分类技术

电子信息与通信学院 冯 镔
fengbin@hust.edu.cn

分类的定义

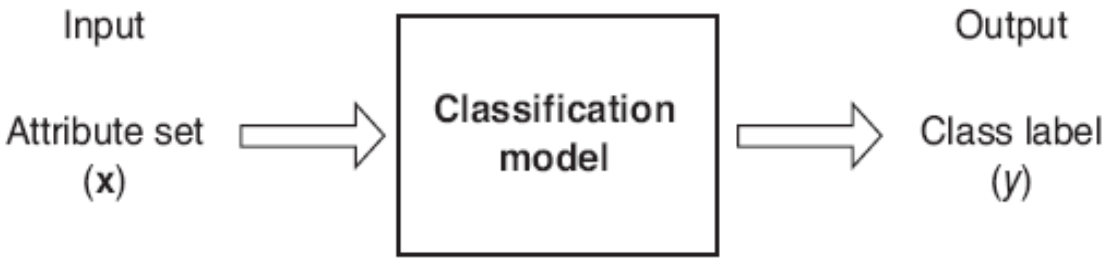
- 给定一个记录的集合（训练集）
  - 每个记录包含一系列属性，其中一个属性是类标号
- 找到一个类标号属性的模型，它是其他属性的函数
- 目标：尽可能精确的预测出一个未知记录所属的类别
  - 通常采用测试集合来检验模型的精度

分类的定义

- 分类任务就是通过学习得到一个目标函数f，把每个属性集x映射到一个预先定义的类标号y上
- Y=response, output
- X=(x1,…,xp)=predictor, input
- 回归Regression: y∈ |R
- 分类Classification: y∈{c1,…ck}

分类的定义

- 利用训练集，希望建立一个模型 $\hat{f}(x)$，当有一个输入x的时候，可以预测一个输出y

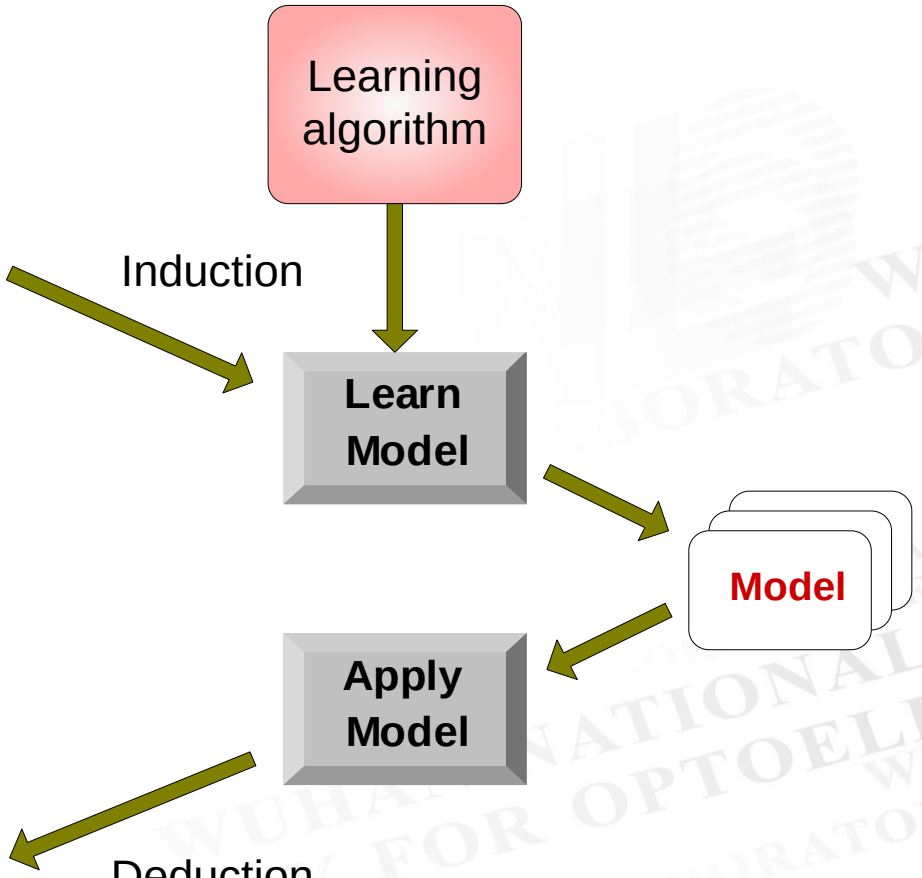


- 预测的精确
- 理解哪些属性影响到输出，如何影响

分类的定义

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

分类的定义

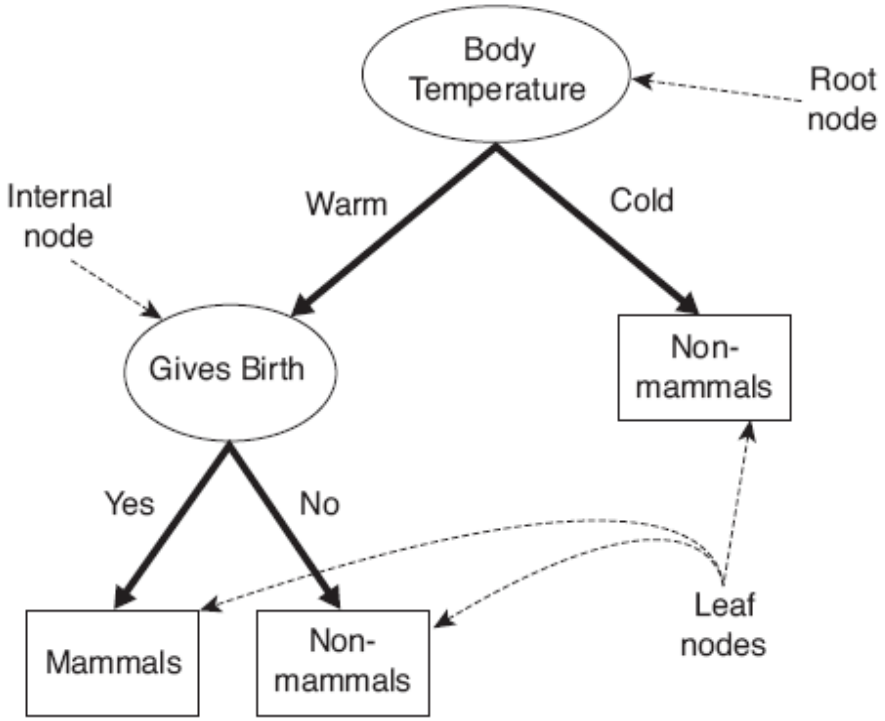
- 混淆矩阵

		预测的类	
		类 = 1	类 = 0
实际的类	类 = 1	f_{11}	f_{10}
	类 = 0	f_{01}	f_{00}

- 准确率 $= \dfrac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$ 错误率 $= \dfrac{f_{01} + f_{10}}{f_{11} + f_{10} + f_{01} + f_{00}}$

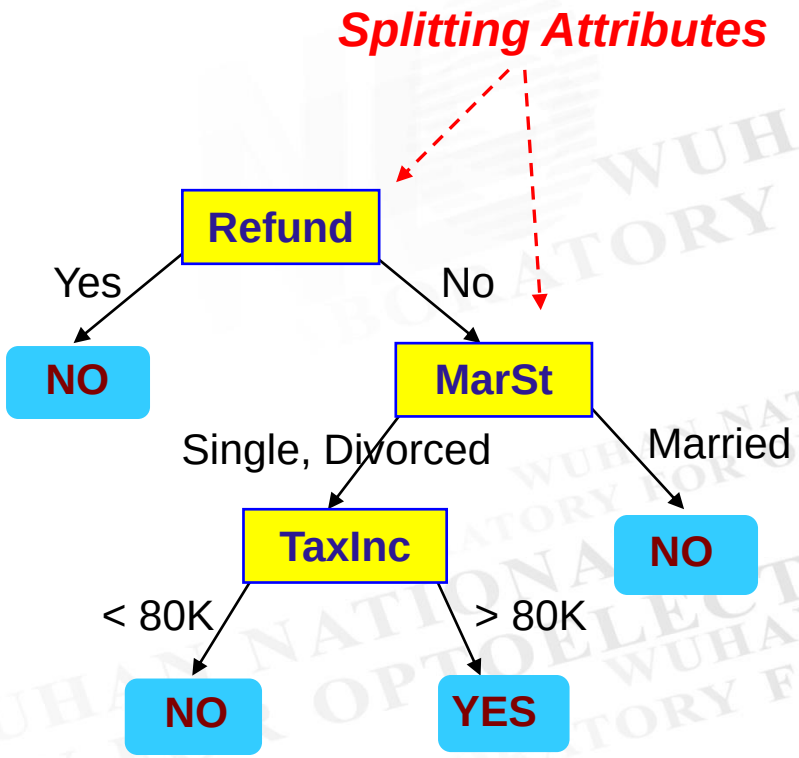
决策树

- 决策树：一组嵌套的判定规则
- 根结点
- 内部结点
- 叶结点



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

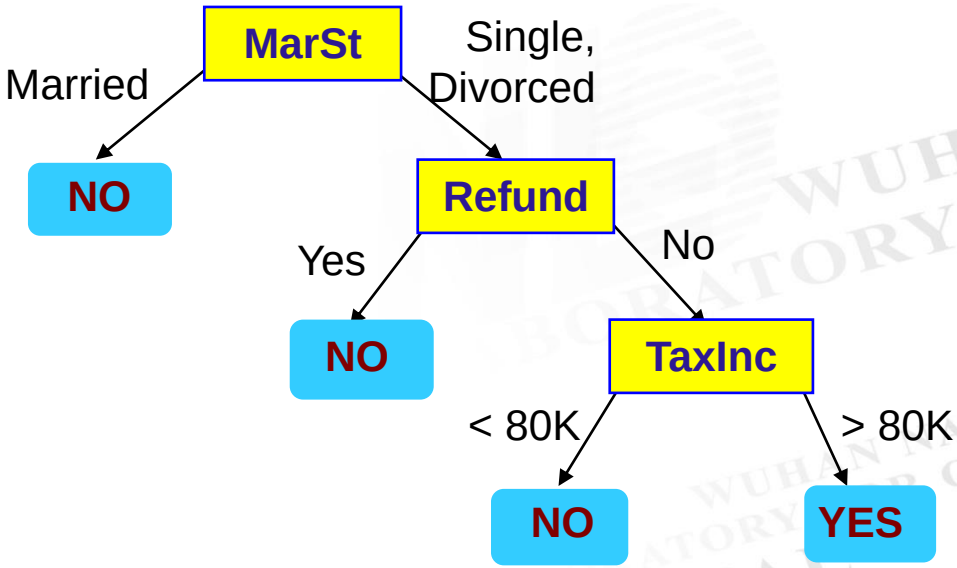


训练数据

模型：决策树

决策树的例子

Tid	Refund (categorical)	Marital Status (categorical)	Taxable Income (continuous)	Cheat (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



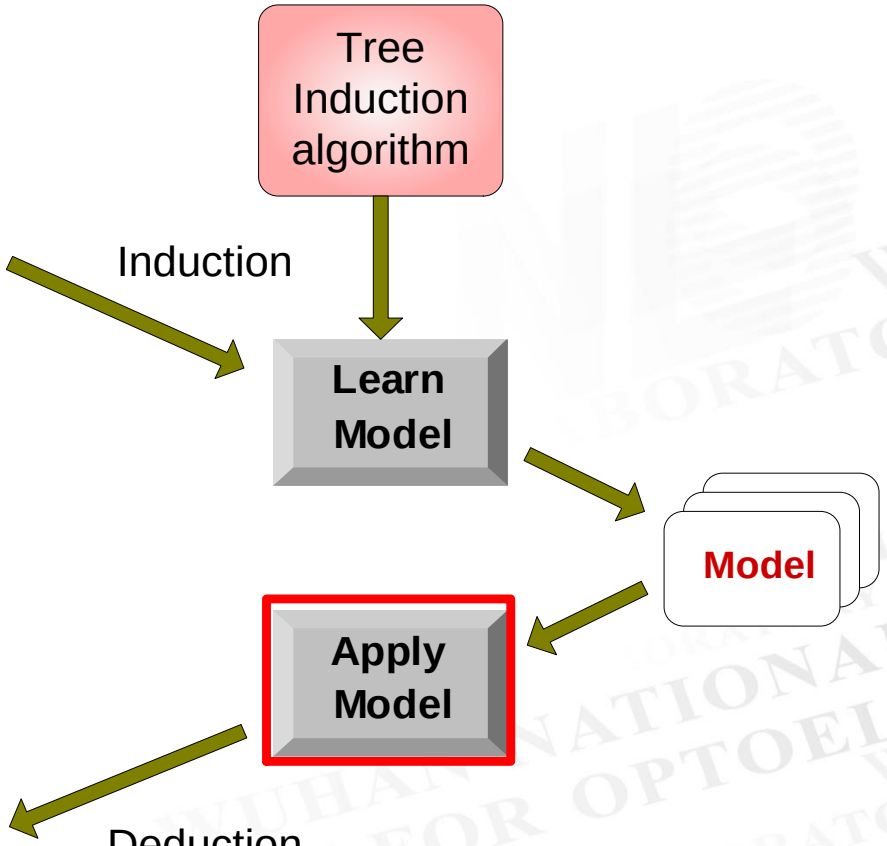
There could be more than one tree that fits the same data!

对于同一个数据集，可能会产生多个不同的树

决策树

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

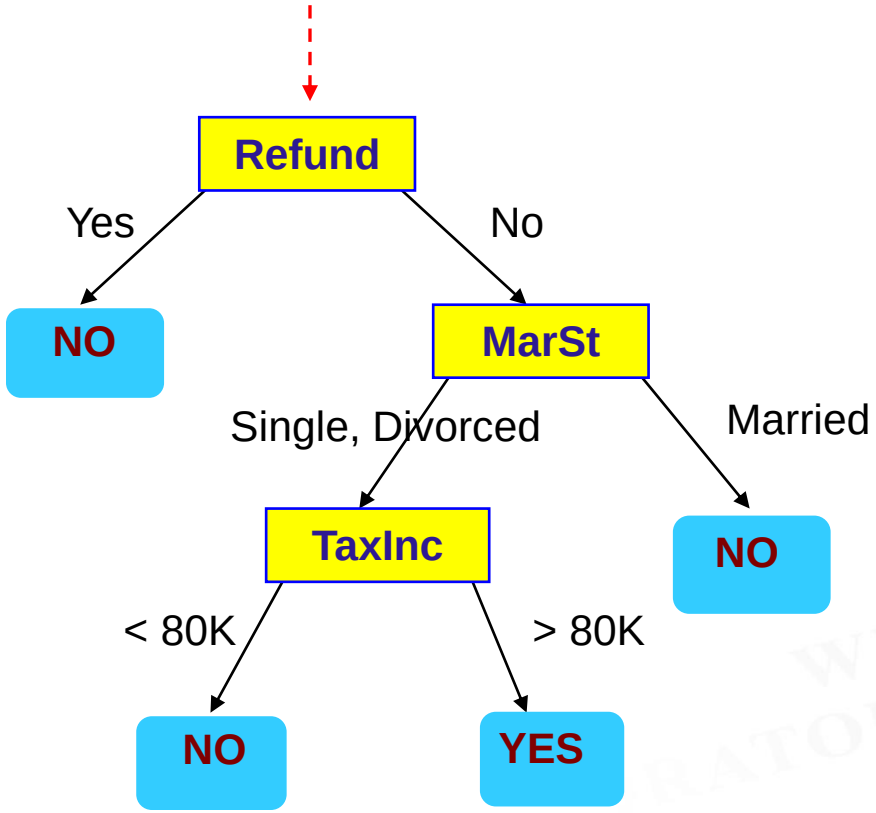


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

决策树

Start from the root of tree.



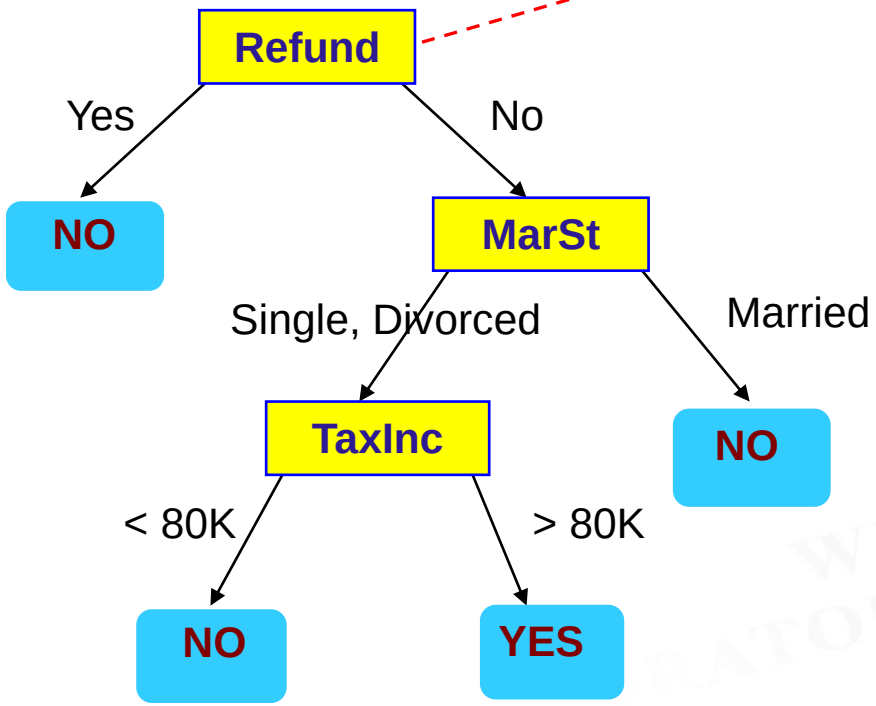
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

决策树

Test Data

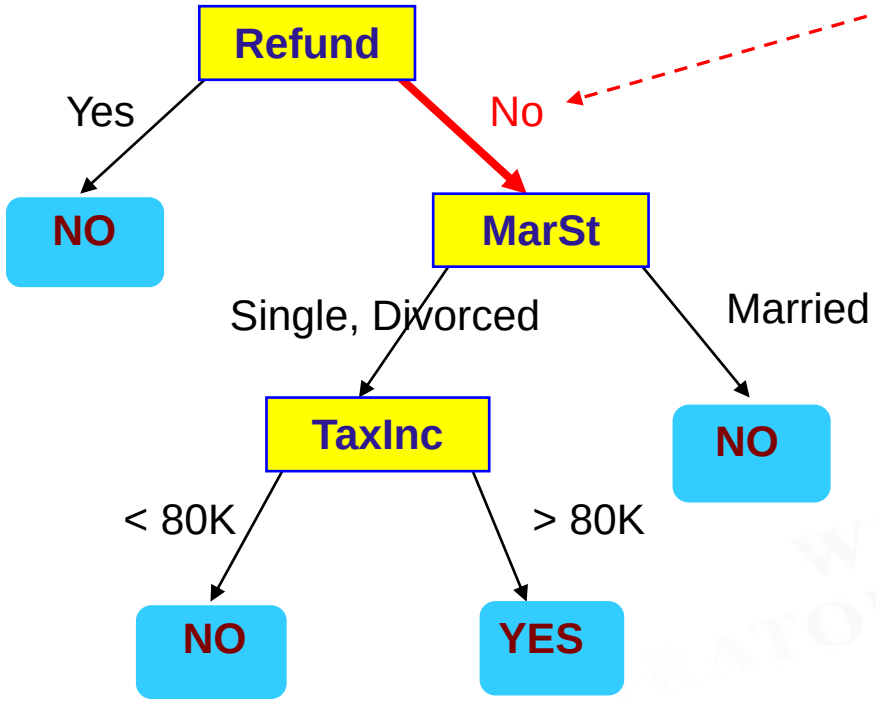
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



决策树

Test Data

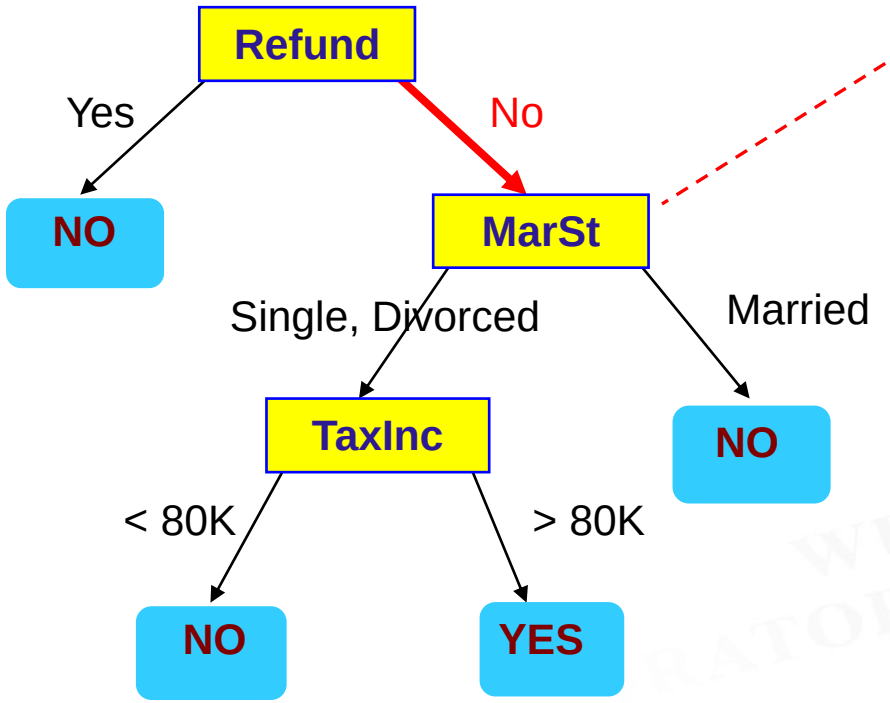
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



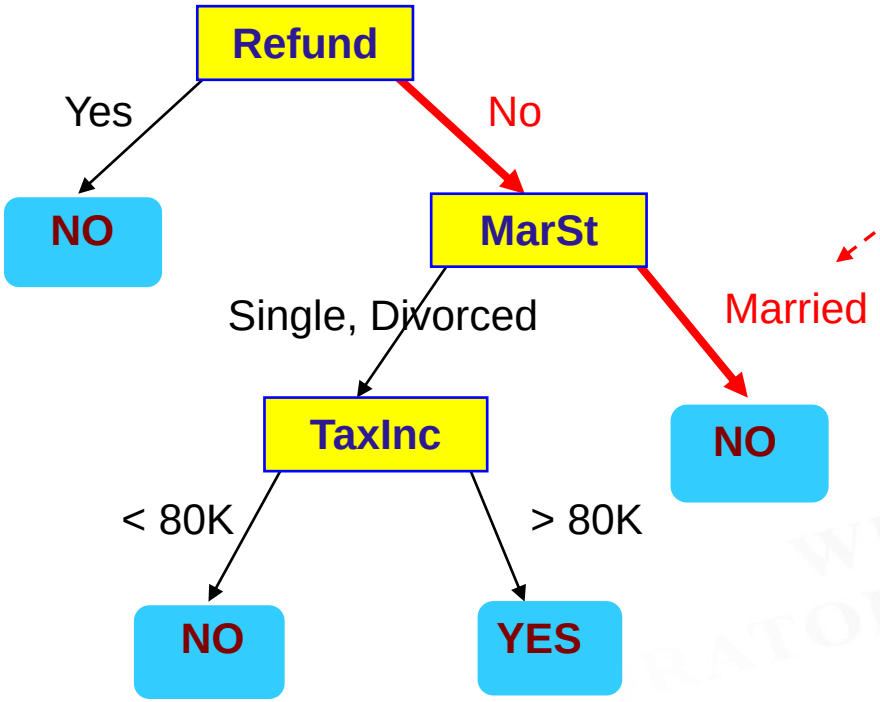
决策树

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



决策树



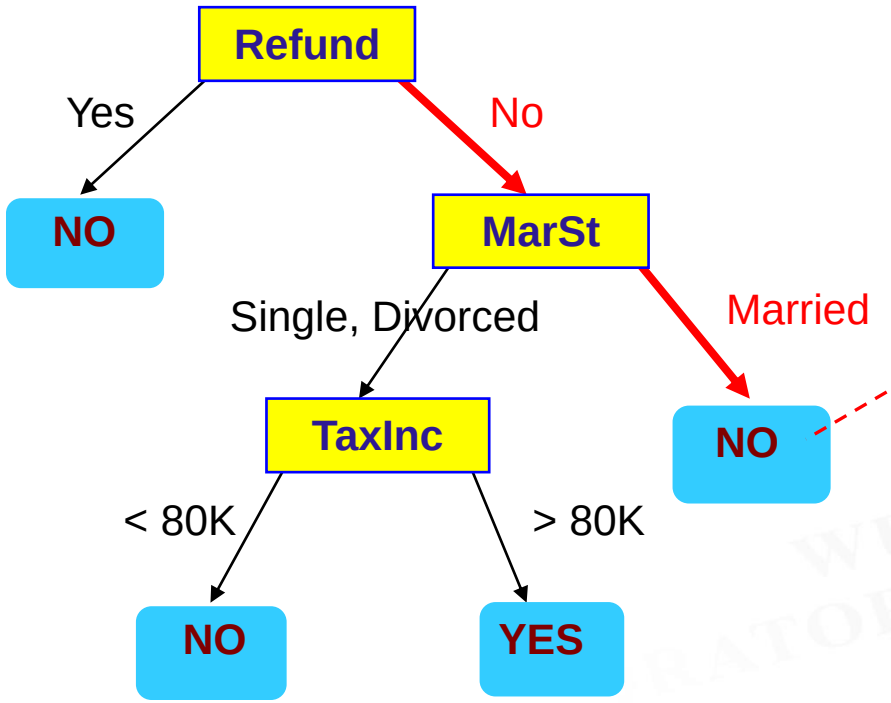
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

决策树

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

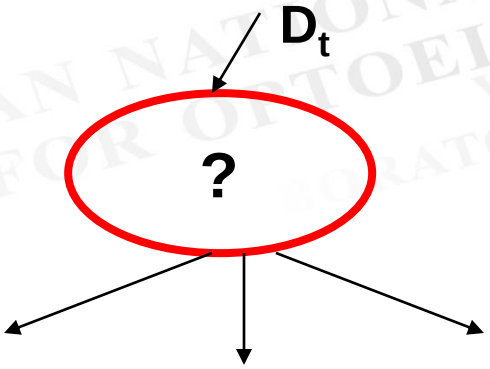


Assign Cheat to “No”

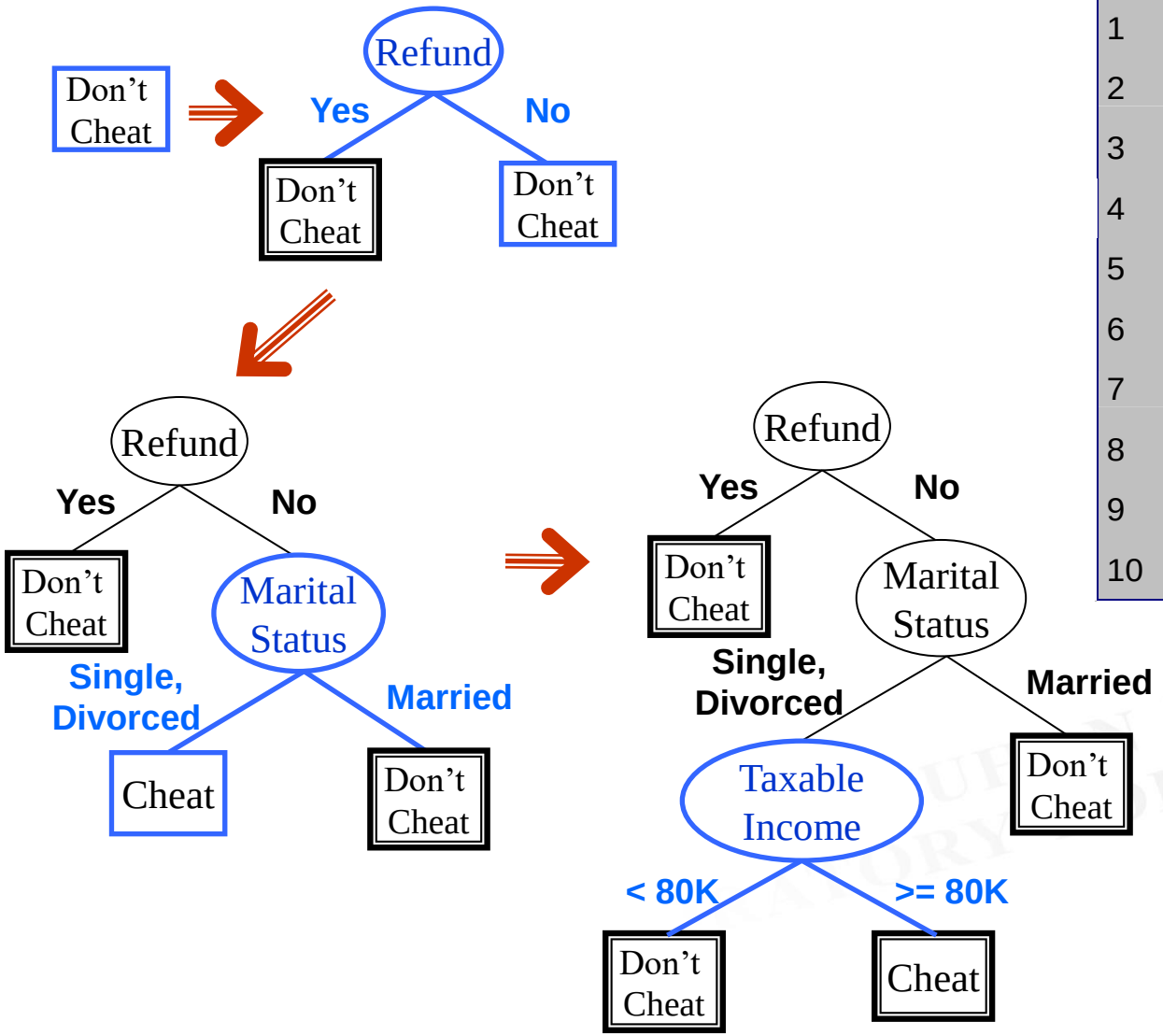
Hunt算法

- Let D_t be the set of training records that reach a node t
- General Procedure:
  - If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
  - If D_t is an empty set, then t is a leaf node labeled by the default class, y_d
  - If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

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

Hunt算法

- 特殊情况处理：
- 1. 若第二步创建的子女结点为空，即不存在与这些结点相关联的记录。这时该结点成为叶结点，类标号为其父结点上训练记录中的多数类
- 2. 第二步中，若所有记录都具有相同的属性值(除目标属性外)，则该结点为叶结点，标号为与该结点相关联的记录中多数类

决策树归纳

- 贪心策略
  - 选择最优化当前结点所包含记录集的属性测试条件分割记录
- 如何分裂训练记录？
  - 如何表示属性测试条件
  - 如何确定最佳分割
- 如何停止分裂过程？

决策树归纳

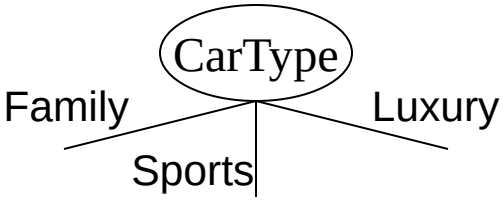
- 贪心策略
  - 选择最优化当前结点所包含记录集的属性测试条件分割记录
- 如何分裂训练记录？
  - 如何表示属性测试条件
  - 如何确定最佳分割
- 如何停止分裂过程？

如何表示测试条件

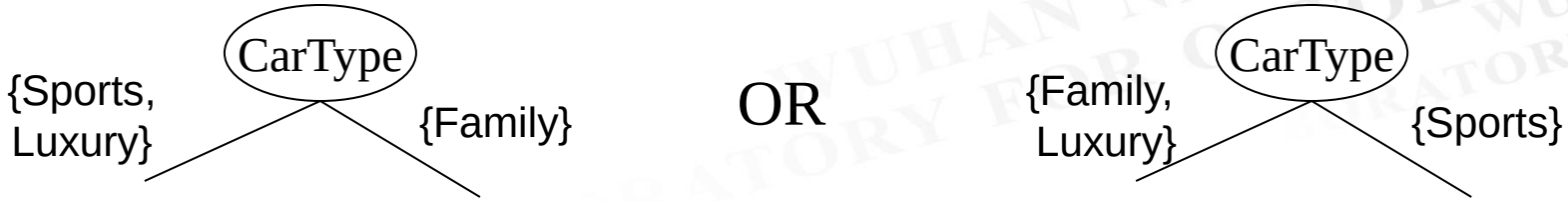
- 根据属性类型的不同：
  - 标称属性
  - 序数属性
  - 连续属性
- 根据分割的数量
  - 二元划分
  - 多元化分

标称属性划分

- 多元划分：输出数取决于该属性不同属性值的个数

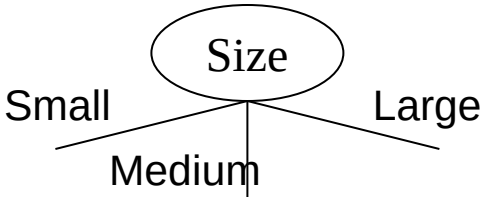


- 二元划分：将属性值分成两个子集（$2^{k-1}-1$），因此需要有一种最优划分的判决方法

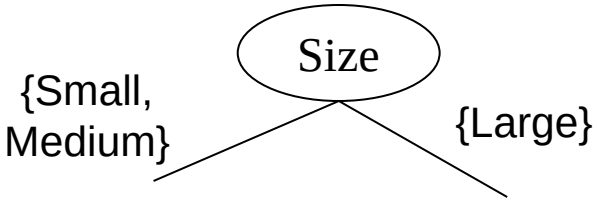


序数属性划分

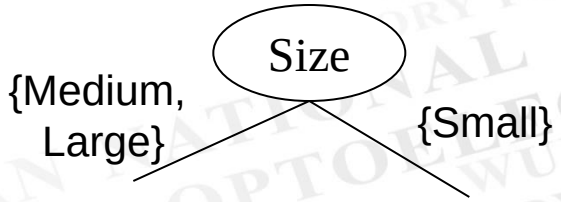
- 多元划分：输出数取决于该属性不同属性值的个数



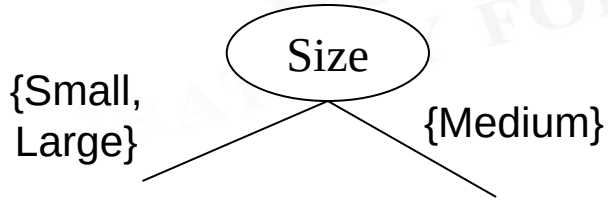
- 二元划分：将属性值分成两个子集（$2^{k-1}-1$），因此需要有一种最优划分的判决方法



OR



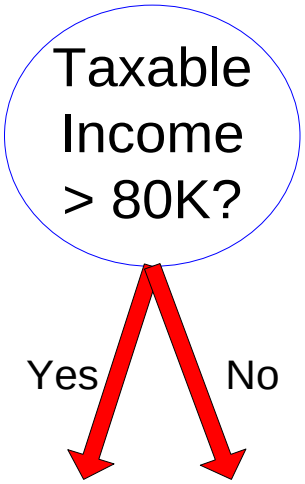
- 这种分割？



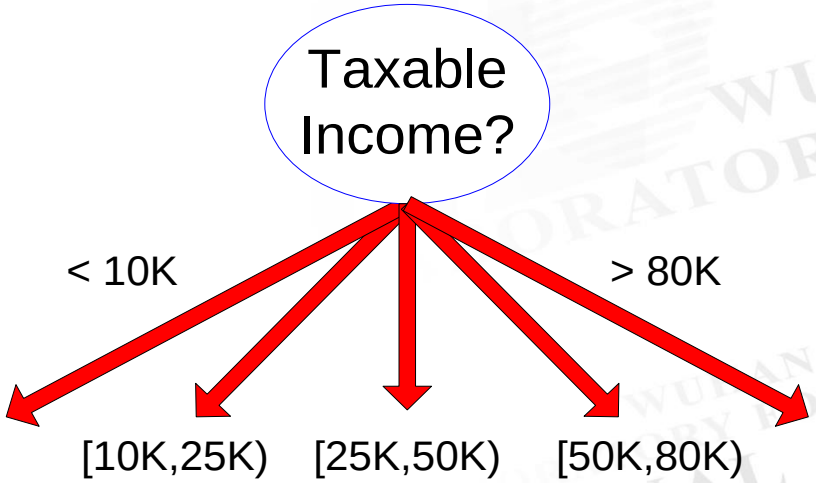
连续属性划分

- 不同的处理方式：
  - 离散化：构造一个有序的类属性
    - 静态划分：算法开始时一次性固定好量化区间
    - 动态划分：等宽法，等频率法，聚类
  - 二进制判决：$(A < v)$ or $(A \geq v)$
    - 需要考虑所有可能的划分，选择最佳
    - 复杂度较高

连续属性划分



(i) Binary split

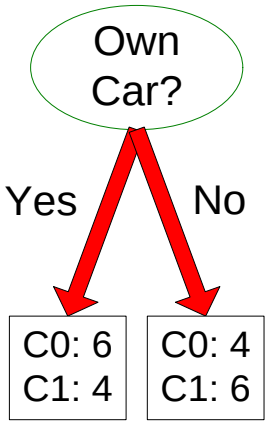


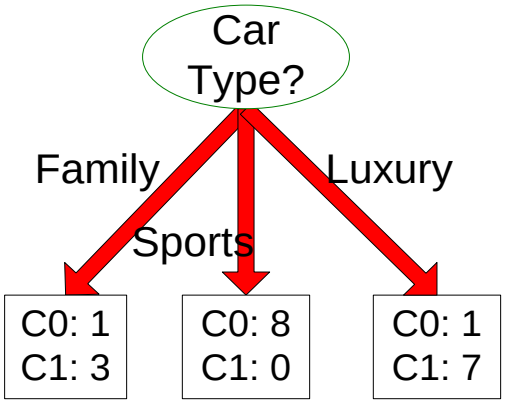
(ii) Multi-way split

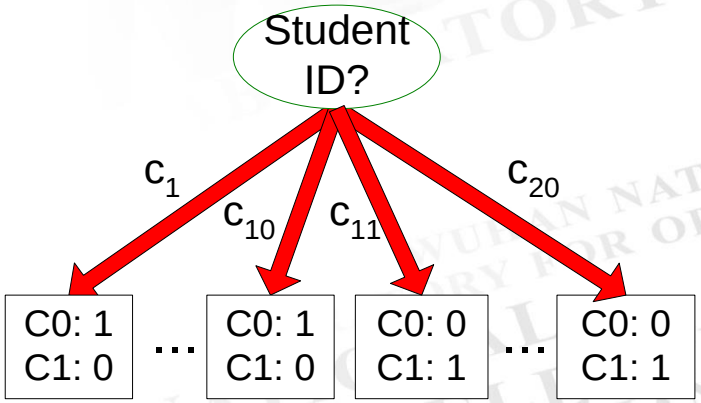
最佳划分的度量

划分前： 类别0的记录10条

类别1的记录10条







哪一种划分最优？

最佳划分的度量

- 贪婪法：
  - 根据子女结点类分布的一致性程度来选择最佳划分

C0: 5 C1: 5	C0: 9 C1: 1
非一致性	一致性
高度不纯	低度不纯

- 如何度量结点的不纯度?

不纯度度量

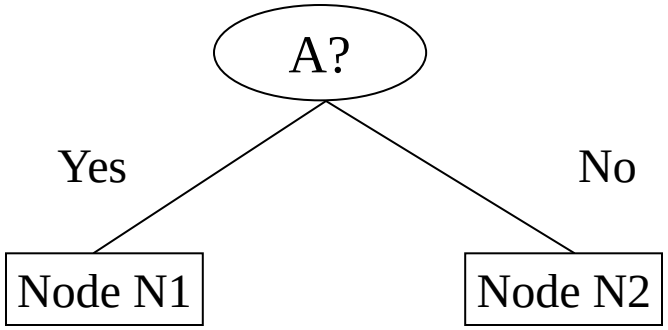
- Gini
- 熵
- 误分类误差

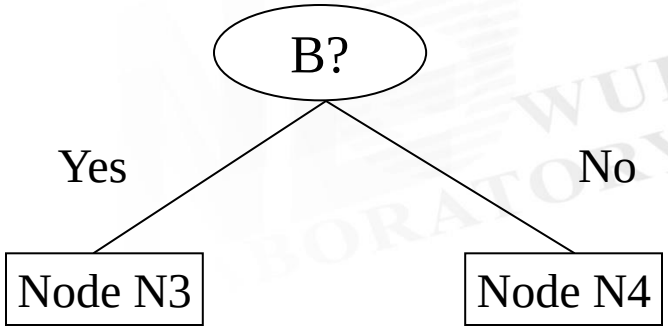
如何寻找最佳划分

Before Splitting:

C0	**N00**
C1	**N01**

→ **M0**





C0	**N10**
C1	**N11**

C0	**N20**
C1	**N21**

C0	**N30**
C1	**N31**

C0	**N40**
C1	**N41**

M1 **M2** **M3** **M4**

M12 **M34**

信息增益 $\Delta = I(parent) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j)$

不纯度度量--Gini

- 对于一个给定的结点t：

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

- $p(j \mid t)$是结点t中类j的相对频率
- 最大值： $(1 - 1/n_c)$，记录在所有类中等分布
- 最小值：0，所有记录属于同一个类

基于Gini的分割

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

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

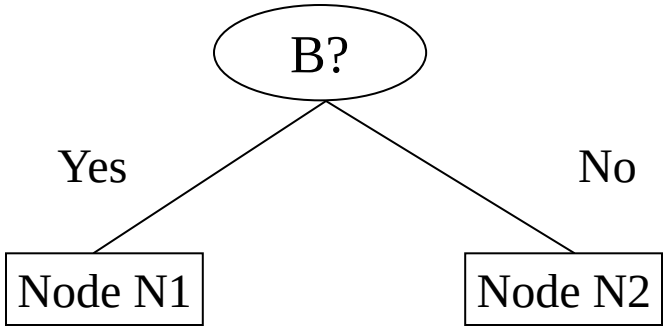
基于Gini的分割

- 应用在CART，SLIQ，SPRINT中
- 当结点p被分割成k个子结点时，分割的质量由如下公式计算

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

- 其中， n_i是结点i包含的记录数
  n是结点p包含的记录数

二元属性：基于Gini的分割



	N1	N2
C1	5	1
C2	2	4
Gini=0.371		

	Parent
C1	6
C2	6
Gini = 0.500	

Gini(N1)
= 1 – (5/7)^2 – (2/7)^2
= 0.408

Gini(N2)
= 1 – (1/5)2 – (4/5)2
= 0.320

Gini(Children)
= 7/12 * 0.408 +
5/12 * 0.320
= 0.371

标称属性：基于Gini的分割

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)

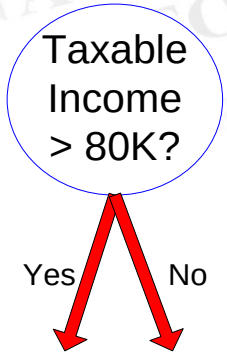
	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

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

连续属性：基于Gini的分割

- 二元划分
- 划分值的选择
  - 可能的划分值数量
  =记录中不同值的个数
- 对于每个分割值，计算两个子集中每个类的个数
  - **A < v and A ≥ v**
- **v的最佳选择**
  - 对于每个可能值，扫描数据集合，计算**Gini**值
  - 计算复杂度高

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：基于Gini的分割

- 更有效的计算方法，对于每个属性
  - 将属性值排序
  - 线性扫描这些属性值，更新统计值，计算Gini
  - 选择Gini值最小的分割值

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
Taxable Income										
Sorted Values →	60	70	75	85	90	95	100	120	125	220
Split Positions →	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
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

➢ 习题2

1. 计算整个训练样本集的Gini指标值
2. 计算属性顾客ID的Gini指标值
3. 计算属性性别的Gini指标值
4. 计算使用多路划分属性车型的Gini指标值
5. 计算使用多路划分属性衬衣尺码的Gini指标值
6. 哪个属性更好？
7. 为什么属性顾客ID的Gini值最低，但却不能作为属性测试条件？

顾客ID	性别	车型	衬衣尺码	类
1	男	家用	小	C0
2	男	运动	中	C0
3	男	运动	中	C0
4	男	运动	大	C0
5	男	运动	加大	C0
6	男	运动	加大	C0
7	女	运动	小	C0
8	女	运动	小	C0
9	女	运动	中	C0
10	女	豪华	大	C0
11	男	家用	大	C1
12	男	家用	加大	C1
13	男	家用	中	C1
14	男	豪华	加大	C1
15	女	豪华	小	C1
16	女	豪华	小	C1
17	女	豪华	中	C1
18	女	豪华	中	C1
19	女	豪华	中	C1
20	女	豪华	大	C1

基于熵的分割

- 结点t的熵定义如下：

$$Entropy(t) = -\sum_j p(j|t)\log p(j|t)$$

  - $p(j|t)$是结点t中类j的相对频率
  - 度量节点的一致性程度
  - 最大值$\log n_c$，当记录在所有类之间等分布时
  - 最小值0，当所有记录属于同一类时
- 和基于Gini的计算较类似

基于熵的分割

$$Entropy(t) = -\sum_j p(j|t)\log_2 p(j|t)$$

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

P(C1) = 0/6 = 0　　P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

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

P(C1) = 1/6　　P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (5/6) = 0.65

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

P(C1) = 2/6　　P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

基于分类误差的分割

- 结点t的分类误差定义为：

$$Error(t) = 1 - \max_i P(i \mid t)$$

- 度量一个结点产生的误分类误差
  - 最大值$(1 - 1/n_c)$，当记录在所有类中等分布时
  - 最小值0，当所有记录属于同一类时

基于分类误差的分割

$$Error(t) = 1 - \max_i P(i \mid t)$$

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

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

Error = 1 – max (0, 1) = 1 – 1 = 0

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

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

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

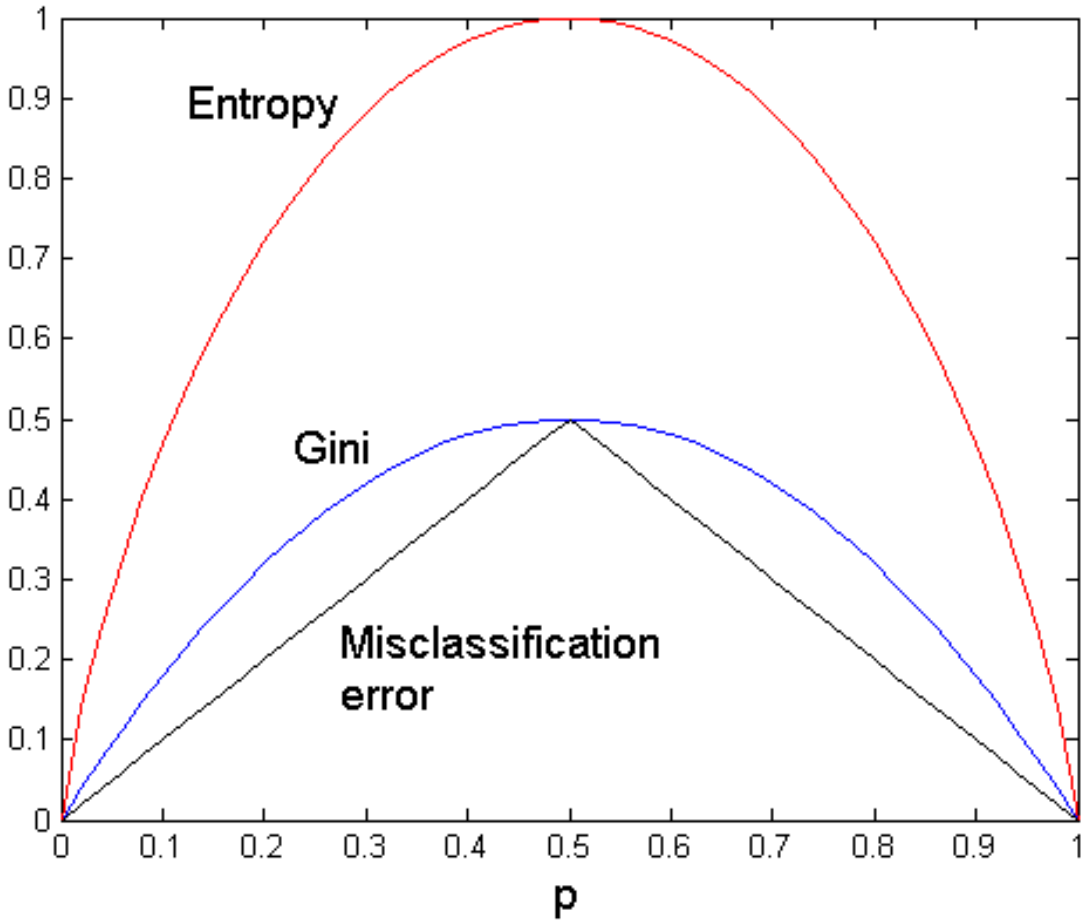
C1	**2**
C2	**4**

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

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

分割准则的比较

- 对于一个两类的问题



增益比

- 增益比的定义：

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

父结点p被分割成k个子集， n_i是子集i的记录数

- 根据分割后的熵(SplitINFO)调整信息增益，避免出现很多的小分割子集
- 用在 C4.5中
- 用来克服利用信息增益分割的不足

决策树归纳

- 习题3
- 1. 整个训练样本集关于类属性的熵是多少
- 2. 关于这些样本，a1和a2的信息增益是多少？
- 3. 对于连续属性a3，计算所有可能的划分的信息增益
- 4. 根据信息增益，哪个是最佳划分（a1，a2和a3中）？
- 5. 根据分类差错率，哪个是最佳划分（a1和a2中）？
- 6. 根据Gini指标，哪个是最佳划分（a1和a2中）？

实例	a1	a2	a3	目标类
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	-
4	F	F	4.0	+
5	F	T	7.0	-
6	F	T	3.0	-
7	F	F	8.0	-
8	T	F	7.0	+
9	F	T	5.0	-

决策树归纳

- 贪心策略
  - 选择最优化当前结点所包含记录集的属性测试条件分割记录
- 如何分裂训练记录？
  - 如何表示属性测试条件
  - 如何确定最佳分割
- 如何停止分裂过程？

决策树归纳的停止条件

- 当所有记录属于同一类时停止
- 当所有记录具有相似的属性值时停止
- 提前终止

决策树归纳算法

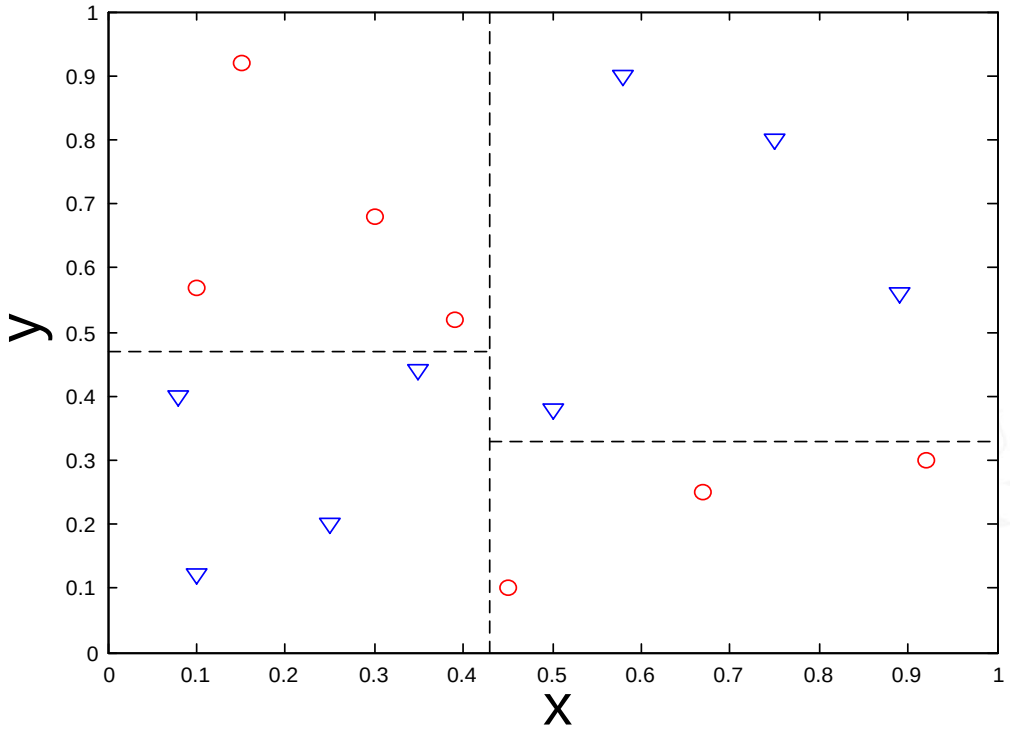
```
TreeGrowth (E, F)
1: if stopping_cond (E, F) = true then
2:     leaf = createNode ()
3:     leaf.label = classify (E)
4:     return leaf
5: else
6:     root = createNode ()
7:     root.test_cond = find_best_split (E, F)
8:     令 V = {v|v 是 root.test_cond 的一个可能的输出}
9:     for 每个 v∈V do
10:        E_v = {e|root.test_cond(e)=v 并且 e∈E}
11:        child = TreeGrowth (E_v, F)
12:        将 child 作为 root 的派生结点添加到树中，并将边(root→child)标记为 v
13:    end for
14: end if
15: return root
```

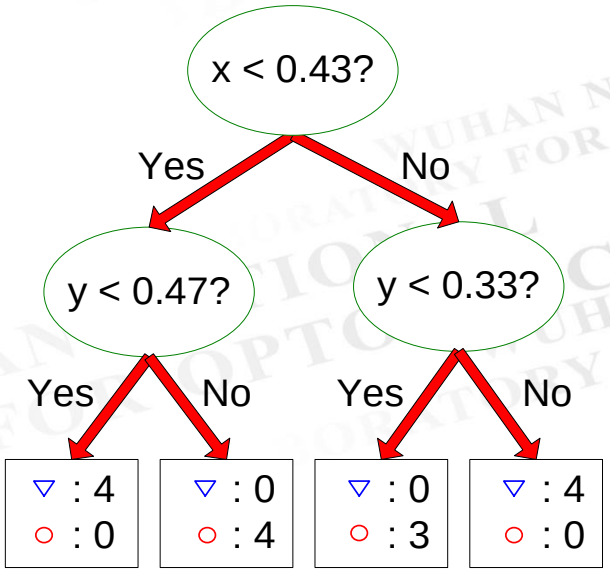
决策树归纳的特点

- 1. 是一种构建分类模型的非参数方法；
- 2. 可快速建立决策树模型，对于未知样本可快速分类
- 3. 容易解释，尤其是较小的决策树
- 4. 对于噪声和冗余属性有较好的鲁棒性
- 5. 支持多分类问题

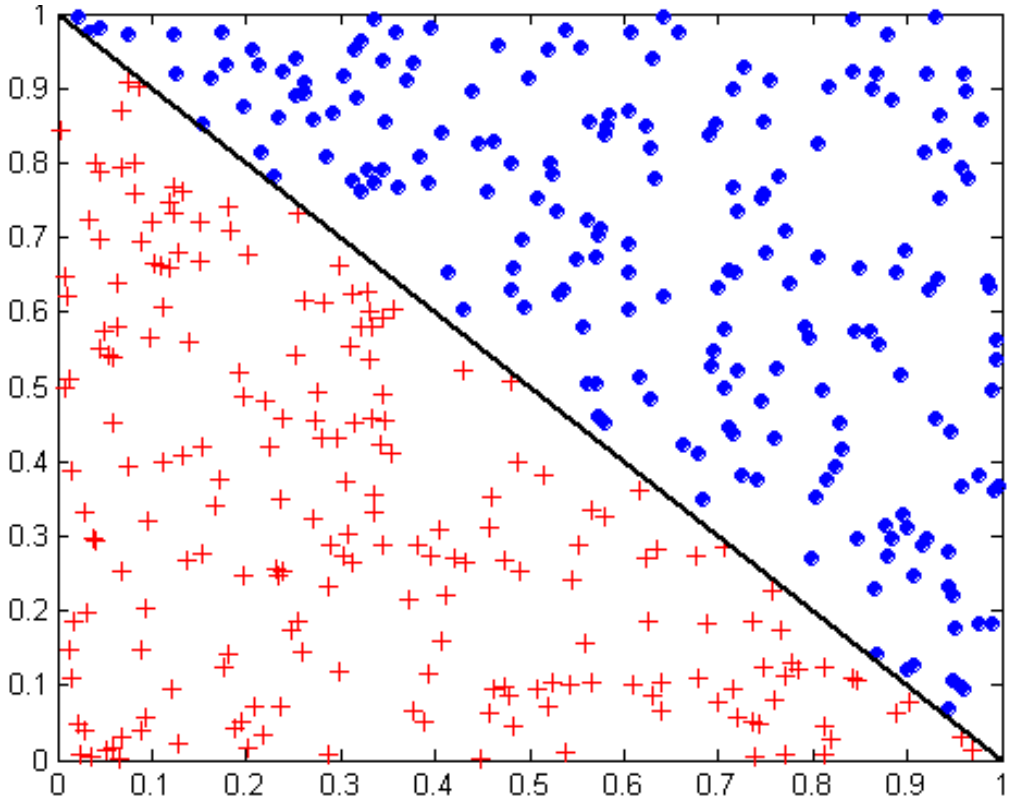
决策树归纳的特点

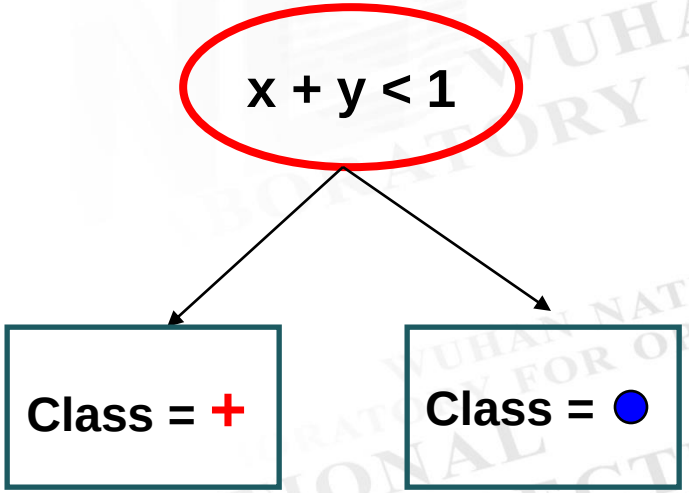
- 6. 不纯性度量方法的选择对决策树算法的性能影响很小
- 7. 单属性测试条件建模能力有限





斜决策树





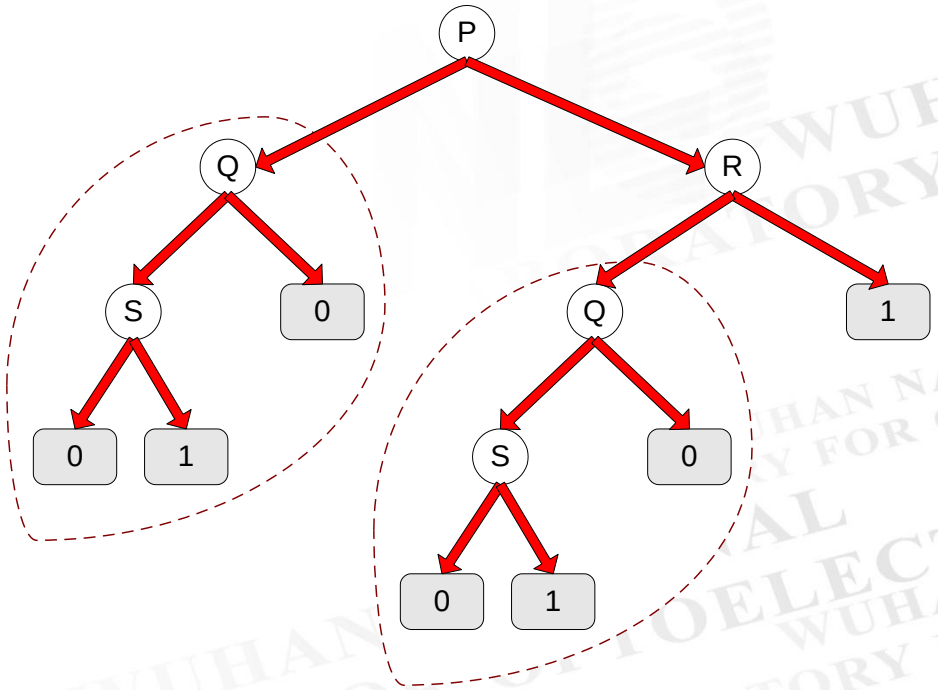
- 测试条件可以包含多个属性
- 更强的表达能力
- 找出最佳测试条件的计算复杂度很高

构造归纳

- 在原有属性的基础上构造新的属性，增加到属性集合中
  - 不需要昂贵的计算开销
  - 会产生冗余属性

决策树归纳的特点

- 8. 重复子树
- 9. 数据碎片



例子：ID3算法

Function ID3(R:一个非类别属性集合，C:类别属性，S:一个训练集) 返回一个决策树：

Begin

If S 为空，返回一个值为 Failure 的单个节点；

If S 是由其值均为相同类别属性值的记录组成，

返回一个带有该值的单个节点；

If R 为空，则返回一个单节点，其值为在 S 的记录中找出的频率最高的类别属性值；

Else 将 R 中属性之间具有最大 gain(D,S)的属性赋给 D；

将属性 D 的值赋值给$\{d_j \mid j=1,2,\ldots m\}$；

将分别由对应于 D 的值为 d_j 的记录组成的 S 的子集赋值给$\{s_j \mid j=1,2,\ldots m\}$；

返回一棵树，其根标记为 D，树枝标记为 $d_1,d_2,\ldots d_m$;

再分别构造以下树：ID3(R-{D},C,S_1)，ID3(R-{D},C,S_2)，…，ID3(R-{D},C,S_m)；

End ID3；

例子：C4.5算法

- 在ID3算法基础上，增加如下功能：
- 用增益比例的概念；
- 可以处理缺少属性值的训练样本；
- 可以处理连续数值型属性；
- 通过使用修剪技术以避免树的不平衡；
- K折迭代交叉验证

例子：C4.5算法

- 合并具有连续值的属性
- 1. 根据属性的值对数据集排序
- 2. 用不同的阈值将数据集动态的进行划分（阈值选择在输出改变处）
- 3. 取两个属性值的中点作为一个阈值，将所有样本划分为两个子集
- 4. 计算所有可能的阈值、增益及增益比，选择最佳划分阈值

- 习题6—考虑如下训练样本集

X	Y	Z	No. of Class C1 Examples	No. of Class C2 Examples
0	0	0	5	40
0	0	1	0	15
0	1	0	10	5
0	1	1	45	0
1	0	0	10	5
1	0	1	25	0
1	1	0	5	20
1	1	1	0	15

- (a)用贪心算法计算两层的决策树，使用分类差错率作为划分标准，决策树的总差错率为多少？

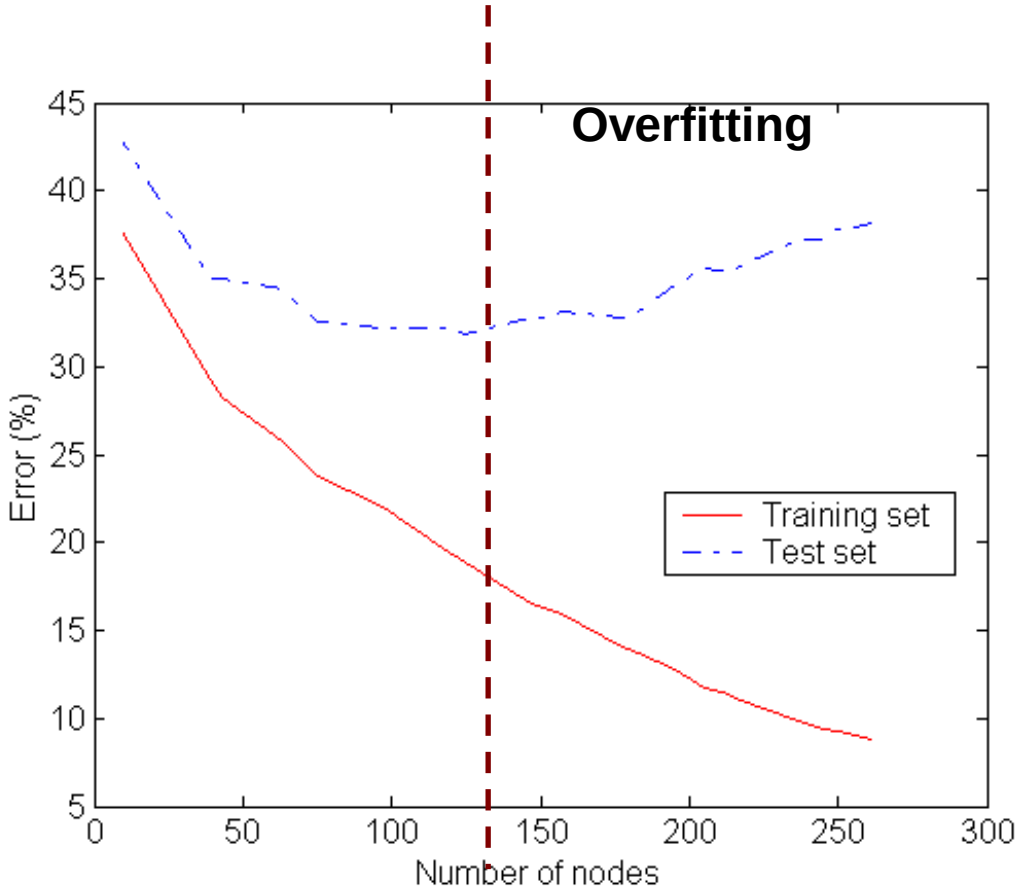
- **(b)使用X作为第一个划分属性，两个后继结点分别在剩余的属性中选择最佳的划分属性，重复步骤(a)，所构造决策树的差错率是多少？**
- **(c)比较(a)和(b)的结果，评述在划分属性选择上启发式贪心法的作用**

分类模型的几个问题

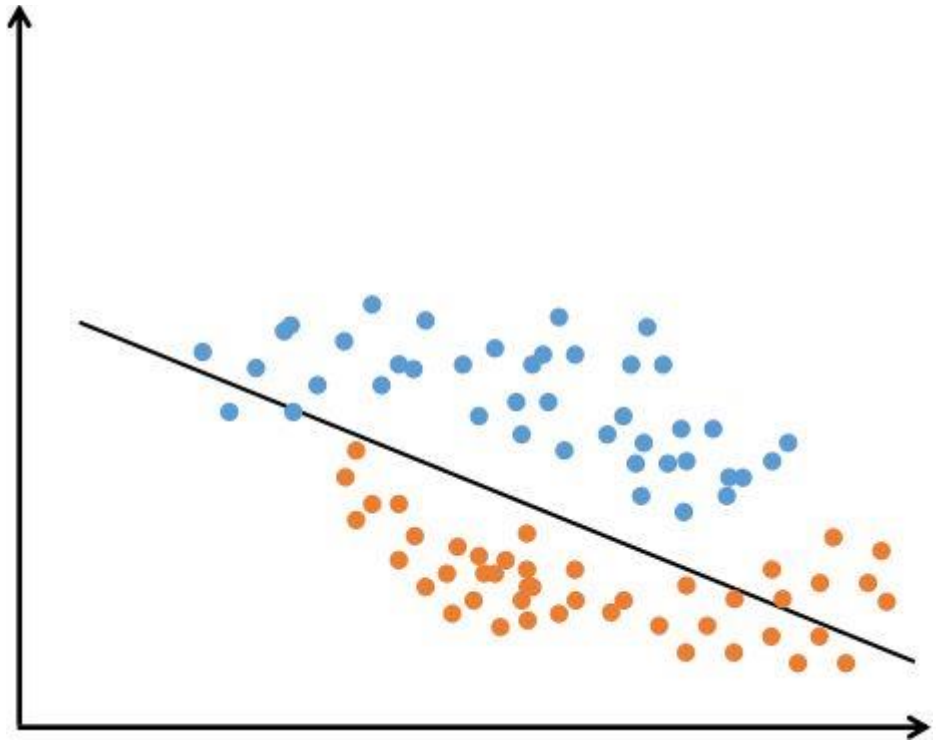
- 拟合不足和过分拟合
  - 训练误差
  - 泛化误差

拟合不足和过分拟合

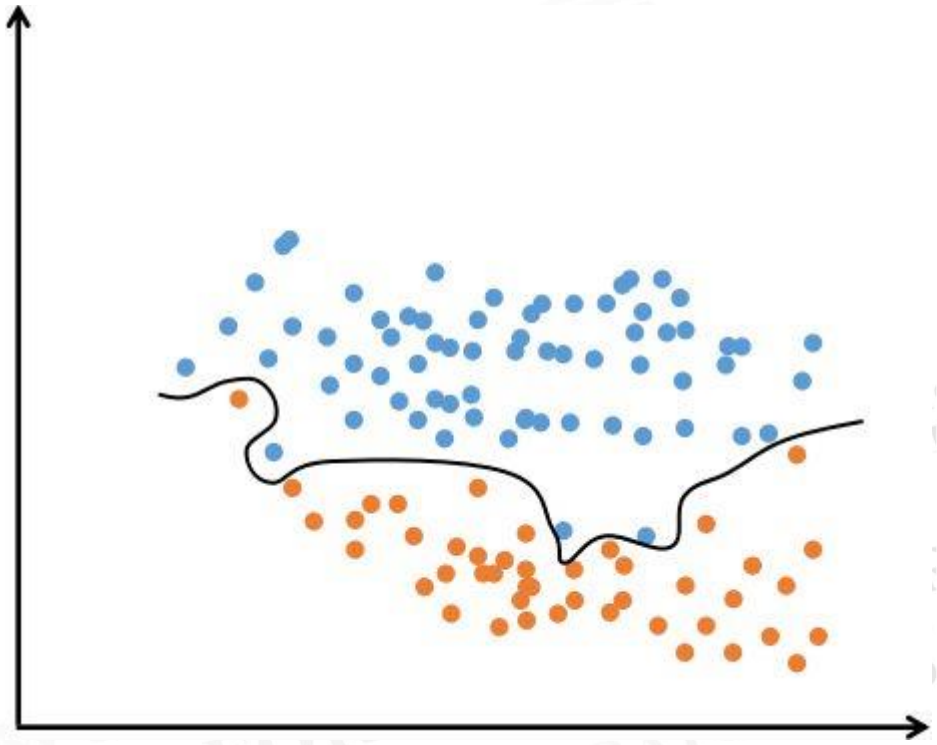
- 拟合不足
  - 训练和泛化误差都很大
- 过分拟合
  - 训练误差小
  - 泛化误差大



拟合不足和过分拟合

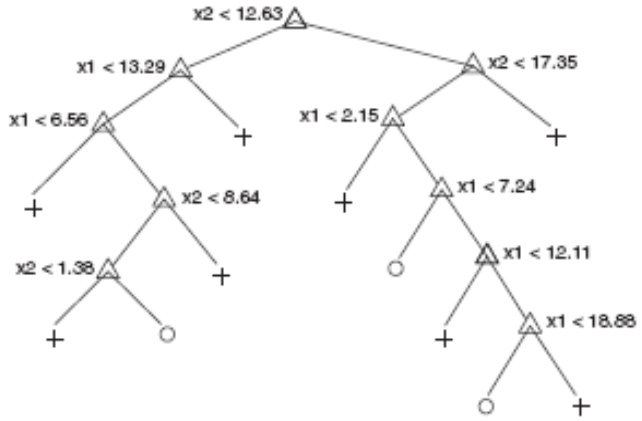


欠拟合

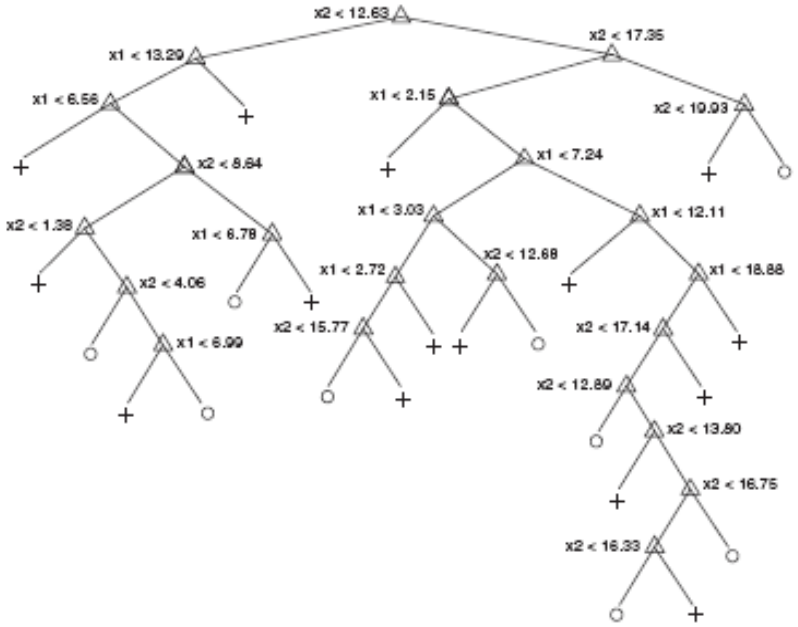


过拟合

拟合不足和过分拟合



(a) Decision tree with 11 leaf nodes.



(b) Decision tree with 24 leaf nodes.

拟合不足和过分拟合

- 模型的泛化误差由两部分组成
  - 偏差 & 方差
- 偏差（Bias）是模型本身导致的误差，是模型的预测值的数学期望和真实值之间的差距

$$\mathrm{Bias}[\tilde{h}(x)] = E[\tilde{h}(x)] - h(x)$$

- 高偏差意味着模型本身的输出值与期望值差距很大
  - 欠拟合

拟合不足和过分拟合

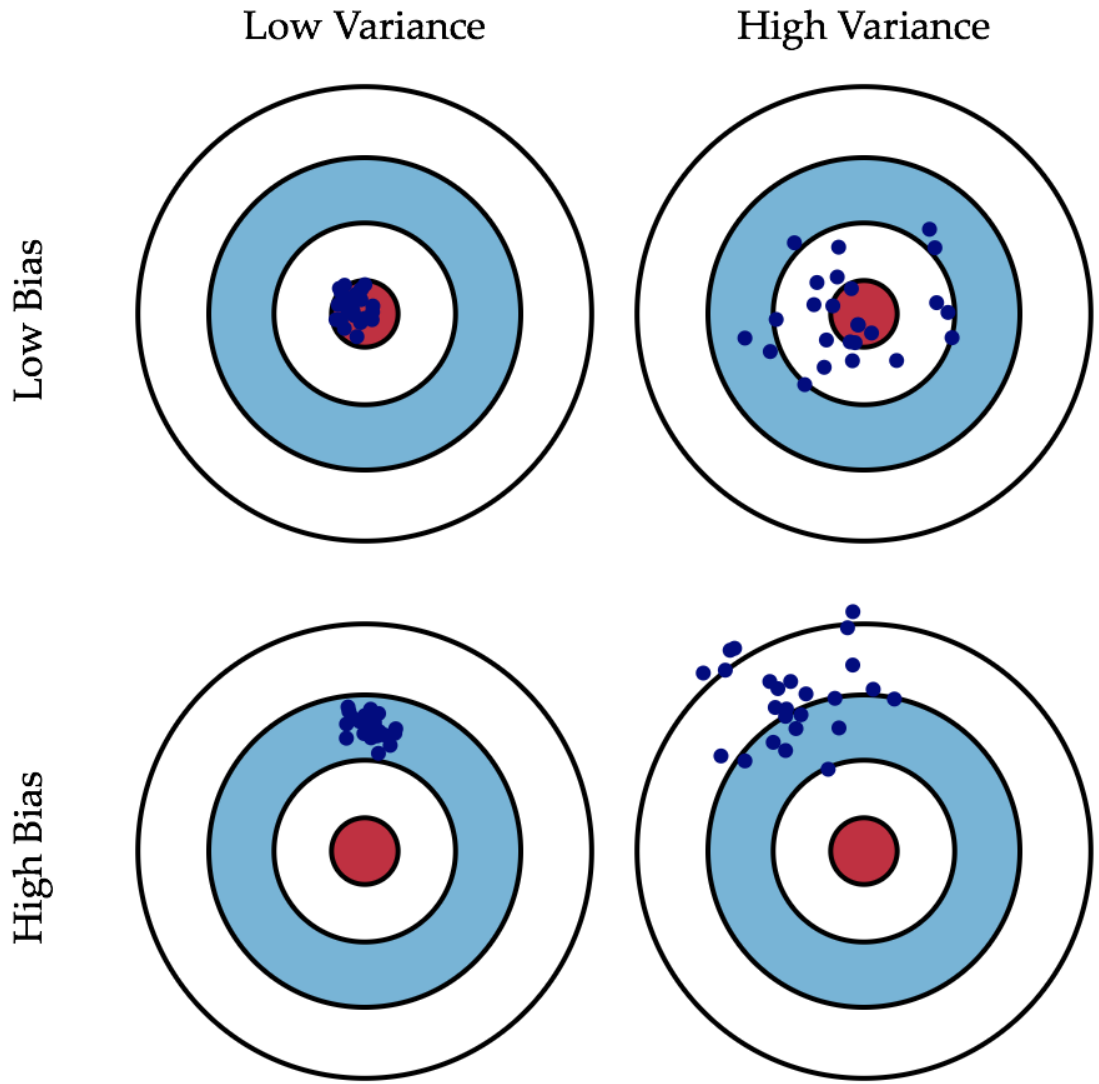
- 方差（**variance**）表示模型预测值的波动程度

$$\mathrm{Var}[\tilde{h}(x)] = \mathrm{E}[(\tilde{h}(x) - \mathrm{E}[\tilde{h}(x)])^2]$$

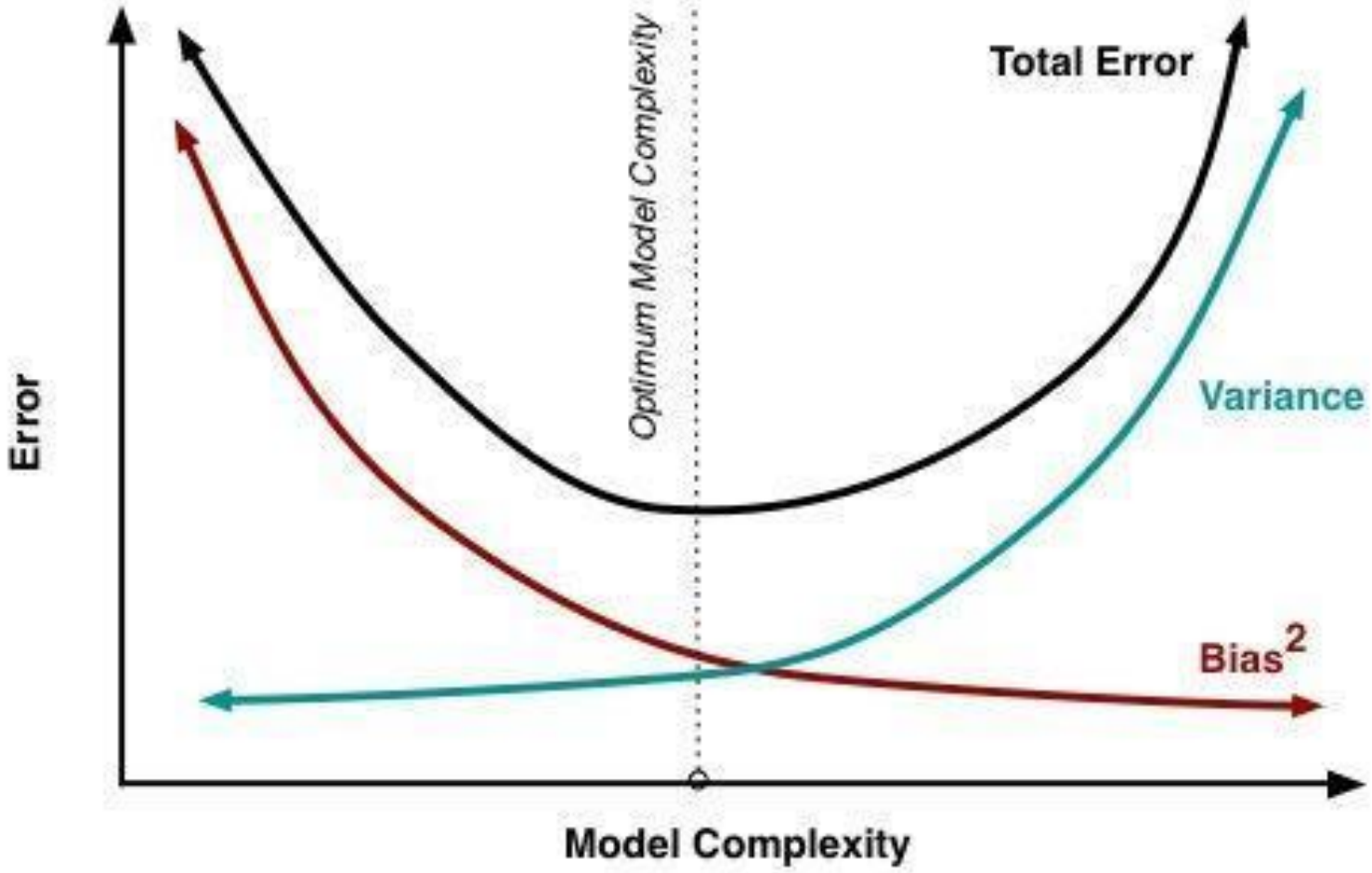
- 高方差意味着模型对训练集中的随机扰动进行建模
  - 过拟合

$$\mathrm{err} = \mathrm{Bias}^2 + \mathrm{Variance} + \mathrm{IrreduciableError}$$

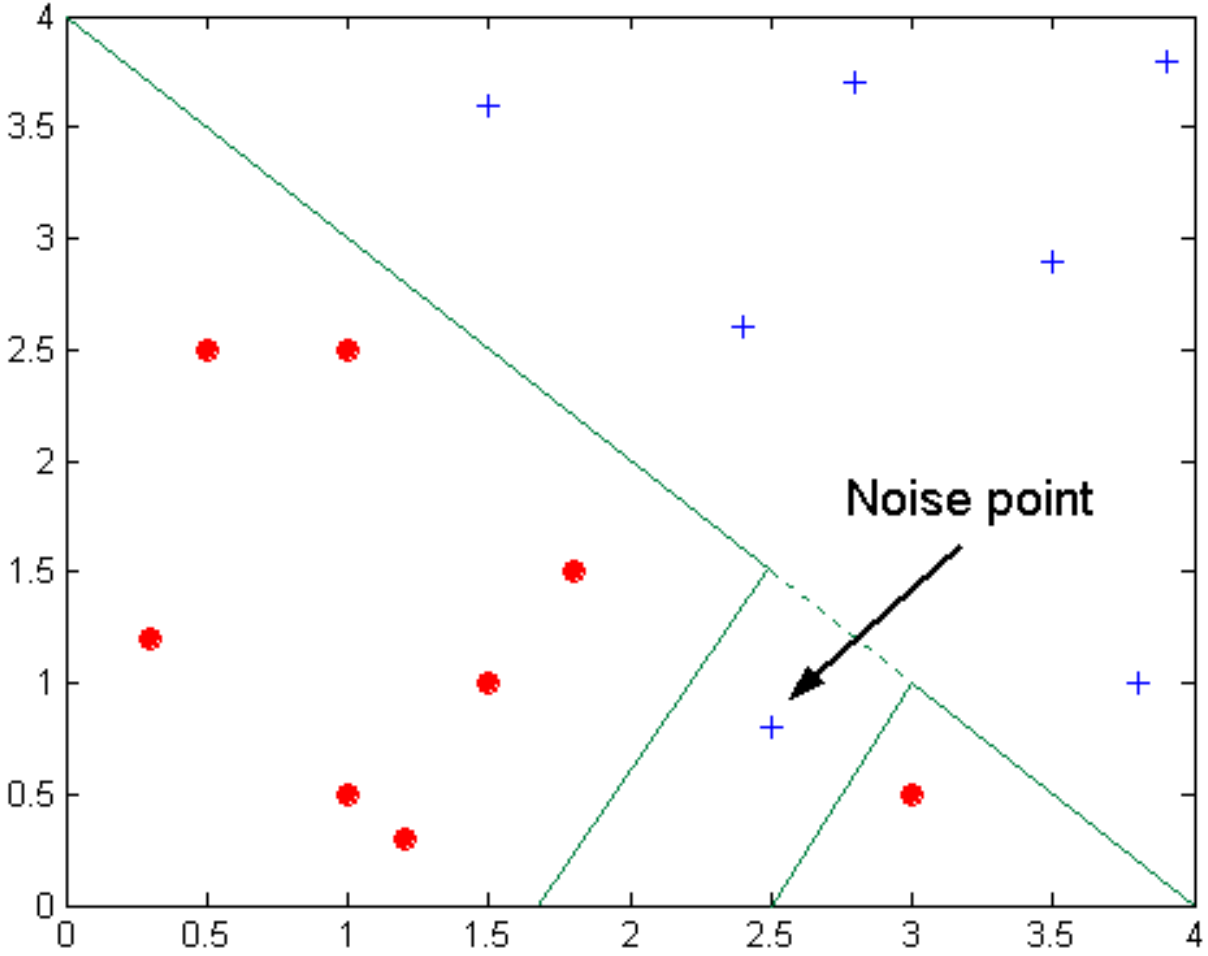
拟合不足和过分拟合



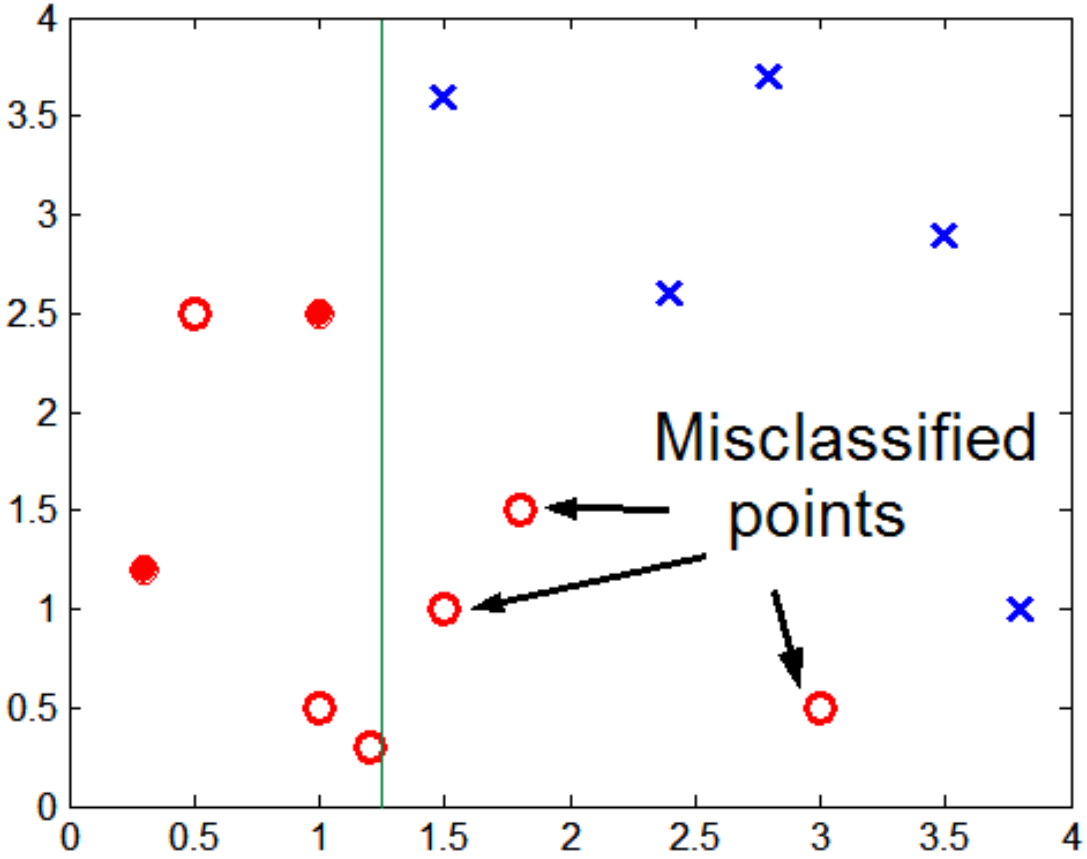
拟合不足和过分拟合


Error
Optimum Model Complexity
Total Error
Variance
Bias^2
Model Complexity

噪声导致的过分拟合



样本不足导致的过分拟合



如何处理过分拟合

- 先剪枝（提前终止规则）
  - 在完全拟合之前就停止树的生长
  - 典型的结点停止条件：
    - 如果所有的记录属于同一类，停止
    - 如果所有的记录有相同的属性，停止
  - 更具限制性的条件
    - 如果记录数小于预设的门限时，停止
    - 如果不纯性度量的增益低于预设的门限时，停止

如何处理过分拟合

- 后剪枝
  - 决策树按照最大规模生长
  - 由底往上修剪决策树
- 用新的叶结点替换子树
  - 叶结点的类标号由多数类指定
- 用子树中最常使用的分支代替子树

拟合不足和过分拟合

- 欠拟合：偏差过大，做特征工程、减小(弱)正则化系数

- 过拟合：方差过大，可增加样本、减少特征、增加(强)正则化系数

如何处理过分拟合

- 正则化：抑制模型向过度复杂方向进化
- 数据增强：训练集越多，过拟合概率越小
  - 图像旋转，缩放，剪切，添加噪声等
  - 同义词替换扩充数据集
  - 不同年龄、性别的语音，方言等
- 集成学习：通过平均多个模型的结果，来降低模型的方差

如何处理过分拟合

- 正则化

$$L(\theta)=\frac{1}{2l}\sum_{i=1}^{l}\left(h_\theta(\mathrm{x}_i)-y_i\right)^2$$

- 为损失函数加上一个惩罚项对复杂的模型进行惩罚，即强制让模型的参数值尽可能小

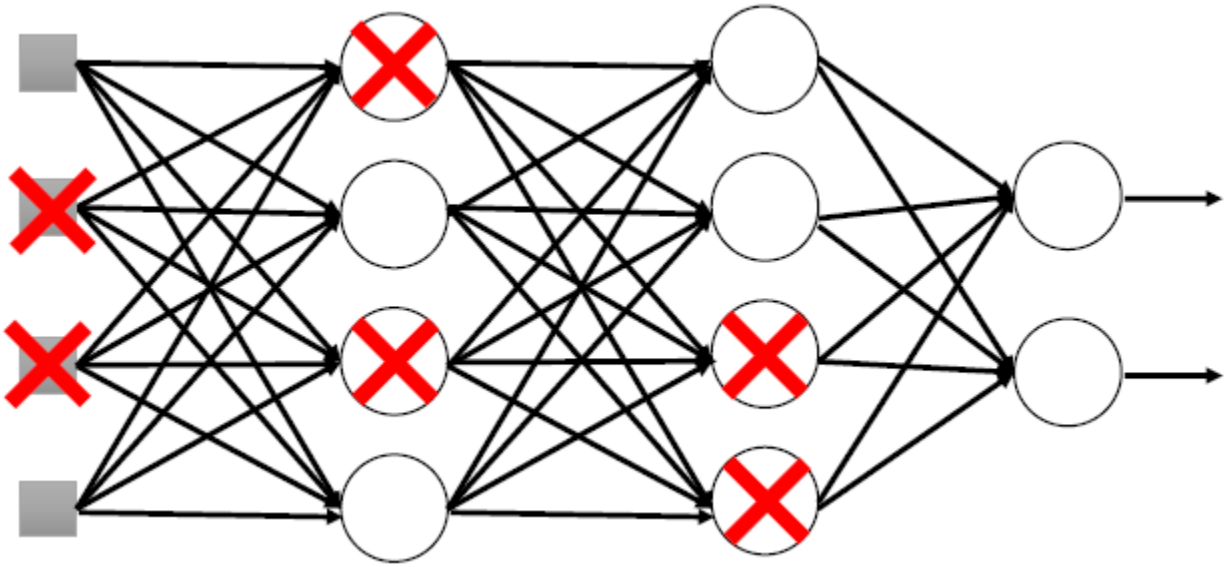
$$L(\theta)=\frac{1}{2l}\sum_{i=1}^{l}\left(h_\theta(\mathrm{x}^i)-y^i\right)^2+\frac{\lambda}{2}r(\theta)$$

- 第二项称为正则化项，目的是让它的值尽可能小，即参数等于或者接近于0
- 正则化项通常使用L2范数或L1范数

如何处理过分拟合

- **Dropout**

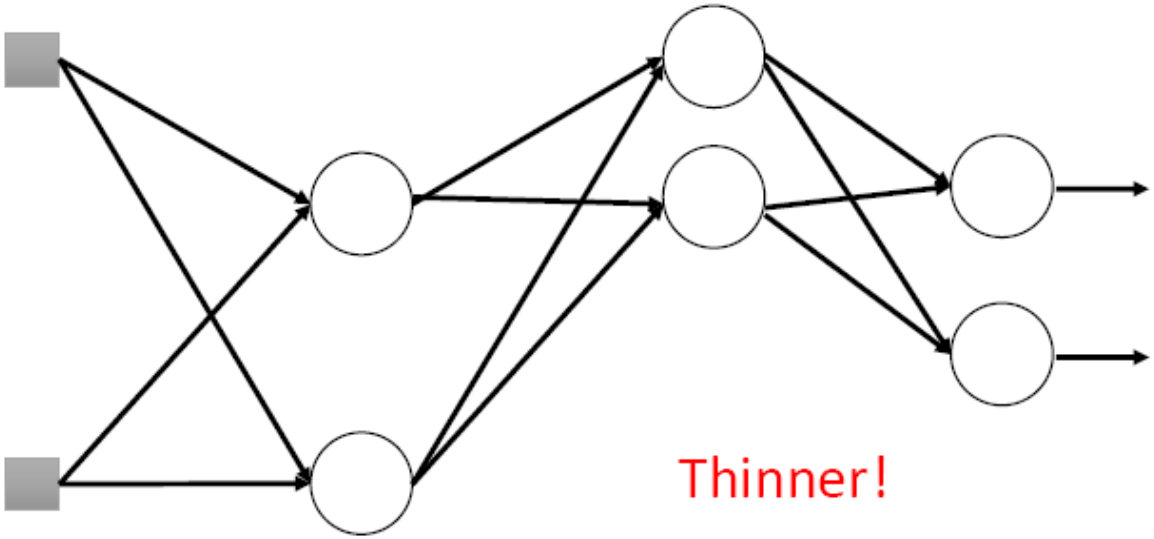
  Training:



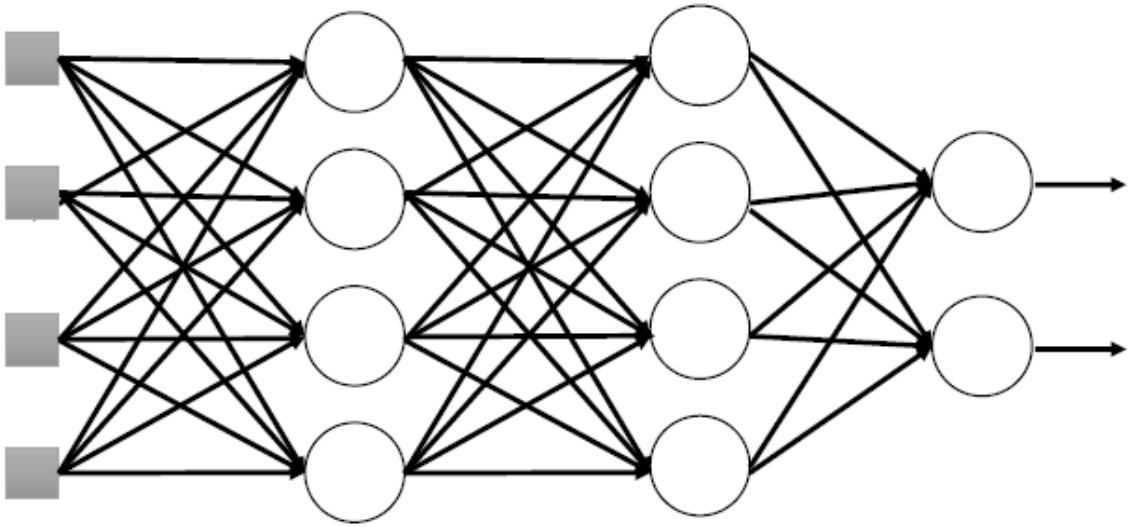
- 每次更新参数前，每个神经元有p%的概率失效

如何处理过分拟合

Training:

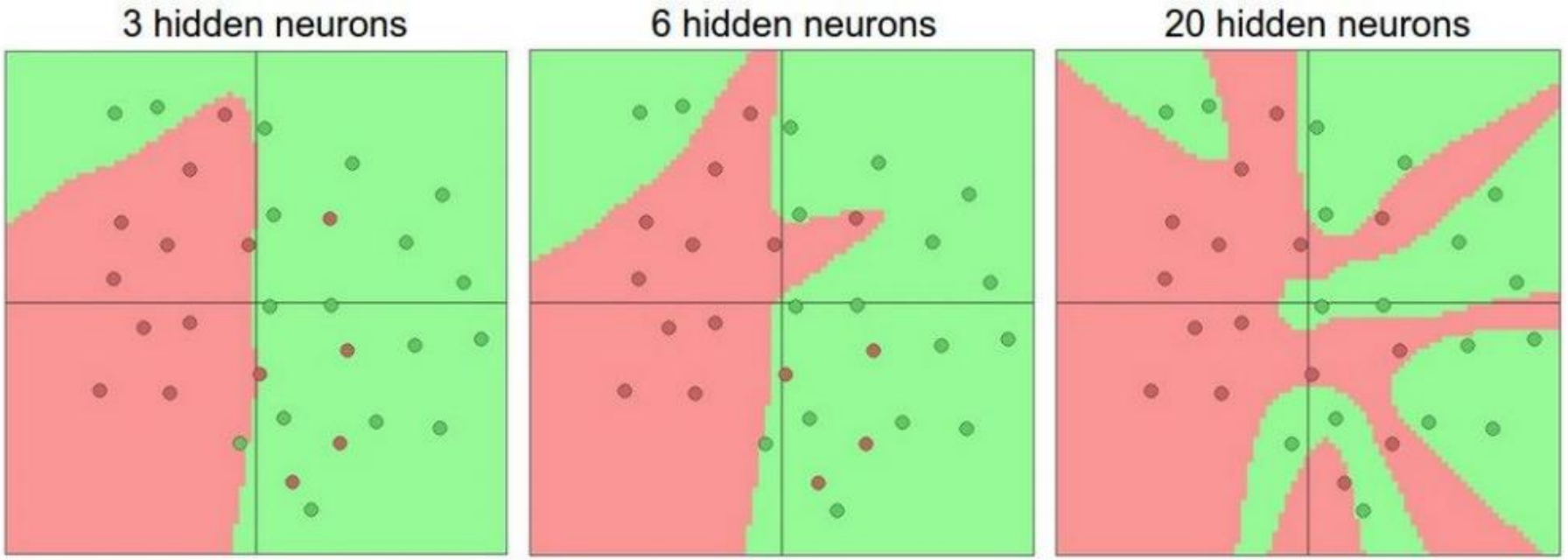


Testing:



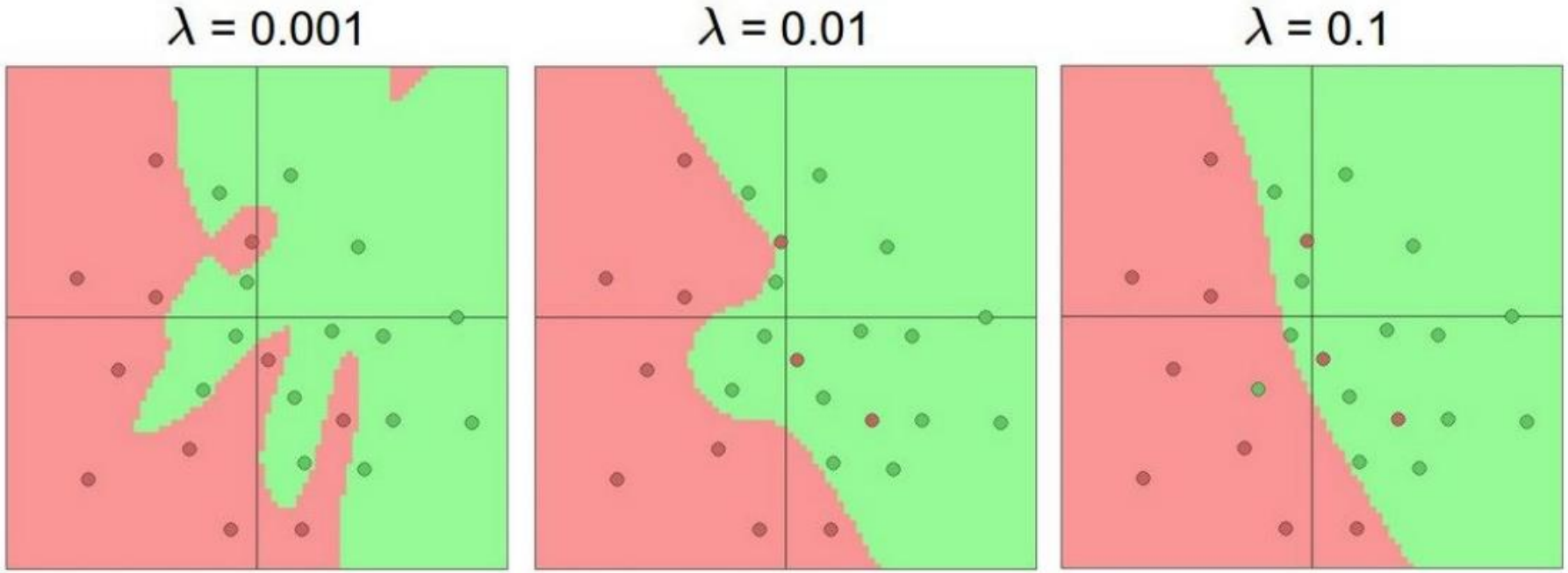
正则化

- 隐藏层不同的神经元个数



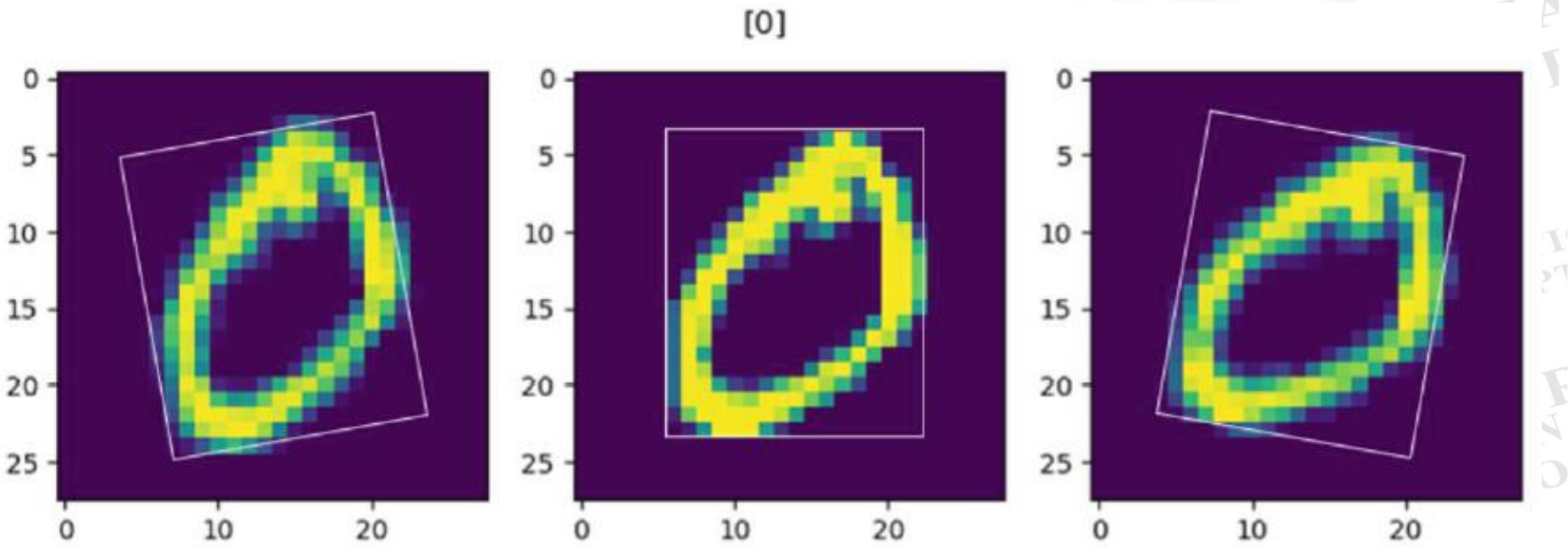
正则化

- 不同的正则化系数



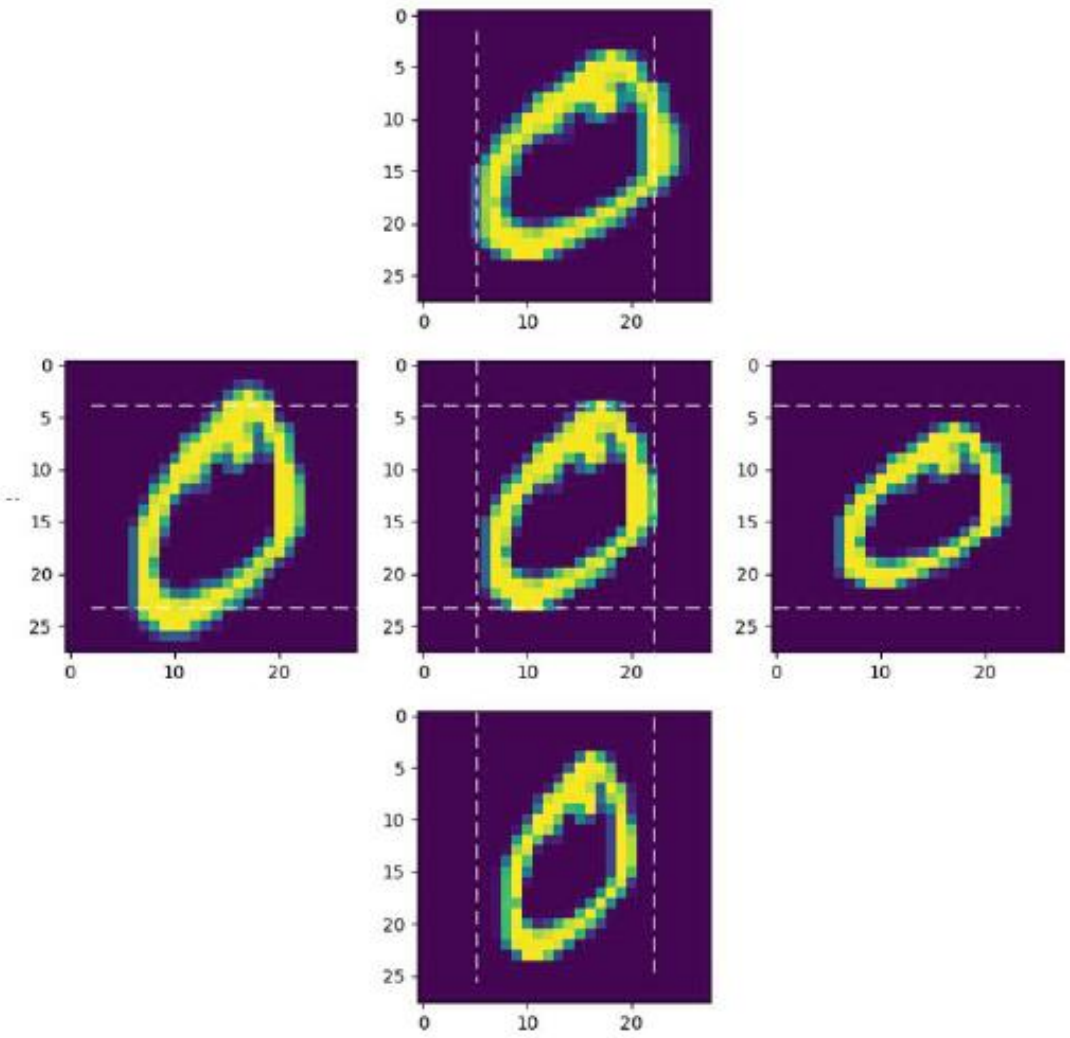
数据增强

- 旋转



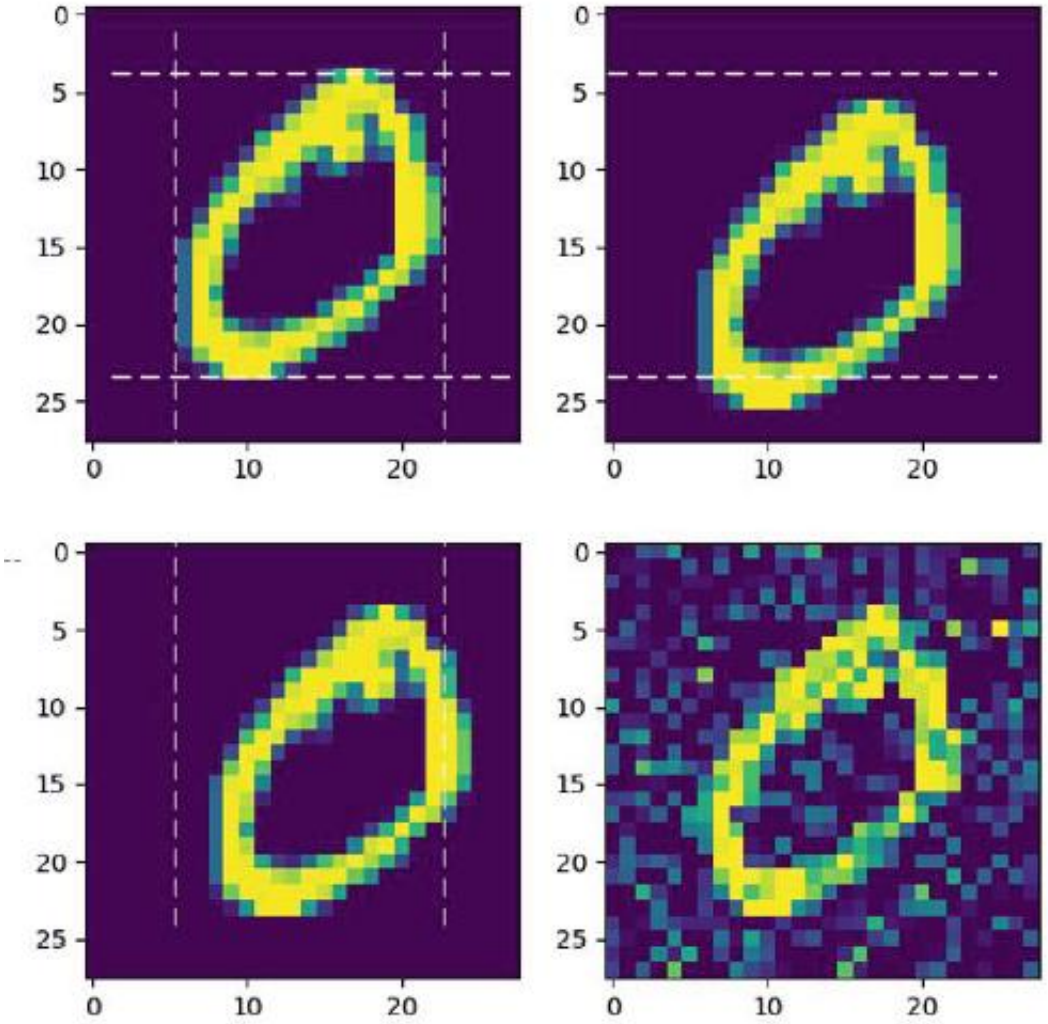
数据增强

- 缩放



数据增强

- 平移和噪声

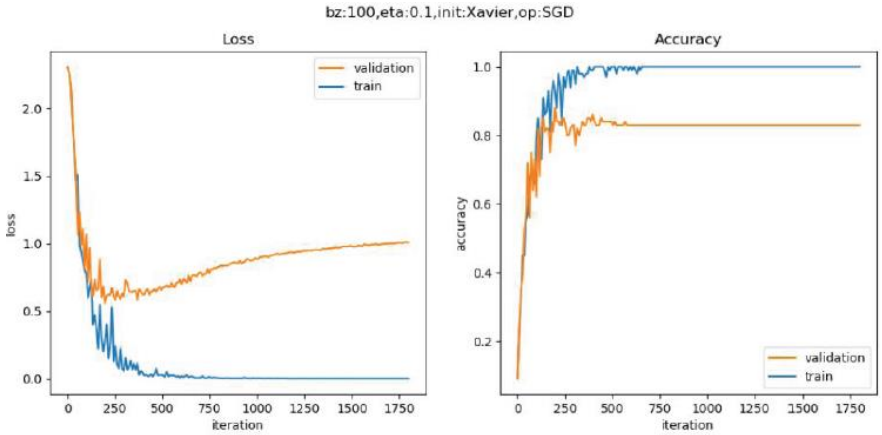


数据增强

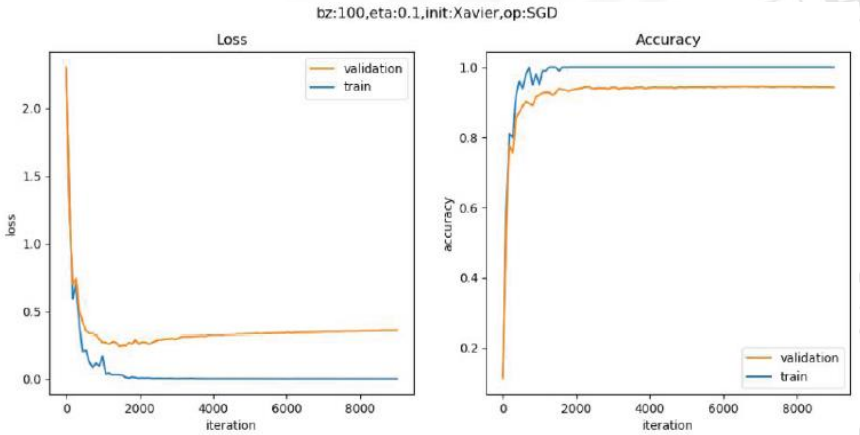
➢其他图像处理方法

- ➢ 翻转图像：即左右镜像，或上下镜像，但不适合数字识别
- ➢ 裁剪图像：从图像中随机剪切一部分，再调整为原图像大小
- ➢ 颜色扰动：通过在颜色空间中添加噪声使图像颜色产生抖动
- ➢ 对比度变化
- ➢ 亮度变化
- ➢ 颜色增强

训练效果对比



数据增强前



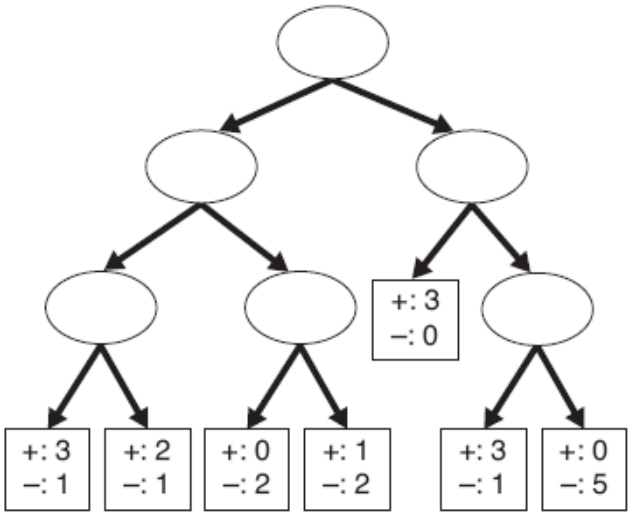
数据增强后

泛化误差估计

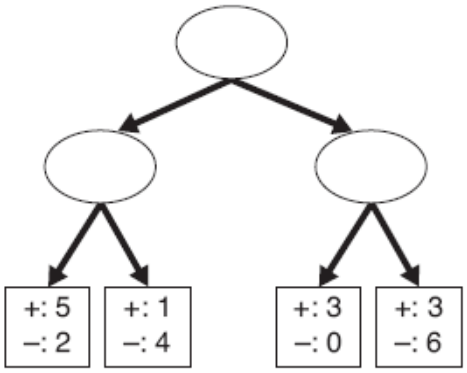
- 模型的复杂度对过分拟合有较大影响
- 理想的复杂度是能产生最低泛化误差的模型的复杂度
- 估计泛化误差，确定合理的模型复杂度

泛化误差估计

- 两个概念
  - 再代入误差：训练集上的误差（ Σ e(t) ）
  - 泛化误差：　测试集上的误差（ Σ e'(t)) ）
- 再代入估计法
  - e'(t) = e(t)
  - 4/24 vs. 6/24



Decision Tree, T_L



Decision Tree, T_R

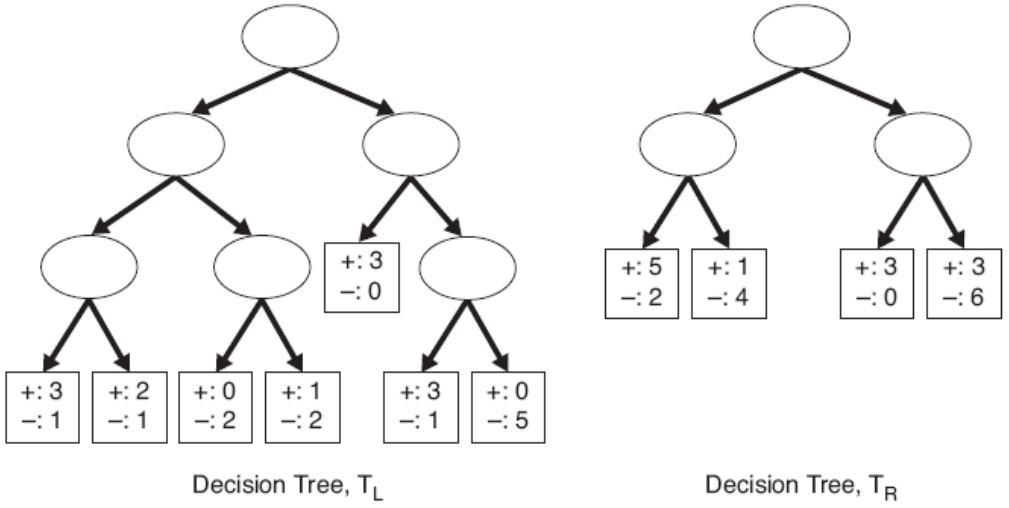
泛化误差估计

- 悲观估计法
- 复杂模型出现过分拟合的可能性更高
- **Occam剃刀原则：**
- 对于两个相似泛化误差的模型，应选择较简单的一个
- 需要在估计泛化误差时结合模型复杂度

泛化误差估计

- 复杂度模型惩罚项

$$e_g(T) = \frac{\sum_{i=1}^{k}[e(t_i)+\Omega(t_i)]}{\sum_{i=1}^{k} n(t_i)} = \frac{e(T)+\Omega(T)}{N_t}$$



惩罚项=0.5

（4+6×0.5）/24 vs. （6+3×0.5）/24

惩罚项=1

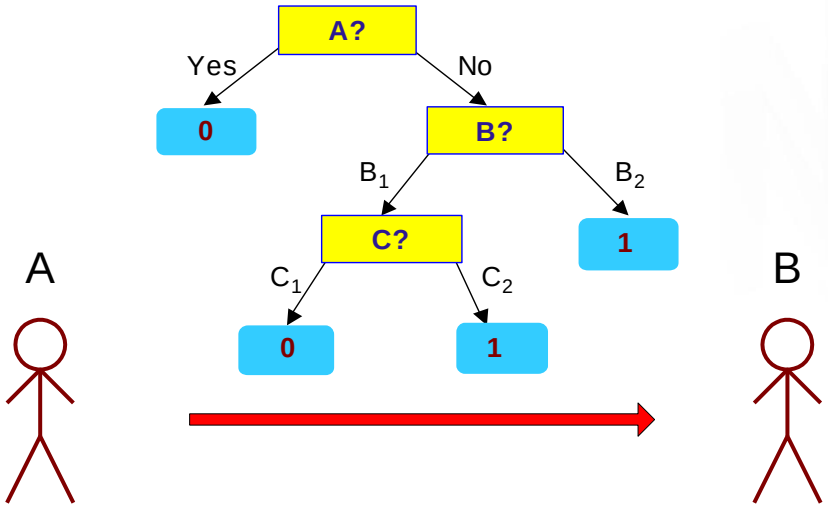
（4+6×1）/24 vs. （6+3×1）/24

Figure 4.27. Example of two decision trees generated from the same training data.

泛化误差估计

- 最小描述长度原则（MDL）

X	y
X_1	1
X_2	0
X_3	0
X_4	1
…	…
X_n	1

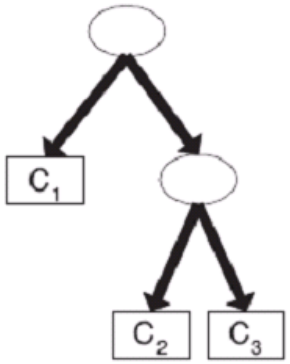


X	y
X_1	?
X_2	?
X_3	?
X_4	?
…	…
X_n	?

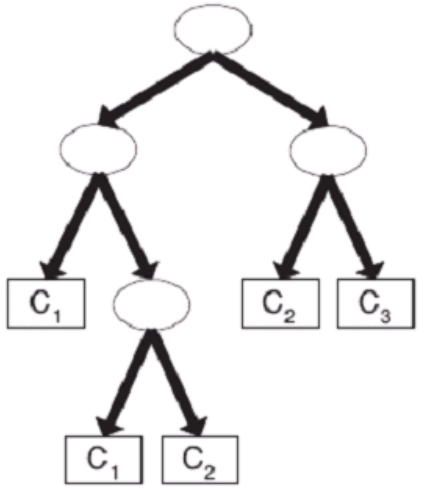
- Cost(Model,Data) = Cost(Data|Model) + Cost(Model)
  - Cost是编码需要的比特数
  - 寻找cost最小的模型
- Cost(Data|Model) 编码误分类信息
- Cost(Model) 是模型开销

泛化误差估计

- MDL 例子
- 16个二元属性
- 3个类
- 每个内部结点用划分属性的ID表示（$\log_2 m$）
- 每个叶结点用相关联类的ID表示（$\log_2 k$）



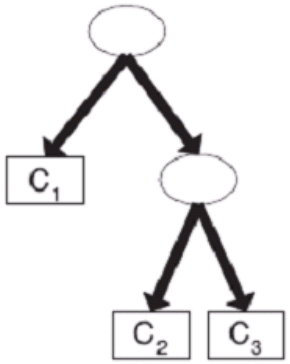
(a) Decision tree with 7 errors



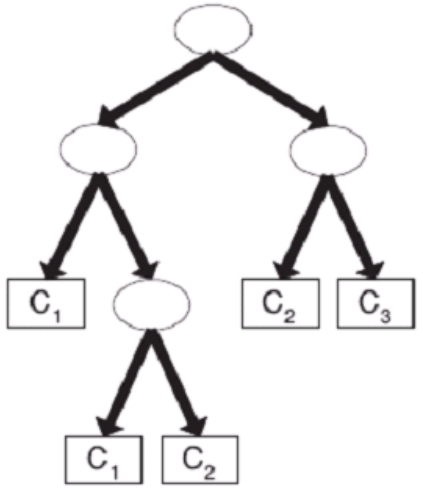
(b) Decision tree with 4 errors

泛化误差估计

- **Cost(tree)**是对决策树的所有结点编码的开销
- **Cost(data|tree)**是对决策树在训练集上分类错误编码的开销



(a) Decision tree with 7 errors



(b) Decision tree with 4 errors

- $2\times4+3\times2+7\times\log_2 n=14+7\times\log_2 n$
- $4\times4+5\times2+4\times\log_2 n=26+4\times\log_2 n$

泛化误差估计

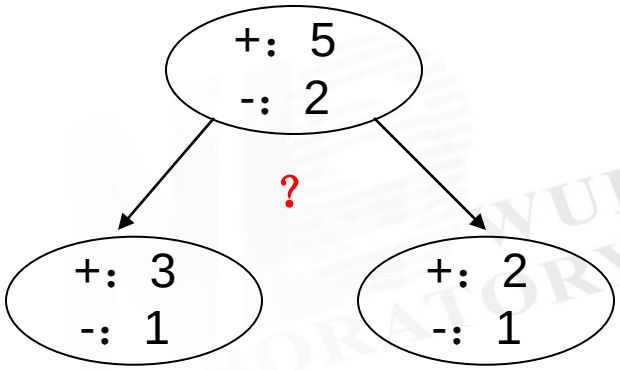
- 泛化误差可用训练误差的统计修正来估计
  - 泛化误差倾向于比训练误差大，所以统计修正通常是计算训练误差的上界

$$e_{upper}(N,e,a)=\frac{e+\frac{z_{a/2}^2}{2N}+z_{a/2}\sqrt{\frac{e(1-e)}{N}+\frac{z_{a/2}^2}{4N^2}}}{1+\frac{z_{a/2}^2}{N}}$$

  - a是置信水平，$z_{a/2}$是标准正态分布的标准化值，N是计算e的训练记录总数

泛化误差估计

- 例4.3
- 令置信度a=25%
- 划分前e=2/7, N=7
  - e_{upper}= 0.503
  - 误分类个数7×0.503=3.521
- 划分后e_1=1/4, N_1=4, e_2=1/3, N_2=3
  - e_{upper1}= 0.537, e_{upper2}= 0.650
  - 误分类个数4×0.537+3×0.650=4.098

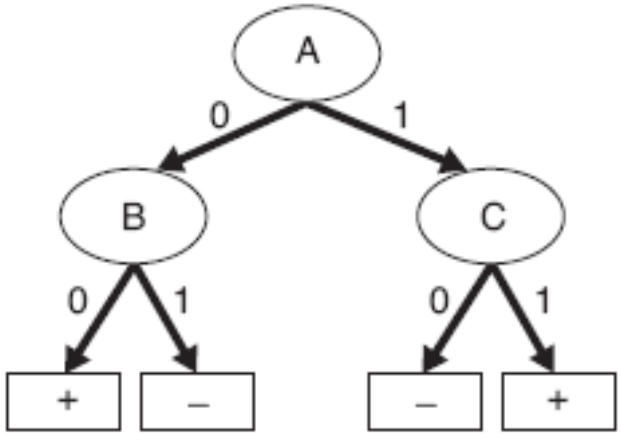


泛化误差估计

- 使用确认集
- 将原始的训练数据集分成两个较小的子集，一个用于训练，另一个作为确认集，用来估计泛化误差
- **2/3 vs. 1/3**
- 训练样本减少了

泛化误差估计

- 习题8



Training:

Instance	A	B	C	Class
1	0	0	0	+
2	0	0	1	+
3	0	1	0	+
4	0	1	1	–
5	1	0	0	+
6	1	0	0	+
7	1	1	0	–
8	1	0	1	+
9	1	1	0	–
10	1	1	0	–

Validation:

Instance	A	B	C	Class
11	0	0	0	+
12	0	1	1	+
13	1	1	0	+
14	1	0	1	–
15	1	0	0	+

- 1. 乐观估计的泛化差错率
- 2. 悲观估计的泛化差错率（惩罚项0.5）
- 3. 确认集估计的泛化差错率

处理丢失的属性值

- 丢失属性值主要从三个方面影响决策树的构建：
- 影响不纯性度量的计算
- 影响如何将具有丢失属性的记录划分到子结点中
- 影响如何分类具有丢失属性的测试记录

计算不纯性度量

Tid	Refund	Marital Status	Taxable Income	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	?	Single	90K	Yes

Missing value

Before Splitting:

Entropy(Parent)
= -0.3 log(0.3)-(0.7)log(0.7) = 0.8813

	Class = Yes	Class = No
Refund=Yes	**0**	**3**
Refund=No	**2**	**4**
Refund=?	**1**	**0**

Split on Refund:

Entropy(Refund=Yes) = 0

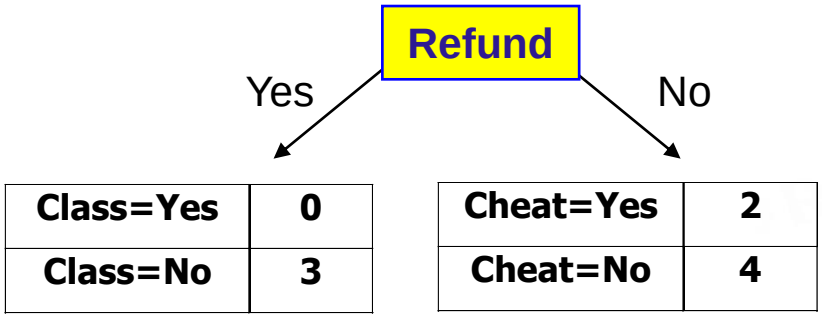
Entropy(Refund=No)
= -(2/6)log(2/6) – (4/6)log(4/6) = 0.9183

Entropy(Children)
= 0.3 (0) + 0.6 (0.9183) = 0.551

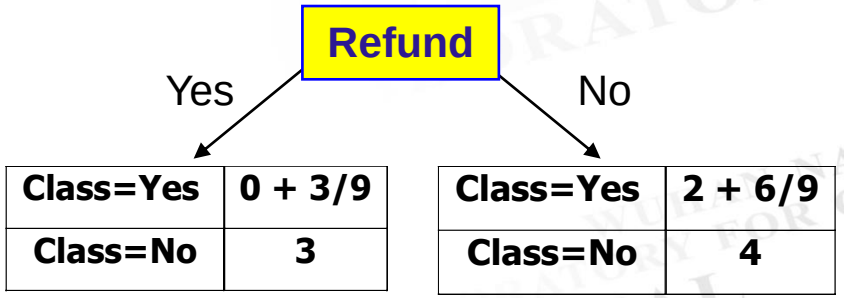
Gain = 0.9 × (0.8813 – 0.551) = 0.3303

划分结点

Tid	Refund	Marital Status	Taxable Income	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



Tid	Refund	Marital Status	Taxable Income	Class
10	?	Single	90K	Yes



Probability that Refund=Yes is 3/9

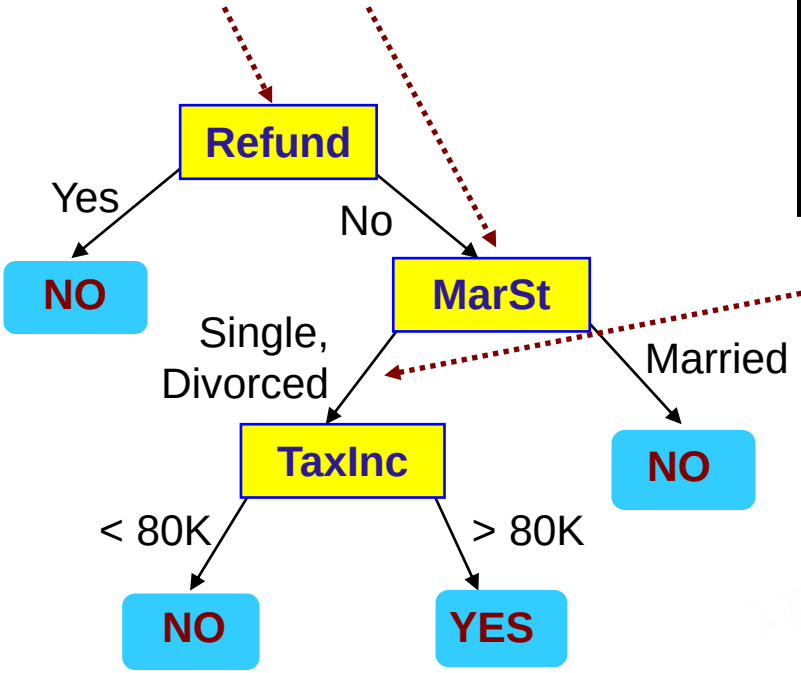
Probability that Refund=No is 6/9

Assign record to the left child with weight = 3/9 and to the right child with weight = 6/9

预测测试数据

New record:

Tid	Refund	Marital Status	Taxable Income	Class
11	No	?	85K	?



	Married	Single	Divorced	Total
Class=No	3	1	0	4
Class=Yes	0	1.67	1	2.67
Total	3	2.67	1	6.67

Probability that Marital Status = Married is 3/6.67

Probability that Marital Status ={Single,Divorced} is 3.67/6.67

分类器性能评估

- 性能评估的度量
  - 如何计算一个模型的性能？
- 性能评估的方法
  - 如何获得可靠的估计？
- 模型比较的方法
  - 如何比较不同模型之间的相对性能？

性能评估的度量

- 混淆矩阵

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

性能评估的度量

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

➢精度的定义:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

性能评估的度量

- 代价矩阵

	PREDICTED CLASS		
ACTUAL CLASS	**C(i|j)**	Class=Yes	Class=No
	Class=Yes	**C(Yes|Yes)**	**C(No|Yes)**
	Class=No	**C(Yes|No)**	**C(No|No)**

C(i|j): Cost of misclassifying class j example as class i

性能评估的度量

Cost Matrix	PREDICTED CLASS	
	C(i|j)	+	-
ACTUAL CLASS	+	-1	100
	-	1	0

Model M_1	PREDICTED CLASS	
	+	-
ACTUAL CLASS +	150	40
ACTUAL CLASS -	60	250

Accuracy = 80%

Cost = 3910

Model M_2	PREDICTED CLASS	
	+	-
ACTUAL CLASS +	250	45
ACTUAL CLASS -	5	200

Accuracy = 90%

Cost = 4255

代价 vs. 精度

Count	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

Accuracy is proportional to cost if

1. C(Yes|No)=C(No|Yes) = q
2. C(Yes|Yes)=C(No|No) = p

$N = a + b + c + d$

$Accuracy = (a + d)/N$

Cost	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	p	q
	Class=No	q	p

$$Cost = p\,(a + d) + q\,(b + c)$$
$$= p\,(a + d) + q\,(N - a - d)$$
$$= q\,N - (q - p)(a + d)$$
$$= N\,[q - (q-p) \times Accuracy]$$

性能评估的方法

- 如何获得可靠的性能估计？

- 除了学习算法外，模型的性能还依赖很多其他的因素：
  - 类分布
  - 误分类代价
  - 训练和测试集大小

性能评估的方法

- 保持法
  - 将原始数据划分成两个不相交的集合，分别为训练集和检验集
  - 如：2/3用来训练，1/3用来测试
- 局限性：
  - 用于训练的被标记样本较少
  - 模型依赖于训练集和检验集的构成
  - 训练集和检验集不是相互独立的

性能评估的方法

- 随机抽样法
  - 重复式的保持法
  - 总准确率 $acc_{sub} = \sum_{i=1}^{k} acc_i / k$
  - 没有控制每个记录用于训练和检验的次数，有些记录用于训练可能比其他记录频繁

性能评估的方法

- **Bootstrap**
  - 训练记录采用有放回抽样
  - 每个记录被自助抽样抽取的概率约为**63.2%**
  - **$1-(1-1/N)^N$**
  - 大小为**N**的自助样本大约包含原始数据中**63.2%**的记录

性能评估的方法

- 交叉检验
  - 将数据分成k个不相交的子集
  - K折交叉验证: 在k-1个部分上训练，在另一份上测试
  - 留一： k即为数据集的大小N
- 计算开销很大