

---

# Réseaux bayésiens en modélisation d'utilisateurs

**Christophe Gonzales — Pierre-Henri Wuillemin**

*Laboratoire d'Informatique de Paris VI (LIP6)  
Université Paris 6  
tour 46-0, 2<sup>ème</sup> étage — 4, place Jussieu  
75252 Paris Cedex 05, France  
e-mail : {Christophe.Gonzales,Pierre-Henri.Wuillemin}@lip6.fr*

---

*RÉSUMÉ. Cet article a pour but de présenter les réseaux bayésiens, outil issu de l'Intelligence Artificielle autant que de la Statistique, et de montrer leur intérêt pour la Modélisation de l'Utilisateur. Il s'attache à présenter différentes applications de cet outil, à proposer des pistes pour son utilisation dans ce domaine et enfin à montrer les voies de recherche actuelles.*

*ABSTRACT. Bayesian networks are tools that were developed by the Artificial Intelligence and Statistic communities for handling uncertainties. This paper explains what they are and to what extent they can be useful in User Modeling. In particular, some applications of Bayesian networks are presented, as well as some clues on how they can be used in User Modeling. Finally, the paper gives an account of the current researches in the field.*

*MOTS-CLÉS. Réseaux bayésiens, Probabilités, Intelligence Artificielle, ITS, Modélisation utilisateurs*

*KEY WORDS. Bayesian Networks, Probabilities, Artificial Intelligence, ITS, User Modeling*

---

La Modélisation de l'Utilisateur nécessite une bonne gestion des incertitudes car elle est fondée sur un recueil de données par principe incertaines (difficulté de trouver des données objectives) et incomplètes (difficulté de posséder l'ensemble des données au bon moment). C'est l'un des problèmes majeurs d'une telle modélisation que de réussir à contourner, à dépasser ces faiblesses. Or, dans le domaine de l'IA, le but même des Réseaux Bayésiens (que l'on notera "RB" par la suite) est d'offrir un outil de gestion des incertitudes à la fois souple et performant. Il est important de noter ici que la différence formelle entre Modélisation de l'Utilisateur et Modélisation de l'Apprenant n'existe pas du point de vue des réseaux bayésiens : en effet, l'utilisateur ou l'apprenant sont les acteurs dont le réseau essaye de deviner les buts, les plans, les connaissances à partir de leurs actions. Seules les questions posées au réseau différeront mais leur traitement sera identique. Il semble donc intéressant d'étudier et de développer leurs applications dans ce qu'on nommera pour simplifier Modélisation de l'Utilisateur, mais

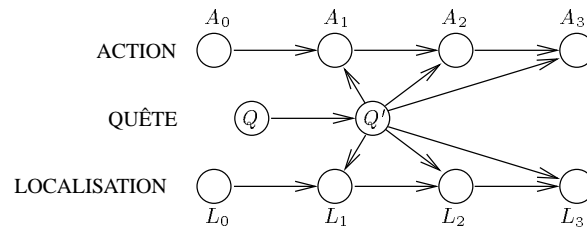
qui englobe aussi les approches d'enseignement assisté ou ITS<sup>1</sup>. Dans ce but, cet article tentera d'aborder les principales questions qui se posent quant à l'utilisation des RB.

La première section donne une vue d'ensemble des RB. La seconde se focalise sur les types de problèmes — rencontrés en particulier dans les ITS — qu'ils permettent de modéliser. La troisième section montre comment utiliser en pratique les RB. Enfin, la dernière section traite des découvertes récentes en matière de RB.

## 1. Introduction aux réseaux bayésiens

Formellement, un réseau bayésien (qui est aussi appelé réseau probabiliste, ou encore, en anglais, "Bayesian network", BBN : "Bayesian Belief Network", "probabilistic network") est un graphe dont les nœuds représentent des variables aléatoires<sup>2</sup> et dont les arcs expriment des interdépendances, des influences entre ces variables. Le graphe modélise donc qualitativement des influences entre variables aléatoires. C'est en fait l'absence d'arc entre deux variables qui certifie l'absence d'influence directe entre elles. Pour quantifier la "force" de ces influences, on adjoint à chaque nœud sa loi de probabilité conditionnellement à ses parents dans le graphe<sup>3</sup>.

Par exemple, le RB de la figure 1, emprunté à [ALB 97], modélise le comportement



**Figure 1.** Exemple de réseau bayésien

d'un joueur dans le jeu de rôle intitulé "Shattered World". Il s'agit pour celui-ci d'effectuer certaines quêtes (ici les nœuds  $Q, Q'$ ) grâce à des actions précises (les  $A_i$ ) dans des lieux précis (les  $L_i$ ). Plus précisément, lorsque le joueur commence le jeu, il peut choisir parmi plusieurs quêtes celle qu'il désire entreprendre, les choix possibles étant, entre autres, "TEDDY-BEAR RESCUE", "WOOD CHOP" ou bien encore "AVATAR" (notés ci-dessous "RESCUE", "CHOP" et "AVATAR").  $Q$  représente la quête que le joueur a choisi d'effectuer. Bien évidemment, lorsque le joueur lance le jeu,  $Q$  n'est pas encore connue ; c'est donc, dans une modélisation probabiliste, une variable aléatoire prenant pour valeur les choix présentés ci-dessus avec certaines probabilités. Lorsque le choix a été réalisé, la variable est instanciée. De même, les nœuds  $A_i$  représentent eux aussi des variables aléatoires : par exemple,  $A_1$  représente la première action que le joueur

<sup>1</sup> Intelligent Tutoring System

<sup>2</sup> Dans cet article, on identifiera une variable aléatoire du système étudié et le nœud représentant cette variable dans le graphe.

<sup>3</sup> Pour un arc  $Y \rightarrow X$  dans le graphe, on dit que "Y est parent de X" et que "X est fils de Y".

va réaliser dans la quête  $Q'$ . Avant qu'il n'ait choisi l'action qu'il compte effectuer,  $A_1$  peut prendre différentes valeurs ("se déplacer vers l'est", "manger"...). Lorsqu'il réalise sa première action, la variable aléatoire  $A_1$  est instanciée à la valeur correspondante.

L'accomplissement de la quête  $Q$  détermine (avec une certaine probabilité) la nouvelle quête  $Q'$  à entreprendre. C'est ce que symbolise l'arc  $Q \rightarrow Q'$ . Quantitativement, la quête  $Q'$  à accomplir dépend de la quête  $Q$  selon la loi de probabilité de  $Q'$  conditionnellement à  $Q$ , notée de manière abrégée  $\text{Prob}(Q'|Q)$ , mais donnée par la table 1 ci-dessous.

$\text{Prob}(Q' Q)$	$Q = \text{"RESCUE"}$	$Q = \text{"CHOP"}$	$Q = \text{"AVATAR"}$
$Q' = \text{"RESCUE"}$	0.7	0.4	0.0
$Q' = \text{"CHOP"}$	0.1	0.5	0.1
$Q' = \text{"AVATAR"}$	0.2	0.1	0.9

**Tableau 1.**  $\text{Prob}(Q'|Q)$ .

Ainsi, si la loi a priori (i.e. avant toute information) de  $Q$  est

$$\begin{cases} \text{Prob}(Q = \text{"RESCUE"}) = 0.3, \\ \text{Prob}(Q = \text{"CHOP"}) = 0.2, \\ \text{Prob}(Q = \text{"AVATAR"}) = 0.5, \end{cases}$$

la loi de  $Q'$  étant donnée par la formule classique

$$\text{Prob}(Q') = \sum_Q \text{Prob}(Q) \text{Prob}(Q'|Q),$$

elle est nécessairement égale à

$$\begin{cases} \text{Prob}(Q' = \text{"RESCUE"}) = 0.29, \\ \text{Prob}(Q' = \text{"CHOP"}) = 0.18, \\ \text{Prob}(Q' = \text{"AVATAR"}) = 0.53. \end{cases}$$

Par contre, si  $Q$  est connue par ailleurs, par exemple  $Q = \text{"RESCUE"}$ , alors, toujours en utilisant la formule ci-dessus, la loi de  $Q'$  devient

$$\begin{cases} \text{Prob}(Q' = \text{"RESCUE"}|Q = \text{"RESCUE"}) = 0.7, \\ \text{Prob}(Q' = \text{"CHOP"}|Q = \text{"RESCUE"}) = 0.1, \\ \text{Prob}(Q' = \text{"AVATAR"}|Q = \text{"RESCUE"}) = 0.2. \end{cases}$$

Pour mener chaque quête  $Q'$  à bien, le joueur doit réaliser certaines actions (les nœuds  $A_i$ ) en des lieux particuliers (les nœuds  $L_i$ ) : c'est le sens des arcs  $Q' \rightarrow A_i$  et  $Q' \rightarrow L_i$ . Les actions et les localisations s'enchaînent dans le temps : l'action  $A_0$  représente le début d'une quête,  $A_1$  représente la première action réalisée par le joueur dans la quête  $Q'$ ,  $A_2$  la deuxième, et ainsi de suite. Les arcs  $A_i \rightarrow A_{i+1}$  (resp.  $L_i \rightarrow L_{i+1}$ ) représentent l'influence que peut avoir la dernière action effectuée par le joueur (resp.

la dernière localisation du joueur) sur la nouvelle. La “force” de ces influences est décrite par la probabilité que l'utilisateur va effectuer telle action  $A_{i+1}$ , sachant qu'il essaye d'accomplir la quête  $Q'$  et que son action précédente était  $A_i$  :  $\text{Prob}(A_{i+1}|A_i, Q')$ . On remarque que ce modèle ne suppose aucune influence entre le lieu et l'action à un même instant (pas d'arc  $L_0 \rightarrow A_0$  par exemple), ce qui n'est vraisemblablement qu'une hypothèse simplificatrice. Il sera question de la manière de construire ce RB aussi bien qualitativement (la topologie) que quantitativement (les tables de probabilités) dans la section 3.

À partir des données du graphe et des lois de probabilité conditionnelle, on peut considérer un RB comme un système expert probabiliste. Les orientations des arcs prennent ainsi une valeur sémantique précise<sup>4</sup> :  $X \rightarrow Y$  s'interprète alors comme une règle “ $X$  est cause de  $Y$ ” (règle plus complexe que dans un système expert classique puisque  $X$  comme  $Y$  peuvent prendre plusieurs valeurs). En particulier, on peut utiliser le réseau pour faire :

- **de la prédiction** (chaînage avant) : étant donné les causes ( $Q', A_2$ ), quelles sont les valeurs probables des conséquences ( $A_3$ ) ? Autrement dit, si l'on sait quelle quête l'utilisateur essaye d'accomplir et si l'on a observé qu'il vient d'effectuer l'action  $A_2$ , que peut-on dire sur la probabilité qu'il effectue telle action  $A_3$  ?
- **du diagnostic** (chaînage arrière) : étant donné des conséquences, quelles sont les causes plausibles ? Autrement dit, si l'on connaît les actions  $A_{i+1}, \dots, A_n$  à effectuer pour accomplir une quête  $Q'$ , peut-on déterminer quelle est la probabilité que l'utilisateur a effectué l'action  $A_i$  ?

D'une manière générale, les RB permettent de calculer rapidement n'importe quelle probabilité marginale (probabilité qu'une variable prenne telle valeur), jointe (probabilité qu'une série de variables prennent telle série de valeurs), conditionnelle (probabilité qu'une variable prenne telle valeur, sachant la valeur d'une autre variable). On peut ainsi déterminer la probabilité que le joueur soit en train d'effectuer telle ou telle quête,  $\text{Prob}(Q)$  (reconnaissance de plan) ou bien encore évaluer la probabilité qu'une action quelconque soit réalisée par l'utilisateur,  $\text{Prob}(A_i)$ . De plus, et c'est là l'une de leurs fonctionnalités les plus intéressantes, les RB permettent de mettre à jour rapidement les probabilités lors de l'arrivée de nouvelles informations. Par exemple, lorsque le joueur débute la quête  $Q' = q'$ , la probabilité que sa deuxième action,  $A_2$ , soit  $a_2$  (“WORSHIP” par exemple) est égale à :

$$\text{Prob}(A_2 = a_2 | Q' = q') = \sum_{a_1} \text{Prob}(A_2 = a_2 | Q' = q', A_1 = a_1) \times \text{Prob}(A_1 = a_1 | Q' = q').$$

Mais après avoir observé que le joueur a effectué l'action  $A_1 : a_1 = \text{“WORSHIP”}$ , la probabilité que  $A_2 = a_2$  est actualisée à la valeur suivante :

$$\text{Prob}(A_2 = a_2 | Q' = q', A_1 = a_1).$$

<sup>4</sup> En fait, dans un réseau bayésien, l'orientation des arcs a une grande importance, et nous verrons précisément dans la section 3 comment la déterminer.

On voit ainsi que les RB permettent de faire de la prédiction en temps réel, prenant en compte toutes les actions effectuées par l'utilisateur.

Le lecteur intéressé par des ouvrages de référence très documentés pourra se reporter aux livres suivants :

- [PEA 88], en particulier aux chapitres 1, 2 et 4 qui décrivent l'intérêt des RB pour la gestion d'incertitudes, l'inférence bayésienne, et la manière de conduire les calculs probabilistes cités ci-dessus ;
- [JEN 96], en particulier aux chapitres 2, 3 et 5 qui expliquent à l'aide de plusieurs exemples ce que sont les RB, comment les construire et comment effectuer les calculs des probabilités ;
- [NEA 90], en particulier au chapitre 5, une introduction aux RB, et aux chapitres 6 et 7 qui, là encore, montrent comment calculer les probabilités dans le réseau.

## 2. Application à la modélisation de l'utilisateur

L'une des difficultés de la modélisation de l'apprenant, ou plus généralement de l'utilisateur, réside dans le fait qu'on ne peut observer directement ce qu'il connaît ou ne connaît pas, mais seulement l'estimer de manière très imparfaite à travers ses actes ; de même, au seul examen de ses actions, on ne peut savoir avec certitude le but qu'il cherche à accomplir. C'est pour tenter de gérer ce type d'incertitudes que les RB peuvent être utilisés — et le sont déjà — en Modélisation de l'Utilisateur.

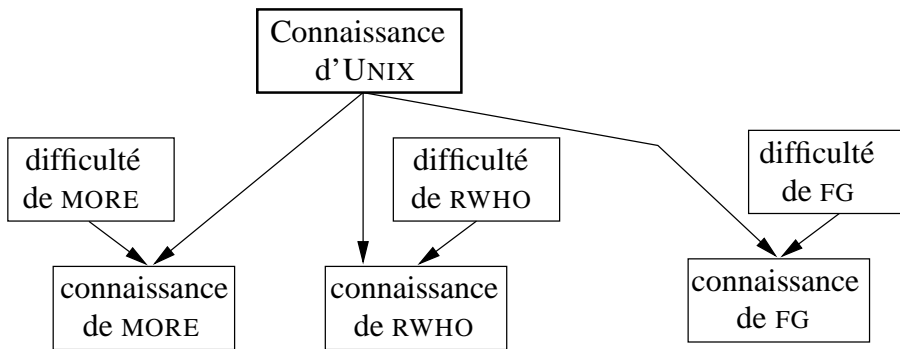
Nous allons présenter dans cette section des problèmes pour lesquels les RB ont déjà été utilisés avec succès. Ceux-ci peuvent être classés en trois catégories : i) l'évaluation de connaissances ("knowledge assessment") ; ii) la reconnaissance de but ("plan recognition") ; iii) la prédiction de comportement ("prediction"). En fait, du point de vue des RB, ces catégories ne se distinguent que par les questions que l'on pose au RB, et la construction du réseau ne dépend absolument pas de ce que l'on compte en faire. Ainsi, le même réseau pourra être utilisé pour prédire le comportement d'un utilisateur ou pour reconnaître les buts qu'il poursuit, seules les questions qu'on lui posera seront différentes.

### 2.1. Evaluation des connaissances ("*knowledge assessment*")

Pour évaluer le niveau de connaissance d'un utilisateur, un expert confrontera toutes les informations qu'il a sur ce dernier à ses propres connaissances, et inférera à partir de là (grâce à un mécanisme qui lui est propre) ce que l'utilisateur semble connaître. Cette tâche peut être automatisée informatiquement à partir de différentes modélisations : citons par exemple les systèmes experts à base de règles, la logique floue (cf. [CHI 89] ou [KAT 92]), ou bien la théorie de Dempster-Shafer (cf. [PET 93]). Dans ces conditions, pourquoi choisir d'utiliser précisément les RB pour inférer les

connaissances d'un utilisateur (qu'on notera  $U$ ) ? Plusieurs raisons peuvent être invoquées :

1. La partie graphique du RB représente le plus exactement possible les démarches de l'expert. Par exemple, [JAM 96] montre comment évaluer le niveau de connaissance de  $U$  en UNIX : l'expert considère que ce niveau est en relation avec la connaissance des commandes "MORE", "RWHO" et "FG". Dans le réseau, il doit donc y avoir des arcs entre le nœud "connaissance d'Unix" et les nœuds "connaissance de MORE", "connaissance de RWHO" et "connaissance de FG". L'orientation de ces arcs sera explicitée dans la section suivante. De plus, l'assimilation d'une commande peut se révéler plus ou moins difficile. On crée donc des nœuds "difficulté" que l'on relie par des arcs aux nœuds "connaissance". Toutes ces informations permettent de déduire le RB de la figure 2.



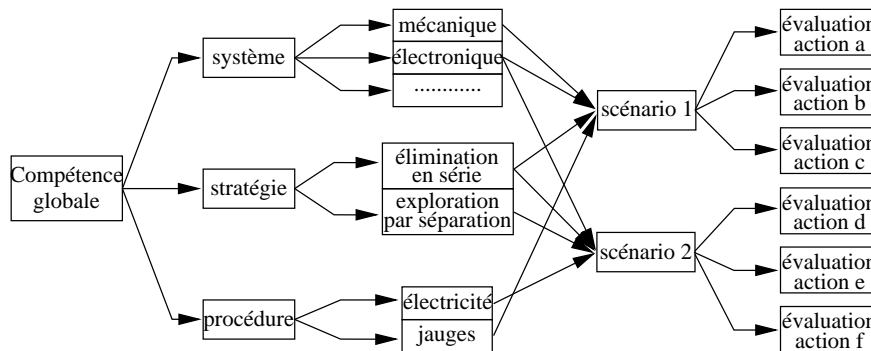
**Figure 2.** Evaluation des connaissances en UNIX d'un utilisateur  $U$

2. Les RB permettent d'évaluer les connaissances en présence d'informations incomplètes. Par exemple, pour estimer parfaitement le niveau en UNIX de  $U$ , l'expert devrait en principe déterminer dans quelle mesure  $U$  connaît les commandes MORE, RWHO et FG ; mais si l'une (ou plusieurs) de ces informations ne lui est pas accessible, le réseau bayésien lui permet d'utiliser des probabilités a priori, c'est-à-dire les probabilités qu'un utilisateur moyen connaisse telle ou telle commande.
3. Les connaissances d'un utilisateur sont inférées principalement à partir des actions qu'il réalise. Le mécanisme de propagation d'informations des RB (cf. sous-section 4.1), en assurant une mise à jour rapide des probabilités du réseau lors de l'arrivée d'informations, permet d'actualiser l'évaluation des connaissances après chaque action de l'utilisateur. De plus, le réseau peut prendre en compte le fait que les informations qui lui sont transmises sont bruitées.
4. Enfin, les RB s'appuyant sur la théorie des probabilités, leur gestion des incertitudes possède un fondement mathématique et théorique rigoureux, répondant

ainsi aux conclusions de [MAR 95], indiquant que l'évaluation des connaissances se doit d'être fondée sur des méthodes statistiques solides.

Les RB permettent de modéliser l'utilisateur  $\mathcal{U}$  grâce à un ensemble de variables décrivant non seulement les capacités et les connaissances de  $\mathcal{U}$ , mais aussi des informations intrinsèques au domaine sur lequel portent les connaissances (difficulté de la commande MORE par exemple). Les probabilités de ces variables sont mises à jour en fonction des actions effectuées par  $\mathcal{U}$ . À titre d'illustration, le lecteur trouvera ci-dessous trois exemples d'application des RB à l'évaluation des connaissances.

HYDRIVE est un ITS (Intelligent Tutoring System) permettant d'évaluer les connaissances d'étudiants en matière de réparation du système hydraulique de l'avion F15 (cf. [MIS 96]). Une étude a montré que celles-ci dépendent des trois composantes suivantes : i) la connaissance du système, c'est-à-dire du fonctionnement des appareils utilisant le système hydraulique ; ii) la connaissance des différentes stratégies permettant de déterminer les problèmes hydrauliques ; iii) les connaissances des procédures de contrôle des appareils. Chaque composante est divisée en plusieurs sous-composantes (connaissances en mécanique, en électronique, . . .), et ces dernières peuvent être évaluées grâce à différents scénarios-catastrophes. Enfin,  $\mathcal{U}$ , ici mécanicien en formation, est évalué, pour chaque scénario, en fonction des actions qu'il effectue. Ces informations nous fournissent directement la partie graphique du réseau bayésien de la figure 3.

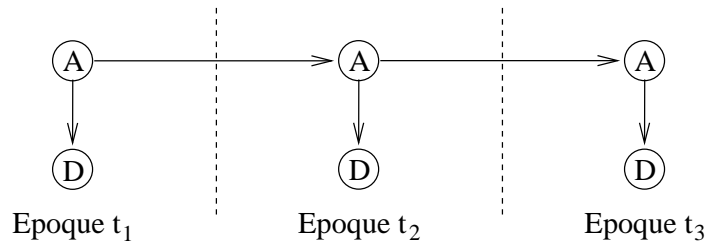


**Figure 3.** Esquisse de RB utilisé par HYDRIVE

Ce réseau se comporte comme si les connaissances de l'étudiant ne variaient pas dans le temps : chaque nouvelle action observée affine leur évaluation, mais en aucun cas elle n'est perçue par HYDRIVE comme une évolution des connaissances. Or le but d'un ITS est d'aider les étudiants au cours du temps. Pour prendre en compte cette notion, il faut utiliser une structure particulière de RB : les RB dynamiques (cf. section 4.3 et [KJA 93b]). [MIS 96] montre comment le réseau de la figure 3 peut être transformé en RB dynamique.

Ce cas d'utilisation des RB dynamiques dans des ITS n'est pas isolé. En effet, [REY 96] montre comment ACT (cf. [COR 92]) peut tirer avantage d'une modélisation par RB dynamique. Plus généralement, il tente d'établir une méthodologie pour déterminer la structure des RB conçus pour des ITS.

Enfin, citons [SCH 97], qui décrit READY, un système aidant, entre autres, les pompiers de Sarrebruck à diagnostiquer la gravité des urgences lors d'appels téléphoniques (c'est donc l'appelant qui est modélisé par le système.) . L'un des avantages de la méthode retenue est de combiner à la fois RB dynamique et RB statique : en effet, certaines variables du réseau (anxiété, difficulté à respirer, ...) varient beaucoup dans le temps, alors que d'autres sont moins sujet à ces brusques variations (expérience des urgences,...). La figure 4 montre comment fonctionne la partie dynamique du réseau :



**Figure 4.** Anxiété et difficulté à respirer dans [SCH 97]

nœud  $A$  représente l'anxiété et  $D$  la difficulté à respirer. Le temps est scindé en différentes époques, dont chacune influence la suivante. Il y a donc duplication des nœuds entre les différentes époques, et les probabilités conditionnelles tiennent compte des époques précédentes : par exemple sur la figure 4, la probabilité conditionnelle de  $A$  à la période 2 est  $\text{Prob}(A = a \text{ en période } 2 | A = a' \text{ en période } 1)$ .

## 2.2. Reconnaissance de plan d'actions ("Plan Recognition")

À l'instar de l'évaluation des connaissances, la reconnaissance de plan d'actions consiste à inférer des informations sur l'utilisateur en se fondant sur l'observation de ses actions. Intrinsèquement, les incertitudes que doivent gérer ces deux domaines sont donc relativement proches. Aussi, les raisons que nous avons évoquées pour justifier une modélisation par RB dans la section précédente sont-elles encore valables ici. Une autre raison peut toutefois être invoquée, qui résulte directement de l'algorithme de propagation d'information des RB : il est possible de déterminer quelles sont les valeurs les plus probables des variables dans le réseau (cf. [JEN 96]). L'application en est évidente : si l'observation des actions est interprétée comme l'arrivée de nouvelles informations, le mécanisme d'inférence du RB pourra déterminer le but recherché par l'utilisateur avec la probabilité maximale.

Cette fonctionnalité est d'un intérêt de tout premier plan et explique en partie l'intérêt croissant vis à vis des RB, de la communauté de recherche sur la Modélisation d'Utilisateur, et en particulier des chercheurs en reconnaissance des plans d'action. Ainsi, [PYN 95] utilise les RB pour la reconnaissance du plan d'un automobiliste à partir de l'observation des mouvements du véhicule. [FOR 95] quant à lui étudie la reconnaissance de plan en contexte automobile par RB dynamique. Dans un domaine plus ludique, [ALB 97] utilise aussi des RB dynamiques pour in-



féder les buts recherchés par les joueurs de “Shattered Worlds”, le jeu de rôle décrit dans la section 1. Enfin, n’oublions pas de citer le projet Lumière (cf. le site internet <http://www.research.microsoft.com/~horvitz/lum.htm> ou [HOR 97]), dont l’objectif est d’estimer par RB les buts recherchés par les utilisateurs de Microsoft Office ainsi que leurs besoins. Lumière permet ainsi à l’Assistant d’Office de proposer précisément aux utilisateurs les aides dont ils ont besoin.

### **2.3. Prédiction**

Enfin, comme nous l’avons remarqué dans la section 1, les RB sont particulièrement bien adaptés pour faire de la prédiction. Dans le réseau bayésien de la figure 1, il est en effet relativement aisé de déterminer la prochaine action la plus probable, connaissant la dernière action effectuée et la quête courante.

Dans un domaine plus scientifique, [CON 97] décrit ANDES, un ITS dédié à l’apprentissage de la physique newtonienne. ANDES utilise le “coached problem solving”, c’est-à-dire une méthode dans laquelle le tuteur pose un problème à l’étudiant, laisse celui-ci le résoudre jusqu’à ce qu’il perçoive que l’étudiant s’oriente dans une mauvaise direction, auquel cas ANDES va essayer de le remettre dans le droit chemin. Pour réaliser cela, le logiciel doit interpréter les actions de l’étudiant ; ANDES le fait grâce à un mécanisme complexe fondé sur un RB. Pour être plus précis, ce RB permet d’inférer : i) une reconnaissance du but recherché par l’étudiant ; ii) une prédiction sur ses prochaines actions ; iii) une évaluation à long terme des connaissances de l’étudiant.

## **3. Utilisation pratique des réseaux bayésiens**

Comme on vient de le voir, les RB semblent prometteurs pour la Modélisation de l’Utilisateur (ou de l’apprenant). Cependant leur utilisation demande certaines précautions du fait même du nombre de connaissances intégrées et de la rigueur des mathématiques sous-jacentes. Le modèle choisi, le graphe construit, les probabilités utilisées ont une importance à ne pas négliger. Dans cette section, il s’agit de donner des pistes à suivre, des écueils à éviter dans le choix et la topologie du réseau, dans la méthode de propagation utilisée, ainsi que dans l’acquisition des probabilités conditionnelles.

### **3.1. Choix de la topologie**

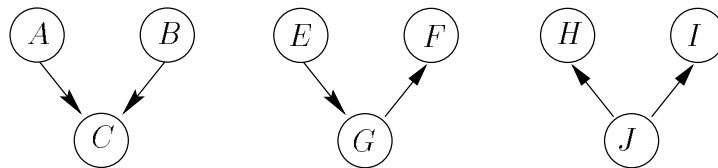
La topologie du RB (c’est-à-dire la structure du graphe) est, pour une grande partie des applications, fournie comme donnée au système (apprentissage automatisé de cette structure cf. sous-section 4.4). Elle peut aussi être explicitée “manuellement” : nous avons vu dans la section 2 plusieurs exemples dans lesquels les données du problème à traiter nous fournissent directement la topologie du RB. En fait, il n’est pas trop difficile de déterminer les nœuds du graphe : par exemple, en faisant une introspection, l’enseignant peut recenser les critères qu’il utilise pour fonder son jugement sur les connaissances de l’étudiant ; chaque critère correspond alors à un nœud. De

même, il est facile d'estimer si un arc doit relier deux nœuds : il suffit de déterminer si une information sur l'un des nœuds (modification de la valeur de la variable aléatoire correspondante, ou tout du moins de sa loi de probabilité) entraîne une modification de la variable correspondante à l'autre nœud. Le point le plus délicat, et qui n'est pas encore résolu à cette étape, est de déterminer le sens des arcs joignant les nœuds. L'un des buts de cette section est d'expliquer comment réaliser cette opération.

Une fois celle-ci achevée, si les probabilités conditionnelles de tous les nœuds sont renseignées, le réseau est opérationnel. Toutefois, il est parfois intéressant, pour des raisons d'efficacité des calculs, de modifier cette topologie ; c'est ce que nous montrons dans la sous-section 3.1.2. Enfin, nous terminons notre tour d'horizon en explicitant la manière d'acquérir les probabilités et le choix de la méthode de propagation des informations dans le réseau.

### 3.1.1. Direction des arcs

La direction des arcs a une grande influence sur le modèle car c'est elle qui détermine quels calculs vont être effectués dans le réseau. Pour mieux le comprendre, considérons par exemple les RB de la figure 5. Jouons au jeu suivant : on lance deux pièces



**Figure 5.** Importance de l'orientation des arcs sur les calculs.

(totalement indépendantes). Si elles atterrissent toutes les deux sur la même face, on gagne 1 franc, sinon, on en perd 1. On peut visiblement représenter ce jeu par l'un des réseaux ci-dessus : par exemple, sur le graphe de gauche,  $A$  et  $B$  représenteraient le résultat du jet de chacune des pièces et  $C$  représenterait une variable aléatoire valant 1 si les valeurs de  $A$  et de  $B$  sont égales et -1 sinon. Apparemment, on pourrait aussi bien utiliser les deux autres graphes pour modéliser le même jeu, mais il n'en est rien car la sémantique des trois graphes est différente.

Dans le graphe de gauche, l'orientation des arcs signifie que les variables  $A$  et  $B$  sont indépendantes mais sont aussi dépendantes conditionnellement à  $C$ , c'est-à-dire que si on n'apporte pas d'information sur  $C$ , connaître la valeur de  $A$  n'ajoutera rien à la connaissance qu'on a de la valeur de  $B$ , mais que, si une information arrive en  $C$ , les informations en  $A$  auront une influence sur  $B$ . Illustration : supposons que l'on ne connaisse pas la valeur de  $C$  (autrement dit, on ne sait pas quelles sont les valeurs des variables  $A$  et  $B$ ), alors savoir que  $A$  vaut "pile" ne nous donne pas d'information sur la valeur de  $B$  d'après notre hypothèse d'indépendance. Au contraire, si l'on sait que la valeur de  $C$  est +1, obtenir l'information que  $A$  vaut "pile" permet de conclure que  $B$  vaut aussi "pile".

D'une manière générale, dans le graphe de gauche,  $A$  ne peut influencer  $B$  (et réci-

proquement) que si une information ne provenant *pas d'un parent de C* est arrivée en  $C$ . Ainsi, ce n'est pas parce que  $A$  modifie la valeur de  $C$ , que  $A$  modifie aussi la valeur de  $B$  : il faut vraiment qu'une information ne provenant ni de  $A$  ni de  $B$  parvienne en  $C$  pour que ce soit le cas. Illustration : modifions le jeu précédent de la manière suivante : en  $C$  nous gagnons toujours une somme  $x$  si les deux pièces ont atterri sur la même face et on perd  $x$  sinon, mais, maintenant,  $x$  vaut 2 francs si  $A$  vaut "pile" et 1 franc si  $A$  vaut "face". Alors, si l'on n'a aucune connaissance sur  $C$ , apprendre que  $A$  vaut "pile" nous permet de réviser nos connaissances :  $C$  ne peut maintenant valoir que 2 francs ou -2 francs. Toutefois, cela ne nous donne toujours aucun indice sur la valeur de  $B$ .

Pour les deux autres graphes, la sémantique associée aux arcs est différente : les nœuds  $E$  et  $F$  [resp.  $H$  et  $I$ ], sont dépendants et sont aussi indépendants conditionnellement à  $G$  [resp.  $J$ ]. C'est-à-dire que si l'on ne connaît pas  $G$  [resp.  $J$ ], il y a dépendance entre les variables, mais que si l'on sait la valeur de  $G$  [resp.  $J$ ] avec certitude, une information sur  $E$  [resp.  $H$ ] ne peut plus apporter d'information sur  $F$  [resp.  $I$ ] et réciproquement. Par exemple, supposons que le réseau bayésien représente l'activité d'un signal d'alarme. Le nœud  $E$  représente le fait qu'il y ait ou non un cambriolage,  $G$  le fait que le signal d'alarme se déclenche,  $F$  le fait que votre voisin vous téléphone ou non à votre bureau pour vous signaler que votre appartement est probablement cambriolé. Clairement, si l'alarme s'est déclenchée, qu'il y ait eu cambriolage ou non n'a aucune influence sur le fait que votre voisin vous téléphone : connaissant  $G$ ,  $E$  et  $F$  sont donc indépendants. Si votre voisin vous téléphone, vous ne pouvez pas savoir avec certitude si l'alarme s'est réellement déclenchée ou non ; par contre, la probabilité qu'il y ait effectivement un cambriolage est certainement plus élevée que si vous n'aviez pas reçu de coup de téléphone : les variables  $E$  et  $F$  sont donc bien dépendantes lorsque  $G$  n'est pas connu avec certitude.

Pour illustrer le réseau de droite, on peut considérer que l'alarme représente le nœud  $J$  et que les nœuds  $H$  et  $I$  sont deux de vos voisins. Si l'un d'eux vous téléphone à propos de votre alarme, il y a fort à parier que l'autre l'aura entendu aussi et vous téléphonera : les variables  $H$  et  $I$  sont dépendantes. Mais une fois que l'on sait que le signal s'est déclenché, le fait qu'un des voisins vous appelle ne vous donne aucune information sur le fait que l'autre va aussi appeler : ce qui compte c'est que l'autre voisin ait aussi entendu l'alarme.

En fait, les calculs sont les mêmes dans les deux réseaux les plus à droite de la figure 5. Pourquoi les différencier dans ce cas ? Tout simplement parce que tout RB, aussi compliqué soit-il, se décompose en fonction uniquement de ces trois cas de figure. De plus, les lois de probabilités conditionnelles que le réseau doit connaître sont différentes : dans les nœuds  $E$ ,  $F$  et  $G$  on doit stocker respectivement les tables  $\text{Prob}(E)$ ,  $\text{Prob}(G|E)$  et  $\text{Prob}(F|G)$  tandis que dans les nœuds  $H$ ,  $I$  et  $J$  on a  $\text{Prob}(H|J)$ ,  $\text{Prob}(I|J)$  et  $\text{Prob}(J)$ . On peut donc être amené à choisir comme modélisation l'un ou l'autre des deux réseaux de droite de la figure 5 en fonction de la difficulté d'estimer les probabilités conditionnelles. Il peut être en effet plus aisé de les estimer dans un réseau que dans l'autre. Dans l'exemple des deux voisins, il n'est pas trop complexe d'estimer la probabilité que chaque voisin vous téléphone connaissant l'état de l'alarme (graphe de droite). Si l'on voulait modéliser le même problème avec le graphe du milieu,  $G$  se-

rait l'état de l'alarme, et  $E$  et  $F$  les voisins. Dans ce cas, déterminer  $\text{Prob}(G|E)$  n'est pas très aisé : il s'agit de déterminer si l'alarme s'est déclenchée sachant que votre voisin vous a appelé ou ne vous a pas appelé (dans le premier cas, peut-être a-t-il entendu une autre alarme que la vôtre ; dans le deuxième, peut-être n'était-il pas dans le voisinage et donc ne pouvait vous prévenir, que l'alarme sonne ou pas).

Concrètement, la plus simple des méthodes pour orienter (et même pour concevoir) un réseau, c'est de le construire causalement. Deux nœuds sont reliés lorsqu'il y a influence entre les deux variables qu'elles représentent. L'orientation de la flèche fait aller de la variable explicative dans cette influence (la "cause") vers la variable expliquée (la "conséquence"). En pratique, il n'est pas toujours évident de distinguer ce qui est cause de ce qui est conséquence. Aussi, pourra-t-on utiliser les explications données ci-dessus : en fonction des dépendances et indépendances conditionnelles, on trouve sans trop de difficulté le sens des flèches. Ainsi, dans l'exemple d'Unix, les experts du domaine ont déterminé que modifier la connaissance de la commande "MORE" devait avoir une conséquence sur celle de "RWHO" et de "FG" : les arcs ne doivent donc pas être orientés des connaissances des commandes vers la connaissance d'Unix. Par symétrie entre les commandes, on en déduit donc qu'ils doivent tous être orientés de la connaissance d'Unix vers celle des commandes.

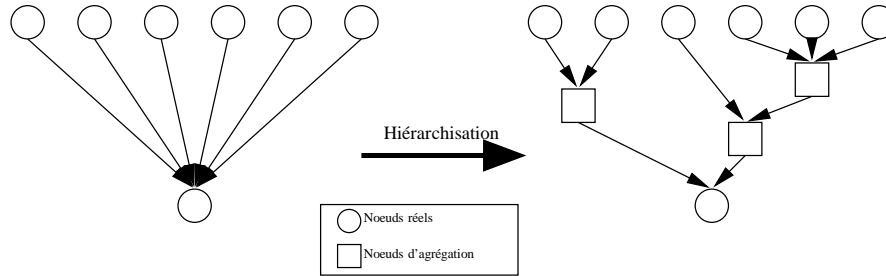
Dans certains articles, les auteurs sont tentés, pour des raisons d'efficacité dans les calculs (voir ci-dessous), de modifier l'orientation des arcs. Effectuer une telle manœuvre, c'est réellement transformer le schéma sous-jacent d'indépendances conditionnelles entre variables. Ainsi, passer d'une relation  $Y \rightarrow X \rightarrow Z$  à  $Y \rightarrow X \leftarrow Z$  change non seulement la sémantique du réseau, mais aussi les matrices de probabilité conditionnelle, leurs contenus et la façon même dont sont menés les calculs. Il ne faut donc pas effectuer trop légèrement une inversion d'arc : c'est en effet tentant car cela peut engendrer une réduction de la taille des matrices de probabilité conditionnelle ou, tout simplement, accélérer les propagations d'informations dans le réseau ; mais c'est au prix d'une transformation des probabilités obtenues qui ne correspondront plus à celles recherchées. Une modification plus mineure du graphe (comme l'ajout de variables cachées), une hiérarchisation est plus souvent appropriée.

### 3.1.2. Hiérarchisation du réseau

La complexité des calculs dans un RB est fortement liée au nombre de parents de chaque nœud. En effet, si  $n_i$  est le nombre de parents du nœud  $I$ , alors la matrice de probabilité conditionnelle associée à  $I$ ,  $\text{Prob}(I|\text{parents de } I)$ , aura  $(n_i + 1)$  dimensions. Or chaque mise à jour des probabilités nécessite le calcul du produit de ces matrices par certains vecteurs. On comprend donc que plus  $n_i$  est élevé, plus les calculs sont coûteux. À l'inverse, un graphe ayant une topologie en "forêt d'arborescences"<sup>5</sup> est souhaitable pour effectuer des calculs rapides. Une telle structure n'est évidemment pas applicable à n'importe quel problème. Cependant, dans la mesure du possible, on s'attachera à modéliser l'utilisateur par un graphe aussi proche que possible d'une forêt d'arborescences.

Ainsi, une hiérarchisation des variables du modèle, en créant, si besoin, des va-

<sup>5</sup> Une forêt d'arborescences est un graphe dont chaque nœud possède au plus un parent.



**Figure 6.** Hiérarchisation d'un modèle

riables d'agrégation comme présenté sur la figure 6, permet de se rapprocher de ce type de structure en réduisant le nombre de parents par nœuds. Par exemple, dans l'ITS de la figure 3, les nœuds "système", "stratégie" et "procédure" ne sont pas vraiment indispensables pour la modélisation mais leur existence la simplifie et accélère les calculs.

De plus, la structure hiérarchique améliore la lisibilité du graphe et la compréhension des relations entre les variables du modèle. Ceci permet de faciliter l'estimation des probabilités conditionnelles : en ce qui concerne l'ITS de la figure 3, il est plus aisé de juger le niveau de compétence globale en fonction des trois niveaux de compétence en "système", "stratégie" et "procédure" plutôt qu'en fonction des sept niveaux de compétence dans les différents sous-domaines (mécanique, électronique, etc.).

En général, une structure au moins en partie hiérarchique ne semble pas impossible en Modélisation de l'Utilisateur, puisque les notions s'y prêtent bien (l'estimateur de la connaissance du domaine est une synthèse de ceux des sous-domaines, etc.).

### 3.2. Acquisition des probabilités

Une fois la topologie du réseau identifiée, il reste à remplir les tableaux de probabilités conditionnelles. La première et la plus simple des méthodes consiste à utiliser des probabilités subjectives, c'est-à-dire choisir des valeurs en fonction de connaissances extérieures (un expert par exemple), en fonction d'une évaluation de la force de la relation entre les variables. Le risque est bien sûr de passer trop rapidement sur cette étape et de considérer que des lois uniformes seront suffisantes (typiquement, une variable  $X$  a deux modalités,  $\{\bar{x}, \underline{x}\}$ , donc  $\text{Prob}(X = \bar{x}) = \text{Prob}(X = \underline{x}) = 0.5$ ). Il est nécessaire de bien saisir la sensibilité du modèle à ces valeurs. Des lois uniformes sont des a priori très forts et qui peuvent souvent être facilement améliorés sans trop de peine : une loi uniforme est choisie parce qu'elle n'est pas informative. Toute information sur la variable ou sur la relation entre la variable et ses parents peut permettre une meilleure évaluation de ces probabilités.

Sans parler d'apprentissage proprement dit, l'utilisation de bases de données est une méthode bien plus satisfaisante quand elle est possible. Les calculs statistiques sont très simples : il s'agit d'estimer les probabilités conditionnelles par des fréquences

d'occurrence d'ensemble de valeurs dans la base :

$$\text{Prob}(X = x | Y = y, Z = z) = \frac{\text{Prob}(X = x, Y = y, Z = z)}{\text{Prob}(Y = y, Z = z)} \approx \frac{N_{X=x, Y=y, Z=z}}{N_{Y=y, Z=z}}.$$

Ce qui se lit : dans un RB où  $X$  a pour parents  $Y$  et  $Z$ , la valeur de la probabilité que  $X = x$  sachant que  $Y = y$  et que  $Z = z$  est approchée par le rapport du nombre de fois où simultanément  $X = x$ ,  $Y = y$  et  $Z = z$  dans la base par le nombre de fois où simultanément  $Y = y$  et  $Z = z$  dans la base. Dans le cas de la Modélisation de l'Utilisateur, il semble certain qu'une partie du réseau restera subjective (toute la modélisation de la connaissance de l'utilisateur). Par contre, il peut s'avérer qu'un certain nombre de liens entre variables objectives du système puissent être avantageusement traités statistiquement pour fournir au moins une partie des données du réseau.

Un des intérêts des RB est de garantir une cohérence des probabilités, même si l'on emploie des méthodes d'acquisition des probabilités conditionnelles différentes selon les nœuds du réseau. Autrement dit, il est tout à fait permis d'estimer les probabilités conditionnelles de certains nœuds à partir de bases de données et celles d'autres nœuds en ayant recours à des experts. Cette propriété des RB est très avantageuse car on peut ne posséder des bases de données que sur certaines parties du réseau (par exemple, les parties "objectives", décrivant le comportement physique du modèle). En fait, on peut même fusionner les connaissances de plusieurs experts sur différentes parties du graphe et des informations calculées à partir de plusieurs bases différentes concernant différentes parties du graphe. Cette remarque montre que les RB possèdent une grande souplesse en ce qui concerne la saisie de leurs informations quantitatives.

### 3.3. *Choix de la méthode de calcul : inférence exacte / inférence approchée*

[COO 88] a montré que, d'une façon générale, quelle que soit la méthode employée, l'inférence bayésienne est un problème NP-difficile. Cependant, malgré leur complexité, les méthodes exactes ont fortement évolué (cf. sous-section 4.1) et peuvent maintenant être utilisées en temps réel dans la plupart des RB, et ce, quel que soit le calcul de probabilité demandé. Il est donc conseillé de commencer par utiliser les méthodes exactes et de n'utiliser des méthodes approchées que lorsque les inférences exactes se sont avérées vraiment trop coûteuses en temps de calcul (cf. sous-section 4.2).

## 4. Nouveautés en réseaux bayésiens

Outre ces remarques d'ordre général sur l'utilisation des réseaux bayésiens, il est nécessaire de noter la grande activité de ce domaine. En effet, les RB deviennent un pôle d'importance sans cesse croissante dans les milieux de l'UAI (Uncertainty in Artificial Intelligence, cf. par exemple <http://www.lis.pitt.edu/~dsl/uai97.html>). Le but de cette section est de montrer des directions de recherche — pour la plupart datant des années 1990 — qui pourraient se révéler intéressantes pour la Modélisation de l'Utilisateur.

#### 4.1. Nouvelles méthodes d'inférence exacte dans un réseau bayésien

Les calculs d'inférences exactes ont beaucoup progressé depuis les années 80 où les premières études sont apparues. Les recherches dans ce domaine ont fait apparaître deux méthodes principales : celle proposée par [PEA 88] et celle proposée par [LAU 88] puis améliorée par [JEN 96].

À partir du RB (graphe orienté), Pearl propose un calcul fondé sur une propagation de messages à travers les arcs. Cette méthode n'est à la base applicable que dans les réseaux sans cycle (deux nœuds ne peuvent y être reliés que par une chaîne<sup>6</sup> au plus). Elle est cependant facilement généralisable à tout graphe sans circuit<sup>7</sup> grâce à la méthode dite du coupe-cycle (voir la figure 7(a)) : l'idée est de couper suffisamment d'arcs (tel que celui en pointillés) pour qu'il n'y ait plus de cycles. Pour que les calculs restent valides, il faut cependant modifier les messages à transmettre dans le réseau : grossièrement, la taille des messages qui devraient transiter dans la forêt obtenue après la coupe est multipliée par celle des nœuds dont sont issus les arcs éliminés. Le coupe-cycle est donc un procédé coûteux à la fois en taille mémoire et en temps de calcul.

Spiegelhalter et Lauritzen, quant à eux, commencent par une modification importante du graphe dont les étapes sont représentées par la figure 7(b) : moralisation (on relie les parents d'un même nœud) et désorientation (on retire les directions des arcs), puis triangulation (on réduit tout cycle du graphe en cycles élémentaires de 3 nœuds en ajoutant des arcs si besoin).

Ce graphe modifié permet de construire un nouveau graphe appelé "arbre de jonction" dans lequel la propagation de messages est simplifiée. Cet arbre s'obtient de la manière suivante : les cliques<sup>8</sup> du graphe triangulé représentent les nœuds de l'arbre de jonction. Ce sont donc des agrégats de nœuds du graphe de départ. Les nœuds de l'arbre de jonction sont reliés entre eux lorsque les cliques correspondantes dans le graphe triangulé ont des nœuds en commun. Une propagation dans l'arbre de jonction permet d'obtenir les probabilités des cliques. Il suffit alors d'effectuer des marginalisations pour obtenir les probabilités des variables du RB de départ. Par exemple, s'il existe une clique  $\{A, B, C\}$  dans le graphe triangulé, cette clique correspond à un seul nœud dans l'arbre de jonction et, après propagation, on connaît  $\text{Prob}(A, B, C)$ . Pour déterminer  $\text{Prob}(A = a)$ , il suffit d'appliquer la formule :

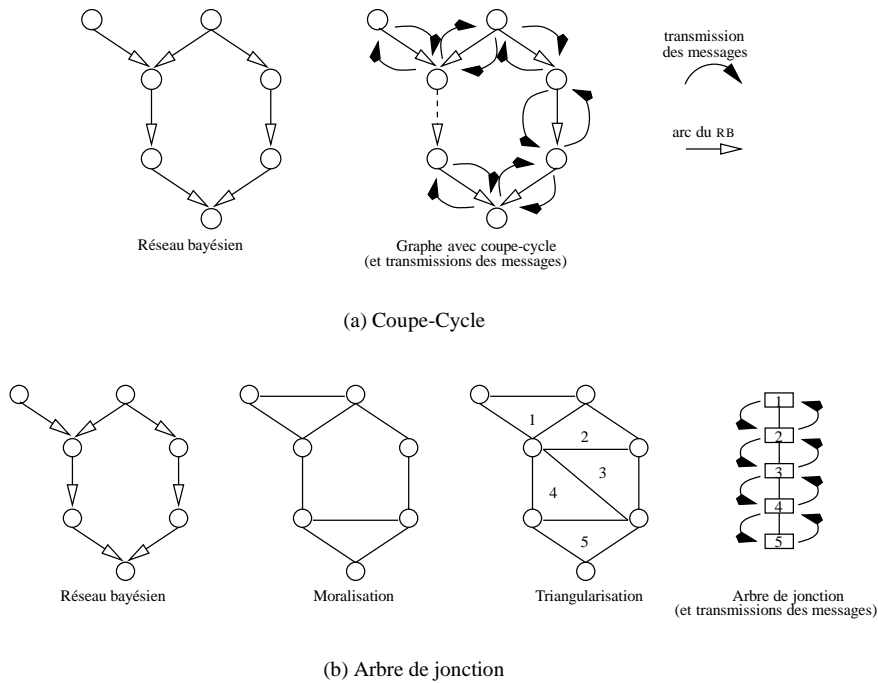
$$\text{Prob}(A = a) = \sum_i \sum_j \text{Prob}(A = a, B = b^i, C = c^j).$$

Cette transformation du RB de départ, plus complexe que celle du coupe-cycle et un peu réductrice, est pourtant la plus étudiée parce qu'elle a le mérite d'être générique et au final, plus simple à mettre en oeuvre. Les études actuelles vont dans le sens d'une

<sup>6</sup> Une chaîne est un ensemble de nœuds  $\{X_{i_1}, \dots, X_{i_k}\}$  tel que les arcs  $X_{i_j} \rightarrow X_{i_{j+1}}$  ou  $X_{i_{j+1}} \rightarrow X_{i_j}$  existent pour tout  $j \in \{1, \dots, k-1\}$ .

<sup>7</sup> Un circuit est un chemin (ensemble de nœuds  $\{X_{i_1}, \dots, X_{i_k}\}$  tel que deux nœuds consécutifs de cet ensemble sont reliés par un arc  $X_{i_j} \rightarrow X_{i_{j+1}}$  qui va d'un nœud à lui-même. Dans un RB pris au sens causal, un circuit indiquerait qu'une variable est "cause d'elle-même"! Il est donc raisonnable de se placer dans le cas de graphes sans circuit (appelés encore DAG, Directed Acyclic Graphs).

<sup>8</sup> Une clique est un sous-graphe complet maximal (ensemble de nœuds  $\{X_{i_1}, \dots, X_{i_k}\}$  tel que, quel que soit  $j, X_{i_j}$  est relié à tous les  $X_{i_p}, p \in \{1, \dots, k\} \setminus \{j\}$ ).



**Figure 7.** Modifications d'un RB par les différentes méthodes de calculs

équivalence théorique des deux méthodes. Cependant les phases cruciales de l'optimisation de ces propagations (d'un côté la recherche d'un coupe-cycle optimal, de l'autre celle d'une triangulation optimale) rendent difficile les comparaisons d'implémentations des deux calculs.

Actuellement, des études sont toujours en cours pour continuer à optimiser les chaînes de calculs, en tentant par exemple de prendre en considération le type d'information à propager ou encore les spécificités topologiques du graphe (et donc du modèle d'indépendance sous-jacent) pour n'effectuer l'inférence probabiliste que sur la partie du réseau influencée par la nouvelle observation (cf. [FAÏ 97]).

Une chose est certaine, la propagation exacte permet aujourd'hui d'effectuer des inférences dans le cas de graphes de grande taille et est à prendre en considération avant d'effectuer des recherches d'algorithmes approximatifs. Comme le montre [JEN 96] avec l'exemple de MUNIN, RB d'un peu plus de 1000 nœuds, des problèmes dont la complexité était telle que la résolution semblait impossible dans les années 70 sont traités aujourd'hui en l'espace de quelques secondes.



## 4.2. Méthodes approchées

Seuls les graphes très complexes, notamment ceux qui comportent beaucoup de cycles, doivent encore utiliser des algorithmes approchés. Ceux-ci sont principalement de deux types : i) ceux qui utilisent des méthodes exactes mais opèrent seulement sur une partie du graphe ; ii) ceux qui utilisent des méthodes stochastiques (simulations).

### 4.2.1. Méthodes exactes sur des topologies approchées

Ces méthodes sont relativement récentes et sont globalement réparties en deux écoles distinctes. La première, suivie par [KJA 93a] et [KJA 94], exploite le fait que certaines dépendances du réseau sont faibles, c'est-à-dire que, qualitativement, il existe un arc entre des nœuds  $X$  et  $Y$  parce que ces variables ne sont pas exactement indépendantes l'une de l'autre, mais que, quantitativement, cette dépendance est insignifiante ; autrement dit, les variables  $X$  et  $Y$  se comportent presque comme si elles étaient indépendantes. L'idée de l'algorithme de propagation est alors d'éliminer de tels arcs, rendant ainsi  $X$  et  $Y$  indépendantes : les calculs en sont accélérés, la taille des matrices de probabilité conditionnelle en est réduite et l'erreur engendrée reste raisonnable (Jensen cite un exemple dans lequel il réduit les temps de calcul drastiquement tout en limitant l'erreur à moins de 5%).

La philosophie de la deuxième école est légèrement différente : il s'agit de conserver le graphe d'origine, mais de n'effectuer la propagation des informations que "partiellement". Là encore, plusieurs méthodes sont utilisées : [HOR 89] et [DAM 93] réalisent la propagation dans la totalité du réseau, mais n'utilisent que des sous-parties des matrices de probabilité conditionnelle. [DAM 93], par exemple, suppose que les lois de probabilité des variables sont "quasi-certaines", c'est-à-dire que les variables possèdent une valeur ayant une probabilité beaucoup plus élevée que les autres.

### 4.2.2. Méthodes stochastiques

Pour traiter les RB complexes, hormis les modifications de topologie décrites ci-dessus, il existe aussi un ensemble de méthodes reposant sur des principes stochastiques.

Une étude statistique classique consiste souvent à rechercher les paramètres de la loi  $\pi$  suivie par un processus en utilisant une base de données (échantillons) qui permet de calculer des estimateurs approchés des différents paramètres de  $\pi$  (moyenne, écart-type, etc.). Bien sûr, cette approximation est d'autant meilleure que la taille de la base est importante. L'idée de départ des méthodes stochastiques est donc d'utiliser ce que l'on connaît de la loi étudiée pour générer automatiquement des échantillons d'une base de données représentative de cette loi (génération d'exemples) ; c'est donc bien de la simulation. Il suffit alors d'utiliser cette base simulée pour calculer les différents estimateurs. À partir de ce même principe, différentes méthodes sont apparues, qui se distinguent par leur façon de mener les simulations, de générer la base d'exemples en fonction de différentes connaissances de la loi étudiée. Citons par exemple, les méthodes dites "probabilistic logic sampling" ([HEN 88]), les méthodes "MCMC" (Markov Chain Monte Carlo). Plus précisément, les MCMC sont une famille de méthodes

stochastiques comprenant entre autre Metropolis ([MET 53] ou [GIL 96]) et échantillonneur de Gibbs ([NEA 93]).

Il est intéressant de noter que l'implémentation de l'échantillonneur de Gibbs est particulièrement aisée dans le cadre des RB. En effet, il devient extrêmement simple et se réduit au choix à chaque itération d'une variable dont on change la valeur en fonction des valeurs de son entourage ([PEA 87], [YOR 92]).

### 4.3. Réseaux dynamiques

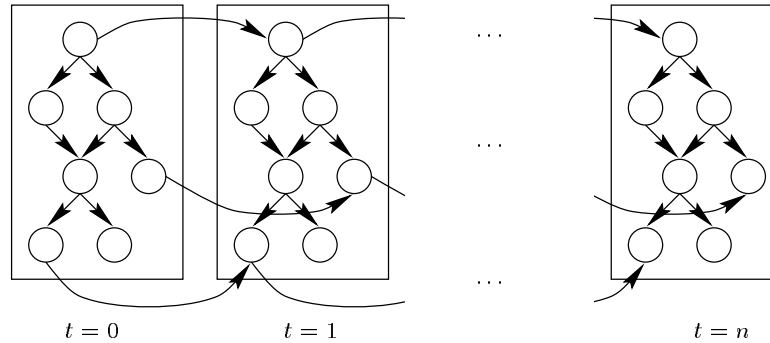
Dans des études où l'aspect dynamique du système est important, le caractère essentiellement atemporel des modèles graphiques probabilistes est un défaut parfois important. En effet, si une variable est observée, ce doit être, en toute rigueur, une fois pour toute, et sa valeur ne peut être remise en question, alors que dans des modèles temporels, les valeurs des observations de l'instant  $t - 1$  et de l'instant  $t$  doivent toutes deux être prises en compte, l'une influençant l'autre.

L'idée la plus simple pour pallier cela consiste à dupliquer les variables du réseau autant de fois qu'il y a d'instant  $t$  : ainsi, on peut remplacer dans le RB une variable (un nœud)  $A$  par un ensemble de nœuds  $A_t, A_{t+1}, \dots, A_{t+p}$  symbolisant  $A$  aux différents instants  $t_i$ . Les arcs du RB permettent en outre de représenter les liens existant entre la valeur de  $A$  à un instant  $t$  et celle prise par  $A$  à l'instant  $t'$ . Ce type de RB permet de prendre en compte des effets temporels, mais son aspect statique (le réseau est créé une fois pour toutes) limite toutefois considérablement son champ d'application : il ne peut ainsi prendre en compte qu'un nombre fixe de  $t_i$ . La temporalité du réseau ne peut être que finie.

Une répétition d'observations est pourtant un cas qui se présente couramment (par exemple, tenter de prendre en compte l'évolution de l'apprenant durant l'apprentissage). Pour résoudre ce problème, une solution proposée, entre autres, par [KJA 93b] est d'utiliser les réseaux probabilistes dynamiques ("Dynamic Bayesian Networks") ou RBD.

Un RBD (voir la figure 8) est un ensemble de RB représentant le système à différents instants. On y ajoute des arcs représentant les relations temporelles. Le réseau constitue ainsi une représentation du modèle durant un certain nombre de périodes (dans la figure 8 par exemple, il y a  $n$  périodes), cette fenêtre temporelle sera déplacée pour permettre la prévision d'actions futures (à  $n + 1$  et après).

Ce modèle permet donc de gérer une série virtuellement infinie d'observations du système, contrairement aux modèles statiques, tout en gardant une bonne flexibilité quant à la taille de la fenêtre temporelle (et donc à la complexité du schéma de propagation). La propagation dans un tel réseau est bien sûr rendue plus complexe, puisque les dépendances sont plus nombreuses. Cependant, des méthodes à ce jour approximatives (à base de simulation type Monte Carlo) permettent d'utiliser pratiquement les RBD.



**Figure 8.** Exemple de réseau bayésien dynamique

#### 4.4. Apprentissage de la structure

Dans tout ce qui précède, les relations entre les variables, c'est-à-dire la topologie du RB, étaient considérées comme connues, spécifiées par un expert ou par le bon sens. Pour au moins certaines parties du réseau, en Modélisation de l'Utilisateur, cet a priori semble cohérent. En effet, les notions caractérisant l'utilisateur dans le RB et leurs relations sont des constructions de l'enseignant ou du domaine, rarement des données objectives. Par exemple, les concepts de "connaissance du domaine" et de "connaissance de sous-domaine" ainsi que leurs relations (supposées souvent hiérarchiques) sont créés par l'enseignant. On peut toutefois considérer que les relations entre variables dans un sous-système observable du RB ne sont pas parfaitement définies et méritent d'être apprises automatiquement (surtout si elles sont susceptibles de changer). Les méthodes d'apprentissage de la topologie méritent donc d'être au moins citées dans cet article.

Il semble un peu difficile, sans être trop technique, de rentrer dans le détail des méthodes actuelles. On peut simplement indiquer trois voies principales de recherche dans le domaine :

1. méthodes bayésiennes : il s'agit d'essayer de maximiser la probabilité d'une topologie conditionnellement à une base de données. Elles sont souvent inspirées par des recherches parallèles dans le domaine du "Machine Learning". On peut citer par exemple l'algorithme K2 ([COO 92]), algorithme (gourmand) d'ajouts d'arcs (en partant d'un graphe sans arc : toutes les variables sont supposées indépendantes). À l'inverse, il existe des méthodes fondées sur l'entropie, c'est-à-dire la perte d'information lors de la suppression d'arcs à partir d'un graphe complet ([HER 90]).
2. méthodes quasi-bayésiennes : ce sont des méthodes essayant, elles aussi, d'améliorer l'ajustement du modèle graphique aux données. Cependant, la fonction d'évaluation de cet ajustement n'est plus une probabilité de topologie conditionnellement à des données ; c'est, par exemple, le critère MDL (Minimal Description Length, [RIS 89]). Le but est ici de trouver un compromis entre complexité

du modèle et ajustement de la base ([LAM 93], [BOU 93]).

3. autres méthodes : ([SPI 93], [FUN 90], ou encore [PEA 91]). On utilise ici des principes statistiques (recherches statistiques d'indépendance par des méthodes type  $\chi^2$ ) ou autres (algorithme EM, recherche de causalité, ...).

Des progrès importants et des recherches de plus en plus nombreuses se font dans le champ de l'apprentissage direct de la topologie du graphe. Il est encore trop tôt pour décider quelle est la voie la plus prometteuse. Les méthodes bayésiennes ont l'avantage de l'exactitude de leurs solutions, mais l'inconvénient de temps de calculs souvent prohibitifs (pour estimer les probabilités de structure conditionnellement à une base de données, on peut être obligé de traiter le cas de tous les réseaux possibles). D'un autre côté, les autres méthodes clament l'inutilité d'une exactitude complète sur une base de données qui ne peut pas être parfaite et préfèrent parler de compromis entre adéquation du modèle de la base, d'une part, et simplicité du modèle d'autre part. L'intérêt pour l'apprentissage est en tout cas réel, autant pour son aptitude à découvrir les relations entre les variables d'une base de données que pour pouvoir créer des réseaux opérationnels. Les méthodes d'apprentissage automatique risquent donc de se révéler intéressantes pour le champ entier des applications des réseaux bayésiens.

#### 4.5. Applications disponibles

Pour être efficace, les RB requièrent des implémentations fines. Aussi le temps nécessaire à la programmation des inférences probabilistes dans un RB peut être un point pénalisant pour l'utilisation de tels outils. Cependant, un nombre assez important de logiciels existent aujourd'hui. Leurs différences sont nombreuses et il n'est pas toujours facile de trouver le programme adéquat à ses besoins. Outre le langage de programmation utilisé (qui peut avoir son importance pour l'utilisateur), les plate-formes et les fonctionnalités sont variées, autant du point de vue des calculs menés, que de ceux de l'interface utilisateur et que du type de réseaux gérés. Il faut aussi distinguer les logiciels offerts dans le domaine public et les logiciels payants (cependant, la plupart des logiciels commerciaux ont des versions de démonstration gratuites utilisables dans de nombreux cas). La plupart des logiciels utilise une méthode fondée sur celle de l'arbre de jonction. Elles ajoutent toutes des extensions aux RB simples : soit elles acceptent des lois continues (paramétrées), soit elles acceptent des probabilités imprécises (Belief Function), soit elles étendent les réseaux bayésiens dans le domaine de la décision (Influence Diagram) et peuvent donc intégrer des nœuds de décision et des notions d'utilité (pour plus d'informations, voir [KRA 71], [WAK 89] ou [FIS 70]).

La liste de logiciels présentée ci-dessous ne se veut pas exhaustive. Ce sont en fait des extraits commentés de la liste maintenue par Russel G. Almond<sup>9</sup> et de celle de l'Air Force Institute of Technology<sup>10</sup>. Tous les logiciels listés sont capables de gérer des réseaux probabilistes simples et de mener des inférences. HUGIN est l'un des produits commerciaux de référence dans le domaine, tant par ses capacités que par ses

<sup>9</sup> voir <http://bayes.stat.washington.edu/almond/belief.html>

<sup>10</sup> voir <http://www.afit.af.mil/Schools/EN/ENG/LABS/AI/BayesianNetworks/tools3.html>

Logiciel	Système / machine	Remarques	Adresse
Graphical-Belief	Common Lisp		<a href="http://bayes.stat.washington.edu/almond/gb/graphical-belief.html">http://bayes.stat.washington.edu/almond/gb/graphical-belief.html</a>
HUGIN	SunOS, Sun Solaris, Windows 95, NT (pour les dernières versions)	Produit commercial "phare". Version de démonstration	<a href="http://www.hugin.dk/">http://www.hugin.dk/</a>
IDEAL	Common Lisp (interface GLIM)	Utilisation non-commerciale	<a href="http://www.rpal.rockwell.com/ideal.html">http://www.rpal.rockwell.com/ideal.html</a>
MacEvidence	MacIntosh		Prakash P. Shenoy School of Business University of Kansas Summerfield Hall Lawrence, KS 66045-2003 USA
MSBN32	Windows 95, Windows NT	Utilisation non-commerciale, licence requise	Microsoft Decision Theory Group <a href="http://www.research.microsoft.com/dtas/msbn/default.htm">http://www.research.microsoft.com/dtas/msbn/default.htm</a> dtg-msbn@microsoft.com
Netica	Windows 95/NT, Macintosh, UNIX	Commercial mais version de démonstration gratuite	<a href="http://www.norsys.com/netica.html">http://www.norsys.com/netica.html</a>
Pulcinella	Common Lisp	Utilisation non-commerciale	Alessandro Saffiotti (asaffio@ulb.ac.be) IRIDIA, Université Libre de Bruxelles 50 av. F. Roosevelt, BP 194-6 1050 Brussels, Belgium
Strategist	Windows 95/NT	Commercial	<a href="http://www.prevision.com/strategist.html">http://www.prevision.com/strategist.html</a>

**Tableau 2.** Logiciels sur les réseaux bayésiens

nombreuses applications et extensions ; il possède une version de démonstration très élaborée.

Enfin ajoutons qu'est développé actuellement au LIP6 (thème SYSDEF) par notre équipe un produit multi-plateforme (Windows 95, NT, stations UNIX), MOBYDICK, qui se veut capable des mêmes genres de fonctionnalités que ces logiciels, en insistant particulièrement sur l'ouverture et la facilité de modification des méthodes de calculs et donc sur la capacité d'adapter facilement le logiciel à des besoins spécifiques.

## Conclusion

Les réseaux bayésiens sont un outil souple et performant de gestion de l'incertitude :

- ils permettent de séparer aspects qualitatifs et aspects quantitatifs des modèles probabilistes grâce à leur présentation graphique et fournissent ainsi des méthodes simples de construction ;
- ils ont la souplesse suffisante pour fusionner différentes sources d'informations (connaissances à partir de bases de données ou d'experts) ;
- ils ont la capacité d'intégrer les modifications de l'environnement d'étude ;
- ils peuvent gérer des séquences temporelles (virtuellement infinies) d'évaluations et de prises de décision ;
- ils permettent de grandes rapidités de réaction car, bien que les propagations requièrent des calculs assez fins, ceux-ci sont bien explicités dans la littérature et peuvent être effectués en temps réel.

C'est pourquoi cet outil, très structuré mathématiquement, offre des voies nouvelles dans les domaines de l'évaluation et de l'accompagnement automatisé de l'apprenant. Il devrait donc prendre une place privilégiée au sein des techniques d'Intelligence Artificielle susceptibles d'être applicables en Modélisation de l'Utilisateur.

Les RB sont encore un domaine de recherche<sup>11</sup> mais ils sont d'ores et déjà à l'œuvre dans de nombreux systèmes opérationnels<sup>12</sup>. Il nous semble que leur richesse et leur souplesse devraient leur assurer un large développement dans les années à venir.

## Références

[ALB 97] ALBRECHT, D., ZUKERMAN, I., NICHOLSON, A., BUD, A., "Towards a bayesian model for keyhole plan recognition in large domains", *User Modeling : Proceedings of the Sixth International Conference, UM'97*, Jameson, A., Paris, C., et Tasso, C., éditeurs, Springer.

---

<sup>11</sup> C'est de plus un domaine très actif ; par exemple, la moitié des interventions dans "Uncertainty in AI" sont consacrées depuis quelques années à ce domaine.

<sup>12</sup> Par exemple, dans le contrôle de qualité des processeurs Intel, pour les pannes des systèmes de propulsion des navettes spatiales de la NASA,...

- [BOU 93] BOUCKAERT, R., “Probabilistic network construction using the mdl principle”, *ECSQARU'93*, p. 41-48.
- [CHI 89] CHIN, D., “KNOME : Modeling what the user knows in UC”, *User Models in Dialog Systems*, 1989, Kobsa, A. et Wahlster, W., éditeurs, Springer, Berlin, p. 74-107.
- [CON 97] CONATI, C., GERTNER, A., VANLEHN, K., DRUDZEL, M. J., “On-line student modeling for coached problem solving using bayesian networks”, *User Modeling : Proceedings of the Sixth International Conference, UM'97*, Jameson, A., Paris, C., et Tasso, C., éditeurs, Vienna, New York. Springer.
- [COO 88] COOPER, G. F., “Probabilistic inference using belief network is np-hard”, Technical Report KSL-87-27, 1988, Medical Computer Science, Stanford University, Stanford, California.
- [COO 92] COOPER, G., HERSKOVITS, E., “A bayesian method for induction of probabilistic networks from data”, *Machine Learning*, vol. 9, 1992, p. 309-347.
- [COR 92] CORBETT, A. T., ANDERSON, J. R., “Student modeling and mastery learning in a computer-based programming tutor”, *Proceedings of ITS'92*, Frasson, C., Gauthier, C., et McCalla, G. I., éditeurs, p. 413-420.
- [DAM 93] D'AMBROSIO, B., “Incremental probabilistic inference”, *Proceedings of the ninth Conference on Uncertainty in Artificial Intelligence*, 1993, Morgan Kaufmann, p. 301-308.
- [FAÿ 97] FAÿ, A., “Sur la propagation de l'information dans les réseaux probabilistes”, Thèse de doctorat de l'université paris VI.
- [FIS 70] FISHBURN, P. C., *Utility Theory for Decision Making*, Wiley, New York, 1970.
- [FOR 95] FORBES, J., HUANG, T., KANAZAWA, K., RUSSELL, S., “The batmobile : Towards a bayesian automated taxi”, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1995, p. 1878-1885.
- [FUN 90] FUNG, R., CRAWFORD, S., “Constructor : A system for the induction of probability models”, *Proceedings of AAAI*, 1990, p. 762-769.
- [GIL 96] GILKS, W., RICHARDSON, S., SPIEGELHALTER, D., *Markov Chain Monte Carlo in Practice*, Interdisciplinary Statistics. Chapman & Hall, 1996.
- [HEN 88] HENRION, M., “Propagating uncertainty in bayesian networks by probabilistic logic sampling”, *proceedings Uncertainty in Artificial Intelligence*, vol. 4, 1988, North Holland, Amsterdam, p. 149-163.
- [HER 90] HERSKOVITS, E., COOPER, G., “Kutató : An entropy-driven system for construction of probabilistic expert systems from databases”, *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 1990, Bonissone, Henrion, Kanal, et Lemmer, éditeurs, p. 117-125.

- [HOR 89] HORVITZ, E., SUERMONDT, J., COOPER, G., “Bounded conditioning : Flexible inference for decisions under scarce resources”, *Proceedings of the fifth Conference on Uncertainty in Artificial Intelligence*, 1989, North Holland, p. 182-193.
- [HOR 97] HORVITZ, E., “Agents with beliefs : Reflections on bayesian methods for user modeling”, *User Modeling : Proceedings of the Sixth International Conference, UM'97*, Jameson, A., Paris, C., et Tasso, C., éditeurs, Vienna, New York. Springer.
- [JAM 96] JAMESON, A., “Numerical uncertainty management in user and student modeling : An overview of systems and issues” *User Modeling and User-Adapted Interaction*, vol. 5, 1996, p. 193-251.
- [JEN 96] JENSEN, F., *An Introduction to Bayesian Networks*, Springer-Verlag, North America, 1996.
- [KAT 92] KATZ, S., LESGOLD, A., EGGAN, G., GORDIN, M., “Modeling the student in sherlock II”, *Journal of Artificial Intelligence in Education*, vol. 3, 1992, p. 495-518.
- [KJA 93a] KJAERULFF, U., “Approximation of bayesian networks through edge removals”, Research Report IR-93-2007, 1993, Department of Mathematics and Computer Science, Aalborg University, Denmark.
- [KJA 93b] KJAERULFF, U., “A computational scheme for dynamic bayesian networks”, Technical Report DK-9220, 1993, Department of Mathematics and Computer Science - Institute for Electronic Systems, Aalborg, Denmark.
- [KJA 94] KJAERULFF, U., “Reduction of computational complexity in Bayesian networks through removal of weak dependences”, *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, 1994, Association for Uncertainty in Artificial Intelligence.
- [KRA 71] KRANTZ, D. H., LUCE, R. D., SUPPES, P., TVERSKY, A., *Foundations of Measurement (Additive and Polynomial Representations)*, vol 1. Academic Press, New York, 1971.
- [LAM 93] LAM, W., BACCHUS, F., “Using causal information and local measures to learn bayesian networks”, *Proceedings of Ninth Conference on UAI*, 1993, p. 243-250.
- [LAU 88] LAURITZEN, S., SPIEGELHALTER, D., “Local computations with probabilities on graphical structures and their application to expert systems”, *The Journal of The Royal Statistical Society – Series B (Methodological)*, vol. 50, n° 2, 1988, p. 157-224.
- [MAR 95] MARTIN, J., VANLEHN, K., “A Bayesien approach to cognitive assessment”, *Cognitively Diagnostic Assessment*, 1995, Nichols, P., Chipman, S., et Brennan, R. L., éditeurs, Hillsdale, NJ. LEA.



- [MET 53] METROPOLIS, N., ROSENBLUTH, A., ROSENBLUTH, M., TELLER, A., TELLER, E., "Equation of state calculations by fast computing machines", *Journal of Chemical Physics*, vol. 21, 1953, p. 1087-1092.
- [MIS 96] MISLEVY, R., GITOMER, D., "The role of probability-based inference in an intelligent tutoring system", *User Modeling and User-Adapted Interaction*, vol. 5, 1996, p. 253-282.
- [NEA 90] NEAPOLITAN, R., *Probabilistic Reasoning in Expert Systems : Theory and Algorithms*, Wiley, New York, 1990.
- [NEA 93] NEAL, R., "Probabilistic inference using markov chain monte carlo methods", Technical Report CRG-TR-93-1, 1993, Department of Computer Science, University of Toronto.
- [PEA 87] PEARL, J., "Evidential reasoning using stochastic simulation of causal models", *Artificial Intelligence*, vol. 32, 1987, p. 245-257.
- [PEA 88] PEARL, J., *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*, Morgan Kaufman, 1988.
- [PEA 91] PEARL, J., VERMA, T., "A theory of inferred causation", *Knowledge Representation and Reasoning - Proceedings of the Second International Conference*, 1991, p. 441-452.
- [PET 93] PETRUSHIN, V., SINITSIA, K., "Using probabilistic reasoning techniques for learner modeling", *Artificial Intelligence in Education : Proceedings of AI-ED 93*, Brna, P., Ohlsson, S., et Pain, H., éditeurs, p. 418-425.
- [PYN 95] PYNADATH, D. V., WELLMAN, M. P., "Accounting for context in plan recognition, with application to traffic monitoring", *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995, p. 472-481.
- [REY 96] REYE, J., "A belief net backbone for student modelling", *Intelligent Tutoring Systems : Proceedings of the Third International Conference, ITS'96*, Frasson, C., Gauthier, G., et Lesgold, A., éditeurs, Springer.
- [RIS 89] RISSANEN, J., "Stochastic complexity on statistical inquiry", *World Scientific*, 1989.
- [SCH 97] SCHÄFER, R., WEYRATH, T., "Assessing temporally variable user properties with dynamic bayesian networks", *User Modeling : Proceedings of the Sixth International Conference, UM'97*, Jameson, A., Paris, C., et Tasso, C., éditeurs, Vienna, New York. Springer.
- [SPI 93] SPIRITES, P., MEEK, C., "Learning bayesian network with discrete variables from data", *Proceedings of First International Conference on Knowledge Discovery and Data Mining*, 1993, Montreal, QU.

[WAK 89] WAKKER, P. P., *Additive Representations of Preferences, A New Foundation of Decision Analysis*, Kluwer Academic Publishers, Dordrecht, 1989.

[YOR 92] YORK, J., "Use of gibbs sampler in expert systems", *Artificial Intelligence*, vol. 56, 1992, p. 115-130.

**Pierre-Henri Willemin** est ingénieur informaticien. Il est actuellement allocataire de recherche et prépare une thèse qu'il effectue au PoleIA (LIP6) sous la direction du professeur J.Y. Jaffray (Paris VI). Ses recherches portent sur les Réseaux Probabilistes d'un point de vue théorique, en particulier sur l'optimisation des algorithmes d'inférence probabiliste et de propagations de messages. Il est aussi le concepteur et développeur de MOBYDICK, projet en cours de modélisation de Réseaux Probabilistes et d'études d'algorithmes de propagation/apprentissage.

**Christophe Gonzales** est maître de conférences à l'université Paris VI. Après une école d'ingénieur en informatique et un DEA de Recherche Opérationnelle à Paris VI, il a effectué une thèse sous la direction du professeur J.Y. Jaffray sur les fonctions d'utilité additives. Il travaille maintenant au sein de l'équipe J.Y. Jaffray sur des aspects théoriques des réseaux bayésiens.