*"""  
Created on 2018/5/2 Wed PM 22:56   
  
mian-component   
  
@author ： yuanyuan.liu@dsglyy.com  
  
Version : IdeaConfig V 0.0, May 02, 2018 DSG Exp$$*

*Update log ：*

*2018/5/2 Wed PM 23:48 梳理误差率的计算原理  
  
"""*

# Adaptive Boosting

对同一个训练集训练不同的[分类器](https://baike.baidu.com/item/%E5%88%86%E7%B1%BB%E5%99%A8)(弱分类器)，然后把这些弱分类器集合起来，构成一个更强的最终分类器（强分类器）。集成方法有多种形式：可以使多种算法的集成，也可以是一种算法在不同设置下的集成，还可以将数据集的不同部分分配不同的分类器，再将这些分类器进行集成。

### 原理概述

基本思想是通过训练数据的分布构造一个分类器，初始的时候每个样本赋予相同的权重，然后以后每次在整各训练数据集上迭代中，如果正确的分类则样本的权值会增加，错误分类权值会减少。最后，将各个训练得到的弱分类器组合成一个强分类器。各个弱分类器的训练过程结束后，加大分类误差率小的弱分类器的权重，使其在最终的分类函数中起着较大的决定作用，而降低分类误差率大的弱分类器的权重，使其在最终的分类函数中起着较小的决定作用。

### 算法推导

Adaboost 迭代算法分为3步：

1. 初始化训练数据的权值分布。如果有个样本，则每一个训练样本最开始时都被赋予相同的权值：。
2. 训练弱分类器。具体训练过程中，如果某个样本点已经被准确地分类，那么在构造下一个训练集中，它的权值就被降低；相反，如果某个样本点没有被准确地分类，那么它的权值就得到提高。然后，权值更新过的样本集被用于训练下一个分类器，整个训练过程如此迭代地进行下去。
3. 将各个训练得到的弱分类器组合成强分类器。各个弱分类器的训练过程结束后，加大分类误差率小的弱分类器的权重，使其在最终的分类函数中起着较大的决定作用，而降低分类误差率大的弱分类器的权重，使其在最终的分类函数中起着较小的决定作用。换言之，误差率低的弱分类器在最终分类器中占的权重较大，否则较小。

 给定训练数据集：，其中表示训练样本的类别标签。

* **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：。



* **步骤*2.*** 进行多轮迭代，用 表示迭代的轮次。

1. 使用具有权值分布的训练数据集学习，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）:



1. 计算在训练数据集上的分类误差率



其中为示性函数，当括号里面的条件满足时为1，不满足的时候为0 ；

上式解释，用分错的样本数除以总的样本数来估计分类器误差率，这里算的是样本数，可以把每个样本记1，在统计学中，计算比份时应该带上样本的权重，在这里分子为分错时1和权重之积，分母为所有样本的对应权重和 1 之积。这里为加权误差率。

1. 计算的系数，表示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）：



这里的对数为, 在计算机中

1. 更新训练样例的权重系数



1. 重复***a***到***d***。得到一系列的权重参数和基分类器
2. 构成及本分类器的线性组合



1. 最终的分类器就是：



AdaBoost 的训练误差界为



这里，，和 分别有以上的式子给出

**证明：**

当时，也就是分类器分错时，数据的实际类别和分类器算出的类别异号，所以，因而，又因为必定小于，所以上式的前部分得证；













因为受制于，所以由上式可知，选择合适的使得最小，从而使得训练误差下降的最快。

在二分类问题中Aadboost的训练误差界为



这里

**证明：**

由的定义式以及误差率得



在二分类中当时   可得（注意和的不同）：



当时  ，可得（注意和的不同）：



所以：



将 代入：



下面将用泰勒展开式子来证明：

### 泰勒公式

泰勒公式是将一个在处具有阶导数的函数利用关于的次[多项式](https://baike.baidu.com/item/%E5%A4%9A%E9%A1%B9%E5%BC%8F)来逼近函数的方法。

若函数在包含的某个闭区间上具有阶导数，且在开区间上具有阶[导数](https://baike.baidu.com/item/%E5%AF%BC%E6%95%B0" \t "_blank)，则对[闭区间](https://baike.baidu.com/item/%E9%97%AD%E5%8C%BA%E9%97%B4)上任意一点，成立下式：



其中，表示的阶导数，等号后的多项式称为函数在处的泰勒展开式，剩余的是泰勒公式的余项，是的高阶无穷小。 和  在出的泰勒展开为：









### 前向分布算法

加法模型



其中，为基函数，为基函数的参数，为基函数的系数。显然上式十一个加法模型。

在给定训练数据及损失函数的条件下，学习加法模型成为经验风险极小化即损失函数极小化问题：



这个问题为在上式取的最优解的情况的下的取值。这是一个复杂的优化问题。

前向分布算法（forward stagewise algorithm）求解这一问题的想法就是：因为学习的是加法模型，如果能够从前向后，每一步只学习一个基函数及其系数，逐步逼近优化目标函数：



给定训练数据集。损失函数和基函数的集合，学习加法模型的前向分部算法如下：

输入：训练数据集;损失函数；基函数；

输出：加法模型

1. 初始化
2. 对
3. 极小化损失函数



得到参数

1. 更新



1. 得到加法模型



这样，前向分步算法将同时求解从到所有参数的优化问题简化为逐次求解各个

的优化问题。

### 前向分步算法与Adaboost

Adaboost 中的加法模型为：



损失函数为指数函数：



前向分布算法下的模型为：



为了求解模型需要最小化损失函数，我们的目标是要最小化损失函数，通过最小化损失函数来得到模型中所需的参数。而在Adaboost算法中，每一轮都需要更新样例的权重参数，故而在每一轮的迭代中需要将损失函数极小化，然后据此得到每个样例的权重更新参数。这样在每轮的迭代过程中只需要将当前基函数在训练集上的损失函数最小即可（前向分步算法的思想）：



现在我们需要通过极小化上面的损失函数，得到 。设：



于是有：





在上式中 为已知量，为迭代的次数，只和之前的迭代有关，为本次的迭代的目的，在次可以先固定，先求的让损失函数最小的。

设：



对求偏导：



令，得：



两边同时去自然对数：



到此，我们在定义： 为数据的权值。先计算权值的更新：



对做规范化处理：





规范化后的权值保证了

有了权值之后，就可以得到加权误差率为 ：



因此为：



在 

在adaboost 中，在计算第轮的时，数据的权值还未更新，所以他的