商务智能
Business Intelligence

分类：基本概念、决策树、贝叶斯方法、模型评价

朱卫平 博士
计算机学院
武汉大学

分类

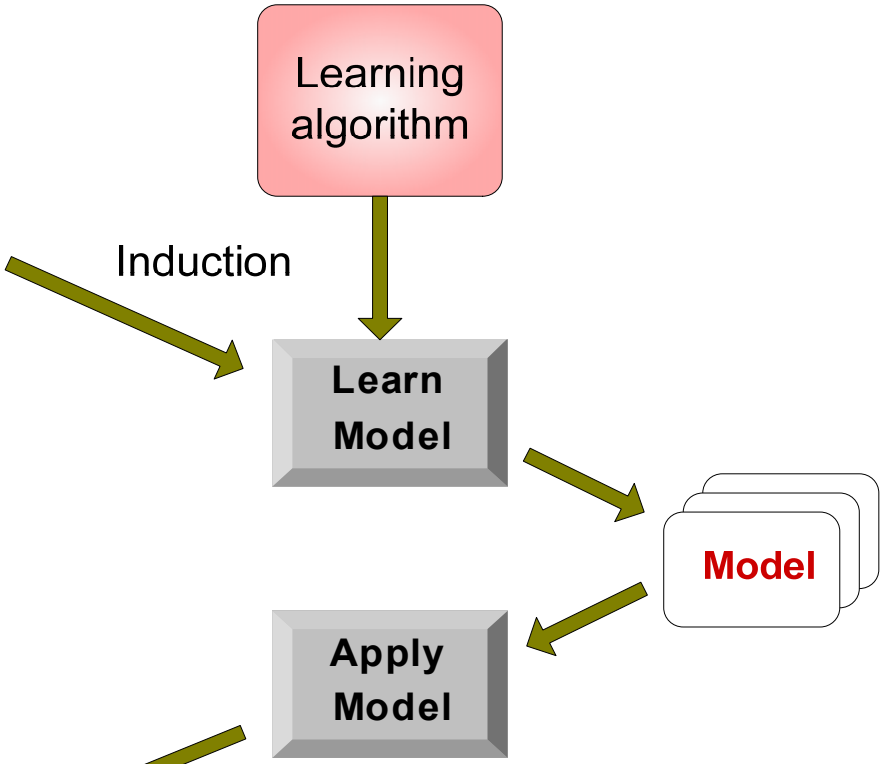
- 分类是利用一个分类函数（分类模型、分类器），该模型能把数据库中的数据映射到一个给定类别中。

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



训练集与测试集

- 训练集：数据库中为建立模型而被分析的数据元组形成训练集。
- 训练集中的单个元组称为训练样本，每个训练样本有一个类别标记。
- 一个具体样本的形式可为：(v1, v2, ..., vn; c)；其中vi表示属性值，c表示类别。
- 测试集：用于评估分类模型的准确率

数据分类：两步过程 (1)

- 第一步，建立一个模型，描述预定数据类集和概念集
  - 假定每个元组属于一个预定义的类，由一个类标号属性确定
  - 学习模型可以用分类规则、决策树或数学公式的形式提供

数据分类：两步过程 (2)

- 第二步，使用模型，对将来的或未知的对象进行分类
  - 首先评估模型的预测准确率
    - 对每个测试样本，将已知的类标号和该样本的学习模型类预测比较
    - 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“过分适应数据”的情况
  - 如果准确性能被接受，则分类规则就可用来对新数据进行分类

有监督的学习 vs. 无监督的学习

- 有监督的学习（用于分类）
  - 模型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）
  - 每个训练样本的类编号是未知的，要学习的类集合或数量也可能是事先未知的
  - 通过一系列的度量、观察来建立数据中的类编号或进行聚类

分类模型的构造方法

- **机器学习方法：**
  - 决策树法
  - 规则归纳

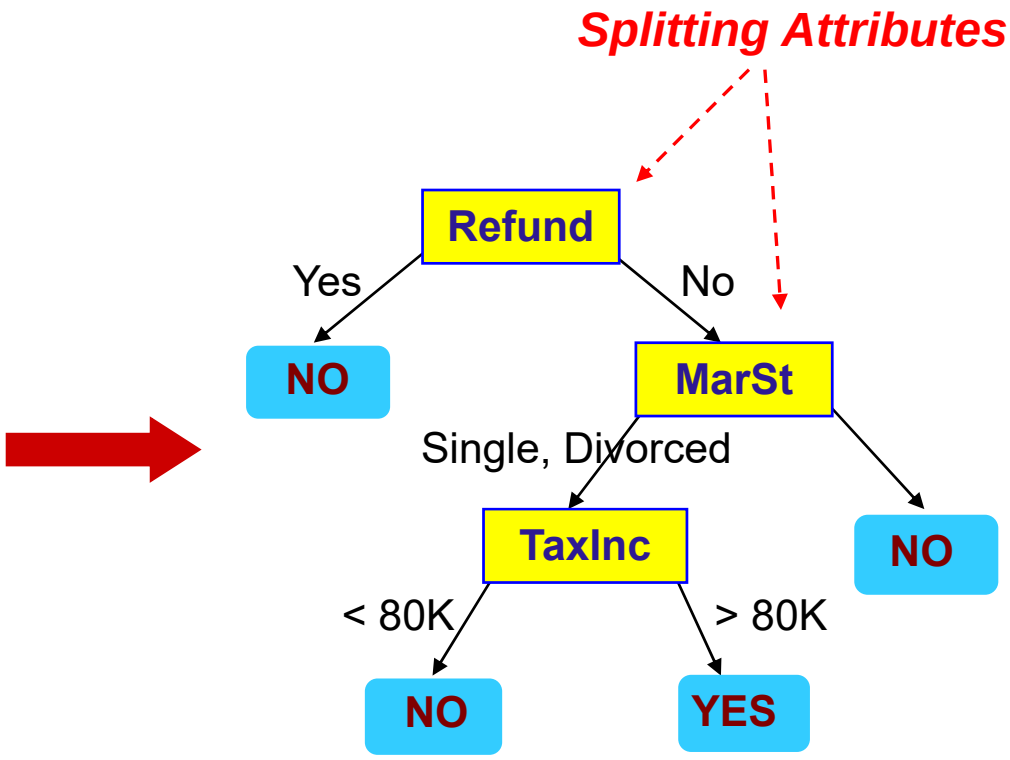
- **统计方法：**知识表示是判别函数和原型事例
  - 贝叶斯法
  - 非参数法(近邻学习或基于事例的学习)

- **神经网络方法：**
  - BP算法,模型表示是前向反馈神经网络模型

一个决策树的例子

Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

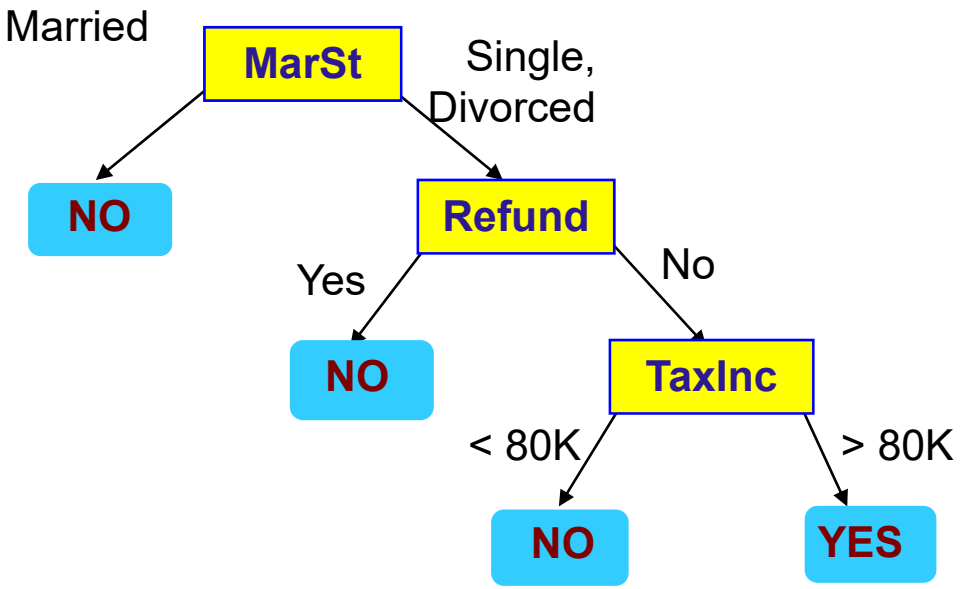
训练数据



模型：决策树

另一个决策树的例子

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



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
  - 每个树叶节点代表类或类分布
- 决策树的生成由两个阶段组成
  - 决策树构建
    - 开始时，所有的训练样本都在根节点
    - 递归的通过选定的属性，来划分样本
  - 树剪枝
    - 许多分枝反映的是训练数据中的噪声和孤立点，树剪枝试图检测和剪去这种分枝

用决策树归纳分类

- 决策树的使用：对未知样本进行分类
  - 通过将样本的属性值与决策树相比较

为了对未知数据对象进行分类识别，可以根据决策树的结构对数据集中的属性进行测试，从决策树的根节点到叶节点的一条路径就形成了相应对象的类别测试。决策树可以很容易转换为分类规则

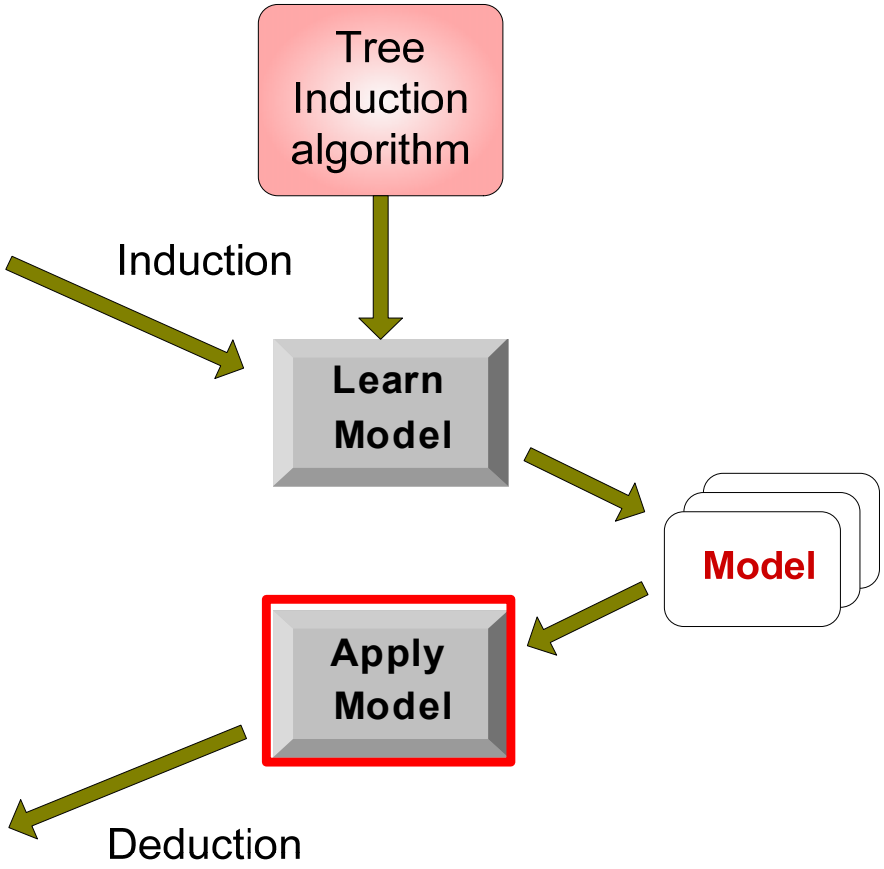
决策树分类任务

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

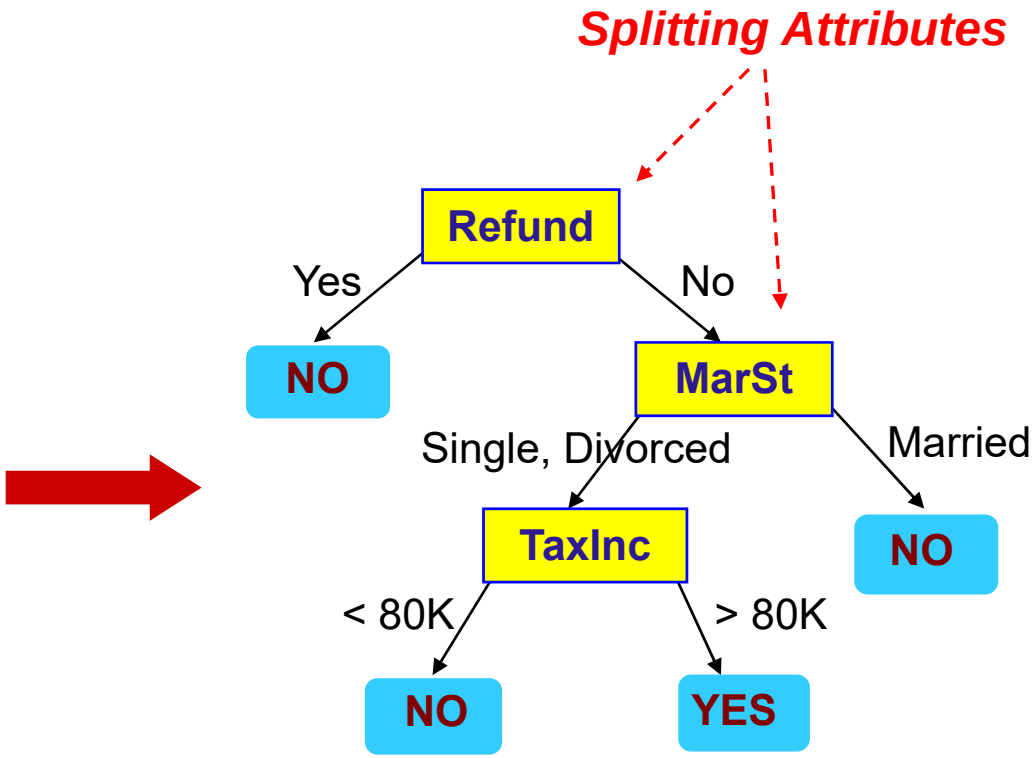
Test Set



应用决策树进行分类

Tid	Refund	Marital Status	Taxable Income	Cheat
		categorical	continuous	class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

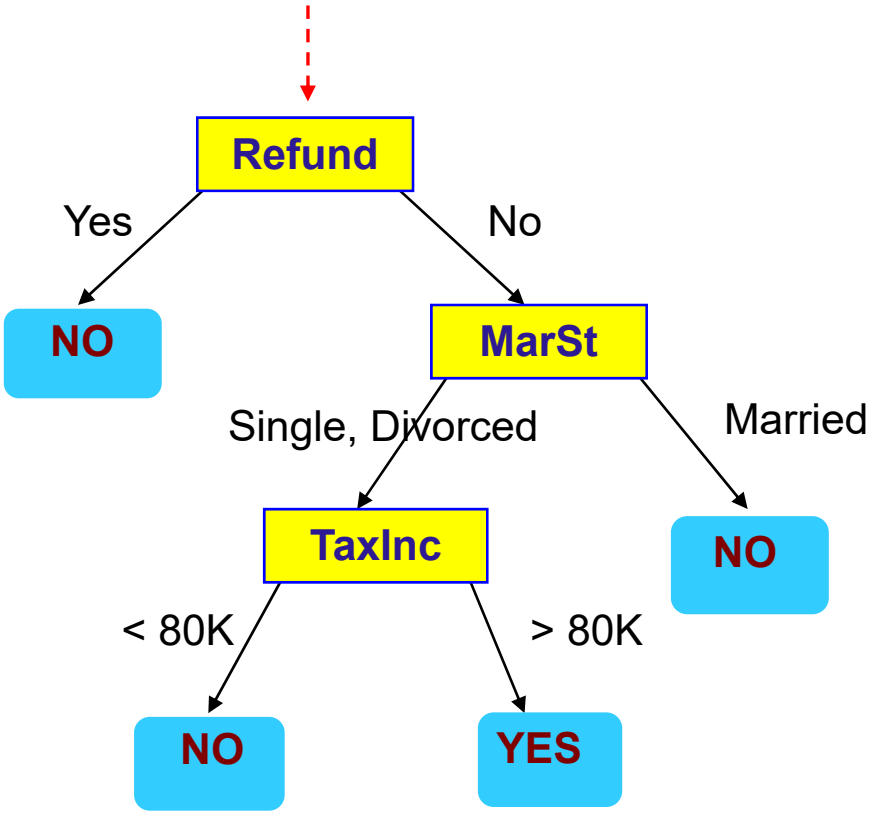
训练数据



模型：决策树

应用决策树进行分类

Start from the root of tree.



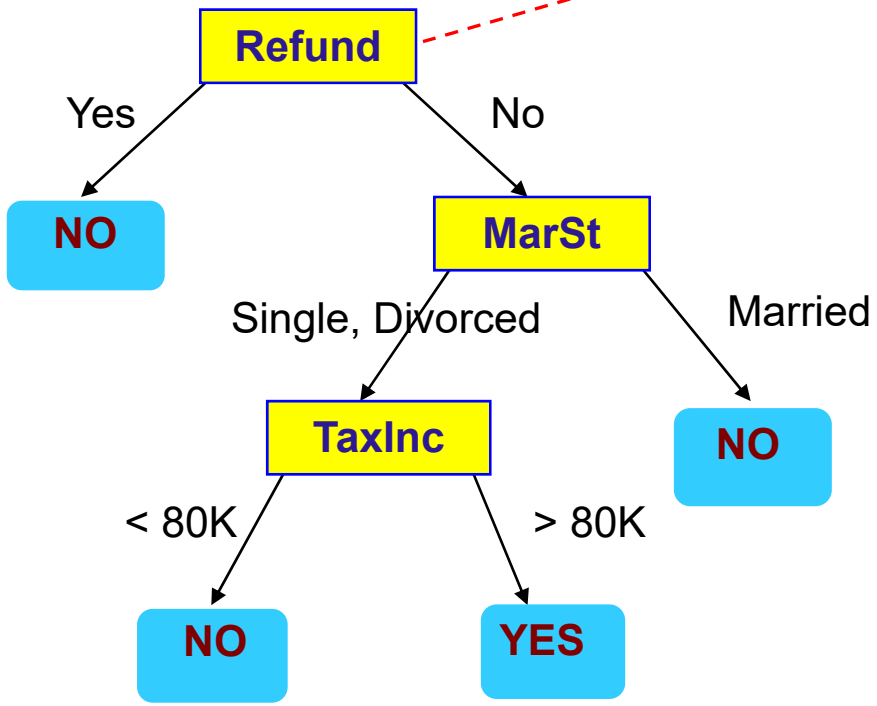
测试数据

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

应用决策树进行分类

测试数据

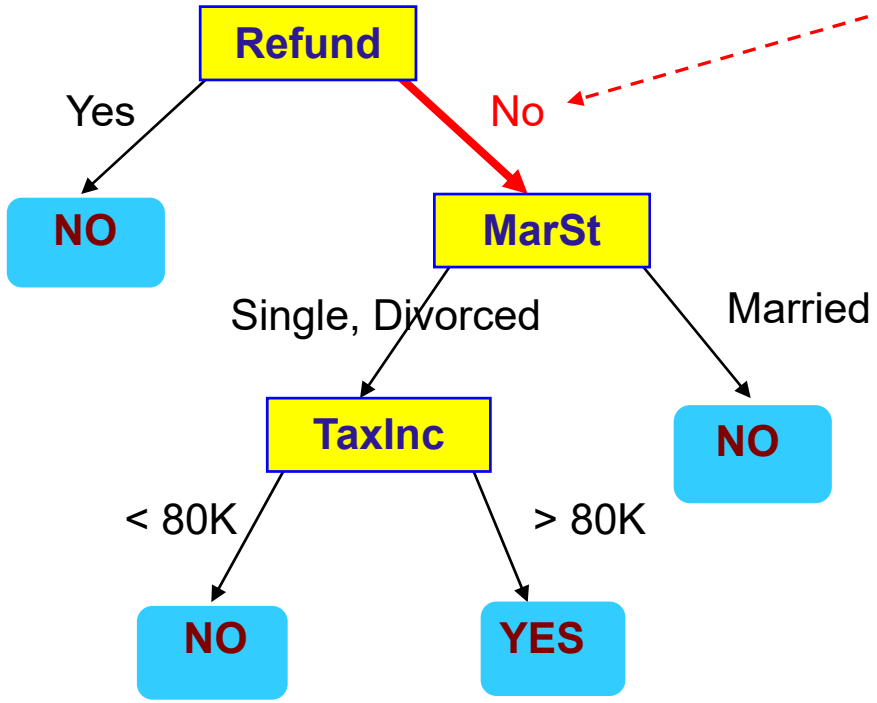
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



应用决策树进行分类

测试数据

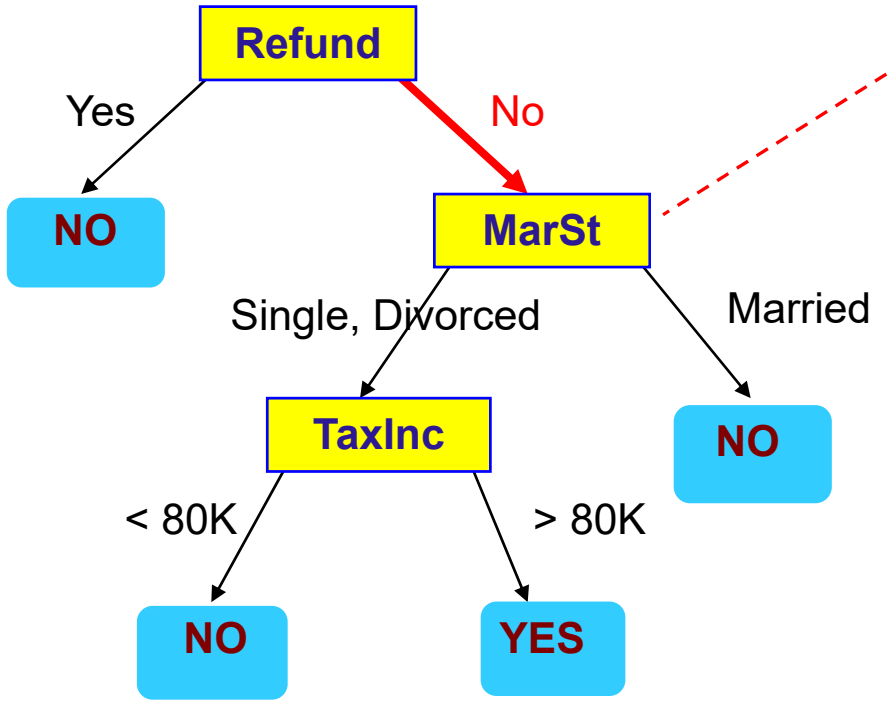
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



应用决策树进行分类

测试数据

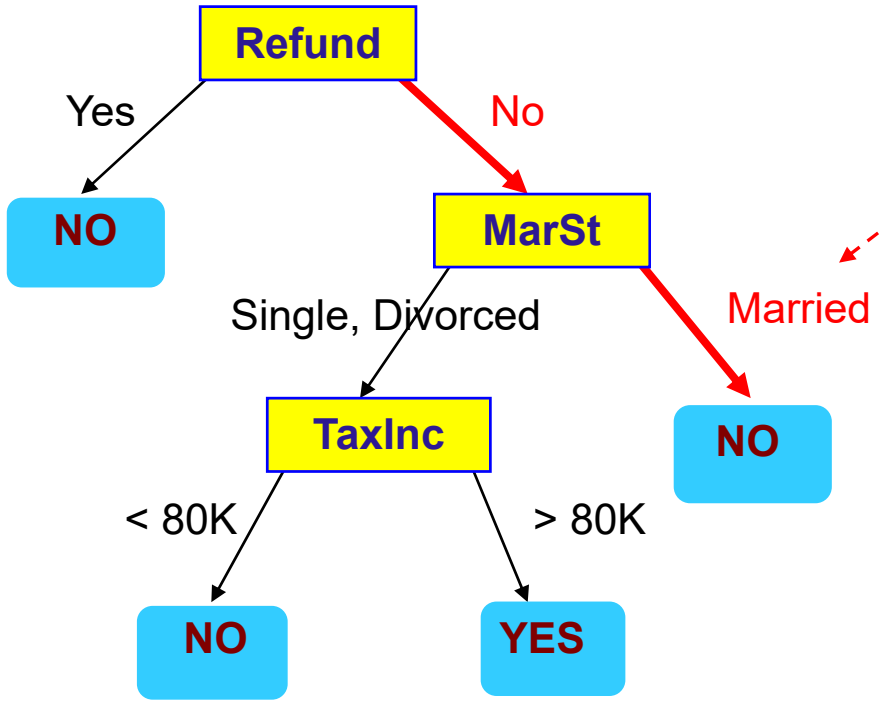
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



应用决策树进行分类

测试数据

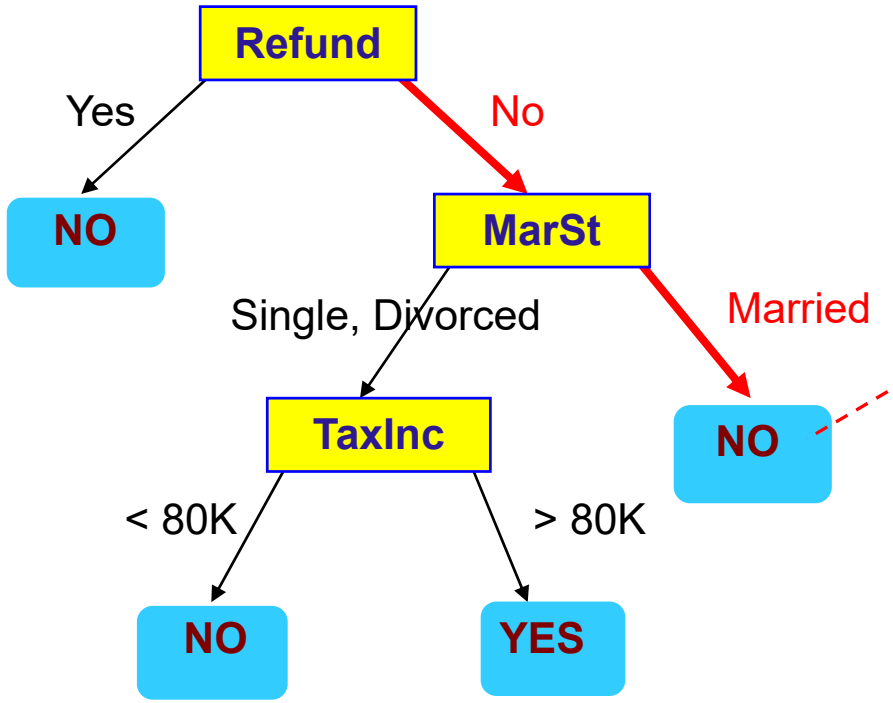
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



应用决策树进行分类

测试数据

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to “No”

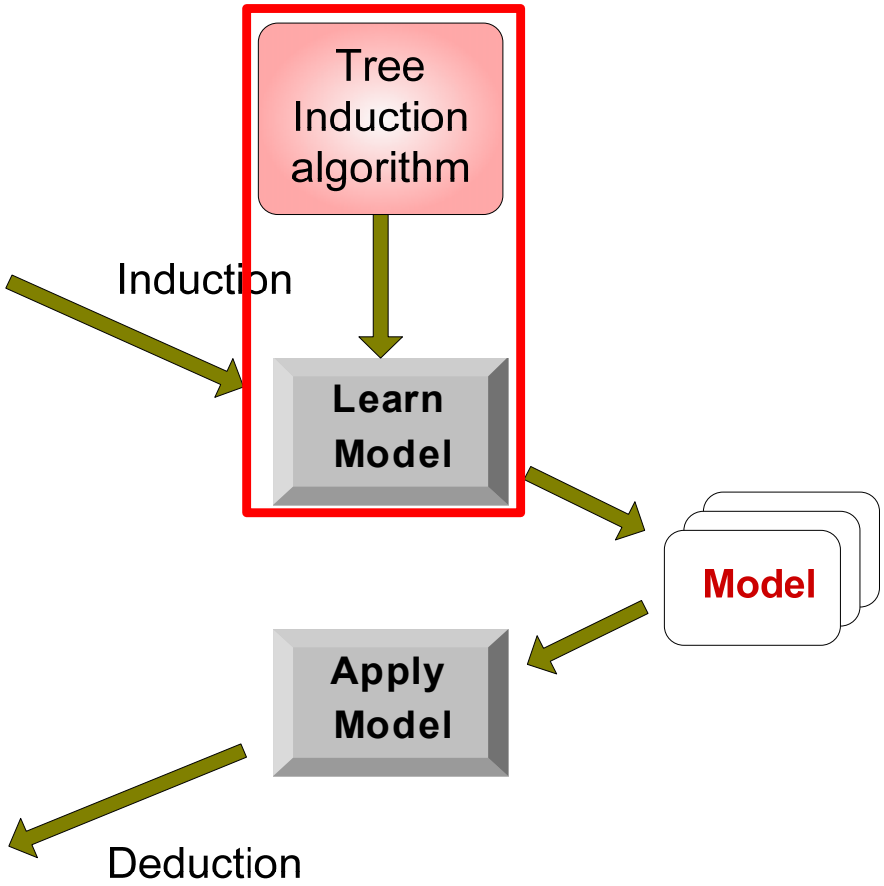
决策树分类

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



决策树算法

- 决策树算法:
  - **Hunt算法**
  - **信息增益——Information gain （ID3）**
  - **增益比率——Gain ration（C4.5）**
  - **基尼指数——Gini index (CART，SLIQ，SPRINT)**

决策树算法

算法generate_decision_tree .由数据分区D中的训练元组产生决策树

输入：

- 数据分区D训练元组和它们对应类标号的集合。
- Attribute_list,候选属性的集合
- Attribute_selection_method,一个确定“最好地”划分数据元组为个体类的分裂准则的过程。这个准则由分裂属性(splitting_attribute)和分裂点或划分子集组成。

输出：一棵决策树

方法：

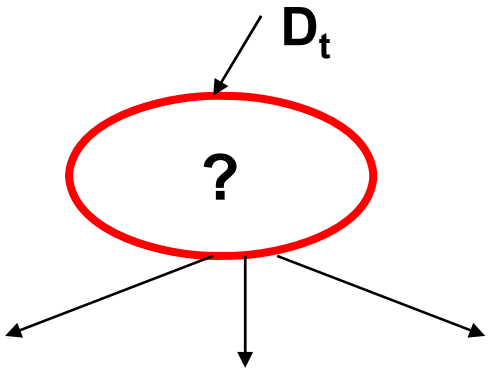
（1）创建一个结点N；
（2）If D中的元组都在同一类C中then
（3） 返回 N作为叶节点，以类C标记；
（4）if attribute_list为空 then
（5） 返回N作为叶节点，标记为D中的多数类； //多数表决
（6）使用Attribute_selection_method(D, attribute_list),找到“最好的” splitting_criterion；
（7）用splitting_criterion标记结点N；
（8）if splitting_attribute是离散值的，并且允许多路划分then //不限于二叉树
（9） attribute_list←attribute_list-splittng_attribute; //删除分裂属性
（10）for splitting_criterion的每个输出j //划分元组并对每个分区产生子树
（11） 设D_j是D中满足输出j的数据元组的集合； //一个分区
（12） if D_j为空then
（13） 加一个树叶到结点N，标记为D中的多数类；
（14） else 加一个由generate_decision_tree (D_j, attribute_list)返回的结点到N；
endfor
(15) 返回N；

Hunt算法

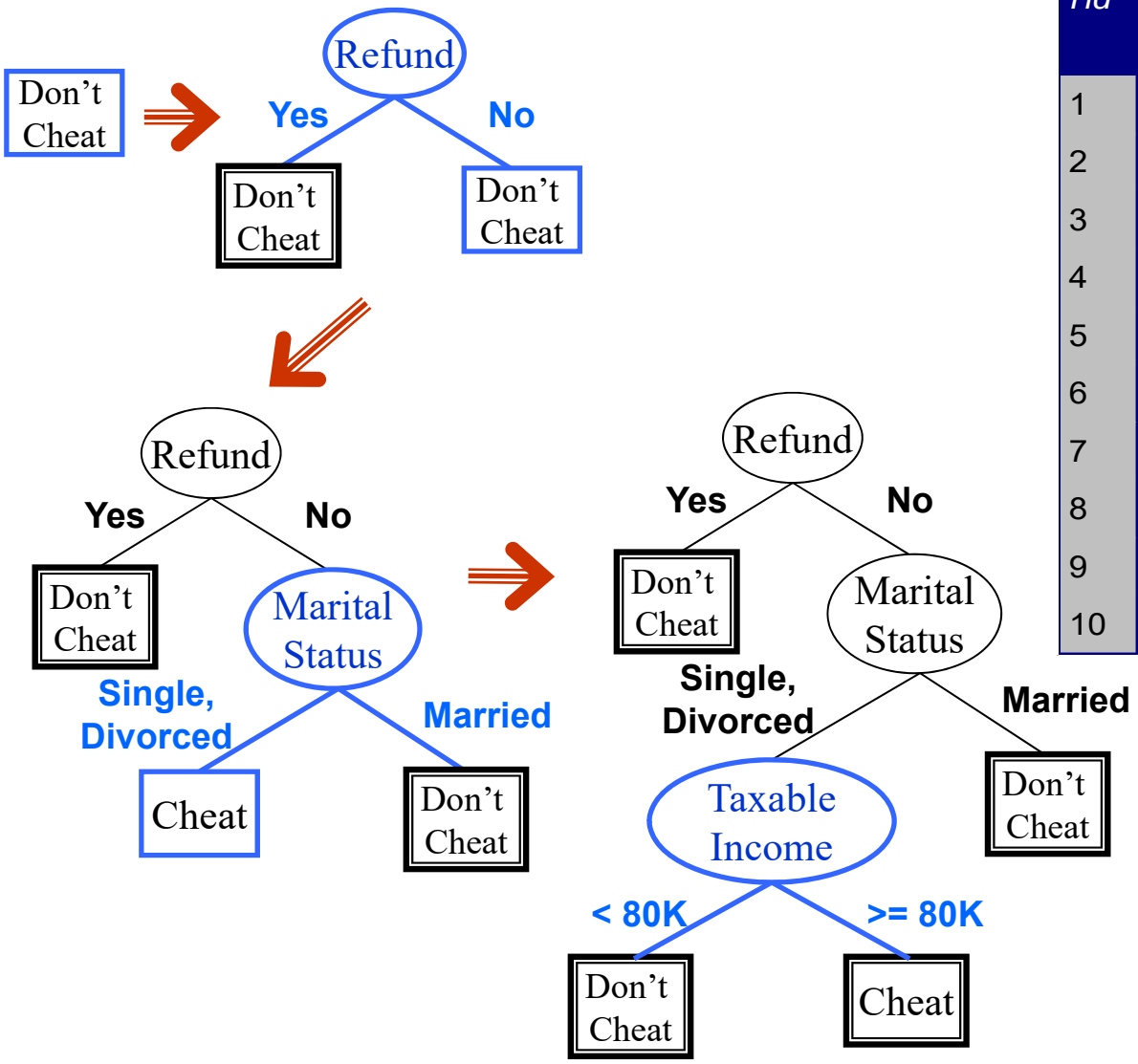
- 设 D_t 是与结点 t相关联的训练记录集
- 算法步骤:
  - 如果D_t 中所有记录都属于同一个类 y_t, 则t是叶结点，用y_t标记
  - 如果 D_t 中包含属于多个类的记录，则选择一个属性作为测试条件，将记录划分成较小的子集。

    对于测试条件的每个输出，创建一个子结点。然后递归地调用该算法

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Hunt算法



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

决策树

- Hunt算法采用贪心策略构建决策树.
  - 在选择划分数据的属性时，采取一系列局部最优决策来构造决策树.

- 决策树归纳的设计问题
  - 如何分裂训练记录
    - 怎样为不同类型的属性指定测试条件?
    - 怎样评估每种测试条件?
  - 如何停止分裂过程

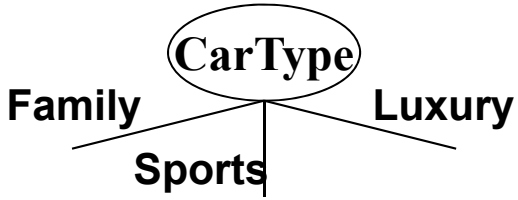
怎样为不同类型的属性指定测试条件?

- 依赖于属性的类型
  - 标称
  - 序数
  - 连续

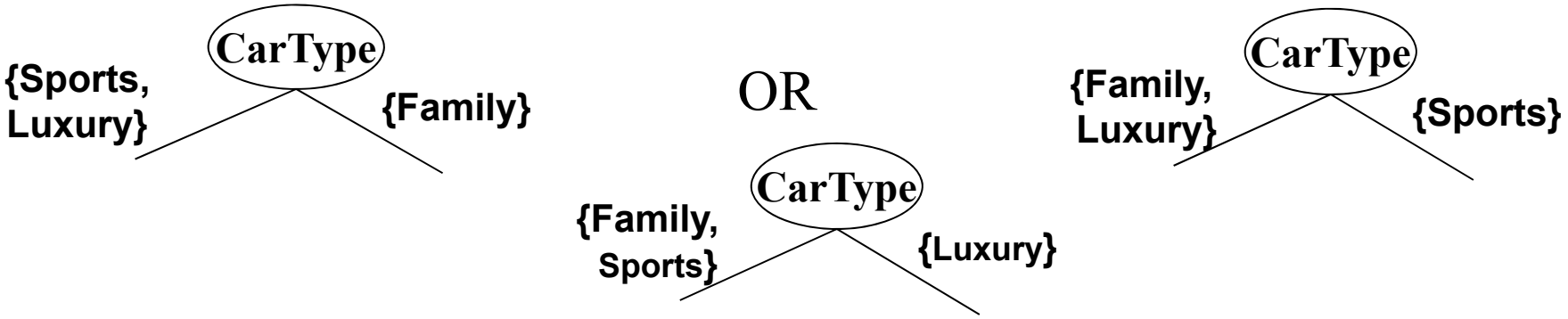
- 依赖于划分的路数
  - 2路划分
  - 多路划分

基于标称属性的分裂

- 多路划分: 划分数（输出数）取决于该属性不同属性值的个数.

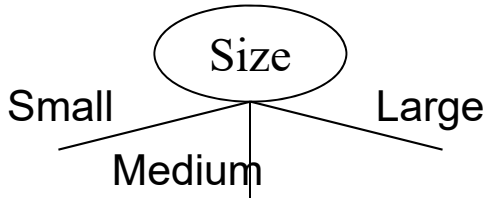


- 二元划分: 划分数为2，这种划分要考虑创建k个属性值的二元划分的所有$2^{k-1}-1$种方法.

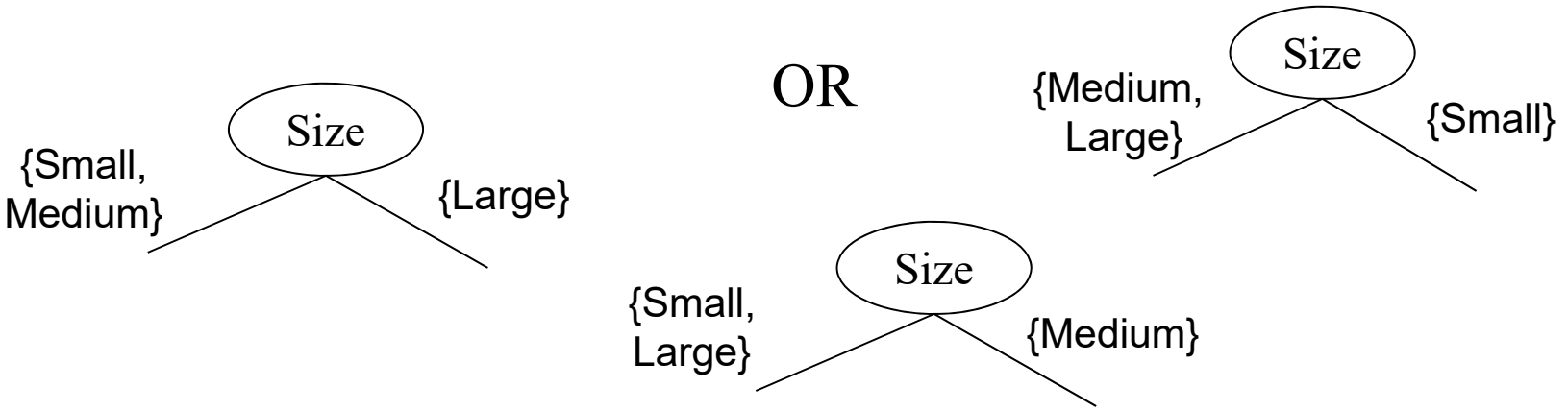


基于序数属性的划分

- 多路划分: 划分数（输出数）取决于该属性不同属性值的个数.



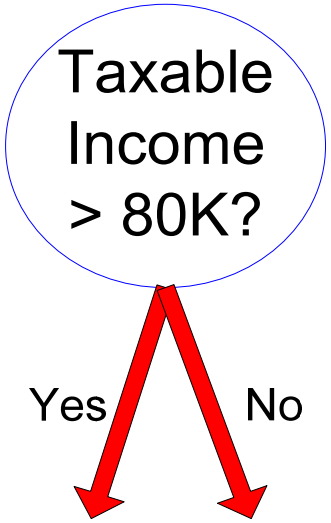
- 二元划分:



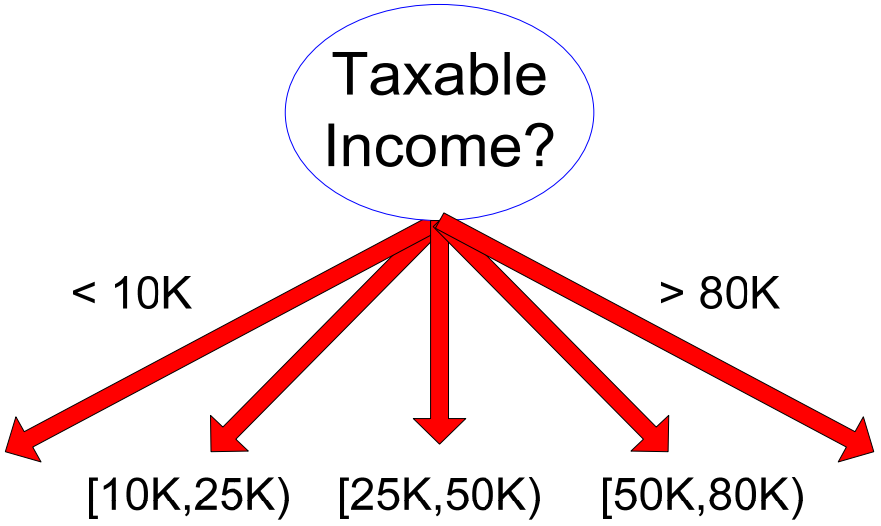
基于连续属性的划分

- 多路划分: $v_i \le A < v_{i+1}$ $(i=1,\ldots,k)$
- 二元划分: $(A < v)$ or $(A \ge v)$
  - 考虑所有的划分点，选择一个最佳划分点v

基于连续属性的划分



(i) Binary split



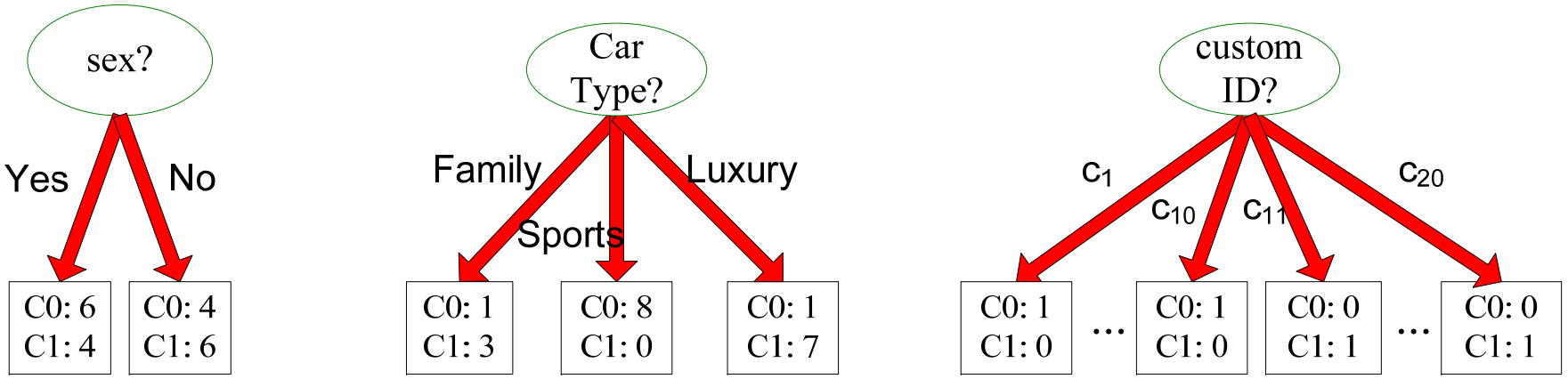
(ii) Multi-way split

决策树

- 决策树归纳的设计问题
  - 如何分裂训练记录
    - 怎样为不同类型的属性指定测试条件?
    - 怎样评估每种测试条件?
  - 如何停止分裂过程

怎样选择最佳划分？

在划分前: **10** 个记录 **class 0,10** 个记录 **class 1**



怎样选择最佳划分？

- 选择最佳划分的度量通常是根据划分后子结点不纯性的程度。不纯性的程度越低，类分布就越倾斜
- 结点不纯性的度量:

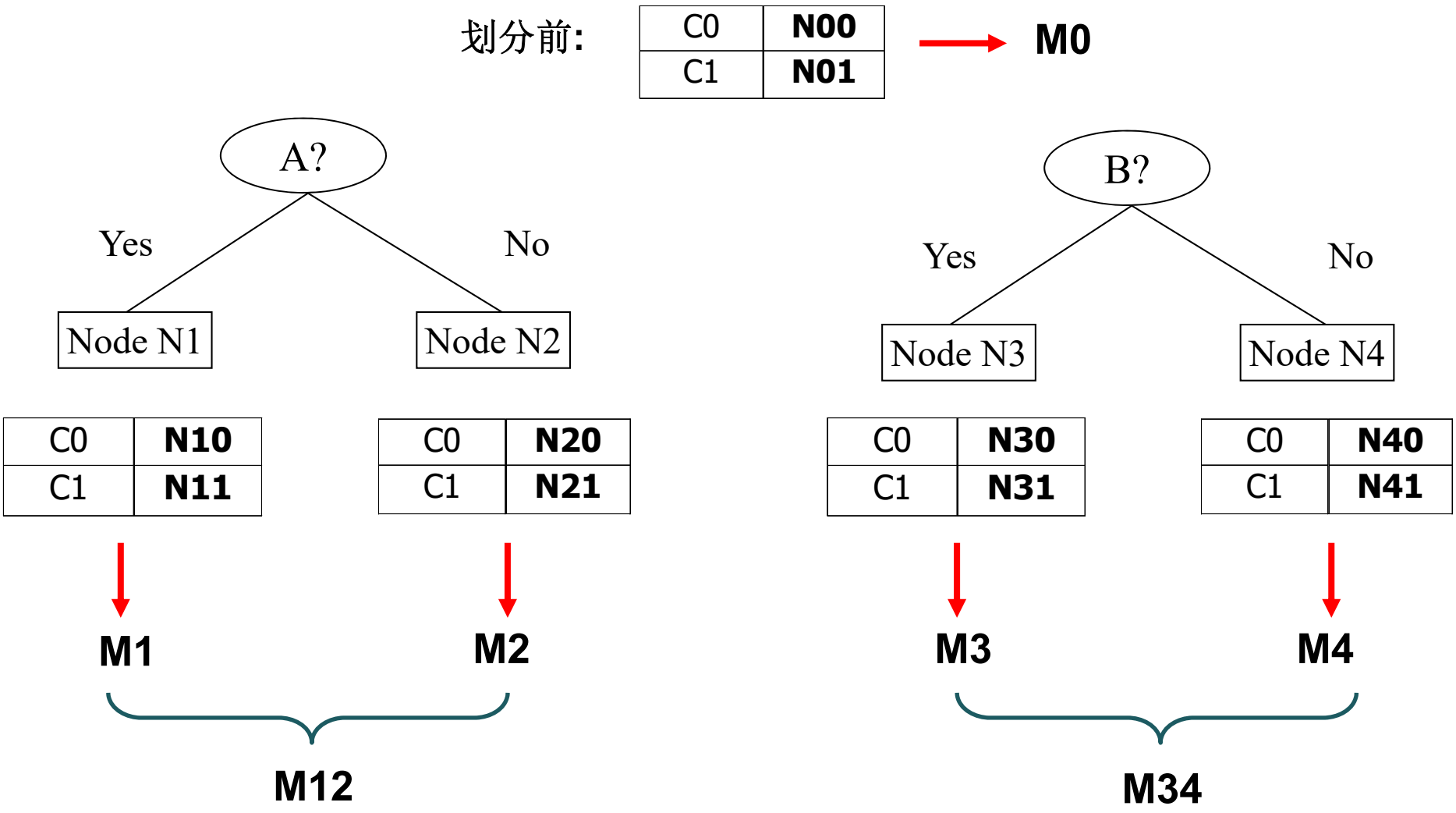
C0: 5 C1: 5

不纯性大

C0: 9 C1: 1

不纯性小

怎样找到最佳划分？


划分前:
C0
C1
N00
N01
M0
A?
Yes
No
Node N1
Node N2
B?
Yes
No
Node N3
Node N4
C0
C1
N10
N11
C0
C1
N20
N21
C0
C1
N30
N31
C0
C1
N40
N41
M1
M2
M3
M4
M12
M34
Gain = M0 – M12 vs M0 – M34

结点不纯性的测量

- Gini
- Entropy
- classification error

不纯性的测量: GINI

- 给定结点t的Gini值计算 :

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

($p(j \mid t)$ 是在结点t中，类j发生的概率).

  - 当类分布均衡时，Gini值达到最大值 $(1 - 1/n_c)$
  - 相反当只有一个类时，Gini值达到最小值0

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

计算 GINI的例子

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – (1/6)² – (5/6)² = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – (2/6)² – (4/6)² = 0.444

基于 GINI的划分

- 当一个结点 p 分割成 k 个部分 (孩子), 划分的质量可由下面公式计算

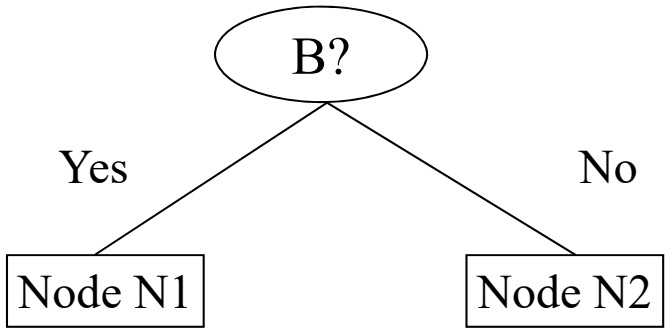
$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

n_i = 孩子结点 i的记录数,

n = 父结点 p的记录数.

二元属性: 计算 GINI

- 对于二元属性，结点被划分成两个部分
- 得到的GINI值越小，这种划分越可行.



	Parent
C1	6
C2	6
Gini = 0.500	

Gini(N1)
= 1 – $(5/7)^2$ – $(2/7)^2$
= 0.194

Gini(N2)
= 1 – $(1/5)^2$ – $(4/5)^2$
= 0.528

	N1	N2
C1	5	1
C2	2	4
Gini=0.333		

Gini $_{split}$
= 7/12 * 0.194 +
5/12 * 0.528
= 0.333

标称属性:计算Gini

- 多路划分
- 二元划分
- 一般多路划分的Gini值比二元划分小，这一结果并不奇怪，因为二元划分实际上合并了多路划分的某些输出，自然降低了子集的纯度

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split
(find best partition of values)

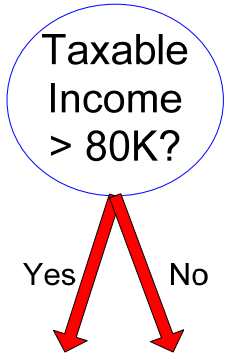
	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

连续属性: 计算 Gini

- 使用二元划分
- 划分点v选择
  - N个记录中所有属性值作为划分点
- 对每个划分进行类计数, A < v and A ≥ v
- 计算每个候选点v的Gini指标，并从中选择具有最小值的候选划分点
- 时间复杂度为(n^2)

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



连续属性: 计算 Gini

- 降低计算复杂性的方法
  - 将记录进行排序
  - 从两个相邻的属性值之间选择中间值作为划分点
  - 计算每个候选点的Gini值
  - 时间复杂度为nlogn

排序后的值 →
划分点 →

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
Taxable Income										
	60	70	75	85	90	95	100	120	125	220
	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	3 0	3 0
No	0 7	1 6	2 5	3 4	3 4	3 4	3 4	4 3	5 2	6 1	7 0
Gini	0.420	0.400	0.375	0.343	0.417	0.400	*0.300*	0.343	0.375	0.400	0.420