Adaboost的执行流程如下：

训练数据中的每个样本，并赋予其一个权重，这些权重构成了向量D。一开始，这些权重都初始化成相等值，首先在训练数据上训练出一个弱分类器并计算该分类器的错误率，然后在同一数据集上再次训练弱分类器。在分类器的第二次训练当中，将会重新调整每个样本的权重，其中第一次分对的样本的权重将会降低，而第一次分错的样本的权重将会提高。

为了从所有的弱分类器中得到最终的分类结果，AdaBoost为每个分类器都分配了一个权重值alpha，这些alpha值是基于每个弱分类器的错误率进行计算的。其中，错误率的定义为：

而alpha的计算公式如下：



计算出alpha值之后，可以对权重向量D进行更新，以使得那些正确分类的样本的权重降低而错分样本的权重升高。D的计算方法如下：

如果某个样本被正确分类，那么该样本的权重更新为：



而如果某个样本被错分，那么该样本的权重更改为：



完整公式：



在计算出D之后，AdaBoost又开始进行下一轮迭代。AdaBoost算法会不断地重复训练和调整权重的过程，直到训练错误率为0或者弱分类器的数目达到用户的指定值为止。

弱分类器：单层决策树

|  |
| --- |
| 1. 将最小错误率miniError设为无穷大 2. 对数据集中的每一个特征（第一层循环）： 3. 对每个步长（第二层循环）： 4. 对每个不等号（第三层循环）： 5. 建立一棵单层决策树并利用加权数据集对它进行测试 6. 如果错误率低于minError，则将当前单层决策树设为最佳单层决策树 7. 返回最佳单层决策树 |

结果的决定方式：类概率（软投票）

算法描述：

|  |
| --- |
| **Input:** *训练*  基学习算法  训练轮数T  **Output:** |
| 1. (初始化权重分布) 2. For t = 1,2,…,T do 3. ; 4. ； 5. If then break 7. Calculate the id[j] of substring, (id[j],pos[j])=***POS\_MAP*** (superS,SA[j],|r[i]|); 8. If(id[j] is not in S\* && LenL+LenR<=tor)\\位置 9. Calculate tor\*=***ED***(r[i]c,idj,posj) assuming r[i] corresponds id[j] at pos[j]; 10. If(tor\*<=tor)\\证是确认 11. put id[j] into S\*; 12. Return S\*; |