Course 4

分类与预测

Classification and Prediction

Data Mining

数据挖掘

大纲

- 分类与预测
- 用决策树归纳分类
- 其他分类方法
- 预测
- 评估分类器或预测器的准确率

分类 VS. 预测

- 分类与预测是两种数据分析的形式，用来提取模型以描述数据或预测未来数据趋势。
  - 分类 (Classification)：
    - 预测分类标号
    - 在训练样本集和值(类标号)的基础上分类数据(建立模型)，并使用它分类新数据
  - 预测 (Prediction)：
    - 处理连续函数值并建立模型。比如预测空缺值，或是预测客户消费在计算机设备上的金额。
- 典型应用
  - 信誉证实
  - 目标市场
  - 医疗诊断
  - 性能预测

分类与预测的范例

- 分类
  - 银行贷款员需要分析数据，来弄清哪些贷款申请者是安全的，哪些是有风险的。故将贷款申请者分为“安全”和“有风险”两类。
  - 因此，我们需要建构一个分类模型，来预测类别属性的编号，即预测顾客属性类别。
- 预测
  - 银行贷款员需要预测贷款给某个顾客多少钱是安全的。
  - 因此，需要建构一个预测模型，预测一个连续值函数或有序值，常用方法是回归分析。

分类(Classification)

分类的意义

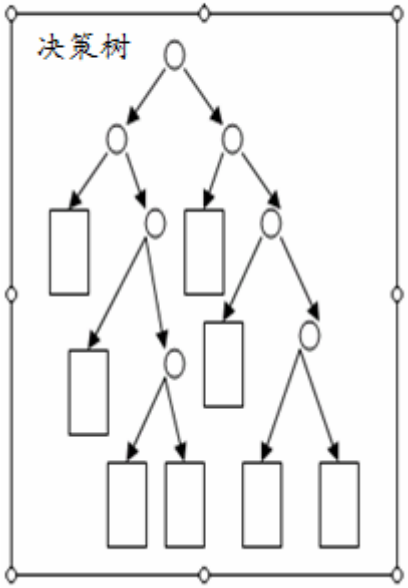
数据库

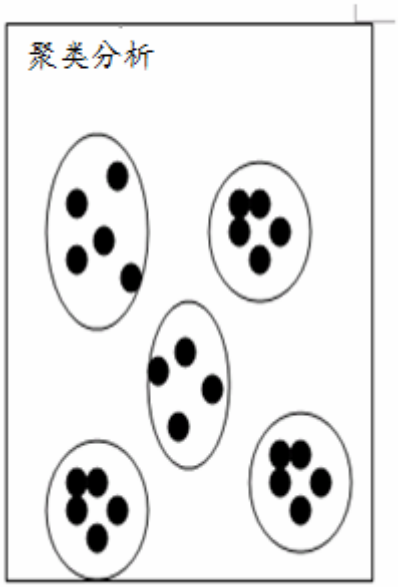
编號	性別	年齡	婚姻	家庭人數	購買RV房車
A0001	Male	45	未婚	1	是
A0002	Male	52	已婚	7	是
A0003	Female	38	已婚	5	是
A0004	Male	25	已婚	5	否
A0005	Female	48	已婚	4	是
A0006	Male	32	未婚	3	是
A0007	Female	65	已婚	4	否
A0008	Male	33	已婚	3	是
A0009	Male	45	已婚	4	是
A0010	Female	52	未婚	1	是
A0011	Male	38	未婚	1	否
…	…	…	…	…	…
Z0099	Male	22	未婚	4	是

预测

了解类别属性
与
特征

分类模型





数据分类：一个两步过程

- 第一步，建立描述预先定义的数据类或概念集的分类器
  - 又称学习或训练阶段
  - 假设每一元组/样本属于一个预先定义的类,由一个类标号属性的数据库属性确定
  - 基本概念
    - 训练集：用来建立模型的元组集
    - 训练样本：训练数据集中的单个样本（元组）
  - 学习模型可以用分类规则、决策树或数学公式的形式表示

数据分类：一个两步过程

- 第二步，使用模型，对将来的或未知的对象进行分类
  - 首先评估模型的分类准确率
    - 对每个测试样本进行处理，将此样本已知的类标号和该样本的模型分类结果比较
    - 模型在给定测试集上的准确率，是正确被模型分类的测试样本的百分比
    - 测试集要独立于训练样本集，否则会出现“过分拟合”的情况
  - 如果测试准确率是可接受的，模型将用来区别未知的数据

Example

2. 模型评估

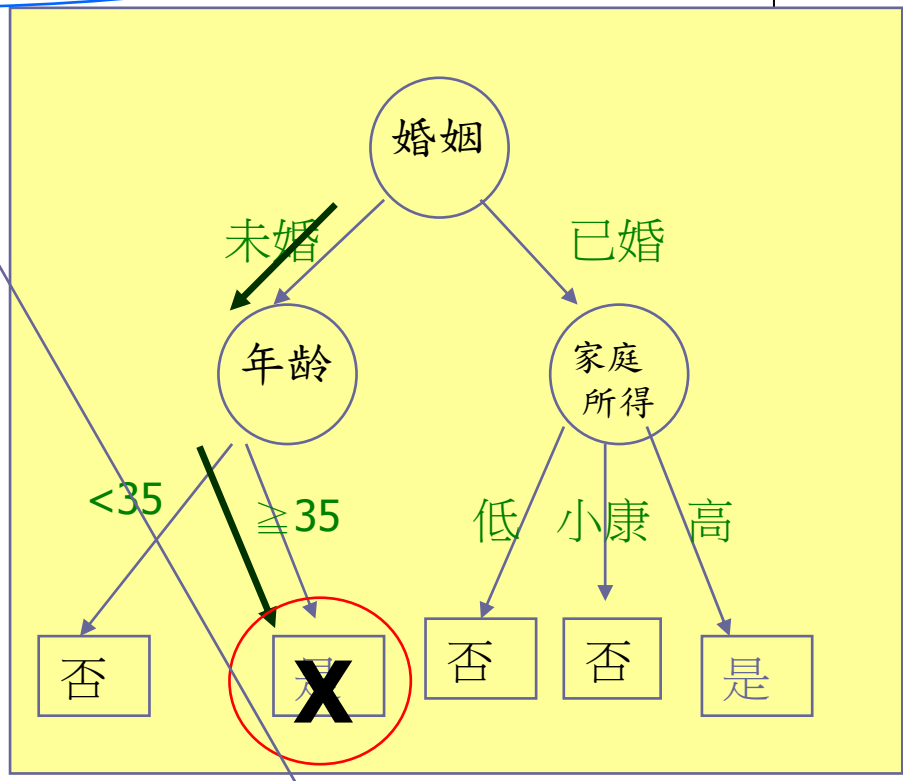
1.建立模型

数据

編號	性別	年齡	婚姻	家庭所得	購買RV房車
A0001	Male	<35	未婚	高所得	否
A0002	Male	<35	未婚	小康	否
A0003	Female	≧35	已婚	高所得	是
A0004	Male	≧35	未婚	低所得	是
A0005	Female	≧35	已婚	高所得	否
A0006	Male	≧35	已婚	低所得	否
A0007	Female	≧35	未婚	小康	否
A0008	Male	≧35	已婚	高所得	是
A0009	Male	<35	已婚	低所得	是

训练样本

测试样本



实际结果	預測結果
是	錯誤
是	正確
否	錯誤

错误率为 66.67%

修改模型

3.使用模型

編號	性別	年齡	婚姻	家庭所得	購買RV房車
W0144	Male	55	已婚	高所得	?

分类的目的

- 寻找影响某一重要变化的因素。
- 了解某一类别的特征。
- 建立分类规则。
  - 例如: 营销策略(市场区隔)
    银行(审核信用卡的额度)
    医疗诊断(SARS)

有监督的学习 VS. 无监督的学习

- 有监督的学习（用于分类）
  - 分类器的学习在被告知每个训练样本属于哪个类的“监督”下进行。
  - 新数据使用训练数据集中所得到的规则进行分类
- 无监督的学习（用于聚类）
  - 每个训练样本的类标号是未知的，要学习的类的个数或集合也可能事先不知道。
  - 透过一系列的度量、观察，将数据依照相似的元组进行聚集动作，来建立数据中的类标号或进行聚类

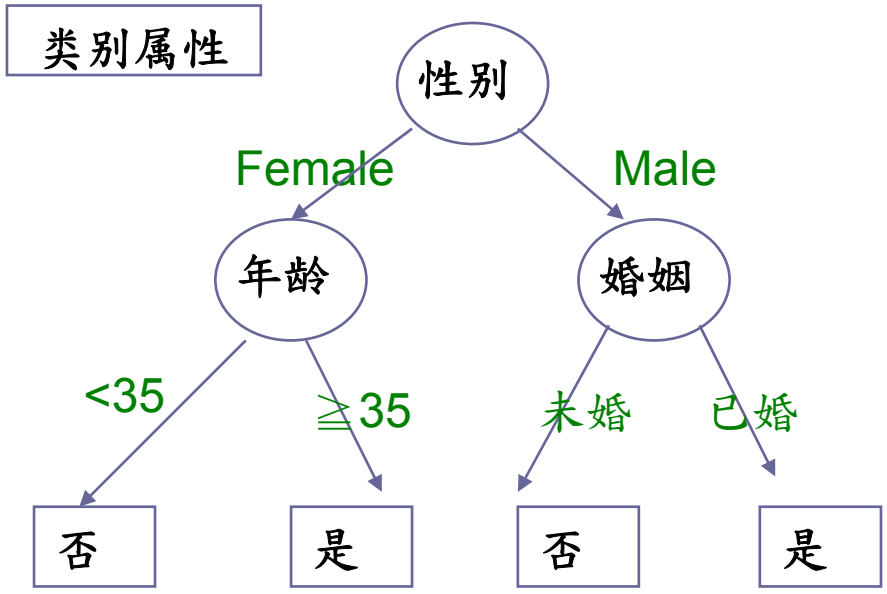
有监督的学习 VS. 无监督的学习

1.有监督(supervised learning)的机器学习法------

决策树(Decision Tree)

数据库

編號	性別	年齡	婚姻	家庭人數	購買RV房車
A0001	Male	45	未婚	1	是
A0002	Male	52	已婚	7	是
A0003	Female	38	已婚	5	是
A0004	Male	25	已婚	5	否
A0005	Female	48	已婚	4	是
A0006	Male	32	未婚	3	是
A0007	Female	65	已婚	4	否
A0008	Male	33	已婚	3	是
A0009	Male	45	已婚	4	是
A0010	Female	52	未婚	1	是
A0011	Male	38	未婚	1	否
…	…	…	…	…	…
Z0099	Male	22	未婚	4	是

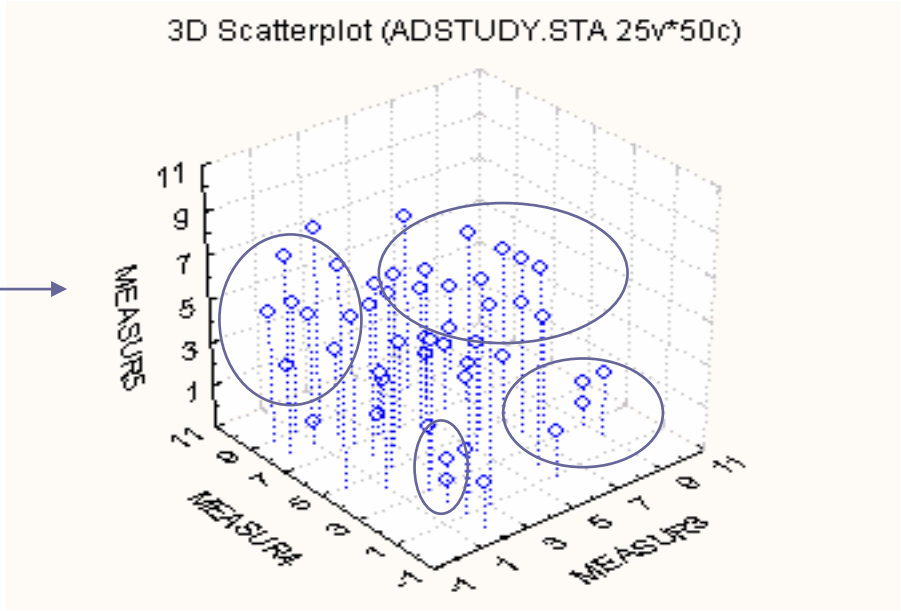


有监督的学习 VS. 无监督的学习

2.无监督(unsupervised learning)的机器学习法-----

聚类分析法(Cluster Analysis)

編號	性別	年齡	婚姻	家庭人數
A0001	Male	45	未婚	1
A0002	Male	52	已婚	7
A0003	Female	38	已婚	5
A0004	Male	25	已婚	5
A0005	Female	48	已婚	4
A0006	Male	32	未婚	3
A0007	Female	65	已婚	4
A0008	Male	33	已婚	3
A0009	Male	45	已婚	4
A0010	Female	52	未婚	1
A0011	Male	38	未婚	1
…	…	…	…	…
Z0099	Male	22	未婚	4



数值预测的两步过程

- 数值预测也是一个两步的过程，类似于前面描述的数据分类
  - 对于预测，没有“类标号属性”
  - 要预测的属性是连续值，而不是离散值，该属性可简称“预测属性”
    - 例：银行贷款员需要预测贷给某个顾客多少钱是安全的
- 预测模型可以看作一个映射函数y=f(x)
  - 其中x是输入；y是连续或有序的输出值
  - 与分类模型类似，预测模型的准确率计算，也要使用单独的测试集来进行

准备分类和预测的数据

- 数据进行预处理，可以提高分类或预测过程的准确性、有效性和可伸缩性
  - 数据清理
    - 消除或减少噪音，处理缺失值，从而减少学习时的混乱
  - 相关性分析
    - 数据中的有些属性可能与当前任务不相关；也有些属性可能是冗余的；删除这些属性可以加快学习步骤，使学习结果更精确
  - 集选择
    - 找出属性的归约子集
  - 数据变换与归约
    - 可以将数据泛化到较高层概念；或将数据进行规范化，使其落入一个指定的区间。

比较与评估分类和预测方法

- 使用下列标准比较与评估分类和预测方法
  - **准确率**：正确的预测新的或先前未见过的数据的类标号的能力
    - 训练测试法(training-and-testing)
    - 交互验证法(cross-validation)
  - **速度**：产生和使用分类器或预测器的计算花费
  - **鲁棒性**：给定噪声数据或具有缺失值的数据，分类器或预测器正确预测的能力
  - **可伸缩性**：对大量数据，有效的构造分类器或预测器的能力
  - **可解释性**：分类器或预测器提供的理解和洞察的水平，是主观的，较难评估。

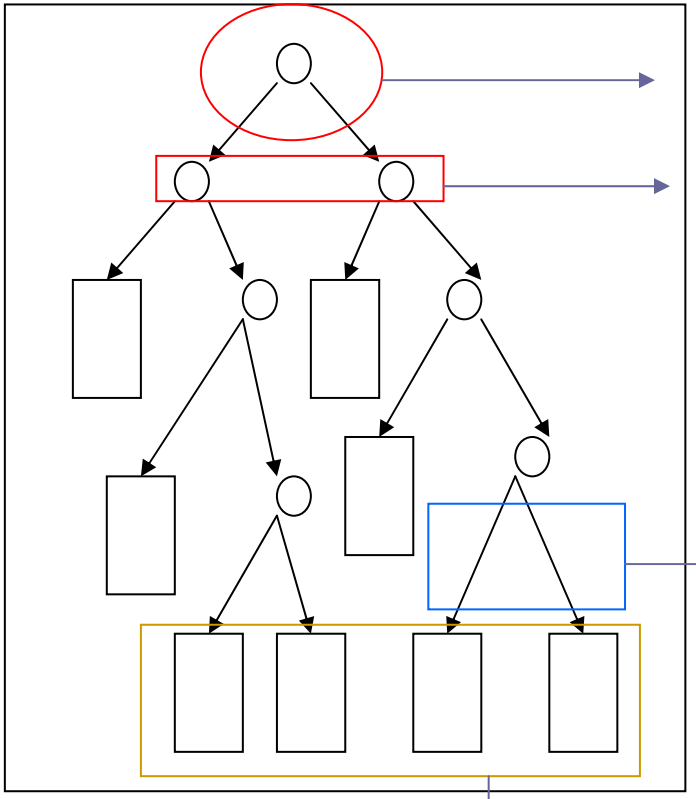
大纲

- 分类与预测
- 用决策树归纳分类
- 其他分类方法
- 预测
- 评估分类器或预测器的准确率

用决策树归纳分类

- 什么是决策树？
  - 类似于流程图的树结构
  - 每个**内部节点**表示在一个属性上的测试
  - 每个**分枝**代表一个测试输出
  - 每个**树叶节点**代表不同的类标号
- 判定树的生成包括两个过程
  - 树的建构
    - 首先所有的训练样本都在根结点
    - 基于所选的属性循环的划分样本
  - 树剪枝
    - 识别和删除哪些反应映噪声或孤立点的分支

决策树(Decision Tree)介绍



根节点(root node)

中间节点(non-leaf node)

（代表测试的属性）

⇒统称“内部节点”（Internal Node）

分支(branches)

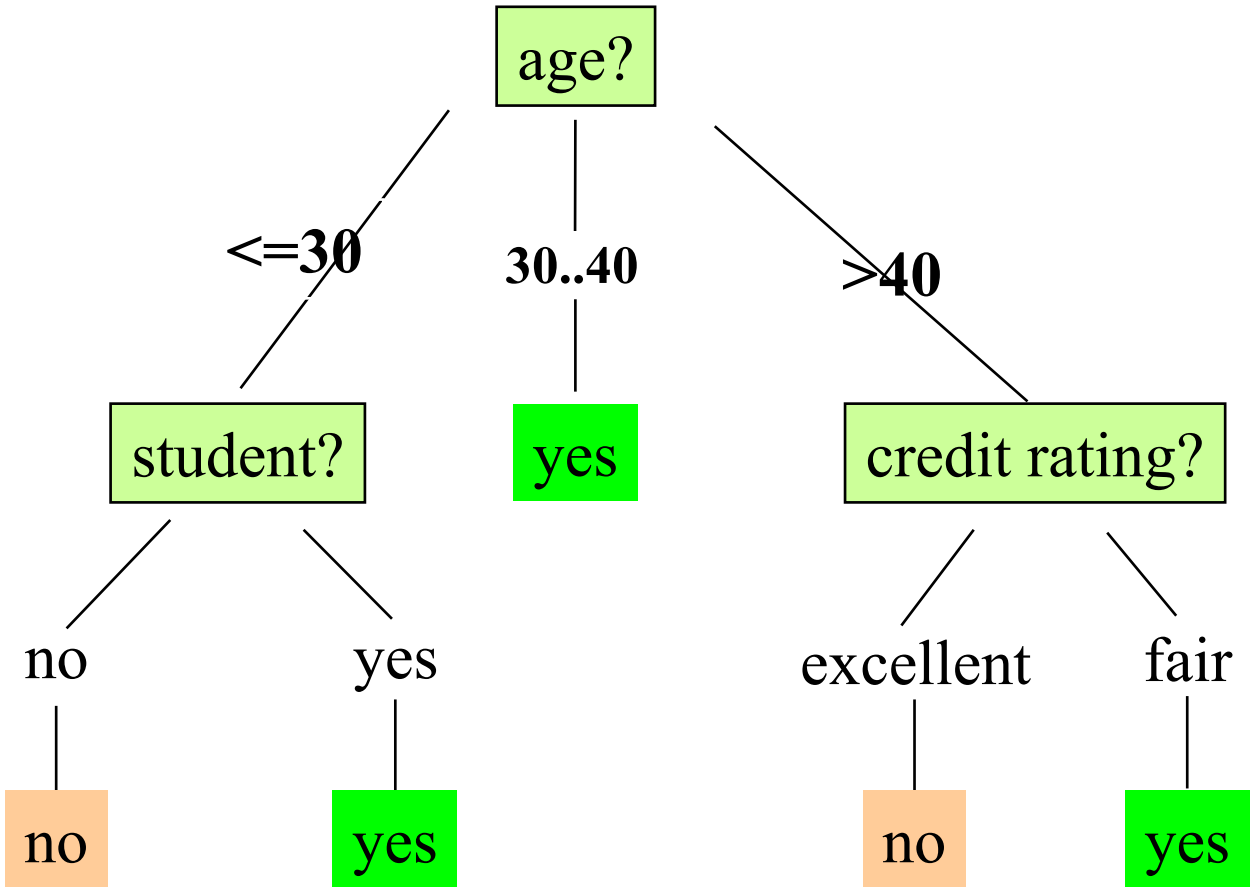
（代表测试输出）

树叶节点(leaf node)

（代表分类后所获得的类标号）

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

概念“buys_computer”的决策树



- 决策树的使用：对未知样本进行分类
  - 透过将样本的属性值与决策树相比较
  - 给定一个类标号未知的元组X，在决策树上测试元组的属性值，跟踪一条由根到叶节点的路径，该叶节点就存放着该元组的类预测。
- 决策树容易转换成为分类规则

决策树学习算法

- 算法：从分类数据S中产生决策树
- 输入数据：
  - 分类数据S：包含类标号的训练元组
  - *Attribute_list*：候选属性
  - *Attribute_selection_method*：指定选择属性的启发式过程，所选择的属性按类“最好”的区分元组
- 输出结果：决策树

数据库

編號	性別	年齡	婚姻	家庭人數	購買RV房車
A0001	Male	45	未婚	1	是
A0002	Male	52	已婚	7	是
A0003	Female	38	已婚	5	是
A0004	Male	25	已婚	5	否
A0005	Female	48	已婚	4	是
A0006	Male	32	未婚	3	是
A0007	Female	65	已婚	4	否
A0008	Male	33	已婚	3	是
A0009	Male	45	已婚	4	是
A0010	Female	52	未婚	1	是
A0011	Male	38	未婚	1	否
…	…	…	…	…	…
Z0099	Male	22	未婚	4	是

- 基本的算法概念:

1. **数据设定**：将原始数据分成两组，一部分为**训练数据**，一部分为**测试数据**
2. **决策树生成**：使用训练数据来**建立决策树**，而在每一个内部节点，则依据属性选择度量来评估选择哪个属性做分支，这又称节点分裂（Splitting Node）
3. **剪枝**：去掉一些可能是噪音或者异常的数据

将以上1~3步骤不断重复进行，直到所有的新产生节点都是树叶节点为止，且:

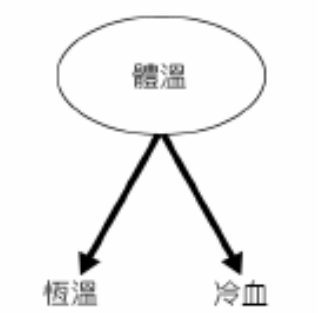
- 该数据集中，每一笔数据都已经被归类在相对应的类别中，没有任何尚未处理的数据
- 该数据集中，已经没有办法再找到新的属性来进行节点分裂

决策树生成

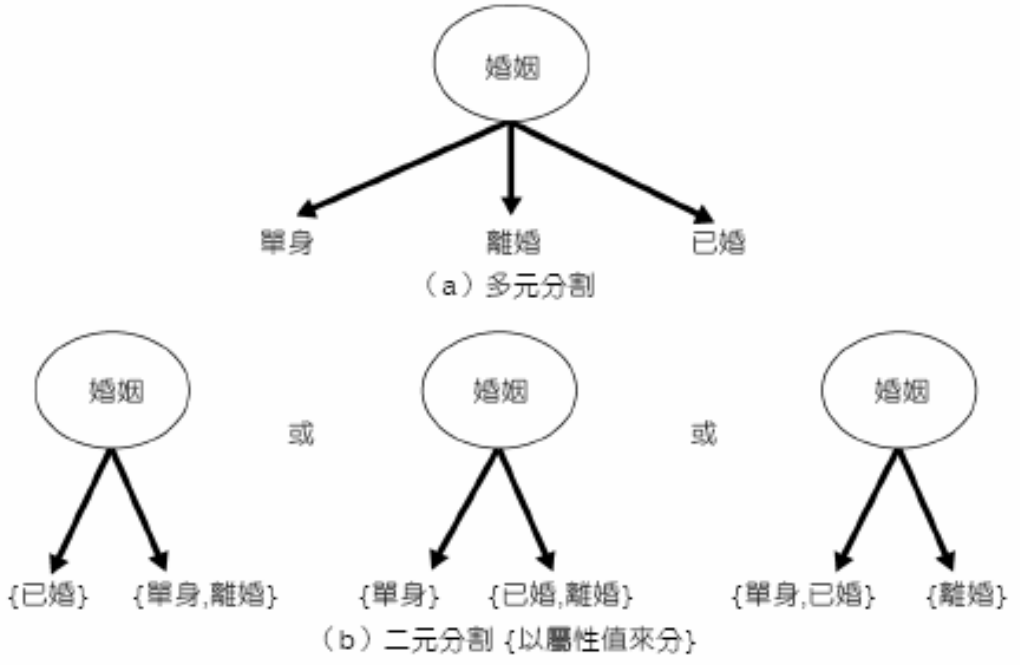
- 依据属性选择度量，从现有的属性中，挑出**分类能力最好**的属性做为树的**内部节点(根节点、中间节点)**
- 根据内部节点的所有值，产生出对应的**分支**
- 针对每一个新产生的分支，将训练数据**重新排列**，以进行下一个内部节点的产生
- 重复上面的过程 ，直到满足终止条件

不同类型的属性分裂

- 二元属性
  - 分裂后会产生两个不同方向的结果

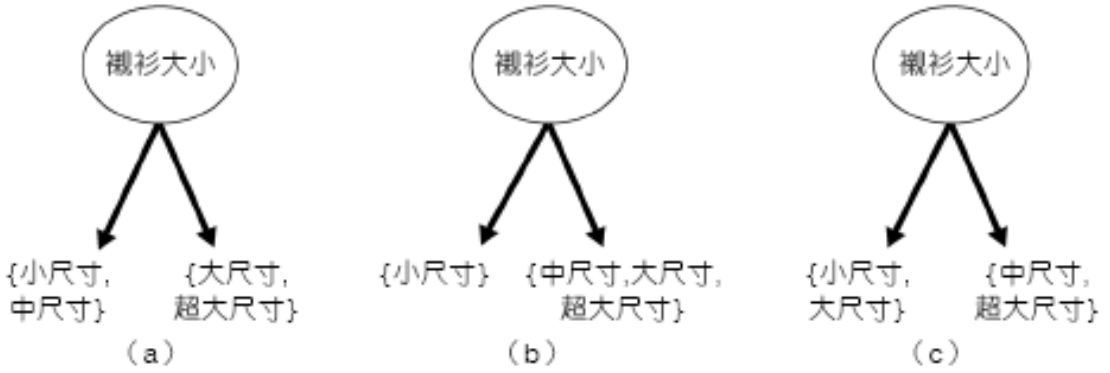


- 类别属性
  - 可分裂出不同值域的分支
  - 每个分支还可以以集合的型态表示

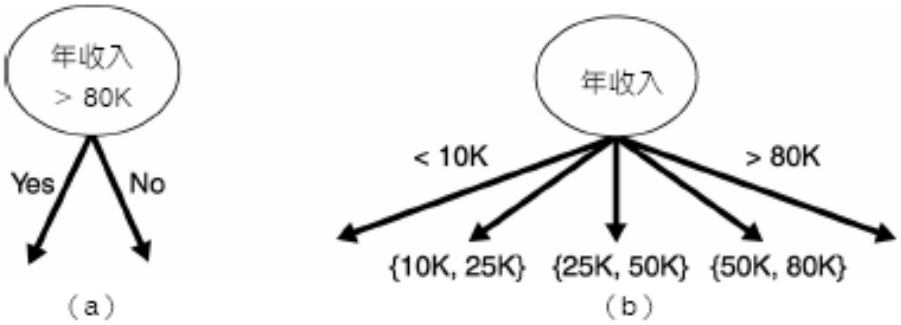


连续属性

- 可产生出二元或是多元分裂，连续属性内的值可以使用集合，但是在形成集合时必须没有违反属性值的顺序



- 可采用离散化的方式，将数据分裂成许多区间，但是仍需要保持数据的顺序性。



属性选择度量

- 属性选择度量是一种选择分裂条件的方法，将给定类标号的训练元组数据“最好”的分裂成各个类别。
  - 理想情况，每个划分是纯的，即落在给定划分的所有元组都属于相同的类。
  - 因此，属性选择度量也称为分裂准则
- 常用的属性选择度量有：
  - 信息增益(Information Gain) – ID3
  - 增益率 (Gain Ratio) – C4.5
  - Gini系数 (Gini Index) – CART
  - χ^2独立性测试 – CHAID
  - …

ID3（Iterative Dichotomiser）决策树

- ID3在建构决策树过程中，以信息增益 (Information Gain) 为准则，并选择具有**最高信息增益**的属性作为分类属性。
  - 以**熵 (Entropy)**为基础

用熵衡量数据的一致性

- 熵，可当作信息的度量(**不确定性**)指标，当熵值愈大，则代表蕴含的信息的程度愈高。
- 【说明范例】丢硬币
  - 若硬币是公平的，则丢出正面与反面的机率是一样的
  - 若硬币是动过手脚的，则丢出正面与反面的机率不会是一样的
  - 给定一组丢硬币后之数据集合S，该组数据的熵值计算公式为

$$Entropy(S)=-p_+\log_2 p_+ - p_-\log_2 p_-$$

  - 若丢了14次硬币，出现了9个正面与5个反面(记为$[9_+, 5_-]$)，则对于这个范例的熵为：

$$Entropy([9_+, 5_-])= -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.94$$

    - 若硬币丢出正面与反面的数量是一样，则熵为1
    - 若硬币是动过手脚的，不论怎么丢都只会出现正面（或反面），则熵为0

- 如果目标属性具有c个不同的值，那么S相对于c个状态的分类的熵定义为：

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

其中p_i为每个状态出现的机率

信息增益（Information Gain）

- 我们利用信息增益来衡量**某个属性用来分类训练数据的能力**。
- 一个评估属性A相对于元组集合D的信息增益Gain(D, A)被定义为：

$$Gain(D,A) = -\sum_{i=1}^{m} p_i \log_2(p_i) - \sum \frac{(D_j)}{|D|} \times Entropy(D_j) = Info(D) - Info_A(D)$$

  - $Info(D)$ ：原来的信息需求
  - $Info_A(D)$ ：根据属性A的v个值，对D进行划分所需要的期望信息，所需的信息越小，划分的纯度越高。
  - Gain(D, A) ：通过A的划分我们得到了多少
    - Gain值愈大，表示因为知道A的值，而使得完成元组分类还需要的信息愈小，因此使用A来分类数据会愈佳
    - Gain值愈小，表示就算知道A的值，完成元组分类还需要的非常多的信息，因此使用A来分类数据会愈差

- 【说明范例】天气评估
  - 假设有一套天气评估系统S，它有一些评估属性 (如: 风力、湿度、...)。
  - 以风力 (Wind)为例，它在所有的训练数据中所会出现的值为: weak, strong
  - 若目前有14个范例数据，其中有9个正例与5个反例(记为[9_+, 5_-])
  - 这14个范例数据中，关于风力的数据:
    - Wind = weak在所有范例中有6个正例与2个反例 [6_+, 2_-]
    - Wind = strong在所有范例中有3个正例与3个反例 [3_+, 3_-]
- 我们想要得知风力这个属性的信息增益为多少。

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

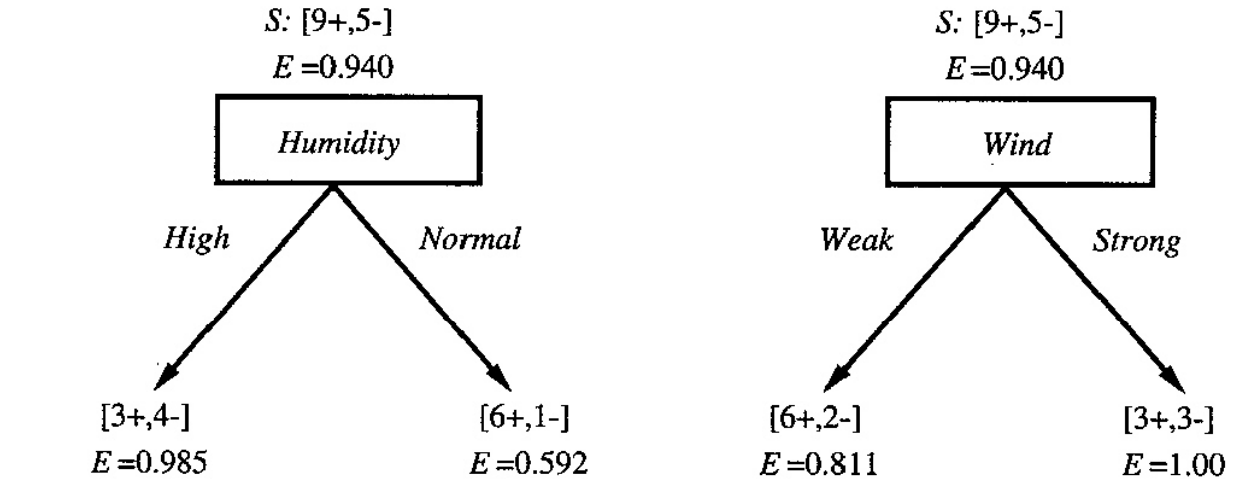
$$Values(Wind) = Weak, Strong$$

$$S = [9+, 5-]$$

$$S_{Weak} \leftarrow [6+, 2-]$$

$$S_{Strong} \leftarrow [3+, 3-]$$

Which attribute is the best classifier?



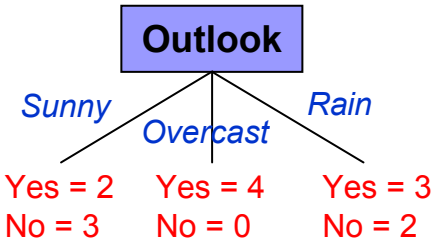
Gain (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151

Gain (S, Wind)
= .940 - (8/14).811 - (6/14)1.0
= .048

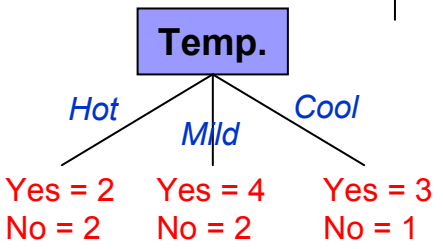
FIGURE 3.3
Humidity provides greater information gain than *Wind*, relative to the target classification.

ID3算法举例

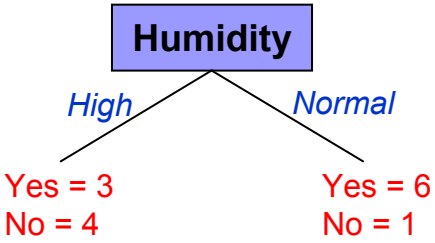
Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



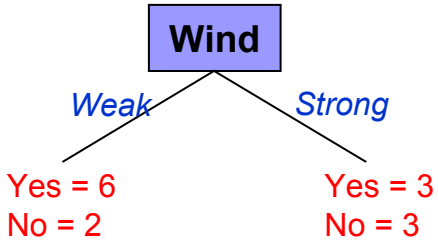
Gain(S, Outlook) = 0.246



Gain(S, Temperature) = 0.029



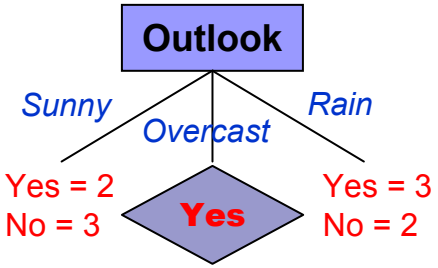
Gain(S, Humidity) = 0.151

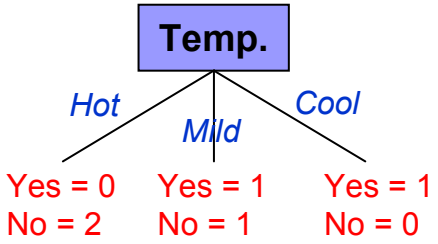


Gain(S, Wind) = 0.048

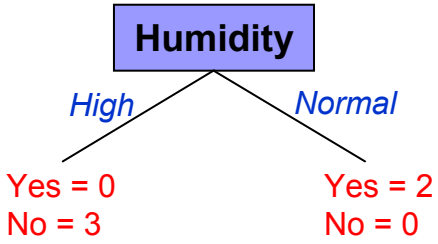
- 挑出具**最大信息增益**的属性，因此以Outlook为根节点(root)
- 由于Outlook的三个评估值中，Overcast(多云)的这个评估值得到4个正例 (Yes)，没有任何反例，因此Outlook = Overcast可得到一个**叶子节点 "Yes"**。

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

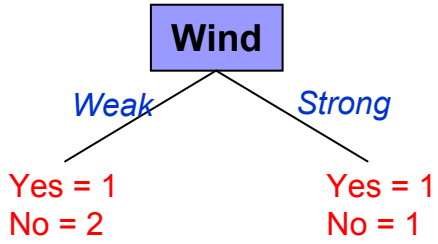




$Gain(S_{Sunny}, Temp.) = 0.57$

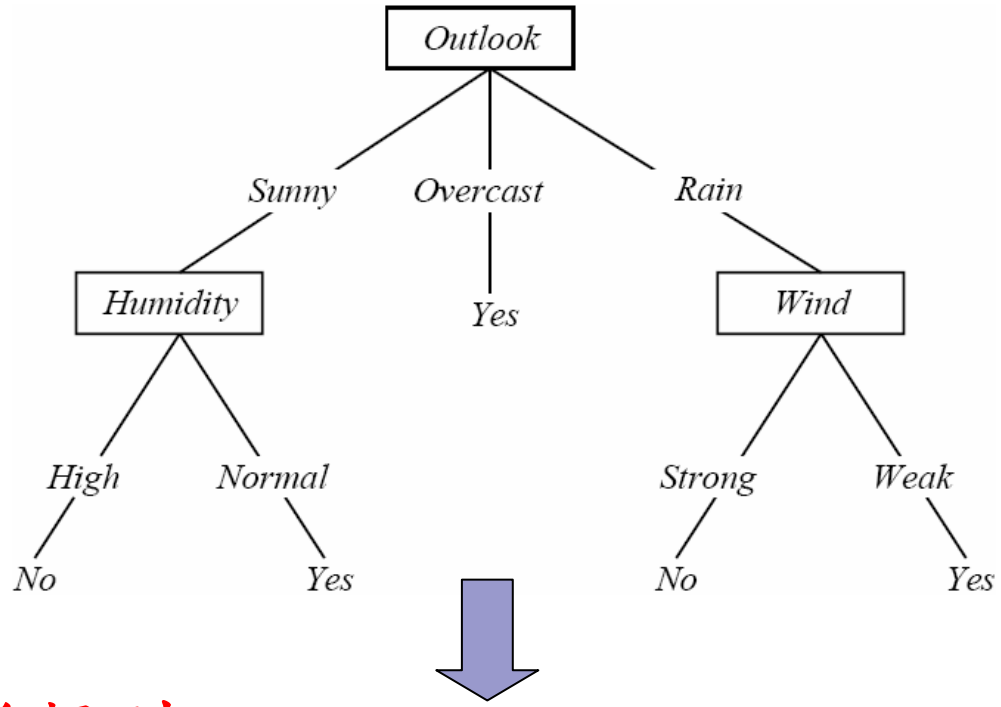


$Gain(S_{Sunny}, Humidity) = 0.97$



$Gain(S_{Sunny}, Wind) = 0.019$

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Strong	Yes
D14	Rain	Mild	High	Strong	No



分类规则:

If Outlook = Sunny and Humidity = High Then Play Tennis =No
If Outlook = Sunny and Humidity = Normal Then Play Tennis =Yes
If Outlook = Overcast Then Play Tennis =Yes
If Outlook = Rain and Wind = Strong Then Play Tennis =No
If Outlook = Rain and Wind = Weak Then Play Tennis =Yes

增益率(C4.5)

- 信息增益会倾向于选择拥有类别数较多的属性
  - 例如："产品编号"是一个仅包含唯一值的属性，每一个产品的产品编号皆不同。
  - 若依产品编号进行分裂，会产生出许多分支，且每一个分支都是很单一的结果，其信息增益会最大。但这个属性对于建立决策树是没有意义的。
- C4.5 (ID3后继)利用分裂信息来克服偏倚 (信息增益规范化)

$$SplitInfo_A(S) = -\sum_{j=1}^{v} \frac{|S_j|}{|S|} \times \log_2(\frac{|S_j|}{|S|})$$

  - 增益率GainRatio(A) = Gain(S, A)/SplitInfo$_A$(S)
  - 拥有最大增益率的属性被设为分裂属性
- 试用AllElectronics公司顾客数据库的训练数据计算 "income"的增益率。

- AllElectronics公司顾客数据库的训练数据

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- 试用AllElectronics公司顾客数据库的训练数据计算“income”的增益率。

$$SplitInfo_{income}(S) = -\frac{4}{14}\times\log_2(\frac{4}{14}) - \frac{6}{14}\times\log_2(\frac{6}{14}) - \frac{4}{14}\times\log_2(\frac{4}{14}) = 1.557$$

  - GainRatio(income) = 0.029/1.557 = 0.019

Gini指标(CART)

- 数据合集S包含n个类别，Gini指标 Gini(S) 定义为

$$Gini(S) = 1 - \sum_{i=1}^{m} p_i^2$$

p——被正确分类的概率

p_j为在S中的元组属于类别j的概率

- 利用属性A分裂S为S_1与S_2(二元属性)。则根据此分裂S的Gini指标为：

$$Gini_A(S) = \frac{|S_1|}{|S|} Gini(S_1) + \frac{|S_2|}{|S|} Gini(S_2)$$

- 若属性A为多元属性，则Gini指标为:

$$Gini_A(S) = \sum_{j=1}^{v} \frac{|S_j|}{|S|} Gini(S_j)$$

- 不纯度的降低值为:

$$\Delta Gini(A) = Gini(S) - Gini_A(S)$$

- 若属性A拥有最大不纯度的降低值(或：用属性A进行分裂，得到的子集合的Gini指标$Gini_A(S)$最小，不纯度小)，则该属性将被选为分裂属性

- 例. 数据集合S有9笔购买计算机(buy_computer)为yes，剩下5笔为no

$$Gini(S) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- 假设 “income”属性分裂 S 中 10 笔数据被分类到 S_1: {低, 中} 剩下 4 笔到 S_2: {高}

$$Gini_{income\in\{低,中\}}(S) = \left(\frac{10}{14}\right)Gini(S_1) + \left(\frac{4}{14}\right)Gini(S_2)$$

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) = 0.443$$

  - Income属性中的值尚有其它可能的划分：{低, 高}与{中}、{低}与{中, 高}。但是 $Gini_{\{中,低\}} = 0.443 = Gini_{\{高\}}$，比其它可能划分的Gini数值小。所以属性收入的{中, 低}与{高}被选为分裂属性的条件，因为它拥有最小的Gini指标值。
  - 此属性甚至比其它属性如 “age”与 “credit_rating”较佳。

属性选择指标比较

- 三种常用指标
  - 信息增益:
    - 趋向于包含多个值的属性
  - 增益率:
    - 会产生不平衡的分裂，其中一个划分比其他划分小得多
  - Gini指标:
    - 倾向于包含多个值的属性
    - 当类别个数很多时会有困难
    - 倾向那些会导致相等大小的划分和纯度

防止分类中的过分拟合

- 产生的决策树会出现过分适应数据的问题
  - 决策树学习可能遭遇模型**过分拟合**（overfitting）的问题
    - 过分拟合是指模型过度训练，导致模型记住的不是训练数据的一般特性，反而是训练数据的异常特性。
    - 由于数据中的噪声和离群点，许多分枝反映的是训练数据中的异常
    - 对新样本的分类将会变得很不精确
  - 决策树的剪枝来移除最不可靠的分支。
- 剪枝的两种方法
  - 先剪枝 (Prepruning)
  - 后剪枝 (Postpruning)

- 先剪枝 (Prepruning)
  - 通过提前停止树的构造达到剪枝的目的。
  - 当选择某一个属性做为决策树的一个内部节点，若会导致该属性的指标低于事先定义的阈值时，则给定属性的进一步划分将停止，该节点成为叶子节点。
  - 指标可使用信息增益、Gini指标...等
  - 选择一个合适的阈值往往很困难。较高的阈值会导致过分简化的决策树；较低的阈值会导致过分复杂的决策树
- 后剪枝 (Postpruning)
  - 由“完全生长”的树剪去分枝
  - CART使用代价复杂度剪枝(Cost Complexity Pruning)方法，以决策树的叶节点的个数与树的错误率构成代价复杂度函数。
  - C4.5使用悲观剪枝(Pessimistic Pruning)方法，也是使用错误率来进行剪枝。

决策树的可伸缩性

- 分类挖掘是一个在统计学和机器学习的领域也被广泛研究的问题，并提出了许多算法，但是这些算法都只能在内存中运行。
  - ID3, C4.5, CART在对少量数据建立决策树非常有效率。
  - 如果训练数据大到无法全部储存在计算机的主存储器中怎么办？
- 可伸缩性问题：要求以合理的速度对数以百万计的样本和数以百计的属性 (字段) 进行分类挖掘。
- 由大型数据库建构决策树：
  - 首先将样本划分成子集合，每个子集合可被入主存储器中
  - 由每个子集合建构一颗决策树
  - 将每个子集合的分类组合在一起

大纲

- 分类与预测
- 用决策树归纳分类
- 其他分类方法
- 预测
- 评估分类器或预测器的准确率

其它分类方法

- 贝叶斯分类
- 神经网络
- SVM
- k-最近邻分类
- 基于案例的推理
- 遗传算法
- 模糊集方法

贝叶斯理论：基础

- 假设 X 是数据元组: 类别未知
- 假设 H 为某种假设，例如：X属于某个类别C
- 在分类问题我们想要决定P(H|**X**)。
  - P(H|**X**)：在数据元组X之下假设H成立的概率，后验概率
- P(H) (先验概率)
  - 例.,它代表不管任何讯息，H都成立的概率
- P(**X**): X在样本数据集合中出现概率
- P(**X**|H)：当假设H成立下，样本X出现的概率

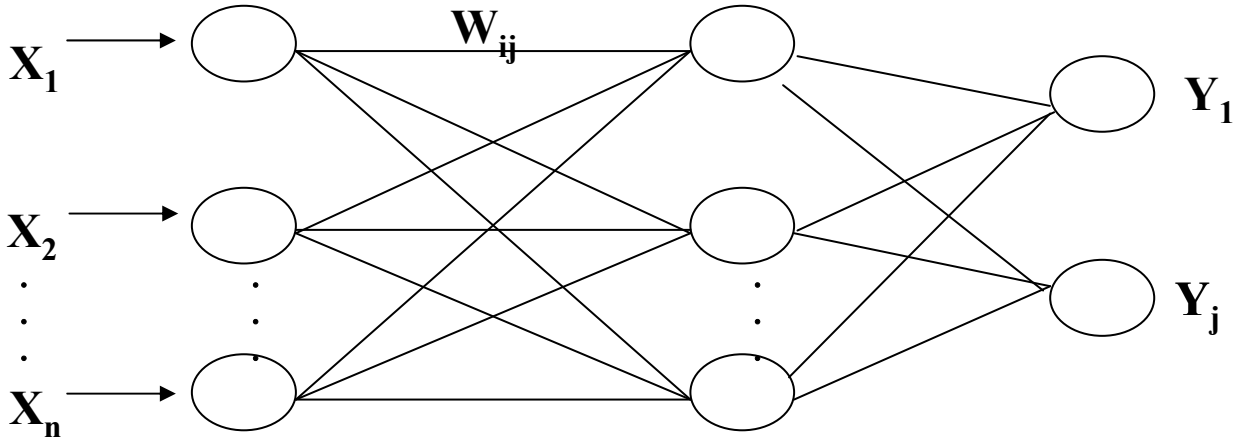
- 贝叶斯分类利用统计学中的贝叶斯定理，来预测元组归属某个类别的概率。
  - 即：给定一个样本，计算该样本属于一个特定的类别的概率。
- 假设**X**为训练数据,透过贝叶斯理论假设H的后验概率为

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$$

- 预测**X**属于类C_j当且仅当概率 $P(C_i|\mathbf{X})$ 在所有k个类别概率$P(C_k|X)$中为最高
- 朴素贝叶斯分类：假设每个属性之间都是相互独立的。

神经网络学习

- 神经网络:一组互相连接的输入输出单元 (input/output units)，而每个单元都附带有一个权重 (weight)
- 后向传播:一种神经网络 (neural network) 的学习方法
- 在学习的过程，权重会被更新以便能正确预测输入值的类别
- 类神经网络也被称为连接者学习 (connectionist learning)，因为它将所有的单元连接起来

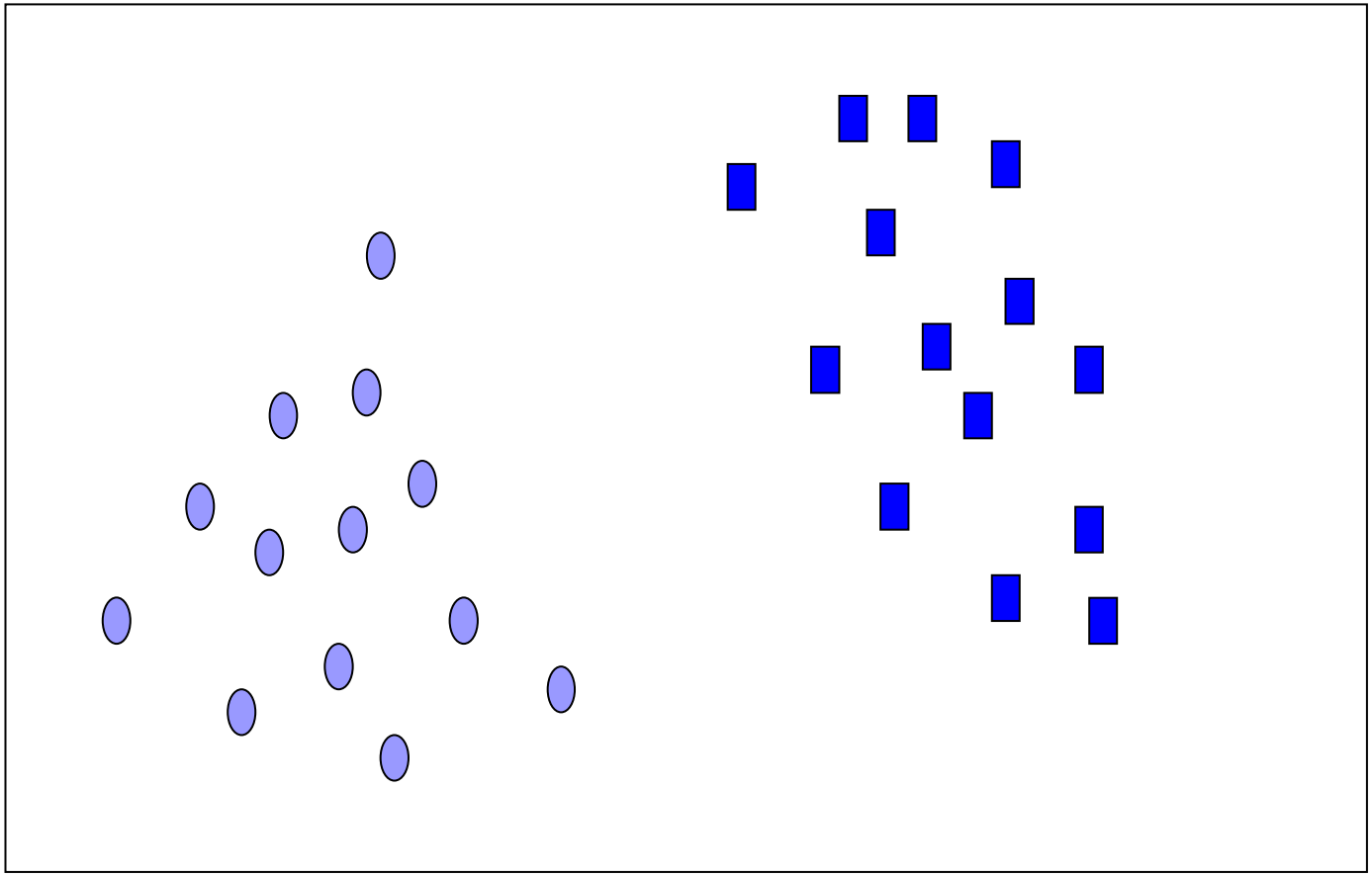


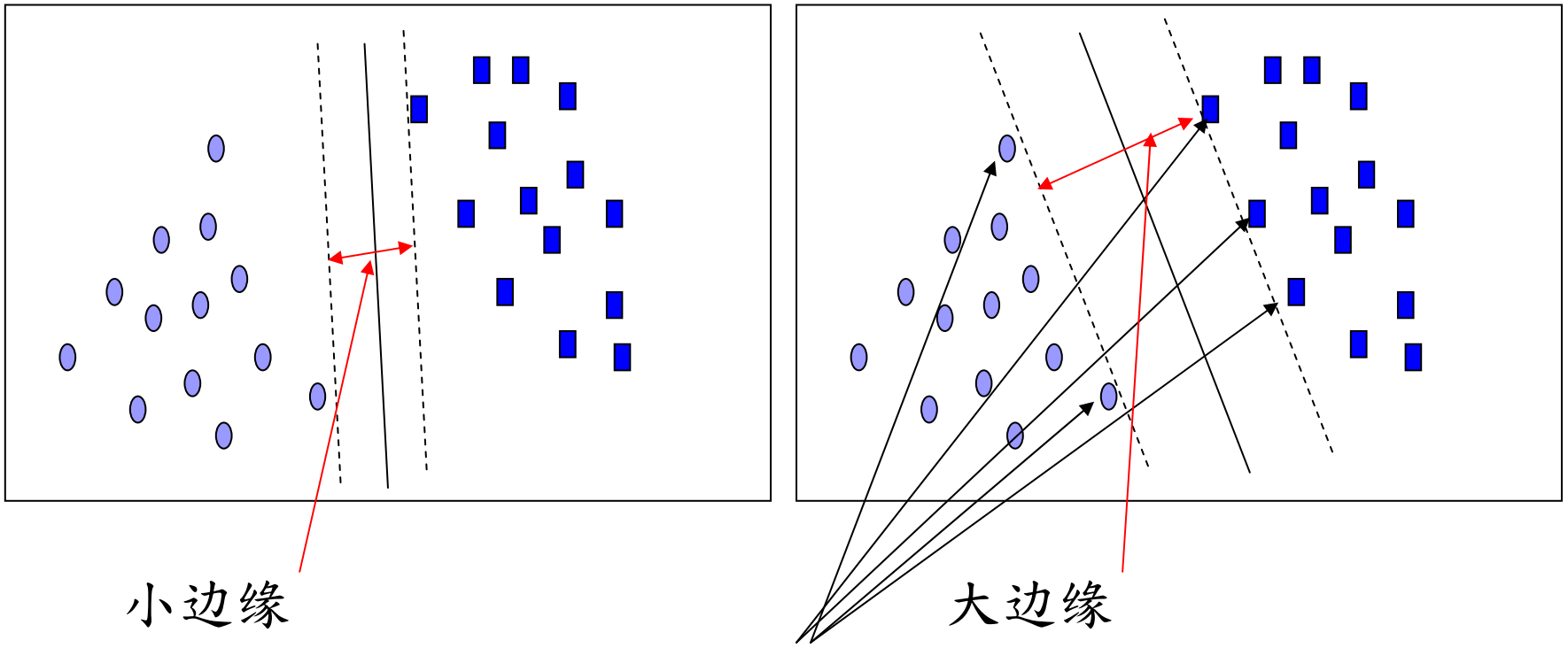
- 优势
  - 对噪声值的容忍度
  - 对未知数据模式的分类能力
  - 非常适用于连续值的数据
  - 成功的应用于现实世界数据
- 劣势
  - 需要长的训练时间
  - 需要大量的参数，且这些参数主要靠经验确定
  - 低理解度，一般很难去理解权重与隐藏单元背后的意义

SVM—支持向量机

- 一个在分类线性与非线性数据的有效方法
- 利用一个非线性映射将原训练数据对应到较高的维上
  - 一个数据集合被认为是n维的，数据在这个n维空间中被分为两类；SVM的目的是找到一个n-1维的超平面，来划分n维向量空间的数据。
- 透过适当非线性映射至较高维度，两种类别的数据都可以透过超平面进行分裂
- 支持向量机利用支持向量 (support vectors) 与边缘 (margins) 来寻找超平面

SVM一般原理





支持向量

- 特色:训练时间需要很久，但是由于能够描述复杂非线性决策边缘，支持向量机具有很高的正确率
  - 可用于分类或预测
- 应用:
  - 手写数字辨识
  - 目标识别
  - 时间序列预测

大纲

- 分类与预测
- 用决策树归纳分类
- 其他分类方法
- 预测
- 评估分类器或预测器的准确率

什么是预测？

- 预测是构造和使用模型来预测一个连续值，而不是一个分类标号。
- 预测和分类的异同
  - 相同点
    - 两者都需要构建模型
    - 都用模型来估计未知值
    - 预测当中主要的估计方法是回归分析
      - 线性回归和多元回归
      - 非线性回归
  - 不同点
    - 分类法主要是用来预测类标号（分类属性值）
    - 预测法主要是用来估计连续值（量化属性值）

大纲

- 分类与预测
- 用决策树归纳分类
- 其他分类方法
- 预测
- 评估分类器或预测器的准确率

评估分类器或预测器的准确率

- 测试(Holdout)法
  - 将数据随机切成两个独立部分
    - 训练数据 (例., 2/3) 用于建立模型
    - 测试数据 (例., 1/3) 用于评估模型的正确率
  - 随机取样: 另一种测试法
    - 进行k次测试，正确率为k次的平均
- 交叉验证 (*k*-fold, k = 10 为最普遍)
  - 随机将原始数据分裂成k个互不相交的子集 (folds) $D_1, D_2, \cdots, D_k$ ，每个大小相近
  - 当执行第i次的时候，将D_i设为测试数据，剩下的当作训练数据。
  - 分层交叉验证:对原始数据进行分裂，而每个互不相交部分的数据类别分布大约相似

- 自助法 (Bootstrap)
  - 适用于小数据集
  - 使用有放回均匀抽样(uniformly with replacement) 的方式从原始数据选取训练样本
    - 每一次被选为当作训练数据的元组都具有相同的机率
- 有许多不同的自助法，而最常用的是.632自助
  - 假设数据包含个d元组，我们会利用有放回均匀抽样的方式从原始数据选择d个元组，把这些元组当作训练数据，而把不在训练数据中出现的元组当作测试数据。假设我们重复许多次这样的步骤，63.2%的原始数据元组会在自助区域中，而剩下的36.8%的原始数据会在测试数据中(因为 $(1 - 1/d)^d \approx e^{-1} = 0.368$)

总结

- 分类是一个被广泛学习的问题(主要被用于统计学,机器学习和神经网络)
- 分类可能是数据挖掘技术中使用最广泛中的一个,具有很大的可扩充性
- 可伸缩性仍是数据库应用中的一个重要问题,因此把分类和数据库技术结合起来一个很有前途的主题