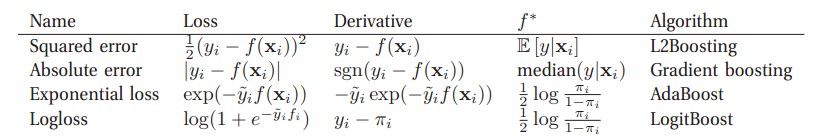
集成式学习补充

* 1. 什么是集成式学习 (boosting and bagging)

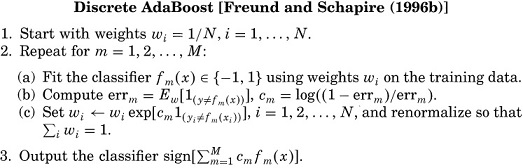
提升方法是基于这样一种思想:对于一个复杂任务来说,将多个专家的判断进行适当的综合 所得出的判断,要比其中任何一个专家单独的判断好。通俗点说,就是“三个臭皮匠顶个诸葛亮”。

* 1. Boosting家族

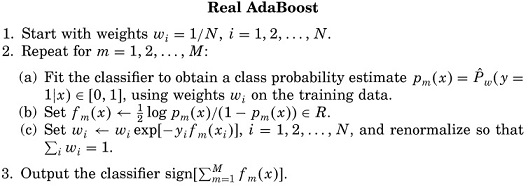
在1.2小节中，我们介绍了adaboost算法的基本流程，而在1.3小节中我们通过损失函数的概念推导出了adaboost算法的数学原理，从本质上看，adaboost将可以用指数损失函数定义，而在每次添加的弱分类器的权重是通过最小化指数损失函数得到。一般将上述adaboost称之为Discrete boost算法，它表示一种基于指数损失函数的二分类问题。当然我们可以通过不通过的损失函数得到不同的boosting算法，将这种基于梯度最小化方法得到boosting方法称之为GradientBoosting算法， 

不同Gradient Boosting算法的计算流程：

Discrete Adaboost算法的步骤如下：

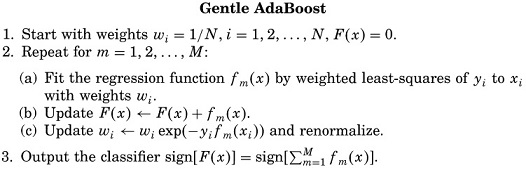


　　可以看出，Discrete AdaBoost的每一个弱分类的输出结果是1或-1，并没有属于某个类的概率，略显粗糙。如果让每个弱分类器输出样本属于某个类的概率，则可以得到Real AdaBoost算法，其步骤如下：



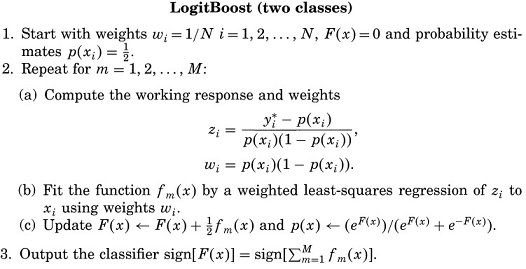
　　Real Adaboost每个弱分类器输出样本属于某类的概率后，通过一个对数函数将0-1的概率值映射到实数域，最后的分类器是所有映射函数的和。

将Real Adaboost算法每次迭代的两部合并，直接产生一个映射到实数域的函数，则就成了Gentle AdaBoost， 其算法步骤如下：



　　Gentle AdaBoost则在每次迭代时，基于最小二乘去做一个加权回归，最后所有回归函数的和作为最终的分类器。

　　LogitBoost算法则和Gentle AdaBoost算法有点相像，不过其每次进行回归拟合的变量z是在不断更新的，Gentle AdaBoost使用的是y。LogitBoost算法步骤如下：



最后总结一下GradientBoosting算法的基本流程：

