# Rețele Bayesiene

## Context exerciții

Pentru toate exercițiile de mai jos, fișierul de intrare are formatul descris mai jos.

**pentru fiecare nod** (cu părinți) avem:

* un rând care începe cu "nod:" urmat de identificatorul lui (un string care definește la ce se referă; pentru simplitate, vom considera stringul fără spații, cuvintele diferite din identificator fiind separate prin underline, cum ar fi citeste\_carte\_domeniu)
* un rând cu părinții nodului respectiv (adică identificatorii lui)
* urmat de 2n rânduri cu toate combinațiile de adevărat/fals pentru părinți si probabilitatea ca în acea situație valoarea nodului curent să fie adevărată. Fiecare astfel de rând începe cu valori binare (din setul: 1,0). Valoarea binară de pe poziția i corespunde valorii de adevăr pentru părintele de pe poziția i. Rândul se încheie cu probabilitatea condiționată ca nodul curent să fie adevărat pentru valorile de adevăr indicate ale părinților

Pentru fiecare nod fără părinți avem:

* un rând care începe cu "nod:" urmat de identificatorul lui
* un rând cu probabilitatea ca nodul să fie adevărat.

Pe penultimul rând, precedate de textul "observatii:" vor fi trecute variabilele observate și valoare lor de adevăr. Variabilele au formatul nume\_variabila:valoare\_adevar și sunt separate cu ";" urmat de spațiu.

Pe ultimul rând, precedat de textul "interogare:" va fi trecut identificatorul variabilei de interogare.

Exemplu de fișier de intrare:

nod: angajat

0.4

nod: vine\_la\_ore

angajat

1 0.2

0 0.7

nod: jocuri\_calculator

0.5

nod: planifica\_timp

0.4

nod: timp\_studiu\_acasa

jocuri\_calculator, planifica\_timp

1 1 0.5

1 0 0.1

0 1 0.95

0 0 0.3

nod: pasionat

0.35

nod: citeste\_carte\_domeniu

pasionat, timp\_studiu\_acasa

1 1 0.85

1 0 0.45

0 1 0.3

0 0 0.05

nod: face\_temele

vine\_la\_ore, timp\_studiu\_acasa

1 1 0.95

1 0 0.35

0 1 0.7

0 0 0.15

nod: promoveaza\_materia

face\_temele

1 0.9

0 0.4

observatii:

angajat:0; jocuri\_calculator:1; pasionat:1; planifica\_timp:0

interogare: promoveaza\_materia

## Exerciții

1) **[deja rezolvat]** Definiți o clasa Nod, care va reprezenta un nod din rețeaua Bayesianăcu câmpurile: identificator, parinti, fii, probabl. In constructor, vom defini pentru parinti si fii, ca valori implicite lista vidă. Proprietatea probabl, reprezintă probabilitatea ca variabila asociată nodului să fie adevărată și va avea ca valoare implicită None.

În plus, va exista și proprietatea tabel, în care va fi stocat tabelul de probabilități condiționate sub formă de listă de liste. Pentru fiecare listă, primele elemente vor reprezenta valori booleene corespunzătoare părinților nodului și ultima valoare va fi probabilitatea condiționată corespunzătoare nodului curent.

Va exista și proprietatea val (folosită pentru nodurile observate).

Se vor defini metodele:

* \_\_str\_\_ care afișează nodul cu probabilitatea asociată.
* \_\_repr\_\_ care afișează nodul cu toate informațiile ( toate proprietățile și valorile lor)

2) **[deja rezolvat]** Definiți o clasă numită Rețea. Aceasta va curpinde toate datele rețelei. Rețeaua va fi organizată sub formă de dictionar de noduri (din clasa Nod definită mai sus)

Definiți un constructorul care va primi calea către fisierul de intrare. Se vor citi din fișier datele corespunzătoare rețelei. Se va crea rețeaua Bayesiană având un dicționar cu cheile egale cu identificatorii nodurilor si valorile egale cu obiectele de tip Nod corespunzătoare. Se vor memora nodurile cu valorile observate.

3) **[deja rezolvat]** Implementați în clasa retea metodele definite la curs (toți parametrii sunt noduri; deci variabile aleatoare):

* interogare\_retea(self, nod)
* suport\_except(self, x,v)
* dovezi\_except(self, x,v)

Afișați la fiecare pas calculele intermediare făcute pentru fiecare nod.

4) Se va implementa în clasa Retea o funcție numită valideaza() care să verifice dacă graful direcționat (orientat) dat în fișierul de intrare poate fi o rețea Bayesiană. Veți testa dacă:

* este acicilic
* s-au dat probabilități pentru toate combinațiile posibile de valori booleene ale cauzelor (părinților) pentru un nod dat.
* probabilitățile sunt corecte ( în intervalul [0,1])

5) Definiți în clasa Retea o funcție numită **blocat\_condiționat(drum, multime\_noduri)** care verifică dacă drumul este blocat condiționat de mulțimea de noduri. Atât drumul cât și mulțimea de noduri vor fi date ca liste de identificatori de noduri. Dacă unul dintre identificatorii din liste nu există în rețea, metoda va arunca o eroare.

6) (depinde de realizarea exercitiului 5) Definiți în clasa Retea o funcție numită **d\_separa(x,y, e)** care verifică dacă mulțimea **e** d-separă mulțimile **x** și **y**.

7) Pentru programul implementat la laborator, faceți următoarele îmbunătățiri:

* identificatorii vor avea un cod numeric asociat, de la 0 la N-1, unde N e numărul de noduri din rețeaua Bayesiană.
* se vor folosi doar codurile numerice în listele de identificatori
* parinții și fiii vor fi memorați ca referințe către nodurile corespunzătoare (în loc să fie memorați identificatorii)
* tabelul de probabilități va fi memorat ca o listă de tupluri de forma (cod, probabilitate). codul va fi o sumă de puteri ale lui 2 cu exponent mai mic strict decât numărul de parinți ai nodului. Codul va indica valorile True/False pentru părinți.

8) Pentru programul implementat la laborator să se afișeze pentru o variabilă dată suportul cauzal, E+X (toate variabilele dovezi conectate prin intermediul părinților) și suportul probatoriu, E-X (variabilele dovezi conectate prin intermediul fiilor). Atenție, nu e vorba doar de părinți și fii, ci de **toate nodurile**.

9) Definiți propria rețea Bayesiană pentru ce problemă doriți voi, cu următoarele restricții:

* rețeaua să aibă minim 10 noduri
* rețeaua să fie conexă (nu tare conexă) - adică să existe un drum nedirecționat între orica două noduri din rețea.
* să existe:
  + măcar un nod cu minim 3 părinți:
  + măcar un nod cu minim 3 fii
  + minim un nod cu 2 părinți
  + minim un nod cu 2 fii
  + măcar un drum direcționat cu lungime minim 5.

10) Implementați rețeaua Bayesiană din **input1.txt** folosind modulul **pgmpy** (<https://pgmpy.org/>). Veți folosi aceleași variabile observate și aceeași variabilă de interogare din fișier. Implementarea se va face automat pe baza fișierului (parsând fisierul, nu manual!).