Linear Regression

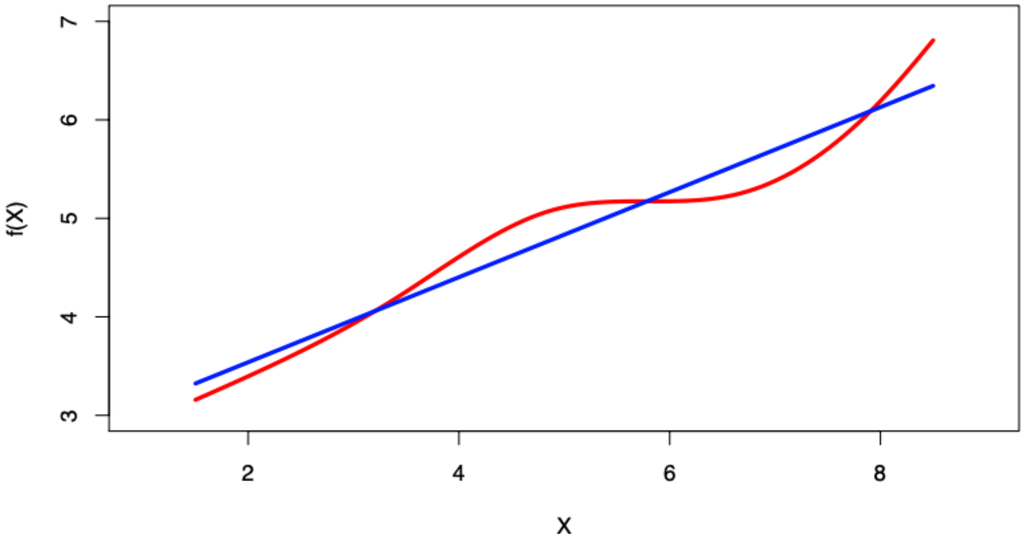
线性回归

Outline

- 一元线性回归
- 最小二乘估计 & 梯度下降算法
- 多元线性回归
- 方差分析
- 线性模型检验
- 模型诊断

线性回归

- 线性回归是最简单的有监督学习方法，其假设响应变量对特征的依赖关系是线性的。
- 相较现代机器学习方法，虽然看起来过于简化，但却非常有效并被广泛使用。
- 我们将会看到，许多方法可以看做线性回归的推广，线性回归中的很多概念仍适用于其他方法。
- 真实的回归函数往往不是线性的。



例：各平台广告投放与销量的关系

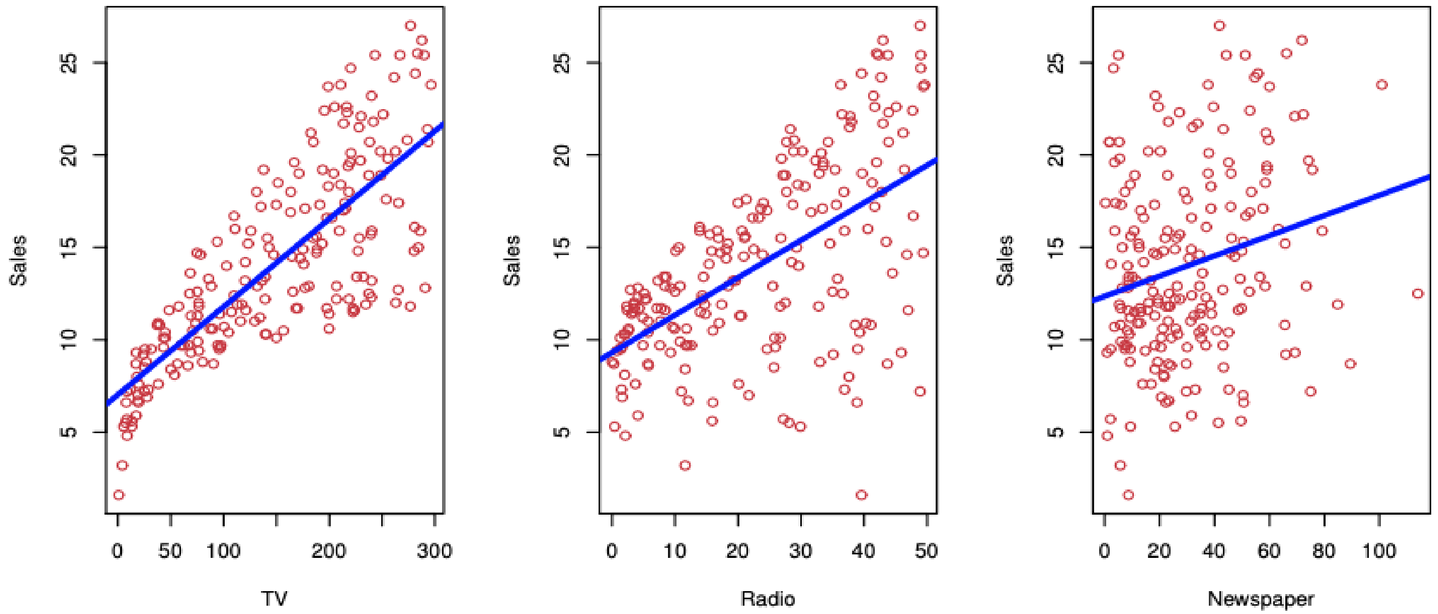


FIGURE 2.1. *The* `Advertising` *data set. The plot displays* `sales`*, in thousands of units, as a function of* `TV`*,* `radio`*, and* `newspaper` *budgets, in thousands of dollars, for* 200 *different markets. In each plot we show the simple least squares fit of* `sales` *to that variable, as described in Chapter 3. In other words, each blue line represents a simple model that can be used to predict* `sales` *using* `TV`*,* `radio`*, and* `newspaper`*, respectively.*

思考

- 广告预算与销量是否有关？
- 如果有关，这种关系有多强？
- 更具体的说，哪一种媒体的广告预算与销量有关？关系有多强？
- 这种关系是不是线性的？
- 我们可以多准确的预测未来的销量？

- 这些问题通过线性回归都可以得到解答

一元线性回归

基于披萨店的数据

- Armand比萨饼连锁店，大学校园附近
- 管理人员确信，季度销售收入与校园学生人数有密切联系
- 管理人员希望定量地理解校园学生人数与连锁店季度销售额之间的关系
- 考虑在两所大学附近新开设连锁店，两个校园的学生人数分别为10000与18000，希望预测新连锁店的季度销售额
- 简单线性回归：只有一个自变量

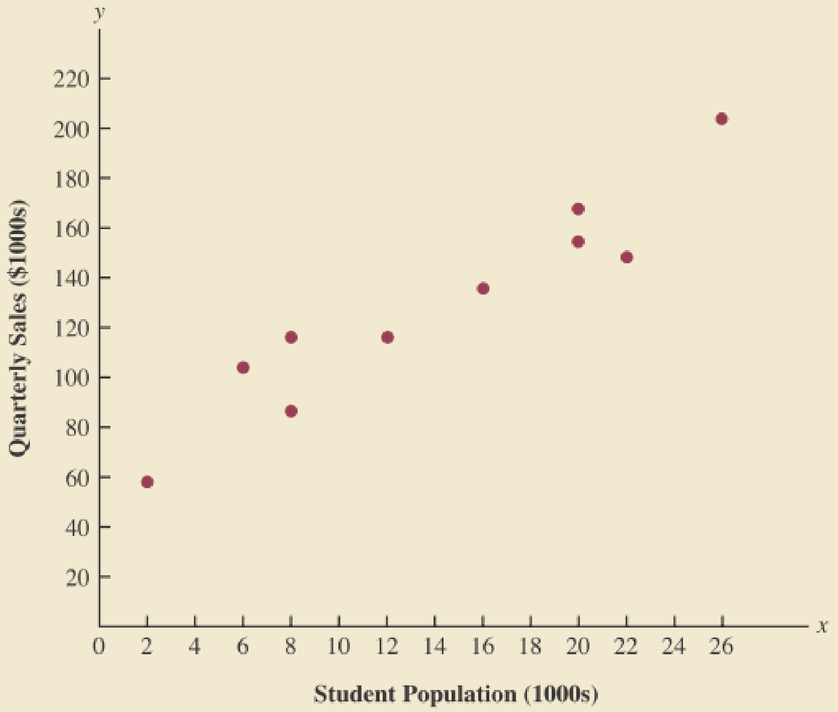
X(学生人数)	自变量 independent variable	协变量 covariate	解释变量 explanatory variable	regressor
Y(销售额)	dependent variable	响应变量 response variable, outcome	被解释变量 explained variable	regressand

基于披萨店的数据

TABLE 14.1 STUDENT POPULATION AND QUARTERLY SALES DATA FOR 10 ARMAND'S PIZZA PARLORS

Restaurant i	Student Population (1000s) x_i	Quarterly Sales ($1000s) y_i
1	2	58
2	6	105
3	8	88
4	8	118
5	12	117
6	16	137
7	20	157
8	20	169
9	22	149
10	26	202

FIGURE 14.3 SCATTER DIAGRAM OF STUDENT POPULATION AND QUARTERLY SALES FOR ARMAND'S PIZZA PARLORS



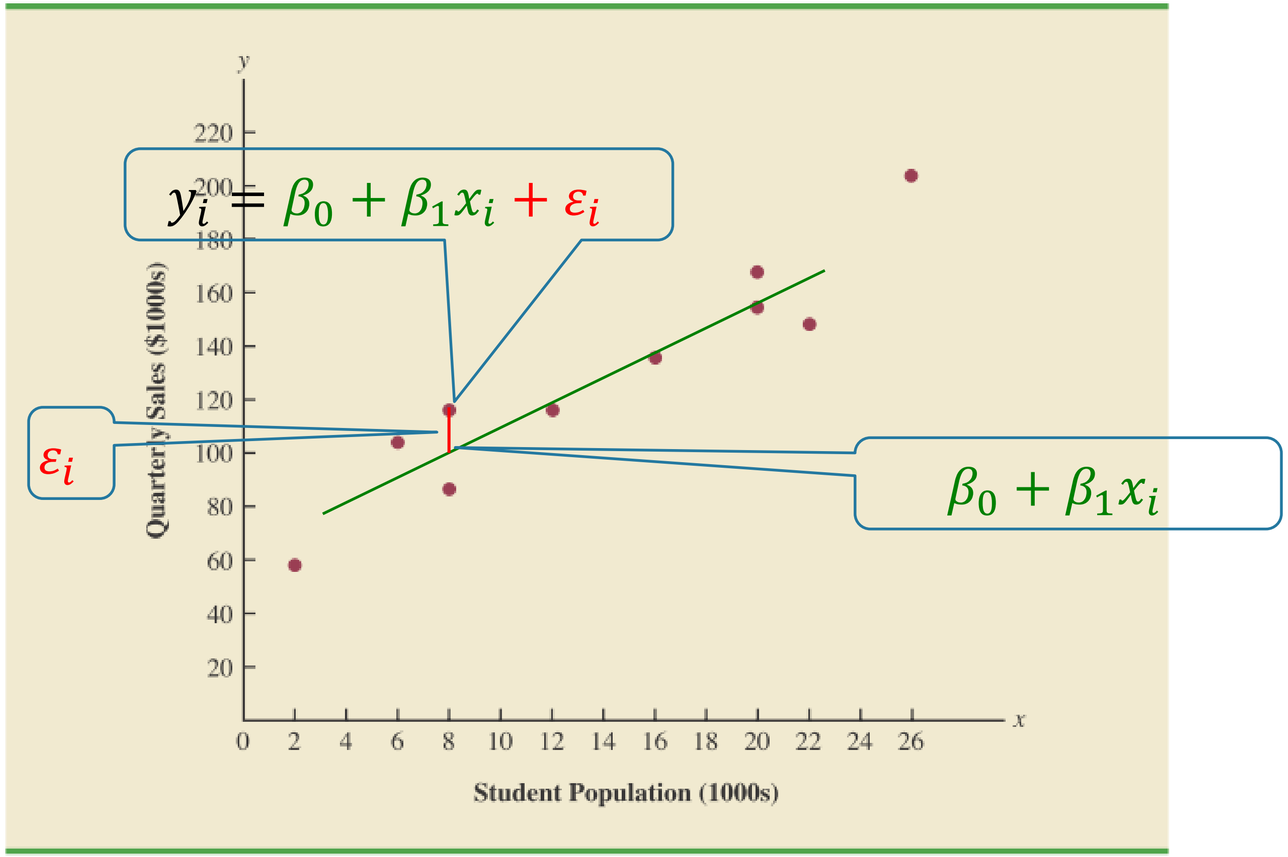
- 描述统计1：$r = 0.95$
- 描述统计2：散点图
- 基于描述统计：学生数量x和季度销售额y之间
  - 正相关；
  - 基本是线性关系；
- 用线性模型刻画这种关系

模型

- 假设模型：

$$Y = \beta_0 + \beta_1 X + \epsilon$$

其中β_0与β_1是两个未知参数，分别代表截距项和斜率； ϵ 为误差项。



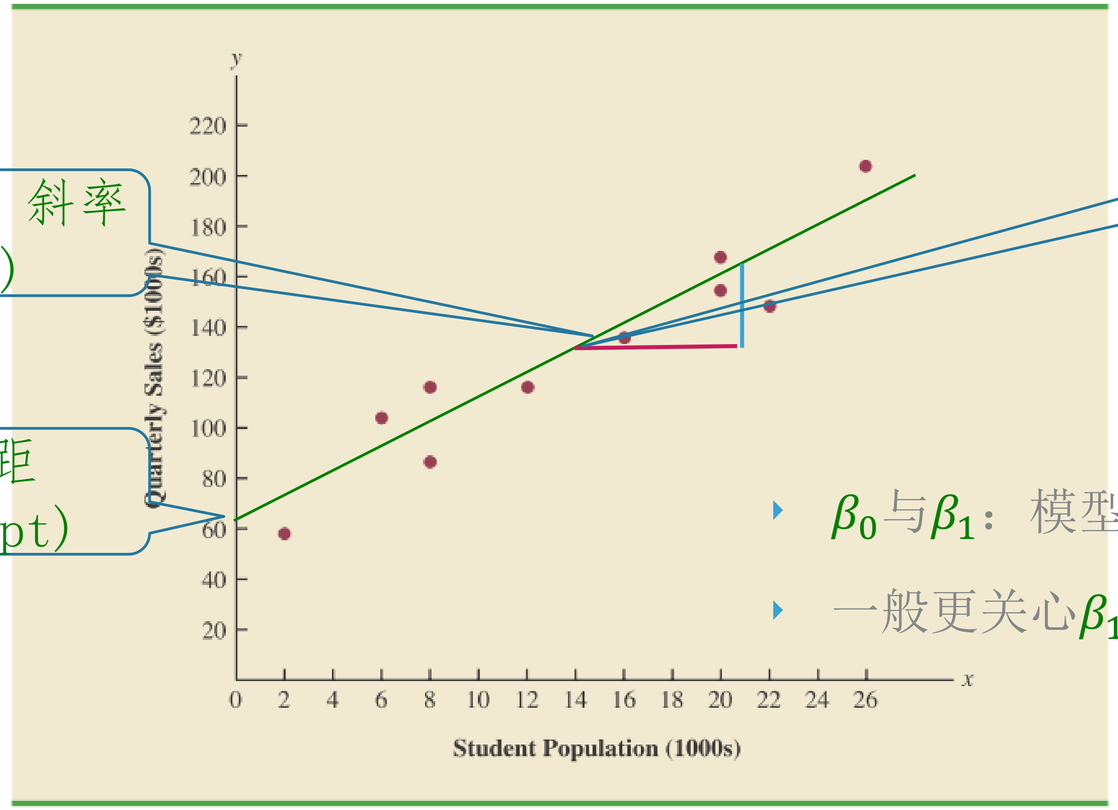
模型

- 假设模型：

$$Y = \beta_0 + \beta_1 X + \epsilon$$

其中β_0与β_1是两个未知参数，分别代表截距项和斜率； ϵ为误差项。

FIGURE 14.3 SCATTER DIAGRAM OF STUDENT POPULATION AND QUARTERLY SALES FOR ARMAND'S PIZZA PARLORS



$\beta_1 = \tan\theta$：斜率 (slope)

β_0：截距 (intercept)

θ

- β_0与β_1：模型的系数coefficients
- 一般更关心β_1，因为β_1蕴含了x与y之间的关系

一元线性回归

- 假设模型：

$$Y = \beta_0 + \beta_1 X + \epsilon$$

其中β_0与β_1是两个未知参数，分别代表截距项和斜率； ϵ 为误差项。

- 给定参数的估计$\hat{\beta}_0$与$\hat{\beta}_1$，我们可以基于x预测y：

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

- 一个自然的想法：使训练数据集的$\hat{y}$与y之间尽可能的接近。

一元线性回归

- 假设模型：

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$
$$E(Y_i|X_i) = \beta_1 X_i$$

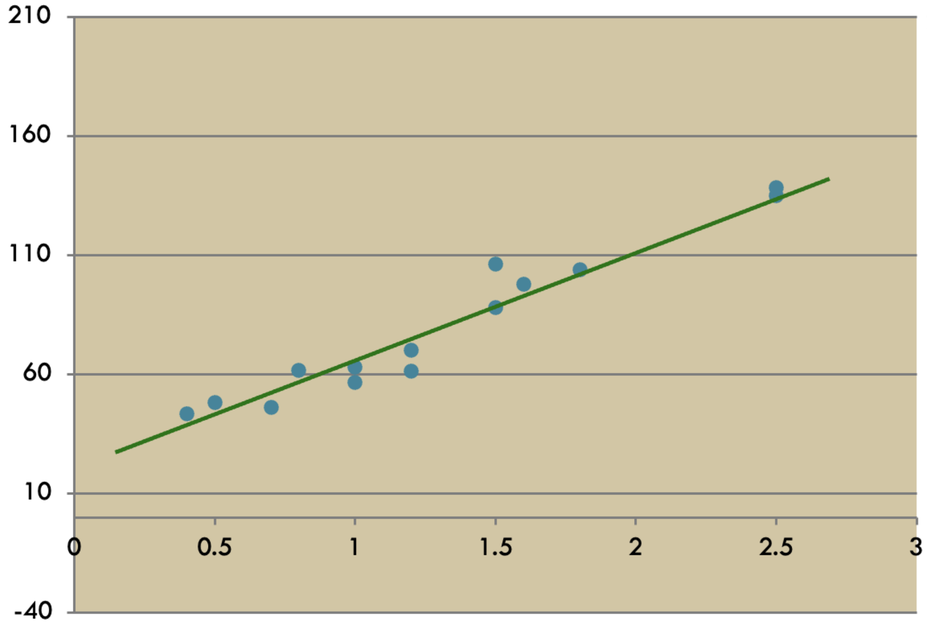
其中β_0与β_1是两个未知参数，分别代表截距项和斜率； ϵ 为误差项。

- 模型假设

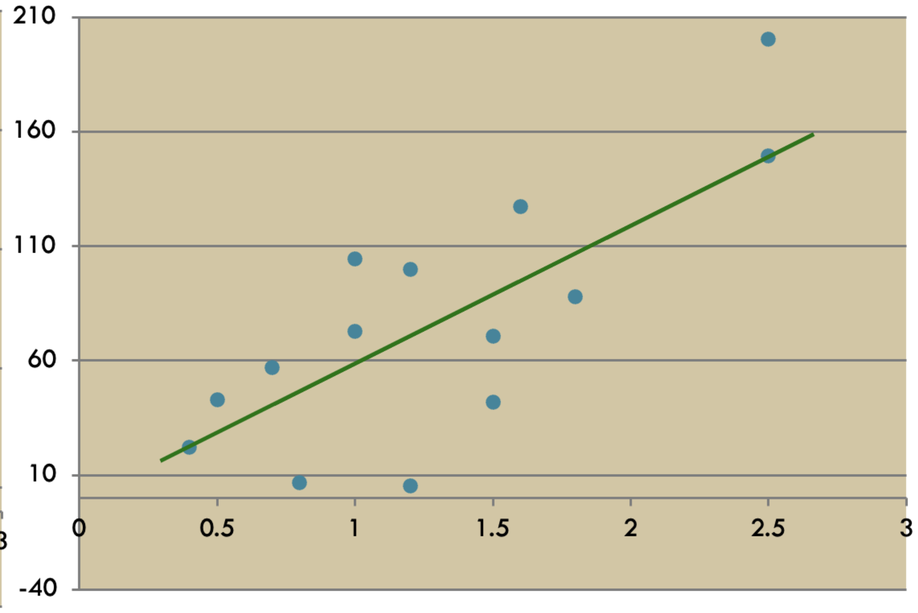
  - $\epsilon_i \sim i.i.d.\mathrm{N}(0, \sigma^2)$
  - ϵ_i之间相互独立
  - ϵ_i与X_i不相关
  - 均值为0
  - 方差相等

一元线性回归

- 误差方差σ^2大小的影响



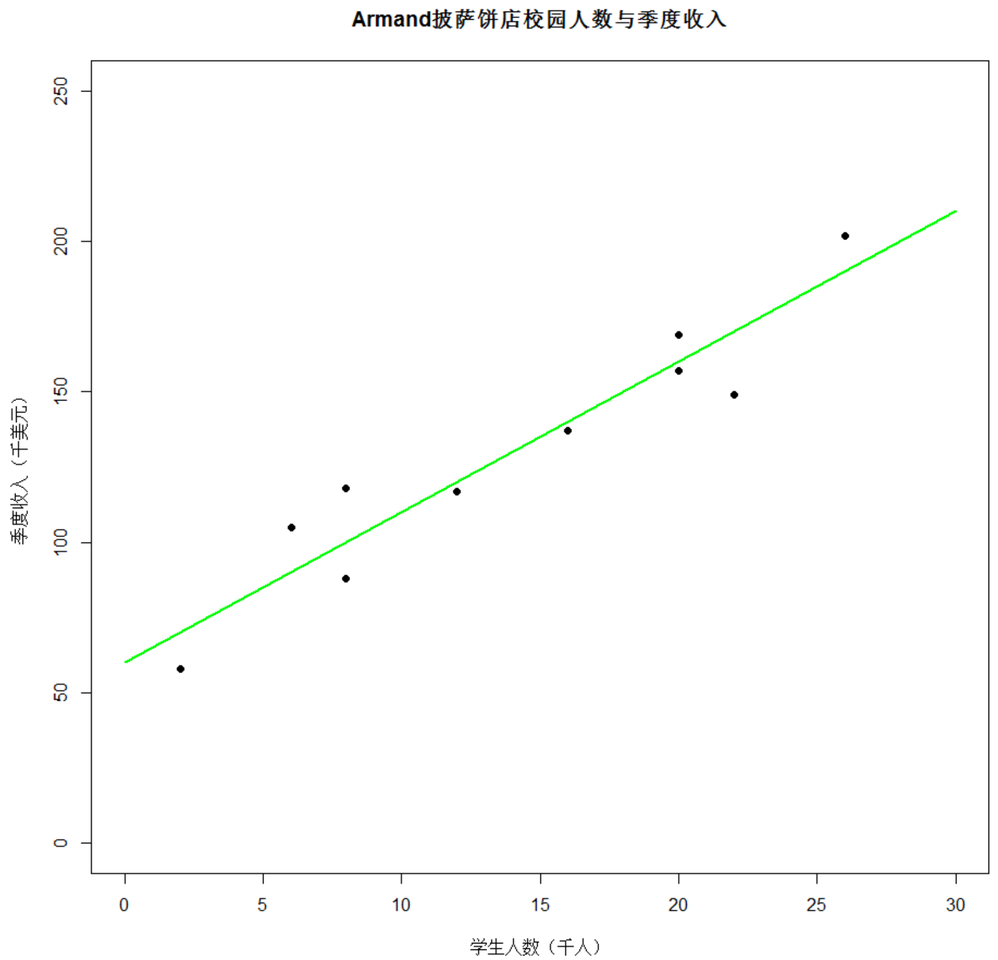
$\sigma^2 = 1$



$\sigma^2 = 5$

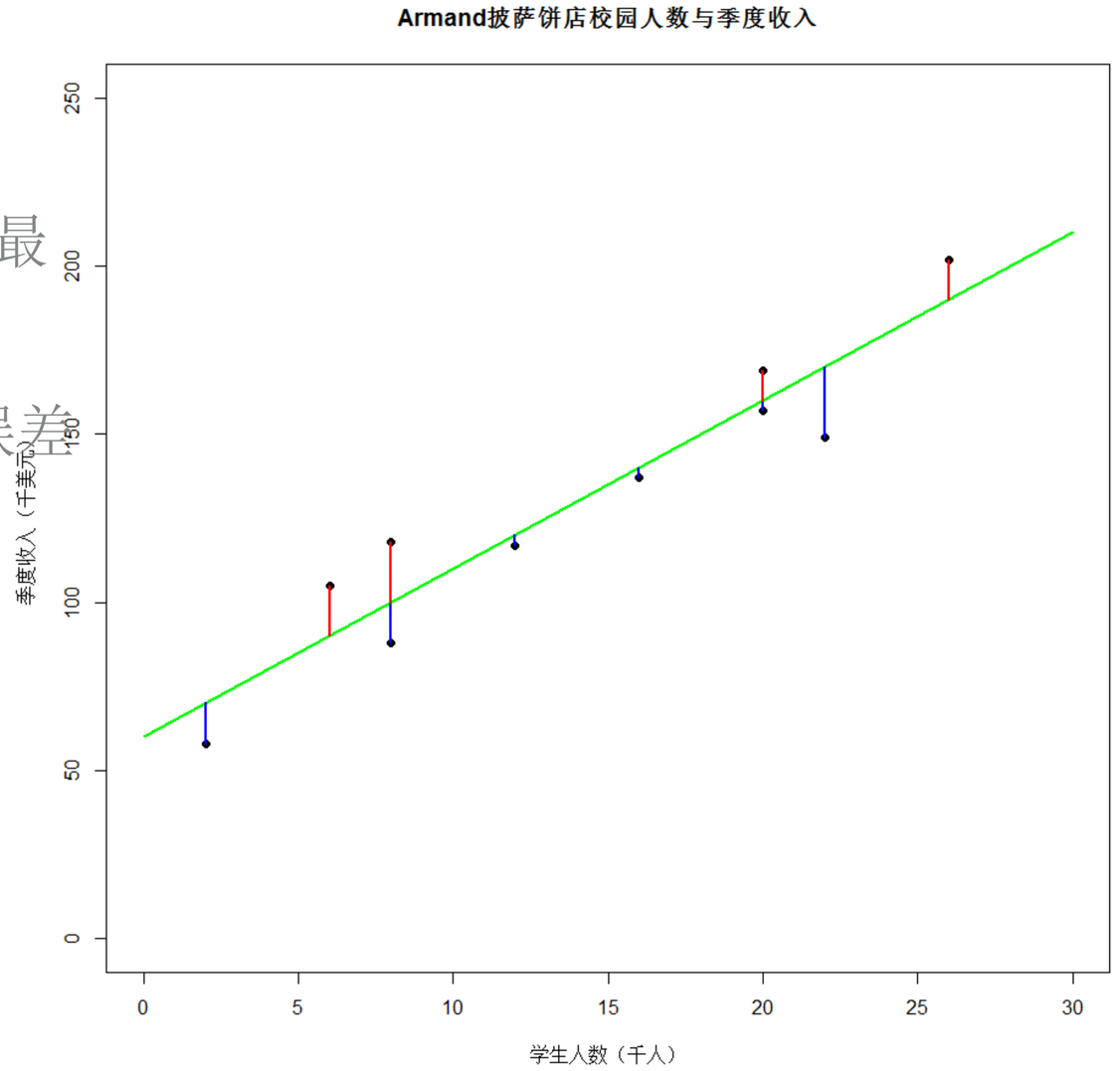
参数估计

- 训练数据：$\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$
- n：训练数据集中的样本个数
- $(x^{(i)}, y^{(i)})$：第i个训练样本
- 待估参数：
  - β_0
  - β_1
  - σ^2



参数估计

- 找一条线：$\hat{y}^{(i)} = \hat{\beta}_0 + \hat{\beta}_1 x^{(i)}$，使得总的误差最小
- 因为误差有正有负，会互相抵消，所以要使误差平方和最小
- 找$\hat{\beta}_0, \hat{\beta}_1$，使得$\sum_{i=1}^{n} \left(y^{(i)} - \beta_0 - \beta_1 x^{(i)}\right)^2$最小



参数估计——最小二乘估计

- 令$\hat{y}^{(i)} = \beta_0 + \beta_1 x^{(i)}$表示基于$x^{(i)}$对$y$的预测，则$e^{(i)} = y^{(i)} - \hat{y}^{(i)}$表示第$i$个观测的预测残差（residual）
- 定义残差平方和（residual sum of squares, RSS）：

$$RSS = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2 = \sum_{i=1}^{n} \left(y^{(i)} - \beta_0 - \beta_1 x^{(i)}\right)^2$$

- 最小二乘估计，即寻找最小化RSS的β_0与β_1：

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sum_{i=1}^{n}(x^{(i)} - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

- 附带估计

$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ （y_i的拟合值）

$\hat{\varepsilon}_i = y_i - \hat{y}_i$ （残差）

$\widehat{\sigma^2} = s^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}$ （σ^2的估计）

系数解释

- 右图：最小二乘估计学生人数与季度收入的线性回归模型
- 定量理解自变量与相应变量关系

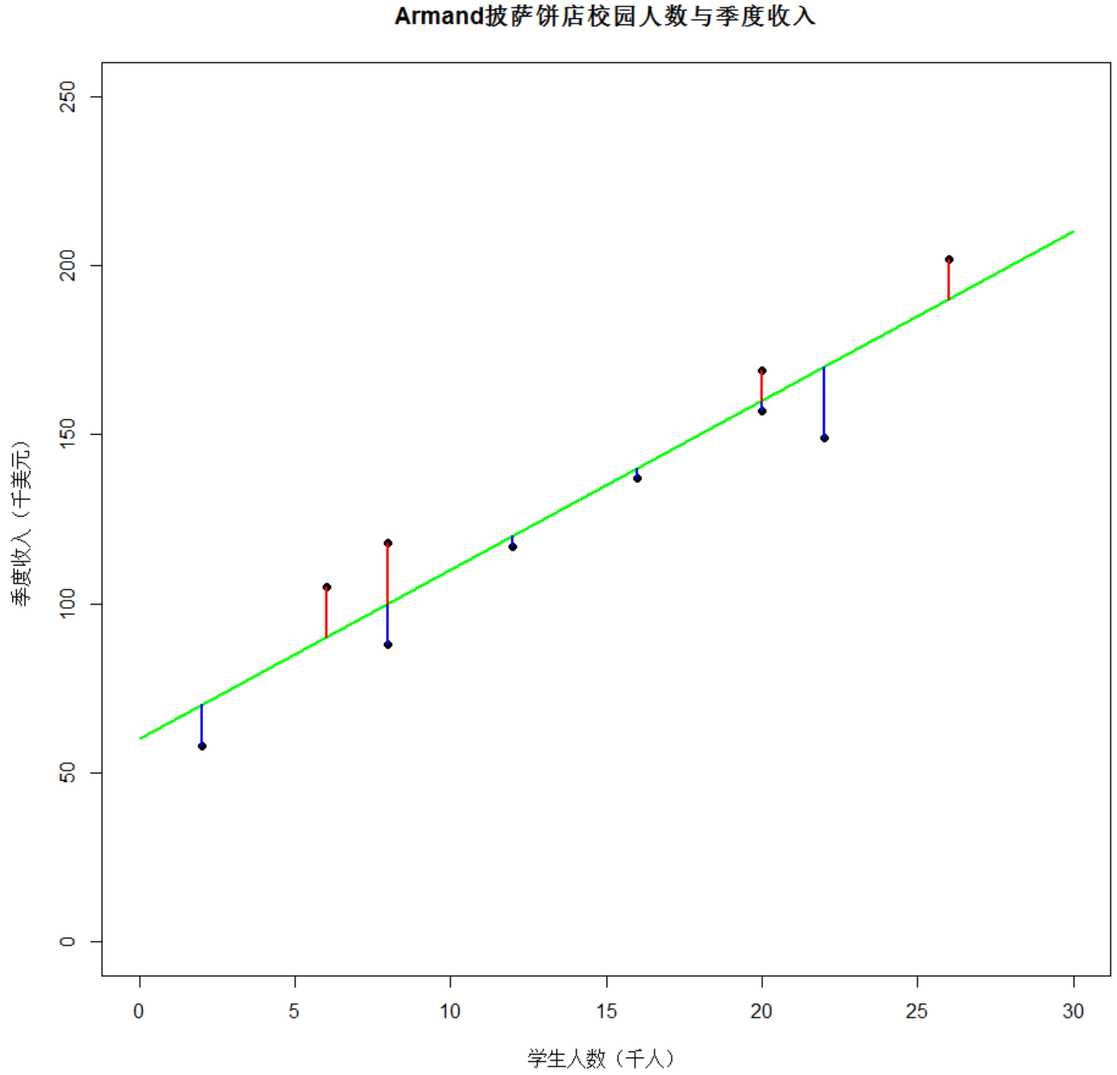
当附近没有校园，即校园人数为0 ($x = 0$)时，

假设此时模型仍然成立（由于x=0不在原数据的范围内，不能轻易推广模型结论），

连锁店的季度销售额期望的估计值为6万美元

校园学生人数每**增加**一个单位 (1000人)，连锁店**期望增加**的季度销售额估计值为5000美元；

校园学生人数每**减少**一个单位 (1000人)，连锁店**期望减少**的季度销售额估计值为5000美元；



参数估计——极大似然估计

- 假设$y^{(i)} = \beta_0 + \beta_1 x^{(i)} + \epsilon^{(i)}$，$\epsilon^{(i)}$独立同分布，分布为正态分布$N(0, \sigma^2)$
- 给定$(\beta_0, \beta_1, x^{(i)})$，$y^{(i)}$的分布为正态分布$N(\beta_0 + \beta_1 x^{(i)}, \sigma^2)$

$$f\left(y^{(i)} \middle| x^{(i)}; \beta_0, \beta_1\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \beta_0 - \beta_1 x^{(i)}\right)^2}{2\sigma^2}\right)$$

- 似然函数：

$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} f\left(y^{(i)} \middle| x^{(i)}; \beta_0, \beta_1\right) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \beta_0 - \beta_1 x^{(i)}\right)^2}{2\sigma^2}\right)$$

- 极大似然估计的思想告诉我们：应该选择使$L(\beta_0, \beta_1)$最大的参数(β_0, β_1)。

参数估计——极大似然估计

对数似然：$l(\beta_0,\beta_1)=\log L(\beta_0,\beta_1)$

$$=\log\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(y^{(i)}-\beta_0-\beta_1x^{(i)}\right)^2}{2\sigma^2}\right)$$

$$=\sum_{i=1}^{n}\log\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(y^{(i)}-\beta_0-\beta_1x^{(i)}\right)^2}{2\sigma^2}\right)$$

$$=n\log\frac{1}{\sqrt{2\pi}\sigma}-\frac{1}{2\sigma^2}\cdot\sum_{i=1}^{n}\left(y^{(i)}-\beta_0-\beta_1x^{(i)}\right)^2$$

- 最大化 $l(\beta_0,\beta_1)$ 等价于最小化 $\sum_{i=1}^{n}\left(y^{(i)}-\beta_0-\beta_1x^{(i)}\right)^2$，即RSS
- 在这样的概率假设下，(β_0,β_1)的最小二乘估计等价于极大似然估计

参数估计——极大似然估计

对数似然：$l(\beta_0,\beta_1)=\log L(\beta_0,\beta_1)$

$$=\log\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(y^{(i)}-\beta_0-\beta_1x^{(i)}\right)^2}{2\sigma^2}\right)$$

$$=\sum_{i=1}^{n}\log\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(y^{(i)}-\beta_0-\beta_1x^{(i)}\right)^2}{2\sigma^2}\right)$$

$$=n\log\frac{1}{\sqrt{2\pi}\sigma}-\frac{1}{2\sigma^2}\cdot\sum_{i=1}^{n}\left(y^{(i)}-\beta_0-\beta_1x^{(i)}\right)^2$$

- 最大化 $l(\beta_0,\beta_1)$ 等价于最小化 $\sum_{i=1}^{n}\left(y^{(i)}-\beta_0-\beta_1x^{(i)}\right)^2$，即RSS
- 在这样的概率假设下，(β_0,β_1)的最小二乘估计等价于极大似然估计

模型评估

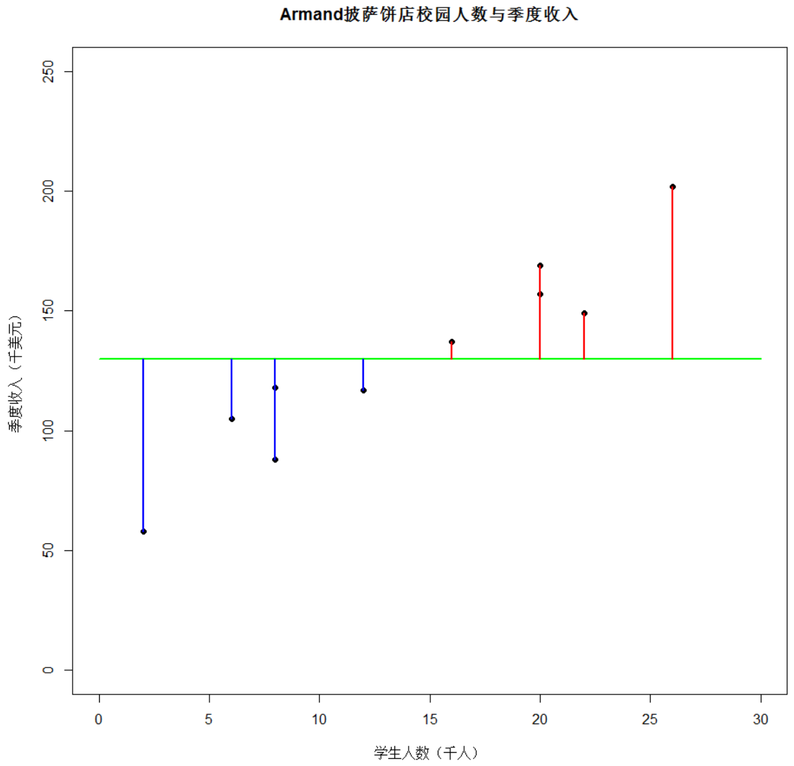
- 得到系数估计后，需思考以下问题
- 针对于我们的数据，简单线性回归到底好不好？有多好？
- 我们用最小二乘法估计出来的参数准不准确？显不显著？

模型评估

- 增加自变量X的好处在哪里？
- 比较有无X时，拟合Y的误差大小？

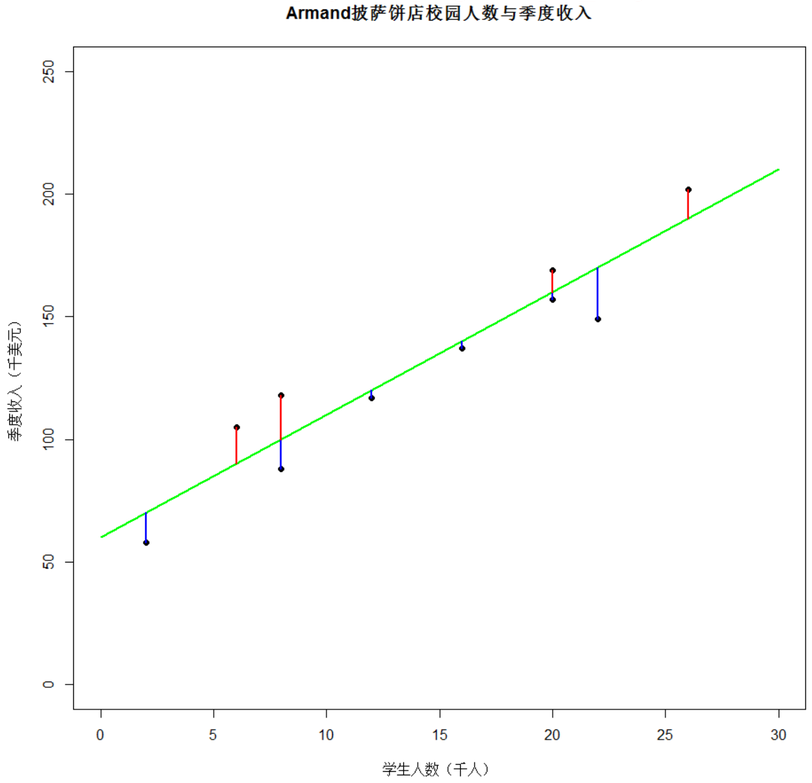
没有x_i时

拟合值：$\bar{y}$



有x_i时

拟合值：$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$



模型评估

- 增加自变量X的好处在哪里？
- 比较有无X时，拟合Y的误差大小？

- **没有x_i时**
  - **拟合值：$\bar{y}$**
  - 没有x_i时拟合值的误差总和：$\boldsymbol{SST = \sum_{i=1}^{n}(y_i - \bar{y})^2}$
  - **总平方误差**（Total Sum of Squares, SST）

- **有x_i时**
  - **拟合值：$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$**
  - 有x_i时拟合值的误差总和：$\boldsymbol{SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
  - **残差平方和**（Sum of Squares due to Error, SSE）

- 有x_i时减少的误差总和：$\boldsymbol{SSR = SST - SSE = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}$
  - **回归平方和**（Sum of Squares due to Regression, SSR）
  - 减少的误差总和，占，原有的误差总和，的比例：$R^2 = \frac{SSR}{SST}$

模型评估–判定系数R^2

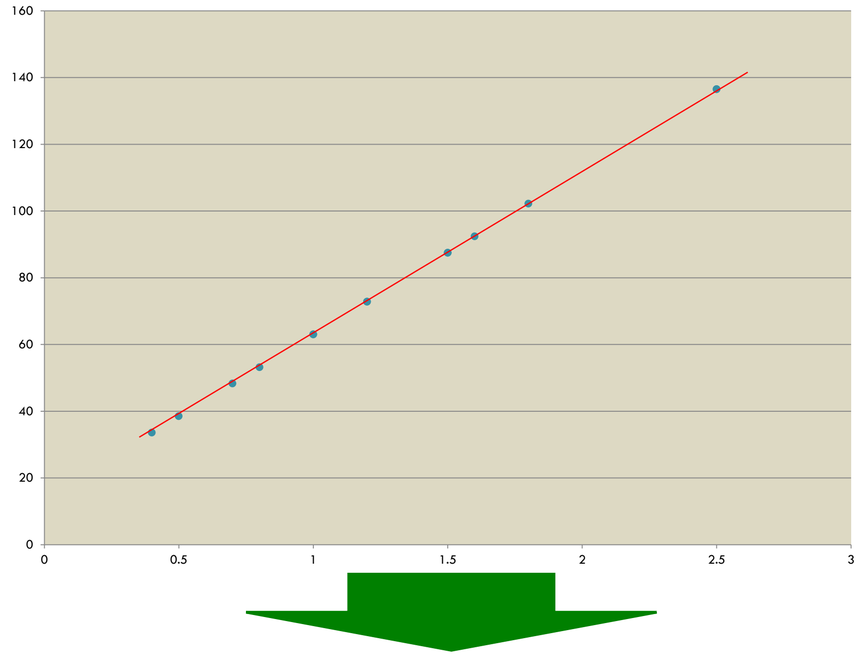
- $SST = \sum_{i=1}^{n}(y_i - \bar{y})^2$：没有$x$时的$Y$的不确定性
- $SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$：使用自变量$x$后的$Y$的不确定性
- $SSR = SST - SSE = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$：模型减少的不确定性（模型的解释能力）

减少了多少？
减少了$R^2 \times 100\%$

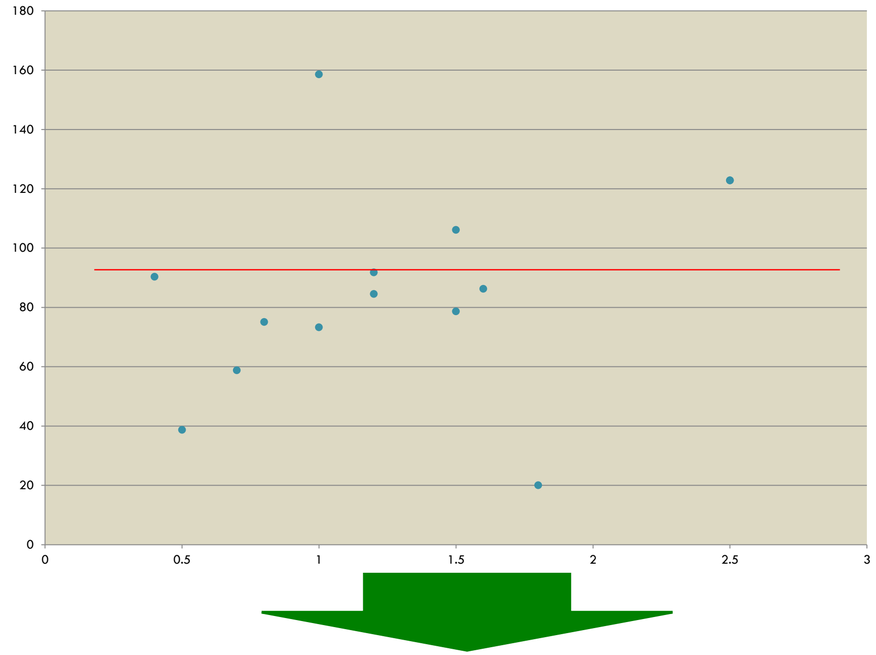
$R^2 = \frac{SSR}{SST}$　　（判定系数 coefficient of determination）

$0 \le R^2 \le 1$
R^2越接近于0，模型解释能力越差
R^2越接近于1，模型解释能力越强

模型评估–判定系数R^2

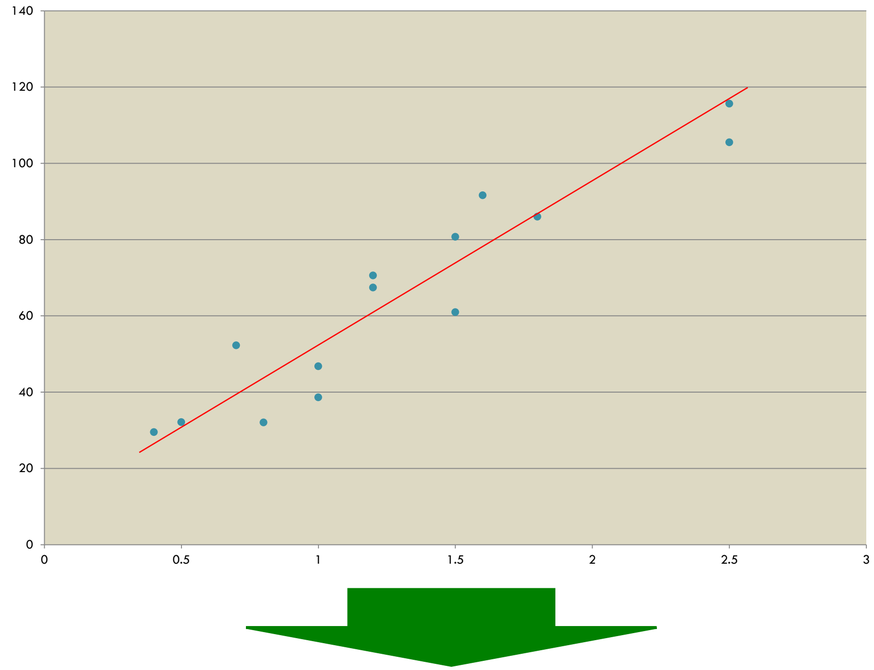


- $R^2=1$：响应变量Y取值的变差可以**完全**由自变量x的取值解释

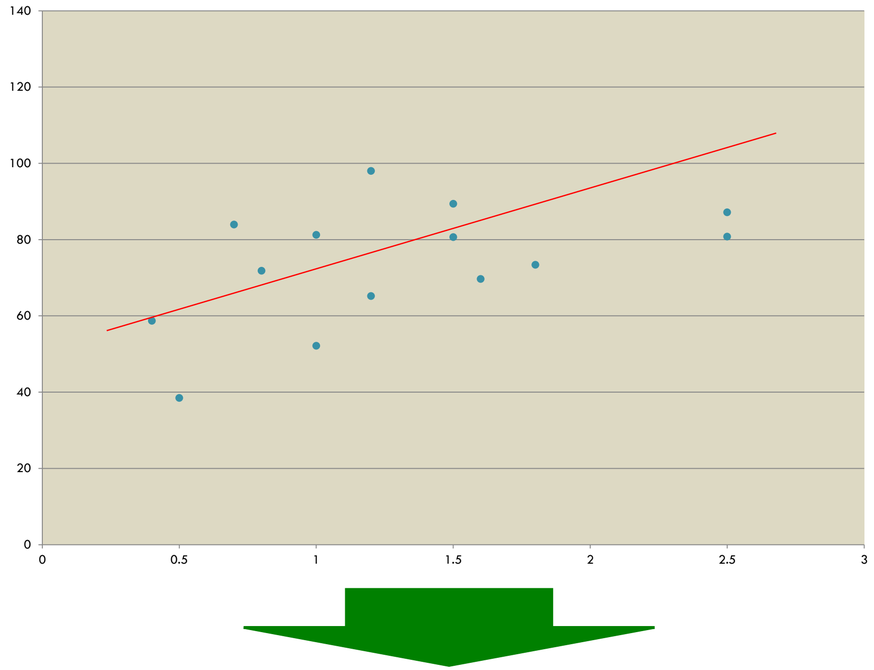


- $R^2=0$：响应变量Y取值的变差**不能**由自变量x的取值解释

模型评估–判定系数R^2



- $R^2=0.8$：响应变量Y取值的变差**大部分**可以由自变量x的取值解释



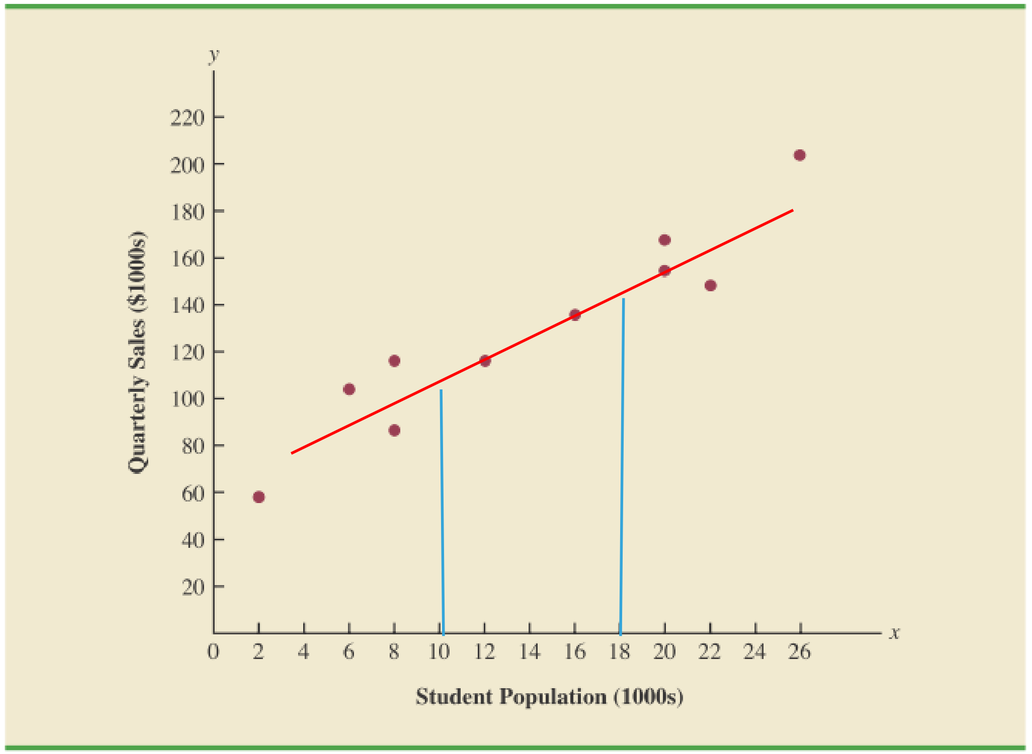
- $R^2=0.4$：响应变量Y取值的变差**一部份**可以由自变量x的取值解释

模型预测

- 定量理解： $\hat{\beta}_1$
  - 预测：假定学生数量x和季度销售额y之间关系不变，则在学生人数10000与18000的校园开店，预计季度销售额是多少？

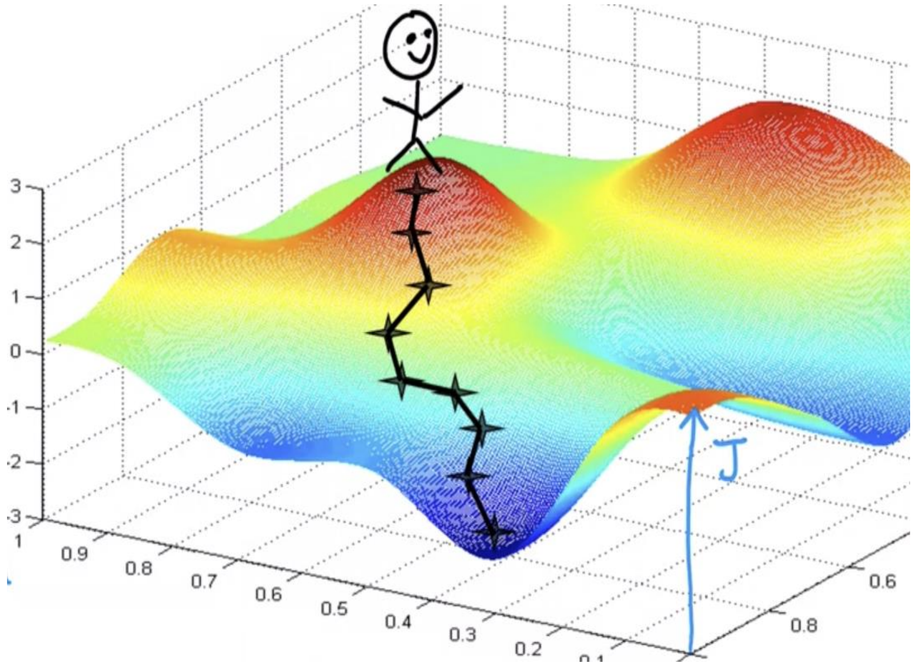
$$\hat{y}_i = 60 + 5x_i$$

FIGURE 14.3 SCATTER DIAGRAM OF STUDENT POPULATION AND QUARTERLY SALES FOR ARMAND'S PIZZA PARLORS



计算机视角——迭代优化

- 损失函数： $J(\beta_0,\beta_1)=\frac{1}{2n}\sum_{i=1}^{n}\left(y^{(i)}-\beta_0-\beta_1 x^{(i)}\right)^2$。
  目标：minimize $J(\beta_0,\beta_1)$。
- 迭代：从某个初始值出发，不断变换(β_0,β_1)使$J(\beta_0,\beta_1)$变小，直到幸运地收敛到一组(β_0,β_1)使$J(\beta_0,\beta_1)$达到最小。
- **梯度下降算法**（gradient descent algorithm）
- 当我站在山顶，环顾四周，思考一个问题：如果我将朝着一个方向迈出一小步，同时我想最快速度的抵达山谷，我该迈向哪个方向？
- 答案显而易见：朝着最陡峭的方向往下走



梯度下降算法

- 重复直到收敛：

$$\begin{cases} \beta_0 := \beta_0 - \alpha \dfrac{\partial}{\partial \beta_0} J(\beta_0, \beta_1) \\ \beta_1 := \beta_1 - \alpha \dfrac{\partial}{\partial \beta_1} J(\beta_0, \beta_1) \end{cases}$$

- α：学习率（learning rate），通常为很小的正数，控制每一步的步幅大小
- $\dfrac{\partial}{\partial \beta_j} J(\beta_0, \beta_1)$：偏导数，控制了每一步的方向和步幅大小
- **同时**更新所有参数

$$tmp\_w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$$tmp\_b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

$$w = tmp\_w$$

$$b = tmp\_b$$

Incorrect

$$tmp\_w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$$w = tmp\_w$$

$$tmp\_b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

$$b = tmp\_b$$

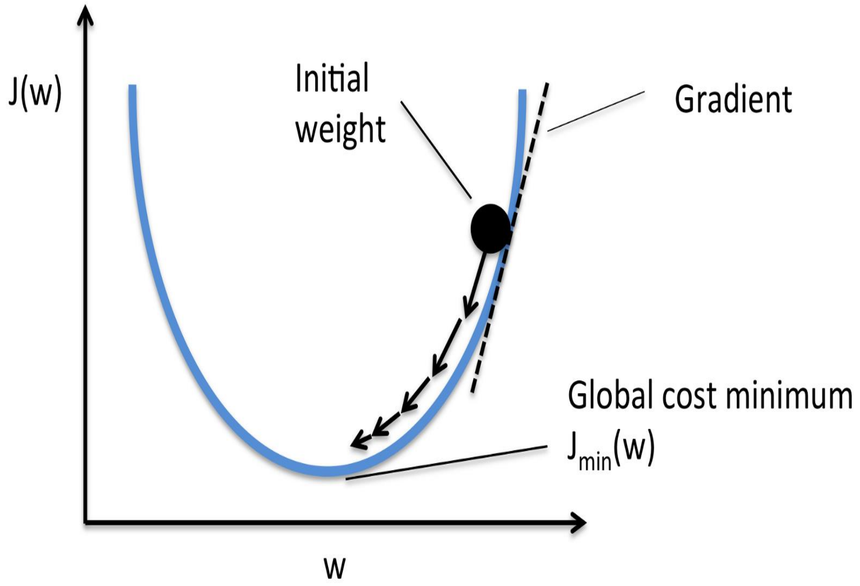
梯度下降——INTUITION

- 考虑一个更简单的例子（如右图）：

$$J(w)$$

$$w = w - \alpha \frac{\partial}{\partial w} J(w)$$

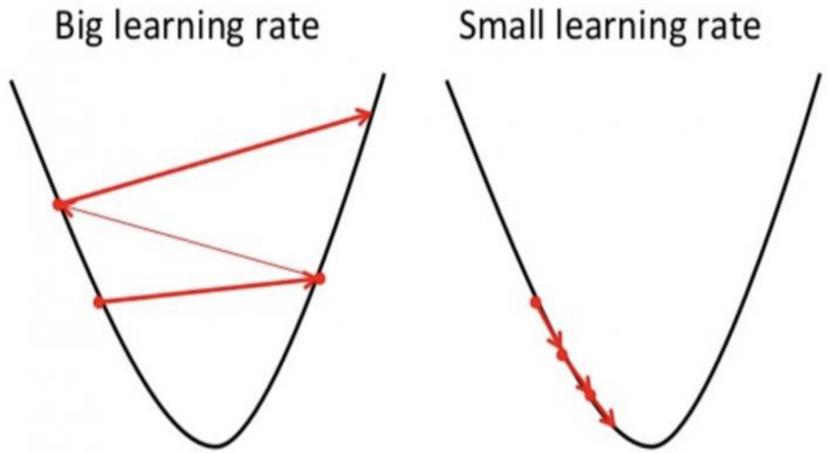
$$\min_{w} J(w)$$

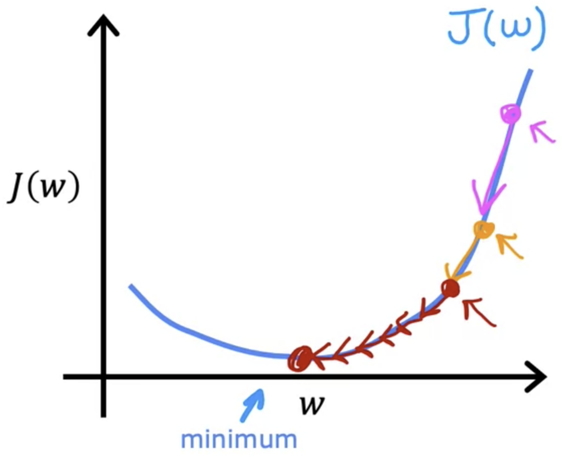


- 当w在最优点右侧时，$\frac{\partial}{\partial w} J(w) > 0$，$w = w - \alpha \cdot (\text{positive number})$，$w$减小
- 当w在最优点左侧时，$\frac{\partial}{\partial w} J(w) < 0$，$w = w - \alpha \cdot (\text{negative number})$，$w$增大
- 最终都会使w收敛到最优点

学习率α

- 梯度下降算法每次在J最快下降的方向上迈一步，学习率α决定了步幅
- α需要事先人为指定，属于超参数
  - α如果太小，梯度下降会很慢
  - α如果太大，梯度下降可能会错过最小值点，甚至不收敛
- 一个选取恰当的固定的α可以保证达到**局部**最优点





梯度下降——线性回归

- 首先我们考虑只有一个训练样本(x, y)

$$\frac{\partial}{\partial \beta_0} J(\beta_0, \beta_1) = \frac{\partial}{\partial \beta_0} \frac{1}{2}(y - \beta_0 - \beta_1 x)^2$$

$$= 2 \cdot \frac{1}{2}(y - \beta_0 - \beta_1 x) \cdot -1$$

$$= -(y - \beta_0 - \beta_1 x)$$

$$\frac{\partial}{\partial \beta_1} J(\beta_0, \beta_1) = \frac{\partial}{\partial \beta_1} \frac{1}{2}(y - \beta_0 - \beta_1 x)^2$$

$$= 2 \cdot \frac{1}{2}(y - \beta_0 - \beta_1 x) \cdot -x$$

$$= -(y - \beta_0 - \beta_1 x)x$$

- $\beta_0 \coloneqq \beta_0 + \alpha(y - \beta_0 - \beta_1 x)$
- $\beta_1 \coloneqq \beta_1 + \alpha(y - \beta_0 - \beta_1 x)x$

- 更新的幅度与$(y - \beta_0 - \beta_1 x)$成正比。
- 因此，如果这个样本的预测值和实际的y很接近，那么几乎没有必要调整参数。
- 反之，如果预测值有很大误差，那么我们就需要大幅调整参数。

梯度下降——多个训练样本

- 当有大量训练样本时，我们有了不同的考虑（收敛速度、计算速度），衍生出了不同的梯度下降算法：
- 批量梯度下降（batch gradient descent, BGD）
- 随机梯度下降（stochastic gradient descent, SGD）
- 小批量梯度下降（mini-batch gradient descent, MBGD）
- ……

批量梯度下降

- Repeat until convergence {

$$\beta_0 := \beta_0 + \alpha \cdot \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \beta_0 - \beta_1 x^{(i)}\right)$$

$$\beta_1 := \beta_1 + \alpha \cdot \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \beta_0 - \beta_1 x^{(i)}\right) x^{(i)}$$

}

- 在每次迭代时，使用整个训练数据集的信息

随机梯度下降

- Repeat until convergence {

  for i from 1 to n {

  $$\beta_0 := \beta_0 + \alpha\left(y^{(i)} - \beta_0 - \beta_1 x^{(i)}\right)$$

  $$\beta_1 := \beta_1 + \alpha\left(y^{(i)} - \beta_0 - \beta_1 x^{(i)}\right)x^{(i)}$$

  }

}

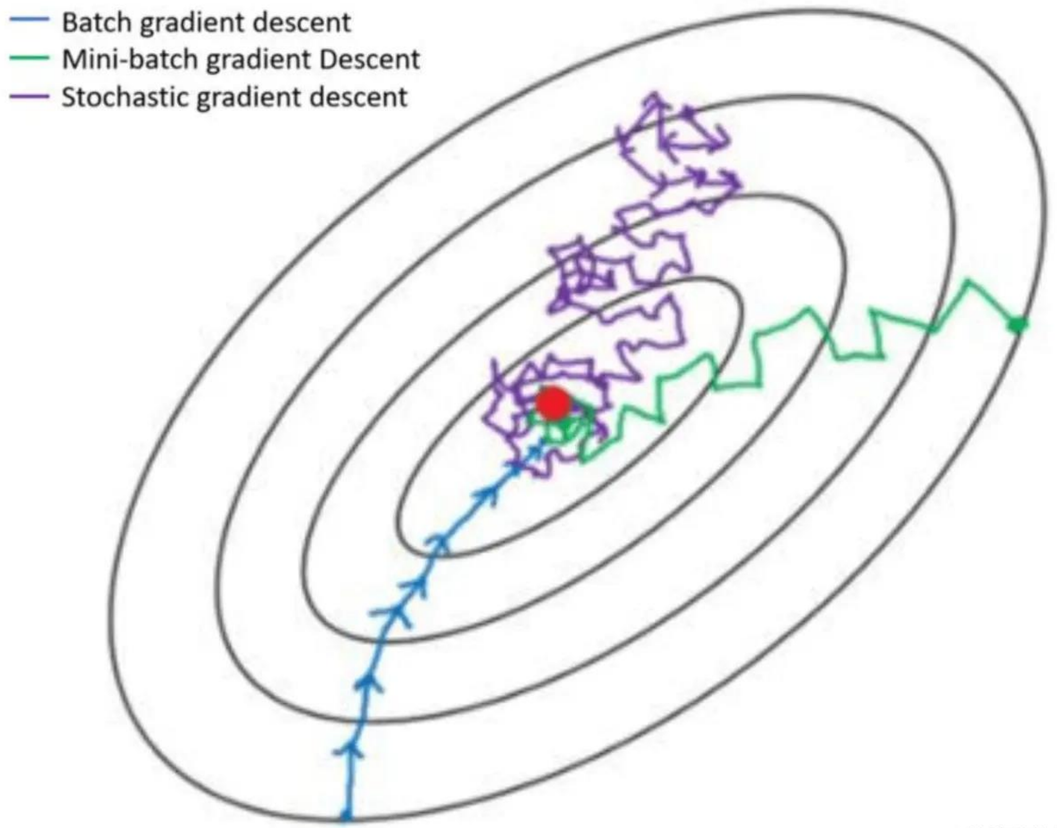
- 在每次迭代时只用1个训练数据
- 随机：虽然不是每次迭代得到的损失函数都向着全局最优方向， 但是大的整体的方向是向全局最优解的，最终的结果往往是在全局最优解附近。

BGD VS. SGD

- BGD
  - 每次迭代所有样本都对参数的调整有贡献，其计算得到的是一个标准梯度，对于凸优化问题一定可以得到全局最优
  - 样本量很大时，一次迭代耗时巨大
- SGD
  - 计算快，快速收敛；但不能使用向量化计算，浪费计算机性能
  - 单个样本的梯度并不是准确的梯度，因此不是每次迭代得到的损失函数都向着全局最优方向， 但是大的的方向是向着全局最优的，最终的结果往往是在全局最优附近，可以接受
- 有没有折中方案，综合两个方法的优点？

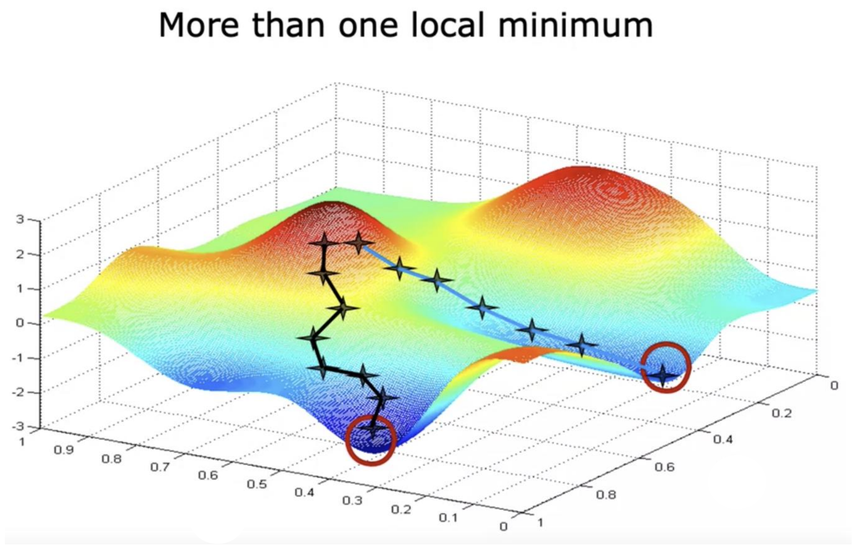
小批量梯度下降(MINI-BATCH)

- 每次更新时利用b个样本数据运行BGD，b远小于n
- 思想：用mini batch的梯度来近似batch的梯度，获取比SGD更好的收敛性以及比BGD更快的速度
- 在深度学习中广泛应用

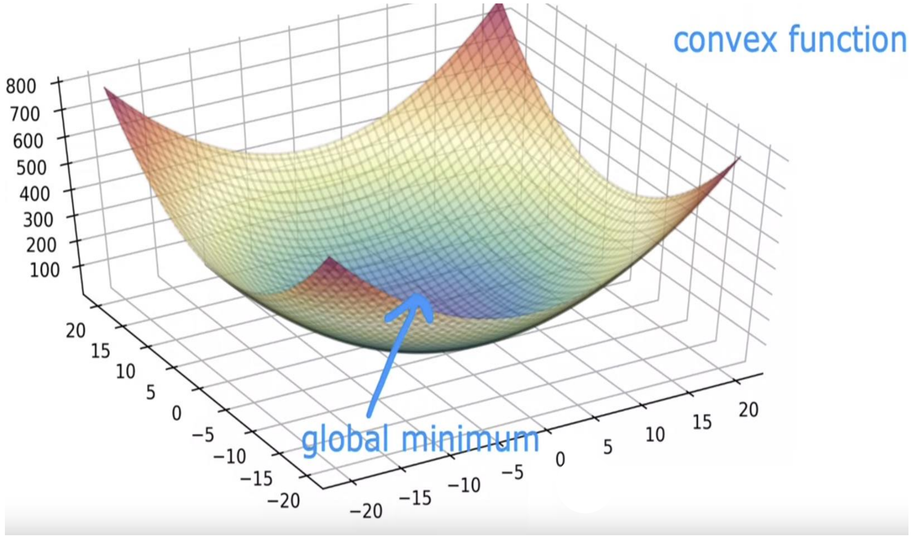


局部最优&全局最优

- 当损失函数存在多个局部最优点时，GD容易收敛到局部最优，而不是全局最优
- 主要取决于初始点的选取



- 线性回归没有这个问题，因为其损失函数是一个凸函数，只有一个全局最小值点
- 只要选取合适的学习率，就一定可以收敛到全局最优



多元线性回归

多元线性回归

- 同时考虑多个特征与响应变量Y的关系，最直接的做法是对每个特征单独进行一元线性回归
  - 同时给定多个特征的值，无法对响应变量进行预测
  - 没有考虑特征之间的协同关系
- $Y = f(X_1, \dots, X_d) + \epsilon = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_d X_d + \epsilon$
- β_j表示保持其他特征不变的情况下，一个单位X_j的提升对Y的平均效应

- 引入$X_0 = 1, \quad \boldsymbol{\beta} = (\beta_0, \beta_1 \dots, \beta_d)^T, \quad \boldsymbol{X} = (X_0, X_1 \dots, X_d)^T$
- 可以得到向量化表示：$Y = \boldsymbol{\beta^T X} + \epsilon = \sum_{j=0}^{d} \beta_j X_j + \epsilon$

记号

- 训练数据：$\{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})\}$，其中$\boldsymbol{x}^{(i)} = \left(x_0^{(i)}, x_1^{(i)}, \ldots, x_d^{(i)}\right)^T$
- n：训练数据集样本个数
- d：特征维数
- $(\boldsymbol{x}^{(i)}, y^{(i)})$：第$i$个训练样本

最小二乘估计

- 给定$\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_d$，我们可以根据特征对$y$进行预测：

$$\hat{y} = \sum_{j=0}^{d} \hat{\beta}_j x_j$$

- 与一元线性回归相同，我们最小化残差平方和

$$RSS = \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2 = \sum_{i=1}^{n} \left(y^{(i)} - \hat{\beta}_0 - \hat{\beta}_1 x_1^{(i)} - \cdots - \hat{\beta}_d x_d^{(i)}\right)^2$$

就可以得到多元最小二乘估计：

$$\text{Normal Equation: } \widehat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{Y},\ \boldsymbol{X} = \left(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\right)^T, \boldsymbol{Y} = (y_1, \ldots, y_n)^T$$

NORMAL EQUATION

- 只针对线性回归
- 无需迭代即可估计$\boldsymbol{\beta}$
- 缺点：
  - 无法推广到其他的机器学习算法
  - 当特征维度很高时计算太慢

- 成熟的机器学习软件包中通常默认使用Normal Equation来实现线性回归
- 除此之外，我们仍可以使用梯度下降算法

多元回归例子

- 例：Country Kitchen公司零食年销售额

地区	年销售额（百万美元）	广告投放额（百万美元）	促销投放额（百万美元）	往年竞争者年销售额（百万美元）
Selkirk	101.8	1.3	0.2	20.4
Susquehanna	44.4	0.7	0.2	30.5
Kittery	108.3	1.4	0.3	24.6
Acton	85.1	0.5	0.4	19.6
Finger Lakes	77.1	0.5	0.6	25.5
Berkshire	158.7	1.9	0.4	21.7
Central	180.4	1.2	1.0	6.8
Providence	64.2	0.4	0.4	12.6
Nashua	74.6	0.6	0.5	31.3
Dunster	143.4	1.3	0.6	18.6
Endicott	120.6	1.6	0.8	19.9
Five-Towns	69.7	1.0	0.3	25.6
Waldeboro	67.8	0.8	0.2	27.4
Jackson	106.7	0.6	0.5	24.3
Stowe	119.6	1.1	0.3	13.7

$i = 1, \ldots, n$

$n = 15$

自变量的个数$p = 3$

多元回归例子

- 例：Country Kitchen公司零食年销售额
- 模型：$Y_i = \beta_0 + X_{i1}\beta_1 + X_{i2}\beta_2 + X_{i3}\beta_3 + \epsilon_i$，　ϵ_i零均值、同方差、不相关

β_0的估计值：$\hat{\beta}_0 = 65.70$

当所有的自变量为0时，公司年销售额期望的估计值为65.70百万美元（若此时模型还成立）

β_1的估计值：$\hat{\beta}_1 = 48.98$

给定所有其他自变量保持不变时，每**增加**一个单位（1百万美元）的广告投放额，公司期望**增加**的年销售额估计值为48.98百万美元

β_2的估计值：$\hat{\beta}_2 = 59.65$

给定所有其他自变量保持不变时，每**增加**一个单位（1百万美元）的促销投放额，公司期望**增加**的年销售额估计值为59.65百万美元

β_3的估计值：$\hat{\beta}_3 = -1.84$

给定所有其他自变量保持不变时，竞争者年销售额每**增加**一个单位（1百万美元），公司期望**减少**的年销售额估计值为1.84百万美元

模型总体评估

- $\boldsymbol{SST = \sum_{i=1}^{n}(y_i - \bar{y})^2}$：没有$x$时的$Y$的不确定性
- $\boldsymbol{SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$：使用自变量$x$后的$Y$的不确定性
- $\boldsymbol{SSR = SST - SSE = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}$：模型减少的不确定性（模型的解释能力）

$R^2 = \frac{SSR}{SST}$　　（判定系数 coefficient of determination）

- R^2的一个问题：其总是随着自变量个数p的增加而增加
- 在一个模型中引入与响应变量独立的自变量时也可以在表面上提高R^2
  - 修正判定系数：Adjusted R^2：

$$R_a^2 = 1-(1-R^2)\left(\frac{n-1}{n-p-1}\right)$$

  - 相比于R^2，Adjusted R^2还考虑了模型中使用的自变量个数p与样本容量n

模型显著性检验

模型：$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i, i = 1, \cdots, n$ ， $\varepsilon_i \sim_{i.i.d.} N(0, \sigma^2)$

- $H_0: \beta_1 = \cdots = \beta_p = 0$ v. s. $H_1: not\ H_0$

- 如果原假设成立，则表明模型中所有的p个自变量与Y的取值均无关，也就是说整个回归模型是无效的

- H_0成立时，$F = \frac{SSR/p}{SSE/(n-p-1)} \sim F(p, n-p-1)$

- 零食年销售额例子中，检验统计量 $F = 18.29$. p–值为 $0.0001388 < 0.05$. 在0.05水平下拒绝H_0.
- 结论：整个回归模型是显著有效的

参数显著性检验

模型：$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i, i = 1, \cdots, n$ ， $\varepsilon_i \sim_{i.i.d.} N(0, \sigma^2)$

- **模型的显著性检验**
- $H_0: \beta_1 = \cdots = \beta_p = 0$ v. s. $H_1: not\ H_0$
- F检验

- **各个系数的显著性检验**
- $H_0: \beta_i = 0$ v. s. $H_1: \beta_i \neq 0$
- t检验

模型评估/模型诊断

模型：$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i, i = 1, \cdots, n$ ， $\varepsilon_i \sim_{i.i.d.} N(0, \sigma^2)$

潜在问题

- 真实的回归函数并不是线性的（绘制变量间图形）
- 该模型可能忽略了一些重要的预测因子（变量选择）
- 误差可能有着非恒定方差（残差散点图及检验）
- 误差可能并不独立（残差检验）
- 误差可能并不服从正态分布（残差检验）
- 数据中存在着异常值（异常值检验）
- 自变量之间本身存在着较强相关性（共线性检验）

定性自变量

- 考虑考虑一个简单的回归模型，它由一个定量(连续) 变量X_1 和具有两个水平M_1 和M_2 的定性变量X_2 所构成
- 定义哑变量（Dummy variable）

$$X_2 = \begin{cases} 1, \text{若水平为} M_1 \\ 0, \text{若水平为} M_2 \end{cases} \longrightarrow \text{基准组}$$

- 回归模型可写为

$$E(Y|X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

$$E(Y|X) = \begin{cases} \beta_0 + \beta_1 X_1 + \beta_2, & \text{若水平为} M_1 \\ \beta_0 + \beta_1 X_1, & \text{若水平为} M_2 \end{cases}$$

- 系数解释：其余变量不变时，与基准组相比，该水平下因变量期望的差异

定性自变量

- 模型中，可以有多个定性变量
- 每个定性变量可以有多个水平
- 设置比水平数少一个的虚拟变量。
- 例如季节这一定性变量，有春、夏、秋、冬四个水平，应该引入多少个变量？

$$X_1 = \begin{cases} 1, \text{夏天} \\ 0, \text{else} \end{cases} \qquad X_2 = \begin{cases} 1, \text{秋天} \\ 0, \text{else} \end{cases} \qquad X_3 = \begin{cases} 1, \text{冬天} \\ 0, \text{else} \end{cases}$$

- 交互效应模型：进一步考虑在不同水平下，其他自变量的影响不同，即各水平下其他变量线性模型的斜率不同（引入了其他变量与定性变量的交叉项）

梯度下降算法

	普通表示	向量化表示
参数	$\beta_0, \beta_1, \dots, \beta_d$	$\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_d)^T$
模型	$f(x_0, x_1, \dots, x_d) = \sum_{j=0}^{d} \beta_j x_j$	$f(\boldsymbol{x}) = \boldsymbol{\beta^T x}$
损失函数	$J(\beta_0, \beta_1, \dots, \beta_d) = \frac{1}{2n} \sum_{i=1}^{n} \left(y^{(i)} - \sum_{j=0}^{d} \beta_j x_j^{(i)} \right)^2$	$J(\boldsymbol{\beta}) = \frac{1}{2n} \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\beta^T x^{(i)}} \right)^2$
梯度下降算法	$\beta_j := \beta_j - \alpha \frac{\partial}{\partial \beta_j} J(\beta_0, \beta_1, \dots, \beta_d)$	$\beta_j := \beta_j - \alpha \frac{\partial}{\partial \beta_j} J(\boldsymbol{\beta}), \qquad j = 0, \dots, d$

梯度下降算法

- 重复直到收敛 {

$$\beta_j := \beta_j + \alpha \cdot \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \beta^T \boldsymbol{x}^{(\boldsymbol{i})}\right)x_j^{(i)}, j = 0, \ldots, d$$

}

- 同时更新$\beta_j, j = 0, \ldots, d$

总结

- 本章我们主要学习了：
  - 一元/多元线性回归
  - 最小二乘估计
  - 梯度下降算法