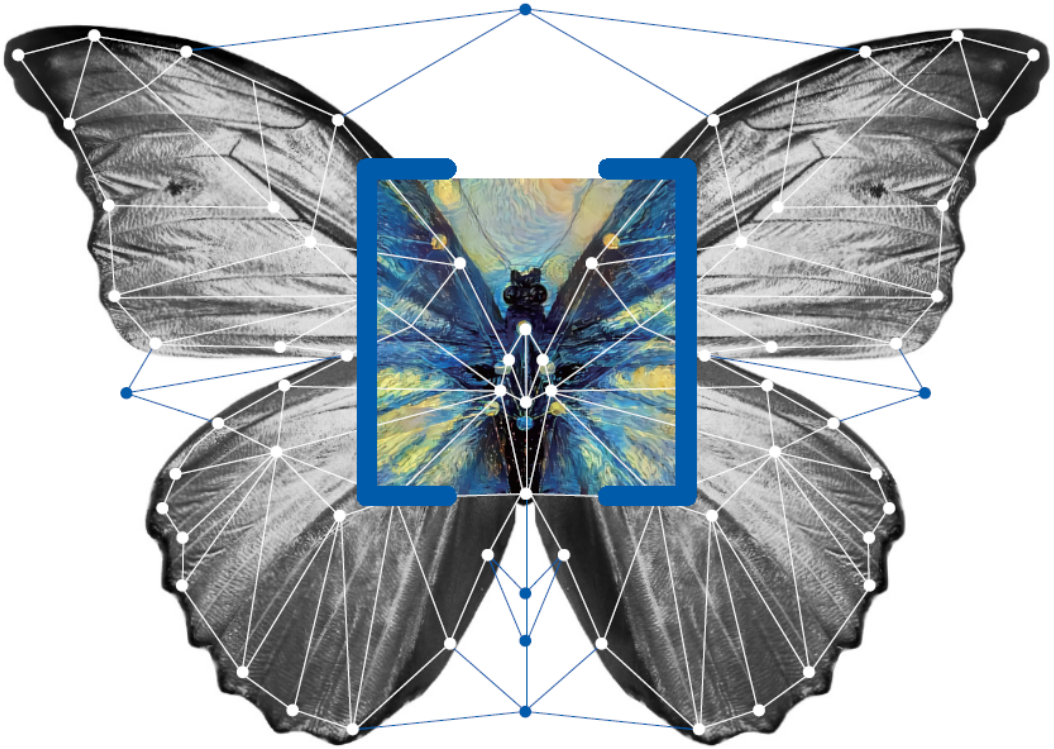
한빛아카데미
Hanbit Academy, Inc.



MACHINE 기계 학습 LEARNING

오일석 지음

2장. 기계 학습과 수학

PREVIEW

기계 학습에서 수학의 역할

- 수학은 목적함수를 정의하고, 목적함수가 최저가 되는 점을 찾아주는 최적화 이론 제공
- 최적화 이론에 규제, 모멘텀, 학습률, 멈춤조건과 같은 제어를 추가하여 알고리즘 구축
- 사람은 알고리즘을 설계하고 데이터를 수집함

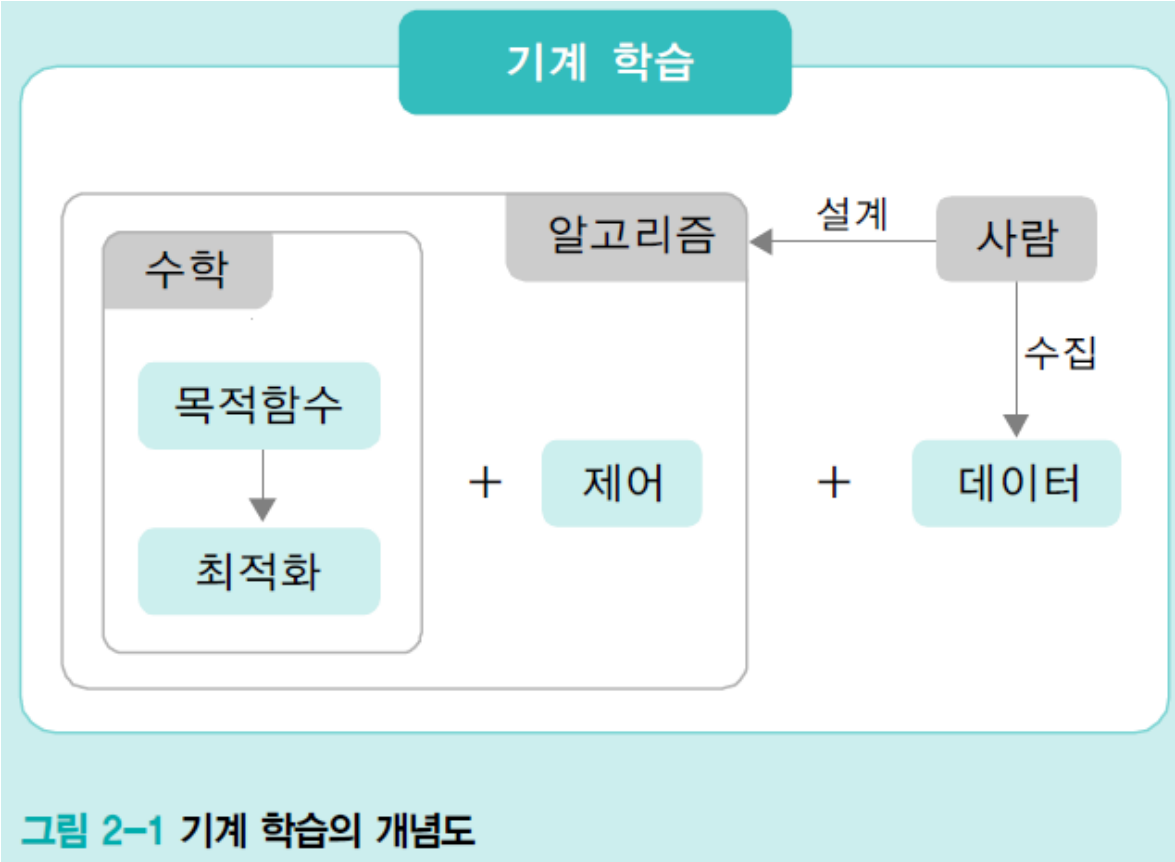


그림 2-1 기계 학습의 개념도

각 절에서 다루는 내용

- 2.1절: 선형대수를 다룬다.
- 2.2절: 확률과 통계를 다룬다.
- 2.3절: 최적화 이론을 다룬다.

- 선형대수: 이 분야의 개념을 이용하면 학습 모델의 매개변수집합, 데이터, 선형연산의 결합 등을 행렬 또는 텐서로 간결하게 표현할 수 있다. 데이터를 분석하여 유용한 정보를 알아내거나 특징 공간을 변환하는 등의 과업을 수행하는 데 핵심 역할을 한다.
- 확률과 통계: 데이터에 포함된 불확실성을 표현하고 처리하는 데 활용한다. 베이즈 이론과 최대 우도 기법을 이용하여 확률 추론을 수행한다.
- 최적화: 목적함수를 최소화하는 최적해를 찾는 데 활용하며, 주로 미분을 활용한 방법을 사용한다. 수학자들이 개발한 최적화 방법을 기계 학습이라는 도메인에 어떻게 효율적으로 적용할지가 주요 관심사이다.

2.1 선형대수

- 2.1.1 벡터와 행렬
- 2.1.2 놈과 유사도
- 2.1.3 퍼셉트론의 해석
- 2.1.4 선형결합과 벡터공간
- 2.1.5 역행렬
- 2.1.6 행렬 분해

2.1.1 벡터와 행렬

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

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

2.1.1 벡터와 행렬

행렬

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

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

← 행 row

↑ 열 column

2.1.1 벡터와 행렬

- 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,

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

2.1.1 벡터와 행렬

- 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

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

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

2.1.1 벡터와 행렬

행렬 연산

- 행렬 곱셈 $\mathbf{C} = \mathbf{AB}$, 이때 $c_{ij} = \sum_{k=1,s} a_{ik} b_{kj}$ (2.1)

2*3 행렬 $\mathbf{A} = \begin{pmatrix} 3 & 4 & 1 \\ 0 & 5 & 2 \end{pmatrix}$와 3*3행렬 $\mathbf{B} = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 0 & 5 \\ 4 & 5 & 1 \end{pmatrix}$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

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

2.1.1 벡터와 행렬

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

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

2.1.2 놈과 유사도

- 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

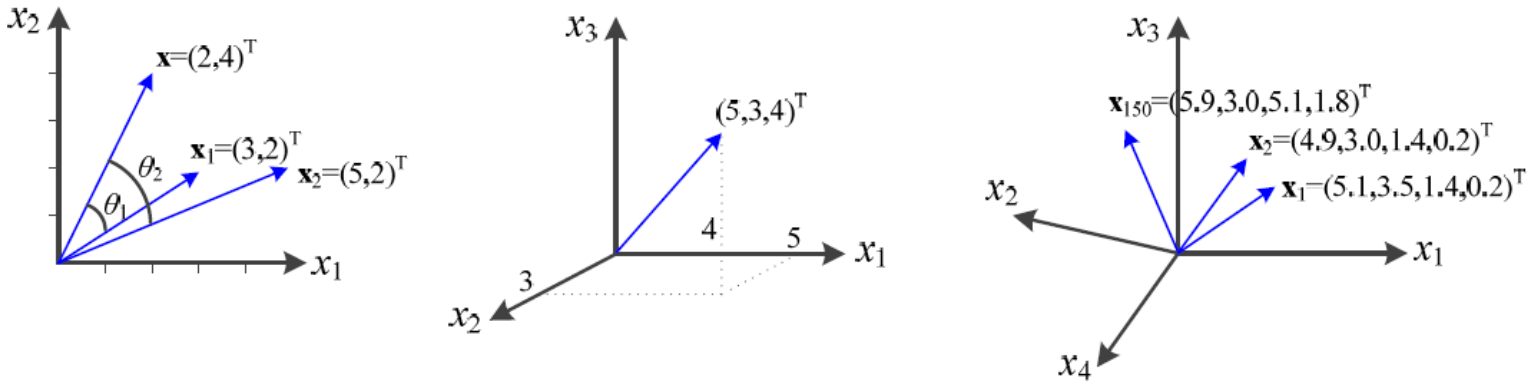
$$\text{프로베니우스 놈: } \|\mathbf{A}\|_F = \left(\sum_{i=1,n} \sum_{j=1,m} a_{ij}^2 \right)^{\frac{1}{2}} \tag{2.6}$$

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

2.1.2 놈과 유사도

유사도와 거리

벡터를 기하학적으로 해석



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

그림 2-2 벡터를 기하학적으로 해석

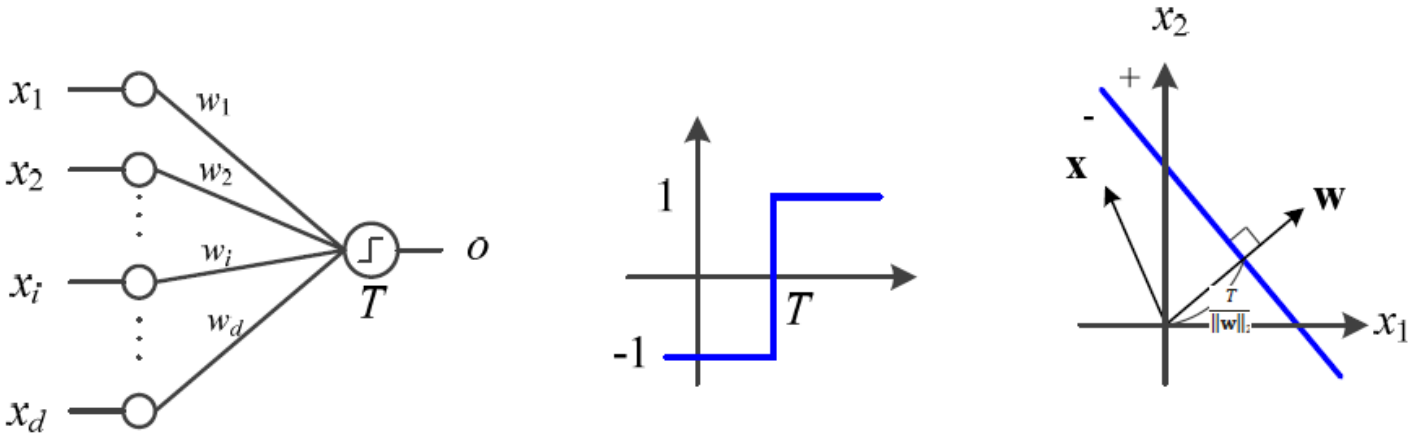
코사인 유사도

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

2.1.3 퍼셉트론의 해석

퍼셉트론

- 1958년 로젠블렛이 고안한 분류기 모델



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

그림 2-3 퍼셉트론의 구조와 동작

- 퍼셉트론의 동작을 수식으로 표현하면,

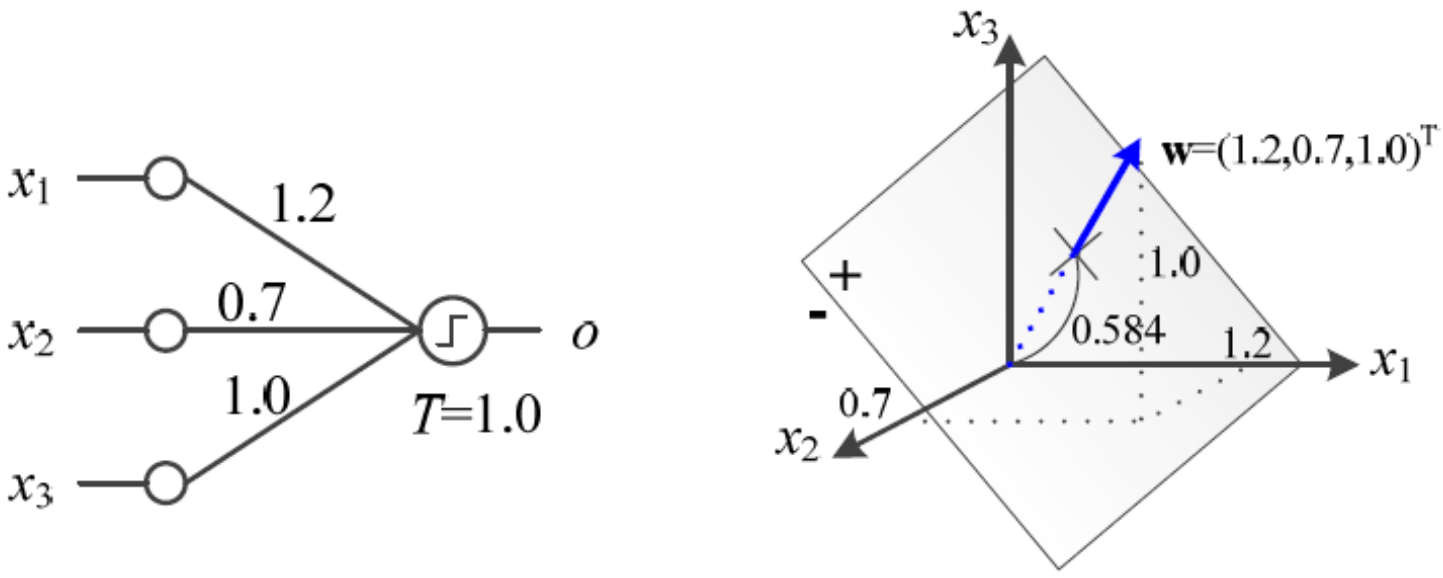
$$o = \tau(\mathbf{w} \cdot \mathbf{x}), \quad \text{이때} \quad \tau(a) = \begin{cases} 1, & a \geq T \\ -1, & a < T \end{cases} \tag{2.8}$$

  - 활성 함수 τ로는 계단함수 사용

2.1.3 퍼셉트론의 해석

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)

2.1.3 퍼셉트론의 해석

출력이 여러 개인 퍼셉트론

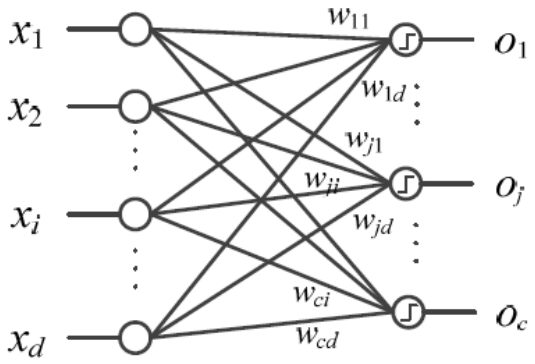


그림 2-5 출력이 여러 개인 퍼셉트론

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

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

- 동작을 수식으로 표현하면,

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

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

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

- 가중치 벡터를 각 부류의 기준 벡터로 간주하면, c개 부류의 유사도를 계산하는 셈

2.1.3 퍼셉트론의 해석

학습의 정의

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

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

- 식 (2.11)은 학습 과정
  - 학습은 훈련집합의 샘플에 대해 식 (2.11)을 가장 잘 만족하는 $\mathbf{W}$를 찾아내는 작업

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

현대 기계 학습에서 퍼셉트론의 중요성

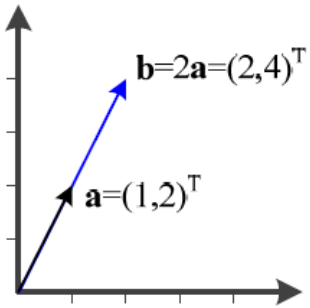
- 딥러닝은 퍼셉트론을 여러 층으로 확장하여 만듦

2.1.4 선형결합과 벡터공간

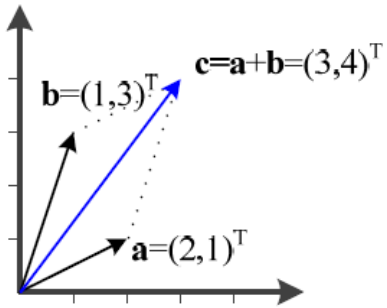
- 벡터
  - 공간상의 한 점으로 화살표 끝이 벡터의 좌표에 해당
- 선형결합이 만드는 벡터공간
  - 기저벡터 $\mathbf{a}$와 $\mathbf{b}$의 선형결합

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

  - 선형결합으로 만들어지는 공간을 벡터공간이라 부름

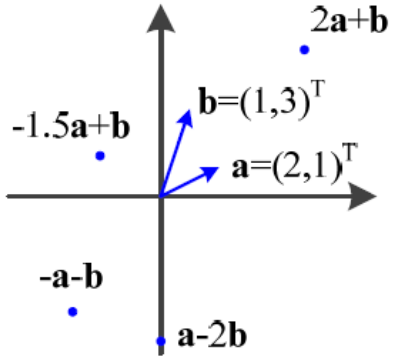


(a) 벡터에 스칼라 곱

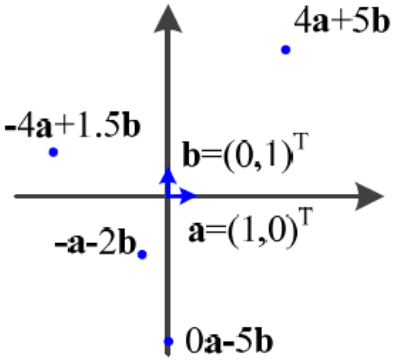


(b) 두 벡터의 덧셈

그림 2-6 벡터의 연산



(a) 기저 벡터와 벡터공간

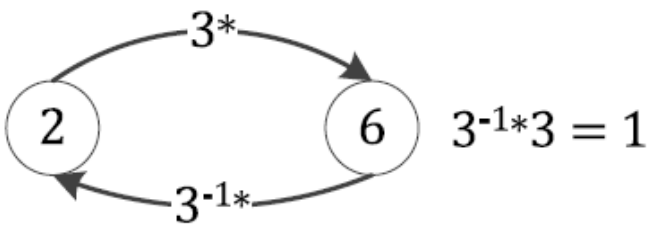


(b) 정규직교 기저 벡터

그림 2-7 벡터공간

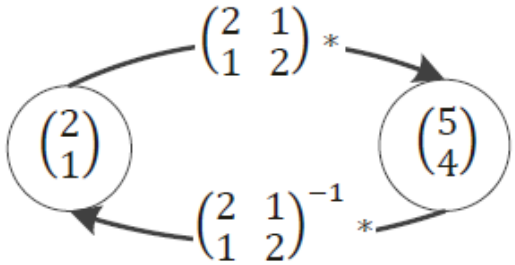
2.1.5 역행렬

역행렬의 원리



$3^{-1}*3 = 1$

(a) 역수의 원리



$\begin{pmatrix}2 & 1\\1 & 2\end{pmatrix}^{-1} * \begin{pmatrix}2 & 1\\1 & 2\end{pmatrix} = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} = \mathbf{I}$

(b) 역행렬의 원리

그림 2-9 역행렬

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

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

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

2.1.5 역행렬

- 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.

2.1.5 역행렬

- 행렬 **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

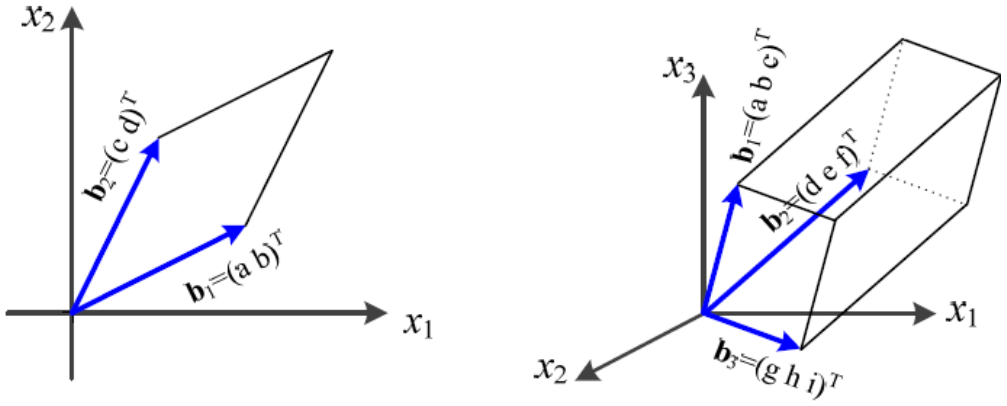


그림 2-10 행렬식의 기하학적 해석

2.1.5 역행렬

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 $(x_1 \quad x_2)\begin{pmatrix}1 & 0\\ 0 & 2\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} = x_1^2 + 2x_2^2$ 이므로

  $\begin{pmatrix}1 & 0\\ 0 & 2\end{pmatrix}$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, $\mathbf{x}^T\mathbf{A}\mathbf{x}\leq 0$

2.1.6 행렬 분해

- 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 $\mathbf{v_1} = \begin{pmatrix}1\\ 1\end{pmatrix}$, $\mathbf{v_2} = \begin{pmatrix}1\\ -1\end{pmatrix}$

2.1.6 행렬 분해

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.

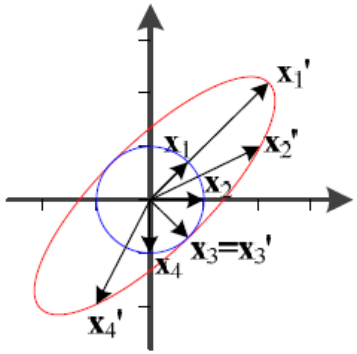


그림 2–12 고유 벡터의 공간 변환

2.1.6 행렬 분해

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

2.1.6 행렬 분해

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

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

  - 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
  - $\mathbf{\Sigma}$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.

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

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

2.2 확률과 통계

- 2.2.1 확률 기초
- 2.2.2 베이즈 정리와 기계 학습
- 2.2.3 최대 우도
- 2.2.4 평균과 분산
- 2.2.5 유용한 확률분포
- 2.2.6 정보이론

- 기계 학습이 처리할 데이터는 불확실한 세상에서 발생하므로, 불확실성을 다루는 확률과 통계를 잘 활용해야 함

2.2.1 확률 기초

확률변수random variable

- 예) 윷

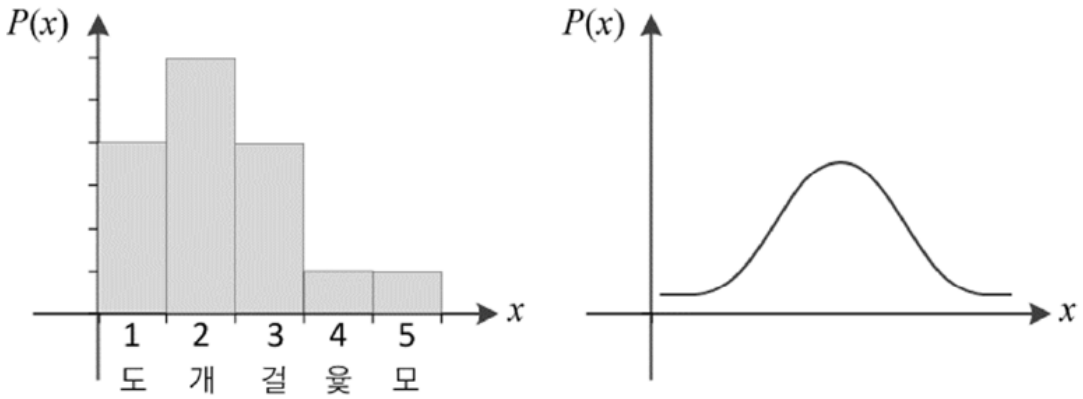
그림 2-13 윷을 던졌을 때 나올 수 있는 다섯 가지 경우(왼쪽부터 도, 개, 걸, 윷, 모)

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

2.2.1 확률 기초

확률분포

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



(a) 이산인 경우의 확률질량함수 (b) 연속인 경우의 확률밀도함수

그림 2-14 확률분포

확률벡터random vector

- 예) Iris에서 확률벡터 $\mathbf{x}$는 4차원 $\mathbf{x} = (x_1, x_2, x_3, x_4)^{\mathrm{T}} = (\text{꽃받침 길이}, \text{꽃받침 너비}_1, \text{꽃잎 길이}, \text{꽃잎 너비})^{\mathrm{T}}$

2.2.1 확률 기초

간단한 확률실험 장치

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

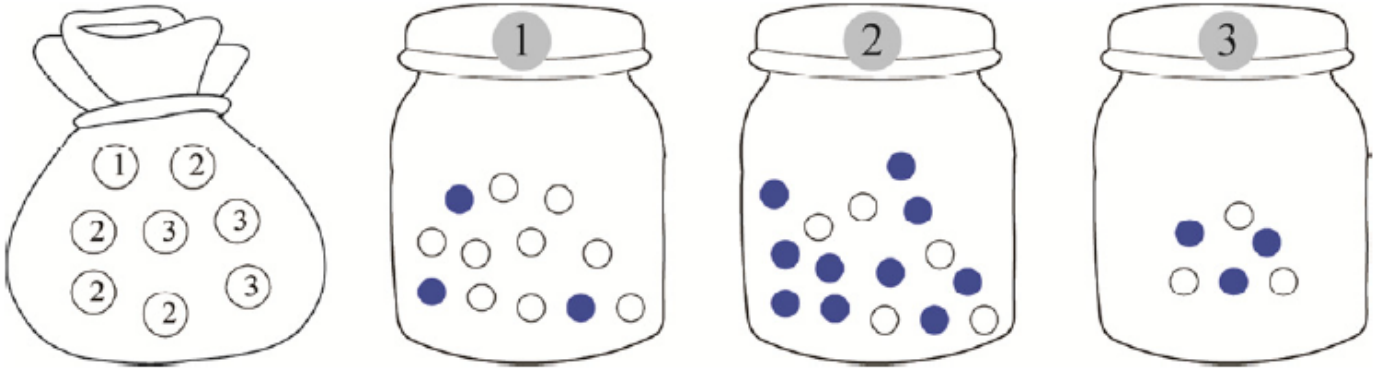


그림 2-15 확률 실험

2.2.1 확률 기초

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

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

- 하얀 공이 뽑힐 확률

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

- 합 규칙

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

2.2.2 베이즈 정리와 기계 학습

- 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."

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

2.2.2 베이즈 정리와 기계 학습

- 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

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

2.2.2 베이즈 정리와 기계 학습

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)

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

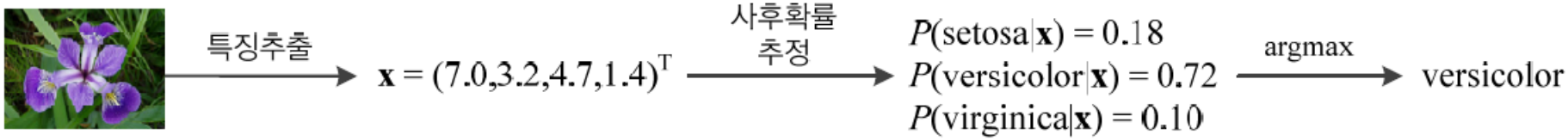


그림 2-16 붓꽃의 부류 예측 과정

- 사후확률 $P(y|\mathbf{x})$를 직접 추정하는 일은 아주 단순한 경우를 빼고 불가능
- 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)

2.2.3 최대 우도

- 매개변수 Θ를 모르는 상황에서 매개변수를 추정하는 문제

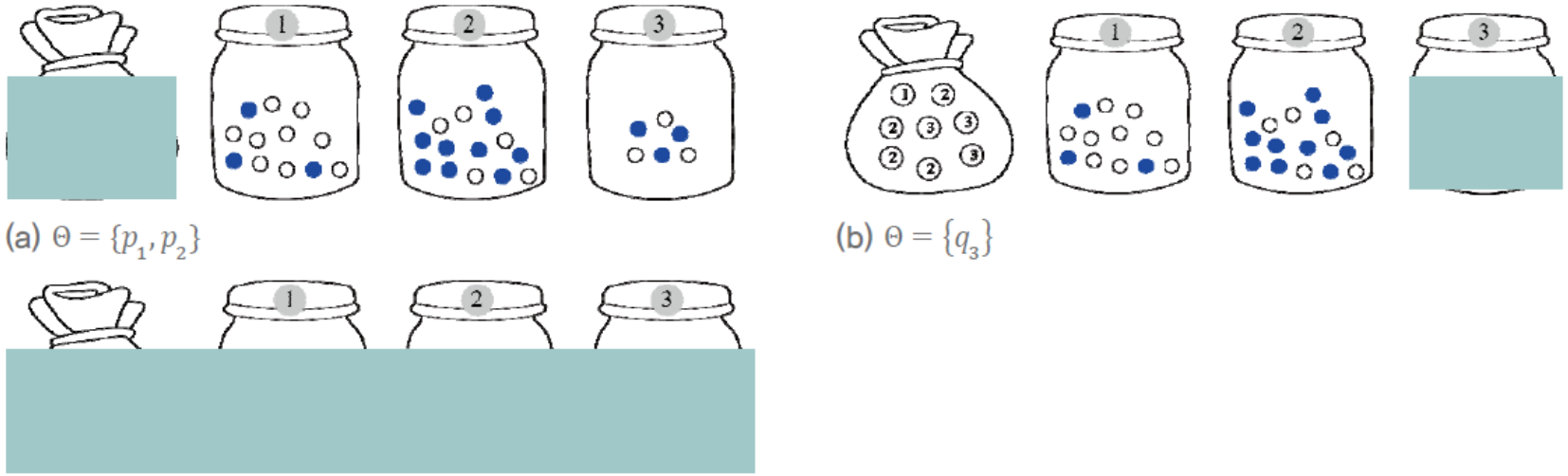


그림 2-17 매개변수가 감추어진 여러 가지 상황

  - 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."

2.2.3 최대 우도

최대 우도법

- [그림 2-17(b)] 문제를 수식으로 쓰면,

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

- 일반화 하면,

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

- 수치 문제를 피하기 위해 로그 표현으로 바꾸면,

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

2.2.4 평균과 분산

- 데이터의 요약 정보로서 평균과 분산

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

- 평균 벡터와 공분산 행렬

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

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

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

2.2.4 평균과 분산

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.

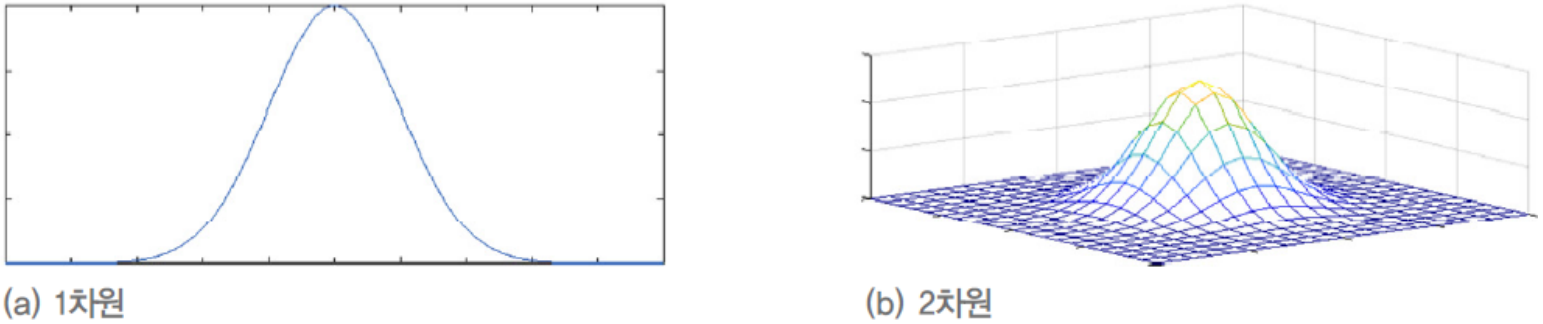
$$\boldsymbol{\Sigma} = \begin{pmatrix}0.0661 & 0.0527 & 0.0181 & 0.0083\\0.0527 & 0.0736 & 0.0181 & 0.0130\\0.0181 & 0.0181 & 0.0125 & 0.0056\\0.0083 & 0.0130 & 0.0056 & 0.0048\end{pmatrix}$$

2.2.5 유용한 확률분포

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

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

2.2.5 유용한 확률분포

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

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

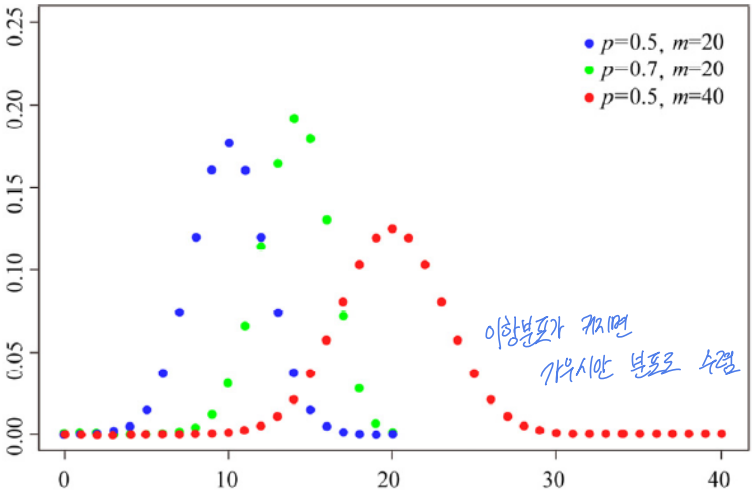


그림 2-20 이항 분포

m 증가

$\sqrt{mp(1-p)}$

이항분포가 커지면 가우시안 분포로 수렴

m이 커지면 가우시안분포로 나타남

2.2.6 정보이론

- 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)

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

- 엔트로피
  - 확률변수 x의 불확실성을 나타내는 엔트로피

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

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

2.2.6 정보이론

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 1/60이라는 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?

2.2.6 정보이론

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,

$$\begin{aligned} H(P,Q) &= -\sum_{x} P(x)\log_2 Q(x) \\ &= -\sum_{x} P(x)\log_2 P(x) + \sum_{x} P(x)\log_2 P(x) - \sum_{x} P(x)\log_2 Q(x) \\ &= H(P) + \underbrace{\sum_{x} P(x)\log_2 \frac{P(x)}{Q(x)}}_{\text{KL 다이버전스}} \end{aligned}$$

2.2.6 정보이론

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

$$P와\ Q의\ 교차\ 엔트로피\ H(P,Q) = H(P) + \sum_{x} P(x) \log_2 \frac{P(x)}{Q(x)} \qquad (2.49)$$

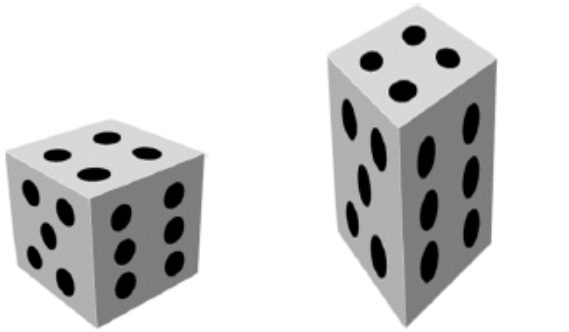
$$= P의\ 엔트로피 + P와\ Q\ 간의\ KL\ 다이버전스$$

2.2.6 정보이론

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.

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

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

2.3 최적화

- 2.3.1 매개변수 공간의 탐색
- 2.3.2 미분
- 2.3.3 경사 하강 알고리즘

- 순수 수학 최적화와 기계 학습 최적화의 차이
  - 순수 수학의 최적화 예) $f(x_1, x_2) = -\left(\cos\left(x_1{}^2\right) + \sin\left(x_2{}^2\right)\right)^2$의 최저점을 찾아라.
  - 기계 학습의 최적화는 단지 훈련집합이 주어지고, 훈련집합에 따라 정해지는 목적함수의 최저점을 찾아야 함
    - 데이터로 미분하는 과정 필요 → 오류 역전파 알고리즘 (3.4절)
    - 주로 SGD(스토캐스틱 경사 하강법) 사용

2.3.1 매개변수 공간의 탐색

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)

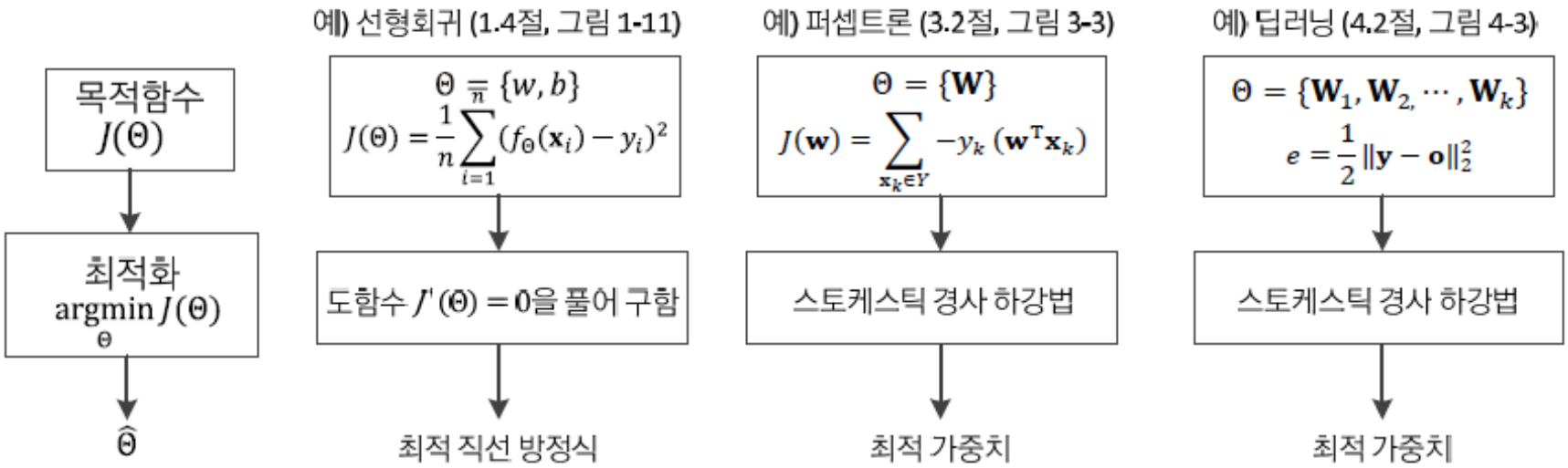


그림 2-22 최적화를 이용한 기계 학습의 문제풀이 과정

2.3.1 매개변수 공간의 탐색

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

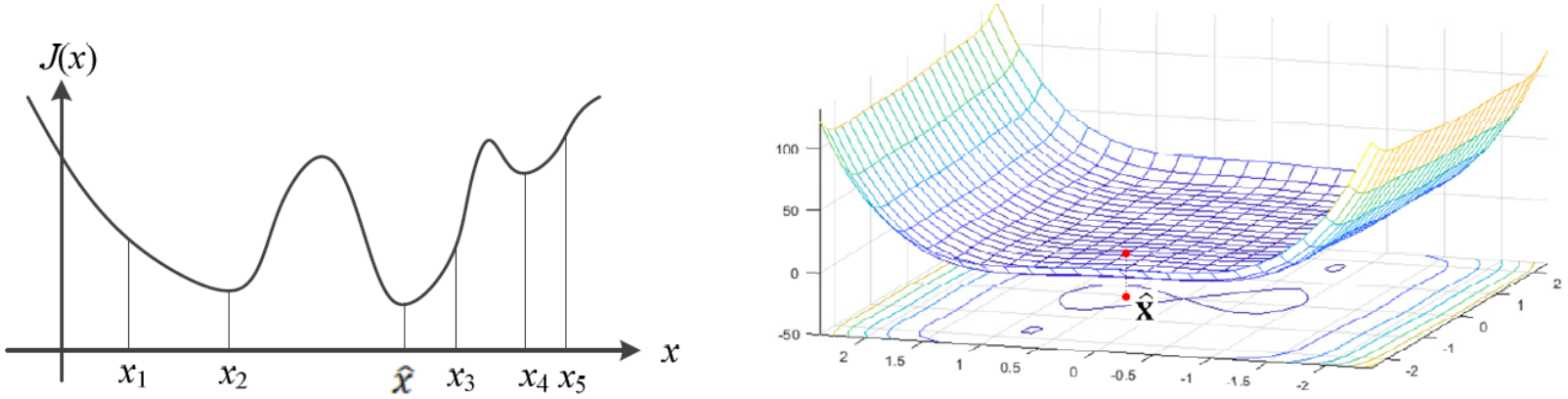


그림 2-23 최적해 탐색

기계 학습이 해야 할 일을 식으로 정의하면,

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

2.3.1 매개변수 공간의 탐색

최적화 문제 해결

- 낱낱탐색exhaustive search 알고리즘
  - 차원이 조금만 높아져도 적용 불가능
  - 예) 4차원 Iris에서 각 차원을 1000구간으로 나눈다면 총 1000^4개의 점을 평가해야 함

- 무작위 탐색 알고리즘
  - 아무 전략이 없는 순진한 알고리즘

알고리즘 2-1 낱낱탐색 알고리즘

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

```
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
```

알고리즘 2-2 무작위 탐색 알고리즘

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

```
1  min을 충분히 큰 값으로 초기화한다.
2  repeat
3     무작위로 해를 하나 생성하고 Θ_current라 한다.
4     if(J(Θ_current) < min) min=J(Θ_current), Θ_best = Θ_current
5  until(멈춤 조건)
6  Θ̂ = Θ_best
```

2.3.1 매개변수 공간의 탐색

- [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 $\boldsymbol{\Theta}$을 설정한다.
2 repeat
3 $J(\boldsymbol{\Theta})$가 작아지는 방향 $d\boldsymbol{\Theta}$를 구한다.
4 $\boldsymbol{\Theta} = \boldsymbol{\Theta} + d\boldsymbol{\Theta}$
5 until(멈춤 조건)
6 $\hat{\boldsymbol{\Theta}} = \boldsymbol{\Theta}$

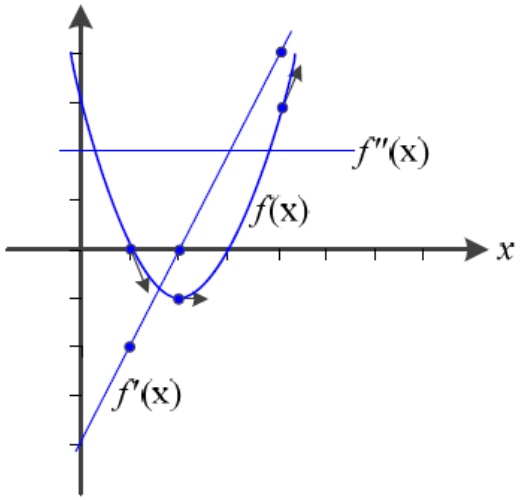
2.3.2 미분

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 $d\mathbf{0}$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

2.3.2 미분

편미분

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

2.3.2 미분

- 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.

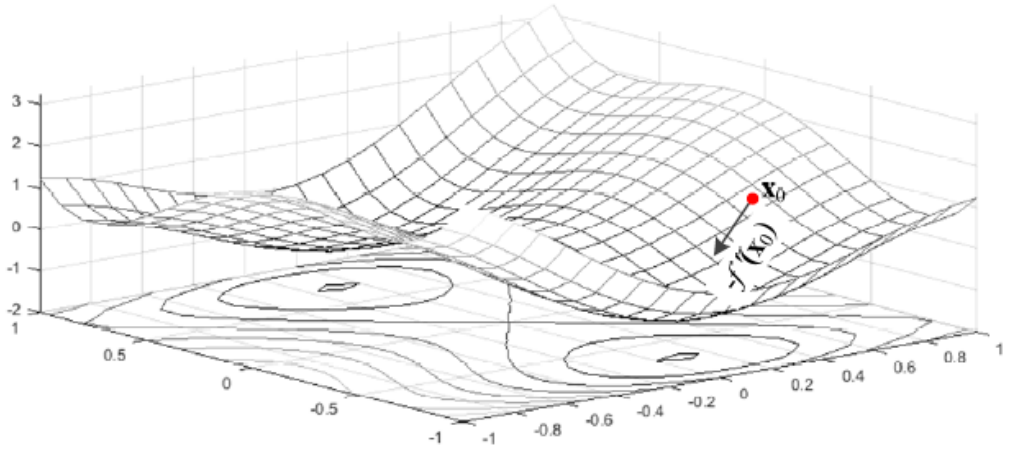


그림 2-25 그레이디언트는 최저점으로 가는 방향을 알려 줌

2.3.2 미분

독립변수와 종속변수의 구분

- 식 (1.2)에서 x는 독립변수, y는 종속변수

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

- 기계 학습에서 이런 해석은 무의미 (왜냐하면 예측 단계를 위한 해석에 불과)
- 최적화는 예측 단계가 아니라 학습 단계에 필요
  - 식 (1.8)에서 Θ가 독립변수이고 $e = J(\Theta)$라 하면 e가 종속변수임

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

  - [그림 2-22]는 여러 가지 사례를 보여줌

2.3.2 미분

연쇄법칙

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

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

- 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

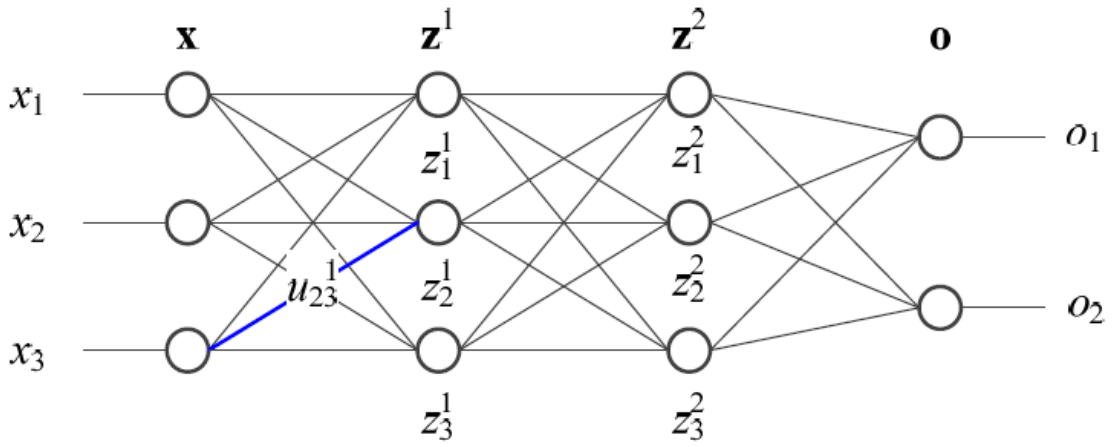


그림 2-26 다층 퍼셉트론은 합성함수

2.3.2 미분

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)

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

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

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

2.3.3 경사 하강 알고리즘

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

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

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

훈련집합

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

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

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 Θ를 설정한다.
2  repeat
3     𝕏에 있는 샘플의 그레이디언트 ∇1, ∇2, ⋯, ∇n을 계산한다.
4     ∇total = (1/n) Σ(i=1,n) ∇i   // 그레이디언트 평균을 계산
5     Θ = Θ − ρ∇total
6  until(멈춤 조건)
7  Θ̂ = Θ
```

2.3.3 경사 하강 알고리즘

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
```

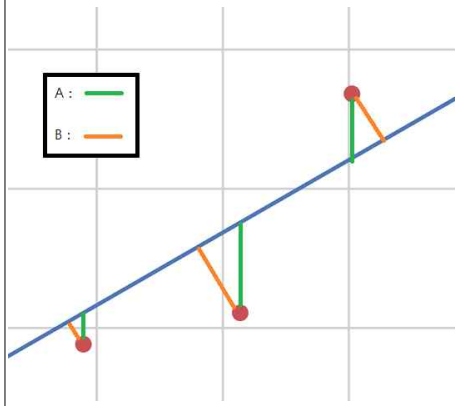
- 다른 방식의 구현([알고리즘 2-5]의 라인 3~6을 다음 코드로 대치)

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

3주차 조별보고서 (Default)	
작성일: 2019년 9월 20일	작성자: 김영연
조 모임 일시: 2019년 9월 20일 9교시	모임장소: 학교 앞 카페
참석자: 위성조, 이충현, 최진성, 이재은, 김영연	조원: 위성조, 이충현, 최진성, 이재은, 김영연

구 분	내 용
학습 범위와 내용	1.3 데이터의 중요성과 희소성 1.4 선형 회귀 1.5 모델 선택의 중요성과 방법 1.6 현대 기계 학습에서 데이터 확대와 가중치 감쇠 1.7 기계 학습의 유형 1.8 기계 학습의 역사와 사회적 의미
논의 내용	저희 Default 조는 개인 레포트/질문을 작성한 뒤에, 금요일에 모여 수업시간에 교수님이 말씀하신 손실함수 계산방식에 대한 논의를 제일 먼저 하였습니다. 그 후에 서로의 질문을 맡아 찾아보고 그에 대한 내용을 서로 얘기하는 방식으로 회의를 진행했습니다.

Q1. 수업 중 손실함수 계산방식에 대한 논의



A1.

1. 먼저 A를 사용하는 것이 효율적이라는 의견은 다음과 같다.

- 손실함수의 계산에서 주로 B대신에 A를 사용하는 이유는 손실함수는 데이터와 모델 사이의 차이를 측정하고, 이를 통해 모델을 어떤 식으로 바꿔야 하는지 결정하는 데 도움이 되기 위한 것이지, 실제 데이터와 모델 사이의 정확한 차이를 구하기 위해서 사용되는 것이 아니므로, 점과 직선 사이의 거리를 구하기 위해 피타고라스 정리를 이용해 제곱과 루트 계산을 하는 것보다 y값의 차이만을 계산하면 되는 비용 효율적인 방법을 택한 것이 아닌가 생각한다.

2. B를 사용하는 것에 대한 의견은 다음과 같다.

실제 값과 예측 값에 대한 오차를 표현할 때, 오차를 B로 사용하면 과연 더 줄어들지 의문이 들었다. 그러나 이는 실제 값과 예측 값에 대한 x값이 같아야 하는데 사선 방향으로 표시되면 x값이 2개가 존재하므로 표시하면 안 된다.

Q2. 진성: 과소적합과 과잉적합은 데이터 집합에 따라 발생할 수 있는 모델의 문제점을 보여주고 있는데, 이를 해결할 방법이 무엇인지에 대해 궁금하다.

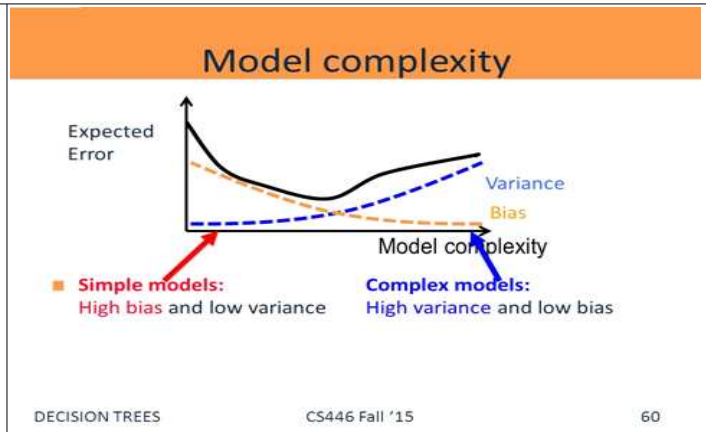
A2.

feature를 많이 반영하면 high variance가 되어 발생하는 과잉적합을 해결하기 위해서는 크게 3가지 방법이 있다. 첫번째로는 정규화로 학습데이터의 일부분만 학습을 시키고, 나머지는 학습을 시키지 않고 테스트에 활용하는 방법이다. 두번째는 교차검증이다. 검증용 데이터를 고정하지 않고 랜덤하게 변경해가며 사용하는 기법이다. 여기서 검증용데이터란 아래의 그림과 같이 훈련 데이터를 나누어 사용하는 것 중 한 종류이다.

세번째는 가지치기이다. 가지치기란 가지치기하듯이 특정 가지 이후를 잘라내어 단순하게 만듦으로써 overfitting을 해결하는 것이다.

반면에, feature를 적게 반영하여 high bias가 되어 발생하는 과소적합을 해결하기 위해서는 무조건 더 많은 데이터를 늘려서는 안 된다. 그것은 오히려 에러를 더 증가 시킬 수 있기 때문에, feature를 더 많이 반영해서 분산을 높여야 한다. 또는, 분산이 높은 기계학습모델

	Decision Tree, KNN, SVM등을 사용하는 방법 등이 있다. 위의 내용을 논의하며, bias과 variance의 trade-off 관계에서 각 학습 모델에 맞는 적절한 지점을 찾는 것이 중요함을 알 수 있다.
	Q3. 재은: 과대적합 문제를 해결하기 위해서는 훈련 데이터를 더 많이 모으거나, 정규화를 거치거나, 훈련 데이터의 잡음을 줄이는 등의 방법이 쓰입니다. 그에 반해, 과소적합 문제는 파라미터가 더 많은 모델을 선택하거나 과대적합이 되기 전까지 학습을 시킴으로써 해결합니다. 그렇다면 과대적합과 과소적합 중 더 안 좋은 문제가 무엇인가요? 각각의 문제에서 해결하는 방식에서는 어떤 방식이 더 효율적인지 알고 싶습니다. A3. 과소적합 및 과대적합 모두 피해야 할 문제임은 분명하나, 굳이 구분하자면 과대적합이 과소적합보다는 나을 수 있다. 그 이유로는 과대적합의 경우 적어도 훈련 데이터셋에 대해서는 높은 성능을 보여주기 때문이다. 과대적합과 과소적합 문제에서 비용문제를 제외한다면 언제나 효율적인 해결책은 훈련 데이터량을 늘리는 것이지만, 비용 문제로 데이터량을 늘리기 힘든 경우, 모델의 복잡도를 우선 조정해 보는 것이 효율적이라고 판단된다.



위 그래프에서 볼 수 있듯이, 모델의 복잡도에 따라 에러율이 달라지는데, 평균적으로 bias와 variance 가 만나는 지점에서 에러율이 가장 낮아지므로, 과소 및 과대적합 모두에 적용할 수 있는 방식이므로, 먼저 모델의 복잡도를 변경해 보는 것이 나을 것이다.

그러나, 모델의 복잡도를 변경한 뒤 훈련을 진행하고 테스트를 해야 하므로, 시간이 많이 걸리기에, 돈과 시간 모두 부족한 상황이라면, 다른 방식을 고민해 봐야 할 것 같다.

Q4 영연: 고차원 공간에 내제된 manifold를 찾는 방법은 무엇인지 궁금합니다.

A4.

학습 데이터 셋에 존재하는 수많은 이미지를 고차원 공간 속에 매핑 시키면 유사한 이미지는 특정 공간에 함께 있을 것이다. 그리고 그 점들의 집합을 잘 아우르는 부분공간이 존재하

	는데 그것이 매니폴드라고 한다. 매니폴드는 다음과 같은 가정을 가지고 있다. -고차원 데이터의 밀도는 낮지만, 이들의 집합을 포함하는 저차원의 매니폴드가 있다. -이 저차원의 매니폴드를 벗어나는 순간 밀도는 급격히 낮아진다. 매니폴드 공간은 본래 고차원 공간의 부분공간이기 때문에 차원수가 상대적으로 작아진다. 이는 데이터 차원 축소가 가능하다고 볼 수 있다. 따라서 차원 축소가 잘 되었다는 것은 매니폴드 공간을 잘 찾았다는 뜻이다. 본래 고차원 공간에서 각 차원들을 잘 설명하는 새로운 특징(feature)을 축으로 하는 공간을 찾았다는 뜻으로 해석할 수도 있다. 매니폴드를 잘 찾으면 이미지들의 의미적인 유사성을 잘 보존할 수 있다. 또한 유사한 데이터를 획득하여 학습에 없는 데이터를 획득할 가능성도 있다.
	Q5. 충현: 수업 중에 의사결정트리(Decision Tree)이론에서 대해 언급했다. 트리를 만들려면 데이터를 모으고 각 데이터의 특징을 추출한다고 들었다. 그리고 트리가 모아지면 숲(Forest)가 되어진다. 그러면 예를 들어서 50개의 작은 의사결정 트리가 모아져서 하나의 큰 의사결정 숲이 된다고 하면, 큰 의사결정 숲이 50개의 작은 의사결정 트리보다 더 효율적일까? **A5.** 의사결정트리는 이해하고 해석하기 쉬운데다가 동시에 숫자형/범주형 데이터를 다룰 수 있지만, 새로운 데이터에 대한 일반화 성능이 좋지 않게 overfitting 되기 쉽다는 점이 있다. 또한 의사결정트리가 많이 발생하게 된다면 어떤 것이 최적의 의사결정트리인지 정하기 힘들어진다는 단점이 있다.

	Random Forest는 전체 데이터가 아닌 샘플의 결과물을 랜덤한 순서로 각 트리의 입력 값으로 넣어 학습하는 방식이기 때문에 앙상블 학습 방법이라고도 불리며, 이렇게 구성된 다수의 결정 트리로부터 분류, 또는 평균 예측치를 출력함으로써 동작한다. 즉, 의사결정트리의 단점을 상쇄하기 위해서 고안된 알고리즘이므로 많은 데이터와 많은 의사결정트리가 필요한 작업에는 효율이 뛰어나지만, 데이터가 적고 의사결정트리 또한 많이 필요하지 않은 곳에서는 오히려 데이터의 결과 값이 한 쪽에 너무 치우치거나 몇 개의 의사결정트리를 사용하는 것 보다 효율이 떨어질 수 있다. 따라서, 위로 미루어 보아 무조건적으로 큰 의사결정트리가 작은 의사결정트리보다 더 효율적이라고 확신할 수 없다 생각되어진다.
질문 내용	1. 과소적합과 과대적합에 대해 논의하던 도중 feature라는 용어를 알게 되었습니다. 과소적합을 해결할 때 무조건 데이터를 증가시키는 것이 아니라 feature를 증가시키라는 말이 나오는데, 이 말이 그 모델의 특징을 더 많이 나타내는 데이터를 증가시키라는 것이 맞나요? 그게 맞다면 그 feature가 유용한 feature인지 아닌지 구분은 어떻게 하는지 궁금합니다. 2. 가중치 감쇠에서 가중치를 감소시켜 노이즈 및 아웃라이어의 영향을 줄인다는 개념은 이해하겠는데, 그 방식으로 제시되는 L1, L2 정규화의 경우 손실함수의 값을 증가시키는 방향으로 작용한다는 것을 알게 되었습니다. 손실함수의 값이 클수록 모델과 데이터 사이의 괴리가 크다는 것을 의미하고, 이는 모델의 더 큰 변화를 야기할 것 같은데, 어떻게 손실함수 증가가 가중치의 감소를 이끌어 내는지 잘 모르겠습니다.