模式识别与机器学习

Lecture Notes

笔记五

模型评估与工程实践

目录

1 数据集划分与模型拟合理论

1.1 数据集划分 (Dataset Partitioning)

在机器学习任务中，我们通常将原始数据集划分为三个互斥的部分：训练集、验证集和测试集。这种划分是评估模型泛化能力、防止过拟合的基础。

1.1.1 三大集合的定义与功能

- **训练集 (Training Set)**：用于训练模型，通过数据让模型确定拟合曲线的参数（权重和偏置）。
- **验证集 (Validation Set)**：也称为开发集 (Dev Set)。主要用于：
  1. 进行模型选择 (Model Selection)。
  2. 辅助构建模型，通过验证误差来调整超参数 (Hyperparameters)。
  3. 这是一个可选步骤，但在深度学习中至关重要。
- **测试集 (Test Set)**：仅用于测试已经训练完成的模型之最终准确度。它不参与任何训练或调参过程，是衡量泛化能力的“最终考卷”。

1.1.2 常见划分比例

根据数据规模的不同，划分比例会有显著差异：

场景	训练集	验证集	测试集
传统机器学习	60% / 70%	20% / 10%	20%
大数据/深度学习	98%	1%	1%

表 1: 不同数据规模下的建议划分比例

1.1.3 实践进阶：K 折交叉验证 (K-fold Cross-Validation)

当数据集规模较小时，单次划分（Hold-out）可能导致评估结果具有偶然性。为了更充分地利用数据并获得稳健的模型评价，通常采用 **K 折交叉验证**：

- **核心流程：**

  1. 将训练集随机划分为 K 个大小相等的互斥子集（通常 $K = 5$ 或 10）。
  2. 每次迭代中，选取其中 1 份作为验证集，其余 $K - 1$ 份作为训练集。
  3. 重复 K 次，确保每份数据都被作为验证集使用过一次。

- **结果评估：**最终性能指标为 K 次实验结果的平均值。

- **意义：**有效地利用了每一条数据，减小了因划分不均带来的偏差，使超参数（Hyperparameters）的调整更加可靠。

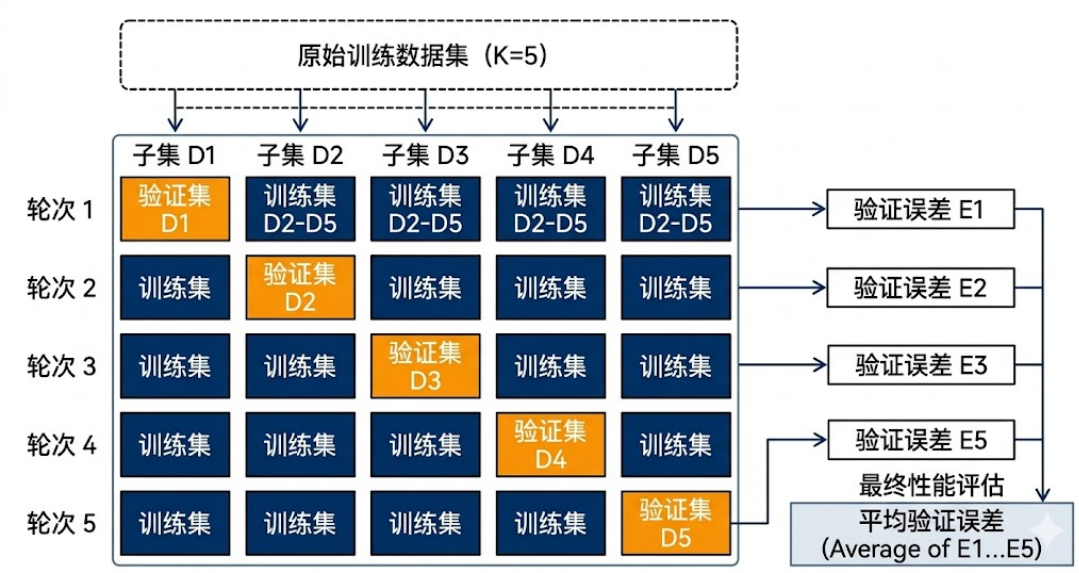

图 1: K 折交叉验证 (K=5) 示意图

1.2 欠拟合与过拟合 (Underfitting vs. Overfitting)

模型学习的核心目标是寻找训练误差与泛化误差之间的平衡。

1.2.1 基本定义

- **欠拟合：**模型无法准确捕捉数据规律，训练集和测试集错误率均高，通常因模型过于简单。

- **过拟合**：模型在训练集上表现极好，但测试集错误率高，因模型过度复杂而“背下”了噪声。
- **正合适**：模型成功捕捉底层规律，泛化能力良好。

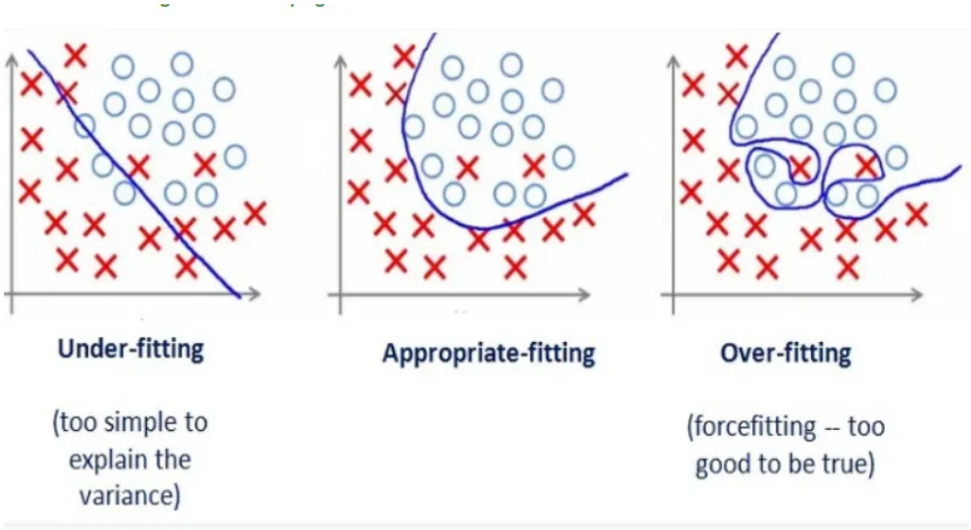

图 2: 欠拟合、过拟合与正合适的对比示意图

1.2.2 数学视角：多项式拟合

假设我们使用多项式函数进行回归分析，其数学表达式为：

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

其中 M 代表多项式的阶数，即模型复杂度。

- 当 $M = 0, 1$ 时：模型呈直线，无法拟合正弦波，导致欠拟合。
- 当 $M = 3$ 时：曲线平滑且接近数据分布趋势，拟合正合适。
- 当 $M = 9$ 时：曲线经过了每一个训练样点，但剧烈震荡，导致过拟合。

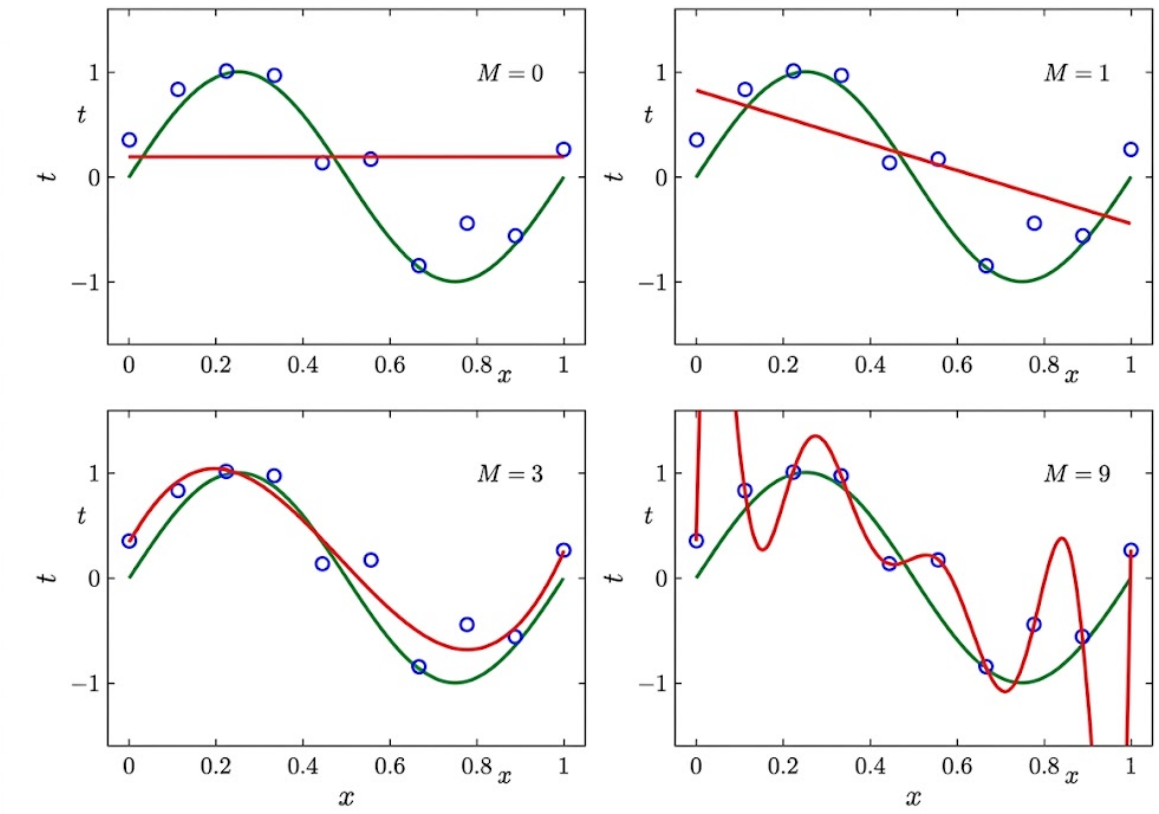

图 3: 多项式拟合对数据分布的影响示例

1.3 模型容量与误差曲线

随着模型复杂度的增加：

1. 训练误差单调下降。

2. 泛化误差先降后升。

3. 最优容量位于泛化误差最低点，左侧欠拟合，右侧过拟合。

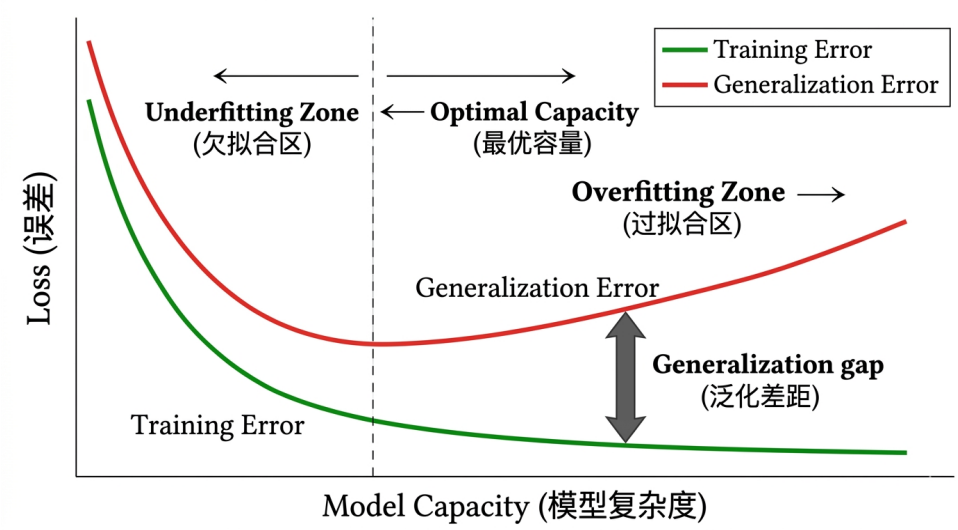

图 4: 误差随复杂度的变化趋势

1.4 理论升华：偏差-方差权衡 (Bias-Variance Tradeoff)

误差的来源不仅仅是模型没学好，从数学角度看，泛化误差（Generalization Error）可以分解为三部分：

$$E(y - \hat{f}(x))^2 = \text{Bias}[\hat{f}(x)]^2 + \text{Var}[\hat{f}(x)] + \sigma^2$$

- **偏差 (Bias)**：描述模型预测值的期望与真实值之间的差距。
  - **高偏差**意味着模型过于简单，无法学习数据的基本规律，对应 **欠拟合**。
- **方差 (Variance)**：描述模型在不同训练集上的表现稳定性（即模型预测值的波动程度）。
  - **高方差**意味着模型对训练集中的随机噪声过于敏感，导致模型在不同数据上的表现不稳定，对应 **过拟合**。
- **噪声 (σ^2)**：数据本身的不可消除误差（不可约误差），代表了当前任务的精度上限。

结论：随着模型复杂度增加，偏差通常会减小，但方差会增大。机器学习的目标就是找到一个“甜点位（Sweet Spot）”，使得偏差与方差之和最小。

下面我们通过偏差-方差权衡靶心示意图来理解偏差和方差如何影响预测点在四个象限的分布：

- **低偏差、低方差 (Low Bias, Low Variance)**：所有预测点紧密聚集在靶心。这是最理想的状态，模型既准确又稳定。
- **低偏差、高方差 (Low Bias, High Variance)**：预测点虽然整体围绕中心，但分布非常发散。这通常对应 **过拟合 (Overfitting)**，模型对训练数据的随机噪声过于敏感。
- **高偏差、低方差 (High Bias, Low Variance)**：预测点紧密聚集，但远离中心目标。这对应 **欠拟合 (Underfitting)**，模型过于简单，无法捕捉数据的底层规律。
- **高偏差、高方差 (High Bias, High Variance)**：预测点既偏离目标又非常散乱。这是模型表现最差的情况。

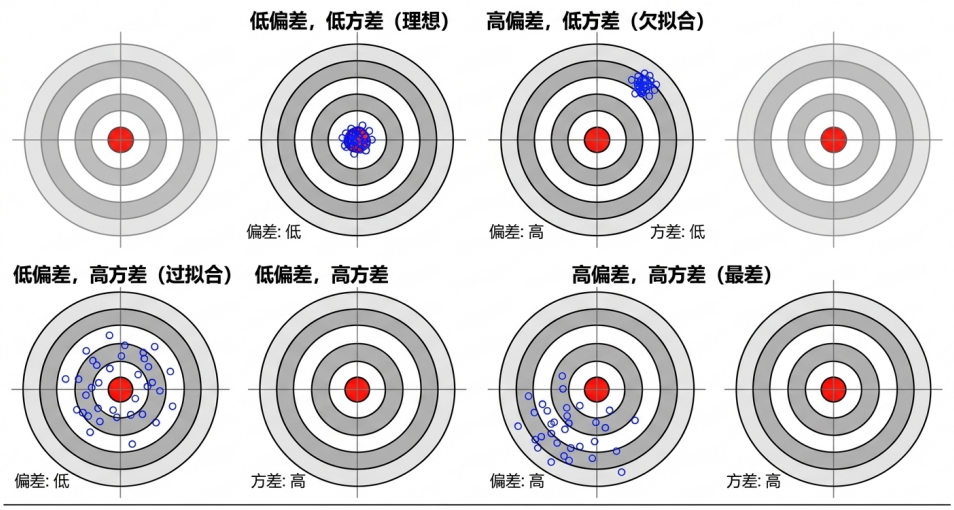

图 5: 偏差-方差权衡靶心示意图

1.5 结论与对策

- **解决欠拟合**：增加训练时间、引入更多特征、降低正则化强度。
- **解决过拟合**：增加训练数据、采用正则化、简化模型结构、提前停止。

2 机器学习流水线

2.1 机器学习流水线全流程详解

机器学习流水线（Machine Learning Pipeline）是将机器学习项目从原始数据到落地应用的标准化、可复用的全流程体系，核心分为三大阶段：**数据准备**、**模型构建**、**模型应用**，每个阶段包含多个关键子步骤，是工业界 AI 项目落地的核心框架。

2.1.1 第一阶段：数据准备

数据是机器学习的基石，高质量数据是模型效果的根本保障，该阶段包含 4 个核心子步骤：

1. **数据收集**：从业务系统、传感器、公开数据集、第三方数据源等多渠道获取原始数据，覆盖任务所需的特征与标签，确保数据的代表性与完整性。

2. **数据清洗**：处理原始数据中的缺失值、异常值、重复值、格式错误等问题，统一数据标准，消除噪声，提升数据质量，是避免模型偏差的关键环节。

3. **特征工程**：对清洗后的数据进行特征提取、特征选择、特征转换、特征归一化/标准化等操作，将原始数据转化为模型可学习的有效特征，直接决定模型的上限。

4. **分离训练集与验证集**：将处理好的数据集按比例（如 7:2:1）划分为训练集、验证集、测试集，训练集用于模型训练，验证集用于调参与模型选择，测试集用于最终效果评估，避免数据泄露与过拟合。

2.1.2 第二阶段：模型构建

模型构建是流水线的核心，基于高质量数据完成模型的训练、验证与优化，包含 5 个核心子步骤：

1. **创建模型**：根据任务类型（分类、回归、聚类等）选择合适的模型架构，如传统机器学习模型（逻辑回归、随机森林、SVM）或深度学习模型（CNN、Transformer），搭

建模型的基础结构。

2. **模型训练**：使用训练集数据输入模型，通过优化算法（如梯度下降）最小化损失函数，迭代更新模型参数，让模型学习数据中的规律。

3. **模型验证与评估**：使用验证集/测试集评估模型性能，选择匹配任务的指标（分类任务用准确率、F1、AUC；回归任务用 MAE、RMSE 等），验证模型的泛化能力。

4. **模型优化**：通过超参数调优（网格搜索、随机搜索、贝叶斯优化）、正则化、集成学习等方法优化模型，解决过拟合/欠拟合问题，提升模型效果。

5. **最终模型**：完成所有优化后，确定最终可部署的模型版本，用于后续业务应用。

2.1.3 第三阶段：模型应用

模型应用是流水线的最终目标，将训练好的模型落地到实际业务中：

$$\textbf{待测数据} \rightarrow \textbf{ML 模型} \rightarrow \textbf{预测结果}$$

将业务中产生的新数据输入训练好的模型，输出预测结果，支撑业务决策，如推荐系统的商品推荐、风控系统的风险识别、计算机视觉的图像识别等，同时需持续监控模型效果，应对数据分布漂移等问题。

2.2 总结

机器学习流水线是 AI 项目落地的核心框架，涵盖数据准备、模型构建与模型应用三大阶段。数据准备是流水线的基石，高质量的数据决定了模型效果的上限；模型构建通过训练、验证与优化将数据转化为可用模型；模型应用则将训练好的模型部署到实际业务中，持续产生价值。掌握标准化的机器学习流水线，是工业界 AI 项目成功落地的关键能力。

3 过拟合与欠拟合及其缓解方法

3.1 引言：偏差与方差的权衡

在机器学习中，模型在训练集与测试集上的表现差异，本质是偏差（Bias）与方差（Variance）的权衡问题：

- 偏差过大 → 模型过于简单，无法学习数据规律 → **欠拟合**
- 方差过大 → 模型过于复杂，学习到噪声 → **过拟合**

3.2 过拟合（Overfitting）

3.2.1 定义

过拟合是指模型在**训练集上表现极好，但在未知测试集上表现很差**的现象。

- 模型不仅学到了数据的真实规律
- 还把训练集中的**随机噪声、异常点**当作通用规律记忆

特点：训练误差极低，但泛化能力更差。

3.2.2 过拟合的处理方法

获得更多训练数据 / 数据增强 增加数据是解决过拟合最有效的方法。

数据增强是在不额外采集真实样本的前提下，通过对现有数据进行合理变换，生成大量新的训练样本，从而扩大数据集规模、降低噪声影响、提升模型泛化能力的方法。在计算机视觉任务中，数据增强是最常用、最有效的缓解过拟合手段之一，常见实现方式包括：

- 随机翻转、镜像：对图像进行水平/垂直翻转，不改变语义信息。
- 随机裁剪：从原图中截取不同区域作为新样本，增强模型的位置鲁棒性。

- 扭曲与形变：对图像进行旋转、缩放、拉伸等几何变换。
- 颜色通道变换：对 R、G、B 三通道添加微小扰动，改变亮度、对比度、饱和度。

数据增强的核心作用是让模型看到更多” 相似但不同” 的样本，减少对单一训练样本的过度依赖，从而抑制过拟合。

- 更多样本可稀释噪声影响
- 计算机视觉常用：翻转、镜像、裁剪、扭曲、颜色抖动

降维

- 手工剔除无关特征
- 使用 PCA 等算法减少特征维度
- 减少网络层数、神经元数量，降低模型容量

正则化（Regularization） 正则化思想：**保留所有特征，但约束参数不要过大，避免模型过度复杂**。

L1 正则化（Lasso） 在原始损失函数后加入权重的**绝对值和**：

$$J_{\mathrm{L1}}(w) = J_0(w) + \lambda \sum_{i=1}^{n} |w_i|$$

作用：产生稀疏权重，自动实现特征选择。

L2 正则化（Ridge 回归） 加入**权重的平方和**：

$$J_{\mathrm{L2}}(w) = J_0(w) + \lambda \sum_{i=1}^{n} w_i^2$$

作用：让权重整体变小、平滑，显著抑制过拟合。

弹性网络（ElasticNet） 结合 L1 与 L2：

$$J_{\text{elastic}}(w) = J_0(w) + \lambda_1 \sum |w_i| + \lambda_2 \sum w_i^2$$

正则化总结 正则化本质是在**拟合训练数据**与**保持模型简单性**之间做权衡。

- 权重大：模型对特征敏感，易过拟合
- 权重小：模型更稳定，泛化能力更强
- L2 正则化：在参数空间形成**圆形约束**，使最优解靠近原点，权重整体缩小
- L1 正则化：形成**菱形约束**，最优解靠近坐标轴，部分权重变为 0
- λ 为正则化系数，越大约束越强，模型越简单

L1 侧重稀疏与特征选择，L2 侧重平滑与防过拟合，弹性网络兼顾两者优势。

早停机制（Early Stopping） 早停是一种在模型训练过程中动态防止过拟合的简单高效策略。随着训练轮数增加，模型在训练集上的误差会持续下降，但在验证集上的误差会先下降、到达最低点后再次上升，这标志着模型开始过拟合。早停机制的核心思想是：

- 在训练过程中持续监控验证集的误差或准确率。
- 当验证集性能不再提升、甚至开始下降时，立即停止训练。
- 保留验证集性能最优时的模型参数，不再继续迭代。

早停无需修改模型结构、不增加计算开销，是深度学习中最常用的抑制过拟合方法之一。

训练过程中持续监控验证集误差：

- 验证误差开始上升时立即停止训练
- 避免模型继续学习训练集噪声

集成学习方法 通过 Bagging、随机森林等方式融合多个模型，降低单一模型过拟合风险。

3.3 欠拟合（Underfitting）

3.3.1 定义

模型过于简单，**训练集与测试集表现都很差**，无法捕捉数据基本模式。

3.3.2 欠拟合的处理方法

1. **添加新特征**特征不足或相关性弱是欠拟合主要原因，可通过特征组合、特征挖掘提升表达能力。
2. **增加模型复杂度**线性模型添加高次项；神经网络增加层数或神经元数量。
3. **减小正则化系数** λ 正则化过强会加剧欠拟合，需降低约束强度。

3.4 总结对比

问题类型	典型表现	核心解决思路
欠拟合	训练误差高、测试误差高	加特征、加复杂度、减正则
过拟合	训练误差低、测试误差高	加数据、正则化、早停、降复杂度

表 2: 过拟合与欠拟合对比

4 偏差与方差

4.1 泛化误差

机器学习模型的**泛化误差**是指模型在未知、全新数据（测试集）上进行预测时所产生的误差，即对从真实数据分布 $P(X,Y)$ 中独立同分布采样的新样本的**期望预测误差**。

已知训练数据集 D 的概率分布，对测试样本 x，令 y 为 x 在数据中的标记，y_D 为 x 的真实标记，$f(x;D)$ 为训练集 D 上学得模型在 x 上的预测输出。以回归任务为例，泛化误差 $E(f;D)$ 可分解为：

$$E(f;D) = \text{bias}^2(x) + \text{var}(x) + \epsilon^2 \tag{1}$$

其中三项分别为：

- **偏差（Bias）：** $\text{bias}(\hat{y}) = E[\hat{y}] - y$，衡量估计量期望值与真实参数值之间的差异，描述模型的**拟合能力**。
- **方差（Variance）：** $\text{var}(x) = E_D[(f(x;D) - \bar{f}(x))^2]$，衡量模型对不同训练集的敏感程度，描述模型的**稳定性**。
- **噪声（Noise）：** $\epsilon^2 = E[(y - y_D)^2]$，数据本身固有的随机误差，是无法通过优化模型来消除的下界。

4.1.1 偏差-方差分解的证明

设学习算法的期望预测为 $\bar{f}(x) = E_D[f(x;D)]$，则：

$$\begin{aligned} E(f;D) &= E_D\left[(f(x;D) - \bar{f}(x))^2\right] + E_D\left[(\bar{f}(x) - y_D)^2\right] \\ &= E_D\left[(f(x;D) - \bar{f}(x))^2\right] + (\bar{f}(x) - y)^2 + E_D\left[(y - y_D)^2\right] \\ &= \text{var}(x) + \text{bias}^2(x) + \epsilon^2 \end{aligned} \tag{2}$$

4.1.2 偏差-方差权衡（Bias-Variance Tradeoff）

偏差与方差通常呈现相互制约的关系：

- 模型**过于简单**（欠拟合）：高偏差、低方差——模型不能很好地拟合训练数据，在所有数据集上都表现较差。
- 模型**过于复杂**（过拟合）：低偏差、高方差——模型过度拟合训练集的随机噪声，泛化能力差。
- 最优模型应在偏差和方差之间取得适当的平衡，使泛化误差最小化。

5 交叉验证：K-Fold 数据集划分方法

5.1 方法简介

K 折交叉验证（K-Fold Cross Validation）是一种常用的模型评估方法，通过充分利用有限数据，得到更稳健的性能估计。

5.2 基本步骤

1. **随机打乱全部数据集**（若数据存在顺序依赖，如时间序列，则不能随机打乱，需用其他方式）。
2. **将数据集等分为 K 个互斥子集**（每个子集称为一个" 折"），每折大小尽量相等。
3. **进行 K 轮训练与验证：**第 i 轮将第 i 折作为验证集，其余 $K-1$ 折合并作为训练集，在训练集上训练模型，在验证集上评估性能。
4. **计算 K 轮评估指标的均值**作为模型的最终性能估计。

每轮训练集和验证集互斥，每个样本恰好被验证一次。

5.3 示例（K=5）

表 3: 5 折交叉验证示例

轮次	验证折	训练折
1	折 1	折 2 + 折 3 + 折 4 + 折 5
2	折 2	折 1 + 折 3 + 折 4 + 折 5
3	折 3	折 1 + 折 2 + 折 4 + 折 5
4	折 4	折 1 + 折 2 + 折 3 + 折 5
5	折 5	折 1 + 折 2 + 折 3 + 折 4

5.4 K 值的选取

- $K = 5$ **或** $K = 10$ 是工程中最常用的选择，在计算代价与估计准确性之间取得良好平衡。

- $K = m$（**留一法，LOOCV**）：每次只留一个样本作验证，估计无偏但计算开销大，适合小数据集。
- K 越大，模型训练数据越多，偏差越小，但方差增大，计算代价也越高。

交叉验证通常用于模型选择（比较不同超参数组合）和模型评估（估计最终模型的泛化性能）。

6 回归问题的评价指标

6.1 引言

在回归问题中，目标是建立预测连续型目标变量的模型。评价指标衡量模型预测值与真实值之间的差异。本节将系统介绍 MSE、RMSE、MAE 和 R^2 四种核心指标。

6.2 基于误差的评价指标

6.2.1 均方误差（MSE）

$$\mathrm{MSE}(y, \hat{y}) = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2 \tag{3}$$

- **优点**：数学性质优良，处处可导，是线性回归等模型损失函数的基础，便于优化。
- **缺点**：对异常值（Outliers）非常敏感，误差被平方放大后极端值影响显著。
- **量纲**： MSE 的量纲是原始数据量纲的平方（如预测房价，单位为" 万元 ^{2}"）。

6.2.2 均方根误差（RMSE）

$$\mathrm{RMSE}(y, \hat{y}) = \sqrt{\mathrm{MSE}} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2} \tag{4}$$

RMSE 是 MSE 的平方根，量纲与原始数据相同，解释性更强。由于是 MSE 的单调变换，模型比较中两者结论一致。同样对异常值敏感。

6.2.3 平均绝对误差（MAE）

$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^{m} |y^{(i)} - \hat{y}^{(i)}| \tag{5}$$

- **优点**：对异常值的鲁棒性比 MSE/RMSE 强，量纲与原始数据相同，直观可解释。
- **缺点**：在零点不可导，使用梯度下降等方法时需特殊处理。

6.3 拟合优度指标：决定系数 R^2

前述绝对误差指标的数值大小与数据范围密切相关，难以跨数据集横向比较。R^2 是无量纲的相对性评价指标。

6.3.1 基本公式

$$R^2(y, \hat{y}) = 1 - \frac{\mathrm{SSE}}{\mathrm{SST}} = 1 - \frac{\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^{m}(y^{(i)} - \bar{y})^2} \tag{6}$$

其中 SSE 为残差平方和，SST 为总平方和（$\bar{y}$ 为真实值均值）。

6.3.2 统计学含义

R^2 代表模型所解释的方差占总方差的比例：

- $R^2 = 1$：模型完美拟合，$\mathrm{SSE} = 0$，解释了 100% 的方差。
- $R^2 = 0$：模型表现等同于始终预测均值 $\bar{y}$ 的朴素模型，没有提供额外信息。
- $R^2 < 0$：模型预测结果甚至劣于直接使用均值预测，通常意味着严重的模型设定错误。

等价形式：$R^2 = 1 - \dfrac{\mathrm{MSE}_M}{\mathrm{Var}(y)}$，即模型相对于” 无信息” 基准模型的改进程度。

6.3.3 调整 R^2

多元线性回归中，增加自变量（即使与 y 无关）会使 R^2 永远不下降，引发过拟合假象。为此引入：

$$R^2_{\mathrm{adj}} = 1 - \frac{\mathrm{SSE}/(m-p-1)}{\mathrm{SST}/(m-1)} \tag{7}$$

其中 p 为自变量个数，只有新增变量真正改善模型时，调整 R^2 才会增加。

6.4 指标选择指南

表 4: 回归评价指标对比

指标	核心特点	适用场景
MSE	对异常值敏感，可导	作为损失函数用于模型训练;数据较” 干净” 时
RMSE	量纲与 y 相同,可解释性强	与预测目标单位一致的性能报告
MAE	对异常值鲁棒，不可导	数据中存在明显异常值时
R^2	无量纲,衡量解释方差比例	不同数据集上模型拟合优度的横向比较

实际项目中建议：结合使用多个指标（如同时报告 RMSE 和 R^2），从绝对误差和相对拟合度两个角度全面评估，并结合业务需求选择侧重点。

7 分类问题评价指标

7.1 引言：为什么需要评价指标

仅知道模型” 预测对了多少” 往往不足以全面评估其性能。例如在癌症筛查中，将健康人误诊为癌症（假阳性）与将病人漏诊为健康（假阴性）的代价完全不同。因此需要一套系统的评价指标来衡量模型在不同维度的表现。

7.2 混淆矩阵：一切指标的基础

混淆矩阵（Confusion Matrix）是一个 2×2 的表格，展示预测结果与真实标签的对应关系：

真实 \预测	**Positive**	**Negative**
Positive	TP（真正例）	FN（假负例）
Negative	FP（假正例）	TN（真负例）

表 5: 混淆矩阵结构

记忆技巧：第二个单词（Positive/Negative）表示**模型的预测结果**。

- **False Positive（假阳性）**：模型预测为正，但实际为负（第一类错误）。
- **False Negative（假阴性）**：模型预测为负，但实际为正（第二类错误）。

7.3 常用评价指标详解

7.3.1 准确率（Accuracy）

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

简单易懂，适合类别均衡的场景。但当类别极不均衡时（如 99% 负例），模型即使全猜负类也能获得 99% 准确率，准确率指标完全失效。

7.3.2 精确率（Precision）与召回率（Recall）

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

- **精确率**：模型预测为正类的样本中，有多少是真正正类，反映” 不误报” 的能力。
- **召回率**：所有真实正类中，模型成功找出了多少，反映” 不漏报” 的能力。

一句话区别：精确率问” 模型说是正例的，到底有多准？”；召回率问” 真实的正例，模型找到了多少？”

7.3.3 F1 分数（F1 Score）

精确率和召回率往往相互制约。F1 分数是二者的调和平均，综合体现两方面表现：

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

F1 取值范围为 $[0, 1]$，值越大表明模型在精确率和召回率之间取得了更好的平衡。

7.4 实例分析：猫狗分类

以 100 张图片（猫 60 只，狗 40 只）为例，模型预测出 50 只猫（其中 40 只是真猫，10 只是狗），得：TP=40, FP=10, FN=20, TN=30。

指标	公式	本例数值
准确率	$(TP+TN)/S$	$70/100 = 0.70$
精确率	$TP/(TP+FP)$	$40/50 = 0.80$
召回率	$TP/(TP+FN)$	$40/60 \approx 0.667$
F1 分数	$2 \cdot P \cdot R/(P+R)$	≈ 0.727

表 6: 猫狗分类实例评价指标汇总

7.5 阈值对 Precision 与 Recall 的权衡

逻辑回归等模型输出的是概率，通过阈值（默认 0.5）将概率转化为类别标签：

- **提高阈值**（如 $0.5 \to 0.8$）：模型更” 保守”，仅预测高置信度样本为正类。FP 减少 $\Rightarrow$ 精确率上升；TP 可能减少 $\Rightarrow$ 召回率下降。
- **降低阈值**（如 $0.5 \to 0.2$）：模型更” 激进”，更多样本被预测为正类。FP 增加 $\Rightarrow$ 精确率下降；TP 增加 $\Rightarrow$ 召回率上升。

遍历所有可能的阈值，以召回率为横轴、精确率为纵轴绘制 **PR 曲线**，曲线下面积（Average Precision, AP）是评价模型排序能力的常用指标。

业务场景决定优化目标：

- **垃圾邮件检测**：提高精确率（宁可漏掉部分垃圾邮件，也不误判正常邮件）。
- **癌症初筛**：提高召回率（宁可误诊，也不漏掉真正的病人，后续可复检剔除假阳性）。

7.6 多分类问题中的宏平均与微平均

在多分类问题中，需对每类分别计算精确率/召回率，再汇总：

- **宏平均（Macro-average）**：先计算每个类别的指标，再取算术平均。平等对待每个类别，适合关注罕见类别的表现。
- **微平均（Micro-average）**：将所有类别的 TP、FP、FN 累加后统一计算。更偏向样本量大的类别，适合整体性能评估。

8 ROC 曲线与 AUC 指标

8.1 传统指标的困境：引入 AUC 的动机

在极度不平衡数据（Class Imbalance）下，准确率会产生严重的" 评估失效"（Accuracy Paradox）。

课堂案例： 1000 张照片的猫狗二分类，猫 990 张，狗 10 张。一个无论输入什么都预测为" 猫" 的模型：Accuracy = 99%，Recall = 100%，Precision = 99%，看似完美，却**完全无法识别任何一只狗**。

这说明在样本极不均衡时，依赖固定阈值的传统指标会受到先验概率的严重干扰，无法客观反映模型对少数类的真实鉴别能力。为此引入 **AUC（Area Under Curve）**。

8.2 ROC 曲线：打破单一阈值的桎梏

8.2.1 两个基础量

二分类评估中的两个核心比例：

$$\mathrm{TPR} = \frac{TP}{TP + FN}, \qquad \mathrm{FPR} = \frac{FP}{FP + TN} \tag{11}$$

- **TPR（真正率/召回率）**：真实正类中，模型成功找出的比例，希望越大越好。
- **FPR（假正率）**：真实负类中，被模型误判为正类的比例，希望越小越好。

8.2.2 ROC 曲线的绘制

逻辑回归等分类模型输出的是概率值，通过不同阈值可得到不同的分类结果。将阈值从 1.0 连续滑动至 0.0，记录每个阈值对应的 (FPR, TPR) 点对，将这些点连接起来即得 **ROC 曲线（Receiver Operating Characteristic Curve）**。

ROC 曲线以 FPR 为横轴，TPR 为纵轴：

- 曲线越靠近**左上角**越好（FPR $= 0$，TPR $= 1$ 为理想点）。
- 曲线接近**对角线**（$y = x$）时，模型区分能力与随机猜测相当。

ROC 曲线描述的是：当分类阈值不断变化时，模型” 多识别出正类” 与” 多错判负类” 之间的整体变化关系，解耦了对单一阈值的依赖。

8.3 AUC 的定义与直观理解

AUC（Area Under the ROC Curve）即 ROC 曲线下的积分面积，是对模型整体分类能力的单一数值概括。

- AUC 越接近 1：模型越擅长把正类排在负类前面。
- AUC ≈ 0.5：模型区分能力与随机猜测相当。
- AUC < 0.5：模型的排序方向可能反了。

8.4 AUC 的概率学本质

AUC 具有极明确的概率论等价解释：

$$AUC = P(\text{score}(x_{\text{pos}}) > \text{score}(x_{\text{neg}}))$$

随机取一个正样本和一个负样本，模型赋予正样本预测概率严格大于负样本的概率。

AUC 并不关注预测概率的绝对数值，而是关注模型能否产生正确的**相对偏序关系**——即正样本的得分是否稳定高于负样本。这使 AUC 成为搜索、广告推荐等排序系统的核心评价指标。

8.5 AUC 的三种计算方法

以 4 个样本（正样本 A, C；负样本 B, D）为例：

样本	预测概率	真实标签	排名（升序）
D	0.2	负类（0）	1
C	0.4	正类（1）	2
B	0.6	负类（0）	3
A	0.8	正类（1）	4

8.5.1 方法一：梯形面积法

通过离散数据点逼近 ROC 曲线下方积分面积：

$$AUC \approx \sum_{i=1}^{n} \frac{(TPR_i + TPR_{i-1}) \times (FPR_i - FPR_{i-1})}{2} \tag{12}$$

8.5.2 方法二：Wilcoxon-Mann-Whitney Test（两两比较法）

基于 AUC 的概率学本质，穷举所有正负样本对进行比较：

$$AUC = \frac{\sum I(P_{\text{pos}}, P_{\text{neg}})}{M \times N}, \quad I(P_{\text{pos}}, P_{\text{neg}}) = \begin{cases} 1.0, & P_{\text{pos}} > P_{\text{neg}} \\ 0.5, & P_{\text{pos}} = P_{\text{neg}} \\ 0.0, & P_{\text{pos}} < P_{\text{neg}} \end{cases} \tag{13}$$

以上例计算：$A(0.8)$ 对 $B(0.6)$ 得 1 分，$A(0.8)$ 对 $D(0.2)$ 得 1 分，$C(0.4)$ 对 $B(0.6)$ 得 0 分，$C(0.4)$ 对 $D(0.2)$ 得 1 分，共 3 分。$AUC = 3/4 = 0.75$。

8.5.3 方法三：正样本 Rank 法（工业界常用）

时间复杂度从 $O(MN)$ 降为 $O((M+N)\log(M+N))$：

$$AUC = \frac{\sum_{i \in \text{pos}} \text{rank}_i - \frac{M(1+M)}{2}}{M \times N} \tag{14}$$

减去常数项 $\frac{M(1+M)}{2}$ 的意义：剔除正样本内部互相排序产生的基准 Rank 和，剩余数值即为" 正样本排在负样本前面的总次数"。

代入本例：$\sum \text{rank}_{\text{pos}} = 2 + 4 = 6$，$\frac{M(1+M)}{2} = 3$，$AUC = (6-3)/(2 \times 2) = 0.75$。三种方法结果完全一致。

8.6 ROC 与 PR 曲线的区别

- ROC 曲线关注 TPR 与 FPR，适合正负样本比较均衡的场景。
- PR 曲线关注 Precision 与 Recall，在类别极不均衡（正样本极少）时更为敏感。
- 极度不平衡数据中，大量 TN 会稀释 FPR，导致 AUC 虚高；此时应转用 PR-AUC 进行严苛评估。

两者并非相互替代，应根据任务特点选择合适的评价方式。

9 ROC 曲线与 AUC 指标——概念与直觉理解

9.1 本部分的主要内容

这一部分主要针对三个问题：

- 什么是 ROC 曲线；
- ROC 曲线下面积为什么叫 AUC；
- 为什么 AUC 还可以用” 排序” 或者” 正负样本距离” 的思路来理解。

ROC 和 AUC 都是在评价一个二分类模型” 区分正类和负类的能力”。

9.2 两个基础量

在二分类问题中，通常会先算两个比例：

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

它们的含义分别是：

- TPR：真正率，也叫召回率，表示” 真正的正类里，有多少被模型找出来了”；
- FPR：假正率，表示” 真正的负类里，有多少被模型误判成正类了”。

我们希望：

- TPR 越大越好；
- FPR 越小越好。

9.3 什么是 ROC 曲线

很多分类模型输出的不是直接的类别，而是一个分数或概率，概率的大小最终决定分类结果，只要我们改变分类阈值，最终的分类结果就会变化，TPR 和 FPR 也会跟着

变化。

把不同阈值下得到的一组组点

$$(\mathrm{FPR}, \mathrm{TPR})$$

画在平面直角坐标系里，并连接起来，就得到 ROC 曲线。

其中：

- 横轴是 FPR；
- 纵轴是 TPR。

所以 ROC 曲线描述的是：当分类阈值不断变化时，模型" 多识别出正类" 和" 多错判负类" 之间的整体变化关系。

9.4 ROC 曲线怎么看好坏

一般来说，ROC 曲线越靠近左上角越好。

因为左上角对应的是：

$$\mathrm{FPR} = 0, \qquad \mathrm{TPR} = 1$$

这表示模型把正类几乎都找到了，同时几乎没有把负类错分成正类。这当然是最理想的情况。

相反，如果 ROC 曲线接近对角线，那么模型的效果就比较一般。对角线大致表示：模型区分正负样本的能力和" 随机乱猜" 差不多。

9.5 什么是 AUC

AUC 是

$$\text{Area Under the ROC Curve}$$

的缩写，意思就是：**ROC 曲线下面积**。

既然 ROC 是一条曲线，那么它下面围起来的面积就可以用一个数来概括模型性能。这个数就是 AUC。

9.5.1 AUC 最直观的理解

- AUC 越接近 1，说明模型越擅长把正类排在前面、负类排在后面；
- AUC 约等于 0.5，说明模型区分能力接近随机；
- AUC 小于 0.5，说明模型的排序方向可能反了。

因此，AUC 可以看成对模型整体分类能力的一个综合评价，而且它不依赖某一个固定阈值。

9.6 为什么 ROC 曲线下面积就是 AUC

ROC 曲线是在坐标系中画出的，横轴是 FPR，纵轴是 TPR，所以曲线下方围成的面积天然就能作为一个整体指标。

也就是说：ROC 给出的是" 变化过程"，AUC 给出的是" 整体总结"。前者是图，后者是数。图更直观，数更方便比较模型。

9.7 为什么 AUC 还可以从排序角度理解

这是 AUC 最常见、也最有用的一个解释：

> **AUC 可以理解为：随机取一个正样本和一个负样本，模型把正样本分数排在负样本前面的概率。**

为什么会有这个说法？因为 ROC 的本质和" 按预测分数从高到低排序" 是连在一起的。

当我们按照模型给出的分数给所有样本排序时：

- 如果正样本大多排在前面，说明模型区分能力强，ROC 曲线就会更靠左上，AUC 也更大；
- 如果正负样本混在一起，排序比较乱，ROC 曲线就会更接近对角线，AUC 也会更接近 0.5。

所以从排序角度看，AUC 不只是一个几何面积，它还在衡量模型” 把正类排在负类前面” 的能力。

9.8 为什么它又能和距离视角对应

如果一个模型学得比较好，那么在它输出的分数空间里：

- 正样本的分数通常会整体偏大；
- 负样本的分数通常会整体偏小。

这相当于说，正负样本在模型的判别结果上被” 拉开了距离”。

拉开的程度越明显：

- 正样本越容易排在负样本前面；
- ROC 曲线越容易向左上方鼓起；
- AUC 就越大。

所以，” 距离更开” 与” 排序更清楚” 本质上是在描述同一件事，只是角度不同。

9.9 和 PR 曲线有什么区别

- ROC 关注的是 TPR 和 FPR；
- PR 曲线关注的是 Precision 和 Recall。

一般来说：

- 当正负样本比较均衡时，ROC 很常用；
- 当类别很不均衡、尤其正样本很少时，PR 曲线往往更敏感。

所以这两种图并不是相互替代的关系，而是看任务特点选择更合适的评价方式。

9.10 总结

- ROC 曲线是在不同阈值下，用 FPR 和 TPR 画出来的曲线。
- ROC 曲线越靠近左上角，模型通常越好。
- AUC 就是 ROC 曲线下面积，是对模型整体区分能力的概括。
- AUC 还可以理解为" 正样本得分高于负样本得分的概率"。
- 从距离角度看，正负样本分得越开，AUC 往往越大。

10 不平衡数据处理

摘要

数据不平衡是机器学习分类任务中极为常见的问题，广泛存在于风控、医疗诊断、工业故障检测、入侵检测等关键场景中。本笔记系统梳理数据不平衡的定义、核心影响及主流解决方法，并补充工程实践中的进阶思路与避坑指南。

10.1 什么是数据不平衡现象？

10.1.1 数据不平衡的定义与量化指标

数据不平衡（Class Imbalance），是指在分类任务中，不同类别的样本数量存在显著比例差异的现象。

- 少数类（Minority Class）：样本占比极低
- 多数类（Majority Class）：样本占比极高

不平衡率定义为：

$$IR = \frac{\text{多数类样本数量}}{\text{少数类样本数量}} \tag{15}$$

当 $IR > 10$ 时，通常被认为是中度不平衡；当 $IR > 100$ 时，属于高度不平衡场景。

典型业务场景包括：

- 金融风控：欺诈交易（少数类）vs 正常交易（多数类）
- 医疗诊断：患病样本（少数类）vs 健康样本（多数类）
- 工业故障检测：故障设备数据（少数类）vs 正常设备数据（多数类）
- 网络入侵检测：攻击流量（少数类）vs 正常流量（多数类）

10.1.2 为什么数据不平衡会对模型造成致命影响？

数据不平衡之所以被称为“关键难题”，核心原因是它会打破传统机器学习模型的学习逻辑，带来三重严重问题：

准确率陷阱 传统分类任务中，准确率（Accuracy）是最常用的指标，但在不平衡数据中完全失效。

模型学习偏差 大多数机器学习模型的优化目标是最小化整体损失，因此会优先学习多数类的特征，导致少数类的决策边界被严重挤压。

评价指标失效 不平衡数据中，我们必须放弃准确率，转而关注针对少数类的评价指标：

- 召回率（Recall）
- 精确率（Precision）
- F1 分数

- AUC-PR 曲线

10.2 处理不平衡数据有哪些常用方法？

10.2.1 数据层面：采样方法

欠采样（Under-sampling） 减少多数类样本，使类别分布趋于平衡。

过采样（Over-sampling） 复制或扩充少数类样本。

SMOTE（合成少数类过采样） 通过插值生成新的少数类样本：

$$x_{new} = x + \lambda \cdot (\hat{x} - x), \quad \lambda \in [0, 1] \tag{16}$$

10.2.2 算法层面：代价敏感学习

修改损失函数，为少数类赋予更高的误分类代价：

$$L = -\frac{1}{N} \sum_{i=1}^{N} [w_1 y_i \log \hat{y}_i + w_0 (1 - y_i) \log(1 - \hat{y}_i)] \tag{17}$$

10.2.3 进阶方法

- Balanced Bagging
- SMOTEBoost
- 使用 Recall / F1 / AUC-PR 作为评价指标

10.3 实践选择指南

场景特点	推荐方法	优点	注意事项
样本量充足，中度不平衡	SMOTE + 代价敏感	效果稳定	避免在噪声数据上使用 SMOTE
样本量极小，高度不平衡	过采样 + 正则化	快速平衡数据	严格监控过拟合风险
多数类样本冗余，高度不平衡	EasyEnsemble	减少训练成本	避免丢失多数类关键信息
少数类误分类代价极高	代价敏感学习	直接引导模型关注少数类	合理设置代价权重

表 7: 不平衡数据处理方法的场景选择指南

10.4 小结

数据不平衡的核心矛盾是：

模型整体优化目标 vs 少数类识别需求

解决思路分为两类：

1. 数据层面：采样方法
2. 算法层面：代价敏感学习

11 Focal Loss 学习笔记

摘要

Focal Loss 由 Lin 等人于 ICCV 2017 / TPAMI 2020 提出，用于解决单阶段目标检测中的极端类别不平衡问题。其核心思想是：降低易分样本的权重，使模型更关注难分样本。

11.1 引言与文献定位

Focal Loss 最初由 Lin 等人在 *Focal Loss for Dense Object Detection* 中提出，会议版发表于 ICCV 2017，期刊版发表于 IEEE TPAMI 2020。两者的方法核心一致：通过修改分类损失，缓解单阶段密集检测中极端的前景/背景类别不平衡问题。

11.2 什么是 Focal Loss，来自哪篇文章？

11.2.1 从二分类交叉熵开始

设标签 $y \in \{0, 1\}$，logit 为 z，预测概率为

$$p = \sigma(z) = \frac{1}{1 + e^{-z}}.$$

定义

$$p_t = \begin{cases} p, & y = 1, \\ 1 - p, & y = 0. \end{cases}$$

标准交叉熵为

$$\mathrm{CE}(p_t) = -\log(p_t).$$

11.2.2 Focal Loss 的定义

Focal Loss 在 CE 前乘上一个难度调制项：

$$\mathrm{FL}(p_t) = -(1 - p_t)^{\gamma} \log(p_t), \qquad \gamma \geq 0.$$

加入类别平衡系数 α_t 后：

$$\mathrm{FL}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t).$$

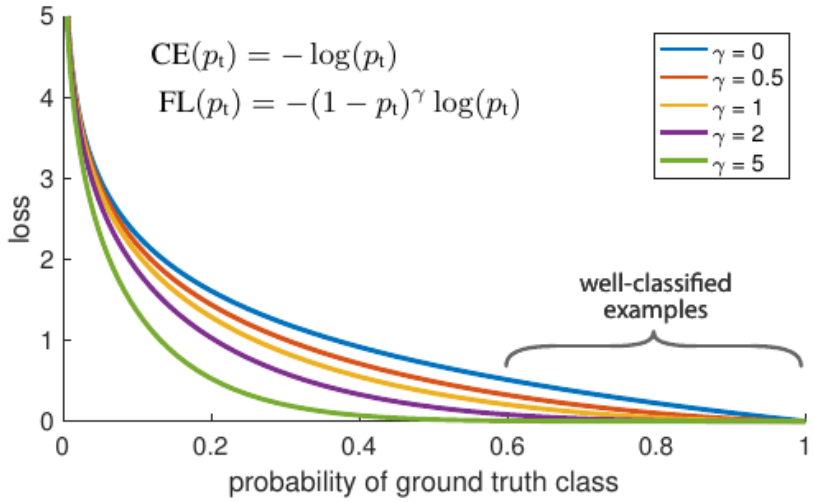

图 6: （原论文 Figure 1）：不同 γ 下的 Focal Loss 曲线。

11.2.3 出处与 RetinaNet 的关系

Focal Loss 与 RetinaNet 一同提出，用于证明：解决 dense detector 的训练不平衡问题后，one-stage detector 也能达到 SOTA。

11.3 原理和 insight 是什么？

11.3.1 核心机制：重新分配训练注意力

Focal Loss 的核心是给每个样本的 CE 损失乘上一个难度权重：

$$\mathrm{FL}(p_t) = w(p_t)\cdot \mathrm{CE}(p_t), \qquad w(p_t) = \alpha_t(1-p_t)^{\gamma}.$$

11.3.2 γ 的作用

$\gamma = 0$ 时退化为带权交叉熵；γ 越大，easy samples 被压得越狠。实践中常用 $\gamma = 2, \alpha = 0.25$。

11.3.3 与 hard example mining 和 robust loss 的区别

- Focal Loss vs OHEM：OHEM 硬删除 easy samples，Focal Loss 连续降权

- Focal Loss vs robust loss：robust loss 压 hard（怕噪声），Focal Loss 压 easy（怕不平衡）

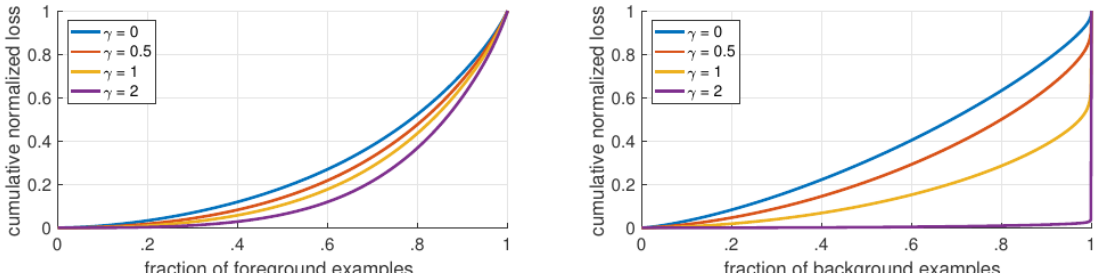

图 7: （原论文 Figure 4）：不同 γ 下正负样本的 normalized loss 分布。

11.4 如何看待难学样本和噪声样本？

11.4.1 难学样本 噪声样本

至少要区分三类样本：

- easy sample：模型已学会
- informative hard sample：有学习价值
- noisy sample：标注错误或歧义

11.4.2 目标检测中噪声更复杂

检测中的 hardness 来源包括：

$$\text{persistent-hard} = \text{真实困难} + \text{标注噪声} + \text{边界不确定} + \text{匹配偏差}.$$

11.4.3 实践判断

场景	判断
类别极不平衡，标注质量高	原始 Focal Loss 适合
大量 easy negatives	Focal Loss 很对症
标签 / 框噪声明显	需引入去噪或不确定性建模
分类分数与定位质量脱节	优先考虑 QFL / GFL / Varifocal Loss

11.5 延伸变体与最新研究方向

11.5.1 代表性变体

方法	核心改动	理解要点
QFL / GFL	连续质量标签	不只问是不是目标，还问框好不好
DFL / GFL	边框回归分布化	不只报一个坐标，而是表达不确定性
GFLV2	定位质量估计	框分布越集中越可信
Varifocal Loss	IoU-aware cls	好框应拿高分
EFL	长尾动态 focusing	稀有类与常见类不应一致对待

11.5.2 近期方向总结

$$\text{difficulty-aware} \longrightarrow \text{quality-aware} \longrightarrow \text{uncertainty / noise-aware.}$$

11.6 小结

Focal Loss 的核心贡献，是将 dense object detection 的类别不平衡问题转化为损失函数层面的样本重加权问题。它有效压低 easy negatives，使模型关注 hard examples。

但它不能区分 hardness 的来源，因此在噪声标签、框偏移、长尾分布等场景下，需要进一步引入 quality-aware、uncertainty-aware 或 noise-aware 的机制。