PETRESCU Mary-Violeta

METODE DE CONSTRUIRE A UNUI ARBORE DE DECIZIE

Craiova
Octombrie, 2024

Copyright © 2024, Autor
Tehnoredactare computerizată: PETRESCU Mary-Violeta
Corectura aparţine autorului

Responsabilitatea pentru conţinutul lucrării revine în exclusivitate autorului
Studiu științific

Publicat: octombrie 2024
http://www.edituraelse.ro/ped/ped.htm

METODE DE CONSTRUIRE A UNUI ARBORE DE DECIZIE

A. Concepte teoretice privind arborele de decizie

Arborele de decizie (DT) este unul din algoritmii utilizați în rezolvarea problemelor de clasificare, fiind considerat a fi un clasificator de forma unui arbore cu rădăcini, unde rădăcina nu are margini de intrare și toate celelalte noduri au exact o margine.

Un nod cu margini de ieșire este numit nod test (decizional) și reprezintă un test pentru un anumit atribut, iar fiecare arc care pleacă din nodul terminal este o valoare a atributului respectiv. Celelalte noduri sunt numite frunze sau noduri terminale. Unui nod terminal îi corespunde o clasă (a se vedea Figura 1).

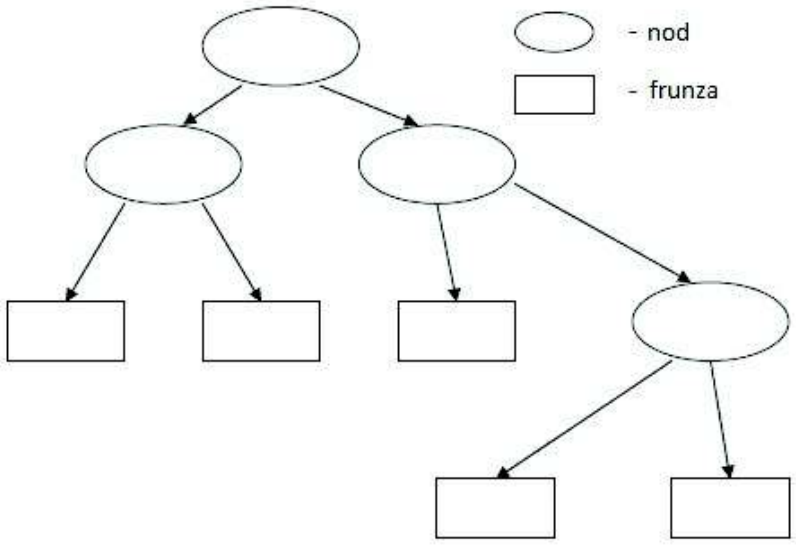


Figura 1. Arbore de decizie simplu

În principiu, se pot construi mai mulţi arbori de decizie având dat un set de atribute, dar unii dintre ei au o precizie mai mare de clasificare decât alţii.

De-a lungul anilor au fost creați o serie de algoritmi pentru obţinerea de arbori cu o precizie (acuratețe) mai mare de clasificare decât alţii.Cei mai cunoscuţi algoritmi sunt: CLS-Concept LearningSystem (Hunt et al.,1966), ID3 (Quinlan, 1986), C4.5 (Quinlan, 1993), CART (Breiman et al.,1984).

Obiectivul principal în construirea unui arbore de clasificare este găsirea celei mai bune divizări. Fiecare fracțiune divizată a eșantionului de date în două sau mai multe părți și fiecare submulțime a fracțiunii divizate are una sau mai multe clase în ea. Dacă există decât o singură clasă într-o submulțime, atunci este pur, altfel este impur.

Această măsură a impurității este specificat a fi criteriul de divizare. O măsură a impurității are proprietatea de a fi minimă atunci când există doar instanțe dintr-o singură clasă maximă, atunci când toate clasele au același număr de instanțe.

Există mai multe tipuri de criterii de divizare, cum ar fi: Entropia (Entropy), Câștigul de informație (Information Gain), Indexul Gini (Gini Index), Măsura de clasificare greșită (Misclassificationmeasure):

- *Entropia (Entropy)*este o măsură a dezordinii sau impurității.

$$\text{Entropy(S)}= H(S) = -\sum_{i=1}^{c} p_i \log_2 p_i \qquad (1)$$

unde am notat: S - setul de obiecte; p_i - ponderea elementelor i din setul S; c - numărul de clase.

Dacă entropia este 0 atunci, toate obiectele lui S aparţin aceleiaşi clase, dacă entropia este 1 atunci există un număr egal de elemente în fiecare clasă, iar dacă aceasta este între 0 şi 1 numărul de obiecte diferă de la o clasă la alta.

- *Câștigul de informație (Information Gain)*

Câștigul de informație este utilizat pentru a determina cât de bine un atribut face distincția între datele de instruire în funcție de clasele variabilei decizionale. Această măsură a informațiilor dobândite de la o divizare particulară este popularizată de (Quinlan, 1986).

Quinlan utilizează faimoasa formulă a Teoriei Informaționale, adică numărul minim de biți, pentru a codifica clasificarea unui membru arbitrar al unei colecții S. Entropia definită într-o colecție de date de instruire, S este:

$$H(S) = -\sum_{i=1}^{c} p_i \log_2 p_i \qquad (2)$$

Entropia poate fi interpretată ca numărul minim de biți necesari pentru a codifica clasificarea unui membru arbitrar al S.

Câștigul de informație reprezintă scăderea așteptată în entropie atunci când se face partiția exemplelor unui set S, în funcție de un atribut a_k. Este definit ca:

$$InfoGain(S, a_k) = H(S) - \sum_{v \in Values(a_k)} \frac{|S_v|}{|S|} H(S_v) \qquad (3)$$

unde :

- Valorile (a_k) suntmultimeatuturorvalorilorposibilepentru un atribut;
- S_v este un subset de exemple în S care au are valoarea v pentruatributul a_k;
- $|S|$ estenumărul de exemple în S;
- $|S_v|$ estenumărul de exemple în S_v.

Primul termen al ecuației îl reprezintă entropia setului de date original, iar doilea termen descrie entropia setului de date după partiția acestuia folosind atributul a_k.

Acesta nu reprezintă nimic mai mult decât o sumă a entropiilor fiecărui subset S dată de numărul de exemple aparținând subsetului.

Experimentele au arătat totuși că măsura numită *câștig de informație* (*informationgain*) tinde să favorizeze predictorii care au un număr mare de modalități.

Quinlan (1986) sugerează o soluție bazată pe considerarea cantității de informație necesară pentru a determina valoarea unui predictor a_k pentru un set de date S.

Aceasta este dată de entropia predictorului a pentru setul de date S, adică $H(S,a_k)$ și este cunoscută sub numele de *informație de divizare* (*split information*) pentru a_k și S:

$$SplitInfo(S,a_k) = H(S,a_k) = -\sum_{v \in Values(a_k)} \frac{|S_v|}{|S|} log \frac{|S_v|}{|S|} \qquad (4)$$

Plecând de la aceste observații, Quinlan (1986) propune o altă măsură pentru a alege cel mai potrivit predictor în raport cu modalitățile căruia să se divizeze setul de date S, în locul utilizării măsurii informationgain, $InfoGain(S,a_k)$.

Mai precis, se selectează acel predictor pentru divizarea lui S, pentru care se obține cel mai mare *raport al câștigului de informație* (*gainratio*) , adică:

$$GainRatio(S,a_k) = \frac{InfoGain(S,a_k)}{SplitInfo(S,a_k)} \qquad (5)$$

Alegerea predictorului pe baza căruia să se dividă setul de date S se poate face utilizând una dintre cele două măsuri prezentate: fie InfoGain, fie GainRatio, recomandabil ultima. Mai precis, se alege predictorul pentru care măsura respectivă are cea mai mare valoare.

- *Indexul Gini (Gini index)*

Breiman et al.(1984) a introdus pentru prima data criteriul (sau indexul) Gini și este inițial definit ca fiind probabilitatea de clasificare greșită a unui set de instanțe, sau mai degrabă ca impuritatea divizării. Indexul Gini al unui nod t este:

$$Gini(t) = 1 - \sum_{i=1}^{c} p_i^2 \tag{6}$$

De la acest index, ne putem referi la indexul Gini al divizării ca fiind:

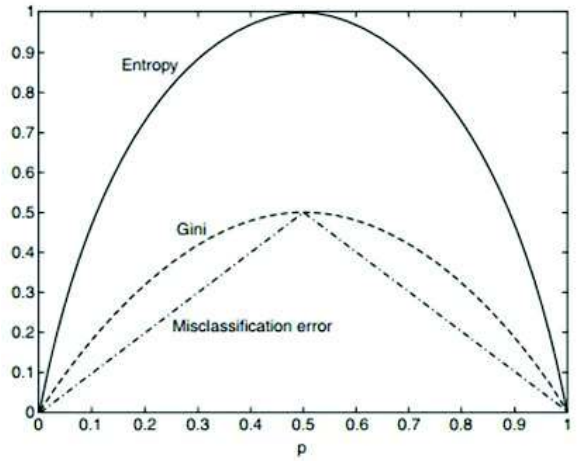
$$Gini(S) = \sum_{k=1}^{m} \frac{n_k}{n} \left[1 - \sum_{l=1}^{c} \left(\frac{n_{kl}}{n_k} \right)^2 \right] \tag{7}$$

- *Măsura de clasificare greșită (Misclassificationmeasure)*

Măsura de clasificare greșită măsoară eroarea în timpul clasificării unui nod atunci când se utilizează a anumită partiționare. Aceasta măsură are următoarea forma:

$$Misclassification_error(S) = 1 - \max_i p_i \tag{8}$$

Din formula (8) se observă că maximul erorii se obține în momentul în care obiectele din categorii difer sunt distribuite în nod în mod egal, obținându-se astfel cea mai mică informație. O comparație între entropie, indexul Gini și măsura de clasificare greșită pentru o problemă de clasificare binară este reprezentată grafic în figura de mai jos (a se vedea Figura 2).



Sursa: Tan et al.,2006

Figura 2. Comparație între măsurile impurității

unde p indică proporția obiectelor care aparțin uneia din cele 2 clase. Din figura 2 observăm că cele trei măsuri ating valoarea maximă în cazul în care distribuția clasei este uniformă (p = 0.5), iar valoarea minimă se atinge atunci când toate înregistrările aparțin aceleiași clase (când p = 0 sau p = 1), (Tan et al.,2006).

Fiind des folosite în problemele de clasificare este bine să menționăm faptul că arborii de decizie nu sunt indicați să fie utilizați, atunci când există clasificări cu multe clase.

Cu toate acestea, algoritmi eficienți au fost dezvoltați pentru a induce o acuratețe rezonabilă, utilizând o tehnică Greedy care este de fapt un procedeu recursiv de „sus în jos" (”divideandconquer”).

Obiectivul principal privind partiționarea optimă a unui nod este atinsă, în condițiile în care se cunoaște faptul că algoritmii Greedy identifică optimele locale în scopul obținerii optimului global.

Un astfel de algoritm pe care îl voi prezenta în continuare este algoritmul lui Hunt, care este la baza dezvoltării celorlalți algoritmi de construire a unui arbore de clasificare și decizie.

Algoritmul lui Hunt

În cele ce urmează, vom prezenta algoritmul lui Hunt (Tan et al.,2006), unde se consideră D_t mulţimea elementelor care se găsesc în nodul t, iar $y=\{y_1, y_2,..., y_c\}$ este mulţimea etichetelor claselor corespunzătoare nodului t.

În continuare, vor fi prezentați pașii algoritmului lui Hunt, iar pentru fiecare nod se va aplica același procedeu recursiv:

- pasul 1: dacă D_t conţine elemente care aparţin aceleiaşi clase notată cu y_t, atunci t va fi frunza etichetată cu y_t ;
- pasul 2: dacă D_t conţine elemente ce aparţin la mai mult de o clasă, atunci alegem un atribut test pentru a împărţi mulţimea D_t în submulţimi sau noduri mult mai mici.

Metoda ”divideandconquer”

Metoda “divideandconquer” (metoda ”dezbină şi cucereşte”) cu privire la inducția arborilor de decizie, uneori numită și inducția de sus în jos a arborilor decizionali, a fost dezvoltată și îmbunătățită de-a lungul multor ani de J. Ross Quinlan de la Universitatea Sydney.

Această abordare a problemei învățatului de la un set de situații independente duce, în mod normal, la un stil de reprezentare numit ”arbore de decizie”.

Nodurile dintr-un arbore de decizie implică testarea unui anumit atribut, de obicei, testul dintr-un nod compară valoarea atributului cu o constantă.

Cu toate acestea unii arbori de decizie compară două atribute unul cu celălalt, sau folosesc o anumită funcție a unuia sau a mai multor atribute.

Nodurile-frunză oferă o clasificare care se aplică tuturor situațiilor care ajung la frunză, sau un set de clasificări, sau o distribuție a probabilității peste toate clasificările posibile.

Pentru a clasifica o situație necunoscută, aceasta este direcționată descendent în arborele de decizie în funcție de valorile atributelor testate în noduri succesive, iar când se ajunge la o frunză, situația este clasificată în funcție de clasa desemnată frunzei.

Dacă atributul care este testat într-un nod este unul nominal, numărul de ramuri este, de obicei, numărul posibil de valori ale atributului.

În acest caz, pentru ca există câte o ramură pentru fiecare valoare posibilă, același atribut nu va mai fi testat mai jos de-a lungul arborelui.

Uneori, valorile atributelor sunt împărțite în două subseturi, iar arborele are doar două ramuri, în funcție de care valoare a subsetului este conținută de arbore; în acest caz, atributul poate fi testat de mai multe ori.

Dacă atributul este numeric, testul dintr-un nod determină dacă valoarea sa este mai mică sau mai mare decât o constantă predeterminată, oferind astfel o divizare în două.

Alternativ, o divizare în trei poate fi folosită, caz pentru care există mai multe posibilități. Dacă lipsa valorii este tratată ca o valoare de atribut în propriul său drept, acest lucru va crea o a treia ramură.

O alternativă pentru un atribut numeric întreg ar fi divizarea în trei astfel: mai mic decât, egal cu, mai mare decât.

O altă alternativă pentru un atribut numeric real, ar fi compararea valorii acestuia cu un interval, întrucât egalitatea cu o singură constantă nu ar fi importantă, astfel obținându-se trei clase: sub interval, inclus în interval, mai mare decât intervalul.

De obicei, un interval numeric este testat de mai multe ori de-a lungul căii descendente a arborelui, de la rădăcină la frunze, fiecare testare implicând o constantă diferită.

B. Metodede exprimare a condițiilor de testare a atributelor

Vom folosi setul de date despre vreme, un set mic, pe care îl vom utiliza pentru a ilustra clasificarea metodelor. Acesta face referire la condițiile prielnice pentru a juca un anumit joc. În general, situațiile într-un set de date sunt caracterizate de valoarea trăsăturilor sau atribute, care măsoară diferite aspecte ale situației.

În acest caz există patru atribute: Aspectul vremii (Outlook), Temperatura (Temperature), Umiditatea (Humidity) şi Vânt (Windy).Rezultatul este dacăjoci, respectiv, da (yes) sau nu (no).

Toate cele patru atribute au valori ce sunt mai degrabă categorii simbolice ce sunt prezentate în cea mai simplă formă în tabelul de mai jos (Tabelul 1), nu numere.

Aspectul vremii poate fi: însorit (sunny), înnorat (overcast) sau ploios (rainy); temperatura poate face referire la foarte cald (hot), moderat (mild) sau răcoros (cool); umiditatea poate să fie ridicată (high) sau normală (normal); iar vântul poate fi prezent sau absent. Acest lucru creează 36 de combinaţii posibile ($3 \cdot 3 \cdot 2 \cdot 2 = 36$), dintre care 14 sunt prezente în setul de exemple introdus.

Tabelul 1. Date meteorologice cu atribute simbolice

Aspectul vremii	Temperatura	Umiditatea	Vânt	Joc da/nu
însorit	cald	ridicată	absent	nu
însorit	cald	ridicată	prezent	nu
înnorat	cald	ridicată	absent	da
ploios	moderat	ridicată	absent	da
ploios	răcoros	normală	absent	da
ploios	răcoros	normală	prezent	nu
înnorat	răcoros	normală	prezent	da
însorit	moderat	ridicată	absent	nu
însorit	răcoros	normală	absent	da
ploios	moderat	normală	absent	da
însorit	moderat	normală	prezent	da
înnorat	moderat	ridicată	prezent	da
înnorat	cald	normală	absent	da
ploios	moderat	ridicată	prezent	nu

Două din atribute – temperatura şi umiditatea – au valori numerice sunt prezentate într-o formă mai complexă în tabelul de mai jos (a se vedea Tabelul 2). Acest lucru este numit problemă de atribut numeric – în acest caz, o problemă de atribut mixtă, pentru că nu toate atributele sunt numerice.

Tabelul 2. Date meteorologice cu atribute simbolice şi numerice

Aspectul vremii	Temperatura	Umiditatea	Vânt	Joc da/nu
însorit	85	85	absent	nu
însorit	80	90	prezent	nu
înnorat	83	86	absent	da
ploios	70	96	absent	da
ploios	68	80	absent	da
ploios	65	70	prezent	nu
înnorat	64	65	prezent	da
însorit	72	95	absent	nu
însorit	69	70	absent	da

ploios	75	80	absent	da
însorit	75	70	prezent	da
înnorat	72	90	prezent	da
înnorat	81	75	absent	da
ploios	71	91	prezent	nu

Metode de discretizare

Atributele numerice se pot converti în cele nominale, folosind o metodă de discretizare. Procedurile de discretizare care fac uz de valorile variabilei de decizie (etichetele de clasă) se numesc *metode de discretizare supervizată*.

Discretizarea se face în raport cu un anumit criteriu de partiționare. Măsurile bazate pe entropie au fost cele tradițional folosite. Un astfel de demers a fost introdus de Fayyad&Irani (1993), care utilizează cantitatea de informație bazată pe entropia de clasă a partițiilor candidate, în scopul selectării delimitatorilor dintre diviziunile partiției. Mai întâi se găsește un singur delimitator, ce minimizează media ponderată a entropiilor de clasă asociate celor două diviziuni ale partiției, iar apoi se aplică recursiv procedura de discretizare fiecăreia dintre diviziunile induse.

O euristică bazată pe MDLP este de asemenea utilizată pentru a controla numărul de intervale produse pentru partiția fiecărui domeniu.

Presupunem că nodul curent conține o mulțime de date T, formată din n cazuri (exemple). Fiecare domeniu al unui predictor continuu $a \in A$ se consideră inițial nepartiționat, adică format dintr-un singur interval $[a_{min}, a_{max}]$, ce acoperă întregul domeniu.

Dacă există n_a valori distincte ale predictorului respectiv în mulțimea curentă T de exemple, atunci vor exista $n_a - 1$ delimitatori *potențiali* ce ar putea fi folosiți pentru partiționarea domeniului $[a_{min}, a_{max}]$. În cazul unei discretizări dihotomice stricte, s-ar genera astfel o partiție de forma: $P_a = \{[a_{min}, t_a], (t_a, a_{max}]\}$.

Pentru a găsi delimitatorul t_a ce maximizează criteriul de divizare pentru predictorul a în nodul curent T, exemplele din T sunt sortate în raport cu valorile lui a pentru a obține o secvență ordonată de valori distincte: $\{a_1, a_1, \ldots, a_{n\,a}\}$.

Fiecare pereche de puncte adiacente sugerează un potențial delimitator $t_a = (a_i + a_{i+1})/2$.

Totuși, în afara păstrării doar a valorilor distincte, există și o altă cale de a reduce numărul potențial al delimitatorilor.

Fayyad a arătat că numai valorile lui a pentru care se trece de la o valoare a variabilei de decizie d la altă valoare (are loc o schimbare de clasă) pot fi utilizați ca delimitatori, deoarece numai pentru aceștia se obține informația maximă în clasificare.

Altfel spus, dacă a_i și a_{i+1} corespund aceleași clase, un delimitator plasat între ele nu poate conduce la o partiție cu un câștig de informație maxim.

Pentru exemplificare, voi folosi datele din tabelul 2 cu privire la predictorul ”Temperatură”. Secvența ordonată a valorilor acestui predictorşi etichetele de clasă corespunzând variabilei de decizie d (”joc golf”) sunt prezentate în tabelul de mai jos (Tabelul 3).

Tabelul 3. Secvența ordonată a predictorului ”Temperatura” și etichete de clasă pentru variabila de decizie d

64	65	68	69	70	71	72	72	75	75	80	81	83	85
da	nu	da	da	da	nu	nu	da	da	da	nu	da	da	nu

Pentru a găsi perechile adiacente ce pot fi utilizate la calculul potențialilor delimitatori, se reține doar valorile distincte, iar dintre acestea se vor folosi efectiv doar cele reprezentând granițe de clasă. În plus, valorile cu apariții repetate ce sunt asignate la clase diferite (cum este cazul valorii 72) vor fi și ele reținute. Pentru exemplul nostru se obțin 7 potențiali delimitatori (Tabelul 4).

Tabelul 4. Delimitatori

64	65	68	69	70	71	72	75	80	81	83	85
yes	no	yes	yes	yes	no	no	yes	no	yes	yes	no
	t_a^1	t_a^2			t_a^3		t_a^4	t_a^5	t_6		t_7
	64.5	66.5			70.5		73.5	77.0	80.5	84.0	

Numărul punctelor ce candidează pentru statutul de delimitator s-a redus astfel de la 13 la 7.

În general, vom admite că există k clase C_1, C_2, ..., C_k (în exemplul nostru $k=2$, deci C_1 =„ yes”, C_2 = “ no”). Fie $|C_j|/|T|$ proporția exemplelor din T ce aparțin clasei C_j. Incertitudinea reziduală în clasificare este dată de entropia claselor:

$$H(T) = -\sum_{j=1}^{k} |C_j|/|T| \cdot log_2\left(|C_j|/|T|\right)$$

adică: $H(T) = -\left(\frac{9}{14} log \frac{9}{14} + \frac{5}{14} log \frac{5}{14}\right) = 0.94029$, unde $|C_1| = 9$ („yes”), $|C_2| = 5$ („ no”), iar $|T| = 14$.

După ce mulțimea de exemple T este partiționată de un delimitator t_a^j, entropia informației de clasificare este exprimată ca medie ponderată a entropiilor de clasă rezultate:

$$H_{avg}(a,T) = \sum_{\ell=1}^{q} \frac{|T_\ell|}{|T|} H(T_\ell)$$

unde q este numărul de diviziuni ale partiției generate pentru predictorul a.

Delimitatorul t_a^j, ce va fi selectat pentru generarea partiției dintre toți delimitatorii candidați, va fi acela care îndeplinește una dintre următoarele condiții alternative (echivalente):

- minimizează $H_{avg}(a,T)$;
- maximizează $InfoGain\ (a,T) = H(T) - H_{avg}(a,T)$);
- maximizează $GainRatio(a,T) = \frac{InfoGain(a,T)}{SplitInfo(a,T)}$.

C. Construirea arborilor de decizie

Metoda care a fost descrisă folosind criteriul de câştigare al informației este aceeași cu cea ştiută drept algoritmul ID3. Utilizarea câştigului informației (gainratio) a fost una din multele îmbunătățiri care au fost aduse algoritmului ID3 de-a lungul anilor, Quinlan l-a descris drept stabil într-o varietate mare de circumstanțe. Deşi este o soluție stabilă şi practică, acesta sacrifică o parte din motivarea pur teoretică a criteriului de câştig informațional.

O serie de îmbunătățiri pentru ID3 a culminat cu un sistem practic şi influent în inducția arborilor de decizie, numit C4.5. Aceste îmbunătățiri includ metode care se ocupă de atributele numerice, valorile absente, datele fără sens şi generarea regulilor din arbori.

Problema construirii unui arbore de decizie poate fi exprimată în mod recursiv. În primul rând, se selectează un atribut care se plasează la nodul de la rădăcină şi se creează o ramură pentru fiecare valoare posibilă. Acesta va diviza setul de exemple în subseturi, unul pentru fiecare valoare a atributului.

Acum procesul poate fi repetat recursiv pentru fiecare ramură în parte, folosind doar acele exemple care chiar ating ramura. Dacă se întâmplă ca toate exemplele folosite într-un nod sa aibă aceeași clasificare, trebuie oprită dezvoltarea acelei părți a arborelui.

Singurul lucru care mai trebuie decis este cum să se determine care atribut trebuie divizat, dat fiind un set de exemple cu clase diferite.

Să luăm în considerare datele referitoare la vreme. Există patru posibilități pentru fiecare divizare și la nivelul cel mai înalt ele produc arbori (a se vedea Figura 3). Care este cea mai bună alegere? Numerele de clase ”da” și ”nu” sunt prezentate pe frunze.

Orice frunză cu doar o clasă – da sau nu – nu va trebui divizată mai departe, iar procesul recursiv în josul ramurii va lua sfârșit. Pentru că ne dorim arbori mici, am vrea ca acest lucru să se întâmple cât mai curând posibil. Dacă am avea o măsură pentru puritatea fiecărui nod, am putea alege atributul care produce cel mai pur nod.

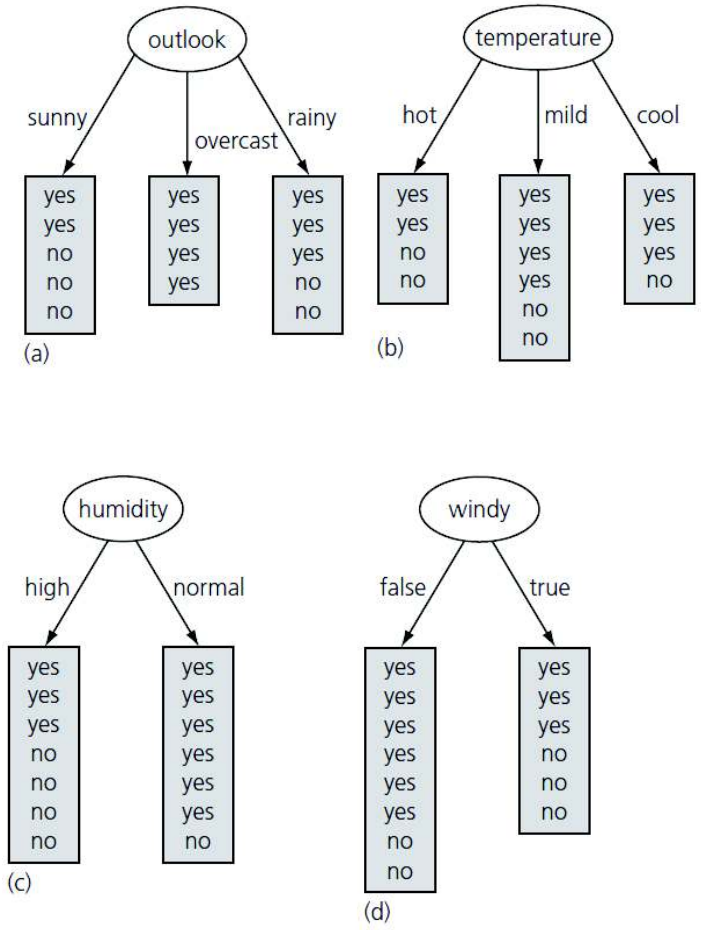


Figura 3.Posibilitățile de împărțire și ramurile binare pe care le produc pentru datele referitoare la vreme.

Unitatea de măsură a purității pe care o vom folosi este numită informație și este măsurată în unități numiți biți. Asociată cu un nod al arborelui, aceasta reprezintă cantitatea de informație ce se presupune că va fi necesară pentru a specifica dacă un nou exemplu trebuie clasificat ”da” sau ”nu”, dat fiind faptul că exemplul a ajuns la acel nod.

Spre deosebire de biții din memoria unui calculator, cantitatea de informație așteptată implică fracțiuni ale unui bit – și este adesea mai puțin de unu, deci îl calculăm pe baza numărului de clase ”da” și ”nu” la nod.

Când evaluăm primul arbore (a se vedea Figura 4) numărul claselor cu ”da” (yes) și „nu” (no) în nodurile frunză sunt [2,3], [4,0] și [3,2].

Valorile informațiilor pentru aceste noduri sunt: H([2, 3]) = 0.971 biți, H([4, 0]) = 0.0 biți, H([3, 2]) = 0.971 biți.

Putem calcula valoarea medie a informației acestora, luând în considerare numărul de exemple care sunt acceptate pe fiecare ramură, cinci pe prima și trei și patru pe cea de-a doua:H([2,3], [4,0], [3,2]) = (5/14)·0.971+(4/14)·0+(5 14)·0.971 = 0.693 biți.

Această medie reprezintă cantitatea de informație pe care o presupunem ca fiind necesară pentru a specifica clasa unui nou exemplu. Înainte să fi creat oricare din structurile în formare ale arborelui din figura3, exemplele de instruire de la rădăcină conțineau nouă noduri „da” și cinci noduri „nu”, corespunzând unei valori a informației de H([9,5]) = 0.940 biți.

Prin urmare arborele din figura 3.a) este responsabil pentru câștigul de informație.

InfoGain(outlook) = H ([9,5]) - H ([2,3], [4,0], [3,2]) = 0.940 - 0.693 = 0.247 biți, care poate fi interpretat ca valoarea informațională a crea o ramură pe atributul ”Aspectul vremii ” (outlook).

Deci, direcțiile de dezvoltare sunt clare.Vom calcula câștigul de informații pentru fiecare atribut și vom alege pe cel care câștigă cele mai multe informații pentru a-l împărți.

În situația din figura 3 avem InfoGain(aspectul vremii)=0.247 biți, InfoGain (temperatura)=0.029 biți, InfoGain (umiditatea)=0.152 biți, InfoGain (vânt) = 0.048 biți,
și selectăm ”aspectul vremii” (outlook) ca atribut de împărţire la rădăcina arborelui.

Să sperăm că aceasta se potriveşte cu intuiția ca fiind cea mai bună variantă. Este singura alegere în care un nod este complet pur, iar acest lucru oferă un avantaj considerabil față de celelalte atribute. Următoarea alegere este ”umiditatea” care este și cea mai bună alegere, pentru că produce un nod mai mare, care este aproape complet pur. Apoi continuăm în mod recursiv.

Posibilitățile pentru încă o ramură la nodul atins când atributul ”aspectul vremii” indică vreme însorită sunt prezentate în figura de mai jos (a se vedea Figura 4). În mod clar, încă o divizare a atributului ”aspectul vremii” nu va produce nimic nou, aşa că vom lua în considerare doar celelalte trei atribute: InfoGain (temperatura) = 0.571 biți;InfoGain (umiditatea) = 0.971 biți; InfoGain (vânt) = 0.020 biți.

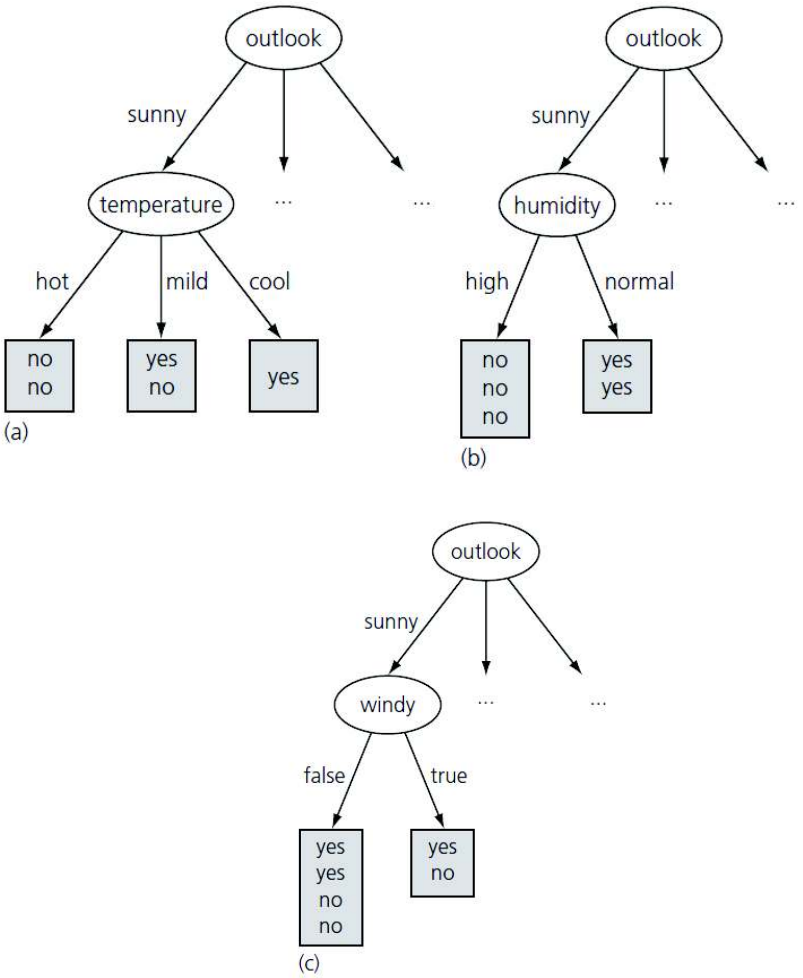


Figura4. Ramurile arborelui extinse pentru datele meteorologice

Deci, selectăm ”umiditatea” ca și atribut de divizare în acest punct. Nu este nevoie să mai împărțim aceste noduri mai departe, așa că ramura este terminată. Aplicarea continuă a aceleiași idei duce la arborele de decizie prezentate mai jos (a se vedea Figura 5) pentru datele meteorologice.

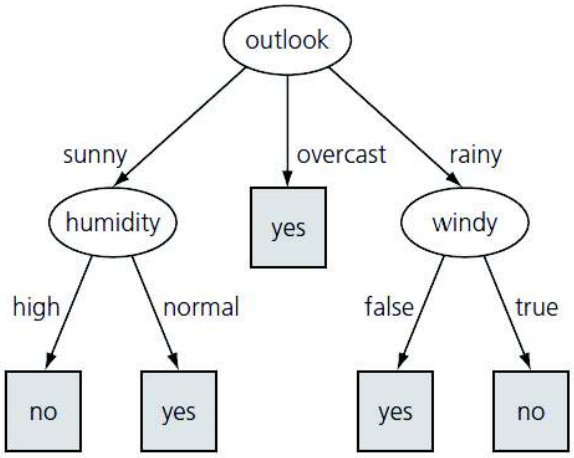


Figura 5. Arbore decizional pentru datele meteorologice.

În mod ideal, procesul ia sfârşit când toate frunzele sunt pure, adică atunci când ele conțin toate exemplele ce au aceeași clasificare. Totuși, s-ar putea ca această situație fericită să nu fie posibilă, pentru că nu există nimic care să oprească setul de formare ce conține două exemple cu seturi identice de atribute, dar cu clase diferite. Prin urmare, ne oprim atunci când datele nu pot fi divizate mai departe.

Bibliografie:

1.Breiman, L. Friedman, J.H. Olshen, R.A. & Stone, C.J.,1984. *ClassifficationandRegressionTrees*. Belmont, CA : Wadsworth International Group.

2.Fayyad, U.M. &Irani, K.B.,1993. *Multi-Interval Discretization of Continuous-ValuedAttributes for ClassificationLearning*. In Proceedings of the International JointConference on Uncertainty in AI (Q334 .I571 1993), pp. 1022-1027. Disponibil online la : http:// ijcai. org/Past%20Proceedings/IJCAI-93-VOL2/PDF/022.pdf

3.Hunt, E.B. Matin, J. & Stone, P.J.,1966.*Experiments in induction*.New York: Academic Press.

4.Quinlan, J.R.,1986.*Induction of decisiontrees*.Kluwer Academic Publishers, Boston, Machine Learning1:pp.81-106.Disponibil online la: http://hunch.net/~coms-4771/ quinlan.pdf

5.Tan, P. Steinbach, M. &Kumar, V.,2006.*Introduction to Data Mining*.Pearson Education,Ch.4, pp.145-205.Disponibil online la:http://www-users.cs.umn.edu /~kumar / dmbook/ch4.pdf.