참고: https://bcho.tistory.com/1139, https://bcho.tistory.com/1142?category=555440

1. Linear Regression을 통한 머신 러닝의 개념 이해

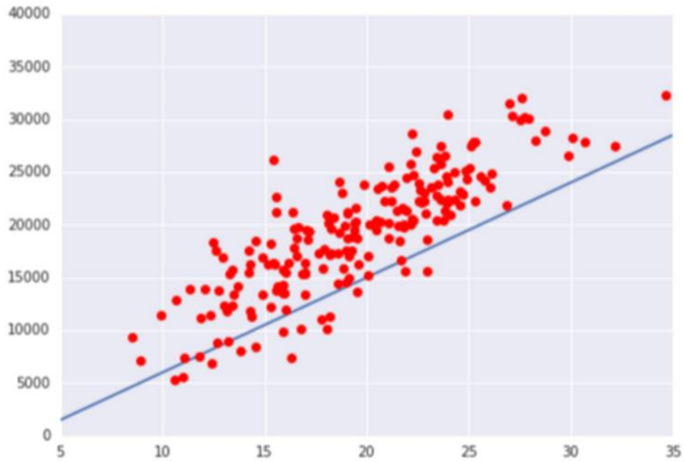
선형회귀란 결과값이 있고 그 결과값을 결정할 것이라고 추정되는 입력 값과 결과값의 연관관계를 찾는 것이고 이를 선형 관계를 통해 찾는 방법이 선형회귀이다. 예를 들어 택시 요금은 교통체증의 정도에 따라 차이는 있겠지만 기본적으로 거리에 비례해서 요금이 부과되는데. 이런 경우 결과값 (요금)과 입력 값(거리)의 관계를 찾아야 한다.

2. 가설 정의

1) Linear Regression

거리와 요금이 서로 비례하기 때문에 거리(x)와 요금(y)간의 상관관계는 다음과 같이 일차 방정식과 형태 그래프를 그리게 된다고 가정하자. W(weight)는 그래프 각도, b는 bias를 뜻한다.

$$\mathrm{H(x)} = Wx + b$$

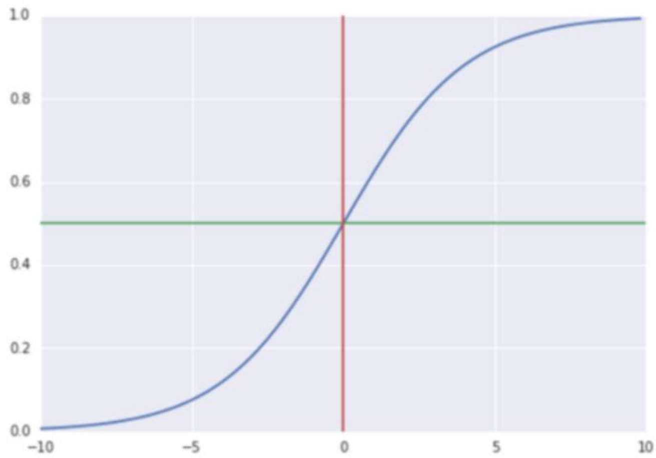


하지만 이 그래프는 최적의 그래프는 아니다. 최적의 그래프를 찾기 위해서 cost function을 이용한다.

2) Logistic Regression

선형 회귀 분석 모델은 이항분류에 적적하지 않다. 이런 경우 logistic function, 또는 sigmoid을 이용해 가설을 정의한다.

$$\mathrm{g(x)} = \frac{1}{1 + e^{-x}} = \mathrm{sigmoid(Wx + b)}$$



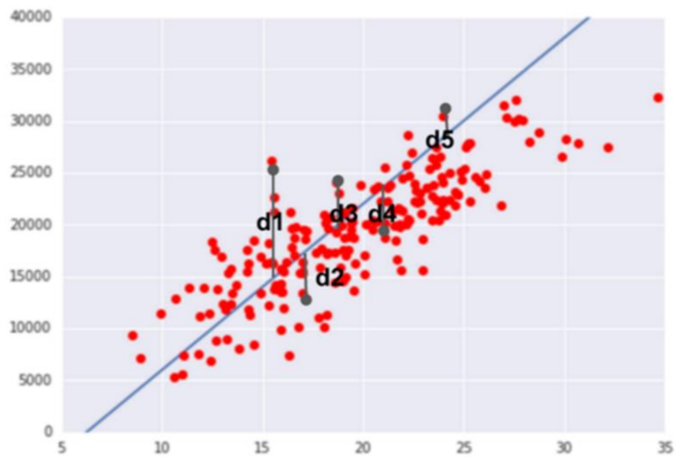
$$\mathrm{Wx} + \mathrm{b} < 0 \text{ 이면 } \mathrm{g(x)} = 0$$

$$\mathrm{Wx} + \mathrm{b} > 0 \text{ 이면 } \mathrm{g(x)} = 1$$

3. Cost Function

1) Linear Regression

다시 Linear Regression으로 돌아가보면 우리가 구하고자 하는 그래프는 실제 값에서 그래프까지 차이가 가장 작은 값을 구하고자 하는 것이다. 아래와 같이 y = Wx + b와 같은 그래프를 그렸다고 할 때



원래 값에서 우리가 예측한 값의 차이는 (원래값과 계산된 값의 차이 = 측정값 – 그래프의 값) 이 된다. 즉, $\mathrm{Error} = \hat{y} - y$가 된다. 모든 데이터의 error을 구하기 위해선 각 데이터의 error의 합을 구하면 되는데 error은 양수가 될 수 도 있고 음수가 될 수도 있다. 이런 상황에서 모든 error을 더하게 된다면 서로 상쇄되어 오차의 합을 제대로 측정할 수 없기 때문에 주로 error의 절댓값의 합이나 제곱합을 구한 뒤 데이터의 수만큼 나누어주어 평균을 구한다. 식으로 나타내면

$$\text{cost} = \frac{1}{m}\sum_{i=1}^{m}(H(x)_i - y_i)^2$$

이 함수를 Mean Squared Error라고 부른다.

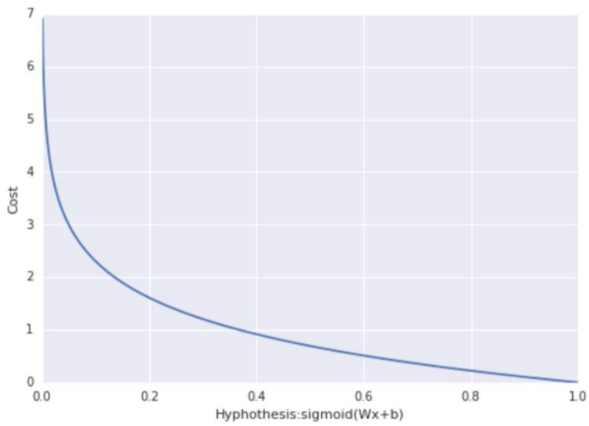
2) Logistic Regression

Logistic Regression에서 linear regression과 같은 cost function을 사용하게 된다면 optimization을 할 때 문제가 생긴다. 이에 대해서는 optimization파트의 gradient descent를 설명한 후 더 자세히 설명하도록 할 것이다.

따라서 logistic regression에서는 다른 함수를 사용하여 cost function을 정의한다. Cost function은 측정한 값과 가설에 의해 예측된 값의 차이를 나타내는 함수로, 개별 값이 작을수록 적절한 모델이 된다. 그래서 측정값과 가설 값이 같거나 유사할수록 cost function의 결과값이 작게 나와야 한다. 따라서 y=0인 경우와 y=1인 경우를 나눠서 생각해야 한다.

먼저 y=1일때를 알아보면 cost function은 다음과 같다

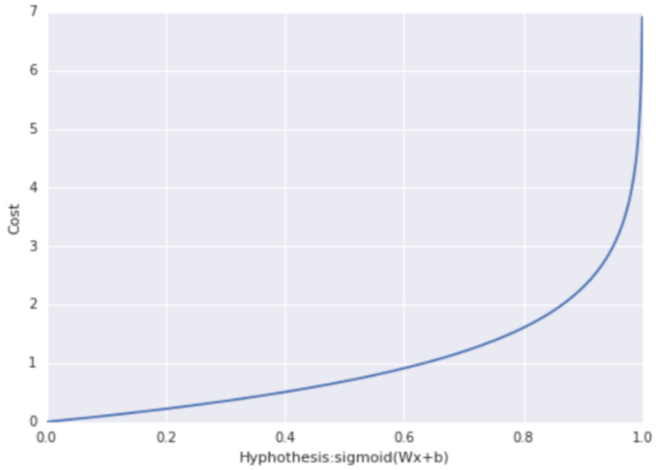
$$\text{cost} = \frac{1}{m}\sum -\log(g(x))$$



측정값이 1이기 때문에 g(x)의 결과가 1이면 예측이 잘 된 것이고 1에서 멀어져서 0으로 갈수록 예측된 값과 측정된 값의 차이가 크다고 할 수 있는데, 위의 그래프에서 보면 가설에 의해서 계산한 결과가 1에 가까울수록 cost는 0으로 수렴하고, 가설에 의해 계산된 결과가 0에 수렴할수록 cost는 높아지는 것을 볼 수 있다. 즉 y=1에서는 가설이 1에 수렴해야 하기 때문에 1에 가까워질수록 cost가 낮아지는 그래프를 띄게 된다.

Y=0일때문 마찬가지의 원리인데 측정값이 0이기 때문에 가설에 의한 결과값이 0이 되어야 한다. 0에서 멀어질 경우 cost가 늘어나고 0에서 가까워질 경우 cost가 줄어드는 형태의 비용 함수를 정의해야 한다.

$$\text{cost} = \frac{1}{m}\sum -\log(1 - g(x))$$



이 두 함수를 합치면 logistic regression의 cost function은 다음과 같이 나타낼 수 있다.

$$\text{cost} = \frac{1}{m}\sum -\text{ylog}(\text{g}(\text{x})) - (1-\text{y})\text{log}(1-\text{g}(\text{x}))$$

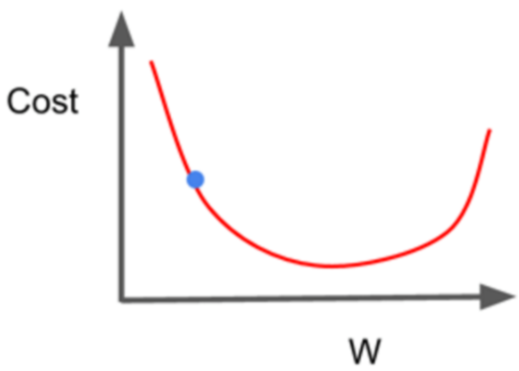
이 함수를 Cross Entropy라고 부른다.

4. Optimization

위에서 구한 cost function의 최소값을 찾는 알고리즘을 optimization이라고 하는데 상황에 따라 여러 종류의 optimizer을 사용할 수 있다. 여기서는 Gradient descent(경사 하강법)이라는 optimizer에 대해서 소개하도록 하겠다.

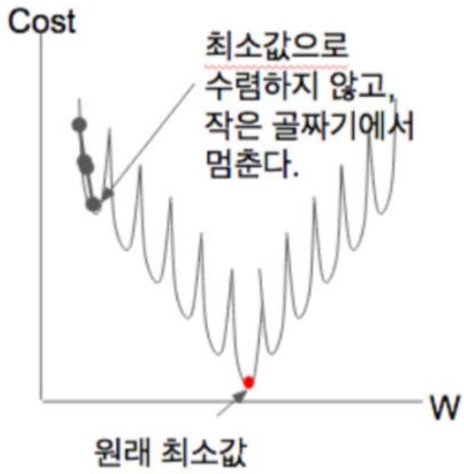
1) Gradient Descent

위에서 찾은 linear regression의 cost function을 그래프로 나타내면 다음과 같은 함수 형태가 된다.



이 그래프에서 W에 대한 적정 값에 대한 예측을 시작하는 점을 위의 그림에서 파란 점이라고 하면, 경사 하강법은 현재 W의 위치에 대해서, 경사가 아래로 되어 있는 부분으로 점을 움직이는 방법이다. 어느 방향으로 W를 움직이면 Cost 값이 작아지는지는 현재 W위치에서 비용 함수를 미분하면 된다

Logistic regression에서 linear regression에서 찾은 cost function을 그대로 사용하여 gradient descent를 통한 optimization을 하게 된다면 다음과 같은 그래프가 나오게 된다.



따라서 새로 정의한 logistic regression의 cost function을 통해 optimization을 진행하게 된다면 linear regression과 마찬가지로 global optimum에 도달할 수 있게 된다.