提升方法(boosting)详解

作者博客：@灵魂机器 www.weibo.com/soulmachine

最后更新日期：2012-12-11

提升方法（boosting）是一种常用的统计学习方法，应用广泛且有效。在分类问题中，它通过改变训练样本的权重，学习多个分类器，并将这些分类器进行线性组合，提高分类的性能。

本章首先介绍提升方法的思路和代表性的提升算法 AdaBoost，然后通过训练误差分析探讨 AdaBoost 为什么能够提高学习精度，并且从前向分布加法模型的角度解释 AdaBoost，最后叙述提升方法更具体的事例——提升术（boosting tree）。AdaBoost 算法是 1995 年由 Freund 和 Schapire 提出的，提升树是 2000 年由 Friedman 等人提出的。

1 Adaboost 算法基本原理

1.1 提升方法的基本思路

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”。

Leslie Valiant 首先提出了“强可学习（strongly learnable）”和“弱可学习（weakly learnable）”的概念，并且指出：在概率近似正确（probably approximately correct, PAC）学习的框架中，一个概念（一个类），如果存在一个多项式的学习算法能够学习它，并且正确率很高，那么就称这个概念是强可学习的，如果正确率不高，仅仅比随即猜测略好，那么就称这个概念是弱可学习的。2010 年的图灵奖给了 L. Valiant，以表彰他的 PAC 理论。非常有趣的是 Schapire 后来证明强可学习与弱可学习是等价的，也就是说，在 PAC 学习的框架下，一个概念是强可学习的充要条件是这个概念是可学习的。

这样一来，问题便成为，在学习中，如果已经发现了“弱学习算法”，那么能否将它提升（boost）为“强学习算法”。大家知道，发现弱学习算法通常比发现强学习算法容易得多。那么如何具体实施提升，便成为开发提升方法时所要解决的问题。关于提升方法的研究很多，有很多算法被提出。最具代表性的是 AdaBoost 算法（Adaptive Boosting Algorithm），可以说，AdaBoost 实现了 PAC 的理想。

对于分类问题而言，给定一个训练数据，求一个比较粗糙的分类器（即弱分类器）要比求一个精确的分类器（即强分类器）容易得多。提升方法就是从弱学习算法出发，反复学习，得到一系列弱分类器，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（训练数据中的各个数据点的权值分布），调用弱学习算法得到一个弱分类器，再改变训练数据的概率分布，再调用弱学习算法得到一个弱分类器，如此反复，得到一系列弱分类器。

这样，对于提升方法来说，有两个问题需要回答：一是在每一轮如何如何改变训练数据的概率分布；而是如何将多个弱分类器组合成一个强分类器。

关于第一个问题，AdaBoost 的做法是，提高那些被前几轮弱分类器线性组成的分类器错误分

类的的样本的权值。这样一来，那些没有得到正确分类的数据，由于权值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”。至于第二个问题，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。

AdaBoost 的巧妙之处就在于它将这些想法自然而然且有效地实现在一种算法里。

1.2 AdaBoost 算法

输入：训练数据集 $T=\{(x_1,y_1),(x_2,y_2),...,(x_N,y_N)\}$，其中 $x_i \in X \subseteq R^n$，表示输入数据，$y_i \in Y=\{-1,+1\}$，表示类别标签；弱学习算法。

输出：最终分类器 G(x)。

流程：

(1) 初始化训练数据的概率分布，刚开始为均匀分布

$D_1=(w_{11},w_{12},...,w_{1N})$, 其中 $w_{1i}=\frac{1}{N}$, i=1,2,..,N

D_m 表示在第 m 轮迭代开始前，训练数据的概率分布（或权值分布），w_{mi} 表示在第 i 个样本的权值，$\sum_{i=1}^{N} w_{mi}=1$。

(2) 对 m=1,2,…,M，

(a) 使用具有权值分布 D_m 的训练数据集进行学习（任意选一种模型都可以，例如朴素贝叶斯，决策树，SVM 等，并且每一轮迭代都可以用不同的模型），得到一个弱分类器

$$G_m(x)=X \rightarrow \{-1,+1\}$$

(b) 计算 $G_m(x)$在训练数据集上的分类误差率

$$e_m=P(G_m(x_i)\neq y_i)=\sum_{i=1}^{N} w_{mi} I(G_m(x_i)\neq y_i) \quad \text{(公式 1)}$$

(c) 计算弱分类器 $G_m(x)$的系数

$$\alpha_m=\frac{1}{2}\log\frac{1-e_m}{e_m} \quad \text{(公式 2)}$$

(d) 更新训练数据的权值分布

$$D_{m+1}=(w_{m+1,1},w_{m+1,1},...,w_{m+1,N}) \quad \text{(公式 3)}$$

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} \exp(-\alpha_m y_i G_m(x)), i = 1, 2, ..., N \quad \text{(公式 4)}$$

这里，Z_m 是规范化因子

$$Z_m = \sum_{i=1}^{N} w_{mi} \exp(-\alpha_m y_i G_m(x)) \quad \text{(公式 5)}$$

这样 $\sum_{i=1}^{N} w_{m+1,i} = 1$，它使 D_{m+1} 称为一个概率分布。

(3) 将 M 个基本分类器进行线性组合

$$f(x) = \sum_{i=1}^{M} \alpha_m G_m(x) \quad \text{(公式 6)}$$

得到最终分类器

$$G(x) = sign(f(x)) = sign\left(\sum_{i=1}^{M} \alpha_m G_m(x)\right) \quad \text{(公式 7)}$$

对 AdaBoost 算法作如下说明：

步骤(1) 初始时假设训练数据集具有均匀分布，即每个训练样本在弱分类器的学习中作用相同。

步骤(2) (c) α_m 表示 $G_m(x)$在最终分类器中的重要性。由式(公式 2)可知，当 $e_m \leq 1/2$ 时，$\alpha_m \geq 0$，并且 α_m 随着 e_m 的减小而增大，即意味着误差率越小的基本分类器在最终分类器中的作用越大。

(d) 式(公式 4)可以写成：

$$w_{m+1,i} = \begin{cases} \frac{w_{mi}}{Z_m} e^{-\alpha_m}, G_m(x) = y_i \\ \frac{w_{mi}}{Z_m} e^{\alpha_m}, G_m(x) \neq y_i \end{cases}$$

由此可知，被弱分类器 $G_m(x)$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。因此误分类样本在下一轮学习中起到更大的作用。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。

步骤(3) 这里，α_m 之和并不等于 1。f(x)的符号决定实例 x 的类别，f(x)的绝对值表示分类的确信度。利用基本分类器进行线性组合得到最终分类器是 AdaBoost 的另一个特点。

1.3 AdaBoost 的例子

例 1 给定如表 1 所示训练数据。假设弱分类器由 G(x)=sign(x-v)产生，其中 v 为常量，表示阀值。试用 AdaBoost 算法学习一个强分类器。

表 1 训练数据样本

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

解 初始化训练数据的权值分布

$$D_1 = (w_{11}, w_{11}, \cdots, w_{1N})$$
$$w_{1i} = 0.1, i = 1, 2, \cdots, 10$$

当 m=1，进行第一轮迭代

(a) 在权值分布为 D_1 的情况下，用一根垂直扫描线从左到右扫描，会发现，阀值 v 取 2.5 时分类误差率最低，故基本分类器 $G_1(x)=\text{sign}(x-2.5)$。

(b) $G_1(x)$在训练数据集上的误差率 $e_1 = \sum_{i=1}^{10} w_{1i} I(G_1(x_i) \neq y_i) = \sum_{i=1}^{10} \frac{1}{10} I(G_1(x_i) \neq y_i) = 0.3$ ，第 7,8,9 个实例被误分类。

(c) 计算 $G_1(x)$的系数：$\alpha_1 = \frac{1}{2}\log\frac{1-e_1}{e_1}$ =0.4236。

(d) 更新训练数据的权值分布：

$$D_2 = (w_{21}, w_{22}, \cdots, w_{210})$$
$$w_{2i} = \frac{w_{1i}}{2}\exp(-\alpha_1 y_i G_1(x_i)), i = 1, 2, \cdots, 10$$
$$D_2 = (0.07143, 0.07143, 0.07143, 0.07143, 0.07143, 0.07143,$$
$$0.16667, 0.16667, 0.16667, 0.07143)$$
$$f_1(x) = 0.4236 G_1(x)$$

分类器 $\text{sign}[f_1(x)]$在训练数据集上有 3 个误分类点，因此，继续迭代。

当 m=2，进行第二轮迭代

(a) 在权值分布为 D_2 的情况下，阀值 v 取 8.5 时分类误差率最低，故基本分类器 $G_2(x)=\text{sign}(x-8.5)$。

(b) $G_2(x)$在训练数据集上的误差率 e2=0.07143+0.07143+0.07143+0.07143，第 4,5,6 个实例被错误分类。

(c) 计算 $G_2(x)$的系数：$\alpha_2=0.6496$。

(d) 更新训练数据的权值分布：

$$D_3 = (0.0455, 0.0455, 0.0455, 0.1667, 0.1667, 0.1667, 0.1060, 0.1060, 0.1060, 0.0455)$$
$$f_2(x) = 0.4236G_1(x) + 0.6496G_2(x)$$

分类器 sign[$f_2(x)$]在训练数据集上有 3 个误分类点，因此，继续迭代。

当 m=3，进行第三轮迭代

(a) 在权值分布为 D_3 的情况下，阀值 v 取 5.5 时分类误差率最低，故基本分类器 $G_3(x)$=-sign(x-5.5)，注意，这里符号反向了。

(b) $G_3(x)$在训练数据集上的误差率 e3=0.0455+0.0455+0.0455+0.0455=0.1820，第 1,2,3,10 个实例被误分类。

(c) 计算 $G_3(x)$的系数：$\alpha_2=0.7514$。

(d) 更新训练数据的权值分布：

$$D_4 = (0.125, 0.125, 0.125, 0.102, 0.102, 0.102, 0.065, 0.065, 0.065, 0.125)$$
$$f_3(x) = 0.4236G_1(x) + 0.6496G_2(x) + 0.7514G_3(x)$$

分类器 sign[$f_3(x)$]在训练数据集上的误分类点个数为 0，因此，终止迭代。

于是，最终分类器为

$$G(x) = sign[f_3(x)] = sign[0.4236G_1(x) + 0.6496G_2(x) + 0.7514G_3(x)]$$

注意，$G_1(x)$，$G_2(x)$和 $G_3(x)$，是一个 sign 函数，从图像看是一个方波图，而最终分类器 G(x)也是一个方波图，由三个波形图叠加合成。从信号的角度看，这是振幅叠加。$G_1(x)$，$G_2(x)$和 $G_3(x)$都是弱分类器，分类正确率仅大于 0.5，但线性组合而成的分类器 G(x)正确率是 100%，是一个强分类器。

2 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少训练误差，关于这个问题有下面的两个定理：

定理 1 (AdaBoost 的训练误差界) AdaBoost 算法的最终分类器的训练误差界为

$$\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leq \frac{1}{N}\sum_{i=1}^{N} \exp(-y_i f(x_i)) = \prod_{m=1}^{M} Z_m \qquad \text{(公式 8)}$$

这里，G(x),f(x)和 Zm 分别由(公式 7)、(公式 6)和(公式 5)给出。

证明 当 $G(x_i) \neq y_i$ 时，$I(G(x_i) \neq y_i)=1$，$y_i f(x_i) < 0$，因而 $\exp(-y_i f(x_i)) \geq 1$，所以 $I(G(x_i) \neq y_i) \leq \exp(-y_i f(x_i))$ 成立；

当 $G(x_i) = y_i$ 时，$I(G(x_i) \neq y_i)=0$，又因为 $\exp(-y_i f(x_i)) \geq 0$，所以 $I(G(x_i) \neq y_i) \leq \exp(-y_i f(x_i))$ 成立；

可见，$I(G(x_i) \neq y_i) \leq \exp(-y_i f(x_i))$ 恒成立。

所以 $\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leq \frac{1}{N}\sum_{i=1}^{N} \exp(-y_i f(x_i))$ 成立。

不等式的后半部分推导要用到 Z_m 的定义式(公式 5)及(公式 4)的变形：

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

先推导如下：

$$
\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\exp(-y_i f(x_i)) \\
&=\frac{1}{N}\sum_{i=1}^{N}\exp\left(-\sum_{m=1}^{M}\alpha_m y_i G_m(x_i)\right) \\
&=w_{1i}\sum_{i=1}^{N}\exp\left(-\sum_{m=1}^{M}\alpha_m y_i G_m(x_i)\right) \\
&=\sum_{i=1}^{N}\left(w_{1i}\exp\left(-\sum_{m=1}^{M}\alpha_m y_i G_m(x_i)\right)\right) \\
&=\sum_{i=1}^{N}\left(w_{1i}\prod_{m=1}^{M}\exp(-\alpha_m y_i G_m(x_i))\right) \\
&=Z_1\sum_{i=2}^{N}\left(w_{2i}\prod_{m=1}^{M}\exp(-\alpha_m y_i G_m(x_i))\right) \\
&=Z_1 Z_2\sum_{i=3}^{N}\left(w_{3i}\prod_{m=1}^{M}\exp(-\alpha_m y_i G_m(x_i))\right) \\
&=\cdots \\
&=\prod_{m=1}^{M}Z_m
\end{aligned}
$$

（**定理 1** 的证明@特级飞行员舒克有很大贡献）

这一定理说明，可以在每一轮选取最适当的 G_m 使得 Z_m 最小，从而使训练误差下降最快。对二类分类问题，有如下结果：

定理 2 （二类分类问题 AdaBoost 的训练误差界）

$$
\prod_{m=1}^{M}Z_m=\prod_{m=1}^{M}2\sqrt{e_m(1-e_m)}=\prod_{m=1}^{M}\sqrt{1-4\gamma_m^2}\le\exp\left(-2\sum_{m=1}^{M}\gamma_m^2\right) \qquad \text{(公式 9)}
$$

在这里，$\gamma_m=\frac{1}{2}-e_m$ 。

证明 由 Z_m 的定义式(公式 5)得

$$
\begin{aligned}
&Z_m=\sum_{i=1}^{N}w_{mi}\exp(-\alpha_m y_i G_m(x)) \\
&=\sum_{y_i=G_m(x_i)}w_{mi}e^{-\alpha_m}+\sum_{y_i\neq G_m(x_i)}w_{mi}e^{\alpha_m} \\
&=(1-e_m)e^{-\alpha_m}+e_m e^{\alpha_m},\text{代入}e_m=\frac{1}{2}\log\frac{1-e_m}{e_m} \\
&=2\sqrt{e_m(1-e_m)}=\sqrt{1-4\gamma_m^2}
\end{aligned}
$$

因此等式 $\prod_{m=1}^{M} Z_m = \prod_{m=1}^{M} 2\sqrt{e_m(1-e_m)} = \prod_{m=1}^{M} \sqrt{1-4\gamma_m^2)}$ 成立。

接下来要证明不等式 $\sqrt{1-4\gamma_m^2} \le \exp(-2\gamma_m^2)$，两边平方，即 $1-4\gamma_m^2 \le \exp(-4\gamma_m^2)$。(这个思路由@liyong3forever 贡献，李航书中的那个泰勒展开式的思路并不好)

因为 $\gamma_m = \frac{1}{2} - e_m$，em∈[0,1]，所以 $\gamma_m \in [-\frac{1}{2}, \frac{1}{2}]$，$4\gamma_m^2 \in [0,1]$，令 $x = 4\gamma_m^2 \in [0,1]$，问题变成了证明在闭区间[0,1]上不等式 $1-x \le e^{-x}$ 成立。这里，可以利用函数的单调性，导数等性质来证明，也可以用 MATLAB 画出图像来实际看看。

推论 1 如果存在 γ>0，对所有 m 有 $\gamma_m \ge \gamma$，则

$$\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \ne y_i) \le \exp(-2M\gamma^2) \qquad \text{(公式 10)}$$

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。

注意，AdaBoost 算法不需要知道下界 γ。这正是 Freund 与 Schapire 设计 AdaBoost 时所考虑的。与一些早期的提升方法不同，AdaBoost 具有适应性，即它能适应弱分类器各自的训练误差率。这也是它的名称的由来，Ada 是 Adaptive 的简写。