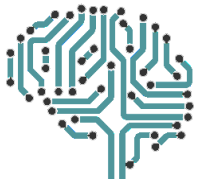
暨南大學
JINAN UNIVERSITY



机 器 学 习

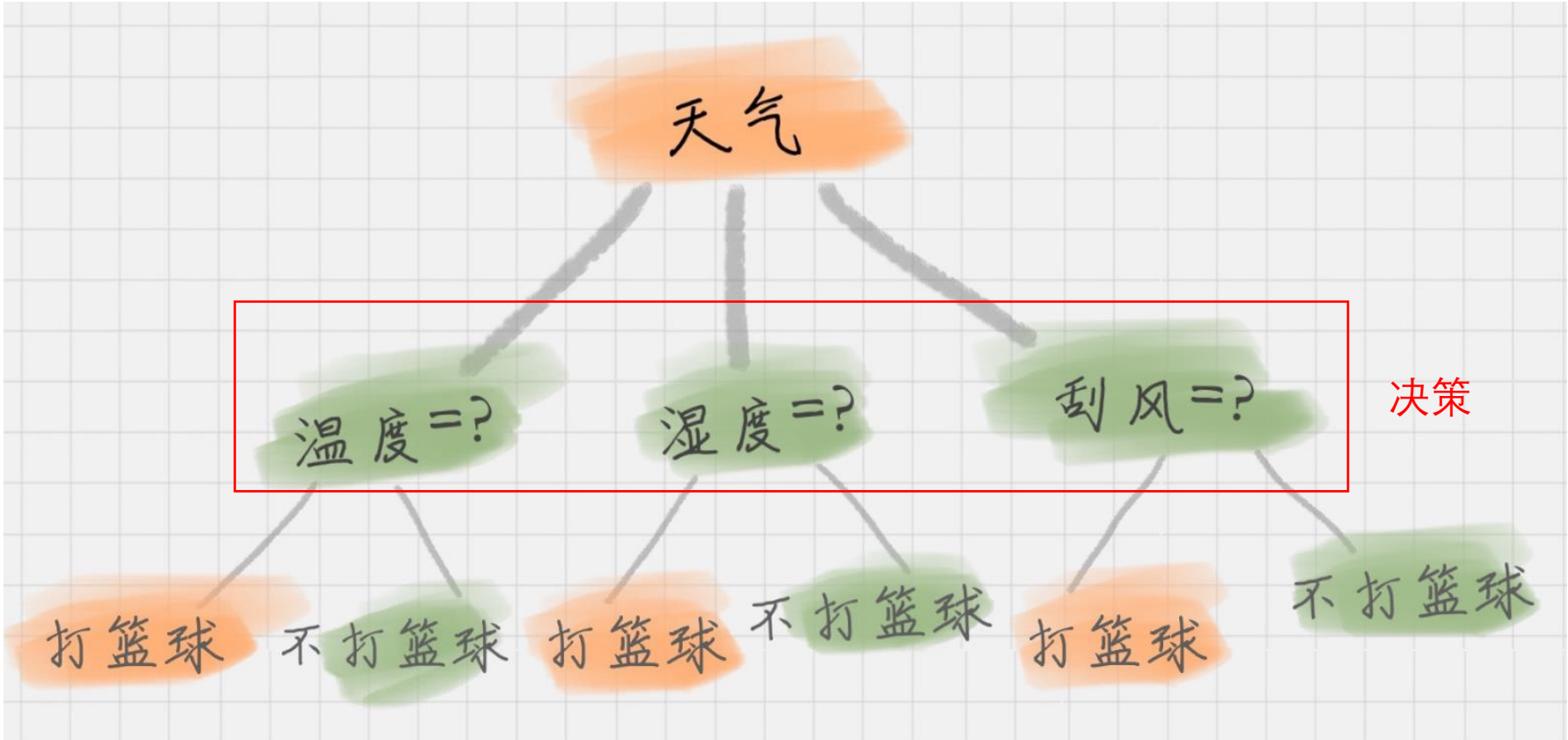
第六章：决策树

黄斐然

2022/3/28

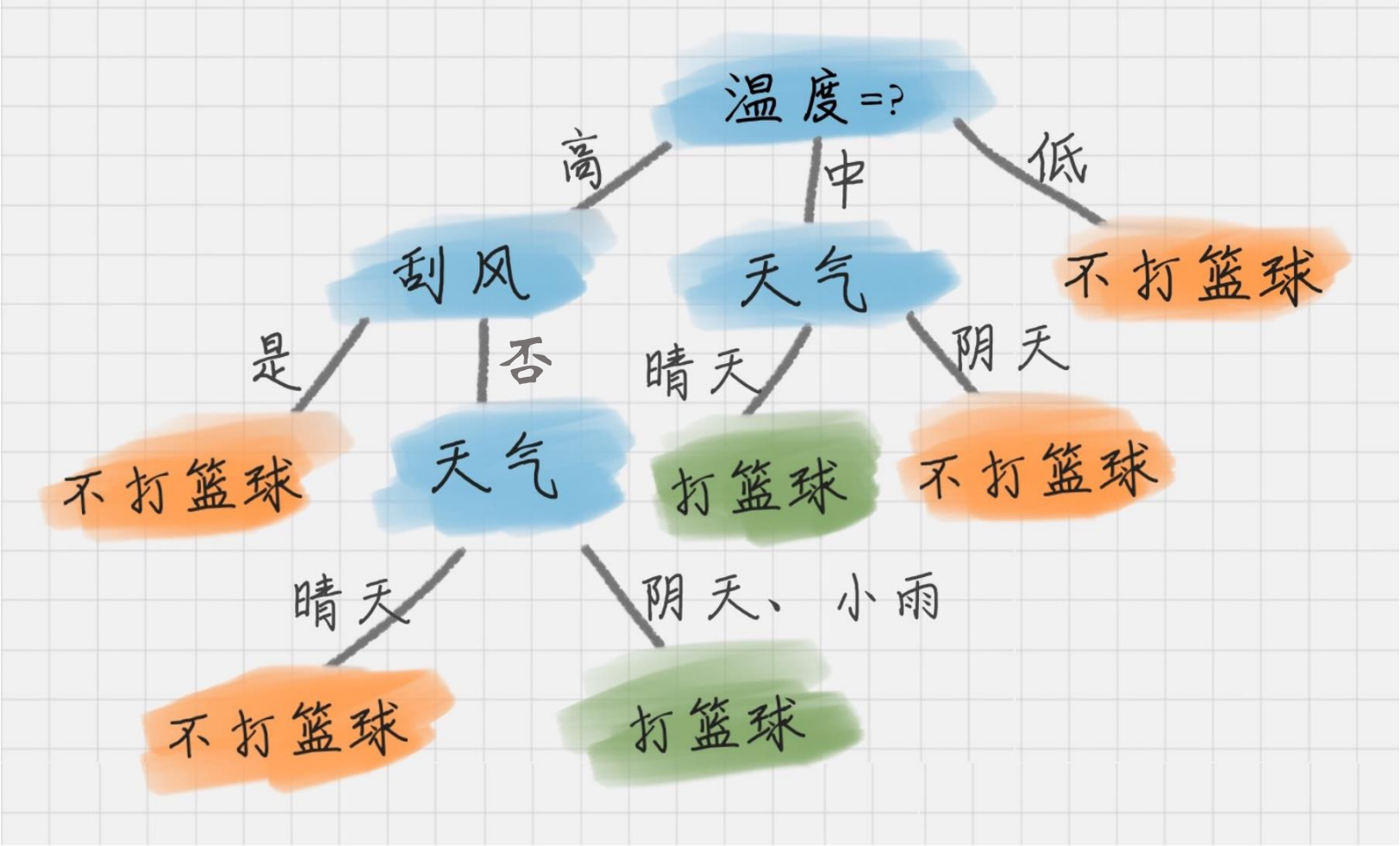
一个问题

- 去不去打篮球?



一个问题

- 去不去打篮球?



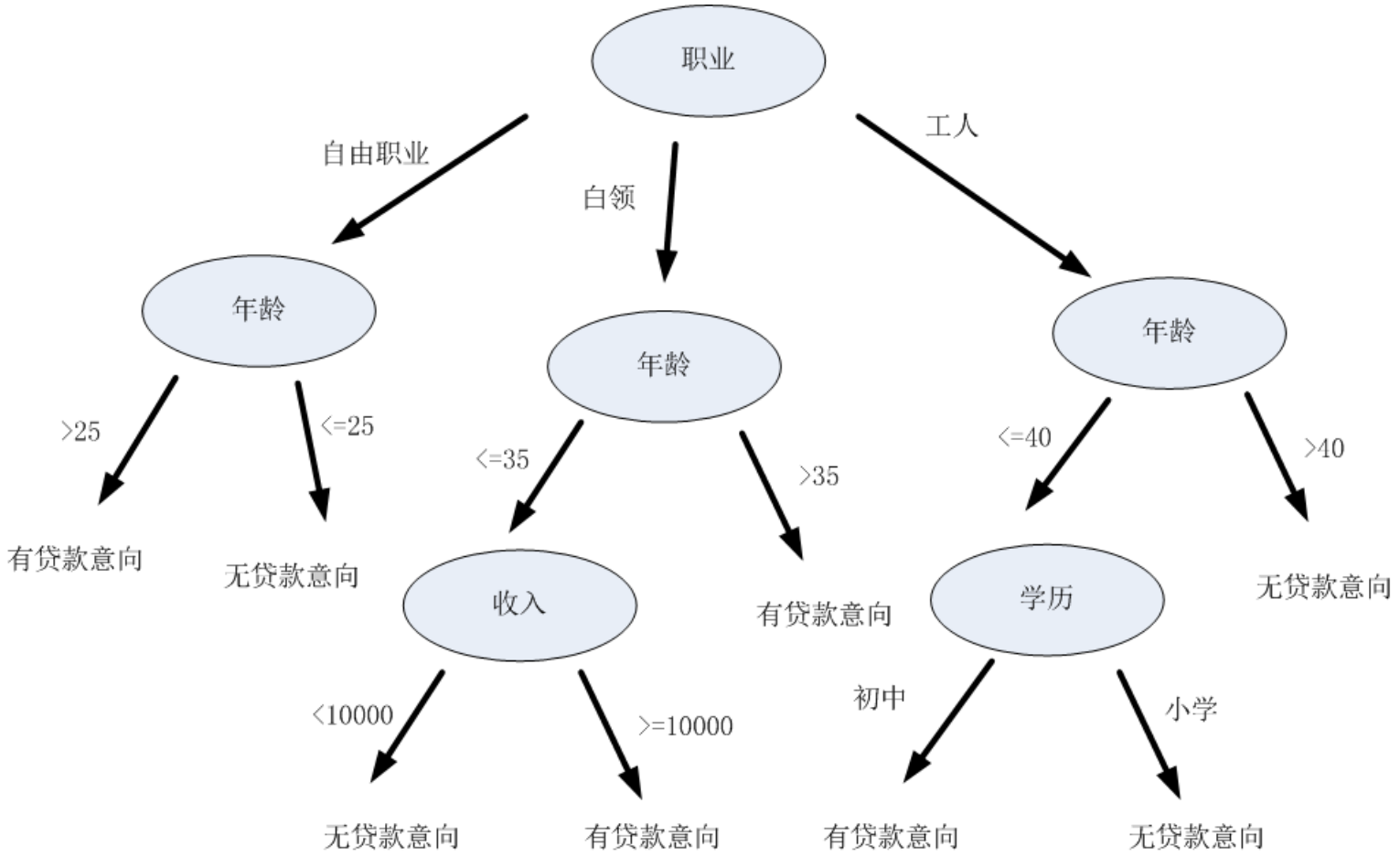
决策树是什么

- 决策树是最基本的分类方法，也可以用于回归。
- 决策树模型呈树形结构。
- 在分类问题中，表示基于特征对实例进行分类的过程，它可以认为是if-then规则的集合。在决策树的结构中，每一个实例都被一条路径或者一条规则所覆盖。
- 通常决策树学习包括三个步骤：特征选择、决策树的生成和决策树的修剪。

是否贷款？

职业	年龄	性别	收入	学历	是否贷款
自由职业	28	男	5000	高中	是
工人	36	女	5500	高中	否
工人	42	男	2800	初中	是
白领	45	女	3300	小学	是
白领	25	男	10000	初中	是
白领	32	女	8000	硕士	是
白领	28	女	13000	硕士	是
自由职业	21	男	4000	本科	否
自由职业	22	女	3200	初中	否
工人	33	男	3000	高中	否
工人	48	女	4200	初中	否

是否贷款？


职业
自由职业
白领
工人
年龄
年龄
年龄
>25
<=25
<=35
>35
<=40
>40
有贷款意向
无贷款意向
收入
有贷款意向
学历
无贷款意向
<10000
>=10000
初中
小学
无贷款意向
有贷款意向
有贷款意向
无贷款意向

1、特征选择

- 特征选择在于选取对训练数据具有分类能力的特征。
- 如果利用一个特征进行分类的结果与随机分类的结果没有很大差别，则称这个特征是没有判别力的。这样的特征对分类器学习的作用很小。反之，如果一个特征对分类的结果影响很大，则称这个特征是有判别力的（discriminative）。
- 通常特征选择的准则是信息增益。

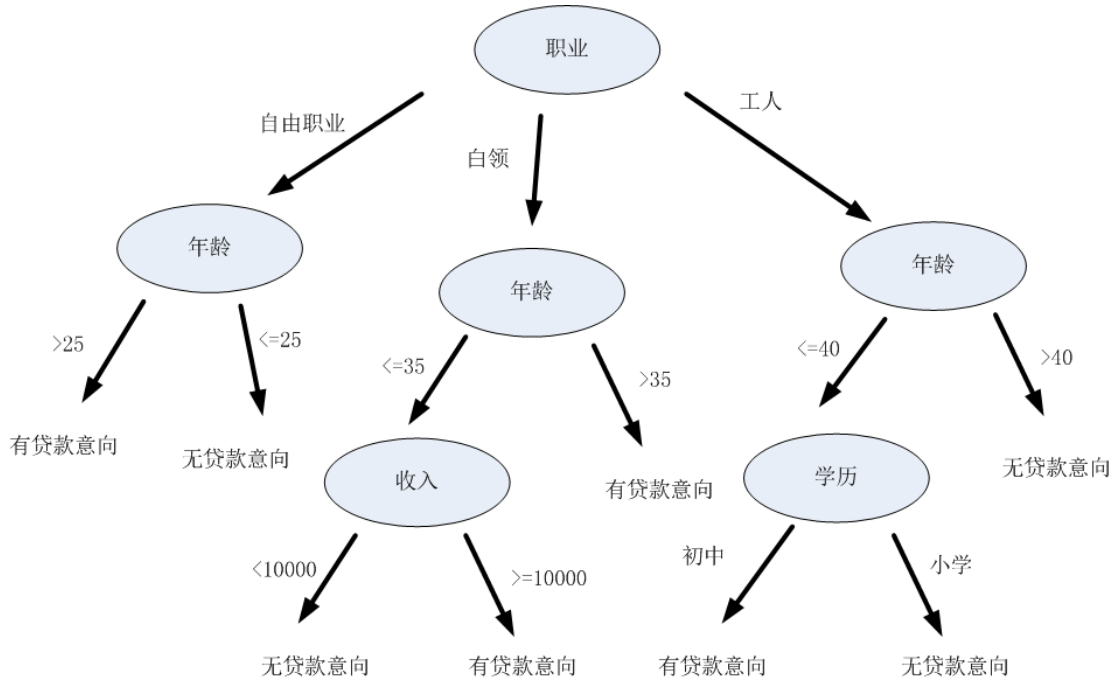
1、特征选择

- 没有判别力的特征：
  - 性别
- 有判别力的特征：
  - 收入
  - 学历

性别	收入	学历	是否贷款
男	5000	高中	是
女	5500	高中	否
男	2800	初中	是
女	3300	小学	是
男	10000	初中	是
女	8000	硕士	是
女	13000	硕士	是
男	4000	本科	否
女	3200	初中	否
男	3000	高中	否
女	4200	初中	否

1、特征选择

- 如果一个特征具有更好的分类能力，那么就更应该选择这个特征。信息增益就能很好的表示这一直观的准则。



信息量怎么衡量


07月31日 周二 农历六月十九
32 ℃
多云(实时)
28 ~ 35℃
多云
西北风3-4级
51 良
周三
08月01日
28 ~ 34℃
阵雨
北风3-4级
优
周四
08月02日
27 ~ 34℃
阵雨转多云
北风3-4级
优
周五
08月03日
28 ~ 32℃
小雨转晴
东南风3级
优
周六
08月04日
28 ~ 35℃
小雨转晴
东南风3-4级
优


22:57
华北制药
8.80
成交 86.04万手
时间 15:04
分时
五日
日K
周K
月K
MA5 10.60
MA10 10.23
MA20 9.06
13.55
11.15
8.75
6.35
3.95
13.0
4.48
2013-11
2014-05
2014-11
2015-05
406.9
万手
不复权
前复权
后复权
成交量
MACD
DMI

信息熵

信息熵

- 信息是个很抽象的概念。人们常常说信息很多，或者信息较少，但却很难说清楚信息到底有多少。比如一本五十万字的中文书到底有多少信息量。
- 直到1948年，香农提出了“信息熵”的概念，才解决了对信息的量化度量问题。信息熵这个词是C.E.Shannon（香农）从热力学中借用过来的。热力学中的热熵是表示分子状态混乱程度的物理量。香农用信息熵的概念来描述信源的不确定度。

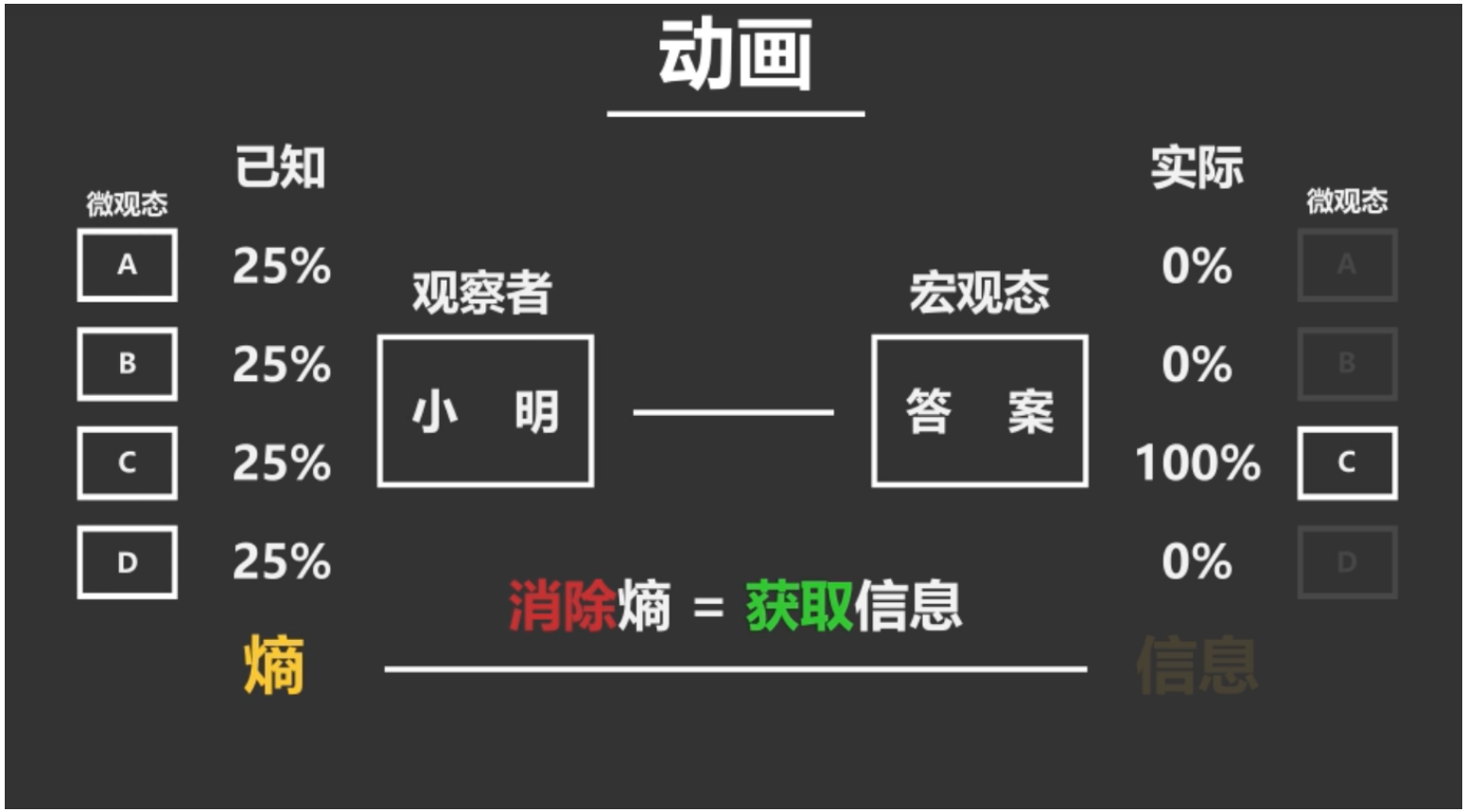
信息熵

熵

- 熵反映的是一个系统的混乱程度，一个系统越混乱，其熵就越大；越是整齐，熵就越小。

信息熵

- 信息熵



- 需要引入消除不确定性的信息量越多，则信息熵越高，反之则越低。

信息熵

信息熵的计算

- 假设训练数据集为D，|D|表示样本个数。设有K个类，每个类表示成 C_k，k=1,2,3,4...k, $|C_k|$为属于类C_k的样本个数，则样本数为：

$$\sum_{k=1}^{K} = |D|$$

- 对于信息熵来说，计算如下：

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log \frac{|C_k|}{|D|}$$

信息熵

信息熵

$$H(X) = -\sum_{x \in \chi} p(x) \log p(x)$$

- 示例1：x={0,1}, p(0) = 0.5, p(1) = 0.5

  H(X) = -(0.5*log(0.5)+ 0.5*log(0.5)) = log2 → 熵大，不确定性大，信息量少

- 示例2：x={0,1}, p(0) = 0, p(1) = 1

  H(X) = -(0*log(0)+ 1*log(1)) = 0 → 熵小，不确定性小，信息量大

信息熵的计算

- 类的个数 K=2，类k可以为{0,1}，样本数 |D| = 11

- 不贷款的类共有样本数：$|C_0|$=5。贷款的类共有样本数： $|C_1|$=6。

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log \frac{|C_k|}{|D|}$$

= - (5/11 * log2(5/11) + 6/11 * log2(6/11))

= 0.994

是否贷款
是
否
是
是
是
是
是
否
否
否
否

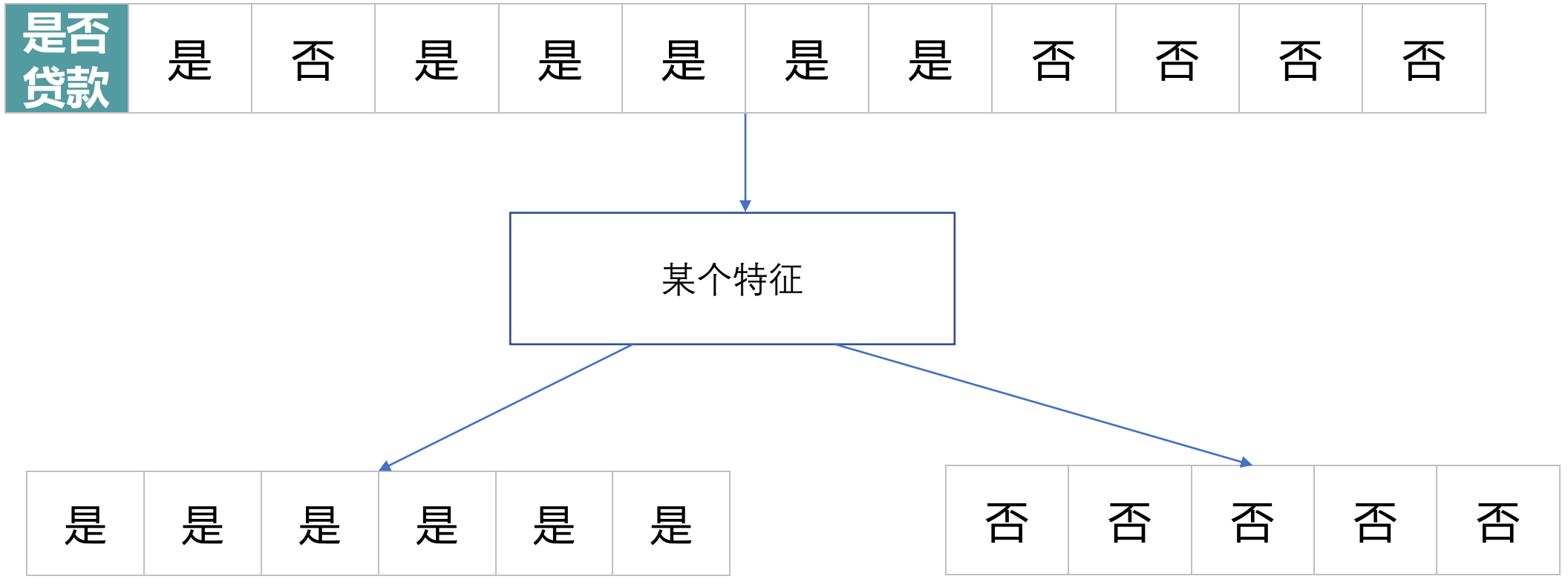
信息增益

信息增益

- 决策树的过程其实是在寻找某一个特征对整个分类结果的不确定减少的过程。
- 信息增益（information gain）表示得知特征X的信息而是的类Y的信息的不确定性减少的程度，所以我们对于选择特征进行分类的时候，当然选择信息增益较大的特征，这样具有较强的分类能力。

信息增益

- 最理想情况



信息增益

信息增益计算

- —特征A对训练数据集D的信息增益g(D,A)，定义为集合D的信息熵(也叫经验熵)H(D)与特征A给定条件下D的信息条件熵H(D|A)之差，即公式为：

$$g(D,A) = H(D) - H(D|A)$$

条件熵的计算

- 设特征A有n个不同的取值$\{a_1,a_2,...,a_n\}$,根据特征A的取值将D划分为n个子集D_1，D_2，...，D_n，$|D_i|$为样本个数，其中D_i中属于C_k类的样本的集合为D_{ik}，那么条件熵计算如下：

$$H(D|A)=\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\sum_{k=1}^{K}\frac{|D_{ik}|}{|D_i|}\log\frac{|D_{ik}|}{|D_i|}$$

特征A	是否贷款
大于	是
小于等于	否
大于	是
大于	是
大于	是
大于	是
大于	是
小于等于	否
小于等于	否
小于等于	否
小于等于	否

条件熵的计算

- 最理想有n个不同的取值{a,b},根据特征A的取值将D划分为n个子集D_1，D_2，…，D_n，$|D_i|$为样本个数，其中D_i中属于C_k类的样本的集合为D_{ik}，那么条件熵计算如下：

$$H(D|A)=\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\sum_{k=1}^{K}\frac{|D_{ik}|}{|D_i|}\log\frac{|D_{ik}|}{|D_i|}$$

= - 6/11* (6/6 * log(6/6) + 0/6 * log(0/6))

- 5/11* (0/5 * log(0/5) + 5/5 * log(5/5))

= 0

最理想特征	是否贷款
a	是
b	否
a	是
a	是
a	是
a	是
a	是
b	否
b	否
b	否
b	否

条件熵的计算——使用性别特征

- 特征A性别有2个不同的取值{男，女}，其中D_0中属于C_0（男）类的样本数为D_{00}=2，其中D_0中属于C_1（女）类的样本数为D_{01}=3； D_1中属于C_0（男）类的样本数为D_{10}=3，其中D_1中属于C_1（女）类的样本数为D_{11}=3；那么条件熵计算如下：

$$H(D|A)=\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\sum_{k=1}^{K}\frac{|D_{ik}|}{|D_i|}\log\frac{|D_{ik}|}{|D_i|}$$

= - 5/11* (2/5 * log(2/5) + 3/5 * log(3/5))

- 6/11* (3/6 * log(3/6) + 3/6 * log(3/6)) = 0.987

性别	是否贷款
男	是
女	否
男	是
女	是
男	是
女	是
女	是
男	否
女	否
男	否
女	否

条件熵的计算——使用收入特征

- 设特征A收入有2个不同的取值{>=5000，<5000}，其中D_0中属于C_0（>=5000）类的样本数为D_{00}=1，其中D_0中属于C_1（<5000）类的样本数为D_{01}=4； D_1中属于C_0（>=5000）类的样本数为D_{10}=4，其中D_1中属于C_1（<5000）类的样本数为D_{11}=2；那么条件熵计算如下：

$$H(D|A)=\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\sum_{k=1}^{K}\frac{|D_{ik}|}{|D_i|}\log\frac{|D_{ik}|}{|D_i|}$$

= - 5/11* (1/5 * log(1/5) + 4/5 * log(4/5))

- 6/11* (4/6 * log(4/6) + 2/6 * log(2/6)) = 0.829

收入	是否贷款
5000	是
5500	否
2800	是
3300	是
10000	是
8000	是
13000	是
4000	否
3200	否
3000	否
4200	否

1、信息增益

- 信息增益的计算：
  - 1、若使用性别特征，则：

$$g(D,A) = H(D) - H(D|A)$$

$$= 0.994 - 0.987 = 0.007$$

  - 2、若使用收入特征，则：

$$g(D,A) = H(D) - H(D|A)$$

$$= 0.994 - 0.829 = 0.165$$

  - 在这两个特征挑选，应选择收入特征。

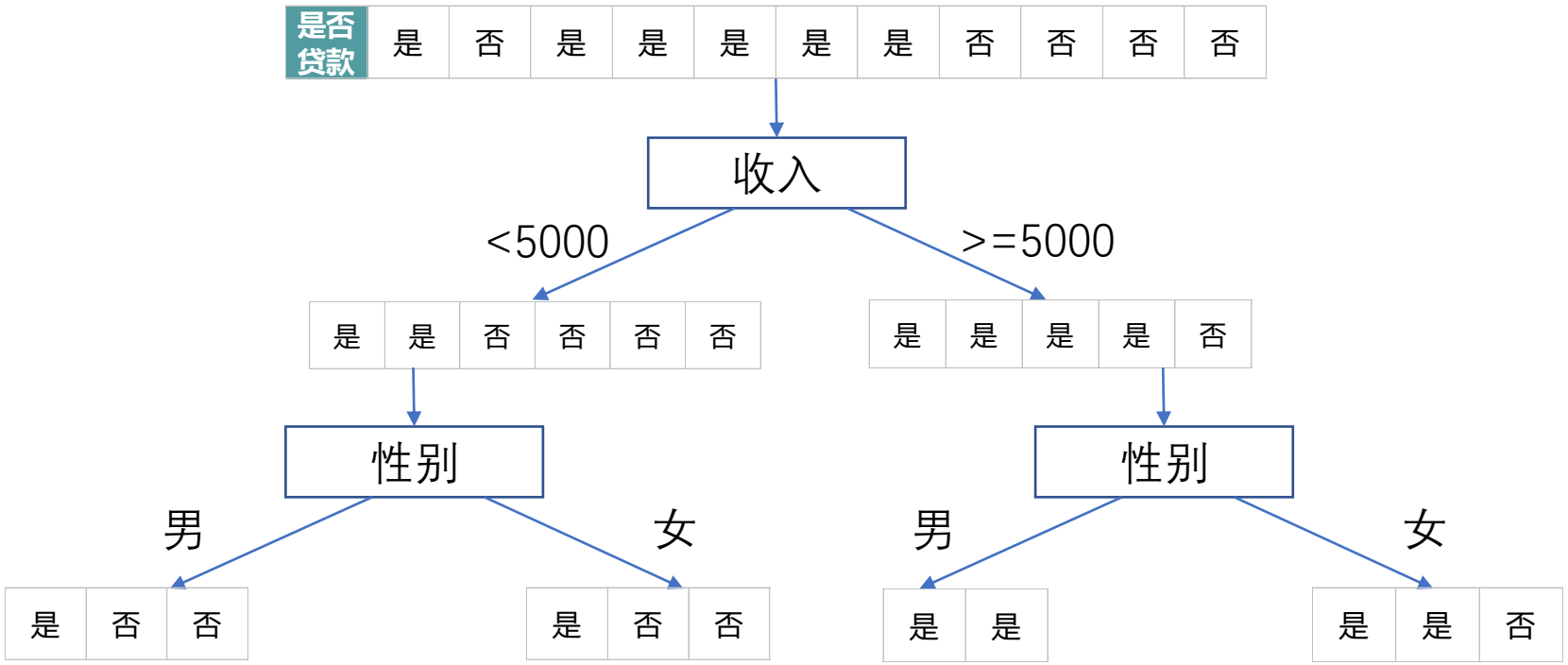
性别	收入	是否贷款
男	5000	是
女	5500	否
男	2800	是
女	3300	是
男	10000	是
女	8000	是
女	13000	是
男	4000	否
女	3200	否
男	3000	否
女	4200	否

2、构建决策树

- 构建决策树：
  - 步骤1：将所有的数据看成是一个节点，进入步骤2；
  - 步骤2：从所有的数据特征中挑选使得信息增益最大的数据特征对节点进行分割，进入步骤3；
  - 步骤3：生成若干孩子节点，对每一个孩子节点进行判断，如果满足停止分裂的条件，进入步骤4；否则，进入步骤2；
  - 步骤4：设置该节点是子节点，其输出的结果为该节点数量占比最大的类别。

2、构建决策树

- 构建决策树：



第二层节点信息增益

- 左子树：

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log \frac{|C_k|}{|D|}$$

= - (2/6 * log2(2/6) + 4/6 * log2(4/6))

= 0.918

$$H(D|A) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log \frac{|D_{ik}|}{|D_i|}$$

= - 3/6* (1/3 * log(1/3) + 2/3 * log(2/3))

- 3/6* (1/3 * log(1/3) + 2/3 * log(2/3)) = 0.918

性别	是否贷款
男	是
女	是
男	否
女	否
男	否
女	否

性别	是否贷款
男	是
女	是
男	否
女	否
男	否
女	否

第二层节点信息增益

- 右子树：

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log \frac{|C_k|}{|D|}$$

= - (4/5 * log2(4/5) + 1/5 * log2(1/5))

= 0.722

性别	是否贷款
男	是
女	否
男	是
女	是
女	是

$$H(D|A) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log \frac{|D_{ik}|}{|D_i|}$$

= - 2/5* (2/2 * log(2/2) + 0/2 * log(0/2))

- 3/5* (2/3 * log(2/3) + 1/3 * log(1/3)) = 0.551

性别	是否贷款
男	是
女	否
男	是
女	是
女	是

构建决策树的三个重要问题

- （1）数据如何分割
- （2）如何选择分裂的属性
- （3）什么时候停止分裂

数据如何分割

- 分裂属性的数据类型分为离散型和连续性两种情况
  - 对于离散型的数据，按照属性值进行分裂，每个属性值对应一个分裂节点；
  - 对于连续性的数据，一般性的做法是对数据按照该属性进行排序，再将数据分成若干区间，如[0,10]、[10,20]、[20,30]…，一个区间对应一个节点，若数据的属性值落入某一区间则该数据就属于其对应的节点。

如何选择分裂的属性

- 决策树采用贪婪思想进行分裂，最理想的情况是能找到一个属性刚好能够将不同类别分开，但是大多数情况下分裂很难一步到位，我们希望每一次分裂之后孩子节点的数据尽量”纯”。
- 选择分裂属性是要找出能够使所有孩子节点数据最纯的属性，决策树使用信息增益、或信息增益率、或GINI系数作为选择属性的依据。

什么时候停止分裂

- 一般情况下为了降低决策树复杂度和提高预测的经度，会适当提前终止节点的分裂。
  - （1）所有特征已经使用完毕，不能继续进行分裂。
    - 被动式停止分裂的条件，当已经没有可分的属性时，直接将当前节点设置为叶子节点。
  - （2）熵或者基尼值小于阈值。
    - 由上述可知，熵和基尼值的大小表示数据的复杂程度，当熵或者基尼值过小时，表示数据的纯度比较大，如果熵或者基尼值小于一定程度数，节点停止分裂。

什么时候停止分裂

- 一般情况下为了降低决策树复杂度和提高预测的经度，会适当提前终止节点的分裂。
  - （3）决策树的深度达到指定的条件
    - 节点的深度可以理解为节点与决策树跟节点的距离，如根节点的子节点的深度为1，因为这些节点与跟节点的距离为1，子节点的深度要比父节点的深度大1。决策树的深度是所有叶子节点的最大深度，当深度到达指定的上限大小时，停止分裂。
  - （4）最小节点数
    - 当节点的数据量小于一个指定的数量时，不继续分裂。两个原因：一是数据量较少时，再做分裂容易强化噪声数据的作用；二是降低树生长的复杂性。提前结束分裂一定程度上有利于降低过拟合的影响。

决策树三大算法

- 1、**ID3算法**。ID3算法的核心是在决策树各个节点上应用 信息增益 准则选择特征，递归的构建决策树。
- 2、**C4.5算法**。C4.5算法用信息增益率选择特征，在树的构造过程中会进行剪枝操作优化，能够自动完成对连续属性的离散化处理。
- 3、**CART算法**。分类与回归树(classification and regression tree, CART)既可以用于回归也可以用于分类。使用二元切分法来处理连续型数值。CART算法使用Gini增长率作为分割属性选择的标准。

使用sklearn实现

使用sklearn实现：

- from sklearn.treeimport DecisionTreeClassifier
- DecisionTreeClassifier(criterion=’gini’, splitter=’best’, max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False)
- 重要参数：
  - criterion ：选择特征的方法。默认为基尼系数。也可以选择熵 ‘entropy’
  - splitter：切分规则。默认'best'：表示选择最优的切分。'random'：表示随机切分。
  - max_depth：可以为整数或者None，指定树的最大深度。
  - min_samples_split：默认2。指定每个内部结点包含的最少的样本数。
  - min_samples_leaf：默认1。指定每个叶结点包含的最少的样本数。

使用sklearn实现

- 使用sklearn实现：

In [278]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import KFold
from sklearn import metrics
import numpy as np

dataset = load_breast_cancer()
X = dataset['data']
y = dataset['target']

avg_scores = []
for i in range(5):

    kf = KFold(n_splits=10,shuffle=True)
    score = []
    for train_inx, test_inx in kf.split(X):
        clf = DecisionTreeClassifier(criterion='entropy', splitter='best',
    max_depth=None, min_samples_split=2, min_samples_leaf=1,
    min_weight_fraction_leaf=0.0, max_features=None,random_state=None,
    max_leaf_nodes=None, presort=False).fit(X[train_inx],y[train_inx])
        y_pre = clf.predict(X[test_inx])
        y_test = y[test_inx]
        score.append(metrics.accuracy_score(y_test,y_pre))
    avg_scores.append(np.mean(score))

print(np.mean(avg_scores))
```

```
0.9363721804511277
```

决策树的优点

- 决策树优点：
  - 不需要任何领域知识或参数假设。
  - 适合高维数据。
  - 简单易于理解。
  - 短时间内处理大量数据，得到可行且效果较好的结果。
  - 能够同时处理数据型和常规性属性。

决策树的缺点

- 决策树缺点：
  - 对于各类别样本数量不一致数据，信息增益偏向于那些具有更多数值的特征。
  - 易于过拟合。
  - 忽略属性之间的相关性。
  - 不支持在线学习。

问题？

暨南大學
JINAN UNIVERSITY