

2024-2025学年 第1学期(秋)



数据挖掘

第9章 回归方法

2024 年 10 月

案例引入

- 买房问题

房屋价格销售表

假设有一个房屋销售的数据如下：

面积(m ²)	销售价钱 (万元)
123	250
150	320
87	160
102	220
...	...



案例引入



一般想法：

画图看看到底两者是什么关系，或者像是什么关系，总之就是期望能够找到两者之间的**映射关系**

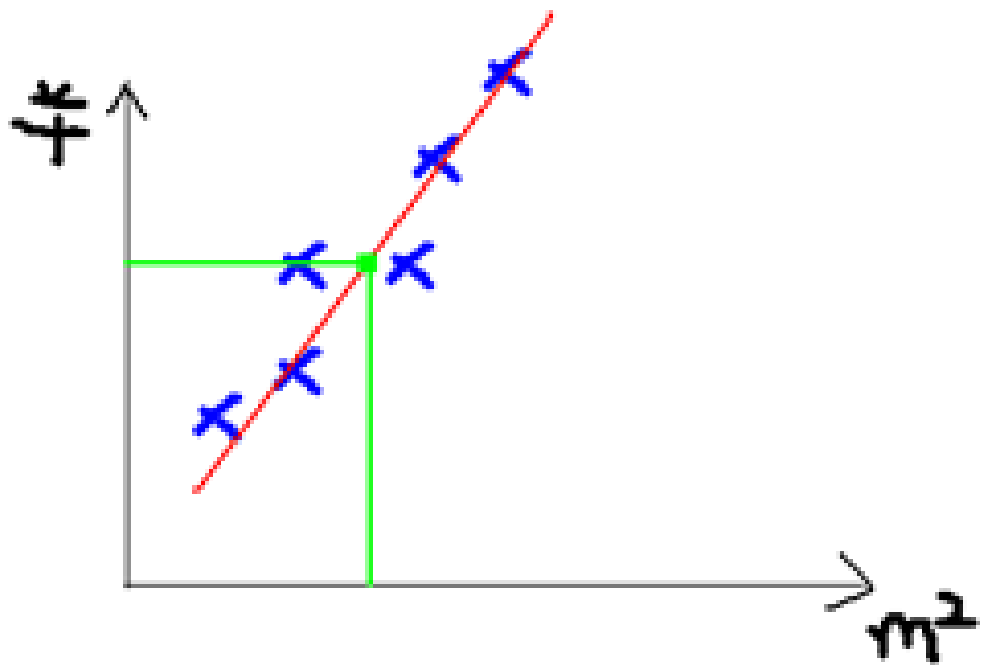


案例引入



一般想法：

我们可以用一条直线去尽量准的拟合这些数据，从而找到面积与房价之间的因果关系：直线的斜率就是每平的均价



案例引入



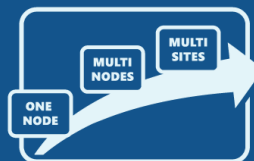
房屋销售记录表

训练集(training set)或者训练数据(training data)。一条训练数据是由一对输入数据和输出数据组成的



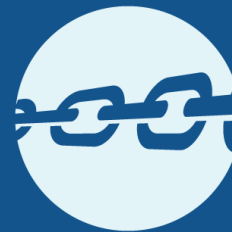
房屋面积

输入数据, 一般称为 x



房屋销售价

输出数据, 一般称为 y



拟合的函数 (假设或者模型)

一般写做 $y = h(x)$



南京大學
NANJING UNIVERSITY

目录

01

线性回归

02

优化求解

03

逻辑回归

03

决策树回归

回归问题

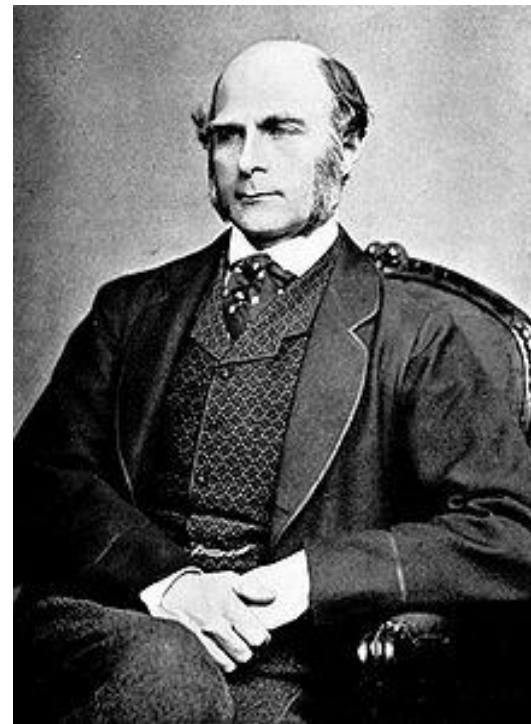
- 回归分析
 - 如果把其中的一些因素（房屋面积）作为**自变量**，而另一些随自变量的变化而变化的变量作为**因变量**（房价），研究他们之间的非确定映射关系，这种分析就称为**回归分析（regression）**。
 - 回归分析是研究一个或多个自变量与一个因变量之间是否存在某种**线性关系**或**非线性关系**的一种统计学方法。

回归问题的起源

- 回归问题的来源

英国著名的统计学家F. Galton研究了1078对夫妇及其一个成年儿子的身高关系。他们以儿子身高作为纵坐标、夫妇平均身高为横坐标作散点图，结果发现二者的关系近似于一条直线。经计算得到了如下方程：

$$y=0.8567+0.516x$$



Francis Galton
英国19世纪统计学家

Galton引进“**回归**” (regression) 一词来表达这种方程关系

线性回归

- 线性回归假设特征和响应满足线性关系
- 一元线性回归问题函数关系可表示

$$y = a + bx$$

- 根据上式，在确定a、b的情况下，给定一个x值，我们就能够得到一个确定的y值，然而根据上式得到的y值与实际的y值存在一个误差
 - a、b为参数(parameters)、或称回归系数(regression coefficients)
- 采用什么样的线性关系误差刻画更好呢？

模型刻画

用什么样的方法刻画点与直线的距离会方便有效？

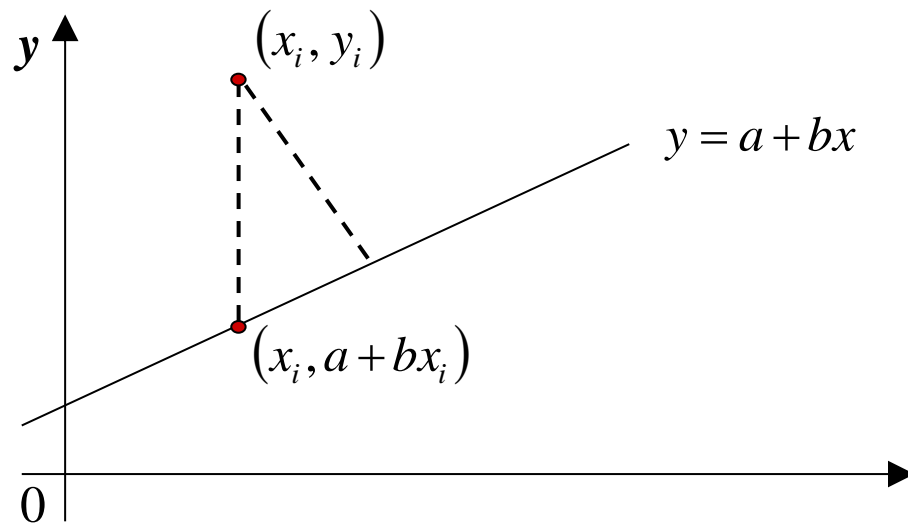
设直线方程为 $y=a+bx$ ，样本点 $A(x_i, y_i)$

方法一：点到直线的距离公式

$$d = \frac{|bx_i - y_i + a|}{\sqrt{b^2 + 1}}$$

方法二：

$$[y_i - (a + bx_i)]^2$$



显然方法二能有效地表示点 A 与直线 $y=a+bx$ 的距离，而且比方法一更方便计算，所以我们用它来表示二者之间的接近程度

最小二乘法

- 基本思想

- 保证直线与所有点接近

- 详细做法

- 若有n个样本点： $(x_1, y_1), \dots, (x_n, y_n)$ ，可以用下面的表达式来刻画这些点与直线 $y = a + bx$ 的接近程度：

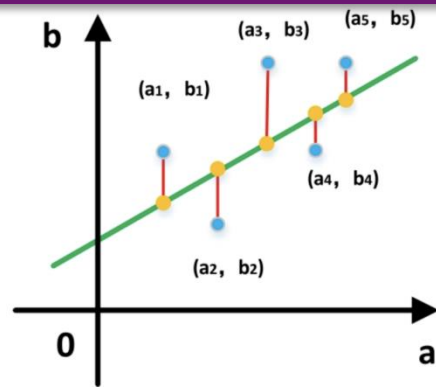
$$[y_1 - (a + bx_1)]^2 + \dots + [y_n - (a + bx_n)]^2$$

- 使上式达到最小值的直线 $y = a + bx$ 就是所求的直线，这种方法称为最小二乘法。

- 求a和b的偏导数，可得

如果用 \bar{x} 表示 $\frac{x_1 + x_2 + \dots + x_n}{n}$ ，用 \bar{y} 表示 $\frac{y_1 + y_2 + \dots + y_n}{n}$ 则可得到

$$b = \frac{x_1 y_1 + \dots + x_n y_n - n \bar{x} \bar{y}}{x_1^2 + \dots + x_n^2 - n \bar{x}^2}, a = \bar{y} - b \bar{x}$$



最小二乘法

例题1 从某大学中随机选出8名女大学生，其身高和体重数据如下表：

编号	1	2	3	4	5	6	7	8
身高	165	165	157	170	175	165	155	170
体重	48	57	50	54	64	61	43	59

求根据一名女大学生的身高预报她的体重的回归方程，并预报一名身高为172 c m的女大学生的体重。

填空题

$$b = \frac{x_1y_1 + \dots + x_ny_n - n\bar{x}\bar{y}}{x_1^2 + \dots + x_n^2 - n\bar{x}^2}, a = \bar{y} - b\bar{x}$$

编号	1	2	3	4	5	6	7	8
身高	165	165	157	170	175	165	155	170
体重	48	57	50	54	64	61	43	59

分析:

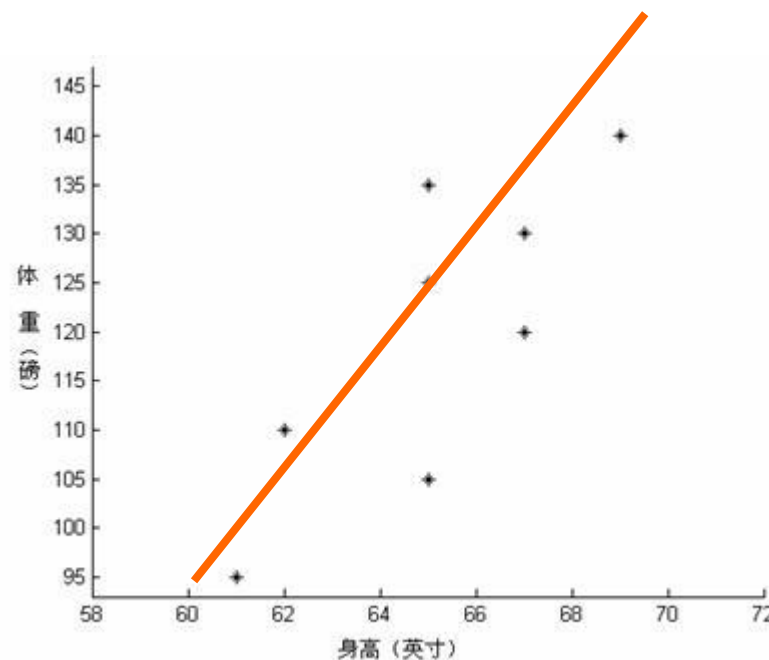
身高 y 为自变量

体重 x 为因变量

$$y = 0.849x - 85.172$$

身高 $172cm$ 女大学生体重

$$\hat{y} = 0.849 \times 172 - 85.712 = 60.316(kg)$$





南京大學
NANJING UNIVERSITY

目录

01

线性回归

02

优化求解

03

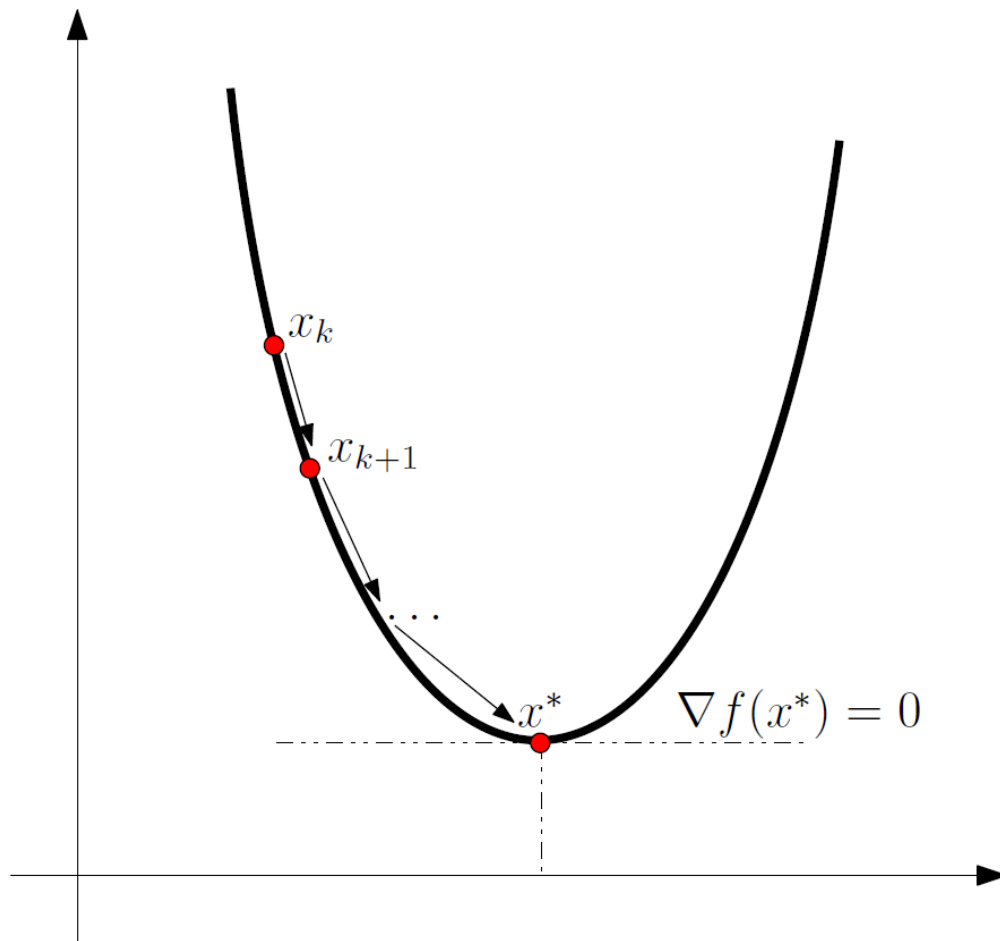
逻辑回归

03

决策树回归

优化求解 —— 梯度下降法

$$\min_x f(x)$$



优化求解 —— 梯度下降法

- 基本思想
 - 向着梯度的反方向调整
 - 步长不能太大，也不能太小

The diagram illustrates the gradient descent formula: $\Theta^1 = \Theta^0 - \alpha \nabla J(\Theta)$ evaluated at Θ^0 . Annotations include:

- current position**: points to Θ^0 (blue bubble)
- opposite direction**: points to the minus sign (black bubble)
- small step**: points to α (green bubble)
- direction of fastest increase**: points to $\nabla J(\Theta)$ (purple bubble)
- next position**: points to Θ^1 (red bubble)

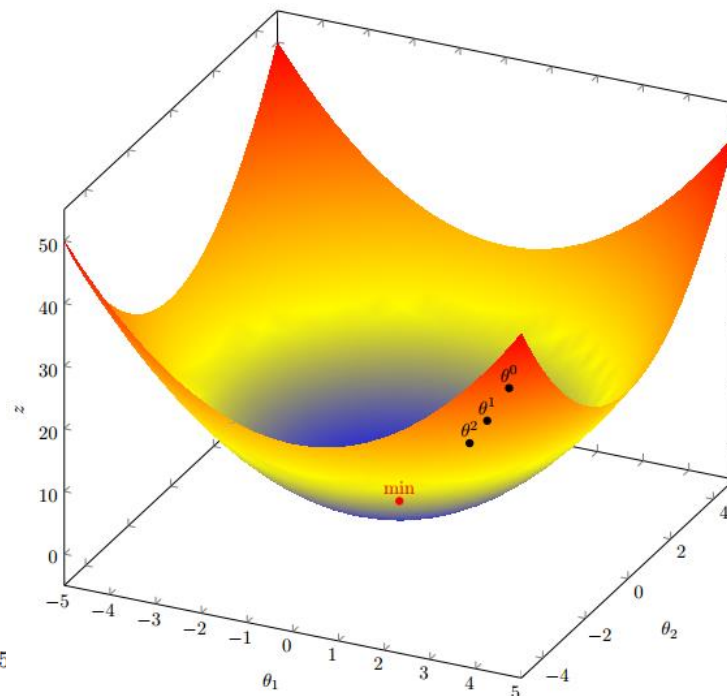
优化求解 —— 梯度下降法求解例子

- 原函数 $J(\Theta) = \theta_1^2 + \theta_2^2$.
- 函数的微分 $\nabla J(\Theta) = \langle 2\theta_1, 2\theta_2 \rangle$
- 初始条件 $\Theta^0 = (1, 3)$ $\alpha = 0.1$.
- 调整过程

$\Theta^1 = \Theta^0 - \alpha \nabla J(\Theta)$ evaluated at Θ^0

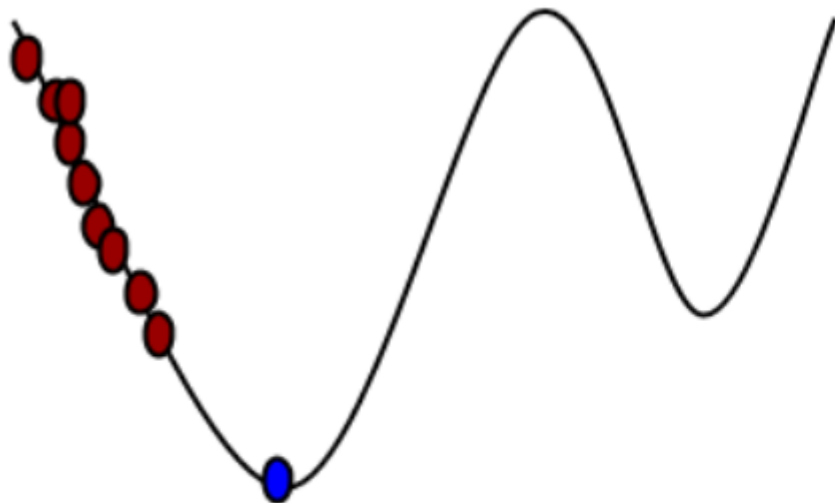
Annotations:
- current position: Θ^0
- next position: Θ^1
- opposite direction: $-\alpha \nabla J(\Theta)$
- small step: α
- direction of fastest increase: $\nabla J(\Theta)$

$$\begin{aligned}\Theta^0 &= (1, 3) \\ \Theta^1 &= \Theta^0 - \alpha \nabla J(\Theta) \\ &= (1, 3) - 0.1(2, 6) \\ &= (0.8, 2.4) \\ \Theta^2 &= (0.8, 2.4) - 0.1(1.6, 4.8) \\ &= (0.64, 1.92) \\ \Theta^3 &= (0.512, 1.536) \\ \Theta^4 &= (0.4096, 1.2288000000000001) \\ &\vdots \\ \Theta^{10} &= (0.10737418240000003, 0.32212254720000005) \\ &\vdots \\ \Theta^{50} &= (1.1417981541647683e^{-05}, 3.425394462494306e^{-05}) \\ &\vdots \\ \Theta^{100} &= (1.6296287810675902e^{-10}, 4.888886343202771e^{-10})\end{aligned}$$



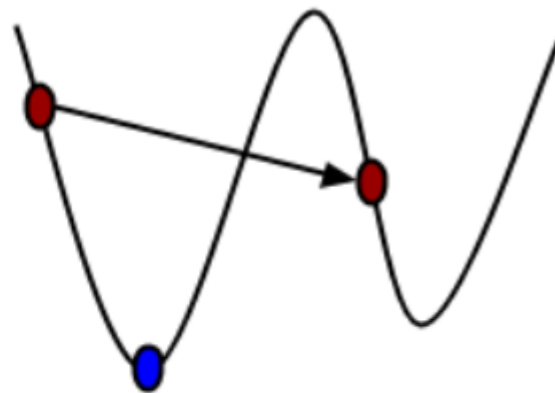
优化求解-学习速率的影响

学习率过小，
收敛速度太慢。



very small learning
rate needs lots of
steps

学习率过大，
不会收敛。



too big learning rate:
missed the minimum

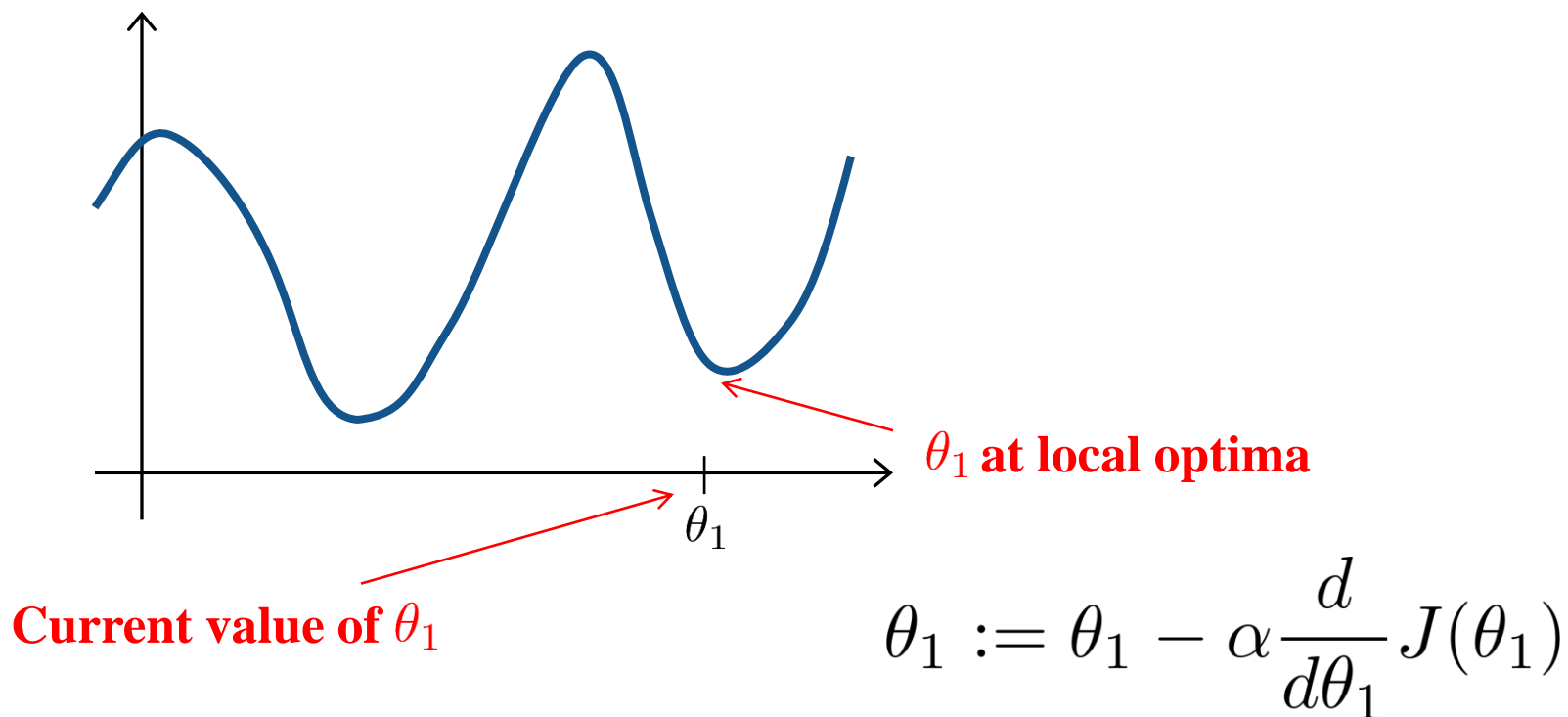
优化求解 —— 梯度下降法收敛

$$\Theta^1 = \Theta^0 - \alpha \nabla J(\Theta) \quad \text{evaluated at } \Theta^0$$

Diagram illustrating the gradient descent update formula with annotations:

- Θ^1 : next position
- Θ^0 : current position
- α : small step
- $\nabla J(\Theta)$: direction of fastest increase
- opposite direction

梯度下降法不能保证一定收敛到全局最优值



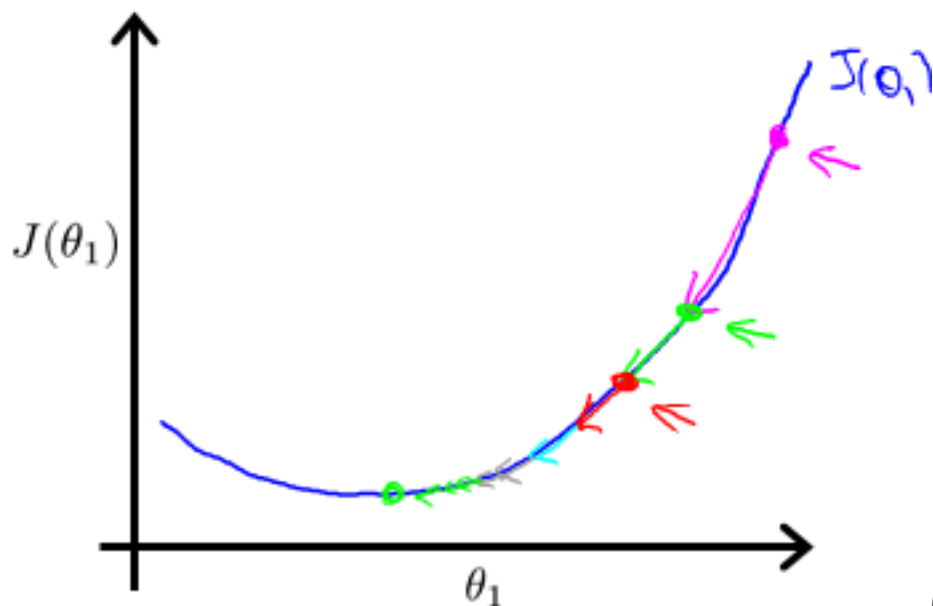
优化求解 —— 变种梯度下降法1

$$\Theta^1 = \Theta^0 - \alpha \nabla J(\Theta) \quad \text{evaluated at } \Theta^0$$

Annotations:

- current position: Θ^0
- next position: Θ^1
- small step: α
- opposite direction: $-\nabla J(\Theta)$
- direction of fastest increase: $\nabla J(\Theta)$

自动调整学习率的梯度下降法



学习率随梯度
大小调整

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

优化求解 —— 变种梯度下降法2

- 批处理梯度下降法(一批数据后更新权值)

- **优点:** 由全数据集确定的方向能够更好地代表样本总体, 从而更准确地朝向极值所在的方向、易并行
- **缺点:** 当样本数目 m 很大时, 每迭代一步都需要对所有样本计算, 训练过程会很慢

Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^m (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)} \quad (\text{for every } j).$$

}

- 随机梯度下降法 (每一个数据更新1次权值)

- **优点:** 每一轮参数更新快, 计算量小
- **缺点:** 准确度下降、稳定性弱、不易于并行

Loop {

for i=1 to m, {

$$\theta_j := \theta_j + \alpha (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)} \quad (\text{for every } j).$$

}

}



南京大學
NANJING UNIVERSITY

目录

01

线性回归

02

优化求解

03

逻辑回归

03

决策树回归

逻辑回归 —— 案例引入

- 案例引入

表1 年龄(Age)和冠心病(CD)发病情况

Age	CD	Age	CD	Age	CD
22	0	40	0	54	0
23	0	41	1	55	1
24	0	46	0	58	1
27	0	47	0	60	1
28	0	48	0	60	0
30	0	49	1	62	1
30	0	49	0	65	1
32	0	50	1	67	1
33	0	51	0	71	1
35	1	51	1	77	1
38	0	52	0	81	1

问题

◆ 比较病人和非病人的平均年龄

- 非病人: 38.6 岁
- 病人: 58.7 岁

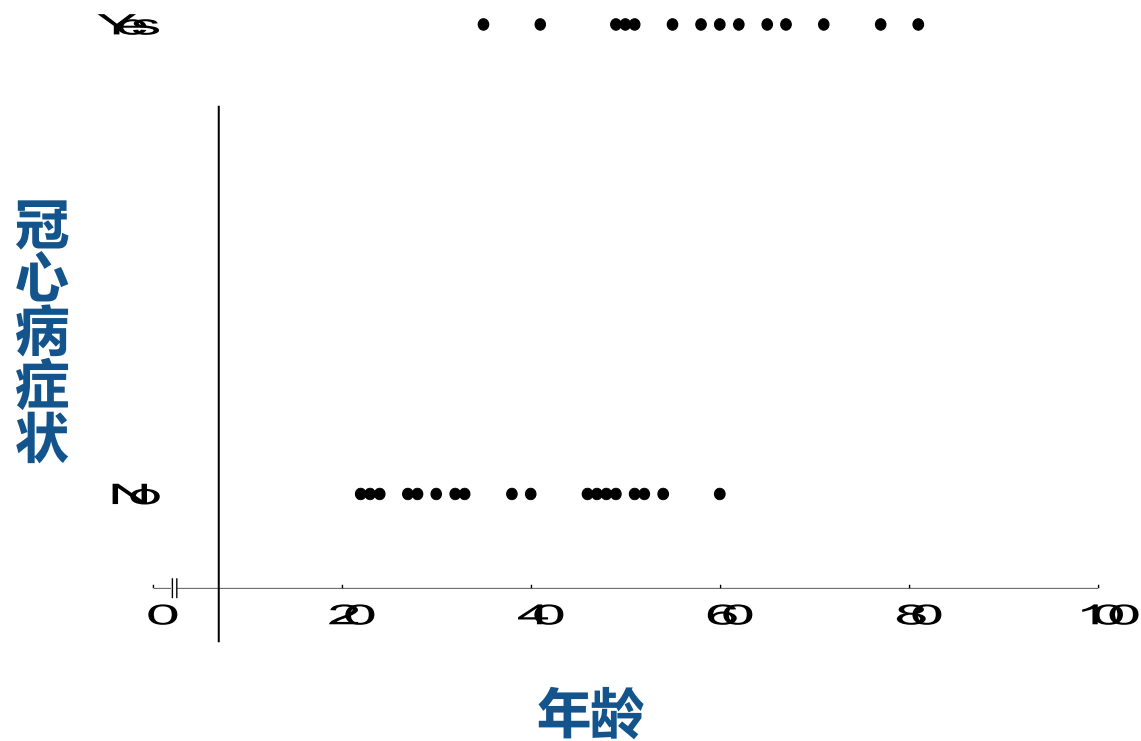
◆ 能不能用线形回归?



能



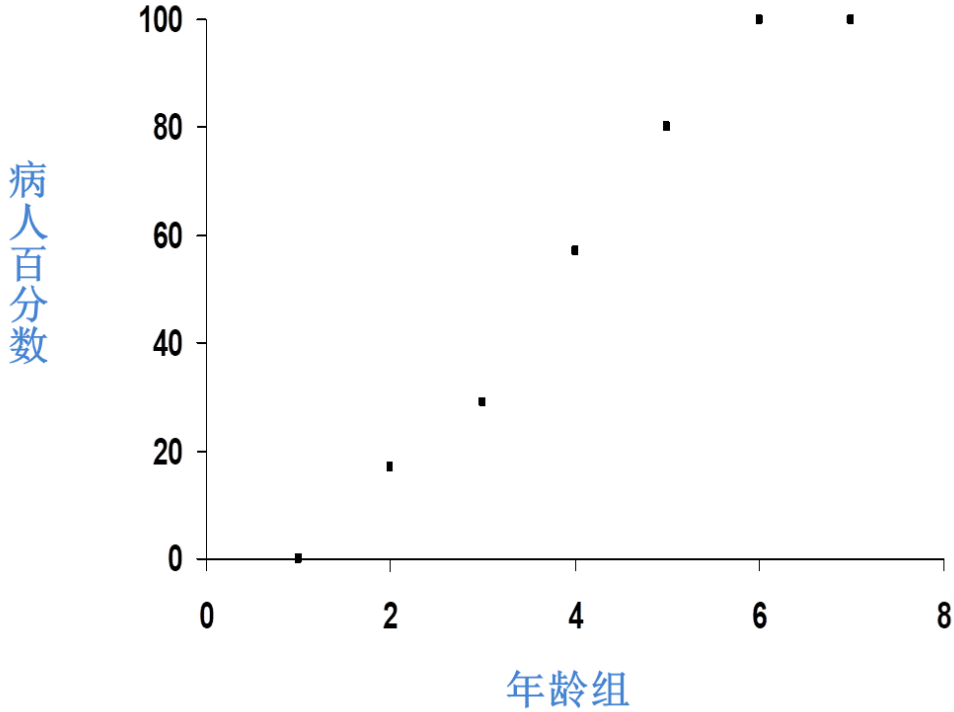
不能



案例引入

表2 按年龄组划分的冠心病发病情况

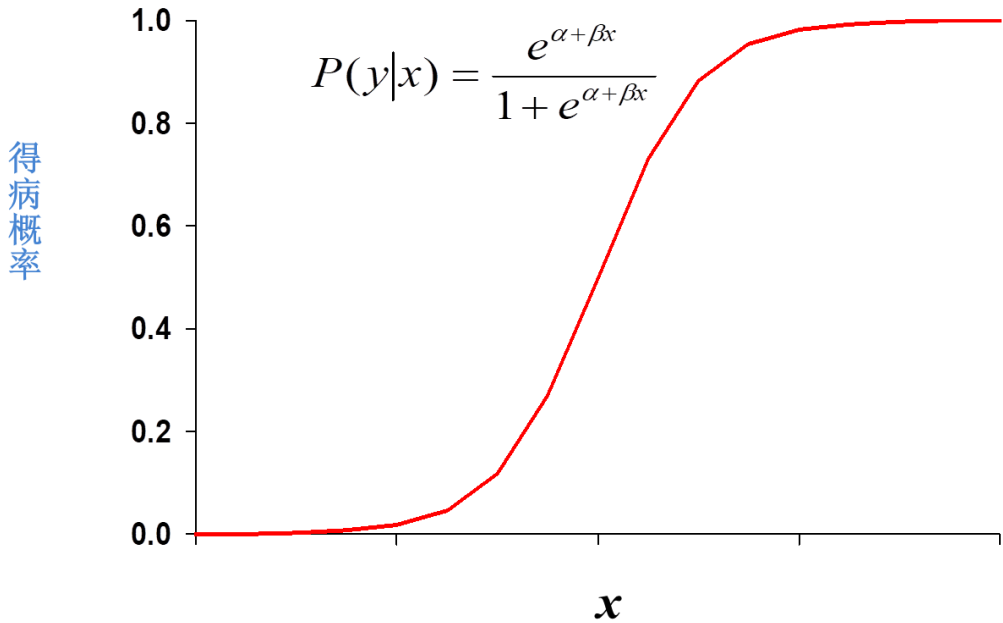
年龄组	人数	冠心病人数	累积%
20 - 29	5	0	0
30 - 39	6	1	17
40 - 49	7	2	29
50 - 59	7	4	57
60 - 69	5	4	80
70 - 79	2	2	100
80 - 89	1	1	100



案例引入

表2 按年龄组划分的冠心病发病情况

年龄组	人数	冠心病人数	累积%
20 - 29	5	0	0
30 - 39	6	1	17
40 - 49	7	2	29
50 - 59	7	4	57
60 - 69	5	4	80
70 - 79	2	2	100
80 - 89	1	1	100



逻辑回归函数

1838年由比利时数学家Verhulst首次提出。1920年美国学者 Bearl & Reed 在研究果蝇的繁殖中发现和使用该函数，并在人口估计和预测中推广使用

logistic函数的值域为 $[0,1]$

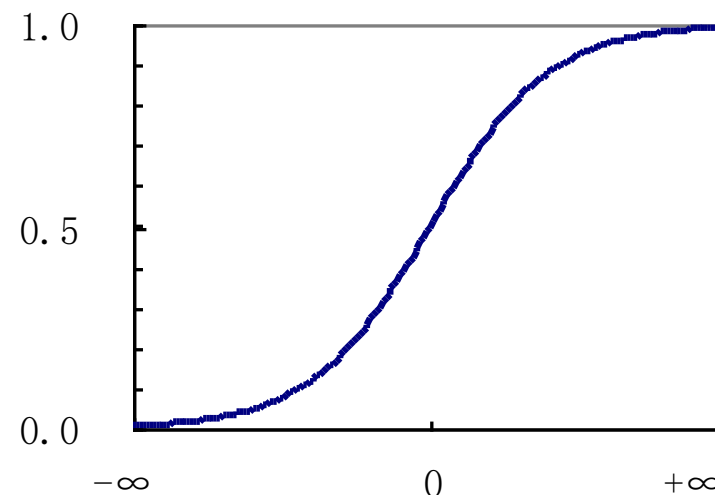
$$f(x) = \frac{e^x}{1 + e^x}$$

用 $p_i = P(y_i = 1 | x_{i1}, x_{i2}, \dots, x_{ip})$

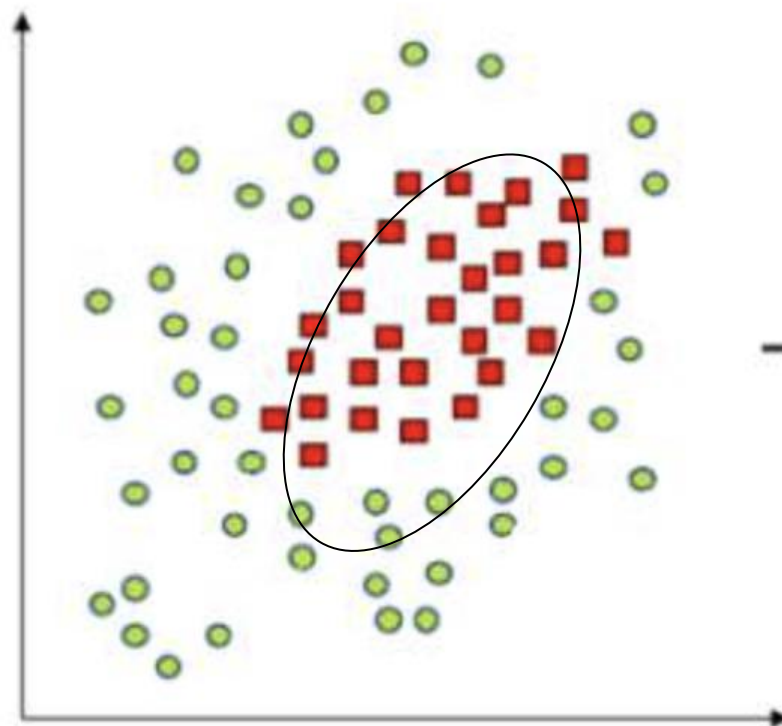
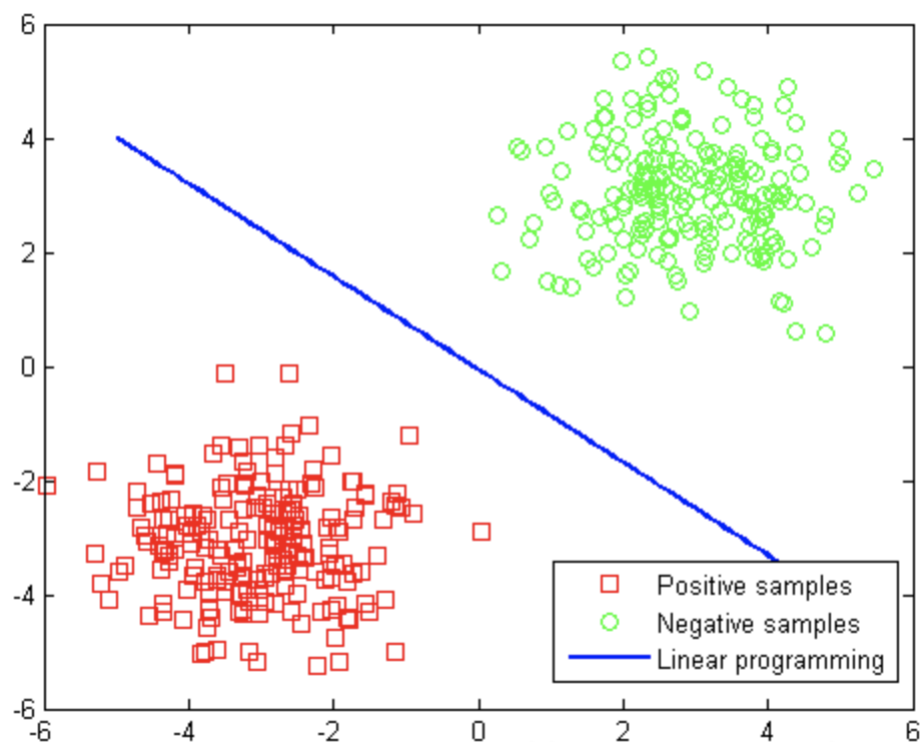
作为因变量，得到logistic回归模型

$$p_i = \frac{\exp(\alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip})}{1 + \exp(\alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip})}$$

$$\ln \frac{p_i}{1 - p_i} = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}$$

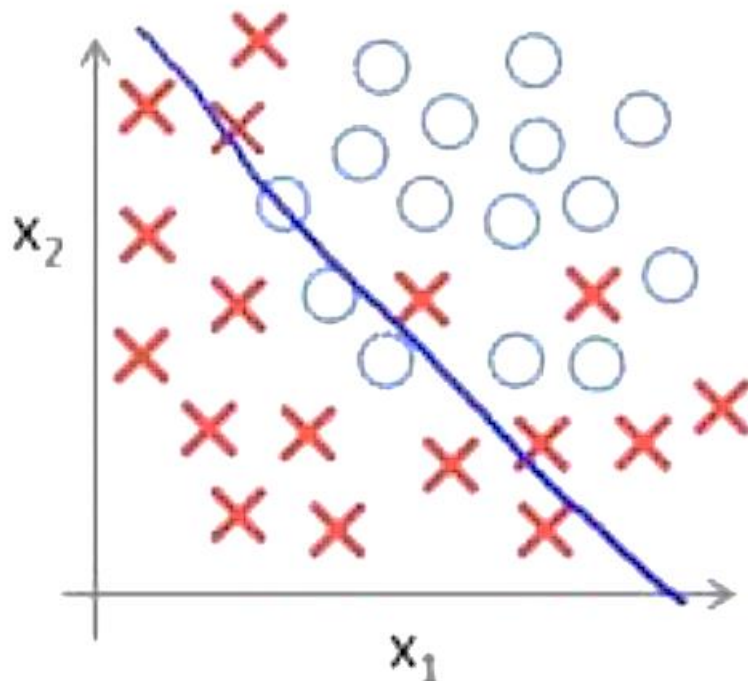


逻辑回归特点：线性分类器



逻辑回归特点：线性分类器

$$f(x) = \frac{e^x}{1 + e^x}$$



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

(g = sigmoid function)

$$p_i = \frac{\exp(\alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})}{1 + \exp(\alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})}$$

优势比OR

$$\ln \left(\frac{P}{1-P} \right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m = \text{logit} P$$

- 优势比 OR (odds ratio)

流行病学衡量危险因素作用大小的**比例**指标。

$$OR_j = \frac{P_1/(1-P_1)}{P_0/(1-P_0)} \quad \text{表示自变量变化以后, 发病概率的变化情况}$$

式中 P_1 和 P_0 分别表示在 X_j 取值为 c_1 及 c_0 时的发病概率, OR_j 称作多变量调整后的优势比, 表示扣除了其他自变量影响后危险因素的作用。

优势比OR

$$\ln \left(\frac{P}{1-P} \right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m = \text{logit} P$$

- 与 $\text{logistic} P$ 的关系：

对比某一危险因素两个不同暴露水平 $x_j=c_1$ 与 $x_j=c_0$ 的发病情况（假定其它因素的水平相同），其优势比的自然对数为：

$$\begin{aligned} \ln OR_j &= \ln \left[\frac{P_1/(1-P_1)}{P_0/(1-P_0)} \right] = \text{logit} P_1 - \text{logit} P_0 \\ &= (\beta_0 + \beta_j c_1 + \sum_{t \neq j}^m \beta_t X_t) - (\beta_0 + \beta_j c_0 + \sum_{t \neq j}^m \beta_t X_t) \\ &= \beta_j (c_1 - c_0) \end{aligned}$$

优势比OR

$$\begin{aligned}\ln OR_j &= \ln \left[\frac{P_1/(1-P_1)}{P_0/(1-P_0)} \right] = \text{logit}P_1 - \text{logit}P_0 \\ &= (\beta_0 + \beta_j c_1 + \sum_{t \neq j}^m \beta_t X_t) - (\beta_0 + \beta_j c_0 + \sum_{t \neq j}^m \beta_t X_t) \\ &= \beta_j (c_1 - c_0)\end{aligned}$$

即 $OR_j = \exp[\beta_j (c_1 - c_0)]$

若 $X_j = \begin{cases} 1 & \text{暴露} \\ 0 & \text{非暴露} \end{cases}, \quad c_1 - c_0 = 1,$

则有 OR_j

$$= \exp \beta_j, \quad \beta_j \begin{cases} = 0, OR_j = 1 & \text{无作用} \\ > 0, OR_j > 1 & \text{危险因子} \\ < 0, OR_j < 1 & \text{保护因子} \end{cases}$$

优势比OR

例子：在一个具有17个家庭的样本里，共有3家的收入为¥ 10000，5家的收入为¥ 11000，9家的收入为¥ 12000。在收入为¥ 10000的家庭里，1个主妇不工作，2个主妇工作；在收入为¥ 11000的家庭里，1个主妇不工作，4个主妇工作；在收入为¥ 12000的家庭里，1个主妇不工作，8个主妇工作。

主妇工作状态对家庭收入的影响

收入	主妇工作状态		总计
	0（不工作）	1（工作）	
10	1	2	3
11	1	4	5
12	1	8	9
总计	3	14	17

优势比OR

收入	主妇工作状况		工作概率P
	0 (不工作)	1 (工作)	
10	1	2	2/3
11	1	4	4/5
12	1	8	8/9

$$OR_j = \frac{P_1 / (1 - P_1)}{P_0 / (1 - P_0)}$$

X分别取10和11时, odd=4/2=2

X分别取12和11时, odd=8/4=2

优势比OR

收入	主妇工作状况		工作概率P
	0 (不工作)	1 (工作)	
10	1	2	2/3
11	1	4	4/5
12	1	8	8/9

$$OR_j = \frac{P_1/(1 - P_1)}{P_0/(1 - P_0)}$$

X分别取10和11时, odd=4/2=2 则有 $OR_j = \exp \beta_j$, β_j $\begin{cases} =0, OR_j =1 & \text{无作用} \\ >0, OR_j >1 & \text{危险因子} \\ <0, OR_j <1 & \text{保护因子} \end{cases}$

X分别取12和11时, odd=8/4=2

- 收入每增加1个单位, 主妇工作的Odds增加到原来的2倍
- 说明收入对工作状态有正关系, 收入越高, 工作概率越高
- 在疾病检测中, 说明一个因素越高, 使得疾病概率越高

优势比OR

$$\ln \left(\frac{P}{1-P} \right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m = \text{logit} P$$

模型参数的意义

常数项 β_0

表示暴露剂量为0时个体发病与不发病概率之比的自然对数。

回归系数 $\beta_j (j = 1, 2, \cdots, m)$

表示自变量 X_j 改变一个单位时logitP 的改变量。

logistic回归例子

例1 自变量是二值分类型变量

某医院为了研究导致手术切口感染的原因，收集了295例手术者情况，其中，手术时间小于或等于5小时的有242例，感染者13例；手术时间大于5小时的有53例，感染者7例。试建立手术切口感染(y)关于手术时间(x)的logistic回归模型。

y \ x	x	
	1 (>5小时)	0 (≤ 5小时)
1 (感染)	7	13
0 (未感染)	46	229
总和	53	242

逻辑回归参数估计

记得到一个实际观测值 $y_i (i = 1, 2, \dots, n)$ 的概率为

$$P(y_i) = p_i^{y_i} (1 - p_i)^{1-y_i} \quad \leftarrow \begin{aligned} p(y = 1|x; w) &= \phi(w^T x + b) = \phi(z) \\ p(y = 0|x; w) &= 1 - \phi(z) \\ P(y|x; \theta) &= (h_\theta(x))^y (1 - h_\theta(x))^{1-y} \end{aligned}$$

则似然函数为 $L = \prod_{i=1}^n P(y_i) = \prod_{i=1}^n p_i^{y_i} (1 - p_i)^{1-y_i}$

两边取对数: $\ln L = \sum_{i=1}^n [y_i \ln p_i + (1 - y_i) \ln(1 - p_i)] =$

$$\sum_{i=1}^n \left[y_i \ln \frac{p_i}{1 - p_i} + \ln(1 - p_i) \right]$$
$$\ln \left(\frac{P}{1 - P} \right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_m X_m = \text{logit} P$$

最后得到: $\ln L = \sum_{i=1}^n [y_i (\alpha + \beta x_i) - \ln(1 + \exp(\alpha + \beta x_i))]$

当使得 $\ln L$ 取得最大值时, $-\frac{1}{m} \ln L$ 最小时, 参数估计值即为所求。

逻辑回归参数估计


- 使用梯度下降方法，迭代求解参数

参数估计: $\alpha = -2.869$ ($p=0.0001$),

$\beta = 0.986$ ($p=0.0468$).

回归模型:

如何解释系数的实际意义?


$$p(y = 1 | x) = \frac{e^{-2.869 + 0.986 x}}{1 + e^{-2.869 + 0.986 x}}$$

逻辑回归参数解释

例如 手术感染问题

$y \backslash x$	1 (>5小时)	0 (≤ 5小时)
1 (感染)	7	13
0 (未感染)	46	229
总和	53	242

$$\text{即 } OR_j = \exp[\beta_j(c_1 - c_0)]$$

$$\text{若 } X_j = \begin{cases} 1 & \text{暴露} \\ 0 & \text{非暴露} \end{cases}, \quad c_1 - c_0 = 1,$$

$$\text{则有 } OR_j = \exp \beta_j, \quad \beta_j \begin{cases} = 0, OR_j = 1 & \text{无作用} \\ > 0, OR_j > 1 & \text{危险因子} \\ < 0, OR_j < 1 & \text{保护因子} \end{cases}$$

Logistic 回归模型:

$$p(y = 1 | x) = \frac{e^{-2.869 + 0.986 x}}{1 + e^{-2.869 + 0.986 x}}$$

从 $\beta=0.986$, 得到 $RR \approx OR = e^\beta = 2.681$ 。

所以, 手术时间大于5小时的感染率是手术时间小于或等于5小时的感染率的2.681倍, 即感染的可能性增加了186.1% 。

逻辑回归正则化

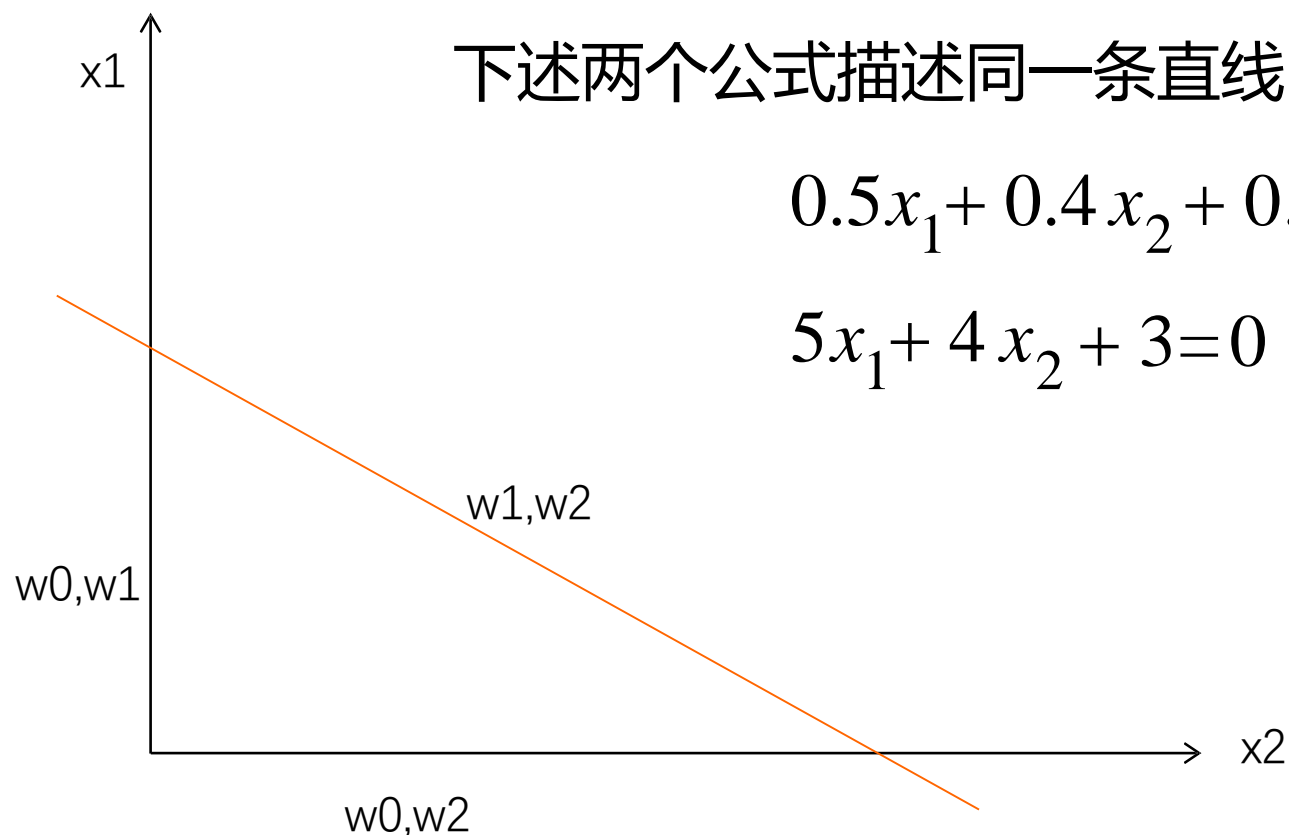
$$\text{Logit}(p_i) = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

$w_1 x_1 + w_2 x_2 + w_0 = 0$ 对应于平面的一根直线

下述两个公式描述同一条直线，哪个好？

$$0.5x_1 + 0.4x_2 + 0.3 = 0$$

$$5x_1 + 4x_2 + 3 = 0$$



W在数值上越小越好，这样越能抵抗数据的扰动

逻辑回归正则化

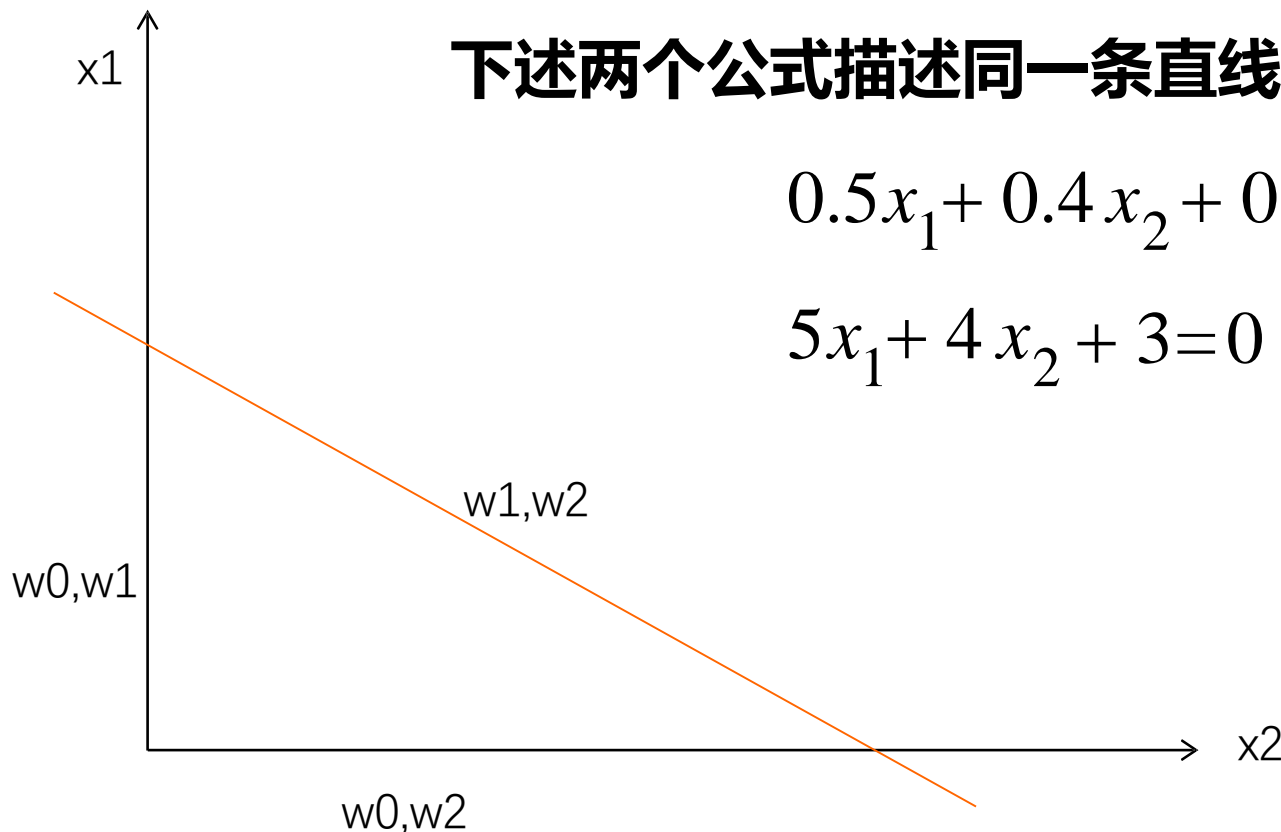
$$\text{Logit}(p_i) = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

$w_1 x_1 + w_2 x_2 + w_0 = 0$ 对应于平面的一根直线

下述两个公式描述同一条直线，哪个好？

$$0.5x_1 + 0.4x_2 + 0.3 = 0$$

$$5x_1 + 4x_2 + 3 = 0$$



正则化表达式

$$L_1 = \sum_{i=0}^m |w_i|$$

$$L_2 = \sum_{i=0}^m w_i^2$$

W在数值上越小越好，这样越能抵抗数据的扰动

逻辑回归正则化

平衡训练误差与泛化能力

$$E = \sum_{i=1}^n \left(y_i - \frac{1}{1 + e^{-(w_1 x_{i1} + w_2 x_{i2} + w_0)}} \right)^2 + \lambda \mathcal{L}_1$$
$$E = \sum_{i=1}^n \left(y_i - \frac{1}{1 + e^{-(w_1 x_{i1} + w_2 x_{i2} + w_0)}} \right)^2 + \lambda \mathcal{L}_2$$

正则化表达式

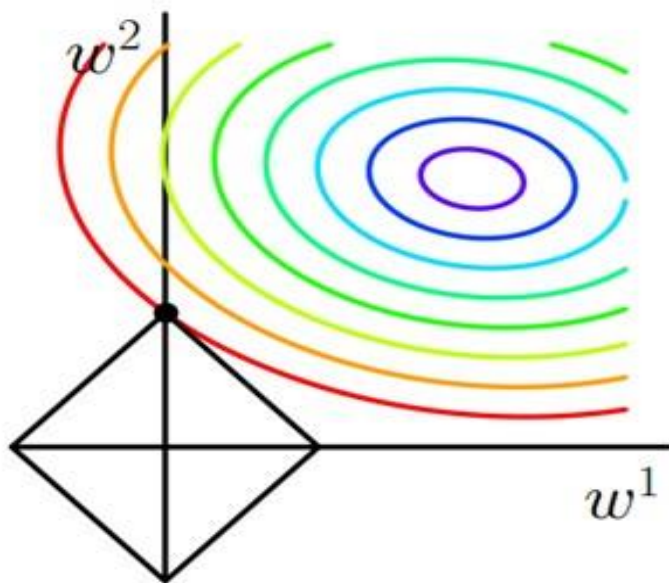
$$L_1 = \sum_{i=0}^m |w_i|$$

$$L_2 = \sum_{i=0}^m w_i^2$$

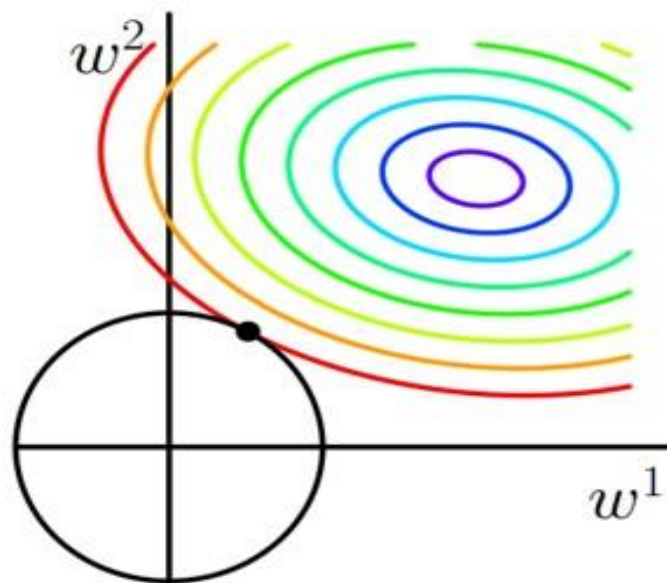
惩罚项：若学习到大权值使得误差小，但是再加上正则化式子以后使得上面E值变大。

因此，最小化E值使得求解的权值尽可能相对较小。

逻辑回归正则化



(a) ℓ_1 -ball meets quadratic function. ℓ_1 -ball has corners. It's very likely that the meet-point is at one of the corners.



(b) ℓ_2 -ball meets quadratic function. ℓ_2 -ball has no corner. It is very unlikely that the meet-point is on any of axes."

正则化表达式

$$L_1 = \sum_{i=0}^m |w_i|$$

$$L_2 = \sum_{i=0}^m w_i^2$$

逻辑回归正则化

一个有趣的结论

L_1 倾向于使得 w 要么取1, 要么取0**稀疏编码**

L_2 倾向于使得 w 整体偏小**岭回归**

正则化表达式

$$L_1 = \sum_{i=0}^m |w_i|$$

$$L_2 = \sum_{i=0}^m w_i^2$$

逻辑回归正则化

适用场景：

L_1 适合挑选特征

L_2 也称为岭回归，有很强的概率意义

正则化表达式

$$L_1 = \sum_{i=0}^m |w_i|$$

$$L_2 = \sum_{i=0}^m w_i^2$$

逻辑回归训练方法优化

梯度下降法的选择

1. SGD

2. L-BFGS

	数值归一化	正则化	梯度下降法	分类个数	数据选择
LogisticRegression With L-BFGS	需要均值归一化，算法融入方差归一化	支持L2	L-BFGS (收敛快，考虑二阶导数)	支持多分类	加载所有数据都参与训练
<i>LogisticRegression With SGD</i>	不归一化，需要专门在外面进行归一化	支持L1, L2	SGD	不支持多分类	随机从训练集选取 (支持 MiniBatch Fraction)



南京大學
NANJING UNIVERSITY

目录

01

线性回归

02

优化求解

03

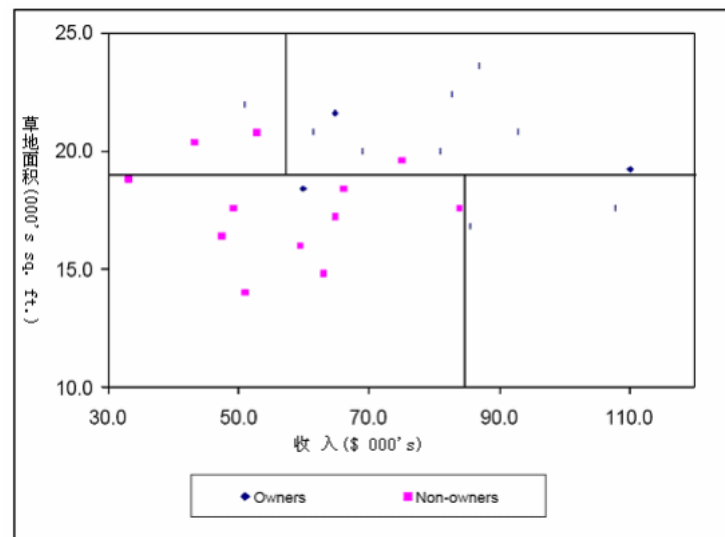
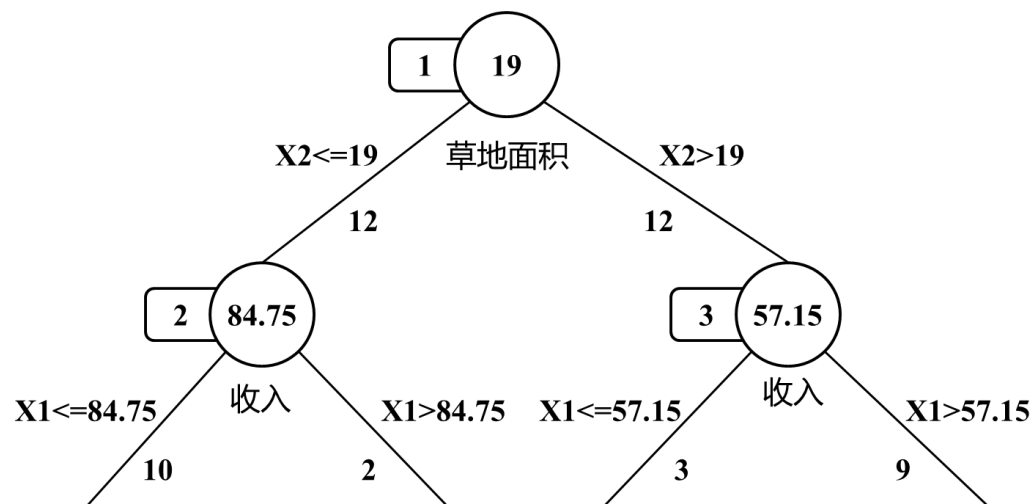
逻辑回归

04

决策树回归

决策树回归

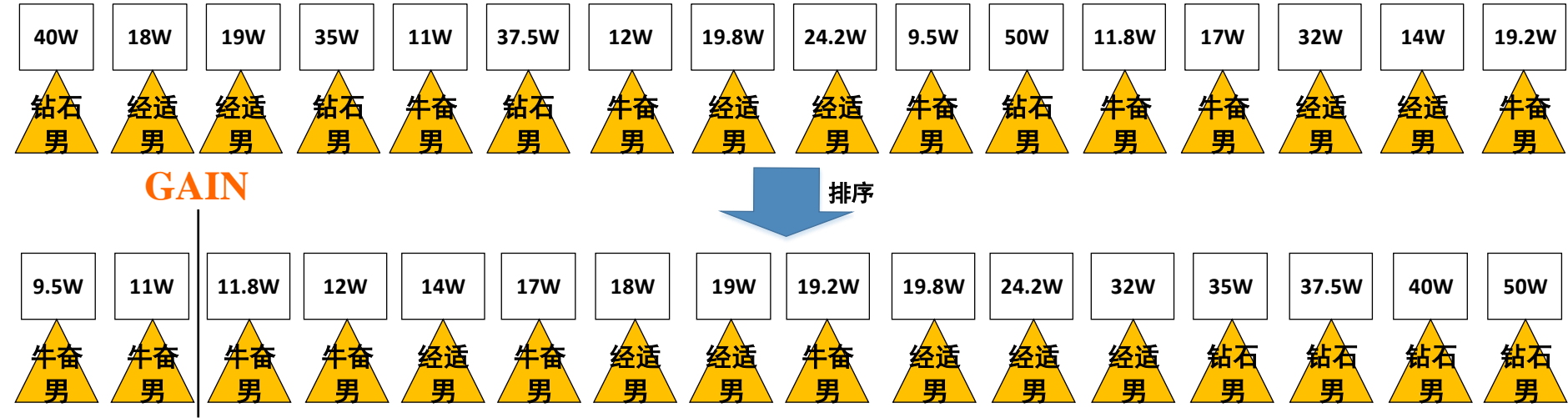
- 决策树是将空间用超平面进行划分的一种方法，每次分割的时候，都将当前的空间一分为二，这样使得每一个叶子节点都是在空间中的一个不相交的区域，在进行决策的时候，会根据输入样本每一维feature的值，一步一步往下，最后使得样本落入N个区域中的一个（假设有N个叶子节点），如下图所示。



- 既然是决策树，那么必然会在以下两个核心问题：如何选择划分点？如何决定叶节点的输出值？——
决策树分类选择划分点，使得信息增益最大，叶节点输出即类别
- 一个回归树对应着输入空间（即特征空间）的一个划分以及在划分单元上的输出值。分类树中**采用信息增益**等方法，通过计算选择最佳划分点。**而在回归树中，采用的是启发式的方法。**

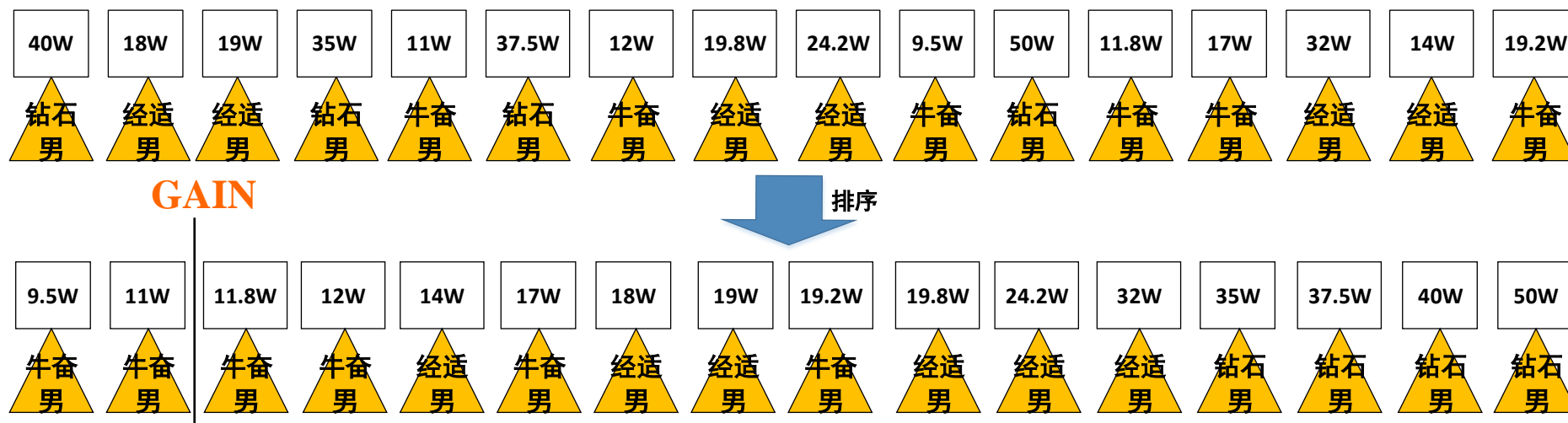
决策树分类最佳划分点选择

- 分割区间的策略
 - 从最小值开始建立分割区间，开始计算各自的信息增益，选择信息增益最大的一个分割区间作为最佳划分点



决策树分类最佳划分点选择

- 分割区间的策略
 - 从最小值开始建立分割区间，开始计算各自的信息增益，选择**信息增益最大**的一个分割区间作为最佳划分点



- 假如n个特征，每个特征有 s_i ($i \in (1, n)$)个取值，则遍历所有特征，尝试该特征所有取值，对空间进行划分，直到**取到特征j的取值s**，使得**损失函数最小**，这样就得到了一个划分点。

$$\min_{j, s} [\min_{c_1} Loss(y_i, c_1) + \min_{c_2} Loss(y_i, c_2)]$$

其中一个特征损失函数最小值

决策树回归 —— 例子

- X的取值范围[0.5, 10.5],y的取值范围: [5.0,10.10],用树桩做基函数;

x_i	1	2	3	4	5	6	7	8	9	10
y_i	5.56	5.70	5.91	6.40	6.80	7.05	8.90	8.70	9.00	9.05

- 求f1(x)回归树T1(x),
$$\min_s \left[\min_{c_1} \sum_{x_i \in R_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2} (y_i - c_2)^2 \right]$$
- 求切分点s: $R_1 = \{x | x \leq s\}$, $R_2 = \{x | x > s\}$

决策树回归 —— 例子

- X的取值范围[0.5, 10.5],y的取值范围: [5.0,10.10],用树桩做基函数;

x_i	1	2	3	4	5	6	7	8	9	10
y_i	5.56	5.70	5.91	6.40	6.80	7.05	8.90	8.70	9.00	9.05

- 求 $f_1(x)$ 回归树 $T_1(x)$,
$$\min_s \left[\min_{c_1} \sum_{x_i \in R_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2} (y_i - c_2)^2 \right]$$

- 求切分点 s : $R_1 = \{x | x \leq s\}$, $R_2 = \{x | x > s\}$

- 解题过程: 各切分点:

1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5

决策树回归 —— 例子

- X的取值范围[0.5, 10.5],y的取值范围: [5.0,10.10],用树桩做基函数;

x_i	1	2	3	4	5	6	7	8	9	10
y_i	5.56	5.70	5.91	6.40	6.80	7.05	8.90	8.70	9.00	9.05

- 求 $f_1(x)$ 回归树 $T_1(x)$,
$$\min_s \left[\min_{c_1} \sum_{x_i \in R_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2} (y_i - c_2)^2 \right]$$

- 求切分点s: $R_1 = \{x | x \leq s\}$, $R_2 = \{x | x > s\}$
- 求在 R_1, R_2 内部使平方损失误差达到最小值的 c_1, c_2 :

$$c_1 = \frac{1}{N_1} \sum_{x_i \in R_1} y_i, \quad c_2 = \frac{1}{N_2} \sum_{x_i \in R_2} y_i$$

- 解题过程: 各切分点:

1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5

决策树回归 —— 例子

- X的取值范围[0.5, 10.5],y的取值范围: [5.0,10.10],用树桩做基函数;

x_i	1	2	3	4	5	6	7	8	9	10
y_i	5.56	5.70	5.91	6.40	6.80	7.05	8.90	8.70	9.00	9.05

- 求 $f_1(x)$ 回归树 $T_1(x)$,
$$\min_s \left[\min_{c_1} \sum_{x_i \in R_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2} (y_i - c_2)^2 \right]$$

- 求切分点 s : $R_1 = \{x | x \leq s\}$, $R_2 = \{x | x > s\}$
- 求在 R_1, R_2 内部使平方损失误差达到最小值的 c_1, c_2 :

$$c_1 = \frac{1}{N_1} \sum_{x_i \in R_1} y_i, \quad c_2 = \frac{1}{N_2} \sum_{x_i \in R_2} y_i$$

- 解题过程: 各切分点:

$$1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5$$

$$m(s) = \min_{c_1} \sum_{x_i \in R_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2} (y_i - c_2)^2$$

当 $s = 1.5$ 时, $R_1 = \{1\}$, $R_2 = \{2, 3, \dots, 10\}$, $c_1 = 5.56$, $c_2 = 7.50$,

$$m(s) = \min_{c_1} \sum_{x_i \in R_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2} (y_i - c_2)^2 = 0 + 15.72 = 15.72$$

决策树回归 —— 例子

- 全部:

s	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
$m(s)$	15.72	12.07	8.36	5.78	3.91	1.93	8.01	11.73	15.74

回归树 T_1

$$T_1(x) = \begin{cases} 6.24, & x < 6.5 \\ 8.91, & x \geq 6.5 \end{cases} \quad c_1 = \frac{1}{N_1} \sum_{x_i \in R_1} y_i, \quad c_2 = \frac{1}{N_2} \sum_{x_i \in R_2} y_i$$
$$f_1(x) = T_1(x)$$

决策树回归 —— 提升树

- 全部:

s	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
$m(s)$	15.72	12.07	8.36	5.78	3.91	1.93	8.01	11.73	15.74

回归树 T_1

$$T_1(x) = \begin{cases} 6.24, & x < 6.5 \\ 8.91, & x \geq 6.5 \end{cases}$$

$$f_1(x) = T_1(x)$$

x_i	1	2	3	4	5	6	7	8	9	10
y_i	5.56	5.70	5.91	6.40	6.80	7.05	8.90	8.70	9.00	9.05

计算
数据
残差

$$r_{2i} = y_i - f_1(x_i)$$

x_i	1	2	3	4	5	6	7	8	9	10
r_{2i}	-0.68	-0.54	-0.33	0.16	0.56	0.81	-0.01	-0.21	0.09	0.14

用 f_1 拟合数据的平方误差:

$$L(y, f_1(x)) = \sum_{i=1}^{10} (y_i - f_1(x_i))^2 = 1.93$$

第二步: 求 T_2 ,

$$T_2(x) = \begin{cases} -0.52, & x < 3.5 \\ 0.22, & x \geq 3.5 \end{cases}$$

$$f_2(x) = f_1(x) + T_2(x) = \begin{cases} 5.72, & x < 3.5 \\ 6.46, & 3.5 \leq x < 6.5 \\ 9.13, & x \geq 6.5 \end{cases}$$

$$L(y, f_2(x)) = \sum_{i=1}^{10} (y_i - f_2(x_i))^2 = 0.79$$

决策树回归 —— 提升树

- 全部:

s	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
$m(s)$	15.72	12.07	8.36	5.78	3.91	1.93	8.01	11.73	15.74

回归树 T_1

$$T_1(x) = \begin{cases} 6.24, & x < 6.5 \\ 8.91, & x \geq 6.5 \end{cases}$$

$$f_1(x) = T_1(x)$$

$$r_{2i} = y_i - f_1(x_i)$$

x_i	1	2	3	4	5	6	7	8	9	10
r_{2i}	-0.68	-0.54	-0.33	0.16	0.56	0.81	-0.01	-0.21	0.09	0.14

用 f_1 拟合数据的平方误差:

$$L(y, f_1(x)) = \sum_{i=1}^{10} (y_i - f_1(x_i))^2 = 1.93$$

第二步: 求 T_2 ,

$$T_2(x) = \begin{cases} -0.52, & x < 3.5 \\ 0.22, & x \geq 3.5 \end{cases}$$

$$f_2(x) = f_1(x) + T_2(x) = \begin{cases} 5.72, & x < 3.5 \\ 6.46, & 3.5 \leq x < 6.5 \\ 9.13, & x \geq 6.5 \end{cases}$$

$$L(y, f_2(x)) = \sum_{i=1}^{10} (y_i - f_2(x_i))^2 = 0.79$$

决策树回归 —— 提升树

- 全部:

s	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
$m(s)$	15.72	12.07	8.36	5.78	3.91	1.93	8.01	11.73	15.74

回归树 T_1

$$T_1(x) = \begin{cases} 6.24, & x < 6.5 \\ 8.91, & x \geq 6.5 \end{cases}$$
$$f_1(x) = T_1(x)$$

每一次进行回归树生成时采用的训练数据 r 都是上次预测结果 $f_m(x)$ 与训练数据值 y_i 之间的残差。这个残差会逐渐的减小。

$$r_{2i} = y_i - f_1(x_i)$$

x_i	1	2	3	4	5	6	7	8	9	10
r_{2i}	-0.68	-0.54	-0.33	0.16	0.56	0.81	-0.01	-0.21	0.09	0.14

用 f_1 拟合数据的平方误差:

$$L(y, f_1(x)) = \sum_{i=1}^{10} (y_i - f_1(x_i))^2 = 1.93$$

第二步: 求 T_2 ,

$$T_2(x) = \begin{cases} -0.52, & x < 3.5 \\ 0.22, & x \geq 3.5 \end{cases}$$
$$f_2(x) = f_1(x) + T_2(x) = \begin{cases} 5.72, & x < 3.5 \\ 6.46, & 3.5 \leq x < 6.5 \\ 9.13, & x \geq 6.5 \end{cases}$$
$$L(y, f_2(x)) = \sum_{i=1}^{10} (y_i - f_2(x_i))^2 = 0.79$$

决策树回归 —— 提升树

- 则接下来

$$T_3(x) = \begin{cases} 0.15, & x < 6.5 \\ -0.22, & x \geq 6.5 \end{cases} \quad L(y, f_3(x)) = 0.47$$

$$T_4(x) = \begin{cases} -0.16, & x < 4.5 \\ 0.11, & x \geq 4.5 \end{cases} \quad L(y, f_4(x)) = 0.30$$

$$T_5(x) = \begin{cases} 0.07, & x < 6.5 \\ -0.11, & x \geq 6.5 \end{cases} \quad L(y, f_5(x)) = 0.23$$

$$T_6(x) = \begin{cases} -0.15, & x < 2.5 \\ 0.04, & x \geq 2.5 \end{cases}$$

$$f_6(x) = f_5(x) + T_6(x) = T_1(x) + \dots + T_5(x) + T_6(x)$$

$$= \begin{cases} 5.63, & x < 2.5 \\ 5.82, & 2.5 \leq x < 3.5 \\ 6.56, & 3.5 \leq x < 4.5 \\ 6.83, & 4.5 \leq x < 6.5 \\ 8.95, & x \geq 6.5 \end{cases}$$

$$L(y, f_6(x)) = \sum_{i=1}^{10} (y_i - f_6(x_i))^2 = 0.17$$

- 此时已满足误差要求，则那么 $f(x)=f_6(x)$ 即为所求**提升树**。

决策树回归 —— 提升树

- 则接下来

$$T_3(x) = \begin{cases} 0.15, & x < 6.5 \\ -0.22, & x \geq 6.5 \end{cases} \quad L(y, f_3(x)) = 0.47$$

$$T_4(x) = \begin{cases} -0.16, & x < 4.5 \\ 0.11, & x \geq 4.5 \end{cases} \quad L(y, f_4(x)) = 0.30$$

$$T_5(x) = \begin{cases} 0.07, & x < 6.5 \\ -0.11, & x \geq 6.5 \end{cases} \quad L(y, f_5(x)) = 0.23$$

$$T_6(x) = \begin{cases} -0.15, & x < 2.5 \\ 0.04, & x \geq 2.5 \end{cases}$$

$$f_6(x) = f_5(x) + T_6(x) = T_1(x) + \dots + T_5(x) + T_6(x)$$

$$= \begin{cases} 5.63, & x < 2.5 \\ 5.82, & 2.5 \leq x < 3.5 \\ 6.56, & 3.5 \leq x < 4.5 \\ 6.83, & 4.5 \leq x < 6.5 \\ 8.95, & x \geq 6.5 \end{cases}$$

$$L(y, f_6(x)) = \sum_{i=1}^{10} (y_i - f_6(x_i))^2 = 0.17$$

- 此时已满足误差要求，则那么 $f(x)=f_6(x)$ 即为所求提升树。

x_i	1	2	3	4	5	6	7	8	9	10
y_i	5.56	5.70	5.91	6.40	6.80	7.05	8.90	8.70	9.00	9.05

对于训练集以为的新数据，可以实现回归预测

决策树回归算法

- 每一次进行回归树生成时采用的训练数据r都是**上次预测结果 $f_{m-1}(x)$ 与训练数据值 y_i 之间的残差**。这个残差会逐渐的减小。

- 算法流程:

1. 初始化 $f_0(x) = 0$;

2. 对于 $m=1,2,\dots,M$

a.按照 $r = y_i - f_{m-1}(x)$ 计算残差作为新的训练数据的 y

b.拟合残差 r 学习一颗回归树, 得到这一轮的回归树 $T(x; \Theta_m)$

c.更新

$$f_M(x) = \sum_{m=1}^M T(x; \Theta_m)$$

3. 得到回归提升树:

$$f_m(x) = f_{m-1}(x) + T(x; \Theta_m)$$

a步	$r_{2i} = y_i - f_1(x_i)$									
x_i	1	2	3	4	5	6	7	8	9	10
r_{2i}	-0.68	-0.54	-0.33	0.16	0.56	0.81	-0.01	-0.21	0.09	0.14

用 f_1 拟合数据的平方误差:

$$L(y, f_1(x)) = \sum_{i=1}^{10} (y_i - f_1(x_i))^2 = 1.93$$

第二步: 求 T_2 ,

$$T_2(x) = \begin{cases} -0.52, & x < 3.5 \\ 0.22, & x \geq 3.5 \end{cases} \quad \text{b步}$$

c步

$$f_2(x) = f_1(x) + T_2(x) = \begin{cases} 5.72, & x < 3.5 \\ 6.46, & 3.5 \leq x < 6.5 \\ 9.13, & x \geq 6.5 \end{cases}$$

$$L(y, f_2(x)) = \sum_{i=1}^{10} (y_i - f_2(x_i))^2 = 0.79$$

$$T_6(x) = \begin{cases} -0.15, & x < 2.5 \\ 0.04, & x \geq 2.5 \end{cases}$$

均方误差变小

$$f_6(x) = f_5(x) + T_6(x) = T_1(x) + \dots + T_5(x) + T_6(x)$$

$$= \begin{cases} 5.63, & x < 2.5 \\ 5.82, & 2.5 \leq x < 3.5 \\ 6.56, & 3.5 \leq x < 4.5 \\ 6.83, & 4.5 \leq x < 6.5 \\ 8.95, & x \geq 6.5 \end{cases} \quad \text{(3) 步}$$

$$L(y, f_6(x)) = \sum_{i=1}^{10} (y_i - f_6(x_i))^2 = 0.17$$

均方误差变小

小结

- 线性回归
 - 连续型因变量-数值预测
- 逻辑回归
 - 分类型因变量-类别预测
- 决策树回归-数值预测
 - 连续型因变量, 非线性预测