

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

决策树算法原理(下)

在决策树算法原理(上)这篇里，我们讲到了决策树里ID3算法，和ID3算法的改进版C4.5算法。对于C4.5算法，我们也提到了它的不足，比如模型是用较为复杂的熵来度量，使用了相对较为复杂的多叉树，只能处理分类不能处理回归等。对于这些问题，CART算法大部分做了改进。CART算法也就是我们下面的重点了。由于CART算法可以做回归，也可以做分类，我们分别加以介绍，先从CART分类树算法开始，重点比较和C4.5算法的不同点。接着介绍CART回归树算法，重点介绍和CART分类树的不同点。然后我们讨论CART树的建树算法和剪枝算法，最后总结决策树算法的优缺点。

1. CART分类树算法的最优特征选择方法

我们知道，在ID3算法中我们使用了信息增益来选择特征，信息增益大的优先选择。在C4.5算法中，采用了信息增益比来选择特征，以减少信息增益容易选择特征值多的特征的问题。但是无论是ID3还是C4.5,都是基于信息论的熵模型的，这里面会涉及大量的对数运算。能不能简化模型同时也不至于完全丢失熵模型的优点呢？有！CART分类树算法使用基尼系数来代替信息增益比，基尼系数代表了模型的不纯度，基尼系数越小，则不纯度越低，特征越好。这和信息增益(比)是相反的。

具体的，在分类问题中，假设有K个类别，第k个类别的概率为 $p_k$ ，则基尼系数的表达式为：

$$Gini(p) = \sum_{k=1}^K p_k(1 - p_k) = 1 - \sum_{k=1}^K p_k^2$$

如果是二类分类问题，计算就更加简单了，如果属于第一个样本输出的概率是p，则基尼系数的表达式为：

$$Gini(p) = 2p(1 - p)$$

对于个给定的样本D,假设有K个类别, 第k个类别的数量为 $C_k$ ,则样本D的基尼系数表达式为：

公告

★珠江追梦，饮岭南茶，恋鄂北家★  
昵称：刘建平Pinard  
园龄：1年5个月  
粉丝：1176  
关注：13  
+加关注

< 2018年4月 >						
日	一	二	三	四	五	六
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

常用链接

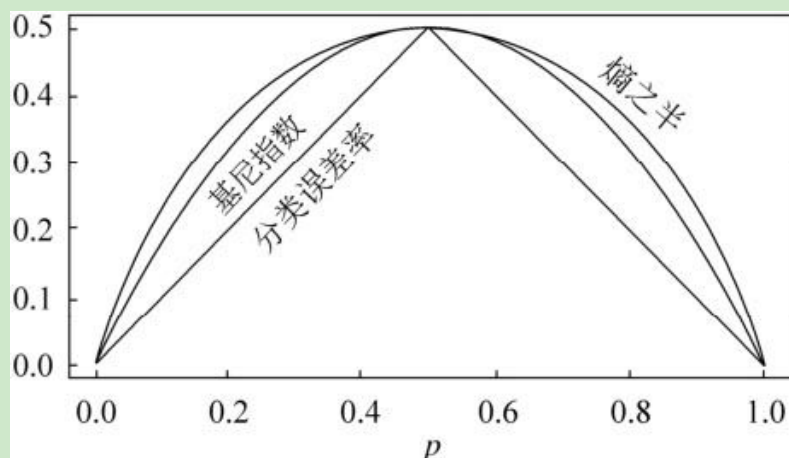
我的随笔  
我的评论  
我的参与

$$Gini(D) = 1 - \sum_{k=1}^K \left( \frac{|C_k|}{|D|} \right)^2$$

特别的，对于样本D,如果根据特征A的某个值a,把D分成D1和D2两部分，则在特征A的条件下，D的基尼系数表达式为：

$$Gini(D, A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

大家可以比较下基尼系数表达式和熵模型的表达式，二次运算是不是比对数简单很多？尤其是二类分类的计算，更加简单。但是简单归简单，和熵模型的度量方式比，基尼系数对应的误差有多大呢？对于二类分类，基尼系数和熵之半的曲线如下：



从上图可以看出，基尼系数和熵之半的曲线非常接近，仅仅在45度角附近误差稍大。因此，基尼系数可以做为熵模型的一个近似替代。而CART分类树算法就是使用的基尼系数来选择决策树的特征。同时，为了进一步简化，CART分类树算法每次仅仅对某个特征的值进行二分，而不是多分，这样CART分类树算法建立起来的是二叉树，而不是多叉树。这样一可以进一步简化基尼系数的计算，二可以建立一个更加优雅的二叉树模型。

## 2. CART分类树算法对于连续特征和离散特征处理的改进

对于CART分类树连续值的处理问题，其思想和C4.5是相同的，都是将连续的特征离散化。唯一的区别在于在选择划分点时的度量方式不同，C4.5使用的是信息增益比，则CART分类树使用的是基尼系数。

最新评论

我的标签

### 随笔分类(101)

- 0040. 数学统计学(4)
- 0081. 机器学习(62)
- 0082. 深度学习(10)
- 0083. 自然语言处理(23)
- 0121. 大数据挖掘(1)
- 0122. 大数据平台(1)
- 0123. 大数据可视化

### 随笔档案(101)

- 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

具体的思路如下，比如m个样本的连续特征A有m个，从小到大排列为 $a_1, a_2, \dots, a_m$ ，则CART算法取相邻两样本值的中位数，一共取得m-1个划分点，其中第i个划分点 $T_i$ 表示为： $T_i = \frac{a_i + a_{i+1}}{2}$ 。对于这m-1个点，分别计算以该点作为二元分类点时的基尼系数。选择基尼系数最小的点作为该连续特征的二元离散分类点。比如取到的基尼系数最小的点为 $a_t$ ，则小于 $a_t$ 的值为类别1，大于 $a_t$ 的值为类别2，这样我们就做到了连续特征的离散化。要注意的是，与离散属性不同的是，如果当前节点为连续属性，则该属性后面还可以参与子节点的产生选择过程。

对于CART分类树离散值的处理问题，采用的思路是不停的二分离散特征。

回忆下ID3或者C4.5，如果某个特征A被选取建立决策树节点，如果它有A1,A2,A3三种类别，我们会在决策树上一下建立一个三叉的节点。这样导致决策树是多叉树。但是CART分类树使用的方法不同，他采用的是不停的二分，还是这个例子，CART分类树会考虑把A分成{A1}和{A2, A3}，{A2}和{A1, A3}，{A3}和{A1, A2}三种情况，找到基尼系数最小的组合，比如{A2}和{A1, A3}，然后建立二叉树节点，一个节点是A2对应的样本，另一个节点是{A1,A3}对应的节点。同时，由于这次没有把特征A的取值完全分开，后面我们还有机会在子节点继续选择到特征A来划分A1和A3。这和ID3或者C4.5不同，在ID3或者C4.5的一棵子树中，离散特征只会参与一次节点的建立。

## 3. CART分类树建立算法的具体流程

上面介绍了CART算法的一些和C4.5不同之处，下面我们看看CART分类树建立算法的具体流程，之所以加上了建立，是因为CART树算法还有独立的剪枝算法这一块，这块我们在第5节讲。

算法输入是训练集D，基尼系数的阈值，样本个数阈值。

输出是决策树T。

我们的算法从根节点开始，用训练集递归的建立CART树。

1) 对于当前节点的数据集为D，如果样本个数小于阈值或者没有特征，则返回决策子树，当前节点停止递归。

2) 计算样本集D的基尼系数，如果基尼系数小于阈值，则返回决策子树，当前节点停止递归。

3) 计算当前节点现有的各个特征的各个特征值对数据集D的基尼系数，对于离散值和连续值的处理方法和基尼系数的计算见第二节。缺失值的处理方法和上篇的C4.5算法里描述的不同。

4) 在计算出来的各个特征的各个特征值对数据集D的基尼系数中，选择基尼系数最小的特征A和对应的特征值a。根据这个最优特征和最优特征值，把数据集划分成两部分D1和D2，同时建立当前节点的左右节点，做节点的数据集D为D1，右节点的数据集D为D2。

5) 对左右的子节点递归的调用1-4步，生成决策树。

深度学习进阶书

深度学习入门书

### 积分与排名

积分 - 303959

排名 - 604

### 阅读排行榜

1. 梯度下降 (Gradient Descent) 小结(10587 0)
2. 梯度提升树(GBDT)原理小结(52770)
3. 线性判别分析LDA原理总结(34992)
4. scikit-learn决策树算法类库使用小结(29932)
5. 谱聚类 (spectral clustering) 原理总结(230 74)

### 评论排行榜

1. 梯度提升树(GBDT)原理小结(86)
2. 谱聚类 (spectral clustering) 原理总结(75)
3. 梯度下降 (Gradient Descent) 小结(65)
4. 卷积神经网络(CNN)反向传播算法(59)
5. 集成学习之Adaboost算法原理小结(50)

### 推荐排行榜

1. 梯度下降 (Gradient Descent) 小结(44)
2. 奇异值分解(SVD)原理与在降维中的应用(1 7)
3. 集成学习原理小结(15)
4. 卷积神经网络(CNN)反向传播算法(14)
5. 集成学习之Adaboost算法原理小结(13)

对于生成的决策树做预测的时候，假如测试集里的样本A落到了某个叶子节点，而节点里有多个训练样本。则对于A的类别预测采用的是这个叶子节点里概率最大的类别。

## 4. CART回归树建立算法

CART回归树和CART分类树的建立算法大部分是类似的，所以这里我们只讨论CART回归树和CART分类树的建立算法不同的地方。

首先，我们要明白，什么是回归树，什么是分类树。两者的区别在于样本输出，如果样本输出是离散值，那么这是一颗分类树。如果果样本输出是连续值，那么那么这是一颗回归树。

除了概念的不同，CART回归树和CART分类树的建立和预测的区别主要有下面两点：

- 1)连续值的处理方法不同
- 2)决策树建立后做预测的方式不同。

对于连续值的处理，我们知道CART分类树采用的是用基尼系数的大小来度量特征的各个划分点的优劣情况。这比较适合分类模型，但是对于回归模型，我们使用了常见的和方差的度量方式，CART回归树的度量目标是，对于任意划分特征A，对应的任意划分点s两边划分成的数据集D1和D2，求出使D1和D2各自集合的均方差最小，同时D1和D2的均方差之和最小所对应的特征和特征值划分点。表达式为：

$$\min_{A,s} \left[ \min_{c_1} \sum_{x_i \in D_1(A,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in D_2(A,s)} (y_i - c_2)^2 \right]$$

其中， $c_1$  为D1数据集的样本输出均值， $c_2$  为D2数据集的样本输出均值。

对于决策树建立后做预测的方式，上面讲到了CART分类树采用叶子节点里概率最大的类别作为当前节点的预测类别。而回归树输出不是类别，它采用的是用最终叶子的均值或者中位数来预测输出结果。

除了上面提到了以外，CART回归树和CART分类树的建立算法和预测没有什么区别。

## 5. CART树算法的剪枝

CART回归树和CART分类树的剪枝策略除了在度量损失的时候一个使用均方差，一个使用基尼系数，算法基本完全一样，这里我们一起来讲。

由于决策时算法很容易对训练集过拟合，而导致泛化能力差，为了解决这个问题，我们需要对CART树进行剪枝，即类似于线性回归的正则化，来增加决策树的返回能力。但是，有很多的剪枝方法，我们应该这么选择呢？CART采用的办法是后剪枝法，即先生成决策树，然后产生所有可能的剪枝后的CART树，然后使用交叉验证来检验各种剪枝的效果，选择泛化能力最好的剪枝策略。

也就是说，CART树的剪枝算法可以概括为两步，第一步是从原始决策树生成各种剪枝效果的决策树，第二部是用交叉验证来检验剪枝后的预测能力，选择泛化预测能力最好的剪枝后的数作为最终的CART树。

首先我们看看剪枝的损失函数度量，在剪枝的过程中，对于任意的一棵子树 $T$ ，其损失函数为：

$$C_{\alpha}(T_t) = C(T_t) + \alpha|T_t|$$

其中， $\alpha$ 为正则化参数，这和线性回归的正则化一样。 $C(T_t)$ 为训练数据的预测误差，分类树是用基尼系数度量，回归树是均方差度量。 $|T_t|$ 是子树 $T$ 的叶子节点的数量。

当 $\alpha = 0$ 时，即没有正则化，原始的生成的CART树即为最优子树。当 $\alpha = \infty$ 时，即正则化强度达到最大，此时由原始的生成的CART树的根节点组成的单节点树为最优子树。当然，这是两种极端情况。一般来说， $\alpha$ 越大，则剪枝剪的越厉害，生成的最优子树相比原生决策树就越偏小。对于固定的 $\alpha$ ，一定存在使损失函数 $C_{\alpha}(T)$ 最小的唯一子树。

看过剪枝的损失函数度量后，我们再来看看剪枝的思路，对于位于节点 $t$ 的任意一颗子树 $T_t$ ，如果没有剪枝，它的损失是

$$C_{\alpha}(T_t) = C(T_t) + \alpha|T_t|$$

如果将其剪掉，仅仅保留根节点，则损失是

$$C_{\alpha}(T) = C(T) + \alpha$$

当 $\alpha = 0$ 或者 $\alpha$ 很小时， $C_{\alpha}(T_t) < C_{\alpha}(T)$ ，当 $\alpha$ 增大到一定的程度时

$$C_{\alpha}(T_t) = C_{\alpha}(T)$$

。当 $\alpha$ 继续增大时不等式反向，也就是说，如果满足下式：

$$\alpha = \frac{C(T) - C(T_t)}{|T_t| - 1}$$

$T_t$ 和 $T$ 有相同的损失函数，但是 $T$ 节点更少，因此可以对子树 $T_t$ 进行剪枝，也就是将它的子节点全部剪掉，变为一个叶子节点 $T$ 。

最后我们看看CART树的交叉验证策略。上面我们讲到，可以计算出每个子树是否剪枝的阈值 $\alpha$ ，如果我们把所有的节点是否剪枝的值 $\alpha$ 都计算出来，然后分别针对不同的 $\alpha$ 所对应的剪枝后的最优子树做交叉验证。这样就可以选择一个最好的 $\alpha$ ，有了这个 $\alpha$ ，我们就可以用对应的最优子树作为最终结果。

好了，有了上面的思路，我们现在来看看CART树的剪枝算法。

输入是CART树建立算法得到的原始决策树 $T$ 。

输出是最优决策子树 $T_\alpha$ 。

算法过程如下：

- 1) 初始化 $\alpha_{min} = \infty$ ，最优子树集合 $\omega = \{T\}$ 。
- 2) 从叶子节点开始自下而上计算各内部节点 $t$ 的训练误差损失函数 $C_\alpha(T_t)$ （回归树为均方差，分类树为基尼系数），叶子节点数 $|T_t|$ ，以及正则化阈值 $\alpha = \min\{\frac{C(T)-C(T_t)}{|T_t|-1}, \alpha_{min}\}$ ，更新 $\alpha_{min} = \alpha$
- 3) 得到所有节点的 $\alpha$ 值的集合 $M$ 。
- 4) 从 $M$ 中选择最大的值 $\alpha_k$ ，自上而下的访问子树 $t$ 的内部节点，如果 $\frac{C(T)-C(T_t)}{|T_t|-1} \leq \alpha_k$ 时，进行剪枝。并决定叶节点 $t$ 的值。如果是分类树，则是概率最高的类别，如果是回归树，则是所有样本输出的均值。这样得到 $\alpha_k$ 对应的最优子树 $T_k$
- 5) 最优子树集合 $\omega = \omega \cup T_k$ ， $M = M - \{\alpha_k\}$ 。
- 6) 如果 $M$ 不为空，则回到步骤4。否则就已经得到了所有的可选最优子树集合 $\omega$ 。
- 7) 采用交叉验证在 $\omega$ 选择最优子树 $T_\alpha$

## 6. CART算法小结

上面我们对CART算法做了一个详细的介绍，CART算法相比C4.5算法的分类方法，采用了简化的二叉树模型，同时特征选择采用了近似的基尼系数来简化计算。当然CART树最大的好处是还可以做回归模型，这个C4.5没有。下表给出了ID3，

C4.5和CART的一个比较总结。希望可以帮助大家理解。

算法	支持模型	树结构	特征选择	连续值处理	缺失值处理	剪枝
ID3	分类	多叉树	信息增益	不支持	不支持	不支持
C4.5	分类	多叉树	信息增益比	支持	支持	支持
CART	分类，回归	二叉树	基尼系数，均方差	支持	支持	支持

- 看起来CART算法高大上，那么CART算法还有没有什么缺点呢？有！主要的缺点我认为如下：
- 1) 应该大家有注意到，无论是ID3, C4.5还是CART,在做特征选择的时候都是选择最优的一个特征来做分类决策，但是大多数，分类决策不应该是由某一个特征决定的，而是应该由一组特征决定的。这样绝息到的决策树更加准确。这个决策树叫做多变量决策树(multi-variate decision tree)。在选择最优特征的时候，多变量决策树不是选择某一个最优特征，而是选择最优的一个特征线性组合来做决策。这个算法的代表是OC1，这里不多介绍。
  - 2) 如果样本发生一点点的改动，就会导致树结构的剧烈改变。这个可以通过集成学习里面的随机森林之类的方法解决。

## 7. 决策树算法小结

终于到了最后的总结阶段了，这里我们不再纠结于ID3, C4.5和 CART，我们来看看决策树算法作为一个大类别的分类回归算法的优缺点。这部分总结于scikit-learn的英文文档。

- 首先我们看看决策树算法的优点：
- 1) 简单直观，生成的决策树很直观。
  - 2) 基本不需要预处理，不需要提前归一化，处理缺失值。
  - 3) 使用决策树预测的代价是 $O(\log_2 m)$ 。m为样本数。
  - 4) 既可以处理离散值也可以处理连续值。很多算法只是专注于离散值或者连续值。
  - 5) 可以处理多维度输出的分类问题。
  - 6) 相比于神经网络之类的黑盒分类模型，决策树在逻辑上可以得到很好的解释
  - 7) 可以交叉验证的剪枝来选择模型，从而提高泛化能力。
  - 8) 对于异常点的容错能力好，健壮性高。

我们再看看决策树算法的缺点:

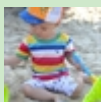


- 1) 决策树算法非常容易过拟合, 导致泛化能力不强。可以通过设置节点最少样本数量和限制决策树深度来改进。
- 2) 决策树会因为样本发生一点点的改动, 就会导致树结构的剧烈改变。这个可以通过集成学习之类的方法解决。
- 3) 寻找最优的决策树是一个NP难的问题, 我们一般是通过启发式方法, 容易陷入局部最优。可以通过集成学习之类的方法来改善。
- 4) 有些比较复杂的关系, 决策树很难学习, 比如异或。这个就没有办法了, 一般这种关系可以换神经网络分类方法来解决。
- 5) 如果某些特征的样本比例过大, 生成决策树容易偏向于这些特征。这个可以通过调节样本权重来改善。

以上就是决策树的全部内容了, 里面有很多我个人思考的逻辑在, 希望能对大家有所帮助, 有错误的话请指正。

( 欢迎转载, 转载请注明出处。欢迎沟通交流: [pinard.liu@ericsson.com](mailto:pinard.liu@ericsson.com) )

分类: [0081. 机器学习](#)

[好文要顶](#)[关注我](#)[收藏该文](#)

刘建平Pinard

关注 - 13

粉丝 - 1176

[+加关注](#)

8

推荐

0

反对

« 上一篇: [决策树算法原理\(上\)](#)

» 下一篇: [scikit-learn决策树算法类库使用小结](#)

posted @ 2016-11-11 16:10 刘建平Pinard 阅读(16271) 评论(37) 编辑 收藏

## 评论列表


#1楼 2016-11-21 18:17 stream886



博主更新速度真快, 支持!

支持(0) 反对(0)




#2楼 2016-12-27 19:43 xulu1352 



博主，大牛，我也在常驻岭南，跪拜大牛。

支持(0) 反对(0)

#3楼 2017-04-14 14:20 Allen\_xiaoshi 



有一个问题


CART回归树

假如 有4个变量，每个变量有10个值(连续变量)

那么去计算每个变量的每个划分点的均方误差来选择优先的变量和划分点

其实就是去计算 $4 \times 9 = 36$ 次吗，然后比大小吗

支持(0) 反对(0)


#4楼[楼主 



@ Allen\_xiaoshi

是这样的


支持(0) 反对(0)

#5楼 2017-06-16 12:58 NaKy 



您好，我突然有个问题，不知道哪出错了。既然熵和基尼指数取值相似，那为什么在ID3,C4.5中要取信息增益，而不是直接取熵最小的作为最优特征呢？

支持(0) 反对(0)


#6楼[楼主 



@ NaKy

你好，其实ID3,C4.5就是基于熵的，由于我们关注的数据输出的熵在知道一些特征信息后，一般熵会变小。变小的部分即为信息增益。直接取熵做比较的话无法反应特征和输出之间的关系。

支持(0) 反对(0)


#7楼 2017-06-18 10:06 niudong 



“从描述可以看出，如果离散特征A有n个取值，则可能的组合有 $n(n-1)/2$ 种”

这个组合数是不是错了，因为每次并不是只从里面挑选2个出来，应该是分成两组。这样组合数就大多了。

支持(0) 反对(0)

#8楼[楼主 

“

@ niudong

你好，这里写的的确不对，已修改，感谢指出错误。


支持(0) 反对(0)

#9楼 2017-09-23 20:50 xx132 

“

您好，想请问下剪枝算法中的阈值的作用是不是防止计算出来的 $\alpha$ 的值太大而使得决策树太过于简单这种现象的发生


支持(0) 反对(0)

#10楼 2017-09-24 17:22 Agust 

“

大神，你好，文章里有有一句没看懂，可否解释一下“要注意的是，与离散属性不同的是，如果当前节点为连续属性，则该属性后面还可以参与子节点的产生选择过程。”这句话的意思是不是：连续的属性不仅仅是二分类，可以做多分类的情况？然后把这个多分类从多叉树变为二叉树，所以该属性还可以参与子节点产生选择的过程？

支持(0) 反对(0)

#11楼 2017-09-24 17:46 Bitter-Coffe 

“

您好，想问一下‘决策树对于异常点的容错能力好，健壮性高’是相对于哪种模型而言，并且它与‘CART如果样本发生一点的改动，就会导致树结构的剧烈改变’是否矛盾，谢谢！

支持(0) 反对(0)


#12楼 2017-09-25 00:26 Delaiah 

“

@ Bitter-Coffe

我是这样认为的：“异常点容错好”，是说某些样本的特征值异常，任然不影响它的准确分类。“若样本发生改变，会导致树结构剧烈改变”：是说若样本的分布改变，会影响树的生成，这里可以参考随机森林。

支持(0) 反对(0)


#13楼[楼主 

“

@ xx132

你好，是的，你可以理解为它的值是对决策树过于复杂和过于简单这个矛盾之间的一个折衷点。

支持(0) 反对(0)

#14楼[楼主 

“

@ Delaiah

你说的非常准确，感谢回复。

支持(0) 反对(0)

#15楼[楼主] 2017-09-25 11:55 刘建平Pinard



@ 菜鸟搬运工

你好，是的。比如一个取值范围很大的连续值特征，你不能简单粗暴的就取值一个点，一分为二，然后这两边各自类别中的不同连续值的差异就白白浪费掉了。所以在子树的分裂中还要继续考虑各自类别中的连续值是否有必要再分。

支持(0) 反对(0)

#16楼 2017-09-29 10:59 Bitter-Coffe



@ Delaiah

非常感谢！！

支持(0) 反对(0)

#17楼 2017-10-03 17:20 Bitter-Coffe



反复看了决策树剪枝算法，感觉还是理解的不够透彻，问题如下：

1>剪枝算法的步骤2的阈值时自下而上的到目前为止的最小值，那么步骤4为什么是小于阈值时进行剪枝，谢谢博主！

支持(0) 反对(0)

#18楼[楼主] 2017-10-09 11:24 刘建平Pinard



@ Bitter-Coffe

你好，回到第五节里面，什么时候需要剪枝呢？就是子树不剪枝的损失比剪枝的损失大，即此时满足：

$$C_{\alpha}(T_t) > C_{\alpha}(T)$$

即：


$$C(T_t) + \alpha|T_t| > C(T) + \alpha$$

整理上面的不等式可以得到：

$$\alpha > \frac{C(T) - C(T_t)}{|T_t| - 1}$$

所以步骤4是小于阈值进行剪枝。


支持(0) 反对(0)

#19楼 2017-10-14 14:57 开心就好硕 



博主，有没有代码实现呢，拜读下，谢谢

支持(0) 反对(0)

#20楼[楼主 




@ 开心就好硕

你好，对于决策树，主要是使用sklearn的API来分析实战。我下一篇就是代码实现，使用sklearn来做决策树分析。

scikit-learn决策树算法类库使用小结：

<http://www.cnblogs.com/pinard/p/6056319.html>

支持(0) 反对(0)


#21楼 2017-10-28 13:57 tianyu123193 



@ NaKy

您好，我觉得这样说或许便于理解，对于树状分类图，你希望他的分类准确的话，在知道上层属性A的情况下想要其分类下的下层属性是尽量准确的，那么需要B在知道A之后的不确定新大大减少。也就是说分类过程应该沿着信息不确定减少最快的方向进行。

支持(0) 反对(0)


#22楼 2017-11-10 13:28 cigarbj 



@ tianyu123193

我觉得@NaKy 所困扰的可能是这个问题：既然熵和基尼系数都是不纯度的表达，那为什么在ID3划分时考虑的是熵的增益，而在CART中只考虑基尼系数而不是基尼系数的增益？

支持(0) 反对(0)


#23楼 2017-11-10 14:07 cigarbj 



“但是对于回归模型，我们使用了常见的均方差的度量方式，CART回归树的度量目标是，对于任意划分特征A，对应的任意划分点s两边划分成的数据集D1和D2，求出使D1和D2各自集合的均方差最小，同时D1和D2的均方差之和最小所对应的特征和特征值划分点”

-----  
楼主，回归树的度量目标应该是SSE（中文名应该叫“和方差”）吧，均方差就和划分后的结点样本的数量有关了，而且您下面的公式表达的也是SSE

支持(0) 反对(0)


#24楼[楼主 

“

@ cigarbj

你好，这里说的的确不准确，已经修改。

支持(0) 反对(0)


#25楼 2017-11-11 17:09 zts131420 

“

如果我们把所有的节点是否剪枝的值 $\alpha$ 都计算出来，然后分别针对不同的 $\alpha$ 所对应的剪枝后的最优子树做交叉验证。这样就可以选择一个最好的 $\alpha$ ，有了这个 $\alpha$ ，我们就可以用对应的最优子树作为最终结果。

为什么下面的算法还要找出 $\alpha_{min}$ 呢？也没有用到

支持(0) 反对(0)


#26楼[楼主 

“

@ zts131420

你好，这个 $\alpha_{min}$ 在剪枝选择的过程中不断减少。可以减少后面算法中一些无谓的剪枝决策。不使用它也是可以的，不过算法中M的集合就会变大，多很多没有意义的计算。

支持(0) 反对(0)


#27楼 2017-11-13 12:24 zts131420 

“

@ 刘建平Pinard

懂了 多谢大佬....请问你工作了几年了啊....这些博文是我见过最好的系列文章真的,真真能称得上深入浅出..

支持(2) 反对(0)


#28楼[楼主 

“

@ zts131420

感谢赞赏，我做码农快11年了，混的不好，惭愧。

支持(0) 反对(0)


#29楼 2017-11-13 18:33 tianyu123193 

“

@ cigarbj@NaKy

是这样啊，我觉得熵和基尼系数都是不纯度的表达，这种说法只是笼统的，具体基尼系数代表了模型的不纯度，基尼系数越小，则不纯度越低，这里与经典决策树的概念相似，都是统计分类下的样本纯度，而熵度量了事物的不确定性，信息增益度量了在知道Y以后X剩下的不确定性，概念分清应该就明白了。

支持(1) 反对(0)

#30楼 2017-12-27 15:49 有前途的胡先森 

“

博主，您好，谢谢您的总结，有个问题我想问您，关于决策树，比如CART决策树，预测样本到某个最后叶子节点，其最后的叶子节点是一个类别。但是这个预测的prob是怎么来的呢？

支持(0) 反对(0)

#31楼[楼主] 2017-12-27 16:16 刘建平Pinard

“

@ 有前途的胡先森

你好，我猜你说的是sklearn里面预测prob是怎么来的，这个sklearn官方文档里面已经有解释了，也就是在该叶子节点里，最终确定的类别对应的训练样本数占划分到该叶子节点所有训练样本数的比例。

比如该叶子节点用训练样本建立决策树时候，划分到5个样本，4个类别0, 1个类别1的。那么我们用投票法确定该叶子节点类别为0.当有一个需要预测的样本在走了一遍决策树后划分到了该叶子节点，那么它的prob就是0.8。

下面是官方文档原话：

The predicted class probability is the fraction of samples of the same class in a leaf.

支持(0) 反对(0)

#32楼 2017-12-27 17:00 有前途的胡先森

“

嗯嗯，谢谢您的秒回复，还有一个问题请教您：1.分类决策树，还是拿CART来说，对于样本分布不均匀影响程度大不大，也就是不同类别数量可能有明显差异。2.如何解决这个数据不平衡问题？您有什么思路或者别的算法可以谈谈么？

支持(0) 反对(0)

#33楼[楼主] 2017-12-28 10:32 刘建平Pinard

“

@ 有前途的胡先森

1.这个影响还是很大的，也就是所谓的类别不平衡问题。

2.解决这个问题，有很多思路，最常见简单的就是加上类别权重和（或）样本权重，即样本数少的类别样本权重高，在sklearn中即为class\_weight和sample\_weight两个参数。另一个常见思路是采样，即通过采样增加少数类别样本数，或者子采样选择部分多数类别样本数，以达到类别平衡。

支持(1) 反对(0)

#34楼 2018-01-07 16:16 JeaDong233

“

老师，您第二节里面好像写错了：C4.5使用的是信息增益比，不是信息增益

另外，第三节的最后一段话：（对于生成的决策树做预测的时候，假如测试集里的样本A落到了某个叶子节点，而节点里有多条训练样本。则对于A的类别预测采用的是这个叶子节点里概率最大的类别。）

感到困惑。CART不是一直不断二分吗？那A最后也会得到一个明确的分类才对。是不是因为最后基尼系数小于阈值了，所以A可能进入一个有很多分类的节点，才考虑概率最大的类别？

支持(0) 反对(0)

#35楼[楼主] 2018-01-08 11:17 刘建平Pinard

@ JeaDong233

你好。

1.的确是信息增益比，已经修改，感谢指出错误。

2. CART树会有剪枝正则化，上面讲到过，所以不是一直二分直到某一个叶子节点类别相同。不继续二分的原因比较多，除了上面提到的剪枝，大部分决策树还会有一个不纯度阈值可以调参，比如sklearn的决策树就有min\_impurity\_split这个参数，当某个节点的不纯度高于阈值则也不会继续二分。

所以进入一个有很多分类的节点，考虑概率最大的类别是很常见的。

支持(0) 反对(0)

#36楼 2018-01-09 15:35 榛子巧克力

博主你好，请问在CART树剪枝这里，计算出了 $\alpha$ 后还要进行交叉验证，那为什么不直接基于误判进行剪枝呢？也就是说既然要利用到测试集，为什么不直接将所有非叶子节点的子树依次替换成叶子节点，再利用测试集的测试结果决定是否用该叶子节点来代替非叶子节点的子树（也就是进行剪枝）呢？我以为不需要测试集的悲观剪枝才会引入正则化因子 $\alpha$ 来计算错误率。

支持(0) 反对(0)

#37楼[楼主] 2018-01-10 11:29 刘建平Pinard

@ 榛子巧克力


你好，你的方法也可行，但是不是我们常用的数据处理的流程。

我们一般的做法，划分训练集，测试集。接着对训练集划分S份，进行S折交叉验证，选择合适的模型参数(包括正则化参数)。最后拿我们得到的交叉验证后的最优模型参数去验证测试集的预测效果。

所以在决策树剪枝这里，一般不能在训练模型的时候就直接使用测试集的信息。

支持(1) 反对(0)

[刷新评论](#) [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](#)。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库！

【报名】2050 大会 - 博客园程序员团聚（5.25 杭州·云栖小镇）

【招聘】花大价钱找技术大牛我们是认真的！

【腾讯云】买域名送解析+SSL证书+建站





腾讯云

## 助力开发者快速搭建小程序

一站式配置主机和域名  
套餐11元/月起

立即抢购

### 最新IT新闻:

- 彭蕾将卸任蚂蚁金服董事长，CEO井贤栋将兼任董事长一职
  - 小米河南营销团队23名员工遭解聘 新零售之路坎坷前行
  - 乐视网在近一年内3次换帅 融创系刘淑青任董事长
  - Facebook违背和美政府隐私协议 将面临10亿美元罚款
  - 董明珠：依托诚信走出去 以自主创新应对贸易摩擦
- » 更多新闻...

### 最新知识库文章:

- 写给自学者的入门指南
  - 和程序员谈恋爱
  - 学会学习
  - 优秀技术人的管理陷阱
  - 作为一个程序员，数学对你到底有多重要
- » 更多知识库文章...

Copyright ©2018 刘建平Pinard