用集成方法识别垃圾邮件和正常邮件

一、随机森林

由于随机森林中每个决策树均有未使用的样本，因此测试时可采用这些未用到的样本进行包外估计，所以可以用全部训练集对分类器做训练。令基分类器个数为 T，每一个基分类器 t 均对样本进行有放回采样，共采样 n（样本总数）次，得到训练集样本编号 D_t，再随机选取 m=int(log(57))=4 个特征 AT_t，再取 D_t 在总样本编号集中的补集 D_test_t。

再用 D_t 与 AT_t 对应的样本对该决策树做训练，需要对决策树剪枝：令每一个叶子结点至少包含 n*10%的样本，并令其深度最大值为 d.对于 d 的取值：令 d 从 1 开始不断增大,计算该决策树在 D_test_t 与 AT_t 构成的样本上的错误率，当错误率开始增大，或 d<6 时停止，得到决策树 Tr_t，T 个决策树平均最大深度约为 1.35。

测试：对于每一个样本 i，在未用到该样本的基分类器上做测试并对分类结果进行投票来衡量总分类器错误率。因此，令 pre_test 为总分类器对每个样本分类结果，num_test 为每个样本出现在 D_test_t 中的次数。对于每一个 Tr_t，预测 D_test_t，将结果计入 pre_test 对应位置中，每一个决策树得到结果相加，并根据 D_test_t 填写 num_test。计算 pre_test=pre_test/num_test，得到每个样本平均预测结果，由于样本分类编号为 0/1，因此新的 pre_test 中小于 0.5 的元素分类为 0，大于 0.5 的元素分类为 1，这样得到新的 pre_test。求出 pre_test 与样本原标签 e_label 差的一范数，得到训练器分错的样本个数，在除 n 得到分类器错误率。

T 的选取：T 取 100 时，运行 5 次后，总分类器平均错误率约为 0.19513149315366224，此时各基分类器平均错误率约为：0.2741280576859674。T 取 500 时，运行 5 次总分类器平均错误率约为：0.19656596392088677。与 T 为 100 时没有明显变化，因此 T 取 100.此时总分类器错误率比各基分类器平均错误率下降了约 0.07899656453230514.

二、AdaBoost

首先将样本分为训练集和测试集，从样本中随机选取 1/4 比例的样本作为测试集，剩下的作为训练集。

对训练集做 AdaBoost，其中训练各决策树时做与随机森林中同样的剪枝，得到的每个决策树平均深度约为 1.78。最后对总分类器做测试，求出总分类器错误率。

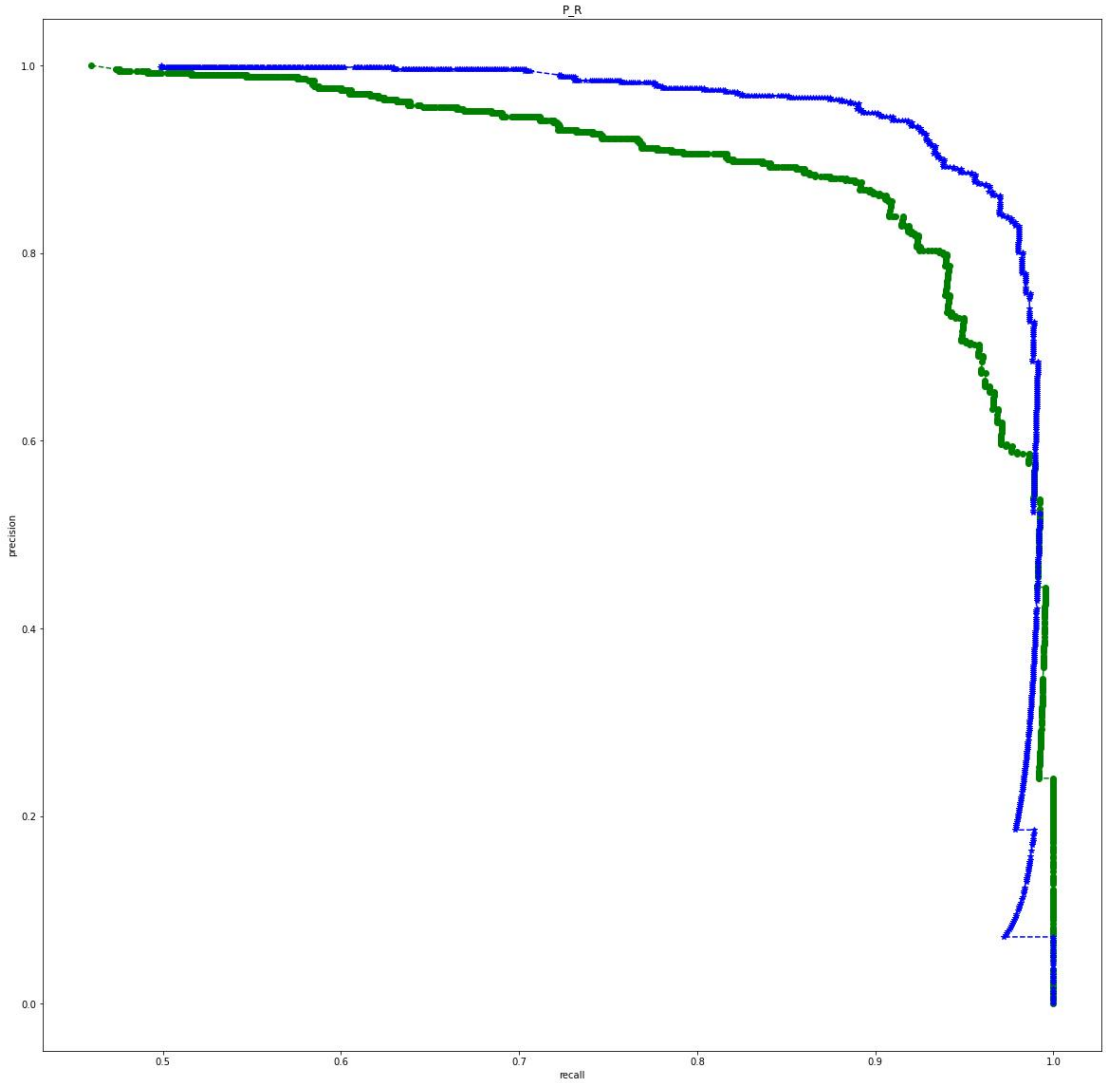
基分类器个数 T 的选取：T 取 100 时，运行 5 次后，总分类器平均错误率约为 0.12267593397046046，正确率已经接近 90%，将 T 增大到 500 后，运行 5 次总分类器平均错误率约为 0.11120764552562987，运行时间增长而错误率没有明显下降,因此取 T 为 100。此时各基分类器平均错误率约为:0.2030443092962642，总分类器错误率比各基分类器平均错误率下降了约 0.08036837532580375.

三、比较随机森林与 AdaBoost

将样本划分为测试集和训练集，令测试集比例为 0.25，调用 sklearn 中随机森林和 AdaBoost 做训练，对两个算法中的基分类器进行同样的剪枝，最大深度取一、二中 d_的均值：$(1.35+1.78)/2=1.565\approx 2$，并令叶子结点包含的最小样本数为 10%*训练样本总数。

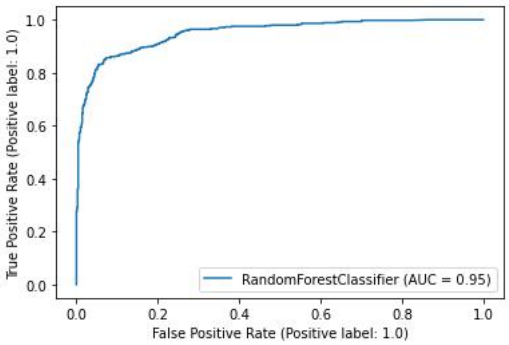
当树的个数 T 取 100 时，随机森林在测试集上准确率约为：0.8844483058210252，AdaBoost 在测试集上准确率约为：0.9374456993918332，AdaBoost 准确率更高。

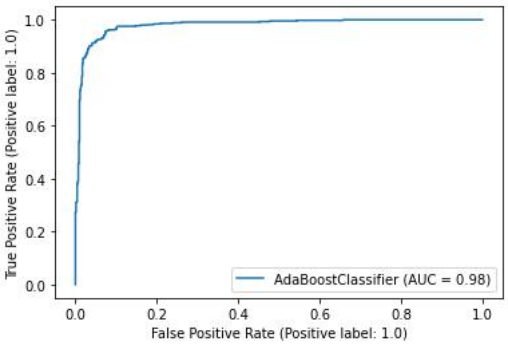
P-R 曲线如下，其中绿色为随机森林，蓝色为 AdaBoost：



随机森林对应曲线基本在 AdaBoost 曲线下方，因此在该问题中，AdaBoost 对正例（垃圾邮件）识别更准确。但在查全率接近 1 时，两种算法查全率相等时，AdaBoost 查准率略低于随机森林，即随机森林对于处于正负例交界处的正例分类更好。

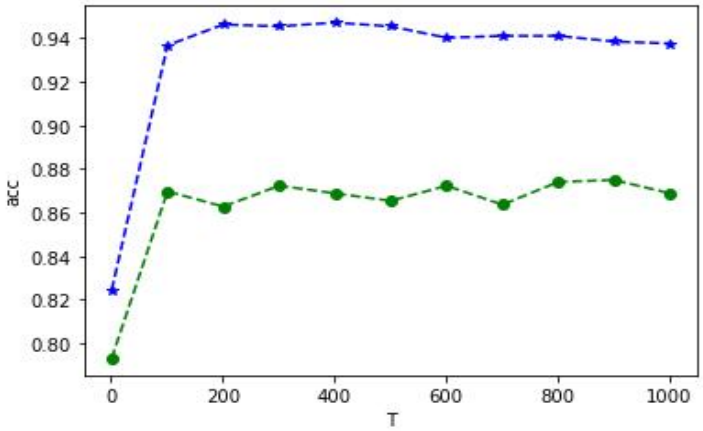
下图为随机森林和 AdaBoost 的 ROC 曲线：





随机森林 AUC 值为 0.95，AdaBoost 的 AUC 值为 0.98，因此在该问题中，AdaBoost 对样本排序的质量优于随机森林。

T 从 1 开始，以 100 为步长增大到 1001 时，计算两种算法测试集错误率，结果如下（绿色为随机森林，蓝色为 AdaBoost）：



在此过程中，随机森林准确率一直低于 AdaBoost，并且两个分类器准确率在 T 从 1 到 100 时提升明显，T>100 后两个算法准确率变化均不明显。

```
(比较两个算法性能)
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
```

```
    from sklearn.metrics import precision_recall_curve
    from sklearn import tree
    from sklearn import metrics

    spambase=np.loadtxt('spambase.data',delimiter=',')
    e_A=spambase[:,0:57]  #4601 个样本各属性值
    e_label=spambase.T[57]  #4601 个样本标签（1 垃圾，0 非垃圾）
    n=len(e_A)#样本个数

    #采样
    p=0.25#测试集比例
    X_train,X_test,Y_train,Y_test=train_test_split(e_A,e_label,test_s
ize=p)

    #训练
    T=100#基分类器个数
    d=2#决策树 max_depth

    #随机森林
    RF=RandomForestClassifier(n_estimators=T,max_depth=d,min_samples_
leaf=int(0.1*len(X_train)),n_jobs=-1)
    #所有核并行运算
    RF=RF.fit(X_train,Y_train)
    P_RF=RF.predict_proba(X_test)#测试集样本分类概率
    RF_acc=RF.score(X_test,Y_test)#准确率

    #AdaBoost,
    AB=AdaBoostClassifier(tree.DecisionTreeClassifier(max_depth=d,min
_samples_leaf=int(0.1*len(X_train))),n_estimators=T)
    AB=AB.fit(X_train,Y_train)
    P_AB=AB.predict_proba(X_test)#测试集样本分类概率
    AB_acc=AB.score(X_test,Y_test)#准确率

    #画图
    #roc
    metrics.plot_roc_curve(RF,X_test,Y_test)
    metrics.plot_roc_curve(AB,X_test,Y_test)
    #P-R
    precision1,recall1,thresholds1=precision_recall_curve(Y_test,P_RF.
T[1])#蓝
    precision2,recall2,thresholds2=precision_recall_curve(Y_test,P_AB.
T[1])#绿
    plt.figure(figsize=(20,20))
    plt.title('P_R')
```

```
    plt.xlabel('recall')
    plt.ylabel('precision')
    plt.plot(precision1,recall1,'b*--')
    plt.plot(precision2,recall2,'go--')
```

（算法准确率随 T 的变化）

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn import tree

spambase=np.loadtxt('spambase.data',delimiter=',')
e_A=spambase[:,0:57]  #4601 个样本各属性值
e_label=spambase.T[57]  #4601 个样本标签（1 垃圾，0 非垃圾）
n=len(e_A)#样本个数

#采样
p=0.25#测试集比例
X_train,X_test,Y_train,Y_test=train_test_split(e_A,e_label,test_size=
p)

T=1#基分类器个数
T_=[]
RF_acc_=[]#每个 T 对应 RF 上准确率
AB_acc_=[]#每个 T 对应 AB 上准确率
while(T<=2001):
    T_.append(T)
    d=2#决策树 max_depth

    #随机森林

RF=RandomForestClassifier(min_samples_leaf=int(0.1*len(X_train)),n_es
timators=T,n_jobs=-1)#所有核并行运算
    RF=RF.fit(X_train,Y_train)
    RF_acc=RF.score(X_test,Y_test)#准确率
    RF_acc_.append(RF_acc)

    #AdaBoost,

AB=AdaBoostClassifier(tree.DecisionTreeClassifier(min_samples_leaf=in
```

```
t(0.1*len(X_train))),n_estimators=T)
    AB=AB.fit(X_train,Y_train)
    AB_acc=AB.score(X_test,Y_test)#准确率
    AB_acc_.append(AB_acc)

    T=T+100


#画图
plt.figure()
plt.xlabel('T')
plt.ylabel('acc')
plt.plot(T_,RF_acc_,'go--')#随机森林，绿色
plt.plot(T_,AB_acc_,'b*--')#AdaBoost，蓝色
```
