Inteligență Artificială - Tema 2
Inferență exactă în Rețele Bayesiene

Tudor Berariu, Alexandru Sorici

November 2018

1 Scopul temei și enunțul pe scurt

Obiectivul acestei teme îl reprezintă înțelegerea și implementarea algoritmului de *propagare a convingerilor* (eng. *belief propagation*) pentru inferență în modele grafice probabilistice, rezumându-ne la aplicarea acestuia în cazul rețelelor bayesiene.

Problema inferenței presupune calculul unor distribuții marginale de probabilități $P(\mathbf{X} \mid \mathbf{Z} = \mathbf{z})$. Algoritmul de *propagare a convingerilor* simplifică problema prin manipularea unui *arbore de clici* în care fiecare nod va conține în final distribuția de probabilitate comună a unei submulțimi de variabile. Tema are două părți: construirea unui arbore de clici plecând de la graful orientat aciclic al unei rețele Bayesiene (Secțiunea 2) și propagarea convingerilor pentru inferență presupunând construit acel arbore (Secțiunea 3).

2 Obținerea arborelui de clici

Prima parte a temei o constituie implementarea unui algoritm ce construiește un arbore de clici plecând de la graful orientat aciclic al rețelei Bayesiene. Fie $\mathcal{G}$ acest graf inițial și $\mathcal{X}$ mulțimea variabilelor aleatoare.

1. Se construiește graful neorientat $\mathcal{U}$ transformând arcele din $\mathcal{G}$ în muchii.

2. Se construiește graful $\mathcal{H}$ obținut prin moralizarea grafului $\mathcal{U}$. Asta înseamnă că se adaugă muchii între oricare doi părinți ai unei variabile din $\mathcal{G}$.

3. Se construiește graful cordal $\mathcal{H}^*$ adăungând muchii în $\mathcal{H}$ astfel încât să nu existe cicluri elementare de lungime mai mare decât 3 care să nu aibă o coardă. Algoritmul de propagare a convingerilor este mai eficient dacă se aplică unor clici mai mici. De aceea este important să adăugăm cât mai puține muchii în etapa de triangulare. Problema de a găsi numărul minim de muchii de adăugat pentru a obține un graf cordal este NP-dură, dar există euristici puternice ce pot fi folosite. Euristica recomandată este următoarea: se elimină pe rând nodurile adăugându-se muchii între toți vecinii nodului eliminat. La fiecare pas se alege nodul în urmă eliminării căruia s-ar adăuga numărul minim de muchii. Mai multe în exemplul de la final sau în secțiunea 3.1 aici[1].

4. Se construiește graful de clici $\mathcal{C}$ extrăgându-se clicile maximale din $\mathcal{H}^*$. Pentru asta se poate aplica algoritmul Bron-Kerbosh[2]. Fiecare nod din $\mathcal{C}$ va fi asociat unei clici maximale din $\mathcal{H}^*$, deci va avea asociată o submulțime a variabilelor aleatoare. Muchiile din $\mathcal{C}$ vor fi etichetate cu variabilele comune ale nodurilor și vor avea ponderi date de numărul acestora.

5. Se extrage din $\mathcal{C}$ un arbore de acoperire de cost maxim $\mathcal{T}$. Pentru pasul acesta se poate aplica algoritmul lui Krushkal [3].

6. Se iau tabelele de probabilități condiționate ale rețelei Bayesiene și se transformă în factori. Fiecare factor este asociat unui nod din $\mathcal{T}$ care conține toate variabilele factorului. După ce fiecare factor inițial (câte unul pentru fiecare variabilă din $\mathcal{X}$), este asociat unui nod, se face produsul tuturor factorilor unui nod pentru a se obține un factor unic pentru fiecare nod din $\mathcal{T}$.

[1] `http://staff.utia.cas.cz/vomlel/55900566.pdf`
[2] `https://en.wikipedia.org/wiki/Bron%E2%80%93Kerbosch_algorithm`
[3] `https://en.wikipedia.org/wiki/Kruskal%27s_algorithm`

3 Propagarea convingerilor

Algoritmul de propagare a convingerilor pornește de la un graf de clici $\mathcal{T}$ și obține pentru fiecare nod un factor care este proporțional cu distribuția de probabilitate a variabilelor din nod. Obținând asta, se calculează ușor probabilități marginale. Pașii pentru propagarea convingerilor în arbore pentru a calcula apoi $P(\mathbf{X} \mid \mathbf{Z} = \mathbf{z})$ sunt următorii.

1. Se alege o rădăcină în $\mathcal{T}$. Făcând asta se creează implicit toate relațiile părinte-copil dintre noduri adiacente în $\mathcal{T}$.

2. Se reduc factorii din fiecare nod la intrările ce corespund observației $\mathbf{Z} = \mathbf{z}$. În acest moment fiecare nod are un factor $\phi_u^0 : \forall u \in \mathcal{T}$.

3. Se propagă mesaje dinspre frunze către rădăcină. Fiecare nod așteaptă mesajele de la copii lui. Aceste mesaje sunt, la rândul lor factori. Odată ce a primit mesaje de la toții copiii săi, un nod își actualizează factorul înmulțindu-l cu toate mesajele primite.

$$\phi_u^1 = \phi_u^0 \times \prod_{(c \to u) \in \mathcal{T}} m_{c \to u}$$

   Apoi proiectează factorul abia calculat asupra valorilor și trimite acest mesaj părintelui. Procesul de proiecție presupune eliminarea prin însumare a tuturor celorlalte variabile. Presupunând p părintele unui nod u:

$$m_{u \to p} = Proj_{u \cap p}\left(\phi_u^1\right)$$

   Procesul pornește de la frunze (acestea nu așteaptă niciun mesaj) și se oprește la rădăcină (care nu trebui să trimită mesaj mai departe). În acest moment fiecare nod are un set de factori actualizați $\phi_u^1 : \forall u \in \mathcal{T}$.

4. Se propagă mesaje dinspre rădăcină către frunze. Fiecare nod așteaptă un mesaj de la părintele lui și trimite câte un mesaj fiecărui copil. După ce primește un mesaj de la părinte, fiecare nod își calculează convingerile finale ϕ_u:

$$\phi_u = \phi_u^1 \cdot m_{p \to u}$$

   apoi trimite câte un mesaj fiecărui copil. Mesajul este calculat prin împărțirea convingerilor finale la mesajul primit anterior de la acel copil, rezultatul fiind, desigur, proiectat pe mulțimea de variabile comune ale celor două clici (noduri).

$$m_{u \to c} = Proj_{u \cap c}\left(\phi_u / m_{c \to u}\right)$$

   Procesul se termină atunci când toate frunzele au primit mesaj de la părinte și au calculat convingrile finale.

Factorul final ϕ_u al unui nod u ce corespunde unei clici $\{X_1, X_2, \dots X_M\}$ este proporțional cu distribuția $P(X_1, X_2, \dots X_M \mid \mathbf{Z} = \mathbf{z})$. Distribuția aceasta se obține ușor normalizând factorul. Prin marginalizare se calculează apoi distribuția unei variabile anume.

O prezentare succintă a conceptului de factor găsiți aici[4].

4 Cerințe

Cerința 1. Dându-se o rețea bayesiană să se construiască arborele de clici urmând procesul descris în Secțiunea 2.

Cerința 2. Să se calculeze probabilități marginale $P(\mathbf{X} = \mathbf{x} \mid \mathbf{Z} = \mathbf{z})$ utilizând algoritmul de propagare a convingerilor descris în Secțiunea 2. Este garantat că toate variabilele din $\mathbf{X}$ se vor găsi împreună într-una dintre clicile obținute.

BONUS. Să se calculeze probabilități marginale pentru submulțimi de variabile $\mathbf{X} \subseteq \mathcal{X}$ (care pot fi sau nu în aceeași clică). Pentru aceasta trebuie aplicat un algoritm de eliminare a variabilelor pe un subarbore $\mathcal{T}' \subseteq \mathcal{T}$ care să acopere toate variabilele necesare. Mai multe aici: `http://pgm.stanford.edu/Algs/page-371.pdf`.

[4] `https://www.coursera.org/lecture/probabilistic-graphical-models/factors-tEZ6S`

Observație Toate variabilele sunt binare. Puteți optimiza implementarea ținând cont de asta.

Observație Citiți toate detaliile din fișier pentru a putea testa și cu alte rețele sau interogări în momentul prezentării temei.

Observație În fișierul dat, primele 15 interogări corespund cerinței 2, iar ultimele 5 cerinței BONUS.

5 Fișierul de intrare

Fișierul de intrare are pe prima linie două valori întregi N, numărul de variabile din rețeaua bayesiană, și M, numărul de inferențe de calculat.

Fiecare dintre următoarele N rânduri conține trei secțiuni separate de simbolul `;`. Prima secțiune reprezintă numele variabilei. A doua secțiune reprezintă numele părinților, iar ce de-a treia tabela de probabilități condiționate. De exemplu, linia

```
A ; ; 0.1
```

ar corespunde unei variabile A fără părinți cu probabilitatea $P(A = 1) = 0.1$. Un alt exemplu: linia

```
C ; A B ; 0.2 0.4 0.5 0.3
```

ar corespunde unei variabile C ai cărei părinți sunt A și B cu tabela de probabilități: $P(C = 1 \mid A = 0, B = 0) = 0.2$, $P(C = 1 \mid A = 0, B = 1) = 0.4$, $P(C = 1 \mid A = 1, B = 0) = 0.5$ și $P(C = 1 \mid A = 1, B = 1) = 0.3$.

Următoarele M rânduri conțin câte două secțiune separate de simbolul `|` reprezentând o probabilitate condiționată de tipul $P(\mathbf{X} = \mathbf{x} \mid \mathbf{Z} = \mathbf{z})$. De exemplu, $P(C = 1, G = 0 \mid A = 0, F = 1)$ ar fi reprezentată prin linia următoare

```
C=1, G=0 | A=0, F=1
```

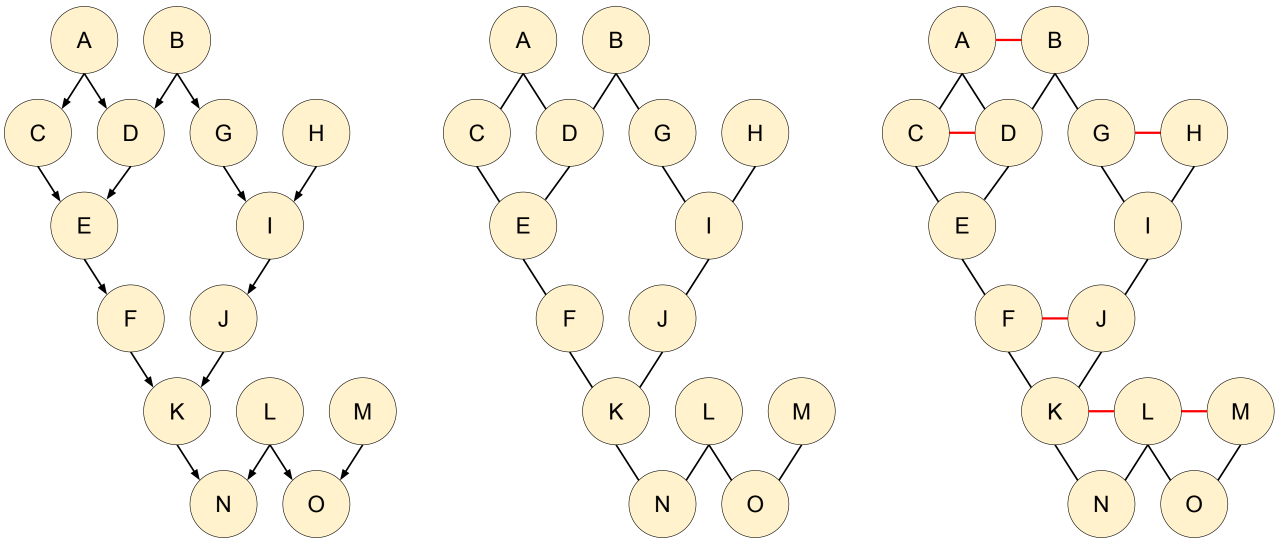
Fișierul conține pe ultimele M linii valorile corecte pentru cele M mase de proabilitate cerute.

6 Exemplu

Urmăriți Figurile 1, 2, 3 și 4 pentru obținerea arborelui de clici din fișierul de test. Acesta nu este singurul arbore de clici la care se poate ajunge. Procesul depinde de mai multe alegeri aleatoare: nodul eliminat la un pas dintre mai multe cu același cost în procesul de triangulare; ordonarea muchiilor cu același cost la găsirea arborelui de acoperire de cost maxim.

7 Alte observații

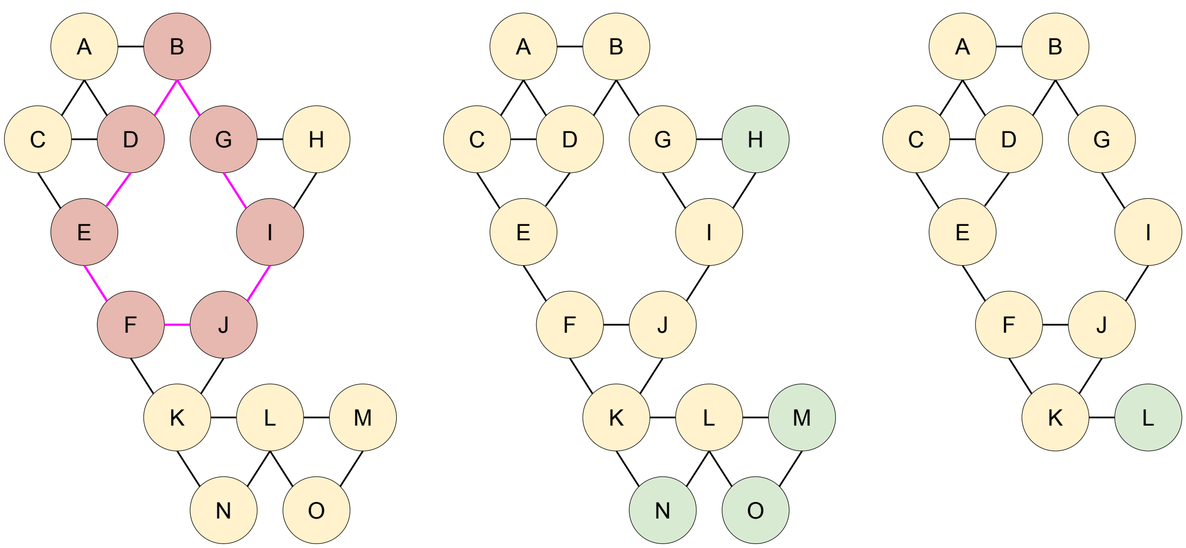
Documentul nu a discutat despre proprietatea pe care trebuie să o aibă arborele de clici pentru a putea aplica propagarea convingerilor, *running intersection property*, pentru a nu complica mai mult enunțul temei. Urmând pașii descriși în Secțiunea 2 și selectând arborele de acoperire de cost maxim din graful de clici este garantată obținerea unui arbore cu această proprietate.



(a) Rețeaua Bayesiană reprezentată printr-un graf orientat aciclic

(b) Graful neorientat obținut prin transformarea arcelor în muchii

(c) Graful moralizat

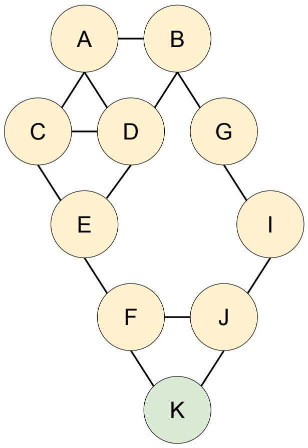


(d) Graful nu este cordal. Există cicluri precum cel marcat $B-G-I-J-F-E-D$ care nu au coarde.

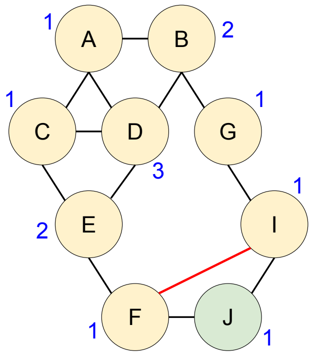
(e) Există noduri care au toți vecinii deja conectați. Pot fi eliminate primele.

(f) Nodul L poate fi eliminat fără a adăuga muchii suplimentare

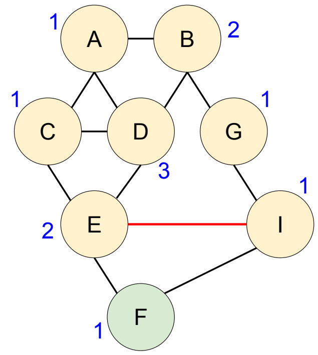
Figura 1: Moralizarea grafului inițial și primii pași din algoritmul de triangulare.



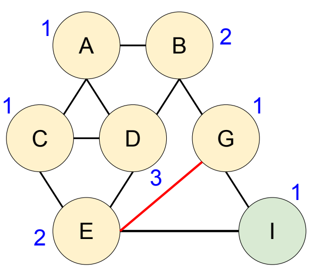
(a) Nodul K poate fi eliminat fără a adăuga muchii suplimentare.



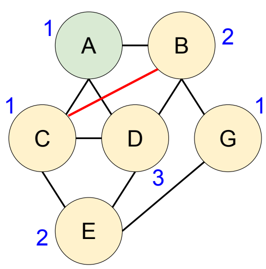
(b) Nu mai există noduri ce pot fi eliminate fără a adăuga o altă muchie. Dintre cele care au cost 1 alegem la întâmplare nodul J, drept urmare muchia $F-I$ va face parte din graful final.



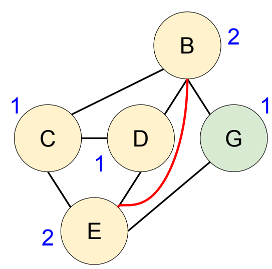
(c) Se elimină nodul F, se adaugă muchia $E-I$.



(d) Se elimină nodul I, se adaugă muchia $E-G$.

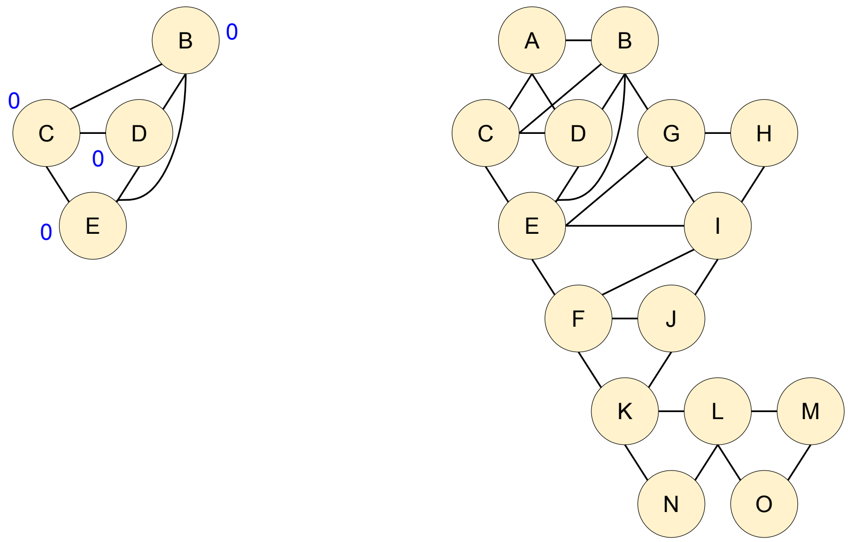


(e) Se elimină nodul A, se adaugă muchia $B-C$.



(f) Se elimină nodul G, se adaugă muchia $B-E$. Eliminarea lui nodului D sau a nodului C ar fi dus la adăugarea aceleiași muchii.

Figura 2: Pașii următori din procesul de triangulare.



(a) Graful rămas este o clică. Nu se mai pot adăuga muchii. S-a terminat triangularea.

(b) Graful cordal final.

Figura 3: Obținerea grafului final din care vor fi extrase clicile maximale.

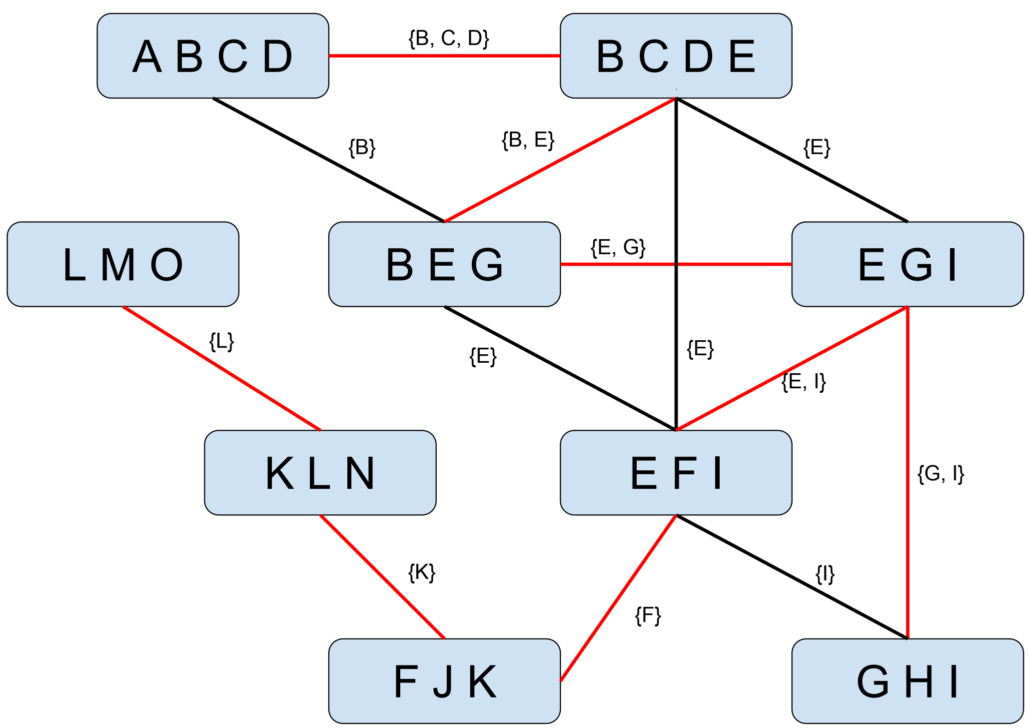


Figura 4: Graful de clici, iar cu roșu un subarbore de acoperire de cost maxim.