선택

- 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.
```

Chapter1.5 – 모델 선택

- 부트스트랩bootstrap
  - 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

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

Chapter1.5 – 모델 선택

- [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 틀을 선택함
  - 그 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

Chapter1.5 – 모델 선택

- 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

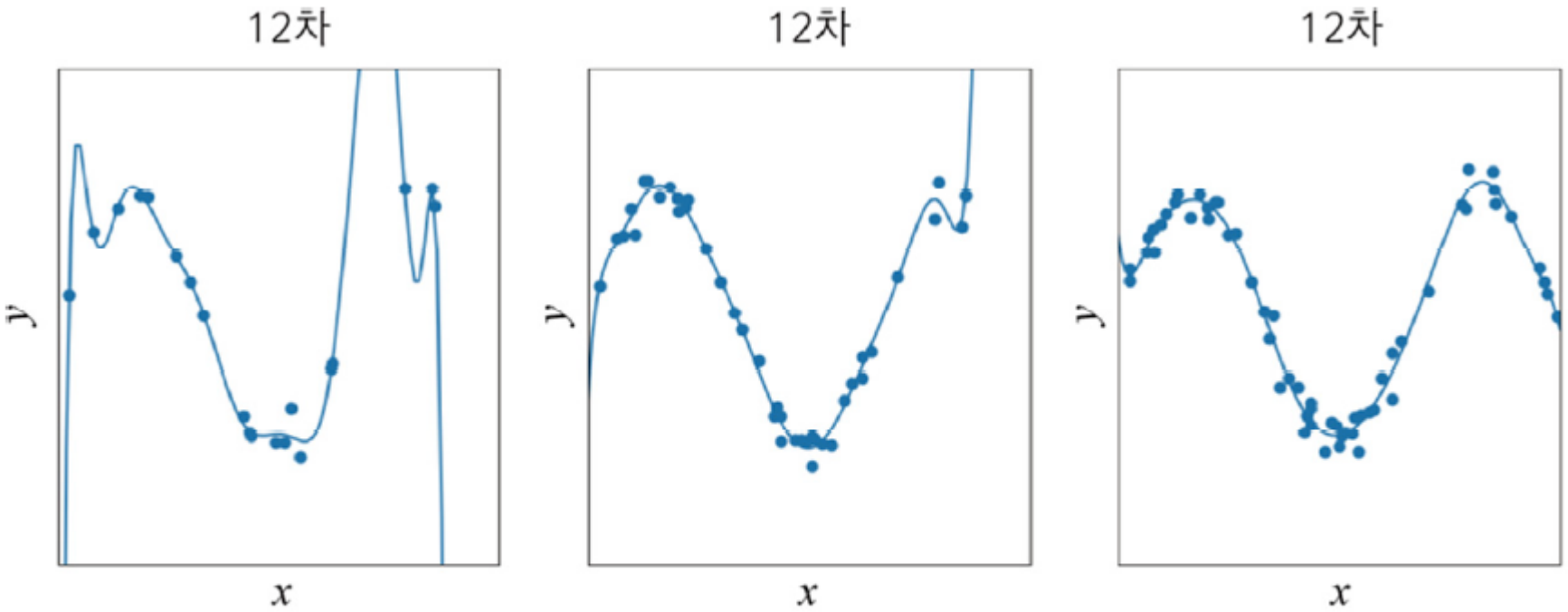
  규제regularization 기법을 적용함

  - 예) [그림 1-13]의 경우 12차 다항식을 선택한 후 적절히 규제를 적용

Chapter1.6 규제

Chapter1.6 – 규제

- 데이터를 더 많이 수집하면 일반화 능력이 향상됨



(a) 훈련집합의 크기 = 20 (b) 훈련집합의 크기 = 40 (c) 훈련집합의 크기 = 60

그림 1-17 데이터를 확대하여 일반화 능력을 향상함

Chapter1.6 – 규제

- 데이터 수집은 많은 비용이 듦
  - 그라운드 트루스를 사람이 일일이 레이블링해야 함
- 인위적으로 데이터 확대
  - 훈련집합에 있는 샘플을 변형함
  - 약간 회전 또는 와핑 (부류 소속이 변하지 않게 주의)

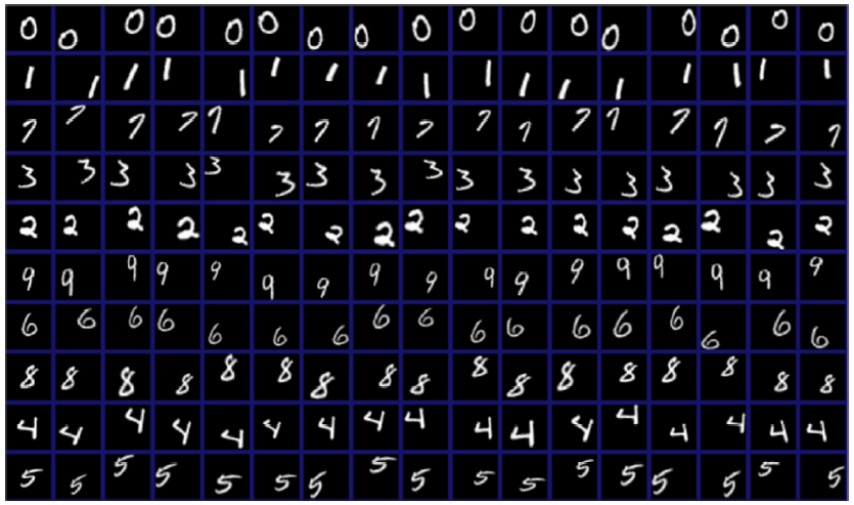


그림 5-24 필기 숫자 데이터의 다양한 변형[8]

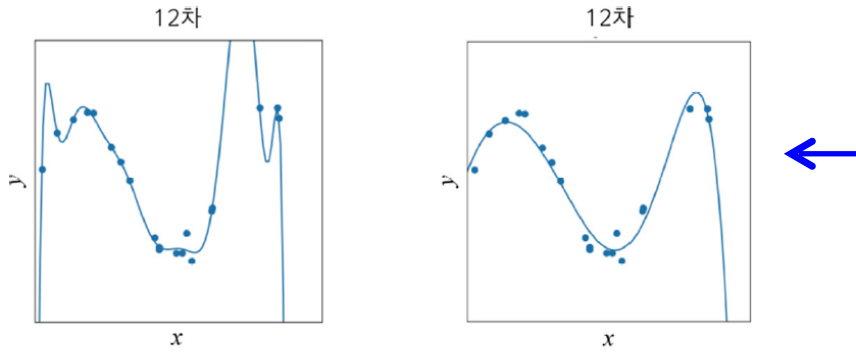
Chapter1.6 – 규제

- 가중치를 작게 조절하는 기법
  - [그림 1-18(a)]의 12차 곡선은 가중치가 매우 큼

$$y = 1005.7x^{12} - 27774.4x^{11} + \cdots - 22852612.5x^{1} - 12.8$$

  - 가중치 감쇠는 개선된 목적함수를 이용하여 가중치를 작게 조절하는 규제 기법
    - 식 (1.11)의 두 번째 항은 규제 항으로서 가중치 크기를 작게 유지해줌

$$J(\Theta) = \frac{1}{n}\sum_{i=1}^{n}(f_{\Theta}(\mathbf{x}_i) - y_i)^2 + \lambda\|\Theta\|_2^2 \qquad (1.11)$$



(a) 가중치 감쇠 적용 안 함[식 (1.8)의 목적함수] (b) 가중치 감쇠 적용함[식 (1.11)의 목적함수]

$y = 10.779x^{12} - 42.732x^{11} + \cdots - 2.379x^{1} + 0.119$

그림 1-18 가중치 감쇠에 의한 규제 효과

Chapter1.6 – 규제

- L1 Regularization (Lasso)

L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda \sum_{j=0}^{M} |W_j|$$

- L2 Regularization (Ridge)

L2 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda \sum_{j=0}^{M} W_j^2$$

Loss function

Regularization Term

출처: https://seongyun-dev.tistory.com/52

Chapter1.6 – 규제

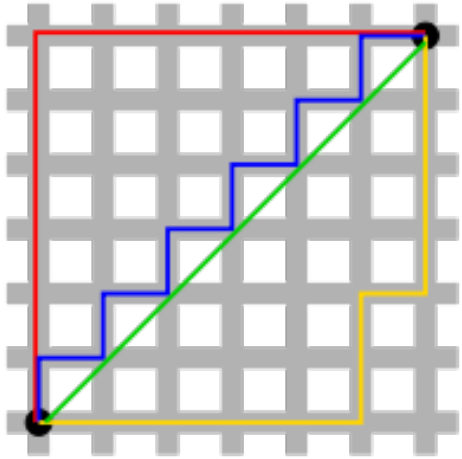
- L1 Regularization (Lasso)

$$Cost = \frac{1}{n}\sum_{i=1}^{n}\{L(y_i, \hat{y_i}) + \frac{\lambda}{2}|w|\}$$

$L(y_i, \hat{y_i})$: 기존의 Cost function

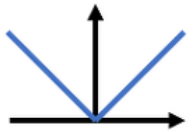
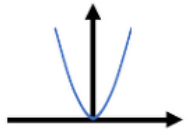
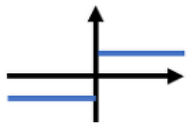
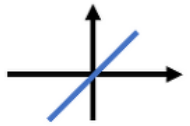
- L2 Regularization (Ridge)

$$Cost = \frac{1}{n}\sum_{i=1}^{n}\{L(y_i, \hat{y_i}) + \frac{\lambda}{2}|w|^2\}$$



출처: https://light-tree.tistory.com/125

Chapter1.6 – 규제

구분	Lasso	Ridge
수식	$\|\boldsymbol{w}\|_1$ 	$\frac{1}{2}\|\boldsymbol{w}\|^2$ 
미분	$\begin{cases} 1 & w < 0 \\ -1 & w > 0 \end{cases}$ 	$\boldsymbol{w}$ 
특성	• 가중치 값을 정확하게 0으로 만듦 • 중요한 특징을 '선택'하는 효과 • 모델에 Sparsity를 가함.	• 큰 가중치의 값을 작게 만듦 • 모델 전반적인 복잡도를 감소시키는 효과 • 가중치의 값이 0이 되게 하지는 못함

출처: https://seongyun-dev.tistory.com/52

Chapter1.7 기계 학습 유형

Chapter1.7 – 기계 학습 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
  - 6장의 주제

Chapter1.7 – 기계 학습 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

Chapter1.7 – 기계 학습 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

Chapter1.7 – 기계 학습 유형

- 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

Chapter1.8 기계 학습의 과거와 현재, 미래

Chapter1.8 – 기계 학습의 과거와 현재, 미래

1843	에이더 "… 해석엔진은 꽤 복잡한 곡을 작곡할 수도 있다."라는 논문 발표[Ada1843]
1950	인공지능 여부를 판별하는 튜링 테스트[Turing1950]
1956	최초의 인공지능 학술대회인 다트머스 콘퍼런스 개최. '인공지능' 용어 탄생[McCarthy1955]
1958	로젠블렛이 퍼셉트론 제안[Rosenblatt1958]
	인공지능 언어 Lisp 탄생
1959	사무엘이 기계 학습을 이용한 체커 게임 프로그램 개발[Samuel1959]
1969	민스키가 퍼셉트론의 과대포장 지적. 신경망 내리막길 시작[Minsky1969]
	제1회 IJCAI International Joint Conference on Artificial Intelligence 개최
1972	인공지능 언어 Prolog 탄생
1973	Lighthill 보고서로 인해 인공지능 내리막길, 인공지능 겨울AI winter 시작
1974	웨어보스가 오류 역전파 알고리즘을 기계 학습에 도입[Werbos1974]
1975경	의료진단 전문가 시스템 Mycin – 인공지능에 대한 관심 부활
1979	「IEEE Transactions on Pattern Analysis and Machine Intelligence」 저널 발간
1980	제1회 ICML International Conference on Machine Learning 개최
	후쿠시마가 NeoCognitron 제안[Fukushima1980]
1986	「Machine Learning」 저널 발간
	『Parallel Distributed Processing』 출간
	다층 퍼셉트론으로 신경망 부활

Chapter1.8 – 기계 학습의 과거와 현재, 미래

1987	Lisp 머신의 시장 붕괴로 제2의 인공지능 겨울
	UCI 리포지토리 서비스 시작
	NIPS Neural Information Processing Systems 콘퍼런스 시작
1989	「Neural Computation」 저널 발간
1993	R 언어 탄생
1997	IBM 딥블루가 세계 체스 챔피언인 카스파로프 이김
	LSTM Long short-term memory 개발됨
1998경	SVM이 MNIST 인식 성능에서 신경망 추월
1998	르쿤이 CNN의 실용적인 학습 알고리즘 제안[LeCun1998]
	『Neural Networks: Tricks of the Trade』 출간
1999	NVIDIA 사에서 GPU 공개
2000	「Journal of Machine Learning Research」 저널 발간
	OpenCV 최초 공개
2004	제1회 그랜드 챌린지(자율 주행)
2006	층별학습 탄생[Hinton2006a]
2007경	딥러닝이 MNIST 인식 성능에서 SVM 추월
2007	GPU 프로그래밍 라이브러리인 CUDA 공개

Chapter1.8 – 기계 학습의 과거와 현재, 미래

	어번 챌린지(도심 자율 주행)
	Scikit-learn 라이브러리 최초 공개
2009	Theano 서비스 시작
2010	ImageNet 탄생
	제1회 ILSVRC 대회
2011	IBM 왓슨이 제퍼디 우승자 꺾음
2012	MNIST에 대해 0.23% 오류율 달성
	AlexNet 발표 (3회 ILSVRC 우승)
2013	제1회 ICLR International Conference on Learning Representations 개최
2014	Caffe 서비스 시작
2015	TensorFlow 서비스 시작
	OpenAI 창립
2016	알파고와 이세돌의 바둑 대회에서 알파고 승리[Silver2016]
	『Deep Learning』 출간
2017	알파고 제로[Silver2017]

Chapter2.1 선형대수

Chapter2.1 – 선형대수

- 벡터
  - 샘플을 특징 벡터로feature vector 표현
  - 예) Iris 데이터에서 꽃받침의 길이, 꽃받침의 너비, 꽃잎의 길이, 꽃잎의 너비라는 4개의 특징이 각각 5.1, 3.5, 1.4, 0.2인 샘플

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}$$

  - 여러 개의 특징 벡터를 첨자로 구분

$$\mathbf{x}_1 = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 4.9 \\ 3.0 \\ 1.4 \\ 0.2 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 4.7 \\ 3.2 \\ 1.3 \\ 0.2 \end{pmatrix}, \cdots, \mathbf{x}_{150} = \begin{pmatrix} 5.9 \\ 3.0 \\ 5.1 \\ 1.8 \end{pmatrix}$$

Chapter2.1 – 선형대수

- 행렬
  - 여러 개의 벡터를 담음
  - 훈련집합을 담은 행렬을 설계행렬이라 부름
  - 예) Iris 데이터에 있는 150개의 샘플을 설계 행렬 $\mathbf{X}$로 표현

$$\mathbf{X} = \begin{pmatrix} 5.1 & 3.5 & 1.4 & 0.2 \\ 4.9 & 3.0 & 1.4 & 0.2 \\ 4.7 & 3.2 & 1.3 & 0.2 \\ 4.6 & 3.1 & 1.5 & 0.2 \\ \vdots & \vdots & \vdots & \vdots \\ 6.2 & 3.4 & 5.4 & 2.3 \\ 5.9 & 3.0 & 5.1 & 1.8 \end{pmatrix} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \\ \vdots & \vdots & \vdots & \vdots \\ x_{149,1} & x_{149,2} & x_{149,3} & x_{149,4} \\ x_{150,1} & x_{150,2} & x_{150,3} & x_{150,4} \end{pmatrix}$$

행row

열column

Chapter2.1 – 선형대수

- 행렬 $\mathbf{A}$의 전치행렬 $\mathbf{A}^{\mathrm{T}}$

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}, \quad \mathbf{A}^{\mathrm{T}} = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{pmatrix}$$

예를 들어, $\mathbf{A} = \begin{pmatrix} 3 & 4 & 1 \\ 0 & 5 & 2 \end{pmatrix}$라면 $\mathbf{A}^{\mathrm{T}} = \begin{pmatrix} 3 & 0 \\ 4 & 5 \\ 1 & 2 \end{pmatrix}$

  - Iris의 설계 행렬을 전치행렬 표기에 따라 표현하면,

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_{150}^{\mathrm{T}} \end{pmatrix}$$

Chapter2.1 – 선형대수

- 행렬을 이용하면 수학을 간결하게 표현할 수 있음
  - 예) 다항식의 행렬 표현

$$
\begin{aligned}
f(\mathbf{x}) &= f(x_1, x_2, x_3) \\
&= 2x_1x_1 - 4x_1x_2 + 3x_1x_3 + x_2x_1 + 2x_2x_2 + 6x_2x_3 - 2x_3x_1 + 3x_3x_2 + 2x_3x_3 + 2x_1 + 3x_2 - 4x_3 + 5 \\
&= (x_1 \quad x_2 \quad x_3)\begin{pmatrix} 2 & -4 & 3 \\ 1 & 2 & 6 \\ -2 & 3 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + (2 \quad 3 \quad -4)\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + 5 \\
&= \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} + \mathbf{b}^{\mathrm{T}}\mathbf{x} + c
\end{aligned}
$$

- 특수한 행렬들

$$
\text{정사각행렬}\begin{pmatrix} 2 & 0 & 1 \\ 1 & 21 & 5 \\ 4 & 5 & 12 \end{pmatrix}, \quad \text{대각행렬}\begin{pmatrix} 50 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 8 \end{pmatrix},
$$

$$
\text{단위행렬}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \text{대칭행렬}\begin{pmatrix} 1 & 2 & 11 \\ 2 & 21 & 5 \\ 11 & 5 & 1 \end{pmatrix}
$$

Chapter2.1 – 선형대수

- 행렬 연산
  - 행렬 곱셈

$$\mathbf{C} = \mathbf{AB}, \text{이때 } c_{ij} = \sum a_{ik} b_{kj} \qquad (2.1)$$

$2*3$ 행렬 $\mathbf{A} = \begin{pmatrix} 3 & 4 & 1 \\ 0 & 5 & 2 \end{pmatrix}$와 $3*3$행렬 $\mathbf{B} = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 0 & 5 \\ 4 & 5 & 1 \end{pmatrix}$을 곱하면 $2*3$ 행렬 $\mathbf{C} = \mathbf{AB} = \begin{pmatrix} 14 & 5 & 24 \\ 13 & 10 & 27 \end{pmatrix}$

  - 교환법칙 성립하지 않음: $\mathbf{AB} \neq \mathbf{BA}$
  - 분배법칙과 결합법칙 성립: $\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB} + \mathbf{AC}$이고 $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$

  - 벡터의 내적

$$\text{벡터의 내적 } \mathbf{a} \cdot \mathbf{b} = \mathbf{a}^{\mathrm{T}}\mathbf{b} = \sum_{k=1,d} a_k b_k \qquad (2.2)$$

$\mathbf{x}_1 = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}$와 $\mathbf{x}_2 = \begin{pmatrix} 4.9 \\ 3.0 \\ 1.4 \\ 0.2 \end{pmatrix}$의 내적 $\mathbf{x}_1 \cdot \mathbf{x}_2$는 37.49

Chapter2.1 – 선형대수

- 텐서
  - 3차원 이상의 구조를 가진 숫자 배열
  - 예) 3차원 구조의 RGB 컬러 영상

$$\mathbf{A} = \begin{pmatrix} \begin{pmatrix} 3 & 0 & 1 & 2 & 6 & 7 \\ 3 & 1 & 2 & 3 & 5 & 6 \\ 1 & 2 & 2 & 2 & 2 & 3 \\ 3 & 0 & 0 & 1 & 1 & 0 \\ 5 & 4 & 1 & 3 & 3 & 3 \\ 2 & 2 & 1 & 2 & 2 & 1 \end{pmatrix} & \begin{pmatrix} 6 \\ 3 \\ 0 \\ 3 \\ 1 \\ 1 \end{pmatrix} \\ \begin{pmatrix} 2 & 0 & 2 & 2 & 3 & 1 \end{pmatrix} & \begin{pmatrix} 6 \end{pmatrix} \end{pmatrix} \begin{pmatrix} \begin{pmatrix} 4 & 1 & 0 & 3 & 2 & 2 \\ 2 & 0 & 2 & 2 & 3 & 1 \end{pmatrix} & \begin{pmatrix} 6 \\ 3 \\ 0 \\ 3 \\ 1 \\ 1 \end{pmatrix} \\ \begin{pmatrix} 3 & 0 & 1 & 2 & 6 & 7 \end{pmatrix} & \begin{pmatrix} 6 \end{pmatrix} \end{pmatrix}$$

Chapter2.1 – 선형대수

- 벡터와 행렬의 크기를 놈으로 측정
  - 벡터의 p차 놈

$$p\text{차 놈: } \|\mathbf{x}\|_p = \left(\sum_{i=1,d} |x_i|^p \right)^{\frac{1}{p}} \tag{2.3}$$

$$\text{최대 놈: } \|\mathbf{x}\|_\infty = \max(|x_1|, |x_2|, \cdots, |x_d|) \tag{2.4}$$

  - 예) $\mathbf{x} = (3 \ \ -4 \ \ 1)$ 일 때, 2차 놈은 $\|\mathbf{x}\|_2 = (3^2 + (-4)^2 + 1^2)^{1/2} = 5.099$

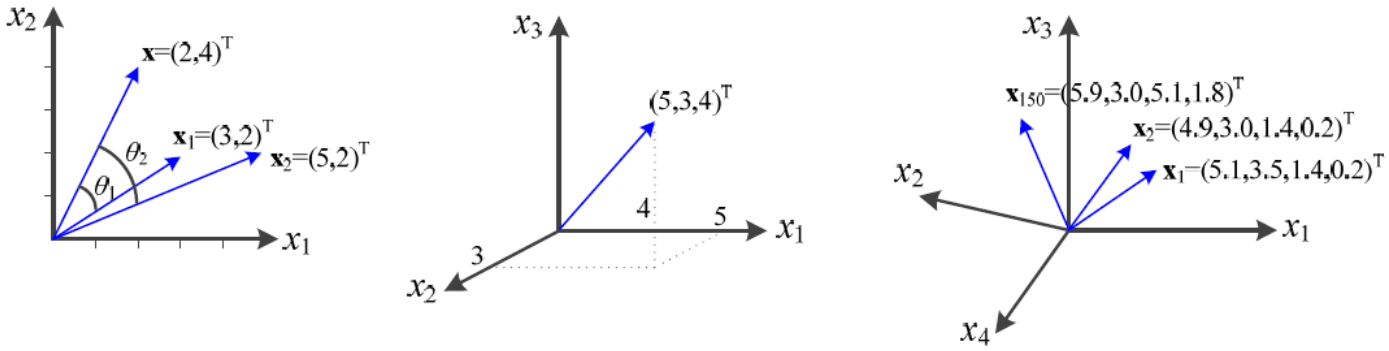
  - 행렬의 프로베니우스 놈

$$\text{프로베니우스 놈: } \|\mathbf{A}\|_F = \left(\sum_{i=1,n} \sum_{j=1,m} a_{ij}^2 \right)^{\frac{1}{2}} \tag{2.6}$$

예를 들어, $\left\| \begin{pmatrix} 2 & 1 \\ 6 & 4 \end{pmatrix} \right\|_F = \sqrt{2^2 + 1^2 + 6^2 + 4^2} = 7.550$

Chapter2.1 – 선형대수

- 유사도와 거리
  - 벡터를 기하학적으로 해석



(a) 2차원 벡터 (b) 3차원 벡터 (c) 4차원 벡터(Iris 데이터)

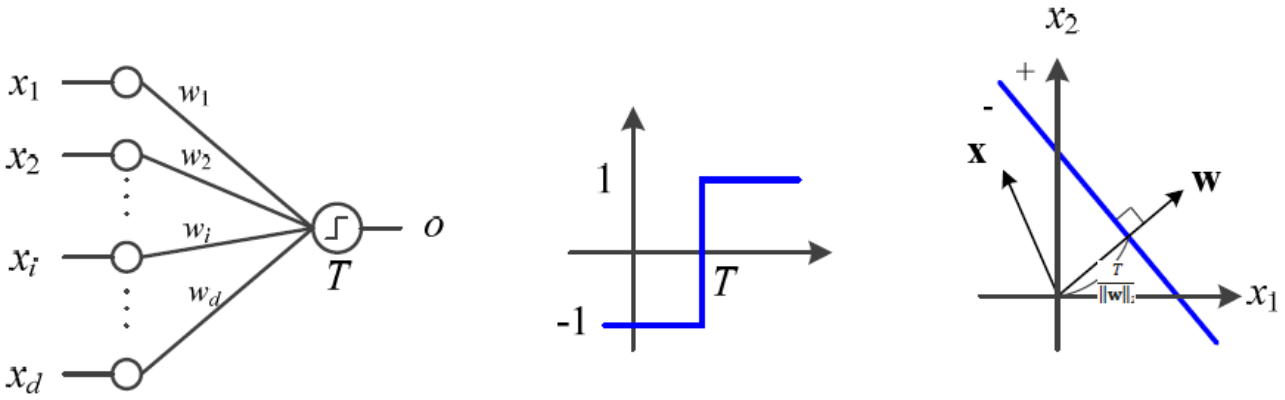
그림 2-2 벡터를 기하학적으로 해석

  - 코사인 유사도

$$cosine\_similarity(\mathbf{a},\mathbf{b}) = \frac{\mathbf{a}}{\|\mathbf{a}\|} \cdot \frac{\mathbf{b}}{\|\mathbf{b}\|} = cos(\theta) \qquad (2.7)$$

Chapter2.1 – 선형대수

- 퍼셉트론
  - 1958년 로젠블렛이 고안한 분류기 모델



(a) 퍼셉트론 구조 (b) 계단형 활성함수(비선형) (c) 퍼셉트론의 공간 분할

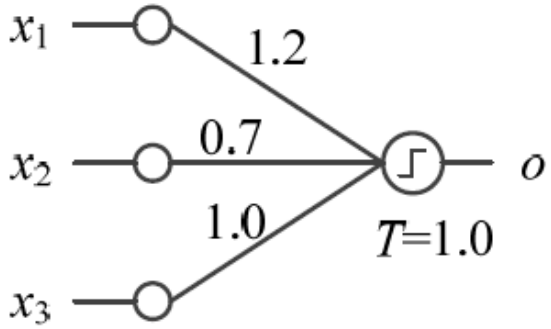
그림 2-3 퍼셉트론의 구조와 동작

- 퍼

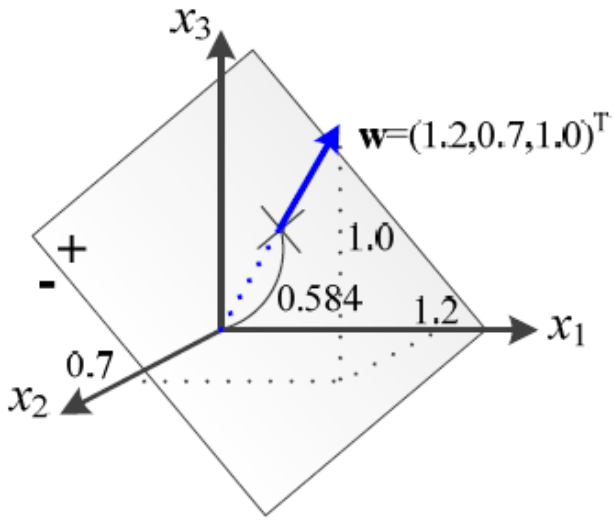
$$o = \tau(\mathbf{w} \cdot \mathbf{x}), \quad \text{이때} \quad \tau(a) = \begin{cases} 1, & a \geq T \\ -1, & a < T \end{cases} \qquad (2.8)$$

Chapter2.1 – 선형대수

- 퍼셉트론
  - [그림 2-3(c)]의 파란 직선은 두 개의 부분공간을 나누는 결정직선decision line
    - $\mathbf{w}$에 수직이고 원점으로부터 $\frac{T}{\|\mathbf{w}\|_2}$만큼 떨어져 있음
  - 3차원 특징공간은 결정평면decision plane, 4차원 이상은 결정 초평면decision hyperplane
  - 예) 3차원 특징공간을 위한 퍼셉트론



(a) 퍼셉트론



(b) 공간 분할(2부류 분류)

그림 2-4 퍼셉트론의 예(3차원)

Chapter2.1 – 선형대수

- 출력이 여러 개인 퍼셉트론

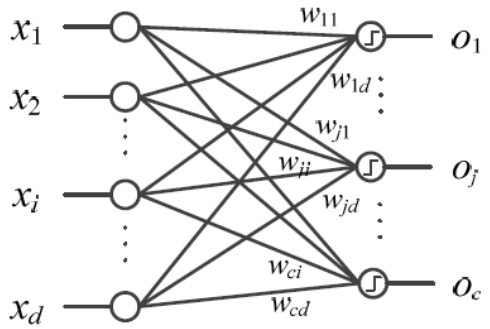


그림 2-5 출력이 여러 개인 퍼셉트론

출력은 벡터 $\mathbf{o} = (o_1, o_2, \cdots, o_c)^{\mathrm{T}}$로 표기

j번째 퍼셉트론의 가중치 벡터를
$\mathbf{w}_j = (w_{j1}, w_{j2}, \cdots, w_{jd})^{\mathrm{T}}$와 같이 표기

- [illegible]

행렬로 간결하게 쓰면 $\mathbf{o} = \boldsymbol{\tau}(\mathbf{W}\mathbf{x})$

$$\mathbf{o} = \boldsymbol{\tau}\begin{pmatrix} \mathbf{w}_1 \cdot \mathbf{x} \\ \mathbf{w}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{w}_c \cdot \mathbf{x} \end{pmatrix} \longrightarrow$$

이때 $\mathbf{W} = \begin{pmatrix} \mathbf{w}_1^{\mathrm{T}} \\ \mathbf{w}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{w}_c^{\mathrm{T}} \end{pmatrix}$

- 가중치 벡터들 각 부류의 기준 벡터로 간주하 [illegible]

Chapter2.1 – 선형대수

- 학습의 정의
  - 식 (2.10)은 학습을 마친 프로그램을 현장에 설치했을 때 일어나는 과정

분류라는 과업: $\hat{\mathbf{o}}^{?} = \boldsymbol{\tau}(\hat{\mathbf{W}}^{앎}\hat{\mathbf{x}}^{앎})$ (2.10)

  - 식 (2.11)은 학습 과정
    - 학습은 훈련집합의 샘플에 대해 식 (2.11)을 가장 잘 만족하는 **W**를 찾아내는 작업

학습이라는 과업: $\hat{\mathbf{o}}^{앎} = \boldsymbol{\tau}(\hat{\mathbf{W}}^{?}\hat{\mathbf{x}}^{앎})$ (2.11)

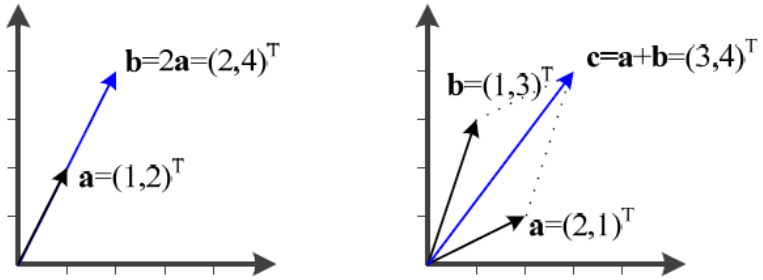
- 현대 기계 학습에서 퍼셉트론의 중요성
  - 딥러닝은 퍼셉트론을 여러 층으로 확장하여 만듦

Chapter2.1 – 선형대수

- 벡터
  - 공간상의 한 점으로 화살표 끝이 벡터의 좌표에 해당
- 선형결합이 만드는 벡터공간
  - 기저벡터 **a**와 **b**의 선형결합

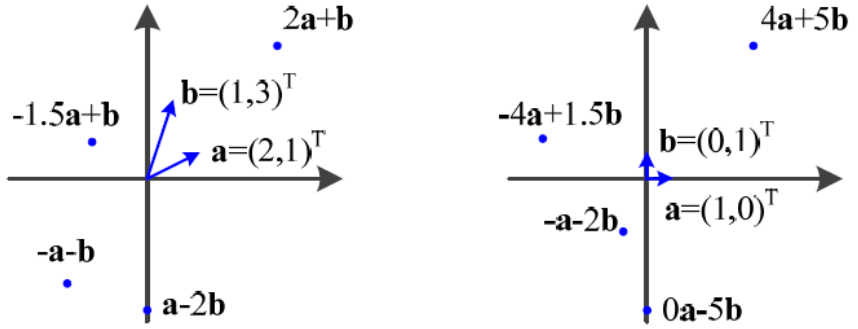
$$\mathbf{c} = \alpha_1 \mathbf{a} + \alpha_2 \mathbf{b} \qquad (2.12)$$

  - 선형결합으로 만들어지는 공간을 벡터공간이라 부름



(a) 벡터에 스칼라 곱 (b) 두 벡터의 덧셈

그림 2-6 벡터의 연산

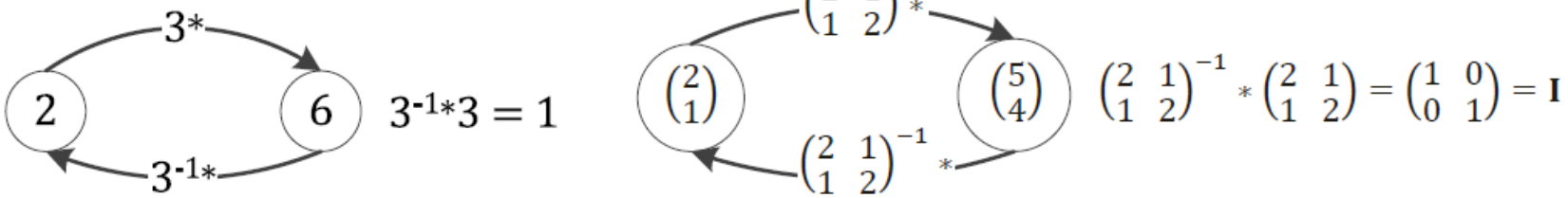


(a) 기저 벡터와 벡터공간 (b) 정규직교 기저 벡터

그림 2-7 벡터공간

Chapter2.1 – 선형대수

- 역행렬의 원리



(a) 역수의 원리 (b) 역행렬의 원리

그림 2–9 역행렬

- 정사각행렬 $\mathbf{A}$의 역행렬 $\mathbf{A}^{-1}$

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

- 예를 들어, $\begin{pmatrix} 2 & 1 \\ 6 & 4 \end{pmatrix}$의 역행렬은 $\begin{pmatrix} 2 & -0.5 \\ -3 & 1 \end{pmatrix}$

Chapter2.1 – 선형대수

- 정리

정리 2–1 다음 성질은 서로 필요충분조건이다.

- $\mathbf{A}$는 역행렬을 가진다. 즉, 특이행렬이 아니다.
- $\mathbf{A}$는 최대계수를 가진다.
- $\mathbf{A}$의 모든 행이 선형독립이다.
- $\mathbf{A}$의 모든 열이 선형독립이다.
- $\mathbf{A}$의 행렬식은 0이 아니다.
- $\mathbf{A}^{\mathrm{T}}\mathbf{A}$는 양의 정부호positive definite 대칭 행렬이다.
- $\mathbf{A}$의 고윳값은 모두 0이 아니다.

Chapter2.1 – 선형대수

- 행렬 $\mathbf{A}$의 행렬식 $det(\mathbf{A})$

$$\left.\begin{aligned} det\begin{pmatrix} a & b \\ c & d \end{pmatrix} &= ad - bc \\ det\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} &= aei + bfg + cdh - ceg - bdi - afh \end{aligned}\right\} \quad (2.15)$$

예를 들어 $\begin{pmatrix} 2 & 1 \\ 6 & 4 \end{pmatrix}$의 행렬식은 2*4−1*6=2

- 기하학적 의미
  - 2차원에서는 2개의 행 벡터가 이루는 평행사변형의 넓이
  - 3차원에서는 3개의 행 벡터가 이루는 평행사각기둥의 부피

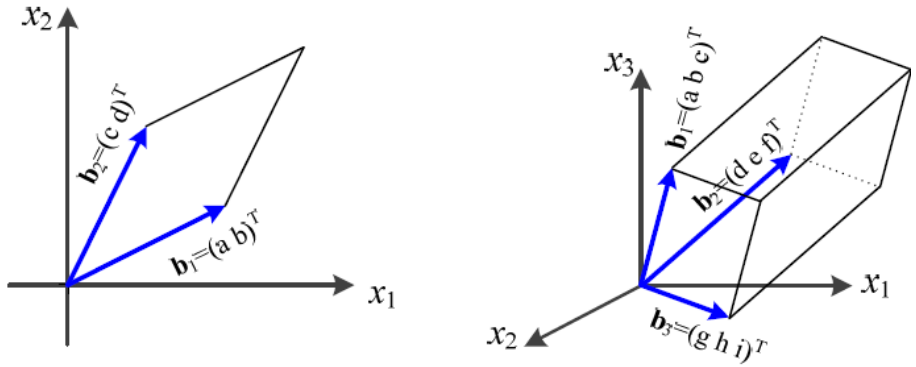


그림 2–10 행렬식의 기하학적 해석

Chapter2.1 – 선형대수

- 정부호 행렬

양의 정부호 행렬 : $\mathbf{0}$이 아닌 모든 벡터 $\mathbf{x}$에 대해, $\mathbf{x}^T\mathbf{A}\mathbf{x}>0$

- 예를 들어, $\begin{pmatrix}1 & 0\\ 0 & 2\end{pmatrix}$는 $(x_1 \quad x_2)\begin{pmatrix}1 & 0\\ 0 & 2\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} = x_1^2 + 2x_2^2$

$\begin{pmatrix}1 & 0\\ 0 & 2\end{pmatrix}$는 양의 정부호 행렬

양의 준정부호positive semi-definite 행렬: $\mathbf{0}$이 아닌 모든 벡터 $\mathbf{x}$에 대해, $\mathbf{x}^T\mathbf{A}\mathbf{x}\geq 0$

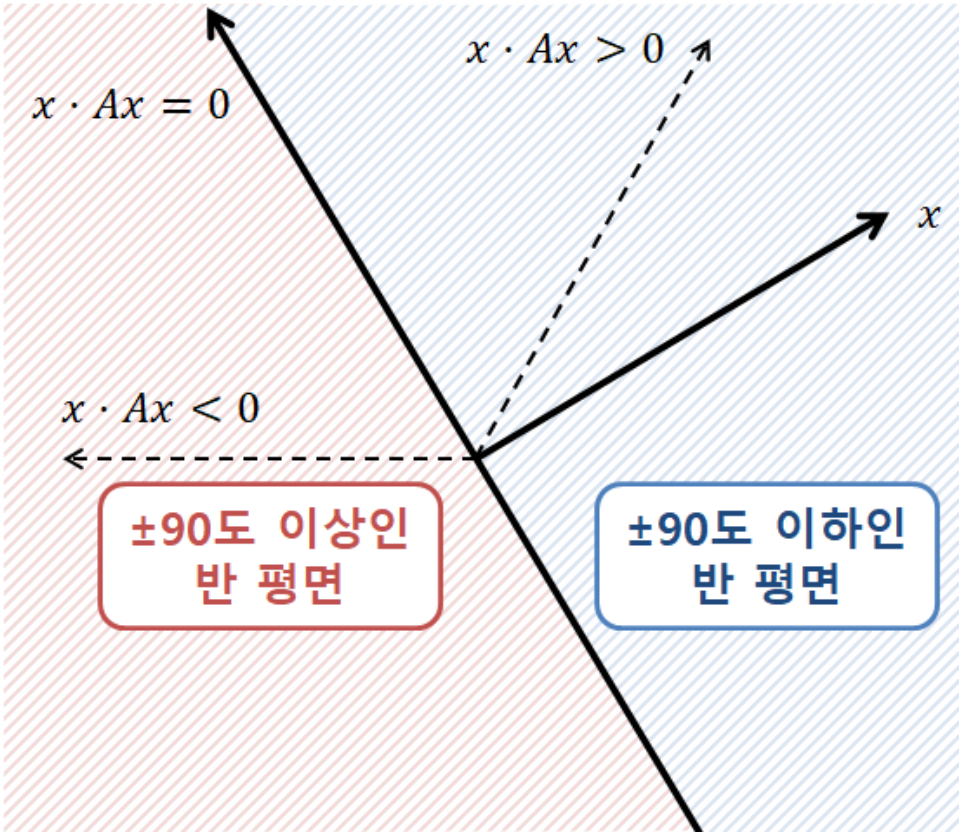
음의 정부호negative definite 행렬: $\mathbf{0}$이 아닌 모든 벡터 $\mathbf{x}$에 대해, $\mathbf{x}^T\mathbf{A}\mathbf{x}< 0$

음의 준정부호negative semi-definite 행렬: $\mathbf{0}$이 아닌 모든 벡터 $\mathbf{x}$에 대해, $\mathbf{x}^T\mathbf{A}\mathbf{x}\leq 0$

Chapter2.1 – 선형대수

- 양의 정부호 행렬

- 양의 실수처럼 양의 정부호 행렬을 이용한 선형변환은 입력 벡터를 '뒤집어주지는 않는' 것



출처: https://angeloyeo.github.io/2021/12/20/positive_definite.html

Chapter2.1 – 선형대수

- 분해란?
  - 정수 3717은 특성이 보이지 않지만, 3*3*7*59로 소인수 분해를 하면 특성이 보이듯이,

    행렬도 분해하면 여러모로 유용함

- 고윳값과 고유 벡터
  - 고유 벡터 $\mathbf{v}$와 고윳값 λ

    $$\mathbf{Av} = \lambda\mathbf{v} \qquad (2.20)$$

  - 예를 들어, $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = 3\begin{pmatrix} 1 \\ 1 \end{pmatrix}$이고 $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = 1\begin{pmatrix} 1 \\ -1 \end{pmatrix}$이므로, $\lambda_1 = 3, \lambda_2 = 1$이고

    $\mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\mathbf{v}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

Chapter2.1 – 선형대수

- 고윳값과 고유 벡터의 기하학적 해석

예제 2–5

[그림 2-12]의 반지름이 1인 원 위에 있는 4개의 벡터 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$가 $\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$에 의해 어떻게 변환되는지 살펴보자. 변환 후의 벡터를 각각 $\mathbf{x}_1', \mathbf{x}_2', \mathbf{x}_3', \mathbf{x}_4'$로 표기한다.

$$\mathbf{x}_1' = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 3/\sqrt{2} \\ 3/\sqrt{2} \end{pmatrix}$$
$$\mathbf{x}_2' = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$
$$\mathbf{x}_3' = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}$$
$$\mathbf{x}_4' = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \end{pmatrix}$$

눈 여겨 볼 점은 $\mathbf{A}$의 고유 벡터 $\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}$과 방향이 같은 $\mathbf{x}_1$과 $\mathbf{x}_3$이다. 이들은 변환 때문에 길이가 달라지더라도 방향은 그대로 유지한다. 식 (2.20)을 충실히 따르고 있다. 이때 길이의 변화는 고윳값 λ에 따른다. 즉, $\mathbf{x}_1$은 3배만큼, $\mathbf{x}_3$은 1배만큼 길이가 변한다. 나머지 $\mathbf{x}_2$와 $\mathbf{x}_4$는 길이와 방향이 모두 변한다. 파란 원 위에 있는 모든 점을 변환하면 빨간색의 타원이 된다. 파란 원 위에 존재하는 무수히 많은 점(벡터) 중에 방향이 바뀌지 않는 것은 고유 벡터에 해당하는 $\mathbf{x}_1$과 $\mathbf{x}_3$뿐이다.

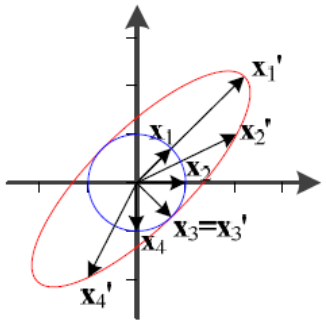


그림 2–12 고유 벡터의 공간 변환

Chapter2.1 – 선형대수

- 고윳값 분해eigen value decomposition

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} \qquad (2.21)$$

  - $\mathbf{Q}$는 $\mathbf{A}$의 고유 벡터를 열에 배치한 행렬이고 $\mathbf{\Lambda}$는 고윳값을 대각선에 배치한 대각행렬
  - 예를 들어, $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix}$
  - 고윳값 분해는 정사각행렬에만 적용 가능한데, 기계 학습에서는 정사각행렬이 아닌 경우의 분해도 필요하므로 고윳값 분해는 한계를 가짐

Chapter2.1 – 선형대수

- $n*m$ 행렬 $\mathbf{A}$의 특잇값 분해SVD(singular value decomposition)

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathbf{T}} \qquad (2.22)$$

  - 왼쪽 특이행렬 $\mathbf{U}$는 $\mathbf{A}\mathbf{A}^{\mathrm{T}}$의 고유 벡터를 열에 배치한 $n*n$ 행렬
  - 오른쪽 특이행렬 $\mathbf{V}$는 $\mathbf{A}^{\mathrm{T}}\mathbf{A}$의 고유 벡터를 열에 배치한 $m*m$ 행렬
  - $\boldsymbol{\Sigma}$는 $\mathbf{A}\mathbf{A}^{\mathrm{T}}$의 고윳값의 제곱근을 대각선에 배치한 $n*m$ 대각행렬

예를 들어, $\mathbf{A}$를 $4*3$ 행렬이라고 했을 때 다음과 같이 특잇값 분해가 된다.

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 2 \\ 3 & 1 & 0 \\ 1 & 0 & 2 \end{pmatrix} = \begin{pmatrix} -0.1914 & -0.2412 & 0.1195 & -0.9439 \\ -0.5144 & 0.6990 & -0.4781 & -0.1348 \\ -0.6946 & -0.6226 & -0.2390 & 0.2697 \\ -0.4651 & 0.2560 & 0.8367 & 0.1348 \end{pmatrix}$$

$$\begin{pmatrix} 3.7837 & 0 & 0 \\ 0 & 2.7719 & 0 \\ 0 & 0 & 1.4142 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -0.7242 & -0.4555 & -0.5177 \\ -0.6685 & 0.2797 & 0.6891 \\ 0.1690 & -0.8452 & 0.5071 \end{pmatrix}$$

Chapter2.2 확률과 통계

Chapter2.2 – 확률과 통계

- 확률변수random variable
  - 예) 윷

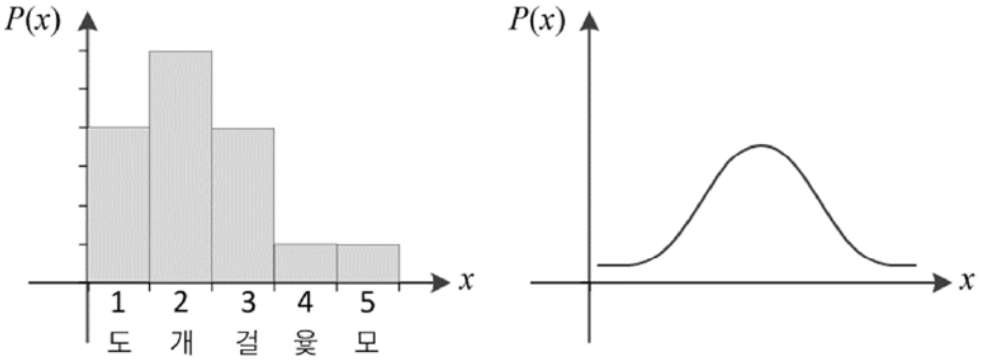
그림 2-13 윷을 던졌을 때 나올 수 있는 다섯 가지 경우(왼쪽부터 도, 개, 걸, 윷, 모)

  - 다섯 가지 경우 중 한 값을 갖는 확률변수 x
  - x의 정의역은 {도, 개, 걸, 윷, 모}

Chapter2.2 – 확률과 통계

- 확률분포

$$P(x = \text{도}) = \frac{4}{16}, P(x = \text{개}) = \frac{6}{16}, P(x = \text{걸}) = \frac{4}{16}, P(x = \text{윷}) = \frac{1}{16}, P(x = \text{모}) = \frac{1}{16}$$



(a) 이산인 경우의 확률질량함수 (b) 연속인 경우의 확률밀도함수

그림 2-14 확률분포

- 확
  - 예) Iris에서 확률벡터 $\mathbf{x}$는 4차원 $\mathbf{x} = (x_1, x_2, x_3, x_4)^{\mathrm{T}}$

  = (꽃받침 길이, 꽃받침 너비, 꽃잎 길이, 꽃잎 너비)^T

Chapter2.2 – 확률과 통계

- 간단한 확률실험 장치
  - 주머니에서 번호를 뽑은 다음, 번호에 따라 해당 병에서 공을 뽑고 색을 관찰함
  - 번호를 y, 공의 색을 x라는 확률변수로 표현하면 정의역은 y∈{①,②,③}, x∈{파랑, 하양}

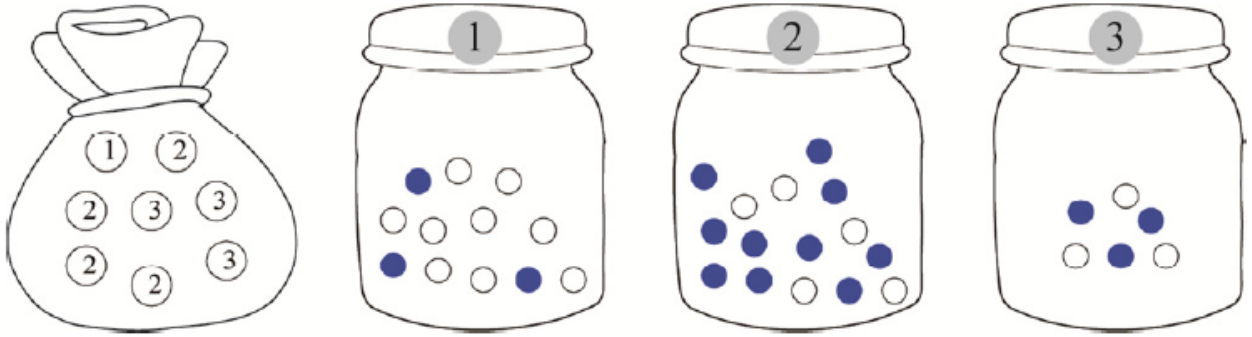


그림 2–15 확률 실험

Chapter2.2 – 확률과 통계

- 곱 규칙과 합 규칙
  - ①번 카드를 뽑을 확률은 $P(y = ①) = P(①) = 1 / 8$
  - 카드는 ①번, 공은 하양일 확률은 $P(y = ①, x = \text{하양}) = P(①, \text{하양})$ ← 결합확률

$$P(y = ①, x = \text{하양}) = P(x = \text{하양}|y = ①)P(y = ①) = \frac{9}{12}\frac{1}{8} = \frac{3}{32}$$

  - 곱 규칙

$$\text{곱 규칙: } P(y, x) = P(x|y)P(y) \qquad (2.23)$$

  - 하얀 공이 뽑힐 확률

$$P(\text{하양}) = P(\text{하양}|①)P(①) + P(\text{하양}|②)P(②) + P(\text{하양}|③)P(③)$$
$$= \frac{9}{12}\frac{1}{8} + \frac{5}{15}\frac{4}{8} + \frac{3}{6}\frac{3}{8} = \frac{43}{96}$$

  - 합 규칙

$$\text{합 규칙: } P(x) = \sum_y P(y, x) = \sum_y P(x|y)P(y) \qquad (2.24)$$

Chapter2.2 – 확률과 통계

- 베이즈 정리 (식 (2.26))

$$P(y,x) = P(x|y)P(y) = P(x,y) = P(y|x)P(x)$$

$$\longrightarrow \quad P(y|x) = \frac{P(x|y)P(y)}{P(x)} \qquad (2.26)$$

  - 다음 질문을 식 (2.27)로 쓸 수 있음

    "하얀 공이 나왔다는 사실만 알고 어느 병에서 나왔는지 모르는데, 어느 병인지 추정하라."

$$\hat{y} = \underset{y}{\operatorname{argmax}} P(y|x) \qquad (2.27)$$

Chapter2.2 – 확률과 통계

- 베이즈 정리 (식 (2.26))
  - 베이즈 정리를 적용하면,

$$\hat{y} = \underset{y}{\operatorname{argmax}} P(y|x = \text{하양}) = \underset{y}{\operatorname{argmax}} \frac{P(x = \text{하양}|y)P(y)}{P(x = \text{하양})}$$

  - 세 가지 경우에 대해 확률을 계산하면,

$$P(①|\text{하양}) = \frac{P(\text{하양}|①)P(①)}{P(\text{하양})} = \frac{\frac{9}{12}\frac{1}{8}}{\frac{43}{96}} = \frac{9}{43}$$

$$P(②|\text{하양}) = \frac{P(\text{하양}|②)P(②)}{P(\text{하양})} = \frac{\frac{5}{15}\frac{4}{8}}{\frac{43}{96}} = \frac{16}{43} \longrightarrow$$ ③번 병일 확률이 가장 높음

$$P(③|\text{하양}) = \frac{P(\text{하양}|③)P(③)}{P(\text{하양})} = \frac{\frac{3}{6}\frac{3}{8}}{\frac{43}{96}} = \frac{18}{43}$$

- 베이즈 정리의 해석

$$\overbrace{P(y|x)}^{\text{사후확률}} = \frac{\overbrace{P(x|y)}^{\text{우도}}\ \overbrace{P(y)}^{\text{사전확률}}}{P(x)}$$

Chapter2.2 – 확률과 통계

- 기계 학습에 적용
  - 예) Iris 데이터 분류 문제
    - 특징 벡터 $\mathbf{x}$, 부류 $y \in$ {setosa, versicolor, virginica}
    - 분류 문제를 argmax로 표현하면 식 (2.29)

$$\hat{y} = \underset{y}{\operatorname{argmax}} P(y|\mathbf{x}) \qquad (2.29)$$

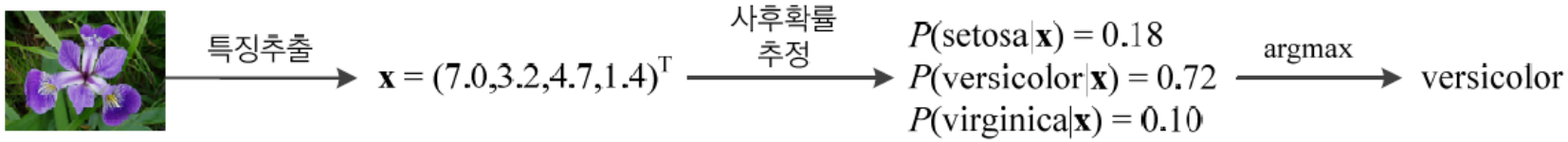


그림 2-16 붓꽃의 부류 예측 과정

  - 사
  - 따라서 베이즈 정리를 이용하여 추정함
    - 사전확률은 식 (2.30)으로 추정
    - 우도는 6.4절의 밀도 추정 기법으로 추정

사전확률: $P(y = c_i) = \frac{n_i}{n}$ (2.30)

Chapter2.2 – 확률과 통계

- 매개변수 Θ를 모르는 상황에서 매개변수를 추정하는 문제

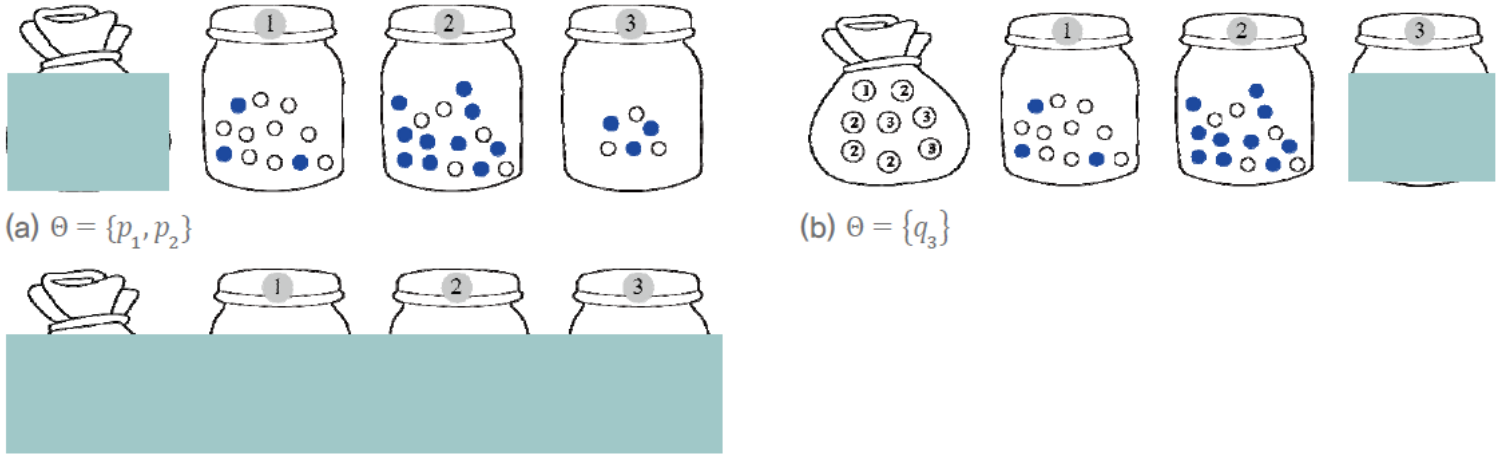


그림 2–17 매개변수가 감추어진 여러 가지 상황

- 예) [그림 2-17(b)] 상황

  데이터집합 $\mathbb{X}$={●○○●○●○○●●○○}

"데이터 $\mathbb{X}$가 주어졌을 때, $\mathbb{X}$를 발생시켰을 가능성을 최대로 하는 매개변수 $\Theta = \{q_3\}$의 값을 찾아라."

Chapter2.2 – 확률과 통계

- 최대 우도법
  - [그림 2-17(b)] 문제를 수식으로 쓰면,

$$\hat{q}_3 = \underset{q_3}{\operatorname{argmax}} P(\mathbb{X}|q_3) \qquad (2.31)$$

  - 일반화 하면,

$$\text{최대 우도 추정:} \quad \hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} P(\mathbb{X}|\Theta) \qquad (2.32)$$

  - 수치 문제를 피하기 위해 로그 표현으로 바꾸면,

$$\text{최대 로그우도 추정:} \quad \hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \log P(\mathbb{X}|\Theta) = \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{n} \log P(\mathbf{x}_i|\Theta) \qquad (2.34)$$

Chapter2.2 – 확률과 통계

- 데이터의 요약 정보로서 평균과 분산

$$\left.\begin{aligned} \text{평균}\ \ \mu &= \frac{1}{n}\sum_{i=1}^{n} x_i \\ \text{분산}\ \ \sigma^2 &= \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2 \end{aligned}\right\} \qquad (2.36)$$

- 평균 벡터와 공분산 행렬

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \qquad (2.37)$$

$$\boldsymbol{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})(\mathbf{x}_i-\boldsymbol{\mu})^{\mathrm{T}} \qquad (2.39)$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_{22} & & \sigma_{2d} \\ \vdots & & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_{dd} \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & & \sigma_{2d} \\ \vdots & & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{pmatrix}$$

Chapter2.2 – 확률과 통계

- 공분산(covariance): 어떤 스칼라(scalar)인 두 확률 변수 X, Y가 있을 때 두 변수 사이의 상관관계

$$
\begin{aligned}
Cov(X, Y) &= E[(X - E[X])(Y - E[Y])] \\
&= E[XY - XE[Y] - YE[X] + E[X]E[Y]] \\
&= E[XY] - E[X]E[Y] - E[Y]E[X] + E[X]E[Y] \\
\therefore Cov(X, Y) &= E[XY] - E[X]E[Y]
\end{aligned}
$$

출처: https://m.blog.naver.com/PostView.naver?blogId=waterforall&logNo=222789143718&categoryNo=95&proxyReferer=

Chapter2.2 – 확률과 통계

- 공분산의 성질

$$[1] \quad Cov(X, Y) = Cov(Y, X)$$

$$[2] \quad Cov(X, a) = 0 \quad , \quad (a = constant)$$

$$[3] \quad Cov(X, X) = Var(X) \geq 0$$

$$[4] \quad Cov(X_1 + X_2, Y) = Cov(X_1, Y) + Cov(X_2, Y)$$

$$[5] \quad Cov(aX, Y) = aCov(X, Y)$$

출처: https://m.blog.naver.com/PostView.naver?blogId=waterforall&logNo=222789143718&categoryNo=95&proxyReferer=

Chapter2.2 – 확률과 통계

- 공분산 행렬(covariance matrix)

$$Cov(X,X) = E[(X - E[X])(X - E[X])^T] = E[XX^T] - E[X]E[X^T]$$

$$Cov(X,X) = \begin{bmatrix} Cov(X_1,X_1) & Cov(X_1,X_2) & \cdots & Cov(X_1,X_n) \\ Cov(X_2,X_1) & Cov(X_2,X_2) & \cdots & Cov(X_2,X_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_n,X_1) & Cov(X_n,X_2) & \cdots & Cov(X_n,X_n) \end{bmatrix}$$

$$Cov(X,X) = \begin{bmatrix} Var(X_1) & Cov(X_1,X_2) & \cdots & Cov(X_1,X_n) \\ Cov(X_2,X_1) & Var(X_2) & \cdots & Cov(X_2,X_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_n,X_1) & Cov(X_n,X_2) & \cdots & Var(X_n) \end{bmatrix}$$

출처: https://m.blog.naver.com/PostView.naver?blogId=waterforall&logNo=222789143718&categoryNo=95&proxyReferer=

Chapter2.2 – 확률과 통계

- 평균 벡터와 공분산 행렬 예제

예제 2-7

Iris 데이터베이스의 샘플 중 8개만 가지고 공분산 행렬을 계산하자.

$$\mathbb{X} = \{\mathbf{x}_1 = \begin{pmatrix}5.1\\3.5\\1.4\\0.2\end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix}4.9\\3.0\\1.4\\0.2\end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix}4.7\\3.2\\1.3\\0.2\end{pmatrix}, \mathbf{x}_4 = \begin{pmatrix}4.6\\3.1\\1.5\\0.2\end{pmatrix}, \mathbf{x}_5 = \begin{pmatrix}5.0\\3.6\\1.4\\0.2\end{pmatrix}, \mathbf{x}_6 = \begin{pmatrix}5.4\\3.9\\1.7\\0.4\end{pmatrix}, \mathbf{x}_7 = \begin{pmatrix}4.6\\3.4\\1.4\\0.3\end{pmatrix}, \mathbf{x}_8 = \begin{pmatrix}5.0\\3.4\\1.5\\0.2\end{pmatrix}\}$$

먼저 평균벡터를 구하면 $\boldsymbol{\mu} = (4.9125,\ 3.3875,\ 1.45,\ 0.2375)^{\mathrm{T}}$이다. 첫 번째 샘플 $\mathbf{x}_1$을 식 (2.39)에 적용하면 다음과 같다.

$$(\mathbf{x}_1 - \boldsymbol{\mu})(\mathbf{x}_1 - \boldsymbol{\mu})^{\mathrm{T}} = \begin{pmatrix}0.1875\\0.1125\\-0.05\\-0.0375\end{pmatrix}(0.1875 \quad 0.1125 \quad -0.05 \quad -0.0375)$$

$$= \begin{pmatrix}0.0325 & 0.0211 & -0.0094 & -0.0070\\0.0211 & 0.0127 & -0.0056 & -0.0042\\-0.0094 & -0.0056 & 0.0025 & 0.0019\\-0.0070 & -0.0042 & 0.0019 & 0.0014\end{pmatrix}$$

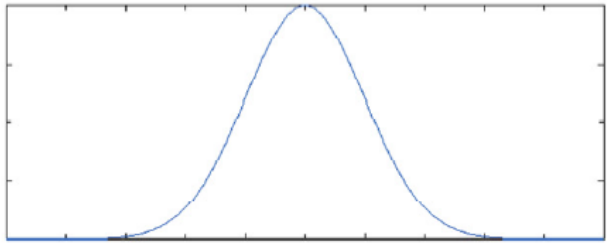
나머지 7개 샘플도 같은 계산을 한 다음, 결과를 모두 더하고 8로 나누면 다음과 같은 공분산 행렬을 얻는다.

$$\boldsymbol{\Sigma} = \begin{pmatrix}0.0661 & 0.0527 & 0.0181 & 0.0083\\0.0527 & 0.0736 & 0.0181 & 0.0130\\0.0181 & 0.0181 & 0.0125 & 0.0056\\0.0083 & 0.0130 & 0.0056 & 0.0048\end{pmatrix}$$

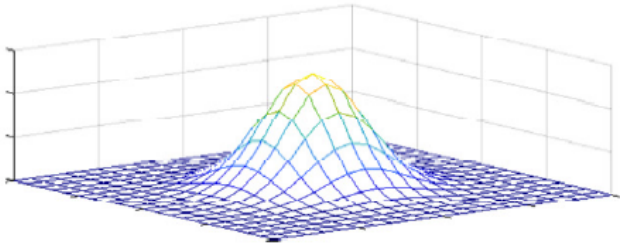
Chapter2.2 – 확률과 통계

- 가우시안 분포
  - 평균 μ와 분산 σ^2으로 정의

$$N(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$



(a) 1차원



(b) 2차원

그림 2–19 가우시안 분포

  - 다차원 가우시안 분포: 평균벡터 $\boldsymbol{\mu}$와 공분산행렬 $\boldsymbol{\Sigma}$로 정의

$$N(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{|\boldsymbol{\Sigma}|}\sqrt{(2\pi)^d}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Chapter2.2 – 확률과 통계

- 베르누이 분포
  - 성공($x = 1$) 확률 p이고 실패($x = 0$) 확률이 1 - p인 분포

$$Ber(x;p) = p^x(1-p)^{1-x} = \begin{cases} p, & x = 1\text{일 때} \\ 1-p, & x = 0\text{일 때} \end{cases}$$

- 이항 분포
  - 성공 확률이 p인 베르누이 실험을 m번 수행할 때 성공할 횟수의 확률분포

$$B(x;m,p) = C_m^x p^x (1-p)^{m-x} = \frac{m!}{x!\,(m-x)!} p^x (1-p)^{m-x}$$

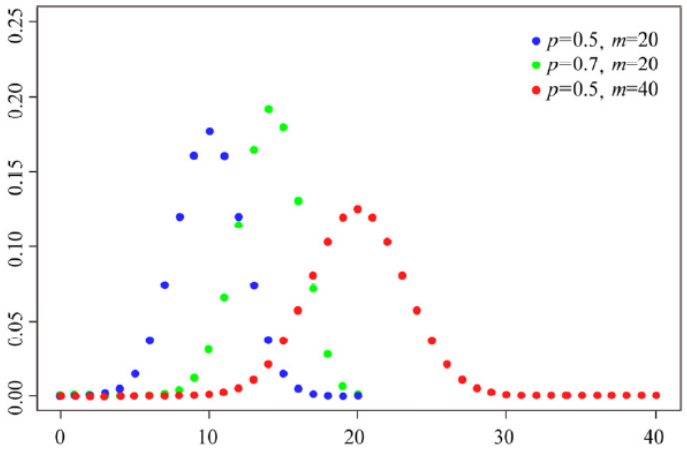


그림 2-20 이항 분포

Chapter2.2 – 확률과 통계

- 메시지가 지닌 정보를 수량화할 수 있나?
  - “고비 사막에 눈이 왔다”와 “대관령에 눈이 왔다”라는 두 메시지 중 어느 것이 더 많은 정보를 가지나?
  - 정보이론의 기본 원리 → 확률이 작을수록 많은 정보
- 자기 정보self information
  - 사건(메시지) e_i의 정보량 (단위: 비트 또는 나츠)

$$h(e_i) = -\log_2 P(e_i) \quad \text{또는} \quad h(e_i) = -\log_e P(e_i) \tag{2.44}$$

Chapter2.2 – 확률과 통계

- 엔트로피
  - 확률변수 x의 불확실성을 나타내는 엔트로피

이산 확률분포 $H(x) = -\sum_{i=1,k} P(e_i)\log_2 P(e_i)$ 또는 $H(x) = -\sum_{i=1,k} P(e_i)\log_e P(e_i)$ (2.45)

연속 확률분포 $H(x) = -\int_{\mathbb{R}} P(x)\log_2 P(x)$ 또는 $H(x) = -\int_{\mathbb{R}} P(x)\log_e P(x)$ (2.46)

Chapter2.2 – 확률과 통계

- 자기 정보와 엔트로피 예제

예제 2-8

윷을 나타내는 확률변수를 x라 할 때 x의 엔트로피는 다음과 같다.

$$H(x) = -\left(\frac{4}{16}\log_2\frac{4}{16} + \frac{6}{16}\log_2\frac{6}{16} + \frac{4}{16}\log_2\frac{4}{16} + \frac{1}{16}\log_2\frac{1}{16} + \frac{1}{16}\log_2\frac{1}{16}\right) = 2.0306\text{비트}$$

주사위는 눈이 6개인데 모두 1/6이라는 균일한 확률을 가진다. 이 경우 엔트로피를 계산하면 다음과 같다.

$$H(x) = -\left(\frac{1}{6}\log_2\frac{1}{6} + \frac{1}{6}\log_2\frac{1}{6} + \frac{1}{6}\log_2\frac{1}{6} + \frac{1}{6}\log_2\frac{1}{6} + \frac{1}{6}\log_2\frac{1}{6} + \frac{1}{6}\log_2\frac{1}{6}\right) = 2.585\text{ 비트}$$

- 주사위가 윷보다 엔트로피가 높은 이유는?

Chapter2.2 – 확률과 통계

- 교차 엔트로피cross entropy
  - 두 확률분포 P와 Q 사이의 교차 엔트로피

$$H(P,Q) = -\sum_{x} P(x)\log_2 Q(x) = -\sum_{i=1,k} P(e_i)\log_2 Q(e_i) \qquad (2.47)$$

  - 식을 전개하면,

$$H(P,Q) = -\sum_{x} P(x)\log_2 Q(x)$$

$$= -\sum_{x} P(x)\log_2 P(x) + \sum_{x} P(x)\log_2 P(x) - \sum_{x} P(x)\log_2 Q(x)$$

$$= H(P) + \underbrace{\sum_{x} P(x)\log_2 \frac{P(x)}{Q(x)}}_{\text{KL 다이버전스}}$$

Chapter2.2 – 확률과 통계

- KL 다이버전스
  - 식 (2.48)은 P와 Q 사이의 KL 다이버전스
  - 두 확률분포 사이의 거리를 계산할 때 주로 사용

$$KL(P \parallel Q) = \sum_x P(x) \log_2 \frac{P(x)}{Q(x)} \qquad (2.48)$$

- 교차 엔트로피와 KL 다이버전스의 관계

$$P와\ Q의\ 교차\ 엔트로피\ H(P,Q) = H(P) + \sum_x P(x) \log_2 \frac{P(x)}{Q(x)} \qquad (2.49)$$
$$= P의\ 엔트로피 + P와\ Q\ 간의\ KL\ 다이버전스$$

Chapter2.2 – 확률과 통계

- Jensen-Shannon divergence(JSD):
  - KLD를 거리 개념으로 해석할 수 있게 변환

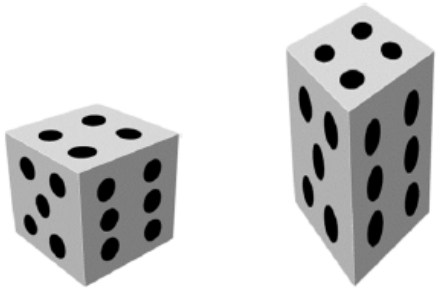
$$\mathrm{JSD}(p,q) = \frac{1}{2}D_{KL}(p||\frac{p+q}{2}) + \frac{1}{2}D_{KL}(q||\frac{p+q}{2})$$

Chapter2.2 – 확률과 통계

예제 2-9

[그림 2-21]과 같이 정상적인 주사위와 찌그러진 주사위가 있는데, 정상적인 주사위의 확률분포는 P, 찌그러진 주사위의 확률분포는 Q를 따르며, P와 Q가 다음과 같이 분포한다고 가정하자.

$$P(1)=\frac{1}{6},\ P(2)=\frac{1}{6},\ P(3)=\frac{1}{6},\ P(4)=\frac{1}{6},\ P(5)=\frac{1}{6},\ P(6)=\frac{1}{6}$$
$$Q(1)=\frac{3}{12},\ Q(2)=\frac{1}{12},\ Q(3)=\frac{1}{12},\ Q(4)=\frac{1}{12},\ Q(5)=\frac{3}{12},\ Q(6)=\frac{3}{12}$$



(a) 정상 주사위 (b) 찌그러진 주사위

그림 2-21 확률분포가 다른 두 주사위

확률분포 P와 Q 사이의 교차 엔트로피와 KL 다이버전스는 다음과 같다.

$$H(P,Q)=-\left(\frac{1}{6}\log_2\frac{3}{12}+\frac{1}{6}\log_2\frac{1}{12}+\frac{1}{6}\log_2\frac{1}{12}+\frac{1}{6}\log_2\frac{1}{12}+\frac{1}{6}\log_2\frac{3}{12}+\frac{1}{6}\log_2\frac{3}{12}\right)=2.7925$$
$$KL(P\parallel Q)=\frac{1}{6}\log_2\frac{2}{3}+\frac{1}{6}\log_2 2+\frac{1}{6}\log_2 2+\frac{1}{6}\log_2 2+\frac{1}{6}\log_2\frac{2}{3}+\frac{1}{6}\log_2\frac{2}{3}=0.2075$$

[예제 2-8]에서 P의 엔트로피 $H(P)$는 2.585이었다. 따라서 식 (2.49)가 성립함을 알 수 있다.

Chapter2.3 최적화

Chapter2.3 – 최적화

- 학습 모델의 매개변수 공간
  - 높은 차원에 비해 훈련집합의 크기가 작아 참인 확률분포를 구하는 일은 불가능함
  - 따라서 기계 학습은 적절한 모델을 선택하고, 목적함수를 정의하고, 모델의 매개변수 공간을 탐색하여 목적함수가 최저가 되는 최적점을 찾는 전략 사용 → 특징 공간에서 해야 하는 일을 모델의 매개변수 공간에서 하는 일로 대치한 셈
  - [그림 2-22]는 여러 예제 (Θ는 매개변수, $J(\Theta)$는 목적함수)

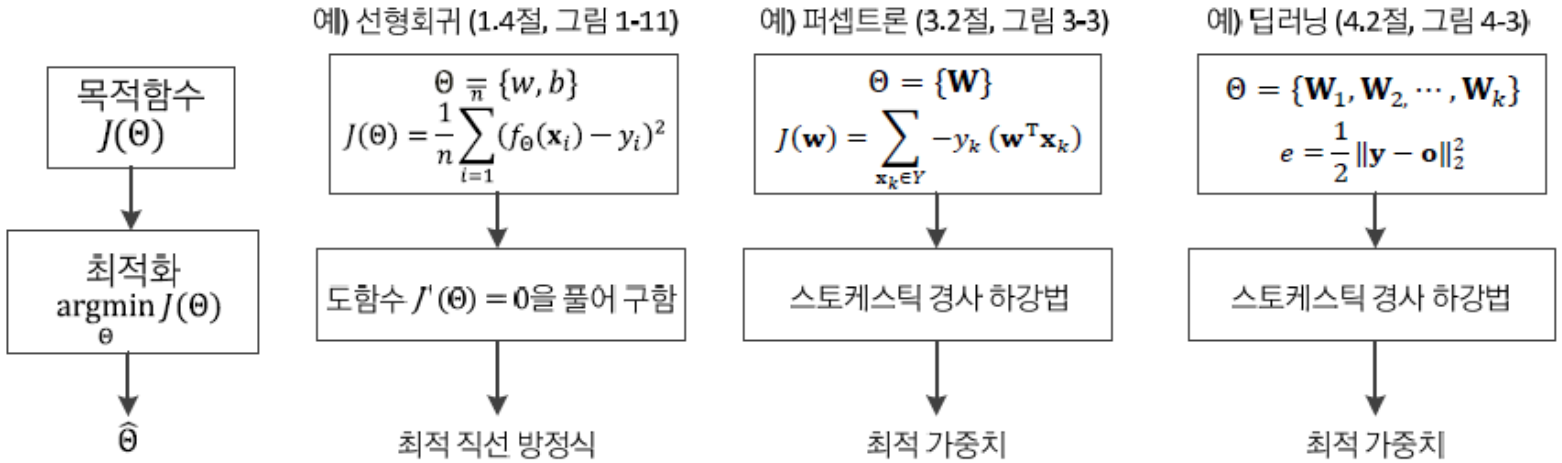
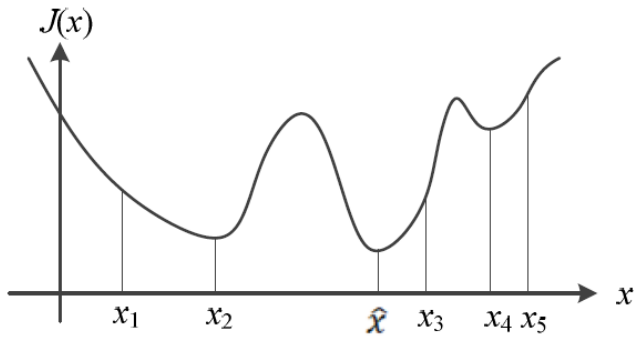


그림 2-22 최적화를 이용한 기계 학습의 문제풀이 과정

Chapter2.3 – 최적화

- 학습 모델의 매개변수 공간
  - 특징 공간보다 수 배~수만 배 넓음
    - [그림 2-22]의 선형회귀에서는 특징 공간은 1차원 , 매개변수 공간은 2차원
    - MNIST 인식하는 딥러닝 모델은 784차원 특징 공간, 수십만~수백만 차원의 매개변수 공간
  - [그림 2-23] 개념도의 매개변수 공간: $\hat{x}$은 전역 최적해, x_2와 x_4는 지역 최적해
  - x_2와 같이 전역 최적해에 가까운 지역 최적해를 찾고 만족하는 경우 많음



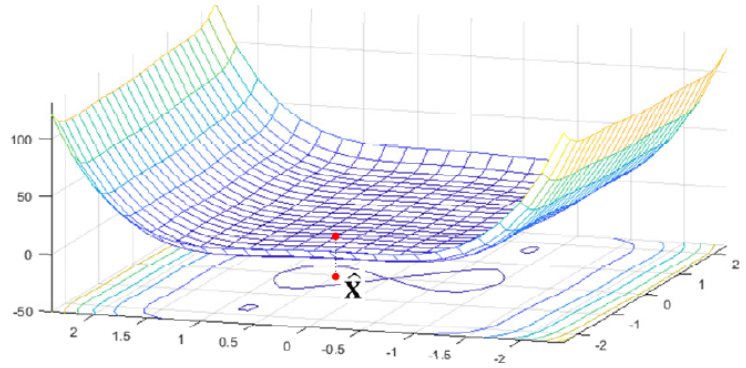


그림 2-23 최적해 탐색

Chapter2.3 – 최적화

- 기계 학습이 해야 할 일을 식으로 정의하면,

$$J(\boldsymbol{\Theta})\text{를 최소로 하는 최적해 } \hat{\boldsymbol{\Theta}}\text{을 찾아라. 즉, } \hat{\boldsymbol{\Theta}} = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} J(\boldsymbol{\Theta}) \qquad (2.50)$$

Chapter2.3 – 최적화

- 최적화 문제 해결
  - 낱낱탐색(완전 탐색)exhaustive search 알고리즘
    - 차원이 조금만 높아져도 적용 불가능
    - 예) 4차원 Iris에서 각 차원을 1000구간으로 나눈다면 총 1000^4개의 점을 평가해야 함

알고리즘 2-1 낱낱탐색 알고리즘

입력 : 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$
출력 : 최적해 $\hat{\boldsymbol{\Theta}}$

```
1  가능한 해를 모두 생성하여 집합 S에 저장한다.
2  min을 충분히 큰 값으로 초기화한다.
3  for (S에 속하는 각 점 Θ_current에 대해)
4      if(J(Θ_current) < min) min=J(Θ_current), Θ_best = Θ_current
5  Θ̂ = Θ_best
```

Chapter2.3 – 최적화

- 무작위 탐색 알고리즘
  - 아무 전략이 없는 순진한 알고리즘

알고리즘 2-2 무작위 탐색 알고리즘

입력 : 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$
출력 : 최적해 $\hat{\boldsymbol{\Theta}}$

1 min을 충분히 큰 값으로 초기화한다.
2 repeat
3 무작위로 해를 하나 생성하고 $\boldsymbol{\Theta}_{current}$라 한다.
4 if($J(\boldsymbol{\Theta}_{current}) < min$) $min = J(\boldsymbol{\Theta}_{current})$, $\boldsymbol{\Theta}_{best} = \boldsymbol{\Theta}_{current}$
5 until(멈춤 조건)
6 $\hat{\boldsymbol{\Theta}} = \boldsymbol{\Theta}_{best}$

Chapter2.3 – 최적화

- [알고리즘 2-3]은 기계 학습이 사용하는 전형적인 알고리즘
  - 라인 3에서는 목적함수가 작아지는 방향을 주로 미분으로 찾아냄

알고리즘 2-3 **기계 학습이 사용하는 전형적인 탐색 알고리즘(1장의 [알고리즘 1-1]과 같음)**

입력 : 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$

출력 : 최적해 $\hat{\boldsymbol{\Theta}}$

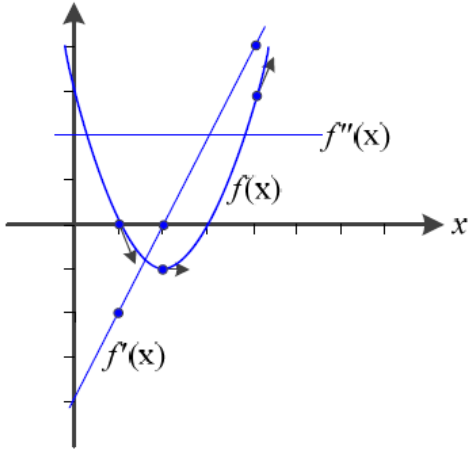
```
1  난수를 생성하여 초기해 Θ을 설정한다.
2  repeat
3      J(Θ)가 작아지는 방향 dΘ를 구한다.
4    Θ = Θ + dΘ
5  until(멈춤 조건)
6  Θ̂ = Θ
```

Chapter2.3 – 최적화

- 미분에 의한 최적화
  - 미분의 정의

$$f'(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}, \qquad f''(x) = \lim_{\Delta x \to 0} \frac{f'(x+\Delta x) - f'(x)}{\Delta x} \tag{2.51}$$

  - 1차 도함수 $f'(x)$는 함수의 기울기, 즉 값이 커지는 방향을 지시함
  - 따라서 $-f'(x)$ 방향에 목적함수의 최저점이 존재
  - [알고리즘 2-3]에서 $d\Theta$로 $-f'(x)$를 사용함← 경사 하강 알고리즘의 핵심 원리



$y = f(x) = x^2 - 4x + 3$

$y' = f'(x) = 2x - 4$

그림 2-24 간단한 미분 예제

Chapter2.3 – 최적화

- 편미분
  - 변수가 여러 개인 함수의 미분
  - 미분값이 이루는 벡터를 그레이디언트라 부름
  - 여러 가지 표기: $\nabla f,\ \dfrac{\partial f}{\partial \mathbf{x}},\ \left(\dfrac{\partial f}{\partial x_1}, \dfrac{\partial f}{\partial x_2}\right)^{\mathrm{T}}$
  - 예)

$$\left.\begin{aligned} f(\mathbf{x}) = f(x_1, x_2) &= \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \\ \nabla f = f'(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}} &= \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right)^{\mathrm{T}} = (2x_1^5 - 8.4x_1^3 + 8x_1 + x_2, 16x_2^3 - 8x_2 + x_1)^{\mathrm{T}} \end{aligned}\right\} \quad (2.52)$$

- 기계 학습에서 편미분
  - 매개변수 집합 Θ에 많은 변수가 있으므로 편미분을 사용

Chapter2.3 – 최적화

- 편미분으로 얻은 그레이디언트에 따라 최저점을 찾아가는 예제

예제 2–10

초기점 $\mathbf{x}_0$=(-0.5,0.5)^{T}라고 하자. $\mathbf{x}_0$에서의 그레이디언트는 $f'(\mathbf{x}_0) = (-2.5125, -2.5)^T$ 즉, $\nabla f|_{\mathbf{x}_0} = (-2.5125, -2.5)^T$이다. [그림 2-25]는 $\mathbf{x}_0$에서 그레이디언트를 화살표로 표시하고 있어, $-f'(\mathbf{x}_0)$은 최저점의 방향을 제대로 가리키는 것을 확인할 수 있다. 하지만 얼마만큼 이동하여 다음 점 $\mathbf{x}_1$로 옮겨갈지에 대한 방안은 아직 없다. 2.3.3절에서 공부하는 경사 하강법은 이에 대한 답을 제공한다.

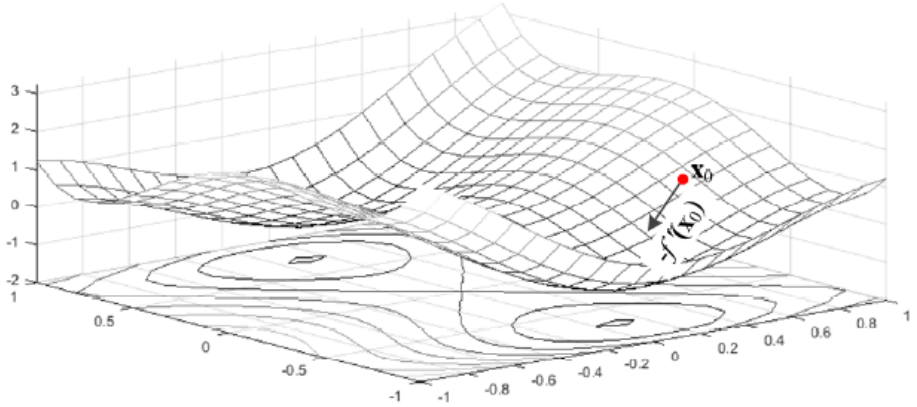


그림 2–25 그레이디언트는 최저점으로 가는 방향을 알려 줌

Chapter2.3 – 최적화

- 독립변수와 종속변수의 구분
  - 식 (1.2)에서 x는 독립변수, y는 종속변수

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

  - 최적화는 예측 단계가 아니라 학습 단계에 필요
    - 식 (1.8)에서 Θ가 독립변수이고 $e = J(\Theta)$라 하면 e가 종속변수임

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

Chapter2.3 – 최적화

- 연쇄법칙
  - 합성함수 $f(x) = g(h(x))$와 $f(x) = g(h(i(x)))$ 의 미분

$$\left.\begin{aligned} f'(x) &= g'(h(x))h'(x) \\ f'(x) &= g'(h(i(x)))h'(i(x))i'(x) \end{aligned}\right\} \quad (2.53)$$

  - 예) $f(x) = 3(2x^2-1)^2 - 2(2x^2-1) + 5$ 일 때 $h(x) = 2x^2 - 1$ 로 두면,

$$f'(x) = \underbrace{(3 * 2(2x^2-1) - 2)}_{g'(h(x))}\underbrace{(2 * 2x)}_{h'(x)} = 48x^3 - 32x$$

- 다층 퍼셉트론은 합성함수
  - $\frac{\partial o_i}{\partial u_{23}^1}$를 계산할 때 연쇄법칙 적용
  - 3.4절(오류 역전파)에서 설명

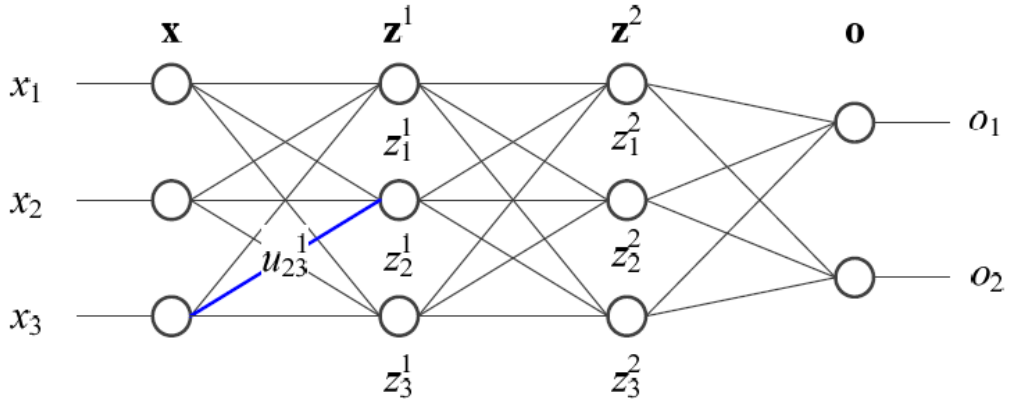


그림 2-26 다층 퍼셉트론은 합성함수

Chapter2.3 – 최적화

- 야코비언 행렬
  - 함수 $\mathbf{f}: \mathbb{R}^d \mapsto \mathbb{R}^m$을 미분하여 얻은 행렬

$$\text{야코비안 행렬 } \mathbf{J} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_d} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_d} \end{pmatrix}$$

예)

$\mathbf{f}: \mathbb{R}^2 \mapsto \mathbb{R}^3$인 $\mathbf{f}(\mathbf{x}) = (2x_1 + x_2^2, -x_1^2 + 3x_2, 4x_1x_2)^{\mathrm{T}}$

$$\mathbf{J} = \begin{pmatrix} 2 & 2x_2 \\ -2x_1 & 3 \\ 4x_2 & 4x_1 \end{pmatrix} \qquad \mathbf{J}|_{(2,1)^{\mathrm{T}}} = \begin{pmatrix} 2 & 2 \\ -4 & 3 \\ 4 & 8 \end{pmatrix}$$

- 헤시안 행렬
  - 2차 편도함수

$$\text{헤시안 행렬 } \mathbf{H} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 x_1} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2 x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \cdots & \frac{\partial^2 f}{\partial x_n x_n} \end{pmatrix}$$

예)

$$f(\mathbf{x}) = f(x_1, x_2)$$
$$= \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

$$\mathbf{H} = \begin{pmatrix} 10x_1^4 - 25.2x_1^2 + 8 & 1 \\ 1 & 48x_2^2 - 8 \end{pmatrix}$$

$$\mathbf{H}|_{(0,1)^{\mathrm{T}}} = \begin{pmatrix} 8 & 1 \\ 1 & 40 \end{pmatrix}$$

Chapter2.3 – 최적화

- 식 (2.58)은 경사 하강법이 낮은 곳을 찾아가는 원리
  - $\mathbf{g} = d\boldsymbol{\Theta} = \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{\Theta}}$이고, ρ는 학습률

$$\boldsymbol{\Theta} = \boldsymbol{\Theta} - \rho \mathbf{g} \qquad (2.58)$$

- 배치 경사 하강 알고리즘
  - 샘플의 그레이디언트를 평균한 후 한꺼번에 갱신

알고리즘 2-4 배치 경사 하강 알고리즘(BGD)

입력 : 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 학습률 ρ
출력 : 최적해 $\hat{\boldsymbol{\Theta}}$

1 난수를 생성하여 초기해 $\boldsymbol{\Theta}$를 설정한다.
2 repeat
3 $\mathbb{X}$에 있는 샘플의 그레이디언트 $\nabla_1, \nabla_2, \cdots, \nabla_n$을 계산한다.
4 $\nabla_{total} = \frac{1}{n}\sum_{i=1,n} \nabla_i$ // 그레이디언트 평균을 계산
5 $\boldsymbol{\Theta} = \boldsymbol{\Theta} - \rho \nabla_{total}$
6 until(멈춤 조건)
7 $\hat{\boldsymbol{\Theta}} = \boldsymbol{\Theta}$

훈련집합

$$\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$$

$$\mathbb{Y} = \{y_1, y_2, \cdots, y_n\}$$

Chapter2.3 – 최적화

- 스토캐스틱 경사 하강SGD(stochastic gradient descent) 알고리즘
  - 한 샘플의 그레이디언트를 계산한 후 즉시 갱신
  - 라인 3~6을 한 번 반복하는 일을 한 세대라 부름

알고리즘 2-5 스토케스틱 경사 하강 알고리즘(SGD)

입력 : 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 학습률 ρ
출력 : 최적해 $\hat{\Theta}$

```
1  난수를 생성하여 초기해 Θ를 설정한다.
2  repeat
3    𝕏의 샘플의 순서를 섞는다.
4    for (i=1 to n)
5          i번째 샘플에 대한 그레이디언트 ∇_i를 계산한다.
6          Θ = Θ − ρ∇_i
7  until(멈춤 조건)
8  Θ̂ = Θ
```

Chapter2.3 – 최적화

- 다른 방식의 구현([알고리즘 2-5]의 라인 3~6을 다음 코드로 대치)

알고리즘 2-5 스토케스틱 경사 하강 알고리즘(SGD)

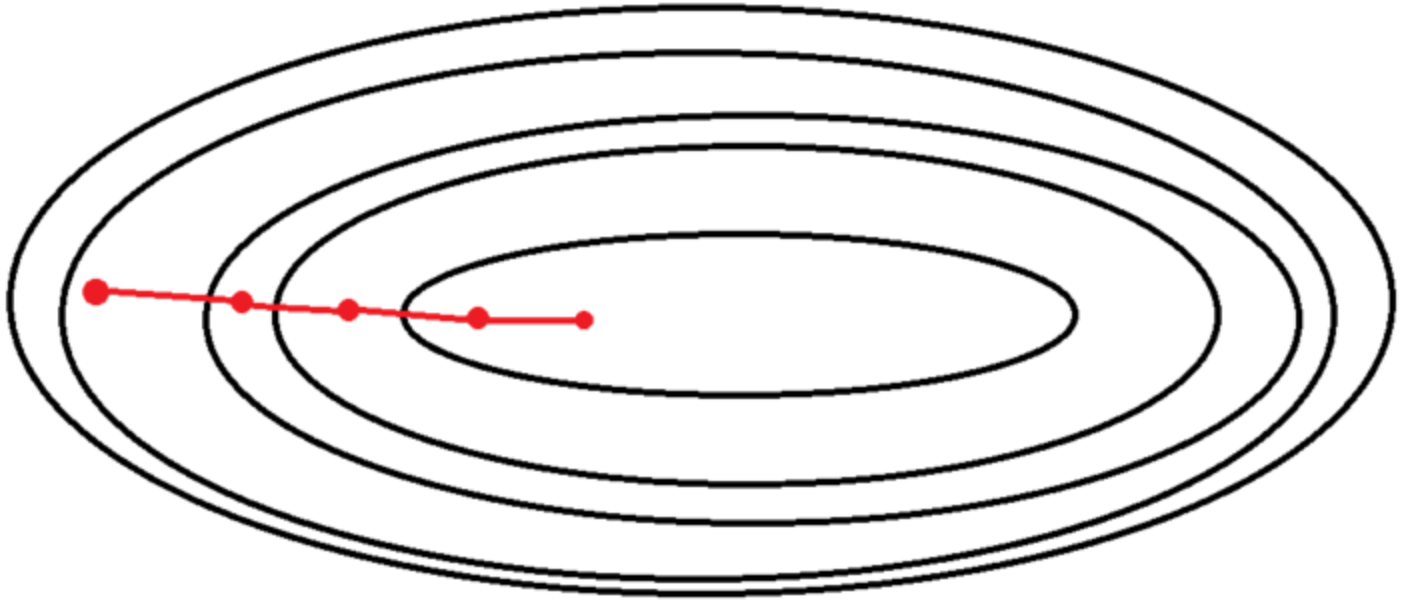
입력 : 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 학습률 ρ
출력 : 최적해 $\hat{\Theta}$

```
1  난수를 생성하여 초기해 Θ를 설정한다.
2  repeat
3    𝕏의 샘플의 순서를 섞는다.
4    for (i=1 to n)
5          i번째 샘플에 대한 그레이디언트 ∇_i를 계산한다.
6          Θ = Θ − ρ∇_i
7  until(멈춤 조건)
8  Θ̂ = Θ
```

```
3  𝕏에서 임의로 샘플 하나를 뽑는다.
4  뽑힌 샘플의 그레이디언트 ∇를 계산한다.
5  Θ = Θ − ρ∇
```

Chapter2.3 – 최적화

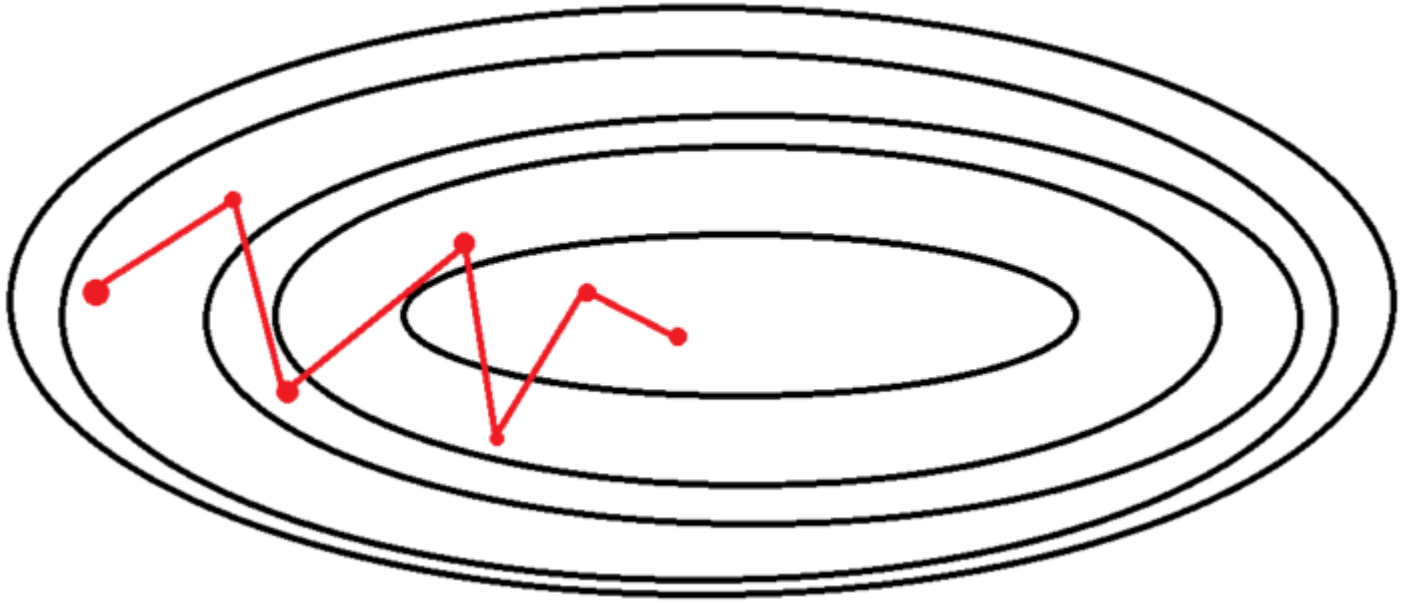
- 배치 경사 하강법 (Batch Gradient Descent: BGD)
  - 전체 데이터를 통해 학습시키기 때문에, 가장 업데이트 횟수가 적다. (1 Epoch 당 1회 업데이트)
  - 전체 데이터를 모두 한 번에 처리하기 때문에, 메모리가 가장 많이 필요하다.
  - 항상 같은 데이터 (전체 데이터)에 대해 경사를 구하기 때문에, 수렴이 안정적이다.



출처: https://skyil.tistory.com/68

Chapter2.3 – 최적화

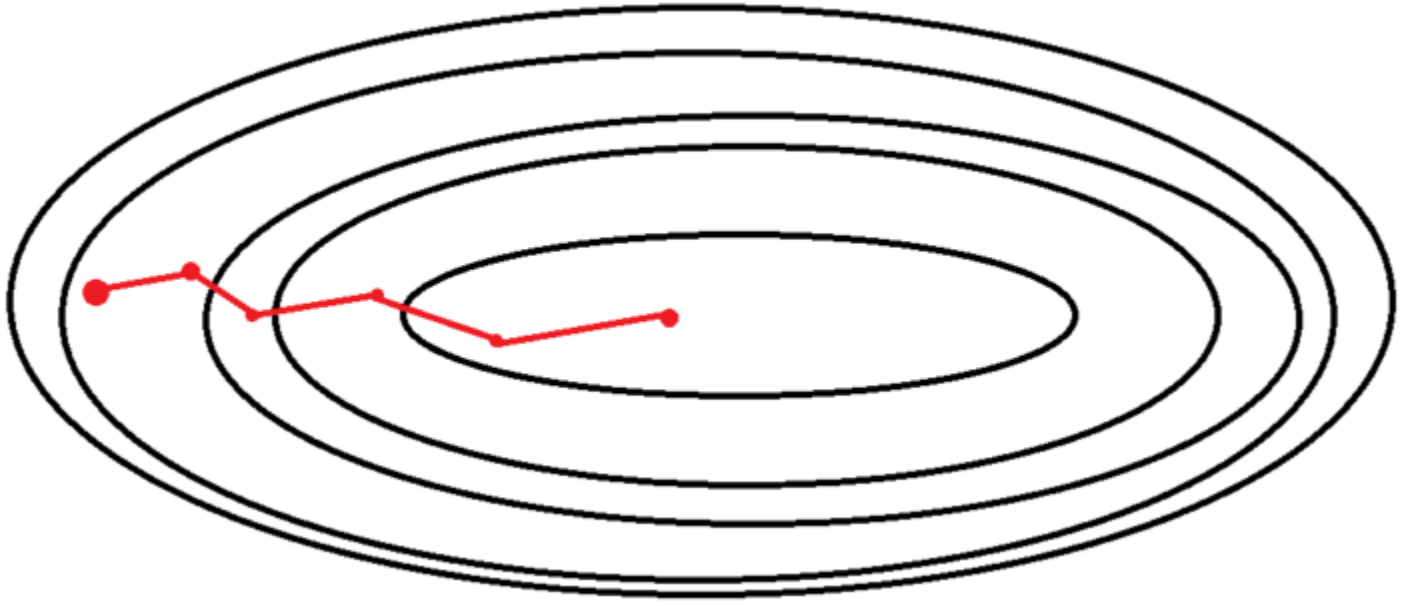
- 확률적 경사 하강법 (Stochastic Gradient Descent: SGD)
  - 한 번에 하나의 데이터를 이용하므로 GPU의 병렬 처리를 그다지 잘 활용하지는 못한다.
  - 1회 학습할 때 계산량이 줄어든다.
  - Global Minimum에 수렴하기 어렵다.
  - 노이즈가 심하다. (Shooting이 너무 심하다.)



출처: https://skyil.tistory.com/68

Chapter2.3 – 최적화

- 미니 배치 확률적 경사 하강법 (Mini-Batch Stochastic Gradient Descent: MSGD)
  - BGD보다 계산량이 적다. (Batch Size에 따라 계산량 조절 가능)
  - Shooting이 적당히 발생한다. (Local Minima를 어느정도 회피할 수 있다.)



출처: https://skyil.tistory.com/68

감사합니다