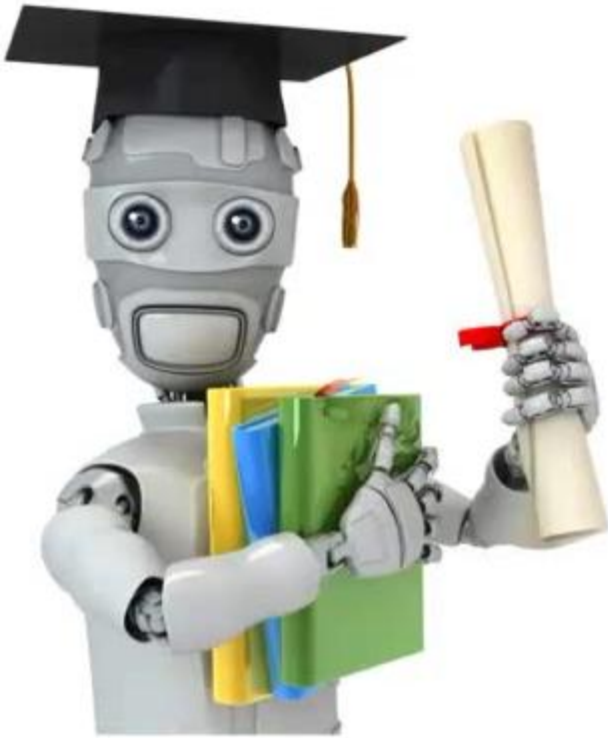
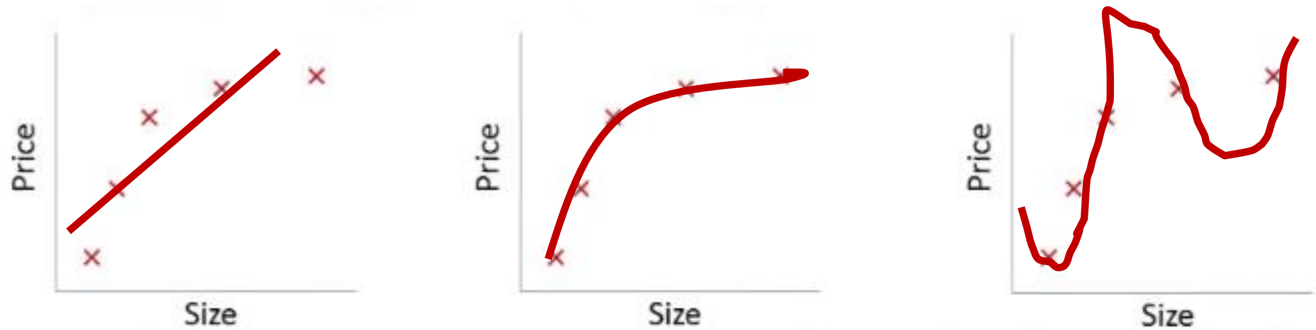


Machine Learning

Regularization

The problem of overfitting

Example : Linear regression (housing prices)



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

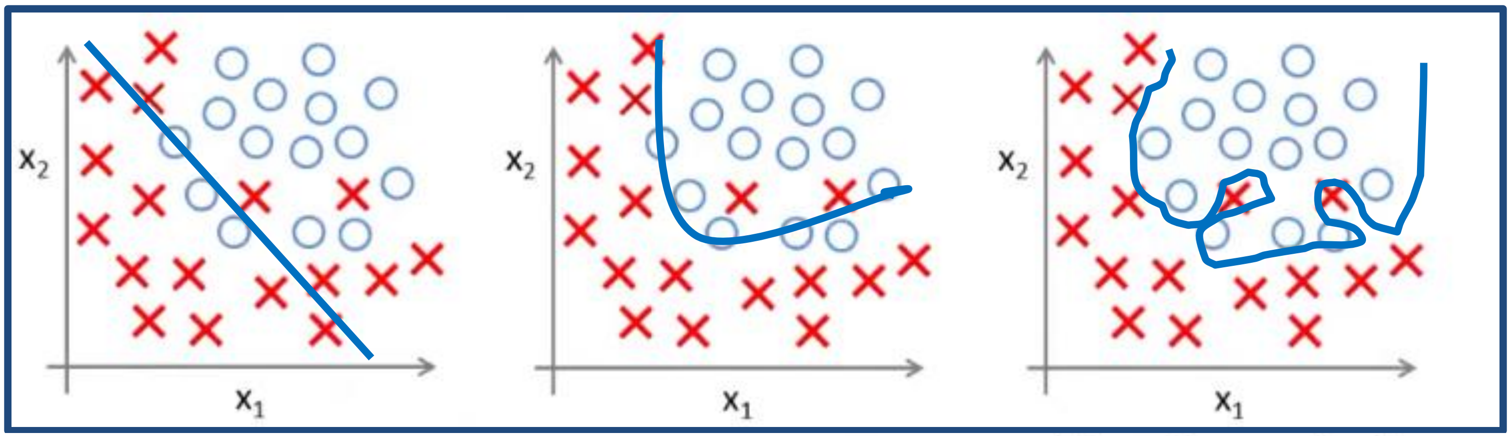
'underfit' 'high bias'
(红点的纵坐标偏离直线）

'just right'

'overfit' 'high variance'
（横坐标改变一点点，纵坐标可能发生很大改变）

Overfitting: If we have too many features, the learned hypothesis may fit the training set very well ($J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \approx 0$), but fail to generalize to new examples (predict prices on new examples).

Example: Logistic regression



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

(g = sigmoid function)

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2$
$+\theta_3 x_1^2 + \theta_4 x_2^2$
$+\theta_5 x_1 x_2)$

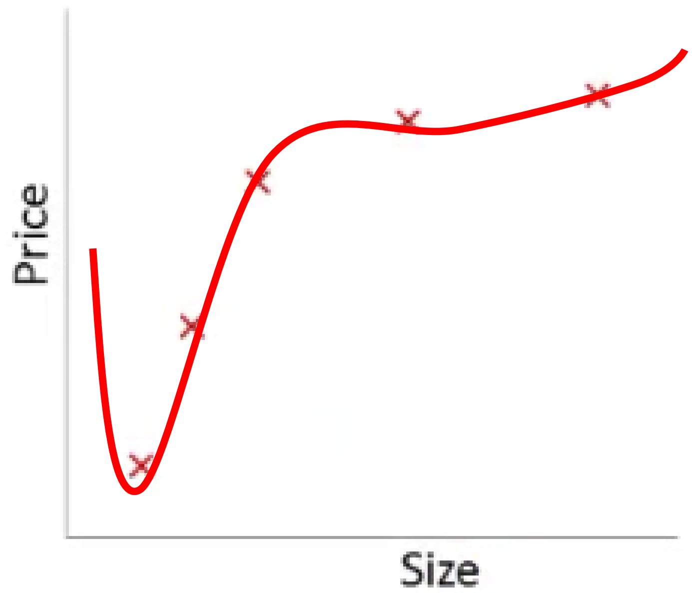
$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2$
$+\theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2$
$+\theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots)$

‘underfit’

‘overfit’

Addressing overfitting

$x_1 =$ size of house
$x_2 =$ no. of bedrooms
$x_3 =$ no. of floors
$x_4 =$ age of house
$x_5 =$ average income in neighborhood
$x_6 =$ kitchen size
$\vdots$
x_{100}



Addressing overfitting

Options

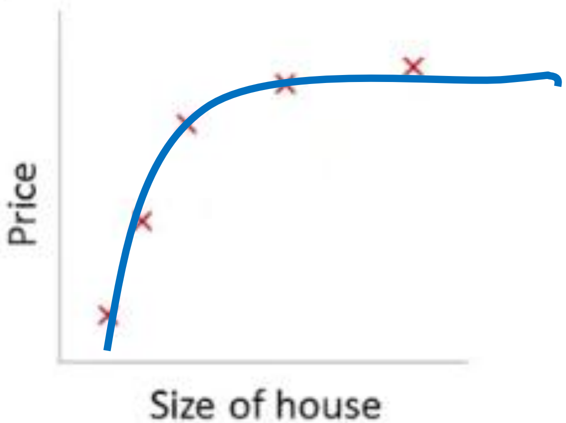
1. Reduce number of features.
   - Manually select which features to keep.
   - Model selection algorithm (later in course).
2. Regularization.
   - Keep all the features, but reduce magnitude/values of parameters θ_j.
   - Works well when we have a lot of features, each of which contributes a bit to predicting y.

Machine Learning

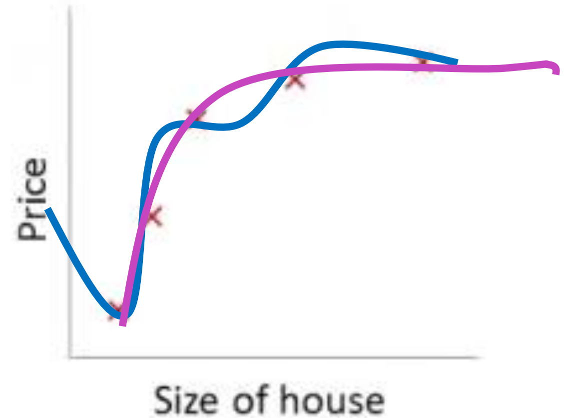
Regularization

Cost function

Intuition



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



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

Suppose we penalize and make θ_3, θ_4 really small.

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

$$\theta_3 \approx 0, \theta_4 \approx 0$$

Regularization.

Small values for parameters $\theta_0, \theta_1, \ldots, \theta_n$

- "Simpler" hypothesis
- Less prone to overfitting

$\theta_3 \approx 0, \theta_4 \approx 0$

Housing:

- Features: $x_1, x_2, \ldots, x_{100}$
- Parameters: $\theta_0, \theta_1, \theta_2, \ldots, \theta_{100}$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \boxed{+\lambda \sum_{i=1}^{n} \theta_j^{\,2}}$$

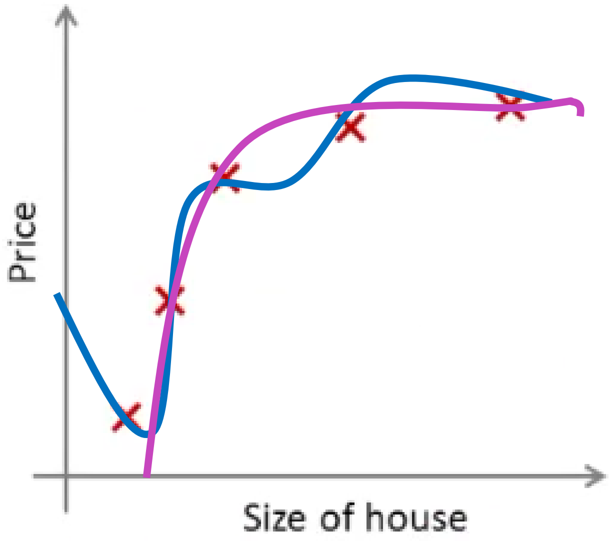
Regularization term

Regularization.

Regularization parameter

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

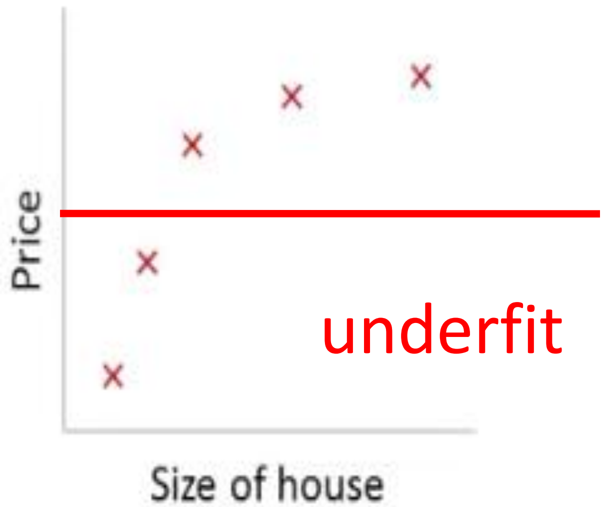
$$\min_\theta J(\theta)$$



In regularized linear regression, we choose θ to minimize

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

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?



$$\theta_1 \approx 0, \theta_2 \approx 0,$$

$$\theta_3 \approx 0, \theta_4 \approx 0,$$

$$h(x) = \theta_0$$

$$\theta_0 + \cancel{\theta_1 x} + \cancel{\theta_2 x^2} + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4} = h(x)$$

Gradient descent

Repeat {

$$\theta_0 := \theta_0 - \boxed{\alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}}$$

$$\boxed{\frac{\partial}{\partial\theta_0}J(\theta)}$$

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

$$(j = \cancel{0}, 1, 2, 3, \dots, n)$$

$$\boxed{\frac{\partial}{\partial\theta_j}J(\theta)}$$

}

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

< 1 0.99

Normal equation

$$X = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad \boxed{R^m}$$

m*(n+1)

$$\min_{\theta} J(\theta)$$

$$\theta = (X^T X + \ldots\ldots \lambda \underbrace{\begin{bmatrix} 0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}}_{(n+1)*(n+1)} \ldots\ldots)^{-1} X^T y$$

Non-invertibility (optional/advanced).

Suppose $\underset{\text{(\#examples)}}{m} \leq \underset{\text{(\#features)}}{n}$,

$$\theta = \underline{(X^T X)}^{-1} X^T y$$

Non-invertibility

If $\lambda > 0$,

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

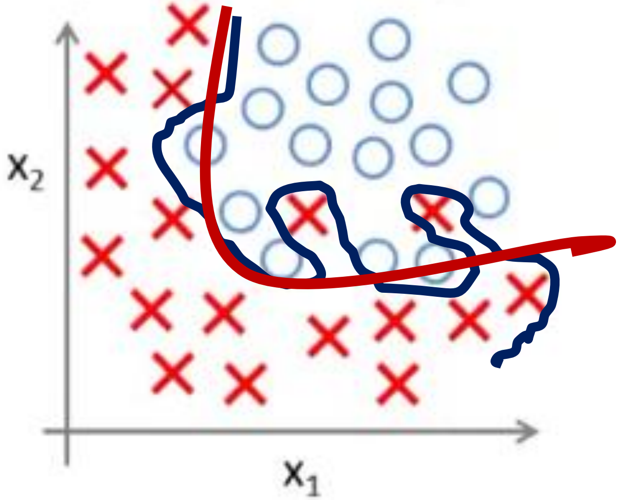
invertibility

Machine Learning

Regularization

Regularized logistic regression

Regularized logistic regression.



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \dots)$$

Cost function:

$$J(\theta) = -\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\right]$$

$$\boxed{+\frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^{\ 2}}$$

Gradient descent

Repeat $\{$

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

$$(j = 0, 1, 2, 3, \ldots, n)$$

$\}$

Gradient descent

Repeat {

$$\boxed{h_\theta(x) = \frac{1}{1+e^{-\theta^T X}}}$$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

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

$$(j = \cancel{0}, 1, 2, 3, \ldots, n)$$

}

$$\boxed{\frac{\partial}{\partial \theta_j} J(\theta)}$$

$$\boxed{\theta_1, \theta_2 \ldots.. \theta_n}$$