

## Modélisation de réseaux bayésiens de très grandes tailles

Lionel Torti<sup>1</sup> et Pierre-Henri Wuillemin<sup>1</sup>

1 : Laboratoire d'informatique de Paris 6, 104 avenue du Président Kennedy, 75016 Paris.

Contact : lionel.torti@lip6.fr

---

### Résumé

La représentation de connaissances incertaines est un problème important dans le domaine de l'intelligence artificielle. Les réseaux bayésiens proposent une solution intéressante pour de nombreuses raisons théoriques et pratiques mais ont le désavantage d'être difficiles à créer autant qu'à maintenir. Dans cet article, nous comparons trois formalismes qui tentent de répondre à ces problèmes en enrichissant les réseaux bayésiens avec des notions issues des paradigmes orientés objets et de la logique du premier ordre. Nous verrons que ces propositions fournissent de nouveaux outils de modélisation des réseaux bayésiens, mais posent de nouveaux problèmes propres aux réseaux de grande taille.

### Abstract

The representation of uncertainty is a challenge in the domain of artificial intelligence. The Bayesian networks are a tool which solves this problem and which has many advantages, both theoretical and practical. However large bayesian networks are difficult to design and to maintain. The goal of this paper is to provide a comparison of three formalism attempting to solve these difficulties with concepts taken from the oriented object paradigm and first order logic. However even if these formalisms give new tools to build bayesian networks, they create new problems centered over large scale networks.

**Mots-clés :** réseau bayésien, PRM, MEBN, OOBN, modélisation

**Keywords:** bayesian network, PRM, MEBN, OOBN, modelisation

---

La représentation de connaissances incertaines est un enjeu de taille dans un grand nombre de domaines tels que la prise de décision automatisée, la recherche dans des bases de connaissances importantes, le diagnostic, etc. Les réseaux bayésiens (ou encore *Bayesian Networks*, noté "BN" par la suite) offrent un outil de modélisation et de gestion des incertitudes à la fois souple et performant qui répond en grande partie à ces besoins. Depuis plus de vingt ans, la modélisation probabiliste est en expansion constante et offre un cadre de plus en plus couramment utilisé aussi bien dans le domaine de l'intelligence artificielle, en tant que *système expert probabiliste*, qu'en statistique comme outil de modélisation.

Les BN sont utilisés dans un grand nombre de domaines tels que le diagnostic, le contrôle et le *planning*, l'analyse statistique de données. Ils peuvent être utilisés également comme généralisation des filtres de Kalman pour le contrôle, les chaînes de Markov cachées et donc la reconnaissance de la parole, les systèmes dynamiques, la révision, pour la recherche de connaissance et l'extraction de donnée ou *Data Mining*, etc.

En terme d'utilisation industrielle, les réseaux bayésiens connaissent un succès indéniable dans les domaines mentionnés précédemment. La maîtrise de plus en plus avancée de ce formalisme par les industriels entraîne un passage à l'échelle de la taille des BN modélisés et utilisés.

Or ce passage à l'échelle requiert des techniques de modélisations difficilement réalisables avec des BN. D'un point de vue général, une phase de modélisation peut être très complexe et doit pouvoir prendre en compte la modularité du système étudié. L'étude d'une série de sous-systèmes reliés entre eux est plus simple que l'étude d'un système monolithique. Cette décomposition permet

aussi de définir différentes “granularités” ou niveaux de visualisation, d’intégration. Le concepteur a alors le choix entre visualiser globalement son système ou entrer dans les détails de la modélisation d’un des sous-systèmes (qui peut lui-même posséder récursivement d’autres niveaux d’intégration, etc).

Les réseaux bayésiens souffrent ici d’une grande faiblesse. La structure d’un BN est un graphe qui n’est pas “hiérarchisable” dans le sens où il n’y a pas de moyen direct d’enrichir la représentation, ce qui en fait une structure “plate”.

Le problème ainsi posé est analogue aux interrogations sur les moyens d’une gestion rationnelle des programmes informatiques : comment hiérarchiser le code de manière à le localiser, à le développer, à le “réutiliser” et à le maintenir facilement ? Ces interrogations ont mené à la création des langages structurés puis des langages à objets.

Notons que pour des raisons historiques le terme *réseau bayésien orienté objet*, [9] et [4], est peu utilisé au profit des termes *modèles relationnels* [8], ou encore *modèle probabiliste du premier ordre* [3]; ces modèles intègrent tous à des degrés différents le paradigme orienté objet.

Après une brève introduction sur les BN, nous passerons en revue trois de ces modèles en mettant en avant les critères suivants :

1. l’amélioration du processus d’ingénierie (*authoring*) des modèles, en facilitant la gestion de la complexité, la réutilisation des composants et le travail collaboratif ;
2. l’augmentation du pouvoir expressif (i.e. de la richesse et de la flexibilité) du cadre de modélisation disponible, notamment pour la modélisation des systèmes dont la structure précise et les frontières ne seront complètement connues qu’au moment de l’exploitation (nombre de composants, configurations) ;
3. l’exploitation de ces modèles (en inférence, apprentissage automatique, planification, etc.) en présence de ressources (mémoire et puissance/temps de calcul) limitées.

## 1. Introduction

Les BN reposent sur une représentation de la connaissance utilisant la théorie des graphes orientés. Un graphe orienté  $G$  est caractérisé par la donnée des ensembles  $V_G$  et  $E_G \subset V_G \times V_G$ . On note  $X \rightarrow Y$  un élément de  $E_G$ , où  $X$  et  $Y$  sont des éléments de  $V_G$ .  $V_G$  est l’ensemble des nœuds du graphe,  $E_G$  l’ensemble de ses arcs. Pour un arc  $X \rightarrow Y$ , “le nœud  $X$  est parent du nœud  $Y$ ” et “le nœud  $Y$  est fils du nœud  $X$ ”. Enfin, un circuit est une séquence d’arcs  $X_1 \rightarrow X_2 \rightarrow \dots \rightarrow X_{i-1} \rightarrow X_i$  telle que  $X_1 = X_i$ .

Formellement, un réseau bayésien est un graphe dont les nœuds représentent des variables aléatoires et dont les arcs expriment des interdépendances, des influences entre ces variables. Pour quantifier l’importance de ces influences, à chaque nœud est adjointe sa loi de probabilité conditionnellement à ses parents dans le graphe. Le graphe modélise qualitativement des influences entre variables aléatoires. C’est l’absence d’arc entre deux variables qui indique l’absence d’influence directe entre elles. Le graphe est supposé sans circuit (en anglais, *DAG* : *Directed Acyclic Graph*). L’orientation des arcs n’a pas de sens causal. En fait, elle dépend plus d’un ordre arbitraire d’énumération des variables lors de la création du réseau que de la causalité :  $X \rightarrow Y$  signifie plutôt “ $X$  a été énuméré avant  $Y$  et il existe une influence entre  $X$  et  $Y$ ” que “ $X$  est cause de  $Y$ ” – d’où l’impossibilité de circuit dans le graphe.

La structure du graphe de la figure 1 suffit pour savoir que dans ce modèle :

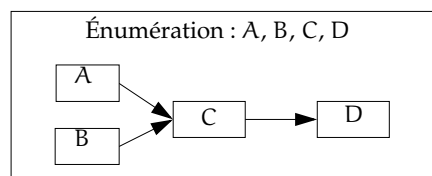


FIGURE 1 – Un réseau bayésien. Au nœud  $C$ , par exemple, est associée la probabilité  $P(C|A, B)$ .

- A et C s’influencent directement alors qu’il n’existe aucune influence directe entre A et B.
- On ne peut conclure que A est cause de C. On ne peut pas non plus garantir qu’avec un autre ordre d’énumération, B n’influencerait pas directement D.
- Il est possible d’extraire des connaissances qualitatives plus fines de cette représentation : les indépendances conditionnelles (voir [2]). Dans la figure 1,  $A \perp D|C$  (A est indépendant de D conditionnellement à C) indique que toute l’information que la connaissance de A peut apporter sur D est contenue dans la donnée de C.

Les indépendances conditionnelles sont aussi importantes d’un point de vue calculatoire puisque ce sont elles qui permettent de factoriser la probabilité jointe de l’ensemble du système.

Reste donc à définir l’aspect quantitatif du modèle. À tout nœud du graphe est associé une table de probabilité qui définit localement la loi de ce nœud, conditionnellement à ses parents dans le graphe. Cet ensemble de lois locales permet de définir la loi jointe comme un produit de tables de petites tailles (relativement à celle de la loi jointe). Dans la figure 1, on calcule la probabilité jointe  $P(A, B, C, D)$  par  $P(A).P(B).P(C|A, B).P(D|C)$ . Plus généralement, si  $V$  est l’ensemble des nœuds du graphe :

$$P(V) = \prod_{N \in V} P(N | \pi_N)$$

où  $\pi_N$  est l’ensemble des parents du nœud  $N$  dans le graphe. Cette propriété est si importante pour l’intérêt des réseaux bayésiens que ceux-ci sont parfois considérés avant tout comme un moyen de factoriser une loi jointe d’un grand nombre de variables<sup>1</sup>. En effet, avant l’émergence de cet outil, la manipulation de modèles probabilistes de grande taille n’était pas pensable.

## 2. Approches actuelles

La capacité de représentation des réseaux bayésiens ont permis à de nombreux domaines de tirer profits de la théorie des probabilités. Cependant, le succès grandissant, des demandes de plus en plus fortes sont apparues et depuis la dernière décennie des propositions d’extension des BN apparaissent pour répondre aux lacunes de ces derniers.

Dans cette partie nous allons étudier trois propositions d’extension des réseaux bayésiens à l’aide d’un exemple didactique : une représentation de relations entre enseignants, enseignements et élèves dans une université. Cet exemple extrait de [8] permet de mettre en avant la réutilisation de fragments de réseaux, la définition de relations et l’agrégation d’informations. La structure du problème est simple : chaque professeur dispense un nombre quelconque de cours ; chaque étudiant est inscrit à un nombre donné de cours et possède une note pour chacun d’entre eux. Nous souhaitons également agréger les avis des élèves sur chaque professeur et calculer leur moyenne.

Le choix de ces trois modèles n’est pas anodin, car bien que la littérature regorge de propositions d’extension des BN, modèles regroupés plus communément sous le nom de modèle probabiliste du premier ordre (MPPO), les OOBN [9], les PRM [8] et les MEBN [5] sont les MPPO les plus proches des BN en terme de représentation graphique et de sémantique.

Ces trois représentations permettent toutes de générer très facilement des BN de très grande taille par la réutilisation de fragment de réseaux (1000 professeurs, 10000 étudiants, etc.), taille si importante que ces BN sont impossibles à construire, à traiter ou même simplement à représenter en mémoire.

Ces approches se démarquent par leur volonté de fournir un cadre théorique à la répétition de structure dans un réseau bayésien, répétition réalisée par les instanciations des différentes classes dans le modèle. De même, il existe une volonté d’unifier l’expressivité de la logique du premier ordre avec l’utilisation des probabilités pour représenter le raisonnement dans l’incertain, deux domaines de l’aide à la décision qui jusqu’à présent cohabitaient difficilement.

Dans la suite de cet article, nous ne différencierons pas les termes *variables*, *nœuds* et *attributs*. De même pour les notions de *classes* et de *fragments de réseau bayésien*. Lorsque nous parlerons d’*instances* d’une classe, l’analogie devra être faite avec la programmation orientée objet. Dernier

1. En supposant, par exemple, un modèle contenant 100 variables binaires où aucune variable n’a plus de 4 parents, la mémoire nécessaire pour représenter par un BN la loi jointe des 100 variables est plus petite que  $100.2^{4+1} = 3200$  ; à comparer avec la mémoire nécessaire pour représenter par extension une fonction de 100 variables binaires :  $2^{100}$ .

point, il est important de noter que le terme modèle peut être utilisé autant pour désigner un modèle probabiliste, comme les réseaux bayésiens, que pour un modèle issu de la modélisation d'un système par un expert.

### 2.1. Oriented Object Bayesian networks

Un réseau bayésien orienté objet, ou OOBN ([4] et [9]), est issu d'une application directe du paradigme objet. L'élément de base est la classe, un fragment de réseau bayésien, dont les nœuds sont décomposés en trois ensembles : une interface d'entrée, de sortie et des nœuds internes.

La figure 2 montre une hiérarchie de classes possible pour notre exemple. Les nœuds en pointillés correspondant aux nœuds d'entrées, appelés également nœuds références. Ils désignent les variables aléatoires définies en dehors de la classe et permettent la création de relations entre les classes.

Les nœuds en gras indiquent que la classe les possédant contient une instanciation d'une autre classe. Ceci permet d'introduire la notion de composition dans les réseaux bayésiens. De ce fait il est possible de décomposer une modélisation d'un système en fragments de réseaux bayésiens, les classes ; de créer des dépendances entre elles, les références ; et de mettre en avant des relations de composition, telle que la classe *Professor Smith* constituée de deux instances de la classe *Course* : *GraphTheory* et *GameTheory*.

Un dernier point concerne la possibilité de créer une classe héritant d'une autre, dans l'exemple la classe *John Doe* hérite de la classe *Student*. Ceci indique que la classe fille hérite de l'ensemble des nœuds défini par la classe mère, ici les nœuds *Intelligence*, *Ranking* et *Grade*, incluant les relations de dépendances entre ces variables et leur table de probabilité conditionnelle.

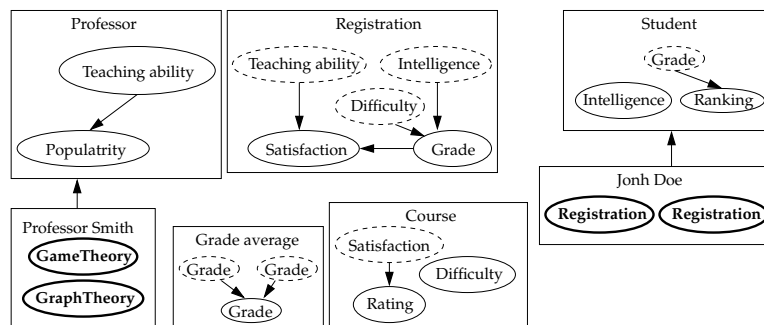
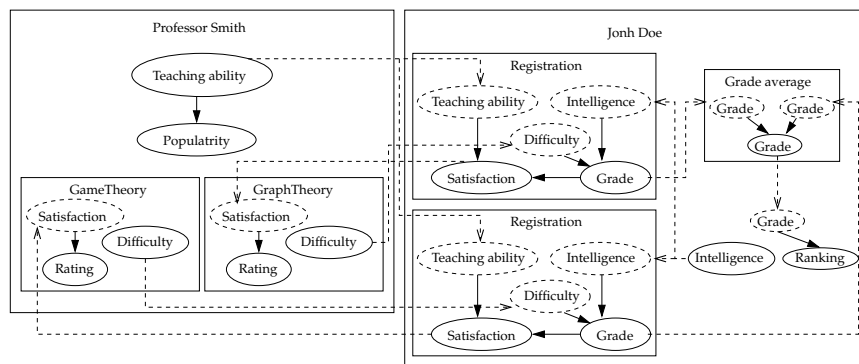


FIGURE 2 – Hiérarchie de classes d'un OOBN.

Une personne familière avec la programmation orientée objet remarquera sûrement les classes *Professor Smith* et *John Doe* qui héritent, respectivement, de *Professor* et de *Student*. Si nous étions dans le cadre de la programmation avec un langage orienté objet, ces deux classes n'hériteraient pas de *Professor* ou de *Student* mais en seraient des instances. Cette bizarrerie est due à l'impossibilité dans les OOBN de définir des relations multiples, i.e. de tableaux. Cette lacune force la définition de sous-classes possédant le nombre approprié d'instances de la référence multiple. Dans notre exemple cela ne pose pas de problème, mais il est raisonnable de penser qu'un professeur enseigne généralement plus de deux matières et, surtout, qu'une classe contient généralement bien plus qu'un seul élève.

De plus, la figure 3 nous montre que les liens entre les nœuds *concrets* et leurs références créent un désordre handicapant considérablement la lecture du modèle, un désordre qui croîtra d'autant que le modèle possédera de cas particuliers. Il faut ajouter que l'exemple des enseignements dans une université est un exercice difficile pour les OOBN, qui sont beaucoup plus adaptés à la modélisation de systèmes hiérarchisés, comme une organisation militaire, pour ne citer qu'un seul exemple [7].

Un dernier point important concerne la classe *Grade average* qui a pour rôle de calculer la moyenne obtenue par un élève sur l'ensemble des enseignements qu'il suit. Ce besoin d'agréger, ou de

FIGURE 3 – Instanciation complète de l'exemple *John Doe*.

quantifier des flux d'informations est un besoin répandu et largement utilisé, introduit dès le début de la formalisation des réseaux bayésiens par l'intermédiaire des *noisy-or gates* [6].

Cette classe pose plusieurs problèmes. Le premier est commun à la nécessité de spécialiser les classes pour instancier les relations multiples : dans notre exemple nous serions contraints de définir autant de variantes de la classe *Grade average* que de groupes d'étudiants suivant le même nombre de cours ; cela sans pouvoir bénéficier de mécanisme d'héritage, ou de création de lien automatique entre les nœuds références de la classe *Grade average* et les nœuds *Grade* des différentes instances de la classe *Register*. Le second problème est plus problématique : toute classe agrégat impose la création d'une table de probabilités conditionnelles de taille exponentielle en fonction du nombre de nœuds agrégés. Ceci rend la tâche quasi impossible pour l'expert chargé de modéliser un système de ce type. De plus, la taille des tables ainsi générées dépassera rapidement la capacité de traitement de n'importe quel algorithme d'inférence.

Pour illustrer l'aspect combinatoire de ce problème, supposons que nous souhaitons appliquer notre exemple à un élève suivant neuf cours d'une université française. Pour chaque cours, cet élève reçoit une note sur 20. Notre table, si nous nous contentons de notes allant des nombres entiers de 0 à 20 contiendrait  $20^{10}$  éléments.

## 2.2. Multi-Entity Bayesian Networks

La figure 4 illustre notre exemple dans le cadre des MEBN. Ce formalisme inspiré de la logique du premier ordre est un des plus expressifs dans la famille des MPPO.

L'unité de base dans un MEBN est appelée *M-Fragment*, ces fragments définissent les différents morceaux de réseau bayésien qui, regroupés ensemble, constituent une *M-Theory*. Comme dans les OOBN, un *M-Frag* est constitué d'un ensemble de nœuds groupés en trois catégories : les nœuds d'entrées, les nœuds résidants et les nœuds contextuels. Si les deux premiers types de nœuds sont semblables à ceux présents dans les OOBN, les nœuds contextuels sont propres aux MEBN.

Les MEBN étendent les BN par une intégration de la logique du premier ordre dont les nœuds contextuels sont le medium. Contrairement à la sémantique habituelle d'un nœud dans un réseau bayésien, ceux ci ne représentent pas des variables aléatoires mais des prédicats dont l'utilité est de spécifier le contexte dans lequel le *M-Frag* doit être placé. Bien que dans notre exemple nous n'utilisons que le prédicat *IsA*, il est possible d'utiliser n'importe quel prédicat formulable dans la logique du premier ordre. Le lecteur attentif notera l'absence d'équivalent de la classe *Grade average* de l'exemple précédent.

Cela est rendu possible grâce à la solution alternative offerte par les MEBN pour définir les tables de probabilités conditionnelles, utilisée ici pour le nœud *Grade(c,s)* du *M-Frag Student*, et qui emploie un pseudo-code illustré dans la table 1.

Cette méthode offre une grande liberté pour définir les tables de probabilités conditionnelles, mais possède plusieurs inconvénients. Le premier vient du constat qu'aucune formalisation de ce pseudo-code n'est donnée et ce malgré de nombreuses publications. Ceci rend difficilement

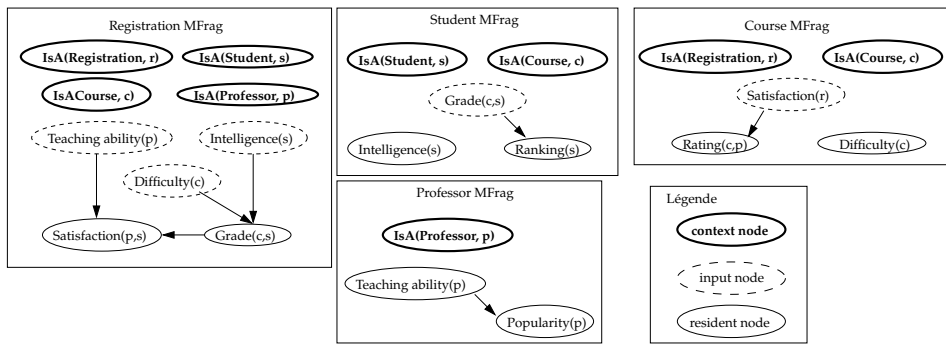


FIGURE 4 – Une *M-Theory* constituée de *M-Fragments*.

exploitable ce type de définitions par les algorithmes d'inférences. La deuxième vient du fait qu'il est nécessaire de régénérer ces tables à chaque modification de la *M-Theory*. Dans notre exemple cela se produirait si nous ajoutions un nouvel élève par exemple. Il serait également nécessaire de régénérer ces tables si un des parents du nœud est observé, par exemple lorsqu'un élève obtient une note pour un de ces enseignements.

Croire que cette génération serait peu coûteuse serait une erreur. En effet il est, a priori, possible de réaliser des jointures avec la base de données définie par les nœuds contextuels et les instances, or dans le cas d'un modèle de grande taille, le coût d'une telle opération pourrait s'avérer élevé. De même une utilisation excessive de boucles ralentirait considérablement la génération dans une base de connaissance de grande taille.

Dernier point en défaveur de cette approche : la définition de ces tables à l'aide d'un pseudo-code est un exercice délicat et vraisemblablement hors de portée de nombreux experts, comme le montre la Table 1, le pseudo-code est assez éloigné d'un tableur, outil particulièrement prisé par les experts pour ce genre d'exercice.

### 2.3. Probabilistic Relational Models

En opposition avec les MEBN, les PRM sont issus d'une approche se basant directement sur les réseaux bayésiens sans inclure la logique du premier ordre. La thèse de Pfeffer [8] est un raffinement des OOBN qui tente de résoudre les difficultés de représentations de références multiples et inclut de nouvelles notions que nous n'aborderons pas dans cet article, telle que l'incertitude structurelle et existentielle.

La brique élémentaire d'un PRM est la classe, un fragment de réseau bayésien, mais contrairement aux OOBN la notion d'interface est abandonnée.

Ce changement de philosophie est permis par l'introduction de la notion de référence, à travers le terme de *slot*, représenté par les traits en pointillés dans la figure 5. Ces références sont multiples

```

distribution[0, 1, 2, ..., 19, 20] = fonction {
  // Nous supposons que la table est initialisé avec des 0
  for each instantiation i in {Grade(c,s) in parents(Ranking(s))} {
    for each grade in i { mean+ = grade ; }
    mean /= count(Grade(c,s) in parents(Ranking(s)));
    if (mean == 0) { P(0|i) = 1 ; }
    else if (mean == 1) { P(1|i) = 1 ; }
    // ...
    else if (mean == 20) { P(20|i) = 1 ; }
  }
}

```

TABLE 1 – Un exemple de pseudo-code définissant la table de probabilité du nœud  $Grade(c,s)$ .

par défaut, telles que les relations entre les classes *Professor* et *Course*, et *Student* et *Register* dans notre exemple.

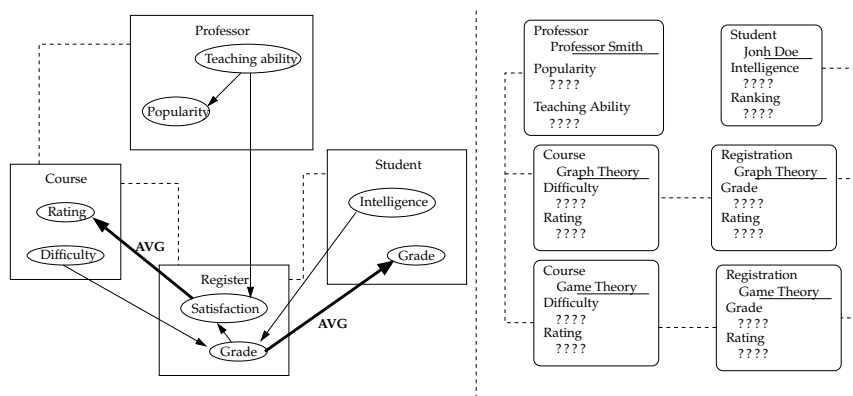


FIGURE 5 – Définitions de classes d'un PRM

La définition des données probabilistes est réalisée grâce à l'utilisation de tables de probabilités conditionnelles. En comparaison au pseudo-code disponible dans les MEBN, nous pourrions croire que cela est insuffisant. Toutefois les PRM intègre une notion permettant de quantifier le flux d'informations passant par les références multiples : les agrégateurs. La relation entre *Student.Grade* et *Register.Grade* en est un exemple d'utilisation. Cet opérateur, ou encore quantificateur, a pour rôle de résumer l'information contenue dans les multiples instances au sein d'une relation multiple entre deux classes. Dans notre exemple, les notes obtenues par un élève sont l'information résumée.

Bien que l'idée soit séduisante, il n'existait aucune méthode d'implémentation de tels quantificateurs au moment des premières publications sur les PRM, alors qu'une approche naïve nécessiterait la création dynamique des tables en fonctions du type d'agrégateur utilisé. Cette démarche a l'inconvénient de générer des tables de probabilités conditionnelles excessivement grandes, à l'instar des OOBN. Toutefois l'axiomatisation des *noisy-or* réalisée par Cozman [1] permet de définir des fonctions capables d'agréger l'information et ce indépendamment du nombre d'entrées dans la fonction, c'est-à-dire du nombre d'instances dans la relation.

Cette axiomatisation fut proposée après les premières publications sur les PRM et a, dans un sens, contribué à l'essor de ce formalisme dans la communauté car elles fournissent un cadre théorique efficace et simple d'utilisation<sup>2</sup>.

Pour conclure, nous noterons qu'il est possible de créer des hiérarchies de classes au sein d'un modèle. La relation de hiérarchie ne repose pas sur les notions d'interfaces comme pour les OOBN, mais procède de la même manière : une classe fille hérite de l'ensemble des nœuds, de références et des tables de probabilités conditionnelles.

### 3. Similarités et différences entre OOBN, PRM et MEBN

Les trois propositions que nous avons abordées permettent toutes de définir le réseau bayésien sous-jacent d'un modèle. Cette caractéristique incite à penser que ces formalismes sont plus ou moins équivalents. En effet, les fragments de réseaux bayésiens, sont à la base de chacune de ces propositions. Toutefois les différences sont perceptibles essentiellement sur la représentation et la factorisation des tables de probabilités de grandes tailles. Nous avons pu constater que les OOBN ne fournissent aucun mécanisme d'aide, les MEBN évitent le problème en utilisant un langage de

2. Les fonctions définies par cette axiomatisation rendent totalement automatique le traitement des références multiples et permettent une amélioration non négligeable des performances lors de l'inférence.

programmation et les PRM se reposaient sur une notion d'agrégateurs qui, il y a encore peu, était très floue.

Une autre caractéristique commune entre MEBN et PRM est l'intégration de la logique du premier ordre, nous avons pu constater qu'elle était directement intégrée au sein du formalisme des MEBN, mais elle est également présente sous une forme un peu plus restreinte dans les PRM. La principale utilité des nœuds contextuels est de fournir un squelette d'instanciation : en précisant les différentes conditions nécessaires à l'utilisation d'un *M-Frag* elles induisent des contraintes fortes sur la topologie des interconnexions entre instances. Or dans les PRM, la notion de *slot chains*, ou chaîne de références, joue exactement le même rôle. Il suffit alors d'ajouter des quantificateurs universels, les opérateurs *or* et *and* dans l'ensemble des agrégateurs autorisés pour obtenir une capacité d'expression équivalente (ou presque) des MEBN, avec cependant un résultat beaucoup moins compact qu'un nœud contextuel.

#### 4. Conclusions

Nous avons montré qu'il existe plusieurs propositions pour modéliser des réseaux bayésiens de grandes tailles. Bien qu'encore perfectible, ces approches permettent un gain considérable tant dans la factorisation des modèles que dans la maintenance.

Toutefois la transcription de la connaissance des experts dans une base de connaissance orientée objet ou logique est souvent un exercice difficile : la complexité inhérente aux modèles du premier ordre rend caduque la facilité d'utilisation qui a permis aux réseaux bayésiens de connaître le succès qui est le leur. Cette difficulté d'utilisation par les experts de tels modèles rend critique l'intégration des notions de relations entre classes, de la quantification des dépendances conditionnelles et de la gestion des tables de probabilités de grandes tailles. Des langages de spécification et des environnements de manipulations graphiques sont encore à inventer.

De plus, la taille des réseaux générés pose des problèmes d'inférence qui doivent encore être résolus. Ceci implique un développement conjoint des formalismes avec les méthodes d'inférences, travail commencé mais encore inachevé.

---

*Nos remerciements s'adressent à la DGA pour le financement de ma thèse et ainsi qu'à l'ensemble des partenaires du projet ANR SKOOB : Bayesia, le CHU de Nancy, le CRAN, EDF R&D, l'ERPI, l'INERIS et la SOREDAB.*

---

#### Bibliographie

1. Fabio G. Cozman. Axiomatizing noisy-or. Technical report, In Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-04, 2004).
2. Dan Geiger, Thomas Verma, and Judea Pearl. Identifying independence in Bayesian networks. *networks*, 20(5) :507–534, aug 1990. Special Issue on Influence Diagrams.
3. Joseph Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46 :311–350, 1990.
4. D. Koller and A. Pfeffer. Object-oriented Bayesian networks. In *Proceedings of the 13th Annual Conference on Uncertainty in AI (UAI)*, pages 302–313, 1997. Winner of the Best Student Paper Award.
5. Kathryn Blackmond Laskey. MEBN : A language for first-order bayesian knowledge bases. *Artificial Intelligence*, 172 :140–178, 2008.
6. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, September 1988.
7. Avi Pfeffer, Daphne Koller, Brian Milch, and Ken T. Takusagawa. SPOOK : A system for probabilistic object-oriented knowledge representation. In *In Uncertainty in Artificial Intelligence : Proceedings of the Fifteenth Conference*, pages 541–550. Morgan Kaufmann, 1999.
8. Avrom J. Pfeffer. *Probabilistic Reasoning for Complex Systems*. PhD thesis, Stanford University, January 2000.
9. Pierre-Henri Wuillemin and Olav Bangso. Réseaux probabilistes orientés objet. In *Langages et modèles à objet*, pages 123–138, Mont Saint-Hilaire, Québec, 1 2000. Hermès.