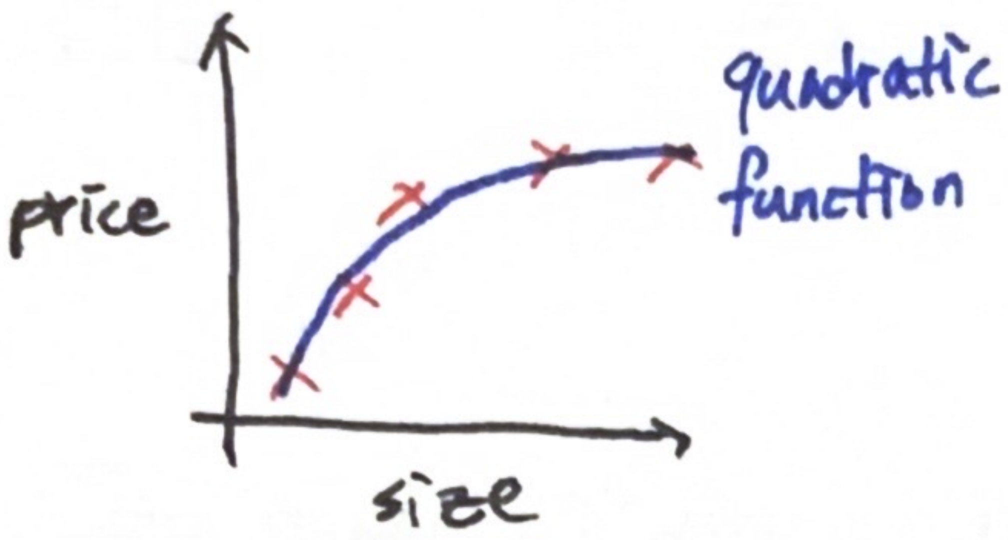
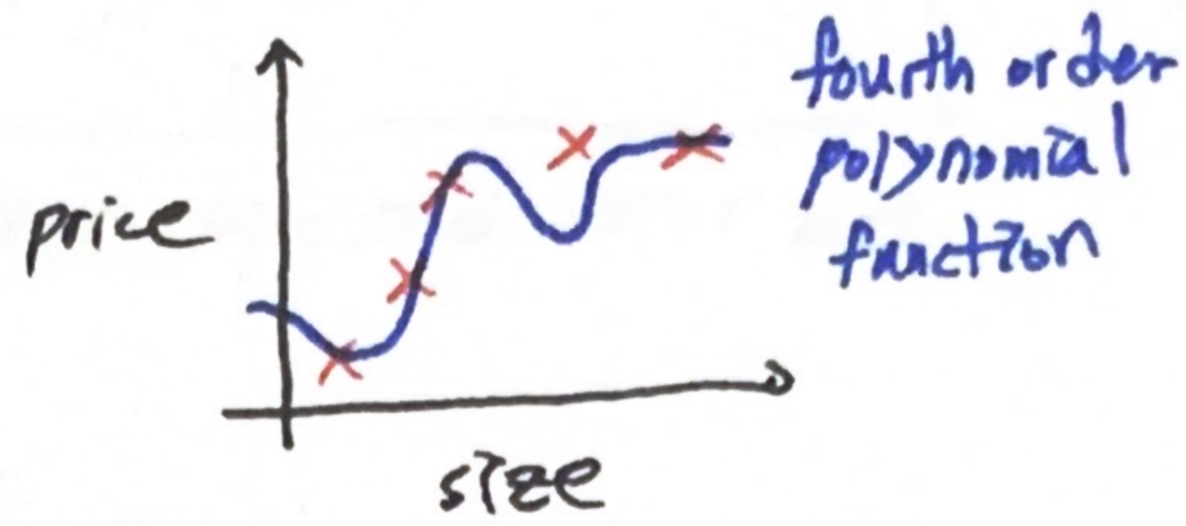
< Cost Function with Regularization >





$w_1x + w_2x^2 + b$

$\Downarrow$

"generalization"

$W_1x + W_2x^2 + W_3x^3 + W_4x^4 + b$

$\Downarrow$

"overfitting"

- W_3, W_4를 0에 가까운 수로 만들수 있다면 overfitting 된 모델을 quadratic function 모델처럼 만들수 있음

⇒ W_3, W_4에 penalty를 부여함으로써 overfitting 방지

Objective : make W_3, W_4 really small ($\simeq 0$)

$$\min_{\vec{w},b} \frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + 1000\,W_3^2 + 1000\,W_4^2 \quad (W_3 \simeq 0,\ W_4 \simeq 0)$$

- linear regression의 squared error cost function 예시
- W_3, W_4에 큰 값 (e.g.1000)을 곱함으로써 W_3, W_4의 값이 크다면 cost value가 커지게 수식을 수정하여 penalty를 부여해준 것
- 위 cost function 이 작은 값을 갖기 위해서는 W_3, W_4 값이 매우 작아야함 ($\simeq 0$)
- W_3, W_4를 매우 작은 값으로 minimize 함으로써 4차함수 모델(overfit)을 사용하더라도(=모든 feature를 사용하더라도) 2차함수 모델(generalized)에 거의 근접하도록 할 수 있음

* 만약 feature 수가 너무 많아 어떤 parameter (w_j)를 minimize 해야할 지 모를 때?

⇒ "모든 feature에 적용하는 parameter들에 penalty를 부여"

= 모든 feature를 사용하면서 overfitting을 방지 = "Regularization"

$$J(\vec{w},b) = \frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

(n = number of features
m = number of training examples)

* lambda λ = regularization parameter

= penalty for all parameters