大数据分析

分类分析模型

相亲问题

一个妈妈要给自己的女儿介绍男朋友

- 女儿：多大年纪了？
- 母亲：26
- 女儿：长的帅不帅？
- 母亲：挺帅的
- 女儿：收入高不？
- 母亲：不算很高，中等情况。
- 女儿：是公务员不？
- 母亲：是，在税务局上班呢。
- 女儿：那好，我去见见。

明天适合打球吗？

日期	天气	温度(华氏度)	湿度	起风	打球?
1	晴	85	85	F	No
2	晴	80	90	T	No
3	阴	83	78	F	Yes
4	雨	70	96	F	Yes
5	雨	68	80	F	Yes
6	雨	65	70	T	No
7	阴	64	65	T	Yes
8	晴	72	95	F	No
9	晴	69	70	F	Yes
10	雨	75	80	F	Yes
11	晴	75	70	T	Yes
12	阴	72	90	T	Yes
13	阴	81	75	F	Yes
14	雨	71	80	T	No
15	阴	85	90	F	?
16	雨	80	79	F	?
17	晴	78	70	T	?

分类

- 在已知研究对象可分为若干类的情况下，确定新的对象属于哪一类。

- Given a collection of records (training set)
  - Each record is by characterized by a tuple (x,y), where x is the attribute set and y is the class label
- Task:
  - Learn a model that maps each attribute set x into one of the predefined class labels y

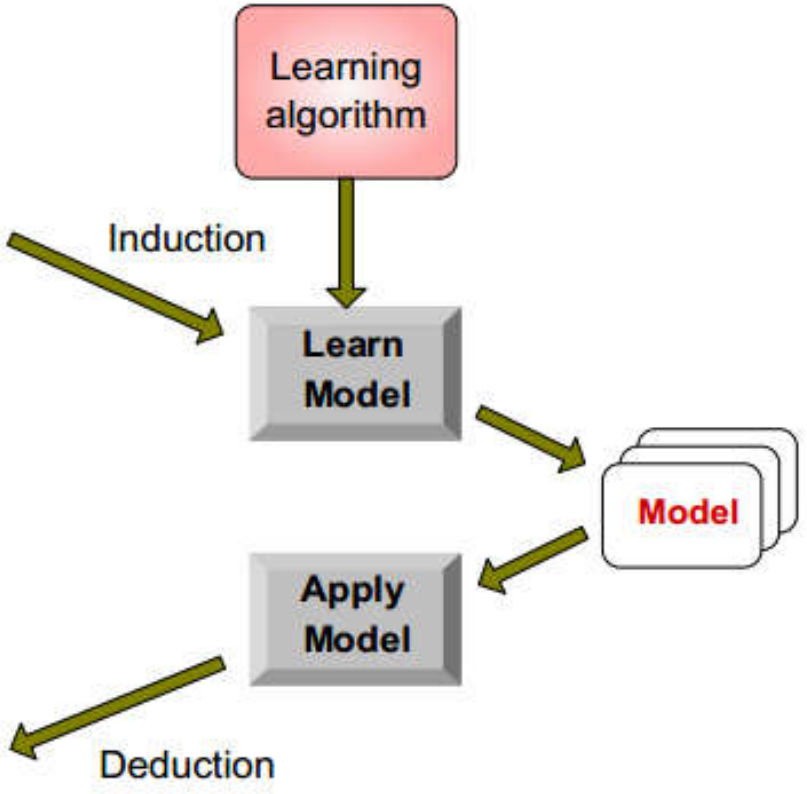
建立分类模型的一般方法

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

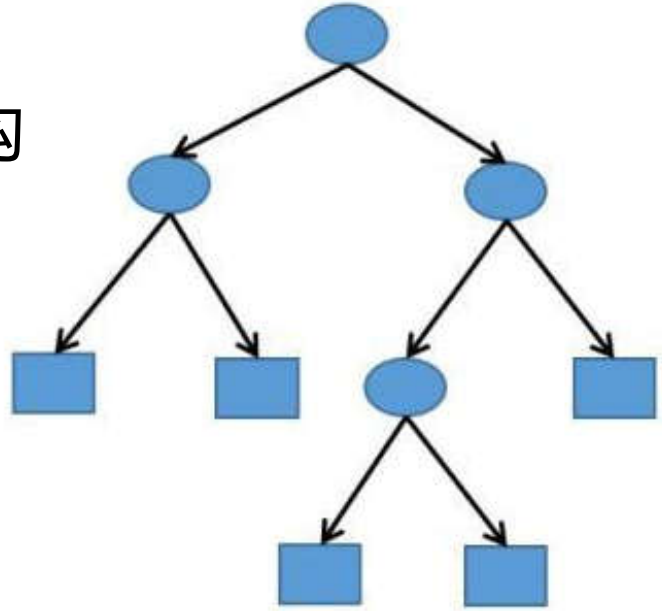


分类技术

- 决策树
- 最近邻分类器
- 朴素贝叶斯分类器
- 基于规则的分类器
- 支持向量机
- 人工神经网络
- 。。。

决策树

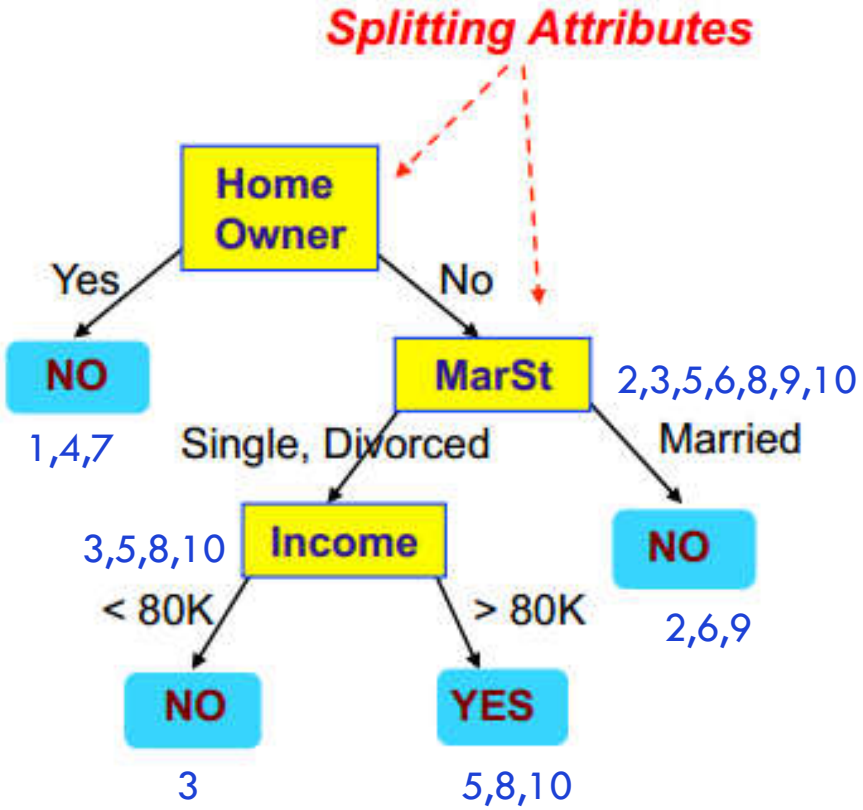
- 决策树是一种由节点和有向边组成的层次结构
- 根节点：没有入边，但有零条或多条出边
- 内部节点：恰有一条入边，两条或多条出边
- 叶节点：恰有一条入边，但没有出边



决策树建模

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

Training Data

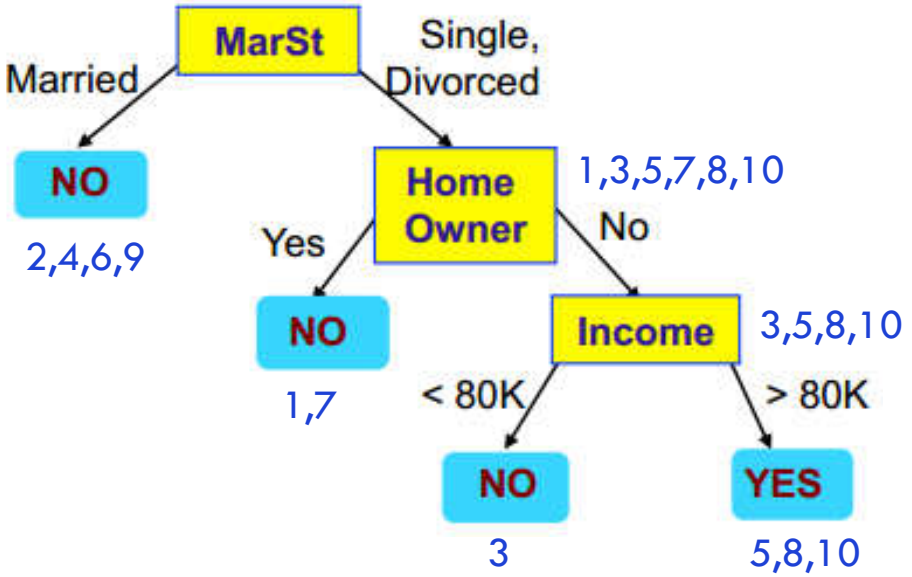


Model: Decision Tree

决策树建模（续）

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

Training Data



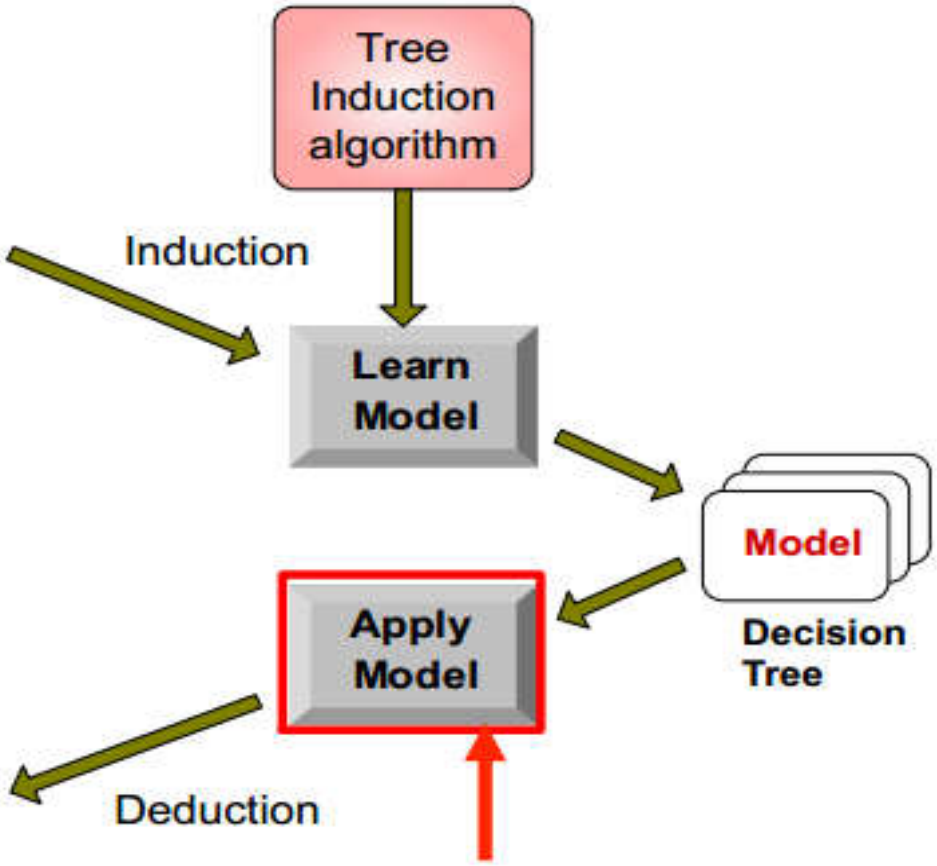
决策树应用

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

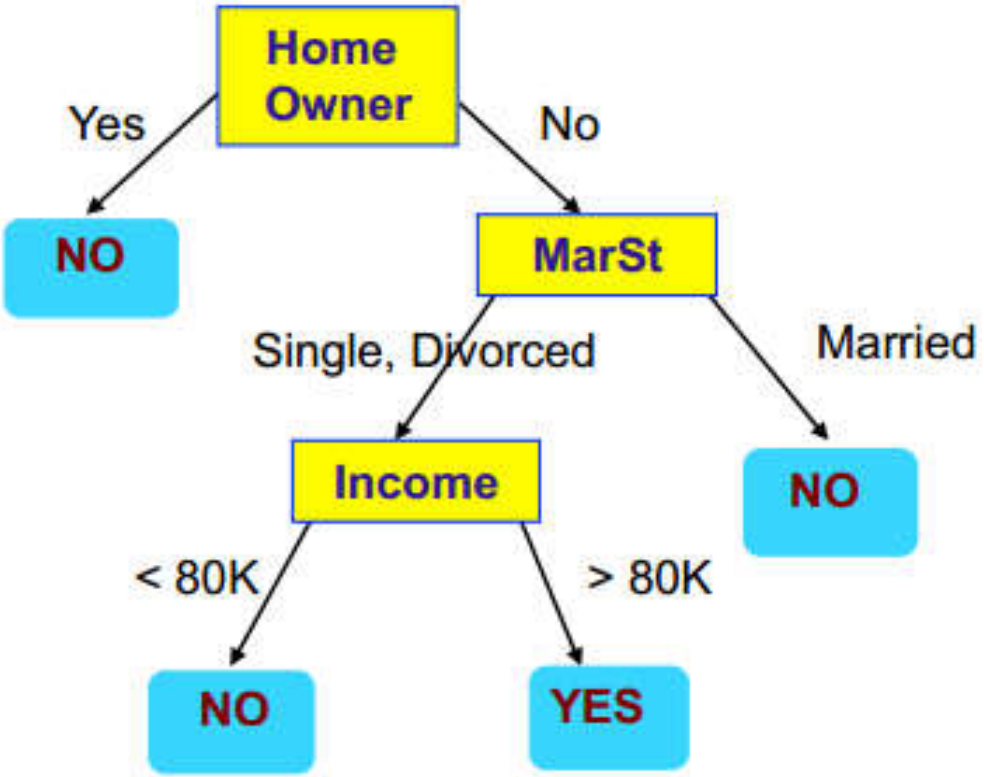
Training Set

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



决策树应用（续）



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

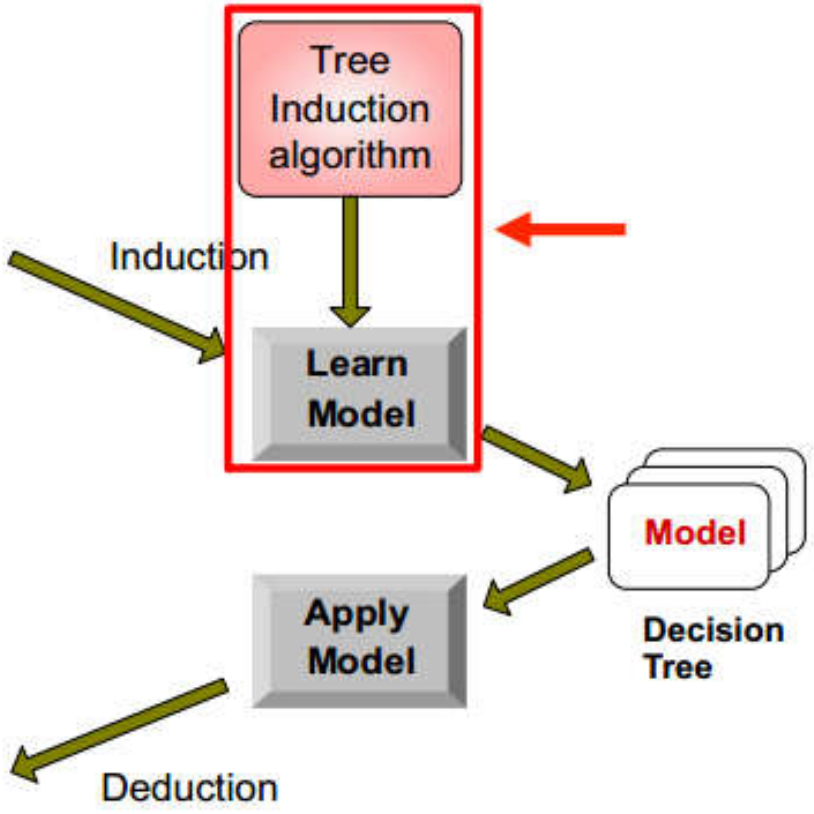
如何建立决策树

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



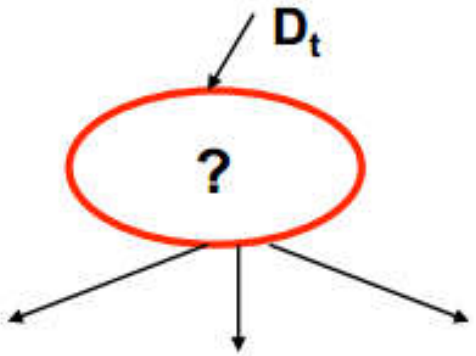
决策树算法

- Many Algorithms:
  - Hunt's Algorithm (one of the earliest)
  - CART
  - ID3, C4.5

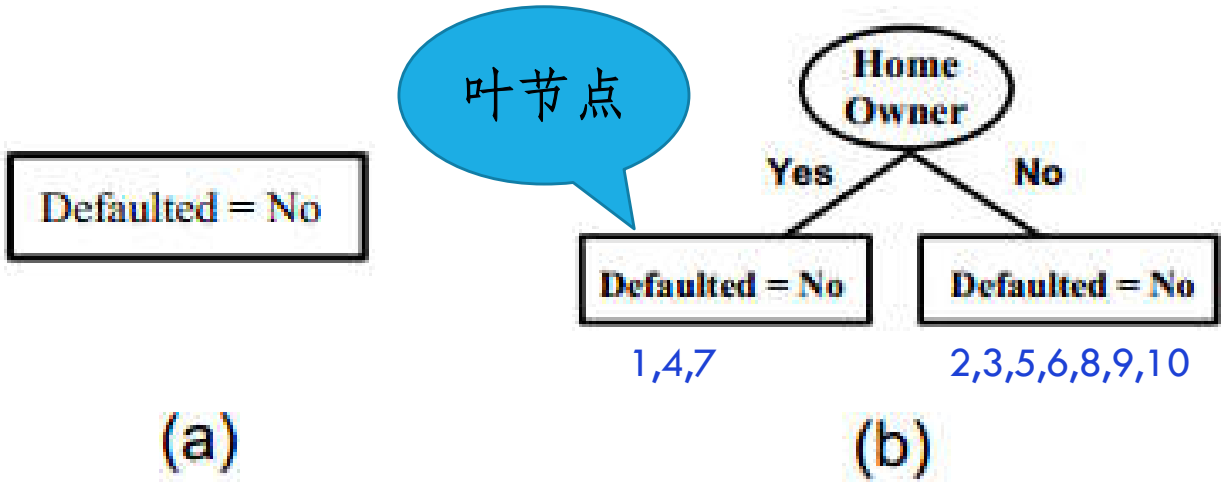
HUNT算法

- 设D_t是与节点t相关联的训练记录集，y_i是类标号
- Hunt算法的递归定义：
  - （1）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；
  - （2）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，创建一个子女结点，并根据测试结果将D_t中的记录分布到子女结点中。对于每个子女结点，递归地调用该算法。

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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



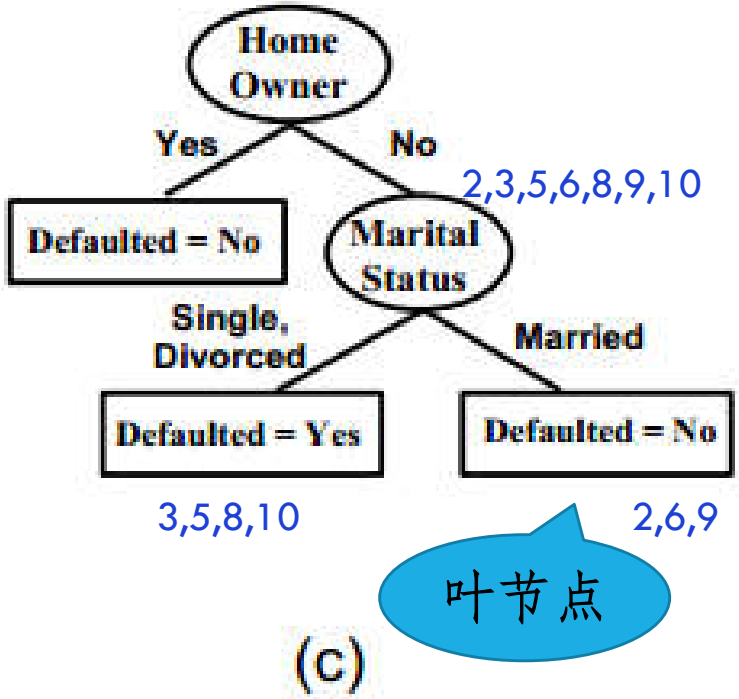
HUNT算法（续）

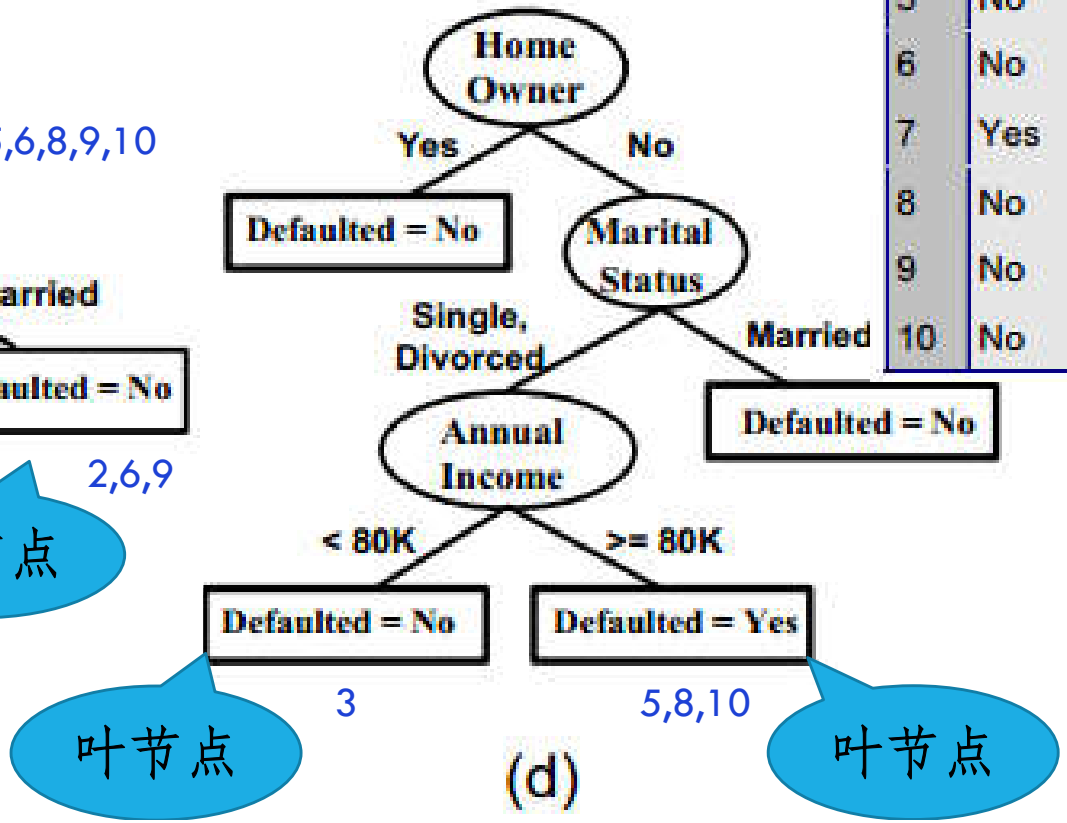


ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

HUNT算法（续）

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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





决策树的设计问题

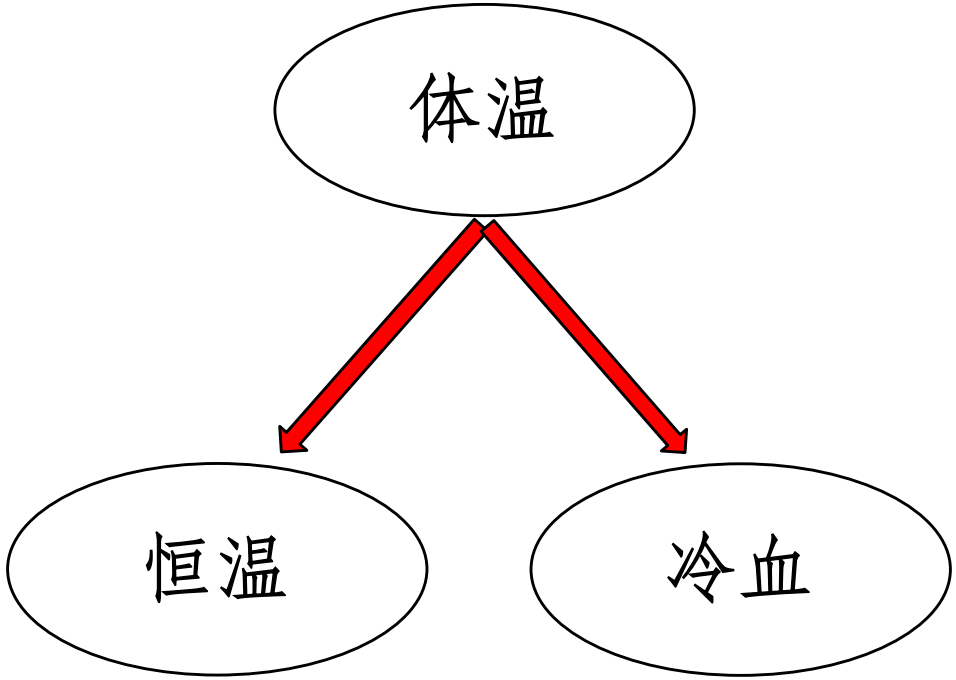
- 如何分裂训练记录？
- 如何停止分裂过程？

表示属性的方法

- Depends on attribute types
  - Binary
  - Nominal
  - Ordinal
  - Continuous

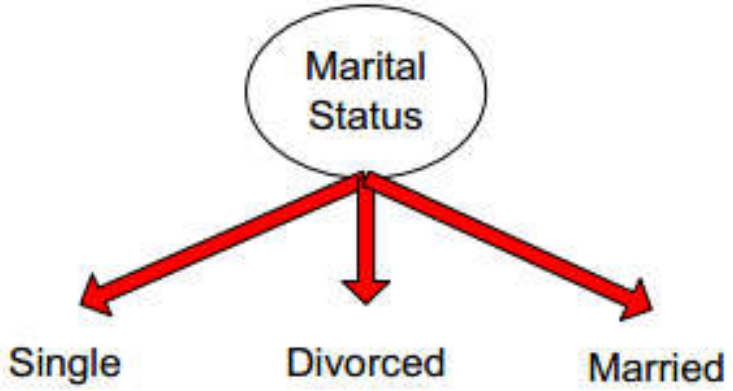
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

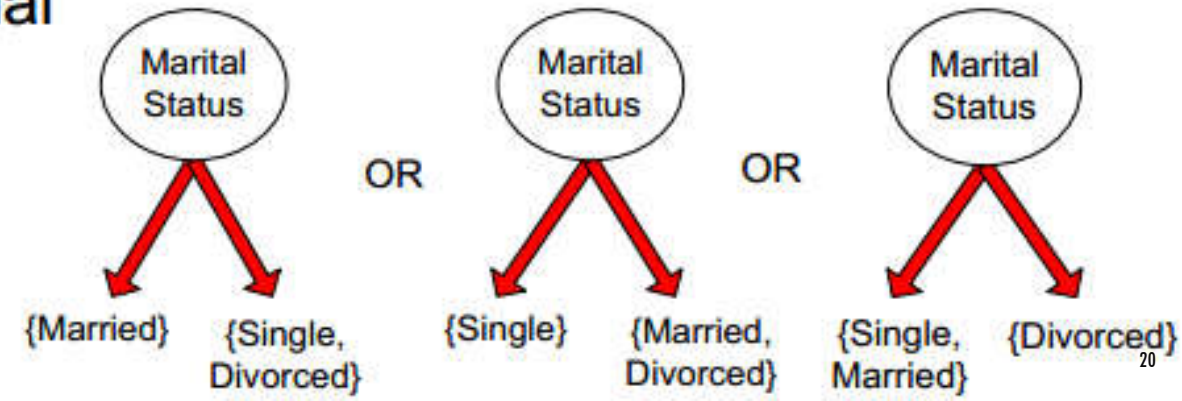
二元属性



标称属性

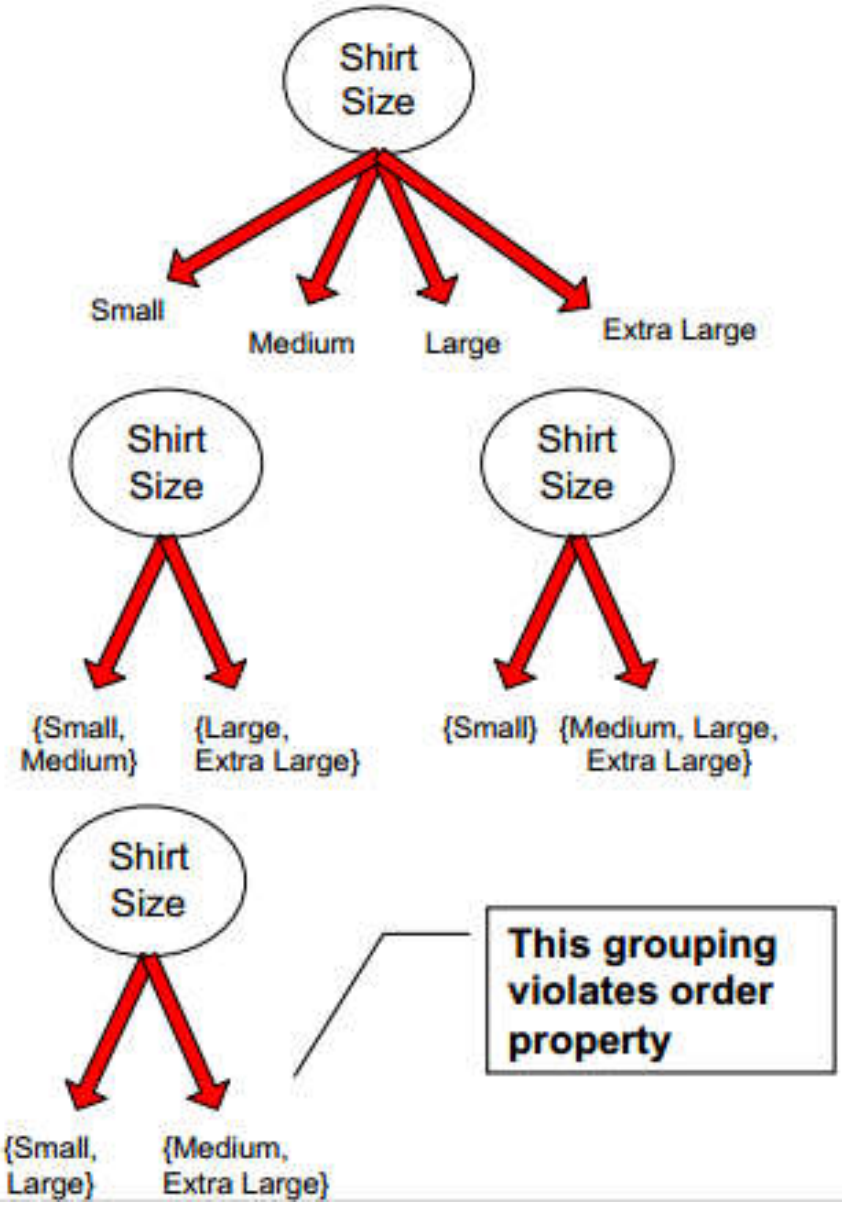
- Multi-way split:
  - Use as many partitions as distinct values.
- Binary split:
  - Divides values into two subsets
  - Need to find optimal partitioning.





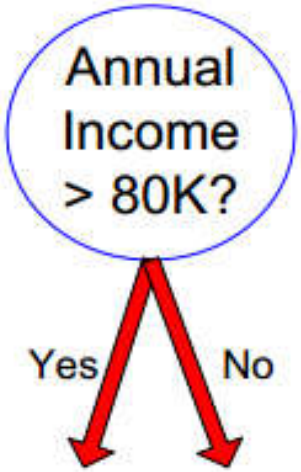
序数属性

- Multi-way split:
  - Use as many partitions as distinct values
- Binary split:
  - Divides values into two subsets
  - Need to find optimal partitioning
  - Preserve the order property among attribute values



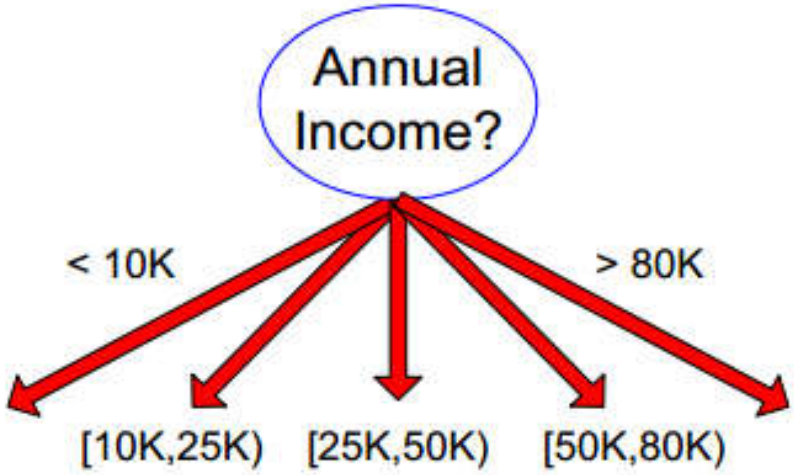
连续属性

二元划分 $A < v$ 或 $A \geq v$



(i) Binary split

多路划分 $v_i \leq A < v_{i+1}\ (i = 1, 2 \ldots, k)$



(ii) Multi-way split

划分度量方法

- Greedy approach:
  - Nodes with purer class distribution are preferred

- Need a measure of node impurity:

C0: 5 C1: 5	C0: 9 C1: 1
High degree of impurity	**Low degree of impurity**

不纯性度量

- Gini Index

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

- Entropy

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

- Misclassification error

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

最佳划分标准

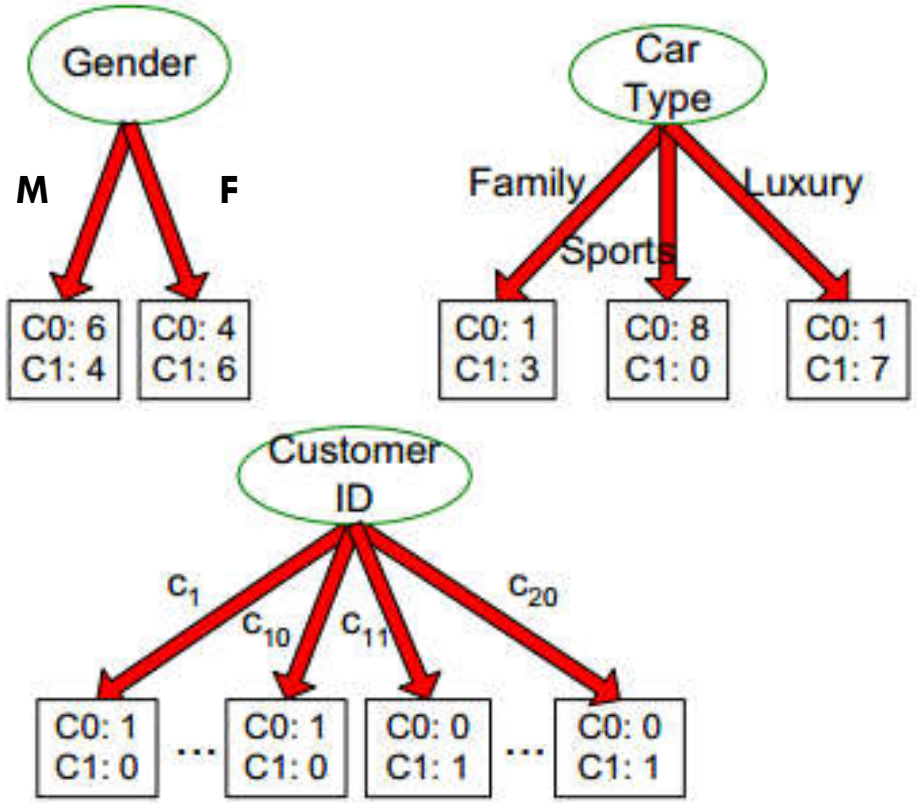
- 比较父节点（划分前）的不纯程度P和子女节点（划分后）的不纯程度M（加权平均）
- 它们的差越大，属性测试条件的效果越好

Choose the attribute test condition that produces the highest gain

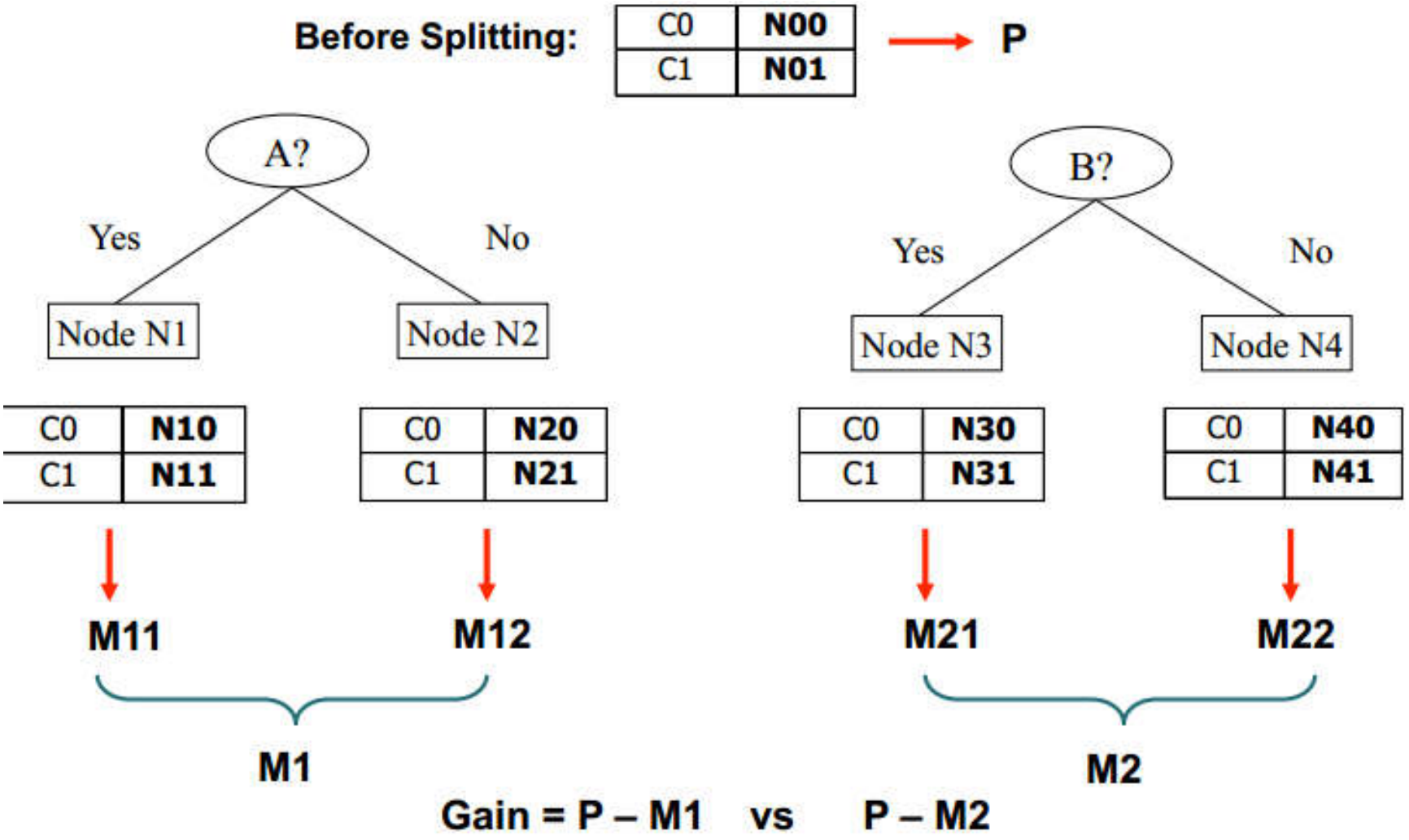
Gain = P – M

or equivalently, lowest impurity measure after splitting (M)

举例



Customer Id	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1


Before Splitting:
C0 N00
C1 N01
P
A?
Yes
No
Node N1
Node N2
B?
Yes
No
Node N3
Node N4
C0 N10
C1 N11
C0 N20
C1 N21
C0 N30
C1 N31
C0 N40
C1 N41
M11
M12
M21
M22
M1
M2
Gain = P – M1 vs P – M2

基尼指数

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

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

基尼指数（续）

- When a node p is split into k partitions (children)

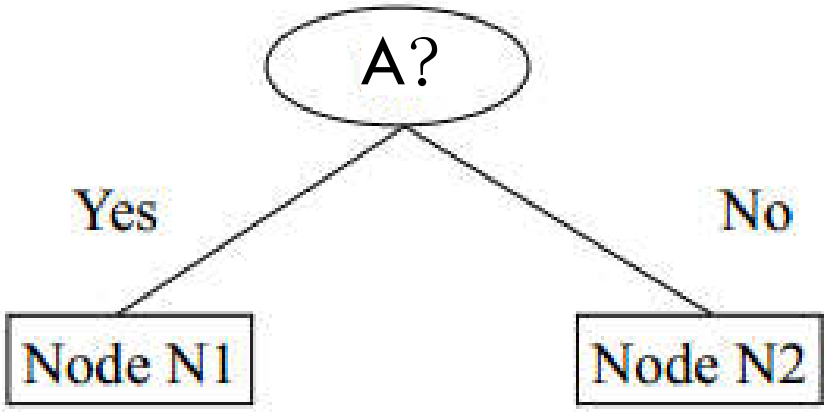
$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

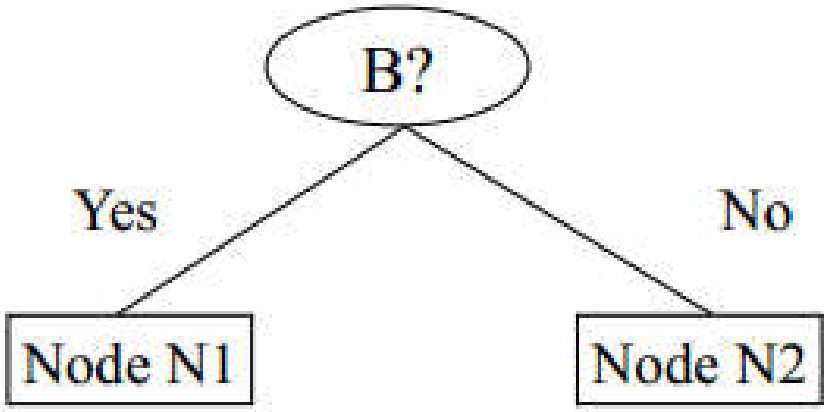
where, n_i = number of records at child i,

n = number of records at parent node p.

- Choose the attribute that minimizes weighted average Gini index of the children

二元划分的基尼指数





	Parent
C1	6
C2	6
Gini = 0.500	

A	N1	N2
C1	4	2
C2	3	3
Gini=0.486		

B	N1	N2
C1	1	5
C2	4	2
Gini=0.371		

多路划分的基尼指数

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

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini			

Two-way split
(find best partition of values)

	CarType	
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini		

	CarType	
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini		

连续属性的基尼指数

- 考虑二元划分“年收入≤ v”
- 按照年收入将训练记录排序
- 从两个相邻的排过序的属性值中选择中间值作为候选划分点
- 计算每个候选划分点的Gini值，从中选择具有最小值的候选划分点

ID	Home Owner	Marital Status	Annual Income	Defaulted
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

连续属性的基尼指数（续）

ID	Home Owner	Marital Status	Annual Income	Defaulted
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

Sorted Values →
Split Positions →

Cheat	No		No		No		Yes		Yes		Yes		No		No		No		No		No	
Annual 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
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	

- 进一步优化：仅考虑位于不同类标号的两个相邻记录之间的候选划分点

熵

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

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

信息增益

Gain = P – M

- Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

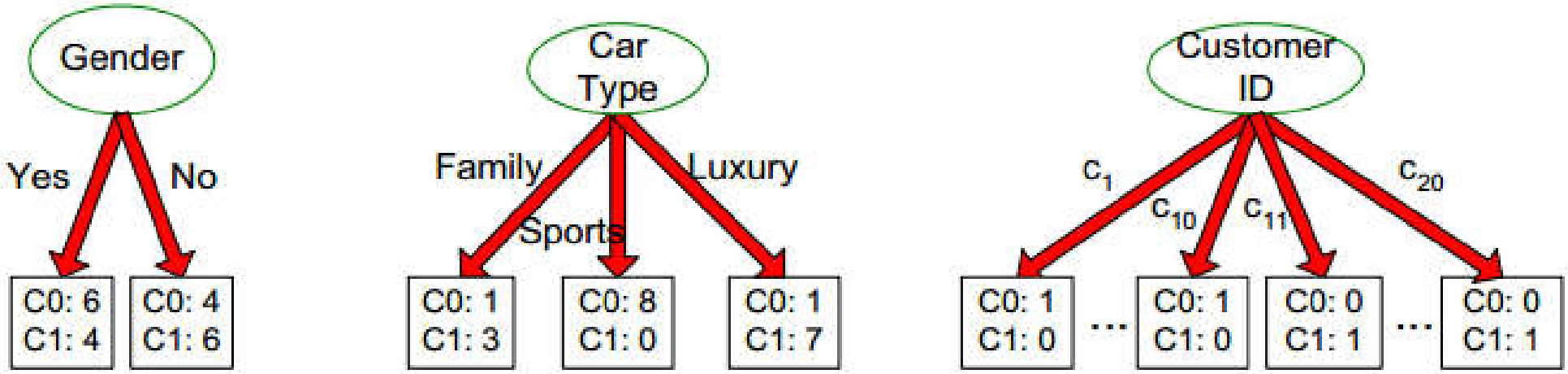
Parent Node, p is split into k partitions;

n_i is number of records in partition i

  - Choose the split that achieves most reduction (maximizes GAIN)

信息增益（续）

- Info Gain tends to prefer splits that result in large number of partitions, each being small but pure



  - Customer ID has highest information gain because entropy for all the children is zero

增益率

- Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \qquad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions

n_i is the number of records in partition i

– Adjusts Information Gain by the entropy of the partitioning (SplitINFO).

分类误差

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

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

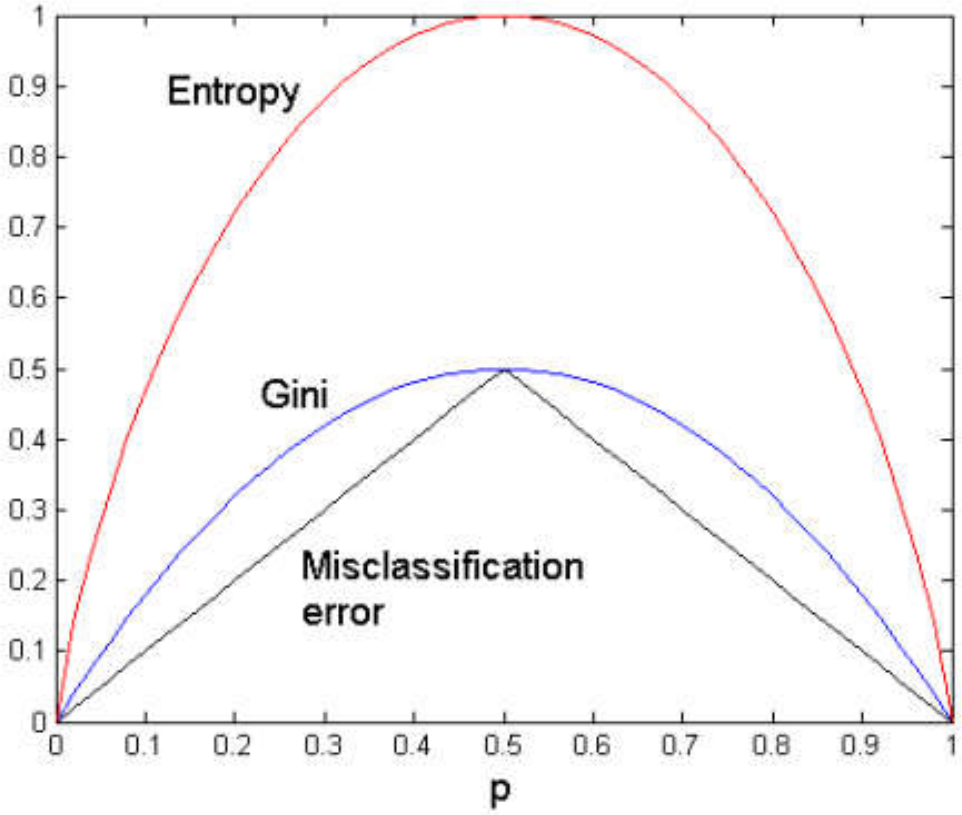
C1	2
C2	4

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

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

不纯性度量之间的比较

For a 2-class problem:



Consider the training examples shown in Table for a binary classification problem.

(a) Compute the Entropy for the overall collection of training examples.

(b) Compute the Entropy for the Movie ID attribute.

(c) Compute the Entropy for the Format attribute.

(d) Compute the Entropy for the Movie Category attribute using multiway split.

(e) Which of the three attributes has the lowest Entropy? Which one will you use for splitting at the root node? Briefly explain your choice

Movie ID	Format	Movie Category	Class
1	DVD	Entertainment	C0
2	DVD	Comedy	C0
3	DVD	Documentaries	C0
4	DVD	Comedy	C0
5	DVD	Comedy	C0
6	DVD	Comedy	C0
7	Online	Comedy	C0
8	Online	Comedy	C0
9	Online	Comedy	C0
10	Online	Documentaries	C0
11	DVD	Comedy	C1
12	DVD	Entertainment	C1
13	Online	Entertainment	C1
14	Online	Documentaries	C1
15	Online	Documentaries	C1
16	Online	Documentaries	C1
17	Online	Documentaries	C1
18	Online	Entertainment	C1
19	Online	Documentaries	C1
20	Online	Documentaries	C1

(a) There are two classes, *C0: 10* and *C1: 10*. Entropy can be calculated as
Entropy(*Class*) = -(10/20) □ log2(10/20) - (10/20) □ log2(10/20) = 1

(b) For attribute “Movie ID", there are 20 different values.
Entropy(*Movie ID*) = -20 □ (1/20) □ [(0/1) □ log2(0/1) + (1/1) □ log2(1/1)] = 0

(c) For attribute “Format”, there are 2 classes, *DVD: 8*, *Online: 12*.
Entropy(*F ormat*) = -(8/20) □ [(6/8) □ log2(6/8) + (2/8) □ log2(2/8)
- (12/20) □ [(4/12) □ log2(4/12) + (8/12) □ log2(8/12)] = 0.87

(d) For attribute “Movie Category", there are 3 classes, *Comedy: 8*, *Documentaries: 8*, *Entertainment: 4*.
Entropy(*MovieCategory*) = -(4/20) □ [(1/4) □ log2(1/4) + (3/4) □ log2(3/4)]
- (8/20) □ [(7/8) □ log2(7/8) + (1/8) □ log2(1/8)]
- (8/20) □ [(2/8) □ log2(2/8) + (6/8) □ log2(6/8)] = 0.70

(e) The entropy of attribute ”Movie ID" has the lowest entropy.

(f) Generally, we would not use ”Movie ID" as the node since it is just the record ID.
If the split node is “Format" , information gain is
Gain(*F ormat*) = 1-0.87=0.13
If the split node is “Movie Category" ,information gain is
Gain(*Movie Category*) = 1-*0.70*= 0.30
Basically, we would choose the root node which gives the most information gain, which is “Movie Category".

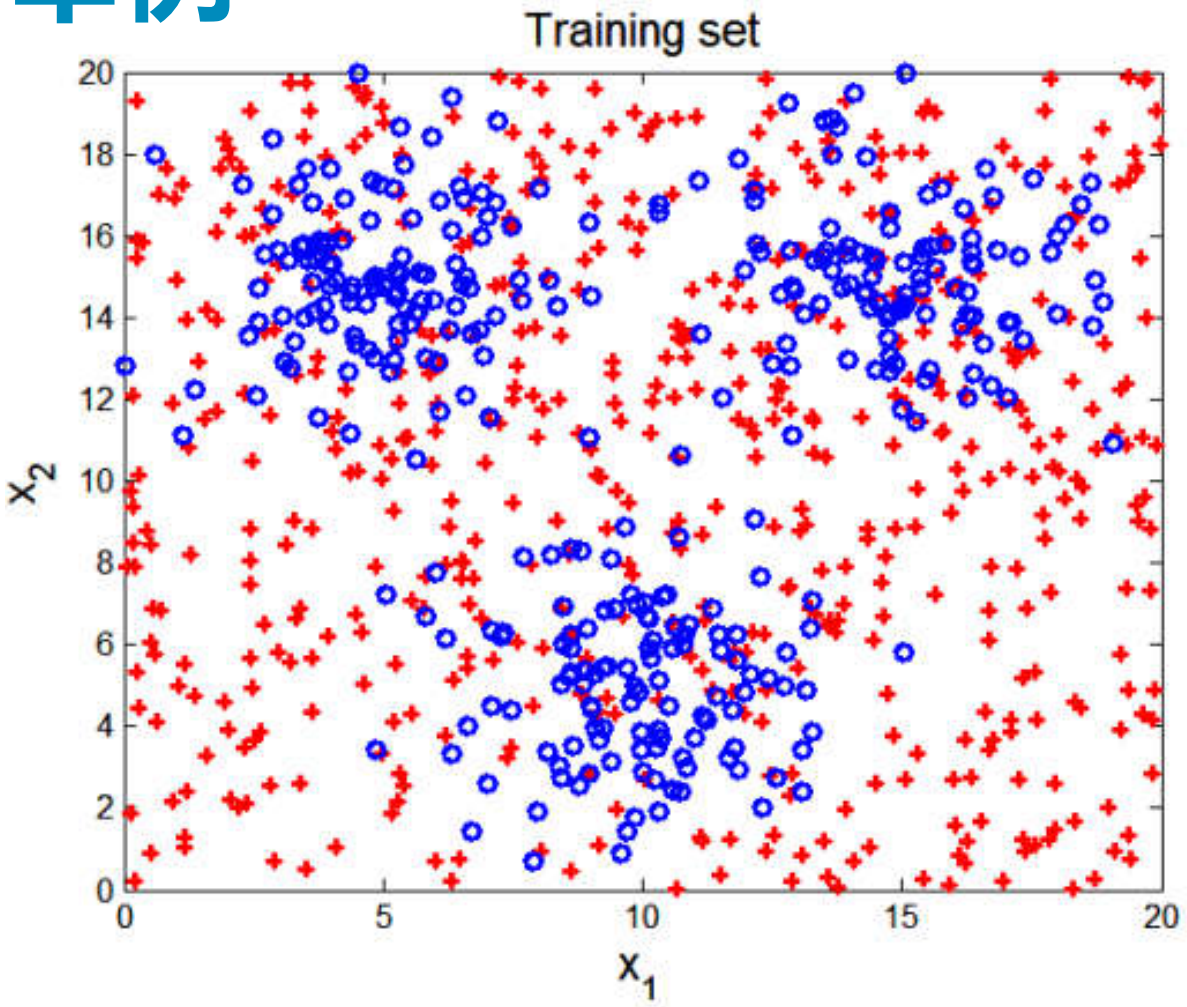
决策树的特点

- 构建代价小
- 未知样本分类速度快
- 抗噪声能力强
- 准确率可与其他算法相媲美（简单数据集）
- 。。。

模型的误差

- Training errors (apparent errors)
  - Errors committed on the training set
- Test errors
  - Errors committed on the test set
- Generalization errors
  - Expected error of a model over random selection of records from same distribution

举例



Two class problem:

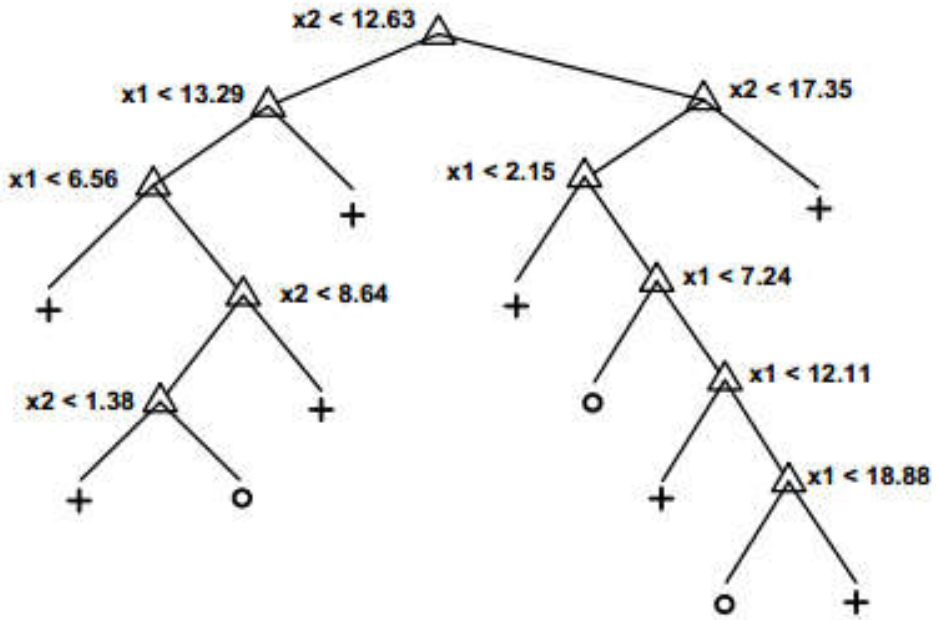
+, o

3000 data points (30% for training, 70% for testing)

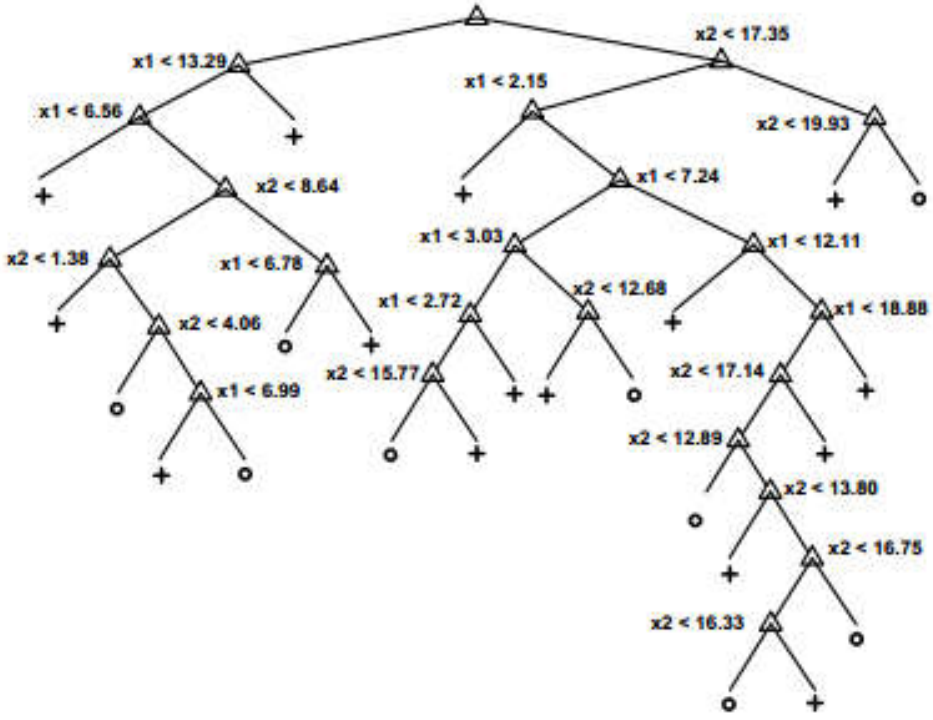
Data set for + class is generated from a uniform distribution

Data set for o class is generated from a mixture of 3 gaussian distributions, centered at (5,15), (10,5), and (15,15)

决策树

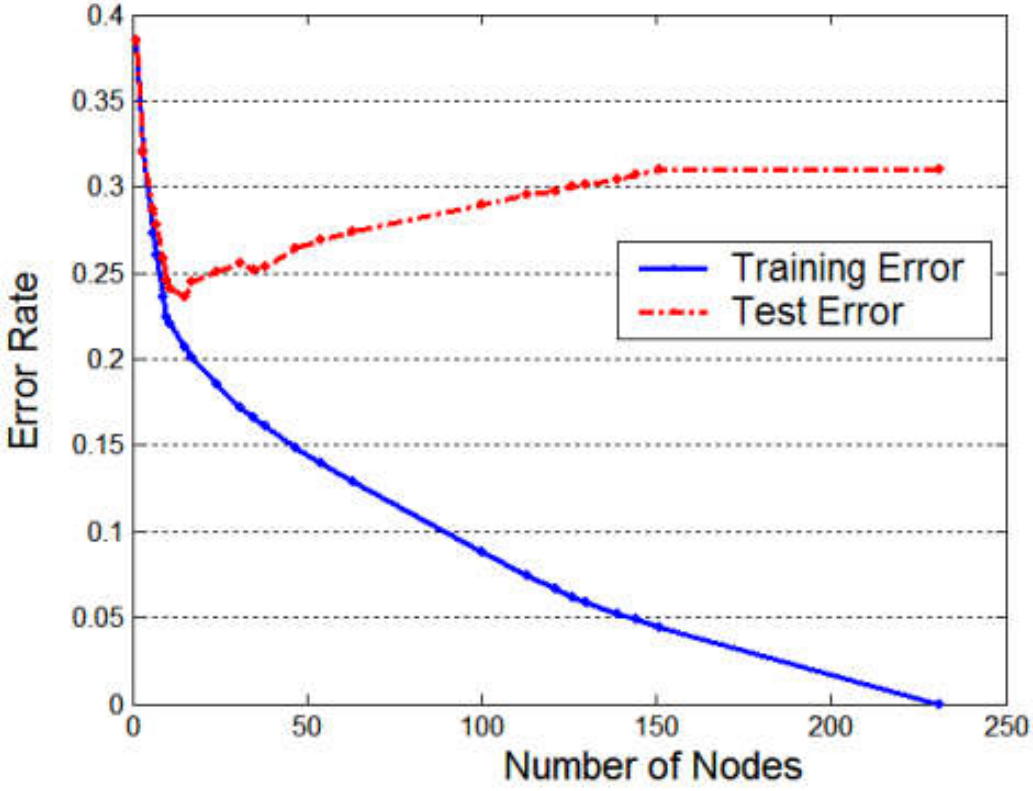


Decision Tree with 11 leaf nodes



Decision Tree with 24 leaf nodes

模型误差



Underfitting: when model is too simple, both training and test errors are large

Overfitting: when model is too complex, training error is small but test error is large

造成模型过分拟合的原因

- 噪声导致的过分拟合
- 缺乏代表性样本导致的过分拟合
- 。。。

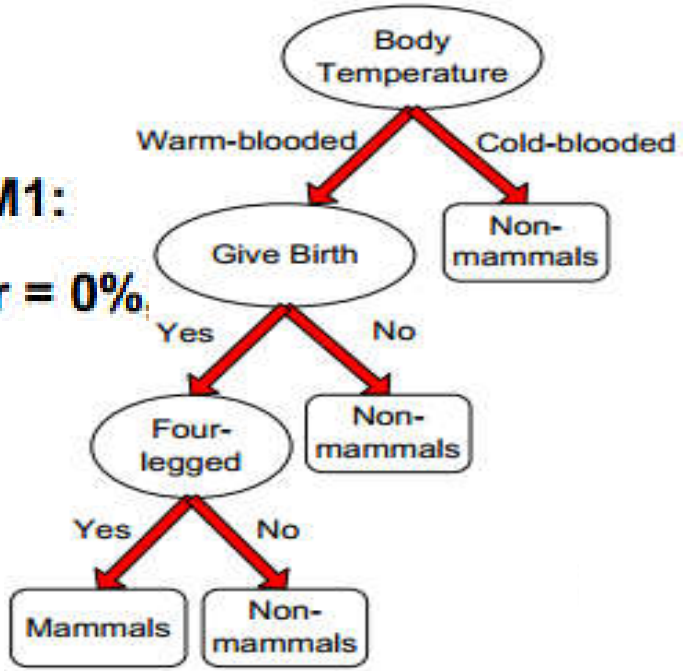
噪声导致的过分拟合

Training Set:

名称	体温	胎生	4条腿	冬眠	类标号
豪猪	恒温	是	是	是	是
猫	恒温	是	是	否	是
蝙蝠	恒温	是	否	是	否*
鲸	恒温	是	否	否	否*
蝾螈	冷血	否	是	是	否
巨蜥	冷血	否	是	否	否
蟒蛇	冷血	否	否	是	否
鲑鱼	冷血	否	否	否	否
鹰	恒温	否	否	否	否
虹鳉	冷血	是	否	否	否

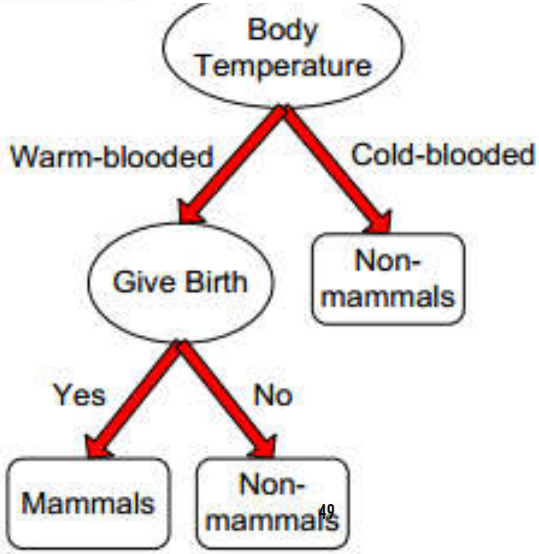
Model M1:

train err = 0%



Model M2:

train err = 20%



噪声导致的过分拟合

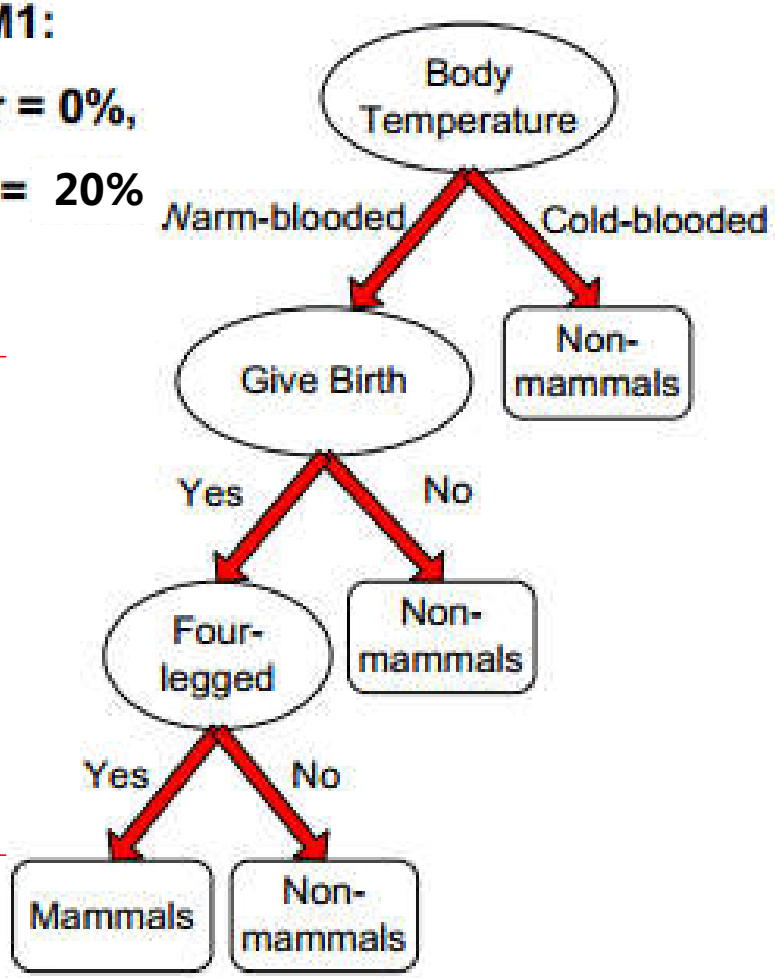
Model M1:

train err = 0%,

test err = 20%

Test Set:

名称	体温	胎生	4条腿	冬眠	类标号
人	恒温	是	否	否	否
鸽子	恒温	否	否	否	否
象	恒温	是	是	否	是
豹纹鲨	冷血	是	否	否	否
海龟	冷血	否	是	否	否
企鹅	冷血	否	否	否	否
鳗	冷血	否	否	否	否
海豚	恒温	是	否	否	否
希拉毒蜥	冷血	否	是	是	否



噪声导致的过分拟合

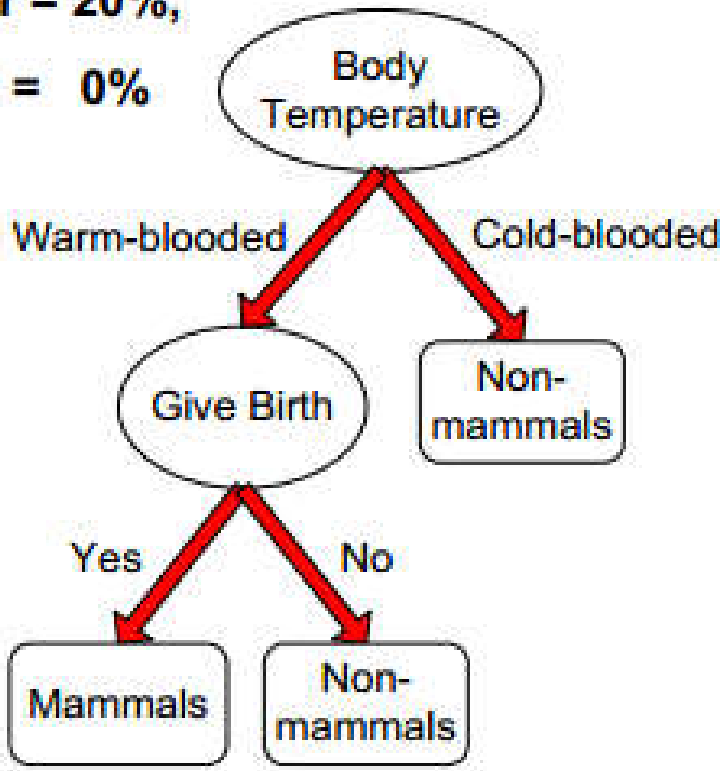
Test Set:

名称	体温	胎生	4条腿	冬眠	类标号
人	恒温	是	否	否	是
鸽子	恒温	否	否	否	否
象	恒温	是	是	否	是
豹纹鲨	冷血	是	否	否	否
海龟	冷血	否	是	否	否
企鹅	冷血	否	否	否	否
鳗	冷血	否	否	否	否
海豚	恒温	是	否	否	是
希拉毒蜥	冷血	否	是	是	否

Model M2:

train err = 20%,

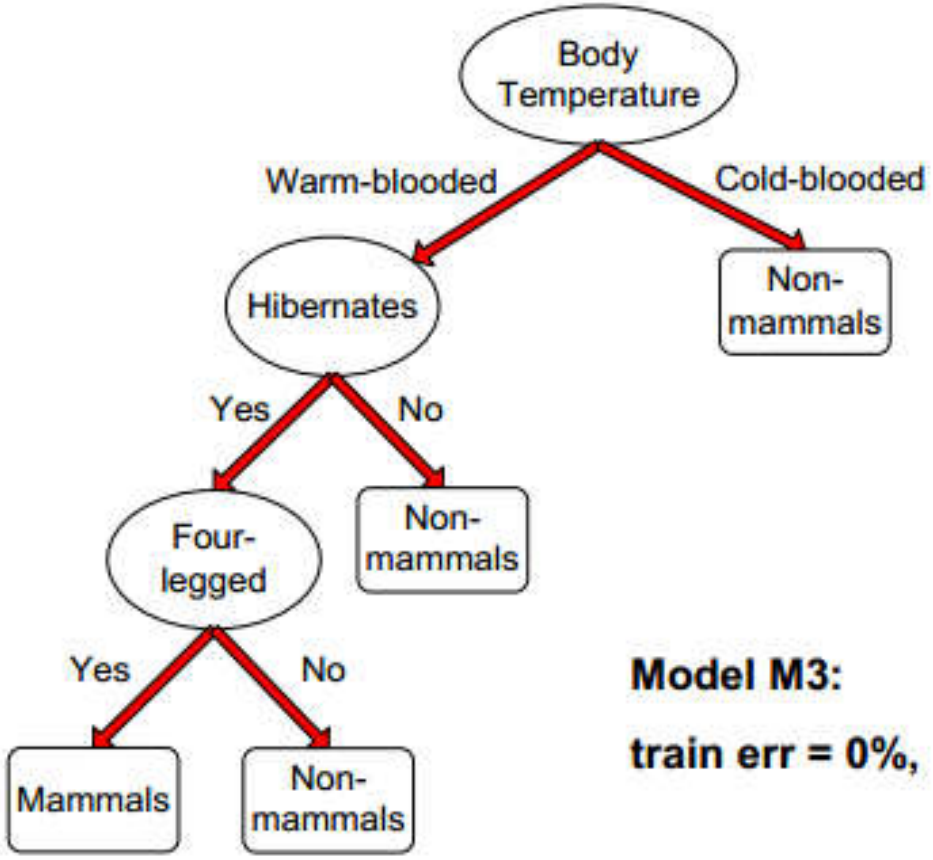
test err = 0%



缺乏代表性样本导致的过分拟合

Training Set:

名称	体温	胎生	4条腿	冬眠	类标号
蝾螈	冷血	否	是	是	否
虹鳉	冷血	是	否	否	否
鹰	恒温	否	否	否	否
弱夜鹰	恒温	否	否	是	否
鸭嘴兽	恒温	否	是	是	是



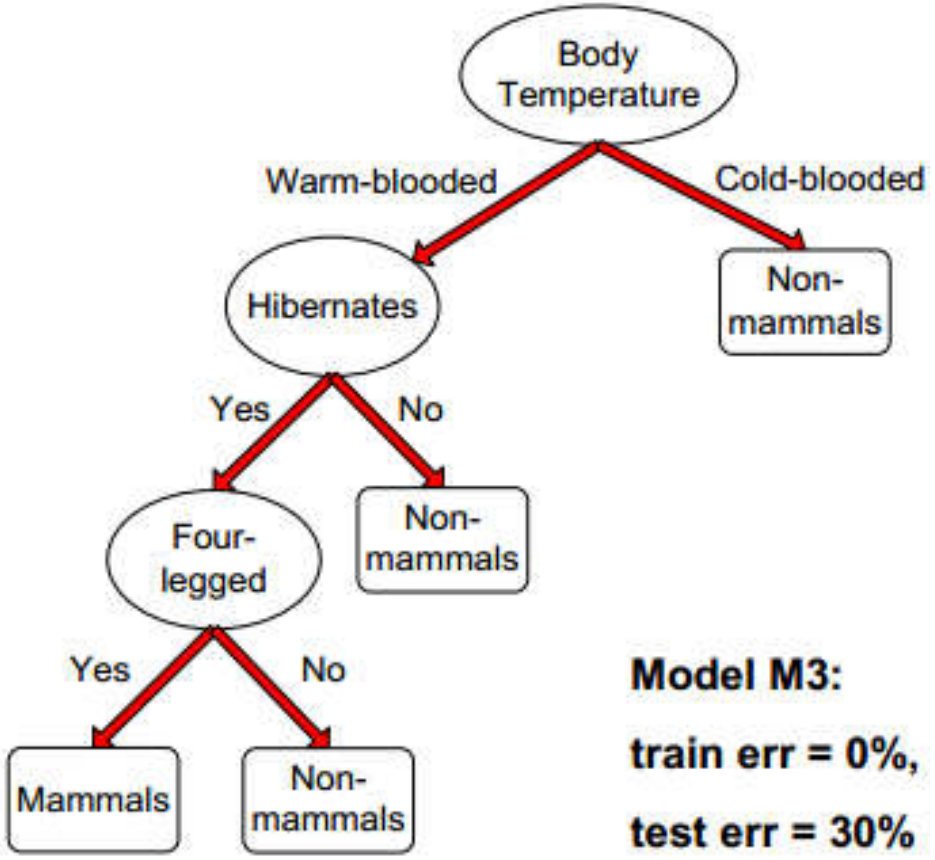
Model M3:

train err = 0%,

缺乏代表性样本导致的过分拟合

Test Set:

名称	体温	胎生	4条腿	冬眠	类标号
人	恒温	是	否	否	否
鸽子	恒温	否	否	否	否
象	恒温	是	是	否	否
豹纹鲨	冷血	是	否	否	否
海龟	冷血	否	是	否	否
企鹅	冷血	否	否	否	否
鳗	冷血	否	否	否	否
海豚	恒温	是	否	否	否
希拉毒蜥	冷血	否	是	是	否



Model M3:

train err = 0%,

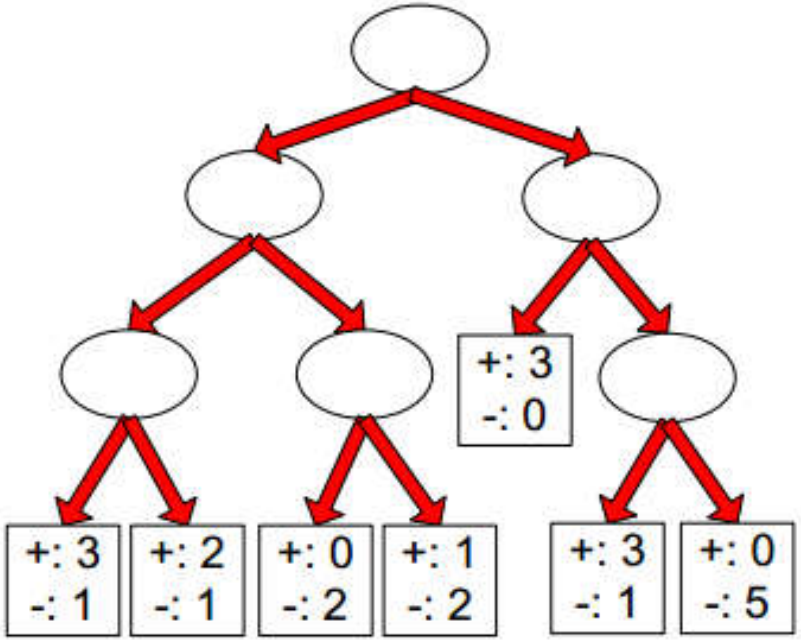
test err = 30%

泛化误差估计

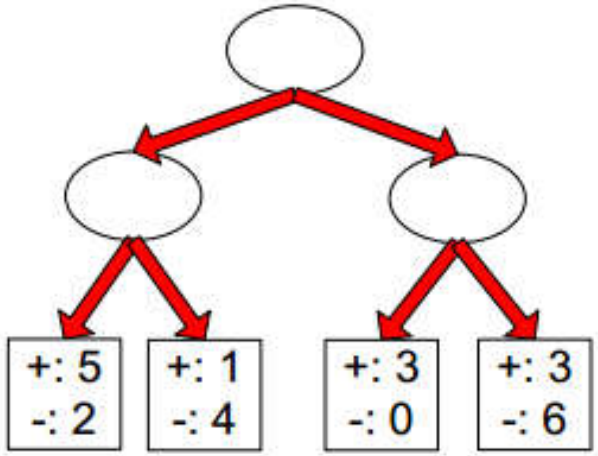
- 使用再代入估计
- 结合模型复杂度
- 使用验证集

使用再代入估计

- Using training error as an optimistic estimate of



Decision Tree, T_L



Decision Tree, T_R

$e(T_L) = 4/24$

$e(T_R) = 6/24$

结合模型复杂度

- Rationale: Occam's Razor
  - Given two models of similar generalization errors, one should prefer the simpler model over the more complex model
  - A complex model has a greater chance of being fitted accidentally by errors in data
  - Therefore, one should include model complexity when evaluating a model

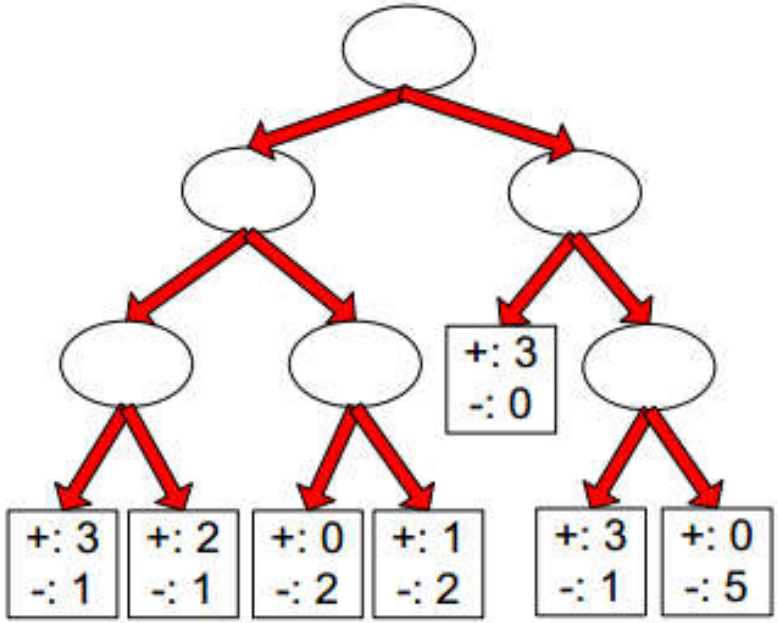
悲观误差估计

- Given a decision tree node t
  - $n(t)$: number of training records classified by t
  - $e(t)$: misclassification error of node t
  - Training error of tree T:

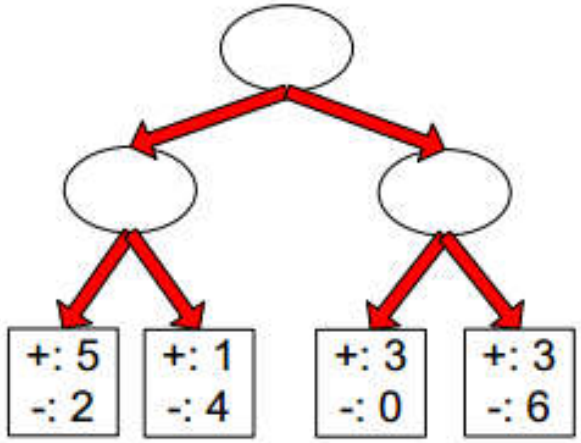
$$e'(T) = \frac{\sum_i [e(t_i) + \Omega(t_i)]}{\sum_i n(t_i)} = \frac{e(T) + \Omega(T)}{N}$$

  - Ω: is the cost of adding a node
  - N: total number of training records

悲观误差估计



Decision Tree, T_L



Decision Tree, T_R

$e(T_L) = 4/24$

$e(T_R) = 6/24$

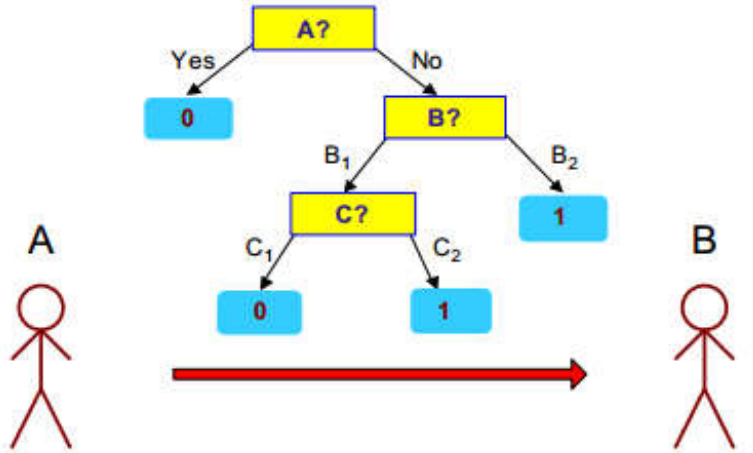
$\Omega = 1$

$e'(T_L) = (4 + 7 \times 1)/24 = 0.458$

$e'(T_R) = (6 + 4 \times 1)/24 = 0.417$

最小描述长度原则

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 is the number of bits needed for encoding.
  - Search for the least costly model.
- Cost(Data|Model) encodes the misclassification errors.
- Cost(Model) uses node encoding (number of children) plus splitting condition encoding.

使用验证集

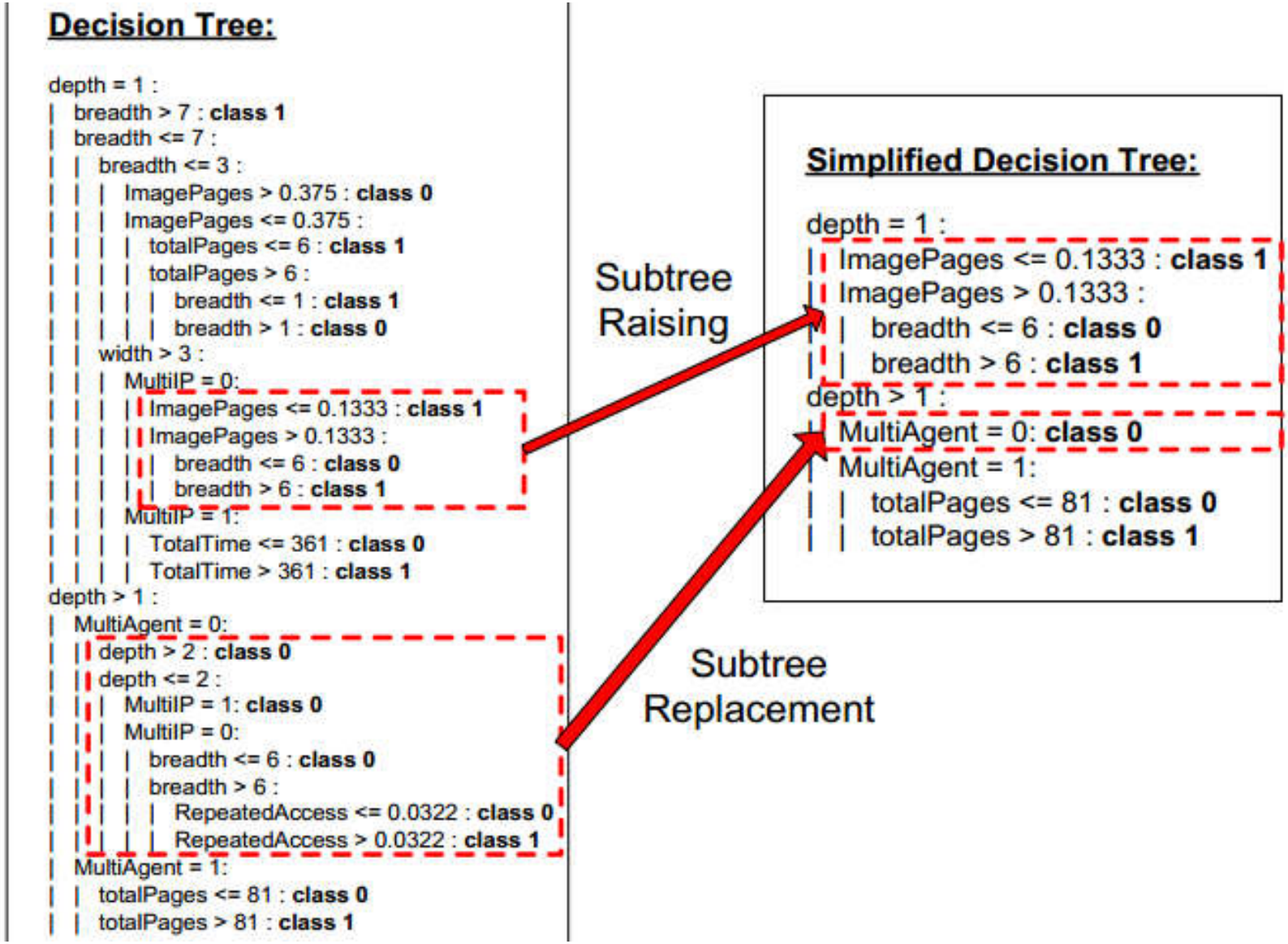
- Divide training data into two parts:
  - Training set:
    - use for model building
  - Validation set:
    - use for estimating generalization error
    - Note: validation set is not the same as test set
- Drawback:
  - Less data available for training

如何处理过分拟合

- Pre-Pruning (Early Stopping Rule)
  - Stop the algorithm before it becomes a fully-grown tree
  - Typical stopping conditions for a node:
    - Stop if all instances belong to the same class
    - Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain).
    - Stop if estimated generalization error falls below certain threshold

如何处理过分拟合

- Post-pruning
  - Grow decision tree to its entirety
  - Subtree replacement
    - If generalization error improves after trimming, replace sub-tree by a leaf node
    - Class label of leaf node is determined from majority class of instances in the sub-tree
  - Subtree raising
    - Replace subtree with most frequently used branch


Decision Tree:
depth = 1 :
| breadth > 7 : class 1
| breadth <= 7 :
| | breadth <= 3 :
| | | ImagePages > 0.375 : class 0
| | | ImagePages <= 0.375 :
| | | | totalPages <= 6 : class 1
| | | | totalPages > 6 :
| | | | | breadth <= 1 : class 1
| | | | | breadth > 1 : class 0
| | width > 3 :
| | | MultiIP = 0:
| | | | ImagePages <= 0.1333 : class 1
| | | | ImagePages > 0.1333 :
| | | | | breadth <= 6 : class 0
| | | | | breadth > 6 : class 1
| | | MultiIP = 1:
| | | | TotalTime <= 361 : class 0
| | | | TotalTime > 361 : class 1
depth > 1 :
| MultiAgent = 0:
| | depth > 2 : class 0
| | depth <= 2 :
| | | MultiIP = 1: class 0
| | | MultiIP = 0:
| | | | breadth <= 6 : class 0
| | | | breadth > 6 :
| | | | | RepeatedAccess <= 0.0322 : class 0
| | | | | RepeatedAccess > 0.0322 : class 1
| MultiAgent = 1:
| | totalPages <= 81 : class 0
| | totalPages > 81 : class 1
Subtree Raising
Subtree Replacement
Simplified Decision Tree:
depth = 1 :
| ImagePages <= 0.1333 : class 1
| ImagePages > 0.1333 :
| | breadth <= 6 : class 0
| | breadth > 6 : class 1
depth > 1 :
| MultiAgent = 0: class 0
| MultiAgent = 1:
| | totalPages <= 81 : class 0
| | totalPages > 81 : class 1

鸢尾花分类



鸢尾花分类

萼片长度	萼片宽度	花瓣长度	花瓣宽度	类别
4.8	3.4	1.9	0.2	0
5	3.2	1.2	0.2	0
6	2.2	5	1.5	2
6.5	3.2	5.1	2	2
5.5	4.2	1.4	0.2	0
6.4	3.2	5.3	2.3	2
6.8	3	5.5	2.1	2
6.1	2.9	4.7	1.4	1
5.7	3	4.2	1.2	1
...	...	...	...	...

模型评价

$$Precision = \frac{TP}{TP + FP} \times 100\%$$

$$Recall = \frac{TP}{TP + FN} \times 100\%$$

		预测值	
		1	0
实际值	1	TP	FN
	0	FP	TN

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{\text{样例总数} + TP - TN}$$

泰坦尼克号生还者预测

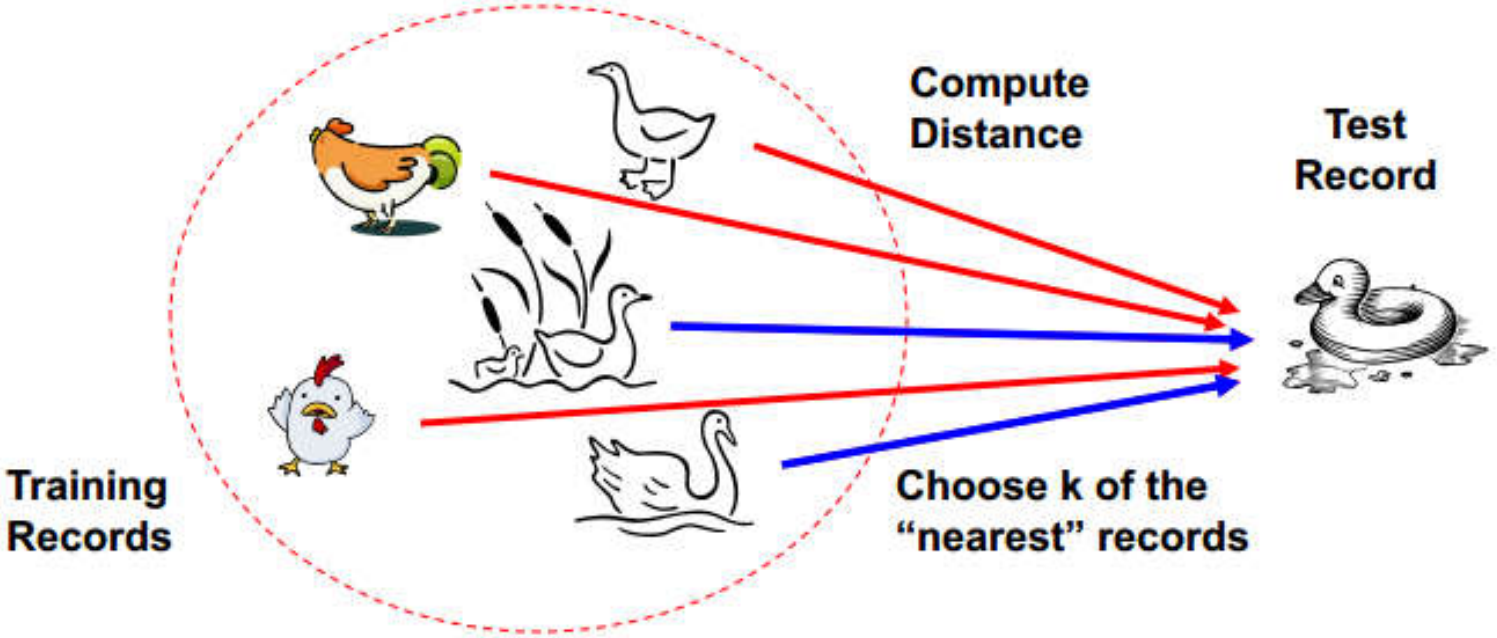
使用scikit-learn建立基于信息熵的决策树模型。

Survived	PassengerId	Pclass	Sex	Age
0	1	3	male	22
1	2	1	female	38
1	3	3	female	26
1	4	1	female	35
0	5	3	male	35
0	6	3	male	

为了说明的方便，数据集有许多属性被删除了。通过观察可知：列Survived是指是否存活，是类别标签，属于预测目标；列Sex的取值是非数值型的。我们在进行数据预处理时应该合理应用Pandas的功能，让数据能够被模型接受。

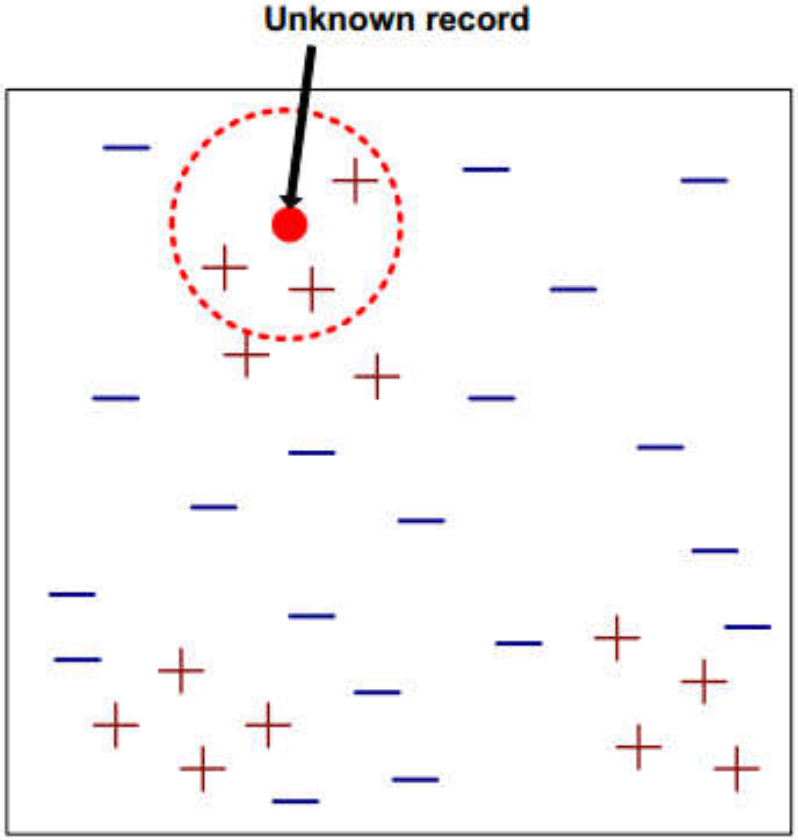
最近邻分类器

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



最近邻分类器

- 把每个样例看作d维（属性个数）空间上的一个数据点
- 计算测试样例与训练集中其他数据点的邻近度
- 样例z的k-最邻近是指和z距离最近的k个数据点
- 根据其近邻的类标号进行分类，如有多个类标号，则取其最近邻的多数类



最近邻分类器-步骤

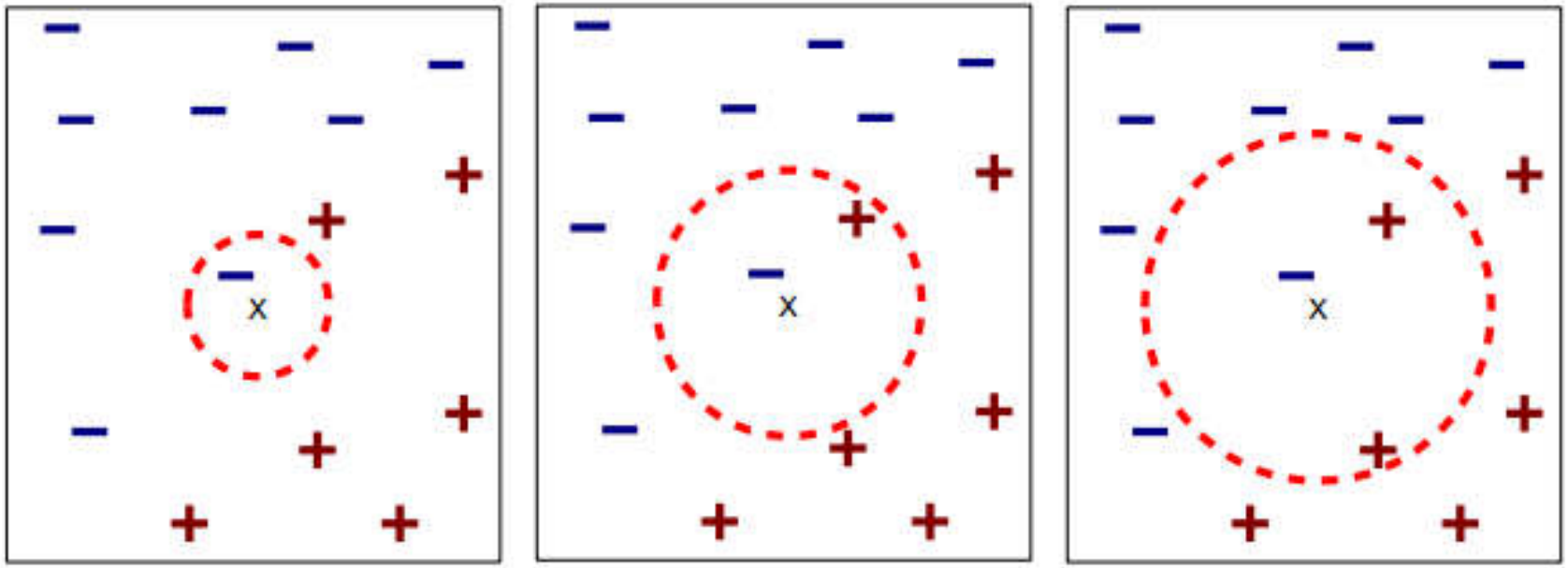
1. 算距离：给定测试对象，计算它与训练集中的每个对象的距离
2. 找邻居：圈定距离最近的k个训练对象，作为测试对象的近邻
3. 做分类：根据这k个近邻归属的主要类别，来对测试对象分类

最近邻分类器-算距离

距离越近应该意味着这两个点属于一个分类的可能性越大。

距离	定义式	说明
绝对值距离	$d_{ij}(1)=\sum_{k=1}^{p}\lvert x_{ik}-x_{jk}\rvert$	绝对值距离是在一维空间下进行的距离计算
欧式距离	$d_{ij}(2)=\sqrt{\sum_{k=1}^{p}(x_{ik}-x_{jk})^2}$.	欧式距离是在二维空间下进行的距离计算
闵可夫斯基距离	$d_{ij}(q)=\left[\sum_{k=1}^{p}(x_{ik}-x_{jk})^q\right]^{1/q}$，$q>0$.	闵可夫斯基距离是在 q 维空间下进行的距离计算
切比雪夫距离	$d_{ij}(\infty)=\max_{1\leqslant k\leqslant p}\lvert x_{ik}-x_{jk}\rvert$.	切比雪夫距离是 q 取正无穷大时的闵可夫斯基距离，即切比雪夫距离是在 $+\infty$ 维空间下进行的距离计算
Lance 距离	$d_{ij}(L)=\sum_{k=1}^{p}\frac{\lvert x_{ik}-x_{jk}\rvert}{x_{ik}+x_{jk}}$	减弱极端值的影响能力
归一化距离	$d_{ij}=\sum_{k=1}^{p}\frac{\lvert x_{ik}-x_{jk}\rvert}{\max(x_k)-\min(x_k)}$	自动消除不同变量间的纲量影响，其中每个变量 k 的距离取值均是 [0,1]

最近邻分类器-找邻居



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

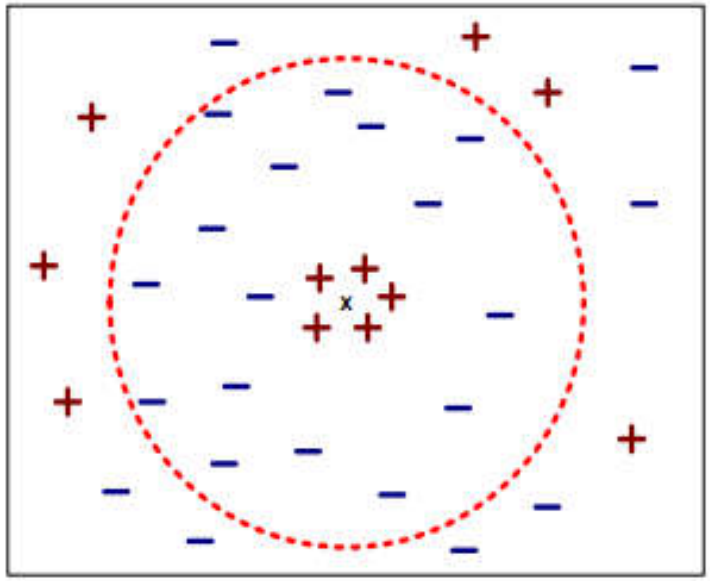
最近邻分类器-做分类

1. 投票决定：少数服从多数；
2. 加权投票法：根据距离的远近，距离越近则权重越大（权重为距离平方的倒数）。

$$d(p,q)=\sqrt{\sum_i (p_i-q_i)^2}$$

Weigh the vote according to distance

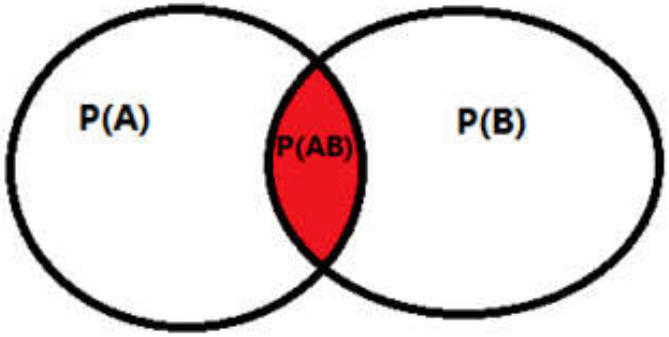
- weight factor, $w = 1/d^2$



最近邻分类器的特征

- 不需要建立模型，但分类测试样例的开销大
- 基于局部信息进行预测，对噪声非常敏感
- 需要选取合适的k值和进行数据预处理
  - Example:
    - height of a person may vary from 1.5m to 1.8m
    - weight of a person may vary from 90lb to 300lb
    - income of a person may vary from $10K to $1M

贝叶斯定理



$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

Likelihood of evidence B if A is true

Prior probability

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Posterior probability of A given the evidence B

Prior probability that evidence B is true

举例

- Probability of Kill
  - P(A): 0.6
  - P(B): 0.5
- What is the probability that it is shot down by A?
  - C: The target is killed.

$$P(A \mid C) = \frac{P(C \mid A)P(A)}{P(C)} = \frac{1 \times 0.6}{0.6 \times 0.5 + 0.4 \times 0.5 + 0.6 \times 0.5} = \frac{3}{4}$$

朴素贝叶斯分类器

X=(Home Owner = No,
Marital Status = Married,
Annual Income = 120K)
比较后验概率P(Yes|X)和P(No|X)

11	No	Married	120K	?

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

朴素贝叶斯分类器

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

假设属性之间条件独立

$$P(X|Y=y) = \prod_{i=1}^{d} P(X_i|Y=y)$$

$$P(Y|X) = \frac{P(Y)\prod_{i=1}^{d} P(X_i|Y)}{P(X)}$$

估计属性的条件概率

- 离散属性
  - 根据类y中属性值等于xi的训练实例的比例来估计条件概率$P(X_i=x_i|Y=y)$
- 连续属性

（1）离散化，用相应的离散区间替换连续属性值，通过计算类y的训练记录中落入X_i对应区间的比例来估计$P(X_i=x_i|Y=y)$

（2）假设连续变量服从某种概率分布，使用训练数据估计分布的参数

高斯分布 $$P(X_i=x_i \mid Y=y_i)=\frac{1}{\sqrt{2\pi}\sigma_{ij}} e - \frac{(x_i-\mu_{ij})^2}{2{\sigma_{ij}}^2}$$

用样本均值$\bar{x}$和样本方差s^2来估计参数μ_{ij}和${\sigma_{ij}}^2$

P(Home Owner = Yes | No)=3/7
P(Home Owner = No | No)=4/7
P(Home Owner =Yes | Yes)=0
P(Home Owner =No | Yes)=1
P(Marital Status = Single | No)=2/7
P(Marital Status = Divorced | No)= 1/7
P(Marital Status = Married | No)=4/7
P(Marital Status = Single | Yes)=2/3
P(Marital Status = Divorced| Yes)=1/3
P(Marital Status = Married| Yes)=0

年收入样本均值和方差
类No：$\bar{x} = 110$，s^2=2975
类Yes：$\bar{x} = 90$，s^2=25

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

11	No	Married	120K	?

P(X | No)=P(Home Owner = No| No)×
P(Marital Status = Married| No)×
P(Annual Income = 120K| No)
=4/7×4/7×0.0072=0.0024

P(X | Yes)=P(Home Owner = No| Yes)×
P(Marital Status = Married| Yes)×
P(Annual Income = 120K| Yes)
=1×0× 1.2 ×10^{-9}=0

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

11	**No**	**Married**	**120K**	?

$$P(No|X) = \frac{P(X|No)P(No)}{P(X)} > P(Yes|X) = \frac{P(X|Yes)P(Yes)}{P(X)}$$

拉普拉斯修正

- 问题：训练集上，很多样本的取值可能并不在其中，但是这不并代表这种情况发生的概率为0
- 解决方法：进行平滑处理，[illegible]修正。

先验概率 P(c)

$$P(c)=\frac{Dc}{D} \quad \xrightarrow{\text{拉普拉斯修正}} \quad P(c)=\frac{Dc+1}{D+N}$$

训练集中第c类样本的数量

样本总数

标签的类别数

类条件概率P(x | c)

$$P(x_i \mid c)=\frac{D_{c,xi}}{Dc} \quad \xrightarrow{\text{拉普拉斯修正}} \quad P(x_i \mid c)=\frac{D_{c,xi}+1}{Dc+Ni}$$

第i个属性上值为xi的数量

第i个属性的类别数

$$P(No)=\frac{7+1}{10+2}=\frac{2}{3} \qquad P(Yes)=\frac{3+1}{10+2}=\frac{1}{3}$$

$$P(\text{Home Owner} = \text{No} \mid \text{No})=\frac{4+1}{7+2}$$

$$P(\text{Marital Status} = \text{Married} \mid \text{No})=\frac{4+1}{7+3}$$

$$P(\text{Home Owner} = \text{No} \mid \text{Yes})=\frac{3+1}{3+2}$$

$$P(\text{Marital Status} = \text{Married} \mid \text{Yes})=\frac{0+1}{3+3}$$

$P(X \mid No)=5/9\times1/2\times0.0072=0.002$

$P(X \mid Yes)=4/5\times1/6\times 1.2\times10^{-9}=1.6\times10^{-10}$

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

11	No	Married	120K	?

$$P(No|X)=\frac{P(X|No)P(No)}{P(X)} \quad > \quad P(Yes|X)=\frac{P(X|Yes)P(Yes)}{P(X)}$$

朴素贝叶斯分类器特点

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
  - How to deal with these dependencies? Bayesian Belief Networks

练习

Class:

C1:buys_computer = 'yes'

C2:buys_computer = 'no'

Data to be classified:

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

age	income	student	credit_rating	com
<=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

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357
- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4
- X = (age <= 30 , income = medium, student = yes, credit_rating = fair)

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")