线性回归

目录

Contents

学习目标

Learning Objectives

1. 理解线性回归是什么?
2. 知道一元线性回归和多元线性回归的区别
3. 知道线性回归的应用场景

线性回归概念 - 举个栗子

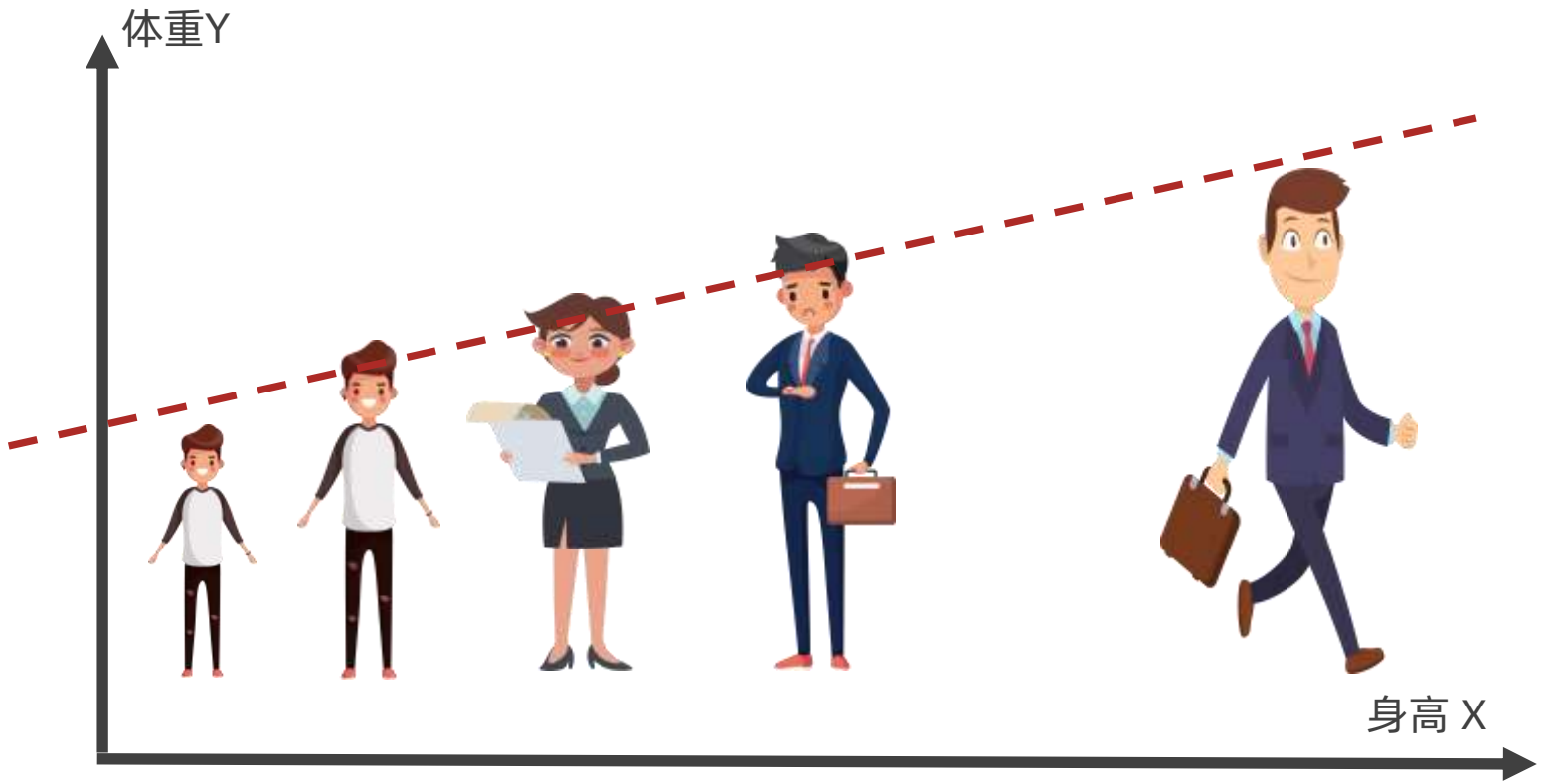
- 假若有了身高和体重数据，来了播仔的身高，你能预测播仔体重吗？

编号	身高	体重
1	160	56.3
2	166	60.6
3	172	65.1
4	174	68.5
5	180	75
6	176	?

- 这样的问题是回归问题，该如何求解呢？

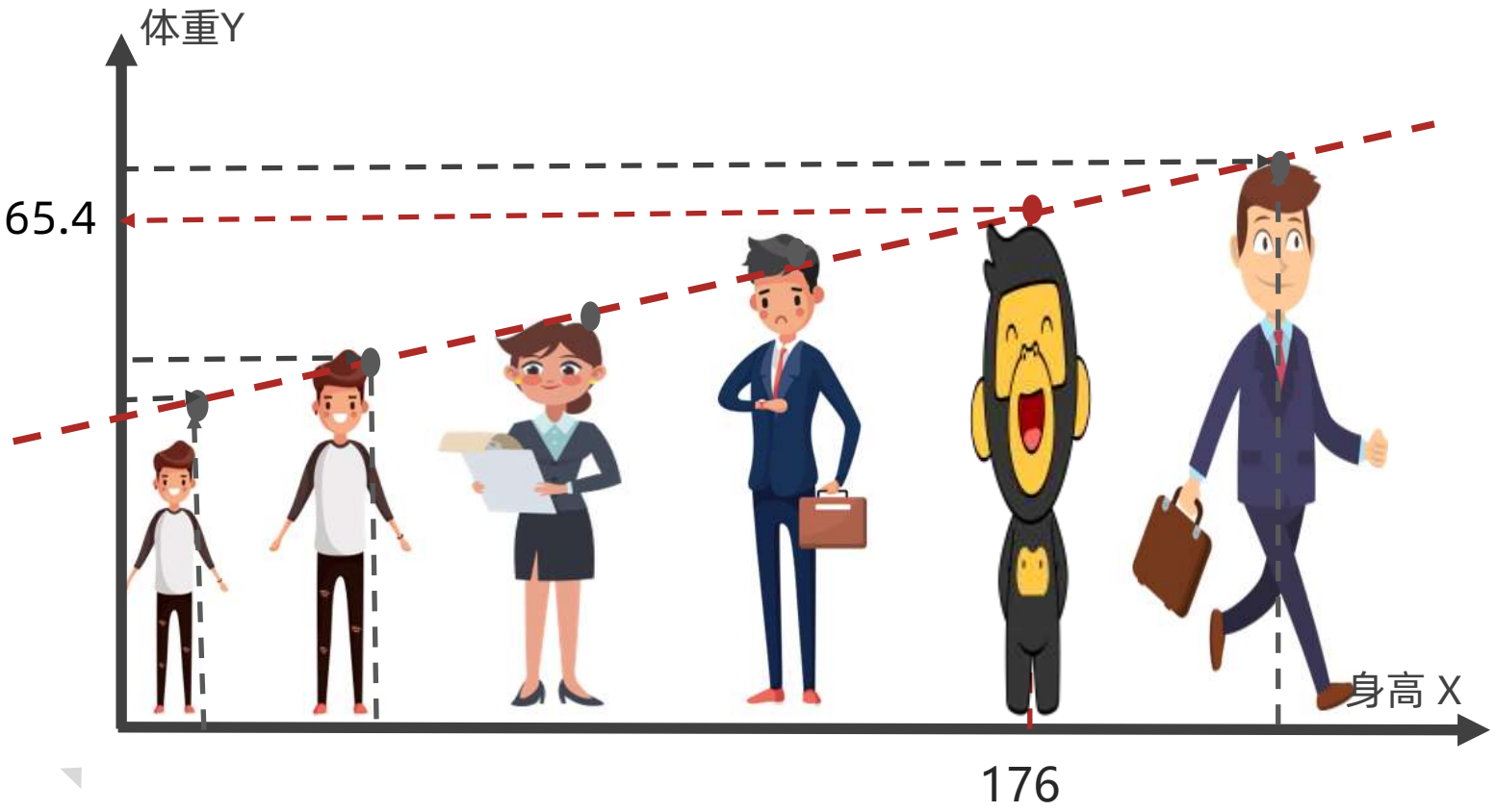
线性回归概念 - 举个栗子

- 思路：先从已知身高X和体重Y中找规律，再预测



线性回归概念 - 举个栗子

- 数学问题：用一条线来拟合身高和体重之间的关系，再对新数据进行预测



编号	身高	体重
1	160	56.3
2	166	60.6
3	172	65.1
4	174	68.5
5	180	75
6	176	?

方程 Y = kX + b

k160 + b = 56.3 -- (1)

k166 + b = 60.6 -- (2)

。。。。

k: 斜率 b:截距

若：y = 0.9 x + (-93)

0.9 * 176 + (-93) = 65.4

线性回归概念

- 定义　**利用回归方程(函数)** 对 **一个或多个自变量(特征值)和因变量(目标值)之间** 关系进行建模的一种分析方式。


房子面积

房子位置

房子楼层

房子朝向

房子价格

- 数学公式：　$h_{(w)} = w_1x_1 + w_2x_2 + w_3x_3 + \ldots + b = w^T\mathrm{x} + b$

其中 w为：$\begin{pmatrix} \mathrm{b} \\ \mathrm{w}_1 \\ \mathrm{w}_2 \\ \vdots \end{pmatrix}$，x为$\begin{pmatrix} 1 \\ \mathrm{x}_1 \\ \mathrm{x}_2 \\ \vdots \end{pmatrix}$　根据矩阵运算：$w^T\mathrm{x}$ 为 $[\mathrm{b}, w_1\ w_2\ \cdots] @ \begin{pmatrix} 1 \\ \mathrm{x}_1 \\ \mathrm{x}_2 \\ \vdots \end{pmatrix}$

线性回归概念

- 线性回归分类
  - 一元线性回归：y = wx +b
  - 目标值只与一个因变量有关系

编号	身高	体重
1	160	56.3
2	166	60.6
3	172	65.1
4	174	68.5
5	180	75
6	176	?

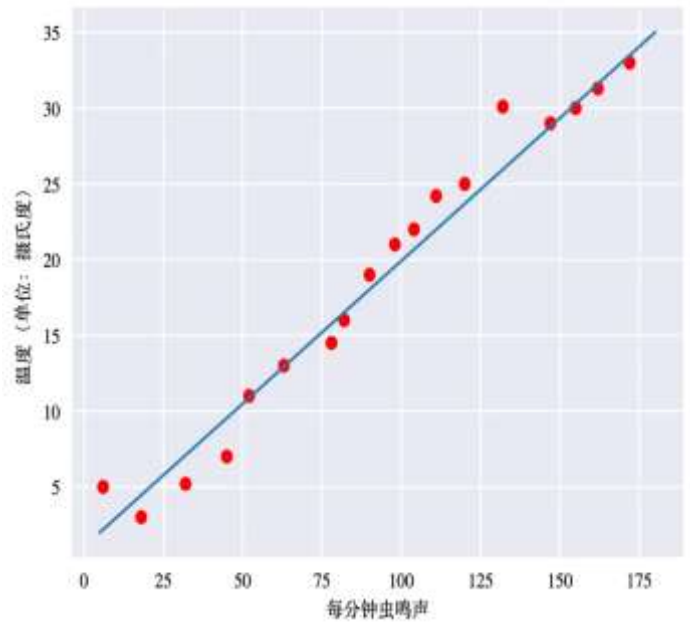
  - 多元线性回归：$y = w_1x_1 + w_2x_2 + w_3x_3 + \ldots + b$
  - 目标值只与多个因变量有关系

	房子面积	房子位置	房子楼层	房子朝向	房子价格
数据1	80	1	3	0	81
数据2	100	2	5	1	121
数据3	80	3	3	0	102
…					
数据n	90	2	4	1	?

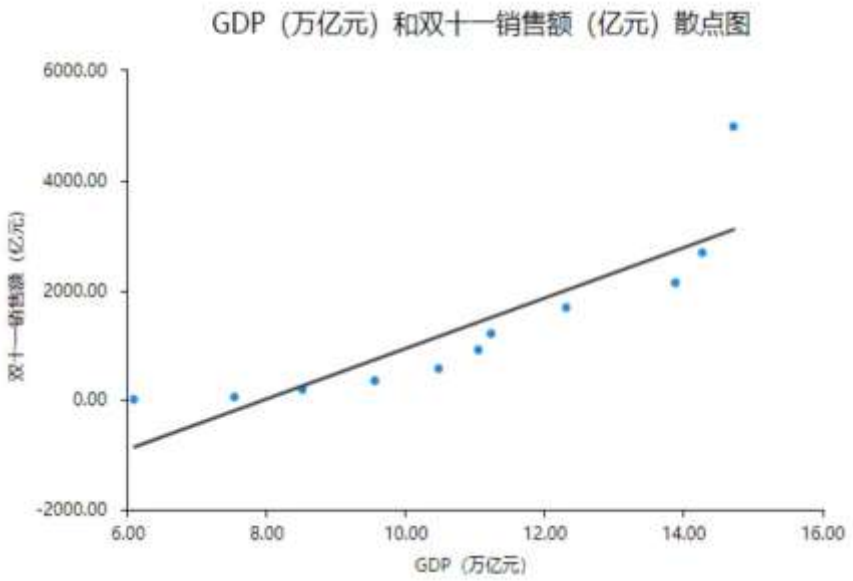
线性回归概念

- 应用常见场景是非常多

钢轨伸缩长度与温度

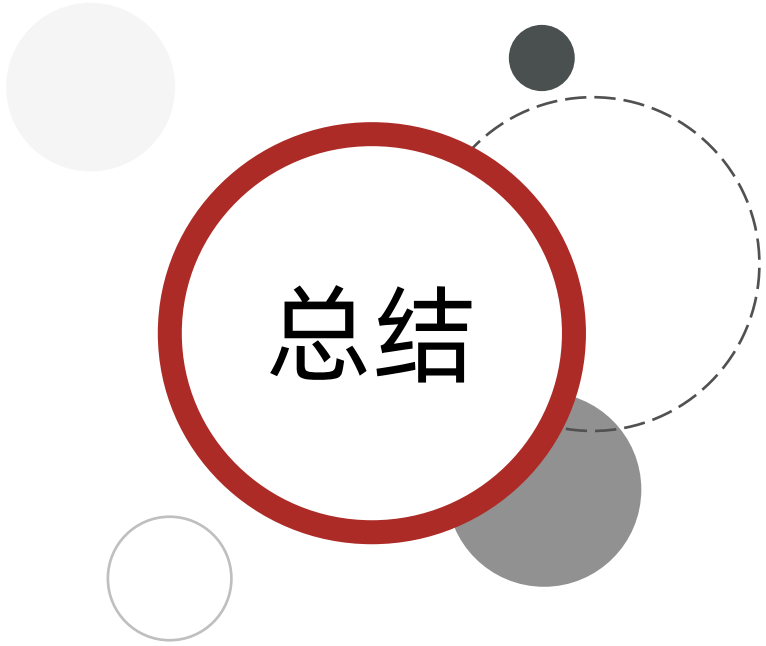


昆虫鸣叫次数与天气



国内GDP与双十一销售额

如何利用线性回归API来快速的解决实际问题呢？



1 线性回归概念

- 利用 回归方程(函数) 对 一个或多个自变量(特征值)和因变量(目标值)之间 关系进行建模的一种分析方式。
- 数学公式： $h_{(w)} = w_1x_1 + w_2x_2 + w_3x_3 + \ldots + b = w^T\mathrm{x} + b$

2 线性回归分类

- 一元线性回归、多元线性回归

目录

Contents

学习目标

Learning Objectives

1. 知道线性回归API的使用
2. 知道损失函数是什么

案例 一元线性回归案例：预测播仔身高

- 已知数据：

编号	身高
1	160
2	166
3	172
4	174
5	180
6	176

?

体重
56.3
60.6
65.1
68.5
75
?

- 需求：播仔身高是176，请预测体重？

对于这个回归案例如何利用API快速求解呢？

线性回归API介绍

1 导入 线性回归包 → 2 准备 数据 → 3 实例化 线性回归模型 → 4 训练 线性回归模型 → 5 模型 预测

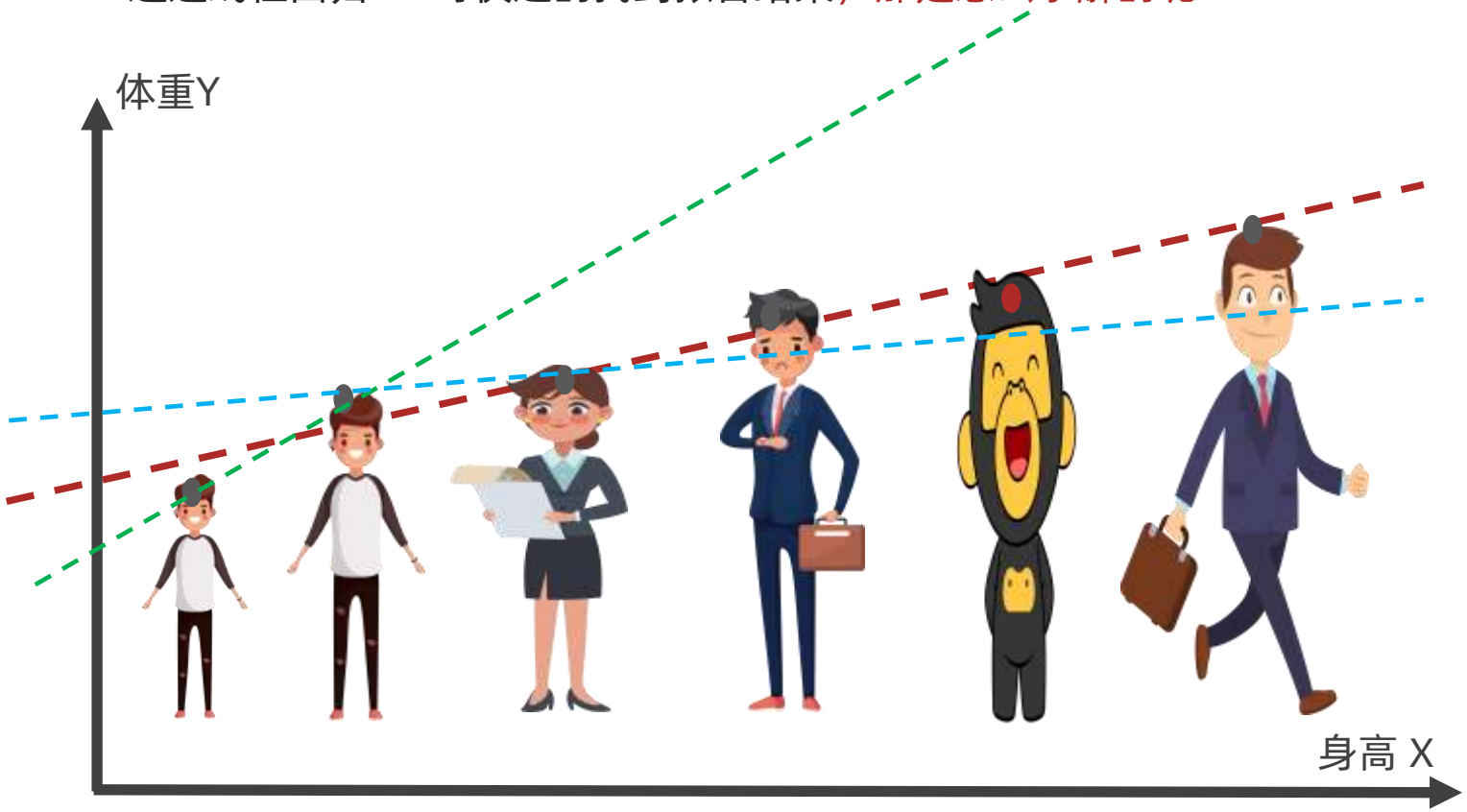
导包 from sklearn.linear_model import LinearRegression	X: 身高数据 x = [[160], [166], ...] Y:体重数据 y = [56.3, 60.6, ...]	*使用类* LinearRegression 实例化对象 estimator	estimator.fit(x, y) 从数据中获取规律 查看模型参数 斜率 coef_ 截距 intercept_	estimator.predict ([[176]])

线性回归API介绍

```
# 1 导入依赖包
from sklearn.linear_model import LinearRegression

def dm01_lr预测播仔身高():
    # 2 准备数据身高和体重
    x = [[160], [166],  [172], [174],[180]]
    y = [56.3, 60.6, 65.1, 68.5, 75]

    # 3 实例化 线性回归模型 estimator
    estimator = LinearRegression()

    # 4 训练 线性回归模型fit() h(w) = w1x1 + w2x2 + b
    estimator.fit(x, y)

    # 打印 线性回归模型参数 coef_ intercept_
    print('estimator.coef_-->', estimator.coef_)
    print('estimator.intercept_-->', estimator.intercept_)

    # 5 模型预测predit()
    myres = estimator.predict([[176]])
    print('myres-->', myres)
```

线性回归的求解方法

通过线性回归API可快速的找到拟合结果，那是怎么求解的呢？

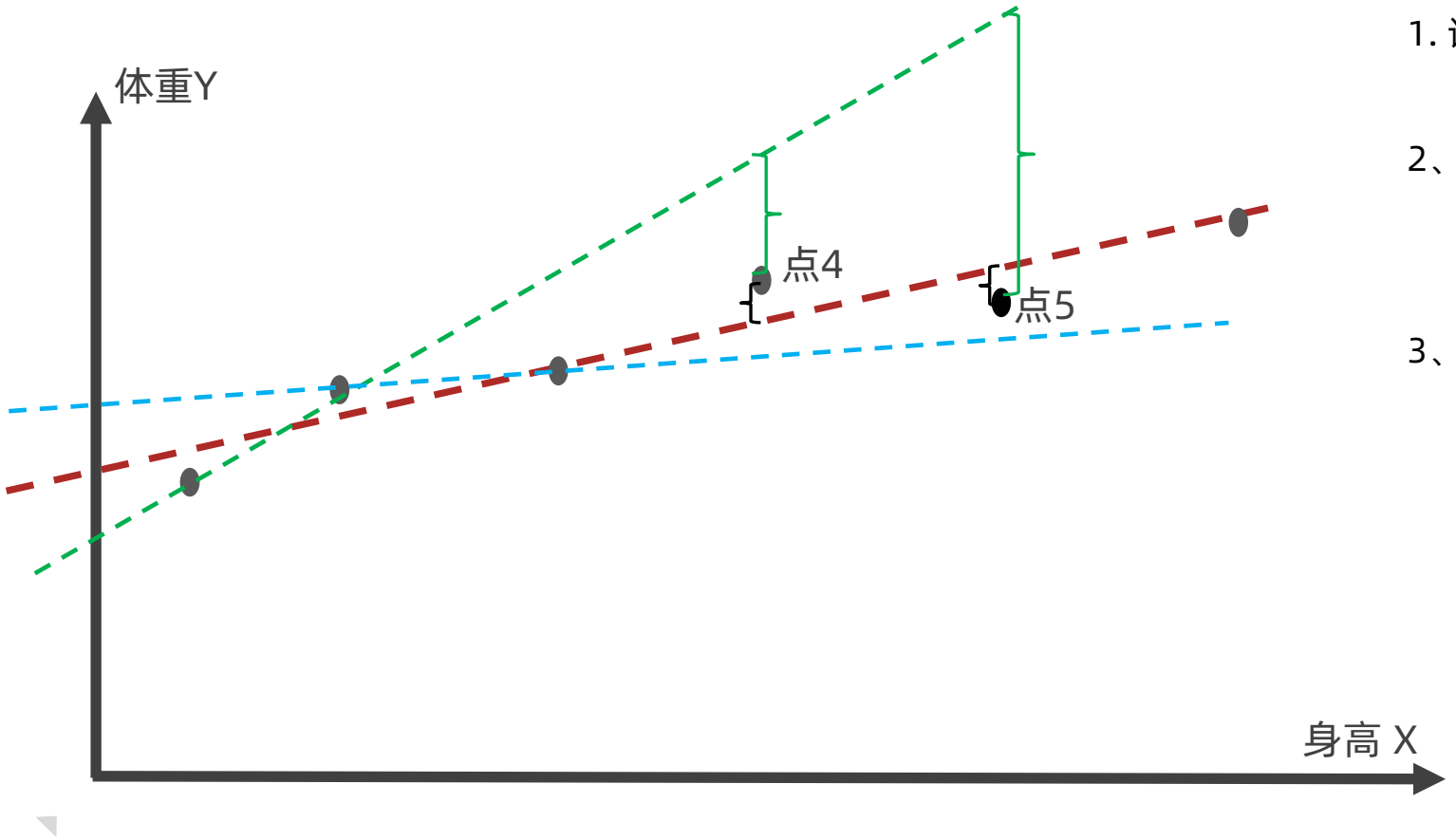


如何学习数据分布规律？

如何找到最优解？

损失函数

- 需要设置一个评判标准



1. 误差概念：用预测值y - 真实值y就是误差

2、损失函数：衡量每个样本预测值与真实值效果的函数，也叫代价函数、成本函数、目标函数

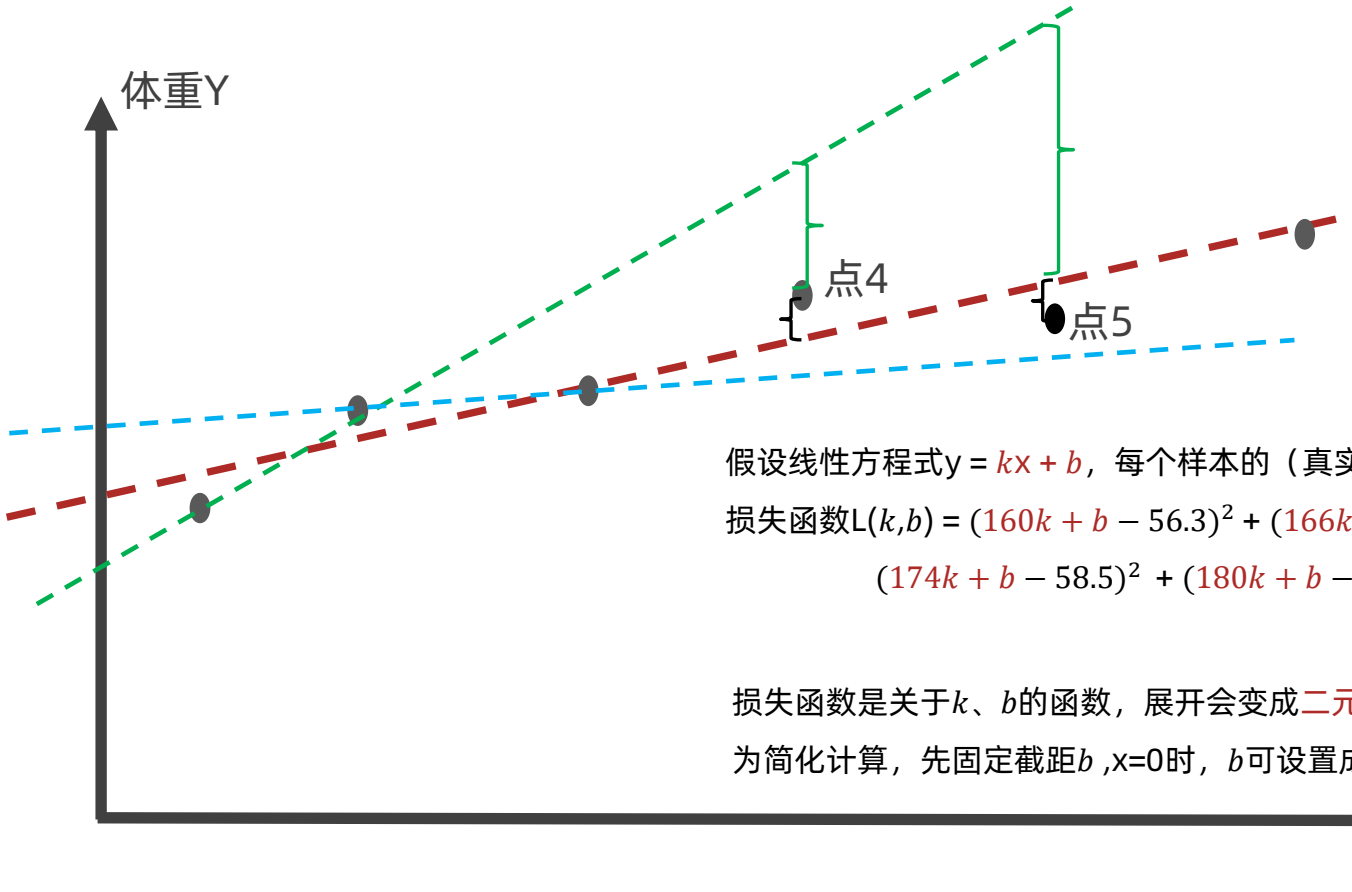
3、“红色直线能更好的拟合所有点”也就是误差最小；误差和最小

如何求损失函数误差最小呢？也就是损失函数的最优解？

损失函数

- 损失函数数学如何表达呢？又如何求损失函数的最小值呢？

编号	身高	体重
1	160	56.3
2	166	60.6
3	172	65.1
4	174	68.5
5	180	75
6	176	?



假设线性方程式y = $kx + b$，每个样本的（真实值-预测值）来形成损失函数

损失函数L(k,b) = $(160k + b - 56.3)^2$ + $(166k + b - 60.6)^2$ + $(172k + b - 65.1)^2$ + $(174k + b - 58.5)^2$ + $(180k + b - 56.3)^2$ 求此损失函数小值

损失函数是关于k、b的函数，展开会变成二元二次方程。

为简化计算，先固定截距b ,x=0时，b可设置成一个负值，b 固定成-100

损失函数L(k,b=-100) = $(160k + (-100) - 56.3)^2$ + $(166k + (-100) - 60.6)^2$ + $(172k + (-100) - 65.1)^2$ + $(174k + (-100) - 58.5)^2$ + $(180k + (-100) - 56.3)^2$

求损失函数最小值（求损失函数最优解）

- 当损失函数取最小值时，得到k就是最优解

编号	身高	体重
1	160	56.3
2	166	60.6
3	172	65.1
4	174	68.5
5	180	75
6	176	?

损失函数L(k, b=-100) = $(160k + (-100) - 56.3)^2 + (166k + (-100) - 60.6)^2 + (172k + (-100) - 65.1)^2 + (174k + (-100) - 58.5)^2 + (180k + (-100) - 56.3)^2$

$= (160k - 156.3)^2 + (166k - 160.6)^2 + (172k - 165.1)^2 + (174k - 158.5)^2 + (180k - 156.3)^2$

$= (160k)^2 - 2 * 160k * (156.3) + (156.3)^2 + (166k + 160.6)^2 + \ldots$

$= 145416\,k^2 - 281671.6\,k + 136496.32$

k = -(-281671.6) / (2*145416) = 0.9685

让损失函数值最小，就相当于直线拟合了所有的点！！

截距等于-100情况下，最优解，求播仔体重：

y = 0.9685x + (-100) = 0.9685*176 - 100 = 70.456

想求一条直线更好的拟合所有点 y = kx + b

==> 引入损失函数(衡量预测值和真实值效果) Loss(k, b)

==> 通过一个优化方法，求损失函数最小值，得到K最优解

损失函数的种类和数学表达

- 损失函数用来衡量真实值和预测值之间的差异，为优化参数指明了方向

编号	身高	体重
1	160	56.3
2	166	60.6
3	172	65.1
4	174	68.5
5	180	75
6	176	?

（1）均方误差 (Mean-Square Error, MSE)

损失函数J(w, b) = $\frac{1}{\mathrm{m}}\sum_{i=0}^{m}(h(x^{(i)}) - y^{(i)})^2$

（2）平均绝对误差 (Mean Absolute Error , MAE)

损失函数J(w, b) = $\frac{1}{\mathrm{m}}\sum_{0}^{\mathrm{m}}\left|h(x^{(i)}) - y^{(i)}\right|$

（3）均方根误差(Root Mean Square Error, RMSE)

损失函数J(w, b) = $\sqrt{\frac{1}{\mathrm{m}}\sum_{i=0}^{m}(h(x^{(i)}) - y^{(i)})^2}$

线性回归问题求解需要什么？

数据 → 线性回归模型 → 损失函数 → 优化方法

样本数据

特征值x和目标值y组成，
假设数据分布线性的

线性方程式
（假设函数）

1、样本x特征值有1个，
$Y = kx + b$ 线性关系
2 、样本特征值x有多个

损失函数种类

1、MSE
2、MAE
3、RMSE

让损失函数最小的方法

1、梯度下降法：利用梯度逐步逼近最优解
2、通过求方程法（求导、求偏导）

总结

1 线性回归API

- 实例化模型: estimator = LinearRegression()
- 模型训练:　estimator.fit(x, y)
- 模型预测:　estimator.predict([[4.0]])

2 损失函数概念

- 误差：预测值 - 真实值的差值
- 损失函数：衡量预测值和真实值效果的函数，也叫代价函数、目标函数

3 回归问题中的损失函数

- 最小二乘法：误差平方和
  - $\sum_{i=0}^{m}(h(x^{(i)}) - y^{(i)})^2$
- 均方误差 (Mean-Square Error, MSE)
  - $\frac{1}{m}\sum_{i=0}^{m}(h(x^{(i)}) - y^{(i)})^2$

均方根误差(Root Mean Square Error, RMSE)

损失函数J(w, b) $=\sqrt{\frac{1}{m}\sum_{i=0}^{m}(h(x^{(i)}) - y^{(i)})^2}$

- 平均绝对误差 (Mean Absolute Error , MAE)
  - $\frac{1}{m}\sum_{1}^{m}\left|h(x^{(i)}) - y^{(i)}\right|$

练习

- **有关损失函数下列说法正确的是？（多选）**

A）损失函数(Loss Function)又被称为代价函数(Cost Function)

B）损失函数可用来描述预测值的分布，看是否为均匀分布还是正态分布、或其他分布

C）损失函数可用来描述输出（预测值）和观测结果（真实值）效果，可衡量模型效果好坏

D）不同的任务比如分类、回归、聚类问题，一般会采用各自的损失函数

E）线性回归求解一般需要数据、假设函数、损失函数、损失函数优化方法等部分，相互配合共同完成

答案：ACDE

目录

Contents

学习目标

Learning Objectives

1. 知道机器学习中常见的数据表述
2. 知道什么是导数
3. 知道什么是偏导
4. 知道向量和矩阵的简单运算

基础数学 - 机器学习中常见的数据表述

- 为什么要学习标量、向量、矩阵、张量？

  宗旨：用到就学什么，不要盲目的展开、大篇幅学数学

- 标量scalar： 一个独立存在的数，只有大小没有方向

- 向量vector： 向量指一列顺序排列的元素。默认是列向量
  - 比如张三的数理化成绩信息：70,80,90 $\begin{pmatrix}70\\80\\90\end{pmatrix} \in R^3$
  - 向量有大小和方向
- 矩阵matrix： 二维数组
  - 比如张三、李四的数理化成绩信息：70,80,90； 75,85,95 $\begin{pmatrix}70,80,90\\75,85,95\end{pmatrix} \in R^{2*3}$
- 张量Tensor： 数组，张量是基于向量和矩阵的推广
  - 数学中的张量 $tensor \in R^{2*3*4}$：2个3*4矩阵 3个2*4 或4个2*3

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$A \epsilon R^{m \times n}$

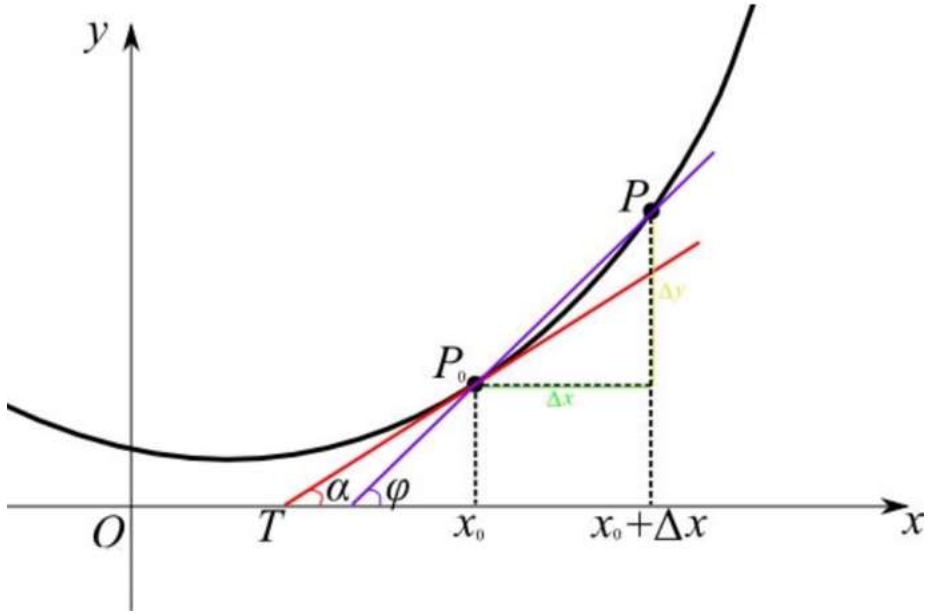
m代表多少行
n代表特征数

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ a_{31} & a_{32} & \cdots & a_{3n} \\ \cdots & \cdots & & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

导数 - 基本概念

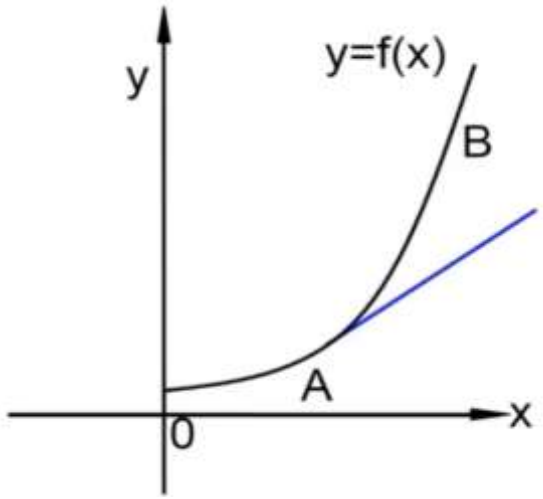
当函数 y=f (x) 的自变量 x 在一点 x_0 上产生一个增量Δx时，函数输出值的增量Δy与自变量增量Δx的比值在Δx趋于0时的极限A如果存在，A即为在 x_0 处的导数，记作f' (x_0) 。

$$f'(x_0)=\lim_{\Delta x\to 0}\frac{\Delta f}{\Delta x}=\lim_{\Delta x\to 0}\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x}$$



导数的几何意义

函数 y=f(x)在点x_0处的导数的几何意义，就是曲线y=f(x)在点$P(x_0, f(x_0))$处的切线的斜率，即曲线y=f(x)在点$P(x_0, f(x_0))$ 处的切线的斜率是 $f'(x_0)$.



常见函数的导数

公式	例子
$(C)' = 0$	$(5)' = 0 \quad (10)' = 0$
$(x^\alpha)' = \alpha x^{\alpha-1}$	$(x^3)' = 3x^2 \quad (x^5)' = 5x^4$
$(a^x)' = a^x \ln a$	$(2^x)' = 2^x \ln 2 \quad (7^x)' = 7^x \ln 7$
$(e^x)' = e^x$	$(e^x)' = e^x$
$(\log_a x)' = \frac{1}{x \ln a}$	$(\log_{10} x)' = \frac{1}{x \ln 10} \quad (\log_6 x)' = \frac{1}{x \ln 6}$
$(\ln x)' = \frac{1}{x}$	$(\ln x)' = \frac{1}{x}$
$(\sin x)' = \cos x$	$(\sin x)' = \cos x$
$(\cos x)' = -\sin x$	$(\cos x)' = -\sin x$

导数的四则运算

$[u(x)\pm v(x)]'=u'(x)\pm v'(x)$	$(e^x+4\ln x)'=(e^x)'+(4\ln x)'=e^x+\frac{4}{x}$
$[u(x)\cdot v(x)]'=u'(x)\cdot v(x)+u(x)\cdot v'(x)$	$(\sin x\cdot\ln x)'=(\sin x)'\cdot\ln x+\sin x\cdot(\ln x)'=\cos x\cdot\ln x+\sin x\cdot\frac{1}{x}$
$\left[\frac{u(x)}{v(x)}\right]'=\frac{u'(x)\cdot v(x)-u(x)\cdot v'(x)}{v^2(x)}$	$\left(\frac{e^x}{\cos x}\right)'=\frac{(e^x)'\cdot\cos x-e^x\cdot(\cos x)'}{\cos^2(x)}=\frac{e^x\cdot\cos x-e^x\cdot(-\sin x)}{\cos^2(x)}$
$\{g[h(x)]\}'=g'(h)\cdot h'(x)$	$(e^{2x})'=e^{2x}\cdot(2x)'=2e^{2x}$　　$(sin2x)'=cos2x\cdot(2x)'=2cos2x$

- 复合函数求导：g(h)是外函数 h(x)是内函数。先对外函数求导，再对内函数求导
- 举个例子:计算该函数$y=(x^2+2x)^2$ 的导函数

$$y'=2(x^2+2x)^{(2-1)}(x^2+2x)'=2(x^2+2x)(2x+2)=4(x^3+3x^2+2x)=4x^3+12x^2+8x$$

导数与运算规则

- 求导练习

1. $y = x^3 - 2x^2 + sinx$,求f`(x)
2. $(e^x + 4lnx)$`
3. $(sinx * lnx)$`
4. $(\frac{e^x}{cosx})$`
5. y=sin2x, 求$\frac{dy}{dx}$

导数与运算规则

1

$$\begin{aligned} y' &= (x^3 - 2x^2 + \sin x)' \\ &= (x^3)' - (2x^2)' + (\sin x)' \\ &= 3x^2 - 4x + \cos x \end{aligned}$$

2

$$(e^x + 4\ln x)' = (e^x)' + (4\ln x)' = e^x + \frac{4}{x}$$

3

$$(\sin x \cdot \ln x)' = (\sin x)' \cdot \ln x + \sin x \cdot (\ln x)' = \cos x \cdot \ln x + \sin x \cdot \frac{1}{x}$$

4

$$\left(\frac{e^x}{\cos x}\right)' = \frac{(e^x)' \cdot \cos x - e^x \cdot (\cos x)'}{\cos^2(x)} = \frac{e^x \cdot \cos x - e^x \cdot (-\sin x)}{\cos^2(x)}$$

5

$$(sin2x)' = cos2x \cdot (2x)' = 2cos2x$$

导数求极值

- 导数为0的位置是函数的极值点
- 求函数 y = x^2 - 4x + 5 的极小值

求导法：对x求导，令导数=0：　y' = 2x -4 = 0　x = 2。

所以 y 的极小值 = 1

偏导

- Z是关于x和y的函数记成z(x,y)，求解 $z = (x-2)^2 + (y-3)^2$ 的极小值

$\frac{\partial z}{\partial x} = ((x-2)^2 + (y-3)^2)' = ((x-2)^2)' = 2(x-2)(x-2)' = 2(x-2)*1 = 0$

x = 2时可以在x方向求导极小值

$\frac{\partial z}{\partial y} = ((x-2)^2 + (y-3)^2)' = ((y-3)^2)' = 2(y-3)(y-3)' = 2(y-3)*1 = 0$

y=3时可以在y方向求导极小值

多元函数的偏导(练习)

- 1 计算Z = x^2 + 2xy - 3y^2在x、y方向的偏导数

$$\frac{\partial z}{\partial x} = (x^2 + 2xy - 3y^2)' = 2x + 2y - 0 = 2x + 2y$$

$$\frac{\partial z}{\partial y} = (x^2 + 2xy - 3y^2)' = 0 + 2x - 6y = 2x - 6y$$

- 2 计算Z = x^y 在x、y方向的偏导数

$$\frac{\partial z}{\partial x} = (x^y)' = yx^{y-1}$$

$$\frac{\partial z}{\partial y} = (x^y)' = x^y \ln x$$

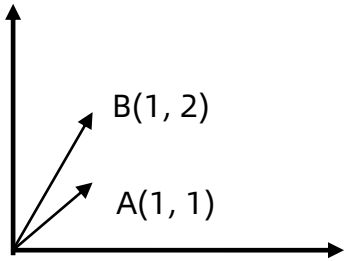
- 3 计算Z = e^{2x^2-3y} 在x、y方向的偏导数 注意：本题目是复合函数、多元函数求偏导

$$\frac{\partial z}{\partial x} = (e^{2x^2-3y})' = (e^{2x^2-3y})*(2x^2 - 3y)' = 4xe^{2x^2-3y}$$

$$\frac{\partial z}{\partial y} = (e^{2x^2-3y})' = (e^{2x^2-3y})*(2x^2 - 3y)' = -3e^{2x^2-3y}$$

向量和矩阵 - 向量运算

- 向量是有大小和方向
  - 几何意义上表示：向量(1,1), 向量（1,2）



- 向量基运算

$$\begin{pmatrix}1\\2\\3\end{pmatrix} + \begin{pmatrix}4\\5\\6\end{pmatrix} = \begin{pmatrix}5\\7\\9\end{pmatrix} \in R^3 \qquad \begin{pmatrix}1\\2\\3\end{pmatrix} - \begin{pmatrix}4\\5\\6\end{pmatrix} = \begin{pmatrix}-3\\-3\\-3\end{pmatrix} \in R^3 \qquad 3 * \begin{pmatrix}4\\5\\6\end{pmatrix} = \begin{pmatrix}12\\15\\18\end{pmatrix} \in R^3$$

- 向量矩阵转置 Transpose

$$X = \begin{pmatrix}1\\2\\3\end{pmatrix} \qquad x^T = (1,2,3) \qquad Y = \begin{bmatrix}11 & 12 & 13\\21 & 22 & 23\end{bmatrix} \qquad Y^T = \begin{bmatrix}11 & 21\\12 & 22\\13 & 23\end{bmatrix}$$

向量和矩阵 - 范数Norm

- 范数(norm)是数学中的一种基本概念，具有长度的意义
  - 1范数(L1范数)-向量中各个元素绝对值之和
  - 2范数(L2范数)-向量的模长，每个元素平方求和，再开平方根
  - p范数(Lp范数)-向量中每一个元素p幂求和，在开p次根
- L1范数

  $x^T = (1,\ 2,\ -3)$ $\|x\|_1 = |1| + |2| + |-3| = 6$

- L2范数

  $x^T = (1,\ 2,\ -3)$ $\|x\|_2 = \sqrt[2]{1^2 + 2^2 + (-3)^2} = \sqrt{12}$

  $x^T = (1,\ 2,\ -3)$ 注意：向量的转置@向量 $x^T x = 1^2 + 2^2 + (-3)^2 = 12$

  x为向量：$x^T x$ 与 $\|x\|_2^2$ 是一样的

- Lp范数

$$\|x\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{\frac{1}{p}}$$

向量和矩阵 - 矩阵 Matrix 1

- 矩阵是数学中的一种基本概念，表达m行n列的数据等
- 矩阵在机器学习中的表达

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \in R^{2*2}$$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \in R^{2*3}$$

一个矩阵m行n列：$A \in R^{m*n}$　　一个数据集$X \in R^{N*D}$ N多少行数据, D特征数

- 矩阵加法和减法

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 6 \\ 8 & 12 & 12 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} - \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- 矩阵乘法： 对应行列元素相乘，然后再加和再一起

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \in R^{2*3} \qquad B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 1 & 1 \end{bmatrix} \in R^{3*2} \qquad C = \begin{bmatrix} 1*1+2*3+3*1, & 1*2+2*4+3*1 \\ 4*1+5*3+6*1, & 4*2+5*4+6*1 \end{bmatrix} = \begin{bmatrix} 10, & 13 \\ 25, & 34 \end{bmatrix} \in R^{2*2}$$

$A \in R^{m*n}$, $B \in R^{n*d}$ ➔ $A@B=C \in R^{m*d}$

基础数学向量和矩阵 - 矩阵 Matrix 2

- 矩阵转置

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \in R^{2*3} \qquad A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \in R^{3*2}$$

- 矩阵@矩阵的转置

$A@A^T$ 是方阵， $A^T@A$ 是方阵

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \in R^{2*3}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} @ \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} = \begin{bmatrix} 1*1+2*2+3*3 & 1*4+2*5+3*6 \\ 4*1+5*2+6*3 & 4*4+5*5+6*6 \end{bmatrix} = \begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} @ \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1*1+4*4 & 1*2+4*5 & 1*3+4*6 \\ 2*1+5*4 & 2*2+5*5 & 2*3+5*6 \\ 3*1+6*4 & 3*2+6*5 & 3*3+6*6 \end{bmatrix} = \begin{bmatrix} 17 & 22 & 27 \\ 22 & 29 & 36 \\ 27 & 36 & 45 \end{bmatrix}$$

- 方阵：一种特殊的矩阵，其行数=列数

对称方阵：一种特殊的方阵，沿着主对角线，其元素对称 $a_{ij} = a_{ji}$

- 单位阵：一种特殊的方阵 符号E 或者 I

主对角线为1，其他为0

举个栗子：
三阶单位阵：

$$I_{3*3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

基础数学向量和矩阵 - 矩阵 Matrix 3

- 矩阵乘法的性质
  - 矩阵的乘法不满足交换律：A×B≠B×A
    - eg: A∈ R^{5*2} , B∈ R^{2*5}　A@B∈ R^{5*5} B@A=∈ R^{2*2}
    - 特殊条件下满足： AB = BA 的前提是A、B是同阶方阵　eg: $A_{2*2} \times B_{2*2} = B_{2*2} \times A_{2*2}$
  - 矩阵的乘法满足结合律。即：A×(B×C) = (A×B)×C
    - eg: A∈ R^{5*2} , B∈ R^{2*5}　C∈ R^{5*3} (A×B) ×C数据形状R^{5*3}
  - 矩阵与单位矩阵相乘等于矩阵本身
    - eg: A x I = A　I x A = A　I为单位矩阵
  - 矩阵的逆

若矩阵A = $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ ∈ R^{2*2}　　A x B = I单位矩阵 则B为A的逆矩阵，记为：A^{-1}

基础数学向量和矩阵 - 矩阵 Matrix 4

- 矩阵转置的性质
  - $(A^T)^T = A$
  - $(A+B)^T = A^T + B^T$
  - $(kA)^T = kA^T$ k为一个常数
  - $(AB)^T = B^T A^T$

比如：

$$\left(\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix} @ \begin{bmatrix}1 & 2\\ 3 & 4\\ 1 & 1\end{bmatrix}\right)^T = \begin{bmatrix}1 & 2\\ 3 & 4\\ 1 & 1\end{bmatrix}^T @ \begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}^T$$

总结

1 标量、向量、矩阵、张量概念

2 导数 Derivative

- 函数上某一个点求切线就是导数。瞬时速度变化率

3 导数的求解方法

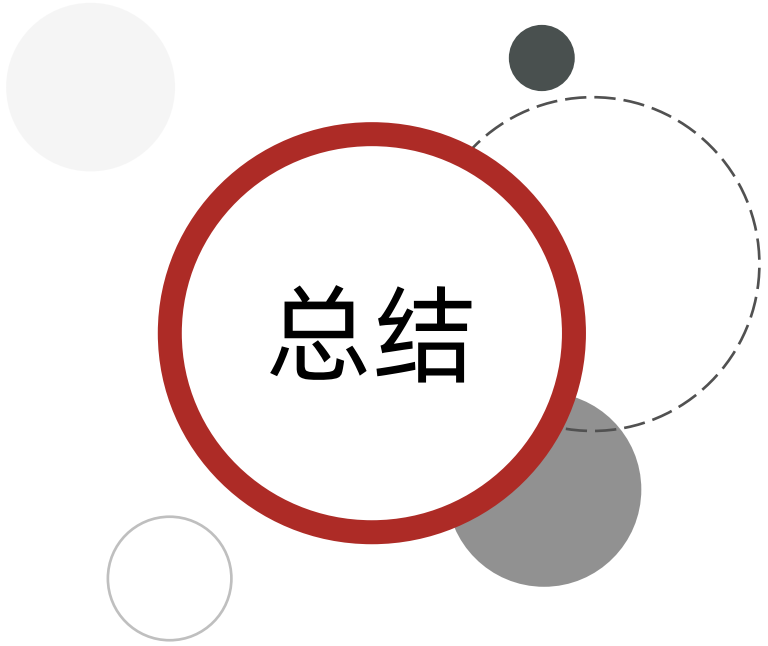
- 常数、指数函数、幂指数、正弦余弦都有自己的导函数，会查表应用
- 特别注意复合函数求导：先对外函数求导，在对内函数求导

4 利用导数求极值

- 导数为0的位置即为极值点

5 偏导数

- U是关于x、y、z的函数，记为u(x, y, z)，只在x分量上求导，则为求偏导。
- 各个分量上求偏导，会形成一个向量($\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z}$),这就组成了导数



1 向量的基本运算

- 1范数(L1范数)-向量中各个元素绝对值之和
- 2范数(L2范数)-向量的模长，每个元素平方求和，再开平方根
- x^T x 与 $\|x\|_2^2$是一样的
- p范数(Lp范数)-向量中每一个元素p幂求和，在开p次根

2 矩阵的运算

- 1 矩阵数学表示 X∈ R^{N*D}
- 2 矩阵加减法：形状相同对应元素加减
- 3 矩阵乘法：条件A∈ R^{m*n} , B∈ R^{n*d} ➔ A@B=C ∈ R^{m*d}
- 4 矩阵的乘法不满足交换律：A×B≠B×A； 矩阵的乘法满足结合律
- 5 矩阵与单位矩阵相乘等于矩阵本身
- 6 A@B = I单位矩阵 则B为A的逆矩阵，记为：A^{-1}
- 7 矩阵转置： $(A+B)^T = A^T + B^T$ ， $(AB)^T = B^T A^T$

练习

1. 已知函数f(x,y)，其中x，y均为变量，那么f(x,y)的偏导数df/dx是（）

A. f(x,y)的导数

B. f(x,y)的一阶偏导数

C. f(x,y)的二阶偏导数

D. f(x,y)的对x的偏导数

1 答案 D

2. 已知函数f(x)，令g(x)=f'(x)，则g'(x)=（）

A. f''(x)

B. f'(x)

C. f(x)

D. 不能确定

2 答案A

3. 已知函数f(x)，若在x=a时，函数取极小值，则（）

A. f'(a)=0

B. f'(a)>0

C. f'(a)<0

D. 不能确定

3 答案A

练习

例1. 设 $f(x,y)=x^2+y^4+y$，求 $\left.\frac{\partial f}{\partial x}\right|_{(0,0)}, \left.\frac{\partial f}{\partial y}\right|_{(0,0)}$.

解：先求偏导再代值：$\frac{\partial f}{\partial x}=2x, \frac{\partial f}{\partial y}=4y^3+1$，$\left.\frac{\partial f}{\partial x}\right|_{(0,0)}=0, \left.\frac{\partial f}{\partial y}\right|_{(0,0)}=1.$

注：此题也可按另外两种方法计算。

例2. 已知 $f(x,y)=\begin{cases}\frac{xy}{x^2+y^2}, (x,y)\neq(0,0)\\ 0, \quad (x,y)=(0,0)\end{cases}$，求 $\left.\frac{\partial f}{\partial x}\right|_{(0,0)}, \left.\frac{\partial f}{\partial y}\right|_{(0,0)}$.

解：由偏导数的定义得 $\left.\frac{\partial f}{\partial x}\right|_{(0,0)}=\lim\limits_{x\to 0}\frac{f(x,0)-f(0,0)}{x}=\lim\limits_{x\to 0}\frac{0}{x}=0$，同理 $\left.\frac{\partial f}{\partial y}\right|_{(0,0)}=0.$

也可按先代值再求导的方法计算：由 $f(x,0)=0$，所以 $\left.\frac{\partial f}{\partial x}\right|_{(0,0)}=0$，同理 $\left.\frac{\partial f}{\partial y}\right|_{(0,0)}=0.$

注：此题中函数是一个分段函数，不能像普通函数那样先求偏导再代值计算。

练习

1. 有关矩阵相乘说法正确的是( )

A. 两个矩阵能相乘的条件为：A∈ R^{m*n} , B∈ R^{d*m} ➔ A@B=C ∈ R^{n*d}

B. A∈ R^{4*5} A@A^T=C A^T@A =D, 则C和D都是方阵

C. A∈ R^{4*5} A@A^T=C A^T@A =D, 则C和D都是方阵，而且C=D

D. A∈ R^{4*5} 则A的逆矩阵一定存在

答案解析：1 A矩阵条件错应该中间一致。B对 C错 D错 有些矩阵不一定有逆 答案：B

2. 已知 矩阵A = $\begin{bmatrix}1 & 2 & -1\\ 4 & 5 & -1\end{bmatrix} \in R^{2*3}$ ， B = $\begin{bmatrix}1 & -1\\ 1 & -1\\ 1 & -1\end{bmatrix} \in R^{3*2}$

请计算 A@B = ()

答案解析2 $\begin{bmatrix}2 & -2\\ 8 & -8\end{bmatrix}$

3. 已知 矩阵A = $\begin{bmatrix}1 & -1\\ 2 & -1\end{bmatrix} \in R^{2*2}$ ，请计算A的逆？

答案解析3 $\begin{bmatrix}1 & -1\\ 2 & -1\end{bmatrix}$ @ $\begin{bmatrix}a & b\\ c & d\end{bmatrix}$ = $\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$

最终求： $\begin{bmatrix}a & b\\ c & d\end{bmatrix}$ = $\begin{bmatrix}-1 & 1\\ 2 & 1\end{bmatrix}$

目录

Contents

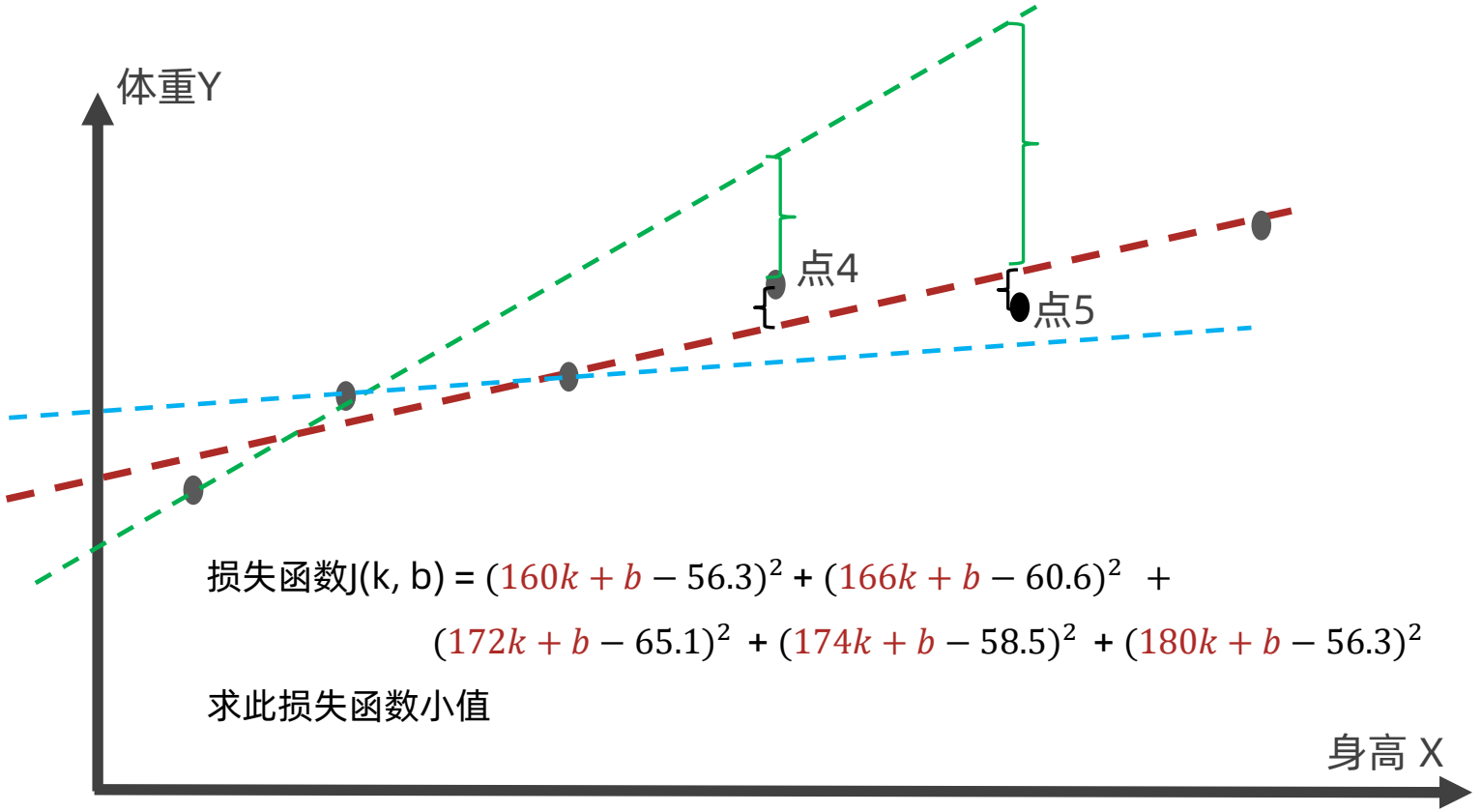
学习目标

Learning Objectives

1. 知道正规方程法求解过程

一元线性回归 - 解析解

- 损失函数用来衡量真实值和预测值之间的差异，为优化参数指明了方向



编号	身高	体重
1	160	56.3
2	166	60.6
3	172	65.1
4	174	68.5
5	180	75
6	176	?

损失函数J(k, b) = $(160k + b - 56.3)^2$ + $(166k + b - 60.6)^2$ + $(172k + b - 65.1)^2$ + $(174k + b - 58.5)^2$ + $(180k + b - 56.3)^2$

求此损失函数小值

求损失函数最小值：损失函数J(k, b) = $\sum_{i=0}^{m}(h(x^{(i)}) - y^{(i)})^2 = \sum_{i=0}^{m}(kx^{(i)} + b - y^{(i)})^2$ 其中i代表第几个样本

一元线性回归 - 解析解

一元线性回归损失函数J(k, b) = $\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2 = \sum_{i=1}^{m}(kx^{(i)} + b - y^{(i)})^2$的极小值

损失函数是关于k、b的函数，对k、b分别求偏导设置成0，得到2个方程

$\frac{\partial J(k,b)}{\partial k} = \sum_{i=1}^{m} 2(kx^{(i)} + b - y^{(i)})^{(2-1)} (kx^{(i)} + b - y^{(i)})' = \sum_{i=1}^{m}(2kx^{(i)^2} + 2bx^{(i)} - 2x^{(i)}y^{(i)}) = 0$ ------1式

$\frac{\partial J(a,b)}{\partial b} = \sum_{i=1}^{m} 2(kx^{(i)} + b - y^{(i)})^{(2-1)} (kx^{(i)} + b - y^{(i)})' = \sum_{i=1}^{m}(2kx^{(i)} + 2b - 2y^{(i)}) = 0$ ------2式

对1式、2式化简，$y^{(i)}$代表第i个样本的预测值

$k\sum_{i=1}^{m} x^{(i)^2} + b\sum_{i=1}^{m} x^{(i)} - \sum_{i=1}^{m} x^{(i)}y^{(i)} = 0$ ------3式

$k\sum_{i=1}^{m} x^{(i)} + bm - \sum_{i=1}^{m} y^{(i)}) = 0$ ------4式

对数据带入3式、4式求解k、b

$k*(160^2+166^2+172^2+174^2+180^2) + b*(160+166+172+174+180) - (160*56.3+166*60.6+172*65.1+174*68.5+180*75) = 0$ ------5式

$k*(160+166+172+174+180) + b*5 - (56.3+60.6+65.1+68.5+75) = 0$ ------6式

145416*k + 852*b - 55683.8 = 0

852*k + 5*b - 325.5 = 0 请求解k、b的值

根据k、b进行预测：
k = 0.0397 b = 60.7615
y = 0.0397 * x + 60.7615

编号	身高	体重
1	160	56.3
2	166	60.6
3	172	65.1
4	174	68.5
5	180	75
6	176	?

y = 0.0397 * 176 + 60.7615 = 67

多元线性回归 - 正规方程法

多元线性回归方程的已知：

多元线性回归方程式： $y = w_1x_1 + w_2x_2 + w_3x_3 + \ldots + b = w^T x + b$

有数据集D = $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, $x_i \in R^d$, $y_i \in R$, , b ∈ R

其中模型权重w是一个向量$w = \{w_1, w_2, w_3, \ldots. w_d\}$ ；其中代表特征数

多元线性回归方程的损失函数：

第1个样本的预测值： $\widehat{y_1} = w_1x_{11} + w_2x_{12} + w_3x_{13} + \ldots + w_dx_{1d} + b$

第1个样本的损失

$$\varepsilon_1^2 = (\widehat{y_1} - y_1)^2 = (w_1x_{11} + w_2x_{12} + w_3x_{13} + \ldots + w_dx_{1d} + b - y_1)^2 = \left(\left(\sum_{j=1}^{d} w_jx_{1j} + b\right) - y_1\right)^2$$

n个样本样本损失最小: 相当于把第1个样本损失 + 第2个样本损失 + ... 第n个样本的损失

$$\text{Loss(W)} = \sum_{i=1}^{n}(\widehat{y_i} - y_i)^2 = \sum_{i=1}^{n}\sum_{j=1}^{d}(w_jx_{ij} + b - y_i)^2$$

只要让多元线性回归损失函数取最小值，此时的权重w(w就是一个向量)就是最优解！

求最优解的方法：

1 解矩阵方程（也就是正规方程）

2 通过梯度下降的方法求解

多元线性回归 -正规方程法

- 损失函数普通方式转成矩阵方式（正规方程）：

$$J(w) = (h(x_1) - y_1)^2 + (h(x_2) - y_2)^2 + \cdots + (h(x_m) - y_m)^2$$
$$= \sum_{i=1}^{m} (h(x_i) - y_i)^2 = \|Xw - y\|_2^2$$

- 损失函数最小值：

$$2(Xw - y) * X = 0 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (1)$$
$$2(Xw - y) * (XX^T) = 0X^T \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2)$$
$$2(Xw - y) * (XX^T)(XX^T)^{-1} = 0X^T(XX^T)^{-1} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (3)$$
$$2(Xw - y) = 0 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (4)$$
$$Xw = y \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (5)$$
$$X^T Xw = X^T y \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (6)$$
$$(X^T X)^{-1}(X^T X) * w = (X^T X)^{-1} * X^T y \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (7)$$
$$w = (X^T X)^{-1} * X^T y \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (8)$$

多元线性回归 -正规方程法

- 正规方程w公式的解释说明：

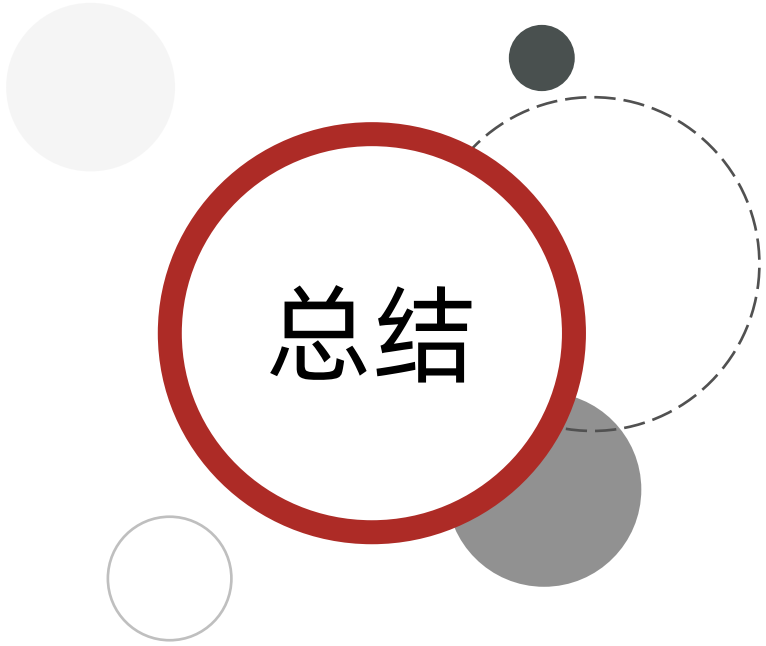
x_0	Size (feet²) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

X(0)	X(1)	X(2)	X(3)	X(4)	y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}$$

正规方程的w：

$$w = (X^T X)^{-1} X^T y = \left\{ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2104 & 1416 & 1534 & 852 \\ 5 & 3 & 3 & 2 \\ 1 & 2 & 2 & 1 \\ 45 & 40 & 30 & 36 \end{bmatrix} \times \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \right\}^{-1} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2104 & 1416 & 1534 & 852 \\ 5 & 3 & 3 & 2 \\ 1 & 2 & 2 & 1 \\ 45 & 40 & 30 & 36 \end{bmatrix} \times \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$



1 多元线性回归的正规方程

- 线性回归最小而成损失函数

  J(w) = $\|Xw - y\|_2^2$ 取值最小

  其解为：$w = (X^T X)^{-1} X^T y$

目录

Contents

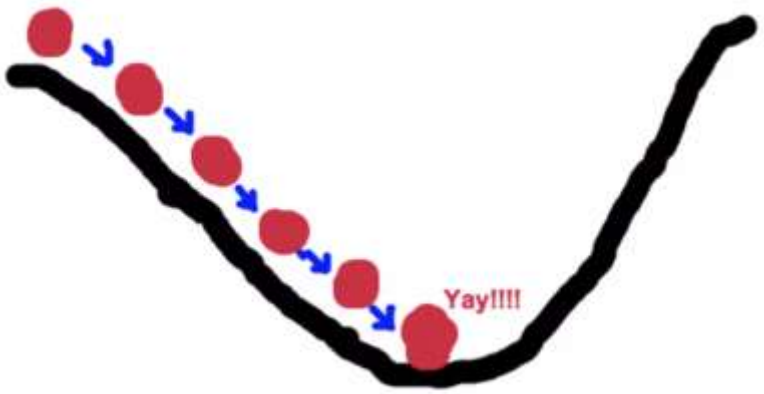
学习目标

Learning Objectives

1. 掌握梯度下降算法的思想
2. 知道梯度下降算法的分类
3. 知道正规方程和梯度下降算法的特点

梯度下降法 - 梯度下降和梯度

- 什么是梯度下降法
  - 顾名思义：沿着梯度下降的方向求解极小值
  - 举个例子：坡度最陡下山法

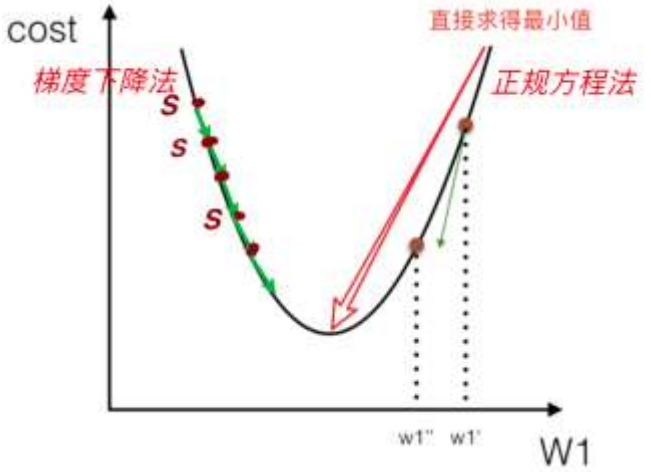


梯度下降过程就和下山场景类似

可微分的损失函数，代表着一座山

寻找的函数的最小值，也就是山底

- 输入：初始化位置S；每步距离为a 。输出：从位置S到达山底
- 步骤1：令初始化位置为山的任意位置S
- 步骤2：在当前位置环顾四周，如果四周都比S高返回S；否则执行步骤3
- 步骤3：在当前位置环顾四周，寻找坡度最陡的方向，令其为x方向
- 步骤4：沿着x方向往下走，长度为a，到达新的位置S'
- 步骤5：在S'位置环顾四周，如果四周都比S'高，则返回S'。否则转到步骤3



梯度下降法 - 梯度下降和梯度

- 什么是梯度 gradient grad
  - 单变量函数中，梯度就是某一点切线斜率（某一点的导数）；梯度方向为函数增长最快的方向
  - 多变量函数中，梯度就是某一个点的偏导数；有方向：偏导数分量的向量方向

- 梯度下降公式
  - 循环迭代求当前点的梯度，更新当前的权重参数

$$\theta_{i+1} = \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta)$$

    - α: 学习率(步长) 不能太大, 也不能太小. 机器学习中：0.001 ~ 0.01
    - 梯度是上升最快的方向, 我们需要是下降最快的方向, 所以需要加负号

梯度下降法 - 梯度下降和梯度

- 单变量梯度下降 - 举个栗子

函数：$J(\theta) = \theta^2$, 求当 θ 为何值时，J(θ) 值最小　J(θ) 函数关于 θ 的导数为: 2θ

初始化：起点为：1，学习率：α = 0.4

我们开始进行梯度下降的迭代计算过程:

第一步：θ = 1

第二步：θ = θ - α * (2θ) = 1 - 0.4 * (2*1) = 0.2

第三步：θ = θ - α * (2θ) = 0.2 - 0.4 * (2*0.2) = 0.04

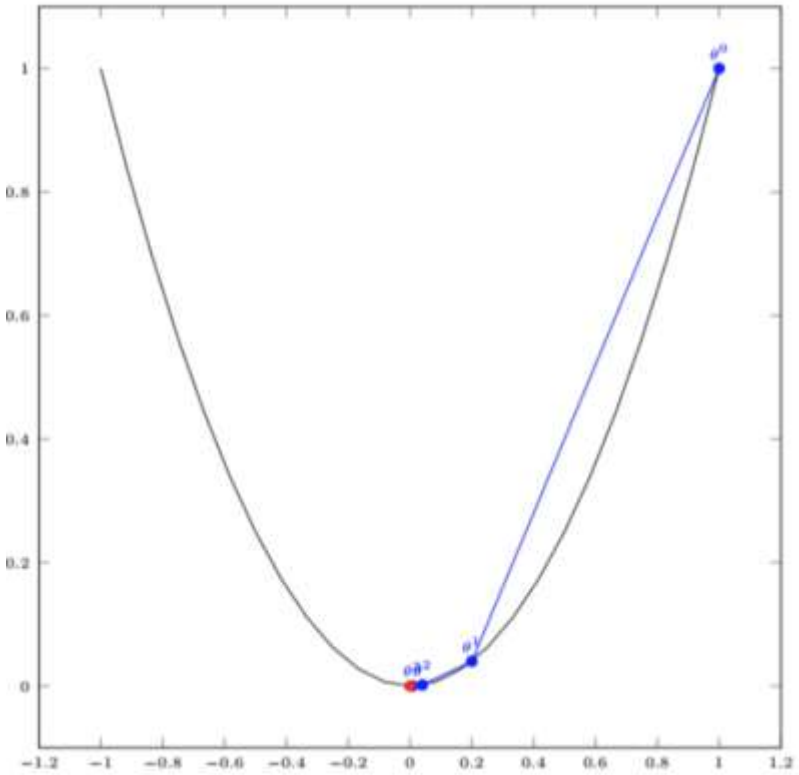
第四步：θ = θ - α * (2θ) = 0.04 - 0.4 * (2*0.04) = 0.008

第五步：θ = θ - α * (2θ) = 0.008 - 0.4 * (2*0.008) = 0.0016

....

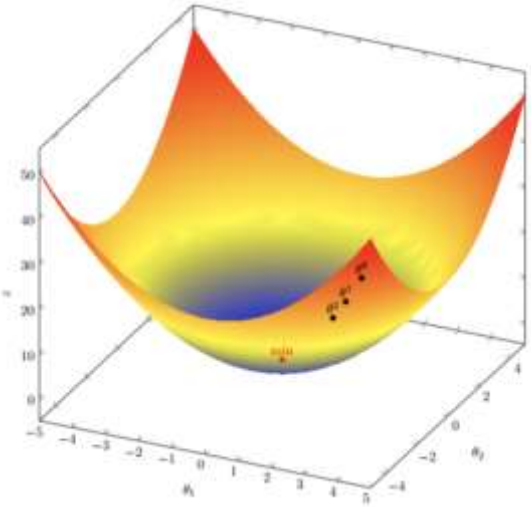
第N步：θ 已经极其接近最优值 0，J(θ) 也接近最小值。

小结：经过四次的运算，即走了四步，基本抵达了函数的最低点



梯度下降法 - 梯度下降和梯度

- 多变量梯度下降 - 举个栗子



函数：$J(\theta) = \theta_1^2 + \theta_2^2$ ，求 θ_1、 θ_2 为何值时，J(θ) 的值最小

J(θ) 函数关于 θ_1的导数为: $2\theta_1$ ，J(θ) 函数关于 θ2 的导数为: $2\theta_2$

则 J(θ) 的梯度为：$(2\theta_1, 2\theta_2)$

初始化： 起点为: (1, 3) 学习率为:α = 0.1

最小值是 (0，0)点，下面使用梯度下降法一步步的计算

我们开始进行梯度下降的迭代计算过程:

第一步：$(\theta_1, \theta_2) = (\theta_1, \theta_2) - \alpha * (2\theta_1, 2\theta_2) = (\theta_1 - \alpha*2\theta_1, \theta_2 - \alpha*2\theta_2) = (1-0.1*2, 3-0.1*6)=(0.8, 2.4)$

第二步：$(\theta_1, \theta_2) = (\theta_1, \theta_2) - \alpha * (2\theta_1, 2\theta_2) = (\theta_1 - \alpha*2\theta_1, \theta_2 - \alpha*2\theta_2) = (0.8-0.1*1.6, 2.4-0.1*4.8)=(0.64, 1.92)$

...

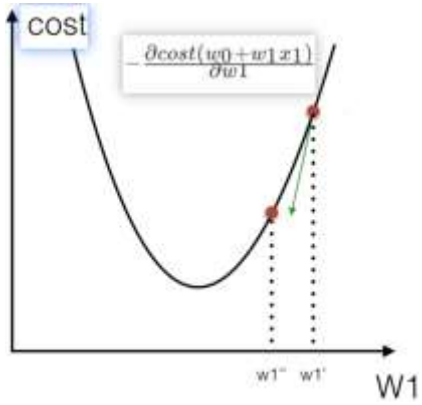
第N步： θ_1 、 θ_2 已经极其接近最优值，J(θ) 也接近最小值。

梯度下降法 - 梯度下降和梯度

- 梯度下降优化过程

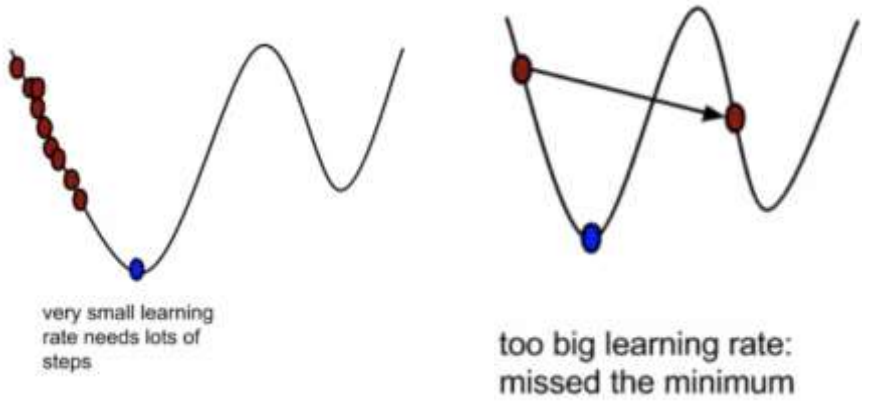
1. 给定初始位置、步长（学习率）
2. 计算该点当前的梯度的负方向
3. 向该负方向移动步长
4. 重复 2-3 步 直至收敛
    - 两次差距小于指定的阈值
    - 达到指定的迭代次数

- 梯度下降公式中，为什么梯度要乘以一个负号
    - 梯度的方向实际就是函数在此点上升最快的方向！
    - 需要朝着下降最快的方向走，负梯度方向， 所以加上负号



- 有关学习率步长(Learning rate)

1. 步长决定了在梯度下降迭代的过程中，每一步沿梯度负方向前进的长度
2. 学习率太小，下降的速度会慢
3. 学习率太大：容易造成错过最低点、产生下降过程中的震荡、甚至梯度爆炸



梯度下降法 -案例银行信贷

- 案例：梯度下降法计算每个样本产生的梯度

银行信贷只考虑每月薪资（元）、存款余额（元）、房产面积（平方米）

已知：8位贷款人的数据 。

目标：计算数据样本产生的梯度，求解线性回归模型

贷款编号	姓名	每月工资	存款余额	房产面积	授信额度(元)
1	张一	6000	12000	55	30000
2	张二	8000	10000	65	45300
3	张三	7500	16000	60	46000
4	赵六	10000	15000	75	55500
5	钱七	9000	21000	70	58000
6	孙八	12000	19000	85	75400
7	周九	11000	26000	90	81000
8	吴十	13000	32000	80	76800

已知：数据，假设函数，损失函数

$h_{(\boldsymbol{\theta})} = \boldsymbol{\theta}_1 x_1 + \boldsymbol{\theta}_2 x_2 + \boldsymbol{\theta}_3 x_3 + \ldots + b = \boldsymbol{\theta}_0 x_0 + \boldsymbol{\theta}_1 x_1 + \boldsymbol{\theta}_2 x_2 + \boldsymbol{\theta}_3 x_3 \text{ -}\ldots$　　b置换成$\theta_0 x_0$, x_0=1

$J_{(\boldsymbol{\theta})} = \frac{1}{2m}\sum_{i=1}^{m}(h_{\boldsymbol{\theta}}(x^{(i)}) - y^{(i)})^2$　　m条样本，$h_{\boldsymbol{\theta}}(x^{(i)})$第i个样本的预测值，$y^{(i)}$第i个样本真实值

目标：计算8个样本的产生的平均梯度，带入梯度下降公式更新权重 $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \boldsymbol{\alpha}\frac{\partial}{\partial \boldsymbol{\theta}_i} J(\boldsymbol{\theta})$

梯度下降法-案例银行信贷

- 梯度下降法的向量/标量表示

θ是一个向量 $(\theta_0, \theta_1, \theta_2, \dots \theta_d)^T$，损失函数对向量θ求导，如下：

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial \frac{1}{2m}\sum_{i=1}^{m}(h_{\boldsymbol{\theta}}(x^{(i)})-y^{(i)})^2}{\partial \boldsymbol{\theta}} = \frac{1}{m}\sum_{i=1}^{m}(h_{\boldsymbol{\theta}}(x^{(i)})-y^{(i)})^{(2-1)}\,(h_{\boldsymbol{\theta}}(x^{(i)})-y^{(i)})'_{\boldsymbol{\theta}} = \frac{1}{m}\sum_{i=1}^{m}(h_{\boldsymbol{\theta}}(x^{(i)})-y^{(i)})\,\frac{\partial(\boldsymbol{\theta}_0 x_0 + \boldsymbol{\theta}_1 x_1 + \boldsymbol{\theta}_2 x_2 + \boldsymbol{\theta}_3 x_3 - \dots)}{\partial(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots \boldsymbol{\theta}_d)} = \frac{1}{m}\sum_{i=1}^{m}(h_{\boldsymbol{\theta}}(x^{(i)})-y^{(i)})\,x^i$$

线性回归梯度下降法的梯度向量表示：$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{m}\sum_{i=1}^{m}(h_{\boldsymbol{\theta}}(x^{(i)})-y^{(i)})\,x^i$ # x^i为第i个样本的向量表示 $(x_{i0}, x_{i1}, x_{i2}, \dots x_{id})$

梯度下降法的梯度向量表示改写成标量表示：

$\frac{\partial J(\theta_0, \theta_1, \theta_2, \dots \theta_d)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_{\boldsymbol{\theta}}(x^{(i)})-y^{(i)})\,x_j^{(i)}$ # $x_j^{(i)}$第i个样本的第j个特征值，可看成x_{ij} 就是第i行第j列的数据值

有了每个分量的梯度，带入标量梯度下降公式计算

$\theta_j := \theta_j - \boldsymbol{\alpha}\frac{1}{m}\sum_{i=1}^{m}((h_{\boldsymbol{\theta}}(x^{(i)})-y^{(i)})\,x_j^{(i)})$ # 8个样本分别在3个特征值分量上产生偏导数，计算每个分量的平均偏导数

梯度下降法 - 案例银行信贷

贷款编号	姓名	每月工资	存款余额	房产面积	授信额度(元)
1	张一	6000	12000	55	30000
2	张二	8000	10000	65	45300
3	张三	7500	16000	60	46000
4	赵六	10000	15000	75	55500
5	钱七	9000	21000	70	58000
6	孙八	12000	19000	85	75400
7	周九	11000	26000	90	81000
8	吴十	13000	32000	80	76800

根据标量梯度下降来计算 $\theta_j := \theta_j - \boldsymbol{\alpha} \frac{1}{m} \sum_{i=1}^{m} \left(\left(h_{\boldsymbol{\theta}}(x^{(i)}) - y^{(i)} \right) x_j^{(i)} \right)$

线性回归式为： $h_{(\boldsymbol{\theta})} = \boldsymbol{\theta}_0 + \boldsymbol{\theta}_1 x_1 + \boldsymbol{\theta}_2 x_2 + \boldsymbol{\theta}_3 x_3 = \boldsymbol{\theta}_0 x_0 + \boldsymbol{\theta}_1 x_1 + \boldsymbol{\theta}_2 x_2 + \boldsymbol{\theta}_3 x_3$

假设 $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 = \boldsymbol{\theta}_3 = 1$，学习率 $\boldsymbol{\alpha} = 0.001$ $x_0=1$ 。计算如下：

8个样本会在各个分量上产生8个梯度。先在偏置分量上计算平均梯度

张一 样本：在偏置分量上产生的梯度 $\left(\left(h_{\boldsymbol{\theta}}(x^{(1)}) - y^{(1)} \right) x_0^{(1)} \right)$ = (1 + 1*6000 + 1*12000 + 1*55 - 30000) *1 = -11944

第2个样本，在偏置分量上产生的梯度 $\left(\left(h_{\boldsymbol{\theta}}(x^{(2)}) - y^{(2)} \right) x_0^{(2)} \right)$ = (1 + 1*8000 + 1*10000 + 1*65 - 45300) *1 = -27234

第3个样本，在偏置分量上产生的梯度 $\left(\left(h_{\boldsymbol{\theta}}(x^{(3)}) - y^{(3)} \right) x_0^{(3)} \right)$ = (1 + 1*7500 + 1*16000 + 1*60 - 46000) *1 = -22439

第4个样本，在偏置分量上产生的梯度 $\left(\left(h_{\boldsymbol{\theta}}(x^{(4)}) - y^{(4)} \right) x_0^{(4)} \right)$ = -30424，第5/6/7/8计算数据： -27929、-44314、-43909、-31719

$\boldsymbol{\theta}_0$的梯度为： $\nabla \boldsymbol{\theta}_0 = \frac{1}{m} \sum_{i=1}^{m} \left(\left(h_{\boldsymbol{\theta}}(x^{(i)}) - y^{(i)} \right) x_0^{(i)} \right)$ = -29989

再更新$\boldsymbol{\theta}_0$的梯度为 $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_0 - \boldsymbol{\alpha} \nabla \boldsymbol{\theta}_0$ = 1 - （0.001*(-29989)) = 30.989

梯度下降法 - 案例银行信贷

- 案例：梯度下降法计算每个样本产生的梯度

计算在每月工资上产生的梯度，θ_1的梯度为：$\nabla\theta_1 = \frac{1}{m}\sum_{i=1}^{m}((h_\theta(x^{(i)}) - y^{(i)})\,x_1^{(i)})$ = -305067937.5

计算在存款余额上产生的梯度，θ_2的梯度为：$\nabla\theta_2 = \frac{1}{m}\sum_{i=1}^{m}((h_\theta(x^{(i)}) - y^{(i)})\,x_2^{(i)})$ = -602021125

计算在房产面积上产生的梯度，θ_3的梯度为：$\nabla\theta_3 = \frac{1}{m}\sum_{i=1}^{m}((h_\theta(x^{(i)}) - y^{(i)})\,x_3^{(i)})$ = -2283290

再利用梯度下降公式更新θ_1、θ_2、θ_3

$\theta_1 = \theta_1 - \alpha\,\nabla\theta_1 = 1 - (0.001 * -305067937.5) = 305768.94$

$\theta_2 = \theta_2 - \alpha\,\nabla\theta_2 = 1 - (0.001 * -602021125) = 602022.125$

$\theta_3 = \theta_3 - \alpha\,\nabla\theta_3 = 1 - (0.001 * -2283290) = 2284.29$

经过第1轮迭代：初始化值θ[1, 1, 1, 1] 迭代成：

θ = [30.989, 305768.94, 602022.125, 2284.29]

后续多次迭代，可计算出最终的向量θ！！！

贷款编号	姓名	每月工资	存款余额	房产面积	授信额度(元)
1	张一	6000	12000	55	30000
2	张二	8000	10000	65	45300
3	张三	7500	16000	60	46000
4	赵六	10000	15000	75	55500
5	钱七	9000	21000	70	58000
6	孙八	12000	19000	85	75400
7	周九	11000	26000	90	81000
8	吴十	13000	32000	80	76800

梯度下降法分类

- 全梯度下降算法 FGD

  每次迭代时，使用全部样本的梯度值

  $$\theta_{i+1} = \theta_i - \alpha \sum_{j=0}^{m} (h_\theta(x_0^{(j)}, x_1^{(j)}, \cdots, x_n^{(j)}) - y_j) x_i^{(j)}$$

  有m个样本，求梯度时用了所有m个样本

- 随机梯度下降算法 SGD

  每次迭代时，随机选择并使用一个样本梯度值

  $$\theta_{i+1} = \theta_i - \alpha (h_\theta(x_0^{(j)}, x_1^{(j)}, \cdots, x_n^{(j)}) - y_j) x_i^{(j)}$$

  ```
  from sklearn.linear_model import SGDRegressor
  ```

- 小批量梯度下降算法 mini-batch

  每次迭代时，随机选择并使用小批量的样本梯度值

  从m个样本中，选择x个样本进行迭代(1<x<m)，

  $$\theta_{i+1} = \theta_i - \alpha \sum_{j=t}^{t+x-1} (h_\theta(x_0^{(j)}, x_1^{(j)}, \cdots, x_n^{(j)}) - y_j) x_i^{(j)}$$

  若batch_size=1，则变成了SGD；若batch_size=n，则变成了FGD

- 随机平均梯度下降算法 SAG

  每次迭代时，随机选择一个样本的梯度值和以往样本的梯度值的均值

  $$\theta_{i+1} = \theta_i - \frac{\alpha}{n} \sum_{j=1}^{n} (h_\theta(x_0^{(j)}, x_1^{(j)}, \ldots x_n^{(j)}) - y_j) x_i^{(j)}$$

1. 随机选择一个样本，假设选择 D 样本，计算其梯度值并存储到列表：[D]，然后使用列表中的梯度值均值，更新模型参数。
2. 随机再选择一个样本，假设选择 G 样本，计算其梯度值并存储到列表：[D, G]，然后使用列表中的梯度值均值，更新模型参数。
3. 随机再选择一个样本，假设又选择了 D 样本，重新计算该样本梯度值，并更新列表中 D 样本的梯度值，使用列表中梯度值均值，更新模型参数。
4. ...以此类推，直到算法收敛。

梯度下降法分类-特点

- 全梯度下降算法 FGD

  由于使用全部数据集，训练速度较慢

- 随机梯度下降算法 SGD

  简单，高效，不稳定。SG每次只使用一个样本迭代，若遇上噪声则容易陷入局部最优解

- 小批量梯度下降算法 mini-batch

  结合了 SG 的胆大和 FG 的心细，它的表现也正好居于 SG 和 FG 二者之间。

  目前使用最多，正是因为它避开了 FG 运算效率低成本大和 SG 收敛效果不稳定的缺点

- 随机平均梯度下降算法 SAG

  训练初期表现不佳，优化速度较慢。这是因为我们常将初始梯度设为0，

  而 SAG 每轮梯度更新都结合了上一轮梯度值。

- 目前使用较多的是：小批量梯度下降

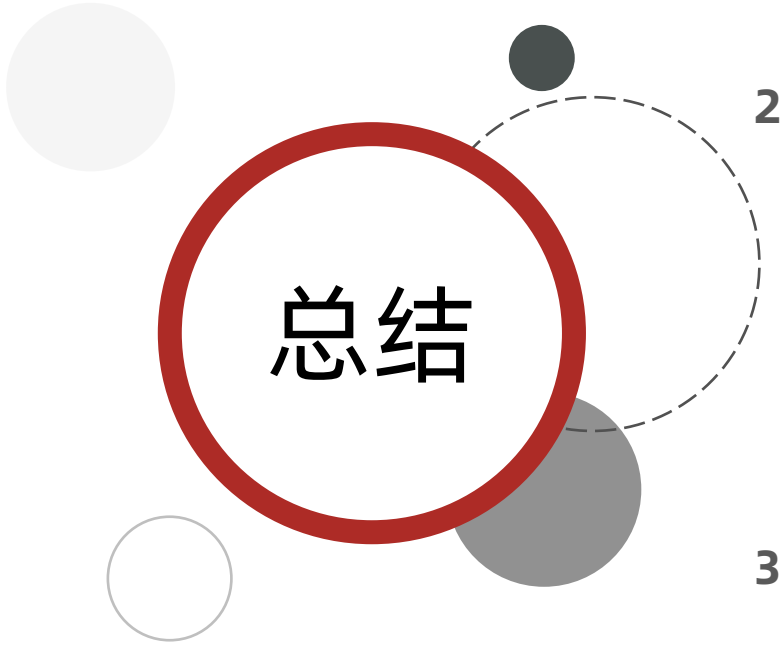
梯度下降法与正规方程对比

正规方程

- 不需要学习率
- 一次运算得出，一蹴而就
- 应用场景：小数据量场景、精准的数据场景
- 缺点：计算量大、容易收到噪声、特征强相关性的影响
- 注意：X^TX的逆矩阵不存在时，无法求解
- 注意：计算X^TX的逆矩阵非常耗时
- 如果数据规律不是线性的，无法使用或效果不好

梯度下降

- 需要选择学习率
- 需要迭代求解
- 特征数量较大可以使用
- 应用场景：更加普适，迭代的计算方式，适合于嘈杂、大数据应用场景
- 注意：梯度下降在各种损失函数（目标函数）求解中大量使用。深度学习中更是如此，深度学习模型参数很轻松就上亿，只能通过迭代的方式求最优解。



1 梯度概念

- 梯度的方向就是上升最快的方向
- 梯度：单变量函数就是导数；多变量函数就是偏导数

2 梯度下降法

- 梯度下降公式：
  - w = w - a * 梯度
  - $(w_1, w_2, \dots w_n) = (w_1, w_2, \dots w_n)$ - a * (梯度1，梯度2... 梯度n)
- 学习率 俗称步长
  - 学习率太小学的慢；学习率太大，梯度下降中抖动、震荡、易错过最优解

3. 梯度下降算法分类

1. 全梯度下降(FGD)：使用全部样本的梯度值
2. 随机梯度下降(SGD)：随机选择一个样本的梯度值更新权重
3. 小批量梯度下降(Mini-Batch)：随机选择小部分样本的梯度值更新权重
4. 随机平均梯度下降(SAG)：使用以前产生的梯度值来指导当前的梯度计算

练习

1、关于损失函数下列说法正确的是？（**多选**）

A）损失函数(Loss Function)又被称为代价函数(Cost Function)、目标函数

B）它是模型输出（预测值）和观测结果（真实值）之间概率分布差异的量化

C）线性回归的损失函数形如：

$$J(w) = (h(x_1) - y_1)^2 + (h(x_2) - y_2)^2 + \ldots + (h(x_n) - y_n)^2 = \sum_{i=1}^{n}(h(x_i) - y_i)^2$$

D）线性回归可采用的是MSE来衡量模型的损失

答案：ABCD

2、 关于正规方程的说法正确的是（多选）？ $w = (X^T X)^{-1} X^T y$

A）它是线性回归中参数向量w的解析式，通过损失函数求解而来

B）方阵 X^TX 必须是可逆的，否则无法求解

C）使用正规方程求解最优参数时，它的计算规模随着数据维度的增加而增加

D）X 是特征矩阵，y是目标值向量

答案：ABCD

练习

3、关于梯度下降说法正确的是？（**多选**） $w_{i+1} := w_i - \alpha \frac{\partial}{\partial w_i} J(w)$

A）目的是求解一组权重 w 的值，使得关于 w 的函数 J(w) 取得最小值

B）梯度的本质是一个矢量

C）沿着梯度的方向是函数值下降最快的方向

D）权重的迭代公式中步长需要手动设定，不可过大或过小

答案：ABD

4、 下列关于其它常见的梯度下降方法的描述正确的是？

A）全梯度下降每次更新权重都要使用全部的数据集数据

B）随机梯度下降每次更新权重只需要使用数据集中某一个样本的数据

C）小批量梯度下降法综合了FGD和SGD的优势，缓解了两者的缺陷

D）SAG在任何情况下都比其它梯度下降方法表现要好

答案：ABC

D错误，每个方法有自己的应用场景

目录

Contents

学习目标

Learning Objectives

1. 掌握常用的回归评估方法
2. 了解不同评估方法的特点

线性回归模型评估 - MAE、MSE、RMSE三种指标

- **为什么要进行线性回归模型的评估**
  - 我们希望衡量预测值和真实值之间的差距，
  - 会用到MAE、MSE、RMSE多种测评函数进行评价

- **平均绝对误差 Mean Absolute Error (MAE)**

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

n 为样本数量, y 为实际值, ŷ为预测值

MAE 越小模型预测约准确

from sklearn.metrics import mean_absolute_error

mean_absolute_error(y_test,y_predict)

- **均方误差 Mean Squared Error (MSE)**

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

n 为样本数量, y 为实际值, ŷ为预测值

MSE 越小模型预测约准确

from sklearn.metrics import mean_squared_error

mean_squared_error(y_test,y_predict)

- **均方根误差 Root Mean Squared Error (RMSE)**

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

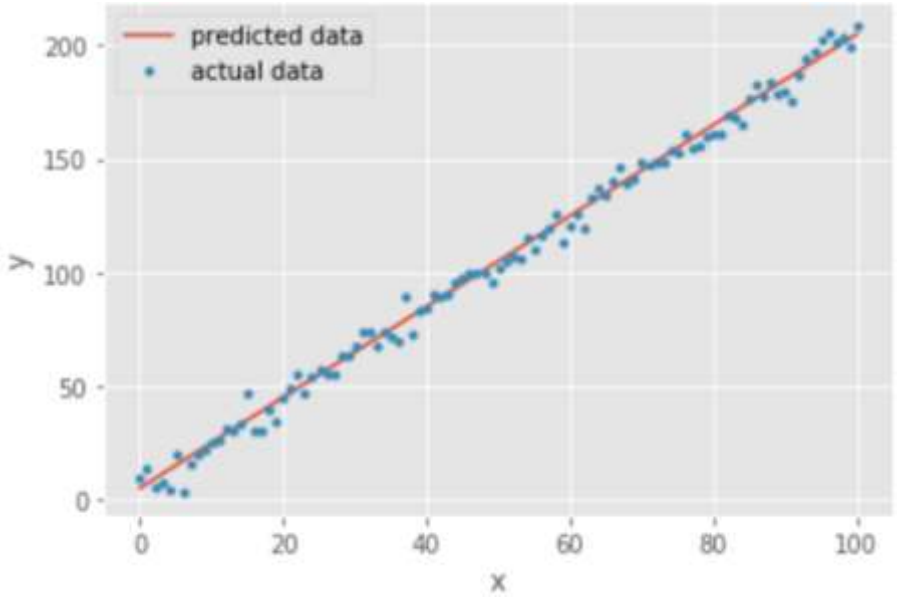
n 为样本数量, y 为实际值, ŷ为预测值

RMSE 越小模型预测约准确

RMSE 是 MSE 的平方根，某些情况下比MES更有用

线性回归模型评估 - MAE、MSE、RMSE三种指标对比

- 举个栗子：我们绘制了一条直线 y = 2x +5 用来拟合 y = 2x + 5 + e. 这些数据点，其中e为噪声



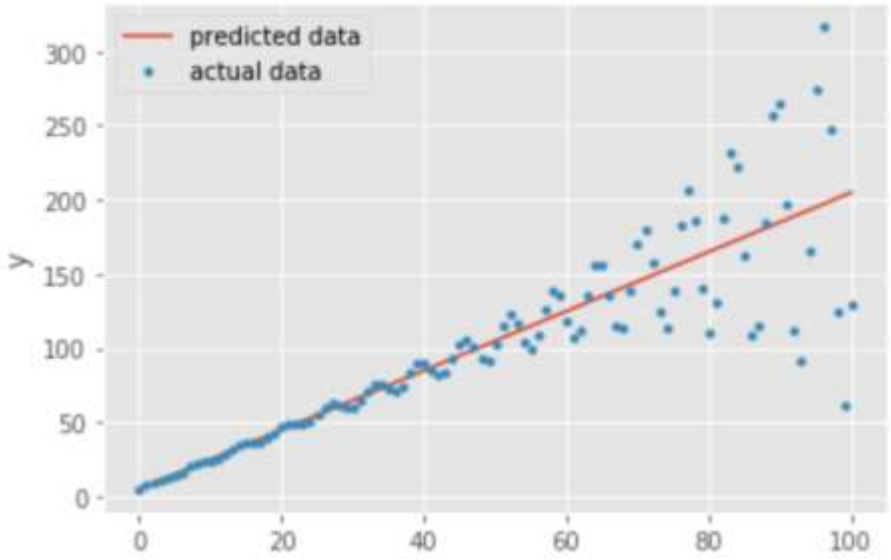

```
MAE = 3.7301886749473785
MSE = 22.527408286222677
RMSE = 4.746304697996398
```

- MAE 和 RMSE 非常接近，都表明模型的误差很低。MAE 或 RMSE 越小，误差越小！

- RMSE的计算公式中有一个平方项，因此大的误差将被平方，因此会增加 RMSE 的值，大多数情况下RMSE>MAE，
  比如两个误差大小为1, 3求MAE和RMSE。 **MAE:** (1+3)/2 = 2 **RMSE:** $\sqrt{(1^2+3^2)/2} = \sqrt{10/2} = \sqrt{5} = 2.236$
- RMSE 会放大预测误差较大的样本对结果的影响，而 MAE 只是给出了平均误差。
- 结论： RMSE > MAE都能反应真实误差，但是RMSE会对异常点更加敏感。

线性回归模型评估 - MAE、MSE、RMSE三种指标对比2

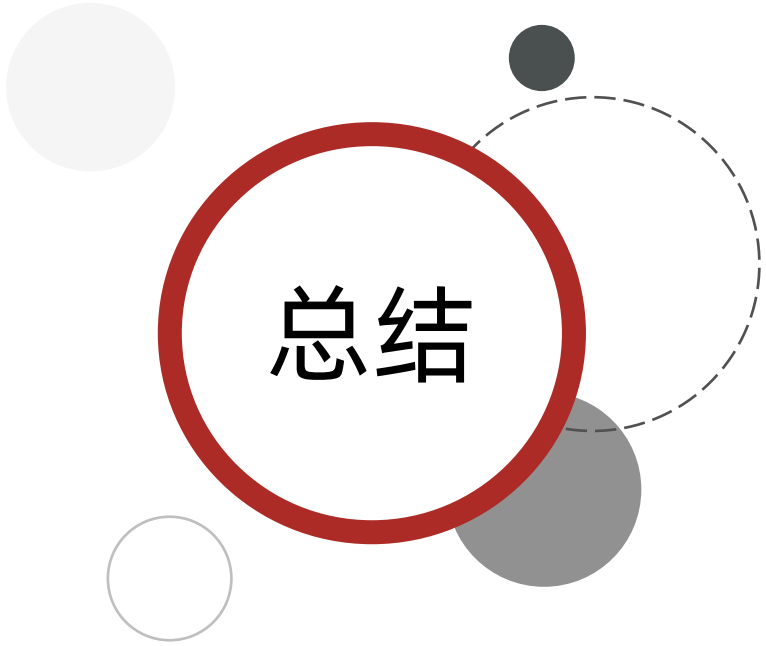
- 再举个栗子：橙色线 y = 2x +5。蓝色的点 **y = y + sin(x)*exp(x/20) + e** 其中 exp() 表示指数函数




```
MAE = 19.138201842683475
MSE = 1147.8294224102203
RMSE = 33.87963137949143
```

- 对比第一张图，所有指标都变大了，RMSE 几乎是 MAE 值的两倍，因为它对预测误差较大的点比较敏感
- MAE和RMSE同样是真实反应误差，能不能说RMSE是更好的指标？我们只按照RMSE指标来优化模型，让模型的误差会降的更低？

- 一直按照RMSE指标来训练模型，如果RMSE指标训练的非常低，说明什么？
- 说明模型对异常点（对噪声）也拟合的非常好。这样模型就容易过拟合了。所以评价指标要综合的看



1 一般使用MAE 和 RMSE 这两个指标

- MAE反应的是“真实”的平均误差，RMSE会将误差大的数据点放大
- MAE 不能体现出误差大的数据点，RMSE放大大误差的数据点对指标的影响，但是对异常数据比较敏感

2 综合结论

- 都能反映出预测值和真实值之间的误差
- MAE对误差大小不敏感
- RMSE会放大预测误差较大的样本的影响
- RMSE对异常数据敏感

目录

Contents

线性回归正规方程API和梯度下降API

- sklearn提供两种实现的API， 根据选择使用

（1）sklearn.linear_model.LinearRegression(fit_intercept=True)

- 通过正规方程优化
- 参数：fit_intercept，是否计算偏置
- 属性：LinearRegression.coef_ （回归系数） LinearRegression.intercept_（偏置）

（2）sklearn.linear_model.SGDRegressor(loss="squared_loss", fit_intercept=True, learning_rate ='constant', eta0=0.01)

- SGDRegressor类实现了随机梯度下降学习，它支持不同的损失函数和正则化惩罚项，来拟合线性回归模型。
- 参数
  - loss（损失函数类型）eg：loss = squared_loss
  - fit_intercept（是否计算偏置）
  - learning_rate （学习率策略）： string, optional ， 可以配置学习率随着迭代次数不断减小
    - 比如：学习率不断变小策略： ‘invscaling’: eta = eta0 / pow(t, power_t=0.25)
  - eta0=0.01 （学习率的值）
- 属性
  - SGDRegressor.coef_ （回归系数）SGDRegressor.intercept_ （偏置）

波士顿房价预测

- 1 案例背景 - 数据来源

实例数量:	506
属性数量:	13 数值型或类别型，帮助预测的属性

:中位数（第14个属性）经常是学习目标

属性信息 (按顺序):	• CRIM 城镇人均犯罪率 • ZN 占地面积超过2.5万平方英尺的住宅用地比例 • INDUS 城镇非零售业务地区的比例 • CHAS 查尔斯河虚拟变量 (= 1 如果土地在河边；否则是0) • NOX 一氧化氮浓度（每1000万份） • RM 平均每居民房数 • AGE 在1940年之前建成的所有者占用单位的比例 • DIS 与五个波士顿就业中心的加权距离 • RAD 辐射状公路的可达性指数 • TAX 每10,000美元的全额物业税率 • PTRATIO 城镇师生比例 • B 1000(Bk - 0.63)^2 其中 Bk 是城镇的黑人比例 • LSTAT 人口中地位较低人群的百分数 • MEDV 以1000美元计算的自有住房的中位数
缺失属性值:	无
创建者:	Harrison, D. and Rubinfeld, D.L.

这是UCI ML（欧文加利福尼亚大学 机器学习库）房价数据集的副本。 http://archive.ics.uci.edu/ml/datasets/Housing

该数据集是从位于卡内基梅隆大学维护的StatLib图书馆取得的。

UCI Machine Learning Repository

波士顿房价预测

- 2 案例分析

  回归当中的数据大小不一致，是否会导致结果影响较大。所以需要做标准化处理。

  - 数据分割与标准化处理
  - 回归预测
  - 线性回归的算法效果评估

- 3 回归性能评估

  均方误差(Mean Squared Error)MSE)评价机制：

  $$MSE = \frac{1}{m}\sum_{i=1}^{m}(y^i - \bar{y})^2$$

  sklearn.metrics.mean_squared_error(y_true, y_pred)

  均方误差回归损失

  - y_true:真实值
  - y_pred:预测值
  - return:浮点数结果

波士顿房价预测

- 导入库

```
# 1.导入依赖包
# from sklearn.datasets import load_boston   # 数据集已废弃
from sklearn.preprocessing import StandardScaler        # 特征处理
from sklearn.model_selection import train_test_split    # 数据集划分
from sklearn.linear_model import LinearRegression       # 正规方程的回归模型
from sklearn.linear_model import SGDRegressor           # 梯度下降的回归模型
from sklearn.metrics import mean_squared_error          # 均方误差评估

from sklearn.linear_model import Ridge, RidgeCV

import pandas as pd
import numpy as np
```

波士顿房价预测

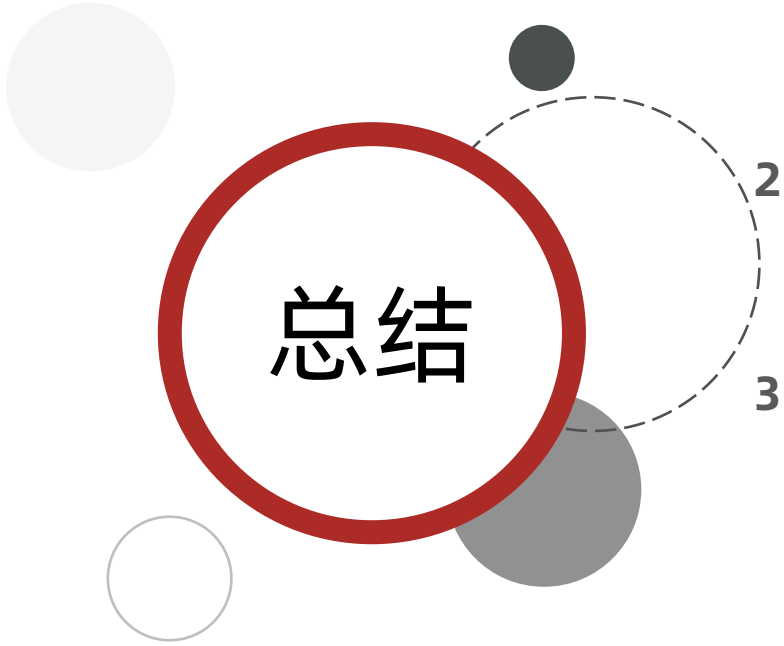
- 正规方程法

```
# 正规方程法
def linear_model1():
    # 2.数据预处理
    # 2.1 获取数据
    data_url = "http://lib.stat.cmu.edu/datasets/boston"
    raw_df = pd.read_csv(data_url, sep="\s+", skiprows=22, header=None)
    data = np.hstack([raw_df.values[::2, :], raw_df.values[1::2, :2]])
    target = raw_df.values[1::2, 2]

    # 2.2 数据集划分
    x_train, x_test, y_train, y_test = train_test_split(data, target, random_state=22)

    # 2.3 特征工程-标准化
    transfer = StandardScaler()
    x_train = transfer.fit_transform(x_train)
    x_test = transfer.transform(x_test)

    # 3.模型训练，机器学习-线性回归
    # 3.1 实例化模型(正规方程)
    estimator = LinearRegression()
    # 3.2 模型训练
    estimator.fit(x_train, y_train)

    # 4.模型预测
    y_predict = estimator.predict(x_test)
    print("预测值为:", y_predict)
    print("模型的权重系数为:", estimator.coef_)

    # 5.模型评估，均方误差
    error = mean_squared_error(y_test, y_predict)
    print("误差为:", error)
```

波士顿房价预测

- 梯度下降法

```
# 梯度下降法
def linear_model2():
    # 2.数据预处理
    # 2.1 获取数据
    data_url = "http://lib.stat.cmu.edu/datasets/boston"
    raw_df = pd.read_csv(data_url, sep="\s+", skiprows=22, header=None)
    data = np.hstack([raw_df.values[::2, :], raw_df.values[1::2, :2]])
    target = raw_df.values[1::2, 2]
    # 2.2 数据集划分
    x_train, x_test, y_train, y_test = train_test_split(data, target, random_state=22)
    # 2.3 特征工程-标准化
    transfer = StandardScaler()
    x_train = transfer.fit_transform(x_train)
    x_test = transfer.transform(x_test)

    # 3.模型训练，机器学习-线性回归
    # 3.1 实例化模型(梯度下降法)
    estimator = SGDRegressor()
    # estimator = SGDRegressor(max_iter=1000, learning_rate="constant", eta0=0.001)
    # 3.2 模型训练
    estimator.fit(x_train, y_train)

    # 4.模型预测
    y_predict = estimator.predict(x_test)
    print("预测值为:", y_predict)
    print("模型的权重系数为:", estimator.coef_)
    print("模型的偏置为:", estimator.intercept_)

    # 5.模型评估, 均方误差
    error = mean_squared_error(y_test, y_predict)
    print("误差为:", error)
```



1 正规方程

- sklearn.linear_model.LinearRegression()

2 梯度下降

- sklearn.linear_model.SGDRegressor(）

3 梯线性回归性能评估

- sklearn.metrics.mean_squared_error

练习

1、本案例中，以下哪个选项是用来评估线性回归模型的方法?

A) 最小二乘法

B) 均方误差

C) 平均绝对误差

D) 决定性系数

答案：B

目录

Contents

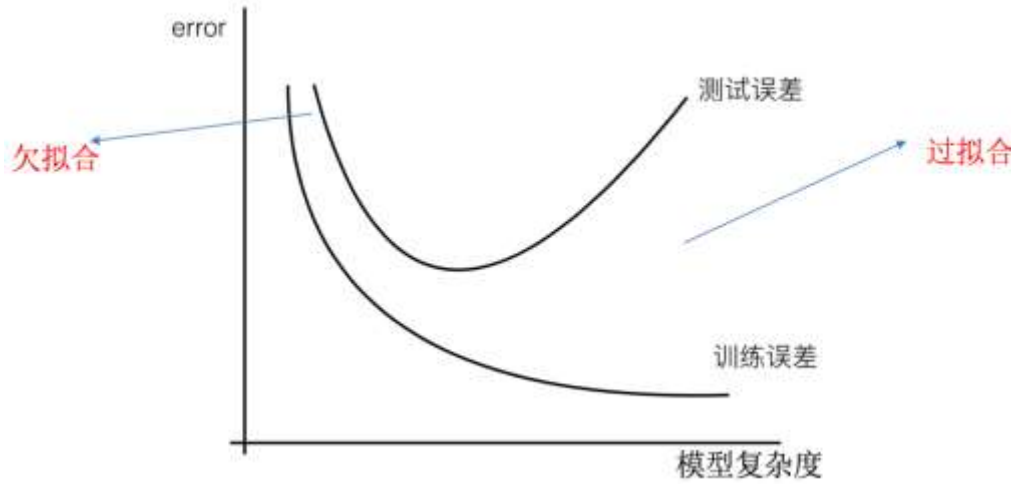
学习目标

Learning Objectives

1. 掌握过拟合、欠拟合的概念
2. 掌握过拟合、欠拟合产生的原因
3. 知道什么是正则化，以及正则化的方法

欠拟合与过拟合

- 欠拟合与过拟合概念复习



- 欠拟合：模型在训练集上表现不好，在测试集上也表现不好。模型过于简单
- 过拟合：模型在训练集上表现好，在测试集上表现不好。模型过于复杂

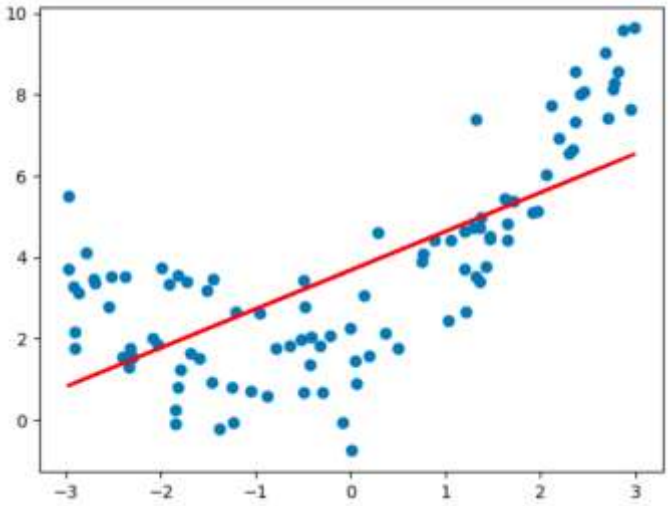
- 欠拟合在训练集和测试集上的误差都较大
- 过拟合在训练集上误差较小，而测试集上误差较大

欠拟合与过拟合

```python
# 1.导入依赖包
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error # 计算均方误差
from sklearn.model_selection import train_test_split

def dm01_模型欠拟合():
    # 2.准备数据x y(增加上噪声)
    np.random.seed(666)
    x = np.random.uniform(-3, 3, size=100)
    y = 0.5 * x ** 2 + x + 2 + np.random.normal(0, 1, size=100)

    # 3 训练模型
    # 3.1 实例化线性回归模型
    estimator = LinearRegression()
    # 3.2 模型训练
    X = x.reshape(-1, 1)
    estimator.fit(X, y)

    # 4 模型预测
    y_predict = estimator.predict(X)

    # 5 模型评估，计算均方误差
    # 5.1 模型评估MSE
    myret = mean_squared_error(y, y_predict)
    print('myret-->', myret)
    # 5.2 展示效果
    plt.scatter(x, y)
    plt.plot(x, y_predict, color='r')
    plt.show()
```

通过代码展示欠拟合

myret--> 3.0750025765636577



数据是抛物线非线性的，

用线性模型去拟合.。

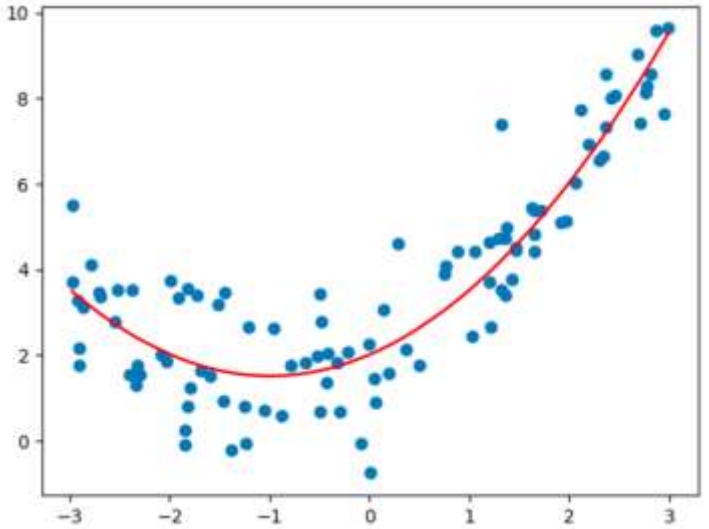
模型过于简单，出现欠拟合

欠拟合与过拟合

```
def dm02_模型ok():
    # 2.准备数据x y(增加上噪声)
    np.random.seed(666)
    x = np.random.uniform(-3, 3, size=100)
    y = 0.5 * x ** 2 + x + 2 + np.random.normal(0, 1, size=100)

    # 3.模型训练
    # 3.1 实例化线性回归模型
    estimator = LinearRegression()
    # 3.2 模型训练
    X = x.reshape(-1, 1)
    # print('X.shape-->', X.shape)
    X2 = np.hstack([X, X ** 2]) # 数据增加二次项
    estimator.fit(X2, y)

    # 4.模型预测
    y_predict = estimator.predict(X2)

    # 5.模型评估，计算均方误差
    myret = mean_squared_error(y, y_predict)
    print('myret-->', myret)

    # 6 展示效果
    plt.scatter(x, y)
    # 画图plot折线图时 需要对x进行排序, 取x排序后对应的y值
    plt.plot(np.sort(x), y_predict[np.argsort(x)], color='r')
    plt.show()
```

通过代码展示正好拟合

myret--> 1.0987392142417856



数据是一元二次方程抛物线形状的，

给模型送入的数据，增加x^2项特征，

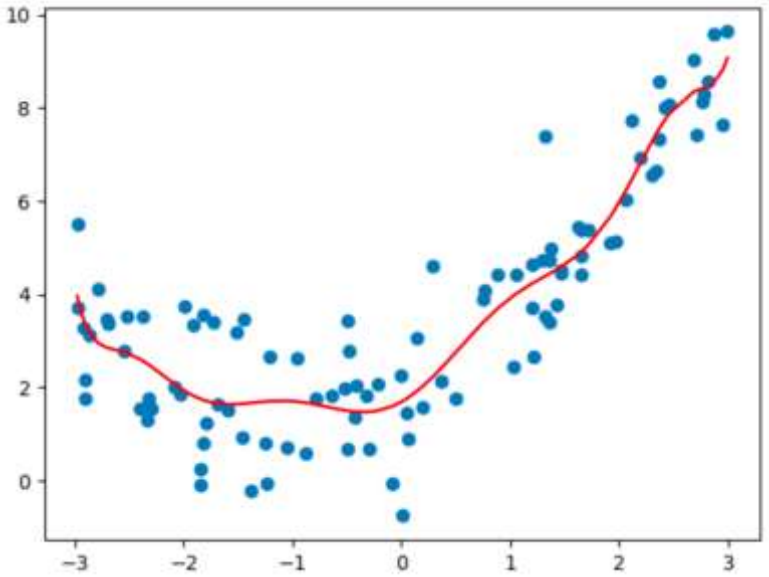
再用线性模型去拟合。模型很ok

欠拟合与过拟合

```
def dm03_模型过拟合():
    # 2.准备数据x y(增加上噪声)
    np.random.seed(666)
    x = np.random.uniform(-3, 3, size=100)
    y = 0.5 * x ** 2 + x + 2 + np.random.normal(0, 1, size=100)

    # 3 训练模型
    # 3.1 实例化线性回归模型
    estimator = LinearRegression()
    # 3.2 模型训练
    X = x.reshape(-1, 1)
    # print('X.shape-->', X.shape)
    X3 = np.hstack([X, X**2, X**3, X**4, X**5, X**6, X**7, X**8, X**9, X**10])  # 数据增加高次项
    estimator.fit(X3, y)

    # 4.模型预测
    y_predict = estimator.predict(X3)

    # 5.模型评估，计算均方误差
    # 5.1 模型评估MSE
    myret = mean_squared_error(y, y_predict)
    print('myret-->', myret)
    # 5.2 展示效果
    plt.scatter(x, y)
    # 画图时输入的x数据: 要求是从小到大
    plt.plot(np.sort(x), y_predict[np.argsort(x)], color='r')
    plt.show()
```

通过代码展示过拟合

myret--> 1.0508466763764124



数据是抛物线形状的，

给模型送入的数据，增加x^2、x^3、x^4 ...高次项特征，

再用线性模型去拟合。模型过于复杂，出现过拟合

欠拟合与过拟合 - 出现原因和解决方案

- 欠拟合出现的原因
  - 学习到数据的特征过少

- 解决办法 【从数据、模型、算法的角度去想解决方案】
  - 添加其他特征
    - 有时出现欠拟合是因为特征项不够导致的，可以添加其他特征项来解决
    - “组合”、“泛化”、“相关性”三类特征是特征添加的重要手段
  - 添加多项式特征项
    - 模型过于简单时的常用套路，例如将线性模型通过添加二次项或三次项使模型泛化能力更强

欠拟合与过拟合 - 出现原因和解决方案

- 过拟合出现的原因
  - 原始特征过多，存在一些嘈杂特征， 模型过于复杂是因为模型尝试去兼顾各个测试数据点

- 解决办法
  - 重新清洗数据
    - 对于过多异常点数据、数据不纯的地方再处理
  - 增大数据的训练量
    - 对原来的数据训练的太过了，增加数据量的情况下，会缓解
  - 正则化
    - 解决模型过拟合的方法，在机器学习、深度学习中大量使用
  - 减少特征维度，防止维灾难
    - 由于特征多，样本数量少，导致学习不充分，泛化能力差。

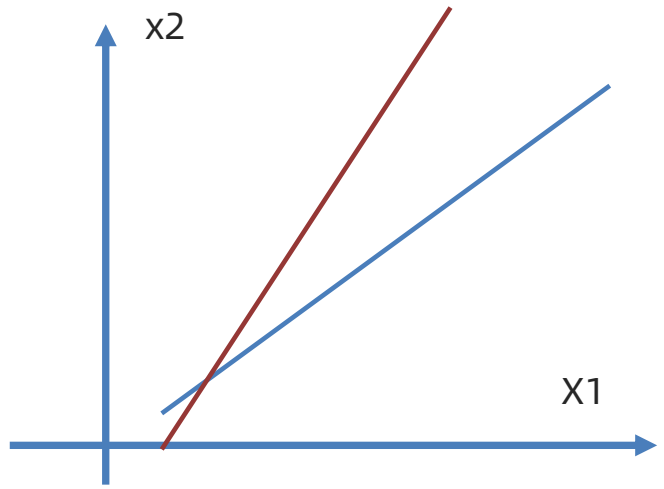
欠拟合与过拟合 - 正则化

- 正则化概念（出现的原因）

  在模型训练时，数据中有些特征影响模型复杂度、或者某个特征的异常值较多，所以要尽量减少这个特征的影响（甚至删除某个特征的影响），这就是正则化。

- 正则化如何消除异常点带来的w值过大过小的影响？

  在损失函数中增加正则化项，分为L1正则化、L2正则化



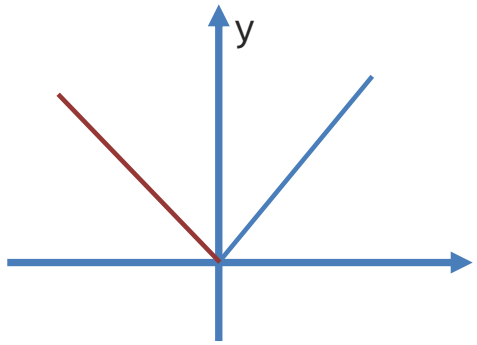
红色异常点A(x1,x2), 因x2过大，会让对应的权重系数k2过小

欠拟合与过拟合 - L1正则化

- L1正则化，在损失函数中添加L1正则化项

$$J(w) = \text{MSE}(w) + \alpha \sum_{i=1}^{n} |w_i|$$

- α 叫做惩罚系数，该值越大则权重调整的幅度就越大，即：表示对特征权重惩罚力度就越大
- L1 正则化会使得权重趋向于 0，甚至等于 0，使得某些特征失效，达到特征筛选的目的



y = |x| 图像的导函数为分段函数

x大于0 导数为 1

x小于0 导数为-1

x 等于- 导数为 0

- 使用 L1 正则化的线性回归模型是 Lasso 回归

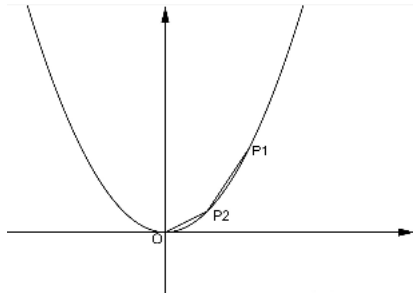
```
from sklearn.linear_model import Lasso
```

欠拟合与过拟合 - L2正则化

- L2正则化，在损失函数中添加L2正则化项

$$J(w) = \text{MSE}(w) + \alpha \sum_{i=1}^{n} w_i^2$$

  - α 叫做惩罚系数，该值越大则权重调整的幅度就越大，即：表示对特征权重惩罚力度就越大
  - L2 正则化会使得权重趋向于 0，一般不等于 0



  - 使用 L2 正则化的线性回归模型是岭回归

```
from sklearn.linear_model import Ridge
```

欠拟合与过拟合-Lasso回归

```
# 1.导入依赖包
from sklearn.linear_model import Lasso

def dm04_模型过拟合_L1正则化():
    # 2.准备数据x y(增加上噪声)
    np.random.seed(666)
    x = np.random.uniform(-3, 3, size=100)
    y = 0.5 * x ** 2 + x + 2 + np.random.normal(0, 1, size=100)

    # 3 训练模型
    # 3.1 实例化L1正则化模型 做实验:alpha惩罚力度越来越大,k值越来越小,返回会欠拟合
    estimator = Lasso(alpha=0.005, normalize=True)
    # 3.2 模型训练
    X = x.reshape(-1, 1)
    X3 = np.hstack([X, X ** 2, X ** 3, X ** 4, X ** 5, X ** 6, X ** 7, X ** 8, X ** 9, X ** 10])  # 数据增加二次项
    estimator.fit(X3, y)
    print('estimator.coef_', estimator.coef_)

    # 4.模型预测
    y_predict = estimator.predict(X3)

    # 5.模型评估，计算均方误差
    # 5.1 模型评估MSE
    myret = mean_squared_error(y, y_predict)
    print('myret-->', myret)
    # 5.2 展示效果
    plt.scatter(x, y)
    # 画图时输入的x数据: 要求是从小到大
    plt.plot(np.sort(x), y_predict[np.argsort(x)], color='r')
    plt.show()
```

- 对过拟合模型L1正则化调整

Lasso回归L1正则

会将高次方项系数变为0

欠拟合与过拟合-Ridge回归

```
# 1.导入依赖包
from sklearn.linear_model import Ridge
def dm05_模型过拟合_L2正则化():
    # 2.准备数据x y(增加上噪声)
    np.random.seed(666)
    x = np.random.uniform(-3, 3, size=100)
    y = 0.5 * x ** 2 + x + 2 + np.random.normal(0, 1, size=100)

    # 3.训练模型
    # 3.1 实例化L2正则化模型
    estimator = Ridge(alpha=0.005, normalize=True)
    # 3.2 模型训练
    X = x.reshape(-1, 1)
    X3 = np.hstack([X, X ** 2, X ** 3, X ** 4, X ** 5, X ** 6, X ** 7, X ** 8, X ** 9, X ** 10]) # 数据增加二次项
    estimator.fit(X3, y)
    print('estimator.coef_', estimator.coef_)

    # 4.模型预测
    y_predict = estimator.predict(X3)

    # 5.模型评估，计算均方误差
    # 5.1 模型评估，MSE
    myret = mean_squared_error(y, y_predict)
    print('myret-->', myret)
    # 5.2 展示效果
    plt.scatter(x, y)
    # 画图时输入的x数据: 要求是从小到大
    plt.plot(np.sort(x), y_predict[np.argsort(x)], color='r')
    plt.show()
```

- 对过拟合模型L2正则化调整

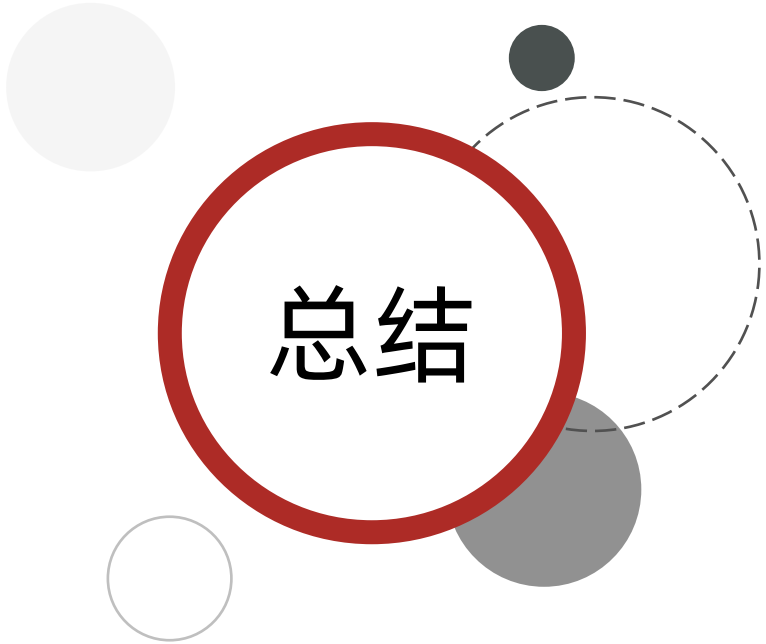
  Ridge线性回归L2正则

  不会将系数变为0

  但是对高次方项系数影响较大

- 工程开发中L1、L2使用建议：

  一般倾向使用L2正则。

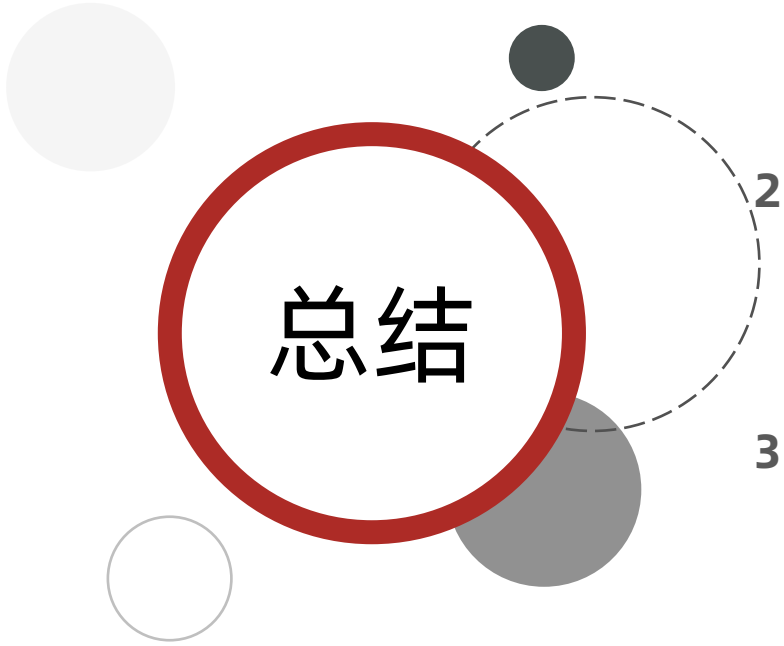


1 欠拟合

- 在训练集上表现不好，在测试集上表现不好
- 解决方法，继续学习
  - 1 添加其他特征项
  - 2 添加多项式特征

2 过拟合

- 在训练集上表现好，在测试集上表现不好
- 解决方法
  - 1 重新清洗数据集
  - 2 增大数据的训练量
  - 3 正则化
  - 4 减少特征维度



1 正则化

- 异常点数据会造成权重系数过大、过小。尽量减少这些特征的影响（甚至删除某个特征的影响），这就是正则化。
- 为了减少过拟合的影响，控制模型的参数。尤其是高次项的权重参数

2 L1正则化

- 会直接把高次项前面的系数变为0
- Lasso回归

3 L2正则化

- 把高次项前面的系数变成特别小的值
- 岭回归（Ridge回归）

练习

1、下列关于欠拟合与过拟合的描述正确的是？

A）欠拟合：模型学习到的特征过少，无法准确的预测未知样本

B）过拟合：模型学习到的特征过多，导致模型只能在训练样本上得到较好的预测结果，而在未知样本上的效果不好

C）欠拟合可以通过增加特征来解决

D）过拟合可以通过正则化、异常值检测、特征降维等方法来解决

答案解析：A欠拟合出现的原因 B过拟合出现的原因 C增加模型的复杂度 D减低模型复杂度。

答案：ABCD

练习

2 下列关于过拟合问题的解决方式以及描述正确的是？

A）使用岭回归能够防止训练所得的模型发生过拟合

B）使用 Lasso 回归也能防止模型产生过拟合，这时所得模型的权重系数部分为0

C）L2正则化能够让模型产生一些平滑的权重系数

D）Early stopping 是当模型训练到某个固定的验证错误率阈值时，及时停止模型训练

答案：ABCD

练习

填空：

sklearn.linear_model.Ridge() 岭回归的API中：

① alpha表示正则化系数，正则化系数越大，表示正则化力度 _____ ，

所得模型的权重系数 _____ ；反之，所得模型的权重系数 _____ 。

答案：① 越大；越小；越大

sklearn.linear_model.SGDRegressor() 使用随机梯度下降法优化的线性回归API：

② 当它的参数 penalty 为 l2 、参数 loss 为 squared_loss 时，达到的效果与上述的

岭回归API相同，只不过 SGDRegressor 只能使用 _____ 去优化损失，而 Ridge

的选择则更加丰富。

答案：② 普通的随机梯度下降法


黑马程序员
www.itheima.com
传智教育旗下高端IT教育品牌