Adaboost

1. Boosting 思想

先从初始训练集训练出一个基学习器，再根据基学习器的表现对训练样本分布进行调整，使得先前基学习器做错的训练样本在后续受到更多关注，然后基于调整后的样本分布来训练下一个基学习器。如此重复进行，直至基学习器数目达到事先指定的值 T， 最终将这 T 个基学习器进行加权结合。

- **对提升方法来说，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概率分布?
    2. 如何将弱分类器组合成一个强分类器?
- **$Boosting$要求基学习器能对特定的数据分布进行学习，一般两种方法（这两种没有显著优劣差别）：**
    1. 重赋权法（$re-weighting$）：适用于可以接受带权样本的基学习器（损失函数对样本加权计算）；
    2. 重采样法（$re-sampling$）：适用于无法接受带权样本的基学习器（抽样时不同样本抽中的概率不同）；

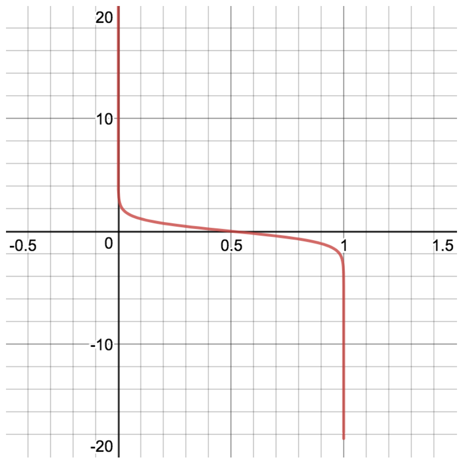
2. AdaBoost 算法

- **针对$Boosting$的两个问题，$AdaBoost$的解决思路：**
    1. 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，分类问题被一系列的弱分类器“分而治之”。
    2. 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，减小分类误差率大的弱分类器的权值，使其在表决中起较小的作用。
- **具体算法：**

> **输入：** 1. $T=(x_1,y_1),...,(x_N,y_N), x_i \in x \subseteq R^n, y_i \in y \subseteq \{-1,+1\}$ 2. 基学习器
>
> **输出：** 最终分类器$G(x)$
>
> **过程：**
>
> 1. 初始化训练数据的权值分布
> $D_1=(w_{11},w_{12},w_{13},...,w_{1N}),\ \ w_{1i}=\frac{1}{N},\ \ i=1,2,3,...,N$
> 2. $for\ M=1,2,3,...,m\ \ Do$
> $a)$ 使用具有权值分布D_m的训练数据集学习，得到基学习器$G_m(x): X->\{-1,+1\}$
>
> $b)$ 计算基学习器G_m在训练数据集上的分类误差率。注意：权重作为计算每个样本对于总体分类误差率的权重。
>
> $e_m=P(G_m(x_i)\neq y_i)=\sum_{i=1}^{N}w_{mi}I(G_m(x_i)\neq y_i)$
>
> $c)$ 计算$G_m(x)$的系数，即基学习器组合时的权重，准确率高的基学习器加法模型中权重更大。

> $\alpha_m = \frac{1}{2}log\frac{1-e_m}{e_m}$ (1)
>
> α_m的推导有两种：i) 最小化训练误差界Z_m进行推导；ii) 最小化损失函数进行推导，本质上一样。
>
> $d)$ 更新训练数据集的权值分布
>
> $D_{m+1} = (w_{m+1,1}, w_{m+1,2}, w_{m+1,3}, \ldots, w_{m+1,N})$， $w_{1i} = \frac{1}{N}$， $i = 1, 2, \ldots, N$
>
> $w_{m+1,i} = \frac{w_{m,i}}{Z_m}exp(-\alpha_m y_i G_m(x_i))$， $i = 1, 2, \ldots, N$ (2)
>
> 即 $\omega_{m+1,i} = \begin{cases} \frac{\omega_{m,i}}{Z_m}e^{-\alpha_m} & if\ y_i = G_m(x_i) \\ \frac{\omega_{m,i}}{Z_m}e^{\alpha_m} & if\ y_i \neq G_m(x_i) \end{cases}$
>
> $Z_m = \sum_{i=1}^{m} w_{m,i}exp(-\alpha_m y_i G_m(x_i))$, 规范化因子使D_{m+1}成为一个概率分布
>
> 3. 构建基本分类器的线性组合
>
> $f(x) = \sum_{m=1}^{M} \alpha_m G_m(x)$
>
> $G(x) = sign(f(x)) = sign(\sum_{m=1}^{M} \alpha_m G_m(x))$

- **对算法的几点说明：**
  - $2.b$：$Gm(x)$ 在加权的训练数据集上的分类误差率是$Gm(x)$ 误分类样本的权值之和，由此可以看出数据权值分布Dm与基本分类器$Gm(x)$的分类误差率的关系。误分类样本权值和越大，分类误差率越大。
  - $2.c$：由系数计算公式可知，当$e_m \leq \frac{1}{2}$ 时，$\alpha_m \geq 0$ ，并且a_m随着e_m的减小而增大，所以分类误差率越小的基本分类器在最终分类器中的作用越大。

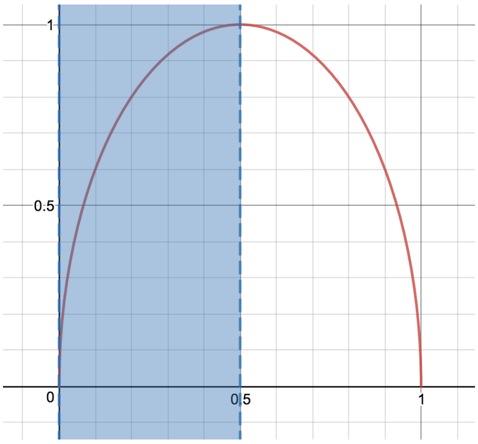


  - $2.d$ 由式可知，被基本分类器$G_m(x)$误分类样本的权值得以扩大，而被正确分类样本的权值却得以缩小。两相比较，误分类样本的权值被放大$e^{2\alpha_m} = \frac{e_m}{1-e_m}$倍。因此，误分类样本在下一轮学习中起更大的作用。
  - **不改变所给的训练数据**，而不断改变训练数据**权值的分布**，使得训练数据在基本分类器的学习中起不同的作用，**这是AdaBoost的一个特点。** w_{mi} **影响的是分类误差率**e_m**。**
  - 3 系数a_m表示了基本分类器$G_m(x)$的重要性，**这里，所有**a_m**之和并不为1。**$f(x)$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。**

3. AdaBoost 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少训练误差，即在训练数据集上的分类误差率。

1. 定理1

> **定理：**
>
> $AdaBoost$ 算法最终分类器的训练误差界为
>
> $\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leq \frac{1}{N}\sum_i exp(-y_i f(x_i)) = \prod_m Z_m$ (3)
>
> 其中：
>
> $G(x) = sign(f(x)) = sign(\sum_{m=1}^{M} \alpha_m G_m(x))$
>
> $f(x) = \sum_{m=1}^{M} \alpha_m G_m(x)$
>
> $Z_m = \sum_{i=1}^{m} w_{m,i} exp(-\alpha_m y_i G_m(x_i))$
>
> **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。不断增加弱分类器，训练误差的上界会不断下降。**
>
> 因为Z_m取值范围$[0,1]$，所以迭代次数越多，最终分类器训练误差界越小。
>
> 
>

2. 定理2

> **定理：**二分类问题$AdaBoost$ 的训练误差界
>
> $\prod_{m=1}^{M} Z_m = \prod_{m=1}^{M}[2\sqrt{e_m(1-e_m)}] = \prod_{m=1}^{M}\sqrt{1-4\gamma_m^2} \leq exp(-2\sum_{m=1}^{M}\gamma_m^2)$ (4)
>
> 其中：$\gamma_m = \frac{1}{2} - e_m$

3. 推论

> 如果存在$\gamma > 0$，对所有m 有$\gamma_m \geq \gamma$，则
>
> $\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leq exp(-2M\gamma^2)$ (5)
>
> **这表明在此条件下AdaBoost的训练误差是以指数速率下降的。这一性质当然是很有吸引力的。**

相关证明见《统计学习方法》和https://www.jianshu.com/p/bfba5a91ba15。**后面补上自己的证明过程。**

4. AdaBoost 推导

运用加法模型与前向分布算法，可以推导出$AdaBoost$ 算法（确定参数α_k，$\omega_{k+1,i}$）。

4.1 加法模型与前向分步算法

> **加法模型：**
>
> $f(x_i)=\sum_{m=1}^{M}\beta_m b(x;\gamma_m)$
>
> **给定数据和损失函数，训练加法模型即求解经验风险最小化（损失函数最小化）问题：**
>
> $$\min_{\beta_m,\gamma_m}=\sum_{i=1}^{N}L(y_i,\sum_{m=1}^{M}\beta_m b(x_i;\gamma_m))\ (6)$$
>
> 即同时求解从$m=1$到M的β_m,γ_m的负责优化问题。

> **前向分步算法**（$forwardstagewisealgorithm$）
>
> 求解这一优化问题的思想：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(6)，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：
>
> $$\min_{\beta,\gamma}\sum_{i=1}^{M}L(y_i,\beta b(x_i;\gamma))\ (7)$$
>
> ---
>
> **输入：**数据集T，损失函数$L(y,f(x))$，基函数集$\{b(x;\gamma)\}$
>
> **输出：**加法模型$f(x)$
>
> (1) 初始化；
>
> (2) 对m=1,2,...,M
>
> $a)$ 极小化损失函数：$(\beta_m,\gamma_m)=\arg\min_{\beta,\gamma}\sum_{i=1}^{N}L(y_i,f_{m-1}(x_i)+\beta b(x_i;\gamma))=>(\beta_m,\gamma_m)$
>
> $b)$ 更新。$f_m(x)=f_{m-1}(x)+\beta_m b(x;\gamma_m)$
>
> (3) 得到加法模型：$f(x)=f_M(x)=\sum_{m=1}^{M}\beta_m b(x;\gamma_m)$
>
> **这样，前向分步算法将同时求解从$m=1$到M所有参数β_m,γ_m的优化问题简化为逐次求解各个β_m,γ_m的优化问题。**

4.2 推导 AdaBoost

$AdaBoost$算法是前向分步加法算法的特例。这时，模型是由基本分类器组成的加法模型，**损失函数是指数函数。**

> **推导：**
>
> - $Adaboost$是若干个弱分类器加权和而得到最终的强分类器，因此是一个加法模型。
> - 损失函数定义为指数损失函数：$L(y,f(x))=exp(-yf(x))$
> - 确定3个要素：$G_m(x)$，基分类器权重系数α_m，样本权重如何更新ω_{m+1}，
>
> ---
>
> 假设经m轮迭代，已经得到$f_{m-1}(x)$:
> $f_{m-1}(x)=f_{m-2}(x)+\alpha_{m-1}G_{m-1}(x)=\alpha_1G_1(x)+...+\alpha_{m-1}G_{m-1}(x)$
>
> 在第m轮迭代得到$\alpha_m,G_m(x),f_m(x)$：$f_m(x)=f_{m-1}(x)+\alpha_mG_m(x)$

目标： 使用前向分布算法确定α_m和$G_m(x)$，使$f_m(x)$在训练数据集T上的**指数损失**最小，即

$(\alpha_m, G_m(x)) = \arg\min\limits_{\alpha,G} \sum\limits_{i=1}^{N} exp[-y_i(f_{m-1}(x_i) + \alpha_m G_m(x_i))]$ (8)

令$\tilde{\omega}_{m,i} = exp(-y_i f_{m-1}(x_i))$，即求解如下**损失函数**：

$(\alpha_m, G_m(x)) = \arg\min\limits_{\alpha,G} \sum\limits_{i=1}^{N} \tilde{\omega}_{m,i} exp[-y_i \alpha G(x_i)]$ (9)

这里，$\tilde{\omega}_{m,i}$因为既不依赖α也不依赖于$G(x)$，所以与最小化无关。但$\tilde{\omega}_{m,i}$依赖于$f_{m-1}(x)$，随着每一轮迭代而发生改变。

1) 先求$G_k^*(x)$

把α看成常数，损失函数可理解为求基分类器$G_m(x)$使得加权训练样本分类误差尽可能少。等价于

$G_m^*(x) = \arg\min\limits_{G} \sum\limits_{i=1}^{N} \tilde{\omega}_{m,i} I(y_i \neq G(x_i))$

2) 再求α_k^*

化简损失函数$\sum\limits_{i=1}^{N} \tilde{\omega}_{m,i} exp[-y_i \alpha G(x_i)]$

$$
\begin{aligned}
\sum_{i=1}^{N} \tilde{\omega}_{m,i} exp[-y_i \alpha G(x_i)] &= \sum_{y_i = G_m(x_i)}^{N} \tilde{\omega}_{m,i} e^{-\alpha} + \sum_{y_i \neq G_m(x_i)}^{N} \tilde{\omega}_{m,i} e^{\alpha} \\
&= \sum_{y_i = G_m(x_i)}^{N} \tilde{\omega}_{m,i} e^{-\alpha} + \sum_{i=1}^{N} \tilde{\omega}_{m,i} e^{\alpha} I(y_i \neq G(x_i)) \\
&= \sum_{i=1}^{N} \tilde{\omega}_{m,i} e^{-\alpha} - \sum_{i=1}^{N} \tilde{\omega}_{m,i} e^{-\alpha} I(y_i \neq G(x_i)) + \sum_{i=1}^{N} \tilde{\omega}_{m,i} e^{\alpha} I(y_i \neq G(x_i)) \\
&= \sum_{i=1}^{N} \tilde{\omega}_{m,i} e^{-\alpha} + (e^{\alpha} - e^{-\alpha}) \sum_{i=1}^{N} \tilde{\omega}_{m,i} I(y_i \neq G(x_i))
\end{aligned}
$$

求导取0可得：

$$
\begin{aligned}
-e^{-\alpha} \sum_{i=1}^{N} \tilde{\omega}_{m,i} + (e^{\alpha} + e^{-\alpha}) \sum_{i=1}^{N} \tilde{\omega}_{m,i} I(y_i \neq G(x_i)) = 0 \\
(e^{\alpha} + e^{-\alpha}) \frac{\sum\limits_{i=1}^{N} \tilde{\omega}_{m,i} I(y_i \neq G(x_i))}{\sum\limits_{i=1}^{N} \tilde{\omega}_{m,i}} - e^{-\alpha} = 0 \\
(e^{\alpha} + e^{-\alpha}) e_m - e^{-\alpha} = 0
\end{aligned}
$$

求得：

$\alpha_m = \frac{1}{2} ln(\frac{1-e_m}{e_m})$

3) 最后看样本权重更新

利用$f_m(x) = f_{m-1}(x) + \alpha_m G_m(x)$, $\tilde{\omega}_{m,i} = exp(-y_i f_{m-1}(x_i))$

> $$\begin{aligned}\tilde{\omega}_{m+1,i} &= exp(-y_i f_m(x_i)) \\ &= exp(-y_i(f_{m-1}(x_i)+\alpha_m G_m(x_i))) \\ &= e^{-y_i f_{m-1}(x_i)} e^{-y_i \alpha_m G_m(x_i)} \\ &= \tilde{\omega}_m e^{-y_i \alpha_m G_m(x_i)}\end{aligned}$$
>
> 推导完成。

5. 实例

6. 后续问题

6.1 $adaboost$ 不易过拟合问题

6.2 如何理解基分类器准确率不低于0.5

参考文献：

1. 《统计学习方法》李航
2. 《机器学习》周志华
3. http://kubicode.me/2016/04/18/Machine%20Learning/AdaBoost-Study-Summary/
4. https://zhuanlan.zhihu.com/p/38507561 (算法的一些细节)
5. http://www.csuldw.com/2016/08/28/2016-08-28-adaboost-algorithm-theory/（通过训练误差界推导α_m）
6. http://blog.sina.com.cn/s/blog_6ae183910101chcg.html（一些后续问题思考）