

Contents

1	Introduction	3
2	Réalisation	4
2.1	L'outil cygwin	4
2.2	Étape 1:	4
2.2.1	Utilisation du prod1vid.m	4
2.2.2	Réglage de paramètres	5
2.3	Étape 2	5
2.3.1	Explication + observation	5
2.4	Étape 3	6
2.4.1	Génération automatique aléatoire des Graphes	6
2.5	Résultats	6
2.6	Observation pour chaque type	6
2.7	Observation et comparaison entre les différents types	6

List of Figures

Chapter 1

Introduction

bla bla for tp and le but of ir

Chapter 2

Réalisation

2.1 L'outil cygwin

why did we use it and all

2.2 Étape 1:

2.2.1 Utilisation du prod1vid.m

principalement prod1vid construit un réseau causal probabiliste basé sur le produit tel que les connexions entre les nœuds sont aléatoires , ainsi que les valeurs initiales attribués à la variable d'intérêt et l'évidence . Pour exécuter le programme il faut:

- sur Matlab taper : prod1vid

ce que le programme offre en sortie est environnement ou on peut voir toutes les variables et le graphe (matrice) crée. on peut alors afficher :

- la variable d'intérêt sachant l'évidence
- temps de la propagation
- type de graphe (multi-connected (multi-connectés) ou polytree (polyarbre))

Fonctionnement du programme

Après avoir étudié le programme on a pu résumer son fonctionnement dans les étapes qui suivent :

1. Initialisation du nombre de parents max globale et nombre de noeuds du graphe à construire
2. Création de liens de façon aléatoire entre les noeuds.
3. Utilisation de processus de fixation après la création aléatoire afin d'éviter les noeuds isolés et sous graphes isolés (les inconvénients de l'aléatoire)
4. Prise de considération des domaines des variables (représentés par les noeuds) cas binaire etc ...

5. Génération de la distribution aléatoire initiale du graphe crée (de possibilité initiales).
6. génération aléatoire d'une évidence : une évidence est une information nouvelle qui viens et à qui on aimerait calculer l'influence qu'elle aura sur la variable d'intérêt (évidente est comme une condition).
7. Détermination si le graphe est polytree ou multi-connected
8. Lancement de la propagation (algorithme de propagation)

Concernant **Prodevid2** c'est le même fonctionnement à part qu'on a droit à deux évidences donc deux informations vont influencer notre réseau , en théorie on peut penser que la propagation des deux évidences prendra plus de temps que celle d'une seule , on testera ce cas dans ce qui suit .

2.2.2 Réglage de paramètres

Jeu de test

On a choisi de fixer le nombre de noeuds à :10
 et le nombre de parent max à :2
 ce qui est censé nous donnée un polytree .

Affichage

affichage du temps de propagation each time à chaque execution du prog et du degré de possibilité (basé sur le produit) de la variable d'interet

Explication et observation

une explication sur ce que t'as compris or any observation (ça serait bien de parler comment fait l'algo generalement et commenter tes resultat)

2.3 Etape 2

2.3.1 Explication + observation

i dunno what u can put here

2.4 Etape 3

2.4.1 Génération automatique aléatoire des Graphes

Génération des Polytree(youpai)

Génération des Multiconnected

Génération des simplement connected

2.5 Résultats

NbrNœuds/NbrParents	Temps Propagation	Temps Inférence	Degrés Possibilité
25 Nœuds / 1 Parents	0.156598 sec	171601 milisec	0.049787
50 Nœuds / 1 Parents	0.282039 sec	218400 milisec	1
15 Nœuds / 3 Parents	0.052538 sec	156000 milisec	0.049787
25 Nœuds / 4 Parents	0.081438 sec	156000 milisec	1
25 Nœuds / 7 Parents	0.078056 sec	249600 milisec	1
25 Nœuds / 10 Parents	0 sec	0 milisec	0
30 Nœuds / 4 Parents	0 sec	0 milisec	1

2.6 Observation pour chaque type

2.7 Observation et comparaison entre les différents types