如何算出模型参数的估计值

梯度下降法

小胖

目录

算法思路

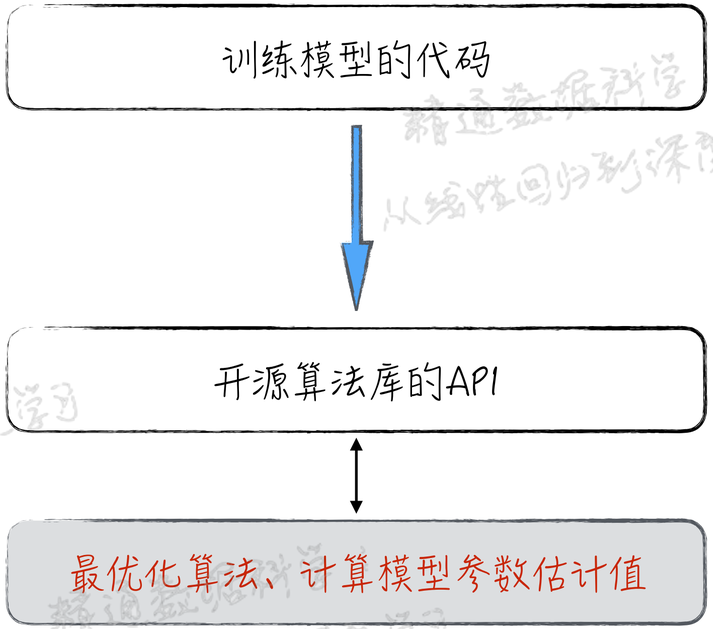
最优化算法

搭建模型的步骤：

- 从场景入手分析数据
- 通过数学变换、套用模型架构解决问题
- 借助开源算法库的API，实现模型

仅仅会调用开源算法库API是远远不够的

- 还需要了解算法库内部是如何算出模型参数的估计值

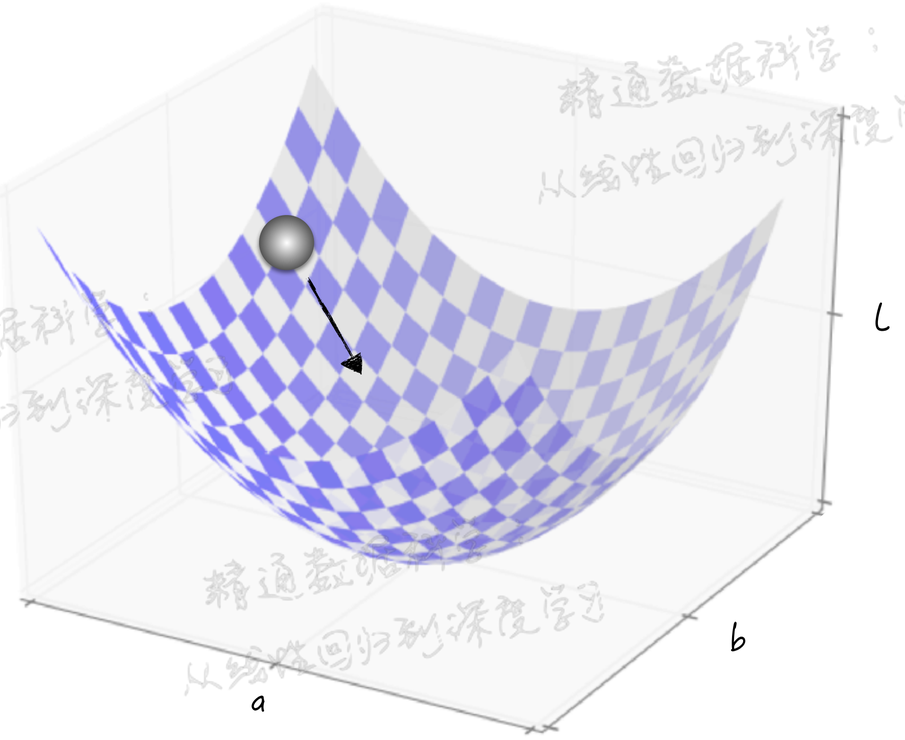


算法思路

模拟滚动

算法目的：找到函数的最小值以及相应的参数值

梯度下降法：通过模拟小球滚动的方法来得到函数的最小值点



目录

泰勒级数

损失函数

损失函数等于每点损失之和

损失函数对于模型参数都是可微的

泰勒展开式

线性回归

$$L = \sum_{i=1}^{n} (y_i - X_i\beta)^2 = \sum_{i=1} L_i$$

逻辑回归

$$L = -\sum_{i=1}^{n} y_i \ln h(X_i) + (1 - y_i)\ln[1 - h(X_i)] = \sum_{i=1} L_i$$

$$L(\beta_1, \ldots, \beta_n) \approx L(a_1, \ldots, a_n) + \sum_j \frac{\partial L}{\partial \beta_j}(\beta_j - a_j)$$

$$\frac{\partial L}{\partial \beta_j} = \sum_i \frac{\partial L_i}{\partial \beta_j}$$

泰勒级数

迭代公式

泰勒展开式

$$L(\beta_1,\ldots,\beta_n) \approx L(a_1,\ldots,a_n) + \sum_j \frac{\partial L}{\partial \beta_j}(\beta_j - a_j)$$

假设损失函数：

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

随机选取起点（a0, b0）：

$$\Delta L = L(a_1, b_1) - L(a_0, b_0)$$

$$\Delta L \approx \frac{\partial L}{\partial a}\Delta a + \frac{\partial L}{\partial b}\Delta b$$

如果令：

$$(\Delta a, \Delta b) = -\gamma(\frac{\partial L}{\partial a}, \frac{\partial L}{\partial b})$$

$$\Delta L \approx -\gamma[(\frac{\partial L}{\partial a})^2 + (\frac{\partial L}{\partial b})^2] \leq 0$$

得到迭代公式：

$$a_{k+1} = a_k - \gamma\frac{\partial L}{\partial a}, b_{k+1} = b_k - \gamma\frac{\partial L}{\partial b}$$

泰勒级数

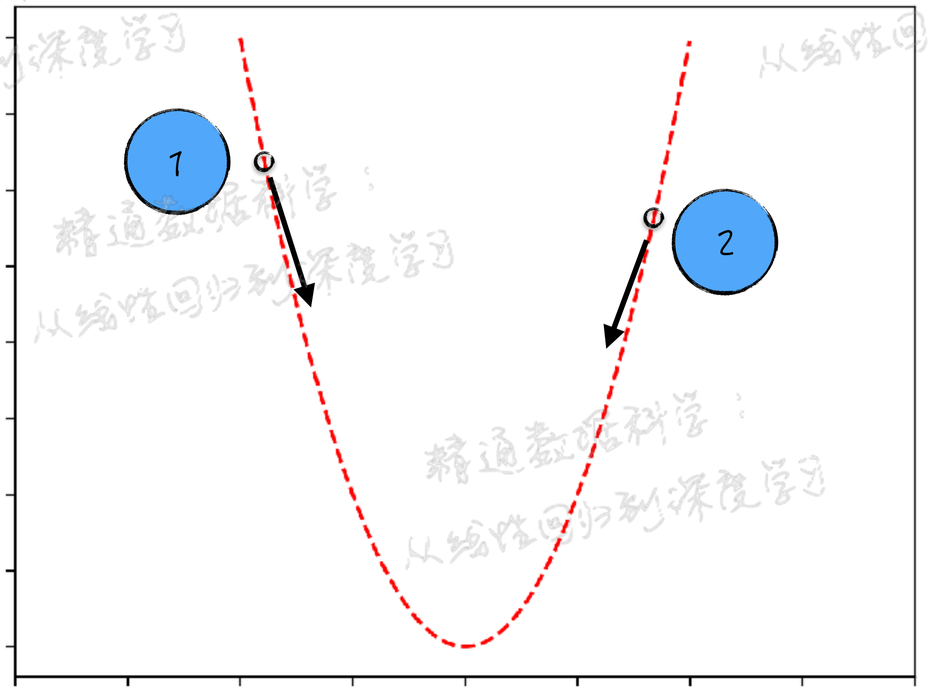
梯度下降法

得到迭代公式：$a_{k+1} = a_k - \gamma \frac{\partial L}{\partial a}, b_{k+1} = b_k - \gamma \frac{\partial L}{\partial b}$

γ 是学习速率，决定移动步伐的大小

$(\frac{\partial L}{\partial a}, \frac{\partial L}{\partial b})$ 是函数的梯度，决定移动的方向

数学上可以证明，沿着梯度，是函数值**下降最快**的方向

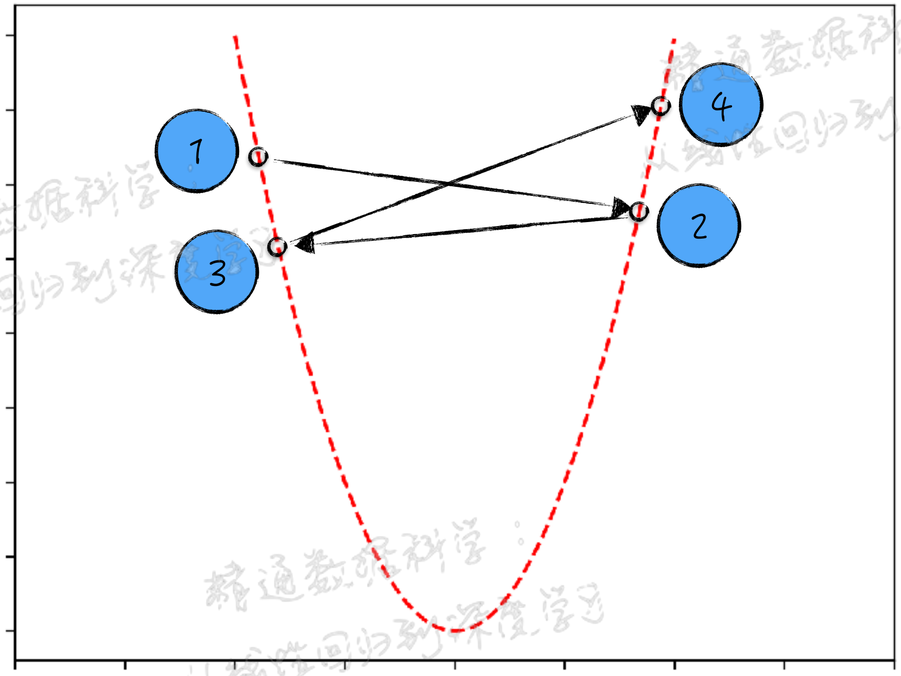


目录

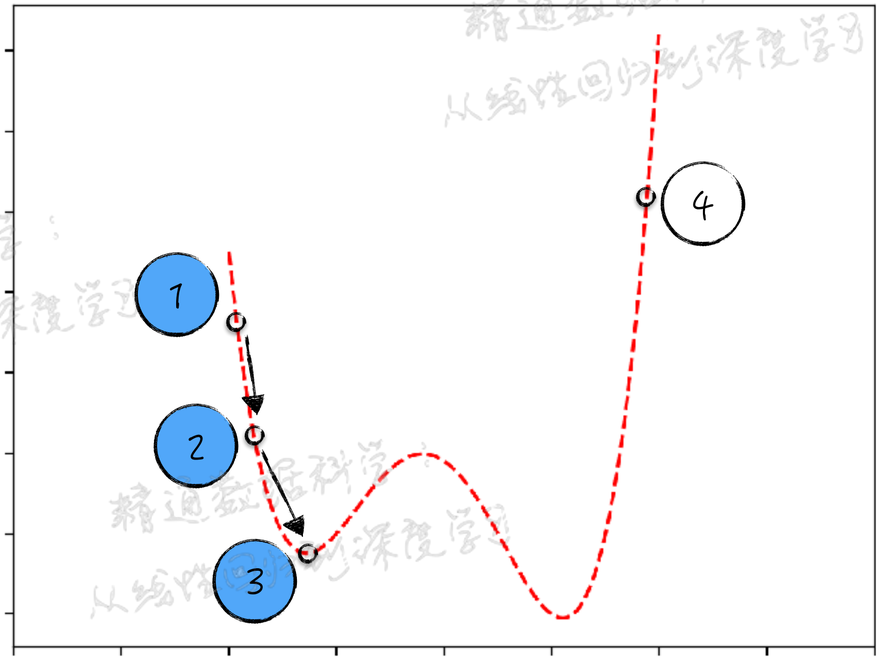
注意事项

学习速率、全局最优

学习速率（learning rate）过大



局部最低与全局最低

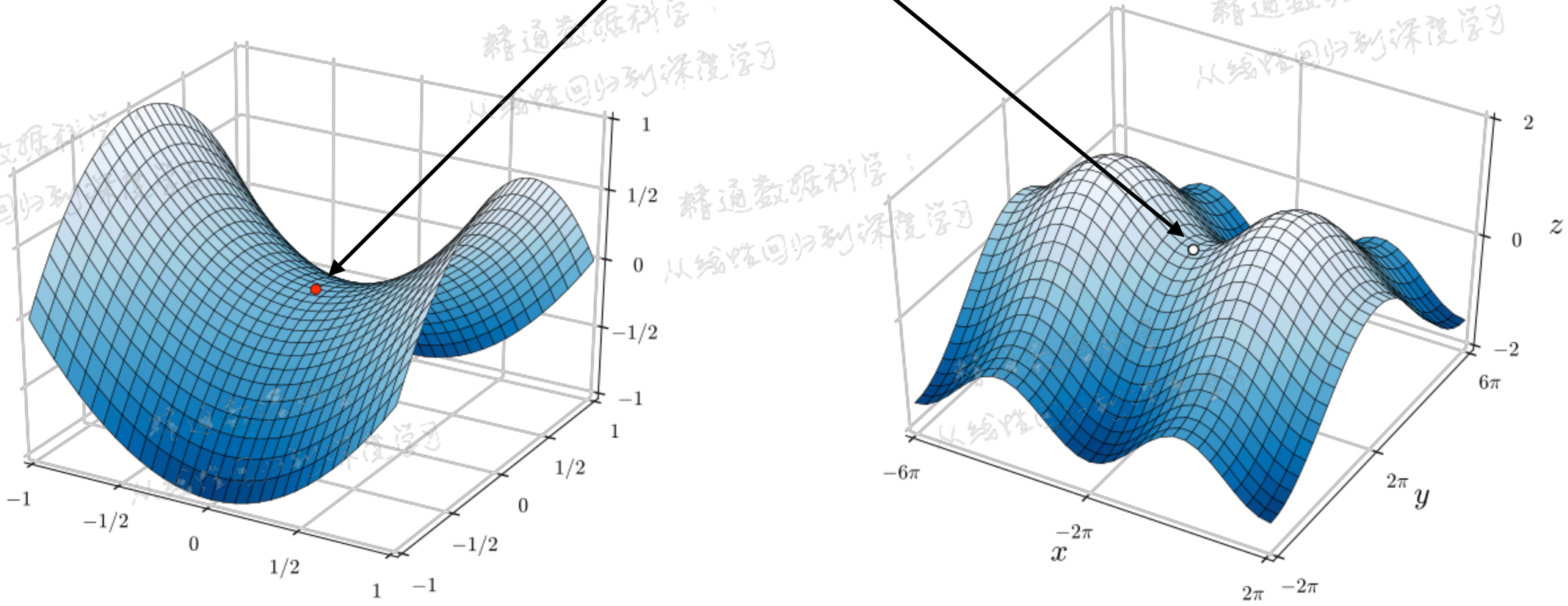


注意事项

鞍点

鞍点（saddle point）

梯度等于0，但并不是极小值点



THANK YOU