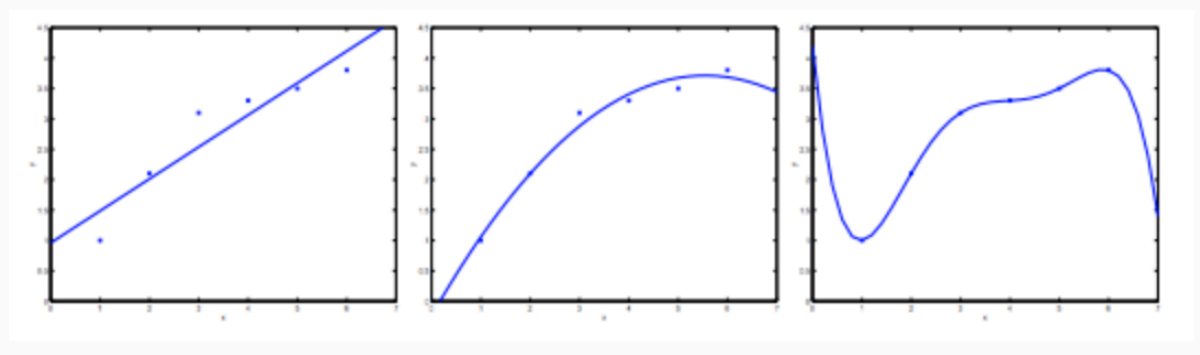
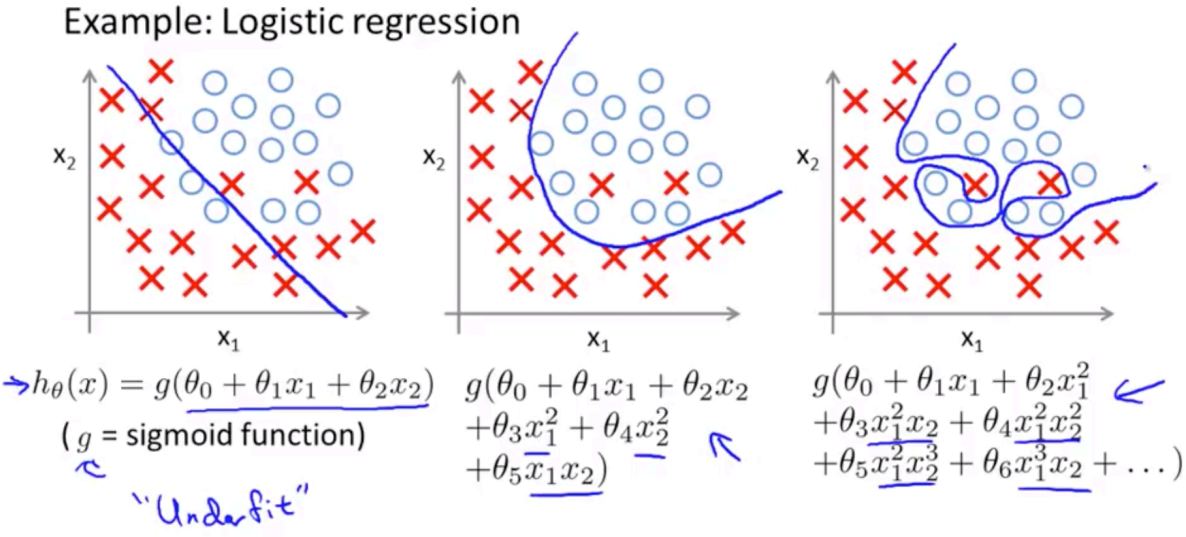
3주차 - Solving the Problem of Overfitting



- The Problem of Overfitting

위 그림과 같이 어떤 데이터가 주어졌을 때 선형회귀로 y값을 예측한다고 해보자. 만약 특성값 x에 대한 1차식으로 이루어져 있다면 왼쪽과 같을 것이고 차수가 올라갈 수록 오른쪽 그림과 같을 것이다.
왼쪽 그림과 같이 특성값이 부족하거나 y값을 전혀 예측할 수 없는 경우 'underfitting' 혹은 'high bias'라고 하며 이는 부족한 특성값으로 억지로 끼워맞추려고 한다는 뜻에서 bias를 사용했다.
가운데 그림은 적당히 작동하므로 'just right'이라고 하고
오른쪽 그림의 경우 training set에 있는 데이터에 대해서는 완벽히 작동하지만, 새로운 예제에 대해서는 전혀 예측하지 못한다. 이런 경우를 'overfitting', 'high variance'라고 한다. 이런 상황은 주로 특성값이 주어진 데이터보다 많을 때 발생하는데, 차수가 큰 다항함수가 가지고오는 변동성을 가지고 'variance'라고 이름이 붙여진 것이다.

이런 과적합 문제는 선형회귀뿐만 아니라 분류문제에서도 발생한다.



과적합 문제를 해결하는 방법은 주로 아래 2가지 방법이다.

```
1) Reduce the number of features:
* Manually select which features to keep.
* Use a model selection algorithm (studied later in the course).
```

```
2) Regularization
* Keep all the features, but reduce the magnitude of parameters θj.
(j번째 특성을 약화시키는 것을 말한다.)
* Regularization works well when we have a lot of slightly useful features.
```

- Regularization

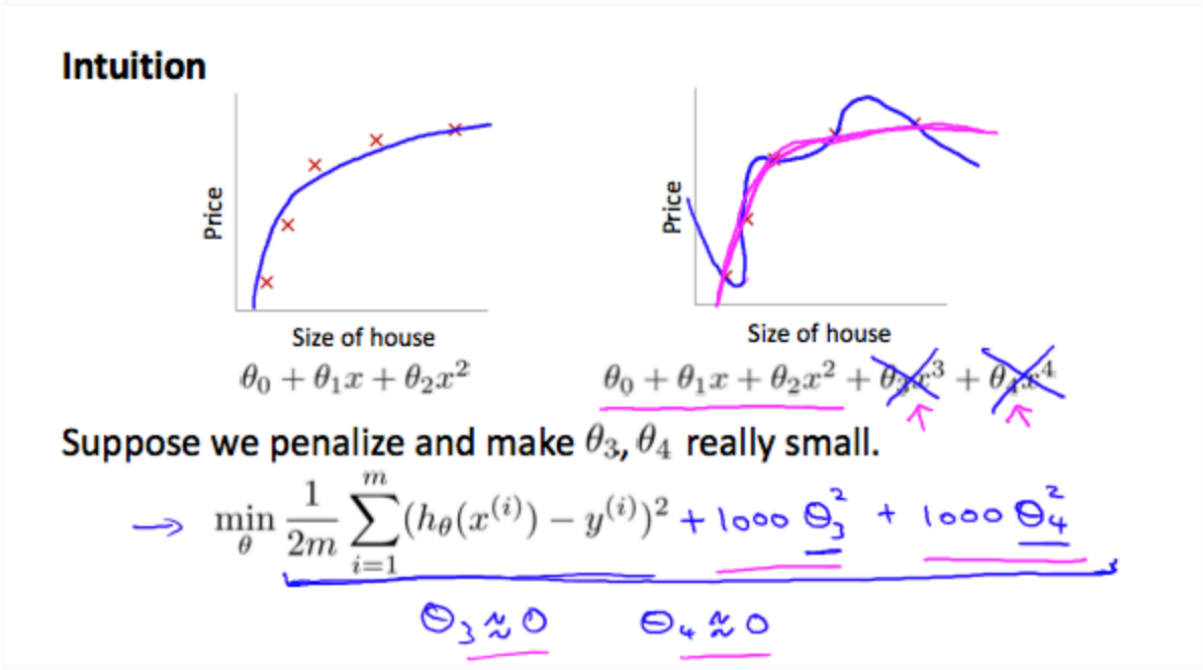
정규화는 overfitting의 상황을 완화하는 방법 중 한가지다.
기초적인 컨셉은 Cost Function을 약간 수정해서 특정 특성의 효과를 줄이는 것이다.

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

예를들어 위와 같이 특성 4개로 hypotheses를 잡았다고 생각해보면,
cost function에 다음과 같이 세번째와 네번째 특성에 대한 식을 추가할 수 있다.

$$min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000 \cdot \theta_3^2 + 1000 \cdot \theta_4^2$$

이렇게 되면 cost function의 값을 최소로 만들기 위한 theta의 값을 정할 때 기존 편차의 제곱을 줄이는 것보다 뒤의 추가된 식을 줄여야 하는 부담이 더 크다. 따라서 세번째와 네번째 theta값은 작은 값을 택할 수 밖에 없고 그 영향력이 줄게 되는 것이다.(이것을 theta j를 penalize한다고 부른다).



따라서 삼차식과 사차식의 영향이 줄었으므로 이차식과 거의 비슷한 양상을 띄게되고 위 그림에서 분홍색 곡선과 같이 overfitting을 완화할 수 있게 되는 것이다.
위 방식을 일반화 한다면 정규화는 다음과 같이 세타에 대한 이차항을 추가하여 세타가 작은 값을 가지게 하고 좀 더 '간단한' hypothesis를 만들게 한다.

$$min_\theta \frac{1}{2m} \left[\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right]$$

이때, 추가된 theta에 대한 식 앞에 붙은 람다를 regularization parameter라고 한다.

-주의-

theta0에 대한 regularization은 관습적으로 하지 않는다. 따라서 위 식에서 시그마를 보면 j=1부터 시작하는 것을 알 수 있다.

- Regularized Linear Regression
  - Gradient Descent

아래 과정은 정규화가 적용된 경사하강법이다.

$$
\begin{aligned}
&\text{Repeat } \{ \\
&\quad \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)} \\
&\quad \theta_j := \theta_j - \alpha \left[\left(\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \right) + \frac{\lambda}{m} \theta_j \right] \qquad j \in \{1, 2 \ldots n\} \\
&\}
\end{aligned}
$$

이전에도 말했듯이 정규화는 관습적으로 세타0 이후부터 진행하기 때문에 세타0는 기존 경사하강법을 적용하고 그 이후 파라미터에 대하여서는 정규화된 J(theta)를 적용하여 경사하강법을 적용한다.
대괄호 안에 있는 식은 정규화된 비용함수의 도함수를 적어놓은 것인데 자명하므로 증명은 생략한다.

한편, 위 식을 세타j에 대하여 묶고 기존 비용함수의 도함수를 따로 표기하면 다음과 같이 만들 수 있다.

$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

m(# of training set), 알파(learning rate), 감마(regularization parameter) 모두 양수 이므로 세타 j 앞에 곱해진 부분은 항상 1보다 작게 된다. 따라서 정리된 식을 직관적으로 보게되면, 세타의 값은 이전 값보다 작게 시작하면서 경사하강법은 똑같이 적용되는 것을 볼 수 있다. (정규화의 의미를 좀 더 직관적으로 볼 수 있다.)

- Normal Equation

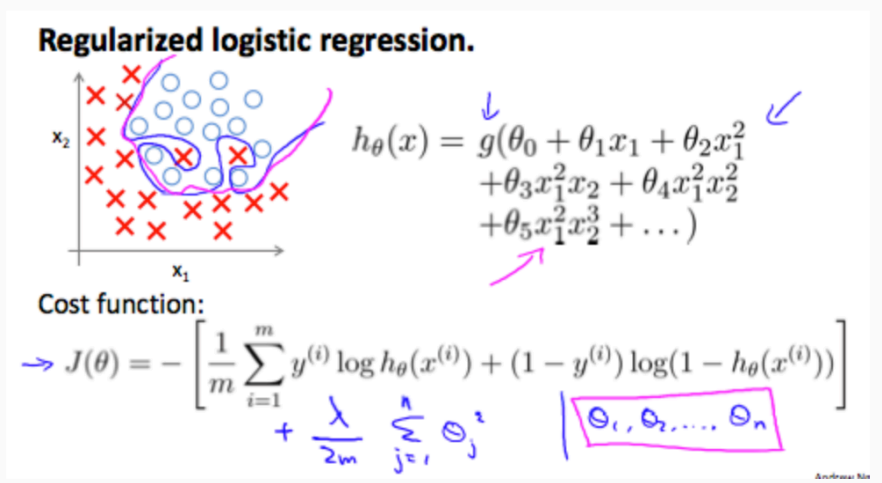
아래 과정은 정규화가 적용된 Normal Equation 방법이다.

$$
\begin{aligned}
&\theta = \left(X^T X + \lambda \cdot L\right)^{-1} X^T y \\
&\text{where } L = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}
\end{aligned}
$$

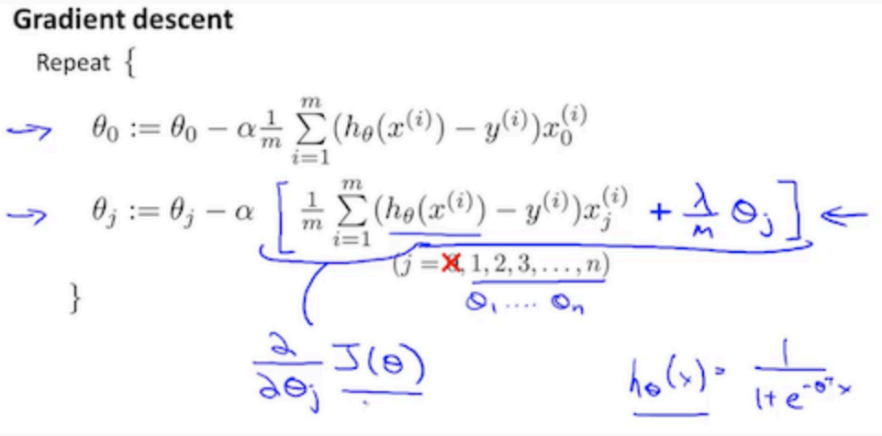
여기서 L은 (n+1) x (n+1) 행렬로 첫번째를 제외한 모든 대각부분이 1인 것을 볼 수 있는데 이것은 세타 0 부분은 정규화하지 않겠다는 의미를 가진다. 여기서 람다는 실수이다.
만약 m < n(특성이 데이터보다 많음)이면 T(X) * X의 역행렬이 존재하지 않고 m = n이면 존재하지 않을 수도 있는데, 여기서 감마 * L을 더해줌으로서 역행렬을 가질 수 있게 된다.
이것과 관련해서는 다음에 증명을 하도록 하겠다.

- Regularized Logistic Regression

로지스틱 회귀에 대해서도 선형회귀와 거의 비슷하게 정규화를 적용할 수 있다.



비용함수가 외형적으로 똑같았던 것처럼 정규화가 적용된 비용함수도 외형적으로 똑같다.



<참조>

regularization term에 2와 m이 나누어진 이유