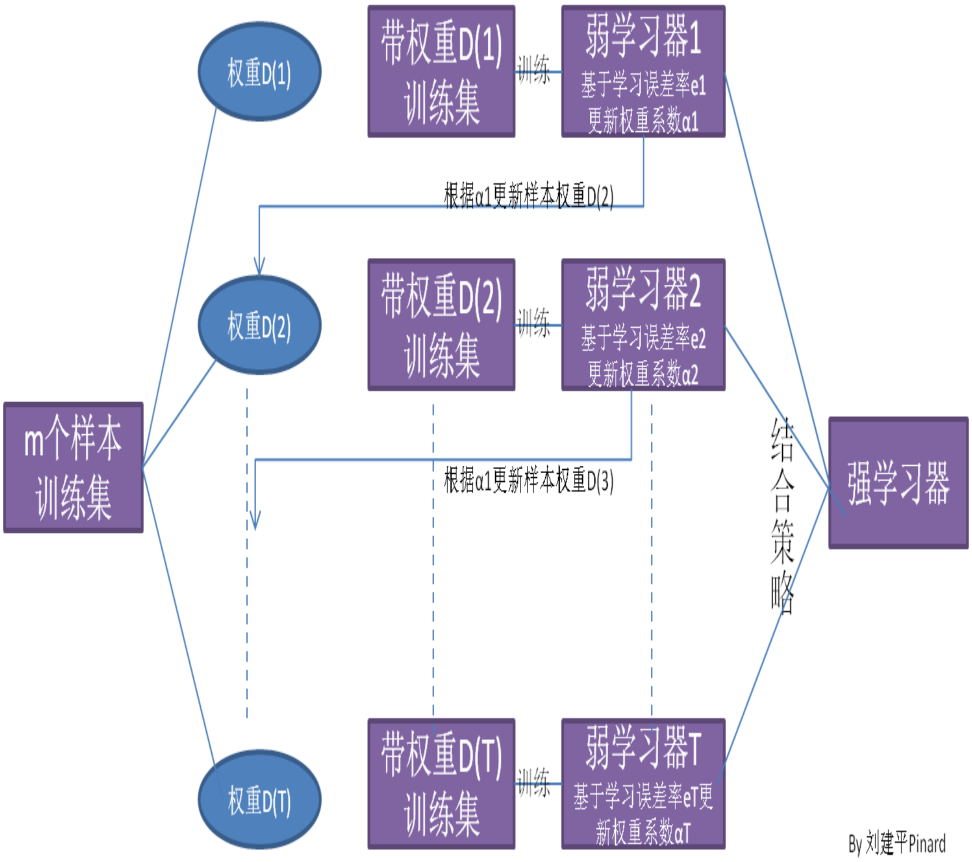
集成学习之Adaboost算法原理小结

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表现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

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

训练集的在第k个弱学习器的输出权重为

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

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

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

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$D(k) = (w_{k1}, w_{k2}, \ldots w_{km})$，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分类采用的是加权平均法，最终的强分类器为

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

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

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

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$\alpha_k G_k(x), k = 1,2,....K$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。

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

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

可见强学习器的确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

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

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$w_{ki}^{'} = exp(-y_if_{k-1}(x))$, 它的值不依赖于$\alpha, G$,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。

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

最后看样本权重的更新。利用$f_{k}(x) = f_{k-1}(x) + \alpha_kG_k(x)$和$w_{ki}^{'} = exp(-y_if_{k-1}(x))$，即可得：

$$w_{k+1,i}^{'} = w_{ki}^{'}exp[-y_i\alpha_kG_k(x)]$$

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

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

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

$$w_{k+1,i} = \frac{w_{ki}}{Z_K} exp(-\alpha_k y_i G_k(x_i)) \;\; i = 1, 2, \ldots m$$

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：

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

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

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

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

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

c) 计算每个样本的相对误差:

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

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

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：

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

其中，$g(x)$是所有$\alpha_k G_k(x), k = 1, 2, \ldots. K$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

« 上一篇：集成学习原理小结
» 下一篇：scikit-learn Adaboost类库使用小结

< Prev 1 2

评论列表

#51楼 2018-04-10 21:25 扶瑶直上

@ 刘建平Pinard
你好，"对于权重大的样本分类错误造成的损失函数更大"，请问这句话怎么得到的？我只看到分类错误的话权重会越来越大。

支持(0) 反对(0)

#52楼[楼主] 2018-04-11 14:29 刘建平Pinard

@ 扶瑶直上
根据上面讲到的损失函数可以看出：

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

当w'_{ki}较大时，误分类的损失函数会变得更大。

支持(0) 反对(0)

#53楼 2018-04-11 19:34 扶瑶直上

@ 刘建平Pinard
非常感谢您的回复！对于每一步的基本分类器，我想问下，是不是都得根据最小分类误差率来选择？

支持(0) 反对(0)

#54楼[楼主] 2018-04-12 17:17 刘建平Pinard

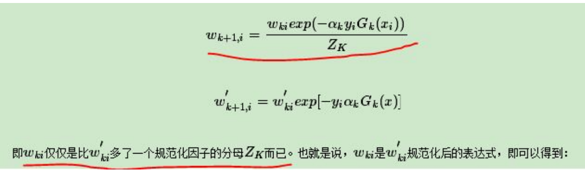
@ 扶瑶直上
你好，是的，从这个式子就可以看出：

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

支持(0) 反对(0)

#55楼 2018-04-16 09:57 用户名菜园子

@ 刘建平Pinard
楼主你好，关于样本权重还是有点疑问，希望能够得到你的解答



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

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

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：

归一化我明白，但是这里有个前提，是已经知道样本权重wk+1,i是这样计算，才会得出w_{ki}^{'}与w_{ki}仅相差一个归一化因子Z的结论吧？

支持(0) 反对(0)

#56楼[楼主] 2018-04-16 21:44 刘建平Pinard

@ 用户名白菜
你好，是的，在原文中第三节末尾有提到怎么求递推公式：

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

$$w'_{ki} = exp(-y_i f_{k-1}(x))$$

$$w'_{k+1,i} = exp(-y_i f_k(x))$$

由上面三个式子很容易就推导出了$w'_{k+1,i}$和w'_{ki}的关系。

支持(0) 反对(0)

#57楼 2018-04-17 00:13 用户名菜园子

@ 刘建平Pinard
感谢你的回复，这个递推公式理解的
有三个点的逻辑顺序没有理清楚
1，文中，你先令

w'_{k+1}=w_{k,i}exp(y_i*f_k-1(xi))
这个是文中定义的，很明确，一开始是没有任何意义的，对吧？
2，即文中说的
w'_{k+1}与w_{k+1}仅相差一个归一化因子
你回答说w_{k+1}是预先定义的，那请问样本权重为什么定义成
w_{k+1,i}=w_{k,i}exp(a_i*y_i*G_k(xi))/Zk
3，第三节末尾的递推公式
其实那个公式递推的是w'_{k+1}，因为知道w_{k+1}与w'_{k+1}=之间的关系才可以转化为w_{k+1}的递推公式的吧？
2里面如果w_{k+1}是预先定义了，那么也不需要通过3来得到递推公式了吧？

支持(0) 反对(0)

#58楼[楼主] 2018-04-17 23:26 刘建平Pinard

@ 用户名白菜
你好！你的理解没错。
1.按我的原文第二节， 这里的确一开始是没有意义的。在推导那一节才能知道原因。
2. 归一化是为了所有的可能取值的权重和为1.
3. 其实第二节是先给了结果，第三节才是推导。所以你的理解也没有错，只是我第二节直接给结果了，让你觉得逻辑不通。

不错这些都不重要，你理解就好。

支持(0) 反对(0)

#59楼 2018-04-20 16:00 榛子巧克力

你好，请问第三节，分类误差率的推导部分，ek的表达式是怎么从w`ki转换到wki的呢？即怎么从前向分步算法转换到adaboost中的wki呢？

支持(0) 反对(0)

#60楼[楼主] 2018-04-20 22:37 刘建平Pinard

@ 榛子巧克力
请参考41楼，42楼的回复。

支持(0) 反对(0)

#61楼 2018-04-25 11:48 xiaoyusmd

请教一下大佬，为什么adaboost的损失模型是指数损失模型呢？期待你的回答，谢谢！

支持(1) 反对(0)

#62楼[楼主] 2018-04-25 16:45 刘建平Pinard

@ xiaoyusmd
这个我回答不好。只能说这个算法选择的就是这个损失函数。

选择对数损失函数应该也是可以的，但是adaboost优化使用对数损失函数没有指数函数方便简洁，这是我能想到的一个较次要的原因。

支持(0) 反对(0)

#63楼 2018-04-26 09:02 xiaoyusmd

@ 刘建平Pinard
谢谢回答。

支持(0) 反对(0)

#64楼 2018-04-27 08:21 闻道暮东

感谢博主一系列博客，收益匪浅。
29-31楼有提到：
极小化指数损失函数等价于最小化分类误差
请问等价的原因是在adaboost中基分类器返回的结果是{-1,+1}吗？
如果$G_m(x)$返回的是一个连续型变量，似乎推导不出这个等价。

支持(0) 反对(0)

#65楼[楼主] 2018-04-27 22:38 刘建平Pinard

@ 闻道暮东
你好，是的，如果不是-1和1（比如0和1），这里的损失就乱套了。毕竟一个损失函数的基本要求是误分类的时候损失变大，正确分类的时候变小。

支持(0) 反对(0)

#66楼 2018-05-05 20:17 骑着鱼去飞

第五节：回归问题的提升，最后的公式，如果我的弱分类器无限多，那这个式子不是要永远地加下去？
二分类问题不存在类似地状况，因为即使无限地加下去，它加的数字都是有正有负，最终看符号就可以。
对于回归问题，如果我预测一个东西，因为预测的目标肯定是正的，所以我的弱回归模型预测出来，一般都是正的，一直加下去，肯定会加到正无穷。
这只是从结果分析它的合理性，我估计从原理上分析，也会有不合理的地方。

当然最有可能的还是我的理解有问题。
希望博主可以帮忙解答

支持(0) 反对(0)

#67楼[楼主] 2018-05-05 20:20 刘建平Pinard

@ 骑着鱼去飞
你好，实际应用中，我们的迭代次数是事先指定的，也就是弱学习器的个数是一定的。所以不会出现你永远加下去的情况。

支持(0) 反对(0)

#68楼 2018-05-06 10:14 骑着鱼去飞

@ 刘建平Pinard
感谢博主的回复，我昨天看了原著的论文，以及sklearn的源码
如果按以上两者为正确答案的话，对于回归问题，强学习器并不是简单的对弱学习器进行加权（跟分类问题毕竟不一样）而是按照某种方式对结果进行排序，取类似于中值的东西。
如果博主有时间，还是希望对这段重新斟酌一下。
对于回归问题的提升算法，大家关注的比较少，博主这里写得还是蛮清楚，属于对我启发比较大的，在这点上，无论如何还是要表示感谢。
x

支持(0) 反对(0)

#69楼[楼主] 2018-05-06 19:15 刘建平Pinard

@ 骑着鱼去飞
你好，的确是对加权的弱回归器取中位数。我改一下。

支持(0) 反对(0)

#70楼 2018-05-09 10:46 圆圆小金子

大神，您好。请教您一个小问题啊，关于损失函数求导那里,最终结果用的是log，可是e为底的指数，不应该是ln(e)=1吗？ log(e)并不为1。 所以最终
α=1/2*ln((1-ek)/ek)

不太明白这里，跟之前高数里面的表达不一样，是ml里面不太注重这个吗？
还是说最终效果都是一样的，所以直接用的log，而不是ln。

支持(0) 反对(0)

#71楼 2018-05-09 17:20 小熊_看看

@ 圆圆小金子
Python中，log默认是自然对数（底数是e），如果想用10 做底数，需要设置参数，底数可手动设置。

支持(0) 反对(0)

#72楼 2018-05-09 17:22 圆圆小金子

@ 小熊_看看
soga，在算法的理论推导的时候，是不是一般都写log了，就属于ML领域默认的规则吧，我看LR里面好像也都写log，我的思维偏向于高数里面的表达了。

支持(0) 反对(0)

#73楼 2018-05-15 08:06 puzzor_2014

@ 刘建平Pinard
您好，感谢大神的博客分享，膜拜下。我目前也卡在29楼提到的这个问题上，请问为什么极小化指数损失函数等价于最小化分类误差。我了解的不是很透彻，请问能直观得到吗还是有定理证明。这个问题在《统计学习方法》中也是没给证明。

支持(0) 反对(0)

#74楼 2018-05-15 15:17 圆圆小金子

请问第3节中，Gk(X)是怎么得到的，这里没看懂
α是通过求导计算的，通时用到了前面的符号ek，这个比较清楚

支持(0) 反对(0)

#75楼[楼主] 2018-05-15 16:09 刘建平Pinard

@ puzzor_2014
这个推导其实并不难。
对于通用的指数损失函数：

$$\ell_{exp}(H|D) = E_{x\sim D}[e^{-f(x)H(x)}]$$

对指数损失函数求偏导,我们可以得到：

$$\frac{\partial \ell_{exp}(H|D)}{\partial H(x)} = -e^{-H(x)}P(f(x)=1|x) + e^{H(x)}P(f(x)=-1|x)$$

令上式等于0,可以得到$H(x)$表达式：

$$H(x) = \frac{1}{2}\ ln\ \frac{P(f(x)=1|x)}{P(f(x)=-1|x)}$$

因此可以得到：

$$\begin{aligned} sign(H(x)) &= sign(\frac{1}{2}\ ln\ \frac{P(f(x)=1|x)}{P(f(x)=-1|x)}) & (1) \\ &= \begin{cases} 1, & P(f(x)=1|x) > P(f(x)=-1|x) \\ -1, & P(f(x)=1|x) < P(f(x)=-1|x) \end{cases} & (2) \\ &= \underset{y \in (-1,1)}{argmax}\ P(f(x)=y|x) & (3) \end{aligned}$$

这意味着$sign(H(x))$达到了贝叶斯最优错误率。若指数损失函数最小化，则分类错误率也将最小化；这就说明指数损失函数是分类任务原本0/1损失函数的一致的替代损失函数。

支持(1) 反对(0)

#76楼[楼主] 2018-05-15 16:35 刘建平Pinard

@ 圆圆小金子
参看29楼的回复，这个与你选择的弱学习器有关。

支持(0) 反对(0)

#77楼 2018-05-24 10:48 carolxuan

回归部分的样本权重更新公式：误差率越大的样本权重更新变小了？1-eki 趋于0

支持(0) 反对(0)

#78楼[楼主] 2018-05-24 23:02 刘建平Pinard

@ carolxuan
你好，请注意第二节末尾强学习器的表达式，不是直接α去乘的弱学习器。可能我的表述产生了一些误解。

支持(0) 反对(1)

#79楼 2018-05-25 10:43 carolxuan

学习器表达式没有问题，但是样本权重更新公式不是应该遵循分错误差越大的样本下一次迭代中权重更高吗，但是现在的样本权重公式是误差率大的样本反而权重变小了，不是吗？

支持(0) 反对(0)

#80楼[楼主] 2018-05-26 18:01 刘建平Pinard

@ carolxuan
你好，你可以理解为此时权重不是α,而是$ln\frac{1}{\alpha k}$

支持(0) 反对(0)

#81楼 2018-05-31 07:57 renminghuang

老师您好：
第3部分求解第m轮弱分类器Gm(x)有点疑惑。
Gk(x)=wki*exp(-yi α Gk(xi))，
推出Gk(x)=argmin wki*(I(yi不等于Gk(xi)))

但当yi与Gk(xi)不相等时，则yi*Gk(xi)=-1，此时exp(-yi α f(xi))=exp(α)
而当yi与Gk(xi)相等时，则yi*Gk(xi)=1,此时exp(-yi α f(xi))=exp(-α)

因为α>0，所以exp(-α)<exp(α)，应该是Gk(x)=argmin(I(yi=Gk(xi)))

没开通博客，不让上传图片，符号可能看着有点吃力，不好意思。

支持(0) 反对(0)

#82楼[楼主] 2018-05-31 22:48 刘建平Pinard

@ renminghuang
你好，这里要理解的是我们的目标函数$G_k(x)$是为了让误分类的样本的损失尽可能的小，对于正常分类的样本其损失可以忽略，因为$e^{-\alpha}$较小。所以必须是最小化误分类的样本的损失。

支持(0) 反对(0)

#83楼 2018-06-10 15:51 CuiPeng

您好，有两个疑惑
（1） 这里是误差损失为线性时的情况，如果我们用平方误差，则eki=(yi−Gk(xi))2E2k,如果我们用的是指数误差，则**eki=1−exp（−yi+Gk(xi))/Ek）**

指数误差的公式似乎有问题，误差e不应该在（0,1）之间么？−yi+Gk(xi) 没有加绝对值是什么含义？

（2）我们再来看看如何得到弱学习器权重系数α。这里有：
αk=ek/（1−ek)
学习器权重alpha 不应该跟误差 e成反比么？公式似乎是正比

望解惑，多谢！

支持(0) 反对(0)

#84楼[楼主] 2018-06-11 22:17 刘建平Pinard

@ CuiPeng
你好！
1）忘记加绝对值了，已经更正，感谢指出错误。
2） 这里写的有点误导，实际上这个α_k还不是最终的系数，你看第五节最后一个表达式就知道了。当时这个表达式的流程参考了原论文，但是和分类的意义稍有不同。

支持(0) 反对(0)

#85楼 2018-06-12 00:20 CuiPeng

@ 刘建平Pinard
明白了，多谢回复

支持(0) 反对(0)

#86楼 2018-06-14 18:48 robin_X

@ 刘建平Pinard

> 引用
> @AlbertSR
> 你好，注意到：
> wk+1,i=wkiexp(−αkyiGk(xi))ZK" role="presentation">wk+1,i=wkiexp(−αkyiGk(xi))ZKwk+1,i=wkiexp(−αkyiGk(xi))ZKw_{k+1,i} = \frac{w_{ki}exp(-\alpha_ky_iG_k(x_i))}{Z_K}
> wk+1,i′=wki′exp[−yiαkGk(x)]" role="presentation">w′k+1...

大神， 有个疑惑还望指教， 41楼这里的第一个式子不是我们想要推导的结果吗？为什么这里用作已知条件了？怎么理解。谢谢

支持(0) 反对(0)

#87楼[楼主] 2018-06-14 22:22 刘建平Pinard

@ robin_X
你好，这里只是为了表示w_{ki}和w'_{ki}之间的关系而已。并不是在做推导。
w_{ki}递推关系式是从w'_{ki}的递推关系式并考虑规范化得到的

支持(0) 反对(0)

#88楼 2018-06-15 10:46 robin_X

@ 刘建平Pinard
首先非常感谢您的回复， 如果这不是个已知条件的话， 我还是比较疑惑

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

这个公式是怎么计算的，我想我的疑惑可以和57#有点像，还望大神指教；还有一个问题，您第5节的“Adaboost回归问题的算法流程”和回归问题的提升树，也就是拟合残杀的那个模型是什么样的关系呢？

支持(0) 反对(0)

#89楼[楼主] 2018-06-17 12:05 刘建平Pinard

@ robin_X
你好，这个加权分类误差率是已知的定义。所以不需要推导。如果不加权的话，分类误差率和普通的分类算法就没有区别，也就是误分类的数量占比。

第5节的回归流程是没有限定误差的计算方法的。如果是使用平方误差，那么拟合的就是残差。

支持(0) 反对(0)

< Prev 1 2

刘新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。

最新IT新闻:
· 智利法院裁定银行需要为加密货币交易所开设银行账户
· 倪光南：投资芯片产业不要期待一两年就取得回报
· 比亚迪：比亚迪及子公司印章并未出借给李娟或遗失
· 小米即将被纳入"恒生指数"？错了
· 3D打印技术使移动工厂成为现实
» 更多新闻...

最新知识库文章:
· 危害程序员职业生涯的三大观念
· 断点单步跟踪是一种低效的调试方法
· 测试 | 让每一粒尘埃有的放矢
· 从Excel到微服务
· 如何提升你的能力？给年轻程序员的几条建议
» 更多知识库文章...