CONSTRUCŢIA ARBORILOR DE CLASIFICARE - ALGORITMUL ID3

ing. Lorina Negreanu

Universitatea "Politehnica" Bucureşti

Rezumat: Achiziţia automată de cunoştinţe este unul dintre principalele domenii de aplicabilitate a conceptelor şi tehnicilor de învăţare automată. O metoda eficientă şi deja foarte populară pentru inducerea regulilor de clasificare din exemple este algoritmul ID3 (Quinlan). Prezentul articol descrie algoritmul ID3 şi prezintă o extindere a utilizării sale pentru probleme cu caracter nedeterminist. De asemenea sunt prezentate pe scurt câteva optimizări ale algoritmului, respectiv algoritmii ID3 Generalizat, ID4 şi ID5.

Cuvinte cheie: arbore de clasificare, atribut, clasă, exemple, mesaj, informaţie, grad de certitudine, nedeterminism, bază de cunoştinţe, diagnoză.

1. Prezentarea algoritmului ID3

1.1. Principiul generării inductive a arborilor de clasificare

Algoritmul ID3, un descendent al algoritmului CLS [2], porneşte de la o mulţime de obiecte descrise printr-o mulţime fixă de proprietăţi şi construieşte un arbore de decizie (clasificare) care poate clasifica toate obiec-tele din mulţime.

Un obiect este caracterizat printr-o mulţime fixă de atribute (proprietăţi), fiecare atribut având la rândul său o mulţime de valori posibile. Descrierea unui obiect (exemplu) constă într-o succesiune de valori de atribut. Dacă C este o mulţime de obiecte, o regulă de clasificare sub forma unui arbore de decizie poate fi construită astfel:

- dacă C este mulţimea vidă ea este asociată arbitrar cu orice clasă;
- dacă toate obiectele din C aparţin aceleiaşi clase, atunci arborele de decizie constă doar dintr-o frunză ce are numele clasei respective;
- altfel, se selectează un atribut şi se împarte C în submulţimile disjuncte C_1, C_2, .., C_n, unde C_i conţine acei membri din C care au valoarea i a atributului selectat; pentru fiecare dintre aceste submulţimi se aplică strategia prezentată.

În final, se generează un arbore în care frunzele sunt etichetate cu nume de clase, iar nodurile interioare specifică atributul care trebuie testat, având arce care pleacă din nod pentru fiecare valoare posibilă a atributului respectiv.

Pentru a ilustra cele prezentate, se consideră atributele *înălţime* având valorile {înalt, scund}, *culoare_păr* având valorile {şaten, blond, roşcat}, *culoare_ochi* având valorile {verde, negru}. Clasele posibile sunt {+, -}. Mulţimea exemplelor (C) este următoarea:

$C =$

înălţime	*culoare_păr*	*culoare_ochi*	*clasa*
scund	blond	verde	+
înalt	blond	negru	-
înalt	roşcat	verde	+
scund	şaten	verde	-
înalt	şaten	verde	-
înalt	blond	verde	+
înalt	şaten	negru	-
scund	blond	negru	-

Pentru a construi arborele de decizie trebuie selectat un atribut care să fie rădăcina arborelui. Fie acesta, atributul *culoare_păr*. Arborele care rezultă pentru acest pas este prezentat în figura 1.

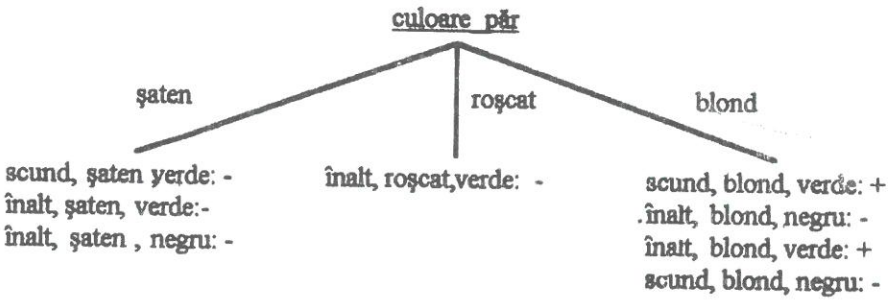

Figura 1. Arbore de decizie parţial

Submulţimile corespunzătoare valorilor *şaten* şi *roşcat* conţin obiecte care aparţin aceleaşi clase, deci ele nu mai trebuie prelucrate. Pentru ramura corespunzătoare valorii de atribut *blond*, trebuie ales un alt atribut pentru a fi testat. Fie acesta *culoare_ochi*. Arborele care se obţine este prezentat în figura 2.

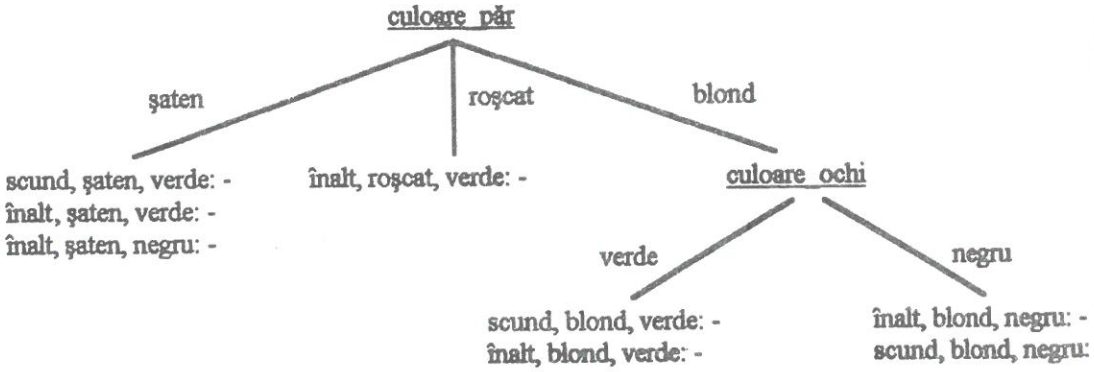

Figura 2. Arbore de clasificare complet

Toate submulţimile conţin obiecte care aparţin aceleiaşi clase, deci pot fi înlocuite cu noduri etichetate cu numele claselor, obţinându-se arborele prezentat în figura 3.

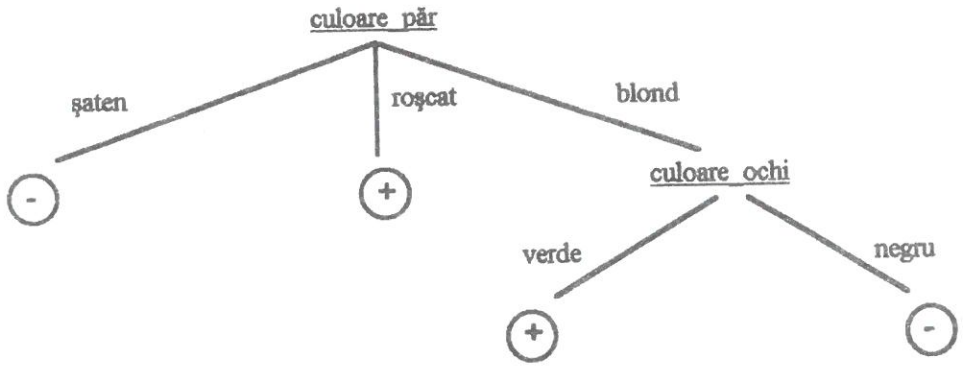

Figura 3. Arbore de clasificare

Pentru a clasifica un obiect, se porneşte din rădăcina arborelui de decizie şi se urmează calea dictată de valorile proprii obiectului respectiv, până se ajunge la o frunză. Aşa cum se observă, clasificarea unui obiect implică evaluarea unui număr mic de atribute, respectiv a celor care se află pe calea de la rădăcină la frunza corespunzătoare. Arborele construit trebuie să poată clasifica şi obiecte care nu au fost în mulţimea iniţială de exemple. Obiectele care au aceleaşi valori de atribut, dar aparţin unor clase diferite, reprezintă inconsistenţe ale mulţimii de exemple şi desigur ele nu pot fi clasificate.

În general, dimensiunea arborelui de clasificare depinde de ordinea în care sunt alese atributele pentru a fi evaluate. Deci, problema care se pune este următoarea: având o mulţime de exemple şi un număr de arbori de clasificare diferiţi, care este cel care are cele mai multe şanse să clasifice corect obiecte care nu aparţin mulţimii iniţiale?

Algoritmul ID3 porneşte de la premisa că arborele optim este cel mai "simplu" arbore care satisface mulţimea de exemple. Raţiunea acestei euristici este principiul enunţat de logicianul medieval William din Occam în 1324 (*Occam's Razor)*: este inutil să faci cu mai mult, ceea ce poţi să faci cu mai puţin.

Dacă presupunem că mulţimea iniţială de exemple este suficientă pentru a construi o generalizare validă, atunci problema se reduce la diferenţierea proprietăţilor necesare.

1.2. Evaluarea unei proprietăţi

O proprietate poate fi evaluată prin prisma contribuţiei informaţionale pe care o aduce pentru procesul de clasificare. De exemplu, dacă dorim să determinăm specia unui animal, proprietatea "are pene", contribuie cu o anumită cantitate de informaţie la atingerea scopului. ID3 măsoară câştigul potenţial de informaţie al fiecărei proprietăţi, punând-o ca rădăcină a subarborelui curent şi alege proprietatea pentru care câştigul este maxim. Baza matematică pentru această evaluare este teoria informaţională a lui Shanon(1984).

Un mesaj poate fi considerat ca o instanţă într-un univers de mesaje posibile. Actul transmiterii unui mesaj este similar selectării unui mesaj din univers. Din acest punct de vedere conţinutul de informaţie al unui mesaj poate fi considerat dependent de mărimea universului şi de frecvenţa de apariţie a mesajului. Shanon a formalizat aceste intuiţii definind cantitatea de informaţie dintr-un mesaj ca o funcţie a probabilităţii de apariţie a fiecărui mesaj.

Fie $M = \{m_1, m_2, \ldots, m_n\}$ un univers de mesaje şi $p(m_i)$, i = 1,n probabilitatea de apariţie a fiecărui mesaj. Cantitatea de informaţie a unui mesaj este dată de relaţia: .

$$I(M) = \Sigma_{i=1}^{n} - p(m_i)\log_2(p(m_i))$$

Algoritmul ID3 utilizează teoria informaţiei pentru a selecta atributul (testul) care aduce cel mai mare câştig de informaţie pentru clasificarea mulţimii de exemple. Pentru un anumit test, câştigul de informaţie obţinut, dacă testul respectiv este făcut în rădăcina arborelui, este egal cu diferenţa dintre informaţia din arbore şi informaţia necesară pentru realizarea clasificării după ce testul a fost efectuat. Cantitatea de informaţie necesară pentru a construi arborele este definită ca o medie a cantităţii de informaţie din fiecare subarbore ponderată cu procentul exemplelor prezente în subarbore.

Fie C o mulţime de exemple. Dacă proprietatea P, care are *n* valori, devine rădăcina subarborelui curent, aceasta va partiţiona mulţimea C în submulţimile $\{C_1, C_2, \ldots, C_n\}$ corespunzătoare valorilor atributului. Cantitatea aşteptată de informaţie, necesară construirii arborelui a cărui rădăcină este P, este:

$$E(P) = \Sigma_{i=1}^{n} \frac{|C_i|}{|C|} I(C_i)$$

Câştigul obţinut prin testarea proprietăţii P este diferenţa dintre conţinutul de informaţie al arborelui şi cantitatea de informaţie necesară pentru construcţia arborelui:

$$c(P) = I(C) - E(P)$$

Ca rădăcină a arborelui curent este selectată proprietatea care aduce cel mai mare câştig de informaţie.

În cazul exemplului prezentat,

$$I(C) = -3/8\log_2 3/8 - 5/8\log_2 5/8 = 0{,}954$$

Pentru a decide care atribut să devină rădăcina arborelui, trebuie calculată cantitatea de informaţie estimată pentru fiecare atribut în parte.

În cazul atributului *înălţime*, pentru valoarea *înalt* :

$$I(C_1) = -2/5\log_2 2/5 - 3/5\log_2 3/5 = 0.971$$

iar pentru valoarea *scund*:

$$I(C_2) = -1/3\log_2 1/3 - 2/3\log_2 2/3 = 0.918$$

Cantitatea estimată de informaţie, este:

$$E(inaltime) = 5/8*0.971 + 3/8*0.918 = 0.951$$

iar câştigul obţinut prin testarea acestui atribut este:

$$0.954 - 0.951 = 0.003,$$

practic neglijabil.

În cazul celui de al doilea atribut, *culoare_păr*, valorile sunt *roşcat, blond, şaten,* iar cantitatea de informaţie estimată, calculată analog, este:

E(culoare_păr) = 0.5

Câştigul obţinut prin testarea acestui atribut este:

$$0.954 - 0.5 = 0.454$$

Printr-un calcul analog, câştigul de informaţie în cazul testării atributului *culoarea_ochi* este 0.347. Principiul maximi-zării câştigului de informaţie va determina algoritmul ID3 să selecteze atributul *culoare_păr* ca rădăcină a arborelui de decizie.

Metoda descrisă pentru construcţia arborilor de clasificare presupune că operaţiile asupra mulţimii C de obiecte (cum ar fi determinarea mulţimii corespunzătoare valorii de atribut A_i a atributului *A)* pot fi făcute eficient, ceea ce practic înseamnă că mulţimea C trebuie păstrată în memoria internă. Ce se întâmplă însă dacă dimensiunea mulţimii C nu permite acest lucru? Soluţia adoptată de ID3 se bazează pe o rafinare incrementală a arborilor de clasificare.

Se selectează aleator o submulţime de instanţe, numită *fereastră*. Pentru această submulţime se construieşte arborele de decizie. Dintre exemplele care au rămas în afara ferestrei se selectează cele care nu pot fi clasificate cu arborele construit şi se formează o nouă fereastră care include şi aceste exemple. Se construieşte un nou arbore de decizie. Procesul se reia până când se construieşte un arbore pentru care nu mai există excepţii, deci care clasifică, în mod corect, toate exemplele din mulţimea iniţială C.

Pentru formarea unei noi ferestre, în [4] sunt prezentate două metode. Prima metodă măreşte fereastra curentă prin adăugarea unui număr de excepţii până la o limită dată. Cea de a doua metodă încearcă să identifice obiectele "cheie" din fereastra curentă şi să le înlocuiască pe celelalte cu excepţiile determinate, păstrând astfel dimensiunea ferestrei constantă. Ambele metode au fost testate pentru probleme de clasificare netriviale pentru care mulţimea iniţială avea aproximativ 2000 exemple, implicând 14 atribute.

Rezultatele obţinute au demonstrat că a fost posibilă construcţia unui arbore de decizie, care clasifica toate exemplele din mulţime corect, el fiind obţinut dintr-o fereastră care reprezenta doar o mică fracţiune a mulţimii iniţiale. De asemenea, un lucru foarte important, este acela că timpul necesar pentru obţinerea arborelui de clasificare creşte liniar cu dificultatea problemei.

2. Utilizarea algoritmului ID3 pentru probleme cu caracter nedeterminist

Algoritmul ID3 prezentat a fost dezvoltat pentru rezolvarea problemelor deterministe de tip cauză-efect. Pornind de la o mulţime dată de exemple C, se construieşte un arbore de decizie. Clasificarea unui nou exemplu implică un algoritm de parcurgere top-down. Se porneşte din rădăcină şi se urmează calea dictată de valorile de atribut existente în exemplul respectiv până se ajunge la o frunză, dacă exemplul poate fi clasificat cu acel arbore. În frunză există o singură clasă, adică un singur diagnostic cu grad de certitudine 1. Cu alte cuvinte, restricţiile de aplicare ale algoritmului de clasificare sunt următoarele:

- valorile atributelor unei căi trebuie să fie cunoscute;
- este posibil un singur diagnostic, pentru că o frunză conţine o singură clasă.

Dar, de cele mai multe ori, problemele reale au un caracter nedeterminist. În cazul problemei de clasificare, nedeterminismul se reflectă în următoarele caracteristici:

-valorile atributelor nu sunt cunoscute în totalitate;

-pentru o cale din arbore cores-pund mai multe diagnosti-ce(clase) cu grade diferite de certitudine.

În acest caz, mulţimea iniţială de exemple are următoarea formă:

p_1	p_2	. . . . .	p_n	*diagnostic*
v_1^1	v_2^1	. . . . .	v_n^k	d_1 cu grad de certitudine c_1
v_1^i	_	. . . . .	v_n^l	d_2 cu grad de certitudine c_2

Figura 4. Tabel de exemple cu informaţie incompletă şi incertă

unde $i, j, k < n$; liniuţa orizontală repre-zintă o valoare de atribut necunoscută.

În cazul unei valori de atribut necunoscute, se consideră că oricare dintre valorile permise ale atributului este posibilă şi se generează exemplul respectiv pentru toate valorile atributului. Este posibil ca prin multiplicarea exemplului pentru diferite valori de atribut să se obţină exemple identice din punct de vedere al valorilor de atribut dar având clase diferite. Caz în care în arborele de clasificare va apare o frunză cu mai multe diagnostice cu grad de certitudine diferit. Figura 5 ilustrează acest caz.

p_1	p_2	p_3	p_4	*diagnostic*
v_1^1	v_2^1	v_3^2	v_4^3	d_1 cu grad de certitudine c_1
v_1^1	_	v_3^2	v_4^3	d_2 cu grad de certitudine c_2

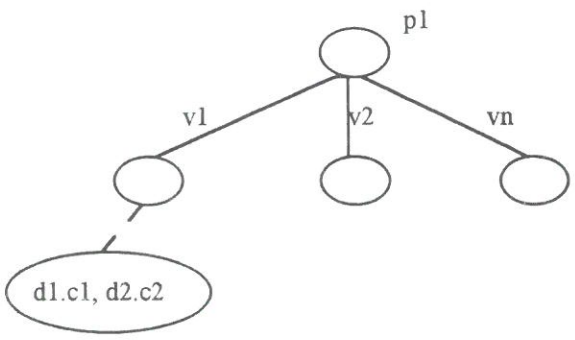

Figura 5. Exemple identice cu diagnostice diferite

2.1. Descrierea unei baze de cunoştinţe ale cărei reguli au fost generate prin ID3

Aşa cum a fost deja menţionat, algoritmul ID3 poate fi utilizat pentru sintetizarea regulilor unei baze de cunoştinţe, din exemple. Regulile sunt chiar arborii de clasificare obţinuţi dintr-o mulţime de exemple.

Sistemul cadru SCR (dezvoltat de autor, 1987) pentru generarea sistemelor expert bazate pe reguli, utilizează modelul cunoştinţelor incomplete şi incerte prezentat. Baza de cunoştinţe este formată din reguli generate din exemple, utilizând algoritmul ID3, modificat pentru tratarea cazurilor de nedeterminism.

Structura unui modul al bazei de cunoştinţe

Baza de cunoştinţe a sistemului SCR este formată din mai multe module a căror structură este ilustrată în figura următoare:

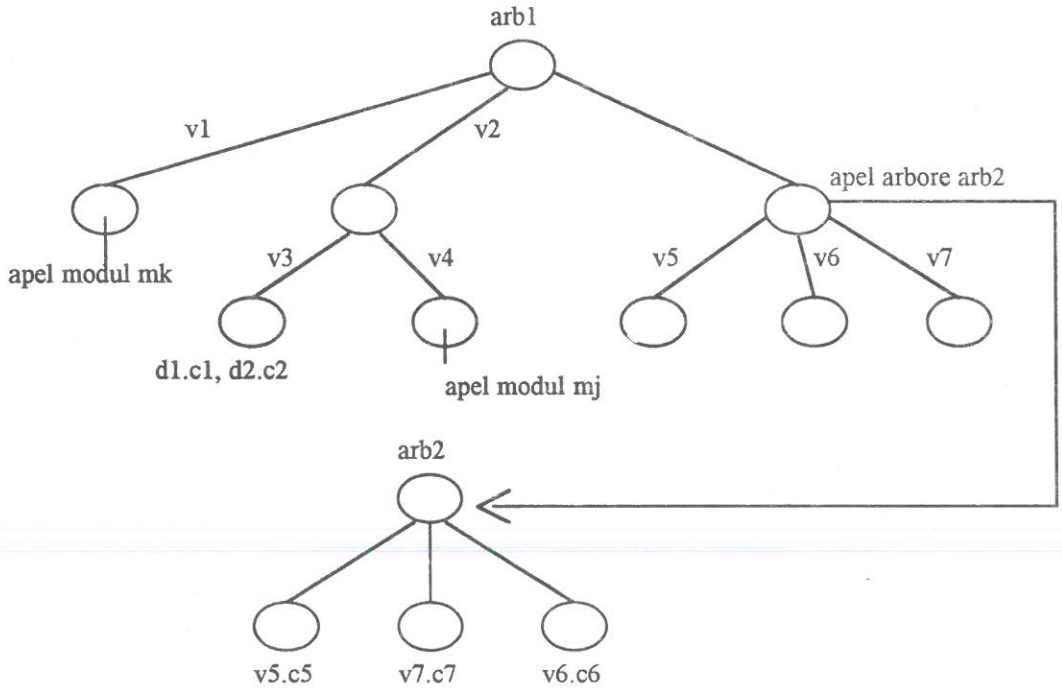

Figura 6. Structura unui modul din baza de cunoştinţe

Un modul al bazei de cunoştinţe poate fi format din mai mulţi arbori care se căracterizează prin următoarele elemente particulare:

- un nod intern al unui arbore poate desemna o proprietate sau invocarea unui arbore, din acelaşi modul, pentru stabilirea valorilor atributului din nod; este cazul invocării arborelui *arb2*;
- o frunză poate conţine unul sau mai multe diagnostice cu grade de certitudine diferite sau poate invoca un alt modul al bazei de cunoştinţe, caz în care procesul de diagnoză continuă în acel modul.

Pentru a avea o reprezentare uniformă a unui arbore au fost impuse următoarele restricţii de uniformitate:

- un modul poate conţine un singur arbore;
- un nod poate invoca un alt modul:
- dacă nodul este o frunză atunci valorile întoarse de modul sunt diagnostice;
- dacă nodul este intern atunci valorile reîntoarse sunt folosite mai departe în procesul de diagnoză.

2.2.Controlul procesului de diagnoză

În cazul parametrilor cunoscuţi à priori se selectează căile corespunzătoare valorilor respective. O dată ce calea este selectată, se realizează o "tăiere" a arborilor neselectaţi. Pentru atributele necunoscute care intervin, se evaluează atributul cel mai important, care trebuie obţinut şi se reia procesul.

Aşa cum rezultă din structura unui modul al bazei de cunoştinţe, valorile atributului unui nod care au fost obţinute prin interogarea unui alt modul pot fi ponderate. Aceasta însemnă că arcele arborelui pot avea ponderi şi deci meritul unei frunze (deci al unei clase) depinde de ponderile căii care duce la frunză.

Această organizare a bazei de cunoştinţe face posibilă ierarhizarea procesului de diagnoză, aşa cum se observă şi în figura următoare:

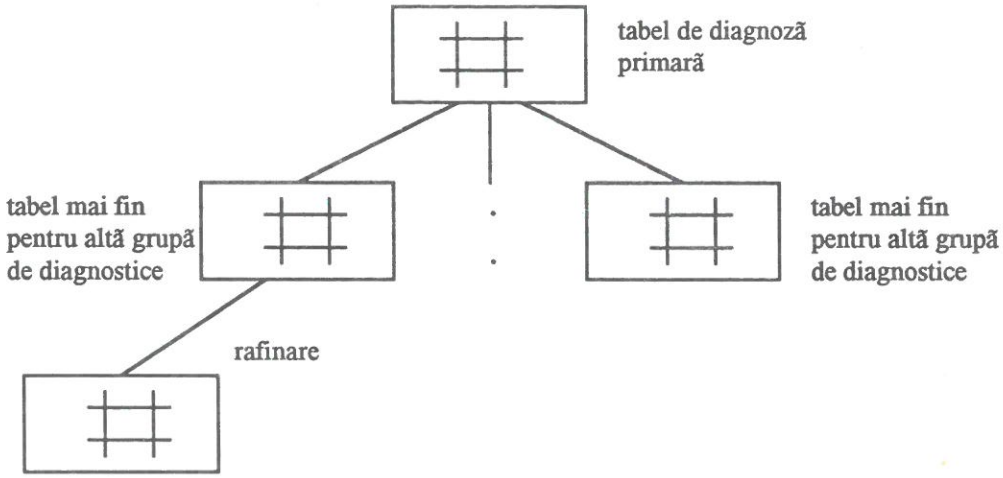

Figura 7. Rafinarea procesului de diagnoză

De exemplu, pentru diagnoza unui sistem de calcul, structura bazei de cunoştinţe (modulele) ar putea fi următoarea:

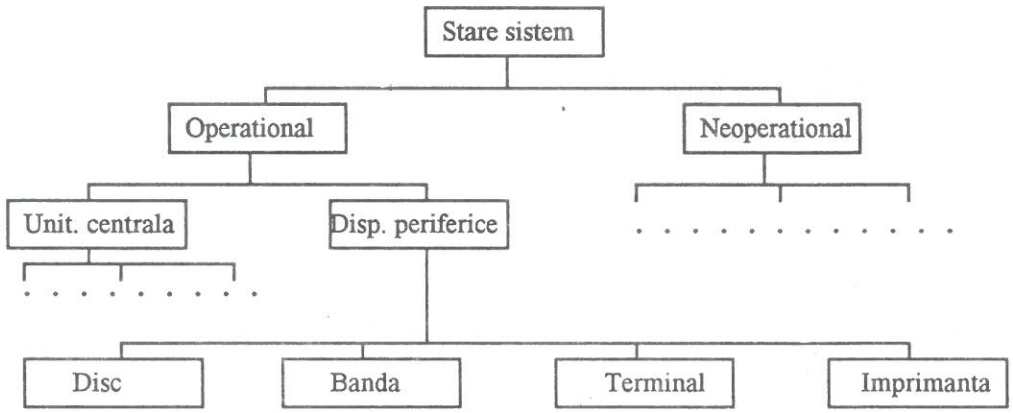

Figura 8. Diagnoza unui sistem de calcul

Avantajul cel mai important al ierarhizării procesului de diagnoză constă într-o rapidă diagnosticare prin restrângerea spaţiului de căutare. Desigur există avantaje semnificative şi din punct de vedere al construcţiei bazei de cunoştinţe, un exemplu fiind paralelizarea construcţiei ei. De exemplu, introducerea exemplelor pentru modulul disc poate fi realizată în paralel cu oricare alt modul.

3. Optimizări ale algoritmului ID3

3.1. ID3 generalizat

Algoritmul ID3 realizează o căutare euristică, de tip "hill-climbing", în spaţiul arborilor de decizie posibili. Căutările în spaţiul stărilor de tip "hill-climbing" sunt căutări fără revenire care se bazează pe criterii de optim local în alegerea căii. Dar nu întotdeauna optimele locale conduc la un optim global, astfel încât soluţia determinată nu este întotdeauna cea mai bună. Din punct de vedere al algoritmului ID3, această "slăbiciune" constă în pierderea unor arbori de decizie mai buni, pentru aceleaşi date.

Există două aspecte ale unor arbori "mai puţin buni". Problema *valorilor irelevante* şi problema *arcelor care lipsesc*. Atunci când ID3 alege un atribut pentru rădăcina subarborelui curent, creează arce pentru toate valorile atributului respectiv care apar în exemple. Anumite valori pot fi relevante pentru clasificare, iar altele nu. Problema *valorilor irelevante* constă în crearea subarborilor pentru valori irelevante, care conduc la supraspecializarea regulilor. Se construiesc reguli pentru care trebuie verificate precondiţii care nu sunt necesare sau nu sunt relevante.

Problema *arcelor care lipsesc* constă, în esenţă, în reducerea capacităţii inductive a arborelui de clasificare. Ea este datorată faptului că

anumite submulţimi corespunzătoare unor noduri interne nu conţin exemple pentru toate valorile atributului din nod. O astfel de situaţie este ilustrată în figura 9.

A	*B*	*Clasa*
a_1	b_1	C_1
a_1	b_2	C_2
a_2	b_2	C_2
a_2	b_3	C_1
a_3	b_1	C_3
a_3	b_2	C_3

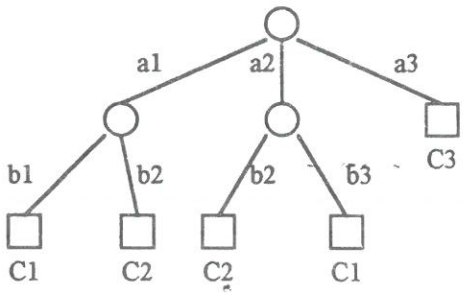

Figura 9. Arbore de decizie obţinut prin ID3

Pentru exemplele :

$e = A = a_3, B = b_3,$

$e' : A = a_1, B = b_3$

arborele construit îl poate clasifica pe primul, dar nu şi pe al doilea, pentru că în submulţimea corespunzătoare arcului etichetat cu a_1 nu există nici un exemplu pentru valoarea b_3.

Pentru a rezolva acest tip de probleme Cheng&All a propus o versiune generalizată a algorimului ID3 (GID3). În esenţă algoritmul este acelaşi, cu excepţia faptului că nu se creează arce pentru toate valorile atributului. Se iau în considerare doar valorile semnificative, iar celelalte se grupează într-o valoare considerată implicită. Alegerea atributelor semnificative se face în funcţie de un grad de toleranţa. Valorile de atribut care ies din plaja de toleranţa sunt grupate într-o valoare implicită. Prezentarea detaliată a algoritmului este în [1].

Arborele obţinut aplicând algoritmul ID3 generalizat, este următorul:

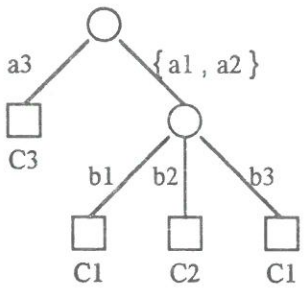

Figura 10. Arbore de decizie obţinut prin GID3

Acest arbore poate clasifica ambele exemple.

Arborii generaţi utilizând algoritmul GID3 sunt mai compacţi, mai fiabili şi mai generali decât cei obţinuţi prin aplicarea algoritmului ID3. O calitate importantă a algoritmului GID3 este faptul că aceste îmbunătăţiri sunt obţinute fără o creştere semnificativă a complexităţii algoritmului.

3.2. ID3 incremental

Construcţia arborilor de clasificare prin algoritmul ID3 este foarte bună în cazul în care baza de date (mulţimea de exemple) nu se modifică semnificativ. Pentru probleme în care noi exemple apar în mod regulat, ar fi de dorit ca arborele deja construit să poată fi modificat, în loc să fie reconstruit de fiecare dată.

Shlimmer şi Fischer (1986) au creat algoritmul ID4 care construieşte incremental arbori de clasificare, similari cu cei construiţi de ID3. Ideea algoritmului este de elimina subarborele corespunzător unui atribut atunci când acesta trebuie înlocuit cu un altul mai bun.

Utgoff creează o variantă a algoritmul ID4, ID5, care diferă în modul de înlocuire a atributului de test. În loc să elimine subarborele corespunzător atributului care trebuie înlocuit, restructurează arborele propagând noul atribut. Propagarea atributului implică un proces de restructurare a arborelui fără pierdere de informaţie. El creează frunze pentru informaţia din exemple care nu se regăseşte implicit în arborele existent astfel încât o propagare în sus să fie posibilă. Descrierea detaliată a algoritmului este prezentată în [4].

Costul utilizării algoritmului ID5 este mai mic decât cel al utilizării algoritmului ID3 pentru reconstrucţia arborelui, la adăugarea unui nou exemplu care nu poate fi clasificat, cu excepţia cazurilor în care atributele care trebuie înlocuite au un număr mare de valori.

Bibliografie

1. **CHENG J. , USAMA, M. F., KEKI, B. I., ZHAOGANG, Q.:** Improved Decision Trees: A Generalized Version of ID3.În: Proc. of the 5th Int. Conf. on MACHINE LEARNING, June 12-14, 1988.
2. **HUNT, E. B., MARIN, J., STONE, P. T.:** Experiments in Induction, Academic Press, New York, 1966.
3. **QUINLAN, J. R.:** Induction of Decision Trees. Machine Learning, Kluwer, 1986
4. **UTGOFF, P. E.:** ID5: An incremental ID3. În:Proc. of the 5th Int. Conf. on MACHINE LEARNING, June 12-14, 1988.