CART 分类回归树

1. CART 算法简介
2. CART 解决分类问题
3. CART 分类预测
4. CART 建立回归树与预测
5. CART 递归调整与封装

1.1 CART 算法

CART(Classification and Regression Tree)

是 L.Breiman(布雷曼) 等人在 1984 年提出的决策树算法

能解决的分类和回归问题

分类问题： Gini 系数

回归问题：最小均方误差、最小绝对误差 (设定阈值)

CART 可以处理连续和离散的特征

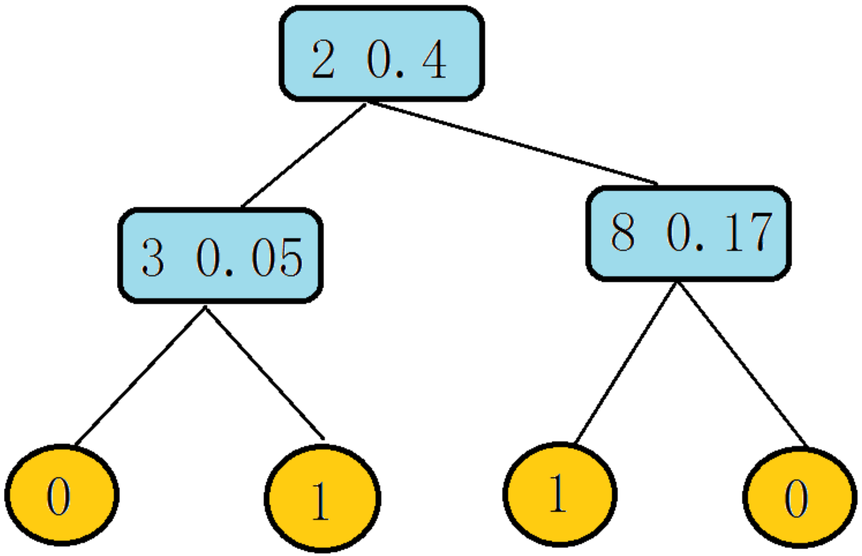
每次决策采用二分方法将样本集分为两个子集

因此所得的树结构是二叉树

验证剪枝：用独立的验证数据集对训练集生长的树进行剪枝

1.2 CART 树结构

二叉树（连续性变量）



二叉树：每个决策点只有两个分支。

每个决策点是特征与分割值，叶子节点表示一种分类或者一个预测值。

也就是每个决策点不仅要确定选取哪个特征，与此同时还要确定“二分值”

2.1 基尼系数

1984 年，Breiman 博士提出基尼系数，用来在决策树的生成中度量**样本的不确定性程度**。

假设样本集有 m 个分类，样本点属于第 i 类的概率 p_i，定义该概率分布的基尼系数为：

$$\mathrm{Gini}(p)=\sum_{i=1}^{m}p_i\left(1-p_i\right)=1-\sum_{i=1}^{m}p_i^{2}$$

对于给定的样本集 D(标签有 m 种不同情况)，设共有 N 个样本，且属于第 i 种标签的样本有 n_i 个，则该样本集的基尼系数为：

$$\mathrm{Gini}(D)=\sum_{i=1}^{m}\frac{n_i}{N}\left(1-\frac{n_i}{N}\right)=1-\sum_{i=1}^{m}\left(\frac{n_i}{N}\right)^{2}$$

2.1 基尼系数：（以二分类为例）

假设: $x \in \{\alpha_1, \alpha_2\}$

两种情况对应的概率为: $p = \{p_1, p_2\}$

p_1=0.5, p_2=0.5

$$Gini = (p_1(1-p_1) + p_2(1-p_2))$$
$$= (0.5 \times 0.5 + 0.5 \times 0.5) = 0.5$$

p_1=0.2, p_2=0.8

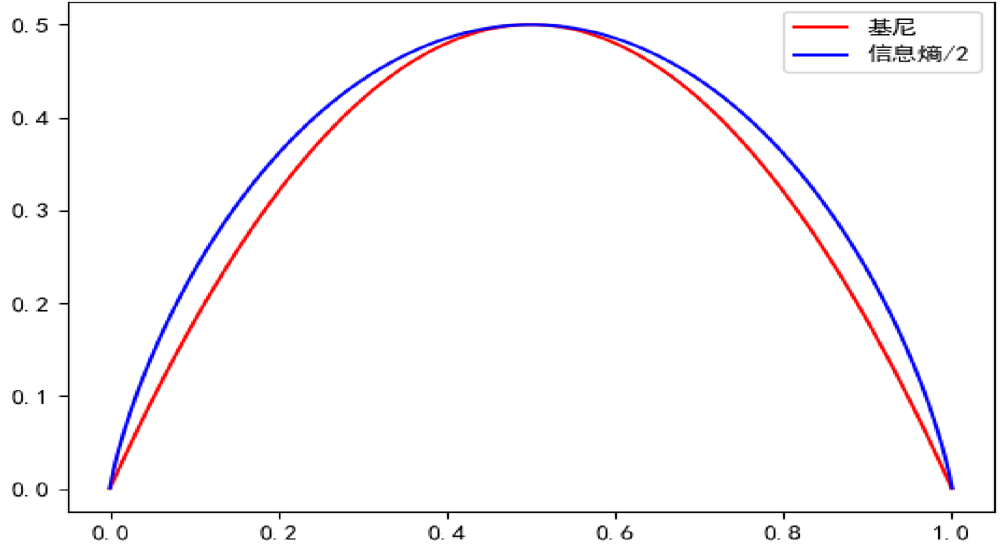
$$Gini = 1 - (p_1{}^2 + p_2{}^2)$$
$$= 1 - (0.2^2 + 0.8^2) = 0.32$$

p_1=0, p_2=1

$$Gini = (p_1(1-p_1) + p_2(1-p_2))$$
$$= (0 \times 1 + 1 \times 0) = 0$$

2.2 基尼系数与熵的关系

以二分类为例，可以发现当其中某一类的概率越大时，基尼系数越小，也就是，分布数据分布越不纯净，基尼系数越大；当数据符合均匀分布时，基尼系数取到最大值



利用基尼系数能够将数据集划分为较为均衡的两部分

2.3 样本集的基尼系数

$$D = \left[\begin{array}{ccc|c} 0 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 \\ 1 & 2 & 2 & 2 \\ 3 & 2 & 2 & 3 \end{array}\right]$$

有三个标签：分别是 1 , 2 , 3
出现次数分别是　　1 , 2 , 1

由于此时并不知道总体的真实分布，只能将样本中标签的频率作为真实概率的估计。即：

$$p(y=1)=\frac{1}{4}, p(y=2)=\frac{1}{2}, p(y=3)=\frac{1}{4}$$

$$gini(D)=1-\left[\left(\frac{1}{4}\right)^2+\left(\frac{1}{2}\right)^2+\left(\frac{1}{4}\right)^2\right]=0.625$$

2.4 条件基尼系数

连续型特征的条件基尼系数

按特征 x^i ，二分标准 v 划分后得到的条件基尼指数

$$\text{Gini}(D|(x^i,v)) = \frac{|D|x^i<v|}{|D|} Gini(D|x^i<v) + \frac{|D|x^i \geq v|}{|D|} Gini(D|x^i \geq v)$$

按第一个特征，二分标准 **2** 划分后得到的条件基尼指数

$$D = \begin{bmatrix} 1 & 1 & | & 1 \\ 1 & 2 & | & 2 \\ 2 & 2 & | & 2 \\ 2 & 2 & | & 3 \end{bmatrix}$$

$$(D|x^1<2) = \begin{bmatrix} 1 & 1 & | & 1 \\ 1 & 2 & | & 2 \\ 2 & 2 & | & 2 \end{bmatrix}$$

$$(D|x^1 \geq 2) = \begin{bmatrix} 2 & 2 & | & 3 \end{bmatrix}$$

2.4 条件基尼系数

$$Gini(D|(x^1,2))=\left(\frac{3}{4}\cdot Gini(D|x^1<2)+\frac{1}{4}\cdot Gini(D|x^1\geq 2)\right)$$

$$=0.75\cdot\left(\frac{1}{3}\cdot\left(1-\frac{1}{3}\right)+\frac{2}{3}\cdot\left(1-\frac{2}{3}\right)\right)+0.25\cdot(0)$$

$$=0.333$$

CART 选择决策点选择时，摒弃了增益的方法，直接比较条件基尼系数，

条件基尼系数越小的（特征，二分标准），不纯度降低越多

优点：减少变量

条件基尼系数的计算可以直接传入左右子数据集

2.5 离散特征的条件基尼系数

按特征x^i（离散特征），二分标准α划分后得到的条件基尼系数

$$\mathrm{Gini}(D|(x^i,\alpha)) = \frac{|D|x^i=\alpha|}{|D|}Gini(D|x^i=\alpha) + \frac{|D|x^i \neq \alpha|}{|D|}Gini(D|x^i \neq \alpha)$$

可以将离散特征的二分方法表示成和连续特征相同的形式：
对离散特征进行 one-hot 编码

假设样本集第i个特征x^i有三种取值 $\{\alpha_1,\alpha_2,\alpha_3\}$

用三个特征代替原特征
分别为 $i\_\alpha_1, i\_\alpha_2, i\_\alpha_3$

i	i_α_1	i_α_2	i_α_3
x_2	0	1	0
x_3	0	0	1

2.6　选择最优特征与二分标准

三个函数：创建 CART.py

函数： split_data()， condition_gini()， get_split()

利用 (特征，二分标准) 对样本集进行分割

计算条件基尼系数

获得最优的特征与最优二分标准

决策点过程：

1. 对每个特征，以及可能的二分标准，将数据集分为左右两个子数据集
2. 计算分割的左右子数据集的条件基尼系数，在所有分割情况中，基尼系数最小的特征与二分标准即为最优切分点

2.6 按特征索引和二分标准分割

split_data 函数
输入：样本集 data，特征索引 index，二分标准 value
输出：左子样本集，右子样本集
注意：与 ID3 不同，第 index 个特征，不用删除

对样本集 data 中的每个样本 sample
如果 sample 中的第 index 个特征的值小于 value：
    将 sample 放到左子集
否则：
    将 sample 放到右子集

2.6 基尼系数

1. 求基尼系数　2. 求条件基尼系数　3. 合并

一：样本总体的基尼系数 gini：

输入：样本集 data

输出： giniScore

找到标签的所有取值情况

对于每种标签，计算在 data 中出现的频数以及频率

将频率作为该标签的估计概率

利用基尼系数公式，求得 giniScore

2.6 基尼系数

```
def gini(data):
    lenData=len(data)
    label_list= [row[-1] for row in data]
    labelSpace=set(label_list)
    giniScore=0.0
    for label in labelSpace:
        pi=label_list.count(label) /lenData
        giniScore+=pi*(1-pi)
    return giniScore
```

2.6 基尼系数函数设计

计算条件基尼系数 condition_gini
输入：左子集 left，右子集 right
输出： conGini
分别对左子集和右子集
    计算样本集的 giniScore
    计算该样本集与总样本集比例
    通过累加，计算条件基尼系数 conGini

输入的部分可将左右子集放入元组 groups 中即 groups=(left,right)，累加的过程变为循环 groups 中每个元素

2.6 基尼系数函数设计

```
def condition_gini(groups):
    conGini=0.0
    totalLen=len(groups[0])+len(groups[1])
    for data in groups:
        giniScore=gini(data)
        conGini+=len(data)/totalLen*giniScore
    return conGini
```

2.6 最优特征与最优二分标准

get_split 函数

输入： 样本集 data

输出： index , value ， 子样本集 groups=[left , right]

\# 具有两层循环 (准备特征个数， b_gini=1)

外层是对特征索引进行循环

内层是对该特征的二分标准进行循环

对样本集进行划分 (调用 split_data 函数)

并求出基尼系数 (调用 condition_gini 函数)

通过比较得到最优情况 (giniScore 越小越好)

\# 将特征空间的每个值作为二分标准

2.6 最优特征与最优二分标准

```
def get_split(data):
	label_list=list(set([d[-1] for d in data]))
	b_gini=999
	b_index, b_value, b_groups = -1, -1, []
	for index in range(len(data[0])-1):
		fea_value=list(set([d[index] for d in data]))
		for value in fea_value:
			groups=split(data,index,value)
			gini=condition_gini(groups)
			if gini<b_gini:
				b_gini=gini                                             b_index=index; 
b_value=value;                                  b_groups=groups
	return b_index,b_value,b_groups
```

2.7 例子

```
dataSet=[[1,1,'yes'],
         [1,1,'yes'],
         [1,0,'no'],
         [0,1,'no'],
         [0,1,'no']]
```

求得样本集的最优特征索引和最优“二分标准”

2.8 基尼系数函数的合并

由于摒弃了增益的方式，不需要计算原样本集的基尼系数，只需要计算条件基尼系数，直接求条件基尼系数

输入：元组 groups　　输出：conGini

对 groups 中每个元素 data

　　找到 data 中标签的所有取值情况

　　　对于每种标签，

　　　　计算在 data 中出现的频数以及频率

　　　　用频率与基尼系数公式计算 **data 的 giniScore**

　　　　计算该样本集与总样本集比例

　　　　通过累加，计算条件基尼系数 **conGini**

2.6 基尼系数

gini(groups)：

```
total=len(groups[0])+len(groups[1])
gini_score =0
for data in groups:
    dataLen=len(data)
    labelSpace=set([row[-1] for row in data])
    for value in labelSpace:
        pi=[d[-1] for d in data].count(value)
        pi=pi/dataLen
        gini+=dataLen/total*pi*(1-pi)
return gini_score
```