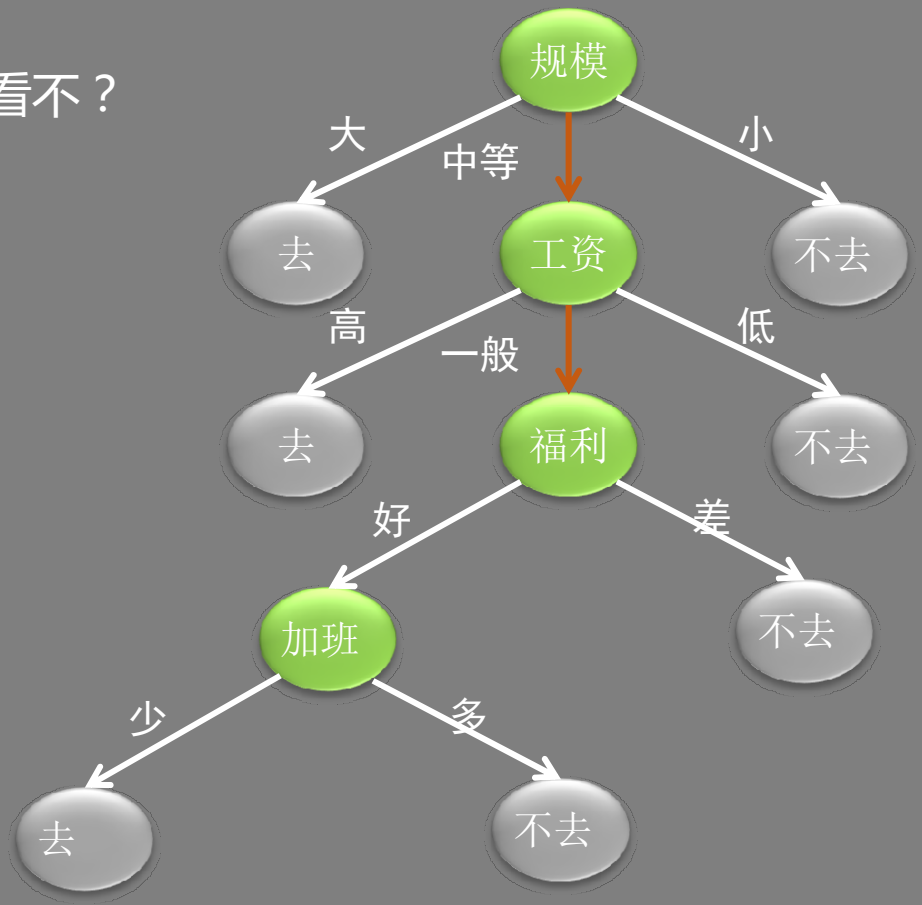
ZIXING AI
自兴人工智能

决策树学习
(Decision Tree Learning)

高琰

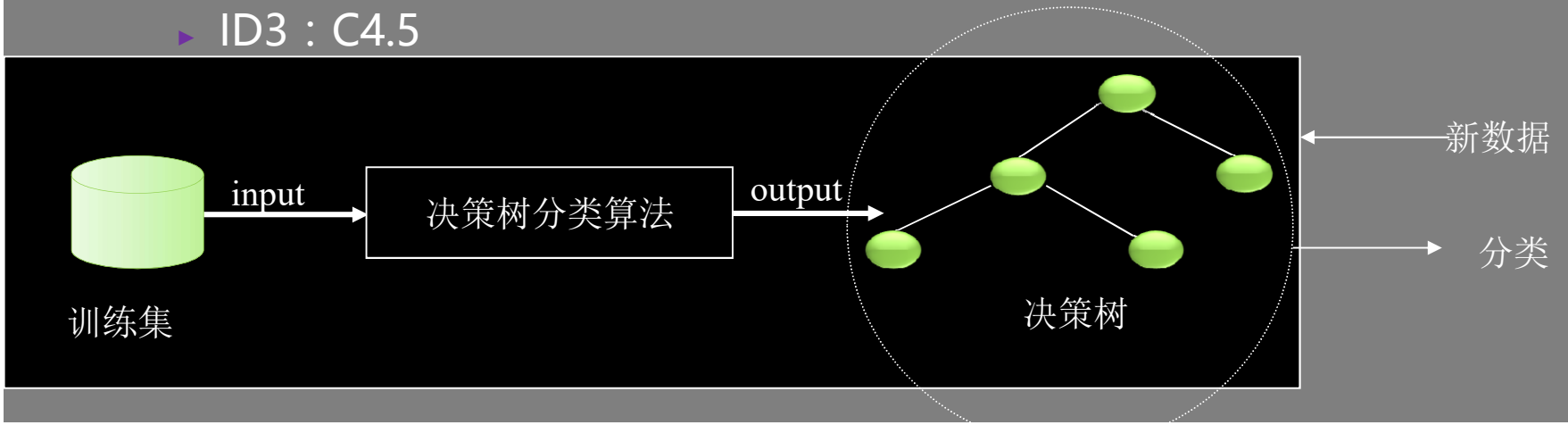
决策树学习

- 学生甲：XX公司来学校招聘，去看看不？
- 学生乙：是大公司吗？
- 甲：中等
- 乙：工资高吗？
- 甲：一般
- 乙：有五险一金吗？
- 甲：有
- 乙：活累不累，老要加班吗？
- 甲：据说还好
- 乙：那就去看看吧



决策树学习

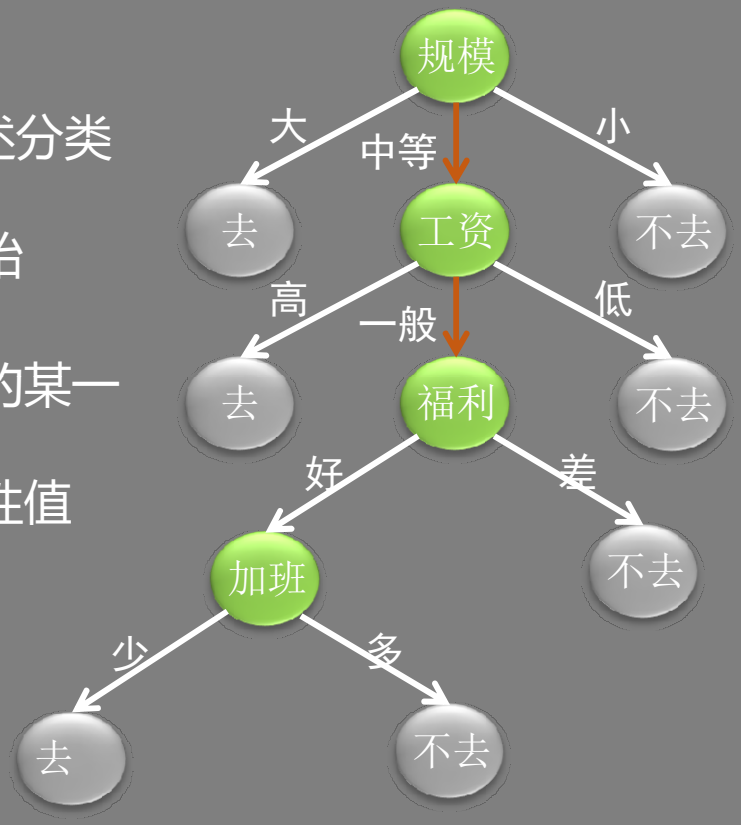
- 决策树是有监督学习
  - 给定训练样例，其中包含正例和反例
  - 从训练样例创建决策树
  - 流行算法
  - ID3：C4.5



决策树学习

ZIXING AI
自 兴 人 工 智 能

- 决策树结构
  - 由节点和边构成的用来描述分类过程的层次数据结构
  - 树的根接点——分类的开始
  - 叶节点——分类结果
  - 中间节点——相应实例中的某一属性
  - 边——某一属性可能的属性值



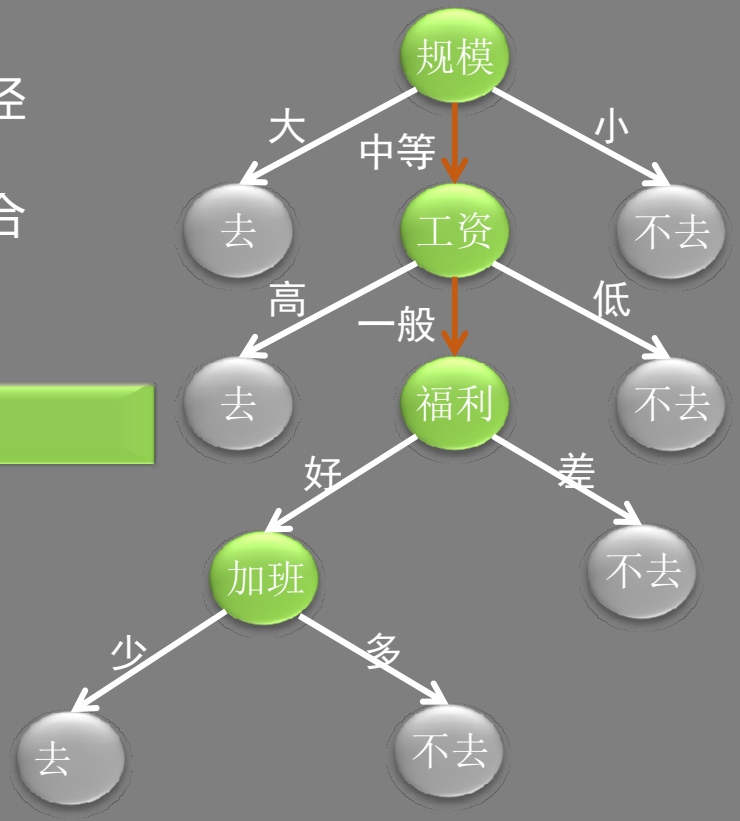
决策树学习

ZIXING AI
自兴人工智能

决策树

- 从根节点到叶节点的每一条路径都代表一个具体的实例
- 同一路径上的所有属性之间为合取关系
- 不同路径之间为析取关系

会去招聘会对应的表达式：
(规模 = 大)∨(规模 = 中等∧工资 = 高)∨(规模 = 中等∧工资 = 一般∧福利 = 好∧加班 = 少)

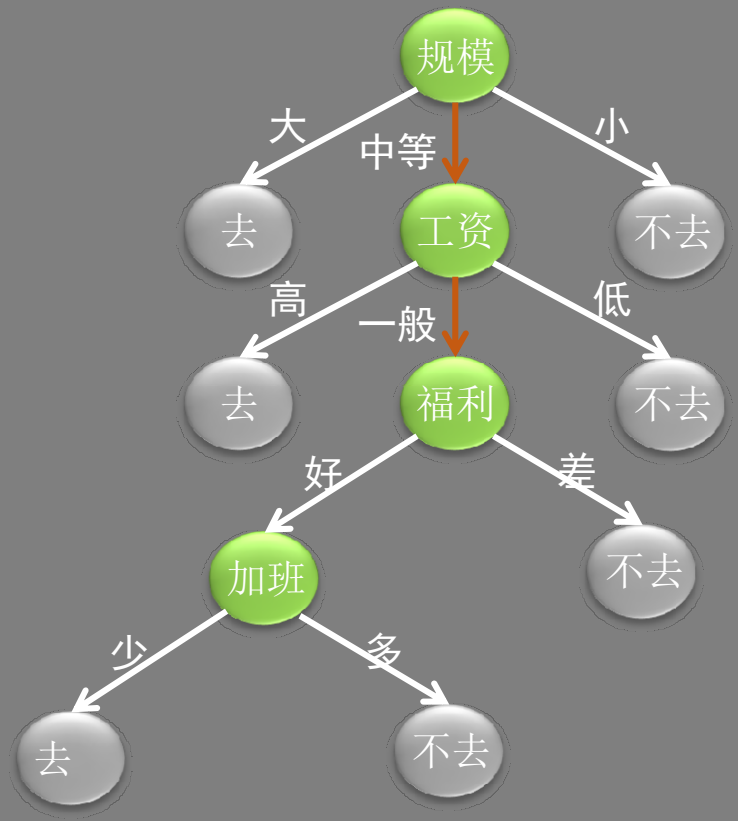


决策树还可以表示成规则的形式：
IF 规模=中等 AND 工资=高 THEN 去招聘会
…

ZIXING AI
自兴人工智能

决策树的生成

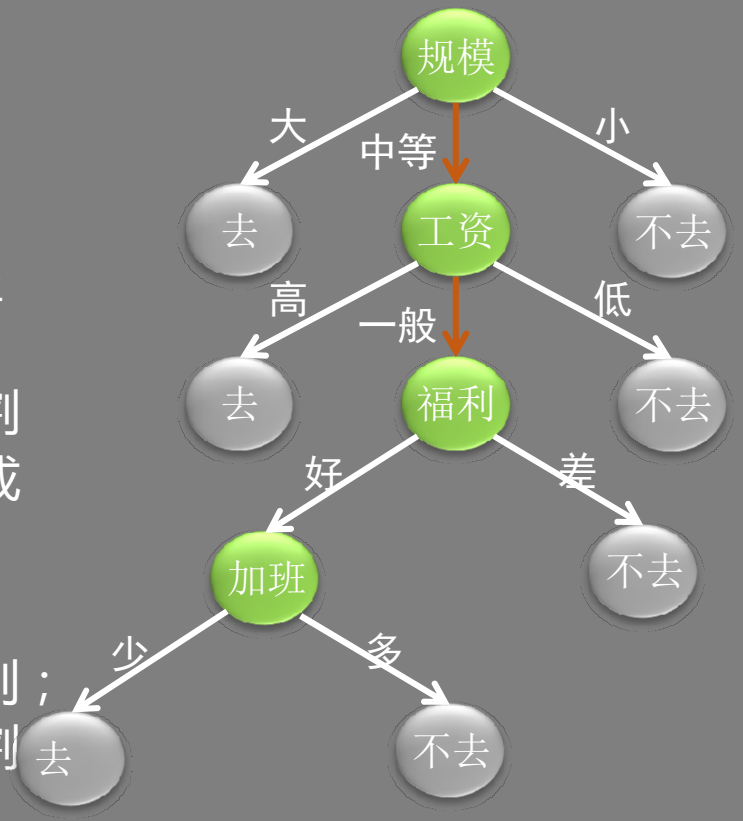
- 基本思想：分两个步骤
  - 树的生成
  - 树的修剪：去掉一些可能是噪音或者异常的数据
- 决策树使用: 对未知数据进行分割
  - 按照决策树上采用的分割属性逐层往下，直到叶子节点



决策树的生成

ZIXING AI
自 兴 人 工 智 能

- 基本算法（贪心算法）
  - 自上而下分而治之的方法
  - 开始时所有的实例都在根节点
  - 属性都是分类型 (如果是连续的，将其离散化)
  - 所有记录用所选属性递归的进行分割
  - 属性的选择是基于一个启发式规则或者一个统计的度量 (如信息增益)
- 停止分割的条件
  - 一个节点上的实例都属于同一个类别；
  - 没有属性可以再用于对数据进行分割

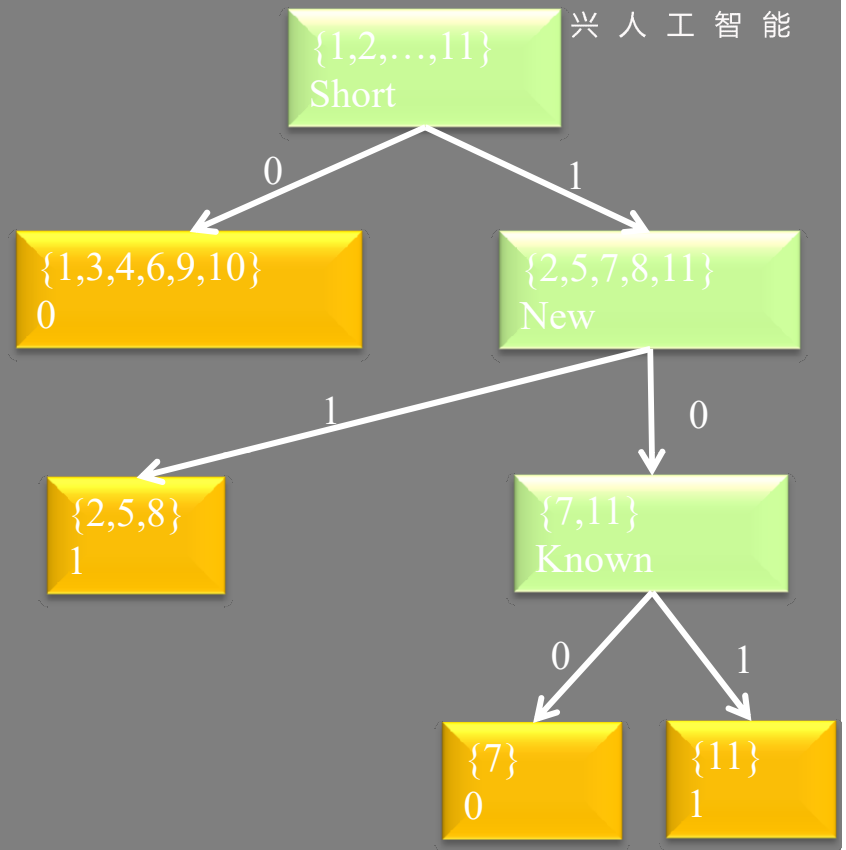


决策树的生成

例：邮件阅读

训练样例

序号	Known	New	Short	Home	Reads
1	1	1	0	1	0
2	0	1	1	0	1
3	0	0	0	0	0
4	1	0	0	1	0
5	1	1	1	1	1
6	1	0	0	0	0
7	0	0	1	0	0
8	0	1	1	0	1
9	1	0	0	1	0
10	1	1	0	0	0
11	1	0	1	1	1



ID3算法

ZIXING AI
自 兴 人 工 智 能

// 昆兰（J.R.Quinlan）于1979年提出的一种以信息熵（Entropy）的下降速度作为属性选择标准的一种学习算法

创建树的Root结点
如果Examples都为正，那么返回label=+中的单结点Root
如果Examples都为反，那么返回lable=-单结点树Root
如果Attributes为空，那么返回单节点树Root，lable=Examples中最普遍的类别
否则开始
根据什么确定？
A←Attributes中分类能力最好的属性
Root的决策属性←A
对于每个可能值
在Root下加一个新的分支对应测试A=v_i
令Example$_{vi}$为Examples中满足A属性值为v_i的子集
如果Examples$_{vi}$为空
在这个新分支下加一个叶结点,节点的lable=Examples中最普遍的目标属性值
否则在这个新分支下加一个子树ID3(example$_{vi}$,target-attribute,attributes-|A|)
结束

属性选择的度量

ZIXING AI
自 兴 人 工 智 能

- ID3以信息熵(**Entropy**)为基础计算信息增益(**Information gain**)
  - ID3, C4.5
- 熵
  - 刻画系统的混乱程度
  - 熵越大，系统越混乱
  - 信息熵是对信息源整体不确定性的度量，反映的是信源每发出一个信息所提供的平均信息量
  - 若在系统S中，类别属性具有n个不同信息，则S相对于n种信息的分类信息熵为

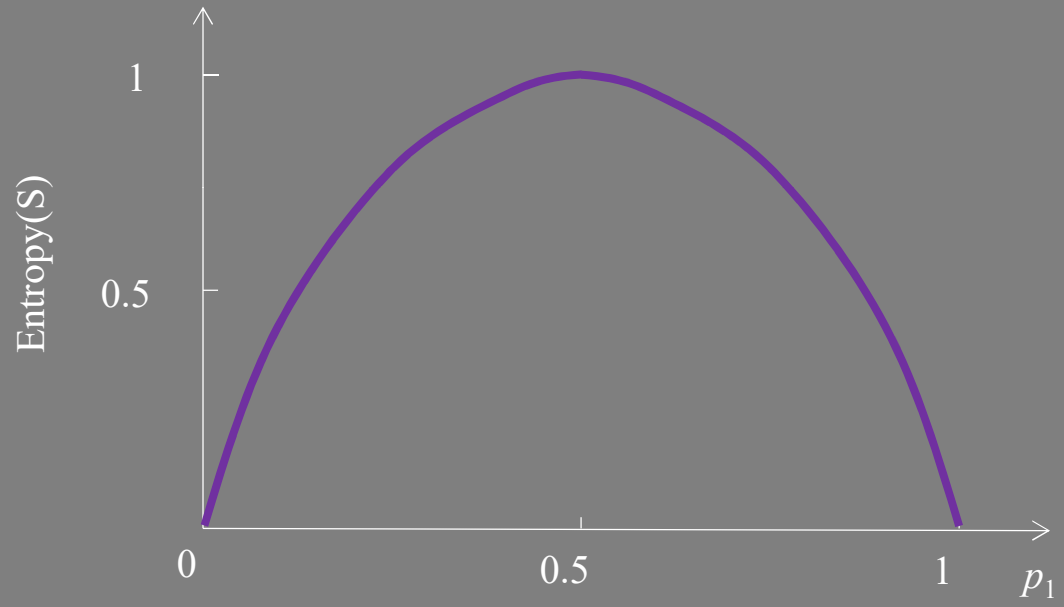
$$Entropy(S) = \sum_{i=1}^{n}(-p_i log_2 p_i)$$

其中p_i为S中属于类别i的比例

ZIXING AI
自兴人工智能

属性选择的度量

- 包含两类信息的系统的熵



计算熵的python程序

ZIXING AI
自 兴 人 工 智 能

```
from math import log
def calcShannonEnt(dataSet):
    numEntries = len(dataSet)
    labelCounts = {}
    for featVec in dataSet: # the number of unique elements and their occurance
        currentLabel = featVec[-1]
        if currentLabel not in labelCounts.keys(): labelCounts[currentLabel] = 0
        labelCounts[currentLabel] += 1
    shannonEnt = 0.0
    for key in labelCounts:
        prob = float(labelCounts[key])/numEntries
        shannonEnt -= prob * log(prob,2) #log base 2
    return shannonEnt
```

序号	Known	New	Short	Home	Reads
1	1	1	0	1	0
2	0	1	1	0	1
3	0	0	0	0	0
4	1	0	0	1	0
5	1	1	1	1	1
6	1	0	0	0	0
7	0	0	1	0	0
8	0	1	1	0	1
9	1	0	0	1	0
10	1	1	0	0	0
11	1	0	1	1	1

属性选择的度量

ZIXING AI
自 兴 人 工 智 能

- 信息熵的计算例
  - 设前述的邮件阅读问题为系统S，共11个样例，其中正例4个，反例7个，则

    Entropy(S)=-4/11 × $\log_2$(4/11)-7/11 × $\log_2$(7/11)=0.9457
- 分类的期望熵
  - 设系统S的熵为Entropy(S)，S具有属性A，*Values*(A)表示属性A所以可能的取值，S_v是S中属性A的值为v的集合，则

    用A分类S后熵的期望值=$\sum_{v\in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$

期望熵是每个子集的熵的加权和，权值为属于S_v与S中的样例的比例

- 例：邮件阅读问题中属性*short*有两种取值：0和1，其中取0的6个，全部为反例；取1的5个，其中4个正例，1个反例

$$\sum_{v\in\{0,1\}} \frac{|S_v|}{|S|} Entropy(S_v) = 6/11 Entropy(S_0) + 5/11 Entropy(S_1)$$

$$= 6/11(-6/6 log_2(6/6)) + 5/11(-4/5 log_2(4/5) - 1/5 log_2(1/5))$$

=0.32815

降低了

ZIXING AI
自兴人工智能

属性选择的度量

- 信息增益(Information Gain)
  - 使用S的属性对S分类后，S的熵降低了
  - 最佳的分类属性是使用该属性对训练样例进行分割后能导致期望熵降低得最多
  - 信息增益描述熵的减少量

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

  - 最佳分类属性——信息增益最大的属性

ZIXING AI
自兴人工智能

属性选择的度量

- **信息增益**(Information Gain)
  缺点：偏向于选择取值较多的特征
- **信息增益比**(Information Gain Rate)

信息增益：$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$

信息增益比：$Gr(S,A) = \frac{Gain(S,A)}{H_A(S)}$

$H_A(S) = -\sum_{i=1}^{n} \frac{|S_i|}{|S|} log_2 \frac{|S_i|}{|S|}$，$n$是特征A取值的个数

ZIXING AI
自兴人工智能

例

序号	属性值			决策方案
	x_1学历层次	x_2专业类别	x_3学习基础	y_i
1	1研究生	1电信类	1修过AI	y_3不修AI
2	1研究生	1电信类	2未修AI	y_1必修AI
3	1研究生	2机电类	1修过AI	y_3不修AI
4	1研究生	2机电类	2未修AI	y_2选修AI
5	2本科	1电信类	1修过AI	y_3不修AI
6	2本科	1电信类	2未修AI	y_2选修AI
7	2本科	2机电类	1修过AI	y_3不修AI
8	2本科	2机电类	2未修AI	y_3不修AI

例

ZIXING AI
自 兴 人 工 智 能

- 首先建立包含所有训练样例的根节点，计算其信息熵

$$Entropy(S) = \sum_{i=1}^{3}(-p_i log_2 p_i)$$

$$= -(1/8)\log_2(1/8) - (2/8)\log2(2/8) - (5/8)\log2(5/8) = 1.2988$$

- 计算S关于每个属性的期望熵

$$Entropy(S/x_i) = \sum_t \frac{|S_t|}{|S|} Entropy(S_i)$$

其中，t为属性x_i的属性值，S_t为$x_i=t$时的例子集，$|S|$和$|S_i|$分别是例子集S和S_i的大小。

  - 计算S关于属性x_1的期望熵

$x_1=1$: $S_1=\{1, 2, 3, 4\}$　　$x_1=2$: $S_2=\{5, 6, 7, 8\}$

$Entropy(S_1)$ = -(1/4)log2(1/4)- (1/4)log2(1/4)- (2/4)log2(2/4)=1.5

$Entropy(S_2)$ = -(1/4)log2(1/4)- (3/4)log2(3/4=0.8113

$Entropy(S/x_1)$ = (4/8) * 1.5+(4/8) * 0.8113=1.1557

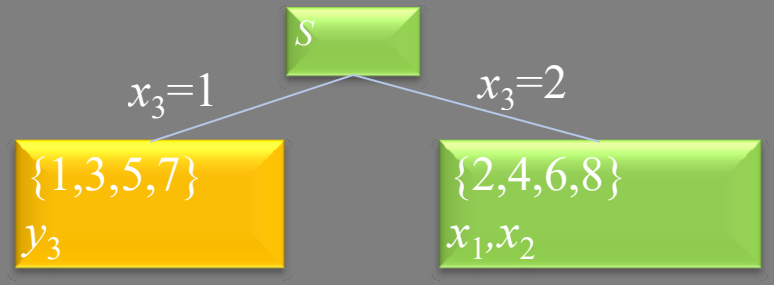
例

ZIXING AI
自 兴 人 工 智 能

- 类似计算S关于属性x_2 ，x_3的期望熵

$$Entropy(S/x_2) = 1.1557$$
$$Entropy(S/x_3) = 0.75$$

显然，期望熵越小，信息增益越大，应选择属性x_3对根节点进行扩展



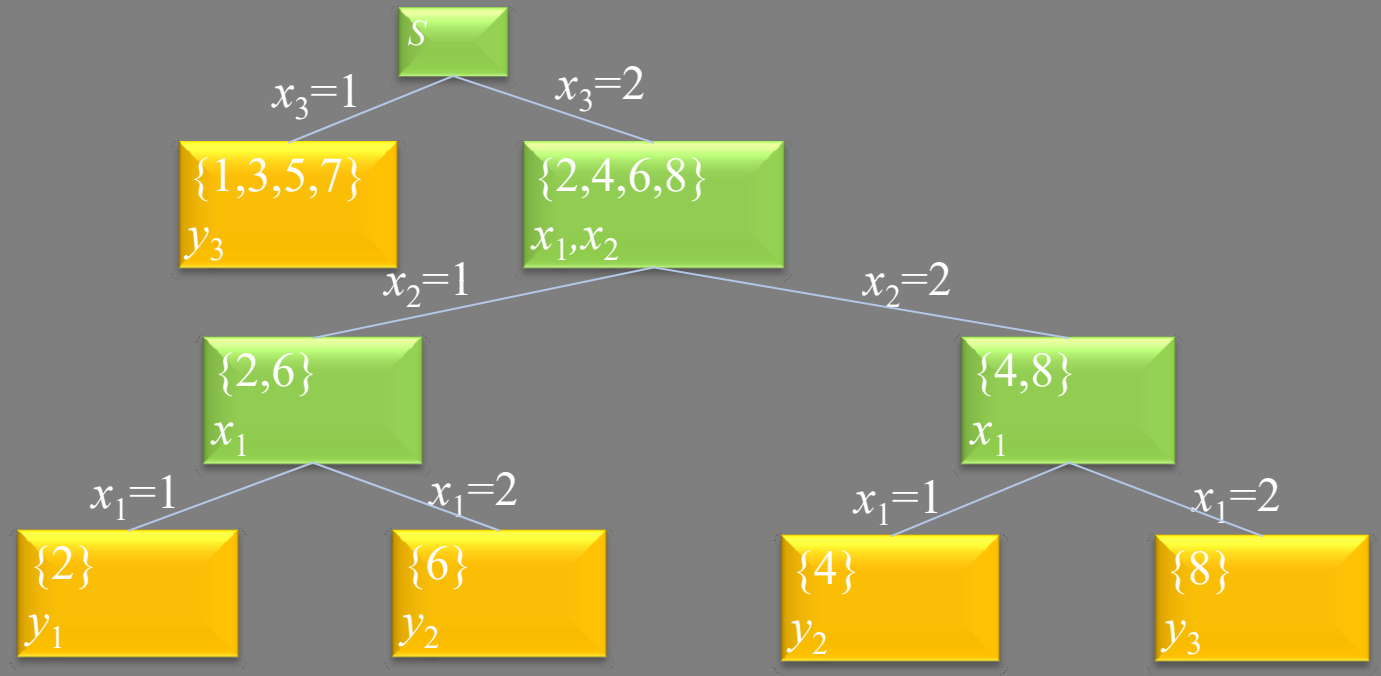
用属性x_3分类后的部分决策树

- 左边的节点只包含一种信息，无需再分
- 右边的节点在新的分类集中重复上述过程，计算属性x_1和x_2的期望熵均为1。任选一个，本例先选择x_2

例

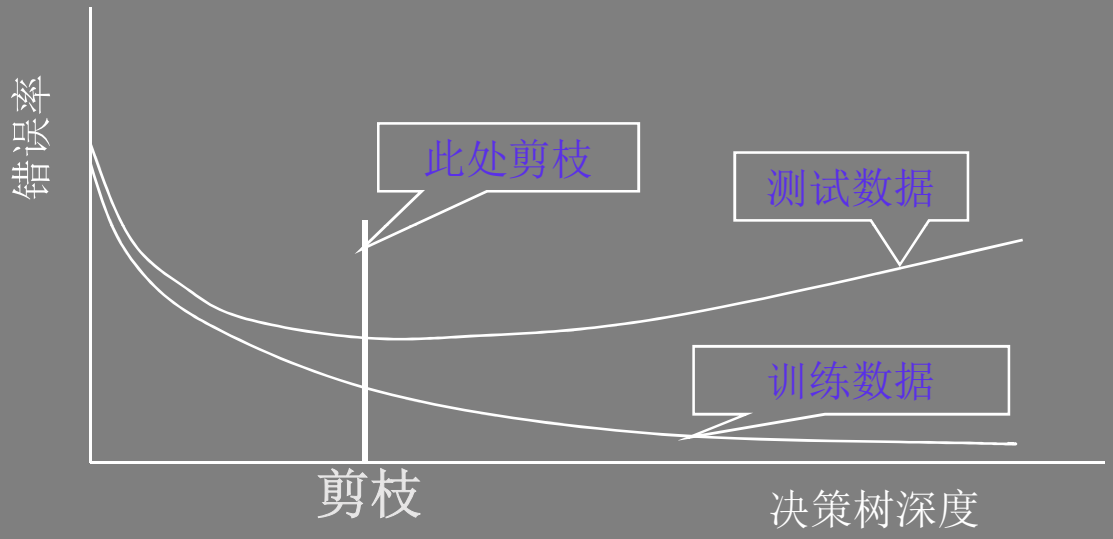
ZIXING AI
自 兴 人 工 智 能

- 最终决策树



决策树剪枝

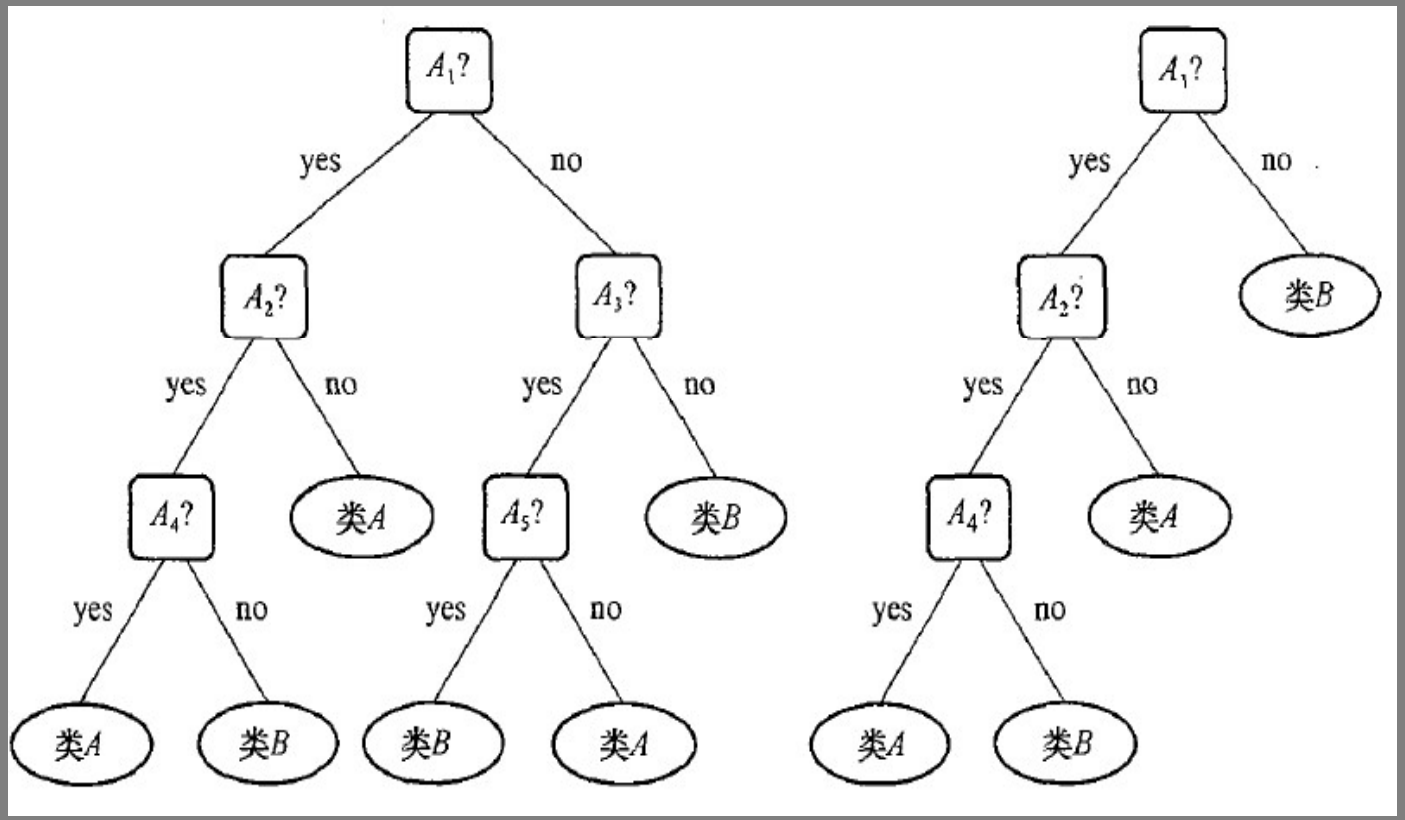
ZIXING AI
自 兴 人 工 智 能



- 避免过拟合
- 决策树泛化

决策树剪枝

ZIXING AI
自 兴 人 工 智 能

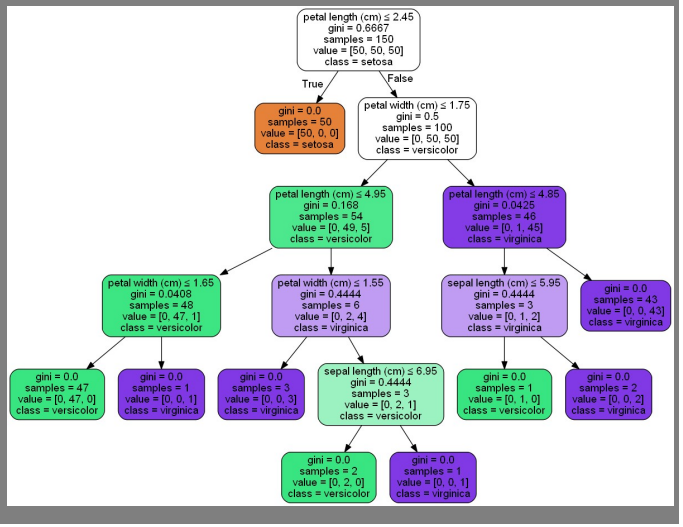


例 Scikit-learn中的决策树学习

```
#需要安装 pydotplus:    pip install pydotplus
#需要安装graphviz软件，且在环境变量path中添加graphviz的bin目录

from sklearn import tree
import pydotplus
from IPython.display import Image
from sklearn.datasets import load_iris

iris = load_iris()
#可以使用"gini"或者"entropy"，前者代表基尼系数，后者代表信息增益
clf = tree.DecisionTreeClassifier(criterion='entropy')
clf = clf.fit(iris.data, iris.target)
```

例 Scikit-learn中的决策树学习

```
dot_data = tree.export_graphviz(clf,
out_file=None,
                feature_names=iris.feature_names,
                class_names=iris.target_names,
                filled=True, rounded=True,
                special_characters=True)

graph0 =
pydotplus.graph_from_dot_data(dot_data)
Image(graph0.create_png())
```



主要参数

- criterion ：规定了该决策树所采用的的最佳分割属性的判决方法，有两种：“gini”，“entropy”。
- max_depth ：限定了决策树的最大深度，对于防止过拟合非常有用。
- min_samples_leaf ：限定了叶子节点包含的最小样本数，这个属性对于防止上文讲到的数据碎片问题很有作用
- 类别权重class_weight: 定样本各类别的的权重，主要是为了防止训练集某些类别的样本过多，导致训练的决策树过于偏向这些类别。这里可以自己指定各个样本的权重，或者用“balanced”，如果使用“balanced”，则算法会自己计算权重，样本量少的类别所对应的样本权重会高。当然，如果你的样本类别分布没有明显的偏倚，则可以不管这个参数，选择默认的"None" .不适用于回归树。

使用建议

- 1）当样本少数量但是样本特征非常多的时候，决策树很容易过拟合，一般来说，样本数比特征数多一些会比较容易建立健壮的模型

  2）如果样本数量少但是样本特征非常多，在拟合决策树模型前，推荐先做维度规约，比如主成分分析（PCA），特征选择（Losso）或者独立成分分析（ICA）。这样特征的维度会大大减小。再来拟合决策树模型效果会好。

  3）推荐多用决策树的可视化，同时先限制决策树的深度（比如最多3层），这样可以先观察下生成的决策树里数据的初步拟合情况，然后再决定是否要增加深度。

  4）在训练模型先，注意观察样本的类别情况（主要指分类树），如果类别分布非常不均匀，就要考虑用class_weight来限制模型过于偏向样本多的类别。

ZIXING AI
自 兴 人 工 智 能

总结

- 决策树的生成
  - ID3（信息增益，熵）
  - C4.5（信息增益比）
- scikit-learn中决策树学习的使用

ZIXING AI
自兴人工智能

项目1

- 使用决策树预测隐形眼镜类型

1、数据集：
http://archive.ics.uci.edu/ml/machine-learning-databases/lenses/

2、利用scikit-learn中的DecisionTreeClassfier进行分类,输出准确率.