经典模型

第二章 线性回归分析

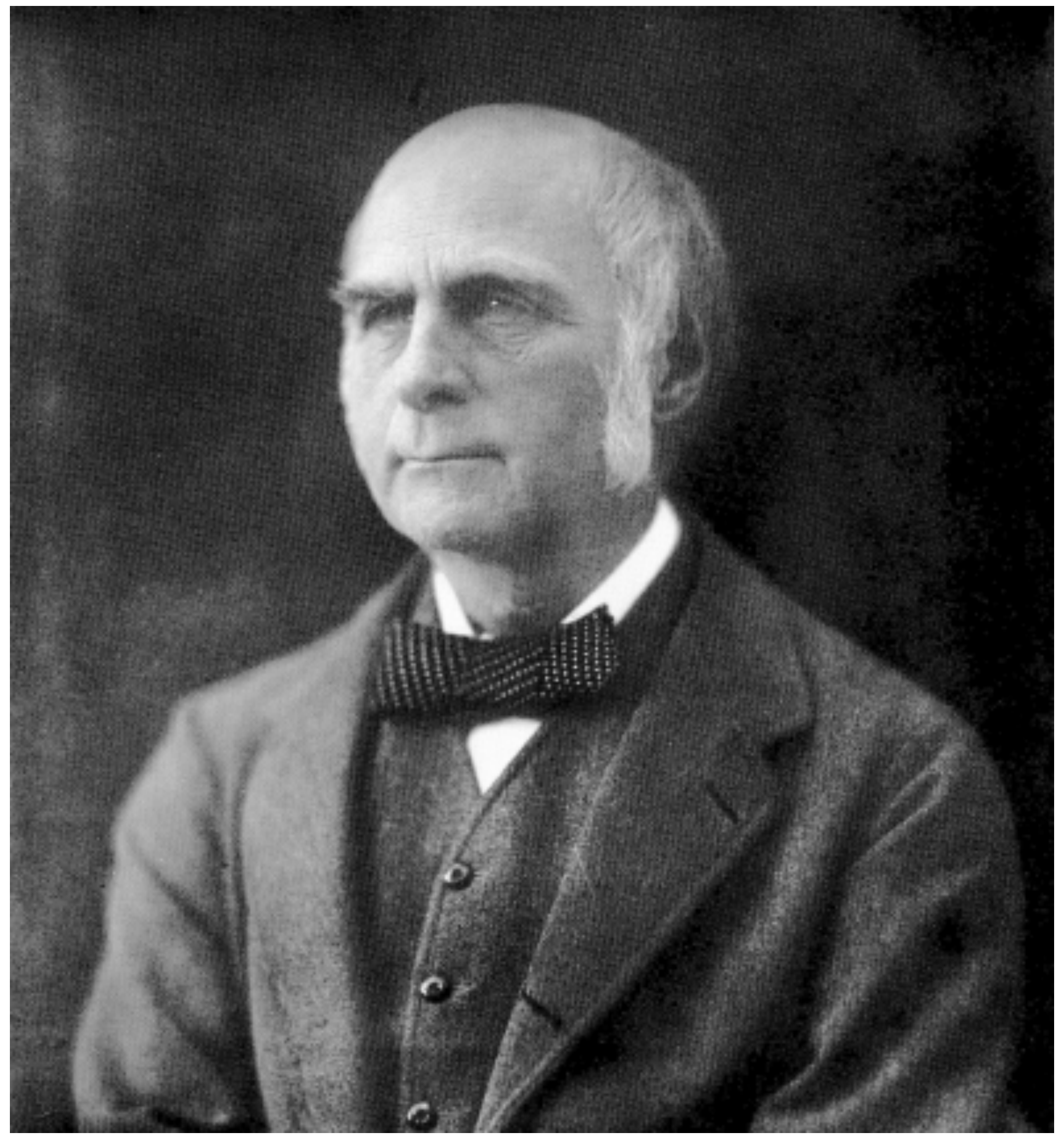
1. 回归分析
2. 最小二乘线性回归
3. 正则化
4. 小练习
5. 小结

1. 回归分析

19世纪，英国著名生物学家兼统计学家弗朗西斯·高尔顿（Francis Galton），在研究人类遗传问题时发现：父母高于平均身高时，他们的儿子身高比他更高的概率要小于比他更矮的概率；父母矮于平均身高时，他们的儿子身高比他更矮的概率要小于比他更高的概率。父母的身高可以预测子女的身高，两者近乎一条直线。身高有“回归”到平均数去的趋势。

若 x 表示父母平均身高，则孩子身高

$$y = 0.8567 + 0.516x.$$


弗朗西斯·高尔顿

- 回归分析（**Regression analysis**）是一种预测性的建模，研究**自变量**（**independent variables**）和**因变量**（**目标，dependent variables**）之间的关系。
- 回归分析常用于预测分析，时间序列模型以及发现变量之间的因果关系。
- 回归问题的数学描述：

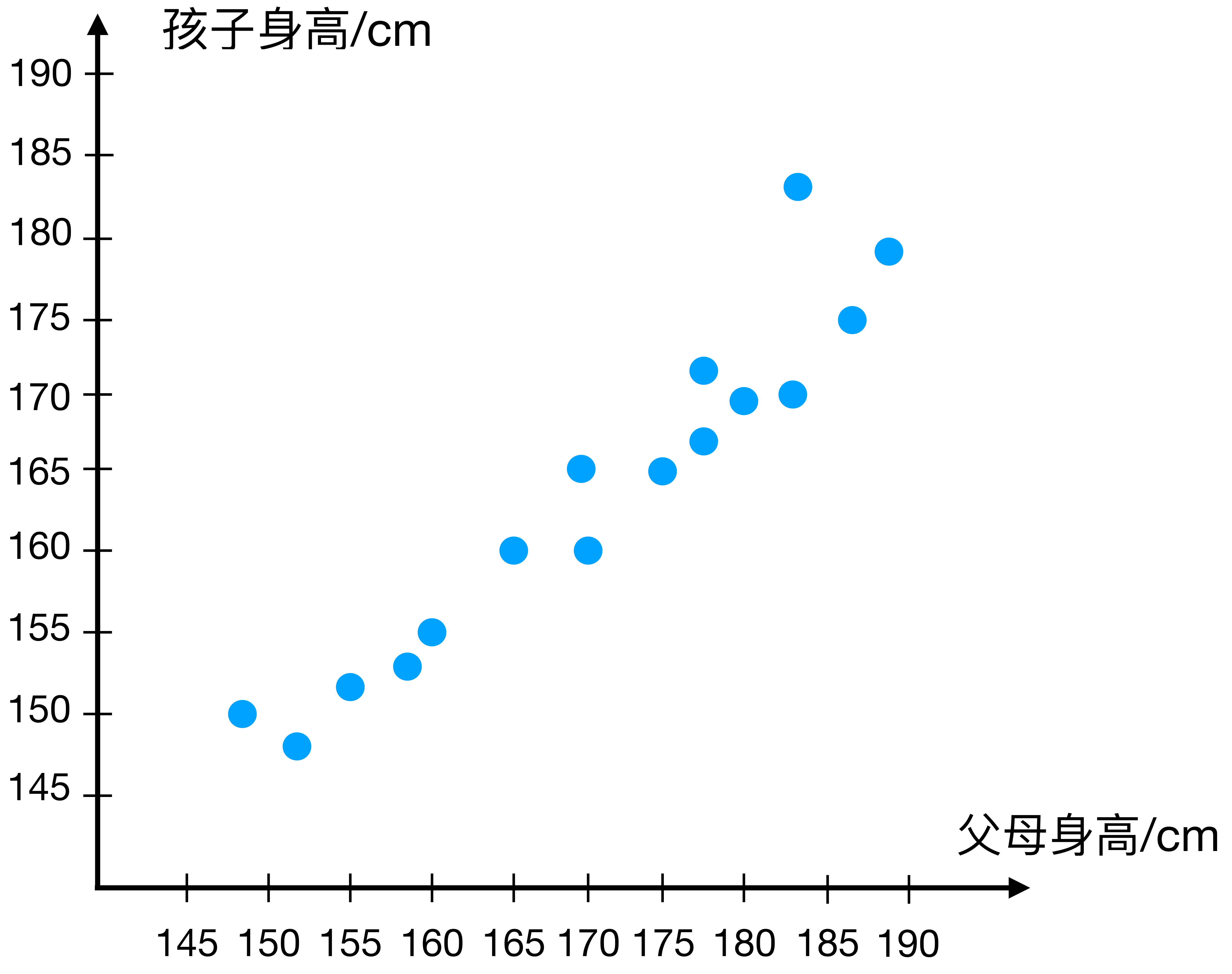
  给定训练集 $\{(\boldsymbol{x}_i, y_i), i = 1,\ldots,n\}, \boldsymbol{x}_i \in R^P, y_i \in R$

  其中 $y_i = f(\boldsymbol{x}_i) + \epsilon_i$

  要求：以训练集学习模型，对未知观测 $\boldsymbol{x}$ 进行预测： $\hat{y} = \hat{f}(\boldsymbol{x})$

2. 最小二乘线性回归

对于一类数据，出现近似如右图的分布，可考虑使用**线性模型**对数据进行拟合。



线性模型形式： $f(\boldsymbol{x}) = \mathbf{w}^T\mathbf{x} + b$

线性回归

给定数据集 $D\ \{(\boldsymbol{x}_i, y_i), i = 1,2,...,n\}, \boldsymbol{x}_i \in R^P, y_i \in R$，基于数据集 D 学习一个线性预测模型 $f(\boldsymbol{x})$，使之对于任意的 $\boldsymbol{x} \in R^p$，尽可能准确预测实值输出 $\hat{y} = f(\boldsymbol{x})$。

其中模型（预测函数）为线性回归模型：

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b,$$

其中

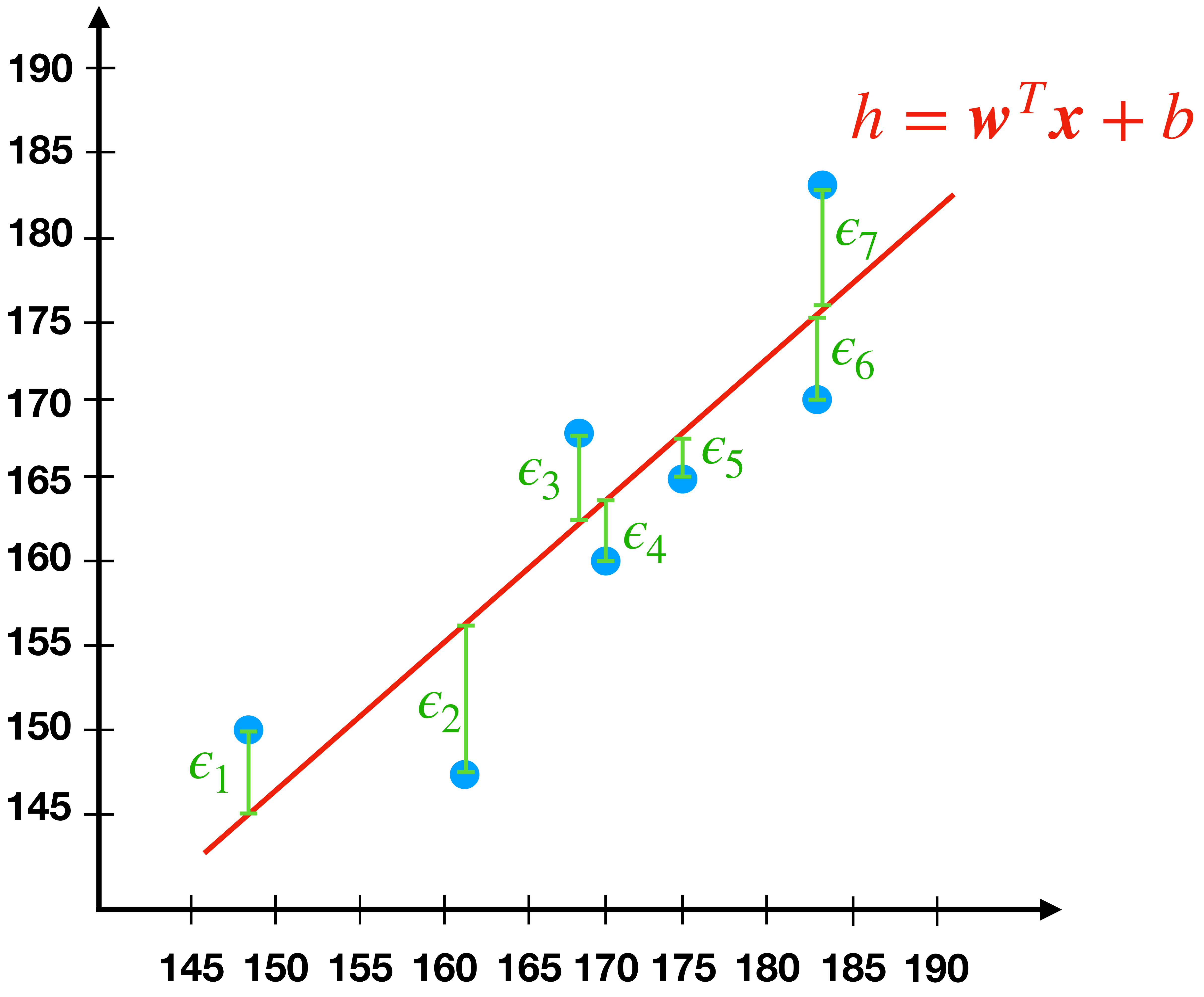
$$\boldsymbol{x} = [x_1, x_2, \ldots, x_p]^T \quad \boldsymbol{w} = [w_1, w_2, \ldots, w_p]^T.$$

最小二乘法

最小二乘准则：

各个训练样本预测残差平方和最小。线性回归模型使用最小二乘法进行训练。

即期望使得 ϵ^2 的和最小。



最小二乘法

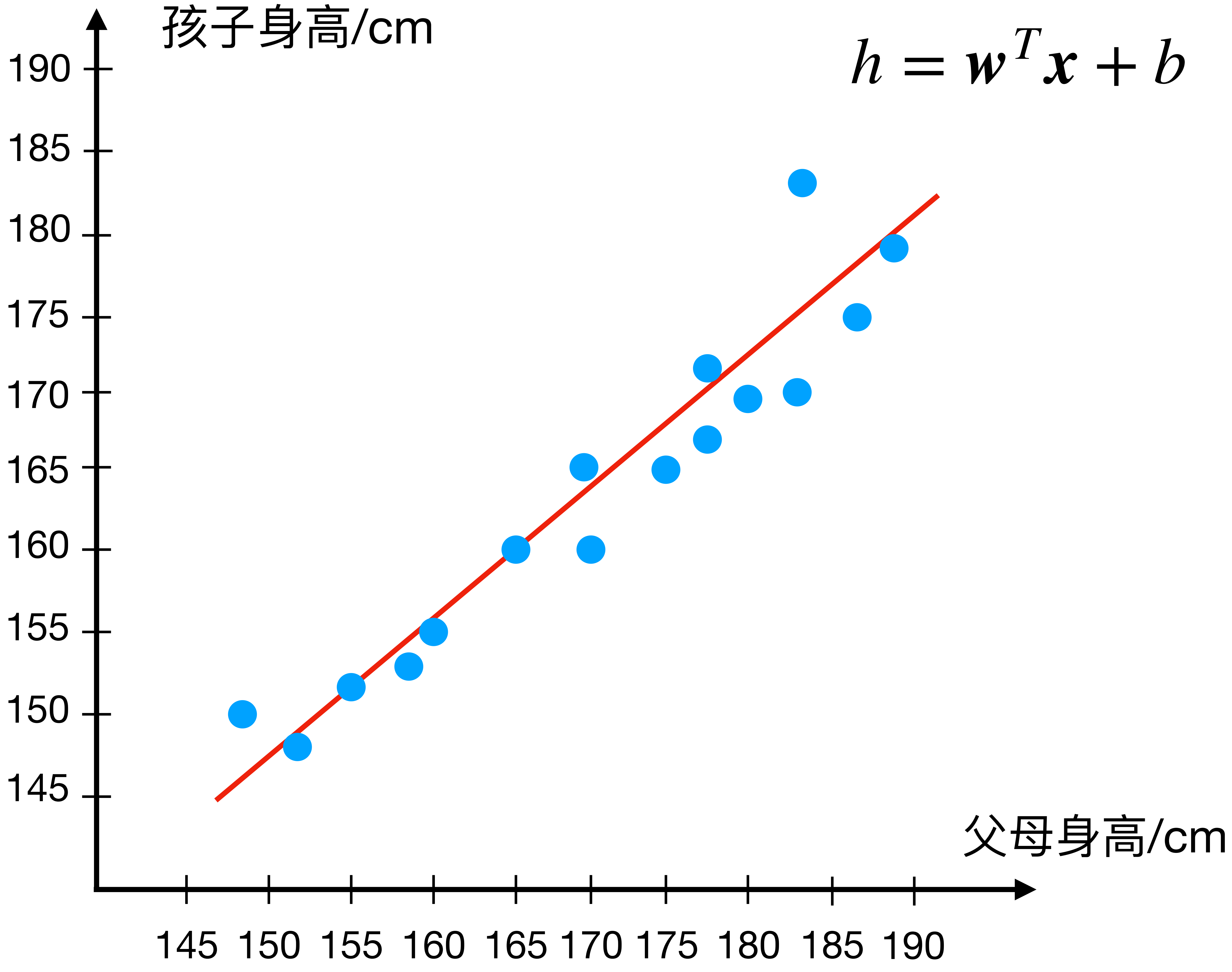
线性回归模型使用最小二乘法进行训练。

根据最小二乘准则，可构建代价函数：

$$J(\boldsymbol{w}, b) = \sum_{i=1}^{n} \epsilon^2 = \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i))^2.$$

通过最小化代价函数，求得模型的参数

$$[\boldsymbol{w}^*, b^*] = \arg\min_{\boldsymbol{w}, b} J(\boldsymbol{w}, b) = \arg\min_{\boldsymbol{w}, b} \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i))^2.$$



如果使用最小二乘法求得模型参数，可得到如左图模型图像（红色直线）。

如何求得模型参数?

模型参数的两种求解方法

最小二乘法的目的是最小化代价函数，求得模型参数，实际中可使用如下方法求解参数：

- **解析法**：根据函数在极值满足参数的梯度为 0 的特点进行求解。当样本数量较少时使用此法速度较快，但可能遇到矩阵不可逆的情况。
- **数值优化法**：利用梯度下降法等方法迭代求解。当样本数量较多时使用此法较合适，但优化算法是否收敛以及收敛速度不确定。

解析法

要求解 $[\boldsymbol{w}^*, b^*] = \underset{\boldsymbol{w},b}{\arg\min}\, J(\boldsymbol{w}, b) = \underset{\boldsymbol{w},b}{\arg\min} \sum_{i=1}^{n} [y_i - f(\boldsymbol{x})]$

令： $\dfrac{\partial J(\boldsymbol{w}, b)}{\partial \boldsymbol{w}} = 0 \quad \dfrac{\partial J(\boldsymbol{w}, b)}{\partial b} = 0$

相关公式

1. 矩阵相乘的转置 $(AB)^T = B^T A^T$

2. 若两个向量

$$\boldsymbol{a} = \boldsymbol{a}_1 + \boldsymbol{a}_2, \boldsymbol{b} = \boldsymbol{b}_1 + \boldsymbol{b}_2$$

则

$$\begin{aligned}
\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^T \boldsymbol{b} = \boldsymbol{b}^T \boldsymbol{a} \\
= (\boldsymbol{a}_1 + \boldsymbol{a}_2)^T (\boldsymbol{b}_1 + \boldsymbol{b}_2) \\
= \boldsymbol{a}_1^T \boldsymbol{b}_1 + \boldsymbol{a}_1^T \boldsymbol{b}_2 + \boldsymbol{a}_2^T \boldsymbol{b}_1 + \boldsymbol{a}_2^T \boldsymbol{b}_2
\end{aligned}$$

3. 若向量 $\boldsymbol{a} = \boldsymbol{a}_1 + \boldsymbol{a}_2$

则

$$
\begin{aligned}
\|\boldsymbol{a}\|^2 &= \boldsymbol{a} \cdot \boldsymbol{a} = \boldsymbol{a}^T \boldsymbol{a} \\
&= (\boldsymbol{a}_1 + \boldsymbol{a}_2)^T (\boldsymbol{a}_1 + \boldsymbol{a}_2) \\
&= \boldsymbol{a}_1^T \boldsymbol{a}_1 + \boldsymbol{a}_1^T \boldsymbol{a}_2 + \boldsymbol{a}_2^T \boldsymbol{a}_1 + \boldsymbol{a}_2^T \boldsymbol{a}_2 \\
&= \boldsymbol{a}_1^T \boldsymbol{a}_1 + 2\boldsymbol{a}_1^T \boldsymbol{a}_2 + \boldsymbol{a}_2^T \boldsymbol{a}_2
\end{aligned}
$$

4. 若A 表示实矩阵，$\boldsymbol{x}, \boldsymbol{a}$ 表示未知数向量和实向量，

则

$$\frac{\partial(\boldsymbol{a}^T\boldsymbol{x})}{\partial\boldsymbol{x}} = \frac{\partial(\boldsymbol{x}^T\boldsymbol{a})}{\partial\boldsymbol{x}} = \boldsymbol{a}$$

$$\frac{\partial(\boldsymbol{x}^T A\boldsymbol{x})}{\partial\boldsymbol{x}} = (A + A^T)\boldsymbol{x}$$

解析法

为了方便，此处我们使：

$$\hat{\boldsymbol{w}} = [\boldsymbol{w}^T, b]^T = \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} \boldsymbol{x}_1^T & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_n^T & 1 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

则 $\boldsymbol{y} = \mathbf{X}\hat{\boldsymbol{w}} + \boldsymbol{\epsilon}$，$\mathbf{X}\hat{\boldsymbol{w}}$ 代表模型在全体样本输入时对应的输出。

则代价 $J(\hat{\boldsymbol{w}}) = \|\boldsymbol{\epsilon}\|_2^2 = \left\|\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}\right\|_2^2$

解析法

$$J(\hat{\boldsymbol{w}}) = \|\boldsymbol{\epsilon}\|_2^2 = \left\| \boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}} \right\|_2^2$$

$$= (\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}})^T(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}) = \boldsymbol{y}^T\boldsymbol{y} - 2\hat{\boldsymbol{w}}^T\mathbf{X}^T\boldsymbol{y} + \hat{\boldsymbol{w}}^T\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{w}}$$

令 $\nabla J(\hat{\boldsymbol{w}}) = -2\mathbf{X}^T\boldsymbol{y} + 2\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{w}} = \mathbf{0}$ 则 $\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{w}} = \mathbf{X}^T\boldsymbol{y}$

若 $\mathbf{X}^T\mathbf{X}$ 满秩，则有 $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}$

即 $\hat{\boldsymbol{w}}* = \begin{bmatrix} \boldsymbol{w}* \\ b* \end{bmatrix} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}$

小练习

给定训练样本：

$$\boldsymbol{x}_1 = (1,0)^T, y_1 = \frac{1}{2},$$

$$\boldsymbol{x}_2 = (0,1)^T, y_2 = 1,$$

$$\boldsymbol{x}_3 = (1,1)^T, y_3 = -1$$

若使用线性回归模型拟合上述样本，试用正规方程求解线性回归的参数。

答案

令 $\mathbf{X} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$， $\boldsymbol{y} = \begin{bmatrix} \frac{1}{2} \\ 1 \\ -1 \end{bmatrix}$

则 $\mathbf{X}^T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \\ 2 & 2 & 3 \end{bmatrix}$

因为 $\mathrm{Rank}(\mathbf{X}^T\mathbf{X}) = 3$ ，所以 $\mathbf{X}^T\mathbf{X}$ 可逆，利用高斯约旦法求得 $(\mathbf{X}^T\mathbf{X})^{-1} = \begin{bmatrix} 2 & 1 & -2 \\ 1 & 2 & -2 \\ -2 & -2 & 3 \end{bmatrix}$

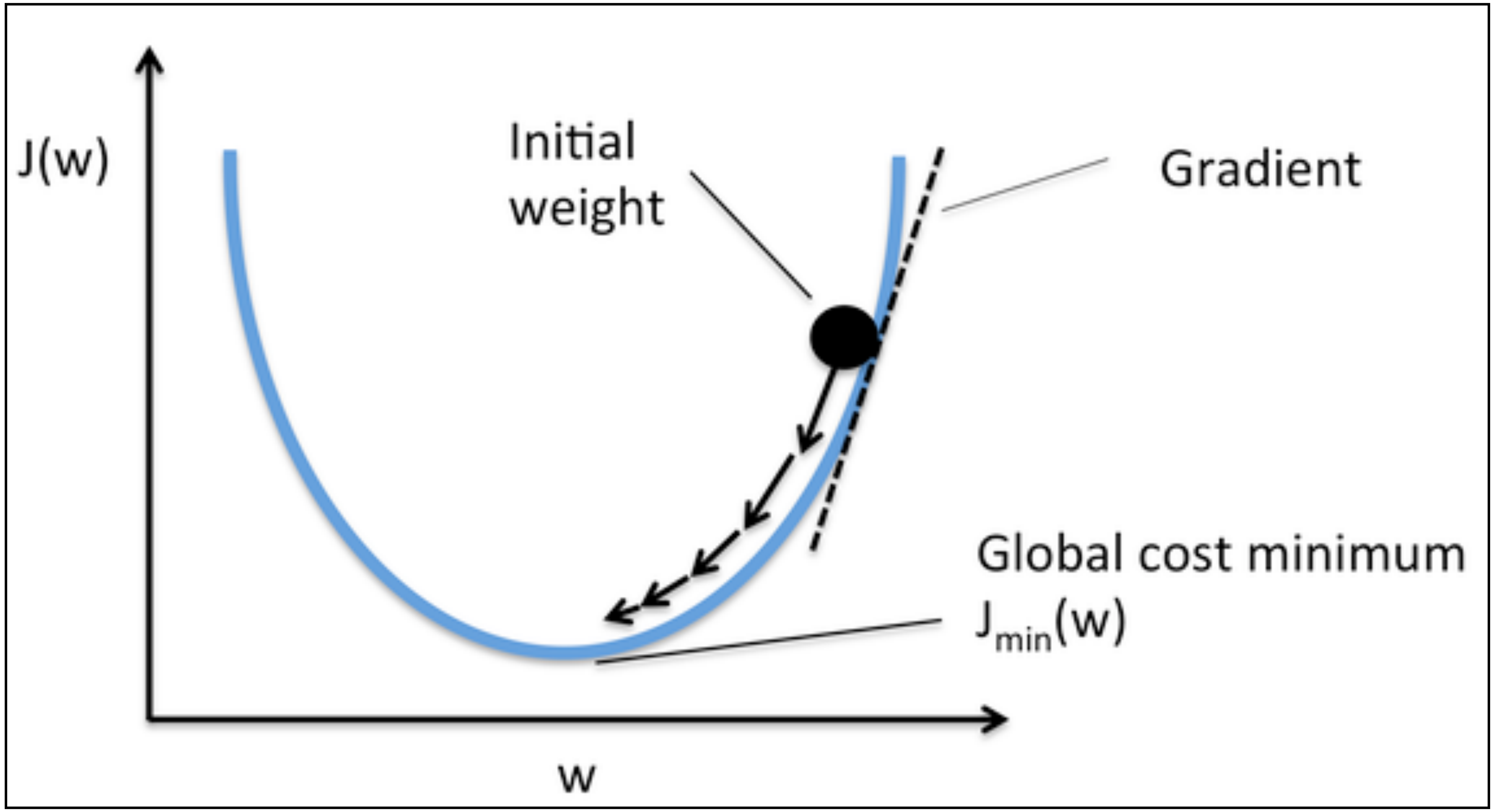
所以 $\hat{\boldsymbol{w}}* = \begin{bmatrix} \boldsymbol{w}* \\ b* \end{bmatrix} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y} = \begin{bmatrix} -2 \\ -1.5 \\ 2.5 \end{bmatrix}$

梯度下降法

梯度下降法（**Gradient Descent**），也称为**最速下降法**（**Steepest Descent**），是一种一阶**最优化算法**。通过对函数某一点对应的**梯度的反方向**以规定**步长**（**Step size**）距离进行**迭代搜索**以找到函数的局部极小值。

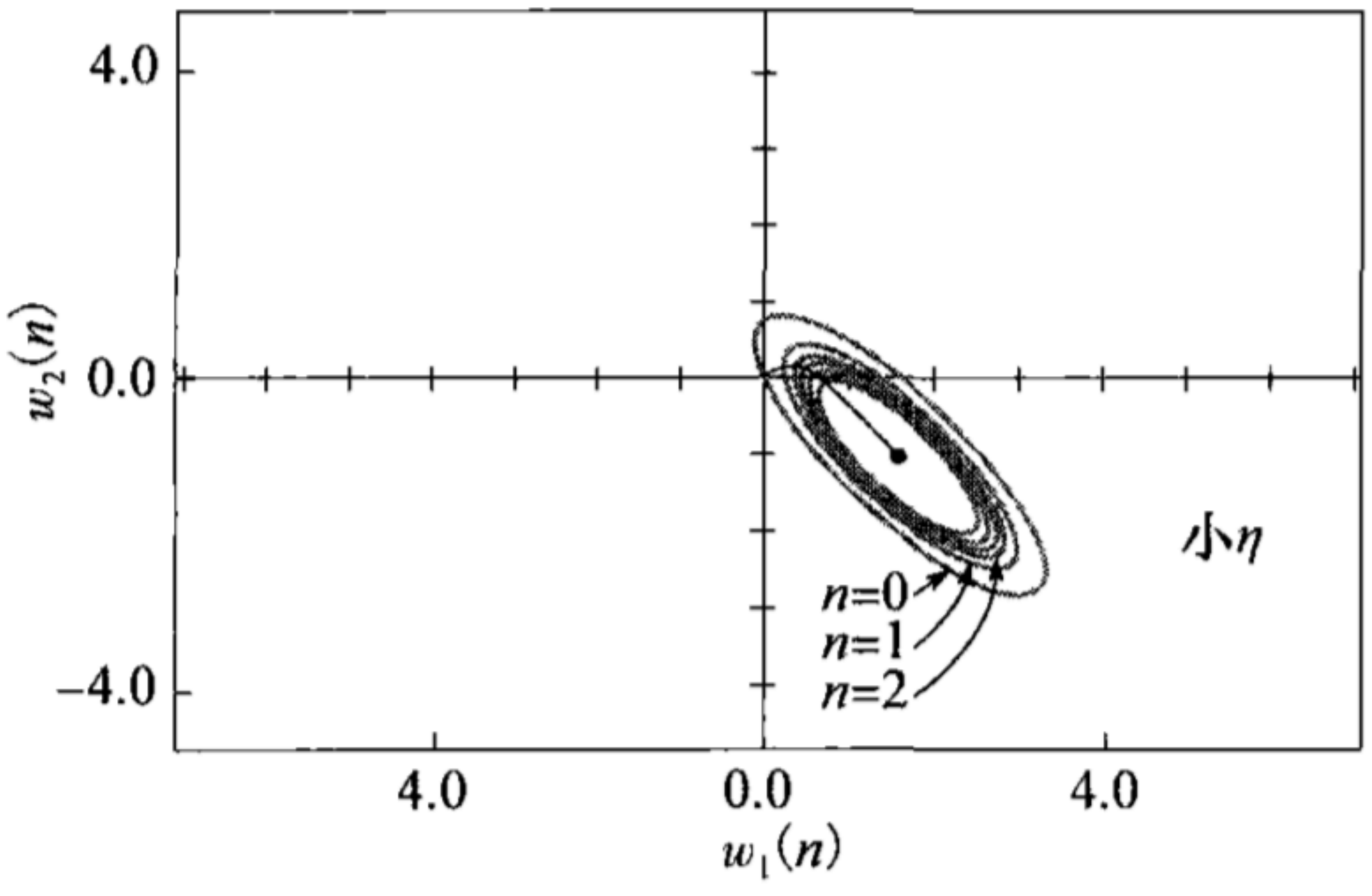
迭代规则：$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \alpha \nabla J(\boldsymbol{w})$，其中 $\boldsymbol{w}$ 表示模型参数，$J(\boldsymbol{w})$ 表示代价，t 表示当前迭代次数，α 表示步长（也叫学习率，梯度下降法超参数）。

梯度下降法示意图

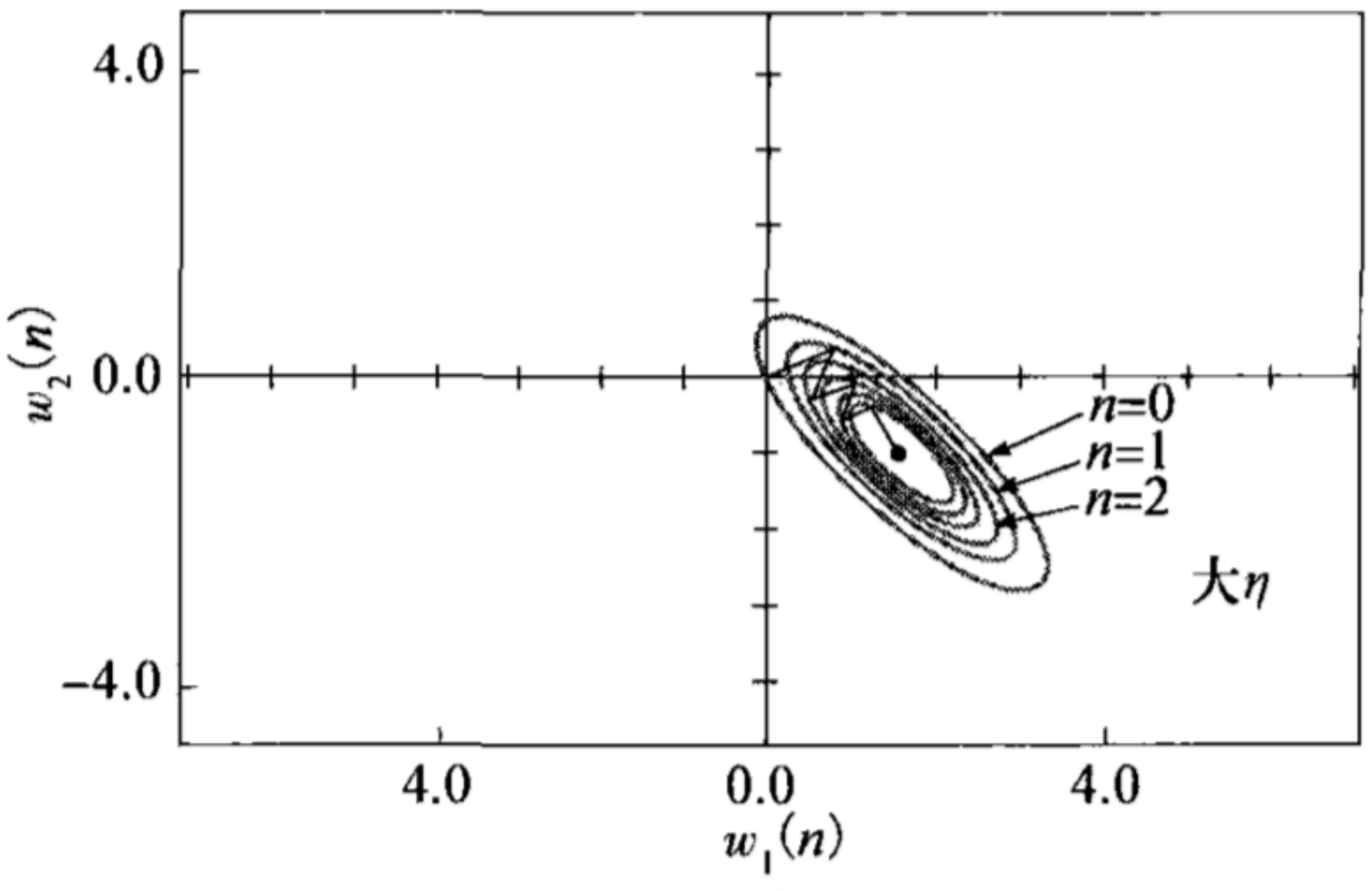


1. 梯度是一阶偏导数组成的向量
2. 梯度的方向就是增长最快的方向（最陡的方向）
3. 梯度下降的方向就是梯度的反方向
4. 梯度下降法就是逐步迭代更新参数的过程，每次迭代使用最新梯度值来更新参数

梯度下降法中的学习率



a）



b）

学习率较小时，下降速度相对较慢，学习率较大时，下降速度相对较快，但过大的学习率可能导致模型不收敛或者出现剧烈波动。

梯度下降法中的收敛准则

- 不能证明梯度下降法是收敛的，并且没有明确定义的算法停止准则。
- 通常使用如下方法对是否收敛进行判断：
  - 当梯度向量的欧几里得范数达到一个充分小的阈值时。
  - 当迭代的每一个回合的均方误差变化的绝对速率足够小时。
- 当目标函数是凸函数时，梯度下降法的解是全局最优解。一般情况下，其解不保证是全局最优解。其下降速度也不保证是最快的。

梯度下降法算法描述

输入：目标函数 $J(\boldsymbol{w})$，梯度函数 $g(\boldsymbol{w}) = \nabla J(\boldsymbol{w})$，计算精度 ϵ

输出：$J(\boldsymbol{w})$ 的极小值点 $\boldsymbol{w}*$

① 取初始值 $\boldsymbol{w}^{(0)} \in R^d$，置 $t = 0$

② 计算 $J(\boldsymbol{w}^{(t)})$

③ 计算梯度 $g_t = g(\boldsymbol{w}^{(t)})$，当 $\|g_t\| < \epsilon$ 时，停止迭代，令 $\boldsymbol{w}* = \boldsymbol{w}^{(t)}$，否则，令 $p_t = -g(\boldsymbol{w}^{(t)})$，求学习率 λ_t，使 $J(\boldsymbol{w}^{(t)} + \lambda_t p_t) = \min\limits_{\lambda \geqslant 0} J(\boldsymbol{w}^{(t)} + \lambda p_t)$。（实际中因搜索学习率的最佳值比较困难，往往使用固定值作为学习率或者使用学习率衰减等策略确定学习率）

④ 置 $\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \lambda_t p_t$，计算 $J(\boldsymbol{w}^{(t+1)})$，当 $\|J(\boldsymbol{w}^{(t+1)}) - J(\boldsymbol{w}^{(t)})\| < \epsilon$ 或 $\|\boldsymbol{w}^{(t+1)} - \boldsymbol{w}^{(t)}\| < \epsilon$ 时，停止迭代，令 $\boldsymbol{w}* = \boldsymbol{w}^{(t+1)}$

⑤ 否则，置 $t = t + 1$，转 ③ 。

批量与小批量算法

- **批量梯度下降法（Gradient Descent, GD）**：使用全部训练样本估计梯度进行训练。计算量大，但更多样本来估计梯度的回报是小于线性，且大多数样本对梯度做出了非常相似的贡献，所以一般不使用批量算法。

- **小批量梯度下降法（Mini Batch Gradient Descent，MBGD）**：使用部分训练样本估计梯度进行训练。通常使用此法。

- **随机梯度下降法（Stochastic Gradient Descent，SGD）**：每次从固定训练集中抽取一个训练样本估计梯度进行训练。

- **在线梯度下降法（online Gradient Descent）**：每次从连续产生样本的数据流中抽取一个样本进行训练。

- 通常的，把小批量梯度下降法也习惯性的叫做随机梯度下降法。

使用MBGD求解线性回归模型参数

训练集$\{(\boldsymbol{x}_i, y_i), i=1,2,...,n\}, \mathbf{x} \in R^d, y \in R$，随机打乱样本顺序。设小批量大小为 m，使用固定学习率 α，设定一种停止迭代规则。

① 从打乱训练集中顺序选择 m 个样本，组成一个小批次样本

② 使用抽取出的小批次样本计算模型代价 $J(\boldsymbol{w}, b) = \dfrac{1}{2m}\sum_{i=1}^{m}[y_i - (\boldsymbol{w}^T\boldsymbol{x}_i + b)]^2$

③ 计算 m 个样本的平均梯度 $\bar{\boldsymbol{g}}_{\boldsymbol{w}} = \dfrac{1}{m}\sum_{i=1}^{m} - \boldsymbol{x}_i[y_i - (\boldsymbol{w}^T\boldsymbol{x}_i + b)], \bar{g}_b = \dfrac{1}{m}\sum_{i=1}^{m} - [y_i - (\boldsymbol{w}^T\boldsymbol{x}_i + b)]$

④ 更新模型参数 $\boldsymbol{w} = \boldsymbol{w} - \alpha\bar{\boldsymbol{g}}_{\boldsymbol{w}}$，$b = b - \alpha\bar{g}_b$,判断是否满足停止迭代规则，如满足，则停止迭代，将此时参数作为最优参数，如不满足，则回到 ① 继续迭代。

最小二乘线性回归的优缺点

- 模型简单，训练方便、快捷，且具有很好的可解释性。
- 当描述样本的特征之间存在明显的相关性时，会导致某些预测变量以及与其相关程度强的预测变量，具有较大的系数估计值，但因符号相反而相互抵消。
- 最小二乘估计结果不稳定：数据集较小的变化，甚至会导致估计结果较大的差异。

3. 正则化

正则化

- **参数范数惩罚**（**参数范数正则化**），通过对目标代价函数 J 添加一个参数范数惩罚，限制模型的学习能力。正则化后的总体代价函数为：$\tilde{J}(\boldsymbol{w}, b) = J(\boldsymbol{w}, b) + \lambda\Omega(\boldsymbol{w})$，其中 $\Omega(\boldsymbol{w})$ 表示**惩罚项**，$\lambda \in [0,\infty]$ 表示惩罚项与标准代价函数 J 的相对贡献。

- L_1 正则化，也被称为**套索回归**（**LASSO Regression**）。通过在代价函数中**引入参数的一范数惩罚**来实现。即 $\tilde{J}(\boldsymbol{w}, b) = J(\boldsymbol{w}, b) + \lambda\|\boldsymbol{w}\|_1$。

- L_2 正则化，也被称为**岭回归**（**Ridge Regression**）。通过在代价函数中**引入参数的二范数惩罚**来实现。即 $\tilde{J}(\boldsymbol{w}, b) = J(\boldsymbol{w}, b) + \lambda\|\boldsymbol{w}\|_2^2$

套索回归

- 代价函数为 $\tilde{J}(\boldsymbol{w}) = J(\boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_1$ （注意：此处将 b 合并入 $\boldsymbol{w}$）
- 解析解为 $\boldsymbol{w}^* = (\boldsymbol{x}^T\boldsymbol{x})^{-1}[\boldsymbol{x}^T y - \frac{\lambda}{2}\mathbf{1}(\boldsymbol{w})]$ ，其中 $\mathbf{1}(w_i) = \begin{cases} 1, & w_i \geqslant 0 \\ -1, & w_i < 0 \end{cases}$
- 当调节参数 λ 足够大时，L_1 惩罚项具有将某些参数估计值强制设定为0的作用，即得到 $\boldsymbol{w}$ 的稀疏向量，因此套索回归具有变量选择的作用。
- 主要用于高维特征空间的模型选择。

岭回归

- 代价函数为 $\tilde{J}(\boldsymbol{w}) = J(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_2^2$ （注意：此处将 b 合并入 $\boldsymbol{w}$）
- 解析解为 $\boldsymbol{w}* = (\boldsymbol{x}^T\boldsymbol{x} + \lambda I)^{-1}\boldsymbol{x}^T\boldsymbol{y}$
- 岭回归的最终模型包含全部的p个变量，尽管惩罚项的存在可以使系数向零的方向进行缩减，但是不会把任何一个变量的系数确切的压缩为0。
- 当变量的数目p非常大时，不便于模型解释，不适合特征维数过高情况。

注意事项

岭回归、套索回归训练与预测之前，需要对样本输入、输出标准化。甚至需要对训练样本中的预测变量进行尺度规范化。

一般的使用 $z - score$ 标准化：

$$\tilde{x_{ij}} = \frac{x_{ij} - \bar{x}_j}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}.$$

4. 小练习

小练习（一）

1. 尝试独立推导正规方程（最小二乘线性回归的解析解法）。
2. 思考：参数范数惩罚中的 λ 值如何确定? λ 取值范围是多少? 如何通过 λ 提升惩罚力度?
3. 根据小批量梯度下降法原理，分别写出在套索回归、岭回归中的算法流程。

答案

1. 略

2. λ 是超参数， λ 可使用交叉验证法确定，也可使用验证集验证确定；$\lambda \in [0,\infty]$；增大 λ 值。

3. 主流程与最小二乘线性回归相同，参数更新时，梯度稍有不同。套索回归对 $\boldsymbol{w}$ 梯度计算公式 $\bar{\boldsymbol{g}} = \sum_{i=1}^{m} -\boldsymbol{x}_i \cdot [y_i - (\boldsymbol{w}^T\boldsymbol{x}_i + b)] + 2\lambda\boldsymbol{w}$ ，岭回归对 $\boldsymbol{w}$ 梯度计算公式 $\bar{\boldsymbol{g}} = \sum_{i=1}^{m} -\boldsymbol{x}_i \cdot [y_i - (\boldsymbol{w}^T\boldsymbol{x}_i + b)] + \lambda\mathbf{1}(\boldsymbol{w})$。

5. 小结

- 线性回归模型为 $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$，代价函数为 $J(\boldsymbol{w}, b) = \sum_{i=1}^{n} [y_i - f(\boldsymbol{x}_i)]^2$ ，通过最小化代价函数 J 求得参数 $[\boldsymbol{w}^*, b^*]$
- 线性回归的正规方程解为 $\hat{\boldsymbol{w}}^* = [\boldsymbol{w}^{*T}, b^*]^T = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}$
- 梯度下降法是利用梯度的反方向迭代求解模型参数
- 批量、小批量、随机梯度下降法分别使用全部、部分、单个样本估计梯度
- 在线梯度下降法使用单个在线生成的样本估计梯度
- 通过引入参数范数正则化可以降低模型复杂度，一范数正则化也被称为套索回归，可用来变量选择，二范数正则化也被称为岭回归。

THANKS