第二章 示例学习

一. 示例学习的问题描述（见表2.1,表2.2）

二. 决策树学习（ID3算法）

1. ID3算法：

输入：例子集（正例、反例）；

输出：决策树

从树的根结点开始，每次都用“最好的属性”划分结点，直到所有结点只含一类例子为止。

表2.1

例子号	高度	头发	眼睛	类别
1	矮	淡黄	兰	+
2	高	淡黄	兰	+
3	高	红	兰	+
4	高	淡黄	褐	–
5	矮	黑	兰	–
6	高	黑	兰	–
7	高	黑	褐	–
8	矮	淡黄	褐	–

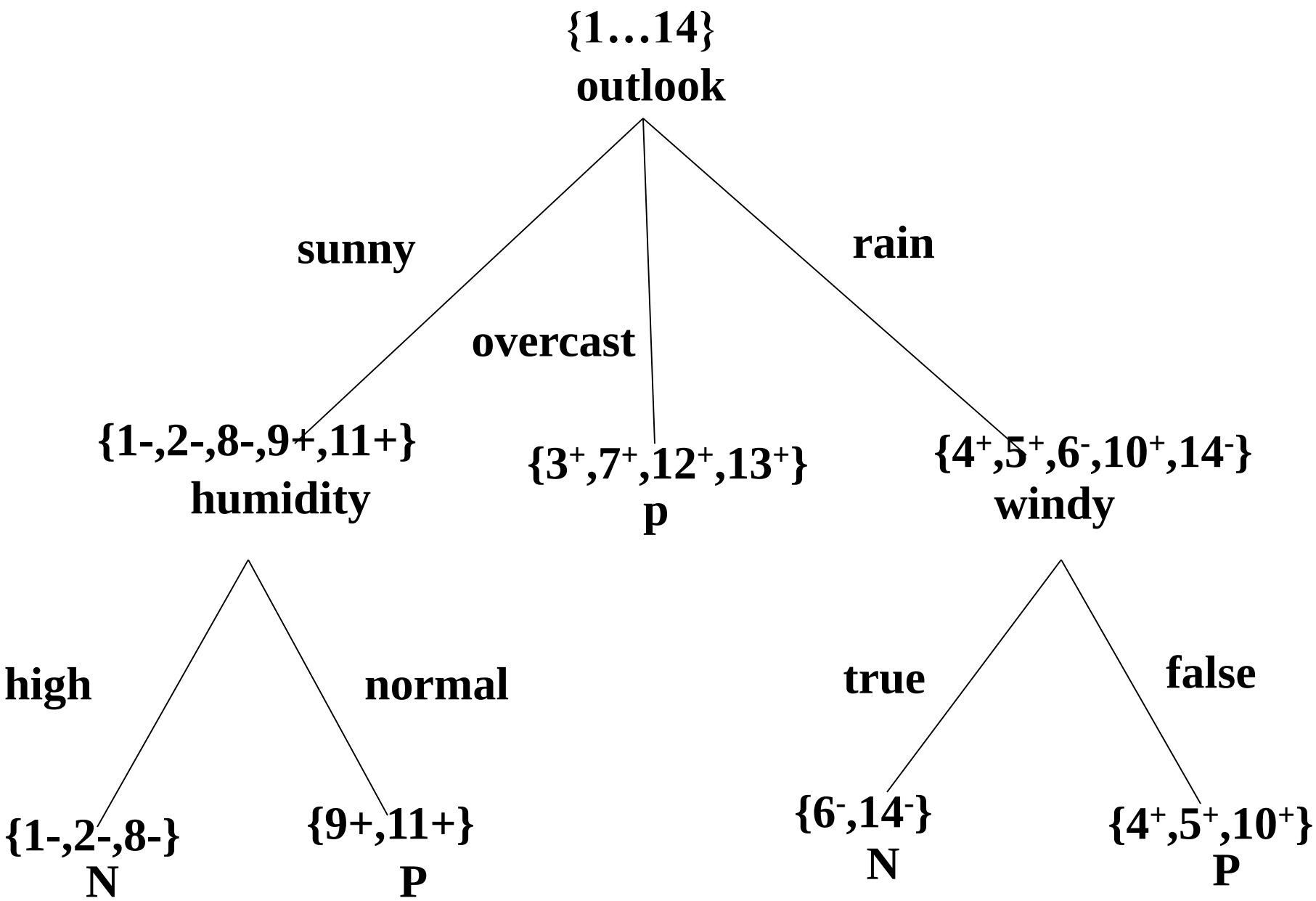
[头发=淡黄∨红色][眼睛=蓝色] → +

[头发=黑色] ∨[眼睛=褐色] → –

表2.2

Day	Outlook	Temperature	Humidity	Wind	Class
1	sunny	hot	High	False	N
2	sunny	hot	High	True	N
3	overcast	hot	High	False	P
4	rain	mild	High	False	P
5	rain	cool	Normal	False	P
6	rain	cool	Normal	True	N
7	overcast	cool	Normal	True	P
8	sunny	mild	High	False	N
9	sunny	cool	normal	false	p

10	**Rain**	**Mild**	**Normal**	**False**	**P**
11	**Sunny**	**Mild**	**Normal**	**True**	**P**
12	**Overcast**	**Mild**	**High**	**True**	**P**
13	**Overcast**	**Hot**	**Normal**	**False**	**P**
14	**rain**	**Mild**	**High**	**True**	**N**


{1…14}
outlook
sunny
overcast
rain
{1-,2-,8-,9+,11+}
humidity
{3+,7+,12+,13+}
p
{4+,5+,6-,10+,14-}
windy
high
normal
true
false
{1-,2-,8-}
N
{9+,11+}
P
{6-,14-}
N
{4+,5+,10+}
P

2. 信息增益

Gain(A)=I(p,n)-E(A)

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

$$E(A) = \sum_{i=1}^{v}\frac{p_i+n_i}{p+n}I(p_i,n_i)$$

其中，p、n是结点node的正、反例个数。A要扩展结点node的属性，p_i、n_i是C 被A划分成的V个子集{C_1, …Cv}的正、反例个数。

属性outlook,有三个值，{sunny,overcast,rain},用outlook扩展根结点得到三个子集{C_1,C_2,C_3}。C_1={1^-,2^-,8^-,9^+,11^+},C_2={3^+,7^+,12^+,13^+}, C_3={4^+,5^+,6^-,10^+,14^-}

根结点:P=9,n=5

$$I(9,5) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940 \text{ bits}$$

P_1=2, n_1=3　　I(2,3)=0.971
P_2=4, n_2=0　　I(4,0)=0
P_3=3, n_3=2　　I(3,2)=0.971

$$E(outlook) = \frac{5}{14}I(p_1,n_1) + \frac{4}{14}I(p_2,n_2) + \frac{5}{14}I(p_3,n_3)$$
$$= 0.694 \text{ bits}$$

Gain(outlook)=0.940-E(outlook)=0.246bits

gain(temperature) = 0.029 bits
gain(humidity) = 0.151 bits
gain(windy) = 0.048 bits

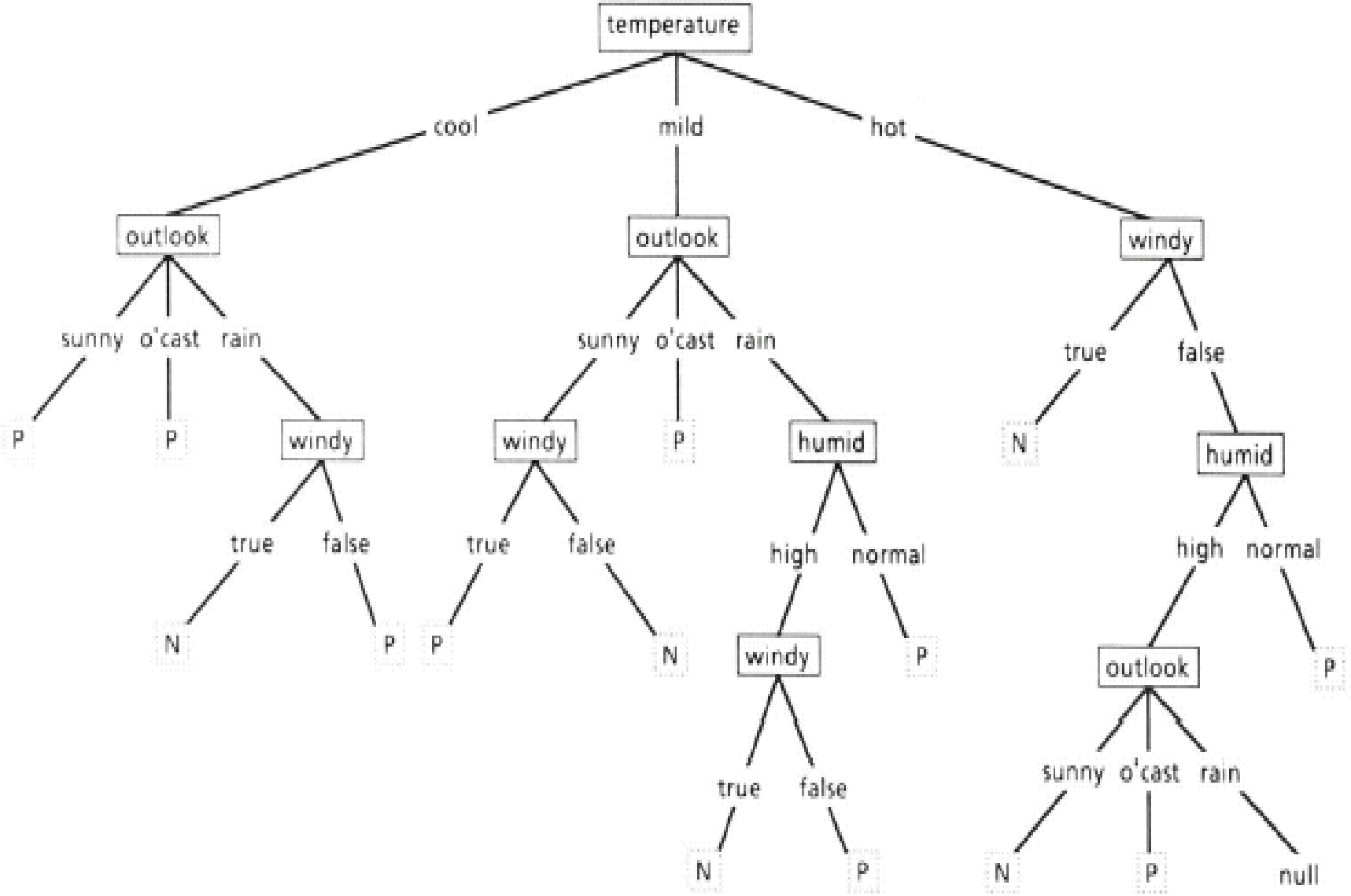


Figure 3. A complex decision tree.

3. 决策树学习的常见问题

1）不相关属性(irrelevant attributes)

属性A有v个属性值，A的第I个属性值对应Pi个正例、n_i个反例。

$$p_i^{'} = p \times \frac{p_i + n_i}{p+n},\ n_i^{'} = n \times \frac{p_i + n_i}{p+n}$$

$$\sum_{i=1}^{v} \frac{(p_i - p_i^{'})^2}{p_i^{'}} + \frac{(n_i - n_i^{'})^2}{n_i^{'}}$$

2) 不充足属性（Inadequate attributes)

两类例子具有相同属性值。没有任何属性可进一步扩展决策树。哪类例子多，叶结点标为哪类。

3）未知属性值

① “最通常值”办法

② 决策树方法: 把未知属性作为“类”，原来的类作为“属性”

Day	Outlook	Temperature	Wind	Class	Humidity
1	sunny	hot	False	N	High
2	sunny	hot	True	N	High
3	overcast	hot	False	P	High
4	rain	mild	False	P	High
5	rain	cool	False	P	Normal
6	rain	cool	True	N	
7	overcast	cool	True	P	Normal
8	sunny	mild	False	N	High
9	sunny	cool	false	p	normal

10	**Rain**	**Mild**	**False**	**P**	**Normal**
11	**Sunny**	**Mild**	**True**	**P**	**Normal**
12	**Overcast**	**Mild**	**True**	**P**	**High**
13	**Overcast**	**Hot**	**False**	**P**	**Normal**
14	**rain**	**Mild**	**True**	**N**	**High**

③ **Bayesian 方法**

$$\text{prob}(A = A_i \mid \text{class} = P) = \frac{\text{prob}(A = A_i \ \&\ \text{class} = P)}{\text{prob}(\text{class} = P)} = \frac{p_i}{p}$$

$$prob(A = Ai \mid class = N) = \frac{n_i}{n}$$

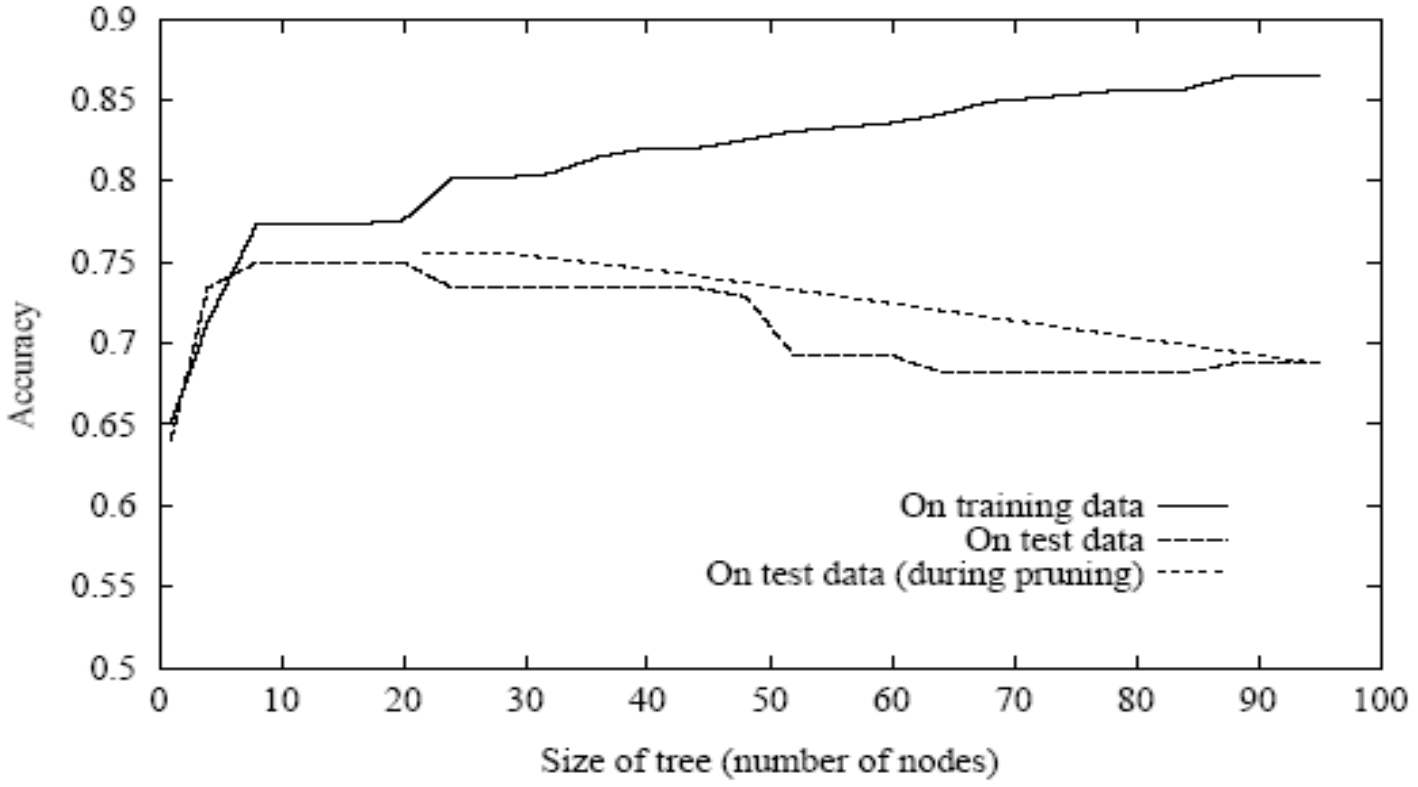
④ 按比例将未知属性值例子分配到各子集中：
属性A有v个值{A_1,…,Av}, A值等于Ai的例子数p_i和n_i，未知属性值例子数分别为p_u和n_u, 在生成决策树时Ai的例子数
$P_i+p_u \cdot ratio_i$ $n_i+n_u \cdot ratio_i$

$$\text{ratio}_i = \frac{p_i + n_i}{\sum_i (p_i + n_i)}$$

4. 属性选择标准

$$IV(A) = -\sum_{i=1}^{v} \frac{p_i + n_i}{p+n} \log_2 \frac{p_i + n_i}{p+n} \qquad \text{gain}(A) \ / \ IV(A)$$

5. Overfitting(过适合)



Reduced-Error Pruning

Split data into *training* and *validation* set

Do until further pruning is harmful:

1. Evaluate impact on *validation* set of pruning each possible node (plus those below it)
2. Greedily remove the one that most improves *validation* set accuracy

- produces smallest version of most accurate subtree
- What if data is limited?

三. 规则学习算法

1. 基本概念：

定义1（例子）. 设$E=D_1 \times D_2 \times \ldots \times Dn$ 是n维有穷向量空间，其中 Dj是有穷离散符号集。E中的元素$e=(V_1,V_2, \ldots,Vn)$简记为<Vj>叫做例子。其中$Vj \in Dj$。

例如：对表2.1

D_1={高，矮}；D_2={淡黄，红，黑}；D_3={兰，褐}

$E=D_1 \times D_2 \times D_3$

例子 e=(矮，淡黄，兰）

定义2.选择子是形为$[x_j=A_j]$的关系语句，其中x_j为第j个属性，$Aj \subseteq Dj$; 公式（或项）是选择子的合取式，即 $\bigwedge_{j \in J}[xj=Aj]$, 其中 $J \subseteq \{1, \ldots,n\}$; 规则是公式的析取式，即 $\bigvee_{i=1}^{l} Li$，其中Li为公式。

一个例子e=<V_1, …Vn>满足选择子（公式、规则）的条件也称做选择子（公式、规则）覆盖该例子。

例如： 例子e=<矮，淡黄，兰> 满足选择子[头发=淡黄∨红色]和 [眼睛=蓝色] ；满足公式[头发=淡黄∨红色] [眼睛=蓝色] 。

定义3：普化(generalize) :减少规则的约束，使其覆盖更多的训练例子叫普化。

定义4：特化(specialize) : 增加规则的约束，使其覆盖训练例子较少叫特化。

定义5：一致：只覆盖正例不覆盖反例的规则被称为是一致的。

定义6：完备：覆盖所有正例的规则被称为是完备的。

2. GS算法：

GS算法

输入： 例子集；

输出： 规则；

原则： （a) 从所有属性值中选出覆盖正例最多的属性值；

(b) 在覆盖正例数相同的情况下，优先选择覆盖反例少的属性值；

设PE,NE是正例，反例的集合。 PE',NE'是临时正，反例集。CPX表示公式，F表示规则（概念描述）。

(1) F←false;

(2) PE' ←PE, NE' ←NE, CPX←true;

(3) 按上述(a) (b)两原则选出一个属性值V_0，设V_0为第j_0个属性的取值，CPX←CPX∧ $[Xj_0=V_0]$

(4) PE' ← CPX覆盖的正例，NE' ← CPX覆盖的反例，如果NE'不为空，转(3);

否则，继续执行(5);

(5) PE←PE\PE', F ←F ∨CPX, 如果PE= ϕ ,停止，否则转(2);

(5) PE←PE\PE', F ←F ∨CPX, 如果PE= ϕ ,停止，否则转(2);

GS算法举例：

例子集见表2.3
学习结果：
[ESR=normal][Ausculation=bublelike] ∨
[X-ray=spot][ESR=normal] $\Longrightarrow$ 肺炎

3.AQ算法：
1) **普化(generalize) :**
2) **特化(specialize) :**
3) **一致**
4) **完备**

表2.3 肺炎与肺结核两组病历

	no	Fever	Cough	X-ray	ESR	Auscultat.
肺炎	1	high	heavy	Flack	Normal	Bubblelike
	2	mediu	heavy	Flack	Normal	Bubblelike
	3	low	slight	Spot	Normal	Dry-peep
	4	high	mediu	Flack	Normal	Bubblelike
	5	mediu	slight	Flack	Normal	Bubblelike
肺结核	1	absent	slight	Strip	Normal	Normal
	2	high	heavy	Hole	Fast	Dry-peep
	3	low	slight	Strip	Normal	Normal
	4	absent	slight	Spot	Fast	Dry-peep
	5	low	mediu	flack	fast	Normal

[ESR=Normal]

	no	Fever	Cough	X-ray	ESR	Auscultat.
肺炎	1	high	heavy	Flack	Normal	Bubblelike
	2	mediu	heavy	Flack	Normal	Bubblelike
	3	low	slight	Spot	Normal	Dry-peep
	4	high	mediu	Flack	Normal	Bubblelike
	5	mediu	slight	Flack	Normal	Bubblelike
肺结核	1	absent	slight	Strip	Normal	Normal
	3	low	slight	Strip	Normal	Normal

[ESR=Normal][Auscultat= Bubblelike]

	no	Fever	Cough	X-ray	ESR	Auscultat.
肺炎	1	high	heavy	Flack	Normal	Bubblelike
	2	mediu	heavy	Flack	Normal	Bubblelike
	3					
	4	high	mediu	Flack	Normal	Bubblelike
	5	mediu	slight	Flack	Normal	Bubblelike
肺结核						

第二轮

	no	Fever	Cough	X-ray	ESR	Auscultat.
肺炎						
	3	low	slight	Spot	Normal	Dry-peep
肺结核	1	absent	slight	Strip	Normal	Normal
	2	high	heavy	Hole	Fast	Dry-peep
	3	low	slight	Strip	Normal	Normal
	4	absent	slight	Spot	Fast	Dry-peep
	5	low	mediu	flack	fasts	Normal

[X-ray= Spot]

	no	Fever	Cough	X-ray	ESR	Auscultat.
肺炎						
	3	low	slight	Spot	Normal	Dry-peep
肺结核						
	4	absent	slight	Spot	Fast	Dry-peep

[X-ray=spot][ESR=normal]

	no	Fever	Cough	X-ray	ESR	Auscultat.
肺炎						
	3	low	slight	Spot	Normal	Dry-peep
肺结核						

3. AQ算法：
输入：例子集、参数#SOL、#CONS、Star的容量m、优化标准；
输出：规则；

1）Pos和NEG分别代表正例和反例的集合
① 从Pos中随机地选择一例子
② 生成例子e相对于反例集NEG的一个约束Star(reduced star), G(e|NEG,m) , 其中元素不多于m个。
③ 在得到的star中，根据设定的优化标准LEF找出一个最优的公式D。
④ 若公式D完全覆盖集合Pos,则转⑥
⑤ 否则，减少Pos的元素使其只包含不被D覆盖的例子。从步骤①开始重复整个过程。
⑥ 生成所有公式D的析取，它是一个完备且一致的概念描述。

2) Star生成: Induce方法
①例子e的各个选择符被放入PS(partial star)中,将ps中的元素按照各种标准排序.
②在ps中保留最优的m个选择符.
③对ps中的选择符进行完备性和一致性检查,从ps中取出完备一致的描述放入SOLUTION表中,若SOLUTION表的大小大于等于参数#SOL,则转⑤.一致但不完备的描述从ps中取出放入表CONSISTENT中,若CONSISTENT表的大小大于等于参数#COS,则转⑤;
④对每个表达式进行特殊化处理,所有得到的表达式根据优化标准排列,仅保留m个最优的.重复步骤③, ④.
⑤得到的一般化描述按优先标准排序,保留m个最优的表达式构成约束Star(e|NEG,m).

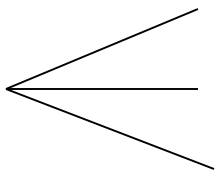
举例:
例子集: 表2.3
#SOL=2

#CONS=2
M=2
优化标准: 正例数/反例数

种子 e_1^+ : [Fever=high][Cough=heavy][X-ray=flack][ESR=normal][Auscultation=bubblelike]

第一轮:
（进入Induce算法）
Ps:
⑤ [Fever=high] <2,1>
④[Cough=heavy] <2,1>
②[X-ray=flack] <4,1>
③[ESR=normal] <5,2>
①[Ausculation=bubblelike] <4,0>
保留m个表达式
[Auscultation=bubblelike] 一致的表达式,放入CONSISTENT中
[X-ray=flack]

特化;

[X-ray=flack] 

[x-ray=flack][ESR=normal] <4,0>

[x-ray=flack][Cough=heavy] <2,0>

[x-ray=flack][Fever=high] <2,0>

保留2个表达式，2个表达式均为一致的,放入CONSISTENT中,按优先标准排序CONSISTENT中表达式,保留m(2)个表达式.

[Ausculation=bubblelike]

[x-ray=flack][ESR=normal]

（出Induce算法）

选出一个最优的作为D

D: [Auscultation=bubblelike]

将D覆盖的正例去掉. 去掉 $e_1^+, e_2^+, e_4^+, e_5^+$ 第一轮结束.

第二轮:

种子 e_3^+ : [Fever=low][Cough=slight][x-ray=spot][ESR=normal]

[Auscultation=dry-peep]

Ps:

④[fever=low] <1,2>

⑤[Cough=slight] <1,3>

①[x-ray=spot] <1,1>

②[ESR=normal] <1,2>

③[Ausculation=dry-peep] <1,2>

保留m(2)个表达式:

[ESR=normal]

[x-ray=spot]

特殊化:

[ESR=normal]

- [ESR=normal] [fever=low] <1,1>
- [ESR=normal] [Cough=slight] <1,2>
- [ESR=normal] [Ausculation=dry-peep] <1,0>

[x-ray=spot]

- [x-ray=spot] [ESR=normal] <1,0>
- [x-ray=spot] [Ausculation=dry-peep] <1,1>
- [x-ray=spot] [fever=low] <1,0>
- [x-ray=spot] [Cough=slight] <1,1>

保留m(2)个表达式

[x-ray=spot] [ESR=normal]

[x-ray=spot] [fever=low]

上面2个表达式都是一致的,放入CONSISTENT表中,按优先标准排序,并选出一个最优的作为D

D: [x-ray=spot] [ESR=normal]

将D覆盖的正例从pos中去掉,去掉 e_3^+ ,pos空.

生成规则:

[Ausculation=bubblelike] $\vee$

[x-ray=spot] [ESR=normal]$\rightarrow$肺炎

算法结束.

4.扩张矩阵：

定义1(扩张矩阵)：已知$e^+=<\mathbf{V}_1^+,\cdots,\mathbf{V}_n^+>$及反例矩阵NE. 对每一$j\in N$, 用“死元素”*对$\mathbf{V}_j^+$在NE中第j列的所有出现做代换，这样得出的矩阵叫做正例e^+在反例NE背景下的扩张矩阵。记为$EM(e^+|NE)$, 或简记为$EM(e^+)$。

表2.7正例矩阵与反例矩阵

k	X1	X2	X3	k	X1	X2	X3
1	0	0	0	1	1	0	1
2	1	2	0	2	0	1	0
3	1	0	0	3	1	1	0
4	0	0	2	4	1	1	2
				5	0	0	1

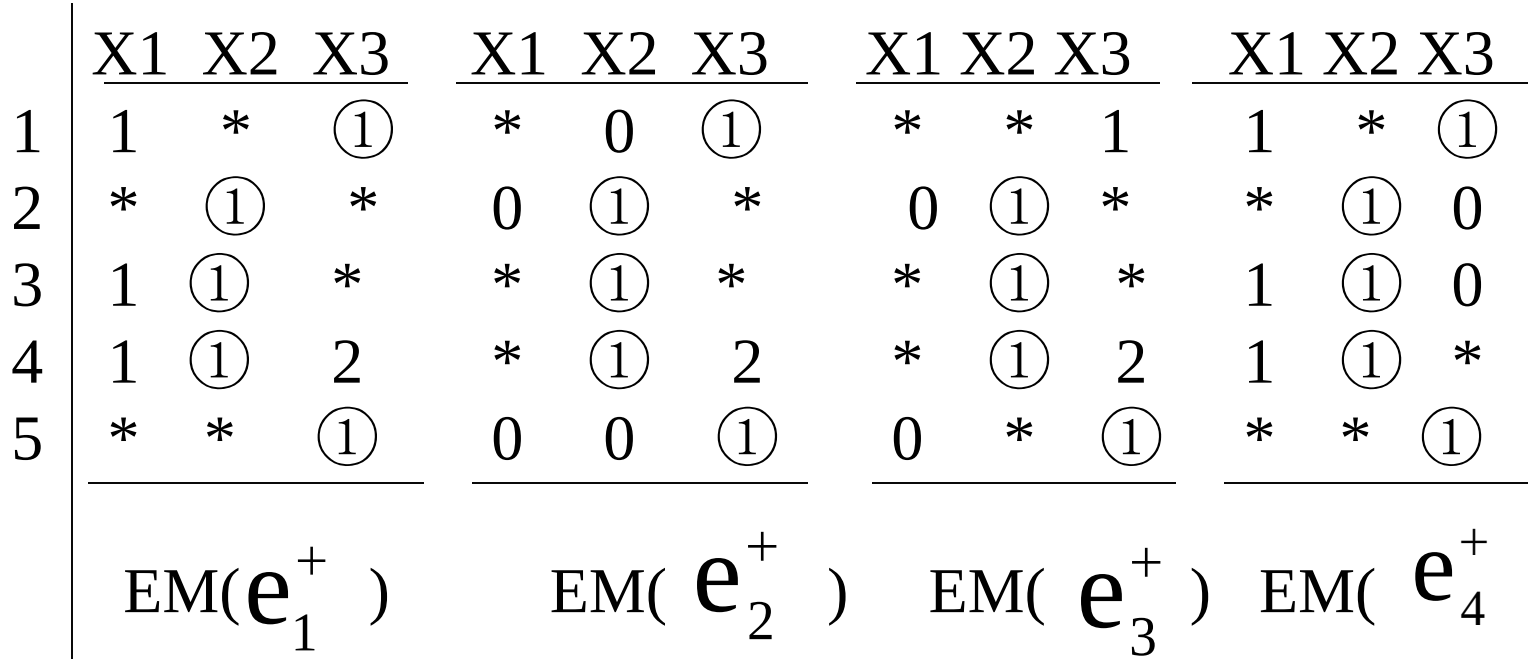


图2.2 正例在反例背景下的扩张矩阵

定义2: 在一个扩张矩阵中，由分别来自不同行的m个非死元素连接组成它的一条**路（径）**；在两个以上的扩张矩阵中，具有相同值的对应的非死元素叫做它们的**公共元素**；在两个或两个以上扩张矩阵中出现的路叫**公共路**；具有公共路的两个扩张矩阵叫做**相交**的，否则叫做不相交的。

5. 算法AE1

优先选择“最大公共元素”，即在最多数目的扩张矩阵中出现的元素。

6. 广义扩张矩阵与AE9算法

① 广义扩张矩阵：已知反例矩阵NE和一个公式L=$\bigwedge_{j\in J}[x_j = A_j]$对NE的每一列j∈N,N={1,2,…,n},如果j∉ J,则用死元素“*”对NE中第j列的所有元素做代换；如果j∈J,则用“*”对NE中第j列属于Aj的所有元素做代换。这样得到的矩阵叫做公式L的广义扩张矩阵。记为EM(L).

②必选元素：设EM(L)是一致公式L的扩张矩阵，如果在EM(L)中的某一行中只有一个非死元素，则该元素叫做必选元素。

③公式的合并：已知公式$L = \bigwedge_{j\in J}[x_j = A_j]$及公式F= $\bigwedge_{j\in J'}[X_j = B_j]$则将L和F对应的选择子的取值合并得到一个新的公式，叫做L和F的合并，记为L⊕F。即L⊕F= $\bigwedge_{j\in J\cap J'}[X_j = A_j \cup B_j]$.

定理2.1 公式L覆盖公式A又覆盖公式B，当且仅当它覆盖A⊕B。

定理2.2 一个例子集合被一个一致的规则所覆盖，则这些例子的合并也是一致的。

	x1	x2	x3	x4
公式L	{0,3}	{0,1}	{1,3}	
1	1	1	1	1
2	0	2	2	2
3	3	1	2	2
4	4	0	3	3

NE

(a) 公式L(L=[X1=0 ∨3][X2=0∨1][X3=1 ∨3]) 与反例矩阵NE

	X1	X2	X3	X4
公式				
	1	*	*	*
	*	2	2	*
	*	*	2	*
	4	*	*	*

L的扩张矩阵EM(L)箭头经过的路对应于公式

$[X_1 \neq 1,4][X_3 \neq 2]=[X_1=\{0,2,3\}][X_3=\{0,1,3\}]$

[算法AE9]

(1) 从正例集PE中选择一个种子e. F←e, path←ϕ ,CPE← ϕ.

(2) 做F的扩张矩阵EM(F)。如果有必选元素则放入path中，同时删去NE中该必选元素出现的行（反例），如果NE空则终止；如非空则删去PE和CPE中出现该必选元素的对应行，重复执行直至EM(F)中不存在必选元素为止。

(3)如果PE非空，检查PE中的每一正例，看它与F的合并是否是一致公式；如不是则从PE中删去该正例；若是则保留一个覆盖正例数目最多的一个合并取代F, 将PE中被F覆盖的正例放入CPE中，重复步骤(2)和(3),直至PE变空。

(4) 如果PE空而CPE非空，则检查CPE中的每一个正例，看它与F合并后是否为一致公式，若不是则从CPE中删去该正例；

若是则生成合并公式，保留一个覆盖最多正例的合并取代F, 从CPE中删去被新的F覆盖的正例，重复(2),(4)直至CPE变空。

(5) 此时PE和CPE均空，但NE非空，做EM(F). 将其中含有最多非死元素的列中的非死元素放入path中，并从NE中删去含这些非死元素的行，重复这一过程，直到NE变空。将path转变为相应的公式。

[应用举例]

将表2.7的第一个反例<1,0,1>改为<1,0,2>

(1) 选择第一个正例e_1^+ =< 0,0,0 > 做种子，F← e_1^+ ,path ←ϕ, CPE ←ϕ.

(2) 做F的扩张矩阵EM(F), 如下图(a)

	X1	X2	X3
公式F	0	0	0
1	1	*	2
2	*	①	*
3	1	1	*
4	1	1	2
5	*	*	①

(a)

	X1	X2	X3
公式F	{0,1}	{0,2}	{0}
1	*	*	②

(b)

path ←path∪ {l_{22},l_{53}},
(3) 将路径转变为公式[x2≠ 1][x3 ≠1]
删去路径对应的反例，
因PE与CPE中没有该必选元素出现，所以PE与CPE不动
(4) PE非空，做F ← F ⊕ (e_1^+)=<{0},{0},{0}>, F是一致的。做F ← F ⊕ (e_2^+)=<{0,1},{0,2},{0}>, F是一致的。继续合并F ← F ⊕ (e_3^+)=<{0,1},{0,2},{0}>, F是一致的。继续合并F ← F ⊕ (e_4^+)=<{0,1},{0,2},{0,2}>, F是不一致的。从PE中删除e_4^+ ,保留覆盖最多正例的F =<{0,1},{0,2},{0}>。CPE={e_1^+ ,e_2^+ ,e_3^+ }做EM(F),如图(b)所示
Path={l_{22},l_{53},l_{13}}, NE空，将Path转变为公式
[x2≠ 1][x3 ≠1, 2]=[x2={0,2}][x3=0]
上面公式覆盖=e_1^+ , e_2^+ , e_3^+ ，但不覆盖 e_4^+
第二轮：
对 e_4^+ 执行上面过程。

参考文献：

1. Induction of Decision Trees, Machine Learning 1: 81-106, 1986.
2. Machine learning: An Artificial Intelligence Approach Edited by R.S. Michalski P39-135.
3. 归纳学习—算法，理论，应用 洪家荣 P.1-33