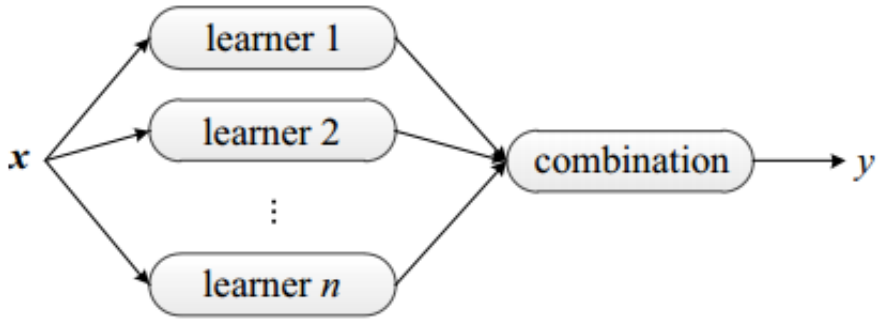
机器学习算法系列（6）：AdaBoost

一、集成学习

1.1 定义

所谓集成学习（ensemble learning），是指通过构建多个弱学习器，然后结合为一个强学习器来完成分类任务。并相较于弱分类器而言，进一步提升结果的准确率。严格来说，集成学习并不算是一种分类器，而是一种学习器结合的方法。

下图显示了集成学习的整个流程：首次按产生一组“个体学习器”，这些个体学习器可以是同质的（homogeneous）（例如全部是决策树），这一类学习器被称为基学习器（base learner），相应的学习算法称为“基学习算法”；集成也可包含不同类型的个体学习器（例如同时包含决策树和神经网络），这一类学习器被称为“组件学习器”（component learner）。



集成学习通过将多个学习器进行结合，可获得比单一学习器显著优越的泛化性能，它基于这样一种思想：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，直观一点理解，就是我们平时所说的“三个臭皮匠，顶个诸葛亮”，通过使用多个决策者共同决策一个实例的分类从而提高分类器的泛化能力。

1.2 集成学习的条件

当然，这种通过集成学习来提高学习器（这里特指分类器）的整体泛化能力也是有条件的：

- 首先，分类器之间应该具有差异性，即要有“多样性”。很容易理解，如果使用的是同一个分类器，那么集成起来的分类结果是不会有变化的。‘
- 其次，每个个体分类器的分类精度必须大于0.5，如果$p < 0.5$那么随着集成规模的增加，分类精度会下降；但如果是大于0.5的话，那么最后最终分类精度是可以趋于1的。

因此，要获得好的集成，个体学习器应该“好而不同”，即个体学习器要有一定的“准确性”，即学

习器不能太坏，并且要有“多样性”，即学习器间具有差异。

1.3 集成学习的分类

当前，我们可以立足于通过处理数据集生成差异性分类器，即在原有数据集上采用抽样技术获得多个训练数据集来生成多个差异性分类器。根据个体学习器的生成方式，目前集成学习方法大致可分为两大类：第一类是个体学习器之间存在强依赖关系、必须串行生成的序列化方法，这种方法的代表是“Boosting”；第二类是个体学习器间不存在强依赖关系、可同时生成的并行化方法，它的代表是“Bagging”和“Random Forest”

- **Bagging**：通过对原数据进行有放回的抽取，构建出多个样本数据集，然后用这些新的数据集训练多个分类器。因为是有放回的采用，所以一些样本可能会出现多次，而其他样本会被忽略。该方法是通过降低基分类器方差来改善泛化能力，因此Bagging的性能依赖于基分类器的稳定性，如果基分类器是不稳定的，Bagging有助于减低训练数据的随机扰动导致的误差，但是如果基分类器是稳定的，即对数据变化不敏感，那么Bagging方法就得不到性能的提升，甚至会降低。
- **Boosting**：提升方法是一个迭代的过程，通过改变样本分布，使得分类器聚集在那些很难分的样本上，对那些容易错分的数据加强学习，增加错分数据的权重，这样错分的数据再下一轮的迭代就有更大的作用（对错分数据进行惩罚）。
- **Bagging与Boosting的区别：**
  - 二者的主要区别是取样方式不同。Bagging采用均匀取样，而Boosting根据错误率来取样，因此Boosting的分类精度要优于Bagging。Bagging的训练集的选择是随机的，各轮训练集之间相互独立，而Boostlng的各轮训练集的选择与前面各轮的学习结果有关；Bagging的各个预测函数没有权重，而Boosting是有权重的；Bagging的各个预测函数可以并行生成，而Boosting的各个预测函数只能顺序生成。对于象神经网络这样极为耗时的学习方法。Bagging可通过并行训练节省大量时间开销。
  - bagging是减少variance，而boosting是减少bias。Bagging 是 Bootstrap Aggregating 的简称，意思就是再取样 (Bootstrap) 然后在每个样本上训练出来的模型取平均，所以是降低模型的 variance. Bagging 比如 Random Forest 这种先天并行的算法都有这个效果。Boosting 则是迭代算法，每一次迭代都根据上一次迭代的预测结果对样本进行加权，所以随着迭代不断进行，误差会越来越小，所以模型的 bias 会不断降低。这种算法无法并行。

二、AdaBoost算法

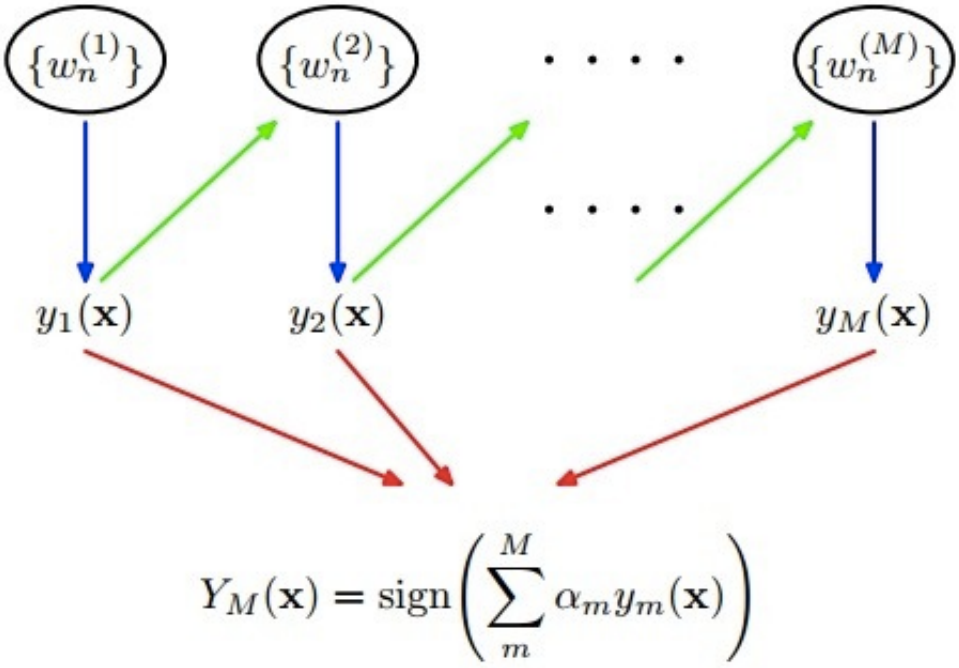
2.1 AdaBoost算法思想

对于分类问题而言，给定一个训练样本集，求比较粗糙的分类规则（弱分类器）要比求精确地分类规则（强分类器）容易得多。提升算法就是从弱学习算法出发，反复学习，得到一系列弱分类

器（又称为基本分类器），然后组合这些弱分类器，构成一个强分类器。大多数的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），针对不同的训练数据分布调用弱学习算法学习一系列弱分类器。

这样，对提升方法来说，有两个问题需要回答：一是在每一轮如果改变训练数据的权值或概率分布；二是如何将弱分类器组合成一个强分类器。对于第一个问题，AdaBoost的做法是，提高那些被前一轮弱分类器错误分类样本的权值，而降低那些被正确分类样本的权值。这样一来，那些没有得到正确分类的数据，由于其权值的加大而受到后一轮的弱分类器的更大关注，于是，分类问题就被一系列的弱分类器“分而治之”。至于第二个问题，即弱分类器的组合，AdaBoost采取加权多数表决的方法。具体地，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，减小分类误差率较大的弱分类器的权值，使其在表决中起较小的作用。

AdaboostBoost的算法的框架如下图所示



具体来说，整个AdaBoost算法包括以下三个步骤：

- 1）**初始化训练样本的权值分布。**如果有N个样本，则每一个训练样本最开始时都被赋予相同的权值：$1/N$。
- 2）**训练弱分类器。**具体训练过程中，如果某个样本已经被准确地分类，那么在构造下一个训练集中，它的权值就会被降低；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，权值更新过的样本被用于训练下一个分类器，整个训练过程如果迭代地进行下去，使得分类器在迭代过程中逐步改进。
- 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权重较大，否则较小。得到最终分类器。

2.2 AdaBoost算法流程

现在叙述AdaBoost算法。假定给定一个二类分类的训练数据集

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$$

其中y_i属于二分类的标记组合，即$y_i \in \{+1, -1\}$，AdaBoost算法利用一下算法，从训练数据中学习一系列弱分类器或基本分类器，并将这些弱分类器线性组合成一个强分类器。

步骤一：首先，初始化训练数据的权值分布。假设每一个训练样本最开始时都被赋予相同的权值：$1/N$，即每个训练样本在基本分类器的学习中作用相同，这一假设保证步骤一能够在原始数据上学习基本分类器$G_1(x)$，数学化的语言表示为：

$$D_1 = \left(w_{11}, w_{12}, \cdots, w_{1i}, \cdots, w_{1N}\right), \ w_{1i} = \frac{1}{N}, i = 1, 2, \cdots, N$$

步骤二：AdaBoost反复学习基本分类器，在每一轮$m = 1, 2, \cdots, M$顺次执行下列操作：

- 1）使用当前权值分布为D_m的训练数据集，学习得到基分类器

$$G_m(x): \chi \to \{-1, +1\}$$

- 2）计算上一步得到的基分类器$G_m(x)$在训练数据集上的分类误差率e_m为

$$e_m = P\left(G_m(x) \neq y_i\right) = \frac{\sum_{i=1}^{N} w_{mi} I\left(G_m\left(x_i\right) \neq y_i\right)}{\sum_{i=1}^{N} w_{mi}} = \sum_{i=1}^{N} w_{mi} I\left(G_m\left(x_i\right) \neq y_i\right)$$

  这里w_{mi}表示第m轮中第i个实例的权值，$\sum_{i=1}^{N} w_{mi} = 1$。这表明，$G_m(x)$在加权的训练数据集上的分类误差率是被$G_m(x)$误分类样本的权值之和，由此可以看出数据权值分布$D_m$与基本分类器$G_m(x)$的分类误差率的关系。
- 3）计算G_m前面的权重系数a_m，该系数表示G_m在最终分类器中的重要程度，目的在于使我们得到基分类器在最终分类器中所占的权值，系数计算公式如下：

$$a_m = \frac{1}{2}\log\frac{1 - e_m}{e_m}$$

  这里的对数是自然对数，由表达式可知，当$e_m \leq \frac{1}{2}$时，$a_m \geq 0$，并且a_m随着e_m的减小而增大，意味着分类误差越小的基本分类器在最终分类器的作用越大，而$e_m \geq \frac{1}{2}$则刚好相反，这正好验证了集成学习中每个个体分类器的分类精度必须大于0.5的前提条件。

- 4）更新训练数据集的权值分布为下一轮作准备

$$D_{m+1} = \left(w_{m+1,1}, w_{m+1,2}, \cdots, w_{m+1,i}, \cdots, w_{m+1,N}\right)$$

其中

$$w_{m+1,i} = \frac{w_{mi}}{Z_m}\exp\left(-\alpha_m y_i G_m\left(x_i\right)\right), i = 1, 2, \cdots, N$$

我们也可以写成：

$$w_{m+1,i} = \begin{cases} \frac{w_{mi}}{Z_m}e^{-a_m}, & G_m\left(x_i\right) = y_i \\ \frac{w_{mi}}{Z_m}e^{a_m}, & G_m(xi) \neq y_i \end{cases}$$

由此可知，被基本分类器$G_m(x)$误分类样本的权值得以扩大（$a_m > 0$，则$1 < e^{a_m}$），而被正确分类样本的权值得以缩小（$a_m > 0$，则$0 < e^{-a_m} < 1$）。两两比较，误分类样本的权值$e^{2a_m} = \frac{e_m}{1-e_m}$倍。因此，误分类样本在下一轮学习中起更大的作用。不改变所给的训练数据，而不断改变训练数据权值的分布，使得训练数据在基本分类器的学习中起不同的作业，这是AdaBoost的一个特点。 这里我们还引入了一个规范化因子，它的作用在于使D_{m+1}成为一个概率分布。具体公式为

$$Z_m = \sum_{i=1}^{N} w_{mi}\exp\left(-\alpha_m y_i G_m\left(x_i\right)\right), i = 1, 2, \cdots, N$$

重复步骤二中的1至4步骤，得到一系列的权重参数a_m和基分类器G_m

步骤三：将上一步得到的基分类器根据权重参数线性组合

$$f(x) = \sum_{m=1}^{M} a_m G_m(x)$$

得到最终分类器$G_{(x)}$

$$G(x) = sign(f(x)) = sign\left(\sum_{m=1}^{M} a_m G_m(x)\right)$$

线性组合$f(x)$实现了M个基本分类器的加权表决。系数a_m表示了基本分类器$G_m(x)$的重要性，这里，所有的a_m之和并不为1。$f(x)$的符号决定实例x的类，$f(x)$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的另一特点。

2.3 AdaBoost算法的一个实例

下图为给定的训练样本，假设$Y \in \{+1, -1\}$，且弱分类器由$x < v$或$x > v$产生（v为阈值，目的在于使分类器在训练样本上的分类误差率最低），接下来我们就要使用AdaBoost算法得到一个强分类器。

序号	1	2	3	4	5	6	7	8	9	10
x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1

首先，初始化训练数据的权值分布，得到：

$$D_1 = (w_{11}, w_{12}, w_{1,10}), w_{1i} = \frac{1}{10}, i = 1, 2, \cdots, 10$$

在此基础上，开始M轮迭代。
根据X和Y的对应关系，要把这10个数据分为两类，一类是1，一类是-1，根据数据的特点发现：（0，1，2，6，7，8）对应的类是1，$(3, 4, 5, 9)$对于的类是-1，抛开孤独的9不说，$(0, 1, 2), (3, 4, 5), (6, 7, 8)$这是3类不同的数据，分别对应的类是$(1, -1, 1)$,直观上推测可知，可以找到对应的数据分界点，比如2.5 、 5.5 、 8.5，将这几类数据分成两类。

1.第一次迭代（m=1）：

- 1）在第一次迭代时，已知$w_{1i} = \frac{1}{10}$，经过计算可得：在权值分布为D_1的训练数据上，阈值v取2.5或8.5时分类误差率为0.3，取5.5时分类误差率为0.4，遵循分类误差率最低原则，从2.5或8.5 中任意选取一个阈值，这里选取2.5，故基本分类器为

$$G_1(x) = \begin{cases} 1 & x < 2.5 \\ -1 & x > 2.5 \end{cases}$$

- 2）$G_1(x)$在训练集上的误差率：

$$e_1 = P(G_1(x_i) \neq y_i) = 0.3$$

- 3) 根据e_1计算得到$G_1(x)$的系数：

$$a_1 = \frac{1}{2}\log\frac{1 - e_1}{e_1} = 0.4236$$

这个系数就代表$G_1(x)$在最终的分类函数中所占的权值。

- 4）更新训练数据的权值分布，用于下一轮迭代

$$D_{m+1} = \left(w_{m+1,1}, w_{m+1,2}, \cdots, w_{m+1,i}, \cdots, w_{m+1,N}\right)$$

其中

$$w_{m+1,i} = \frac{w_{mi}}{Z_m}\exp\left(-\alpha_m y_i G_m\left(x_i\right)\right), i = 1, 2, \cdots, N$$

由此得到$D_2 = (0.0715, 0.0715, 0.715, 0.0715, 0.0715, 0.715, 0.166, 0.166, 0.166, 0.0715)$根据$D_2$可知，分对的样本权重由0.1下降到了0.0715，分错的样本$(6, 7, 8)$的权值由0.1上升至0.166。 此时分类函数为$f_1(x) = 0.4236G_1(x)$，第一个分类器$sign[f_1(x)]$在训练样本上有三个误分类点（第一轮的误分类点即第一个基分类器的误分类点）。

2.第二次迭代（m=2）：

- 1）在上一轮迭代中，我们获知了新一轮的权重分布D_2，在此基础上，经过计算可得，阈值v是8.5时分类误差率最低，因此第二个基本分类器为

$$G_2(x) = \begin{cases} 1 & x < 8.5 \\ -1 & x > 8.5 \end{cases}$$

- 2）误差率：

$$e_2 = P(G_2(x_i) \neq y_i) = 0.0715 \times 3 = 0.2143$$

- 3）$G_2(x)$的系数为:

$$a_2 = \frac{1}{2}\log\frac{1-e_2}{e_2} = 0.6496$$

- 4）更新训练样本的权值分布，得到
$D_3 = (0.0455, 0.0455, 0.0455, 0.1667\ ,\ \ 0.1667, 0.1667, 0.1060, 0.1060, 0.1060, 0.0455)$，相较于$D_2$，被分错的样本3，4，5的权值变大，其他被分对的样本的权值变小。经过第二轮迭代后，分类函数为$f_2(x) = 0.4236G_1(x) + 0.6496G_2(x)$，第二个分类器为$sign[f_2(x)] = sign[0.4236G_1(x) + 0.6496G_2(x)]$。将10个样本点依次带入到第二个分类器中，可得到此时依然有着3个误分类点$(3, 4, 5)$，为此需要进行下一轮迭代。

3.第三次迭代（m=3）：

- 1）在上一轮迭代中，我们获知了新一轮的权重分布D_3，在此基础上，经过计算可得，阈值v

是5.5时分类误差率最低，因此第三个基本分类器为

$$G_3(x) = \begin{cases} 1 & x > 5.5 \\ -1 & x < 5.5 \end{cases}$$

- 2）误差率:

$$e_3 = P(G_3(x_i) \neq y_i) = 0.0455 \times 4 = 0.1820$$

- 3）$G_3(x)$的系数为

$$a_3 = \frac{1}{2}\log\frac{1-e_3}{e_3} = 0.7514$$

- 4）更新训练样本的权值分布，得到
  $D_4 = （0.125，0.125，0.125，0.102，0.102，0.102，0.065，0.065，0.065，0.125）$
  此时分类函数为$f_3(x) = 0.4236G_1(x) + 0.6496G_2(x) + 0.7514G_3(x)$，第三个分类器为$sign[f_3(x)] = sign[0.4236G_1(x) + 0.6496G_2(x) + 0.7514G_3(x)]$，同样将10个样本点依次代入第三个分类器中，可发现没有误分类点，全部样本点已正确分类，因此停止迭代。算法运行完毕。

从上述过程可以发现，如果某些样本被分错，那么它们在下一轮迭代中的权重将会被增大，同时，其它被分错的样本在下一轮迭代中的权值将会被减小。就这样，分错样本权值增大，分对样本权值变小。而每一轮的迭代中，总是选取让误差率最低的阈值来设计基本分类器，因此误差率e(所有被$G_m(x)$误分类样本的权值之和)在迭代中将不断降低。

三、AdaBoost算法的训练误差分析

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

3.1 AdaBoost的训练误差界

首先，给出AdaBoost的训练误差界的定理：

$$\frac{1}{N}\sum_{i=1}^{N} I\left(G\left(x_i \neq y_i\right)\right) \leq \frac{1}{N}\sum_{i} \exp\left(-y_i f\left(x_i\right)\right) = \prod_{m} Z_m$$

其中

$$G(x) = sign(f(x)) = sign\left(\sum_{m=1}^{M} a_m G_m(x)\right)$$

$$Z_m = \sum_{i=1}^{N} w_{mi}\exp\left(-a_m y_i G_m\left(x_i\right)\right)$$

$$f(x) = \sum_{m=1}^{M} a_m G_m(x)$$

证明如下

- 1）当$G(x_i) \neq y_i$时，$y_i f(x_i) < 0$，因而$exp(-y_i f(x_i)) \geq 1$。由此直接推导出前半部分。
- 2）后半部分的推导需要用到

$$w_{m+1,i} = \frac{w_{mi}}{Z_m}\exp\left(-\alpha_m y_i G_m\left(x_i\right)\right), i = 1, 2, \cdots, N$$

$$w_{mi}\exp\left(-\alpha_m y_i G_m\left(x_i\right)\right) = Z_m w_{m+1,i}$$

推导如下：

$$\frac{1}{N}\sum_i \exp\left(-y_i f\left(x_i\right)\right)$$

$$= \frac{1}{N}\sum_i \exp\left(-\sum_{m=1}^{M} a_m y_i G_m\left(x_i\right)\right)$$

$$= \sum_i w_{1i}\prod_{m=1}^{M}\exp\left(-a_m y_i G_m\left(x_i\right)\right)$$

$$= Z_1\sum_i w_{2i}\prod_{m=2}^{M}\exp\left(-a_m y_i G_m\left(x_i\right)\right)$$

$$= Z_1 Z_2\sum_i w_{3i}\prod_{m=3}^{M}\exp\left(-a_m y_i G_m\left(x_i\right)\right)$$

$$= Z_1 Z_2 \cdots Z_{M-1}\sum_i w_{Mi}\exp\left(-a_M y_i G_m\left(x_i\right)\right)$$

$$= \prod_{m=1}^{M} Z_m$$

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

3.2 二类分类问题AdaBoost的训练误差界

对于二分类问题，有如下结果：

$$\prod_{m=1}^{M} Z_m = \prod_{m=1}^{M} \left[2\sqrt{e_m\left(1-e_m\right)} \right] = \prod_{m=1}^{M} \sqrt{\left(1-4\gamma_m^2\right)} \leq \exp\left(-2\sum_{m=1}^{M} \gamma_m^2 \right)$$

这里

$$\gamma_m = \frac{1}{2} - e_m$$

证明：

- 1）首先，等式部分

$$Z_m = \sum_{i=1}^{N} w_{mi} \exp\left(-a_m y_i G_m\left(x_i\right) \right)$$

$$= \sum_{y_i = G_m\left(x_i\right)} w_{mi} \exp\left(-a_m\right) + \sum_{y_i \neq G_m\left(x_i\right)} w_{mi} \exp\left(a_m\right)$$

$$= \left(1-e_m\right) e^{-a_m} + e_m e^{a_m}$$

$$= 2\sqrt{e_m\left(1-e_m\right)}$$

$$= \sqrt{1-4\gamma_m^2}$$

- 2）不等式部分，先由e^x和$\sqrt{(1-x)}$在$x=0$处的泰勒展开式

$$e^x = 1 + x + \frac{1}{2}x^2 + \cdots + \frac{1}{n!}x^n + O\left(x^n\right)$$

$$(1-x)^{\frac{1}{2}} = 1 - \frac{1}{2}x - \frac{1}{8}x^2 + \cdots + O\left(x^n\right)$$

推出不等式

$$\sqrt{\left(1-4\gamma_m^2\right)} \leq \exp\left(-2\gamma_m^2\right)$$

进而得到。

3.3 推论

由上述两个定理推出，如果存在$\gamma > 0$，对所有m有$\gamma_m \geqslant \gamma$，则

$$\frac{1}{N}\sum_{i=1}^{N} I\left(G\left(x_i\right) \neq y_i\right) \leq \exp\left(-2M\gamma^2\right)$$

这个结论表明在此条件下Adaboost的训练误差是以指数速率下降的。

四、AdaBoost算法的数学推导

AdaBoost算法还有另一个解释，AdaBoost算法可以被认为模型是加法模型，损失函数为指数函数、学习算法为前向分步算法的二类分类学习方法。

4.1 前向分布算法

在推导之前，先敲定几个概念：

- **加法模型（additive model）：**

$$f(x) = \sum_{m=1}^{M} \beta_m b\left(x;\gamma_m\right)$$

  。其中$b\left(x;\gamma_m\right)$为基函数，β_m为基函数的系数。
- **损失函数极小化：**在给定训练数据及损失函数$L(y,f(x))$的条件下，学习加法模型$f(x)$成为经验风险极小化即损失函数极小化

$$\min_{\beta_m,\gamma_m} \sum_{i=1}^{N} L\left(y_i, \sum_{m=1}^{M} \beta_m b\left(x_i;\gamma_m\right)\right)$$

- **前向分布算法（forward stagewise algorithm）：**该算法的基本思路为：由于学习的是加法模型，如果可以从前往后，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（即损失函数极小化表达式），那么就可以简化优化的复杂度。具体地，每一步只需要优化

如下损失函数：

$$\min_{\beta,\gamma}\sum_{i=1}^{N}L\left(y_i,\sum_{m=1}^{M}\beta b(x;\gamma)\right)$$

4.2 基于前向分布算法的AdaBoost推导

前向分布算法学习的是加法模型，当基函数为基本分类器时，该加法模型等价于AdaBoost的最终分类器$f(x)=\sum_{m=1}^{M}a_mG_m(x)$，它由基本分类器$G_m(x)$及其系数$a_m$组成。
前向分步算法逐一学习基函数，这一过程与AdaBoost算法逐一学习基本分类器的过程一致。
下面证明前向分步算法的损失函数是指数函数$L(y,f(x))=\exp[-yf(x)]$时，其学习的具体操作等价于AdaBoost算法学习的具体操作。

假设经过$m-1$轮迭代前向分步算法已经得到$f_{m-1}(x)$：

$$f_{m-1}(x)=f_{m-2}(x)+a_{m-1}G_{m-1}(x)$$

$$=a_1G_1(x)+a_2G_2(x)+\cdots+a_{m-1}G_{m-1}(x)$$

在m轮迭代得到a_m，$G_m(x)$和$f_m(x)$，表示为

$$f_m(x)=f_{m-1}(x)+a_mG_m(x)$$

此时参数a_m和$G_m(x)$均未知。因此，我们的目标是要得到最小化损失函数，通过最小化损失函数来得到模型中所需要的参数。而在AdaBoost算法中，每一轮都需要更新样本的权值参数，故而在每一轮的迭代中需要加工损失函数极小化，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，最终使得$f_m(x)$在训练样本上的指数损失最小。
极小化损失函数为：

$$\left(a_m,G_m(x)\right)=arg\min_{a,G}\sum_{i=1}^{N}\exp\left[-y_i\left(f_{m-1}\left(x_i\right)+aG\left(x_i\right)\right)\right]$$

我们先假定$G_1(x),G_2(x),\cdots,G_{m-1}(x)$和$a_1,a_2,\cdots,a_{m-1}$已知，求解$G_m(x)$和$a_m$。
可以将上式表示为

$$\left(a_m,G_m(x)\right)=arg\min_{a,G}\sum_{i=1}^{N}\bar{w}_{mi}\exp\left[-y_iaG\left(x_i\right)\right]$$

其中

$$\bar{w}_{mi} = \exp\left[- y_i f_{m-1}\left(x_i\right)\right]$$

因为$\bar{w}_{mi}$既不依赖a也不依赖于G，所以与最小化无关。但它依赖于$f_{m-1}(x)$，随着每一次迭代而发生改变。

现证使上式达到最小的a_m^*和$G_m^*(x)$就是AdaBoost算法所得到的a_m和$G_m(x)$。求解可分为两步

- 1）首先，求$G_m^*(x)$。对任意的$a > 0$，使指数损失函数最小的$G(x)$由下式得到：

$$G_m^*(x) = arg \min_G \sum_{i=1}^{N} \bar{w}_{mi} I\left(y_i \neq G\left(x_i\right)\right)$$

  此分类器$G_m^*(x)$即为AdaBoost算法的基本分类器$G_m(x)$，因为它是使第m轮加权训练数据分类误差率最小的基本分类器。

- 2）然后，求a_m^*

$$\sum_{i=1}^{N} \bar{w}_{mi} \exp\left[- y_i a G\left(x_i\right)\right] = \sum_{y_i = G_m\left(x_i\right)} \bar{w}_{mi} e^{-a} + \sum_{y_i \neq G_m\left(x_i\right)} \bar{w}_{mi} e^{a}$$

$$= e^{a} \sum_{y_i \neq G_m\left(x_i\right)} \bar{w}_{mi} - e^{-a} \sum_{y_i \neq G_m\left(x_i\right)} \bar{w}_{mi} + e^{-a} \sum_{y_i \neq G_m\left(x_i\right)} \bar{w}_{mi} + e^{-a} \sum_{y_i = G_m\left(x_i\right)} \bar{w}_{mi}$$

$$= \left(e^{a} - e^{-a}\right) \sum_{i=1}^{N} \bar{w}_{mi} I\left(y_i \neq G\left(x_i\right)\right) + e^{-a} \sum_{i=1}^{N} \bar{w}_{mi}$$

  将已求得的$G_m^*(x)$代入上式，对a求导并使导数为0，即得到使其损失函数最小的a。设

$$g(a) = \left(e^{a} - e^{-a}\right) \sum_{i=1}^{N} \bar{w}_{mi} I\left(y_i \neq G\left(x_i\right)\right) + e^{-a} \sum_{i=1}^{N} \bar{w}_{mi}$$

  求导，并令其为0

$$\frac{\partial g(a)}{\partial a} = \left(e^{a} + e^{-a}\right) \sum_{i=1}^{N} \bar{w}_{mi} I\left(y_i \neq G\left(x_i\right)\right) - e^{-a} \sum_{i=1}^{N} \bar{w}_{mi} = 0$$

  得到

$$\left(e^{a} + e^{-a}\right) \sum_{i=1}^{N} \bar{w}_{mi} I\left(y_i \neq G\left(x_i\right)\right) = e^{-a} \sum_{i=1}^{N} \bar{w}_{mi}$$

$$\left(e^{2a}+1\right)\sum_{i=1}^{N}\bar{w}_{mi}I\left(y_i\neq G\left(x_i\right)\right)=\sum_{i=1}^{N}\bar{w}_{mi}$$

$$\left(e^{2a}+1\right)\frac{\sum_{i=1}^{N}\bar{w}_{mi}I\left(y_i\neq G\left(x_i\right)\right)}{\sum_{i=1}^{N}\bar{w}_{mi}}=1$$

令

$$e_m=\frac{\sum_{i=1}^{N}\bar{w}_{mi}I\left(y_i\neq G\left(x_i\right)\right)}{\sum_{i=1}^{N}\bar{w}_{mi}}$$

e_m是分类误差率，得到

$$e_m\left(e^{2a}+1\right)=1$$

最终得到使损失函数最小的a

$$a_m^*=\frac{1}{2}\log\frac{1-e_m}{e_m}$$

这就是之前我们的权重系数a_m的来源。
最后来看一下每一轮样本权值的更新。由

$$f_m(x)=f_{m-1}(x)+a_mG_m(x)$$

以及

$$\bar{w}_{mi}=\exp\left[-y_if_{m-1}(x)\right]$$

可得

$$\bar{w}_{m+1,i}=\exp\left[-y_if_m(x)\right]=\exp\left[-y_i\left(f_{m-1}(x)+a_mG_m(x)\right)\right]$$

$$=\bar{w}_{mi}\exp\left[-y_ia_mG_m(x)\right]$$

从这一步中我们可以看到，这与开篇中所提到的AdaBoost的算法流程中的权重系数($w_{m+1,i}$)仅相差一个规范化因子Z_m，因而是等价的。