기계학습 (Machine Learning)

- Machine Learning and Python/Mathematics

정보통신공학과

Prof. Jinkyu Kang

수업의 목차

- **기계학습과 Python**
  - 라이브러리 및 도구들
  - 구글 코랩 (Colab)

- **기계학습과 수학**
  - 선형 대수
  - 확률과 통계
  - 최적화 이론

확률과 통계

- 확률 기초
- 베이즈 정리와 기계 학습
- 최대 우도
- 평균과 분산
- 유용한 확률분포
- 정보이론

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

확률과 통계 | 확률 기초

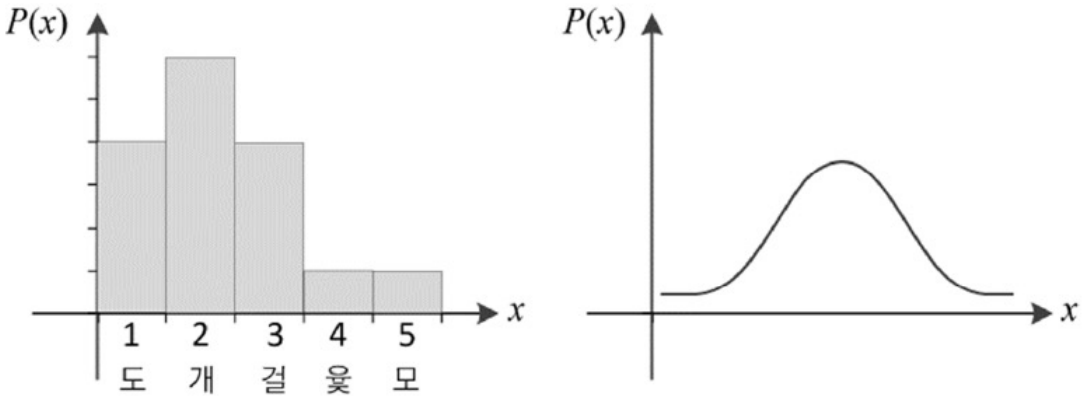
- 확률변수random variable
  - 예) 윷

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

확률과 통계 | 확률 기초

- 확률분포

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

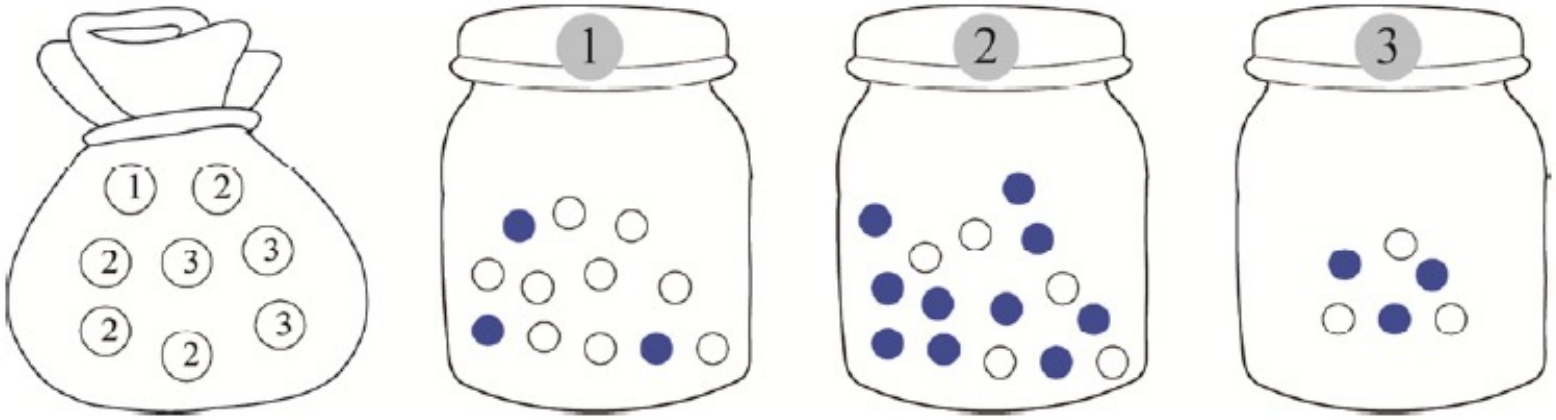


(a) 이산인 경우의 확률질량함수 (b) 연속인 경우의 확률밀도함수

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

확률과 통계 | 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$\}$



확률과 통계 | 확률 기초

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

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

  - 곱 규칙

$$곱\ 규칙:\ P(y,x) = P(x|y)P(y)$$

  - 하얀 공이 뽑힐 확률

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

  - 합 규칙 $합\ 규칙:\ P(x) = \sum_y P(y,x) = \sum_y P(x|y)P(y)$

확률과 통계 | 베이즈 정리와 기계 학습

- 베이즈 정리

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

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

  - 다음 질문을 식으로 쓸 수 있음

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

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

확률과 통계 | 베이즈 정리와 기계 학습

- 베이즈 정리
  - 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

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

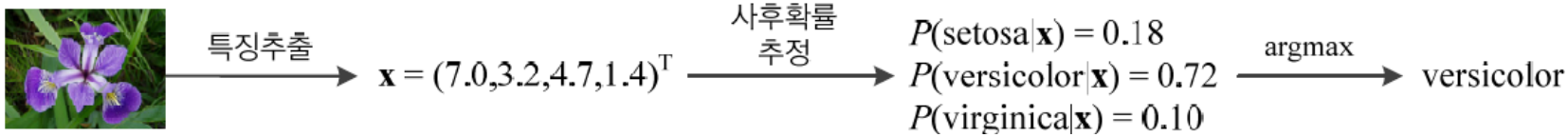
- 베이즈 정리의 해석

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

확률과 통계 | 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 아래 식

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



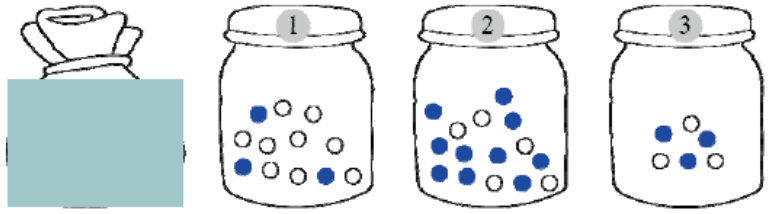
  - 사후확률 $P(y|\mathbf{x})$를 직접 추정하는 일은 아주 단순한 경우를 빼고 불가능
  - 따라서 베이즈 정리를 이용하여 추정함
    - 사전확률

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

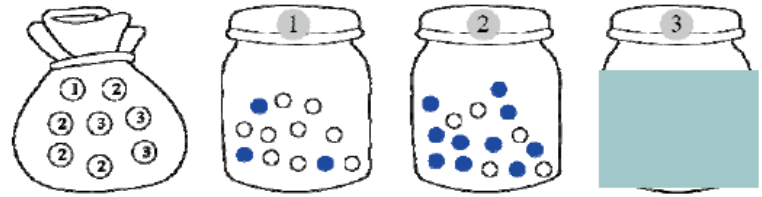
    - 우도는 밀도 추정 기법으로 추정

확률과 통계 | 최대 우도

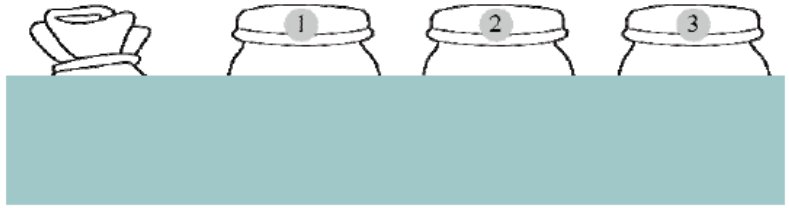
- 매개변수 Θ를 모르는 상황에서 매개변수를 추정하는 문제



(a) $\Theta = \{p_1, p_2\}$



(b) $\Theta = \{q_3\}$



(c) $\Theta = \{p_1, p_2, q_1, q_2, q_3\}$

- 예) [그림 (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."

확률과 통계 | 최대 우도

- 최대 우도법
  - [그림 (b)] 문제를 수식으로 쓰면,

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

  - 일반화 하면,

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

  - 수치 문제를 피하기 위해 로그 표현으로 바꾸면,

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

확률과 통계 | 평균과 분산

- 데이터의 요약 정보로서 평균과 분산

$$\text{평균 } \mu = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \text{분산 } \sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2$$

- 평균 벡터와 공분산 행렬

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

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

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

확률과 통계 | 평균과 분산

- 평균 벡터와 공분산 행렬 예제

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.

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

확률과 통계 | 평균과 분산

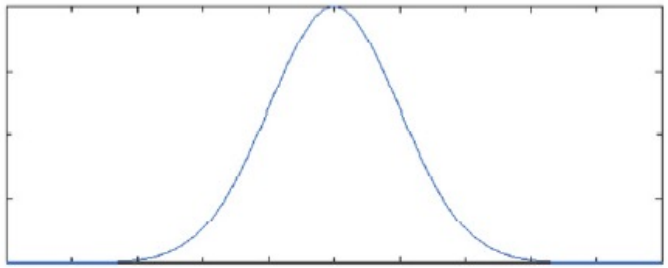
- 평균 벡터와 공분산 행렬 예제 (Colab)

```
import numpy as np

x = np.array([[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], [5.0,3.6,1.4,0.2],
[5.4,3.9,1.7,0.4], [4.6,3.4,1.4,0.3], [5.0,3.4,1.5,0.2]])
print(x, '\n\n')

x_mean = x.mean(axis=0)
print(x_mean, '\n\n')

x_var = np.cov(x.T, ddof=0)
print(x_var)
```

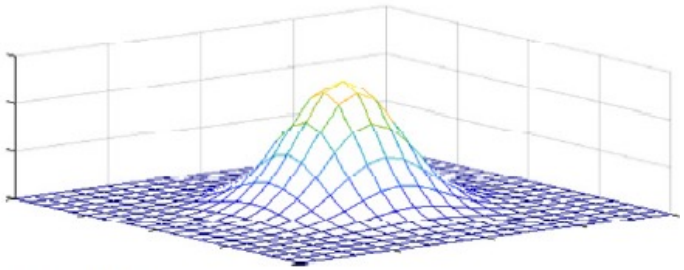
확률과 통계 | 유용한 확률분포

- 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

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

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

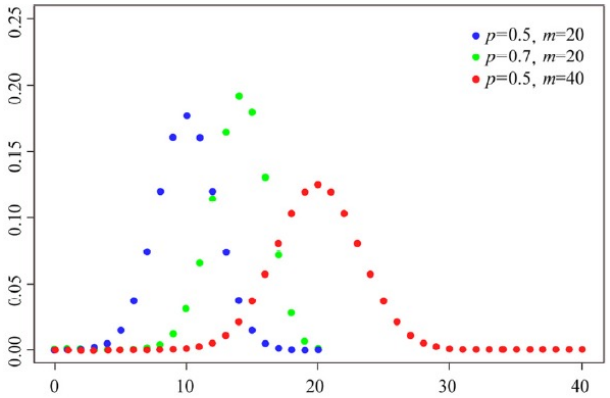
확률과 통계 | 유용한 확률분포

- 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

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



확률과 통계 | 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)

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

- 엔트로피
  - 확률변수 x의 불확실성을 나타내는 엔트로피

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

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

확률과 통계 | 정보이론

- 자기 정보와 엔트로피 예제

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?

확률과 통계 | 정보이론

- 자기 정보와 엔트로피 예제 (Colab)

```
import math

prob_yut = np.array([4/16, 6/16, 4/16, 1/16, 1/16])
H_yut = 0;
for i in prob_yut:
H_yut += - i*math.log2(i)

print(H_yut, '\n\n')

prob_dice = np.array([1/6, 1/6, 1/6, 1/6, 1/6, 1/6])
H_dice = 0;
for i in prob_dice:
H_dice += - i*math.log2(i)

print(H_dice)
```

확률과 통계 | 정보이론

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

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

  - 식을 전개하면,

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

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

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

KL 다이버전스

확률과 통계 | 정보이론

- KL 다이버전스
  - P와 Q 사이의 KL 다이버전스

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

  - 두 확률분포 사이의 거리를 계산할 때 주로 사용

- 교차 엔트로피와 KL 다이버전스의 관계

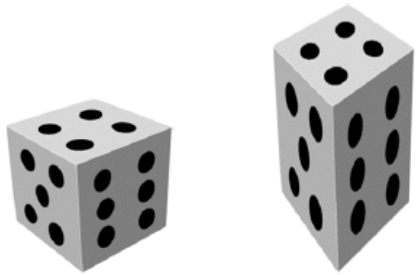
$$P\text{와 } Q\text{의 교차 엔트로피 } H(P,Q) = H(P) + \sum_x P(x) \log_2 \frac{P(x)}{Q(x)}$$

$$= P\text{의 엔트로피} + P\text{와 } Q \text{ 간의 } KL \text{ 다이버전스}$$

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.

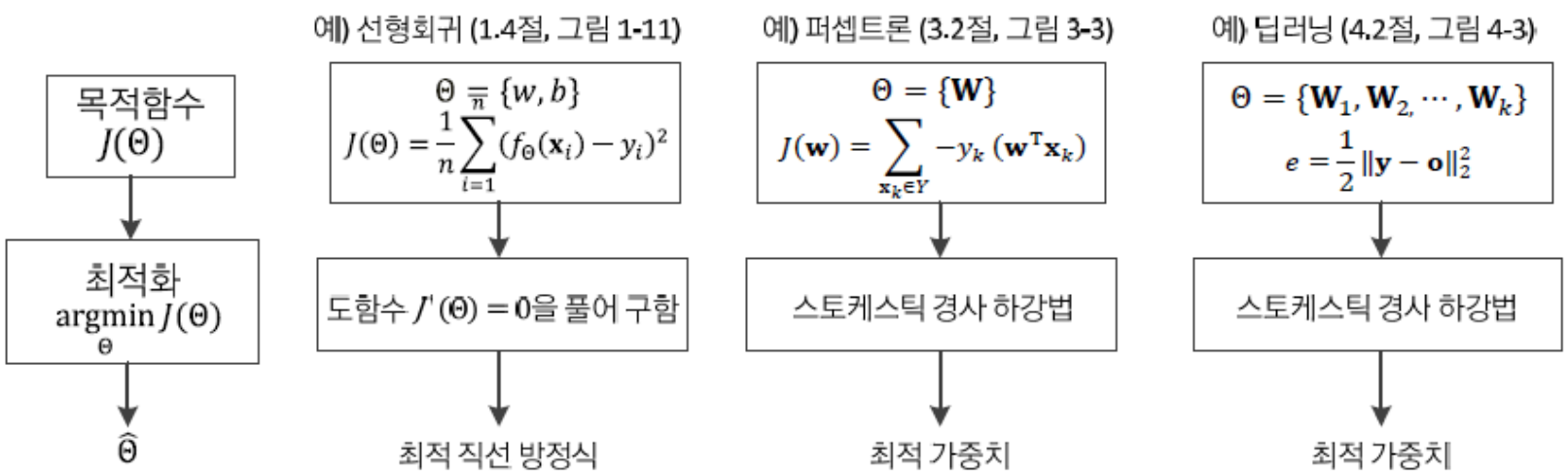
최적화

- 매개변수 공간의 탐색
- 미분
- 경사 하강 알고리즘

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

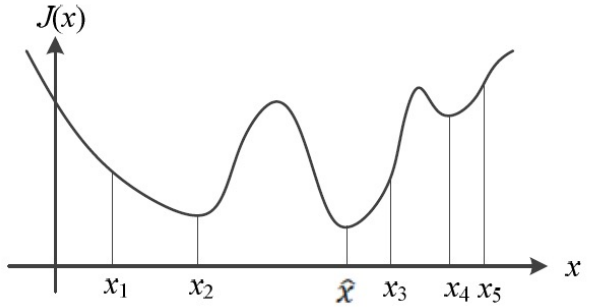
최적화 | 매개변수 공간의 탐색

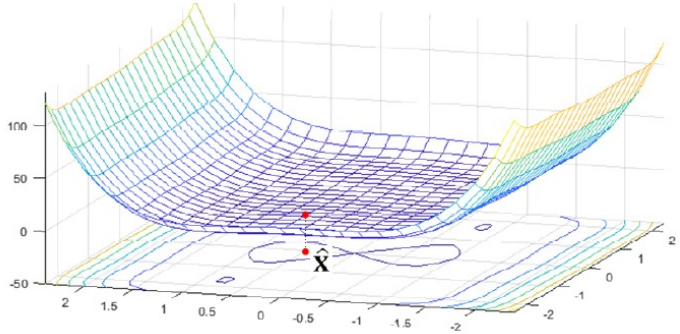
- 학습 모델의 매개변수 공간
  - 높은 차원에 비해 훈련집합의 크기가 작아 참인 확률분포를 구하는 일은 불가능함
  - 따라서 기계 학습은 적절한 모델을 선택하고, 목적함수를 정의하고, 모델의 매개변수 공간을 탐색하여 목적함수가 최저가 되는 최적점을 찾는 전략 사용 → 특징 공간에서 해야 하는 일을 모델의 매개변수 공간에서 하는 일로 대치한 셈
  - 아래 그림은 여러 예제 (Θ는 매개변수, $J(\Theta)$는 목적함수)



최적화 | 매개변수 공간의 탐색

- 학습 모델의 매개변수 공간
  - 특징 공간보다 수 배~수만 배 넓음
    - 선형회귀에서는 특징 공간은 1차원 , 매개변수 공간은 2차원
    - MNIST 인식하는 딥러닝 모델은 784차원 특징 공간, 수십만~수백만 차원의 매개변수 공간
  - 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





- 기계 학습이 해야 할 일을 식으로 정의하면,

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

최적화 | 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 $\hat{\boldsymbol{\Theta}}$

1 가능한 해를 모두 생성하여 집합 S에 저장한다.
2 min을 충분히 큰 값으로 초기화한다.
3 for (S에 속하는 각 점 $\boldsymbol{\Theta}_{current}$에 대해)
4 　　if($J(\boldsymbol{\Theta}_{current}) < min$) $min = J(\boldsymbol{\Theta}_{current})$, $\boldsymbol{\Theta}_{best} = \boldsymbol{\Theta}_{current}$
5 $\hat{\boldsymbol{\Theta}} = \boldsymbol{\Theta}_{best}$

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 $\hat{\boldsymbol{\Theta}} = \boldsymbol{\Theta}_{best}$

최적화 | 매개변수 공간의 탐색

- 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 $\hat{\boldsymbol{\Theta}}$

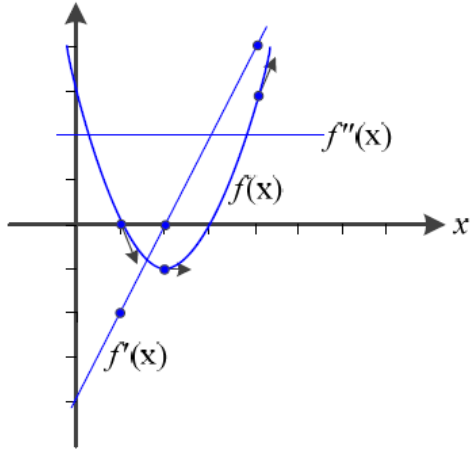
```
1  난수를 생성하여 초기해 Θ을 설정한다.
2  repeat
3      J(Θ)가 작아지는 방향 dΘ를 구한다.
4      Θ = Θ + dΘ
5  until(멈춤 조건)
6  Θ̂ = Θ
```

최적화 | 미분

- 미분에 의한 최적화
  - 미분의 정의

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

  - 1차 도함수 $f'(x)$는 함수의 기울기, 즉 값이 커지는 방향을 지시함
  - 따라서 $-f'(x)$ 방향에 목적함수의 최저점이 존재
  - [알고리즘 2-3]에서 $d\mathbf{\Theta}$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

최적화 | 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)

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

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

최적화 | 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.

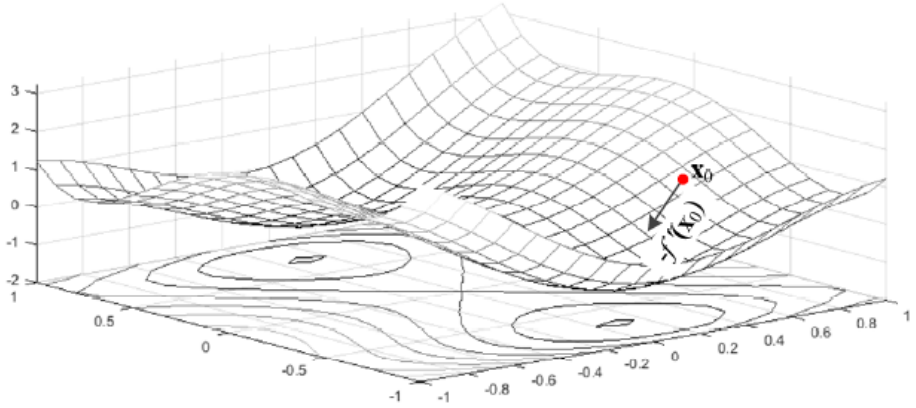


그림 2-25 그레이디언트는 최저점으로 가는 방향을 알려 줌

최적화 | 경사 하강 알고리즘

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

$$\boldsymbol{\Theta} = \boldsymbol{\Theta} - \rho \mathbf{g}$$

- 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 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
```

훈련집합

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

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

최적화 | 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 $\hat{\boldsymbol{\Theta}}$

```
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∇
```

Thank you