Modele grafice. Reţele Bayesiene
Cursul 2

Programare şi modelare probabilistă - anul III

Facultatea de Informatică, UAIC

e-mail: adrian.zalinescu@uaic.ro

web: https://sites.google.com/view/fiicoursepmp/home

16 Octombrie 2023

Plan

Modele grafice

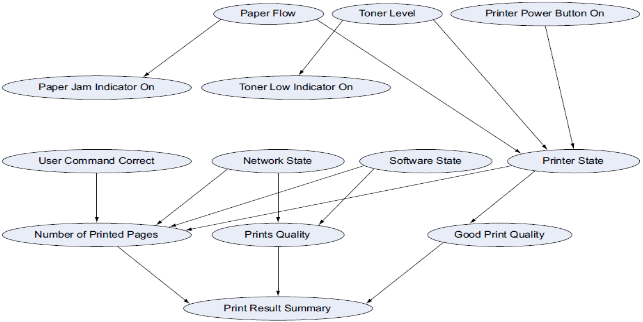
Cum putem face inferenţe într-un mediu aleator?
Putem folosi *modelele grafice direcţionate*:

- o reprezentare a distribuţiei multivariate

$$P(X_1, X_2, \ldots, X_n)$$

  factorizată de dependenţe condiţionale.
- reprezentăm variabilele prin noduri şi dependenţele prin muchii.

Exemplu:

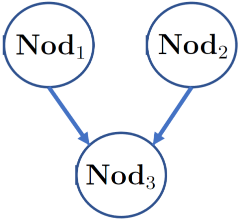


De ce folosim modelele grafice direcţionate?

Modelele grafice sunt generative:

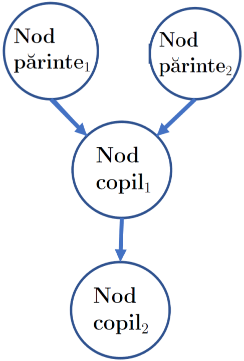
- pot defini distribuţii de probabilitate;
- sunt expresive în termeni de structură a dependenţelor;
- se pot genera realizări (eşantioane) din respectivele distribuţii.

Grafuri



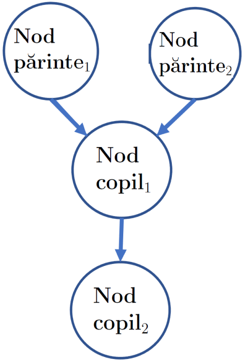
- *Nodurile* conţin distribuţii condiţionale (CPD);
- *Muchiile neorientate* definesc conexiunea între noduri (sau scheletul grafului);
- Informaţia sau influenţa curge de-a lungul *muchiilor orientate*;
- Nodurile conectate prin muchii se numesc *vecini*.

Grafuri



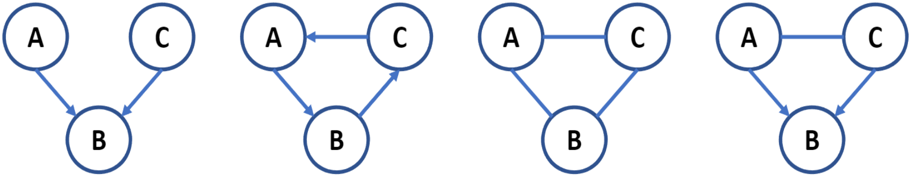
- *Nodurile părinţi* influenţează prin muchii orientate *nodurile copii*;
- Nodurile copii sunt *descendenţi* ai nodurilor părinţi;
- *Ascendenţii* sunt părinţii, bunicii, etc;
- Ex.: ascendenţii nodului NC_2: $A(\mathrm{NC}_2) = \{\mathrm{NP}_1, \mathrm{NP}_2, \mathrm{NC}_1\}$;
- *Descendenţii* unui nod sunt influenţaţi de acesta;
- Ex.: descendenţii nodului NP_1: $D(\mathrm{NP}_1) = \{\mathrm{NC}_1, \mathrm{NC}_2\}$.

Grafuri



- *Gradul unui nod:* numărul de vecini;
- *Gradul interior al unui nod:* numărul de părinţi;
- *Gradul exterior al unui nod:* numărul de copii;
- Exemplu: $IN(\mathrm{NC}_1) = 2$;
- Exemplu: $OUT(\mathrm{NC}_1) = 1$;
- Un *nod rădăcină* nu are ascendenţi: mulţimea acestora este $\{\mathrm{NP}_1, \mathrm{NP}_2\}$;
- Un *nod frunză* nu are descendenţi: mulţimea acestora este $\{\mathrm{NC}_2\}$.

Tipuri de Grafuri

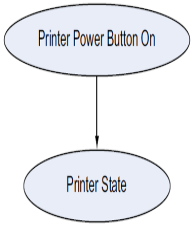


Graf aciclic orientat | Graf ciclic orientat | Graf neorientat | Graf parţial orientat

Tipuri de dependenţă în modele grafice

- *dependenţe direcţionate* de la o variabilă la alta.
  De obicei modelează o relaţie cauză-efect.
- *dependenţe nedirecţionate* , care modelează relaţii dintre variabile ce nu se influenţează direct una pe cealaltă.

Exemplu:



Starea imprimantei depinde de starea butonului de pornire.

Atenţie! Direcţia de inferenţă nu e neapărat cea de dependenţă.

Relaţii cauză-efect

- *temporale.* Exemplu: se apasă butonul de oprire mai întâi şi apoi imprimanta se va opri.
- *cauză-efect de stare.* Exemplu: o variabilă reprezintă dacă butonul de oprire este în starea “off”, iar cealaltă dacă imprimanta este oprită.
- *starea de adevăr a unei măsurări.* Când o variabilă *măsoară* valoarea unei alte variabile. Exemplu: o variabilă care reprezintă dacă becul ce indică conectarea imprimantei la sursa de curent electric este aprins. Măsurătorile sunt de obicei observate şi, în acest caz, raţionamentul merge în sens invers, către valoare.
- *de la parametru la o variabilă ce utilizează parametrul.* Exemplu: presupunerea despre o monedă, care reprezintă probabilitatea de a obţine “stemă”, şi o aruncare a monezii.

Relaţii asimetrice adiţionale

- *parte-întreg.* Exemplu: o imprimantă cu toner şi suport pentru alimentarea cu hârtie. Defectarea unei componente duce la defectarea întregii imprimante. Alteori, proprietăţi ale întregului pot influenţa proprietăţi ale componentelor.
- *specific-general.* Exemplu: un utilizator care a avut de a face cu multe probleme specifice de tip “paper jam” sau “poor printing quality” va şti cum să rezolve o problemă mai generală de tip “poor printing” (specific $\rightarrow$ general). La crearea unui obiect, se generează mai întâi proprietăţile generale şi apoi cele specifice (general $\rightarrow$ specific).
- *concret/detaliat - abstract/sumar.* Exemplu: relaţia dintre rezultatul unui test şi nota finală. Mai multe rezultate la teste influenţează nota finală.

Dependenţe nedirecţionate

- *două efecte ale aceeaşi cauze.* Exemplu: două măsurători ale aceleeaşi valori, dar nu există o variabilă pentru acea valoare.
- *două cauze ale aceluiaşi efect cunoscut.* Exemplu: De regulă, suportul de alimentare cu hârtie şi tonerul sunt independente. În cazul când starea generală a imprimantei este un efect (cunoscut), atunci cele două cauze devin depedente prin efect. Aceasta se numeşte *dependenţă indusă*.

Dependenţe în modele grafice orientate

Factorizarea unei distribuţii reduce mult complexitatea computaţională:

- *O distribuţie bivariată poate fi factorizată* ca o distribuţie condiţională şi una necondiţională:
$$P(X, Y) = P(X|Y)P(Y) = P(Y|X)P(X).$$

- Observăm că factorizarea nu este unică.

În cazul multivariat, formula înmulţirii sau formula înlănţuirii probabilităţilor stă la baza factorizării distribuţiilor:

- Mai întâi,
$$P(X_1, X_2, \ldots, X_n) = P(X_1|X_2, \ldots, X_n)P(X_2, \ldots, X_n).$$
- Continuând,
$$\begin{aligned} P(X_1, X_2, \ldots, X_n) &= P(X_1|X_2, \ldots, X_n) \cdot P(X_2|X_3, \ldots, X_n) \cdot \ldots \\ &\ldots \cdot P(X_{n-1}|X_n) \cdot P(X_n). \end{aligned}$$
- Factorizarea poate fi efectuată în orice altă ordine.
- Din păcate, a obţine cea mai bună factorizare este o problemă NP-dificilă.

Reţele Bayesiene

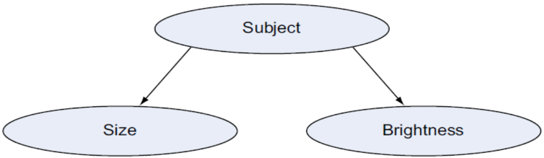
O reţea Bayesiană este o reprezentare a unui model probabilist cu trei componente:

1. o mulţime de variabile împreună cu domeniile valorilor corespunzătoare;
2. un graf aciclic orientat în care fiecare variabilă este un nod;
3. pentru fiecare variabilă, o distribuţie de probabilitate condiţională (CPD) peste variabilele care definesc părinţii.

Exemplu de reţea Bayesiană

Probability values for Subject

Subject	
People	Landscape
0.8	0.2



Probability values for Size, given Subject

Subject	Size		
	Small	Medium	Large
People	0.25	0.25	0.5
Landscape	0.25	0.5	0.25

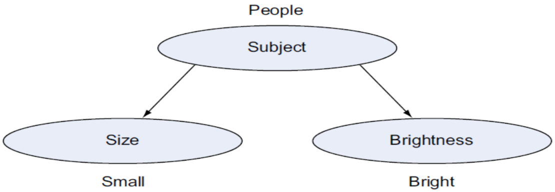
Probability values for Brightness, given Subject

Subject	Brightness	
	Dark	Bright
People	0.8	0.2
Landscape	0.3	0.7

Figure 5.4 A three-node Bayesian network

P(Subject + People)

Subject	
People	Landscape
0.8	0.2



P(Size = Small /
Subject = People)

Subject	Size		
	Small	Medium	Large
People	0.25	0.25	0.5
Landscape	0.25	0.5	0.25

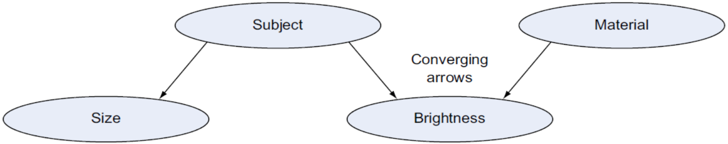
P(Brightness = Bright /
Subject = People)

Subject	Brightness	
	Dark	Bright
People	0.8	0.2
Landscape	0.3	0.7

P(Subject = People, Size = Small, Brightness = Bright) = 0.8 x 0.25 x 0.2 = 0.04

Convergența arcelor și dependența indusă

1. "Subject" și "Material" sunt independente când nimic nu e observat.
2. "Subject" și "Material" NU sunt independente condițional când "Brightness" este observată.



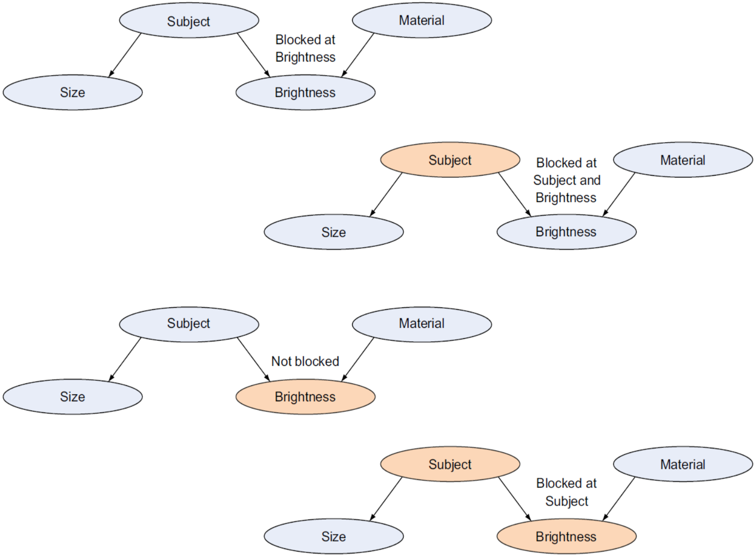
Raţionamentul într-o reţea (d-separare)

- Într-o reţea, raţionamentul poate evolua pe un drum atât timp cât acesta este *neblocat* la o anumită variabilă.
- În multe cazuri un drum este *blocat* la o variabilă dacă variabila este observată.

Exemplu: dacă “Subject” este observat, atunci drumul “Size-Subject-Brightness” este blocat. Dacă se observă “Size”, atunci nu se poate schimba încrederea despre “Brightness” dacă “Subject” este şi ea observată.

- În alte cazuri, un drum este *blocat* la o variabilă dacă variabila este neobservată şi devine *neblocat* când variabila devine observată (exemplu pe slide-ul următor).

Exemple de drumuri blocate/neblocate



- Când se raţionează de la un efect X la o cauză a sa Y , X nu este independent de Y, dar devine independent condiţional când e observat Z, dacă Z blochează drumul de la X la Y.
- Acelaşi lucru se întâmplă când raţionamentul merge de la o cauză la un efect indirect sau între două efecte ale aceleeaşi cauze.
- Pentru două cauze X şi Y ale aceluiaşi efect Z, lucrurile merg în sens contrar. X şi Y sunt independente, dar nu şi independente condiţional când e observat Z (din cauza dependenţei induse).

Dependenţe în reţelele Bayesiene

- Într-un graf orientat, alegerea părinţilor determină semantica.
- Factorizarea întregii distribuţii este definită de *semantica globală* a unui graf aciclic orientat.
- Semantica globală este specificată de

$$P\left(X\right) = \prod_{i=1:d} P\left(X_i \,|\, \pi(X_i)\,\right),$$

unde X_i, $1 \leq i \leq d$ reprezintă nodurile grafului, $X = (X_1, \ldots, X_d)$, iar $\pi(X_i)$ mulţimea părinţilor lui X_i.

Exemplu de factorizare a unei distribuţii

- Un student caută un loc de muncă la o companie IT;
- Angajatorul ia decizia de angajare bazat pe un test de aptitudini şi pe calitatea scrisorii de recomandare de la profesorul de ML;
- Scorul la testul de aptitudini depinde doar de inteligenţa studentului;
- Nota la ML depinde de inteligenţa studentului, dar şi de gradul de dificultate al materiei;
- Din păcate, profesorul nu-şi mai aminteşte de student şi îşi bazează scrisoarea de recomandare doar pe nota obţinută.

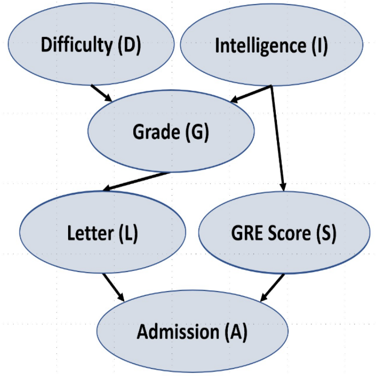
Distribuţia comună pentru acest exemplu este dată de:

$$P(D, I, S, G, L, A),$$

unde

- $D \in \{0, 1\}$: dificultatea cursului de ML;
- $I \in \{0, 1\}$: inteligenţa studentului;
- $S \in \{0, 1\}$: scorul la testul de aptitudini;
- $G \in \{0, 1, 2\}$: nota (grade) la cursul de ML;
- $L \in \{0, 1\}$: calitatea scrisorii de recomandare;
- $A \in \{0, 1\}$: decizia de angajare.

Astfel, tabelul complet al distribuţiei are $2 \times 2 \times 2 \times 3 \times 2 \times 2 = 96$ intrări.



- $D \perp I$;
- $L \perp D, I \mid G$
- $S \perp D, G, L \mid I$;
- $A \perp D, I, G \mid L, S$.

Astfel,

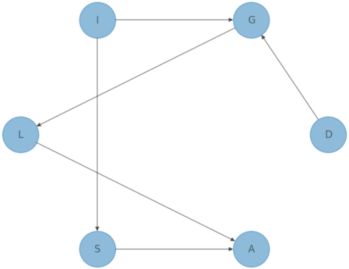
$$P(D,I,G,L,S,A) = P(D)P(I)P(G|I,D)P(L|G)P(S|I)P(A|L,S).$$

Modelarea BN cu pgmpy

```
from pgmpy.models import BayesianNetwork
from pgmpy.factors.discrete import TabularCPD
import networkx as nx

student_model = BayesianNetwork([('D', 'G'), ('I', 'G'), ('G', 'L'),
                                 ('I', 'S'), ('L','A'),('S','A')])

pos = nx.circular_layout(student_model)
nx.draw(student_model, with_labels=True, pos=pos, alpha=0.5, node_size=2000)
```



Definirea variabilelor rădăcină:

```
CPD_D = TabularCPD(variable='D', variable_card=2, values=[[0.3], [0.7]])
print(CPD_D)
CPD_I = TabularCPD(variable='I', variable_card=2, values=[[0.2], [0.8]])
print(CPD_I)
```

▷
```
+------+-----+
| D(0) | 0.3 |
+------+-----+
| D(1) | 0.7 |
+------+-----+
+------+-----+
| I(0) | 0.2 |
+------+-----+
| I(1) | 0.8 |
+------+-----+
```

Definirea variabilor cu un părinte:

```
CPD_L = TabularCPD(variable='L', variable_card=2,
                   values=[[0.9, 0.6, 0.01],
                           [0.1, 0.4, 0.99]],
                   evidence=['G'],
                   evidence_card=[3])
CPD_S = TabularCPD(variable='S', variable_card=2,
                   values=[[0.8, 0.1],
                           [0.2, 0.9]],
                   evidence=['I'],
                   evidence_card=[2])
print(CPD_L,CPD_S)
```

▷
```
+------+------+------+------+
| G    | G(0) | G(1) | G(2) |
+------+------+------+------+
| L(0) | 0.9  | 0.6  | 0.01 |
+------+------+------+------+
| L(1) | 0.1  | 0.4  | 0.99 |
+------+------+------+------+
+------+------+------+
| I    | I(0) | I(1) |
+------+------+------+
| S(0) | 0.8  | 0.1  |
+------+------+------+
| S(1) | 0.2  | 0.9  |
+------+------+------+
```

Definirea variabilor cu doi părinţi:

```
CPD_G = TabularCPD(variable='G', variable_card=3,
                   values=[[0.3, 0.7,  0.02, 0.2],
                           [0.4, 0.25, 0.08, 0.3],
                           [0.3, 0.05, 0.9,  0.5]],
                  evidence=['I', 'D'],
                  evidence_card=[2, 2])
print(CPD_G)

CPD_A = TabularCPD(variable='A', variable_card=2,
                   values=[[0.9, 0.8,  0.7, 0.2],
                           [0.1, 0.2, 0.3, 0.8]],
                  evidence=['L', 'S'],
                  evidence_card=[2, 2])
print(CPD_A)
```

I	I(0)	I(0)	I(1)	I(1)
D	D(0)	D(1)	D(0)	D(1)
G(0)	0.3	0.7	0.02	0.2
G(1)	0.4	0.25	0.08	0.3
G(2)	0.3	0.05	0.9	0.5

L	L(0)	L(0)	L(1)	L(1)
S	S(0)	S(1)	S(0)	S(1)
A(0)	0.9	0.8	0.7	0.2
A(1)	0.1	0.2	0.3	0.8

Adăgarea distribuţiilor condiţionale la model:

```
student_model.add_cpds(CPD_D, CPD_I, CPD_S, CPD_G, CPD_L, CPD_A)
student_model.get_cpds()
```

▷
```
[<TabularCPD representing P(D:2) at 0x17aeca4e5f0>,
 <TabularCPD representing P(I:2) at 0x17ae87d5420>,
 <TabularCPD representing P(S:2 | I:2) at 0x17aecb65570>,
 <TabularCPD representing P(G:3 | I:2, D:2) at 0x17ae87d5120>,
 <TabularCPD representing P(L:2 | G:3) at 0x17aecb66aa0>,
 <TabularCPD representing P(A:2 | L:2, S:2) at 0x17ae87a6a70>]
```

Verificarea modelului:

```
student_model.check_model()
```

▷
```
True
```

Verificarea independenţelor:

```
student_model.local_independencies(['D','G','S','I','L','A'])
```

- (D ⊥ S, I)
  (G ⊥ S | D, I)
  (S ⊥ G, L, D | I)
  (I ⊥ D)
  (L ⊥ S, D, I | G)
  (A ⊥ G, D, I | S, L)

Inferenţa:

```
from pgmpy.inference import VariableElimination

infer = VariableElimination(student_model)
posterior_p = infer.query(["D","I"], evidence={"A": 1})
print(posterior_p)
```

```
▷ +------+------+------------+
  | D    | I    |   phi(D,I) |
  +======+======+============+
  | D(0) | I(0) |     0.0298 |
  +------+------+------------+
  | D(0) | I(1) |     0.3291 |
  +------+------+------------+
  | D(1) | I(0) |     0.0492 |
  +------+------+------------+
  | D(1) | I(1) |     0.5918 |
  +------+------+------------+
```