

机器学习实验---集成学习

一、实验目的

1. 理解 bagging 和 adaboost 算法原理，能实现这两种算法；
2. 掌握不同集成方式的基本思想，以及其优缺点；
3. 掌握使用 sklearn 实现决策树+bagging / adaboost 的方法；

二、实验内容

1. 从 UCI 数据库中下载一个多分类数据集，进行数据说明，设计一个决策树+bagging / adaboost 的集成二分类模型。以 2/3 的数据为训练集，1/3 为测试集。

三、实验报告要求

1. 按实验内容撰写实验过程；
2. 报告中涉及到的代码，每一个模块需要有详细的注释；

四、参考样例

```
# encoding=utf-8

import time
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier, AdaBoostClassifier
from sklearn import preprocessing

if __name__ == '__main__':

    print("Start read data...")

    time_1 = time.time()

    raw_data = pd.read_csv('breast-cancer-
wisconsin.data', header=None) # 读取 csv 数据
    data = raw_data.values

    features = data[:, 1:-1]
    # 删除缺失值
```

```

index0 = np.where(features[:,5]!='?')
features = features[index0].astype('int32')
labels = data[:, -1][index0]

# 避免过拟合，采用交叉验证，随机选取 33%数据作为测试集，剩余为训练集
train_attributes, test_attributes, train_labels, test_labels = train_test_split(features, labels, test_size=0.33, random_state=0)
time_2 = time.time()
print('read data cost %f seconds' % (time_2 - time_1))

# 通过生成决策树
print('Start training...')

# 调用 sklearn 的包进行构造
dt = DecisionTreeClassifier(criterion='entropy')
dt.fit(train_attributes, train_labels.astype('int32'))

dt_score = accuracy_score(dt.predict(test_attributes), test_labels.astype('int32'))

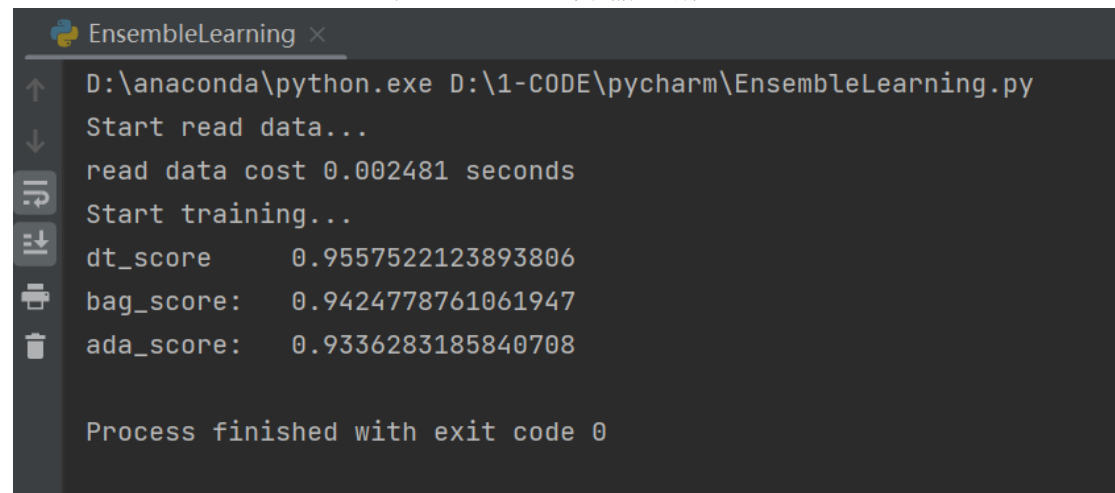
# 利用 bagging 实现决策树增强的效果
bag_clf = BaggingClassifier(DecisionTreeClassifier(criterion='entropy'),
                             n_estimators=500,
                             max_samples=80,
                             bootstrap=True,
                             n_jobs=-1,
                             oob_score=True)
bag_clf.fit(train_attributes, train_labels.astype('int32'))
bag_score = accuracy_score(bag_clf.predict(test_attributes), test_labels.astype('int32'))

# 利用 adaboost 实现决策树增强的效果
ada_boost_clf = AdaBoostClassifier(DecisionTreeClassifier(criterion='entropy'),
                                    algorithm="SAMME.R", n_estimators=500, learning_rate=1)
ada_boost_clf.fit(train_attributes, train_labels.astype('int32'))
ada_score = accuracy_score(ada_boost_clf.predict(test_attributes), test_labels.astype('int32'))
print('dt_score', dt_score, 'bag_score: ', bag_score, 'ada_score: ', ada_score)

```

五、运行结果

1. 利用两种方法预测测试集上的结果，并输出精度。



```
EnsembleLearning x
D:\anaconda\python.exe D:\1-CODE\pycharm\EnsembleLearning.py
Start read data...
read data cost 0.002481 seconds
Start training...
dt_score      0.9557522123893806
bag_score:    0.9424778761061947
ada_score:    0.9336283185840708

Process finished with exit code 0
```

六、实验小结

本次实验是理解 bagging 和 adaboost 算法的原理并实现。理解集成学习的基本思想，以及其适用场景。并能够将这两种基本的集成学习方法应用到不同的任务中，实现性能的增强。