Inteligență Artificială

Universitatea Politehnica Bucuresti
Anul universitar 2021-2022

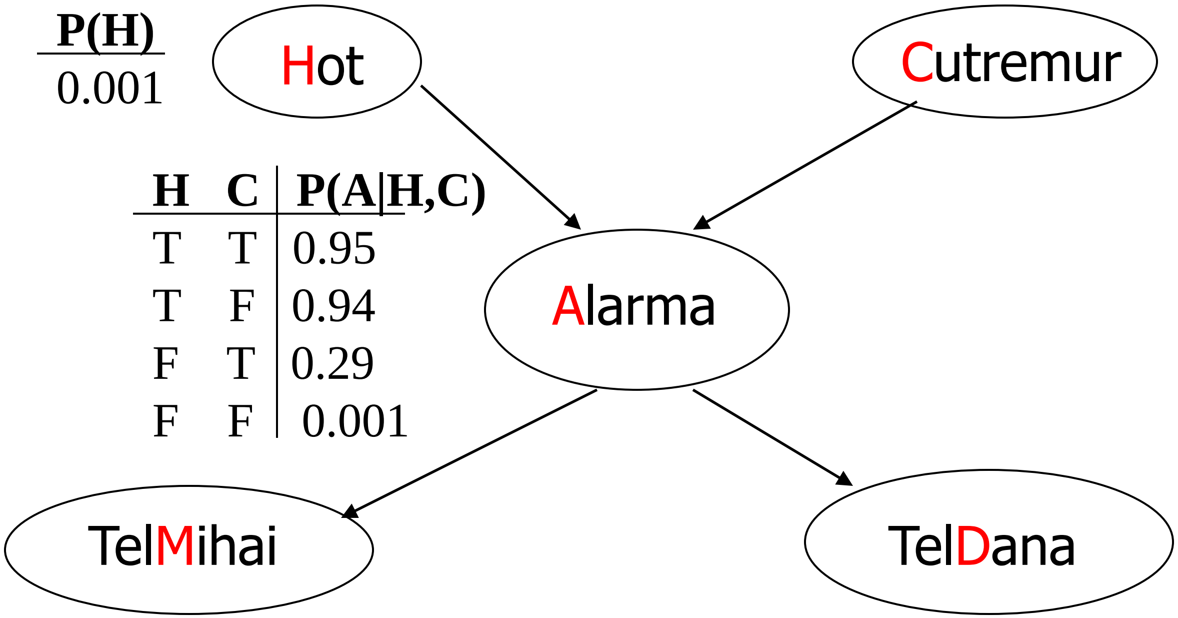
Adina Magda Florea

Curs nr. 5 si 6

Reprezentarea cunostintelor incerte

- Teoria probabilitatilor
- Retele Bayesiene
- Inferente exacte si aproximative in retele Bayesiene

Inferente probabilistice



P(H)
0.001

P(C)
0.002

H	C	P(A|H,C)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

A	P(M|A)
T	0.9
F	0.05

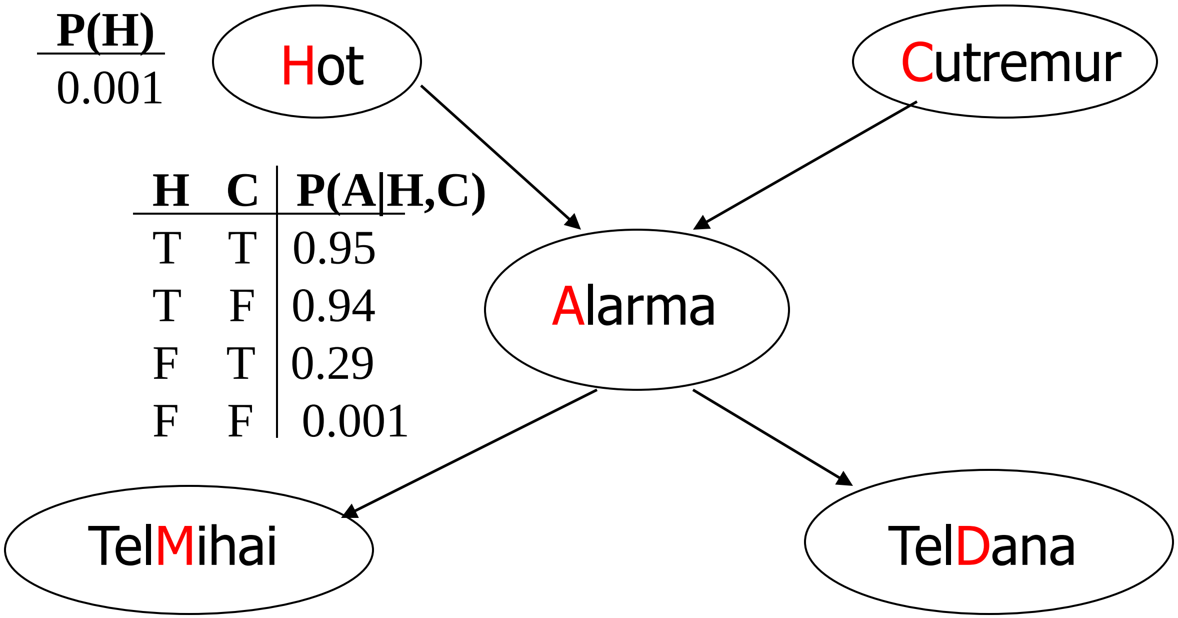
A	P(D|A)
T	0.7
F	0.01

$$P(M \wedge D \wedge A \wedge {\sim}H \wedge {\sim}C) =$$

$$P(M|A) * P(D|A) * P(A|{\sim}H \wedge {\sim}C) * P({\sim}H) \wedge P({\sim}C) =$$

$$0.9 * 0.7 * 0.001 * 0.999 * 0.998 = 0.00062$$

Inferente probabilistice



P(H)
0.001

P(C)
0.002

H	C	P(A|H,C)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

A	P(M|A)
T	0.9
F	0.05

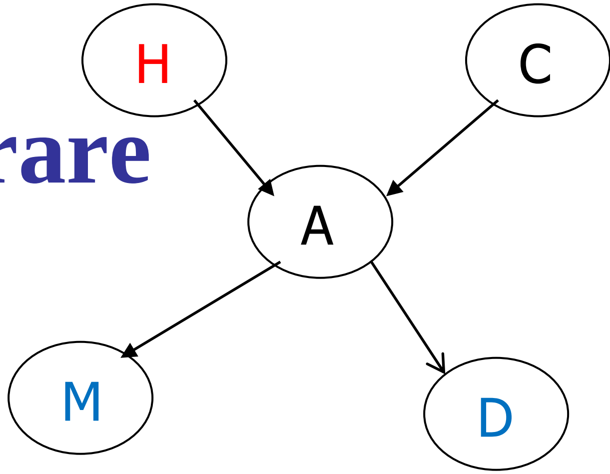
A	P(D|A)
T	0.7
F	0.01

$$P(A|H) = P(A|H,C) * P(C|H) + P(A|H,\sim C) * P(\sim C|H)$$

$$= P(A|H,C) * P(C) + P(A|H,\sim C) * P(\sim C)$$

$$= 0.95 * 0.002 + 0.94 * 0.998 = 0.94002$$

2.5 Inferență prin enumerare



X – variabila de interogare (Carie)
E – lista de variabile probe (Dur_d)
e – lista valorilor observate pt aceste variabile E
Y – lista variabilelor neobservate (restul) (Evid)

$$\mathbf{P}(X \mid e) = \alpha\, \mathbf{P}(X, e) = \alpha \Sigma_Y\, \mathbf{P}(X, e, Y)$$

$$\mathbf{P}(\mathbf{H} \mid \mathbf{M},\mathbf{D}) = \alpha\, P(H,M,D) =$$

$$\alpha\, \Sigma_C\, \Sigma_A\, P(H)P(C)P(A \mid H,C)P(M \mid A)P(D \mid A)$$

n var bool -> $O(2^n)$

Inferență prin enumerare

$\mathbf{P(H|M,D)} = \alpha \sum_C \sum_A P(H)P(C)P(A|H,C)P(M|A)P(D|A)$

$P(H|M,D) = \alpha\, P(H) \sum_C P(C) \sum_A P(A|H,C)P(M|A)P(D|A)$

$= \alpha <0.00059224,\ 0.0014919> =_{aprox} <0.284,\ 0.716>$

Complexitate spatiu – $O(n)$

Complexitate timp $O(2^n)$

Inferență prin enumerare - algoritm

algoritm Enumerare(X,e,rb) intoarce distributie X

X – var de interogare

e – valori observate pt E

rb – Retea Bayesiana cu var $\{X\} \cup E \cup Y$

1. $Q(X) \leftarrow$ o distributie X, initial vida
2. **pentru** fiecare valoare x_i a lui X **repeta**

   $Q(x_i) \leftarrow$ **EnumToate(rb.Vars, $\mathbf{e_{xi}}$)**

   unde $\mathbf{e_{xi}}$ este **e** extins cu $X=x_i$
3. **intoarce** Normalizare(Q(x))

sfarsit

Inferență prin enumerare - algoritm

algoritm EnumToate(Vars,e) intoarce un numar real

1. daca Vars = [] atunci intoarce 1.0
2. Y ← first(Vars)
3. **daca** Y are valoare ***y*** in ***e***

   atunci intoarce P(y|parinti(Y)) * **EnumToate**(Rest(Vars),***e***)

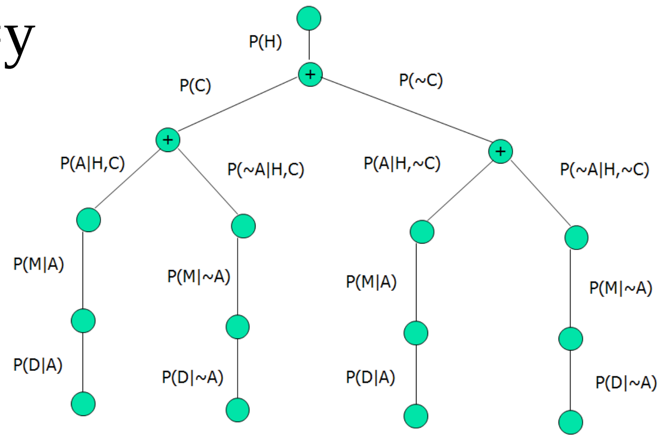
   altfel intoarce

   Σ_y P(y|parinti(Y)) * **EnumToate**(Rest(Vars),$\boldsymbol{e_y}$)

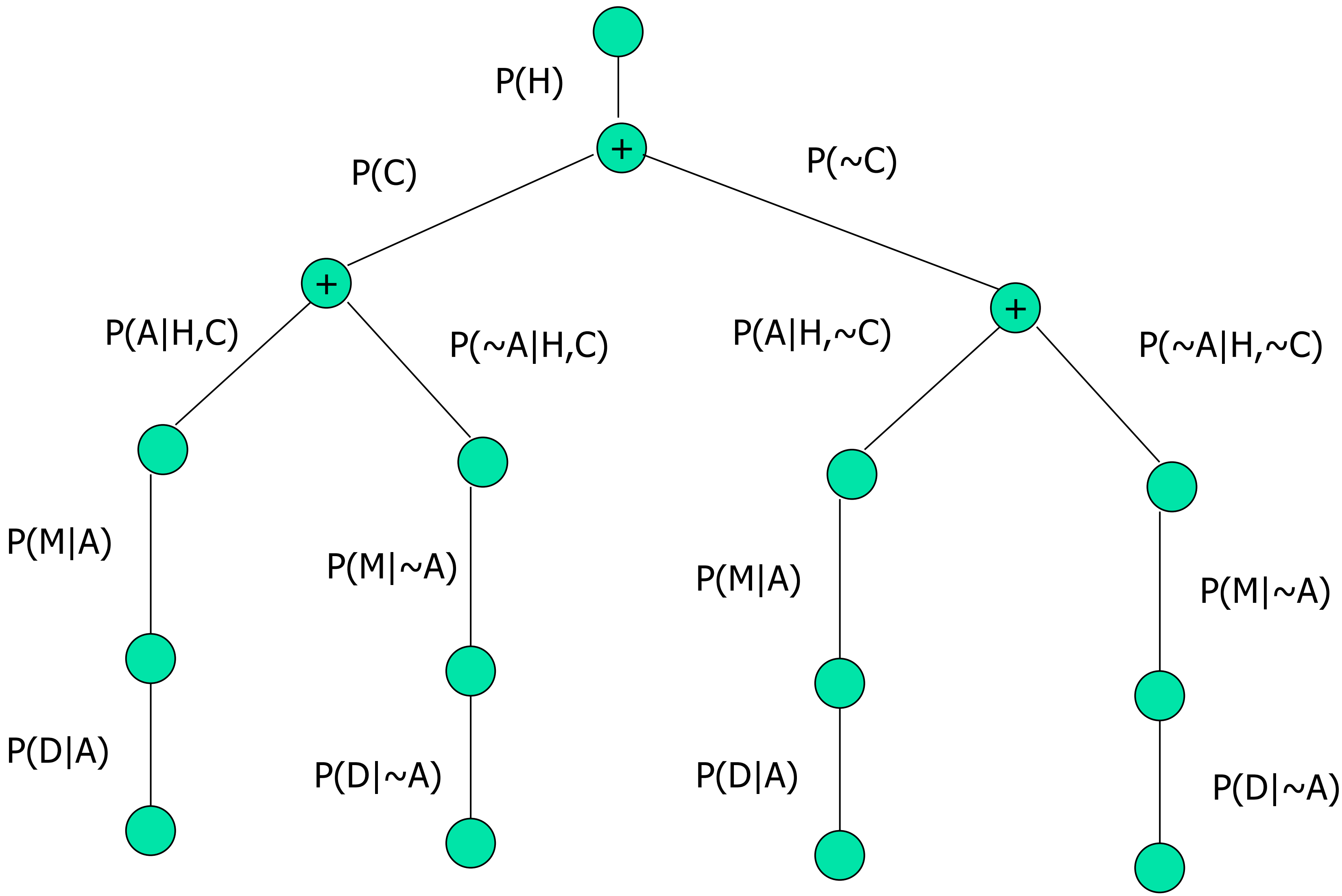
   unde $\boldsymbol{e_y}$ este ***e*** extins cu Y=y

sfarsit

P(H|M,D)= $\alpha\ \Sigma_C\ \Sigma_A$ P(H)P(C)P(A|H,C)P(M|A)P(D|A)



$$\mathbf{P}(X \mid e) = \alpha\, \mathbf{P}(X, e) = \alpha \Sigma_Y\, \mathbf{P}(X, e, Y)$$



$$\mathbf{P}(\mathbf{H}|\mathbf{M,D}) = \alpha\, \Sigma_C\, \Sigma_A\, P(H)P(C)P(A|H,C)P(M|A)P(D|A)$$

Eliminarea variabilelor (Variable elimination)

Pt a elimina calculele duplicate – se calculeaza o data si se salveaza rezultatul

Idee – se calculeaza ecuatia de la dreapta la stanga

$$P(H|M,D) = \alpha \, \Sigma_C \, \Sigma_A \, \underbrace{P(H)}_{f1(H)} \; \underbrace{P(C)}_{f2(C)} \; \underbrace{P(A|H,C)}_{f3(A,H,C)} \; \underbrace{P(M|A)}_{f4(A)} \; \underbrace{P(D|A)}_{f5(A)}$$

f5(A) = [P(D|A) P(D|~A)] = [.07 .001]

f4(A) = [P(M|A) P(M|~A)] = [.09 .05]

f3(A,H,C) va fi o matrice de 2 x 2 x 2

[P(A|H,C) P(~A|H,C) P(A|H, ~C) P(~A|H, ~C) … P(~A|~H,~C)]

Eliminarea variabilelor

$P(H|M,D) = \alpha \, \Sigma_C \, \Sigma_A \, P(H) \; P(C) \; \underline{P(A|H,C) \, P(M|A) \, P(D|A)}$

f1(H) f2(C) f3(A,H,C) f4(A) f5(A)

Insumam peste A in f3, f4, f5 – rezulta un factor f6(H,C)

f6(H,C) $= \Sigma_A$ **f3**(A,H,C) x **f4**(A) x **f5**(A) =

f3(a,H,C) x **f4**(a) x **f5**(a) + **f3**(~a,H,C) x **f4**(~a) x **f5**(~a)

Insumam peste C – rezulta un factor f7(H)

$P(H|M,D) = \alpha$ **f1**(H) x $\underline{\Sigma_C \text{ } \mathbf{f2}(C) \times \mathbf{f6}(H,C)}$

f7(H) $= \Sigma_C$ **f2**(C) x **f6**(H,C) = **f2**(c) x **f6**(H,c) + **f2**(~c) x **f6**(H,~c)

$P(H|M,D) = \alpha$ **f1**(H) x **f7**(H)

Avem nevoie de 2 operatii

- **Pointwise product** x
- **Insumarea unei variabile dintr-un produs de factori**

Implementarea operațiilor

Pointwise product

$(\mathbf{f_1} \text{ x } \mathbf{f_2})(X_1,\ldots,X_i,Y_1,\ldots,Y_j,Z_1,\ldots,Z_k) =$

$$\mathbf{f_1}(X_1,\ldots,X_i,Y_1,\ldots,Y_j) \text{ x } \mathbf{f_2}(Y_1,\ldots,Y_j,Z_1,\ldots,Z_k)$$

Insumarea unei variabile dintr-un produs de factori

- Insumarea peste **X1**, rezultatul este un factor peste $X_2,\ldots,X_i$

$$(\Sigma_{X1}\mathbf{f})(X_2,\ldots,X_i) = \mathbf{f}(X_1=v_1,\ldots,X_i)+\ldots+\mathbf{f}(X_1=v_k,\ldots,X_i)$$

Implementarea operațiilor - exemplu

Pointwise product

f1(A,B) x f2(B,C) = f3(A,B,C) are 2^{1+1+1} = 8 intrari

A	B	f1(A,B)	B	C	f2(B,C)	A	B	C	f3(A,B,C)
a	a	.3	a	a	.2	a	a	a	.3 x .2 = .06
a	f	.7	a	f	.8	a	a	f	.3 x .8 = .24
f	a	.9	f	a	.6	a	f	a	.7 x .6 = .42
f	f	.1	f	f	.4	a	f	f	.7 x .4 = .28
						f	a	a	.9 x .2 = .18
						f	a	f	.9 x .8 = .72
						f	f	a	.1 x .6 = .06
						f	f	f	.1 x .4 = .04

Implementarea operațiilor - exemplu

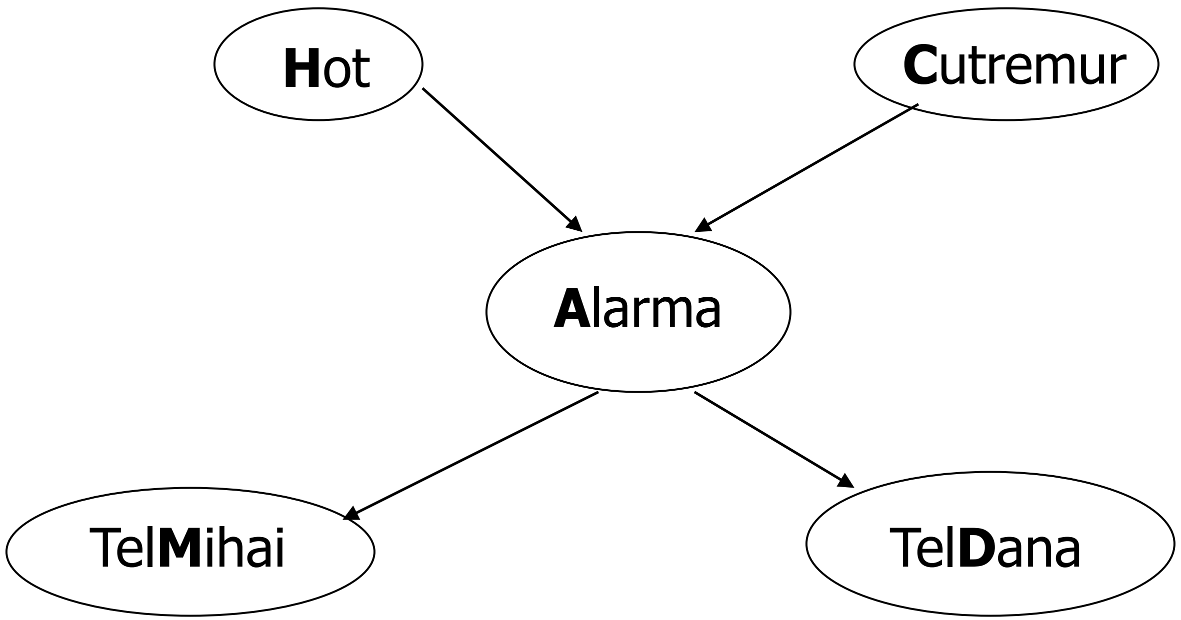
Insumarea unei variabile dintr-un produs de factori

- Insumarea peste **A** din **f3(A,B,C)**

f(B,C) = Σ_A f3(A,B,C) = f3(a,B,C) + f3(~a,B,C) =

$$\begin{bmatrix} .06 & .24 \\ .42 & .28 \end{bmatrix} + \begin{bmatrix} .18 & .72 \\ .06 & .04 \end{bmatrix} = \begin{bmatrix} .24 & .96 \\ .48 & .32 \end{bmatrix}$$

2.6 Tipuri de inferență



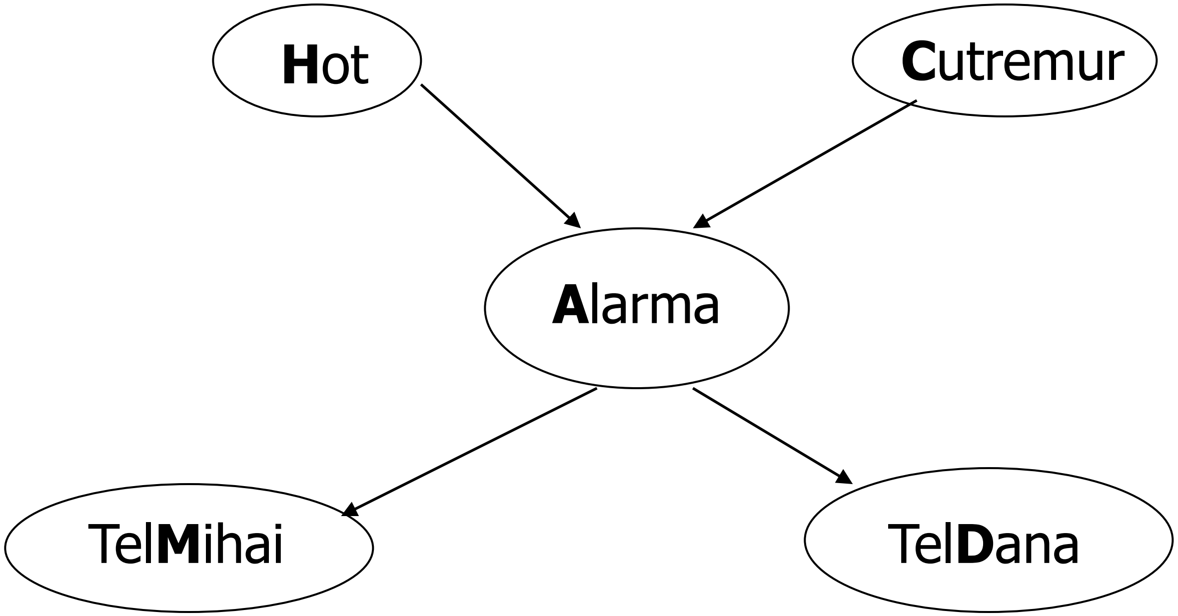
Inferente de diagnosticare (efect $\rightarrow$ cauza)

P(Hot | TelMihai)

Inferente cauzale (cauza $\rightarrow$ efect)

P(TelMihai |Hot), P(TelDana | Hot)

Tipuri de inferență



Inferente intercauzale (intre cauza si efecte comune)

$P(Hot \mid Alarma \wedge TelDana)$

Inferente mixte

$P(Alarma \mid TelMihai \wedge \sim Cutremur) \rightarrow$ diag + cauzal

$P(Hot \mid TelMihai \wedge \sim Cutremur) \rightarrow$ diag + intercauzal

2.7 Independenta in RB

- Evenimentele **X si Y sunt independente conditional fiind dat un eveniment Z** daca

  Stiind ca Z apare, aparitia lui X nu influenteaza aparitia lui Y si aparitia lui Y nu influenteaza aparitia lui X

- **Structura de graf RB codifica anumite relatii de independenta: fiecare nod este independent conditional de nondescendenti, fiind dati parintii lui**

Independenta in RB

- Mai sunt si alte relatii de independenta?
- Doua sau mai multe relatii de independenta conditionala pot conduce la o noua relatie folosind un mecanism bazat pe axiome ale grafurilor
- **D-separabilitatea** este un concept care surprinde astfel de relatii de independenta conditionala derivate
- Z **d-separa** pe X si Y intr-un DAG daca X si Y sunt independente conditional fiind dat Z dpv al axiomelor grafului.
- **$(X \perp Y \mid Z)$ – X este independent conditional de Y fiind dat Z**

D-separabilitate

D-separabilitate

- **Cand nu este indeplinita ($X \perp Y \mid Z$) ?**

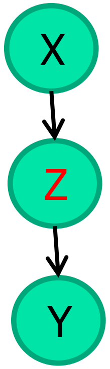
Conexiuni directe $X \rightarrow Y$

- Se pot influenta indiferent de Z

Conexiuni indirecte: X si Y nu sunt direct conectate dar exista un lant intre ele (trail)

D-separabilitate

- Fie cazul a 3 noduri

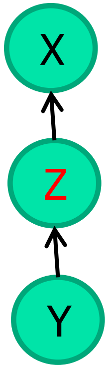


Caz a: Efect cauzal

Lantul cauzal $\mathbf{X \rightarrow Z \rightarrow Y}$

- X nu influenteaza Y via Z daca Z observat

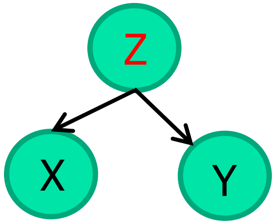
Caz b: Efect evidenta



- $\mathbf{Y \rightarrow Z \rightarrow X}$
- Identic cu cazul **a:**
- X nu influenteaza Y via Z daca Z observat
- **daca $(X \perp Y \mid Z)$ nu este indeplinit atunci nici $(Y \perp X \mid Z)$ nu este indeplinit**

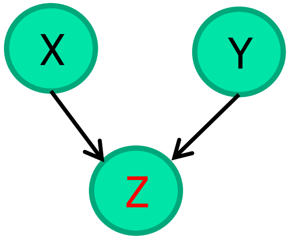
D-separabilitate

Caz c: Cauza comuna



- **X ← Z → Y**
- X nu influenteaza pe Y via Z daca <u>Z observat</u>

Case d: Efect comun



- **X → Z ← Y**
- Influenta nu poate merge de-a lungul lantului

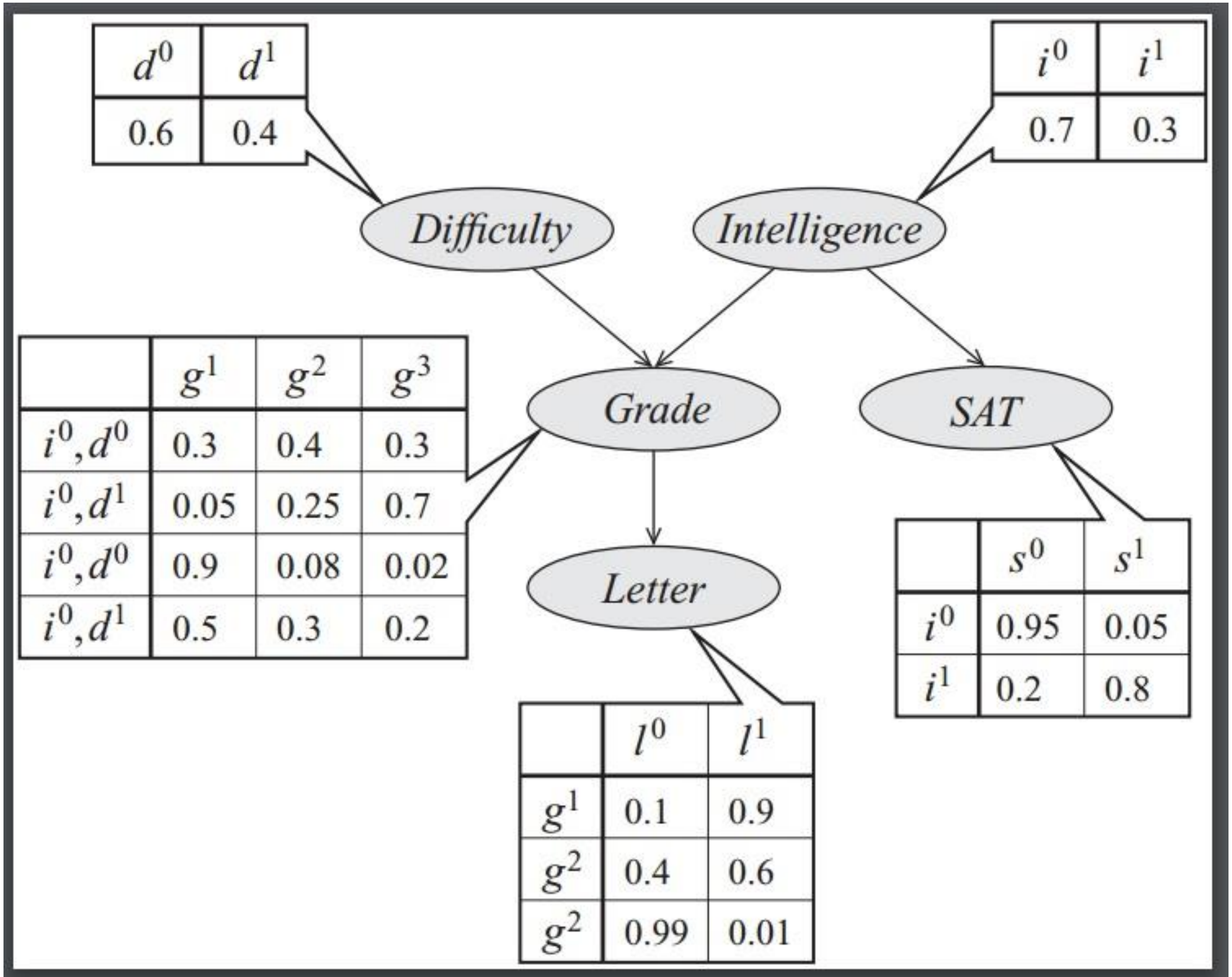
X → Z ← Y daca <u>Z nu este observat</u>

Deci X nu influenteaza pe Y via Z daca <u>Z NU este observat</u>

D-separabilitate

Daca influenta poate curge de la X la Y via Z spunem ca lantul $\mathbf{X \leftrightarrow Z \leftrightarrow Y}$ este <u>activ</u>

- Efect cauzal $\mathbf{X \rightarrow Z \rightarrow Y}$ – activ $\leftrightarrow$ Z nu este observat
- Efect evidenta $\mathbf{Y \rightarrow Z \rightarrow X}$ – activ $\leftrightarrow$ Z nu este observat
- Cauza comuna $\mathbf{X \leftarrow Z \rightarrow Y}$ – activ $\leftrightarrow$ Z nu este observat
- Efect comun $\mathbf{X \rightarrow Z \leftarrow Y}$ - activ $\leftrightarrow$ fie Z fie unul din descendentii lui Z este observat
- **Influenta probabilistica** = **flux in graf** – cand influenta din X poate curge prin Z catre Y a.i. sa afecteze probabilitatea lui Y,
- Deci X nu este conditional independent de Y fiind dat Z



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^0, d^0	0.9	0.08	0.02
i^0, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^2	0.99	0.01

2.8 Inferențe de aproximare în RB

- Inferente exacte in RB:
  - Inferenta prin enumerare
  - Eliminarea variabilelor (eficientizare)

Asa cum s-a aratat anterior, exprimam probabilitatea conditionata ca probabilitate neconditionata, apoi insumam peste variabilele neobservate

$$\mathbf{P}(X \mid e) = \alpha\, \mathbf{P}(X, e) = \alpha \Sigma_Y\, \mathbf{P}(X, e, Y)$$

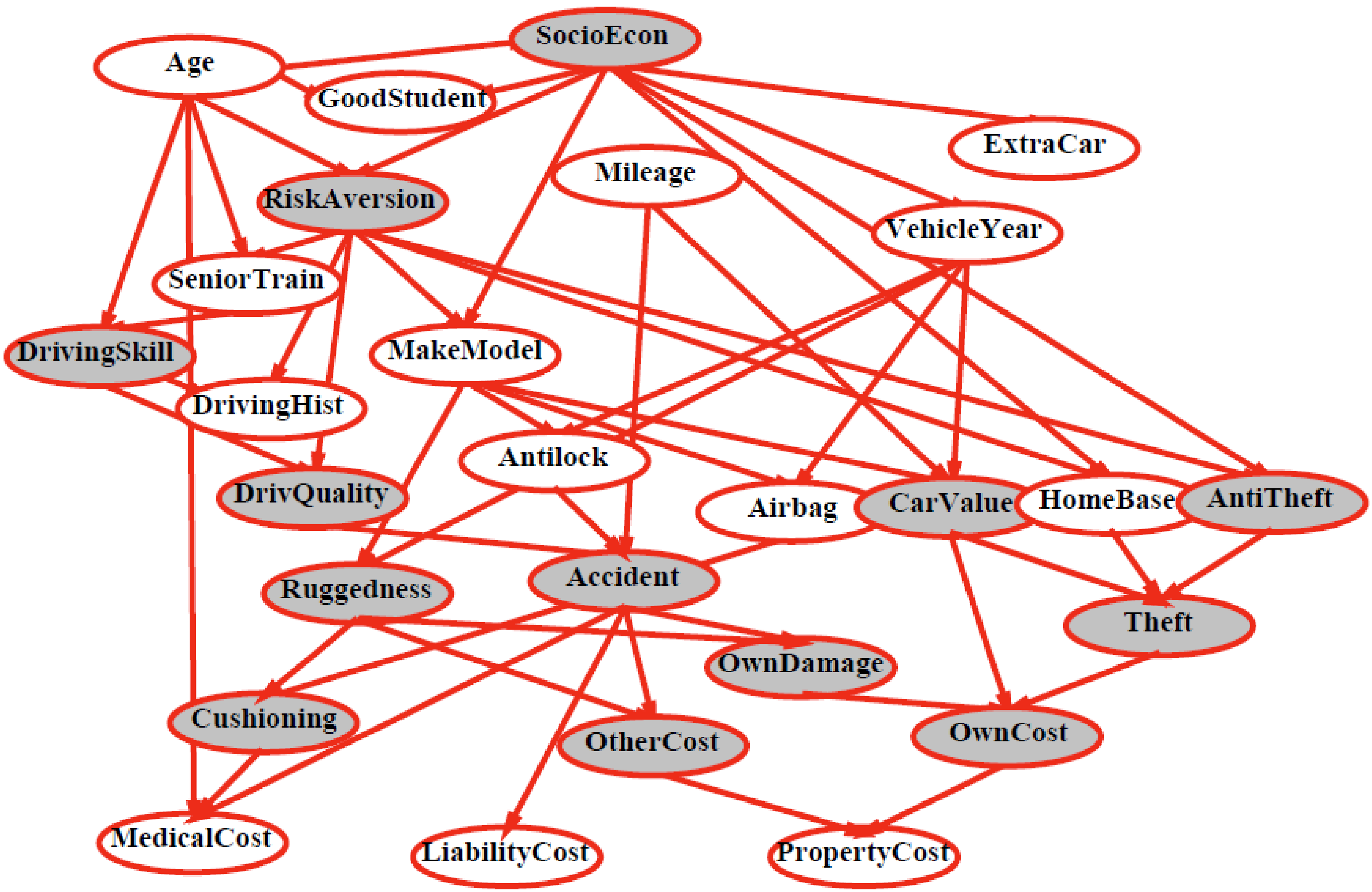
Inferente de aproximare in RB

Inferente exacte in RB foarte costisitoare

- Complexitate spatiu – O(n)
- Complexitate timp $O(2^n)$ – pt *n* variabile booleene

Pentru retele complexe aceste metode devin de multe ori nepractice

Asigurarea auto



Inferențe de aproximare

Inferente de aproximare

- Algoritmi de esantionare aleatoare, cunoscuti si ca algoritmi Monte Carlo
- Ofera raspunsuri aproximative care depind de numarul de esantioane care se genereaza
- Ne intereseaza esantionare pentru calculul probabilitatilor conditionate

Esantionare directa

Esantionare cu lanturi Markov

a) Esantionare directa

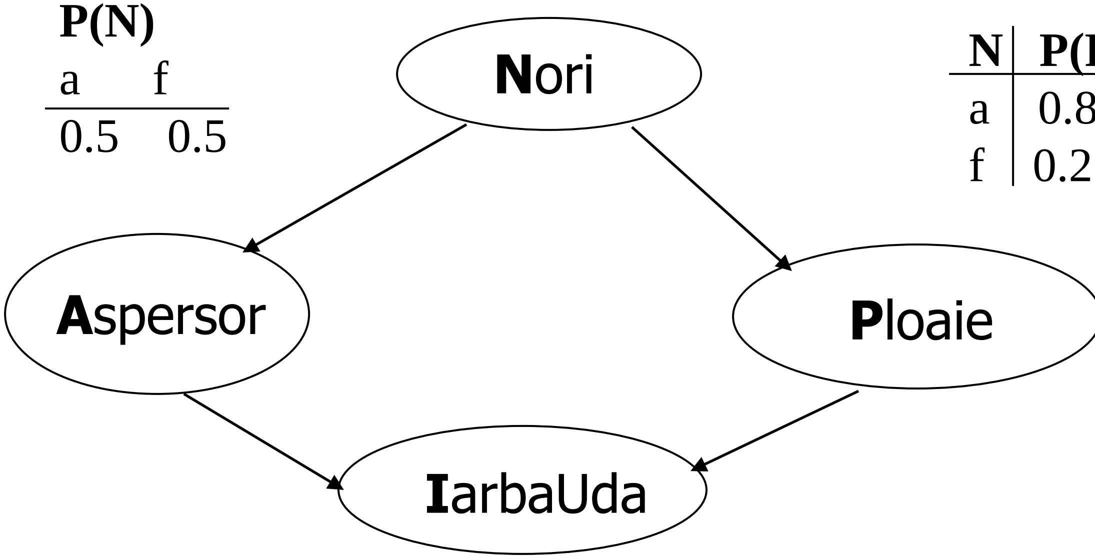
- Se genereaza esantioane dintr-o distributie de probabilitate cunoscuta
- Cel mai simplu proces de esantionare intr-o RB genereaza esantioane/evenimente din retea fara nici o proba
- Ideea este de a esantiona fiecare variabila pe rand in ordine topologica
- Distributia de probabilitate din care se obtin valorile unei variabile din esantion este conditionata de valorile care au fost deja atribuite parintilor

RB exemplu

P(N)

a	f
0.5	0.5

N	**P(A|N)**	
a	0.1	0.9
f	0.5	0.5



N	**P(P|N)**	
a	0.8	0.2
f	0.2	0.8

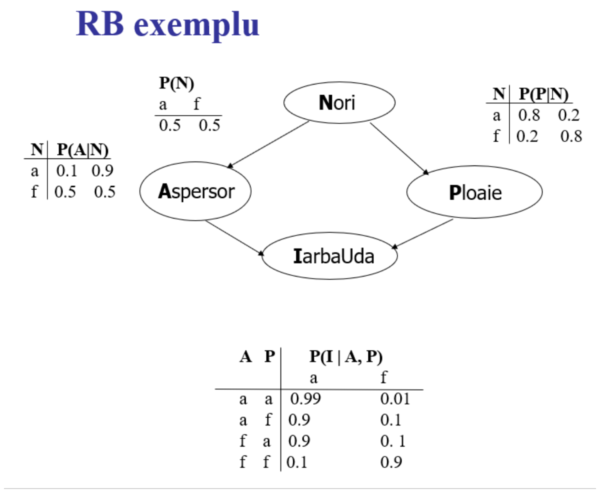
A	**P**	**P(I | A, P)** a	f
a	a	0.99	0.01
a	f	0.9	0.1
f	a	0.9	0. 1
f	f	0.1	0.9

Generarea esantionului

Presupun ordinea [*Nori*, *Aspersor*, *Ploaie*, *IarbaUda*]

1. Esantion **P(Nori)** = <0.5 0.5> **a**
2. Esantion din **P(Aspersor|Nori=a)** = <0.1 0.9> **f**
3. Esantion **P(Ploaie|Nori=a)** = <0.8 0.2> **a**
4. Esantion din **P(IarbaUda|Aspersor=f, Ploaie=a)** = <0.9 0.1> **a**

Esantionarea este **[a, f, a, a]**



Algoritm de esantionare

algoritm Esantionare(rb) **intoarce** esantion

intrari: rb specificand d.p. $P(X_1,\ldots,X_N)$

x $\leftarrow$ esantion cu n elemente

pentru fiecare variabila X_i din $X_1,\ldots,X_N$ **repeta**

x[i] $\leftarrow$ un esantion aleator din $P(X_i|parinti(X_i))$

intoarce x

sfarsit

Algoritm de esantionare

- Intr-un algoritm de esantionare, raspunsurile sunt calculate prin numararea numarului de esantioane generate
- Sa presupunem ca avem N esantionari

  $N_{PS}(x_1,\ldots,x_n)$ – de cate ori a aparut $x_1,\ldots,x_n$ in multimea de esantioane
- Se presupune ca acest numar va converge chiar la probabilitate

  $$\lim_{N\to\infty} (N_{PS}(x_1,\ldots,x_n)/N) = P(x_1,\ldots,x_n)$$
- Fie S_{PS} probabilitatea ca un eveniment(esantion) specific sa fie generat de algoritmul de esantionare

  $$S_{PS}(x_1,\ldots,x_n) = \Pi_{i=1,n} P(x_i|parinti(X_i)) = P(x_1,\ldots,x_n)$$

deoarece fiecare pas de esantionare depinde de valorile parintilor

Algoritm de esantionare

$S_{PS}(x_1,\ldots,x_n) = \Pi_{i=1,n} P(x_i|parinti(X_i)) = P(x_1,\ldots,x_n)$

$\mathbf{\lim_{N\to\infty} (N_{PS}(x_1,\ldots,x_n)/N) = S_{PS}(x_1,\ldots,x_n) = P(x_1,\ldots,x_n)}$

De exemplu sa consideram esantionul

[a, f, a, a] generat anterior

Probabilitatea de esantionare a acestui eveniment

$$\mathbf{S_{PS}(a,f,a,a) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324}$$

Pentru N foarte mare, putem presupune ca 32,4% din esantioane vor fi egale cu acesta

b) Eliminarea esantioanelor nepotrivite

Eliminarea esantioanelor (***Rejection sampling***)

- Se refera la situatia in care avem de estimat o probabilitate conditionata P(X|**e**)
- Se genereaza esantioanele din distributia de probabilitate specificata de retea
- Apoi se elimina acele esantioane care nu se potrivesc cu probele **e**
- In final estimarea lui P(X=x|**e**) este obtinuta prin numararea numarului de aparitii X=x in esantioanele ramase

Eliminarea esantioanelor nepotrivite

- Dorim sa estimam

  $$\boxed{P(Ploaie|Aspersor = a)}$$

- Generam, de ex, 100 esantioane
- Presupunem ca din cele 100 esantioane 73 au Aspersor = f, le eliminam
- Raman 27 cu Aspersor = a
- Din acestea 8 au Ploaie=a si 19 au Ploaie=f
- **P(Ploaie|Aspersor = a) = Normalizare(<8, 19>) =**

  < 0.296, 0.704>

Rapunsul corect este <0.3,0.7>

- Pe masura ce colectam mai multe esantioane, raspunsul va converge catre valoarea reala

Algoritm de esantionare cu eliminare

algoritm **EsantionareEliminare(X,e,rb,N)**

intoarce o estimare P(X|**e**)

intrari: rb specificand d.p. $P(X_1,\ldots,X_N)$

X – variabila de interogare

e – valorile observate pentru variabilele E

N – numarul total de esantioane generate

variabile locale VN – vector care numara aparitiile valorilor lui X

pentru j ← 1,N **repeta**

y ← Esantionare (rb)

daca *y* este consistent cu **e atunci**

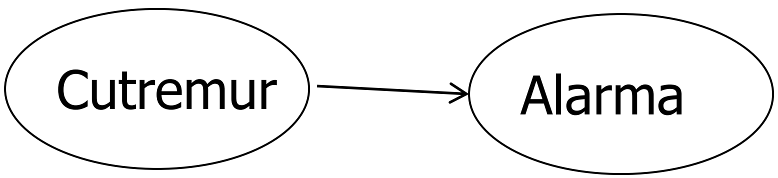
VN[x] ← VN[x]+1 unde x este valoarea lui X in *y*

intoarce Normalizare(VN)

sfarsit

Eliminarea esantioanelor nepotrivite

- Principala problema a acestei abordari este faptul ca elimina prea multe esantioane



- Esantioane pt P(Cutremur|Alarma=a)

  ~~Cutremur =f, Alarma = f~~

  ~~Cutremur =f, Alarma = f~~

  ~~Cutremur =f, Alarma = f~~

  Cutremur =a, Alarma = a

- Procentul de esantioane consistent cu probele scade exponential pe masura ce creste numarul de probe, deci problematic pentru probleme complexe

c) Esantionarea ponderata

Esantionare ponderata ***(Likelihood weighting)***

- Elimina ineficienta potentiala a eliminarii esantioanelor prin generarea numai a acelor evenimente (esantioane) care sunt consistente cu probele **e**
- Se fixeaza valorile variabilelor probe **E** si se esantioneaza numai pentru celelalte variabile
- Acest lucru garanteaza faptul ca fiecare esantion generat este consistent cu probele; dar nu toate evenimentele sunt la fel de importante

Esantionarea ponderata

- Inainte de a le numara, fiecare eveniment este ponderat cu o **plauzibilitate** (likelihood) pe care esantionul o acorda probei

- **Plauzibilitatea (likelihood)** este masurata ca **produsul probabilitatilor conditionate pentru fiecare variabila proba, fiind dati parintii ei**

Esantionarea ponderata - Exemplu

P(Ploaie|Nori=a, IarbaUda=a)

si ordonarea *[Nori, Aspersor, Ploaie, IarbaUda]*

Initial w=1.0

Se genereaza un esantion

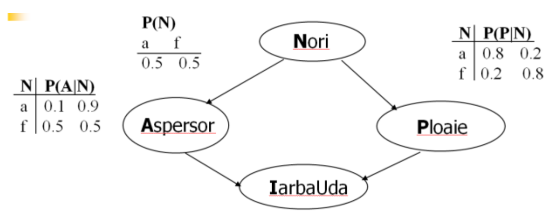
1. *Nori* este o variabila proba cu valoarea **a**

   In consecinta **w ← w x P(Nori=a) = 0.5**

2. *Aspersor* nu este o variabila proba deci esantionam

   P(Aspersor|Nori=a) = (0.1, 0.9) - presupun **f**

3. Similar P(Ploaie|Nori=a) = (0.8, 0.2) - presupun **a**



A	P	P(I | A, P) a	f
a	a	0.99	0.01
a	f	0.9	0.1
f	a	0.9	0. 1
f	f	0.1	0.9

Esantionarea ponderata - Exemplu

P(Ploaie|Nori=a, IarbaUda=a)

si ordonarea *[Nori, Aspersor, Ploaie, IarbaUda]*

4. *IarbaUda* este o variabila proba cu valoarea **a**

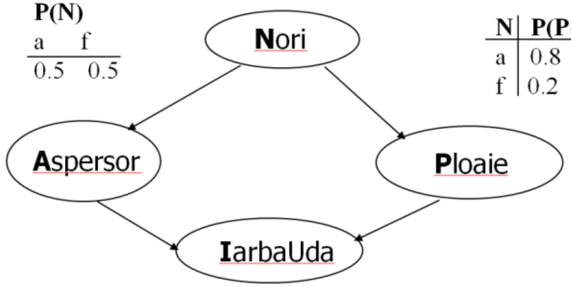
w ← w x P(IarbaUda=a | Aspersor=f, Ploaie=a) =

0.9 x 0.5 = 0.45

Esantionarea ponderata intoarce un esantion **[a,f,a,a] cu ponderea 0.45** in conditiile Ploaie=a

P(N)	
a	f
0.5	0.5

N	P(A|N)	
a	0.1	0.9
f	0.5	0.5



N	P(P|N)	
a	0.8	0.2
f	0.2	0.8

A	P	P(I | A, P) a	f
a	a	0.99	0.01
a	f	0.9	0.1
f	a	0.9	0. 1
f	f	0.1	0.9

Algoritm de esantionare ponderata

algoritm EsantionarePonderata(X,e,rb,N)

intoarce o estimare P(X|e)

intrari: rb specificand d.p. $P(X_1,\ldots,X_N)$

X – variabila de interogare

e – valorile observate pentru variabilele E

N – numarul total de esantioane generte

variabile locale W – vector care numara aparitiile valorilor lui X, ponderate

pentru j ← 1,N **repeta**

x,*w* ← Esantion-Ponderat(rb,e)

W[x] ← W[x]+*w*   unde x este valoarea lui X in ***x***

intoarce Normalizare(W)

sfarsit

Algoritm de esantionare ponderata

algoritm **Esantion-Ponderat(rb,e)**

intoarce un esantion si o pondere

w $\leftarrow$ 1

x $\leftarrow$ un esantion cu n elemente initializate din e

pentru fiecare variabila X_i din $X_1,..,X_n$ **repeta**

daca X_i este o variabila proba cu valoarea x_i in e

atunci w $\leftarrow$ w * $P(X_i=x_i \mid parinti(X_i))$

altfel **x**[i] $\leftarrow$ un esantion aleator din $P(X_i \mid parinti(X_i))$

intoarce **x**, w

sfarsit