다중선형회귀 (Multiple Linear Regression) : 수치형 설명변수 X와 연속형 숫자로 이뤄진 종속변수 Y간의 관계를 선형으로 가정하고 이를 잘 표현할 수 있는 회귀계수를 데이터로부터 추정하는 모델.

(Y = a1x1 + a2x2 + ... + b)

N개의 데이터, K개의 설명변수($x_{11,\cdots,}x_{nk}$), K개의 회귀계수($\beta_0, \cdots, \beta_k$)

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1k} \\ 1 & x_{21} & \cdots & x_{2k} \\ \cdots & \cdots & & \cdots \\ 1 & x_{n1} & \cdots & x_{nk} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \cdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \cdots \\ \varepsilon_n \end{bmatrix}$$

$$\vec{y} = X\vec{\beta} + \vec{\varepsilon}, \quad \vec{\varepsilon} \sim N(E(\vec{\varepsilon}), V(\vec{\varepsilon}))$$

$$E(\vec{\varepsilon}) = \begin{bmatrix} 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}, \quad V(\vec{\varepsilon}) = \sigma^2 I$$

회귀 계수 결정법 - Direct Solution :

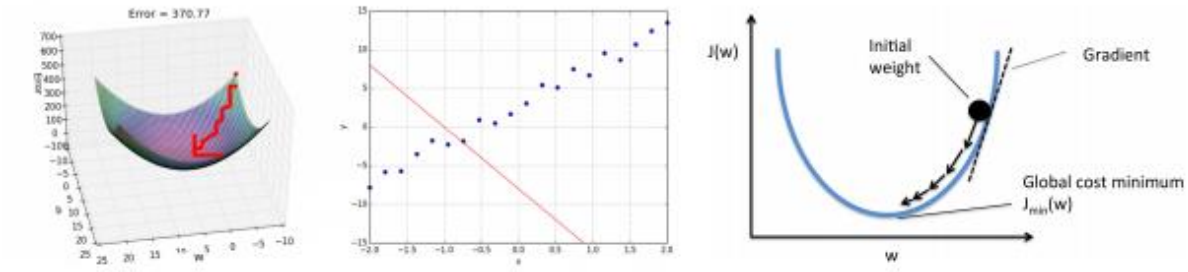
선형회귀 계수들은 실제값(y)와 모델 예측값(y')의 차이, 오차제곱합(error sum of squares)를 최소로 하는 값을 회귀 계수로 선정.

최적의 계수들은 오차제곱합을 회귀 계수에 대해 미분한 식을 0으로 놓고 풀면 명시적인 해를 구할 수 있음. (2차 방정식 최소값 문제)

회귀 계수 결정법 - Numerical Search :

경사하강법(gradient descent)같은 반복적인 방식으로 선형회귀 계수를 구할 수 있음.

경사하강법이란 어떤 함수 값(목적 함수, 비용 함수, 오차값)을 최소화하기 위해 임의의 시작점을 잡은 후 해당 지점에서의 그래디언트(경사)를 구하고, 그래디언트의 반대 방향으로 조금씩 (Learning rate 만큼) 이동하는 과정을 여러 번 반복하는 것. (최소값의 우측의 기울기는 음수, 좌측은 양수이기 때문)

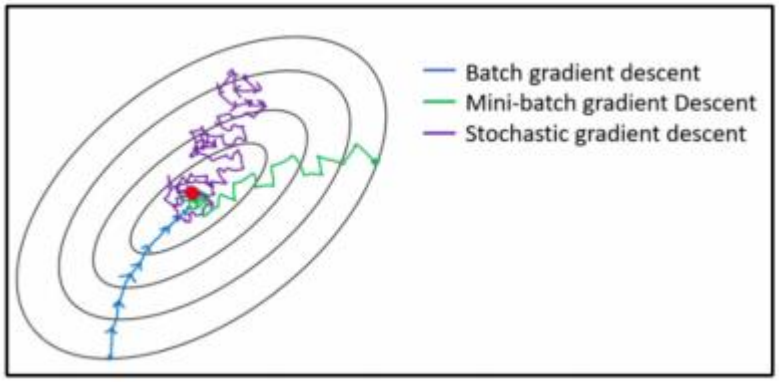


경사하강법의 종류

Batch Gradient Descent (GD) : 파라미터를 업데이트 할 때마다 모든 학습 데이터를 사용하여 cost function의 gradient를 계산. Vanilla Gradient Descent라 불림. 매우 낮은 학습 효율을 보일 수 있다. 1 epoch 마다 모든 데이터(Batch)를 계산 하기 때문에

Stochastic Gradient Descent (SGD) : 파라미터를 업데이트 할 때, 무작위로 샘플링된 학습 데이터를 하나씩만 이용하여 cost function의 gradient를 계산. 모델을 자주 업데이트 하며, 성능 개선 정도를 빠르게 확인 가능. Local minima에 빠질 가능성을 줄일 수 있음. 최소 Cost에 수렴했는지의 판단이 상대적으로 어렵다.

Mini Batch Gradient Descent : 파라미터를 업데이트 할 때마다 일정량의 일부 데이터를 무작위로 뽑아 계산. SGD의 노이즈를 줄이면서, GD의 전체 배치보다 효율적. 널리 사용되는 기법.

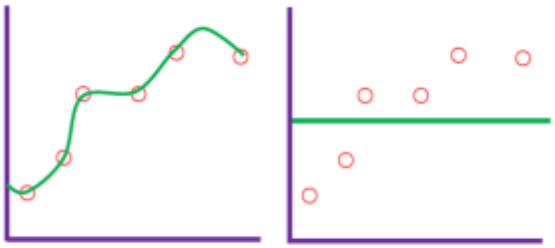


Solver 옵션을 보면 뭘 쓰는지

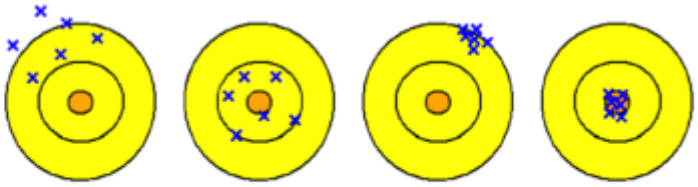
정규화 (regularization) : 회귀계수가 가질 수 있는 값에 제약조건을 부여하여 미래 데이터에 대한 오차 해결 기대. 미래 데이터에 대한 오차의 기대 값은 모델의 Bias와 variance로 분해가능. 정규화는 variance를 감소시켜 일반화 성능을 높이는 기법. 단, 이 과정에서 bias가 증가할 수 있음

왼쪽 그림은 학습데이터를 정말 잘 맞추고 있지만 미래 데이터가 조금만 바뀌어도 예측 값이 들쭉날쭉할 수 있음 overfitting

오른쪽 그림은 가장 강한 수준의 정규화를 수행한 결과로 학습데이터에 대한 설명엮을 다소 포기하는 대신 미래 데이터 변화에 상대적으로 안정적인 결과를 나타냄.



정규화의 결과를 직관적으로 나타낸 그림



Bias	High	Low	High	Low
Variance	High	High	Low	Low

Bias-Variance Decomposition : 일반화 (generalization) 성능을 옾이는 정규화(Regularization), 앙상블(Ensemble) 기법의 이론적 배경. 학습에 쓰지 않은 미래 데이터에 대한 오차의 기대값을 모델의 Bias와 Vatiance로 분해하는 내용.

Bias-Variance의 직관적인 이해 :

첫번째 그림을 보면 예측값(파란 x)의 평균이 과녁 (Truth)와 멀리 떨어져 있어 Bias가 크고, 예측값들이 서로 멀리 떨어져 있어 Variance 또한 큼 (사격 폐급)

네번째 그림의 경우 Bias, Bariance 모두 작음. 제일 이상적임 (사격 특급)

부스팅(Boosting)은 Bias를 줄여 성능을 높이고, 라쏘회귀(Lasso regression)는 Variance를 줄여 성능을 높이는 기법임.

예시 1

성인 여성 33명의 나이와 혈압 데이터. 오차제곱합을 최소로 하는 회귀 계수 계산 결과 및 분석

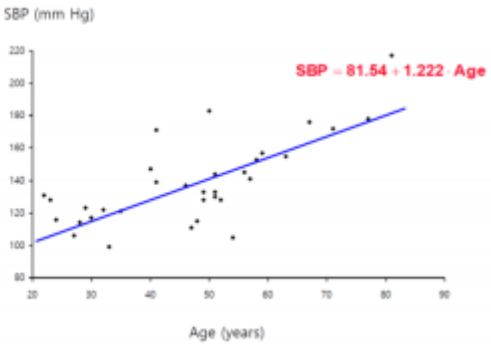
SBP = 81.54 + 1.222AGE

나이라는 변수에 대응하는 계수는 1.222인데 이는 나이 한 살 먹을 때마다 혈압이 1.222mm/Hg 만큼 증가한다는 결과를 보여줌

Age	SBP
22	131
23	128
24	116
27	106
28	114
29	123
30	117
32	122
33	99
35	121
40	147

Age	SBP
41	139
41	171
46	137
47	111
48	115
49	133
49	128
50	183
51	130
51	133
51	144

Age	SBP
52	128
54	105
56	145
57	141
58	153
59	157
63	155
67	176
71	172
77	178
81	217



예시2

성인 여성 33명의 나이와 암 발병 데이터. 오차제곱합을 최소로 하는 회귀 계수 계산 결과 및 분석

다중선형회귀 모델 적용 불가.

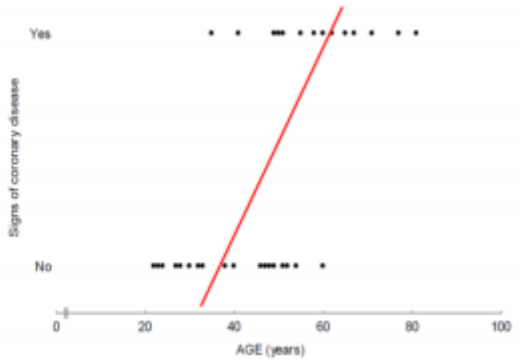
범주형 숫자 (암 발병 여부)는 연속형 숫자 (혈압) 과 달리 의미가 없음. 즉 0(정상)과 1(발병)을

바꿔도 상관 없음 -> 범주형 숫자는 로지스틱 회귀 모델 적용 가능

Age	CD
22	0
23	0
24	0
27	0
28	0
30	0
30	0
32	0
33	0
35	1
38	0

Age	CD
40	0
41	1
46	0
47	0
48	0
49	1
49	0
50	1
51	0
51	1
52	0

Age	CD
54	0
55	1
58	1
60	1
60	0
62	1
65	1
67	1
71	1
77	1
81	1

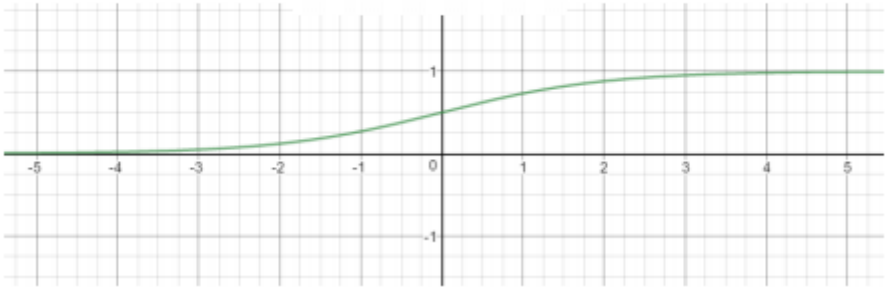


로지스틱 함수 (Logistic functin) : 아래 그림과 같이 S-커브 함수. 실제 많은 자연, 사회 현상에서는 특정 변수에 대한 확률 값이 선형이 아닌 S-커브 형태를 따르는 경우가 많음.

X값으로 어떤 값이든 받을 수가 있지만 출력 결과(y)는 항상 0에서 1 사이 값이 됨 (확률 밀도 함수 (probability density function) 요건을 충족.

시그모이드 함수라고도 부름

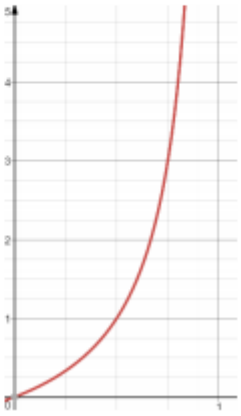
$$y = \frac{1}{1 + e^{-x}}$$



승산 (Odds) : 임의의 사건 A가 발생하지 않을 확률 대비 일어날 확률의 비율.

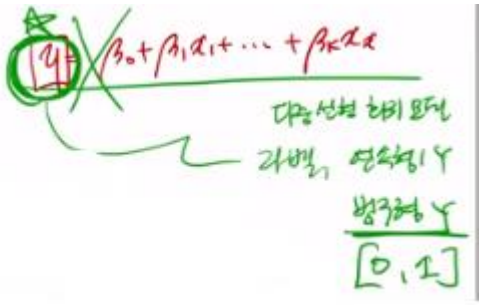
$$odds = \frac{P(A)}{P(A^c)} = \frac{P(A)}{1 - P(A)}$$

P(A)가 1에 가까울 수록 승산은 커지고, 반대로 P(A)가 0이라면 승산은 0



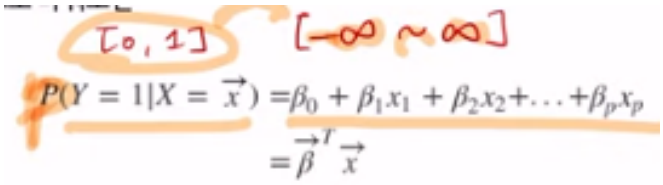
이항 로지스틱 회귀

Y가 범주형일 경우, 다중 선형 회귀 모델을 적용할 수 없음.



y가 범주 형이니까 바꿔 보자.

Y를 확률식으로 바꿔보면



왼쪽은 [0, 1]로 바운드 되는데, 오른쪽은 무한대 라서 레벨이 서로 안맞는다..

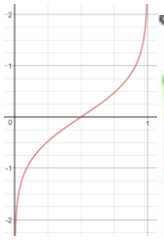
Y를 승산으로 바꿔보면

$$\frac{P(Y=1|X=\vec{x})}{1-P(Y=1|X=\vec{x})} = \vec{\beta}^T\vec{x}$$

[0, ∞] [-∞, ∞]

아직 안맞는다.

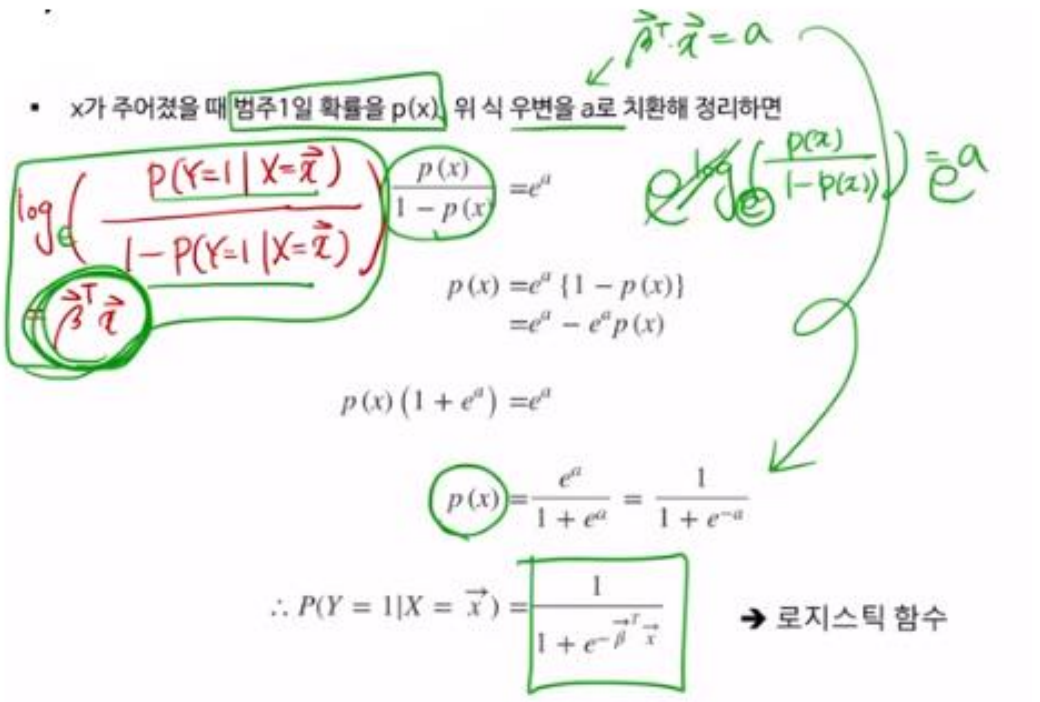
Y 승산에 로그를 취했더니 !!



$$\log\left(\frac{P(Y=1|X=\vec{x})}{1-P(Y=1|X=\vec{x})}\right) = \vec{\beta}^T\vec{x}$$

[-∞, ∞] [-∞, ∞]

이제 이걸 정리해보면



이항 로지스틱 회귀의 결정 경계

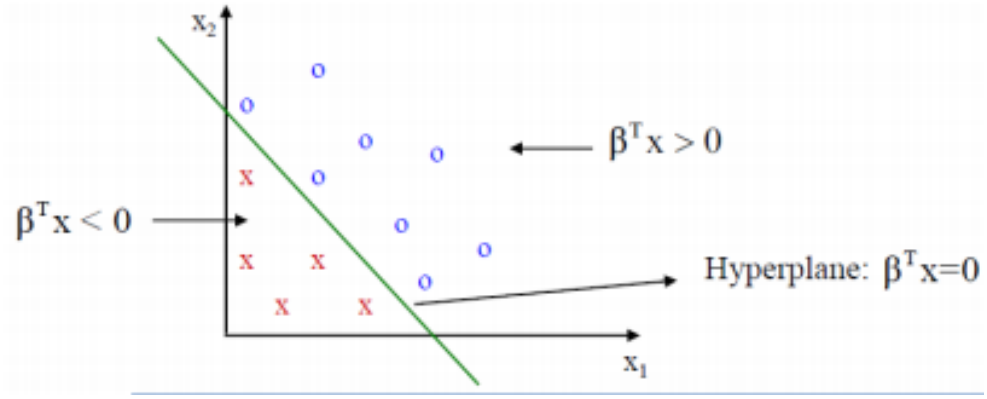
이항 로지스틱 모델은 범주 정보를 모르는 입력 벡터 x를 넣으면 범주 1에 속할 확률을 반환하며, 범주 1로 분류하는 판단 기준은 아래와 같다.

$$P(Y = 1|X = \vec{x}) > P(Y = 0|X = \vec{x})$$

범주가 두 개 뿐이므로, 위 식 좌변을 p(x)로 치환하면,

$$p(x) > 1 - p(x)$$
$$\frac{p(x)}{1 - p(x)} > 1$$
$$\log \frac{p(x)}{1 - p(x)} > 0$$
$$\therefore \vec{\beta}^T \vec{x} > 0$$

마찬가지로 $\beta^T x < 0$ 이면 데이터 범주를 0으로 분류하게 되며, 로지스틱 결정 경계 (decision boundary)는 $\beta^T x = 0$ 인 하이퍼플레인 (hyperplane) 입니다.



Classifier

$$y = \frac{1}{\left(1 + \exp(-\beta^T x)\right)}$$

$$\begin{pmatrix} y \to 1 & if & \beta^T x \to & \infty \\ y = \frac{1}{2} & if & \beta^T x = & 0 \\ y \to 0 & if & \beta^T x \to & -\infty \end{pmatrix}$$

다항 로지스틱 회귀

삼항은 하이퍼 플레인 2개가 필요하다.

세번째 범주에 속할 확률 = 1 - 첫번째 범주일 확률 - 두번째 범주일 확률 임을 생각하자.

- **이항 로지스틱 회귀 모델을 통한 다항 로지스틱 회귀 문제 풀기**

$$\log \frac{P(Y=1|X=\vec{x})}{P(Y=3|X=\vec{x})} = \beta_1^T \vec{x}$$

$$\log \frac{P(Y=2|X=\vec{x})}{P(Y=3|X=\vec{x})} = \beta_2^T \vec{x}$$

- **세번째 범주에 속할 확률 = 1 – 첫번째 범주에 속할 확률 – 두번째 범주에 속할 확률**

$$P(Y=1|X=\vec{x}) = \frac{e^{\vec{\beta_1}^T \vec{x}}}{1 + e^{\vec{\beta_1}^T \vec{x}} + e^{\vec{\beta_2}^T \vec{x}}}$$

$$P(Y=2|X=\vec{x}) = \frac{e^{\vec{\beta_2}^T \vec{x}}}{1 + e^{\vec{\beta_1}^T \vec{x}} + e^{\vec{\beta_2}^T \vec{x}}}$$

$$P(Y=3|X=\vec{x}) = \frac{1}{1 + e^{\vec{\beta_1}^T \vec{x}} + e^{\vec{\beta_2}^T \vec{x}}}$$

- **K개 범주를 분류하는 다항로지스틱 회귀 모델의 입력 벡터 x가 각 클래스로 분류될 확률**

$$P(Y=k|X=\vec{x}) = \frac{e^{\vec{\beta_k}^T \vec{x}}}{1 + \sum_{i=1}^{K-1} e^{\vec{\beta_i}^T \vec{x}}} \quad (k = 0, 1, \ldots, K-1)$$

$$P(Y=K|X=\vec{x}) = \frac{1}{1 + \sum_{i=1}^{K-1} e^{\vec{\beta_i}^T \vec{x}}}$$

- **'로그승산'으로 된 좌변을 '로그확률'로 변경**

$$P(Y=1|X=\vec{x}) = \frac{e^{\vec{\beta_1}^T \vec{x}}}{1 + e^{\vec{\beta_1}^T \vec{x}} + e^{\vec{\beta_2}^T \vec{x}}}$$

$$P(Y=2|X=\vec{x}) = \frac{e^{\vec{\beta_2}^T \vec{x}}}{1 + e^{\vec{\beta_1}^T \vec{x}} + e^{\vec{\beta_2}^T \vec{x}}}$$

$$P(Y=3|X=\vec{x}) = \frac{1}{1 + e^{\vec{\beta_1}^T \vec{x}} + e^{\vec{\beta_2}^T \vec{x}}}$$

$$\Rightarrow$$

$$\log P(Y=1|X=\vec{x}) = \vec{\beta_1}^T \vec{x} - \log Z$$

$$\log P(Y=2|X=\vec{x}) = \vec{\beta_2}^T \vec{x} - \log Z$$

$$\ldots$$

$$\log P(Y=K|X=\vec{x}) = \vec{\beta_K}^T \vec{x} - \log Z$$

- **로그 성질을 활용해 c번째 범주에 속할 확률을 기준으로 식을 정리**

$$\log P(Y=c) + \log Z = \vec{\beta_c}^T \vec{x}$$

$$\log \{P(Y=c) \times Z\} = \vec{\beta_c}^T \vec{x}$$

$$P(Y=c) \times Z = e^{\vec{\beta_c}^T \vec{x}}$$

$$P(Y=c) = \frac{1}{Z} e^{\vec{\beta_c}^T \vec{x}}$$

$$\Rightarrow$$

$$P(Y=c) = \frac{e^{\vec{\beta_c}^T \vec{x}}}{\sum_{k=1}^{K} e^{\vec{\beta_k}^T \vec{x}}}$$

- **전체 확률 합은 1**

$$1 = \sum_{k=1}^{K} P(Y=k) = \sum_{k=1}^{K} \frac{1}{Z} e^{\vec{\beta_k}^T \vec{x}} = \frac{1}{Z} \sum_{k=1}^{K} e^{\vec{\beta_k}^T \vec{x}} \qquad \therefore Z = \sum_{k=1}^{K} e^{\vec{\beta_k}^T \vec{x}}$$