决策树与集成算法

决策树：

多叉树：ID3算法，C4.5算法，

二叉树：CART分类回归树

剪枝：正则化剪枝、验证剪枝

集成算法：

bagging和随机森林

boosting：Adaboost

GBM(GBDT, XGBoost, lightGBM)

算法调参与数据预处理：

调参评价方法：准确率，AUC值，均方误差

调参方法：单验证集，k折交叉验证

数据预处理：标签分布，特征类型，相关性分析，

缺失值填充，独热处理，标准化

本章授课内容

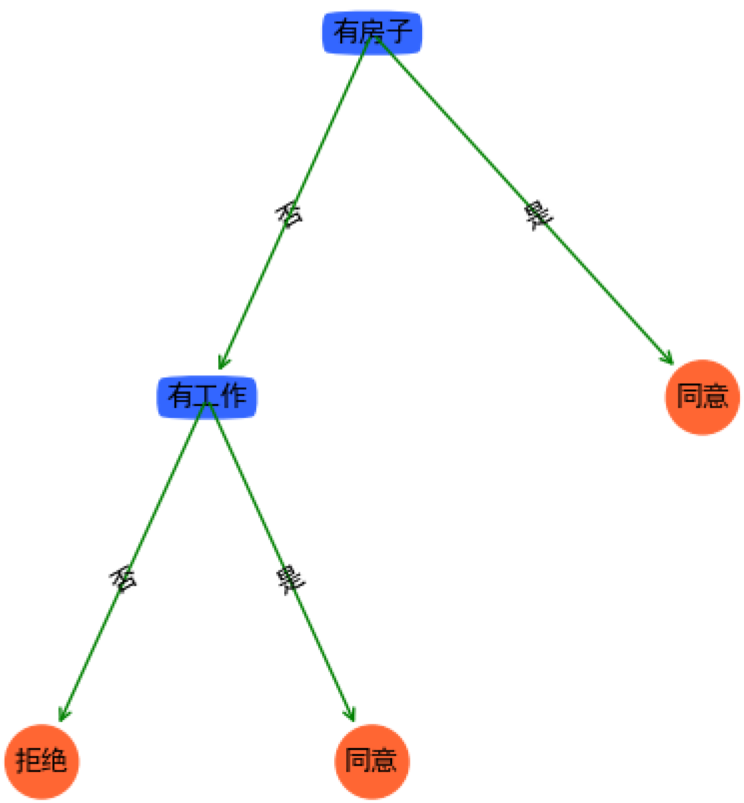
河北师范大学软件学院
Software College of Hebei Normal University

1. 决策树的结构与ID3算法特点
2. 信息熵、条件熵和信息增益
3. 熵与信息增益编程实现
4. ID3算法–递归建树
5. 使用ID3算法进行预测

例1：信贷类别预测样本

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

1.1 决策树的结构



决策树：以树状结构表示数据分类或回归预测的过程。

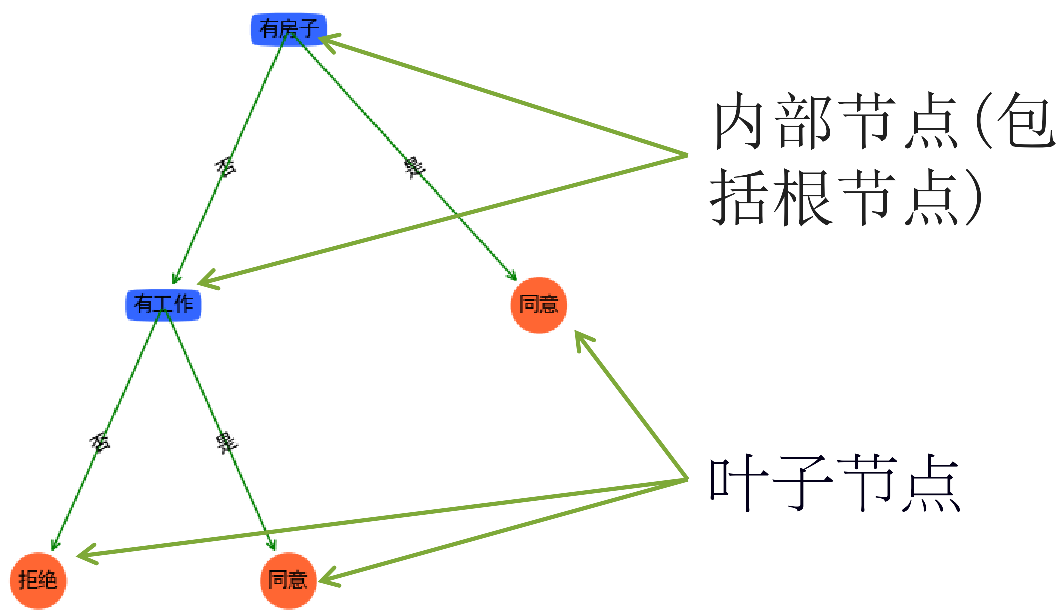
1.决策节点（决策点）

2.分支

3.叶子节点

1.1 决策树的结构(泛化，函数化)

1. 决策树由节点和分支组成。
2. 节点具有两种类型：内部节点和叶子节点。内部节点表示一个**特征**或者一个**特征与分割值**，叶子节点表示一种**分类**或者一个**回归值**。
3. 分支：每个决策点实现一个具有离散输出的函数，记为分支。(多叉树和二叉树)



三个问题：

1. 选取哪个特征作为当前分类的标准
2. 选取了特征如何分类(多分支、二分支)
3. 什么时候结束(从决策点变成叶子节点)

1.3 ID3算法特点

1. 使用**信息熵与信息增益**确定特征
2. 特征有多少种取值情况就有几个分支(**多叉树**)
3. 能处理**离散型的特征**
4. 能解决**分类问题**

2. 1 三种衡量方式

三种不同的衡量方式：

1.信息熵

$$E(P) = -\sum_{i=1}^{N} p_i \ln(p_i)$$

2.基尼系数

$$\mathrm{Gini}(P) = \sum_{i=1}^{N} p_i (1 - p_i) = 1 - \sum_{i=1}^{N} p_i^{\ 2}$$

3. 最小均方误差(用于回归)

$$mes(Y) = \sum_{x_i < p} \left(y_i - c_{jp}^{(1)} \right)^2 + \sum_{x_i \geq p} \left(y_i - c_{jp}^{(2)} \right)^2$$

其中： $c_{jp}^{(1)} = \mathrm{avg}(\mathrm{y}_i \mid x_i < p)$、$c_{jp}^{(2)} = \mathrm{avg}(\mathrm{y}_i \mid x_i \geq p)$

2. 1 信息熵介绍

C.Shannon的信息论

1. Father of information theory
2. 解决了对信息的量化度量问题

熵（entropy）

信息：系统的平均信息量

统计：系统的混乱程度：

若一个系统中存在多个事件：$E_1, E_2, \cdots, E_n$ 每个事件出现的概率是 $p_1, p_2, \cdots, p_n$; 则这个系统的混乱程度为：

$$\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \ln p_1 - p_2 \ln p_2 \ldots - p_n \ln p_n$$

2.1 熵的计算:（以二分类为例）

假设：$x \in \{\alpha_1, \alpha_2\}$

两种情况对应的概率为：$p = \{p_1, p_2\}$

$p_1=0.5, p_2=0.5$

$$Ent = -(p_1 \ln(p_1) + p_2 \ln(p_2))$$
$$= -(0.5*(-0.693)*2) = 0.693$$

$p_1=0.2, p_2=0.8$

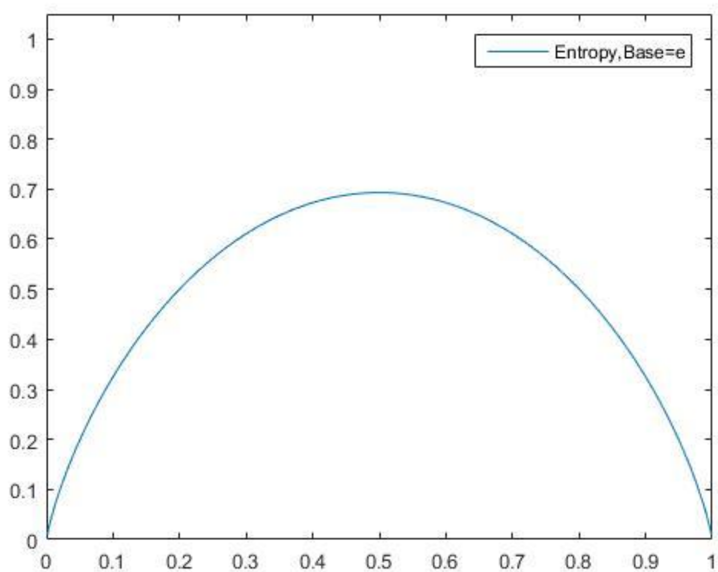
$$Ent = -(0.2\ln(0.2) + 0.8\ln(0.8))$$
$$= -(-0.321+-0.179)=0.500$$

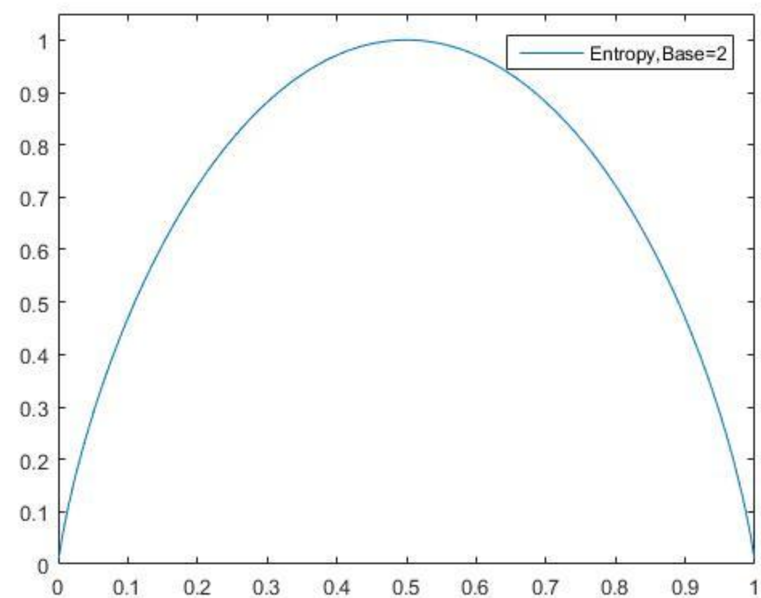
$p_1=0, p_2=1$

$$Ent = -(0\ln(0) + 1\ln(1))$$
$$=0$$

2.1 信息熵的特点

从图中可以看出，当两个类别比例相等时，熵值最大(也就是**均匀分布时熵最大**)；
当其中一个类别的比例越来越大时，熵值就越小(**数据集越纯净，熵越小**)。
为了方便计算，经常选2为底





例1：信贷类别预测样本

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

熵

条件熵

增益

样本空间符号表达：

$$\begin{array}{c} \begin{array}{ccc|c} x^{(1)} & x^{(2)} & x^{(3)} & y \end{array} \\ \mathrm{D}=\left[\begin{array}{ccc|c} x_1^{(1)} & x_1^{(2)} & x_1^{(3)} & y^1 \\ x_2^{(1)} & x_2^{(2)} & x_2^{(3)} & y^2 \\ x_3^{(1)} & x_3^{(2)} & x_3^{(3)} & y^3 \\ x_4^{(1)} & x_4^{(2)} & x_4^{(3)} & y^4 \end{array}\right] \end{array}$$

y^j, y_j都表示

第j个样本的标签

$x^{(i)}$表示所有样本的第i个特征（的值）

$x_j^{(i)}$表示第j个样本的第i个特征的值

$x_j^{(i)} \in \{\alpha_1, \cdots, \alpha_k\}$，且$\alpha_1 \neq \alpha_2 \cdots \neq \alpha_k$

$\{\alpha_1, \cdots, \alpha_k\}$称为第$i$个特征的取值情况

2.1 熵(信息熵、经验熵)

我们要用的：标签的熵

$$\begin{array}{c} \begin{array}{cccc} x^{(1)} & x^{(2)} & x^{(3)} & y \end{array} \\ D = \left|\begin{array}{ccc|c} x_1^{(1)} & x_1^{(2)} & x_1^{(3)} & y^1 \\ x_2^{(1)} & x_2^{(2)} & x_2^{(3)} & y^2 \\ x_3^{(1)} & x_3^{(2)} & x_3^{(3)} & y^3 \\ x_4^{(1)} & x_4^{(2)} & x_4^{(3)} & y^4 \end{array}\right| \end{array}$$

假设
$$\left|\begin{array}{c} \boxed{y^1 = l_1} \\ y^2 = l_2 \\ y^3 = l_2 \\ y^4 = l_2 \end{array}\right|$$

$$E(D) = -\frac{1}{4}\log_2(\frac{1}{4}) - \frac{3}{4}\log_2(\frac{3}{4})$$

假设，N个样本的标签共有m种类别，标签为l_i 的样本有 n_i 个，那么这个样本集标签的信息熵即为：

$$\boxed{E(D) = -\sum_{i=1}^{m}\frac{n_i}{N}\log_2\frac{n_i}{N}}$$

练习

2.2 条件熵

按某个特征分割后各子样本的熵的加权平均

$$
\begin{array}{cccc} x^{(1)} & x^{(2)} & x^{(3)} & y \end{array}
$$

$$
\mathrm{D} = \left|\begin{array}{ccc|c} x_1^{(1)} & x_1^{(2)} & x_1^{(3)} & y^1 \\ x_2^{(1)} & x_2^{(2)} & x_2^{(3)} & y^2 \\ x_3^{(1)} & x_3^{(2)} & x_3^{(3)} & y^3 \\ x_4^{(1)} & x_4^{(2)} & x_4^{(3)} & y^4 \end{array}\right|
$$

$x^{(1)}, x^{(2)}, x^{(3)}$是三个特征

$x^{(1)}$的取值情况，$x_i^{(1)} \in \{\alpha_1, \cdots, \alpha_k\}$,

按特征 $x^{(1)}$分割后的条件熵为：

$$
E(D \mid x^1) = \sum_{j=1}^{k} p(x^1 = \alpha_j) E(D \mid x^1 = \alpha_j)
$$

2.2 条件熵

按第一个特征分割后的条件熵

$$D=\left|\begin{matrix}1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \\ 2 & 2 & 2\end{matrix}\right| \begin{matrix}1 \\ 2 \\ 2 \\ 2\end{matrix}\right|$$

$$(D|x^1=1)=\left|\begin{matrix}1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2\end{matrix}\right| \begin{matrix}1 \\ 2 \\ 2\end{matrix}\right|$$

$$(D|x^1=2)=\left|\begin{matrix}2 & 2 & 2\end{matrix}\right| \begin{matrix}2\end{matrix}\right|$$

$x^{(1)}$的所有取值，$x_i^{(1)}\in\{1,2\}$，

$$E(D|x^1)=3/4*E(D|x^1=1)+1/4*E(D|x^1=2)$$

$$=0.75*(0.528+0.390)+0.25*(0)$$

$$=0.6885$$

练习

2.3 信息增益(Information Gain)

按特征x^i分割后的信息增益为:

$$IG(x^i) = E(D) - E(D|x^i)$$
$$= E(D) - \sum_{j=1}^{k} p(x = \alpha_j^i) E(D|\alpha_j^i)$$

信息增益可以认为是:
未分割之前的不纯度-分割之后的加权平均不纯度

从不纯——变纯的过程(变化的量(减小的量))越大越好

↓

Information Gain 越大的**特征**越能更好的划分数据

↓

IG**最大的特征**应该是最先被划分的特征

2.3 信息增益:

ID3算法选取特征的标准

练习

有自己的房子

是　　　否

ID	年龄	有工作	信贷情况	类别
4	青年	是	一般	是
8	中年	是	好	是
9	中年	否	非常好	是
10	中年	否	非常好	是
11	老年	否	非常好	是
12	老年	都	好	是

表1

ID	年龄	有工作	信贷情况	类别
1	青年	否	一般	否
2	青年	否	好	否
3	青年	是	好	是
5	青年	否	一般	否
6	中年	否	一般	否
7	中年	否	好	否
13	老年	是	好	是
14	老年	是	非常好	是
15	老年	否	一般	否

表2

2.3 信息增益率(IG Ratio)

信息增益 $IG(x^i)$:直观意义是当数据集按 x^i 特征空间进行分类后的不纯度减少量。又称作绝对信息增益。

信息增益率(相对信息增益)：由于每个特征的特征空间长度不一，有时特征空间长度的差异会严重影响划分效果。因此使用信息增益率这种相对减少量进行度量。

按特征 x^i 分割后的信息增益率为：

$$IG\_Ratio(x^i) = \frac{E(D) - E(D|x^i)}{E(x^i)}$$

$$= \frac{E(D) - \sum_{j=1}^{k} p(x^i = \alpha_j^i) E(D|\alpha_j^i)}{E(x^i)}$$

其中：$E(x^i)$ 表示**基于特征** x^i 的熵

3. 如何确定最优特征：

第一步：计算全部样本基于标签的**熵**；

第二布：计算利用每个特征划分后的**条件熵与信息增益**；

第三步：比较每个特征的信息增益，

选择最大信息增益对应的特征对样本进行划分。

创建tree.py

1. 计算样本的信息熵(calc_shannon_ent(data))
2. 获得样本集按某个**特征**及其**取值**划分后的子样本集

   (split_data(data,index,value))
3. 获得信息增益最大的特征(get_best_split(data))

   两层循环，调用了前面两个函数)

3.1 计算信息熵：clac_shannon_ent

计算信息熵的过程：

对不同标签进行计数，然后累加求熵

clac_shannon_ent函数流程：

输入：样本集data

输出：信息熵shannonEnt

计算样本总数num;创建一个空字典label_count={}

对每个样本：

　　如果字典没有该样本标签对应的键：

　　　　添加键等于该标签，值为0

　　字典内相应标签计数加1

初始化shannonEnt=0.0

字典内每个值(该类标签的计数)：

　　计算该类标签的比例pi

　　更新shannonEnt=-pi*log2(pi)

3.2 split_data函数：

1. 按**特征的索引**和**特征的值**获取子集
2. **把这个特征删除**

因为在建树过程中，下一个节点分裂时，要再次计算熵，这样可以减少运算量并避免该特征的影响。

split_data函数流程：

输入：data, index, value
输出：分割的子数据集split_data
sub_data初始化为空列表
对样本集data的每一行：
    如果该行第index个特征等于value：
        删除该特征
        添加到sub_data
返回sub_data

split_data函数注意点：

删除一行中的某个值：
a=[3,2,1,0];index=2
#删除第index个值
del(a[index])
	#也能删除而且更安全
	c=a[:index]
	c.extend(a[index+1:])

3.3 获取信息增益最大的特征

get_best_split：输入：data，输出：最优特征b_index

外层循环：每个特征

　　通过集合的方法获得该特征的特征空间；

　　内层循环：遍历该特征的每一种取值

　　　　计算划分的子样本集

　　　　计算子样本的熵，通过叠加计算条件熵

　　计算信息增益

　　通过比较信息增益获得最优的特征

3.3 利用信息增益获取最优分割特征

样本总体的信息熵base_ent

最优信息增量best_IG=0.0；最优特征索引best_index=-1

计算样本特征的数量feature_num

遍历特征的索引index：

    得到这个特征的特征空间unique_feature；

    初始化条件熵cond_ent=0

    对于特征空间中的每个值：

        调用split得到分割后的子样本集

        cond_ent+=子样本集熵*子样本数量/总数量

    计算当前信息增益current_IG

    如果current_IG>best_IG：

        best_IG=current_IG

        best_index=index

返回最优特征索引best_index

练习