AdaBoost项目实战：参数择优与泛化能力

原创 石头 机器学习算法那些事 2018-12-07

前言

本文以实验的方式验证了AdaBoost的强学习器理论，描述了模型的参数择优过程，和探讨了AdaBoost框架的泛化能力问题。最后，结合实验和理论，对AdaBoost框架进行总结，文章结尾给出源码链接。

目录

AdaBoost框架参数含义

AdaBoostClassifier和AdaBoostRegressor框架的大部分参数相同，因此，下面我们一起讨论这些参数意义，两个类如果有不同点我们会指出。

1） base_estimator：AdaBoostClassifier和AdaBoostRegressor都有该参数，表示弱学习器，原则上可以选择任何一个弱学习器，不过需要支持样本权重。

2） algorithm：这个参数只有AdaBoostClassifier有，主要是scikit-learn实现了两种Adaboost分类算法，SAMME和SAMM.R。不同点在于弱学习器权重的计算方式不同。李航老师《统计学习方法》用的是SAMME算法。

3）loss：这个参数只有AdaBoostRegression，表示损失函数的选择。

4）n_estimators：AdaBoostClassifier和AdaBoostRegressor都有，表示弱学习器的个数。

5） learning_rate：AdaBoostClassifier和AdaBoostRegressor都有，表示每个弱学习器的权重缩减系数，意义等同于正则化项。

决策树与AdaBoost算法比较

AdaBoost算法结合多个弱分类器组成一个强分类器，为了形象化的表示这一含义。本节比较了决策树与AdaBoost算法结果，设置决策树的最大深度为1，相当于该决策树是一个弱分类器。

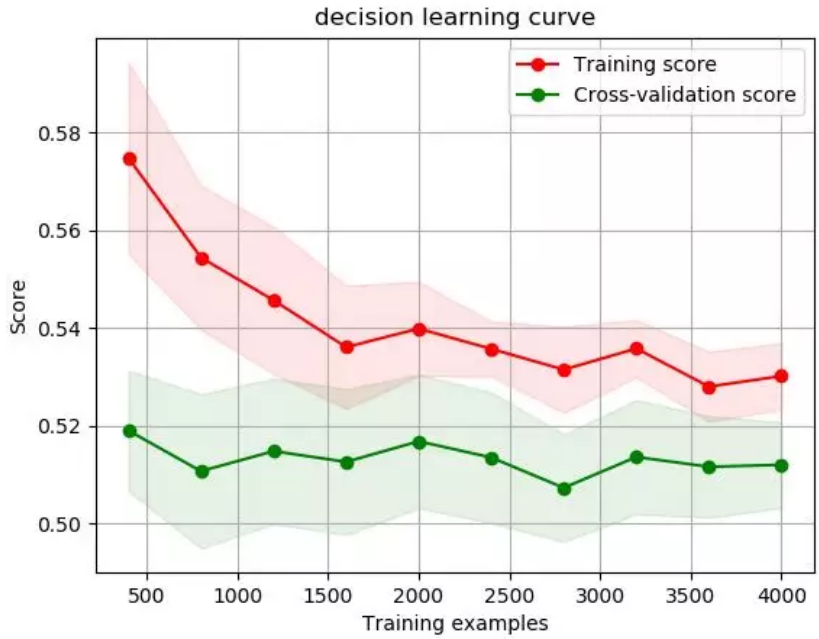
设置决策树最大深度为1：

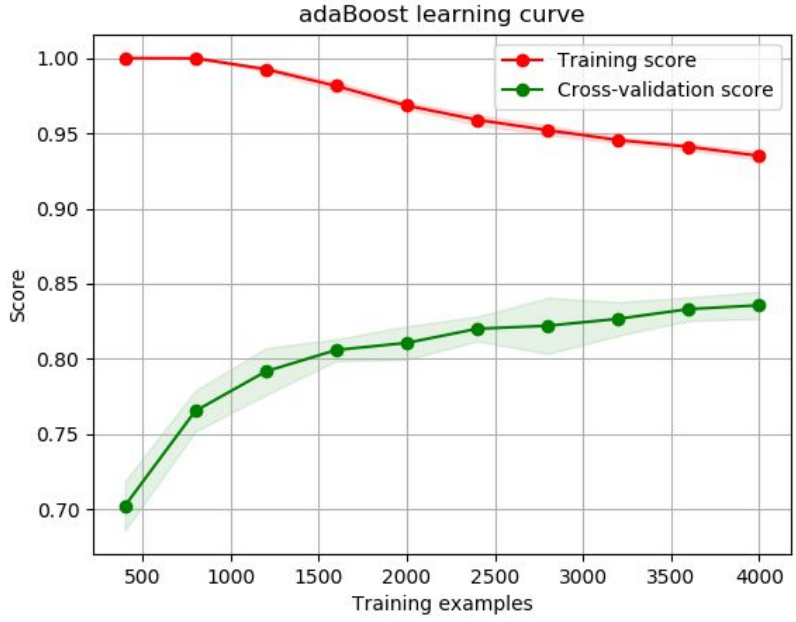
```
estimatorCart = DecisionTreeClassifier(max_depth=1)
```

设置AdaBoost模型的基学习器为该决策树，弱学习器个数是200：

```
estimatorBoost = AdaBoostClassifier(base_estimator=estimatorCart,
                                    n_estimators=200)
```

我们绘制学习曲线来比较这两类算法：





学习曲线分析：第一张图表示随着样本数的增加训练精度和测试精度稳定在0.5左右，与随机分类的概率相等，这表明**模型处于高偏差状态，是一种弱分类器。**第二张图表示随着样本数的增加测试精度稳定增长，当增长到4000例时测试精度接近85%，这表明分类模型的性能较好，是一种强分类器。

结论：AdaBoost可以结合多个弱分类器组成强分类器。

AdaBoost模型的参数择优

上节为了证明AdaBoost的强学习器理论，简单的设置了弱分类器个数和决策树深度，并未对其他参数进行设置，模型还有较大的优化空间。因此本节讨论了如何对AdaBoost模型进行参数择优。

> 集成式模型包括框架和弱学习器，我建议集成式模型的参数择优首先从框架开始，框架参数择优的过程中默认弱学习器是固定的，再对弱学习器的重要参数进行择优，此时框架的参数是择优后的参数。

因此，AdaBoost模型首先对框架进行参数择优，然后再对弱学习器进行参数择优，参数择优算法常常采用交叉验证法。

1）框架参数择优

设置弱分类器个数，我们首先对n_estimators进行网格搜索：

```
# 对框架参数 弱学习器个数进行择优
param_test1 = {"n_estimators":range(150, 300, 50)}
# 框架参数择优
gsearch1 = GridSearchCV(estimator = AdaBoostClassifier(estimatorCart)
                                        ,param_grid=param_test1, scoring="roc_auc", cv=5)
gsearch1.fit(X, y)
print gsearch1.best_params_, gsearch1.best_score_
```

打印结果：

```
{'n_estimators': 250} 0.9360104
```

因此，第一次优化的最佳弱学习器个数：250。

再次优化弱学习器个数，第一次优化是大致估计最佳参数的值，第二次优化在最佳参数的附近找最优值，下面代码是在最优参数相对误差30的范围内重新搜寻：

```
# 继续优化弱学习器个数，在最优化学习器个数的范围内再次搜寻
param_test2 = {"n_estimators":range(n_estimators1-30, n_estimators1+30, 10)}
gsearch2 = GridSearchCV(estimator = AdaBoostClassifier(estimatorCart)
                                        ,param_grid=param_test2, scoring="roc_auc", cv=5)
gsearch2.fit(X, y)
print gsearch2.best_params_, gsearch2.best_score_
```

打印结果：

```
{'n_estimators': 270} 0.938772
```

因此，第二次优化的最佳弱学习器个数：270

其实还可以按照第二次优化思想再次缩小参数的搜寻范围，因为精度提高的很小，所以就不继续对该参数进行优化了。

2） 弱学习器参数择优

AdaBoost框架有很多弱学习器可以选择，本节只讨论弱学习器是决策树的情况，决策树参数含义请参考《随机森林算法参数解释及调优》，这里主要对决策树深度和最小可划分节点数进行参数择优，其他参数可按照同样的思想去优化。

弱学习器的复杂度很低，因此决策树深度在1~2范围内搜寻，最小可划分节点数根据经验在18~22范围内搜寻。对于训练数据很大的情况，最小可划分节点数基本没起作用，训练数据很小的情况可考虑该参数。

弱学习器参数优化代码：

```
for i in range(1,3):        # 决策树最大深度循环
    print i
    for j in range(18,22):
        print j
        bdt = AdaBoostClassifier(DecisionTreeClassifier(max_depth=i,min_samples_split=j),
                                 n_estimators=n_estimators2)
        cv_result = cross_validate(bdt,X,y,return_train_score=False,cv=5)
        cv_value_vec = cv_result["test_score"]
        cv_mean = np.mean(cv_value_vec)
        if cv_mean >= score:
            score = cv_mean
            tree_depth = i
            samples_split = j
```

代码中的n_estimators2的值等于270。

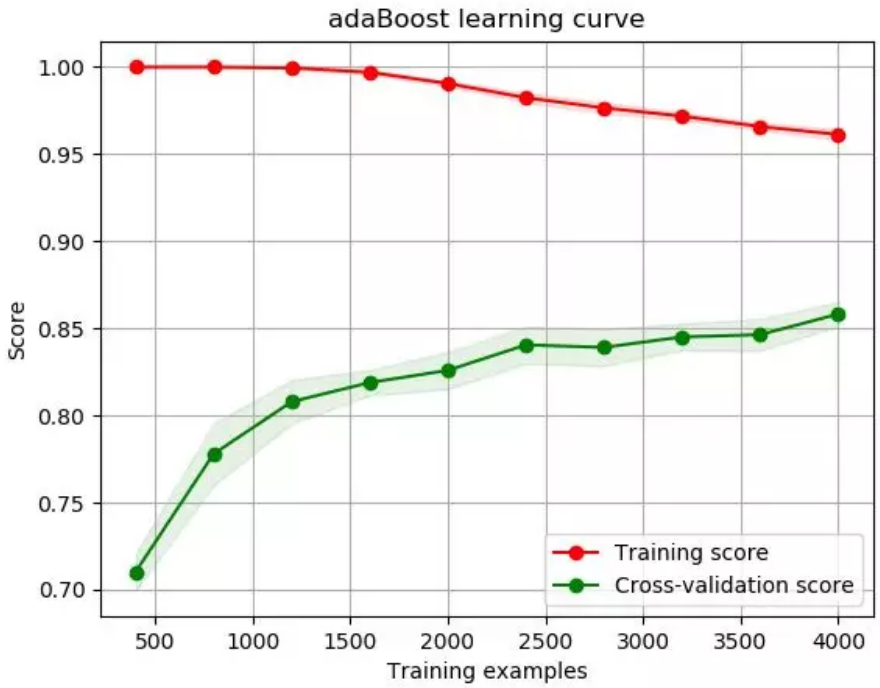
因为训练数据比较大，所以弱学习器的最小可划分节点数（min_samples_split）没起作用，我们设置该参数为20，最后得到最优决策树的最大深度等于1。

用该最优参数重新构建模型，并输出测试结果：

```
bdt = AdaBoostClassifier(DecisionTreeClassifier(max_depth=tree_depth),
                         n_estimators=n_estimators2)
bdt.fit(X_train,y_train)
print bdt.score(X_test,y_test)
```

测试准确率：85.6%

3）重新绘制学习曲线

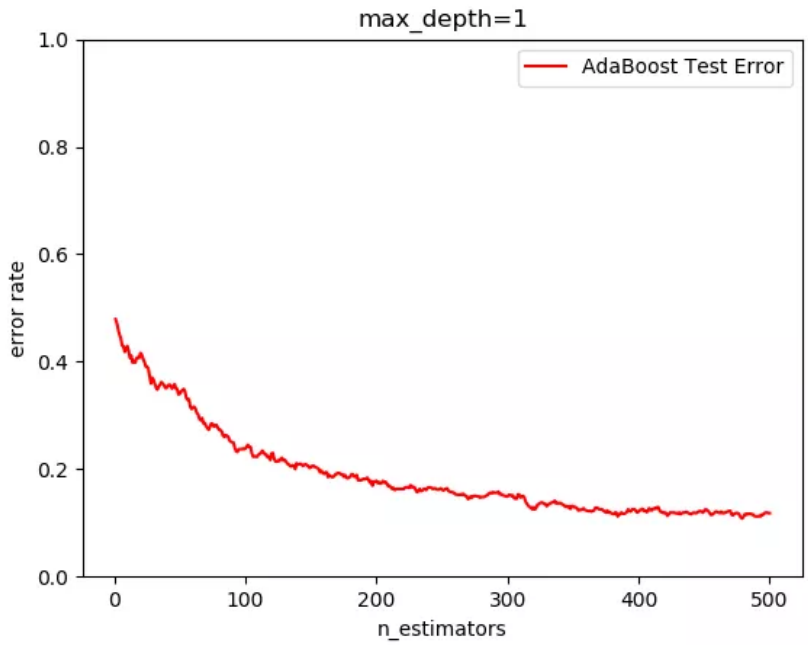


参数优化后的学习曲线较第一节的学习曲线略有提升，相信如果我们再增加训练数据，测试准确率还会提高。

模型泛化能力的探讨

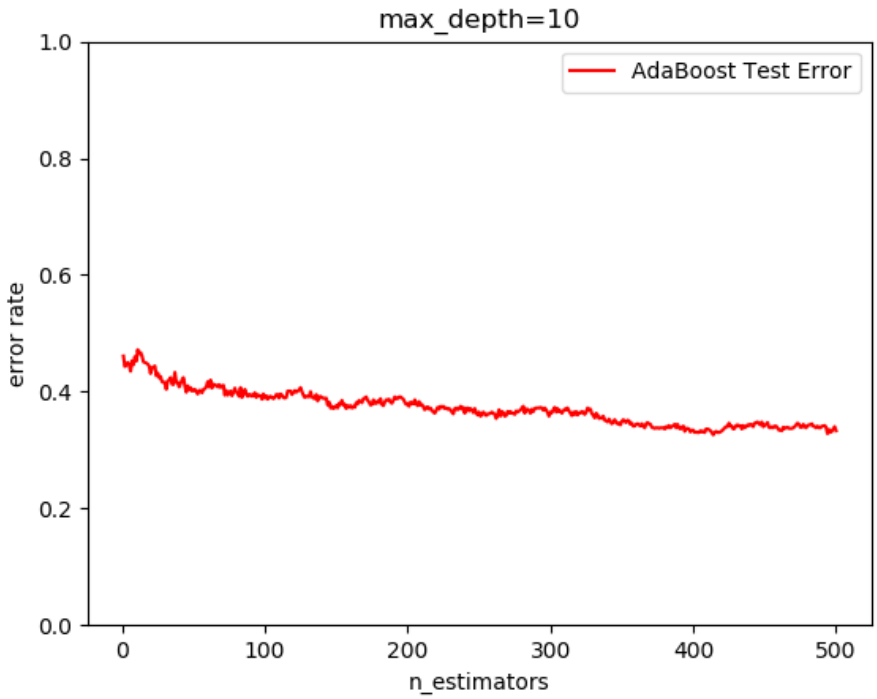
AdaBoost框架的泛化问题是一个很容易让人疑惑的点，网上查的资料很容易让人混淆，下面我通过实验的方式对某些结论进行验证（样本数一定的情况下）：

1）决策树深度设置为1时，观察迭代次数与测试误差的关系：



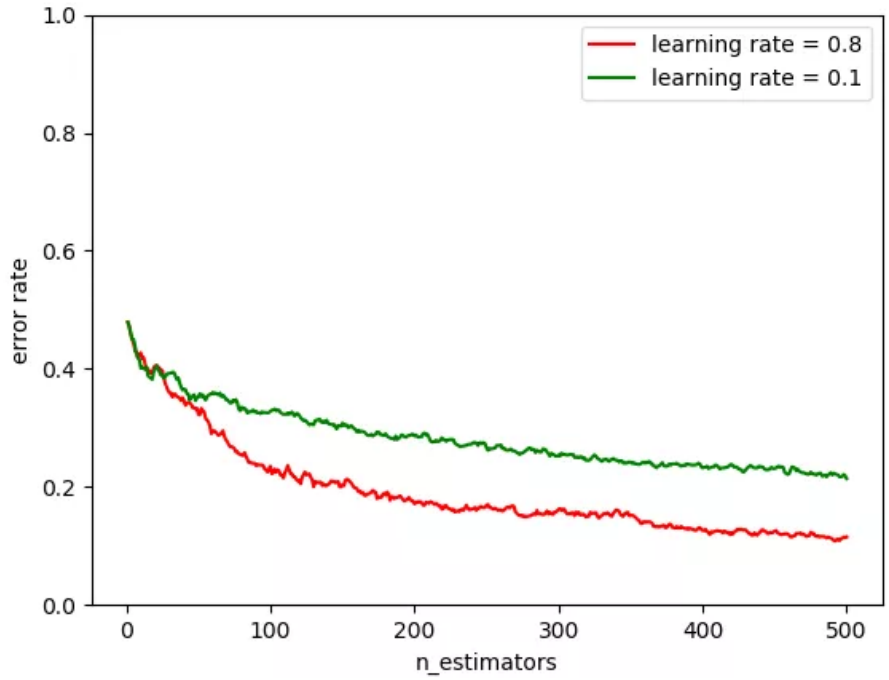
结论：弱学习器的复杂度很小，当迭代次数增加时，测试误差减小，泛化能力增强

2）决策树深度设置为10时，观察迭代次数与测试误差的关系：



结论：弱学习器的复杂度较大，当迭代次数增加时，测试误差无明显变换，即泛化能力没有增强，模型处于过拟合状态。

（3）探讨学习率与迭代次数的关系



结论：当学习率较大时，即步长较大，对于同样的弱学习器个数，学习率大的泛化性能好。（与上节关于学习率与泛化误差的理论相反，以结果为依剧，我支持当前的结论）

总结：（1）弱学习器的复杂度是我们提高模型泛化能力最大的影响因素，若数据集含有很大的噪声，我们也认为该数据的复杂度很高，复杂度很高的模型，增加弱学习器的数目不能增加模型的泛化能力，我们需要想办法去降低模型的复杂度。（2）学习率越大，那么在相同数目的弱学习器下，泛化能力越强。

总结

AdaBoost是通过一系列的弱学习器组成强学习器的方法，切记，基学习器的复杂度尽量小，可通过提高弱学习器数目的方式提高泛化能力。但是若数据含有较大的噪声或弱学习器模型复杂度较高，提高弱学习器个数并不能提高泛化能力。

集成式方法的应用非常广，希望我的文章能帮到你，源码已经给出，希望您拿到源码后自己跑一遍，然后修改模型的关键参数再运行代码，说不定你能得到一些新想法。

参考

https://scikit-learn.org

推荐阅读

集成学习原理总结

比较全面的随机森林算法总结

比较全面的Adaboost算法总结

后台回复“AdaBoost”获取源码链接