寒假第三周读书报告 2.3-2.9

61518426 周之遥

·读书进度

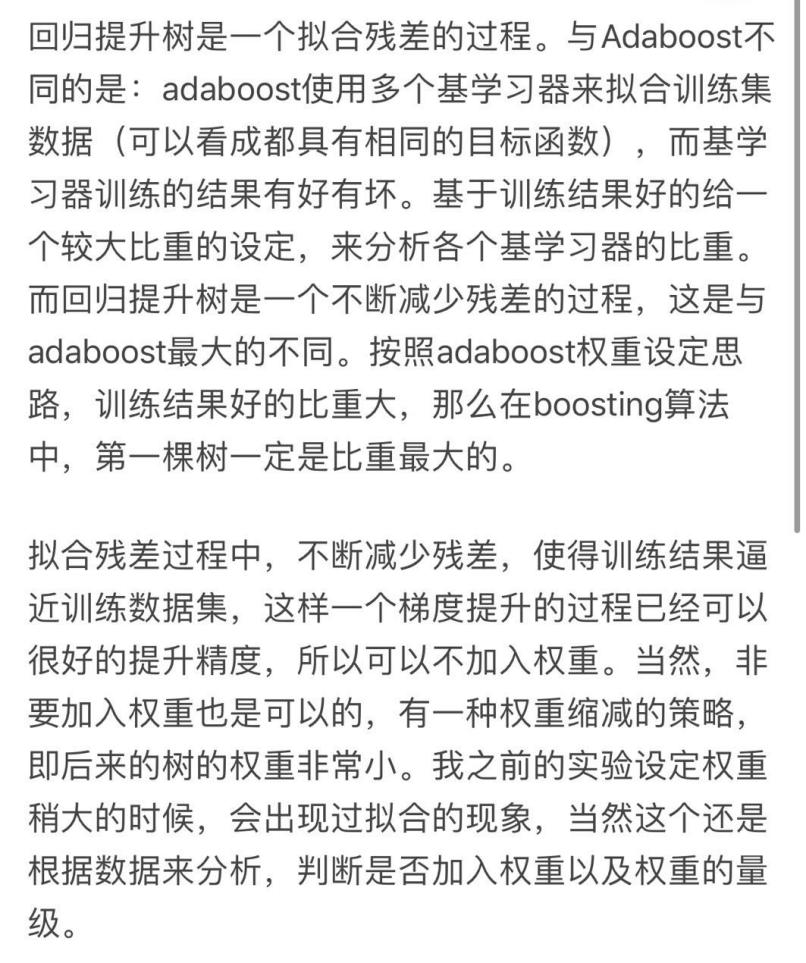
统计学习方法完成第8章：提升方法

掌握了提升方法的概念和典型的提升方法AdaBoost算法，AdaBoost算法是前向学习算法在基函数为基本分类器、损失函数为指数函数情况下的特例。

在基函数为决策树情况下，有提升树的概念。提升树有回归树和分类树，分类树是AdaBoost在基函数为决策树情况下的特例。在损失函数较一般的情况下，可以用梯度提升拟合残差，获得回归树。

·提出问题

1. 为什么提升树模型中没有权值？

 解答：

1. 在梯度提升构建回归树的算法中，(2)b步骤不是已经获得了cmj吗？为什么不采用这个而是保留了叶结点区域但重新用另一种方式获得cmj？
2. 在梯度提升构建回归树的算法中，为什么书上M个子回归树的区域数都一样？
3. AdaBoost如何拓展至多分类问题？

解答：显然分类树就可以多分类，sklearn库中Adaboost默认使用提升树，可以用于多分类。

1. 提升方法中分类器过多是否会出现过拟合状况？有正则化方法吗？

思考：根据资料，提升方法的正则化是在每一轮新学习的分类器项前加上一个因子，即learning rate。而后采用交叉验证的方法。

·解决问题

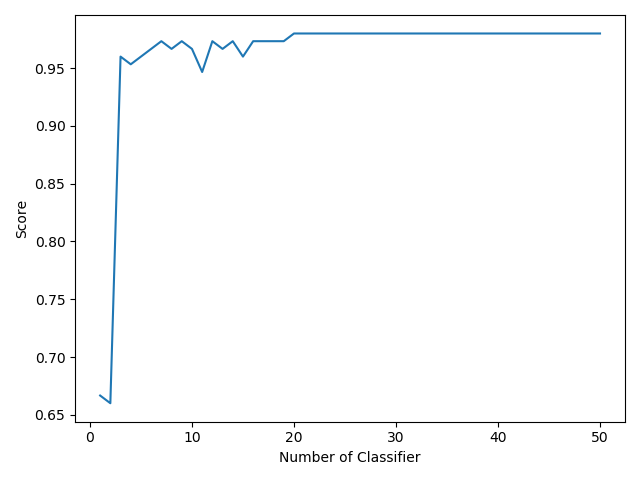
1. 梯度提升算法第一步初始化为什么要估计损失函数极小化的常数值？

解答：因为梯度值的计算从零开始，需要初始化一个简单模型，以计算之后的梯度。

·代码实现

1.AdaBoost用于鸢尾花分类

**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**from** sklearn **import** datasets  
**from** sklearn.ensemble **import** AdaBoostClassifier  
**from** sklearn.tree **import** DecisionTreeClassifier  
  
iris = datasets.load\_iris()  
X = iris.data  
Y=iris.target  
score=[]  
number=np.arange(1,51)  
**for** i **in** range(50):  
 model=AdaBoostClassifier(algorithm=**"SAMME"**,n\_estimators=i+1)  
 model.fit(X,Y)  
 s=model.score(X,Y)  
 print(s)  
 score.append(s)  
plt.plot(number,score)  
plt.xlabel(**'Number of Classifier'**)  
plt.ylabel(**'Score'**)  
plt.show()

如下是结果图：

可见在分类器数量为3时，就已经达到了较好的预测能力(0.96)。