ML-03决策树

201300086

（附DecisionTree.py，可处理信息增益、基尼指数、缺失值，输入数据集直接输出决策树）

一、信息熵

(1)

1. 对于不含冲突样本（即属性值相同但标记不同的样本）的训练集, 必存在与训练集一致（训练误差为 0）的决策树. 如果训练集可以包含无穷多个样本, 是否一定存在与训练集一致的深度有限的决策树? 并说明理由 (仅考虑每次划分仅包含一次属性判断的决策树).

一定存在，反证法如下：

在默认属性与原训练集一样多时（有限个属性），当训练集可以包含无穷多个样本时，假设不存在与训练集一致的决策树：

那么训练集训练出的决策树上至少有一个节点满足这个结点存在无法划分的多个数据，这是因为如果节点上没有冲突数据，那么总是能够将数据分开。这与训练集不含冲突数据矛盾。

(2)

信息熵 $\mathrm{Ent}(D)$ 定义如下

$$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k$$

请证明信息熵的上下界为

$$0 \leq \mathrm{Ent}(D) \leq \log_2 |\mathcal{Y}|$$

并给出等号成立的条件.

LHS:

$$\text{首先一定有：} \begin{cases} \sum_{k=1}^{|\mathcal{Y}|} p_k = 1 \\ |\mathcal{Y}| \geq 1 \\ 0 \leq p_k < 1 \end{cases} \text{因此}\forall k\text{满足} \begin{cases} log_2 p_k \leq 0 \\ -p_k log_2 p_k \geq 0 \end{cases}$$

$$\text{所以} Ent(D) \geq 0, \text{取等条件} |\mathcal{Y}| = 1$$

RHS:

$$令f(x)=xlog_2x,则f^{'}(x)=log_2x+ln2,f^{''}(x)=\frac{ln2}{x}>0恒成立$$

$$因此f(x)是下凸函数，由琴声不等式，对于\sum_{k=1}^{|\mathcal{Y}|}p_k=1，一定有$$

$$\frac{\sum_{k=1}^{|\mathcal{Y}|}f(p_k)}{|\mathcal{Y}|}\geq f(\frac{\sum_{k=1}^{|\mathcal{Y}|}p_k}{|\mathcal{Y}|})$$

$$即\sum_{k=1}^{|\mathcal{Y}|}p_klog_2p_k\geq|\mathcal{Y}|f(\frac{1}{|\mathcal{Y}|})=-log_2|\mathcal{Y}|$$

$$所以Ent(D)\leq log_2|\mathcal{Y}|,取等条件\forall k满足p_k=\frac{1}{|\mathcal{Y}|}$$

(3)

在 ID3 决策树的生成过程中, 需要计算信息增益 (information gain) 以生成新的结点. 设离散属性 a 有 V 个可能取值 $\{a^1,a^2,\cdots,a^V\}$, 请考教材 4.2.1 节相关符号的定义证明:

$$\operatorname{Gain}(D,a)=\operatorname{Ent}(D)-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\operatorname{Ent}\left(D^v\right)\geq 0 \tag{3}$$

即信息增益非负.

$$记p_{kv}：取值为a^v的结点中第k类样本在D^v中所占比例$$

$$显然有\begin{cases}\sum_{k=1}^{|\mathcal{Y}|}p_{kv}=1\\ \sum_{v=1}^{V}\frac{|D^v|}{|D|}p_{kv}=p_k\end{cases}$$

$$由于\forall k,v满足1\geq p_{kv}\geq p_k,log_2p_{kv}\geq log_2p_k,放缩如下$$

$$\begin{aligned}-\sum_{v=1}^{V}\frac{|D^v|}{|D|}Ent(D^v) &= \sum_{v=1}^{V}\frac{|D^v|}{|D|}\sum_{k=1}^{|\mathcal{Y}|}p_{kv}log_2p_{kv}\\ &= \sum_{k=1}^{|\mathcal{Y}|}\sum_{v=1}^{V}\frac{|D^v|}{|D|}p_{kv}log_2p_{kv}\\ &\geq \sum_{k=1}^{|\mathcal{Y}|}\sum_{v=1}^{V}\frac{|D^v|}{|D|}p_{kv}log_2p_k\\ &= \sum_{k=1}^{|\mathcal{Y}|}log_2p_k\sum_{v=1}^{V}\frac{|D^v|}{|D|}p_{kv}\\ &= \sum_{k=1}^{|\mathcal{Y}|}p_klog_2p_k=-Ent(D)\end{aligned}$$

$$所以Gain(D,a)=\operatorname{Ent}(D)-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\operatorname{Ent}\left(D^v\right)\geq 0$$

二、划分

(1)信息增益

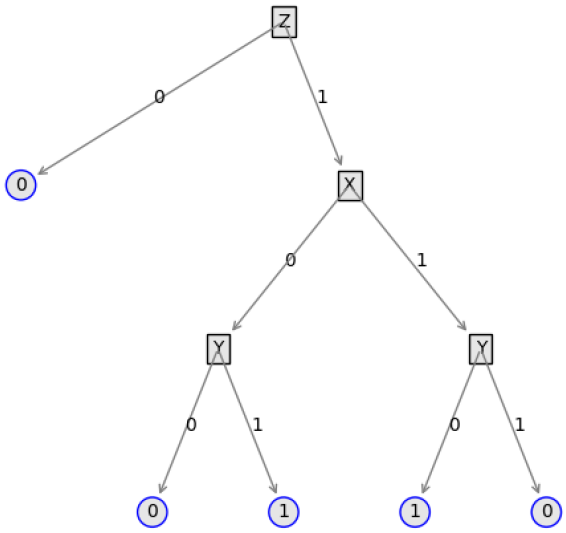
本题主要展现决策树在不同划分标准下划分的具体计算过程. 假设一个包含三个布尔属性 X, Y, Z 的属性空间, 目标函数 $f = f(X, Y, Z)$ 作为标记空间, 它们形成的数据集如1所示.

1. 请使用信息增益作为划分准则画出决策树的生成过程. 当两个属性信息增益相同时, 依据字母顺序选择属性.

通过python计算出每个结点信息增益如下：（用字典树的形式暂存决策树并输出）

```
node  1 :  X  :  0.0 Y  :  0.0 Z  :  0.31127812445913283
node  2 :  X  :  0.0 Y  :  0.0
node  3 :  Y  :  1.0
node  4 :  Y  :  1.0
{'Z': {0: '0', 1: {'X': {0: {'Y': {0: '0', 1: '1'}}, 1: {'Y': {0: '1', 1: '0'}}}}}}
```

再利用matlab将上述字典树可视化：



(2)基尼指数

编号	X	Y	Z	f	编号	X	Y	Z	f
1	1	0	1	1	5	0	1	0	0
2	1	1	0	0	6	0	0	1	0
3	0	0	0	0	7	1	0	0	0
4	0	1	1	1	8	1	1	1	0

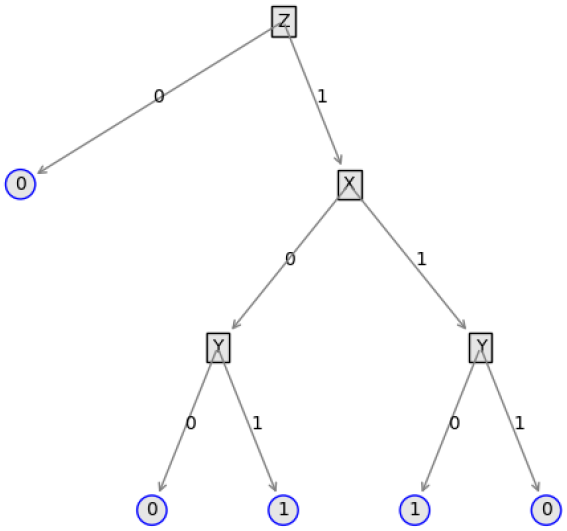
Table 1: 布尔运算样例表

2. 请使用基尼指数作为划分准则画出决策树的生成过程, 当两个属性基尼指数相同时, 依据字母顺序选择属性.

通过python计算出每个结点基尼指数如下：（用字典树的形式暂存决策树并输出）

```
node  1 :  X  :  0.375 Y  :  0.375 Z  :  0.25
node  2 :  X  :  0.5 Y  :  0.5
node  3 :  Y  :  0.0
node  4 :  Y  :  0.0
{'Z': {0: '0', 1: {'X': {0: {'Y': {0: '0', 1: '1'}}, 1: {'Y': {0: '1', 1: '0'}}}}}}
```

再利用matlab将上述字典树可视化：



三、剪枝

(1)验证

教材 4.3 节介绍了决策树剪枝相关内容, 给定包含 5 个样例的人造数据集如表3a所示, 其中“爱运动”、“爱学习”是属性, “成绩高”是标记. 验证集如表3b所示. 使用信息增益为划分准则产生如图1所示的两棵决策树. 请回答以下问题:

(a) 训练集

编号	爱运动	爱学习	成绩高
1	是	是	是
2	否	是	是
3	是	否	否
4	是	否	否
5	否	否	是

(b) 验证集

编号	爱运动	爱学习	成绩高
6	是	是	是
7	否	是	否
8	是	否	否
9	否	否	否

Table 2: 人造数据集

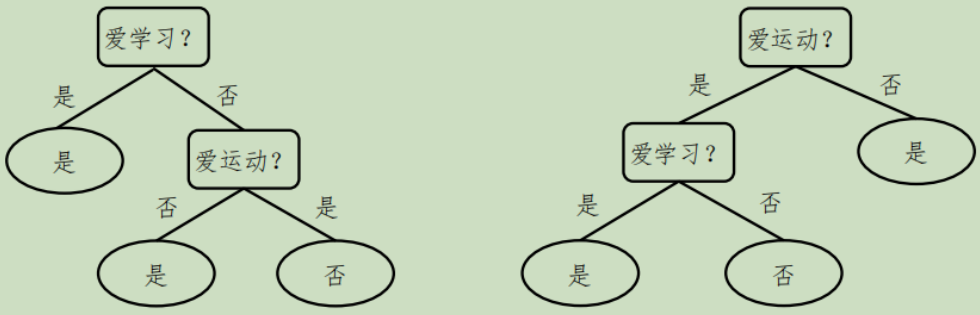


Figure 1: 人造数据决策树结果

1. 请验证这两棵决策树的产生过程.

由于第一个结点”爱学习“和”爱运动“的信息增益都是0.41997，所以会产生两个决策树分别以这两个属性进行第一次划分

左侧决策树第二层的"爱运动"和右侧决策树第二层的"爱学习"信息增益都是0.918

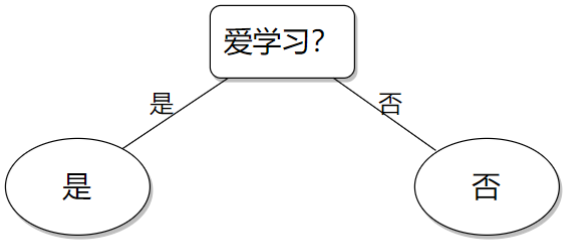
```
node  1 :  爱运动  :  0.4199730940219749 爱学习  :  0.4199730940219749
node  2 :  爱学习  :  0.9182958340544896
{'爱运动': {'是': {'爱学习': {'是': '是', '否': '否'}}, '否': '是'}}
```

(2)剪枝

2. 对图1的结果基于该验证集进行预剪枝、后剪枝，给出剪枝后的决策树.

1.左树预剪枝

- node 1"爱学习"：划分
  - 若不剪枝每个样本都被标记为是，只有样例6正确，准确率25%
  - 若剪枝，只有样例7错误，准确率75%
- node 2"爱运动"：禁止划分
  - 若不剪枝即保留现状，准确率75%
  - 若剪枝，样例9错误，准确率下降到50%

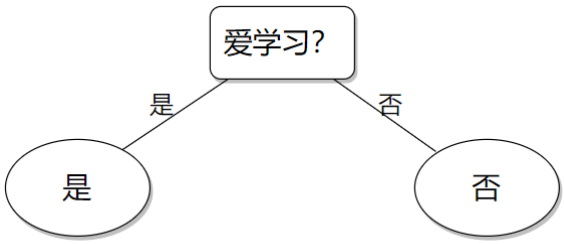


2.右树预剪枝

- node 1"爱运动"：禁止划分
  - 若不剪枝每个样本都被标记为是，只有样例6正确，准确率25%
  - 若剪枝，只有样例8正确，准确率不变，还是25%

3.左树后剪枝

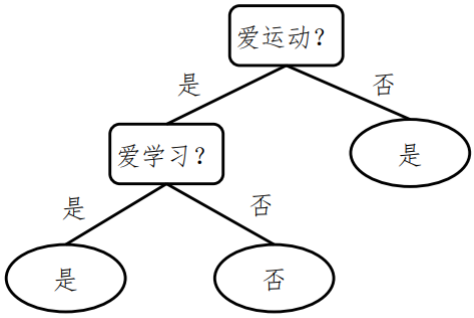
- node 1"爱运动"：剪枝
  - 若不剪枝，样例9错误，准确率50%
  - 若剪枝，只有样例7错误，准确率75%
- node 2"爱学习"：禁止剪枝
  - 若不剪枝，只有样例7错误，准确率75%
  - 若剪枝，只有样例6正确，准确率25%



4.右树后剪枝

都要剪，不能只剪一个

- node 1"爱学习"：禁止剪枝
  - 若不剪枝，准确率50%
  - 若剪枝，无树，准确率25%
- node 2"爱运动"：禁止剪枝
  - 若不剪枝，准确率50%
  - 若剪枝，准确率25%



(3)分析

左树：

- 预剪枝和后剪枝的决策树相同，没有任何差别
- 训练集80%，验证集75%

右树：

- 预剪枝训练集60%，测试集25%
- 后剪枝训练集100%，测试集50%

综上所述，后剪枝的拟合能力更强

四、连续&缺失

(1)连续

考虑如表 4所示数据集，仅包含一个连续属性，请给出将该属性“数字”作为划分标准时的决策树划分结果。

属性	类别
3	正
4	负
6	负
9	正

该属性候选划分点集合$T=\{3.5,5,7.5\}$

$$Ent(D)=-2*\frac{1}{2}log_2\frac{1}{2}=1$$

第一层

$$\begin{cases}Gain(D,a,3.5)=1-\frac{3}{4}(-\frac{2}{3}log_2\frac{2}{3}-\frac{1}{3}log_2\frac{1}{3})=0.3112781244591328\\Gain(D,a,5)=0\\Gain(D,a,7.5)=1-\frac{3}{4}(-\frac{2}{3}log_2\frac{2}{3}-\frac{1}{3}log_2\frac{1}{3})=0.3112781244591328\end{cases}$$

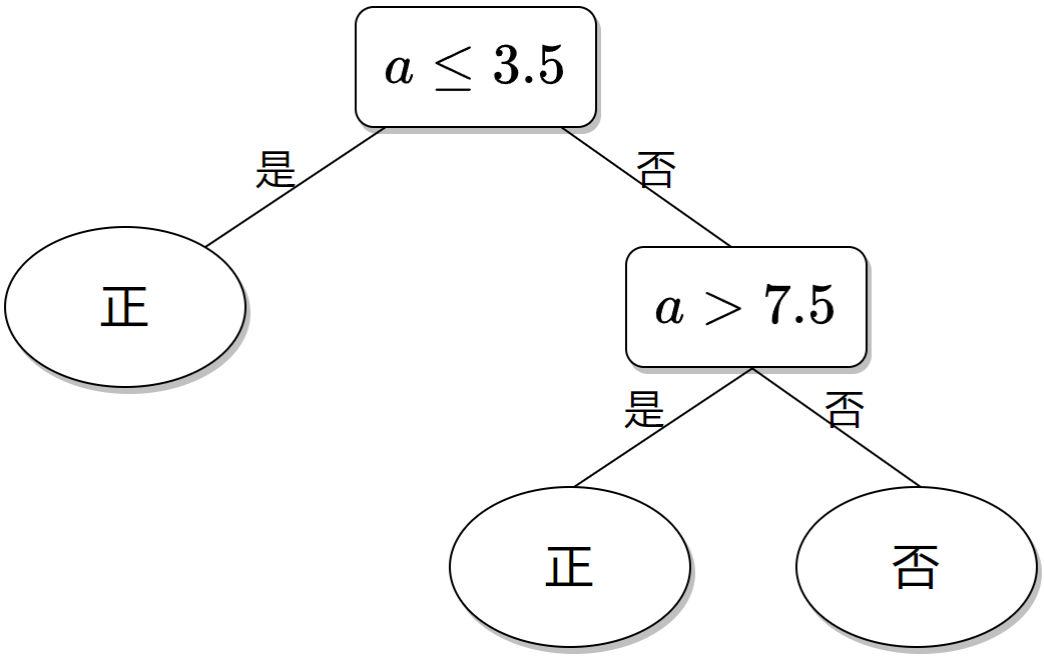
因此第一层选择3.5作为划分

第二层

$$Ent(D')=-\frac{1}{3}log_2\frac{1}{3}-\frac{2}{3}log_2\frac{2}{3}=0.9182958340544897$$

$$\begin{cases}Gain(D',a,5)=Ent(D')-\frac{2}{3}(-\frac{1}{2}log_2\frac{1}{2}-\frac{1}{2}log_2\frac{1}{2})=0.25162916738782304\\Gain(D',a,7.5)=Ent(D')-\frac{3}{4}(-\frac{2}{3}log_2\frac{2}{3}-\frac{1}{3}log_2\frac{1}{3})=0.9182958340544897\end{cases}$$

因此第二层选择7.5作为划分



(2)缺失

2. 请阐述决策树如何处理训练时存在缺失值的情况，具体如下：考虑表 1的数据集，如果发生部分缺失，变成如表 5所示数据集（假设 X, Y, Z 只有 0 和 1 两种取值）. 在这种情况下，请考虑如何处理数据中的缺失值，并结合问题 二第 1 小问的答案进行对比，论述方法的特点以及是否有局限性。

X	Y	Z	f
1	0	-	1
-	1	0	0
0	-	0	0
0	1	1	1
-	1	0	0
0	0	-	0
1	-	0	0
1	1	1	0

Table 5: 缺失数据集

- **使用python计算划分属性选择**

$$\begin{aligned}\mathrm{Gain}(D,a) &= \rho \times \mathrm{Gain}(\tilde{D},a) \\ &= \rho \times \left(\mathrm{Ent}(\tilde{D}) - \sum_{v=1}^{V} \tilde{r}_v \,\mathrm{Ent}\left(\tilde{D}^v\right)\right)\end{aligned}$$

- **更改样本比例的计算方式，从个数比改为权重`w`比，得到第一层信息增益如下：**

```
node  1 :  X  :  0.0 Y  :  0.03308281331130081 Z  :  0.23751681623626558
```

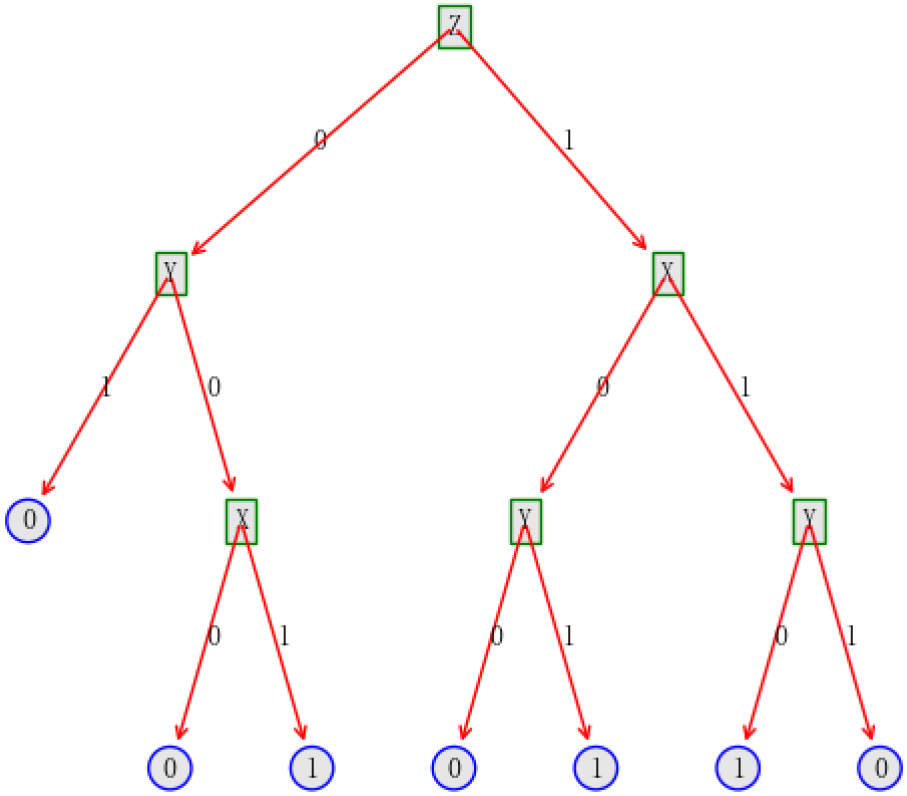
- **所以将Z作为第一个划分属性，后续结点同理，具体信息增益如下：(结点按递归顺序排列)**

```
node  2 :  X  :  0.1477829985375175 Y  :  0.2012050593046014
node  3 :  X  :  0.3705065005495053
node  4 :  X  :  0.1887218755408673 Y  :  1.1102230246251565e-16
node  5 :  Y  :  0.8112781244591328
node  6 :  Y  :  0.8112781244591328
```

- **因此字典决策树为：**

```
{'Z': {0: {'Y': {1: '0', 0: {'X': {0: '0', 1: '1'}}}}, 1: {'X': {0: {'Y': {0: '0', 1: '1'}}, 1: {'Y': {0: '1', 1: '0'}}}}}}
```

- **向python输入数据集，输出可视化决策树如下：**



对比分析：

- 与无缺失值的方法相比，划分属性时首先需要排除含缺失值的样本，而后缺失值样本需要按权重分到不同分支中，计算量和复杂程度略有增加
- 这个决策树比原来的更为复杂，Z=0时仍然向下展开分支

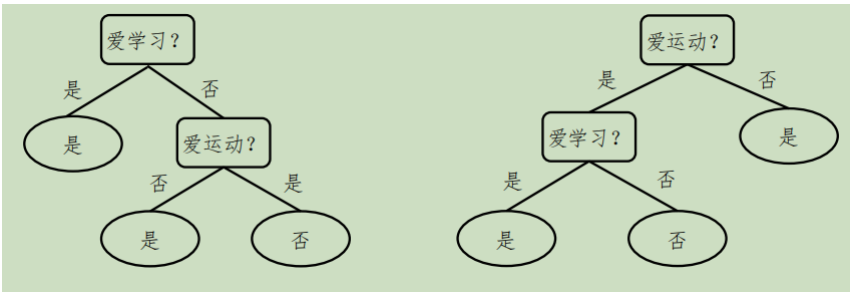
局限性：

- 由于对缺失值的处理只能是按权重分摊，但这不符合事实，导致了一些分支出现了不该有的结点，导致了决策树更加复杂，过拟合风险增大

(3)输出标签

3. 请阐述决策树如何处理测试时存在缺失值的情况，具体如下：对于问题二训练出的决策树，考虑表6所示的含有缺失值的测试集，输出其标签，并论述方法的特点以及是否有局限性。

编号	爱运动	爱学习	成绩高
6	是	-	
7	-	是	
8	否	-	
9	-	否	



（使用问题三左边的决策树作为本题计算依据）

方法：

对于缺失值，按权重分摊到各分支，得到概率

权重从问题三训练集获得：

第二层结点中“是”40%，“爱运动？”60%

第三层结点中“是”33%，“否”67%

输出标签：

编号6：是：$40\% + 60\% \times 0 = 40\%$，否：$60\%$，所以否

编号7：是

编号8：是：$40\% + 60\% \times 1 = 100\%$，所以是

编号9：是：33%，否：67%，所以否

方法特点：

即使有缺失值也能预测标签，但只有概率，准确率较低

局限性：

只有概率准确率低，且如果各分支概率接近，出错概率极大上升

五、多变量决策树

考虑如下包含 10 个样本的数据集, 每一列表示一个样本, 每个样本具有二个属性, 即 $\boldsymbol{x}_i = (x_{i1};\ x_{i2})$.

编号	1	2	3	4	5	6	7	8	9	10
A_1	24	53	23	25	32	52	22	43	52	48
A_2	40	52	25	77	48	110	38	44	27	65
标记	1	0	0	1	1	1	1	0	0	1

1. 计算根结点的熵;
2. 构建分类决策树, 描述分类规则和分类误差;
3. 根据 $\alpha x_1 + \beta x_2 - 1$, 构建多变量决策树, 描述树的深度以及 α 和 β 的值.

(1)根节点熵

$$Ent(D) = -\left(0.4\log_2 0.4 + 0.6\log_2 0.6\right) = 0.97$$

(2)连续值

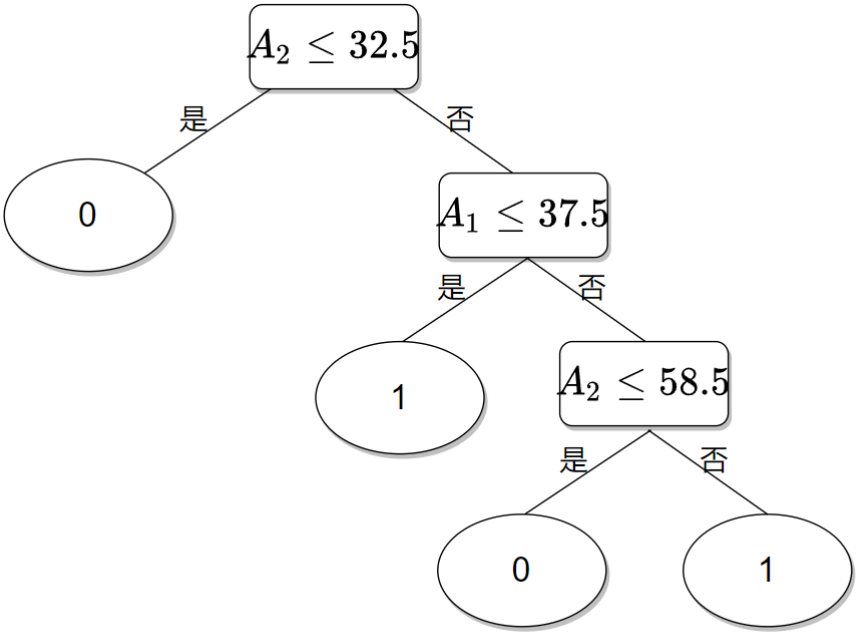
属性A_1候选划分点集合$T = \{22.5, 23.5, 24.5, 28.5, 37.5, 45.5, 50, 52, 52.5\}$

属性A_1最优化划分值52，信息增益0.14

属性A_2候选划分点集合$T = \{26, 32.5, 39, 42, 46, 50, 58.5, 71, 93.5\}$

属性A_2最优化划分值32.5，信息增益0.32

所以选择A_2的32.5作为第一层划分，后续同理



分类误差为0，因为没有冲突样本

(3)多变量决策树

分析：

不妨假设两层的决策树可以满足要求，先看看能否实现

令$f(d) = \alpha x_1 + \beta x_2 - 1$，则对于标记不同的两个样本$d_1, d_2$，一定有$f(d_1) \times f(d_2) \leq 0$

几何意义：

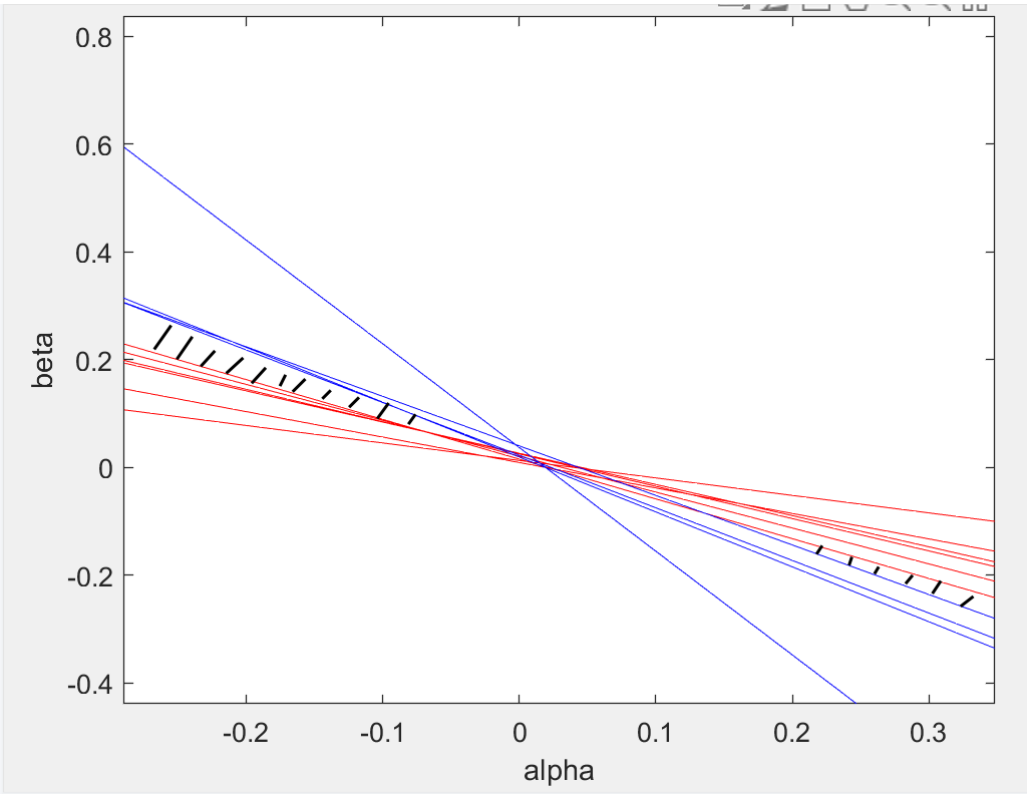
把`(α,β)`看作是直角坐标系中的一个点，把10个样本对应的函数$f(d_i)$当作是对点`(α,β)`的线性规划，本题的目的就是要找到`(α,β)`的位置满足所有样本的限制。如图：

其中标记为1的样本对应红线

其中标记为0的样本对应蓝线

直观上讲，所有满足条件的`(α,β)`（即能经过一次划分就区分出两种标记）一定满足所有红线都在点的一侧，所有蓝线都在另一侧

阴影区域即为所有`(α,β)`的解



在阴影区域任取一个点 `(0.3,-0.21)` 作为 `(α,β)` 的解，得到多变量决策树：