

Rapport TP RCR

TP N°5 : Réseaux Bayésiens

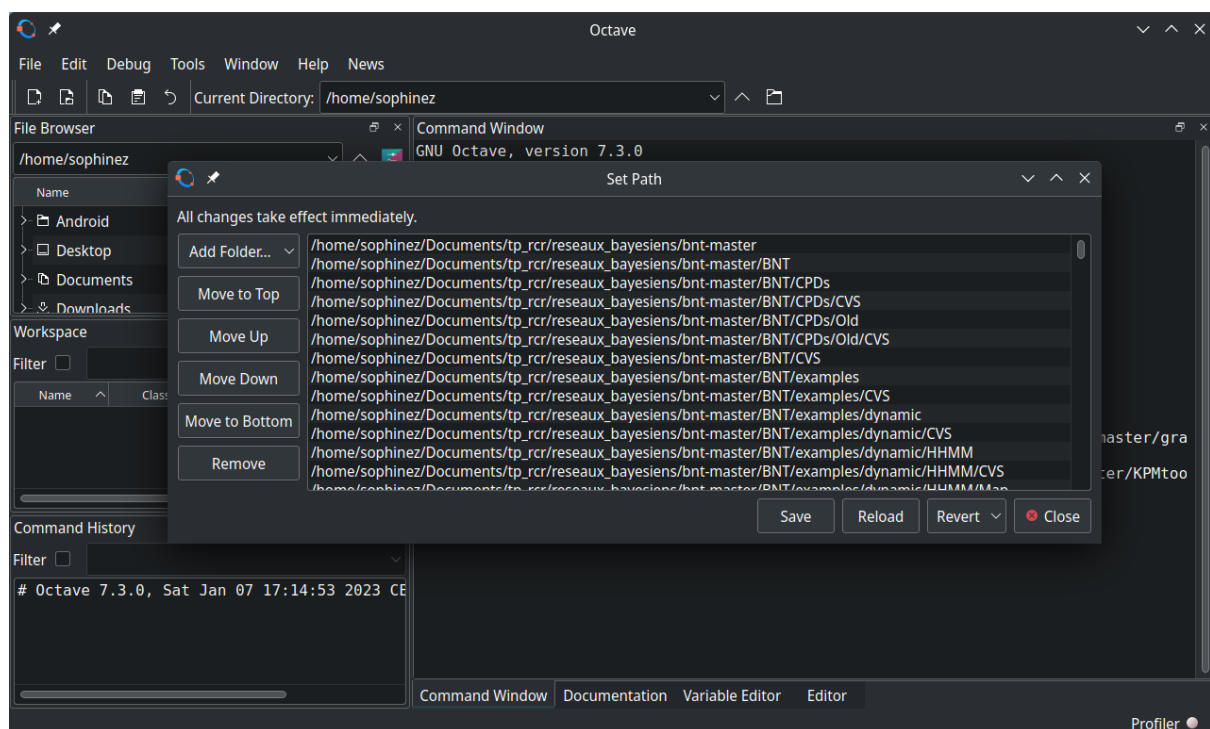
Travail réalisé par : Sophinez Azouaou 181833011664
Samy Aghiles Aouabed
Section M1. IV,
Groupe 1

Introduction :

Dans ce TP nous allons exploiter la toolbox BNT sur des graphes à connexion multiple et polyarbres.

Installation :

Pour la réalisation de ce TP, le logiciel Octave open SOURCE a été utilisé pour l'exploitation de la toolbox BNT.



Après avoir téléchargé le fichier compressé de la toolbox depuis le [repo github](#) officiel, nous l'avons décompressé, importé et ajouté à Octave comme illustré ci-dessus.

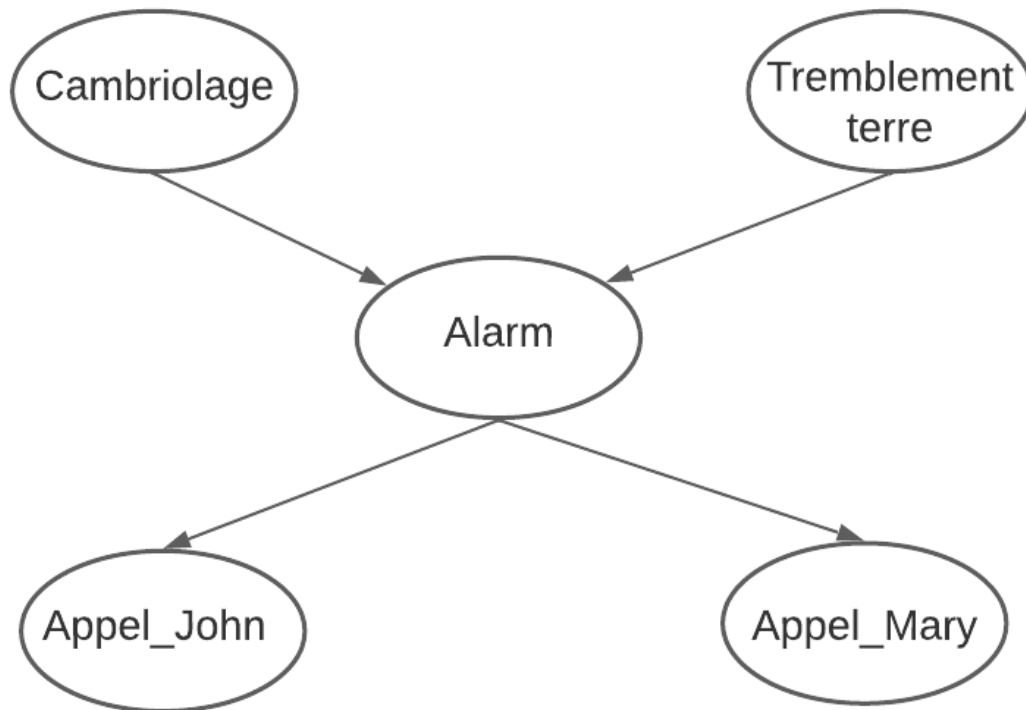
Toolbox BNT : Bayesian Networks Toolbox

1. Polyarbres :

Rappel : Un polyarbre est un graphe acyclique orienté (DAG) pour lequel il n'y a qu'un chemin entre toute paire de nœuds.

Pour cette partie, nous prenons l'exemple du Burglar Alarm :

1. 1. Composante graphique :



1. 2 Composante de calcul :

La distribution des probabilités conditionnelles pour ce modèle est comme suit :

Nous prenons nos variables comme suit :

- Cambriolage : **C**
- Tremblement Terre : **T**
- Alarme : **A**
- Appel_John : **J**
- Appel_Mary : **M**

P(C = F)	P(C = V)
0.999	0.001

P(T = F)	P(T = V)
0.998	0.002

C	T	$P(A = V \mid C, T)$	$P(A = F \mid C, T)$
Faux	Faux	0.001	0.999
Vrai	Faux	0.94	0.06
Faux	Vrai	0.29	0.71
Vrai	Vrai	0.95	0.05

A	$P(J = V \mid A)$	$P(J = F \mid A)$
Faux	0.05	0.95
Vrai	0.9	0.1

A	$P(M = V \mid A)$	$P(M = F \mid A)$
Faux	0.01	0.99
Vrai	0.7	0.3

BNT exploitation :

1. Spécification de la structure de l'arbre sur Octave :

```
>> nodes_nbr = 5;
>> dag = zeros(nodes_nbr, nodes_nbr);
>> c = 1; t = 2; a = 3; j = 4; m = 5;
>> dag(c, a) = 1;
>> dag(t, a) = 1;
>> dag(a, [j, m]);
>>
```

2. Création de l'arbre bayésien :

```
>> discrete_nodes = 1:nodes_nbr;
>> node_sizes = 2*ones(1,nodes_nbr);
>> bayesian_net = mk_bnet(dag, node_sizes);
>>
```

3. Insertion des paramètres :

```
>> bayesian_net.CPD{c} = tabular_CPD(bayesian_net, c, [ 0.999 0.001 ]);
>> bayesian_net.CPD{t} = tabular_CPD(bayesian_net, t, [ 0.998 0.002 ]);
>> bayesian_net.CPD{a} = tabular_CPD(bayesian_net, a, [0.999 0.06 0.71 0.05 0.001 0.94 0.29 0.95]);
warning: Matlab-style short-circuit operation performed for operator &
warning: called from
    mk_stochastic at line 12 column 18
    tabular_CPD at line 62 column 9

>> bayesian_net.CPD{j} = tabular_CPD(bayesian_net, j, [ 0.95 0.1 0.005 0.9 ]);
>> bayesian_net.CPD{m} = tabular_CPD(bayesian_net, m, [ 0.99 0.3 0.01 0.7 ]);
>> |
```

4. Inference :

La toolbox BNT offre différents “engins” pour l’inférence, nous créons un engin qui utilise la méthode de l’arbre de jonction avec la commande suivante :

```
engine = jtree_inf_engine(bayesian_net);
```

Calculons la probabilité : **$P(A \mid C=\text{Faux}, T=\text{Vrai})$**

```
>> evidence = cell(1,nodes_nbr);
>> evidence{c} = 2;
>> evidence{t} = 1;
```

Les résultats obtenus sont :

```
ans =

    0.057455
    0.942545

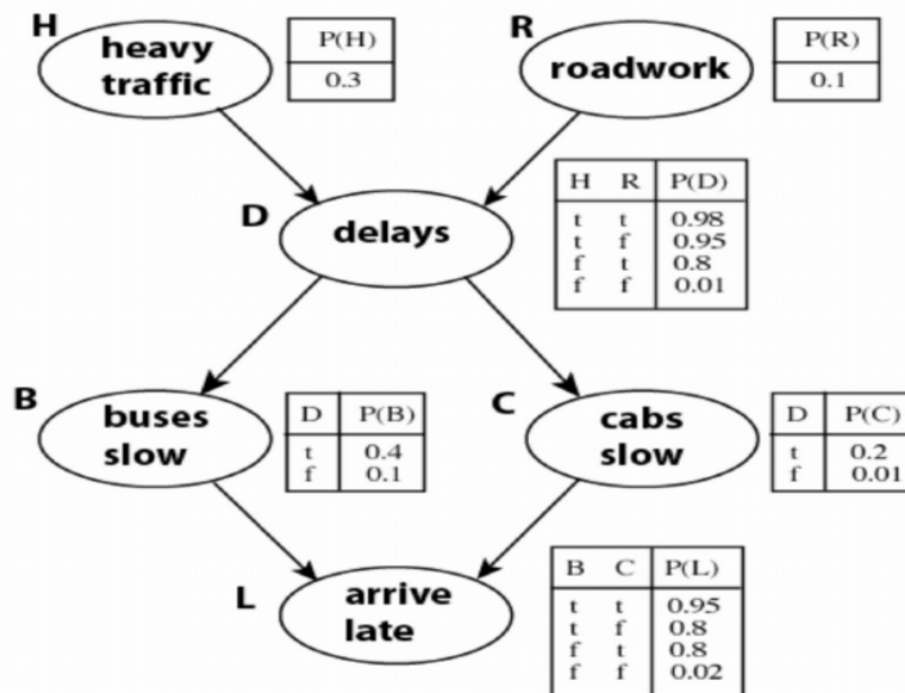
>>
```

$P(A= \text{Faux} \mid C =\text{Faux}, T=\text{Vrai}) = 0.05$

$P(A= \text{Vrai} \mid C =\text{Faux}, T=\text{Vrai}) = 0.94$

2. Arbres à connexions multiples :

Dans cette partie, nous prenons un exemple trouvé sur internet :



```

Command Window
>> n = 6;
>> dag = zeros(n, n);
>> heavy_traffic = 1; roadwork = 2; delays = 3; buses_slow = 4; cabs_slow = 5; arrive_late = 6;
>> dag(heavy_traffic, delays) = 1;
>> dag(roadwork, delays) = 1;
>> dag(delays, [buses_slow, cabs_slow]) = 1;
>> dag(buses_slow, arrive_late) = 1;
>> dag(cabs_slow, arrive_late) = 1;
>> discrete_nodes = 1:n;
>> node_sizes = 2*ones(1,n);
>> bnet = mk_bnet(dag, node_sizes);
>>

>> bnet.CPD{heavy_traffic} = tabular_CPD(bnet, heavy_traffic, [0.7 0.3]);
>> bnet.CPD{roadwork} = tabular_CPD(bnet, roadwork, [0.9 0.1]);
>> bnet.CPD{delays} = tabular_CPD(bnet, delays, [0.99 0.05 0.2 0.02 0.01 0.95 0.8 0.9]);
warning: Matlab-style short-circuit operation performed for operator &
warning: called from
    mk_stochastic at line 12 column 18
    tabular_CPD at line 62 column 9

>> bnet.CPD{buses_slow} = tabular_CPD(bnet, buses_slow, [0.9 0.6 0.1 0.4]);
>> bnet.CPD{cabs_slow} = tabular_CPD(bnet, cabs_slow, [0.99 0.8 0.01 0.2]);
>> bnet.CPD{arrive_late} = tabular_CPD(bnet, arrive_late, [0.98 0.2 0.2 0.05 0.02 0.8 0.8 0.95]);
warning: Matlab-style short-circuit operation performed for operator &
warning: called from
    mk_stochastic at line 12 column 18
    tabular_CPD at line 62 column 9

>>

```

→ Pour la partie inférence, calculons

$P(\text{delays} \mid \text{heavy_traffic} = \text{Faux}, \text{roadwork} = \text{Vrai})$

```

>> engine = jtree_inf_engine(bnet);
>> evidence = cell(1, n);
>> evidence{heavy_traffic} = 2;
>> evidence{roadwork} = 1;
>> [engine, loglik] = enter_evidence(engine, evidence);

```

```
>> marg = marginal_nodes(engine, delays);  
>> marg.T  
ans =  
  
    0.050000  
    0.950000  
  
>> |
```

Les résultats sont donc interprétés ainsi :

$P(\text{delays} = \text{Vrai} \mid \text{heavy_traffic} = \text{Faux}, \text{roadwork} = \text{Vrai}) = 0.95$

$P(\text{delays} = \text{Faux} \mid \text{heavy_traffic} = \text{Faux}, \text{roadwork} = \text{Vrai}) = 0.05$