机器学习导引

第三章　线性回归（讲义）

主讲：何新卫

一、回归问题与线性模型

1.1 回归：预测连续数值

回归（regression）属于**有监督学习**：给定一个或多个自变量（特征）x，预测一个 **连续的数值**目标 y。它与分类的区别在于输出类型——分类输出有限个离散类别，回归输出连续实数。典型例子：由鱼的体长预测体重、由房屋特征预测房价、由气温历史预测明日温度。

以渔业为例，体长-体重分析既能评估鱼类整体的生长与健康状况，也能为资源保护提供理论依据。其目标就是找到一个函数 h_θ，把稳定、易得的体长换算成体重。

1.2 问题设置与符号

我们希望把标量目标 t 预测为输入 x 的函数。给定训练集

$$\mathcal{D} = \left\{ (x^{(i)}, t^{(i)}) \right\}_{i=1}^{N},$$

其中 $x^{(i)}$ 为第 i 个样本的**输入**，$t^{(i)}$ 为对应的**真实目标**。上标 (i) 只是样本编号，不是幂次。为与真实目标区分，我们用 $y^{(i)}$ 表示模型的 **预测值**。

1.3 线性回归模型与假设

线性回归假设因变量与自变量成**线性关系**：

$$y = w_1 x_1 + w_2 x_2 + \cdots + w_D x_D + b = \mathbf{w}^\top \mathbf{x} + b,$$

其中 $\mathbf{x} = (x_1, \ldots, x_D)^\top$ 为输入向量，$\mathbf{w} = (w_1, \ldots, w_D)^\top$ 为权重，b 为偏置；$\mathbf{w}$ 与 b 合称**参数**。参数的每一组具体取值确定一个 **假设（hypothesis）**，也就是一个具体函数

$$h_\theta(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b, \qquad \theta = (\mathbf{w}, b).$$

二、损失函数与代价函数

2.1 平方误差损失

要衡量某个假设拟合得好不好，需要损失函数。采用**平方误差（squared error）**：

$$\mathcal{L}(y, t) = \frac{1}{2}(y - t)^2,$$

其中 $y - t$ 称为**残差（residual）**，目标是使其绝对值尽可能小。系数 $\frac{1}{2}$ 只是为求导方便，用于消去平方求导产生的因子 2。

2.2 代价函数（均方误差）

单样本损失不够，我们关心整个训练集上的平均表现，即**代价函数**：

$$J(\mathbf{w}, b) = \frac{1}{2N} \sum_{i=1}^{N} \left(y^{(i)} - t^{(i)} \right)^2 = \frac{1}{2N} \sum_{i=1}^{N} \left(\mathbf{w}^\top x^{(i)} + b - t^{(i)} \right)^2 .$$

这就是**均方误差（Mean Squared Error, MSE）**（严格说还带系数 $\frac{1}{2}$）。注意 J 的自变量是参数 $(\mathbf{w}, b)$——数据是给定的，我们能调的只有这组参数。

2.3 向量化与数据矩阵

把全部训练样本逐行堆叠成**数据矩阵** $\mathbf{X} \in \mathbb{R}^{N \times D}$，目标排成向量 $\mathbf{t} \in \mathbb{R}^N$，则预测向量与代价函数可一次性算出：

$$\mathbf{y} = \mathbf{X}\mathbf{w} + b\mathbf{1}, \qquad J = \frac{1}{2N} \|\mathbf{y} - \mathbf{t}\|^2 .$$

向量化既让式子更简洁，也大幅提升计算速度（借助 BLAS/LAPACK，或 GPU 上的矩阵乘法）。

三、优化问题

3.1 最小化代价函数

训练线性回归 = 求解下述优化问题：

$$\min_{\mathbf{w}, b} \; J(\mathbf{w}, b).$$

由微积分，光滑函数的极小值（若存在）出现在**临界点**，即所有偏导数均为零的点。由此可得两种求解路线：**直接求解法（解析解）**与**梯度下降法**。

3.2 凸性与全局最小

展开 $J(\mathbf{w}, b)$，它对 $\mathbf{w}$、b 都是二次的，图像是一个**碗状抛物面**。因此 J 是凸函数，有**唯一的全局最小值**，不存在局部极小值陷阱。这保证了解析解与梯度下降（步长合适时）都能收敛到同一个最优参数。

四、解析解：矩阵形式推导

4.1 预备：逐分量梯度

对单个预测值 $y = \mathbf{w}^\top \mathbf{x} + b$，

$$\frac{\partial y}{\partial w_j} = x_j, \qquad \frac{\partial y}{\partial b} = 1.$$

由链式法则，损失对参数的偏导为

$$\frac{\partial \mathcal{L}}{\partial w_j} = (y - t)\, x_j, \qquad \frac{\partial \mathcal{L}}{\partial b} = y - t,$$

再对 N 个样本平均，得到代价函数的梯度分量：

$$\frac{\partial J}{\partial w_j} = \frac{1}{N} \sum_{i=1}^{N} \left(y^{(i)} - t^{(i)} \right) x_j^{(i)}, \qquad \frac{\partial J}{\partial b} = \frac{1}{N} \sum_{i=1}^{N} \left(y^{(i)} - t^{(i)} \right).$$

4.2 吸收偏置项

为把 b 一并纳入矩阵推导，在 $\mathbf{X}$ 最左侧补一列全 1，并把 b 作为 $\mathbf{w}$ 的第一个分量。为记号简洁，下文统一记模型为 $\mathbf{y}=\mathbf{X}\mathbf{w}$，其中 $\mathbf{X}\in\mathbb{R}^{N\times D}$ 已含全 1 列、$\mathbf{w}\in\mathbb{R}^{D}$ 已含 b。

4.3 代价函数的向量展开

$$J(\mathbf{w})=\frac{1}{2N}\|\mathbf{X}\mathbf{w}-\mathbf{t}\|^2=\frac{1}{2N}(\mathbf{X}\mathbf{w}-\mathbf{t})^\top(\mathbf{X}\mathbf{w}-\mathbf{t}).$$

展开括号：

$$(\mathbf{X}\mathbf{w}-\mathbf{t})^\top(\mathbf{X}\mathbf{w}-\mathbf{t})=\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w}-\mathbf{w}^\top\mathbf{X}^\top\mathbf{t}-\mathbf{t}^\top\mathbf{X}\mathbf{w}+\mathbf{t}^\top\mathbf{t}.$$

由于 $\mathbf{w}^\top\mathbf{X}^\top\mathbf{t}$ 是标量，等于其转置 $\mathbf{t}^\top\mathbf{X}\mathbf{w}$，故

$$J(\mathbf{w})=\frac{1}{2N}\left(\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w}-2\,\mathbf{t}^\top\mathbf{X}\mathbf{w}+\mathbf{t}^\top\mathbf{t}\right).$$

4.4 梯度与法方程

使用两条矩阵求导规则（$\mathbf{A}$ 对称时 $\nabla_{\mathbf{w}}(\mathbf{w}^\top\mathbf{A}\mathbf{w})=2\mathbf{A}\mathbf{w}$；$\nabla_{\mathbf{w}}(\mathbf{c}^\top\mathbf{w})=\mathbf{c}$），对 $\mathbf{w}$ 求梯度：

$$\nabla_{\mathbf{w}}J=\frac{1}{2N}\left(2\mathbf{X}^\top\mathbf{X}\mathbf{w}-2\mathbf{X}^\top\mathbf{t}\right)=\frac{1}{N}\left(\mathbf{X}^\top\mathbf{X}\mathbf{w}-\mathbf{X}^\top\mathbf{t}\right).$$

令梯度为零，得到**法方程（normal equations）**：

$$\mathbf{X}^\top\mathbf{X}\,\mathbf{w}=\mathbf{X}^\top\mathbf{t}.$$

4.5 闭式解与讨论

当 $\mathbf{X}^\top\mathbf{X}$ 可逆时，解析解为

$$\boxed{\mathbf{w}^*=(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{t}.}$$

先看清各矩阵的维度（设训练集有 N 个样本、D 个特征，且 $\mathbf{X}$ 已含偏置对应的全 1 列）：

矩阵	维度
数据矩阵 $\mathbf{X}$	$N\times D$
目标向量 $\mathbf{t}$	$N\times 1$
参数向量 $\mathbf{w}$	$D\times 1$
$\mathbf{X}^\top$	$D\times N$
$\mathbf{X}^\top\mathbf{X}$	$D\times D$
$\mathbf{X}^\top\mathbf{t}$	$D\times 1$
$(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{t}$	$D\times 1$

乘法链校验：$(\mathbf{X}^\top\mathbf{X})^{-1}$ 为 $D\times D$，乘 $\mathbf{X}^\top$（$D\times N$）得 $D\times N$，再乘 $\mathbf{t}$（$N\times 1$）得 $D\times 1$，恰好等于参数 $\mathbf{w}$ 的维度，说明维度自洽。

这即最小二乘的**闭式解（closed-form solution）**。几点说明：

- 系数 $\frac{1}{2N}$ 不影响最优解（梯度为零的方程两边同乘 N 后不变）；
- $\mathbf{X}^\top\mathbf{X}$ 为 $D\times D$ 矩阵，可逆的充要条件是 $\mathbf{X}$ **列满秩**（通常要求 $N\geq D$ 且特征间无完全线性相关）；
- 若 $\mathbf{X}^\top\mathbf{X}$ 奇异，可用 Moore–Penrose 伪逆 $(\mathbf{X}^\top\mathbf{X})^{+}\mathbf{X}^\top\mathbf{t}$ 得到最小范数解；

· 求逆复杂度为 $O(D^3)$，特征维度过高时直接求解代价大，这是引入梯度下降的原因之一。

五、梯度下降法

5.1 基本思想

梯度下降是**迭代算法**：先把参数初始化为某合理值（如全 0），再反复沿 **最陡下降方向（负梯度方向）** 微调参数，直到满足停止条件。

5.2 更新规则与学习率

由「若 $\partial J/\partial w_j > 0$ 则增大 w_j 使 J 增大，反之减小」可推出更新规则：

$$w_j \leftarrow w_j - \alpha \frac{\partial J}{\partial w_j} = w_j - \frac{\alpha}{N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right) x_j^{(i)}.$$

其中 α 是**学习率（learning rate）**，控制步长；通常取值很小，如 0.01 或 0.0001。向量形式为

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \nabla_{\mathbf{w}} J = \mathbf{w} - \frac{\alpha}{N}\mathbf{X}^{\top}(\mathbf{X}\mathbf{w} - \mathbf{t}).$$

梯度指向函数上升最快的方向，因此负梯度方向就是下降最快的方向。

5.3 为什么还需要梯度下降

· **适用范围更广**：不限于线性回归，也适用于神经网络等非线性模型；
· **更易实现**：配合自动微分（automatic differentiation），无需手推导数；
· **高维更高效**：解析解要矩阵求逆（$O(D^3)$），维度高时梯度下降更快。

六、多项式回归与特征映射

当数据呈非线性时，可拟合低阶多项式（多项式回归）。定义**特征映射** $\psi(x) = (1, x, x^2, x^3)^{\top}$，则

$$y = w_3 x^3 + w_2 x^2 + w_1 x + w_0 = \mathbf{w}^{\top}\psi(x).$$

形式上仍是线性模型（只是输入由 x 换成 $\psi(x)$），因此前面全部推导与算法 **原封不动适用**：多项式回归几乎是「免费」得来的。

七、泛化、过拟合与正则化

7.1 欠拟合与过拟合

· **欠拟合**：模型过于简单，连训练数据都无法拟合好（偏差过大）；
· **过拟合**：模型过于复杂，完美拟合训练数据甚至噪声，却无法泛化到新数据。

判断模型好坏要看**泛化能力**，而非训练集上的表现。

7.2 L2 正则化与权重衰减

过拟合多项式的系数通常很大，因此希望让权重保持较小。选择 L2 惩罚项 $R(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 = \frac{1}{2}\sum_j w_j^2$，

得到正则化代价

$$J_{\text{reg}} = J + \lambda R = J + \frac{\lambda}{2}\sum_j w_j^2,$$

在「拟合数据」与「控制权重范数」之间权衡，λ 为超参数（用验证集调整）。对其应用梯度下降可得

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha\left(\nabla_{\mathbf{w}} J + \lambda \mathbf{w}\right) = (1-\alpha\lambda)\mathbf{w} - \alpha\nabla_{\mathbf{w}} J,$$

即每一步先按 $(1-\alpha\lambda)$ 收缩权重，再沿梯度更新，故称**权重衰减（weight decay）**。

7.3 L1 正则化

L1 正则化在代价函数中加入系数绝对值之和 $\sum_j |w_j|$ 的惩罚，促使部分权重变为 0，使模型变得**稀疏**（可作特征选择）。

八、本章小结

- **模型**：$y = \mathbf{w}^\top \mathbf{x} + b$，参数是 $\mathbf{w}, b$；
- **损失**：平方误差 $\frac{1}{2}(y-t)^2$；代价 = 全部样本平均（MSE）；
- **优化**：凸问题，有唯一全局最小；
- **解析解**：$\mathbf{w}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{t}$ （法方程 $\mathbf{X}^\top\mathbf{X}\mathbf{w} = \mathbf{X}^\top\mathbf{t}$）；
- **梯度下降**：$\mathbf{w} \leftarrow \mathbf{w} - \alpha\nabla J$，适用范围更广、高维更高效；
- **特征映射**：$\psi(x)$ 把多项式回归变成线性模型；
- **正则化**：L2 权重衰减防过拟合，L1 稀疏。

九、课堂自测题

9.1 选择题

Q1　回归问题要预测的目标 y 通常是？
A. 有限个离散类别　B. 连续数值　C. 一段序列

Q2　代价函数 $J(\mathbf{w}, b)$ 的自变量是什么？
A. 输入 x　B. 真实目标 t　C. 参数 $\mathbf{w}, b$

Q3　MSE 代价函数是碗状凸曲面，这意味着？
A. 有唯一全局最小值　B. 有多个局部最小值　C. 可能没有最小值

Q4　求解线性回归模型参数的主要方法有？
A. 解析解（法方程）　B. 梯度下降　C. 支持向量机　D. 决策树

Q5　下列哪些做法有助于缓解过拟合？
A. 增加训练数据　B. 进一步提高模型复杂度　C. 加正则化　D. 用验证集调超参

9.2 参考答案与解析

Q1 B——回归输出连续实数，分类才输出离散类别。
Q2 C——数据是给定的，优化的是参数 $\mathbf{w}, b$。
Q3 A——凸函数有唯一全局最小，无局部极小陷阱。

Q4 A、B——法方程与梯度下降是两种主要参数估计方法，C、D 是其他算法。
Q5 A、C、D——增加数据、正则化、验证集调参都能缓解过拟合，提高复杂度反而加剧过拟合。

十、推荐阅读

- Altman N, Krzywinski M. Simple linear regression[J]. Nature Methods, 2015, 12(11): 999–1000. doi:10.1038/nmeth.3627.（课程阅读材料：Ch3-阅读材料-naturemathods2015.pdf）
- University of Toronto CSC411 (Roger Grosse). Lecture 6: Linear Regression. `https://www.cs.toronto.edu/~rgrosse/courses/csc411_f18/slides/lec06-slides.pdf`

附：中英文术语对照

线性回归	Linear Regression	回归	Regression
分类	Classification	监督学习	Supervised Learning
自变量 / 输入	Independent Variable / Input	因变量 / 目标	Dependent Variable / Target
预测值	Prediction	真实目标	Target
参数	Parameter	权重	Weight
偏置	Bias / Intercept	假设	Hypothesis
损失函数	Loss Function	平方误差	Squared Error
残差	Residual	代价函数	Cost Function
均方误差	Mean Squared Error (MSE)	数据矩阵	Design Matrix
向量化	Vectorization	优化问题	Optimization Problem
临界点	Critical Point	偏导数	Partial Derivative
梯度	Gradient	链式法则	Chain Rule
解析解 / 闭式解	Closed-Form Solution	法方程	Normal Equations
最小二乘	Least Squares	矩阵求逆	Matrix Inversion
梯度下降	Gradient Descent	学习率	Learning Rate
迭代算法	Iterative Algorithm	自动微分	Automatic Differentiation
特征映射	Feature Mapping	多项式回归	Polynomial Regression
泛化	Generalization	欠拟合	Underfitting
过拟合	Overfitting	正则化	Regularization
L2 正则化	L2 Regularization	权重衰减	Weight Decay
L1 正则化	L1 Regularization	稀疏	Sparsity
超参数	Hyperparameter	验证集	Validation Set
学习率	Learning Rate	全 1 向量	All-Ones Vector