

Université de Rouen  
UFR des Sciences

HABILITATION A DIRIGER LES RECHERCHES  
Spécialité Informatique, Automatique et Traitement du Signal

RÉSEAUX BAYÉSIENS : APPRENTISSAGE ET  
MODÉLISATION DE SYSTÈMES COMPLEXES

Philippe LERAY

Maître de Conférences  
LITIS – EA 4051  
Département ASI, INSA de Rouen

Soutenu le \_ novembre 2006 devant le jury composé de

Patrice AKNIN, INRETS, examinateur  
Salem BENFERHAT, Université d'Artois, rapporteur  
Stéphane CANU, INSA de Rouen, examinateur  
Bernard DUBUISSON, Université Technologique de Compiègne, rapporteur  
Isabelle GUYON, ClopiNet, rapporteur  
Laurent HEUTTE, Université de Rouen, examinateur



# Table des matières

<b>I Réseaux bayésiens : apprentissage et modélisation de systèmes complexes</b>	<b>7</b>
<b>1 Introduction aux réseaux bayésiens</b>	<b>9</b>
<b>2 Apprentissage des paramètres</b>	<b>13</b>
2.1 A partir de données complètes . . . . .	14
2.1.1 Apprentissage statistique . . . . .	14
2.1.2 Apprentissage bayésien . . . . .	15
2.2 A partir de données incomplètes . . . . .	16
2.2.1 Nature des données manquantes . . . . .	16
2.2.2 Traitement des données MCAR . . . . .	17
2.2.3 Traitement des données MAR . . . . .	17
2.2.4 Apprentissage statistique et algorithme EM . . . . .	17
2.2.5 Apprentissage bayésien et algorithme EM . . . . .	19
2.3 Incorporation de connaissances . . . . .	20
2.3.1 Comment demander à un expert d'estimer une probabilité?	21
2.3.2 Quelles probabilités estimer? . . . . .	21
2.3.3 Comment fusionner les avis de plusieurs experts? . . . . .	24
<b>3 Apprentissage de la structure</b>	<b>25</b>
3.1 Introduction . . . . .	26
3.2 Hypothèses . . . . .	27
3.2.1 Fidélité . . . . .	27
3.2.2 Suffisance causale . . . . .	27
3.2.3 Notion d'équivalence de Markov . . . . .	27
3.3 Recherche d'indépendances conditionnelles . . . . .	30
3.3.1 Tests d'indépendance conditionnelle . . . . .	31
3.3.2 Algorithmes PC et IC . . . . .	32
3.3.3 Quelques améliorations . . . . .	36
3.4 Algorithmes basés sur un score . . . . .	37
3.4.1 Les scores possibles . . . . .	37
3.4.2 Déterminer un a priori sur les structures . . . . .	40
3.4.3 Pourquoi chercher la meilleure structure? . . . . .	40
3.4.4 Recherche dans l'espace des réseaux bayésiens . . . . .	41
3.4.5 Algorithmes basés sur un score et données incomplètes . .	50
3.5 Recherche dans l'espace des classes d'équivalence de Markov . . .	53
3.6 Méthodes hybrides . . . . .	61
3.7 Incorporation de connaissances . . . . .	62

3.7.1	Structures de réseaux bayésiens pour la classification . . .	62
3.7.2	Structures de réseaux bayésiens avec variables latentes . .	65
3.7.3	Autres structures particulières . . . . .	66
3.8	Découverte de variables latentes . . . . .	66
3.8.1	Recherche d'indépendances conditionnelles . . . . .	66
3.8.2	Algorithmes basés sur un score . . . . .	68
3.9	Cas particulier des réseaux bayésiens causaux . . . . .	68
3.9.1	Définition . . . . .	69
3.9.2	Apprentissage sans variables latentes . . . . .	69
3.9.3	Apprentissage avec variables latentes . . . . .	70
<b>4</b>	<b>Conclusion et Perspectives</b>	<b>72</b>
<b>5</b>	<b>Références</b>	<b>74</b>
<b>II</b>	<b>Notice des titres et travaux</b>	<b>87</b>
<b>6</b>	<b>Curriculum Vitae</b>	<b>89</b>
6.1	Etat civil . . . . .	90
6.2	Formation . . . . .	90
6.3	Parcours . . . . .	90
6.4	Thèmes de Recherche . . . . .	91
6.5	Publications . . . . .	91
6.6	Encadrement . . . . .	91
6.7	Rayonnement Scientifique . . . . .	92
6.8	Responsabilités administratives . . . . .	92
<b>7</b>	<b>Travaux scientifiques et réalisations technologiques</b>	<b>93</b>
7.1	Depuis ma nomination . . . . .	95
7.2	Vulgarisation scientifique . . . . .	98
7.2.1	Rédaction d'un livre . . . . .	98
7.2.2	Réalisation et diffusion d'outils informatiques . . . . .	98
7.3	Perpectives personnelles . . . . .	98
<b>8</b>	<b>Encadrement scientifique</b>	<b>101</b>
8.1	Thèses soutenues . . . . .	102
8.2	Thèses en cours . . . . .	103
8.3	Postdoc . . . . .	104
8.4	DEA, M2 Recherche . . . . .	104
<b>9</b>	<b>Rayonnement scientifique et industriel</b>	<b>105</b>
9.1	Rayonnement scientifique . . . . .	106
9.1.1	Jury de thèse . . . . .	106
9.1.2	Animation scientifique . . . . .	106
9.1.3	Séminaires invités et formations spécifiques . . . . .	107
9.1.4	Commission de spécialistes . . . . .	108
9.2	Relations industrielles . . . . .	108
9.2.1	Schneider Electric . . . . .	108
9.2.2	RATP . . . . .	108
9.2.3	Thales Air Defence . . . . .	108

9.2.4	EADS . . . . .	109
<b>10</b>	<b>Responsabilités administratives</b>	<b>110</b>
10.1	Direction du département ASI de l'INSA de Rouen . . . . .	112
10.1.1	Le département . . . . .	112
10.1.2	Activités administratives INSA . . . . .	113
10.1.3	Activités administratives internes au département . . . . .	113
10.1.4	Bilan d'activité 2003-2006 . . . . .	114
10.2	Responsabilités antérieures dans le département . . . . .	115
10.2.1	Gestion de la filière Traitement de l'Information . . . . .	115
10.2.2	Responsable du site internet asi.insa-rouen.fr . . . . .	115
10.3	Responsabilités au sein du laboratoire PSI . . . . .	115
10.3.1	Chargé des relations avec les doctorants PSI . . . . .	115
10.3.2	Membre du comité éditorial du site internet PSI . . . . .	115
<b>11</b>	<b>Enseignements</b>	<b>116</b>
11.1	Informatique . . . . .	117
11.2	Traitement de l'Information . . . . .	118
11.3	Récapitulatif . . . . .	121
<b>12</b>	<b>Liste complète des publications</b>	<b>123</b>
12.1	Livres . . . . .	124
12.2	Chapitres de livres . . . . .	124
12.3	Revue internationale à comité de lecture . . . . .	124
12.4	Revue nationale à comité de lecture . . . . .	124
12.5	Colloques internationaux avec actes . . . . .	125
12.6	Colloques nationaux avec actes . . . . .	126
12.7	Ateliers, communications sans actes et rapports techniques . . . . .	127
<b>A</b>	<b>Sélection de publications</b>	<b>129</b>
A.1	P. Leray and O. Francois. Réseaux bayésiens pour la classification – méthodologie et illustration dans le cadre du diagnostic médical. <i>Revue d'Intelligence Artificielle</i> , 18/2004 :169–193, 2004 . . . . .	130
A.2	I. Zaarour, L. Heutte, P. Leray, J. Labiche, B. Eter, and D. Mellier. Clustering and bayesian network approaches for discovering handwriting strategies of primary school children. <i>International Journal of Pattern Recognition and Artificial Intelligence</i> , 18(7) :1233–1251, 2004 . . . . .	155
A.3	S. Meganck, P. Leray, and B. Manderick. Learning causal bayesian networks from observations and experiments : A decision theoretic approach. In <i>Proceedings of the Third International Conference, MDAI 2006</i> , volume 3885 of <i>Lecture Notes in Artificial Intelligence</i> , pages 58–69, Tarragona, Spain, 2006. Springer . . . . .	187
A.4	S. Maes, S. Meganck, and P. Leray. <i>Causality and Probability in the Sciences</i> , chapter An integral approach to causal inference with latent variables, 23 pages, Texts In Philosophy series. London College Publications, 2007 . . . . .	199



Première partie

Réseaux bayésiens :  
apprentissage et modélisation  
de systèmes complexes



# Chapitre 1

## Introduction aux réseaux bayésiens

La représentation des connaissances et le raisonnement à partir de ces représentations a donné naissance à de nombreux modèles. Les modèles graphiques probabilistes, et plus précisément les réseaux bayésiens, initiés par Judea Pearl dans les années 1980, se sont révélés des outils très pratiques pour la représentation de connaissances incertaines, et le raisonnement à partir d'informations incomplètes.

Un réseau bayésien  $\mathcal{B} = (\mathcal{G}, \theta)$  est défini par

- $\mathcal{G} = (X, E)$ , graphe dirigé sans circuit dont les sommets sont associés à un ensemble de variables aléatoires  $X = \{X_1, \dots, X_n\}$ ,
- $\theta = \{P(X_i | Pa(X_i))\}$ , ensemble des probabilités de chaque nœud  $X_i$  conditionnellement à l'état de ses parents  $Pa(X_i)$  dans  $\mathcal{G}$ .

Ainsi, la partie graphique du réseau bayésien indique les dépendances (ou indépendances) entre les variables et donne un outil visuel de représentation des connaissances, outil plus facilement appréhendable par ses utilisateurs. De plus, l'utilisation de probabilités permet de prendre en compte l'incertain, en quantifiant les dépendances entre les variables. Ces deux propriétés ont ainsi été à l'origine des premières dénominations des réseaux bayésiens, "systèmes experts probabilistes", où le graphe était comparé à l'ensemble de règles d'un système expert classique, et les probabilités conditionnelles présentées comme une quantification de l'incertitude sur ces règles.

Pearl *et al.* ont aussi montré que les réseaux bayésiens permettaient de représenter de manière compacte la distribution de probabilité jointe sur l'ensemble des variables :

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)) \quad (1.1)$$

Cette décomposition d'une fonction globale en un produit de termes locaux dépendant uniquement du nœud considéré et de ses parents dans le graphe, est une propriété fondamentale des réseaux bayésiens. Elle est à la base des premiers travaux portant sur le développement d'algorithmes d'inférence, qui calculent

la probabilité de n'importe quelle variable du modèle à partir de l'observation même partielle des autres variables. Ce problème a été prouvé NP-complet, mais a abouti à différents algorithmes qui peuvent être assimilés à des méthodes de propagation d'information dans un graphe. Ces méthodes utilisent évidemment la notion de probabilité conditionnelle, i.e. quelle est la probabilité de  $X_i$  sachant que j'ai observé  $X_j$ , mais aussi le théorème de Bayes, qui permet de calculer, inversement, la probabilité de  $X_j$  sachant  $X_i$ , lorsque  $P(X_i | X_j)$  est connu.

Nos travaux ne concernent pas ces algorithmes d'inférence. Par contre, le chapitre 5 de [145] constitue une très bonne introduction à ces méthodes.

Il faut noter que l'appellation "réseaux bayésiens" prête à confusion. En effet, ceux-ci ne sont pas forcément des modèles bayésiens, au sens statistique du terme.<sup>1</sup> Ce sont des modèles graphiques probabilistes utilisant le théorème de Bayes pour "raisonner".

La modélisation d'un problème par un réseau bayésien, puis l'utilisation d'algorithmes d'inférence, ont fait des réseaux bayésiens des outils idéaux pour le raisonnement ou le diagnostic à partir d'informations incomplètes. Quelle est, par exemple, la probabilité qu'un patient soit atteint de telle ou telle maladie, sachant que certains symptômes ont été observés, mais que d'autres informations ne sont pas connues ? Quelle est la configuration des variables représentant l'état de chacun des composants d'un système, sachant que tel ou tel comportement a été remarqué ?

Ainsi Microsoft a proposé dès 1994 un assistant de dépannage pour les problèmes d'impression dans Windows 95. Leur programme commence par proposer la solution qui paraît la plus probable pour résoudre le problème détecté. L'utilisateur a alors trois solutions :

- indiquer que la solution a effectivement résolu le problème, ce qui met fin à la session d'assistance,
- indiquer qu'il est dans l'incapacité de tester la solution proposée. Le système doit donc proposer une autre solution sans avoir obtenu d'information supplémentaire,
- ou indiquer que cette solution n'est pas la bonne, après l'avoir testée, ce qui donne un renseignement additionnel au système qui pourra donc en tenir compte pour inférer une nouvelle proposition.

Les réseaux bayésiens modélisant efficacement la loi de probabilité jointe de l'ensemble des variables, sont un formalisme privilégié pour l'utilisation de méthodes d'échantillonnage stochastique. Celles-ci permettent de générer à volonté des données simulées. Les réseaux bayésiens sont alors des outils de simulation qui permettent à l'expert d'observer le comportement de son système dans des contextes qu'il n'est pas forcément capable de tester lui-même.

La figure 1.1 est un exemple de réseau bayésien modélisant un cycle de maintenance sur un système. Les modules au centre du modèle représentent quatre experts indépendants estimant l'état du système, en fonction d'un contexte qui leur est propre. Les propositions de maintenance de ces experts sont ensuite fusionnées dans le module suivant pour arriver à une décision finale de maintenance. La partie supérieure du modèle permet de représenter l'évolution

---

<sup>1</sup>i.e. définissant une loi a priori sur ses paramètres, puis faisant évoluer cette loi à l'aide de données.

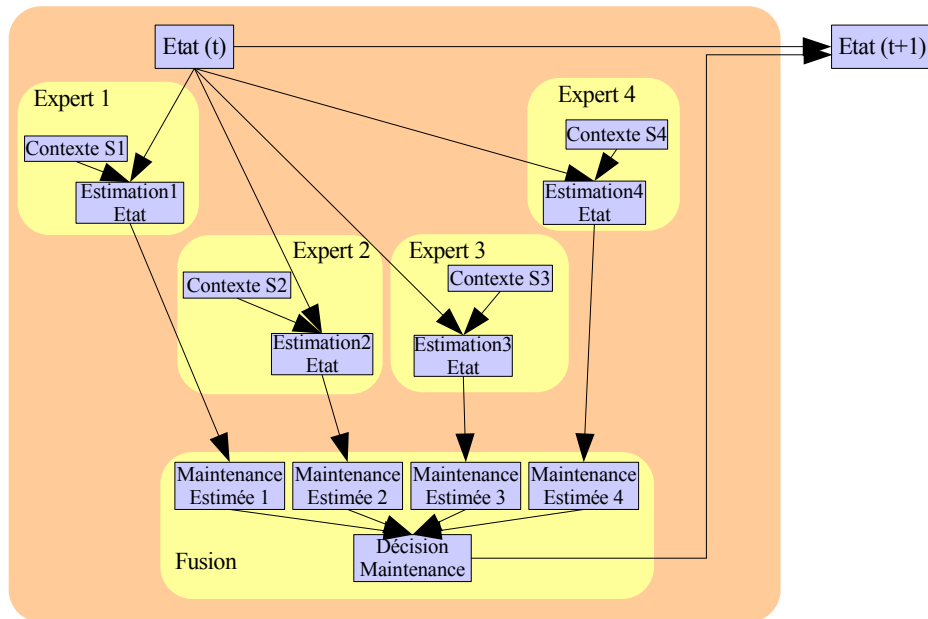


FIG. 1.1 – Un exemple de réseau bayésien : application à la modélisation de la maintenance d'un système à partir de plusieurs expertises.

temporelle de l'état du système : cet état dépend du fait qu'une décision de maintenance a été prise, mais aussi de son état précédent, permettant ainsi de représenter des modèles de dégradation. A partir d'un tel modèle, l'industriel peut vérifier la qualité de chacun des experts, et jouer sur les valeurs des contextes pour tenter d'améliorer la qualité de son système.

En parallèle avec les algorithmes d'inférences, d'autres approches ont été proposées pour l'apprentissage à partir de données, ou d'expertises, des probabilités conditionnelles quantifiant les dépendances entre les variables d'un réseau bayésien de structure connue. Le chapitre 2 passe donc en revue les différentes méthodes existantes.

Celles-ci ont l'avantage de faire intervenir l'expert à plusieurs niveaux. Il peut ainsi déterminer complètement ces paramètres, à partir de techniques d'élicitation de probabilité ou d'autres techniques développées en Ingénierie des Connaissances. A l'opposé, il peut laisser les données guider complètement cette estimation, grâce aux techniques d'estimation statistique. A mi-chemin entre ces deux modes de fonctionnement, les techniques d'estimation bayésienne permettent l'utilisation conjointe de données et de connaissances a priori qui peuvent être fixées par l'expert.

La modularité des réseaux bayésiens permet aussi à l'expert de choisir n'importe quelle source d'estimation disponible (lui-même ou des données) pour estimer les paramètres d'une partie du système, et une autre pour telle autre partie. Cette flexibilité autorise alors la construction progressive de modèles de plus en plus complexes, à l'aide de sources d'informations hétérogènes.

Reprenons par exemple le réseau bayésien de la figure 1.1. Les probabilités

conditionnelles des modules représentant les experts peuvent être estimées indépendamment, à partir d'expériences ou d'expertises différentes. De même, les paramètres du module de fusion (décision de maintenance), ou ceux correspondant à la modélisation dynamique de l'état du système, pourront être estimés à partir de données ou fixés par des connaissances expertes.

Dans certains problèmes, l'expert est souvent amené à construire lui-même le graphe du réseau bayésien, en réfléchissant en terme de causalité. A l'opposé, l'apprentissage du graphe à partir de données se fait dans un cadre plus général que celui des réseaux bayésiens causaux, cadre dans lequel plusieurs graphes seront équivalents, mais où un seul capturera éventuellement les relations de causalité du problème. Cette problématique d'apprentissage de la structure fera l'objet de notre chapitre 3.

Depuis 1990, les travaux de recherche dans ce domaine se sont essentiellement intéressés à l'apprentissage des réseaux bayésiens lorsque toutes les variables sont connues (pas de variables latentes), et lorsque ces variables sont complètement observées (pas de données manquantes), problème lui aussi NP-complet.

La prise en compte de données incomplètes, comme la découverte de variables latentes, posent encore de sérieux défis en terme de complexité.

La découverte de réseaux bayésiens complètement causaux à partir de données est une question qui a été abordée plus récemment. Les travaux sur le sujet s'accordent sur le fait qu'il est impossible de travailler à partir de données d'observations uniquement. Les plans d'expériences, c'est à dire la façon dont les données ont été obtenues, sont des informations essentielles pour capturer la notion de causalité.

Les méthodes proposées dans le cadre de l'apprentissage de structure ont transformé les réseaux bayésiens non seulement en outil de représentation des connaissances, mais aussi en outil de découverte de celles-ci. Lorsque ces connaissances font appel à la notion de causalité, les résultats de telles méthodes sont à examiner avec précaution. Tout d'abord, la causalité est une notion complexe que les spécialistes peinent à définir unanimement. Ensuite, les hypothèses sous-jacentes à ces algorithmes ne doivent pas être oubliées, sous peine d'aboutir à des interprétations causales erronées, voire dangereuses. Par exemple, peut-on toujours affirmer que toutes les variables nécessaires à notre modélisation sont connues ?

L'essentiel de nos travaux de recherche concerne l'apprentissage de structure d'un réseau bayésien, en levant progressivement les différentes hypothèses posées :

- pas de variables latentes, données complètes
- pas de variables latentes, données incomplètes
- comment découvrir des variables latentes ?
- comment arriver à une structure réellement causale ?

Nous aborderons aussi le cas particulier de la classification, pour lequel des structures spécifiques de réseaux bayésiens ont été étudiées.

## Chapitre 2

# Apprentissage des paramètres

---

<b>2.1</b>	<b>A partir de données complètes</b>	<b>14</b>
2.1.1	Apprentissage statistique	14
2.1.2	Apprentissage bayésien	15
<b>2.2</b>	<b>A partir de données incomplètes</b>	<b>16</b>
2.2.1	Nature des données manquantes	16
2.2.2	Traitement des données MCAR	17
2.2.3	Traitement des données MAR	17
2.2.4	Apprentissage statistique et algorithme EM	17
2.2.5	Apprentissage bayésien et algorithme EM	19
<b>2.3</b>	<b>Incorporation de connaissances</b>	<b>20</b>
2.3.1	Comment demander à un expert d'estimer une probabilité?	21
2.3.2	Quelles probabilités estimer?	21
2.3.3	Comment fusionner les avis de plusieurs experts?	24

---

## 2.1 A partir de données complètes

Nous cherchons ici à estimer les distributions de probabilités (ou les paramètres des lois correspondantes) à partir de données disponibles.

L'estimation de distributions de probabilités, paramétriques ou non, est un sujet très vaste et complexe. Nous décrivons ici les méthodes les plus utilisées dans le cadre des réseaux bayésiens, selon que les données à notre disposition sont complètes ou non, en conseillant la lecture de ([70], [83] et [75]) pour plus d'informations.

### 2.1.1 Apprentissage statistique

Dans le cas où toutes les variables sont observées, la méthode la plus simple et la plus utilisée est l'estimation statistique qui consiste à estimer la probabilité d'un événement par la fréquence d'apparition de l'événement dans la base de données. Cette approche, appelée maximum de vraisemblance (MV), nous donne alors :

$$\hat{P}(X_i = x_k \mid pa(X_i) = x_j) = \hat{\theta}_{i,j,k}^{MV} = \frac{N_{i,j,k}}{\sum_k N_{i,j,k}} \quad (2.1)$$

où  $N_{i,j,k}$  est le nombre d'événements dans la base de données pour lesquels la variable  $X_i$  est dans l'état  $x_k$  et ses parents sont dans la configuration  $x_j$ .

Soit  $\mathbf{x}^{(l)} = \{x_{k_1}^{(l)} \dots x_{k_n}^{(l)}\}$  un exemple de notre base de données. La vraisemblance de cet exemple conditionnellement aux paramètres  $\theta$  du réseau est :

$$\begin{aligned} P(\mathcal{X} = \mathbf{x}^{(l)} \mid \theta) &= P(X_1 = x_{k_1}^{(l)}, \dots, X_n = x_{k_n}^{(l)} \mid \theta) \\ &= \prod_{i=1}^n P(X_i = x_{k_i}^{(l)} \mid pa(X_i) = x_j^{(l)}, \theta) \\ &= \prod_{i=1}^n \theta_{i,j^{(l)},k^{(l)}} \end{aligned}$$

La vraisemblance de l'ensemble des données  $\mathcal{D}$  est :

$$L(\mathcal{D} \mid \theta) = \prod_{l=1}^N P(\mathcal{X} = \mathbf{x}^{(l)} \mid \theta) = \prod_{i=1}^n \prod_{l=1}^N \theta_{i,j^{(l)},k^{(l)}}$$

L'examen détaillé du produit  $\prod_l \theta_{i,j^{(l)},k^{(l)}}$  nous montre que le terme  $\theta_{i,j,k}$  (pour  $i, j, k$  fixés) apparaît autant de fois que l'on trouve la configuration  $X_i = x_k$  et  $pa(X_i) = x_j$  dans les données, soit  $N_{i,j,k}$ . La vraisemblance des données peut donc se réécrire :

$$L(\mathcal{D} \mid \theta) = \prod_{i=1}^n \prod_{l=1}^N \theta_{i,j^{(l)},k^{(l)}} = \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{i,j,k}^{N_{i,j,k}} \quad (2.2)$$

La log-vraisemblance s'écrit alors :

$$LL(\mathcal{D} \mid \theta) = \log L(\mathcal{D} \mid \theta) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{i,j,k} \log \theta_{i,j,k} \quad (2.3)$$

Nous savons aussi que les  $\theta_{i,j,k}$  sont liés par la formule suivante :

$$\sum_{k=1}^{r_i} \theta_{i,j,k} = 1 \quad \text{soit} \quad \theta_{i,j,r_i} = 1 - \sum_{k=1}^{r_i-1} \theta_{i,j,k}$$

Réécrivons la log-vraisemblance à partir des  $\theta_{i,j,k}$  indépendants :

$$LL(\mathcal{D} | \theta) = \sum_{i=1}^n \sum_{j=1}^{q_i} \left( \sum_k^{r_i-1} N_{i,j,k} \log \theta_{i,j,k} + N_{i,j,r_i} \log \left( 1 - \sum_{k=1}^{r_i-1} \theta_{i,j,k} \right) \right)$$

Et sa dérivée par rapport à un paramètre  $\theta_{i,j,k}$  est :

$$\frac{\partial LL(\mathcal{D} | \theta)}{\partial \theta_{i,j,k}} = \frac{N_{i,j,k}}{\theta_{i,j,k}} - \frac{N_{i,j,r_i}}{\left( 1 - \sum_{k=1}^{r_i-1} \theta_{i,j,k} \right)} = \frac{N_{i,j,k}}{\theta_{i,j,k}} - \frac{N_{i,j,r_i}}{\theta_{i,j,r_i}}$$

La valeur  $\hat{\theta}_{i,j,k}$  du paramètre  $\theta_{i,j,k}$  maximisant la vraisemblance doit annuler cette dérivée et vérifie donc :

$$\frac{N_{i,j,k}}{\hat{\theta}_{i,j,k}} = \frac{N_{i,j,r_i}}{\hat{\theta}_{i,j,r_i}} \quad \forall k \in \{1, \dots, r_i - 1\}$$

soit

$$\frac{N_{i,j,1}}{\hat{\theta}_{i,j,1}} = \frac{N_{i,j,2}}{\hat{\theta}_{i,j,2}} = \dots = \frac{N_{i,j,r_i-1}}{\hat{\theta}_{i,j,r_i-1}} = \frac{N_{i,j,r_i}}{\hat{\theta}_{i,j,r_i}} = \frac{\sum_{k=1}^{r_i} N_{i,j,k}}{\sum_{k=1}^{r_i} \hat{\theta}_{i,j,k}} = \sum_{k=1}^{r_i} N_{i,j,k}$$

d'où

$$\hat{\theta}_{i,j,k} = \frac{N_{i,j,k}}{\sum_{k=1}^{r_i} N_{i,j,k}} \quad \forall k \in \{1, \dots, r_i\}$$

### 2.1.2 Apprentissage bayésien

L'estimation bayésienne suit un principe quelque peu différent. Il consiste à trouver les paramètres  $\theta$  les plus probables *sachant que les données ont été observées*, en utilisant des a priori sur les paramètres. La règle de Bayes nous énonce que :

$$P(\theta | \mathcal{D}) \propto P(\mathcal{D} | \theta)P(\theta) = L(\mathcal{D} | \theta)P(\theta)$$

Lorsque la distribution de l'échantillon suit une loi multinomiale (voir équation 2.2), la distribution a priori conjuguée est la distribution de Dirichlet :

$$P(\theta) \propto \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} (\theta_{i,j,k})^{\alpha_{i,j,k}-1}$$

où  $\alpha_{i,j,k}$  sont les coefficients de la distribution de Dirichlet associée à la loi a priori  $P(X_i = x_k | pa(X_i) = x_j)$ . Un des avantages des distributions exponentielles comme celle de Dirichlet est qu'elle permet d'exprimer facilement la loi a posteriori des paramètres  $P(\theta | \mathcal{D})$  [113] :

$$P(\theta | \mathcal{D}) \propto \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} (\theta_{i,j,k})^{N_{i,j,k} + \alpha_{i,j,k} - 1}$$

En posant  $N'_{i,j,k} = N_{i,j,k} + \alpha_{i,j,k} - 1$ , on retrouve le même genre de formule que dans l'équation 2.2. Un raisonnement identique permet de trouver les valeurs des paramètres  $\theta_{i,j,k}$  qui vont maximiser  $P(\theta | \mathcal{D})$ .

L'approche de maximum a posteriori (MAP) nous donne alors :

$$\hat{P}(X_i = x_k | pa(X_i) = x_j) = \hat{\theta}_{i,j,k}^{MAP} = \frac{N_{i,j,k} + \alpha_{i,j,k} - 1}{\sum_k (N_{i,j,k} + \alpha_{i,j,k} - 1)} \quad (2.4)$$

où  $\alpha_{i,j,k}$  sont les paramètres de la distribution de Dirichlet associée à la loi a priori  $P(X_i = x_k | pa(X_i) = x_j)$ .

Une autre approche bayésienne consiste à calculer l'espérance a posteriori des paramètres  $\theta_{i,j,k}$  au lieu d'en chercher le maximum. Nous abtenons alors par cette approche d'espérance a posteriori (EAP) (voir [113]) :

$$\hat{P}(X_i = x_k | pa(X_i) = x_j) = \hat{\theta}_{i,j,k}^{EAP} = \frac{N_{i,j,k} + \alpha_{i,j,k}}{\sum_k (N_{i,j,k} + \alpha_{i,j,k})} \quad (2.5)$$

Les estimations que nous venons d'évoquer (maximum de vraisemblance, maximum a posteriori et espérance a posteriori) ne sont valables que si les variables sont entièrement observées. Les méthodes suivantes vont donc s'appliquer aux cas où certaines données sont manquantes.

## 2.2 A partir de données incomplètes

Dans les applications pratiques, les bases de données sont très souvent incomplètes. Certaines variables ne sont observées que partiellement ou même jamais, que ce soit en raison d'une panne de capteurs, d'une variable mesurable seulement dans un contexte bien précis, d'une personne sondée ayant oublié de répondre à une question, etc...

Après avoir constaté l'existence de différents types de données incomplètes, nous aborderons les deux cas traitables automatiquement, pour ensuite nous concentrer sur un des algorithmes les plus utilisés, l'algorithme EM.

### 2.2.1 Nature des données manquantes

Notons  $\mathcal{D} = \{X_i^l\}_{1 \leq i \leq n, 1 \leq l \leq N}$  notre ensemble de données, avec  $\mathcal{D}_o$  la partie observée mais incomplète de  $\mathcal{D}$ , et  $\mathcal{D}_m$  la partie manquante. Notons aussi  $\mathcal{M} = \{M_{il}\}$  avec  $M_{il} = 1$  si  $X_i^l$  est manquant, et 0 sinon.

Le traitement des données manquantes dépend de leur nature. [115] distingue plusieurs types de données manquantes :

- MCAR (Missing Completly At Random) :  $P(\mathcal{M} | \mathcal{D}) = P(\mathcal{M})$ , la probabilité qu'une donnée soit manquante ne dépend pas de  $\mathcal{D}$ ,
- MAR (Missing At Random) :  $P(\mathcal{M} | \mathcal{D}) = P(\mathcal{M} | \mathcal{D}_o)$ , la probabilité qu'une donnée soit manquante dépend des données observées,
- NMAR (Not Missing At Random) : la probabilité qu'une donnée soit manquante dépend à la fois des données observées et manquantes.

Les situations MCAR et MAR sont les plus faciles à résoudre car les données observées contiennent toutes les informations nécessaires pour estimer la distribution des données manquantes. La situation NMAR est plus délicate car

il faut alors faire appel à des informations extérieures pour réussir à modéliser la distribution des données manquantes et revenir à une situation MCAR ou MAR.

### 2.2.2 Traitement des données MCAR

Lorsque les données manquantes sont de type MCAR, la première approche possible et la plus simple est l'*analyse des exemples complets*. Cette méthode consiste à estimer les paramètres à partir de  $\mathcal{D}_{co}$  ensemble des exemples complètement observés dans  $\mathcal{D}_o$ . Lorsque  $\mathcal{D}$  est MCAR, l'estimateur basé sur  $\mathcal{D}_{co}$  n'est pas biaisé. Malheureusement, lorsque le nombre de variables est élevé, la probabilité qu'un exemple soit complètement mesuré devient faible, et  $\mathcal{D}_{co}$  peut être vide ou insuffisant pour que la qualité de l'estimation soit bonne.

Une autre technique, l'*analyse des exemples disponibles*, est particulièrement intéressante dans le cas des réseaux bayésiens. En effet, puisque la loi jointe est décomposée en un produit de probabilités conditionnelles, il n'est pas nécessaire de mesurer toutes les variables pour estimer la loi de probabilité conditionnelle  $P(X_i | Pa(X_i))$ , mais seulement des variables  $X_i$  et  $Pa(X_i)$ . Il suffit donc d'utiliser tous les exemples où  $X_i$  et  $Pa(X_i)$  sont complètement mesurés pour l'estimation de  $P(X_i | Pa(X_i))$ .

### 2.2.3 Traitement des données MAR

De nombreuses méthodes tentent d'estimer les paramètres d'un modèle à partir de données MAR. Citons par exemple le *sequential updating* [122], l'*échantillonnage de Gibbs* [63], et l'algorithme *expectation maximisation* (EM) [44, 88].

Plus récemment, les algorithmes *bound and collapse* [107] et *robust bayesian estimator* [108] cherchent à résoudre le problème quel que soit le type de données manquantes.

L'application de l'algorithme itératif EM aux réseaux bayésiens a été proposée dans [39] et [95] puis adaptée aux grandes bases de données dans [128]. Nous allons présenter les grandes lignes de cet algorithme dans le cas de l'apprentissage statistique puis de l'apprentissage bayésien.

### 2.2.4 Apprentissage statistique et algorithme EM

Soit  $\log P(\mathcal{D} | \theta) = \log P(\mathcal{D}_o, \mathcal{D}_m | \theta)$  la log-vraisemblance des données.  $\mathcal{D}_m$  étant une variable aléatoire non mesurée, cette log-vraisemblance est elle-aussi une variable aléatoire fonction de  $\mathcal{D}_m$ . En se fixant un modèle de référence  $\theta^*$ , il est possible d'estimer la densité de probabilité des données manquantes  $P(\mathcal{D}_m | \theta^*)$ , et ainsi de calculer  $Q(\theta : \theta^*)$ , espérance de la log-vraisemblance précédente :

$$Q(\theta : \theta^*) = E_{\theta^*} [\log P(\mathcal{D}_o, \mathcal{D}_m | \theta)] \quad (2.6)$$

$Q(\theta : \theta^*)$  est donc l'espérance de la vraisemblance d'un jeu de paramètres  $\theta$  quelconque, calculée en utilisant une distribution des données manquantes  $P(\mathcal{D}_m | \theta^*)$ .

Cette équation peut se réécrire de la façon suivante (cf. eq. 2.3) :

$$Q(\theta : \theta^*) = \sum_{i=1}^n \sum_{k=1}^{r_i} \sum_{j=1}^{q_k} N_{ijk}^* \log \theta_{i,j,k} \quad (2.7)$$

où  $N_{i,j,k}^* = E_{\theta^*} [N_{i,j,k}] = N * P(X_i = x_k, Pa(X_i) = pa_j | \theta^*)$  est obtenu par inférence dans le réseau de paramètres  $\theta^*$  si les  $\{ X_i, Pa(X_i) \}$  ne sont pas complètement mesurés et par simple comptage sinon.

L'algorithme EM est très simple : soient  $\theta^{(t)} = \{\theta_{i,j,k}^{(t)}\}$  les paramètres du réseau bayésien à l'itération  $t$ .

- *expectation* : estimer les  $N^*$  de l'équation 2.7 à partir des paramètres de référence  $\theta^{(t)}$ ,
- *maximisation* : choisir la meilleure valeur des paramètres  $\theta^{(t+1)}$  en maximisant  $Q$ ,

$$\theta_{i,j,k}^{(t+1)} = \frac{N_{i,j,k}^*}{\sum_k N_{i,j,k}^*} \quad (2.8)$$

- répéter ces deux étapes tant que l'on arrive à augmenter la valeur de  $Q$ .

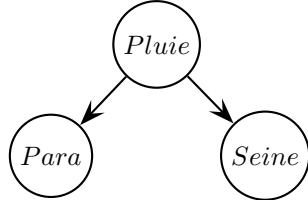
[44] a prouvé la convergence de cet algorithme, ainsi que le fait qu'il n'était pas nécessaire de trouver l'optimum global  $\theta^{(t+1)}$  de la fonction  $Q(\theta : \theta^{(t)})$  mais uniquement une valeur qui permette à la fonction  $Q$  d'augmenter (*Generalized EM*).

De nombreuses heuristiques ont été conçues pour accélérer ou améliorer la convergence de l'algorithme EM [95]. Citons par exemple, l'ajout d'un moment  $\gamma$ , proposé par Nowlan [98] qui permet d'accélérer la convergence si le paramètre  $\gamma$  est bien réglé :

$$\theta_{i,j,k}^{(t+1)} \leftarrow \theta_{i,j,k}^{(t+1)} + \gamma \theta_{i,j,k}^{(t)} \quad (2.9)$$

**Exemple simple :**

Prenons le réseau bayésien et la base d'exemples définis ci-dessous (où « ? » représente une donnée manquante) :



Pluie	Seine
o	?
n	?
o	n
n	n
o	o

*Pluie* = « il pleut à Rouen »,  
*Seine* = « la Seine déborde »,  
*Para* = « j'ai sorti mon parapluie ».

Commençons par définir quels sont les paramètres à estimer :

- $P(Pluie) = [\theta_P \ 1 - \theta_P]$
- $p(P(Seine | Pluie = o)) = [\theta_{S|P=o} \ 1 - \theta_{S|P=o}]$
- $P(Seine | Pluie = n) = [\theta_{S|P=n} \ 1 - \theta_{S|P=n}]$
- idem pour  $P(Para | Pluie)$ ...

Concentrons-nous sur l'estimation des paramètres  $\theta_{S|P=o}$  et  $\theta_{S|P=n}$  avec l'algorithme EM.

TAB. 2.1: Exécution de l'algorithme EM (à suivre ...)

### Initialisation

Les valeurs initiales des paramètres sont :  $\theta_{S|P=o}^{(0)} = 0.3$ ,  $\theta_{S|P=n}^{(0)} = 0.4$

### Première itération

Le calcul de l'étape E est résumé dans le tableau ci-après (les valeurs suivies d'un + sont obtenues par calcul des probabilités selon le modèle  $\theta^{(0)}$ ) :

Pluie	Seine	$P(S   P = o)$		$P(S   P = n)$	
		$S = o$	$S = n$	$S = o$	$S = n$
o	?	0.3 +	0.7 +	0	0
n	?	0	0	0.4 +	0.6 +
o	n	0	1	0	0
n	n	0	0	0	1
o	o	1	0	0	0
	$N^*$	1.3	1.7	0.4	1.6

L'étape M nous donne  $\theta_{S|P=o}^{(1)} = \frac{1.3}{1.3+1.7} = 0.433$  et  $\theta_{S|P=n}^{(1)} = \frac{0.4}{0.4+1.6} = 0.2$

### Deuxième itération

Etape E (les valeurs suivies d'un + sont obtenues par calcul des probabilités selon le modèle  $\theta^{(1)}$  obtenu à l'itération précédente) :

Pluie	Seine	$P(S   P = o)$		$P(S   P = n)$	
		$S = o$	$S = n$	$S = o$	$S = n$
o	?	0.433 +	0.567 +	0	0
n	?	0	0	0.2 +	0.8 +
o	n	0	1	0	0
n	n	0	0	0	1
o	o	1	0	0	0
	$N^*$	1.433	1.567	0.2	1.8

Etape M :  $\theta_{S|P=o}^{(1)} = \frac{1.433}{1.433+1.567} = 0.478$  et  $\theta_{S|P=n}^{(1)} = \frac{0.2}{0.2+1.8} = 0.1$

### Convergence

Après quelques itérations de l'algorithme EM, les valeurs de paramètres convergent vers  $\theta_{S|P=o}^{(t)} = 0.5$  et  $\theta_{S|P=n}^{(t)} = 0$

Dans cet exemple très simple, les données manquantes sont MCAR et les approches *analyse des exemples complets* ou *analyse des exemples disponibles* (cf. p.17) auraient directement fourni la solution.

TAB. 2.1: Exécution de l'algorithme EM

## 2.2.5 Apprentissage bayésien et algorithme EM

L'algorithme EM peut aussi s'appliquer dans le cadre bayésien. Pour l'apprentissage des paramètres, il suffit de remplacer le maximum de vraisemblance de l'étape M par un maximum (ou une espérance) a posteriori. Nous obtenons

dans le cas de l'espérance a posteriori :

$$\theta_{i,j,k}^{(t+1)} = \frac{N_{i,j,k}^* + \alpha_{i,j,k}}{\sum_k (N_{i,j,k}^* + \alpha_{i,j,k})} \quad (2.10)$$

**Exemple simple** : Reprenons l'exemple précédent. Il nous faut ajouter un a priori sur les paramètres, par exemple une distribution de Dirichlet uniforme avec  $\alpha_{i,j,k} = 1$ . L'algorithme EM utilisant un maximum de vraisemblance nous donne :

$$\begin{aligned} - \theta_{S|P=o}^{(1)} &= \frac{1.3+1}{1.3+1.7+2} = 0.46 \text{ et } \theta_{S|P=n}^{(1)} = \frac{0.4+1}{0.4+1.6+2} = 0.35 \\ - \theta_{S|P=o}^{(2)} &= \frac{1.46+1}{1.46+1.54+2} = 0.492 \text{ et } \theta_{S|P=n}^{(1)} = \frac{0.35+1}{0.35+1.65+2} = 0.338 \\ - \dots \\ - \theta_{S|P=o}^{(t)} &= 0.5 \text{ et } \theta_{S|P=n}^{(t)} = 0.333 \end{aligned}$$

L'ajout d'un a priori uniforme sur les paramètres a « empêché » la valeur  $\theta_{S|P=n}^{(t)}$  de tendre vers 0 alors que la configuration  $\{S = o \text{ et } P = n\}$  n'est pas présente dans les données.

TAB. 2.2: Exécution de l'algorithme EM avec a priori de Dirichlet

## 2.3 Incorporation de connaissances

Dans de nombreuses applications réelles, il n'existe pas (ou très peu) de données. Dans ces situations, l'apprentissage des paramètres du réseau bayésien passe par l'utilisation de connaissances d'experts pour tenter d'estimer les probabilités conditionnelles. Cette difficulté, souvent appelée élicitation de probabilités dans la littérature, est générale dans le domaine de l'acquisition de connaissances.

Nous décrirons tout d'abord l'utilisation d'une échelle de probabilité permettant à l'expert d'estimer de manière quantitative ou qualitative la probabilité d'un événement quelconque.

Malheureusement, chaque paramètre d'un réseau bayésien est une loi de probabilité conditionnelle dont la taille augmente exponentiellement par rapport au nombre de parents de la variable considérée. Il n'est donc pas réaliste d'interroger un expert sur toutes les valeurs de chacune de ces lois. Nous détaillerons quelques méthodes permettant de simplifier une loi de probabilité conditionnelle, ce qui diminue le nombre de questions à poser à l'expert. Nous proposerons aussi quelques règles pour vérifier la cohérence des estimations de l'expert.

Pour finir, nous aborderons le problème de l'estimation de la probabilité d'un événement en présence de plusieurs experts ou de *sources d'information multiples*. Comment prendre en compte la fiabilité de ces experts et de ces sources ? et que faire lorsqu'ils sont en désaccord ?

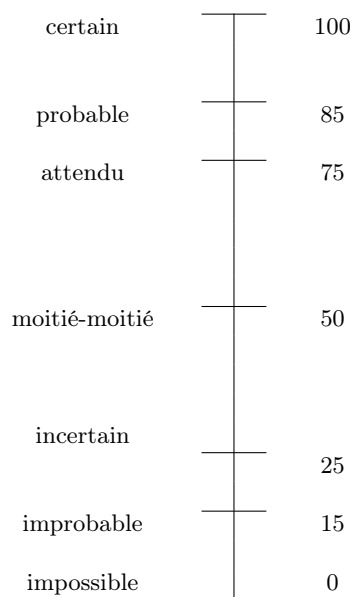


FIG. 2.1 – Echelle de probabilité

### 2.3.1 Comment demander à un expert d’estimer une probabilité ?

De nombreux travaux comme ceux de [109] abordent le sujet de l’éllicitation de probabilités. La tâche la plus difficile est souvent de trouver un expert disponible et familiarisé à la notion de probabilité. Ensuite il faut considérer les biais éventuels parfois subconscients (un expert va souvent surestimer la probabilité de réussite d’un projet le concernant, etc.). La dernière étape consiste à fournir à l’expert des outils associant des notions qualitatives et quantitatives, pour qu’il puisse associer une probabilité aux différents événements. L’outil le plus connu et le plus facile à mettre en œuvre est l’échelle de probabilité [50] présentée figure 2.1. Cette échelle permet aux experts d’utiliser des informations à la fois textuelles et numériques pour assigner un degré de réalisation à telle ou telle affirmation, puis éventuellement de comparer les probabilités des événements pour les modifier. [132] propose une étude détaillée des techniques d’éllicitation de probabilités pour résoudre un problème de diagnostic médical.

### 2.3.2 Quelles probabilités estimer ?

Nous supposons ici que l’expert doit estimer la probabilité conditionnelle  $P(Y | X_1, X_2, \dots, X_n)$  et que toutes nos variables ( $Y$  et  $X_i$ ) soient binaires (de valeurs respectives  $\{y$  et  $\bar{y}\}$  et  $\{x_i$  et  $\bar{x}_i\}$ ).

L’expert devra donc estimer  $2^n$  valeurs, ce qui est peu réaliste pour des problèmes complexes (manque de temps, fiabilité des  $2^n$  valeurs, etc.). Plusieurs approches permettent de simplifier cette probabilité conditionnelle par diverses formes d’approximation comme le modèle OU bruité, les facteurs d’interpolation ou le modèle log-linéaire.

### Modèle OU bruité

Le modèle OU bruité, proposé initialement par Pearl [102], pose les hypothèses suivantes :

- La probabilité suivante (probabilité que  $X_i$  cause  $Y$  lorsque les autres variables  $X_j$  sont absentes) est facile à estimer :

$$p_i = P(y \mid \bar{x}_1, \bar{x}_2, \dots, x_i, \dots, \bar{x}_n) \quad (2.11)$$

- Le fait que  $X_i$  cause  $Y$  est indépendant des autres variables  $X_j$  (pas d'effet mutuel des variables).

Ces hypothèses permettent alors d'affirmer que :

- Si un des  $X_i$  est vrai, alors  $Y$  est presque toujours vrai (avec la probabilité  $p_i$ ),
- Si plusieurs  $X_i$  sont vrais, alors la probabilité que  $Y$  soit vrai est :

$$P(y \mid \mathcal{X}) = 1 - \prod_{i \mid X_i \in \mathcal{X}_p} (1 - p_i) \quad (2.12)$$

où  $\mathcal{X}_p$  est l'ensemble des  $X_i$  vrais.

Ce modèle a été étendu au cas où  $Y$  peut être vrai sans qu'une seule des causes soit vraie (leaky noisy-OR gate) [71] et aux variables multivaluées (generalized noisy-OR gate) [71, 47, 126]. Il s'intègre très facilement aux algorithmes d'inférence tels que les algorithmes de « message passing » ou d'arbre de jonction.

Il est important de noter que cette modélisation simplifiée des probabilités conditionnelles peut aussi être utilisée dans le cadre de l'apprentissage, lorsque le nombre de données est faible. Cette approche a donné de bons résultats dans des domaines tels que le diagnostic médical [106, 99] ou le diagnostic de pannes [16].

### Facteurs d'interpolation

L'utilisation de facteurs d'interpolation a été proposée par [18] pour la détermination pratique de tables de probabilités conditionnelles. A la différence du modèle précédent, l'expert est consulté pour déterminer la probabilité des événements suivants :

$$\bar{p}_i = P(y \mid x_1, x_2, \dots, \bar{x}_i, \dots, x_n) \quad (2.13)$$

$$\bar{p} = P(y \mid \bar{x}_1, \bar{x}_2, \dots, \bar{x}_i, \dots, \bar{x}_n) \quad (2.14)$$

$$p = P(y \mid x_1, x_2, \dots, x_i, \dots, x_n) \quad (2.15)$$

Ces valeurs permettent de calculer les facteurs d'interpolation  $IF_i$  de la façon suivante :

$$IF_i = \frac{\bar{p}_i - \bar{p}}{p - \bar{p}} \quad (2.16)$$

Chacun de ces facteurs peut être interprété comme l'effet relatif (par rapport à  $\bar{p}_i$ , situation où tous les  $X_i$  sont "absents") du passage de  $X_i$  de  $\bar{x}_i$  à  $x_i$  (lorsque tous les autres  $X_j$  sont à  $x_j$ ).

Dans le cas le plus simple proposé par Cain, *parents non modifiants*, l'effet de chaque  $X_i$  sur  $Y$  ne dépend pas de la valeur des autres  $X_j$ . Avec cette hypothèse,

le facteur d'interpolation est donc de manière plus générale l'effet de la variation de  $X_i$  quelles que soient les valeurs prises par les autres  $X_j$ , ce qui nous permet de calculer par récurrence la valeur de n'importe quelle probabilité  $P(y | \mathcal{X})$ , par exemple :

$$P(y | x_1, x_2, \dots, \bar{x}_i, \dots, \bar{x}_j, \dots, x_n) = \bar{p} + IF_i(\bar{p}_j - \bar{p}) \quad (2.17)$$

et ainsi de suite pour les probabilités où  $k$   $X_i$  sont "absents" ( $\bar{x}_i$ ) en faisant intervenir les probabilités où  $(k-1)$   $X_i$  sont "absents" et le facteur d'interpolation de l'autre variable.

Cain adapte ensuite cette utilisation de facteurs d'interpolation à des variables discrètes quelconques. L'approche se généralise aussi au cas où certains parents sont *modifiants* en estimant des facteurs d'interpolation spécifiques à chaque configuration de ces parents modifiants.

### Modèles log-linéaires

Les modèles log-linéaires [34] peuvent aussi être utilisés pour simplifier le nombre de paramètres d'une loi de probabilité conditionnelle, ou plus généralement la loi de probabilité jointe  $P(Y, X_1, X_2, \dots, X_n)$  d'une variable et de ses parents.

Le principe, très général, de ces modèles est de décomposer le logarithme d'une loi de probabilité en une somme de termes décrivant les interactions entre les variables. Cette décomposition est dite *saturée* lorsque tous les termes sont présents dans la décomposition, et *non saturée* lorsque des hypothèses supplémentaires sont ajoutées, comme par exemple le fait que certaines variables soient indépendantes, pour supprimer des termes dans la décomposition.

Dans le cas qui nous intéresse, nous savons aussi que les parents sont mutuellement indépendants. De plus, [38] propose de ne garder que les termes d'interaction d'ordre inférieur ou égal à 2 ( $u_i, u'_i$ ), arrivant au modèle log-linéaire non saturé suivant :

$$\log P(Y, X_1, \dots, X_n) = u + \sum_i u_i(x_i) + \sum_i u'_i(x_i, y) \quad (2.18)$$

La détermination de ces termes d'interaction passe par la résolution d'un système linéaire, en utilisant certaines contraintes comme le fait que la somme des  $P(Y, X_1, \dots, X_n)$  doit être égale à 1. En supposant que l'expert soit interrogé sur toutes les probabilités marginales  $P(x_i)$ ,  $P(y)$ , et sur toutes les probabilités conditionnelles  $P(y | x_i)$  et  $P(y | \bar{x}_i)$ , [38] montre qu'il reste encore  $2^n - 2n$  contraintes à satisfaire pour déterminer complètement les paramètres du modèle log-linéaire.

Cette approche permet donc d'obtenir une modélisation plus générale que les deux premières, mais nécessite davantage d'estimations de la part de l'expert lorsque le nombre de parents d'une variable est important.

### Cohérence des estimations

Les méthodes que nous venons d'étudier permettent de simplifier une distribution de probabilité conditionnelle en estimant un nombre réduit de probabilités d'événements, à l'aide par exemple d'une échelle de probabilité.

[38] propose une série de règles afin de vérifier la cohérence des estimations de l'expert, et éventuellement de corriger automatiquement certaines des proba-

bilités estimées. Cette approche décrite ci-dessous dans le cadre de l'utilisation de modèles log-linéaires se généralise assez facilement aux autres approches :

1. Estimation par l'expert des probabilités marginales  $P(x_i)$  et  $P(y)$ . Ces probabilités correspondent à des événements "non conditionnés" qui sont en général faciles à estimer. Ces valeurs ne sont pas suffisantes, mais permettront par la suite de vérifier la cohérence des estimations de l'expert.
2. Estimation des probabilités conditionnelles  $P(y | x_i)$  et  $P(y | \bar{x}_i)$  pour toutes les variables  $X_i$ .
3. Utilisation des redondances pour vérifier la cohérence des estimations. En effet, nous savons que, pour chaque variable  $X_i$  :

$$P(y) = P(y | x_i)P(x_i) + P(y | \bar{x}_i)(1 - P(x_i)) \quad (2.19)$$

Puisque chacune de ces valeurs a été estimée par l'expert, nous pouvons donc comparer le  $P(y)$  estimé et celui obtenu par l'équation 2.19 pour détecter des incohérences éventuelles.

4. Correction des incohérences. Cette correction peut être soit manuelle, en redemandant à l'expert de ré-estimer les  $P(y | x_i)$  et  $P(y | \bar{x}_i)$  incriminés, soit automatique, en les modifiant tout en gardant leurs proportions respectives pour que l'équation 2.19 soit vérifiée.

### 2.3.3 Comment fusionner les avis de plusieurs experts ?

En ingénierie de la connaissance, l'ingénieur doit souvent faire face à des sources d'informations de diverses natures : experts, données collectées selon des moyens variés, etc. La prise en compte de ces différentes expertises doit se faire avec précaution. Afin d'éviter d'utiliser des données biaisées, Druzdzel et al. [51] proposent un critère pour vérifier si les diverses sources d'informations ont été utilisées dans les mêmes conditions.

Supposons maintenant que plusieurs experts proposent une estimation des mêmes valeurs. Comment faut-il combiner ces différents résultats, en sachant que les experts ne sont pas forcément tous fiables, ou le sont uniquement sur une partie du problème ? La prise en compte de données incertaines a été abordée avec différentes méthodes dont la logique floue [14], les réseaux de neurones (avec par exemple les mélanges d'experts proposés par [73]), ou la théorie des fonctions de croyances [121]. Pour ce dernier cas, S. Populaire et al. [105] proposent une méthode qui permet de combiner l'estimation des probabilités faite par un expert avec celle obtenue grâce à des données.

# Chapitre 3

## Apprentissage de la structure

---

<b>3.1</b>	<b>Introduction</b>	<b>26</b>
<b>3.2</b>	<b>Hypothèses</b>	<b>27</b>
3.2.1	Fidélité	27
3.2.2	Suffisance causale	27
3.2.3	Notion d'équivalence de Markov	27
<b>3.3</b>	<b>Recherche d'indépendances conditionnelles</b>	<b>30</b>
3.3.1	Tests d'indépendance conditionnelle	31
3.3.2	Algorithmes PC et IC	32
3.3.3	Quelques améliorations	36
<b>3.4</b>	<b>Algorithmes basés sur un score</b>	<b>37</b>
3.4.1	Les scores possibles	37
3.4.2	Déterminer un a priori sur les structures	40
3.4.3	Pourquoi chercher la meilleure structure?	40
3.4.4	Recherche dans l'espace des réseaux bayésiens	41
3.4.5	Algorithmes basés sur un score et données incomplètes	50
<b>3.5</b>	<b>Recherche dans l'espace des classes d'équivalence de Markov</b>	<b>53</b>
<b>3.6</b>	<b>Méthodes hybrides</b>	<b>61</b>
<b>3.7</b>	<b>Incorporation de connaissances</b>	<b>62</b>
3.7.1	Structures de réseaux bayésiens pour la classification	62
3.7.2	Structures de réseaux bayésiens avec variables latentes	65
3.7.3	Autres structures particulières	66
<b>3.8</b>	<b>Découverte de variables latentes</b>	<b>66</b>
3.8.1	Recherche d'indépendances conditionnelles	66
3.8.2	Algorithmes basés sur un score	68
<b>3.9</b>	<b>Cas particulier des réseaux bayésiens causaux</b>	<b>68</b>
3.9.1	Définition	69
3.9.2	Apprentissage sans variables latentes	69
3.9.3	Apprentissage avec variables latentes	70

---

### 3.1 Introduction

Dans le chapitre 2, nous avons examiné différentes méthodes d'apprentissage des paramètres d'un réseau bayésien à partir de données complètes ou incomplètes, ou à l'aide d'un expert, en supposant que la structure de ce réseau était déjà connue. Se pose maintenant la question de l'apprentissage de cette structure : comment trouver la structure qui représentera le mieux notre problème.

Avant d'aborder les deux grandes familles d'approches (recherche d'indépendances conditionnelles et méthodes basées sur un score), nous commencerons par rappeler le cadre dans lequel nous travaillons. Ainsi l'apprentissage de la structure d'un réseau bayésien à partir de données revient à trouver un graphe qui soit une P-map d'un modèle d'indépendance associé à une distribution de probabilité dont nous possédons un échantillon. Il faut donc être sûr de l'existence d'une telle P-map (**fidélité**) et de bien connaître toutes les variables (**suffisance causale**).

Nous évoquerons ensuite une notion générale, l'**équivalence de Markov**, qui nous sera utile dans les deux types d'approche, notion liée au fait que plusieurs graphes avec le même squelette pourront représenter les mêmes indépendances conditionnelles.

Comme précédemment, nous pourrions aussi distinguer trois cas :

- les données sont complètes et représentent totalement le problème.
- les données sont incomplètes et/ou il existe des variables latentes.
- peu de données sont disponibles, et il faut utiliser une connaissance experte.

Une première approche, proposée initialement par Spirtes et al. d'un côté, et Pearl et Verma de l'autre, consiste à rechercher les différentes **indépendances conditionnelles** qui existent entre les variables. Les autres approches tentent de quantifier l'adéquation d'un réseau bayésien au problème à résoudre, c'est-à-dire d'**associer un score à chaque réseau bayésien**. Puis elles recherchent la structure qui donnera le meilleur score dans l'espace  $\mathbb{B}$  des graphes acycliques dirigés. Un parcours exhaustif est impossible en pratique en raison de la taille de l'espace de recherche. La formule 3.1 démontrée par [114] prouve que le nombre de structures possibles à partir de  $n$  nœuds est super-exponentiel (par exemple,  $NS(5) = 29281$  et  $NS(10) = 4.2 \times 10^{18}$ ).

$$NS(n) = \begin{cases} 1 & , n = 0 \text{ ou } 1 \\ \sum_{i=1}^n (-1)^{i+1} \binom{n}{i} 2^{i(n-1)} NS(n-i), & n > 1 \end{cases} \quad (3.1)$$

Pour résoudre ce problème ont été proposées un certain nombre d'heuristiques de **recherche dans l'espace**  $\mathbb{B}$ , qui restreignent cet espace à l'espace des arbres (MWST), ordonnent les nœuds pour limiter la recherche des parents possibles pour chaque variable (K2), ou effectuent une recherche gloutonne dans  $\mathbb{B}$  (GS).

En partant du principe que plusieurs structures encodent les mêmes indépendances conditionnelles (équivalence de Markov) et possèdent le même score, d'autres méthodes proposent de **parcourir l'espace  $\mathbb{E}$  des représentants des classes d'équivalence de Markov**, espace certes super-exponentiel (mais légèrement plus petit) mais qui possède de meilleures propriétés.

Nous nous intéresserons aussi aux méthodes qui permettent d'incorporer

des connaissances a priori sur le problème à résoudre en détaillant plus précisément l'**apprentissage de structure dans le cadre de la classification**, et l'**apprentissage de structure lorsque des variables latentes sont définies explicitement**.

Pour répondre à ces différentes questions, nous examinerons successivement les méthodes existantes, en détaillant à chaque fois une des approches les plus représentatives. Nous finirons en abordant quelques problèmes ouverts dans l'apprentissage de structure : la découverte automatique de variables latentes et l'apprentissage de réseaux bayésiens "réellement" causaux.

## 3.2 Hypothèses

### 3.2.1 Fidélité

Les liens entre modèle d'indépendance et réseau bayésien sont largement décrits dans [145]. Un réseau bayésien n'est pas capable de représenter n'importe quelle distribution de probabilité (ou la liste des indépendances conditionnelles associées). La première hypothèse que nous poserons est donc l'existence d'un réseau bayésien qui soit la P-map du modèle d'indépendance associé à la distribution de probabilité  $P$  sous-jacente à nos données. Cette hypothèse se retrouve souvent sous le terme de fidélité (*faithfulness*) entre le graphe et  $P$ .

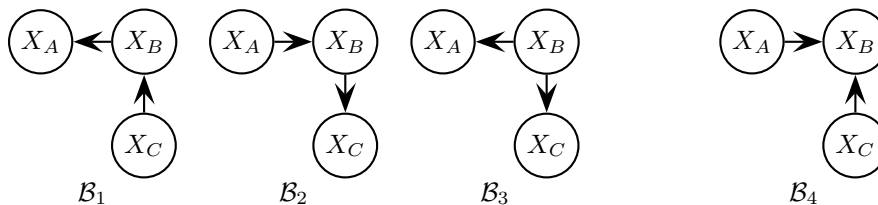
### 3.2.2 Suffisance causale

L'autre hypothèse importante est celle de suffisance causale. Un ensemble de variables  $\mathcal{X}$  est suffisant causalement pour une population donnée  $\mathcal{D}$  si et seulement si dans cette population, chaque cause  $Y$  commune à plusieurs variables de  $\mathcal{X}$  appartient aussi à  $\mathcal{X}$ , ou si  $Y$  est constant pour toute la population. Cela signifie que l'ensemble  $\mathcal{X}$  est suffisant pour représenter toutes les relations d'indépendances conditionnelles qui pourraient être extraites des données.

### 3.2.3 Notion d'équivalence de Markov

Deux réseaux bayésiens  $\mathcal{B}_1$  et  $\mathcal{B}_2$  sont dits équivalents au sens de Markov ( $\mathcal{B}_1 \equiv \mathcal{B}_2$ ) s'ils représentent les mêmes relations d'indépendance conditionnelle.

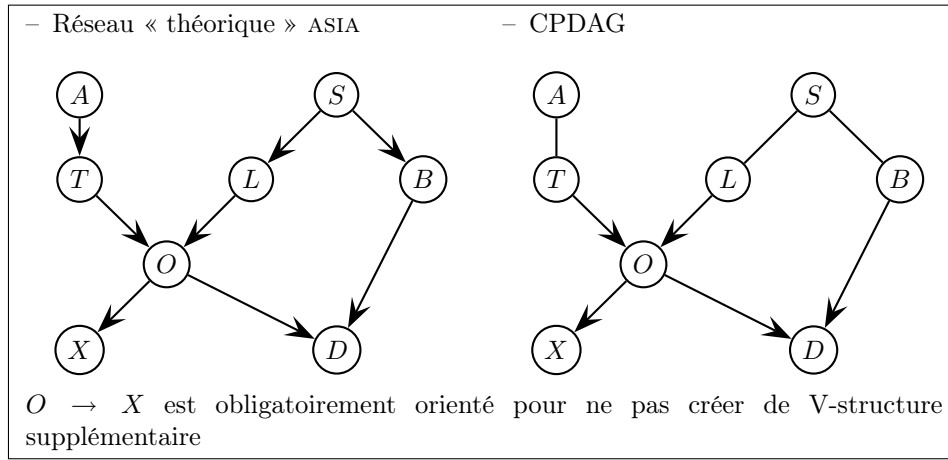
Afin d'illustrer simplement cette notion, montrons que les structures  $\mathcal{B}_1$ ,  $\mathcal{B}_2$  et  $\mathcal{B}_3$  décrites ci-dessous sont équivalentes.



Montrons-le pour  $\mathcal{B}_1$  et  $\mathcal{B}_2$  :

Selon  $\mathcal{B}_1$  :  $P(X_A, X_B, X_C)_{\mathcal{B}_1} = P(X_A | X_B) * P(X_B | X_C) * P(X_C)$

Selon  $\mathcal{B}_2$  :  $P(X_A, X_B, X_C)_{\mathcal{B}_2} = P(X_A) * P(X_B | X_A) * P(X_C | X_B)$



TAB. 3.1 – Exemple de réseau bayésien et son représentant dans l'espace des classes d'équivalence de Markov

Mais d'après la définition d'une probabilité conditionnelle,

$$\begin{aligned}
 P(X_A, X_B) &= P(X_A | X_B) * P(X_B) * P(X_A) * P(X_B | X_A) \\
 P(X_B, X_C) &= P(X_C | X_B) * P(X_B) * P(X_C) * P(X_B | X_C)
 \end{aligned}$$

donc

$$\begin{aligned}
 P(X_A, X_B, X_C)_{\mathcal{B}_2} &= P(X_A) * P(X_B | X_A) * P(X_C | X_B) \\
 &= P(X_A | X_B) * P(X_B) * P(X_C | X_B) \\
 &= P(X_A | X_B) * P(X_B | X_C) * P(X_C) \\
 &= P(X_A, X_B, X_C)_{\mathcal{B}_1}
 \end{aligned}$$

Les réseaux bayésiens  $\mathcal{B}_1$  et  $\mathcal{B}_2$  sont donc équivalents (id. avec  $\mathcal{B}_3$ ).

Par contre, ces trois structures ne sont pas équivalentes à la V-structure  $\mathcal{B}_4$ . En effet, nous avons  $P(X_A, X_B, X_C)_{\mathcal{B}_4} = P(X_A) * P(X_C) * P(X_B | X_A, X_C)$  et le terme  $P(X_B | X_A, X_C)$  ne peut pas se simplifier.

Verma et Pearl [134] ont démontré que tous les DAG équivalents possèdent le même squelette (graphe non dirigé) et les mêmes V-structures. Une classe d'équivalence, c'est-à-dire un ensemble de réseaux bayésiens qui sont tous équivalents, peut donc être représentée par le graphe sans circuit partiellement dirigé (PDAG) qui a la même structure que tous les réseaux équivalents, mais pour lequel les arcs réversibles (n'appartenant pas à des V-structures, ou dont l'inversion ne génère pas de V-structure) sont remplacés par des arêtes (non orientées). Le DAG partiellement dirigé ainsi obtenu est dit *complété* (CPDAG) ou graphe essentiel [6]. La table 3.1 nous donne le graphe ASIA et son CPDAG représentant dans l'espace des classes d'équivalence de Markov. Ce CPDAG possède bien le même squelette que le DAG initial ainsi que ses deux V-structures. De plus, l'arc  $O \rightarrow X$  est obligatoirement orienté dans ce sens pour ne pas créer de V-structure supplémentaire.

Algorithme DAGtoCPDAG

- Ordonner les arcs du DAG
- $\forall \text{arc}, \text{étiquette}(\text{arc}) \leftarrow \emptyset$
- $\mathcal{A} \leftarrow$  liste des arcs non étiquetés
- Répéter
  - $(X_i, X_j) \leftarrow \min_{\mathcal{A}}(\text{arc})$  (plus petit arc non étiqueté)
  - $\forall X_k / \text{étiquette}(X_k, X_i) = \text{NonRéversible}$
  - $Fin \leftarrow \text{Faux}$
  - si  $X_k \notin pa(X_j)$  alors
    - $\text{étiquette}(*, X_j) \leftarrow \text{NonRéversible}$
    - $\mathcal{A} \leftarrow \mathcal{A} \setminus (*, X_j)$
    - $Fin \leftarrow \text{Vrai}$
  - sinon
    - $\text{étiquette}(X_k, X_j) \leftarrow \text{NonRéversible}$
    - $\mathcal{A} \leftarrow \mathcal{A} \setminus (X_k, X_j)$
  - si  $Fin = \text{Faux}$  alors
    - si  $\exists \text{arc}(X_k, X_j) / X_k \notin pa(X_i) \cup \{X_i\}$  alors
      - $\forall (X_k, X_j) \in \mathcal{A},$
      - $\text{étiquette}(X_k, X_j) \leftarrow \text{NonRéversible}$
      - $\mathcal{A} \leftarrow \mathcal{A} \setminus (X_k, X_j)$
    - sinon
      - $\forall (X_k, X_j) \in \mathcal{A},$
      - $\text{étiquette}(X_k, X_j) \leftarrow \text{réversible}$
      - $\mathcal{A} \leftarrow \mathcal{A} \setminus (X_k, X_j)$

Tant que  $\mathcal{A} \neq \emptyset$

Ordonner-Arc

- Trier les  $X_i$  dans l'ordre topologique
- $k \leftarrow 0$
- $\mathcal{A} \leftarrow$  liste des arcs (non ordonnés)
- Répéter
  - $X_j \leftarrow \min_j(X_j / (X_i, X_j) \in \mathcal{A})$   
plus petit nœud destination d'un arc non ordonné
  - $X_i \leftarrow \max_i(X_i / (X_i, X_j) \in \mathcal{A})$   
plus grand nœud origine d'un arc non ordonné vers  $X_j$
  - $Ordre(X_i, X_j) \leftarrow k$
  - $k \leftarrow k + 1$
  - $\mathcal{A} \leftarrow \mathcal{A} \setminus (X_i, X_j)$

Tant que  $\mathcal{A} \neq \emptyset$

TAB. 3.2 – Algorithme DAGtoCPDAG

Chickering [31] propose une méthode pour passer du DAG d'un réseau bayésien au CPDAG représentant sa classe d'équivalence de Markov. Pour cela, il faut commencer par ordonner tous les arcs du réseau de départ (algorithme *Ordonner-Arc*, table 3.2), puis parcourir l'ensemble des arcs ainsi ordonnés pour « simplifier » les arcs réversibles (algorithme *DAGtoCPDAG*, table 3.2).

Il existe plusieurs algorithmes « inverses » capables de générer un des réseaux

<p>Algorithme PDAGtoDAG</p> <ul style="list-style-type: none"> <li>• <math>\mathcal{B} \leftarrow PDAG</math></li> <li>• <math>\mathcal{A} \leftarrow</math> liste des arêtes de <math>PDAG</math></li> <li>• Répéter <ul style="list-style-type: none"> <li>Recherche d'un nœud <math>X_i</math> tel que <ul style="list-style-type: none"> <li>- il n'existe aucun arc <math>X_i \leftarrow X_j</math> dans <math>\mathcal{A}</math></li> <li>- et pour tout <math>X_j</math> tel qu'il existe <math>X_i - X_j</math> dans <math>\mathcal{A}</math>,  <math>X_j</math> est adjacent à tous les autres nœuds adjacents à <math>X_i</math></li> </ul> </li> <li>Si <math>X_i</math> n'existe pas alors  <math>PDAG</math> n'admet aucune extension complètement dirigée</li> <li>sinon <ul style="list-style-type: none"> <li><math>\forall X_j</math> tel que <math>X_i - X_j \in \mathcal{A}</math>  <math>X_i \rightarrow X_j</math> dans <math>\mathcal{B}</math></li> <li><math>\mathcal{A} \leftarrow \mathcal{A} \setminus (X_i, X_j)</math></li> </ul> </li> </ul> </li> </ul> <p>Tant Que <math>\mathcal{A} \neq \emptyset</math></p>	
Notations :	<ul style="list-style-type: none"> <li><math>PDAG</math>   graphe acyclique partiellement dirigé</li> <li><math>\mathcal{B}</math>   DAG complètement dirigé, extension consistante de <math>PDAG</math></li> </ul>

TAB. 3.3 – Algorithme PDAGtoDAG

bayésiens équivalents à partir d'un PDAG, si ce PDAG est bien le représentant d'une classe d'équivalence (on dit alors que le DAG résultant est une extension consistante du PDAG de départ). Nous décrivons dans la table 3.3 l'algorithme PDAGtoDAG proposé par Dor et Tarsi [48]. Notons qu'il est aussi possible d'utiliser les règles d'orientation d'arcs proposées par les algorithmes IC et PC que nous décrivons dans les prochaines sections (table 3.4) puisqu'elles résolvent également la même tâche.

### 3.3 Recherche d'indépendances conditionnelles

Cette première série d'approches d'apprentissage de structure, souvent appelée recherche sous contraintes, est issue des travaux de deux équipes « concurrentes », Pearl et Verma d'une part avec les algorithmes IC et IC\*, Spirtes, Glymour et Scheines de l'autre avec les algorithmes SGS, PC, CI, FCI. Citons de même, plus récemment, l'algorithme BN-PC de Cheng et al. [21, 22, 25]. Ces algorithmes sont tous basés sur un principe identique :

- construire un graphe non dirigé contenant les relations entre les variables, à partir de tests d'indépendance conditionnelle,
- détecter les V-structures (en utilisant aussi des tests d'indépendance conditionnelle),
- « propager » les orientations de certains arcs,
- prendre éventuellement en compte les causes artificielles dues à des variables latentes (cf. section 3.8).

La caractéristique principale de toutes ces méthodes réside dans la détermination à partir de données des relations d'indépendance conditionnelle entre deux variables quelconques conditionnellement à un ensemble de variables. Ceci nous amènera à évoquer les **tests statistiques d'indépendance** classiquement utilisés. Nous passerons ensuite en revue les algorithmes principaux issus de ces travaux et les améliorations qui y ont été apportées.

### 3.3.1 Tests d'indépendance conditionnelle

Les tests statistiques classiquement utilisés pour tester l'indépendance conditionnelle sont les tests du  $\chi^2$  et du rapport de vraisemblance  $G^2$ . Détaillons le test d'indépendance du  $\chi^2$  puis son utilisation dans le cadre de l'indépendance conditionnelle.

Soit deux variables discrètes  $X_A$  et  $X_B$ , de taille respective  $r_A$  et  $r_B$ . Soit  $N_{ab}$  le nombre d'occurrences de  $\{X_A = x_a \text{ et } X_B = x_b\}$  dans la base d'exemples,  $N_{a.}$  le nombre d'occurrences de  $\{X_A = x_a\}$  et  $N_{.b}$  le nombre d'occurrences de  $\{X_B = x_b\}$ .

Le test du  $\chi^2$  met en concurrence deux modèles :

- le modèle observé  $p_o = P(X_A, X_B)$ , représenté par les occurrences observées  $O_{ab} = N_{ab}$
- le modèle théorique  $p_t = P(X_A)P(X_B)$ , représenté par les occurrences théoriques  $T_{ab} = \frac{N_{a.} * N_{.b}}{N}$ .

Soit la statistique suivante (de degré de liberté  $df = (r_A - 1)(r_B - 1)$ ) :

$$\chi^2 = \sum_{a=1}^{r_A} \sum_{b=1}^{r_B} \frac{(O_{ab} - T_{ab})^2}{T_{ab}} = \sum_{a=1}^{r_A} \sum_{b=1}^{r_B} \frac{(N_{ab} - \frac{N_{a.} * N_{.b}}{N})^2}{\frac{N_{a.} * N_{.b}}{N}} \quad (3.2)$$

L'hypothèse d'indépendance entre  $X_A$  et  $X_B$  est vérifiée si et seulement si  $\chi^2 < \chi_{théorique}^2(df, 1 - \alpha)$  (pour un seuil de confiance  $\alpha$ )

Lorsqu'un effectif  $T_{ab}$  est faible ( $T_{ab} < 10$ ), la formule 3.2 n'est plus applicable. Il faut alors remplacer le terme  $\frac{(O_{ab} - T_{ab})^2}{T_{ab}}$  par  $\frac{(|O_{ab} - T_{ab}| - 0.5)^2}{T_{ab}}$  (Correction de Yates).

Spirtes et al. proposent aussi d'utiliser le rapport de vraisemblance  $G^2$  (qui suit encore une loi du  $\chi^2$  de degré de liberté  $df = (r_A - 1)(r_B - 1)$ ) :

$$G^2 = 2 \sum_{a=1}^{r_A} \sum_{b=1}^{r_B} O_{ab} \ln\left(\frac{O_{ab}}{T_{ab}}\right) = 2 \sum_{a=1}^{r_A} \sum_{j=1}^{q_B} N_{ab} \ln\left(\frac{N_{ab} * N}{N_{a.} * N_{.b}}\right) \quad (3.3)$$

Notons que ce rapport de vraisemblance est relativement proche de l'information mutuelle entre les variables  $X_A$  et  $X_B$ , notion qui sera reprise par certaines fonctions de score des réseaux bayésiens (voir équations 3.7 et 3.8).

Les équations 3.2 et 3.3 testent l'indépendance entre deux variables. L'utilisation de ces tests pour la recherche de structure dans les réseaux bayésiens nécessite une adaptation pour les tests d'indépendance conditionnelle entre deux variables  $X_A$  et  $X_B$  conditionnellement à un ensemble quelconque de variables  $\mathcal{X}_C$ . Pour cela le principe ne change pas, il faut mettre en concurrence les deux modèles suivants :

- le modèle observé  $p_o = P(X_A, X_B | \mathcal{X}_C)$ , représenté par les occurrences observées  $O_{abc} = N_{abc}$  où  $N_{abc}$  est le nombre d'occurrences de  $\{X_A = x_a, X_B = x_b \text{ et } \mathcal{X}_C = \mathbf{x}_c\}$ .
- le modèle théorique  $p_t = P(X_A | \mathcal{X}_C)P(X_B | \mathcal{X}_C)$ , représenté par les occurrences théoriques  $T_{abc} = \frac{N_{a..c} * N_{.bc}}{N_{..c}}$ .

Soit la statistique suivante (de degré de liberté  $df = (r_A - 1)(r_B - 1)r_C$ ) :

$$\chi^2 = \sum_{a=1}^{r_A} \sum_{b=1}^{r_B} \sum_{c=1}^{r_C} \frac{(O_{abc} - T_{abc})^2}{T_{abc}} \quad (3.4)$$

L'hypothèse d'indépendance entre  $X_A$  et  $X_B$  conditionnellement à  $\mathcal{X}_C$  est vérifiée si  $\chi^2 < \chi_{théorique}^2(df, 1 - \alpha)$  (pour un seuil de confiance  $\alpha$ ).

Se pose ici un inconvénient majeur lorsque le nombre de variables disponibles est important : plus  $\mathcal{X}_C$  est grand, plus la somme de l'équation 3.4 contient de termes ( $df$  croît exponentiellement) et plus les  $N_{abc}$  sont faibles, ce qui rend le test du  $\chi^2$  peu applicable en grande dimension.

Spirtes et al. ont recours à une heuristique simple pour pallier cet inconvénient : si le nombre de données n'est pas suffisamment important par rapport au degré de liberté ( $df < \frac{N}{10}$ ), alors l'hypothèse est rejetée et les variables  $X_A$  et  $X_B$  sont déclarées dépendantes conditionnellement à  $\mathcal{X}_C$ .

Grâce à ces tests statistiques, il est possible de déterminer une série de contraintes sur la structure du réseau bayésien recherché : une indépendance entre deux variables se traduit par l'absence d'arc entre deux nœuds, une dépendance conditionnelle correspond à une V-structure, etc. Nous allons maintenant étudier les deux familles d'algorithmes qui utilisent ces informations pour apprendre la structure du réseau bayésien.

### 3.3.2 Algorithmes PC et IC

La détermination des indépendances conditionnelles à partir de données permet donc de générer la structure du réseau bayésien représentant toutes ces indépendances. Sur ce principe, Spirtes, Glymour et Scheines [123] ont tout d'abord élaboré l'algorithme SGS. Celui-ci part d'un graphe non orienté complètement relié, et teste toutes les indépendances conditionnelles pour supprimer des arêtes. Il cherche ensuite toutes les V-structures et propage l'orientation des arcs obtenus sur les arêtes adjacentes. Cette méthode requiert malheureusement un nombre de tests d'indépendance conditionnelle exponentiel par rapport au nombre de variables. Spirtes et al. ont alors proposé une variation de SGS, l'**algorithme PC** [123] détaillé dans la table 3.4 qui limite les tests d'indépendance aux indépendances d'ordre 0 ( $X_A \perp X_B$ ), puis aux indépendances conditionnelles d'ordre 1 ( $X_A \perp X_B | \mathcal{X}_C$ ), et ainsi de suite.

L'exemple 3.5 illustre la façon dont les tests d'indépendance conditionnelle permettent de simplifier le graphe non dirigé complètement connecté du départ (étapes 1a à 1c), puis dirigent les arêtes des V-structures détectées dans les données (étape 2). À l'issue de ces deux étapes, le graphe obtenu est un CPDAG qu'il faut finir d'orienter, en prenant soin de ne pas ajouter de V-structures non détectées précédemment (étapes 3 et 4). Notons que les règles proposées par Spirtes et al. pour ces deux dernières étapes peuvent être implémentées

Algorithme PC

- Construction d'un graphe non orienté
  - Soit  $\mathcal{G}$  le graphe reliant complètement tous les nœuds  $\mathcal{X}$
  - $i \leftarrow 0$
  - Répéter
    - Recherche des indépendances cond. d'ordre  $i$
    - $\forall \{X_A, X_B\} \in \mathcal{X}^2$  tels que  $X_A - X_B$  et  $Card(Adj(\mathcal{G}, X_A, X_B)) \geq i$
    - $\forall \mathcal{S} \subset Adj(\mathcal{G}, X_A, X_B)$  tel que  $Card(\mathcal{S}) = i$
    - si  $X_A \perp X_B \mid \mathcal{S}$  alors
      - suppression de l'arête  $X_A - X_B$  dans  $\mathcal{G}$
      - $SepSet(X_A, X_B) \leftarrow SepSet(X_A, X_B) \cup \mathcal{S}$
      - $SepSet(X_B, X_A) \leftarrow SepSet(X_B, X_A) \cup \mathcal{S}$
    - $i \leftarrow i + 1$
  - Jusqu'à  $Card(Adj(\mathcal{G}, X_A, X_B)) < i, \forall \{X_A, X_B\} \in \mathcal{X}^2$
- Recherche des V-structures
  - $\forall \{X_A, X_B, X_C\} \in \mathcal{X}^3$  tels que  $\neg \overline{X_A X_B}$  et  $X_A - X_C - X_B$ ,
  - si  $X_C \notin SepSet(X_A, X_B)$  alors on crée une V-structure :  
 $X_A \rightarrow X_C \leftarrow X_B$
- Ajout récursif de  $\rightarrow$ 
  - Répéter
  - $\forall \{X_A, X_B\} \in \mathcal{X}^2$ ,
  - si  $X_A - X_B$  et  $X_A \rightsquigarrow X_B$ , alors ajout d'une flèche à  $X_B$  :  
 $X_A \rightarrow X_B$
  - si  $\neg \overline{X_A X_B}, \forall X_C$  tel que  $X_A \rightarrow X_C$  et  $X_C - X_B$  alors  $X_C \rightarrow X_B$
  - Tant qu'il est possible d'orienter des arêtes

Notations :

$\mathcal{X}$	ensemble de tous les nœuds
$Adj(\mathcal{G}, X_A)$	ensemble des nœuds adjacents à $X_A$ dans $\mathcal{G}$
$Adj(\mathcal{G}, X_A, X_B)$	$Adj(\mathcal{G}, X_A) \setminus \{X_B\}$
$X_A - X_B$	il existe une arête entre $X_A$ et $X_B$
$X_A \rightarrow X_B$	il existe un arc de $X_A$ vers $X_B$
$\overline{X_A X_B}$	$X_A$ et $X_B$ adjacents $X_A - X_B, X_A \rightarrow X_B$ ou $X_B \rightarrow X_A$
$X_A \rightsquigarrow X_B$	il existe un chemin dirigé reliant $X_A$ et $X_B$

TAB. 3.4 – Algorithme PC

de manière plus systématique par l'algorithme de Dor et Tarsi (cf. Algo. 3.3) détaillé dans la section 3.2.3.

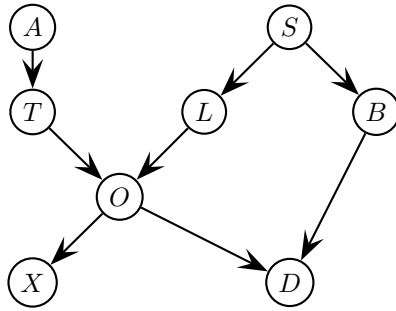
La première étape de l'algorithme PC (recherche d'indépendances conditionnelles) est l'étape la plus coûteuse de l'algorithme. Spirtes et al. ont suggéré plusieurs simplifications ou heuristiques afin de diminuer cette complexité.

- Dans l'algorithme PC\*, ils proposent de ne plus parcourir tous les  $\mathcal{S}$  possibles, mais seulement les ensembles de variables adjacentes à  $X_A$  ou  $X_B$  qui sont sur un chemin entre  $X_A$  et  $X_B$ . Cette solution est malheureusement inutilisable avec un trop grand nombre de variables puisqu'elle équivaut à stocker tous les chemins possibles dans le graphe.
- Trois heuristiques permettent d'accélérer l'algorithme PC en choisissant judicieusement les nœuds  $X_A$  et  $X_B$  et l'ensemble  $\mathcal{S}$  :

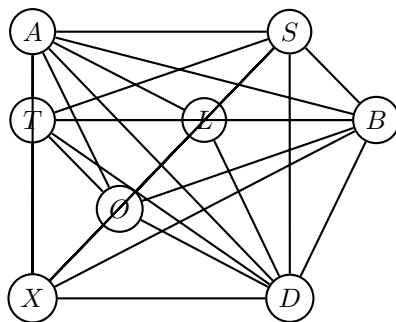
- PC-1 : les couples de variables  $\{X_A, X_B\}$  et les ensembles  $\mathcal{S}$  possibles sont parcourus dans l'ordre lexicographique.
- PC-2 : les couples de variables  $\{X_A, X_B\}$  sont testés dans l'ordre croissant de la statistique utilisée pour le test d'indépendance (des moins dépendants aux plus dépendants). Les ensembles  $\mathcal{S}$  sont parcourus dans l'ordre lexicographique.
- PC-3 : pour une variable  $X_A$  fixée, sont testés d'abord les  $X_B$  les moins dépendants à  $X_A$  conditionnellement aux ensembles  $\mathcal{S}$  les plus dépendants à  $X_A$ .

L'algorithme IC (*Inductive Causation*), développé par Pearl [103], est basé sur le même principe, mais construit le graphe non orienté en ajoutant des arêtes au lieu d'en supprimer. Il faut noter que Pearl [104] a présenté en 1991 un algorithme IC différent qui prend en compte les variables latentes. Cet algorithme, renommé IC\* dans [103], est détaillé dans la section 3.8.

- Le réseau « théorique » ASIA est utilisé pour générer 5000 exemples :

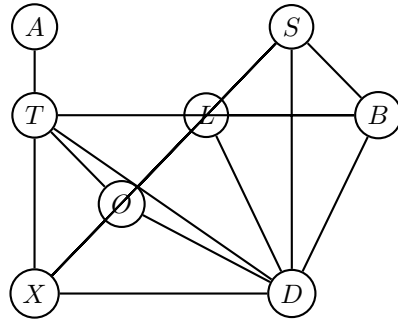


- Etape 0 : Génération du graphe non orienté reliant tous les nœuds :



TAB. 3.5: Exécution de l'algorithme PC (à suivre ...)

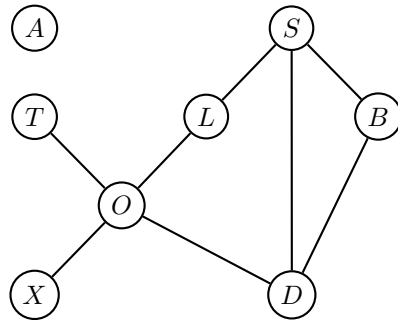
- Etape 1a : Suppression des indépendances conditionnelles d'ordre 0 :



Test du  $\chi^2$  sur les données :

$S \perp A$	$L \perp A$	$B \perp A$
$O \perp A$	$X \perp A$	$D \perp A$
$T \perp S$	$L \perp T$	
$O \perp B$	$X \perp B$	

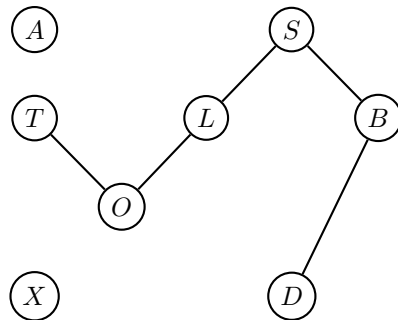
- Etape 1b : Suppression des indépendances conditionnelles d'ordre 1



Test du  $\chi^2$  sur les données :

$T \perp A \mid O$	$O \perp S \mid L$
$X \perp S \mid L$	$B \perp T \mid S$
$X \perp T \mid O$	$D \perp T \mid O$
$B \perp L \mid S$	$X \perp L \mid O$
$D \perp L \mid O$	$D \perp X \mid O$

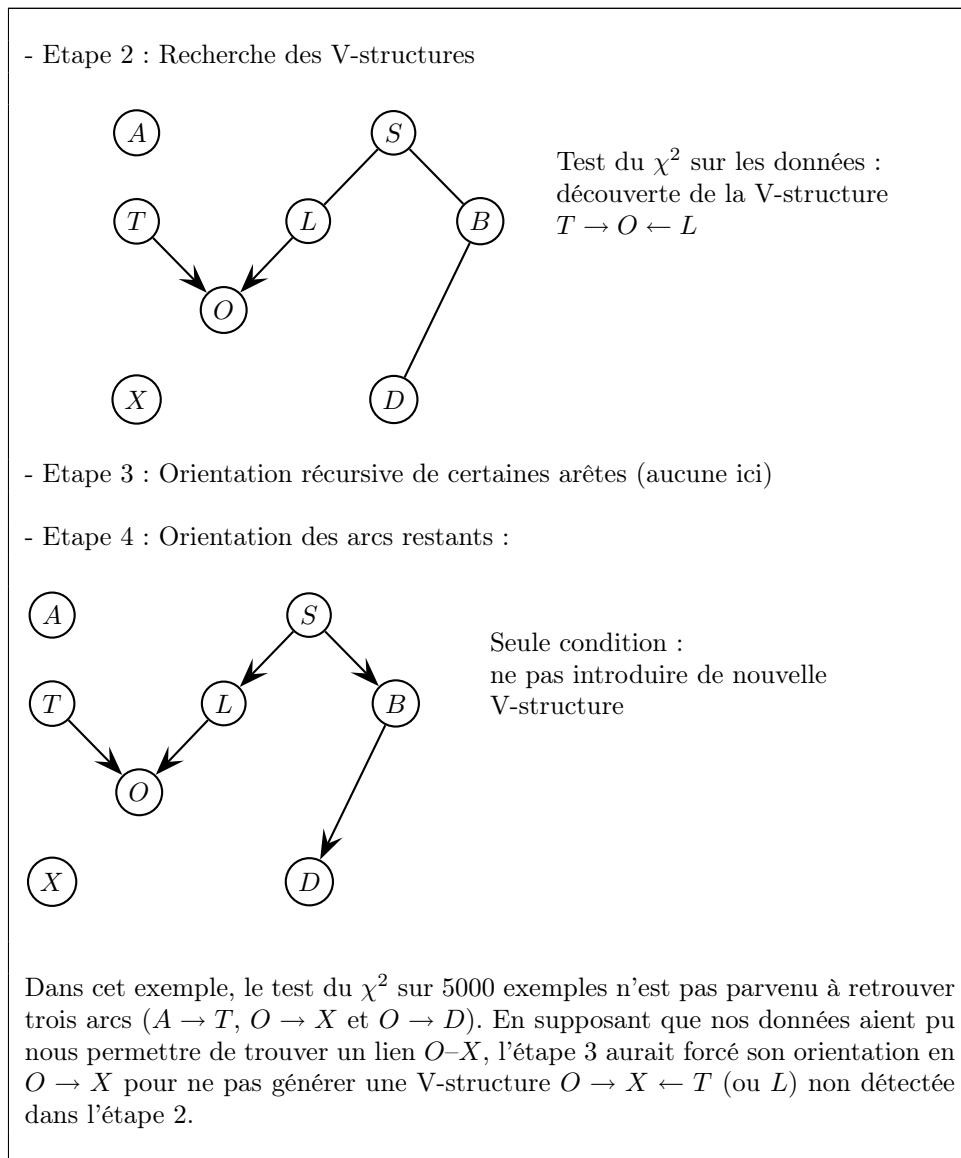
- Etape 1c : Suppression des indépendances conditionnelles d'ordre 2



Test du  $\chi^2$  sur les données :

$D \perp S \mid \{L, B\}$
$X \perp O \mid \{T, L\}$
$D \perp O \mid \{T, L\}$

TAB. 3.5: Exécution de l'algorithme PC (à suivre ...)



TAB. 3.5: Exécution de l'algorithme PC

### 3.3.3 Quelques améliorations

Des travaux récents ont repris le principe des algorithmes IC et PC en tentant de diminuer le nombre de tests d'indépendance conditionnelle nécessaires dans les deux premières étapes de ces algorithmes. Ces travaux s'inspirent aussi de méthodes d'apprentissage basées sur des scores que nous présenterons en 3.4. Citons, par exemple, l'approche par squelette de van Dijk et al. [133], celle de de Campos et al. [43], ou les deux algorithmes BN-PC A et B proposés par Cheng et al. [25] qui ont donné naissance à un logiciel d'apprentissage de réseaux bayésiens *Belief Network PowerConstructor*.

L'algorithme BN-PC-B [22] est le plus général des deux. Le principe de cet algorithme est simple et se décompose en trois phases : (1) utiliser l'arbre de recouvrement maximal (MWST, voir algorithme 3.6), arbre qui relie les variables de manière optimale au sens de l'information mutuelle comme graphe non dirigé de départ, puis (2) effectuer un nombre réduit de tests d'indépendance conditionnelle pour ajouter des arêtes à cet arbre, et (3) finir avec une dernière série de tests pour supprimer les arêtes inutiles et détecter les V-structures. Le graphe partiellement dirigé obtenu à l'issue de la phase C est alors orienté complètement de la même manière que pour les algorithmes IC et PC.

Afin de diminuer le nombre de  $O(n^4)$  tests d'indépendance conditionnelle à effectuer dans le pire des cas pour BN-PC-B, l'algorithme BN-PC-A [21] considère un ordre des nœuds qui permet d'orienter les arêtes dès la phase 1 de l'algorithme. Sa complexité est alors au maximum en  $O(n^2)$  au lieu de  $O(n^4)$ .

### 3.4 Algorithmes basés sur un score

Contrairement à la première famille de méthodes qui tentait de retrouver des indépendances conditionnelles entre les variables, les approches suivantes vont soit chercher la structure qui maximise un certain score, soit chercher les meilleures structures et combiner leurs résultats.

Pour que ces approches à base de score soient réalisables en pratique, nous verrons que le score doit être décomposable localement, c'est-à-dire s'exprimer comme la somme de scores locaux au niveau de chaque nœud. Se pose aussi le problème de parcours de l'espace  $\mathbb{B}$  des réseaux bayésiens à la recherche de la meilleure structure. Comme une recherche exhaustive est impossible, les algorithmes proposés travaillent sur un espace réduit (espace des arbres, ordonnancement des nœuds), ou effectuent une recherche gloutonne dans cet espace.

#### 3.4.1 Les scores possibles

La plupart des scores existants dans la littérature appliquent le principe de parcimonie du rasoir d'Occam : trouver le modèle qui corresponde le mieux aux données  $\mathcal{D}$  mais qui soit le plus simple possible. Ainsi ces scores sont souvent décomposables en deux termes : la vraisemblance  $L(\mathcal{D} \mid \theta, \mathcal{B})$ , et un second terme qui tient compte de la complexité du modèle, à l'aide entre autres, du nombre de paramètres nécessaires pour représenter le réseau.

Soit  $X_i$  un nœud du réseau bayésien de taille  $r_i$ , et  $pa(X_i)$  ses parents. Le nombre de paramètres nécessaires pour décrire la distribution de probabilité  $P(X_i \mid pa(X_i) = x_j)$  est égal à  $r_i - 1$ . Pour représenter  $P(X_i \mid pa(X_i))$ , il faudra donc  $Dim(X_i, \mathcal{B})$  paramètres, avec :

$$Dim(X_i, \mathcal{B}) = (r_i - 1) \prod_{X_j \in pa(X_i)} r_j = (r_i - 1)q_i \quad (3.5)$$

Le nombre de paramètres nécessaires pour décrire toutes les distributions de probabilité du réseau  $\mathcal{B}$  est  $Dim(\mathcal{B})$  :

$$Dim(\mathcal{B}) = \sum_{i=1}^n Dim(X_i, \mathcal{B}) = \sum_{i=1}^n (r_i - 1)q_i \quad (3.6)$$

A partir de cela, différents scores ont été proposés :

**L'entropie conditionnelle** de la structure  $\mathcal{B}$  [15]

$$H(\mathcal{B}, \mathcal{D}) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} -\frac{N_{i,j,k}}{N} \log\left(\frac{N_{i,j,k}}{N_{i,j}}\right) \quad (3.7)$$

En partant de l'équation 2.3, il est possible de faire le lien entre l'entropie et le maximum de la log-vraisemblance :

$$\begin{aligned} \log L(\mathcal{D} \mid \theta, \mathcal{B}) &= \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{i,j,k} \log \theta_{i,j,k} \\ \log L(\mathcal{D} \mid \theta^{MV}, \mathcal{B}) &= \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{i,j,k} \log\left(\frac{N_{i,j,k}}{N_{i,j}}\right) \\ \log L(\mathcal{D} \mid \theta^{MV}, \mathcal{B}) &= -N \times H(\mathcal{B}, \mathcal{D}) \end{aligned} \quad (3.8)$$

La vraisemblance – ou l'entropie – n'impose aucun contrôle sur la complexité de la structure recherchée. Au contraire, pour un ensemble de données  $\mathcal{B}$  fixé, la structure la plus vraisemblable sera celle qui possède le plus de paramètres, c'est-à-dire la structure reliant toutes les variables [55].

**Les critères AIC** [3] **et BIC** [118] peuvent aussi s'appliquer aux réseaux bayésiens :

$$ScoreAIC(\mathcal{B}, \mathcal{D}) = \log L(\mathcal{D} \mid \theta^{MV}, \mathcal{B}) - Dim(\mathcal{B}) \quad (3.9)$$

$$ScoreBIC(\mathcal{B}, \mathcal{D}) = \log L(\mathcal{D} \mid \theta^{MV}, \mathcal{B}) - \frac{1}{2} Dim(\mathcal{B}) \log N \quad (3.10)$$

A la différence de la vraisemblance, ces deux équations 3.9 et 3.10 illustrent bien la volonté de rechercher un modèle capable de bien modéliser les données tout en restant simple.

### La longueur de description minimale

Il est aussi possible d'appliquer le principe de longueur de description minimale MDL [112]. Ce principe général affirme que le modèle représentant au mieux un ensemble de données est celui qui minimise la somme des deux termes suivants : (1) la longueur de codage du modèle et (2) la longueur de codage des données lorsque ce modèle est utilisé pour représenter ces données. Plusieurs travaux ont transposé ce principe aux réseaux bayésiens : Bouckaert [15], Lam et Bacchus [84] et Suzuki [127]. Nous ne citerons ici que l'approche de Lam et Bacchus [84] :

$$ScoreMDL(\mathcal{B}, \mathcal{D}) = \log L(\mathcal{D} \mid \theta^{MV}, \mathcal{B}) - |\mathcal{A}_{\mathcal{B}}| \log N - c \cdot Dim(\mathcal{B}) \quad (3.11)$$

où  $|\mathcal{A}_{\mathcal{B}}|$  est le nombre d'arcs dans le graphe  $\mathcal{B}$  et  $c$  est le nombre de bits utilisés pour stocker chaque paramètre numérique.

**Le score BD** (*bayesian Dirichlet*)

Cooper et Herskovits [35] proposent un score basé sur une approche bayésienne. En partant d'une loi a priori sur les structures possibles  $P(\mathcal{B})$ , le but est d'exprimer la probabilité a posteriori des structures possibles sachant que les données  $\mathcal{D}$  ont été observées  $P(\mathcal{B} | \mathcal{D})$ , ou plus simplement  $P(\mathcal{B}, \mathcal{D})$  :

$$\begin{aligned} \text{ScoreBD}(\mathcal{B}, \mathcal{D}) &= P(\mathcal{B}, \mathcal{D}) = \int_{\theta} L(\mathcal{D} | \theta, \mathcal{B}) P(\theta | \mathcal{B}) P(\mathcal{B}) d\theta \\ &= P(\mathcal{B}) \int_{\theta} L(\mathcal{D} | \theta, \mathcal{B}) P(\theta | \mathcal{B}) d\theta \end{aligned} \quad (3.12)$$

L'intégrale de l'équation 3.12 n'est pas toujours exprimable simplement. D'une manière générale, Chickering et Heckerman [27] montrent comment utiliser l'approximation de Laplace pour calculer cette intégrale (avec un échantillon de grande taille), et qu'une simplification de cette approximation mène au *ScoreBIC*.

Avec les hypothèses classiques d'indépendance des exemples, et en prenant une distribution a priori de Dirichlet sur les paramètres, il est néanmoins possible d'exprimer le *ScoreBD* facilement :

$$\text{ScoreBD}(\mathcal{B}, \mathcal{D}) = P(\mathcal{B}) \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij} + \alpha_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + \alpha_{ijk})}{\Gamma(\alpha_{ijk})} \quad (3.13)$$

où  $\Gamma$  est la fonction *Gamma*

**Le score BDe** (*Bayesian Dirichlet Equivalent*)

Ce critère suggéré par Heckerman [68] s'appuie sur la même formule que le score *bayesian Dirichlet* avec des propriétés supplémentaires intéressantes, comme la conservation du score pour des structures équivalentes (voir p. 53). Le score BDe utilise une distribution a priori sur les paramètres définie par :

$$\alpha_{ijk} = N' \times P(X_i = x_k, pa(X_i) = x_j | \mathcal{B}_c) \quad (3.14)$$

où  $\mathcal{B}_c$  est la structure a priori n'encodant aucune indépendance conditionnelle (graphe complètement connecté) et  $N'$  est un nombre d'exemples « équivalent » défini par l'utilisateur.

Dans le cas où la distribution de probabilité conditionnelle en  $X_i$  est uniforme, Heckerman et al. arrivent aux coefficients de Dirichlet de l'équation 3.15 qui correspondent à un a priori uniforme non informatif proposé tout d'abord par [17]. Le score BDe utilisant les  $\alpha_{ijk}$  décrits dans l'équation 3.15 est souvent appelé score BDeu.

$$\alpha_{ijk} = \frac{N'}{r_i q_i} \quad (3.15)$$

Heckerman et al. [68] prouvent aussi que le score BDe utilisant les a priori définis par l'équation 3.14 n'a plus besoin d'utiliser une distribution de Dirichlet comme loi a priori sur les paramètres.

**Le score  $BD\gamma$**  (*bayesian Dirichlet généralisé*)

Borgelt et Kruse [13] ont généralisé le score BD en introduisant un hyper-paramètre  $\gamma$  :

$$ScoreBD\gamma(\mathcal{B}, \mathcal{D}) = P(\mathcal{B}) \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\gamma N_{ij} + \alpha_{ij})}{\Gamma((\gamma + 1)N_{ij} + \alpha_{ij})} \dots \quad (3.16)$$

$$\dots \prod_{k=1}^{r_i} \frac{\Gamma((\gamma + 1)N_{ijk} + \alpha_{ijk})}{\Gamma(\gamma N_{ijk} + \alpha_{ijk})}$$

Borgelt et al. démontrent aussi que leur fonction de score permet de passer du score bayésien ( $\gamma = 0$ ) à l'entropie conditionnelle ( $\gamma \rightarrow +\infty$ ), contrôlant ainsi la tendance à sélectionner des structures simples.

### 3.4.2 Déterminer un a priori sur les structures

Certains scores ( $ScoreBD$ ,  $ScoreBDe$  et  $ScoreBD\gamma$ ) utilisent des métriques bayésiennes et nécessitent la détermination d'une loi de probabilité a priori sur les structures. Cette distribution de probabilités est soit uniforme (la solution la plus simple), soit calculable à partir de connaissances a priori fixées par un expert (en fixant une distribution de probabilité sur les arcs possibles ou une structure de référence).

- La loi uniforme est la distribution sur les structures la plus simple :

$$P(\mathcal{B}) = \text{constante}$$

- Il est également possible de décomposer la probabilité d'une structure comme produit des probabilités de chaque relation parent-nœud :

$$P(\mathcal{B}) = \prod_{i=1}^n P(pa_i^{\mathcal{B}} \rightarrow X_i)$$

où  $P(pa_i^{\mathcal{B}} \rightarrow X_i)$  est la probabilité que  $pa_i^{\mathcal{B}}$  soient les parents de  $X_i$ . Ces probabilités locales peuvent être fournies par exemple par un expert, comme le proposent Richardson et al. [110].

- Une autre façon de prendre en compte les connaissances expertes est de privilégier les structures proches du réseau a priori  $\mathcal{B}_e$  donné par un expert :

$$P(\mathcal{B}) \propto \kappa^\delta$$

où  $\delta$  est le nombre d'arcs différents entre  $\mathcal{B}$  et  $\mathcal{B}_e$  et  $\kappa$  un coefficient de pénalisation [68].

### 3.4.3 Pourquoi chercher la meilleure structure ?

Dans de nombreux domaines, la structure de score maximal est souvent beaucoup plus vraisemblable que les autres (voir [69, 59]). Par contre, il existe aussi des situations où plusieurs structures candidates sont à peu près aussi

vraisemblables. Dans ce cas, [59] proposent, toujours dans le cadre des approches bayésiennes, le principe de *model averaging*. Le but n'est pas d'interroger le meilleur modèle, mais de faire la moyenne sur tous les réseaux possibles.

Supposons par exemple que nous cherchions la probabilité de la variable  $X_A$  :

$$P(X_A | \mathcal{D}) = \sum_{\mathcal{B}} P(X_A | \mathcal{B}, \mathcal{D})P(\mathcal{B} | \mathcal{D}) \quad (3.17)$$

L'équation 3.1 montre que l'espace des réseaux bayésiens est super-exponentiel. Il n'est donc pas envisageable de calculer tous les termes de cette somme. L'approximation la plus courante est issue des méthodes MCMC [89] où quelques structures sont générées puis utilisées dans le calcul de 3.17. Une autre approche possible consiste à utiliser les méthodes de type *bootstrap* [58] pour produire différents ensembles de données qui serviront à obtenir plusieurs structures candidates, et à utiliser l'équation 3.17 avec ces structures.

### 3.4.4 Recherche dans l'espace des réseaux bayésiens

L'estimation du score d'un réseau bayésien peut mener à de nombreux calculs inutiles et rendre les méthodes d'apprentissage de structure inutilisables en pratique. La première précaution à prendre concerne l'utilisation d'un score décomposable localement pour ne pas recalculer complètement le score d'une nouvelle structure.

$$Score(\mathcal{B}, \mathcal{D}) = \text{constante} + \sum_{i=1}^n score(X_i, pa_i) \quad (3.18)$$

Il est facile de montrer que les scores évoqués précédemment, ou leur logarithme pour  $Score_{BD}$  et  $Score_{BDe}$ , sont des scores décomposables. Par la suite, nous noterons  $Score(\cdot)$  le score global et  $score(\cdot)$  le score local en chaque nœud.

Cette décomposition locale du score permet une évaluation rapide de la variation du score entre deux structures en fonction d'un nombre réduit de scores locaux liés aux différences entre ces deux structures. Il reste maintenant à parcourir l'espace  $\mathbb{B}$  des réseaux bayésiens pour trouver la structure qui possède le meilleur score. Nous avons vu en 3.2.3 qu'une recherche exhaustive n'est pas concevable. Plusieurs heuristiques permettent de remédier à ce problème, soit en réduisant l'espace de recherche à un sous-espace particulier (l'espace des arbres), soit en ordonnant les nœuds pour ne chercher les parents d'un nœud que parmi les nœuds suivants, soit en effectuant une heuristique de parcours de l'espace  $\mathbb{B}$  de type recherche gloutonne.

### Restriction à l'espace des arbres

Cette méthode utilise une notion classique en recherche opérationnelle, l'arbre de recouvrement maximal (Maximum Weight Spanning Tree) : l'arbre qui passe par tous les nœuds et maximise un score défini pour tous les arcs possibles.

Chow et Liu [33] ont proposé d'utiliser un score basé sur un critère d'information mutuelle :

$$\begin{aligned} W_{CL}(X_A, X_B) &= \sum_{x_a, x_b} P(X_A = x_a, X_B = x_b) \log \frac{P(X_A = x_a, X_B = x_b)}{P(X_A = x_a)P(X_B = x_b)} \\ &= \sum_{a,b} \frac{N_{ab}}{N} \log \frac{N_{ab}N}{N_a \cdot N_b} \end{aligned} \quad (3.19)$$

Heckerman [68] suggère d'utiliser un score quelconque, localement décomposable, en définissant le poids d'une arête par :

$$W(X_A, X_B) = score(X_A, X_B) - score(X_A, \emptyset) \quad (3.20)$$

où  $score(X_A, X_B)$  est le score local en  $X_A$  en supposant que  $X_B$  est son parent, et  $score(X_A, \emptyset)$  est le score local en  $X_A$  en supposant qu'il ne possède aucun parent.

Parmi toutes les heuristiques qui permettent de construire l'arbre optimal à partir des poids des arêtes, nous utiliserons l'algorithme de Kruskal (voir par exemple [117, 37, 2]). Celui-ci part d'un ensemble de  $n$  arbres d'un seul nœud (un par variable) et les fusionne en fonction du poids des arêtes (voir algorithme 3.6).

L'arbre de recouvrement maximal est un arbre non orienté reliant toutes les variables. Notons que cet arbre non orienté est le représentant de la classe d'équivalence de Markov de tous les arbres dirigés possédant ce même squelette. En effet, par définition, un arbre orienté ne peut pas contenir de V-structure, donc tous les arbres de même squelette sont équivalents au sens de Markov (cf. section 3.2.3).

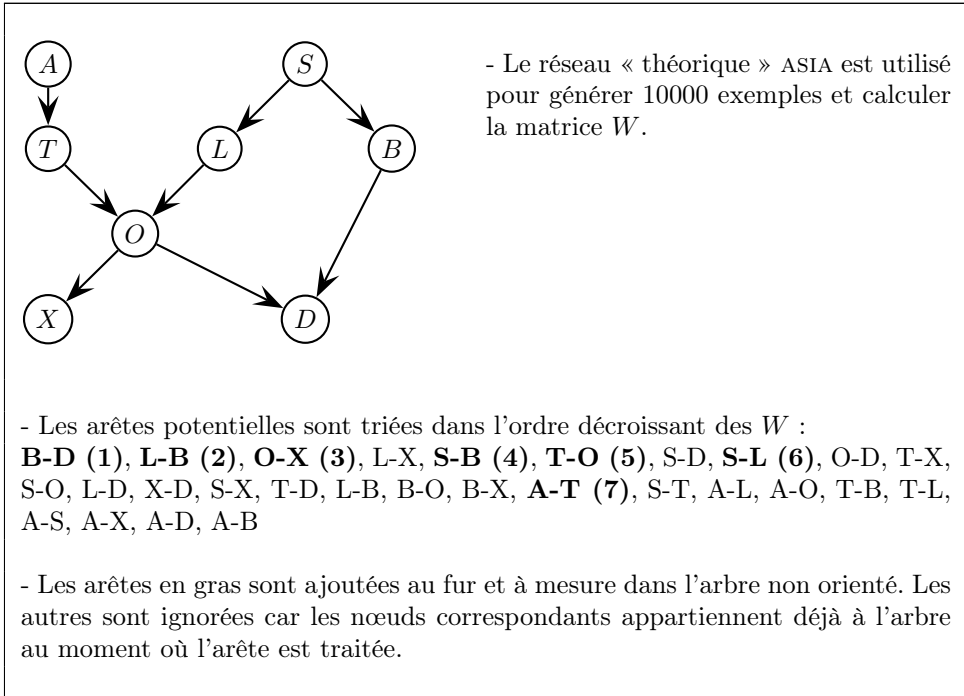
L'orientation de cet arbre non orienté pourrait donc être réalisée grâce à l'algorithme 3.3, ou plus simplement, en choisissant arbitrairement un nœud racine et en dirigeant chaque arête à partir de ce nœud. Pour cela, il suffit d'effectuer un parcours en profondeur de l'arbre en mémorisant le père de chaque nœud, puis de se servir de cette information pour orienter les arêtes.

Nous appellerons algorithme *MWST* « dirigé », l'algorithme de construction d'un arbre orienté qui utilise l'algorithme de Kruskal pour obtenir l'arbre de recouvrement optimal non orienté, puis qui oriente les arêtes à partir d'un nœud racine arbitraire.

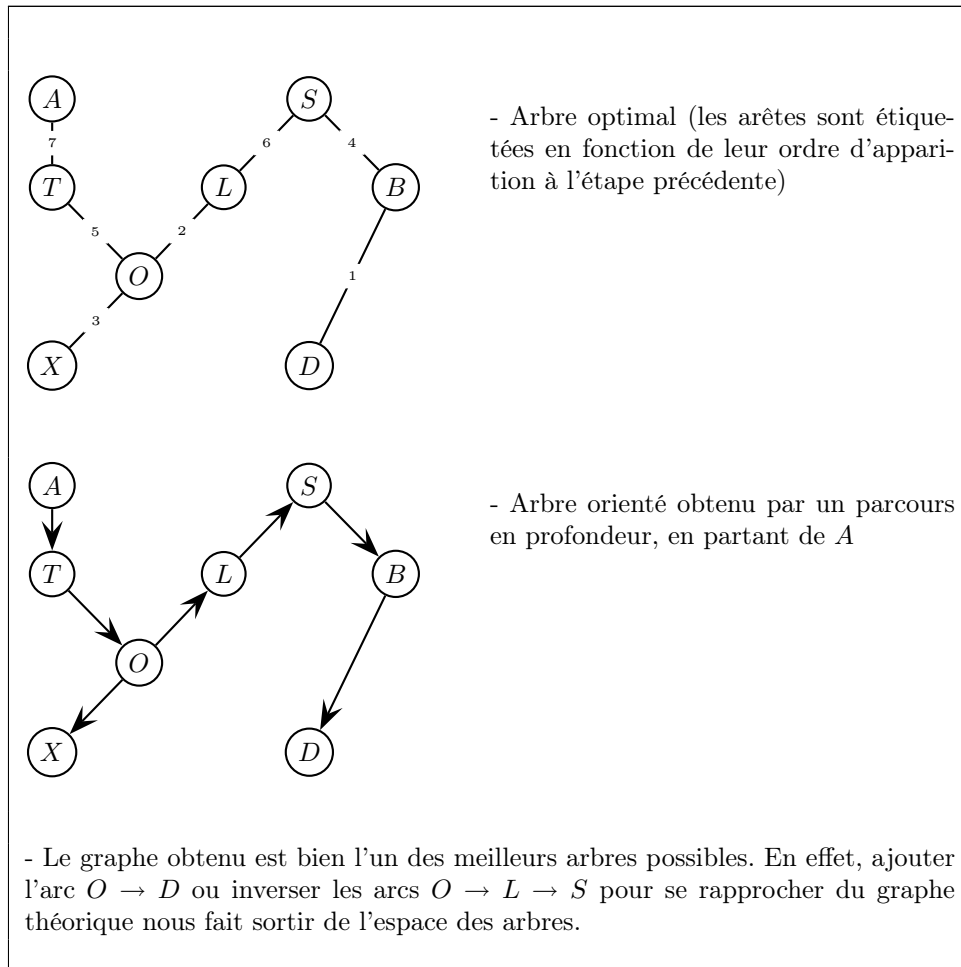
L'exemple 3.7 illustre certains avantages et inconvénients de cet algorithme. Il permet d'obtenir rapidement un arbre orienté très proche de la structure d'origine. De plus, par définition de l'arbre de recouvrement, aucun nœud ne sera écarté de la structure, ce qui permet de retrouver des liens difficiles à apprendre, comme le lien  $A \leftarrow T$  de l'exemple, qui n'a pas un poids  $W$  très fort et qui est le dernier lien ajouté. Cette propriété peut cependant devenir gênante puisqu'elle forcera des variables à appartenir au graphe même si elles ne sont pas utiles au problème.

<p>Algorithme MWST « dirigé »</p> <ul style="list-style-type: none"> <li>• Construction de l'arbre optimal (Kruskal) <ul style="list-style-type: none"> <li><math>\forall X_i, \mathcal{T}(X_i) = \{X_i\}</math></li> <li><math>\mathcal{B} \leftarrow \emptyset</math></li> <li><math>\forall (X_i, X_j) \in \mathcal{A}</math> <ul style="list-style-type: none"> <li>si <math>\mathcal{T}(X_i) \neq \mathcal{T}(X_j)</math> alors <ul style="list-style-type: none"> <li>• <math>\mathcal{B} \leftarrow \mathcal{B} \cup (X_i, X_j)</math></li> <li>• <math>\mathcal{T}' \leftarrow \mathcal{T}(X_i) \cup \mathcal{T}(X_j)</math></li> <li>• <math>\mathcal{T}(X_i) \leftarrow \mathcal{T}'</math></li> <li>• <math>\mathcal{T}(X_j) \leftarrow \mathcal{T}'</math></li> </ul> </li> </ul> </li> </ul> </li> <li>• Orientation des arêtes <ul style="list-style-type: none"> <li><math>\mathcal{B} \leftarrow \emptyset</math></li> <li><math>\{pa_i\} \leftarrow \text{ParcoursProfondeur}(\mathcal{B}, X_r)</math></li> <li><math>\forall X_i,</math> <ul style="list-style-type: none"> <li>si <math>pa_i \neq \emptyset</math> alors ajout de <math>pa_i \rightarrow X_i</math> dans <math>\mathcal{B}</math></li> </ul> </li> </ul> </li> </ul>													
<p>Notations :</p> <table border="0"> <tr> <td><math>\mathcal{A}</math></td> <td>liste des arêtes <math>(X_i, X_j)</math> dans l'ordre décroissant des <math>W</math></td> </tr> <tr> <td><math>\mathcal{T}(X_i)</math></td> <td>arbre passant par le nœud <math>X_i</math></td> </tr> <tr> <td><math>X_r</math></td> <td>racine choisie pour orienter l'arbre</td> </tr> <tr> <td><math>pa_i</math></td> <td>parent du nœud <math>X_i</math></td> </tr> <tr> <td><math>\mathcal{B}</math></td> <td>arbre optimal non orienté</td> </tr> <tr> <td><math>\mathcal{B}</math></td> <td>structure finale obtenue par l'algorithme</td> </tr> </table>		$\mathcal{A}$	liste des arêtes $(X_i, X_j)$ dans l'ordre décroissant des $W$	$\mathcal{T}(X_i)$	arbre passant par le nœud $X_i$	$X_r$	racine choisie pour orienter l'arbre	$pa_i$	parent du nœud $X_i$	$\mathcal{B}$	arbre optimal non orienté	$\mathcal{B}$	structure finale obtenue par l'algorithme
$\mathcal{A}$	liste des arêtes $(X_i, X_j)$ dans l'ordre décroissant des $W$												
$\mathcal{T}(X_i)$	arbre passant par le nœud $X_i$												
$X_r$	racine choisie pour orienter l'arbre												
$pa_i$	parent du nœud $X_i$												
$\mathcal{B}$	arbre optimal non orienté												
$\mathcal{B}$	structure finale obtenue par l'algorithme												

TAB. 3.6 – Algorithme MWST « dirigé »



TAB. 3.7: Exécution de l'algorithme MWST « dirigé » (à suivre ...)



TAB. 3.7: Exécution de l'algorithme MWST « dirigé »

### Ordonnement des nœuds

Un autre moyen de limiter l'espace de recherche consiste à rester dans l'espace des réseaux bayésiens, tout en ajoutant un ordre sur les nœuds pour se limiter dans la recherche des arcs intéressants : si  $X_i$  est avant  $X_j$  alors il ne pourra y avoir d'arc de  $X_j$  vers  $X_i$ . Cette hypothèse forte réduit le nombre de structures possibles de  $NS(n)$  (eq.3.1) à  $NS'(n) = 2^{n(n-1)/2}$ . Par exemple,  $NS'(5) = 1024$  contre  $NS(5) = 29281$  et  $NS'(10) = 3.5 \times 10^{13}$  contre  $NS(10) = 4.2 \times 10^{18}$ .

Pour rendre cette idée exploitable, il faut diminuer encore l'espace de recherche en ajoutant des heuristiques supplémentaires. Ainsi l'algorithme K2 de Cooper et Herskovits [35] détaillé dans la table 3.8 reprend le score *bayesian Dirichlet* (eq. 3.13) avec un a priori uniforme sur les structures. Ce score peut s'écrire de la façon suivante :

$$ScoreBD(\mathcal{B}, \mathcal{D}) \propto \prod_{i=1}^n g(i, pa_i)$$

<p>Algorithme K2</p> <p>Pour <math>i = 1</math> à <math>n</math>  <math>pa_i \leftarrow \emptyset</math>  <math>g_{old} \leftarrow g(i, pa_i)</math>  <math>OK \leftarrow vrai</math>  R�p�ter  <ul style="list-style-type: none"> <li>• Chercher <math>X_j \in Pred(X_i) \setminus pa_i</math> qui maximise <math>g(i, pa_i \cup \{X_j\})</math></li> <li>• <math>g_{new} \leftarrow g(i, pa_i \cup \{X_j\})</math></li> <li>• Si <math>g_{new} &gt; g_{old}</math> alors  <math>g_{old} \leftarrow g_{new}</math>  <math>pa_i \leftarrow pa_i \cup \{X_j\}</math>  sinon <math>OK \leftarrow faux</math></li> </ul> Tant Que <math>OK</math> et <math> pa_i  &lt; u</math></p>	
Notations :	<p><math>Pred()</math> relation d'ordre sur les n�uds <math>X_i</math></p> <p><math>u</math> borne sup. du nombre de parents possibles pour un n�ud</p> <p><math>pa_i</math> ensemble des parents du n�ud <math>X_i</math></p> <p><math>g(i, pa_i)</math> score local d�fini dans l'�quation (3.21)</p>

TAB. 3.8 – Algorithme K2

avec

$$g(i, pa_i) = \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij} + \alpha_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + \alpha_{ijk})}{\Gamma(\alpha_{ijk})} \quad (3.21)$$

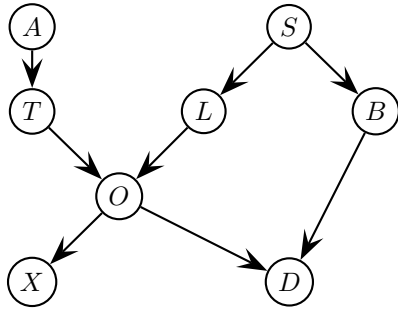
Pour maximiser *ScoreBD*, Cooper et Herskovits proposent d'effectuer une recherche gloutonne en cherchant les parents  $pa_i$  du n ud  $X_i$  qui vont maximiser  $g(i, pa_i)$ , et ainsi de suite, sans remettre en cause les choix effectu s pr c demment. Ils sugg rent aussi de fixer une borne sup rieure  $u$  au nombre de parents possibles pour un n ud.

L'algorithme K3 pr sent  par Bouckaert [15] reprend le principe de l'algorithme K2 en rempla ant le score *bayesian Dirichlet* par un score MDL. L'algorithme BENEDICT d velopp  par Acid et de Campos [1] emprunte le m me proc d  en utilisant comme score l'information mutuelle conditionnelle.

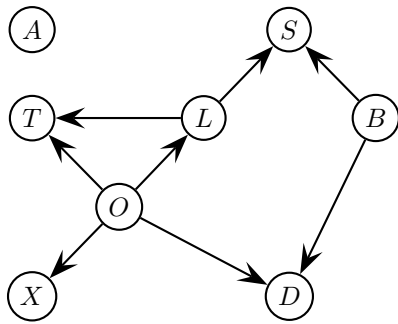
L'inconv nient principal de ces m thodes r side dans la d termination de l'ordre des n uds. Ceci est illustr  dans l'exemple 3.9 : en utilisant l'ordre topologique du r seau recherch , l'algorithme parvient   retrouver la structure recherch e (a). Par contre, dans deux situations plus r alistes (b) et (c), l'algorithme donne des structures de qualit  variable. Dans l'exemple (b), l'ordonnement des n uds emp che de retrouver la V-structure  $T \rightarrow O \leftarrow L$  et g n re   la place la meilleure structure entre les trois n uds, compte tenu des contraintes fix es.

Pour tenter de r soudre ce probl me d'initialisation, les travaux de [72] utilisent une approche de type algorithmes g n tiques pour trouver l'ordonnement optimal des n uds et ainsi la meilleure structure gr ce   l'algorithme K2.

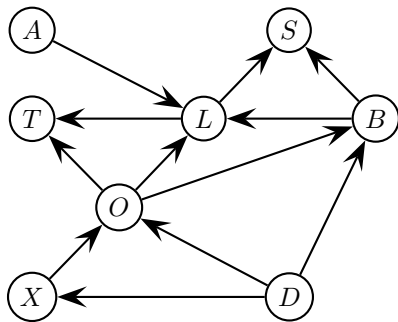
Reprenons les 1000 exemples générés pour le problème ASIA, et utilisons l'algorithme K2 à partir de trois initialisations différentes.



(a) Graphe obtenu avec un ordonnancement des nœuds *biaisé* (*ASTLBOXD*, ordre topologique du graphe ASIA)



(b) Graphe obtenu avec un ordonnancement des nœuds aléatoire (*OLBXASDT*)



(c) Graphe obtenu avec un autre ordonnancement des nœuds aléatoire (*TALDSXOB*)

Commentaires : l'initialisation de l'algorithme K2 est problématique. Deux initialisations différentes (b) et (c) mènent à des résultats de qualité variable.

TAB. 3.9: Exécution de l'algorithme K2

Opérateur	$INSERT(X_A, X_B)$	$DELETE(X_A, X_B)$	$REVERSE(X_A, X_B)$
Variation du score	$s(X_B, Pa_{X_B}^{+X_A})$ $-s(X_B, Pa_{X_B})$	$s(X_B, Pa_{X_B}^{-X_A})$ $-s(X_B, Pa_{X_B})$	$s(X_B, Pa_{X_B}^{-X_A})$ $-s(X_B, Pa_{X_B})$ $+s(X_A, Pa_{X_A}^{+X_B})$ $-s(X_A, Pa_{X_A})$

Notations :  $Pa_{X_i}^{-X_j} = Pa(X_i) \setminus \{X_j\}$   $Pa_{X_i}^{+X_j} = Pa(X_i) \cup \{X_j\}$

TAB. 3.10 – Exemple d’opérateurs dans l’espace des réseaux bayésiens et calcul de la variation du score pour chacun des opérateurs

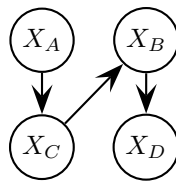
### Recherche gloutonne dans $\mathbb{B}$

Vue la taille super-exponentielle de l’espace des réseaux bayésiens, une autre solution logique est d’utiliser des méthodes d’optimisation simples pour parcourir cet espace moins brutalement que les méthodes de type K2, sans toutefois parcourir tout l’espace.

Les principales différences entre les méthodes proposées résident dans la façon de parcourir l’espace, c’est-à-dire dans le choix des opérateurs générant le voisinage d’un graphe, et l’utilisation d’heuristiques supplémentaires pour simplifier le voisinage obtenu.

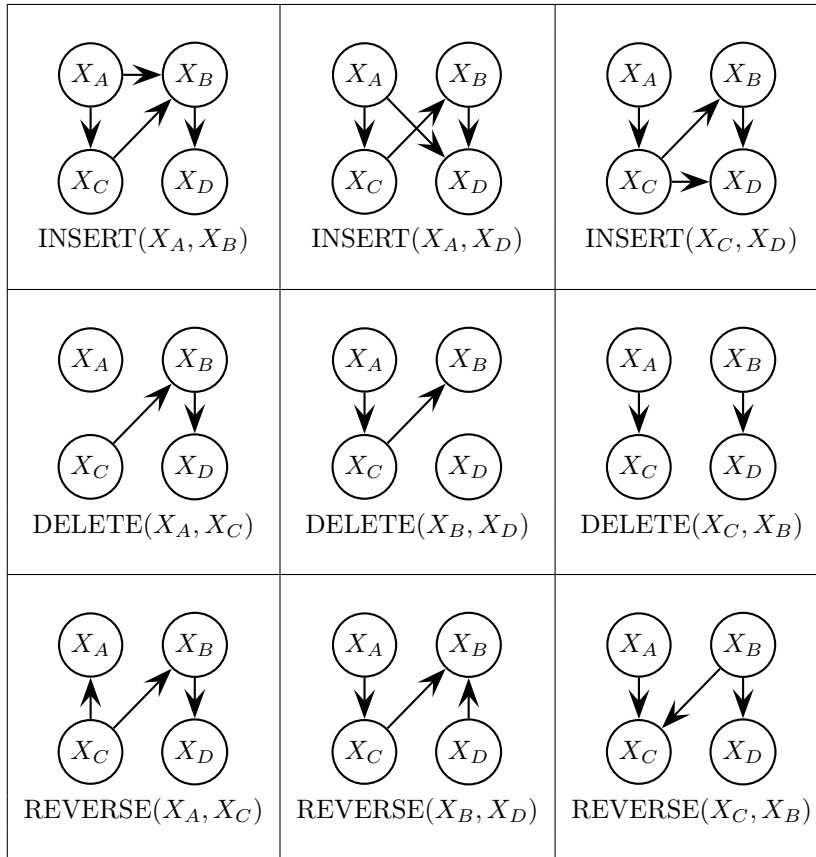
Chickering et al. [26] utilisent l’algorithme classique de recherche gloutonne (*Greedy Search*) dans l’espace des réseaux bayésiens décrit dans la table 3.12. La notion de voisinage utilisée, définie à l’aide de trois opérateurs : ajout, suppression ou inversion d’arc, est illustrée dans l’exemple 3.11. L’utilisation d’un score décomposable localement nous permet de calculer rapidement la variation du score pour les structures obtenues avec ces trois opérateurs (voir table 3.10).

Considérons le graphe  $\mathcal{B}$  suivant, ainsi qu’un voisinage défini par les trois opérateurs *ajout* ( $INSERT$ ), *suppression* ( $DELETE$ ) et *retournement* ( $REVERSE$ ) d’arc. Remarquons que les graphes résultants ne sont retenus que s’ils sont sans circuit.



TAB. 3.11: Exemple de voisinage GS (à suivre ...)

- Génération du voisinage de  $\mathcal{B}$  :



Notons que pour cet exemple de petite taille, le voisinage comprend déjà neuf DAG dont il faut à présent évaluer la qualité. Pour des structures plus complexes, la taille du voisinage devient beaucoup plus importante, ce qui rend nécessaire l'utilisation de scores locaux pour limiter les calculs, et l'implémentation d'un système de cache pour ne pas recalculer plusieurs fois chaque score local.

TAB. 3.11: Exemple de voisinage GS

<p>Algorithme Recherche Gloutonne</p> <ul style="list-style-type: none"> <li>• Initialisation du graphe <math>\mathcal{B}</math> (<i>Graphe vide, aléatoire, donné par un expert ou arbre obtenu par MWST</i>)</li> <li>• <math>Continuer \leftarrow Vrai</math></li> <li>• <math>Score_{max} \leftarrow score(\mathcal{B})</math></li> <li>• Répéter <ul style="list-style-type: none"> <li>• Génération de <math>V_{\mathcal{B}}</math>, voisinage de <math>\mathcal{B}</math>, à l'aide d'opérateurs : <ul style="list-style-type: none"> <li>- Ajout d'arc, suppression d'arc, inversion d'arc (les graphes ainsi obtenus doivent être acycliques)</li> </ul> </li> <li>• Calcul du score pour chaque graphe de <math>V_{\mathcal{B}}</math></li> <li>• <math>\mathcal{B}_{new} \leftarrow \operatorname{argmax}_{\mathcal{B}' \in V_{\mathcal{B}}} (score(\mathcal{B}'))</math></li> <li>• Si <math>score(\mathcal{B}_{new}) \geq Score_{max}</math> alors <ul style="list-style-type: none"> <li><math>Score_{max} \leftarrow score(\mathcal{B}_{new})</math></li> <li><math>\mathcal{B} \leftarrow \mathcal{B}_{new}</math></li> </ul> </li> </ul> </li> <li>sinon <math>Continuer \leftarrow Faux</math></li> </ul> <p>Tant Que <math>Continuer</math></p>	
Notations :	
$Score()$	fonction de score sur les structures possibles
$V_{\mathcal{B}}$	ensemble des DAG voisins du DAG $\mathcal{B}$ courant
$\mathcal{B}$	structure finale obtenue par l'algorithme

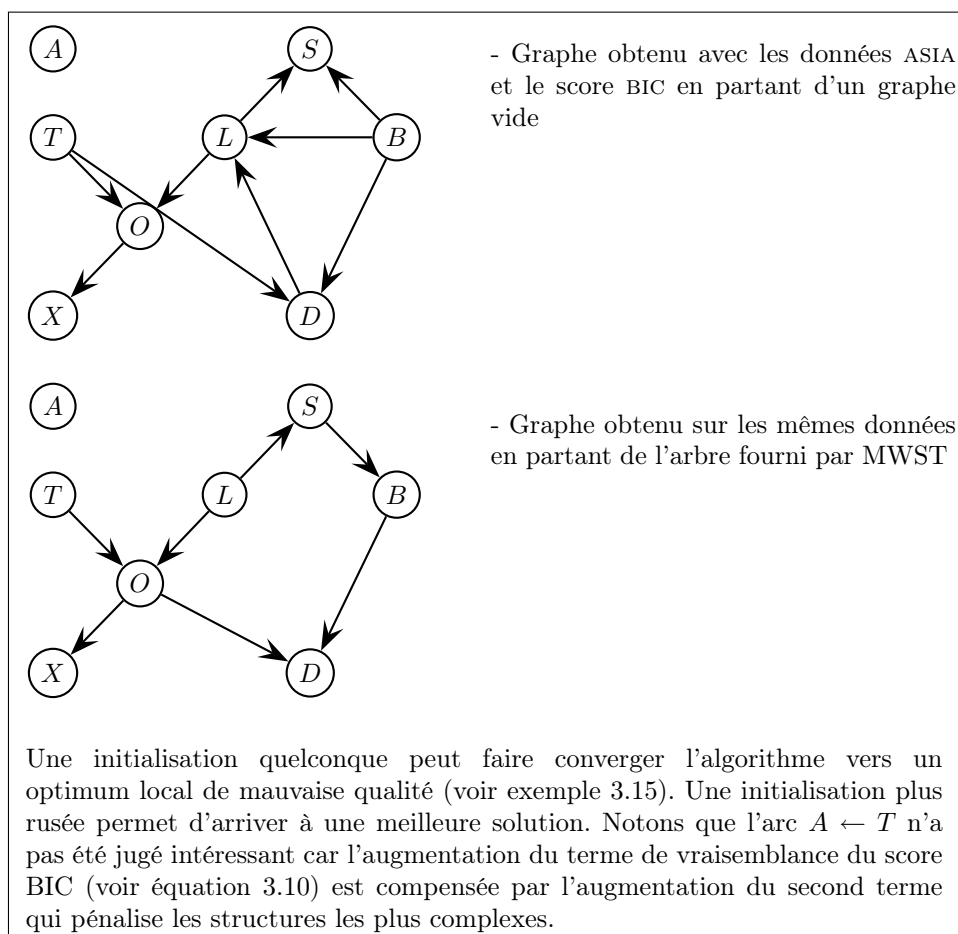
TAB. 3.12 – Algorithme Recherche Gloutonne (GS)

L'algorithme de recherche gloutonne est connu pour converger vers un optimum local et souvent de mauvaise qualité (voir exemple 3.15). Une façon simple d'éviter de tomber dans cet optimum local est de répéter plusieurs fois la recherche gloutonne à partir d'initialisations tirées aléatoirement. Cette méthode connue sous le nom de *iterated hill climbing* ou *random restart* permet de découvrir plusieurs optima, et a donc plus de chances de converger vers la solution optimale si la fonction de score n'est pas trop « bruitée ».

Dans le même esprit, d'autres techniques d'optimisation peuvent être utilisées, comme par exemple le recuit simulé (*Simulated Annealing*) [80]. Citons aussi les travaux de Larrañaga et al. [87] qui se servent d'algorithmes génétiques pour parcourir l'espace des DAG.

Jouffe et Munteanu ([77], [78]) proposent une autre série d'opérateurs pour éviter de tomber dans des minima locaux facilement reconnaissables (voir exemple p.54), ainsi qu'une méthode de parcours encore plus simple dans l'espace des ordonnancements possibles, en utilisant ensuite l'algorithme K2 pour calculer la meilleure structure possible pour chaque ordonnancement.

Les méthodes itératives comme la recherche gloutonne souffrent souvent de problèmes d'initialisation. Il est parfois possible d'utiliser des connaissances expertes pour définir un graphe de départ. Dans le cas contraire, sur une idée de [70], nous avons utilisé dans [153] l'arbre obtenu par l'algorithme MWST décrit précédemment, ce qui permet souvent d'arriver à une meilleure solution qu'avec une initialisation aléatoire (ou « vide »), ou à la même solution mais en moins



TAB. 3.13 – Résultat de l'algorithme GS avec le score BIC

d'itérations. L'exemple 3.13 nous montre l'intérêt d'une initialisation « rusée » : en partant d'un graphe vide, l'algorithme converge vers une solution moyenne alors qu'une initialisation à l'aide de l'arbre optimal aboutit à une solution plus proche de la réalité. Il faut remarquer ici un des inconvénients des méthodes à base de score : les dépendances faibles entre variables ( $A \leftarrow T$  dans l'exemple) ne sont pas jugées intéressantes car l'augmentation du terme de vraisemblance du score est compensée par l'augmentation du second terme qui pénalise les structures les plus complexes.

### 3.4.5 Algorithmes basés sur un score et données incomplètes

Le premier problème à résoudre, lorsque les données sont incomplètes, concerne le calcul de la vraisemblance ou plus généralement du score pour une structure fixée, puis sa maximisation.

Concernant la maximisation de cette vraisemblance, nous avons déjà évoqué en 2.2 comment le principe de l'algorithme EM pouvait être utilisé pour estimer

<p>Algorithme EM structurel générique</p> <ul style="list-style-type: none"> <li>• Initialiser <math>i \leftarrow 0</math></li> <li>• Initialisation du graphe <math>\mathcal{G}^0</math> (<i>Graphe vide, aléatoire, donné par un expert ou arbre obtenu par MWST-EM</i>)</li> <li>• Initialisation des paramètres <math>\Theta^0</math></li> <li>• Répéter <ul style="list-style-type: none"> <li>• <math>i \leftarrow i + 1</math></li> <li>• <math>(\mathcal{B}^i, \Theta^i) = \operatorname{argmax}_{\mathcal{B}, \Theta} Q(\mathcal{B}, \Theta : \mathcal{B}^{i-1}, \Theta^{i-1})</math></li> </ul> </li> </ul> <p>Tant Que <math> Q(\mathcal{B}^i, \Theta^i : \mathcal{B}^{i-1}, \Theta^{i-1}) - Q(\mathcal{B}^{i-1}, \Theta^{i-1} : \mathcal{B}^{i-1}, \Theta^{i-1})  &gt; \epsilon</math></p>	
Notations :	<p><math>Q(\mathcal{B}, \Theta : \mathcal{B}^*, \Theta^*)</math>   Espérance de la vraisemblance d'un réseau bayésien &lt; <math>\mathcal{B}, \Theta</math> &gt; calculée à partir de la distribution de probabilité des données manquantes <math>P(\mathcal{D}_m   \mathcal{B}^*, \Theta^*)</math></p>

TAB. 3.14 – Algorithme EM structurel générique

les paramètres  $\theta$  d'une structure  $\mathcal{B}$  fixée. Ce même principe s'applique aussi naturellement à la recherche conjointe de  $\theta$  et  $\mathcal{B}$  pour donner ce que Friedman a d'abord appelé *EM pour la sélection de modèle* [56] puis *EM structurel*[57]. L'algorithme 3.14 présente très sommairement l'application de l'algorithme EM à l'apprentissage de structure.

L'étape de maximisation dans l'espace des paramètres de l'algorithme EM paramétrique (cf. p.16) est maintenant remplacée par une maximisation dans l'espace  $\{\mathcal{B}, \Theta\}$ . Cela revient, à chaque itération, à chercher la meilleure structure et les meilleurs paramètres associés à cette structure. En pratique, ces deux étapes sont clairement distinctes<sup>1</sup> :

$$\mathcal{B}^i = \operatorname{argmax}_{\mathcal{B}} Q(\mathcal{B}, \bullet : \mathcal{B}^{i-1}, \Theta^{i-1}) \quad (3.22)$$

$$\Theta^i = \operatorname{argmax}_{\Theta} Q(\mathcal{B}^i, \Theta : \mathcal{B}^{i-1}, \Theta^{i-1}) \quad (3.23)$$

où  $Q(\mathcal{B}, \Theta : \mathcal{B}^*, \Theta^*)$  est l'espérance de la vraisemblance d'un réseau bayésien <  $\mathcal{B}, \Theta$  > calculée à partir de la distribution de probabilité des données manquantes  $P(\mathcal{D}_m | \mathcal{B}^*, \Theta^*)$ .

Il faut noter que la recherche dans l'espace des graphes (équation 3.22) nous ramène au problème initial, c'est à dire, trouver le maximum de la fonction de score dans tout l'espace des DAG. Heureusement, grâce aux travaux de Dempster (*Generalised EM*), il est possible de remplacer cette étape de recherche de l'optimum global de la fonction  $Q$  par la recherche d'une meilleure solution augmentant le score, sans affecter les propriétés de convergence de l'algorithme. Cette recherche "d'une meilleure solution" (au lieu de "la meilleure") peut alors s'effectuer dans un espace plus limité, comme par exemple  $\mathcal{V}_{\mathcal{B}}$ , l'ensemble des voisins du graphe  $\mathcal{B}$  comme défini pour une recherche gloutonne classique.

Concernant la recherche dans l'espace des paramètres (équation 3.23), [56] suggère de répéter l'opération plusieurs fois, en utilisant une initialisation intelligente. Cela revient alors à exécuter l'algorithme EM paramétrique pour chaque structure  $\mathcal{B}^i$  à partir de la structure  $\mathcal{B}^0$ .

<sup>1</sup>la notation  $Q(\mathcal{B}, \bullet : \dots)$  utilisée dans l'équation 3.22 correspond à  $E_{\Theta}[Q(\mathcal{B}, \Theta : \dots)]$  pour un score bayésien ou à  $Q(\mathcal{B}, \Theta^{MV} : \dots)$  où  $\Theta^{MV}$  est obtenu par maximum de vraisemblance

La fonction  $Q$  à maximiser est très liée à la notion de score dans le cas des données complètes, puisqu'il s'agit de l'espérance de cette fonction de score en utilisant une densité de probabilité sur les données manquantes fixée  $P(\mathcal{D}_m \mid \mathcal{B}^*, \Theta^*)$ . Dans ses deux articles concernant les algorithmes EM structurels, Friedman adapte respectivement le score BIC et le score BDe pour les données manquantes. Décrivons ici le cas du score BIC :

$$Q^{BIC}(\mathcal{B}, \Theta : \mathcal{B}^*, \Theta^*) = E_{\mathcal{B}^*, \Theta^*} [\log P(\mathcal{D}_o, \mathcal{D}_m \mid \mathcal{B}, \Theta)] - \frac{1}{2} \text{Dim}(\mathcal{B}) \log N \quad (3.24)$$

Comme le score BIC,  $Q^{BIC}$  est lui aussi décomposable :

$$Q^{BIC}(\mathcal{B}, \Theta : \mathcal{B}^*, \Theta^*) = \sum_i Q^{bic}(X_i, P_i, \Theta_{X_i|P_i} : \mathcal{B}^*, \Theta^*) \quad (3.25)$$

où

$$Q^{bic}(X_i, P_i, \Theta_{X_i|P_i} : \mathcal{B}^*, \Theta^*) = \sum_{X_i=x_k} \sum_{P_i=pa_j} N_{ijk}^* \log \theta_{ijk} - \frac{\log N}{2} \text{Dim}(X_i, \mathcal{B}) \quad (3.26)$$

avec  $N_{ijk}^* = E_{\mathcal{B}^*, \Theta^*} [N_{ijk}] = N * P(X_i = x_k, P_i = pa_j \mid \mathcal{B}^*, \Theta^*)$  obtenu par inférence dans le réseau  $\{\mathcal{B}^*, \Theta^*\}$  si  $\{X_i, P_i\}$  ne sont pas complètement mesurés, ou calculé classiquement sinon.

Les deux algorithmes EM structurels présentés par Friedman peuvent ainsi être assimilés à des algorithmes de recherche gloutonne (avec un score BIC ou BDe), avec un apprentissage EM paramétrique à chaque itération.

A partir de ces considérations, et de nos travaux concernant l'initialisation des algorithmes de recherche gloutonne par l'arbre optimal reliant toutes les variables (MWST), nous avons proposé dans [161] une adaptation de MWST aux bases de données incomplètes (MWST-EM) pouvant aussi être utilisée comme initialisation des algorithmes EM structurels classiques.

L'algorithme MWST-EM est ainsi une instanciation de l'algorithme EM structurel générique (cf. algo. 3.14) où la maximisation sur  $\mathcal{B}$  (équation 3.22) ne s'effectue plus dans tout l'espace des DAG mais seulement dans l'espace des arbres. Cette simplification permet d'éviter de limiter la recherche dans le voisinage du graphe courant, comme doivent le faire les algorithmes EM structurels précédents, puisqu'il est possible de trouver directement le meilleur arbre maximisant une fonction  $Q$  fixée.

### 3.5 Recherche dans l'espace des classes d'équivalence de Markov

Certaines méthodes décrites précédemment ne travaillent pas réellement dans l'espace  $\mathbb{B}$  des réseaux bayésiens. Par exemple, des algorithmes tels que PC, IC ou BN-PC permettent d'obtenir le CPDAG représentant de la classe d'équivalence qu'il faut ensuite finir d'orienter. De même, l'algorithme MWST nous donne une structure non orientée qui est aussi le représentant de la classe d'équivalence de tous les arbres orientés possédant le même squelette. L'orientation finale de ces graphes peut mener à des DAG orientés différemment, mais impossibles à distinguer d'après les données.

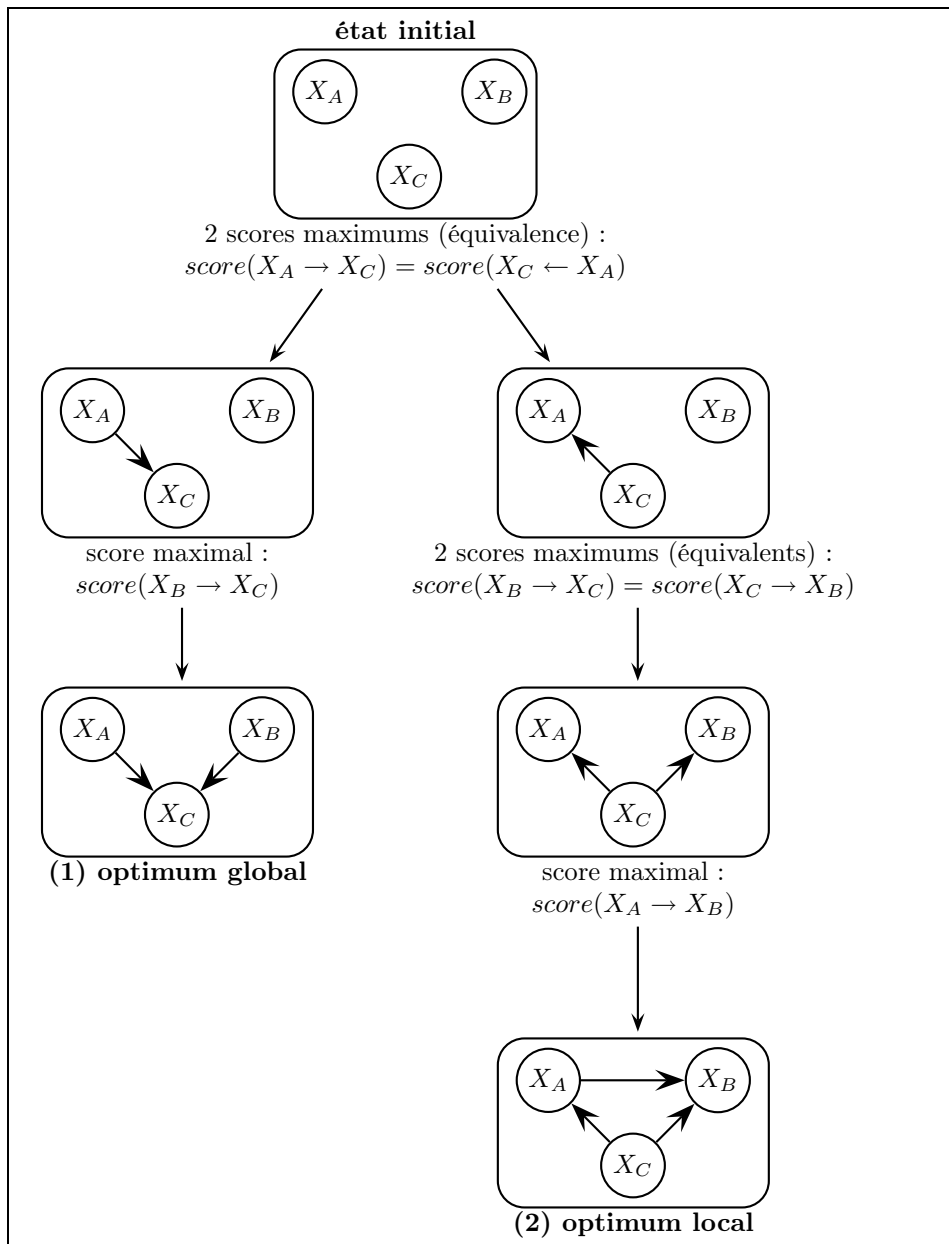
Chickering [28] a montré que des réseaux bayésiens équivalents obtiennent le même score, pour la plupart des scores (AIC, BIC, BDe, MDL). L'utilisation de ces scores dans l'espace  $\mathbb{B}$  des réseaux bayésiens aboutit alors à des découvertes de structures non globalement optimales [92]. La table 3.15 nous montre l'exemple d'une recherche gloutonne (par ajout d'arcs) qui cherche à retrouver une V-structure initiale dans l'espace  $\mathbb{B}$  des réseaux bayésiens à trois variables. Les scores classiques conservant les équivalences, l'algorithme peut se retrouver soit dans la situation 1 (découverte d'une structure optimale, c'est-à-dire la structure initiale) soit dans la situation 2 (découverte d'une structure optimale localement).

Pour éviter ce genre de problème sans utiliser de techniques d'optimisation complexes comme le recuit simulé ou les algorithmes génétiques, certaines méthodes proposent de travailler directement dans l'espace  $\mathbb{E}$  des classes d'équivalence, ou de tenir compte des propriétés d'équivalence pour mieux parcourir l'espace  $\mathbb{B}$ .

L'espace  $\mathbb{E}$  est quasiment de même taille que l'espace  $\mathbb{B}$  des réseaux bayésiens. Gillispie et Perlman [64] ont montré que le nombre moyen de DAG par classe d'équivalence semblait converger vers une valeur asymptotique proche de 3.7 (en observant ce résultat jusqu'à  $n = 10$  variables). Deux choix s'offrent donc à nous : soit travailler directement dans l'espace  $\mathbb{B}$ , en tenant compte des propriétés de  $\mathbb{E}$  en ajoutant des heuristiques pour éviter de tomber dans des minima locaux (Munteanu et al. [92]), ou en bridant les opérateurs de voisinage (Castelo et al. [19]) ; soit travailler directement dans l'espace  $\mathbb{E}$ . Ainsi Chickering [28, 29] propose une série d'opérateurs dans l'espace des PDAG (insérer une arête, supprimer une arête, insérer un arc, supprimer un arc, inverser un arc, créer une V-structure). Malheureusement, ces opérateurs sont trop lourds et l'algorithme nécessite de nombreuses opérations entre l'espace des CPDAG, des PDAG intermédiaires et l'espace des DAG. Bendou et Munteanu [11] ont recours au même ensemble d'opérateurs, mais en travaillant directement dans un espace intermédiaire, l'espace des graphes chaînés maximaux.

Concernant la multitude d'opérateurs à utiliser lors de la recherche gloutonne, une avancée significative est apportée grâce à la conjecture de Meek [91] démontrée dans [31]. Chickering montre qu'il suffit d'effectuer une recherche gloutonne en ajoutant des arcs puis une autre recherche gloutonne en en supprimant pour arriver à la structure optimale.

Cet algorithme GES (Greedy Equivalence Search) se sert uniquement de deux opérateurs d'insertion et de suppression proposés dans [8], [31], [30] et [32]. La table 3.16 nous décrit les opérateurs INSERT et DELETE ainsi que



TAB. 3.15 – Exemple de découverte d’une structure de réseau bayésien non globalement optimale par une méthode d’ajout d’arcs dans l’espace  $\mathbb{B}$  des réseaux bayésiens [92] : au lieu de retrouver la V-structure initiale (1), l’algorithme pourra converger vers un optimum local (2)

leur condition de validité et le calcul de la variation du score qu’ils entraînent.

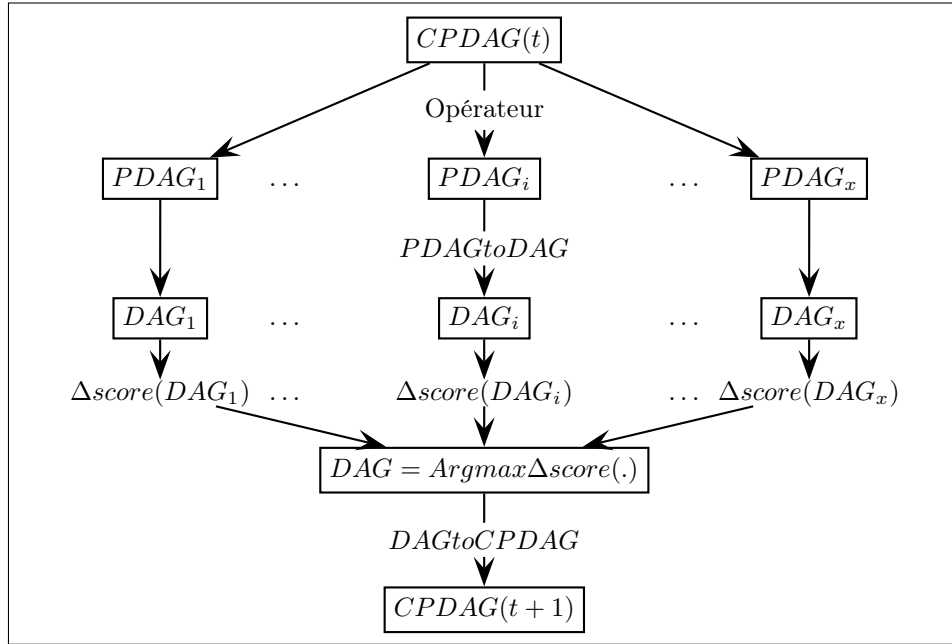
Ces deux opérateurs servent à construire les limites d’inclusion inférieure  $V^-(\mathcal{E})$  et supérieure  $V^+(\mathcal{E})$  du CPDAG courant  $\mathcal{E}$ .

Soit  $\mathcal{E}$  un CPDAG, la limite d’inclusion supérieure  $V^+(\mathcal{E})$  est l’ensemble des

Opérateur	$INSERT(X_A, X_B, T)$	$DELETE(X_A, X_B, H)$
Conditions de validité	<ul style="list-style-type: none"> <li>• <math>NA_{X_B, X_A} \cup T</math> est une clique</li> <li>• chaque chemin semi-dirigé <math>X_B \dots X_A</math> contient un nœud dans <math>NA_{X_B, X_A} \cup T</math></li> </ul>	<ul style="list-style-type: none"> <li>• <math>NA_{X_B, X_A} \setminus H</math> est une clique</li> </ul>
Variation du score	$s(X_B, NA_{X_B, X_A} \cup T \cup Pa_{X_B}^{+X_A})$ $-s(X_B, NA_{X_B, X_A} \cup T \cup Pa_{X_B})$	$s(X_B, \{NA_{X_B, X_A} \setminus T\} \cup Pa_{X_B}^{-X_A})$ $-s(X_B, \{NA_{X_B, X_A} \setminus T\} \cup Pa_{X_B})$
Effet	$X_A - X_B$ devient $X_A \rightarrow X_B$ $\forall X_t \in T,$ $X_t - X_B$ devient $X_t \rightarrow X_B$	$X_A - X_B$ devient $X_A - X_B$ $\forall X_h \in H,$ $X_B - X_h$ devient $X_B \rightarrow X_h$ $X_A - X_h$ devient $X_A \rightarrow X_h$

Notations :  $Pa_{X_i}^{-X_j} = Pa(X_i) \setminus \{X_j\}$        $Pa_{X_i}^{+X_j} = Pa(X_i) \cup \{X_j\}$   
 $NA_{X_B, X_A} = \{X_t / (X_t \rightarrow X_A \text{ ou } X_t \leftarrow X_A) \text{ et } X_t - X_B\}$

TAB. 3.16 – Exemple d’opérateurs dans l’espace des classes d’équivalence de Markov, condition de validité et calcul de la variation du score pour chacun des opérateurs



TAB. 3.17 – Algorithme GES, exemple d’itération dans l’espace  $\mathbb{E}$  des CPDAG

CPDAG voisins de  $\mathcal{E}$  définis par :

$$\mathcal{E}^+ \in V^+(\mathcal{E}) \text{ ssi } \exists \mathcal{G} \equiv \mathcal{E} / \{\mathcal{G}^+ = \{\mathcal{G} + 1 \text{ arc}\} \text{ et } \mathcal{G}^+ \equiv \mathcal{E}^+\}$$

Soit  $\mathcal{E}$  un CPDAG, la limite d’inclusion inférieure  $V^-(\mathcal{E})$  est l’ensemble des CPDAG voisins de  $\mathcal{E}$  définis par :

$$\mathcal{E}^- \in V^-(\mathcal{E}) \text{ ssi } \exists \mathcal{G} \equiv \mathcal{E} / \{\mathcal{G}^- = \{\mathcal{G} - 1 \text{ arc}\} \text{ et } \mathcal{G}^- \equiv \mathcal{E}^-\}$$

Algorithme Greedy Equivalence Search (insertion d'arc)

- $\mathcal{G} \leftarrow \mathcal{G}_0$
- $Score \leftarrow -\infty$
- Répéter
  - $Score_{max} \leftarrow -\infty$
  - $\forall (X_A, X_B) \in \mathcal{X}^2 / X_A \text{ non adjacent à } X_B$
  - $NNA_{X_B, X_A} = \{X_t / X_t \text{ non adjacent à } X_A \text{ et } X_t - X_B\}$
  - $NA_{X_B, X_A} = \{X_t / (X_t \rightarrow X_A \text{ ou } X_t \leftarrow X_A) \text{ et } X_t - X_B\}$
  - $\forall T \in \text{powerset}(NNA_{X_B, X_A})$
  - $\mathcal{G}_{new} \leftarrow \mathcal{G}$
  - $Test_1 \leftarrow NNA_{X_B, X_A} \cup T \text{ est une clique}$
  - $Test_2 \leftarrow \exists X_B \overset{\text{part.}}{\rightsquigarrow} X_A \text{ dans } \mathcal{G} \setminus (NA_{X_B, X_A} \cup T)$
  - Si  $Test_1$  et  $\neg Test_2$  alors
    - $\mathcal{G}_{new} \leftarrow \mathcal{G} + INSERT(X_A, X_B, T)$ , c'est-à-dire :
      - $X_A - X_B$  devient  $X_A \rightarrow X_B$  dans  $\mathcal{G}_{new}$
      - $\forall X_t \in T, X_t - X_B$  devient  $X_t \rightarrow X_B$  dans  $\mathcal{G}_{new}$
    - $DAG_{new} \leftarrow CPDAGtoDAG(\mathcal{G}_{new})$
    - $Score_{new} \leftarrow score(DAG_{new})$
    - Si  $Score_{new} > Score_{max}$  alors
      - $DAG_{max} = DAG_{new}$
      - $Score_{max} = Score_{new}$
  - $Score_{old} \leftarrow Score$
  - $Score \leftarrow Score_{max}$
  - Si  $Score \geq Score_{old}$  alors  $G \leftarrow DAGtoCPAG(DAG_{max})$

Tant Que  $Score \geq Score_{old}$

TAB. 3.18 – Algorithme GES (insertion d'arcs)

La première étape de cet algorithme, détaillée dans la table 3.18, est donc une recherche gloutonne dans la limite d'inclusion supérieure, afin de complexifier la structure tant que le score augmente. L'étape suivante (table 3.19) est une recherche gloutonne dans la limite d'inclusion inférieure, pour simplifier la structure « maximale » obtenue et converger vers la structure optimale. L'exemple 3.20 illustre cette recherche pour quatre nœuds, en donnant les CPDAG générés à chaque étape.

L'algorithme *Greedy Equivalence Search* ne s'affranchit pas totalement de l'espace  $\mathbb{B}$  des DAG. En effet, les fonctions de score existantes ne travaillent que dans cet espace. Il faut donc y revenir à chaque itération pour calculer le score d'un des DAG de la classe d'équivalence (voir la table 3.17).

*Greedy Equivalence Search* tire avantageusement parti des propriétés de l'espace  $\mathbb{E}$  pour converger vers la structure optimale. Il ouvre aussi des perspectives intéressantes qui devraient rapidement voir le jour : pourquoi ne pas adapter GES aux données incomplètes avec le même principe que l'algorithme EM structurel travaillant dans  $\mathbb{B}$ , pour obtenir un EM structurel dans l'espace  $\mathbb{E}$  ?

Algorithme Greedy Equivalence Search (Suppression d'arc)

$Score \leftarrow Score_{old}$

Répéter

$Score_{max} \leftarrow -\infty$

$\forall (X_A, X_B) \in \mathcal{X}^2 / X_A$  adjacent à  $X_B$

$NA_{X_B, X_A} = \{X_t / (X_t \rightarrow X_A \text{ ou } X_t \leftarrow X_A) \text{ et } X_t - X_B\}$

$\forall H \in \text{powerset}(NA_{X_B, X_A})$

$\mathcal{G}_{new} \leftarrow \mathcal{G}$

Si  $NA_{X_B, X_A} \setminus H$  est une clique alors

$\mathcal{G}_{new} \leftarrow \mathcal{G} + DELETE(X_A, X_B, H)$ , c'est-à-dire :

$X_A - X_B$  (ou  $X_A \rightarrow X_B$ ) devient  $X_A X_B$  dans  $\mathcal{G}_{new}$

$\forall X_h \in H$ ,

$X_B - X_h$  devient  $X_B \rightarrow X_h$  dans  $\mathcal{G}_{new}$

$X_A - X_h$  (s'il existe) devient  $X_A \rightarrow X_h$  dans  $\mathcal{G}_{new}$

$DAG_{new} \leftarrow CPDAGtoDAG(\mathcal{G}_{new})$

$Score_{new} \leftarrow score(DAG_{new})$

Si  $Score_{new} > Score_{max}$  alors

$DAG_{max} = DAG_{new}$

$Score_{max} = Score_{new}$

$Score_{old} \leftarrow Score$

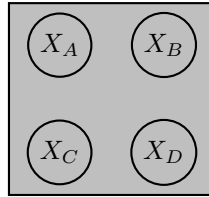
$Score \leftarrow Score_{max}$

Si  $Score \geq Score_{old}$  alors  $G \leftarrow DAGtoCPAG(DAG_{max})$

Tant Que  $Score \geq Score_{old}$

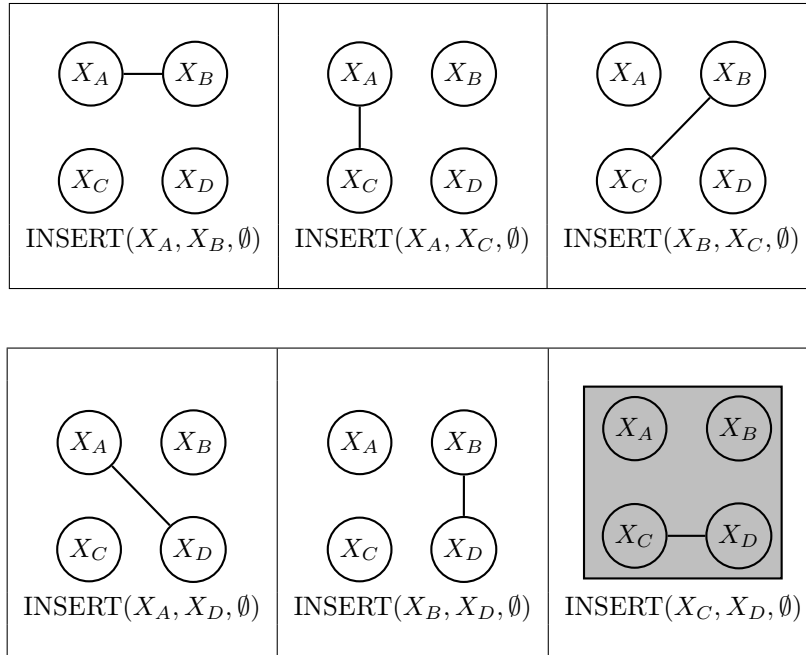
TAB. 3.19 – Algorithme GES (suppression d'arcs)

Soient quatre nœuds  $X_A, X_B, X_C$  et  $X_D$ . L'opérateur INSERT de l'algorithme GES nous donne la limite d'inclusion supérieure du graphe courant. Cette série de PDAG est transformée en DAG grâce à l'algorithme de Dor et Tarsi (voir table 3.3) pour pouvoir appliquer la fonction de score, puis en CPDAG grâce à l'algorithme de Chickering (voir table 3.2).

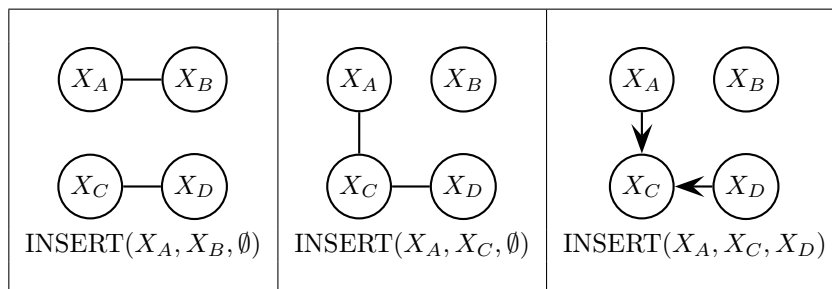


TAB. 3.20: Exécution de l'algorithme GES pour quatre nœuds (à suivre...)

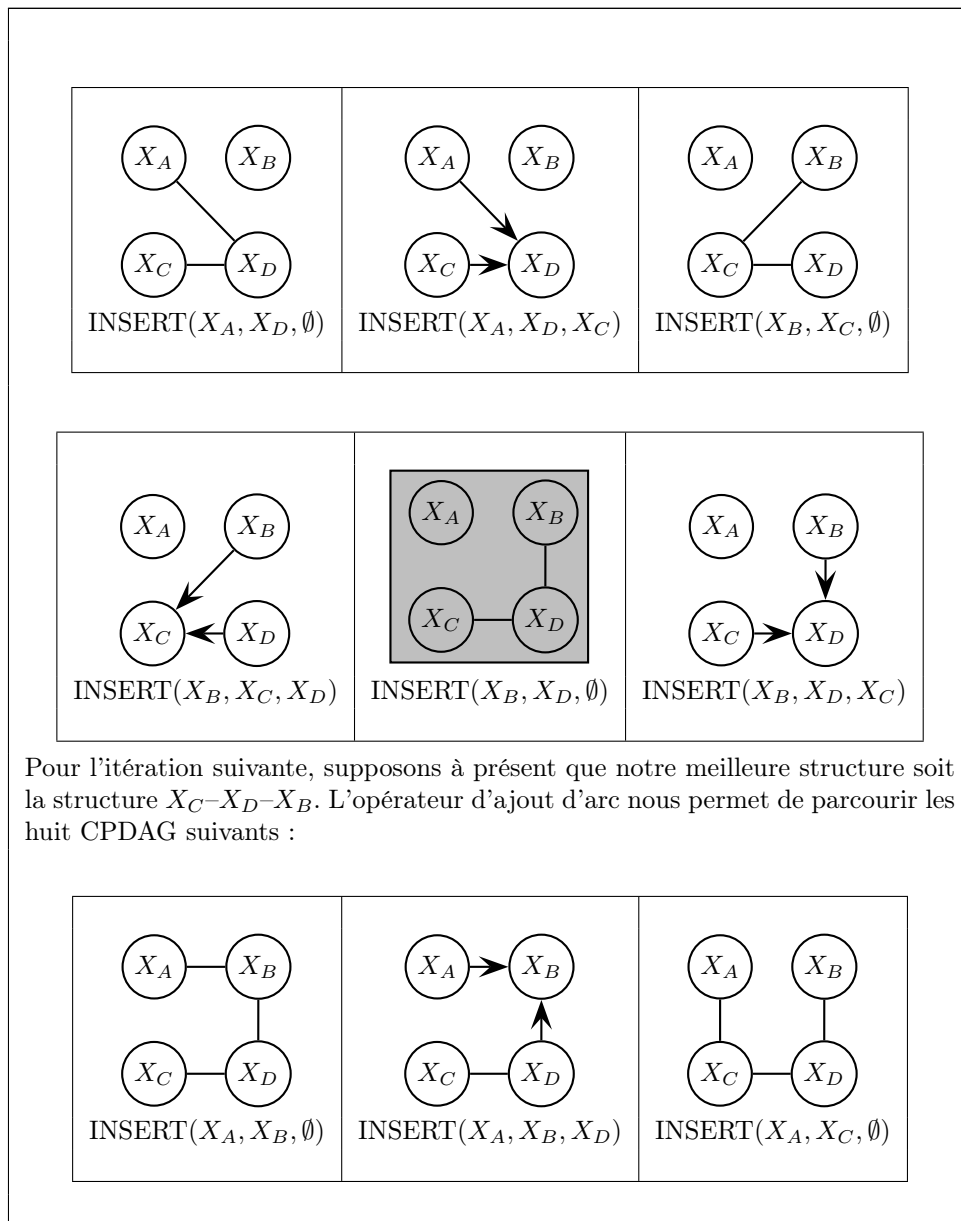
La première itération de l'algorithme GES revient à tester les six CPDAG suivants, qui sont effectivement les représentants des classes d'équivalence des douze DAG qui possèdent un unique arc.



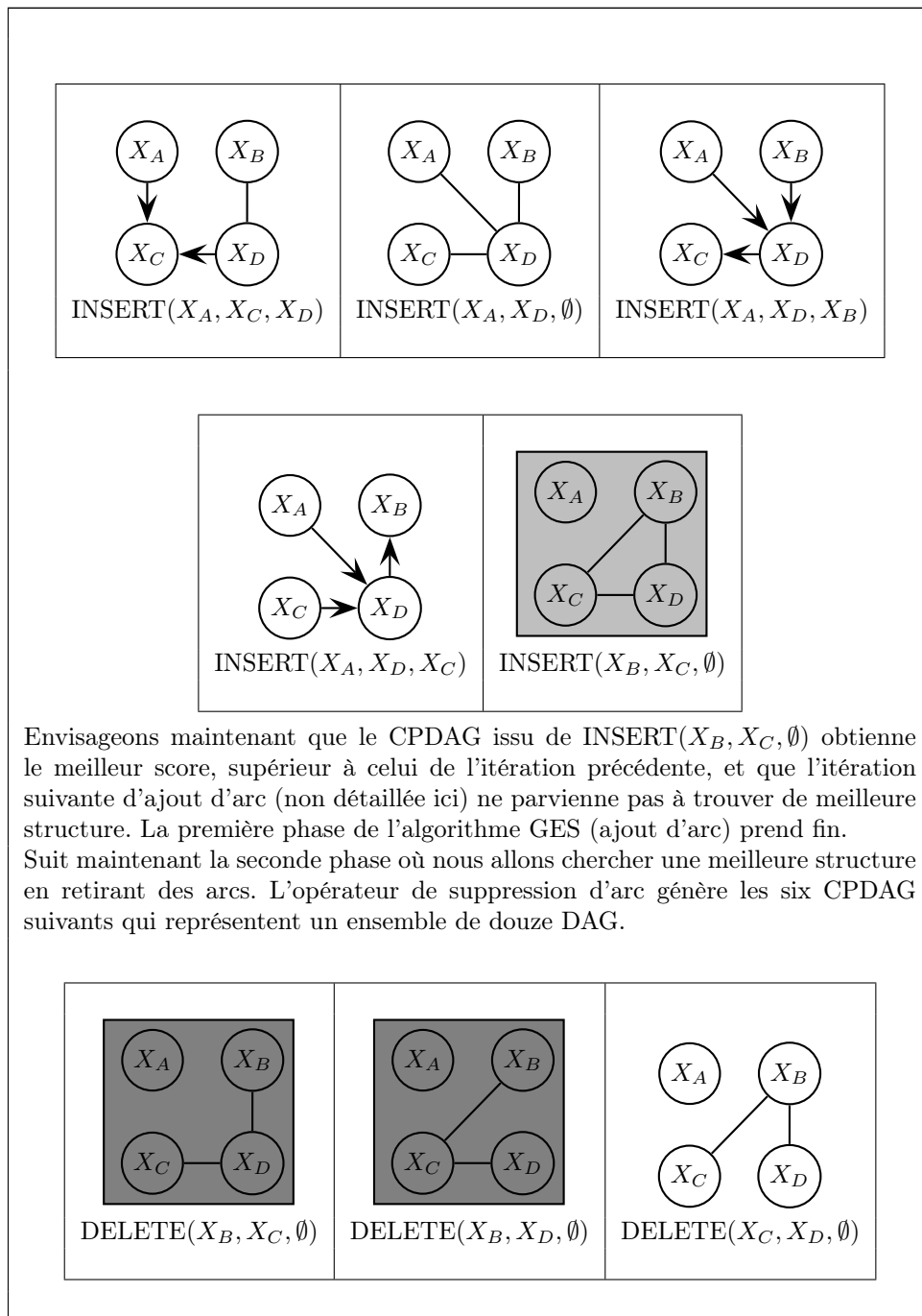
Supposons que le score obtenu par le CPDAG  $X_C-X_D$  soit le meilleur. GES va appliquer une nouvelle fois l'opérateur d'insertion pour obtenir neuf autres CPDAG. Ces graphes correspondent aux classes d'équivalence possibles pour les vingt DAGs à deux arcs obtenus après insertion d'un arc sur chacun des DAG équivalents au CPDAG précédent  $X_C-X_D$  :



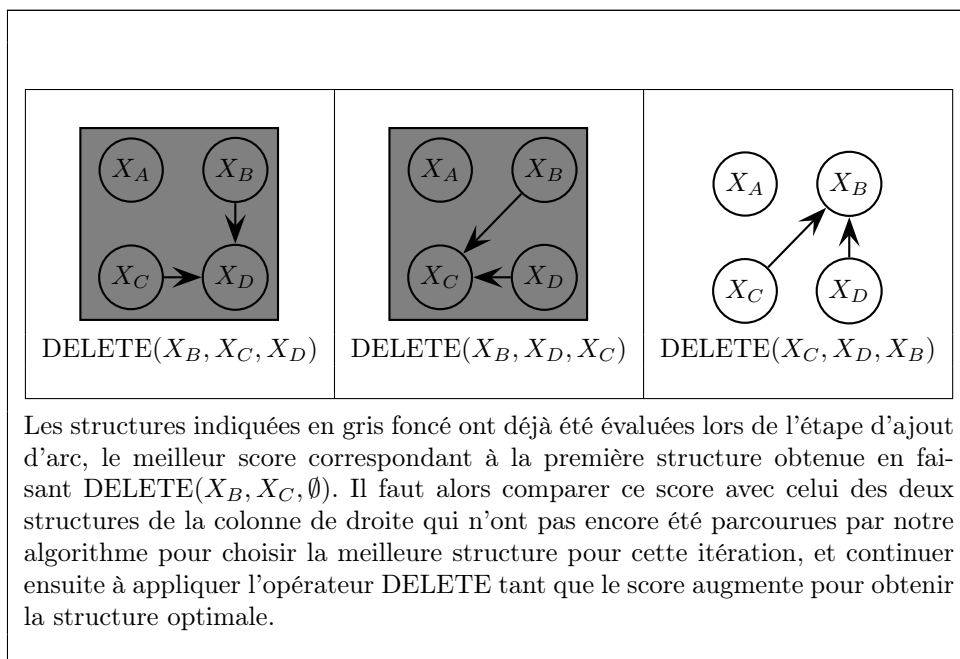
TAB. 3.20: Exécution de l'algorithme GES pour quatre nœuds (à suivre...)



TAB. 3.20: Exécution de l'algorithme GES pour quatre nœuds (à suivre...)



TAB. 3.20: Exécution de l'algorithme GES pour quatre nœuds (à suivre...)



TAB. 3.20: Exécution de l'algorithme GES pour quatre nœuds

### 3.6 Méthodes hybrides

Afin de tirer parti des avantages respectifs des algorithmes de recherche d'indépendances conditionnelles et de ceux basés sur l'utilisation d'un score, de nombreux travaux ont mené à des méthodes hybrides.

Ainsi, plusieurs approches emploient les informations issues d'une première phase de recherche d'indépendances conditionnelles pour guider la phase suivante, une recherche dans l'espace des DAG. Singh et Valtorta [120] ou Lamma et al. [85] génèrent, grâce à cette recherche d'indépendances conditionnelles, un ordonnancement des variables qui sert de point de départ à l'algorithme K2. Wong et al. [136] ont recours au même genre d'information pour contraindre une heuristique de parcours de l'espace des DAG par algorithmes génétiques.

D'autres approches, symétriques aux précédentes, vont profiter des avantages des méthodes à base de score pour aider les algorithmes d'apprentissage de structure par recherche d'indépendance conditionnelle. Dash et Druzdzel [41] partent du fait que l'algorithme PC est sensible, tout d'abord, aux heuristiques employées pour ne pas parcourir tous les ensembles de conditionnement, et ensuite au seuil du test statistique utilisé. Ils suggèrent alors un parcours aléatoire de l'espace de ces deux paramètres (ordre permettant de limiter les ensembles de conditionnement ainsi que le niveau de signification du test) en se servant d'un score bayésien pour comparer les réseaux obtenus. Sur le même principe général, Dash et Druzdzel [42] présentent un nouveau test d'indépendance conditionnelle *Hybrid Independence Test* se servant de certains avantages des approches à base de score comme l'ajout possible d'a priori et le recours à l'algorithme EM pour prendre en compte les données incomplètes.

## 3.7 Incorporation de connaissances

Nous avons pour l'instant décrit les différentes familles de méthodes d'apprentissage de structure à partir de données. Ces méthodes n'intègrent aucune connaissance précise sur la tâche à résoudre ou aucune connaissance d'experts sur la structure à trouver.

Si l'expert fournit directement la structure du réseau bayésien, le problème est résolu. Par contre, dans la plupart des cas, les connaissances de l'expert sur la structure ne sont que partielles. Cheng et al. [25] ont listé ces connaissances a priori :

1. Déclaration d'un nœud racine, c'est-à-dire sans parent,
2. Déclaration d'un nœud feuille, c'est-à-dire sans enfant,
3. Existence (ou absence) d'un arc entre deux nœuds précis,
4. Indépendance de deux nœuds conditionnellement à certains autres,
5. Déclaration d'un ordre (partiel ou complet) sur les variables.

A cette liste, nous ajouterons les points suivants :

6. Déclaration d'un nœud « cible » : essentiellement pour des tâches de classification,
7. Existence d'une variable latente entre deux nœuds.

Quel que soit le type de connaissance apportée par l'expert, il est souvent nécessaire d'utiliser des données pour trouver la structure du réseau bayésien. Les a priori de type 1. à 5. peuvent être facilement pris en compte par les algorithmes d'apprentissage de structure évoqués en 3.3 et 3.4. Nous allons donc approfondir les points 6. et 7. : **l'apprentissage de structure dans le cadre de la classification**, et **l'apprentissage de structure lorsque des variables latentes sont définies explicitement**.

### 3.7.1 Structures de réseaux bayésiens pour la classification

Dans les tâches de classification, une variable précise correspond à *la classe* qu'il faut « reconnaître » à partir des autres variables (les *caractéristiques*). Plusieurs méthodes d'apprentissage vont donc proposer des structures où ce nœud classe aura un rôle central ([55], [23], [24]).

#### Structure de Bayes naïve

Le classifieur de Bayes naïf correspond à la structure la plus simple qui soit, en posant l'hypothèse que les caractéristiques  $X_1 \dots X_{n-1}$  sont indépendantes conditionnellement à la classe  $X_c$ . Cela mène à la structure type de la figure 3.1. Cette structure, pourtant très simple, donne de très bons résultats dans de nombreuses applications [86].

#### Structure augmentée

Afin d'alléger l'hypothèse d'indépendance conditionnelle des caractéristiques, il a été suggéré « d'augmenter » la structure naïve en ajoutant des liens entre certaines caractéristiques ([79], [55], [116]).

Parmi les différentes méthodes avancées pour augmenter le réseau bayésien naïf, citons TANB (Tree Augmented Naive Bayes) qui utilise une structure

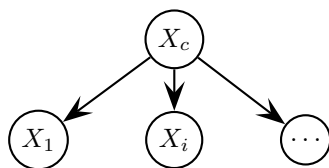


FIG. 3.1 – Réseau bayésien naïf

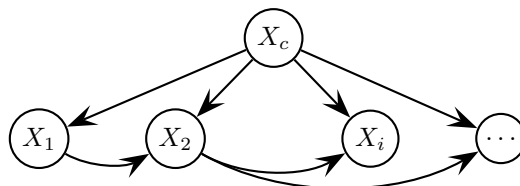


FIG. 3.2 – Réseau bayésien naïf augmenté (par un arbre)

naïve entre la classe et les caractéristiques et un arbre reliant toutes les caractéristiques. [61] a montré que la structure augmentée – par un arbre – optimale s’obtenait facilement en utilisant MWST sur les caractéristiques, et en reliant la classe aux caractéristiques comme pour une structure naïve. La seule différence réside dans le calcul de l’intérêt de connecter deux nœuds, où il faut remplacer l’information conditionnelle (équation 3.19) ou la différence de score (équation 3.20) utilisées par une information mutuelle ou une différence de score conditionnelle à la variable classe.

[55] et [65] ont établi que l’utilisation de telles structures donne de meilleurs résultats qu’une approche de recherche de structure brute à base de score (c’est-à-dire ne tenant pas compte de la spécificité du nœud classe).

Plusieurs extensions de TANB ont été étudiées récemment. L’arbre obtenu par TANB va obligatoirement relier chaque variable caractéristique avec une autre de ces variables. Pour assouplir cette hypothèse, [116] propose avec l’algorithme FANB (*Forest Augmented Naive Bayes*) de ne pas rechercher le meilleur arbre, mais la meilleure forêt, i.e. l’ensemble optimal d’arbres disjoints sur l’ensemble des variables caractéristiques. Pour cela, il utilise les spécificités de l’algorithme de recherche de l’arbre de recouvrement maximal proposé par Kruskal (voir par exemple [117, 37, 2]) pour trouver ces ensembles d’arbres disjoints.

D’autres extensions adaptent les méthodes au cas des bases de données incomplètes. Citons par exemple [40] qui abordent l’apprentissage de ces structures augmentées lorsque la variable classe est partiellement observée. L’algorithme MWST-EM proposé par [161] et évoqué p.50 peut aussi être appliqué pour trouver une structure de type TANB ou FANB, avec l’avantage supplémentaire de pouvoir traiter les situations où n’importe quelle variable peut être partiellement observée (et pas uniquement la variable classe).

### Multi-net

Cette approche originale exposée par [62] et [55] suppose (1) que les relations de causalité ou d’indépendance conditionnelles entre les variables ne sont pas nécessairement les mêmes selon les modalités de la classe, et (2) que la structure représentant les relations entre les caractéristiques pour une modalité

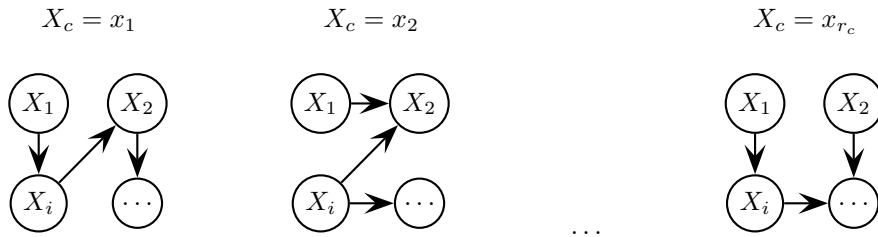


FIG. 3.3 – Approche multi-net

de la classe fixée est souvent plus simple que la structure représentant les relations entre toutes les variables (caractéristiques et classe). Au lieu de rechercher la structure optimale englobant les  $n$  variables, classes comprises, l'approche *multi-net* consiste à chercher  $r_c$  structures reliant uniquement les  $n - 1$  caractéristiques, avec une structure pour chaque modalité  $i$  de la classe ( $i \in [1 \dots r_c]$ ), comme illustré dans la figure 3.3.

Selon l'hypothèse (2), la plupart des approches de ce type utilisent des méthodes simples comme MWST ou BN-PC pour trouver chacune des structures, au lieu d'algorithmes plus lourds comme la recherche gloutonne.

### Apprentissage des modèles discriminants

Toutes les méthodes d'apprentissage de paramètres ou de structure évoquées jusqu'ici maximisent la vraisemblance sur toutes les variables, la variable classe ne tenant pas une place particulière lors de l'apprentissage. En prenant l'exemple de la régression logistique, Ng et Jordan [97] montrent que cet apprentissage *génératif* n'est pas le plus adapté dans le cas particulier de la classification, et qu'il est préférable d'utiliser un apprentissage de type *discriminant*. Pour cela, la fonction objectif n'est plus la vraisemblance de toutes les variables, mais la vraisemblance de la variable classe conditionnellement à toutes les autres, fonction permettant de mesurer directement le pouvoir discriminant du réseau bayésien.

Greiner et al. [66] développent ainsi un algorithme d'apprentissage des paramètres d'un réseau bayésien maximisant la vraisemblance conditionnelle (ELR). Il faut noter que cet apprentissage n'est plus aussi simple que dans le cas génératif. Dans la plupart des cas classiques, la maximisation de la vraisemblance revient à estimer les statistiques essentielles de l'échantillon (fréquence d'apparition d'un événement dans le cas discret, moyenne et variance empiriques dans le cas gaussien). Or, la maximisation de la vraisemblance conditionnelle n'est plus si aisée et passe par une étape d'optimisation itérative, comme la descente de gradient proposée dans l'algorithme ELR.

L'apprentissage de la structure d'un modèle discriminant est donc encore plus problématique. En effet, les méthodes d'apprentissage de structure évoquées précédemment sont des méthodes itératives conjuguant une étape de maximisation dans l'espace des graphes et une étape de maximisation dans l'espace des paramètres. Remplacer la vraisemblance par la vraisemblance conditionnelle conduirait donc à ajouter une étape d'optimisation itérative (celle concernant les paramètres) dans le parcours itératif de l'espace des graphes, ce qui rend la méthode particulièrement coûteuse en temps de calcul. Grossman et Domingos [67] ont alors eu l'idée de garder l'étape classique d'estimation des paramètres

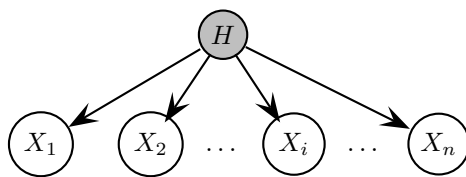


FIG. 3.4 – Modèle latent

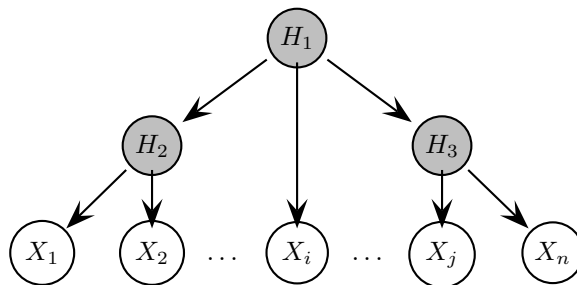


FIG. 3.5 – Modèle latent hiérarchique

par maximisation de la vraisemblance, mais d'employer un score prenant en compte le pouvoir discriminant du réseau bayésien pour le parcours dans l'espace des graphes. Le score présenté s'inspire du score BIC, en utilisant cette fois-ci la vraisemblance conditionnelle à la place de la vraisemblance classique.

### 3.7.2 Structures de réseaux bayésiens avec variables latentes

La connaissance apportée par un expert peut aussi se traduire par la création de variables latentes entre deux ou plusieurs nœuds, remettant en cause l'hypothèse de suffisance causale.

C'est le cas par exemple pour des problèmes de classification non supervisée où la classe n'est jamais mesurée. Il est donc possible de proposer l'équivalent d'un réseau bayésien naïf, le **modèle latent**, mais où la classe (représentée en gris dans la figure 3.4) ne fait pas partie des variables mesurées.

Les **modèles hiérarchiques latents** illustrés par la figure 3.5 ont été suggérés par [12] pour la visualisation de données et [142] pour la classification non supervisée. Ils généralisent la structure de modèle latent en faisant le parallèle avec les arbres phylogénétiques utilisés en bioinformatique ou avec les méthodes de classification hiérarchique.

L'apprentissage des paramètres pour le modèle latent ou le modèle hiérarchique latent s'appuie fortement sur l'algorithme EM. Cheeseman et al. ont ainsi développé AUTOCLASS [20], un algorithme bayésien de classification non supervisée utilisant l'algorithme EM. Attias et al. [7] ont utilisé les approches variationnelles popularisées par Jordan et al. [76] pour généraliser l'algorithme EM pour les modèles latents.

Peña et al. [100] simplifient la procédure de recherche de l'algorithme EM structurel pour rechercher une structure latente « augmentée », tout en employant une variante plus rapide de l'algorithme EM.

Dans ce genre de modèles, la détermination de la cardinalité des variables latentes est une tâche difficile, que nous décrirons plus en détail dans la section 3.8.

### 3.7.3 Autres structures particulières

La modélisation de systèmes complexes passe souvent par la détermination de régularités dans le modèle. La connaissance de ces régularités permet alors de restreindre l'identification du modèle à celles de ses composants qui peuvent se répéter plusieurs fois.

Ce type de modélisation se retrouve par exemple dans le formalisme des Réseaux Bayésiens Orientés Objets (OOBN [10]). Ces OOBN introduisent la notion d'objet dans un réseau bayésien, objet qui pourra se retrouver plusieurs fois dans le modèle, puis de relations entre les objets. La détermination de la structure d'un OOBN se traduit donc par la recherche de la structure interne de chaque objet et de la structure représentant les interactions entre les objets [9].

Le formalisme des Réseaux Bayésiens Temporels [93], et plus particulièrement celui des 2TBN (*Two-slice Temporal Bayesian Network*) reprend le même raisonnement. Dans ces modèles, les relations entre les variables sont décomposées en deux catégories. La première concerne les relations *intra-slice* entre les variables à un instant donné  $t$ , supposant que ces relations sont constantes au cours du temps.<sup>2</sup> L'autre catégorie de relation, *inter-slice*, décrit les dépendances entre les variables à un instant  $t$  et celles à un instant  $t + 1$ . Comme pour les Modèles de Markov Cachés, ce genre de décomposition suppose que la loi jointe sur toutes les variables dépend seulement des probabilités conditionnelles *intra-slices* et *inter-slices*. La détermination de la structure d'un 2TBN peut donc elle-aussi se simplifier en la recherche de ces deux catégories de relations, comme exposé par [60].

## 3.8 Découverte de variables latentes

Les algorithmes présentés dans les sections 3.3, 3.4 et 3.6 font l'hypothèse de suffisance causale. Or, cette hypothèse est souvent fautive pour des problèmes réels où toutes les variables ne sont pas forcément disponibles, et où, par exemple, certaines variables peuvent être reliées par une cause commune non mesurée.

Conscients de ce fait, des chercheurs ont tenté d'étendre la plupart des méthodes existantes à la découverte de variables latentes.

### 3.8.1 Recherche d'indépendances conditionnelles

Les auteurs respectifs de PC et IC (cf. p.32) ont utilisé la notion de causalité, dont nous parlons plus en détail dans la prochaine section, pour découvrir la présence de variables latentes à partir de la recherche d'indépendances conditionnelles. Pour cela, ils ont déterminé plusieurs genres de causalité (notations issues de [124]) :

---

<sup>2</sup>Pour cette raison, la terminologie Réseaux Bayésiens Temporels est plus appropriée que celle de Réseaux Bayésiens Dynamiques

<p>Algorithme IC*</p> <ul style="list-style-type: none"> <li>• Construction d'un graphe non orienté  Soit <math>\mathcal{G}</math> le graphe ne reliant aucun des nœuds <math>\mathcal{X}</math>  <math>\forall \{X_A, X_B\} \in \mathcal{X}^2</math>  Recherche de <math>Sepset(X_A, X_B)</math> tel que <math>X_A \perp X_B \mid Sepset(X_A, X_B)</math>  si <math>Sepset(X_A, X_B) = \emptyset</math> alors ajout de l'arête <math>X_A \text{ } o\text{-}o \text{ } X_B</math> dans <math>\mathcal{G}</math></li> <li>• Recherche des V-structures  <math>\forall \{X_A, X_B, X_C\} \in \mathcal{X}^3 / X_A</math> et <math>X_B</math> non adjacents et <math>X_A^* \rightarrow X_C^* \rightarrow X_B</math>,  si <math>X_C \notin SepSet(X_A, X_B)</math> alors on crée une V-structure :  <math>X_A^* \rightarrow X_C \leftarrow X_B</math></li> <li>• Ajout récursif de <math>\rightarrow</math>  Répéter  <math>\forall \{X_A, X_B\} \in \mathcal{X}^2</math>,  si <math>X_A^* \rightarrow X_B</math> et <math>X_A \rightsquigarrow X_B</math>, alors ajout d'une flèche à <math>X_B</math> :  <math>X_A^* \rightarrow X_B</math>  si <math>X_A</math> et <math>X_B</math> non adjacents, <math>\forall X_C</math> tel que <math>X_A^* \rightarrow X_C</math> et <math>X_C^* \rightarrow X_B</math>  alors <math>X_C \rightarrow X_B</math>  Tant qu'il est possible d'orienter des arêtes</li> </ul>															
<p>Définitions et notations :</p> <table border="0"> <tr> <td>Cause véritable</td> <td><math>X_A \rightarrow X_B</math></td> </tr> <tr> <td>Cause potentielle</td> <td><math>X_A \text{ } o\text{-}o \text{ } X_B</math> : <math>X_A \rightarrow X_B</math> ou <math>X_A \leftrightarrow X_B</math></td> </tr> <tr> <td>Cause artificielle</td> <td><math>X_A \leftrightarrow X_B</math> : <math>X_A \leftarrow H \rightarrow X_B</math></td> </tr> <tr> <td>Cause indéterminée</td> <td><math>X_A \text{ } o\text{-}o \text{ } X_B</math> : <math>X_A \rightarrow X_B, X_A \leftarrow X_B</math> ou <math>X_A \leftrightarrow X_B</math></td> </tr> <tr> <td><math>\mathcal{X}</math></td> <td>ensemble de tous les nœuds</td> </tr> <tr> <td><math>X_A^* \rightarrow X_B</math></td> <td><math>X_A \rightarrow X_B</math> ou <math>X_A \leftarrow X_B</math> ou <math>X_B \text{ } o\text{-}o \text{ } X_A</math></td> </tr> <tr> <td><math>X_A \rightsquigarrow X_B</math></td> <td>il existe un chemin dirigé reliant <math>X_A</math> et <math>X_B</math></td> </tr> </table>		Cause véritable	$X_A \rightarrow X_B$	Cause potentielle	$X_A \text{ } o\text{-}o \text{ } X_B$ : $X_A \rightarrow X_B$ ou $X_A \leftrightarrow X_B$	Cause artificielle	$X_A \leftrightarrow X_B$ : $X_A \leftarrow H \rightarrow X_B$	Cause indéterminée	$X_A \text{ } o\text{-}o \text{ } X_B$ : $X_A \rightarrow X_B, X_A \leftarrow X_B$ ou $X_A \leftrightarrow X_B$	$\mathcal{X}$	ensemble de tous les nœuds	$X_A^* \rightarrow X_B$	$X_A \rightarrow X_B$ ou $X_A \leftarrow X_B$ ou $X_B \text{ } o\text{-}o \text{ } X_A$	$X_A \rightsquigarrow X_B$	il existe un chemin dirigé reliant $X_A$ et $X_B$
Cause véritable	$X_A \rightarrow X_B$														
Cause potentielle	$X_A \text{ } o\text{-}o \text{ } X_B$ : $X_A \rightarrow X_B$ ou $X_A \leftrightarrow X_B$														
Cause artificielle	$X_A \leftrightarrow X_B$ : $X_A \leftarrow H \rightarrow X_B$														
Cause indéterminée	$X_A \text{ } o\text{-}o \text{ } X_B$ : $X_A \rightarrow X_B, X_A \leftarrow X_B$ ou $X_A \leftrightarrow X_B$														
$\mathcal{X}$	ensemble de tous les nœuds														
$X_A^* \rightarrow X_B$	$X_A \rightarrow X_B$ ou $X_A \leftarrow X_B$ ou $X_B \text{ } o\text{-}o \text{ } X_A$														
$X_A \rightsquigarrow X_B$	il existe un chemin dirigé reliant $X_A$ et $X_B$														

TAB. 3.21 – Algorithme IC\*

- **cause véritable** ( $X_A \rightarrow X_B$ ),
- **cause artificielle** ( $X_A \leftrightarrow X_B$ ) :  
 $X_A$  est vu comme la cause de  $X_B$  et réciproquement. Ces deux variables sont en réalité les conséquences d'une cause commune  $H$  non mesurée ( $X_A \leftarrow H \rightarrow X_B$ ),
- **cause potentielle** ( $X_A \text{ } o\text{-}o \text{ } X_B$ ) :  
 $X_A$  peut être soit la cause de  $X_B$  ( $X_A \rightarrow X_B$ ), soit la conséquence avec  $X_B$  d'une variable latente ( $X_A \leftrightarrow X_B$ ),
- **cause indéterminée** ( $X_A \text{ } o\text{-}o \text{ } X_B$ ) :  
Il est impossible de savoir si  $X_A$  cause  $X_B$  ou l'inverse, ou si elles sont les conséquences d'une variable latente ( $X_A \leftrightarrow X_B$ ).

La prise en compte de ces types de causalité dans les algorithmes précédents a abouti à l'algorithme FCI (*Fast Causal Inference*) pour Spirtes et al. [125, 124] et l'algorithme IC\* pour Pearl et al. [103] (détaillé dans la table 3.21). Comme pour PC et IC, la différence principale entre ces deux méthodes réside dans la construction du graphe non orienté de départ : suppression d'arêtes à partir d'un graphe complètement connecté pour FCI, et ajout d'arêtes à partir d'un graphe vide pour IC\*. La détermination du type de causalité s'effectue d'abord

lors de l'étape de détection de V-structures où certains arcs sont orientés, puis lors de l'étape suivante où des relations de causalité ambiguës sont levées.

Récemment, J. Zhang [141] a montré que les règles d'orientation proposées dans l'algorithme FCI ne sont pas complètes, élaborant une version augmentée et complète de l'algorithme.

Précisons que, même si ces méthodes se basent sur la notion de causalité, le réseau bayésien obtenu n'est pas un réseau bayésien causal, tel que défini dans la section 3.9.1. La structure obtenue est celle du représentant de la classe d'équivalence de Markov et l'orientation finale de cette structure ne tient plus nécessairement compte de l'idée de causalité.

### 3.8.2 Algorithmes basés sur un score

La découverte de variables latentes et le réglage de la cardinalité de ces variables sont souvent incorporés au processus d'apprentissage, et plus précisément aux méthodes de type recherche gloutonne.

Récemment, N. Zhang [143] a adapté l'algorithme EM structurel pour les modèles hiérarchiques latents. Cette adaptation tente d'optimiser la taille des variables latentes pendant l'apprentissage simultané de la structure et des paramètres, en suggérant d'autres opérateurs tels que l'ajout ou la suppression d'une variable latente, ou l'augmentation de la cardinalité d'une variable latente.

Martin et Vanlehn [90] suggèrent une heuristique permettant de ne pas ajouter une variable latente à n'importe quel moment lors de la recherche gloutonne précédente, mais dans des situations bien précises. En effet, ils considèrent que l'apparition d'une clique, i.e. un groupe de variables complètement connectées, et donc mutuellement dépendantes, peut alors n'être qu'un optimum local dû au fait qu'elles possèdent en commun une unique cause cachée. Leur opérateur d'ajout d'une variable latente introduit donc un nouveau nœud  $H_i$  dans le graphe, en remplaçant tous les arcs de la clique par des arcs partants de  $H_i$ .

La détermination de la cardinalité des variables latentes peut aussi être séparée du processus d'apprentissage pour rentrer dans le cadre de la sélection de modèles. Ainsi, plusieurs modèles peuvent être appris, avec différentes configurations de ces cardinalités. Le meilleur modèle, au sens d'un critère de score comme le critère BIC [54, 144], permettra ensuite de sélectionner les meilleures cardinalités des variables latentes. Malheureusement, l'utilisation de ces critères n'est pas toujours appropriée pour des modèles latents. Comment calculer par exemple la dimension effective du réseau bayésien  $Dim(\mathcal{B})$  lorsqu'il y a des variables latentes? Des corrections aux critères classiques ont été proposées par [82] pour les modèles hiérarchiques latents.

## 3.9 Cas particulier des réseaux bayésiens causaux

La notion de causalité est souvent associée au formalisme des réseaux bayésiens, parfois même à tort puisque le graphe complètement orienté obtenu à partir d'un algorithme d'apprentissage de structure n'est pas nécessairement causal.

La causalité est un champ d'étude très large, qui a motivé de nombreux travaux, de la Biologie [119] à l'Informatique en passant par la Philosophie [135].

Après avoir défini ce qu'est un réseau bayésien causal, et la notion d'intervention, nous nous intéresserons à la détermination de la structure de ces réseaux lorsque toutes les variables sont connues, puis dans un cas plus général.

### 3.9.1 Définition

Un réseau bayésien causal est un réseau bayésien pour lequel tous les arcs représentent des relations de causalité.

Leurs premiers avantages sont leur lisibilité et leur facilité d'interprétation pour les utilisateurs.

Un autre avantage des réseaux bayésiens causaux réside dans la possibilité de pouvoir estimer l'influence sur n'importe quelle variable du graphe d'une intervention externe sur une de ces variables. Cette notion importante d'intervention (ou manipulation) a amené Pearl [103] à distinguer le concept de mesure d'une variable ( $X_A = a$ ) à celle de manipulation de la variable  $X_A$  grâce à l'opérateur *do*-calcul.  $do(X_A = a)$  signifie ainsi qu'une intervention externe a forcé la variable  $X_A$  à prendre la valeur  $a$ .

Le principe de probabilité conditionnelle  $P(X_A | X_B)$ , symétrique grâce au théorème de Bayes, ne permet pas de représenter les relations, asymétriques, de causalité. L'usage de cet opérateur répond à ce problème. Si  $X_A$  est la cause de  $X_B$ , nous obtenons :

$$\begin{aligned} P(X_B = b | do(X_A = a)) &= P(X_B = b | X_A = a) \\ P(X_A = a | do(X_B = b)) &= P(X_A = a) \end{aligned}$$

Ces considérations ont débouché sur des travaux très intéressants sur l'idée d'identifiabilité, c'est-à-dire dans quelles conditions il est possible de calculer  $P(X_i | do(X_j))$ ,  $X_i$  et  $X_j$  étant n'importe quel nœud du graphe, et sur l'inférence causale, i.e. fournir des algorithmes capables de réaliser efficacement ce calcul lorsqu'il est possible.

### 3.9.2 Apprentissage sans variables latentes

Lorsqu'un expert détermine lui-même la structure d'un réseau bayésien, il utilise souvent implicitement la notion de causalité. À l'opposé, l'apprentissage du graphe à partir de données se fait dans un cadre plus général que celui des réseaux bayésiens causaux, cadre dans lequel plusieurs graphes seront équivalents, mais où un seul capturera éventuellement les relations de causalité du problème.

La découverte de réseaux bayésiens complètement causaux à partir de données est une question qui a été abordée plus récemment. Les avancées sur le sujet s'accordent sur le fait qu'il est impossible de travailler uniquement à partir de données d'observations. Les plans d'expériences, c'est à dire la façon dont les données ont été obtenues, sont des informations essentielles pour capturer la notion de causalité puisqu'ils définissent explicitement sur quelle(s) variable(s) a eu lieu l'intervention.

Les travaux théoriques de Eberhardt et al. [52] montrent que le nombre maximal d'interventions à effectuer sur le système est de  $N - 1$ , où  $N$  est le nombre de variables.

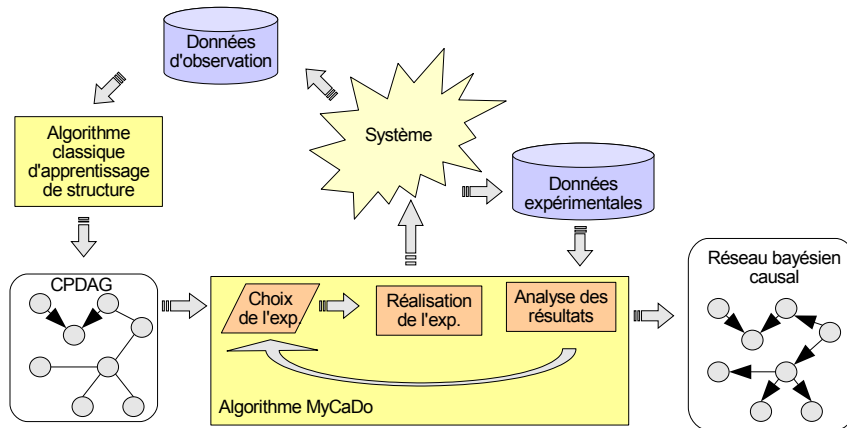


FIG. 3.6 – Apprentissage de la structure d’un réseau bayésien causal à partir de données d’observation et d’expérimentation : l’algorithme MyCaDo (MY CAusal DiscOvery) [159].

Deux types d’approches ont été élaborés. Les travaux de Cooper et Yo [36], Tong et Koller [131] ou Murphy [94] se placent dans le cadre de l’apprentissage actif, où les seules données seront celles obtenues par expérimentation, et où le modèle va être construit au fur et à mesure de ces expériences.

Nos travaux [159] partent d’une hypothèse différente. Nous supposons qu’un ensemble de données d’observation est déjà disponible, et a permis d’obtenir le représentant de la classe d’équivalence de Markov. Il reste donc à finir d’orienter cette structure à partir d’expérimentations sur le système. Notre algorithme, itératif, est résumé dans la figure 3.6. Il propose à l’utilisateur l’expérience à réaliser qui pourrait lui permettre d’orienter potentiellement le plus d’arêtes. Une fois que les résultats de cette expérience sont pris en compte dans le graphe, il faut ré-évaluer la situation pour choisir l’expérience suivante. Cette approche permet aussi de tenir compte des coûts éventuels d’expérimentation ou d’observation de chaque variable.

### 3.9.3 Apprentissage avec variables latentes

Un modèle causal semi-markovien (SMCM) [103] est un graphe sans circuit avec à la fois des arcs dirigés et bi-dirigés. Les nœuds du graphe sont associés aux variables observables, tandis que les arcs bi-dirigés représenteront implicitement des variables latentes.

Un avantage de ces modèles est cette représentation implicite des variables latentes dans le graphe. Contrairement aux approches à base de score abordées dans la section précédente, il n’est plus nécessaire de déclarer explicitement les variables latentes, ni de trouver la cardinalité de ces variables.

Spirtes *et al.* [125, 124] et Tian et Pearl [103, 129, 130] ont conçu des algorithmes efficaces permettant de répondre aux questions d’identifiabilité et d’inférence dans ces modèles.

Concernant l’apprentissage de réseaux bayésiens causaux avec variables la-

tentes, les chercheurs se sont tournés vers un autre formalisme, celui des graphes ancestraux maximaux (MAG), développés initialement par Richardson et Spirtes [111].

Ces travaux consistent à caractériser les classes d'équivalences des graphes ancestraux maximaux et à construire des opérateurs qui permettent de générer des graphes équivalents [4, 5, 139, 140]. La finalité de ces études est d'arriver à un algorithme s'inspirant de GES, décrit dans la section 3.5, mais travaillant dans l'espace des représentants des classes d'équivalence des MAG au lieu des DAG.

Malheureusement, comme pour l'algorithme GES, ces travaux ne permettent toujours pas de déterminer une structure qui soit complètement causale. De plus, il n'existe pas à notre connaissance d'algorithme d'inférence probabiliste ou causal travaillant à partir des graphes ancestraux maximaux.

Ces observations sont à l'origine de nos travaux les plus récents [160, 146, 147], où nous suggérons une approche mixte s'inspirant des principes décrits pour l'algorithme MyCaDo dans la section précédente.

La finalité de notre d'approche est d'utiliser des données d'observations et les algorithmes d'apprentissage de structure d'un MAG (ou du représentant de sa classe d'équivalence). Ensuite, l'idée est de mettre en œuvre une série d'expérimentations pour finir d'orienter "causalement" ce MAG, et surtout le transformer en un SMCM dans lequel il sera possible d'effectuer à la fois de l'inférence probabiliste et causale.

## Chapitre 4

# Conclusion et Perspectives

L'utilisation des réseaux bayésiens pour la modélisation de systèmes complexes passe inévitablement par la détermination de la structure et des paramètres du réseau. Nous avons vu dans les deux chapitres précédents comment construire ce modèle à partir d'expertises, ou de données, qu'elles soient complètes ou non.

De multiples méthodes existent, mais beaucoup de problèmes restent encore à résoudre concernant la prise en compte de données incomplètes, par exemple, ou l'apprentissage de modèles comme les réseaux bayésiens temporels.

Il est de plus en plus fréquent d'être confronté à des applications où les données sont nombreuses mais incomplètes, et que les utilisateurs aimeraient exploiter, de manière à en extraire le plus d'informations possible. Il ne suffit plus d'apprendre automatiquement un modèle qui réponde à une tâche précise, il faut arriver à un modèle permettant de découvrir des relations, des explications dont les utilisateurs pourront tirer profit.

La découverte de structures causales sans variables latentes est une première étape, pas encore complètement résolue, mais elle n'est pas suffisante. Rares sont les problèmes où les variables pertinentes sont toutes connues, et l'oublier peut aboutir à la découverte de relations causales erronées. Il faut donc à présent continuer à explorer la piste des réseaux bayésiens causaux avec variables latentes comme les modèles causaux semi-markoviens (SMCM). Ces modèles sont bien étudiés en ce qui concerne l'inférence causale, mais l'apprentissage de leur structure reste à approfondir.

De nombreuses extensions des réseaux bayésiens ont été proposées. Ainsi, les diagrammes d'influence permettent de représenter les problèmes de décision, où certaines variables sont des décisions prises ou des actions réalisées par l'utilisateur, d'autres sont observées à l'issue de ce choix, et les dernières représentent le coût de telle ou telle décision dans un certain contexte. Il est aussi possible de faire le lien avec des formalismes comme les processus de décision de Markov (MDP) qui peuvent être considérés comme le croisement des réseaux bayésiens temporels et des diagrammes d'influence. De plus, la prise en compte de variables latentes ou de données incomplètes mène à un autre formalisme, celui des processus de décision de Markov partiellement observés (POMDP). Une communauté très importante de chercheurs étudie l'apprentissage des paramètres

de ces modèles, avec des méthodes issues, entre autres, de l'apprentissage par renforcement, ainsi que l'inférence, i.e. l'obtention de la séquence de décision (la politique) optimale. La détermination de la structure de tels modèles est un problème peu abordé à notre connaissance.

Les modèles les plus communément employés ne considèrent que des densités de probabilités conditionnelles de variables discrètes. Dans le cas des variables continues, l'utilisation de lois gaussiennes conditionnelles est possible dans certains algorithmes, mais cette approximation est assez pauvre. Certaines heuristiques permettent alors d'approcher la loi conditionnelle par un mélange de gaussiennes, multipliant ainsi le nombre de paramètres à estimer (dont le nombre de gaussiennes utilisées). Les méthodes à noyau ont fait leur preuve dans de nombreux problèmes de classification, étendant les modèles exponentiels classiques. Une piste de recherche concerne donc l'incorporation de ces noyaux dans la paramétrisation des lois conditionnelles d'un réseau bayésien, afin de mieux les modéliser.

Il serait aussi envisageable d'exploiter les méthodes à noyaux dans le cadre d'un autre modèle graphique causal, le modèle d'équations structurelles (SEM), où les dépendances entre les variables sont définies par des équations linéaires. L'apprentissage de la structure de ces modèles, dans leur version linéaire, fait déjà l'objet de travaux assez proches de ceux proposés dans le cadre des réseaux bayésiens.

Une autre piste intéressante est liée aux travaux de Jordan *et al.* [75] qui ont mis en avant la notion plus générale de modèle graphique probabiliste, unifiant ainsi des approches développées auparavant de façon concurrente comme les réseaux bayésiens, les modèles de Markov cachés, les filtres de Kalman ou les champs aléatoires de Markov. Il est donc tout à fait concevable d'utiliser les principes développés pour les réseaux bayésiens à la découverte de la structure d'autres modèles graphiques, qu'ils soient dirigés ou pas, comme c'est le cas pour les champs aléatoires de Markov.

## Chapitre 5

## Références

Les références [145] à [190] figurent dans le chapitre 12, p.123

- [1] Silvia Acid and Luis M. de Campos. A hybrid methodology for learning belief networks : Benedict. *Int. J. Approx. Reasoning*, 27(3) :235–262, 2001.
- [2] A. Aho and J. Ullman. *Concepts fondamentaux de l'informatique*. Dunod, 1998.
- [3] H Akaike. Statistical predictor identification. *Ann. Inst. Statist. Math.*, 22 :203–217, 1970.
- [4] A.R. Ali and T. Richardson. Markov equivalence classes for maximal ancestral graphs. In *Proc. of the 18th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 1–9, 2002.
- [5] Ayesha R. Ali, Thomas Richardson, Peter Spirtes, and J. Zhang. Orientation rules for constructing markov equivalence classes of maximal ancestral graphs. Technical Report 476, Dept. of Statistics, University of Washington, 2005.
- [6] S. Andersson, D. Madigan, and M. Perlman. A characterization of markov equivalence classes for acyclic digraphs. Technical Report 287, Department of Statistics, University of Washington, 1995.
- [7] Hagai Attias. Inferring parameters and structure of latent variable models by variational bayes. In Kathryn B. Laskey and Henri Prade, editors, *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 21–30, S.F., Cal., July 30–August 1 1999. Morgan Kaufmann Publishers.
- [8] Vincent Auvray and Louis Wehenkel. On the construction of the inclusion boundary neighbourhood for markov equivalence classes of bayesian network structures. In Adnan Darwiche and Nir Friedman, editors, *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 26–35, S.F., Cal., 2002. Morgan Kaufmann Publishers.
- [9] O. Bangsø, H. Langseth, and T. D. Nielsen. Structural learning in object oriented domains. In I. Russell and J. Kolen, editors, *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS-01)*, pages 340–344, Key West, Florida, USA, 2001. AAAI Press.
- [10] Olav Bangso and Pierre-Henry Wuillemin. Object oriented bayesian networks : A framework for topdown specification of large Bayesian networks and repetitive structures, Technical Report CIT-87.2-00-obphw1. Technical report, Department of Computer Science, University of Aalborg, September 2000.
- [11] M. Bendou and P. Munteanu. Nouvel algorithme d'apprentissage des classes d'équivalence des réseaux bayésiens. In Michel Liquière et Marc Sebban, editor, *Sixième Conférence Apprentissage CAp'2004*, pages 129–141, Montpellier, France, 2004. Presses Universitaires de Grenoble.
- [12] C.M. Bishop and M.E. Tipping. A hierarchical latent variable model for data visualisation. *IEEE T-PAMI*, 3(20) :281–293, 1998.
- [13] C. Borgelt and R. Kruse. *Graphical Models - Methods for Data Analysis and Mining*. John Wiley & Sons, Chichester, United Kingdom, 2002.

- [14] Bernadette Bouchon-Meunier and Christophe Marsala. *Logique floue, principes, aide à la décision*. Traité IC2, série informatique et systèmes d'information. Editions Hermes, 2003.
- [15] R. Bouckaert. Probabilistic network construction using the minimum description length principle. *Lecture Notes in Computer Science*, 747 :41–48, 1993.
- [16] Mark Brodie, Irina Rish, and Sheng Ma. Intelligent probing : A cost-effective approach to fault diagnosis in computer networks. *IBM Systems Journal*, 41(3) :372–385, 2002.
- [17] W. Buntine. Theory refinement on bayesian networks. In Bruce D'Ambrosio, Philippe Smets, and Piero Bonissone, editors, *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, pages 52–60, San Mateo, CA, USA, July 1991. Morgan Kaufmann Publishers.
- [18] Jeremy Cain. *Planning improvements in natural resources management – guidelines for using Bayesian networks to support the planning and management of development programmes in the water sector and beyond*. Centre for Ecology and Hydrology, UK, 2004.
- [19] R. Castelo and T. Kocka. Towards an inclusion driven learning of bayesian networks. Technical Report UU-CS-2002-05, Institute of information and computing sciences, University of Utrecht, 2002.
- [20] P. Cheeseman and J. Stutz. Bayesian classification (AUTOCLASS) : Theory and results. In U. Fayyad, G. Piatetsky-Shapiro, P Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 607–611. AAAI Press/MIT Press, 1996.
- [21] Jie Cheng, David Bell, and Weiru Liu. An algorithm for bayesian network construction from data. In *Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics AISTAT'97*, pages 83–90, 1997.
- [22] Jie Cheng, David Bell, and Weiru Liu. Learning belief networks from data : An information theory based approach. In *Proceedings of the sixth ACM International Conference on Information and Knowledge Management CIKM*, pages 325–331, 1997.
- [23] Jie Cheng and Russell Greiner. Comparing bayesian network classifiers. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 101–108, San Francisco, CA, 1999. Morgan Kaufmann Publishers.
- [24] Jie Cheng and Russell Greiner. Learning bayesian belief network classifiers : Algorithms and system. In *Proceedings of the Canadian Conference on AI 2001*, volume 2056, pages 141–151, 2001.
- [25] Jie Cheng, Russell Greiner, Jonathan Kelly, David Bell, and Weiru Liu. Learning Bayesian networks from data : An information-theory based approach. *Artificial Intelligence*, 137(1-2) :43–90, 2002.
- [26] D. Chickering, D. Geiger, and D. Heckerman. Learning bayesian networks : Search methods and experimental results. In *Proceedings of Fifth Conference on Artificial Intelligence and Statistics*, pages 112–128, 1995.
- [27] D. Chickering and D. Heckerman. Efficient Approximation for the Marginal Likelihood of Incomplete Data given a Bayesian Network. In *UAI'96*, pages 158–168. Morgan Kaufmann, 1996.

- [28] David Chickering. A transformational characterization of equivalent bayesian network structures. In Philippe Besnard and Steve Hanks, editors, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI'95)*, pages 87–98, San Francisco, CA, USA, August 1995. Morgan Kaufmann Publishers.
- [29] David Chickering. Learning equivalence classes of bayesian network structures. In Eric Horvitz and Finn Jensen, editors, *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages 150–157, San Francisco, August 1–4 1996. Morgan Kaufmann Publishers.
- [30] David Chickering. Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2 :445–498, February 2002.
- [31] David Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3 :507–554, November 2002.
- [32] David Chickering and Christopher Meek. Finding optimal bayesian networks. In Adnan Darwiche and Nir Friedman, editors, *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 94–102, S.F., Cal., 2002. Morgan Kaufmann Publishers.
- [33] C.K. Chow and C.N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3) :462–467, 1968.
- [34] R. Christensen. *Log-Linear Models and Logistic Regression*. Springer, 1997.
- [35] G. Cooper and E. Hersovits. A bayesian method for the induction of probabilistic networks from data. *Maching Learning*, 9 :309–347, 1992.
- [36] Gregory F. Cooper and Changwon Yoo. Causal discovery from a mixture of experimental and observational data. In *UAI '99 : Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 116–125, 1999.
- [37] T. Cormen, C. Leiserson, and R. Rivest. *Introduction à l'algorithmique*. Dunod, 1994.
- [38] F. Corset. *Optimisation de la maintenance à partir de réseaux bayésiens et fiabilité dans un contexte doublement censuré*. PhD thesis, Université Joseph Fourier, 2003.
- [39] Robert Cowell, A. Dawid, Steffen Lauritzen, and David Spiegelhalter. *Probabilistic Networks and Expert Systems*. Statistics for Engineering and Information Science. Springer-Verlag, 1999.
- [40] Fabio Cozman and Ira Cohen. Unlabeled data can degrade classification performance of generative classifiers. In *Fifteenth International Florida Artificial Intelligence Society Conference*, pages 327–331, 2002.
- [41] Denver Dash and Marek J. Druzdzel. A hybrid anytime algorithm for the construction of causal models from sparse data. In Kathryn B. Laskey and Henri Prade, editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, UAI 99*, pages 142–149. Morgan Kaufmann, 1999.

- [42] Denver Dash and Marek J. Druzdzel. Robust independence testing for constraint-based learning of causal structure. In *UAI '03, Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence*, pages 167–174, 2003.
- [43] Luis de Campos and Juan Huete. A new approach for learning belief networks using independence criteria. *International Journal of Approximate Reasoning*, 24(1) :11–37, 2000.
- [44] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39 :1–38, 1977.
- [45] M.C. Desmarais, P. Meshkinfam, and M. Gagnon. Learned student models with item to item knowledge structures. *User Modeling and User-Adapted Interaction*, page (to appear), 2006.
- [46] M. Deviren and K. Daoudi. Apprentissage de structures de réseaux bayésiens dynamiques pour la reconnaissance de la parole. In *XXIVèmes Journées d'étude sur la parole*, 2002.
- [47] F. Diez. Parameter adjustment in Bayes networks. The generalized noisy OR-gate. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 99–105, Washington D.C., 1993. Morgan Kaufmann, San Mateo, CA.
- [48] D. Dor and M. Tarsi. A simple algorithm to construct a consistent extension of a partially oriented graph. Technical Report R-185, Cognitive Systems Laboratory, UCLA Computer Science Department, 1992.
- [49] D. Draper. Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society, series B*, 57(1) :45–70, 1995.
- [50] M. Druzdzel, L. Van der Gaag, M. Henrion, and F. Jensen. Building probabilistic networks : “where do the numbers come from?” guest editors introduction. *IEEE Transactions on Knowledge and Data Engineering*, 12(4) :481–486, 2000.
- [51] Marek Druzdzel and F. Diez. Criteria for combining knowledge from different sources in probabilistic models. In *Working Notes of the workshop on Fusion of Domain Knowledge with Data for Decision Support, Sixteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, pages 23–29, Stanford, CA, 30 June 2000.
- [52] F. Eberhardt, C. Glymour, and R. Scheines. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among  $n$  variables. In *Proc. of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 178–183, 2005.
- [53] Gal Elidan and Nir Friedman. Learning the dimensionality of hidden variables. In *Uncertainty in Artificial Intelligence : Proceedings of the Seventeenth Conference (UAI-2001)*, pages 144–151, San Francisco, CA, 2001. Morgan Kaufmann Publishers.
- [54] Chris Fraley and Adrian Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8) :578–588, 1998.
- [55] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3) :131–163, 1997.

- [56] Nir Friedman. Learning belief networks in the presence of missing values and hidden variables. In *Proc. 14th International Conference on Machine Learning*, pages 125–133. Morgan Kaufmann, 1997.
- [57] Nir Friedman. The bayesian structural EM algorithm. In Gregory Cooper and Serafin Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 129–138, San Francisco, July 24–26 1998. Morgan Kaufmann.
- [58] Nir Friedman, Moises Goldszmidt, and Abraham Wyner. Data analysis with bayesian networks : A bootstrap approach. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 206–215, San Francisco, CA, 1999. Morgan Kaufmann Publishers.
- [59] Nir Friedman and Daphne Koller. Being bayesian about network structure. In C. Boutilier and M. Goldszmidt, editors, *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 201–210, SF, CA, June 30 – July 3 2000. Morgan Kaufmann Publishers.
- [60] Nir Friedman, Kevin Murphy, and Stuart Russell. Learning the structure of dynamic probabilistic networks. In Gregory F. Cooper and Serafin Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 139–147, San Francisco, July 24–26 1998. Morgan Kaufmann.
- [61] Dan Geiger. An entropy-based learning algorithm of bayesian conditional trees. In *Uncertainty in Artificial Intelligence : Proceedings of the Eighth Conference (UAI-1992)*, pages 92–97, San Mateo, CA, 1992. Morgan Kaufmann Publishers.
- [62] Dan Geiger and David Heckerman. Knowledge representation and inference in similarity networks and bayesian multinets. *Artificial Intelligence*, 82(1–2) :45–74, 1996.
- [63] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6) :721–741, November 1984.
- [64] Steven Gillispie and Christiane Lemieux. Enumerating markov equivalence classes of acyclic digraph models. In *Uncertainty in Artificial Intelligence : Proceedings of the Seventeenth Conference (UAI-2001)*, pages 171–177, San Francisco, CA, 2001. Morgan Kaufmann Publishers.
- [65] Russell Greiner, Adam Grove, and Dale Schuurmans. Learning Bayesian nets that perform well. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pages 198–207, San Francisco, CA, 1997. Morgan Kaufmann Publishers.
- [66] Russell Greiner, Xiaoyuan Su, Bin Shen, and Wei Zhou. Structural extension to logistic regression : Discriminative parameter learning of belief net classifiers. *Machine Learning Journal*, 59(3) :297–322, 2005.
- [67] Daniel Grossman and Pedro Domingos. Learning bayesian network classifiers by maximizing conditional likelihood. In *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004)*, page (CDRom), 2004.

- [68] D. Heckerman, D. Geiger, and M. Chickering. Learning Bayesian networks : The combination of knowledge and statistical data. In Ramon Lopez de Mantaras and David Poole, editors, *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 293–301, San Francisco, CA, USA, July 1994. Morgan Kaufmann Publishers.
- [69] D. Heckerman, C. Meek, and G. Cooper. A bayesian approach to causal discovery. Technical Report MSR-TR-97-05, Microsoft Research, 1997.
- [70] David Heckerman. A tutorial on learning with bayesian network. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 301–354. Kluwer Academic Publishers, Boston, 1998.
- [71] M. Henrion. Some practical issues in constructing belief networks. In L. N. Kanal, T. S. Levitt, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 3*, volume 8 of *Machine Intelligence and Pattern Recognition*, pages 161–174. North-Holland, Amsterdam, 1989.
- [72] William Hsu, Haipeng Guo, Benjamin Perry, and Julie Stilson. A permutation genetic algorithm for variable ordering in learning Bayesian networks from data. In W. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. Miller, E. Burke, and N. Jonoska, editors, *GECCO 2002 : Proceedings of the Genetic and Evolutionary Computation Conference*, pages 383–390, New York, 9-13 July 2002. Morgan Kaufmann Publishers.
- [73] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3 :79–87, 1991.
- [74] F. Jensen, S. Lauritzen, and K. Olesen. Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly*, 4 :269–282., 1990.
- [75] M. Jordan. *Learning in Graphical Models*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
- [76] Michael Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence Saul. An introduction to variational methods for graphical models. In Michael Jordan, editor, *Learning in Graphical Models*, pages 105–162. Kluwer Academic Publishers, Boston, 1998.
- [77] L. Jouffe and P. Munteanu. Smart-greedy+ : Apprentissage hybride de réseaux bayésiens. In *Colloque francophone sur l'apprentissage, CAP, St. Etienne*, June 2000.
- [78] L. Jouffe and P. Munteanu. New search strategies for learning bayesian networks. In *Proceedings of Tenth International Symposium on Applied Stochastic Models and Data Analysis, ASMDA, Compiègne*, pages 591–596, June 2001.
- [79] E. Keogh and M. Pazzani. Learning augmented bayesian classifiers : A comparison of distribution-based and classification-based approaches. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, pages 225–230, 1999.
- [80] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598 :671–680, 1983.

- [81] U. Kjaerulff. Triangulation of graphs : Algorithms giving small total state space. Technical report, Departement of Matematics and computer science, Aalborg Univercity Denmark, 1990.
- [82] T. Kocka and N. Zhang. Dimension correction for hierarchical latent class models. In Adnan Darwiche and Nir Friedman, editors, *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 267–274, S.F., Cal., 2002. Morgan Kaufmann Publishers.
- [83] Paul Krause. Learning probabilistic networks, 1998.
- [84] Wai Lam and Fahiem Bacchus. Using causal information and local measures to learn bayesian networks. In David Heckerman and Abe Mamdani, editors, *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 243–250, San Mateo, CA, USA, July 1993. Morgan Kaufmann Publishers.
- [85] Evelina Lamma, Fabrizio Riguzzi, and Sergio Storari. Exploiting association and correlation rules - parameters for improving the k2 algorithm. In Ramon López de Mántaras and Lorenza Saitta, editors, *Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'2004*, pages 500–504. IOS Press, 2004.
- [86] Pat Langley, Wayne Iba, and Kevin Thompson. An analysis of bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 223–228, San Jose, CA, 1992. AAAI Press.
- [87] P. Larrañaga, C. Kuijpers, R. Murga, and Y. Yurramendi. Learning bayesian network structures by searching the best order ordering with genetic algorithms. *IEEE Transactions on System, Man and Cybernetics*, 26 :487–493, 1996.
- [88] S. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19 :191–201, 1995.
- [89] D. Madigan, A. Raftery, J. York, J. Bradshaw, and R. Almond. Strategies for graphical model selection. In P. Cheeseman and R. Oldford, editors, *Selecting Models from Data : Artificial Intelligence and Statistics IV*, pages 91–100. Springer, 1993.
- [90] J. Martin and K. Vanlehn. Discrete factor analysis : Learning hidden variables in bayesian network. Technical report, Department of Computer Science, University of Pittsburgh, 1995.
- [91] C. Meek. *Graphical Models : Selecting causal and statistical models*. PhD thesis, Carnegie Mellon University, 1997.
- [92] P. Munteanu and M. Bendou. The EQ framework for learning equivalence classes of bayesian networks. In *First IEEE International Conference on Data Mining (IEEE ICDM)*, pages 417–424, San José, November 2002.
- [93] K. Murphy. *Dynamic bayesian Networks : Representation, Inference and Learning*. PhD thesis, University of california, Berkeley, 2002.
- [94] Kevin P. Murphy. Active learning of causal bayes net structure. Technical report, Department of Computer Science, UC Berkeley, 2001.
- [95] Radford Neal and Geoffrey Hinton. A view of the EM algorithm that justifies incremental, sparse and other variants. In Michael Jordan, editor,

- Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, Boston, 1998.
- [96] C. Needham, J. Bradford, A. Bulpitt, and D.R. Westhead. Application of bayesian networks to two classification problems in bioinformatics. In S. Barber, P.D. Baxter, K.V. Mardia, and R.E. Walls, editors, *Quantitative Biology, Shape Analysis, and Wavelets*, pages 87–90. Leeds University Press, 2005.
- [97] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers : A comparison of logistic regression and naive bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 841–848, Cambridge, MA, 2002. MIT Press.
- [98] S. Nowlan. *Soft competitive adaptation : Neural Network Learning Algorithms based on Fitting Statistical Mixtures*. PhD thesis, Carnegie Mellon Univ., Pittsburgh, 1991.
- [99] Agnieszka Onisko, Marek Druzdzal, and Hanna Wasyluk. Learning bayesian network parameters from small data sets : Application of noisy-or gates. *International Journal of Approximate Reasoning*, 27(2) :165–182, 2001.
- [100] J. Peña, J. Lozano, and P. Larrañaga. An improved bayesian structural EM algorithm for learning bayesian networks for clustering. *Pattern Recognition Letters*, 21 :779–786, 2000.
- [101] J. Pearl. Reverend bayes on inference engines : A distributed hierarchical approach. *Proceedings AAAI National Conference on AI*, pages 133–136, 1982.
- [102] J. Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29 :241–288, 1986.
- [103] Judea Pearl. *Causality : Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, England, 2000.
- [104] Judea Pearl and Tom Verma. A theory of inferred causation. In James Allen, Richard Fikes, and Erik Sandewall, editors, *KR’91 : Principles of Knowledge Representation and Reasoning*, pages 441–452, San Mateo, California, 1991. Morgan Kaufmann.
- [105] S. Populaire, T. Dencœux, A. Guilikeng, P. Gineste, and J. Blanc. Fusion of expert knowledge with data using belief functions : a case study in wastewater treatment. In *Proceedings of the 5th International Conference on Information Fusion, IF 2002*, pages 1613 – 1618, 2002.
- [106] Malcolm Pradhan, Gregory Provan, Blackford Middleton, and Max Henrion. Knowledge engineering for large belief networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pages 484–490, San Francisco, CA, 1994. Morgan Kaufmann Publishers.
- [107] Marco Ramoni and Paola Sebastiani. Parameter estimation in bayesian networks from incomplete databases. *Intelligent Data Analysis*, 2(1-4) :139–160, 1998.
- [108] Marco Ramoni and Paola Sebastiani. Robust learning with missing data. *Machine Learning*, 45 :147–170, 2000.

- [109] S. Renooij. Probability elicitation for belief networks : Issues to consider. *Knowledge Engineering Review*, 16(3) :255–269, 2001.
- [110] M. Richardson and P. Domingos. Learning with knowledge from multiple experts. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003)*, pages 624–631, Washington, DC, 2003. Morgan Kaufmann.
- [111] T. Richardson and P. Spirtes. Ancestral graph markov models. Technical Report 375, Dept. of Statistics, University of Washington, 2002.
- [112] J. Rissanen. Modelling by shortest data description. *Automatica*, 14 :465–471, 1978.
- [113] Christian Robert. *The bayesian Choice : a decision-theoretic motivation*. Springer, New York, 1994.
- [114] R. Robinson. Counting unlabeled acyclic digraphs. In C. Little, editor, *Combinatorial Mathematics V*, volume 622 of *Lecture Notes in Mathematics*, pages 28–43, Berlin, 1977. Springer.
- [115] D.B. Rubin. Inference and missing data. *Biometrika*, 63 :581–592, 1976.
- [116] J. Sacha, L. Goodenday, and K. Cios. Bayesian learning for cardiac spect image interpretation. *Artificial Intelligence in Medecine*, 26 :109–143, 2002.
- [117] M. Sakarovitch. *Optimisation Combinatoire –Méthodes Mathématiques et Algorithmiques : Graphes et Programmation Linéaire*. Hermann, Paris, 1984.
- [118] G Schwartz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2) :461–464, 1978.
- [119] Bill Shipley. *Cause and Correlation in Biology*. Cambridge University Press, 2000.
- [120] Moninder Singh and Marco Valtorta. An algorithm for the construction of bayesian network structures from data. In David Heckerman and E. H. Mamdani, editors, *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence UAI 93*, pages 259–265. Morgan Kaufmann, 1993.
- [121] P. Smets. Data fusion in the transferable belief model. In *Proceedings of FUSION'2000*, pages 21–33, Paris, France, 2000.
- [122] D. Spiegelhalter and S. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20 :579–605, 1990.
- [123] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. Springer-Verlag, 1993.
- [124] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. The MIT Press, 2nd edition, 2000.
- [125] Peter Spirtes, Christopher Meek, and Thomas Richardson. Causal inference in the presence of latent variables and selection bias. In Philippe Besnard and Steve Hanks, editors, *UAI '95 : Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, August 18-20, 1995, Montreal, Quebec, Canada*, pages 499–506. Morgan Kaufmann, 1995.

- [126] Sampath Srinivas. A generalization of the noisy-or model. In David Heckerman and Abe Mamdani, editors, *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 208–218, San Mateo, CA, USA, July 1993. Morgan Kaufmann Publishers.
- [127] Joe Suzuki. Learning Bayesian belief networks based on the MDL principle : An efficient algorithm using the branch and bound technique. *IEICE Transactions on Information and Systems*, E82-D(2) :356–367, 1999.
- [128] Bo Thiesson, Christopher Meek, and David Heckerman. Accelerating EM for large databases. *Machine Learning*, 45(3) :279–299, 2001.
- [129] Jin Tian and Judea Pearl. On the testable implications of causal models with hidden variables. In *UAI '02, Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence*, pages 519–527, 2002.
- [130] Jin Tian and Judea Pearl. In the identification of causal effects. Technical Report R-290-L, UCLA, 2003.
- [131] Simon Tong and Daphne Koller. Active learning for structure in bayesian networks. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001*, pages 863–869. Morgan Kaufmann, 2001.
- [132] L. van der Gaag, S. Renooij, C. Witteman, B. Aleman, and B. Taal. Probabilities for a probabilistic network : a case study in oesophageal cancer. *Artificial Intelligence in Medicine*, 25(2) :123–148, june 2002.
- [133] S. van Dijk, L. van der Gaag, and D. Thierens. A skeleton-based approach to learning bayesian networks from data. In *Proceedings of the Seventh Conference on Principles and Practice of Knowledge Discovery in Databases*. Kluwer, 2003.
- [134] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In M. Henrion, R. Shachter, L. Kanal, and J. Lemmer, editors, *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 220–227, San Francisco, 1991. Morgan Kaufmann.
- [135] Jon Williamson. *Bayesian Nets And Causality : Philosophical And Computational Foundations*. Oxford University Press, 2005.
- [136] Man Leung Wong, Shing Yan Lee, and Kwong Sak Leung. Data mining of bayesian networks using cooperative coevolution. *Decision Support Systems*, 38(3) :451–472, 2004.
- [137] C. Xie, H. Wang, and L. Nozick. Modeling travel mode choice behavior by a bayesian network. In *85th Transportation Research Board Annual Meeting, January 22-26, Washington, D.C.*, page (CDROM), 2006.
- [138] Raanan Yehezkel and Boaz Lerner. Bayesian network structure learning by recursive autonomy identification. In Dit-Yan Yeung, James T. Kwok, Ana L. N. Fred, Fabio Roli, and Dick de Ridder, editors, *Proceedings of Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops, SSPR 2006 and SPR 2006, Hong Kong, China*, volume 4109 of *Lecture Notes in Computer Science*, pages 154–162. Springer, 2006.
- [139] J. Zhang and P. Spirtes. A characterization of markov equivalence classes for ancestral graphical models. Technical Report 168, Dept. of Philosophy, Carnegie-Mellon University, 2005.

- [140] J. Zhang and P. Spirtes. A transformational characterization of markov equivalence for directed acyclic graphs with latent variables. In *Proc. of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 667–674, 2005.
- [141] Jiji Zhang. *Causal Inference and Reasoning in Causally Insufficient Systems*. PhD thesis, Carnegie Mellon University, July 2006.
- [142] N. Zhang. Hierarchical latent class models for cluster analysis. In *Proceedings of AAAI'02*, pages 230–237, 2002.
- [143] N. Zhang. Structural EM for hierarchical latent class model. Technical report, HKUST-CS03-06, 2003.
- [144] N. Zhang, T. Nielsen, and F. Jensen. Latent variable discovery in classification models. *Artificial Intelligence in Medicine*, 30(3) :283–299, 2004.