Intelligence Artificielle

Représentation des connaissances

Emmanuel ADAM

Université Polytechnique des Hauts-De-France

UPHF/INSA HdF

Introduction

Représentation des connaissances

- Etude en Philosophie, Logique, Linguistique, Psychologie cognitive, Intelligence Artificielle
- Informatique : théorie et traitement de l'information (Le Robert)
  - Nécessité de représenter la connaissance
  - But : manipulation par " systèmes experts "
  - Objectif : faciliter, aider la décision
- Connaître = Mémoriser + Raisonner
- Représenter = Formaliser + Inférer

Typologie de la connaissance

Les différents types de connaissances

De définition : toujours vraie
Un triangle est un polygone ayant exactement 3 côtés

Evolutive/atemporelle : peut être modifiée
Anne est étudiante en Master 1 TNSID à l'UVHC

Incertaine/certaine :
La Lune provient d'une collision de la Terre

Floue/précise : évaluation difficile
Les lendemains de fêtes ne sont pas très productifs

Typique/universelle : peut être contredit
Habituellement, le prof arrive en retard

Ambiguë : plusieurs significations
Nikos savait que que Lisandro allait gagner.
savoir / se douter ?

Problèmes de représentation

Pas de formalisme idéal

- Modalités : je crois que, je pense que, il est probable que, . . .
- Evolutivité : les connaissances changent
- Typicalité et partage de propriété
- Connaissances incomplètes
- . . .

$\rightsquigarrow$ pas de formalisme idéal !

Propriété de représentation

Objectif de la représentation

- Adéquation représentationnelle
- Adéquation et efficacité inférentielle
- Efficacité acquisitionnelle et extensibilité
- Simplicité (utilisable par un non informaticien)
- Connaissance explicite
- ...

Famille de connaissances

Savoir et Savoir-Faire

- Représentation déclarative
  - Les connaissances n'ont pas d'ordre
  - Connaissances indépendantes ou liées
  - Pertinence fixée a priori
  - $\bigoplus$ modularité, connaissance stockée une seule fois
- Représentation procédurale
  - Les connaissances ont un caractère opératoire
  - Pertinence définie par un programme qui traite les connaissances
  - $\bigoplus$ facilité de codage, représentation des connaissances sur des opérations

Logique des prédicats

Exemple de représentation

La logique des prédicats (*cf. cours I.A. M1 TNSI*) est la représentation la plus courante des faits et règles

- MV est une sonde : $sonde(mv)$
- Toutes les sondes volent : $\forall x, sonde(x) \rightarrow vole(x)$
- Toutes les sondes, qui ne sont pas en panne, volent : $\forall x, sonde(x) \wedge \neg panne(x) \rightarrow vole(x)$

Avantage : de nombreux principe de résolution, de détection d'inconsistance (clauses de Horn, ...)

Outil pour logique des prédicats

Exemple d'outil

Le langage Prolog permet (entres autres) de raisonner sur base de prédicats

- MV est une sonde : $sonde(mv)$
  `sonde(mv).`
- Toutes les sondes volent : $\forall x, sonde(x) \rightarrow vole(x)$
  `vole(X) :- sonde(X).`
  On peut poser la question, qui vole ? `?  vole(X)`
  Reponse : `X = mv`
- Toutes les sondes, qui ne sont pas en panne, volent :
  $\forall x, sonde(x) \wedge \neg panne(x) \rightarrow vole(x)$
  `vole(X) :- sonde(X), not(enPanne(X)).`

Exemples de représentation

Prédicats n-aires

- Prédicat unaire :
  - MV est une sonde : $sonde(mv)$
- Prédicat binaire :
  - MV est une (is a) sonde : $is\_a(mv, sonde)$
- Prédicat ternaire :
  - MV voyage de la Terre vers Mars : $voyage(mv, terre, mars)$

Prédicats binaires

Du ternaire au binaire : exemple

- les types des attributs du prédicat *voyage* sont : $Voyageur$, $Origine$, $Destination$: $Voyage(Voyageur, Origine, Destination)$
- pour le voyage sur Mars ($voyageMars(mv, terre, mars)$), le $Voyageur$ est mv, l'$Origine$ est la $terre$, la $Destination$ est $mars$
- on réécrit en :
  $voyageur(voyageMars, mv) \wedge origine(voyageMars, terre)$
  $\wedge destination(voyageMars, mars)$

Du n-aire au binaire : règle

- Soit un predicat : $nomPredicat(val_1, val_2, \ldots, val_n)$ où les types des valeurs $val_1, \ldots, val_n$ sont $typeV_1, \ldots, typeV_n$
- La transformation donne :
  $typeV_1(nomPredicat, val_1) \wedge \cdots \wedge typeV_n(nomPredicat, val_n)$

Exercices prédicats 1

Formalisation.1

a. François est chez-lui ou chez Julie.
b. Si François n'est pas chez-lui, il est chez Julie.
c. Vous pouvez déduire vos frais médicaux si votre revenu est inférieur à 18 000e et que vous avez plus de 70 ans.
d. Vous ne pouvez pas déduire vos frais médicaux si vous n'avez pas plus de 70 ans ou que votre revenu est inférieur à 18 000e.
e. Jean réussira son examen ou il n'est pas fort en logique.
f. Si Marie n'est pas fort en logique, Jean n'est pas forte non plus en logique et elle ne réussira pas son examen.
g. Jean et Marie réussiront leur examen s'ils sont forts en logique.
h. Neige en novembre, Noël en décembre.

Exercices prédicats 2

Formalisation.2

a. p sinon q
b. p à moins que q
c. p autrement q
d. Il suffit que p pour q
e. Il est nécessaire que p pour q
f. p seulement si q
g. p si q

Logique modale

Logique aléthique (aristotélicienne)

Notions de possibilités/nécessité

- Nécessaire (il est nécessaire d'avoir A) : $\Box A \equiv \neg\Diamond\neg A$
- Possible (il est possible d'avoir A) : $\Diamond A \equiv \neg\Box\neg A$
- Non Nécessaire (il n'est pas nécessaire d'avoir A) : $\neg\Box A \equiv \Diamond\neg A$
- Impossible (il est impossible d'avoir A) : $\neg\Diamond A \equiv \Box\neg A$

Logique modale

Logique épistémique

Notions de croyances

- Savoir que A : $\Box A \quad \equiv \quad \neg\Diamond\neg A$
- Croire que A (A est compatible avec les croyances) : $\Diamond A \quad \equiv \quad \neg\Box\neg A$
- Ne pas Savoir que A : $\neg\Box A \quad \equiv \quad \Diamond\neg A$
- Ne pas Croire que A : $\neg\Diamond A \quad \equiv \quad \Box\neg A$

Logique épistémique : exemples

- Tout ce que sait Jorah, Daenerys le sait
  $\forall x, \Box(jorah)x \rightarrow \Box(daenerys)x$
- Si quelqu'un sait qu'il gèle, alors il gèle
  $\exists x, \Box(x)geler \rightarrow geler$

Logique modale : schémas d'axiomes

Schémas d'axiomes

- axiome K : $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$: s'il est nécessaire que A implique B, alors s'il est nécessaire que A soit vrai alors il est nécessaire que B soit vrai
- axiome D : $\Box A \rightarrow \Diamond A$: S'il est nécessaire que A soit vrai alors il est possible que A soit vrai
- axiome T : $\Box A \rightarrow A$: S'il est nécessaire que A soit vrai alors A est vrai
- axiome 4 : $\Box A \rightarrow \Box\Box A$: S'il est nécessaire que A soit vrai alors il est nécessaire qu'il soit nécessaire que A soit vrai
- axiome B : $A \rightarrow \Box\Diamond A$: Si A est vrai, alors il est nécessaire qu'il soit possible que A soit vrai
- axiome 5 : $\Diamond A \rightarrow \Box\Diamond A$: Si A est possible, alors il est nécessaire qu'il soit possible que A soit vrai

Logique modale : les Systèmes

Schémas d'axiomes

règle d'inférence classique : $A, A \rightarrow B \vdash B$

règle d'inférence de nécessitation : $\Box A \vdash A$

- Système K : constitué de l'Axiome K
- Système D : constitué des Axiomes K & D
- Système T : constitué des Axiomes K & T
- Système S4 : constitué des Axiomes K & T & 4
- Système B : constitué des Axiomes K & T & B
- Système S5 : constitué des Axiomes K & T & 5

Logique modale : exercices

Traduire en logique modale

1. Stéphane croit qu'il gagnera les élections.
2. Delphine sait que tous les candidats croient qu'ils gagneront les élections.

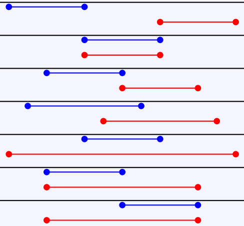
Logique temporelle

Intervalle, instant et événement

Différentes logiques.. Ici présentations de Logique de Allen. Nouveaux prédicats :

- $holds(p, t)$: p est vrai pendant l'intervalle t
- $occurs(e, t) \rightarrow Pt$: t apparaît suite à l'événement e implique que les préconditions nécessaires à t sont validées

before : $x < y$:	$(x_1^+ < x_2^-)$
equals : $x = y$:	$(x_1^- = x_2^-) \wedge (x_1^+ = x_2^+)$
meets : $x \text{ m } y$:	$(x_1^+ = x_2^-)$
overlaps : $x \text{ o } y$:	$(x_1^+ > x_2^-) \wedge (x_1^+ < x_2^+)$
during : $x \text{ d } y$:	$(x_1^- > x_2^-) \wedge (x_1^+ < x_2^+)$
starts : $x \text{ s } y$:	$(x_1^- = x_2^-) \wedge (x_1^+ < x_2^+)$
finishes : $x \text{ f } y$:	$(x_1^- > x_2^-) \wedge (x_1^+ = x_2^+)$



44 axiomes. Exemple : $(x < y) \wedge (y < z) \rightarrow (x < z)$

Logique temporelle : LTL

Base de LTL

- CTL (Computation Tree Logic, 1981)
- LTL (Linear Temporal Logic, 1977)
  - $\bigcirc\phi$ ou $X\ \phi$: ϕ doit être vrai au prochain état
  - $\Diamond\phi$ ou $F\ \phi$: ϕ sera vrai dans un des prochains états
  - $\Box\phi$ ou $G\ \phi$: ϕ est toujours vrai dans les prochains états
  - $\psi\ \mathcal{U}\ \phi$: ψ doit être vraie au moins jusqu'à ce que ϕ devienne vraie (ce quelle **devra** être dans le futur)

Exemple de LTL

- $G(msg \rightarrow F\ accuse)$: il est toujours vrai qu'un message est suivi d'un futur accusé de reception

Logique temporelle : LTL

Extension à LTL

- $\psi \; \mathcal{W} \; \phi$: ψ doit être vraie au moins jusqu'à ce que ϕ devienne vraie. Si ϕ ne le devient jamais, ψ doit le rester pour toujours.
- $\psi \; \mathcal{R} \; \phi$: ϕ doit être vraie tant que ψ est fausse. Donc ϕ restera pour toujours vraie si ψ ne le devient jamais.
- $\psi \; \mathcal{M} \; \phi$: ϕ doit être vraie tant que ψ fausse, ψ **devra** devenir vraie dans le futur.

Logique temporelle : LTL

Propagation de négation en LTL

- $\neg \bigcirc \phi \equiv \bigcirc \neg \phi$: il est faux que ϕ sera vraie au prochain état = au prochain état, ϕ sera faux
- $\neg \Box \phi \equiv \Diamond \neg \phi$: il est faux que ϕ sera toujours vraie dans le futur = il existe un futur état où ϕ sera faux
- $\neg \Diamond \phi \equiv \Box \neg \phi$: il est faux qu'il existe un futur état où ϕ sera vraie = ϕ sera toujours fausse
- $\neg(\psi \; \mathcal{U} \; \phi) \equiv (\neg \psi \; \mathcal{R} \; \neg \phi)$: il est faux que ψ est vrai jusqu'à ce que ϕ le soit= ϕ est faux tant que ψ est vraie
- $\neg(\psi \; \mathcal{R} \; \phi) \equiv (\neg \psi \; \mathcal{U} \; \neg \phi)$
- $\neg(\psi \; \mathcal{W} \; \phi) \equiv (\neg \psi \; \mathcal{M} \; \neg \phi)$
- $\neg(\psi \; \mathcal{M} \; \phi) \equiv (\neg \psi \; \mathcal{W} \; \neg \phi)$

Logique temporelle : LTL

Exemple de LTL

- $\mathit{afficherAccueil}\ \mathcal{U}\ \mathit{frappeEcran}$: on affiche le message d'accueil jusqu'à un tappotement de l'utilisateur
- on réalise la négation de la phrase : $\neg(\mathit{afficherAccueil}\ \mathcal{U}\ \mathit{frappeEcran}) \equiv (\neg \mathit{afficherAccueil}\ \mathcal{R}\ \neg \mathit{frappeEcran})$: on ne tappote pas l'écran tant que le message d'accueil est affiché

Logiques multivaluée, floue

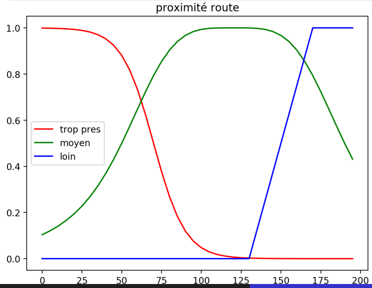
Valeurs de vérité

- Exemple sur un véhicule autonome détectant une personne sur un trottoir :
  - ‘Si une personne est trop près du bord du trottoir
    Alors je ralenti fortement’
  - ’Si un pre-ado est au centre du trottoir,
    Alors je ralentis fortement’
  - ’Si un post-ado est au centre du trottoir,
    Alors je ralentis normalement’
  - ’Si une personne est éloignée de la route,
    Alors je ralentis faiblement’
- Principe :
  1. les faits sont ’fuzzifiés’, transformés en probabilités d’appartenir à des classes
  2. les règles sont appliquées et fournissent des valeurs pour les conséquents
  3. les conséquents sont ’défuzzifiés’ pour déterminer leurs classes

Logiques floue

Fuzzification

- La "fuzzification" consiste à transformer les entrées (antécédents) en probabilités (sur $[0, 1]$) d'appartenance à des ensemble.
  Les fonctions de fuzzification peuvent être linéaires, par morceaux, sigmoïdes, ...
- Exemple : d'après la figure ci-dessous, un objet situé à 50cm est classé à 88% comme étant 'trop pres', à 50%comme étant moyennement près, et à 0% comme étant éloigné



La logique floue par l'exemple

Exemple de Fuzzification (!)

- Identifier la personne selon sa taille : pre-ado, post-ado
- au plus simple, on utilise une fonction par morceaux.
  taille moyenne à 12 ans $\approx 152cm \rightarrow$:
  $$preAdo(x) = \begin{cases} 1 & \text{si } x < 120 \\ (x-120)/40 & \text{si } x \leq 160 \\ 0 & \text{si } x > 160 \end{cases}$$
- la taille d'un "grand" est posée à partir de 130cm jusqu'à 170cm:
  $postAdo(x) = 0$ si $x < 130$; $(x-130)/40$ si $x \leq 170$; 1 si $x > 170$

Fuzzification des conséquents

- Les classes des conséquents doivent également être "fuzzifiées".
- Ici, on nomme 'freinage' le conséquent que l'on partage en 3 ensembles : léger, normal, fort.

La logique floue par l'exemple

Création des règles

- Une fois les antécédents et conséquents fuzzifiés, il est possible de créer les règles :
  $R_1 = bord_{.trop\ pres} \rightarrow frein_{.fort}$
  $R_2 = bord_{.centre} \ \wedge \ taille_{.pre\ ado} \rightarrow frein_{.fort}$
  $R_3 = bord_{.centre} \ \wedge \ taille_{.post\ ado} \rightarrow frein_{.normal}$
  $R_4 = bord_{.eloigne} \rightarrow frein_{.leger}$

Defuzzification !

- La défuzzification consiste à définir les classes pour les conséquents.
- Les règles activées fournissent des valeurs différentes pour un même conséquents
- $\rightarrow$ une aggrégation des valeurs est effectuée, selon une méthode (le max par exemple)

Logiques floue, traduire le ou, le et

Valeurs de vérité selon Zadeh

- Pour toutes les techniques, on a:
  - $\neg P \equiv (1 - P)$
- Min et Max de [Zadeh] :
  - $P \wedge Q \equiv min(P, Q)$
  - $P \vee Q \equiv max(P, Q)$

a	b	$\neg a$	$a \vee b$	$a \wedge b$	$a \vee \neg a$	$a \wedge \neg a$
0	0	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	1	0	1	0
1	1	0	1	1	1	0
0.8	0.3	0.2	0.8	0.3	0.8	0.2
0.3	0.7	0.7	0.7	0.3	0.7	0.3

Logiques floue, traduire le ou, le et

Valeurs de vérité selon Bandler

- Pour toutes les techniques, on a:
  - $\neg P \equiv (1 - P)$
- Produit et Somme algébrique [Bandler]:
  - $P \wedge Q \equiv (P \times Q)$
  - $P \vee Q \equiv ((P + Q) - (P \times Q))$

a	b	$\neg a$	$a \vee b$	$a \wedge b$	$a \vee \neg a$	$a \wedge \neg a$
0	0	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	1	0	1	0
1	1	0	1	1	1	0
0.8	0.3	0.2	0.86	0.24	0.84	0.16
0.3	0.4	0.7	0.58	0.12	0.79	0.21

Logiques floue, traduire le ou, le et

Valeurs de vérité selon Lukasiewicz

- Pour toutes les techniques, on a:
  - $\neg P \equiv (1 - P)$
- Sommes Bornés [Lukasiewicz]:
  - $P \wedge Q \equiv max(0, P + Q - 1)$
  - $P \vee Q \equiv min(1, P + Q)$

a	b	$\neg a$	$a \vee b$	$a \wedge b$	$a \vee \neg a$	$a \wedge \neg a$
0	0	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	1	0	1	0
1	1	0	1	1	1	0
0.8	0.3	0.2	1	0.1	1	0
0.3	0.4	0.7	0.7	0	1	0

La logique floue : exemple en Python I

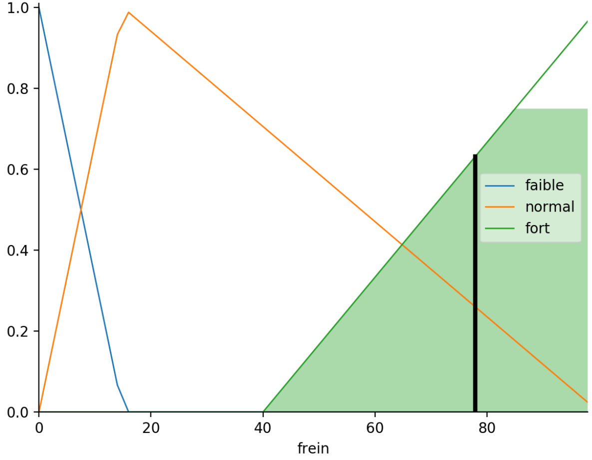
```
###les imports :
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt

#### Antecedent et Consequent
taille = ctrl.Antecedent(np.arange(0,200,5), 'taille')
bord = ctrl.Antecedent(np.arange(0,200,5), 'bord')
frein = ctrl.Consequent(np.arange(0,100,2), 'frein')
##decoupe des antecedents
##decoupe de taille en pre(ado),post(ado)
preado_lo = fuzz.trapmf(taille.universe, [0,0,120,160])
preado_hi = fuzz.trapmf(taille.universe, [130,170,200,200])
taille['pre'] = preado_lo
taille['post'] = preado_hi

##decoupe de bord en troppres,centre, loin
bord_lo = fuzz.sigmf(bord.universe, 70,-0.1)
bord_md = fuzz.gbellmf(bord.universe, 70, 2, 120)
bord_hi = fuzz.trapmf(bord.universe, [130,170,200,200])
bord['troppres'] = bord_lo
bord['centre'] = bord_md
bord['loin'] = bord_hi


##decoupe de frein en faible, normal, fort
frein['faible'] = fuzz.trimf(frein.universe,[0,0,15])
```

La logique floue : exemple en Python II

```
frein['normal'] = fuzz.trimf(frein.universe,[0,15,100])
frein['fort'] = fuzz.trimf(frein.universe,[40,100,100])

##definition des regles
rul1 = ctrl.Rule( bord['troppres'], frein['fort'])
rul2 = ctrl.Rule(taille['pre'] & bord['centre'], frein['fort'])
rul3 = ctrl.Rule(taille['post'] & bord['centre'], frein['normal'])
rul4 = ctrl.Rule(bord['loin'], frein['faible'])

##activation du systeme
frein_ctrl=ctrl.ControlSystem([rul1,rul2,rul3,rul4])
freinage = ctrl.ControlSystemSimulation(frein_ctrl)

##definition des entrees
freinage.input['taille'] = 130
freinage.input['bord'] = 180
##lancement du calcul
freinage.compute()
##affichage du consequent
print(freinage.output['frein'])
##affichage graphique
frein.view(sim=freinage)
```

Logiques floue

Degré de freinage pour une personne de 130cm à 80cm du bord



Réseaux sémantiques

Représentation graphique

Représentation graphique de prédicats sous forme de réseaux :

- nœud = variable
- arc = prédicat

Utilisation d'algorithmes de parcours de réseaux pour trouver les relations entre termes.

Web sémantique

Limitations

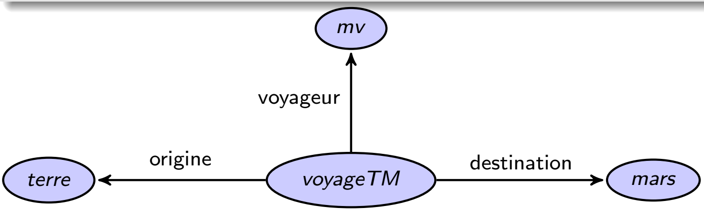
Difficulté de représenter les quantificateurs (existentiels, universels), la négation.

Réseaux sémantiques

Représentation graphique

EXemple de réseau : *MV va de la Terre à Mars.*

1. $voyageur(voyageTM, mv) \land origine(voyageTM, terre) \land destination(voyageTM, mars)$

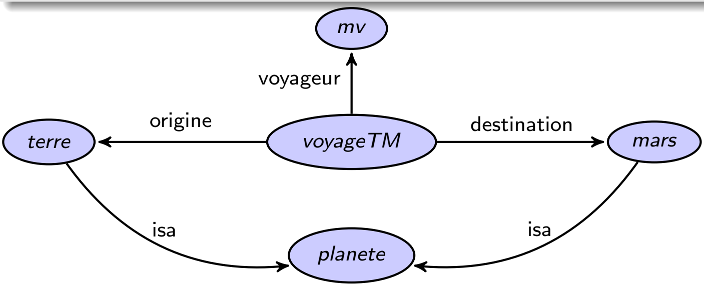


Réseaux sémantiques

Représentation graphique

Enrichissement du réseau : *Terre et Mars sont des planètes*

2. $isa(terre, planete) \wedge isa(mars, planete)$

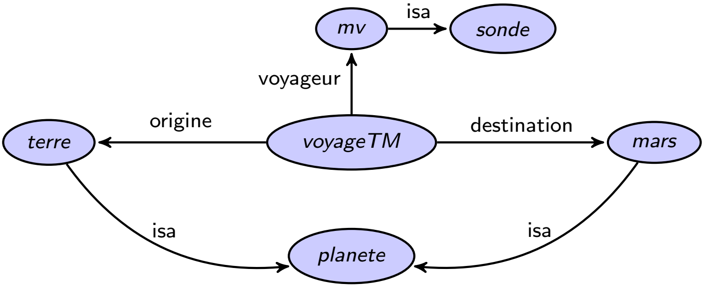


Réseaux sémantiques

Représentation graphique

Enrichissement du réseau : *MV est une sonde*

1. $voyageur(voyageTM, mv) \wedge origine(voyageTM, terre) \wedge destination(voyageTM, mars)$
2. $isa(terre, planete) \wedge isa(mars, planete)$
3. $isa(mv, sonde)$



Exercices réseaux sémantiques

Représentation

a. Le pull est bleu
b. Mon pull est gris
c. Shazia est plus petite qu'Arnaud
d. Shazia qui fait 1.68 m est plus petite qu'Arnaud qui mesure 1.85 m
e. Mehdi a prêté le livre "La Proie" de 'M. Crichton' à Marie

Réseaux Bayesiens

Connaissances probabilistes

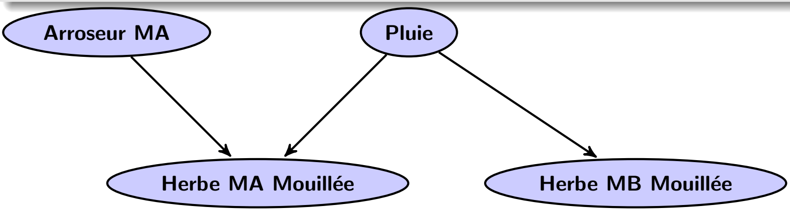
Réseaux Bayesiens (ou diagramme d'influences)

- Modèle de présentation des connaissances probabilistes
- Décrit les relations causales entre variables par un graphe
- Pour : l'aide à la décision, le diagnostic, le datamining

Exemple de réseau Bayesien

Qui arrose le jardin?

Dans une ville, deux maisons mitoyennes *MA* et *MB* possèdent des jardins. Un arroseur automatique se trouve le jardin de *MA*.
Si l'herbe est mouillée dans les jardins de *MA* et *MB*, quelle est la probabilité que ce soit dûe (*a*) à la pluie; (*b*) à l'arroseur automatique ?



Exemple de réseau Bayesien

Exemple de définition des variables

Les variables sont :

- $A \in \{on, off\}$: état de l'arroseur
- $P \in \{true, false\}$: définit le fait qu'il ait plu ou non
- $HA \in \{true, false\}$: définit le fait que l'herbe de MA soit mouillée
- $HB \in \{true, false\}$: définit le fait que l'herbe de MB soit mouillée

- A chaque noeud on définit des tables de probabilités.
- A chaque noeud possédant un/des entrant/s, on définit des tables de probabilités sachant la valeurs des noeuds 'parents'.

Exemple de réseau Bayesien

Table sur la pluie

L'arroseur est en mode *on* 10% du temps.
Il pleut 4 jours sur 10 dans la région de *MA* & *MB*.
Les tables associées aux variables *A* et *C* sont :

P(A)		**P(P_{luie})**	
true	*false*	*true*	*false*
0.1	0.9	0.4	0.6

Relation *MB* / Pluie

La probalité que l'herbe de *MB* soit mouillée sachant la donnée pluie, $P(MB|P)$, est :

P(MB|P_{luie})		
$P_{luie} \downarrow$ MB$\rightarrow$	*true*	*false*
$(P_{luie} = true)$	0.99	0.1
$(P_{luie} = false)$	0.01	0.9

Exemple de réseau Bayesien

Relation entre *MA*, la Pluie et l'Arroseur

La probabilité que l'herbe de *MA* soit mouillée sachant l'état de l'arroseur et de la pluie, notée $P(MA|A, P)$ est :

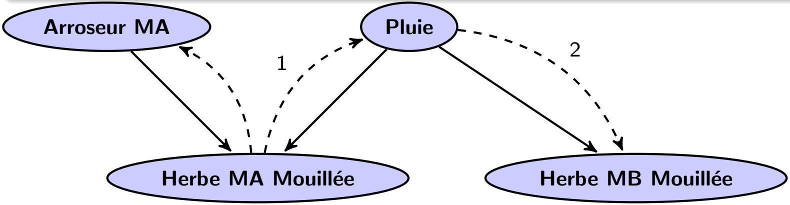
A	*on*	*on*	*off*	*off*
P_{luie}	*true*	*false*	*true*	*false*
$P(MA_{true}\|A, P_{luie})$	0.999	0.95	0.99	0.1
$P(MA_{false}\|A, P_{luie})$	0.001	0.05	0.01	0.9

Table: $P(MA|A, P_{luie})$

Exemple de réseau Bayesien

Qui arrose le jardin?

- Initialement, il y a une chance sur deux que l'herbe de *MA* soit mouillée.
- SI l'observation au matin montre que *MA* est mouillée, la probabilité que l'arroseur ait été en action augmente (10% → 19%), ainsi que celle qu'il ait plu (40% → 78%).
- Par conséquence, la possibilité que l'herbe de *MB* soit mouillée augmente (79.1%).



Réseaux Bayesiens: Définitions

Eléments de définition

Réseaux Bayesiens $= G$: graphe orienté sans cycle

- Noeud ($V = $ *Vertex*) : variable aléatoire (continue ou discrète)
- Arc ($E = $ *Edge*) : relation de causalité entre variables

- Pour chaque variable $x_i \in V$, la distribution de probabilité conditionnelle est : $P(x|parents(x))$
  où $parents(x) =$ les parents directs de x dans G
- La distribution de probabilité jointe s'écrit :
  $P(x_1, \ldots, x_n) = \prod_{x_i} P(x_i|parents(x_i))$

  Exemple :
  $P(A, P, MA, MB) = P(A) \times P(P) \times P(MB|P) \times P(MA|A, P)$

Réseaux Bayesiens: Définitions

Objectif

Calculer la probabilité a posteriori de certaines variables en fonction de certaines observations (évidences)

- Evidence = variable dont la valeur est connue avec certitude
- Inférence = propagation des évidence = calcul des probabilités a posteriori

Apprentissages possibles

- Possibilité d'apprendre les valeurs des tables de probabilité à partir d'un échantillon d'exemples (aisé)
- Possibilité d'apprendre la structure du graphe à partir d'un échantillon d'exemples (délicat et à valider par un expert du domaine)

Réseaux Bayesiens: Définitions

Théorème de Bayes

$$P(B|A) = \frac{P(A|B) \times P(B)}{\sum_i (P(A|B_i) \times P(B_i))}$$

Pleut-il?

- Il pleut 5 jours / 7 $\rightarrow P(Pluie) = 0.7$ et $P(Pluie_c) = 0.3$
- Lorsqu'il pleut, je prend mon parapluie 3 fois sur 5 : $P(Para|Pluie) = 0.6$
- Sinon, au cas où, je prends mon parapluie 1 fois sur 10 : $P(Para|Pluie_c) = 0.1$
- Sachant que j'ai mon parapluie, quelle est la probabilité qu'il pleuve ?

Réseaux Bayesiens: Définitions

Pleut-il?

- Je vois personne avec un parapluie, quelle est la probabilité qu'il pleuve ?
- $P(Pluie|Para) =$
  $$\frac{P(Para|Pluie) \times P(Pluie)}{(P(Para|Pluie) \times P(Pluie) + P(Para|Pluie_c) \times P(Pluie_c))} =$$
  $$\frac{0.6 \times 0.7}{(0.6 \times 0.7 + 0.1 \times 0.3)} = \frac{0.42}{0.45} = 0.93$$
  $\rightarrow$ Il y a 93% de chance qu'il pleuve quand je vois une personne avec un parapluie.

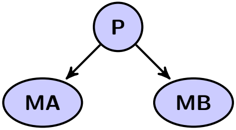
Remarques sur les réseaux Bayesien

Graphes équivalents

Les graphes $(A) \rightarrow (B) \rightarrow (C)$, $(A) \leftarrow (B) \rightarrow (C)$, $(A) \leftarrow (B) \leftarrow (C)$ sont équivalents

Liaison divergente

MA et MB ne sont dépendant que si P est inconnu.
MA et MB sont indépendant sachant P :
$MA \perp MB | P$



Liaison convergente

A et P ne sont dépendant que si MA est connu.
Sinon A et P sont indépendant : $A \perp P$