机器学习实验---调用sklearn库实现GBM

原创

卷心菜菜 于 2017-10-28 14:23:33 发布 3173 收藏 2

分类专栏： 机器学习 文章标签： 机器学习

本文链接：https://blog.csdn.net/qq5q13638/article/details/78375518

导语

一直在纠结GBM这个方法到底是怎么调用的，因为貌似sklearn中并没有一个专门的方法叫做GBM。emmm，后来好像其实一般就说gradient boosting。根据官方文档我们迅速来学习它的使用方式吧~

简单步骤

1、导入分类器和make_hastie_10_2

```
from sklearn.datasets import make_hastie_10_2
from sklearn.ensemble import GradientBoostingClassifier
```

第一句话是导入make_hastie_10_2这个东西，我看到这里的时候也不知道是个啥，等下用到的时候我们再来解释吧

然后下一句就是导入的GBM中分类器的语句了

2、导入数据

```
X,y = make_hastie_10_2(random_state=0)
X_train,X_test = X[:2000],X[2000:]
y_train,y_test = y[:2000],y[2000:]
```

这里就用到了我们之前导入的make_hastie_10_2，貌似这是一个生成训练集和测试集的方法吗？猴赛雷的样子~

```
In[3]: from sklearn.datasets import make_hastie_10_2
In[4]: X,y = make_hastie_10_2(random_state=0)
In[5]: type(X)
Out[5]: numpy.ndarray
In[6]: X.shape
Out[6]: (12000L, 10L)
In[7]: y.shape
Out[7]: (12000L,)
```


所以，回看我们这句话，应该就是从这12000个数据中取前2000行作为训练集，2000行之后的作为测试集

3、开始训练

```
>>>clf = GradientBoostingClassifier(n_estimators = 100,learning_rate = 1.0,max_depth = 1,random_state = 0).fit(X_train,y_train)
>>>clf.score(X_test,y_test)
```

这段代码的第一句就已经在使用fit方法吃数据了，然后第二句话，clf.score则是在使用吃饱了的分类器对测试集进行测试，看看准确率。

补充：预测

以上所说的确实能够让我们知道这个算法的准确率，但是，我是要运用这个算法的呀，我光知道你准不准确有什么用啊？

还记得我们之前所用的机器学习算法的时候，是用：

```
    pre2 = clf.predict(X2)
```

这样的语句来将对X2特征的判断赋值给pre2，在这里我们同样这样操作：

```
In[5]: pre = clf.predict(X_test)
In[6]: pre
Out[6]: array([ 1.,  1.,  1.,  1.,  1.])
```


其中的输出就是分类的向量了。

参数说明

学会怎么样一步一步去实现还不够，我们接下来侃侃在训练的时候的一些参数的含义和意义：

```
>>>clf = GradientBoostingClassifier(n_estimators = 100,learning_rate = 1.0,max_depth = 1,random_state = 0).fit(X_train,y_train)
```

n_estimators：定义了需要使用到的决策树的数量，虽然GBM有较多决策树的时候还可以保持稳健，但是还是有可能会发生过度拟合，所以要用交叉验证进行检验

learning_rate：控制权重更新的幅度，通常来说这个值不应该设得太大，默认值是0.1，较小的学习速度更有稳健性

max_depth：定义了输的最大深度，可以控制过拟合

random_state：int，RandomState实例或无，可选（默认=无）
如果int，random_state是随机数生成器使用的种子; 如果RandomState的实例，random_state是随机数生成器; 如果没有，随机数生成器是由np.random使用的RandomState实例。这个地方想来搞懂GBM的原理之后再来理解会更加容易吧~

那么关于GBM的介绍就到这里，进一步的理解如果有机会以后再补充