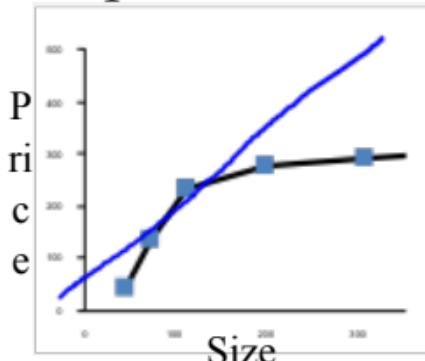


Machine Learning

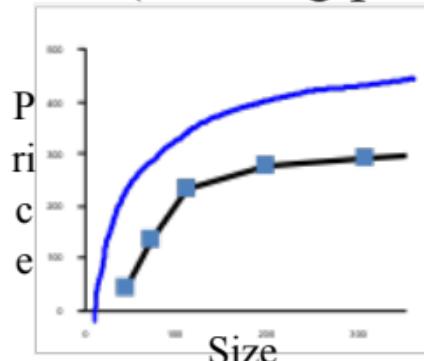
Regularization

The problem of overfitting

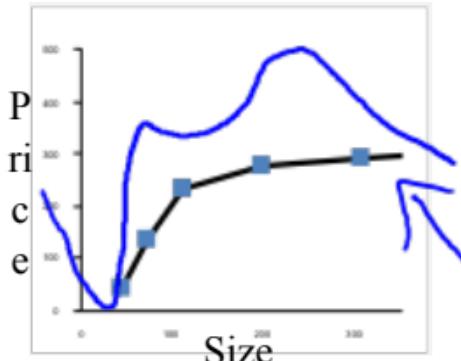
Example: Linear regression (housing prices)



$\rightarrow \theta_0 + \theta_1 x$
"Underfit" "High bias"



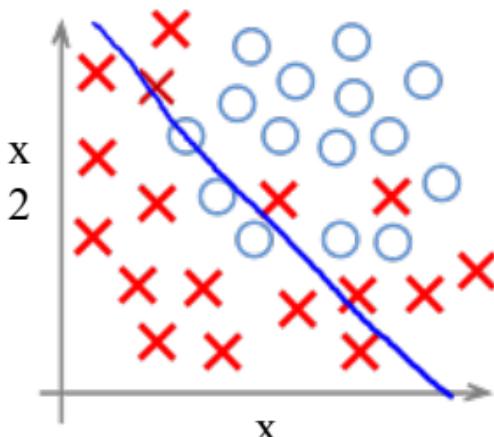
$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2$
"Just right"



$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$
"Overfit" "High variance"

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

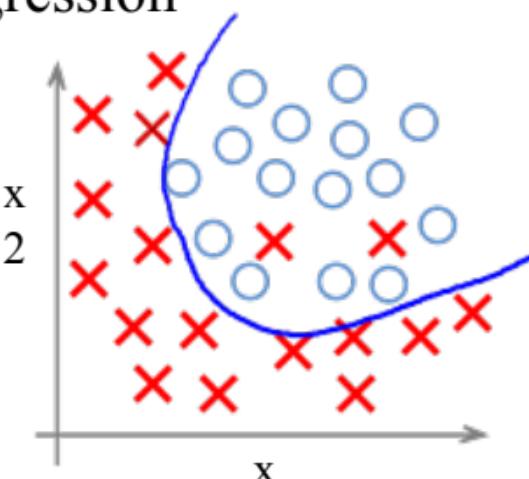


$$\rightarrow h_{\theta}(x) = g(\underline{\theta_0 + \theta_1 x_1 + \theta_2 x_2})$$

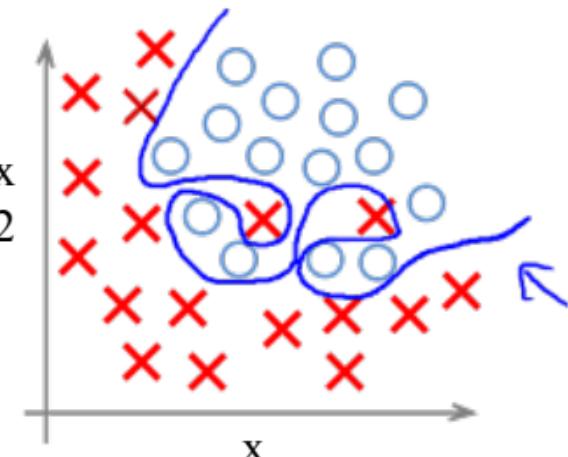
(g = sigmoid function)



"Underfit"



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

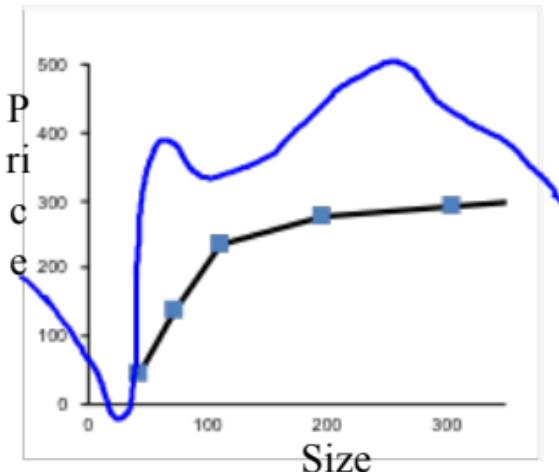


$$g(\underline{\theta_0 + \theta_1 x_1^1 + \theta_2 x_1^2} \\ + \theta_3 \underline{x_1^2 x_2} + \theta_4 \underline{x_1^2 x_2^2} \\ + \theta_5 \underline{x_1^2 x_2^3} + \theta_6 \underline{x_1^3 x_2} + \dots)$$

"Overfit"

Addressing overfitting:

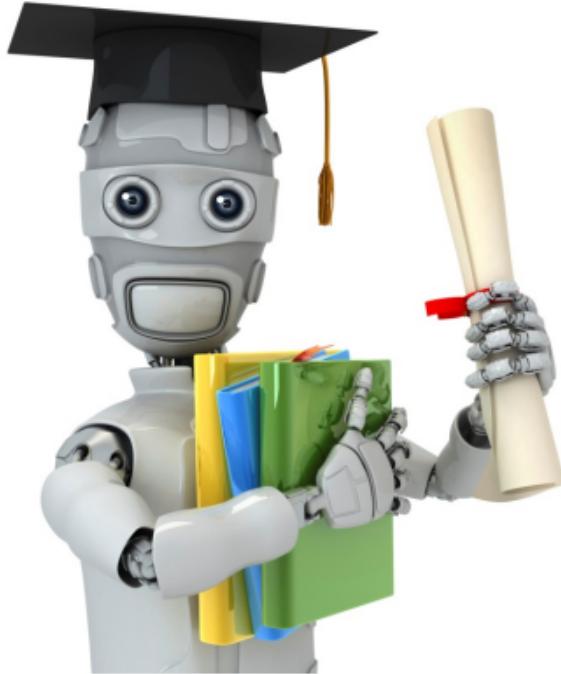
- x_1 = size of house
- x_2 = no. of bedrooms
- x_3 = no. of floors
- x_4 = age of house
- x_5 = average income in neighborhood
- x_6 = kitchen size
- :
- x_{100}



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 .

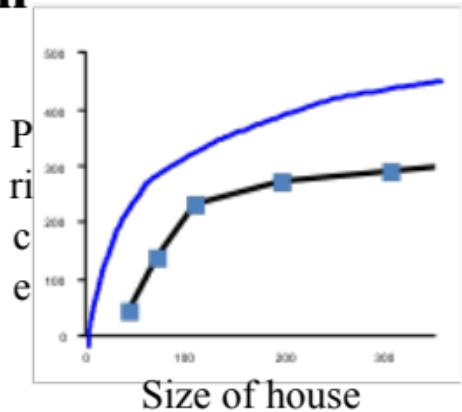


Machine Learning

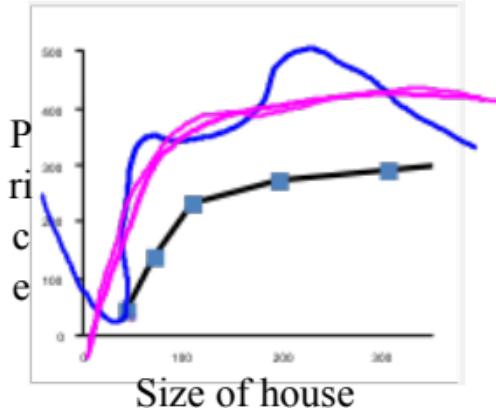
Regularization

Cost function

Intuition



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



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

Suppose we penalize and make θ_3, θ_4 really small.

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

$$\underline{\theta_3 \approx 0}$$

$$\underline{\theta_4 \approx 0}$$

Regularization.

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

“Simpler” hypothesis

Less prone to overfitting

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

Housing:

Features: x_1, x_2, \dots, x_{100}

Parameters: $\theta_0, \theta_1, \theta_2, \dots, \theta_{100}$

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

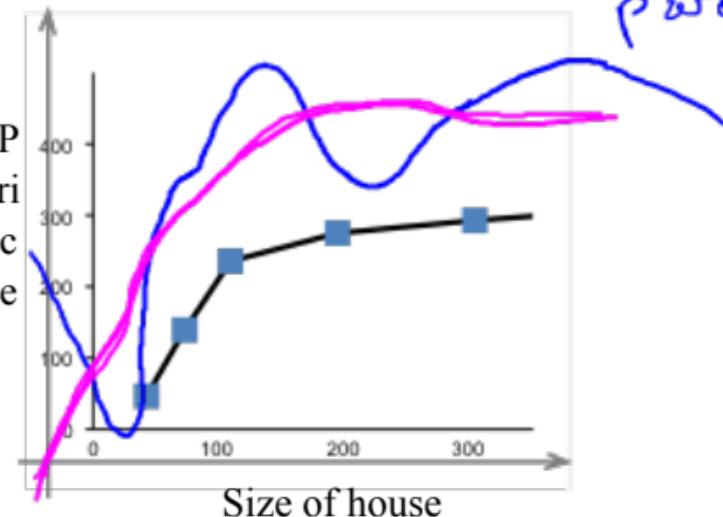
~~$\theta_1, \theta_2, \theta_3, \dots, \theta_{100}$~~

Regularization.

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

min _{θ} $J(\theta)$

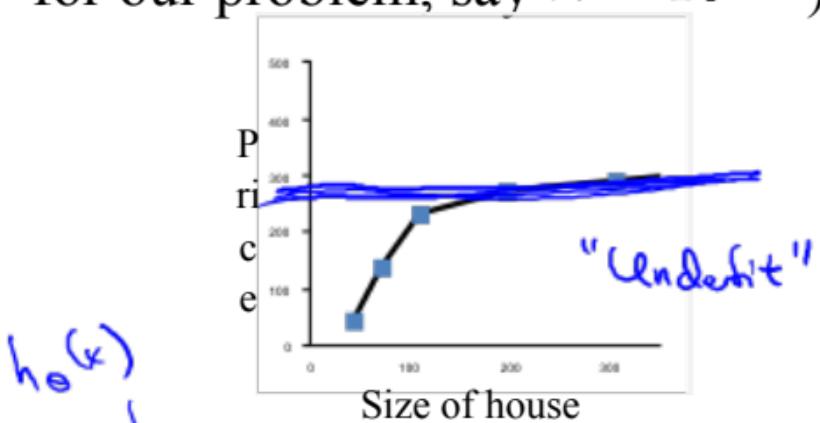
regularization parameter



In regularized linear regression, we choose θ to minimize

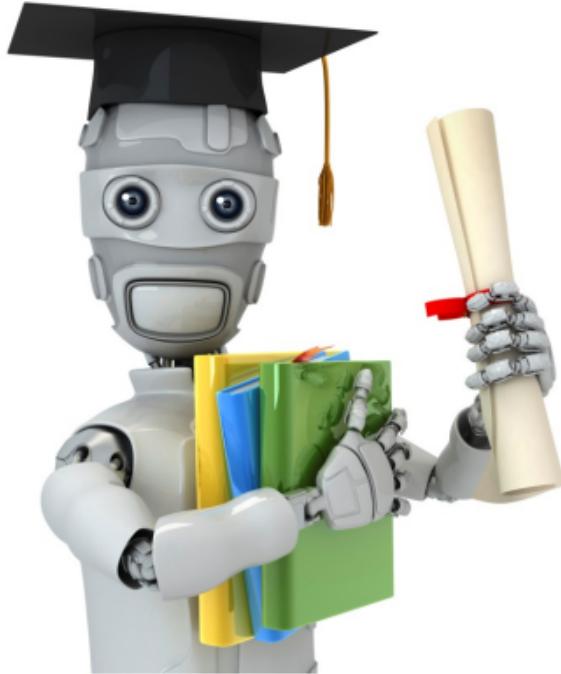
$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n \theta_j^2 \right]$$

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



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

$$\begin{aligned} & \underline{\theta_1}, \underline{\theta_2}, \underline{\theta_3}, \underline{\theta_4} \\ & \theta_0 \approx 0, \theta_2 \approx 0 \\ & \theta_3 \approx 0, \theta_4 \approx 0 \\ & \boxed{\hat{h}_\theta(x) = \theta_0} \end{aligned}$$



Machine Learning

Regularization

Regularized linear regression

Regularized linear regression

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

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


Gradient descent

Repeat {

$$\frac{\theta_0}{\uparrow}$$

$$\theta_1, \theta_2, \dots, \theta_n$$

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

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

$$\rightarrow \theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} - \frac{\lambda}{m} \theta_j \right] \quad (j = \cancel{0}, 1, 2, 3, \dots, n)$$

}

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

$$\rightarrow J(\theta)$$

$$1 - \alpha \frac{\lambda}{m} < 1 \quad \underline{0.99} \quad \theta_j \times 0.99$$

$$\theta_j^*$$

Normal equation

$$X = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} \quad \text{m} \times (n+1)$$

$$y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad \mathbb{R}^m$$

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

$$\frac{\partial}{\partial \theta_j} J(\theta) \stackrel{\text{set } 0}{=} 0 \quad \Rightarrow$$

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

e.g. n=2

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(n+1) × (n+1)

Non-invertibility (optional/advanced).

Suppose $m \leq n$, 
(#examples) (#features)

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

 non-invertible / singular

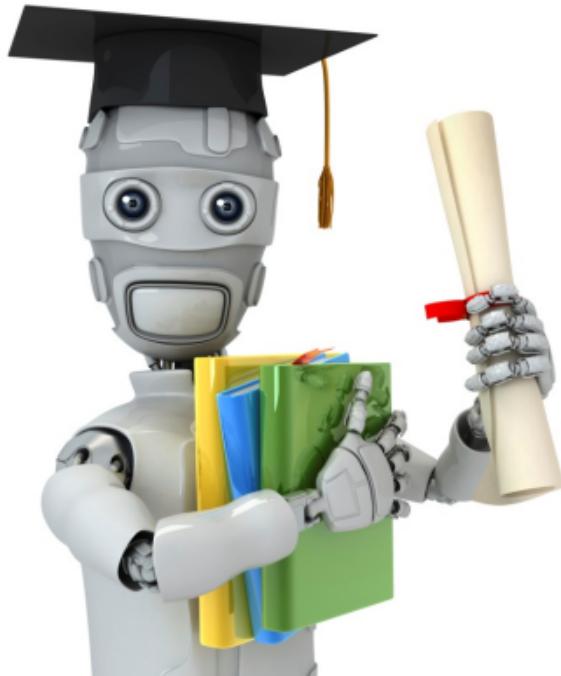
pinu

 inv
 λ

If $\lambda > 0$,

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

 invertible.

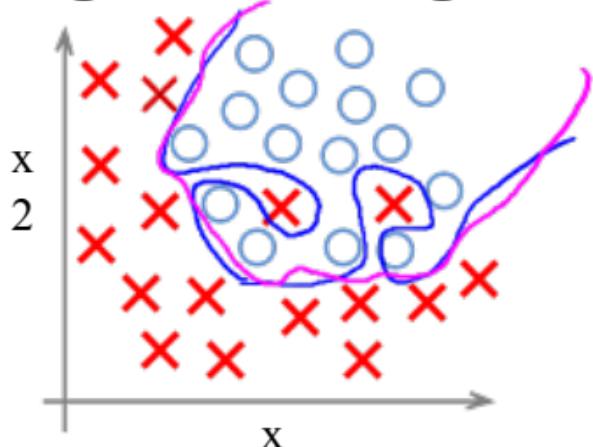


Machine Learning

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:

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

$\theta_1, \theta_2, \dots, \theta_n$

Gradient descent

Repeat {

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

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

$\underbrace{(j = 1, 2, 3, \dots, n)}$
 $\underbrace{\theta_1, \dots, \theta_n}$

}

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

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

Advanced optimization

f_{minunc} (Q cost function) $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$ theta(1) \leftarrow
theta(2) \leftarrow theta(n+1)

→ **function [jVal, gradient] = costFunction(theta)**

jVal = [code to compute $J(\theta)$] ;

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

→ gradient(1) = [code to compute $\frac{\partial}{\partial \theta_0} J(\theta)$] ;

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

→ gradient(2) = [code to compute $\frac{\partial}{\partial \theta_1} J(\theta)$] ;

$$\left(\frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)} \right) - \frac{\lambda}{m} \theta_1$$

$J(\theta)$

→ gradient(3) = [code to compute $\frac{\partial}{\partial \theta_2} J(\theta)$] ;

$$\vdots \quad \left(\frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)} \right) - \frac{\lambda}{m} \theta_2$$

gradient(n+1) = [code to compute $\frac{\partial}{\partial \theta_n} J(\theta)$] ;