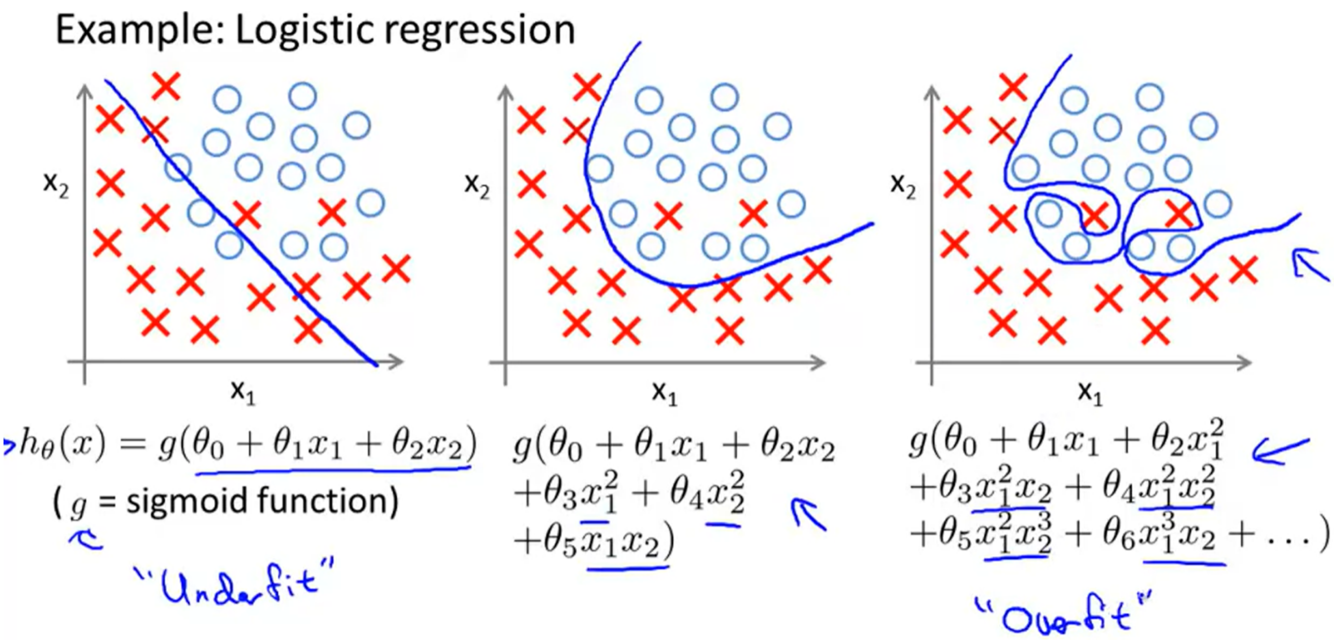
Week 3.4 Solving the Problem of Overfitting

The Problem of Overfitting

> **Overfitting:** If we have too many features, the learned hypothesis may fit the training set very well, but fail to generalize to new examples (predict prices on new examples).

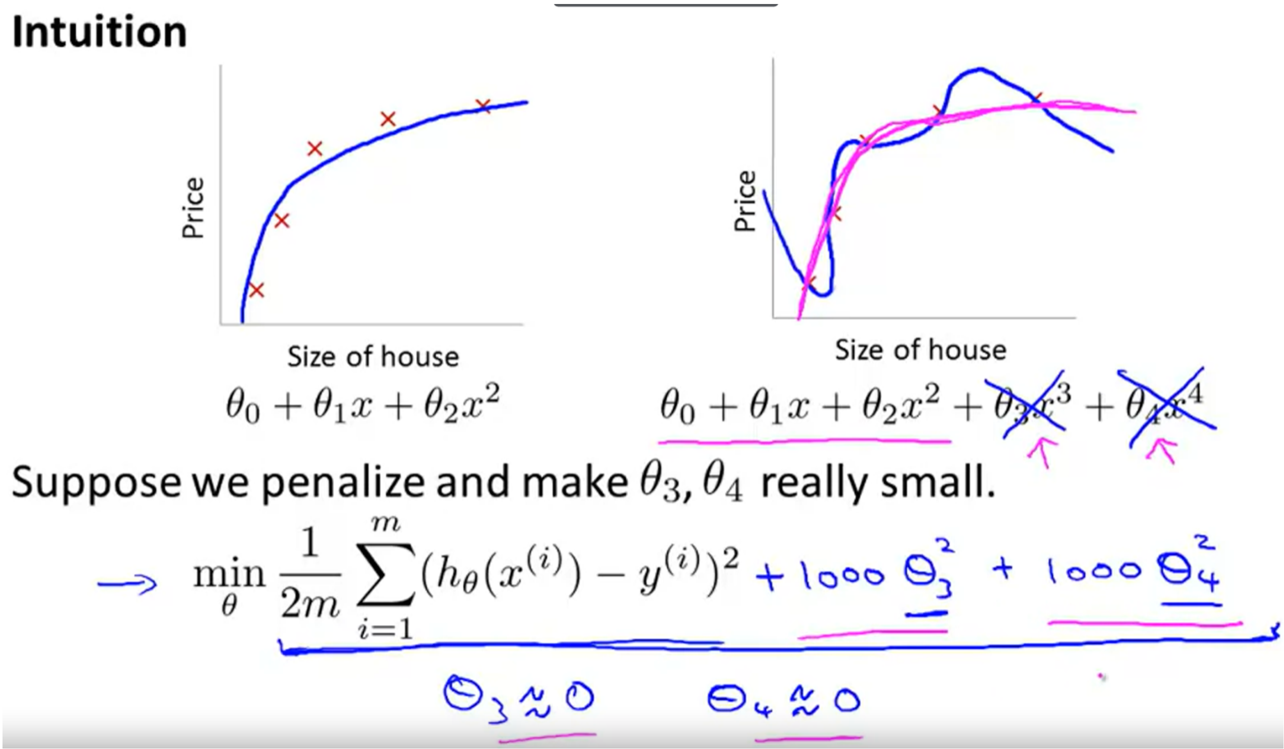


Options:

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

Cost Function

- penalize parameters (small values)



- what if λ is too large: **results in underfitting**

Regularized Linear Regression

- For gradient descent, we repeat
- $$\theta_0 = \theta_0 - \alpha \tfrac{1}{m} \textstyle\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)} x_0^{(i)})$$
  $$\theta_j = \theta_j - \alpha [\tfrac{1}{m} \textstyle\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \tfrac{\lambda}{m} \theta_j]$$
- We rewrite formula for θ_j:
- $$\theta_j = \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{x(i)}) x_j^{(i)}$$
- $1 - \alpha \frac{\lambda}{m}$ is a bit smaller than 1

Normal Equation Method

Non-invertibility (optional/advanced).

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

$\theta = \underbrace{(X^TX)^{-1}}_{\text{non-invertible / singular}}X^Ty$

pinv

inv

If $\lambda > 0$,

$$\theta = \Bigg(\underbrace{X^TX + \lambda \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}}_{\text{invertible.}}\Bigg)^{-1} X^Ty$$

Regularized Logistic Regression

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

- Pseudocode:

---

fminunc (@costFunction...)

$\theta = \begin{bmatrix}\theta_0 \\ \theta_1 \\ \vdots \\ \theta_n\end{bmatrix}$ theta(1), theta(2), theta(n+1)

Advanced optimization

`function [jVal, gradient] = costFunction(theta)`

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

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

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

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

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

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

$J(\theta)$

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

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

⋮

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

-

- [哲哲的ML笔记（十二：逻辑回归中的代价函数） - 简书 (jianshu.com)](jianshu.com)

  推导过程:

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

  $$y^{(i)}\log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1-y^{(i)}\right)\log\left(1-h_\theta\left(x^{(i)}\right)\right) = y^{(i)}\log\left(\frac{1}{1+e^{-\theta^T x^{(i)}}}\right) + \left(1-y^{(i)}\right)\log\left(1-\frac{1}{1+e^{-\theta^T x^{(i)}}}\right)$$
  $$= -y^{(i)}\log\left(1+e^{-\theta^T x^{(i)}}\right) - \left(1-y^{(i)}\right)\log\left(1+e^{\theta^T x^{(i)}}\right)$$

  所以: $\frac{\partial}{\partial\theta_j}J(\theta) = \frac{\partial}{\partial\theta_j}[-\frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\log\left(1+e^{-\theta^T x^{(i)}}\right) - \left(1-y^{(i)}\right)\log\left(1+e^{\theta^T x^{(i)}}\right)]]$

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

-