

# 模型选择与正则化 (Model Selection & Regularization )

梁毅雄

Machine Learning

[yxliang@csu.edu.cn](mailto:yxliang@csu.edu.cn)

Some materials from Andrew Ng, Zico Kolter, Hung-yi Lee  
and others

# 偏差与方差(Bias and Variance)

假设用某个函数 $h(x)$ 去近似真实函数 $y(x)$ , 其偏差和方差分别为

$$bias(h(x)) = E[h(x) - y(x)]$$

$$var(h(x)) = E\{h(x) - E[h(x)]\}^2 = E[h(x)^2] - E[h(x)]^2$$

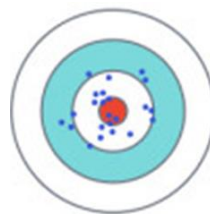
Large Bias

Small Variance



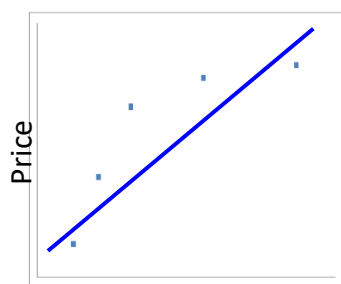
Small Bias

Large Variance



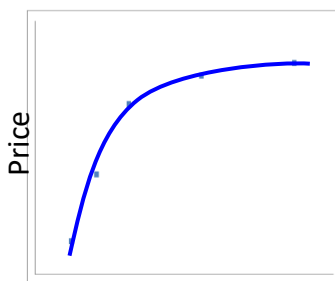
# 过拟合问题

例子: 线性回归 (房屋价格)



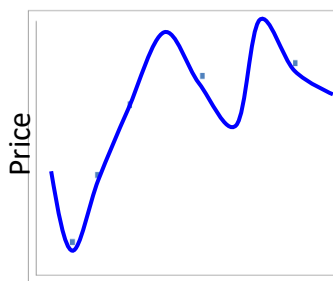
Size  
 $\theta_0 + \theta_1 x$

Underfitting: Large bias



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

Good fitting



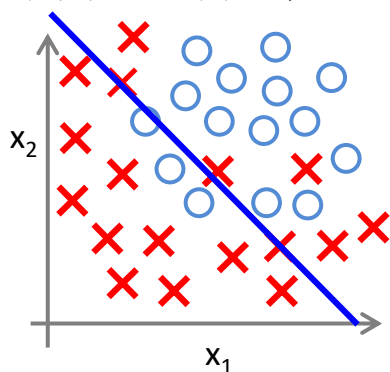
Size  
 $\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Overfitting: Large Variance

过拟合: 如果多项式阶数较大, 训练得到的模型对于训练集能正确分类  $J(\theta) = -\frac{1}{2m} \sum_{i=1}^m [(h_{\theta}(x^{(i)}) - y^{(i)})]^2 \approx 0$ , 但是对于新的样本预测效果却不好.

# 过拟合问题

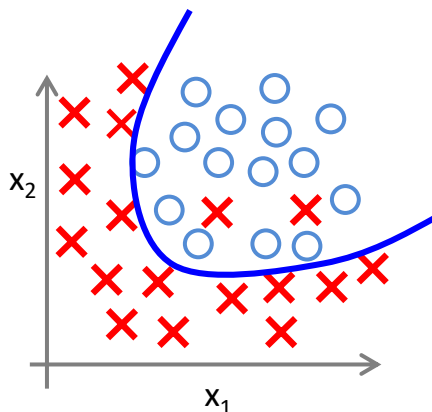
例子: 逻辑回归



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

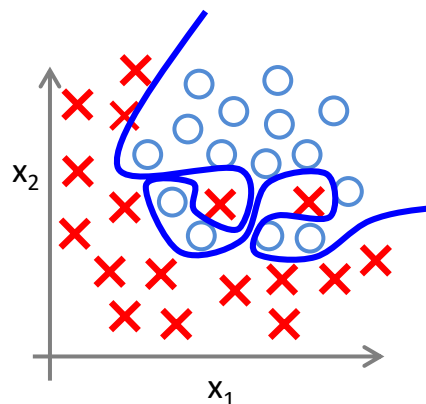
( $g$  = sigmoid function)

Underfitting: Large bias



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

Good fitting



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

Overfitting: Large Variance

过拟合: 如果多项式阶数较大, 训练得到的模型对于训练集能正确分类( $J(\theta) = -\frac{1}{2m} [\sum_{i=1}^m y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))]$   $\approx 0$ ), 但是对于新的样本预测效果却不好.

# 过拟合问题

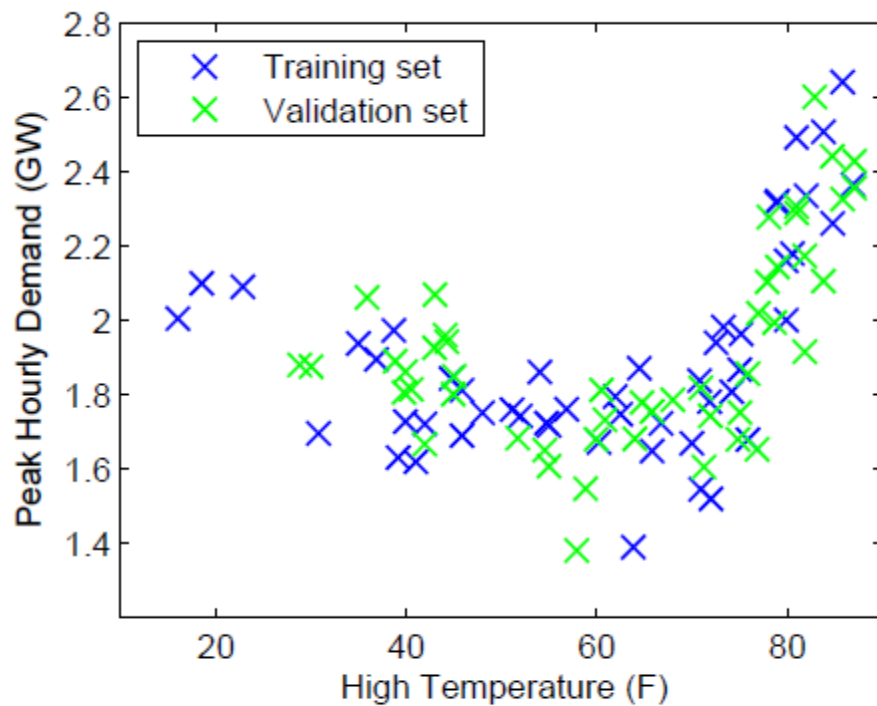
- 实际应用中容易出现过拟合（模型足够复杂）
- 问题1： 如何判断是否出现了过拟合或者欠拟合问题？  
（诊断）
- 问题2： 如何解决过拟合或者欠拟合问题？  
（开处方治疗）

# 模型选择

$$\theta^* = \arg \min_{\theta} \frac{1}{m} \sum_{i=1}^m \ell(h_{\theta}(x^{(i)}), y^{(i)})$$

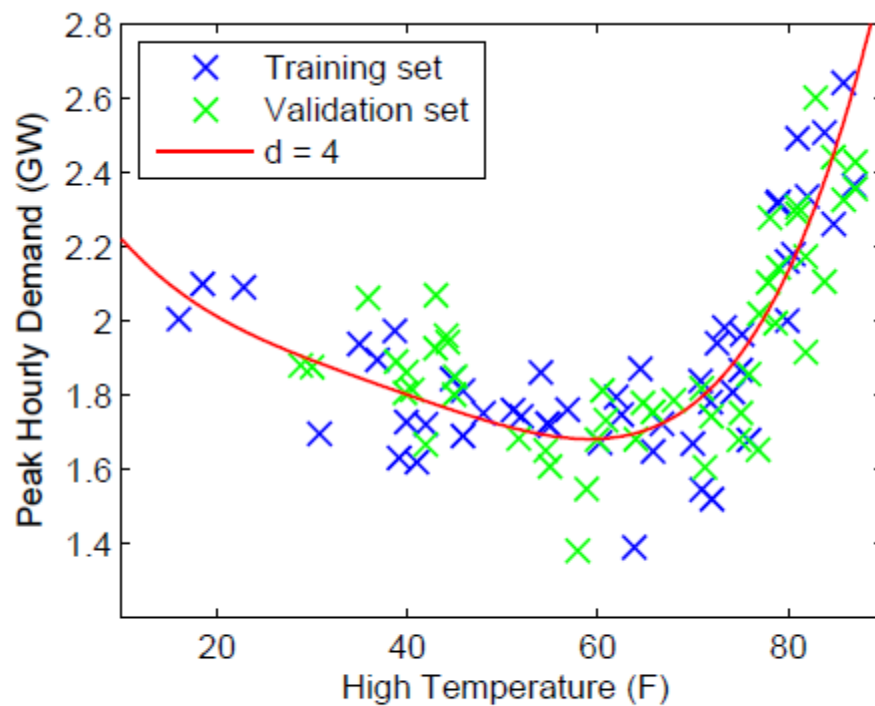
- 最小化训练集上的损失(损失错误)
- 一般而言，模型越复杂（如多项式阶数越高或特征越多），训练得到的模型经验错误越低，但却更容易出现过拟合
- 选择哪个模型更合适？
- 把训练集**随机**分成两部分：用于训练参数的训练集和用于模型选择的验证集(Validation Set)

# 模型选择



Training set and validation set

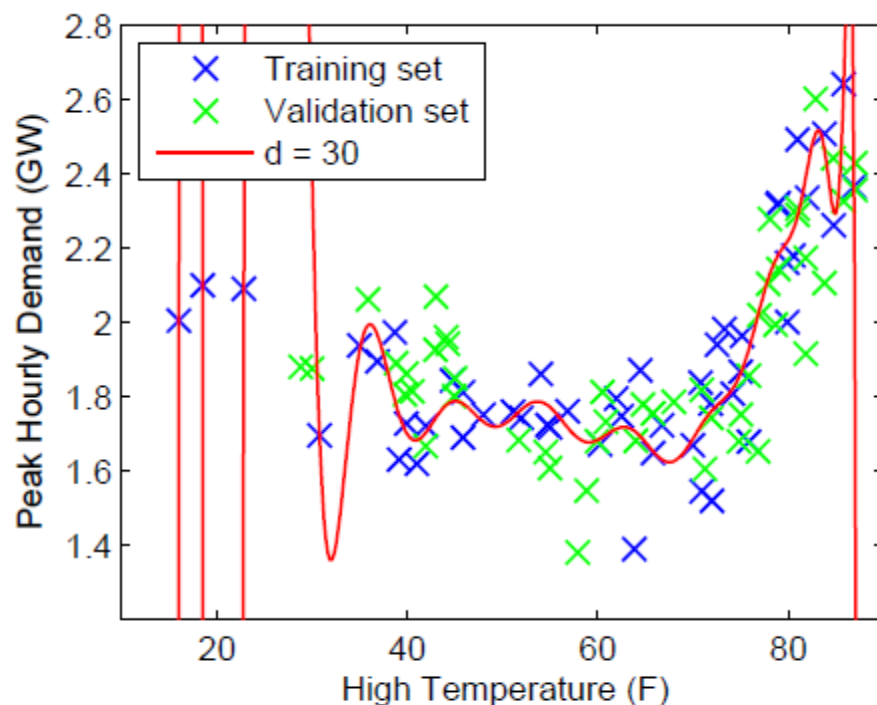
# 模型选择



Training set and validation set, fourth degree polynomial



# 模型选择

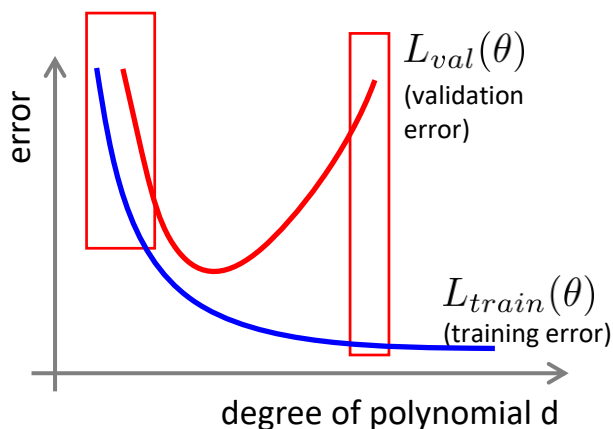


Training set and validation set, 30th degree polynomial

# 诊断偏差和方差

训练误差: 
$$L_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

验证误差: 
$$L_{val}(\theta) = \frac{1}{2m_{val}} \sum_{i=1}^{m_{val}} (h_{\theta}(x_{val}^{(i)}) - y_{val}^{(i)})^2$$



**偏差大(underfit):**

训练误差: 大

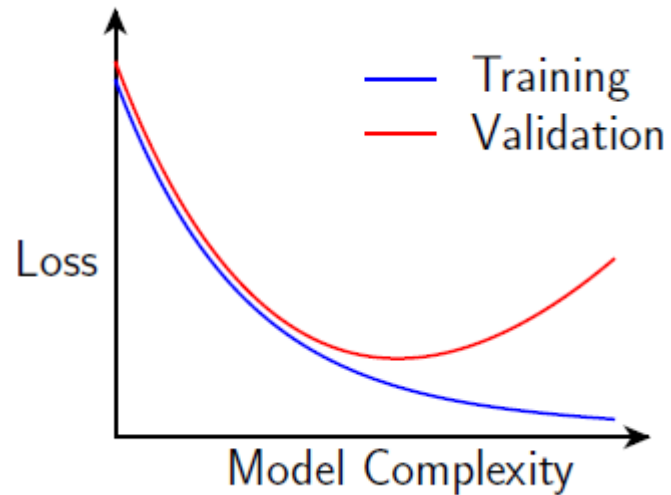
训练误差与验证误差差别较小

**方差大(overfit):**

训练误差: 小

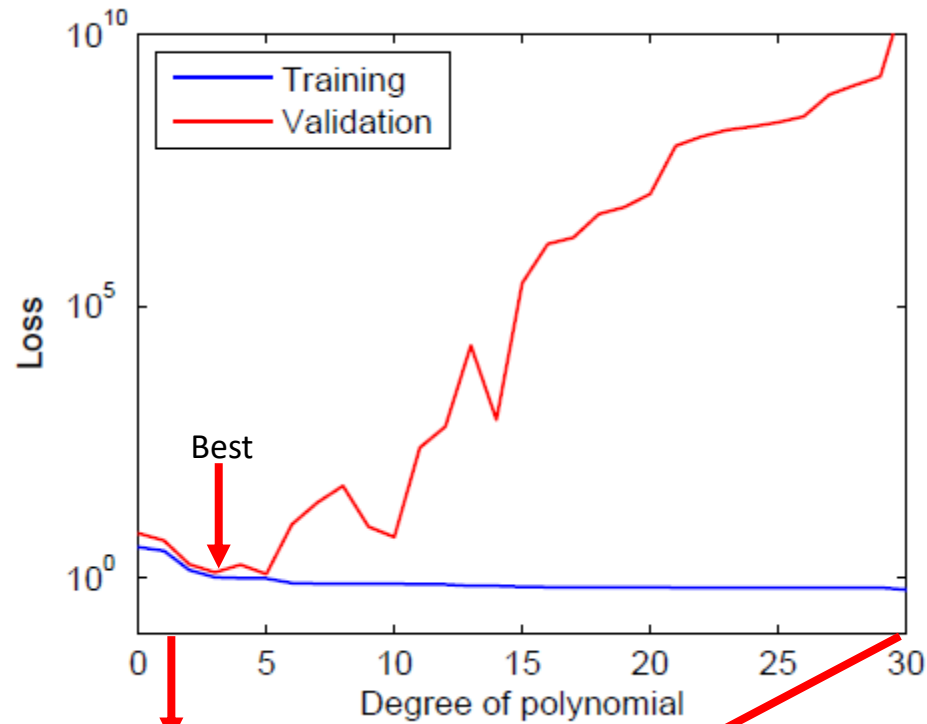
验证误差远大于训练误差

# 模型选择

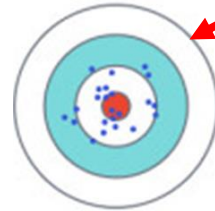
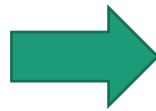
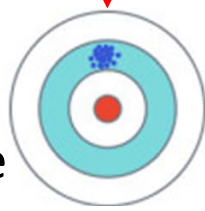


- There is usually a trade-off between bias and variance.
- Select a model that balances two kinds of error to minimize total error

# 模型选择



Large Bias  
Small Variance



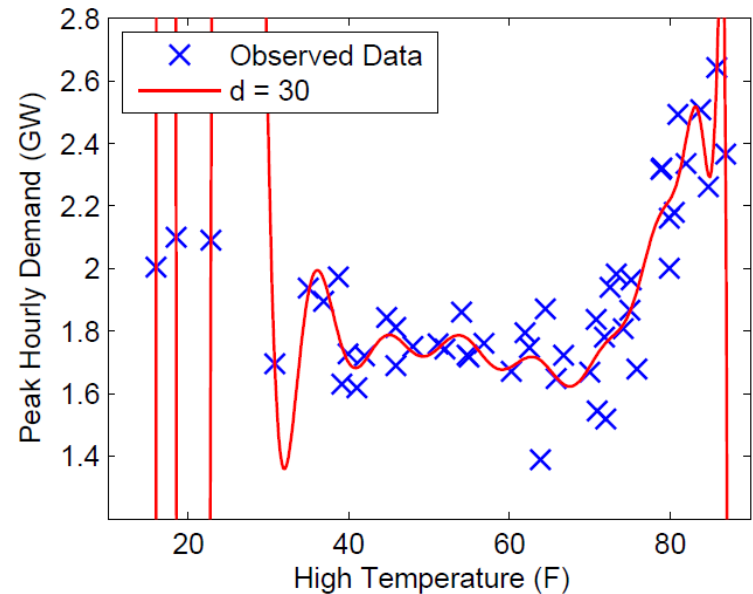
Small Bias  
Large Variance

# 解决欠拟合和过拟合问题

- 欠拟合(Large Bias): 增加模型的复杂度
  - 收集新的特征
  - 增加多项式组合特征
  - ... ( $x_1^2, x_2^2, x_1x_2$ , etc)
- 过拟合(Large Variance)
  - 增加数据 (Very effective, but not always practical)
  - 降低模型的复杂度
    - 减少特征 (人为的选择一些特征, 特征选择)
    - 正则化(Regularization): 非常有效的方法, 可大幅度降低方差(增加偏差)
    - ...

# 正则化线性回归

- Regularized Linear Regression
  - Intuition: A 30<sup>th</sup> degree polynomial that passes exactly through many of the data points requires very large entries in  $\theta$
  - We can directly prevent large entries in  $\theta$  by penalizing the magnitude of its entries



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

$\lambda$  : 正则化参数 (因子)

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

思考：正则化参数  $\lambda$  的取值范围？

# 正则化线性回归

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

$$J(\theta) = L(\theta) + \lambda R(\theta)$$

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

$$L(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

思考：若 $\lambda$ 的值足够大，如  $\lambda = 10^{10}$ ，下面正确的是：

- A. Algorithm works fine
- B. Algorithm fails to eliminate overfitting
- C. Algorithm results in underfitting
- D. Algorithm results in overfitting
- E. Gradient descent will fail to converge

# 正则化线性回归

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

**Gradient descent**

Repeat {

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

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

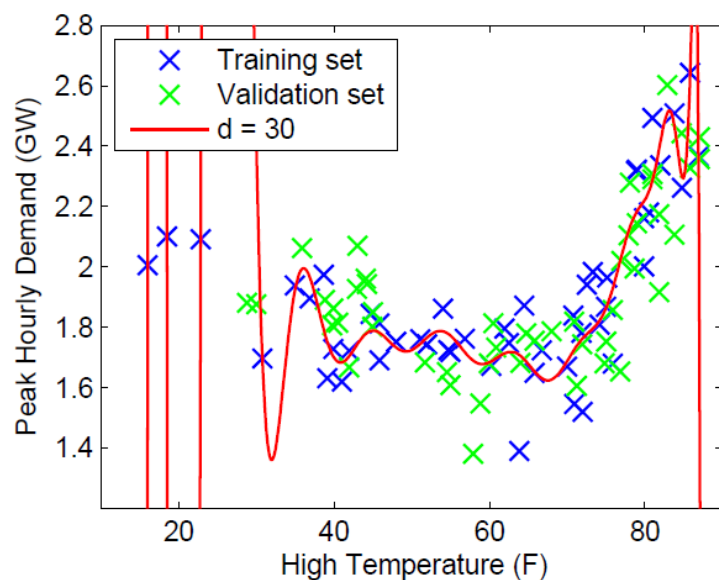
( $j = \text{red X}, 1, 2, 3, \dots, n$ )

}

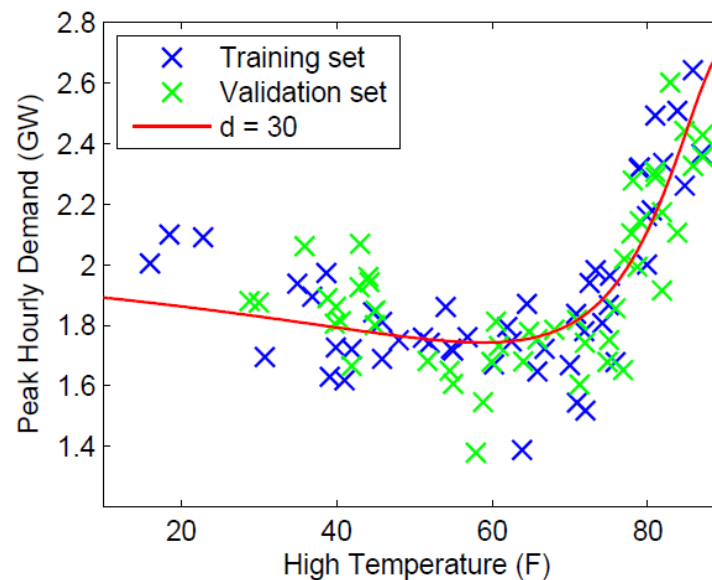
$$\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$



# 正则化线性回归



Degree 30 polynomial, with  $\lambda = 0$  (unregularized)



Degree 30 polynomial, with  $\lambda = 1$

# 正则化线性回归

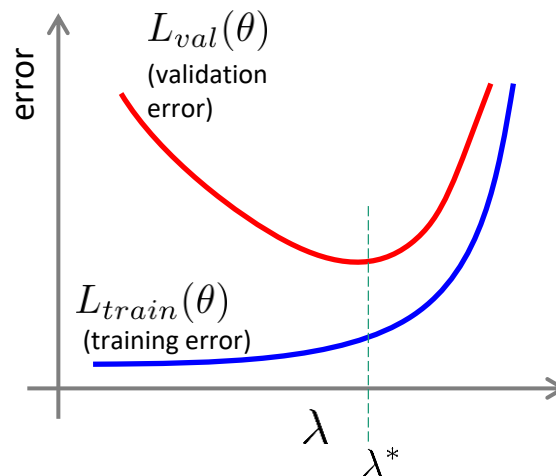
- 如何选择正则化参数 $\lambda$  ?

Model:  $h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

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

$$L(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

1. Try  $\lambda = 0$
2. Try  $\lambda = 0.01$
3. Try  $\lambda = 0.02$
4. Try  $\lambda = 0.04$
5. Try  $\lambda = 0.08$
- $\vdots$
12. Try  $\lambda = 10$



# 正则化线性回归: Normal equation

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

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

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

# 正则化Logistic回归

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

## Gradient descent

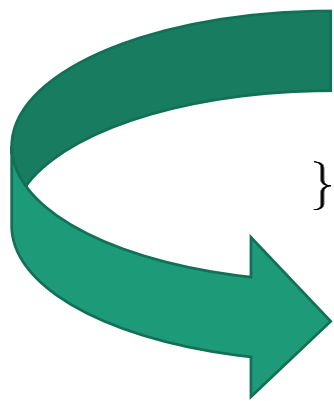
Repeat {

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

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad (j = \text{red X}, 1, 2, 3, \dots, n)$$

}

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

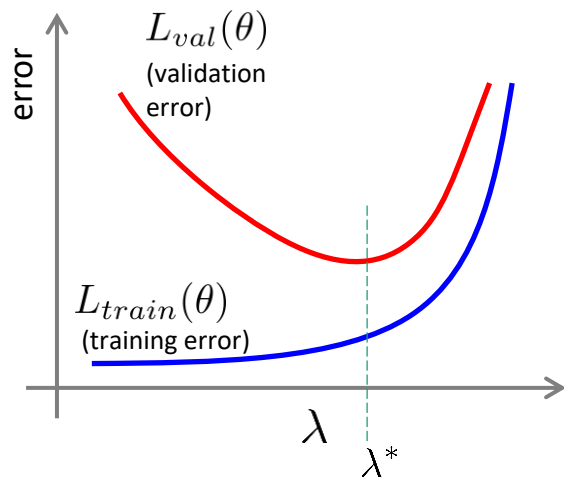


# 正则化Logistic回归

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

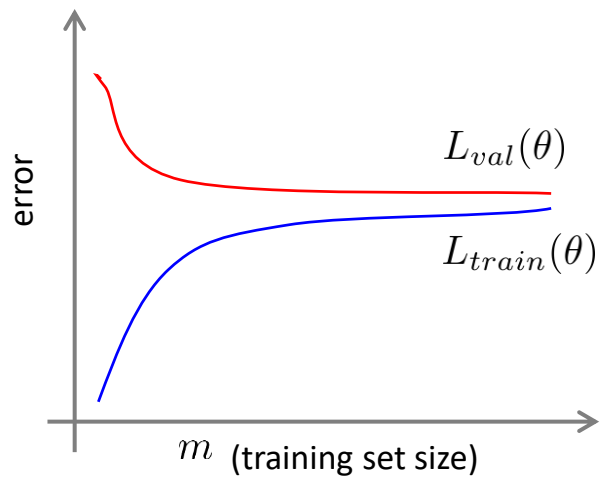
$$J(\theta) = L(\theta) + \lambda R(\theta)$$

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

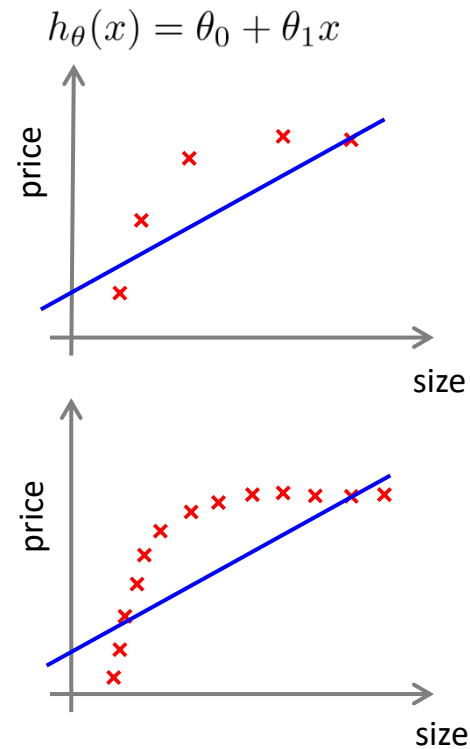


思考：是否可以选择其他的  $L_{val}(\theta)$  ？

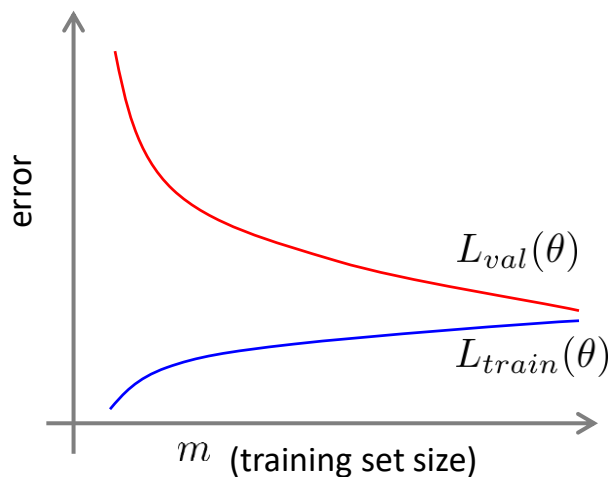
# 学习曲线



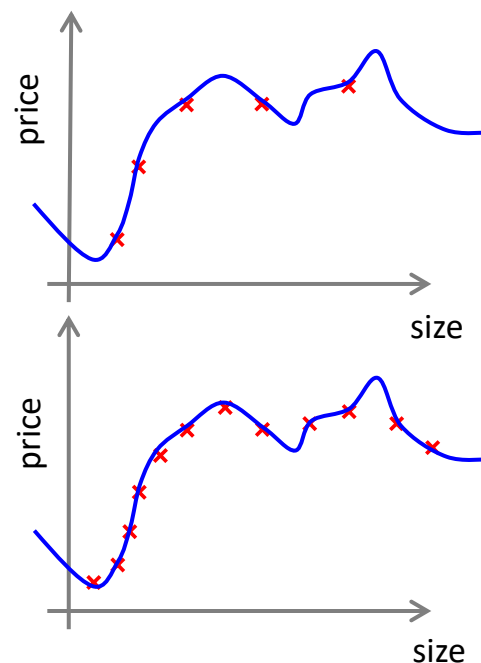
If a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much.



# 学习曲线



If a learning algorithm is suffering from high variance, getting more training data is likely to help.



## 思考：

假设已经训练好了用于预测房价的正则化线性回归模型，但是，当在新的数据上进行测试时出现了很严重预测错误。下一步该怎么做呢？

- 获得更多的训练数据？
- 尝试较小的特征集？
- 尝试其他附加特征？
- 尝试加入多项式组合特征？
- 尝试减少正则化参数 $\lambda$ ？
- 尝试增加正则化参数 $\lambda$ ？



# 模型性能评估

- 我们用训练集优化参数

$$\theta^* = \arg \min_{\theta} \frac{1}{m} \sum_{i=1}^m \ell(h_{\theta}(x^{(i)}), y^{(i)})$$

- 用验证集选择模型
- 但我们真正关心的是模型在新的测试数据上的性能

# 模型性能评估

Dataset

| Size | Price |
|------|-------|
| 2104 | 400   |
| 1600 | 330   |
| 2400 | 369   |
| 1416 | 232   |
| 3000 | 540   |
| 1985 | 300   |
| 1534 | 315   |
| 1427 | 199   |
| 1380 | 212   |
| 1494 | 243   |

随机选取

$(x^{(1)}, y^{(1)})$

$(x^{(2)}, y^{(2)})$

$\vdots$

$(x^{(m)}, y^{(m)})$

Training Set

$(x_{val}^{(1)}, y_{val}^{(1)})$

$(x_{val}^{(2)}, y_{val}^{(2)})$

$\vdots$

$(x_{val}^{(m_{val})}, y_{val}^{(m_{val})})$

Validation Set  
(Development set)

$(x_{test}^{(1)}, y_{test}^{(1)})$

$(x_{test}^{(2)}, y_{test}^{(2)})$

$\vdots$

$(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

Testing Set

# 模型性能评估

- 训练集: 训练参数
- 验证集(开发集, Development set): 用于调参(如正则化参数、多项式阶数等)、特征选择以及 other decisions regarding the learning algorithm
- 测试集: 仅仅用于性能评估, not to make any decisions about regarding what learning algorithm or parameters to use.

# 模型性能评估

- 验证集和测试集的选择：
  - Choose validation and test sets to reflect data you expect to get in the future and want to do well on.
  - 验证集和测试集应具有同分布

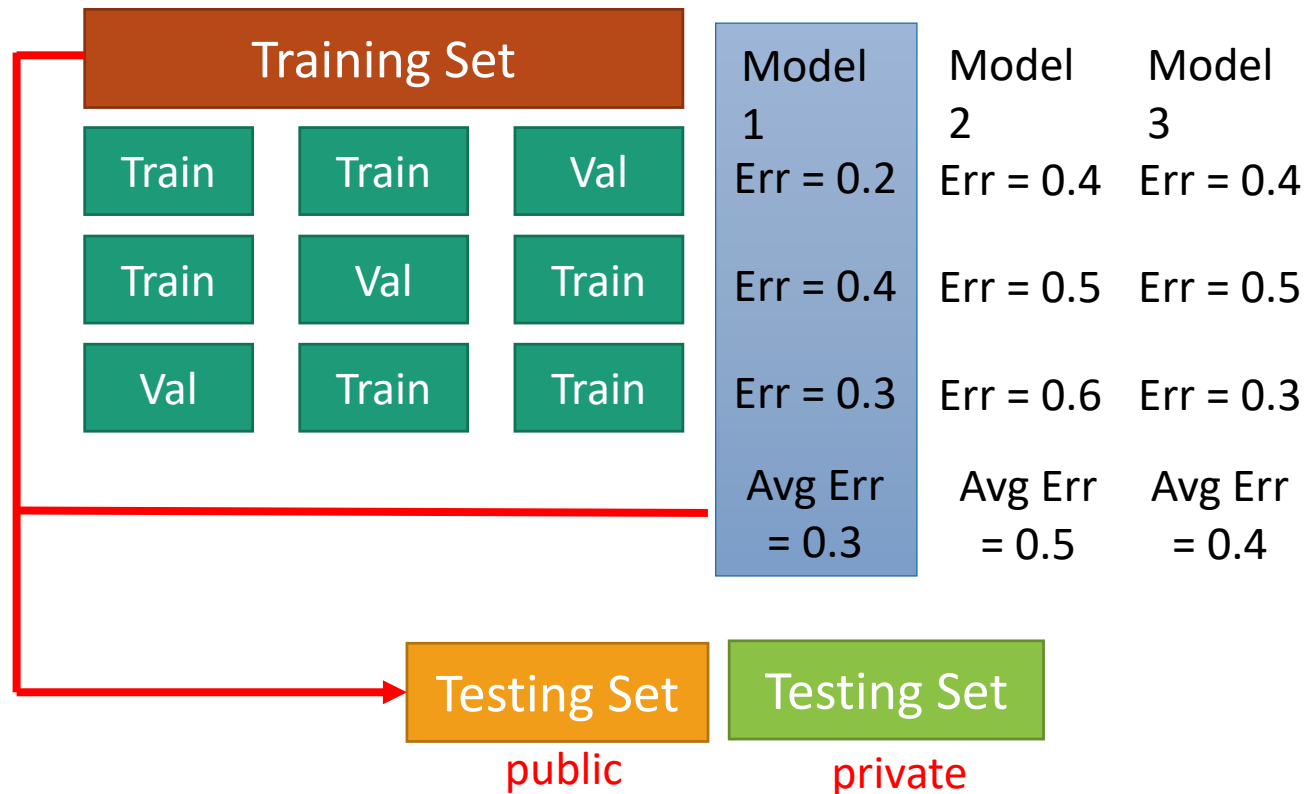
思考：假设验证集和测试集具有同分布，若算法在验证集上效果较好但在测试集上性能很差，下一步该怎么办？

- 验证集和测试集的大小
  - 验证集：1,000 to 10,000 examples are common; Should be large enough
  - 测试集：中小规模数据情况下一般取30%；大数据情况下，large enough
  - No need to have excessively large validation/test beyond what is needed to evaluate the performance of your algorithms

# 模型性能评估

- 交叉验证（ $k$ -fold Cross Validation）：
  - 数据集规模较小情况下采用
  - 把数据随机划分为 $k$ 等份，每次用其中的 $(k - 1)$ 份做训练，剩下的做验证
  - 计算平均误差（和方差）

# $k$ -fold Cross Validation



# Thanks!

Any questions?