

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 citește_carte_domeniu)
- un rând cu părinții nodului respectiv (adică identificatorii lui)
- urmat de 2ⁿ rânduri cu toate combinațiile de adevărat/fals pentru părinți și 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 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 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: citește_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, vine_la_ore
1 1 0.99
1 0 0.55
0 1 0.6
0 0 0.1
angajat:0; jocuri_calculator:1; pasionat:1; planifica_timp:0
promoveaza_materia

```

Exerciții

1) **[deja rezolvat]** Definiți o clasă Nod, care va reprezenta un nod din rețeaua Bayesiană cu câmpurile: identificator, parinti, fii, probabl. În constructor, vom defini pentru parinti și 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.

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 cuprinde toate datele rețelei. Rețeaua va fi organizată sub formă de listă de noduri (din clasa *Nod* definită mai sus) Definiți un constructor care va primi calea către fișierul de intrare. Se vor citi din fișier datele corespunzătoare rețelei. Se va crea rețeaua Bayesiană având un dicționar cu cheile egale cu identificatorii nodurilor și 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(nod)`
- `suport_except(x,v)`
- `dovezi_except(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 aciclic
- 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) 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
- părinții și fiii vor fi memorați ca referințe către nodurile corespunzătoare
- tabelul de probabilități va fi memorat ca o listă de tuple de forma (cod, probabilitate). codul va fi o sumă de puteri ale lui 2 cu exponent mai mic strict decât numărul de părinți ai nodului. codul va indica valorile True/False pentru părinți.

8) Pentru programul implementat la laborator să se afișeze pentru o variabilă dată suportul causal (variabilele dovezi conectate prin intermediul părinților) și suportul probatoriu (variabilele dovezi conectate prin intermediul fiilor).

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 oricare 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 input3.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!).