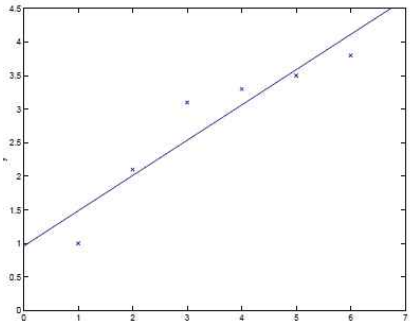
Machine Learning Lecture Note

This lecture note is based on the CS229 lecture materials from Standford University. **Lecture 3. Locally weighted regression, Probabilistic interpretation, Logistic regression, Newton's method**

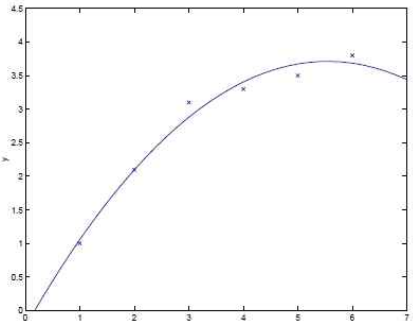
Parametic learning algorithm: 데이터에 맞는 학습 파라미터의 수가 고정되어 있고 유한하다. 즉, 학습을 통해 학습 파라미터가 결정되면, 미래의 상태 예측을 위해 더 이상 훈련 데이터를 유지할 필요가 없다.

Non-parametic learning algorithm: 미래의 상태 예측을 위해 전체 훈련 데이터를 유지해야 한다. 가설 함수 (목적 함수; object/cost function)를 표현하기 위해 유지해야 하는 정보의 양이 학습 셋의 크기에 따라 선형적으로 증가한다.

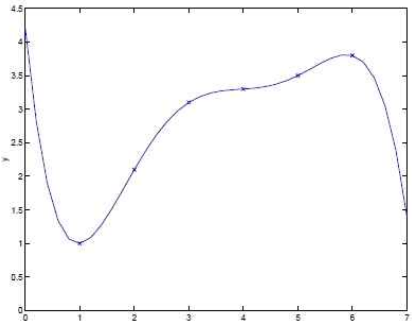
Locally weighted linear regression은 non-parametic learning algorithm에 해당한다.



$y = \theta_0 + \theta_1 x$



$y = \theta_0 + \theta_1 x + \theta_2 x^2$



$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5$

데이터 $x \in R$로 부터 y를 예측하는 문제에서, 데이터의 학습에 위 그림처럼 데이터의 feature가 많이 추가될수록 학습 그래프는 데이터의 분포와 더욱 일치하는 결과를 얻는다. 그러나 이것은 좋은 학습 알고리즘이 아니다. 왼쪽 그래프는 데이터의 특징을 제대로 추출하지 못한 underfitting에 해당하고, 오른쪽 그래프는 주어진 학습 데이터 셋에 과하게 일치시킨 overfitting에 해당한다.

좋은 성능을 내기 위한 학습 알고리즘에서는 학습 알고리즘에 적용할 feature의 범위를 선택하는 것이 중요하다.

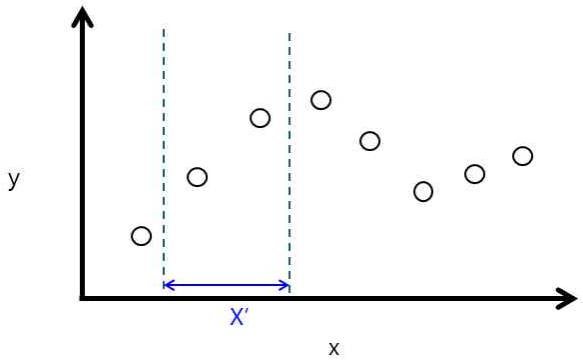
Locally weighted regression (LWR)

Linear regression

1. Fit θ to minimize $\sum_i (y^i - \theta^T x^i)^2$
2. Output $\theta^T x$

Locally weighted regression

3. Fit θ to minimize $\sum_i w^i (y^i - \theta^T x^i)^2$
4. Output $\theta^T x$



가중치 w는 포인트 x에 대하여 가우시안 분포의 밀도와 유사하다.

$$w^i = \exp\left(-\frac{(x^i - x)^2}{2\tau^2}\right), \quad (\tau : \text{bandwidth})$$

$|x^i - x|$가 작으면 w는 1에 가까워지고, $|x^i - x|$이 크면 w는 0에 가까워진다. w를 통해서 부분 영역 (local region)에 대한 학습을 수행하여 그 결과를 얻을 수 있다.

확률적 해석 (**Probabilistic interpretation**)

타겟 변수와 입력 데이터의 관계는 다음과 같이 나타낼 수 있다.

$$y^i = \theta^T x^i + \epsilon^i$$

가정) $\epsilon^i \sim N(0, \sigma^2)$ 이면, ϵ^i의 확률밀도는 다음과 같다.

$$p(\epsilon^i) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{(\epsilon^i)^2}{2\sigma^2}\right),$$

이 식은 다음의 의미를 내포한다.

$$p(y^i|x^i;\theta) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{(y^i - \theta^T x^i)^2}{2\sigma^2}\right), \text{ 즉 } y^i|x^i;\theta \sim N(\theta^T x^i, \sigma^2)$$

파라미터 θ의 관점으로 이 식을 변형하면, likelihood 함수로 나타낼 수 있다.

$$\begin{aligned} L(\theta) &= p(y|x;\theta) \\ &= \prod_{i=1}^{m} p(y^i|x^i;\theta) \\ &= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{(y^i - \theta^T x^i)^2}{2\sigma^2}\right) \end{aligned}$$

Likelihood 함수를 단순화하기 위해 log를 취하면:

$$\begin{aligned} l(\theta) &= \log L(\theta) \\ &= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{(y^i - \theta^T x^i)^2}{2\sigma^2}\right) \\ &= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{(y^i - \theta^T x^i)^2}{2\sigma^2}\right) \\ &= m \log \frac{1}{\sqrt{2\pi}\,\sigma} - \frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{m}(y^i - \theta^T x^i)^2 \end{aligned}$$

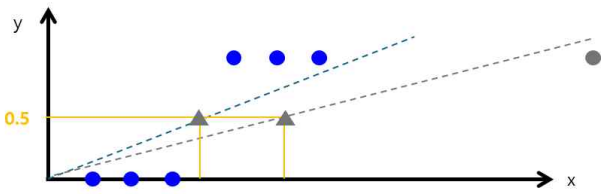
Maximum likelihood estimation: $L(\theta)$를 최대화 하기 위한 θ를 선택하는 것이다.

Log likelihood 함수를 최대화 하기 위해서는 다음 항을 최소화해야 한다:

$$\frac{1}{2}\sum_{i=1}^{m}(y^i - \theta^T x^i)^2, \quad \textbf{Origianl least-squares cost function}$$

Classification (분류)

$y \in \{0,1\}$, (binary classification: 이진 분류)



선형 회귀(Linear regression)은 분류 문제에서 좋은 알고리즘은 아니다. 그림에서처럼 하나의 샘플 데이터에 의해 분류 지점(decision boundary)이 달라지는 것으로 인해 예측값과 실제 데이터 사이의 차이가 커진다.

Logistic regression (로지스틱 회귀)

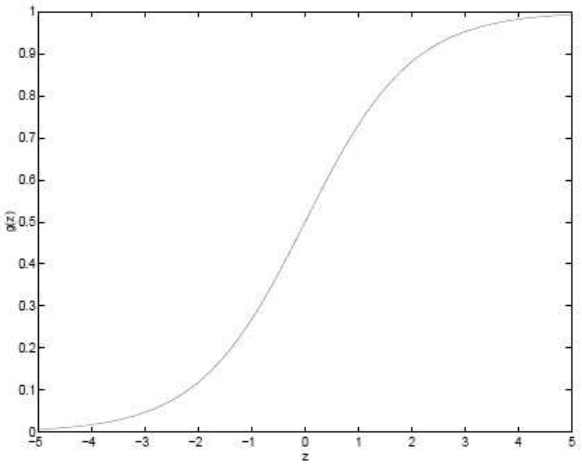
$h_\theta(x) \in \{0, 1\}$

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}} \quad \Rightarrow g(z) = \frac{1}{1+e^{-z}}$$

"sigmoid" or "logistic" function

$z \to \infty$ 이면, $g(z)$는 1에 가까워지고, $z \to -\infty$ 이면, $g(z)$는 0에 가까워진다.

$$\begin{aligned} g'(z) &= \frac{d}{dz}\frac{1}{1+e^{-z}} = \frac{1}{(1+e^{-z})^2}(e^{-z}) \\ &= \frac{1}{1+e^{-z}}\left(1-\frac{1}{1+e^{-z}}\right) \\ &= g(z)(1-g(z)) \end{aligned}$$



데이터의 분포를 다음과 같이 가정하면,

$$p(y=1|x;\theta) = h_\theta(x), \quad p(y=0|x;\theta) = 1-h_\theta(x)$$

위 식은 아래와 같이 다시 작성될 수 있다.

$$p(y|x;\theta) = h_\theta(x)^y(1-h_\theta(x))^{1-y}, \quad y \in \{0,1\}$$

$y=1$ 이면 $p(y|x;\theta) = h_\theta(x)$ 이고, $y=0$ 이면 $p(y|x;\theta) = 1-h_\theta(x)$

Logistic regression model은 Least squares regression처럼 maximum likelihood를 통해 유도될 수 있다.

$$\begin{aligned} L(\theta) &= p(y|X;\theta) \\ &= \prod_{i=1}^{m} p(y^i|x^i;\theta) \\ &= \prod_{i=1}^{m} h_\theta(x^i)^{y^i}\left(1-h_\theta(x^i)\right)^{1-y^i} \end{aligned}$$

연산을 단순화하기 위해 log likelihood로 수정하면,

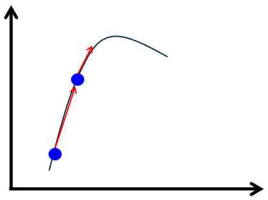
$$\begin{aligned} l(\theta) &= \log L(\theta) \\ &= \sum_{i=1}^{m}\left[y^i \log h_\theta(x^i) + (1-y^i)\log\left(1-h_\theta(x^i)\right)\right] \end{aligned}$$

$l(\theta)$를 최대화하기 위한 θ를 선택하기 위해 Batch gradient ascent 알고리즘을 사용한다.

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

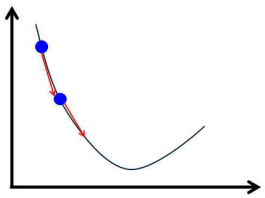
gradient ascent 알고리즘

$$\theta_j := \theta_j + \frac{\partial}{\partial \theta_j} l(\theta)$$



gradient descent 알고리즘

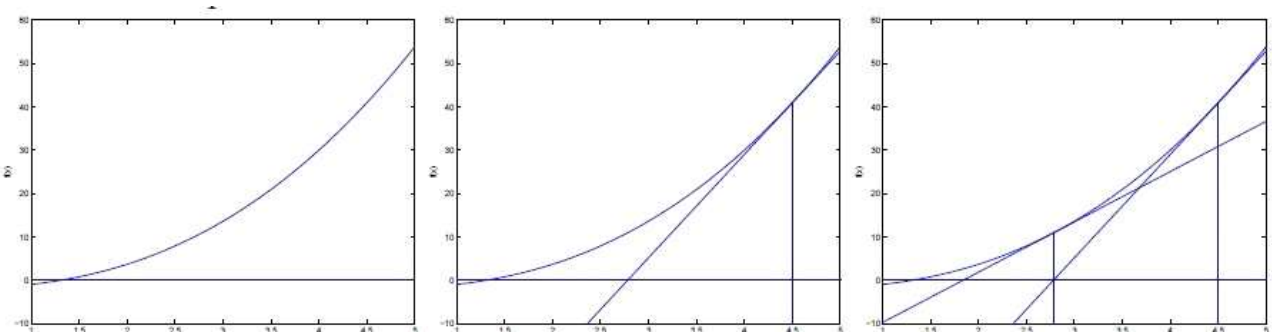
$$\theta_j := \theta_j - \frac{\partial}{\partial \theta_j} J(\theta)$$



$$\begin{aligned}
\frac{\partial}{\partial \theta_j} l(\theta) &= \left(y \frac{1}{g(\theta^T x)} - (1-y) \frac{1}{1-g(\theta^T x)} \right) \frac{\partial}{\partial \theta_j} g(\theta^T x) \\
&= \left(y \frac{1}{g(\theta^T x)} - (1-y) \frac{1}{1-g(\theta^T x)} \right) g(\theta^T x)(1-g(\theta^T x)) \frac{\partial}{\partial \theta_j} \theta^T x \\
&= \left(y(1-g(\theta^T x)) - (1-y) g(\theta^T x) \right) x_j \\
&= (y - h_\theta(x)) x_j
\end{aligned}$$

$$\therefore \theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^i - h_\theta(x^i) \right) x_j^i$$

Newton's method ($l(\theta)$를 최대화하기 위한 다른 방법)



임의의 함수 $f : R \mapsto R$ 에 대하여 $\theta \in R$ 이면, 뉴턴 기법은 다음과 같다.

$$\theta := \theta - \frac{f(\theta)}{f'(\theta)}$$

θ에서 함수 f의 기울기는 $f'(\theta)$이고, 기울기가 0이 되는 지점 (즉, $f'(\theta) = 0$)은 $\frac{f(\theta)}{f'(\theta)}$ 이다.

만약, $f(\theta) = l'(\theta)$로 둔다면 $l(\theta)$를 최대화하기 위해 $f(\theta) = l'(\theta) = 0$이 되도록 하기 위한 θ는 다음과 같다:

$$\theta := \theta - \frac{l'(\theta)}{l''(\theta)}$$

위 식을 다차원(multidimensional)으로 일반화 하면 Newton-Raphson 기법이 된다.

$$\theta := \theta - H^{-1} \nabla_\theta l(\theta), \qquad H_{ij} = \frac{\partial^2 l(\theta)}{\partial \theta_i \partial \theta_j} \text{ (Hessian matrix)}$$

뉴턴 기법은 경사하강법(gradient descent) 보다 빠르게 수렴하지만, Hessian 연산으로 인해 경사하강법 보다 많은 연산 비용이 발생한다.