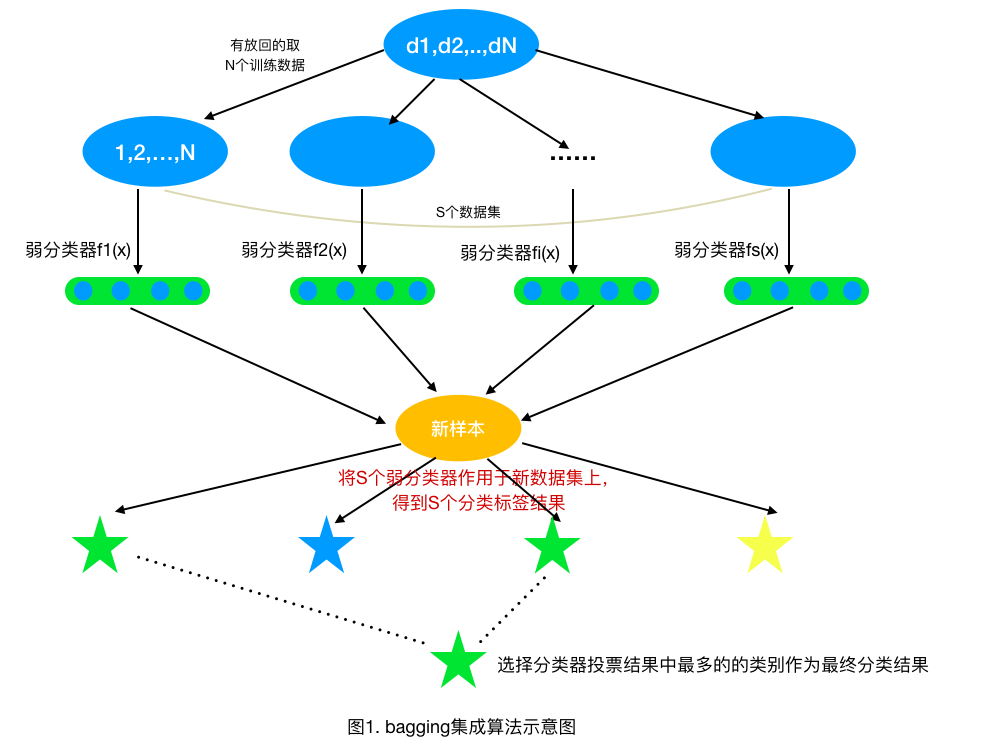
我们学过决策树、朴素贝叶斯、SVM、K近邻等分类器算法，他们各有优缺点；自然的，我们可以将这些分类器组合起来成为一个性能更好的分类器，这种组合结果被称为 **集成方法** (ensemble method)或者 **元算法** (meta-method)。使用集成算法时有多种形式：

* 不同算法的集成
* 同一种算法在不同设置下的集成
* 数据集不同部分分配 给不同分类器之后的集成

**1、bagging 和boosting综述**

bagging 和boosting中使用的分类器类型都是一样的,即上述第二种形式。

**bagging**，也称为自举汇聚法（boostrap aggegating) 是在原始数据集中有放回的选择S次后得到S个新数据集的一种技术。新数据集和原数据集大小相等，但是有可能某一条数据被选择了好几次，而原数据集中某些数据在新数据集中可能不出现。在S个数据集建好之后，将某个算法分别作用于每个数据集就得到S个分类器。对新数据集进行分类时，就用这S个分类器进行分类，与此同时，选择分类器投票结果中最多的的类别作为最终分类结果，如图1所示。Random Forests是一种更先进的bagging算法，下文详细介绍。



**boosting** 与bagging很类似，不同的是Boosting是通过串行训练而获得的，而每个新分类器都是根据已经训练好的分类器的性能来进行训练的。AdaBoost是这一种常用的boosting方法。

**2、一种提升算法：AdaBoost**

在概率近似正确的学习框架（probably approximately corect,PAC）中，一个概念，如果存在一个多项式的学习算法能够学习他，并且正确率很高，在统计学习方法中，称这个概念是 **强可学习(strongly learnable)**的；而如果正确率仅仅比随机猜测（正确率大于0.5）略好，那么称这个概念是 **弱可学习(weakly learnable)** 的。然而Schapire证明了再PAC学习框架下，一个概念是强可学习的充分必要条件是这个概念是弱可学习的。

那么也就是说，我们可以将“弱学习算法”提升为“强学习算法”，毕竟弱学习算法比强学习算法要好找的多了。问题是怎么来提升呢？有很多算法，最具代表性的就是AdaBoost算法。大多数提升方法都是改变训练数据的概率分布（训练数据的权值分布）。

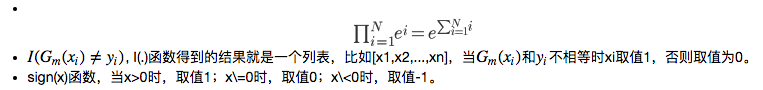
因此，对提升算法有两个问题需要回答：

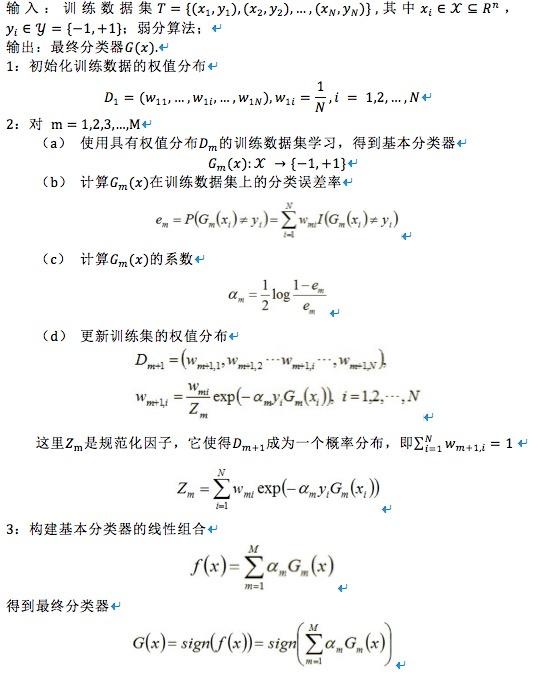
1. 在每一轮如何改变训练数据的权值或者概率分布
2. 如何将若干分类器组合成一个强分类器

对于第一个问题AdaBoost通过集中关注那些错分的数据，即将错分的数据赋予较大的权重，没错分的数据赋予较小的权重，然后再将这有具有新权重的数据集进行训练，从而来获得新分类器；对于第二个问题，AdaBoost采取加权多数表决的方法。

**2.1、AdaBoost算法**

首先提一下几个数学表达式的意思，当做本小节的先验知识吧。

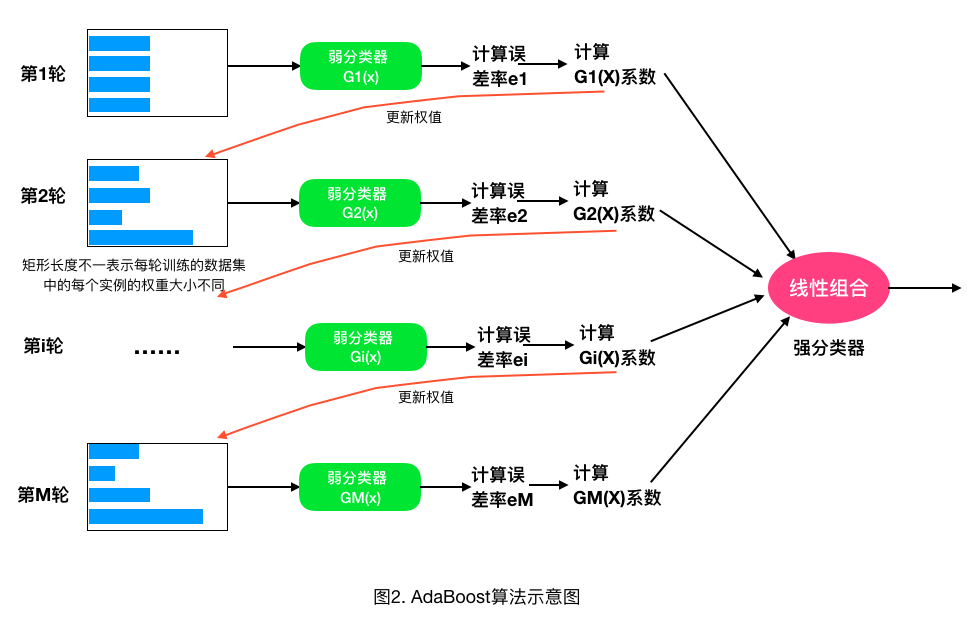


AdaBoost算法  


在上述算法中，值得注意的是：

* 在2（c）中，由弱分类器的系数计算公式可知，在当前的弱分类器分类后，其分类误差率em如果小于0.5，系数$\alpha \_{m}$将大于0，并且系数随着误差率em的减少而增大，所以分类误差率小的基本分类器在最终分类器中的作用越大。
* 在2（d）中，更新权值分布时，被基本分类器误分类的样本的权值得以扩大，而被正确分类的样本的权值得以缩小，即误分类样本在下一次训练中起更大的作用。
* 在步骤（3）中，f(x)的符号决定实例x的类，它的绝对值表示分类的确信度。**利用基本分类器的线性组合构建最终分类器是AdaBoost的一个特点**

AdaBoost算法的示意图如图2所示。



**2.2、AdaBoost算法的误差分析**

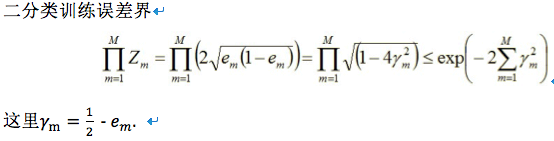
AdaBoost最基本的性质是它能在学习的过程中不断的减少训练误差，即在训练数据集上的分类误差率。那么训练误差是不是能无限制的减少呢？AdaBoost训练误差界定理回答了这个问题。

AdaBoost算法最终分类器的训练误差界为：  
http://images2015.cnblogs.com/blog/1098949/201707/1098949-20170721172042199-1805180371.png

推导从略，见《统计学习方法》一书。

这一定理说明，**可以在每一轮选取适当的Gm使得Zm最小，从而使训练误差下降最快。**

对于二类分类问题有：



AdaBoost的训练误差是以指数速率下降的。AdaBoost具有适应性，即它能适应弱分类器各自的训练误差率，这也是它名称的由来，即adaptive。

**3、Random Forests（RF）**

RF在实际中使用非常频繁，其本质上和bagging并无不同，只是RF更具体一些。一般而言可以将RF理解为**bagging和DT（CART）的结合**。RF中的基学习器使用的是CART树，由于算法本身能降低方差(variance)，所以会选择完全生长的CART树。抽样方法使用bootstrap，除此之外，RF认为随机程度越高，算法的效果越好。所以RF中还经常随机选取样本的特征属性、甚至于将样本的特征属性通过映射矩阵映射到随机的子空间来增大子模型的随机性、多样性。**RF预测的结果为子树结果的平均值**。RF具有很好的降噪性，相比单棵的CART树，RF模型边界更加平滑，置信区间也比较大。一般而言，RF中，树越多模型越稳定。

**3.1 随机森林算法**

**随机森林训练过程如下：**

(1)给定训练集S，测试集T，特征维数F。确定参数：使用到的CART的数量t，每棵树的深度d，每个节点使用到的特征数量f，终止条件：节点上最少样本数s，节点上最少的信息增益m  
对于第1-t棵树，i=1-t：  
(2)从S中有放回的抽取大小和S一样的训练集S(i)，作为根节点的样本，从根节点开始训练  
(3)如果当前节点上达到终止条件，则设置当前节点为叶子节点，如果是分类问题，该叶子节点的预测输出为当前节点样本集合中数量最多的那一类c(j)，概率p为c(j)占当前样本集的比例；如果是回归问题，预测输出为当前节点样本集各个样本值的平均值。然后继续训练其他节点。如果当前节点没有达到终止条件，则从F维特征中无放回的随机选取f维特征。利用这f维特征，寻找分类效果最好的一维特征k及其阈值th，当前节点上样本第k维特征小于th的样本被划分到左节点，其余的被划分到右节点。继续训练其他节点。有关分类效果的评判标准在后面会讲。  
(4)重复(2)(3)直到所有节点都训练过了或者被标记为叶子节点。  
(5)重复(2),(3),(4)直到所有CART都被训练过。

**利用随机森林的预测过程如下：**

对于第1-t棵树，i=1-t：  
(1)从当前树的根节点开始，根据当前节点的阈值th，判断是进入左节点(\

**参考资料**

1. 《统计机器学习》，李航
2. 《机器学习实战》，Peter
3. <http://www.cnblogs.com/hrlnw/p/3850459.html>