模型评估与优化

汪小圈

2025-03-17

内容安排

- 为什么需要评估模型?
- 评估指标
  - 分类模型评估指标
  - 回归模型评估指标
- 交叉验证
- 超参数调优
- 模型优化策略
- 正则化方法

为什么需要评估模型?

- **避免过拟合与欠拟合**
  - 模型可能在训练数据上表现很好，但在未见过的数据上表现很差 (过拟合)
  - 模型可能无法捕捉到数据中的基本模式 (欠拟合)
- **选择最佳模型**
  - 需要比较不同模型或模型配置的性能
  - 选择在验证数据上表现最佳的模型
- **了解模型性能**
  - 了解模型在不同情况下的表现
  - 发现模型的优势和局限性
- **指导模型改进**
  - 评估结果可以帮助识别模型的弱点
  - 指导进一步的模型优化

分类模型评估指标 (1)

- **准确率 (Accuracy)**
  - 分类正确的样本数占总样本数的比例
  - 适用于类别分布均衡的数据集
  - Accuracy = $\frac{\text{正确分类的样本数}}{\text{总样本数}}$
- **精确率 (Precision)**
  - 预测为正例的样本中，真正例的比例
  - 关注模型预测正例的准确性
  - Precision = $\frac{TP}{TP+FP}$
    - TP (True Positive): 真正例
    - FP (False Positive): 假正例

分类模型评估指标 (2)

- **召回率 (Recall)**
  - 所有实际正例中，被模型正确预测为正例的比例
  - 关注模型发现所有正例的能力
  - Recall = $\frac{TP}{TP+FN}$
    - FN (False Negative): 假负例
- **F1 分数 (F1-Score)**
  - 精确率和召回率的调和平均值
  - 综合考虑精确率和召回率
  - F1-Score = $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

分类模型评估指标 (3)

- **AUC-ROC 曲线**
  - ROC 曲线：不同阈值下，真正例率 (TPR) 与假正例率 (FPR) 的关系曲线
  - AUC (Area Under Curve)：ROC 曲线下的面积
  - AUC 值越大，模型性能越好
  - 适用于评估二分类模型的排序能力
  - TPR (True Positive Rate): $\frac{TP}{TP+FN}$，等于召回率
  - FPR (False Positive Rate): $\frac{FP}{FP+TN}$

分类模型评估：混淆矩阵

- **混淆矩阵 (Confusion Matrix)**
  - 总结分类模型预测结果的表格
  - 直观展示模型在每个类别上的预测情况
  - 可用于计算精确率、召回率、F1 分数等指标

	预测为正例	预测为负例
实际正例	TP	FN
实际负例	FP	TN

回归模型评估指标 (1)

- **均方误差 (Mean Squared Error, MSE)**
  - 预测值与真实值之差的平方的平均值
  - 对误差进行平方，放大误差较大的样本的影响
  - MSE 越小，模型性能越好
  - $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- **均绝对误差 (Mean Absolute Error, MAE)**
  - 预测值与真实值之差的绝对值的平均值
  - 避免正负误差相互抵消
  - 对异常值不敏感
  - $\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

回归模型评估指标 (2)

- **均方根误差 (Root Mean Squared Error, RMSE)**
  - 均方误差的平方根
  - 与原始数据量纲一致，更易于解释
  - $\mathrm{RMSE} = \sqrt{\mathrm{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
- **R 平方 (R-squared)**
  - 模型解释的方差比例
  - 取值范围为 $[0, 1]$，值越大，模型拟合程度越好
  - $R^2 = 1$ 表示模型完美拟合数据
  - $R^2 = 0$ 表示模型性能与使用均值作为预测值相当
  - $R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$

交叉验证 (1)

- **目的**
  - 更可靠地评估模型的泛化能力
  - 避免模型在特定数据集划分上的偶然性
- **基本思想**
  - 将数据集分成若干份
  - 轮流使用其中一份作为验证集，其余作为训练集
  - 多次训练和评估，取平均性能作为最终评估结果

交叉验证 (2)

- **常用方法**
  - **k 折交叉验证 (k-Fold Cross-Validation)**
    - 将数据集分成 k 份
    - 每次使用 1 份作为验证集，其余 k-1 份作为训练集
    - 重复 k 次，取平均性能
    - 常用 k 值：5 或 10
  - **留一交叉验证 (Leave-One-Out Cross-Validation, LOOCV)**
    - k 折交叉验证的特殊情况，k 等于样本总数
    - 每次只使用一个样本作为验证集
    - 计算成本高，适用于小数据集
  - **分层 k 折交叉验证 (Stratified k-Fold Cross-Validation)**
    - 保证每个 fold 中各类别样本比例与原始数据集相同
    - 适用于类别不平衡的数据集

超参数调优 (1)

- **超参数 (Hyperparameters)**
  - 模型训练前需要手动设置的参数
  - 例如：学习率、正则化系数、决策树的最大深度等
  - 不同于模型参数，超参数不是通过训练优化的
- **目的**
  - 找到最佳的超参数组合
  - 使模型在验证集上获得最佳性能

超参数调优 (2)

- **常用方法**
  - **网格搜索 (Grid Search)**
    - 预先定义超参数的候选值
    - 穷举所有可能的超参数组合
    - 评估每种组合的性能，选择最佳组合
    - 优点：全面；缺点：计算成本高
  - **随机搜索 (Random Search)**
    - 在预定义的超参数空间中随机采样
    - 通常比网格搜索更高效
    - 适用于超参数空间较大的情况
  - **贝叶斯优化 (Bayesian Optimization)**
    - 建立超参数与模型性能之间的概率模型
    - 根据该模型智能选择下一组超参数进行评估
    - 更高效地找到最佳超参数组合

模型优化策略 (1)

- **特征工程 (Feature Engineering)**
  - **特征转换**
    - 标准化、归一化、对数变换等
    - 使特征更符合模型假设
    - 对类别特征进行编码（独热编码、标签编码等）
  - **特征组合**
    - 将多个特征进行组合，生成新的交叉特征
    - 捕捉特征之间的交互关系
    - 例如：年龄与收入的乘积
  - **特征选择**
    - 选择最相关的特征子集
    - 去除冗余或不相关的特征
    - 降低模型复杂度，提高泛化能力

模型优化策略 (2)

- **模型选择 (Model Selection)**
  - **模型比较**
    - 尝试不同的机器学习模型（线性模型、树模型、神经网络等）
    - 在同一数据集上评估不同模型的性能
    - 选择性能最佳的模型
  - **模型融合**
    - 将多个不同模型的预测结果进行融合
    - 获得更好的预测性能
    - 例如：stacking、blending 等集成方法

模型优化策略 (3)

- **集成学习 (Ensemble Learning)**
  - **Bagging**
    - 通过 bootstrap 采样创建多个训练集
    - 在每个训练集上训练一个基学习器
    - 将多个基学习器的预测结果平均或投票
    - 例如：随机森林 (Random Forest)
  - **Boosting**
    - 迭代训练基学习器，每个基学习器纠正前一个的错误
    - 将多个基学习器加权组合
    - 例如：梯度提升树 (GBDT)、XGBoost、LightGBM
- **数据增强 (Data Augmentation)**
  - 通过对训练数据进行变换，增加训练数据的多样性
  - 提高模型的泛化能力
  - 例如：图像旋转、平移、缩放；文本同义词替换；音频添加噪声等

正则化方法 (1)

- **L1 正则化 (Lasso Regularization)**
  - 添加模型权重的 L1 范数惩罚项
  - 使权重稀疏化，有助于特征选择
  - 可以将一部分权重压缩为 0
  - $\text{Loss}_{regularized} = \text{Loss}_{original} + \lambda \sum_i |w_i|$
- **L2 正则化 (Ridge Regularization)**
  - 添加模型权重的 L2 范数惩罚项
  - 减小模型权重，使模型更平滑
  - 权重趋向于变小，但不会变为 0
  - $\text{Loss}_{regularized} = \text{Loss}_{original} + \lambda \sum_i w_i^2$

正则化方法 (2)

- **Elastic Net**
  - 结合 L1 和 L2 正则化的方法
  - 既可以进行特征选择，又可以减小模型权重
  - $\text{Loss}_{regularized} = \text{Loss}_{original} + \lambda_1 \sum_i |w_i| + \lambda_2 \sum_i w_i^2$
- **Dropout**
  - 在训练过程中随机将一部分神经元的输出置为 0
  - 强制网络学习更鲁棒的特征表示
  - 减少神经元之间的共适应性
  - 常用于深度神经网络

正则化方法 (3)

- **Early Stopping (提前终止)**
  - 监控验证集上的性能指标
  - 当验证集性能不再提升或开始下降时，停止训练
  - 简单易用，无需额外计算
  - 有效防止过拟合
- **Batch Normalization (批量归一化)**
  - 对神经网络每一层的输入进行归一化
  - 加速模型训练，提高训练稳定性
  - 减轻内部协变量偏移问题
  - 具有一定的正则化效果
- **模型剪枝 (Pruning)**
  - 决策树剪枝：剪去不必要的节点
  - 神经网络剪枝：移除不重要的连接或神经元
  - 减小模型复杂度，提高模型效率

总结

- 模型评估是机器学习流程中至关重要的一步
  - 了解模型性能
  - 选择最佳模型
  - 指导模型改进
- 不同的任务需要选择不同的评估指标
  - 分类任务：准确率、精确率、召回率、F1 分数、AUC 等
  - 回归任务：MSE、MAE、RMSE、R^2 等
- 交叉验证和超参数调优可以提高模型的泛化能力
- 模型优化是一个迭代过程，需要尝试不同的策略
  - 特征工程
  - 模型选择
  - 集成学习
  - 正则化方法