梯度提升树

**0.Adaboost**

Adaboost二分类问题算法流程

输入：样本集，输出为{-1，+1}，基分类器迭代次数K

输出：最终的强分类器

1. 初始化样本权重
2. 对于 k =1 ,2, … K
3. 使用具有权重的样本集来训练，得到基分类器
4. 计算的分类误差率

1. 计算基分类器的权重

1. 更新样本集的权重分布

这里是规范化因子

1. 构建最终分类器：

**1.提升树**

提升方法实际采用加法模型（即基函数的线性组合）与前向分步算法。以决策树为基函数的提升方法称为提升树（boosting tree）。

这里讲的所谓的提升树模型以最简单的数学逻辑体现了一系列Boosting tree算法的核心思想。它与adaboost最为相近，只不过区别在于，这里的算法逻辑在每一次迭代过程中不同样本点的权重是不变的。而对于adaboost，在每一步预测错误的数据点的权重是有所增强的。

对于回归问题，提升树使用以下前向分步算法：

1. 初始化
2. 第m步的模型为

其中为当前模型，通过经验风险极小化确定下一棵决策树的参数

当采用平方误差损失函数时，

其损失变为

这里， 是当前模型拟合数据的残差（residual）。

所以，对回归问题的提升树来说，只需拟合当前模型的残差。

1. 最终的强学习器为

**2.梯度提升树**

当损失函数是平方损失和指数损失函数时，每一步优化是很简单的。但是对于其他一系列损失函数而言，往往每一步优化并不那么容易。针对这一问题，Friedman提出了梯度提升（gradient boosting）算法。该算法利用损失函数的负梯度在当面模型的值

作为残差的近似值，拟合一个回归树。

算法流程

输入：训练数据集

输出：回归树

1. 初始化

(2) 对 *m = 1, 2, …, M*

(a)对 *i= 1,2,…,N,* 计算

(b)对拟合一个回归树，得到第m棵树的叶节点区域

(c)对 *j =1,2,.., J,* 计算

(d)更新

(3)得到回归树

() =

**例子**

给定如下表所示的训练数据，选择MSE作为建树的分裂准则，选择MSE作为损失函数，树的深度设置为1，学习率为0.1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | 5.56 | 5.7 | 5.91 | 6.4 | 6.8 | 7.05 | 8.9 | 8.7 | 9.0 | 9.05 |

1. 初始化，当损失函数为MSE时，

为样本真实值的平均值

1. 拟合第一棵树（m=1）

（a）计算负梯度值：

)

具体结果如下表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
|  | -1.747 | -1.607 | -1.397 | -0.907 | -0.507 |
|  | 6 | 7 | 8 | 9 | 10 |
|  | -0.257 | -1.593 | 1.393 | 1.693 | 1.743 |

以为目标值进行拟合。

（b）树的分裂:

对于连续值特征，穷尽每个值*v*，把每个值*v*作为一个分裂点（<= *v*和 > *v*）,然后计算两个分支的

选择最小的的分裂点*v*

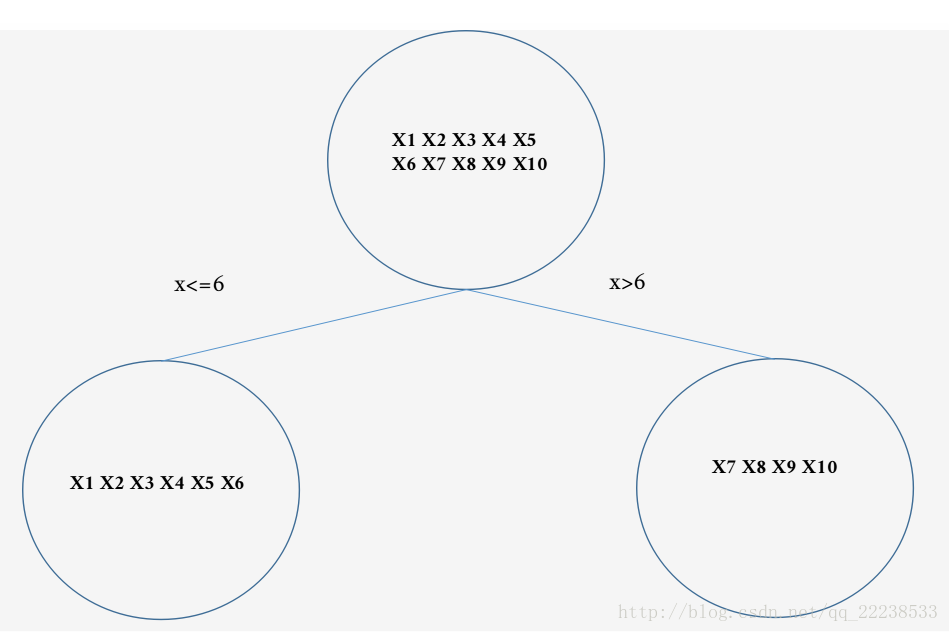
对于类别型特征，通过=和来划分

当选择1作为分裂点时，

当选择2作为分裂点时，

依次，穷尽所有取值。

可以得到当选择6作为分裂点时最小



至此，我们完成了第一棵树的拟合，得到和

为,为,

(c)更新的值，

比如更新其中一个样本的值：

=7.307-0.1\*1.0703=7.20

至此，一轮迭代（第一棵树拟合）完成，下面开始第二轮迭代。

拟合第二棵树（m=2）

计算,例如计算)

=5.56-7.20=-1.64

可得到下表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
|  | -1.64 | -1.50 | -1.29 | -0.80 | -0.40 |
|  | 6 | 7 | 8 | 9 | 10 |
|  | -0.15 | 1.43 | 1.23 | 1.53 | 1.58 |

在第二棵树中，拟合的是新的梯度值。下面的过程就是建树计算叶子节点的值、叶子节点的区间更新

最后得到两个叶子节点值分别为：

更新的值，

**二分类任务的梯度提升树**

损失函数为logloss

其中，

带入

=

=

+

=

因此，负梯度为

初始化

证明：

1. z为我们要计算的常量：
2. GBDT分类器的函数表达式:

3.二分类问题的损失函数：

4．对式3的p求导，得：

5.对2式的z求导，得

6.根据式4和式5，得

7．令式7等于0

8.式2带入式7得

叶子节点的值为