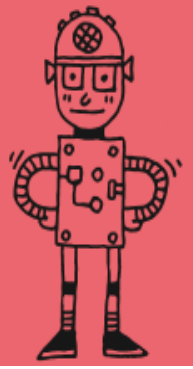
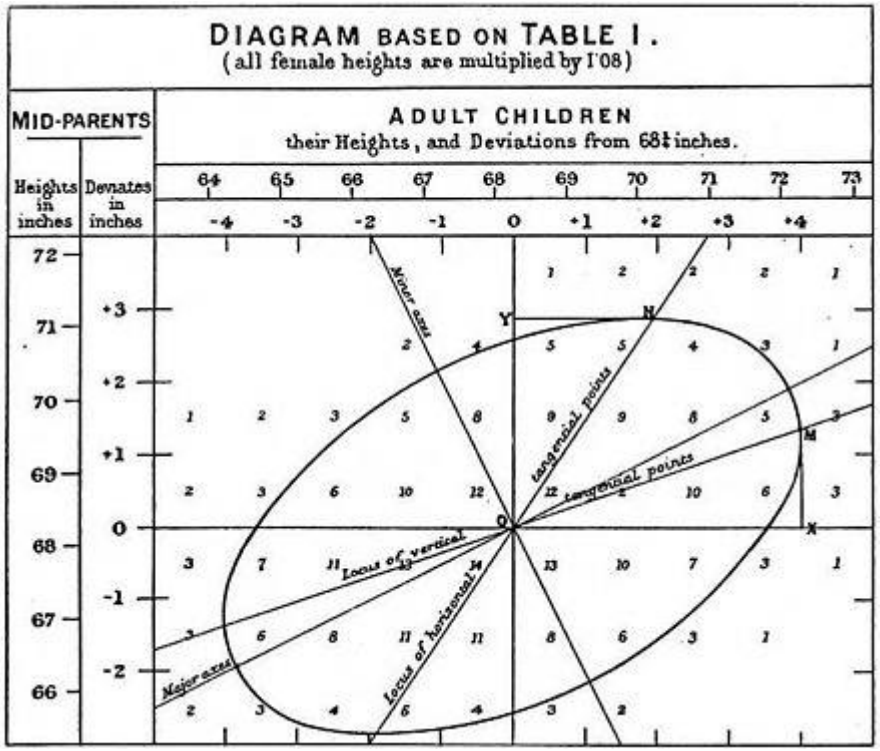
Regression

데이터사이언스 학과
김 승 환
swkim4610@inha.ac.kr



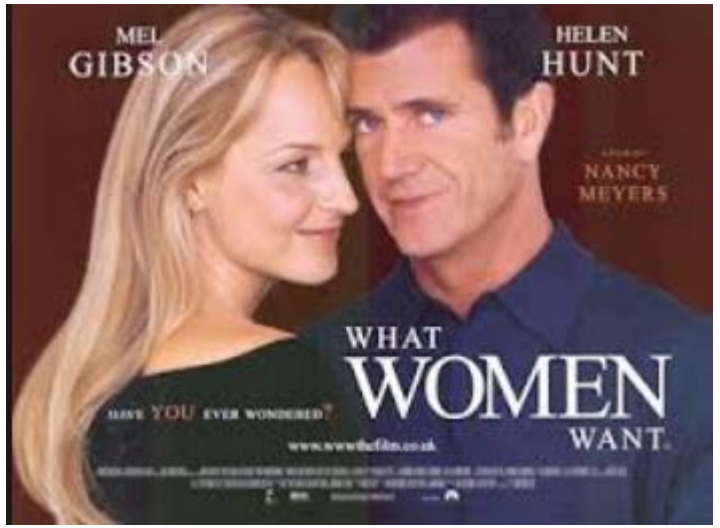
1 Regression 의미

Francis Galton's 1875 illustration of the correlation between the heights of adults and their parents. The observation that adult children's heights tended to deviate less from the mean height than their parents suggested the concept of "regression toward the mean", giving regression its name.



1 Regression 의미

그녀가 원하는 것을 알 수 있는 함수가 있다면?



Function Notation

$$y = f(x)$$

Output

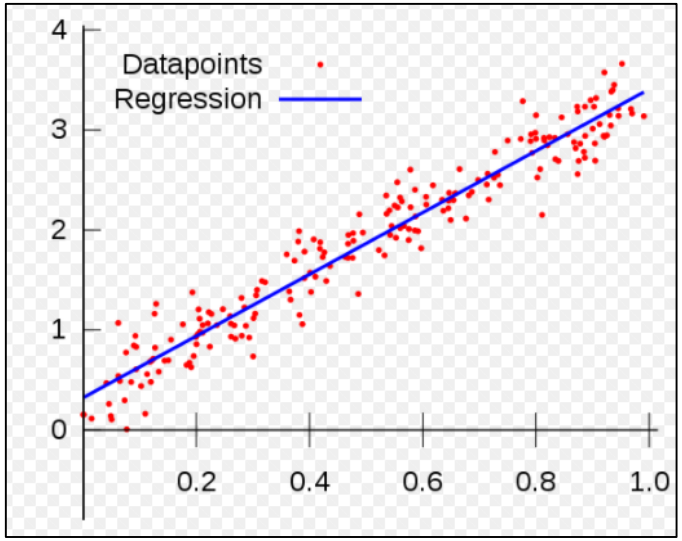
Name of Function

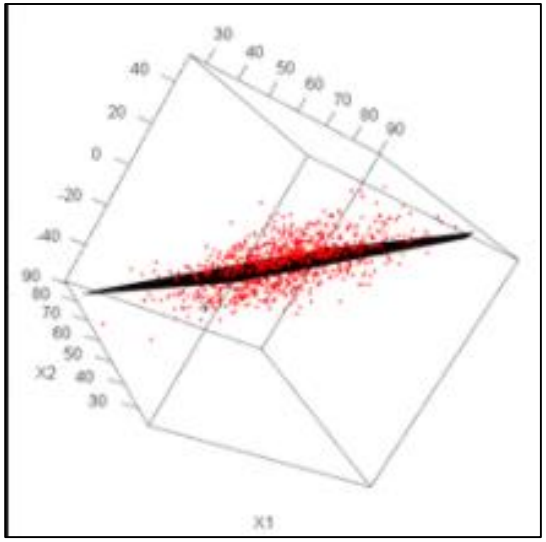
Input

상사가 무엇을 원하는지 알 수 있는 함수가 있다면?

2 Linear Regression

회귀분석은 관측 값을 가장 잘 지나가는 직선 혹은 곡선의 방정식을 구하는 방법론



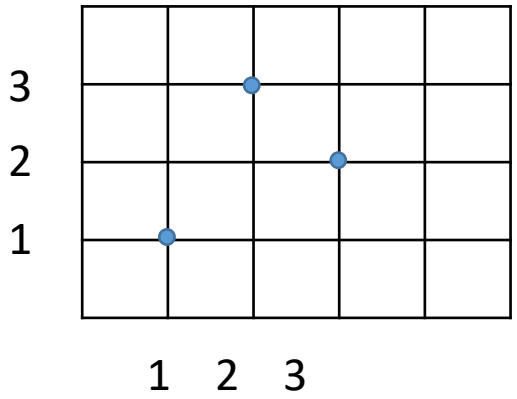


$$Y = \alpha + \beta x + \epsilon$$

$$Y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

직선으로 예측했을 경우, 오차 ε 이 존재, 오차가 작을 수록 좋은 모형임

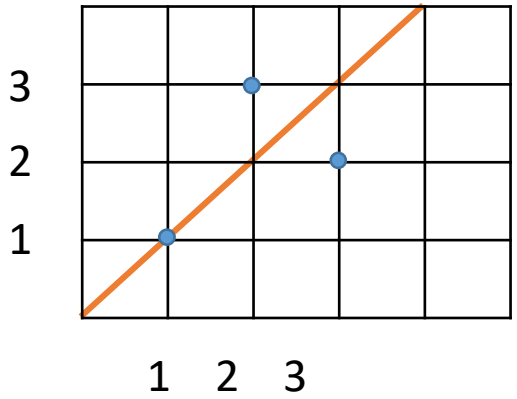
2 Linear Regression



i	x	y
1	1	1
2	2	3
3	3	2

$$y_i = \alpha + \beta x_i + \epsilon_i$$

$$1 = \alpha + \beta * 1 + \epsilon_1$$
$$3 = \alpha + \beta * 2 + \epsilon_2$$
$$2 = \alpha + \beta * 3 + \epsilon_3$$



$$\hat{y}_i = \alpha + \beta x_i$$
$$= 0 + 1 \cdot x_i$$

$$1 = 0 + 1 * 1 + \epsilon_1, \epsilon_1 = 0$$
$$3 = 0 + 1 * 2 + \epsilon_2, \epsilon_2 = 1$$
$$2 = 0 + 1 * 3 + \epsilon_3, \epsilon_3 = -1$$

$\sum_{i=1}^{3} |\epsilon_i|$ 혹은 $\sum_{i=1}^{3} \varepsilon_i^2$ 를 최소화하는 α, β = ?

2 Linear Regression

$$\sum_{i=1}^{n} x_i = x_1 + x_2 + \cdots + x_n \qquad \sum_{i=1}^{n} c = c + c + \cdots + c = n \cdot c$$

$$\sum_{i=1}^{n} c \cdot x_i = c \cdot x_1 + c \cdot x_2 + \cdots + c \cdot x_n = c \sum_{i=1}^{c} x_i$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$\sum_{i=1}^{n} (x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - n \cdot \bar{x}^2$$

$$\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^{n} x_i y_i - n \cdot \bar{x} \cdot \bar{y}$$

3 Least Square method

회귀모형에서 미지수는 SSE를 최소화하는 미지수 알파, 베타를 구하는 계산을 수행하여 얻어짐

(SSE를 일반적으로 cost 함수라고 함)

$$SSE = \sum(\varepsilon_i{}^2) = \sum(Y_i - \alpha - \beta x_i)^2$$

- Least Square Estimation for α, β

$$\frac{\partial SSE}{\partial \alpha} = 2\sum(Y_i - \alpha - \beta x_i)(-1) = 0 \cdots\cdots\cdots ①$$

$$\frac{\partial SSE}{\partial \beta} = 2\sum(Y_i - \alpha - \beta x_i)(-x_i) = 0 \cdots\cdots\cdots ②$$

①× $\bar{x}$ 를 하면,

$$\alpha\sum x_i + \beta\bar{x}\sum x_i = n\bar{x}\bar{Y} \cdots\cdots\cdots\cdots ③$$

$$\alpha\sum x_i + \beta\sum x_i{}^2 = \sum x_i Y_i \cdots\cdots\cdots ④$$

④-③을 하면,

$$\hat{\beta} = \frac{\sum x_i Y_i - n\bar{x}\bar{Y}}{\sum x_i{}^2 - n\bar{x}^2} = \frac{\sum(x_i - \bar{x})(Y_i - \bar{Y})}{\sum(x_i - \bar{x})^2}, \qquad \hat{\alpha} = \bar{Y} - \hat{\beta}\bar{x}$$

4 Multiple Linear Regression

다중 회귀모형은 종속변수에 영향을 주는 다수의 독립변수가 존재하는 경우의 회귀모형임

예를 들어, 연봉을 종속변수로 볼 때, 연봉에 영향을 주는 요인으로
나이, 성별, 학력, 직업, 월 근무시간 등이 있을 것이다. 이를 수식으로 표현하면 아래와 같다.

$$\text{연봉} = \beta_0 + \beta_1 \text{나이} + \beta_2 \text{성별} + \beta_3 \text{학력} + \beta_4 \text{직업} + \beta_5 \text{월근무시간} + \epsilon$$

이를 일반화하여 표현하면 아래와 같다.

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} + \epsilon_i, \epsilon_i \sim N(0, \sigma^2),\ i = 1, \cdots, n \quad \Rightarrow \quad Y = X \cdot \beta + \varepsilon$$

나이, 성별, 학력, 직업, 월 근무시간

연봉

$$X = \begin{pmatrix} 1 & x_{11} & x_{21} & \cdots & x_{k1} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{kn} \end{pmatrix}, Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix}$$

4 Multiple Linear Regression

여기서 미지 모수 β 는 아래와 같이 구할 수 있다.

$$SSE = (Y - X\beta)'(Y - X\beta)$$
$$= Y'Y - Y'X\beta - \beta'X'Y + \beta'X'X\beta$$
$$= Y'Y - 2\beta'X'Y + \beta'X'X\beta$$

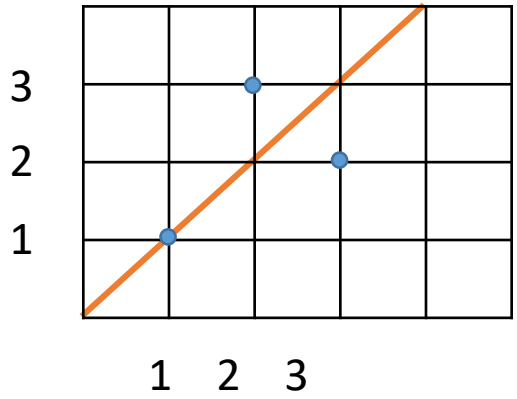
$$\frac{\partial SSE}{\partial \beta} = 2(X'X)\hat{\beta} - 2(X'Y) = 0$$

$$\therefore \hat{\beta} = (X'X)^{-1}X'Y$$

4 숙제

H.W #1: 아래의 x, y 값을 선형 회귀식을 구하고 아래와 같은 Plot을 그리시오.

x	y
1	1
2	3
3	2



$$\hat{\beta} = \frac{\sum x_i Y_i - n\bar{x}\bar{Y}}{\sum x_i^2 - n\bar{x}^2} = \frac{\sum (x_i - \bar{x})(Y_i - \bar{Y})}{\sum (x_i - \bar{x})^2}, \qquad \hat{\alpha} = \bar{Y} - \hat{\beta}\bar{x}$$

H.W #2: 위 문제를 행렬연산으로 구하시오.

$$\hat{\beta} = (X'X)^{-1}X'Y$$

5 Logistic Regression

예측하고자 하는 종속변수가 Binary 일 경우,
예를 들어, Y=0 은 정상 제품, Y=1은 불량 제품을 나타내고, X = (1, 2,1)은 부품의 속성을 나타내는 독립변수이라고 하자. 아래의 Data에서는 X = (1, 2, 1)일 때, Y=0 이므로 정상제품임을 의미한다.

$$X = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 3 & 2 \\ 1 & 3 & 4 \\ 1 & 5 & 5 \\ 1 & 7 & 5 \\ 1 & 2 & 5 \end{pmatrix}, Y = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix}$$

여기서, β는 미지수임

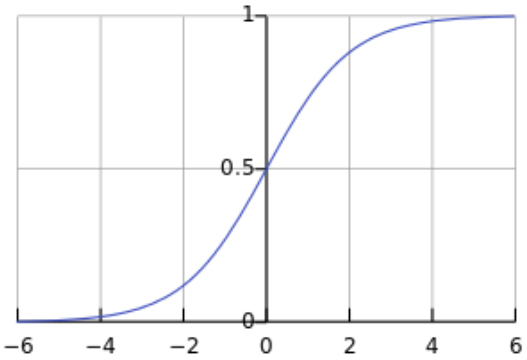
우리의 목표는 베타들을 구하고 X=(1, 2, 4) 와 같이 새로운 제품이 들어 왔을 때, 이 제품이 불량인지 아닌지 로지스틱 회귀모형을 통해 판단해보는 것이다.

이 경우, 일반적 회귀결과, $\hat{y}$ 의 치역은 $(-\infty, \infty)$이 된다. 우리가 원하는 $\hat{y}$ 은 0과 1이므로 적절치 않다.

5 Logistic Regression

이 문제를 해결하기 위해 로지스틱 함수를 사용한다.

로지스틱 함수는 0~1 사이에 값을 가지는 함수로 S 혹은 Sigma 모양과 비슷하여 Sigmoid 함수 라고도 한다. 우변에 Logistic 함수를 적용하여 양변의 스케일 문제를 해결할 수 있다.
이처럼 Logistic 함수를 이용하여 회귀분석을 수행하는 것이 Logistic Regression(Cox, 1958)이라고 부른다.



$$0 < Y = \frac{1}{1 + e^{-x}} < 1$$

$$\hat{y} = \frac{1}{1 + \exp(-\hat{\beta}_0 - \hat{\beta}_1 x_1 - \cdots - \hat{\beta}_k x_k)}$$

$$log\left(\frac{p}{1-p}\right) = X \cdot \beta = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k \quad p = \Pr(Y = 1)$$

5 Logistic Regression

로지스틱 회귀모형은 아래와 같은 수식으로 표현 가능하다.

$$\hat{Y} = \frac{1}{1 + \exp(-X \cdot \hat{\beta})}$$

$$\frac{\hat{Y}}{1-\hat{Y}} = \frac{\frac{1}{1+\exp(-X\cdot\hat{\beta})}}{\frac{\exp(-X\cdot\hat{\beta})}{1+\exp(-X\cdot\hat{\beta})}} = \frac{1}{\exp(-X\cdot\hat{\beta})} = \exp(X\cdot\hat{\beta})$$

$log\left(\frac{p}{1-p}\right) = X\cdot\beta = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$ 로 표현한다.

$$ln\left(\frac{p_i}{1-p_i}\right) = X\cdot\beta + \epsilon, \quad p_i = \Pr(Y=1)$$

$\hat{\beta}$ 이 주어졌을 때, Y에 대한 추정 값은 아래의 식으로 구할 수 있다.

$$\hat{Y} = \frac{1}{1+\exp(-X\cdot\hat{\beta})} = \frac{1}{1+\exp(-X\cdot\hat{\beta})}\cdot\frac{\exp(X\cdot\hat{\beta})}{\exp(X\cdot\hat{\beta})} = \frac{\exp(X\cdot\hat{\beta})}{1+\exp(X\cdot\hat{\beta})}$$

6 Cost 함수

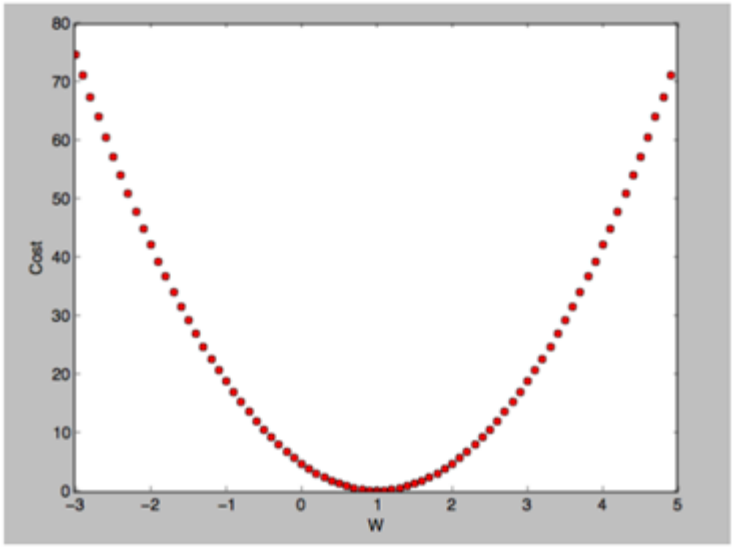
$\hat{\beta}$ 은 아래 Cost 함수를 최소화하는 베타 값이다.
Cost 함수는 정답이면 작아지고, 오답이면 커지는 함수이면서 동시에 Convex해야 한다.
아래 Cross Entropy 함수는 Convex하면서 비용함수의 성질을 만족한다.
여기서, $\hat{\beta}$ 대신 W를 사용하자.

$$Cost(W) = \begin{cases} \sum -log(H(X)), & y = 1 \\ \sum -log(1 - H(X)), & y = 0 \end{cases}$$

$$H(X) = \frac{1}{1+exp(-WX)}$$

y=1 일 때, H(x)=1 이면 cost = 0,
H(x)=0 이면 cost = ∞

y=0 일 때, H(x)=1 이면 cost = ∞,
H(x)=0 이면 cost = 0



Convex Function에서 기울기가 "0" 인 지점을 찾아가기 위해 임의의 초기값 W에서 기울기 방향으로 조금씩(learning rate: α) 움직여 가는 방법을 사용한다.
즉, 임의의 초기값에서 출발하여 기울기 방향으로 움직이면 언젠가는 기울기가 "0" 인 지점에 도착한다는 원리

Q: 왜 SSE 함수를 Cost 함수로 사용하지 않았을까?

7 Convexity

Cross Entropy 함수를 왜 쓰는지 알아보기 위해 아래와 같은 실험을 해보자.
아래는 로지스틱 회귀모형에서 모수인 절편과 기울기 a, b를 일정한 간격으로 mesh 형태로 바꾸면서 Cost 함수를 계산해 Plot 하는 코드다. Cost 함수는 SSE, Cross Entropy 두 종류를 사용했다.

convexity.ipynb

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

def sig(x):
    return 1/(1+np.exp(-x))

x = np.array([1,2,3,4,5,6])
y = np.array([0,0,0,1,1,1])
xy = np.array([[1,0],[2,0],[3,0],[4,1],[5,1],[6,1]])
n = 100
a = np.linspace(-5, 5, n)
b = np.linspace(-2.5, 2.5, n)
a, b = np.meshgrid(a, b)
cost1 = np.zeros((100, 100)) # cross entropy function
for val in xy:
    temp = -val[1] * np.log(sig(a+b*val[0])) - (1-val[1]) * np.log(1-sig(a+b*val[0]))
    cost1 += temp
```

8 Cross Entropy

Cross Entropy 함수는 Logistic Regression의 Maximum Likelihood 함수다.

최대 가능도 함수(Maximum Likelihood function)에 대해 알아보자.

$Y_i = \frac{1}{1+e^{(-\beta_0-\beta_1 x_i+\epsilon)}}$ 의 식에서 Y_i 는 0 혹은 1의 값을 갖는 확률변수다.

그러므로 $Y_i \sim Bernoulli(p_i)$ 이다. 여기서, $p_i = \Pr(Y_i = 1)$ 이고 확률질량함수(P.M.F.)는 아래와 같다.

$f(y) = P(Y = y) = p_i^y(1-p_i)^{1-y}$

만약, $Y_i|X_i = [y_1, y_2, \cdots, y_n]$ 이 관측되었다면 이렇게 관측될 확률은 아래와 같다.

$$(p_i{}^{y_1}(1-p_i)^{1-y_1}) \times (p_i{}^{y_2}(1-p_i)^{1-y_2}) \times \cdots \times (p_i{}^{y_n}(1-p_i)^{1-y_n}) = \prod_{i=1}^{n} p_i{}^{y_i}(1-p_i)^{1-y_i}$$

이 수식을 미지 모수 p_i 에 대한 최대 가능도 함수라고 한다.

위 수식에서 y_i는 주어진 값이고 미지수는 오직 p_i 뿐이다.

이 함수를 최대화 하는 p_i 가 가장 관측 값을 잘 설명하는 값이라고 생각하는 것이 MLE 추정(Maximum Likelihood Estimation)이다.

8 Cross Entropy

p_i 는 y_i 에 대한 추정값으로 $p_i > 0.5$ → $y_i = 1$ 아니면 $y_i = 0$ 이 되므로 아래와 같은 식이 성립한다.

$$p_i = \frac{1}{1 + e^{-\widehat{\beta_0} - \widehat{\beta_1} x_i}}, \qquad L(p_i) = \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1 - y_i}$$

이 식을 최대화 하는 β_0, β_1 이 우리가 구하고자 하는 미지수이고 이를 MLE(Maximum Likelihood Estimator)라고 한다. Likelihood 함수에 로그를 취하면, 아래와 같다.

$$\ln L(\beta_0, \beta_1) = \sum_{i=1}^{n} y_i \ln \frac{1}{1 + e^{-\widehat{\beta_0} - \widehat{\beta_1} x_i}} + (1 - y_i) ln(1 - \frac{1}{1 + e^{-\widehat{\beta_0} - \widehat{\beta_1} x_i}})$$

$$= \sum_{i=1}^{n} y_i \ln p_i + (1 - y_i) ln(1 - p_i)$$

이 식에 "-1" 을 곱한 식이 Cross Entropy 함수다.

최대가능도 함수를 최대화하는 것이나 Cross Entropy 함수를 최소화하는 것이나 같은 것이다.

이 함수를 최소화하는 β_0, β_1 를 구하면 MLE가 된다.

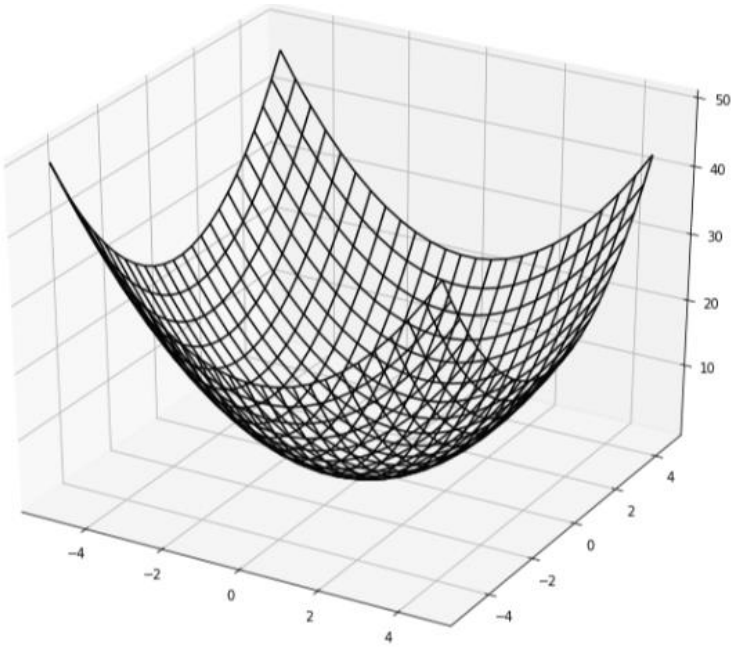
9 Gradient Descent

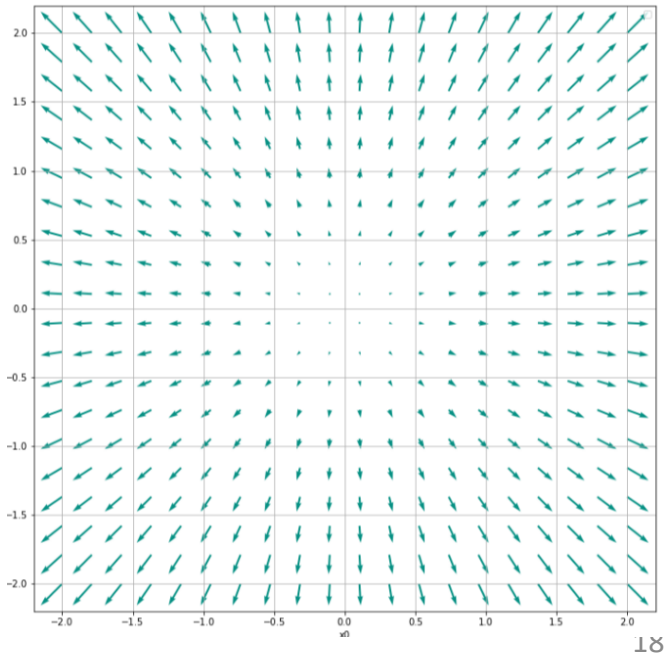
좌측은 2차원 함수의 모양을 나타내고 우측은 이 함수의 기울기 벡터를 구한 것이다.
기울기 벡터의 반대방향으로 가면 최소가 되는 x_0, x_1을 구할 수 있다.

$$x_0 = x_0 - \eta \frac{\partial f}{\partial x_0}$$

$$x_1 = x_1 - \eta \frac{\partial f}{\partial x_1}$$

$$f(x_0, x_1) = x_0^2 + x_1^2$$





9 Gradient Descent

문제는 Cross Entropy 함수를 최소화하는 미지 모수를 수학적으로 구할 수 없다는 것이다.

이러한 문제를 수치해석적 방법으로 해결하는 알고리즘이 최대 경사 하강법(Steepest Descending Algorithm) 이다.

Convex Function에서 기울기가 "0" 인 지점을 찾아가기 위해 임의의 초기값 W에서 기울기 방향으로 조금씩(learning rate: α) 움직여 가는 방법을 사용한다.
즉, 임의의 초기값에서 출발하여 기울기 방향으로 움직이면 언젠가는 기울기가 "0" 인 지점에 도착한다.

$$W := W - \alpha \frac{\partial}{\partial W} \frac{1}{2m} \sum_{i=1}^{m} (Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha \frac{1}{2m} \sum_{i=1}^{m} 2(Wx^{(i)} - y^{(i)})x^{(i)}$$

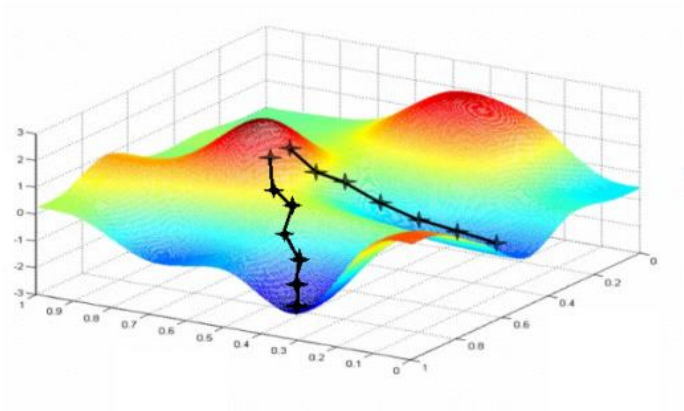
$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} (Wx^{(i)} - y^{(i)})x^{(i)}$$

Cost 함수

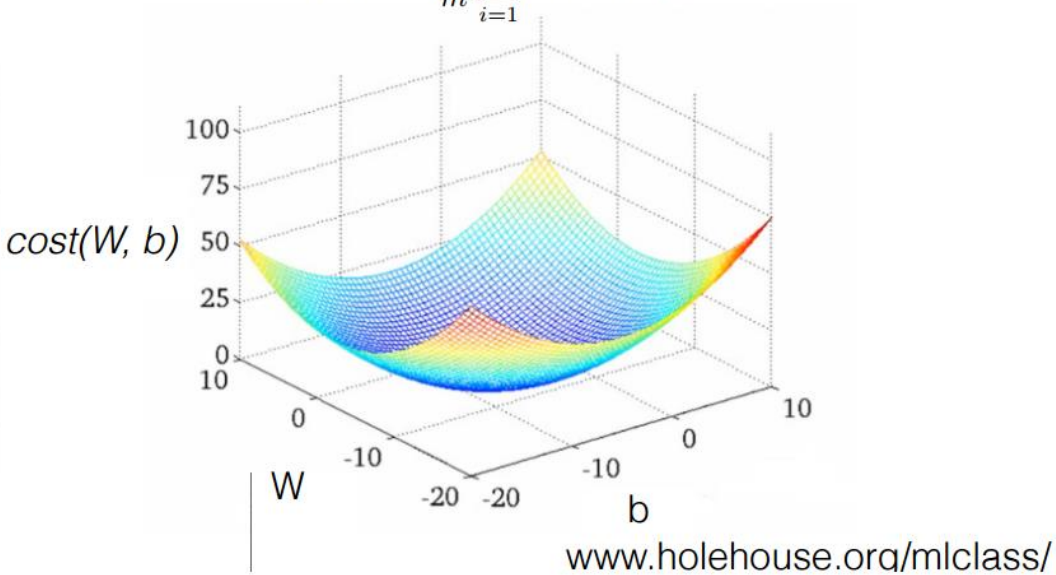
위의 식은 Cost 함수를 미분하여 기울기를 구하고 기울기에 Learning Rate 알파를 곱한 만큼을 빼 주는 방법으로 해를 향해 이동하는 식이다.

9 Gradient Descent

- Local Minimum & Global Minimum



$$cost(W,b) = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$



Gradient Descent Algorithm이 항상 최소, 최대값의 해를 보장하지는 않는다. 즉, "Local Minimum" 이슈가 존재한다. 이는 좌측처럼 어디서 출발하는가에 따라 서로 다른 곳을 해라고 판단하는 문제이다. 만약, 우리의 Cost 함수가 우측과 같다면 즉, 매끈한 모양의 Convex 함수라면 Gradient Descent Algorithm은 항상 Global Minimum 포인트에 도달한다.

9 Gradient Descent

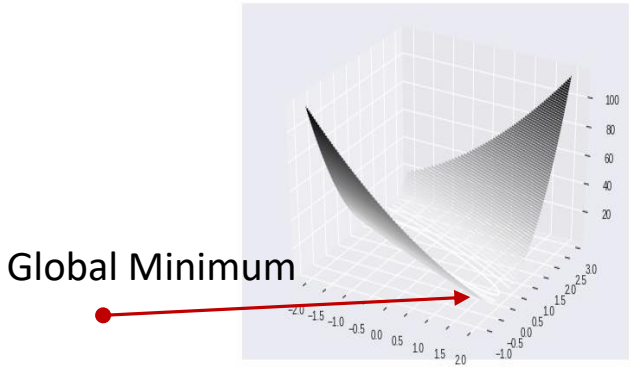
이제 Cost 함수를 최소화하는 W를 Gradient Descent Algorithm으로 구해보자.

$$Cost(W) = \frac{1}{n}\sum[-y(logH(X)) - (1-y)log(1-H(X))]$$

$$W := W - \alpha\frac{\partial}{\partial W}Cost(W)$$

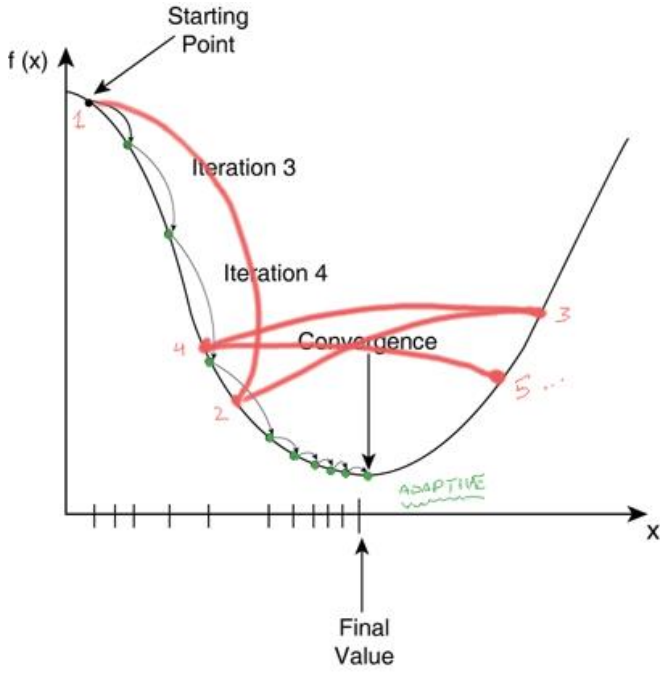
Gradient Descending Algorithm은 초기값에서 출발하여 1 step에 Learning Rate 속도로 이동하여 해로 다가가는 알고리즘이다. 일반적으로 초기값은 난수로 지정하고, Learning Rate는 대강 지정한 다음, 예를 들어, 10,000회 이동하라는 명령을 한다. 이러한 명령으로 해를 찾을 때, 어떤 이슈가 있을까?

초기값이 해로부터 멀면 해에 주어진 반복에서 수렴하지 못할 수 있다.
learning rate가 너무 크면 해를 지나쳐 버릴 수 있고 반대로 너무 작으면 해에 도달하지 못하거나 도달하는데 많은 시간이 소요된다.



10 Learning Rate

아래의 그림에서 빨강색은 Learning Rate가 너무 커서 해를 찾지 못하는 경우이고,
연두색은 적절한 Learning Rate 를 사용하여 효과적으로 접근하는 그림이다.
실제 문제에서는 여러 개의 Learning Rate를 이용하여 시행착오법으로 적절한 Learning Rate를 찾을 수 있다.



10 숙제

H.W #3: Logistic Regression에서 손실함수의 도함수가 아래와 같음을 보이시오.

$$Cost(w,b) = -\sum \left[y_i log(S(wx_i + b)) + (1 - y_i) log(1 - S(wx_i + b)) \right]$$

$$\frac{\partial}{\partial b} Cost(w,b) = -\sum \left[y_i - S(wx_i + b) \right]$$

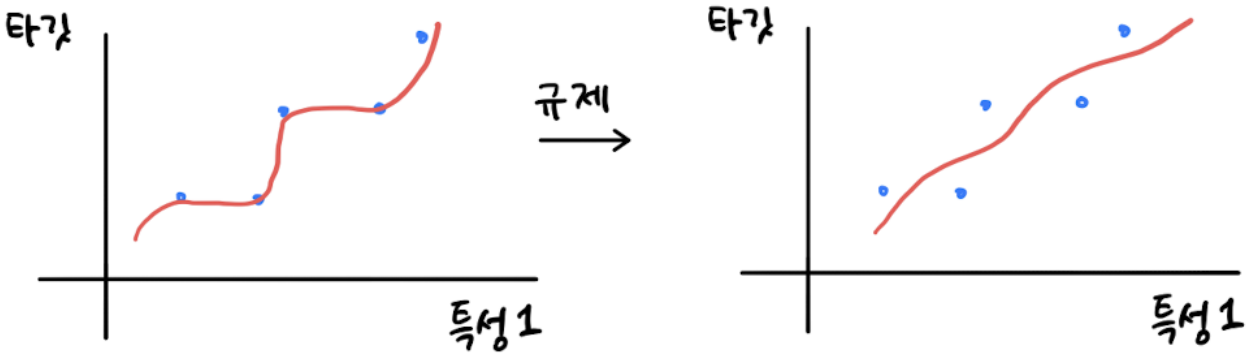
$$\frac{\partial}{\partial w} Cost(w,b) = -\sum \left[y_i - S(wx_i + b) \right] x_i$$

H.W #4: 위의 도함수를 이용해 Gradient Descent 알고리즘으로 beta를 구하시오.
학습률 람다는 0.1로 하고 2000 회 반복 하강한다.

$$X = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}, Y = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \beta = \begin{pmatrix} b \\ w \end{pmatrix} \quad b = b - \lambda \frac{\partial}{\partial b} Cost(w,b), w = w - \lambda \frac{\partial}{\partial w} Cost(w,b)$$

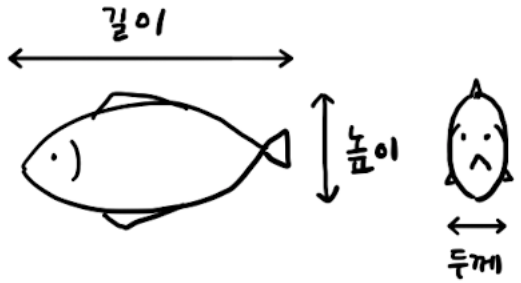
11 Overfitting 규제

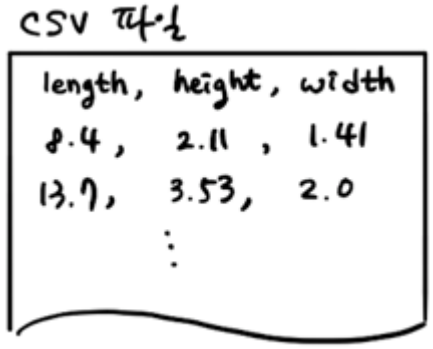
- 회귀 분석에서 n에 비해 p 가 커지면 오버피팅이 발생한다.
- 하지만, 중요한 독립변수를 제거하는 것도 방법이 아니다.
- 때로는 데이터의 수가 독립변수의 수보다 작은 경우도 존재한다.
- 만약, 선형 회귀모형에서 고차다항회귀를 사용하면 모든 점을 다 지나가는 회귀곡선을 만들 수 있는데 이것은 오버피팅이다.



11 Overfitting 규제

- 오버피팅을 이해하기 위해 아래 데이터를 회귀분석 해보자.
- 예제는 농어의 길이, 높이, 두께를 독립변수로 하고 타겟을 무게로 하는 데이터다.
- 고차 다항회귀를 위해 세 개 독립변수를 각 변수의 제곱과 변수들 간의 곱을 만들어 총 9개의 독립변수를 만든다.






```
poly = PolynomialFeatures(include_bias=False)

poly.fit(train_input)
train_poly = poly.transform(train_input)
```

```
poly.get_feature_names()
['x0', 'x1', 'x2', 'x0^2', 'x0 x1',
 'x0 x2', 'x1^2', 'x1 x2', 'x2^2']
```

11 Overfitting 규제

- 9개의 독립변수로 회귀를 수행한 결과, 훈련 데이터의 결정계수는 0.99 이고 테스트 데이터 결정계수는 0.97이 나왔다.
- 훈련 데이터에 비해 테스트 데이터의 결정계수가 낮은 것이 약간 이슈다.
- 좀 더 오버피팅 상황을 만들기 위해 5차 다항식을 추가하고 모형을 진행한 결과, 훈련 데이터의 결정계수는 0.999이고 테스트 데이터 결정계수는 -144가 나왔다.
- 결정계수가 음수인 것이 이해가 안된다.
- R^2 가 음수가 되는 경우는 SSE가 SST보다 커지는 경우다.
- 이 경우는 회귀추정이 평균보다 안 맞는 경우를 말한다.

$$R^2 = SSR/SST = 1 - SSE/SST = 1 - \frac{\sum \left(y_i - \hat{y}\right)^2}{\sum \left(y_i - \overline{y}\right)^2}$$

11 Overfitting 규제

R^2의 수리적 이해

$$\sum(Y_i - \bar{Y})^2 = \sum(Y_i - \hat{Y}_i + \hat{Y}_i - \bar{Y})^2$$
$$= \sum(Y_i - \hat{Y}_i)^2 + \sum(\hat{Y}_i - \bar{Y})^2 + 2\sum(Y_i - \hat{Y}_i)(\hat{Y}_i - \bar{Y})$$
$$= \sum(Y_i - \hat{Y}_i)^2 + \sum(\hat{Y}_i - \bar{Y})^2$$

$$SST = SSE + SSR, R^2 = \frac{SSR}{SST}$$

$$SST = \sum(Y_i - \bar{Y})^2 \quad SSE = \sum(Y_i - \hat{Y}_i)^2 \quad SSR = \sum(\hat{Y}_i - \bar{Y})^2$$

$$\sum(Y_i - \hat{Y}_i)(\hat{Y}_i - \bar{Y})$$
$$= \sum \hat{Y}_i e_i - \bar{Y}\sum e_i = \sum \hat{Y}_i e_i = \sum(\hat{\beta}_0 + \hat{\beta}_1 x_i)e_i = \hat{\beta}_0 \sum e_i + \hat{\beta}_1 \sum x_i e_i$$
$$= \widehat{\beta_1}\sum x_i(y_i - \widehat{\beta_0} - \widehat{\beta_1}x_i) = \widehat{\beta_1}(\sum x_i y_i - \widehat{\beta_0}\sum x_i - \widehat{\beta_1}\sum x_i^2)$$
$$= \widehat{\beta_1}(\sum x_i y_i - (\bar{y} - \widehat{\beta_1}\bar{x})n\bar{x} - \widehat{\beta_1}\sum x_i^2) = \widehat{\beta_1}\left((\sum x_i y_i - n\bar{x}\bar{y}) - \widehat{\beta_1}(\sum x_i^2 - n\bar{x}^2)\right)$$
$$= \widehat{\beta_1}\left(\sum(x_i - \bar{x})(y_i - \bar{y}) - \widehat{\beta_1}(\sum(x_i - \bar{x})^2)\right) = 0$$

11 Overfitting 규제

- 이에 대한 해결책으로 SSE에 베타 제곱 합이 커지는 것을 방지하는 규제항을 더한 Ridge Regression을 사용할 수 있다.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

- Kernel Regularization의 의미를 좀 더 살펴 보자.
- $Y=f(x)+\epsilon, \hat{Y}=\hat{f}(x)$ 이라고 하자.

$$E\left[\left(\hat{Y}-Y\right)^2\right]=E\left(\hat{Y}^2\right)+E(Y^2)-2E\left(\hat{Y}\cdot Y\right)$$

$$=V\left(\hat{Y}\right)+E\left(\hat{Y}\right)^2+V(Y)+E(Y)^2-2E\left(\hat{Y}\cdot Y\right)$$

$$\because 2E\left(\hat{Y}\cdot Y\right)=2E\left(\hat{Y}(f(x)+\epsilon)\right)=2E\left(\hat{Y}\cdot f(x)\right)+2E\left(\hat{Y}\cdot\epsilon\right)=2f(x)E\left(\hat{Y}\right)$$

$$=V\left(\hat{Y}\right)+V(Y)+E\left(\hat{Y}\right)^2+E(Y)^2-2f(x)E(\hat{Y})$$

$$=V\left(\hat{Y}\right)+\left(E\left(\hat{Y}\right)-f(x)\right)^2+V(Y)$$

모형추정분산 $bias^2$ σ^2

- 만약, 모형선택이 정확했다면, $E\left(\hat{Y}\right)=f(x)$ 이므로 bias=0 이 된다.

11 Overfitting 규제

- 베타 제곱 대신 절대값을 사용하는 아이디어가 Lasso Regression 이다.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| \;=\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|.$$

- Ridge와 Lasso를 조합하여 회귀계수를 구하는 아이디어가 Elastic Net 이다.
- 여기서, α_1, α_2는 조절모수다.

$$\hat{\beta}(\alpha)=\underset{\beta}{argmin}[\;\frac{\sum_{i=1}^{n}(y_i-(\beta_i x_i+\beta_0))^2}{n}+\alpha_1\sum_{j=1}^{k}|\beta_j|+\alpha_2\sum_{j=1}^{k}(\beta_j)^2\;]$$

Ridge_Lasso Regression.ipynb