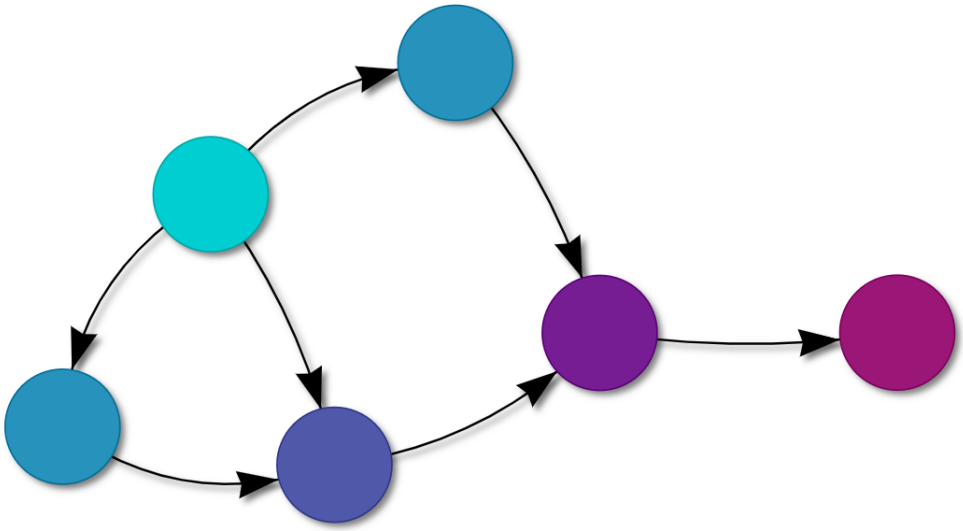
Structuri de date - Seria CD

Rețele Bayesiene

Tema 3

Deadline: ~~28.05.2021~~ 30.05.2021

Responsabil: Mihai Nan (*mihai.nan@upb.ro*)



Facultatea de Automatică şi Calculatoare
Universitatea Politehnica din Bucureşti
Anul universitar 2020 - 2021

Obiective

În urma realizării acestei teme, studentul va fi capabil:

- să implementeze şi să utilizeze grafuri în rezolvarea unei probleme;
- să înţeleagă noțiunea de graf orientat aciclic;
- să implementeze un algoritm care permite verificarea independenței condiționale într-o rețea Bayesiană;
- să transpună o problemă din viaţa reală într-o problemă care uzitează grafuri.

1 Descriere

1.1 Noțiuni generale

Rețelele Bayesiene sunt modele grafice utilizate pentru reprezentarea eficientă a unor distribuții comune peste un spațiu mai mare de variabile aleatoare. **Rețelele Bayesiene** sunt grafuri orientate aciclice în care nodurile reprezintă variabile aleatoare, iar arcele reprezintă relații cauzale directe între acestea.

Rețele Bayesiene pot lucra și cu variabile continue, dar în cadrul acestei teme ne vom referi doar la rețele ce descriu probabilități pentru variabile discrete.

Figura 1 reprezintă o rețea bayesiană cu două variabile binare: A și B. Pentru a simplifica notațiile vom considera evenimentele $A \equiv \mathsf{A} = 1$ și $\neg A \equiv \mathsf{A} = 0$ notând, deci, $p(A) = p(\mathsf{A} = 1)$ și probabilitatea evenimentului complementar $p(\neg A) = p(\mathsf{A} = 0)$. Convenția este folosită pentru toate variabilele binare B, C, D, etc. ce vor apărea mai jos.

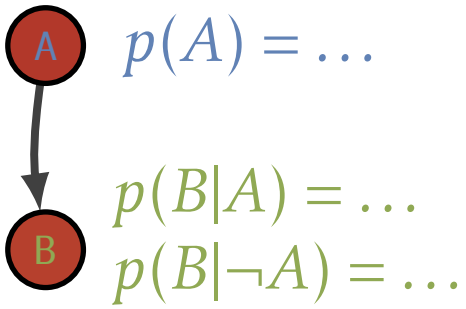


Figure 1: Rețea Bayesiană simplă

Fiecare nod dintr-o rețea bayesiană este însoțit de o tabelă de probabilități condiționate care descriu șansele ca variabila respectivă să ia o valoare atunci când variabilele părinte (cauzele) au fost observate. Pentru o variabilă care nu are părinți (cauze), probabilitățile de realizare ale acesteia nu vor fi condiționate.

Mai precis, pentru rețeaua bayesiană din Figura 1 trebuie precizate $p(A)$, $p(B|A)$ și $p(B|\neg A)$.

În cazul general, în care fiecare variabilă X poate lua valori dintr-o mulțime de valori $D(X)$, numărul de valori ce trebuie specificate pentru un nod oarecare este $(|D(X)| - 1) \prod_{\mathsf{Y} \in Par(\mathsf{X})} |D(Y)|$, unde $|M|$ este cardinalul mulțimii M, iar $Par(X)$ reprezintă mulțimea părinților (cauzelor) variabilei X.

Important

Structura grafică a unei rețele bayesiene are forma unui **graf orientat aciclic**.

1.2 Independență condițională

Unul dintre conceptele cheie din teoria probabilităților care face ca lucrul cu rețelele bayesiene să fie eficient este *independența condițională*. Dacă observarea unei variabile Y face ca variabila X să nu fie dependentă de o a treia variabilă Z, atunci spunem că X este condițional independentă de Z dată fiind Y. Intuitiv, această relație indică faptul că atunci când rezultatul lui Y este cunoscut, observarea lui Z nu aduce nicio informație suplimentară despre X (sau, echivalent, nu ar modifica incertitudinea asupra variabilei X). Una dintre notațiile uzuale pentru independența condițională este următoarea:

$$X \perp Z|Y \Leftrightarrow p(X|Y,Z) = p(X|Y)$$

Relația de independență condițională se generalizează și la mulțimi de variabile. În formula următoarea, **X**, **Y** și **Z** reprezintă mulțimi de variabile aleatoare.

$$\mathbf{X} \perp \mathbf{Z}|\mathbf{Y} \Leftrightarrow p(\mathbf{X}|\mathbf{Y},\mathbf{Z}) = p(\mathbf{X}|\mathbf{Y})$$

1.3 D-Separabilitate

Noțiunea de *D-separabilitate* folosește o analiză a grafului care descrie rețeaua bayesiană pentru a determina dacă două mulțimi de variabile sunt independente condițional sau nu.

O *cale neorientată* între o mulțime de noduri **X** și o mulțime de noduri **Z** este orice secvență de arce (indiferent de orientarea lor) care pleacă de la un membru al mulțimii **X** și ajunge la un membru al mulțimii **Z**.

Dându-se o mulțime de noduri **O** observate, o cale **p** este *blocată* dacă una dintre următoarele trei condiții este adevărată:

1. există un nod $Y \in \mathbf{O}$ care reprezintă destinația unui arc din **p** și sursa unui alt arc din **p** (înlănțuire cauzală);
2. există un nod $Y \in \mathbf{O}$ care reprezintă sursa a două arce diferite din **p** (cauză comună);
3. niciun nod Y care reprezintă destinația a două arce din **p** nu se află în **O** și nici vreun descendent de-ai lui nu se află în **O**.

O mulțime de noduri **O** *d-separă* două mulțimi **X** și **Z** dacă orice cale neorientată între **X** și **Z** este blocată prin observarea variabilelor din **O**.

2 Cerințe

2.1 Cerința 1

Pentru această cerință va trebui să se modeleze o rețea bayesiană sub forma unui graf reprezentat prin liste de adiacență și să se verifice dacă aceasta este corectă.

În fișierul `bnet.in` este oferită o reprezentare a unei rețele bayesiene, sub formă de text. Pornind de la această reprezentare, se va produce graful asociat, apoi se va verifica acest graf este un graf neorientat aciclic. Dacă graful conține cel puțin un ciclu, atunci se va afișa mesajul `imposibil`, iar dacă graful nu conține niciun ciclu, atunci se va afișă mesajul `corect` în fișierul `bnet.out`.

Important

Este obligatoriu ca implementarea acestei cerințe să fie realizată utilizând o structură de date de tip graf orientat reprezentat prin liste de adiacență.
Orice alt tip de implementare **NU** va fi punctat.
Pentru a verifica dacă un graf orientat este sau nu aciclic, puteți folosi unul dintre algoritmii de parcurgere discutați în cadrul cursului.

2.2 Cerința 2

Pentru a determina dacă două variabile A și B sunt sau nu independente condițional, dându-se o mulțime de noduri $\mathbf{O}$ observate, vom folosi următorul algoritm.

Pasul 1. Construim graful ancestral

Pornim de la graful inițial care descrie rețeaua bayesiană și păstrăm doar nodurile corespunzătoare variabilelor menționate și cele pentru toți strămoșii lor.

Pasul 2. Moralizăm graful ancestral

Pentru fiecare pereche de noduri cu un copil comun, adăugăm o muchie neorientată între ele. Dacă pentru un nod avem mai mult de doi părinți, adăugăm câte o muchie neorientată între oricare pereche de părinți.

Pasul 3. Transformăm graful moralizat într-un graf neorientat

Modificăm graful prin eliminarea orientării arcelor, obținând astfel un graf moralizat neorientat.

Pasul 4. Eliminăm nodurile observate și muchiile care le conțin

Din graful moralizat neorientat ștergem toate nodurile din mulțimea $\mathbf{O}$ și toate muchiile care au una dintre extremități un nod din această mulțime.

Pasul 5. Determinăm rezultatul pe baza grafului obținut

- Dacă nu există o cale între nodurile corespunzătoare variabilelor A și B în graful obținut, atunci putem garanta că acestea sunt independente condițional.
- Dacă există o cale între nodurile corespunzătoare variabilelor A și B în graful obținut, atunci **NU** putem garanta că acestea sunt independente condițional.
- Dacă pentru una dintre variabilele A sau B nu mai avem un nod asociat în graful obținut, atunci putem garanta că acestea sunt independente condițional.

Exemplu de aplicare

Pornim de la următoarea rețea bayesiană:

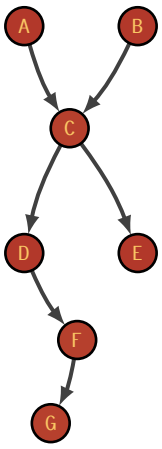


Figure 2: Exemplu de rețea bayesiană pentru care vom aplica algoritmul prezentat anterior

Vrem să verificăm dacă variabilele A și B sunt independente condițional, dându-se mulțimea de observații $\mathbf{O} = \{D, F\}$.

Pasul 1.

Vom păstra, în primul rând, nodurile corespunzătoare variabilelor menționate: $\mathbf{V_1} = \{A, B\} \cup \mathbf{O} = \{A, B, D, F\}$.

De asemenea, vom păstra nodurile pentru toți strămoșii nodurilor din mulțimea $\{A, B, D, F\}$.

$$\mathbf{V_2} = \mathbf{V_1} \cup \{\text{strămoși}(X) | X \in \mathbf{V_1}\} = \{A, B, C, D, F\}$$

Graful ancestral rezultat este:

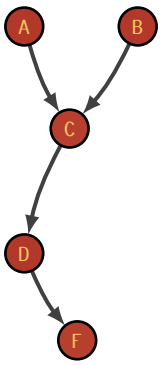


Figure 3: Graful ancestral obținut

Pasul 2.

Vom verifica dacă există noduri care au cel puțin doi părinți. În cazul nostru, există nodul corespunzător variabilei C care are doi părinți. Vom uni acești părinți printr-o muchie.

Graful moralizat obținut este:

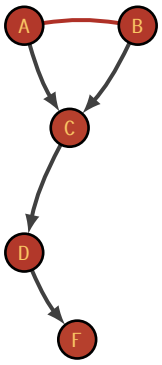


Figure 4: Graful moralizat obținut

Pasul 3.

Nu vom mai ține cont de orientările arcelor și le vom transforma pe toate în muchii.

Graful moralizat neorientat obținut este:

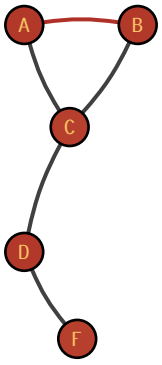


Figure 5: Graful moralizat **neorientat** obținut

Pasul 4.

Eliminăm nodurile corespunzătoare variabilelor observate din **O** și muchiile care conțin aceste noduri. Cu alte cuvinte, vom elimina nodurile D și F, respectiv muchiile (C, D) și (D, F).

Graful final obținut este:

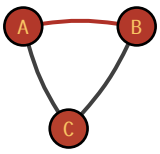


Figure 6: Graful final obținut

Pasul 5.

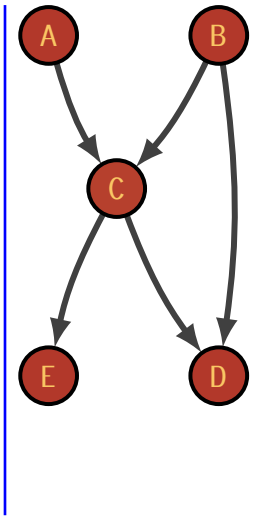
Observăm că între A și B există o cale (compusă chiar din muchia (A, B)) $\Rightarrow$ **NU** putem garanta că acestea sunt independente condițional.

3 Formatul fișierelor

3.1 Fișier de intrare

Un exemplu de fișier de intrare este următorul:

```
5 5
A B C D E
A C
B C
C D
B D
C E
8
E ; D|
E ; D|B
E ; D|C
A ; B|
A ; B|C
A ; B|D
A ; B|D E
A ; B|D E A
```



Formatul acestui fișier este următorul:

Cerința 1

- pe prima linie se găsesc două numere naturale n și m despărțite prin câte un spațiu (n – numărul de noduri din graf, m – numărul de arce din graf);
- pe a doua linie se găsesc denumirile nodurilor din graf (fiecare nod va avea ca nume un șir de caractere de cel mult 10 caractere) despărțite prin câte un spațiu;
- pe următoarele m linii se vor găsi arcele grafului (fiecare linie conține numele extremităților arcului despărțite prin câte un spațiu);

Cerința 2

- după ce se descrie graful, va exista o linie care va conține un număr natural k ce reprezintă numărul de interogări;
- pe următoarele k linii se vor găsi interogările.

O interogare este de forma `X ; Y|O1 O2 ... On` unde X și Y sunt cele două variabile pentru care vrem să verificăm independență condițională, iar $O1$, $O2$, ..., On reprezintă variabilele observate.

Important

Pot exista interogări pentru care să nu avem nicio variabilă observată.
Numele nodurilor nu vor conține spații.
Grafurile din testele pentru cerința 2 vor fi **grafuri orientate aciclice**.

3.2 Fișier de ieșire

Cerința 1

Fișierul de ieșire, corespunzător celui de intrare prezentat anterior ca exemplu, pentru prima cerință va avea conținutul:

```
corect
```

Cerința 2

Fișierul de ieșire, corespunzător celui de intrare prezentat anterior ca exemplu, pentru prima cerință va avea conținutul:

```
neindependente
neindependente
independente
independente
neindependente
neindependente
neindependente
independente
```

4 Restricţii şi precizări

Temele trebuie să fie încărcate pe vmchecker. **NU** se acceptă teme trimise pe e-mail sau altfel decât prin intermediul vmchecker-ului.

O rezolvare constă într-o arhivă de tip **zip** care conţine toate fişierele sursă alături de un **Makefile**, ce va fi folosit pentru compilare, şi un fişier **README**, în care se vor preciza detaliile implementării.

Makefile-ul trebuie să aibă obligatoriu regulile pentru **build** şi **clean**. Regula `build` trebuie să aibă ca efect compilarea surselor şi crearea binarului **bnet**.

Programul vostru va primi, ca argumente în linia de comandă, o opţiune, pentru fiecare cerinţă în parte, în felul următor:

pentru cerința 1: `./bnet -c1`

pentru cerința 2: `./bnet -c2`

5 Punctaj

Cerinta	Punctaj
Cerinţa 1	30 de puncte
Cerinţa 2	60 de puncte
Codying style, README, warning-uri	10 puncte

Atenţie!

Orice rezolvare care nu conţine structurile de date specificate nu este punctată.
Temele vor fi punctate doar pentru testele care sunt trecute pe vmchecker.
Nu lăsaţi warning-urile nerezolvate, deoarece veţi fi depunctaţi.
Dealocaţi toată memoria alocată pentru reţinerea informaţiilor, deoarece se vor depuncta pierderile de memorie.
Tema este individuală! Toate soluţiile trimise vor fi verificate, folosind o unealtă pentru detectarea plagiatului.