高级人工智能作业 4

集成学习分类问题

姓名：董广念 学号：11849058

本报告主要使用随机森林方法解决6分类问题。采用[1]中提供的”train.csv”文件作为训练数据集；“train.csv”文件中数据本地交叉验证作为参数选择主要依据；“test.csv”文件作为测试数据集给出预测结果上交。使用sklearn.ensemble[2]中的RandomForestClassifier，ExtraTreesClassifier，AdaBoostClassifier，GradientBoostingClassifier模型组合，使用投票方法给出预测结果。

报告分为（1）问题描述（2）算法设计（3）实验结果（4）实验结论（5）参考文献，五部分组成。

# 问题描述

使用一种现有的集成学习或其他软件工具，使用[1]中提供的训练数据集训练模型，找到一种能够最优化选用模型分类正确率的数据预处理方法和参数，并对[1]中提供的测试数据分类作为实验结果上交评分。

# 算法设计

本部分介绍（1）算法结构（2）软件工具（3）调参过程。

## 算法结构

本算法分为两个部分，结构如下:

1. 模型训练
2. 数据预测

## 软件工具

本部分介绍sklearn中主要使用的分类器以及其提供的参数。

sklearn为目前较流行的机器学习软件包，为用户提供了数据预处理，数据特征处理，有监督学习，无监督学习，模型选择验证等方面成熟算法的实现并为用户提供接口。

本次作业主要使用sklearn库中集成学习算法[2]，网格搜素算法[3]。

在集成学习方法中使用如下4种方法：

1. 随机森林

随机森林是由决策树作为基础弱学习器组成的集成学习方法，基本思想为：首先使每个基础弱学习器对应数据集中随机的属性；然后针对该属性对数据集进行bootstrap抽样，抽取同数据集大小相同的抽样集合；将该集合作为训练数据集训练该弱学习器。不断迭代直至所有弱学习器训练完成。分类新数据时则将新数据放入所有的弱学习器里进行预测，而后使用投票的方式产生最终预测结果。

对于学习问题对于随机森林的训练分为主要分为两部分。（1）森林参数（2）决策树参数

1. 森林参数
2. n\_estimators（弱学习器的数量）：数量过小则会增大训练集上的误差；数量过大则会导致过拟合。
3. max\_features（单个学习器处理的特征数）：数量过小会造成单个学习器分类精度变低，需要更多学习器才能达到整体好的分类效果，如果分到分类性不强的属性会使得单个学习器分类正确率小于50%；如果过大则会使的学习器趋同，降低多样性，便失去了集成学习的意义。
4. warm\_start（热初始化内部参数）：在参数搜索中利用上一个训练好的模型的内部参数初始化以达到节省训练时间的目的
5. 决策树参数

决策树参数的影响在课堂和作业3中均有讨论，故不再解释。

1. criterion（节点划分标准）：可选择“信息熵”和“基尼指数”两种决策树节点划分依据。
2. max\_depth（最大深度）：用来限制单个决策树的最大深度。
3. min\_samples\_split（最小分裂样本数）：前剪枝参数，在模型训练过程中树分支继续向下生长所需要分得的最小样本数量。
4. min\_samples\_leaf（叶子节点最小样本数）：后剪枝参数，训练过程中叶子节点最少持有的样本数，若小于此数值则做后剪枝操作
5. class\_weight（类权重）：为每个分类赋权。
6. min\_weight\_fraction\_leaf（叶结点最小权）：后剪枝参数，如果为每类赋权则落入每个叶结点的数据有对应权重，若权重小于该值则后剪枝
7. max\_leaf\_nodes（最大叶结点数量）：前剪枝参数，按照节点分类的信息增益构造决策树一直到有max\_leaf\_nodes个叶结点。
8. min\_impurity\_decrease（最小信息增益）：前剪枝参数，如果一个节点分裂的最大信息增益不超过该数值，则不分裂。
9. bootstrap（抽样方法）：构建决策树的时候是否适用bootstrap方法抽样。
10. oob\_score（袋外得分）：是否适用样本外数据验证该树的准确率。
11. n\_jobs（并行任务数）：训练时可利用操作系统调用的线程数量
12. random\_state（随机状态）：构建决策树的随机数种子或随机数发生器。
13. 极限树（ExtraTreesClassifier）

极限树是随机森林的变种，与随机森林的唯一区别在于决策树生长的分类方法：随机森林在分支时总是选择最具分类性的数据维度以及其阈值作为分支标准；在极限树分支过程中随机生成每个数据维度的阈值而后在这些阈值中选择最佳阈值作为分支标准。这一定程度上增加了随机森林的训练集误差，可能会提高模型的泛化能力。

极限树的参数与随机森林完全相同故不再赘述。

1. AdaBoost

Adaboost方法为对弱学习器迭代训练，在训练的过程中不断提高错误分类样本的权重，最终输出所有学习器以及基于该学习器分类精度的权重数组。可以指定弱学习器的类型（神经网络、决策树、支持向量机…）。

参数如下：

1. base\_estimator（基础学习器）：adaboost集成学习方法中的弱学习器。
2. n\_estimators（学习器数量）：与上文相同。
3. learning\_rate（学习率）：影响未正确分类的数据权重
4. algorithm：（学习算法）：“SAMME“和“SAMME.R”两种算法可选。
5. random\_state（随机状态）：与上文相同。
6. bagging

bagging是一种使用bootstrap抽样方式训练单个分类器的集成学习算法，基本方法与随机树学习方法相同，区别在于一方面提供了更细致的抽样方法，另一方面可以指定弱学习器的类型（神经网络、决策树、支持向量机…）

参数如下：

1. base\_estimator（基础学习器）：与上文相同。
2. n\_estimators（学习器数量）：与上文相同。
3. max\_samples（样本数量）：训练每个弱学习器时在数据集中抽取的样本比例。
4. max\_features（特征数量）：训练每个弱学习器时使用的原始数据集特征比例。
5. bootstrap（抽样方法）：用于单个学习器的训练数据是否是有放回抽样。
6. bootstrap\_features（抽样方法）：用于单个学习器的特征抽样是否为有放回抽样。
7. oob\_score（袋外得分）：测试单个学习器时是否使用非样本数据
8. warm\_start（热启动）：使用上一个学习器的内部参数初始化。
9. n\_jobs（多线程）：利用所几个线程执行训练任务。

## 调参过程

2.3.1 参数表

本次作业中调整过的决策森林参数以及其数值如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表 1** 算法参数设置 | | | | |
| **编号** | **学习器** | **参数名** | **意义** | **数值** |
| 1 | 随机森林 | n\_estimators | 弱学习器的数量 | 180 |
| 2 | criterion | 分支标准 | entropy |
| 3 | max\_features | 决策树分支时考虑的属性数量 | 2 |
| 4 | max\_depth | 决策树最大深度 | None |
| 5 | min\_samples\_split | 决策树最小节点可分割样本数 | 2 |
| 6 | min\_samples\_leaf | 决策树叶子节点最小样本数 | 1 |
| 7 |  | bootstrap | 训练样本袋外抽样 | True |
| 1 | 极限树 | n\_estimators | 弱学习器的数量 | 180 |
| 2 | criterion | 分支标准 | gini |
| 3 | max\_features | 决策树分支时考虑的属性数量 | 2 |
| 4 | max\_depth | 决策树最大深度 | None |
| 5 | min\_samples\_split | 决策树最小节点可分割样本数 | 12 |
| 6 | min\_samples\_leaf | 决策树叶子节点最小样本数 | 1 |
| 7 | bootstrap | 训练样本袋外抽样 | False |
| 1 | Adaboost | base\_estimator | 弱学习器类型 | 决策树1 |
| 2 | n\_estimators | 弱学习器数量 | 180 |
| 3 | learning\_rate | 学习率 | 1.3 |
| 4 | algorithm | 算法 | SAMME |
| 1 | Boosting | base\_estimator | 弱学习器的类型 | 决策树2 |
| 2 | n\_estimators | 弱学习器的数量 | 180 |
| 1 | 投票器 | Voting | 投票方法 | Soft |
| 1 | 决策树1 | criterion | 分支标准 | entropy |
| 2 | max\_depth | 最大深度 | 19 |
| 3 | max\_features | 训练属性数量 | log2 |
| 1 | 决策树2 | criterion | 分支标准 | entropy |
| 2 | max\_depth | 最大深度 | None |
| 3 | max\_features | 属性数量 | Log2 |

2.3.2 搜索思路及过程

首先使用作业3中表现最好的svm方法作为弱学习器分类。在作业3所做预处理以及参数设置数据和弱学习器参数。使用Adaboost方法，在表3所示参数范围内网格搜索。

|  |  |
| --- | --- |
| **表3** svm作为弱学习器网格搜素参数 | |
| **参数** | **范围** |
| 学习率 | 0.05-2，0.05为步长，共40个采样点 |
| 弱学习器数量 | 10-300，5为步长，共58个采样点 |

共2320次训练，40线程运行时间超过24小时。猜测神经网络训练时间更长，所以放弃使用svm和神经网络作为弱学习器。使用决策树。

为增加弱学习器的随机性故采用不同的训练方法，对参数类似的训练方法采用不同的剪枝方式来增加弱学习器的多样性。

调参过程运用的主要方法为sklearn.model\_selection[3]提供的网格搜索方法进行参数的搜索。调参过程未涉及所有参数。

参数的搜索的过程为：

1. 首先对随机森林的部分参数进行网格搜索，可视化至最优参数图像呈现二次型图样。
2. 固定随机森林的最优参数，针对未搜索参数对极限树进行网格搜索。直至参数呈现二次型图样。
3. 对于adaBoost和bagging方法，固定弱学习器数量与随机森林相同，其他参数随机猜测产生。

2.3.3 详细搜索过程

1. 在一定区间范围内对随机森林进行网格搜索：

无限制训练单颗随机树，得到最大树深度为29，参数搜索范围与最终取值如下

|  |  |  |
| --- | --- | --- |
| max\_depth: 1-29, 29(None) | n\_estimators: 10-300, 180 | max\_features: 1-6, 2 |

1. 验证网格搜索数据的准确性：

以网格搜索得到的最优解为参照，运用控制变量法，以参数为自变量，5段交叉验证得分为因变量，绘制图形查看参数最优性（图1，图2，图3，表4）。若在一个以及以上属性呈现峰值图形，其他维度在最优解附近保持稳定，则可以判断该组合为搜索空间内的最优组合。

|  |  |
| --- | --- |
| **图1** 交叉验证得分随弱学习器深度变化图 | **图2** 交叉验证得分随弱学习器数量变化图 |
| **图3** 交叉验证得分随弱学习器分得特征数量变化图 | |  |  | | --- | --- | | **表4** 划分系数平均得分 | | | criterion | 得分 | | entropy | 0.7113 | | gini | 0.7065 | |

如图1所示，在本文给定参数附近，树深在15以后便不再产生剧烈变化。最后在不限制树深的情况下取得了最优解。如图2所示，弱学习器的数量在150后便基本趋于稳定，在180时达到最优解。如图3所示，弱学习器分得的特征数量为2时最佳。表2所示为本轮网格搜索所有结果的信息熵和基尼系数得分的均值。信息熵更高，故在固定参数criterion为信息熵（entropy）。

1. 以增大个体多样性为原则训练极限树

因为极限树与随机森林参数大致相同，为增加弱学习器的多样性，将极限树所有参数中布尔类型参数全都与随机森林取相反，基础变量“n\_estimators ”与随机森林相同，网格搜索“min\_samples\_split”，“min\_samples\_leaf”两个随机森林未考虑的参数。

参数搜索范围与最终取值如下

|  |  |  |
| --- | --- | --- |
| min\_samples\_split: 1-1000, 12 | max\_features: 2-5, 5 | min\_samples\_leaf: 1-1000, 1 |

1. 极限树参数验证：

|  |  |  |
| --- | --- | --- |
| **图4** 叶子最小分割样本数与得分变化图 | | **图5** 分得特征数与得分变化图 |
| **图6** 叶子最小样本数与得分变化图 | |

如图4，5，6所示，在所取点周边得分均呈明显下降趋势，故取对应搜索结果。AdaBoost和boosting的参数为多次尝试后的随机取值。

# 实验结果

本次实验所用计算资源如表2所示：

|  |  |
| --- | --- |
| **表2** 实验计算资源 | |
| 设备 | 计算资源 |
| Windows10 PC | Intel i7-8700 3.2GHz CPU; 24GB RAM |
| Linux server | E5-2640 v4 2.4GHz CPU\*2; 64GB RAM |

因为所用的分类器带有随机森林、极限树，这两个分类器带有随机性，故重复运行实验100次，在交叉验证系数为5的条件下结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表5** 4个集成学习器投票运行100次交叉验证5次得分结果 | | | | |
| **公榜得分** | **平均得分** | **平均差** | **方差** | **平均单次训练时间** |
| 75.66% | 75.56% | 0.000853 | 7.29e-7 | 19.05(s) |

随机森林重复运行100次，交叉验证系数为5，实验结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表6** 随机森林运行100次交叉验证5次得分结果 | | | | |
| **公榜得分** | **平均得分** | **平均差** | **方差** | **平均单次训练时间** |
| 75.66% | 74.56% | 0.001302 | 1.70e-6 | 4.02(s) |

极限树重复运行100次，交叉验证系数为5，实验结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表5** 极限树运行100次交叉验证5次得分结果 | | | | |
| **公榜得分** | **平均得分** | **平均差** | **方差** | **平均单次训练时间** |
| 75.66% | 75.35% | 0.000946 | 8.95e-7 | 1.23(s) |

# 实验结论

在训练时间的角度来看，虽然在提高弱学习器多样性的角度上来说应该加入svm和神经网络这两个分类器。但由于训练时间过长，在有限的时间内这两者不适合作为基础学习器进行集成学习。

在正确率的角度上来看，在作业3中，svm最佳正确率为72%，集成学习最低正确率为74.56%，由此看来集成学习在分类问题上有一定的效果。但由于数据集的分类难度未知，3%是否是一个显著性的提升仍无法判断。

在交叉验证得分的角度来看，极限树表现最好，且平均差方差最小反映出了极限树的稳定性。与随机树、adaBoost，boosting结合后有一些提升但提升不大。

# 参考文献

1. <https://www.kaggle.com/c/aai-assignment-4/data>
2. <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble>
3. <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV>