开课吧
kaikeba.com
慧科集团旗下企业

# 线性回归

机器学习算法之二

- 1 线性回归
- 2 岭回归
- 3 LASSO回归

# 线性回归

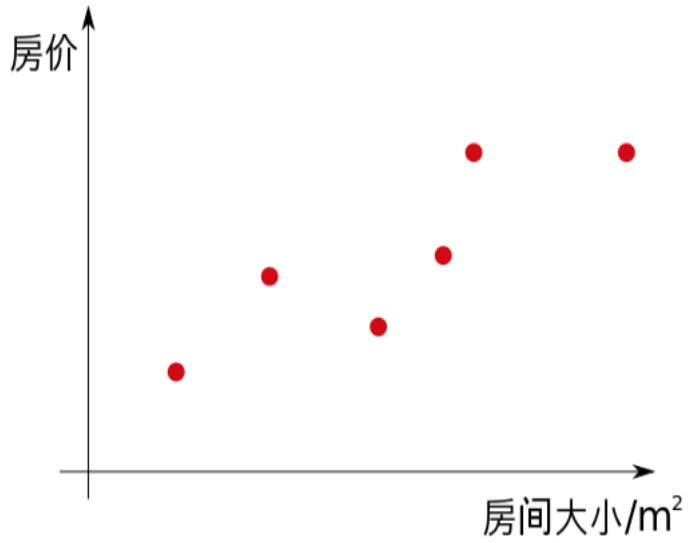
- 回归（吴恩达-房价预测实例）

| Living area (feet$^2$) | Price (1000$s) |
|---|---|
| 2104 | 400 |
| 1600 | 330 |
| 2400 | 369 |
| 1416 | 232 |
| 3000 | 540 |
| ⋮ | ⋮ |

假设我们有某个地区住房面积和相应房价的数据集合。对于这样的给定的数据， 我们的目的是要利用已有的信息，来对房价建立预测模型。即对于给定的房屋信息(房屋面积)预测其房价。

# 回归

- 回归分析常用于分析两个变量X和Y 之间的关系。




| id | date | price | bedrooms | bathrooms | s |
|---|---|---|---|---|---|
| 7129300520 | 20141013T000000 | 221900.0 | 3 | 1.00 | 1 |
| 6414100192 | 20141209T000000 | 538000.0 | 3 | 2.25 | 2 |
| 5631500400 | 20150225T000000 | 180000.0 | 2 | 1.00 | 7 |
| 2487200875 | 20141209T000000 | 604000.0 | 4 | 3.00 | 1 |
| 1954400510 | 20150218T000000 | 510000.0 | 3 | 2.00 | 1 |

# 回归

- 二元线性回归
  - 1.代数方法
  - 2.线性几何
  - 3. 梯度下降

- 代数表示

上面是求最值的问题，我们会想到导数和偏导数，这里在偏导数等于0的地方能取到极值，并且也是最值。
分别对a和b求偏导得到如下表达式：

$$\frac{\partial}{\partial a} = \sum_{i=1}^{n} 2x_i(ax_i + b - y_i)$$
$$\frac{\partial}{\partial b} = \sum_{i=1}^{n} 2(ax_i + b - y_i)$$

通过对二元一次方程组

$$\sum_{i=1}^{n} 2x_i(ax_i + b - y_i) = 0$$
$$\sum_{i=1}^{n} 2(ax_i + b - y_i) = 0$$

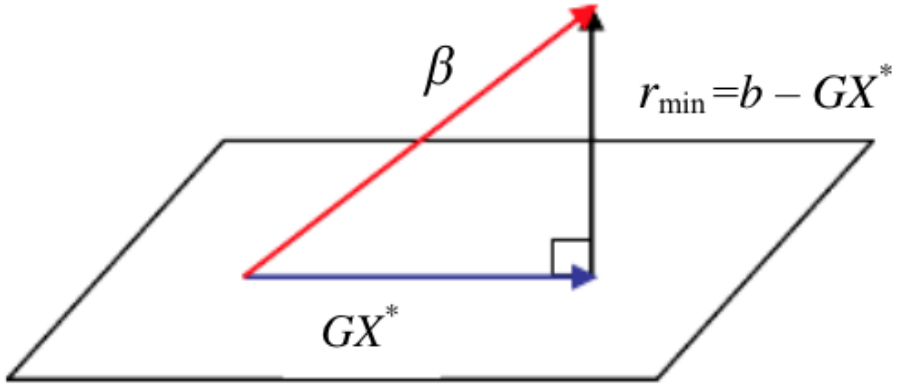
进行求解，可以得到如下解：

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$
$$b = \frac{\sum_{i=1}^{n} x_i^2 \sum y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

- 线性几何

$$G^T(b - GX^*) = 0$$

- 1 X* 为平面上向量
- 2 法向量r
- 3 列向量正交
- 4 求出 $X=(G^TG)^{-1}G^Tb$



- 梯度下降:
  - 确定损失函数

迭代公式： $\theta^t = \theta^{t-1} + \Delta\theta$

将 $L(\theta^t)$ 在 $\theta^{t-1}$ 处进行一阶泰勒展开：

$$L(\theta^t) = L(\theta^{t-1} + \Delta\theta)$$
$$\approx L(\theta^{t-1}) + L'(\theta^{t-1})\Delta\theta$$

要使得 $L(\theta^t) < L(\theta^{t-1})$ ， 可取： $\Delta\theta = -\alpha L'(\theta^{t-1})$ ， 则： $\theta^t = \theta^{t-1} - \alpha L'(\theta^{t-1})$

# 线性回归

考虑多特征，一般形式：

## 线性回归的一般形式

$$h(\theta) = \sum_{j=0}^{n} \theta_j x_j$$

$$J(\theta) = \frac{1}{2m} \sum_{i=0}^{m} (y^i - h_\theta(x^i))^2$$

**h(x)**是需要拟合的函数。

**J(θ)**称为均方误差或cost function。用来衡量训练集众的样本对线性模式的拟合程度。

**m**为训练集众样本的个数。

**θ**是我们最终需要通过梯度下降法来求得的参数。

习惯的把第一项$x_0$写为1

# 线性回归的一般形式

$$h(\theta) = \sum_{j=0}^{n} \theta_j x_j$$

$$J(\theta) = \frac{1}{2m} \sum_{i=0}^{m} (y^i - h_\theta(x^i))^2$$

$$\left[ (X\theta - Y)^T (X\theta - Y) \right]$$

$$x^{(i)} = \begin{pmatrix} x_0^{(i)} \\ x_1^{(i)} \\ \vdots \\ x_n^{(i)} \end{pmatrix} \quad \theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{pmatrix} \quad X = \begin{pmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(m)})^T \end{pmatrix} \quad Y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

$$X\theta - Y = \begin{pmatrix} (x^{(1)})^T\theta - y^{(1)} \\ (x^{(2)})^T\theta - y^{(2)} \\ \vdots \\ (x^{(m)})^T\theta - y^{(m)} \end{pmatrix} = \begin{pmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ h_\theta(x^{(2)}) - y^{(2)} \\ \vdots \\ h_\theta(x^{(m)}) - y^{(m)} \end{pmatrix}$$

$$\nabla_\theta J(\theta)=\nabla_\theta\left(\frac{1}{2}(X\theta-y)^T(X\theta-y)\right)=\nabla_\theta\left(\frac{1}{2}\left(\theta^T X^T-y^T\right)(X\theta-y)\right)$$

$$=\nabla_\theta\left(\frac{1}{2}\left(\theta^T X^T X\theta-\theta^T X^T y-y^T X\theta+y^T y\right)\right)$$

$$=\frac{1}{2}\left(2X^T X\theta-X^T y-\left(y^T X\right)^T\right)=X^T X\theta-X^T y\xrightarrow{\text{求驻点}}0$$

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

约定布局为列形式

$$\vec{x}=\begin{pmatrix}x_1\\x_2\\\vdots\\x_n\end{pmatrix}\qquad A\cdot\vec{x}=\begin{pmatrix}a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n\\a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n\\\vdots\\a_{m1}x_1+a_{m2}x_2+\cdots+a_{mn}x_n\end{pmatrix}$$

$$\frac{\partial A\vec{x}}{\partial\vec{x}}=\begin{bmatrix}a_{11}&a_{21}&\cdots&a_{m1}\\a_{12}&a_{22}&\cdots&a_{m2}\\\vdots&\vdots&\ddots&\vdots\\a_{1n}&a_{2n}&\cdots&a_{mn}\end{bmatrix}$$

# 矩阵推导

$$\frac{\partial\left(\vec{x}^T \cdot A \cdot \vec{x}\right)}{\partial \vec{x}} \longrightarrow \left(A^T + A\right) \cdot \vec{x}$$

$$\vec{x}^T A \cdot \vec{x} \longrightarrow \sum_{j=1}^{n} \sum_{i=1}^{n} a_{ij} x_i x_j$$

简单起见，取2x2矩阵，课后手动推导一下：

# 矩阵推导

$$\nabla_\theta J(\theta)=\nabla_\theta\left(\frac{1}{2}(X\theta-y)^T(X\theta-y)\right)=\nabla_\theta\left(\frac{1}{2}\left(\theta^T X^T-y^T\right)(X\theta-y)\right)$$

$$=\nabla_\theta\left(\frac{1}{2}\left(\theta^T X^T X\theta-\theta^T X^T y-y^T X\theta+y^T y\right)\right)$$

$$=\frac{1}{2}\left(2X^T X\theta-X^T y-\left(y^T X\right)^T\right)=X^T X\theta-X^T y\xrightarrow{\text{求驻点}}0$$

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

- 梯度下降

**线性回归的一般形式**

$$h(\theta) = \sum_{j=0}^{n} \theta_j x_j$$

$$J(\theta) = \frac{1}{2m} \sum_{i=0}^{m} (y^i - h_\theta(x^i))^2$$

步骤：

1. 求梯度
2. 负梯度更新
3. 迭代收敛

- 梯度下降

**线性回归的一般形式**

$$h(\theta) = \sum_{j=0}^{n} \theta_j x_j$$

$$J(\theta) = \frac{1}{2m} \sum_{i=0}^{m} (y^i - h_\theta(x^i))^2$$

$$\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta_j} &= \frac{1}{m} \sum_{i=0}^{m} (y^i - h_\theta(x^i)) \frac{\partial}{\partial \theta_j} (y^i - h_\theta(x^i)) \\
&= -\frac{1}{m} \sum_{i=0}^{m} (y^i - h_\theta(x^i)) \frac{\partial}{\partial \theta_j} (\sum_{j=0}^{n} \theta_j x_j^i - y^i) \\
&= -\frac{1}{m} \sum_{i=0}^{m} (y^i - h_\theta(x^i)) x_j^i
\end{aligned}$$

批量梯度 $$\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j} := \theta_j + \alpha \frac{1}{m} \sum_{i=0}^{m} (y^i - h_\theta(x^i)) x_j^i$$

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

# 岭回归

## 1. 病态系统

现在有线性系统： Ax = b， 解方程

$$\begin{bmatrix} 400 & -201 \\ -800 & 401 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=\begin{bmatrix} 200 \\ -200 \end{bmatrix}$$

很容易得到解为： **x1 = -100, x2 = -200**. 如果在样本采集时存在一个微小的误差，比如，将 A 矩阵的系数 400 改变成 401：

$$\begin{bmatrix} 401 & -201 \\ -800 & 401 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=\begin{bmatrix} 200 \\ -200 \end{bmatrix}$$

则得到一个截然不同的解： **x1 = 40000, x2 = 79800**.

当解集 x 对 A 和 b 的系数高度敏感，那么这样的方程组就是病态的 (ill-conditioned).

$$\begin{bmatrix} 400+10 & -201 \\ 800 & 400+10 \end{bmatrix}$$

解为：

$x_1 = 5.726$

$x_2 = 10.685$

$$\begin{bmatrix} 400+10 & -201 \\ 801 & 400+10 \end{bmatrix}$$

解为：

$x_1 = 5.888$

$x_2 = 11.016$

线性系统 Ax = b 为什么会病态？归根到底是由于 A 矩阵列向量线性相关性过大

- 条件数

当 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ 具有特征值分解时，其条件数为

$$\max_{i,j} \left| \frac{\lambda_i}{\lambda_j} \right|.$$

一般条件数大于10，可能存在病态情况。

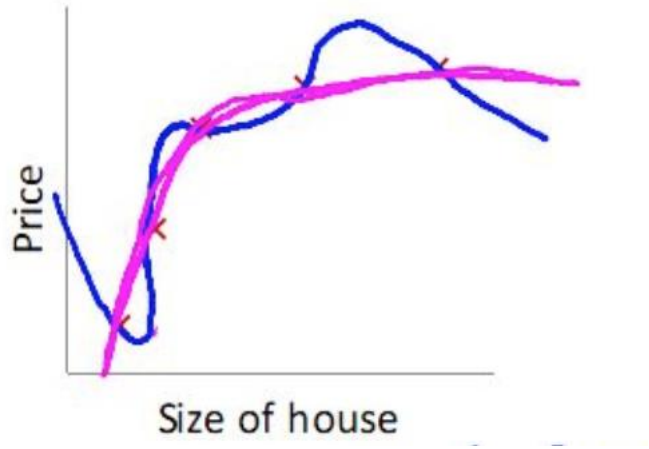
- 因此，对岭回归解法写做：

$$(X^T X + \lambda I)^{-1} X^T y$$

- 即把$X^TX$近半正定矩阵，用正定矩阵替代。

- 正则化的理解：

  - 1.方程本身的理解




$$\nabla_\theta J(\theta)=\nabla_\theta\left(\frac{1}{2}(X\theta-y)^T(X\theta-y)\right)=\nabla_\theta\left(\frac{1}{2}(\theta^T X^T-y^T)(X\theta-y)\right)+\lambda\theta^2$$

要求参数不能差异过大，加入模型复杂度的约束。

2. 线性代数方法理解

$$\nabla_\theta J(\theta)=\nabla_\theta\left(\frac{1}{2}(X\theta-y)^T(X\theta-y)\right)=\nabla_\theta\left(\frac{1}{2}(\theta^T X^T-y^T)(X\theta-y)\right) \qquad +\lambda\theta^2$$

$$=\nabla_\theta\left(\frac{1}{2}(\theta^T X^T X\theta-\theta^T X^T y-y^T X\theta+y^T y)\right) \qquad +\lambda\theta^2$$

$$=\frac{1}{2}\left(2X^T X\theta-X^T y-(y^T X)^T\right)=X^T X\theta-X^T y \xrightarrow{\text{求驻点}} 0 \qquad (X^T X+\lambda I)^{-1}X^T y$$

岭回归就是L2正则化

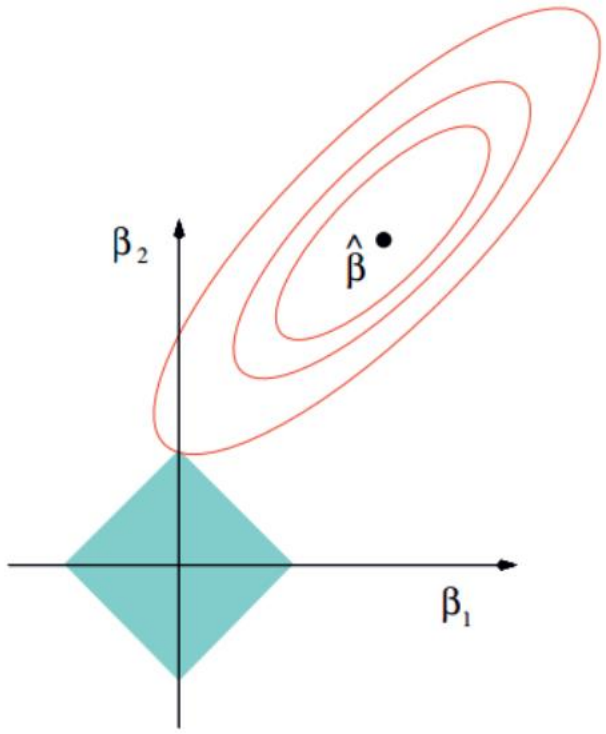
# LASSO

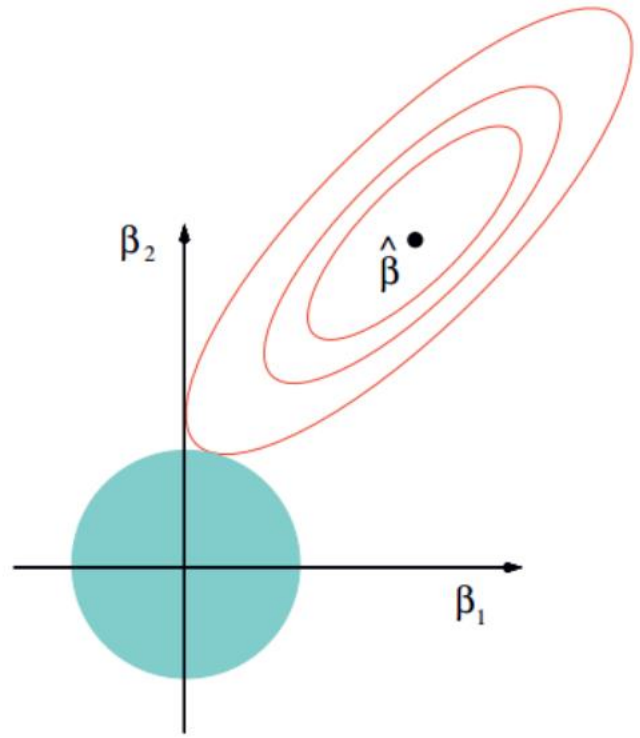
$$\nabla_\theta J(\theta) = \nabla_\theta \left( \frac{1}{2} (X\theta - y)^T (X\theta - y) \right) = \nabla_\theta \left( \frac{1}{2} (\theta^T X^T - y^T)(X\theta - y) \right) \quad +\lambda||\theta||$$

LASSO兴起是在L2正则化之后：

1. 没有碰到过多高维数据
2. 对LASSO的理解不够

- LASSO具有特征选择的作用




L1正则化是指权值向量w中各个元素的绝对值之和，对应图的方块

$$\nabla_\theta J(\theta)=\nabla_\theta\left(\frac{1}{2}(X\theta-y)^T(X\theta-y)\right)=\nabla_\theta\left(\frac{1}{2}(\theta^T X^T-y^T)(X\theta-y)\right) \qquad +\lambda||\theta||$$

L2正则化是指权值向量w中各个元素的平方和然后再求平方根，对应图的圆

$$\nabla_\theta J(\theta)=\nabla_\theta\left(\frac{1}{2}(X\theta-y)^T(X\theta-y)\right)=\nabla_\theta\left(\frac{1}{2}(\theta^T X^T-y^T)(X\theta-y)\right) \qquad +\lambda\theta^2$$

LASSO $$\nabla_\theta J(\theta)=\nabla_\theta\left(\frac{1}{2}(X\theta-y)^T(X\theta-y)\right)=\nabla_\theta\left(\frac{1}{2}(\theta^T X^T-y^T)(X\theta-y)\right) \quad +\lambda||\theta||$$

Ridge $$\nabla_\theta J(\theta)=\nabla_\theta\left(\frac{1}{2}(X\theta-y)^T(X\theta-y)\right)=\nabla_\theta\left(\frac{1}{2}(\theta^T X^T-y^T)(X\theta-y)\right) \quad +\lambda\theta^2$$

ElasticNet $$\nabla_\theta J(\theta)=\nabla_\theta\left(\frac{1}{2}(X\theta-y)^T(X\theta-y)\right)=\nabla_\theta\left(\frac{1}{2}(\theta^T X^T-y^T)(X\theta-y)\right) \quad +\lambda\theta^2 \quad +\lambda||\theta||$$

# 极大似然与均方误差

记样本为$(x^{(i)}, y^{(i)})$，对样本的预测为$\hat{y}^{(i)}|_{\theta}$该记法表示该预测依赖于参数$\theta$的选取。我们有：

$$y = \hat{y}|_{\theta} + \epsilon$$

其中，$\epsilon$是一个误差函数，我们通常认为其服从正态分布即

$$\epsilon \sim N(0, \sigma^2)$$

因此有

$$y - \hat{y}|_{\theta} \sim N(0, \sigma^2)$$
$$y \sim N(\hat{y}|_{\theta}, \sigma^2)$$

# 极大似然与均方误差

$$
\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2
\end{aligned}
$$

开课吧 kaikeba.com 慧科集团旗下企业

# 逻辑回归

- 1 分类问题
- 2 sigmod函数
- 3 理论推导
- 4 softmax回归

# logistic 回归是什么

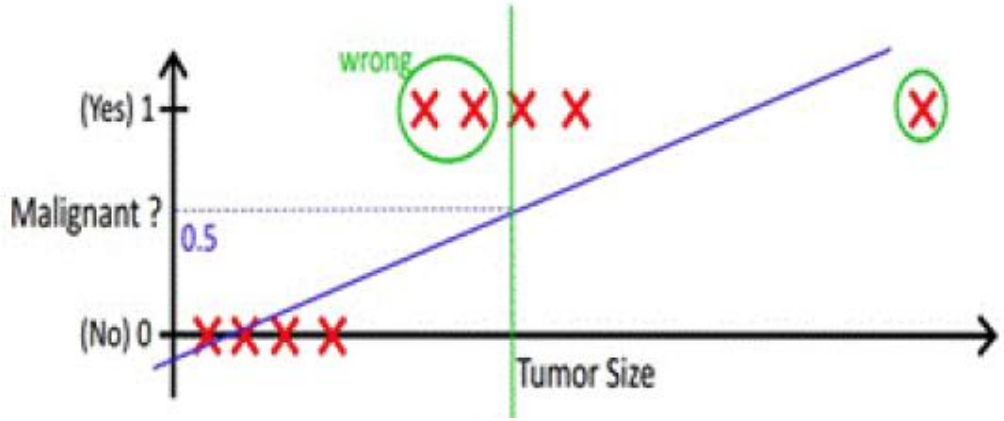
- 工业中的分类问题更多。
- 用户分级
- 是否放款
- 是否点击
- 是否配对
- 是否推荐

分类？

or

回归？

# 回归作为分类可行么？




| | |
|---|---|
| 用户 | 购买频次，浏览频次，时间，地理位置 ... |
| 品类 | 销量，购买用户，浏览用户 ... |
| 交叉 | 购买频次，浏览频次，购买间隔 ... |

$\eta = \log(\Phi/(1-\Phi))$

参数Φ指的是y=1的概率，即事件发生的概率

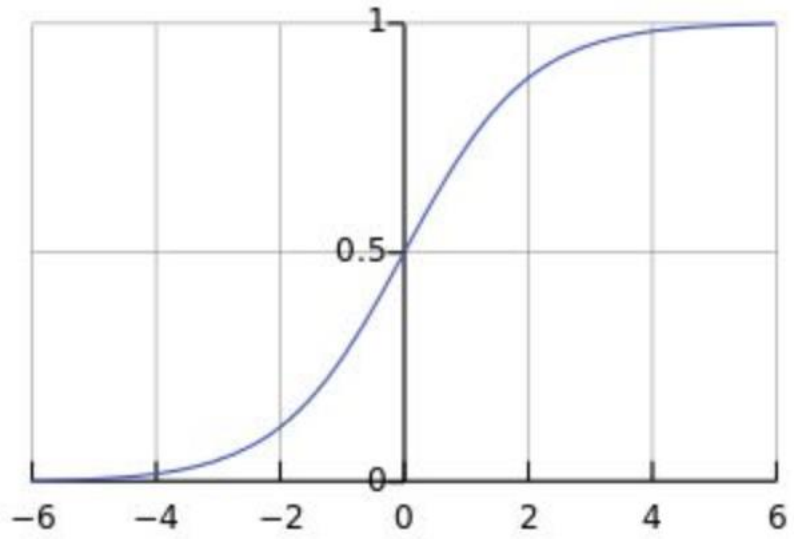
$\eta = \theta^T x$

联立两个公式：

$$\phi = \frac{1}{1+e^{-\theta^T x}}$$

这就是有名的sigmoid，也叫**Logistic函数**

- Sigmod函数求导：




$$
\begin{aligned}
f'(z) &= \left(\frac{1}{1+e^{-z}}\right)' \\
&= \frac{e^{-z}}{(1+e^{-z})^2} \\
&= \frac{1+e^{-z}-1}{(1+e^{-z})^2} \\
&= \frac{1}{(1+e^{-z})}\left(1-\frac{1}{(1+e^{-z})}\right) \\
&= f(z)(1-f(z))
\end{aligned}
$$

函数：$f(z)=\dfrac{1}{1+e^{-z}}$

极大似然的思想：

$$
\begin{aligned}
L(\theta) &= p(\vec{y} \mid X; \theta) \\
&= \prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)}; \theta) \\
&= \prod_{i=1}^{m} \left(h_\theta(x^{(i)})\right)^{y^{(i)}} \left(1 - h_\theta(x^{(i)})\right)^{1-y^{(i)}}
\end{aligned}
$$

$$
\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))
\end{aligned}
$$

$$y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))$$

选择其中一个样本：　　其中　　$h_\theta(x) = g(z)$　　$z = \theta^T x$　　$g(z) = \frac{1}{1+e^{-z}}$

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} \ell(\theta) &= \left( y \frac{1}{g(\theta^T x)} - (1-y) \frac{1}{1 - g(\theta^T x)} \right) \frac{\partial}{\partial \theta_j} g(\theta^T x) \\
&= \left( y \frac{1}{g(\theta^T x)} - (1-y) \frac{1}{1 - g(\theta^T x)} \right) g(\theta^T x)(1 - g(\theta^T x)) \frac{\partial}{\partial \theta_j} \theta^T x \\
&= \left( y(1 - g(\theta^T x)) - (1-y) g(\theta^T x) \right) x_j \\
&= (y - h_\theta(x))\, x_j
\end{aligned}$$

随机梯度迭代公式：

$$\theta_j := \theta_j + \alpha \left( y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)}$$

# 构造代价函数角度

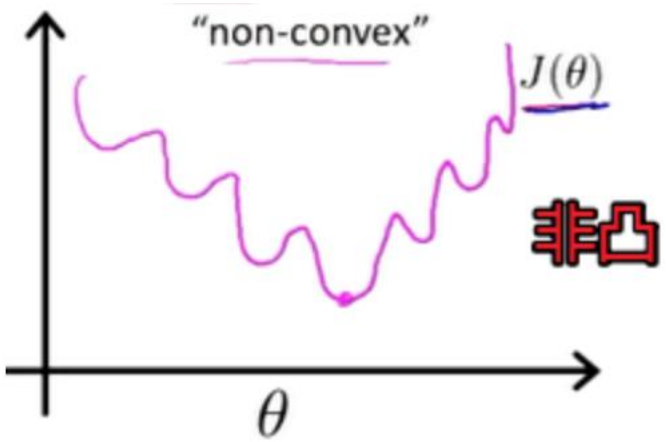
- 选择sigmoid函数做映射

代价函数：$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

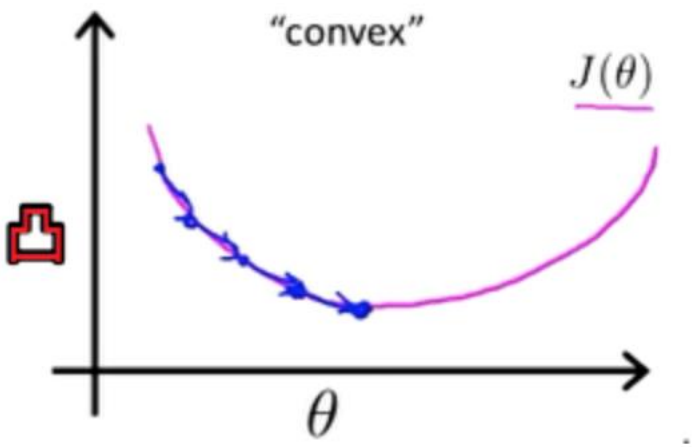
其中 $$h_\theta(x) = g(z) \qquad z = \theta^T x \qquad g(z) = \frac{1}{1+e^{-z}}$$

如何确定Cost的显示的形式？

# 理论推导




不好的损失函数




好的损失函数

选择平方差损失函数，经过sigmoid映射之后，为非凸。不合适

# 理论推导

选择平方差损失函数：

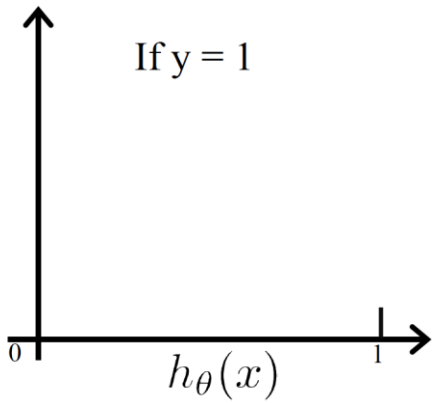
$$J(\theta) = \sum_{i=1}^{m} (y_i - h(x_i))^2$$

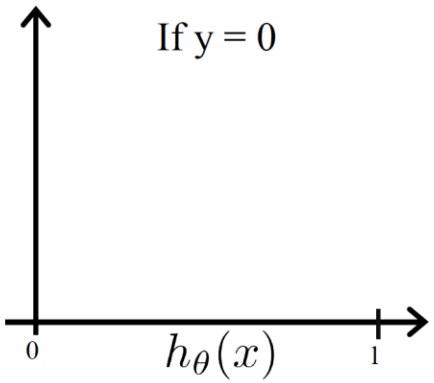
求导：

$$J'(\theta) = \sum_{i=1}^{m} 2(y_i - h(x_i))(-h'(\theta))\partial f = -2\sum_{i=1}^{m} (y_i - h(x_i))h(x_i)(1 - h(x_i))x_i$$

构造代价函数：

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)), & \text{if } y = 1 \\ -log(1 - h_\theta(x)), & \text{if } y = 0 \end{cases}$$




等价于：

$$Cost(h_\theta(x), y) = -ylog(h_\theta(x)) - (1 - y)log(1 - h_\theta(x))$$

- logistic回归是一种广义线性模型（generalized linear model），因此与多重线性回归分析有很多相同之处。它们的模型形式基本上相同，都具有 wx+b，其中w和b是待求参数，其区别在于他们的因变量不同，多重线性回归直接将wx+b作为因变量，即y =wx+b，而logistic回归则通过函数L将wx+b对应一个隐状态p，p =L(wx+b),然后根据p 与1-p的大小决定因变量的值。如果L是logistic函数，就是logistic回归。

- 指数分布族

$$p(y;\eta) = b(y)\exp(\eta^T T(y) - a(\eta))$$

T:充分统计量
$\eta$ :自然参数
$a(\eta)$被称为对数配分函数，实际上是归一化因子

我们见过的大多数分布都属于指数分布族，Bernoulli伯努利分布、Gaussian高斯分布、multinomial多项分布、Poisson泊松分布、gamma分布、指数分布、Dirichlet分布

- **指数分布族**
  - 正态分布明显指数分布

$$
\begin{aligned}
p(y;\mu) &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}(y-\mu)^2\right) \\
&= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}y^2\right)\cdot\exp\left(\mu y-\frac{1}{2}\mu^2\right)
\end{aligned}
$$

$$
\begin{aligned}
\eta &= \mu \\
T(y) &= y \\
a(\eta) &= \mu^2/2 \\
&= \eta^2/2 \\
b(y) &= (1/\sqrt{2\pi})\exp(-y^2/2)
\end{aligned}
$$

- **指数分布族**
  - 伯努力分布

$$
\begin{aligned}
p(y;\phi) &= \phi^y(1-\phi)^{1-y} \\
&= \exp(y\log\phi + (1-y)\log(1-\phi)) \\
&= \exp\left(\left(\log\left(\frac{\phi}{1-\phi}\right)\right)y + \log(1-\phi)\right)
\end{aligned}
$$

$$
\eta = \log(\phi/(1-\phi))
$$

$$
\begin{aligned}
T(y) &= y \\
a(\eta) &= -\log(1-\phi) \\
&= \log(1+e^{\eta}) \\
b(y) &= 1
\end{aligned}
$$

参数Φ指的是y=1的概率，
即事件发生的概率

- **广义线性模型三个假设：**
- 1. 我们的目标是求条件概率$p(y \mid x; \theta)$
- 2. 条件概率分布服从指数分布族
- 3. 指数族分布中的$\eta$和输入变量x的关联是线性的：$\eta = \theta^T x$

- 我们需要解决的是分类问题，并且是只分为两类，马上想到用
  - 1.伯努利分布来估计事件发生的概率。
  - 2.伯努利分布属于指数族分布
  - 3.不妨接受假设：$\eta = \theta^T x$

- 多分类softmax

如果需要分为k个类别，概率函数的取值：

Sigmoid 是 softmax 一个特例

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)}=1|x^{(i)};\theta) \\ p(y^{(i)}=2|x^{(i)};\theta) \\ \vdots \\ p(y^{(i)}=k|x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T \cdot x^{(i)}}} \begin{bmatrix} e^{\theta_1^T \cdot x^{(i)}} \\ e^{\theta_2^T \cdot x^{(i)}} \\ \vdots \\ e^{\theta_k^T \cdot x^{(i)}} \end{bmatrix}$$

代价函数：

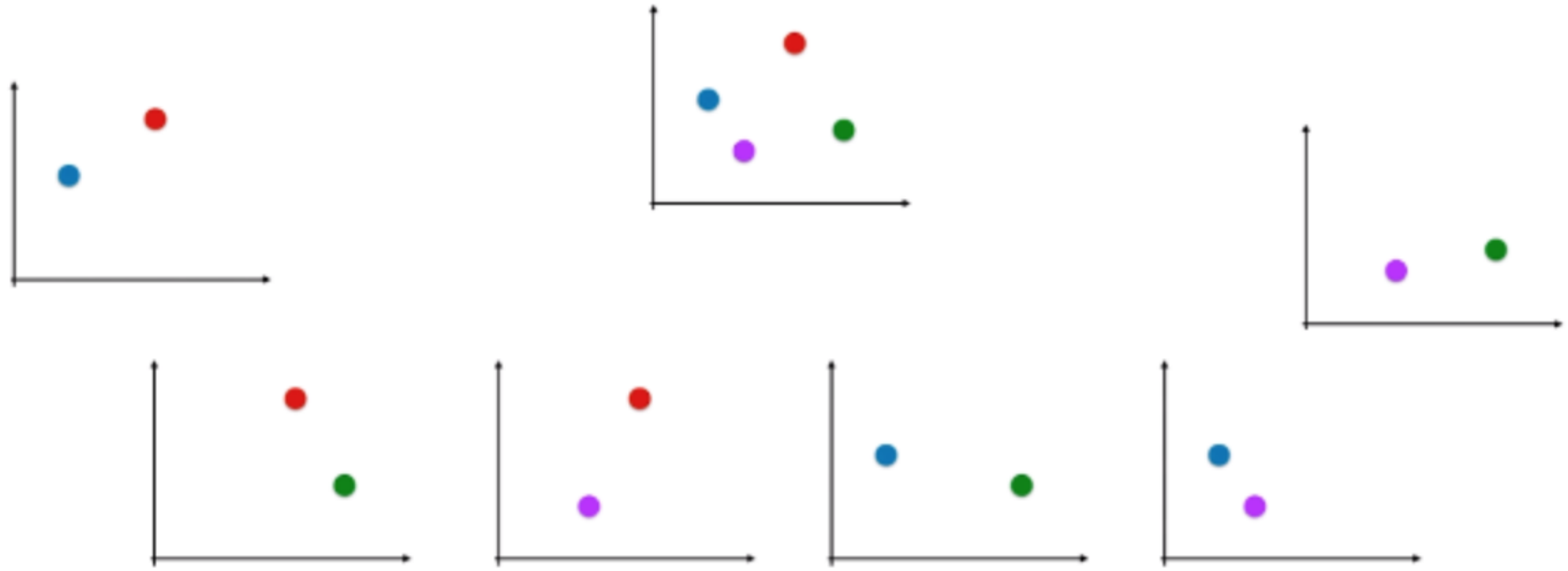
$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)}=j\} \cdot log(p(y^{(i)}=j|x^{(i)};\theta))\right]$$

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)}=j\} \cdot (\theta_j^T x^{(i)} - log(\sum_{l=1}^{k} e^{\theta_l^T \cdot x^{(i)}}))\right]$$

- **OvR 与 OVO**
  - 本质上是在二分类的问题上进行的策略改造。改造方法不是指针对逻辑回归算法，而是在机器学习领域有通用性
  - **OvR（One vs Rest）**
  - **OvO（One vs One）**

- **OvR**
  - **思想**：n 种类型的样本进行分类时，**分别**取一种样本作为一类，将剩余的所有类型的样本看做另一类，这样就形成了 **n 个**二分类问题，使用逻辑回归算法对 n 个数据集训练出 n 个模型，将待预测的样本传入这 n 个模型中，所得概率最高的那个模型对应的样本类型即认为是该预测样本的类型；

- **OvO**
  - **思想：** n 类样本中，每次挑出 2 种类型，两两结合，一共有 $C_n^2$ 种二分类情况，使用 $C_n^2$ 种模型预测样本类型，有 $C_n^2$ 个预测结果，种类最多的那种样本类型，就认为是该样本最终的预测类型；



- LR优缺点和注意事项:
  - 优点
    - 以概率的形式输出结果,可以用于排序
    - 可解释性强，可控度高
    - 训练快、效果不错
  - 注意事项：
    - 注意样本保持均衡
    - 合适的正则化

- sklearn是机器学习中一个常用的python第三方模块，网址：http://scikit-learn.org/stable/index.html

- 里面对一些常用的机器学习方法进行了封装，只需要简单的调用sklearn里的模块就可以实现大多数机器学习任务。

- 常用的算法包括SVM、KNN、贝叶斯、逻辑回归、线性回归、决策树等均有实现。

- Sklearn 进行鸢尾花分类（课堂实践）：

  - Iris（鸢尾花）数据集是多重变量分析的数据集。
    数据集包含150行数据，分为3类，每类50行数据。
    每行数据包括4个属性：Sepal Length（花萼长度）、Sepal Width（花萼宽度）、Petal Length（花瓣长度）、Petal Width（花瓣宽度）。可通过这4个属性预测鸢尾花属于3个种类的哪一类。

逻辑回归需要注意的参数：

1. 正则化选择参数：penalty

目的解决过拟合，一般penalty选择L2正则化就够了。
模型系数稀疏化，选择L1

2.优化算法：

**solver : *str, {'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'}, default: 'liblinear'.***

- For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga' are faster for large ones.
- For multiclass problems, only 'newton-cg', 'sag', 'saga' and 'lbfgs' handle multinomial loss; 'liblinear' is limited to one-versus-rest schemes.
- 'newton-cg', 'lbfgs' and 'sag' only handle L2 penalty, whereas 'liblinear' and 'saga' handle L1 penalty.

逻辑回归需要注意的参数：

3. 分类方式：multi_class
ovr和multinomial两个值可以选择，默认是 ovr

4. class_weight 和 sample_weight
用于解决样本不平衡问题的。