3.1 Arbori de decizie — Probleme rezolvate

1. **(Arbori de decizie; optimalitate, relativ la numărul de noduri)**

Reprezentaţi arborele/arborii de decizie care are/au numărul minim de noduri posibile şi corespunde/corespund funcţiei boolene $(\neg A \vee B) \wedge \neg(C \wedge A)$ definită peste atributele boolene A, B şi C.

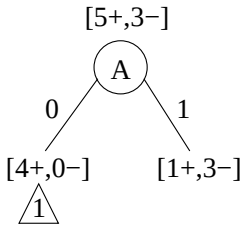
Răspuns:

Vom determina arborele de decizie optimal (ca număr de noduri) parcurgând în mod *exhaustiv spaţiul de versiuni*, adică mulţimea tuturor arborilor de decizie (construiţi cu variabilele A, B şi C) care sunt *consistenţi* cu funcţia dată. Aşadar, vom examina ce se întâmplă când în nodul rădăcină se pun pe rând atributele A, B şi respectiv C.

Notăm cu X funcţia $(\neg A \vee B) \wedge \neg(C \wedge A)$, ale cărei valori sunt date în tabelul alăturat.

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

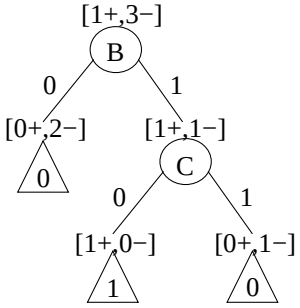
• ***Cazul 1*: Dacă în nodul rădăcină se plasează atributul A, va rezulta următoarea re-partiţionare a mulţimii de exemple pentru conceptul X:**

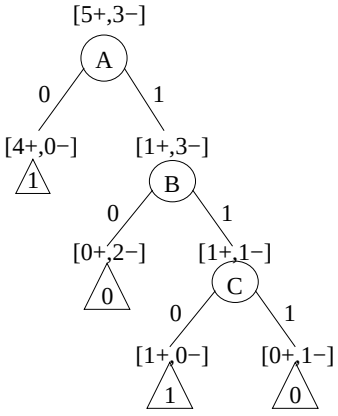


Subarborele drept va trebui să reprezinte arborele de decizie pentru funcţia

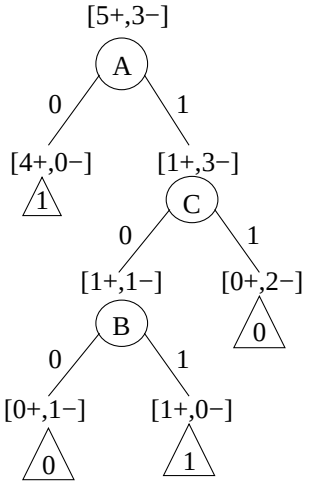
$$X_1 = X[A/1] = (\neg 1 \vee B) \wedge \neg(C \wedge 1) = B \wedge \neg C,$$

pentru care o reprezentare optimă este redată mai jos, în partea stângă:



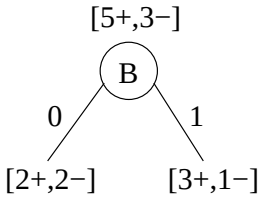


Prin urmare, un arbore optim (ca număr de noduri) care are variabila A în nodul rădăcină este cel reprezentat mai sus, în partea dreaptă.



***Observaţie*: Evident, există încă un arbore optim care are variabila A în nodul rădăcină (el corespunde unei alte reprezentări optimale a conjuncţiei $B \wedge \neg C$ faţă de cea de mai sus). Vedeţi desenul alăturat.**

- ***Cazul 2*: Dacă în nodul rădăcină se alege atributul B,**



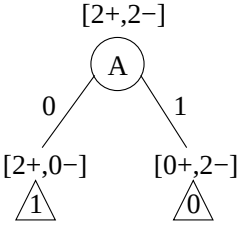
subarborele stâng şi subarborele drept trebuie să reprezinte arborele de decizie pentru funcţiile

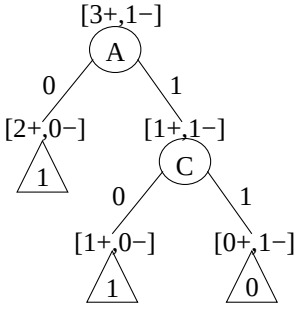
$$X_2 = X[B/0] = (\neg A \vee 0) \wedge \neg(C \wedge A) = \neg A \wedge (\neg C \vee \neg A) = (\neg A \wedge \neg C) \vee \neg A = \neg A,$$

şi respectiv

$$X_3 = X[B/1] = (\neg A \vee 1) \wedge \neg(C \wedge A) = 1 \wedge (\neg C \vee \neg A) = \neg C \vee \neg A,$$

care au ca reprezentări optime arborii de mai jos:

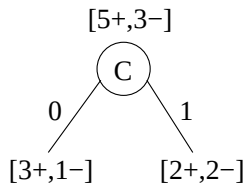




Pentru arborele din dreapta există un arbore de decizie echivalent, obţinut prin interschimbarea lui A cu C.

Prin urmare, orice arbore optim având variabila B în rădăcină are 3 niveluri şi 4 noduri, aşadar cu un nod (de test) mai mult decât cel determinat în primul caz.

- ***Cazul 3*: În sfârşit, când în nodul rădăcină se alege atributul C,**



subarborii drept şi stâng trebuie să reprezinte arborele de decizie pentru funcţiile

$$X_4 = X[C/0] = (\neg A \vee B) \wedge \neg(0 \wedge A) = (\neg A \vee B) \wedge \neg 0 = \neg A \vee B,$$

şi respectiv

$$X_5 = X[C/1] = (\neg A \vee B) \wedge \neg(1 \wedge A) = (\neg A \vee B) \wedge \neg A = \neg A$$

Urmând un raţionament similar cu cel de la cazul anterior, putem spune că orice arbore optim cu atributul C în rădăcină are 3 niveluri şi 4 noduri, cu un nod (de test) mai mult decât cel determinat în primul caz.

Aşadar, putem concluziona că arborii de decizie optimi corespunzători funcţiei date sunt cei determinaţi în primul caz.

2. (Algoritmul ID3: aplicare)

■• ***CMU, 2002 spring, A. Moore, midterm example questions, pr. 2***

Ai naufragiat pe o insulă pustie, unde nu găseşti niciun alt fel de hrană decât ciuperci. Despre unele dintre aceste ciuperci se ştie că sunt otrăvitoare, despre altele se ştie că sunt comestibile, iar despre restul nu se ştie ce fel sunt. Ai rămas singur pe insulă — foştii tăi camarazi, fiind epuizaţi de foame, au folosit metoda 'trial and error'... — şi ai la dispoziţie următoarele date:

Exemplu	*Uşoară*	*Mirositoare*	*ArePete*	*Netedă*	*Comestibilă*
A	1	0	0	0	1
B	1	0	1	0	1
C	0	1	0	1	1
D	0	0	0	1	0
E	1	1	1	0	0
F	1	0	1	1	0
G	1	0	0	1	0
H	0	1	0	0	0
U	0	1	1	1	?
V	1	1	0	1	?
W	1	1	0	0	?

Atunci când nu vei mai avea la dispoziţie pentru a supravieţui decât ciuperci U, V, sau W, ai putea estima care dintre ele sunt comestibile, folosind arbori de decizie.

În primele trei întrebări care urmează, ne vom referi la ciupercile $A - H$:

a. Care este entropia atributului *Comestibilă*?

b. Doar privind datele — adică fără a face explicit calculul câştigului de informaţie (engl., information gain) pentru cele patru atribute — poţi determina ce atribut vei alege ca rădăcină a arborelui de decizie?

c. Calculează câştigul de informaţie pentru atributul pe care l-ai ales la întrebarea precedentă.

d. Elaborează întregul arbore de decizie ID3 bazat pe datele din tabel şi apoi clasifică ciupercile U, V, W.

e. Exprimă cu ajutorul calculului propoziţional (logica predicatelor de ordinul 0) clasificarea produsă de arborele de decizie obţinut. (*Comestibilă* $\leftrightarrow \ldots$)

f. Există vreun risc dacă vei consuma ciuperci care au fost clasificate de arborele de decizie ca fiind comestibile? De ce da? sau, de ce nu?

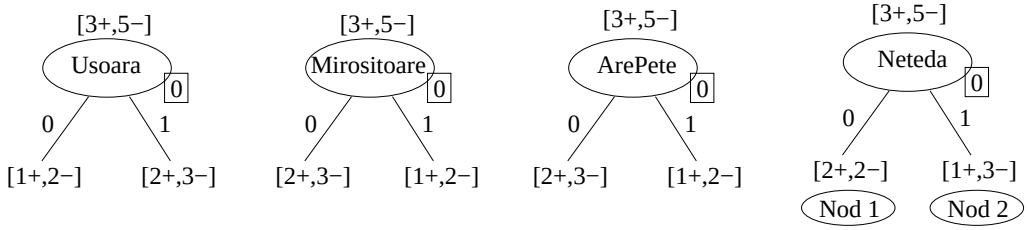
Răspuns:

a. Entropia atributului *Comestibilă* este:

$$\begin{aligned} H_{Comestibilă} &= H[3+,5-] \stackrel{def.}{=} -\frac{3}{8}\log_2\frac{3}{8} - \frac{5}{8}\log_2\frac{5}{8} = \frac{3}{8}\log_2\frac{8}{3} + \frac{5}{8}\log_2\frac{8}{5} = \\ &= \frac{3}{8}3 - \frac{3}{8}\log_2 3 + \frac{5}{8}3 - \frac{5}{8}\log_2 5 = 3 - \frac{3}{8}\log_2 3 - \frac{5}{8}\log_2 5 \approx \\ &\approx 0.9544 \end{aligned}$$

Notaţia $[3+,5-]$ simbolizează o mulţime partiţionată în 3 exemple pozitive şi 5 exemple negative. Vom folosi acest gen de notaţie peste tot în continuare, cu mici variaţii determinate de valorile pe care le poate lua atributul de ieşire. De exemplu, dacă vorbim despre o mulţime cu 5 obiecte roşii, 3 albastre şi 4 verzi, am putea nota: $[5R, 3A, 4V]$.

b. În rădăcina arborelui de decizie se alege atributul care aduce cel mai mare câştig de informaţie. Adică, atributul care, intuitiv vorbind, partiţionează cel mai bine datele de antrenament în raport cu atributul de ieşire. În cazul nostru, variantele pe care le avem la dispoziţie pentru rădăcina arborelui (nodul 0) sunt:



Este uşor de observat că atributele *Uşoară*, *Mirositoare* şi *ArePete* împart mulţimea exemplelor în mod similar: o submulţime cu 3 elemente, dintre care unul este pozitiv iar două sunt negative, şi o submulţime cu 5 elemente, dintre care două sunt pozitive, iar trei sunt negative.

Dacă am considera un arbore de decizie cu un singur nod de test în care plasăm atributul *Netedă*, atunci numărul minim de erori la antrenare pe care îl putem obţine este 3, utilizând următoarea clasificare:

- *Netedă* $= \mathbf{0}$ **:** *Comestibilă* $= \mathbf{1} \Rightarrow$ **ciupercile** E **şi** H **sunt clasificate greşit**
- *Netedă* $= \mathbf{1}$ **:** *Comestibilă* $= \mathbf{0} \Rightarrow$ **ciuperca** C **este clasificată greşit**

Dacă, în schimb, vom pune în rădăcina arborelui de decizie unul dintre celelalte trei atribute, spre exemplu atributul *Uşoară*, şi dacă vom lua votul majoritar în fiecare nod descendent din nodul rădăcină, eroarea rezultată la antrenare va fi aceeaşi ca mai sus (3/8), însă toate instanţele vor fi clasificate la fel (şi anume, negativ). Dacă nu lucrăm cu vot majoritar pentru ambii descendenţi, ci doar pentru cel cu entropie mai mică (în vreme ce pentru celălalt nod descendent luăm decizia contrară), se observă că pentru atributul *Uşoară* vom obţine 4 erori pe setul de antrenament, iar pentru atributul *Netedă* vom obţine 3 erori.

Sumarizând, suntem înclinaţi să credem că ar fi o alegere sensibil mai bună să punem în rădăcină atributul *Netedă*. Pentru o justificare numerică riguroasă a acestei alegeri folosind criteriul maximizării câştigului de informaţie, vezi punctul d.

c. Pentru a obţine câştigul de informaţie pentru atributul *Netedă*, se fac calculele:

$$\begin{aligned} H_{0/Netedă} &\overset{def.}{=} \frac{4}{8}H[2+,2-] + \frac{4}{8}H[1+,3-] = \frac{1}{2}\cdot 1 + \frac{1}{2}\left(\frac{1}{4}\log_2\frac{4}{1} + \frac{3}{4}\log_2\frac{4}{3}\right) \\ &= \frac{1}{2} + \frac{1}{2}\left(\frac{1}{4}\cdot 2 + \frac{3}{4}\cdot 2 - \frac{3}{4}\log_2 3\right) = \frac{1}{2} + \frac{1}{2}\left(2 - \frac{3}{4}\log_2 3\right) \\ &= \frac{1}{2} + 1 - \frac{3}{8}\log_2 3 = \frac{3}{2} - \frac{3}{8}\log_2 3 \approx 0.9056 \end{aligned}$$

$$\begin{aligned} IG_{0/Netedă} &\overset{def.}{=} H_{Comestibilă} - H_{0/Netedă} \\ &= 0.9544 - 0.9056 = 0.0488 \end{aligned}$$

În cele de mai sus am notat cu $H_{0/Netedă}$ entropia *partiţiei* [mulţimii de exemple de antrenament] determinate de alegerea atributului *Netedă* în nodul 0,[56] iar cu $IG_{0/Netedă}$ câştigul de informaţie corespunzător acestei alegeri. În general, prin notaţia $H_{n/A}$ vom înţelege entropia partiţiei determinate de alegerea atributului A în nodul n.

d. Arborele de decizie ID3 se construieşte pornind din rădăcină şi alegând atributul pentru fiecare nod de test în modul următor:

***Nodul 0* (rădăcina):**
Să verificăm dacă alegerea făcută la punctul b este cea corectă:

$$\begin{aligned} H_{0/Uşoară} &\overset{def.}{=} \frac{3}{8}H[1+,2-] + \frac{5}{8}H[2+,3-] \\ &= \frac{3}{8}\left(\frac{1}{3}\log_2\frac{3}{1} + \frac{2}{3}\log_2\frac{3}{2}\right) + \frac{5}{8}\left(\frac{2}{5}\log_2\frac{5}{2} + \frac{3}{5}\log_2\frac{5}{3}\right) \\ &= \frac{3}{8}\left(\frac{1}{3}\log_2 3 + \frac{2}{3}\log_2 3 - \frac{2}{3}\cdot 1\right) + \frac{5}{8}\left(\frac{2}{5}\log_2 5 - \frac{2}{5}\cdot 1 + \frac{3}{5}\log_2 5 - \frac{3}{5}\log_2 3\right) \\ &= \frac{3}{8}\left(\log_2 3 - \frac{2}{3}\right) + \frac{5}{8}\left(\log_2 5 - \frac{3}{5}\log_2 3 - \frac{2}{5}\right) \end{aligned}$$

[56] Mai riguros, folosind terminologia din *Teoria informaţiei*, vom spune că notaţia $H_{0/Netedă}$ se referă la *entropia condiţională medie* a atributului de ieşire *Comestibilă* în raport cu atributul de intrare *Netedă*.

$$= \frac{3}{8}\log_2 3 - \frac{2}{8} + \frac{5}{8}\log_2 5 - \frac{3}{8}\log_2 3 - \frac{2}{8}$$
$$= \frac{5}{8}\log_2 5 - \frac{4}{8} \approx 0.9512$$

Urmează că

$$IG_{0/Uşoară} \stackrel{def.}{=} H_{Comestibilă} - H_{0/Uşoară} = 0.9544 - 0.9512 = 0.0032,$$

deci

$$IG_{0/Uşoară} = IG_{0/Mirositoare} = IG_{0/ArePete} = 0.0032 < IG_{0/Netedă} = 0.0488$$

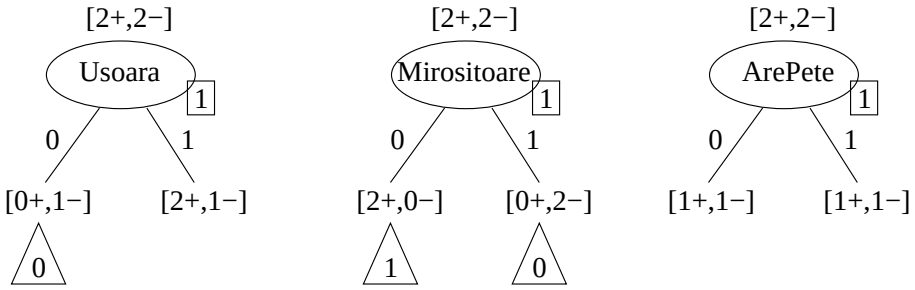
Am avut deci dreptate să alegem atributul *Netedă* **la punctul** b**.**

***Observaţie importantă*: În loc să fi calculat efectiv aceste câştiguri de informaţie, pentru a determina atributul cel mai „bun", ar fi fost suficient să elaborăm un *raţionament* de tip *relaţional*, bazat pe comparaţia dintre valorile entropiilor condiţionale medii $H_{0/Netedă}$ şi $H_{0/Uşoară}$:**

$$IG_{0/Netedă} > IG_{0/Uşoară} \Leftrightarrow H_{0/Netedă} < H_{0/Uşoară}$$
$$\Leftrightarrow \frac{3}{2} - \frac{3}{8}\log_2 3 < \frac{5}{8}\log_2 5 - \frac{1}{2} \Leftrightarrow 12 - 3\log_2 3 < 5\log_2 5 - 4$$
$$\Leftrightarrow 16 < 5\log_2 5 + 3\log_2 3 \Leftrightarrow 16 < 11.6096 + 4.7548 \textbf{ (adev.)}$$

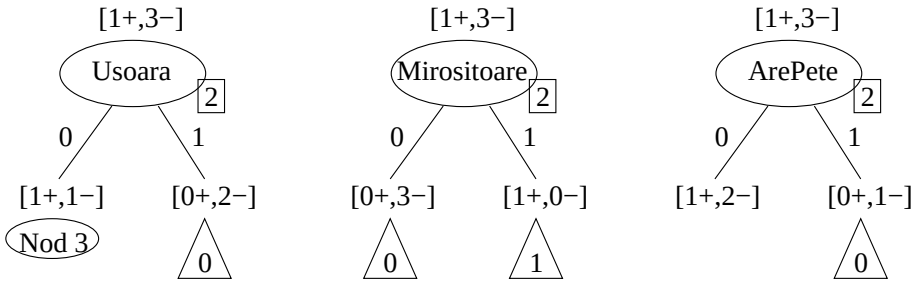
Aşa vom proceda adeseori în rezolvările de la acest capitol.

***Nodul 1*: Trebuie să clasificăm acele exemple care au** *Netedă* $= 0$**; avem de ales între 3 atribute -** *Uşoară*, *Mirositoare* **şi** *ArePete*.



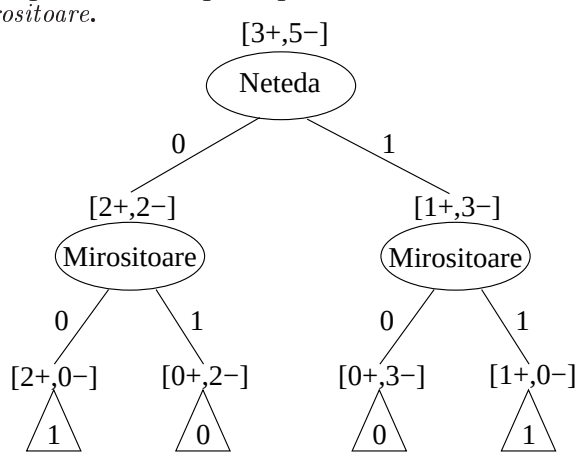
Avem $H_{1/Mirositoare} = \frac{2}{4}H[2+,0-] + \frac{2}{4}H[0+,2-] = 0$**. Oricare ar fi valorile pentru** $H_{1/ArePete}$ **şi** $H_{1/Uşoară}$**, întrucât ştim că entropia are întotdeauna valori nenegative, rezultă că atributul** *Mirositoare* **maximizează în nodul 1 câştigul de informaţie. În imaginea de mai sus valorile din triunghi reprezintă decizia luată de subarborele construit în nodul-frunză respectiv.**

***Nodul 2*: Avem de clasificat exemplele pentru care** *Netedă* $= 1$**. Atributele disponibile sunt:** *Uşoară*, *Mirositoare* **şi** *ArePete*.



Evident, $H_{2/Mirositoare} = \frac{3}{4}H[0+,3-] + \frac{1}{4}H[1+,0-] = \frac{3}{4}\cdot 0 + \frac{1}{4}\cdot 0 = 0$ **Aşadar, pentru nodul 2 putem alege atributul** *Mirositoare*.

Arborele complet arată astfel:



Parcurgând arborele construit, ciupercile U, V **şi** W **vor fi clasificate astfel:**

U	$\boldsymbol{Netedă = 1,\ Mirositoare = 1 \Rightarrow Comestibilă = 1}$
V	$\boldsymbol{Netedă = 1,\ Mirositoare = 1 \Rightarrow Comestibilă = 1}$
W	$\boldsymbol{Netedă = 0,\ Mirositoare = 1 \Rightarrow Comestibilă = 0}$

e. $Comestibilă \leftrightarrow (\neg Netedă \wedge \neg Mirositoare) \vee (Netedă \wedge Mirositoare)$

Acelaşi lucru poate fi exprimat şi sub forma unui pseudo-cod ***if*** $\cdots$ ***then Comestibilă else*** $\neg$ ***Comestibilă*****:**

IF $(\boldsymbol{Netedă = 0}$ **AND** $\boldsymbol{Mirositoare = 0})$ **OR**
$(\boldsymbol{Netedă = 1}$ **AND** $\boldsymbol{Mirositoare = 1})$
THEN ***Comestibilă*;**
ELSE $\neg$***Comestibilă*;**

f. Arborele de decizie produs de către algoritmul ID3 elaborat mai sus este consistent cu datele de antrenament pe care le-am avut la dispoziţie (fiindcă aceste date sunt necontradictorii). Întrucât în realitate clasificarea poate depinde şi de alte trăsături/informaţii decât cele de care dispunem noi, nu avem garanţia că arborele ID3 face identificarea corectă a etichetei/clasei pentru toate instanţele din setul de test. Aşadar, nu putem fi siguri că nu ne vom îmbolnăvi dacă vom consuma ciupercile U **şi** V**, sau că ne vom îmbolnăvi dacă vom consuma ciuperca** W**. În multe aplicaţii practice, calitatea unui model de învăţare automată (în cazul de faţă, un arbore de decizie) se verifică pe un set de *date de validare*.**

3. **(Algoritmul ID3, aplicat pe expresii booleene; exploatarea simetriilor operaţiilor $\vee, \wedge$ în alegerea nodurilor; analiza „optimalităţii“ arborelui ID3, ca număr de noduri de test)**

* ***prelucrare de Liviu Ciortuz după Tom Mitchell, "Machine Learning", 1997, ex. 3.1.b***

Considerăm următoarea funcţie booleană: $A \vee (B \wedge C)$. Presupunem că această funcţie este deja definită — adică valoarea ei este cea cunoscută din logica propoziţiilor —, însă dorim să o reprezentăm ca arbore de decizie.

a. Aplicaţi algoritmul ID3 [tabelei de adevăr corespunzătoare] acestei funcţii.
***Observaţie*: Dacă exploataţi simetriile, veţi avea nevoie doar de puţine calcule, altfel vă veţi complica inutil.**

b. Arborele ID3 obţinut la punctul precedent este optimal?
Alfel spus, puteţi găsi un alt arbore de decizie, de adâncime mai mică sau cu număr mai mic de noduri (comparativ cu arborele obţinut la punctul a), care să reprezinte această funcţie? (Ţineţi cont că în fiecare nod al unui arbore de decizie se poate testa un sigur atribut.)

Răspuns:

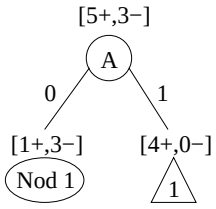
A	B	C	$Y = A \vee (B \wedge C)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

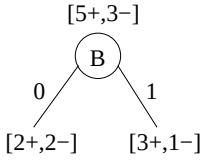
a. Observăm că funcţia dată este simetrică în B şi C, datorită comutativităţii funcţiei logice $\wedge$. O consecinţă a acestui fapt este că dacă, pe parcursul algoritmului ID3, avem de ales (şi) între cele două atribute este nevoie să-l studiem doar pe unul dintre ele, celălalt comportându-se identic.

***Nodul 0* (rădăcină):**

$$\begin{aligned} H_{0/A} &= \frac{4}{8}H[1+,3-] + \frac{4}{8}H[4+,0-] = \\ &= \frac{1}{2}H[1+,3-] + \frac{1}{2}\cdot 0 = \\ &= \frac{1}{2}H[1+,3-] \end{aligned}$$

$$\begin{aligned} H_{0/B} &= \frac{4}{8}H[2+,2-] + \frac{4}{8}H[3+,1-] = \\ &= \frac{1}{2}\cdot 1 + \frac{1}{2}H[3+,1-] = \\ &= \frac{1}{2} + \frac{1}{2}H[1+,3-] \end{aligned}$$



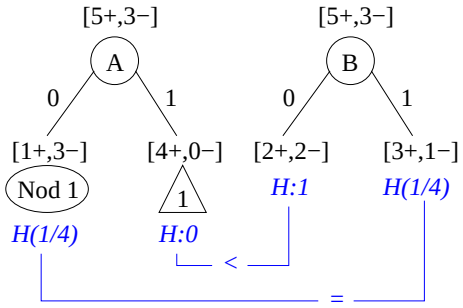


Este evident că $H_{0/A} < H_{0/B}$, deci vom alege atributul A în rădăcină.

***Observaţie importantă*:**

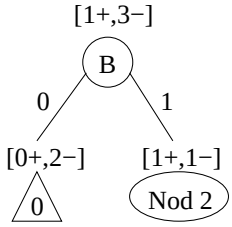
La aceeaşi concluzie se putea ajunge *imediat* pe baza unui *raţionament calitativ*, şi anume, comparând atent cei doi arbori („compaşi de de decizie“) de mai

sus. Mai precis, vom compara (două câte două) entropiile condiţionale specifice din nodurile descendente, precum şi ponderile cu care se combină aceste entropii în scrierea entropiilor condiţionale medii corespunzătoare atributelor A şi B. Putem pune în evidenţă acest fapt în figura următoare,



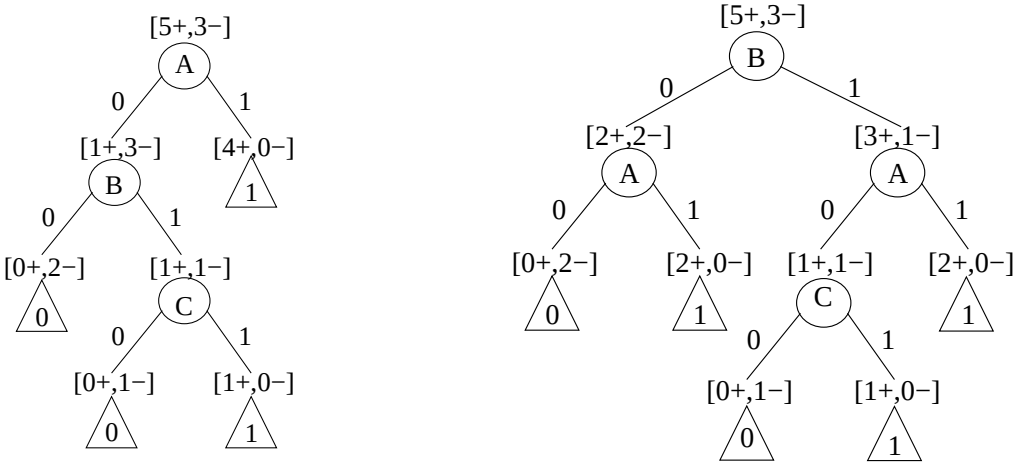
în care simbolul H, scris uneori însoţit de un argument (aşadar, ca $H(p)$), se referă la entropia unei variabile Bernoulli de parametru p. Mai facem *precizarea* că semnele $<$ şi $=$ din figura de mai sus se referă de fapt nu [doar] la entropiile condiţionale specifice, ci [şi] la produsul acestora cu ponderile asociate în mod corespunzător: $\frac{4}{8}H[1+,3-] = \frac{4}{8}H[3+,1-]$ şi respectiv $\frac{4}{8}H[4+,0-] < \frac{4}{8}H[2+,2-]$.

***Nodul 1*: Avem de clasificat indivizii care au $A = 0$ şi putem alege între atributele B şi C. Datorită simetriei, îl putem alege pe oricare dintre ele. Pentru fixare îl alegem pe B.**



***Nodul 2*: La acest punct a mai rămas disponibil doar atributul C.**

Arborele construit de ID3 este cel reprezentat mai jos, în partea stângă:



b. Pentru a vedea dacă arborele construit de algoritmul ID3 este cel optimal, trebuie să reconsiderăm toate deciziile pe care le-am luat în construirea acestuia:

– La nodul 1 al arborelui avem de clasificat exemplele pentru care $A = 0$, deci funcţia care trebuie reprezentată de subarborele în cauză este $f' = f[A/0] = 0 \vee (B \wedge C) = B \wedge C$, funcţie care este reprezentată în mod optimal de subarborele construit de ID3. (Notaţia $A/0$ semnifică faptul că variabila logică A este instanţiată la valoarea 0.) Prin urmare, nu există un arbore mai bun care să reprezinte funcţia dată şi să aibă în rădăcină atributul A.

– În rădăcină am ales atributul A în detrimentul celorlalte două atribute deoarece am demonstrat că aduce cel mai mare câştig de informaţie. Să vedem ce se întâmplă dacă alegem unul dintre atributele B sau C. După cum am discutat mai sus, datorită simetriei, pe oricare dintre cele două l-am alege, arborele rezultat ar avea aceeaşi formă. Pentru fixare, să-l alegem pe B.

Subarborele stâng şi drept vor trebui să reprezinte funcţiile:

$$f'' = f[B/0] = A \vee (0 \wedge C) = A \vee 0 = A$$

şi respectiv

$$f''' = f[B/1] = A \vee (1 \wedge C) = A \vee C$$

Arborele minimal care poate fi construit în aceste circumstanţe este cel reprezentat mai sus în partea dreaptă. Acest arbore are 3 niveluri şi 4 noduri, cu un nod în plus faţă de cel construit de algoritmul ID3.

Putem deci conchide că arborele construit respectând specificaţiile algoritmului ID3 este cel optimal.

***Observaţie*:**

Această problemă pune în evidenţă două modalităţi de parcurgere a spaţiului de versiuni pentru un concept, în particular unul din logica propoziţiilor, care este reprezentat cu ajutorul arborilor de decizie. Pe de o parte avem explorarea (incompletă) făcută de algoritmul ID3 care este de tip "greedy", iar pe de altă parte avem explorarea exhaustivă. Prima strategie de explorare procedează la o căutare „orientată" a soluţiei (şi din această cauză este mai eficientă, dar se va vedea, ca revers, că nu asigură întotdeauna găsirea optimului), iar cea de-a doua strategie de explorare, deşi asigură găsirea optimului, nu este utilizabilă în cazurile (frecvente!) în care spaţiul de versiuni este foarte mare.

4. (ID3, ca algoritm "greedy"; un exemplu când arborele ID3 nu este optimal ca număr de noduri şi de niveluri)

prelucrare de Liviu Ciortuz după
CMU, 2003 fall, T. Mitchell, A. Moore, midterm exam, pr. 9.a

Fie atributele binare de intrare A, B, C, atributul de ieşire Y şi următoarele exemple de antrenament:

A	B	C	Y
1	1	0	0
1	0	1	1
0	1	1	1
0	0	1	0

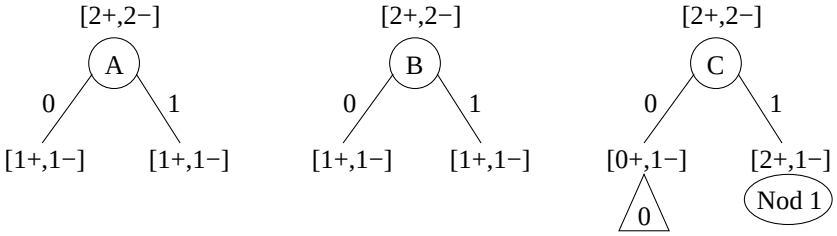
a. Determinaţi arborele de decizie calculat de algoritmul ID3. Este acest arbore de decizie consistent cu datele de antrenament?

b. Există un arbore de decizie de adâncime mai mică (decât cea a arborelui ID3) consistent cu datele de mai sus? Dacă da, ce concept (logic) reprezintă acest arbore?

Răspuns:

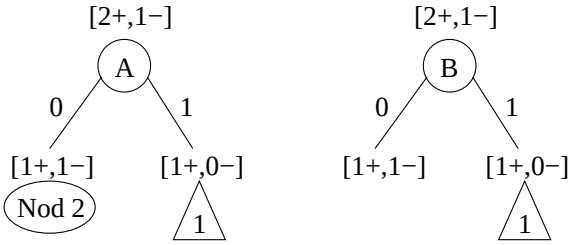
a. Se construieşte arborele de decizie cu algoritmul ID3 astfel:

***Nodul 0* (rădăcina):**



Observăm că $H_{0/A} = H_{0/B} = \frac{2}{4}H[1+,1-] + \frac{2}{4}H[1+,1-] = H[1+,1-] = 1$**, care este valoarea maximă a entropiei [condiţionale medii a] unei variabile boolene. Prin urmare,** $H_{0/C}$ **nu poate fi decât mai mică sau egală cu** $H_{0/A}$ **şi** $H_{0/B}$**. Deci vom alege în nodul rădăcină atributul** C**.**

***Nodul 1*: Avem de clasificat instanţele cu** $C = 1$**, deci alegerea se face între atributele** A **şi** B**.**

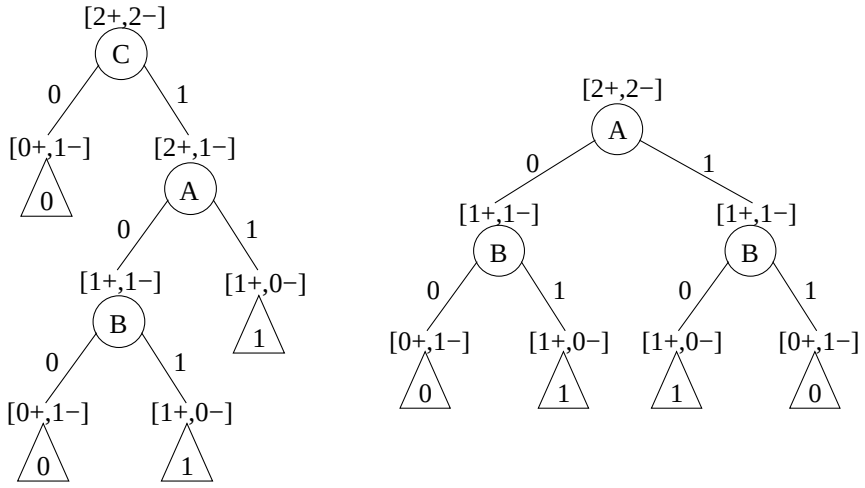


Cele două entropii condiţionale medii sunt egale:

$$H_{1/A} = H_{1/B} = \frac{2}{3}H[1+,1-] + \frac{1}{3}H[1+,0-]$$

Aşadar, putem alege oricare dintre cele două atribute. Pentru fixare, îl alegem pe A**.**

***Nodul 2*: La acest nod nu mai avem decât atributul** B**, deci îl vom pune pe acesta. Arborele complet este reprezentat în partea stângă:**



Prin construcţie, algoritmul ID3 este consistent cu datele de antrenament dacă acestea sunt consistente (i.e., necontradictorii). În cazul nostru, se verifică imediat că datele de antrenament sunt consistente.

b. Se observă că atributul de ieşire Y reprezintă de fapt funcţia logică A XOR B. Reprezentând această funcţie ca arbore de decizie, vom obţine arborele desenat mai sus în partea dreaptă. Acest arbore are cu un nivel mai puţin decât arborele construit cu algoritmul ID3. Prin urmare, arborele ID3 nu este optim din punctul de vedere al numărului de niveluri.

5. **(Algoritmul ID3: aplicare pe date inconsistente, "decision stumps", calculul acurateţii)**

• ○ ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, HW1, pr. 2.ab***

Tabelul de mai jos sumarizează situaţia celor 2201 de pasageri şi membri ai echipajului de la bordul vasului Titanic, în urma naufragiului din data de 15 Aprilie 1912. Pentru fiecare combinaţie de valori ale celor 3 variabile (Clasă, Sex, Vârstă) am indicat în tabel câţi oameni au supravieţuit şi câţi nu au supravieţuit. (*Observaţie*: Datele originale au patru valori pentru atributul Clasă; am comasat valorile II, III, şi Echipaj într-o singură valoare, denumită „Inferioară".)

			Supravieţuitori		
Clasa	Sexul	Vârsta	Nu	Da	Total
I	Masculin	Copil	0	5	5
I	Masculin	Adult	118	57	175
I	Feminin	Copil	0	1	1
I	Feminin	Adult	4	140	144
Inferioară	Masculin	Copil	35	24	59
Inferioară	Masculin	Adult	1211	281	1492
Inferioară	Feminin	Copil	17	27	44
Inferioară	Feminin	Adult	105	176	281
		Total	1490	711	2201

Pentru a vă uşura calculele pe care va trebui să le faceţi, am făcut noi totalurile pentru fiecare variabilă:

	Supravieţuitori		
Clasa	Nu	Da	Total
I	122	203	325
Inferioară	1368	508	1876

	Supravieţuitori		
Sexul	Nu	Da	Total
Masculin	1364	367	1731
Feminin	126	344	470

	Supravieţuitori		
Vârsta	Nu	Da	Total
Copil	52	57	109
Adult	1438	654	2092

a. Folosind un arbore de decizie, dorim să prezicem variabila de ieşire Y (Supravieţuitor), pornind de la atributele de intrare C (Clasa), S (Sexul), V (Vârsta). Utilizaţi criteriul câştigului de informaţie pentru a alege care dintre aceste trei atribute C, S sau V trebuie să fie folosit în nodul-rădăcină al arborelui de decizie.

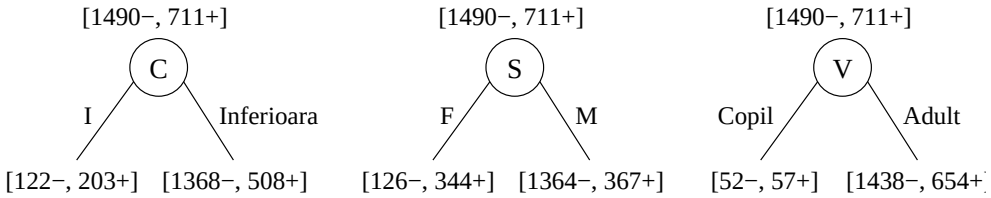
De fapt, ce vi se cere este să învăţaţi un arbore de decizie de adâncime 1 care foloseşte doar atributul din rădăcină pentru a clasifica datele. (Astfel de arbori de decizie de adâncime 1 sunt adesea numiţi în terminologia de limbă engleză "decision stumps".) Parcurgeţi toate etapele rezolvării, redând inclusiv calculele pentru câştigul de informaţie al fiecărui atribut.

b. Care este acurateţea [medie] obţinută pe datele de antrenament de către arborele de decizie cu adâncime 1 de la punctul precedent?

c. Dacă aţi crea un arbore de decizie care foloseşte toate cele trei variabile, care ar fi acurateţea lui [medie] pe datele de antrenament? (*Observaţie*: Nu trebuie neapărat să creaţi arborele de decizie pentru a afla răspunsul!)

Răspuns:

a. Totalurile care au fost furnizate în enunţ pentru fiecare dintre variabilele C, S şi V ne servesc foarte bine pentru a crea rapid cei trei "decision stumps":



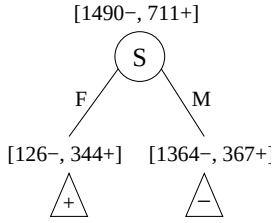
Analizând datele conform figurii de mai sus, se poate „intui" că atributul S va avea un câştig de informaţie (în raport cu atributul de ieşire Y − *Supravieţuitor*) mai bun decât al celorlalte două atribute de intrare (C şi V). Intuiţia se verifică făcând calculele:

$$\begin{aligned} IG(Y,C) &= H[1490-,711+] - \left(\frac{325}{2201}H[122-,203+] + \frac{1876}{2201}H[1368-,508+]\right) \\ &= 0.048501 \\ IG(Y,S) &= H[1490-,711+] - \left(\frac{470}{2201}H[126-,344+] + \frac{1731}{2201}H[1364-,367+]\right) \\ &= 0.142391 \end{aligned}$$

$$IG(Y,V) = H[1490-,711+] - \left(\frac{109}{2201}H[52-,57+] + \frac{2092}{2201}H[1438-,654+]\right)$$
$$= 0.006411.$$

Deci, într-adevăr, câştigul maxim de informaţie se obţine pentru atributul S.

b. Arborele de decizie de adâncime 1 care are în nodul rădăcină atributul S este cel din figura alăturată. Acurateţea [medie a] acestui arbore de decizie este:



$$\frac{470}{2201}\cdot\frac{344}{470} + \frac{1731}{2201}\cdot\frac{1364}{1731} = \frac{344+1364}{2201} = \frac{1708}{2201} = 0.776.$$

c. Se poate constata imediat că arborele ID3 produs pe datele din această problemă va avea 8 noduri-frunză, iar în fiecare dintre aceste noduri-frunză se va asigna câte una dintre mulţimile descrise (pe linie) în coloanele 4 şi 5 ale tabelului principal din enunţ: $[0-,5+]$, $[118-,57+]$, $\ldots$, $[17-,27+]$, $[105-,176+]$. Decizia care va fi luată în fiecare nod-frunză este dictată de votul majoritar, şi anume: $+, -, \ldots, +$ şi respectiv $+$.

Putem calcula acurateţea [medie] astfel:

$$\frac{5+118+1+140+35+1211+27+176}{2201} = \frac{1713}{2201} = 0.778.$$

Se observă că se produce (din păcate) o creştere foarte mică în raport cu acurateţea celui mai bun "decision stump": doar 0.002.

6. **(Algoritmul ID3: cazul când există repetiţii şi inconsistenţe în datele de antrenament; o margine superioară pentru eroarea la antrenare în funcţie de numărul de valori ale variabilei de ieşire)**

CMU, 2002 fall, Andrew Moore, midterm exam, pr. 1.fg

A	B	C	Z
0	0	0	0
0	0	1	0
0	0	1	0
0	1	0	0
0	1	1	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	0	1
1	1	1	0
1	1	1	1

Presupunem că învăţăm un arbore de decizie care să prezică atributul de ieşire Z pornind de la atributele de intrare A, B, C. Se folosesc datele de antrenament din tabelul alăturat.

a. Care va fi eroarea la antrenare pe acest set de date? Exprimaţi răspunsul sub forma fracţiei de înregistrări care vor fi clasificate eronat ($n/12$).

b. Considerăm un arbore de decizie construit pe un set arbitrar de date. Dacă atributul de ieşire este cu valori discrete şi poate lua k valori distincte, care este eroarea de antrenare maximă (exprimată ca fracţie)?

Răspuns:

a. Este uşor de observat că datele de antrenament conţin „inconsistenţe" (contradicţii, relativ la etichetare), şi anume la exemplele $(0,1,1)$ şi $(1,1,1)$.

Fiecare dintre aceste exemple sunt etichetate o dată cu 0 şi altă dată cu 1. Prin urmare, jumătate din aceste exemple vor fi clasificate eronat de arborele învăţat de către algoritmul ID3. Eroarea la antrenare va fi deci $\frac{2}{12}$.

b. Vom analiza pe rând mai multe cazuri, care sunt din ce în ce mai generale.

Cazul *i*: Mai întâi vom calcula eroarea la antrenare pentru cazul în care setul de date de antrenament este compus din k instanţe care sunt identice ca tupluri de valori pentru atributele ce le caracterizează, dar sunt clasificate pe rând cu fiecare din cele k valori posibile ale atributului de ieşire. Este evident că arborele de decizie învăţat va clasifica eronat $k-1$ instanţe. Aşadar, în acest caz, eroarea la antrenare va fi

$$E = \frac{k-1}{k}$$

Cazul *ii*: Aceeaşi eroare [la antrenare] ca mai sus se va înregistra dacă în locul fiecărei instanţe dintre cele considerate la cazul precedent vom avea l instanţe identice, inclusiv în ce priveşte eticheta. (În total sunt kl instanţe de antrenament.)

$$E = \frac{(k-1)\cdot l}{k\cdot l} = \frac{k-1}{k}$$

Cazul *iii*: Dacă relaxăm condiţia de mai sus considerând $l_1, l_2, \ldots, l_k$ instanţe identice, iar $l = \max_{i=1}^{k} l_i$, este imediat că eroarea maximă se va atinge în cazul $l_1 = l_2 = \ldots = l_k = l$, şi va avea aceeaşi valoare ca mai sus.

Aşadar, în continuare vom putea renunţa la a considera factorul de multiplicare l, fără ca prin aceasta să restrângem generalitatea raţionamentului.

Cazul *iv*: Fie n exemple de antrenament (instanţe etichetate) dintre care d sunt distincte (ca tupluri de valori ale atributelor de intrare). Fie k_1, k_2 $\cdots$ k_d numărul de instanţe etichetate pentru fiecare caz distinct în parte din cele d. Atunci vom avea:

$$k_1 \le k,\, k_2 \le k, \cdots, k_d \le k \Rightarrow n = k_1 + k_2 + \cdots + k_d \le k\cdot d \text{ deci } n \le k\cdot d$$

Eroarea maximă la antrenare va fi dată de formula

$$E = \frac{(k_1-1)+(k_2-1)+\cdots+(k_d-1)}{n} = \frac{n-d}{n}$$

Avem:

$$E \le \frac{k-1}{k} \Leftrightarrow \frac{n-d}{n} \le \frac{k-1}{k} \Leftrightarrow k\cdot n - k\cdot d \le k\cdot n - n \Leftrightarrow k\cdot d \ge n \text{ (adev.)}$$

Prin urmare, eroarea maximă la antrenare care poate fi atinsă atunci când atributul de ieşire poate lua k valori distincte este $\frac{k-1}{k}$.

7. **(ID3, aspecte computaţionale: influenţa atributelor duplicate, respectiv a instanţelor de antrenament duplicate asupra arborelui ID3 rezultat)**

CMU, 2009 spring, Ziv Bar-Joseph, final exam, pr. 3.1-2

Se doreşte construirea unui arbore de decizie pentru n vectori, cu m atribute.

a. Să presupunem că există i şi j astfel încât pentru TOŢI vectorii X din datele de antrenament, aceste atribute au valori egale ($x_i = x_j$ pentru toţi vectorii, unde x_i este valoarea atributului i în vectorul X). Să presupunem de asemenea că în cazul în care ambele atribute duc la acelaşi câştig de informaţie vom folosi atributul i. Ştergerea atributului j din datele de antrenament poate schimba arborele de decizie obţinut? Explicaţi pe scurt.

b. Să presupunem că există în mulţimea de antrenament doi vectori egali X şi Z (adică, toate atributele lui X şi Z sunt exact la fel, inclusiv etichetele). Ştergerea vectorului Z din datele de antrenament poate schimba arborele de decizie obţinut? Explicaţi pe scurt.

Răspuns:

a. Nu, îndepărtarea atributului j nu schimbă arborele de decizie, deoarece atributele i şi j conduc la valori egale pentru câştigul de informaţie în fiecare nod al arborelui.

b. Da, în acest caz arborele de decizie se poate schimba, fiindcă entropia condiţională — care se calculează în fiecare nod pentru a determina atributul cu câştigul de informaţie cel mai mare — depinde de numărul de instanţe de antrenament luate în considerare.

8. **(Arbori de decizie: o margine superioară pentru numărul de noduri frunză, în funcţie de numărul atributelor şi numărul de exemple)**

CMU, 2005 fall, T. Mitchell, A. Moore, midterm exam, pr. 2.d

Presupunem că învăţăm un arbore de decizie folosind un set de R instanţe de antrenament descrise de M atribute de intrare având valori binare.

Care este numărul maxim posibil de noduri frunză din arborele de decizie, presupunând că fiecărui nod frunză îi este asociat măcar un exemplu de antrenament? Încercuiţi unul din răspunsurile de mai jos; justificaţi alegerea făcută.

R, $\log_2(R)$, R^2, 2^R, M, $\log_2(M)$, M^2, 2^M,
$\min(R, M)$, $\min(R, \log_2(M))$, $\min(R, M^2)$, $\min(R, 2^M)$,
$\min(\log_2(R), M)$, $\min(\log_2(R), \log_2(M))$, $\min(\log_2(R), M^2)$, $\min(\log_2(R), 2^M)$,
$\min(R^2, M)$, $\min(R^2, \log_2(M))$, $\min(R^2, M^2)$, $\min(R^2, 2^M)$,
$\min(2^R, M)$, $\min(2^R, \log_2(M))$, $\min(2^R, M^2)$, $\min(2^R, 2^M)$,
$\max(R, M)$, $\max(R, \log_2(M))$, $\max(R, M^2)$, $\max(R, 2^M)$,
$\max(\log_2(R), M)$, $\max(\log_2(R), \log_2(M))$, $\max(\log_2(R), M^2)$, $\max(\log_2(R), 2^M)$,
$\max(R^2, M)$, $\max(R^2, \log_2(M))$, $\max(R^2, M^2)$, $\max(R^2, 2^M)$,
$\max(2^R, M)$, $\max(2^R, \log_2(M))$, $\max(2^R, M^2)$, $\max(2^R, 2^M)$.

Răspuns:

Notăm cu $\max_{frunze}$ **valoarea căutată. Trebuie luate în considerare două aspecte:**

(a) **fiecare nod frunză trebuie să clasifice măcar un exemplu de antrenament** $\Rightarrow$ **nu putem să avem mai multe frunze decât exemple de antrenament** $\Rightarrow \max_{frunze} \leq R$

(b) **fiecare atribut poate fi testat o singură dată pe un drum de la rădăcină la o frunză oarecare** $\Rightarrow$ **arborele obţinut va avea adâncimea cel mult** M $\Rightarrow \max_{frunze} \leq 2^M$

$$\left.\begin{array}{l} \max_{frunze} \leq R \\ \max_{frunze} \leq 2^M \end{array}\right\} \Rightarrow \max_{frunze} \leq \min(R, 2^M)$$

Această valoare poate fi atinsă: putem să luăm, de exemplu, cazul trivial când avem o singură instanţă de antrenament. Prin urmare, avem $\max_{frunze} = \min(R, 2^M)$**.**

3.2 Arbori de decizie — Probleme propuse

28. **(Arbori de decizie; optimalitate, ca număr minim de noduri)**

*

Reprezentaţi arborele/arborii de decizie care are/au numărul minim posibil de noduri (de test) şi corespunde/corespund funcţiei boolene $(A \text{ XOR } B) \wedge C$ definită peste atributele boolene A, B şi C.

29. **(Expresivitatea arborilor de decizie: un rezultat privind funcţiile boolene)**

Orice funcţie booleană (care primeşte n argumente din mulţimea $\{0,1\}$ şi întoarce un element din mulţimea $\{0,1\}$) poate fi reprezentată cu ajutorul unui arbore de decizie. Adevărat sau fals?

În cazul afirmativ, explicaţi succint cum anume poate fi construit arborele de decizie respectiv.

În cazul negativ, daţi un exemplu de funcţie booleană pentru care nu se poate construi un arbore de decizie consistent cu funcţia respectivă.

30. **(Calcularea câştigului de informaţie pe "decision stumps")**

□ • ***CMU, 2013 fall, W. Cohen, E. Xing, Sample Questions, pr. 4***

Studentul Timmy doreşte să ştie cum [ar trebui să procedeze cel mai bine ca] să promoveze examenul de învăţare automată. Pentru aceasta, a cules informaţii de la studenţii care au urmat acest curs în anii precedenţi şi apoi a decis să-şi construiască un *model* bazat pe arbori de decizie. A colectat în total nouă *instanţe*/exemple, descrise cu ajutorul a două *trăsături* (văzute în cele ce urmează ca două variabile aleatoare, S şi A): „este bine să stai şi să înveţi până noaptea târziu înainte de examen" (S) şi „este bine să mergi la toate cursurile şi seminariile" (A). Timmy dispune acum de următoarele „statistici" (care sunt de fapt *partiţionări* ale datelor sale):

$$Set(all) = [5+, 4-]$$
$$Set(S+) = [3+, 2-], Set(S-) = [2+, 2-]$$
$$Set(A+) = [5+, 1-], Set(A-) = [0+, 3-]$$

Presupunând că se foloseşte drept criteriu de selecţie a celei mai bune trăsături câştigul maxim de informaţie, ce trăsătură va alege Timmy? Care este valoarea câştigului de informaţie?

Puteţi folosi la calcule următoarele aproximaţii:

N	3	5	7
$\log_2 N$	1.5850	2.3219	2.8073

31. **(Implementare: entropie, entropie condiţională specifică, entropie condiţională medie, câştig de informaţie)**

□ ***Liviu Ciortuz, 2016***

Folosind limbajul de programare pe care-l preferaţi, implementaţi un program care, pornind de la o structură de date de tip compas de decizie (engl., decision stump), calculează entropia, entropiile condiţionale specifice, entropia condiţională medie, precum şi câştigul de informaţie aferent.

În mod concret, programul va primi ca *input*

– m — numărul de valori posibile ale etichetei / atributului de ieşire (în mod implicit, se va considera $m = 2$);

– n — numărul de valori ale atributului (notat mai jos cu A) în raport cu care se face partiţionarea mulţimii de instanţe asociate nodului-rădăcină al compasului de decizie (valoarea implicită: $n = 2$);

– partiţiile (de fapt, count-urile) corespunzătoare nodurilor descendente. Pornind de la aceste partiţii, programul va calcula partiţia asociată nod-rădăcină al compasului de decizie.

De *exemplu*, pentru primul compas de decizie de la problema 30, inputul va avea forma $[3, 2]$, $[2, 2]$, în vreme ce pentru al doilea compas de decizie va fi $[5, 1]$, $[0, 3]$.

Programul va calcula şi apoi va afişa

– entropia atributului / variabilei de ieşire (notată aici cu Y);

– entropiile condiţionale specifice pentru fiecare descendent din nodul-rădăcină;

– entropia condiţională medie a atributului A;

– câştigul de informaţie al atributului [de ieşire] Y în raport cu atributul [de intrare] A.

32. **(Algoritmul ID3: aplicare pe expresii booleene; exploatarea simetriilor operaţiilor $\vee, \wedge$ în alegerea nodurilor; analiza „optimalităţii“ arborelui ID3)**

∗ ***prelucrare de Liviu Ciortuz după Tom Mitchell, "Machine Learning", 1997, ex. 3.1.d***

Considerăm următoarea funcţie booleană: $(A \wedge B) \vee (C \wedge D)$. Valorile pe care le ia această funcţie, calculate conform diferitelor valori de adevăr atribuite variabilelor/atributelor A, B, C şi D sunt cele cunoscute din logica propoziţiilor. Dorim însă să reprezentăm această funcţie ca arbore de decizie.

a. Aplicaţi algoritmul ID3 acestei funcţii.
***Observaţie*: Dacă exploataţi simetriile, este nevoie doar de puţine calcule, altfel vă complicaţi în mod inutil.**

b. Arborele ID3 obţinut la punctul precedent este optimal?
Alfel spus, puteţi găsi alt arbore de decizie de adâncime mai mică sau cu număr mai mic de noduri (de test) pentru această funcţie? (Ţineţi cont că în fiecare nod al unui arbore de decizie se poate testa un sigur atribut.)

33. **(Algoritmul ID3: aplicare analiza „optimalităţii“ arborelui ID3)**

• ∗ ***CMU, 2005 spring, C. Guestrin, T. Mitchell, midterm, pr. 4***

Agenţia spaţială NASA doreşte să distingă între marţieni (M) şi pământeni (H) folosind următoarele caracteristici: $Green \in \{N, Y\}$, $Legs \in \{2, 3\}$, $Height \in \{S, T\}$, $Smelly \in \{N, Y\}$.

Datele de antrenament de care dispunem sunt prezentate în tabelul alăturat.

	Species	Green	Legs	Height	Smelly
1	M	N	3	S	Y
2	M	Y	2	T	N
3	M	Y	3	T	N
4	M	N	2	S	Y
5	M	Y	3	T	N
6	H	N	2	T	Y
7	H	N	2	S	N
8	H	N	2	T	N
9	H	Y	2	S	N
10	H	N	2	T	Y

a. Învăţaţi un arbore de decizie folosind algoritmul ID3 şi trasaţi arborele respectiv.

b. Descrieţi conceptul M (marţian) ca un set de reguli conjunctive din logica propoziţiilor. Spre exemplu:

if $Green = Y$ **and** $Legs = 2$ **and** $Height = T$ **and** $Smelly = N$ **then** M;
else
if ... **then** M; **else** H.

c. Soluţia de la punctul b **de mai sus foloseşte cel mult 4 atribute în fiecare conjuncţie. Găsiţi un set de reguli conjunctive care folosesc doar 2 atribute pentru fiecare conjuncţie, păstrând însă eroarea la antrenare zero. Această ipoteză mai simplă poate fi reprezentată ca un arbore de decizie de adâncime 2? Justificaţi răspunsul.**

34. **(Algoritmul ID3: aplicare; cazul instanţelor de antrenament cu multiple apariţii)**

CMU, 2010 fall, Aarti Singh, HW2, pr. 5.1

Tabelul alăturat descrie instanţe (înregistrări) pozitive şi instanţe negative pentru persoane cărora banca le-a acordat (sau nu le-a acordat) un card de credit. Fiecare linie din tabel indică nişte combinaţii de valori observate pentru atributele considerate (*Gender*, *Income* şi *Approved*) şi de câte ori a fost înregistrată respectiva combinaţie de valori.

Gender	*Income*	*Approved*	*Counts*
F	Low	+	10
F	High	+	95
M	Low	+	5
M	High	+	90
F	Low	-	80
F	High	-	20
M	Low	-	120
M	High	-	30

De exemplu, (F, Low, +) a apărut de 10 ori, iar (F, Low, -) de 80 de ori.

a. Calculaţi entropia atributului *Approved* pe acest set de date de antrenament (folosind logaritmul cu baza 2).

b. Calculaţi de asemenea câştigurile de informaţie IG(*Approved, Gender*) şi IG(*Approved, Income*).

c. Desenaţi un arbore de decizie produs de către algoritmul ID3 (fără post-pruning) pe baza acestui set de date de antrenament.

35. **("Decision stump" produs de ID3: raţionament calitativ pe un exemplu simplu)**

• ○ *CMU, 2010 fall, Ziv Bar-Joseph, midterm exam, pr. 5.a*

Vrem să construim un arbore de decizie care să ne ajute să prezicem întârzierile avioanelor. Timp de câteva luni am colectat informaţii, iar un rezumat al acestora este prezentat în tabelul următor:

Atribut	Valoare = *Da*		Valoare = *Nu*	
	#Zboruri		#Zboruri	
	amânate	neamânate	amânate	neamânate
Ploaie	30	10	10	30
Vânt	25	15	15	25
Vara	5	35	35	5
Iarna	20	10	20	30
Ziua	20	20	20	20
Noaptea	15	10	25	30

a. Pe baza acestui tabel precizaţi ce atribut ar trebui să fie pus în rădăcina arborelui de decizie. (Justificaţi pe scurt, nu este necesar să calculaţi valorile exacte pentru câştigul de informaţie.)

b. Pe baza aceluiaşi tabel, precizaţi care dintre atribute ar trebui să apară pe al doilea nivel (nivelul de sub rădăcină) al arborelui de decizie. (Justificaţi pe scurt, nu este necesar să calculaţi valorile exacte pentru câştigul de informaţie.)

36. **(Clasificare ternară: "decision stump" produs de ID3, pe date care conţin duplicări şi „zgomote")**

• *

Presupunem că se dau şase date de antrenament (precizate în tabel) pentru o problemă de clasificare cu două atribute binare şi trei clase $Y \in \{1, 2, 3\}$. Se va crea un arbore ID3, bazat pe câştigul de informaţie.

X_1	X_2	Y
1	1	1
1	1	1
1	1	2
1	0	3
0	0	2
0	0	3

a. Calculaţi câştigul de informaţie atât pentru X_1 cât şi pentru X_2. Se va folosi aproximarea $\log_2 3 = 19/12$ şi se va scrie câştigul de informaţie sub formă de fracţii.

b. Pe baza rezultatelor anterioare, ce atribut va fi folosit pentru primul nod al arborelui ID3? Desenaţi arborele de decizie care rezultă folosind doar acest singur nod. Etichetaţi corespunzător nodul, ramurile şi eticheta prevăzută în fiecare frunză.

c. Cum va clasifica acest arbore instanţa determinată de $X_1 = 0$ şi $X_2 = 1$?

37. **(O aproximare a numărului de instanţe greşit clasificate care au fost asignate la un nod frunză dintr-un arbore ID3)**

• ○ ***CMU, 2003 fall, Andrew Moore, midterm exam, pr. 9.b***

Învăţăm un arbore de decizie folosind un set de date de antrenament cu atributul de ieşire (*class*) având valorile 0 sau 1.

Presupunând că pentru un nod frunză l din acest arbore,

– există M instanţe de antrenament asignate la acel nod, iar

– entropia sa este H,

schiţaţi un algoritm simplu care ia ca valori de intrare M şi H şi furnizează la ieşire numărul de exemple de antrenament clasificate greşit de către nodul frunză l.

***Sugestie*: Folosiţi o aproximare simplă (polinomială) pentru funcţia entropie $H(p)$.**

38. **(Algoritmul ID3: eroarea la antrenare)**

• ○ ***CMU, 2003 fall, Andrew Moore, HW1, pr. 2.1***

Un student mi-a spus următoarele:

– el poate să construiască un set de instanţe cu atributele de intrare discrete şi atributul de ieşire binar;

– mie îmi dă voie să aleg o parte din acest set de instanţe (dar nu toate!) pentru a antrena un arbore de decizie;

– indiferent de modul cum mi-aş alege datele de antrenament din setul construit de el, eroarea de clasificare pe care arborele de decizie (obţinut în urma antrenării) o va face pe instanţele care nu au fost incluse în setul de antrenament va fi de cel puţin 50%.

Credeţi că studentul are dreptate? Explicaţi de ce sau daţi un exemplu.