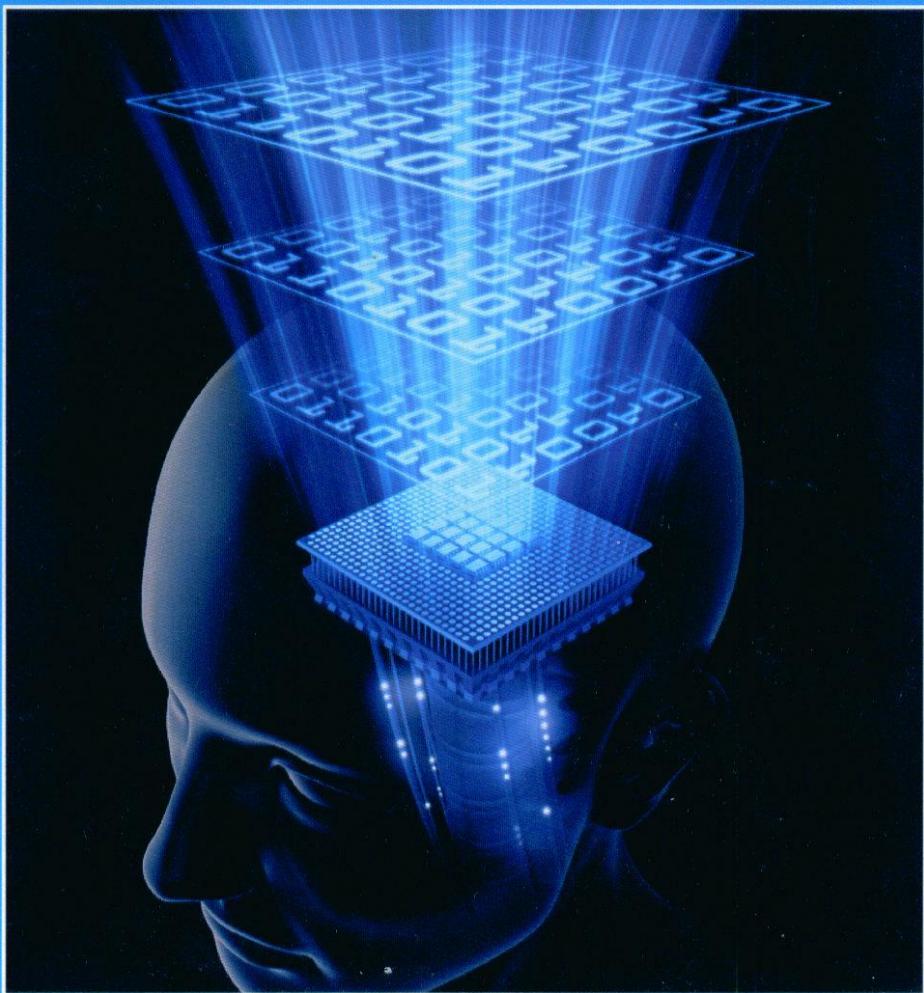


Florin Leon

Inteligentă artificială: raționament probabilistic, tehnici de clasificare



Florin Leon

Inteligentă artificială:
raționament probabilistic, tehnici de clasificare


Tehnypress
IAȘI – 2012

Referenți științifici:

Prof. dr. ing. Doru PĂNESCU

Prof. dr. ing. Silvia CURTEANU

Editura TEHNOPRESS

Str. Pinului nr. 1A

700109 Iași

Tel./fax: 0232 260092

E-mail: tehnopress@yahoo.com

<http://www.tehnopress.ro>

Editură acreditată CNCSIS, cod CNCSIS 89

Descrierea CIP a Bibliotecii Naționale a României

LEON, FLORIN

Inteligentă artificială: raționament probabilistic, tehnici de clasificare / Florin Leon. – Iași: Tehnopress, 2012

Bibliogr.

ISBN 978-973-702-932-4

*Familiei mele și în special soției mele, Crina,
fără de care această carte nu s-ar fi scris acum*

Cuprins

Partea I. Raționament probabilistic

Capitolul 1. Probabilități și paradoxuri

1.1. Interpretări ale probabilităților	11
1.1.1. Interpretarea frecventistă	11
1.1.2. Interpretarea fizică	13
1.1.3. Interpretarea subiectivistă	15
1.1.4. Probabilitățile în lumea cuantică	17
1.2. Paradoxuri	24
1.2.1. Problema „Monty-Hall”	24
1.2.2. Paradoxul cutiei lui Bertrand	27
1.2.3. Eroarea jucătorului de ruletă	27
1.2.4. Eroarea procurorului	27
1.2.5. Paradoxul lui Simpson	28

Capitolul 2. Fundamente teoretice

2.1. Probabilități condiționate. Teorema lui Bayes	31
2.2. Independență și independență condiționată	35
2.3. Rețele bayesiene	37
2.4. Algoritmul Bayes-Ball	42
2.5. Sortarea topologică	49
2.6. Construcția automată a rețelelor bayesiene	52

Capitolul 3. Raționamente exacte

3.1. Calculul probabilității unei observații	55
3.2. Calculul probabilităților marginale.....	57
3.3. Inferența prin enumerare.....	59
3.4. Inferența prin eliminarea variabilelor	62
3.5. Variabile cu valori multiple. Ignorarea variabilelor irelevante..	69
3.6. Cea mai probabilă explicație.....	71

Capitolul 4. Raționamente aproximative

4.1. Introducere	73
4.2. Inferența stochastică prin ponderarea verosimilității	74
4.3. Alte metode de inferență aproximativă	83

Capitolul 5. Teoria evidențelor

5.1. Teoria Dempster-Shafer	85
5.2. Surse multiple de evidență.....	91
5.3. Reguli alternative de combinare a evidențelor	96
5.3.1. Regula lui Yager	97
5.3.2. Regula Han-Han-Yang.....	100
5.4. Concluzii	104

Partea a II-a. Tehnici de clasificare

Capitolul 6. Problematica generală

6.1. Introducere	107
6.2. Învățarea supervizată	109
6.3. Definirea unei probleme de clasificare.....	113
6.4. Tipuri de atrbute.....	114
6.5. Estimarea capacitatei de generalizare.....	116
6.6. Aplicații ale tehniciilor de clasificare	119

Capitolul 7. Arbori de decizie

7.1. Algoritmul lui Hunt	121
7.2. Specificarea testelor de atribute	122
7.3. Măsuri de omogenitate	127
7.4. Partitionarea.....	130
7.5. Probleme cu atribute simbolice	131
7.6. Probleme cu atribute numerice	140
7.7. Aplicarea modelului.....	149
7.8. Erorile modelului.....	149
7.9. Câștigul proporțional	152
7.10. Alți algoritmi de inducție a arborilor de decizie	153
7.11. Concluzii	154

Capitolul 8. Clasificatorul bayesian naiv

8.1. Modelul teoretic	155
8.2. Probleme cu atribute simbolice	157
8.3. Considerente practice	160
8.3.1. Corectia Laplace.....	160
8.3.2. Precizia calculelor	164
8.4. Probleme cu atribute numerice	164
8.5. Concluzii	167

Capitolul 9. Clasificarea bazată pe instanțe

9.1. Introducere	169
9.2. Metrii de distanță.....	170
9.3. Scalarea atributelor.....	172
9.4. Calculul distanțelor pentru diferitele tipuri de atribute.....	173
9.5. Numărul optim de vecini	173
9.6. Blestemul dimensionalității.....	177
9.7. Ponderarea instanțelor.....	177

9.8. Ponderarea și selecția atributelor	178
9.9. Exemplu de clasificare	181
9.10. Concluzii	186
Referințe	187

Partea I

Rationament probabilistic

Probabilități și paradoxuri

1.1. Interpretări ale probabilităților

1.1.1. Interpretarea frecventistă

Mai întâi, vom menționa unele aspecte legate de natura probabilităților, interesante și din punct de vedere filosofic.

O definiție des întâlnită este următoarea: *probabilitatea* unui eveniment A reprezintă fracțiunea de lumi posibile în care A este adevărat (figura 1.1). Ca în teoria universurilor paralele, se consideră că se pot întâmpla *toate* posibilitățile, dar la un moment dat numai într-o fracțiune din „lumile posibile” respective se întâmplă un anumit eveniment.

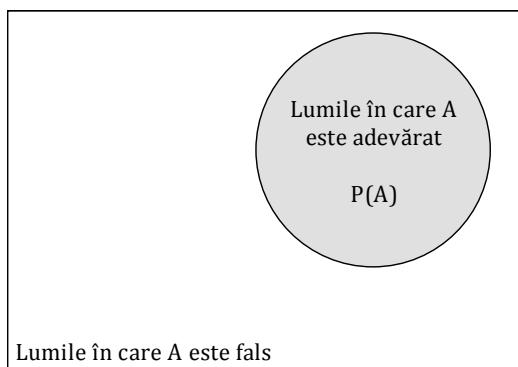


Figura 1.1. Ilustrarea interpretării frecventiste

Această definiție este legată de *interpretarea frecventistă*, care se reduce de fapt la organizarea unui experiment și la numărare: numărăm cazurile în care evenimentul este adevărat.

Dacă vrem să aflăm care este probabilitatea să se defecteze un calculator, numărăm câte calculatoare s-au defectat din numărul total de calculatoare și împărțim cele două valori.

Interpretarea frecventistă postulează că probabilitatea unui eveniment este frecvența sa relativă în timp, adică frecvența relativă de apariție după repetarea procesului de un număr mare de ori în condiții similare. Evenimentele se consideră guvernate de unele fenomene fizice aleatorii, fie fenomene predictibile în principiu, dacă am dispune de suficiente informații, fie fenomene impredictibile prin natura lor esențială.

Aruncarea unui zar sau învârtirea ruletei sunt exemple de fenomene predictibile în principiu, pe când descompunerea radioactivă este un exemplu de fenomen impredictibil. Descompunerea radioactivă este procesul prin care nucleul unui atom instabil pierde energie prin emiterea de radiație ionizantă. Emisia este spontană iar atomul se descompune fără alte interacțiuni fizice cu alte particule din afara sa. Descompunerea radioactivă este un proces stochastic la nivelul unui singur atom; conform teoriei mecanicii cuantice, este imposibil să se prezică momentul când va avea loc aceasta. Totuși, probabilitatea că un atom se va descompune este constantă în timp și deci pentru un număr mare de atomi identici, rata de descompunere a ansamblului este predictibilă, pe baza constantei de descompunere (sau a perioadei de înjumătățire).

În cazul aruncării unui ban corect, interpretarea frecventistă consideră că probabilitatea de a cădea „cap” este $1/2$ nu pentru că există 2 rezultate egal probabile, ci pentru că serii repetate de încercări au arătat în

mod empiric că frecvența converge la 1/2 când numărul de încercări tinde la infinit.

Mai formal, dacă n_a este numărul de apariții ale unui eveniment A după n încercări, atunci $P(A) = \lim_{n \rightarrow \infty} \frac{n_a}{n}$.

O problemă fundamentală în definirea frecventistă a probabilităților este următoarea. Limita unui sir infinit de încercări este independentă de segmentele sale inițiale finite. Dacă un ban cade „cap” de o sută de ori la rând, asta nu spune de fapt mai nimic despre probabilitatea de a cădea „cap” când numărul de încercări tinde la infinit. Este în mod evident imposibilă repetarea de un număr infinit de ori a unui experiment pentru a-i determina probabilitatea reală. Dar dacă se efectuează doar un număr finit de încercări, pentru fiecare serie de încercări va apărea o frecvență relativă diferită, chiar dacă probabilitatea reală ar trebui să fie aceeași întotdeauna. Dacă putem măsura probabilitatea doar cu o anumită eroare, această eroare de măsurare poate fi exprimată doar ca o probabilitate (însuși conceptul pe care dorim să-l definim). Prin urmare, definiția frecvenței relative devine circulară (Hájek, 2012).

1.1.2. Interpretarea fizică

Interpretarea fizică (engl. “propensity”) afirmă că probabilitățile sunt niște proprietăți ale obiectelor sau evenimentelor, predispoziții ale unui anumit tip de situații să producă anumite rezultate sau frecvențe relative pentru un număr mare de experimente. Predispozițiile nu sunt frecvențe relative, ci *cauze* ale frecvențelor relative stabile observate și ar trebui să explice *de ce* repetarea unui experiment generează anumite rezultate cu aproximativ aceleași rate de apariție.

Ne putem întreba dacă probabilitățile sunt niște caracteristici intrinseci ale obiectelor, asemănătoare cu proprietățile lor fizice, ca masa de exemplu. Dacă la aruncarea unui ban corect probabilitatea de a cădea pe fiecare din cele două fețe este 50% (banul poate să cadă și pe cant, dar aceasta este o problemă marginală, o excepție cu probabilitate foarte mică), care este cauza pentru procentele de 50-50%? De ce sunt egale probabile cele două fețe? Este aceasta o proprietate a banului ca obiect? Ar putea exista o altă zonă din univers, cu alte legi ale fizicii, în care această proporție să nu fie 50-50%?

Rezultatul unui experiment fizic este produs de o mulțime de condiții inițiale. Când repetăm un experiment, de fapt realizăm un alt experiment, cu condiții inițiale mai mult sau mai puțin similare. Un experiment determinist va avea de fapt întotdeauna o predispoziție de 0 sau 1 pentru un anumit rezultat. De aceea, predispoziții nebanele (diferite de 0 sau 1) există doar pentru experimente cu adevărat nedeterministe.

În această interpretare, rezultatul unui eveniment se bazează pe proprietățile fizice obiective ale obiectului sau procesului care generează evenimentul. Rezultatul aruncării unui ban se poate considera ca fiind determinat de exemplu de proprietățile fizice ale banului, cum ar fi forma simetrică plată și cele două fețe.

Este greu să definim probabilitățile ca predispoziții, pentru că (cel puțin deocamdată) nu știm ce sunt, ci doar ce (bănuim că) fac. Însă, aşa cum magnitudinea unei sarcini electrice nu poate fi definită explicit, folosind noțiuni elementare, ci doar în măsura efectelor pe care le produce (atragerea sau respingerea altor sarcini electrice), tot astfel predispoziția este ceea ce face ca un experiment să aibă o anumită probabilitate.

În acest context, legea numerelor mari reflectă faptul că frecvențele relative stable sunt o manifestare a predispozițiilor, adică a probabilităților invariante singulare (care se referă la evenimentul considerat în sine, nu la seria de încercări repetate). Pe lângă explicarea apariției frecvențelor relative stable, ideea de predispoziție este motivată de dorința de a înțelege probabilitățile singulare din mecanica cuantică, precum probabilitatea de descompunere a unui atom la un anumit moment (Hájek, 2012).

1.1.3. Interpretarea subiectivistă

Interpretarea frecventistă și cea fizică se mai numesc „obiectiviste”, deoarece presupun probabilitățile ca fiind componente ale lumii fizice. Discuția despre probabilitățile obiective are sens doar în contextul unor experimente aleatorii bine definite.

Interpretarea subiectivistă sau bayesiană consideră că probabilitatea unui eveniment este o măsură a convingerii subiective, personale, că evenimentul va avea loc.

Probabilități subiectiviste pot fi atribuite oricărei propoziții, chiar și atunci când nu este implicat niciun proces aleatoriu. Ele reprezintă gradul în care propoziția este sprijinită de evidențele disponibile. În general, aceste probabilități sunt considerate grade de încredere, care arată cât de siguri suntem că propoziția respectivă este adevărată.

Principiul indiferenței afirmă că atunci când există $n > 1$ posibilități mutual exclusive (distințe) și exhaustive colectiv (care acoperă toate posibilitățile), indistinctibile cu excepția denumirilor lor, fiecărei posibilități trebuie să i se atribuie o probabilitate egală cu $1 / n$ (Keynes, 1921).

Un ban simetric are două fețe, denumite arbitrar „cap” și „pajură”. Presupunând că banul va cădea pe o față sau pe alta, rezultatele aruncării banului sunt mutual exclusive, exhaustive și interschimbabile, dacă ignorăm evenimentul mai puțin probabil de a atteriza pe cant, deoarece acest rezultat nu este interschimbabil cu celelalte două. Conform principiului, fiecărei fețe i se atribuie probabilitatea de 1/2.

În această analiză se presupune că forțele care acționează asupra banului nu sunt cunoscute. Dacă le-am cunoaște cu precizie, am putea prezice traectoria banului conform legilor mecanicii clasice.

Interpretarea subiectivistă permite raționamentul cu propoziții a căror valoare de adevăr este incertă. Pentru a evalua probabilitatea unei ipoteze, trebuie specificate unele probabilități a-priori, care sunt actualizate apoi în lumina datelor relevante noi care apar.

În abordarea subiectivistă, probabilitatea măsoară o convingere personală, unei ipoteze i se atribuie o probabilitate, pe când în abordarea frecventistă, ipoteza este testată fără a i se atribui o probabilitate inițială.

Abordările obiectiviste sunt nepractice pentru cele mai multe probleme de decizie din lumea reală. În abordarea frecventistă, este necesar ca un proces să aibă o natură repetitivă pentru a-i putea fi măsurată probabilitatea. Aruncarea unui ban este un astfel de proces, însă incertitudinile privind un război nuclear de exemplu nu pot fi tratate astfel. Nu au existat războaie nucleare până acum și mai mult, repetarea lor este greu de imaginat. Pentru un astfel de proces complex care presupune analiza circumstanțelor care conduc la un război nuclear, este greu de realizat o estimare bazată pe considerente obiective (Lee, 1989).

Abordarea subiectivistă permite combinarea naturală a frecvențelor cu judecățile experților. Probabilitățile numerice pot fi extrase din baze de

date, pot fi estimate de către experți sau pot fi o combinație între cele două variante.

1.1.4. Probabilitățile în lumea cuantică

Dezvoltând discuția în domeniul mecanicii cuantice, putem considera extensia modelului de probabilități unidimensional, din lumea macroscopică, în care probabilitățile sunt numere reale pozitive și pentru o distribuție suma tuturor probabilităților este 1, la un model bidimensional, în care avem aşa-numitele *amplitudini de probabilitate*.

Amplitudinea de probabilitate este un număr complex al cărui modul ridicat la pătrat reprezintă o probabilitate. De exemplu, dacă amplitudinea de probabilitate a unei stări cuantice este $\alpha = \gamma + i\delta$, atunci probabilitatea de a măsura acea stare este $|\alpha|^2 = \gamma^2 + \delta^2$.

Dacă o particulă elementară poate avea 2 stări, notate $|0\rangle$ și $|1\rangle$, atunci ea poate fi simultan într-o *superpoziție* a acestor stări: $\alpha|0\rangle + \beta|1\rangle$, unde $|\alpha|^2 + |\beta|^2 = 1$. Dacă particula este observată, vom detecta starea $|0\rangle$ cu probabilitatea $|\alpha|^2$ și starea $|1\rangle$ cu probabilitatea $|\beta|^2$. De asemenea, starea particulei „colapsează” în starea observată, ori $|0\rangle$ ori $|1\rangle$.

Două experimente tipice pun în evidență comportamentul total diferit al elementelor mecanicii cuantice față de cele ale mecanicii clasice.

Primul presupune trimiterea de fotonii sau electronii perpendicular spre un perete, prin două fante care au mărimea și distanța dintre ele comparabile cu lungimea de undă a particulelor incidente. Chiar dacă doar o singură particulă elementară este emisă la un moment dat, pe perete apare un model de interferență, ca și cum fiecare particulă ar trece simultan prin

ambele fante și ar interfera cu ea însăși, după cum se poate vedea în figura 1.2 (BlackLight Power, 2005).

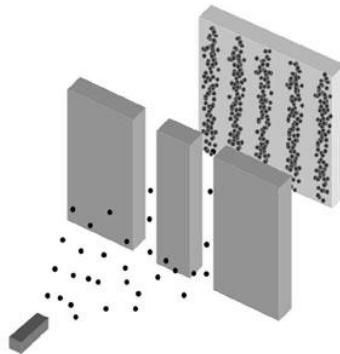


Figura 1.2. Comportamentul de tip undă

Dacă în dreptul fantelor se pune un detector, astfel încât să se determine exact prin ce fantă trece particula, rezultatul de pe perete apare ca două aglomerări distincte în dreptul fiecărei fante, fără modelul de interferență, ca în figura 1.3 (BlackLight Power, 2005). Prin acțiunea de observare, starea particulelor cuantice colapsează din superpoziție într-o stare „clasică”.

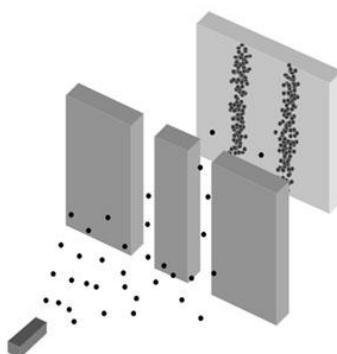


Figura 1.3. Comportamentul de tip particulă

În primul caz, particulele elementare se comportă ca niște unde, iar în al doilea caz se comportă ca niște particule. Este ca și cum rezultatul ar depinde de scopul măsurătorii, dacă dorim să detectăm particule sau unde.

Un alt experiment interesant care pune în evidență caracterul cuantic al particulelor elementare este interferometrul Mach-Zehnder (Harrison, 2008), prezentat în figura 1.4.

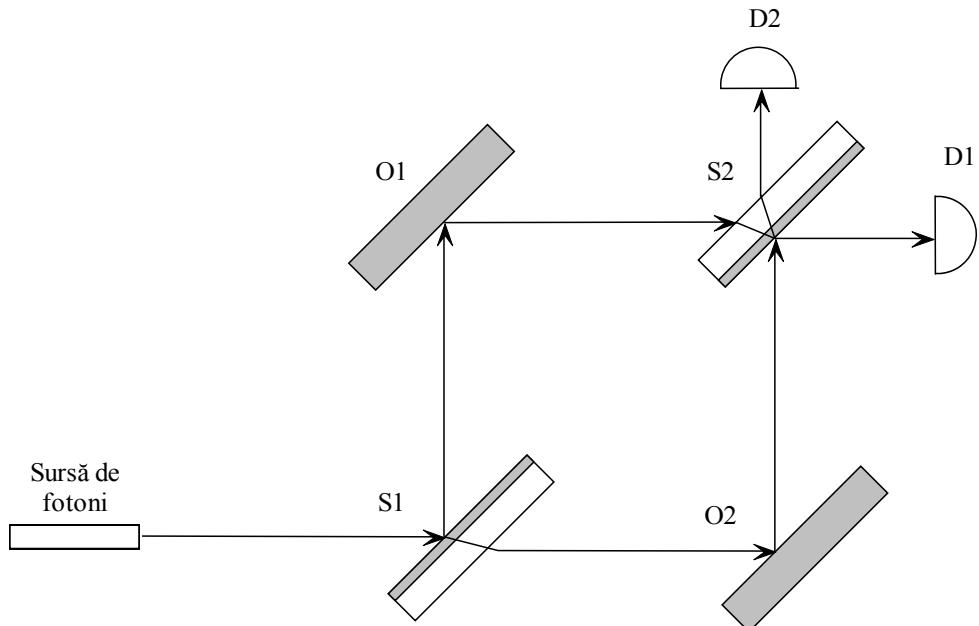


Figura 1.4. Interferometrul Mach-Zehnder

Acesta este un dispozitiv utilizat în general pentru a măsura interferența clasică. Este compus dintr-o sursă de fotoni, două oglinzi normale, O1 și O2, două oglinzi semitransparente S1 și S2 și două detectoare de particule D1 și D2. O oglindă semi-transparentă reflectă jumătate din lumina primită și refractă cealaltă jumătate prin ea.

La fiecare reflectare, fotonii își schimbă fază cu o jumătate de lungime de undă. Schimbări de fază au loc și la traversarea materialului

oglinzilor semitransparente. Ideea de bază este că schimbările de fază pentru ambele căi („sus” și „dreapta”) sunt egale în cazul detectorului D1 și prin urmare apare o interferență constructivă. În cazul detectorului D2, fotonii venind de pe cele două căi prezintă o diferență de fază de o jumătate de lungime de undă, ceea ce determină o interferență destructivă completă. În cazul clasic, toată lumina ajunge la detectorul D1 și deloc la detectorul D2.

În cazul cuantic, un foton traversează singur sistemul, însă apare același rezultat, ca și cum ar parcurge simultan cele două căi posibile și ar interfera cu el însuși.

Dacă am dori să observăm exact calea pe care merge fotonul, eliminând de exemplu oglinda S2 (figura 1.5), rezultatele devin cele clasice, jumătate din fotoni fiind detectați de către D1 și jumătate de către D2.

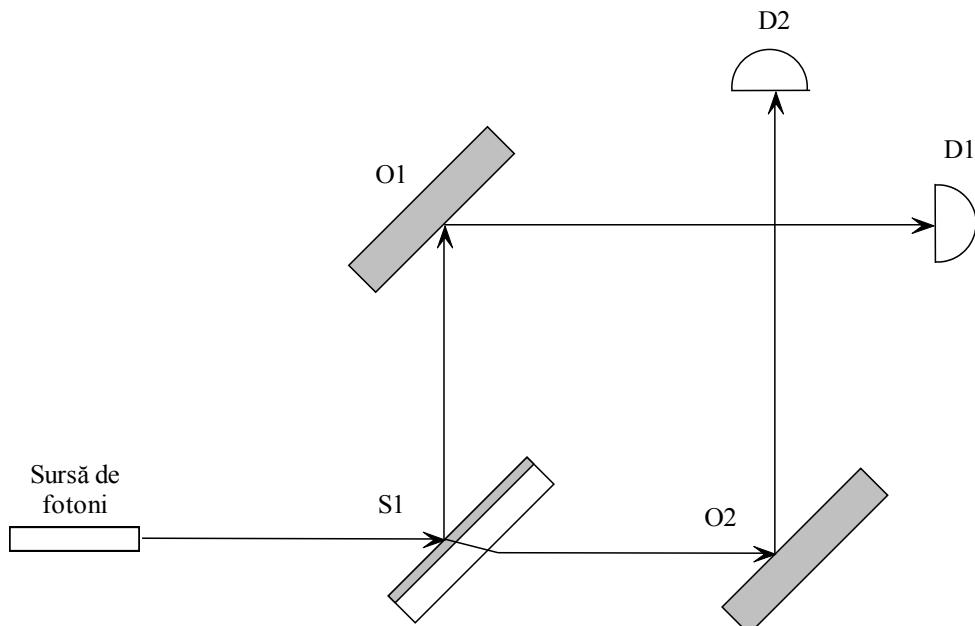


Figura 1.5. Configurația pentru observarea căii fotonului

Calculul cuantic este o direcție promițătoare de cercetare care încearcă să utilizeze fenomenele cuantice precum superpoziția (remarcată în experimentele anterioare prin faptul că particula elementară traversează simultan două căi) pentru a crește performanțele algoritmilor.

Unitatea de bază pentru informația cuantică este *qubit*-ul (engl. “quantum bit”), un sistem cuantic care poate avea 2 stări. Prelucrarea datelor se face aplicând aşa numitele *porți cuantice*, care transformă starea sistemului. Din punct de vedere matematic, aceste transformări sunt reprezentate ca niște matrice cu care se înmulțesc de obicei amplitudinile de probabilitate pentru a lua noi valori.

Să considerăm următorul exemplu (Aaronson, 2006), în care un qubit este inițial în starea $|0\rangle$, adică amplitudinea de probabilitate corespunzătoare stării $|0\rangle$ este 1 iar cea corespunzătoare stării $|1\rangle$ este 0. Vom mai considera transformarea dată de următoarea poartă cuantică, al cărei efect este rotirea unui vector cu 45° în sens antiorar (trigonometric):

$$U = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}. \quad (1.1)$$

Prin aplicarea acestei transformări, qubit-ul va ajunge într-o superpoziție:

$$\begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}, \quad (1.2)$$

în care, dacă qubit-ul este observat, stările $|0\rangle$ și $|1\rangle$ au probabilități egale de apariție, $1/2$. Operația este echivalentă aruncării unui ban.

Dacă însă mai aplicăm o dată aceeași transformare pentru starea rezultată, vom avea:

$$\begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (1.3)$$

ceea ce corespunde unei stări deterministe de $|1\rangle$.

Este ca și cum am da cu banul o dată și apoi, fără a vedea ce rezultat am obținut, mai aruncăm banul o dată și obținem un rezultat determinist.

Și în acest caz avem de-a face cu un fenomen de interferență, după cum putem vedea în figura 1.6 (Aaronson, 2006). Există două căi care conduc în starea $|0\rangle$, însă una din căi are o amplitudine pozitivă și cealaltă are o amplitudine negativă. Prin urmare, aceste amplitudini interferează destructiv și se anulează. Pe de altă parte, căile care conduc în starea $|1\rangle$ au ambele amplitudini pozitive și acestea interferează constructiv.

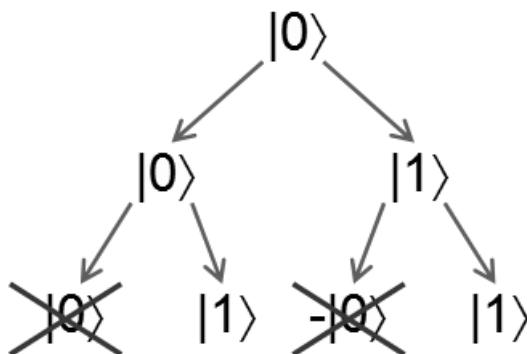


Figura 1.6. Interferența amplitudinilor de probabilitate

Acest tip de interferență nu se observă în lumea clasică deoarece probabilitățile nu pot fi negative. Anularea dintre amplitudinile pozitive și negative este unul din principalele aspecte care fac mecanica cuantică diferită de mecanica clasică.

Una din ideile de bază ale calculului cuantic este tocmai folosirea superpozițiilor și aplicarea unor transformări astfel încât rezultatele nedorite să se anuleze și, când sistemul este observat, să colapseze cu probabilitate cât mai mare într-o stare corespunzătoare rezultatului dorit.

Faptul că observarea unui sistem cuantic conduce la modificarea sa ridică numeroase probleme filosofice legate de rolul factorului conștiință în modelarea realității. Putem influența evenimentele aleatorii, schimbându-le probabilitățile?

În termodinamică, toate particulele au o mișcare browniană aleatorie, însă dacă ne ridicăm la un nivel superior, se pot observa comportamente deterministe. și în cazul oamenilor, interacțiunile acestora pot fi considerate aleatorii (deterministe sau nu, sunt oricum atât de complexe încât sunt probabil imposibil de modelat), însă din punct de vedere statistic se poate vedea cum evoluează o societate. Referindu-ne la modificarea probabilităților unor evenimente, noi suntem la nivelul microscopic, însă ar fi interesant dacă ne-am putea ridica la un nivel superior pentru a putea observa sau modifica sistemul.

Programul *Princeton Engineering Anomalies Research* (PEAR, 2010) a încercat între anii 1979-2007 un studiu experimental extins asupra interacțiunilor dintre conștiința umană și anumite dispozitive, multe bazate pe procese aleatorii, pentru a stabili măsura în care mintea poate influența realitatea fizică. După 28 de ani de investigații și mii de experimente cu milioane de încercări, rezultatul (controversat) a fost că, în medie, mintea

umană poate modifica 2-3 evenimente din 10000, dincolo de variațiile statistice normale.

1.2. Paradoxuri

Modul cum percep oamenii probabilitățile nu este perfect corect, întrucât nu există o capacitate înnăscută de a lucra cu acestea. La fel ca și în cazul logicii, oamenii au nevoie de un anumit antrenament pentru a înțelege toate subtilitățile raționamentelor probabilistice. Vom prezenta în cele ce urmează câteva paradoxuri, situații care conduc la concluzii neintuitive, dar care se dovedesc corecte în urma unei analize mai profunde.

1.2.1. Problema „Monty-Hall”

Problema „Monty-Hall” a fost popularizată în Statele Unite de o emisiune a canalului CBS din 1963. Problema este însă mai veche, fiind cunoscută într-o altă variantă încă de pe vasele cu zbaturi de pe Mississippi.

Concurrentul este pus în fața a trei uși, după cum se poate vedea în figura 1.7. În spatele a două uși este câte o capră iar în spatele unei uși este o mașină. Scopul jocului este desigur câștigarea mașinii. Jucătorul va câștiga ce se află în spatele ușii alese. Participantul trebuie să aleagă o ușă. Apoi, prezentatorul îi deschide o altă ușă, din celelalte două rămase, în spatele căreia este o capră. Apoi îl întreabă dacă vrea sau nu să își schimbe alegerea inițială către ușa a treia.

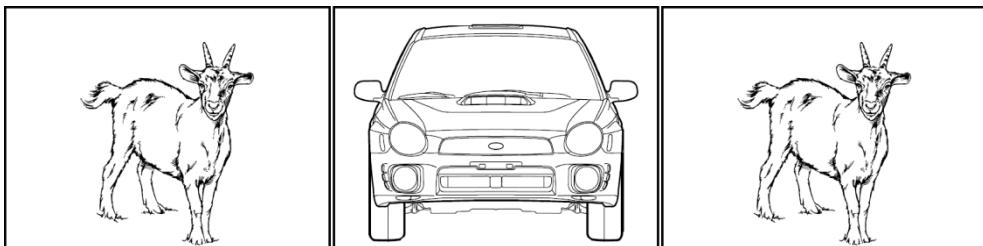


Figura 1.7. Problema „Monty-Hall”

Multe persoane consideră că probabilitatea inițială de câștig este $1/3$ iar dacă s-a deschis o ușă, probabilitatea devine acum $1/2$, indiferent de ușă aleasă. Având în vedere orgoliul propriu, faptul că inițial au ales ceva și nu vor să-și schimbe alegerea, probabilitățile părând oricum egale, ele au tentația de a nu-și schimba decizia.

Se poate vedea însă din figura 1.8 că prin schimbarea ușii, participantul are de două ori mai multe șanse de câștig. Intuitiv, prin faptul că prezentatorul i-a arătat o capră, i-a dat o informație, ceea ce schimbă probabilitățile. Ușa inițială a rămas cu probabilitatea de câștig de $1/3$, ceea ce este clar, dar faptul că i-a fost arătată o capră a transformat probabilitatea celelalte uși în $2/3$.

Să presupunem că jocul ar avea 100 de uși și prezentatorul i-ar fi deschis 98. Este clar că probabilitatea ușii inițiale este de $1/100$, iar restul până la 1 corespunde acum celeilalte uși.

Analizând toate deciziile posibile, dacă participantul a ales inițial mașina și își schimbă decizia, va pierde. Acesta este cazul defavorabil. În celelalte două situații însă, strategia de schimbare a deciziei este câștigătoare. Se vede că păstrarea alegerii inițiale are probabilitatea de câștig de $1/3$ iar schimbarea are probabilitatea de câștig de $2/3$.

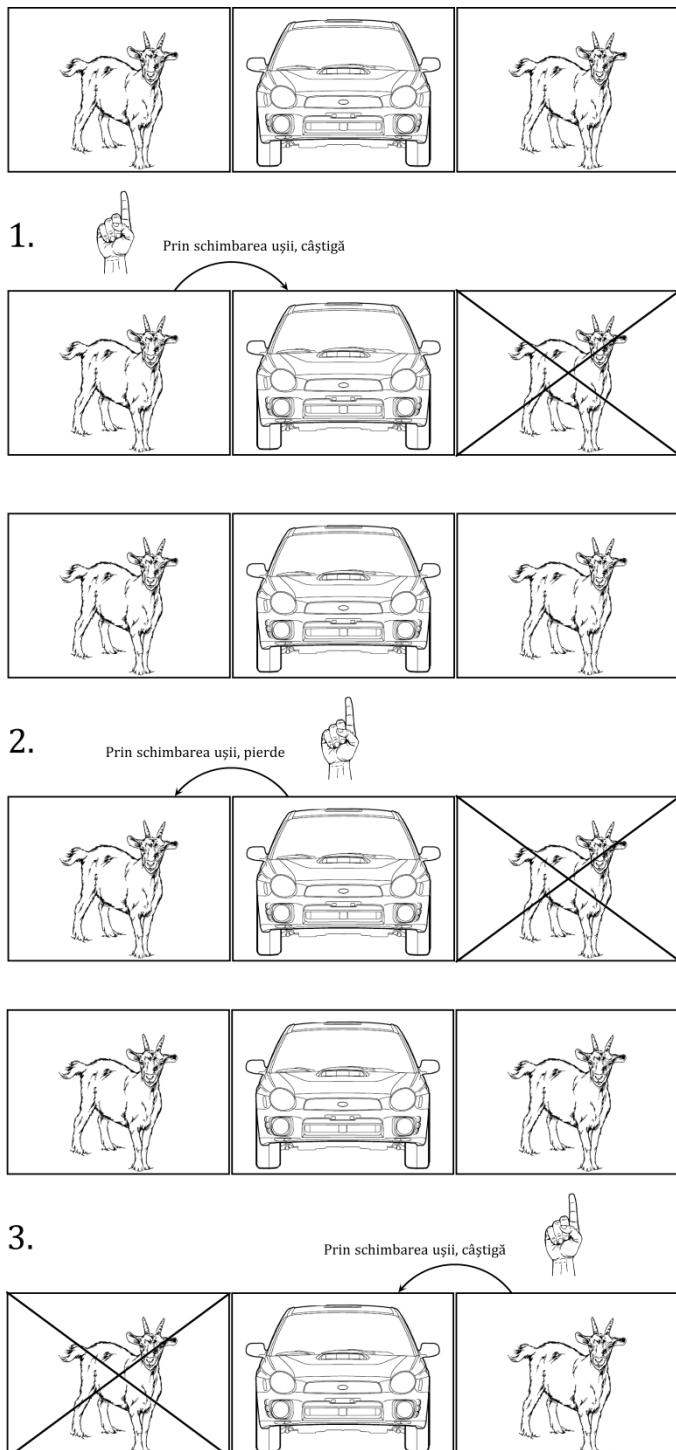


Figura 1.8. Situațiile posibile pentru problema „Monty-Hall”

1.2.2. Paradoxul cutiei lui Bertrand

Paradoxul cutiei lui Bertrand este echivalent din punct de vedere logic cu problema „Monty-Hall”. Există trei cutii al căror conținut nu este vizibil. Una conține două monede de aur, una conține două monede de argint iar a treia conține o monedă de aur și una de argint. După ce se alege o cutie în mod aleatoriu și se scoate o monedă, dacă aceasta este de aur, s-ar părea că probabilitatea ca moneda rămasă să fie tot de aur este de $1/2$. De fapt, probabilitatea este de $2/3$.

1.2.3. Eroarea jucătorului de ruletă

Un alt paradox este eroarea jucătorului de ruletă (engl. “gambler’s fallacy”). Culorile roșu și negru au fiecare probabilitatea de apariție de $1/2$. Să presupunem că a ieșit roșu de mai multe ori la rând. Eroarea este de a considera că, data următoare, negrul are mai multe șanse de apariție.

De fapt, fiecare experiment (acționare a ruletei) este independent. Prin urmare, de fiecare dată, roșul și negrul au aceleași probabilități de apariție. În 1913, la Monte Carlo, negrul a ieșit de 26 de ori la rând. Aceasta înseamnă că există o probabilitate foarte mică ca o culoare să iasă și de 100 de ori la rând, dar nu este imposibil. Toate aceste experimente succesive cu același rezultat, combinate, au o probabilitate foarte mică, însă fiecare experiment luat individual are aceleași probabilități.

1.2.4. Eroarea procurorului

Eroarea procurorului (engl. “prosecutor’s fallacy”) apare în situații

precum următoarea. Să considerăm că a fost prins un suspect de crimă, căruia i se fac analize de sânge pentru a fi comparate cu probele de la locul faptei. Grupa de sânge găsită la fața locului este o grupă rară, AB cu Rh negativ, care are 1% frecvență în populație. De asemenea, s-au mai găsit urme de păr blond, persoanele blonde constituind tot 1% din populație. Suspectul are grupa de sânge respectivă și este blond, prezența împreună a acestor trăsături având împreună probabilitatea de 0,01%. Prin urmare, probele ar indica faptul că suspectul este vinovat cu o probabilitate de 99,99%. Dacă însă orașul în care s-a petrecut crima are o populație de 100.000 de locuitori, înseamnă că, statistic, alți 10 oameni au aceleași trăsături și deci, probabilitatea suspectului de a fi vinovat este acum de doar 10%. Probabilitatea vinovăției a scăzut de la 99,99% la 10% doar luând în calcul această informație suplimentară.

Înseamnă că aceste probe sunt inutile? Nu, ele contează dacă sunt combinate cu alte probe, de exemplu o cameră video care identifică suspectul cu o probabilitate de 70%. În acest caz, probabilitățile de a fi nevinovat se înmulțesc: $0,1 \cdot 0,3 = 0,03$ și suspectul apare ca vinovat cu probabilitatea de 97%.

1.2.5. Paradoxul lui Simpson

Paradoxul lui Simpson (1951) nu se referă la probabilitățile elementare, ci la prelucrarea statistică a datelor. În general, se consideră că atunci când mulțimea de date este mai mare, concluziile trase sunt mai sigure, o consecință a legii numerelor mari din abordarea frecventistă a probabilităților. Paradoxul lui Simpson pare să infirme această euristică, demonstrând că este necesară foarte multă atenție atunci când mulțimi de

date mici se combină într-o mulțime de date mai mare. Uneori concluziile mulțimii mari sunt opuse concluziilor mulțimilor mici și în acest caz, de multe ori concluziile mulțimii mari sunt de fapt greșite.

Să considerăm un doctor care propune un tratament nou pentru o anumită afecțiune și dorește să îl compare cu un tratament standard din punct de vedere al timpului necesar pentru vindecare (scenariul este adaptat după un exemplu prezentat de Ooi, 2004). Cele două tipuri de tratament sunt aplicate unor bolnavi, iar rezultatele totale sunt prezentate în tabelul de mai jos. Pentru fiecare tratament și pentru fiecare rezultat, se indică numărul de pacienți aflați în situația respectivă.

Rezultat (timp de vindecare)	Tratament	
	Standard	Nou
Lung	2725 (55%)	3625 (80%)
Scurt	2275 (45%)	875 (20%)
Total	5000 (100%)	4500 (100%)

Din aceste date se poate trage concluzia că tratamentul nou pare clar inferior celui standard. 80% din pacienții supuși tratamentului nou au avut un timp lung de vindecare, comparativ cu 55% din pacienții supuși tratamentului standard. De asemenea, doar 20% din pacienții supuși tratamentului nou au avut un timp scurt de vindecare, comparativ cu 45% din pacienții care au primit tratamentul standard.

Să luăm acum în calcul următoarea informație: doctorul a făcut experimentul asupra unor bolnavi din Iași și respectiv din Bacău, în număr aproximativ egal. Însă doctorul, locuind în Iași, a avut mai mulți pacienți din acest oraș supuși noului tratament. Cei mai mulți pacienți din Bacău au primit tratamentul standard. Rezultatele detaliate sunt prezentate în tabelul următor.

Rezultat (timp de vindecare)	Tratament pentru pacienții din Iași		Tratament pentru pacienții din Bacău	
	Standard	Nou	Standard	Nou
Lung	475 (95%)	3600 (90%)	2250 (50%)	25 (5%)
Scurt	25 (5%)	400 (10%)	2250 (50%)	475 (95%)
Total	500 (100%)	4000 (100%)	4500 (100%)	500 (100%)

Se vede aici că pentru bolnavii din Iași tratamentul nou este mai bun: 90% față de 95% au avut un timp lung de vindecare și 10% față de 5% au avut un timp mai scurt de vindecare. O situație asemănătoare apare pentru bolnavii din Bacău: doar 5% față de 50% au avut un timp lung de vindecare și 95% față de 50% au avut un timp mai scurt de vindecare.

Practic, atunci când considerăm individual submulțimile de bolnavi din Iași și Bacău, tratamentul nou pare mai bun. Când considerăm mulțimea totală de bolnavi, tratamentul nou pare inferior celui standard.

Acest lucru ar fi echivalent cu următoarea situație. Dacă trebuie să recomandăm un tratament unui bolnav, fără să cunoaștem din ce oraș provine, îi vom recomanda tratamentul standard. Dacă știm din ce oraș provine, îi vom recomanda tratamentul nou. Însă cunoașterea sau nu a orașului nu ar trebui să aibă nicio influență asupra deciziei privind tratamentul.

Paradoxul este cauzat de combinarea unor grupuri inegale. În exemplul de mai sus, pacienții din Iași care au primit tratamentul standard sunt de 8 ori mai puțini decât cei care au primit tratamentul nou, iar pacienții din Bacău care au primit tratamentul nou sunt de 9 ori mai puțini decât cei care au primit tratamentul standard.

Soluția este proiectarea experimentelor astfel încât să nu se combine mulțimi de date de dimensiuni inegale, provenind din surse diferite (Rogers, 2001).

Fundamente teoretice

2.1. Probabilități condiționate. Teorema lui Bayes

Vom aminti acum câteva noțiuni legate de probabilitățile condiționate. Când trebuie să definim $P(A|B)$, presupunem că se cunoaște B și se calculează probabilitatea lui A în această situație. Adaptând un exemplu al lui Moore (2001), să considerăm evenimentul D (durere de cap) și să presupunem că are, în general, o probabilitate de 1/10. Probabilitatea de a avea gripă (evenimentul G) este de numai 1/40. După cum se vede în figura 2.1, dacă cineva are gripă, probabilitatea de a avea și dureri de cap este de 1/2. Deci probabilitatea durerii de cap, dată fiind gripe, este de 1/2. Această probabilitate corespunde intersecției celor două regiuni, cu aria egală cu jumătate din G .

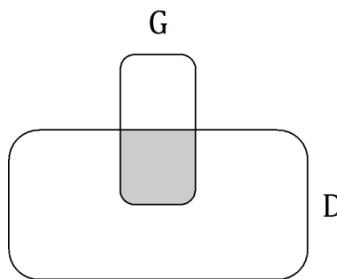


Figura 2.1. Reprezentare grafică a unei probabilități condiționate

Pe baza acestei relații rezultă *teorema lui Bayes* (1763), care este importantă pentru toate raționamentele probabilistice pe care le vom studia.

Considerăm formula probabilităților condiționate:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \quad (2.1)$$

Putem exprima probabilitatea intersecției în două moduri și de aici deducem expresia lui $P(B|A)$ în funcție de $P(A|B)$:

$$P(A \cap B) = P(A|B) \cdot P(B), \quad (2.2)$$

$$P(A \cap B) = P(B|A) \cdot P(A), \quad (2.3)$$

$$\Rightarrow P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}. \quad (2.4)$$

Această ecuație reprezintă un rezultat fundamental. Mai clar, putem considera următoarea expresie alternativă:

$$P(I|E) = \frac{P(E|I) \cdot P(I)}{P(E)}, \quad (2.5)$$

unde I este ipoteza, E este evidența (provenind din datele observate), $P(I)$ este *probabilitatea a-priori* a ipotezei, adică gradul inițial de încredere în ipoteză, $P(E|I)$ este *verosimilitatea* datelor observate (engl. “likelihood”), adică măsura în care s-a observat evidența în condițiile îndeplinirii ipotezei, iar $P(I|E)$ este *probabilitatea a-posteriori* a ipotezei, dată fiind evidența.

Relația este importantă deoarece putem calcula astfel probabilitățile cauzelor, date fiind efectele. Este mai simplu de cunoscut când o cauză

determină un efect, dar invers, când cunoaștem un efect, probabilitățile cauzelor nu pot fi cunoscute imediat. Teorema ne ajută să diagnosticăm o anumită situație sau să testăm o ipoteză.

În ecuația 2.4, se poate elimina $P(B)$ folosind *regula probabilității totale* (engl. “law of total probability”). $P(B)$ va fi exprimat astfel:

$$P(B) = \sum_j P(B|A_j)P(A_j). \quad (2.6)$$

Expresia poate fi înțeleasă mai ușor observând situația din figura 2.2.

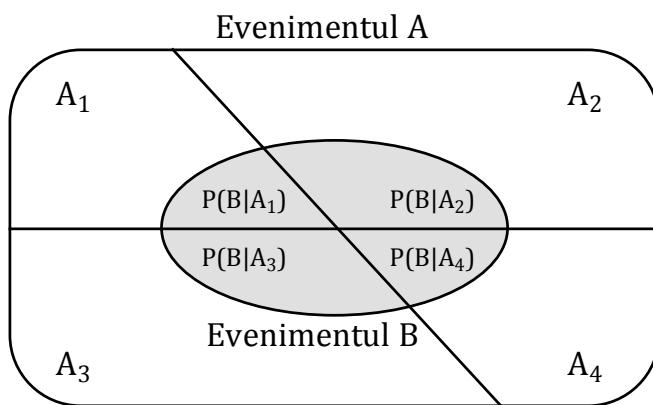


Figura 2.2. Reprezentare grafică a legii probabilității totale

Probabilitatea fiecărei valori A_i a lui A , condiționată de B , este:

$$P(A_i|B) = \frac{P(B|A_i) \cdot P(A_i)}{\sum_j P(B|A_j)P(A_j)}. \quad (2.7)$$

Să considerăm următorul exemplu de diagnosticare. Știm că probabilitatea de apariție a meningitei în populația generală este

$P(M) = 0,002\%$. De asemenea, probabilitatea ca o persoană să aibă gâtul înțepenit este $P(G) = 5\%$. Mai stim că meningita cauzează gât înțepenit în jumătate din cazuri: $P(G|M) = 50\%$.

Dorim să aflăm următorul lucru: dacă un pacient are gâtul înțepenit, care este probabilitatea să aibă meningită?

Aplicând teorema lui Bayes, vom avea:

$$P(M|G) = \frac{P(G|M) \cdot P(M)}{P(G)} = 0,02\%.$$

G este un simptom pentru M . Dacă există simptomul, care este probabilitatea unei posibile cauze, adică $P(M)$? Rezultatul este $0,02\%$, deci o probabilitate mică, deoarece probabilitatea meningitei însăși este foarte mică în general.

Acest rezultat este important la diagnosticarea bolilor rare. Testele au și ele o marjă de eroare, foarte mică, dar ea există. Aceasta, coroborată cu probabilitatea mică a bolii, nu conduce automat la concluzia că persoana are boala respectivă, chiar dacă testul iese pozitiv.

În exemplul următor, B reprezintă boala, iar T reprezintă rezultatul pozitiv al testului. Să presupunem că:

$$P(B) = 0,01$$

$$P(T|B) = 0,99$$

$$P(T|\neg B) = 0,02.$$

Conform expresiei 2.7, putem calcula probabilitatea bolii dacă testul a ieșit pozitiv:

$$P(B|T) = \frac{P(T|B) \cdot P(B)}{P(T)} = \frac{P(T|B) \cdot P(B)}{P(T|B) \cdot P(B) + P(T|\neg B) \cdot P(\neg B)} = 0,33$$

Prin urmare, chiar dacă testul iese pozitiv, probabilitatea de a avea boala este de doar 33%. În general, trebuie evitată greșeala de a considera că $P(A|B) = P(B|A)$. Se poate vedea chiar din exemplul anterior că $P(B|T) \neq P(T|B)$.

2.2. Independență și independență condiționată

Vom prezenta în continuare câteva exemple în care vom explica noțiunile de *independență* și *independență condiționată*.

Să presupunem că Ion și Maria dau cu banul. Ion dă cu un ban și Maria dă cu alt ban. Aceste evenimente nu se influențează în niciun fel. Dacă Ion dă cu banul, acest lucru nu aduce nicio informație asupra rezultatului acțiunii Mariei. În acest caz, evenimentele sunt independente, deoarece rezultatul unui experiment nu influențează celălalt experiment.

În cazul în care dau cu același ban, dacă Ion dă de 100 de ori și iese de 70 de ori pajură, ceea ce înseamnă că banul nu este corect, acest rezultat dă informații asupra experimentului Mariei. Dacă Maria va da cu banul de 100 de ori, probabil rezultatul va fi similar. Cele două evenimente nu mai sunt independente.

În schimb, dacă un expert analizează banul și constată că este măsluit, atunci experimentul lui Ion (70% pajură) nu mai aduce nicio informație asupra experimentului Mariei (tot 70% pajură). Rezultatul se poate prevedea datorită analizei expertului. Experimentele lui Ion și al

Mariei devin independente dată fiind noua evidență, a faptului că banul este măsluit.

Fie variabila A rezultatul experimentului lui Ion și B rezultatul experimentului Mariei. Fie C variabila „banul este influențat în favoarea pajurei”. În ultimul exemplu, se spune că A și B sunt independente condițional, dat fiind C .

În unele situații, independența condițională poate fi ușor confundată cu independența. Se presupune că evenimentele sunt independente, când de fapt sunt doar independente condițional.

De exemplu, Ion și Maria locuiesc în zone diferite ale orașului și vin la serviciu cu tramvaiul, respectiv cu mașina. Se poate considera că dacă întârzie unul din ei, nu se poate spune nimic despre întârzierea celuilalt, sunt evenimente independente. Dar să presupunem că la un moment dat aflăm că vatmanii sunt în grevă. Ion întârzie deoarece nu merg tramviale. Însă datorită grevei, probabil traficul auto crește, ceea ce o afectează pe Maria. Astfel, întârzierea lui Ion nu mai este independentă de întârzierea Mariei. Ele sunt independente condițional, dat fiind evenimentul grevei vatmanilor, care este cauza lor comună.

Să mai considerăm cazul în care o răceală poate detemina o persoană să strănuie, dar și o alergie poate determina același lucru. În general, dacă nu știm că persoana a strănutat, răceala și alergia sunt evenimente independente. Dacă știm însă că persoana strănută, aceste evenimente nu mai sunt independente. De exemplu, dacă mai știm că persoana este răcită, probabil că răceala determină strănutul și deci probabilitatea de a avea și alergie se reduce. Informații suplimentare privind răceala modifică probabilitatea alergiei.

2.3. Rețele bayesiene

În continuare, ne vom concentra asupra reprezentării informațiilor legate de evenimente probabilistice, care ne va ajuta să realizăm eficient raționamente.

Determinarea probabilității unei combinații de valori se poate realiza astfel:

$$P(x_1, \dots, x_n) = P(x_n|x_{n-1}, \dots, x_1) \cdot P(x_{n-1}, \dots, x_1). \quad (2.8)$$

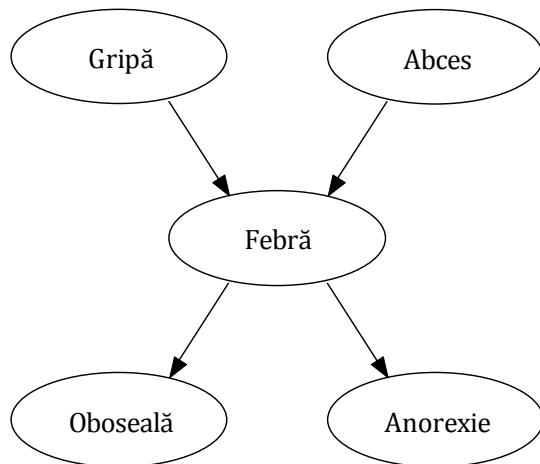
Aplicând în continuare această regulă vom obține *regula de înmulțire a probabilităților* (engl. “chain rule”):

$$\begin{aligned} P(x_1, \dots, x_n) &= P(x_n|x_{n-1}, \dots, x_1) \cdot P(x_{n-1}|x_{n-2}, \dots, x_1) \cdot P(x_2|x_1) \\ &\quad \cdot P(x_1), \end{aligned} \quad (2.9)$$

exprimată mai concis astfel:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i|x_{i-1}, \dots, x_1). \quad (2.10)$$

O rețea bayesiană arată ca în figura 2.3: este un graf orientat aciclic (engl. “directed acyclic graph”), în care evenimentele sau variabilele se reprezintă ca noduri, iar relațiile de corelație sau cauzalitate se reprezintă sub forma arcelor dintre noduri.

**Figura 2.3.** Rețea bayesiană

În acest exemplu, se consideră că atât gripa cât și abcesul pot determina febra. De asemenea, febra poate cauza o stare de oboseală sau lipsa poftei de mâncare (anorexie).

Sensul săgețiilor arcelor sunt dinspre părinți, cum ar fi gripa și abcesul, înspre fii, precum febra. Deși în acest exemplu relațiile sunt cauzale, în general o rețea bayesiană reflectă relații de corelație, adică măsura în care aflarea unor informații despre o variabilă-părinte aduce noi informații despre o variabilă-fiu.

Condiția ca o rețea bayesiană să fie un graf orientat aciclic înseamnă că arcele pot forma bucle, dar nu pot forma cicluri, după cum se vede în figura 2.4.

Având în vedere că determinarea probabilităților nodurilor presupune de fapt înmulțiri și adunări în care sunt implicați părinții variabilelor, dacă rețeaua ar permite cicluri, acest lucru ar conduce la repetarea la infinit a unor calcule.

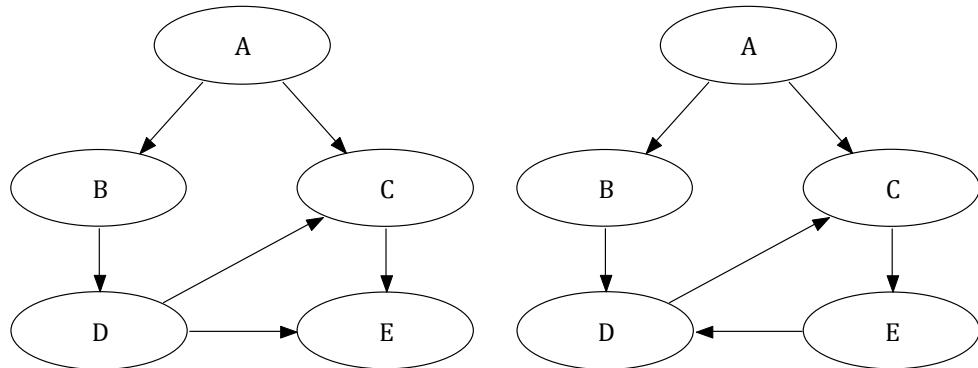


Figura 2.4. a) Rețea bayesiană validă; b) Rețea bayesiană invalidă

Fiecare variabilă are o mulțime de valori. În cazul cel mai simplu, variabilele au valori binare, de exemplu *Da* și *Nu*. În general însă, o variabilă poate avea oricâte valori.

Tabelul 2.1. Tabelele de probabilități pentru rețeaua bayesiană

$P(Gripă = Da)$	$P(Gripă = Nu)$
0,1	0,9

$P(Abces = Da)$	$P(Abces = Nu)$
0,05	0,95

<i>Gripă</i>	<i>Abces</i>	$P(Febră = Da)$	$P(Febră = Nu)$
Da	Da	0,8	0,2
Da	Nu	0,7	0,3
Nu	Da	0,25	0,75
Nu	Nu	0,05	0,95

<i>Febră</i>	$P(Oboseală = Da)$	$P(Oboseală = Nu)$
Da	0,6	0,4
Nu	0,2	0,8

<i>Febră</i>	$P(Anorexie = Da)$	$P(Anorexie = Nu)$
Da	0,5	0,5
Nu	0,1	0,9

Asociate cu variabilele, o rețea bayesiană conține o serie de tabele de probabilități, precum cele din tabelul 2.1. Pentru nodurile fără părinți se indică probabilitățile marginale ale fiecărei valori (adică fără a lăua în considerare valorile celorlalte variabile). Pentru celelalte noduri, se indică probabilitățile condiționate pentru fiecare valoare, ținând cont de fiecare combinație de valori ale variabilelor părinte.

În general, o variabilă binară fără părinți va avea un singur parametru independent, o variabilă cu 1 părinte va avea 2 parametri independenți iar o variabilă cu n părinți va avea 2^n parametri independenți în tabela de probabilități corespunzătoare.

Presupunerea modelului bazat pe rețele bayesiene este că o variabilă nu depinde decât de părinții săi și deci ecuația 2.10 devine:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \pi(x_i)), \quad (2.11)$$

unde $\pi(x_i)$ reprezintă mulțimea părinților variabilei x_i , din sirul ordonat al nodurilor în care părinții unui nod apar întotdeauna înaintea nodului respectiv. Modul în care se poate determina acest sir va fi tratat în secțiunea 2.5 referitoare la sortarea topologică.

Dacă avem n variabile binare, distribuția comună de probabilitate conține câte o probabilitate pentru toate 2^n combinații. Însă întrucât suma probabilităților este 1, ultima combinație nu mai este independentă de celelalte și poate fi dedusă ca 1 minus suma celorlalte. Distribuția comună este deci specificată de $2^n - 1$ parametri.

După cum se vede în tabelele de probabilități din tabelul 2.1, pentru exemplul din figura 2.3 cu 5 variabile sunt necesari numai 10 parametri față

de $2^5 - 1 = 31$. Toate variabilele fiind binare, pentru o anumită combinație de valori a părinților, probabilitatea unei valori a unui fiu este 1 minus probabilitatea celeilalte valori, de exemplu:

$$P(\text{Oboseală} = \text{Da}) = 1 - P(\text{Oboseală} = \text{Nu}).$$

Într-o rețea bayesiană, în loc să calculăm probabilitățile tuturor elementelor din distribuția comună de probabilitate, considerăm numai probabilitățile condiționate de părinți.

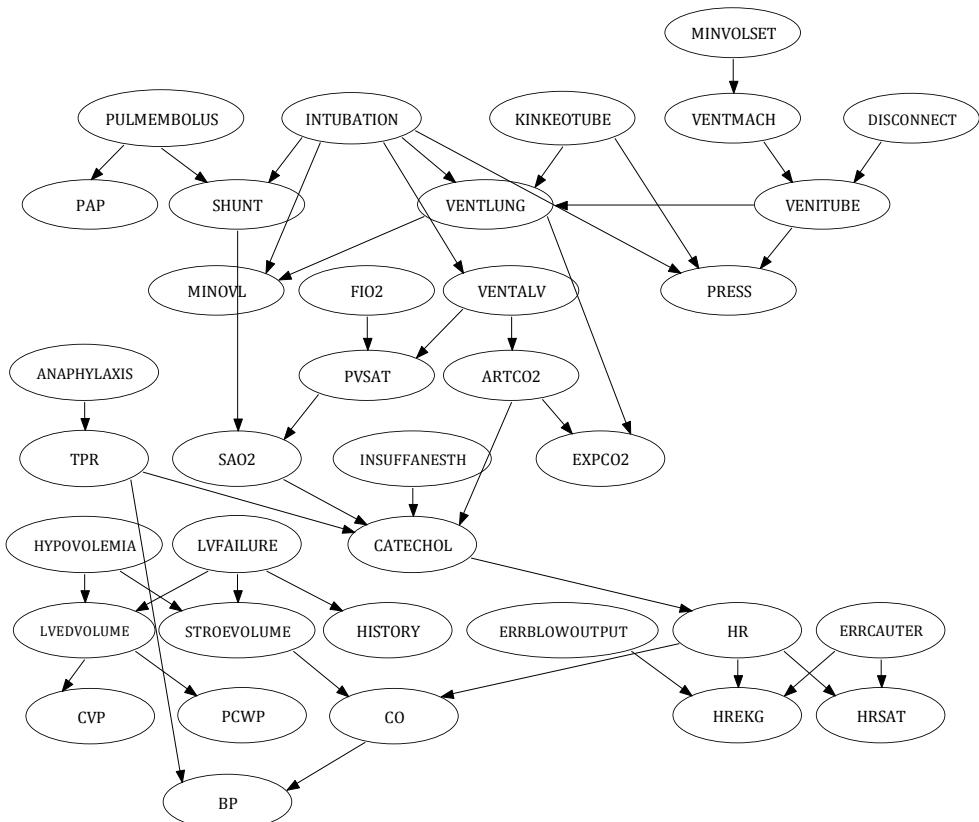


Figura 2.5. Rețea bayesiană complexă

Un alt exemplu privind un sistem de monitorizare a pacienților la terapie intensivă (Russell & Norvig, 2002) cu 37 de variabile, prezentat în figura 2.5, surprinde reducerea clară a numărului de parametri de la $2^{37} - 1 \approx 10^{11}$ la 509.

Reducerea complexității calculelor de probabilități este unul din scopurile principale ale rețelelor bayesiene.

2.4. Algoritmul Bayes-Ball

Pe baza structurii unei rețele bayesiene, putem determina și relațiile de independență sau dependență condiționată dintre noduri.

În exemplele din secțiunea 2.2, am menționat două situații în care se poate preciza relația de independență între evenimente, în prezența sau lipsa unor evidențe. Putem relua acum aceste situații în reprezentarea grafică a rețelor bayesiene.

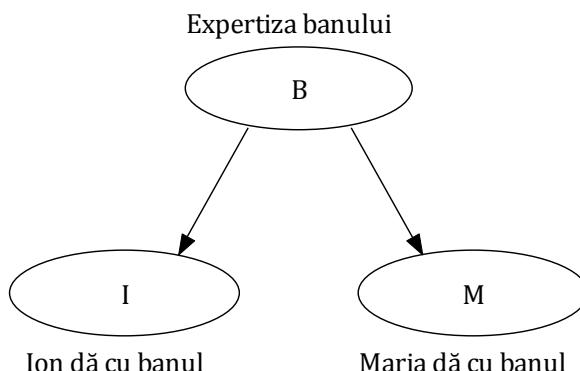


Figura 2.6. Cauză comună

În figura 2.6 este prezentat scenariul *cauzei comune*. Dacă lipsesc alte evidențe, I și M nu sunt independente, deoarece au o cauză comună ascunsă. Dacă se cunoaște însă B , ele devin independente.

Formal, putem scrie că:

$$P(I|M) \neq P(I), \text{ notat și } I \not\perp\!\!\!\perp M$$

$$P(I|M, B) = P(I|B), \text{ notat și } I \perp\!\!\!\perp M | B$$

În figura 2.7 este prezentat scenariul *revocării prin explicare* (engl. “explaining away”). R și A sunt independente în lipsa altor evidențe. Dacă se cunoaște însă S , ele nu mai sunt independente.

Formal, putem scrie că:

$$P(R|A) = P(R), \text{ notat și } R \perp\!\!\!\perp A$$

$$P(R|A, S) \neq P(R|S), \text{ notat și } R \not\perp\!\!\!\perp A | S$$

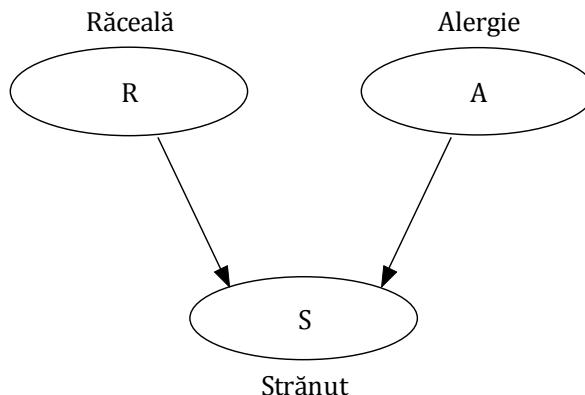


Figura 2.7. Revocare prin explicare

O modalitate simplă pentru a deduce relațiile de independență și independență condiționată între noduri este propusă de *algoritmul Bayes-Ball* (Shachter, 1998), care prin analogie cu jocul de baseball, presupune trimiterea unei mingi în rețea, care poate trece mai departe sau poate fi

blocată de anumite noduri. Dacă mingea nu poate ajunge de la un nod A la un nod B , atunci aceste noduri sunt independente.

Regulile de mișcare a mingii prin rețea iau în calcul direcțiile arcelor (dacă mingea vine de la un părinte la un fiu sau invers), precum și tipurile de noduri: neobserve sau observe (de evidență). În figura 2.8 (Paskin, 2003), săgețile normale arată faptul că mingea trece mai departe, iar barele perpendiculare pe capetele săgeților indică faptul că mingea este blocată.

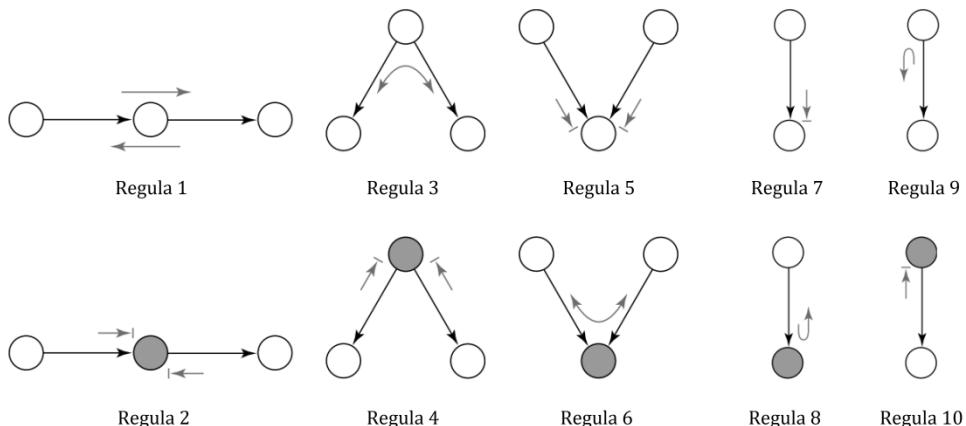


Figura 2.8. Regulile de traversare a rețelei ale algoritmului Bayes-Ball

Nodurile neobserve sunt marcate cu alb, iar cele de evidență sunt marcate cu gri. Algoritmul poate determina dacă nodul A este independent sau nu de nodul B date fiind nodurile C_1, C_2 etc. În acest caz, nodurile C_i sunt marcate cu gri. Minge pleacă din nodul A și parcurge rețeaua, ajungând sau nu la B . Dacă nu ajunge, atunci $A \perp\!\!\!\perp B | C_i$. Dacă ajunge, atunci $A \not\perp\!\!\!\perp B | C_i$.

Nodurile pe care le atinge mingea sunt dependente condițional iar nodurile care nu sunt atinse de mingea sunt independente condițional de nodul de start.

Regula 1 specifică relațiile de dependență condițională între „bunici”, părinți și fiu. Dacă părintele este observat (regula 2), „bunicul” devine independent condițional de fiu. Am putea aminti aici presupunea Markov: viitorul și trecutul sunt independente condițional dat fiind prezentul.

Pentru regula 3, un nod părinte pentru doi fiu, dacă nodul este neobservat, fiile sunt dependenți deoarece au o cauză comună ascunsă, aşa că mingea trece. Dacă nodul este observat (regula 4), fiile devin independenți condițional, aşa că mingea va fi blocată.

Pentru regula 5, un nod cu doi părinți, dacă nodul este neobservat, atunci părinții săi sunt independenți iar mingea nu trece. Dacă nodul este observat (regula 6), părinții devin dependenți iar mingea trece datorită revocării prin explicare.

Regulile 7-10 tratează cazurile în care mingea ajunge la un nod terminal, unde este ori blocată (7, 10), ori reflectată înapoi (8, 9).

Pentru a reține mai ușor regulile 3-10, putem face următoarea convenție. Numim noduri „albe” nodurile neobservate și „negre” nodurile de evidență. De asemenea, numim capetele săgeților „albe” dacă niciun arc nu este incident în acel nod (de exemplu arcele care pleacă din nodurile de sus în regulile 3-10) și „negre” dacă arcele sunt incidente în acel nod (arcele care intră în nodurile de jos în regulile 3-10). Regula generală este simplă: dacă nodurile și arcele au aceeași culoare, mingea trece sau este reflectată. Dacă au culori diferite, mingea este blocată.

Pentru a exemplifica, vom considera rețeaua bayesiană din figura 2.9, pe baza căreia vom răspunde la o serie de interogări privind independența și independența condiționată a nodurilor cu ajutorul algoritmului Bayes-Ball.

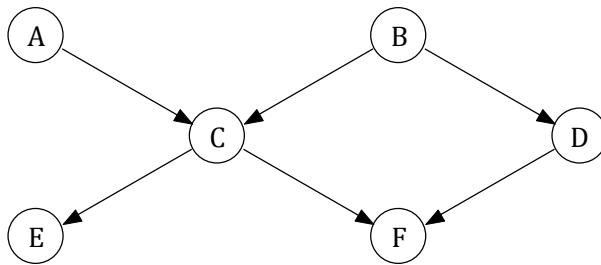
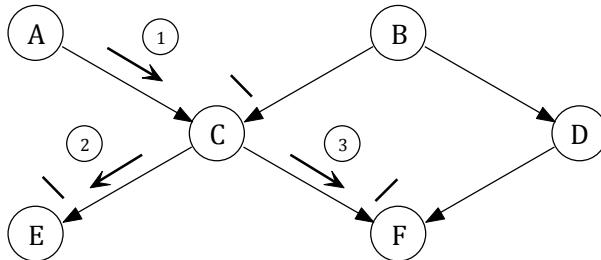
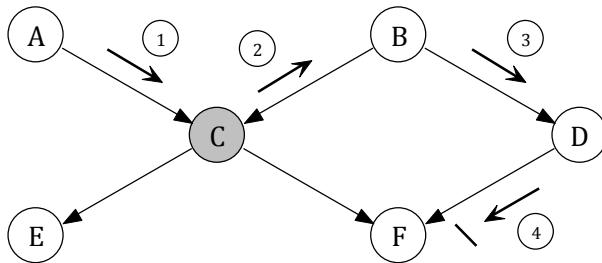


Figura 2.9. Rețea bayesiană pentru exemplificarea algoritmului Bayes-Ball

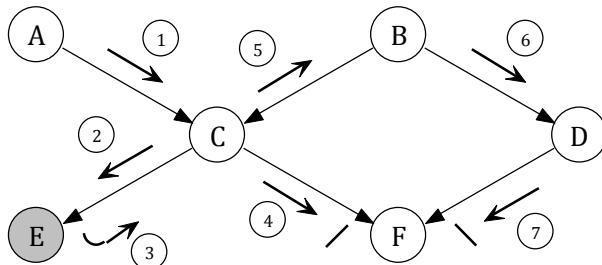
Mai întâi, să determinăm dacă A este independent de D . S-au notat încercuitele numerele etapelor de aplicare a algoritmului și sensul de trimitere a mingii pe arcele rețelei. Se vede în figura de mai jos că mingea nu ajunge la D și prin urmare răspunsul la întrebare este afirmativ.



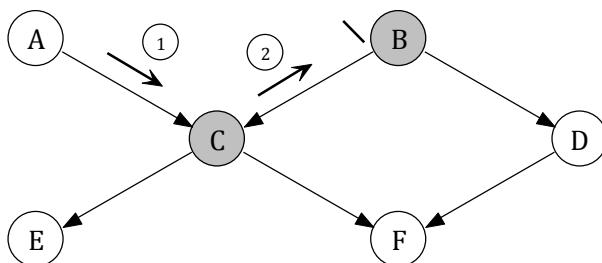
Următoarea interogare este dacă A este independent de D dat fiind C . În figura următoare, se observă nodul evidență C marcat cu gri. Întrucât mingea ajunge de la A la D , răspunsul este în acest caz negativ.



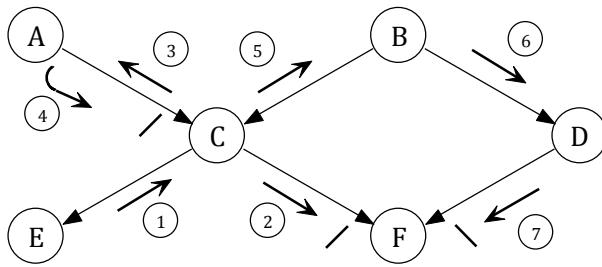
Dorim să aflăm în continuare dacă A este independent de D dat fiind E . Aplicând algoritmul ca în figura următoare, răspunsul este negativ.



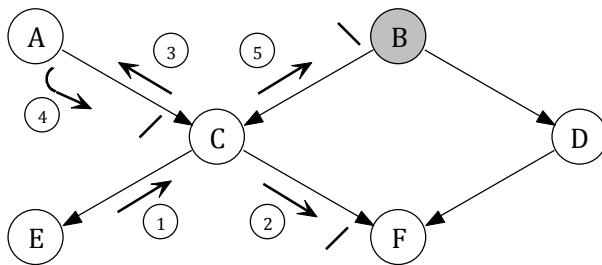
În continuare, vom răspunde dacă A este independent de D dați fiind B și C . Conform rezolvării din figura de mai jos, de data aceasta este rezultatul este afirmativ.



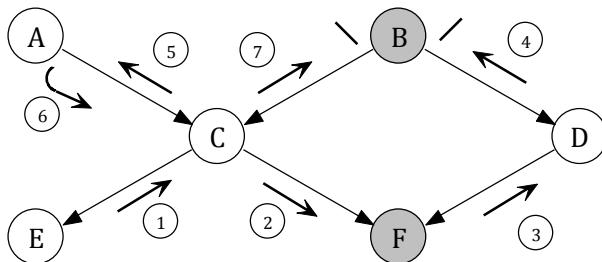
Schimbând nodul de start, dorim să știm dacă E este independent de D . Răspunsul este negativ, după cum se poate observa din figura următoare.



Pentru interogarea privind independența lui E față de D , dat fiind B , răspunsul este afirmativ.



În fine, pentru interogarea privind independența lui E față de D , dat fiind B și F , răspunsul este negativ.



Trebuie menționat faptul că mingea poate merge în direcția opusă arcului din graf. Un arc poate fi parcurs de două ori în direcții opuse.

Însă un arc poate fi parcurs doar o singură dată într-o anumită direcție. De aceea, în scopul implementării, pentru a evita parcurgerea de mai multe ori a arcelor, nodurile evidență sunt marcate ca vizitate când mingea ajunge la ele iar celelalte noduri primesc un marcaj „sus” (engl. “top”) când trimit mingea părinților și un marcaj „jos” (engl “bottom”) când trimit mingea copiilor. De exemplu, un nod marcat „sus” nu mai poate trimite mingea din nou părinților.

2.5. Sortarea topologică

Sortarea sau ordonarea topologică a unui graf este o ordonare liniară a nodurilor sale astfel încât, pentru fiecare arc $A \rightarrow B$, A apare înaintea lui B . Pentru o rețea bayesiană, sortarea topologică asigură faptul că nodurile părinte vor apărea înaintea nodurilor fiu. Orice graf orientat aciclic, cum este o rețea bayesiană, are cel puțin o ordonare topologică.

Algoritmii corespunzători au de obicei o complexitate de timp liniară în numărul de noduri n și de arce a : $O(n + a)$.

Pentru exemplificare, vom considera algoritmul propus de Kahn (1962), care este mai ușor de înțeles, nefiind recursiv. Pseudocodul este următorul:

```
L ← listă inițial vidă care va conține elementele sortate  
S ← mulțimea nodurilor fără părinți
```

```
cât timp S nu este vidă  
    scoate un nod n din S  
    introdu n în L  
    pentru-fiecare nod m cu un arc e de la n la m  
        scoate arcul e din graf
```

dacă m nu are alte arce incidente **atunci**
 introdu m în S
sfârșit-dacă

sfârșit-pentru-fiecare
sfârșit-cât-timp

dacă graful mai are arce **atunci**
întoarce eroare (graful are cel puțin un ciclu)
altfel
întoarce L (elementele sortate topologic)
sfârșit-dacă

Dacă graful este orientat aciclic, soluția se va găsi în lista L . Dacă nu, algoritmul detectează faptul că există cicluri în graf și sortarea topologică este imposibilă. Lista S poate fi implementată ca o coadă sau ca o stivă. În funcție de ordinea nodurilor extrase din S , se pot crea soluții diferite.

Pentru exemplificare, vom considera graful din figura 2.10.

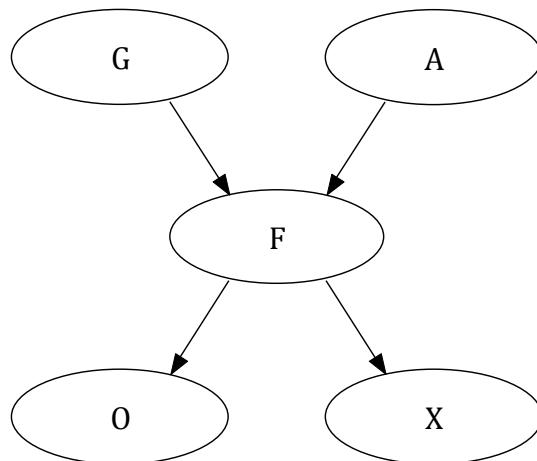


Figura 2.10. Graf simplu pentru sortare topologică

Pașii de execuție ai algoritmului sunt următorii:

1. $L = \emptyset, S = \{G, A\}$
2. $L = \{G\}, S = \{A\}$
3. se elimină arcul GF , F nu poate fi adăugat în S pentru că mai există arcul AF
4. $L = \{G, A\}, S = \emptyset$
5. se elimină arcul AF , $S = \{F\}$
6. $L = \{G, A, F\}, S = \emptyset$
7. se elimină arcul FO , $S = \{O\}$
8. se elimină arcul FX , $S = \{O, X\}$
9. $L = \{G, A, F, O\}, S = \{X\}$
10. $L = \{G, A, F, O, X\}, S = \emptyset$

Soluția este deci: $\{G, A, F, O, X\}$.

Să considerăm și un exemplu puțin mai complex, prezentat în figura 2.11.

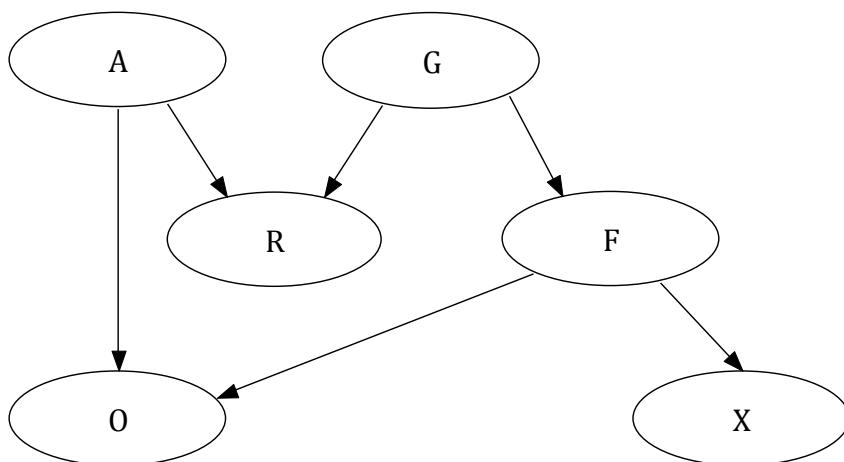


Figura 2.11. Graf mai complex pentru sortare topologică

În acest caz, pașii de execuție ai algoritmului sunt următorii:

1. $L = \emptyset, S = \{A, G\}$
2. $L = \{A\}, S = \{G\}$
3. se elimină arcul AR
3. se elimină arcul AO
4. $L = \{A, G\}, S = \emptyset$
5. se elimină arcul $GR, S = \{R\}$
3. se elimină arcul $GF, S = \{R, F\}$
4. $L = \{A, G, R\}, S = \{F\}$
4. $L = \{A, G, R, F\}, S = \emptyset$
5. se elimină arcul $FO, S = \{O\}$
3. se elimină arcul $FX, S = \{O, X\}$
4. $L = \{A, G, R, F, O\}, S = \{O\}$
4. $L = \{A, G, R, F, O, X\}, S = \emptyset$

Soluția este: $\{A, G, R, F, O, X\}$.

2.6. Construcția automată a rețelelor bayesiene

Pe măsură ce complexitatea modelelor crește, o problemă importantă este construirea rețelelor bayesiene în mod automat, doar pe baza datelor existente. Putem identifica aici două tipuri de probleme:

- determinarea parametrilor direct din date;
- determinarea structurii optime direct din date.

Dacă structura de noduri și arce este cunoscută, determinarea parametrilor direct din date presupune estimarea probabilităților condiționate ale fiecărui nod în funcție de părinți, ca frecvențe relative de apariție a valorilor în cadrul datelor eșantionate disponibile.

Determinarea automată a structurii optime este mult mai dificilă. Dacă nu se cunosc relațiile de cauzalitate sau de corelație între variabile, teoretic, nodurile ar putea fi introduse în rețea în orice ordine. Pentru aceleași evenimente, pot exista mai multe rețele echivalente. În momentul în care este introdus un nod nou, se actualizează relațiile cu nodurile existente pentru crearea arcelor. În cazul cel mai defavorabil, în care nodurile sunt considerate într-o ordine nefericită, o rețea bayesiană devine echivalentă din punct de vedere al numărului de parametri cu distribuția comună de probabilitate.

Un exemplu în acest sens poate fi observat în figura 2.12 (adaptată după Russell & Norvig, 2002), în care ordinea în care se introduc nodurile rețelei din figura 2.3 este de sus în jos: *Oboseală, Anorexie, Abces, Gripă, Febră*, iar arcele sunt create pentru a respecta relațiile de corelație.

Desigur, nu se mai respectă relațiile cauzale logice, dar dacă luăm evenimentele în această ordine, ele nu sunt independente.

Dacă primul nod introdus în rețea este *Oboseala*, și următorul este *Anorexia*, trebuie adăugat un arc între cele două, deoarece informațiile despre primul afectează informațiile despre al doilea. *Anorexia* nu este cauzată de *Oboseală*, dar în lipsa altor informații, valorile lor sunt corelate datorită cauzei comune. Același tip de raționament se aplică pentru celelalte noduri introduse în rețea. În final, aceasta necesită 31 de parametri, la fel ca distribuția comună de probabilitate.

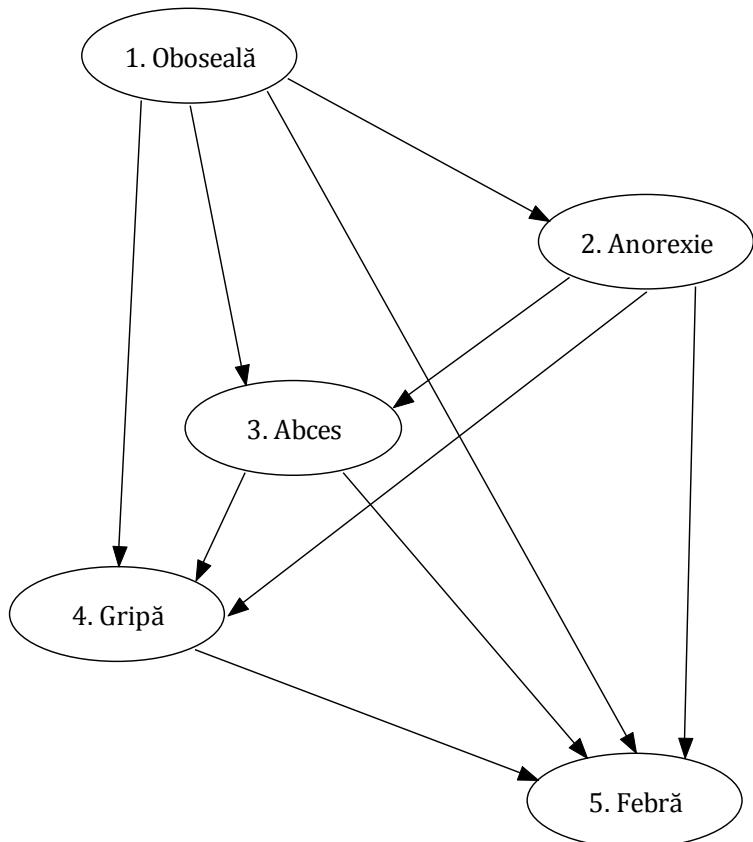


Figura 2.12. Cazul cel mai defavorabil pentru structura unei rețele bayesiane

Din cauza acestor dificultăți, de multe ori se preferă o abordare hibridă, în care structura rețelei este definită de un expert uman, care analizează pe cât posibil relațiile cauzale dintre evenimente, iar parametrii sunt mai apoi estimați în mod automat din date.

Raționamente exacte

3.1. Calculul probabilității unei observații

În acest capitol vom prezenta, pe baza mai multor exemple, o serie de calcule și raționamente care permit prelucrarea informațiile stocate într-o rețea bayesiană.

Vom considera aceeași rețea din capitolul 2. Pentru a fi mai ușor de urmărit, redăm și aici structura (figura 3.1) și tabelele de probabilități (tabelul 3.1).

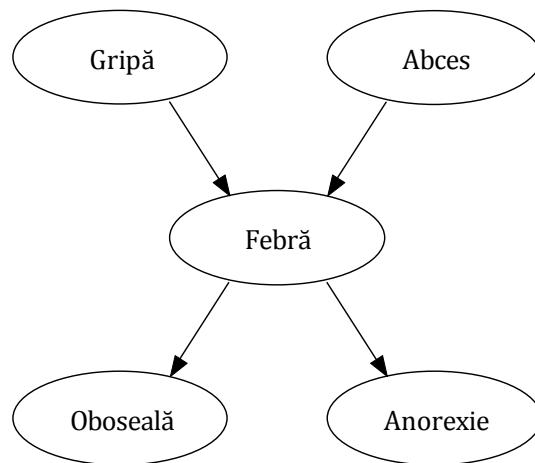


Figura 3.1. Rețea bayesiană cu 5 noduri și 4 arce

Tabelul 3.1. Tabelele de probabilități pentru rețeaua bayesiană

$P(Gripă = Da)$	$P(Gripă = Nu)$
0,1	0,9

$P(Abces = Da)$	$P(Abces = Nu)$
0,05	0,95

$Gripă$	$Abces$	$P(Febră = Da)$	$P(Febră = Nu)$
Da	Da	0,8	0,2
Da	Nu	0,7	0,3
Nu	Da	0,25	0,75
Nu	Nu	0,05	0,95

$Febră$	$P(Oboseală = Da)$	$P(Oboseală = Nu)$
Da	0,6	0,4
Nu	0,2	0,8

$Febră$	$P(Anorexie = Da)$	$P(Anorexie = Nu)$
Da	0,5	0,5
Nu	0,1	0,9

Un prim tip de calcul este determinarea probabilității unei observații, în cazul în care sunt cunoscute valorile tuturor nodurilor din rețea.

De exemplu, să presupunem situația în care o persoană are febră, dar nu are gripă, abces, oboseală și anorexie. Pentru simplitate, vom nota nodurile cu F , G , A , O și X , respectiv,. De asemenea, vom nota cu D și N ca indici valorile Da și Nu . Dacă variabila $Febră$ are valoarea Da , vom nota acest fapt cu F_D . Dacă variabila $Abces$ are valoarea Nu , vom nota acest fapt cu A_N .

Pentru situația de mai sus, trebuie să calculăm probabilitatea $P(F_D, G_N, A_N, O_N, X_N)$. Conform ecuației 2.11, descompunem această probabilitate într-un produs de probabilități condiționate în care factorii sunt probabilitățile tuturor nodurilor, condiționate de părinți:

$$\begin{aligned}
 P(F_D, G_N, A_N, O_N, X_N) = \\
 P(F_D | G_N, A_N) \cdot P(G_N) \cdot P(A_N) \cdot P(O_N | F_D) \cdot P(X_N | F_D) = \\
 0,05 \cdot 0,9 \cdot 0,95 \cdot 0,4 \cdot 0,5 = 0,00855 \approx 1\%.
 \end{aligned}$$

Valorile probabilităților condiționate se caută în tabelele de probabilități ale rețelei. De exemplu, valoarea lui $P(F_D | G_N, A_N)$ corespunde valorii *Da* din ultima linie a tabelei pentru *Febră*.

Rezultatul de aproximativ 1% ne spune că este puțin probabil ca o persoană să aibă febră în lipsa cauzelor și efectelor cunoscute pentru aceasta.

3.2. Calculul probabilităților marginale

Într-o rețea bayesiană putem calcula și probabilitățile marginale ale tuturor nodurilor, adică probabilitățile nodurilor în sine, care nu mai depind de părinți (în sensul probabilităților condiționate). În tabelele date, numai nodurile fără părinți au probabilități marginale, cum ar fi *Gripă* și *Abces*. Dorim să deducem în general care sunt probabilitățile celorlalte noduri. Calculele reprezintă într-un fel o sumă a probabilităților condiționate pentru fiecare valoare, ponderate cu probabilitățile marginale de apariție a valorilor părinților.

Pentru valoarea *Da* a variabilei *Febră* vom avea:

$$\begin{aligned}
 P(F_D) = \\
 P(F_D | G_D, A_D) \cdot P(G_D) \cdot P(A_D) + \\
 P(F_D | G_D, A_N) \cdot P(G_D) \cdot P(A_N) +
 \end{aligned}$$

$$\begin{aligned}
 & P(F_D|G_N, A_D) \cdot P(G_N) \cdot P(A_D) + \\
 & P(F_D|G_N, A_N) \cdot P(G_N) \cdot P(A_N) = \\
 & 0,8 \cdot 0,1 \cdot 0,05 + 0,7 \cdot 0,1 \cdot 0,95 + 0,25 \cdot 0,9 \cdot 0,05 + 0,05 \\
 & \quad \cdot 0,9 \cdot 0,95 = 0,1245 \approx 12\%.
 \end{aligned}$$

Având în vedere faptul că singurele valori ale variabilei *Febră* sunt *Da* și *Nu*, probabilitatea valorii *Nu* va reprezenta restul până la 1:

$$P(F_N) = 1 - P(F_D) = 0,8755 \approx 88\%.$$

Același tip de calcule se realizează pentru variabila *Oboseală*:

$$\begin{aligned}
 P(O_D) &= \\
 P(O_D|F_D) \cdot P(F_D) + P(O_D|F_N) \cdot P(F_N) &= \\
 0,6 \cdot 0,1245 + 0,2 \cdot 0,8755 &= 0,2498 \approx 25\%, \\
 P(O_N) &= 1 - P(O_D) = 0,7502 \approx 75\%.
 \end{aligned}$$

și de asemenea, pentru *Anorexie*:

$$\begin{aligned}
 P(X_D) &= \\
 P(X_D|F_D) \cdot P(F_D) + P(X_D|F_N) \cdot P(F_N) &= \\
 0,5 \cdot 0,1245 + 0,1 \cdot 0,8755 &= 0,1498 \approx 15\%, \\
 P(X_N) &= 1 - P(X_D) = 0,8502 \approx 85\%.
 \end{aligned}$$

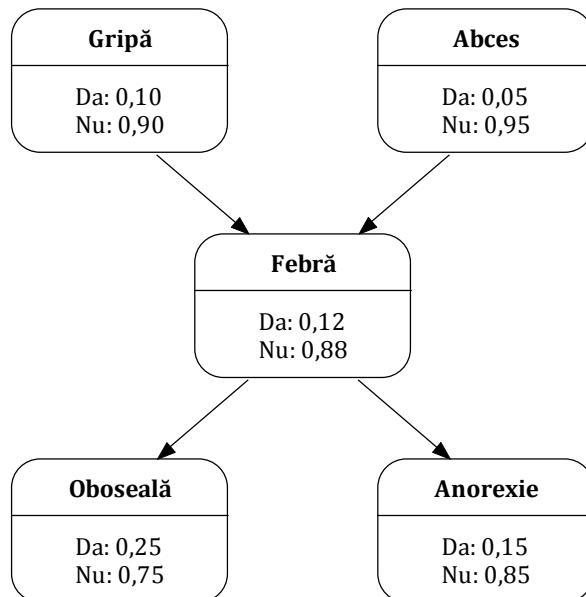


Figura 3.2. Probabilitățile marginale ale nodurilor rețelei

Figura 3.2 prezintă succint probabilitățile marginale pentru toate nodurile din rețea.

3.3. Inferență prin enumerare

Spre deosebire de calculul probabilității atunci când cunoaștem valorile tuturor variabilelor, în cazul inferenței prin enumerare dorim să răspundem la întrebări generale privind probabilitățile unor noduri, cunoscând doar valorile unora dintre noduri.

Folosind acest procedeu, putem răspunde practic la orice întrebare privind evenimentele codate în rețea. Având în vedere niște evidențe, adică observații sau evenimente despre care stim că s-au întâmplat, putem calcula probabilitățile tuturor celorlalte noduri din rețea.

Mai exact, scopul inferenței prin enumerare este de a calcula probabilitatea unei variabile interogate (engl. “query”), date fiind variabilele observate (evidență).

Idea de bază este tot calcularea unui produs de probabilități condiționate, însă în cazul variabilelor despre care nu se cunoaște nimic (nu sunt nici observate și nici interogate), se sumează variantele corespunzătoare *tuturor* valorilor acestora.

Să considerăm următoarea întrebare: „Care este probabilitatea ca o persoană să aibă gripă, dacă prezintă simptome de oboseală și anorexie?”

Vom calcula *independent* $P(G_D|O_D, X_D)$ și $P(G_N|O_D, X_D)$.

Pentru $P(G_D|O_D, X_D)$, variabilele rămase sunt *Abcesul* și *Febra*. În consecință, vom suma probabilitățile corespunzătoare tuturor valorilor acestor variabile: $a \in \{A_D, A_N\}$ și $f \in \{F_D, F_N\}$. De asemenea, pentru a crește eficiența calculelor, se recomandă ca variabilele rămase să fie mai întâi sortate topologic, astfel încât părinții să apară înaintea copiilor. În acest caz, se vor putea descompune mai ușor sumele, scoțând în față factorii care nu depind de o anumită variabilă.

$$P(G_D|O_D, X_D) =$$

$$\alpha \cdot \sum_{a \in \{A_D, A_N\}} \sum_{f \in \{F_D, F_N\}} P(G_D, a, f, O_D, X_D) =$$

$$\alpha \cdot \sum_a \sum_f P(G_D) \cdot P(a) \cdot P(f|G_D, a) \cdot P(O_D|f) \cdot P(X_D|f) =$$

$$\alpha \cdot P(G_D) \cdot \sum_a P(a) \cdot \sum_f P(f|G_D, a) \cdot P(O_D|f) \cdot P(X_D|f) =$$

$$\alpha \cdot P(G_D) \cdot \sum_a P(a) \cdot [P(F_D|G_D, a) \cdot P(O_D|F_D) \cdot P(X_D|F_D) +$$

$$\begin{aligned}
 & P(F_N|G_D, a) \cdot P(O_D|F_N) \cdot P(X_D|F_N)] = \\
 & \alpha \cdot P(G_D) \cdot \{P(A_D) \cdot [P(F_D|G_D, A_D) \cdot P(O_D|F_D) \cdot P(X_D|F_D) + \\
 & P(F_N|G_D, A_D) \cdot P(O_D|F_N) \cdot P(X_D|F_N)] + \\
 & P(A_N) \cdot [P(F_D|G_D, A_N) \cdot P(O_D|F_D) \cdot P(X_D|F_D) + \\
 & P(F_N|G_D, A_N) \cdot P(O_D|F_N) \cdot P(X_D|F_N)]\} = \\
 & \alpha \cdot 0,1 \cdot \{0,05 \cdot [0,8 \cdot 0,6 \cdot 0,5 + 0,2 \cdot 0,2 \cdot 0,1] + \\
 & 0,95 \cdot [0,7 \cdot 0,6 \cdot 0,5 + 0,3 \cdot 0,2 \cdot 0,1]\} = \\
 & \alpha \cdot 0,02174.
 \end{aligned}$$

În exemplul de mai sus, se observă că $P(a)$ nu depinde de f și prin urmare, suma corespunzătoare variabilei *Abces* a fost scoasă în fața sumei corespunzătoare variabilei *Febră*, evitându-se duplicarea unor calcule. Nodul *Abces*, neavând părinți, este în fața *Febrei* în sortarea topologică.

Se remarcă variabila α care intervene în expresia probabilității. Vom explica sensul acesteia imediat, după ce vom considera și calculele pentru $P(G_N|O_D, X_D)$, în mod analog:

$$\begin{aligned}
 & P(G_N|O_D, X_D) = \\
 & \alpha \cdot \sum_{a \in \{A_D, A_N\}} \sum_{f \in \{F_D, F_N\}} P(G_N, a, f, O_D, X_D) = \\
 & \alpha \cdot \sum_a \sum_f P(G_N) \cdot P(a) \cdot P(f|G_N, a) \cdot P(O_D|f) \cdot P(X_D|f) = \\
 & \alpha \cdot P(G_N) \cdot \sum_a P(a) \cdot \sum_f P(f|G_N, a) \cdot P(O_D|f) \cdot P(X_D|f) = \\
 & \alpha \cdot P(G_N) \cdot \{P(A_D) \cdot [P(F_D|G_N, A_D) \cdot P(O_D|F_D) \cdot P(X_D|F_D) +
 \end{aligned}$$

$$\begin{aligned}
 & P(F_N|G_N, A_D) \cdot P(O_D|F_N) \cdot P(X_D|F_N)] + \\
 & P(A_N) \cdot [P(F_D|G_N, A_N) \cdot P(O_D|F_D) \cdot P(X_D|F_D)] + \\
 & P(F_N|G_N, A_N) \cdot P(O_D|F_N) \cdot P(X_D|F_N)]\} = \\
 & \alpha \cdot 0,9 \cdot \{0,05 \cdot [0,25 \cdot 0,6 \cdot 0,5 + 0,75 \cdot 0,2 \cdot 0,1] + \\
 & 0,95 \cdot [0,05 \cdot 0,6 \cdot 0,5 + 0,95 \cdot 0,2 \cdot 0,1]\} = \\
 & \alpha \cdot 0,03312.
 \end{aligned}$$

Se știe că $P(G_D|O_D, X_D) + P(G_N|O_D, X_D) = 1$, deoarece *Da* și *Nu* sunt singurele valori posibile pentru *Gripă*. Având în vedere că $P(G_D|O_D, X_D) = \alpha \cdot 0,02174$ și $P(G_N|O_D, X_D) = \alpha \cdot 0,03312$, există $\alpha = 18,23$, astfel încât suma celor două probabilități să fie 1. În consecință, rezultatul interogării este:

$$P(G_D|O_D, X_D) = 0,39628 \approx 40\%,$$

$$P(G_N|O_D, X_D) = 0,60372 \approx 60\%.$$

3.4. Inferență prin eliminarea variabilelor

Dacă urmărim cu atenție pașii efectuați pentru a determina probabilitățile condiționate prin metoda inferenței prin enumerare, constatăm o repetare a unor calcule, evidențiată mai jos:

$$\begin{aligned}
 & P(G_D|O_D, X_D) = \\
 & \alpha \cdot P(G_D) \cdot \{P(A_D) \cdot [P(F_D|G_D, A_D) \cdot P(O_D|F_D) \cdot P(X_D|F_D)] +
 \end{aligned}$$

$$\begin{aligned}
 & P(F_N|G_D, A_D) \cdot P(O_D|F_N) \cdot P(X_D|F_N)] + \\
 & P(A_N) \cdot [P(F_D|G_D, A_N) \cdot P(O_D|F_D) \cdot P(X_D|F_D)] + \\
 & P(F_N|G_D, A_N) \cdot P(O_D|F_N) \cdot P(X_D|F_N)] = \\
 & \alpha \cdot 0,1 \cdot \{0,05 \cdot [0,8 \cdot \mathbf{0,6} \cdot \mathbf{0,5} + 0,2 \cdot \mathbf{0,2} \cdot \mathbf{0,1}] + \\
 & 0,95 \cdot [0,7 \cdot \mathbf{0,6} \cdot \mathbf{0,5} + 0,3 \cdot \mathbf{0,2} \cdot \mathbf{0,1}]\}.
 \end{aligned}$$

Calcularea unor sume parțiale și folosirea lor directă atunci când este nevoie poate aduce o îmbunătățire semnificativă a vitezei de execuție în cazul unor rețele bayesiene complexe și a unui număr mic de variabile observate, echivalent cu un număr mare de sume.

Metoda eliminării variabilelor se bazează pe *factorizare*. Probabilitatea fiecărei variabile reprezintă un factor:

$$\begin{aligned}
 P(G_D|O_D, X_D) = \\
 \alpha \cdot \sum_{a \in \{A_D, A_N\}} \sum_{f \in \{F_D, F_N\}} P(G_D, a, f, O_D, X_D) = \\
 \alpha \cdot \sum_a \sum_f P(G_D) \cdot P(a) \cdot P(f|G_D, a) \cdot P(O_D|f) \cdot P(X_D|f) = \\
 \alpha \cdot \underbrace{P(G_D)}_{\mathbf{G}} \cdot \underbrace{\sum_a P(a)}_{\mathbf{A}} \cdot \underbrace{\sum_f P(f|G_D, a)}_{\mathbf{F}} \cdot \underbrace{P(O_D|f)}_{\mathbf{O}} \cdot \underbrace{P(X_D|f)}_{\mathbf{X}}
 \end{aligned}$$

Compunerea acestor factori se realizează cu o operație numită *produs punct cu punct* (engl. “pointwise product”), în care variabilele rezultatului reprezintă reuniunea variabilelor operanzilor.

De exemplu, pentru 3 variabile X , Y și Z , produsul punct cu punct al factorilor $f_1(X, Y)$ și $f_2(Y, Z)$ este:

$$f_1(X, Y) \times f_2(Y, Z) = f_3(X, Y, Z). \quad (3.1)$$

Modul în care se calculează valorile factorului rezultat este mai ușor de înțeles pe baza exemplului din tabelul 3.2.

Tabelul 3.2. Exemplu de factorizare

X	Y	$f_1(X, Y)$	Y	Z	$f_2(Y, Z)$	X	Y	Z	$f_3(X, Y, Z)$
A	A	0,1	A	A	0,5	A	A	A	0,1 · 0,5
A	F	0,2	A	F	0,6	A	A	F	0,1 · 0,6
F	A	0,3	F	A	0,7	A	F	A	0,2 · 0,7
F	F	0,4	F	F	0,8	A	F	F	0,2 · 0,8
						F	A	A	0,3 · 0,5
						F	A	F	0,3 · 0,6
						F	F	A	0,4 · 0,7
						F	F	F	0,4 · 0,8

Factorul f_1 depinde de 2 variabile și să presupunem pentru simplitate că sunt binare, cu valorile *Adevărat* (A) și *Fals* (F). Prin urmare, factorul va avea $2^2 = 4$ valori, corespunzătoare tuturor combinațiilor de valori pentru variabile. Analog pentru factorul f_2 . Valoarea factorului f_3 pentru o anumită combinație de valori ale variabilelor sale este egală cu produsul valorilor factorilor-operanți atunci când variabilele acestora iau aceleași valori ca acelele din factorul-rezultat.

De exemplu:

$$f_3(X_A, Y_F, Z_A) = f_1(X_A, Y_F) \cdot f_2(Y_F, Z_A) = 0,2 \cdot 0,7 = 0,14,$$

$$f_3(X_F, Y_A, Z_A) = f_1(X_F, Y_A) \cdot f_2(Y_A, Z_A) = 0,3 \cdot 0,5 = 0,15.$$

În cele ce urmează, vom aplica metoda eliminării variabilelor pentru a răspunde la aceeași întrebare ca în secțiunea 3.3: „Care este probabilitatea ca o persoană să aibă gripă, dacă prezintă simptome de oboseală și anorexie?”

Am văzut că:

$$P(G_D | O_D, X_D) = \alpha \cdot \underbrace{P(G_D)}_{\mathbf{G}} \cdot \sum_a \underbrace{P(a)}_{\mathbf{A}} \cdot \sum_f \underbrace{P(f|G_D, a)}_{\mathbf{F}} \cdot \underbrace{P(O_D|f)}_{\mathbf{O}} \cdot \underbrace{P(X_D|f)}_{\mathbf{X}}$$

Mai întâi, vom calcula factorii corespunzători variabilelor, utilizând probabilitățile condiționate din tabelele de probabilități.

Calculele se fac de la dreapta la stânga, variabilele fiind sortate topologic, de la stânga la dreapta. Aici, sortarea topologică a variabilelor este: { *Gripă, Abces, Febră, Obboseală, Anorexie* }.

Pentru variabilele *Anorexie*, respectiv *Obboseală*, factorii au valorile date de probabilitățile condiționate, depinzând de părintele lor, *Febra*:

F	$f_X(F)$
D	0,5
N	0,1

F	$P(X F)$
D	0,5
N	0,1

F	$f_O(F)$
D	0,6
N	0,2

F	$P(O F)$
D	0,6
N	0,2

Compunem acum aceste două variabile, rezultând factorul f_{OX} , care depinde și el (numai) de *Febră*:

$$f_{OX}(F) = f_O(F) \times f_X(F).$$

F	$f_{ox}(F)$
D	$0,6 \cdot 0,5 = 0,3$
N	$0,2 \cdot 0,1 = 0,02$

=

F	$f_o(F)$
D	0,6
N	0,2

x

F	$f_x(F)$
D	0,5
N	0,1

În continuare, calculăm factorul f_{FOX} , prin compunerea factorului f_{ox} cu factorul corespunzător *Febrei*, următoarea variabilă de la dreapta spre stânga. *Febra* depinde de *Gripă* și *Abces*, prin urmare:

$$f_{FOX}(F, G, A) = f_F(F, G, A) \times f_{ox}(F).$$

F	G	A	$f_{fox}(F,G,A)$
D	D	D	$0,8 \cdot 0,3 = 0,24$
D	D	N	$0,7 \cdot 0,3 = 0,21$
D	N	D	$0,25 \cdot 0,3 = 0,075$
D	N	N	$0,05 \cdot 0,3 = 0,015$
N	D	D	$0,2 \cdot 0,02 = 0,004$
N	D	N	$0,3 \cdot 0,02 = 0,006$
N	N	D	$0,75 \cdot 0,02 = 0,015$
N	N	N	$0,95 \cdot 0,02 = 0,019$

=

F	G	A	$P(F G,A)$
D	D	D	0,8
D	D	N	0,7
D	N	D	0,25
D	N	N	0,05
N	D	D	0,2
N	D	N	0,3
N	N	D	0,75
N	N	N	0,95

x

F	$f_{ox}(F)$
D	0,3
N	0,02

În acest moment, am ajuns la suma după valorile variabilei *Febră*. Pentru a continua calculele, se procedează la *eliminarea prin sumare* (engl. “sum out”) a variabilei, de unde vine și numele metodei.

F este eliminată iar factorul său va depinde numai de *G* și *A*. Valorile factorului rezultat, notat $f_{\bar{F}OX}$, se calculează, pentru fiecare combinație a valorilor variabilelor rămase, ca sumă a valorilor factorului inițial, f_{FOX} pentru toate valorile variabilei eliminate.

De exemplu:

$$f_{\bar{F}OX}(G_D, A_D) = f_{FOX}(F_D, G_D, A_D) + f_{FOX}(F_N, G_D, A_D).$$

G	A	$f_{\bar{F}OX}(G, A)$
D	D	$0,24 + 0,004 = 0,244$
D	N	$0,21 + 0,006 = 0,216$
N	D	$0,075 + 0,015 = 0,09$
N	N	$0,015 + 0,019 = 0,034$

Se poate vedea de exemplu că valoarea 0,244 apare și în calculele metodei de inferență prin enumerare ($0,8 \cdot 0,6 \cdot 0,5 + 0,2 \cdot 0,2 \cdot 0,1$). Aceste 4 valori din tabelul de mai sus sunt exact valorile parantezelor interioare din exemplul din secțiunea 3.3. De această dată însă, calculele se fac o singură dată, la determinarea factorului.

Urmează apoi tratarea variabilei *Abces*:

$$f_{A\bar{F}OX}(G, A) = f_A(A) \times f_{\bar{F}OX}(G, A).$$

Valorile factorului sunt cele din tabelul de mai jos:

G	A	$f_{A\bar{F}OX}(G, A)$
D	D	$0,05 \cdot 0,244 = 0,0122$
D	N	$0,95 \cdot 0,216 = 0,2052$
N	D	$0,05 \cdot 0,09 = 0,0045$
N	N	$0,95 \cdot 0,034 = 0,0323$

=

A	$f_A(A)$
D	0,05
N	0,95

x

G	A	$f_{\bar{F}OX}(G, A)$
D	D	0,244
D	N	0,216
N	D	0,09
N	N	0,034

La fel, având acum o sumă după *A*, această variabilă este eliminată prin sumare, rezultând factorul $f_{\bar{A}\bar{F}OX}(G)$, cu valorile din tabelul următor:

G	$f_{\bar{A}\bar{F}OX}(G)$
D	$0,0122 + 0,2052 = 0,2174$
N	$0,0045 + 0,0323 = 0,0368$

Se poate observa că aceste valori sunt la fel cu acele din parantezele exterioare din calculele din secțiunea 3.3, înainte de înmulțirile cu $P(G_D)$, respectiv cu $P(G_N)$.

Ultimul factor este:

$$f_{G\bar{A}\bar{F}OX}(G) = f_G(G) \times f_{\bar{A}\bar{F}OX}(G, A).$$

cu valorile de mai jos:

G	$f_{G\bar{A}\bar{F}OX}(G)$
D	$0,1 \cdot 0,0122 = 0,02174$
N	$0,9 \cdot 0,0323 = 0,03312$

=

G	$f_G(G)$
D	0,1
N	0,9

 \times

G	$f_{\bar{A}\bar{F}OX}(G)$
D	0,2174
N	0,0368

Se vede că valorile sunt aceleasi cu rezultatele finale din exemplul secțiunii 3.3. și în cazul metodei de eliminare a variabilelor, trebuie normalize probabilitățile determinate, astfel încât suma lor să fie 1.

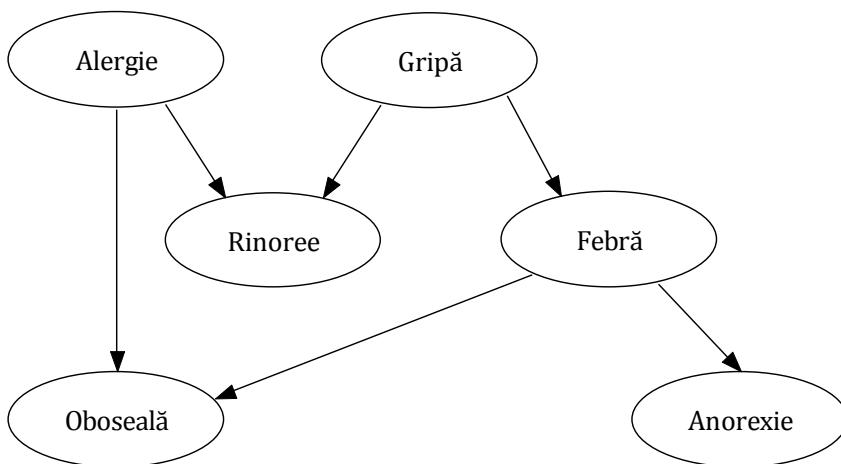


Figura 3.3. Rețea bayesiană cu 6 noduri și 6 arce

3.5. Variabile cu valori multiple. Ignorarea variabilelor irelevante

În această secțiune, vom considera un caz mai general de rețele bayesiene, cu mai multe variabile și eliminând restricția ca acestea să aibă valori binare. Pentru analiză, vom utiliza rețea din figura 3.3.

Tabelele de probabilități sunt date în continuare. Variabila *Febră* are 3 valori: *Absentă* (*A*), *Mică* (*M*) și *Ridicată* (*R*).

Tabelul 3.3. Tabelele de probabilități pentru rețea bayesiană

$P(Alergie = Da)$	$P(Alergie = Nu)$
0,05	0,95

$P(Gripă = Da)$	$P(Gripă = Nu)$
0,1	0,9

<i>Gripă</i>	<i>Alergie</i>	$P(Rinoree = Da)$	$P(Rinoree = Nu)$
Da	Da	0,95	0,05
Da	Nu	0,8	0,2
Nu	Da	0,9	0,1
Nu	Nu	0,1	0,9

<i>Gripă</i>	$P(Febră = Absentă)$	$P(Febră = Mică)$	$P(Febră = Ridicată)$
Da	0,1	0,25	0,65
Nu	0,9	0,05	0,05

<i>Febră</i>	<i>Alergie</i>	$P(Oboseală = Da)$	$P(Oboseală = Nu)$
Absentă	Da	0,3	0,7
Absentă	Nu	0,1	0,9
Mică	Da	0,5	0,5
Mică	Nu	0,4	0,6
Ridicată	Da	0,7	0,3
Ridicată	Nu	0,6	0,4

<i>Febră</i>	$P(Anorexie = Da)$	$P(Anorexie = Nu)$
Absentă	0,1	0,9
Mică	0,2	0,8
Ridicată	0,5	0,5

Calculele de probabilități și metodele de inferență sunt la fel.

Să considerăm următoarea interogare: „Care este probabilitatea ca o persoană să aibă alergie, dacă manifestă oboseală și rinoree?”

$$P(A_D | O_D, R_D) = \alpha \cdot \sum_{g \in \{G_D, G_N\}} \sum_{f \in \{F_A, F_M, F_R\}} \sum_{x \in \{X_D, X_N\}} P(g, f, O_D, R_D, x).$$

Pentru a optimiza rezolvarea, utilizăm următorul rezultat.

O variabilă Y este *irrelevantă* pentru o interogare dacă $Y \notin \Pi(\{Q\} \cup E)$, unde Q este variabila interogată, E este mulțimea variabilelor observate (evidență) iar $\Pi(M)$ este mulțimea predecesorilor tuturor nodurilor din mulțimea M .

În exemplul nostru, $Q = \{A\}$, $E = \{O, R\}$ și $\Pi(\{A, O, R\}) = \{A, G, F\}$. Variabila *Anorexie* nu aparține acestei mulțimi. Prin urmare, această variabilă este irelevantă pentru interogarea curentă și poate fi ignorată:

$$P(A_D | O_D, R_D) = \alpha \cdot \sum_{g \in \{G_D, G_N\}} \sum_{f \in \{F_A, F_M, F_R\}} \sum_{x \in \{X_D, X_N\}} P(g, f, O_D, R_D, x) = \\ \alpha \cdot \sum_g \sum_f P(G_D) \cdot P(a) \cdot P(f|G_D) \cdot P(O_D|a, f) \cdot P(R_D|a, G_D)$$

Calculele de probabilități se realizează ca în secțiunile anterioare, obținând în final:

$$P(A_D|O_D, R_D) = 0,24554 \approx 25\%,$$

$$P(A_N|O_D, R_D) = 0,75446 \approx 75\%.$$

3.6. Cea mai probabilă explicație

Un alt tip de rezultat pe care îl putem obține pe baza unei rețele bayesiene este *cea mai probabilă explicație* (engl. “most probable explanation”) pentru o evidență.

Fie E mulțimea variabilelor observate (evidență) și M mulțimea celorlalte variabile, numită și mulțimea de explicare. Se dorește găsirea combinației de valori pentru variabilele din mulțimea de explicare, pentru care probabilitatea tuturor valorilor astfel obținute ale nodurilor rețelei să fie maximă: $\text{argmax}_m P(m, e)$, unde m este combinația de valori ale variabilelor din M iar e este combinația (dată) de valori ale variabilelor din E .

Ca exemplu, pentru rețeaua din figura 3.3, dorim să calculăm cea mai probabilă explicație pentru trei situații: $E_1 = \{R_D, O_D, X_N, F_A\}$, $E_2 = \{R_D, O_D, X_N, F_M\}$ și $E_3 = \{R_D, O_D, X_N, F_R\}$. Mai exact, dorim să găsim cauza (alergie sau gripă) atunci când o persoană are rinoree (îi curge nasul), oboseală dar nu are anorexie (lipsa poftei de mâncare). Cele trei situații sunt diferențiate de nivelul febrei: în primul caz este absentă, în al doilea caz este mică iar în al treilea caz este ridicată.

În tabelul 3.4, primele patru coloane indică valorile celor patru variabile de evidență. Coloana 5 prezintă probabilitatea valorii A_D , în condițiile evidențelor date. Coloana 6 prezintă probabilitatea valorii G_D , în condițiile evidențelor date. Variabilele A și G sunt binare, astfel încât este suficientă menționarea probabilității unei singure valori. Coloana 7 indică probabilitatea $P(m, e)$ pentru toate combinațiile de valori ale variabilelor de

explicare. Combinăția pentru care această probabilitate este maximă este marcată cu litere aldine. Ultima coloană indică rezultatele: valorile variabilelor de explicare care determină cea mai probabilă explicație.

Tabelul 3.4. Căutarea exhaustivă a celei mai probabile explicații

R	O	X	F	A_D	G_D	$P(m, e)$	A, G – CMPE
D	D	N	A	57%	5%	A_N, G_N 0,00693 A_N, G_D 0,00068 A_D, G_N 0,00984 A_D, G_D 0,00013	A_D, G_N
D	D	N	M	15%	75%	A_N, G_N 0,00137 A_N, G_D 0,00608 A_D, G_N 0,00081 A_D, G_D 0,00048	A_N, G_D
D	D	N	R	10%	89%	A_N, G_N 0,00128 A_N, G_D 0,01482 A_D, G_N 0,00071 A_D, G_D 0,00108	A_N, G_D

În primul caz, în absența febrei, cel mai probabil este că persoana are alergie și nu gripă. Dacă are febră, mică sau ridicată, este mai probabil ca persoana să aibă gripă și nu alergie.

Valorile din coloanele 5 și 6 sunt în concordanță cu cea mai probabilă explicație, totuși trebuie precizat că $P(A_D)$ și $P(G_D)$ sunt calculate considerând izolat variabilele A și G . Cea mai probabilă explicație le consideră împreună. În cazul în care rețeaua conține un număr mare de noduri, pot exista diferențe între cele două tipuri de rezultate.

Raționamente aproximative

4.1. Introducere

Pentru probleme din „lumea reală” au fost construite rețele bayesiene cu sute de noduri, pentru care algoritmii exacti își ating limitele, întrucât inferența este o problemă NP-dificilă (engl. “NP-hard”). Pentru rețele foarte complexe, inferența aproximativă este singura posibilitate de a obține un rezultat. De asemenea, pentru alte probleme în care precizia nu este un factor critic, inferența aproximativă poate aduce un câștig important din punct de vedere al vitezei de calcul.

Algoritmii de eșantionare stohastică sunt cele mai des întâlnite metode aproximative de inferență. Ideea de bază este generarea aleatorie a unor instanțieri ale rețelei, adică determinarea de valori coerente pentru variabilele sale, și apoi calcularea frecvențelor instanțierilor în care apar valorile dorite pentru anumite variabile. Avantajul acestei clase de algoritmi este faptul că precizia calculelor crește cu numărul de eșantioane generate și este relativ independentă de dimensiunea rețelei. De asemenea, timpul de execuție este relativ independent de topologia rețelei și variază liniar cu numărul de eșantioane. Pentru aplicațiile în care timpul este un factor critic, algoritmul poate fi oprit oricând, producând un rezultat cu o precizie în general dependentă de numărul de eșantioane generate până în acel moment (Cheng, 2001).

Metodele care vor fi descrise în continuare sunt de tipul *eșantionării progresive* (engl. “forward sampling”), în care eșantioanele sunt generate în ordinea topologică a nodurilor din rețea.

4.2. Inferență stochastică prin ponderarea verosimilității

Metoda de inferență prin *ponderarea verosimilității* (engl. “likelihood weighting”) este o metodă simplă și eficientă de aproximare stochastică (Fung & Chang, 1990; Shachter & Peot, 1990). Ideea de bază este generarea aleatorie a unor instanțieri ale rețelei și calculul probabilităților dorite ca frecvențe relative de apariție. Valorile variabilelor neobservate au probabilități de apariție în conformitate cu probabilitățile nodurilor, iar nodurile de evidență iau mereu valorile observate.

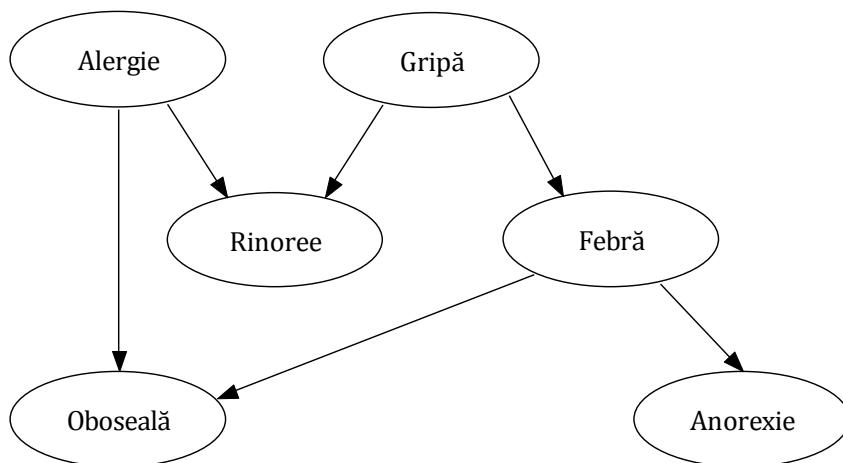


Figura 4.1. Rețea bayesiană cu 6 noduri și 6 arce

Să considerăm aceeași rețea din figura 4.1 cu tabelele de probabilități din tabelul 4.1. Vom determina probabilitatea nodurilor neobservate din rețea, considerând următoarea evidență: *Rinoree = Da* și *Oboseală = Da*.

Tabelul 4.1. Tabelele de probabilități pentru rețeaua bayesiană

$P(Alergie = Da)$	$P(Alergie = Nu)$
0,05	0,95

$P(Gripă = Da)$	$P(Gripă = Nu)$
0,1	0,9

<i>Gripă</i>	<i>Alergie</i>	$P(Rinoree = Da)$	$P(Rinoree = Nu)$
Da	Da	0,95	0,05
Da	Nu	0,8	0,2
Nu	Da	0,9	0,1
Nu	Nu	0,1	0,9

<i>Gripă</i>	$P(Febră = Absentă)$	$P(Febră = Mică)$	$P(Febră = Ridicată)$
Da	0,1	0,25	0,65
Nu	0,9	0,05	0,05

<i>Febră</i>	<i>Alergie</i>	$P(Oboseală = Da)$	$P(Oboseală = Nu)$
Absentă	Da	0,3	0,7
Absentă	Nu	0,1	0,9
Mică	Da	0,5	0,5
Mică	Nu	0,4	0,6
Ridicată	Da	0,7	0,3
Ridicată	Nu	0,6	0,4

<i>Febră</i>	$P(Anorexie = Da)$	$P(Anorexie = Nu)$
Absentă	0,1	0,9
Mică	0,2	0,8
Ridicată	0,5	0,5

Nodurile fără părinți vor fi instanțiate potrivit probabilităților lor marginale.

Nodul *Alergie* va lua valoarea *Da* cu probabilitatea de 5% și *Nu* cu probabilitatea de 95%. Din punct de vedere practic, acest lucru corespunde

unui prag cu valoare 0,05. Se generează după distribuția uniformă un număr aleatoriu r între 0 și 1. Dacă $r < 0,05$, nodul va lua valoarea *Da*, altfel va lua valoarea *Nu*.

Analog, nodul *Gripă* va lua valoarea *Da* cu probabilitatea de 10% și *Nu* cu probabilitatea de 90%.

Să presupunem că nodurile *Alergie* și *Gripă* au fost instanțiate ambele cu valoarea *Nu*.

Următorul nod, în ordinea dată de sortarea topologică, este *Rinoree*. Acesta este un nod evidență și va fi instanțiat întotdeauna cu valoarea observată, *Da*.

Urmează nodul *Febră* iar probabilitățile valorilor acestuia sunt cele condiționate de valorile deja cunoscute:

$$P(F_A|A_N, G_N, R_D, O_D) = P(F_A|G_N) = 0,9,$$

$$P(F_M|A_N, G_N, R_D, O_D) = P(F_M|G_N) = 0,05,$$

$$P(F_R|A_N, G_N, R_D, O_D) = P(F_R|G_N) = 0,05.$$

Din punct de vedere al implementării, vor exista în acest caz două praguri corespunzătoare probabilităților cumulate: 0,9 și $0,9 + 0,05 = 0,95$. Se generează un număr aleatoriu r între 0 și 1. Dacă $r < 0,9$, nodul va lua valoarea *Absentă*, dacă $r \in [0,9; 0,95)$, nodul va lua valoarea *Mică*, iar dacă $r \geq 0,95$, nodul va lua valoarea *Mare*.

Să presupunem că nodul *Febră* este instanțiat cu valoarea *Absentă*. Mai rămâne variabila evidență *Oboseală*, care va lua automat valoarea *Da*, și variabila *Anorexie*, cu următoarele probabilități:

$$P(X_D|A_N, G_N, R_D, F_A, O_D) = P(X_D|F_A) = 0,1,$$

$$P(X_N|A_N, G_N, R_D, F_A, O_D) = P(X_N|F_A) = 0,9.$$

Să presupunem că valoarea va fi Nu .

Astfel, am generat o instanțiere a rețelei (un caz) cu următoarele valori: $s = (A_N, G_N, R_D, F_A, O_D, X_N)$.

Ponderea unui caz este:

$$w(s) = \frac{\prod_{x \in U} P(x|\pi(x))}{\prod_{x \in U \setminus E} P(x|\pi(x))}, \quad (4.1)$$

unde U este mulțimea tuturor nodurilor iar E este mulțimea nodurilor evidență.

Pentru cazul de mai sus, se inițializează cele două produse cu $w_1 = 1$ și $w_2 = 1$. Vom considera pe rând variabilele din rețea:

$$P(Alergie = Nu) = 0,95$$

$$\Rightarrow w_1 \leftarrow w_1 \cdot 0,95 = 0,95, w_2 \leftarrow w_2 \cdot 0,95 = 0,95$$

$$P(Gripă = Nu) = 0,9$$

$$\Rightarrow w_1 \leftarrow w_1 \cdot 0,9 = 0,855, w_2 \leftarrow w_2 \cdot 0,9 = 0,855$$

$$P(Febră = Absență) = 0,9$$

$$\Rightarrow w_1 \leftarrow w_1 \cdot 0,9 = 0,7695, w_2 \leftarrow w_2 \cdot 0,9 = 0,7695$$

$$P(Oboseală = Da) = 0,1$$

$$\Rightarrow w_1 \leftarrow w_1 \cdot 0,1 = 0,07695, w_2 \text{ nu se modifică (Oboseala este evidență)}$$

$$P(Anorexie = Nu) = 0,9$$

$$\Rightarrow w_1 \leftarrow w_1 \cdot 0,9 = 0,069255, w_2 \leftarrow w_2 \cdot 0,9 = 0,69255$$

$$P(Rinoree = Da) = 0,1$$

$\Rightarrow w_1 \leftarrow w_1 \cdot 0,1 = 0,0069255$, w_2 nu se modifică (*Rinoreea* este evidență)

Prin urmare, ponderea cazului este:

$$w(s) = \frac{w_1}{w_2} = \frac{0,0069255}{0,69255} = 0,01.$$

Se repetă procesul pentru un număr prestabilit de eșantioane. În continuare, vom considera 10 eșantioane. Pentru rezultate semnificative, numărul trebuie să fie mult mai mare. De asemenea, pentru a sublinia faptul că rezultatele sortării topologice nu sunt unice, vom considera ordinea următoare: { *A, G, F, O, X, R* }.

Eșantion 1: Nu, Nu, Absentă, Da, Nu, Da

Variabila: Alergie (interrogare), $P = 0,95$, $w_1 = 0,95$, $w_2 = 0,95$

Variabila: Gripă (interrogare), $P = 0,9$, $w_1 = 0,855$, $w_2 = 0,855$

Variabila: Febră (interrogare), $P = 0,9$, $w_1 = 0,7695$, $w_2 = 0,7695$

Variabila: Oboseală (evidență), $P = 0,1$, $w_1 = 0,07695$

Variabila: Anorexie (interrogare), $P = 0,9$, $w_1 = 0,069255$, $w_2 = 0,69255$

Variabila: Rinoree (evidență), $P = 0,1$, $w_1 = 0,0069255$

$$w_1 = 0,0069255, w_2 = 0,69255 \Rightarrow w = 0,01$$

Eșantion 2: Nu, Nu, Absentă, Da, Nu, Da

$$w_1 = 0,0069255, w_2 = 0,69255 \Rightarrow w = 0,01$$

Eșantion 3: Nu, Nu, Absentă, Da, Nu, Da

$$w_1 = 0,0069255, w_2 = 0,69255 \Rightarrow w = 0,01$$

Eșantion 4: Nu, Nu, Absentă, Da, Nu, Da

w1 = 0,0069255, w2 = 0,69255 ⇒ w = 0,01

Eșantion 5: Da, Nu, Absentă, Da, Da, Da

Variabila: Alergie (interogare), P = 0,05, w1 = 0,05, w2 = 0,05

Variabila: Gripă (interogare), P = 0,9, w1 = 0,045, w2 = 0,045

Variabila: Febră (interogare), P = 0,9, w1 = 0,0405, w2 = 0,0405

Variabila: Oboselă (evidență), P = 0,3, w1 = 0,01215

Variabila: Anorecie (interogare), P = 0,1, w1 = 0,001215, w2 = 0,00405

Variabila: Rinoree (evidență), P = 0,9, w1 = 0,0010935

w1 = 0,0010935, w2 = 0,00405 ⇒ w = 0,27

Eșantion 6: Nu, Da, Mică, Da, Nu, Da

Variabila: Alergie (interogare), P = 0,95, w1 = 0,95, w2 = 0,95

Variabila: Gripă (interogare), P = 0,1, w1 = 0,095, w2 = 0,095

Variabila: Febră (interogare), P = 0,25, w1 = 0,02375, w2 = 0,02375

Variabila: Oboselă (evidență), P = 0,4, w1 = 0,0095

Variabila: Anorecie (interogare), P = 0,8, w1 = 0,0076, w2 = 0,019

Variabila: Rinoree (evidență), P = 0,8, w1 = 0,00608

w1 = 0,00608, w2 = 0,019 ⇒ w = 0,32

Eșantion 7: Nu, Nu, Absentă, Da, Nu, Da

w1 = 0,0069255, w2 = 0,69255 ⇒ w = 0,01

Eșantion 8: Nu, Nu, Absentă, Da, Nu, Da

w1 = 0,0069255, w2 = 0,69255 ⇒ w = 0,01

Eșantion 9: Nu, Nu, Absentă, Da, Nu, Da

w1 = 0,0069255, w2 = 0,69255 ⇒ w = 0,01

Eșantion 10: Nu, Nu, Absentă, Da, Nu, Da

w₁ = 0,0069255, w₂ = 0,69255 ⇒ w = 0,01

În final, are loc o fază de normalizare, în care se calculează suma ponderilor cazurilor în care o variabilă de interogare a avut o anumită valoare, împărțită la suma ponderilor tuturor cazurilor:

$$P(Var = val) = \frac{w_T}{w_S} = \frac{\sum_{s \in T} w(s)}{\sum_{s \in S} w(s)}, \quad (4.2)$$

unde S este mulțimea tuturor eșantioanelor iar $T \subseteq S$ este submulțimea de eșantioane în care variabila Var apare cu valoarea val .

Pentru problema considerată, rezultatele obținute sunt cele de mai jos.

Alergie

Nu: w_T = 0,4, w_S = 0,67, P = 0,597

Da: w_T = 0,27, w_S = 0,67, P = 0,403

Gripă

Nu: w_T = 0,35, w_S = 0,67, P = 0,5224

Da: w_T = 0,32, w_S = 0,67, P = 0,4776

Febră

Absentă: w_T = 0,35, w_S = 0,67, P = 0,5224

Mică: w_T = 0,32, w_S = 0,67, P = 0,4776

Ridicată: w_T = 0,0, w_S = 0,67, P = 0,0

Oboseală (nod evidență)

Nu: w_T = 0,0, w_S = 0,67, P = 0,0

Da: w_T = 0,67, w_S = 0,67, P = 1,0

AnorexieNu: $w_T = 0,4$, $w_S = 0,67$, $P = 0,597$ Da: $w_T = 0,27$, $w_S = 0,67$, $P = 0,403$ **Rinoree (nod evidență)**Nu: $w_T = 0,0$, $w_S = 0,67$, $P = 0,0$ Da: $w_T = 0,67$, $w_S = 0,67$, $P = 1,0$

În tabelul 4.2 se poate vedea cum evoluează probabilitățile nodurilor atunci când variază numărul de eșantioane. Cu cât acest număr este mai mare, cu atât este mai apropiată valoarea aproximativă calculată de probabilitatea exactă.

Tabelul 4.2. Evoluția probabilităților nodurilor cu numărul de eșantioane

Variabilă / valoare	Probabilitatea (număr de eșantioane)				
	10	1.000	10.000	100.000	1.000.000
Alergie = nu	0,597	0,717	0,7589	0,7599	0,754
Alergie = da	0,403	0,283	0,2411	0,2401	0,246
Febră = mică	0,4776	0,1563	0,1583	0,1653	0,1639
Febră = ridicată	0,0	0,5487	0,543	0,5453	0,5405
Febră = absentă	0,5224	0,295	0,2987	0,2894	0,2956
Anorexie = nu	0,597	0,6794	0,6885	0,66	0,6673
Anorexie = da	0,403	0,3206	0,3115	0,34	0,3327
Gripă = nu	0,5224	0,4075	0,3806	0,3786	0,3844
Gripă = da	0,4776	0,5925	0,6194	0,6214	0,6156

Un exemplu de variație și convergență a probabilităților cu numărul de eșantioane este prezentat în figura 4.2. În afară de variabila *Febră*, care are trei valori, celelalte variabile sunt binare și de aceea graficul prezintă doar o singură valoare, cu probabilitatea p , cealaltă având probabilitatea $1 - p$.

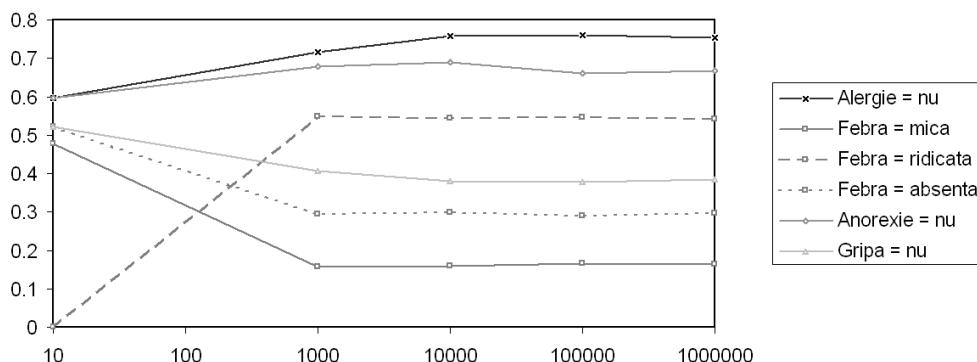


Figura 4.2. Convergența valorilor aproximative la probabilitățile exacte când numărul de eșantioane crește

Când numărul de eșantioane este mare, se vede că valorile converg, de exemplu valoarea *Nu* pentru variabila *Alergie* tinde la 75%, aşa cum s-a calculat în secțiunea 3.5.

La fel, se pot estima și probabilitățile marginale ale variabilelor, în absența evidențelor.

Datorită simplității sale, algoritmul cu ponderarea verosimilității este una din cele mai des utilizate metode de simulare pentru inferență în rețele bayesiene. De multe ori, precizia sa este comparabilă cu a metodelor mai sofisticate, întrucât poate genera mai multe eșantioane decât alți algoritmi în același interval de timp.

Atunci când probabilitatea evidenței este foarte mică, convergența sa poate fi foarte lentă. Viteza de convergență scade de asemenea când există multe noduri de evidență, deoarece probabilitatea evidenței în ansamblu scade exponențial, fiind un produs de numere subunitare. De asemenea, în lipsa unui expert, este dificil de apreciat cât de corecte sunt rezultatele oferite, pentru că metoda nu permite calcularea unor intervale de încredere bine precizate (Cheng, 2001).

4.3. Alte metode de inferență aproximativă

După cum am spus mai sus, un dezavantaj al inferenței prin ponderarea verosimilității este convergența lentă când probabilitatea evidenței este foarte mică. Una din ideile de bază ale algoritmilor mai recenti este că la început se eșantionează după o distribuție diferită de cea definită de probabilitățile (condiționate ale) nodurilor, astfel încât valorile puțin probabile să aibă șanse mai mari de apariție. Pe măsură ce numărul de eșantioane crește, distribuția după care se eșantionează converge la distribuția definită de tabelele de probabilități ale nodurilor.

Dintre algoritmii mai complecși care dau în schimb rezultate bune din punct de vedere al vitezei de convergență și a preciziei amintim *AIS-BN* (Cheng & Druzdzel, 2000) și *EPIS-BN* (Yuan & Druzdzel, 2003). Rata de convergență poate fi de asemenea îmbunătățită folosind tehnici mai elaborate pentru eșantionare, cum ar fi *hipercubul latin* (engl. “Latin hypercube”), în care distribuția de probabilitate este divizată într-un număr de intervale echiprobabile și fiecare interval este eșantionat o singură dată (Cheng & Druzdzel, 2000). Avantajul față de metoda Monte Carlo, pur aleatorie, este generarea unei mulțimi de eșantioane care reflectă mai precis forma distribuției considerate.

Teoria evidențelor

5.1. Teoria Dempster-Shafer

Având în vedere că pentru o propoziție pot exista diferite evidențe care să o susțină în grade diferite, posibil contradictorii, o primă modalitate de combinare a gradelor de încredere deriveate din surse independente de evidență a fost dezvoltată de către Dempster (1968) și studentul său, Shafer (1976). Această abordare a fost considerată foarte potrivită pentru aplicarea în sistemele expert din anii '80.

Teoria evidențelor poate fi considerată într-un fel o extensie a modelului clasic de probabilități, deoarece în locul unui singur număr, reprezentând o probabilitate, se lucrează cu intervale de încredere pentru evenimente.

Limita inferioară a intervalului care exprimă încrederea că un eveniment se poate întâmpla se numește *convingere* (engl. “belief”), notată *Bel*, iar cea superioară – *plauzibilitate* (engl. “plausibility”), notată *Pl*. Pentru un eveniment:

$$Pl(A) = 1 - Bel(\neg A). \quad (5.1)$$

Este important de subliniat faptul că aici se calculează *independent* valorile pentru *A* și non-*A*. Fiecare eveniment are un grad de susținere între 0 și 1: 0 înseamnă că nu există susținere iar 1 înseamnă o susținere totală

pentru acesta. Spre deosebire de modelul bayesian, suma convingerilor într-un eveniment și în negativul acestuia nu este 1. Ambele valori pot fi 0, dacă nu există evidențe nici pentru și nici împotriva acestuia. În consecință, dacă nu avem informații nici despre A și nici despre $\neg A$, intervalul de încredere este $[0, 1]$, în locul unei probabilități de 0,5. Pe măsură ce se acumulează informații, intervalul se micșorează, respectându-se relația:

$$Bel(A) \leq P(A) \leq Pl(A). \quad (5.2)$$

Dacă se cunoaște precis probabilitatea evenimentului, atunci intervalul se reduce la un punct și rezultatul este echivalent cu modelul clasic:

$$Bel(A) = P(A) = Pl(A). \quad (5.3)$$

De exemplu, la aruncarea unui ban, probabilitatea a-priori să iasă „cap” este 0,5. Însă neștiind nimic despre ban, orice rezultat ar putea fi posibil. În lipsa oricărora informații, intervalul de încredere este $[0,1]$:

$$Bel(cap) = 0,$$

$$Pl(cap) = 1 - Bel(\neg cap) = 1 - 0 = 1.$$

Dacă un expert afirată că este 90% sigur că banul este corect, atunci:

$$Bel(cap) = 0,9 \cdot 0,5 = 0,45,$$

$$Pl(cap) = 1 - Bel(\neg cap) = 1 - 0,9 \cdot 0,5 = 0,55.$$

Intervalul de încredere este: $[0,45, 0,55]$.

Să mai considerăm un exemplu. Două site-uri de știri relatează despre o demonstrație. Primul site are nivelul de încredere de 80% iar al doilea are nivelul de încredere de 60%. Ambele afirmă că demonstrația a fost una mare, cu peste 10000 de participanți. Intuitiv, putem considera că probabilitatea ca ambele site-uri să mintă este $(1 - 0,8) \cdot (1 - 0,6) = 0,08$. Probabilitatea ca măcar unul din site-uri să spună adevărul este $1 - 0,08 = 0,92$. Prin urmare, intervalul de încredere este: $[0,92, 1]$. Limita superioară este 1 deoarece nu există evidențe împotriva faptului că demonstrația a fost mare.

Însă poate există situația în care primul site afirmă că demonstrația a fost mare iar al doilea afirmă contrariul. Ne interesează intervalul de încredere pentru evenimentul „demonstrația a fost mare”. În acest caz, nu pot fi ambele site-uri de încredere, deoarece mărturiile se contrazic. Rămân următoarele posibilități:

- Doar primul site este de încredere (demonstrația a fost mare):
 $0,8 \cdot (1 - 0,6) = 0,32$;
- Doar al doilea site este de încredere (demonstrația nu a fost mare):
 $(1 - 0,8) \cdot 0,6 = 0,12$;
- Niciun site nu este de încredere (nu știm nimic precis):
 $(1 - 0,8) \cdot (1 - 0,6) = 0,08$.

Suma tuturor probabilităților nenule, care va servi ca factor pentru normalizare, este: $0,32 + 0,12 + 0,08 = 0,52$. Prin urmare, convingerea că demonstrația a fost mare este: $0,32/0,52 = 0,62$ iar convingerea că demonstrația nu a fost mare este: $0,12/0,52 = 0,23$. Plauzibilitatea unei

demonstrații mari este: $1 - 0,23 = 0,77$. Intervalul de încredere este deci: $[0,62, 0,77]$.

Pentru a formaliza modelul, fie Θ mulțimea tuturor ipotezelor mutual excluzive, numită și *cadrul de discernământ* (engl. “frame of discernment”).

Pentru exemplul anterior, $\Theta = \{Mare, Mică\}$.

Nivelul de încredere al unei evidențe este probabilitatea ca evidența să fie de încredere, de exemplu gradul în care fiecare din cele două site-uri pot fi crezute ($0,8$ și respectiv $0,6$).

Fie m o funcție numită *atribuire de convingeri de bază* (engl. “basic belief assignment”, BBA) sau *funcție de masă* (engl. “mass function”), $m: \wp(\Theta) \rightarrow [0, 1]$, unde $\wp(\Theta)$ este mulțimea părților lui Θ (mulțimea de mulțimi care se pot forma din elementele lui Θ).

Valorile lui m , de tipul $m(A)$ se numesc *mase de convingeri de bază* (engl. “basic belief masses”, BBM). Aplicând relațiile teoriei Dempster-Shafer, vom avea întotdeauna:

$$\sum_{A \in \wp(\Theta)} m(A) = 1. \quad (5.4)$$

După cum am menționat, teoria ne permite să combinăm convingerile care apar din surse multiple de evidență. Ecuația fundamentală Dempster-Shafer pentru combinarea evidențelor date de m_1 și m_2 într-o nouă atribuire de convingeri de bază m_3 este:

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) \cdot m_2(Y)}. \quad (5.5)$$

Se consideră că $m(\emptyset) = 0$.

Pentru exemplul cu demonstrația, când evidențele celor două site-uri coincid, vom avea:

$$m_1(\{\text{Mare}\}) = 0,8$$

$$m_1(\Theta) = 0,2$$

$$m_2(\{\text{Mare}\}) = 0,6$$

$$m_2(\Theta) = 0,4.$$

Se observă că incertitudinea rămasă după precizarea explicită a tuturor evidențelor se atribuie cadrului de discernământ Θ .

Aplicarea regulii de combinare a evidențelor se poate face mai ușor considerând un tabel de forma:

X	$m_1(X)$	Y	$m_2(Y)$	\cap	\times
{Mare}	0,8	{Mare}	0,6	{Mare}	0,48
{Mare}	0,8	Θ	0,4	{Mare}	0,32
Θ	0,2	{Mare}	0,6	{Mare}	0,12
Θ	0,2	Θ	0,4	Θ	0,08

Penultima coloană reprezintă intersecția dintre mulțimile de elemente X și Y , iar ultima coloană indică produsul maselor de convingeri de bază corespunzătoare.

Întrucât evidențele nu se contrazic, nu apare în intersecție mulțimea vidă. În consecință, valoarea numitorului din ecuația 5.5 va fi: $1 - 0 = 1$.

Din tabel rezultă noua funcție de masă m_3 :

$$m_3(\{\text{Mare}\}) = 0,48 + 0,32 + 0,12 = 0,92$$

$$m_3(\Theta) = 0,08.$$

Prin urmare:

$$\text{Bel}(\{\text{Mare}\}) = 0,92$$

$$\text{Pl}(\{\text{Mare}\}) = 1 - \text{Bel}(\{\text{Mică}\}) = 1.$$

Intervalul de încredere pentru demonstrația mare este același ca mai sus: $[0,92, 1]$.

În al doilea caz, în care primul site afirmă că demonstrația a fost una mare, iar al doilea că a fost una mică, vom avea:

$$m_1(\{\text{Mare}\}) = 0,8$$

$$m_1(\emptyset) = 0,2$$

$$m_2(\{\text{Mică}\}) = 0,6$$

$$m_2(\emptyset) = 0,4.$$

Trebuie să subliniem faptul că ar fi incorrect să considerăm $m_2(\{\text{Mare}\}) = 0,4$, deoarece al doilea site a spus doar că demonstrația a fost mică, nu a spus nimic despre o demonstrație mare.

Tabelul pentru calcule va fi următorul:

X	$m_1(X)$	Y	$m_2(Y)$	\cap	\times
$\{\text{Mare}\}$	0,8	$\{\text{Mică}\}$	0,6	\emptyset	0,48
$\{\text{Mare}\}$	0,8	\emptyset	0,4	$\{\text{Mare}\}$	0,32
\emptyset	0,2	$\{\text{Mică}\}$	0,6	$\{\text{Mică}\}$	0,12
\emptyset	0,2	\emptyset	0,4	\emptyset	0,08

Având în vedere că valoarea asociată mulțimii vide este 0,48, numitorul fracției va fi $1 - 0,48 = 0,52$.

Funcția de masă m_3 va fi definită astfel:

$$m_3(\{Mare\}) = 0,32 / 0,52 = 0,62$$

$$m_3(\{Mică\}) = 0,12 / 0,52 = 0,23$$

$$m_3(\emptyset) = 0,08 / 0,52 = 0,15.$$

Prin urmare:

$$\text{Bel}(\{Mare\}) = 0,62$$

$$\text{Bel}(\{Mică\}) = 0,23$$

$$\text{Pl}(\{Mare\}) = 1 - \text{Bel}(\neg\{Mare\}) = 1 - \text{Bel}(\emptyset \setminus \{Mare\}) =$$

$$1 - \text{Bel}(\{Mică\}) = 0,77$$

$$\text{Pl}(\{Mică\}) = 1 - \text{Bel}(\{Mare\}) = 0,38.$$

Intervalele de încredere sunt: pentru $\{Mare\} - [0,62, 0,77]$ iar pentru $\{Mică\} - [0,23, 0,38]$.

5.2. Surse multiple de evidență

În cele ce urmează, vom considera un exemplu mai complex pentru a urmări modul în care se combină succesiv mai multe evidențe. Să presupunem că un pacient internat în spital poate avea răceală, gripă, meningită sau indigestie.

Vom nota acest lucru astfel: $\Theta = \{R, G, M, I\}$.

Pacientul are febră și grețuri. Din studii anterioare, știm că febra susține răceala sau gripe la un nivel de 60%, meningita la un nivel de 20% iar indigestia la un nivel de 10%. Formal, vom avea:

$$m_1(\{R,G\}) = 0,6$$

$$m_1(\{M\}) = 0,2$$

$$m_1(\{I\}) = 0,1$$

$$m_1(\emptyset) = 0,1.$$

De asemenea, grețurile susțin ipoteza meningitei sau indigestiei la nivelul de 70%:

$$m_2(\{M,I\}) = 0,7$$

$$m_2(\emptyset) = 0,3.$$

În tabelul următor se prezintă modul de combinare a acestor două evidențe inițiale:

X	$m_1(X)$	Y	$m_2(Y)$	\cap	\times
$\{R,G\}$	0,6	$\{M,I\}$	0,7	\emptyset	0,42
$\{R,G\}$	0,6	\emptyset	0,3	$\{R,G\}$	0,18
$\{M\}$	0,2	$\{M,I\}$	0,7	$\{M\}$	0,14
$\{M\}$	0,2	\emptyset	0,3	$\{M\}$	0,06
$\{I\}$	0,1	$\{M,I\}$	0,7	$\{I\}$	0,07
$\{I\}$	0,1	\emptyset	0,3	$\{I\}$	0,03
\emptyset	0,1	$\{M,I\}$	0,7	$\{M,I\}$	0,07
\emptyset	0,1	\emptyset	0,3	\emptyset	0,03

Valoarea asociată mulțimii vide este 0,42 și deci numitorul fracției va fi $1 - 0,42 = 0,58$.

Funcția de masă m_3 va fi definită astfel:

$$m_3(\{R,G\}) = 0,18 / 0,58 = 0,3103$$

$$m_3(\{M\}) = (0,14 + 0,06) / 0,58 = 0,3448$$

$$m_3(\{I\}) = (0,07 + 0,03) / 0,58 = 0,1724$$

$$m_3(\{M,I\}) = 0,07 / 0,58 = 0,1207$$

$$m_3(\emptyset) = 0,03 / 0,58 = 0,0517.$$

Din masele de convingeri de bază putem deduce convingerile:

$$\text{Bel}(\{R,G\}) = 0,3103$$

$$\text{Bel}(\{M\}) = 0,3448$$

$$\text{Bel}(\{I\}) = 0,1724$$

$$\text{Bel}(\{M,I\}) = 0,1207 + 0,3448 + 0,1724 = 0,6379.$$

Pentru calculul lui $\text{Bel}(\{M,I\})$ se observă că se iau în calcul toate submulțimile mulțimii $\{M, I\}$, respectiv $\{M\}$, $\{I\}$ și $\{M, I\}$.

Convingerea asociată unei mulțimi este suma maselor tuturor submulțimilor acesteia.

Putem calcula apoi plauzibilitățile mulțimilor:

$$\text{Pl}(\{R,G\}) = 1 - \text{Bel}(\{M,I\}) = 1 - 0,6379 = 0,3621$$

$$\text{Pl}(\{M\}) = 1 - \text{Bel}(\{R,G,I\}) = 1 - (m_3(\{R,G\}) + m_3(\{I\})) = 0,5173$$

$$\text{Pl}(\{I\}) = 1 - \text{Bel}(\{R,G,M\}) = 1 - (m_3(\{R,G\}) + m_3(\{M\})) = 0,3449$$

$$\text{Pl}(\{M,I\}) = 1 - \text{Bel}(\{R,G\}) = 0,6897.$$

Intervalele de încredere vor fi cele de mai jos:

$$\{R,G\}: [0,3103, 0,3621]$$

$$\{M\}: [0,3448, 0,5173]$$

$$\{I\}: [0,1724, 0,3449]$$

$$\{M,I\}: [0,6379, 0,6897].$$

Să considerăm acum că pacientul face un test care sprijină la nivelul 99% faptul că pacientul nu are meningită, ceea ce înseamnă că rămân celelalte trei posibilități – răceală, gripă sau indigestie:

$$m_4(\{R,G,I\}) = 0,99$$

$$m_4(\emptyset) = 0,01.$$

Testul spune doar că *nu este* meningită, nu spune nimic despre probabilitatea de *a fi* meningită. După cum am precizat, în teoria evidențelor posibilitățile A și $\neg A$ sunt considerate independent.

Interpretarea este diferită față de cazul în care am considerat $m_4(\{M\}) = 0,01$. În această din urmă situație, intervalele ar rămâne aproape la fel, cu o foarte ușoară creștere a încrederii lui M .

Tabelul care ne ajută să adăugăm noua evidență este prezentat mai jos:

X	$m_3(X)$	Y	$m_4(Y)$	\cap	\times
$\{R,G\}$	0,3103	$\{R,G,I\}$	0,99	$\{R,G\}$	0,3072
$\{M\}$	0,3448	$\{R,G,I\}$	0,99	\emptyset	0,3414
$\{I\}$	0,1724	$\{R,G,I\}$	0,99	$\{I\}$	0,1707
$\{M,I\}$	0,1207	$\{R,G,I\}$	0,99	$\{I\}$	0,1195
Θ	0,0517	$\{R,G,I\}$	0,99	$\{R,G,I\}$	0,0512
$\{R,G\}$	0,3103	Θ	0,01	$\{R,G\}$	0,0031
$\{M\}$	0,3448	Θ	0,01	$\{M\}$	0,0034
$\{I\}$	0,1724	Θ	0,01	$\{I\}$	0,0017
$\{M,I\}$	0,1207	Θ	0,01	$\{M,I\}$	0,0012
Θ	0,0517	Θ	0,01	Θ	0,0005

Valoarea asociată mulțimii vide este 0,3414 și deci numitorul fracției va fi 0,6586.

Funcția de masă m_5 va fi definită astfel:

$$m_5(\{R,G\}) = (0,3072 + 0,0031) / 0,6586 = 0,4712$$

$$m_5(\{M\}) = 0,0034 / 0,6586 = 0,0052$$

$$m_5(\{I\}) = (0,1707 + 0,1195 + 0,0017) / 0,6586 = 0,4432$$

$$m_5(\{M,I\}) = 0,0012 / 0,6586 = 0,0018$$

$$m_5(\{R,G,I\}) = 0,0512 / 0,6586 = 0,0777$$

$$m_5(\emptyset) = 0,0005 / 0,6586 = 0,0008.$$

Convingerile mulțimilor vor fi:

$$\text{Bel}(\{R,G\}) = 0,4712$$

$$\text{Bel}(\{M\}) = 0,0052$$

$$\text{Bel}(\{I\}) = 0,4432$$

$$\text{Bel}(\{M,I\}) = 0,0052 + 0,0018 = 0,007$$

$$\text{Bel}(\{R,G,I\}) = 0,4712 + 0,4432 + 0,0777 = 0,9921.$$

Plauzibilitățile mulțimilor vor fi:

$$\text{Pl}(\{R,G\}) = 1 - \text{Bel}(\{M,I\}) = 1 - 0,007 = 0,993$$

$$\text{Pl}(\{M\}) = 1 - \text{Bel}(\{R,G,I\}) = 1 - 0,9921 = 0,0079$$

$$\text{Pl}(\{I\}) = 1 - \text{Bel}(\{R,G,M\}) = 1 - (0,4712 + 0,0052) = 0,5236$$

$$\text{Pl}(\{M,I\}) = 1 - \text{Bel}(\{R,G\}) = 1 - 0,4712 = 0,5288$$

$$\text{Pl}(\{R,G,I\}) = 1 - \text{Bel}(\{M\}) = 1 - 0,0052 = 0,9948.$$

Intervallele de încredere ale mulțimilor sunt acum următoarele:

- $\{R,G\}$: [0,4712, 0,993]
- $\{M\}$: [0,0052, 0,0079]
- $\{I\}$: [0,4432, 0,5236]
- $\{M,I\}$: [0,007, 0,5288]
- $\{R,G,I\}$: [0,9921, 0,9948].

Tabelul următor arată cum s-au modificat intervallele de încredere prin adăugarea evidenței testului care a exclus meningita:

X	<i>Interval de încredere</i> m_3	<i>Interval de încredere</i> m_5
$\{R,G\}$	[0,3103, 0,3621]	[0,4712, 0,993]
$\{M\}$	[0,3448, 0,5173]	[0,0052, 0,0079]
$\{I\}$	[0,1724, 0,3449]	[0,4432, 0,5236]
$\{M,I\}$	[0,6379, 0,6897]	[0,007, 0,5288]
$\{R,G,I\}$		[0,9921, 0,9948]

5.3. Reguli alternative de combinare a evidențelor

Atunci când evidențele sunt conflictuale, rezultatele obținute prin aplicarea regulii de combinare a lui Dempster pot fi contraintuitive și neplauzibile. O astfel de situație este cea din exemplul următor (Zadeh, 1979). Un pacient este examinat de doi medici, care stabilesc că ar putea avea meningită (M), o contuzie (C) sau o tumoare pe creier (T). Ambii medici consideră că tumoarea este improbabilă, în schimb nu se pun de acord asupra diagnosticului cel mai probabil, rezultând situația următoare:

$$\Theta = \{M, C, T\}$$

$$m_1(\{M\}) = 0,99$$

$$m_1(\{T\}) = 0,01$$

$$m_2(\{C\}) = 0,99$$

$$m_2(\{T\}) = 0,01.$$

Tabelul indică modul de combinare a celor două evidențe:

X	$m_3(X)$	Y	$m_4(Y)$	\cap	\times
{M}	0,99	{C}	0,99	\emptyset	0,9801
{M}	0,99	{T}	0,01	\emptyset	0,0099
{T}	0,01	{C}	0,99	\emptyset	0,0099
{T}	0,01	{T}	0,01	{T}	0,0001

Valoarea corespunzătoare mulțimii vide este în acest caz $0,9801 + 0,0099 + 0,0099 = 0,9999$ și deci numărătorul fracției va fi 0,0001.

Funcția de masă m_3 va fi doar:

$$m_3(\{T\}) = 0,0001 / 0,0001 = 1.$$

Rezultatul este constraintiv, deoarece tăruirea apare ca sigură deși ambi medici au considerat-o foarte improbabilă.

5.3.1. Regula lui Yager

În cazul în care mai multe evidențe trebuie combinate simultan, regula lui Yager (1987) este:

$$m_{n+1}(Z) = \sum_{\bigcap_{i=1}^n X_i = Z} m_1(X_1) \cdot \dots \cdot m_n(X_n). \quad (5.6)$$

Spre deosebire de regula lui Dempster, $m(\emptyset) \geq 0$.

Regula lui Yager nu normalizează conflictul, ci îl adaugă la mulțimea Θ .

Pentru combinarea a două surse de evidență se folosesc următoarele relații:

$$m_3(Z) = \sum_{X \cap Y = Z \neq \emptyset} m_1(X) \cdot m_2(Y), \quad (5.7)$$

$$m_3(\Theta) = m_1(\Theta) \cdot m_2(\Theta) + \sum_{X \cap Y = \emptyset} m_1(X) \cdot m_2(Z). \quad (5.8)$$

În cazul exemplului propus de Zadeh, vom avea:

$$m_3(\{M\}) = 0$$

$$m_3(\{C\}) = 0$$

$$m_3(\{T\}) = 0,01 \cdot 0,01 = 0,0001$$

$$m_3(\Theta) = 0 + (0,9801 + 0,0099 + 0,0099) = 0,9999.$$

Pentru exemplul cu cele două raportări ale demonstrației, atunci când evidențele coincid (ambele surse raportează o demonstrație mare), rezultatele sunt identice cu cele ale regulii lui Dempster, întrucât numitorul pentru normalizare este 1:

X	$m_3(X)$	Y	$m_4(Y)$	\cap	\times
{Mare}	0,8	{Mare}	0,6	{Mare}	0,48
{Mare}	0,8	Θ	0,4	{Mare}	0,32
Θ	0,2	{Mare}	0,6	{Mare}	0,12
Θ	0,2	Θ	0,4	Θ	0,08

Prin urmare:

$$m_3(\{Mare\}) = 0,48 + 0,32 + 0,12 = 0,92$$

$$m_3(\Theta) = 0,08 + 0 = 0,08.$$

Când evidențele diferă, vom avea situația de mai jos:

X	$m_3(X)$	Y	$m_4(Y)$	\cap	\times
{Mare}	0,8	{Mică}	0,6	\emptyset	0,48
{Mare}	0,8	Θ	0,4	{Mare}	0,32
Θ	0,2	{Mică}	0,6	{Mică}	0,12
Θ	0,2	Θ	0,4	Θ	0,08

și deci:

$$m_3(\{Mare\}) = 0,32$$

$$m_3(\{Mică\}) = 0,12$$

$$m_3(\Theta) = 0,08 + 0,48 = 0,56.$$

În consecință, convingerile și plauzibilitățile se modifică:

$$\text{Bel}(\{Mare\}) = 0,32$$

$$\text{Pl}(\{Mare\}) = 1 - \text{Bel}(\{Mică\}) = 1 - 0,12 = 0,88.$$

Intervalurile de încredere sunt acum:

{Mare}: [0,32, 0,88] față de [0,62, 0,77] după regula lui Dempster;

{Mică}: [0,12, 0,68] față de [0,23, 0,38] după regula lui Dempster.

În cazul rezultatelor folosind regula lui Yager, se poate observa că diferența dintre convingere și plauzibilitate este egală cu masa de convingeri de bază atribuită mulțimii Θ .

5.3.2. Regula Han-Han-Yang

În acest caz, formula de combinare a două evidențe este următoarea (Han, Han & Yang, 2008):

$$m_3(Z) = \sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y) + w_1 \cdot \sum_{Z \cap A = \emptyset} m_1(Z) \cdot m_2(A) + \\ w_2 \cdot \sum_{B \cap Z = \emptyset} m_1(B) \cdot m_2(Z), \quad (5.9)$$

unde w_1 și w_2 se calculează ca mai jos.

Vom introduce *funcția de probabilitate pignistică* (în latină „pignus” însemnând pariu), care definește probabilitățile pe care o persoană rațională le atrbuie unor opțiuni atunci când trebuie să ia o decizie, de exemplu să facă un pariu (Smets & Kennes, 1994):

$$B_i(A) = \sum_{T \subseteq \Theta} m_i(T) \cdot \frac{|A \cap T|}{|T|} \quad (5.10)$$

Entropia definită de valorile funcției este:

$$E_i = - \sum_{\theta \in \Theta} B_i(\theta) \cdot \log_2(B_i(\theta)) \quad (5.11)$$

Astfel, se definesc ponderile celor două evidențe care se combină:

$$w_1 = \frac{e^{-E_1}}{e^{-E_1} + e^{-E_2}} \quad (5.12)$$

$$w_2 = 1 - w_1 \quad (5.13)$$

Pentru exemplul propus de Zadeh, vom avea:

$$m_3(\{M\}) = 0 + w_1 \cdot (0,9801 + 0,0099) + w_2 \cdot 0$$

$$\begin{array}{ccc} & \nearrow & \nearrow \\ \{M\} \cap \{C\} = \emptyset & & \{M\} \cap \{T\} = \emptyset \end{array}$$

$$m_3(\{C\}) = 0 + w_1 \cdot 0 + w_2 \cdot (0,9801 + 0,0099)$$

$$\begin{array}{ccc} & \nearrow & \nearrow \\ \{M\} \cap \{\underline{C}\} = \emptyset & & \{T\} \cap \{\underline{C}\} = \emptyset \end{array}$$

$$m_3(\{T\}) = 0,0001 + w_1 \cdot 0,0099 + w_2 \cdot 0,0099$$

$$\begin{array}{ccc} & \nearrow & \nearrow \\ \{\underline{T}\} \cap \{C\} = \emptyset & & \{M\} \cap \{\underline{T}\} = \emptyset \end{array}$$

$$B_1(M) = m_1(T) \cdot \frac{|M \cap T|}{|T|} = 0,01 \cdot \frac{0}{1} = 0$$

$$B_1(T) = m_1(T) \cdot \frac{|T \cap T|}{|T|} = 0,01$$

Analog,

$$B_2(C) = m_2(T) \cdot \frac{|C \cap T|}{|T|} = 0$$

$$B_2(T) = m_2(T) \cdot \frac{|T \cap T|}{|T|} = 0,01$$

În acest caz, este clar că $E_1 = E_2$ și deci $w_1 = w_2 = 0,5$.

Prin urmare:

$$m_3(\{M\}) = 0 + 0,5 \cdot 0,99 = 0,495$$

$$m_3(\{C\}) = 0,495$$

$$m_3(\{T\}) = 0,01$$

Pentru exemplul cu demonstrația despre care o sursă spune că a fost mare și cealaltă spune că a fost mică, vom avea:

$$m_3(\{Mare\}) = 0,8 \cdot 0,4 + w_1 \cdot 0,8 \cdot 0,6$$

$$m_3(\{Mică\}) = 0,2 \cdot 0,6 + w_2 \cdot 0,8 \cdot 0,6$$

$$B_1(\{Mare\}) = m_1(\{Mare\}) \cdot 1 + m_1(\{Mică\}) \cdot 0 = 0,8$$

$$B_2(\{Mică\}) = m_2(\{Mică\}) \cdot 1 + m_2(Mare) \cdot 0 = 0,6$$

$$E_1 = -(0,8 \cdot \log_2 0,8) = 0,3798$$

$$E_2 = -(0,6 \cdot \log_2 0,6) = 0,6521$$

$$w_1 = \frac{e^{-0,3798}}{e^{-0,3798} + e^{-0,6521}} = 0,568$$

$$w_2 = 1 - 0,568 = 0,432$$

Prin urmare:

$$m_3(\{Mare\}) = 0,593$$

$$m_3(\{Mică\}) = 0,327$$

$$m_3(\emptyset) = 0,08$$

Intervalle de încredere sunt:

$$\{Mare\}: [0,593, 0,673]$$

$$\{Mică\}: [0,327, 0,407].$$

Diferența dintre cele două valori este și aici egală cu $m_3(\emptyset)$.

5.4. Concluzii

Aplicând regula de combinare a lui Dempster, ordinea în care apar informațiile influențează rezultatul. Există reguli de combinare alternative neinfluențate de acest aspect.

Într-un studiu privind combinarea informațiilor oferite de o mulțime de senzori (engl. “sensor fusion”), s-a constatat că regula lui Yager a dat cele mai bune rezultate în comparație cu alte reguli (Seo & Sycara, 2006).

Există și alte modalități de combinare a evidențelor, însă avantajele oferite pentru majoritatea problemelor practice sunt discutabile, având în vedere menținerea unui echilibru între complexitatea calculelor și credibilitatea rezultatelor.

Partea a II-a

Tehnici de clasificare

Problematica generală

6.1. Introducere

Dincolo de aplicațiile practice ale tehnicielor de clasificare ce urmează a fi prezentate, acestea au o foarte mare importanță pentru că dău o imagine asupra modului în care gândește omul în relația cu mediul înconjurător. Mediul fiind foarte complex, creierul trebuie să selecteze anumite informații, construind modele pentru a reduce numărul și diversitatea stimulilor. Clasificarea (sau categorizarea) stabilește clase care includ un grup de obiecte cu attribute comune. Modul în care creierul prelucrează informațiile din mediu nu este unitar, având părți componente diferite care lucrează diferit. Pentru unele aspecte creăm modele bazate pe reguli explicite (de exemplu, „dacă semaforul este roșu, atunci trebuie să aşteptăm”). În alte situații, lucrăm prin analogie. Poate nu știm exact regula după care se clasifică un măr ca fiind măr, dar implicit știm că un anumit obiect este apropiat ca și culoare, formă sau dimensiuni față de un prototip de măr, cunoscut. Când aplicăm analogii cu situații întâlnite sau aspecte văzute anterior, uneori este greu să explicăm exact de ce. Am putea, însă ar trebui să depunem un efort, deoarece are loc un proces de căutare pentru a extrage reguli explicite din modelul bazat pe analogie. Regulile sunt verbalizabile, de aceea sunt în general mai concise pentru a fi înțelese și reținute. Clasificarea prin similaritate poate lua în calcul, în mod implicit, un număr mai mare de aspecte. Mai există și abordarea probabilistică întrucât

dăm importanță mai mare de obicei lucrurilor care se întâmplă mai frecvent. Toate aceste metode reflectă modalități diferite pe care le folosește omul în propriul raționament.

Tehnicile de *învățarea automată* (engl. “machine learning”) aparțin tipului de raționament inductiv. Învățăm din mediu și încercăm să tragem niște concluzii generale pe baza a ceea ce experimentăm practic. Nu avem o teorie generală, avem doar date provenite din experiență și încercăm să conturăm un model general, dacă el există.

După ce este găsit și verificat un astfel de model, se poate aplica raționamentul deductiv, de exemplu aplicarea unei teoreme cunoscute la un caz concret. Acum direcția de raționament este inversă, de la cazul general la cazul particular pe care dorim să îl rezolvăm.

Din punct de vedere practic, *învățarea automată* este motivată de faptul că un sistem clasic specializat dar care nu învăță de obicei realizează calcule numeroase pentru rezolvarea unei probleme, însă nu memorează soluția și de aceea, de fiecare dată când are nevoie de soluție, realizează aceeași secvență de calcule complexe. Dacă nu învăță, comportamentul (calculele) se repetă de fiecare dată.

Învățarea înseamnă modul în care se schimbă sistemul, astfel încât să rezolve aceeași problemă mai eficient (cu performanțe superioare sau cu mai puține resurse), dar și alte probleme diferite, deși asemănătoare (definiție adaptată după Simon, 1983). Sistemul trebuie să generalizeze pentru a putea rezolva probleme noi.

De asemenea, putem spune că un sistem învăță dacă își îmbunătățește performanțele la îndeplinirea unei sarcini pe baza experienței, din punct de vedere al costurilor sau al calității (definiție adaptată după Mitchell, 1997).

Modelele sunt antrenate cu niște date, însă apoi sunt aplicate pe alte date, noi. Dacă nu am face aşa, ar fi suficientă stocarea propriu-zisă a răspunsurilor, echivalentă cu învățarea pe de rost.

6.2. Învățarea supervizată

Învățarea supervizată presupune învățarea unei ipoteze, aproximarea unei funcții, în sensul cel mai larg (funcția nu trebuie să fie neapărat reală, poate fi de exemplu și dacă cineva își plătește un credit luat de la bancă sau nu).

Fie f funcția țintă. Datele cunoscute sunt niște exemple, niște eșantionări ale funcției, o mulțime de perechi $(\mathbf{x}, f(\mathbf{x}))$.

Scopul este găsirea unei ipoteze h astfel încât $h \approx f$, pentru toate elementele mulțimii de exemple de antrenare.

Dacă valorile funcției sunt reale, discutăm despre o problemă de *regresie*. Dacă valorile funcției sunt discrete, discutăm despre o problemă de *clasificare*.

Acest tip de învățare se numește „supervizată” deoarece pentru fiecare instanță se cunoaște valoarea funcției $f(\mathbf{x})$, pe lângă valorile vectorului de intrare \mathbf{x} .

Să considerăm o funcție reală, pentru care cunoaștem punctele din tabelul 6.1.

Tabelul 6.1. Exemplu de funcție eșantionată pentru regresie

\mathbf{x}	$f(\mathbf{x})$
0,50	0,25
1,00	1,00
1,50	0,50
2,00	4,00

În cazul regresiei, problema este găsirea unei funcții de interpolare. De exemplu, putem folosi o funcție liniară (figura 6.1). Desigur, avem erori.

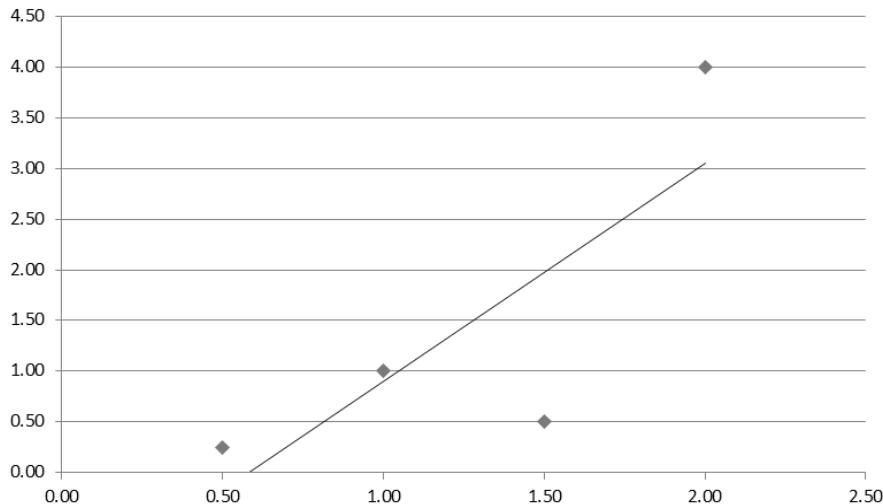


Figura 6.1. Regresie cu o funcție liniară

Putem folosi o funcție polinomială de gradul 2 (figura 6.2).

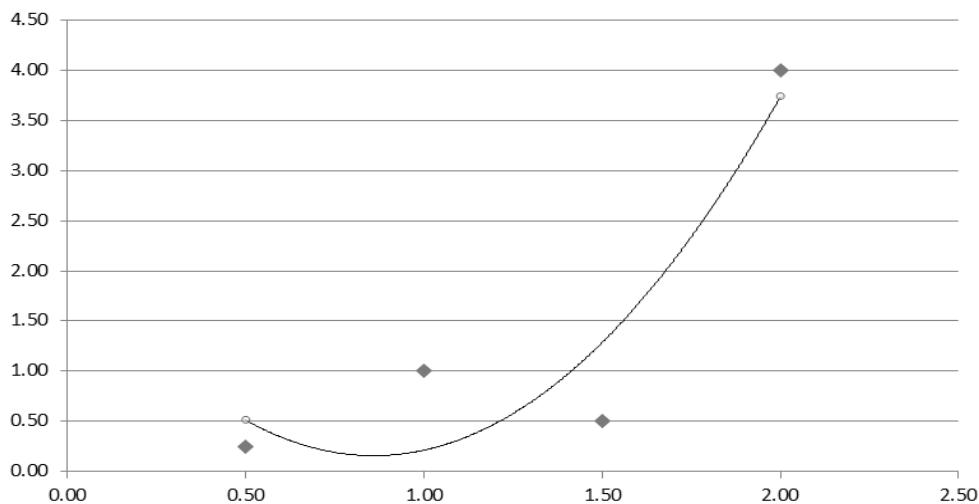


Figura 6.2. Regresie cu o funcție polinomială de gradul 2

Putem folosi o funcție polinomială de gradul 3 (figura 6.3).

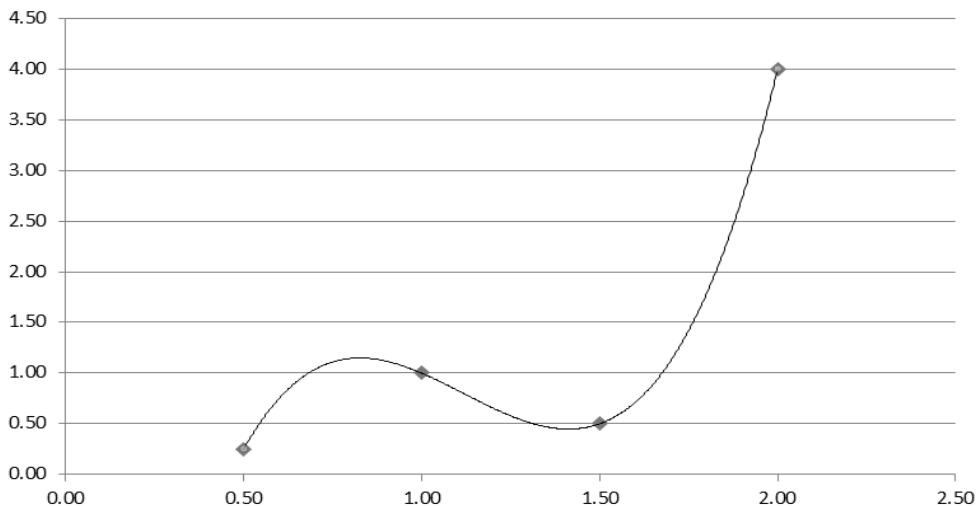


Figura 6.3. Regresie cu o funcție polinomială de gradul 3

De asemenea, putem folosi o funcție polinomială de grad mai mare (figura 6.4).

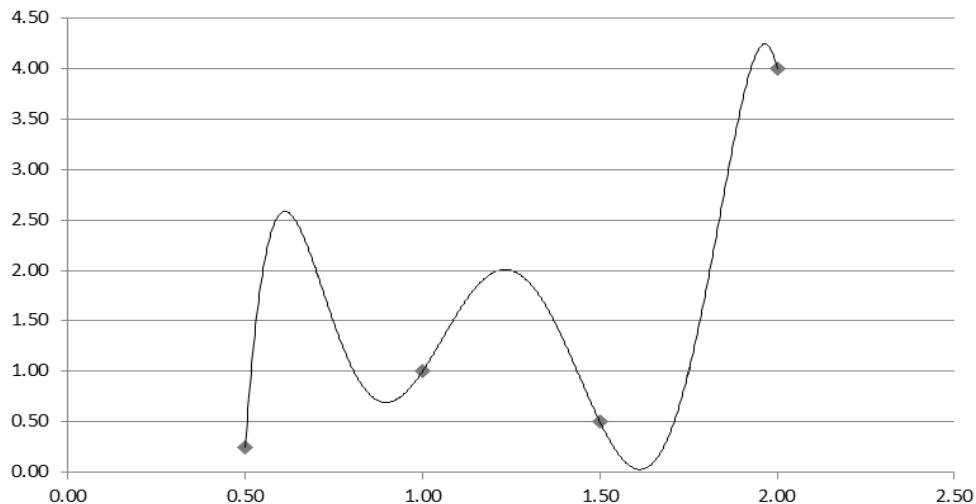


Figura 6.4. Regresie cu o funcție polinomială de gradul 6

Funcțiile date de polinoamele de grad 1 și 2 sunt prea simple, dar de la gradul 3 în sus, toate funcțiile trec prin punctele specificate. Problema este următoarea: care este cea mai bună interpolare?

O recomandare în acest sens este *briciul lui Occam*. Occam a fost un călugăr franciscan englez care a trăit în secolul al XIV-lea și a propus *legea economiei* (lat. „lex parsimoniae”): entitățile nu trebuie multiplicate dincolo de necesitate.

Ideea de bază este că la egalitate, când mai multe modele au aceleași performanțe, trebuie preferat modelul mai simplu. Cu alte cuvinte, nu trebuie să complicăm lucrurile fără rost.

Pentru exemplele de mai sus, briciul lui Occam ar recomanda utilizarea polinomului de gradul 3, deoarece este cel mai simplu model care trece corect prin toate punctele.

Un model prea complicat conduce la fenomenul de *suprapotrivire* (engl. “overfitting”). Trece foarte bine prin punctele date, dar pentru punctele intermediare nu se comportă foarte bine.

Briciul lui Occam este un principiu util, însă nu trebuie aplicat mecanic. De exemplu, există algoritmul *AdaBoost* (Freund & Schapire, 1995), care este o modalitate de a transforma orice algoritm de clasificare slab într-un clasificator puternic (spunem că este un meta-algoritm). Ideea de bază este executarea mai multor runde de clasificare. În fiecare rundă, este aplicat clasificatorul slab și se determină instanțele clasificate greșit. Acestea li se dă o pondere mai mare în următoarea rundă de clasificare. Rezultă o serie de clasificatori, fiecare cu anumite ponderi, care în final dau câte un vot proporțional cu ponderea pentru clasificarea unei noi instanțe. În acest caz, creșterea numărului de runde, echivalentă cu creșterea

complexității modelului, în general nu determină scăderea capacitații de generalizare.

Prin urmare, briciul lui Occam este o euristică, nu o lege fundamentală.

6.3. Definirea unei probleme de clasificare

În continuare, ne vom concentra asupra problemelor de clasificare, mai apropiate de modul în care omul prelucreză informațiile. Trebuie spus totuși că și problemele de regresie sunt foarte importante din punct de vedere matematic și pentru numeroare aplicații din lumea reală.

În general, structura unei probleme de clasificare este definită de un tabel de tipul tabelului 6.2. Aici este prezentat un exemplu adaptat după Quinlan (1986) pe care îl vom folosi pentru a descrie toate metodele de clasificare din capitolele următoare. Se pune problema de a determina dacă în funcție de condițiile meteorologice se poate juca sau nu golf.

Tabelul 6.2. Problemă de clasificare

Starea vremii	Temperatură	Umiditate	Vânt	Joc
Soare	Mare	Mare	Absent	Nu
Soare	Mare	Mare	Prezent	Nu
Înnorat	Mare	Mare	Absent	Da
Ploaie	Medie	Mare	Absent	Da
Ploaie	Mică	Normală	Absent	Da
Ploaie	Mică	Normală	Prezent	Nu
Înnorat	Mică	Normală	Prezent	Da
Soare	Medie	Mare	Absent	Nu
Soare	Mică	Normală	Absent	Da
Ploaie	Medie	Normală	Absent	Da
Soare	Medie	Normală	Prezent	Da
Înnorat	Medie	Mare	Prezent	Da
Înnorat	Mare	Normală	Absent	Da
Ploaie	Medie	Mare	Prezent	Nu

Pe linii avem *instanțele*, care se cunosc și în baza cărora se realizează modelul de clasificare. Pentru problema de față, sunt situații din trecut, pentru care știm dacă s-a putut juca sau nu golf. Tabelul întreg reprezintă *multimea de antrenare*.

Coloanele reprezintă *atributele*, care au *valori*. Instanțele sunt identificate de valorile atributelor. De exemplu, atributul *Umiditate* are două valori: *Normală* și *Mare*. Prima instanță este definită de valorile: { *Starea vremii* = *Soare*, *Temperatură* = *Mare*, *Umiditate* = *Mare*, *Vânt* = *Absent*, *Joc* = *Nu* }.

De obicei, ultimul atribut este *clasa*. În exemplul nostru, clasa este *Joc*.

Funcția generică de care discutam mai sus și care trebuie aproximată este aici: $h(\mathbf{x}) \approx Joc(Starea\ vremii,\ Temperatură,\ Umiditate,\ Vânt)$.

Pe baza acestor date simple, dorim să identificăm cunoștințe, adică modele mai generale și mai abstracte. Dacă la un moment dat știm condițiile meteorologice, aplicând modelul general putem determina dacă este bine să jucăm sau nu golf.

6.4. Tipuri de atrbute

Există patru tipuri de atrbute, organizate pe două coordonate:

- Atribute *discrete (simbolice)*: de tip *nominal* și *ordinal*;
- Atribute *continue (numerice)*: de tip *interval* și *rațional*.

Atribute discrete

Între valorile unui atribut nominal nu există o relație. Exemple de astfel de atribute sunt: culoarea ochilor (dacă nu o considerăm ca RGB), numele, sexul, CNP-ul ca obiect (nu ordonat ca număr), numeralele de pe tricourilor unor jucători de fotbal, o mulțime de țări etc.

Valorile atributelor ordinale sunt simbolice, însă între ele există relații de ordine, de exemplu înălțimea unei persoane discretizată în categoriile mică, medie și mare, în general, orice atribut ale căruia valori are sens să le considerăm în astfel de categorii (foarte mic, mic, mediu, mare, foarte mare etc.). Rangurile (primul, al doilea, al treilea), calificativele (satisfăcător, bine, foarte bine, excelent) sunt de asemenea atribute ordinale.

Există mici diferențe de tratare a atributelor nominale față de cele ordonale, mai ales la algoritmii bazați pe instanțe, prezentați în capitolul 9.

Atribute continue

Exemple de atribute de tip interval sunt: data calendaristică, temperatura în grade Celsius, nivelul de încredere într-o personalitate publică pe o scară de la 1 la 5 etc. Exemple de atribute raționale sunt: lungimea, distanța, greutatea, prețurile, temperatura în grade Kelvin etc.

Diferența dintre cele două tipuri este următoarea. „Rațional” vine de la „ratio”, care exprimă o fracție. În limba latină, „ratio” înseamnă atât judecată cât și calcul. Putem spune că o distanță de 2 km este de 2 ori mai mare decât o distanță de 1 km. Putem calcula un raport între aceste valori. La fel la prețuri: 10 lei este de 2 ori mai mult decât 5 lei. Pentru temperatura în grade Kelvin există 0 absolut. La atributele de tip interval, chiar dacă sunt continue, nu putem face acest tip de operație. De exemplu, o temperatură de 20 de grade Celsius nu este de 2 ori mai mare decât una de 10 grade și nu

are sens un raport de -2 între 20°C și -10°C. De asemenea, pentru datele calendaristice, nu putem face un raport între 1 martie 2010 și 1 februarie 2010.

Din punct de vedere al algoritmilor, nu prea există diferențe de prelucrare a atributelor de tip interval și a celor de tip rațional. Toate attributele continue pot fi tratate la fel. De exemplu, putem transforma o dată calendaristică într-un număr întreg care să specifică diferența în zile față de o anumită dată de referință.

Unii algoritmi sunt mai potriviti pentru un anumit tip de attribute. Pentru arborii de decizie (capitolul 7) și clasificarea bayesiană naivă (capitolul 8) am preferat să avem date discrete. Atributele continue pot fi discretizate, însă prin această transformare se pot pierde informații. Algoritmii bazați pe instanțe (capitolul 9), dimpotrivă, tratează în mod natural valori continue ale atributelor.

6.5. Estimarea capacitatei de generalizare

După realizarea modelului, este foarte importantă determinarea capacitatei sale de generalizare. De obicei, avem o singură mulțime de date. Pentru estimarea capacitatei de generalizare, împărțim datele existente într-o parte cu care să construim modelul (*mulțimea de antrenare*) și o parte pe care să îl verificăm (*mulțimea de validare sau de test*).

Există mai multe metode de a împărți datele în aceste două mulțimi.

Cea mai simplă este *împărțirea 2/3 – 1/3*. Două treimi din date se folosesc pentru antrenare iar restul de o treime pentru testarea modelului construit.

Cea mai folosită metodă este *validarea încrucișată* (engl. “cross-validation”), în care împărțim instanțele în n grupuri. De exemplu, dacă avem 100 de instanțe, le împărțim în 10 grupuri de câte 10. Construim modelul pe $n-1$ grupuri (de exemplu pe 90 de instanțe) și îl testăm pe grupul rămas. Apoi se schimbă grupul rămas pentru testare, se repetă procedura de n ori și se calculează rata medie de eroare.

Rata de eroare este raportul dintre numărul de instanțe clasificate greșit și numărul total de instanțe considerate pentru un test.

O altă metodă, utilă mai ales atunci când există un număr mic de date, este aşa numita *lasă una deoparte* (engl. “leave one out”). Dacă avem 10 instanțe, construim modelul cu 9 instanțe și îl verificăm cu a zecea, apoi schimbăm instanța rămasă pentru test și repetăm procesul de 10 de ori. Se calculează apoi rata de eroare medie a algoritmului pentru instanța de test, adică de câte ori este clasificată corect, în medie. În această situație, cu doar 10 instanțe, împărțirea 2/3 – 1/3 ar conduce la utilizarea a 7 instanțe pentru antrenare. În schimb, cu această metodă putem utiliza 9 instanțe.

Când folosim o mulțime de antrenare și una de test, rezultatele clasificării instanțelor de test sunt de obicei mai proaste decât dacă am folosi toate datele disponibile pentru antrenare. Trebuie să subliniem totuși faptul că scopul principal al clasificării nu este aproximarea perfectă a datelor de antrenare, care s-ar putea face și prin simpla memorare a acestora, ci crearea unui model care să se comporte bine pentru date noi.

Discutând despre capacitatea de generalizare, două noțiuni sunt foarte importante:

- *Sub-potrivirea* (engl. “underfitting”): ipoteza este prea simplă și nu poate învăța modelul din date;

- *Supra-potrivirea* (engl. “overfitting”): ipoteza este prea complexă și este influențată de zgomot și date irelevante.

Dacă un model este prea simplu, cum era exemplul de regresie cu funcția liniară, acesta nu poate să învețe datele. Oricum am orienta dreapta respectivă, ea nu poate trece prin toate punctele.

Dimpotrivă, un model suprapotrivit are performanțe foarte bune pe mulțimea de antrenare, dar performanțe slabe pe mulțimea de validare. Este cazul funcției polinomiale de gradul 6 din figura 6.4. De asemenea, datele de antrenare ar putea fi afectate de zgomot (pot apărea erori de măsurare, greșeli umane de transcriere, clasificări incorecte ale experților etc.). Mai ales în astfel de cazuri, un model mai „suplu” poate scădea influența datelor eronate, conducând la predicții mai bune.

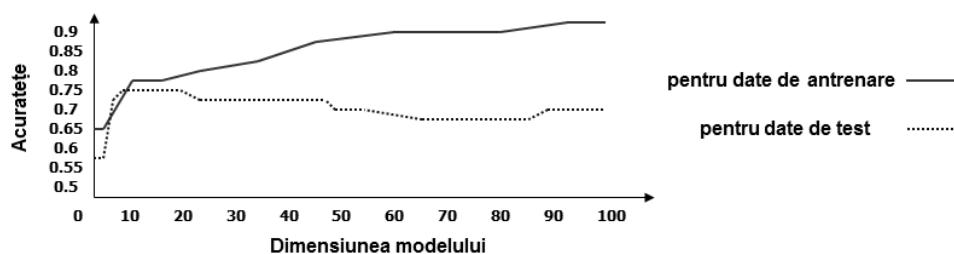


Figura 6.5. Evoluția acurateței pe măsura creșterii complexității modelului

Atunci când crește complexitatea modelului, la început de multe ori acuratețea (1 minus rata de eroare) pentru mulțimile de antrenare și de test crește constant, deoarece modelul reușește să potrivească datele din ce în ce mai bine. La un moment dat, există un *punct de maximă generalizare*, după care modelul începe să supra-potrivească, să fie mai complex decât ar

trebuie. Performanțele pentru datele de antrenare continuă să crească, însă performanțele pentru datele de test încep să scadă. În acest punct ar trebui să oprim dezvoltarea modelului, în care acuratețea pentru mulțimea de validare este maximă, chiar dacă am putea obține performanțe mai bune pe mulțimea de antrenare continuând procesul (figura 6.5).

6.6. Aplicații ale tehnicielor de clasificare

Există numeroase aplicații pentru algoritmii de clasificare:

- Învățarea tratamentelor optime din înregistrările medicale. Există multe înregistrări cu pacienți care au primit diferite tratamente pentru anumite simptome și se cunoaște dacă medicamentele respective au funcționat sau nu. Când trebuie tratat un nou pacient, cu anumite simptome și caracteristici (vârstă, sex, greutate, alte boli cunoscute etc.), se poate estima dacă o anumită schemă de tratament va funcționa sau nu;
- Clasificarea celulelor din tumorii ca benigne sau maligne pe baza radiografiilor;
- Predicția ratei de recuperare a pacienților cu pneumonie;
- Clasificarea structurilor secundare ale proteinelor. Conformația unei proteine este determinată de secvența sa de aminoacizi. În urma secvențierii ADN-ului uman, a apărut o cantitate uriașă de date liniare (șiruri de baze nucleice). În prezent, se depun eforturi ca din aceste șiruri să se extragă structura lor spațială (secundară), astfel încât pasul următor să fie predicția proprietăților pe baza componentelor structurale;

- Clasificarea plășilor electronice ca legitime sau frauduloase. Având în vedere cheltuielile tipice ale unei persoane, se poate determina un profil și astfel se poate verifica dacă o plată de la un anumit moment este similară cu cele anterioare;
- Recunoașterea vorbirii, echivalentă cu clasificarea secvențelor de sunete în cuvinte;
- Recunoașterea optică a caracterelor, ce presupune identificarea caracterelor dintr-o imagine;
- Clasificarea textelor. În acest caz, atributele sunt chiar cuvintele documentelor respective (după filtrarea cuvintelor uzuale – conjuncții, prepoziții, pronume – și eliminarea inflexiunilor) iar valorile atributelor sunt frecvențele de apariție ale acestora. Exemple sunt clasificarea știrilor în categorii precum politică, meteo, sport etc. sau clasificarea email-urilor în “spam” (mesaje nedorite) și “ham” (mesaje legitime).

Arborei de decizie

7.1. Algoritmul lui Hunt

Algoritmul lui Hunt (1962) este o procedură generică pentru construirea unui arbore de decizie. Inițial, toate instanțele din mulțimea de antrenare corespund rădăcinii arborelui. Ideea de bază este partaționarea fiecărui nod, adică împărțirea instanțelor în mai multe grupe, corespunzătoare fiilor nodului curent, astfel încât nodurile rezultate să fie cât mai omogene, adică toate instanțele din acel nod, sau majoritatea lor, să aparțină aceleiași clase. Într-o frunză, omogenitatea ar trebui să fie maximă, adică toate instanțele să aparțină aceleiași clase. Desigur, în anumite cazuri este imposibilă obținerea unor frunze perfect omogene, ceea ce conduce la apariția erorilor de clasificare. După partaționarea unui nod, rezultă mai multe noduri fiu, pe care le partaționăm în mod recursiv după același criteriu.

Mai formal, fie D_n mulțimea instanțelor de antrenare dintr-un nod n . Algoritmul lui Hunt are următorii pași (Tan, Steinbach & Kumar, 2006):

- Dacă D_n conține instanțe din aceeași clasă y_n , atunci n este o frunză etichetată y_n ;
- Dacă D_n este o mulțime vidă, atunci n este o frunză etichetată cu clasa implicită (engl. “default”) y_d ;

- Dacă D_n conține instanțe care aparțin mai multor clase, se utilizează un test de atribute pentru a parta datele în mulțimi mai mici;
- Se aplică recursiv procedura pentru fiecare submulțime.

Se poate observa că algoritmul lui Hunt recomandă o strategie greedy, deoarece se partionează mulțimea de instanțe cu un test care maximizează la un moment dat un anumit criteriu, cum ar fi omogenitatea nodurilor fiu rezultate.

Cheia este identificarea testului de atribut pe care se bazează partaționările. Algoritmul lui Hunt nu spune nimic despre natura acestuia. După cum vom vedea, există numeroase metode de inducție a arborilor de decizie, care diferă prin modalitatea de determinare a testul de atribut, pe lângă alte proceduri suplimentare de optimizare a rezultatelor.

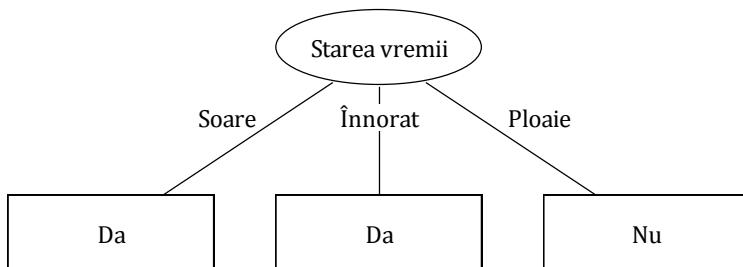
7.2. Specificarea testelor de atribut

Partaționarea datelor presupune în primul rând specificarea testului și apoi, în funcție de acesta, determinarea partaționării optime a unui nod. Partaționările sunt diferite în funcție de tipul atributelor.

Pentru un atribut nominal, putem partaționa nodul după fiecare valoare a atributului. Când avem mai multe valori discrete, creăm câte un fiu pentru fiecare valoare a atributului respectiv. Pentru a exemplifica, vom considera o variantă simplificată a problemei de clasificare introduse în capitolul 6, cu un singur atribut nominal și clasa.

Starea vremii	Joc
Soare	Da
Înnorat	Da
Ploaie	Nu

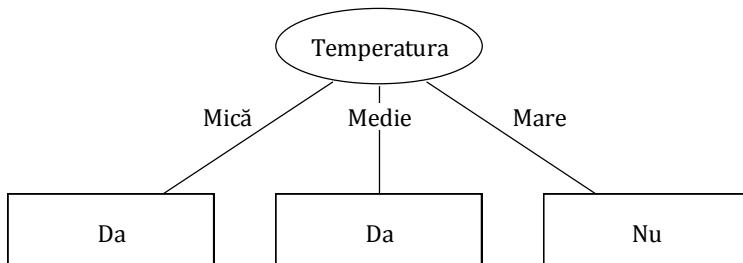
Atributul *Starea vremii* are 3 valori, deci vor rezulta 3 frunze corespunzătoare acestor valori.



Teoretic, am putea grupa valorile într-un număr mai mic de mulțimi, pentru a avea o frunză *Da* pentru valorile { *Soare*, *Înnorat* } și una *Nu* pentru { *Ploaie* } însă în cazul general aceasta este o problemă de căutare și optimizare.

Pentru atribute ordonale, procedura este similară: putem face partaționarea pentru fiecare valoare a atributului.

Temperatură	Joc
Mică	Da
Medie	Da
Mare	Nu



Relația de ordine ne-ar putea ajuta, pentru că am putea grupa valorile adiacente, de exemplu { *Mică*, *Medie* } și { *Mare* }. O partiționare după { *Mică*, *Mare* } și { *Medie* } este mai puțin intuitivă, dar nu incorectă, deoarece datele de antrenare ar putea fi de tipul:

Temperatură	Joc
<i>Mică</i>	Nu
<i>Medie</i>	Da
<i>Mare</i>	Nu

La atributele continue, într-o primă abordare, ar trebui discretizate valorile, deoarece numărul de fii ai unui nod este întreg, nu real.

O primă metodă de discretizare este sortarea valorilor, stabilirea unui număr de intervale egale (histograma) și introducerea valorilor în aceste intervale.

Umiditate	Joc
65	Da
70	Da
72	Da
75	Da
80	Da
85	Da
86	Nu
90	Nu
90	Nu
91	Nu
93	Nu
95	Nu

Să considerăm discretizarea în 3 intervale. Valorile numerice sunt distribuite în domeniul [65, 95], prin urmare vom considera următoarele intervale egale: [65, 75], (75, 85], respectiv (85, 90]. Atributul continuu va fi transformat într-unul ordinal, cu valorile *Mică*, *Medie* și *Mare*:

Umiditate-Dis1	Joc
Mică	Da
Medie	Da
Medie	Da
Mare	Nu

O altă metodă de discretizare este cea cu frecvențe egale, astfel încât fiecare interval să conțină un număr egal de valori. Nu mai contează mărimea valorilor, ci numărul lor.

Pentru exemplul de mai sus, cu 12 valori, primele 4 vor primi valoarea *Mică*, următoarele 4 – valoarea *Medie* și ultimele 4 – valoarea *Mare*.

Umiditate-Dis2	Joc
Mică	Da
Medie	Da
Medie	Da
Medie	Nu
Medie	Nu
Mare	Nu

O a treia modalitate este prin clusterizare (engl. “clustering”), care realizează o grupare mai naturală a valorilor. Clusterizarea aparține tipului

de învățare nesupervizată, deoarece acolo nu mai cunoaștem valoarea „clasei” după care se face gruparea. Se dau doar valorile propriu-zise ale celorlalte atribută și algoritmul trebuie să grupeze instanțele astfel încât în interiorul unui grup (cluster) asemănarea instanțelor componente să fie cât mai mare, iar între clustere diferite asemănarea instanțelor să fie cât mai mică. Acest tip de învățare nu face obiectul capitolului de față, însă un algoritm simplu de clusterizare care poate fi aplicat pentru discretizarea atributelor continue pentru o problemă de clasificare este *k-means* (MacQueen, 1967).

O variantă alternativă pentru tratarea atributelor continue este partaționarea binară, adică determinarea unei singure valori cu care să se compare valorile atributului considerat. În acest caz, ar trebui tratate toate partaționările posibile și prin urmare este necesar un efort de calcul mai mare. Pentru exemplul de mai sus, să presupunem că s-a determinat valoarea de referință 85, iar testul este specificat astfel: ($A_i \leq 85$)? Valorile atributului rezultat prin această transformare vor fi *Da* sau *Nu*:

Umiditate	Umiditate-Bin	Joc
65	Da	Da
70	Da	Da
72	Da	Da
75	Da	Da
80	Da	Da
85	Da	Da
86	Nu	Nu
90	Nu	Nu
90	Nu	Nu
91	Nu	Nu
93	Nu	Nu
95	Nu	Nu

7.3. Măsuri de omogenitate

Pentru a face o partiționare la un moment dat, avem nevoie de o măsură a omogenității. O astfel de măsură a impurității unui nod, este *entropia*, utilizată de algoritmii *ID3* (Iterative Dichotomiser 3, Quinlan, 1983) și *C4.5* (Quinlan, 1993) pentru a descoperi arbori de dimensiuni cât mai reduse.

În *teoria informației* (Shannon, 1948), entropia măsoară incertitudinea asociată cu o variabilă aleatorie, care se reflectă în cantitatea de informație conținută de un mesaj. Cu cât este mai mare cantitatea de informație, cu atât entropia este mai mare.

Entropia este definită astfel:

$$H(X) = - \sum_{i=1}^n P(x_i) \cdot \log_b P(x_i), \quad (7.1)$$

unde X este o variabilă aleatorie discretă cu valorile posibile $\{x_1, \dots, x_n\}$, $P(x_i)$ este probabilitatea de apariție a valorii x_i iar b este baza logaritmului. Dacă $b = 2$, atunci unitatea de măsură a entropiei este *bitul*.

Prin convenție, când $P(x_i) = 0$, se consideră că întreg termenul $P(x_i) \cdot \log_b P(x_i)$ este 0, deoarece: $\lim_{p \rightarrow 0^+} p \log p = 0$.

În cazul problemei de clasificare, probabilitățile $P(x_i)$ sunt estimate ca frecvențe relative de apariție a valorilor x_i în mulțimea de antrenare.

Pentru 2 clase, graficul entropiei este cel din figura 7.1. Valoarea maximă este 1 când într-un nod ambele clase au un număr egal de instanțe (ambele posibilități sunt egale probabile și au probabilitatea 1/2). Valoarea

minimă este 0, când toate instanțele aparțin unei singure clase: dacă aparțin primei clase, atunci $P(x_1) = 1$ și $P(x_2) = 0$, iar dacă aparțin celei de a doua clase, atunci $P(x_1) = 0$ și $P(x_2) = 1$.

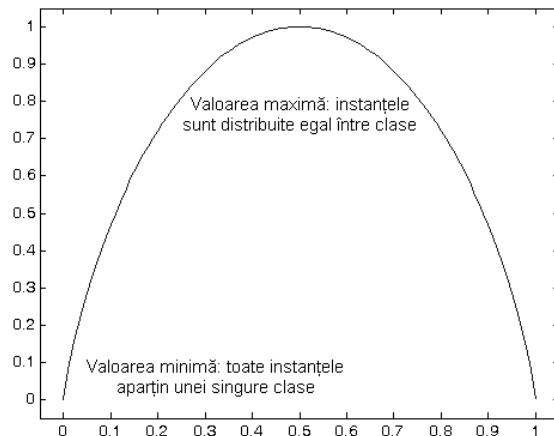


Figura 7.1. Graficul entropiei pentru 2 clase

Pentru 3 clase, graficul entropiei este prezentat în figura 7.2 (Lawrence, 1997).

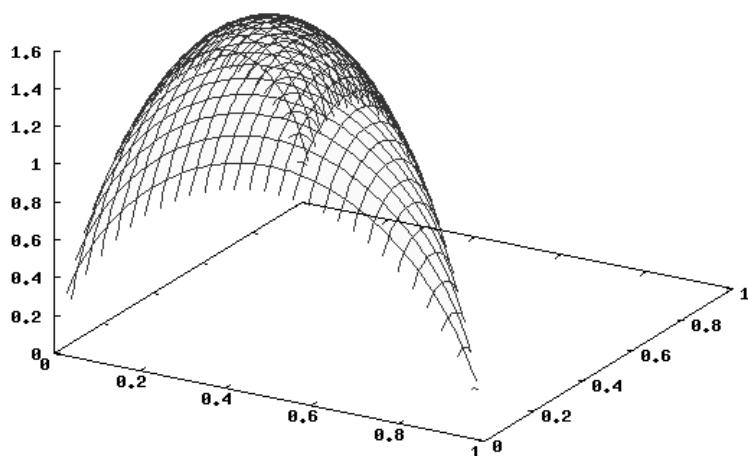


Figura 7.2. Graficul entropiei pentru 3 clase

În acest caz, valoarea maximă este atunci când toate cele 3 clase au probabilități egale, de $1/3$, și în acest caz valoarea maximă este $\log_2(3) = 1,58$. Valoarea minimă este de asemenea 0, când o singură clasă are probabilitatea 1 iar celelalte au probabilitatea 0.

Pentru clasificare, am dori ca toate instanțele dintr-un nod să aparțină aceleiași clase, ceea ce corespunde minimului entropiei. Dacă entropia este mare, atunci avem un număr relativ egal de instanțe în fiecare clasă (omogenitate mică), ceea ce ne îndepărtează de scopul clasificării.

Alternativ, în locul entropiei, se poate folosi, ca în algoritmul *CART* (Breiman et al., 1984), *indexul Gini*, definit astfel:

$$G(X) = 1 - \sum_{i=1}^n (P(x_i))^2. \quad (7.2)$$

Utilizând indexul Gini, efortul de calcul este mai mic deoarece se evită calcularea logaritmilor. Pentru 2 clase, forma indexului Gini este asemănătoare funcției de entropie, cu o valoare maximă de 0,5, după cum se poate vedea în figura 7.3.

Rezultatele obținute folosind drept criteriu de omogenitate entropia sau indexul Gini sunt de cele mai multe ori identice, mai ales când numărul de clase este mic. Diferențele apar când avem un număr mare de clase și realizăm partiționări binare. Entropia accentuează o împărțire echilibrată astfel încât cele două noduri fiu să aibă dimensiuni apropriate, în timp ce indexul Gini favorizează partiționările care pun instanțele din clasa cea mai mare într-un singur nod pur și instanțele din toate celelalte clase în celălalt nod fiu (Breiman, 1996).

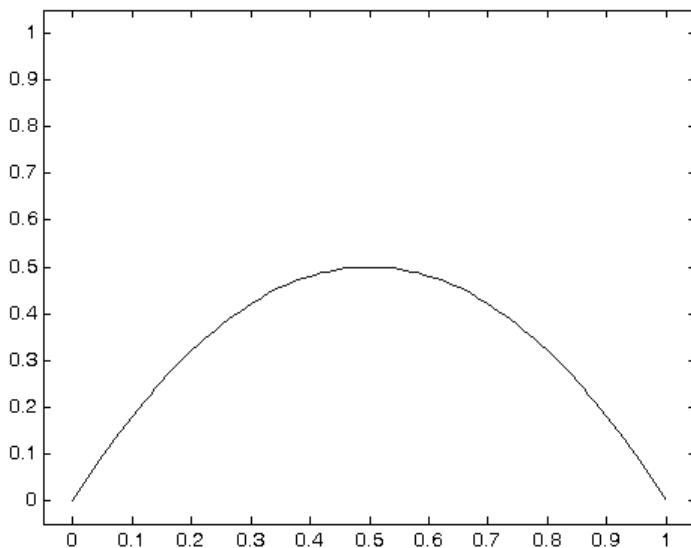


Figura 7.3. Graficul indexului Gini pentru 2 clase

7.4. Partiționarea

După cum am spus, din nodul părinte, prin partiționare, trebuie să rezulte noduri cât mai omogene. Procedura prin care aplicăm criteriul de omogenitate precum entropia pentru selectarea unui atribut după care se va face partiționarea este prezentată în continuare.

Când un nod părinte p este partiționat în k fiu, calitatea partiționării se calculează ca o medie ponderată a entropiilor nodurilor fiu rezultate:

$$H_s = \sum_{i=1}^k \frac{n_i}{n} \cdot H_i, \quad (7.3)$$

unde n_i este numărul de instanțe din nodul fiu i , n este numărul de instanțe din nodul părinte p , iar s este o partiționare (engl. “split”) din mulțimea tuturor partiționărilor posibile.

Mai întâi se calculează entropiile tuturor nodurilor fiu rezultate H_i și apoi se ponderează acestea cu numărul de instanțe din fiecare nod fiu.

Creșterea omogenității submulțimilor rezultate este echivalentă cu maximizarea *câștigului informațional* (engl. “information gain”):

$$\Delta_s = H_p - H_s = H_p - \sum_{i=1}^k \frac{n_i}{n} \cdot H_i. \quad (7.4)$$

Deoarece entropia nodului părinte H_p este aceeași pentru toate partiționările, se preferă valoarea minimă pentru sumă, astfel încât partiționarea dorită este:

$$s^* = \operatorname{argmax}_s \Delta_s = \operatorname{argmin}_s H_s = \operatorname{argmin}_s \sum_{i=1}^{k_s} \frac{n_i}{n} \cdot H_i. \quad (7.5)$$

Numărul de noduri fiu k_s depinde de tipul și de numărul de valori ale atributului după care se face partiționarea s , după cum am spus în secțiunea 7.2.

7.5. Probleme cu attribute simbolice

Vom considera ca exemplu problema prezentată în tabelul 7.1, cu 14 instanțe și definită de 4 attribute simbolice și clasa. Pentru a construi arborele

de decizie după algoritmul *ID3*, detaliat mai sus, mai întâi trebuie să partaționăm întreaga mulțime de antrenare, pe rând, după fiecare atribut, și să determinăm cea mai bună partaționare. Coloana *Nr. instanță* nu face parte din problemă și a fost inclusă doar pentru a urmări mai ușor prelucrările efectuate.

Tabelul 7.1. Problemă de clasificare cu atribute simbolice

Nr. instanță	Starea vremii	Temperatură	Umiditate	Vânt	Joc
1	Soare	Mare	Mare	Absent	Nu
2	Soare	Mare	Mare	Prezent	Nu
3	Înnorat	Mare	Mare	Absent	Da
4	Ploaie	Medie	Mare	Absent	Da
5	Ploaie	Mică	Normală	Absent	Da
6	Ploaie	Mică	Normală	Prezent	Nu
7	Înnorat	Mică	Normală	Prezent	Da
8	Soare	Medie	Mare	Absent	Nu
9	Soare	Mică	Normală	Absent	Da
10	Ploaie	Medie	Normală	Absent	Da
11	Soare	Medie	Normală	Prezent	Da
12	Înnorat	Medie	Mare	Prezent	Da
13	Înnorat	Mare	Normală	Absent	Da
14	Ploaie	Medie	Mare	Prezent	Nu

Să considerăm mai întâi atributul *Starea vremii* (*S*), care are 3 valori: *Soare*, *Înnorat* și *Ploaie*. Dacă se partaționează nodul rădăcină după acest atribut, vor rezulta 3 noduri fiu, câte unul pentru fiecare valoare. Trebuie să determinăm entropia acestor noduri potențiale.

Pentru valoarea *Soare* (*S*), nodul rezultat va avea 2 instanțe din clasa *Da* și 3 din clasa *Nu*.

Nr. instanță	Starea vremii	Joc
1	Soare	Nu
2	Soare	Nu
8	Soare	Nu
9	Soare	Da
11	Soare	Da

Prin urmare, entropia sa va fi:

$$H_{S_S} = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0,971.$$

Pentru valoarea *Înnorat* (*I*), nodul rezultat va avea toate cele 4 instanțe din clasa *Da*. Este evident că: $H_{S_I} = 0$.

Nr. instanță	Starea vremii	Joc
3	Înnorat	Da
7	Înnorat	Da
12	Înnorat	Da
13	Înnorat	Da

Pentru valoarea *Ploaie* (*P*), nodul rezultat va avea 3 instanțe din clasa *Da* și 2 din clasa *Nu*.

Nr. instanță	Starea vremii	Joc
4	Ploaie	Da
5	Ploaie	Da
6	Ploaie	Nu
10	Ploaie	Da
14	Ploaie	Nu

Entropia sa va fi:

$$H_{S_P} = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0,971.$$

Putem calcula acum entropia medie a partiționării după atributul *Starea vremii*:

$$H_S = \frac{5}{14} \cdot 0,971 + \frac{4}{14} \cdot 0 + \frac{5}{14} \cdot 0,971 = 0,694.$$

Același tip de calcule se realizează pentru următorul atribut, *Temperatură* (T), cu valorile: *Mică* (L), *Medie* (M) și *Mare* (H). Pentru a calcula mai ușor, sortăm tabelul după coloana *Temperatură* și numărăm valorile clasei pentru fiecare valoare a atributului.

Nr. instantă	Temperatură	Joc
5	Mică	Da
6	Mică	Nu
7	Mică	Da
9	Mică	Da
4	Medie	Da
8	Medie	Nu
10	Medie	Da
11	Medie	Da
12	Medie	Da
14	Medie	Nu
1	Mare	Nu
2	Mare	Nu
3	Mare	Da
13	Mare	Da

Se poate vedea ușor că:

$$H_{T_L} = -\frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{4} = 0,811$$

$$H_{T_M} = -\frac{4}{6} \log_2 \frac{4}{6} - \frac{2}{6} \log_2 \frac{2}{6} = 0,918$$

$$H_{T_H} = -\frac{2}{4} \log_2 \frac{2}{4} - \frac{2}{4} \log_2 \frac{2}{4} = 1.$$

Prin urmare:

$$H_T = \frac{4}{14} \cdot 0,811 + \frac{6}{14} \cdot 0,918 + \frac{4}{14} \cdot 1 = 0,911.$$

La fel procedăm pentru atributul *Umiditate* (*U*), cu valorile: *Normală* (*N*) și *Mare* (*M*).

Nr. instanță	Umiditate	Joc
5	Normală	Da
6	Normală	Nu
7	Normală	Da
9	Normală	Da
10	Normală	Da
11	Normală	Da
13	Normală	Da
1	Mare	Nu
2	Mare	Nu
3	Mare	Da
4	Mare	Da
8	Mare	Nu
12	Mare	Da
14	Mare	Nu

Vom avea:

$$H_{U_N} = -\frac{6}{7} \log_2 \frac{6}{7} - \frac{1}{7} \log_2 \frac{1}{7} = 0,592$$

$$H_{T_M} = -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7} = 0,985$$

$$\Rightarrow H_U = \frac{7}{14} \cdot 0,592 + \frac{7}{14} \cdot 0,985 = 0,789.$$

În final, pentru atributul *Vânt* (*V*) cu valorile *Absent* (*A*) și *Prezent* (*P*):

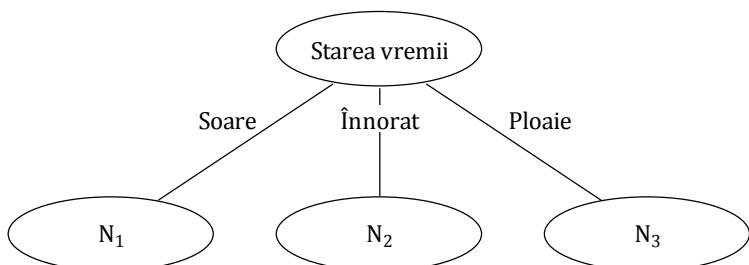
Nr. instanță	Vânt	Joc
1	Absent	Nu
3	Absent	Da
4	Absent	Da
5	Absent	Da
8	Absent	Nu
9	Absent	Da
10	Absent	Da
13	Absent	Da
2	Prezent	Nu
6	Prezent	Nu
7	Prezent	Da
11	Prezent	Da
12	Prezent	Da
14	Prezent	Nu

$$H_{V_A} = -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \frac{2}{8} = 0,811$$

$$H_{V_P} = -\frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6} = 1$$

$$\Rightarrow H_V = \frac{8}{14} \cdot 0,811 + \frac{6}{14} \cdot 1 = 0,892.$$

Valoarea maximă a câștigului informațional este corespunzătoare minimului entropiei ponderate $H_S = 0,694$ și deci prima partiționare se va face după atributul *Starea vremii*.



După ce am făcut o partitōnare, eliminăm atributul respectiv din mulțimea de date și eliminăm și instanțele care au valoarea atributului considerat diferită de valoarea de pe ramura nodului fiu corespunzător.

Pentru nodul N_1 se repetă procedura, eliminând atributul *Starea vremii* și păstrând doar instanțele care au ca valoare a acestuia *Soarele* (5 instanțe).

Nr. instanță	Starea vremii	Temperatură	Umiditate	Vânt	Joc
1	Soare	Mare	Mare	Absent	Nu
2	Soare	Mare	Mare	Prezent	Nu
3	Înnorat	Mare	Mare	Absent	Da
4	Ploaie	Medie	Mare	Absent	Da
5	Ploaie	Mică	Normală	Absent	Da
6	Ploaie	Mică	Normală	Prezent	Nu
7	Înnorat	Mică	Normală	Prezent	Da
8	Soare	Medie	Mare	Absent	Nu
9	Soare	Mică	Normală	Absent	Da
10	Ploaie	Medie	Normală	Absent	Da
11	Soare	Medie	Normală	Prezent	Da
12	Înnorat	Medie	Mare	Prezent	Da
13	Înnorat	Mare	Normală	Absent	Da
14	Ploaie	Medie	Mare	Prezent	Nu

Pentru atrbutele rămase, calculăm din nou entropile.

Pentru *Temperatură*:

$$H_{T_L} = 0 \text{ (1 instanță în clasa } Da\text{)}$$

$$H_{T_M} = 1 \text{ (1 instanță în clasa } Da \text{ și 1 instanță în clasa } Nu\text{)}$$

$$H_{T_H} = 0 \text{ (2 instanțe în clasa } Nu\text{)}$$

$$\Rightarrow H_T = \frac{1}{5} \cdot 0 + \frac{2}{5} \cdot 1 + \frac{2}{5} \cdot 0 = 0,4.$$

Pentru *Umiditate*:

$$H_{U_N} = 0$$

$$H_{U_M} = 0$$

$$\Rightarrow H_U = 0.$$

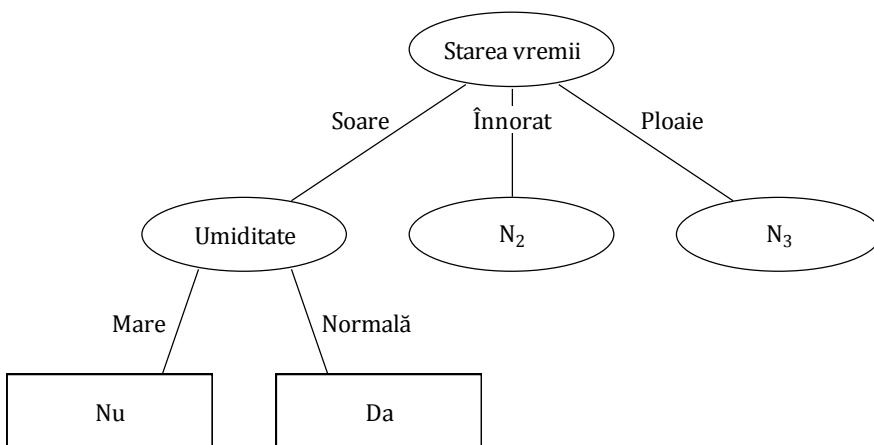
Pentru *Vânt*:

$$H_{V_A} = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} = 0,918$$

$$H_{V_P} = 1$$

$$\Rightarrow H_V = \frac{3}{5} \cdot 0,918 + \frac{2}{5} \cdot 1 = 0,951.$$

$H_U = 0$ este valoarea minimă, prin urmare nodul N_1 va fi partiționat după *Umiditate*. Observăm că nodurile fiu rezultate sunt frunze omogene și deci nu mai este nevoie de o altă partiționare.



În nodul N_2 avem următoarea mulțime de date:

Nr. instanță	Starea vremii	Temperatură	Umiditate	Vânt	Joc
3	Înnorat	Mare	Mare	Absent	Da
7	Înnorat	Mică	Normală	Prezent	Da
12	Înnorat	Medie	Mare	Prezent	Da
13	Înnorat	Mare	Normală	Absent	Da

Nodul este omogen și deci va fi la rândul său frunză, fără a mai trebui partiționat.

În sfârșit, în nodul N_3 avem următoarea mulțime de date:

Nr. instanță	Starea vremii	Temperatură	Umiditate	Vânt	Joc
4	Ploaie	Medie	Mare	Absent	Da
5	Ploaie	Mică	Normală	Absent	Da
6	Ploaie	Mică	Normală	Prezent	Nu
10	Ploaie	Medie	Normală	Absent	Da
14	Ploaie	Medie	Mare	Prezent	Nu

Aplicând aceeași procedură, vom obține: $H_T = 0,951$, $H_U = 0,951$ și $H_V = 0$. Ultima valoare este minimă și deci vom partiționa nodul N_3 după atributul $Vânt$, rezultând de asemenea două frunze.

Arborele final va fi cel din figura 7.4.

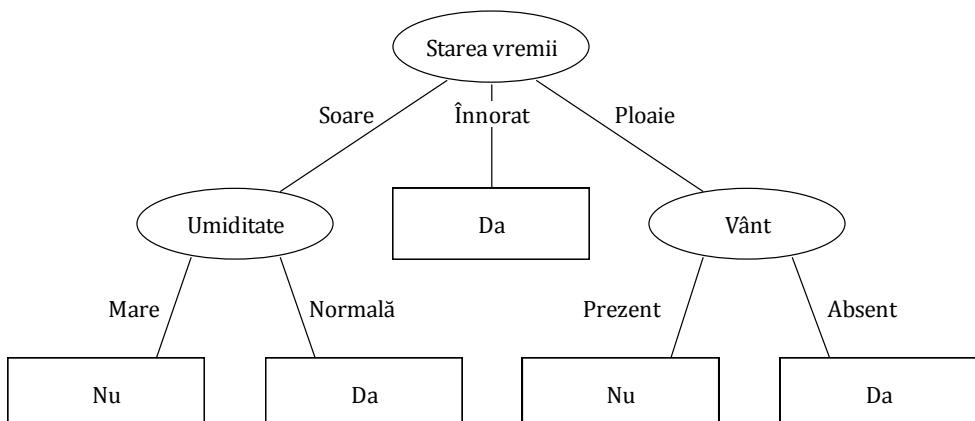


Figura 7.4. Arborele de decizie pentru problema cu atributuri simbolice

Temperatura este un atribut irelevant pentru această clasificare.

Se observă că pe măsură ce mulțimile se partionează, calculele devin din ce în ce mai simple. Din punct de vedere al implementării, cele mai dificile aspecte sunt crearea submulțimilor corespunzătoare nodurilor din arbore și aplicarea recursivă a procedurii pentru acestea.

7.6. Probleme cu atribute numerice

Multe probleme de clasificare conțin date numerice, precum cea din tabelul 7.2, o variantă a celei studiate în secțiunea 7.5, unde *Temperatura* este dată în grade Celsius iar *Umiditatea* în procente.

Tabelul 7.2. Problemă de clasificare cu atribute mixte

Nr. instanță	Starea vremii	Temperatură	Umiditate	Vânt	Joc
1	Soare	29,4	85	Absent	Nu
2	Soare	26,7	90	Prezent	Nu
3	Înnorat	28,3	86	Absent	Da
4	Ploaie	21,1	96	Absent	Da
5	Ploaie	20,0	80	Absent	Da
6	Ploaie	18,3	70	Prezent	Nu
7	Înnorat	17,8	65	Prezent	Da
8	Soare	22,2	95	Absent	Nu
9	Soare	20,6	70	Absent	Da
10	Ploaie	23,9	80	Absent	Da
11	Soare	23,9	70	Prezent	Da
12	Înnorat	22,2	90	Prezent	Da
13	Înnorat	27,2	75	Absent	Da
14	Ploaie	21,7	91	Prezent	Nu

Pe lângă procedeele de discretizare, datele continue pot fi tratate ca atare. Metoda prezentată în cele ce urmează este cea adoptată de algoritmul C4.5.

Procedura generală de partiționare este aceeași, pe baza câștigului informațional. Atributele simbolice sunt prelucrate la fel.

Pentru attributele numerice se dorește o partiționare binară, adică se caută o valoare astfel încât instanțele cu valoarea mai mică decât referința să aparțină unui fiu iar cele cu valoarea mai mare decât referința să aparțină celuilalt fiu. Problema se reduce la determinarea valorii de referință. Pentru aceasta, într-o primă variantă, se sortează valorile atributului numeric și se încearcă *toate* referințele potențiale dintre fiecare două valori alăturate.

Vom aplica această modalitate de calcul pentru atributul *Temperatură*.

Pozиїile de partiționare testate și entropia corespunzătoare partiționării sunt prezentate în tabelul de mai jos.

Pentru prima poziție de partiționare și pentru ultima, se vede că toate instanțele intră într-un singur nod fiu, 9 dintre ele cu clasa *Da* și 5 cu clasa *Nu*. Nicio instanță nu are temperatură mai mică decât 17,8 și toate instanțele au temperatură mai mare sau egală cu 17,8. La fel, toate instanțele au temperatură mai mică sau egală cu 29,4 și nicio instanță nu are temperatură mai mare decât 29,4. Este firesc, deoarece toate instanțele au temperatură mai mare sau egală cu prima valoare și mai mică sau egală cu ultima valoare, în sirul sortat.

Temperatură	Joc	Poziții de partităionare	Număr Joc = Da	Număr Joc = Nu	Entropiile submulțimilor	Entropia partităionării
		< 17,8 ≥ 17,8	0 9	0 5	0,000 0,940	0,940
17,8	Da					
		≤ 18,05 > 18,05	1 8	0 5	0,000 0,961	0,893
18,3	Nu					
		≤ 19,15 > 19,15	1 8	1 4	1,000 0,918	0,930
20,0	Da					
		≤ 20,3 > 20,3	2 7	1 4	0,918 0,946	0,940
20,6	Da					
		≤ 20,85 > 20,85	3 6	1 4	0,811 0,971	0,925
21,1	Da					
		≤ 21,4 > 21,4	4 5	1 4	0,722 0,991	0,895
21,7	Nu					
		≤ 21,95 > 21,95	4 5	2 3	0,918 0,954	0,939
22,2	Nu					
		≤ 22,2 > 22,2	5 4	3 2	0,985 0,863	0,924
22,2	Da					
		≤ 23,05 > 23,05	5 4	3 2	0,954 0,918	0,939
23,9	Da					
		≤ 23,9 > 23,9	6 3	3 2	0,918 0,971	0,937
23,9	Da					
		≤ 25,3 > 25,3	7 2	3 2	0,881 1,000	0,915
26,7	Nu					
		≤ 26,95 > 26,95	7 2	4 1	0,946 0,918	0,940
27,2	Da					
		≤ 27,75 > 27,75	8 1	4 1	0,918 1,000	0,930
28,3	Da					
		≤ 28,85 > 28,85	9 0	4 1	0,890 0,000	0,827
29,4	Nu	≤ 29,4 > 29,4	9 0	5 0	0,940 0,000	0,940

Ca exemplu de calcul, să considerăm partitōnarea după referința $28,85 = (28,3 + 29,4) / 2$. 13 instanțe au temperatură mai mică sau egală cu 28,85, din care 9 aparțin clasei *Da* și 4 clasei *Nu*. 1 instanță are temperatură mai mare decât 28,85, aparținând clasei *Nu*. Pentru primul nod fiu, entropia este:

$$H_{T \leq 28,85} = -\frac{9}{13} \log_2 \frac{9}{13} - \frac{4}{13} \log_2 \frac{4}{13} = 0,89.$$

Pentru al doilea nod fiu, entropia va fi 0, fiind omogen: $H_{T > 28,85} = 0$.

Prin urmare, entropia medie a acestei partitōnări va fi:

$$H_{T_{28,85}} = \frac{13}{14} \cdot 0,89 + \frac{1}{14} \cdot 0 = 0,827.$$

Se observă că această valoarea este minimă și va corespunde atributului *Temperatură*.

Această abordare, de a considera toate valorile intermediare, nu este însă optimă. Pentru un număr mare de instanțe, efortul de calcul crește foarte mult întrucăt numărul de partitōnări potențiale este foarte mare.

Putem observa însă în tabelul următor că după entropia inițială de 0,94, entropia scade până la valoarea 0,893, deoarece a fost introdusă într-un nod separat o instanță *Da*. Apoi entropia crește la 0,93, pentru că în acel nod a intrat și o instanță *Nu*, scăzându-i omogenitatea. În continuare, pe măsură ce alte 3 instanțe *Da* intră pe rând în nod, entropia tot scade, până la 0,895. În acest moment intră o instanță *Nu*, iar entropia crește din nou. La fel, de jos în sus, entropia scade până la prima schimbare de clasă.

		0,940
17,8	Da	↙
		0,893
18,3	Nu	↗
		0,930
20,0	Da	↗
		0,940
20,6	Da	↙
		0,925
21,1	Da	↙
		0,895
21,7	Nu	↗
		0,939
22,2	Nu	↙
		0,924
22,2	Da	↗
		0,939
23,9	Da	↙
		0,937
23,9	Da	↙
		0,915
26,7	Nu	↗
		0,940
27,2	Da	↙
		0,930
28,3	Da	↙
		0,827
		↗
29,4	Nu	0,940

Prin urmare, nu are rost să calculăm entropia decât în momentul în care se schimbă clasa între două instanțe alăturate în sirul sortat. Până când se schimbă clasa, entropia continuă să scadă deoarece intră într-un nod fiu mai multe instanțe din aceeași clasă.

Pentru atributul *Temperatură*, ar fi fost necesare doar 9 partiționări în loc de 15. În funcție de natura datelor, reducerea numărului poate fi mult mai drastică.

Vom utiliza această optimizare pentru calculul entropiei atributului *Umiditate*, după cum se poate observa în tabelul următor.

Umiditate	Joc	Poziții de partităionare	Număr Joc = Da	Număr Joc = Nu	Entropiile submulțimilor	Entropia partităionării
		< 65 >= 65	0 9	0 5	0,000 0,940	0,940
65	Da					
		≤ 67,5 > 67,5	1 8	0 5	0,000 0,961	0,893
70	Da					
		≤ 70 > 70				-
70	Da					
		≤ 70 > 70	3 6	0 5	0,000 0,994	0,781
70	Nu					
		≤ 72,5 > 72,5	3 6	1 4	0,811 0,971	0,925
75	Da					
		≤ 77,5 > 77,5				-
80	Da					
		≤ 80 > 80				-
80	Da					
		≤ 82,5 > 82,5	6 3	1 4	0,592 0,985	0,788
85	Nu					
		≤ 85,5 > 85,5	6 3	2 3	0,811 1,000	0,892
86	Da					
		≤ 88 > 88				-
90	Da					
		≤ 90 > 90	8 1	2 3	0,722 0,811	0,747
90	Nu					
		≤ 90,5 > 90,5				-
91	Nu					
		≤ 93 > 93				-
95	Nu					
		≤ 95,5 > 95,5	8 1	5 0	0,961 0,000	0,893
96	Da	≤ 96 > 96	9 0	5 0	0,940 0,000	0,940

Valoarea minimă este: $H_{U_{90}} = 0,747$. Pentru atributele simbolice *Starea vremii* și *Vânt*, valorile sunt la fel ca acelea calculate în secțiunea 7.5: $H_S = 0,694$ și $H_V = 0,892$.

Rezultatele pentru atrbute simbolice și numerice sunt tratate la fel, comparând entropiile medii ponderate obținute pentru toate atrbutele și alegând minimul. Prima partiționare va fi tot după *Starea vremii*.

Nodul corespunzător valorii *Innorat* este omogen, după cum se vede în tabelul de mai jos.

Starea vremii	Temperatură	Umiditate	Vânt	Joc
Înnorat	28,3	86	Absent	Da
Înnorat	17,8	65	Prezent	Da
Înnorat	22,2	90	Prezent	Da
Înnorat	27,2	75	Absent	Da

Submulțimea de instanțe din nodul valorii *Ploaie* este cea din tabelul următor.

Starea vremii	Temperatură	Umiditate	Vânt	Joc
Ploaie	21,1	96	Absent	Da
Ploaie	20,0	80	Absent	Da
Ploaie	23,9	80	Absent	Da
Ploaie	18,3	70	Prezent	Nu
Ploaie	21,7	91	Prezent	Nu

Pentru a evita calculele destul de laborioase, putem să analizăm tabelul și să observăm că dacă sortăm valorile *Temperaturii* sau *Umidității*, valorile clasei *Da* și *Nu* vor fi intercalate. Pentru atrbutul *Vânt*, este clar că printr-o singură partiționare vor rezulta două frunze omogene. Prin urmare, vom partiționa acest nod după atrbutul *Vânt*.

În cazul nodului corespunzător valorii *Soare*, submulțimea de instanțe de antrenare este prezentată în tabelul următor.

Starea vremii	Temperatură	Umiditate	Vânt	Joc
Soare	20,6	70	Absent	Da
Soare	23,9	70	Prezent	Da
Soare	29,4	85	Absent	Nu
Soare	26,7	90	Prezent	Nu
Soare	22,2	95	Absent	Nu

Atributul *Vânt* va avea o entropie medie de 0,951, ca și în cazul simbolic.

Aplicăm procedura partiționărilor multiple după valori intermediare pentru atributul *Temperatură*, după cum se observă în tabelul de mai jos.

Temperatură	Joc	Pozitii de partiționare	Număr Joc = Da	Număr Joc = Nu	Entropiile submulțimilor	Entropia partiționării
		< 20,6 >= 20,6	0 2	0 3	0,000 0,971	0,971
20,6	Da					
		≤ 21,4 > 21,4	1 1	0 3	0,000 0,811	0,649
22,2	Nu					
		≤ 23,05 > 23,05	1 1	1 2	1,000 0,918	0,951
23,9	Da					
		≤ 25,3 > 25,3	2 0	1 2	0,918 0,000	0,551
26,7	Nu					
		≤ 28,05 > 28,05	2 0	2 1	1,000 0,000	0,800
29,4	Nu	≤ 29,4 > 29,4	2 0	3 0	0,971 0,000	0,971

Valoarea minimă este 0,551 pentru referința 25,3.

Pentru *Umiditate*, jumătate din partiționări sunt evitate considerând doar referințele unde se schimbă clasa și rezultă o valoare minimă 0 pentru referința 77,5. Este clar că vom partiționa după *Umiditate*, rezultând și aici două frunze omogene.

Umiditate	Joc	Poziții de partităionare	Număr Joc = Da	Număr Joc = Nu	Entropiile submulțimilor	Entropia partităionării
		< 70 >= 70	0 2	0 3	0,000 0,971	0,971
70	Da					
		≤ 70 > 70				-
70	Da					
		$\leq 77,5$ $> 77,5$	2 0	0 3	0,000 0,000	0,000
85	Nu					
		$\leq 87,5$ $> 87,5$				-
90	Nu					
		$\leq 92,5$ $> 92,5$				-
95	Nu	≤ 95 > 95	2 0	3 0	0,971 0,000	0,971

Arborele final rezultat este cel prezentat în figura 7.5.

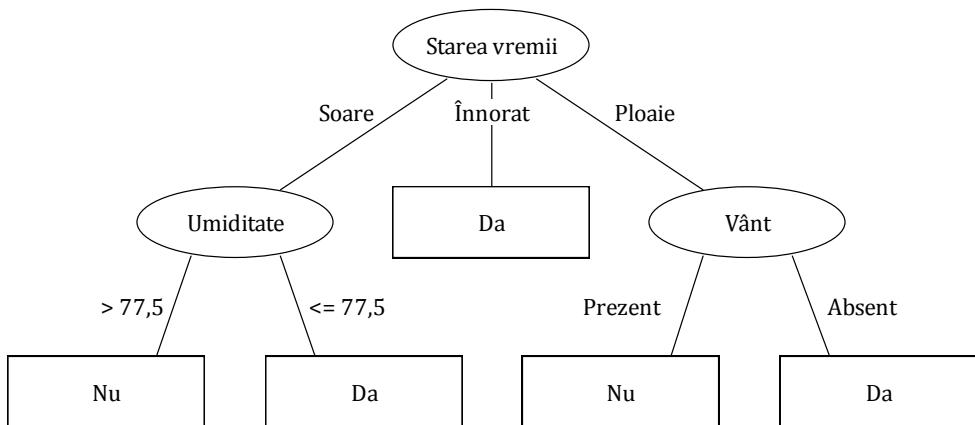


Figura 7.5. Arborele de decizie pentru problema cu atrbute mixte

7.7. Aplicarea modelului

Cunoscându-se arborele de decizie construit, clasificarea unei noi instanțe este foarte simplă. Parcurcând arborele cu valorile atributelor instanței, se clasifică aceasta într-o din clase.

Se testează mai întâi atributul corespunzător rădăcinii arborelui. Conform valorii atributului respectiv al instanței de interogare, se merge pe ramura aferentă, se testează apoi atributul corespunzător nodului în care s-a ajuns și aşa mai departe până se ajunge într-un nod frunză care are clasa specificată.

Să considerăm instanța de interogare $x_q = (\text{Soare}, 26, 72, \text{Absent})$. Clasificarea acesteia se realizează urmărind arborele din figura 7.5. Rădăcina arborelui presupune un test al atributului *Starea vremii*. Pentru x_q , valoarea acestuia este *Soare*, prin urmare continuăm parcurgerea arborelui pe ramura corespunzătoare acestei valori. Urmează testul după atributul *Umiditate*. Valoarea acestuia pentru x_q este 72, care este mai mică sau egală cu 77,5. Urmând ramura corespunzătoare, ajungem în frunza etichetată *Da*. În consecință, instanța x_q este clasificată în clasa *Da*.

7.8. Erorile modelului

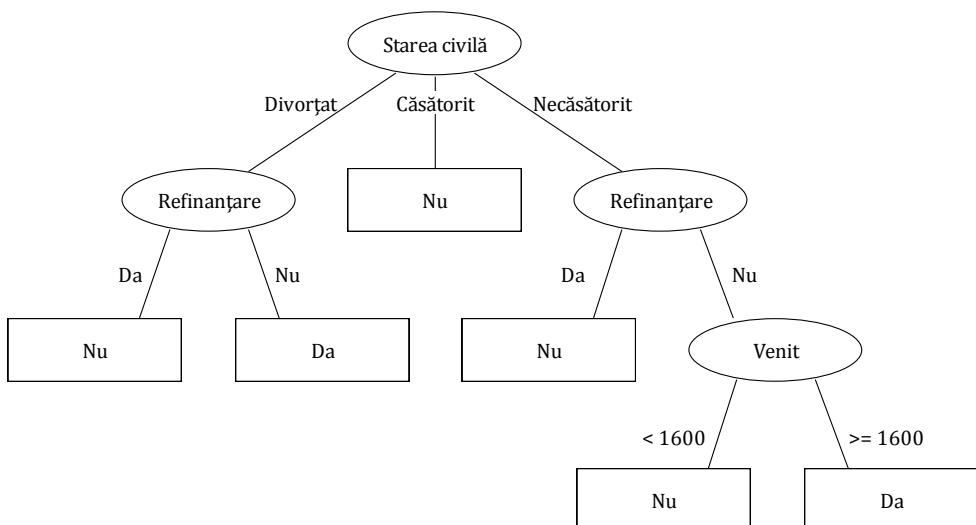
Aplicând metodele prezентate în acest capitol, pot exista mai multe atrbute cu același valoare minimă a entropiei medii iar partitioarea se poate face după oricare din ele, însă arborii de decizie finali rezultați pot fi complet diferenți în funcție de această decizie.

De asemenea, pot exista mulțimi de date pentru care un model să fie greu de realizat și în consecință să existe erori chiar și la antrenare. De

exemplu, să considerăm o problemă de clasificare (adaptată după Tan, Steinbach & Kumar, 2006) unde dorim să prezicem dacă o persoană va returna un credit de la o bancă, pe baza următoarelor attribute: *Refinanțare* (dacă și-a refinanțat creditul), *Starea civilă* și *Venit* (să spunem lunar, în lei).

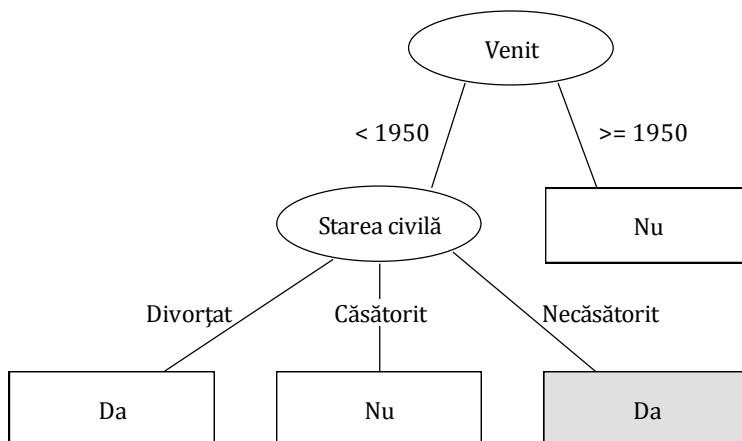
Refinanțare	Starea civilă	Venit	Nerambursare
Da	Necăsătorit	2500	Nu
Nu	Căsătorit	2000	Nu
Nu	Necăsătorit	1400	Nu
Da	Căsătorit	2400	Nu
Nu	Divorțat	1900	Da
Nu	Căsătorit	1200	Nu
Da	Divorțat	4400	Nu
Nu	Necăsătorit	1700	Da
Nu	Căsătorit	1500	Nu
Nu	Necăsătorit	1800	Da

Când începem să aplicăm procedura de construire a arborelui de decizie, constatăm că partiiionările după *Starea civilă* și după *Venit* sunt la fel de bune, având același câștig informațional. În funcție de această decizie inițială, rezultatele pot fi foarte diferite.



Dacă se alege partiționarea după *Starea civilă*, arborele rezultat va fi cel din figura anterioară.

Dacă se alege partiționarea alternativă după *Venit*, arborele rezultat va fi următorul.



În frunza *Da* marcată cu gri, mulțimea de date este cea din tabelul de mai jos.

Refinanțare	Stare civilă	Venit	Nerambursare
Da	Necăsătorit	2500	Nu
Nu	Căsătorit	2000	Nu
Nu	Necăsătorit	1400	Nu
Da	Căsătorit	2400	Nu
Nu	Divorțat	1900	Da
Nu	Căsătorit	1200	Nu
Da	Divorțat	4400	Nu
Nu	Necăsătorit	1700	Da
Nu	Căsătorit	1500	Nu
Nu	Necăsătorit	1800	Da

Prin excluderea atributelor deja tratate, nu mai există alte opțiuni pentru teste care să diferențieze clasele instanțelor rămase și se ajunge într-o

frunză pe care o marcăm cu clasa majoritară a instanțelor sale. Arborele de decizie are o eroare chiar pe mulțimea de antrenare, ceea ce înseamnă că nu a putut fi creat un model suficient de bun pentru datele disponibile.

În general, erorile corespunzătoare mulțimii de test sunt mai numeroase decât cele corespunzătoare mulțimii de antrenare.

7.9. Câștigul proporțional

O altă problemă care afectează performanțele acestui tip de algoritmi este faptul că măsura câștigului informațional favorizează atributele cu un număr mare de valori. De exemplu, dacă am fi utilizat în clasificare coloana Nr. *instanță*, cu 14 valori diferite, entropia medie ar fi fost 0, încărcat ar fi rezultat o partiționare cu 14 fii omogeni. Capacitatea de generalizare este în mod clar afectată de suprapotrivire.

În acest scop, s-a propus măsura *câștigului proporțional* (engl. “gain ratio”, Quinlan, 1986).

Folosind semnificația notațiilor din ecuațiile 7.3 și 7.4, se introduce noțiunea de *informație intrinsecă*:

$$I = - \sum_{i=1}^k \frac{n_i}{n} \cdot \log_2 \frac{n_i}{n} \quad (7.6)$$

Câștigul proporțional este definit drept raportul dintre câștigul informațional și informația intrinsecă:

$$CP = \frac{\Delta}{I} \quad (7.7)$$

La alegerea unui atribut pentru partitioare, câştigul proporţional trebuie maximizat.

7.10. Alți algoritmi de inducție a arborilor de decizie

Procedura prezentată de construire a arborilor de decizie este euristică și deci nu garantează obținerea arborelui optim. Găsirea celui mai bun arbore posibil este o problemă de optimizare NP dificilă, care presupune încercarea tuturor combinațiilor posibile de attribute. În general, utilizând criterii și metode diferite de partitioare, pot rezulta mai mulți arbori pentru aceeași mulțime de antrenare.

Algoritmul *C4.5* (Quinlan, 1993) este o extensie a algoritmului *ID3* care, pe lângă tratarea atributelor continue, permite și partitioarea de mai multe ori după același atribut. De asemenea, poate „reteza” sau simplifica arborele generat (engl. “pruning”) pentru a crește capacitatea de generalizare. Dacă arborele este prea mare (are prea multe frunze), este că și cum am crea câte o regulă pentru fiecare instanță, ceea ce evident poate conduce la suprapotrivire. După construirea unui arbore, anumite ramuri cu prea puține instanțe și care nu au un aport foarte important la clasificare pot fi retezate, astfel încât să crească şansele de a generaliza mai bine. Dacă datele sunt afectate de zgomot, retezarea poate elimina instanțele eronate.

Există și o modalitate aleatorie de generare a arborilor (*Random Tree*). În acest caz nu se mai folosește un criteriu de omogenitate, ci se alege aleatoriu un atribut după care se face partitioarea. Arborii sunt de dimensiuni mai mari, pe mulțimea de antrenare dau erori în general nule, dar nu se garantează o capacitate de generalizare la fel de bună. În practică însă, funcționează destul de bine.

Mai există și o tehnică ce grupează mai mulți arbori aleatorii (*Random Forest*), în care arborii sunt construiți separat și pentru clasificarea unei noi instanțe fiecare dă un vot privind clasa. Rezultatul este clasa care obține cele mai multe voturi.

7.11. Concluzii

Printre avantajele clasificării cu arbori de decizie, menționăm:

- Sunt relativ ușor de construit, deși necesită totuși o serie de calcule;
- Sunt rapizi la clasificarea instanțelor noi;
- Sunt ușor de interpretat, mai ales pentru arbori de dimensiuni mici;
- Un arbore de decizie poate fi interpretat ca o mulțime de reguli, de exemplu: „Dacă vremea este însorită și umiditatea este mai mică sau egală cu 77,5, atunci se poate juca golf”.

Clasificatorul bayesian naiv

8.1. Modelul teoretic

Metoda de clasificare bayesiană naivă (engl. “Naïve Bayes”) se bazează pe calcularea probabilităților ca o anumită instanță să aparțină claselor problemei.

Într-o rețea bayesiană, se evita calcularea întregii distribuții comune de probabilitate după regula de înmulțire a probabilităților (engl. “chain rule”) presupunând că un nod depinde doar de părinții săi din graf. În metoda naivă, presupunerea simplificatoare este și mai puternică, considerând că toate atributele sunt independente dată fiind clasa. Acest fapt nu este neapărat adevărat, de cele mai multe ori, dimpotrivă, condiția de independență poate să nu fie satisfăcută. Cu toate acestea, s-a constatat că deseori metoda are rezultate foarte bune.

Formal, se consideră fiecare atribut și clasa ca variabile aleatorii. Se dă o instanță definită de valorile atributelor (A_1, \dots, A_n) . Scopul este determinarea clasei C pentru această combinație de valori, ceea ce este echivalent cu găsirea valorii C_j care maximizează probabilitatea clasei dată fiind instanța:

$$C^* = \operatorname{argmax}_{c_j} P(C_j | A_1, \dots, A_n). \quad (8.1)$$

Această probabilitate trebuie estimată direct din datele mulțimii de antrenare, pe baza frecvențelor relative de apariție a valorilor atributelor.

Conform teoremei lui Bayes:

$$P(C_j|A_1, \dots, A_n) = \frac{P(A_1, \dots, A_n|C_j) \cdot P(C_j)}{P(A_1, \dots, A_n)}. \quad (8.2)$$

$P(A_1, \dots, A_n)$ este aceeași pentru toate valorile clasei întrucât este vorba despre aceeași instanță pentru toate clasele. Ea depinde doar de valorile atributelor instanței și nu de clase, astfel încât o putem ignora atunci când vrem să maximizăm cantitatea din partea dreaptă a ecuației (8.2). Problema de clasificare devine echivalentă cu alegerea valorii clasei care maximizează numărătorul:

$$C^* = \operatorname{argmax}_{C_j} P(A_1, \dots, A_n|C_j) \cdot P(C_j). \quad (8.3)$$

Rămâne de estimat probabilitatea instanței dată fiind clasa. Considerând că toate atributele sunt independente dată fiind clasa (presupunerea fundamentală a metodei bayesiene naive), putem exprima acest produs sub forma:

$$P(A_1, \dots, A_n|C_j) = P(A_1|C_j) \cdot \dots \cdot P(A_n|C_j). \quad (8.4)$$

După cum vom vedea în continuare, putem estima ușor din datele mulțimii de antrenare $P(A_i|C_j)$ pentru toate valorile atributelor A_i și clasei C_j . Problema de clasificare devine următoarea:

$$C^* = \operatorname{argmax}_{C_j} P(C_j) \cdot \prod_{i=1}^n P(A_i|C_j). \quad (8.5)$$

Metoda se reduce la găsirea clasei pentru care valoarea produsului este maximă.

8.2. Probleme cu attribute simbolice

Pentru a exemplifica modalitatea de calcul, vom considera aceeași problemă ca în capitolul 7, privind oportunitatea de a juca golf sau nu (tabelul 8.1).

Tabelul 8.1. Problemă de clasificare cu attribute simbolice

Nr. instanță	Starea vremii	Temperatură	Umiditate	Vânt	Joc
1	Soare	Mare	Mare	Absent	Nu
2	Soare	Mare	Mare	Prezent	Nu
3	Înnorat	Mare	Mare	Absent	Da
4	Ploaie	Medie	Mare	Absent	Da
5	Ploaie	Mică	Normală	Absent	Da
6	Ploaie	Mică	Normală	Prezent	Nu
7	Înnorat	Mică	Normală	Prezent	Da
8	Soare	Medie	Mare	Absent	Nu
9	Soare	Mică	Normală	Absent	Da
10	Ploaie	Medie	Normală	Absent	Da
11	Soare	Medie	Normală	Prezent	Da
12	Înnorat	Medie	Mare	Prezent	Da
13	Înnorat	Mare	Normală	Absent	Da
14	Ploaie	Medie	Mare	Prezent	Nu

Metoda bayesiană naivă clasifică o instanță dată, care poate să fie una nouă, care nu aparține mulțimii de antrenare.

În cazul de față, fie această instanță: $x_q = (\text{Soare}, \text{Mare}, \text{Normală}, \text{Absent})$.

Trebuie să realizăm un număr de calcule egal cu numărul de clase. Apoi vom alege clasa pentru care probabilitatea de apartenență a instanței este maximă.

Mai întâi vom calcula produsul pentru clasa $Joc = Da$ (J_D). Din tabelul sortat după valoarea clasei, se observă că această valoare apare de 9 ori.

Nr. instanță	Starea vremii	Temperatură	Umiditate	Vânt	Joc
3	Înnorat	Mare	Mare	Absent	Da
4	Ploaie	Medie	Mare	Absent	Da
5	Ploaie	Mică	Normală	Absent	Da
7	Înnorat	Mică	Normală	Prezent	Da
9	Soare	Mică	Normală	Absent	Da
10	Ploaie	Medie	Normală	Absent	Da
11	Soare	Medie	Normală	Prezent	Da
12	Înnorat	Medie	Mare	Prezent	Da
13	Înnorat	Mare	Normală	Absent	Da
1	Soare	Mare	Mare	Absent	Nu
2	Soare	Mare	Mare	Prezent	Nu
6	Ploaie	Mică	Normală	Prezent	Nu
8	Soare	Medie	Mare	Absent	Nu
14	Ploaie	Medie	Mare	Prezent	Nu

Prin urmare:

$$P(J_D) = \frac{9}{14}.$$

Pentru calculul probabilităților condiționate din produs, numărăm instanțele care au valoarea dorită a atributului, numai în clasa considerată. De exemplu, pentru atributul *Starea vremii*, ne interesează câte instanțe au valoarea *Soare* (dată de instanța de interogare x_q).

Nr. instantă	Starea vremii	Joc
9	Soare	Da
11	Soare	Da
3	Înnorat	Da
7	Înnorat	Da
12	Înnorat	Da
13	Înnorat	Da
4	Ploaie	Da
5	Ploaie	Da
10	Ploaie	Da

Din tabelul de mai sus, se poate vedea că numai 2 instanțe din cele 9 din clasa *Da* au valoarea *Soare*. Deci:

$$P(S_S|J_D) = \frac{2}{9}$$

Analog se procedează și pentru restul atributelor, obținând:

$$P(T_H|J_D) = \frac{2}{9}$$

$$P(U_N|J_D) = \frac{6}{9}$$

$$P(V_A|J_D) = \frac{6}{9}$$

Aceleași calcule se realizează pentru clasa *Nu* (*N*):

$$P(J_N) = \frac{5}{14}$$

$$P(S_S|J_N) = \frac{3}{5}$$

$$P(T_H|J_N) = \frac{2}{5}$$

$$P(U_N|J_N) = \frac{1}{5}$$

$$P(V_A|J_N) = \frac{2}{5}.$$

Putem calcula acum produsele pentru fiecare clasă:

$$P(J_D) \cdot P(x_q|J_D) = \frac{9}{14} \cdot \frac{2}{9} \cdot \frac{2}{9} \cdot \frac{6}{9} \cdot \frac{6}{9} = 14,109 \cdot 10^{-3},$$

$$P(J_N) \cdot P(x_q|J_N) = \frac{5}{14} \cdot \frac{3}{5} \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{2}{5} = 6,857 \cdot 10^{-3}.$$

Valoarea maximă este prima și deci instanța va fi clasificată în clasa *Da*.

8.3. Considerente practice

8.3.1. Corecția Laplace

Deoarece clasificarea se bazează pe calcularea unor produse, dacă un factor este 0, întregul produs devine 0. Să considerăm următoarea mulțime de antrenare:

Starea vremii	Umiditate	Joc
Înnorat	Mare	Da
Ploaie	Mare	Da
Înnorat	Mare	Da
Soare	Mare	Nu
Soare	Mare	Nu
Ploaie	Normală	Nu

și instanța pe care dorim să o clasificăm: $x_q = (\hat{\text{Innorat}}, \text{Normală})$.

Mai întâi realizăm calculele pentru clasa *Da*:

$$P(J_D) = \frac{3}{6}$$

$$P(S_I|J_D) = \frac{2}{3}$$

$$P(U_N|J_D) = \frac{0}{3}.$$

Apoi realizăm calculele pentru clasa *Nu*:

$$P(J_N) = \frac{3}{6}$$

$$P(S_I|J_N) = \frac{0}{3}$$

$$P(U_N|J_N) = \frac{1}{3}.$$

Calculând produsele de probabilități, se vede că ambele sunt 0 și deci nu putem lua nicio decizie de clasificare a instanței x_q :

$$P(J_D) \cdot P(x_q|J_D) = \frac{3}{6} \cdot \frac{2}{3} \cdot \frac{0}{3} = 0,$$

$$P(J_N) \cdot P(x_q|J_N) = \frac{3}{6} \cdot \frac{0}{3} \cdot \frac{1}{3} = 0.$$

În general, pentru mai multe clase, dacă în fiecare produs există câte un factor nul, atunci toate produsele, pentru toate clasele, se anulează. Totuși, ceilalți factori nenuli ai produsului ne-ar putea da informații relevante pentru clasificare. În acest sens, există metode care garantează că niciun produs nu va fi 0. În locul calculului tipic al frecvențelor relative ca raport între numărul de apariții a unei valori a atributului i în clasa j (n_{ij}) și numărul de apariții a valorii clasei j (n_j):

$$P(A_i | C_j) = \frac{n_{ij}}{n_j}, \quad (8.6)$$

se poate folosi *estimarea-m* (engl. “m-estimate”) care „netezește” probabilitățile aplicând următoarea formulă de calcul:

$$P(A_i | C_j) = \frac{n_{ij} + mp}{n_j + m}. \quad (8.7)$$

Corecția Laplace poate fi considerată un caz particular al estimării-m unde, dacă c este numărul de clase, putem considera $m = c$ și $p = 1/c$, rezultând:

$$P(A_i | C_j) = \frac{n_{ij} + 1}{n_j + c}. \quad (8.8)$$

Practic, se adaugă la numărator 1 și la numitor numărul de clase. În acest mod, toți factorii vor avea valori $v \in (0, 1)$. Probabilitățile sunt estimate ca frecvențe relative din date, dar nu cunoaștem valorile absolute. Teoretic, ar putea să mai existe instanțe pe care încă nu le-am întâlnit și atunci considerăm a-priori că mai există câte o instanță din fiecare clasă.

Din punct de vedere filosofic, faptul că probabilitățile nu pot fi 0 sau 1 ne conduce la ideea că nu putem fi siguri niciodată de ceva că e adevărat sau fals în mod absolut.

Pentru exemplul simplificat, vom avea acum:

$$P(J_D) \cdot P(x_q | J_D) = \frac{3}{6} \cdot \frac{2+1}{3+2} \cdot \frac{0+1}{3+2} = 0,06$$

$$P(J_N) \cdot P(x_q | J_N) = \frac{3}{6} \cdot \frac{0+1}{3+2} \cdot \frac{1+1}{3+2} = 0,04$$

și deci se poate lua o decizie (*Da*), iar rezultatul este conform cu analiza mulțimii de antrenare, unde valoarea *Innorat* apare în clasa *Da* de două ori și valoarea *Normală* apare în clasa *Nu* o singură dată.

Pentru exemplul inițial din secțiunea 8.2, aplicând corecția Laplace vom avea:

$$P(J_D) \cdot P(x_q | J_D) = \frac{9}{14} \cdot \frac{2+1}{9+2} \cdot \frac{2+1}{9+2} \cdot \frac{6+1}{9+2} \cdot \frac{6+1}{9+2} = 19,363 \cdot 10^{-3}$$

$$P(J_N) \cdot P(x_q | J_N) = \frac{5}{14} \cdot \frac{3+1}{5+2} \cdot \frac{2+1}{5+2} \cdot \frac{1+1}{5+2} \cdot \frac{2+1}{5+2} = 10,71 \cdot 10^{-3}.$$

Rezultatul clasificării nu se schimbă deoarece contează doar comparația, nu cantitățile propriu-zise. Pentru metoda bayesiană naivă, partea calitativă este mai importantă decât partea cantitativă.

Corecția Laplace este foarte utilă mai ales la clasificarea textelor, unde atributele sunt cuvintele însele și, având multe documente, este probabil ca din acestea să lipsească anumiți termeni.

8.3.2. Precizia calculelor

O altă problemă care poate să apară este faptul că un produs de probabilități subunitare cu mulți factori poate fi afectat de precizia reprezentării numerelor. Astfel, dacă valoarea produsului devine mai mică decât cantitatea minimă care poate fi reprezentată în virgulă mobilă, rezultatul va deveni 0.

O soluție este *logaritmarea* și în acest caz, produsul de probabilități este înlocuit cu suma logaritmilor de probabilități.

De exemplu, pentru calculul $P(J_D) \cdot P(x_q|J_D)$ de mai sus, vom avea:

$$\begin{aligned}\ln(P(J_D) \cdot P(x_q|J_D)) &= \ln \frac{9}{14} + \ln \frac{2+1}{9+2} + \ln \frac{2+1}{9+2} + \ln \frac{6+1}{9+2} + \ln \frac{6+1}{9+2} \\ &= -0,442 - 1,299 - 1,299 - 0,452 - 0,452 = -3,924 \\ \Rightarrow P(J_D) \cdot P(x_d|J_D) &= e^{-3,924} \cong 19,363 \cdot 10^{-3}.\end{aligned}$$

Pentru simplitate, mai sus am inclus doar 3 zecimale. Desigur, pentru o precizie suficientă a rezultatului, reprezentarea termenilor sumei trebuie să fie corespunzătoare.

8.4. Probleme cu atrbute numerice

Pentru atrbute numerice, există mai multe modalități de abordare. După cum am explicat în secțiunea 7.2, valorile acestora pot fi discretizate, rezultând atrbute ordonale. De asemenea, se poate aplica partiționarea binară: se alege o valoare de referință iar valorile atrbutului rezultat vor fi

Da sau *Nu*, dacă valorile atributului inițial sunt mai mari sau mai mici, respectiv, decât referința.

O abordare specifică metodei bayesiene naive este mai complexă și mai puțin folosită în practică, însă interesantă din punct de vedere conceptual. Ideea de bază este asemănătoare cu aceea de la corecția Laplace sau estimarea-m: estimarea distribuției de probabilitate. Putem presupune de exemplu că valorile atributului numeric respectă o distribuție normală (gaussiană). Nu este obligatoriu să respecte această distribuție; dacă știm că datele urmează alte distribuții, putem estima parametrii acestora. Pentru distribuția normală, parametrii pe care trebuie să îi determinăm sunt media μ și deviația standard σ :

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (8.9)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (8.10)$$

Probabilitățile utilizate apoi în clasificarea bayesiană naivă sunt calculate din distribuția normală:

$$P(A_i | C_j) = \frac{1}{\sqrt{2\pi} \cdot \sigma_{ij}} e^{-\frac{(A_i - \mu_{ij})^2}{2\sigma_{ij}^2}} \quad (8.11)$$

Pentru a exemplifica, vom considera mulțimea de date cu atrbute numerice analizată și în capitolul precedent (tabelul 8.2):

Tabelul 8.2. Problemă de clasificare cu atribute mixte

Nr. instanță	Starea vremii	Temperatură	Umiditate	Vânt	Joc
1	Soare	29,4	85	Absent	Nu
2	Soare	26,7	90	Prezent	Nu
3	Înnorat	28,3	86	Absent	Da
4	Ploaie	21,1	96	Absent	Da
5	Ploaie	20,0	80	Absent	Da
6	Ploaie	18,3	70	Prezent	Nu
7	Înnorat	17,8	65	Prezent	Da
8	Soare	22,2	95	Absent	Nu
9	Soare	20,6	70	Absent	Da
10	Ploaie	23,9	80	Absent	Da
11	Soare	23,9	70	Prezent	Da
12	Înnorat	22,2	90	Prezent	Da
13	Înnorat	27,2	75	Absent	Da
14	Ploaie	21,7	91	Prezent	Nu

și ne propunem clasificarea instanței: $x_q = (\text{Soare}, 26, 72, \text{Absent})$.

Pentru atributele simbolice, calculele sunt aceleași ca în secțiunea 8.2. Pentru atributul *Temperatură*, pentru clasa *Da*, media valorilor este 22,778 iar deviația standard este 3,214. Prin urmare:

$$P(T_{26}|J_D) = \frac{1}{\sqrt{2\pi} \cdot 3,214} e^{-\frac{(26-22,778)^2}{2 \cdot 3,214^2}} = 0,0751.$$

Pentru clasa *Nu*, media valorilor este 23,66 iar deviația standard este 3,922 și vom avea:

$$P(T_{26}|J_N) = 0,0851.$$

Pentru atributul *Umiditate*, probabilitățile vor fi:

$$P(U_{72}|J_D) = \frac{1}{\sqrt{2\pi} \cdot 9,631} e^{-\frac{(72-79,111)^2}{2 \cdot 9,631^2}} = 0,0315,$$

$$P(U_{72}|J_N) = \frac{1}{\sqrt{2\pi} \cdot 8,704} e^{-\frac{(72-86,2)^2}{2 \cdot 8,704^2}} = 0,0121.$$

În final:

$$P(J_D) \cdot P(x_q|J_D) = \frac{9}{14} \cdot \frac{2+1}{9+2} \cdot 0,0751 \cdot 0,0315 \cdot \frac{6+1}{9+2} = 0,264 \cdot 10^{-3}$$

$$P(J_N) \cdot P(x_q|J_N) = \frac{5}{14} \cdot \frac{3+1}{5+2} \cdot 0,0851 \cdot 0,0121 \cdot \frac{2+1}{5+2} = 0,09 \cdot 10^{-3}$$

și rezultatul este din nou clasa *Da*.

8.5. Concluzii

Dintre avantajele metodei de clasificare bayesiene naive menționăm calculele simple și robustețea la zgomot și atributе irelevante. De aceea, este foarte potrivită pentru mulțimi de antrenare de dimensiuni medii sau mari (de exemplu pentru clasificarea documentelor text, detecția spam-ului, diagnoză etc.).

Chiar dacă se bazează pe independența atributelor dată fiind clasa, metoda funcționează de multe ori bine chiar și atunci când presupunerea este infirmată în realitate.

Clasificarea bazată pe instanțe

9.1. Introducere

O altă metodă de clasificare este cea *bazată pe instanțe*, care presupune memorarea efectivă a tuturor instanțelor de antrenare. Mai ales în cazul arborilor de decizie, pentru a clasifica noi instanțe încercăm să realizăm un model al datelor de antrenare. Dacă avem o mulțime de 10000 de instanțe, este teoretic posibil să construim un arbore de decizie de dimensiuni (mult) mai mici, care să le modeleze. Aici, memorăm pur și simplu toate informațiile iar clasificarea presupune estimarea similarității unei noi instanțe față de cele existente, la fel cum clasifică oamenii noi obiecte sau situații prin analogie cu cele cunoscute deja.

Instanțele, definite de valorile atributelor lor, pot fi văzute ca niște puncte într-un spațiu n -dimensional, unde n este numărul de atrbute.

De exemplu, să considerăm o problemă de estimare a riscului cardiovascular al unor pacienți, ținând seama de vârstă (în ani) și de indicele masei corporale: $IMC = G/I^2$, unde G este greutatea (în kilograme) iar I este înățimea (în metri).

După cum se poate vedea în figura 9.1, dacă avem două atrbute numerice, fiecare instanță reprezintă un punct în plan. Instanțele încercuite semnifică pacienți cu risc crescut, iar cele neîncercuite pacienți cu risc scăzut.

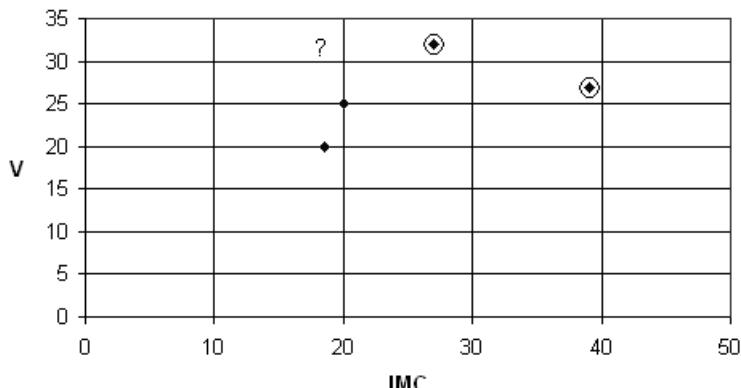


Figura 9.1. Reprezentare grafică a unei probleme de clasificare

În acest plan, calculăm cât de aproape este o nouă instanță, marcată cu „?”, față de cele de antrenare.

În cazul algoritmului *cel mai apropiat vecin* (engl. “nearest neighbor”, *NN*) identificăm instanța din mulțimea de antrenare cea mai apropiată de instanța de interogare și rezultatul clasificării este clasa instanței respective din mulțimea de antrenare.

În cazul algoritmului *cei mai apropiati k vecini* (engl. “k-nearest neighbor”, *kNN*) identificăm cele mai apropiate k instanțe și clasificăm noua instanță pe baza votului majoritar al acestor vecini. Dintre cele k instanțe de antrenare selectate, se numără câte aparțin fiecărei clase a problemei iar rezultatul este clasa în care se găsesc cele mai multe.

9.2. Metrici de distanță

Pentru a vedea „cât de apropiate” sunt instanțele, avem nevoie de o metrică de distanță. De obicei se folosesc particularizări ale distanței Minkowski:

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (9.1)$$

unde \mathbf{x} și \mathbf{y} sunt vectori n -dimensionali iar p este un parametru.

Când $p = 2$ formula indică distanța euclidiană. Când $p = 1$ formula indică distanța Manhattan. Aceste două metrii de distanță sunt cele mai utilizate în practică.

De exemplu, să considerăm două puncte bidimensionale A și B , definite de coordonatele {1, 2}, respectiv {4, 6}. Prin urmare, $\Delta x = 3$ și $\Delta y = 4$. Figura 9.2 indică distanța euclidiană și distanța Manhattan dintre cele două puncte.

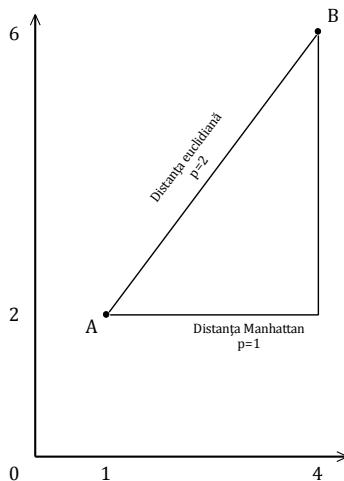


Figura 9.2. Distanța euclidiană și distanța Manhattan

În tabelul 9.1 se calculează valoarea distanței dintre puncte pentru diferite valori ale parametrului p .

Tabelul 9.1. Valorile distanței când parametrul p variază

p	d
$p = 1$	$d = 3 + 4 = 7$
$p = 2$	$d = \sqrt{3^2 + 4^2} = 5$
$p = 3$	$d = \sqrt[3]{3^3 + 4^3} = 4,498$
$p = 10$	$d = \sqrt[10]{3^{10} + 4^{10}} = 4,022$
$p \rightarrow \infty$	$d = \max(3, 4) = 4$

9.3. Scalarea atributelor

În mulțimea de date putem avea atribute de orice fel, cu orice fel de valori. Putem avea de exemplu:

- Înălțimea unei persoane $\in [1,5, 2,1]$ m;
- Greutatea unei persoane $\in [50, 120]$ kg;
- Venitul unei persoane $\in [9600, 60000]$ lei/an.

Toate aceste atribute intervin în clasificare și pentru calculul distanței; dacă vom considera primul și al treilea atribut, este evident că va conta în mult mai mare măsură ultimul. De exemplu, să comparăm două persoane cu venituri 30000 și 31000 și înălțime 1,5 și 2,1. Pentru venituri, aceasta este o diferență mică, în timp ce înălțimile sunt la marginile intervalului. În schimb, valoarea absolută a diferenței de venit este atât de mare, încât domină clasificarea.

De aceea, se folosește în general normalizarea atributelor, ca un pas de preprocesare a datelor, astfel încât toate atributele să aibă valori între 0 și 1, aplicând formula de transformare:

$$x'_i = \frac{(x_i - \min_i)}{(\max_i - \min_i)} \in [0,1] \quad (9.2)$$

9.4. Calculul distanțelor pentru diferitele tipuri de attribute

Deoarece algoritmii NN și kNN se bazează pe calculul unor distanțe, trebuie să definim diferențele dintre valorile atributelor din punct de vedere numeric. Vom descrie niște metode în acest scop pentru fiecare tip de atribut.

Pentru attributele nominale, distanța dintre valorile unui atribut se consideră 0 dacă valorile sunt egale și 1 dacă sunt diferite. Între valorile atributului nu există alte relații și prin urmare nu există distanțe intermediare între 0 și 1.

Pentru attributele ordinale, se pot considera valorile egal distribuite între 0 și 1. De exemplu, pentru trei valori { *Mic*, *Mediu*, *Mare* }, se poate considera transformarea: *Mic* = 0, *Mediu* = 0,5 și *Mare* = 1. Calculul distanțelor va fi valoarea absolută a acestor valori numerice, precum: $d(Mic, Mare) = 1$, $d(Mic, Mediu) = d(Mediu, Mare) = 0,5$ și.a.m.d.

Pentru attributele numerice, distanța este valoarea absolută a diferenței dintre valorile normalizate ale atributelor.

9.5. Numărul optim de vecini

Decizia privind numărul de vecini considerați este foarte importantă. După cum se poate vedea în figura 9.3, mărind numărul de vecini (implicit

distanță față de instanța de interogare), putem să luăm în calcul doi, trei vecini sau chiar toate instanțele de antrenare. Însă dacă instanțele unei clase sunt grupate, cum este cazul de obicei, pe măsură ce mărim distanța și includem mai mulți vecini, s-ar putea să ne îndepărtem prea mult și să includem instanțe din alte clase, care vor afecta rezultatul clasificării.

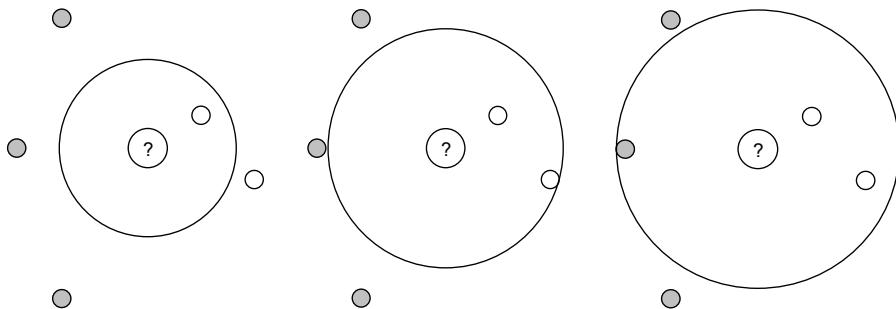


Figura 9.3. Vecinătățile unui punct cu 1, 2 și 3 vecini

Determinarea numărului optim de vecini k poate fi dificilă. Când k este prea mic, clasificarea poate fi afectată de zgomot. Când k este prea mare, vecinătatea poate include puncte din alte clase.

Figura 9.4 prezintă regiunile de decizie pentru o problemă arbitrară de clasificare binară cu instanțe de antrenare bidimensionale „albe” și „negre”, culoarea indicând clasa instanței. Regiunile de decizie indică deciziile de clasificare pentru toate punctele din plan – fiecare punct este considerat o instanță de interogare care trebuie clasificată în funcție de instanțele de antrenare date.

După cum se vede din această figură, la această metodă de clasificare regiunile de decizie sunt oarecum neregulate dar interesante, ceea ce le conferă și o anumită calitate estetică.



Figura 9.4. Regiuni de decizie pentru o problemă de clasificare binară

Atunci când datele de antrenare sunt afectate de zgomot, clasa celui mai apropiat vecin ar putea fi greșită, iar mai mulți vecini ar putea compensa erorile. Figura 9.5 prezintă regiunile de decizie într-o astfel de situație pentru $k = 1$.

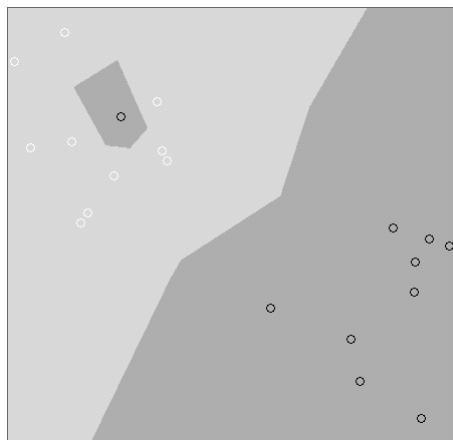


Figura 9.5. Regiuni de decizie pentru date afectate de zgomot cu $k = 1$

Figura 9.6 prezintă regiunile de decizie pentru $k = 3$.

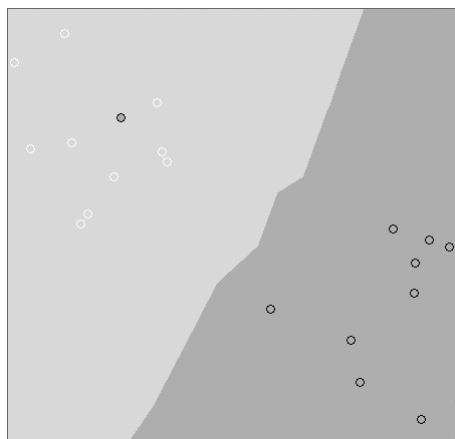


Figura 9.6. Regiuni de decizie pentru date afectate de zgomot cu $k = 3$

Se vede că în ciuda existenței unei instanțe „negre” în zona „albă”, regiunea de decizie „albă” nu mai conține acum o subregiune „neagră”.

În general, nu putem spune care din situațiile prezentate în figurile 9.5 și 9.6 este de dorit. Dacă punctul „negru” izolat este corect, acesta poate reprezenta o excepție importantă, iar algoritmul *NN* este o metodă foarte bună pentru a-l lua în considerare. Metoda bayesiană naivă, unde impactul majorității este puternic, poate l-ar fi ignorat. Pentru un arbore de decizie ar putea reprezenta o eroare pe mulțimea de antrenare, sau dacă se folosește retezarea, frunza corespunzătoare ar putea fi eliminată.

Dacă punctul este însă clasificat greșit, este bine să fie eliminat sau influența lui să fie redusă, ceea ce se poate realiza considerând un număr mai mare de vecini. Pentru fiecare abordare există avantaje și dezavantaje.

În general, numărul optim de vecini se poate determina prin validare încrucișată, metodă prezentată în secțiunea 6.5.

9.6. Blestemul dimensionalității

O altă problemă la care sunt sensibile metodele bazate de instanțe este aşa numitul „blestem al dimensionalității”. Dacă ne imaginăm un segment de dreaptă între 0 și 1, lungimea sa este 1. Dacă ne imaginăm un pătrat, lungimea între vârfurile opuse, de coordonate (0,0) și (1,1), este $\sqrt{2}$. Dacă ne imaginăm un cub, lungimea diagonalei este $\sqrt{3}$. Dacă avem un hipercub cu 100 de dimensiuni, lungimea diagonalei este 10. Cu cât crește numărul de atribute, datele devin mai rare în acel spațiu. Dacă avem 3 valori distribuite egal pe diagonale, în spațiul unidimensional distanța între ele este 0,5 (0 – 0,5 – 1), pe când în spațiul cu 100 de dimensiuni, distanța este 5. Cu cât sunt mai rare datele, cu atât este mai greu pentru clasificator să realizeze un model precis.

De asemenea, dacă vom considera un grid pe care valorile sunt egal distribuite, câte 3 pe fiecare dimensiune, pentru a păstra o distanță constantă între valori, în cazul unidimensional avem nevoie de 3 date, în cazul 2D avem nevoie de 9 date, în cazul 3D avem nevoie de 27 date iar în cazul 100D avem nevoie de $3^{100} = 5 \cdot 10^{47}$ date. Odată cu creșterea dimensionalității, necesarul de date pentru a păstra aceeași densitate crește exponențial.

9.7. Ponderarea instanțelor

După cum am menționat, este greu să determinăm cea mai bună valoare pentru numărul de vecini. Putem însă să ne gândim că vecinul cel mai apropiat, fiind mai asemănător, contează mai mult pentru clasificare

decât instanțele mai îndepărtate. Se poate face astfel ponderarea contribuției instanțelor de antrenare în funcție de distanța față de instanța de interogare. O funcție des utilizată pentru ponderare este inversul pătratului distanței:

$$w_i = \frac{1}{d(\mathbf{x}_q, \mathbf{x}_i)^2} \quad (9.3)$$

unde i este indicele unei instanțe de antrenare iar d este o metrică de distanță.

Distanța este o măsură a similarității: când distanța este mare, similaritatea și deci și ponderea corespunzătoare, tind la 0. Când distanța este mică, ponderea crește.

Dacă $d(\mathbf{x}_q, \mathbf{x}_i) = 0$, ceea ce ar presupune ca ponderea să tindă la infinit, dominând oricum clasificarea, se alege direct clasa corespunzătoare instanței din acel punct: $C(\mathbf{x}_q) = C(\mathbf{x}_i)$.

Prin ponderare, putem folosi întreaga mulțime de antrenare pentru clasificare, încrucișând instanțele mai îndepărtate vor avea pondere mai mică și nu vor conta foarte mult. Este că și cum vecinătatea optimă ar fi determinată în mod dinamic. Astfel, se transformă abordarea locală, cu o submulțime de k instanțe, într-o abordare globală.

9.8. Ponderarea și selecția atributelor

Selecția atributelor este o problemă esențială din punct de vedere psihologic. Ponderea atributelor, nu a instanțelor, este foarte importantă

pentru modul în care iau oamenii decizii. Aceste ponderi semnifică ceea ce considerăm mai semnificativ la un moment dat.

Să spunem că o persoană culege flori, fără nicio altă constrângere. Atributele cele mai importante pot fi culoarea, mirosul, aspectul estetic etc. Dar dacă merge să caute plante medicinale, importanța atributelor se modifică pentru a respecta noile criterii – proprietățile medicinale (după Miclea, 1999).

Instanțele (plantele) și attributele (criteriile) sunt aceleași, dar decizia de clasificare, de a culege sau nu anumite exemplare, depinde de importanța atributelor.

Scopul omului determină importanța atributelor.

Din punct de vedere computațional, schimbarea ponderii atributelor este echivalentă cu lungirea sau scurtarea axelor în spațiul multidimensional. Atributelor mai importante le vor corespunde axe mai lungi, ceea ce determină distanțe mai mari, ce vor avea un efect mai puternic asupra rezultatului clasificării, chiar dacă valorile instanțelor rămân normalizate între 0 și 1. Axele corespunzătoare atributelor mai puțin importante, care pot fi și irelevante, se scurtează.

O modalitate de a determina importanța atributelor este chiar utilizarea câștigului informațional, descris în secțiunea 7.4.

Există și algoritmi specializați, cum ar *Relief* (Kira & Rendell, 1992; Kononenko, 1994). Ideea de bază este următoarea. Se initializează ponderile atributelor cu 0. Pentru fiecare instanță x_q din mulțimea de antrenare, se găsește cea mai apropiată instanță din aceeași clasă (*hit*) și cea mai apropiată instanță dintr-o clasă diferită (*miss*) și se actualizează ponderea atributului după formula:

$$w_a \leftarrow w_a + \frac{|x_{q,a} - x_{miss,a}| - |x_{q,a} - x_{hit,a}|}{x_{max} - x_{min}} \quad (9.4)$$

Pentru a explica funcționarea sa, vom considera o problemă de clasificare binară (D sau N) cu două atribute. Instanțele au următoarele valori pentru atributul 1, în corespondență cu clasa:

Atributul 1	Clasa
$x_{11} = 1$	$C = N$
$x_{21} = 1,5$	$C = N$
$x_{31} = 2$	$C = D$
$x_{41} = 3$	$C = D$

Algoritmul se bazează pe o serie de iterații, în care valorile sunt normalizeze, diferența dintre valoarea maximă și cea minimă fiind $3 - 1 = 2$:

$$x_{11}: \text{hit } x_{21}, \text{ miss } x_{31}, \Delta w_1 = (1 - 0,5) / 2 = 0,25 \Rightarrow w_1 = 0,25$$

$$x_{21}: \text{hit } x_{11}, \text{ miss } x_{31}, \Delta w_1 = (0,5 - 0,5) / 2 = 0 \Rightarrow w_1 = 0,25$$

$$x_{31}: \text{hit } x_{41}, \text{ miss } x_{21}, \Delta w_1 = (0,5 - 1) / 2 = -0,25 \Rightarrow w_1 = 0$$

$$x_{41}: \text{hit } x_{31}, \text{ miss } x_{21}, \Delta w_1 = (1,5 - 1) / 2 = 0,25 \Rightarrow w_1 = 0,25$$

În această situație, $w_1 = 0,25$.

Pentru atributul 2, să presupunem că instanțele au următoarele valori în corespondență cu clasa:

Atributul 2	Clasa
$x_{12} = 1$	$C = N$
$x_{22} = 2$	$C = N$
$x_{32} = 1,5$	$C = D$
$x_{42} = 3$	$C = D$

x_{12} : hit x_{22} , miss x_{32} , $\Delta w_2 = (0,5 - 1) / 2 = -0,25 \Rightarrow w_2 = -0,25$

x_{22} : hit x_{12} , miss x_{32} , $\Delta w_2 = (0,5 - 1) / 2 = -0,25 \Rightarrow w_2 = -0,5$

x_{32} : hit x_{42} , miss x_{12} , $\Delta w_2 = (0,5 - 1,5) / 2 = -0,5 \Rightarrow w_2 = -1$

x_{42} : hit x_{32} , miss x_{22} , $\Delta w_2 = (1 - 1,5) / 2 = -0,25 \Rightarrow w_2 = -1,25$

În această situație, $w_2 = -1,25$.

Se observă că atributul 1 este mai important, deoarece separă mai ușor cele două clase, cu o singură valoare de referință: 1,75.

Spre deosebire de metoda bayesiană naivă, algoritmii NN și kNN sunt mai sensibili la prezența atributelor irelevante. De aceea, eliminarea acestora poate îmbunătăți foarte mult performanțele de clasificare. Cunoșcând ponderile atributelor, poate fi selectată o submulțime de atribute mai importante. Acest fapt conduce de asemenea la scăderea numărului de dimensiuni și la creșterea vitezei clasificării.

9.9. Exemplu de clasificare

Pentru exemplificarea celor doi algoritmi, NN și kNN , vom considera aceeași problemă a jocului de golf în funcție de condițiile meteorologice, cu atribute atât simbolice cât și numerice, prezentată din nou în tabelul 9.1.

Ne propunem clasificarea instanței: $x_q = (\text{Soare}, 26, 72, \text{Absent})$.

Putem interpreta atributul *Starea vremii* ca fiind ordinal, cu ordinea valorilor {*Ploaie*, *Înnorat*, *Soare*}. Acestea vor lua valorile numerice 0, 0,5, respectiv 1. Atributul *Vânt* este binar și putem transforma valorile simbolice *Absent* și *Prezent* în valorile numerice 0, respectiv 1. Mulțimea de antrenare va deveni acum cea din tabelul 9.2.

Tabelul 9.1. Problemă de clasificare cu atribute mixte

Nr. instanță	Starea vremii	Temperatură	Umiditate	Vânt	Joc
1	Soare	29,4	85	Absent	Nu
2	Soare	26,7	90	Prezent	Nu
3	Înnorat	28,3	86	Absent	Da
4	Ploaie	21,1	96	Absent	Da
5	Ploaie	20,0	80	Absent	Da
6	Ploaie	18,3	70	Prezent	Nu
7	Înnorat	17,8	65	Prezent	Da
8	Soare	22,2	95	Absent	Nu
9	Soare	20,6	70	Absent	Da
10	Ploaie	23,9	80	Absent	Da
11	Soare	23,9	70	Prezent	Da
12	Înnorat	22,2	90	Prezent	Da
13	Înnorat	27,2	75	Absent	Da
14	Ploaie	21,7	91	Prezent	Nu

Tabelul 9.2. Transformarea atributelor simbolice în atribute numerice

Nr. instanță	Starea vremii	Temperatură	Umiditate	Vânt	Joc
1	1	29,4	85	0	Nu
2	1	26,7	90	1	Nu
3	0,5	28,3	86	0	Da
4	0	21,1	96	0	Da
5	0	20,0	80	0	Da
6	0	18,3	70	1	Nu
7	0,5	17,8	65	1	Da
8	1	22,2	95	0	Nu
9	1	20,6	70	0	Da
10	0	23,9	80	0	Da
11	1	23,9	70	1	Da
12	0,5	22,2	90	1	Da
13	0,5	27,2	75	0	Da
14	0	21,7	91	1	Nu

Valorile atributului *Temperatură* aparțin intervalului [17,8, 29,4], iar valorile atributului *Umiditate* aparțin intervalului [65, 96]. Acestea trebuie normalize, conform ecuației 9.2, și sunt prezentate în tabelul 9.3.

Tabelul 9.3. Normalizarea atributelor

Temperatură	Umiditate	Temperatură normalizată	Umiditate normalizată
29,4	85	1,000	0,645
26,7	90	0,767	0,806
28,3	86	0,905	0,677
21,1	96	0,284	1,000
20,0	80	0,190	0,484
18,3	70	0,043	0,161
17,8	65	0,000	0,000
22,2	95	0,379	0,968
20,6	70	0,241	0,161
23,9	80	0,526	0,484
23,9	70	0,526	0,161
22,2	90	0,379	0,806
27,2	75	0,810	0,323
21,7	91	0,336	0,839

Ca metrică de distanță vom utiliza distanța euclidiană. Algoritmii presupun calculul distanțelor dintre x_q și toate instanțele din mulțimea de antrenare. Instanța de interogare trebuie să ea transformată în același mod.

Starea vremii	Temperatură	Umiditate	Vânt	Joc
Soare	26	72	Absent	?
1	0.707	0.226	0	?

Distanța dintre x_q și instanța de antrenare x_1 se calculează luând în considerare toate atrbutele:

$$d(x_q, x_1) = \sqrt{(1 - 1)^2 + (0,707 - 1)^2 + (0,226 - 0,645)^2 + (0 - 0)^2} \\ = 0,5113.$$

Analog se procedează pentru toate celelalte instanțe din mulțimea de antrenare, rezultatele fiind precizate în tabelul 9.4.

Tabelul 9.4. Calculul distanțelor

Nr. instanță	Starea vremii	Temperatură	Umiditate	Vânt	Joc	Distanță
1	1	1	0,645	0	Nu	0,5113
2	1	0,767	0,806	1	Nu	1,1576
3	0,5	0,905	0,677	0	Da	0,7019
4	0	0,284	1	0	Da	1,3334
5	0	0,19	0,484	0	Da	1,1549
6	0	0,043	0,161	1	Nu	1,5637
7	0,5	0	0	1	Da	1,3420
8	1	0,379	0,968	0	Nu	0,8113
9	1	0,241	0,161	0	Da	0,4705
10	0	0,526	0,484	0	Da	1,0485
11	1	0,526	0,161	1	Da	1,0183
12	0,5	0,379	0,806	1	Da	1,3015
13	0,5	0,81	0,323	0	Da	0,5196
14	0	0,336	0,839	1	Nu	1,5854

Pentru atributul *Starea vremii* am făcut presupunerea că este de tip ordinal și de aceea pentru instanțele 3, 7, 12 și 13 distanța pe acest atribut este 0,5. Dacă l-am fi considerat simbolic, aceste distanțe ar fi fost 1.

Tabelul de mai jos conține aceleași date sortate în ordine crescătoare a distanței.

Nr. instanță	Starea vremii	Temperatură	Umiditate	Vânt	Joc	Distanță
9	1	0,241	0,161	0	Da	0,4705
1	1	1	0,645	0	Nu	0,5113
13	0,5	0,81	0,323	0	Da	0,5196
3	0,5	0,905	0,677	0	Da	0,7019
8	1	0,379	0,968	0	Nu	0,8113
11	1	0,526	0,161	1	Da	1,0183
10	0	0,526	0,484	0	Da	1,0485
5	0	0,19	0,484	0	Da	1,1549
2	1	0,767	0,806	1	Nu	1,1576
12	0,5	0,379	0,806	1	Da	1,3015
4	0	0,284	1	0	Da	1,3334
7	0,5	0	0	1	Da	1,3420
6	0	0,043	0,161	1	Nu	1,5637
14	0	0,336	0,839	1	Nu	1,5854

Dacă se aplică algoritmul NN , avem nevoie doar de instanța cea mai apropiată de x_q , în acest caz x_9 . Algoritmul nu necesită sortarea datelor după distanță, ci doar găsirea instanței cu distanță minimă. Rezultatul clasificării cu NN va fi clasa instanței x_9 , adică *Da*.

Dacă se aplică algoritmul kNN , putem considera cei mai apropiati (de exemplu 3) vecini, sau pe toți. Tabelul următor indică rezultatele clasificării după numărul de voturi ai vecinilor.

Numărul de vecini	Numărul de voturi	Decizia de clasificare
$k = 1$	1 Da – 0 Nu	Da
$k = 2$	1 Da – 1 Nu	Indecis
$k = 3$	2 Da – 1 Nu	Da
$k = 4$	3 Da – 1 Nu	Da
...
$k = 14$	9 Da – 5 Nu	Da

De asemenea, putem pondera influența vecinilor conform ecuației 9.3. Tabelul 9.5 prezintă ponderile instanțelor, ca inverse ale pătratelor distanței euclidiene.

Tabelul 9.5. Ponderarea instanțelor

Nr. instanță	Joc	Distanță	Pondere
1	Nu	0,5113	3,8254
2	Nu	1,1576	0,7463
3	Da	0,7019	2,0300
4	Da	1,3334	0,5624
5	Da	1,1549	0,7497
6	Nu	1,5637	0,4090
7	Da	1,3420	0,5553
8	Nu	0,8113	1,5194
9	Da	0,4705	4,5171
10	Da	1,0485	0,9096
11	Da	1,0183	0,9643
12	Da	1,3015	0,5903
13	Da	0,5196	3,7035
14	Nu	1,5854	0,3979

Se calculează suma ponderilor pentru instanțele care au clasa *Da*: $S_D = 14,5822$ și suma ponderilor pentru instanțele care au clasa *Nu*: $S_N = 6,898$. Prin urmare, instanța de interogare este clasificată în clasa *Da*.

9.10. Concluzii

Clasificatorii bazați pe instanțe sunt considerați *pasivi* (engl. “lazy learners”), deoarece modelul nu este construit explicit iar efortul de calcul se face la aplicarea modelului, pentru clasificarea instanțelor noi. În schimb, arborii de decizie de exemplu sunt considerați clasificatori *activi* (engl. “eager learners”), deoarece efortul de calcul se face la crearea modelului, înaintea clasificării efective a unei instanțe noi. Clasificatorii pasivi necesită mai puțin timp de calcul pentru antrenare și mai mult pentru predicție.

Ratele de eroare ale algoritmilor *NN* și *kNN* sunt de obicei mici. Dacă nu există zgomot în date, rata de eroare pe mulțimea de antrenare este în general 0. Întrucât folosesc toate informațiile disponibile, și capacitatea de generalizare este de multe ori foarte bună.

Determinarea celui mai apropiat vecin are o complexitate de timp liniară în numărul de instanțe. Pentru mulțimi de antrenare foarte mari, timpul poate fi redus prin utilizarea, de exemplu, a arborilor *k*-dimensionali (engl. “kd-trees”, Bentley, 1975), unde *k* semnifică aici numărul de atrbute (sau dimensiuni). Astfel, complexitatea poate fi redusă la un nivel logaritmic. Cu toate acestea, când numărul de atrbute este foarte mare, comparabil cu numărul de instanțe, performanțele căutării cu arbori *kd* devin apropiate căutării exhaustive. Pentru ca această metodă să fie eficientă, trebuie îndeplinită condiția $n \gg 2^k$, unde *n* este numărul de instanțe iar *k* este numărul de atrbute.

Referințe

- [1] Aaronson, S. (2006). *Quantum Computing Since Democritus*, <http://www.scottaaronson.com/democritus/default.html>
- [2] Amir, E. & Richards, M. (2008). *Variable Elimination in Bayesian Networks*, <http://www.cs.uiuc.edu/class/sp08/cs440/notes/varElimLec.pdf>
- [3] Ardis, P. (2010). *Notes on Dempster Shafer Belief Functions*, www.cs.rochester.edu/~ardis/DempsterShafer.pdf
- [4] Bao, J. (2002). *Artificial Intelligence Exercises*, http://www.cs.iastate.edu/~baojie/acad/teach/ai/hw4/2002-12-06_hw4sol.htm
- [5] Bayes, T. (1763). *An essay towards solving a problem in the doctrine of chances*, Philosophical Transactions of the Royal Society of London, vol. 53, pp. 370-418
- [6] Bentley, J. L. (1975). *Multidimensional binary search trees used for associative searching*, Communications of the ACM, vol. 18, no. 9, pp. 509-517
- [7] BlackLight Power Inc. (2005). *Double Slit. Explanation of Classical Electron Diffraction*, <http://www.blacklightpower.com/theory-2/theory/double-slit/>
- [8] Breiman, L. (1996). *Some Properties of Splitting Criteria*, <http://fbf.cba.ua.edu/~mhardin/BreimanMachineLearning1996.pdf>

- [9] Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984). *Classification and regression trees*, Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, California
- [10] Changing Minds (2012). *Types of data*, http://changingminds.org/explanations/research/measurement/types_data.htm
- [11] Cheng, J. & Druzdzel, M. J. (2000). *AIS-BN: An Adaptive Importance Sampling Algorithm for Evidential Reasoning in Large Bayesian Networks*, Journal of Artificial Intelligence Research, vol. 13, pp. 155-188
- [12] Cheng, J. & Druzdzel, M. J. (2000). *Latin hypercube sampling in Bayesian networks*, in Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS-2000), Jim Etheredge & Bill Manaris (eds), pp. 287-292, Menlo Park, California, AAAI Press
- [13] Cheng, J. (2001). *Efficient stochastic sampling algorithms for Bayesian networks*, Ph.D. Dissertation, School of Information Sciences, University of Pittsburgh, http://www2.sis.pitt.edu/~jcheng/Cheng_dissertation.pdf
- [14] Dempster, A. P. (1968). *A generalization of Bayesian inference*, Journal of the Royal Statistical Society, Series B, vol. 30, pp. 205-247
- [15] Doherty, M., *Learning: Introduction and Overview*, http://www1.pacific.edu/~mdoherty/comp151/lectures/learning_intro.ppt
- [16] Freund, Y. & Schapire, R. E. (1995). *A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting*, Proceedings of the Second European Conference on

- Computational Learning Theory, pp. 23-37, Springer-Verlag
London, UK
- [17] Fung, R. & Chang, K. (1990). *Weighting and integrating evidence for stochastic simulation in bayesian networks*, in M. Henrion, R.D. Shachter, L. N. Kanal, and J. F. Lemmer (eds.), Uncertainty in Artificial Intelligence, vol. 5, pp. 209-219, North Holland, Amsterdam
- [18] Hájek, A. (2012). *Interpretations of Probability, The Stanford Encyclopedia of Philosophy (Summer 2012 Edition)*, Edward N. Zalta (ed.), <http://plato.stanford.edu/archives/sum2012/entries/probability-interpret/>
- [19] Han, D., Han. C. & Yang, Y. (2008). *A Modified Evidence Combination Approach Based on Ambiguity Measure*, Proceedings of the 11th International Conference on Information Fusion, pp. 1-6
- [20] Harrison D. M. (2008). *Mach-Zehnder Interferometer*, <http://www.upscale.utoronto.ca/GeneralInterest/Harrison/MachZehnder/MachZehnder.html>
- [21] Hsu, W. H. (2001). *Decision Trees, Occam's Razor, and Overfitting*, <http://www.kddresearch.org/Courses/Fall-2001/CIS732/Lectures/Lecture-05-20010906.pdf>
- [22] Hunt, E.B. (1962). *Concept learning: An information processing problem*, Wiley, NewYork
- [23] Kahn, A. B. (1962). *Topological sorting of large networks*, Communications of the ACM, vol. 5, no. 11, pp. 558-562
- [24] Keynes, J. M. (1921). *A Treatise on Probability* (Chapter IV. The Principle of Indifference), Macmillan and Co.

- [25] Kira, K. & Rendell, L. A. (1992). *A Practical Approach to Feature Selection*, Proceedings of the Ninth International Workshop on Machine Learning, pp. 249-256
- [26] Kononenko, I. (1994). *Estimation Attributes: Analysis and Extensions of RELIEF*, European Conference on Machine Learning, Catania, Italy, Springer-Verlag
- [27] Kubalik, J. (2000). *Machine Learning*, <http://cyber.felk.cvut.cz/gerstner/HUT2000/ml/ml1.ppt>
- [28] Lawrence, A. E. (1997). *Microprocessor Unit (μ PU): Glossary*, <http://www-staff.lboro.ac.uk/~coael/gloss.html>
- [29] Lee, P. M. (1989). *Bayesian Statistics: an introduction*, Oxford University Press.
- [30] Livescu, K. (2003). *A Practical Introduction to Graphical Models and their use in ASR*, http://people.csail.mit.edu/klivescu/6.345/6.345-spring03_v4.ppt
- [31] MacQueen, J. B. (1967). *Some Methods for classification and Analysis of Multivariate Observations*, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, pp. 281-297
- [32] Miclea, M. (1999). *Psihologie cognitivă*, Polirom, Iași
- [33] Mitchell, T. M. (1997). *Machine Learning*, McGraw-Hill Science/Engineering/Math
- [34] Moore, A. (2001). *Probabilistic and Bayesian Analytics*, <http://www.autonlab.org/tutorials/prob17.pdf>
- [35] Nielsen, M. A. & Chuang, I. L. (2010). *Quantum Computation and Quantum Information*, 10th Anniversary Edition, Cambridge University Press

- [36] Nilsson, N. J. (2001). *Introduction to Machine Learning*,
<http://robotics.stanford.edu/people/nilsson/mlbook.html>
- [37] Ooi, Y. H. (2004). *Simpson's Paradox – A Survey of Past, Present and Future Research*, University of Pennsylvania,
<http://repository.upenn.edu/cgi/viewcontent.cgi?article=1014>
- [38] Paskin, M. (2003). *A Short Course on Graphical Models. Structured Representations*, <http://ai.stanford.edu/~paskin/gm-short-course/lec2.pdf>
- [39] PEAR (2010). *Princeton Engineering Anomalies Research. Scientific Study of Consciousness-Related Physical Phenomena*,
<http://www.princeton.edu/~pear/>
- [40] Quinlan, J. R. (1986). *Induction of Decision Trees*, Machine Learning, vol. 1, pp. 81-106
- [41] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufman Publishers
- [42] Rogers, T. (2001). *Simpsons's Paradox - When Big Data Sets Go Bad*, in Amazing Applications of Probability and Statistics,
<http://www.intuit.com/statistics/SimpsonsParadox.html>
- [43] Russell, S. J. & Norvig, P. (2002). *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2nd Edition
- [44] Seo, Y. W. & Sycara, K. (2006). *Combining multiple hypotheses for identifying human activities*, Technical report CMU-RI-TR-06-31, Robotics Institute, Carnegie Mellon University
- [45] Shachter, R. D. & Peot, M. (1990). *Simulation approaches to general probabilistic inference on belief networks*, in M. Henrion, R. D. Shachter, L. N. Kanal, and J. F. Lemmer (eds.), *Uncertainty in Artificial Intelligence*, vol. 5, pp. 221-231, North Holland, Amsterdam

- [46] Shachter, R. D. (1998). *Bayes-Ball: The Rational Pastime (for Determining Irrelevance and Requisite Information in Belief Networks and Influence Diagrams)*, Proceedings of the Fourteenth Conference in Uncertainty in Artificial Intelligence, pp. 480-487
- [47] Shafer, G. (1976). *A Mathematical Theory of Evidence*, Princeton University Press
- [48] Shannon, C. E. (1948). *A Mathematical Theory of Communication*, Bell System Technical Journal, vol. 27, pp. 379-423 (July), pp. 623-656 (October)
- [49] Simon, H. A. (1983). *Reason in Human Affairs*, Stanford University Press
- [50] Simpson, E. H. (1951). *The Interpretation of Interaction in Contingency Tables*, Journal of Royal Statistical Society Ser.B, vol. 13, no. 2, pp. 238-241
- [51] Smets, P. & Kennes, R. (1994). *The Transferable Belief Model*, Artificial Intelligence, vol. 66, pp. 191-243
- [52] Tan, P. N., Steinbach, M. & Kumar, V. (2006). *Introduction to Data Mining*, Addison-Wesley
- [53] Vucetic, S. (2004). *Alternative Classification Algorithms*, <http://www.ist.temple.edu/~vucetic/cis526fall2004/> lecture8.ppt
- [54] Witten, I. H. & Frank, E. (2000). *Data Mining: Practical machine learning tools with Java implementations*, Morgan Kaufmann, San Francisco
- [55] Yager, R. (1987). *On the Dempster-Shafer Framework and New Combination Rules*, Information Sciences, vol. 41, pp. 93-137

- [56] Yuan, C. & Druzdzel, M. J. (2003). *An importance sampling algorithm based on evidence pre-propagation*, in Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence (UAI-03), pp. 624-631, Morgan Kaufmann Publishers San Francisco, California
- [57] Zadeh, L. (1979). *On the validity of Dempster's rule of combination*, Memo M79/24, University of California, Berkeley, USA