

LeMAS 2016

Bucuresti 21-25 iunie 2016

Arbori de decizie

Adina Magda Florea

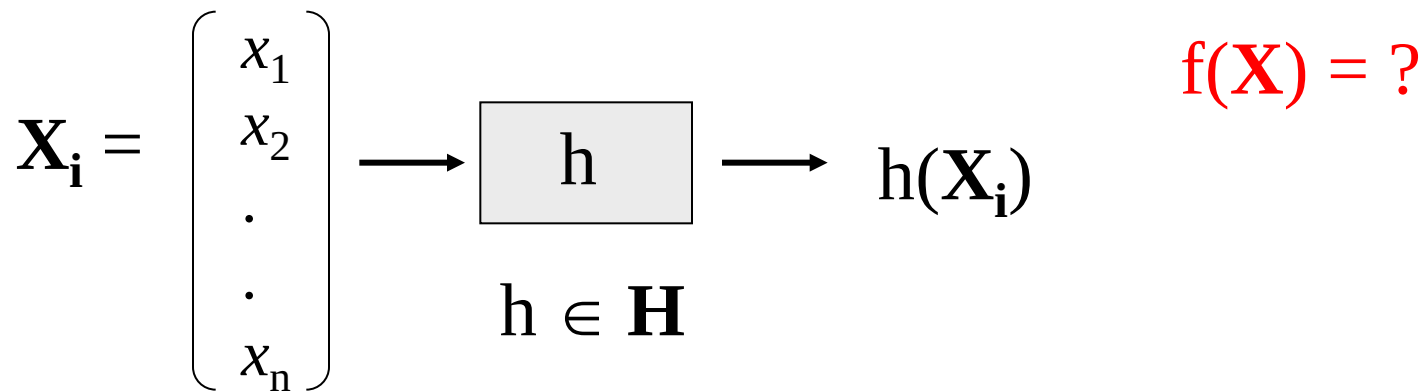


Învățare supervizată

- Clasificare prin metoda arborilor de decizie

Tipuri de învățare

$T = \{X_1, X_2, \dots, X_m\}$ – **multime de invatare**



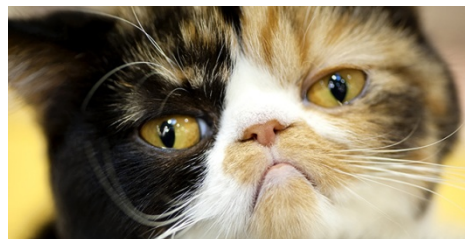
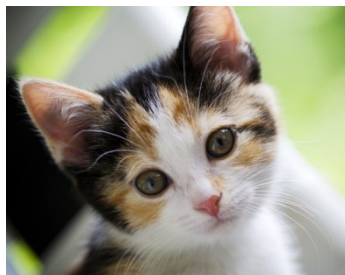
■ **Invatare supervizata** – se cunosc $f(X_1), \dots, f(X_m)$

Gasim **h** a.i. $h(X_i) = f(X_i), i=1, m \rightarrow h(X_i) = f(X_i), \forall i$

Clasificare – **f** - valori discrete – grupeaza exemple

Regresie – **f** – valori continue, estimeaza sau prezice valori

Clasificare vs regresie



■ Caine / Pisica

- Dan 10.000 lei
- George 3.000 lei
- Vlad 15.000 lei
- Maria 5.000 lei

credit de 30.000 lei
credit de 10.000 lei
credit de 50.000 lei
cat credit?

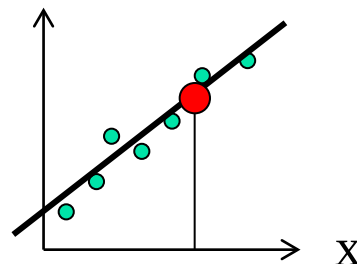
Clasificare vs regresie

- Reprezentam ipotezele de invatare sub o forma care sa permita ca iesirea algoritmului sa fie o eticheta (clasa): arbori, retele neurale, etc.

- Reprezentam ipoteza de invatare ca o functie liniara

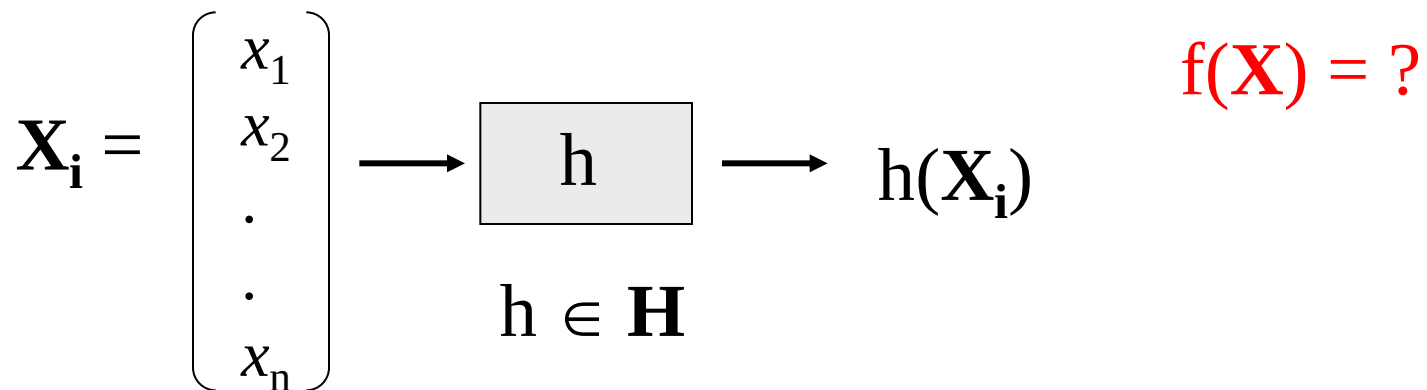
$$h(x) = a_0 + a_1 x$$

- Cat este valoarea y pt punctul rosu?



Tipuri de învățare

$T = \{X_1, X_2, \dots, X_m\}$ – **multime de invatare**

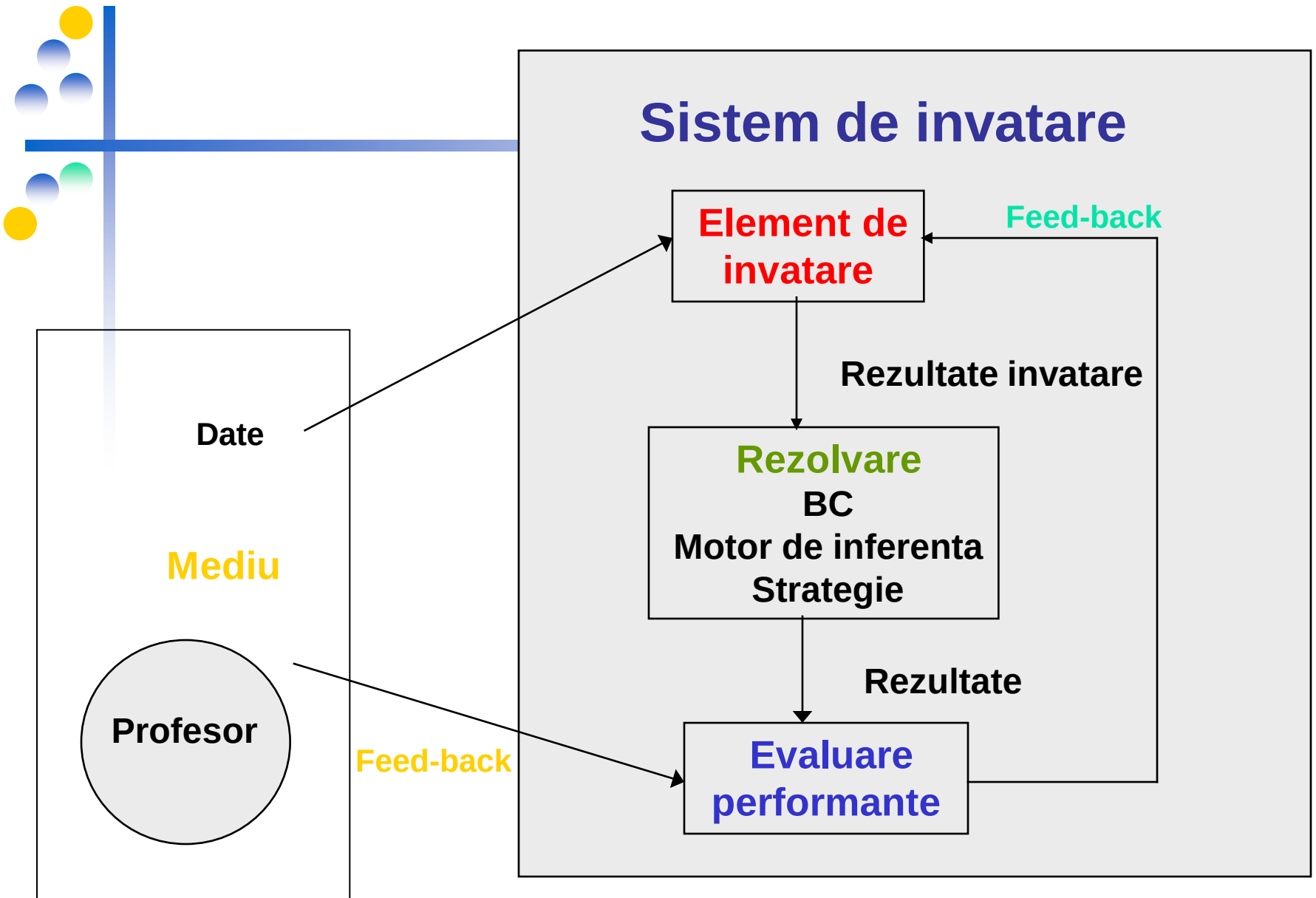


- **Invatare ne-supervizata** – NU se cunosc $f(X_1), \dots, f(X_m)$

Imparte T in submultimi – clase

Se poate vedea tot ca invatarea unei functii – val f = numele submultimii careia ii apartine X_i

- **Invatare prin recompensa** – Se cunosc recompensele pentru valorile $h(X_1), \dots, h(X_m)$, Nu se cunosc valorile lui f



Occam Razor

Principiul lamei lui Occam (*lex parsimoniae*)

- prefer explicatiile simple celor complexe
- selectez ipoteza care implica cele mai putine presupuneri, intre ipoteze similare/egal probabile
- Wiliam of Occam, 1285 – 1347
- filozof englez

"non sunt multiplicanda entia praeter necessitatem"



Conditii pentru o invatare "corecta"

Problema: identifica personaje de film
"bune" sau "rele" dupa modul in care arata

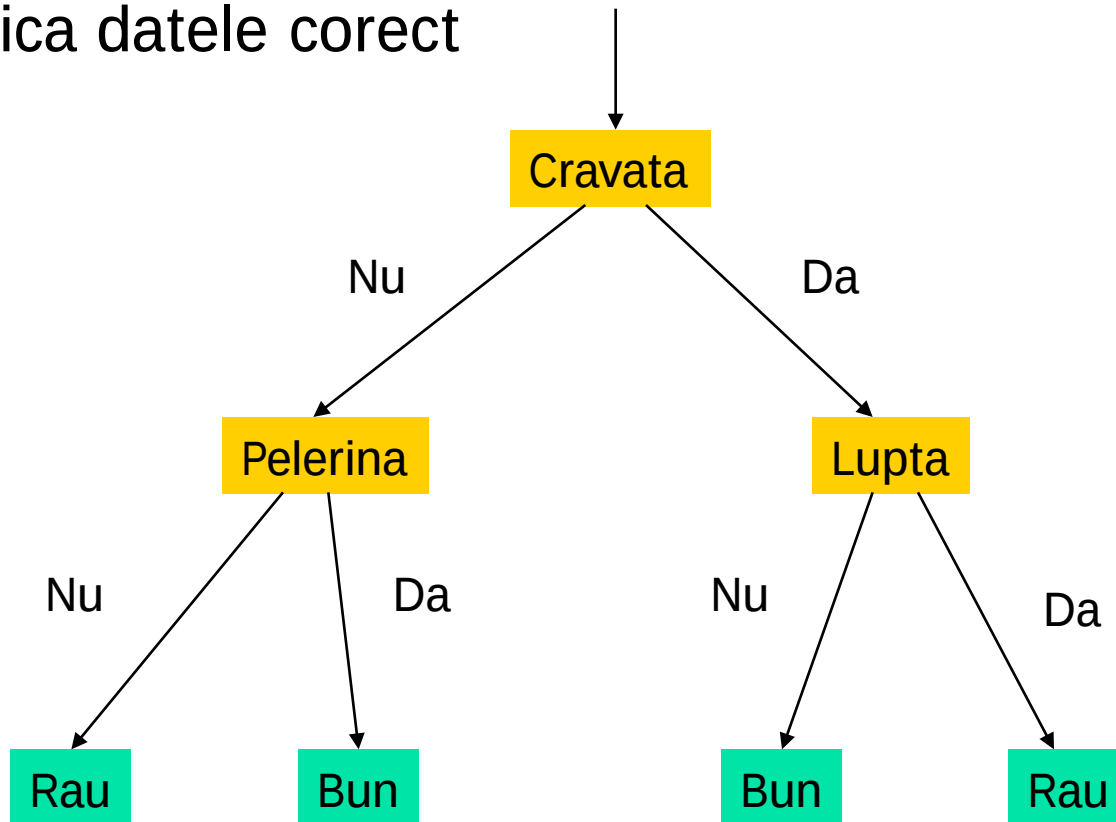


Conditii pentru o invatare "corecta"

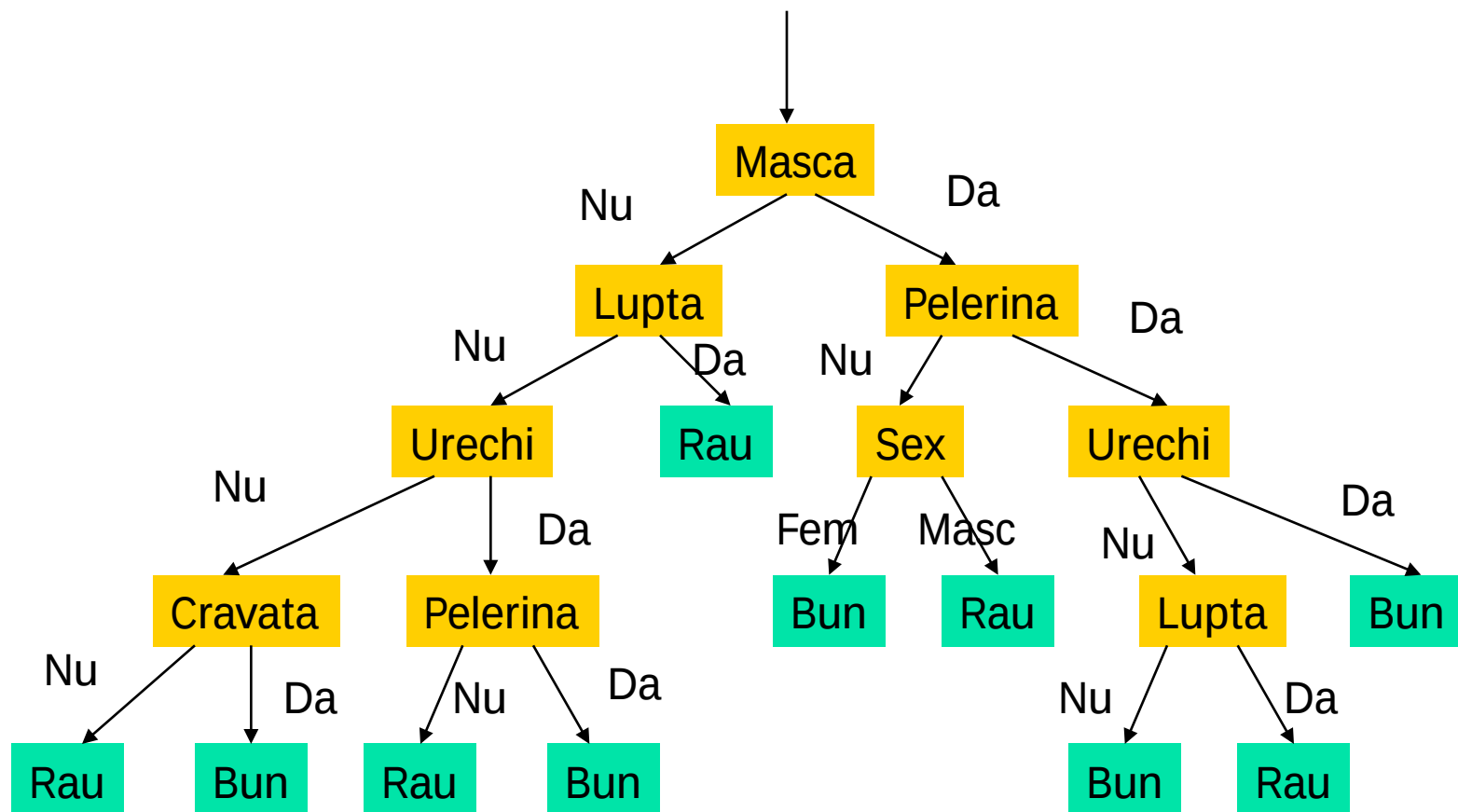
Atribute / Instante	Sex	Masca	Pelerina	Cravata	Urechi	Lupta	Clasa
	Set de invatare						
Batman	Masc	Da	Da	Nu	Da	Nu	Bun
Robin	Masc	Da	Da	Nu	Nu	Nu	Bun
Alfred	Masc	Nu	Nu	Da	Nu	Nu	Bun
Penguin	Masc	Nu	Nu	Da	Nu	Da	Rau
Catwoman	Fem	Da	Nu	Nu	Da	Nu	Rau
Joker	Masc	Nu	Nu	Nu	Nu	Nu	Rau
	Date de test						
Batgirl	Fem	Da	Da	Nu	Da	Nu	??
Fred	Masc	Da	Nu	Nu	Nu	Nu	??

Conditii pentru o invatare "corecta"

Clasifica datele corect



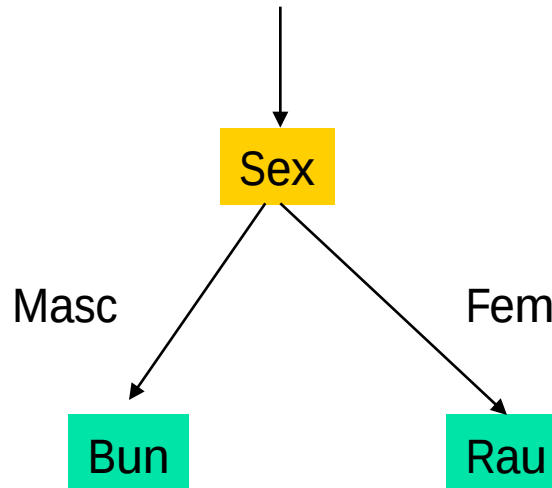
Conditii pentru o invatare "corecta"



Clasifica datele corect dar complexitate prea mare (intuitiv)

Conditii pentru o invatare "corecta"

Prea simplu, nu clasifica corect



Aleg prima varianta (cf. lamei lui Occam)



Conditii pentru o invatare "corecta"

- Clasificatoarele trebuie sa fie suficient de "expresive" pentru a fi in concordanta cu setul de invatare
- Dar clasificatoarele care au o complexitate prea mare pot duce la fenomenul de "overfit" (overfitting) = ipoteza formata/gasita include zgomot sau sabloane de date nerelevante



Invatarea inductiva prin AD

- Reprezentarea = **arbori de decizie** (AD)
- Problema de invatare = **clasificare**
- **Invatare supervizata**
- ID3 – in jur de 1960s
- C4.5 (Quinlan)
- C4.5 – unul din cei mai cunoscuti si mai utilizati algoritmi de invatare
- Ultima versiune de cercetare: C4.8, implementata in Weka
- Versiunea comerciala: C5.0



ID3 (Quinlan)

- Univers de obiecte U descrise in termenii unei colectii de attribute $\{A\}$
- Fiecare **atribut** masoara o caracteristica importanta a unui obiect $o \in U$; domeniul de valori attribute $D_A =$ discret, simbolic (ulterior extins)
- Fiecare obiect apartine unui **clase** dintr-o multime de clase mutual exclusive
- Se da **setul de invatare** (SI) – obiectele = exemple de învățare
- **Problema** = obtinerea unor **reguli de clasificare** / construirea unui **AD** care clasifica corect nu numai $\forall o \in SI$ dar si $\forall o \in U$



ID3 (Quinlan)

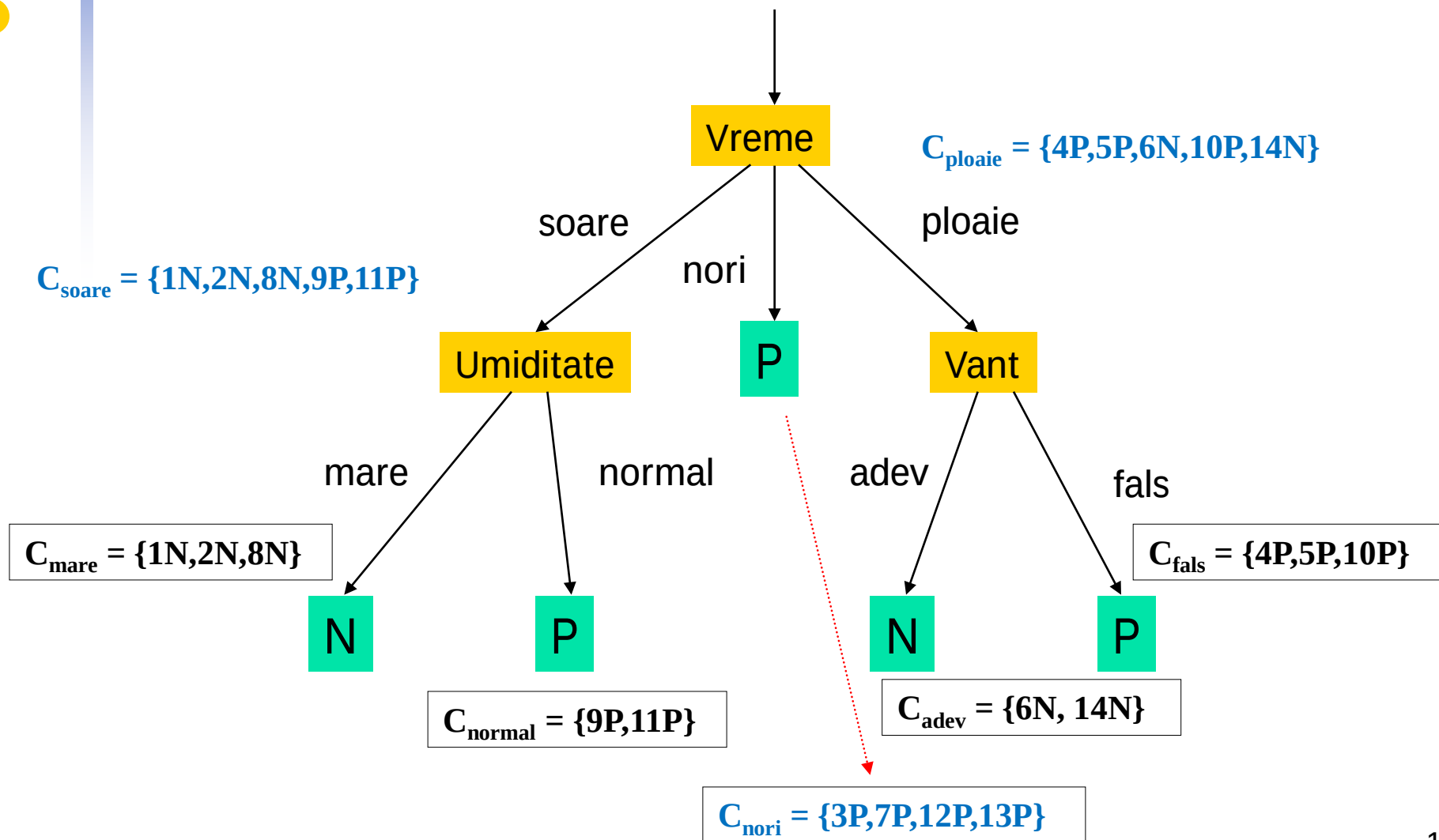
Metoda de constructie

- C = mulțimea de exemple de învățare din SI
- A_j – un atribut test (din mulțimea de attribute $A_1..A_k$) cu valori $a_1^j, .. a_{mk}^j$
- Se împart exemplele din C în funcție de valorile unui atribut A_j
- $[C_1, .., C_{mk}]$, cu $C_i = \{o \in C \mid A_j = a_i^j\}$
- Impărțirea/expandarea AD se oprește când toate C_i aparțin unei aceleiași clase

ID3 – Exemplul 1

No.	Atribute				Clasa
	Vreme	Temperatura	Umiditate	Vant	
1	soare	cald	mare	fals	N
2	soare	cald	mare	adev	N
3	nori	cald	mare	fals	P
4	ploaie	placut	mare	fals	P
5	ploaie	racoare	normal	fals	P
6	ploaie	racoare	normal	adev	N
7	nori	racoare	normal	adev	P
8	soare	placut	mare	fals	N
9	soare	racoare	normal	fals	P
10	ploaie	placut	normal	fals	P
11	soare	placut	normal	adev	P
12	nori	placut	mare	adev	P
13	nori	cald	normal	fals	P
14	ploaie	placut	mare	adev	N

ID3 – Exemplul 1





ID3 – Arbore minim

- Din același SI se pot contrui diferiți AD
 - Cum se poate obține cel mai mic arbore (lama lui Occam) ?
- = Cum selectez atributul din rădăcina unui arbore?



ID3 – Cum selectez A?

- Mulțimea de exemple de învățare C cu $p \in P$ și $n \in N$

Se presupune ca:

- Orice AD corect va clasifica obiectele proportional cu reprezentarea lor in C

Un obiect arbitrar $o \in C$ va fi clasificat:

- $\in P$ cu probabilitatea $p/(p+n)$
- $\in N$ cu probabilitatea $n/(p+n)$
- **Entropia** – măsoară **impuritatea** într-o mulțime de exemple

$$- \sum_{i=1}^n p_i \log p_i$$



Selectia testului (atributului)

- C cu $p \in P$ si $n \in N$
- **Conținutul informațional al unui arbore** $I(AD_{p,n})$

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

- Selecteaza A in radacina; $A \in \{a_1, \dots, a_m\}$
- Fie C_i cu $p_i \in P$ si $n_i \in N$, $i=1, m$
- Continutul de informatie pentru C_i este
$$I(AD_{p_i, n_i}), i=1, v$$



Selectia testului (atributului)

- Dupa selectarea lui A în radacina, **cantitatea de informație** necesară pentru a termina construcția arborelui este suma ponderată a conținutului de informație din toți subarborii

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

- unde ponderea ramurii i este fracțiunea de obiecte din C care aparțin lui C_i ;
- m este numărul de valori ale lui A



Selectia testului (atributului)

- **Câștigul informațional** al unui atribut A obținut prin selecția acestuia ca rădăcină a arborelui de decizie este:

$$G(A) = I(AD_{p,n}) - E(A)$$

- Se selectează A cu câștig informațional maxim
- Recursiv pentru a forma AD corespunzatori $C_1 \dots C_m$

Calcul G(A) pt Ex 1

- 14 exemple, $9 \in P$, $5 \in N$

- $I(AD_{p,n}) = 0.940$ bits

$$-\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14}$$

- vreme

- soare - $2 \in P$, $3 \in N \Rightarrow I(AD_{p1,n1}) = 0.971$

- nori - $4 \in P$, $0 \in N \Rightarrow I(AD_{p2,n2}) = ?$

0

- ploaie - $3 \in P$, $2 \in N \Rightarrow I(AD_{p3,n3}) = ?$

0.971

- $E(vreme) = 0.694$ bits

$$\frac{5}{14} I(AD_{p1,n1}) + \frac{4}{14} I(AD_{p2,n2}) + \frac{5}{14} I(AD_{p3,n3})$$

- $G(vreme) = 0.940 - 0.694 = 0.246$ bits

- $G(temperatura) = 0.029$ bits

$$C_{\text{soare}} = \{1N, 2N, 8N, 9P, 11P\}$$

- $G(umiditate) = 0.151$

$$C_{\text{nori}} = \{3P, 7P, 12P, 13P\}$$

- $G(vant) = 0.048$ bits

$$C_{\text{ploaie}} = \{4P, 5P, 6N, 10P, 14N\}$$



Generalizare la mai multe clase

- **Continutul de informatie**

$$I(Arb) = \sum_{i=1}^v - p(Cl = C_i) * \log_2 p(Cl = C_i)$$

- **Cantitatea de informatie necesara pentru a termina constructia arborelui**

$$E(A) = \sum_{i=1}^v \frac{|C_i|}{|C|} I(C_i)$$

- **Castigul informational**

$$G(A) = I(Arb) - E(A)$$

Algoritm ID3

functie **ind-arbore** (set-invatare, atribute, default)

1. **daca** set-invatare = vid **atunci intoarce** frunza etichetata cu *default* sau "Failure" Caz 1 – ex inv lipsa
2. **daca** toate exemplele din set-invatare sunt in aceeaasi clasa **atunci intoarce** o frunza etichetata cu acea clasa Bine = recunoaste
3. **daca** atribute este vida **atunci intoarce** o frunza etichetata cu disjunctia tuturor claselor din set-invatare Caz 2 – atr inadecvate
4. selecteaza un atribut A, creaza nod pt A si eticheteaza nodul cu A
5. sterge A din atribute → atribute1
6. **m** = cea mai frecventa clasa (set-invatare)
7. **pentru** fiecare valoare V a lui A **repeta**
 - fie partitie_V multimea exemplurilor din set-invatare, cu valoarea V pentru A
 - creaza $\text{nod}_V = \text{ind-arbore}(\text{partitie}_V, \text{atribute1}, m)$
 - creaza legatura nod A – nod_V etichetata cu V

sfarsit