Introduction

Terminology

- **Observation**: 관측치; 어떤 특정한 기준(criteria)을 가지고 측정된 결과를 가리키며, 일반적으로 table의 한 **row**에 해당함
    - But, theoretically, 모든 population을 조사하는 것은 불가능하기 때문에(i.e. population이란 곧 theoretical하게만 존재하기 때문에) 측정된 결과는 결국 sample
    - 따라서 observation은 **sample**이라고도 부르며, data란 결국 observation 또는 sample의 집합
    - Underlying assumption: **Every observation, or sample, is identically and independently distributed!**
        - Independent distribution: 특정한 observation은 다른 observation의 측정값에 영향을 주지 않는다.
            - cf) Time-series data에 대해서는 이러한 가정이 깨짐
        - Identical distribution: observation의 분포는 population의 분포와 일치한다.
            - In other words, 특정한 observation이 data 내에서 발견될 가능성(likelihood)은 전체 population 내에서 발견될 가능성과 일치한다.
            - 즉, 모든 sample(observation)이 population으로부터 추출될 가능성이 동일하다.
        - 쉽게 말하면, 모든 observation은 population으로부터 **“sampled with replacement with the same weight”**
- **Feature**: 특성; 관측치를 측정할 때 사용한 기준을 의미하며, 일반적으로 table의 한 **column**에 해당함
    - **독립변수**로 사용되며, 예측에 사용된다는 의미에서 **Predictor**라고도 부름
- **Target**: 학습 대상; feature들 사이에서 특히 관심의 대상 또는 인과관계에서의 “결과" 부분에 해당하는 값
    - **종속변수**로 사용되며, **어떠한 feature이든 target이 될 수 있다**.
    - 즉 feature 가운데 하나로서 역시 table에서는 column으로 존재한다.
- Statistic, Estimator and Estimate

- **Statistic: 통계량**; "추상적인 표본"을 사용해 특정한 실수값을 계산 및 도출하는 하나의 **함수**
- **Estimator: 추정량**; population distribution에 대한 정보를 가지고 있는(또는 그것을 추론할 때 사용하는) 특정한 수치(quantity)에 대한 통계량 (a special kind of statistic)
  - ex) sample mean $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$
- **Estimate: 추정값**; estimator의 "추상적인 표본" 자리에 실제 data를 plug-in했을 때 도출되는 **결과값**
  - ex) sample mean of [1, 2, 3, 4 ,5] → $\hat{\bar{X}} = 3$
- In general, sample(observation)을 가지고 계산된 estimate에 대해서는 ˆ(hat) 기호를 붙여준다.
  - 한편, statistical learning에서는 predictor - target 사이의 관계를 나타내는 함수 역시 sample로부터 계산되기 때문에 함수 f에 대해서도 $\hat{f}$로 나타낼 수 있다.

Purpose of (supervised) learning

- Target을 Y, Feature를 X라고 했을 때, 우리가 원하는 것은 X - Y 사이의 관계를 **가장 잘 설명**하고, 새로운 observation이 들어왔을 때 그에 상응하는 target 값을 **가장 잘 예측**하는 **함수** $f(x)$
- 이 두 가지 criteria를 모두 만족시키는 **ideal** f는 X가 주어졌을 때 Y가 평균적으로 갖게될 것으로 기대되는 값, 즉 **conditional expectation** $f(x) = \mathbb{E}[Y \mid X = x]$
  - Conditional expectation은 **조건, 즉 X에 관한 함수**(정확히는 X의 observation x를 투입했을 때 값을 반환하는 함수)이며, capital X는 random variable, small x는 specific **observation** value(constant)
- 우리가 찾고자 하는 함수 $f(x)$는 conditional expectation으로 정해졌고, 그에 따라서 우리가 들고 있는 데이터를 통해 계산되는 conditional expectation은 $\hat{f}(x)$라고 할 수 있음
- In short
  - 우리는 주어지지 않은 observation에 대응하는 target value를 알 수 없지만, 그것에 관심이 있다.
  - 따라서 그것을 ***<u>대표하는 값</u>***으로서 해당 target value를 예측하며, 그 대푯값은 feature의 특정한 값이 주어졌을 때 평균적으로 기대되는 값을 나타내는 conditional expectation $\mathbb{E}[Y \mid X = x]$이다.

- 그리고 conditional expectation은 그 정의상 x에 관한 함수이므로 $f(x)$으로 나타낼 수 있다.
  - 그리고 우리는 **주어진 데이터**를 통해 ***conditional expectation 함수 자체를 추정함***으로써 새로운 observation에 대한 prediction도 계산할 수 있는 것이다.
    - 그리고 이렇게 추정된 함수를 $\hat{f}(x)$으로 나타낼 수 있으며, $\hat{f}$는 우리가 알아낼 수 없는 f를 대신하는 역할을 한다.
  - 결국 우리는 ***$\hat{f}(x)$를 알아내기 위한 방법으로 statistical learning을 사용***하며, $\hat{f}$은 ***가지고 있는 데이터***(observation or sample)뿐 아니라 우리가 ***상정한 형태***에 따라서도 달라진다.
    - learning의 결과 해석, 성능 평가, 새로운 observation에 대한 예측 모두 $\hat{f}$를 가지고 이루어진다.
  - ***결론적으로, 새로운 Y를 prediction하기 위해서 $\mathbb{E}[Y \mid X = x] = f(x)$를 상정하고, 그 $f(x)$를 우리에게 주어진 데이터로 estimate하여 $\hat{f}(x)$를 얻는다고 보면 됨.***
    - $\hat{f}$는 특정한 데이터가 주어지기 전에는 estimator, 데이터가 주어진 결과로 추정되는 함수 결과는 estimate이라고 할 수 있다.

f의 종류

- f, 즉 우리가 추정하고자 하는 대푯값, conditional expectation은 위에서도 언급했듯이 어떤 형태의 함수를 상정하는가에 따라서 달라진다.
- Structured vs Parametricity
  - Structure는 특정 함수의 구조 및 형태를 의미하며(ex. linear and non-linear)
  - Parametricity는 해당 함수 $f = \mathbb{E}[Y \mid X]$가 그 함수를 나타내는 parameter를 필요로 하거나 그러한 parameter들로 설명될 수 있는지의 여부
    - 대부분의 ML/DL 모델은 parametric 모델이며, parameter에 따라서 고유한 structure를 갖는다.
    - 다만 DL 모델들은 이러한 structure로부터 좀 더 자유로운 편이다.
  - In general, **non-parametric models do not assume a specific structure** of the function.
    - 대표적인 non-parameteric 학습 방법 및 모델로는 차원 축소와 clustering이 있다.

- Parametric vs Non-parametric
    - parametric 함수를 사용하는 경우, 어떤 형태의 함수를 사용하는지가 중요하다.
    - 또한 parametric 함수는 주어진 데이터를 특정 함수 형태에 “끼워 맞추는” 경우가 있을 수 있기 때문에, 그러한 형태로부터 자유로운 non-parametric 함수를 사용하면 보다 정확한 예측이 가능할 수도 있다.
    - 하지만 parametric 함수는 이미 그 형태가 정해져 있어 데이터의 양이 적더라도 꽤 정확한 예측력을 보일 수 있다.
        - ex. Linear Regression 모델은 100여 개의 observation으로도 추정이 가능하다.

How to estimate and assess f

- Prediction error
    - 앞서도 언급했듯이, prediction이란 우리에게 주어지지 않은 observation에 대응하는 target value의 “대푯값"을 찾는 것이며, 그 대푯값은 곧 X가 주어졌을 때의 평균적인 Y값, 즉 conditional expectation이다.
    - 그리고 우리에게 주어진 데이터를 통해 추정되는 결과(estimate)은 conditional expectation 자체에 대한 추정, 즉 $\hat{f}$이며, 우리는 이것으로 prediction을 대신한다.
    - 따라서, predicted value와 실제 value 사이에는 **필연적으로** 차이가 존재하며, 이러한 차이를 ***<u>예측 오차(Prediction error)</u>***라고 부른다.
    - 이를 수식으로 나타내면 $Y - \hat{f}(x)$과 같이 나타낼 수 있다.
    - 그리고 이 수식은 다시 다음과 같이 decompose될 수 있다.

$$\text{Error} = Y - \hat{f}(x) = [Y - f(x)] + \left[f(x) - \hat{f}(x)\right]$$

        - 이때, $Y - f(x)$은 ***<u>“대푯값으로서 예측함"</u>*** **<u>에 따라서 발생하는 오차</u>**이다. 즉, 알 수 없는 실제 값에 대해 그것의 대푯값으로서 예측하겠다고 가정함으로써 발생하는 차이이다.
            - 이미 감수하겠다고 선언한 차이나 마찬가지이며, 이는 **줄일 수 없는 (irreducible)** 오차이다.
            - 일반적으로 $[Y - f(x)] = \epsilon$으로 많이 나타낸다.
        - 한편 $\left[f(x) - \hat{f}(x)\right]$는 흔하게 접하는 ***<u>“population과 sample 사이에 발생하는 차이"</u>***이다.

        - 그리고 이 차이는 줄일 수 있는(reducible) 오차로서, 보통 ML 및 DL 모형들은 이러한 차이를 **최소화하는 방향으로 $\hat{f}$가 추정되도록 설계**되어 있는 경우가 많다.

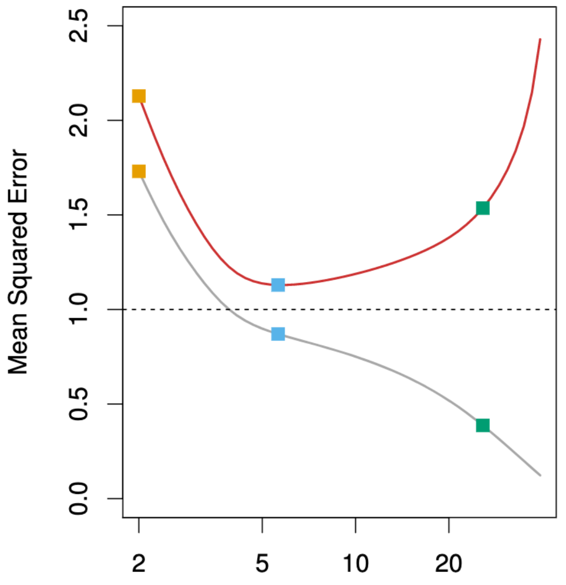
- Mean squared error (MSE)
    - Reducible error를 최소화하는 방향으로 $\hat{f}$를 추정할 때, 최소화를 시키는 objective function으로 가장 널리 사용되는 metric
    - 아래와 같은 수식으로 계산되며, 이 수식 역시 위에서 살펴본 것처럼 **squared reducible error와 variance of irreducible error의 합**으로 decompose될 수 있다.

$$
\begin{aligned}
&\mathbb{E}\left[Y - \hat{f}\left(X\right) \mid X = x\right]^2 \\
&= \mathbb{E}\left[\epsilon + f\left(X\right) - \hat{f}\left(X\right) \mid X = x\right]^2 \\
&= \mathbb{E}\left[\epsilon \mid X = x\right]^2 + \mathbb{E}\left[f\left(X\right) - \hat{f}\left(X\right) \mid X = x\right]^2 \\
&= Var\left(\epsilon\right) + \left[f\left(x\right) - \hat{f}\left(x\right)\right]^2
\end{aligned}
$$

$$
\because \mathbb{E}\left[\epsilon\left(f\left(X\right) - \hat{f}\left(X\right)\right) \mid X = x\right] = \left(f\left(x\right) - \hat{f}\left(x\right)\right)\mathbb{E}\left[\epsilon \mid X = x\right] = 0
$$

    - 모든 오차를 양수로 만들어준다는 점에서 절댓값과 비슷한 효과를 지니는 동시에, 절댓값 함수와 달리 미분이 가능하다는 성질 때문에 가장 널리 사용된다.

- 주의해야 할 점
    - Overfitting
        - "과적합"이라고 하며, 말 그대로 training data에 대해서 학습이 과하게 이루어진 상태로, 굳이 배울 필요가 없는 **training data의 특수성까지 배움**으로써 오히려 새로운 observation에 대한 예측 성능이 더 안 좋아지는 상태
        - 특히 MSE를 최소화하는 $\hat{f}$을 찾고자 할 때, **training error를 완전히 0으로 만들고자 할 경우** overfitting에 매우 취약해진다.
            - In other words, **MSE is biased towards an overfitted model!**
        - 따라서, model assessment는 training에 사용되지 않은 data를 가지고 수행해야 하며, 이러한 data를 test data라고 부른다.
        - 즉, 아래의 그림과 같이 training error - test error가 서로 "적당히" 낮아지는 지점에서 모델을 결정하는 것이 중요하다.



- Bias-Variance Tradeoff
  - Bias: estimator $\hat{f}$의 기댓값과 true value y 사이의 차이; $y - \mathbb{E}\left(\hat{f}\right) = \mathbb{E}\left[y - \hat{f}\right]$
  - Variance: estimator $\hat{f}$에 대해서 그것의 기댓값으로부터 평균적으로 떨어져 있는 정도
    - $\hat{f}$의 robustness를 평가할 때 사용하는 지표
  - 그리고 bias와 variance는 다음의 관계를 갖는다.

$$MSE\left[\hat{f}\left(x_0\right)\right] = Bias\left[\hat{f}\left(x_0\right)\right]^2 + Var\left[\hat{f}\left(x_0\right)\right]$$

pf) estimator $\hat{\theta}$과 true parameter(value) θ의 관계를 통해 파악할 수 있다.

Note that $MSE\left(\theta\right) = \mathbb{E}\left[\hat{\theta} - \theta\right]^2$

$$\begin{aligned}
\mathbb{E}\left[\hat{\theta} - \theta\right]^2 &= \mathbb{E}\left[\hat{\theta} - \mu + \mu - \theta\right]^2 \text{ where } \mu = \mathbb{E}\hat{\theta} \\
&= \mathbb{E}\left[\hat{\theta} - \mu\right]^2 + \mathbb{E}\left[\mu - \theta\right]^2 + \mathbb{E}\left(\hat{\theta} - \mu\right)\left(\mu - \theta\right) \\
&= \mathbb{E}\left[\hat{\theta} - \mathbb{E}\hat{\theta}\right]^2 + \mathbb{E}\left[\theta - \mathbb{E}\hat{\theta}\right]^2 \quad \left(\because \mathbb{E}\left[\hat{\theta} - \mu\right] = 0\right) \\
&= Var\left(\hat{\theta}\right) + \left[\mathbb{E}\left(\theta - \mathbb{E}\hat{\theta}\right)\right]^2
\end{aligned}$$

        - Since θ and $\mathbb{E}\hat{\theta}$ are both constant.
        - Therefore, $MSE\left(\theta\right) = Var + Bias^2$
      - 즉, 우리가 fit한 estimator의 bias가 줄어들면 줄어들수록, 다시 말해 정확도가 높으면 높아질수록 estimator의 분산이 커지며, 이는 곧 estimator가 robust하지 않아 generalize하기 어렵다는 의미이다.

Classification Problem

- Bayes optimal classifier
  - Features $X = x$로 주어졌을 때 target Y가 등장할 확률이 가장 높은 class j로 해당 observation을 분류하는 분류기
  - Theoretically best classifier이며, classifier의 성능을 파악할 때 baseline으로 사용됨
  - 특정한 observation x에 대해 Bayes optimal classifier로 분류된 class를 $C_{\text{Bayes}}\left(x\right)$라고 한다면,

$$C_{\text{Bayes}}\left(x\right) = j^* = \underset{j \in \mathcal{C}}{\arg\max} \Pr\left(Y = j \mid X = x\right)$$

  - 다시 말해, Bayes classifier에서는 주어진 x를 기반으로 그것이 등장할 확률이 가장 높은 class에 x를 할당하는 분류기이며, 그에 따라서 **모든 데이터에 대해 오분류가 0이다**.
- KNN(K-Nearest Neighborhood)
  - 주어진 data point x에 대해서 가장 가까이에 있는 k개의 데이터를 확인하고, k개의 이웃들이 특정한 class j에 속하는 평균적인 비율로써 x가 class j에 속할 확률을 측정하게 됨
    - 쉽게 생각해서 k개의 이웃 가운데 그들이 가장 많이 속한 class로 분류하는 알고리즘

$$\hat{\Pr}\left(Y = j \mid X = x\right) = \frac{1}{k}\sum_{i=1}^{k} I\left(y_i = j\right)$$