机器学习实验 13. 集成学习（一）AdaBoost

AdaBoost 原理概述

AdaBoost 是 adaptive boosting(自适应 boosting)的缩写，其运行过程如下：训练数据中的每个样本，并赋予其一个权重，这些权重构成了向量 D。一开始，这些权重都初始化成相等值。首先在训练数据上训练出一个弱分类器并计算该分类器的错误率，然后在同一数据集上再次训练弱分类器。在分类器的第二次训练当中，将会重新调整每个样本的权重，其中第一次分对的样本的权重将会降低，而第一次分错的样本的权重将会提高。为了从所有弱分类器中得到最终的分类结果，AdaBoost 为每个分类器都分配了一个权重值 alpha，这些 alpha 值是基于每个弱分类器的错误率进行计算的。其中，错误率ε的定义为：

$$\varepsilon = \frac{\text{未正确分类的样本数目}}{\text{所有样本数目}}$$

而 alpha 的计算公式如下：

$$\alpha = \frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$$

计算出 alpha 值之后，可以对权重向量 D 进行更新，以使得那些正确分类的样本的权重降低而错分样本的权重升高。D 的计算方法如下：

如果某个样本被正确分类，那么该样本的权重更改为：

$$D_i^{(t+1)} = \frac{D_i^{(t)} e^{-\alpha}}{\mathrm{Sum}(D)}$$

而如果某个样本被错分，那么该样本的权重更改为：

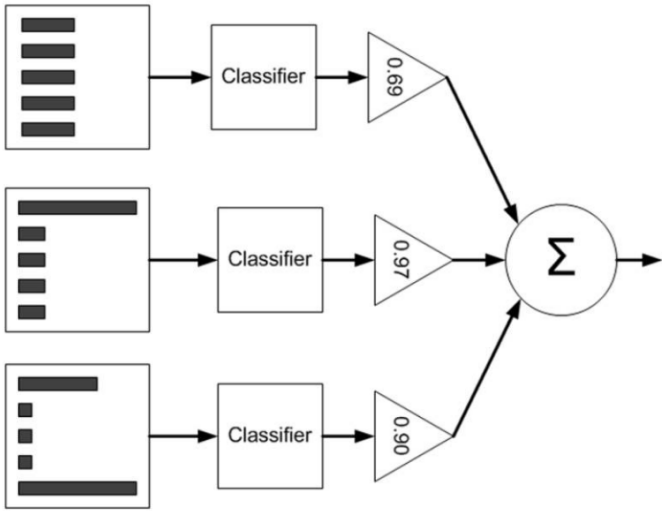
$$D_i^{(t+1)} = \frac{D_i^{(t)} e^{\alpha}}{\mathrm{Sum}(D)}$$

在计算出 D 之后，AdaBoost 对又开始进入下一轮迭代。其思想是将关注点放在被错误分类的样本上，减小上一轮被正确分类的样本权值，提高那些被错误分类的样本权值，AdaBoost 算法会不断地重复训练和调整权重的过程，直到训

练错误率为 0 或者弱分类器的数目达到用户的指定值为止。

注意：alpha 的目的主要是计算每一个分类器实例的权重。

如下图所示，左侧是数据集（排成一列），其中直方图的不同宽度表示每个样例上的不同权重。在经过分类器后，加权的预测结果会通过三角形中的 alpha 值进行加权。在每个三角形中输出加权结果，并在圆形中求和，从而得到最终的输出结果。



AdaBoost 算法的流程

对每次迭代：

- 训练当前迭代最优弱分类器
- 计算最优弱分类器的权重
- 根据错误率更新样本权重
- 达到终止条件则停止，否则不断重复上述三个步骤

（终止条件是强分类器的错误率低于最低错误率阈值或达到最大迭代次数）

AdaBoost 算法的优点：

（1）分类精度高；

（2）AdaBoost 是一种分类器组合框架，可以使用各种方法构建弱分类器；

（3）结构、原理较为简单，不需要做特征筛选；

（4）不用担心过度拟合。

AdaBoost 算法在数据中的一般实现

1) 导入数据集：

[1]:
```
#AdaBoost
from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import cross_val_score #交叉验证
from sklearn.datasets import load_iris
iris=load_iris()  #从sklearn自带数据库中读取数据,iris以字典形式存储
    #iris的4个属性是：萼片宽度  萼片长度  花瓣宽度  花瓣长度
    #标签是花的种类：setosa versicolour virginica
X=iris['data']  #数据集大小为：150个数据样本，每个样本有四个特征
y=iris['target']  #数据对应的标签（有三类）:分别用 0,1,2 表示
print(X.shape)
print(y.shape)
```

```
(150, 4)
(150,)
```

2) 调用 python AdaBoostClassifier 函数进行分类，并运用交叉验证的方法进行模型测评。

[2]:
```
ada=AdaBoostClassifier(random_state=10)
scores=cross_val_score(ada,X,y,cv=3)
        #使用三次交叉验证
print(scores)
        #打印三次交叉验证结果
print(scores.mean())
        #输出三次交叉验证均值
```

```
[0.98039216 0.94117647 0.97916667]
0.9669117647058822
```