COMP 551 Lecture 4 - Model evaluation 2 (M1.3)

Junji Duan

2024/1/17

Today's Outline

- Objectives
  - ross-validation
  - Method comparisons
  - Precision-recall and F1-score

Ross-validation

0.1 K-fold Cross Validation

K 折交叉验证（K-fold Cross Validation）是一种在机器学习中常用的模型评估方法, 尤其适用于数据量不是很大的情况下。这种方法可以有效地利用有限的数据来评估模型的性能, 并减少模型在不同数据集上的性能波动，提高模型泛化能力的评估。

0.1.1 K 折交叉验证的步骤和原理

1. 随机划分数据:

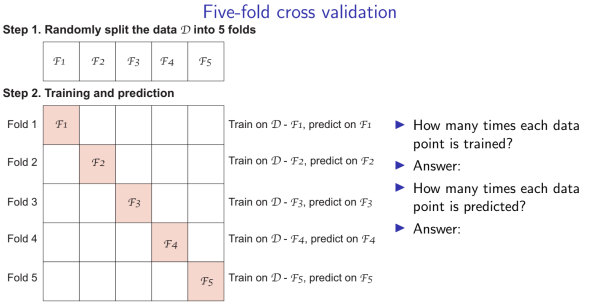
- 将整个数据集 $ | mathcal $\{D\}$ $ 均匀随机地分割成 K 个子集（称为“折”，fold）。每个子集尽量保持数据分布的一致性。例如，在五折交叉验证中，数据被分为五个相等的部分。

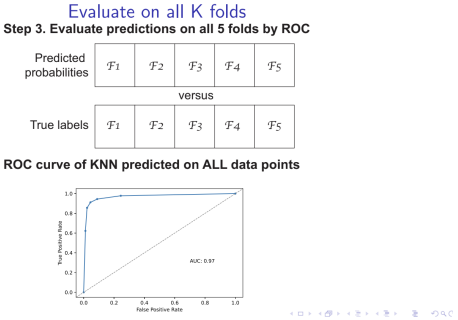
2. 循环训练和预测:

- 在 K 次的训练/测试迭代中, 每次迭代选择一个不同的折作为测试集, 其余的 K-1 折合并作为训练集。
- 这样, 每个数据点正好会被用作测试集一次, 而作为训练集 K-1 次。这也回答了表中“每个数据点被预测的次数”这一问题, 答案是: 每个数据点恰好被预测 1 次。

3. 评估和整合结果:

- 在每一折的测试集上得到的预测结果被用来评估模型的性能, 通常通过计算诸如准确率、召回率、ROC 曲线等指标。
- 最后, 这些指标通常会被平均或以其他方式整合, 以得出模型整体的性能评估。





0.1.2 为什么使用 K 折交叉验证

- 更好地利用数据：与简单的训练/测试分割相比，K 折交叉验证通过在每个可能的训练集上训练模型并在相应的测试集上测试, 使得所有的数据都被用作了训练和测试, 最大化了数据的使用效率。
- 减少偶然性：单一的训练/测试分割可能因数据分割方式的偶然性而导致模型评估结果的不稳定。K 折交叉验证通过平均多次试验结果, 减少了这种偶然性对评估结果的影响。
- 适用于不同模型: 这种方法适用于各种不同的机器学习模型, 使其成为评估模型性能的一个通用方法。

0.1.3 K 折交叉验证的局限性

- 计算成本：进行 K 次训练和测试显然比单次划分更耗时，尤其是在数据集较大或模型复杂时。
- 数据划分的敏感性：虽然减少了偶然性，但最终结果仍可能受到原始数据划分方式的影响，尤其是在数据不均衡时。

通过 K 折交叉验证，我们可以得到一个关于模型在未知数据上表现的更为准确和稳定的评估，这对于模型的选择和优化具有重要意义。

Method comparisons

0.2 常见的机器学习方法

1. K-最近邻 (KNN):

- 这是一种基于实例的学习，模型简单地查找训练集中与新实例最近的 K 个点，并基于这些邻近点的标签通过多数投票或平均来预测新实例的标签。

2. 决策树分类器 (DT):

- 决策树是通过递归地将数据集分割成越来越小的子集而构建的。每个分割都是基于一个使得目标变量的不纯度（例如基尼不纯度或熵）最小化的特征。

3. 逻辑回归 (LR):

- 逻辑回归是一种统计模型，用于二分类问题。它使用逻辑函数将线性回归的输出映射到 0 和 1 之间的概率。

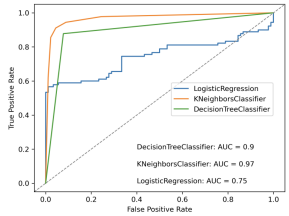
0.3 比较方法: ROC 曲线与 AUC

- ROC 曲线（Receiver Operating Characteristic Curve）是一个重要的评估指标，用来展示在不同阈值设置下，模型的真正率 (TPR) 和假正率 (FPR) 的关系。
- AUC (Area Under the Curve) 是 ROC 曲线下的面积，用以量化模型整体的分类性能。AUC 的值越接近 1，说明模型的性能越好。

0.4 方法性能比较

ROC curves and AUC for all of the four methods

- KNN (K=5) performs the best with 0.97 AUC
- DT achieves 0.85 AUC
- LR did worse (AUC = 0.73) because our data are not linearly separable
- In contrast, DT and KNN are non-linear methods



- $KNN(K=5)$ 的 AUC 为 0.97，显示出最好的性能，这可能是因为 KNN 作为一种非线性方法，能够更好地处理复杂的数据边界。
- 决策树的 AUC 为 0.85，也是一种表现不错的非线性方法，能够通过构建分支处理数据。
- 逻辑回归的 AUC 为 0.73，表现较差，这可能是因为数据在特征空间中不是线性可分的，而逻辑回归假设数据的边界是线性的。

1.K-最近邻 (KNN)

- 性质:
  - 基于实例的学习，使用距离度量来找出最近的 K 个邻居。
  - 非参数方法，即不对数据分布做任何假设。
- 优点:
  - 理解和实现都非常简单。
  - 灵活性高，适应性强，因为它不依赖于数据的先验假设。
  - 可以很好地处理多分类问题。
- 缺点:
  - 随着数据量的增加，计算成本和存储成本极高，因为需要存储全部数据。
  - 对噪声和非相关特征敏感。
  - 需要确定合适的 K 值和距离度量。
- 应用:
  - 推荐系统（如电影或商品推荐）
  - 手写识别
  - 图像识别和视频识别

2. 决策树 (DT)

- 性质:
  - 通过逐步将数据集分割成越来越小的子集来构建树结构。

  - 每个节点代表一个属性上的决策, 每个分支代表一个决策结果, 叶节点代表最终的类别。优点:
  - 易于理解和解释, 树形图很直观。
  - 不需要对数据进行太多预处理, 不需标准化数据。
  - 能够处理数值型和类别型数据。

- 缺点:
  - 易过拟合, 特别是当树太深或数据较少时。
  - 对于某些类型的参数变化很敏感, 可能导致树结构大幅改变。
  - 不支持在线学习, 随着数据的更新可能需要重新构建树。

- 应用:
  - 信用评分
  - 医疗诊断
  - 客户分类

3. 逻辑回归 (LR)

- 性质:
  - 适用于二分类问题。
  - 基于概率的线性分类器, 输出变量的对数几率是输入变量的线性组合。

- 优点:
  - 实现简单, 广泛用于工业问题。
  - 训练速度快。
  - 输出可以解释为概率。

- 缺点:
  - 预测性能依赖于数据的线性可分性。
  - 对模型中的独立变量有较强的假设。
  - 不太适合非线性问题, 需要转换输入或扩展特征。

- 应用:
  - 邮件垃圾分类
  - 疾病预测 (如糖尿病预测)
  - 金融欺诈检测