이론과 실습, 두마리 토끼를 모두 잡는 세상에서 제일 쉬운 머신러닝 수업

기본모델(Regression, SVM, Decision Tree, Naive Bayes)부터
최신모델(Xgboost 등)까지

01

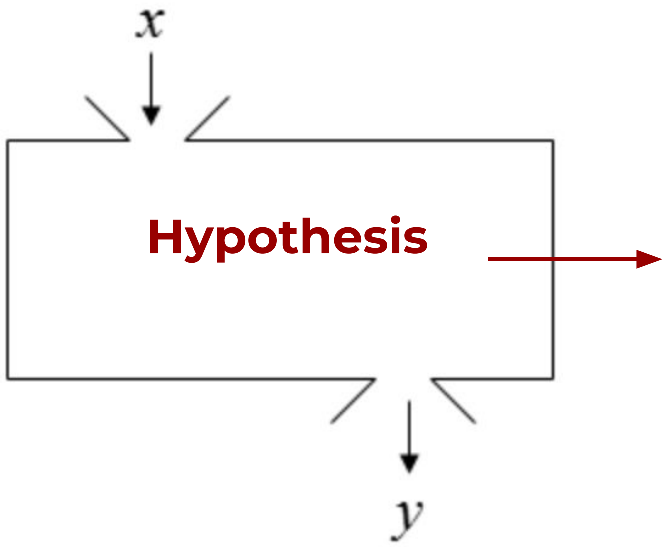
Linear Regression

Ordinary Least Square

01

Hypothesis

Linear Regression은 OLS(Ordinary Least Square) 즉, 최소제곱법이라고도 불리는 가장 오래된 회귀용 선형 알고리즘이다. Linear Regression의 Hypothesis는 다음과 같다.

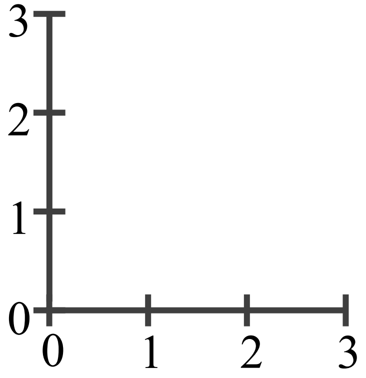


$$w[0] + w[1]*x[1] + w[2]*x[2] + w[3]*x[3] + \dots + w[n]*x[n]$$

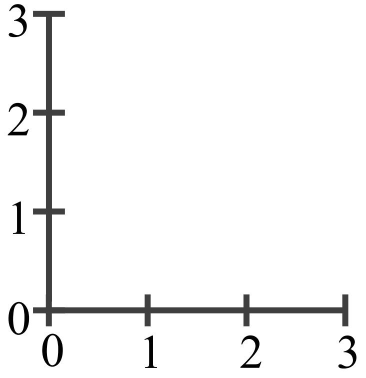
01

Hypothesis

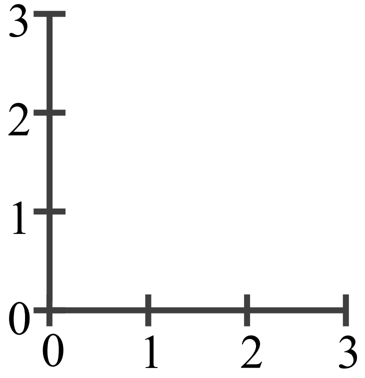
$$
\begin{aligned}
&H(x) = WX \\
\leftrightarrow\ &H(x) = Wx + b \\
\leftrightarrow\ &H(x) = W_1x_1 + W_0 \\
\leftrightarrow\ &H_\theta(x) = \theta_1x_1 + \theta_0
\end{aligned}
$$



$\theta_0 = 2$
$\theta_1 = 0$



$\theta_0 = 0$
$\theta_1 = 1.5$

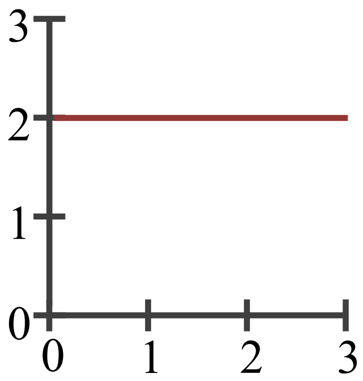


$\theta_0 = 1$
$\theta_1 = 0.5$

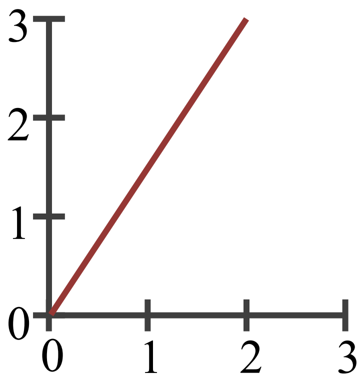
01

Hypothesis

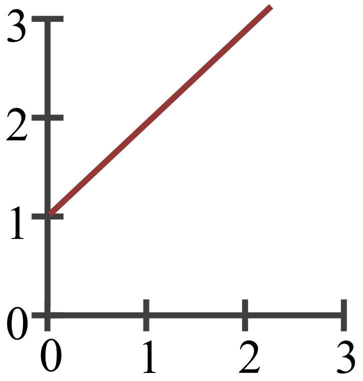
$$H(x) = WX$$
$$\leftrightarrow\ H(x) = Wx + b$$
$$\leftrightarrow H(x) = W_1x_1 + W_0$$
$$\leftrightarrow H_\theta(x) = \theta_1x_1 + \theta_0$$



$\theta_0 = 2$
$\theta_1 = 0$



$\theta_0 = 0$
$\theta_1 = 1.5$



$\theta_0 = 1$
$\theta_1 = 1$

02
Cost

H(x)가 y(target value)에 가까워지도록 만들기 위한 weights를 선택해야 한다. 이를 간단한 수식으로 표현하자면 다음과 같다.

$$\min_{\theta_1,\theta_0} \frac{1}{2m}\sum_{i=1}^{m}(H_\theta(x^{(i)}) - y^{(i)})^2$$

즉, Cost함수는 다음과 같다.

$$J(\theta_1, \theta_0) = \frac{1}{2m}\sum_{i=1}^{m}(H_\theta(x^{(i)}) - y^{(i)})^2$$

03
Linear Regression with Multi-variables

$$H(x) = WX$$
$$\leftrightarrow H(x) = W_0 + W_1 x_1 + W_2 x_2 + W_3 x_3 + \cdots + W_D x_D$$
$$\leftrightarrow H(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_D x_D$$

$$\mathrm{J}(\theta_D, \ldots, \theta_3, \theta_2, \theta_1, \theta_0) = \frac{1}{2m} \sum_{i=1}^{m} (H_\theta\left(x^{(i)}\right) - y^{(i)})^2$$

Hypothesis 함수가 복잡해질 뿐, *Cost* 함수의 틀은 그대로다. 헷갈리지 말자!

04
Regularization

Regularization?? 이토록 단순해보이는 Linear Regression에 또 Regularization을 한다니,
불필요해보인다. 하지만, 사실 상 그렇지 않은 경우가 많다.

다음을 알고 있다면 어렵지 않게 이해할 수 있다.

방정식보다 미지수가 많은 경우, 해는 무수히 많다.

04

Regularization

Regularization?? 이토록 단순해보이는 Linear Regression에 또 Regularization을 한다니,
불필요해보인다. 하지만, 사실 상 그렇지 않은 경우가 많다.

다음을 알고 있다면 어렵지 않게 이해할 수 있다.

방정식(=데이터)보다 미지수(=Weight)가 많은 경우, 해는 무수히 많다.

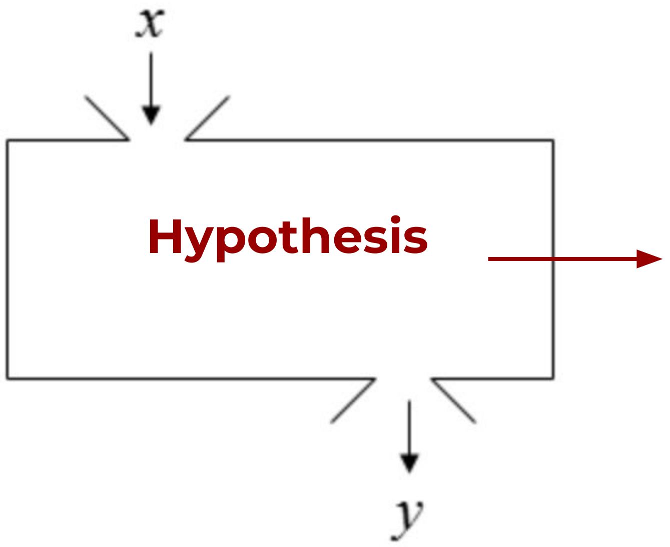
02

Ridge Regression

L2

01
Hypothesis

Ridge Regression은 Linear Regression과 같은 hypothesis를 사용한다. 단, 차이가 있다면 여기서 Weight를 선택하는 기준이 단순히 H(x)가 y(target value)에 가까워지도록 하는 것에 제약 조건을 만족하는 것까지 더해졌다는 것이다.



$$w[0] + w[1]*x[1] + w[2]*x[2] + w[3]*x[3] + \dots + w[n]*x[n]$$

02
Cost

Ridge Regression의 Cost함수는 Linear Regression에 L2 norm의 제한을 걸어준 형태이다.

$$J(\theta_D, \ldots, \theta_3, \theta_2, \theta_1, \theta_0) = \frac{1}{2m}\sum_{i=1}^{m}(H_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum w_i^2$$

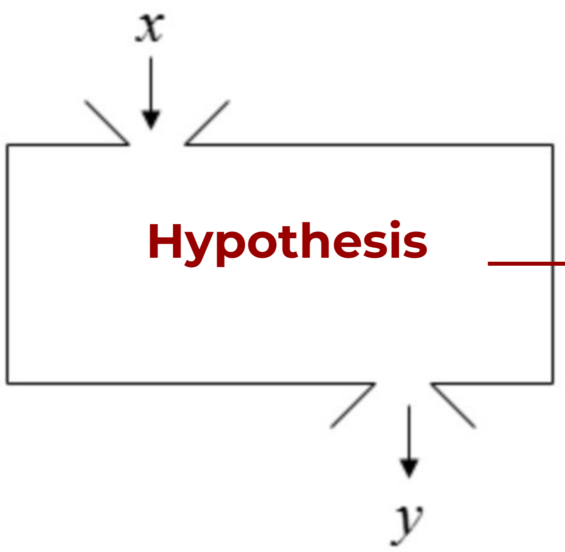
03

Lasso Regression

L1

01
Hypothesis

Lasso Regression은 Ridge와 마찬가지로 Linear Regression과 같은 hypothesis를 사용하며, Weight를 선택하는 기준이 단순히 H(x)가 y(target value)에 가까워지도록 하는 것에 제약 조건을 만족하는 것까지 더해진다.



$$w[0] + w[1]*x[1] + w[2]*x[2] + w[3]*x[3] + \ldots + w[n]*x[n]$$

02
Cost

Lasso Regression의 Cost함수는 Linear Regression에 L1 norm의 제한을 걸어준 형태이다.

$$J(\theta_D, \dots, \theta_3, \theta_2, \theta_1, \theta_0) = \frac{1}{2m}\sum_{i=1}^{m}\left(H_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum |w_i|$$

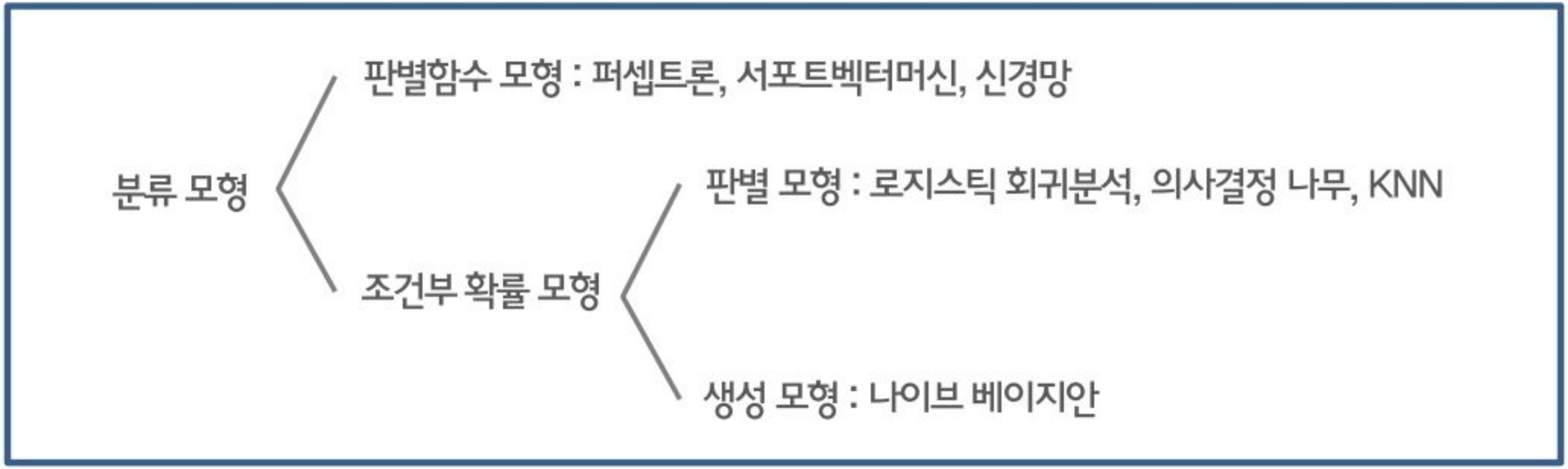
04

Logistic Regression

Sigmoid, Softmax

01
Classification Model

Classification model의 경우, 분류 방식에 따라 다음과 같이 모형을 나눌 수 있다.



■ 분류의 사례

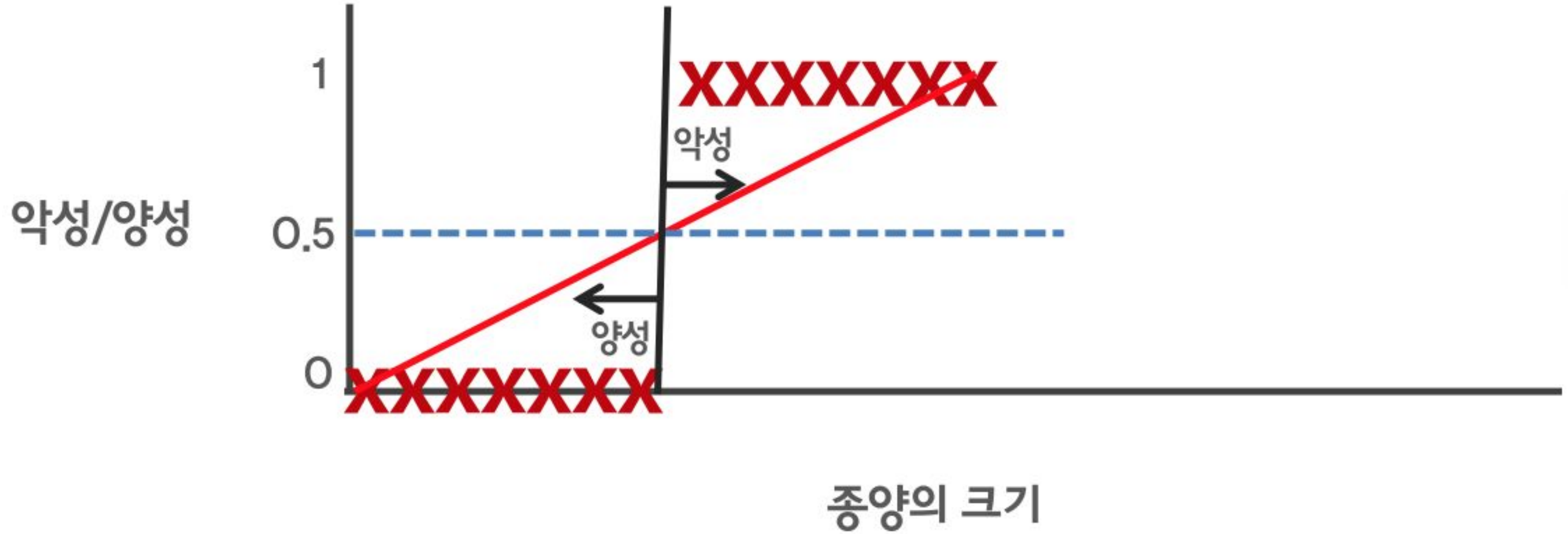
이메일 분류: Spam or Ham

종양 분류: 양성 or 악성

카드 사기 분류: 합법 or 사기

■ 분류의 문제를 Linear Regression으로 풀 수 있을까?

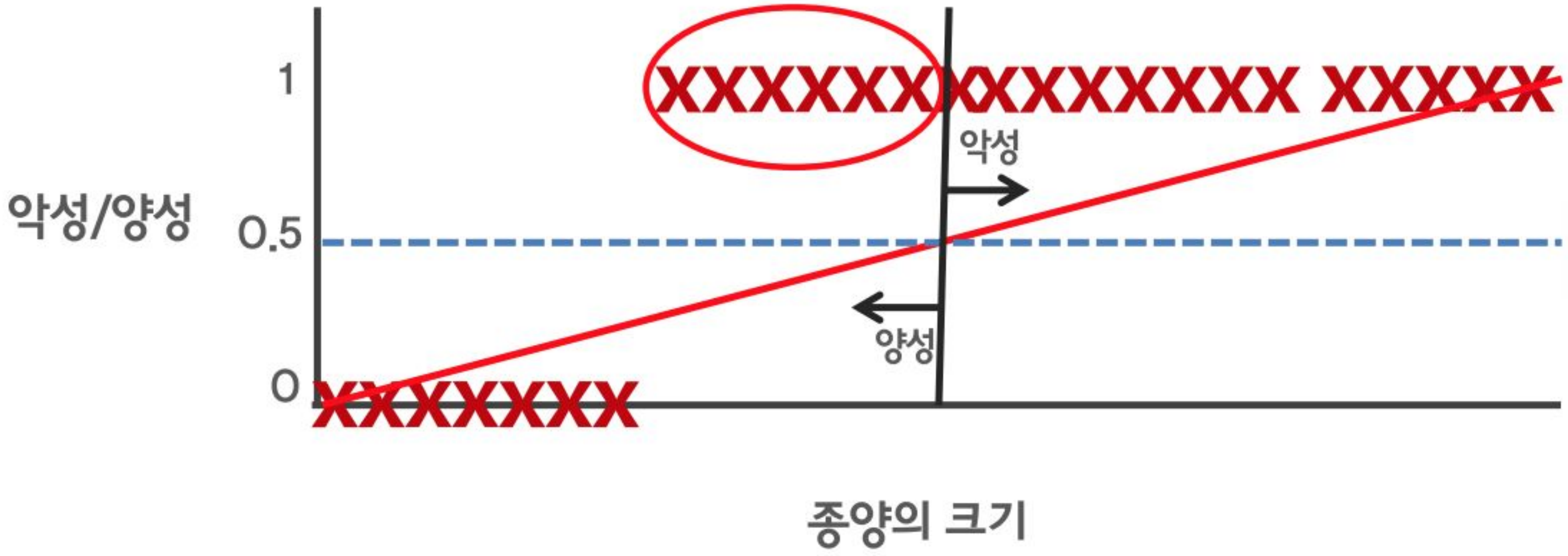
예) 종양의 분류

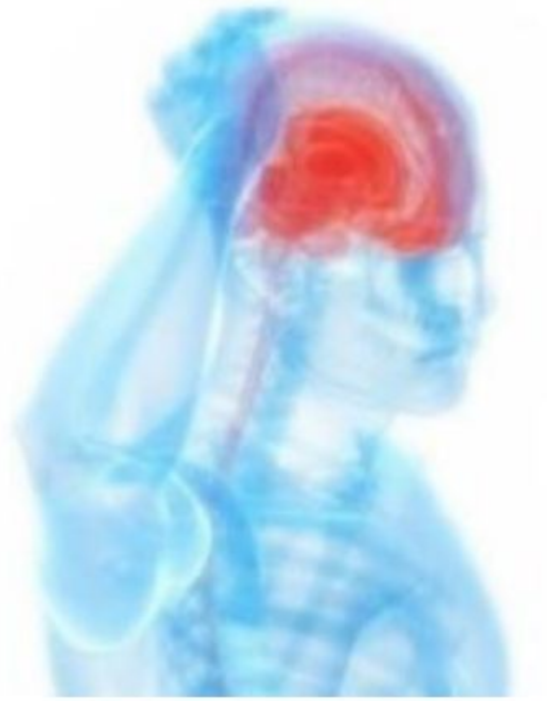


$$\text{If } H(x) \geq 0.5, predict \text{ y=1}$$
$$else, predict \text{ y=0}$$

Q. 그럼 이런 경우도, Linear Regression으로 풀 수 있을까?

악성인데, 양성이라고 판단되는 경우가 발생한다.





02
Hypothesis

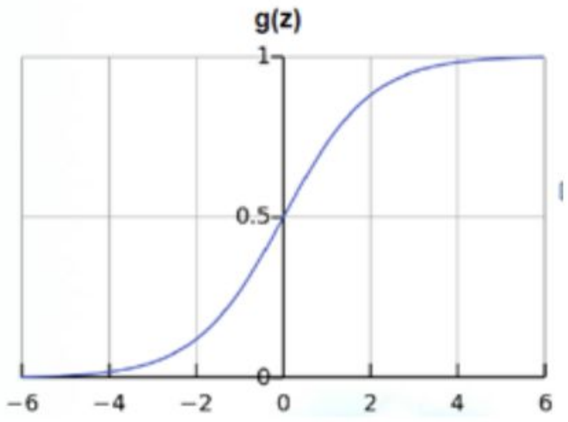
■ 로지스틱 회기 분석(Logistic Regression)

Linear Regression의 Hypothesis 함수 $H(x) = WX$ 는 1 이상의 숫자로 발산하기 때문에 분류 문제에 적합하지 않다. 그래서 이를 0 ~ 1 사이의 값이 나오도록 바꿔주는 함수를 찾고자 했다. 그렇게 여러 연구를 통해 찾은 함수가 바로 시그모이드 함수이다.

■ 시그모이드 함수

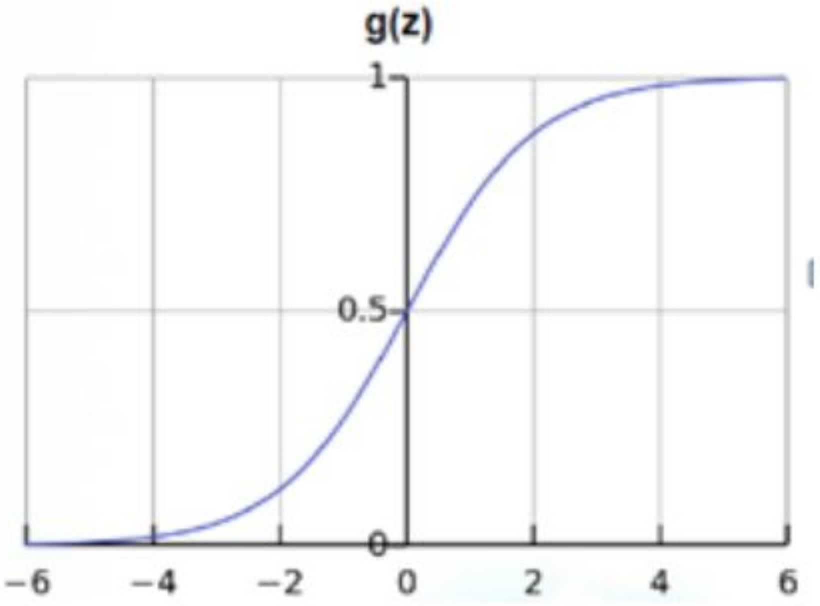
모수 θ는 일반적인 회귀 분석의 종속 변수와 달리 0 부터 1까지의 실수 값만 가질 수 있기 때문에 시그모이드 함수(sigmoid function)이라 불리는 특별한 형태의 함수 f를 사용해야 한다.
시그모이드 함수는 종속 변수의 모든 실수 값에 대해 유한한 구간 (a, b) 사이의 한정된 값과 양의 기울기를 가지는 함수를 말하며 다음과 같은 형태를 나타내는 로지스틱 함수가 주로 사용된다.

$$\text{logitstic}(z) = \frac{1}{1 + \exp(-z)}$$



02
Hypothesis

$$H(x) = g(\theta^T x) = \frac{1}{1 + e^{(-\theta^T x)}}$$



03
Cost

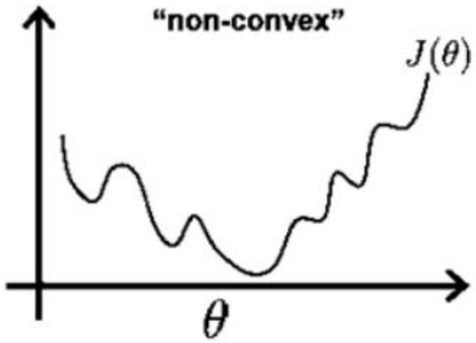
Cost를 계산하는 일반적인 방식으로 보면, 다음과 같다.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(\frac{1}{1+\exp(-\theta^T x^{(i)})} - y^{(i)}\right)^2$$

Non-convex function

➔ Many local min

➔ Gradient descent may not find the global min



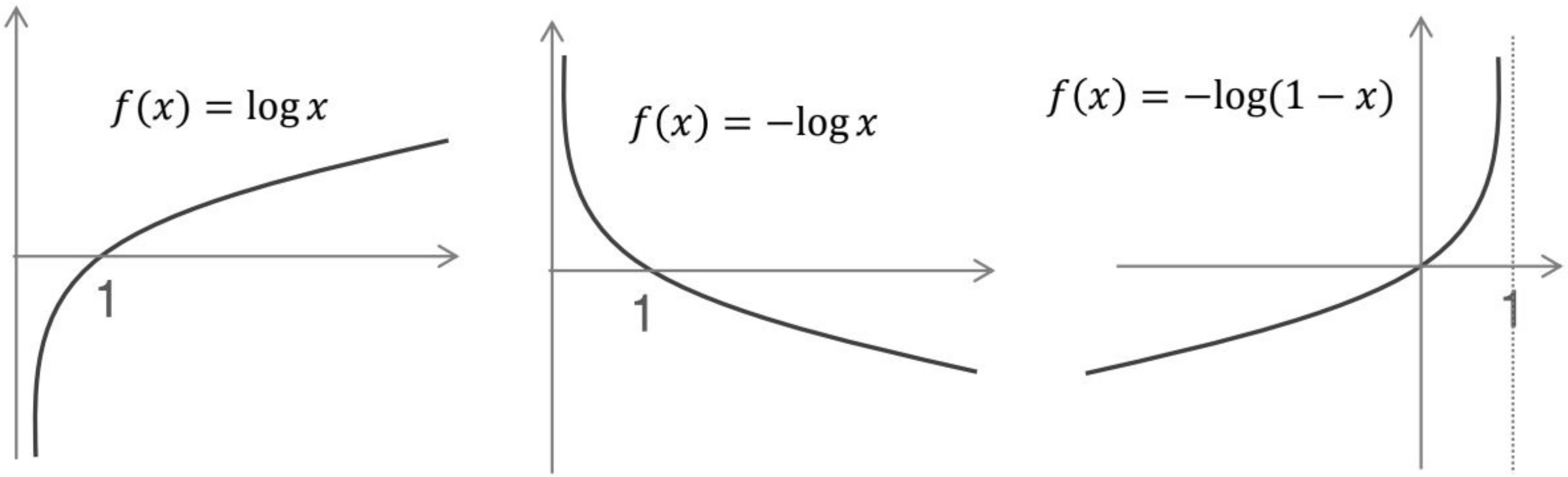
그럼 결론은 복잡한 optimization모델을 사용하는 것이다.

하지만, 이번에는 다른 방식으로 Cost함수를 접근하는 것을 배워보도록 하자.

03
Cost

$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & if \;\; y = 1 \\ -\log(1 - h_\theta(x)) & if \;\; y = 0 \end{cases}$$

■ Log Function



03
Cost

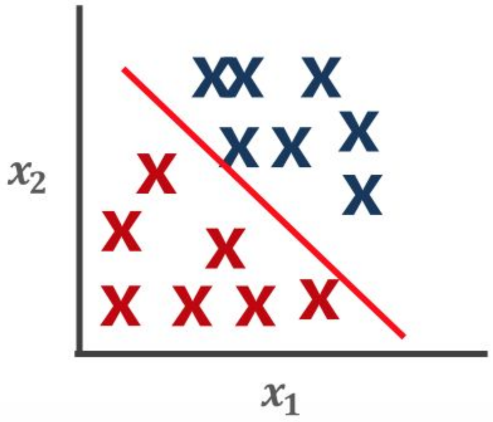
조건문 상태를 없애주는 것이 코딩 구현에서 편리하기 때문에 다음과 같이 바꾼다.

$$Cost(h_\theta(x), y) = -y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$

이렇게 계산되는 Cost값들에 대해 평균을 내는 것이 Cost Function이다.

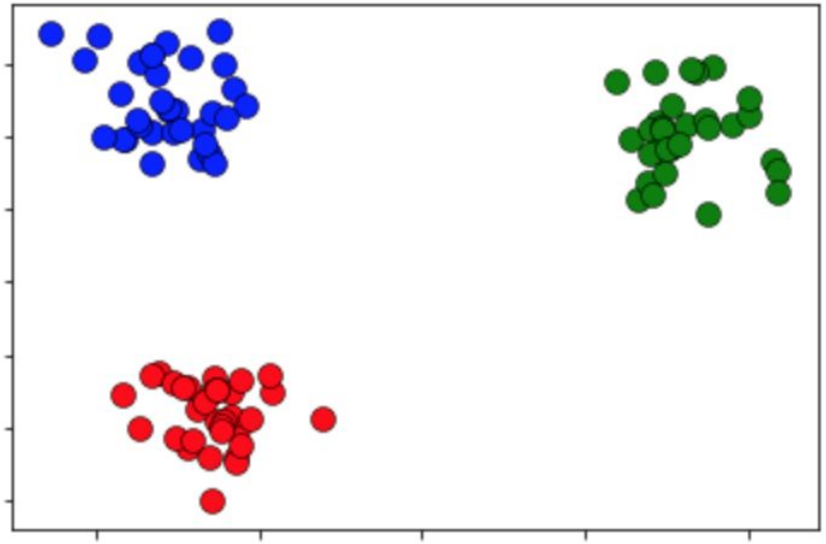
$$J(\theta) = -\frac{1}{m}\sum Cost(h_\theta(x), y) = -\frac{1}{m}\sum -y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$

이를 Gradient Descent 하면 된다. 이 작업을 통해 다음과 같은 Decision Boundary를 찾는 것이다.



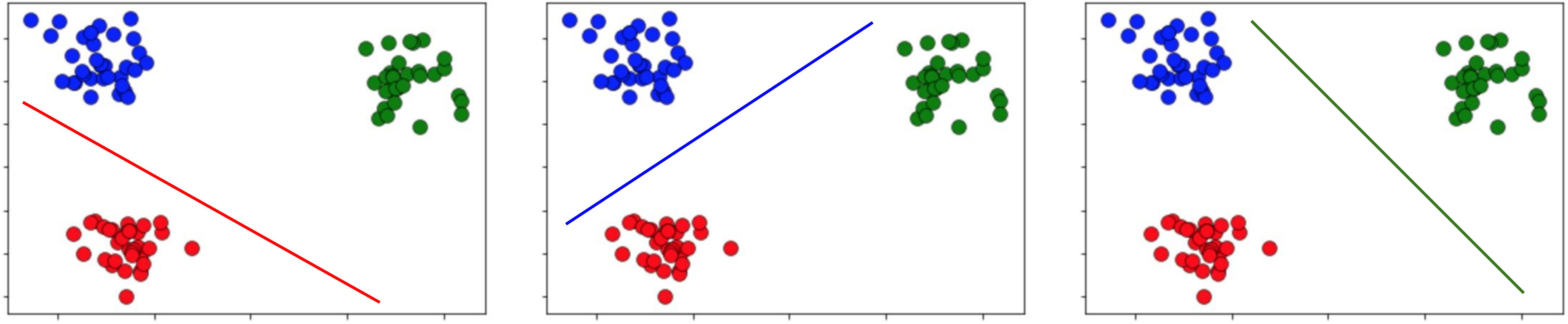
04
Softmax

다음과 같이 이진 분류가 아닌 다중 분류의 경우는 어떻게 해결할 수 있을까?



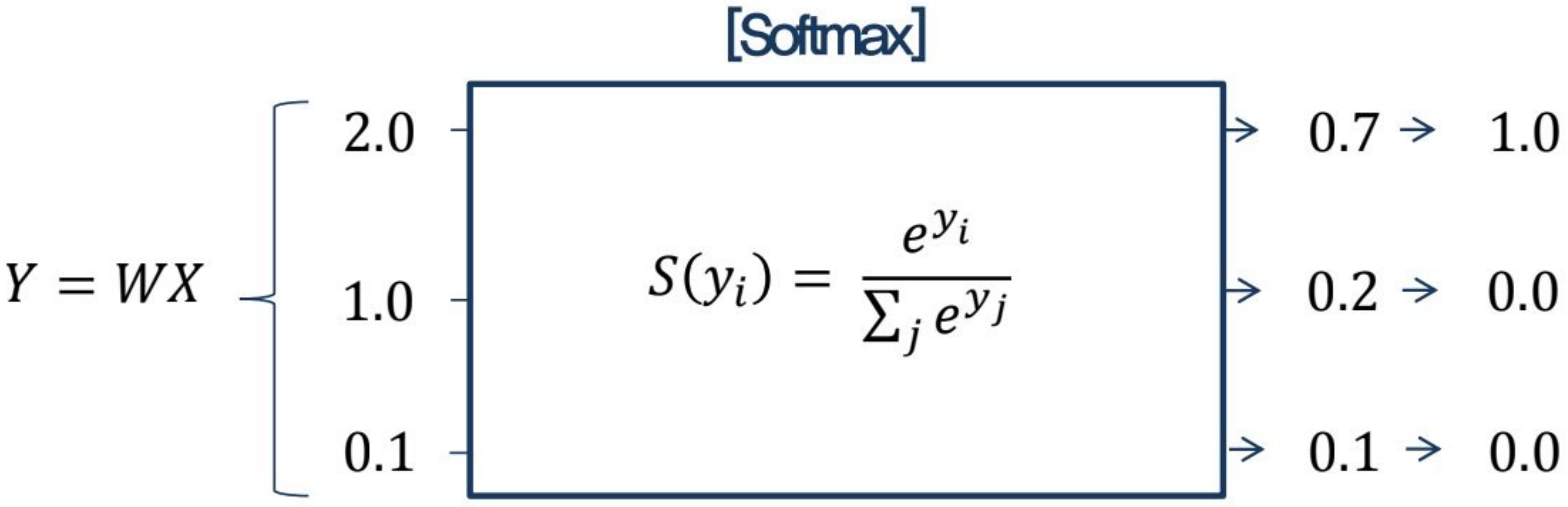
04
Softmax

일반적으로 선형 분류 모형의 경우, 일대다 방식으로 다중 분류문제를 해결한다.



04
Softmax

하지만, Logistic Regression의 경우 아래와 같은 Softmax라는 방식으로 해결할 수 있다.



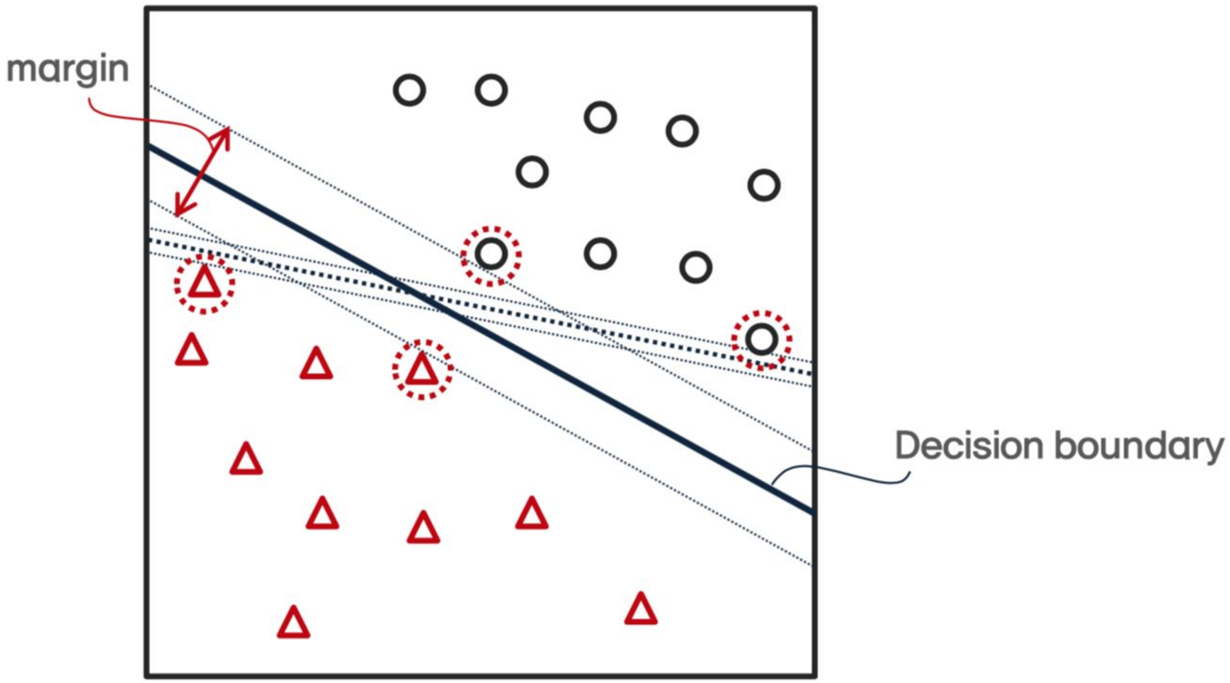
05

Linear SVM

Margin, Support Vector

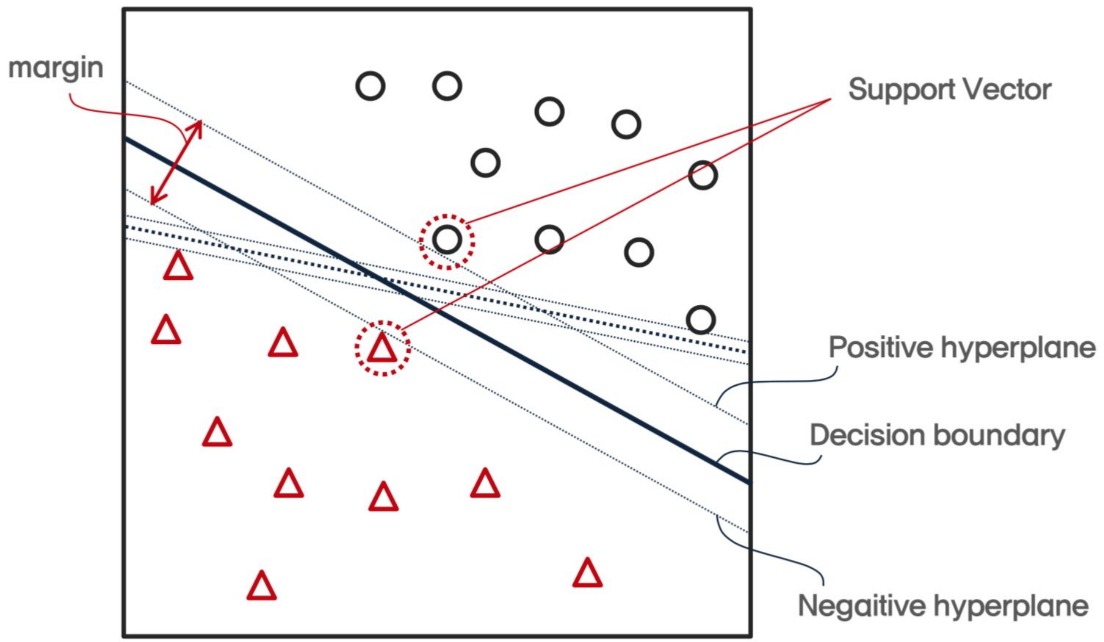
01
Margin

Margin이 클수록 Decision boundary는 안정적이다.



02
Support Vector

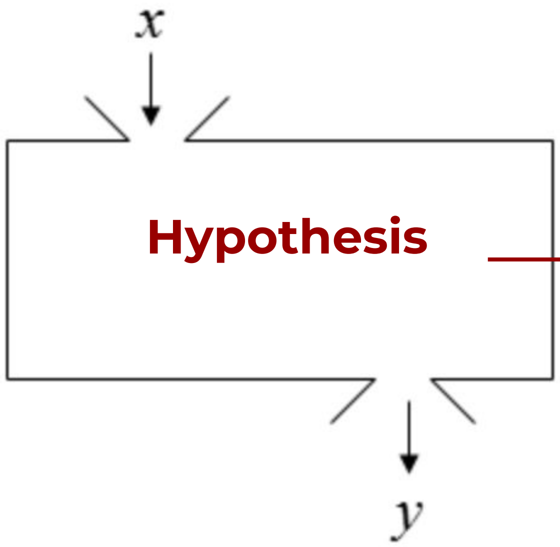
Margin을 결정하는 기준이 되는 trainset의 X 값을 서포트벡터라고 한다.



03

Hypothesis

SVM도 선형 분류 모형이다. 즉, 2개의 카테고리 데이터를 분류하는 선형 hyperplane을 찾는 것이 목적이다.



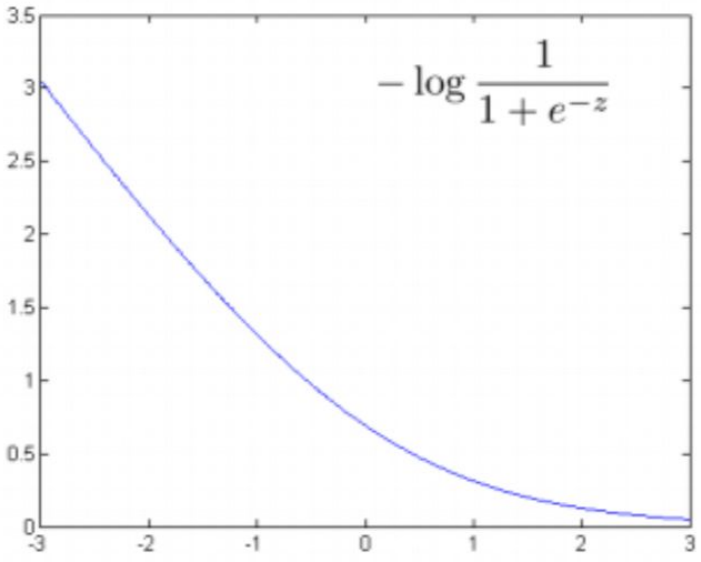
$$w[0] + w[1]*x[1] + w[2]*x[2] + w[3]*x[3] + \ldots + w[n]*x[n]$$

04

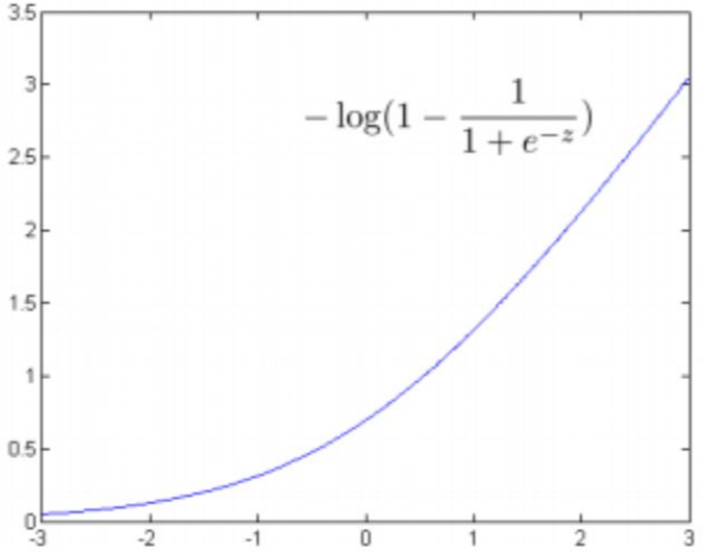
Cost

먼저, logistic regression의 Cost를 다시 확인해보자.

$$if\ y = 1, -\log(h_\theta(x))$$



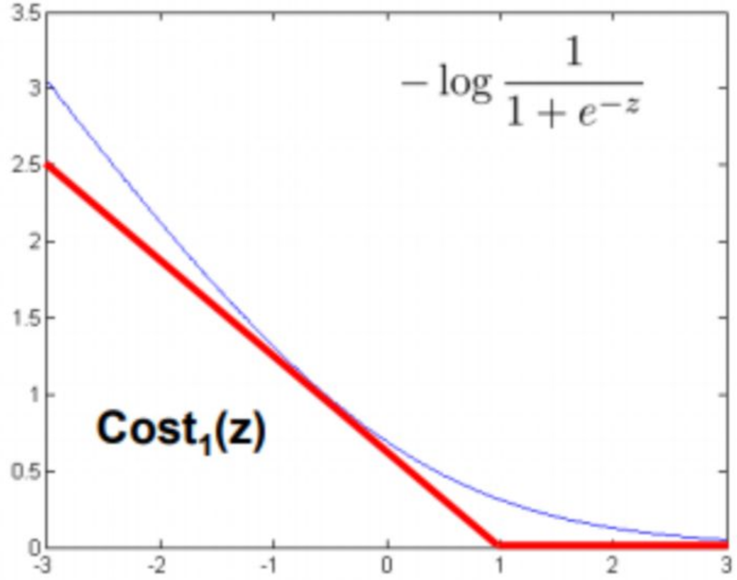
$$if\ y = 0, -\log\left(1 - h_\theta(x)\right)$$



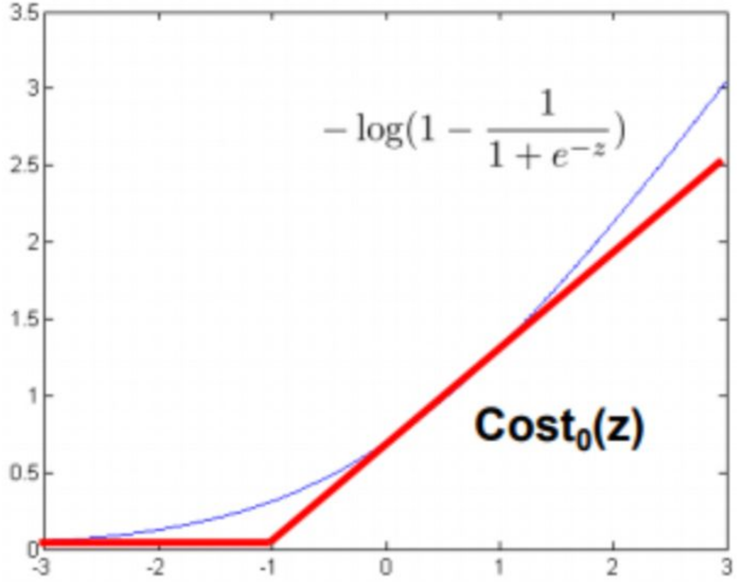
04
Cost

SVM은 마진을 기준으로 Cost를 계산한다.

$if\ y = 1$



$if\ y = 0$



06

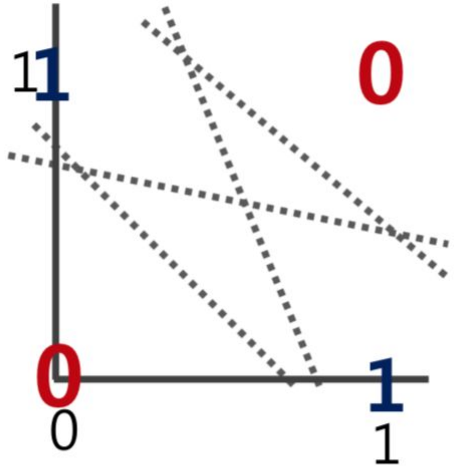
Kernel SVM

Basis function, Kernel

01
비선형 문제 - XOR

비선형 분류문제를 SVM으로 해결할 수 있을까?

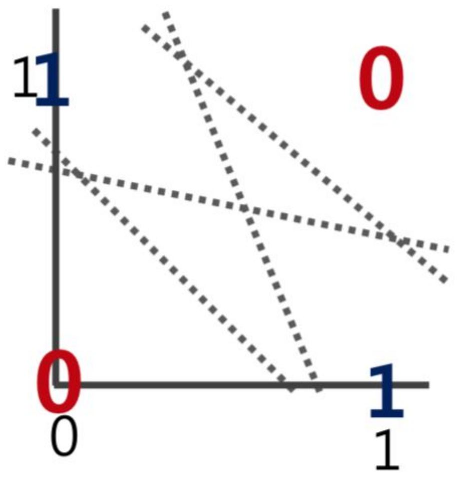
	X2 = 1	X2 = 0
X1 = 1	0	1
X1 = 0	1	0



01
비선형 문제 - XOR

비선형 문제에는 비선형 Decision boundary가 필요하다.

	X2 = 1	X2 = 0
X1 = 1	0	1
X1 = 0	1	0



02

비선형 Decision Boundary

데이터셋의 X값(독립변수)를 변환한다.

$$x = (x_1, x_2, \ \dots, x_D) \longrightarrow \emptyset(x) = (\emptyset_1(x), \emptyset_2(x), \ \dots, \emptyset_M(x))$$

이렇게 변환한 새로운 독립변수를 기저 함수(Basis function)이라고 부른다.

단, 모든 기저함수는 다음과 같이 변환한 벡터를 내적한 값으로 사용된다. 중요한 것은 기저함수 혼자 사용되지 않는다는 것이다.

$$k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

이를 커널(Kernel)이라고 부른다.

03

Kernel

대응하는 기저함수가 존재할 경우, 기저함수를 일일이 정하지 않고 직접 커널을 정의하여 사용할 수 있다. (즉, 편하다.)

그럼 주로 사용되는 Kernel을 살펴보자.

1) 커널 X

$$k(x_1, x_2) = x_1^T x_2$$

2) 다항 커널

$$k(x_1, x_2) = (\gamma(x_1^T x_2) + \theta)^d$$

3) 가우시안 커널

$$k(x_1, x_2) = \exp\left(-\gamma||x_1 - x_2||^2\right)$$

4) 시그모이드 커널

$$k(x_1, x_2) = \tanh(\gamma(x_1^T x_2) + \theta)$$