

联系我们

请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

原

机器学习教程 之 梯度提升方法：GBDT及其扩展模型XGBoost

置顶 2018年04月29日 16:06:59

Gradient boosting是一种广泛被用于回归、分类和排序任务的集成方法，于2001年被Friedman提出。该类算法通过以**上一轮基学习器的误差的负梯度**为训练目标训练本轮的基学习器，不断降低集成模型在训练集上的误差。基于Gradient Boosting算法的学习器被称为Gradient Boosting Machine (GBM)，如果说AdaBoost是boosting方法的集大成者。GBM几乎刷新了各个领域众多数据集的精度记录，有人认为GBM是性能最点激进，但通常各类数据竞赛赢家的策略里也确实都会有这类算法的方法或者思想。

由于集成学习方法在实际应用中出色的性能，我曾经写过几篇这方面的博文，关于集成学习及其boosting方法。

机器学习教程 之 Boosting 与 bagging：集成学习框架
人工智能里的数学修炼 | AdaBoost的数学原理：分布更新推导
机器学习教程 之 集成学习算法：深入剖析AdaBoost

还有一片关于 bagging 类随机森林的
机器学习教程 之 随机森林：算法及其特征选择原理

感兴趣的朋友可以了解一下，可以帮助你们更好的了解集成学习的整体情况。其实很早就打算着手写GBDT还有XGBoost，但是之前一直感觉还不太能驾驭，所以一直没有动手，这次借着课题组报告的机会花了一个星期把GBDT的大纲整理了一下做成了ppt，这篇文章是对ppt的解释和扩充，算是给自己的一个总结和交待。

ppt在这里，想要看精简版的可以了解一下
Gradient Boost Decision Tree (GBDT) 及其拓展模型

这篇文章主要分为三个部分：

- 第一部分 以GBDT的回归模型来详细讲解模型的原理和算法
- 第二部分 讲解GBDT的扩展实现：XGBoost
- 第三部分 这里会介绍一下xgboost库以及常用的GBDT类模型库的调参方法

一、GBDT的回归模型

GBDT可以用于回归、分类、排序等各种任务，这里以回归模型为例进行讲解是因为GBDT在回归问题上可解释性是最强的，分类及排序等模型只需要在回归原理上稍做修改即可。关于GBDT的理解主要有一下几个比较重要的点

1. GBDT的基分类器为CART树（分类回归树）
2. 一种boosting方法：Boosting Tree 提升树
3. 加性模型与前向分布算法
4. Gradient Boosting 梯度提升
5. GBDT采用的一些小技巧

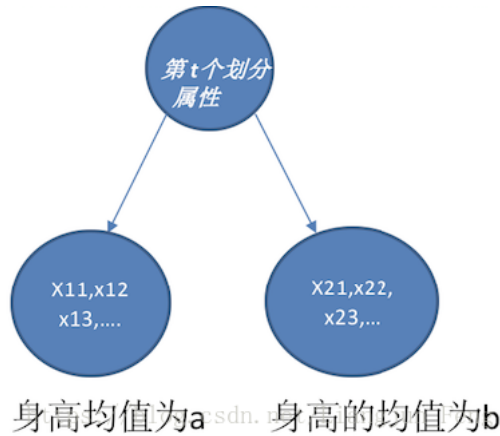
接下来我会从这五个方面，一步一步的把GBDT说清楚

1.1 GBDT的基分类器：CART树

CART树的全称是分类回归树，这是一种**二叉树**，可以处理分类与回归两种问题，是Breiman于1984年提出的一种树模型，后来他还基于这种方法提出了著名的随机森林。

CART树可以处理分类与回归两种问题，处理两类问题的策略不同，但分类树与回归树的生成框架是相同的，唯一不同的地方在于节点的属性选择标准。分类树采用的选择标准是**Gini系数**，认为Gini系数小的属性比较好，回归树采用**最小二乘**计算每个属性的划分误差，认为划分误差小的属性比较好

分类树与普通的决策树计算并没有什么不同，这里主要讲一下最小二乘回归树的具体计算方法，回归树主要有以下几点：



2.分枝时遍历所有的属性进行二叉划分, 挑选使平方误差最小的划分属性作为本节点的划分属性

3.遍历每个属性找到使loss最小的属性, 令其为当前节点的属性
第t个属性的平方误差计算公式为

$$loss_t = \sum_i (x_{1i} - a)^2 + \sum_j (x_{2j} - b)^2$$

4.如果属性有多种可能取值, 需要遍历所有可能的切分点, 找到使loss最小的切分点作为该属性的二叉划分点

5.对子集继续选择性的最优特征进行划分, 直到每个叶子节点仅有一个样本(这种情况是很少见或者难以实现的, 通常会设定一个允许叶子节点保留最少样本数的参数, 满足条件即可停止, 如果叶子节点有多个样本值, 以均值作为该叶子节点的预测值)

以上就是回归树相比于分类树需要注意的地方

- 1 补充: 为什么集成算法大多使用树类模型作为基学习器? 或者说为什么集成学习可以在树类模型上取得成功?
- 2
- 3 树模型具有很多优良的性质, 比如预测速度快、不需要进行特征标准化、可解释性强等优点,
- 4 但是单独使用决策树会有树节点过深导致的过拟合等问题
- 5
- 6 而集成学习的bagging和boosting两类主流算法:
- 7 **bagging**, 关注于提升分类器的泛化能力, 比如随机森林就通过对样本和特征进行采样训练
- 8 基分类器来增强其整体对样本变化的适应能力, 所以在随机森林里对于树的max_depth这个
- 9 参数一般不会加以限制, **bagging**本身就是一种避免过拟合的方法
- 10
- 11 **Boosting**, 关注于提升分类器在训练集上的精度, 这似乎是一种容易过拟合的方法, 但是
- 12 **Boosting**在设计时会有很多避免过拟合的技巧, 比如设置max_depth, 限制树的最大深度、
- 13 限制叶子节点的最少样本数量等方法避免过拟合, 这些限制可能会降低单个树的精度, 但boosting
- 14 可以将集成的精度问题交给其为精度提升而设计的框架去解决
- 15
- 16 所以不论是bagging还是boosting类的方法都可以做到单独的决策树无法做到的方面, 从
- 17 而实现性能的提升

1.2 Boosting Tree 提升树方法

简单来说, Boosting Tree方法通过以**上一轮基学习器的误差**为训练目标训练本轮的基学习器, 不断降低集成模型在训练集上的偏差实现高精度的集成

举个例子: 对于回归问题, 如果我们预测的标签值是20



麦克纳姆轮



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗨 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

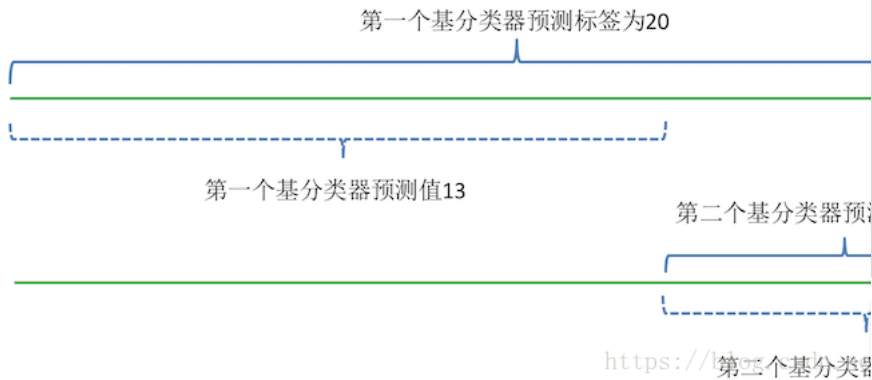
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心



按照如上的图例，第三次的预测标签为 $20 - 13 - 8 = -1$ ，不断重复这个过程，直到生成指定数量的基分类最后将所有预测结果

1.3 加性模型与前向分步算法

Boosting方法大多采用**加性模型**来组合弱学习器，即将集成的学习器表示为如下形式:

$$H(x) = \sum_{m=1}^M \beta_m * h_m(x; a_m)$$

这里 h_m 表示第 m 个基学习器， β_m 表示第 m 个基学习器的系数， a_m 表示第 m 个基学习器的参数， M 表示基学习器的个数

对于刚刚提到的Boosting Tree方法，基学习器的系数 β_m 为1，将加性模型简写为

$$H(x) = \sum_{m=1}^M h_m(x; a_m)$$

在给定损失函数 L 以及训练数据的条件下，学习加性模型 $H(x)$ 就是一个搜寻参数 α 使得**损失函数极小化**的过程，即

$$\min_{\alpha} \sum_{i=1}^N L(y_i; \sum_{m=1}^M h_m(x_i; a_m))$$

直接求解这个极小值是个很复杂的优化问题，我们采用**前向分步算法**来简化这一问题。前向分步算法本质上是一种**贪心算法**，它通过每一步只学习一个基函数及其系数，逐步逼近优化目标来简化复杂度

首先，我们来重写一下加性模型的表达形式，共生成 M 个基分类器，当前迭代次数为 m ，即在第 m 个循环中，Boosting Tree的加性模型可以写为

$$H_m(x) = H_{m-1}(x) + h_m(x; a_m)$$

前向分步算法只要求在第 m 轮时，优化当前轮次的基学习器的参数，使得损失函数最小以简化算法，即

$$\min_{a_m} \sum_{i=1}^N L(y_i; H_{m-1}(x_i) + h_m(x_i; a_m))$$

这里优化当前轮次的基学习器的参数，在经典的GBDT算法中，采取的是非常暴力的遍历的方式，即在每个节点划分的时候，**遍历所有属性和可能的节点划分数值**，在当前指定的损失函数下使损失达到最小，这点在之前提回归树的时候提到了



麦克纳姆轮



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

$$L(y, H_m) = (y - H_m)^2$$

我们有

$$\begin{aligned} &L(y, H_{m-1}(x) + h_m(x; a_m)) \\ &= [y - H_{m-1}(x) - h_m(x; a_m)]^2 \\ &= [r - h_m(x; a_m)]^2 \end{aligned}$$

这里的 $r = y - H_{m-1}(x)$ ，即上一轮基学习器训练完成以后， $m - 1$ 个基学习器与标签的残差，所以来说，每一步只需要拟合当前模型的残差就可以了。这里得强调一下，**Boosting Tree拟合的残差是在平方果，换了别了损失函数拟合的可能就不是残差了**

1.4 Gradient Boosting 梯度提升

Gradient Boosting 是GBDT的核心思想，它相比与 Boosting Tree 的主要改进在于：

Boosting Tree 对于每一轮基学习器，拟合的是**当前模型与标签值的残差**
GBDT 对于每一轮基学习器，拟合的是**当前模型与标签值的残差的负梯度**

这里会有一些问题：

- 1. 什么是残差的负梯度？
- 2. 为什么拟和残差的负梯度可以降低集成模型的损失，即为什么拟合残差的负梯度是可行的？
- 3. 即便拟合残差的负梯度是可行的，为什么不直接拟合残差？拟合残差的负梯度好在哪里？

这几个问题的难度是一次增加的，接下来我们一个一个来回答

1.4.1 什么是残差的负梯度？

对于 $i = 1, 2, 3 \dots, N$ 个样本，每一个样本的残差负梯度的计算公式为：

$$r_{mi} = - \frac{\partial L(y_i, H_{m-1}(x_i))}{H_{m-1}(x_i)}$$

看公式可能不太清楚，举个例子：
对于平方损失函数 $L(y, H(x)) = 1/2(y - H(x))^2$ ，其残差的负梯度为 $y - H(x)$
可以发现，**当损失为平方损失函数时，GBDT 和Boosting Tree的拟合对象是一致的**，但这只是一个特例，GBDT对于回归问题常用的损失函数，及负梯度如下

Setting	Loss Function	$-\partial L(y_i, f(x_i))/\partial f(x_i)$
Regression	$\frac{1}{2}[y_i - f(x_i)]^2$	$y_i - f(x_i)$
Regression	$ y_i - f(x_i) $	$\text{sign}[y_i - f(x_i)]$
Regression	Huber	$y_i - f(x_i)$ for $ y_i - f(x_i) \leq \delta_m$ $\delta_m \text{sign}[y_i - f(x_i)]$ for $ y_i - f(x_i) > \delta_m$ where $\delta_m = \alpha \text{th-quantile}\{ y_i - f(x_i) \}$

1.4.2 拟合残差的负梯度为什么是可行的？

拟合残差的负梯度是可以用泰勒展开来解释的，**泰勒一阶展开式**：

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×



麦克纳姆轮



联系我们



请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

$$f(x, \Delta x) \approx f(x) + f'(x) \Delta x$$

加性模型在第 m 轮可以写为 $H_m = H_{m-1} + h_m$

对 $L(y, H_{m-1} + h_m)$ 作一阶泰勒展开可以有：

$$L(y, H_{m-1} + h_m) \approx L(y, H_{m-1}) + L'_{m-1} h_m$$

其中

$$L'_{m-1} = \frac{\partial L(y, H_{m-1}(x))}{\partial H_{m-1}(x)}$$

即 $m-1$ 轮残差的梯度

加性模型在第 m 轮和 $m-1$ 轮的损失分别为：

$$\text{第}m\text{轮} \quad \text{loss1} = \sum_{i=1}^N (L(y_i, H_{m-1}(x_i)) + L'_{m-1} h_m(x_i; \alpha_m))$$

$$\text{第}m-1\text{轮} \quad \text{loss2} = \sum_{i=1}^N L(y_i, H_{m-1}(x_i))$$

我们希望每一轮性的基学习器之后损失都是下降的，即 $\text{loss}_1 < \text{loss}_2$ ，那么比较上面两个式子，我们可以发现，损失能否下降，关键在于 $\text{Key} = L'_{m-1} h_m(x_i; \alpha_m)$ 项的正负如何

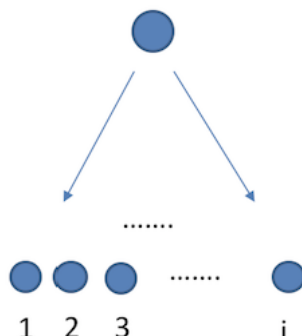
很简单的，我们只需要令 $h_m(x_i; \alpha_m) = -L'_{m-1}$ ，则

$$\text{Key} = L'_{m-1} h_m(x_i; \alpha_m) = -L'_{m-1}{}^2$$

该项便会恒为负，即 **损失会向着下降的方向移动**，这其实也是梯度下降的思想。这里令 $h_m(x_i; \alpha_m) = -L'_{m-1}$ 也就是令第 m 个学习器拟合当前模型残差的负梯度

$$\alpha_m = \min_{\alpha_m} \sum_{i=1}^N [-L'_{m-1} - h_m(x_i; \alpha_m)]^2$$

在之前的boosting Tree还有残差负梯度的推导中，我们已经说明了 $h_m(x_i; \alpha_m)$ 的拟合对象和参数 α_m 的搜索方法（遍历）。这里再具体说一下每一棵树对于残差负梯度的拟合



麦克纳姆轮



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

第 m 个回归树的叶子节点的值 β_m 都是一个向量, j 为第 m 个回归树的叶子节点个数, 需要注意的是, 我们每棵树拟合的每个叶子节点里往往不仅仅只有一个样本, 而这些样本又会有不同的标签值, 那么每个叶子节点的值是怎样确定的呢? 我法, 而GBDT没有采用这种方式, 而是采用了线性搜索一个参数来进行的优化。

$$H_m = H_{m-1} + \rho_{mj} \sum_{j=1}^J b_{mj} I(x \in R_{jm})$$

https://blog.csdn.net/Liangjun_Feng

即以每一个叶子节点为单元进行优化, 这里的 b_{mj} 表示是叶子节点的输出值, ρ_{mj} 表示的是以损失函数最低为目标进行线性训练完成之后, 其叶子节点的输出值, 是要经过一个 ρ_{mj} 的更新的, 对于一些特殊的情况Fredman在论文中给出了 ρ_{mj} 的估量, 关于这部分, 我会具体的在下一篇关于GBDT的博文中进行展开的讲解, 上面的式子可以写为

$$H_m = H_{m-1} + \sum_{j=1}^J \gamma_{mj} I(x \in R_{jm})$$

到目前为止我们已经讨论了GBDT的主要原理, 包括: 每一个基学习器的拟合目标是什么? 为什么对负梯度进行拟合是可行是怎样得到的?

1.4.3为什么GBDT要以残差的负梯度替代Boosting Tree的残差作为基学习器的学习目

这是问的最多的一个问题, 也是很难回答的一个问题。Boosting Tree 和 GBDT 两者的相同之处在于, 都是迭代回归树, 都是累加每棵树的结果作为最终结果, 每棵树都在学习前 $m-1$ 棵树尚存的不足, 从总体流程和输入输出上两者是没有区别的

两者的主要区别就在于每步迭代的时候是否 **使用残差的负梯度** 作为树的拟合对象, 前者不用残差的负梯度而是使用残差, 是全局最优值, 后者使用的是 局部最优方向 (负梯度) * 步长 (β), 即前者是每一步都试图让结果变成最好, 后者则每步试图让结果更好一点

Boosting Tree的最大问题在于, 它依赖残差进行优化, 损失函数一般固定为反应残差的均方差损失函数, 因此当均方差损失函数失效 (该损失函数对异常值敏感) 的时候, 换了其他一般的损失函数, 便很难得到优化的结果。同时, 因为损失函数的问题, 它也很难处理回归之外问题。而后者使用梯度下降的方法, 对于任意可以求导的损失函数它都可以处理

GBDT本质上是以梯度下降的办法简化了Boosting Tree对于损失函数的优化求解问题

1.5 GBDT 里用到的一些技巧

除了以上说明的梯度提升办法, GBDT还采用了很多使用技巧, 之所以说他们是技巧, 是因为这些方法在实际中取得成功, 却很难在理论上证明

1.5.1 Shrinkage 收缩

Shrinkage的思想认为, 每次走一小步逐渐逼近结果的效果, 要比每次迈一大步很快逼近结果的方式更容易得到精确值, 即它不完全信任每一棵残差树, 认为每棵树只学到了真理的一部分累加的时候只累加了一小部分多学几棵树来弥补不足。这个技巧类似于梯度下降里的学习率

数学方程对比: 之前: $H_m(x) = H_{m-1}(x) + \sum_{j=1}^J \gamma_{mj} I$

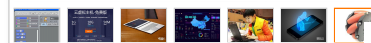
Shrinkage: $H_m(x) = H_{m-1}(x) + \eta * \sum_{j=1}^J \gamma_{mj} I$

https://blog.csdn.net/Liangjun_Feng

学习率 是一个人为设置的参数, 用来调整学习的快慢和精度, 学习率一般取0.001到0.01, 设置的越小, 需要训练的回归树越多。



麦克纳姆轮



联系我们



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

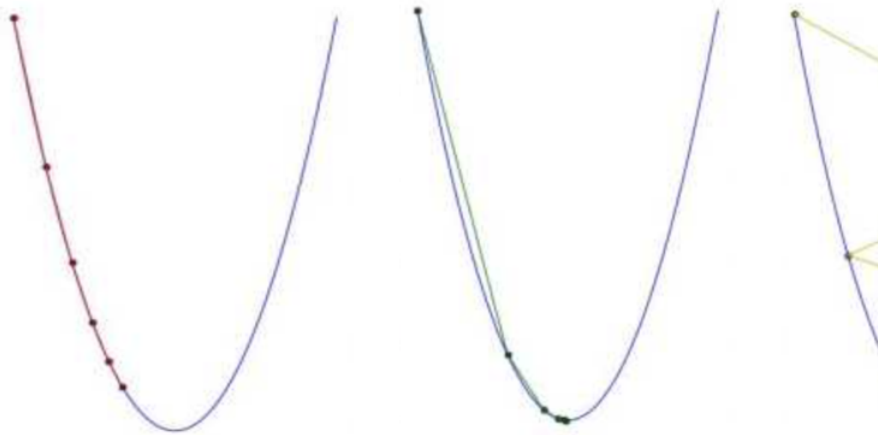
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心



梯度下降里的学习率，从左到右依次增



麦克纳姆轮



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗨 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

1.5.1 重复使用属性

在GBDT的每一棵回归树中，一个属性可以在多个节点用用到,更准确的说，这是CART树本身具有的性质，一个属性和划分点得来的，即便是同一个属性也有可能因为不同的划分点带来不同的划分效果

1.5.2 样本采样

GBDT学习了随机森林里的样本采样方法，每一棵树基于原始样本的一个子集进行训练达到防止过拟合和提升训练速度的目的

到这里为止，我们就讲完了GBDT回归模型的全部内容

二、GBDT的最强实现 Extreme Gradient Boosting (XGBoost)

XGBoost是华盛顿大学的陈天奇博士于2014年开源的一个GBDT库，[论文](#)发于2016年，它在算法层面和实现层面较以往的GBDT有很大改进，在实践中，精度有很大的提升同时，计算速度较传统的GBDT实现也快了近10个量级

XGBoost在算法层面相比于GBDT做出的改进主要有以下几点：

2.1 显式的将树模型的复杂度作为正则项加在优化目标里

数学模型的修改：

$$\begin{aligned} \text{GBDT:} \quad & Obj = \sum_{i=1}^n L(y_i, H_M(x_i)) \\ \text{XGBoost:} \quad & Obj = \sum_{i=1}^n L(y_i, H_M(x_i)) + \sum_{m=1}^M \Omega(h_m) \end{aligned}$$

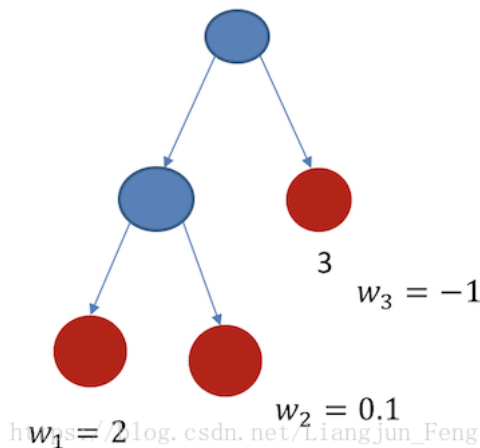
上式中 M 代表基分类器的个数， $\Omega(h_m)$ 代表第 m 棵树模型的复杂度

那么，XGBoost是具体以怎样方式来描述树模型的复杂度的？

首先，我们将一棵回归树的定义做一下细化,将一棵树拆分为结构部分 q 和叶子权重部分 w ：

$$h_m = w_q(x)$$

上式中 q 代表从输入到输出的映射结构， w 是一个向量，向量的元素是每个叶子节点的值。（ $x \in R^l, q: R^l \rightarrow T, w \in R^T$, 这里的 T 代表叶子节点的数量, 下图中的 T 为 3）



每一棵树的正则项 $\Omega(h_m)$ 由两部分组成：

$$\Omega(h_m) = \gamma T + \frac{1}{2} \lambda \sum_{i=1}^T w_i^2$$

2.2 GBDT用了泰勒展开的一阶导数信息，XGBoost将它扩展到了二阶信息

XGBoost的损失函数：

$$\begin{aligned} Obj &= \sum_{i=1}^n L(y_i, H_{m-1}(x_i) + h_m(x_i)) + \Omega(h_m) \\ &\approx \sum_{i=1}^n \left[L(y_i, H_{m-1}(x_i)) + L'_{m-1} h_m(x_i) + \frac{1}{2} L''_{m-1} h_m^2(x_i) \right] + \Omega(h_m) \end{aligned}$$

根据前向分布算法，在第m轮优化的目标就是

$$\begin{aligned} Obj_m &= \sum_{i=1}^n \left[L'_{m-1} h_m(x_i) + \frac{1}{2} L''_{m-1} h_m^2(x_i) \right] + \Omega(h_m) \\ &= \sum_{i=1}^n \left[L'_{m-1} h_m(x_i) + \frac{1}{2} L''_{m-1} h_m^2(x_i) \right] + \gamma T + \frac{1}{2} \lambda \sum_{i=1}^T w_i^2 \end{aligned}$$

将每棵树拆分成叶子节点，即将 h 用 w 表示，可将上式重写为

$$Obj_m = \sum_{j=1}^T \left[\left(\sum_{i \in I_j} L'_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} L''_i + \lambda \right) w_j^2 \right] + \gamma T$$

当树的结构 q 确定时，我们可以通过调整 w_j 即叶子节点的输出降低损失，对上式求导取0，可以得到优化值：

$$\begin{aligned} w_j &= - \frac{\sum_{i \in I_j} L'_i}{\sum_{i \in I_j} L''_i + \lambda} \\ Obj_m(q) &= - \frac{1}{2} \sum_{j=1}^T \frac{(\sum_{i \in I_j} L'_i)^2}{\sum_{i \in I_j} L''_i + \lambda} + \gamma T \end{aligned}$$

上面这个式子可以理解为对树的结构 q 的一个打分函数，损失越小，结构 q 越好，不过我们不太可能在计算时枚举所有的树结构，一般采用的是贪心算法。即每一次尝试去对已有的叶子加入一个分割。对于一个具体的分割方案，我们获得的增益可由下式计算



麦克纳姆轮



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗨 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

👍
4

📖
目录

🔖
收藏

💬
评论

🔄
快问

📧
微信

🐦
微博

👤
QQ

$$Gain = \frac{1}{2} \left[\frac{(\sum_{i \in I_a} L'_i)^2}{\sum_{i \in I_a} L''_i + \lambda} + \frac{(\sum_{i \in I_b} L'_i)^2}{\sum_{i \in I_b} L''_i + \lambda} - \frac{(\sum_{i \in I_j} L'_i)^2}{\sum_{i \in I_j} L''_i + \lambda} \right] - \gamma$$

https://blog.csdn.net/Liangjun_Feng

上式中 a, b 代表新分裂的子节点, j 代表母节点, 上式代表产生的两个新的子节点损失之和与原先一个母节点的损失的差

2.3 GBDT在XGBoost样本采样的基础上，增加了属性采样防止过拟合

GBDT模仿了随机森林的横竖采样，用以加速计算和防止过拟合

除了上述几点算法层面的改进，XGBoost相比于GBDT在具体实现上还有一些其他的设计用以加速运行

三、XGBoost参数讲解

XGBoost作为一个集成的分类器，有着大约20个可调参数，但其中主要的也就一手之数，理解这些参数的意义用它，下面是xgboost的主要参数

```
class XGBRegressor(XGBModel, sklearn.base.RegressorMixin)
```

Parameters
max_depth : int 表示基分类器的最大深度 一般取3~6
learning_rate : float 学习率，一般取0.001~0.1
n_estimators : int 训练的基分类器个数，默认100个
min_child_weight : int 叶子节点里样本预测值的最小和，用来防止过拟合，默认为1
subsample : float 样本采样比例
colsample_bytree : float 属性采样比例
reg_alpha : float (xgb's alpha) L1 正则项的系数
reg_lambda : float (xgb's lambda) L2正则项的系数

关于XGBoost和GBDT的调参方法，我还有另一篇博客讲述了具体的方法

机器学习教程 之 参数搜索：GridSearchCV 与 RandomizedSearchCV || 以阿里JCAI广告推荐数据集与XGBoostClassifier分类器为例

参考文献

[1] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System[J]. 2016:785-794.
[2] Library W P. Classification and regression tree[J].
[3] Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)[J]. Annals of Statistics, 2000, 28(2):337-374.
[4] Friedman J H. Greedy Function Approximation: A Gradient Boosting Machine[J]. Annals of Statistics, 2001, 29(5):1189-1232.



扫码向博主提问



liangjun_feng

集成学习，数据竞赛话题优秀回答者

擅长领域： 人工智能 机器学习 数据竞赛 boosting 集成学习

去开通我的Chat快问

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/Liangjun_Feng/article/details/80142724

文章标签：GBDT XGBoost 调参 原理详解 梯度提升树

个人分类：[机器学习教程](#)

所属专栏：[人工智能的数理模型与优化](#)



麦克纳姆轮



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录 注册 ×

https://blog.csdn.net/Liangjun_Feng/article/details/80142724?spm=5176.9876270.0.0.2777e44a11Dfh5

9/16

查看更多>>



麦克纳姆轮



想对作者说点什么？

机器学习笔记（七）Boost算法（GBDT,AdaBoost，XGBoost）原理及实践

Boost（提升）算法介绍，包括GBDT，AdaBoost和XGBoost的原理及Python中的实践。

sinat_22594309 2017-03-09 12:09:49 阅读数：13351

R | Yandex的梯度提升CatBoost 算法（官方述：超越XGBoost/lightGBM/h2o）

俄罗斯搜索巨头 Yandex 昨日宣布开源 CatBoost，这是一种支持类别特征，基于梯度提升决策树的机器学习方法。CatBoost是由Yandex的机器学习工程师开发的，是 Mat...

sinat_26917383 2017-07-21 11:18:28 阅读数：5426

耳鸣怎么处理？告诉你一个非常不错的小窍门

纳橙·顶新

R+python | XGBoost极端梯度上升以及forecastxgb（预测）+xgboost（回归）双案例解读

一、XGBoost来历 xgboost的全称是eXtreme Gradient Boosting。正如其名，它是Gradient Boosting Machine的一个c++实现，作者为正在华盛顿大...

sinat_26917383 2016-09-22 19:17:11 阅读数：16132

提升方法：GBDT、XGBOOST、AdaBoost

提升 (boosting) 方法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，提高分类器性能。 Table...

hit0803107 2017-05-04 16:09:44 阅读数：2221

王小草【机器学习】笔记--提升之XGBoost工具的应用

笔记整理时间：2016年12月29日 整理者：王小草 欢迎关注：王小草的FM喜马拉雅主播频道：搜索账号名“好吧我真的叫王草”王小草的个人微信公众号：bigdataML 王小草的CSDN博...

sinat_33761963 2016-12-29 16:44:10 阅读数：2264

机器学习提升算法之Adaboost、GB、GBDT与XGBoost算法

一、Gradient boosting(GB) 梯度提升 提升是一个机器学习技术，可以用于回归和分类问题，它每一步产生一个弱预测模型（如决策树），并加权累加到总模型中。如果每一步的弱预测模型生成...

weiyongle1996 2017-09-09 22:11:11 阅读数：1656

一个实用的闭经调节方法，月经调回来。

精益·顶新

Xgboost推导及分析

Author: DivinerShi Xgboost其实就是gbdt的一个改进版本，但是因为效果好，工程建设完善，所以经常和传统的gbdt分开讲。Boost思想的话是每一步产生一个弱预测模型（如...

sxf1061926959 2017-10-21 14:51:03 阅读数：1123

简单易学的机器学习算法——梯度提升决策树GBDT

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

 google19890102 2017-02-10 16:55:26 阅读数：37406

DS&ML_分类算法笔记之随机森林、**梯度提升树**、**XGBoost模型**

对于以决策树为基础的**模型**的相关知识点的总结与整理 **XGBoost**、随机森林RF、**梯度提升树GBDT**&GBRT 简单描述一下这...
对于以.....

 S_gy_Zetrov 2018-04-02 09:10:17 阅读数：262

梯度提升树GBDT原理

1.**模型提升方法**实际采用加法**模型**（即基函数的线性组合）与前向分布算法。以决策树为基函数的**提升方法**称为**提升树**（boosting tree）
二叉分类树，对回归问题决策树是二叉决策树。提...

 a819825294 2016-04-19 11:19:16 阅读数：32578

【机器学习】**GBDT梯度提升**算法调参总结II

对于**GBDT**的参数调整，上一篇已经通过实例讲明，不过调整参数确实重要，继续总结一下通用套路。...

 brucewong0516 2017-12-03 11:36:49 阅读数：317

Spark中组件Mllib的学习39之梯度提升树**（GBT）用于分类***

更多代码请见：https://github.com/xubo245/SparkLearning Spark中组件Mllib的学习之分类篇 1解释（1）**GBDT**基本概念月...

 bob601450868 2016-05-25 16:33:45 阅读数：2243

笔记 | 决策树族——**梯度提升树(GBDT)**

本笔记来源于CDA DSC，L2-R语言课程所学进行的总结。 一、介绍：**梯度提升树**（Gradient Boost Decision Tree）Boosting算法和树**模型**的结合。按次序建...
立...

 sinat_26917383 2016-06-08 20:11:14 阅读数：3603

买科沃斯扫地机器人抽免单

618狂欢科沃斯直降600

GBDT梯度提升树算法原理小结（二）

一、DT：回归树 Regression Decision Tree 提起决策树（DT, Decision Tree）绝大部分人首先想到的就是C4.5分类决策树。但如果一开始就把**GBDT**中的树...

 xiaocong1990 2017-05-16 21:48:41 阅读数：621

GBDT 梯度提升决策树的简单推导

GB, **梯度提升**，通过进行M次迭代，每次迭代产生一个回归树**模型**，我们需要让每次迭代生成的**模型**对训练集的损失函数最小，而如何让损失函数越来越小呢？
我们采用**梯度下降的方法**，在每次迭代时通过向损失函数的负梯...

 shenxiaoming77 2017-03-14 17:31:53 阅读数：3405

关于树的几个ensemble模型的比较（GBDT、xgBoost、lightGBM、RF）

决策树的Boosting**方法**比较 原始的Boost算法是在算法开始的时候，为每一个样本赋上一个权重值，初始的时候，大家都是一样重要的。在每一步训练中得到的**模型**，会使得数据点的估计有对有错，我们就在每一...

 xwd18280820053 2017-03-31 17:22:23 阅读数：6199

GBDT梯度提升树原理剖析

GBDT梯度提升树原理剖析在前面的文章中，我写了ID3及CART决策树的思路，但我们只是将其用来做分类。注意到CART的名字：Classification And Regressi...
on Tree，当我...

 chai_zheng 2017-10-18 11:15:45 阅读数：302

提升树GBDT详解

提升树GBDT 详解 For **Xgboost**:在前几篇笔记中以及详细介绍了决策树及**提升方法**的相关原理知识，本文是**提升树**与**梯度提升方法**的学习笔记，同时阅读了网...
络上的一些关于**GBDT**的博文，加强理解。为了...



麦克纳姆轮



联系我们



请扫描二维码联系客服
✉webmaster@csdn.net
☎400-660-0108
👤QQ客服 🗨客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心



免费云主机试用一年

可以免费试用30天的移动云vps

百度广告

XGBoost简介---相关概念、原理

XGBoost是2014年2月诞生的专注于**梯度提升**算法的**机器学习**函数库，此函数库因其优良的学习效果以及高效的训练速度在Kaggle竞赛中获胜的29个算法中...

u011630575 2017-11-21 09:25:59 阅读数：291

提升方法之AdaBoost、提升树 (GBDT)

引言**提升方法**是一种常用的学习**方法**（确切来说是一种基于统计的学习**方法**），并且广泛有效，基本思想是：不需针对学习便，后文中以分类为例）直接学习出一个**模型**，而是先学习出一个**模型**，...

u013269505 2017-08-11 16:00:25 阅读数：1115

GBDT梯度提升树算法原理小结（三）

首先我们回顾一下Gradient Boosting 的本质，就是训练出，使损失函数最小，即 其求解步骤如下： 所以，我们首先得定义下来我们针对不同场景，...

xiaocong1990 2017-05-16 21:53:42 阅读数：693

机器学习：GBDT

Boosting系列算法里主要的算法有AdaBoost算法和**提升树**系列算法，而**提升树**系列算法里广泛应用的算法是**GBDT**(Gradient Boosted Decision Tree/Gradient Boosting Decision Tree)...

u014120554 2017-01-04 10:42:14 阅读数：646

【机器学习】GBDT梯度下降提升算法及参数寻优实例

GBDT的思想可以用一个通俗的例子解释，假如有个个人30岁，我们首先用20岁去拟合，发现损失有10岁，这时我们用6岁去拟合剩下的损失，发现差距还有4岁，第三轮我们用3岁拟合剩下的差距，差距就只有一岁了（...

brucewong0516 2017-12-01 01:27:49 阅读数：1268

50万码农评论：英语对于程序员有多重要！

不背单词和语法，老司机教你一个数学公式秒懂天下英语

【ML笔记】梯度提升决策树 (GBDT) 和随机森林 (RF) 的异同

梯度提升决策树 (GBDT) 和随机森林 (RF) 的异同

roguesir 2017-07-27 10:29:56 阅读数：3224

机器学习boosting算法——梯度提升树(GBDT)

1 回归树算法分析 Adaboost算法是**提升树**的特殊情况，主要用于分类问题，对于回归问题怎么处理呢？？已知一个训练数据集 $T=\{(x_1,y_1),(x_2,y_2),...,(x_N,y_N)\}$...

ice_actor 2018-01-19 15:20:58 阅读数：167

GBDT：梯度提升决策树

GBDT：梯度提升决策树 字数2887 阅读1208 评论0 喜欢2 综述 GBDT(Gradient Boosting Decision Tree) 又叫 MART (Multipl...

jbctcx 2016-08-10 08:42:41 阅读数：4458

机器学习时代的三大神器:GBDT,XGBOOST和LightGBM

本文主要简要的比较了常用的boosting算法的一些区别，从AdaBoost到LightGBM,包括AdaBoost,GBDT,XGBoost,LightGBM四个**模型**的简单介绍，一步一步从原理到优化...

bbbeoy 2018-03-17 12:50:31 阅读数：186



麦克纳姆轮



联系我们



请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心



3.随机森林 随机森林是一个包含多个决策树的分类器，构建过程如下：1）决策树相当于一个大师，通过自己在数据集中学习但是俗话说得好，一个诸葛亮，玩不过三个臭皮匠。随...

 zhixiongzhao

2017-05-18 19:22:09

阅读数：5690

python是干嘛的

学Python后到底能干什么

百度广告

[面试笔记] 决策树、随机森林、GBDT、XGBoost

本文主要分享了 决策树、随机森林、GBDT、XGBoost 四种模型的原理决策树

 LeYOUNGER

2017-12-11 15:19:03

阅读数：736

机器学习（四）--- 从gdbt到xgboost

gdbt（ 又称Gradient Boosted Decision Tree/Grdient Boosted Regression Tree ），是一种迭代的决策树算法，该算法由多...

 china1000

2016-04-09 19:34:15

阅读数：23939

梯度提升回归树 GBDT java

```
/** * 梯度提升回归树 简单实现 * @author ysh 1208706282 * */ public class Gbdt { static List m...
```

 ysh126

2016-11-10 19:09:41

阅读数：1680

为什么XGBoost在机器学习竞赛中表现如此卓越？

挪威科技大学 Didrik Nielsen 的硕士论文《使用 XGBoost 的树提升：为什么 XGBoost 能赢得「每一场」机器学习竞赛？（ Tree Boosting With XGBoost -...

 Uwr44UOuQcNsUQb60zk2

2017-11-10 06:43:17

阅读数：353

DART booster

DART booster的介绍，本文通过翻译XGBoost官方文档以及理解DART booster原文得来，帮助大家理解DART booster的原理。...

 Yongchun_Zhu

2017-12-07 20:35:57

阅读数：403

科沃斯升级规划快速建图

618狂欢科沃斯直降600

提升树和GBDT

本文首先介绍提升树的模型，算法和梯度提升方法。然后介绍GBDT模型的形式和损失函数。...

 u012151283

2017-08-27 20:10:15

阅读数：263

RF,GBDT,xgboost调参方法整理

1、RF调参方法参考： <http://wakemeup.space/?p=187> <http://blog.csdn.net/y0367/article/details/51501780> [http:...](http://...)

 u014465639

2017-07-04 18:41:19

阅读数：1787

机器学习算法中GBDT和XGBOOST

机器学习算法中GBDT和XGBOOST

 jdbc

2017-01-18 15:29:19

阅读数：1718

统计学习方法--提升树模型（ Boosting Tree ）与梯度提升树（ GBDT ）

原文链接 <http://www.cnblogs.com/daguankele/p/6557328.html> 1、主要内容 介绍提升树模型以及梯度提升树的算法流程 ...

 juzexia

2018-01-04 18:18:25

阅读数：129

机器学习-->集成学习-->Xgboost详解

本篇博客将详细介绍集成学习里面的 一个非常重要的模型 xgboost。Boosting / 提升 / 提升树 是机器学习技术 可以用不同的输入数据 训练 多个分类器...



麦克纳姆轮

联系我们



请扫描二维码联系客服
✉webmaster@csdn.net
☎400-660-0108
👤QQ客服 🗨客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心



招聘滴滴专车商务司机100名

滴滴招聘

百度广告

机器学习（十七）——决策树, 推荐系统进阶

机器学习（十七）——决策树, 推荐系统进阶

antkillerfarm 2017-02-27 13:12:31 阅读数：4089

机器学习高频面试题之---简述GBDT与XGBoost的区别

最近校招面试被问到了这个问题，之前也在搜集了一些资料，在此整理一下，贴在这里。一、基本概念GBDT（又称Gradient Boosted Decision Tree）

jackmcgradylee 2017-09-01 21:17:53 阅读数：1418

GBDT（梯度提升决策树）算法（详细版）

转载地址：https://mp.weixin.qq.com/s?__biz=MzIzNDM2OTMzOQ==&mid=2247485043&idx=1&type=3

program_developer 2018-02-28 19:54:04 阅读数：210

Gradient Boosting Decision Tree (GBDT)-梯度提升决策树

Gradient Boosting Decision Tree (GBDT)-梯度提升决策树GBDT是集成学习的又一代表作，以boosting为基础的分部加法

fengziyihang 2018-03-22 13:50:02 阅读数：74

机器学习总结(四)——随机森林与GBDT(梯度提升决策树)

1. 随机森林简介随机森林采用的是bagging的思想，bagging即：通过在训练样本集中进行有放回的采样得到多个采样集，基于每个采样集训练出一个基学习器，再将基学习器结合起来共同实现分类或者回归。...

manong_wxd 2017-12-07 20:51:07 阅读数：412

科沃斯机器人建图规划扫的快

618狂欢科沃斯直降600



GBDT和Xgboost模型对比总结

一、GBDT有哪些参数,如何确定树的深度,学习率怎样确定。答：本题答案引自<http://www.07net01.com/2016/12/1742073.html> 在sk-learn中，Gr...

panda_zjd 2017-05-10 19:55:10 阅读数：4384

【R机器学习笔记】梯度提升回归树——gbm包

gbm包gbm包是梯度提升回归树 (GBRT) 在R 中的实现。GBRT，全称为Gradient Boosting Regression Tree, 有时也称为GBDT。wiki中对GBRT的定义 ...

langyichao1 2017-04-14 20:28:22 阅读数：3833

Sklearn-GBDT(GradientBoostingDecisionTree)梯度提升树

GBDT类库概述 GBDT有很多简称，有GBT (Gradient Boosting Tree)，GTB (Gradient Tree Boosting)，GBRT (Gradient Boosted Regression Tree)...

CherDW 2017-02-11 14:10:38 阅读数：9553

机器学习基础：集成学习方法应用实验（RF、GBDT）

本文回顾了集成学习的基础知识，并以RF、GBDT为典型算法进行了实验。

Snoopy_Yuan 2017-07-07 21:59:56 阅读数：2058

xgboost/gbdt在调参时为什么树的深度很少就能达到很高的精度？

作者：于菲 链接：<https://www.zhihu.com/question/45487317/answer/99153174> 来源：知乎 著作权归作者所有。商业转载请联系作者获得授权，非商业...

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

瑞士CR手表，今日一折，仅售1280元！

CR瑞士男士手表，官方活动火热开启，一折仅售1280元！支持开箱验货！

百度广告

机器学习算法-梯度树提升GTB (GBRT)

Introduction决策树这种算法有着很多良好的特性，比如说训练时间复杂度较低，预测的过程比较快速，模型容易展示（容易来）等。但是同时，单决策树又有一些不好的地方，比如...

 Dream_angel_Z 2015-08-29 11:11:35 阅读数：18121

机器学习方法篇(9)-----梯度提升决策树GBDT

● 每周一言生命在于运动，无论脑力还是体力。导语前面第3、4两节介绍了决策树，由于决策树简单有效，可解释性强，因学习算法，其中最为知名的就是梯度提升决策树GBDT (Gra...

 cherrylvlei 2017-08-20 22:16:45 阅读数：664

没有更多推荐了，[返回首页](#)





麦克纳姆轮



联系我们




请扫描二维码联系客服

✉webmaster@csdn.net

☎400-660-0108

👤QQ客服 💬客服论坛

关于 招聘 广告服务  百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心



原创	粉丝	喜欢	评论
48	59	36	20

等级： 博客 3 访问：4万+

积分：944 排名：5万+

勋章： 



VR跑步机



最新文章

- MySQL常用命令大全
- 机器学习教程 之 随机森林： 算法及其特征选择原理
- 以最简洁的方式打包发布你自己的pip项目
- 阿里妈妈广告预测线上0.082+baseline，特征部分
- 机器学习教程 之 参数搜索：GridSearchCV 与 RandomizedSearchCV || 以阿里IJCAI广告推荐数据集与XGBoostClassifier分类器为例

博主专栏

 人工智能的数学建模与优化

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！