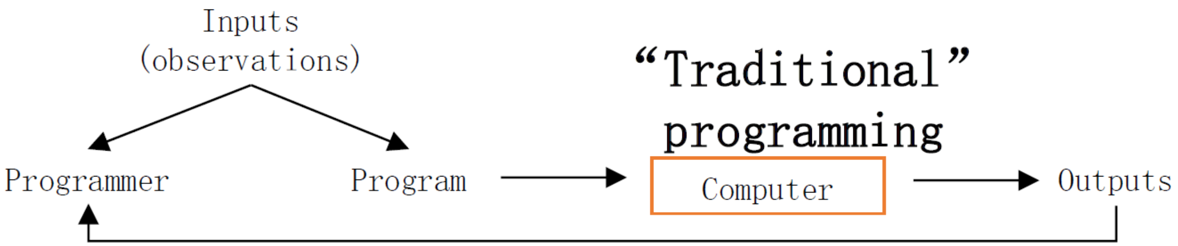
线性回归

01 PART ONE

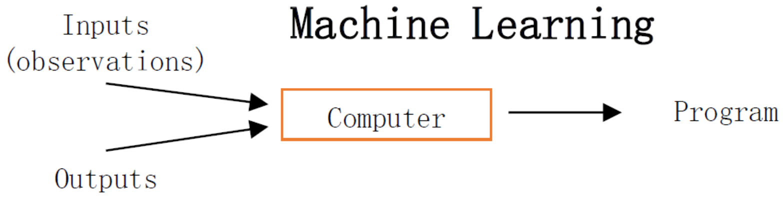
机器学习

什么是机器学习



Field of study that gives computers the ability to learn without being explicitly programmed——Arthur Samuel (1959)

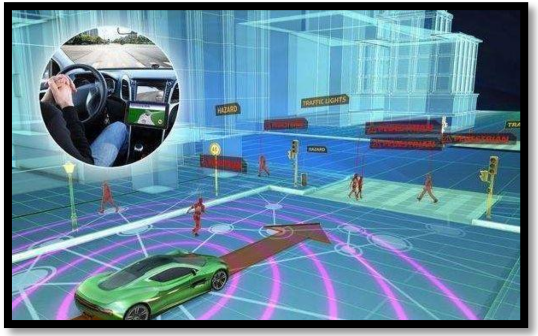
在不直接针对问题进行编程的情况下，赋予计算机学习能力的一个研究领域。



机器学习的应用

推荐系统

自动驾驶



语音识别

垃圾邮件

机器视觉

机器学习的类型

监督学习

- 从有标签的数据中进行学习；
- 在输入和输出之间有着一个特定的关系；
- 例如：垃圾邮件分类。

无监督学习

- 从没有标签的数据中提取一个特殊的结构；
- 例如：文本聚类。

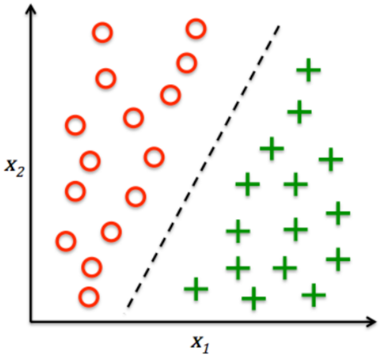
强化学习

- 通过给环境系统一个‘行为’，会有延时的反馈，来进行学习；
- 例如：象棋程序。

监督学习

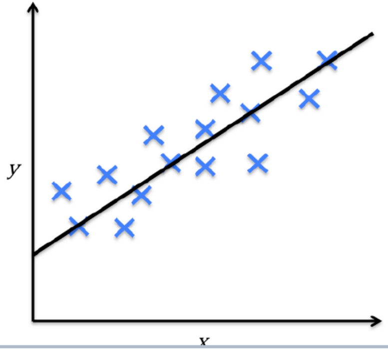
分类问题（监督学习）：

- 根据数据样本上抽取出的特征，判定其属于有限个类别中的哪一个
- 垃圾邮件识别（结果类别：1、垃圾邮件 2、正常邮件）
- 文本情感褒贬分析（结果类别：1、褒 2、贬）
- 图像内容识别识别（结果类别：1、喵星人 2、汪星人 3、人类 4、草泥马 5、都不是）



回归问题（监督学习）：

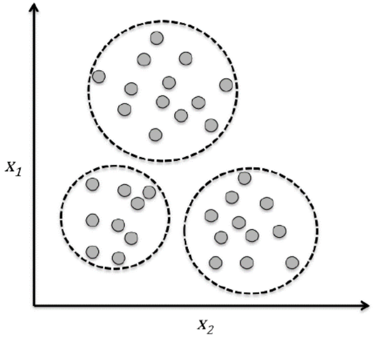
- 根据数据样本上抽取出的特征，预测连续值结果
- 《芳华》票房值
- 魔都房价具体值
- 刘德华和吴彦祖的具体颜值得分



无监督和强化学习

聚类问题(无监督学习)：

- 根据数据样本上抽取出的特征，挖掘数据的关联模式
- 相似用户挖掘/社区发现
- 新闻聚类



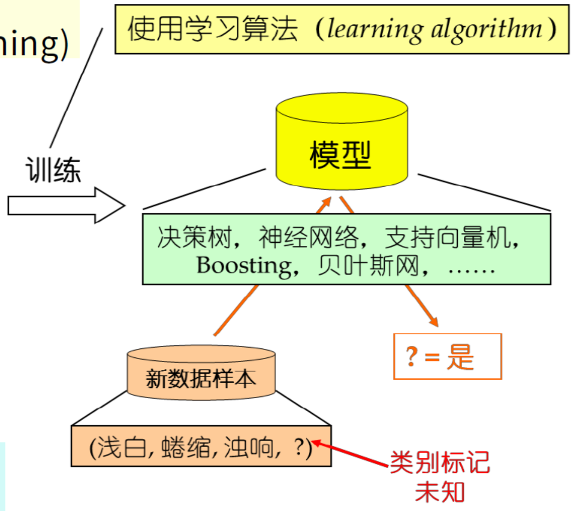
强化问题：

- 研究如何基于环境而行动，以取得最大化的预期利益
- 游戏(“吃鸡”)最高得分
- 机器人完成任务

机器学习模型

· 无监督学习
(unsupervised learning)

· 监督学习
(supervised learning)



色泽	根蒂	敲声	好瓜
青绿	蜷缩	浊响	是
乌黑	蜷缩	浊响	是
青绿	硬挺	清脆	否
乌黑	稍蜷	沉闷	否

· 假设(hypothesis)
· 真相(ground-truth)
· 学习器(learner)

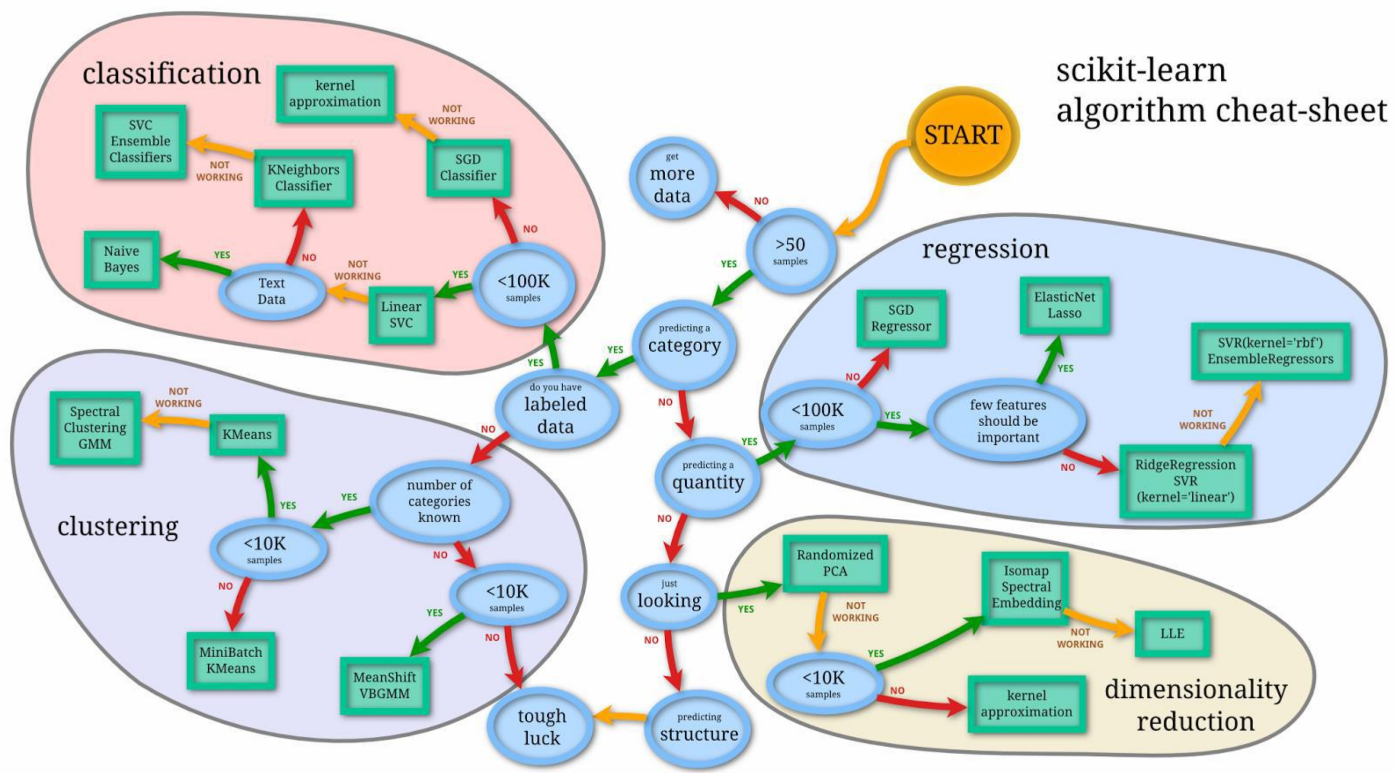
- 分类，回归
- 二分类，多分类
- 正类，反类

- 数据集; 训练, 测试
- 示例(instance)，样例(example)
- 样本(sample)
- 属性(attribute)，特征(feature)；属性值
- 属性空间，样本空间，输入空间
- 特征向量(feature vector)
- 标记空间，输出空间

- 未见样本(unseen instance)
- 未知“分布”
- 独立同分布(i.i.d.)
- 泛化(generalization)

案例 From 周志华《机器学习》

scikit-learn



程序过程

训练集

测试集

```
train_x, train_y, test_x, test_y = getData()

model = somemodel()
model.fit(train_x,train_y)
predictions = model.predict(test_x)

score = score_function(test_y, predictions)
```

模型实例化

模型训练

模型预测

模型评估

02

PART TWO

线性回归

基本概念

- 给定训练数据$\mathcal{D}=\{\mathbf{x}_i, y_i\}_{i=1}^{N}$，其中 $y \in \mathbb{R}$，回归学习一个从输入$\mathbf{x}$到输出y的映射f
- 对新的测试数据$\mathbf{x}$，用学习到的映射对其进行预测：$\hat{y}=f(\mathbf{x})$

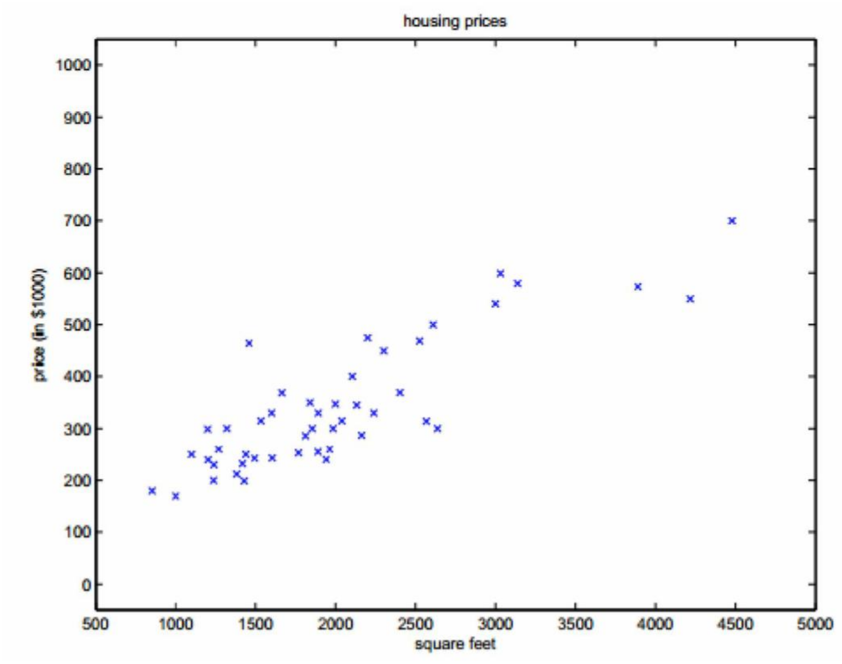
- 若假设映射f是一个线性函数，即

$$y=f(\mathbf{x}|\boldsymbol{\theta})=\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}$$

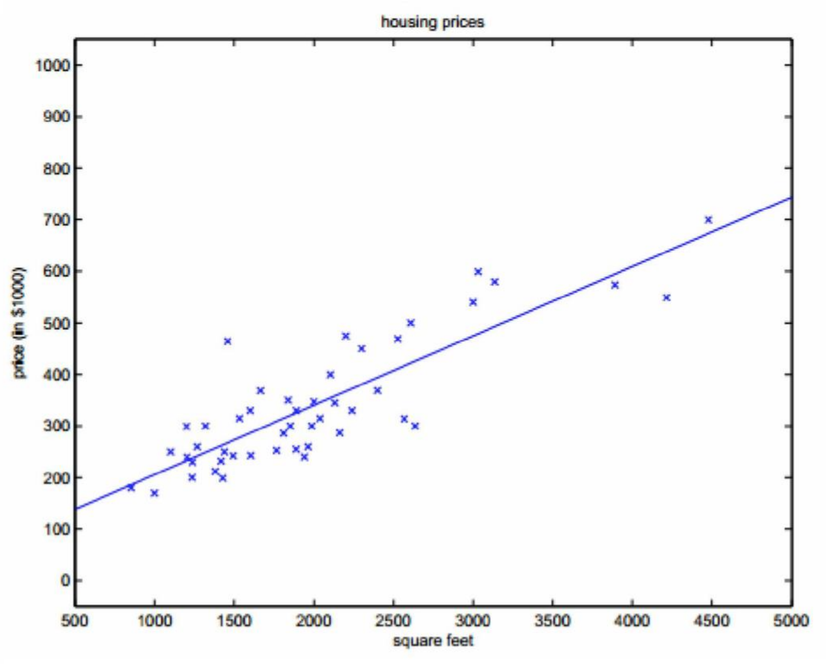
- 我们称之为线性回归模型。

线性回归模型

- y=ax+b



housing prices
price (in $1000)
square feet
1000
900
800
700
600
500
400
300
200
100
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000



housing prices
price (in $1000)
square feet
1000
900
800
700
600
500
400
300
200
100
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

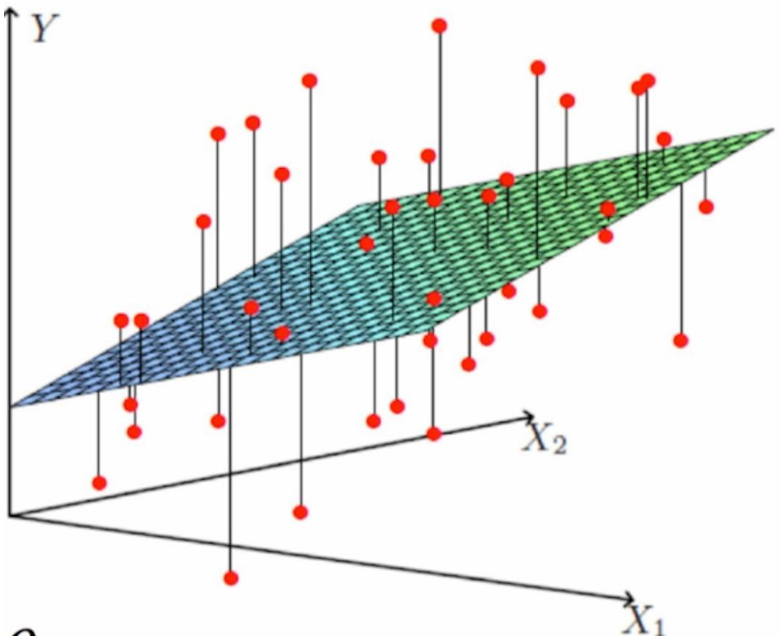
线性回归模型

□ 考虑两个变量

Living area ($feet^2$)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

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

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$



损失函数

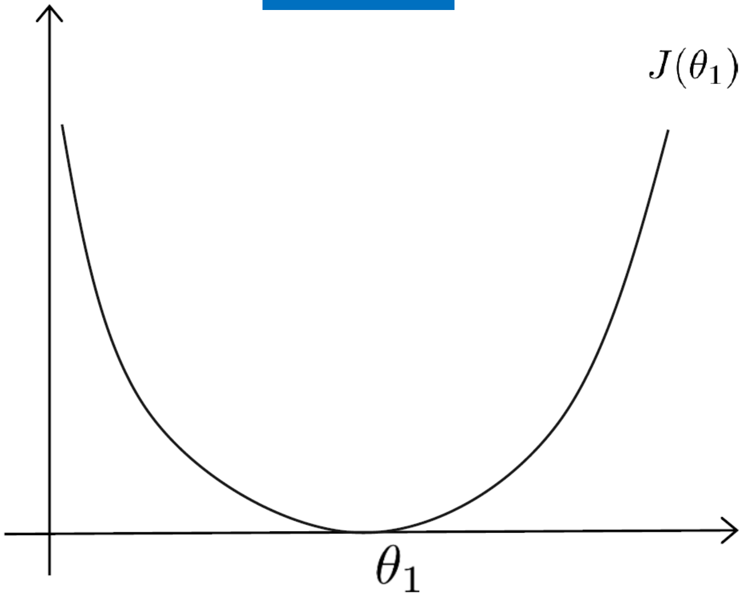
- 损失函数是为了量化了模型预测值与实际观察值之间的误差大小。
- 有了损失函数就可以评价取当前参数时模型性能的好坏。
- 对于线性回归算法，比较常用的损失函数是均方误差(Mean Square Error, MSE)函数

$$J(\theta) = MSE(X, \theta) = \frac{1}{2n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 = \frac{1}{2n}\sum_{i=1}^{n}(\theta_i \cdot x_i - y_i)^2$$

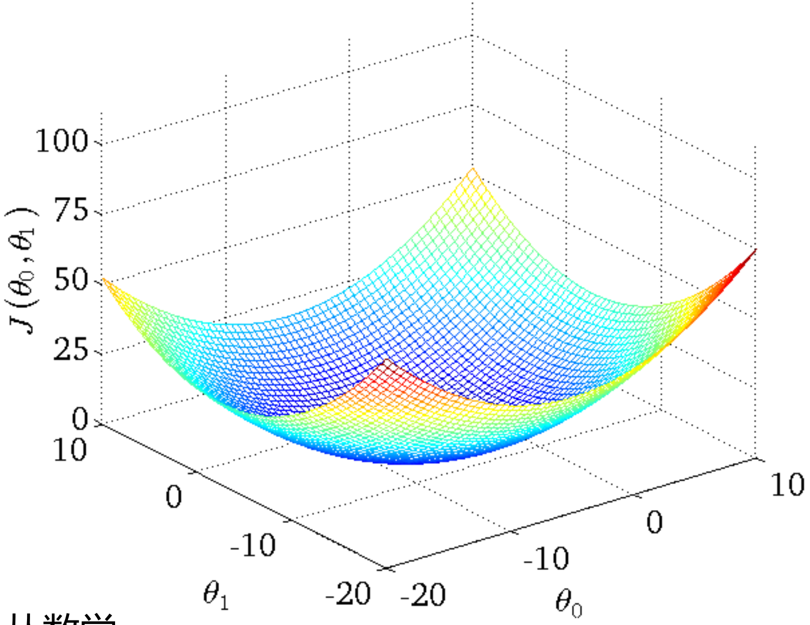
损失函数

损失函数是一个凸函数

单变量



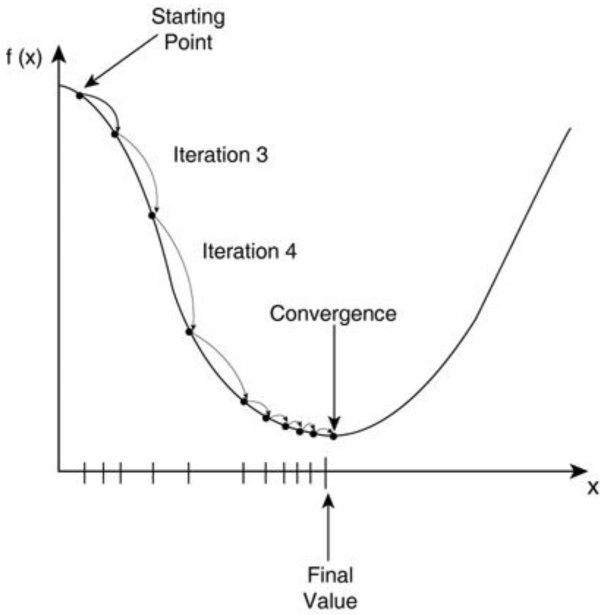
多变量



如果代价函数是一个凸函数（convex function），那么从数学上可以保证肯定能求得全局最优解

梯度下降法

- 逐步最小化损失函数的过程
- 如同下山，找准方向(斜率)，每次迈进一小步，直至山底。



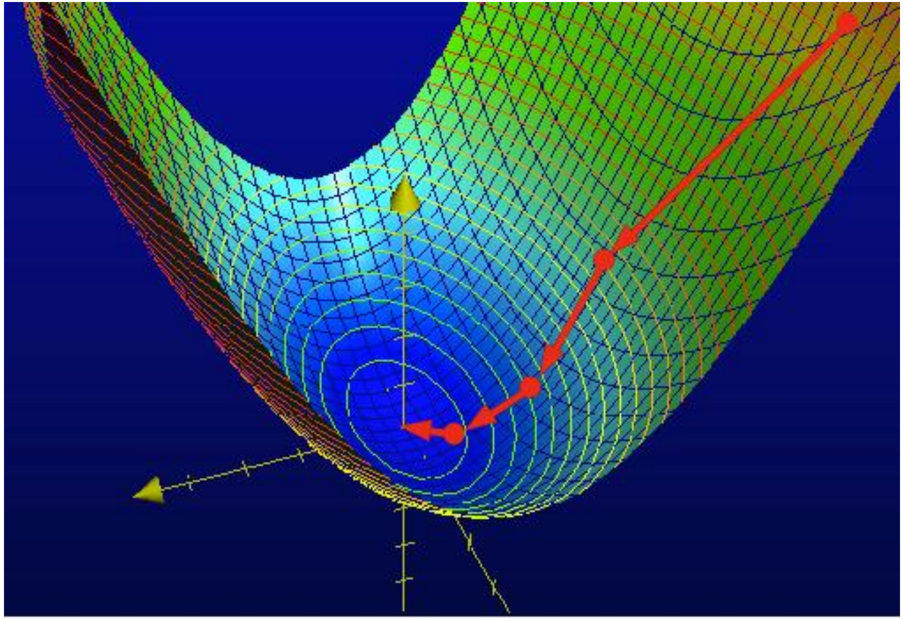
$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

学习率

- 过大的学习率会导致梯度下降时越过代价函数的最小值点；
- 如果学习率太小，训练中的每一步参数的变化会非常小；
- 一般建议，每次可以3倍放大或者3倍缩小来调整。

梯度下降求参数

- 两个变量



$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\Downarrow$$

repeat until convergence {

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

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

}

梯度下降法

- 多个变量

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\
&\}
\end{aligned}
$$

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

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

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

$$\ldots$$

判定系数

在线性回归中，可以把总误差平方和分解成两部分：回归平方和，以及残差平方和：

$$SST = SSR + SSE$$

- $SST = \sum_{i=1}^{n}(y_i - \bar{y})^2$
- $SSR = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$
- $SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

判定系数

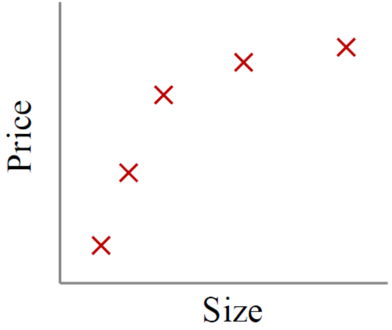
$$R^2 = \frac{SSR}{SST}$$

03

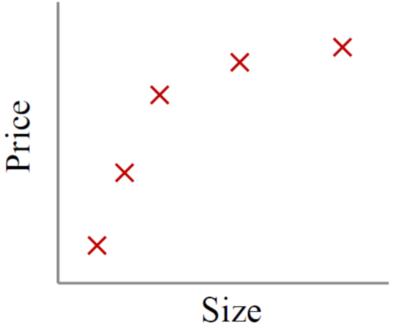
PART THREE

正则化

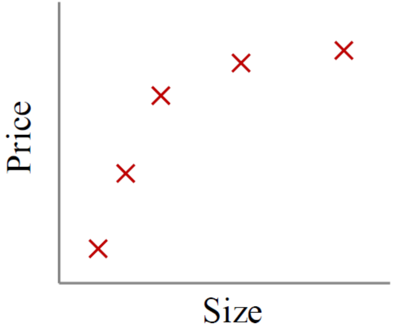
过拟合和欠拟合



$\theta_0 + \theta_1 x$



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



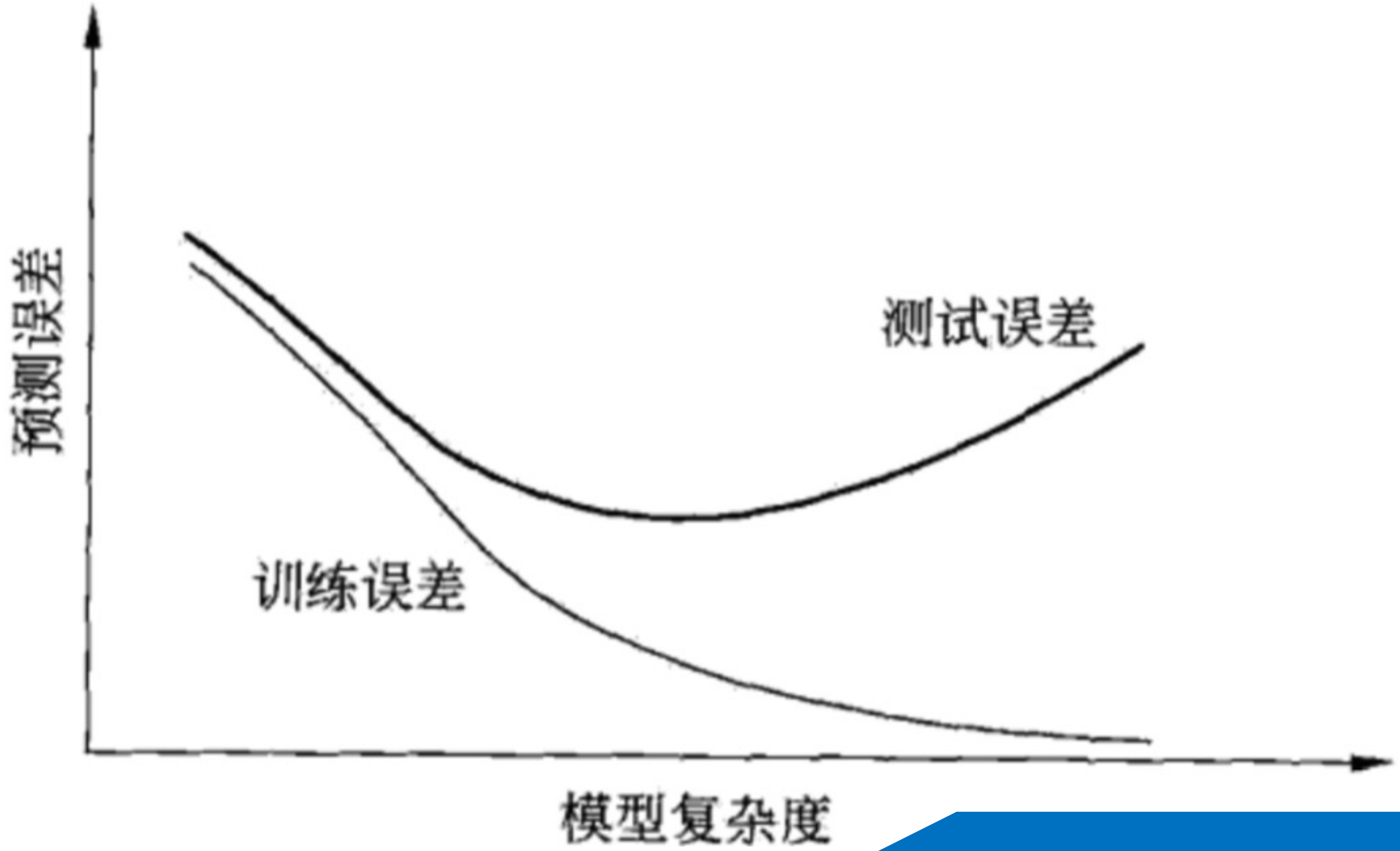
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

过拟合问题：如果我们有非常多特征/模型很复杂，我们的假设函数曲线可以对原始数据拟合得非常好（$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \approx 0$），但丧失了一般性，从而导致对新给的待预测样本，预测效果差.

所有的模型都可能存在过拟合的风险：

- 更多的参数，更复杂的模型，意味着有更强的能力，但也更可能无法无天
- 眼见不一定为实，你看到的内容不一定是全部真实的数据分布，死记硬背不太好

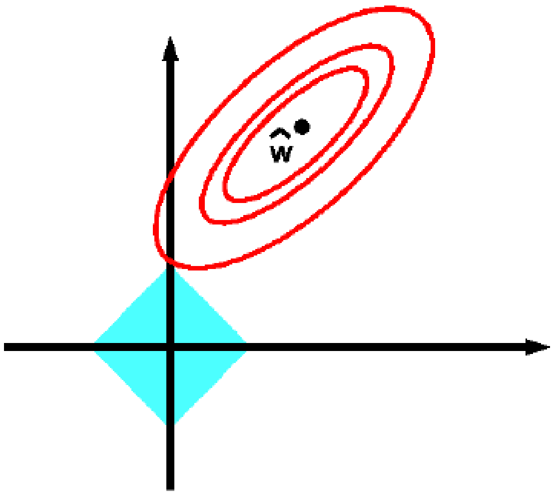
过拟合和欠拟合



正则化

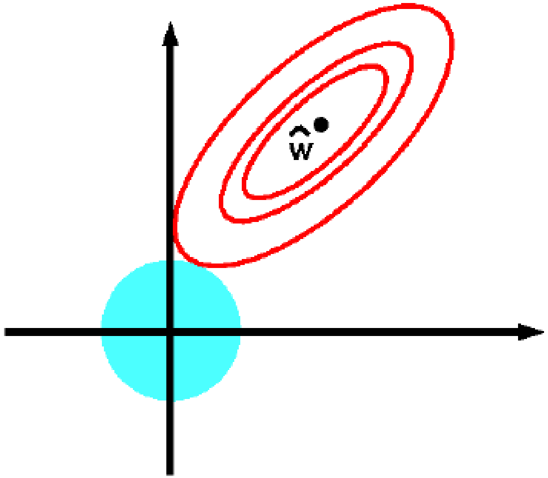
Lasso回归

$$J_L(w) = \frac{1}{2}\|y - Xw\|^2 + \lambda \sum |w_i|$$



岭回归

$$J_R(w) = \frac{1}{2}\|y - Xw\|^2 + \frac{1}{2}\lambda\|w\|^2$$



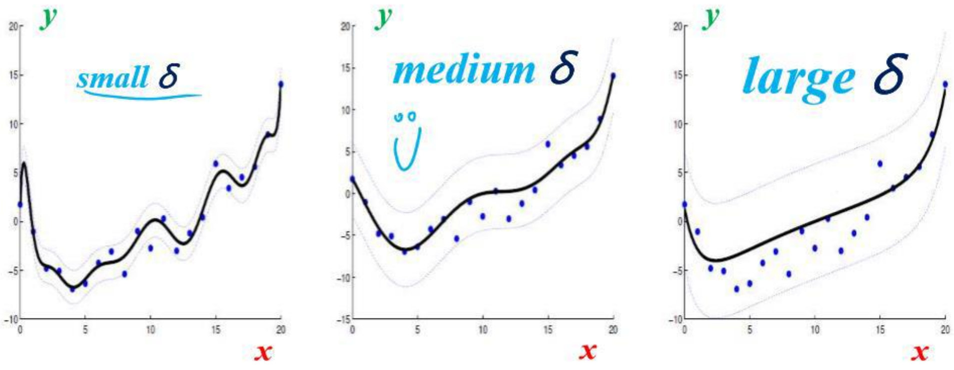
正则化项的超参数

***Example*: Ridge regression with a polynomial of degree 14**

$$\hat{y}(x_i) = 1\,\theta_0 + x_i\,\theta_1 + x_i^2\,\cancel{\theta_2}^{\,0} + \dots + x_i^{13}\,\theta_{13} + x_i^{14}\,\theta_{14}\,\theta$$

$$\Phi = [\,1 \;\; x_i \;\; x_i^2 \;\; \dots \;\; x_i^{13} \;\; x_i^{14}\,]\; x_i^{15} \dots$$

$$J(\theta) = (y - \Phi\theta)^T (y - \Phi\theta) + \underbrace{\delta\,\theta^T\theta}$$



04

PART FOUR

Scikit-learn的实现

Scikitlearn中的linear_model实现

http://sklearn.apachecn.org/cn/0.19.0/modules/linear_model.html#id10

05

PART FIVE

项目实战

案列分析

波士顿房价分析