확률론

- **딥러닝**

  확률론 기반의 기계학습 이론에 바탕을 둠

  기계학습에서 사용되는 손실함수(loss function)들의 작동 원리는 데이터 공간을 통계적으로 해석해서 유도

  - 기계학습의 기본 원리: 예측이 틀릴 위험(risk)을 최소화하도록 데이터를 학습하는 원리
- L_2-노름(회귀 분석에서 손실함수로 사용)

  예측오차의 분산을 가장 최소화하는 방향으로 학습하도록 유도
- 교차엔트로피(cross-entropy)(분류 문제에서 사용)

  모델 예측의 불확실성을 최소화하는 방향으로 학습하도록 유도
- 분산 및 불확실성 최소화하기 위해서는 측정하는 방법 알아야 함

  두 대상을 측정하는 방법을 통계학에서 제공 => 기계학습 이해하려면 확률론의 기본 개념 알아야 함
- 확률분포는 데이터의 초상화
  - 데이터공간: $X \times Y$
  - 데이터를 추출하는 분포: $\mathscr{D}$
  - 데이터 확률변수: $(X, y) \sim \mathscr{D}$
- **확률변수**

  표본공간 내에 있는 각 원소에 하나의 실수값을 대응시키는 함수

  확률 분포 $\mathscr{D}$에 따라 **이산형(discrete) & 연속형(continuous)**으로 구분됨
  - **이산형(discrete) 확률변수**: 확률변수가 취할 수 있는 값이 유한하거나 무한히 많더라도 하나씩 셀 수 있는 경우 (주로 계수자료(count data))

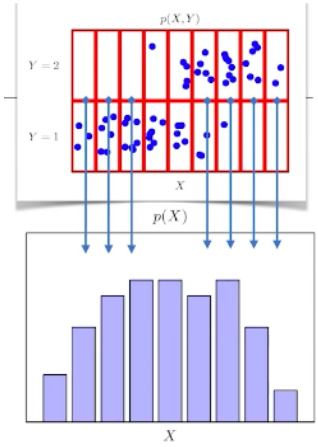
    확률변수가 가질 수 있는 경우의 수를 모두 고려해 **확률을 더해서 모델링**

    $\mathbb{P}(X \in A) = \sum_{X \in A} P(X = x)$
    - $P(X = x)$: 확률변수가 x 값을 가질 확률
  - **연속형(continuous) 확률변수**: 확률변수가 연속적인 구간 내의 값을 취하는 경우(높이, 무게, 온도, 거리, 수명 등 주로 측정자료)

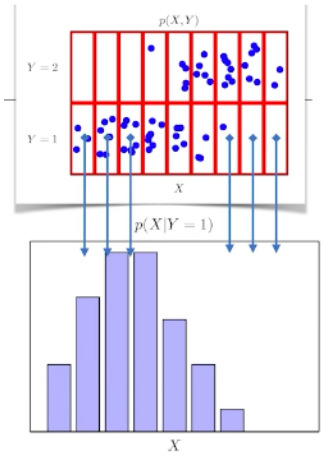
    데이터 공간에 정의된 확률변수의 밀도(density) 위에서의 **적분을 통해 모델링**

    $\mathbb{P}(X \in A) = \int_A P(x)dx$
    - $P(x) = lim_{h \to 0} \frac{\mathbb{P}(x-h \leq X \leq x+h)}{2h}$
    - 밀도는 **누적확률분포의 변화율을 모델링**하여 확률로 해석하면 X
- 결합분포(joint distribution) $P(X, Y)$는 $\mathscr{D}$ 모델링
- $\mathscr{D}$는 이론적으로 존재하는 확률분포 => 사전에 알 수 X
- $P(X)$는 입력 X에 대한 주변확률분포로 y에 대한 정보를 주진 않는다. => **주변확률분포**

  $P(X) = \sum_y P(X, y) \;\; P(X) = \int_y P(X, y)dy$



  - 주변확률분포 $P(X)$는 결합분포 $P(X, y)$에서 유도 가능
- **조건부확률분포 $P(X|y)$**



  - 데이터 공간에서 입력 X와 출력 y 사이의 관계 모델링
  - $P(X|y)$: 특정 클래스가 주어진 조건에서 데이터의 확률분포를 보여줌
- **조건부확률분포 $P(y|X)$**
  - 입력변수 X에 대해 정답이 y일 확률 의미
  - **로지스틱 회귀**에서 사용했던 선형모델과 소프트맥스 함수의 결합 => **데이터에서 추출된 패턴을 기반으로 확률 해석하는데 사용**
    - **로지스틱 회귀(Logistic Function)**

      범주형 변수를 예측하는 모델

      x값으로 어떤 값이든 받을 수가 있고, 출력 결과는 항상 0에서 1사이 값

      => **확률밀도함수(probability density function)** 요건을 충족시키는 함수
  - 분류 문제에서 $softmax(W\phi + b)$은 데이터 X로부터 추출된 특징패턴 $\phi(X)$과 가중치행렬 W을 통해 조건부 확률 $P(y|X)$ 계산
  - 회귀 문제의 경우, 조건부기대값 $\mathbb{E}[y \mid X]$을 추정

    $\mathbb{E}_{y \sim P(y|X)}[y|X] = \int_y P(y|X)dy$
  - 딥러닝: 다층신경망을 사용하여 데이터로부터 특징패턴 ϕ 추출

    특징패턴 학습 위해 어떤 손실함수를 사용할지는 기계학습 문제와 모델에 의해 결정
  - 조건부기대값은 $\mathbb{E} \parallel y - f(x) \parallel_2$을 최소화하는 함수 $f(x)$와 일치
- **기대값**

- 확률분포가 주어지면 데이터를 분석하는 데 사용 가능한 여러 종류의 **통계적 범함수(statistical functional)를 계산 가능**
- **기대값(expectation)은 데이터를 대표하는 통계량**

  +) 확률분포를 통해 다른 통계적 범함수 계산하는데 사용

  $\mathbb{E}_{y\sim P(x)}[f(x)] = \int_X f(x)P(x)dx$

  => 연속확률분포: 적분

  $\mathbb{E}_{y\sim P(x)}[f(x)] = \sum_{x\in X} f(x)P(x)$

  => 이산확률분포: 급수
- 분산, 첨도, 공분산 등 여러 통계량 계산

$$\mathbb{V}(\mathbf{x}) = \mathbb{E}_{\mathbf{x}\sim P(\mathbf{x})}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])^2] \qquad \text{Skewness}(\mathbf{x}) = \mathbb{E}\left[\left(\frac{\mathbf{x} - \mathbb{E}[\mathbf{x}]}{\sqrt{\mathbb{V}(\mathbf{x})}}\right)^3\right]$$

$$\text{Cov}(\mathbf{x}_1, \mathbf{x}_2) = \mathbb{E}_{\mathbf{x}_1,\mathbf{x}_2\sim P(\mathbf{x}_1,\mathbf{x}_2)}[(\mathbf{x}_1 - \mathbb{E}[\mathbf{x}_1])(\mathbf{x}_2 - \mathbb{E}[\mathbf{x}_2])]$$

  위 수식에 f 대신 대입하면 통계량 계산 가능

- **몬테카를로 샘플링**
  - 기계학습의 많은 문제 => 확률분포를 명시적으로 모르는 경우 많음
  - 확률분포를 모를 때, **데이터를 이용하여 기대값을 계산하려면 몬테카를로(Monte Carlo) 샘플링 방법 사용**

    $\mathbb{E}_{X\sim P(X)}[f(X)] \approx \frac{1}{N}\sum_i^N = 1f(X_{(i)}) \;\; x^{(i)} \sim^{i.i.d.} P(x)$
  - 이산형이든 연속형이든 상관 X 성립
  - 몬테카를로 샘플링: 독립추출만 보장

    => 대수의 법칙(law of large number)에 의해 **수렴성 보장**

```
import numpy as np

def nc_int(fun, low, high, sample_size = 100, repeat = 10):
  int_len = np.aba(high - low)
  stat = []
  for _ in range(repeat):
    x = np.random.uniform(low=low, high, high, size=sample_size)
    fun_x = fun(x)
    int_val = int_len * np.mean(fun_x)
    stat.append(int_val)
  return np.mean(stat), np.std(stat)

def f_x(x):
  return np.exp(-x**2)

print(mc(int(f_x, low=-1, hight=1, sample_size=10000, repeat=100)))

# 1.49387543... 0.0039739845...
```

  - 샘플링 개수가 적을 경우 => 오차범위 커짐, 참값에 멀어짐