

刘建平Pinard

十年码农，对数学统计学，数据挖掘，机器学习，大数据平台，大数据平台应用开发，大数据可视化感兴趣。

博客园

首页

新随笔

联系

订阅

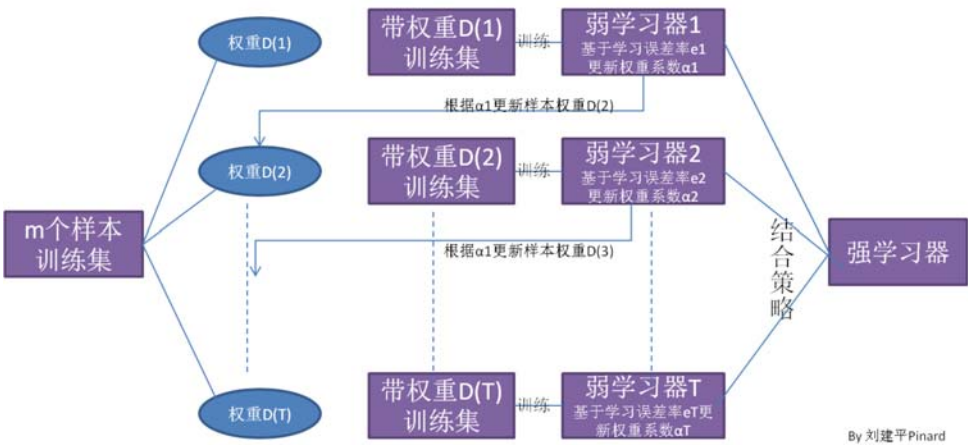
管理

集成学习之Adaboost算法原理小结

在集成学习原理小结中，我们讲到了集成学习按照个体学习器之间是否存在依赖关系可以分为两类，第一个是个体学习器之间存在强依赖关系，另一类是个体学习器之间不存在强依赖关系。前者的代表算法就是是boosting系列算法。在boosting系列算法中，Adaboost是最著名的算法之一。Adaboost既可以用作分类，也可以用作回归。本文就对Adaboost算法做一个总结。

1. 回顾boosting算法的基本原理

在集成学习原理小结中，我们已经讲到了boosting算法系列的基本思想，如下图：



从图中可以看出，Boosting算法的工作机制是首先从训练集用初始权重训练出一个弱学习器1，根据弱学习的学习误差率来更新训练样本的权重，使得之前弱学习器1学习误差率高的训练样本点的权重变高，使得这些误差率高的点在后面的弱学习器2中得到更多的重视。然后基于调整权重后的训练集来训练弱学习器2，如此重复进行，直到弱学习器数达到事先指定的数目T，最终将这T个弱学习器通过集合策略进行整合，得到最终的强学习器。

不过有几个具体的问题Boosting算法没有详细说明。

- 1) 如何计算学习误差率e?
 - 2) 如何得到弱学习器权重系数α?
 - 3) 如何更新样本权重D?
 - 4) 使用何种结合策略?
- 只要是boosting大家族的算法，都要解决这4个问题。那么Adaboost是怎么解决的呢？

2. Adaboost算法的基本思路

我们这里讲解Adaboost是如何解决上一节这4个问题的。

假设我们的训练集样本是

$$T = \{(x, y_1), (x_2, y_2), \dots (x_m, y_m)\}$$

训练集的在第k个弱学习器的输出权重为

$$D(k) = (w_{k1}, w_{k2}, \dots w_{km}); \quad w_{1i} = \frac{1}{m}; \quad i = 1, 2, \dots m$$

首先我们看看Adaboost的分类问题。

分类问题的误差率很好理解和计算。由于多元分类是二元分类的推广，这里假设我们是二元分类问题，输出为{-1, 1}，则第k个弱分类器 $G_k(x)$ 在训练集上的加权误差率为

公告

★珠江追梦，饮岭南茶，恋鄂北家★
昵称：刘建平Pinard
园龄：2年1个月
粉丝：2783
关注：15
+加关注

随笔分类(121)

- 0040. 数学统计学(4)
- 0081. 机器学习(69)
- 0082. 深度学习(11)
- 0083. 自然语言处理(23)
- 0084. 强化学习(12)
- 0121. 大数据挖掘(1)
- 0122. 大数据平台(1)

随笔档案(121)

- 2018年11月 (1)
- 2018年10月 (3)
- 2018年9月 (3)
- 2018年8月 (4)
- 2018年7月 (3)
- 2018年6月 (3)
- 2018年5月 (3)
- 2017年8月 (1)
- 2017年7月 (3)
- 2017年6月 (8)
- 2017年5月 (7)
- 2017年4月 (5)
- 2017年3月 (10)
- 2017年2月 (7)
- 2017年1月 (13)
- 2016年12月 (17)
- 2016年11月 (22)
- 2016年10月 (8)

常去的机器学习网站

52 NLP
Analytics Vidhya
机器学习库
机器学习路线图
强化学习入门书
深度学习进阶书
深度学习入门书

积分与排名

积分 - 366770
排名 - 537

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

接着我们看弱学习器权重系数,对于二元分类问题，第k个弱分类器 $G_k(x)$ 的权重系数为

$$\alpha_k = \frac{1}{2} \log \frac{1 - e_k}{e_k}$$

为什么这样计算弱学习器权重系数？从上式可以看出，如果分类误差率 e_k 越大，则对应的弱分类器权重系数 α_k 越小。也就是说，误差率小的弱分类器权重系数越大。具体为什么采用这个权重系数公式，我们在讲Adaboost的损失函数优化时再讲。

第三个问题，更新更新样本权重D。假设第k个弱分类器的样本集权重系数为 $D(k) = (w_{k1}, w_{k2}, \dots, w_{km})$ ，则对应的第k+1个弱分类器的样本集权重系数为

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

这里 Z_k 是规范化因子

$$Z_k = \sum_{i=1}^m w_{ki} \exp(-\alpha_k y_i G_k(x_i))$$

从 $w_{k+1,i}$ 计算公式可以看出，如果第i个样本分类错误，则 $y_i G_k(x_i) < 0$ ，导致样本的权重在第k+1个弱分类器中增大，如果分类正确，则权重在第k+1个弱分类器中减少。具体为什么采用样本权重更新公式，我们在讲Adaboost的损失函数优化时再讲。

最后一个问题是集合策略。Adaboost分类采用的是加权表决法，最终的强分类器为

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

接着我们看看Adaboost的回归问题。由于Adaboost的回归问题有很多变种，这里我们以Adaboost R2算法为准。

我们先看看回归问题的误差率的问题，对于第k个弱学习器，计算他在训练集上的最大误差

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

然后计算每个样本的相对误差

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

这里是误差损失为线性时的情况，如果我们用平方误差，则 $e_{ki} = \frac{(y_i - G_k(x_i))^2}{E_k^2}$ ，如果我们用的是指数误差，则 $e_{ki} = 1 - \exp(-\frac{y_i - G_k(x_i)}{E_k})$

最终得到第k个弱学习器的 误差率

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

我们再来看看如何得到弱学习器权重系数 α 。这里有：

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

对于更新更新样本权重D，第k+1个弱学习器的样本集权重系数为

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

这里 Z_k 是规范化因子

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

最后是结合策略，和分类问题稍有不同，采用的是对加权的弱学习器取中位数的方法，最终的强回归器为

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

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

3. AdaBoost分类问题的损失函数优化

刚才上一节我们讲到了分类Adaboost的弱学习器权重系数公式和样本权重更新公式。但是没有解释选择这个公式的原因，让人觉得是魔法公式一样。其实它可以从Adaboost的损失函数推导出来。

从另一个角度讲，Adaboost是模型为加法模型，学习算法为前向分步学习算法，损失函数为指数函数的分类问题。

阅读排行榜
1. 梯度下降（Gradient Descent）小结(66)
2. 梯度提升树(GBDT)原理小结(24)
3. 线性判别分析LDA模型基础(78921)
4. word2vec原理(二) 基于Hierarchical Softmax的模型(141)
5. 主成分分析（PCA）原理总结(57210)

评论排行榜
1. 梯度提升树(GBDT)原理小结(249)
2. word2vec原理(二) 基于Hierarchical Softmax的模型(141)
3. 集成学习之Adaboost算法原理小结(137)
4. 谱聚类（spectral clustering）原理总结(118)
5. 梯度下降（Gradient Descent）小结(108)

推荐排行榜
1. 梯度下降（Gradient Descent）小结(66)
2. 奇异值分解(SVD)原理与在降维中的应用(41)
3. 梯度提升树(GBDT)原理小结(24)
4. 集成学习原理小结(24)
5. 集成学习之Adaboost算法原理小结(21)

模型为加法模型好理解，我们的最终的强分类器是若干个弱分类器加权平均而得到的。

前向分步学习算法也好理解，我们的算法是通过一轮轮的弱学习器学习，利用前一个弱学习器的结果来更新后一个弱学习器的训练集权重。也就是说，第k-1轮的强学习器为

$$f_{k-1}(x) = \sum_{i=1}^{k-1} \alpha_i G_i(x)$$

而第k轮的强学习器为

$$f_k(x) = \sum_{i=1}^k \alpha_i G_i(x)$$

上两式一比较可以得到

$$f_k(x) = f_{k-1}(x) + \alpha_k G_k(x)$$

可见强学习器确实是通过前向分步学习算法一步步而得到的。

Adaboost损失函数为指数函数，即定义损失函数为

$$\underbrace{\arg \min}_{\alpha, G} \sum_{i=1}^m \exp(-y_i f_k(x))$$

利用前向分步学习算法的关系可以得到损失函数为

$$(\alpha_k, G_k(x)) = \underbrace{\arg \min}_{\alpha, G} \sum_{i=1}^m \exp[(-y_i)(f_{k-1}(x) + \alpha G(x))]$$

令 $w'_{ki} = \exp(-y_i f_{k-1}(x))$ ，它的值不依赖于 α, G ，因此与最小化无关，仅仅依赖于 $f_{k-1}(x)$ ，随着每一轮迭代而改变。

将这个式子带入损失函数，损失函数转化为

$$(\alpha_k, G_k(x)) = \underbrace{\arg \min}_{\alpha, G} \sum_{i=1}^m w'_{ki} \exp[-y_i \alpha G(x)]$$

首先，我们求 $G_k(x)$ ，可以得到

$$G_k(x) = \underbrace{\arg \min}_G \sum_{i=1}^m w'_{ki} I(y_i \neq G(x_i))$$

将 $G_k(x)$ 带入损失函数，并对 α 求导，使其等于0，则就得到了

$$\alpha_k = \frac{1}{2} \log \frac{1 - e_k}{e_k}$$

其中， e_k 即为我们前面的分类误差率。

$$e_k = \frac{\sum_{i=1}^m w'_{ki} I(y_i \neq G(x_i))}{\sum_{i=1}^m w'_{ki}} = \sum_{i=1}^m w_{ki} I(y_i \neq G(x_i))$$

最后看样本权重的更新。利用 $f_k(x) = f_{k-1}(x) + \alpha_k G_k(x)$ 和 $w'_{ki} = \exp(-y_i f_{k-1}(x))$ ，即可得：

$$w'_{k+1,i} = w'_{ki} \exp[-y_i \alpha_k G_k(x)]$$

这样就得到了我们第二节的样本权重更新公式。

4. AdaBoost二元分类问题算法流程

这里我们对AdaBoost二元分类问题算法流程做一个总结。

输入为样本集 $T = \{(x, y_1), (x_2, y_2), \dots (x_m, y_m)\}$ ，输出为 $\{-1, +1\}$ ，弱分类器算法，弱分类器迭代次数K。

输出为最终的强分类器 $f(x)$

1) 初始化样本集权重为

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

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 $G_k(x)$ 的分类误差率

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

c) 计算弱分类器的系数

$$\alpha_k = \frac{1}{2} \log \frac{1 - e_k}{e_k}$$

d) 更新样本集的权重分布

$$w_{k+1,i} = \frac{w_{ki}}{Z_k} \exp(-\alpha_k y_i G_k(x_i)) \quad i = 1, 2, \dots, m$$

这里 Z_k 是规范化因子

$$Z_k = \sum_{i=1}^m w_{ki} \exp(-\alpha_k y_i G_k(x_i))$$

3) 构建最终分类器为:

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

对于Adaboost多元分类算法，其实原理和二元分类类似，最主要区别在弱分类器的系数上。比如Adaboost SAMME算法，它的弱分类器的系数

$$\alpha_k = \frac{1}{2} \log \frac{1 - e_k}{e_k} + \log(R - 1)$$

其中R为类别数。从上式可以看出，如果是二元分类，R=2，则上式和我们的二元分类算法中的弱分类器的系数一致。

5. Adaboost回归问题的算法流程

这里我们对AdaBoost回归问题算法流程做一个总结。AdaBoost回归算法变种很多，下面的算法为Adaboost R2回归算法过程。

输入为样本集 $T = \{(x, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ，弱学习器算法，弱学习器迭代次数K。

输出为最终的强学习器 $f(x)$

1) 初始化样本集权重为

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

2) 对于k=1, 2, ... K:

a) 使用具有权重 D_k 的样本集来训练数据，得到弱学习器 $G_k(x)$

b) 计算训练集上的最大误差

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

c) 计算每个样本的相对误差:

$$\text{如果是线性误差, 则 } e_{ki} = \frac{|y_i - G_k(x_i)|}{E_k};$$

$$\text{如果是平方误差, 则 } e_{ki} = \frac{(y_i - G_k(x_i))^2}{E_k^2}$$

$$\text{如果是指数误差, 则 } e_{ki} = 1 - \exp\left(-\frac{|y_i - G_k(x_i)|}{E_k}\right)$$

d) 计算回归误差率

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

c) 计算弱学习器的系数

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

d) 更新样本集的权重分布为

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

这里 Z_k 是规范化因子

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

3) 构建最终强学习器为:

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

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

6. Adaboost算法的正则化

为了防止Adaboost过拟合，我们通常也会加入正则化项，这个正则化项我们通常称为步长(learning rate)。定义为 ν ,对于前面的弱学习器的迭代

$$f_k(x) = f_{k-1}(x) + \alpha_k G_k(x)$$

如果我们加上了正则化项，则有

$$f_k(x) = f_{k-1}(x) + \nu \alpha_k G_k(x)$$

ν 的取值范围为 $0 < \nu \leq 1$ 。对于同样的训练集学习效果，较小的 ν 意味着我们需要更多的弱学习器的迭代次数。通常我们用步长和迭代最大次数一起来决定算法的拟合效果。

7. Adaboost小结

到这里Adaboost就写完了，前面有一个没有提到，就是弱学习器的类型。理论上任何学习器都可以用于Adaboost.但一般来说，使用最广泛的Adaboost弱学习器是决策树和神经网络。对于决策树，Adaboost分类用了CART分类树，而Adaboost回归用了CART回归树。

这里对Adaboost算法的优缺点做一个总结。

Adaboost的主要优点有：

- 1) Adaboost作为分类器时，分类精度很高
- 2) 在Adaboost的框架下，可以使用各种回归分类模型来构建弱学习器，非常灵活。
- 3) 作为简单的二元分类器时，构造简单，结果可理解。
- 4) 不容易发生过拟合

Adaboost的主要缺点有：

- 1) 对异常样本敏感，异常样本在迭代中可能会获得较高的权重，影响最终的强学习器的预测准确性。

(欢迎转载，转载请注明出处。欢迎沟通交流：liujianping-ok@163.com)

分类: [0081. 机器学习](#)

标签: [集成学习](#)

好文要顶

关注我

收藏该文





[刘建平Pinard](#)
关注 - 15
粉丝 - 2783

[+加关注](#)

« 上一篇: [集成学习原理小结](#)

» 下一篇: [scikit-learn Adaboost类库使用小结](#)

21

0

posted @ 2016-12-05 22:26 刘建平Pinard 阅读(30632) 评论(137) 编辑 收藏

评论列表

#101楼 2018-07-25 22:12 某小白啊

博主，想请教一个弱弱的问题，这儿AdaBoost分类的label是{1, -1}，那其实平时遇到的分类问题一般的label是{1, 0}，我有俩个问题，# 1如果label是1、0的话好像不太好使用指数损失函数，是吗？# 2如果问题# 1的答案是肯定的话，那么是不是意味着想用AdaBoost做分类，把label从{1, 0}映射成{1, -1}就行了？

支持(0) 反对(0)

#102楼[楼主]] 2018-07-26 10:40 刘建平Pinard	
@ 随风vft 你好，参考我80楼的回答，当时由于分类和回归参考的不同的论文，所以符号可能有些让人误解。	
你可以理解为在回归的时候，弱回归器的系数不是 α_k ,而是 $\ln \frac{1}{\alpha_k}$ ，在分类的时候，弱分类器的系数才是 α_k ,	支持(1) 反对(0)
#103楼[楼主]] 2018-07-26 10:44 刘建平Pinard	
@ 某小白啊 你好，你说的对，如果是指数分类器，lable就不好是0和1，而最好是1和-1，这样会方便优化。不过现在的算法库比如sklearn大部分都可以自己处理这个优化时需要的转化，你不用自己去处理了，就算用0,1也没有关系。	
所以你的class的label用0和1即可，当然理解算法的时候需要知道这个过程就行。	支持(0) 反对(0)
#104楼 2018-07-26 10:54 某小白啊	
@ 刘建平Pinard 赞楼主的耐心答疑啊，感谢~	支持(0) 反对(0)
#105楼 2018-07-26 16:27 tqtgML	
您好大神，我这里有一个问题一直没有想明白，希望您能帮忙解答一下。对于这个adaboost算法来说，一个重要的问题是去确定每一轮提升的样本权重是吧？但是这个样本权重在训练时，怎么引入进去得呢。步骤上说训练带有权重的训练数据，怎么算法带有权值，难道是wi去乘以xi每个样本加那个y值吗。是这样吗？如果是的话，我就理解提升算法是可以用其他比如LR svm 之类的算法。如果不是，那我在每次的提升时，我的若分类器的损失函数 的形式就变了，这样就不好泛化了。谢谢您 期待回复	支持(1) 反对(0)
#106楼 2018-08-01 21:01 小白学机器学习	
您好楼主，@ 刘建平Pinard 我想问一下，adaboost是不是比SVM和神经网络，贝叶斯估计更容易过拟合	支持(0) 反对(0)
#107楼[楼主]] 2018-08-02 12:54 刘建平Pinard	
@ 小白学机器学习 你好，adaboost是集成学习的方法，在同等情况下，要比SVM之类的方法更可能发生拟合。不过我们可以通过一些正则化的参数来调参，减少Adaboost过拟合的情况。所以过拟合也不是绝对的。	支持(0) 反对(0)
#108楼 2018-08-07 16:36 蓝色记忆2013	
博主您好： 有两个问题请教一下： (1) 在选择基分类器的时候，选择分类误差率最小的基分类器和选择指数损失函数最小的基分类器是等价的吧？ (2) 如果（1）的说法对的话，那么adaboost算法在选择基分类器时以分类误差率作为标准而不以指数损失函数作为标准的原因是因为指数运算复杂吗？	支持(0) 反对(0)
#109楼[楼主]] 2018-08-08 10:23 刘建平Pinard	
@ 蓝色记忆2013 你好！ 1) 通过推导会发现选择选择指数损失函数最小的基分类器可以使分类误差率最小，但是实际上还有其他类别的基分类器也可以使分类误差率最小的，比如对数损失函数。 也就是选择指数损失函数是一个充分不必要条件。 2) 分类误差率是衡量算法优劣的标准。而优化指数损失函数的目标就是减少分类误差率。所以我们选择基分类器的时候要充分考虑误差率这个目标。那么可以选择的损失函数可以是指数函数，也可以是对数损失函数。 所以你上面说的“以分类误差率作为标准而不以指数损失函数作为标准的原因是因为指数运算复杂”这句话其实是不能一概而论的。	支持(0) 反对(0)
#110楼 2018-08-21 16:13 蓝色记忆2013	
您好，请问您在41楼提到的 W ki 的所有和等于1对吗？ W ' ki 的和并不等于1吧？	支持(0) 反对(0)
#111楼[楼主]] 2018-08-21 17:22 刘建平Pinard	
@ 蓝色记忆2013 你好！是的， w_{ki} 做了规范化，而 w'_{ki} 没有。 所以 w_{ki} 之和为1，但是 w'_{ki} 不是	

#112楼 2018-08-27 11:48 cuckoonuts

@ 刘建平Pinard
博主您好：
有两个问题请教一下：
1.关于ek
对于任意二分类问题，从常规思路考虑，ek都应该小于0.5，因为大于0.5没有实际应用意义，还不如直接猜，对于大于0.5的ek，完全可以取反，使得ek<0.5。

对于二元分类问题，第k个弱分类器 $G_k(x)$ 的权重系数为

$$\alpha_k = \frac{1}{2} \log \frac{1 - e_k}{e_k}$$

由图中，当ek等于0.5时，ak=0，相当于随机猜。当ek<0.5时，比如我们取0.2，ak=1/2*log4，显然，这个ak>0，博主在2楼对一楼的回复中，似乎在肯定一楼的论述，但是给出的结论是ek<0.5,ak<0,没有提升，附一楼图片

$$\alpha_k = \frac{1}{2} \log \frac{1 - e_k}{e_k}$$

为什么这样计算弱学习器权重系数？从公式可以看出，如果分类误差 e_k 越小，则对应的弱分类器权重系数 α_k 越大。
是说，误差率小的弱分类器权重系数越大，具体为什么采用这个权重系数公式，我们在讲Adaboost的损失函数优化时再

第三个问题，更新更新样本权重D，假设第k个弱分类器的样本权重系数为 $D(k) = (w_{k1}, w_{k2}, \dots, w_{km})$ ，则对应
实际上，对于二分类问题，是不可能发生 $e_k < 0.5$ 的情况，因为假如他0.3，我把结果反转，准确率就变成0.7了。
对于 $e_k < 0.5$ 的情况，分类正确的结果会被更加注意，这点可以理解：因为我这边本身就是一个坏的分类器，你居然还分正确了，就说明其实你是站错队的。

另外，9楼：
9楼的说法是不是刚好都说反了，感觉ek>0.5的事情不应该发生，ek>0.5时应该引起注意，但是博主在10楼给与了肯定
我一下子糊涂了，是自己理解有偏差吗？

d) 更新样本集的权重分布为

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

这里 Z_k 是规范化因子

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

红色区域内的部分是怎么来的？是规定的，还是求解的？

支持(0) 反对(0)

#113楼[楼主] 2018-08-27 15:17 刘建平Pinard

@cuckoonuts
你好！
1) 上面评论里面只是做算法理论探讨，实际算法训练的时候我们都是会按照通常的 e_k 越小则权重越大来处理。毕竟 $e_k > 0.5$ 是小概率事件。

2) 这个我倒没有深究，相关讨论论文都是直接给出。我参考的是这一篇论文：
Boosting for Regression Transfer
David Pardoe and Peter Stone

支持(0) 反对(0)

#114楼 2018-08-29 08:59 kylin0228

博主您好
关于adaboost的回归问题我有个疑问：
请问这和提升数算法的回归问题（李航147）有联系吗？
提升树的回归也是用CART回归，加法模型，前向分布算法，
可是这里你写的adaboost的回归算法我看着和李航那本书上提升树算法的回归问题不一样。。。.
不知道我下面理解的对不对：
首先他没有使用相对误差，用了绝对误差
第二，他并没有更新样本权重
第三，他没有对每个弱学习器加个权重
第四：他用的是不断拟合残差，而adaboost回归我看着并不仅是拟合残差，而是每个弱学习器都用了全部样本再次拟合

在实践中，是adaboost回归常用还是提升树的回归常用呢？

谢谢博主

支持(0) 反对(0)

#115楼[楼主] 2018-08-29 10:58 刘建平Pinard

@ kylin0228
你好，李航的提升树那部分对应我的博客里面的GBDT回归树,不是Adaboost。
Adaboost和GBDT我的博文都有专门讲到。

目前看 GBDT，以及它的升级版xgBoost，lightGBM用的比较多。

支持(0) 反对(0)

#116楼 2018-08-29 15:35 alexinlove

楼主您好，您在第75楼推导的时候我有个问题没搞清楚，为什么对期望的导数会得出条件概率 $P(f(x) = 1|x)$ 和 $P(f(x) = -1|x)$ 呢

支持(0) 反对(0)

#117楼[楼主] 2018-08-30 10:38 刘建平Pinard

@ alexinlove

你好，这个导数并没有影响这个条件概率。原指数损失函数是一个二分类的期望表达式。其实本来就是等于 $= e^{-H(x)}P(f(x) = 1|x) + e^{H(x)}P(f(x) = -1|x)$

支持(0) 反对(0)

#118楼 2018-09-02 14:56 siriushsh

您好，第二部分，您写道“Adaboost分类采用的是加权平均法”，这个平均是否有误？公式没有体现出平均，“加权表决”是否更合理

支持(0) 反对(0)

#119楼[楼主] 2018-09-02 15:56 刘建平Pinard

@ siriushsh

你好，用加权表决来形容分类问题的确更好一些，原文已经修改，感谢指出。

支持(0) 反对(0)

#120楼 2018-09-26 21:53 JustStudy

楼主您好！

发现您的博客写的太好了，初学这个算法，我有一点想不通，文中算法对应的数据集 (x1,y1) ...假如说，多个特征值对应一个目标值的时候怎么办？

支持(0) 反对(0)

#121楼[楼主] 2018-09-27 10:52 刘建平Pinard

@ JustStudy

你好，这里x1可以是一个向量，也就是如果有N个特征，那么x1就是一个N维向量

支持(0) 反对(0)

#122楼 2018-10-30 19:41 z_dominic

非常感谢！

支持(0) 反对(0)

#123楼 2018-11-01 10:49 小北潜行

老师，在最后小结的部分提到了adaboost的优点是不容易过拟合。而在缺点中说对异常值非常的敏感。这两点是否冲突呢？

支持(0) 反对(0)

#124楼 2018-11-03 21:54 橘子味choco

博主，又请教您了，请问我是否可以这样理解：

1. Wmi仅仅只是由W'mi除以归一化后定义出来的？

2. 分类误差率也是由分类错误的损失函数和 除以 所有样本的损失函数和定义出来的？

3. 极小化指数损失函数等价于最小化分类误差，其实是该次迭代时由该次基分类器决定每个样本的损失函数可能扩大可能减小，只要让增加值最小，减少值最大就可以让总的损失函数最小，所以选择分类错误的样本的原损失函数值之和最小的分类器

麻烦您了~

支持(0) 反对(0)

#125楼[楼主] 2018-11-05 12:47 刘建平Pinard

@ 橘子味choco

1. 对的，当然这是可以推导出来的。

2. 在每轮迭代里面，分类误差率应该是错误分类的样本权重和除以总的样本权重和，而总的样本权重和为1。每轮迭代的损失函数目标也是最小化分类加权后的误差，所以不能说是**分类错误的损失函数和 除以 所有样本的损失函数和**，而应该是**分类错误的样本权重和除以 所有样本的权重和**

3. 这部分没问题。

支持(0) 反对(0)

#126楼 2018-11-05 13:18 橘子味choco

@ 刘建平Pinard

非常感谢~

支持(0) 反对(0)

#127楼 2018-11-10 10:36 bakezq

@ 刘建平Pinard

你好，你说的权重决策树如果是连续值的话特征值要乘以权重再计算方差，但是决策树不是计算的giniindex和information gain吗

更新：

我经过查看milk/sklearn源码得知：

有两种做法：

1：一层树，或者说不像是决策树，只是遍历所有feature和feature_value,看哪种划分分离错误率最低，就和公式中写的一样，将错分类的权重相加就行。

2：一层或多层，原理还是使用gini或者entropy,但是在计算gini或者entropy的过程中计算比例的时候不再是计算比例(比如1标签有几个，再除以该节点所有样本个数)

而是变为了：

计算权重比例，比如：

tags: 1, 0, 1

weights:0.5,1.5,0.8

那么tag为1的样本所占的比例为: $0.5 + 0.8 / (0.5 + 1.5 + 0.8)$

tag为0的样本所占的比例为: $1.5 / (0.5 + 1.5 + 0.8)$

支持(0) 反对(0)

#128楼[楼主] 2018-11-11 21:10 刘建平Pinard

@ bakezq

你好，你说的应该是CART分类树，这个是没有问题的。但是CART回归树不是这样，对于CART回归树，其划分方法应该是计算MSE,MAE这样的值。所以我说后面要乘以权重。

<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html#sklearn.tree.DecisionTreeRegressor>

criterion : string, optional (default="mse")

The function to measure the quality of a split. Supported criteria are "mse" for the mean squared error, which is equal to variance reduction as feature selection criterion and minimizes the L2 loss using the mean of each terminal node, "friedman_mse", which uses mean squared error with Friedman's improvement score for potential splits, and "mae" for the mean absolute error, which minimizes the L1 loss using the median of each terminal node.

支持(0) 反对(0)

#129楼 2018-11-11 21:24 bakezq

@ 刘建平Pinard

是的

回归的我还没去看源码，谢谢你的回复。

支持(0) 反对(0)

#130楼 2018-11-13 15:18 Rykkk

您好，有关于利用神经网络做基学习器来做Adaboost的例子或项目可以参考吗

支持(0) 反对(0)

#131楼[楼主] 2018-11-14 11:13 刘建平Pinard

@ Rykkk

你好，我这边没有。Github上应该有一些，你可以去搜索一下。

比如这个：

<https://github.com/szad670401/Rubost-Chinese-License-Plate-Locate-Using-LBP-adaboost-with-CNN-regression>

支持(0) 反对(0)

#132楼 2018-11-14 14:34 Rykkk

@ 刘建平Pinard

谢谢博主您的回复，非常感谢

支持(0) 反对(0)

#133楼 2018-11-26 11:35 ucasgff

@ 刘建平Pinard

有没有Adaboost多分类的详细算法，他的过程和二分类相同只是计算系数那一步不同吗

支持(0) 反对(0)

#134楼[楼主] 2018-11-27 10:10 刘建平Pinard

@ ucasgff

有些小细节的差别，但是不大。你感兴趣的话可以看下这篇论文<Multi-class AdaBoost>

<http://web.stanford.edu/~hastie/Papers/samme.pdf>

支持(0) 反对(0)

#135楼 2018-12-08 15:45 齐在

博主您好！看了您的博客，想请教您几个问题：

1. w'_{ki} 的值依赖于 $f_{k-1}(x)$ ，而 $f_{k-1}(x)$ 依赖于 α, G ，所以 w'_{ki} 还是依赖于 α, G 。不知道我这个理解哪里出了问题。
2. 为什么AdaBoost通过前向分步算法得到的最优 α_k, G_k 就是最终模型的最优 α_k 和 G_k 呢？
3. 您在41楼的回答中提到，图中第一个十字和第二个十字由于只相差一个规范化因子 Z_k ，所以可以得到第三个式子。可是， $w_{k+1,i}$ 和 $w'_{k+1,i}$ 之间是没有相关关系的，只是符号类似而已，而且 $w_{k+1,i}$ 和 $w'_{k+1,i}$ 的初始值也不一样。那怎么能够根据两个长得相似的递推式子就推出第三个式子呢？

#41楼[楼主] 2018-01-08 17:09 刘建平Pinard

AlbertSR
你好，注意到：

$$w_{k+1,i} = \frac{w_{ki} \exp(-\alpha_k y_i G_k(x_i))}{Z_k}$$
$$w'_{k+1,i} = w'_{ki} \exp[-y_i \alpha_k G_k(x)]$$

即 w_{ki} 仅是比 w'_{ki} 多了一个规范化因子的分母 Z_k 而已，也就是说， w_{ki} 是 w'_{ki} 规范化后的表达式，即可以得到：

$$w_{ki} = \frac{w'_{ki}}{\sum_{i=1}^n w'_{ki}}$$

最后看样本权重的更新，利用 $f_k(x) = f_{k-1}(x) + \alpha_k G_k(x)$ 和 $w'_{ki} = \exp(-y_i f_{k-1}(x))$ ，即可得：

$$w'_{k+1,i} = w'_{ki} \exp[-y_i \alpha_k G_k(x)]$$

4. 这样就得到了我们第二节的样本权重更新公式。

《统计学习方法》中也是说这个式子和样本权重更新公式只相差规范化因子，因而等价。我的疑问是：两个相似的公式，只因为相差规范化因子，就可以等价了吗？

这几个问题想了好久都不明白，希望博主能够帮忙解答一下，非常感谢！

支持(0) 反对(0)

#136楼[楼主] 2018-12-10 11:05 刘建平Pinard

@齐在

你好！

- 1) 这里要注意我们是在进行迭代，在第 k 轮迭代的时候，第 $k-1$ 轮的迭代已经结束， $f_{k-1}(x)$ 已经是常数了，所以可以说此时 w'_{ki} 不依赖于 α, G 。
- 2) 这个推导里面已经有的，其实就是通过求导等于0，得到极值点得到的。每一步推导的目的就是得到最优的 α, G ，把所有的 α, G 组合起来就是最终的模型。
- 3) 两者除了更新公式相差规范化因子以外，初始值也是相差规范化因子，因此根据数学归纳法，可以确定那个两者只相差一个规范化因子。

更新公式你已经清楚了，我们看看初值。对于初值 w_{1i} ，其表达式是每个样本权重一样(参看第四节的算法)，即：

$$w_{1i} = \frac{1}{m}$$

对于初值 w'_{1i} ，其表达式是：

$$w'_{1i} = \exp(-y_i f_0(x)) = \exp(-y_i * 0) = 1$$

这里 $f_0(x)$ 初始值为0，不参与实际的1-K轮的迭代，你看明白了这个初始值应该就明白了。

支持(0) 反对(0)

#137楼 2018-12-13 01:01 huaweipretty

博主您好，adaboost回归的最终输出，论文和sklearn的源码中貌似和您写的不太一样。并不是对每个预测结果加权求和，而仅仅取 estimator_weights中位数对应的弱分类器值 sklearn源码如下，不知理解对否

```
def _get_median_predict(self, X, limit):
    # Evaluate predictions of all estimators
    predictions = np.array([
        est.predict(X) for est in self.estimators_[:limit]])

    # Sort the predictions
    sorted_idx = np.argsort(predictions, axis=1)

    # Find index of median prediction for each sample
    weight_cdf = stable_cumsum(self.estimator_weights_[sorted_idx], axis=1)
    median_or_above = weight_cdf >= 0.5 * weight_cdf[:, -1][:, np.newaxis]
    median_idx = median_or_above.argmax(axis=1)

    median_estimators = sorted_idx[np.arange(X.shape[0]), median_idx]

    # Return median predictions
    return predictions[np.arange(X.shape[0]), median_estimators]
```

支持(0) 反对(0)