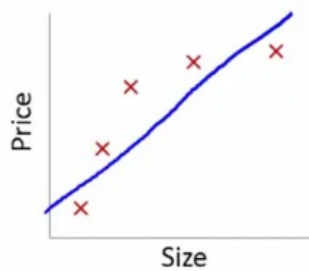
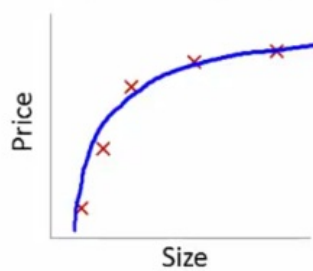


1. 过拟合 (Overfit) 。太多的变量虽然代价函数非常接近0, 但是使得预测偏差变大。线性回归中：

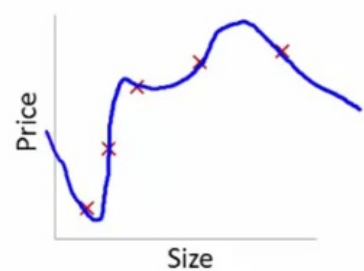
### Example: Linear regression (housing prices)



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



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



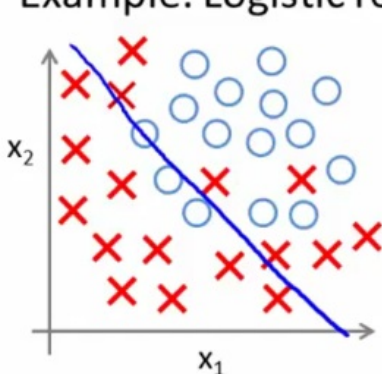
$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$   
"Overfit" "High variance"

**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 (发生 predict prices on new examples).

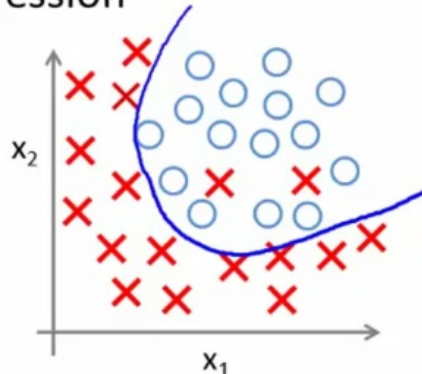
Andrew Ng

逻辑回归中：

### Example: Logistic regression

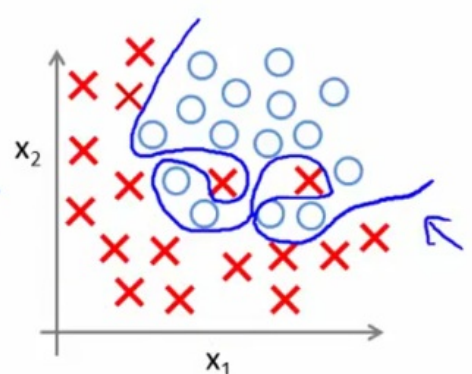


$\rightarrow h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$   
( $g$  = sigmoid function)  
"Underfit"



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

这又是一个过拟合例子



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

2. 解决过拟合：

(1) 减少特征数量。

- 手动选择保持特征
- 模型选择算法

问题是：舍弃掉一些特征便舍弃掉一些信息。

(2) 正规化 (Regularization)

- 保持特征数量，但是减少theta的影响 (减少theta的值)
- 特征数量多的时候很有效，每个特征都有影响。

## Addressing overfitting:

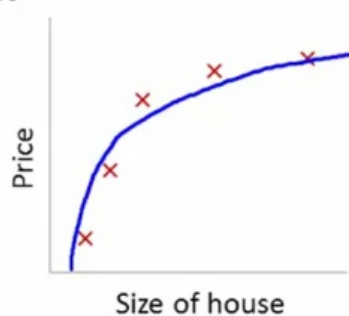
### Options:

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  $\theta_j$ .
  - Works well when we have a lot of features, each of which contributes a bit to predicting  $y$ .

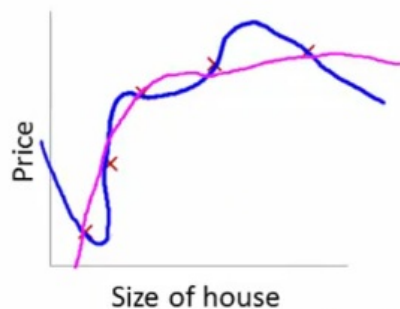
正如我们在房价的例子中看到的那样

通过减小theta的值来减小某些特征的贡献：

### 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  $\theta_3, \theta_4$  really small.

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

这些很小的项 贡献很小  
 $\theta_3 \sim 0$     $\theta_4 \sim 0$

注意，不对theta0进行正规化：

## Regularization.

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

- “Simpler” hypothesis
- Less prone to overfitting

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

Housing:

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

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

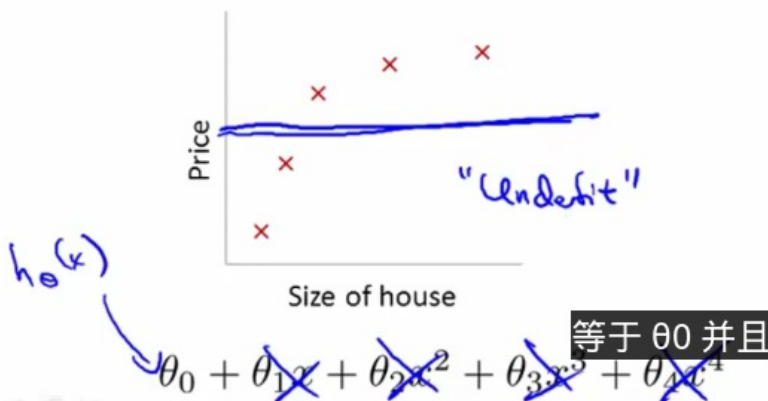
这只有非常小的差异

如果lambda太大， $\theta_j(j \neq 0)$ 都被训练为约为0， $h(x)$ 约为 $\theta_0$ ，模型欠拟合。

In regularized linear regression, we choose  $\theta$  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  $\lambda$  is set to an extremely large value (perhaps far too large for our problem, say  $\lambda = 10^{10}$ )?



$$\begin{aligned} \theta_1, \theta_2, \theta_3, \theta_4 \\ \theta_1 \approx 0, \theta_2 \approx 0 \\ \theta_3 \approx 0, \theta_4 \approx 0 \\ h_{\theta}(x) = \theta_0 \end{aligned}$$

## Gradient descent

$$\theta_0, \theta_1, \theta_2, \dots, \theta_n$$

Repeat {

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

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

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

(j = 1, 2, 3, ..., n)

}

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

$$1 - \alpha \frac{\lambda}{m} < 1$$

更小了 0.99

$$\theta_j \times 0.99$$

$$\theta_j^2$$

正则化正规方程法：

## Non-invertibility (optional/advanced).

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

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

non-invertible / singular

pinv

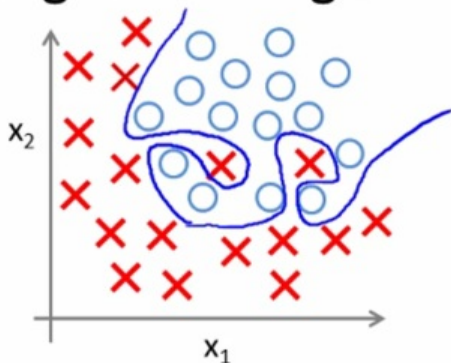
If  $\lambda > 0$ ,

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

正则化逻辑回归：



## 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:

$$\rightarrow J(\theta) = - \left[ \frac{1}{m} \sum_{i=1}^m y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\theta}(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2 \quad | \quad \theta_1, \theta_2, \dots, \theta_n$$

Andrew N

## Gradient descent

Repeat {

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

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

$\underbrace{\phantom{\frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}}}_{\theta_1, \dots, \theta_n}$

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

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

高级优化算法的正则化：

$f_{\min}(\theta)$  (cost function...)

$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$

$\theta_0 \leftarrow \theta_1 \leftarrow \dots \leftarrow \theta_n$

$\text{gradient} = \text{costFunction}(\theta)$

$$\theta(n+1)$$

```
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  $\frac{\partial}{\partial \theta_0} J(\theta)$ ];

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

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

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

→ **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  $\frac{\partial}{\partial \theta_n} J(\theta)$ ];
```