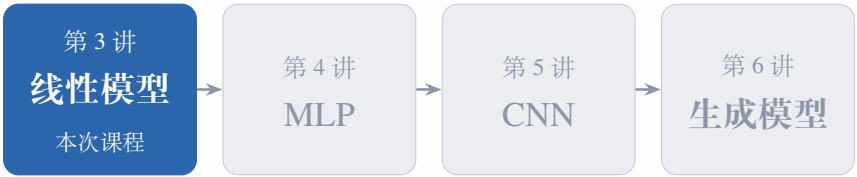
深度学习与计算机视觉

第三讲：从线性模型开始

廖振宇

电子信息与通信学院
华中科技大学
2026 年秋季学期

本讲在课程中的位置



本次课程：讨论“可解释”的线性分类基线；下次课程：非线性神经网络模型

本讲内容

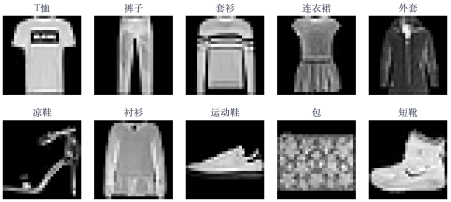
01 用机器学习实现自动图像分类

02 从线性得分到多分类概率

03 机器学习模型的训练和泛化

1 用机器学习实现自动图像分类

1 服饰图片：对计算机只是 784 个数字



来源：https://github.com/zalandoresearch/fashion-mnist

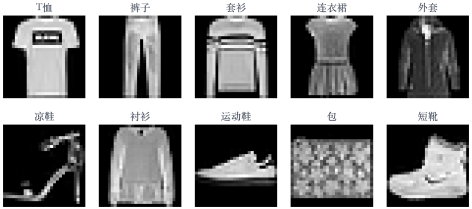
人看到的

轮廓、材质、部件等

机器学习模型“看”到的

28×28 个灰度值：$\mathbf{x} \in [0,1]^{28\times 28}$

1 从图片分类到标签预测



来源：https://arxiv.org/abs/1708.07747

$$\mathbf{x}_i \in \mathbb{R}^{28\times 28}, \qquad y_i \in \{1, \dots, 10\}$$

- 输入 $\mathbf{x}_i$：一张灰度服饰图像
- 标签 y_i：十个类别之一
- 预测 $\hat{y}_i$：机器学习模型给出的类别
- 成功标准：对未见样本 $\mathbf{x}'$ 有 $\hat{y}' = y'$

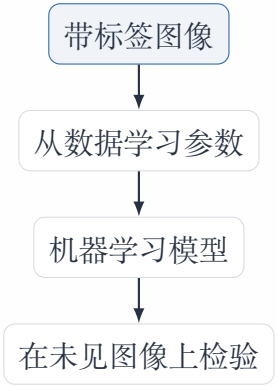
1 机器学习：从经验规则到数据驱动

方案 1：手写规则

“有鞋底”“有鞋帮”“像靴筒”：规则越来越复杂，难覆盖所有可能性

方案 2：数据驱动的机器学习

从（数据、标签）对中“学习”规则

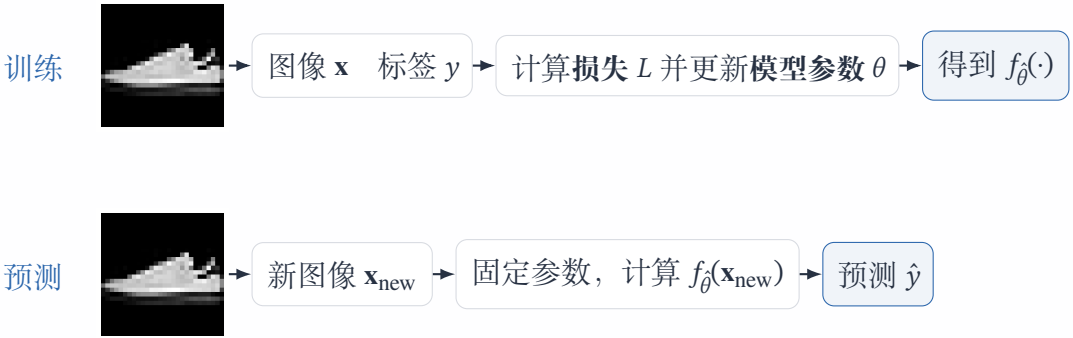


1 机器学习流程：六个相互支撑的组成部分

结论

更复杂的机器学习模型不一定更好：应与数据、（学习）目标和评价指标协同设计

1 训练时有答案，预测时只有输入



1 数据驱动：数据“局限”了模型能够学到什么

问题 1：覆盖不足

训练集中没有某类数据（外观），模型就“学”不到

问题 2：标签有误

错误的监督导致模型性能下降

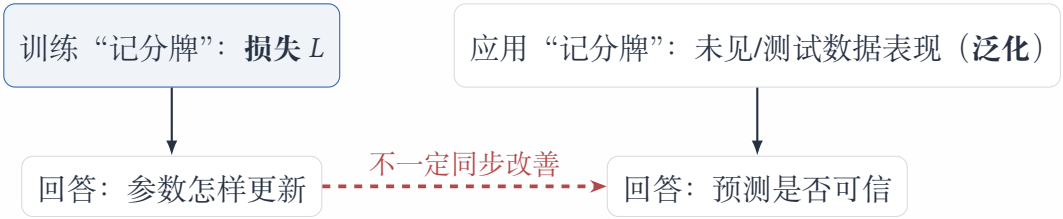
问题 3：分布变化

训练“环境“不同于实际（使用）环境，可能导致性能下降

结论

数据量重要，但数据质量（如是否代表目标/任务）同样重要

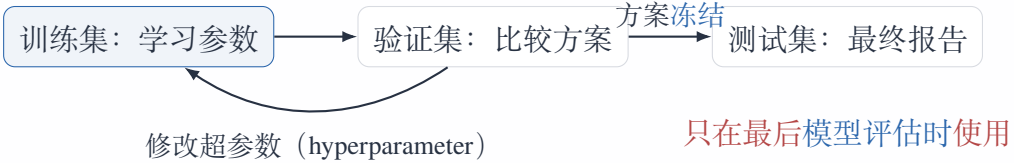
1 模型训练：先定义成功，再决定怎样努力



思考

训练准确率达到 100%，是否已经完成任务？

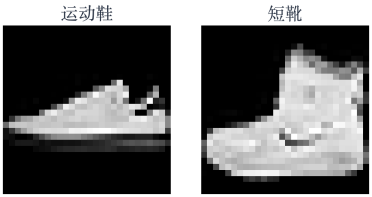
1 训练、验证（validation）和测试



注意

测试集用于模型评估，不是“更大的验证集”；其结果不应参与机器学习模型选择

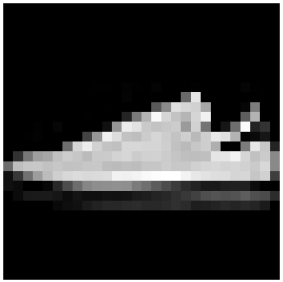
1 用“运动鞋 vs 短靴”分类例子，走完一次机器学习流程



来源：https://github.com/zalandoresearch/fashion-mnist

数据 带标签的运动鞋与短靴图像
表示 28×28 灰度值，展平为 784 维向量 $\mathbf{x}$
模型 得分函数 f_θ 输出图片 $\mathbf{x}$ 属于“运动鞋”的得分 s
损失 预测概率与真实标签 y 相差多大
优化 根据损失梯度调整参数
评价 在未见图像 $\mathbf{x}_{\text{new}}$ 上是否仍能分对

1 分类模型首先需要给出一个可比较的得分



运动鞋相对得分
$s = f_\theta(\mathbf{x})$

↑ 比较

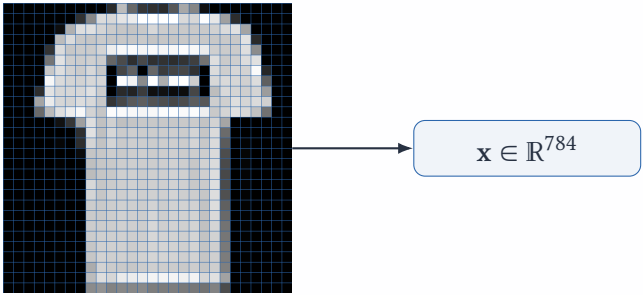
短靴作为参照类

思考

得分 s 应该怎样由 $\mathbf{x}$ 的 784 个像素计算?

2
从线性得分到多分类概率

2 图像展平：保留数值，但丢掉空间关系



按固定像素顺序展开

结论

展平保留每个位置的灰度值，但没有显式表达相邻像素共同构成的局部结构，第五讲将用**卷积神经网络**建模这些邻域关系

线性模型将输入特征加权求和

定义 | 线性模型

$$s = f_\theta(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b = \sum_{j=1}^{784} w_j x_j + b, \qquad \theta = (\mathbf{w}, b)$$

结论

线性回归、逻辑回归与 Softmax 回归都可以共享线性模型/结构，区别在于得分 s 和模型输出的关系

2 权重：每个像素对得分的“支持”和“反对”

$$s = f_\theta(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b = (+0.8)(0.9) + (-0.4)(0.5) + (+0.6)(0.7) + b$$

- 判断与决策

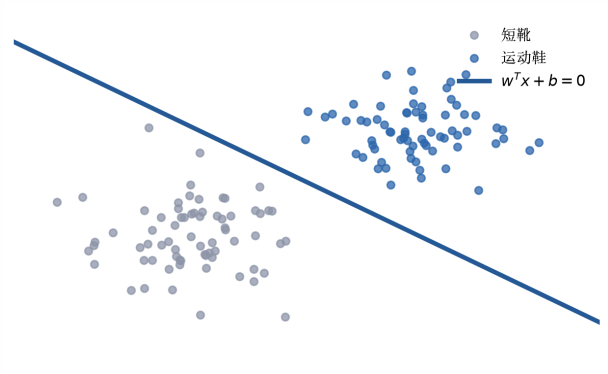
$s \geq s_{\text{threshold}} \Rightarrow \hat{y} = 1$ (判断为：运动鞋)　　$s < s_{\text{threshold}} \Rightarrow \hat{y} = 0$ (判断为：短靴)

$w_j > 0$	$w_j < 0$
像素 x_j 变亮，会提高“运动鞋”得分	像素 x_j 变亮，会降低“运动鞋”得分

思考

如果所有像素都 独立 贡献，机器学习模型能否直接识别“鞋底与鞋帮同时出现”?

2 线性决策边界将输入空间切分为两部分



$$\mathbf{w}^\top \mathbf{x} + b = 0$$

- 二维：直线
- 三维：平面
- 高维：超平面

延伸了解

原始输入空间中，维线性模型产生线性边界，核方法 (Kernel method, SVM) 通过变换/映射产生新的特征

2 从实数得分 s 到概率（估计）

$$s \in (-\infty, +\infty)$$

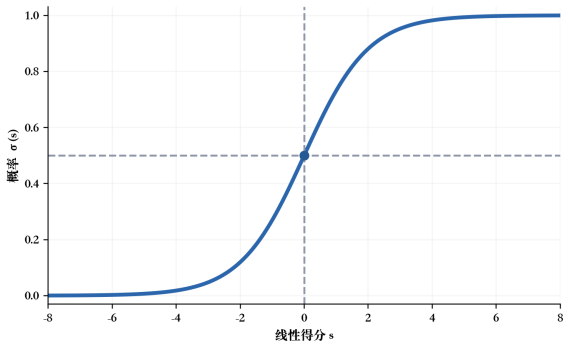
↓ 需要单调映射

$$p \in [0, 1]$$

注意

“得分更高”可用于“判断强弱”，但不能直接“解释”为 80% 概率

2 Sigmoid 函数将得分 s “压缩” $[0, 1]$



概率（估计）$p = \text{Sigmoid}(s)$

$$= \sigma(s) = \frac{1}{1 + e^{-s}}$$

- $s = 0 \Rightarrow p = 0.5$
- s 越大，概率 p 越接近 1
- 单调映射保留得分 s 的大小顺序

2 逻辑回归用 Logit 链接线性模型与概率

定义 | 逻辑回归

$$s = f_\theta(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b, \qquad \mathrm{Logit}(p) = \log \frac{p}{1-p} = s \iff p = \mathrm{Sigmoid}(s)$$

输入 $\mathbf{x}$ → 线性得分 s → Sigmoid $\sigma(s) = \frac{1}{1+e^{-s}}$ → 正类概率 p

广义线性模型用链接函数扩展线性模型

定义 | 线性模型

$$\mu(\mathbf{x}) = s, \qquad s = \mathbf{w}^\top \mathbf{x} + b$$

输出均值本身由线性模型给出

例 | 线性回归

连续输出，使用恒等链接 $g(\mu) = \mu$

定义 | 广义线性模型

$$g(\mu(\mathbf{x})) = s, \qquad s = \mathbf{w}^\top \mathbf{x} + b$$

响应分布来自指数族，链接后的均值由线性模型给出

例 | 逻辑回归

$Y \mid \mathbf{x} \sim \text{Bernoulli}(p)$，使用 Logit 链接

结论

线性回归是广义线性模型的特例；逻辑回归使用 Bernoulli 响应与 Logit 链接，仍被称为线性分类器：因为分类边界还是线性的

2 一张图像由此得到二分类概率

$s = 1.39,$ $p(\text{运动鞋} \mid \mathbf{x}) = \sigma(1.39) \approx 0.80$

短靴 0.20 运动鞋 0.80

结论

大小关系决定类别；概率保留预测的“不确定”度量

2 二元交叉熵衡量二分类概率预测

定义 | 二元交叉熵

$$L(\cdot) = -\big[y\log p + (1-y)\log(1-p)\big], \quad y \in \{0,1\}, \quad p \in [0,1]$$

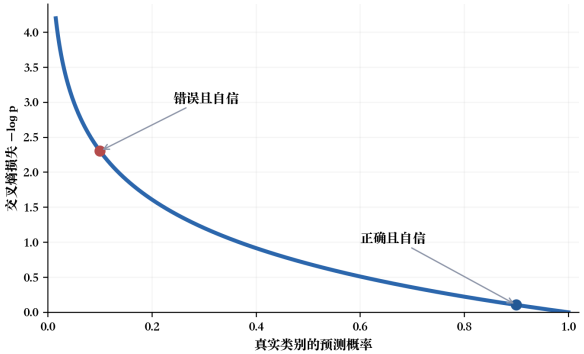
$y = 1$

$L = -\log p$：真实正类概率越高，损失越小

$y = 0$

$L = -\log(1-p)$：错误正类概率越高，损失越大

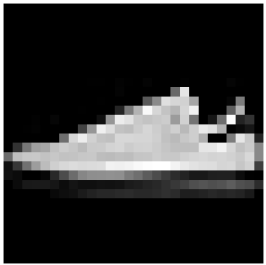
2 交叉熵“惩罚”自信的错误



真实类别概率 p_y

$$L = -\log p_y$$

- $p_y = 0.9 \Rightarrow L \approx 0.11$
- $p_y = 0.5 \Rightarrow L \approx 0.69$
- $p_y = 0.1 \Rightarrow L \approx 2.30$



- 一维：$s_k = \mathbf{w}_k^\top \mathbf{x} + b_k, \qquad k = 1, \dots, 10$
- 向量化

$$\mathbf{s} = f_\theta(\mathbf{x}) = \mathbf{W}^\top \mathbf{x} + \mathbf{b} \in \mathbb{R}^{10}$$

其中 $\mathbf{s} = [s_1, \dots, s_{10}]^\top$，$\mathbf{b} = [b_1, \dots, b_{10}]^\top$，
$\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_{10}] \in \mathbb{R}^{784 \times 10}$

结论

二分类的标量得分 s 在十分类中推广为向量 $\mathbf{s}$，每个类别拥有一组“独立”的权重参数 $(\mathbf{w}_k, b_k)$

2 十个得分让十个类别可以比较，但还不是概率

$$\mathbf{s} = (-1.2, -2.0, -0.7, -1.4, -0.8, 0.4, -0.2, 1.2, -0.5, 0.9)$$

问题 1：可以为负

得分没有 $[0, 1]$ 的概率范围限制

问题 1：总和不固定

十个得分并不会（自动）加和为 1

思考

怎样既保留相对大小，又让十个类别形成一个归一化的“概率分布”？

2 Softmax 回归把多个线性模型归一化为概率

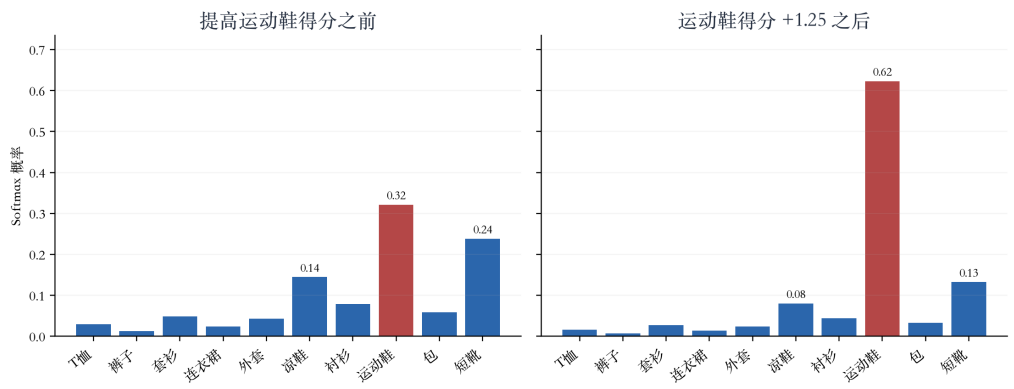
定义 | Softmax 回归（多项逻辑回归）

$$s_k = \mathbf{w}_k^\top \mathbf{x} + b_k, \qquad p_k = \frac{e^{s_k}}{\sum_{j=1}^{10} e^{s_j}}, \qquad \text{满足 } p_k \in [0,1], \quad \sum_{k=1}^{10} p_k = 1$$

结论

Softmax 回归可看作多分类广义线性模型，线性的是类别得分，归一化概率分布向量由 Softmax 映射得到

2 一个得分变化会重新分配 Softmax 概率



2 多分类交叉熵：只读取真实类别的预测概率

定义 | 多分类交叉熵（Cross-entropy）

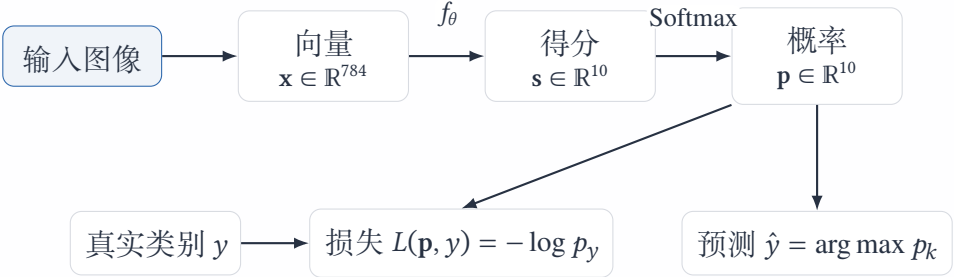
$$L(\cdot) = -\sum_{k=1}^{10} y_k \log p_k = -\log p_y$$

其中 y 是真实类别编号，$\mathbf{y} = \text{onehot}(y_k)$，$p_y$ 是真实类别的预测概率

结论

Softmax 形成多类概率，交叉熵负责判断这些概率与真实标签是否一致

2 一张服饰图像怎样变成十类预测



结论

Softmax + 交叉熵：把图片分类（结果）变成可优化目标

线性模型贯穿二分类与多分类

思考

二分类和多分类都保留线性模型，如何从预测错误中学习模型参数?

3
机器学习模型的训练和泛化

3 训练：在参数空间中寻找更低训练损失

数学记号

$d = 784$ 为输入维度，$K = 10$ 为类别数，n 为训练样本数

$$\mathbf{x}_i \in \mathbb{R}^d, \quad y_i \in \{1, \dots, K\}, \quad \mathbf{y}_i = \text{onehot}(y_i) \in \{0, 1\}^K$$

$$\mathbf{s}_i = f_\theta(\mathbf{x}_i) \in \mathbb{R}^K, \quad \mathbf{p}_i = \text{Softmax}(\mathbf{s}_i) \in \mathbb{R}^K$$

$$\mathbf{W} \in \mathbb{R}^{d \times K}, \quad \mathbf{b} \in \mathbb{R}^K, \quad \boldsymbol{\theta} = \begin{bmatrix} \text{vec}(\mathbf{W}) \\ \mathbf{b} \end{bmatrix} \in \mathbb{R}^q, \quad q = dK + K$$

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^q} L_{\text{train}}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} L(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{y}_i)$$

注

参数空间的每一点都是一个 q 维参数向量 $\boldsymbol{\theta}$；本讲中 $q = 784 \times 10 + 10 = 7850$

3 梯度：指出损失（局部）上升最快的方向

$$\nabla_{\boldsymbol{\theta}} L_{\text{train}}(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial L_{\text{train}}}{\partial \theta_1} \\ \vdots \\ \frac{\partial L_{\text{train}}}{\partial \theta_q} \end{bmatrix} \in \mathbb{R}^q$$

- 一维导数：描述函数随（一个）变量的局部变化
- 梯度：把所有参数的局部变化率形成一个向量
- $-\nabla_{\boldsymbol{\theta}} L_{\text{train}}$ 是局部下降最快的方向

思考

某个偏导数为正时，为降低损失，应增大还是减小该参数？

3 沿梯度反方向更新参数

定义｜梯度下降

$$\theta(t+1) = \theta(t) - \eta \nabla_\theta L(\theta(t))$$

实际常用：mini-batch（批次）损失 L_B

第 t 次更新使用（随机抽取的）样本子集
$B_t \subset \{1, \dots, n\}$，$|B_t| = m$

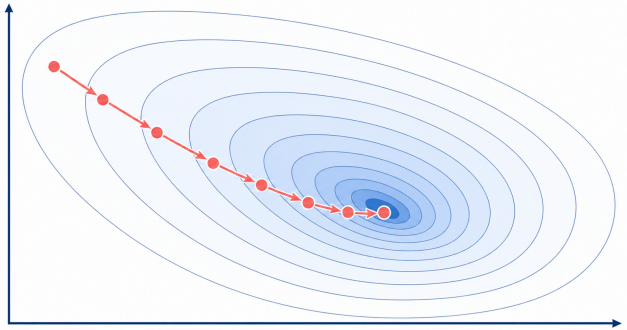
$$L_{B_t}(\theta) = \frac{1}{m} \sum_{i \in B_t} L(\theta; \mathbf{x}_i, \mathbf{y}_i)$$

L_B 就是批次 B 中 m 个样本损失的平均

迭代与步长

$\theta(t)$：更新前参数
$\theta(t+1)$：更新后参数
$\eta > 0$：学习率

3 梯度下降过程可视化

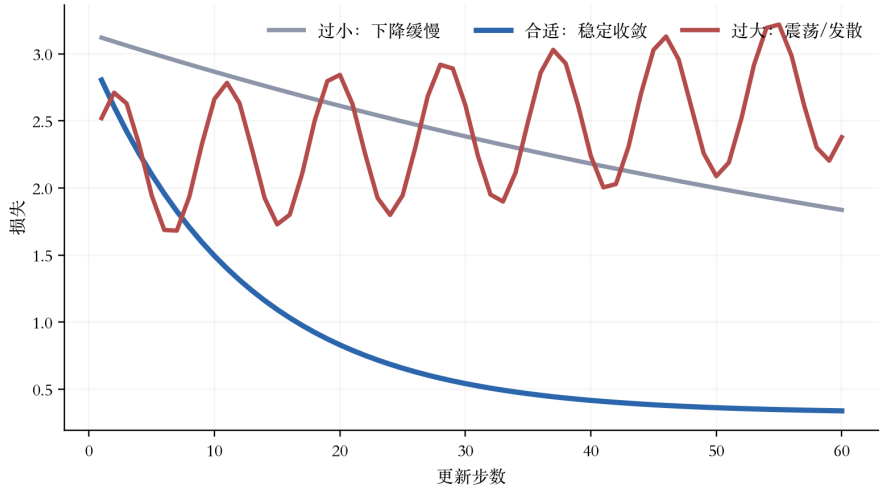


横轴 θ_1，纵轴 θ_2，蓝色曲线为损失等高线

$$\boldsymbol{\theta}(t) = \begin{bmatrix} \theta_1(t) \\ \theta_2(t) \end{bmatrix}$$

- 等高线：相同损失参数
- 梯度 $\nabla_{\boldsymbol{\theta}} L$ 与等高线正交
- 负梯度给出局部下降最快方向
- 形成梯度下降轨迹 $\boldsymbol{\theta}(t)$

3 学习率大小：决定训练是收敛还是震荡



3 Softmax + 交叉熵的梯度

- 对单个样本，$\mathbf{x} \in \mathbb{R}^d$，得分 $\mathbf{s}$，（估计）概率 $\mathbf{p}$，onehot 标签 $\mathbf{y} \in \mathbb{R}^K$
- 其相关关系

$$\mathbf{s} = f_\theta(\mathbf{x}) = \mathbf{W}^\top \mathbf{x} + \mathbf{b} \in \mathbb{R}^K \qquad \mathbf{p} = \mathrm{Softmax}(\mathbf{s})$$

$$L = -\sum_{k=1}^{K} y_k \log p_k \qquad \text{交叉熵损失}$$

- 得到梯度

$$\nabla_{\mathbf{s}} L = \mathbf{p} - \mathbf{y} \in \mathbb{R}^K$$

- 进一步得到

$$\nabla_{\mathbf{W}} L = \mathbf{x}(\mathbf{p} - \mathbf{y})^\top \in \mathbb{R}^{d \times K} \qquad \nabla_{\mathbf{b}} L = \mathbf{p} - \mathbf{y} \in \mathbb{R}^K$$

结论

梯度 $\propto$ “预测概率减去真实标签”

3 基于一个 mini-batch 数据更新一次参数

把一个 mini-batch 的 m 个样本形成批量矩阵

$$\mathbf{X}_{B_t} \in \mathbb{R}^{m\times d}, \qquad \mathbf{Y}_{B_t}, \mathbf{P}_{B_t} \in \mathbb{R}^{m\times K}, \qquad \mathbf{1}_m \in \mathbb{R}^m$$

$$L_{B_t}(\boldsymbol{\theta}(t)) = \frac{1}{m}\sum_{i\in B_t} L(\boldsymbol{\theta}(t); \mathbf{x}_i, \mathbf{y}_i)$$

$$\nabla_{\mathbf{W}} L_{B_t} = \frac{1}{m}\mathbf{X}_{B_t}^{\top}(\mathbf{P}_{B_t} - \mathbf{Y}_{B_t}) \in \mathbb{R}^{d\times K}$$

$$\nabla_{\mathbf{b}} L_{B_t} = \frac{1}{m}(\mathbf{P}_{B_t} - \mathbf{Y}_{B_t})^{\top}\mathbf{1}_m \in \mathbb{R}^K$$

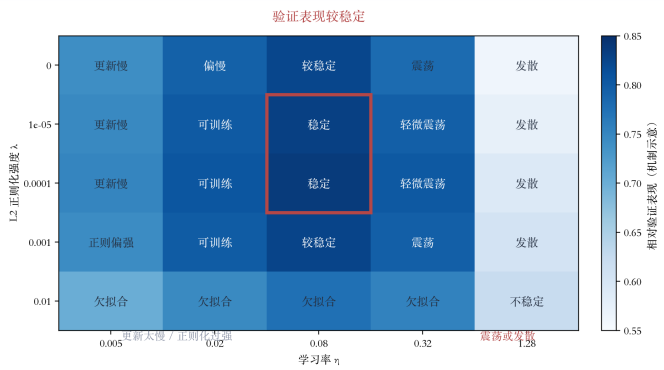
一次 mini-batch 梯度下降更新

1. 用 $\mathbf{W}(t), \mathbf{b}(t)$ 计算批量概率 $\mathbf{P}_{B_t}$
2. 计算 L_{B_t} 及两个梯度

$$\mathbf{W}(t+1) = \mathbf{W}(t) - \eta\nabla_{\mathbf{W}} L_{B_t}$$

$$\mathbf{b}(t+1) = \mathbf{b}(t) - \eta\nabla_{\mathbf{b}} L_{B_t}$$

3 超参数决定模型怎样训练

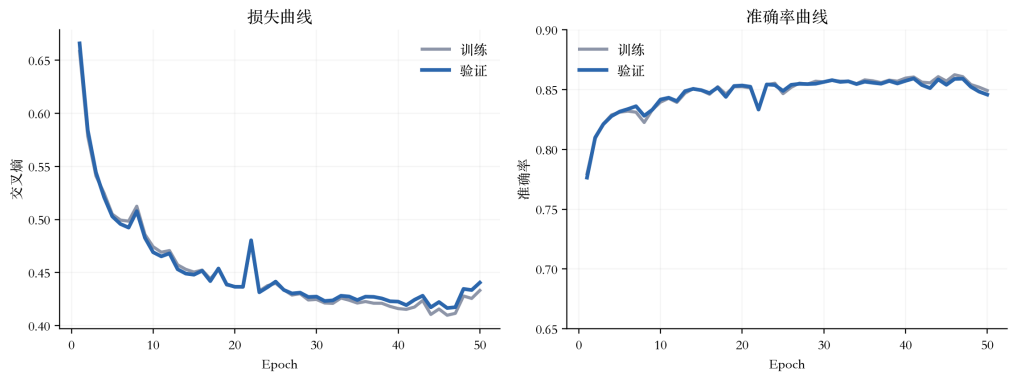


定义 | 模型超参数

训练前设定、用于控制训练过程或模型约束的量；如学习率、正则化强度和 batch size 等

图中颜色与文字为机制示意，不是实测值

3 训练曲线显示机器学习模型是否真的在学习



思考

训练损失下降和准确率上升说明了什么，不能说明什么？

3 泛化是在未见样本上保持可靠表现

定义 | 泛化

机器学习模型从训练样本学到的规律，迁移到同一任务中未参与训练的新样本的能力

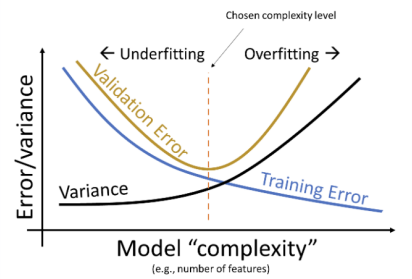
训练集表现

L_{train} 可被优化过程（e.g. 梯度下降）直接降低

泛化：未见数据表现

需验证集和（最终）测试集提供证据

3 欠拟合与过拟合要同时看训练与验证表现



来源：UC Berkeley Data 100：Cross Validation and Regularization

思考

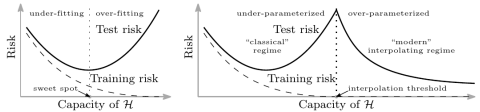
只有一条训练损失曲线，能区分欠拟合与过拟合吗？

定义 | 模型容量

机器学习模型能够表达的数据规律的丰富程度；与模型参数量有关，但不能简单等同于参数量

- **欠拟合**：训练误差与验证误差都高
- **过拟合**：训练误差低，验证误差反而升高

3 双下降：模型容量与泛化并非总是简单 U 型关系



定义 | 双下降

测试误差随模型容量增加先下降、在拟合阈值附近上升，随后可能再次下降

拟合阈值：机器学习模型首次把训练误差降至 ≈ 零的区域

双下降告诉我们什么？

- 过参数区域：$q > n$ 时可能有无穷多个解，最小范数拟合解展现出“双下降”
- 双下降标明不是“参数越多一定越好”：模型结构、正则化与优化（”隐式正则“）都会改变曲线

来源：Belkin et al., PNAS 2019，图 1

3 准确率看总统计量，混淆矩阵帮助理解错误

$$\text{Accuracy} = \frac{\text{预测正确的样本数}}{\text{总样本数}}$$

准确率

总体上有多少样本预测正确

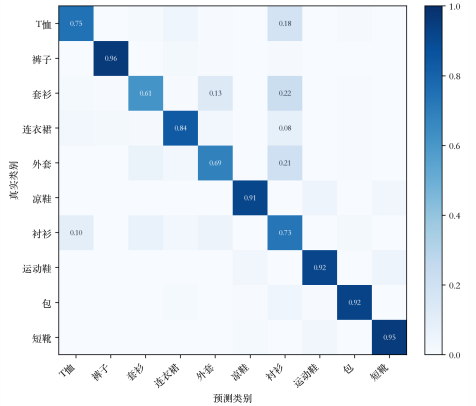
混淆矩阵

哪些真实类别被（系统地）预测成哪些类别

结论

相同准确率可以对应完全不同的错误模式，可由混淆矩阵反映

3 外观相近的类别易被混淆



来源：https://github.com/zalandoresearch/fashion-mnist

实际测试结果

Softmax + 交叉熵线性基线准确率：**82.8%**

- T 恤常被判为衬衫
- 套衫、外套、衬衫相互混淆
- 鞋类与包更易区分

3 召回率：给定真实类别被“答对”的比例

定义 | 召回率

类别 c 的召回率（Recall）

$$\text{Recall}_c = \frac{TP_c}{TP_c + FN_c} = \frac{C_{cc}}{\sum_{j=1}^{K} C_{cj}}$$

C_{cj} 表示真实类别 c 被预测为类别 j 的样本数

True Positive: TP_c

真实为类别 c，并且预测为类别 c

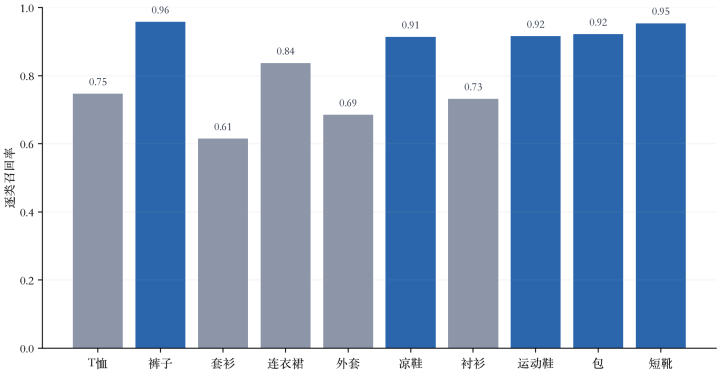
False Negative: FN_c

真实为类别 c，却被预测为其他类别

注

召回率：真实属于类别 c 的样本，有多少被机器学习模型正确识别?

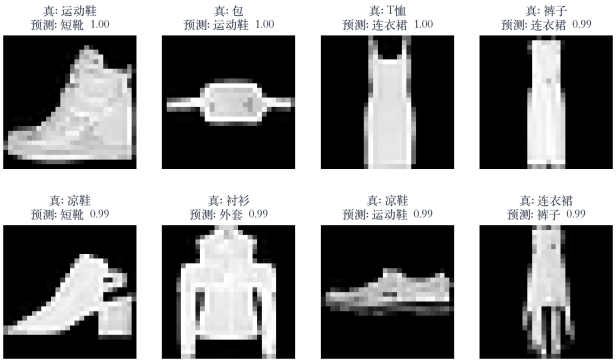
3 逐类召回率：哪些类别系统性“易错”



结论

总体准确率不反应类别差异：套衫、外套与衬衫的召回率明显更低

3 典型错误



思考

哪些错误来自低分辨率图像本身（e.g. 图像模糊导致分类难），哪些更可能来自线性分类器的表达局限？

3 训练完成后，每个类别得到 784 个像素权重

$$\mathbf{W}(T) = \begin{bmatrix} | & & | \\ \mathbf{w}_1(T) & \cdots & \mathbf{w}_K(T) \\ | & & | \end{bmatrix} \in \mathbb{R}^{d\times K} \qquad \mathbf{b}(T) \in \mathbb{R}^K$$

- 得到分别得分

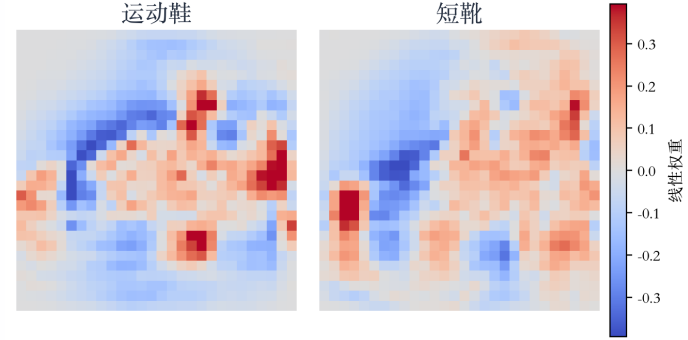
$$s_k = \mathbf{w}_k(T)^\top \mathbf{x} + b_k(T) \qquad d = 784, \quad K = 10$$

权重理解与可视化

考虑训练后对应第 k 个类别的 $d = 784$ 个像素位置的模型权重

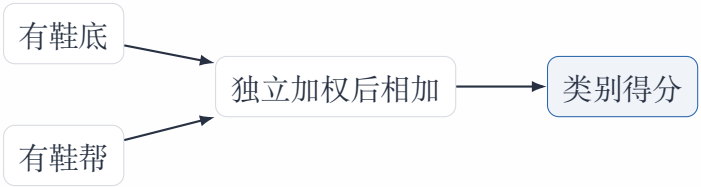
3 权重可视化

$\mathbf{w}_k(T) \in \mathbb{R}^d \quad (d = 784)$ → reshape $(28, 28)$ → 类别 k 的权重图



- 红：增加提高类别得分
- 蓝：增加降低类别得分

3 单纯加权求和无法表达像素间相关关系！



线性分类器无法直接表达“鞋底与鞋帮应同时出现”

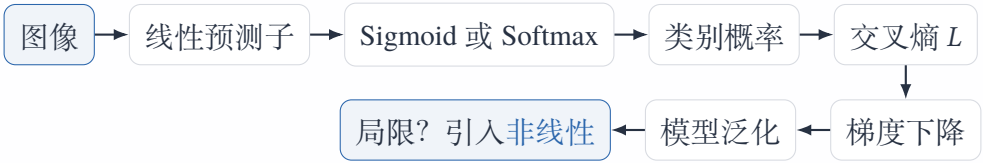
结论

这是线性分类器的表达局限，第四讲将引入**非线性层**学习特征间的组合关系

3 怎么办？

- **特征工程**：把有用结构显式写入输入
- **集成学习**：让多个模型互补，详细内容见附录
- **非线性神经网络**：通过非线性层自动学习组合规律

3 本讲总结



思考

怎样组合多个非线性变换，让机器学习模型表达非线性规律？

附录导览

- **A | 线性与广义线性模型补充** 线性回归、矩阵形式与隐式正则化
- **B | 验证与重采样** 交叉验证、Bootstrap 与评价方式
- **C | 分类评价与部署取舍** 混淆矩阵、Precision、Recall、PR 与 ROC
- **D | 类别不平衡** 准确率失真与常见处理路线
- **E | 集成学习** Bagging、Boosting 与 AdaBoost
- **参考资料** 数据、图片与参考文献

注

各章节相互独立，可按需查阅

附录 A

线性与广义线性模型补充

线性回归、矩阵形式与隐式正则化

A 线性回归与均方误差

线性回归：用同样的线性函数预测连续数值

$$\hat{y}_i = \mathbf{w}^\top \mathbf{x}_i + b, \qquad L(\mathbf{w}, b) = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

线性模型 vs 广义线性模型

线性回归是恒等链接的特例；逻辑回归与 Softmax 回归保留线性预测子，再通过链接函数得到类别概率

A 线性回归的矩阵形式

考虑增广数据矩阵 $\mathbf{X} \in \mathbb{R}^{n \times q}$，其中 $q = d + 1$ $\mathbf{w} \in \mathbb{R}^q$，$\mathbf{y} \in \mathbb{R}^n$

$$L(\mathbf{w}) = \frac{1}{n}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2, \qquad \nabla_{\mathbf{w}} L = \frac{2}{n}\mathbf{X}^\top(\mathbf{X}\mathbf{w} - \mathbf{y})$$

若 $\mathbf{X}^\top\mathbf{X}$ 可逆

$$\hat{\mathbf{w}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$$

实践提示

数值计算通常不显式求逆，而采用 QR、SVD 或（梯度下降等）迭代优化

A 梯度下降的隐式正则化

定义 | 隐式正则化

模型、损失或优化没有显式加入正则项，但（在多个低损失解之间）表现出特定的选择偏好

欠定线性回归：最小范数解

若 $q > n$ 且 $\mathbf{X}\mathbf{w} = \mathbf{y}$ 有多个解，从 $\mathbf{w}(0) = \mathbf{0}$ 出发的梯度下降在适当步长下趋向

$$\mathbf{w}_\star = \arg\min_{\mathbf{X}\mathbf{w}=\mathbf{y}} \|\mathbf{w}\|_2 = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^\dagger\mathbf{y}$$

算法不包含 $\lambda\|\mathbf{w}\|_2^2$，却选择了最小欧氏范数的拟合解

参考：Gunasekar et al., ICML 2018、Soudry et al., JMLR 2018、Lyu & Li, ICLR 2020

广义线性模型：线性可分的逻辑回归

无显式正则化时，$\|\mathbf{w}(t)\|_2 \to \infty$；但归一化方向趋向最大间隔分类器（SVM）

（部分）神经网络

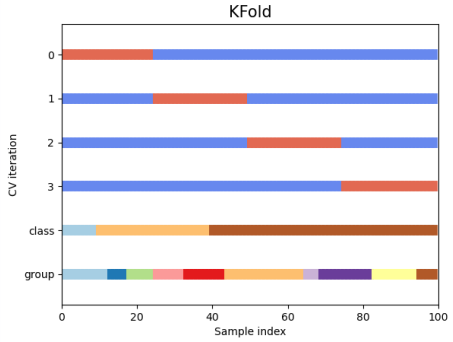
某些齐次网络、损失和优化条件下存在类似结论

附录 B

验证与重采样

交叉验证、Bootstrap 与评价方式

B K 折交叉验证轮流承担验证任务

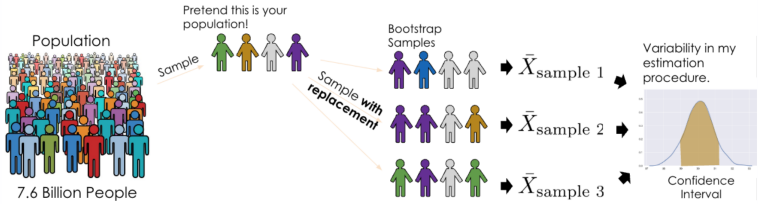


来源：scikit-learn：Visualizing cross-validation behavior

$$\widehat{M}_{\mathrm{CV}} = \frac{1}{K}\sum_{k=1}^{K} M_k$$

- 第 k 轮：第 k 折验证，其余 $K-1$ 折训练
- 模型共训练 K 次，报告均值与波动
- 分类常用分层划分；分组或时间数据需保留依赖结构

B Bootstrap 的有放回重采样机制



$$\left(\mathbf{x}_i^*, y_i^*\right)_{i=1}^n \overset{\text{iid}}{\sim} \hat{P}_n, \qquad \hat{P}_n = \frac{1}{n}\sum_{i=1}^n \delta_{(\mathbf{x}_i, y_i)}$$

同一观测可重复出现；一次重采样平均包含约 63.2% 的不同原始观测

一种机制，两种用途

- 重复计算统计量，估计不确定性
- 生成不同训练集，作为 Bagging 的基础
- 约 36.8% 的袋外样本可辅助评价，但不等同于独立测试集

图片：UC Berkeley Data 100、参考：Efron, Annals of Statistics, 1979

B 评价方式取决于数据量与依赖结构

方法	适合	主要代价或风险
固定留出	数据较多、需要快速迭代	对一次划分较敏感
K 折交叉验证	数据有限、模型训练不太昂贵	训练成本约为 K 倍
Bootstrap	估计统计量不确定性、构造 Bagging 样本	重采样分布与原分布有差异

共同边界

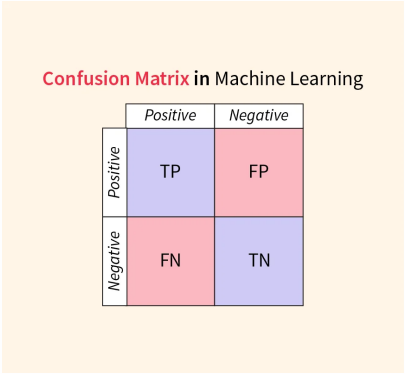
时间序列、同一患者的多次观测或同一视频的相邻帧不能当作独立样本随机打散

附录 C

分类评价与部署取舍

混淆矩阵、Precision、Recall、PR、ROC 与任务代价

C 混淆矩阵把行人检测分为四种结果



来源：Scaler Topics：Confusion Matrix in Machine Learning

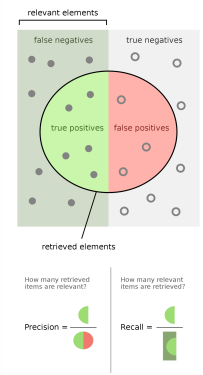
设正类为“前方存在需要处置的行人”

- TP：有行人且检测到
- FN：有行人但漏检
- FP：没有行人却报警
- TN：没有行人且未报警

结论

评价指标的意义来自具体任务中四类结果的代价

C Precision 关注误报，Recall 关注漏报



来源：Shaped：Evaluating recommendation systems

定义 | Precision

$$\text{Precision} = \frac{TP}{TP + FP}$$

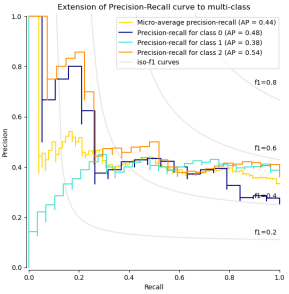
所有报警中有多少确实对应行人，误报越少，Precision 越高

定义 | Recall

$$\text{Recall} = \frac{TP}{TP + FN}$$

所有真实行人中有多少被找到，漏检越少，Recall 越高

C 宏平均让稀有类别拥有同等发言权



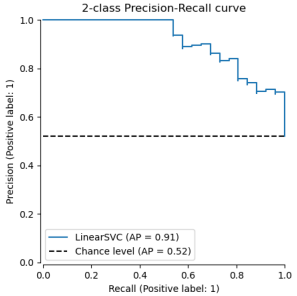
来源：scikit-learn：Precision-Recall 多类别示例

$$F_{1,c} = 2\frac{P_c R_c}{P_c + R_c}, \qquad F_1^{\text{macro}} = \frac{1}{K}\sum_{c=1}^{K} F_{1,c}$$

- 宏平均：先逐类计算，再平均，每类权重相同
- 微平均：先汇总所有类别的 TP/FP/FN，再计算；每个目标权重相同
- 车辆数量远多于行人时，微平均可能掩盖行人类别的失败

图为 scikit-learn Iris 示例，只用于说明多类别曲线和等 F_1 线

C PR 曲线展示阈值变化带来的取舍



来源：scikit-learn：Precision-Recall

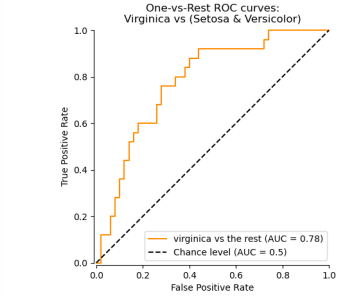
- 降低检测阈值通常找回更多行人，Recall 上升
- 同时可能产生更多误报，Precision 下降
- 正类稀少时，PR 曲线通常比准确率更有信息
- 选择工作点必须结合漏检与误报代价

思考

行人检测应选择曲线上的哪个点，仅看 AP 足够吗？

图为 scikit-learn 二分类示例，不是自动驾驶实测结果

C ROC/AUC 不直接给出部署阈值



来源：scikit-learn：Multiclass ROC（Iris 示例，非自动驾驶实测）

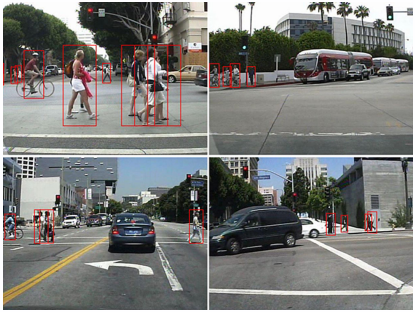
$$\mathrm{TPR} = \frac{TP}{TP + FN}, \qquad \mathrm{FPR} = \frac{FP}{FP + TN}$$

- ROC 随阈值变化描绘 TPR 与 FPR
- AUC 可解释为随机正样本得分高于随机负样本的概率
- 正类极少时，即使 FPR 很小，FP 的绝对数量仍可能很大

边界

AUC 不直接给出某个部署阈值下的 Precision、Recall 或实际代价

C 自动驾驶中漏检与误报的代价并不对称



来源：IEEE Spectrum：Pedestrian Detection

具体场景

正类：车辆前方存在需立即处置的行人

- FN：错失制动时机，安全代价很高
- FP：引起不必要制动、追尾风险或乘坐不适

$$\mathbb{E}[C](\tau) = C_{FP}P_\tau(FP) + C_{FN}P_\tau(FN)$$

注意

最优阈值 τ^* 不必是 0.5；还取决于概率校准、场景与部署分布

附录 D

类别不平衡

准确率失真与常见处理路线

D 类别不平衡为何让准确率失真

若 99% 样本属于负类，恒预测负类也有 99% 准确率，却完全找不到正类

- 同时报告混淆矩阵、逐类召回率和宏平均指标
- 训练分布、测试分布与部署分布需分别说明
- 不要仅为“平衡数字”随意重采样测试集

D 处理类别不平衡的常见路线

1. 数据层：过采样少数类、欠采样多数类
2. 损失层：为不同类别或错误设置不同权重
3. 决策层：根据代价移动分类阈值
4. 评价层：报告逐类与宏平均指标

每种路线都可能改变机器学习模型学到的概率或最终决策，需要在独立验证集上检查

附录 E

集成学习

Bagging、Boosting 与 AdaBoost

E 集成学习用多个模型弥补单模型局限

定义｜集成学习

将多个基学习器的预测组合为一个最终预测，使不同模型的错误能够互补

对 T 个基学习器 h_t

$$H(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

- 关键不是模型数量，而是个体模型有一定能力且错误具有差异
- 分类可投票，回归可平均
- 权重 α_t 可以相同，也可以由训练过程学习

E Bagging 用不同 Bootstrap 样本并行训练

1. 对 $t = 1, \ldots, T$，从原训练集独立生成 Bootstrap 样本 B_t
2. 在 B_t 上训练基学习器 h_t
3. 回归取平均，分类投票

$$H(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} h_t(\mathbf{x})$$

- 个体模型可并行训练
- 对高方差模型尤其有效
- 随机森林进一步随机选择特征，使树之间更不相同

E Boosting 依次修正当前组合仍然犯的错误

伪代码 | 广义序贯 Boosting 框架

1. 输入 $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$，初始化样本重要性 $\mathbf{a}(1) = \frac{1}{n}\mathbf{1}_n$
2. 对 $t = 1, \dots, T$
   - 2.1 按 $\mathbf{a}(t)$ 训练弱学习器 h_t
   - 2.2 根据 h_t 的当前错误确定组合权重 α_t
   - 2.3 更新并归一化 $\mathbf{a}(t+1)$，提高尚未处理好样本的重要性
3. 输出 $H(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^T \alpha_t h_t(\mathbf{x})\right)$

注

这是框架级伪代码；AdaBoost 进一步规定了 α_t 和样本权重的具体更新

E AdaBoost 用指数更新强调错分样本

伪代码 | 二分类 AdaBoost，$y_i \in \{-1, +1\}$

1. 初始化 $a_i(1) = 1/n$
2. 对 $t = 1, \ldots, T$
   - 2.1 在权重 $\mathbf{a}(t)$ 下训练 h_t
   - 2.2 $\varepsilon_t = \sum_{i=1}^{n} a_i(t)\, \mathbb{1}[h_t(\mathbf{x}_i) \neq y_i]$
   - 2.3 $\alpha_t = \frac{1}{2} \log \frac{1-\varepsilon_t}{\varepsilon_t}$
   - 2.4 $a_i(t+1) = \frac{a_i(t) \exp[-\alpha_t y_i h_t(\mathbf{x}_i)]}{Z_t}$，其中 Z_t 负责归一化
3. 输出 $H(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right)$

结论

错分样本权重增加；误差较小的弱分类器获得更大的组合权重

E AdaBoost 何时会遇到困难

- 标签噪声会让算法持续追逐无法解释的样本
- 基学习器若不优于随机猜测，组合权重失去意义
- 顺序训练难以完全并行
- 集成提高性能的同时也增加解释与部署成本

思考

若一个高权重样本其实是错误标签，下一轮弱学习器会发生什么？

数据、图片与参考文献

来源与参考文献

研究论文

[1] H. Xiao, K. Rasul, R. Vollgraf (2017), "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," arXiv:1708.07747

[2] M. Belkin, D. Hsu, S. Ma, S. Mandal (2019), "Reconciling modern machine-learning practice and the bias-variance trade-off," *PNAS*, 116(32):15849–15854

[3] S. Gunasekar, J. D. Lee, D. Soudry, N. Srebro (2018), "Characterizing Implicit Bias in Terms of Optimization Geometry," *ICML*, PMLR 80:1832–1841

[4] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, N. Srebro (2018), "The Implicit Bias of Gradient Descent on Separable Data," *JMLR*, 19(70):1–57

[5] K. Lyu, J. Li (2020), "Gradient Descent Maximizes the Margin of Homogeneous Neural Networks," *ICLR*

[6] B. Efron (1979), "Bootstrap Methods: Another Look at the Jackknife," *Annals of Statistics*, 7(1):1–26

[7] J. A. Nelder, R. W. M. Wedderburn (1972), "Generalized Linear Models," *JRSS A*, 135(3):370–384

数据与本地实验： Fashion-MNIST 来自 Zalando Research 官方仓库；训练曲线、混淆矩阵、逐类召回率、典型错误与权重图由本讲实验生成，参数与结果见 `assets/generated/metrics.json`

网络图源： UC Berkeley Data 100、scikit-learn 官方示例、Scaler、Shaped、IEEE Spectrum

完整网址与逐页用途见同目录 `SOURCES.md`；每张网络图片所在页也保留了来源