

# 第一章：优化迭代方法统一论

## ——优化论五部曲

Jason 博士

网易微专业 × 稀牛学院

人工智能数学基础微专业

线性回归建模

无约束优化梯度分析法

无约束优化迭代法

线性回归求解

网易微专业 犀牛学院 人工智能数学基础 Jason博士

# 线性回归 (1/2)

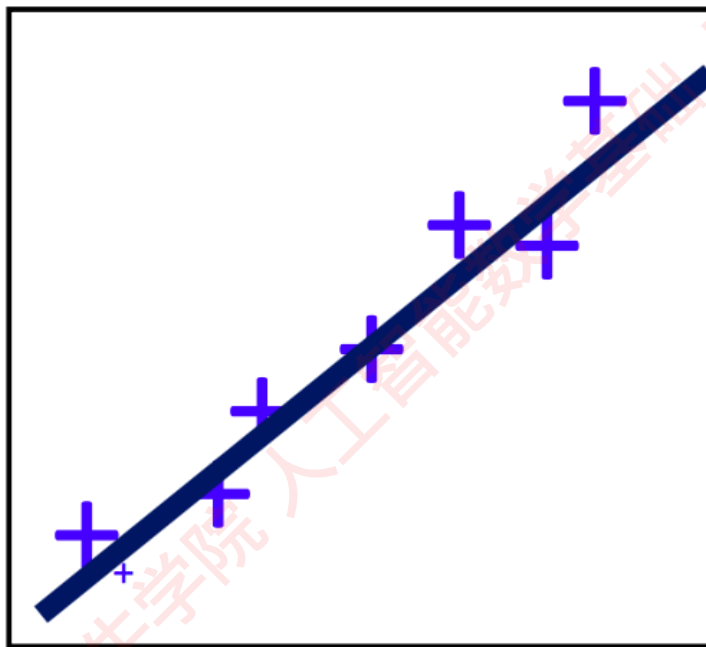
## - 训练, 预测

	1 name	2 sex	3 age	4 wgt	5 smoke	6 sys	7 dia	8 trial1	9 trial
1 YPL-320	'SMITH'	'm'	38	176	1	124	93	18	
2 GLI-532	'JOHNSON'	'm'	43	163	0	109	77	11	
3 PNI-258	'WILLIAMS'	'f'	38	131	0	125	83	-99	
4 MIJ-579	'JONES'	'f'	40	133	0	117	75	6	
5 XLK-030	'BROWN'	'f'	49	119	0	122	80	14	
6 TFP-518	'DAVIS'	'f'	46	142	0	121	70	19	
7 LPD-746	'MILLER'	'f'	33	142	1	130	88	0	
8 ATA-945	'WILSON'	'm'	40	180	0	115	82	-99	
9 VNL-702	'MOORE'	'm'	28	183	0	115	78	2	
10 LQW-768	'TAYLOR'	'f'	31	132	0	118	86	11	
11 QFY-472	'ANDERS...	'f'	45	128	0	114	77	8	
12 UJG-627	'THOMAS'	'f'	42	137	0	115	68	4	
13 XUE-826	'JACKSON'	'm'	25	174	0	127	74	-99	
14 TRW-072	'WHITE'	'm'	39	202	1	130	95	8	

-  $\{(x^{(i)}, y^{(i)})\}$  一个训练样本,  $\{(x^{(i)}, y^{(i)}) ; i = 1, \dots, N\}$  训练样本集

-  $\{(x_1^{(i)}, x_2^{(i)}, y^{(i)})\} \longrightarrow \{(\mathbf{x}^{(i)}, y^{(i)})\}, \mathbf{x}^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \end{bmatrix}$

## 线性回归 (2/2)



- 试图学习:  $f(x) = wx + b$  使得  $f(x^{(i)}) \approx y^{(i)}$
- 试图学习:  $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$  使得  $f(\mathbf{x}^{(i)}) \approx y^{(i)}$
- 核心在于怎么学?

线性回归建模

无约束优化梯度分析法

无约束优化迭代法

线性回归求解

网易微专业 犀牛学院 人工智能数学基础 Jason博士

# 无约束优化问题

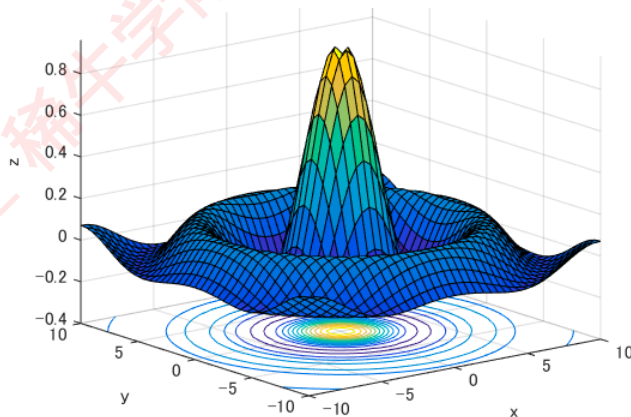
- 自变量为标量的函数  $f: \mathbb{R} \rightarrow \mathbb{R}$

$$\min f(x) \quad x \in \mathbb{R}$$

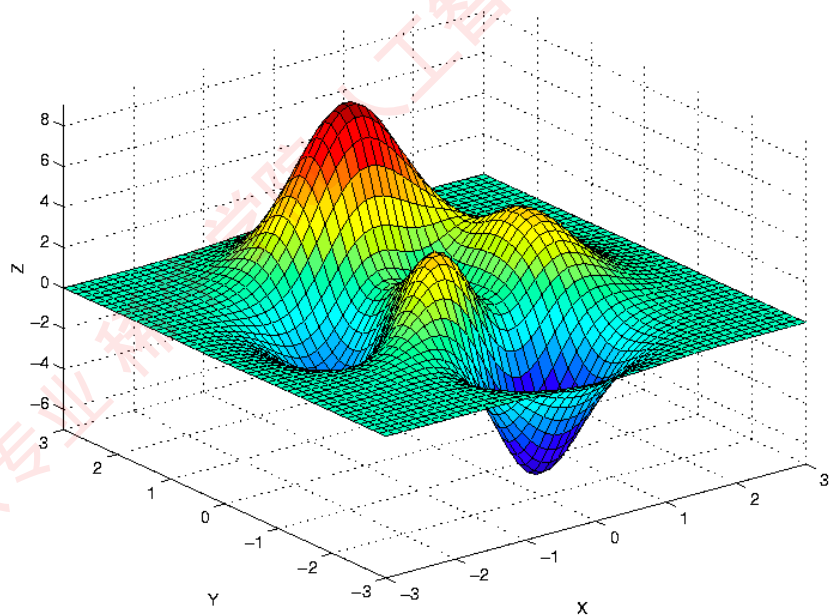
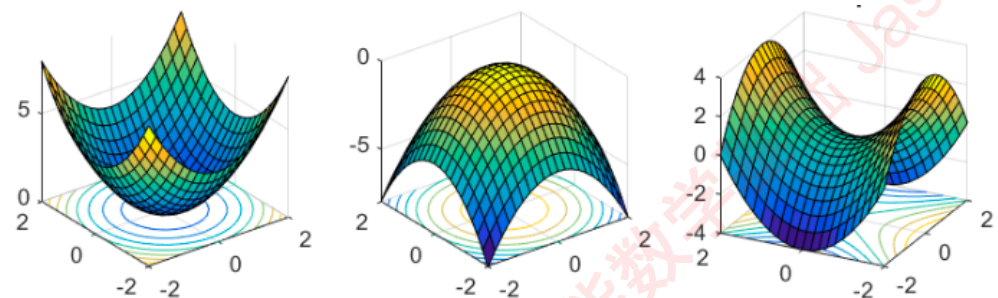
- 自变量为向量的函数  $f: \mathbb{R}^n \rightarrow \mathbb{R}$

$$\min f(\mathbf{x}) \quad \mathbf{x} \in \mathbb{R}^n$$

- 熟悉 Contour



# 优化问题可能的极值点情况



# 梯度和 Hessian 矩阵

- 一阶导数和梯度 (gradient vector)

$$f'(x); \quad \mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix}$$

- 二阶导数和 Hessian 矩阵

$$f''(x); \quad \mathbf{H}(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} & \cdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & & & \\ & & \ddots & & \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} & \end{bmatrix} = \nabla (\nabla f(\mathbf{x}))^T$$



# 二次型

- 给定矩阵  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , 函数

$$\mathbf{x}^T \mathbf{A} \mathbf{x} = \sum_{i=1}^n x_i (\mathbf{A} \mathbf{x})_i = \sum_{i=1}^n x_i \left( \sum_{j=1}^n a_{ij} x_j \right) = \sum_{i=1}^n \sum_{j=1}^n x_i x_j a_{ij} \quad (1)$$

被称为二次型. 尝试将  $f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2$  写成二次型的形式

- 给定对称矩阵  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , 如果对于所有  $\mathbf{x} \in \mathbb{R}^n$ , 有  $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$ , 则为半正定矩阵 (positive semidefinite), 此时特征值  $\lambda(\mathbf{A}) \geq 0$ .
- 如果对于所有  $\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0}$ , 有  $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$ , 则为正定矩阵 (positive definite).
- 负定矩阵, 不定矩阵 (indefinite)

# 具体计算

- 向量  $\mathbf{a}$  和  $\mathbf{x}$  无关, 则  $\nabla (\mathbf{a}^T \mathbf{x}) = \mathbf{a}$ ,  $\nabla^2 (\mathbf{a}^T \mathbf{x}) = \mathbf{0}$
- 对称矩阵  $\mathbf{A}$  与  $\mathbf{x}$  无关, 则  $\nabla (\mathbf{x}^T \mathbf{A} \mathbf{x}) = 2\mathbf{A}\mathbf{x}$ ,  $\nabla^2 (\mathbf{x}^T \mathbf{A} \mathbf{x}) = 2\mathbf{A}$
- 最小二乘

$$\begin{aligned} f(\mathbf{x}) &= \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \\ &= \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} - 2\mathbf{b}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{b} \end{aligned}$$

$$\nabla f(\mathbf{x}) = 2\mathbf{A}^T \mathbf{A} \mathbf{x} - 2\mathbf{A}^T \mathbf{b}$$

- $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x} + 2\mathbf{b}^T \mathbf{x} + c$

# 泰勒级数

## 泰勒级数展开 (标量和向量)

- 输入为标量的泰勒级数展开

$$f(x_k + \delta) \approx f(x_k) + f'(x_k) \delta + \frac{1}{2} f''(x_k) \delta^2 + \cdots + \frac{1}{k!} f^{(k)}(x_k) \delta^k + \cdots \quad (2)$$

- 输入为向量的泰勒级数展开

$$f(\mathbf{x}_k + \boldsymbol{\delta}) \approx f(\mathbf{x}_k) + \mathbf{g}^T(\mathbf{x}_k) \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^T \mathbf{H}(\mathbf{x}_k) \boldsymbol{\delta} \quad (3)$$

# 泰勒级数和极值

## 标量情况

- 输入为标量的泰勒级数展开

$$f(x_k + \delta) \approx f(x_k) + f'(x_k) \delta + \frac{1}{2} f''(x_k) \delta^2$$

- 严格局部极小点指:  $f(x_k + \delta) > f(x_k)$
- 称满足  $f'(x_k) = 0$  的点为平稳点 (候选点).
- 函数在  $x_k$  有严格局部极小值条件为  $f'(x_k) = 0$  且  $f''(x_k) > 0$

# 泰勒级数与极值

## 向量情况

- 输入为向量的泰勒级数展开

$$f(\mathbf{x}_k + \boldsymbol{\delta}) \approx f(\mathbf{x}_k) + \mathbf{g}^T(\mathbf{x}_k) \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^T \mathbf{H}(\mathbf{x}_k) \boldsymbol{\delta}$$

- 称满足  $\mathbf{g}(\mathbf{x}_k) = 0$  的点为平稳点 (候选点), 此时如果有  
     $\mathbf{H}(\mathbf{x}_k) \succ 0$ ,  $x_k$  为一严格局部极小点 (反之, 严格局部最大点)  
    如果  $\mathbf{H}(\mathbf{x}_k)$  不定矩阵, 是一个鞍点 (saddle point)

# 梯度为 0 求解的局限性

计算  $f(x) = x^4 + \sin(x^2) - \ln(x)e^x + 7$  的导数

$$\begin{aligned} f'(x) &= 4x^{(4-1)} + \frac{d(x^2)}{dx} \cos(x^2) - \frac{d(\ln x)}{dx} e^x - \ln(x) \frac{d(e^x)}{dx} + 0 \\ &= 4x^3 + 2x \cos(x^2) - \frac{1}{x} e^x - \ln(x) e^x \end{aligned}$$

思考求  $f'(x) = 0$  ?

线性回归建模

无约束优化梯度分析法

无约束优化迭代法

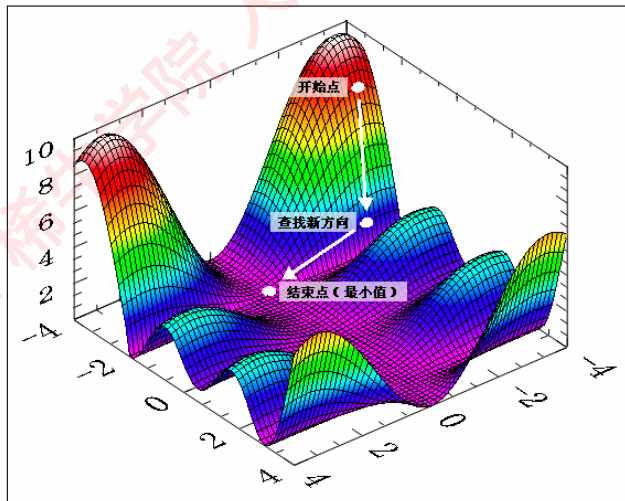
线性回归求解

网易微专业 犀牛学院 人工智能数学基础 Jason博士

# 无约束优化迭代法

## 迭代法的基本结构 (最小化 $f(\mathbf{x})$ )

- 1 选择一个初始点, 设置一个 convergence tolerance  $\epsilon$ , 计数  $k = 0$
- 2 决定搜索方向  $\mathbf{d}_k$ , 使得函数下降. (核心)
- 3 决定步长  $\alpha_k$  使得  $f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)$  对于  $\alpha_k \geq 0$  最小化, 构建  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
- 4 如果  $\|\mathbf{d}_k\| < \epsilon$ , 则停止输出解  $\mathbf{x}_{k+1}$ ; 否则继续重复迭代.





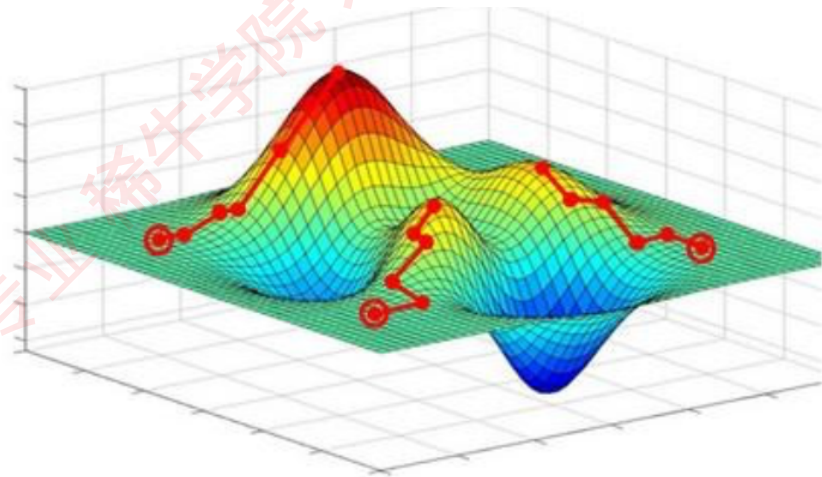
# 梯度下降法

## 梯度下降法

- $\mathbf{d}_k = -\mathbf{g}(\mathbf{x}_k)$ , 思考为什么这么取?

$$f(\mathbf{x}_k + \mathbf{d}_k) \approx f(\mathbf{x}_k) + \mathbf{g}^T(\mathbf{x}_k) \mathbf{d}_k$$

- 需要  $f(\mathbf{x}_k + \mathbf{d}_k) \downarrow$ , 则  $f(\mathbf{x}_k)$  加个负数
- 回忆两个向量的内积,  $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$



# 牛顿法

## 牛顿法

- 方向选取  $\mathbf{d}_k = -\mathbf{H}^{-1}(\mathbf{x}_k) \mathbf{g}(\mathbf{x}_k)$

- 方向选取依据

$$f(\mathbf{x}_k + \mathbf{d}_k) = f(\mathbf{x}_k) + \mathbf{g}^T(\mathbf{x}_k) \mathbf{d}_k + \frac{1}{2} \mathbf{d}_k^T \mathbf{H}(\mathbf{x}_k) \mathbf{d}_k$$

令  $\frac{\partial f(\mathbf{x}_k + \mathbf{d}_k)}{\partial \mathbf{d}_k} = \mathbf{0} \Rightarrow \mathbf{g}(\mathbf{x}_k) + \mathbf{H}(\mathbf{x}_k) \mathbf{d}_k = \mathbf{0}$

- 若 Hessian 矩阵正定, 则有  $\mathbf{d}_k = -\mathbf{H}^{-1}(\mathbf{x}_k) \mathbf{g}(\mathbf{x}_k)$

- 强制要求 Hessian 矩阵正定 (课上解释).

# 牛顿法关键点

- 实际工程中 Hessian 矩阵  $\mathbf{H}$  很难求,  $\mathbf{H}^{-1}$  更加难求.
- 解决思路:
  - 修正牛顿法: 当 Hessian 矩阵不是正定矩阵时, 可对 Hessian 矩阵进行修正:  $\mathbf{H}(\mathbf{x}_k) + \mathbf{E}$ , 典型的方法  $\mathbf{E} = \delta \mathbf{I}$ ,  $\delta > 0$  很小. 思考为什么这么取?
  - 拟牛顿法 (Quasi-Newton methods)

# 拟牛顿法 (1/3)

## 核心思想

- 统一深度下降法和牛顿法:

$$\mathbf{d}_k = -\mathbf{S}_k \mathbf{g}_k \quad (4)$$

其中  $\mathbf{S}_k = \begin{cases} \mathbf{I} & \text{steepest} \\ \mathbf{H}_k^{-1} & \text{Newton} \end{cases}$

- 不直接求  $\mathbf{H}_k^{-1}$ , 尝试用一正定矩阵逼近  $\mathbf{H}_k^{-1}$  (一阶的量慢慢近似二阶的量)
- 定义  $\boldsymbol{\delta}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ ,  $\boldsymbol{\gamma}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$
- 需要  $\mathbf{S}_{k+1} \boldsymbol{\gamma}_k = \boldsymbol{\delta}_k$ , 为什么?
- 但是, 只有  $\boldsymbol{\delta}_k$  和  $\boldsymbol{\gamma}_k$ , 是不可能计算出  $\mathbf{S}_{k+1}$  的, 继续用迭代的方法.

# 拟牛顿法 (2/3)

## DFP

- 给定初始  $\mathbf{S}_0 = \mathbf{I}$
- $\mathbf{S}_{k+1} = \mathbf{S}_k + \Delta \mathbf{S}_k, k = 0, 1, \dots$
- $\Delta \mathbf{S}_k = \alpha \mathbf{u} \mathbf{u}^T + \beta \mathbf{v} \mathbf{v}^T$ , 因此

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \alpha \mathbf{u} \mathbf{u}^T + \beta \mathbf{v} \mathbf{v}^T$$

- 两边乘以  $\gamma_k$ , 有  $\delta_k = \mathbf{S}_k \gamma_k + \underbrace{(\alpha \mathbf{u}^T \gamma_k)}_1 \mathbf{u} + \underbrace{(\beta \mathbf{v}^T \gamma_k)}_{-1} \mathbf{v} = \mathbf{S}_k \gamma_k + \mathbf{u} - \mathbf{v}$
- 解出  $\alpha = \frac{1}{\mathbf{u}^T \gamma_k}, \beta = -\frac{1}{\mathbf{v}^T \gamma_k}$ , 且有  $\mathbf{u} - \mathbf{v} = \delta_k - \mathbf{S}_k \gamma_k$ , 可得  $\mathbf{u}$  和  $\mathbf{v}$ , 从而最终解得:
- Davidon-Feltcher-Powell (DFP) 更新公式

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{\mathbf{S}_k \gamma_k \gamma_k^T \mathbf{S}_k}{\gamma_k^T \mathbf{S}_k \gamma_k} \quad (5)$$

# 拟牛顿法 (3/3)

## BFGS

Broyden-Fletcher-Goldfarb-Shanno (BFGS):  $S_0 = I$

$$S_{k+1} = S_k + \left(1 + \frac{\gamma_k^T S_k \gamma_k}{\delta_k^T \gamma_k}\right) \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{\delta_k \gamma_k^T S_k + S_k \gamma_k \delta_k^T}{\delta_k^T \gamma_k} \quad (6)$$

# 步长求取

- 1 每次迭代固定步长，实际中最常用，例如  $\alpha_k = \alpha = 0.1$
- 2 求导. 例如:  $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x} + 2\mathbf{b}^T \mathbf{x} + c$ , 需要解  $\min_{\alpha \geq 0} f(\mathbf{x} + \alpha \mathbf{d})$  则  $h(\alpha) = f(\mathbf{x} + \alpha \mathbf{d})$ , 则有  $\frac{\partial h(\alpha)}{\partial \alpha} = 0 \Rightarrow \alpha = -\frac{\mathbf{d}^T \nabla f(\mathbf{x})}{2\mathbf{d}^T \mathbf{A} \mathbf{d}}$
- 3 不精确的线搜索和 Armijo 条件

$$f(\mathbf{x}_k + \alpha \mathbf{d}_k) < f(\mathbf{x}_k) + c_1 \alpha \mathbf{g}^T(\mathbf{x}_k) \mathbf{d}_k$$

- 1 设置  $c_1 = 10^{-4}$ , 具体参考<sup>a</sup>. 先从  $\alpha = 1$  搜, 如果 Armijo 条件不满足, 设置一回调因子  $\beta \in (0, 1)$ , 将步长下调至  $\alpha = \beta \alpha$ . 如果还不满足, 继续回调 (backtracking line search), 从而保证步长不至于太小.

<sup>a</sup>Nocedal, Jorge; Wright, Stephen (1999). Numerical Optimization

线性回归建模

无约束优化梯度分析法

无约束优化迭代法

线性回归求解

网易微专业 犀牛学院 人工智能数学基础 Jason博士



## 解法 1: 利用梯度等于 0

- 试图学习:  $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$  使得  $f(\mathbf{x}^{(i)}) \approx y^{(i)}$

- 未知  $\bar{\mathbf{w}} = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}$ , 已知  $\mathbf{X} = \begin{bmatrix} \mathbf{x}^{(1)T} & 1 \\ \vdots & \vdots \\ \mathbf{x}^{(N)T} & 1 \end{bmatrix}_{N \times (d+1)}$ , 则有

$$\mathbf{y} \approx \mathbf{X} \bar{\mathbf{w}}$$

- 损失函数  $\|\mathbf{y} - \mathbf{X} \bar{\mathbf{w}}\|_2^2$ , 求解

$$\min \|\mathbf{y} - \mathbf{X} \bar{\mathbf{w}}\|_2^2$$

-  $g(\bar{\mathbf{w}}) = 0 \Rightarrow 2\mathbf{X}^T(\mathbf{X} \bar{\mathbf{w}} - \mathbf{y}) = 0 \Rightarrow \bar{\mathbf{w}}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

- 正则化

## 解法 2: 梯度下降

- 梯度下降法

$$\begin{aligned} \mathbf{g}(\bar{\mathbf{w}}) &= 2\mathbf{X}^T (\mathbf{X}\bar{\mathbf{w}} - \mathbf{y}) \\ &= 2 \sum_{i=1}^N \mathbf{x}^{(i)} \left( \mathbf{w}^T \mathbf{x}^{(i)} - y^{(i)} \right) \end{aligned}$$

$$\bar{\mathbf{w}} \leftarrow \bar{\mathbf{w}} - \alpha \mathbf{g}(\bar{\mathbf{w}})$$

- 随机梯度下降法（实际中很有用）

$$\left\{ i = 1 : N, 2\mathbf{x}^{(i)} \left( \mathbf{w}^T \mathbf{x}^{(i)} - y^{(i)} \right) \right\}$$

# 案例演示

网易微专业 犀牛学院 人工智能数学基础 Jason博士

# 本章总结

网易微专业 犀牛学院 人工智能数学基础 Jason博士

## 本章参考资料

- Nocedal, Jorge; Wright, Stephen. Numerical Optimization
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.