Categorical predictors

예를들어 $x \in \{0,1\}$인 경우에는 쉽게 처리가 가능하다.

$y = \beta_0 + \beta_1 x$를 하면 β_1은 $x=1$에만 의존하기에

$\begin{pmatrix} y=\beta_0 \\ y=\beta_0+\beta_1 \end{pmatrix}$ 두 식을 비교가 가능하다.

하지만 $x \in \{0,1,2\}$인 경우 1과 2를 비교해야한다. 그러므로

둘에게 1과 2의 대소를 비교하면 안된다.

⇒ 결국 0, 1만 표현해야한다.

A, B, C가 존재하는 경우

$$x_1 = \begin{cases} A & \text{if } x=1 \\ \text{not } A \end{cases} \qquad x_2 = \begin{cases} B & \text{if } x_2=1 \\ \text{not } B & \text{else} \end{cases}$$

그래서 (x_1과 x_2)을 사용해보면

$\begin{matrix} (1,0) \Rightarrow A \\ (0,1) \Rightarrow B \\ (0,0) \Rightarrow C \end{matrix}$ 을 만들수 있다.

이는 $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$로 표현이 가능해진다.

A: $y = \beta_0 + \beta_1$ ⇒ C 대비 사용량 # 120 ⇒ C보다 A가 120더 사용

B: $y = \beta_0 + \beta_2$ → C 대비 사용량 # -3 ⇒ C보다 B가 -3더 사용

C: $y = \beta_0$ ⇒ 사용량. #500

이런 방식은 category가 K개인 경우 (K-1)개의 변수를 도입한다.

↳ 여기서는 Bias가 카테고리 C인 셈이다.

<one-hot-encoding>

모든 category 마다 변수를 할당한다. (여기서는 Bias가 특정 카테고리가 X)

Modeling Interaction

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$ 처럼 Modeling 하면 X_1과 X_2를 독립적으로 본다.

But X_1과 X_2를 동시에 고려하면 시너지 효과를 얻을 수도 있다.

(synergy, interaction effect)

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon$ 로 modeling이 가능하다.

Model 공식자체는 Linear Regression과 같다.

If) X_1과 X_2의 시너지 X 라면 자연스럽게 β_0는 0으로 수렴할 것이다.

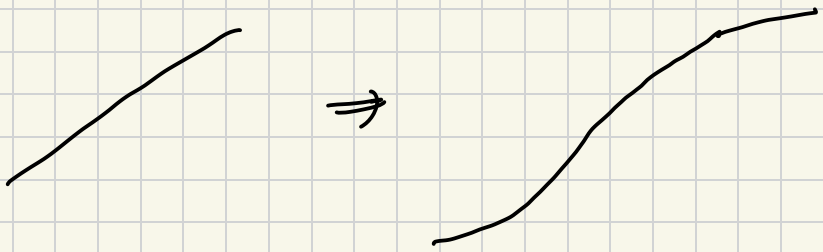
"Hierarchy for interaction"

⇒ 우리가 interaction term을 넣기위해서는 반드시 단일 term을 먼저 넣어주어야한다. 예를 들어) $Y = \beta_0 + \beta_1 X_1 X_2$ (X)

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$ (O)

안그러면 modeling의 안정성이 깨지고, 해석이 난해해진다.

추가로 $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2$ 도 가능하다. 이는 선형 모델의 구조를 유지하면서 비선형성을 추가해주는 역할을 한다.



Feature selection

이상에서 β을 구하는 경우 $\hat{\beta} = (X^TX)^{-1}X^Ty$ 를 통해서 구하게 되는데, $(X^TX)^{-1}$이 존재하지 않을수도 있다. ⇒ 특정 feature 제거가 필요.

그리고 영향력 낮은 β_n이 존재할 수도 있다.

pre-select

〈1. Best subset selection〉 $O(2^p)$

n개의 feature 중에서 k개를 select하는 경우 가능한 nC_k 개의 조합을 모두 고려.

ex) step1 step2 step3 ---->

$y=0.8$ $y=0.2+0.1x_1$ $y=0.1+0.4x_8+0.1x_2$

* curse of dimensionality ⇒ 차원↑일수록 연산량↑이지만 gain↓한다.

step-select

〈2. Forward stepwise〉 $O(p^2)$ But, best solution이 아닐수도.

이전 step에 select된 값들은 유지. (계수는 change 가능)

step1. step2 step3 →-----

$y=0.8$ $y=0.8+0.4x_1$ $y=0.8+0.3x_1+0.1x_7$

〈3. Backward stepwise〉 $O(p^2)$

모든 N개 에서 1개씩 빼나간다 ⇒ (N-1) → (N-2) →--- ⇒ k개 까지.

단, $N>p$를 만족해야한다.

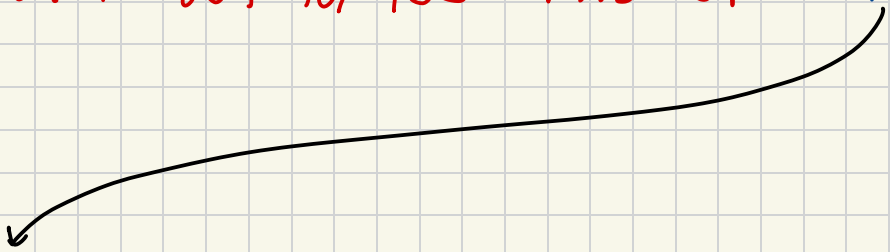
〈4. Stage wise〉

2와 동일. But coefficient 까지도 fix시킨다. But 1번 고른 변수를 다시 고를수있다 그래서 이를 통해서 coefficient를 조절 가능해진다.

Choosing the optimal Model

Stepwise selection에서는 변수의 k가 ↑할수록 성능지표는 올라간다.

But 우리는 data의 "일반적" 특징/패턴을 파악하길 원한다. → validation data로 검증.



Cross validation (train : valid : test).

k-fold cross validation.