Adaboost

1. 原理

Adaboost算法基本原理就是将多个弱分类器（弱分类器一般选用单层决策树）进行合理的结合，使其成为一个强分类器。

Adaboost采用迭代的思想，每次迭代只训练一个弱分类器，训练好的弱分类器将参与下一次迭代的使用。也就是说，在第N次迭代中，一共就有N个弱分类器，其中N-1个是以前训练好的，其各种参数都不再改变，本次训练第N个分类器。其中弱分类器的关系是第N个弱分类器更可能分对前N-1个弱分类器没分对的数据，最终分类输出要看这N个分类器的综合效果。

关于弱分类器，Adaboost一般使用单层决策树作为其弱分类器。单层决策树是决策树的最简化版本，只有一个决策点，也就是说，如果训练数据有多维特征，单层决策树也只能选择其中一维特征来做决策，并且还有一个关键点，决策的阈值也需要考虑。

关于Adaboost的两种权重，Adaboost算法中有两种权重，一种是数据的权重，另一种是弱分类器的权重。其中，数据的权重主要用于弱分类器寻找其分类误差最小的决策点，找到之后用这个最小误差计算出该弱分类器的权重（发言权），分类器权重越大说明该弱分类器在最终决策时拥有更大的发言权。

由于Adaboost中若干个分类器的关系是第N个分类器更可能分对第N-1个分类器没分对的数据，而不能保证以前分对的数据也能同时分对。所以在Adaboost中，每个弱分类器都有各自最关注的点，每个弱分类器都只关注整个数据集的中一部分数据，所以它们必然是共同组合在一起才能发挥出作用。所以最终投票表决时，需要根据弱分类器的权重来进行加权投票，权重大小是根据弱分类器的分类错误率计算得出的，总的规律就是弱分类器错误率越低，其权重就越高。

1. 调包实现

调用sklearn包中的AdaBoostClassifier与其他包

|  |
| --- |
| import time  from sklearn.model\_selection import train\_test\_split  from sklearn.metrics import accuracy\_score  from sklearn.ensemble import AdaBoostClassifier |

加载数据

|  |
| --- |
| features = [] labels = [] fr = open('newHIV-1\_data/746Data.txt') for line in fr.readlines():  curline = line.strip().split(',')  curDataArr = []  for i in range(len(curline[0])):  curDataArr.append(ord(curline[0][i]))  features.append(curDataArr)  labels.append(curline[-1]) |

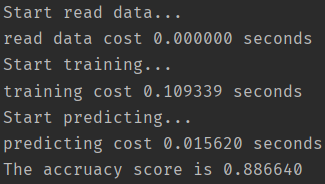
调包实现

|  |
| --- |
| clf = AdaBoostClassifier(n\_estimators=100, algorithm='SAMME.R') clf.fit(train\_features, train\_labels)  test\_predict = clf.predict(test\_features)  score = accuracy\_score(test\_labels, test\_predict) |

数据分割成训练集和测试集，测试集占0.3

|  |
| --- |
| Xtrain, Xtest, Ytrain, Ytest = train\_test\_split(dataArr, targetArr, test\_size=0.3) |

运行



1. 实验

评价标准：准确率

同一个数据集

|  |  |  |
| --- | --- | --- |
| 训练集 | 测试集 | 准确率 |
| 70%746data | 30%746data | 0.88664 |
| 70%1625Data | 30%1625Data | 0.92365 |
| 70%impensData | 30%impensData | 0.875399 |
| 70%schillingData | 30%schillingData | 0.92963 |

不同数据集：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测 训 | 746data | 1625Data | impensData | schillingData |
| 746data | / | 0.886769 | 0.799366 | 0.827934 |
| 1625Data | 0.878016 | / | 0.86906 | 0.870721 |
| impensData | 0.686327 | 0.825846 | / | 0.896088 |
| schillingData | 0.636729 | 0.830769 | 0.87962 | / |

1. 总结
2. 第一感觉就是很像BP神经网络中的权重更新，不过两种方法有着本质的不同。而对于权重的作用，就是为了提高错点的注意力，让下次再分的时候能够分对。
3. 第二感觉就像把一颗很深很大的树拆成了很多很多棵小树，我觉得这么理解没问题，因为Adaboost本来就是把很多棵小树组成了大树。
4. 仍然有投票成分在里面。