机器学习导引

第四章　线性分类（讲义）

主讲：何新卫

一、分类问题与线性分类器

1.1 分类：预测离散类别

分类（classification）属于**有监督学习**：给定输入 $\mathbf{x}$，预测一个**离散的类别标签** y。它与回归最本质的区别在于输出类型——回归输出连续实数，分类输出有限个离散类别。典型例子：垃圾邮件识别（垃圾 v.s. 正常）、新闻分类（政治、体育、财经…）、图像分类（猫、狗、飞机…）、视频动作识别。

二分类预测一个二元因变量，常记作 $y \in \{0, 1\}$ 或 $y \in \{+1, -1\}$；多分类预测三个及以上的离散取值，如 $y \in \{$狗, 猫, 卡车, 飞机$\}$。分类器的形式为

$$h_\theta : \ \mathcal{X} \to \mathcal{Y},$$

其中 $\mathcal{X}$ 是输入空间，$\mathcal{Y}$ 是**离散**的输出空间。

1.2 形式化与训练目标

给定训练集 $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$，其中 $x^{(i)}$ 是第 i 个样本的特征，$y^{(i)}$ 是它对应的标签，上标 (i) 只是样本编号。目标是学一个分类器 h_θ，使新输入能被正确分类。概率视角下，即学习条件概率 $P(y \mid x)$，预测时选择概率最大的类别。

1.3 训练误差与测试误差

引入 **0–1 指示函数**：预测错误记为 1，预测正确记为 0。训练误差是训练集上预测错误的比例，测试误差是测试集上预测错误的比例。目标是两者都尽量小，尤其测试误差要小，代表分类器有好的**泛化能力**。

1.4 二元线性分类器与决策边界

在 D 维空间中，二元线性分类器由权重向量 $\mathbf{w}$ 和标量偏置 b 定义：

$$h_\theta(\mathbf{x}) = \operatorname{sign}\Big(\mathbf{w}^\top \mathbf{x} + b\Big) = \begin{cases} +1, & \mathbf{w}^\top \mathbf{x} + b > 0, \\ -1, & \mathbf{w}^\top \mathbf{x} + b \le 0. \end{cases}$$

其中 $\mathbf{w}$ 是权重向量（法向量），b 是偏置，$\mathbf{x}$ 是输入特征，$\mathbf{w}^\top \mathbf{x} + b = 0$ 是**决策边界（分隔超平面）**。一句话：线性分类器就是用超平面把空间切成两半。二维且 $b = 0$ 时，决策边界是过原点的直线 $\mathbf{w}^\top \mathbf{x} = 0$，$\mathbf{w}$ 是该直线的法向量。

二、从 0–1 损失到逻辑回归

2.1 0–1 损失及其问题

很自然的优化目标是 0–1 损失：

$$\mathcal{L}_{01}(\mathbf{x}, y, \mathbf{w}, b) = \mathbf{1}\Big\{h_\theta(x^{(i)}) \neq y^{(i)}\Big\}, \qquad J(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_{01}\Big(x^{(i)}, y^{(i)}, \mathbf{w}, b\Big).$$

但 0–1 损失在预测值跨越阈值处**跳变**，导数几乎处处为 0，在跳变点又不可导，梯度下降无法有效更新、优化停滞。因此需要一个**光滑、可导**的替代损失，这就引出逻辑回归。

2.2 线性逻辑分类器 LLC

引入**线性逻辑分类器（Linear Logistic Classifier, LLC）**：

$$h(\mathbf{x}; \mathbf{w}, b) = \sigma\Big(\mathbf{w}^\top \mathbf{x} + b\Big), \qquad \sigma(z) = \frac{1}{1+e^{-z}},$$

其中 σ 称为 **sigmoid（逻辑）函数**，把任意实数压缩到 $(0,1)$ 区间。完整写为

$$h(\mathbf{x}; \mathbf{w}, b) = \frac{1}{1+e^{-(\mathbf{w}^\top \mathbf{x}+b)}}.$$

注意：LLC 本身输出的是 0 到 1 之间的**连续得分**，还不是离散类别，只有加上阈值才变成分类器。

2.3 阈值与概率解释

设阈值为 0.5：

$$\hat{y} = \begin{cases} +1, & \sigma\big(\mathbf{w}^\top \mathbf{x} + b\big) > 0.5, \\ -1, & \text{否则}. \end{cases}$$

因 σ 单调递增，$\sigma(z) > 0.5 \iff z > 0$，故上述规则等价于 $\mathbf{w}^\top \mathbf{x} + b > 0$ 判 $+1$。即加上阈值后，LLC 就是线性分类器，决策边界仍是 $\mathbf{w}^\top \mathbf{x} + b = 0$。又因输出落在 $(0,1)$，可将其解释为概率：

$$h(\mathbf{x}) = P(y = +1 \mid \mathbf{x}), \qquad P(y = -1 \mid \mathbf{x}) = 1 - h(\mathbf{x}).$$

把「得分」变成「概率」，是逻辑回归的核心思想。

三、损失函数：从似然到交叉熵

3.1 负对数似然

训练目标是给正确类高概率，于是定义**负对数似然（negative log-likelihood, NLL）**。为简化，令标签 $y \in \{0, 1\}$，模型预测类别 1 的概率为

$$g^{(i)} = \sigma\Big(\mathbf{w}^\top \mathbf{x}^{(i)} + b\Big) = \frac{1}{1 + \exp\big[-\big(\mathbf{w}^\top \mathbf{x}^{(i)} + b\big)\big]}.$$

3.2 似然函数

在样本独立假设下，似然是每个样本「正确类概率」的连乘：

$$L(\mathbf{w}, b) = \prod_{i=1}^{n}\Big(g^{(i)}\Big)^{y^{(i)}}\Big(1 - g^{(i)}\Big)^{1-y^{(i)}}.$$

合并式子的含义：当 $y^{(i)} = 1$，第二项指数 $1 - y^{(i)} = 0$，只剩 $g^{(i)}$；当 $y^{(i)} = 0$，只剩 $1 - g^{(i)}$。

3.3 最大化似然等价于最小化交叉熵

最大化似然 $\Longleftrightarrow$ 最大化对数似然（log 单调递增）$\Longleftrightarrow$ 最小化负对数似然：

$$\min_{\mathbf{w},b} \; -\sum_{i=1}^{n} \Big[y^{(i)} \log g^{(i)} + (1-y^{(i)}) \log \big(1-g^{(i)}\big) \Big].$$

这就是逻辑回归的**交叉熵损失（cross-entropy loss）**，也叫负对数似然损失，通常除以 n 取平均。

3.4 L2 正则化

加上平均系数与 L2 正则化项，得到完整的代价函数：

$$\boxed{J_{\mathrm{lr}}(\mathbf{w},b) = -\frac{1}{n}\sum_{i=1}^{n} \Big[y^{(i)} \log g^{(i)} + (1-y^{(i)}) \log \big(1-g^{(i)}\big) \Big] + \lambda \|\mathbf{w}\|^2.}$$

其中 λ 是正则化系数（超参数），$\lambda\|\mathbf{w}\|^2$ 约束权重不要过大、防止过拟合。

四、梯度推导（重点）

4.1 预备：sigmoid 的导数

sigmoid 的导数可写成其自身与 1 减自身的乘积：

$$\sigma'(z) = \frac{e^{-z}}{(1+e^{-z})^2} = \frac{1}{1+e^{-z}} \cdot \frac{e^{-z}}{1+e^{-z}} = \sigma(z)\big(1-\sigma(z)\big).$$

记 $z^{(i)} = \mathbf{w}^\top \mathbf{x}^{(i)} + b$，则

$$\boxed{\sigma'\left(z^{(i)}\right) = g^{(i)}\left(1-g^{(i)}\right).}$$

这是整个推导的核心结论，后文会反复使用。

4.2 单样本损失对权重分量的偏导

先只考虑单个样本、且不含正则化项的损失：

$$\ell^{(i)} = -\Big[y^{(i)} \log g^{(i)} + (1-y^{(i)}) \log \big(1-g^{(i)}\big) \Big].$$

第一步：求 $g^{(i)}$ 对 w_j 的偏导。由链式法则与 4.1 的结论，

$$\frac{\partial g^{(i)}}{\partial w_j} = \sigma'\left(z^{(i)}\right) \cdot \frac{\partial z^{(i)}}{\partial w_j} = g^{(i)}\left(1-g^{(i)}\right) \cdot x_j^{(i)}.$$

第二步：求 $\ell^{(i)}$ 对 $g^{(i)}$ 的偏导。两项分别对 $g^{(i)}$ 求导，

$$\frac{\partial \ell^{(i)}}{\partial g^{(i)}} = -\frac{y^{(i)}}{g^{(i)}} + \frac{1-y^{(i)}}{1-g^{(i)}} = \frac{g^{(i)}-y^{(i)}}{g^{(i)}\left(1-g^{(i)}\right)}.$$

（这里用了 $-(1-y^{(i)})\log(1-g^{(i)})$ 求导得 $+\dfrac{1-y^{(i)}}{1-g^{(i)}}$ 的结论，注意符号。）

第三步：由链式法则合成，

$$\frac{\partial \ell^{(i)}}{\partial w_j} = \frac{\partial \ell^{(i)}}{\partial g^{(i)}} \cdot \frac{\partial g^{(i)}}{\partial w_j} = \frac{g^{(i)}-y^{(i)}}{g^{(i)}\left(1-g^{(i)}\right)} \cdot g^{(i)}\left(1-g^{(i)}\right) x_j^{(i)}$$

$$\boxed{\frac{\partial \ell^{(i)}}{\partial w_j} = \left(g^{(i)} - y^{(i)}\right) x_j^{(i)}.}$$

注意分子 $g^{(i)}\left(1-g^{(i)}\right)$ 恰好与分母约去，这正是 sigmoid 导数「好算」的关键。

4.3 单样本损失对偏置的偏导

同理，$z^{(i)}$ 对 b 的偏导为 1，故

$$\frac{\partial g^{(i)}}{\partial b} = g^{(i)}\left(1-g^{(i)}\right)\cdot 1 = g^{(i)}\left(1-g^{(i)}\right),$$

$$\frac{\partial \ell^{(i)}}{\partial b} = \frac{g^{(i)}-y^{(i)}}{g^{(i)}\left(1-g^{(i)}\right)}\cdot g^{(i)}\left(1-g^{(i)}\right) = \boxed{g^{(i)}-y^{(i)}.}$$

4.4 平均与正则化：完整梯度

把 n 个样本的偏导相加并除以 n，再加上正则化项 $\lambda\|\mathbf{w}\|^2$ 的导数 $2\lambda w_j$，得到

$$\boxed{\frac{\partial J_{\text{lr}}}{\partial w_j} = \frac{1}{n}\sum_{i=1}^{n}\left(g^{(i)}-y^{(i)}\right)x_j^{(i)} + 2\lambda w_j.}$$

正则化项中不含 b，故对 b 的偏导没有额外项：

$$\boxed{\frac{\partial J_{\text{lr}}}{\partial b} = \frac{1}{n}\sum_{i=1}^{n}\left(g^{(i)}-y^{(i)}\right).}$$

4.5 向量形式与维度校验

把对每个分量 w_j 的结果拼成向量，即得对 $\mathbf{w}$ 的梯度：

$$\boxed{\nabla_{\mathbf{w}} J_{\text{lr}}(\mathbf{w},b) = \frac{1}{n}\sum_{i=1}^{n}\left(g^{(i)}-y^{(i)}\right)\mathbf{x}^{(i)} + 2\lambda\mathbf{w}.}$$

维度校验（设特征为 D 维，即 $\mathbf{x}^{(i)}\in\mathbb{R}^{D\times 1}$）：

项	维度
$\mathbf{x}^{(i)}$	$D\times 1$
$\left(g^{(i)}-y^{(i)}\right)$	标量
$\sum_{i=1}^{n}\left(g^{(i)}-y^{(i)}\right)\mathbf{x}^{(i)}$	$D\times 1$
$\mathbf{w}$ 与 $2\lambda\mathbf{w}$	$D\times 1$
$\nabla_{\mathbf{w}} J_{\text{lr}}$	$D\times 1$（与 $\mathbf{w}$ 同形）

梯度的形状是 $D\times 1$，与 $\mathbf{w}$ 同形状；$\mathbf{x}^{(i)}$ 是第 i 个样本的特征向量；$2\lambda\mathbf{w}$ 来自正则化项 $\lambda\|\mathbf{w}\|^2$ 的导数。直观上看，梯度每一项都是 **「预测概率减去真实标签」的残差乘对应特征**，再加正则化项的导数——这与线性回归中 $(y^{(i)}-t^{(i)})x_j^{(i)}$ 的结构一致。

4.6 梯度下降更新

由梯度下降，迭代更新参数：

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha\,\nabla_{\mathbf{w}} J_{\text{lr}}, \qquad b \leftarrow b - \alpha\,\frac{\partial J_{\text{lr}}}{\partial b},$$

其中 α 是学习率。逻辑回归没有闭式解，但梯度下降能稳定求解；这套思路也是后面神经网络训练的基础。

五、多分类：Softmax

5.1 one-hot 标签

设共 K 个类别 $\{c_1, \ldots, c_K\}$。训练标签改为 K 维 **one-hot** 向量 $\mathbf{y}$：样本属于第 k 类，则第 k 位为 1、其余为 0，如 $\mathbf{y} = [0, 1, 0]^\top$ 表示属于第 2 类。

5.2 线性层输出 logits

线性映射

$$\mathbf{z} = \mathbf{W}^\top \mathbf{x} + \mathbf{b},$$

其中 $\mathbf{z}$ 的每个分量是对应类别的**得分（score）**，也叫 **logits**。注意 logits 还不是概率：可能为负，各分量之和也不为 1，需要 softmax 归一化。

5.3 Softmax 归一化

$$g_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}.$$

性质：每个元素非负 $g_k \geq 0$，所有元素之和为 1（$\sum_k g_k = 1$），可解释为 K 类上的概率分布。假设函数为 $h(\mathbf{x}; \mathbf{w}, \mathbf{b}) = \text{softmax}\big(\mathbf{W}^\top \mathbf{x} + \mathbf{b}\big)$，预测类别取 g 最大元素对应类别：$\hat{y} = \arg\max_k g_k$。

5.4 多分类交叉熵损失

单样本 NLL：

$$\mathcal{L}_{\text{nll}}(\mathbf{g}, \mathbf{y}) = -\sum_{k=1}^{K} y_k \log g_k = -\log g_{k^*},$$

最后一步因 $\mathbf{y}$ 是 one-hot，只剩真实类别 k^* 那一项。数据集上的平均损失为

$$\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{\text{nll}}\Big(\mathbf{g}^{(i)}, \mathbf{y}^{(i)}\Big) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_k^{(i)} \log g_k^{(i)}.$$

六、图像分类与 PyTorch 落地

6.1 图像分类

输入图像通常表示为张量 $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$，$H, W$ 为高和宽，C 为通道数（灰度图 $C = 1$，RGB 图 $C = 3$）；输出为类别标签或概率分布 $\mathbf{g}$；最终预测 $\hat{y} = \arg\max_k g_k$。图像分类的目标是学习从图像到概率的映射 $h_\theta : \mathbf{x} \to \mathbf{g}$。

6.2 像素归一化

实际操作中，总先对像素值做归一化：把所有像素除以 255，缩放到 0 到 1。目的是统一数值尺度，使梯度下降更稳定、收敛更快。

6.3 PyTorch 实现

- `torch.nn.Softmax(dim=...)`：在指定维度上做归一化；
- `torch.nn.CrossEntropyLoss()`：内部已包含 LogSoftmax 与 NLLLoss，因此**输入应是未经 softmax 的 logits**，标签是**类别索引（整数）**，不是 one-hot 向量。

官方文档见 `https://pytorch.org/docs/stable/generated/torch.nn.Softmax.html` 与 `https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html`。

6.4 权重的几何解释

把每个类别对应的权重向量 $\mathbf{W}$ 重塑回 $H \times W \times C$ 的图像并可视化，可观察到它提取到的其实是该类别图像的**平均模板（class template）**——例如「汽车」类的权重看起来像一辆模糊的汽车。这说明线性分类器本质上是在学每个类别的模板，再用内积衡量输入与模板的相似度。

七、本章小结

- **任务**：分类预测离散类别，输出空间是离散的；
- **模型**：二元线性分类器 $h = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$，决策边界 $\mathbf{w}^\top \mathbf{x} + b = 0$；
- **0–1 损失**：不可导、优化停滞，需要光滑替代；
- **逻辑回归**：$h = \sigma(\mathbf{w}^\top \mathbf{x} + b)$，输出解释为概率；
- **损失**：交叉熵 $-\frac{1}{n}\sum \left[y \log g + (1-y)\log(1-g)\right] + \lambda \|\mathbf{w}\|^2$；
- **梯度**：$\nabla_{\mathbf{w}} J = \frac{1}{n}\sum (g^{(i)} - y^{(i)})\mathbf{x}^{(i)} + 2\lambda \mathbf{w}$，$\frac{\partial J}{\partial b} = \frac{1}{n}\sum (g^{(i)} - y^{(i)})$；
- **多分类**：one-hot 标签、logits、softmax 归一化、交叉熵损失；
- **落地**：PyTorch 的 `Softmax` 与 `CrossEntropyLoss`。

八、课堂自测题

8.1 选择题

Q1　分类任务与回归任务最本质的区别是？

A. 是否使用梯度下降　B. 输出是离散类别还是连续数值　C. 是否属于监督学习　D. 是否需要训练数据

Q2　二元线性分类器 $h(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$ 的决策边界是？

A. $\mathbf{w}^\top \mathbf{x} + b = 0$　B. $\mathbf{w}^\top \mathbf{x} + b = 1$　C. $\text{sign}(\mathbf{w}^\top \mathbf{x} + b)$　D. $\sigma(\mathbf{w}^\top \mathbf{x} + b)$

Q3　逻辑回归中引入 sigmoid 函数 $\sigma(z) = \frac{1}{1+e^{-z}}$ 的主要目的是？

A. 直接把输出变成离散标签　B. 把线性得分映射到 $(0, 1)$ 区间，从而可解释为概率　C. 提高计算速度　D. 消除偏置项

Q4　关于 softmax 函数的输出，下列说法正确的是？

A. 各分量之和为 0　B. 各分量非负且之和为 1　C. 各分量一定大于 1　D. 输出与类别数无关

Q5　多分类交叉熵损失中，若真实类别对应的概率 $g_{k^*} = 0.5$，则该样本的损失为？

A. 0　B. 0.5　C. $-\log 0.5 = \log 2$　D. 1

8.2 参考答案与解析

Q1 B——分类输出离散类别，回归输出连续数值。
Q2 A——决策边界就是 $\mathbf{w}^\top \mathbf{x} + b = 0$ 这个分隔超平面。
Q3 B——sigmoid 把线性得分压缩到 $(0, 1)$，可解释为概率，本身不输出离散标签。
Q4 B——softmax 输出非负且和为 1，构成概率分布。
Q5 C——单样本损失为 $-\log($真实类别概率$) = -\log 0.5 = \log 2$。

九、推荐阅读

- MIT 6.036 (Introduction to Machine Learning). Notes: Linear Classifiers. `https://openlearninglibrary.mit.edu/assets/courseware/v1/9d904854b4ae0878cfdcedcdceabf937/asset-v1:MITx+6.036+1T2019+type@asset+block/notes_chapter_Linear_classifiers.pdf`
- Stanford CS231n. Linear Classification. `http://cs231n.github.io/linear-classify/`
- PyTorch Documentation. torch.nn.Softmax / torch.nn.CrossEntropyLoss. `https://pytorch.org/docs/stable/index.html`

附：中英文术语对照

分类	Classification	线性分类器	Linear Classifier
二分类	Binary Classification	多分类	Multi-class Classification
类别标签	Class Label	输入空间	Input Space
输出空间	Output Space	决策边界	Decision Boundary
分隔超平面	Separating Hyperplane	0–1 损失	0–1 Loss
逻辑回归	Logistic Regression	逻辑函数 / Sigmoid	Logistic / Sigmoid Function
负对数似然	Negative Log-Likelihood (NLL)	交叉熵	Cross-Entropy
似然函数	Likelihood	对数似然	Log-Likelihood
权重向量	Weight Vector	偏置	Bias / Intercept
正则化	Regularization	超参数	Hyperparameter
梯度	Gradient	偏导数	Partial Derivative
链式法则	Chain Rule	梯度下降	Gradient Descent
学习率	Learning Rate	残差	Residual
One-hot 编码	One-Hot Encoding	Logits	Logits / Scores
Softmax	Softmax	归一化	Normalization
图像分类	Image Classification	张量	Tensor
通道	Channel	像素归一化	Pixel Normalization
类别模板	Class Template	泛化能力	Generalization