

4. Adaboost

4.1 Adaboost实现原理

4.1.1 Boosting 提升

1) Adaptive Boosting (自适应增强)

它的自适应在于：前一个基本分类器分错的样本会得到加强，加权后全体样本再次被用来训练下一个基分类器。

同时，在每一轮中加入一个新的弱分类器，直到达到某个预定的足够小的错误率，或达到预先指定的最大迭代次数。

具体步骤：

① 初始化训练数据权重分布。如果有 N 个样本，则每个训练样本最开始时都被赋予相同的权重： $1/N$ 。

② 训练弱分类器。如果某个样本点已经被准确地分类，那么在构造下一个训练集中，它的权重就会降低。相反，如果该样本点分类错误，那么它的权重就会提升。然后，权重更新过的样本



扫描全能王 创建

根据弱分类器误差率调整弱分类器权重
与样本权重

Date:

Page:

本集被用于训练下一个分类器，整个训练过程如此迭代地进行下去。

③ 将各个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策作用，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发挥较小作用。

4.2 Adaboost做具体实现

3. Adaboost

boosting家族要解决的问题

- 1) 如何计算误差率 e ?
- 2) 如何得到弱分类器的权重 α ?
- 3) 如何更新样本权重 D ?
- 4) 使用何种集成策略?

4.2.1 分类

Adaboost 损失函数: **指数函数**

即定义损失函数为:

$$\arg \min_{\alpha, f} \sum_{i=1}^m \exp(-y_i f(x_i))$$

m : 样本数
 y_i : 样本真实标签
 $f(x_i)$: 强分类器的预测结果

1) 误差率 e

1) 误差率:

e_k : 第 k 个弱分类器 $G_k(x)$ 在训练集上的加权误差率为:

$$e_k = P(G_k(x_i) \neq y_i) = \sum_{i=1}^m w_i I(G_k(x_i) \neq y_i)$$

\downarrow
样本数

[回归学无]
[分类学有]

2) 弱分类器权重

2) 弱分类器权重:

$$\alpha_k = \frac{1}{2} \log \frac{1 - e_k}{e_k} \quad e_k \downarrow, \alpha_k \uparrow$$

3) 更新样本权重

3) 更新样本权重:

$$w_{k+1, i} = \frac{w_{k, i}}{Z_k} \exp(-\alpha_k y_i G_k(x_i))$$

4) 集成策略

4) 集成策略: 加权表决法, 最终的强分类器

$$f(x) = \text{sign} \left(\sum_{k=1}^K \alpha_k G_k(x) \right)$$

4.2.2 回归

R2 回归流程:

R2 回归算法流程:

输入为样本集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$,

弱学习器数, 弱学习器迭代次数 K 。

输出: 最终的强学习器 $f(x)$

1) 初始化样本权重为:

$$D(1) = (w_{11}, w_{12}, \dots, w_{1m}) : w_{1i} = \frac{1}{m}, i = 1, 2, \dots, m$$

2) 对于 $k = 1, 2, \dots, K$:

a) 使用具有权重 D_k 的样本集来训练数据, 得到弱分类器 $G_k(x)$

★ b) 计算训练集的最大误差

$$E_k = \max |y_i - G_k(x_i)| \quad (i = 1, 2, \dots, m)$$

(分类误差率计算)

★ c) 计算每个样本的相对误差:

• 绝对误差: $e_{ki} = \frac{|y_i - G_k(x_i)|}{E_k}$

• 平方误差: $e_{ki} = \frac{(y_i - G_k(x_i))^2}{E_k^2}$

★ d) 计算回归误差率:

• 指数误差:

$$e_k = \sum_{i=1}^m w_{ki} e_{ki}$$

$$e_{ki} = 1 - \exp \left(-\frac{|y_i - G_k(x_i)|}{E_k} \right)$$

e) 计算弱分类器系数:

$$\alpha_k = \frac{e_k}{1 - e_k}$$

f) 更新样本集的权重分布:

$$w_{k+1,i} = \frac{w_{ki}}{Z_k} \alpha_k^{1 - e_{ki}}$$

$$Z_k = \sum_{i=1}^m w_{ki} \alpha_k^{1 - e_{ki}}$$

$$f(x) = \sum_{k=1}^K (\ln \frac{1}{\alpha_k}) g_k(x)$$

其中 $g_k(x)$ 是所有 $\alpha_k G_k(x)$, $k=1,2,\dots,K$ 的中位数。

但有点看不懂依据, 还是再用加权平均吧:

$$f(x) = \sum_{k=1}^K (\ln \frac{1}{\alpha_k}) G_k(x)$$

当然, 也可以自定义组合策略。

背诵版

回答一个问题:

Adaboost 递归原理: [验证新式不一致 \Leftarrow 保持模式不一致]

过程与做分类一致:

首先初始化所有样本权重相同;

(MSE)

接下来, 训练一个基学习器, 计算在训练集上的最大误差, 然后计算每个样本的相对误差, 得到样本加权误差。

并根据这个误差来设置这个弱学习器的权重和对应训练样本权重。总体还是误差率高, 弱学习器权重低, 对应样本权重高。

最后, 由多个训练若干基学习器组合为强学习器。

4.3 Adaboost 优缺点

⑥ Adaboost优缺点

Adaboost的基学习器最好的是：决策树与神经网络。

对于决策树：Adaboost分类用于CART ~~树~~ 分类树，

Adaboost回归用了CART 回归树

优点：
1) 作为分类器时，分类精度很高
2) 可以集成同时分类与回归
3) 不易发生过拟合



扫描全能王 包

Date: / / Page:

缺点：

对异常样本敏感，异常样本在迭代中可能会获得较高权重，最终影响强学习器的预测准确性。