回归：一元线性回归（解析解与梯度下降算法）

覃雄派

回归：一元线性回归(解析解与梯度下降算法)

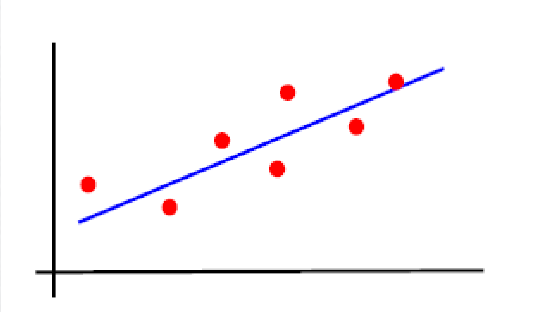
提纲

- 回归问题、模型、建立模型的三个步骤
- 常数模型（平均平方误差|平均绝对误差）
- 一元线性回归（解析解|梯度下降法求解）
- 梯度下降法的讨论

回归：一元线性回归（解析解与梯度下降算法）

- 回归（Regression）问题
  - 本周的股票价格如何变化?
  - 下礼拜一的气温会是多少?
  - 中国第一季度的GDP增长会是多少?
  - 估计回归的参数，如权重

How **much** or How **Many**?



解决方法：建立模型！

回归：一元线性回归（解析解与梯度下降算法）

- 什么是模型（Model）
  - 模型是对现实世界的一种“有用”的简化
  - Model is a useful simplification of reality

- 例子：重力公式
  - $G = mg, g = 9.81$
  - 上述模型简化了以下因素：
  - 不同地区的重力差异
  - 空气阻力
  - 等等

Essentially, all models are wrong, but some are **useful**.

-- George Box (1919 - 2013)

回归：一元线性回归（解析解与梯度下降算法）

- 建立模型的三个步骤
- **Step（1）选择某种模型**
  - 常数模型 – Constant Model
  - 线性回归模型 – Linear Regression Model
  - 更复杂的模型
- **Step（2）选择目标函数**
  - 均方误差（mean square error, MSE）
  - 平均绝对误差（mean absolute error, MAE）
  - 其它目标函数
- **Step（3） 拟合模型（model fitting）：优化目标函数**
  - 最小化/最大化目标函数

回归：一元线性回归（解析解与梯度下降算法）

- 符号定义

符号	符号含义
y	真实的数据值（如小费） • 第i项数据值表示为y_i • 数据集表示为$\{y_1, y_2, \dots y_n\}$
$\hat{y}$	预测的数据值（如预测的小费） • 第i项数据的预测值表示为$\hat{y}_i$
θ	模型的参数（Parameter） • 如“真实”的小费
$\hat{\theta}$	模型的**拟合参数**（fitted parameters） • 我们需要求解的**目标！**

回归：一元线性回归（解析解与梯度下降算法）

- 概念辨析：请说出以下两个概念的区别和联系
  - 估计 Estimation
  - 预测 Prediction

思考：应该如何使用观测数据来估计参数?

- 估计（Estimation）是使用观测到的数据来**拟合参数**

$$\hat{\theta} = f_1(y, x)$$

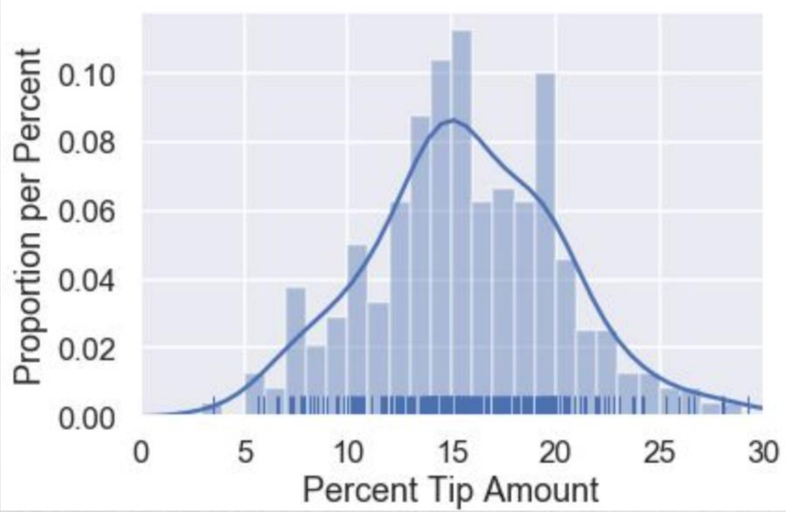
- 预测（Prediction）是使用拟合的参数来**求解未知的数据**

$$\hat{y}_i = f_2(\hat{\theta}, x_i)$$

回归：一元线性回归（解析解与梯度下降算法）

回归：一元线性回归（解析解与梯度下降算法）

- 预测任务：构建模型预测某个数值型变量的取值
  - 预测《数据科学导论》课程学生的GPA
  - 预测北京春天的空气指数
  - 预测在餐厅给服务员多少小费（餐费的百分之几）



- 左图：**244**单小费的分布图
- 常数模型：忽略其他因素，使用同一数值进行预测
  - 15%似乎比25%更好
  - 15%比14%更好吗？
- 我们应该如何确定**最优**的数值？

回归：一元线性回归（解析解与梯度下降算法）

- 常数模型
  - 我们将常数模型表示为：

$$\hat{y} = \theta$$

- 几个基本概念：
  - 参数θ：我们使用参数来定义模型，用来描述输入数据与输出数据之间的关系
    - 注：一些模型不含有参数（nonparametric）
  - 我们的常数模型忽略了输入数据x
  - 我们后面会学习更多的模型：
    - 线性回归模型：$\hat{y} = \theta_0 + \theta_1 \cdot x$
    - 逻辑斯蒂回归模型：$\hat{y} = \frac{1}{1+e^{(-x^T\theta)}}$
- 模型求解目标：找到最优的参数值，**表示为$\hat{\theta}$**

x	y	$\hat{y}$
x_1	y_1	$\hat{y}_1$
x_2	y_2	$\hat{y}_2$
…	…	…
x_n	y_n	$\hat{y}_n$

回归：一元线性回归（解析解与梯度下降算法）

- 常数模型：**损失函数（Loss Function）**
  - 度量模型预测的优劣，即**真实值y_i与预测值$\hat{y}_i$之间的差异**
  - 针对我们的常数模型，度量参数θ与真实观测值之间的误差

- 平方损失（Squared Loss），也称为L2损失

$$L_2(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$$

- 绝对损失（Absolute Loss），也称为L1损失

$$L_1(y_i, \hat{y}_i) = |y_i - \hat{y}_i|$$

x	y	$\hat{y}$
x_1	y_1	$\hat{y}_1$
x_2	y_2	$\hat{y}_2$
…	…	…
x_n	y_n	$\hat{y}_n$

回归：一元线性回归（解析解与梯度下降算法）

- 损失函数与经验风险
  - 给定某个数据集，我们可以度量**平均损失**，也称**经验风险**（Empirical Risk）或目标函数（Objective Function）

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

  - 模型的平均损失度量了模型对于数据的拟合程度
  - 两种典型的平均损失
    - 均方误差（mean squared error, **MSE**）
    - 平均绝对误差（mean absolute error, **MAE**）
  - 模型求解的目标：最小化**平均损失!**

x	y	$\hat{y}$
x_1	y_1	$\hat{y}_1$
x_2	y_2	$\hat{y}_2$
…	…	…
x_n	y_n	$\hat{y}_n$

回归：一元线性回归（解析解与梯度下降算法）

- 常数模型
- 均方误差与平均绝对误差
  - 均方误差：采用均方损失函数，针对所有数据点求平均

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

  - 平均绝对误差：采用平均绝对损失函数，针对所有数据点求平均

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

x	y	$\hat{y}$
x_1	y_1	$\hat{y}_1$
x_2	y_2	$\hat{y}_2$
…	…	…
x_n	y_n	$\hat{y}_n$

回归：一元线性回归（解析解与梯度下降算法）

- 常数模型与均方误差
- 平均损失通常表示为参数θ的函数，例如：

$$R(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta)^2$$

- 思考：如何求解最小化平均损失时参数θ的取值

$$\hat{\theta} = \arg\min_{\theta}\frac{1}{n}\sum_{i=1}^{n}(y_i - \theta)^2$$

- 例子：给定五个数据点y_i为[20,21,22,29,33]，请根据上式求解$\hat{\theta}$

请计算下表单元格的分值

A. θ **=22** **B.** θ **=25** **C.** θ **=24**

回归：一元线性回归（解析解与梯度下降算法）

- 常数模型与均方误差
- 平均损失通常表示为参数θ的函数，例如：

$$R(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta)^2$$

- 思考：如何求解最小化平均损失时参数θ的取值

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta)^2$$

- 例子：给定五个数据点y_i为[20,21,22,29,33]，请根据上式求解$\hat{\theta}$

θ =22代入，
(4+1+0+49+121)/5=175/5

θ =25代入,
(25+16+9+16+64)/5=130/5

θ =24代入，
(16+9+4+25+81)/5=135/5

请计算下表单元格的分值
A. θ =22　B. θ =25　C. θ =24

代入算一算

回归：一元线性回归（解析解与梯度下降算法）

- 常数模型与均方误差：求解过程
- 均方误差：$R(\theta)=\frac{1}{n}\sum_{i=1}^{n}(y_i-\theta)^2$
- 针对参数θ求导：

$$\frac{d}{d\theta}R(\theta)=\frac{1}{n}\sum_{i=1}^{n}\frac{d}{d\theta}(y_i-\theta)^2=\frac{-2}{n}\sum_{i=1}^{n}(y_i-\theta)$$

- 令一阶导数等于0，得到

$$\frac{-2}{n}\sum_{i=1}^{n}(y_i-\theta)=0\Rightarrow\sum_{i=1}^{n}(y_i-\theta)=0$$

$$\Rightarrow\sum_{i=1}^{n}y_i=n\theta\Rightarrow\hat{\theta}=\frac{\sum_{i=1}^{n}y_i}{n}=\mathbf{mean}(y)$$

回归：一元线性回归（解析解与梯度下降算法）

- 常数模型与均方误差
- 当$\hat{\theta} = \textbf{mean}(y)$时，可以求得：

$$R(\hat{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sigma^2$$

- 几个结论：
  - 给定常数模型和均方误差，最优的参数估计是观测数据的**均值**
  - 给定均值作为估计，此时均方误差达到最小，等于观测数据的**方差**
  - 上述结论解释了为什么均值是重要的统计变量
- 注意：
  - 上述结论成立的条件：① 模型为常数；② 损失函数采用均方损失

回归：一元线性回归（解析解与梯度下降算法）

- 常数模型与平均绝对误差
- 平均损失通常表示为参数θ的函数，例如：

$$R(\theta) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \theta|$$

- 思考：如何求解最小化平均损失时参数θ的取值

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} |y_i - \theta|$$

- 例子：给定五个数据点y_i 为[20,21,22,29,33]，请根据上式求解$\hat{\theta}$

请计算下表单元格的分值

A. θ =22　B. θ =25　C. θ =24

回归：一元线性回归（解析解与梯度下降算法）

- 常数模型与平均绝对误差
- 平均损失通常表示为参数θ的函数，例如：

$$R(\theta)=\frac{1}{n}\sum_{i=1}^{n}|y_i-\theta|$$

- 思考：如何求解最小化平均损失时参数θ的取值

$$\hat{\theta}=\arg\min_{\theta}\frac{1}{n}\sum_{i=1}^{n}|y_i-\theta|$$

- 例子：给定五个数据点y_i为[20,21,22,29,33]，请根据上式求解$\hat{\theta}$

θ =22代入，
(2+1+0+7+11)/5=21/5

θ =25代入，
(5+4+3+4+8)/5=24/5

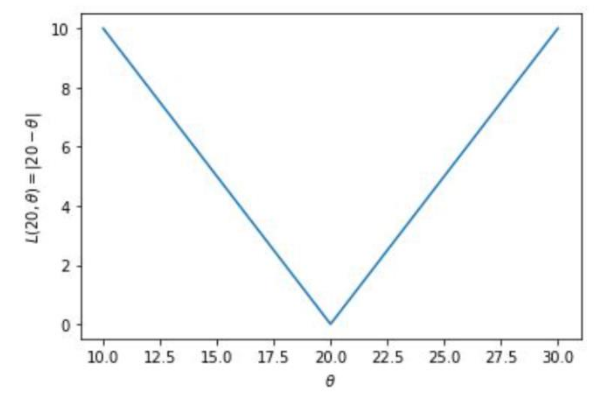
θ =24代入，
(4+3+2+5+9)/5=23/5

请计算下表单元格的分值
A. θ =22 B. θ =25 C. θ =24

代入算一算

回归：一元线性回归（解析解与梯度下降算法）

- 常数模型与平均绝对误差：求解过程
- 思考：求解的关键是如何计算$|y_i-\theta|$的导数



- 提示：将绝对值写成以下分段函数

$$|y_i-\theta| = \begin{cases} y_i-\theta & \textbf{if } \theta \le y \\ \theta - y_i & \textbf{if } \theta > y \end{cases}$$

- 同样地，可以将平均绝对损失针对θ的导数写为分段函数

$$\frac{d}{d\theta}|y_i-\theta| = \begin{cases} -1 & if\ \theta < y_i \\ 1 & if\ \theta > y_i \end{cases}$$

  - 注：$\theta = y_i$时不可导。此处为了简便，忽略该点

回归：一元线性回归（解析解与梯度下降算法）

- 常数模型与平均绝对误差：求解过程
- 平均绝对误差：$R(\theta)=\frac{1}{n}\sum_{i=1}^{n}|y_i-\theta|$
- 针对参数θ求导：

$$\frac{d}{d\theta}R(\theta)=\frac{1}{n}\sum_{i=1}^{n}\frac{d}{d\theta}|y_i-\theta|=\frac{1}{n}[\sum_{\theta<y_i}(-1)+\sum_{\theta>y_i}1]$$

- 令一阶导数等于0，得到

$$\frac{1}{n}\left[\sum_{\theta<y_i}(-1)+\sum_{\theta>y_i}1\right]=0$$

$$\Rightarrow \sum_{\theta<y_i}\mathbf{1}=\sum_{\theta>y_i}\mathbf{1}$$

思考：根据上述公式估计值$\hat{\theta}$应该取多少？

回归：一元线性回归（解析解与梯度下降算法）

- 常数模型与**平均绝对误差**
- 根据之前推导，可以求得：$\hat{\theta} = \textbf{median}(y)$，此时有：
  - $R(\hat{\theta}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \textbf{median}(y)|$

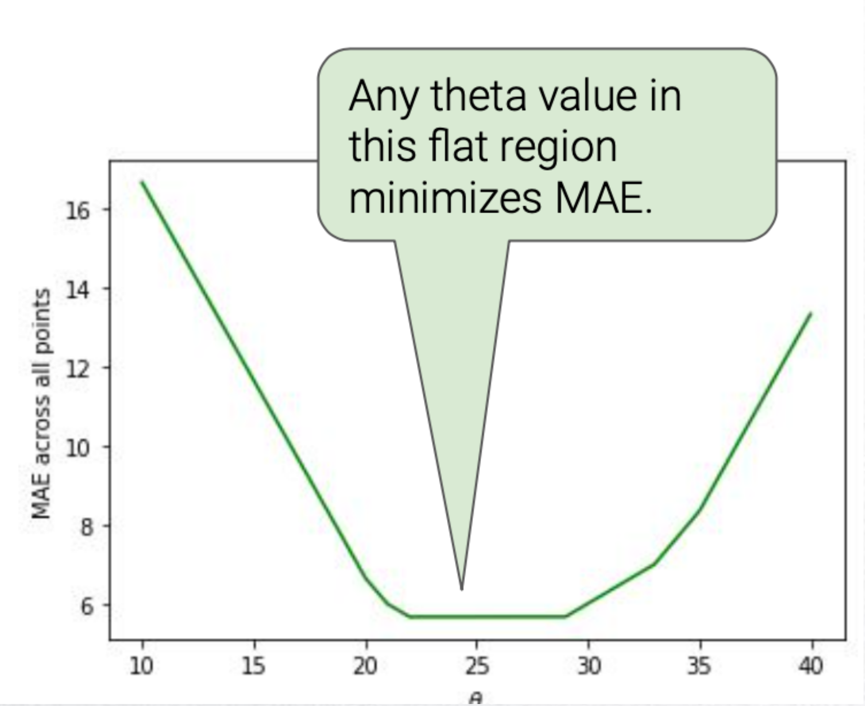
平均绝对离差
Mean Absolute Deviation

- 几个结论：
  - 给定常数模型和平均绝对误差，最优参数估计是观测数据**中位数**
  - 给中位数作为估计，平均绝对误差达到最小
  - 此时的参数估计**不容易受到离群点（Outlier）的影响**

回归：一元线性回归（解析解与梯度下降算法）

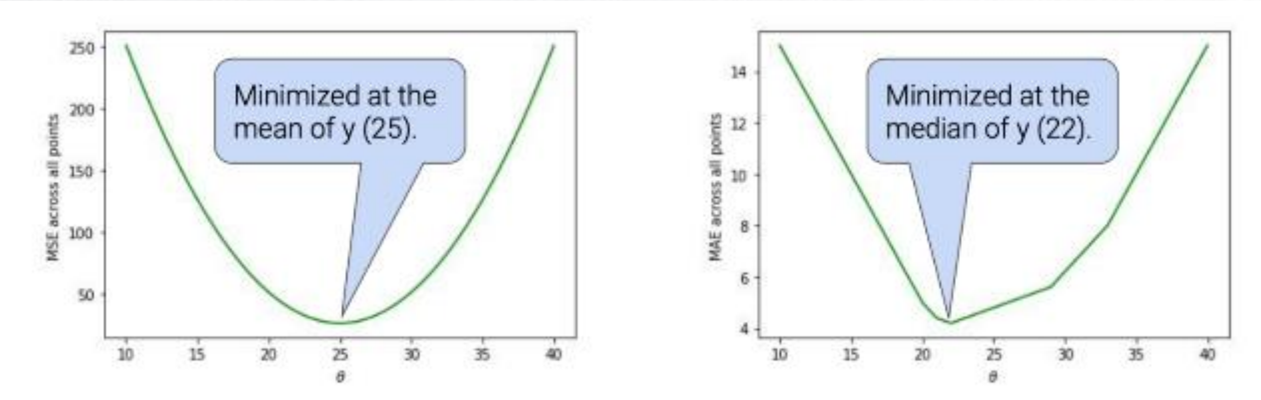
- 常数模型与平均绝对误差
- 思考：
  - 如果观测数据变为[20, 21, 22, 29, 33, 35]
  - 此时的估计值$\hat{\theta}$应为多少?
- 答案：$\hat{\theta}$不唯一，区间[22，29]内任意的取值均可。你能证明吗?

[20, 21, 22, 29, 33, 35]



回归：一元线性回归（解析解与梯度下降算法）

- 常数模型：对比MSE和MAE
- 考虑简单数据集[20,21,22,29,33] 和常数模型



- MSE的平均损失曲线光滑；MAE的平均损失曲线不光滑
  - 选择MSE，估计值$\hat{\theta} = \bar{y}$;
  - 选择MAE，估计值$\hat{\theta} = \textbf{median}(y)$ **（可能不唯一）**

回归：一元线性回归（解析解与梯度下降算法）

- Recap: 几个重要概念：下面几个概念后面会反复出现！
  - 观测值y_i与预测值$\hat{y}_i$
  - 参数θ与拟合参数（或参数估计值）$\hat{\theta}$
  - 估计（Estimation）与预测（Prediction）
  - 损失函数与平均损失（经验风险）
  - 均方误差MSE与平均绝对误差MAE

  - 参数求解 $\Rightarrow$ 平均损失（经验风险）最小化！

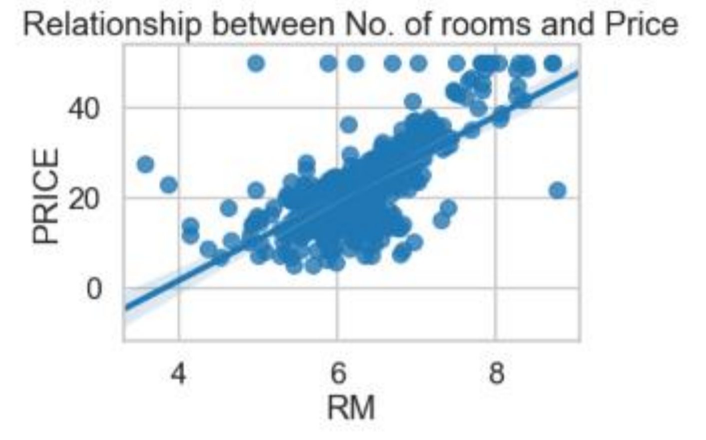
大多数监督学习算法（包括深度学习）的核心思想！

回归：一元线性回归（解析解与梯度下降算法）

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归
  - Simple Linear Regression(SLR) or Linear Regression with One Variable
- 让我们把模型变得更复杂一些
  - 考虑输入变量x

$$\hat{y}_i = \theta_0 + \theta_1 x_i$$



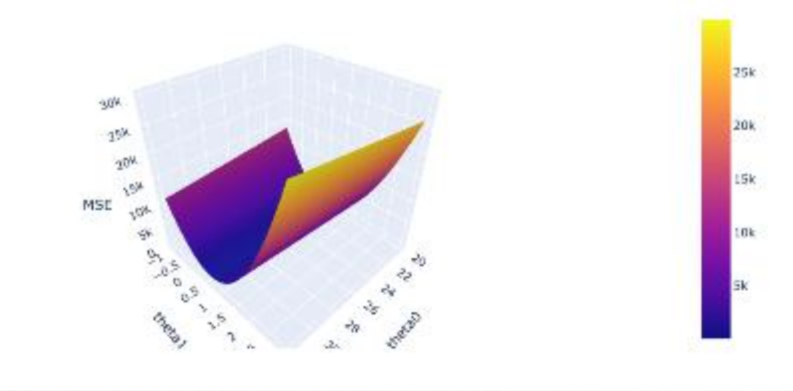
- 为了表示方便，我们将上式写成

$$\hat{y}_i = a x_i + b$$

- 该模型称为简单线性回归模型，简称SLR模型。
  - 例如：右图建立房间数量与房屋价格之间的SLR模型。

回归：一元线性回归（解析解与梯度下降算法）

- SLR模型与MSE目标函数
- 给定SLR模型，均方误差MSE可以写为
  - $R(a,b) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (\boldsymbol{ax_i} + \boldsymbol{b})\right)^2$
- 优化任务：如何计算最优的参数组合
  - $(\hat{a},\hat{b}) = \arg\min_{(a,b)} \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (ax_i + b)\right)^2$
- 数学工具：
  - 计算变量(a,b)的一阶偏导
  - **令一阶偏导为0**，从而求解$(\hat{a},\hat{b})$



1，这是损失函数的可视化效果
2，极值点位置，即导数为0的位置

回归：一元线性回归（解析解与梯度下降算法）

- 求解过程（1）
- 计算目标函数$R(a,b)=\frac{1}{n}\sum_{i=1}^{n}\left(y_i-(\boldsymbol{ax_i}+\boldsymbol{b})\right)^2$对变量$b$的一阶偏导

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

- 令一阶偏导等于0，得到

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

$$\Rightarrow\sum_{i=1}^{n}y_i-a\sum_{i=1}^{n}x_i-b\cdot n=0$$

$\Rightarrow \hat{b}=\bar{y}-a\bar{x}$，其中$\bar{y}$和$\bar{x}$分别$y$和$x$的均值

回归：一元线性回归（解析解与梯度下降算法）

- 求解过程（2）
- 将估计值$\hat{b} = \bar{y} - a\bar{x}$ 代入目标函数$R(a,b)\ = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (\boldsymbol{ax_i + b})\right)^2$，求得

$$R(a,b) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (ax_i + \bar{y} - a\bar{x})\right)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y} - a(x_i - \bar{x}))^2$$

- 计算目标函数$R(a,b)$对变量a的一阶偏导

$$\frac{\partial}{\partial a}R(a,b) = \frac{-2}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y} - a(x_i - \bar{x}))$$

- 令一阶偏导等于0，得到

$$\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) - a\sum_{i=1}^{n}(x_i - \bar{x})^2 = 0$$

$$\Rightarrow \hat{a} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归的解析解
  - $\hat{a} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$
  - $\hat{b} = \bar{y} - a\bar{x}$

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归的解析解：改写（可选内容）
  - 我们使用统计量对上述公式进行替换
  - $\sigma_x^2 = \sum_{i=1}^{n}(x_i - \bar{x})^2$, $\sigma_y^2 = \sum_{i=1}^{n}(y_i - \bar{y})^2$
  - $\sigma_x = \sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}$, $\sigma_y = \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}$
  - $r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sigma_x\sigma_y}$
    - r为统计学经常使用的皮尔森相关系数
  - $r \cdot \sigma_x\sigma_y = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$

$$\hat{a} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{b} = \bar{y} - a\bar{x}$$

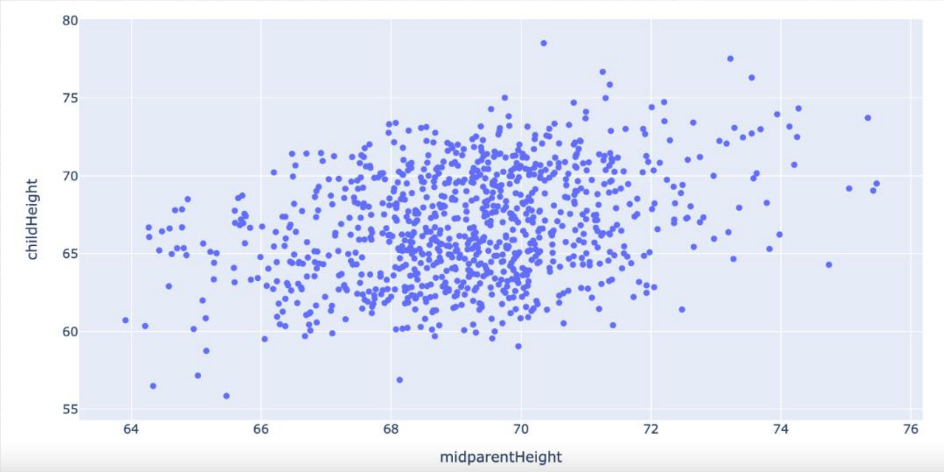
因此，我们得到参数估计

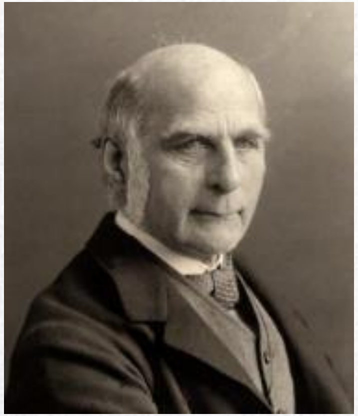
$$\hat{a} = \frac{r\cdot\sigma_x\sigma_y}{\sigma_x\sigma_x} = r\frac{\sigma_y}{\sigma_x} \ ; \ \hat{b} = \bar{y} - \hat{a}\bar{x}$$

回归：一元线性回归（解析解与梯度下降算法）

回归：一元线性回归（解析解与梯度下降算法）

- 变量之间的相关性
- 度量变量之间的相关性（Correlation）是统计中的典型问题
- 举例：你的身高与你父母的身高之间存在相关性吗？



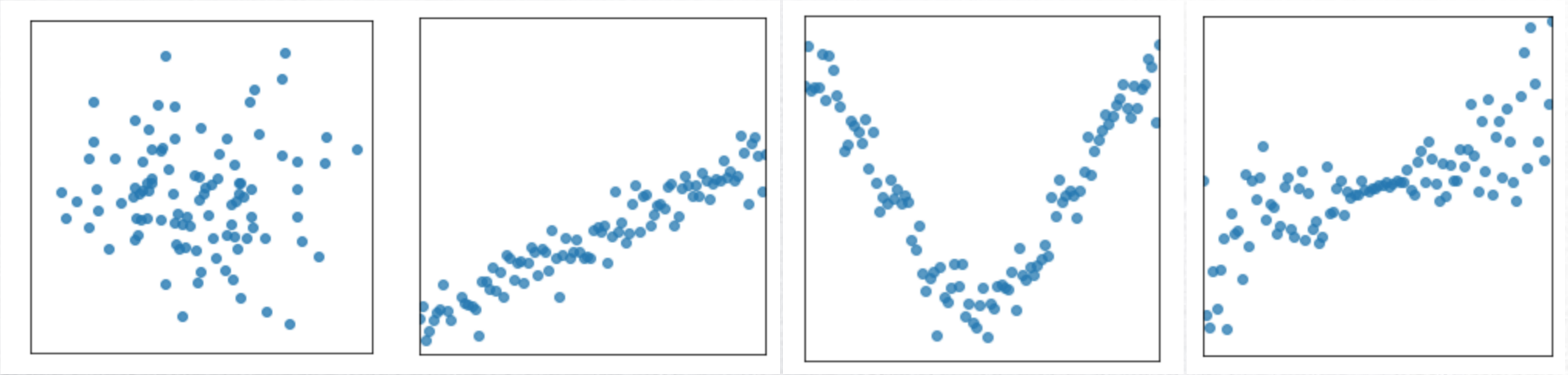


弗朗西斯·高尔顿（Francis Galton）

1855年论文：《遗传的身高向平均数方向的回归》

回归：一元线性回归（解析解与梯度下降算法）

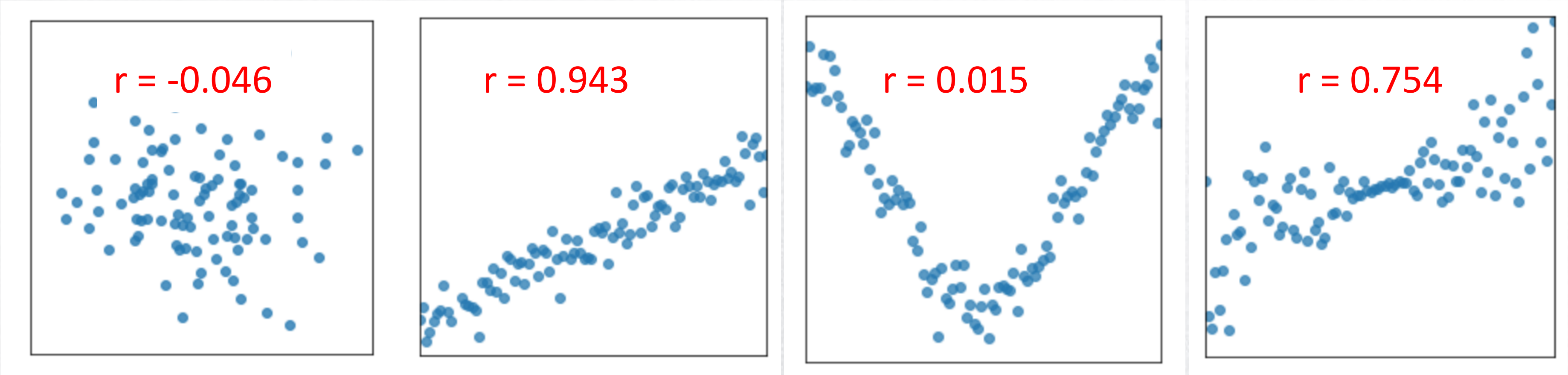
- 思考并回答下面变量之间的相关性



完全随机　　强线性相关　　非线性相关　　线性相关

回归：一元线性回归（解析解与梯度下降算法）

- 皮尔森相关系数
- $r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y}$



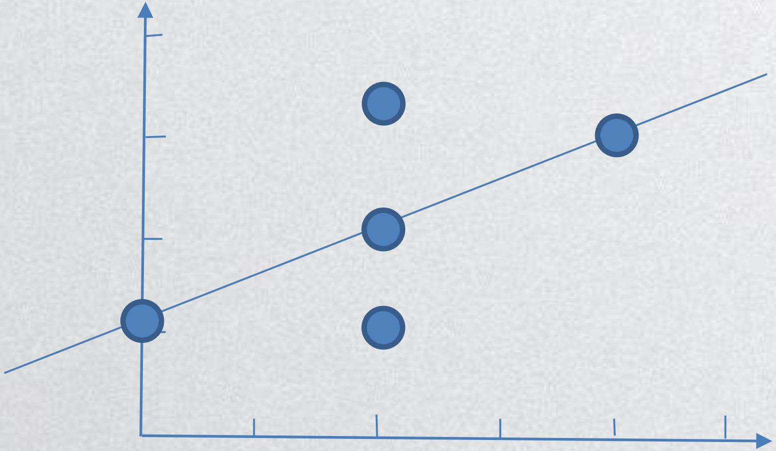
背景：卡尔·皮尔森是高尔顿的学生

回归：一元线性回归（解析解与梯度下降算法）

回归：一元线性回归（解析解与梯度下降算法）

- 实例：
- 假设有如下数据，请对其进行一元线性回归

问题



x	y
0	1
2	1
2	2
2	3
4	3
…	…

很显然$y = \frac{1}{2}x+1$

回归：一元线性回归（解析解与梯度下降算法）

- 针对如下数据，进行计算和验证

x	y
0	1
2	1
2	2
2	3
4	3
…	…

$\bar{x}$=10/5=2

$\bar{y}$=10/5=2

$$\hat{a} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \frac{(-2)(-1) + (0)(-1) + (0)(0) + (0)(1) + (2)(1)}{(-2)^2 + 0^2 + 0^2 + 0^2 + 2^2}$$

$$= \frac{2+2}{(4+4)} = \frac{4}{8} = 0.50$$

$$\hat{b} = \bar{y} - a\bar{x} = 2 - 0.50 * 2 = 1$$

$$\boldsymbol{y = ax + b = 0.5x + 1}$$

$$\hat{a} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{b} = \bar{y} - a\bar{x}$$

π

回归：一元线性回归（解析解与梯度下降算法）

回归：一元线性回归（解析解与梯度下降算法）

- 课堂练习（5-10分钟）
- 给定一组训练数据
  - (2, 4)
  - (5, 1)
  - (8, 9)

$$\hat{a}=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

$$\hat{b}=\bar{y}-a\bar{x}$$

- 请计算一个简单线性回归模型$\hat{y}=ax+b$的最优参数
  - $\hat{a}=?$
  - $\hat{b}=?$
  - 注：可以使用手机计算器、python辅助计算

回归：一元线性回归（解析解与梯度下降算法）

- 课堂练习
- 给定一组训练数据
  - (2, 4)
  - (5, 1)
  - (8, 9)

$$\hat{a} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{b} = \bar{y} - a\bar{x}$$

答案

$\bar{x}$=15/3=5

$\bar{y}$=14/3=4.667

$$\hat{a} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \frac{(-3)(-0.667) + (0)(-3.667) + (3)(4.333)}{(-3)^2 + 0^2 + 3^2}$$

$$= \frac{2.001+12.999}{(9+9)} = \frac{15}{18} = 0.8333$$

$$\hat{b} = \bar{y} - a\bar{x} = 4.667 - 0.8333 * 5 = 0.5005$$

$$\boldsymbol{y = ax + b = 0.8333x + 0.5005}$$

请参考本PPT附带代码

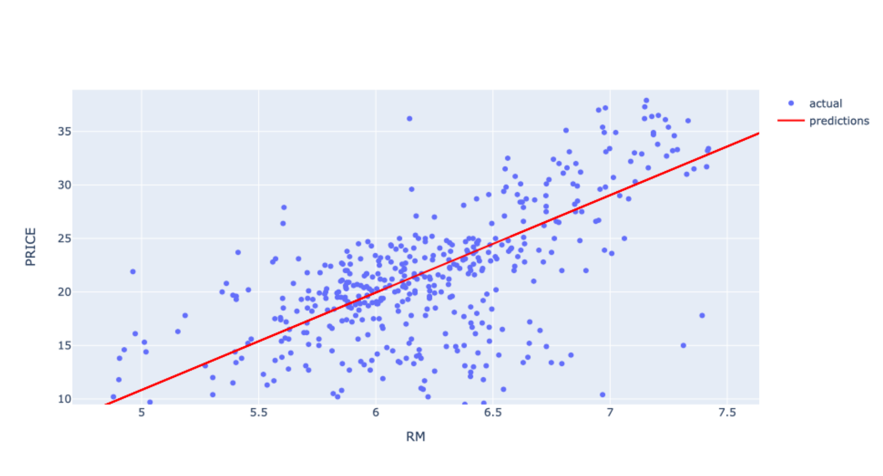
以及https://realpython.com/linear-regression-in-python/

名称	类型	大小	修改日期
01SLR_synthetic.py	Python File	1 KB	2021/10/14 15:50

回归：一元线性回归（解析解与梯度下降算法）

- Boston house price数据集的一元线性回归
- 根据右侧公式，可以计算

$$\text{PRICE} = -34.67 + 9.1 \ * \text{RM}$$



$$\hat{a} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{b} = \bar{y} - a\bar{x}$$

给定一个输入变量，如RM，线性回归计算的结果是二维中的：一条直线

名称	类型	大小	修改日期
04SLR_MLR_boston_house_price.ipynb	IPYNB 文件	5,313 KB	2021/10/16 14:15

回归：一元线性回归（解析解与梯度下降算法）

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
- 梯度下降法的基本原理
  - 假设有一个损失函数L，为参数θ_0和θ_1的函数
  - 1. 我们计算L对θ_0和θ_1的偏导数
    - 偏导数表示，θ_0和θ_1的微小变化导致L发生如何的变化
  - 2. 我们的目标是最小化损失函数
    - 于是我们按照$\theta = \theta - \eta\frac{\partial L}{\partial \theta}$的方式进行参数的修改，$\theta$为$\theta_0$或者$\theta_1$
    - 即可把目标函数修改小一点点
  - 3. 经过一系列迭代，即可把目标函数最小化(符合一定精度要求)

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
  - 理解梯度下降法——举个例子
  - 假设训练数据如下，为了简单起见，这里只有一个样本

训练数据	x	y
sample1	1	2

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
  - 理解梯度下降法——举个例子
  - 假设训练数据如下，为了简单起见，这里只有一个样本

训练数据	x	y
sample1	1	2

  - 用没有节距的一元线性回归模型y=wx进行建模，w为系数

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
  - 理解梯度下降法——举个例子
  - 假设训练数据如下，为了简单起见，这里只有一个样本

训练数据	x	y
sample1	1	2

  - 训练数据体现了数据的2倍数关系，即真实的数据体现的关系是y=f(x)=2x
    - 我们把w设置为某个初始值，比如w=3
    - 现在来看看，梯度下降法如何把3修正到2
    - 损失函数为Loss = $(f(x)-y)^2 = (wx-y)^2$
    - Loss函数针对w求偏导数$\frac{\partial}{\partial w}Loss = 2(wx-y)x$
    - 那么权重w的修正公式为w=w-2η(wx-y)x，η为学习率

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
  - 理解梯度下降法——举个例子
  - 列出了前5次迭代过程中，w的值的变化情况

训练数据	x	y
sample1	1	2

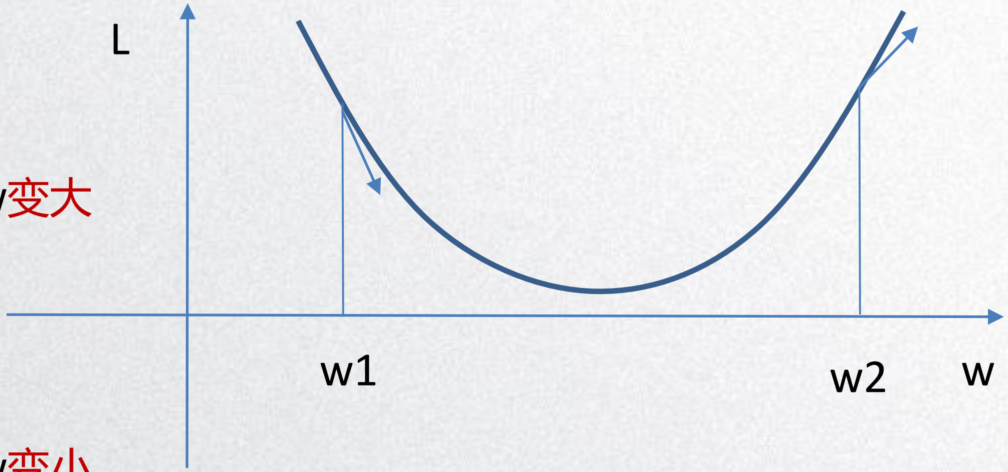
迭代	x	w	y	2(wx-y)x	η	w=w-2η (wx-y)x
1	1	3	2	2	0.1	2.8
2	1	2.8	2	1.6	0.1	2.64
3	1	2.64	2	1.28	0.1	2.512
4	1	2.512	2	1.024	0.1	2.4096
5	1	2.4096	2	0.8192	0.1	2.3277
…						

- 可以看到，按照上述公式，w逐渐逼近2.0，即x和y表达出来的正确的数量关系
- 值得注意的是，如果初始化的w小于2，那么它将从另外一个方向逼近2.0

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
- 梯度下降法直观解释
  - 目标损失函数对于w参数的图像如下
  - 在w1这个点上
    - 梯度$\frac{\partial}{\partial w}Loss$为一个负值
    - 按照w=w-η $\frac{\partial}{\partial w}Loss$调整，w变大
  - 在w2这个点上
    - 梯度$\frac{\partial}{\partial w}Loss$为一个正值
    - 按照w=w-η $\frac{\partial}{\partial w}Loss$调整，w变小

梯度即切线方向的变化量



回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
  - 理解梯度下降法——举个例子
  - 从该例子，扩展思路

- （1）在这里，我们使用线性函数对梯度下降法进行说明，实际应用中可以用非线性函数表达x和y之间的非线性关系
- （2）在这里，只有一个样本，实际应用中一般有很多的样本
- （3）在这里，x只有一个分量，实际应用中一般x是一个多维向量
- 但是这些不影响我们对梯度下降法本质的理解

回归：一元线性回归（解析解与梯度下降算法）

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
  - 为了和后续介绍的多元线性回归统一，我们把一元线性回归写成如下形式
  - $\hat{y}_i = h_\theta(x_\mathrm{i}) = \theta_0 +$，改写为$\hat{y}^{(i)} = h_\theta\left(\mathrm{x}^{(i)}\right) = \theta_0 + \theta_1 \mathrm{x}^{(i)}$
  - 这里假设样本数量为m

x	y	$\hat{y}$
x_1	y_1	$\hat{y}_1$
x_2	y_2	$\hat{y}_2$
…	…	…
x_m	y_m	$\hat{y}_m$

样本用上标i
分量用下标j

我们在这里将使用梯度下降算法，求解一元线性回归的参数，参数有两个分别是系数θ_1和节距θ_0

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
- $\hat{y}^{(i)} = h_\theta\left(\mathrm{x}^{(i)}\right) = \theta_0 + \theta_1 \mathrm{x}^{(i)}$
- 目标函数如下

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

样本用上标i
分量用下标j

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
  - $\hat{y}^{(i)} = h_\theta\left(\mathrm{x}^{(i)}\right) = \theta_0 + \theta_1 \mathrm{x}^{(i)}$
  - 计算目标函数对θ_0和θ_1的偏导数

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

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

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

样本用上标i
分量用下标j

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
- $\hat{y}^{(i)} = h_\theta\left(\mathrm{x}^{(i)}\right) = \theta_0 + \theta_1 \mathrm{x}^{(i)}$
- 计算目标函数对θ_0和θ_1的偏导数，参数的修改

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

$$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta_0, \theta_1)$$

样本用上标i
分量用下标j

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
- $\hat{y}^{(i)} = h_\theta\left(\mathrm{x}^{(i)}\right) = \theta_0 + \theta_1 \mathrm{x}^{(i)}$
- 梯度下降算法

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

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

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

$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

repeat until convergence {

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

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

}

$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

update θ_0 and θ_1 simultaneously

回归：一元线性回归（解析解与梯度下降算法）

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
  - $\hat{y}^{(i)} = h_\theta\left(\mathrm{x}^{(i)}\right) = \theta_0 + \theta_1 \mathrm{x}^{(i)}$
  - 参考代码（1）

名称	类型	大小	修改日期
salary_data.csv	Microsoft Excel 逗号分隔...	1 KB	2020/5/12 0:20
01SLR_synthetic.py	Python File	1 KB	2021/10/14 15:50
02SLR_SGD.py	Python File	3 KB	2021/10/14 16:28

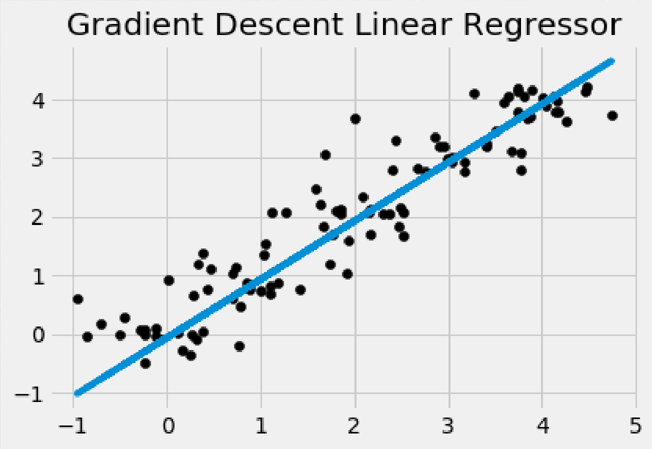
1.用scikit-learn的LinearRegression验证本PPT两个实例的解析解

2.用scikit-learn的LinearRegression和SGDRegressor对salary_data.csv进行回归，并且比较

https://medium.com/@ktv0303/simple-linear-regression-or-linear-regression-with-one-variable-2c37d5ba4fe
https://github.com/karthikeyanthanigai/Simple-linear-regression-ols-vs-sgd-

回归：一元线性回归（解析解与梯度下降算法）

- 一元线性回归：梯度下降法求解
  - $\hat{y}^{(i)} = h_\theta\left(\mathrm{x}^{(i)}\right) = \ \theta_0 + \theta_1 \mathrm{x}^{(i)}$
  - 参考代码（2）：自行实现梯度下降代码

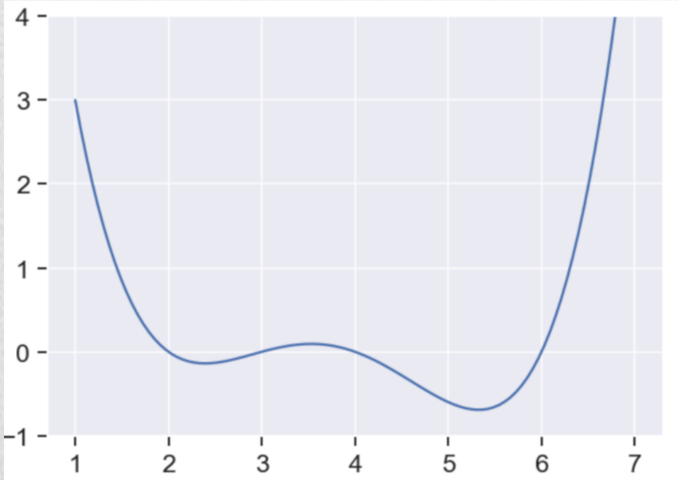


https://www.kaggle.com/residentmario/gradient-descent-with-linear-regression

回归：一元线性回归（解析解与梯度下降算法）

回归：一元线性回归（解析解与梯度下降算法）

- 梯度下降法的讨论
- 求解一维函数的最小值：考虑下面的一维函数
  - $f(x) = (x^4 - 15x^3 + 80x^2 - 180x + 144)/10$



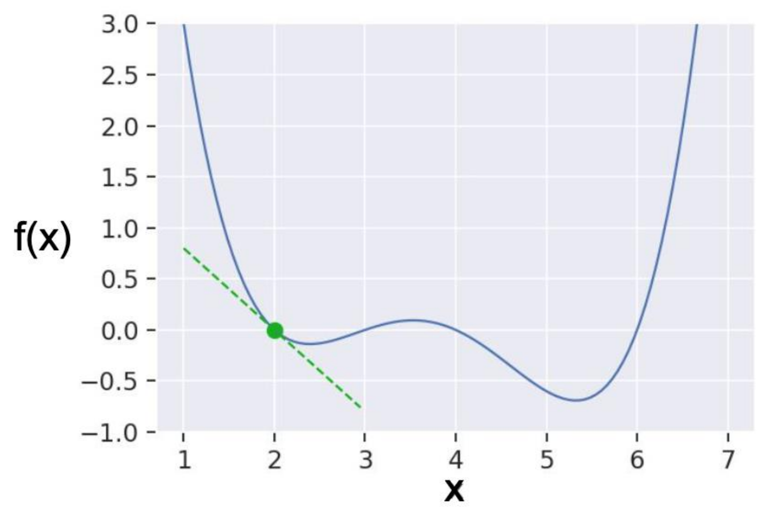
如何求解：

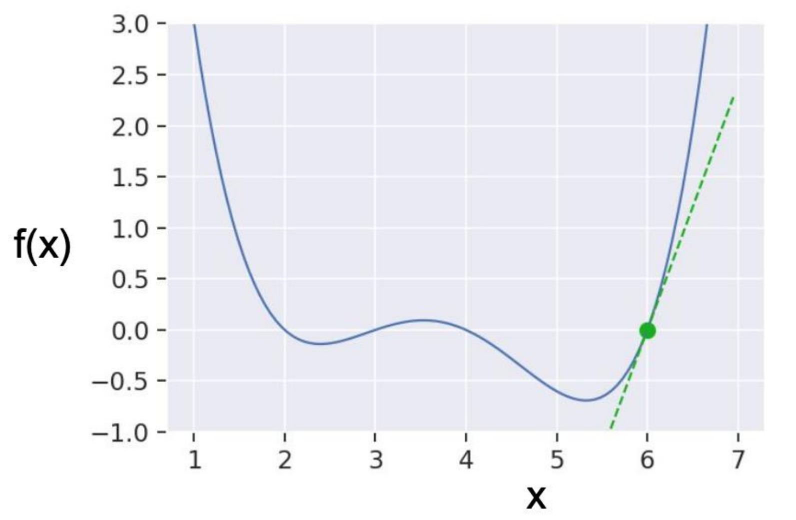
$$\hat{x} = \arg\min_{x} f(x)$$

回归：一元线性回归（解析解与梯度下降算法）

- 一维梯度下降方法
  - 在最小值点左侧，导数是负数 → x应该增大
  - 在最小值点右侧，导数是正数 → x应该减小

导数告诉我们：
- 往哪个方向走
- 应该走多远





回归：一元线性回归（解析解与梯度下降算法）

- 一维梯度下降算法：极简实现版
  - 输入：
    - gradient: 梯度（导数）函数
    - init_guess: 初始猜测
    - learn_rate: 学习率
    - n_iter: 迭代次数
  - 输出：
    - 求解的极小值点$\hat{x}$

$$x^{(t+1)} = x^{(t)} - \alpha \frac{d}{dx} f(x)$$

```
def gradient_descent(gradient, init_guess, learn_rate, n_iter):
    guess = init_guess
    for _ in range(n_iter):
        guess = guess - learn_rate * gradient(guess)
    return guess
```

回归：一元线性回归（解析解与梯度下降算法）

- 定义梯度函数gradient

导数

```python
def derivative_arbitrary(x):
    return (4*x**3 - 45*x**2 + 160*x - 180)/10
```

$$f(x) = (x^4 - 15x^3 + 80x^2 - 180x + 144)/10$$

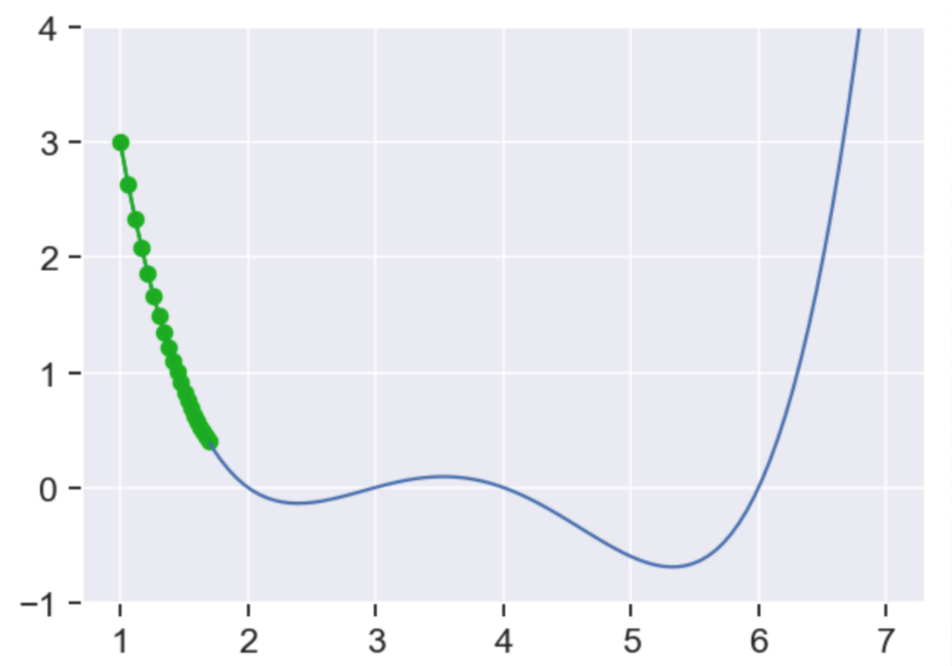
- 选择超参数
  - init_guess：一般是随机选择；作为示例，可以选择 **1**
  - n_iter: 根据实际情况，选择迭代次数；作为示例，请选择 **20**
  - learn_rate: 应该如何选择？做一些尝试

```python
def gradient_descent(gradient, init_guess, learn_rate, n_iter):
    guess = init_guess
    for _ in range(n_iter):
        guess = guess - learn_rate * gradient(guess)
    return guess
```

回归：一元线性回归（解析解与梯度下降算法）

- 学习率的选择
- 选择一个很小的学习率
  - $\alpha = 0.01$

```
0-th iteration: guess = 1, gradient=-6.1
1-th iteration: guess = 1.06, gradient=-5.61
2-th iteration: guess = 1.12, gradient=-5.18
3-th iteration: guess = 1.17, gradient=-4.81
4-th iteration: guess = 1.22, gradient=-4.47
5-th iteration: guess = 1.26, gradient=-4.17
6-th iteration: guess = 1.3, gradient=-3.9
7-th iteration: guess = 1.34, gradient=-3.66
8-th iteration: guess = 1.38, gradient=-3.44
9-th iteration: guess = 1.41, gradient=-3.24
10-th iteration: guess = 1.45, gradient=-3.06
11-th iteration: guess = 1.48, gradient=-2.9
12-th iteration: guess = 1.51, gradient=-2.75
13-th iteration: guess = 1.53, gradient=-2.61
14-th iteration: guess = 1.56, gradient=-2.48
15-th iteration: guess = 1.58, gradient=-2.36
16-th iteration: guess = 1.61, gradient=-2.25
17-th iteration: guess = 1.63, gradient=-2.14
18-th iteration: guess = 1.65, gradient=-2.05
19-th iteration: guess = 1.67, gradient=-1.96
```



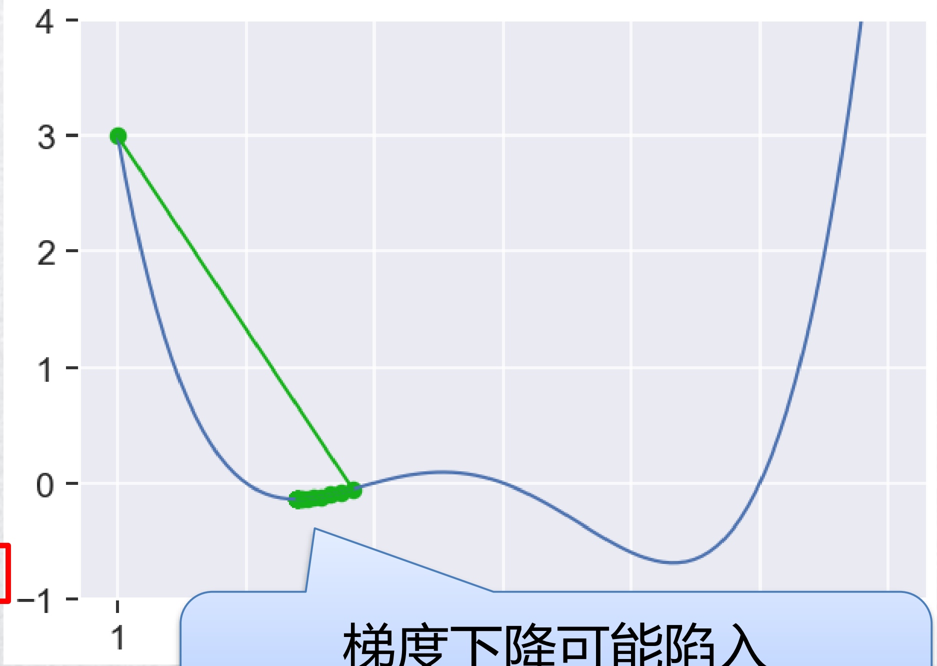
应该如何解决？

回归：一元线性回归（解析解与梯度下降算法）

- 学习率的选择
- 选择一个略小的学习率
  - $\alpha = 0.3$

```
0-th iteration: guess = 1, gradient=-6.1
1-th iteration: guess = 2.83, gradient=0.31
2-th iteration: guess = 2.74, gradient=0.28
3-th iteration: guess = 2.65, gradient=0.24
4-th iteration: guess = 2.58, gradient=0.2
5-th iteration: guess = 2.52, gradient=0.15
6-th iteration: guess = 2.48, gradient=0.1
7-th iteration: guess = 2.45, gradient=0.07
8-th iteration: guess = 2.43, gradient=0.04
9-th iteration: guess = 2.41, gradient=0.03
10-th iteration: guess = 2.41, gradient=0.02
11-th iteration: guess = 2.4, gradient=0.01
12-th iteration: guess = 2.4, gradient=0.01
13-th iteration: guess = 2.4, gradient=0.0
```

梯度已经减为0！

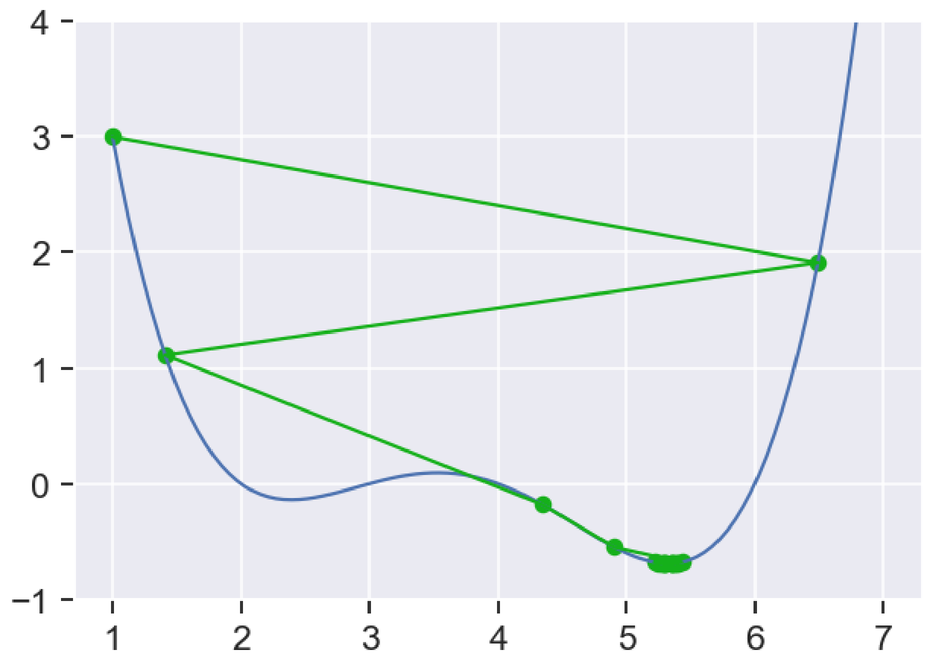


梯度下降可能陷入
局部最优（Local Minima）

回归：一元线性回归（解析解与梯度下降算法）

- 学习率的选择
- 选择一个大一些的学习率
  - $\alpha = 0.9$

```
0-th iteration: guess = 1, gradient=-6.1
1-th iteration: guess = 6.49, gradient=5.64
2-th iteration: guess = 1.41, gradient=-3.26
3-th iteration: guess = 4.34, gradient=-0.62
4-th iteration: guess = 4.91, gradient=-0.58
5-th iteration: guess = 5.43, gradient=0.24
6-th iteration: guess = 5.22, gradient=-0.21
7-th iteration: guess = 5.4, gradient=0.18
8-th iteration: guess = 5.25, gradient=-0.16
9-th iteration: guess = 5.39, gradient=0.14
10-th iteration: guess = 5.26, gradient=-0.12
11-th iteration: guess = 5.38, gradient=0.11
12-th iteration: guess = 5.28, gradient=-0.1
13-th iteration: guess = 5.37, gradient=0.09
14-th iteration: guess = 5.29, gradient=-0.08
15-th iteration: guess = 5.36, gradient=0.07
16-th iteration: guess = 5.3, gradient=-0.06
17-th iteration: guess = 5.35, gradient=0.05
18-th iteration: guess = 5.3, gradient=-0.05
```



更大的学习率可以使算法在更大的范围内进行试探

回归：一元线性回归（解析解与梯度下降算法）

- 学习率的选择
- 选择一个再大一些的学习率
  - $\alpha = 0.95$

```
0-th iteration: guess = 1, gradient=-6.1
1-th iteration: guess = 6.79, gradient=8.44
2-th iteration: guess = -1.22, gradient=-45.07
3-th iteration: guess = 41.59, gradient=21643.42
4-th iteration: guess = -20519.65, gradient=-3457865914535.88
5-th iteration: guess = 3284972598289.43, gradient=1.4179314815282369e+37
6-th iteration: guess = -1.347034907451825e+37, gradient=-9.776795748433592e+110

---------------------------------------------------------------------------
OverflowError                             Traceback (most recent call last)
<ipython-input-39-6bf613c13fd5> in <module>
      1 plot_arbitrary()
----> 2 guess = gradient_descent_with_plot(derivative_arbitrary, arbitrary, 1, 0.95, 20)
      3 print(guess)
```

请你解释出现上述情况的原因？

回归：一元线性回归（解析解与梯度下降算法）

- 学习率的选择
- 选择一个再大一些的学习率
  - $\alpha = 0.95$

```
0-th iteration: guess = 1, gradient=-6.1
1-th iteration: guess = 6.79, gradient=8.44
2-th iteration: guess = -1.22, gradient=-45.07
3-th iteration: guess = 41.59, gradient=21643.42
4-th iteration: guess = -20519.65, gradient=-3457865914535.88
5-th iteration: guess = 3284972598289.43, gradient=1.4179314815282369e+37
6-th iteration: guess = -1.347034907451825e+37, gradient=-9.776795748433592e+110

---------------------------------------------------------------------------
OverflowError                             Traceback (most recent call last)
<ipython-input-39-6bf613c13fd5> in <module>
      1 plot_arbitrary()
----> 2 guess = gradient_descent_with_plot(derivative_arbitrary, arbitrary, 1, 0.95, 20)
      3 print(guess)
```

请你解释出现上述情况的原因？

梯度爆炸，溢出

回归：一元线性回归（解析解与梯度下降算法）

- 学习率的选择
  - 过大的学习率导致学习的不稳定，甚至不能收敛
    - 如之前的“螺旋上升”现象
  - 过小的学习率导致训练步数多读
    - 导致过长的训练时间
  - 复杂的工程问题！
    - 你会设计什么机制?

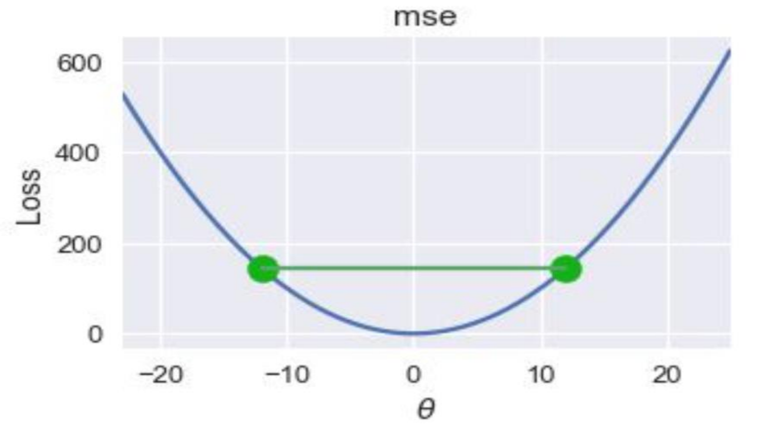
$$\boxed{x^{(t+1)} = x^{(t)} - \alpha\frac{d}{dx}f(x)}$$

```python
def gradient_descent(gradient, init_guess, learn_rate, n_iter):
    guess = init_guess
    for _ in range(n_iter):
        guess = guess - learn_rate * gradient(guess)
    return guess
```

回归：一元线性回归（解析解与梯度下降算法）

- 函数的凸性（Convexity）
- 定义
  - 函数$f(x)$为凸函数的充分必要条件是给定定义域中任意两个点x_1和x_2及某个常数$t \in [0,1]$，都有：

$$t \cdot f(x_1) + (1-t) \cdot f(x_2) \geq f(t \cdot x_1 + (1-t)x_2)$$

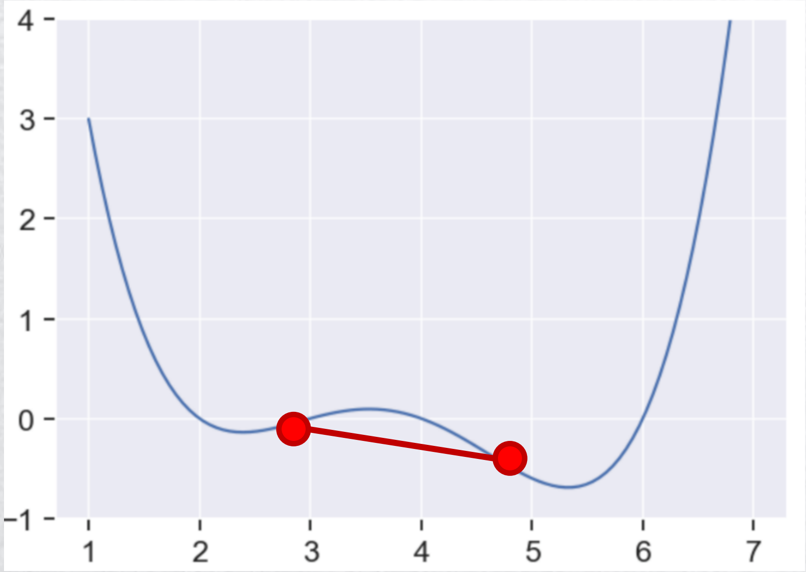


在函数$f(x)$为凸函数的情况下，梯度下降方法能够找到函数$f(x)$的最小值点

MSE目标函数是凸函数吗？
A. 是　B. 否

回归：一元线性回归（解析解与梯度下降算法）

- 函数的凸性（Convexity）
- 请给出此一维函数不符合凸性的例子
  - $f(x) = (x^4 - 15x^3 + 80x^2 - 180x + 144)/10$



回归：一元线性回归（解析解与梯度下降算法）

- 梯度下降法的讨论
- 参考代码

名称	类型	大小	修改日期
05SGD_simple_boston_house_price.ipynb	IPYNB 文件	5,392 KB	2021/10/16 15:37

请打开代码运行与分析

回归：一元线性回归（解析解与梯度下降算法）

回归：一元线性回归（解析解与梯度下降算法）

- 回顾

- 回归问题、模型、建立模型的三个步骤
- 常数模型（平均平方误差|平均绝对误差）
- 一元线性回归（解析解|梯度下降法求解）
- 梯度下降法的讨论