Adaboost导论

七月算法 邹博

2015年4月11日

提升方法

- 一个概念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，这个概念是强可学习的；
- 一个概念如果存在一个多项式的学习算法能够学习它，并且学习的正确率仅比随机猜测略好，那么，这个概念是弱可学习的；
- 强可学习与弱可学习是等价的。
- 在学习中，如果已经发现了"弱学习算法"，能否将他提升为"强学习算法"。

Adaboost

- 设训练数据集T={(x1,y1), (x2,y2)…(xN,yN)}
- 初始化训练数据的权值分布

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

Adaboost：对于m=1,2,…M

- 使用具有权值分布Dm的训练数据集学习，得到基本分类器

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

- 计算Gm(x)在训练数据集上的分类误差率

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

- 计算Gm(x)的系数

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

Adaboost：对于m=1,2,…M

- 更新训练数据集的权值分布

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

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

- 这里，Zm是规范化因子

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

  - 它的目的仅仅是使D_{m+1}成为一个概率分布

$$w_{m+1,i}=\frac{w_{mi}}{Z_m}\exp(-\alpha_m y_i G_m(x_i))\Rightarrow Z_m w_{m+1,i}=w_{mi}\exp(-\alpha_m y_i G_m(x_i))\Rightarrow Z_1 w_{2,i}=w_{1i}\exp(-\alpha_1 y_i G_1(x_i))$$

Adaboost

- 构建基本分类器的线性组合

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

- 得到最终分类器

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

举例

- 给定下列训练样本，试用AdaBoost算法学习一个强分类器。

序号	1	2	3	4	5	6	7	8	9	X
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

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

- $W_{1i} = 0.1$

m=1

序号	1	2	3	4	5	6	7	8	9	X
X	0	1	2	3	4	5	6	7	8	9
Y	1	1	1	−1	−1	−1	1	1	1	−1

- 对于m=1
- 在权值分布为D1的训练数据上，阈值v取2.5时误差率最低，故基本分类器为：

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

m=1

- G1(x)在训练数据集上的误差率 $e1=P(G1(xi)\neq yi) = 0.3$
- 计算G1的系数：

$$\alpha_1 = \frac{1}{2}\log\frac{1-e_1}{e_1} = 0.4236$$

m=1

- 更新训练数据的权值分布：

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

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

- D_2=(0.0715, 0.0715, 0.0715, 0.0715, 0.0715, 0.0715, 0.1666, 0.1666, 0.1666, 0.0715)
  - 计算D_2，是为下一个基本分类器使用
- $f_1(x)=0.4236*G_1(x)$
- 分类器sign(f_1(x))在训练数据集上有3个误分类点。

m=2

X	0	1	2	3	4	5	6	7	8	9
Y	1	1	1	−1	−1	−1	1	1	1	−1
w	0.0715	0.0715	0.0715	0.0715	0.0715	0.0715	0.1666	0.1666	0.1666	0.0715

- 对于m=2
- 在权值分布为D2的训练数据上，阈值v取8.5时误差率最低，故基本分类器为：

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

m=2

- G2(x)在训练数据集上的误差率
  e2=P(G2(xi)≠yi) = 0.2143(0.0715*3)
- 计算G2的系数：

$$\alpha_2 = \frac{1}{2}\log\frac{1-e_2}{e_2} = 0.6496$$

m=2

- 更新训练数据的权值分布：

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

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

- D3=(0.0455, 0.0455, 0.0455, 0.1667, 0.1667, 0.01667, 0.1060, 0.1060, 0.1060, 0.0455)
- f2(x)=0.4236G1(x) + 0.6496G2(x)
- 分类器sign(f2(x))在训练数据集上有3个误分类点。

m=3

X	0	1	2	3	4	5	6	7	8	9
Y	1	1	1	−1	−1	−1	1	1	1	−1
w	0.0455	0.0455	0.0455	0.1667	0.1667	0.1667	0.1060	0.1060	0.1060	0.0455

- 对于m=3
- 在权值分布为D3的训练数据上，阈值v取5.5时误差率最低，故基本分类器为：

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

m=3

- G3(x)在训练数据集上的误差率 e3=P(G3(xi)≠yi) = 0.1820(0.0455*4)
- 计算G3的系数：

$$\alpha_3 = \frac{1}{2}\log\frac{1-e_3}{e_3} = 0.7514$$

m=3

- 更新训练数据的权值分布：

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

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

- D4=(0.125, 0.125, 0.125, 0.102, 0.102, 0.102, 0.065, 0.065, 0.065, 0.125)
- f3(x)=0.4236G1(x) + 0.6496G2(x)+0.7514G3(x)
- 分类器sign(f3(x))在训练数据集上有0个误分类点。

误差上限

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

- 当G(xi)≠yi时， yi*f(xi)<0， 因而exp(-yi*f(xi))≥1， 前半部分得证。

后半部分

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

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

$$= \sum_i w_{1i}\exp(-\alpha_1 y_i G_1(x_i))\prod_{m=2}^{M}\exp(-\alpha_m y_i G_m(x_i))$$

$$= \sum_i Z_1\prod_{m=2}^{M}\exp(-\alpha_m y_i G_m(x_i)) = Z_1\sum_i\prod_{m=2}^{M}\exp(-\alpha_m y_i G_m(x_i))$$

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

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

$$= Z_1 Z_2 \cdots Z_{M-1}\sum_i w_{Mi}\exp(-\alpha_M y_i G_M(x_i))$$

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

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

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

$$\Rightarrow Z_1 w_{2,i} = w_{1i}\exp(-\alpha_1 y_i G_1(x_i))$$

训练误差界

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

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

训练误差界

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

$$= \sum_{y_i = G_m(x_i)} w_{mi} e^{-\alpha_m} + \sum_{y_i \neq G_m(x_i)} w_{mi} e^{\alpha_m}$$

$$= (1 - e_m) e^{-\alpha_m} + e_m e^{\alpha_m}$$

$$= 2\sqrt{e_m(1 - e_m)} \qquad \alpha_m = \frac{1}{2} \log \frac{1 - e_m}{e_m}$$

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

取γ1，γ2…的最小值，记做γ

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

Adaboost算法解释

- AdaBoost算法是模型为加法模型、损失函数为指数函数、学习算法为前向分步算法时的二类学习方法。

前向分步算法

- 考虑加法模型

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

- 其中：
  - 基函数：$b(x;\gamma_m)$
  - 基函数的参数 γ_m
  - 基函数的系数：β_m

前向分步算法的含义

- 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**问题：

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

- 算法简化：如果能够从前向后，每一步只学习一个基函数及其系数，逐步逼近上式，即：每步只优化损失函数：$\min_{\beta,\gamma}\sum_{i=1}^{N}L(y_i,\beta b(x_i;\gamma))$

前向分步算法的算法框架

- 输入：
  - 训练数据集T={(x1,y1), (x2,y2)…(xN,yN)}
  - 损失函数L(y,f(x))
  - 基函数集{b(x; γ)}
- 输出：
  - 加法模型f(x)
- 算法步骤：

前向分步算法的算法框架

- 初始化$f_0(x) = 0$
- 对于m=1,2,..M
  - 极小化损失函数 $(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma))$
    - 得到参数 β_m γ_m
  - 更新当前模型：

$$f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$$

- 得到加法模型 $f(x) = f_M(x) = \sum_{m=1}^{M} \beta_m b(x; \gamma_m)$

前向分步算法与AdaBoost

- AdaBoost算法是前向分步算法的特例，这时，模型是基本分类器组成的**加法模型**，损失函数是**指数函数**。
- 损失函数取：

$$L(y, f(x)) = \exp(-yf(x))$$

证明

- 假设经过m-1轮迭代，前向分步算法已经得到$f_{m-1}(x)$：$$f_{m-1}(x)=f_{m-2}(x)+\alpha_{m-1}G_{m-1}(x)$$ $$=\alpha_1 G_1(x)+\cdots+\alpha_{m-1}G_{m-1}(x)$$
- 在第m轮迭代得到α_m，$G_m(x)$和$f_m(x)$
- 目标是使前向分步算法得到的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

$$(\alpha_m, G_m(x)) = \arg\min_{\alpha, G} \sum_{i=1}^{N} \exp(-y_i(f_{m-1}(x_i)+\alpha G(x_i)))$$

证明

- 进一步：

$$\left(\alpha_m, G_m(x)\right) = \arg\min_{\alpha, G} \sum_{i=1}^{N} \overline{w}_{mi} \exp\left(-y_i \alpha G(x_i)\right)$$

- 其中： $\overline{w}_{mi} = \exp\left(-y_i f_{m-1}(x_i)\right)$
- $\overline{w}_{mi}$ 既不依赖α也不依赖G，所以与最小化无关。但 $\overline{w}_{mi}$ 依赖于 $f_{m-1}(x)$，所以，每轮迭代会发生变化。

基本分类器G*(x)

- 首先求分类器G*(x)
- 对于任意α>0，是上式最小的G(x)由下式得到：

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

- 其中，$\overline{w}_{mi} = \exp(-y_i f_{m-1}(x_i))$

权值的计算

- 求权值：$\sum_{i=1}^{N} \overline{w}_{mi} \exp(-y_i \alpha G(x_i))$

$$= \sum_{y_i = G_m(x_i)} \overline{w}_{mi} e^{-\alpha} + \sum_{y_i \neq G_m(x_i)} \overline{w}_{mi} e^{\alpha}$$

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

- 将G*(x)带入：$G_m^*(x) = \arg\min_G \sum_{i=1}^{N} \overline{w}_{mi} I(y_i \neq G(x_i))$
- 求导，得到

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

分类错误率

- 分类错误率为：

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

权值的更新

- 由模型

$$f_m(x) = f_{m-1}(x) + \alpha_m G_m(x)$$

- 以及权值

$$\overline{w}_{mi} = \exp(-y_i f_{m-1}(x_i))$$

- 可以方便的得到：

$$\overline{w}_{m+1,i} = \overline{w}_{m,i} \exp(-y_i \alpha_m G_m(x))$$

权值和错误率的关键解释

- 事实上，根据Adaboost的构造过程，权值调整公式为：

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

- 二者做除，得到 $e^{2\alpha_m} = \dfrac{e_m}{1-e_m}$
- 从而：

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

总结

- AdaBoost的训练误差是以指数速率下降的
- AdaBoost算法不需要事先知道下界γ，AdaBoost具有自适应性，它能适应若分类器格子的训练误差率。（“适应”Adaptive的由来）

我们在这里

- 更多算法面试题在 7 七月算法官网
  - http://www.julyedu.com/
    - 免费视频
    - 直播课程
    - 问答社区

- contact us：微博
  - @研究者July
  - @七月问答
  - @邹博_机器学习

感谢大家！

恳请大家批评指正！