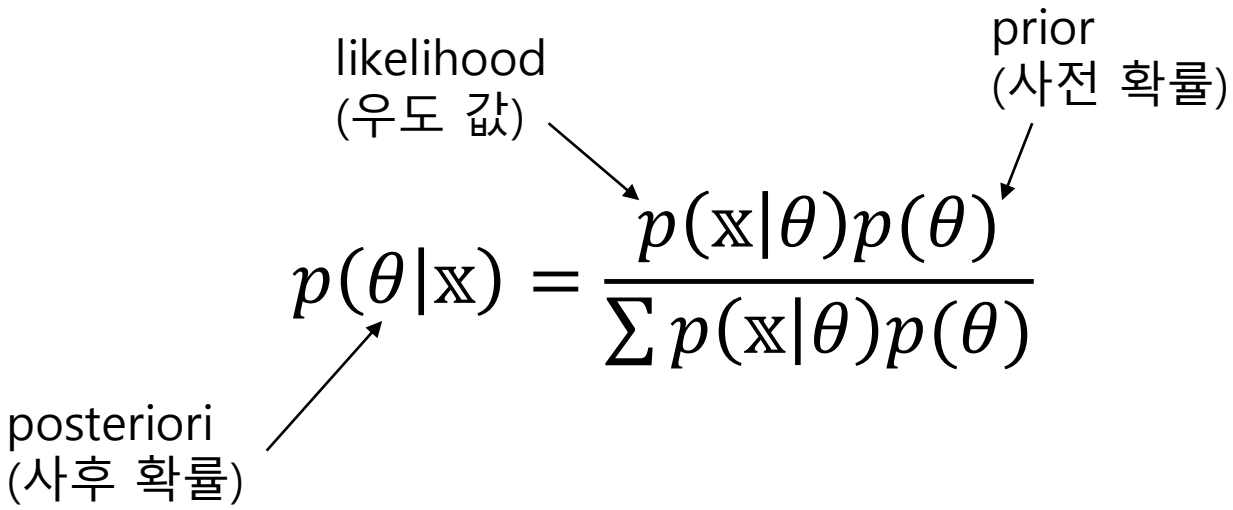
Logistic Regression

Bayes' Rule



사후 확률 : 관찰 값들이 관찰 된 후에 모수(parameter)의 발생 확률을 구한다.

사전 확률 : 관찰 값들이 관찰 되기 전에 모수의 발생 확률을 구한다.

우도 값 : 모수의 값이 주어졌을 때 관찰 값들이 발생할 확률

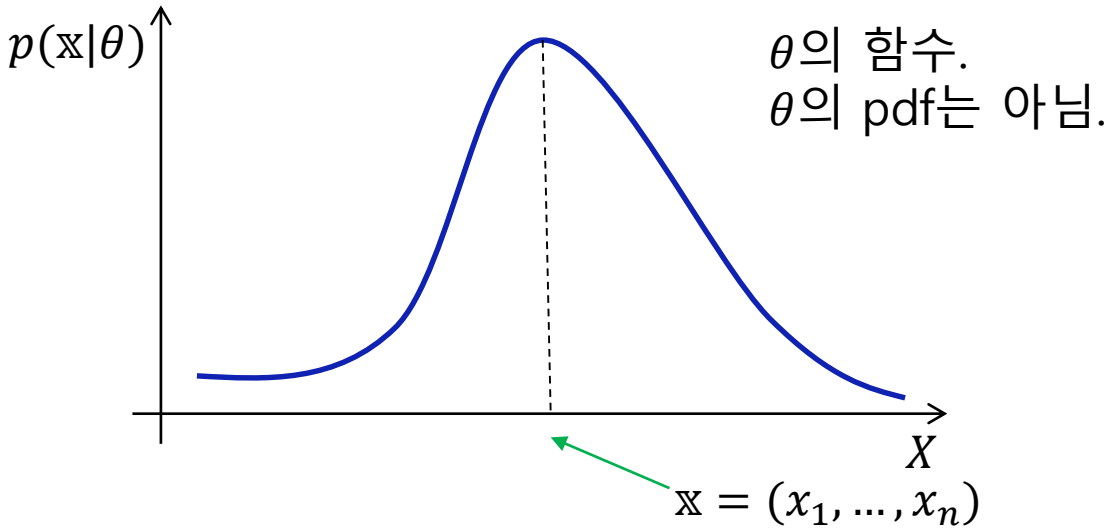
Maximum Likelihood Estimate

$\mathbb{x} = (x_1, \dots, x_n)$

우도(likelihood)는 다음과 같이 정의 된다.

$$\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{p}(\mathbb{x}|\boldsymbol{\theta})$$

변수(parameter) θ가 주어졌을 때, data set $\mathbb{x} = (x_1, \dots, x_n)$ (관찰 된, observed) 를 얻을 수 있는(obtaining) 확률

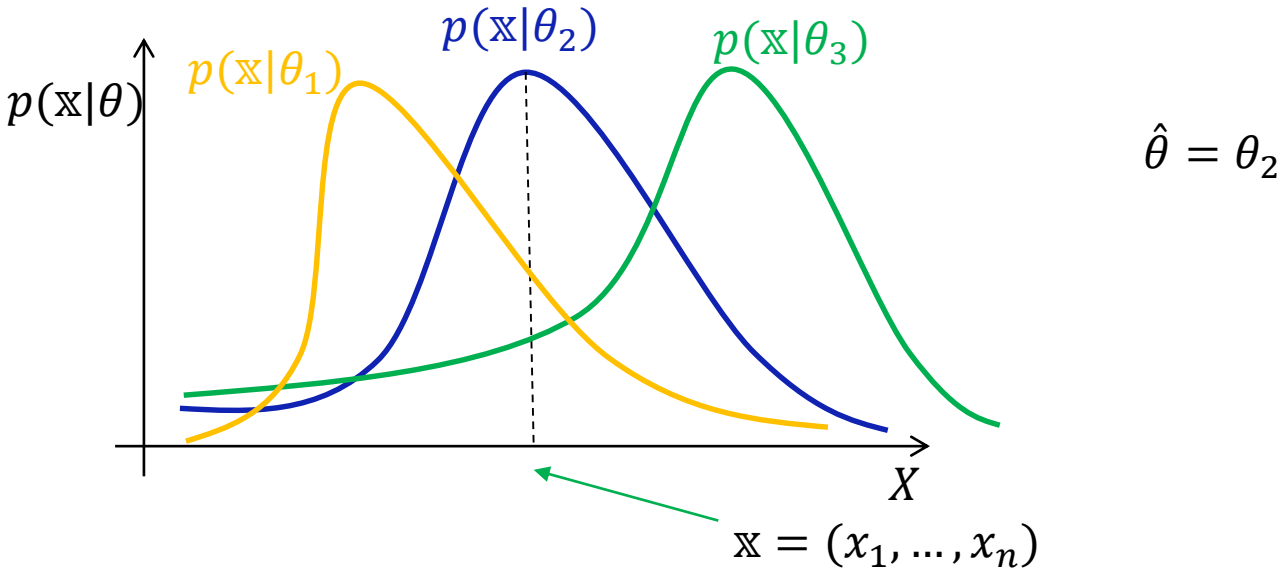


Logistic Regression

Maximum Likelihood Estimate는 다음과 같이 정의 된다.

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} p(\mathbb{x}|\boldsymbol{\theta})$$

관찰 된 data set $\mathbb{x} = (x_1, \dots, x_n)$을 얻을 수 있는 확률이 가장 큰 θ가 MLE이다.



$\hat{\theta} = \theta_2$

Maximum A Posteriori Estimate

우리가 likelihood function $p(\mathbb{x}|\theta)$와 prior $p(\theta)$를 알 때, Bayes rule에 의하여 posteriori function의 값을 구할 수 있다.

likelihood
(우도 값)

prior
(사전 확률)

$$p(\theta|\mathbb{x}) = \frac{p(\mathbb{x}|\theta)p(\theta)}{\sum p(\mathbb{x}|\theta)p(\theta)}$$

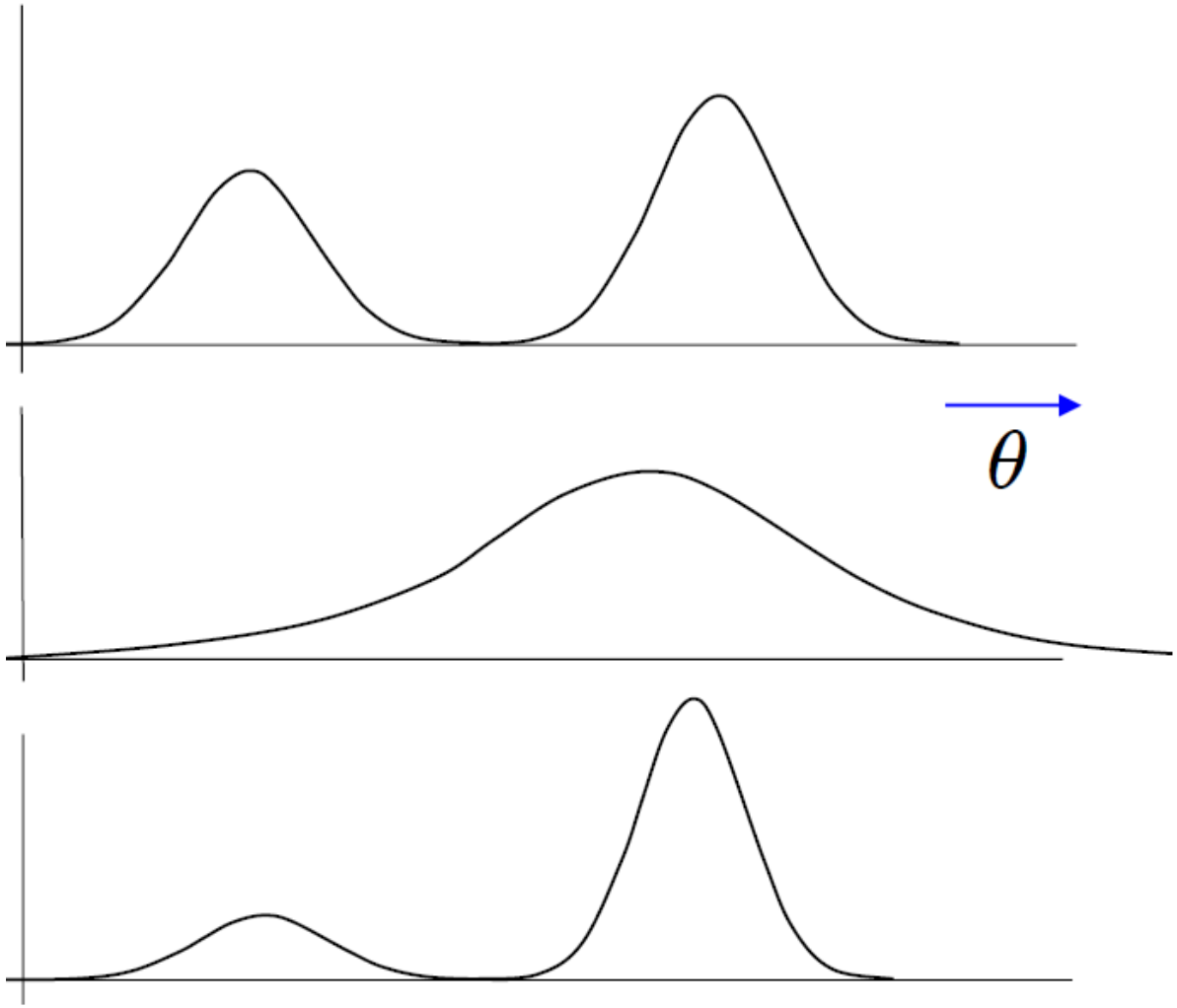
posteriori
(사후 확률)

$$\boldsymbol{p(\theta|\mathbb{x}) \propto p(\mathbb{x}|\theta)p(\theta)}$$

Likelihood $p(\mathbb{x}|\theta)$

Prior $p(\theta)$

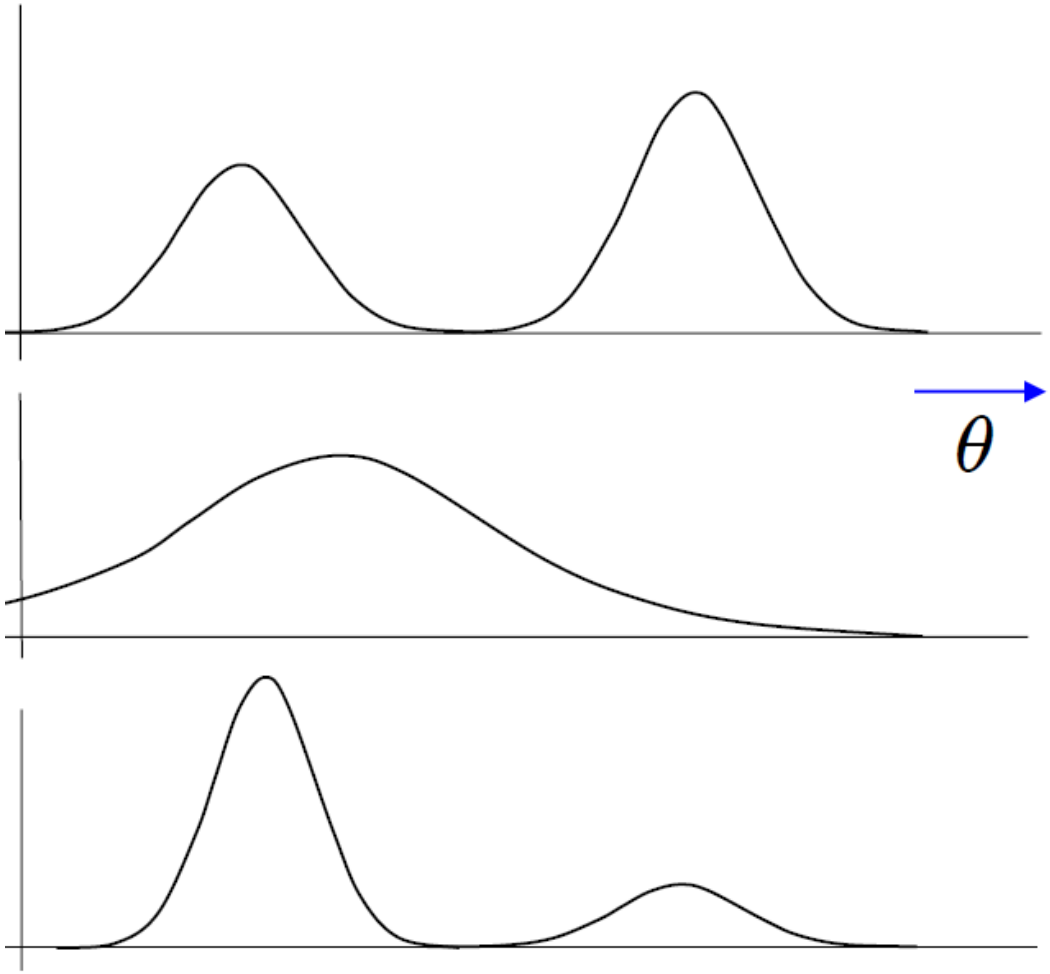
Posterior
$p(\theta|\mathbb{x}) \propto p(\mathbb{x}|\theta)p(\theta)$



Likelihood $p(\mathbb{x}|\theta)$

Prior $p(\theta)$

Posterior
$p(\theta|\mathbb{x}) \propto p(\mathbb{x}|\theta)p(\theta)$

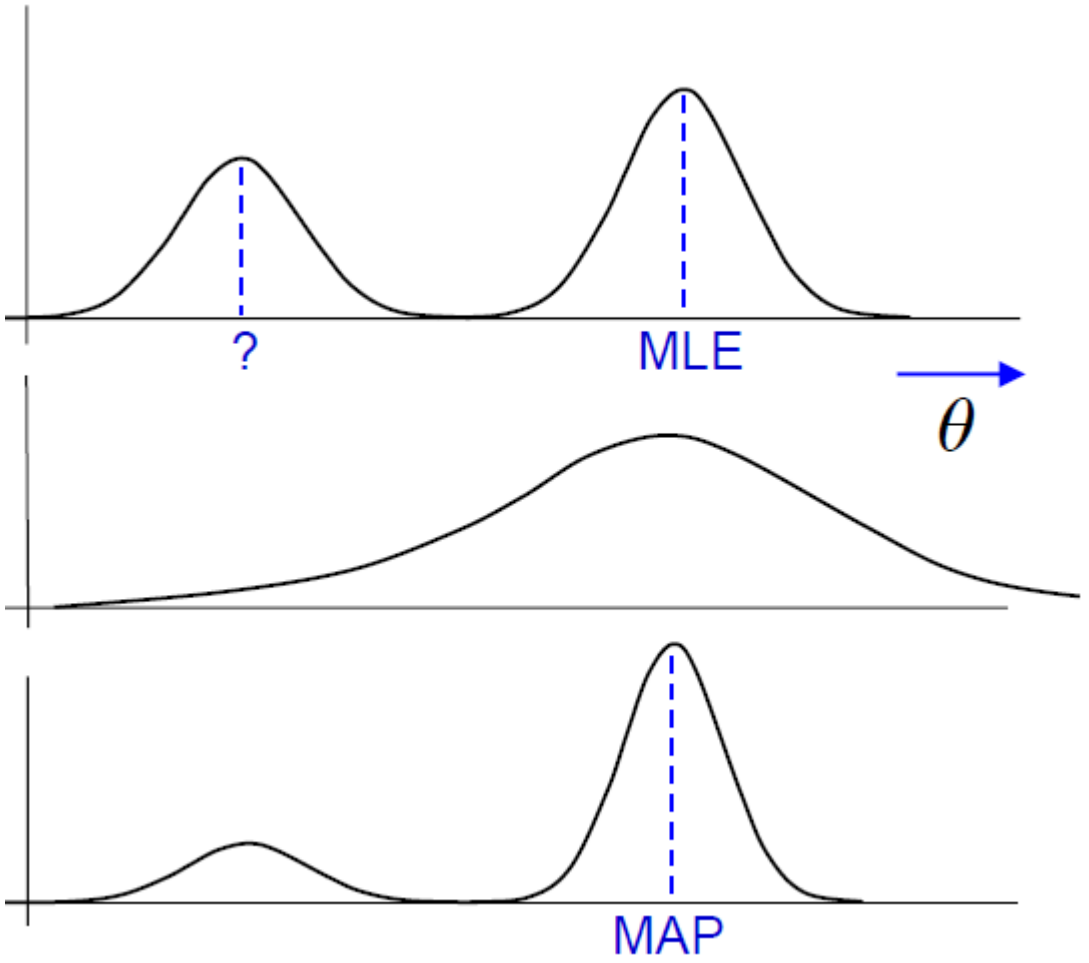


$$\boldsymbol{\theta = \arg\max_{\theta} p(\theta|\mathbb{x})}$$

Likelihood $p(\mathbb{x}|\theta)$

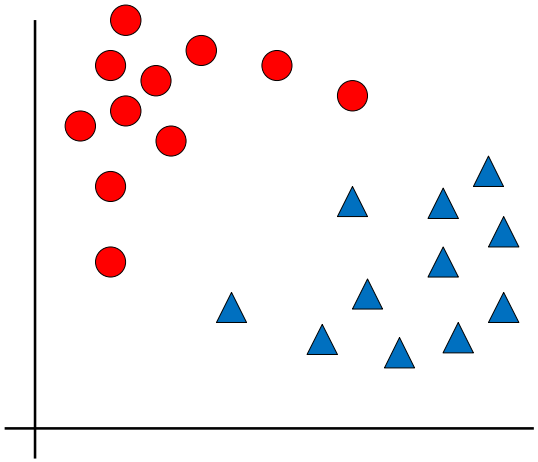
Prior $p(\theta)$

Posterior
$p(\theta|\mathbb{x}) \propto p(\mathbb{x}|\theta)p(\theta)$



Logistic Regression

Question : 2개의 cluster로 나누고 싶다. How?



Generative Classifier

생성하는

Training set을 이용하여, 군집(cluster) 전체의 모델을 찾기 원한다.

우리는 군집(cluster)의 확률 모델을 알고 있기 때문에,

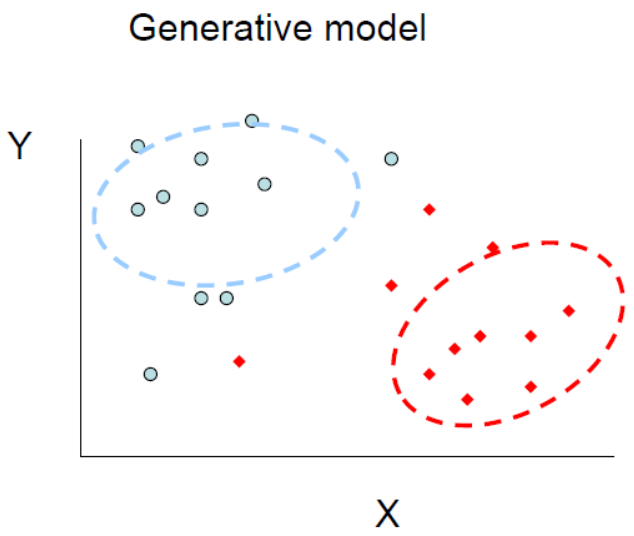
$P(X|Y), P(Y)$

새로운 데이터가 들어올 경우, 군집의 확률 모델을 다시 만들 수 있다.

$P(Y|X)$ Generative

Training set으로 부터 $P(X|Y)$, $P(Y)$를 바로 추정한다.

대표적인 예는 Naive Bayes classifier 이다.



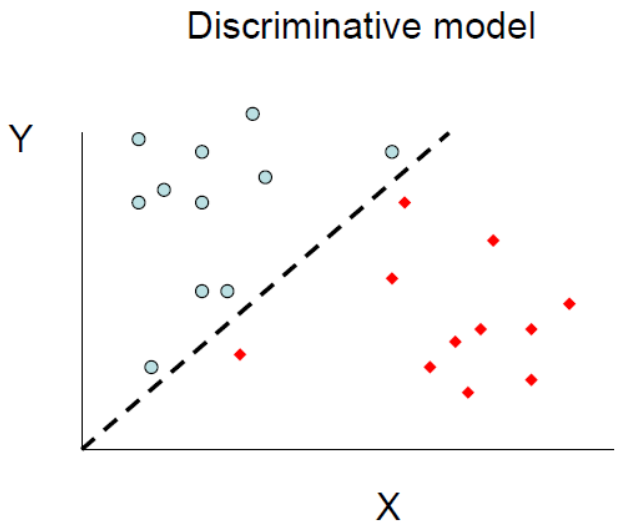
Discriminative Classifier

Training set을 이용하여, 군집을 나누어 줄 수 있는 경계선을 찾길 원한다.

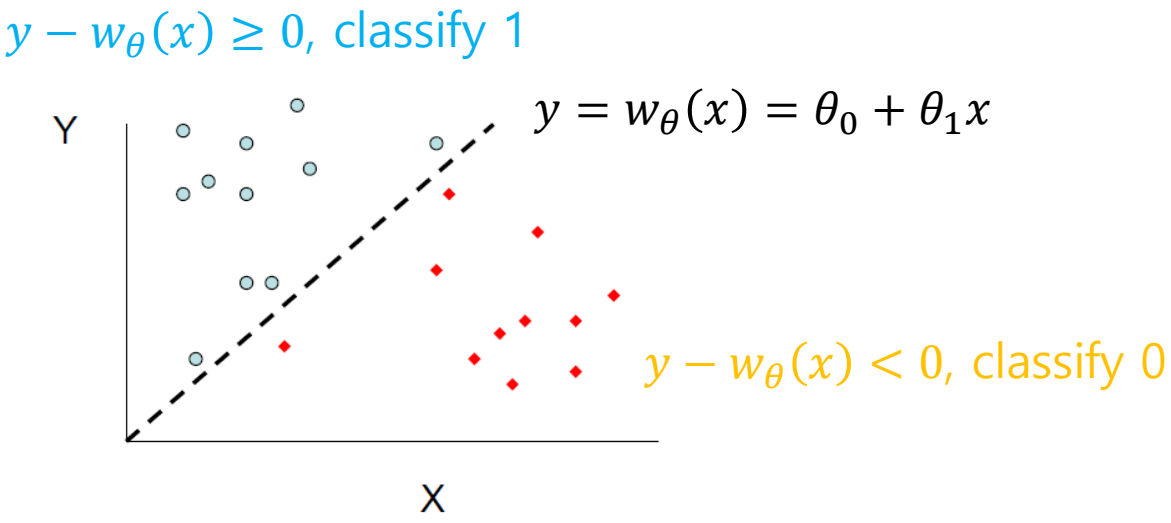
Clusters를 구별할 수 있는 boundary(경계)에 있는 data가 중요하다.

Training set으로 부터 $P(Y|X)$를 바로 추정한다.

가장 대표적인 예는 SVM, Logistic Classifier이다.



그렇다면, 우리는 선형 회귀 분석을 classification에 사용할 수 있을까?

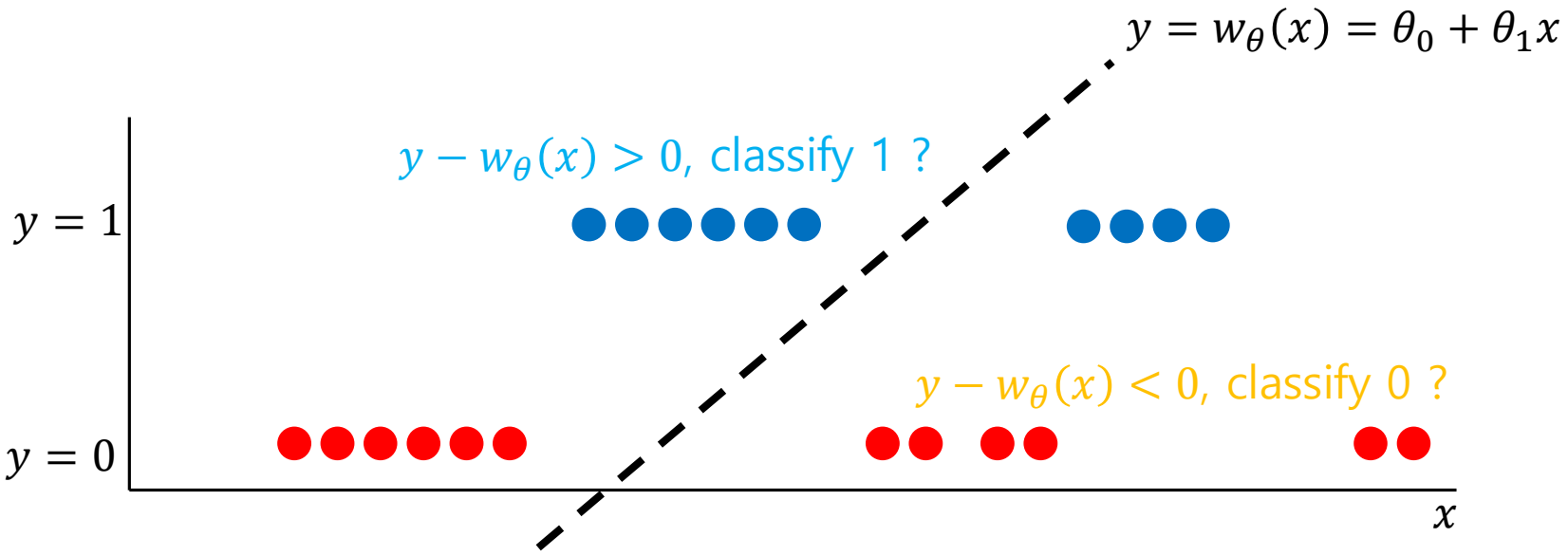


위의 점 선을, training set을 통해 도출된 회귀 선이라고 할 때,
분류(classification)가 가능해 보인다.

몇 몇의 경우(in some cases), linear regression을 적당한 경계 (appropriate boundary)를 결정하는 것에 사용할 수 있다. 즉, 분류(classification)이 가능하다.

하지만, 이 경우는?

y가 연속 값을 갖는 것이 아니고 불연속의 값, 즉 cluster의 index를 가질 때

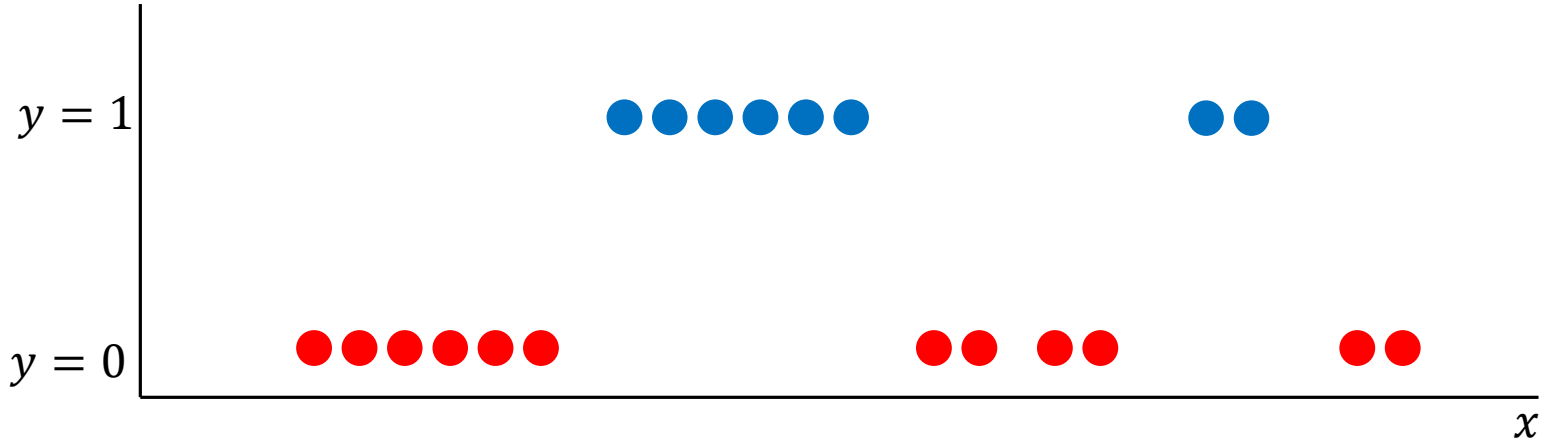


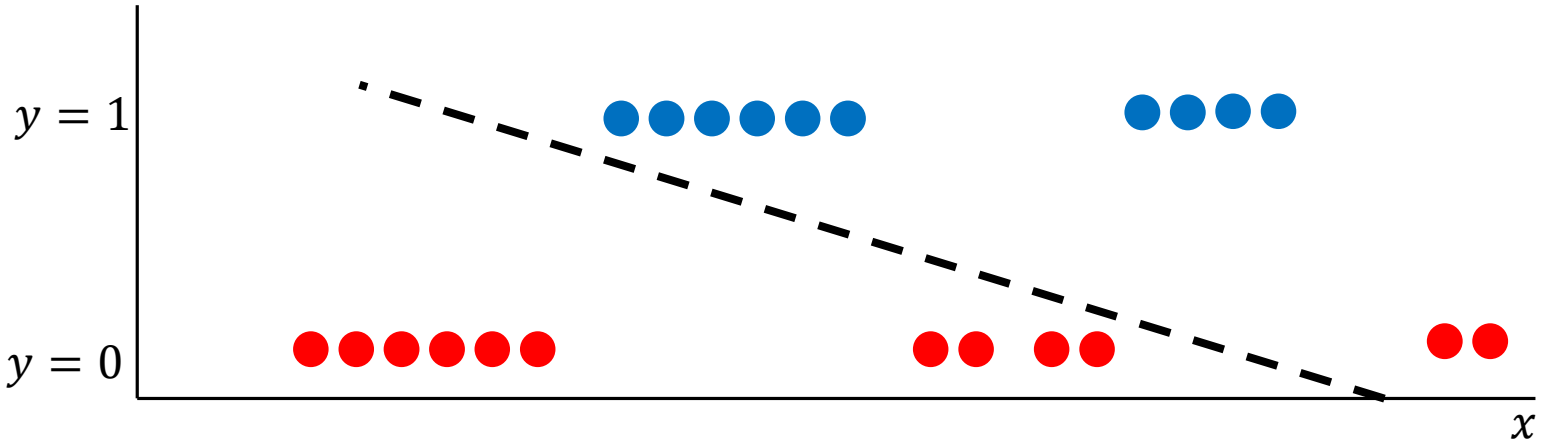
- 이런 경우는, 분류(classification)가 어렵다.
- $y \in \{0,1\}$ 인데 $w_\theta(x)$가 1보다 크거나,
  0보다 작은 수를 가지게 될 수 있다.

→ 결정 경계로 사용 할 수도 없고, data set을 fitting 하지도 않는다.

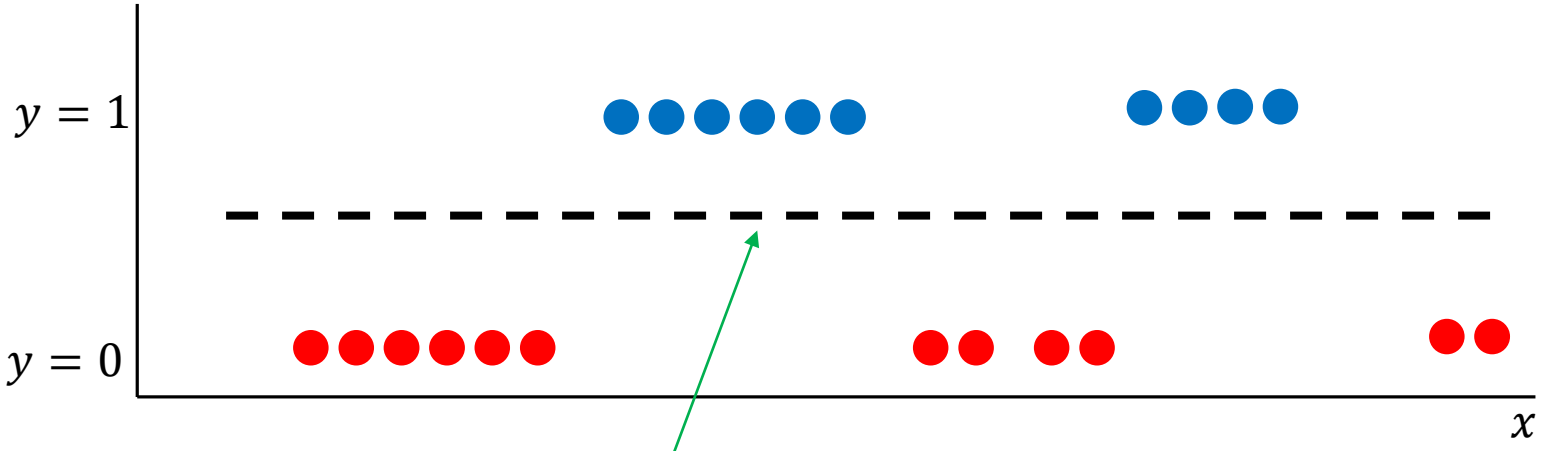
Motivation)

그렇다면, 이와 같은 data set 의 분류(classification)에 사용 될 수 있는 Regression은 없을까?



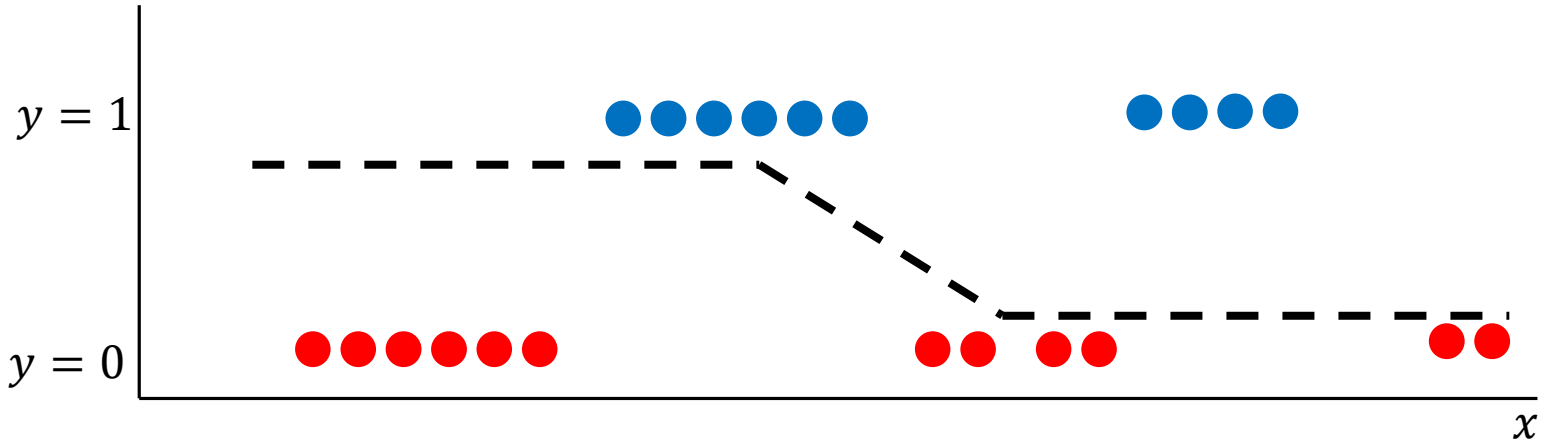


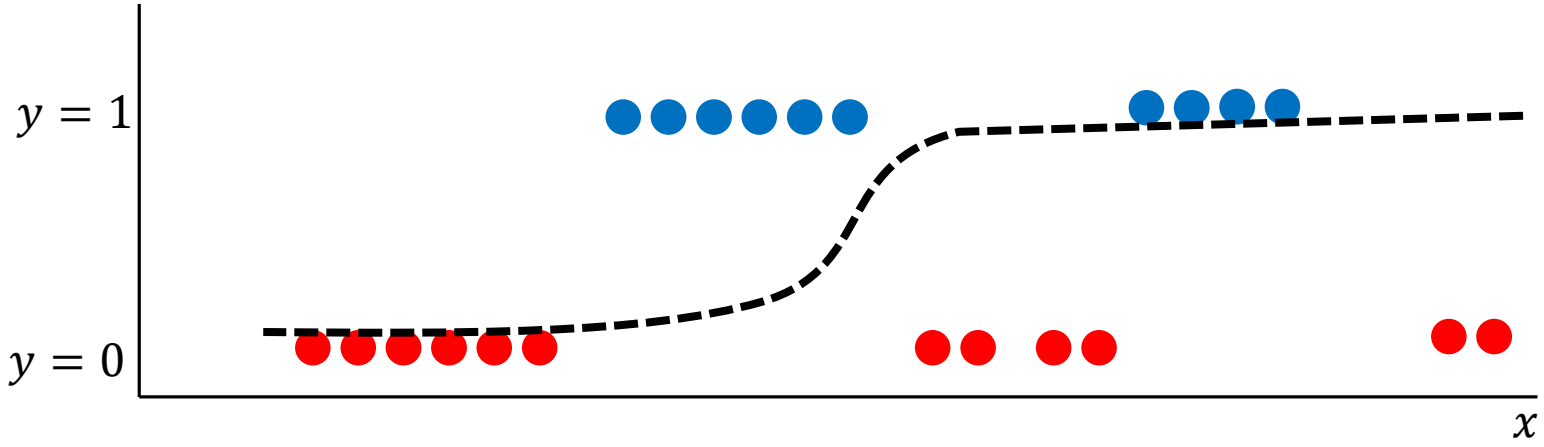
?



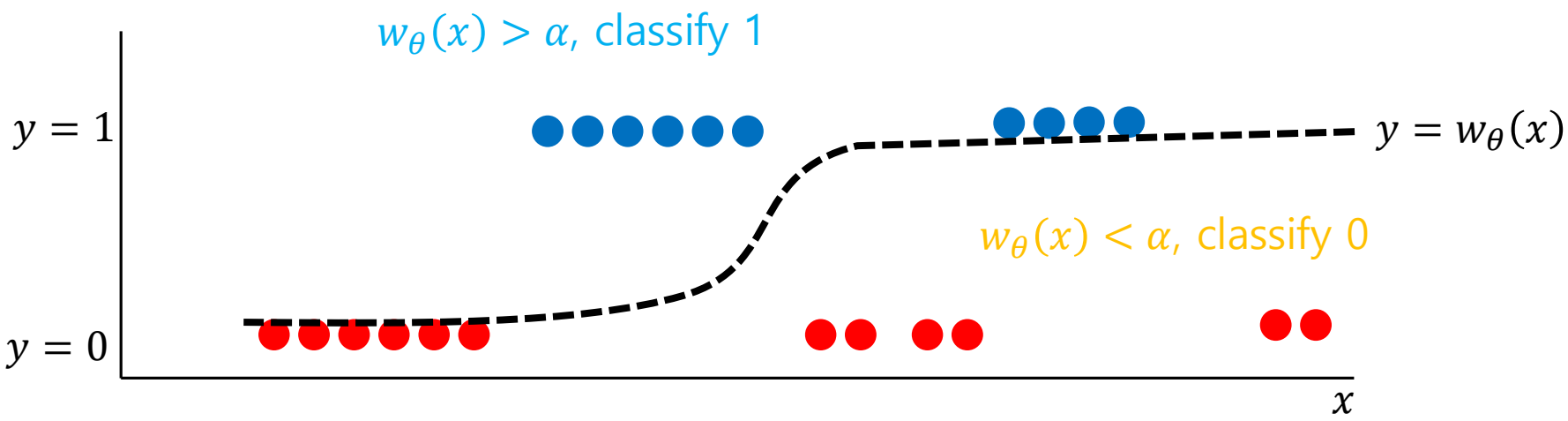
?

$\boldsymbol{y = \theta}$ 라는 식이 나온다. x의 값에 상관없이 y의 값이 고정되어 있으므로, 새로운 input x에 대한 y의 예측 값을 구할 수 없다.





이러한 함수가 있으면 얼마나 좋을까? 위와 같이 fitting 되는 회귀 식은 없나?



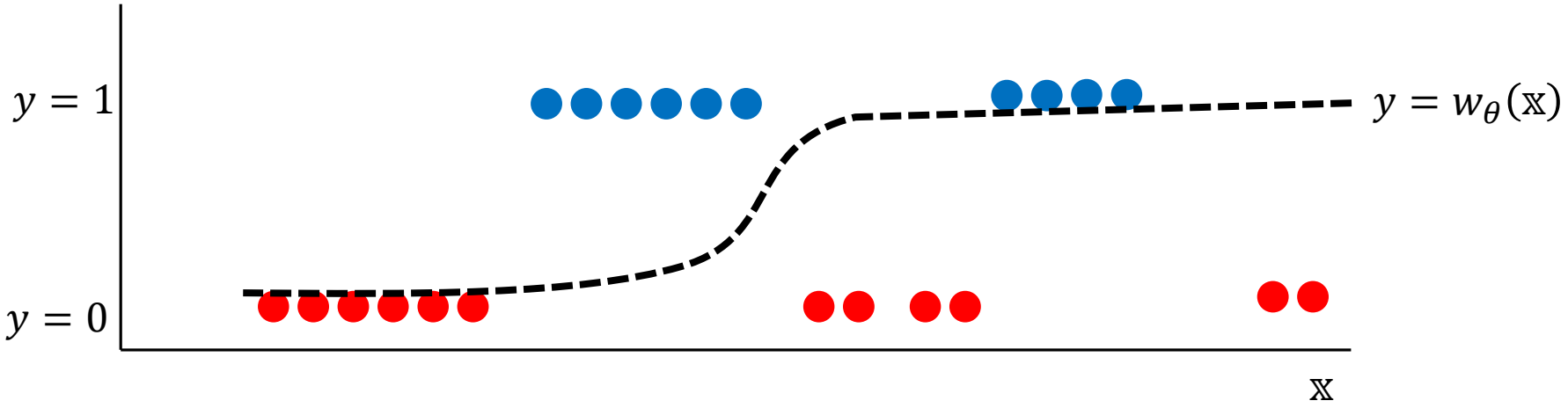
이렇게만 된 다면, discriminative classifier로 regression을 사용할 수 있다.

우리에게 필요한 함수는,
x의 값의 음의 무한대의 방향으로 가게 되면 함수는 1의 값에 가까이 가게 되고
x의 값의 음의 무한대의 방향으로 가게 되면 함수는 0의 값에 가까이 가게 된다.

즉, **Sigmoid 함수**가 대표적이다.
(나중에 자세히 살펴보자.)

Logistic Regression

Motivation



주의 사항 1)

선형 회귀 분석 처럼 경계선을 구할 수 있다. 여기서 사용 될 회귀 분석은 test set을 fitting하기 위함이다.

주의 사항 2)

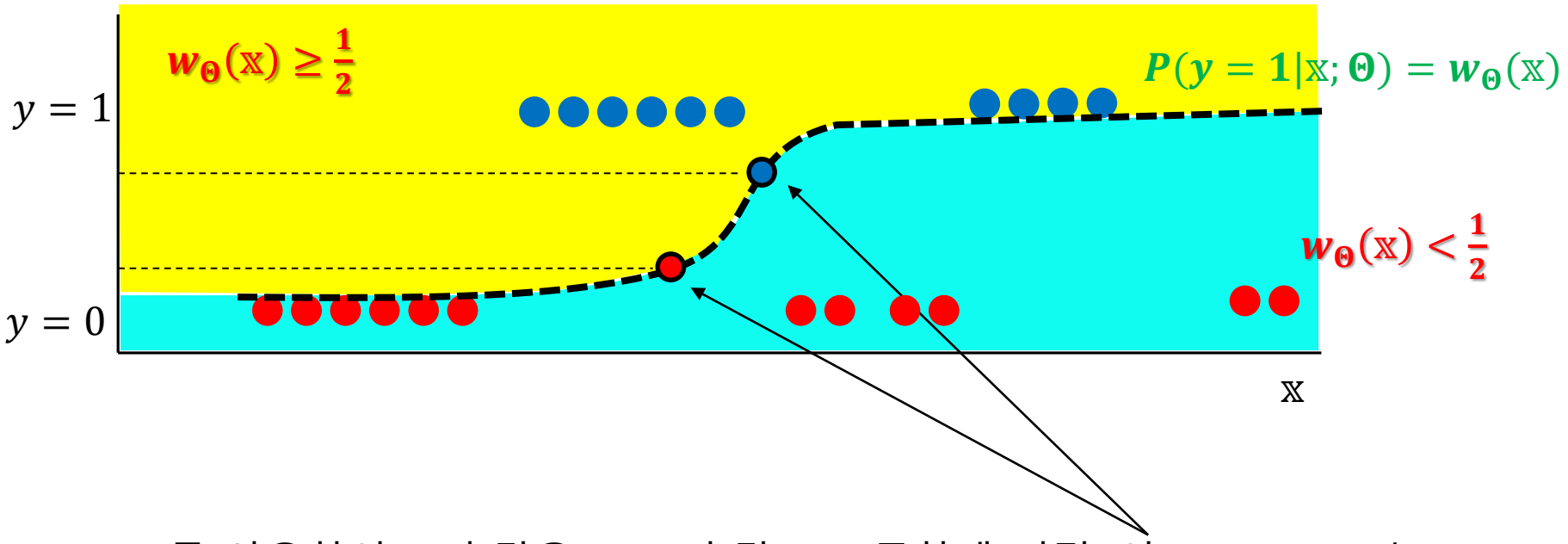
선형 회귀 분석에서는 $w_\theta(\mathbb{x})$ 의 값을 y의 값으로 사용한다. 하지만, 우리는 $w_\theta(\mathbb{x})$ 의 값을 y의 값으로 바로 사용하지 않을 것이다. 2가지의 assumption,

$$P(y = 1|\mathbb{x}; \theta) = w_\theta(\mathbb{x})$$

$$P(y = 0|\mathbb{x}; \theta) = 1 - w_\theta(\mathbb{x})$$

을 사용하여 $w_\theta(\mathbb{x})$ 의 값을 $y = 1$ 일 때의 조건부 확률로 바꿀 것이다.

$p(y|\mathbb{x}; \Theta)$: conditional probability of y given $\mathbb{x}$ under parameter $\Theta = (\theta_1, \dots, \theta_n)$



$y = w_\Theta(\mathbb{x})$를 사용하여, y의 값을 class의 값으로 구하게 되면, 이 data points는 0과 1사이 값이 나오게 되어, class를 구하기가 불가능 해진다. (class의 값은 0, 1이다.)

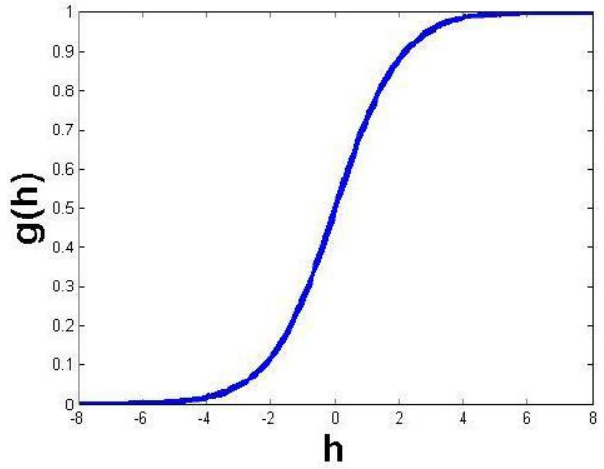
그래서 $P(y=1|\mathbb{x};\Theta) = w_\Theta(\mathbb{x})$를 사용한다. 이 data points의 $w_\Theta(\mathbb{x})$의 값을 가지고, data point가 class 1에 속하는지 class 0에 속하는지 판단하면 된다.

$$P(y=1|\mathbb{x};\Theta) \geq P(y=0|\mathbb{x};\Theta) : \text{class 1로 분류} \rightarrow w_\Theta(\mathbb{x}) \geq \frac{1}{2}$$

$$P(y=1|\mathbb{x};\Theta) < P(y=0|\mathbb{x};\Theta) : \text{class 0로 분류} \rightarrow w_\Theta(\mathbb{x}) < \frac{1}{2}$$

Sigmoid Function

$$g(h) = \frac{1}{1 + e^{-t}} \qquad 0 \le g(h) \le 1$$

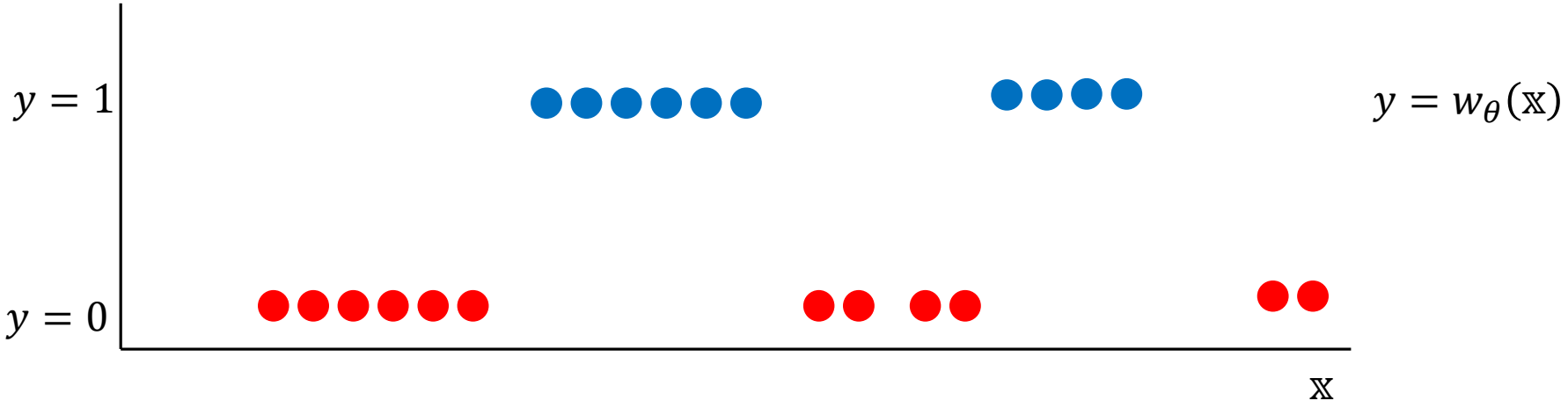


$$P(y = 1|\mathbb{x}; \Theta) = w_\Theta(\mathbb{x}) = g(\Theta^T \mathbb{x}) = \frac{1}{1 + e^{-\Theta^T \mathbb{x}}}$$

$$P(y = 0|\mathbb{x}; \Theta) = 1 - w_\Theta(\mathbb{x}) = 1 - g(\Theta^T \mathbb{x}) = \frac{e^{-\Theta^T \mathbb{x}}}{1 + e^{-\Theta^T \mathbb{x}}}$$

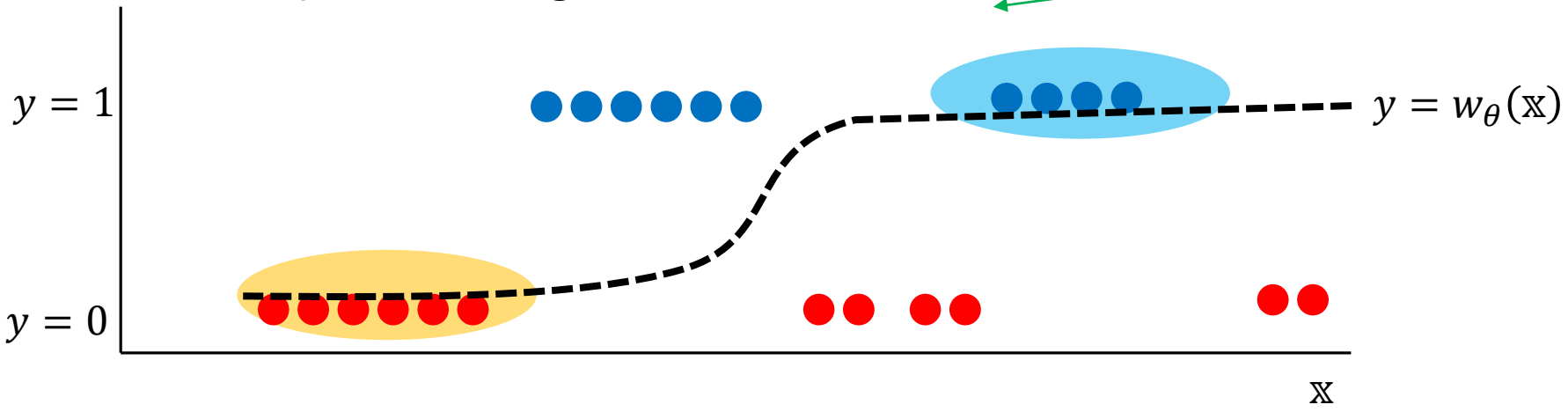
Algorithm

① Training data points

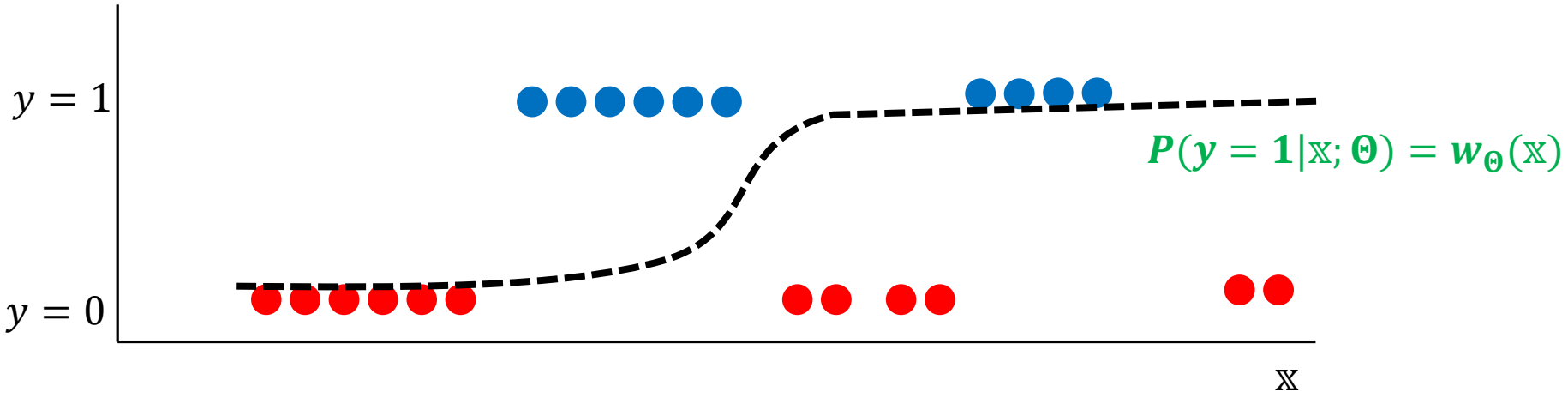


② Fit the data points into sigmoid function $y = w_\Theta(\mathbb{X})$.

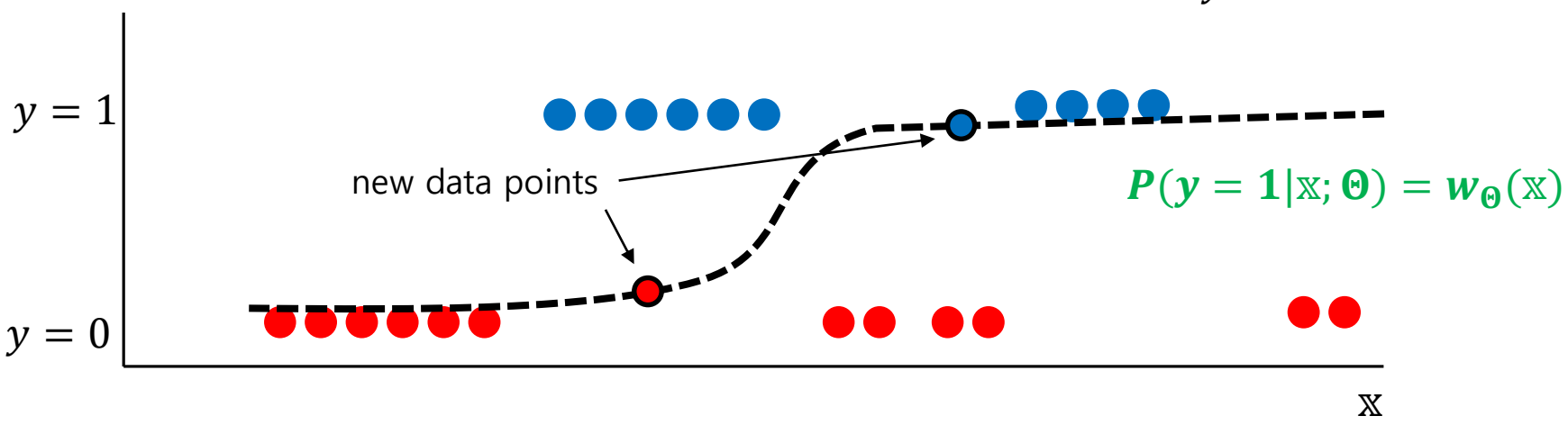
data points를 가장 잘 표현 할 수 있는 모수 $\theta_0, \theta_1, \ldots, \theta_n$를 구한다.



③ If we get the $y = w_\Theta(\mathbb{x})$, then, let $w_\Theta(\mathbb{x}) = P(y = 1|\mathbb{x}; \Theta)$



④ A new data point $\mathbb{x}_n$, if $w_\Theta(\mathbb{x}_n) \geq 0.5$, $\mathbb{x}_n$ is class 1. Others $\mathbb{x}_n$ is class 0.



$$P(y = 1|\mathbb{x}; \Theta) = w_\Theta(\mathbb{x}) = g(\Theta^T\mathbb{x}) = \frac{1}{1 + e^{\Theta^T\mathbb{x}}}$$

Logistic Regression

Question !

그러면, 회귀 식

$$P(y = 1|\mathbb{x}; \Theta) = w_{\Theta}(\mathbb{x}) = g(\Theta^T \mathbb{x}) = \frac{1}{1 + e^{-\Theta^T \mathbb{x}}}$$

어떻게 구할까?

즉, 회귀 식의 모수(parameter)를 어떻게 구할까?

Maximum Likelihood Estimator (M.L.E.)

Maximum Likelihood Estimator (M.L.E.)

Recall!

① $P(y=1|\mathbb{x};\Theta) = w_\Theta(\mathbb{x})$
② $P(y=0|\mathbb{x};\Theta) = 1 - w_\Theta(\mathbb{x})$
③ $w_\Theta(\mathbb{x}) = \frac{1}{1+e^{-\Theta^T\mathbb{x}}}$

$\Rightarrow$ $p(y|\mathbb{x};\Theta) = \left(w_\Theta(\mathbb{x})\right)^y \left(1 - w_\Theta(\mathbb{x})\right)^{1-y}$

$\mathbb{x} = (x_1, \dots, x_n)^T \in R^n, \quad \Theta = (\theta_0, \theta_1, \dots, \theta_n),$

training data points $X = (\mathbb{x}_1, \dots, \mathbb{x}_m)$ 과 각 data points에 대응하는 label $Y = (y_1, \dots, y_m)$이 주어졌을 때, **likelihood**를 구하는 공식은 아래와 같다. 단 $y_i \in \{0,1\}$

$$\boldsymbol{L(\Theta)} = p(Y|X;\Theta) = p(y_1, \dots, y_m \mid \mathbb{x}_1, \dots, \mathbb{x}_m\,;\, \theta_0, \dots, \theta_m)$$
$$= \prod_{i=1}^{m} p(y_i|\mathbb{x}_i;\Theta)$$
$$= \boldsymbol{\prod_{i=1}^{m} \left(w_\Theta(\mathbb{x}_i)\right)^{y_i} \left(1 - w_\Theta(\mathbb{x}_i)\right)^{1-y_i}}$$

우리는, $L(\Theta)$를 최대값이 나오도록 하는 모수 Θ를 찾는 것이 목표이다.

즉, **Maximum Likelihood Estimate**는

$$\hat{\Theta} = \arg\max_{\Theta} L(\Theta)$$

$$= \arg\max_{\Theta} \prod_{i=1}^{m} \left(w_{\Theta}(\mathbb{X}_i)\right)^{y_i} \left(1 - w_{\Theta}(\mathbb{X}_i)\right)^{1-y_i}$$

유도 된 MLE식을 풀기 위해서 log를 사용한다.

$$P(y=1|\mathbb{x};\Theta) = w_\Theta(\mathbb{x}) = \frac{1}{1+e^{-\Theta^T\mathbb{x}}} = \frac{e^{\Theta^T\mathbb{x}}}{1+e^{\Theta^T\mathbb{x}}}$$

$$P(y=0|\mathbb{x};\Theta) = 1 - w_\Theta(\mathbb{x}) = \frac{e^{-\Theta^T\mathbb{x}}}{1+e^{-\Theta^T\mathbb{x}}} = \frac{1}{1+e^{\Theta^T\mathbb{x}}}$$

$$l(\Theta) = logL(\Theta)$$

$$= \sum_{i=1}^{m} \{y_i \log w_\Theta(\mathbb{x}_i) + (1-y_i)\log(1-w_\Theta(\mathbb{x}_i))\}$$

$$= \sum_{i=1}^{m} \left\{ y_i \log \frac{w_\Theta(\mathbb{x}_i)}{(1-w_\Theta(\mathbb{x}_i))} + \log(1-w_\Theta(\mathbb{x}_i)) \right\}$$

$$= \sum_{i=1}^{m} \left\{ y_i \Theta^T \mathbb{x}_i - \log(1+e^{\Theta^T\mathbb{x}_i}) \right\}$$

극대 값을 찾아야 할 목적 함수

MLE의 극대 값을 구하기 위하여 gradient ascent를 사용할 것이다.

$l(\Theta)$를 θ_j에 관하여 미분한 식을 구하자.

$$\mathbb{x}_i = \left(x_1{}^{(i)}, \dots, x_n{}^{(i)}\right)^T \in R^n, \quad \Theta = (\theta_0, \theta_1, \dots, \theta_n),$$

$$\frac{\partial}{\partial \theta_j} l(\Theta) = \sum_{i=1}^{m} \{y_i x_j{}^{(i)} - \frac{e^{\Theta^T \mathbb{x}_i}}{\left(1 + e^{\Theta^T \mathbb{x}_i}\right)} x_j{}^{(i)}\}$$

$$= \sum_{i=1}^{m} \{y_i x_j{}^{(i)} - P(y_i = 1 | \mathbb{x}_i; \Theta) x_j{}^{(i)}\}$$

$$= \sum_{i=1}^{m} x_j{}^{(i)} \{y_i - P(y_i = 1 | \mathbb{x}_i; \Theta)\}$$

Gradient ascent 공식에 의하여,

$$\theta_j = \theta_j + \alpha \frac{\partial}{\partial \theta_j} l(\Theta)$$

where,

$$\frac{\partial}{\partial \theta_j} l(\Theta) = \sum_{i=1}^{m} \{y_i x_j{}^{(i)} - \frac{e^{\Theta^T \mathbb{x}_i}}{\left(1 + e^{\Theta^T \mathbb{x}_i}\right)} x_j{}^{(i)}\}$$

$$= \sum_{i=1}^{m} x_j{}^{(i)} \{y_i - \hat{P}(y_i = 1 | \mathbb{x}_i; \Theta)\}$$

prediction error : 관찰 된 y_i와, y_i의 예측 된 확률의 차이

concave 함수의 성질에 의하여 $l(\Theta)$는 극대 값을 갖는다.

Gradient Ascent Algorithm

1. α를 선택한다.

2. $\Theta = (\theta_0, \theta_1, \ldots, \theta_n)$의 적당한 초기 값을 설정한다.

3. 모든 j에 대하여, $\theta_j \leftarrow \theta_j + \alpha \frac{\partial}{\partial \theta_j} l(\Theta) = \theta_j + \alpha \sum_{i=1}^{m} x_j{}^{(i)} \{y_i - P(y_i = 1 | \mathbb{x}_i; \Theta)\}$

4. if, 모든 j에 대하여 $\sum_{i=1}^{m} x_j{}^{(i)} \{y_i - P(y_i = 1 | \mathbb{x}_i; \Theta)\}$의 값의 변화가 없으면 멈춘다.
   otherwise, 3번으로 간다.

우리는 과적합(overfitting)을 방지하기 위하여 MLE 대신에 MAP를 사용한다. 1

MLE $$\hat{\Theta} = \arg\max_{\Theta} L(\Theta) = \arg\max_{\Theta} \prod_{i=1}^{m} p(y_i|\mathbb{x}_i;\Theta)$$

MAP $$\hat{\Theta} = \arg\max_{\Theta} L(\Theta)p(\Theta) = \arg\max_{\Theta} \prod_{i=1}^{m} p(y_i|\mathbb{x}_i;\Theta)\,p(\Theta)$$

MLE(log) $$\hat{\Theta} = \arg\max_{\Theta} \sum_{i} \log p(y_i|\mathbb{x}_i;\Theta)$$

MAP(log) $$\hat{\Theta} = \arg\max_{\Theta} \sum_{i} \log p(y_i|\mathbb{x}_i;\Theta) + \log p(\Theta)$$

$p(\Theta)$는 여러 가지 분포가 사용될 수 있으나, $\theta_i \sim N(0, \sigma^2)$를 사용하자.

그러면 MAP estimate는,

$$f = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

$$l_{MAP}(\Theta) = logL_{MAP}(\Theta)$$

$$= \sum_{i=1}^{m} \left\{ y_i \Theta^T \mathbb{x}_i - \log(1 + e^{\Theta^T \mathbb{x}_i}) \right\} - \sum_{j=1}^{m} \frac{{\theta_j}^2}{2\sigma^2}$$

극대 값을 찾아야 할 목적 함수

Gradient ascent는

$$\theta_j \leftarrow \theta_j + \alpha \frac{\partial}{\partial \theta_j} l_{MAP}(\Theta) = \theta_j + \alpha \sum_{i=1}^{m} {x_j}^{(i)} \{ y_i - P(y_i = 1 | \mathbb{x}_i; \Theta) \} - \alpha \frac{\theta_j}{\sigma^2}$$

Logistic Regression을 regularize하는 방법은 이 외에도 다양하다.

class가 2보다 클 경우, 즉 Y가 $\{y_1, \dots, y_n\}$의 값을 가질 경우의 logistic regression은

$$P(Y = y_k | \mathbb{X}; \Theta) = \frac{\exp(\theta_{k0} + \sum_{i=1}^{n} \theta_{ki} X_i)}{1 + \sum_{j=1}^{K-1} \exp(\theta_{j0} + \sum_{i=1}^{n} \theta_{ji} X_i)}$$

$$P(Y = y_k | \mathbb{X}; \Theta) = \frac{1}{1 + \sum_{j=1}^{K-1} \exp(\theta_{j0} + \sum_{i=1}^{n} \theta_{ji} X_i)}$$

Gradient ascent는

$$\theta_{ji} \leftarrow \theta_{ji} + \alpha \sum_{i=1}^{m} x_j{}^{(i)} \{\delta(y_i = j) - P(y_i = j | \mathbb{X}_i; \Theta)\}$$

$\delta(y_i = j) : y_i = j$이면 1, 그렇지 않으면 0

Regularization

Logistic regression function을 추정하는데 있어서, 우리는 충분한 양의 data를 가지고 있지 못 할 수 있다.

충분한 양의 training data가 없다면, 좋은 회귀 함수를 추정하기 어렵다.

data points의 차원(dimensionality)가 높은데 training data가 희소(sparse)하면 'Overfitting(과적합)'의 문제가 발생한다. 훈련 데이터 집합은 전체 모집단이 가지고 있는 패턴들을 가지고 있을 수 있다. 또, 일부 누락할 수도 있다. 하지만 문제는 전체 모집단은 가지고 있지 않고, 훈련 데이터 집합만 가지고 있는 특징까지도 알고리즘이 학습(learning)을 한다는 것이다. 즉, 불필요한 것까지 학습해 버린다.

이것을 극복하기 위하여, 우리가 추정하는(fitting) 모수(parameter)에 추가적인 제약(constraint)를 준다. 이것을 '**Regularization**'이라고 한다.

'Regularization'의 방법은 다양하다. 그 중의 하나는 penalized log likelihood function을 이용하는 것이다. 이것은 Θ의 큰(large)값에 제약을 주는 것이다.

제약을 주는 방법은 우리가 사용한 **MLE 대신에 MAP를 사용**하는 것이다.

[MLE와 MAP의 관계]

likelihood (우도 값)

prior (사전 확률)

$$p(\theta|\mathbb{X}) = \frac{p(\mathbb{X}|\theta)p(\theta)}{\sum p(\mathbb{X}|\theta)p(\theta)}$$

posteriori (사후 확률)

$$\boldsymbol{p(\theta|\mathbb{X}) \propto p(\mathbb{X}|\theta)p(\theta)}$$

in logistic

$$\boldsymbol{p(y, \Theta|\mathbb{X}) \propto p(y|\mathbb{X}; \Theta)p(\Theta)}$$

logistic regression에서는 우도 값과 사후 확률의 관계가 위와 같이 된다.

Θ를 추정하기 위하여, 우리는 $p(y|\mathbb{X}; \Theta)$를 사용했으나, **regularization을 위해 $\boldsymbol{p(y, \Theta|\mathbb{X})}$를 사용하는 것**이다. 즉 **$\boldsymbol{p(y|\mathbb{X}; \Theta)p(\Theta)}$를 사용**하는 것이다.