浙江大學
ZHEJIANG UNIVERSITY

人工智能算法与系统-2024 秋

成人死亡率预测

姓名 刘京宗

学号 22451040

学院 软件学院

2024 年 10 月 18 日

成人死亡率预测

1 算法描述

1.1 随机森林回归算法概述

随机森林（Random Forest）是一种集成学习算法，属于 Bagging 方法的代表，它由多棵决策树构成，最终的预测结果通过这些树的平均值来决定。它可以用于分类问题（Random Forest Classifier）和回归问题（Random Forest Regressor），本文主要讨论随机森林回归算法。

随机森林回归的主要特点：

抗过拟合能力强：通过多个树的集成，随机森林可以降低单棵决策树的过拟合风险。具有较好的预测精度：由于它是基于多棵树的预测平均值，所以其预测精度比单棵决策树要高。能处理高维数据和缺失值：它能有效处理带有缺失值和噪声的数据，也不需要对数据进行标准化处理。

1.2 随机森林回归算法的历史演变

决策树的引入：决策树（Decision Tree）算法最早可以追溯到 20 世纪 60 年代，最著名的早期决策树算法是 ID3 算法（1986 年提出），之后又有了 C4.5 和 CART 等优化版本。决策树算法虽然简单易理解，但有一个明显的缺点：容易过拟合，尤其是在处理复杂数据集时。

Bagging（Bootstrap Aggregating）技术的提出：Bagging 技术由 Leo Breiman 于 1996 年提出，它通过构建多个模型（例如多棵决策树）并通过随机抽样（Bootstrap）构建不同的训练集，使模型的泛化能力得到提升。

随机森林算法的提出：随机森林算法是 Leo Breiman 于 2001 年提出的，随机森林不仅使用了 Bagging 技术，还在每棵树的训练过程中引入了特征随机选择，即在每次分裂时，随机选择特征子集来构建树。这样增强了模型的鲁棒性，进一步避免了过拟合。

1.3 随机森林回归的算法流程

1. 训练流程：以下是随机森林回归的主要步骤：

数据集的准备：

输入数据集为 $(X_1, y_1), (X_2, y_2), ..., (X_n, y_n)$，其中 X_i 为特征向量，y_i 为目标值。Bagging 思想：

从训练数据集中使用 Bootstrap 方法随机有放回地抽取多个子样本集，每个子样本集大小与原数据集相同，且允许重复采样。不同的子样本集用于训练不同的决策树。构

建决策树：

对于每一个子样本集，训练一棵决策树。在每个树的节点分裂时，随机选择一部分特征，而不是用所有特征进行最佳分割。这一过程使得树之间的差异性增大，降低模型过拟合的风险。回归预测：

对新样本进行预测时，将新样本输入到每棵决策树中，并计算所有树的预测值的平均值，作为最终的预测结果。

2. 预测流程：对于新的输入数据，随机森林中的每棵决策树都将产生一个预测值，最终随机森林会将这些预测值进行平均，作为最终的回归结果。

3. 算法流程图：下面是随机森林回归的流程图简述：

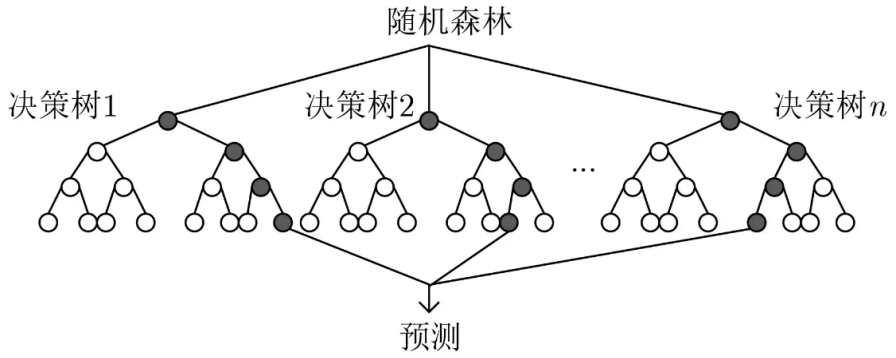

图 1: 随机森林回归的流程图

输入数据集：输入数据包括特征向量 X 和目标值 y。创建多棵决策树：通过 Bootstrap 抽样生成不同的子样本集。对每个子样本集，训练一棵决策树，且在每次分裂节点时，随机选择部分特征用于分裂。预测阶段：对于新的数据点，使用每棵决策树进行预测，输出该数据点的多个预测值。计算平均值：将所有决策树的预测值进行平均，得到最终的回归预测结果。

2 算法性能分析

在本项目中，使用随机森林算法来预测成年人死亡率。我们通过对模型的性能进行定量评估，使用了均方误差（MSE）和决定系数（R^2）等常用指标来分析算法的表现。以下结合数据集的具体信息与实验结果，对算法的性能进行详细分析。

数据集特征原始数据维度：data.shape = (2336, 21)，表明数据集包含 2336 条记录和 21 个特征，这些特征可能涉及到健康状况、社会经济因素等方面。处理后的数据维度：data_norm.shape = (2336, 14)，经过特征选择或数据预处理后，特征数量减少为 14 个。这可能是通过特征工程、去除不相关特征、填补缺失值等方法得到的。训练时间训练时间：2.757 秒。这表明模型在数据集上的训练速度较快，这归因于数据量相对适中以及随机森林可以并行构建多棵树。虽然随机森林算法通过构建多个决策树进行预测，计算

```python
# 计算评估指标
r2 = r2_score(label, y_pred)
mse = mean_squared_error(label, y_pred)

print("MSE is {}".format(mse))
print("R2 score is {}".format(r2))
```

```
data.shape (2336, 21)
data_norm.shape (2336, 14)
type(data_norm) <class 'pandas.core.frame.DataFrame'>
训练时间: 2.7572522163391113 秒
data.shape (2336, 21)
data_norm.shape (2336, 14)
type(data_norm) <class 'pandas.core.frame.DataFrame'>
MSE is 907.0747625428082
R2 score is 0.9416548517534933
```

图 2: Enter Caption

复杂度比单棵决策树高，但其训练速度仍然较为可接受。随着树的数量（n_estimators）增加，训练时间会有所延长，但本次实验中的 2.757 秒是一个相对合理的时间。

模型性能评估均方误差（MSE）：

MSE = 907.0747。均方误差衡量的是模型预测值与真实值之间的平方差的平均值，数值越低，模型的预测效果越好。这个结果表明，模型的误差较低，预测的成年人死亡率与实际值非常接近。具体来说，MSE 约为 907，表示在每 1000 个成年人中，模型的预测误差大约在 9 个人左右（以平方误差的角度衡量）。这一数值对比于成年人死亡率数据规模是较为合理的误差范围。

决定系数（R^2）：

$R^2 = 0.9417$。R^2 用来衡量模型对目标变量的解释能力，范围在 0 到 1 之间。0 表示模型没有任何解释能力，而 1 表示模型完美拟合数据。在本次实验中，R^2 得分为 0.9417，说明随机森林模型能够解释约 94.17% 的数据波动，这表明模型拟合效果非常好，说明所选择的 14 个特征与目标变量之间存在很强的相关性。高 R^2 得分也意味着模型在回归任务中表现优异，它能够在给定数据集上进行高精度的预测。

3 算法进一步研究展望

随机森林已经是一个广泛应用的强大算法，但仍有改进和扩展的空间：

1. 混合模型：可以将随机森林与其他算法结合，如梯度提升决策树（GBDT）或深度学习模型，从而提升性能。

2. 超参数优化：通过更加智能的超参数优化技术，如贝叶斯优化或随机搜索，可以

进一步提高随机森林的预测性能。

3. 处理高维数据：在高维数据场景下，使用降维技术如主成分分析（PCA）结合随机森林，能够有效减少特征数目并保持预测准确性。

4. 并行计算：通过分布式计算技术，如 Hadoop 或 Spark，加速随机森林的训练和预测过程，特别适用于大规模数据集。

总的来说，随机森林在当前的机器学习应用中已经表现出了强大的适应性和稳定性，未来通过结合更多的技术手段，其性能还可以进一步提升。