

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **饶浩聪 谢家骏 闫明远**

**学 号 饶浩聪 201530612644**

**谢家骏 201530613191**

**闫明远** **201530613320**

**邮 箱 (饶浩聪)568302203@qq.com**

**指导教师**  **吴庆耀**

**提交日期** **2017年 12 月 10 日**

## 实验题目: 基于AdaBoost算法的人脸分类

## 2. 实验时间：2017年 12 月 9 日

## 3. 报告人: 饶浩聪 谢家骏 闫明远

## 4. 实验目的:

## 深入理解Adaboost的原理

## 熟悉人脸检测的基本方法

## 学会利用Adaboost解决人脸分类问题，将理论和实际工程接轨

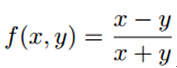
## 体验机器学习的完整过程

## 简述Adaboost原理：

 给定一个训练数据集T={(X1,y1), (X2,y2)…(XN,yN)}，yi属于标记集合{-1,+1}，

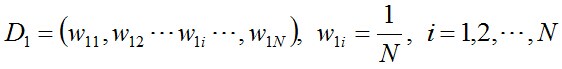
(此处Xi为经过处理后的24\*24灰度图的像素的NPD特征一维矩阵，大小为(24\*24)\*(24\*24-1)/2 =165600 )

   说明: NPD特征就是检测2个像素之间的差异性（有点类似Viola Johns），该差异定义为函数f(x,y)。其中，x，y为任意两个像素的像素值。并且规定f(0,0)=0。



Adaboost的目的就是从训练数据中学习一系列弱分类器或基本分类器，然后将这些弱    Adaboost的算法流程如下：

* **步骤**1. 首先，初始化训练数据的权值分布。每一个训练样本最开始时都被赋予相同的权值：1/N。

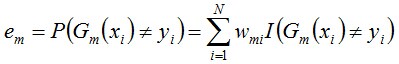


* **步骤**2. 进行多轮迭代，用m = 1,2, ..., M表示迭代的第多少轮

a. 使用具有权值分布Dm的训练数据集学习，得到基本分类器（选取让误差率最低的阈值来设计基本分类器）：

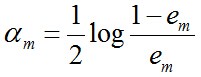
IMG_259

b. 计算Gm(x)在训练数据集上的分类误差率



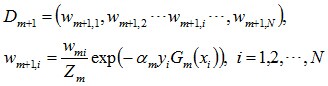
由上述式子可知，Gm(x)在训练数据集上的**误差率**em就是被Gm(x)误分类样本的权值之和。

c. 计算Gm(x)的系数，am表示Gm(x)在最终分类器中的重要程度（目的：得到基本分类器在最终分类器中所占的权重）：



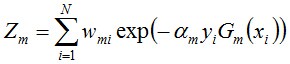
由上述式子可知，em <= 1/2时，am >= 0，且am随着em的减小而增大，意味着分类误差率越小的基本分类器在最终分类器中的作用越大。

d. 更新训练数据集的权值分布（目的：得到样本的新的权值分布），用于下一轮迭代

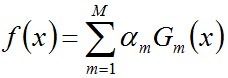


使得被基本分类器Gm(x)误分类样本的权值增大，而被正确分类样本的权值减小。就这样，通过这样的方式，AdaBoost方法能“重点关注”或“聚焦于”那些较难分的样本上。

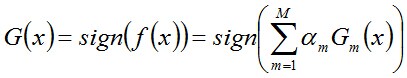
    其中，Zm是规范化因子，使得Dm+1成为一个概率分布：



* **步骤3.** 组合各个弱分类器



从而得到最终分类器，如下：



## 数据集以及数据分析：

本实验提供1000张图片，其中500张是含有人脸的RGB图片，储存在./datasets/original/face 内；另外500张是不含有人脸的RGB图，储存在./datasets/original/nonface 内。

## 7. 实验步骤:

## 1.读取数据集数据。读取图片，将数据转成大小为24\*24的灰度图，数据集正负类样本的个数和比例不限，数据集标签形式不限。

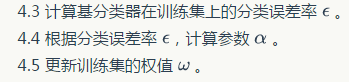
## 2.处理数据集数据，提取NPD特征。使用feature.py中NPDFeature类的方法提取特征。（提示：因为预处理数据集的时间比较长，可以用pickle库中的dump()函数将预处理后的特征数据保存到缓存中，之后可以使用load()函数读取特征数据）

## 3.将数据集切分为训练集和验证集，本次实验不切分测试集。

## 4.根据ensemble.py中的预留的接口编写AdaboostClassifier所有函数。以下为AdaboostClassifier类中的fit()方法的思路：

## 4.1 初始化训练集的权值 ,每一个训练样本被赋予相同的权值。

## 4.2 训练一个基分类器，基分类器可以使用sklearn.tree库中DecisionTreeClassifier(注意训练的时候需要将权重 作为参数传入)。



4.6 重复以上4.2-4.6的步骤进行迭代，迭代次数为基分类器的个数。

5.用AdaboostClassifier中的方法在验证集上进行预测并计算精确率,并用sklearn.metrics库的classification\_report()函数将预测结果写入report.txt中。

## 8. 代码内容:



## 9. 实验结果和曲线图:

## 超参数选择（弱分类器个数、决策树深度等）：

**弱分类器：2-10个，决策树深度最终确定使用2和4，理由如下**

**问题：**

**实验过程中，发现sklearn的决策树分类方法在不设置深度的时候分类效果十分优秀(可能是数据集过于简单)，可达100%，造成adaboost的迭代过程中error值为0而无法进行样本权值的更新以及分类器重要性的更新。**

**措施：**

**经不断测试，发现max\_depth(决策树深度)为6-7时，已经可以达到100%无误差的分类效果，不利于本实验的开展，因此，为了加深对迭代过程的理解，让多个基分类器的组合具有可能性，本实验过程中将决策树深度设置为2或4，一方面是为了让决策树(也就是基分类器)的效果最接近最佳，在实际应用也是采取这样的选择，另一方面能够充分应用进adaboost的实验中。**

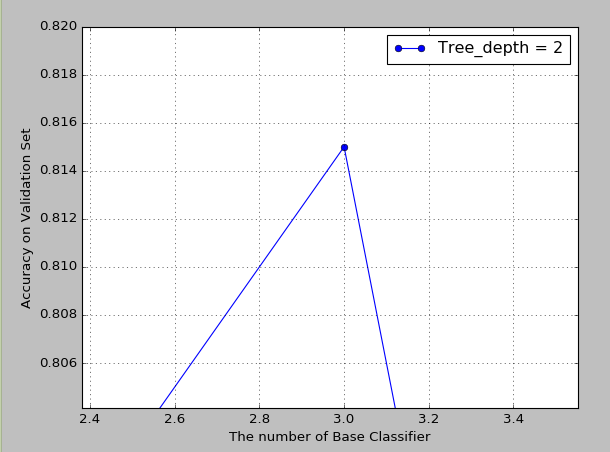
**对比：**

**在决策树深度为2和4的情况下，分别采用5、6、7、8、9、10数量的同一基分类器进行训练，对比最后模型的效果以及预测的准确率(将在后面给出统计图)**

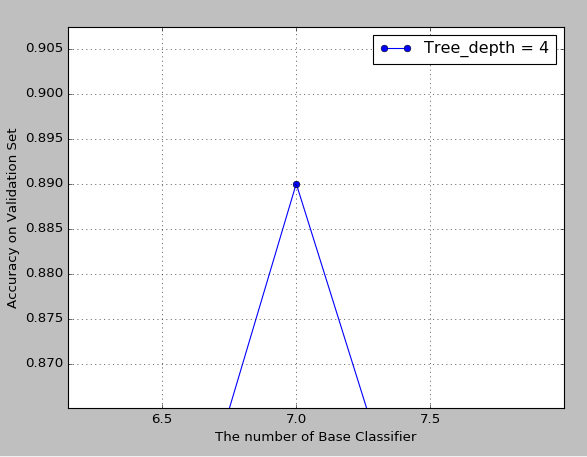
## 

## 预测结果（最佳结果）：(以下均在与训练集分开的200个样本的测试集上进行预测的准确率)

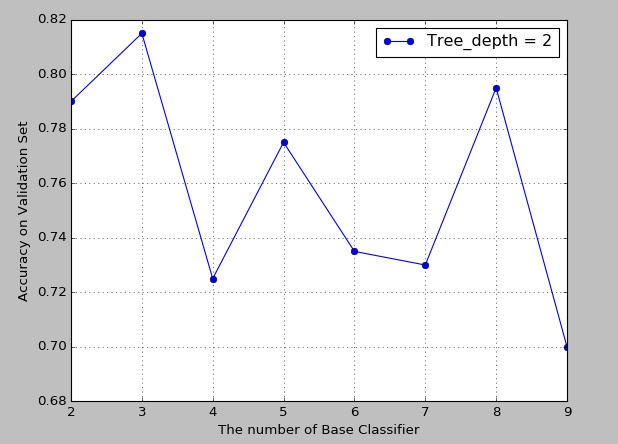
**深度为2的决策树在数量为7的时候有最大的准确率81.5%**

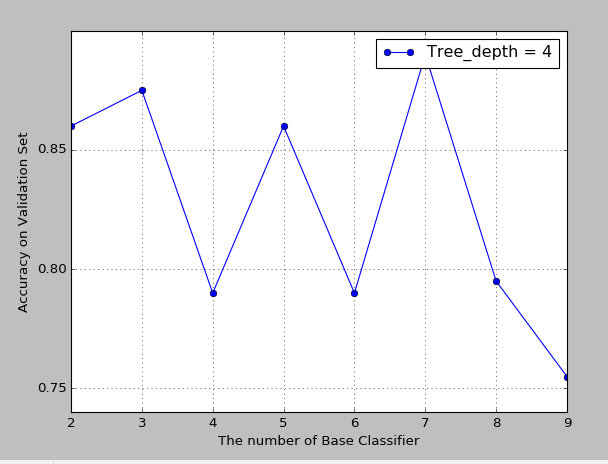


**深度为4的决策树在数量为7的时候有最大的准确率89.0%**



## 精度曲线图：(以下均在与训练集分开的200个样本的测试集上进行预测的准确率) 取2-9数量的深度为2或4的决策树进行对比





## 实验结果分析：

**由上面两图可以看出，决策树深度对模型的分类性能有一定作用。深度为4的决策树在使用相同数量的基分类器(本身)时，分类效果(图片识别效果)比深度为2的整体上好，深度越大，分类分支越长，判断能力越强，分类效果越好。**

**但在深度相同的情况下，基础分类器数量对模型分类效果影响并不是很显著(有可能是使用了决策树这个强的分类模型，可能用一些弱的模型例如线性分类可能会有较大不同)，所以在实际应用中，模型分类效果并不只是简单地提升分类器数量，而是要通过测试来确定最佳分类器数量，从而得到最优的模型。**

## 实验总结：

1. **本次实验通过最基础的数据集处理工作，包含图像的灰度处理，像素矩阵的表示，加深了对图像处理基本流程的理解，更学会了NPD特征的原理，简单高效地表示出一个图像中各个像素之间的差异性，从另一个角度来说就是完整地表示出了一张图的所有特征，巧妙。**
2. **本次实验尝试了把所有数据转换为numpy的数组格式来表示，在探索过程中发现了之前没有尝试过的好方法，比如：之前处理大规模文本时采用pickle存储，IO操作速度很慢，现在开始用numpy.save来高效存储了读取特征并转为npy格式文件，(虽然实验要求是pickle，但我个人觉得有必要改成numpy存储，尤其是大规模数据，这次实际上也挺大的=1000\*165600)**
3. **在feature.py文件里发现了\_\_calculate\_NPD\_table()这个类似于hash表的函数，这个提高效率的转换表思想以后可以借鉴**
4. **通过在AdaBoostClassifier类里添加各类方法来构建分类器，让我更加加深了面向对象编程的理解，以后自己打分类器也会采用这种高效规范的方法**
5. **Adaboost虽然是个简单的组合器思想，但是通过这次实验，以这个组合的思想为契机，更进一步地将理论和实践结合在一起，自己准备打****一个带有图形界面的可调参的可视化的图片训练&预测的程序(基于这个项目+Tkinter)**

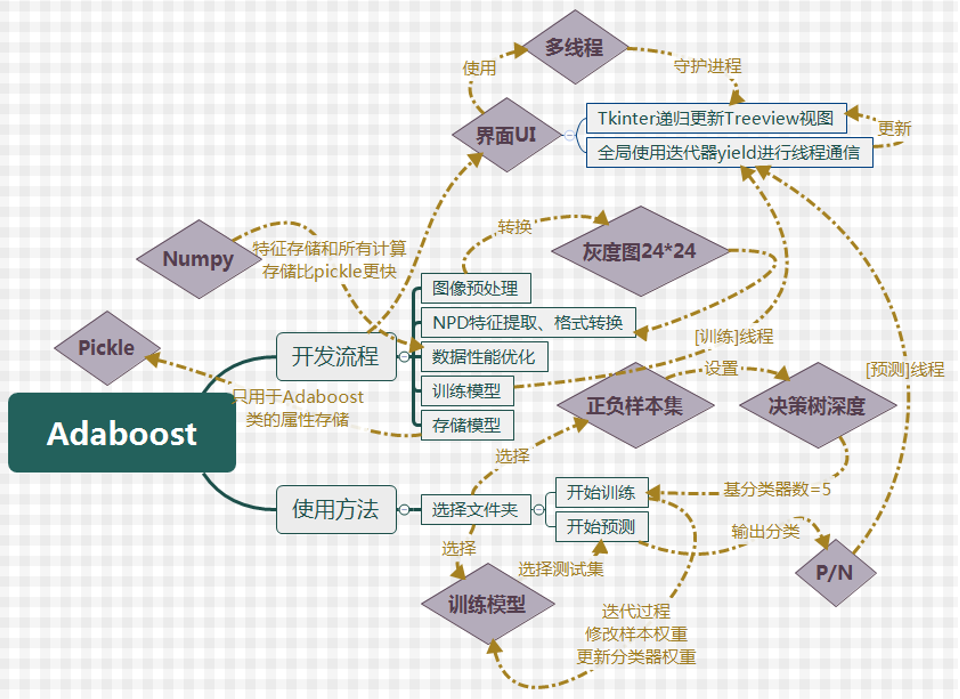
**我的项目地址为：(与实验地址分开)**

**<https://github.com/Kali-Hac/Adaboost_V/tree/dev_v1.x>**

**实验地址为：**

**<https://github.com/Kali-Hac/ML_experiments/tree/Lab3>**

**开发流程图(自己画的思维导图):**



实验参考资料：  
<https://zh.wikipedia.org/wiki/AdaBoost>

<http://blog.csdn.net/qq_14845119/article/details/52576902>

<http://blog.csdn.net/cv_family_z/article/details/50037323>

<http://blog.csdn.net/frog_in_a_well/article/details/30725341>

<http://cwiki.apachecn.org/pages/viewpage.action?pageId=10814387>

<http://www.jianshu.com/p/479e92cf4c2c>