线性回归与广义线性回归

3月机器学习在线班 邹博

2015年3月15日

预备公式：求导

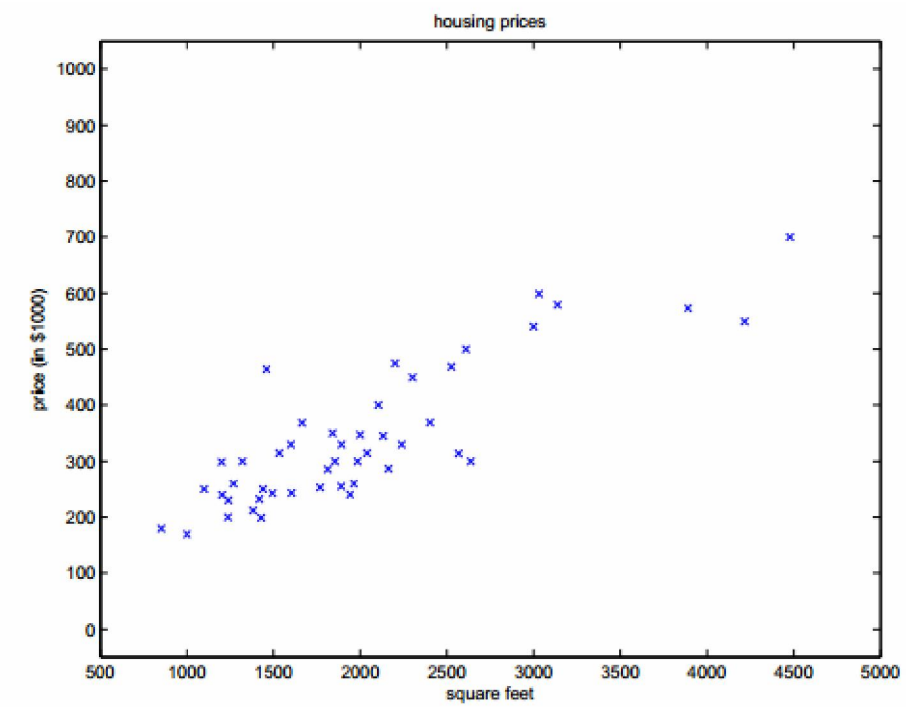
$$\frac{\partial X\theta}{\partial \theta} = X^T$$

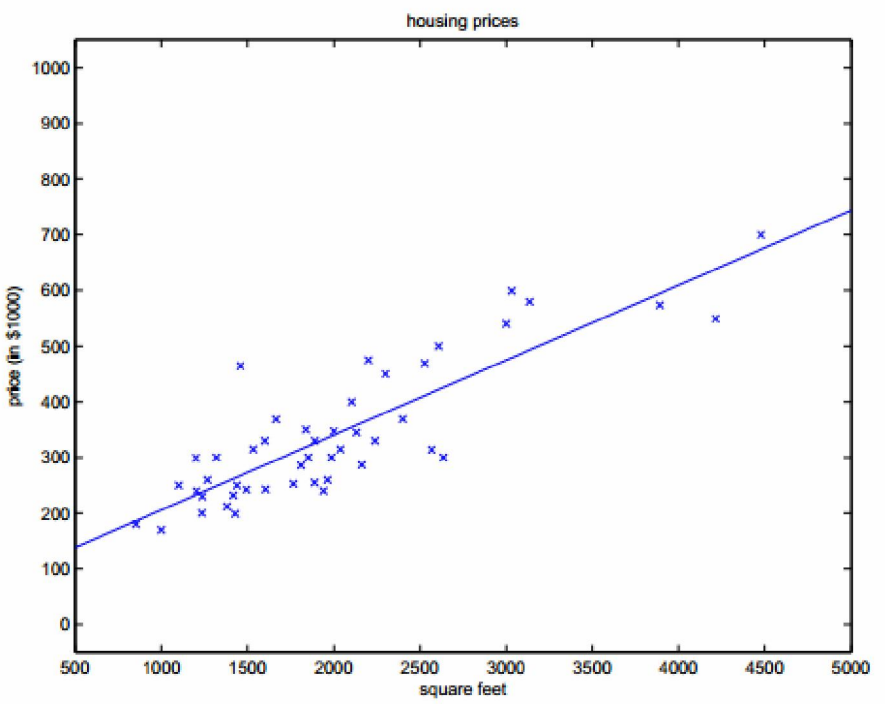
$$\frac{\partial \theta^T X}{\partial \theta^T} = X^T$$

$$\frac{\partial \theta^T X}{\partial \theta} = X$$

线性回归

- y=ax+b


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


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

多个变量的情形

- 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
⋮	⋮	⋮

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

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

最小二乘的目标函数

- m为样本个数，则一个比较“符合常理”的误差函数为：

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

- 符合常理
  - 最小二乘建立的目标函数，即是在噪声为均值为0的高斯分布下，极大似然估计的目标函数

使用极大似然估计解释最小二乘

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

the $\epsilon^{(i)}$ are distributed IID (independently and identically distributed) according to a Gaussian distribution (also called a Normal distribution) with mean zero and some variance σ^2

似然函数

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right)$$

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

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

对数似然

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

θ 的解析式的求解过程

$$\nabla_\theta(X\theta) = X^T$$

$$\begin{aligned}\frac{1}{2}(X\theta-\vec{y})^T(X\theta-\vec{y}) &= \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2 \\ &= J(\theta)\end{aligned}$$

$$\begin{aligned}\nabla_\theta J(\theta) &= \nabla_\theta\frac{1}{2}(X\theta-\vec{y})^T(X\theta-\vec{y}) \\ &= \frac{1}{2}\nabla_\theta\left(\theta^TX^TX\theta-\theta^TX^T\vec{y}-\vec{y}^TX\theta+\vec{y}^T\vec{y}\right) \\ &= \frac{1}{2}\left(X^TX\theta+X^TX\theta-2X^T\vec{y}\right) \\ &= X^TX\theta-X^T\vec{y}\end{aligned}$$

最小二乘意义下的参数最优解

- 参数的解析式

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

- 若X^TX不可逆，上式不可使用

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

附：“简便”方法记忆结论

$$X\theta = y$$

$$X^T X\theta = X^T y$$

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

加入λ扰动后

- X^TX半正定：对于任意的非零向量u

$$uX^TXu = \left(Xu\right)^T Xu \xrightarrow{令v=Xu} v^Tv \geq 0$$

- 所以，对于任意的实数$\lambda>0$，$X^TX+\lambda I$正定，从而可逆。保证回归公式一定有意义。

$$\theta = \left(X^TX+\lambda I\right)^{-1}X^Ty$$

若X^TX阶过高

- 实际中，若X^TX阶过高，仍然需要使用梯度下降的方式计算数值解。

广义逆矩阵（伪逆）$A^+ = \left(A^T A\right)^{-1} A^T$

- 若A为非奇异矩阵,则线性方程组Ax=b的解为$x = A^{-1}b$ 其中A的A的逆矩阵A^{-1}满足$A^{-1}A = AA^{-1} = I$ (I为单位矩阵)。若A是奇异阵或长方阵, $x=A^+ b$。A^+叫做A的伪逆阵。
- 1955年R.彭罗斯证明了对每个m×n阶矩阵A,都存在惟一的n×m阶矩阵X，满足：①AXA=A；②XAX=X；③$(AX)^* = I$；④$(XA)^* = I$。通常称X为A的穆尔-彭罗斯广义逆矩阵,简称M-P逆，记作A^+。
- 在矛盾线性方程组Ax = b的最小二乘解中, $x=A^+b$是范数最小的一个解。
  - 在奇异值分解SVD的问题中，将继续该话题的讨论。

梯度下降算法

- 初始化θ（随机初始化）
- 迭代，新的θ能够使得J(θ)更小
- 如果J(θ)能够继续减少，返回（2）

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

- α称为学习率

梯度方向

$$
\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \left(h_\theta(x) - y \right)^2 \\
&= 2 \cdot \frac{1}{2} \left(h_\theta(x) - y \right) \cdot \frac{\partial}{\partial \theta_j} \left(h_\theta(x) - y \right) \\
&= \left(h_\theta(x) - y \right) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{n} \theta_i x_i - y \right) \\
&= \left(h_\theta(x) - y \right) x_j
\end{aligned}
$$

批处理梯度下降算法

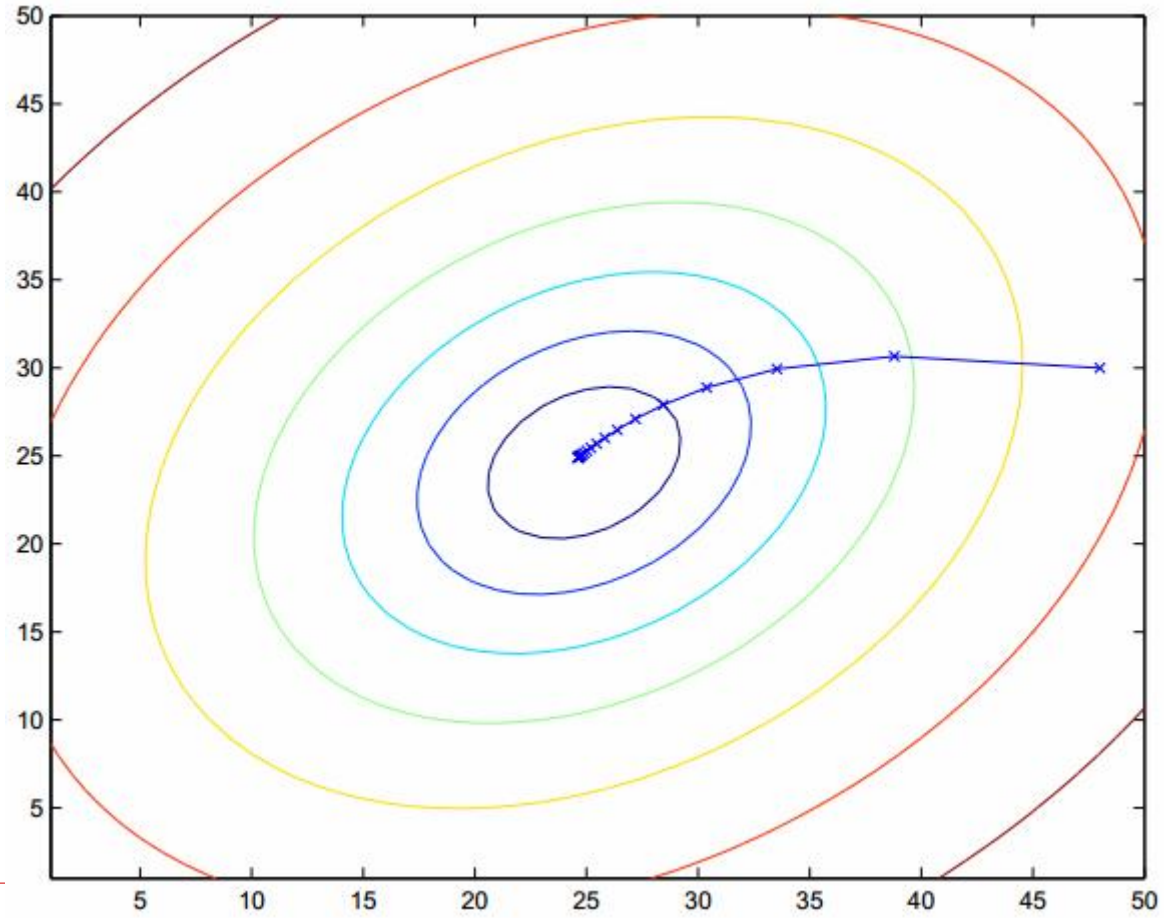
Repeat until convergence {

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

}

gradient descent. Note that, while gradient descent can be susceptible to local minima in general, the optimization problem we have posed here for linear regression has only one global, and no other local, optima; thus gradient descent always converges (assuming the learning rate α is not too large) to the global minimum. Indeed, J is a convex quadratic function.

批处理梯度下降图示



随机梯度下降算法

Loop {

for i=1 to m, {

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

}

}

This algorithm is called **stochastic gradient descent** (also **incremental gradient descent**). Whereas batch gradient descent has to scan through the entire training set before taking a single step—a costly operation if m is large—stochastic gradient descent can start making progress right away, and continues to make progress with each example it looks at. Often, stochastic gradient descent gets θ "close" to the minimum much faster than batch gradient descent. (Note however that it may never "converge" to the minimum, and the parameters θ will keep oscillating around the minimum of $J(\theta)$; but in practice most of the values near the minimum will be reasonably good approximations to the true minimum.[2]) For these reasons, particularly when the training set is large, stochastic gradient descent is often preferred over batch gradient descent.

mini-batch

- 如果不是每拿到一个样本即更改梯度，而是若干个样本的平均梯度作为更新方向，则是mini-batch梯度下降算法。

Repeat until convergence {

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

}

Loop {

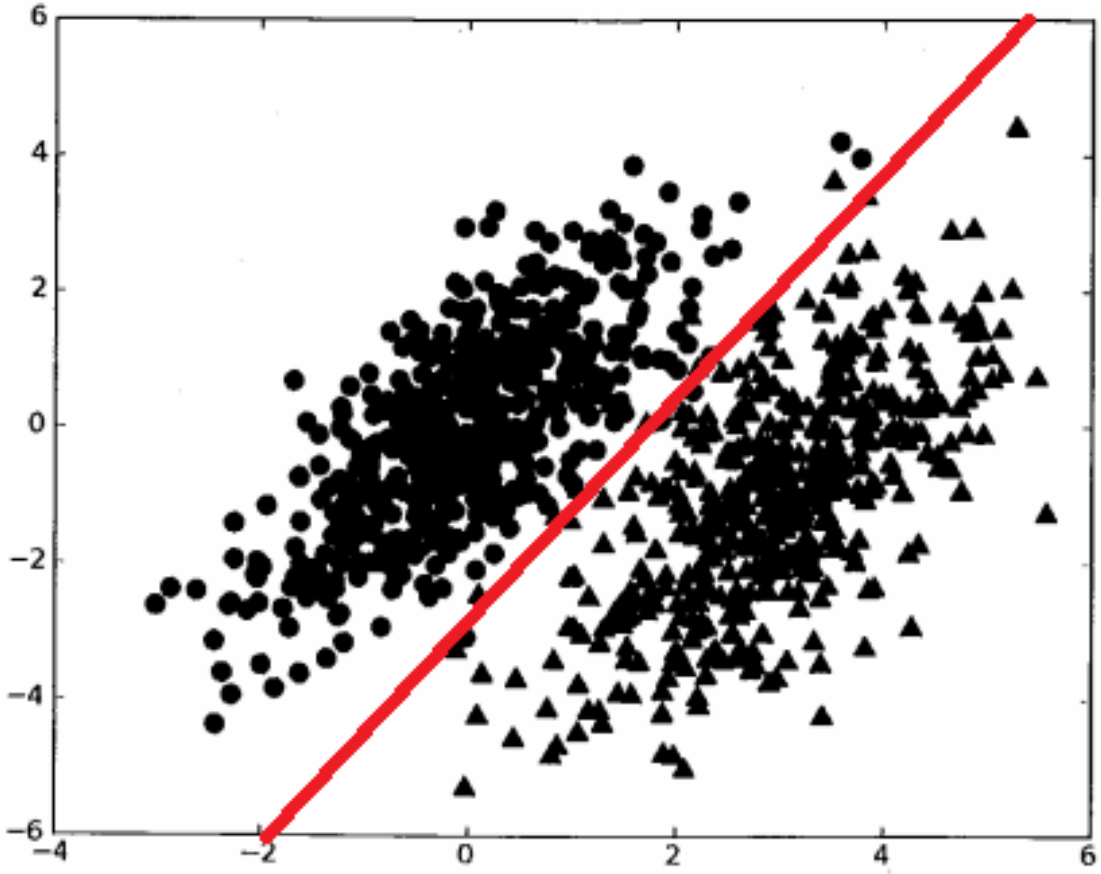
for i=1 to m, {

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

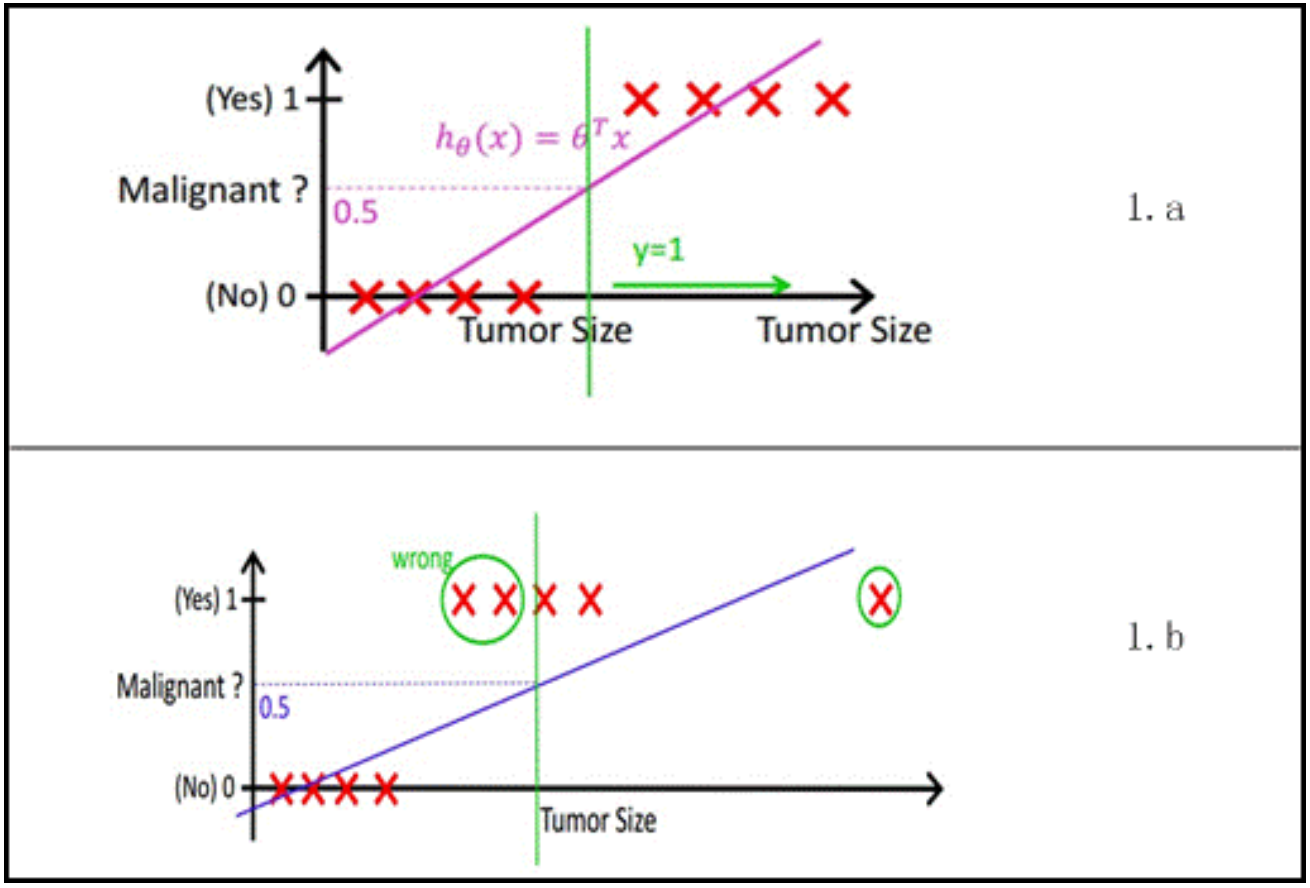
}

}

用回归解决分类问题，如何？

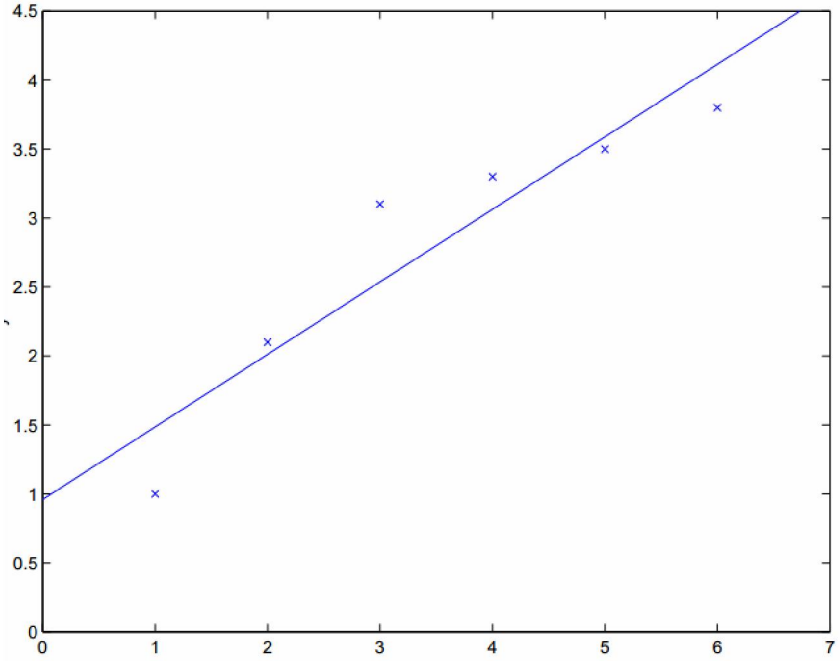


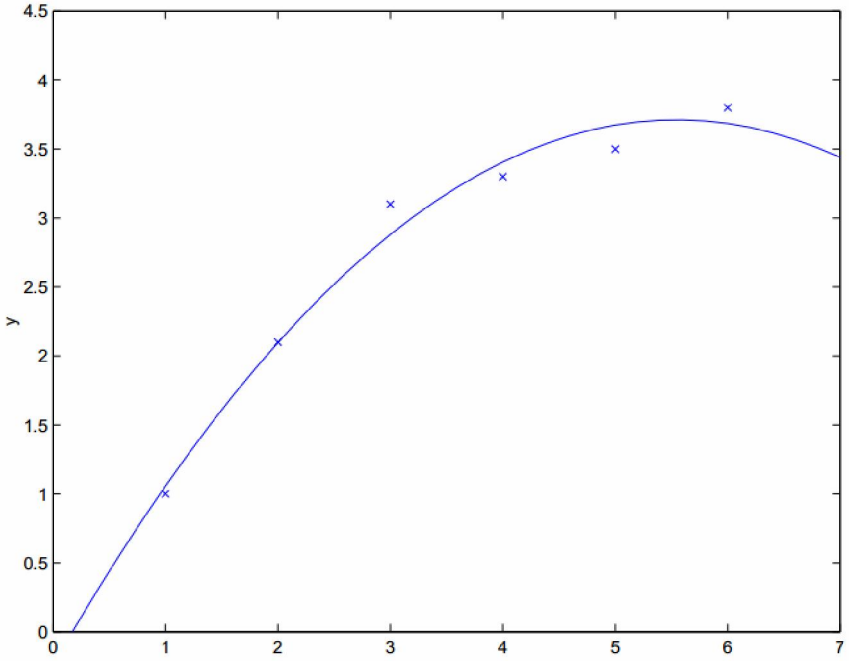
最简单的例子：一维回归


(Yes) 1
Malignant ?
0.5
(No) 0
$h_\theta(x) = \theta^T x$
y=1
Tumor Size
Tumor Size
1. a
wrong
(Yes) 1
Malignant ?
0.5
(No) 0
Tumor Size
1. b

线性回归的进一步分析

- 可以对样本是非线性的，只要对参数θ线性





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

局部加权线性回归

- LWR：Locally Weighted linear Regression

1. Fit θ to minimize $\sum_i (y^{(i)} - \theta^T x^{(i)})^2$
2. Output $\theta^T x$.

1. Fit θ to minimize $\sum_i w^{(i)} (y^{(i)} - \theta^T x^{(i)})^2$
2. Output $\theta^T x$.

权值的设置

- ω的一种可能的选择方式(高斯核函数):

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

- τ称为带宽，它控制着训练样本随着与$x^{(i)}$距离的衰减速率。
- 多项式核函数

$$\kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle + R)^d$$

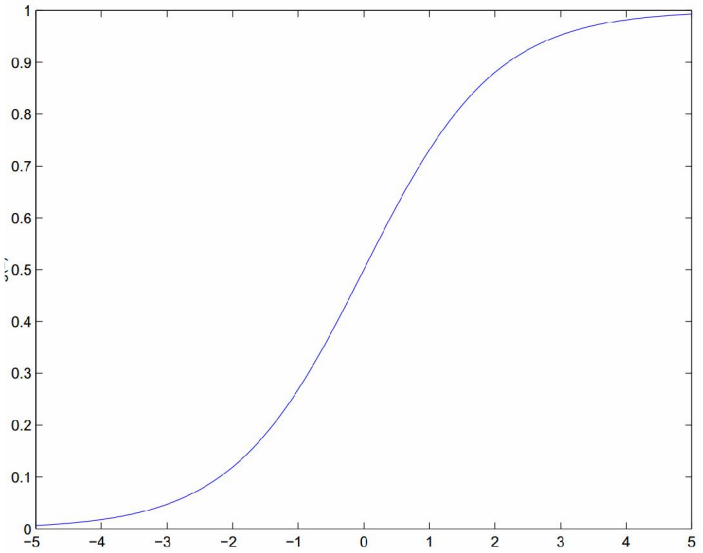
参数算法 – 非参数学习算法

Locally weighted linear regression is the first example we're seeing of a **non-parametric** algorithm. The (unweighted) linear regression algorithm that we saw earlier is known as a **parametric** learning algorithm, because it has a fixed, finite number of parameters (the θ_i's), which are fit to the data. Once we've fit the θ_i's and stored them away, we no longer need to keep the training data around to make future predictions. In contrast, to make predictions using locally weighted linear regression, we need to keep the entire training set around. The term "non-parametric" (roughly) refers to the fact that the amount of stuff we need to keep in order to represent the hypothesis h grows linearly with the size of the training set.

Logistic回归

- Logistic函数

$$g(z) = \frac{1}{1 + e^{-z}}$$



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

Logistic函数的导数

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

Logistic回归参数估计

- 假定：

$$P(y=1 \mid x;\theta) = h_\theta(x)$$
$$P(y=0 \mid x;\theta) = 1 - h_\theta(x)$$

$$p(y \mid x;\theta) = (h_\theta(x))^y \, (1 - h_\theta(x))^{1-y}$$

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

对数似然函数

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

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

参数的迭代

- Logistic回归参数的学习规则：

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

- 比较上面的结果和线性回归的结论的差别：
  - 它们具有相同的形式！

Repeat until convergence {

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

}

Loop {

for i=1 to m, {

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

}

}

对数线性模型

- 一个事件的几率odds，是指该事件发生的概率与该事件不发生的概率的比值。
- 对数几率：logit函数

$$
\begin{aligned}
P(y=1 \mid x;\theta) &= h_\theta(x) \\
P(y=0 \mid x;\theta) &= 1-h_\theta(x)
\end{aligned}
$$

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

复习：指数族

The exponential family

To work our way up to GLMs, we will begin by defining exponential family distributions. We say that a class of distributions is in the exponential family if it can be written in the form

$$p(y;\eta) = b(y)\exp(\eta^T T(y) - a(\eta)) \tag{6}$$

Here, η is called the **natural parameter** (also called the **canonical parameter**) of the distribution; $T(y)$ is the **sufficient statistic** (for the distributions we consider, it will often be the case that $T(y) = y$); and $a(\eta)$ is the **log partition function**. The quantity $e^{-a(\eta)}$ essentially plays the role of a normalization constant, that makes sure the distribution $p(y;\eta)$ sums/integrates over y to 1.

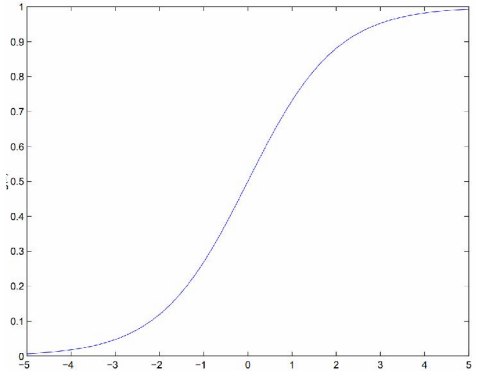
A fixed choice of T, a and b defines a *family* (or set) of distributions that is parameterized by η; as we vary η, we then get different distributions within this family.

复习：指数族

- 指数族概念的目的，是为了说明广义线性模型Generalized Linear Models
- 凡是符合指数族分布的随机变量，都可以用GLM回归分析

广义线性模型GLM

- 因变量y不再只是正态分布，而是扩大为指数族中的任一分布；
- 解释变量x的线性组合不再直接用于解释因变量y的均值u，而是通过一个联系函数g来解释g(u)；这里，要求g连续单调可导
  - 如Logistic回归中的 $g(z) = \frac{1}{1+e^{-z}}$



连接函数

- 连接函数：单调可导
- 恒等：g(u)=u，线性模型即使在正态分布下的恒等连接的广义线性模型，
- 对数：g(u)=ln(u)，因为对数的逆是指数，因此它可以将原本线性关系转变成乘积关系；
- Logit：g(u)=ln(u/1-u)，它的特点为可将预测值控制在0~1之间，对于因变量y为比率时适合使用

其他连接函数

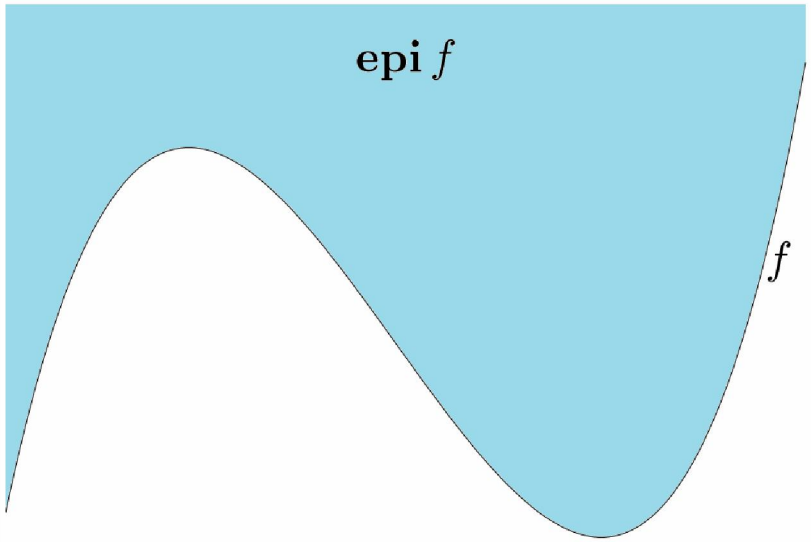
- 如：可以将Logistic函数做拉伸变换，得到新的连接函数

$$g(z) = \frac{1}{1 + e^{-\lambda z}}$$

上境图

- 函数f的图像定义为：$\{(x, f(x)) \,|\, x \in \mathbf{dom}\, f\}$
- 函数f的上境图(epigraph)定义为：

$$\mathbf{epi}\, f = \{(x, t) \mid x \in \mathbf{dom}\, f,\ f(x) \leq t\}$$



凸函数与凸集

- 一个函数是凸函数，当且仅当其上境图是凸集。
  - 思考：如何证明？(提示：定义)
- 进一步，一个函数是凹函数，当且仅当其亚图(hypograph)是凸集。

$$\mathbf{hypo}\, f = \{(x,t) \mid t \leq f(x)\}$$

凸函数的逐点最大值

- f1,f2均为凸函数，定义函数f:

$$f(x) = \max\{f_1(x), f_2(x)\}$$

- 则函数f为凸函数。

$$\begin{aligned}
& f(\theta x + (1-\theta)y) \\
= \ & \max\{f_1(\theta x + (1-\theta)y), f_2(\theta x + (1-\theta)y)\} \\
\leq \ & \max\{\theta f_1(x) + (1-\theta) f_1(y), \theta f_2(x) + (1-\theta) f_2(y)\} \\
\leq \ & \theta \max\{f_1(x), f_2(x)\} + (1-\theta) \max\{f_1(y), f_2(y)\} \\
= \ & \theta f(x) + (1-\theta) f(y)
\end{aligned}$$

思考

- 逐点上确界和上境图的关系
  - 一系列函数逐点上确界函数对应着这些函数上境图的交集。
  - Oxy平面上随意画N条直线，在每个x处取这些直线的最大的点，则构成的新函数是凸函数。
  - 点x到任意集合C的最远距离 $f(x) = \sup_{y \in C} \|x - y\|$
    - f是凸函数
    - 证明：范数是凸的(思考：为什么？)，逐点求上界，仍然是凸的。

共轭函数

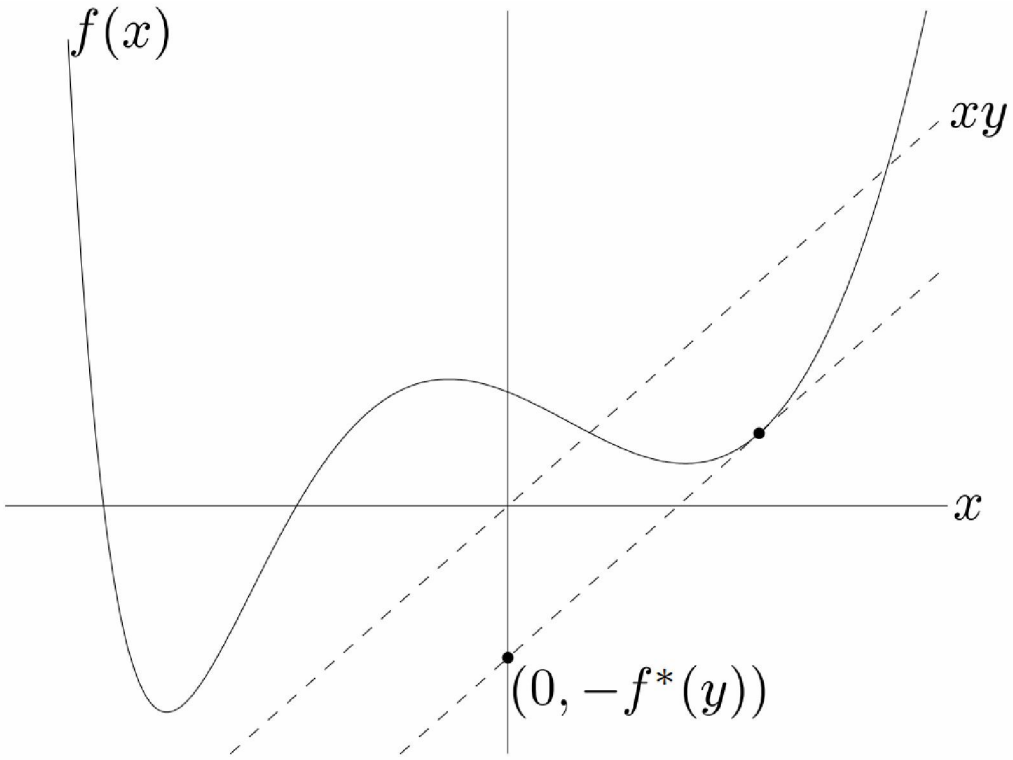
- 原函数 $f : \mathbf{R}^n \to \mathbf{R}$ 共轭函数定义：

$$f^*(y) = \sup_{x \in \mathbf{dom}\, f} \left(y^T x - f(x)\right)$$

- 显然，定义式的右端是关于y的仿射函数，它们逐点求上确界，得到的函数f*(y)一定是凸函数。
- 该名称的原因：
  - 凸函数的共轭函数的共轭函数是其本身。

对共轭函数的理解 $f^*(y) = \sup_{x \in \mathbf{dom}\, f} \left(y^T x - f(x)\right)$

- 如果函数f可微，在满足f'(x)=y的点x处差值最大。



例：求共轭函数

- 可逆对称阵Q，对于任意的向量x，定义函数f：$f(x)=\frac{1}{2}x^TQx$
- 关于(x,y)的函数 $y^Tx-\frac{1}{2}x^TQx$
- 在 $x=Q^{-1}y$ 时取上确界，带入，得到：

$$f^*(y)=\frac{1}{2}y^TQ^{-1}y$$

- f*即是f的共轭函数

思考

- $f(x)=x^2$ 的共轭函数是什么？
- 答：$f(x)=x^2$ 的共轭函数是 $f(y)=\frac{1}{4}y^2$
- 我们发现：

$$f(x)+f(y)=x^2+\frac{1}{4}y^2 \geq xy$$

Fenchel不等式

- 根据定义

$$f^*(y) = \sup_{x \in \mathbf{dom}\, f} \left(y^T x - f(x)\right)$$

- 立刻可以得到：

$$f(x) + f^*(y) \geq x^T y$$

Fenchel不等式的应用

- 根据f(x)及其共轭函数f*(x)

$$f(x)=\frac{1}{2}x^TQx \qquad f^*(y)=\frac{1}{2}y^TQ^{-1}y$$

- 带入Fenchel不等式，得到：

$$x^TQx+y^TQ^{-1}y\geq 2x^Ty$$

凸优化

- 优化问题的基本形式

$$\begin{aligned} \text{minimize} \quad & f_0(x), \quad x \in \mathbf{R}^n \\ \text{subject to} \quad & f_i(x) \le 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, p \end{aligned}$$

优化变量 $x \in R^n$

不等式约束 $f_i(x) \le 0$

等式约束 $h_j(x) = 0.$

无约束优化 $m = p = 0$

优化问题的基本形式

- 优化问题的域 $$D = \bigcap_{i=0}^{m} \mathrm{dom} f_i \cap \bigcap_{j=1}^{p} \mathrm{dom} h_j$$
- 可行点(解)(feasible)
  - x∈D，且满足约束条件
- 可行域(可解集)
  - 所有可行点的集合
- 最优化值

$$p^* = \inf\{f_0(x) \mid f_i(x) \le 0, i = 1, \ldots, m, h_j(x) = 0, j = 1, \ldots, p\}$$

- 最优化解 $$p^* = f_0(x^*)$$

局部最优问题

$$\begin{aligned} \text{minimize} \quad & f_0(x), \quad x \in \mathbf{R}^n \\ \text{subject to} \quad & f_i(x) \le 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, p \\ & \|x - z\|_2 \le R, \; R > 0 \end{aligned}$$

凸优化问题的基本形式

$$\begin{aligned} \text{minimize} \quad & f_0(x), \quad x \in \mathbf{R}^n \\ \text{subject to} \quad & f_i(x) \le 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, p \end{aligned}$$

- 其中，$f_i(x)$为凸函数，$h_j(x)$为仿射函数
- 凸优化问题的重要性质
  - 凸优化问题的可行域为凸集
  - 凸优化问题的局部最优解即为全局最优解

对偶问题

- 一般优化问题的Lagrange乘子法

$$\begin{aligned} \text{minimize} \quad & f_0(x), \quad x \in \mathbf{R}^n \\ \text{subject to} \quad & f_i(x) \le 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, p \end{aligned}$$

- Lagrange函数

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{p} \nu_j h_j(x)$$

  - 对固定的x，Lagrange函数L(x, λ ,v)为关于λ和v的仿射函数

Lagrange对偶函数(dual function)

- Lagrange对偶函数

$$g(\lambda,\nu)=\inf_{x\in D}L(x,\lambda,\nu)=\inf_{x\in D}\left(f_0(x)+\sum_{i=1}^{m}\lambda_i f_i(x)+\sum_{i=1}^{p}\nu_i h_i(x)\right)$$

- 若没有下确界，定义：

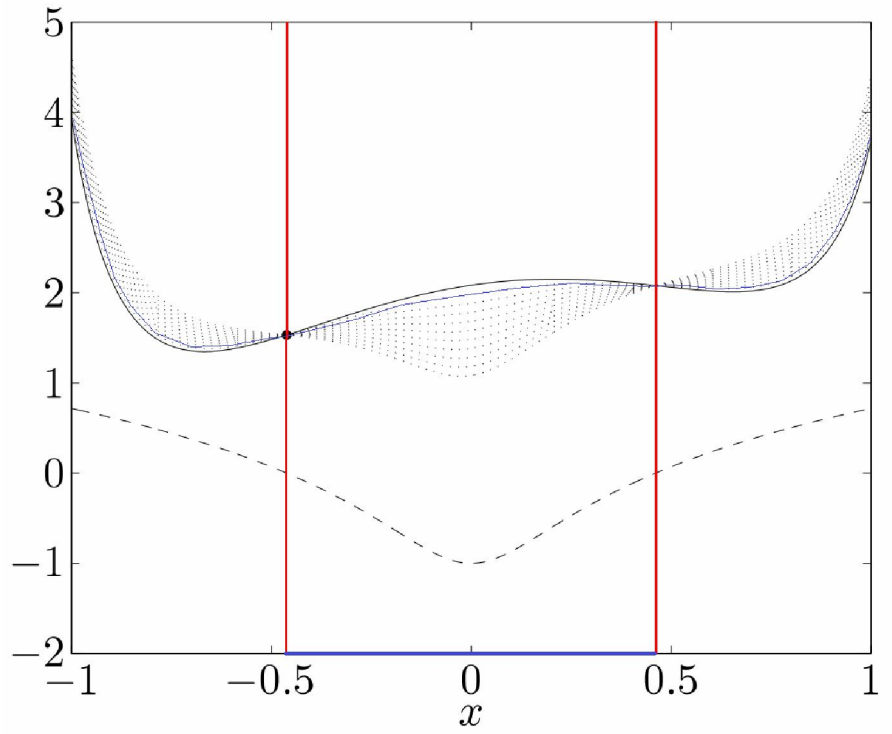
$$g(\lambda,\nu)=-\infty$$

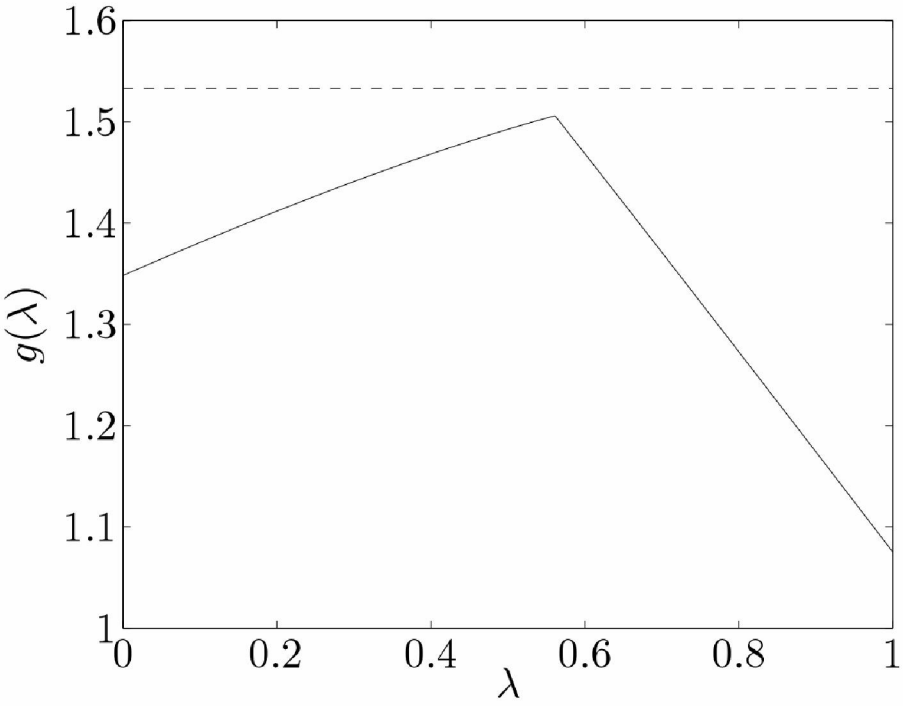
- 根据定义，显然有：对$\forall\ \lambda>0$，$\forall$v，若原优化问题有最优值p*，则

$$g(\lambda,\nu)\le p^*$$

- 进一步：Lagrange对偶函数为凹函数。

左侧为原函数，右侧为对偶函数





鞍点解释

- 为表述方便，假设没有等式约束，只考虑不等式约束，结论可方便的扩展到等式约束。
- 假设x0不可行，即存在某些i，使得$f_i(x)>0$。则选择 $\lambda_i \to \infty$，对于其他乘子，$\lambda_j = 0, j \neq i$
- 假设x0可行，则有$f_i(x) \leqslant 0,(i=1,2,...,m)$,选择

$$\lambda_i = 0, i = 1,2,\cdots,m$$

- 有：

$$\sup_{\lambda>0} L(x,\lambda) = \sup_{\lambda>0}\left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \right) = \begin{cases} f_0(x) & f_i(x) \leq 0, i = 1,2\cdots,m \\ \infty & other \end{cases}$$

鞍点：最优点

- 而原问题是：$\inf_x f_0(x)$
- 从而，原问题的本质为：$\inf_x \sup_{\lambda>0} L(x,\lambda)$
- 而对偶问题，是求对偶函数的最大值，即：

$$\sup_{\lambda>0} \inf_x L(x,\lambda)$$

- 而：

$$\sup_{\lambda>0} \inf_x L(x,\lambda) \le \inf_x \sup_{\lambda>0} L(x,\lambda)$$

证明：$\max_x \min_y f(x,y) \le \min_y \max_x f(x,y)$

- 对于任意的(x,y) ∈ domf

$$f(x,y) \le \max_x f(x,y)$$

$$\Rightarrow \min_y f(x,y) \le \min_y \max_x f(x,y)$$

$$\Rightarrow \max_x \min_y f(x,y) \le \min_y \max_x f(x,y)$$

线性方程的最小二乘问题

- 原问题 $\text{minimize} \quad x^T x, \quad x \in \mathbf{R}^n$

  $\text{subject to} \quad Ax = b$

- Lagrange函数

$$L(x, \nu) = x^T x + \nu^T (Ax - b)$$

- Lagrange对偶函数

$$g(\nu) = -\frac{1}{4}\nu^T A A^T \nu - b^T \nu$$

  - 对L求x的偏导，带入L，得到g
  - 对g求v的偏导，求g的极大值，作为原问题的最小值

求L的对偶函数 $L(x,\nu)=x^Tx+\nu^T(Ax-b)$

$$\frac{\partial L}{\partial x}=\frac{\partial\left(x^Tx+\nu^T(Ax-b)\right)}{\partial x}=2x+A^T\nu\overset{令}{=}0\Rightarrow x^*=-\frac{1}{2}A^T\nu$$

$$L(x,\nu)=x^Tx+\nu^T(Ax-b)$$

$$=\left(-\frac{1}{2}A^T\nu\right)^T\left(-\frac{1}{2}A^T\nu\right)+\nu^T\left(A\left(-\frac{1}{2}A^T\nu\right)-b\right)$$

$$=\frac{1}{4}\nu^TAA^T\nu-\frac{1}{2}\nu^TAA^T\nu-\nu^Tb$$

$$=-\frac{1}{4}\nu^TAA^T\nu-\nu^Tb$$

$$\overset{\Delta}{=}g(\nu)$$

求对偶函数的极大值 $g(\nu)=-\frac{1}{4}\nu^T AA^T\nu-\nu^T b$

$$\frac{\partial g}{\partial \nu}=\frac{\partial\left(-\frac{1}{4}\nu^T AA^T\nu-\nu^T b\right)}{\partial \nu}=-\frac{1}{2}AA^T\nu-b\overset{令}{=}0$$

$$\Rightarrow AA^T\nu=-2b$$

$$\Rightarrow A^T AA^T\nu=-2A^T b$$

$$\Rightarrow A^T\nu=-2\left(A^T A\right)^{-1}A^T b$$

$$\Rightarrow -\frac{1}{2}A^T\nu=\left(A^T A\right)^{-1}A^T b$$

$$\Rightarrow x^*=\left(A^T A\right)^{-1}A^T b$$

极小值点 $x^* = \left(A^T A\right)^{-1} A^T b$

- 极小值：$\min\left(x^T x\right)$

$$= \left(\left(A^T A\right)^{-1} A^T b\right)^T \left(\left(A^T A\right)^{-1} A^T b\right)$$

$$= b^T A\left(A^T A\right)^{-1}\left(A^T A\right)^{-1} A^T b$$

$$= b^T A\left(A^T A\right)^{-2} A^T b$$

- 极小值点的结论，和通过线性回归计算得到的结论是完全一致的。
  - 线性回归问题具有强对偶性。

强对偶条件

- 若要对偶函数的最大值即为原问题的最小值，考察需要满足的条件：

$$
\begin{aligned}
f_0(x^\star) &= g(\lambda^\star, \nu^\star) \\
&= \inf_x \left(f_0(x) + \sum_{i=1}^{m} \lambda_i^\star f_i(x) + \sum_{i=1}^{p} \nu_i^\star h_i(x) \right) \\
&\leq f_0(x^\star) + \sum_{i=1}^{m} \lambda_i^\star f_i(x^\star) + \sum_{i=1}^{p} \nu_i^\star h_i(x^\star) \\
&\leq f_0(x^\star).
\end{aligned}
$$

Karush-Kuhn-Tucker (KKT)条件

$$
\begin{aligned}
f_i(x^\star) &\leq 0, \quad i = 1, \ldots, m \\
h_i(x^\star) &= 0, \quad i = 1, \ldots, p \\
\lambda_i^\star &\geq 0, \quad i = 1, \ldots, m \\
\lambda_i^\star f_i(x^\star) &= 0, \quad i = 1, \ldots, m
\end{aligned}
$$

$$
\nabla f_0(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \nabla f_i(x^\star) + \sum_{i=1}^{p} \nu_i^\star \nabla h_i(x^\star) = 0
$$

参考文献

- Prof. Andrew Ng, Machine Learning, Stanford University
- 统计学习方法，李航著，清华大学出版社，2012年
- Convex Optimization, Stephen Boyd,Lieven Vandenberghe, Cambridge University Press, 2004
  - 中译本：王书宁，许鋆，黄晓霖 译，凸优化，清华大学出版社，2013

感谢大家！

恳请大家批评指正！