1. 提升方法

8.1 提升方法和AdaBoost算法

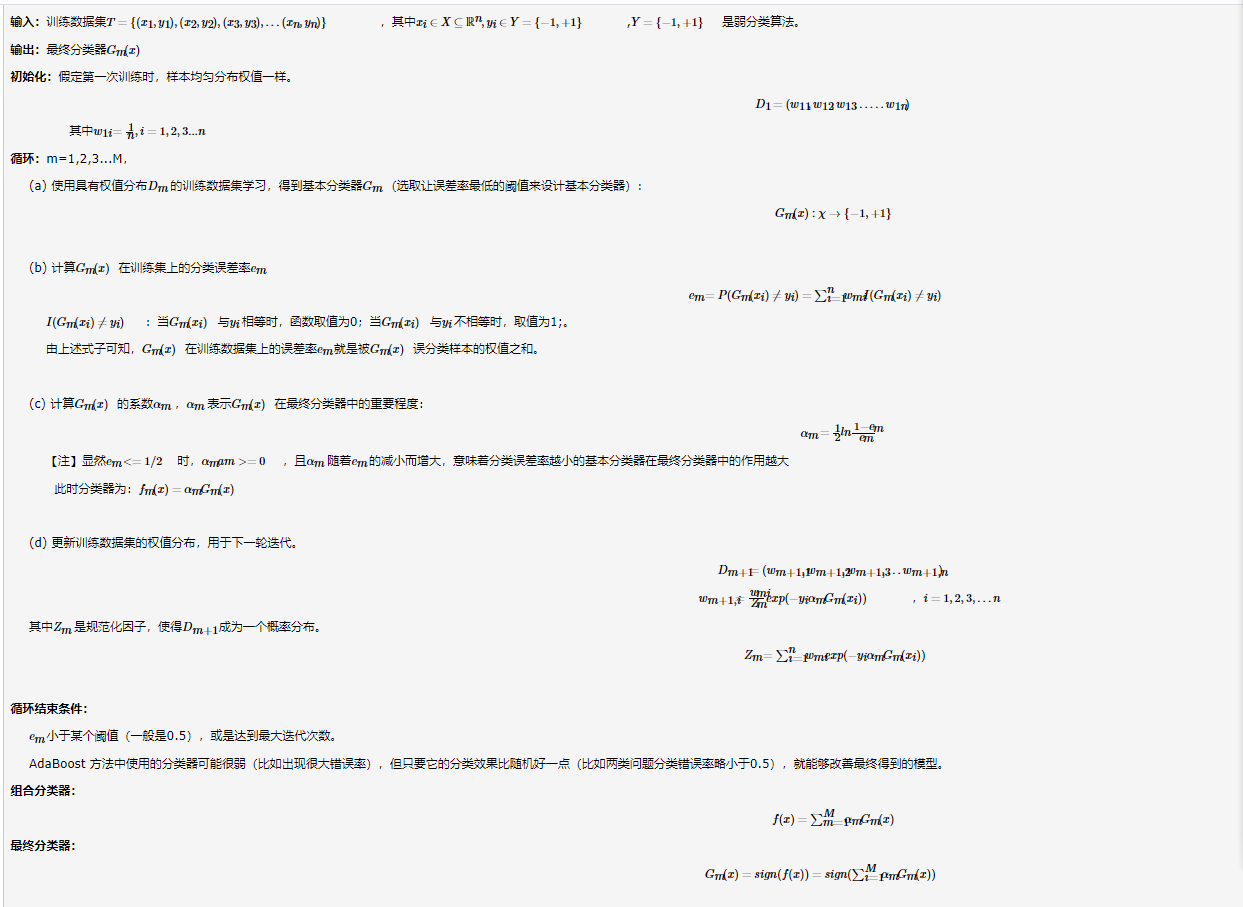
对于一个机器学习问题有如下两个概念：

强可学习：一个概念存在一个多项式的学习算法且他的正确率很高

弱可学习：一个概念存在一个多项式的学习算法且正确率比较低

强可学习和弱可学习是等价的

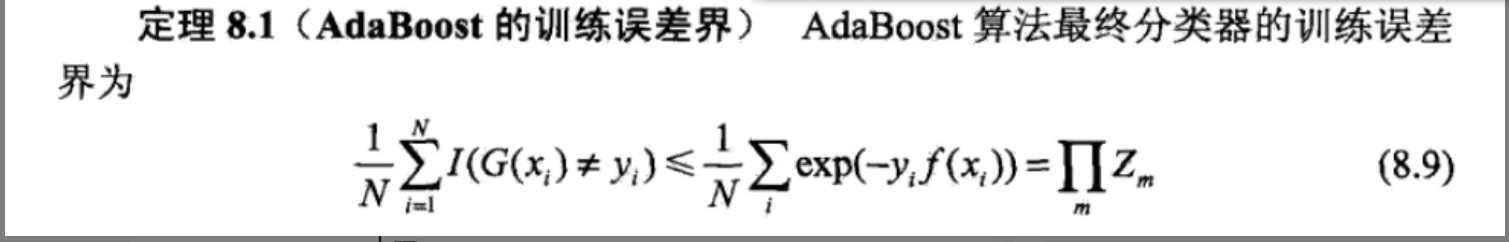
AdaBoost即为将弱可学习提升为强可学习的算法（对与分类问题而言，给定一个训练样本集，求比较粗糙的分类规则（弱分类器）要比求精确的分类规则（强分类器）容易得多。）

算法的流程如下：

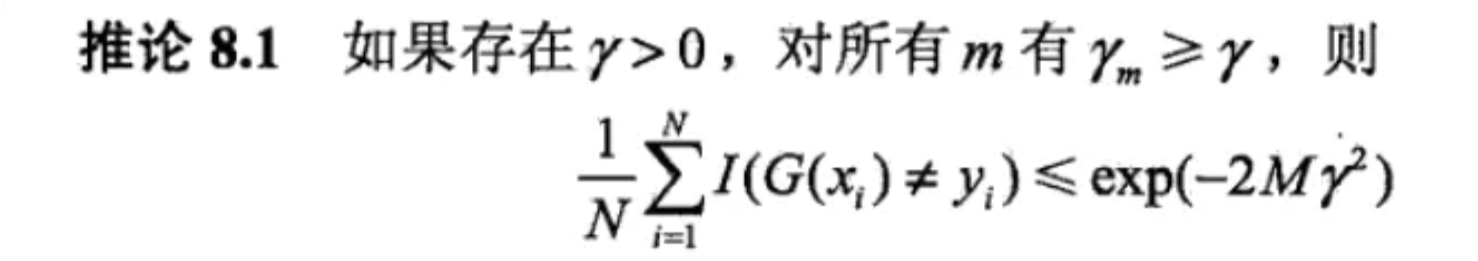
而AdaBoosy的两个基本思想是①提高那些被前一轮弱分类器错误分类的样本的权值，降低那些被正确分类的样本的权值；②采用加权多数表决的方法。具体的，加大分类误差率小的弱分类器的权值，使其在表决中起较大的作用；减小分类误差率大的弱分类器的权值，使其在表决中起较小的作用。

8.2 AdaBoost误差分析

AdaBoost最基本的性质是它能在学习过程中不断减少训练误差，即在训练数据上的分类误差率，关于这个问题有下面的定理：



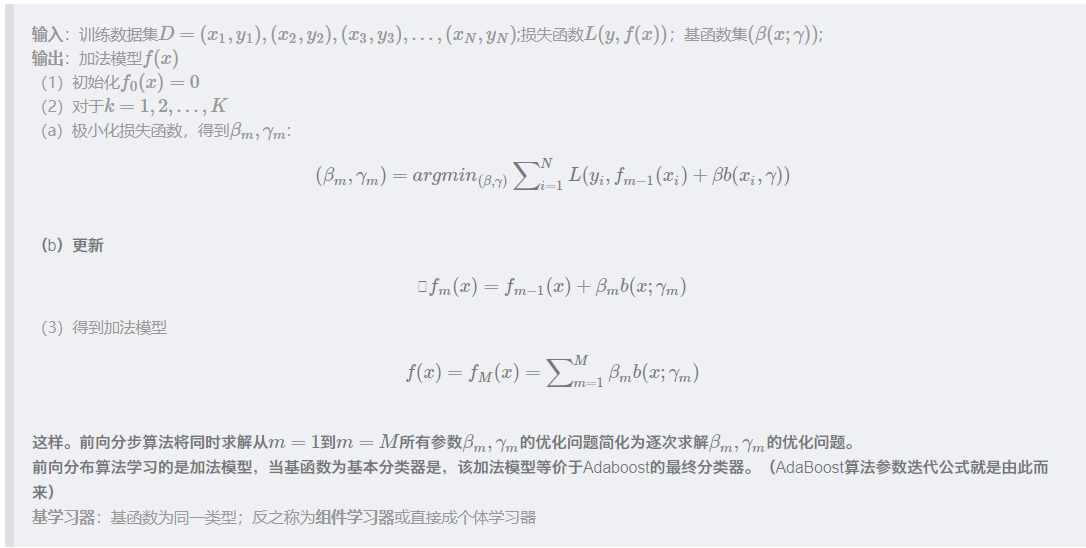
且给出了证明，我们可以发现



在此条件下，Adaboost的训练误差是指数速率下降的，为什么呢？注意到rm=1-em，误差越小，rm越大，则误差上界以指数速率减小

8.3 AdaBoost算法的解释

可以看出Boosting算法要涉及到两个部分，加法模型和前向分步算法： 加法模型就是说强分类器由一系列弱分类器线性相加而成。前向分布就是说在训练过程中，下一轮迭代产生的分类器是在上一轮的基础上训练得来的。

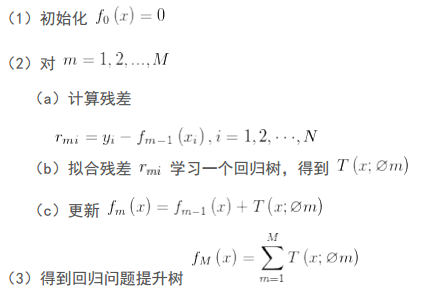


因为学习的是加法模型，如果能够从前向后，每一步只学习一个基函数及其系数，逐步逼近优化目标函数式f(x)=∑Mm=1βmb(x;γm)f(x)=∑m=1Mβmb(x;γm)，那么就可以简化优化的复杂度。具体地，每步只需优化如下损失函数:minβ,γ∑Mi=1L(yi,βb(xi;γ))

8.4 提升树

提升树是以分类树或回归树为基分类器。

提升树的算法如下：



对于一半的函数，我们希望损失函数最小化，为了达到这种目的我们采用梯度提生算法

梯度提升算法的内容：算法将损失函数的负梯度在当前模型的值(梯度负方向是函数值局部下降最快的方向)，作为回归问题中提升树算法的残差的近似值，拟合一个回归树。这个算法主要是给出了一个新的残差的公式，其余仍和之前回归树的算法拟合算法一致

1、初始化，估计使损失函数极小化的常数值，它是只有一个根节点的树，即ganma是一个常数值。

2、

（a）计算损失函数的负梯度在当前模型的值，将它作为残差的估计

（b）估计回归树叶节点区域，以拟合残差的近似值

（c）利用线性搜索估计叶节点区域的值，使损失函数极小化

（d）更新回归树

3、得到输出的最终模型 f(x)