题目 集成学习AdaBoost和Random Forest

姓名 吴紫航 学号 171860659 邮箱 [401986905@qq.com](mailto:401986905@qq.com) 联系方式 call: 18956668797

(南京大学 计算机科学与技术系, 南京 210093)

# 实验目的

(1) 实践集成学习的两大经典想法：”Boosting” and “Bagging”

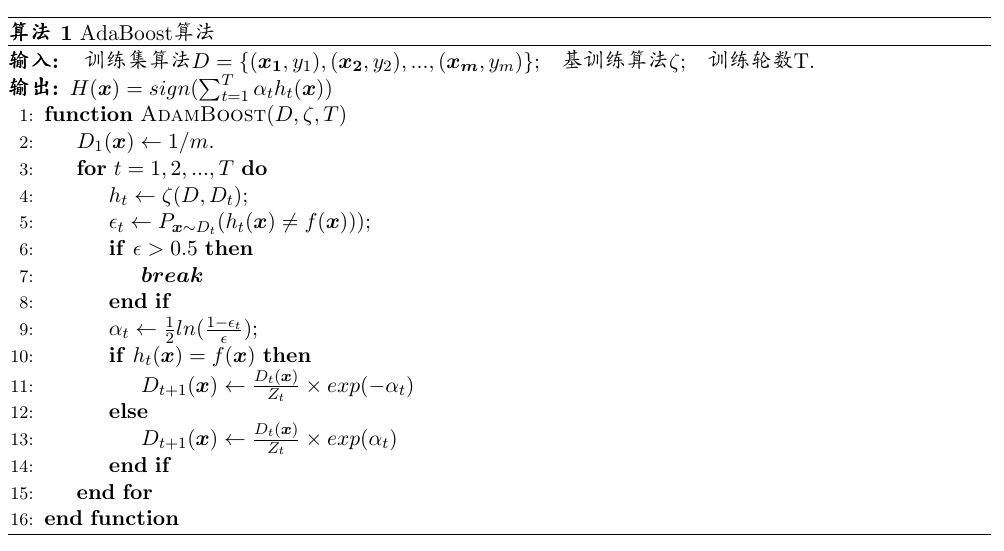
(2) 理解并掌握AdaBoost和Random Forest两个算法的原理和实现

(3) 加强对传统机器学习框架流程的掌握，如模型构建、交叉验证、超参数训练等过程

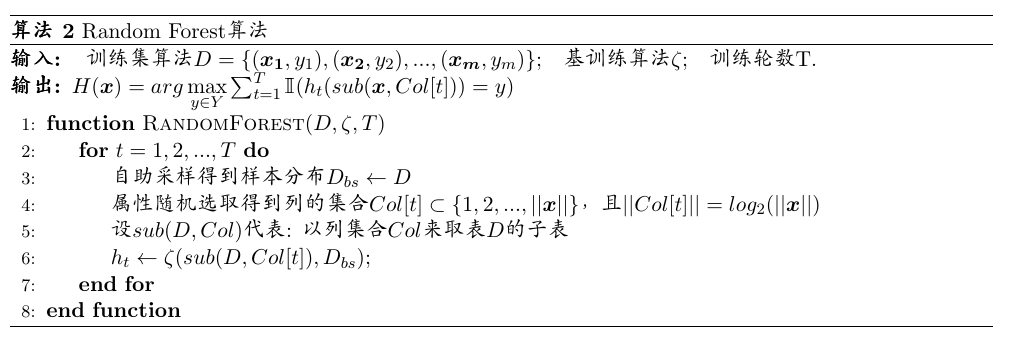
# 算法理论

*本节主要介绍AdaBoost和Random Forest算法层面的伪代码*

## AdaBoost



## Random Forest



# 实现过程

*本节主要介绍代码实现过程主要流程和一些细节*

## 结构配置

|  |  |
| --- | --- |
| **文件** | **内容** |
| AdaBoost.py | AdaBoost代码 |
| RandomForestMain.py | RandomForest代码 |
| adult.data | adult训练集 |
| adult.names | adult数据集介绍 |
| adult.test | adult测试集 |
| ML2020\_PS5.tex | 习题解答latex版本 |
| ML2020\_PS5.pdf | 习题解答pdf版本 |
| report.pdf | 你正在读的报告 |

## 模块介绍

|  |  |
| --- | --- |
| **模块/方法** | **功能** |
| InputData(filePath) | 根据路径获取数据，删除缺失数据的行，并把离散字符串数据映射到整数的方法 |
| AdaBoostModel | AdaBoost模型定义，内置训练、测试以及基于AUC的最佳轮次获取接口 |
| RandomForestModel | RandomForest模型定义，内置训练、测试以及基于AUC的最佳轮次获取接口 |

*注：运行需要安装的py包有: pandas, numpy, sklearn*

## 数据读入

(1) 利用自定义的inputData模块读入adult.data数据, 数据格式为DataFrame

(2) 放弃使用adult.test数据, 把adult.data按3:1 随机划分为trainingDataSet和testDataSet

(3) 把trainingDataSet随机分为5等份, 方便之后进行五折交叉验证

(4) DataFrame数据结构在拆分后，需要重置索引, 调用reset\_index接口即可

## 训练模型

### AdaBoost

*在AdaBoostModel.train()内实现模型的训练算法。主要思想参照2.1的算法即可。这里整理一些细节：*

(1) 单个的弱学习器采用的是单层的**决策树(桩)**, 保证其学习能力较弱

(2) 对于样本的分布，原本采用的方法是直接在fit接口中给参数sample\_weight传值, 但模型训练过程经常发生错误率大于0.5的情况，故现在改进为处理样本分布的方法是**等量加权采样**，这样可以进行重启动(重采样)防止迭代太早结束。

(3) 对于算法2.1的第9行,若错误率为0会导致分母为0, 因此参考**拉普拉斯修正**的思想, 分母修正为max{e,1e-16}

(4) 每轮学好一个新的弱学习器后，都把最新的集成模型在验证集上进行验证，并保存轮次-AUC的结果

### Random Forest

*在RandomForestModel.train()内实现模型的训练算法。调用了RandomForestClassifier接口。这里整理一些细节：*

(1) 使用**自助采样法**保证单学习器之间的行差异性(bootstrap=True)

(2) 使用属性随机抽取保证单学习器之间的列差异性(max\_features=’log2’)

(3) 按可工作的核心数进行**并行化**(n\_jobs=-1)

(4) 由于调包的缘故, 每轮会重新训练完整的森林, 为节约训练开销，对于轮次i, 森林大小设为

## 交叉验证

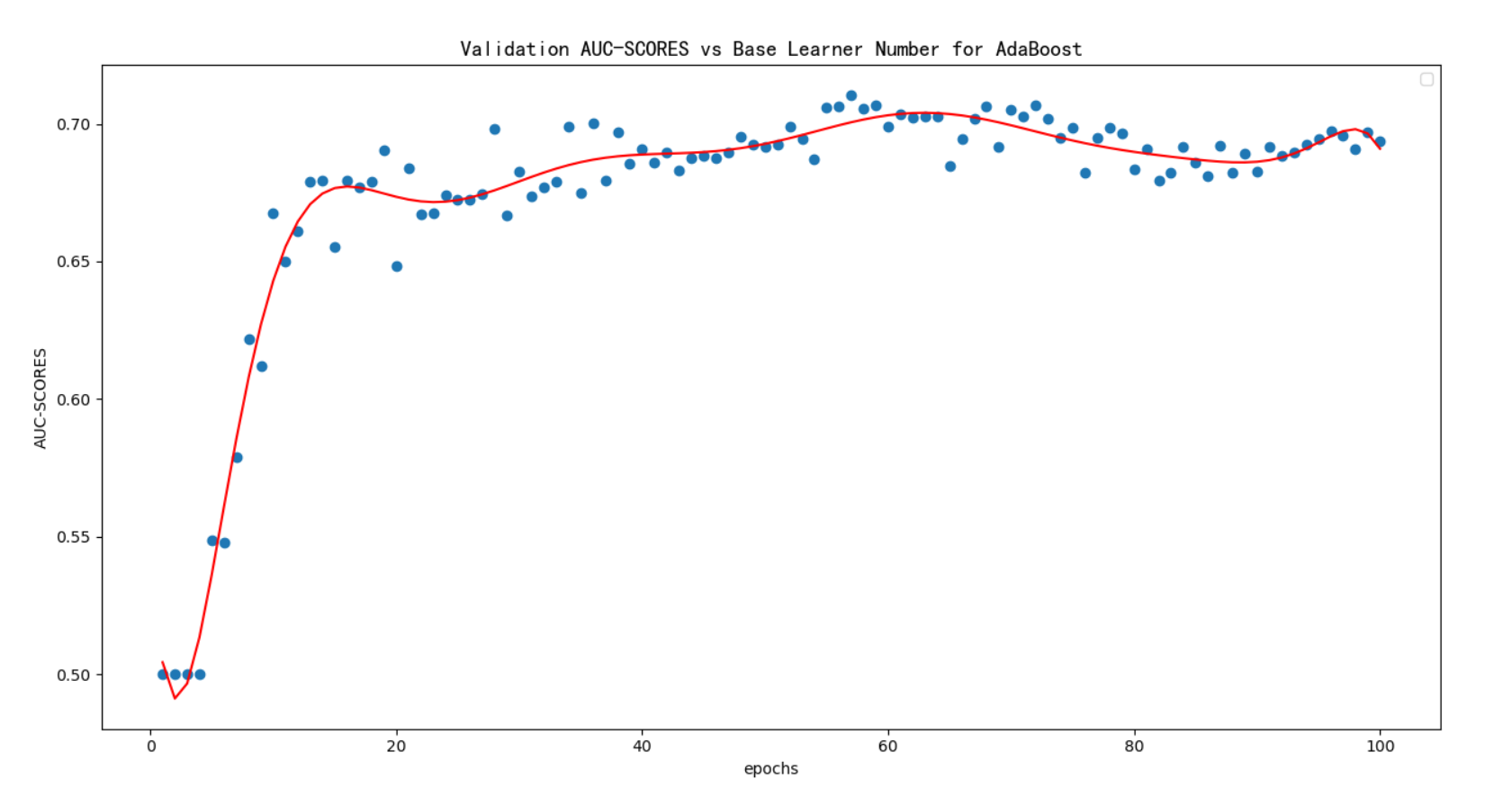


Figure 1 五折交叉验证的AUC vs Base Learner Number for AdaBoost

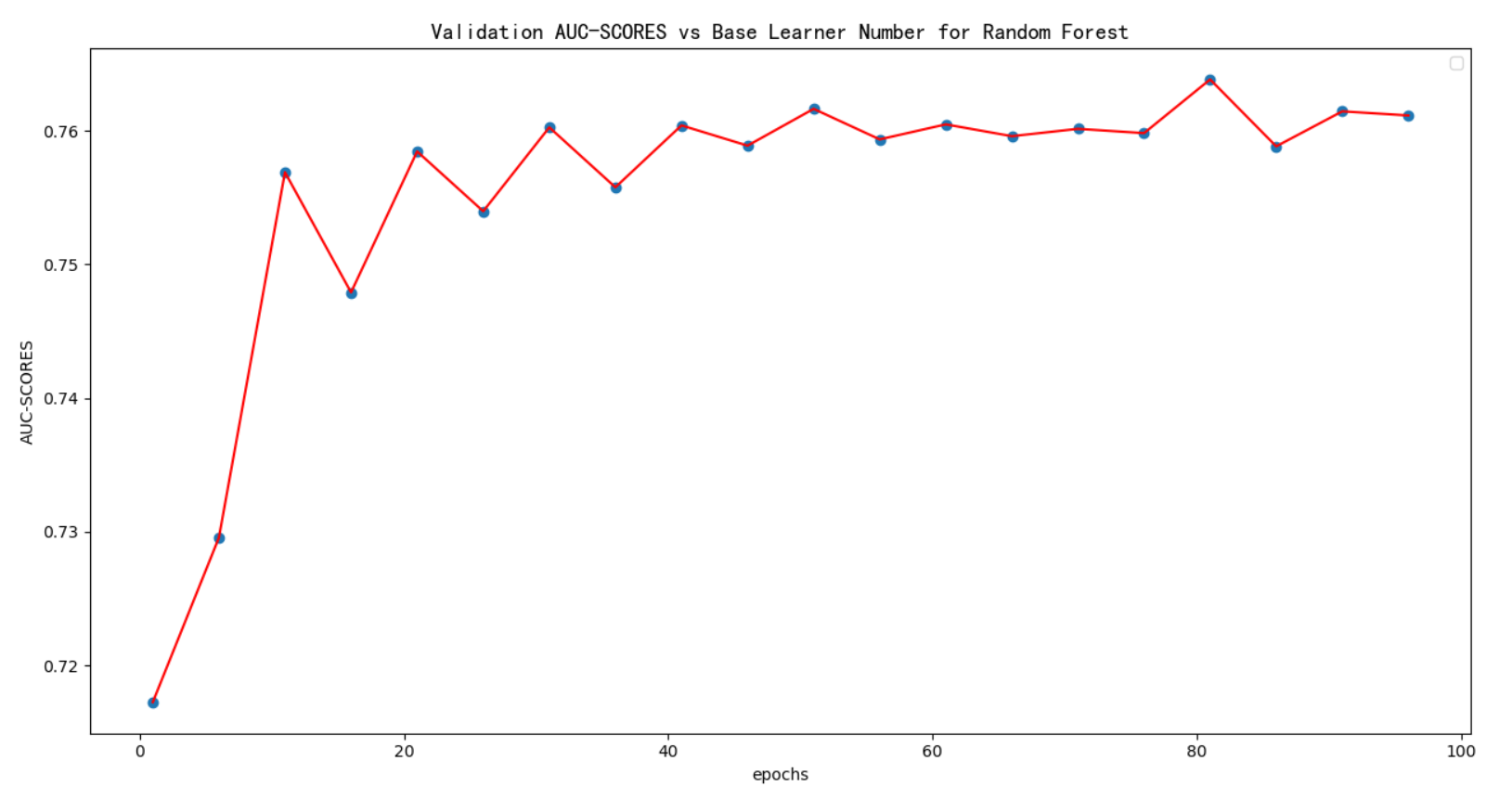


Figure 2 五折交叉验证的AUC vs Base Learner Number for Random Forest

*由五折交叉验证的AUC-epochs曲线, 对于两种算法，最理想的训练轮次分别为57和81*

## 测试结果

*以下给出用最佳轮次进行训练后的模型, 在测试集上的运行结果, 并用决策树桩(弱学习器)进行对比*

***由于是随机算法，每次的运行结果可能会有差异***

|  |  |  |  |
| --- | --- | --- | --- |
| **算法** | **最佳轮次** | **测试集准确率** | **测试集AUC-SCORES** |
| 决策树桩 | \ | 74.58 % | 50.00 % |
| AdaBoost | 57 | 81.74 % | 75.01 % |
| Random Forest | 81 | 84.58 % | 77.06 % |

注：我们对以上结果进行适当分析：

Adult数据集的标签分为0和1，即年收入是否大于50万。这是一个**非常不均匀**的样本分布，我们知道年收入大于50万的肯定明显较少。事实上也确实如此：adult数据集中1标签的个数只有0标签个数的1/3。因此哪怕一个训练器**将所有样本都归类为标签0也将获得75%左右的准确率。**但是这没有任何意义，因此本次实验选取的性能指标是AUC，即考察：**集成后的学习器，比较于单个的学习器，是否在AUC性能上具有提升**。这意味着年收入大于50万的人群也能有足够机会被训练器给识别出来。

# 参考资料

[1] **KNN算法对adult数据集预测**

参考了博客中数据的读取到DataFrame的方法, 以及预处理方法(删除缺失数据、字符离散值映射到整数)

<https://blog.csdn.net/qq_43745026/article/details/107219361>

[2] **pandas dataFraome重置索引**

参考reset\_index接口的使用,用于在划分数据集后重置索引

<https://blog.csdn.net/weixin_39223665/article/details/97397404>

[3] **决策树分类器scikit-learn的使用**

参考了单决策树的接口使用方法和参数说明

<https://zhuanlan.zhihu.com/p/40968625>

[4] **Pandas中loc和iloc函数用法详解**

参考了iloc的使用方法，用于分离特征列和标签列

<https://blog.csdn.net/W_weiying/article/details/81411257>

[5] **pandas.DataFrame.sample 随机选取若干行**

参考了sample接口用于实现抽样(AdaBoost中加权抽样改分布,Random Forest中随机抽样)

<https://blog.csdn.net/zhengxu25689/article/details/87347700>

[6] **sklearn中决策树重要参数详解**

参考了class\_weight参数,因为原本想进行类别均衡, 但发现弱训练器不需要良好的AUC(突出集成学习可以改善AUC), 故最后放弃使用该参数

<https://www.cnblogs.com/juanjiang/p/11003369.html>

[7] **python中train\_test\_split()函数划分训练集、测试集数据**

参考了train\_test\_split()方法用于划分训练集和测试集

<https://blog.csdn.net/peachlychee/article/details/102897919>

[8]**sklearn中的train\_test\_split方法**

参考了train\_test\_split()方法用于划分训练集和测试集

<https://blog.csdn.net/fxlou/article/details/79189106>

[9]**理解随机森林：基于Python的实现和解释**

参考了随机森林的原理和RandomForestClassifier接口的初步使用

<https://zhuanlan.zhihu.com/p/53042043>

[10]**Random Forest(sklearn 参数详解)**

参考了RandomForestClassifier接口的详细参数用法

<https://blog.csdn.net/u012102306/article/details/52228516>