De l'identification de structure de réseaux bayésiens à la reconnaissance de formes à partir d'informations complètes ou incomplètes

THÈSE DE DOCTORAT

présentée et soutenue publiquement le mardi 28 novembre 2006

pour l'obtention du grade de

Docteur de l'Institut National des Sciences Appliquées de Rouen

(spécialité informatique, génie traitement du signal)

par

François Olivier

Composition du jury

Président :	Paquet Thierry	Professeur des Universités, LITIS, Université de Rouen.
Rapporteurs :	Benferhat Salem	Professeur des Universités, CRIL, Université d'Artois,
	Mazer Emmanuel	Directeur de Recherche, CNRS, ProBayes, INRIA Grenoble.
Examinateurs :	Aussem Alexandre	Professeur des Universités, PRISMa, Université de Lyon 1,
	Jaffray Jean-Yves	Professeur des Universités, LIP6, Université de Paris 6.
Directeurs :	Canu Stéphane	Professeur des Universités, LITIS, INSA de Rouen,
	Leray Philippe	Maître de Conférences, LITIS, INSA de Rouen.

Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes.

A la mémoire de mon grand-père,

à ma mère,

à mon père.

Remerciements

J'aimerais remercier toutes les personnes qui m'ont permis de faire ce travail, mais elles sont trop nombreuses. Commençons par le commencement, quoique, nous n'allons peut-être pas remonter jusque là...

Je remercie tous mes enseignants, en particulier, ceux de l'école préparatoire du lycée Pierre D'Ailly de Compiègne, ceux de l'UFR des Sciences de l'université de Picardie Jules Vernes d'Amiens, l'équipe d'enseignement du DEA Informatique et Recherche Opérationnelle de l'université Pierre et Marie Curie (mouture 2001-02) avec Michel Minoux et Jean-Yves Jaffray. Ce dernier ayant fait un cours d'introduction aux modèles qui sont l'objet de ce document.

La personne la plus importante quant à la réalisation de cette thèse est sans nul doute Philippe Leray, un ancien de l'équipe du LiP6, arrivé dans l'équipe de Stéphane Canu au laboratoire PSI[i] de Rouen. Je ne remercierai jamais assez Philippe pour toutes les discutions intéressantes et fructueuses que nous avons eues, mais aussi, pour la patience dont il a fait part à mon égard. Je remercie également Stéphane pour la manière dont il a mené l'équipe 'apprentissage' du laboratoire à laquelle j'appartenais.

Bien sûr, je n'oublie pas les thésards qui m'ont accompagné. Certains ont partagé mon bureau : JC (la rosette), Vincent (la planque), Fabi (le boudeur), mais aussi Aurélie (la retardataire), puis FredSu, la belle Gaëlle, Karina la mexicana, puis encore Gregory, Sam et Stijn (les rares 'bayésiens' belges), Ahmad (pour ses magnifiques friandises libanaises) et Iyadh. Je n'oublie pas les autres doctorants du laboratoire, en particulier : Filip le roumain, Stéphane (collègue représentant au conseil de labo), Julien, Christophe, Guillaume, Clément, Yann, Hervé, Bruno, Tété du Togo . . .

Je remercie également les permanents et personnels administratifs du laboratoire, en particulier : Nathalie (qui gérait les relations labo-doctorants avec Philippe), Sandra, Dominique, Laurence, Brigitte (on ne peut rien faire sans les secrétaires...), Yves, Jacques, Jean-Pierre, Abdelaziz, Laurent, Thierry, Patrick, Abdel, Jean-Philippe, Sébastien, Pierre, Mhamed, Fabrice, Jean-François . . .

Il me faut également remercier les personnes du département ASI de l'INSA de Rouen pour leur bonne humeur, comme Florence (l'âme du département), Nicolas (le motard contestataire du dimanche), Nicolas (MC *es* polémiques au RU), Alain le rital malgache, Gil le togolais libéral, Alexandrina la roumaine, Gwenaëlle la bretonne, Michel, Sébastien (encore un motard) . . .

Pour les nombreuses soirées que j'ai pu faire avec ces personnes et qui m'ont permis de m'aérer l'esprit, j'aimerais citer Loïc, Mickaël, Nicolas, Jérome (Ben), Louloute, Marc et Savéria la corse, Benjamin et Karine, Jérôme, Thomas et Me-

[i] Le laboratoire n'est maintenant plus le même et se nomme le LITIS à présent.

riem la quebecoise, Greg, Mimi, Heeyoung la petite coréenne, et les nombreuses autres personnes avec qui j'ai passé de bons moments.

Je salue tous les chercheurs que j'ai pu rencontrer lors de conférences, ceux-ci étant trop nombreux, je ne peux les citer tous.

Je remercie également les developpeurs et contributeurs des distributions Mandriva (anciennement Mandrakesoft) et Debian, les utilisateurs LaTeX et Beamer, la société MathWorks et les contributeurs à la *Bayes Net Toolbox* et à sa liste de diffusion.

Je remercie les membres de mon jury pour avoir eu la patience de relire et corriger les versions préliminaires de ce manuscrit : Alexandre Aussem, Salem Benferhat, Stéphane Canu, Emmanuel Mazer, et plus particulièrement Jean-Yves Jaffray, mon ancien professeur, pour ses nombreuses corrections et précisions. Il me faut également encore remercier Philippe Leray, l'homme sans qui ce document n'aurait jamais vu le jour.

Et bien sûr, ceux sans qui je ne serais rien :
mes parents, mon frère, ma famille et tous mes amis d'enfance
qui me supportent et soutiennent depuis toujours.

Résumé

Durant ces travaux de thèse, une comparaison empirique de différentes techniques d'apprentissage de structure de réseaux bayésiens a été effectuée, car même s'il peut en exister très ponctuellement, il n'existe pas de comparaisons plus globales de ces algorithmes. De multiples phases de tests nous ont permis d'identifier quelles méthodes souffraient de difficultés d'initialisation et nous avons proposé une technique pour les résoudre. Nous avons ensuite adapté différentes méthodes d'apprentissage de structure aux bases de données incomplètes et avons notamment introduit une technique pour apprendre efficacement une structure arborescente. Cette méthode est ensuite adaptée à la problématique plus spécifique de la classification et permet d'apprendre efficacement et en toute généralité un classifieur de Bayes Naïf augmenté. Un formalisme original permettant de générer des bases de données incomplètes ayant des données manquantes vérifiant les hypothèses MCAR ou MAR est également introduit. De nombreuses bases synthétiques ou réelles ont alors été utilisées pour tester ces méthodes d'apprentissage de structure à partir de bases incomplètes.

Mots-clés: Réseaux Bayésiens, Apprentissage Statistique, Raisonnement Probabiliste, Classification, Aide à la Décision.

Abstract

We have performed an empirical study of various deterministic Bayesian networks structure learning algorithms. The first test step has allowed us to emphasise which learning technics need a specific initialisation et we have proposed a way to do this. In the second stage of this doctoral study, we have adapted some learning technics to incomplete datasets. Then, we have proposed an efficient algorithm to learn a tree-augmented naive Bayes classifier in a general way from an incomplete dataset. We have also introduced an original formalism to model incomplete dataset generation processes with MCAR or MAR assumptions. Finally, various synthetic datasets and real datasets have been used to empirically compare structure learning methods from incomplete datasets.

Keywords: Bayesian Networks, Statistical Learning, Probabilistic Reasoning, Classification, Decision Support.

Table des matières

Partie I : INTRODUCTION ET GÉNÉRALITÉS **1**

CHAPITRE 1 — **Avant-propos et problématique** **3**
1.1 Problématique 5
1.2 Pourquoi les réseaux bayésiens ? 7
1.3 Organisation du document 9

CHAPITRE 2 — **Les modèles graphiques probabilistes** **11**
2.1 Les différents supports graphiques 12
2.2 Les réseaux bayésiens 13
2.2.1 Les réseaux bayésiens multi-agents 16
2.2.2 Les réseaux bayésiens orientés objets 16
2.2.3 Les diagrammes d'influence 17
2.2.4 Les réseaux bayésiens dynamiques 17
2.2.4.1 Les chaînes de Markov 18
2.2.4.2 Les automates stochastiques à états finis 19
2.2.4.3 Les filtres bayésiens 19
2.2.4.4 Les processus de décision markoviens 20
2.2.4.5 Les réseaux bayésiens à temps continu 21
2.2.5 Les réseaux bayésiens multi-entités 21
2.2.6 Les structures adaptées à la classification 22
2.3 Les réseaux bayésiens causaux 23
2.3.1 Les modèles causaux markoviens 23
2.3.2 Les modèles causaux semi-markoviens 24
2.3.3 Les graphes ancestraux maximum 25
2.4 Les réseaux possibilistes 25
2.5 Les modèles non orientés 26
2.5.1 Les modèles de Gibbs ou champs de Markov 26
2.5.2 Les graphes factorisés 27
2.5.3 Les graphes cordaux 27
2.6 Les modèles semi-orientés 27
2.6.1 Les *Chain graphs* 27

2.6.2 Les réseaux bayésiens de niveau deux 28

CHAPITRE 3 — **Le cadre des modèles graphiques** **29**
3.1 Introduction 30
3.1.1 Condition de Markov 30
3.1.2 Fidélité 31
3.1.3 Graphoïdes 33
3.2 Critère de séparation directionnelle 33
3.2.1 *d*-séparation 34
3.2.2 *Minimal I-map* : carte d'indépendances minimale 35
3.2.3 *Maximal D-map* : carte de dépendances maximale 37
3.2.4 *P-map* : carte parfaite 38
3.3 Table de probabilités conditionnelles 38
3.4 Classes d'équivalence de Markov 39

CHAPITRE 4 — **Une utilisation des modèles probabilistes : l'inférence** **41**
4.1 Méthodes d'inférence exactes 42
4.1.1 Messages locaux 42
4.1.2 Ensemble de coupe 42
4.1.3 Arbre de jonction 42
4.1.4 Inversion d'arcs 44
4.1.5 Elimination de variables 44
4.1.6 Explication la plus probable 44
4.1.7 Méthodes symboliques 45
4.1.8 Méthodes différentielles 45
4.2 Méthodes d'inférence approchées 45
4.2.1 Simulation stochastique par Chaîne de Monte-Carlo 46
4.2.2 Méthodes variationnelles 47
4.2.3 Méthodes de recherche de masse 48
4.2.4 *Loopy belief propagation* 48
4.2.5 Simplification du réseau 48

CHAPITRE 5 — **Apprentissage des paramètres** **49**
5.1 Hypothèses pour l'apprentissage de paramètres 50
5.2 Apprentissage des paramètres avec données complètes 51
5.2.1 Sans *a priori* 51
5.2.2 Avec *a priori* de Dirichlet 53
5.3 Apprentissage des paramètres avec données incomplètes 54
5.3.1 Qu'entendons-nous par base incomplète 54
5.3.2 Estimation à partir d'une base d'exemples incomplète 56

Conclusion **59**

Partie II : Apprentissage de Structure à partir de données complètes **61**

Introduction **63**

Chapitre 6 — **Méthodes de recherche d'indépendances conditionnelles** **67**

6.1 Les tests statistiques . . . 68
6.1.1 Le test du χ^2 . . . 68
6.1.2 Le test du rapport de vraisemblance . . . 69
6.2 L'information mutuelle . . . 70
6.3 Les algorithmes PC et IC . . . 71
6.4 L'algorithme QFCI . . . 72
6.5 L'algorithme BNPC . . . 72
6.6 L'algorithme PMMS . . . 72
6.7 L'algorithme *Recursive Autonomy Identification* . . . 73
6.8 La recherche de motifs fréquents corrélés . . . 73
6.9 *The Grow-Shrink algorithm* . . . 73
6.10 Discussion . . . 74

Chapitre 7 — **Fonctions de score** **75**
7.1 Avant-propos . . . 76
7.1.1 Quand utiliser un score ? . . . 76
7.1.2 Dimension d'un réseau bayésien . . . 76
7.1.3 Principe du rasoir d'Occam . . . 77
7.1.4 Score décomposable . . . 77
7.1.5 Score équivalent . . . 78
7.2 La vraisemblance marginale : la mesure bayésienne . . . 78
7.2.1 Le score bayésien . . . 78
7.2.2 Les autres scores bayésiens . . . 79
7.3 Les principaux scores pondérés . . . 80
7.3.1 Approximation du score bayésien par Laplace . . . 80
7.3.2 Les scores basés sur des critères informatifs . . . 81
7.3.3 La longueur de description minimum . . . 83
7.4 Autres mesures adaptées à la classification . . . 84

Chapitre 8 — **Méthodes de recherche de structure à base de score** **87**
8.1 Les heuristiques sur l'espace de recherche . . . 88
8.1.1 Arbre de poids maximal . . . 88
8.1.2 Structure de Bayes Naïve augmentée . . . 88
8.1.2.1 Structure de Bayes naïve . . . 88
8.1.2.2 Structure de Bayes naïve augmentée . . . 88

8.1.2.3 Structure de Bayes naïve augmentée par un arbre . . . 89
8.1.2.4 Classifieur naïf augmenté par une forêt : FAN 89
8.1.3 Algorithme K2 89
8.1.3.1 Méthode générale 89
8.1.3.2 Deux propositions d'ordonnancement : K2+T et K2-T . . 90
8.2 Les heuristiques sur la méthode de recherche 91
8.2.1 Principe de la recherche gloutonne dans l'espace des DAG 91
8.2.2 Différentes initialisations 92
8.2.2.1 Initialisations classiques 92
8.2.2.2 Notre initialisation : GS+T 92
8.2.3 Méthode GES 93
8.2.4 Les méthodes incrémentales 95
8.2.5 Les méthodes mixtes 95
8.3 Les méthodes non-déterministes 95
8.3.1 Utilisation de MCMC 95
8.3.2 Utilisation d'algorithmes évolutionnaires 95
8.3.3 Autres heuristiques d'optimisation 96
8.4 Les méthodes mixtes 96
8.5 L'apprentissage de réseaux bayésiens hybrides 96

CHAPITRE 9 — **Expérimentations** **97**
9.1 Recherche d'une bonne structure 98
9.1.1 Algorithmes utilisés 98
9.1.2 Réseaux tests et techniques d'évaluation 98
9.1.3 Résultats et interprétations 101
9.1.3.1 Influence de la taille de la base 101
9.1.3.2 Distance à la distribution de base 104
9.1.3.3 Stabilité vis-à-vis de la base d'exemples 104
9.1.3.4 Reconnaissance des dépendances faibles 104
9.1.3.5 Choix du score 105
9.1.3.6 Choix de l'initialisation 106
9.1.3.7 Duel inter-méthodes 108
9.2 Recherche d'un bon classifieur 108
9.2.1 Algorithmes utilisés et techniques d'évaluation 108
9.2.2 Résultats et interprétations 110

Conclusion **115**

Partie III : Apprentissage de Structure à partir de données incomplètes **117**

Introduction **119**

Chapitre 10 — **Rappel sur l'algorithme EM** **123**
10.1 L'algorithme *Expectation-Maximisation* . . . 124
10.1.1 Introduction . . . 124
10.1.2 Développement de la méthode . . . 124
10.1.3 *Expectation-Maximisation* . . . 126
10.1.4 Convergence . . . 127
10.2 Les adaptations de l'algorithme EM . . . 127
10.2.1 EM généralisé . . . 127
10.2.2 EM pour la classification . . . 127
10.2.3 EM incrémental . . . 127
10.2.4 Monte-Carlo EM . . . 127
10.2.5 EM variationnel . . . 128

Chapitre 11 — **Fonction de scores avec données incomplètes** **129**
11.1 Adapter les scores pour les bases incomplètes . . . 130
11.2 Approximation d'un score avec des bases incomplètes . . . 131
11.2.1 L'approximation de *Cheeseman et Stutz* . . . 131
11.2.2 Méthode d'évaluation générique de score . . . 132
11.2.2.1 La méthode . . . 132
11.2.2.2 Exemple avec le score *BIC* . . . 132
11.2.2.3 Exemple avec le score *BD* . . . 133

Chapitre 12 — **Méthodes à base de score** **135**
12.1 Méthodes existantes . . . 136
12.1.1 *Structural Expectation-Maximisation* . . . 136
12.1.1.1 AMS-EM : *alternative model selection* . . . 136
12.1.1.2 Bayesian Structural EM . . . 137
12.1.2 Utilisation des MCMC . . . 138
12.1.3 *Hybrid Independence Test* . . . 138
12.2 Méthodes proposées . . . 138
12.2.1 MWST-EM . . . 138
12.2.2 Variantes pour les problèmes de classification . . . 140
12.2.2.1 Classifieur de Bayes naïf augmenté par un arbre . . . 140
12.2.2.2 Classifieur naïf augmenté par une forêt : FAN-EM . . . 140
12.2.2.3 Travaux connexes . . . 141
12.2.3 Une proposition d'initialisation pour SEM . . . 141
12.2.4 Passer à l'espace des équivalents de Markov . . . 142

CHAPITRE 13 — **Génération de données incomplètes** **143**
13.1 Introduction . . . 144
13.2 Échantillonnage à partir de réseaux bayésiens . . . 144
13.3 Génération aléatoire de structures de réseaux bayésiens . . . 145
13.4 Notations et hypothèses préliminaires . . . 146
13.4.1 Hypothèses pour les situations MCAR et MAR . . . 146
13.4.2 L'approche générale . . . 147
13.5 Notre modélisation des processus MCAR . . . 148
13.5.1 Le modèle . . . 148
13.5.2 Identification des paramètres . . . 149
13.5.3 Comment construire les β_i aléatoirement ? . . . 149
13.5.4 Un exemple simple . . . 149
13.5.5 Situation générale . . . 150
13.6 Notre modélisation des mécanismes MAR . . . 150
13.6.1 Le modèle . . . 150
13.6.2 Identification des paramètres . . . 151
13.7 Modélisations de processus de génération de données NMAR . . . 152
13.8 Extension possible . . . 153

CHAPITRE 14 — **Expérimentations** **155**
14.1 Recherche d'une bonne structure . . . 156
14.1.1 Algorithmes utilisés . . . 156
14.1.2 Réseaux tests et techniques d'évaluation . . . 156
14.1.3 Résultats et interprétations . . . 157
14.1.3.1 Etude des cas complets contre méthode EM . . . 157
14.1.3.2 Etude des cas disponibles contre méthode EM . . . 157
14.1.3.3 Stabilité en fonction de la taille de la base . . . 160
14.1.3.4 Stabilité en fonction du taux de données manquantes . 162
14.1.3.5 Duels entre les différentes méthodes gloutonnes . . . 162
14.2 Recherche d'un bon classifieur . . . 164
14.2.1 Techniques utilisées et techniques d'évaluation . . . 164
14.2.2 Résultats et interprétations . . . 164

Conclusion **167**

Partie IV : CONCLUSION ET PERSPECTIVES **169**

CHAPITRE 15 — **Conclusion** **171**

CHAPITRE 16 — **Perspectives** **173**

Références Bibliographiques **177**

ANNEXE A — **Raisonnement probabiliste** **193**
A.1 Différentes interprétations des probabilités . . . 193
A.2 Le paradoxe du changement de porte . . . 194
A.3 Hasard et probabilités . . . 196

ANNEXE B — **Notions de probabilités** **201**
B.1 Rappels de probabilités . . . 201
B.2 Indépendance conditionnelle . . . 202

ANNEXE C — **Notions de graphe** **205**
C.1 Graphes non-orientés . . . 205
C.2 Graphes orientés . . . 207

ANNEXE D — **Solutions logiciels** **209**
D.1 Différentes solutions existantes . . . 209
D.2 SLP : Le *Structure Learning Package* pour BNT . . . 210

ANNEXE E — **Bases d'exemples utilisées** **211**
E.1 Bases d'exemples complètes disponibles . . . 211
E.2 Bases d'exemples incomplètes disponibles . . . 213

ANNEXE F — **Tableaux de résultats** **215**
F.1 Expérimentations à partir de bases complètes générées . . . 215
F.2 Expérimentations à partir de bases complètes disponibles . . . 219
F.3 Expérimentations à partir de bases incomplètes disponibles . . . 222

Liste des figures **225**

Index **227**

Notations

Variables aléatoires

$X_1, \cdots X_n$ variables aléatoires génériques et noms de nœuds génériques d'un graphe.

X, Y, Z vecteurs aléatoires formés d'éléments de $X_1, \cdots X_n$ et sous ensembles de nœuds de $\{X_1, \cdots X_n\}$.

$x, \{X = x\}$ événement.

X_i^l variable aléatoire représentant la valeur du i-ième attribut dans le l-ième exemple de la base $\mathbf{D}$.

Probabilités et indépendances

$\mathbb{P}$ mesure de probabilité.

$\mathbb{P}(\cdot|\cdot)$ probabilité conditionnelle ($\mathbb{P}(X|Z)$).

$\cdot \perp\!\!\!\perp \cdot$ indépendance marginale.

$\cdot \perp\!\!\!\perp \cdot | \cdot$ indépendance conditionnelle ($X \perp\!\!\!\perp Y|Z$).

Graphes et séparation

$\mathcal{G}$ graphe sur l'ensemble de nœuds $X = \{X_1, \cdots X_n\}$.

$\cdot \perp \cdot | \cdot$ critère de séparation dans les graphes.

$\cdot \perp_d \cdot | \cdot$ critère de d-séparation dans les graphes ($X \perp_d Y|Z$).

$\mathcal{E}$ application de $X \times X$ vers $\{0, 1\}$ codant l'existence d'arcs et d'arêtes.

$Pa(X_i)$ vecteur aléatoire de l'ensemble des variables parentes du i-ième nœud.

$Enf(X_i)$ variable aléatoire de l'ensemble des variables parentes filles du i-ième nœud.

$Desc(X_i)$ variable aléatoire de l'ensemble des variables descendantes du i-ième nœud.

$F(X_i)$ frontière de Markov du nœud X_i.

$M(X_i)$ couverture de Markov du nœud X_i.

Paramètres

Θ caractère générique pour représenter l'ensemble des paramètres d'un réseau bayésien ($\Theta = \{\mathbb{P}(X_i = k|Pa(X_i) = j)\}_{i,j,k}$).

$\widehat{\Theta}$ caractère générique pour représenter l'ensemble des paramètres d'un réseau bayésien évalué par *maximum de vraisemblance* ou *maximum a posteriori*.

θ_i tableau à trois dimensions représentant l'ensemble des paramètres pour le nœud x_i.

θ_{ij} matrice de paramètres pour le nœud X_i lorsque ses parents sont dans leur j-ième configuration.

θ_{ijk} paramètre pour le nœud X_i lorsque ses parents sont dans leur j-ième configuration et qu'il est dans sa k-ième état ($\theta_{ijk} = \mathbb{P}(X_i = k|Pa(X_i) = j)$).

Données et instantiations

$\mathbf{D}_c$	base d'exemples complète.
$\mathbf{D}$	base d'exemples incomplète.
$\mathbf{O}$	partie observée de $\mathbf{D}$, sous ensemble de $\{X_i^l\}$.
$\mathbf{H}$	partie non-observée de $\mathbf{D}$, sous ensemble de $\{X_i^l\}$, complémentaire de $\mathbf{O}$.
n	nombre d'attributs dans la base $\mathbf{D}$.
N	nombre d'exemples dans la base $\mathbf{D}$.
q_i	nombre de configurations du vecteur aléatoire de l'ensemble des variables parentes du nœud X_i.
r_i	nombre de configurations de la variable aléatoire discrète X_i.
N_i	histogramme de la variable X_i pour la base $\mathbf{D}$.
N_{ij}	nombre d'exemples où le vecteur aléatoire $Pa(X_i)$ prend sa j-ième valeur dans la base $\mathbf{D}$.
N_{ijk}	nombre d'exemples où le vecteur aléatoire $Pa(X_i)$ prend sa j-ième valeur tandis que la variable X_i prend sa k-ième valeur dans la base $\mathbf{D}$.
x_i^l	observation du i-ième attribut dans le l-ième exemple de la base $\mathbf{D}$.
$pa(X_i)^l$	l-ième observation dans $\mathbf{D}$ de la variable $Pa(X_i)$.
$enf(X_i)^l$	l-ième observation dans $\mathbf{D}$ de la variable $Enf(X_i)$.
$desc(X_i)^l$	l-ième observation dans $\mathbf{D}$ de la variable $Desc(X_i)$.

Première partie

INTRODUCTION ET GÉNÉRALITÉS

1

Avant-propos et problématique

"*La théorie des probabilités n'est autre que le sens commun fait calcul.*"

Marquis Pierre-Simon de Laplace
né en 1749 à Beaumont-en-Auge, Normandie.

Sommaire

Qu'est-ce que le *Machine Learning* 4
Et l'intelligence artificielle dans tout ça ? 4
1.1 Problématique 5
Quels sont les avantages de la théorie des probabilités ? 5
Pourquoi ne pas utiliser la logique floue ? 6
Probabilités et modèles graphiques 7
1.2 Pourquoi les réseaux bayésiens ? 7
Comment réutiliser toutes ces données ? 7
Un réseau bayésien comme un système expert ? . . . 7
La formule d'inversion de Thomas Bayes 9
Pourquoi choisir des modèles orientés ? 9
1.3 Organisation du document 9

Qu'est-ce que le *Machine Learning* ?

Là où les mathématiques donnent un cadre pour savoir ce qu'*est*, ou *ce que représente* quelque chose, l'informatique donne un cadre pour savoir *comment est* cette même chose. Certains problèmes simples d'un point de vue mathématique peuvent être difficiles d'un point de vue de la modélisation ou de la complexité informatique (manipuler des nombres réels ou résoudre le problème du voyageur de commerce).

Il est possible de définir l'informatique en énumérant ce que cette discipline se propose de résoudre :

- identifier les problèmes (particulièrement ceux qui ne sont pas résolus, ou non résolus de manière satisfaisante par les mathématiques),
- trouver comment décomposer ces problèmes en sous-tâches traitables,
- puis résoudre ces sous-tâches.

Il s'agit donc de l'étude systématique de la faisabilité, et cette faisabilité est ensuite traduite en méthodes qui peuvent être automatisées *efficacement*. Le propos du *Machine Learning* est autre.

En informatique, l'expert donne une méthode pour résoudre un problème alors qu'en *Machine Learning*, l'expert donne une méthode pour 'apprendre' à résoudre (automatiquement) une classe de problèmes.

Un programme d'ordinateur qui peut apprendre par l'expérience pour faire une tâche particulière en respectant certains critères ! Quelle idée ambitieuse. Pourtant de nombreuses applications ont déjà vu le jour : reconnaissance de la parole, traitement d'image (détection ou suivi d'objet), recherche d'information textuelle (fouille de textes, modélisation du génome humain, détection de pourriels), aide au diagnostic, détection de fraude, prédictions météorologiques, *etc*.

Le *Machine Learning* est une manière originale et récente de voir les sciences : *les sciences nous aident à comprendre notre environnement, pourquoi ne joueraient-elles pas le même rôle pour les ordinateurs ?*

Certaines des matières du *Machine Learning* sont partagées avec la *philosophie des sciences* (Korb (2001)) comme l'étude des différences entre réalisme et instrumentation ou l'étude de la nature de la causalité. De plus, il est possible de voir une grande similarité entre le *meta-learning* et la recherche de méthodes scientifiques. Ces deux disciplines restent très différentes de par leur histoire et leurs traditions de recherche, mais le savoir qu'elles traitent est très similaire. Leur objectif principal étant d'**expliquer les relations qu'il peut y avoir entre la théorie et les observations**.

Que ces disciplines aient les mêmes soucis est un indicateur de chevauchement. Par ailleurs, les moyens mis en œuvre par ces deux disciplines sont différents puisque là où l'une veut *comprendre les sciences* et travaille dans le langage de celles-ci, l'autre veut *développer des techniques qui permettront aux machines d'apprendre* et utilise alors un formalisme aisé à programmer.

Et l'intelligence artificielle dans tout ça ?

Définissons très généralement un *robot* comme étant une machine devant traiter de l'information pour effectuer une tâche. Comment caractériser l'*intelligence* pour un robot ?

Le terme 'intelligence' regroupe principalement 5 notions : l'apprentissage, le raisonnement, la résolution de problèmes, la perception, la compréhension de langage. Les spécialistes en intelligence artificielle ont deux projets centraux :

(1) étudier la nature de l'intelligence (en essayant de la simuler),

(2) construire des machines qui sont capables

  (2a) d'interagir avec les humains, ou,

  (2b) de résoudre un problème de manière à ce qu'un humain ne puisse pas déceler si c'est une machine ou un humain qui a résolu la tâche.

L'intelligence artificielle (IA) est donc séparée entre trois disciplines, la simulation cognitive (1), l'IA dure (2a) et l'IA appliquée (2b).

L'*intelligence artificielle* interagit donc avec l'informatique et le *Machine Learning*. D'une part, un *robot* doit pouvoir percevoir son environnement, comprendre comment celui-ci interagit avec lui et réagir en conséquence (par le biais d'un raisonnement). D'autre part, un *robot* doit être capable de reconnaître son environnement pour s'y adapter rapidement.

1.1 Problématique

L'objectif de ce travail est de proposer des méthodes pour extraire *automatiquement* de la connaissance à partir de données statistiques. Ce travail s'intègre donc parfaitement dans le *Machine Learning*, puisqu'il s'agit ici, entre autre, d'identifier automatiquement un environnement. Cet environnement sera modélisé par la distribution de probabilités qui a généré les exemples. Une fois cet environnement reconnu, la machine sera alors capable de générer de nouvelles données indiscernables de celles créées par la nature, ou encore d'extraire de la connaissance du modèle obtenu.

Les applications de ce travail peuvent être multiples, par exemple si le but est de ***modéliser automatiquement le fonctionnement d'un système complexe*** ou d'approcher automatiquement le comportement d'un système chaotique, il est possible d'utiliser une base d'exemples de fonctionnement du système pour en extraire un modèle. Cette démarche peut alors avoir un intérêt dans une problématique d'*intelligence artificielle* puisqu'une fois qu'un robot aura acquis une connaissance sur son environnement grâce à ces méthodes, il pourra utiliser le modèle créé, pour interagir plus efficacement avec celui-ci.

La prise en compte de l'*incertitude* est un enjeu important dans le domaine de l'*intelligence artificielle* puisqu'il faut être capable de déterminer quelles sont les causes ou les conséquences les plus vraisemblables à partir d'un état particulier du système pour en déduire les meilleures actions possibles. Or, pour cela, l'usage de règles du type `'si ... alors ...'` est trop rigide et doit être modulé. De nombreux formalismes ont alors été créés pour cet objectif. Citons par exemple la théorie de l'évidence de Dempster-Shafer (1976), des concepts comme le flou introduit par Zadeh (1978) ou encore les réseaux probabilistes de Pearl (1988).

Avantages de la théorie des probabilités pour la modélisation :

Prenons l'exemple du corps humain, système complexe à volonté. Lorsqu'un individu est malade, nous allons observer pour celui-ci un certain nombre de grandeurs (tension, fièvre, etc). En fonction de ces différentes observations et de sa connaissance *a priori* le

médecin va donner son diagnostic. Ce diagnostic est ici évalué en fonction d'un nombre restreint de paramètres, or il se peut que deux individus aient la même forme (les mêmes valeurs pour toutes les grandeurs observées) et que l'un d'eux soit malade, tandis que l'autre est sain.

Les modèles probabilistes ont cet avantage de fournir systématiquement une probabilité à chaque état (ici *sain* ou *malade*). Celle-ci peut alors être considérée comme un indice de confiance dans le résultat (par exemple, cet individu a 85% de (mal)chance d'être malade). Bien sûr un tel diagnostic n'est pas satisfaisant d'un point de vue éthique pour décider d'administrer ou non un traitement. Pour ce type d'exemple, il est évident que n'importe quel système ayant ces propriétés servira essentiellement à aider le praticien dans sa prise de décision, par exemple, en le confortant sur son avis.

Prenons un autre exemple, celui de la détection de panne. Supposons que nous ayons un système qui ne fonctionne plus à son optimum car un de ses éléments est dégradé. Les décideurs ne peuvent pas se permettre que le système cesse de fonctionner, il leur faut donc savoir quelles pièces sont les plus vraisemblablement usées. S'ils disposent alors d'un modèle probabiliste dans lequel tous les éléments primordiaux du système et les fonctions que le système doit remplir sont représentés avec leurs interactions, de plus, s'ils savent quelle fonction du système n'est plus correctement effectuée, alors, en utilisant ce modèle, il leur sera possible de connaître quelles pièces sont le plus vraisemblablement abîmées. Ainsi, la maintenance sera plus efficace et moins coûteuse.

Nous allons nous concentrer sur les réseaux bayésiens que nous définirons en section 2.2. Ces outils permettent alors de résoudre le type de tâches énoncées précédemment. Seulement pour qu'une tâche soit résolue efficacement, il faut qu'un modèle soit correctement construit. Or, comme les systèmes sont gérés par un grand nombre de paramètres, nous ne pouvons demander à un homme seul, ou encore à une équipe de les construire. Nous nous proposons donc de développer certaines méthodes d'apprentissage automatique de ces modèles probabilistes à partir de connaissances expertes et d'un historique : une base d'exemples de fonctionnement et/ou de non fonctionnement du système ou de systèmes similaires.

Pourquoi ne pas utiliser la logique floue pour gérer l'incertain et l'imprécis :

La logique floue permet de modéliser l'imprécis avec des définitions vagues du genre "il fait plutôt chaud", "il est assez grand", *etc*. Mais ces définitions approximatives sont «certaines». Elle permet également de modéliser l'incertain avec une définition du genre "il fait 23 degrés", à laquelle on ajoute une grandeur appelée la 'possibilité'. Si cette possibilité n'est pas l'unité, alors d'autres évènements sont possibles et il y a une incertitude.

Par contre, une distribution de probabilités sur la température évaluée (par exemple une gaussienne autour de 23 degrés) représente l'imprécision sur la mesure, en même temps qu'elle peut coder notre incertitude. Cette confusion entre incertitude et imprécision pourrait être dommageable. Néanmoins, en pratique, il importe rarement de distinguer si une donnée est certaine et imprécise ou encore précise et incertaine. Les personnes intéressées par cette distinction pourront se référer à certains travaux permettant d'étendre les modèles que nous allons introduire au formalisme de la logique floue : les réseaux possibilistes, introduits dans Benferhat, Dubois, Kaci & Prade (2001) et Ben Amor, Benferhat & Mellouli (2002) ont alors un pouvoir expressif plus qualitatif.

Dans le formalisme des réseaux bayésiens, il est néanmoins possible d'introduire une variable supplémentaire par valeur observée pour représenter l'imprécision. Donc, au lieu d'avoir un seul nœud par grandeur, il y en aurait deux, un pour l'imprécision (capteur) sur la valeur de la variable et l'autre pour l'incertitude sur la valeur réelle.

Probabilités et modèles graphiques :
Nos travaux ne se concentrent pas exclusivement sur le formalisme des probabilités et sur les méthodes statistiques. En effet, pour pouvoir interpréter simplement les résultats formels et pour les expliquer aux non spécialistes, il faut pouvoir présenter ces résultats sous forme synthétique. Les techniques pour faire cela sont souvent graphiques (schémas, diagrammes en camembert, *etc*). Les modèles que nous allons introduire ont cet avantage d'être à la fois probabilistes et graphiques. La partie graphique de ces modèles permettra d'identifier simplement quels ensembles de variables sont dépendants ou conditionnellement indépendants. Ces modèles graphiques possèdent par ailleurs un grand pouvoir expressif et permettent d'effectuer des raisonnements probabilistes efficacement (cf. chapitre 4).

1.2 Pourquoi les réseaux bayésiens ?

Une des grandes problématiques de notre époque est de traiter la grande quantité des données qui est mise à notre disposition (notamment grâce à l'informatique) pour en extraire de l'information. Il serait donc intéressant d'avoir un (ou plusieurs) modèle(s) effectuant le lien entre les observations et la réalité pour un objectif précis, et cela, même lorsque les observations sont incomplètes et/ou imprécises.

Comment réutiliser toutes ces données ?
Imaginons un statisticien qui veut analyser un tableau de mesures pour une population donnée. *Il se retrouve face à une immense masse d'informations de laquelle il doit extraire de la connaissance !* Il va donc essayer de retrouver les relations pertinentes entre des variables ou des groupes de variables. L'utilisation des réseaux bayésiens va lui permettre d'obtenir une représentation compacte de ces ensembles de dépendances grâce à la notion de d-séparation (voir section 3.2) et des tables de probabilités conditionnelles (voir section 2.2), à partir de laquelle il lui sera plus simple de raisonner.

Les réseaux bayésiens permettent donc de transformer en modèle interprétable de la connaissance contenue dans des données.

Le modèle graphique probabiliste étant construit, il est alors possible d'effectuer des raisonnements à partir de ce modèle compact sans avoir à se référer à nouveau au tableau de données.

Un réseau bayésien comme un système expert ?
Un système expert à base de règles est souvent défini comme *une application capable d'effectuer des raisonnements logiques comparables à ceux que feraient des experts humains du domaine considéré. Il s'appuie sur des bases de données de faits et de connaissances, ainsi que sur un moteur d'inférence, lui permettant de réaliser des déductions logiques (chaînage avant et arrière).*

En pratique, un système expert modélise le mode de raisonnement de l'expert puis essaye de reproduire ce raisonnement sur de nouvelles requêtes (mode de raisonnement causal).

Or, un modèle probabiliste ne modélise pas le mode de raisonnement de l'expert mais la connaissance qualitative que l'expert a des phénomènes physiques influençant le système (mode de raisonnement fondé). Un tel modèle n'est donc pas un système expert au sens où est habituellement utilisé ce terme, les raisonnements effectués n'étant, par ailleurs, pas logiques, mais probabilistes.

Les réseaux probabilistes sont également une représentation du savoir incertain plus flexible que les systèmes à base de règles. Par exemple, en médecine, une même combinaison de symptômes peut être observée pour différentes maladies. Il n'y a donc pas toujours de règles strictes pour obtenir un diagnostic. Pour un système complexe, un expert humain est capable de porter un jugement même lorsque toutes les données nécessaires ne sont pas observées. Un système expert ne peut pas faire cela alors qu'un modèle probabiliste le permet.

De plus, les réseaux bayésiens sont plus facilement adaptés et mis à jour en fonction du contexte que les systèmes à base de règles. L'expérience montre qu'il est alors plus simple et rapide de créer des modèles graphiques. Ceux-ci étant très intuitifs, la communication avec les experts devient plus simple.

Les réseaux bayésiens permettent de modéliser la connaissance subjective d'un expert, mais ne modélisent pas le mode de raisonnement qu'effectue un expert.

Pour résumer :

- L'aspect graphique des modèles graphiques permet de ***représenter les relations entre les attributs clairement et intuitivement***.
- Leurs orientations (si elles existent) peuvent représenter des relations de cause à effet.
- Les modèles probabilistes sont capables de gérer l'***incertain*** et l'***imprécis***.

Le rôle des graphes dans les modèles probabilistes est triple :

- fournir un moyen simple et efficace d'exprimer des hypothèses (théorème 3.4.1),
- donner une représentation économique des fonctions de probabilité jointe (page 15),
- faciliter l'inférence à partir d'observations (chapitre 4).

Ces modèles deviennent incontournables lorsque nous avons affaire à un problème sujet à l'incertain.

Les modèles probabilistes sont alors utiles pour

- l'extraction de connaissance probabiliste : c'est-à-dire trouver quelles variables sont corrélées, dépendantes ou conditionnellement indépendantes,
- le diagnostic : l'évaluation de $\mathbb{P}(causes|symptômes)$,
- la prédiction : l'évaluation de $\mathbb{P}(symptômes|causes)$,
- la classification : le calcul de $\max_{classes} \mathbb{P}(classe|observations)$.

La formule d'inversion de Thomas Bayes (1764)

Le nom de réseau bayésien provient de la formule d'inversion de Bayes :

$$\mathbb{P}(\mathbf{H}|\mathbf{e}) = \mathbb{P}(\mathbf{e}|\mathbf{H}) \times \frac{\mathbb{P}(\mathbf{H})}{\mathbb{P}(\mathbf{e})} \tag{1.1}$$

pour toute hypothèse $\mathbf{H}$ et toute observation $\mathbf{e}$.
Cette formule peut alors être réécrite pour mettre en évidence une proportionnalité :

$$\textbf{Probabilité a posteriori} \propto \textbf{Vraisemblance} \times \textbf{Probabilité a priori} \tag{1.2}$$

Cette équation simple va être utilisée à tour de bras pour évaluer des probabilités pour des tâches de diagnostic, de prédiction, ou de discrimination.

Pourquoi choisir des modèles orientés ?

Le pouvoir expressif des modèles graphiques orientés, par rapport à ceux non dirigés, n'est ni meilleur, ni moins bon (voir section 2.1) du point de vue des dépendances conditionnelles mais ils peuvent se prêter à une interprétation causale.

Nous avons préféré étudier les modèles orientés car ils sont plus intuitifs et plus visuels. Les orientations guident sur la manière d'interpréter la structure, tout comme elles guident lors de l'inférence pour savoir comment faire circuler l'information.

Ces modèles sont plus simples à construire pour les experts d'un domaine précis qui raisonnent souvent par mécanisme de causes à effets. Il serait alors dommage de s'abstraire de cette information en leur demandant de construire des modèles non orientés.

Une introduction plus générale sur l'utilisation de la formule de Bayes et ses limitations, la relation que les personnes ont avec la notion de probabilité et les liens existants entre les notions de *Hasard*, de *Probabilités* et de *Causalité* figure en annexe A.

1.3 Organisation du document

Cette première partie introductive se poursuit par la description du cadre formel des modèles graphiques probabilistes à travers diverses notions comme la *fidélité* ou la définition de *carte d'indépendances*. Des rappels sur les définitions issues de la théorie des probabilités ou de la théorie des graphes complètent la définition de ce cadre et sont disponibles dans les annexes B et C. Ensuite, le chapitre 2 situe les principaux modèles graphiques probabilistes les uns par rapport aux autres. Cette partie s'achève sur une brève revue de différentes techniques d'inférence pour les réseaux bayésiens, puis de différentes techniques d'apprentissage des paramètres d'un réseau bayésien.

Dans la deuxième partie, nous proposons d'étudier l'efficacité des méthodes d'apprentissage de structure lorsque la base d'apprentissage est complètement observée. Pour répondre à cette question, nous utilisons une approche expérimentale qui consiste en deux études comparatives. La première consiste en une identification de la loi jointe en prennant en considération deux principaux critères (François & Leray (2004b)). Le premier critère est la divergence de Kullback-Leiber qui évalue la ressemblance entre les modèles obtenus par les différentes méthodes d'apprentissage et les modèles génératifs utilisés. Le second critère est le score *BIC* qui rend compte de la vraisemblance des modèles obtenus. La seconde étude compare le pouvoir discriminant des différents méthodes sur diverses tâches de classification (François & Leray (2004a)). Les résultats

obtenus dans le chapitre 9 permettent de confirmer que la méthode GES est la plus efficace pour identifier une structure représentant au mieux une distribution de probabilité lorsqu'aucune information *a priori* n'est apportée et que la base d'exemples est suffisamment grande. Par ailleurs, la méthode d'identification de structure arborescente MWST étant particulièrement stable, elle permet d'obtenir de bons résultats lorsque la taille de la base d'exemples est réduite. Lorsque le résultat de cette méthode est utilisé comme point de départ de techniques plus générales comme K2 ou un algorithme glouton, elle permet d'augmenter significativement la précision et la stabilité des résultats obtenus par ces derniers algorithmes.

La troisième partie est une extension de la deuxième partie à l'étude de la précision des méthodes d'apprentissage de structure lorsque la base d'apprentissage est partiellement observée. Nous avons alors voulu effectuer une comparaison robuste, peu de bases d'exemples incomplètes étant effectivement librement disponibles en ligne. Pour ce faire, nous avons proposé une modélisation originale des différents processus de génération d'exemples incomplets utilisant le formalisme des réseaux bayésiens. L'échantillonneur obtenu permet alors de générer une grande variété de bases d'exemples incomplètes. Une comparaison expérimentale de différents algorithmes d'inférence de structure a alors été effectuée et nous permet de dire dans quels cas il est préférable d'utiliser une technique d'apprentissage utilisant un algorithme EM ou une technique basée sur l'étude des cas disponibles et dans quels cas il est préférable d'utiliser un algorithme général comme SEM ou la technique que nous avons proposée et qui permet d'inférer efficacement une structure arborescente (François & Leray (2005b) en français et en anglais dans Leray & François (2005)). Nous avons ensuite proposé une méthode permettant d'apprendre efficacement un classifieur de Bayes Naïf Augmenté par un arbre à partir de bases d'exemples incomplètes permettant d'obtenir de très bons résultats en classification (François & Leray (2006b))

2

Les modèles graphiques probabilistes

"The reasonable man adapts himself to the world ;
the unreasonable man persists in trying to adapt the world to himself.
Therefore, all progress depends on the unreasonable man."

George Bernard Shaw (1856 - 1950)

Sommaire

2.1 Les différents supports graphiques 12
2.2 Les réseaux bayésiens 13
2.2.1 Les réseaux bayésiens multi-agents 16
2.2.2 Les réseaux bayésiens orientés objets 16
2.2.3 Les diagrammes d'influence 17
2.2.4 Les réseaux bayésiens dynamiques 17
2.2.4.1 Les chaînes de Markov 18
2.2.4.2 Les automates stochastiques à états finis 19
2.2.4.3 Les filtres bayésiens 19
2.2.4.4 Les processus de décision markoviens 20
2.2.4.5 Les réseaux bayésiens à temps continu 21
2.2.5 Les réseaux bayésiens multi-entités 21
2.2.6 Les structures adaptées à la classification 22
2.3 Les réseaux bayésiens causaux 23
2.3.1 Les modèles causaux markoviens 23
2.3.2 Les modèles causaux semi-markoviens 24
2.3.3 Les graphes ancestraux maximum 25
2.4 Les réseaux possibilistes 25
2.5 Les modèles non orientés 26
2.5.1 Les modèles de Gibbs ou champs de Markov 26
2.5.2 Les graphes factorisés 27
2.5.3 Les graphes cordaux 27
2.6 Les modèles semi-orientés 27
2.6.1 Les *Chain graphs* 27
2.6.2 Les réseaux bayésiens de niveau deux 28

Il existe différents formalismes de modèles graphiques probabilistes. Certains sont très génériques et ont alors un grand pouvoir expressif. Mais leurs méthodes d'apprentissage ou d'inférence sont très complexes. D'autres modèles, moins généraux, sont réservés à des applications plus spécifiques, mais leur apprentissage et leur inférence peuvent être effectués plus efficacement.

2.1 Les différents supports graphiques

Les supports graphiques des modèles graphiques probabilistes peuvent être de plusieurs types, qui ont alors des pouvoirs expressifs différents.

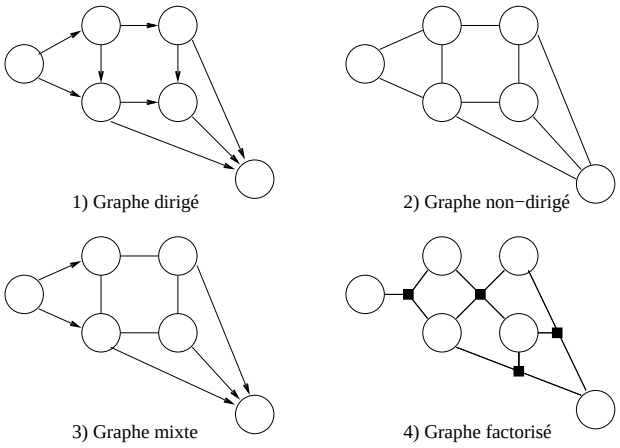


FIG. 2.1 : *Quatre types de modèles graphiques*

Le pouvoir expressif de ces modèles est différent.

Les graphes dirigés sans circuit :
Ils peuvent, grâce à leurs V-structures (voir figure 2.2) modéliser des variables qui sont indépendantes, mais qui deviennent dépendantes étant donné l'état d'une troisième variable (voir section 2.2).

Les graphes non dirigés :
Ils ne peuvent pas modéliser le cas précédent, par contre ils peuvent modéliser une loi qui vérifierait $A \perp\!\!\!\perp C|\{B,D\}$ et $B \perp\!\!\!\perp D|\{A,C\}$ grâce à la structure de la figure 2.3. Or un graphe orienté ne peut pas modéliser ces deux indépendances conditionnelles en même temps (voir section 2.5.1).

Les graphes mixtes (ou *chain graphs*) :
Ils ont l'avantage de cumuler les avantages des graphes dirigés et non dirigés (voir section 2.6.1).

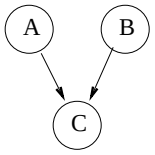


FIG. 2.2 : *La structure en V, l'avantage des modèles dirigés, peut coder simultanément l'indépendance $A \perp\!\!\!\perp B$ et la dépendance $A \not\perp\!\!\!\perp B|C$.*

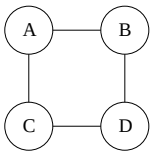


FIG. 2.3 : *Le carré, l'avantage des modèles non dirigés, peut coder simultanément les indépendances* $A \perp\!\!\!\perp C|\{B,D\}$ *et* $B \perp\!\!\!\perp D|\{A,C\}$.

Les graphes factorisés :
Ils sont une autre manière de modéliser les graphes non dirigés. Ils partagent alors le même pouvoir expressif (voir section 2.5.2).

2.2 Les réseaux bayésiens

Les réseaux bayésiens ont été nommés ainsi par Judea Pearl (1985) pour mettre en évidence trois aspects :
- la nature subjective des informations,
- l'utilisation de la règle de Bayes comme principe de base pour la mise à jour des informations,
- la distinction entre les modes de raisonnement causal et fondé (Bayes (1764), voir page 8).

Les approches fondées sur le conditionnement et la formule d'inversion de Bayes sont appelées approches bayésiennes. Le nom de *réseaux bayésiens* est donné par cohérence avec cette appellation. Remarquez que cette appellation n'a rien à voir avec le fait qu'avec les réseaux bayésiens nous soyons obligés d'utiliser des statistiques bayésiennes. Ceci est faux, et par exemple, il est possible d'utiliser des *réseaux bayésiens fréquentistes*. Pour éviter les erreurs, les noms suivants existent également pour ces modèles : les réseaux de croyance (*belief networks*), les modèles graphiques probabilistes orientés (*probabilistic oriented graphical models*), les réseaux probabilistes (*probabilistic networks*), ou encore les réseaux d'indépendances probabilistes.

Ces modèles ainsi que leurs méthodes d'inférence ont été introduits dans Pearl (1988), Lauritzen & Spiegelhalter (1988), Jensen (1996), Jordan (1998), Kim & Pearl (1987) et en français dans Naïm, Wuillemin, Leray, Pourret & Becker (2004).

Définition 2.2.1 (réseaux bayésiens) *Un* `réseau bayésien` $\mathscr{B} = \{\mathcal{G}, \mathbb{P}\}$ *est défini par*
- *un graphe dirigé sans circuit* $\mathscr{G} = (X, E)$ *où* X *est l'ensemble des nœuds (ou sommets) et où* E *est l'ensemble des arcs,*
- *un espace probabilisé* $(\Omega, \mathbb{P})$,
- *un ensemble de variables aléatoires* $X = \{X_1, \cdots, X_n\}$ *associées aux nœuds du graphe et définie sur* $(\Omega, \mathbb{P})$ *telles que* $\mathbb{P}(X_1, \cdots, X_n) = \prod_{i=1}^{n} \mathbb{P}(X_i | Pa(X_i))$ *où* $Pa(X_i)$ *est l'ensemble des parents du nœud* X_i *dans* $\mathscr{G}$.

Lors de nos expérimentations, nous allons nous concentrer sur le cas discret, la définition peut alors être simplifiée.

Définition 2.2.2 (réseaux bayésiens discrets finis) *$\mathscr{B} = (\mathscr{G}, \theta)$ est un `réseau bayésien discret fini` si $\mathscr{G} = (X, E)$ est un graphe dirigé sans circuit dont les sommets représentent un ensemble de variables aléatoires $X = \{X_1, \cdots, X_n\}$, et si $\theta_i = [\mathbb{P}(X_i|Pa(X_i))]$ est la matrice des probabilités conditionnelles du nœud i connaissant l'état de ses parents $Pa(X_i)$ dans $\mathscr{G}$.*

Définition 2.2.3 (paramètres) *Un paramètre θ_i est un tableau contenant l'ensemble des probabilités de la variable X_i pour chacune de ses valeurs possibles sachant chacune des valeurs prises par l'ensemble de ses parents $Pa(X_i)$.*
En particulier, nous noterons θ_{ijk} la probabilité $\mathbb{P}(X_i = k|Pa(X_i) = j)$[i].
Nous noterons $\theta_{ij} = (\theta_{ij2}, \cdots, \theta_{ijr_i})$, les paramètres à évaluer, avec de plus $\theta_{ij1} = 1 - \sum_{k=2}^{r_i} \theta_{ijk}$.

La définition des parents markoviens, c'est à dire l'ensemble $Pa(X_i) \subset \{X_j\}_{j \neq i}$ de variables aléatoires tel que $\mathbb{P}(X_i|Pa(X_i)) = \mathbb{P}(X_i|\{X_j\}_{j\neq i})$ pose la base théorique de la notion d'*influence relative entre les variables* qui subordonne l'incertitude de l'état de certains attributs à l'effet d'une entrée (information sur d'autres attributs). Les réseaux bayésiens ont un pouvoir expressif suffisant pour représenter différents modes de dépendances entre les variables, qu'elles soient causales, hiérarchiques, temporelles, *etc*.

Le nombre de DAG pour représenter toutes les structures possibles pour les réseaux bayésiens est de taille super-exponentielle. En effet, Robinson (1977) a prouvé qu'il était possible de donner ce nombre grâce à la formule récursive de l'équation 2.1.

$$r(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-1)} r(n-i) = n^{2^{\mathcal{O}(n)}} \tag{2.1}$$

Compatibilité de Markov :
Si une fonction de probabilité $\mathbb{P}$ admet une factorisation sous la forme de l'équation 2.2 relativement à un graphe dirigé sans circuit $\mathscr{G}$, on dira que $\mathscr{G}$ représente $\mathbb{P}$ et que $\mathbb{P}$ est *compatible* (ou est *Markov relatif*) à $\mathscr{G}$ (voir section 3.1.1 pour plus de détails)

En modélisation statistique, il est important de *supposer* la compatibilité entre des graphes dirigés sans circuit et des lois de probabilités lorsqu'un processus stochastique est supposé génératif de données, et qu'il soit *possible* d'identifier la loi jointe sous-jacente.

Un réseau bayésien $\mathscr{B} = \{\mathcal{G}, \mathbb{P}\}$ représente une distribution de probabilité sur X dont la loi jointe peut se simplifier de la manière suivante :

$$\mathbb{P}(X_1, X_2, \cdots, X_n) = \prod_{i=1}^{n} \mathbb{P}(X_i|Pa(X_i)) \tag{2.2}$$

Cette décomposition de la loi jointe permet de faire des réseaux bayésiens un modèle économique pour représenter des distributions de probabilités. Par exemple, supposons que

[i] Nous noterons par la suite $\mathbb{P}(X_i = k|Pa(X_i) = j)$ pour $\mathbb{P}(X_i = x_k|Pa(X_i) = pa_{ij})$, la probabilité où X_i prend sa k-ième valeur possible tandis que l'ensemble de variables $Pa(X_i)$ prend sa j-ième valeur possible (qui est un n-uplet).

nous ayons 20 variables ayant trois états et deux parents chacune (en moyenne). Nous avons alors 27 valeurs réelles à spécifier pour un paramètre (avec 2 parents de taille 3), dont seulement 18 indépendantes, car les 9 dernières seront données par normalisation. Or, nous avons vingt variables, d'où 360 valeurs à évaluer et à stocker. Avec des flottants en double précision, la mémoire nécessaire serait d'environ $2,8Ko$. Si nous voulions modéliser ce problème en représentant complètement la distribution de probabilité (donc sans passer par un réseau bayésien), le nombre de valeurs numériques à spécifier serait de $3 + 3 \times 3 + 3 \times 3 \times 3 + \cdots + 3^{20}$ (d'après le théorème de Bayes généralisé, équation B.1.2), soit $5,2$ milliards de valeurs. La mémoire nécessaire serait d'environ $39Go$ en double précision, soit 14000 fois plus d'espace mémoire sur ce petit exemple !

Cette décomposition de la loi jointe permet également d'avoir des algorithmes d'inférence puissants qui font des réseaux bayésiens des outils de modélisation et de raisonnement très pratiques lorsque les situations sont incertaines ou les données incomplètes. Ils sont alors utiles pour les problèmes de classification lorsque les interactions entre les différentes variables peuvent être modélisées par des relations de probabilités conditionnelles.

Qu'est-ce qui fait le succès des réseaux bayésiens ?

D'un point de vue de l'interprétation bayésienne des probabilités, un agent peut ajuster la structure et les probabilités conditionnelles des réseaux bayésiens (*paramètres*) de manière à maximiser l'entropie (ou la vraisemblance du modèle) par rapport à la connaissance *a priori* (par exemple présent dans une base de données). Ainsi, si la *compatibilité de Markov* n'est pas satisfaite, un réseau bayésien ne peut pas représenter correctement la distribution de probabilité qui a engendré les observations. Néanmoins, un réseau bayésien reste une bonne solution à adopter pour la modélisation d'un système lorsque la connaissance *a priori* que nous possèdons sur ce dernier satisfait la *compatibilité de Markov*.

Par ailleurs, il existe plusieurs déclinaisons des réseaux bayésiens.

Les *réseaux bayésiens multi-agents* (section 2.2.1), sont utiles lorsque les informations ne sont disponibles que localement, et que pour diverses raisons (de confidentialité par exemple), les différents agents ne veulent pas partager leurs informations. Dans ce cas, il leur sera tout de même possible d'utiliser leurs informations respectives sans pour autant les divulguer. Le résultat d'une inférence dans ce réseau particulier satisfera alors les différents partis.

Les *réseaux bayésiens de niveau deux* (section 2.6.2), sont utiles pour avoir une visualisation plus concise et donc plus lisible des relations de dépendance entre les attributs.

Les *réseaux bayésiens orientés objets* (section 2.2.2), sont utiles lorsqu'une sous structure est répétée à plusieurs endroits du réseaux global, cela permet d'avoir une représentation plus économique et plus lisible, en particulier si le réseau global est constitué d'une repétition d'une sous-structure particulière. Par exemple, cette représentation peut être utile pour les réseaux bayésiens dynamiques.

Les *diagrammes d'influence* sont une extension des réseaux bayésiens qui introduit des nœuds de nouvelle nature liés à la problématique de l'aide à la décision.

Les *réseaux bayésiens dynamiques* (section 2.2.4), sont utiles pour modéliser des phénomènes dynamiques ou temporels, en utilisant un temps discret.

Les *réseaux bayésiens continus* (section 2.2.4.5), sont utiles pour modéliser des phénomènes temporels lorsque le temps est continu.

Les *filtres bayésiens* (section 2.5) sont des réseaux bayésiens dynamiques particuliers, ils ne possèdent qu'une variable d'état et une variable d'observation. Leurs variantes appellées les filtres de Kalman ont été très utilisées pour des problématiques où le nombre d'états n'était pas discret.

Les *processus de décision markoviens* (section 2.2.4.4), sont une extension des modéles dynamiques et des diagrammes d'influence, ils permettent de traiter des problèmes liés à la décision tout en prenant en compte un aspect temporel.

Les *réseaux bayésiens causaux*, également appelés les modèles markoviens (section 2.3.1), sont utiles si l'on veut s'assurer que toutes les dépendances codées ont une signification réelle. Ils ne doivent donc pas être utilisés comme de simples outils de calcul comme il serait possible de le faire avec un réseau bayésien classique, il faut alors s'appuyer sur leur pouvoir expressif.

Les *réseaux bayésiens multi-entités* (section 2.2.5), sont une extention des réseaux bayésiens et de la logique bayésienne du premier ordre qui permet d'utiliser plusieurs 'petits' réseaux bayésiens pour modéliser un système complexe. Ils contiennent des variables contextuelles, qui peuvent être de différentes natures, notamment être liés à des notions de décision.

2.2.1 Les réseaux bayésiens multi-agents

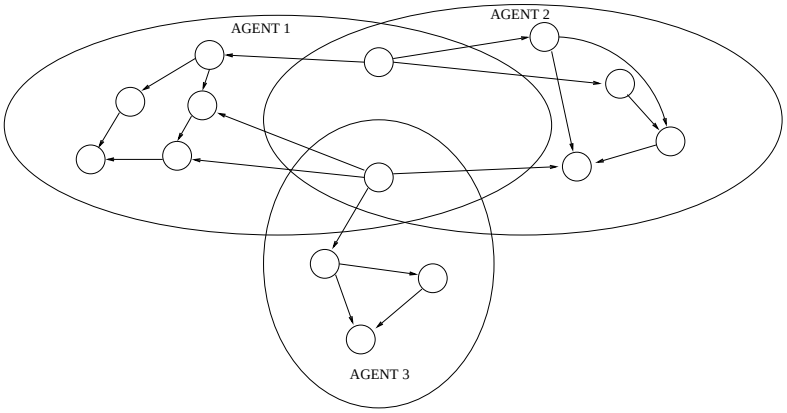


FIG. 2.4 : *Un exemple de RBMA*

Pour des raisons de sécurité, un réseau bayésien peut être partagé par plusieurs agents (ordinateurs, entreprises, etc). Chaque agent ne connaît qu'une partie de la structure et peut y faire de l'inférence localement. Une inférence globale peut être effectuée Xiang (2002) et Maes, Meganck & Manderick (2005) et les différents agents profiteront alors de certains résultats obtenus. Un exemple de réseau bayésien multi-agents (appelé également *Multi-Sectioned Bayesian Network*) est illustré sur la figure 2.4.

2.2.2 Les réseaux bayésiens orientés objets

Lorsque, dans un réseau bayésien, une sous-structure (avec ses paramètres) apparaît de manière répétée, Il est possible de représenter ce dernier à l'aide d'un réseau bayésien dit *orienté objet*. Il existe plusieurs types d'implémentations pour ces modèles. Se référer, par exemple, à Koller & Pfeffer (1997) ou à Bangsø & Wuillemin (2000).

Ces modèles sont particulièrement bien adaptés pour représenter les réseaux bayésiens dynamiques ou encore les réseaux bayésiens multi-agents. Ils permettent également de modéliser des systèmes complexes pour lesquels le même mode de raisonnement apparaît dans différents sous-systèmes.

2.2.3 Les diagrammes d'influence

Les réseaux bayésiens constituent des outils efficaces pour effectuer des tâches d'aide à la décision dans le sens où ils permettent d'évaluer différentes probabilités en fonction de l'état connu du système. Cependant, ils ne permettent pas de modéliser naturellement un système sur lequel l'opérateur peut agir. Il est alors possible d'étendre leur formalisme en introduisant de nouveaux types de nœuds : les *nœuds utilité* et les *nœuds décision*. En effet, les décideurs souhaitent souvent associer une (valeur d') utilité à chaque décision possible. Cette utilité représente la qualité ou encore le coût lié à ces décisions.

Ces modèles, appelés les *diagrammes d'influence*, ont initialement été introduits comme étant une représentation compacte des *graphes de décision* par Howard & Matheson (1981) et Shachter & Kenley (1989) et Jensen (2001). Ils sont à présent plutôt vus comme des extensions des réseaux bayésiens.

Définition 2.2.4 *Un diagramme d'influence est constitué d'un graphe orienté sans circuit contenant des nœuds probabilistes, des nœuds d'utilité et des nœuds de décision vérifiant les conditions structurelles suivantes :*

- *il existe un chemin passant par tous les nœuds décision,*
- *les nœuds utilité n'ont pas d'enfants.*

Nous demanderons de plus les conditions paramétriques suivantes :

- *les nœuds décision et les nœuds probabilistes ont un nombre fini d'états mutuellement exclusifs,*
- *les nœuds d'utilité n'ont pas d'état, il leur est attachée une fonction réelle définie sur l'ensemble des configurations de leurs parents.*

Habituellement, pour différencier les différents types de nœuds, les nœuds probabilistes sont représentés par des cercles, les nœuds décision par des carrés et les nœuds utilité par des losanges comme sur l'exemple de la figure 2.5.

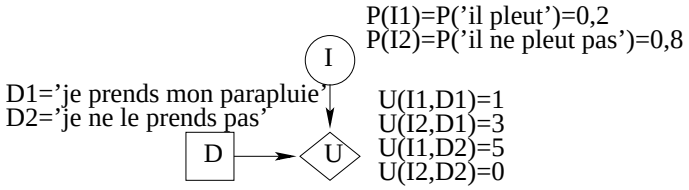


FIG. 2.5 : *Dans cet exemple de diagramme d'influence, l'objectif est de minimiser l'espérance d'utilité* U *en prenant la bonne décision* D *en fonction de l'état du nœud aléatoire* I.

2.2.4 Les réseaux bayésiens dynamiques

Dans de nombreux problèmes réels, le temps peut être une quantité qu'il est important de prendre en compte. Les réseaux bayésiens dynamiques sont alors à la fois une extension des réseaux bayésiens dans laquelle l'évolution temporelle des variables est

représentée (Dean & Kanazawa (1989), Murphy (2002), Bellot (2002), Binder, Murphy & Russell (1997) et Kanazawa, Koller & Russell (1995)).

Parfois, les réseaux bayésiens dynamiques, lorsqu'ils sont très inspirés de chaînes de Markov (*cachées*), et qu'ils ne font intervenir que des variables discrètes, sont encore appelés des *Modèles de Markov* (*cachés*) (Sahani (1999) et Dugad & Desai (1996)). Un exemple de modèle de Markov caché représenté sous forme de réseau bayésien dynamique est visible sur la figure 2.6.

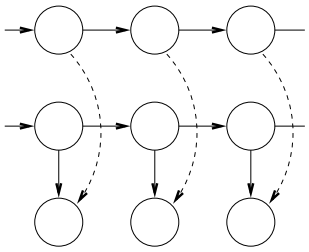


FIG. 2.6 : *Un exemple de modèle de Markov caché représenté sous forme de réseau bayésien dynamique. Ici, il s'agit d'un* HMM *factorisé à deux chaînes.*

Il est possible de se referer à Murphy (2002) pour voir différents types de réseaux bayésiens dynamiques remarquables. Un grand nombre d'applications ont utilisé ce type de modèles, par exemple en reconnaissance de la parole dans Bach & Jordan (2005).

Takikawa, d'Ambrosio & Wright (2002) ont également introduit des modèles appellés les réseaux bayésiens *partiellement dynamiques*, qui, comme leur nom l'indique, peuvent contenir à la fois des nœuds statiques et des nœuds dynamiques.

Commençons par étudier deux exemples simples de réseaux bayésiens dynamiques qui ne possèdent qu'une seule variable qui évolue au cours du temps.

2.2.4.1 Les chaînes de Markov

Dans le cadre d'un processus de Markov discret, le modèle le plus simple est la chaîne de Markov, encore appelée chaîne de Monte-Carlo. Soit X_t, la variable d'état au temps t, qui sont également les variables d'observations. X_t peut prendre ses valeurs parmi un ensemble d'états discrets, $\{1, \cdots, N_e\}$. Pour une description probabiliste complète, il faudrait prendre en considération tous les états précédant le temps t pour évaluer la probabilité d'un état au temps t. Lorsque nous modélisons un processus par une chaîne de Markov, nous supposons que celle-ci ne dépend que de l'état précédent. Nous avons donc la simplification de la loi jointe suivante :

$$\mathbb{P}(X_t = S_i | X_{t-1} = S_j, X_{t-2} = S_k, \cdots) = \mathbb{P}(X_t = S_i | X_{t-1} = S_j) \tag{2.3}$$

Un tel processus peut être représenté par un schéma comme celui de la figure 2.7, dans le cas particulier ou $N_e = 3$.

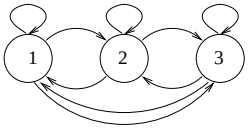


FIG. 2.7 : *Un exemple générique de chaîne de Markov à trois états représentée dans l'espace des états de X. Le modèle de transition représente les probabilités qui sont associées à chaque arc (non indiquées ici).*

Or au vu de la simplification de la loi jointe de l'équation 2.3, il est également possible de représenter une chaîne de Markov dans le formalisme des réseaux bayésiens dynamiques comme indiqué sur la figure 2.8.

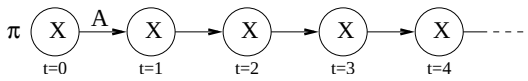


FIG. 2.8 : *Un réseau bayésien dynamique associé à la chaîne de Markov de la figure 2.7.*

Il est possible d'imaginer une chaîne de Markov d'ordre 2, c'est-à-dire pour laquelle l'état à l'instant t dépendrait des états aux instants $t-1$ et $t-2$. Seulement, ce type de chaînes peut alors être vu comme une chaîne de Markov classique (d'ordre 1) ou nous considérons le vecteur aléatoire $Y_t = (X_{t-1}, X_t)$ comme variable aléatoire principale[ii] de cette chaîne.

2.2.4.2 Les automates stochastiques à états finis

Pour une chaîne de Markov, le modèle de transition (A) est constant. Néanmoins, lorsqu'il change en fonction du temps, les modèles mis en jeu conservent parfois la même appellation de chaînes de Markov[iii]. Mais lorsque le modèle de transition change en fonction de l'état au temps t, ces modèles s'appellent alors des *automates stochastiques à états finis* ou peuvent encore porter le nom de *processus de Markov continus* (Dembo & Zeitouni (1986) et Asmussen, Nerman & Olsson (1996)).

Définition 2.2.5 *Un automate stochastique à états finis X_t pour l'espace d'états $X = \{x_1, \ldots, x_n\}$ est défini par une distribution de probabilité initiale P_X^0 et par une matrice dite d'*intensités de transition *définie comme suit :*

$$A_X = \begin{bmatrix} q_{x_1} & q_{x_1x_2} & \cdots & q_{x_1x_n} \\ q_{x_2x_1} & q_{x_2} & \cdots & q_{x_2x_n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{x_nx_1} & q_{x_nx_2} & \cdots & q_{x_n} \end{bmatrix} \tag{2.4}$$

où $q_{x_ix_j}$ est l'intensité de la transition de l'état x_i vers l'état x_j et où $q_{x_i} = \sum_{j \neq i} q_{x_ix_j}$. L'intensité q_{x_i} donne la probabilité 'instantanée' de rester dans l'état x_i, tandis que l'intensité $q_{x_ix_j}$ donne la probabilité de passer de l'état x_i à l'état x_j.

Pour ces modèles, nous avons les équivalences suivantes :
$\mathbb{P}(X_{t+\Delta t} = x_j | X_t = x_i) \underset{\Delta t \to 0}{\approx} q_{x_ix_j}\Delta t$, pour $i \neq j$, et $\mathbb{P}(X_{t+\Delta t} = x_i | X_t = x_i) \underset{\Delta t \to 0}{\approx} 1 - q_{x_i}\Delta t$.

2.2.4.3 Les filtres bayésiens

Les *filtres bayésiens* sont alors plus adaptés à l'expérimentation car en pratique, la valeur que nous observons pour une grandeur est souvent biaisée, ne serait-ce que par la précision du capteur. Un filtre bayésien peut être représenté par un réseau bayésien dynamique (voir paragraphe 2.2.4) qui ne possède qu'une variable d'état et qu'une variable observation. Soit X_t la variable d'état au temps t et O_t celle d'observation au temps t.

[ii] Dans ce cas, il existera des états de Y_t à probabilité nulle même si $\mathbb{P}$ est non nulle sur les X_t.

[iii] La terminologie commence à accepter l'expression chaînes de Markov quand le modèle de transition n'est pas constant.

Alors pour un filtre bayésien nous supposons que nous avons la décomposition de la loi jointe de la formule 2.5 à chaque instant t.

$$\mathbb{P}(X_1, \cdots, X_t, O_t) = \mathbb{P}(X_1)\big[\prod_{i=2}^{t} \mathbb{P}(X_i|X_{i-1})\big]\mathbb{P}(O_t|X_t) \tag{2.5}$$

où le terme $\mathbb{P}(X_1)$ est l'*a priori* initial (souvent noté π), $\mathbb{P}(X_i|X_{i-1})$ (souvent noté A), le *modèle de transition* et $\mathbb{P}(O_t|X_t)$ (souvent noté B), le *modèle d'observation*.

Les filtres bayésiens sont limités aux systèmes qui peuvent être modélisés par une formule similaire à celle de l'équation 2.5. Les réseaux bayésiens permettent alors de modéliser des lois plus complexes mais leurs méthodes d'apprentissage (plus génériques) sont également plus complexes.

Etudions à présent les deux filtres bayésiens les plus utilisés.

Le filtre de Kalman Lorsque, pour un modèle de Markov, l'*a priori* initial est gaussien, les variables d'états et d'observations sont continues et les modèles de transition et de d'observation sont gaussiens, ce modèle est appelé un *filtre de Kalman* (ou encore *systèmes dynamiques linéaires*, Maybeck (1979) et Anderson & Moore (1979)).

La chaîne de Markov cachée Le terme 'cachés' de l'appellation de ces modèles vient du fait que les variables d'intérêt, les X_t, sont non-observables. Nous observons donc une fonction probabiliste de l'état courant. Nous sommes donc en présence d'un processus doublement stochastique, et nous n'observons qu'un autre ensemble de variables, les O_t, qui ne dépendent que des états X_t inconnus Rabiner & Juang (1986).

Ces modèles peuvent être mis sous forme de réseaux bayésiens (Hamilton (1994) et Fine, Singer & Tishby (1998) et Gales (1999)). Par exemple, pour la chaîne de Markov cachée de la figure 2.7, pour laquelle nous n'observons pas les états, il est possible de construire le réseau bayésien dynamique de la figure 2.9.

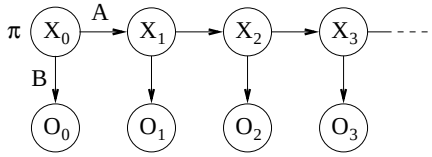


FIG. 2.9 : *Un réseau bayésien dynamique représentant une chaîne de Markov cachée. Seules les variables O_t sont observées.*

Pour ces modèles, certaines techniques *ad hoc* d'apprentissage et d'inférence ont été proposées, Rabiner (1989) en propose une synthèse. Il est d'ailleurs possible de faire le parallèle (voir la table 2.1) entre les méthodes d'inférence, les méthodes d'apprentissage et les méthodes de recherche de l'état le plus vraisemblable proposées pour les Chaînes de Markov cachés, et celles proposées pour les réseaux bayésiens (Murphy (2002)).

2.2.4.4 Les processus de décision markoviens

Comme avec les diagrammes d'influence, qui permettent de mêler les réseaux bayésiens avec une notion d'*utilité*, les *processus de décision markoviens* permettent de mêler l'aspect temporel des réseaux bayésiens dynamiques avec une notion d'utilité pour la décision. Un processus de décision markovien (MDP pour *Markov Decision Processus*, Putterman (1994) et Kaelbling, Littman & Cassandra (1998)) ressemble à une chaîne de

	HMM	RB
Méthodes d'inférence	Forward-Backward	Message Passing (4.1.1)
Méthodes d'apprentissage	Baum-Welch	EM (5.3.2)
Rechercher la chaîne la plus probable	Viterbi	MPE (4.1.6)

TAB. 2.1 : *Parallèle entre les algorithmes spécifiques aux chaînes de Markov cachées et ceux provenant des réseaux bayésiens*

Markov, sauf que la matrice de transition temporelle dépend d'une décision prise par un agent décideur à chaque pas de temps. L'agent reçoit une récompense (*utilité*) qui dépend de sa décision précédente et de l'état courant. Ces modèles sont donc utiles pour les problèmes de décision séquentiels (une décision à chaque pas de temps). Le but étant de trouver une politique de prise de décision qui maximise un critère de score.

Ce modèle nécessite que tous les états soient observables, ce qui n'est pas réaliste. De manière analogue aux chaînes de Markov cachées, il existe donc les *processus de décision Markovien partiellement observables* (POMDP pour *Partially Observable Markov Decision Process*) pour lesquels cette hypothèse est assouplie (Boutilier, Dean & Hanks (1999)).

Pour ces processus, il n'est pas possible de calculer le score d'une politique car celui-ci dépend d'évènements aléatoires. La solution optimale est donc construite en fonction de l'espérance du score de la politique choisie.

Une représentation plus consise que les POMDP, nommée PSR (pour *predicitive state representation*), a été proposée par Singh, Littman, Jong, Pardoe & Stone (2003) pour représenter les processus de décision markoviens.

2.2.4.5 Les réseaux bayésiens à temps continu

L'inconvénient des réseaux bayésiens dynamiques classiques est qu'ils supposent un pas de temps discret ce qui peut être suffisant pour de nombreuses applications, mais peut être un handicap pour de nombreuses autres. Les réseaux bayésiens à temps continu permettent alors de modéliser des processus stochastiques structurés avec un nombre fini d'états qui évoluent continuement dans le temps. Ces modèles ont été introduits par Nodelman & Horvitz (2003). Certaines méthodes d'inférence spécifiques sont alors décrites dans Nodelman, Shelton & Koller (2005) et Nodelman, Koller & Shelton (2005). Ces modèles sont représentés par des graphes dirigés, pouvant posséder des circuits, car dans ce cas, les circuits sont vus de manière temporelle et ne représentent pas réellement une boucle vers la même variable, mais cette variable évaluée *à un temps différent*.

2.2.5 Les réseaux bayésiens multi-entités

Nous pouvons imaginer utiliser plusieurs réseaux bayésiens pour représenter un système complexe. Laskey (2006) propose alors un formalisme qui unifie la *logique du premier ordre* et la *théorie des probabilités*. Ce formalisme est appelé les *réseaux bayésiens multi-entités* (MEBN pour *Multi-entity Bayesian network*). Les MEBN sont formés de *fragments* (appelés *MFrags*) qui représentent alors la distribution jointe d'un sous-ensemble de variables.

Un fragment est constitué d'un ensemble de variables de *contexte*, d'un ensemble de variables d'*entrée*, d'un ensemble de variables *résidentes*, d'un DAG sur les variables d'entrée et les variables résidentes (dans lequel les variables d'entrée sont des nœuds

racines) et d'un ensemble de distributions conditionnelles locales pour chaque variables résidentes. Un *MFrags* est très proche d'un réseau bayésien pour lequel les nœuds (de contexte) sont observés.

Un MEBN est donc ensuite un ensemble de *MFrags* qui doit, entre autre, satisfaire les propriétés qu'une variable ne doit jamais être ancêtre d'elle même (pas de circuit), ou encore que pour toute configuration, il doit exister un *unique MFrags* pour lequel cette configuration est une instance de ces variables résidentes.

Pour l'inférence dans ces modèles, il faut alors simplement construire le réseau bayésien suffisant pour répondre à la requête (alors appelé SSBN pour *Situation-Specific Bayesian Network*) et y effectuer une inférence.

2.2.6 Les structures de réseaux bayésiens adaptées à la classification

Dans les réseaux bayésiens, toutes les variables sont considérées de manière identique. Or, pour une tâche de classification, il est pertinent de considérer différement le nœud représentant la classe.

Par exemple, le classifieur de Bayes naïf est très largement utilisé. Sa structure est fixe et suppose que toutes les variables d'observation sont indépendantes deux à deux conditionnellement à la classe, ce qui revient à une simplification de la loi jointe suivante :

$$\mathbb{P}(C, X_1, ..., X_n) = \mathbb{P}(C)\mathbb{P}(X_1|C)...\mathbb{P}(X_n|C) \tag{2.6}$$

Ce modèle est aisé à mettre en oeuvre et a prouvé son efficacité pour de nombreuses applications. Par exemple, Spiegelhalter (1986) l'a utilisé dans un cadre médical. Androutspoulos, Palouras, Karkaletsis, Sakkis, Spyropoulos & Stamatopoulos (2000) a utilisé ce modèle pour faire de la détection de courriers électroniques indésirables. Et récemment, il a été incorporé à des clients de messageries électronique de renom. Sebe, Lew, Cohen, Garg & T.S. (2002) a utilisé ce modèle pour faire de la détection d'émotion à partir de l'image du visage d'une personne. De l'identification de peintre a été faite sur la base de classifieur naïf par Keren (2002). Ou encore, Zhou, Feng & Sears (2005) l'a utilisé pour automatiser la détection d'erreur d'un système de reconnaissance de la parole.

L'hypothèse à la base de ce modèle est largement non vérifiée et il existe différentes techniques pour l'assouplir, la plus intuitive est alors d'ajouter des dépendances entre les observations, nous obtenons alors une structure dite de *Bayes naïve augmentée*. Par exemple, les modèles appelés BNAN (pour *Bayesian Network Augmented Naive Bayes*) où l'on augmente la structure de Bayes naïve par une structure de réseaux bayésien quelconque. Nous verrons dans la section 8.1.2 comment apprendre automatiquement une structure de Bayes naïve augmentée par une structure arborescente (modèle appellé TAN pour *Tree augmented Naive bayes*).

Les Multi-Nets Bayésiens (BMN pour *Bayesian Multi-Nets* décrit dans Friedman, Geiger & Goldszmidt (1997) et Cheng & Greiner (2001)) sont une généralisation des structures de Bayes naïves augmentées dans le sens où elles autorisent différentes relations entre les attributs pour les différentes valeurs de la classe. Les classifieurs construits en utilisant ces structures permettent alors d'obtenir de meilleur résultats que l'utilisation de classifieurs de Bayes naïfs (TAN ou BNAN).

Les réseaux bayésiens récursifs (RBMN pour *Recursive Bayesian Multi-Nets* introduit dans Peña, Lozano & Larrañaga (2002)) sont des modèles de représentation des connaissances qui utilisent une structure d'arbre de décision pour séparer les contextes puis

contient un réseau bayésien à chaque feuille de l'arbre qui est alors plus efficace car spécifique à chaque contexte.

Citons encore les réseaux bayésiens à base de cas (CBBN pour *Case-Based Bayesian Networks* introduit dans Santos & Hussein (2004)) qui assouplissent la structure d'arbre de décision qui est présente dans les RBMN pour proposer l'utilisation d'un réseau bayésien différent pour chaque configuration possible des variables les plus significatives.

2.3 Les réseaux bayésiens causaux

Un réseau bayésien est dit *causal* si tous les arcs représentent des relations de causalité. Une modélisation suppose une connaissance parfaite du problème et de toutes les causes potentielles. Seulement, il n'est pas toujours possible d'avoir une connaissance parfaite de toutes les variables pouvant entrer dans les relations de causalité. Nous allons introduire différentes modélisations pour les phénomènes causaux en commencant par la plus simple : celle qui suppose la *suffisance causale*.

2.3.1 Les modèles causaux markoviens

Dans les réseaux bayésiens, nous devons identifier l'orientation de tous les arcs. Lorsque nous construisons un réseau bayésien, par exemple à partir d'un graphe essentiel (3.4), toutes les arêtes doivent être orientées. Certaines de ces orientations sont alors choisies arbitrairement. Lorsque nous voulons modéliser de 'vrais' phénomènes de *cause* à *effet*, nous ne pouvons pas nous permettre de faire cela.

Il est possible de représenter toutes sortes de modalités entre les variables dans un réseau bayésien. Elles peuvent être d'ordre causal, temporel, hiérarchique, etc. En général, une seule modalité est utilisée dans un même réseau et la plupart du temps, il s'agit de la causalité (voir page 199). Ceci permet de représenter l'influence directe d'une variable sur une autre : s'il existe un arc dirigé allant d'une variable A à une variable B, alors A est une des causes possibles de B, ou encore A a une influence causale directe sur B.

Les réseaux bayésiens pour lesquels tous les arcs représentent des relations causales sont dit *réseaux bayésiens causaux*. Il faut remarquer que tous les réseaux bayésiens ne sont pas causaux. Souvent, certaines orientations d'un réseau ne peuvent être présentes que pour représenter un aspect temporel ou un autre type de modalité, voire simplement pour profiter de l'avantage calculatoire de ces modèles.

Définition 2.3.1 *Un* modèle causal markovien *(ou* réseau bayésien causal*) est un graphe orienté sans circuit, où l'ensemble des variables vérifie l'*hypothèse de suffisance causale. *De plus, tous les arcs du graphe représentent des relations causales.*

Nous avons ici connaissance de toutes les variables pertinentes pour la modélisation, néanmoins, il est possible que certaines de ces variables soient cachées. Il faut bien faire la distinction entre une variable cachée (existance connue mais non observée) et une variable latente (non connue *a priori* et donc *a fortiori* non observée). Lorsqu'une variable dite latente est introduite dans un modèle, ce n'est alors qu'une variable hypothétique. La découverte des variables latentes est alors un challenge des plus intéressants. Introduisons à présent des modèles causaux qui ne supposent plus l'hypothèse de *suffisance causale*.

2.3.2 Les modèles causaux semi-markoviens

Lorsque nous ne connaissons pas toutes les variables influençant un système, nous pouvons tout de même écrire la formule suivante, où h représente un nombre de variables latentes.

$$\mathbb{P}(X_1, \cdots, X_n) = \sum_{j=1}^{h} \prod_{i=1}^{n} \mathbb{P}(X_i | Pa(X_i), LPa(X_i)) \prod_{j=1}^{h} \mathbb{P}(L_j | Pa(L_i), LPa(L_i)) \tag{2.7}$$

où $Pa(Z)$ représente le vecteur aléatoire contenant l'ensemble des variables observables parentes de Z et $LPa(Z)$ représente l'ensemble des variables non observables parentes de Z.

Définition 2.3.2 *Un* modèle causal semi-markovien *(*SMCM *pour* semi-Markovian causal model*) est un graphe causal sans circuit contenant à la fois des arc dirigés et des arcs bi-dirigés. Les nœuds du graphe représentent les variables observables notées X_i et les arcs bi-dirigés représentent implicitement les variables latentes notées L_i. Dans un* modèle causal semi-markovien*, toute variable latente ne possède pas de parents et possède exactement deux variables observables filles.*

Théorème 2.3.1 (Tian & Pearl (2002a)) *Tout modèle causal avec des variables latentes arbitraires (*c-à-d *pouvant posséder autant de parents et autant de variables filles que nécessaire) peut être transformé en modèle causal semi-markovien. Cette transformation sauvegarde les relations d'indépendance et de causalité entre les variables observées.*

D'après ce résultat, il devient possible d'utiliser les modèles semi-markoviens pour représenter la structure des réseaux bayésiens causaux quelconques quand par la suite nous ne voulons effectuer des inférences que sur les variables observables (Pearl (2000) et Tian & Pearl (2002b)). Un exemple de représentation de modèle semi-markovien est visible sur la figure 2.10(b). Ce résultat nous permet également d'écrire que la distribution de probabilité jointe possède désormais une décomposition de la forme de l'équation 2.8.

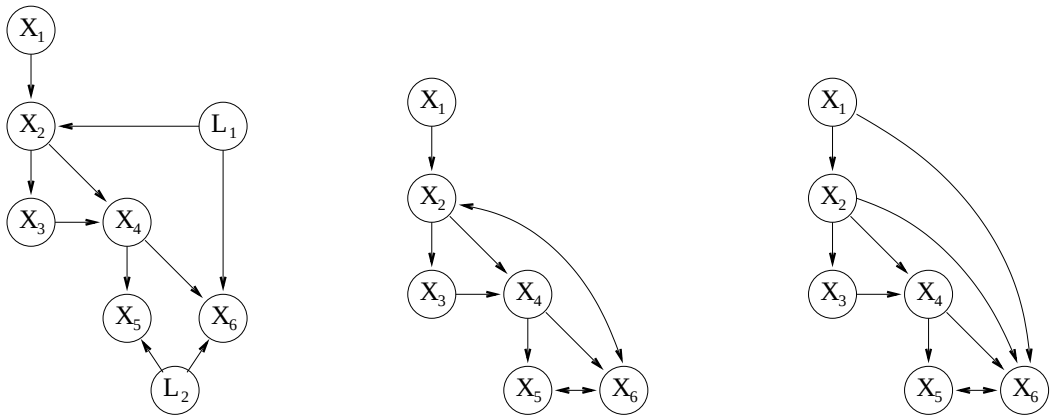
$$\mathbb{P}(X_1, \cdots, X_n) = \sum_{j=1}^{h} \prod_{i=1}^{n} \mathbb{P}(X_i | Pa(X_i), LPa(X_i)) \prod_{j=1}^{h} \mathbb{P}(L_j) \tag{2.8}$$

L'avantage de ces modèles est qu'il est alors possible d'y effectuer une inférence, qui n'est plus une inférence probabiliste, mais qui est une *inférence causale* (Spirtes, Meek & Richardson (1999) et Pearl (2000) et Maes (2005) et Maes, Meganck & Leray (2007)).

L'*inférence causale* est l'évaluation que l'effet d'une manipulation d'un ensemble de variables X sur un ensemble de variables Y. Nous définissons alors l'opérateur do qui consiste en cette manipulation et l'*inférence causale* consiste alors en l'évaluation de la quantité $\mathbb{P}(Y = y | do(X = x))$.

Certaines techniques ont alors été developpées pour effectuer l'apprentissage de ces modèles (par exemple dans Meganck, Leray & Manderick (2006) et Maes (2005)).

Il est toujours possible de mélanger différentes réprésentations ou interprétations des réseaux bayésiens. Par exemple, Maes, Meganck & Manderick (2005) et Maes (2005) introduisent des *réseaux bayésiens multi-agents causaux* et proposent une technique pour effectuer une inférence causale dans ces modèles.



(a) Le réseau original.

(b) Sa réprésentation par un modèle causal semi-markovien.

(c) Sa réprésentation par un graphe ancestral maximal.

FIG. 2.10 : *Différents types de modélisations avec variables latentes.*

2.3.3 Les graphes ancestraux maximum

Les graphes ancestraux, introduit par Richardson & Spirtes (2002), sont une autre modélisation supportant l'existance de variables latentes.

Définition 2.3.3 *Un graphe ancestral (sans conditionnement) est un graphe sans circuit contenant des arcs dirigés et des arcs bi-dirigés de telle manière qu'il n'existe pas d'arcs bi-dirigés entre deux variables qui sont connectées par un chemin dirigé.*

Définition 2.3.4 *Un graphe ancestral est dit* maximal *si pour toute paire de nœuds* (X, Y) *non adjacents, il existe un ensemble de séparation* Z *tel que* $X \perp\!\!\!\perp Y | Z$ *pour* $\mathbb{P}$.

Un graphe ancestral maximal (MAG pour *Maximal Anestral Graph*) est donc un graphe ancestral qui vérifie la condition de Markov. La principale différence entre les MAG et les SMCM est que dans ces premiers les arcs ne représentent pas forcément des relations causales directes, mais plutôt des relations ancestrales pour lesquelles les causes immédiates ne sont pas toujours connues.

L'exemple de la figure 2.10, extrait de Maes, Meganck & Leray (2007), permet de voir la différence entre les graphes ancestraux maximaux et les modèles causaux semi-markoviens.

Certains algorithmes ont été proposé pour effectuer l'apprentissage de ses modèles (Zhang & Spirtes (2005) et Zhang (2006)).

2.4 Les réseaux possibilistes

Il existe une version possibilistes des réseaux bayésiens en utilisant la logique floue et la théorie de possibilité de Zadeh (1978) a lieu de la théorie des probabilités. Gebhardt & Kruse (1995) ont alors proposé une technique d'apprentissage de ces modèles. Sangüesa & Cortés (1997) ont ensuite publié un *survey* décrivant les différentes méthodes d'apprentissage existantes et ont également une nouvelle technique d'apprentissage. Plus récement, Ben Amor, Benferhat & Mellouli (2002) ont introduit un algorithme d'inférence pour les réseaux possibilistes.

2.5 Les modèles non orientés

Introduisons maintenant les modèles non-orientés. La principale différence entre ces modèles et leurs versions orientées est que dans les modèles non-orientés, les relations représentées par les arêtes sont principalement des relations symétriques (corrélation, indépendances conditionnelles, *etc*) alors que les arcs des modèles orientés représentent principalement des relations asymétriques (causalité, temporelles, *etc*). Or, comme nous l'avons vu, s'il n'est pas attaché de signification particulière à l'orientation des arcs, les modèles orientés représentent également très bien des relations symétriques. En contrepartie, ce que les modèles non-dirigés perdent en interprétabilité, leur permet de gagner en simplicité. En effet, l'espace des graphes non-orientés est nettement plus petit que l'espace des graphes orientés, ce nombre $UG(n)$ est alors identique au nombre de relations binaires symétriques et reflexives, et vaut

$$UG(n) = 2^{\frac{n(n-1)}{2}} << r(n) \underset{n\to\infty}{\sim} n^{2^{\mathcal{O}(n)}} \tag{2.9}$$

Les méthodes d'apprentissage qui leur sont propres peuvent alors être plus efficaces.

2.5.1 Les modèles de Gibbs ou champs de Markov

Les *champs de Markov*[iv] (ou MRF pour *Markov Random Fields*) sont les modèles graphiques probabilistes pour lesquels les graphes sont non orientés. Pour les *champs de Markov*, il y a une notion de voisinage $\mathcal{N}_i$ pour chaque variable X_i telle que $X_i \notin \mathcal{N}_i$ et $X_j \in \mathcal{N}_i \Leftrightarrow X_i \in \mathcal{N}_j$. Et donc l'ensemble des variables $X = (X_1, \cdots, X_n)$ est appellé un champ de Markov si $\mathbb{P}(X = x) > 0$.

$$\mathbb{P}(X_i = x_i | X_{\{1,\cdots,n\}\setminus\{i\}} = x_{\{1,\cdots,n\}\setminus\{i\}}) = \mathbb{P}(X_i = x_i | X_{\mathcal{N}_i} = x_{\mathcal{N}_i}) \tag{2.10}$$

Un champ de Markov peut être représenté par une distribution de Gibbs (Besag (1974)). Pour plus de détails sur ces modèles se référer à Geman & Geman (1984).

De nombreux cas particuliers de ces modèles ont été étudiés et utilisés pour des applications spécifiques. Nous donnons ici un exemple de telles spécialisations.

Les modèles d'Ising :

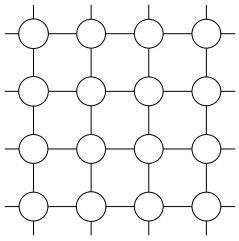


FIG. 2.11 : *Un exemple de champ de Markov : le modèle d'Ising.*

Les *modèles d'Ising* sont des graphes non orientés en grille où chaque nœud est connecté à tous ses voisins en vertical et en horizontal. Ces modèles peuvent être utiles en traitement d'image, pour modéliser des percolateurs ou encore pour faire des versions probabilistes du jeu de la vie, etc.

[iv] Ces modèles sont parfois nommés *réseaux de Markov* ou *modèles de Gibbs-Markov*, ce qui est source de confusion avec les modèles de Markov 2.2.4 ou encore les modèles markoviens 2.3.1.

2.5.2 Les graphes factorisés

Ce sont des graphes bipartis, où les cercles représentent les variables, et les carrés représentent les potentiels de cliques (voir figure 2.1.4). Le carré correspondant est alors relié à tous les nœuds de la clique. Le pouvoir expressif de ces modèles est le même que celui des champs de Markov. Le terme 'factorisé' est là pour mettre en évidence que le potentiel d'une clique est partagé par tous les nœuds de celle-ci.

2.5.3 Les graphes cordaux

Définition 2.5.1 (graphe cordal, corde) *Un* graphe cordal *(ou graphe* triangulé*) est un graphe (non dirigé) pour lequel tout cycle de longueur supérieure ou égale à* 4 *possède une* corde*, c'est-à-dire une arête entre deux sommets non voisins dans ce cycle.*

Les graphes cordaux sont les modèles qui sont à l'intersection des modèles dirigés et de ceux non dirigés. En effet, pour transformer un DAG en champ de Markov, il faut enlever les orientations et effectuer la moralisation (voir définition 4.1.1 page 43), ce qui ajoute de nouvelles arêtes. De même, pour transformer un modèle non dirigé en DAG, il faut effectuer une triangulation (voir définition 4.1.2) puis orienter les arêtes. Le seul cas où ces transformations s'effectuent sans ajout de nouvelles arêtes est lorsque nous sommes en présence d'un graphe cordal.

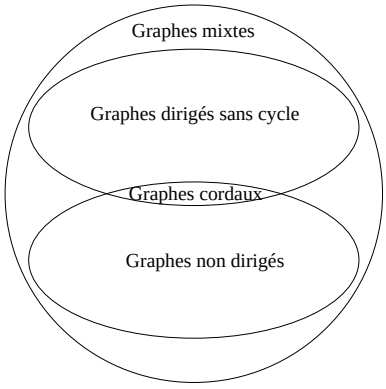


FIG. 2.12 : *Différents modèles graphiques.*

2.6 Les modèles semi-orientés

2.6.1 Les *Chain graphs*

Les *Chain graphs*, également appelés *graphes mixtes*, possèdent les avantages des modèles dirigés et des modèles non dirigés du point de vue de l'expressivité puisqu'ils profitent de la V-structure des modèles dirigés et du carré des modèles non dirigés. Les LWF *chain graph* ont été introduits par Lauritzen & Wermuth (1989), Wermuth & Lauritzen (1990) et Frydenberg (1990). Récemment, Anderson, Madigan & Perlman (2001) ont donné une nouvelle formulation de ces modèles, les AMP *chain graph*. Les méthodes d'inférence pour les LWF *chain graphs* sont bien développés, mais celles pour les AMP *chain graphs* (pour *Alternative Markov Property*) le sont moins.

Les modèles mixtes peuvent être modélisés par des modèles orientés en introduisant des variables auxiliaires, voir la figure 2.13. Cette remarque contribue également au

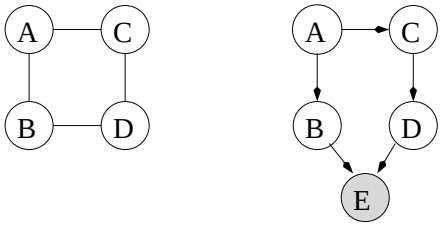


FIG. 2.13 : *Il est possible de représenter un champ de Markov sous forme de* DAG *en introduisant des nœuds auxiliaires. Par exemple, pour le graphe de gauche nous modélisons les indépendances* $A \perp\!\!\!\perp D|\{B,C\}$ *et* $B \perp\!\!\!\perp C|\{A,D\}$*, alors que pour le graphe de droite nous modélisons* $A \perp\!\!\!\perp D|\{B,C,E\}$ *et* $B \perp\!\!\!\perp C|\{A,D,E\}$.

choix des réseaux bayésiens comme modèles de prédilection (vis-à-vis des modèles non orientés).

Par la suite, nous n'allons plus considérer que des réseaux bayésiens. Commençons par introduire les méthodes d'inférence dans ces modèles.

2.6.2 Les réseaux bayésiens de niveau deux

Pour un réseau bayésien classique, les nœuds ne peuvent représenter qu'une seule variable. Cette hypothèse est assouplie avec la définition de réseaux bayésiens de niveau deux introduite par Linda Smail (2004). Les modèles en résultant ont une plus grande lisibilité.

Par exemple, en présence d'une structure compliquée faisant intervenir de nombreuses variables dont les états ne nous intéressent pas mais qui sont pertinentes pour la modélisation, il est possible de les conserver en simplifiant la structure comme il est indiqué sur la figure 2.14.
Pour cet exemple, si l'on regarde le réseau bayésien classique, nous écrirons la loi jointe associée comme $\mathbb{P}(A,B,C,D) = \mathbb{P}(A)\mathbb{P}(B|A)\mathbb{P}(C|A)\mathbb{P}(D|B,C)$. Par contre en regardant le réseau de niveau deux nous écrirons plutôt $\mathbb{P}() = \mathbb{P}(A)\mathbb{P}(B,C|A)\mathbb{P}(D|B,C)$. Ici, nous ne nous intéressons plus à la valeur de chaque nœud intermédiaire, mais plutôt à leur loi jointe. Par ailleurs, nous nous apercevons que dans ce cas, nous récupérons une structure arborescente, alors que ce n'était pas le cas pour le réseau bayésien original.

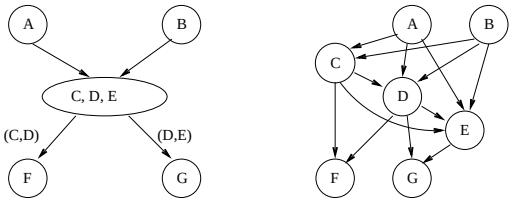


FIG. 2.14 : *Un exemple de réseau bayésien de niveau deux renseigné et un* DAG *équivalent.*

Un réseau bayésien de niveau deux est dit *renseigné* si les arcs sont annotés par les variables qui ont de l'influence sur les nœuds à leurs extrémités.
Un exemple extrait de Smail (2004) est donné sur la figure 2.14. Nous remarquons dans cet exemple que ces modèles sont plus visuels, donc certainement plus intuitifs. Ici, nous obtenons une structure d'arbre alors que le graphe original était complexe.

3

Le cadre des modèles graphiques

$$\mathbb{P}(BlesserHumain) = 0$$
$$\mathbb{P}(Obéir|\neg BlesserHumain) = 1$$
$$\mathbb{P}(SeProtéger|Obéir \wedge \neg BlesserHumain) = 1$$

Manuel de la Robotique, 58e édition (2058 après J.C.)[i]

Sommaire

3.1 Introduction 30
3.1.1 Condition de Markov 30
3.1.2 Fidélité 31
3.1.3 Graphoïdes 33
3.2 Critère de séparation directionnelle 33
3.2.1 *d*-séparation 34
3.2.2 *Minimal I-map* : carte d'indépendances minimale 35
3.2.3 *Maximal D-map* : carte de dépendances maximale 37
3.2.4 *P-map* : carte parfaite 38
3.3 Table de probabilités conditionnelles 38
3.4 Classes d'équivalence de Markov 39

3.1 Introduction

À présent, nous allons mettre en place le formalisme des modèles graphiques probabilistes. Ces modèles permettent de modéliser les systèmes complexes et/ou chaotiques et utilisent à la fois le formalisme de la théorie des probabilités (voir annexe B) et le formalisme de la théorie des graphes (voir annexe C).

Il existe deux principales manières de lier les probabilités avec une représentation graphique. La première s'appuie sur la notion de *condition de Markov* et sur l'hypothèse de fidélité décrite dans Spirtes, Glymour & Scheines (1993) et Spirtes, Glymour & Scheines (2000).

La seconde, plus théorique, s'appuie sur le fait qu'il est possible de trouver une axiomatisation, les *graphoïdes*, commune à la relation de dépendance conditionnelle et à la relation de *séparation* (ou *d-séparation*) dans les graphes comme le montrent Pearl (1988) et Pearl (2000).

3.1.1 Condition de Markov

Définition 3.1.1 *La condition de Markov*[ii] *peut être énoncée comme ceci :*
Pour chaque variable X_i ***et tout sous-ensemble de variables*** Y ***de*** $X \setminus Desc(X_i)$**,** ***alors*** $\mathbb{P}(X_i|Pa(X_i), Y) = \mathbb{P}(X_i|Pa(X_i))$.

En clair, cela signifie que l'ensemble des parents de X_i sépare X_i des autres variables, exceptées ses descendantes. (Pour une version graphique non dirigée, il faudrait remplacer l'ensemble des parents, par la frontière de Markov, et $Y \subset X \setminus M(X_i)$. voir l'annexe C, pour les définitions).

Supposons par ailleurs que le graphe orienté considéré représente des relations causales entre les variables de X. Le graphe est alors appelé un *graphe causal*. Pour exprimer la signification des arcs, il existe alors la *condition de Markov Causale* qui s'exprime alors de la même manière mais lorsque le graphe prend cette signification causale particulière (qui n'existe alors pas pour les graphes non dirigés).

Dans ce cas, il faut de plus que l'ensemble de toutes les variables considérées $\mathcal{X}$ contienne également **toutes les variables représentant les causes des variables dans** $\mathcal{X}$, il s'agit alors de l'hypothèse de *complétude causale*.

En effet si une cause commune à deux variables existe et n'est pas dans X alors le lien entre ces deux variables ne pourra pas être orienté car aucune n'est la cause de l'autre (figure 3.1), mais ce lien devra exister car les variables sont tout de même corrélées. Dans ce cas, la *condition de Markov causale* ne pourrait pas être vérifiée pour X sans l'ajout de cette cause commune.

Donc, en particulier, la condition de Markov causale, même si elle s'exprime de la même manière, est plus forte puisqu'elle implique la complétude causale.

Il est alors maintenant possible d'interpréter les arcs comme des relations causales entre les variables.

---

[i] La citation de la page précédente est une idée originale de Julien Diard (2003), reprise de sa thèse de doctorat. Il s'agit bien sûr d'une traduction en langage probabiliste des trois lois de la robotique inventées par Isaac Asimov (1920-1992).

[ii] Egalement appelée *Local Markov Condition* par certains auteurs. Cette condition présentée ici dans le cadre des modèles dirigés peut être adaptée au cadre des modèles graphiques non dirigés.

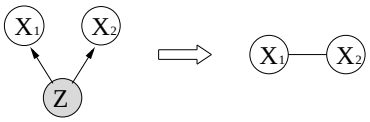


FIG. 3.1 : *Illustration de la complétude pour que la condition de Markov causale soit valide.*

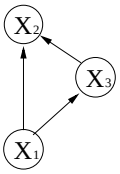


FIG. 3.2 : *Deux chemins d'influence causale ne doivent pas voir leur effet s'annihiler.*

Par exemple, si nous considérons le graphe de la figure 3.2, la variable X_1 a une double influence sur la variable X_2. Une influence directe (par exemple, faire du sport a une influence directe sur le renforcement du métabolisme) et une autre indirecte (faire du sport oxyde les cellules, pour faire simple, ce qui a une influence sur l'affaiblissement du métabolisme).
Dans ce cas, les deux effets sont partagés, et après une observation (et une inférence), nous en déduisons si le type d'activité a eu un effet bénéfique ou non (dans cet exemple, comme la deuxième composante est moins significative, il est toujours bon de faire du sport).

3.1.2 Fidélité

Définition 3.1.2 (fidélité) *Nous dirons qu'un graphe est fidèle*[iii] *à une distribution de probabilité lorsque* ***toute indépendance probabiliste existant entre les variables considérées peut être déduites de la condition de Markov****.*

Revenons sur notre exemple de la figure 3.2 et supposons de plus que les deux processus de dépendance (X_1 vers X_2 directement et via X_3) s'annulent, et que l'influence probabiliste (et non causale) disparaisse. Il n'y a donc plus vraiment de dépendance intéressante à conserver entre les variables X_1 et X_2. Le modèle de la figure 3.2 n'est plus valide du point de vue de la fidélité car il implique que X_1 et X_2 ne peuvent pas être indépendantes donc que les influences (probabilistes) ne doivent pas s'annuler. Néanmoins, comme il est plus complexe (il encode également des dépendances qui n'existent plus), il est toujours possible de l'utiliser, en particulier si l'on ne s'intéresse pas particulièrement à représenter les dépendances causales.
Dans ce cas, pour qu'un modèle satisfasse le principe de fidélité, il faut qu'il vérifie que X_1 et X_2 soient marginalement indépendantes et donc il faut que cette indépendance soit lisible sur le graphe grâce à la condition de Markov. Le seul modèle cohérent devient alors le modèle représenté sur la figure 3.3. La condition de Fidélité ressemble alors plus à un principe méthodologique proche du razoir d'Ockham (voir page 77), puisqu'il permet de choisir le modèle le plus simple dans le sens où il n'introduit pas de dépendances 'probabilistes' superflues, même quand les dépendances causales existent effectivement.

[iii] La fidélité est également appellée *Stability*. Certains auteurs francophones diront qu'un graphe est stable lorsqu'il est fidèle.

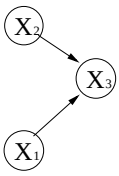


FIG. 3.3 : *La V-structure comme structure simplifiant la structure de la figure 3.2 quand les dépendances s'annihilent.*

Prenons un autre exemple qui relève de la fidélité. Considérons l'univers d'événements $\Omega = \{1,2,3,4,5,6\}$ et les événements $X_1 = \{1,2,3\}$ et $X_2 = \{3,4\}$. Supposons les événements élémentaires équiprobables, nous avons alors, en notant X_i la variable aléatoire indicatrice représentant "*l'événement X_i s'est réalisé*", $\mathbb{P}(X_1) = \mathbb{P}(X_1|X_2) = \frac{1}{2}$ et $\mathbb{P}(X_2) = \mathbb{P}(X_2|X_1) = \frac{1}{3}$. Les variables X_1 et X_2 sont donc (probabilistiquement) indépendantes.

A présent, en considérant l'événement conjoint $X_3 = X_1 \cap X_2 = \{3\}$, les événement X_1, X_2 et X_3 ne sont plus indépendants. En particulier, le fait de savoir que, à la fois X_2 et X_3 se sont réalisés, nous informe que X_1 s'est également réalisé (et de même en inversant X_1 et X_2) donc $X_1 \perp\!\!\!\perp X2$ et $X_1 \not\perp\!\!\!\perp X_2|X_3$ (car $X_1 \not\perp\!\!\!\perp X_2|X_3 = 1$). Nous reconnaissons ici ce qui est appelé une V-structure (*cf.* définition 3.2.2) dans le formalisme des modèles graphiques orientés et nous pouvons alors représenter graphiquement les dépendances entre ces événements par le graphe de la figure 3.3.

Par ailleurs, l'événement X_3 est une cause commune à X_1 et à X_2, en effet, si $\{3\}$ est réalisé alors $\{1,2,3\}$ et $\{3,4\}$ sont également réalisés, donc nous pourions représenter ses trois variables par le graphe dirigé de la figure 3.4.

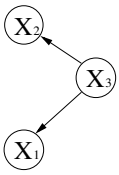


FIG. 3.4 : *Le modèle à arcs divergents, X_3 est une cause commune.*

Mais quel modèle choisir ?

Supposons à présent que la distribution de masse devienne $\mathbb{P}(1) = \mathbb{P}(4) = \mathbb{P}(6) = 0.1$, $\mathbb{P}(2) = \mathbb{P}(5) = 0.2$ et $\mathbb{P}(3) = 0.3$. Dans ce cas, $\mathbb{P}(X_1) = 0.6$ tandis que $\mathbb{P}(X_1|X_2) = 0.75$ et $\mathbb{P}(X_2) = 0.4$ tandis que $\mathbb{P}(X_2|X_1) = 0.5$ et les événements X_1 et X_2 sont ici corrélés. Dans la plupart des situations, lorsque deux événements sont corrélés, c'est qu'ils dépendent tous deux d'un troisième. Cet événement peut alors être considéré comme une cause de X_1 et de X_2.

Comme nous l'avons vu précédemment l'événement X_3 est une cause à X_1 et X_2. Comme les variables X_1 et X_2 ne sont pas marginalement indépendantes, nous allons choisir le modèle de la figure 3.4 pour modéliser les dépendances entre les trois variables X_1, X_2 et X_3. Nous conservons ici l'interprétation causale du graphe.

Par contre, dans le cas de l'équiprobabilité des événements élémentaires, si nous voulons être "fidèle" à l'ensemble des dépendances conditionnelles, nous sommes contraint d'utiliser le modèle de la figure 3.3. Ici, nous perdons alors l'interprétation causale du graphe et nous obtenons de plus un graphe qui est plus complexe que la structure causale.

3.1.3 Graphoïdes

Les graphoïdes sont une approche radicalement différente de l'approche précédente ou nous modélisions la cohérence qu'il y avait dans une représentation graphique des relations de dépendances probabilistes. À présent, une axiomatisation va être introduite, celle de graphoïde, de manière à ce que, à la fois, la relation de dépendance probabiliste et la relation de d-séparation dans les graphes respectent cette modélisation axiomatique. Il existe donc une sorte de relation *bijective* entre ces deux dernières notions, et il parait alors naturel d'utiliser la représentation graphique pour représenter les dépendances probabilistes.

Définition 3.1.3 (semi-graphoïde) *Soient W, X, Y et Z des ensembles de variables aléatoires deux à deux disjoints. Une relation ternaire $(\cdot \perp_g \cdot \,|\, \cdot)$ est dite un* semi-graphoïde *si elle vérifie les axiomes suivants :*

- *symétrie :* $(X \perp_g Y|Z) \Rightarrow (Y \perp_g X|Z)$
- *décomposition :* $(X \perp_g Y \cup W|Z) \Rightarrow (X \perp_g Y|Z)$
- *union faible :* $(X \perp_g Y \cup W|Z) \Rightarrow (X \perp_g Y|Z \cup W)$
- *contraction :* $(X \perp_g Y|Z) \wedge (X \perp_g W|Z \cup Y) \Rightarrow (X \perp_g Y \cup W|Z)$

Propriété 3.1.1 *L'indépendance conditionnelle vérifie les axiomes des semi-graphoïdes.*

À partir de dépendances conditionnelles connues, il est donc possible d'en obtenir de nouvelles en utilisant ces axiomes.

Définition 3.1.4 (graphoïde) *Un* graphoïde[iv] *est un* semi-graphoïde *vérifiant de plus l'axiome suivant :*

- *intersection :* $(X \perp_g Y|Z \cup W) \wedge (X \perp_g W|Z \cup Y) \Rightarrow (X \perp_g Y \cup W|Z)$

Remarquons que si la probabilité $\mathbb{P}$ est strictement positive alors l'indépendance conditionnelle vérifie les axiomes de graphoïde. La notion de graphoïde introduite par Pearl & Paz (1985) et Geiger (1990) permet de faire le lien entre la notion d'indépendance conditionnelle entre les variables aléatoires, et les notions de séparation dans les graphes non orientés et de d-séparation dans les graphes orientés. Cette notion justifie alors l'identification entre variables aléatoires et nœuds d'un graphe qui sera faite par la suite.

Nous avons vu avec la notion de *Fidélité*, que toute indépendance probabiliste peut être déduites de la condition de Markov. La manière de déduire d'autres indépendances consiste en l'exploitation des axiomes des graphoïdes. Réciproquement, ces axiomes permettent de retrouver toutes les indépendances (probabilistes) qui sont représentées par une structure de graphe orientée sans circuit, et donc, si ce graphe est fidèle, par la distribution de probabilité jointe associée.

3.2 Critère de séparation directionnelle

Dans les années 1930, le biologiste Sewall Wright (1934) essayait de trouver un moyen de modéliser statistiquement la structure causale de systèmes biologiques. Il a

[iv] Historiquement, les *semi-graphoïdes* étaient appelés *graphoïdes* et les *graphoïdes* étaient appelés *graphoïdes positifs*.

alors trouvé une représentation unifiée qui combinait les graphes dirigés, pour représenter des hypothèses causales, et des modèles statistiques linéaires, pour représenter des équations de régression linéaires.

Définition 3.2.1 (séparation) *Deux ensembles de nœuds X et Y d'un graphe $\mathcal{G}$ sont dits* séparés *par un ensemble de nœuds Z, si ces deux ensembles de noeuds se retrouvent dans des composantes connexes disjointes du sous-graphe de $\mathcal{G}$ relativement à l'ensemble de sommets $\{1, \cdots, n\} \setminus Z$. Nous noterons alors $X \perp Y | Z$.*
Dans ce cas, toute chaîne reliant un nœud de X à un nœud de Y possède un sommet intermédiaire dans Z.

Cette notion est principalement utilisée pour les graphes non orientés.

Théorème 3.2.1 (Pearl & Paz (1985)) *La séparation dans les graphes non orientés vérifie les axiomes des semi-graphoïdes.*
De plus, la séparation entre ensembles disjoints de nœuds vérifie les axiomes des graphoïdes.

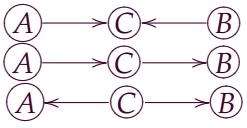
3.2.1 *d*-séparation

La notion de séparation, même si elle est adaptable pour les graphes orientés, n'est pas pertinente car dans le cas de la V-structure (voir définition 3.2.2), deux variables pourront être séparées sans pour autant être indépendantes (voir figure 3.5). Cette notion n'est donc pas assez fine pour être un critère graphique représentant l'indépendance conditionnelle. La notion de d-séparation va alors remplir ce rôle (le 'd' signifie directionnelle), et il sera possible de déduire s'il y a ou non indépendance conditionnelle pour tous les groupes de variables.

Définition 3.2.2 (V-structure) *On appelle* V-structure de sommet puits C*, tout sous-graphe de la forme suivante ou A et B ne sont pas connectés.*

Le nœud C sera également appelé sommet à arcs convergents.

Définition 3.2.3 (chaîne convergente, en série, divergente)
La chaîne suivante est dite convergente *en C :*
La chaîne suivante est dite en série *en C :*
La chaîne suivante est dite divergente *en C :*



Définition 3.2.4 (chaîne active) *Soit X,Y, et Z trois ensembles disjoints de variables. Une chaîne reliant X et Y est dite* active *par rapport à Z, si les deux conditions suivantes sont réunies :*

- *tout sommet à arcs convergents de cette chaîne est dans Z ou possède un descendant dans Z,*
- *aucun autre sommet de la chaîne n'est dans Z.*

Une chaîne qui n'est pas active *est dite* bloquée *par Z.*

Ces définitions doivent se retenir comme suit :

- une chaîne est **active**, si elle **crée des dépendances entre ses nœuds extrémités**,

– une chaîne est **bloquée**, si elle **ne peut pas créer de dépendance entre ses nœuds extrémités**.

Définition 3.2.5 (d-séparation) *On dit que Z* d-sépare *X et Y, ou encore que X et Y sont d-séparés par Z, et on note $X \perp_d Y | Z$, si toute chaîne reliant X à Y est bloquée par Z.*

Pour identifier les chaînes bloquées, il faut prêter une attention particulière aux V-structures. En clair, si on trouve **une** V-structure avec son sommet **puits**, ou **un de ses descendants**, qui est dans l'ensemble de conditionnement, alors la chaîne est bloquée et il **n**'y a **pas** d-séparation (voir figure 3.5).

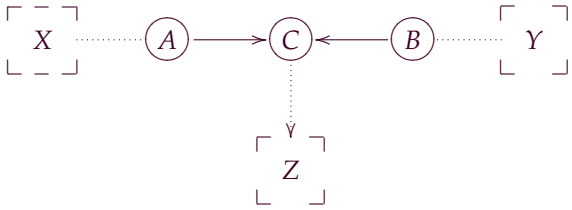


FIG. 3.5 : *Illustration d'un nœud* puits *C et de l'influence d'une V-structure pour identifier une indépendance conditionnelle. Ici, C sépare A et B, et pourtant $A \not\perp\!\!\!\perp B | C$. De plus, $A \perp\!\!\!\perp B$ mais si Z est un descendant de C, nous avons $A \not\perp\!\!\!\perp B | Z$ car toute chaîne reliant A à B est rendue* active *par Z.*

Par ailleurs, lorsqu'il n'y a pas de V-structure, par exemple en présence d'une structure en forme d'arbre, la d-séparation est équivalente à la séparation.

L'intérêt d'une telle notion réside dans le théorème suivant :

Théorème 3.2.2 (Verma & Pearl (1988)) *La relation de d-séparation dans les graphes dirigés sans circuits vérifie les axiomes des semi-graphoïdes, et de plus, si $\mathcal{G}$ est un graphe d'indépendances, alors pour tous X, Y et Z, des sous-ensembles (disjoints) de variables,*

$$X \perp_d Y | Z \iff \text{la décomposition dont } \mathcal{G} \text{ est issue entraîne que } X \perp\!\!\!\perp Y | Z \qquad (3.1)$$

3.2.2 *Minimal I-map* : carte d'indépendances minimale

Soit $X = (X_1, \cdots, X_n)$ la donnée de n variables aléatoires et Z un sous-ensemble de X.

Définition 3.2.6 (carte d'indépendances) *On appelle* carte d'indépendances *d'une distribution $\mathbb{P}$, tout graphe $\mathcal{G}$ vérifiant la condition de Markov, celle-ci pouvant être réécrite de la manière suivante.*

$$X_i \perp\!\!\!\perp X_j | Z \impliedby X_i \perp_d X_j | Z$$

Cela signifie que toute d-séparation trouvée dans le graphe nécessite la présence de la dépendance (conditionnelle) correspondante dans la loi, mais il est possible qu'il existe d'autres dépendances qui ne soient pas codées dans le graphe.

Soit $\mathbb{P}$ une distribution de probabilités[v] sur X et soit $\mathcal{G}$ une carte d'indépendances de sommets les X_i (par abus de langage).

[v] Par la suite, le terme 'indépendance' sera toujours utilisé à la place de 'indépendance pour $\mathbb{P}$' car, nous n'aurons toujours qu'une loi ; dans le cas contraire des précisions seront faites.

ALGORITHME 1 Algorithme générique d'apprentissage de structure

1: Soit G un graphe complet et $X_1, \cdots, X_N$ un ordre d'énumération des noeuds.
2: **Pour** i de 1 à N **Faire**
3: Choisir $Pa(X_i)$ un sous-ensemble minimal de $\{X_1, \cdots, X_{i-1}\}$ tel que

$$X_i \perp\!\!\!\perp \{X_1, \cdots, X_{i-1}\} \setminus Pa(X_i) | Pa(X_i)$$

4: **Fin Pour**

D'après le théorème 3.2.2, il est également possible de réécrire la condition de Markov comme[vi]

$$\{X_i \perp_d X \setminus (Desc(X_i) \cup \{X_i\}) | Pa(X_i)\}$$

Dans ce cas, $\mathbb{P}$ se *factorise* selon $\mathcal{G}$, c'est-à-dire :

$$\mathbb{P}(X_1, X_2, \cdots, X_n) = \prod_{i=1}^{n} \mathbb{P}(X_i | Pa(X_i)) \tag{3.2}$$

où $Pa(X_i)$ représente l'ensemble des nœuds parents de X_i dans $\mathcal{G}$.

Théorème 3.2.3 (Verma & Pearl (1988)) *Si L est une liste des relations causales d'un graphoïde $\mathcal{M}$ alors tout graphe dirigé sans circuit (*DAG*) construit à partir de L est une carte d'indépendances de $\mathcal{M}$.*

Définition 3.2.7 (carte d'indépendances minimale) *Le graphe $\mathcal{G}$ est appelé* carte d' indépendances minimale *pour $\mathbb{P}$, si aucun graphe partiel $\mathcal{G}'$ de $\mathcal{G}$ est une carte d'indépendances pour $\mathbb{P}$.*

Remarquons qu'un graphe complètement connecté est une carte d'indépendances pour toutes les lois de probabilité (faisant intervenir le même nombre de variables).

Remarquons par ailleurs, que les cartes d'indépendances minimales ne sont pas pour autant uniques, par exemple la règle de Bayes donne

$$\mathbb{P}(A,B,C) = \mathbb{P}(A)\mathbb{P}(B|A)\mathbb{P}(C|B) = \mathbb{P}(A|B)\mathbb{P}(B)\mathbb{P}(C|B) = \mathbb{P}(A|B)\mathbb{P}(B|C)\mathbb{P}(C)$$

donc les trois structures

$$(A) \to (B) \to (C), \quad (A) \leftarrow (B) \to (C) \text{ et } (A) \leftarrow (B) \leftarrow (C)$$

sont des cartes d'indépendances pour la loi $\mathbb{P}$ ci-dessus et il est aisé de voir qu'elles sont toutes minimales.

Cette remarque qui paraît anodine ne l'est en fait pas. Par exemple, considérons l'algorithme d'identification de structure décrit par l'algorithme 1. Celui-ci permet de retrouver des cartes d'indépendances minimales à partir d'un ordre sur les nœuds. En prenant en considération différents ordres d'énumérations des nœuds, il est alors possible d'obtenir des cartes d'indépendances très différentes (voir la figure 3.6). Certains de ces résultats n'ont alors que très peu d'expressivité d'un point de vue purement causal.

Remarquons, pour l'exemple de la figure 3.6, que les trois cartes d'indépendances ne représentent pas les mêmes ensembles d'indépendances conditionnelles car les réponses

[vi] à lire X_i est d-séparé de ses *non descendants* par ses parents.

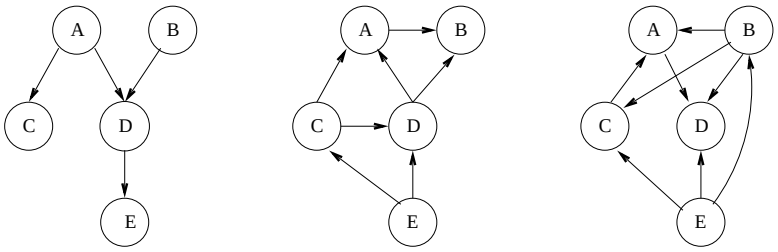


FIG. 3.6 : *Trois cartes d'indépendances minimales : la première représentant des liens de causalité (avec l'ordre $ABCDE$), et deux autres obtenues après tests d'indépendances considérant la première comme référence, l'une pour l'ordre $ECDAB$ et l'autre pour l'ordre $EBCAD$.*

aux tests de l'algorithme 1 n'ont pas été positives pour les mêmes ensembles de conditionnement. Par exemple, d'après l'équation 3.2, la première carte représente la liste de dépendances suivante : $A \perp\!\!\!\perp B, B \perp\!\!\!\perp \{A, C\}, C \perp\!\!\!\perp \{B, D, E\} | A, D \perp\!\!\!\perp C | \{A, B\}, E \perp\!\!\!\perp \{A, B, C\} | D$. Alors qu'aucune des deux autres ne vérifie ne serait-ce que la première relation d'indépendance. Ces cartes restent cependant des cartes minimales car les indépendances conditionnelles qu'elles encodent sont vérifiées par toute loi sous jacente à la première carte. En effet cela est dû au fait qu'une carte minimale représente certaines indépendances conditionnelles, mais pas toutes.

Néanmoins, ces cartes possèdent tout de même des listes de dépendances conditionnelles très proches. Elles vérifient par exemple toutes les relations suivantes : $A \not\perp\!\!\!\perp B | D, A \not\perp\!\!\!\perp C, A \not\perp\!\!\!\perp D, A \not\perp\!\!\!\perp E, B \not\perp\!\!\!\perp D, B \not\perp\!\!\!\perp E, C \not\perp\!\!\!\perp E, D \not\perp\!\!\!\perp E$. Remarquons, par ailleurs que la première carte est la seule à vérifier $A \perp\!\!\!\perp B$, tandis que les autres vérifient $A \not\perp\!\!\!\perp B$.

Cela n'est pas gênant, car pour être une carte d'indépendance minimale, il ne faut faire attention qu'aux relations de dépendances conditionnelles et non aux relations d'indépendances conditionnelles. Il est alors aisé de vérifier que les indépendances conditionnelles codées par les deuxième et troisième cartes de la figure 3.6 sont également vraies pour la première carte. Les autres listes d'indépendances vérifiées étant alors : $\{A, B\} \perp\!\!\!\perp E | \{C, D\}, B \perp\!\!\!\perp \{C, E\} | \{A, D\}, C \perp\!\!\!\perp B | \{A, D, E\}$ pour la deuxième carte et $A \perp\!\!\!\perp E | \{B, C, D\}, B \perp\!\!\!\perp C | \{A, D, E\}, C \perp\!\!\!\perp D | \{A, B, E\}$ pour la troisième.

Avec ce type de méthodes d'apprentissage de structure il faut donc être capable de prendre connaissance d'un ordre topologique le plus proche possible de celui de la structure originale. Or, trouver un ordre topologique optimum est également un problème NP-difficile (Singh & Valtorta (1993)). Nous verrons alors en dans le chapitre 8 comment trouver rapidement un ordre topologique qui n'est pas trop mauvais pour les problèmes de classification.

3.2.3 *Maximal D-map* : carte de dépendances maximale

Une carte de dépendances (définition 3.2.6) représente toutes les indépendances qui peuvent exister dans la distribution de probabilité, mais éventuellement plus. En particulier, un graphe vide est donc une carte de dépendances pour toutes lois (avec un nombre de variables fixées).

Définition 3.2.8 (carte de dépendances minimale) *Le graphe $\mathcal{G}$ est appelé* carte de dépendances minimale *pour $\mathbb{P}$, si pour tout graphe $\mathcal{G}'$ tel que $\mathcal{G}$ en est un graphe partiel, $\mathcal{G}'$ n'est pas une carte de dépendances pour $\mathbb{P}$.*

3.2.4 *P-map* : carte parfaite

Dans certains cas, une carte d'indépendances peut également être une carte de dépendances. Dans ce cas, c'est alors une carte d'indépendances minimale et une carte de dépendances maximale et on dit qu'il s'agit d'une carte *parfaite*.

Définition 3.2.9 *Un* DAG $\mathcal{G}$ *est appelé une* carte parfaite[vii] *d'une distribution* $\mathbb{P}$ *si*

$$(X \perp\!\!\!\perp Y|Z) \text{ pour } \mathbb{P} \iff (X \perp_d Y|Z) \text{ dans } \mathcal{G}$$

Une distribution de probabilité est dite représentable *(par un graphe dirigé sans circuit) s'il existe une carte parfaite pour celle-ci.*

Malheureusement de nombreuses distributions n'admettent pas de carte parfaite. Certaines peuvent néanmoins être représentées par des DAG (voir section 2.1) mais cela est faux dans la majorité des cas, même pour les plus simples[viii].

Par exemple, considérons l'ensemble d'événements équiprobables $\{1,2,3,4\}$ et les variables aléatoires X_1 : *l'événement* $\{1,2\}$ *s'est réalisé*, X_2 : *l'événement* $\{2,3\}$ *s'est réalisé*, X_3 : *l'événement* $\{3,1\}$ *s'est réalisé*.
Dans ce cas, $\mathbb{P}(X_1) = \mathbb{P}(X_2) = \mathbb{P}(X_3) = \frac{1}{2}$ et $\mathbb{P}(X_1 \cap X_2) = \mathbb{P}(X_2 \cap X_3) = \mathbb{P}(X_3 \cap X_1) = \frac{1}{4}$, donc les variables aléatoires X_1, X_2 et X_3 sont deux à deux indépendantes.
De plus, $\mathbb{P}(X_1 \cup X_2 \cup X_3) = \frac{3}{4}$ et $\mathbb{P}(X_1)\mathbb{P}(X_2)\mathbb{P}(X_3) = \left(\frac{1}{2}\right)^3 = \frac{1}{8}$, donc les variables aléatoires X_1, X_2 et X_3 ne sont pas trois à trois indépendantes.
L'indépendence deux à deux contraint la structure à être vide, et la dépendance trois à trois contraint la structure à contenir des arêtes. il n'y a donc pas de structure permettant de représenter fidèlement la loi jointe sur $\{X_1, X_2, X_3\}$.

La loi jointe sur $\{X_1, X_2, X_3\}$ n'est donc pas *représentable*, néanmoins, en fonction de notre objectif, il sera toujours possible de se contenter d'une carte d'indépendance minimale ou d'une carte de dépendance maximale qui, elles, existent toujours.

3.3 Table de probabilités conditionnelles

La paramétrisation d'un réseau bayésien nécéssite à la fois un graphe et un ensemble de distributions de probabilités conditionnelles. Pour ce travail, nous allons nous limiter aux variables aléatoires discrètes. Les distributions conditionnelles seront alors représentées par des matrices telles que la somme des éléments de chaque 'ligne' soit égale à 1, matrice que nous appellerons *tables de probabilités conditionnelles* (voir l'exemple de la figure 3.7).

Les réseaux bayésiens sont également capables de modéliser des distributions gaussiennes (Lauritzen & Wermuth (1989)), des mélanges de gaussiennes, des mélanges d'exponentielles tronquées (Cobb & Shenoy (2005) et Moral, Rumí & Salmerón (2001)).

Dans le cas des modèles conditionnels gaussiens, il est possible d'effectuer une inférence exacte (voir chapitre 4) tant qu'un nœud discret n'est pas enfant d'un nœud continu.

[vii] Il s'agit d'une traduction littérale de l'expression anglo-saxonne, le terme français est *graphe fidèle* ou *carte P-stable*, voir la définition 3.1.2.

[viii] Voir la figure 2.3 pour un autre exemple simple qui n'admet pas de carte parfaite à l'aide de DAG.

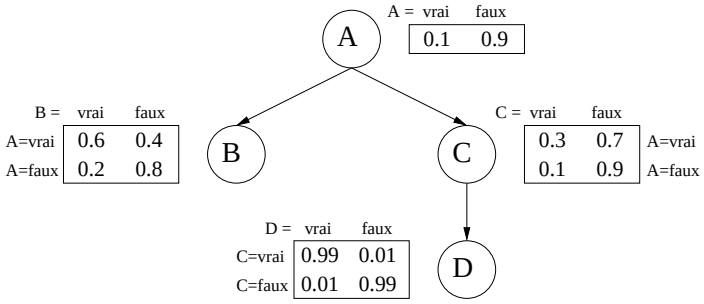


FIG. 3.7 : *Exemple de table de probabilités conditionnelles pour un réseau simple n'ayant que des nœuds binaires. Il est évident que la taille des tables peut être énorme si un nœud possède beaucoup de parents étant chacun de nombreuses modalités. Néanmoins, même dans ce cas, les réseaux bayésiens sont un modèle économique pour représenter une distribution (voir page 15).*

3.4 Classes d'équivalence de Markov

Un moyen simple de caractériser un ensemble de distributions compatibles avec un *graphe dirigé sans circuit* (DAG) est de lister l'ensemble des indépendances (conditionnelles) qui sont représentées par le graphe. Ces indépendances peuvent être aisément lues à partir du DAG en utilisant le critère de d-séparation (voir section 3.2). Or, il arrive que plusieurs DAG encodent le même ensemble d'indépendances conditionnelles. Par exemple, les trois structures de la figure 3.8 encodent toutes l'indépendance conditionnelle $A \perp\!\!\!\perp B|C$.

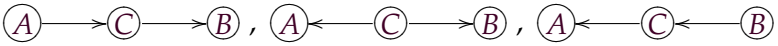
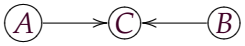


FIG. 3.8 : *Trois structures équivalentes encodant* $A \perp\!\!\!\perp B|C$.

Définition 3.4.1 (équivalence au sens de Markov) *Deux* DAG *sont* équivalents au sens de Markov *(noté* $\equiv$*) s'ils* encodent *la même décomposition de la loi jointe.*

Les trois structures de la figure 3.8 sont donc équivalentes (au sens de Markov)[ix] et représentent toutes la distribution $\mathbb{P}(A,B,C) = \mathbb{P}(A)\mathbb{P}(C|A)\mathbb{P}(B|C) = \mathbb{P}(A|C)\mathbb{P}(C)\mathbb{P}(B|C) = \mathbb{P}(A|C)\mathbb{P}(C|B)\mathbb{P}(B)$.
Par contre, la structure



qui encode l'indépendance conditionnelle $A \perp\!\!\!\perp B$, mais également $A \not\perp\!\!\!\perp B|C$, n'est pas équivalente à ces dernières ($\mathbb{P}(A,B,C) = \mathbb{P}(A)\mathbb{P}(B)\mathbb{P}(C|A,B)$).

Théorème 3.4.1 (Verma & Pearl (1990)) *Deux* DAG *sont équivalents si et seulement si ils ont le même squelette et les mêmes* V-structures.

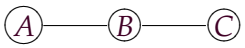
Théorème 3.4.2 *Toutes les cartes parfaites (3.2.4), si elles existent, sont équivalentes au sens de Markov.*

Définition 3.4.2 (arc réversible) *Un arc est dit* réversible *s'il n'intervient pas dans une V-structure et si lorsque son orientation est changée, aucun circuit et aucune nouvelle V-structure n'est créée.*

[ix] par la suite nous dirons seulement 'équivalentes'

Définition 3.4.3 (graphe essentiel) *Le* graphe partiellement dirigé sans circuit *(*PDAG *pour* Partially Directed Acyclic graph*) obtenu en conservant les arcs non réversibles et en transformant les arcs réversibles en arêtes (non dirigées) est appelé* graphe essentiel *(ou* Completed-PDAG *dans la terminologie de Chickering (1996)). Il représente sans ambiguïté la classe d'équivalence de Markov à laquelle appartient le* DAG *initial.*

Par exemple, les trois structures de la figure 3.8 peuvent être représentées pas le graphe essentiel suivant :



Définition 3.4.4 (PDAG instantiable) *Un* PDAG *est* instantiable *s'il est le représentant d'une classe d'équivalence de Markov.*

Dor & Tarsi (1992) ont proposé une méthode pour extraire le graphe essentiel d'un DAG quelconque.
Chickering (1996) a proposé une méthode pour créer un DAG faisant partie de la classe d'équivalence de Markov représentée par un graphe essentiel.

L'ensemble des équivalents de Markov n'est pas *vraiment* plus petit que l'ensemble de DAG. Steinsky (2003) a trouvé une formule récursive pour énumérer le nombre de classes d'équivalence de Markov de taille 1, c'est-à-dire celles qui ne contiennent qu'un seul DAG. Il a alors utilisé cette formule et la formule de l'équation 2.1, pour en déduire qu'il y a une asymptôte pour le quotient entre le nombre de classe d'équivalence de taille 1 et le nombre de DAG aux alentours de $0,07325$. Gillispie & Lemieux (2001) ont montré qu'il y a une asymptote pour le quotient entre la taille de l'espace des CPDAG et la taille de l'espace des DAG aux alentours de $0,2671$.

Donc, même si l'ensemble des classes d'équivalence de Markov est plus petit que l'ensemble des DAG, il n'en conserve pas moins la même complexité.

Gillispie & Lemieux (2001) ont déduit de cela que le quotient entre le nombre de classes d'équivalence de taille 1 et le nombre de classes d'équivalence possède une asymptôte aux alentours de $0,274$. Il est alors remarquable que plus d'un quart des classes d'équivalences de Markov ne contiennent qu'un seul élément. Ils ont également montré que la taille moyenne des classes d'équivalence de Markov semblait posséder une asymptote autour de $3,7$.

4

Une utilisation des modèles probabilistes : l'inférence

"Toute connaissance dégénère en probabilité."

Traité de la nature humaine, David Hume (1711 - 1776)

Sommaire

4.1 Méthodes d'inférence exactes 42
4.1.1 Messages locaux 42
4.1.2 Ensemble de coupe 42
4.1.3 Arbre de jonction 42
Moralisation : 43
Triangulation : 43
L'arbre de jonction : 43
4.1.4 Inversion d'arcs 44
4.1.5 Elimination de variables 44
4.1.6 Explication la plus probable 44
4.1.7 Méthodes symboliques 45
Query DAG*s concept* 45
Algorithme des restrictions successives 45
4.1.8 Méthodes différentielles 45
4.2 Méthodes d'inférence approchées 45
4.2.1 Simulation stochastique par Chaîne de Monte-Carlo 46
Algorithme de Metropolis-Hastings 46
Recuit simulé 47
Echantillonnage de Gibbs 47
4.2.2 Méthodes variationnelles 47
4.2.3 Méthodes de recherche de masse 48
4.2.4 *Loopy belief propagation* 48
4.2.5 Simplification du réseau 48

Une fois que nous possédons un modèle, l'idéal est de pouvoir en faire quelque chose. En présence d'un réseau bayésien, nous pouvons extraire un certain nombre d'informations.

En premier lieu, nous avons accès à la structure du réseau, celle-ci nous permet de savoir quels attributs sont dépendants ou non, pour cela nous utilisons la *d-séparation* et, par exemple, l'algorithme dit du *Bayes-Ball* (Shachter (1998)) qui permet de vérifier si une relation d'indépendance est représentée, ou non, dans la structure.

Ensuite, nous avons accès aux tables de probabilités conditionnelles, qui nous permettent de retrouver de la connaissance statistique. Néanmoins, la majeure partie des probabilités auxquelles nous voudrions avoir accès ne sont pas inscrites dans ces tables. Passons en revue les principales méthodes existantes pour les évaluer.

4.1 Méthodes d'inférence exactes

4.1.1 Messages locaux

Le première méthode d'inférence, introduite par Kim & Pearl (1983) et Pearl (1985), est celle des messages locaux, plus connue sous le nom de *polytree algorithm*. Elle consiste en une actualisation, à tout moment, des probabilités marginales, par transmission de messages entre variables voisines dans le graphe d'indépendance. Cette méthode ne fonctionne de manière exacte que lorsque le réseau bayésien possède une forme d'arbre (ou *polytree* en anglais), elle est donc à recommander dans ce cas.

Par ailleurs, il existe des adaptations de cette méthode (4.1.2, 4.1.5, 4.2.4) qui permettent d'utiliser les messages locaux même lorsque nous ne sommes pas en présence d'une structure arborescente.

4.1.2 Ensemble de coupe

L'algorithme *Loop Cutset Conditioning* a été introduit très tôt par Pearl (1986). Dans cette méthode, la connectivité du réseau est changée en instantiant un certain sous-ensemble de variables appelé l'ensemble de coupe (*loop cutset*). Dans le réseau résultant, l'inférence est effectuée en utilisant l'algorithme des messages locaux. Puis les résultats de toutes les instantiations sont combinés par leurs probabilités *a priori*. La complexité de cet algorithme augmente donc exponentiellement en fonction de la taille de l'ensemble de coupe.

4.1.3 Arbre de jonction

La méthode de l'arbre de jonction (aussi appellée *clustering* ou *clique-tree propagation algorithm*) a été introduite par Lauritzen & Spiegelhalter (1988) et Jensen, Lauritzen & Olesen (1990). Elle est aussi appelée méthode JLO (pour Jensen, Lauritzen, Olessen). Elle est applicable pour toute structure de DAG contrairement à la méthode des messages locaux. Néanmoins, s'il y a peu de circuits dans le graphe, il peut être préférable d'utiliser une méthode basée sur un ensemble de coupe. Cette méthode est divisée en cinq étapes qui sont :

- moralisation du graphe,
- triangulation du graphe moral,
- construction de l'arbre de jonction,
- inférence dans l'arbre de jonction en utilisant l'algorithme des *messages locaux*,

– transformation des potentiels de clique en lois conditionnelles mises à jour.

Moralisation :

La moralisation se décompose suivant les étapes suivantes :

– *'mariage des parents'* : pour les nœuds possédant plusieurs parents, liaison des parents deux à deux avec des arcs supplémentaires.

– récupération du squelette du graphe ainsi obtenu,

Nous obtenons alors un graphe non dirigé dit *moralisé*.

Définition 4.1.1 (graphe moral) *Si $\mathcal{G} = (X, \mathcal{E})$ est un* DAG, *alors $\mathcal{G}' = (X, \mathcal{E}')$ est un graphe moral associé à $\mathcal{G}$ si $\mathcal{G}'$ est un graphe non orienté tel que $\mathcal{E}' \geqslant \mathcal{E}$ et si $[\mathcal{E}(X_{i_1}, X_j) = 1$ et $\mathcal{E}(X_{i_2}, X_j) = 1]$ alors $\mathcal{E}'(X_{i_1}, X_{i_2}) = 1$ (*cf. *annexe C).*

La moralisation est une réécriture de $\mathbb{P}(X_1, \cdots, X_n)) = \prod_i \mathbb{P}(X_i, Pa(X_i)))$ par $\mathbb{P}(X_1, \cdots, X_n)) = \prod_i fct(X_i, Pa(X_i)))$ telle qu'il est toujours possible de retrouver la première expression. Après avoir retiré les orientations, nous pourrions croire que de l'information a été perdue, mais cela n'est pas le cas.

Triangulation :

Pour que les potentiels de toutes les lois conditionnelles soient associés à des sous-graphes complets, il suffit de procéder à la triangulation du graphe moral en y ajoutant des arêtes créant des raccourcis dans tout cycle de longueur 4 ou plus, nous obtiendrons alors *un* graphe moral dit *triangulé*.

Définition 4.1.2 (graphe triangulé) *Soit $\mathcal{G} = (X, \mathcal{E})$ est un graphe moral, un graphe triangulé pour $\mathcal{G}$ est un* graphe cordal *qui est également une carte d'indépendance pour la loi sous-jacente à $\mathcal{G}$.*

Remarquons qu'un graphe triangulé associé à un graphe moral n'est pas unique.

L'arbre de jonction :

Définition 4.1.3 (arbre de jonction) *Un* arbre de jonction *est une structure arborescente ayant l'ensemble des cliques d'un graphe triangulé comme ensemble de nœuds et telle que toutes cliques sur le chemin entre $\mathcal{C}_i$ et $\mathcal{C}_{i'}$ doit contenir les nœuds de $\mathcal{C}_i \cap \mathcal{C}_{i'}$ (propriété de l'intersection courante).*

Tout comme le graphe triangulé n'est pas unique, l'arbre de jonction ne l'est pas non plus. Par ailleurs, chaque nœud du graphe triangulé doit apparaitre dans au moins une clique utilisée dans la construction de l'arbre de jonction.

Propriété 4.1.1 (de l'intersection courante) *Pour toute clique $\mathcal{C}_i$, il y a une clique $\mathcal{C}_{j(i)}$ avec $j(i) < i$ telle que toute variable commune à $\mathcal{C}_i$ et $\mathcal{C}_{i'}$ où $i' < i$ appartient à $\mathcal{C}_{j(i)}$.*

Cette propriété nous guide dans la construction d'un arbre de jonction. En effet, pour que l'information sur une variable remonte jusque la clique initiale qui la contient, il est utile que cette variable appartienne à toutes les cliques intermédiaires. Jensen, Lauritzen & Olesen (1990) a donné une méthode pour cette construction et le résultat suivant :

Théorème 4.1.2 (Jensen, Lauritzen & Olesen (1990)) *A partir d'un graphe moral triangulé, il est toujours possible de construire un arbre de jonction.*

Les variables de l'arbre de jonction sont donc des groupes de variables du graphe d'origine. Par envois de messages (les potentiels de cliques) dans l'arbre de jonction, il est possible d'effectuer l'inférence dans n'importe quel DAG. Les potentiels de clique vont donc être mis à jour, puis il sera possible d'en déduire les lois marginales pour les variables d'origine (grâce à la propriété de l'intersection courante).

Même si cette méthode est plus rapide que le *Cut-Set conditioning*, elle possède le désavantage d'avoir une complexité en mémoire exponentielle en la taille du réseau tandis que la précédente ne nécessite qu'un espace mémoire linéaire en la taille du réseau.

4.1.4 Inversion d'arcs

Cette méthode a été introduite très tôt par Shachter (1986). Il s'agit de changer la direction des arcs en utilisant la règle de Bayes jusqu'à ce que le réseau soit transformé de telle manière que les nœuds de la requête soient des enfants des nœuds observés (Henrion (1990)). Cette méthode est à privilégier soit lorsque l'ensemble des nœuds de la requête est restreint soit lorsque nous savons par avance que nous allons interroger de nombreuses fois les mêmes nœuds en ayant à chaque fois les mêmes nœuds observés.

4.1.5 Elimination de variables

L'élimination de variables est décrite dans Zhang & Poole (1994). Cet algorithme supprime les variables une par une après avoir sommé sur celles-ci. Cette méthode a été généralisée dans Dechter (1998) par l'algorithme *Bucket Elimination*. Un ordre des variables doit être donné en entrée et sera alors l'ordre d'élimination des variables. Le nombre de calculs dépend alors de cet ordre puisqu'il influe sur la taille des facteurs futurs. Trouver le meilleur ordre équivaut au problème de trouver l'arbre de plus petite largeur dans le réseau (Dechter (1998)), ce qui est un problème NP-dur. Cette méthode est avantageuse lorsqu'un ordre d'élimination des variables est déjà connu ou si le réseau est peu dense mais avec de nombreux circuits.

4.1.6 Explication la plus probable

La méthode de l'explication la plus probable (MPE pour *Most probable explanation*) n'est pas réellement une technique d'inférence mais plutôt un problème d'inférence. Ce problème consiste en l'identification de l'état le plus probable. Il est possible d'adapter différentes méthodes d'apprentissage pour répondre à cette question. La technique la plus commune (et exacte, Lauritzen & Spiegelhalter (1988)) pour effectuer cette inférence consiste en le remplacement des signes *sommes* par des *max* et les signes *produits* par des *min* dans les formules de l'inférence classique. Il est possible d'adapter cette méthode pour trouver le deuxième cas le plus probable ou, plus généralement, le n-ième cas le plus probable.

Comme pour les autres problèmes d'inférence, il existe également des algorithmes approchés pour résoudre ce problème : par exemple, Guo, Boddhireddy & Hsu (2004) propose une méthode à base de colonies de fourmis.

4.1.7 Méthodes symboliques

L'inférence probabiliste symbolique (SPI pour *Symbolic Probabilistic Inference*) a été introduite dans Shachter, D'Ambrosio, & DelFabero (1990) et Li & D'Ambrosio (1994). Cette méthode est orientée par un but : *n'effectuer que les calculs nécessaires pour répondre à la requête*. Des expressions symboliques peuvent être obtenues en remettant à plus tard l'évaluation des expressions, et en les gardant sous forme symbolique.

Par ailleurs, Castillo, Gutiérrez, & Hadi (1996) ont proposé une autre technique d'inférence symbolique en modifiant les méthodes existantes d'inférences numériques et en remplacant les paramètres initiaux par des paramètres symboliques.

Ces méthodes ont le désavantage qu'il est difficile de calculer et de simplifier automatiquement des expressions symboliques mais, l'avantage qu'elles orientent les calculs intelligemment.

Citons deux implémentations de ces idées.

***Query* DAG*s concept* :**

Dans la méthode du *Query* DAG*s concept*, introduite dans Darwiche & Provan (1997), le réseau est préalablement transformé en une expression arithmétique appelée le *Query* DAG.

Algorithme des restrictions successives :

Cette méthode consistant en la construction d'une expression symbolique a été introduite par Mekhnacha, Ahuactzin, Bessière, Mazer & Smail (2006). Cette expression doit répondre à l'inférence demandée de manière à organiser la suite des sommes et produits à effectuer pour en minimiser le nombre. Pour minimiser ce nombre d'opérations, il faut trouver un bon ordre pour la marginalisation des variables. Des détails concernant cette méthode peuvent être trouvés dans Smail (2004). Le second objectif de cette méthode est que tous les calculs intermédiaires doivent produire une distribution de probabilité au lieu d'un potentiel. Elle est donc à recommander lorsque cette contrainte est nécessaire.

4.1.8 Méthodes différentielles

Les méthodes différentielles transforment un réseau bayésien en polynôme multivarié (Darwiche (2000)). Elles calculent ensuite les dérivées partielles de ce polynôme. Il est alors possible d'utiliser ces dérivées pour calculer les réponses à de nombreuses requêtes, et cela en temps constant. Cette méthode est très utile lorsque nous effectuons régulièrement les mêmes requêtes car maintenant nous pourrons y répondre en temps constant.

4.2 Méthodes d'inférence approchées

Effectuer une inférence exacte est un problème NP-difficile (Cooper (1992)). Ceci n'est pas étonnant, car il est possible de simuler un problème de *satisfaction de contraintes* à l'aide d'un réseau bayésien (en effet, en utilisant que des probabilités 0 et 1, les nœuds d'un réseau bayesien deviennent des portes logiques).

Lorsque la dimension (voir page 76) du réseau bayésien augmente, il est nécessaire d'utiliser de plus en plus de temps de calcul. Or, si les tables de probabilités conditionnelles ne se pas exactes (car évaluées à partir d'une base de cas peu représentative par exemple), l'interêt d'effectuer une inférence exacte avec ces valeurs approximatives n'est

plus probant. Dans ce cas, il peut être intéressant d'effectuer une inférence approchée pour économiser du temps de calcul.

Par ailleurs, pour certains réseaux bayésiens particuliers (par exemple, qui contiennent des nœuds discrets et des nœuds continus), il n'existe pas d'algorithme d'inférence exact, le seul recours pour évaluer des probabilités est alors d'utiliser des algorithmes d'inférence approchés.

4.2.1 Simulation stochastique par Chaîne de Monte-Carlo

Les méthodes MCMC (pour *Markov Chains Monte Carlo*) décrites dans Gilks, Richardson & Spiegelhalter (1996) permettent d'échantillonner des variables aléatoires en construisant une chaîne de Markov (voir 2.2.4.1). Les deux algorithmes à base de MCMC pour faire de l'estimation de densité les plus répandus, sont l'algorithme de Metropolis-Hastings et l'échantillonneur de Gibbs.

Ces méthodes statistiques approchées ne calculent pas exactement les lois marginales, mais en donnent une estimation. Elles permettent donc de traiter des applications de grande taille en temps raisonnable, ce qui n'est pas le cas des méthodes exactes. Les méthodes statistiques basées sur les principes de Monte-Carlo sont décrites en détail dans Robert & Castela (2004).

Le principe est ici d'effectuer un certain nombre de tirages aléatoires compatibles avec une loi. Pour cela, la décomposition de la loi jointe en produit de lois conditionnelles (voir équation 2.2) permet de mener les calculs. En pratique, chaque variable dont tous les parents sont connus est tirée aléatoirement, jusqu'à ce que toutes les variables aient été simulées. Pour que l'estimation soit fine, il faut alors effectuer un très grand nombre de simulations.

Par ailleurs, ces méthodes sont inefficaces quand certaines probabilités sont très faibles mais elles possèdent l'avantage d'être aisément implémentables et de ne pas être fermées : plus l'algorithme est arrêté tard, plus de cas seront simulés et plus l'évaluation des résultats sera précise.

Algorithme de Metropolis-Hastings :

L'algorithme de Metropolis-Hastings permet de simuler une loi de densité π qui n'est connue qu'à un facteur près, c'est-à-dire que l'on ne connaît que $\frac{\pi(x)}{\pi(y)}$. La transition de l'état x^{t-1} à x suivant la loi $q(x|x^{t-1})$ est alors acceptée avec la probabilité $\alpha(x, x^{t-1})$ définie par

$$\alpha(x, x^{t-1}) = \min\left(1, \frac{\pi(x)q(x|x^{t-1})}{\pi(x^{t-1})q(x^{t-1}|x)}\right)$$

Nous sommes alors en présence d'une chaîne de Markov de loi de transition $p(x, y) = q(x|y)\alpha(x, y)$ avec $y \neq x$. Pour simuler des exemples générés par π, il faut choisir la loi q. La vitesse de convergence dépendra principalement de ce point de départ. Remarquons que cette méthode ne génère pas d'échantillon *i.i.d* car la probabilité d'acceptation dépend de x^{t-1}.

Il suffit ensuite d'effectuer un comptage dans cette base d'exemples pour obtenir une approximation des probabilités recherchées.

Recuit simulé (ou *simulated annealing*) **:**

Cette méthode d'optimisation peut être vue comme un algorithme de Metropolis. Pour minimiser un critère $\mathscr{C}$, il est possible de simuler la loi $\pi(x) = \exp\left(\frac{-\mathscr{C}(x)}{T_i}\right)$, où T_i est la température (généralement définie par $T_i = T_0\beta^i$). Nous sommes donc en présence d'une chaîne de Markov dont le modèle de transition varie au cours du temps. Cette densité tend alors vers un *pic de Dirac*. Il est possible d'utiliser ce type de technique pour faire de la recherche de structure.

La méthode des Modes Conditionnels Itérés (ICM pour *Iterated Conditional Modes*) est une méthode sous-optimale basée sur un recuit-simulé à température nulle.

Echantillonnage de Gibbs :

Dans le cas où les lois conditionnelles sont connues et que l'objectif est de retrouver des lois jointes, il est possible d'utiliser l'échantillonneur de Gibbs (ou *Gibbs sampler*) adapté par Lauritzen (1996). Cette méthode a été introduite par Geman & Geman (1984). Supposons que l'on veuille approcher $\mathbb{P}(x, y)$ connaissant $\mathbb{P}(x|y)$ et $\mathbb{P}(y|x)$. Alors choisissons un point de départ (x_0, y_0). Générons $x_{t+1} \sim \mathbb{P}(x|y_{t-1})$ et $y_{t+1} \sim \mathbb{P}(y|x_{t-1})$. Le couple (x_t, y_t) est alors une chaîne de Markov et l'échantillon converge vers un échantillon qui aurait été généré par la loi $\mathbb{P}(x, y)$.

Dans le cas particulier des réseaux bayésiens, la loi $\mathbb{P}(x|y)$, se simplifie en $\mathbb{P}(x|F(x))$ (voir le paragraphe sur la frontière de Markov C.1.9). Cette méthode est donc particulièrement adaptée aux réseaux bayésiens où nous ne stockons que des distributions conditionnelles. Comme les autres méthodes basées sur la simulation stochastique, celle-ci construit une base d'exemples à partir de laquelle il sera possible d'évaluer les probabilités recherchées. Remarquons que cette méthode ne fonctionne que si les probabilités sont toutes strictement positives. La convergence peut être lente, notamment s'il y a des probabilités très faibles.

4.2.2 Méthodes variationnelles

Dans l'état *actuel* des choses, l'inférence dans les modèles graphiques probabilistes n'est possible que lorsque les distributions de probabilités des variables continues sont gaussiennes et n'ont pas de variables filles discrètes. Lorsque ce cas se présente, il faut avoir recours à des méthodes d'inférence approchée. Une technique pour cela serait de discrétiser toutes les variables, ou encore d'utiliser les méthodes variationnelles. Néanmoins, même lorsque nous ne sommes pas dans ce cas, ces méthodes ont l'avantage d'être peu complexes, et peuvent alors se révélées utiles.

Les méthodes variationnelles ont été introduites pour approcher l'apprentissage par maximum de vraisemblance en présence d'une base d'exemples incomplète. Elles sont alors utilisées pour approcher les intégrales nécessaires pour l'inférence bayésienne (*cf.* Lawrence (2000)) ou encore l'apprentissage bayésien comme proposé par Jordan, Ghahramani, Jaakkola & Saul (1998) et Wainwright & Jordan (2003).

L'approche variationnelle consiste en l'application de l'inégalité de Jensen introduisant une distribution approchée Q.

$$\begin{aligned}\ln \mathbb{P}(X|\Theta) &= \ln \sum_H \mathbb{P}(H, X|\Theta) \\ &= \ln \sum_H Q(H|X) \frac{\mathbb{P}(H, X|\Theta)}{Q(H|X)}\end{aligned}$$

$$\ln \mathbb{P}(X|\Theta) \geqslant \sum_H Q(H|X) \ln \frac{\mathbb{P}(H, X|\Theta)}{Q(H|X)} \tag{4.1}$$

En particulier, si la distribution que nous recherchons est exponentielle et de la forme :

$$\mathbb{P}(H, X|\Theta) = \frac{1}{Z} \exp(g(H, X|\Theta)) \tag{4.2}$$

où g est une fonction des variables aléatoires (discrètes) X et H, et Z un terme de normalisation. Le problème à résoudre devient alors :

$$\ln \mathbb{P}(X|\Theta) \geqslant \sum_H Q(H|X) g(H, X|\Theta) - \sum_H Q(H|X) \ln Q(H|X) - \ln Z \tag{4.3}$$

En mettant de côté le terme de normalisation, il reste à calculer les deux termes de la borne efficacement. Cela peut être fait en temps polynomial, car l'entropie de Q peut être bornée par valeurs inférieures en temps polynomial, et l'espérance de g pour la distribution Q peut également être bornée en temps polynomial. Remarquons que le calcul de l'équation 4.1 revient à calculer la divergence de Kullback-Leiberg entre Q et $\mathbb{P}$ (voir section 6.2). Les méthodes variationnelles, qui ont pour objectif de trouver le maximum de vraisemblance, doivent alors minimiser la KL-divergence (*cf.* section 6.2).

4.2.3 Méthodes de recherche de masse

Ces méthodes supposent qu'une petite partie de l'espace de définition des variables contient une grande partie de la masse de probabilité. Elles recherchent alors les instantiations de haute probabilité et les utilisent pour obtenir une bonne approximation (Henrion (1990) et Poole (1993)). Cette méthode est à utiliser lorsque nous voulons négliger les évènements de probabilités faibles.

4.2.4 *Loopy belief propagation*

L'algorithme des messages locaux de Pearl (1985) ne fonctionne de manière exacte que sur des structures en forme d'arbre. Cependant cette méthode a été généralisée dans Pearl (1988) pour effectuer de l'inférence *approchée* avec une structure de DAG quelconque. Murphy, Weiss & Jordan (1999) proposent une étude empirique de la convergence de cet algorithme en présence de cycles, et parfois, certains résultats trés mauvais peuvent être obtenus avec cette méthode.

4.2.5 Simplification du réseau

Pour effectuer l'inférence plus rapidement, il peut être envisageable de simplifier les paramètres, par exemple en mettant à zéro les probabilités trop faibles, ou encore de simplifier la structure, puis de faire de l'inférence exacte dans ce nouveau réseau. Kjærulff (1993) a été le premier à donner ce type de méthode en proposant de retirer du graphe les liens *faibles* au sens de la dépendance causale. De nombreuses autres techniques ont ensuite été proposées : retirant des arcs, retirant des nœuds ou encore recherchant l'arbre le plus proche du réseau, il est possible de trouver de nombreuses références dans Guo & Hsu (2001).

Les méthodes d'inférence par simplification du réseau sont principalement à construire et à utiliser dans des buts particuliers, par exemple dans le cas de l'utilisation d'une simplification pertinente pour le problème traité.

5

Apprentissage des paramètres

"Calculer la probabilité d'un événement n'a aucun sens
une fois que l'on sait qu'il s'est produit.
L'apparition de la vie, celle des dinosaures, celle des Hommes,
a résulté d'un grand nombre de bifurcations
dans le cours des processus se déroulant sur notre planète ;
chacune de ces bifurcations s'est produite
alors que de nombreuses autres étaient possibles ;
chacune avait une probabilité faible,
mais il fallait bien qu'une de ces possibilités se produise."

La science à l'usage des non-scientifiques, 2003
Albert Jacquard, né en 1925.

Sommaire

5.1 Hypothèses pour l'apprentissage de paramètres 50
5.2 Apprentissage des paramètres avec données complètes . . . 51
5.2.1 Sans *a priori* 51
Le problème des zéros et du sur-apprentissage 52
5.2.2 Avec *a priori* de Dirichlet 53
Une méthode bayésienne, l'espérance *a posteriori* . . 53
Une méthode classique, le maximum *a posteriori* . . . 54
5.3 Apprentissage des paramètres avec données incomplètes . . 54
5.3.1 Qu'entendons-nous par base incomplète 54
Les données dites MCAR 54
Les données dites MAR 55
Les données dites NMAR 55
5.3.2 Estimation à partir d'une base d'exemples incomplète . . . 56

Nous avons vu diverses techniques pour effectuer de l'inférence bayésienne dans les réseaux bayésiens. Mais avant de pouvoir utiliser ces modèles, il faut pouvoir les construire. La construction d'un réseau bayésien se décompose en trois étapes distinctes :

l'étape qualitative : la recherche des relations d'influence pouvant exister entre les variables prises deux à deux. Ceci emmène à une représentation graphique des relations entre les variables, lorsqu'il est possible de les coder par un graphe (voir section 2.1),

l'étape probabiliste : elle introduit l'idée qu'une distribution jointe définie sur les variables a généré la base d'observations et suppose que le graphe créé précédemment est compatible avec celle-ci (voir page 14),

l'étape quantitative : elle consiste en l'évaluation numérique des distributions de probabilités conditionnelles.

Pour la suite de cette section, nous allons supposer que nous connaissons déjà la structure du réseau bayésien à apprendre, et qu'il nous reste à évaluer les probabilités conditionnelles. Il s'agit donc de spécifier les tables de probabilités, souvent appelées paramètres du réseau bayésien[i]. Grâce à l'équation 2.2, la loi jointe peut être représentée par l'ensemble $\{\theta_i = \mathbb{P}(X_i|Pa(X_i))\}_{1\leqslant i\leqslant n}$, où chaque θ_i est la table de probabilités conditionnelles de X_i connaissant l'état de ses parents $Pa(X_i)$. En particulier, le paramètre θ_{ijk} représentera la probabilité $\mathbb{P}(X_i = k|Pa(X_i) = j)$.

Ces probabilités peuvent être données par un expert, ou encore apprises automatiquement à partir d'une base d'exemples. Dans la suite, nous allons montrer comment les apprendre automatiquement. Nous allons également voir comment mêler connaissance *a priori* et apprentissage par le biais des *a priori* de Dirichlet.

5.1 Hypothèses pour l'apprentissage de paramètres

Soit $\mathcal{G}$ un DAG. Commençons par introduire la notion de distribution de Dirichlet, nous verrons par la suite pourquoi elle est si intéressante.

Définition 5.1.1 (distribution de Dirichlet) *Soient $\alpha_1, \alpha_2 \cdots \alpha_r$ tous strictement positifs. Une* distribution de Dirichlet d'exposants $\alpha_1, \alpha_2, \cdots, \alpha_r$ *(également appelés les* hyperparamètres *pour ne pas les confondre avec les paramètres θ_i) possède une densité de probabilité de la forme suivante :*

$$Dir(\theta|\alpha_1, \cdots, \alpha_r) = \frac{\Gamma(\sum_{i=1}^r \alpha_i)}{\prod_{i=1}^r \Gamma(\alpha_i)} \prod_{i=1}^r \theta^{\alpha_i - 1} \tag{5.1}$$

où Γ est la fonction Gamma, qui satisfait $\Gamma(x+1) = x\Gamma(x)$ et $\Gamma(1) = 1$.

Pour l'apprentissage des paramètres de réseaux bayésiens, nous allons faire les hypothèses suivantes :

i.i.d. **Hypothèse d'indépendance** : les exemples de la base sont supposés *indépendants et identiquement distribués*.

[i] Remarquons que la structure d'un réseau bayésien est également un paramètre du réseau à determiner. Cependant, dans la littérature, le terme *paramètre* est souvent reservé aux tables de probabilités conditionnelles.

Dir. **Hypothèse de Dirichlet** : les densités de probabilités des paramètres admettent des densités exponentielles de la forme de Dirichlet.

Mod. **Hypothèse de modularité de la vraisemblance** :

$$\mathbb{P}(X_i|\theta_i, Pa(X_i), \mathscr{G}) = \mathbb{P}(X_i|\theta_i, Pa(X_i)) \tag{5.2}$$

I.P. **Hypothèse d'indépendance paramétrique** : Nous supposerons que les paramètres sont mutuellement indépendants, c'est-à-dire

$$\mathbb{P}(\theta_1, \cdots, \theta_n|\mathscr{G}) = \prod_{i=1}^{n} \mathbb{P}(\theta_i|\mathscr{G}) \tag{5.3}$$

5.2 Apprentissage des paramètres avec une base d'exemples complète

Soit $X = (X_1, \cdots, X_n)$, n variables aléatoires, et $\mathbf{D}$ une base d'exemples complète. Supposons de plus, que la structure du réseau bayésien soit connue et nommée $\mathscr{G}$. Nommons également $\Theta = (\theta_1, \cdots, \theta_n)$ les paramètres du modèle à estimer.

La variable X_i peut prendre ses valeurs parmi $\{1, \cdots, r_i\}$ (ou $\{x_{i1}, \cdots, x_{ir_i}\}$). Soit N_{ij} le nombre de fois où l'ensemble des variables parentes de X_i dans $\mathscr{G}$, $Pa(X_i)$, prend sa j-ième configuration dans la base $\mathbf{D}$, pour tout $j \in \{1, \cdots, q_i\}$, et soit N_{ijk} le nombre de fois où X_i vaut k et $Pa(X_i)$ prend sa j-ième valeur dans la base $\mathbf{D}$, pour tout $k \in \{1, \cdots, r_i\}$. Définissons également $N_i = \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} = \sum_{j=1}^{q_i} N_{ij}$, le nombre de fois où l'attribut X_i prend sa k-ième valeur dans la base $\mathbf{D}$.

5.2.1 Sans *a priori*

Pour estimer les paramètres à partir de données, nous devons choisir un critère à optimiser. En particulier l'estimation par *maximum de vraisemblance* est sans biais. Il s'agit d'évaluer les paramètres Θ qui maximisent $\mathbb{P}(\mathbf{D}|\mathscr{G}, \Theta)$.
Pour un exemple $(x_1, \cdots, x_n)$ de la base $\mathbf{D}$, en utilisant l'hypothèse **Mod.**, nous avons

$$\mathbb{P}(x_1, \cdots, x_n|\theta, \mathscr{G}) = \prod_{i=1}^{n} \mathbb{P}(x_i|pa(x_i), \theta) = \prod_{i=1}^{n} \theta_{ij_0k_0}$$

pour des couples particuliers (j_0, k_0) correspondants à l'entrée $(x_1, \cdots, x_n)$. Par ailleurs, en utilisant l'hypothèse **i.i.d.** (page 50), nous obtenons l'expression de la vraisemblance d'un réseau bayésien $\mathscr{B} = (\mathscr{G}, \Theta)$ par rapport à la base $\mathbf{D}$.

$$L(\Theta, \mathbf{D}) = \mathbb{P}(\mathbf{D}|\mathscr{G}, \Theta) = \prod_{i=1}^{n} \prod_{l=1}^{N} \theta_{ij_lk_l} \tag{5.4}$$

Or le terme $\theta_{ij_lk_l}$ peut être identique pour plusieurs valeurs de l. Donc la vraisemblance se réécrit plus simplement.

$$L(\Theta, \mathbf{D}) = \mathbb{P}(\mathbf{D}|\mathscr{G}, \Theta) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}} \tag{5.5}$$

Souvent pour des raisons de stabilité numérique nous utiliserons la *log*-vraisemblance (et lorsque les θ_{ijk} sont supposés tous non nuls).

$$LL(\Theta, \mathbf{D}) = \ln(\mathbb{P}(\mathbf{D}|\mathscr{G}, \Theta)) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \ln(\theta_{ijk}) \tag{5.6}$$

Le maximum de *log*-vraisemblance est alors le même critère que le maximum de vraisemblance et revient à satisfaire le problème suivant (Naïm, Wuillemin, Leray, Pourret & Becker (2004)) :

$$\begin{cases} \widehat{\Theta} = arg \max_{\Theta} LL(\Theta, \mathbf{D}) \\ \sum_{k=1}^{r_i} \widehat{\theta}_{ijk} = 1, \quad \forall i \in \{1, \cdots, n\} \; \forall j \in \{1, \cdots, q_i\} \end{cases} \tag{5.7}$$

La vraisemblance s'écrit alors uniquement avec les paramètres indépendants.

$$LL(\Theta, \mathbf{D}) = \ln(\mathbb{P}(\mathbf{D}|\mathscr{G}, \Theta)) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left(\sum_{k=1}^{r_i - 1} \left(N_{ijk} \ln(\theta_{ijk}) \right) + N_{ijr_i} \log \left(1 - \sum_{k'=1}^{r_i - 1} \theta_{ijk'} \right) \right) \tag{5.8}$$

Les dérivées partielles de la log-vraisemblance valent alors

$$\frac{\partial LL(\Theta, \mathbf{D})}{\partial \theta_{ijk}} = \frac{N_{ijk}}{\theta_{ijk}} - \frac{N_{ijr_i}}{1 - \sum_{k'=1}^{r_i - 1}} = \frac{N_{ijk}}{\theta_{ijk}} - \frac{N_{ijr_i}}{\theta_{ijr_i}} \tag{5.9}$$

Cette vraisemblance atteint donc son maximum au point $\widehat{\theta}_{ijk}$ tel que

$$\frac{N_{ijk}}{\widehat{\theta}_{ijk}} = \frac{N_{ijr_i}}{\widehat{\theta}_{ijr_i}} \quad \text{donc} \quad \frac{N_{ij1}}{\widehat{\theta}_{ij1}} = \frac{N_{ij2}}{\widehat{\theta}_{ij2}} = \cdots = \frac{N_{ijr_i - 1}}{\widehat{\theta}_{ijr_i - 1}} = \frac{N_{ijr_i}}{\widehat{\theta}_{ijr_i}} = \frac{\sum_{k=1}^{r_i} N_{ijk}}{\sum_{k=1}^{r_i} \widehat{\theta}_{ijk}} = \sum_{k=1}^{r_i} N_{ijk}$$

et les valeurs des paramètres obtenus par maximum de vraisemblance sont :

$$\widehat{\theta_{ijk}^{\text{MV}}} = \frac{N_{ijk}}{\sum_{k=1}^{r_i} N_{ijk}} \tag{5.10}$$

Le problème des zéros et du sur-apprentissage :

Lorsque certaines configurations n'existent pas dans la base d'exemples, ce résultat nous dit que les paramètres correspondants vont être évalués par une valeur nulle. Si, par la suite, le réseau bayésien obtenu est utilisé pour faire de l'inférence, et qu'il est questionné sur une de ces configurations, il prédira que cette configuration possède une probabilité nulle car le zéro est absorbant dans les produits. Or ce n'est pas parce qu'une configuration n'a pas été observée qu'elle est impossible.

Pour éviter cela, il est possible d'introduire des *pseudo-comptes* supplémentaires sur toutes les configurations de sorte qu'elles soient 'virtuellement' représentées dans la base d'exemples. Dans la pratique, nous n'allons pas construire une nouvelle base d'exemples mais utiliser des *a priori* de Dirichlet. Alors toute configuration, même si elle n'est pas représentée dans la base d'exemples se verra attribuée une probabilité non nulle. Elle pourra alors être prédite comme cause ou diagnostic car il n'y aura plus le problème du zéro.

L'estimateur de maximum de vraisemblance avec *a priori* de Dirichlet, appelé l'estimateur de *maximum a posteriori* n'est cependant plus un estimateur sans biais.

De plus, le fait de ne pas utiliser d'*a priori* peut engendrer un sur-apprentissage Si la base d'exemples a été prise dans un cas de fonctionnement particulier du système, de nombreuses configurations pourraient se voir attribuer des probabilités très faibles ou nulles, et d'autres des valeurs plutôt trop élevées. Les *a priori* permettent alors de lisser ces problèmes et d'obtenir un modèle plus générique.

5.2.2 Avec *a priori* de Dirichlet

D'après l'hypothèse **Dir.**, nous avons les lois *a priori* suivantes sur les paramètres, pour tout i

$$\mathbb{P}(\theta_i|\mathscr{G}) = Dir(\theta_i|\alpha_1, \cdots, \alpha_n)$$

Nous pouvons alors remarquer (voir équations 5.14 et 5.16), que le choix des $(\alpha_1, \cdots, \alpha_n)$ représente des décomptes *a priori* pour les différentes configurations possibles.

Après calcul (cf. Robert (1994)), nous obtenons les lois *a posteriori* suivantes

$$\mathbb{P}(\theta_i|\mathbf{D}, \mathscr{G}) = Dir(\theta_i|\alpha_1 + N_1, \cdots, \alpha_n + N_n) \tag{5.11}$$

De plus, nous savons que

$$\int \theta_k Dir(\theta|\alpha_1 + N_1, \cdots, \alpha_n + N_n)d\theta = \frac{\alpha_k + N_k}{\sum_{i=1}^{n} \alpha_i + N_i} \tag{5.12}$$

Une méthode bayésienne, l'espérance *a posteriori* (EAP) :

Pour une approche bayésienne, il faut alors sommer sur toutes les valeurs des paramètres possibles donc

$$\widehat{\mathbb{P}}(X_1 = x_{1k_1}, \cdots, X_n = x_{nk_n}|\mathbf{D}, \mathscr{G}) = \mathbb{E}_{\mathbb{P}(\theta_{\mathscr{G}}|\mathbf{D},\mathscr{G})}\big[\prod_{i=1}^{n} \theta_{ijk}\big] \tag{5.13}$$

$$\widehat{\mathbb{P}}(X_1 = x_{1k_1}, \cdots, X_n = x_{nk_n}|\mathbf{D}, \mathscr{G}) = \int \prod_{i=n}^{n} \theta_{ijk} \mathbb{P}(\theta_{\mathscr{G}}|\mathbf{D}, \mathscr{G})d\theta_{\mathscr{G}}$$

où j_i est la configuration courante des parents de X_i, et encore,

$$\widehat{\mathbb{P}}(X_1 = x_{1k_1}, \cdots, X_n = x_{nk_n}|\mathbf{D}, \mathscr{G}) = \prod_{i=n}^{n} \int \theta_{ijk} \mathbb{P}(\theta_{\mathscr{G}}|\mathbf{D}, \mathscr{G})d\theta_{\mathscr{G}}$$

En utilisant l'équation 5.12, nous obtenons donc

$$\widehat{\mathbb{P}}(X_1 = x_{1k_1}, \cdots, X_n = x_{nk_n}|\mathbf{D}, \mathscr{G}) = \prod_{i=1}^{n} \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}$$

avec

$$\widehat{\theta_{ijk}^{\text{EAP}}} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \tag{5.14}$$

À présent, nous voyons bien que les α_{ijk} jouent un rôle d'occurrences *a priori* supplémentaires où $X_i = {x_i}^k$ et $Pa(X_i) = pa(X_i)_{j_i}$ avant que l'on ait pris connaissance de la base d'exemples.

Une méthode classique, le maximum *a posteriori* (MAP) :

Pour une approche classique, notons qu'il est possible d'utiliser l'estimateur de *maximum de vraisemblance* qui sera ici appelé *maximum a posteriori*.

$$\mathbb{P}(\theta|\mathbf{D},\mathcal{G}) \propto \mathbb{P}(\mathbf{D}|\theta,\mathcal{G})\mathbb{P}(\theta|\mathcal{G}) = L(\theta|\mathbf{D},\mathcal{G})\mathbb{P}(\theta|\mathcal{G})$$

Les hypothèses **Dir.** et **I.P.** (page 51) nous donnent alors

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

La loi *a posteriori* des paramètres est alors (avec l'hypothèse **i.i.d.**)

$$\mathbb{P}(\theta|\mathbf{D}) \propto \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i}\theta_{ijk}^{N_{ijk}+\alpha_{ijk}-1} \tag{5.15}$$

En utilisant un calcul similaire à celui de la section 5.3.2, nous nous apercevons que la distribution de probabilité $\mathbb{P}$ obtenue par *maximum a posteriori* lorsqu'une distribution de Dirichlet d'exposants $\alpha_1,\cdots,\alpha_n$ est mise *a priori* sur les paramètres est donnée par la formule suivante.

$$\widehat{\theta_{ijk}^{\text{MAP}}} = \frac{N_{ijk}+\alpha_{ijk}-1}{\sum_{k=1}^{r_i} N_{ijk}+\alpha_{ijk}-1} \tag{5.16}$$

Nous avons vu diverses méthodes pour effectuer l'apprentissage de paramètres d'un réseau bayésien à partir d'une base d'exemples complète, voyons à présent comment adapter ces méthodes en présence de données incomplètes.

5.3 Apprentissage des paramètres avec une base d'exemples incomplète

5.3.1 Qu'entendons-nous par base incomplète

Soit $\mathbf{D} =< \mathbf{O},\mathbf{H} >= (d_{li})_{m\times n}$ la base d'exemples, avec $\mathbf{O}$ et $\mathbf{H}$ représentant respectivement les parties complètes et manquantes de $\mathbf{D}$ et d_{li} étant la valeur du nœud i pour l'exemple l. Nous noterons également $\mathbf{O}_l$ l'ensemble des variables observées dans le l-ième exemple et $\mathbf{H}_l$ l'ensemble des variables non observées pour le l-ième exemple, d'où $\mathbf{D}_l =< \mathbf{O}_l,\mathbf{H}_l >= (d_{li})_{1\leqslant i\leqslant n}$.

Pour manipuler des données incomplètes efficacement, nous devons savoir de quelle nature elles sont. Rubin (1976) différencie trois types de données manquantes selon le mécanisme qui les a générées .

Appelons $R = (r_{li})_{m\times n}$ la matrice où $r_{li} = 1$ si $d_{li} =$ 'manquant' et 0 sinon. Alors, si Θ sont les paramètres représentant la loi qui a généré la base $\mathbf{D}$ et μ sont les paramètres représentant la loi qui a généré la base $\mathbf{H}$, nous pouvons remarquer que

$$\mathbb{P}(\mathbf{O},\mathbf{H},R|\Theta,\mu) = \mathbb{P}(\mathbf{O},\mathbf{H}|\Theta)\times\mathbb{P}(R|\mathbf{O},\mathbf{H},\mu) \tag{5.17}$$

Les données dites MCAR pour *Missing Completely At Random* :

Dans ce cas, les exemples avec des données manquantes ne peuvent pas être différenciés de ceux complètement observés. Les données manquantes représentent juste une absence de mesure fortuite et isolée. L'absence de mesure suit donc, par exemple, une loi du type $\mathbb{P}(X =$ 'manquant'$) = \varepsilon$. Comme le fait qu'une variable soit présente ou

non est complètement aléatoire, il est donc impossible de modéliser cet état 'manquant' par un nouvel état car cela introduirait un biais. Le fait que la donnée soit manquante est indépendant à la fois des données observées et des autres données incomplètes de la base, c'est-à-dire :

$$\mathbb{P}(R|\mathbf{O},\mathbf{H},\mu) = \mathbb{P}(R|\mu)$$

Les données dites MAR pour *Missing At Random* :

Ici les exemples avec des données manquantes sont différents de ceux complètement observés. Néanmoins le processus qui a généré les données manquantes est prédictible à partir des autres variables de la base plutôt que d'être dû exclusivement à la variable dont la valeur est absente. Par exemple, pour des raisons de coût ou de fiabilité dans une configuration particulière du système, une observation pourrait ne pas être faite systématiquement. Ce cas est donc plus général que le précédent car ici le fait que la donnée soit manquante ne dépend que de l'*état observé* du système, mais reste indépendant des valeurs des variables non observées, d'où :

$$\mathbb{P}(R|\mathbf{O},\mathbf{H},\mu) = \mathbb{P}(R|\mathbf{O},\mu)$$

Nous sommes en présence du cas où une donnée n'est pas mesurée systématiquement, mais la probabilité qu'elle soit manquante dépend de l'état des autres variables observées. Pour les données dites MCAR et MAR, le mécanisme de suppression des données est dit *ignorable* car il est possible d'inférer les données manquantes à l'aide des données observées.

Les données dites NMAR pour *Non Missing At Random* sont également appelées données *non-ignorables* :

Dans ce cas, le mécanisme qui provoque l'absence d'une variable n'est pas aléatoire et ne peut pas non plus être complètement prédit à l'aide des autres variables observées de la base d'exemples. Il n'est donc pas possible de simplifier $\mathbb{P}(R|\mathbf{O},\mathbf{H},\mu)$. Le fait qu'une donnée soit manquante dépend à la fois de l'état observé du système, mais également de l'état 'manquant' des autres variables. Nous pouvons donc également penser que les variables latentes, c'est-à-dire des variables pertinentes pour la modélisation du problème qui ne sont jamais observées peuvent également avoir une influence sur le fait qu'une autre variable soit observée ou non.

Pour ce type de données, il est donc possible de considérer l'état 'manquant' comme un état supplémentaire sans introduire de biais car le fait qu'une variable soit manquante n'est pas aléatoire mais dépend d'une configuration particulière du système. Néanmoins, il faudrait être capable de différencier les différents mécanismes qui font qu'une variable est présente ou non, et ajouter un état supplémentaire pour chacun de ces mécanismes car une variable peut être manquante pour diverses raisons, et confondre ces raisons pourrait alors introduire un biais.

Par la suite, nous supposerons que nous sommes en présence d'une base de données incomplète suivant un mécanisme MAR ou MCAR. Ainsi, nous possédrons toute l'information nécessaire pour estimer la distribution des données manquantes dans la base d'exemples.

5.3.2 Estimation à partir d'une base d'exemples incomplète

L'estimation de paramètres à partir de bases d'exemples incomplètes a fait l'objet de nombreuses méthodes. Voici un rapide survol des méthodes les plus représentatives.

Etude des cas complets Lorsque les données sont MCAR, il est possible de déterminer les paramètres et la structure du réseau bayésien à partir des entrées complètes de la base. Comme les données manquantes sont supposées l'être aléatoirement, nous construisons ainsi un estimateur sans biais. Cette méthode sera appelée par la suite CCA pour *Complete Cases Analysis*. Elle est aussi connue sous les noms de *casewise / listwise data deletion*.

Etude des cas disponibles Un avantage des réseaux bayésiens est qu'il suffit que seules les variables X_i et $Pa(X_i)$ soient observées pour estimer la table de probabilité conditionnelle correspondante (cf équation 5.10). Dans ce cas, il est alors possible d'utiliser tous les exemples (même incomplets) où ces variables sont complètement observées. Dans l'exemple précédent, en supposant que X_i possède trois parents, nous aurions 819 exemples en moyenne pour estimer les paramètres correspondants. Cette méthode sera nommée ACA pour *Available Cases Analysis*. Elle est aussi connue sous le nom de *pairwise data deletion*.

Les méthodes CCA et ACA sont ainsi des techniques qui utilisent soit une sous-base complète, soit plusieurs sous-bases complètes, même lorsque nous sommes en présence d'une base incomplète.

Certaines méthodes dites de *substitutions* remplissent la base d'exemples de différentes manières pour effectuer plusieurs apprentissages à partir de bases d'exemples *complètes*. En voici trois exemples.

Substitution par la moyenne Cette méthode, rapide, possède l'inconvénient de modifier les statistiques essentielles et d'atténuer les variations de scores.

hot deck imputation Ici, les données manquantes sont remplacées par des données évaluées avec une méthode similaire aux kPPV (les k plus proches voisins) sur le reste de la base.

Imputation par régression Une fonction de régression est utilisée pour estimer les données manquantes. Après supposition que les valeurs sont correctement prédites, elles sont utilisées pour remplir la base.

Maximum de vraisemblance Il s'agit ici de remplacer les données manquantes par les valeurs qui réalisent le maximum de vraisemblance. Pour faire cela, il est possible d'utiliser un algorithme de type EM (*Expectation Maximisation*) introduit par Dempster, Laird & Rubin (1977) et revu par Neal & Hinton (1998) pour une description claire dans le cadre des réseaux bayésiens. Cette méthode est assez gourmande en temps de calcul, mais reste efficace comparée aux autres.

Il s'agit de la méthode que nous allons utiliser, la voici plus en détails.

Apprentissage des paramètres avec EM Nous allons, à présent, présenter une adaptation de l'algorithme EM pour l'apprentissage des paramètres d'un réseau bayésien introduite dans Spiegelhalter & Lauritzen (1990) et Heckerman (1999).

ALGORITHME 2 L'algorithme EM de Dempster adapté pour l'apprentissage de paramètres d'un réseau bayésien.

1: Tirage des probabilités au hasard (mais toutes non nulles) pour les paramètres manquants (il est possible de fixer des *a priori* de Dirichlet).

$$\theta_{ijk}^{(0)} \equiv \mathbb{P}(X_i = k / Pa(X_i) = j)$$

2: **Répéter**

3: **Expectation :** Utilisation des paramètres courants ($\theta_{ijk}^{(t)}$) pour estimer l'espérance d'apparition des différentes configurations.

$$\mathbb{E}[N_{ijk}] = \sum_{l=1}^{N} \mathbb{P}(X_i = k | Pa(X_i) = j, \theta_{ijk}^{(t)}, \mathbf{O}_l) \tag{5.18}$$

4: **Maximisation :** Estimation des nouveaux paramètres par *maximum de vraisemblance* (ou *maximum a posteriori*) en utilisant l'espérance des statistiques essentielles obtenue à l'étape précédente.

$$\theta_{ijk}^{(t+1)} = \frac{\mathbb{E}[N_{ijk}] \ (+\alpha_{ijk})}{\sum_k \mathbb{E}[N_{ijk}] \ (+\alpha_{ijk})} \tag{5.19}$$

5: **Jusqu'à** ce que $\theta_{ijk}^{(t+1)} \simeq \theta_{ijk}^{(t)}$

Soit **O** l'ensemble des variables observées dans la base d'exemples **D**. Le principe de la méthode réside en deux étapes, décrites brièvement dans l'algorithme 2. Pour une description plus complète de la méthode EM, de ses variantes, et une preuve de convergence de celle-ci, se reporter au chapitre 10.
L'algorithme EM possède des propriétés de convergence vers un optimum local.

Remarquons que cet algorithme fournit, après convergence, une valeur des paramètres et non une distribution pour ces paramètres.

Cet algorithme est compatible avec l'introduction d'*a priori* de Dirichlet, termes entre parenthèses de l'équation 5.19 dans l'étape de maximisation de l'algorithme EM.

Bauer, Koller & Singer (1997) proposent une méthode pour accélérer la convergence de l'algorithme EM pour l'apprentissage des paramètres.

Gibbs Sampling L'échantillonneur de Gibbs (Geman & Geman (1984) et Gilks, Richardson & Spiegelhalter (1996)) possède l'avantage sur l'algorithme EM que les données simulées permettent d'obtenir une estimation empirique de la variance. Néanmoins, contrairement à l'algorithme EM, l'échantillonneur de Gibbs traite chaque donnée manquante comme une quantité à évaluer et la vitesse de convergence baisse énormément lorsque le nombre de données manquantes croît.

The Robust Bayesian Estimator Lorsque les données ne sont plus MAR, les performances des algorithmes EM et de l'échantillonneur de Gibbs décroissent (Spirtes, Glymour & Scheines (1993)). Dans le cas où aucune information sur la nature des données manquantes n'est fournie ou si elles sont NMAR, il est alors possible de recourir à la méthode de l'*estimateur bayésien robuste* introduite par Ramoni & Sebastiani (2000) et

Sebastiani & Ramoni (2001). Cette méthode utilise alors des intervalles pour estimer différentes probabilités conditionnelles. La première phase consiste en la découverte des extrémités de ces intervalles, puis les intervalles sont affinés au fur et à mesure que l'information devient disponible.

Raw maximum likelihood Cette méthode est également appelée FIML pour *Full Information Maximum Likelihood* décrite dans Wothke & Arbuckle (1996). Elle utilise toutes les données disponibles pour générer les statistiques suffisantes qui ont le maximum de vraisemblance. Pour faire cela elle construit tous les meilleurs moments des 1*er* et 2*nd* ordre (très gourmand).

Imputations multiples Plusieurs bases complètes sont générées à l'aide de nombreux remplissages de la base incomplète (*cf.* Rubin (1981)). Puis différents apprentissages sont effectués à partir de ces différentes bases et les résultats sont combinés. Cette méthode est très gourmande en temps de calcul et pose le problème d'être capable de trouver une bonne manière de combiner les résultats.

Approximate bayesian bootstrap

Cette technique est une mise en œuvre de la méthode d'*imputation multiple* (Rubin (1987)). Pour chaque variable, les valeurs manquantes sont remplacées par échantillonnage en utilisant une distribution construite à partir des données complètes.

Le *Bagging* décrit dans Domingos (1997) et le *Boosting* décrit dans Freund & Schapire (1996) sont des implémentations particulières du *bootstrap*. Les différences entre ces méthodes portent sur la manière de construire la loi à partir des données complètes, ainsi que sur la manière de combiner les résultats en un classifieur unique.

Conclusion

Cette introduction très générale sur les modèles graphiques existants et les méthodes d'inférence qui leur sont associées permet de comprendre ce qu'est un réseau bayésien ainsi que les avantages que peuvent posséder ces modèles. Nous avons également passé en revue différentes déclinaisons de ces derniers. Ceci permet de nous rendre compte que les réseaux bayésiens sont un formalisme unificateur pour différentes modélisations ayant été développées dans la littérature pour des problématiques aussi vastes que la classification, l'extraction d'information, ou encore simplement pour la modélisation.

Les calculs basés sur la formule d'inversion de Bayes, illustrés dans l'annexe A.2 et généralisés grâce aux travaux de Pearl introduits dans la section 4.1, montrent la puissance et la simplicité d'un tel formalisme. Néanmoins, ces modèles peuvent alors s'avérer difficiles à construire manuellement lorsque le nombre de nœuds augmente. Pour cela, des méthodes de créations automatiques de réseaux bayésiens doivent être développées. Certaines heuristiques existent déjà avec des avantages et inconvénients issus du choix d'un compromis entre l'exhaustivité des heuristiques et leur complexité.

Dans la prochaine partie, nous nous proposons de comparer empiriquement certaines techniques existantes pour découvrir la structure d'un réseau bayésien à partir de bases d'exemples complètement observées. Nous proposons et testons alors une méthode d'initialisation de certaines méthodes classiques permettant de garantir leur stabilité tout en conservant, voire en améliorant, la qualité de leurs résultats.

Deuxième partie

Apprentissage de Structure à partir de données complètes

Introduction

Pour résoudre la problématique de l'apprentissage de structure des réseaux bayésiens, il est possible de prendre en compte plusieurs informations.
Nature des données : Les données peuvent être continues ou discrètes, pour la suite des travaux nous n'allons considérer que le cas où celles-ci sont discrètes. Si elles ne le sont pas dans la base d'exemples initiale, cette dernière sera transformée de telle manière que les variables continues soient discrétisées (par exemple de manière optimale tel qu'il est indiqué dans El Matouat, Colot, Vannoorenberghe & Labiche (2000). La catégorisation possède l'avantage de ne pas géner l'inférence par la suite, par contre, elle possède le désavantage de modéliser moins précisément le procesus initial.
Nature de la base d'exemples : Les attributs de la base d'exemples peuvent être complètement ou partiellement observés. Dans cette section, nous allons parler d'apprentissage de structure lorsque la base est complète. L'utilisation de bases d'exemples incomplètes sera discutée dans la troisième partie.
Existence de variable(s) latente(s) : Ce type de variable est très particulier. Il s'agit d'attributs pertinents pour la modélisation du système qui ne sont jamais observés. Par exemple, en apprentissage non-supervisé, la variable classe n'est jamais observée, cependant nous comprenons qu'il est indispensable d'introduire un nœud la représentant dans le réseau bayésien.
Connaissance *a priori* : De l'information *a priori* peut être donnée sur le problème à résoudre. Par exemple, si nous savons que le système suit un fonctionnement dynamique, il est possible de prendre en compte cette information lors du procesus d'apprentissage. Par ailleurs, lorsque nous voulons modéliser un système, il est pertinent de discuter préalablement avec des personnes qualifiées qui connaissent ce dernier. Il est ensuite possible d'utiliser ces informations pour l'apprentissage de structure, par exemple en décidant de conserver un lien durant toute la phase d'apprentissage. L'utilisation de connaissance *a priori* ne sera pas évoquée par la suite, car pour pouvoir profiter de cet atout, il faut utiliser des techniques spécifiques adaptées à la connaissance du problème. Or, par la suite, nous allons rester généralistes, sans nous concentrer sur une application particulière.

Nous allons faire l'hypothèse de variables discrètes, et n'ayant pas eu l'apport d'information *a priori* de la part d'un expert, hormis celle qui peut être simplement représentée sous forme d'*a priori* de Dirichlet. Néanmoins, l'apport de la connaissance de la variable 'classe' sera utilisé pour développer des méthodes plus spécifiques à la classification (le classifieur de Bayes naïf par exemple), ne considérant donc pas toutes les variables de la même manière. Pour la suite de

Nbre de nœuds	2	3	4	5	6	7
Nbre de DAG	3	25	543	29 281	3 781 503	1 138 779 265

Nbre de nœuds	8	9	10
Nbre de DAG	783 702 329 343	1 213 441 454 842 881	4 175 098 976 430 589 143

TAB. 1 : *Nombre de* DAG *existant en fonction du nombre de nœuds.*

ce chapitre nous allons également supposer que la base d'exemples, notée **D**, est complètement observée.

L'apprentissage de structure est un problème très difficile, en particulier à cause de la taille de l'espace de recherche. Robinson (1977) a montré que le nombre de structures différentes pour n nœuds est donné par la formule de récurrence de l'équation 2.1 dont les résultats sont illustrés sur la table 1.

Comme l'équation 2.1 est super-exponentielle, il est difficile d'effectuer un parcours exhaustif de l'espace de recherche en un temps raisonnable dès que le nombre de nœuds dépasse 8. La plupart des méthodes d'apprentissage de structure utilisent alors une heuristique de recherche dans l'espace des graphes dirigés sans circuits (DAG, pour *directed acyclic graphs*).
L'heuristique de recherche peut principalement être de deux types :

- soit sur l'espace de recherche lui même, c'est-à-dire en restreignant la recherche à un sous-ensemble de l'espace des DAG,
- soit sur le parcours de l'espace de recherche, par exemple en utilisant une technique de recherche du type gradient.

Bien sùr, il est possible de mélanger de ces deux types d'heuristiques.

Il existe principalement deux types de techniques pour effectuer de la recherche de structure de réseau bayésien.

1°) Le premier type, plus intuitif, consiste en la création d'une table d'indépendances conditionnelles entre les différents attributs en effectuant des tests statistiques sur la base d'exemples. Or, comme toute table d'indépendance n'est pas codable dans une structure de DAG, il va falloir suivre certaines règles pour construire cette dernière. Certaines techniques utilisant cette approche vont être introduites dans le chapitre 6.

2°) Le seconde méthode d'apprentissage de structure utilise une fonction de score. L'heuristique de recherche consistera alors à optimiser ce score. Différentes fonctions de score seront introduites au chapitre 7. Puis différentes techniques seront exposées dans le chapitre 8.

Chacune de ces deux méthodologies d'apprentisage de structure possèdent des avantages et des inconvénients, certains seront mis en évidence dans le chapitre 9.

Des *méthodes mixtes* ont alors été proposées pour profiter des avantages de chacune de ces méthodologies. Certaines seront évoquées dans la section 8.4.

Pour effectuer l'apprentissage de structure à base de score, il est possible d'avoir recours à deux types de parcours de l'espace de recherche :

– les *techniques de recherche déterministes* sont alors souvent complexes, elles nécessitent parfois une initialisation particulière et convergent vers un optimum local,
– les *techniques de recherche non-déterministes* peuvent être plus rapides mais elles ne donnent alors plus de garanties sur l'optimalité du résultat trouvé. Ces dernières approches seront évoquées dans la section 8.3.

Hypothèses pour l'apprentissage de structure

Pour l'apprentissage de structure, nous allons faire les hypothèses suivantes en supplément des hypothèses utilisées pour l'apprentissage des paramètres :

Mkv **Hypothèse de Markov** : pour un réseau bayésien donné, toute variable est indépendante de ses non descendantes sachant ses variables parentes,

Fid **Hypothèse de fidélité** : un DAG $\mathcal{G}$ et une loi de probabilité $\mathbb{P}$ sont supposés fidèles, c'est-à-dire que les relations d'indépendance valides pour $\mathbb{P}$ sont celles données par l'hypothèse de Markov.

LP **Hypothèse de localité paramétrique** : si $\mathcal{G}_1$ et $\mathcal{G}_2$ sont deux structures telles que le nœud X_i possède le même ensemble de parents pour chacune, alors

$$\mathbb{P}(\theta_i|\mathcal{G}_1) = \mathbb{P}(\theta_i|\mathcal{G}_2) \tag{1}$$

CP **Hypothèse de complétude** : toutes les variables nécessaires pour modéliser le problème sont connues. En particulier, il n'y a pas de variables autres que celles dont nous avons connaissance.

Dans cette partie, nous supposerons de plus que la **base d'exemples** est **complète**. Nous allons commencer par introduire les méthodes d'apprentissage de structure à partir de tests statistiques. Puis nous parlerons de la notion de score et des différents scores classiquement utilisés. Nous exposerons ensuite différentes méthodes d'apprentissage de structure à partir de base d'exemples complète. Nous concluerons cette partie par quelques expérimentations et interprétations.

6

Méthodes basées sur la recherche d'indépendances conditionnelles

"Il entre dans toutes les actions humaines
plus de hasard que de décision."

André Gide (1869 - 1951)

Sommaire

6.1 Les tests statistiques 68
6.1.1 Le test du χ^2 68
6.1.2 Le test du rapport de vraisemblance 69
6.2 L'information mutuelle 70
La divergence de Kullback-Leibler 70
L'information mutuelle 70
L'information mutuelle moyenne conditionnelle 71
6.3 Les algorithmes PC et IC 71
6.4 L'algorithme QFCI 72
6.5 L'algorithme BNPC 72
6.6 L'algorithme PMMS 72
6.7 L'algorithme *Recursive Autonomy Identification* 73
6.8 La recherche de motifs fréquents corrélés 73
6.9 *The Grow-Shrink algorithm* 73
6.10 Discussion 74

6.1 Les tests statistiques

6.1.1 Le test du χ^2

Le test du khi-deux (noté χ^2 et décrit dans Chernoff & Lehmann (1954)) permet de valider des hypothèses concernant une propriété concrète contenue dans une base de cas. Il existe différentes méthodes basées sur le test du χ^2, principalement les tests d'adéquation et les tests d'homogénéité. La version originale a été proposée par Pearson (1892) et les différentes méthodes de tests statistiques sont bien décrites dans Levin (1999). Dans notre cadre, nous l'utiliserons principalement pour le test d'indépendance basé sur les tableaux de contingence.

En particulier, en présence d'une base d'exemples sur deux attributs, nous pouvons construire un tableau d'occurences conjointes des différentes valeurs pour ces variables, puis de comparer aux valeurs théoriques quand l'indépendance des deux variables est supposée. Nous avons donc l'hypothèse nulle H_0 et l'hypothèse alternative H_1 qui valent :

$$H_0 : \text{les deux attributs sont indépendants}$$

$$H_1 : \text{les deux attributs sont dépendants}$$

En fonction d'un seuil, nous choisissons ensuite de conserver l'hypothèse nulle ou non. Théoriquement, cela revient à comparer la valeur de

$$\chi^2 = \sum_{i=1}^{s_1} \sum_{i=1}^{s_2} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{6.1}$$

où s_1 et s_2 représentent respectivement les tailles des deux attributs, O_{ij} représente la fréquence observée lorsque le premier attribut prend sa i-ème valeur tandis que le deuxième attribut prend sa j-ème valeur dans la base d'exemples et E_{ij}, la fréquence théorique supposée par l'hypothèse nulle (donc ici l'indépendance des deux attributs).

Or, comme la somme de k variables aléatoires gaussiennes normalisées et centrées élevée au carré ($X = \sum_{i=1}^{k} X_i^2$ où $X_i \sim \mathcal{N}(O,1)$) suit une loi du χ^2 à k degrés de liberté, notée χ_k^2 où

$$\chi_k^2(x) = \frac{(1/2)^{k/2}}{\Gamma(k/2)} x^{k/2-1} e^{-x/2} \tag{6.2}$$

où Γ est la fonction Gamma d'Euler, et que les termes $\dfrac{O_{ij} - E_{ij}}{\sqrt{E_{ij}}}$ sont supposés suivre une distribution normale (*loi des grands nombres*), alors l'équation 6.1 est supposée suivre une distribution du χ^2 à $(s_1 - 1) \cdot (s_2 - 1)$ degrés de liberté (en effet, les dernières cases du tableau de contingence sont entièrement determinées).
Nous devons ensuite comparer les valeurs χ^2 et x tel que

$$\int_0^x \chi^2_{(s_1-1)\cdot(s_2-1)}(t)dt = 1 - \alpha \tag{6.3}$$

où α est un seuil choisi, typiquement $0,05$, et choisir de conserver l'hypothèse nulle ou de l'infirmer lorsque $\chi^2 > x$.

Cette méthode peut être simplement adaptée pour le test d'indépendance *conditionnelle*.

Remarquons par ailleurs que cette technique peut être utilisée pour tester si un ensemble de n variables sont corrélées ou non ($n \geqslant 2$). Dans ce cas, il faut toujours

construire le tableau de contingence (ici en n dimensions) et comparer la valeur du χ^2 à celle trouvée en fonction de la valeur du seuil pour la distribution $\chi^2_{(s_1-1)\cdot(s_2-1)\cdots(s_n-1)}$. Ceci permet par exemple de détecter si une configuration conjointe particulière de deux variables a une influence forte sur une troisième alors que chacune des valeurs des deux premières variables prises séparément ont peu d'influence sur cette troisième variable.

6.1.2 Le test du rapport de vraisemblance

Le test du G^2, ou test du rapport de vraisemblance (*Likelihood Ratio Test*), permet d'évaluer lequel, parmi deux modèles, représente au mieux les données. Ce test approche le test du χ^2 lorsque la taille de la base d'exemples $\mathbf{D}$ (*i.i.d.*) devient grande.

Les deux modèles considérés doivent être de même nature et le deuxième doit être plus complexe que le premier. Sa complexité s'exprime par le fait qu'il possède plus de paramètres. Les hypothèses à tester sont alors :

$$H_0 : \text{le modèle } \mathcal{M}_0 \text{ est supposé plus représentatif que le modèle } \mathcal{M}_1$$

$$H_1 : \text{le modèle } \mathcal{M}_1 \text{ est supposé plus représentatif que le modèle } \mathcal{M}_0$$

Dans ce cas, il est possible de calculer le rapport de log-vraisemblance suivant.

$$G^2(\mathcal{M}_0, \mathcal{M}_1) = -2\log\left(\frac{\sup\{L(\mathcal{M}_0, \theta_0|\mathbf{D})|\theta_0 \in \Theta_0\}}{\sup\{L(\mathcal{M}_1, \theta_1|\mathbf{D})|\theta_1 \in \Theta_1\}}\right) \tag{6.4}$$

La valeur G^2 obtenue est alors comparée à la valeur critique de la distribution du χ^2 pour n degrés de liberté et un degré de confiance α car la loi de l'équation 6.4 converge asymptotiquement vers une loi du χ^2.

La vraisemblance étant un produit de termes, la log-vraisemblance est alors une somme de termes. Pour l'exemple où il faut décider s'il faut mettre un lien entre deux attributs ou non, soit $\mathcal{M}_0$, le modèle nul où la dépendance est supposée, et $\mathcal{M}_1$, celui qui suppose l'indépendance des attributs, et soient s_1 et s_2 les tailles respectives des deux attributs alors nous obtenons

$$G^2 = \sum_{i=1}^{s_1}\sum_{j=1}^{s_2} N_{ij}\log\left(\frac{\mathbb{P}(\text{configuration}_{ij}|\mathcal{M}_0, \theta_0)}{\mathbb{P}(\text{configuration}_{ij}|\mathcal{M}_1, \theta_1)}\right) = \sum_{i=1}^{s_1}\sum_{j=1}^{s_2} N_{ij}\log\left(\frac{O_{ij}}{E_{ij}}\right) \tag{6.5}$$

où N_{ij} représente le nombre de fois où la configuration (i, j) apparaît dans la base d'exemples, O_{ij} les comptes estimés à partir du modèle $\mathcal{M}_0$ (typiquement $O_{ij} = N_{ij}$ si $\mathcal{M}_0$ est l'hypothèse nulle pour deux attributs) et E_{ij} les comptes estimés à partir du modèle $\mathcal{M}_1$ (typiquement $E_{ij} = N_{i.} \times N_{.j}$ si $\mathcal{M}_1$ est le modèle supposant l'indépendance des deux attributs).

Le nombre n de degrés de liberté est alors la différence entre le nombre de paramètres du modèle $\mathcal{M}_1$ et le nombre de paramètres du modèle $\mathcal{M}_0$. Comme pour le test du χ^2, nous comparons alors la valeur obtenue avec la valeur de la fonction de répartition d'une loi du χ^2 avec le nombre de degrés de liberté correspondant pour décider de conserver l'hypothèse nulle ou non.

Ce test peut alors très bien être utilisé comme critère pour décider lequel parmi plusieurs modèles améliore le mieux une représentation courante (hypothèse nulle).

Dans le cas des bases d'exemples très petites, il est préférable d'utiliser un test de Fisher plutôt qu'un test du χ^2 ou qu'un test de rapport de vraisemblance.

6.2 L'information mutuelle

La divergence de Kullback-Leibler

En se basant sur les hypothèses de Wiener (1948), le nombre $h_i = -\log\left(\mathbb{P}(x_i)\right)$ représente le nombre moyen de *nats* (ou de *bits* si le log est en base 2) nécessaire pour encoder la valeur x_i (***Théorème de Shannon***). L'entropie est alors définie comme étant l'espérance de la variable aléatoire $-\log(\mathbb{P}(X))$ noté $H(X) = -\sum_x \mathbb{P}(X = x)\log(\mathbb{P}(X = x))$. Elle représente le nombre moyen de *nats* nécessaire pour encoder des données provenant de la variable aléatoire X et évalue donc l'uniformité d'une variable.

A présent, nous allons définir le *gain d'information*, qui est une mesure d'information particulière. Par exemple, en présence d'un second évènement y, nous nous interrogeons sur le gain d'information que représente y par rapport à x ?

Pour cela, nous construisons la partition $y_i = \{y \cap x_i\}_{i \in I}$ et les probabilités associées ($\mathbb{P}(y_i) = \mathbb{P}(x_i, y)$). Le *gain d'information* de Kullback-Leibler apporté par l'évènement y sur l'évènement x est alors défini par la formule de l'équation 6.6 (Kullback (1952)).

$$KL(\mathbb{P}(x)||\mathbb{P}(x,y)) = \sum_i \mathbb{P}(x_i)\log\left(\frac{\mathbb{P}(x_i)}{\mathbb{P}(x_i,y)}\right) \tag{6.6}$$

La divergence de Kullback-Leibler est une mesure *asymétrique* de l'éloignement de deux distributions. Elle évalue le nombre de *nats* supplémentaires pour encoder des données provenant de $\mathbb{P}$ en utilisant un code optimisé pour encoder des données provenant de $\mathbb{Q}$ (dans le cas précédent, elle évalue comment la connaissance de y permet de compresser les données x_i).

$$KL(\mathbb{P}||\mathbb{Q}) = \sum_i \mathbb{P}(x_i)\log\left(\frac{\mathbb{P}(x_i)}{\mathbb{Q}(x_i)}\right) \tag{6.7}$$

Remarquons que cette formule a une forme très proche de celle de l'équation 6.5 et peut être adaptée aux réseaux bayésiens (*cf.* page 99).

Parfois, il est possible de rencontrer l'appellation de *distance* de Kullback-Leibler, or cette fonction n'est pas symétrique et ne peut pas être une distance. Néanmoins, il existe des versions symétrisées de ce critère, ne serait-ce que $KL_{sym}(\mathbb{P},\mathbb{Q}) = \frac{1}{2}(KL(\mathbb{P}||\mathbb{Q}) + KL(\mathbb{Q}||\mathbb{P}))$ lorsque nous voulons travailler avec une vraie distance ou encore l'information mutuelle lorsque nous voulons seulement travailler avec une mesure symétrique.

L'information mutuelle

Chow & Liu (1968) ont introduit le critère d'information mutuelle IM, qui peut simplement être exprimé à l'aide de la distance de Kullback-Leibler par

$$IM(X;Y) = KL(\mathbb{P}(X,Y)||\mathbb{P}(X)\mathbb{P}(Y)) = \sum_x\sum_y \mathbb{P}(x,y)\log\frac{\mathbb{P}(x,y)}{\mathbb{P}(x)\mathbb{P}(y)} \tag{6.8}$$

$$IM(X;Y) = \sum_{i=1}^{s_1}\sum_{j=1}^{s_2}\frac{O_{ij}}{N}\log\left(\frac{O_{ij}\times N}{E_{ij}}\right) \tag{6.9}$$

Où O_{ij} représente le nombre de fois où la configuration $x = i$ et $y = j$ apparaît dans la base d'exemples, $E_{ij} = N_{i\cdot} \times N_{\cdot j} = \left(\sum_{j=1}^{s_2} O_{ij}\right) \times \left(\sum_{i=1}^{s_1} O_{ij}\right)$ et N est le nombre de cas dans la base.

L'information mutuelle représente alors le nombre de *nats* que deux variables ont en commun. Cette mesure permet donc de décider, s'il est avantageux de connecter deux variables plutôt que de les conserver indépendantes.

L'information mutuelle moyenne conditionnelle

Il est possible de définir une *Information mutuelle conditionnelle* simplement en utilisant la définition de l'information mutuelle et en remplaçant les probabilités par des probabilités conditionnelles. L'information mutuelle (moyenne) conditionnelle est alors définie par l'équation 6.10.

$$\begin{aligned} IM(X;Y|Z) &= H(X|Z) - H(X|Y,Z) \\ IM(X;Y|Z) &= \sum_z KL(\mathbb{P}(X,Y,Z=z)||\mathbb{P}(X|Z=z)\mathbb{P}(Y|Z=z)\mathbb{P}(Z=z)) \\ &= \sum_z \sum_x \sum_y \mathbb{P}(x,y,z) \log\left(\frac{\mathbb{P}(x,y,z)}{\mathbb{P}(x|z)\mathbb{P}(y|z)\mathbb{P}(z)}\right) \\ &= \sum_z \sum_y \sum_x \mathbb{P}(x,y,z) \log\left(\frac{\mathbb{P}(x,y|z)}{\mathbb{P}(x|z)\mathbb{P}(y|z)}\right) \\ &= \sum_z \sum_x \sum_y \mathbb{P}(x,y,z) \log\left(\frac{\mathbb{P}(x,y,z)\mathbb{P}(z)}{\mathbb{P}(x,z)\mathbb{P}(y,z)}\right) \end{aligned} \tag{6.10}$$

Une évaluation empirique de l'information mutuelle conditionnelle peut alors être donnée par l'équation suivante.

$$IM(X,Y|Z) = \sum_{k=1}^{s_3} \left[\sum_{i=1}^{s_1} \sum_{j=1}^{s_2} O_{ijk} \log\left(\frac{O_{ijk} \cdot \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} O_{ijk}}{\left(\sum_{j=1}^{s_2} O_{ijk}\right) \cdot \left(\sum_{i=1}^{s_1} O_{ijk}\right)} \right) \right] \tag{6.11}$$

Où O_{ijk} représente le nombre de fois où la configuration $x = i$ et $y = j$ et $z = k$ apparaît dans la base d'exemples.

Cette définition sera alors utilisée dans l'algorithme TANB (pour *Tree Augmented Naive bayes classifier*), introduit en section 8.1.2.

Remarquons que l'information mutuelle (moyenne conditionnelle) peut se calculer pour plus de deux variables à l'aide de la formule 6.12. Cette formule est une règle dite de chaînage des information proche du théorème de Bayes généralisé (*c.f.* équation B.1).

$$IM(X_1,\ldots,X_n|Y) = \sum_{i=1}^{n} IM(X_i;Y|X_{i-1},\ldots,X_1) \tag{6.12}$$

6.3 Les algorithmes PC et IC

Une des premières méthodes de recherche d'indépendances conditionnelles efficace proposée fût celle de l'algorithme PC (pour *Peter and Clark*, les prénoms des inventeurs de la méthode). Elle a été introduite par Spirtes, Glymour & Scheines (1993). Elle utilise un test statistique (χ^2 ou $\mathcal{G}^2$ originellement) pour évaluer s'il y a indépendance conditionnelle entre deux variables. Il est alors possible de reconstruire la structure du réseau bayésien à partir de l'ensemble des relations d'indépendances conditionnelles découvertes. En pratique, un graphe complètement connecté sert de point de départ, et lorsqu'une indépendance conditionnelle est détectée à l'aide d'un test statistique, l'arc correspondant est retiré. Les tests statistiques sont effectués suivant un ordre donné préalablement aux variables. Dash & Druzdzel (1999) ont alors montré que cet ordre avait une grande influence sur le résultat rendu par cette méthode.

Un algorithme de principe similaire (IC, pour *Inductive Causation*) a été introduit à la même époque par Pearl & Verma (1991). La principale différence entre ces deux algorithmes est que l'agorithme PC est initialisé avec un graphe complètement connecté

et raisonne par cartes d'indépendances successives jusqu'à obtenir une *I-map* minimale tandis que l'algorithme IC est initialisé avec un graphe vide et raisonne par cartes de dépendances successives jusqu'à obtenir une *D-map* maximale.

La variante IC* de l'algorithme IC, permet de détecter si éventuellement il peut exister des variables latentes (voir Pearl (2000) pour plus de détails). De même, Spirtes, Glymour & Scheines (2000) introduisent une version augmentée de PC qui détecte l'existence de variables latentes et la nomment FCI (pour *Fast Causal Inference*). Ces algorithmes ont alors une complexité exponentielle en nombre de tests d'indépendance.

Une amélioration de la méthode FCI est décrite dans la thèse de Zhang (2006). Elle est simplement appelée AFCI pour *Augmented FCI* car cette méthode utilise plus de règles d'inférence causale que la méthode FCI.

6.4 L'algorithme QFCI

Les méthodes précédentes sont très sensibles au bruit et ne sont pas très efficaces lorsqu'il y a peu d'exemples par rapport au nombre de variables. L'algorithme QFCI (pour *Quantitative Fast Causal Inference*) introduit par Badea (2003) est une amélioration de FCI qui utilise les résultats des tests statistiques pour quantifier la confiance dans les liens découverts. Cette méthode se décompose en cinq phases. Les quatre premières sont très proches de celles de la méthode FCI. La dernière étape utilise la confiance dans les V-structures pour orienter les arcs.

6.5 L'algorithme BNPC

Une autre méthode a été plus récemment proposée par Cheng, Greiner, Kelly, Bell & Liu (2002) nommée BN-PC-B pour *Bayes Net Power Constructor* (et B, car Cheng, Greiner, Kelly, Bell & Liu (2002) introduisent deux algorithmes, le premier, BN-PC-A, nécessitant un ordre d'énumération des variables comme paramètre d'entrée). La méthode BNPC effectue la plupart de son travail dans l'espace des graphes non orientés en trois phases. La première phase consiste à rechercher une structure initiale arborescente (avec un principe décrit dans la section 8.1.1). La deuxième phase recherche des ensembles de *séparation* entre les variables pour décider s'il faut relier les nœuds correspondants, pour ce faire elle utilise des tests d'indépendance conditionnelle en conditionnant sur ces ensembles de séparation (*cut-set*). La version B de l'algorithme utilise alors l'information mutuelle conditionnelle en tant que test d'indépendance *quantitatif*. La troisième phase commence par essayer de retirer des arcs superflus. On obtient alors un graphe non dirigé (un PDAG *non instantiable* dans la plupart des cas). Cette phase se termine avec une heuristique d'orientation pour ce graphe. Cet algorithme est également appelé TPDA pour *Tree Phase Dependency Analysis*. Il est polynomial en nombre de tests d'indépendances (en $\mathcal{O}(n^4)$).

6.6 L'algorithme PMMS

Un autre algorithme polynomial a été proposé par Brown, Tsamardinos & Aliferis (2005). Cette méthode s'appelle PMMS pour *Polynomial Max-Min Skeleton*. Elle utilise une procédure appelée PMMPC pour *Polynomial Max-Min Parents and Children* qui retrouve des ensembles de voisins pour un nœud. L'algorithme PMMS est une variante

polynomiale de l'algorithme MMHC pour *Max-Min Hill Climbing* introduit par Tsamardinos, Brown & Aliferis (2005) qui utilisent alors la procédure MMPC qui est une version exacte non polynomiale de PMMPC.

Cet algorithme consiste en la découverte de la couverture de Markov des différents nœuds, puis en la construction du squelette de la structure en utilisant cette information. Cette première phase est très proche de la méthode de Cheng, Greiner, Kelly, Bell & Liu (2002), elle utilise également l'information mutuelle. Sa complexité est identique, soit en $\mathcal{O}(n^4)$ pour le nombre de tests d'independance.

Ensuite, pour obtenir une structure orientée, un algorithme glouton est utilisé, mais sa recherche est restreinte en utilisant la couverture de Markov connue. Le score utilisé est le score *BDeu* (voir section 7.2.2).

6.7 L'algorithme *Recursive Autonomy Identification*

Cette méthode, introduite par Yehezkel & Lerner (2005), apprend la structure par l'utilisation récursive de tests d'indépendance conditionnelle d'ordre de plus en plus grand. Comparativement aux autres méthodes qui cherchent à simplifier la structure avant de l'orienter, celle-ci effectue les deux tâches conjointement avec une troisième qui consiste en la séparation hiérarchique en sous-structures. Cette troisième tâche permet de diminuer le nombre de tests avec de grands ensembles de conditionnement tout en améliorant la précision en classification des modèles obtenus d'après Yehezkel & Lerner (2005). Sa complexité dans le pire des cas reste néanmoins la même que pour la méthode PC.

6.8 La recherche de motifs fréquents corrélés

Les méthodes présentées jusqu'à présent ne peuvent pas détecter des situations où des états de plusieurs variables ont une influence conjointe sur un état particulier d'une tierce variable. Aussem, Kebaili, Corbex & Marchi (2006) proposent alors une technique basée sur des tests du χ^2 pour l'indépendance conditionnelle et des tests de corrélation pour identifier des conjonctions de variables impliquées dans des relations causales en limitant le nombre de tests d'indépendance conditionnelle. Pour ce faire la méthode repose sur l'extraction d'ensembles fréquents corrélés minimaux grâce à un algorithme par niveau dans le treillis des parties de l'ensemble des variables aléatoires considérées.

De manière analogue, Godenberg & Moore (2004) propose une méthode d'apprentissage spécifique au cas ou le nombre d'attributs est très élevé. Cette méthode consiste en une adaptation de la découverte des ensembles fréquents, une méthode populaire par ailleurs dans la communauté de la recherche d'information, pour construire un réseau bayésien lorsque le nombre de nœuds et d'exemples sont tous deux de plusieurs centaines de milliers.

6.9 *The Grow-Shrink algorithm*

Margaritis (2003) introduit une méthode nommée *the Grow-Shrink algorithm* pour apprendre la structure d'un réseau bayésien en utilisant des tests statistiques. Cette méthode consiste en l'identification de la frontière de Markov de chaque nœud, puis

en l'utilisation de cette connaissance pour apprendre l'ensemble des parents de chaque nœud. La méthode se termine alors par une phase d'orientation des arêtes non encore orientées.

6.10 Discussion

Les méthodes de recherche d'indépendance conditionnelle nécessitent d'effectuer un nombre de tests statistiques exponentiel avec le nombre de nœuds.

En pratique, ces méthodes se limitent à de faibles tailles d'ensemble de conditionnement. En effet, lorsque la taille de l'ensemble de conditionnement augmente, il ne reste que très peu d'exemples pour tester l'indépendance conditionnelle. Les résultats des tests statistiques deviennent alors arbitraires. Certaines méthodes plus récentes permettent de se limiter à un nombre de tests polynomial (*cf.* table 6.1).

Méthodes	Complexité en nombre de tests statistiques	
PC (*sect.* 6.3)	*exponentielle*	en $\mathcal{O}(n^n) \downarrow \mathcal{O}(n^k)$
IC (*sect.* 6.3)	*exponentielle*	en $\mathcal{O}(n^n) \downarrow \mathcal{O}(n^k)$
IC* (*sect.* 6.3)	*exponentielle*	en $\mathcal{O}(n^n) \downarrow \mathcal{O}(n^k)$, non suffisance causale
FCI (*sect.* 6.3)	*exponentielle*	en $\mathcal{O}(n^n) \downarrow \mathcal{O}(n^k)$, non suffisance causale
QFCI (*sect.* 6.4)	*exponentielle*	en $\mathcal{O}(n^n) \downarrow \mathcal{O}(n^k)$
BNPC-A (*sect.* 6.5)	*polynomiale*	en $\mathcal{O}(n^2)$ mais nécéssite un ordre
BNPC-B/TPDA (*sect.* 6.5)	*polynomiale*	en $\mathcal{O}(n^4)$
MMHC (*sect.* 6.6)	*exponentielle*	
PMMS (*sect.* 6.6)	*polynomiale*	en $\mathcal{O}(n^4)$
RAI (*sect.* 6.7)	*exponentielle*	mais environ $\frac{1}{3}$ à $\frac{1}{2}$ plus faible que PC
MFC (*sect.* 6.8)	*exponentielle*	mais peut détecter plus de dépendances mettant en œuvre plus de 2 variables
GSA (*sect.* 6.9)	*exponentielle*	en $\mathcal{O}(n^4 2^n) \downarrow \mathcal{O}(n^4 2^k)$

TAB. 6.1 : *Comparaison des complexités de différents algorithmes d'identification de structure à base de tests statistiques. Le symbole $\downarrow$ signifie que la complexité de ces méthodes peut être rendue polynomiale en utilisant un nombre maximum k de parents admissibles par nœud (n est le nombre de nœuds).*

Ces méthodes ne rendent pas exactement un DAG en sortie, mais un graphe essentiel (voir définition 3.4.3), également appellé PAG pour *Partial Ancestral Graph* [i] ou encore un squelette pour PMMS.

Les méthodes de recherche de dépendances conditionnelles ont l'avantage d'obtenir une structure pour laquelle chaque arc orienté du CPDAG peut être considéré comme une relation de cause à effet. Le graphe est plus aisément interprétable, par contre il n'est que partiellement dirigé.

Dans la prochaine section nous allons introduire différentes fonctions de score qui peuvent être utilisées pour effectuer de l'identification de structure, puis nous introduirons quelques méthodes de recherche.

[i] Un graphe essentiel est encore appelé PDAG dans la terminologie de Pearl. Nous n'utiliserons pas cette appellation car elle est ambiguë. Nous préfèrerons alors la terminologie de Chickering : CPDAG, ou la terminologie francophone : graphe essentiel.

7

Fonctions de score

"Raisonner : Peser des probabilités sur la balance du désir."

Le Dictionnaire du Diable, 1911
Ambrose Bierce (1842 - 1914)

Sommaire

7.1 Avant-propos 76
7.1.1 Quand utiliser un score ? 76
7.1.2 Dimension d'un réseau bayésien 76
7.1.3 Principe du rasoir d'Occam 77
7.1.4 Score décomposable 77
7.1.5 Score équivalent 78
7.2 La vraisemblance marginale : la mesure bayésienne 78
7.2.1 Le score bayésien 78
7.2.2 Les autres scores bayésiens 79
Le score BDe 79
Le score $BDeu$ 80
Le score bayésien généralisé $BD\gamma$ 80
7.3 Les principaux scores pondérés 80
7.3.1 Approximation du score bayésien par Laplace 80
7.3.2 Les scores basés sur des critères informatifs 81
Le score AIC 81
Le score BIC 82
Approximation $BIC - MV$ de l'approximation de Laplace 82
Autres critères de score 82
7.3.3 La longueur de description minimum 83
7.4 Autres mesures adaptées à la classification 84

7.1 Avant-propos

7.1.1 Quand utiliser un score ?

Pourquoi préférer utiliser une fonction de score aux tests statistiques ?
Comme nous l'avons vu précédemment, le nombre de tests statisques à effectuer croît exponentiellement avec le nombre de nœuds. Les méthodes d'identification d'indépendances conditionnelles ont donc une complexité élevée. Or, lorsque nous voulons utiliser le réseau bayésien pour une tâche particulière, il est très intéressant d'utiliser une fonction de score. Par exemple, pour une tâche de classification, cette fonction de score pourrait dépendre du taux de bonne classification (ou encore de l'aire sous la courbe ROC) sans s'y résumer pour éviter le sur-apprentissage bien sûr. Les méthodes à base de score permettent de trouver des structures de réseaux bayésiens beaucoup plus efficacement.

L'interprétabilité des réseaux obtenus dépend alors de l'interprétabilité de la fonction de score utilisée.

Avant de parler de score, introduisons quelques notions utiles par la suite.

7.1.2 Dimension d'un réseau bayésien

Rappelons que $X = (X_1, \cdots, X_n)$ est un vecteur aléatoire composé de n variables aléatoires discrètes. Si r_i est la modalité de la variable X_i, alors le nombre de paramètres nécessaires pour représenter la distribution de probabilité $\mathbb{P}(X_i/Pa(X_i) = pa(x_i))$ est égal à $r_i - 1$.
Pour représenter $\mathbb{P}(X_i/Pa(X_i))$, il faut alors $Dim(X_i, \mathscr{B})$ paramètres avec

$$Dim(X_i, \mathscr{B}) = (r_i - 1) \cdot q_i \tag{7.1}$$

où q_i est le nombre de configurations possibles pour les parents de X_i, c'est-à-dire

$$q_i = \prod_{X_j \in Pa(X_i)} r_j \tag{7.2}$$

Définition 7.1.1 *La dimension d'un réseau bayésien $\mathscr{B}$ est définie par*

$$Dim(\mathscr{B}) = \sum_{i=1}^{n} Dim(X_i, \mathscr{B}) \tag{7.3}$$

La dimension d'un réseau bayésien est une grandeur qui sera utile pour les fonctions de score qui font intervenir un terme de pénalité. Cette grandeur représente le nombre de paramètres indépendants que possède le réseau.

Il est judicieux de pénaliser les réseaux trop complexes. Ceux-ci peuvent alors encoder de nombreuses distributions de probabilité, cependant, ils sont difficilement interprétables quand ils sont fortement connectés. Dans ce cas, que dire de plus quand toute variable dépend de presque toutes les autres ? Un tel réseau permet donc de modéliser de nombreuses dépendances, par contre, il possède un grand nombre de paramètres. Donc, si la base ne contient pas beaucoup d'exemples, l'apprentissage des paramètres ne sera alors pas très précis : il n'y aura alors que très peu d'exemples pour chaque configuration d'un nœud et de l'ensemble de ses nombreux parents. Il faut alors trouver un bon compromis *expressivité*/*complexité* du réseau. Pour cela, il est possible de se conformer au principe du rasoir d'Occam.

7.1.3 Principe du rasoir d'Occam

Le rasoir de William d'Ockham (Guillaume d'Occam étant la variante latine de ce nom, 1285-1349) est un principe attribué au moine écrivain et théologien franciscain. Penseur du XIVè siècle, il est né à Occam, un village du comté de Surrey. Ses enseignements furent parmi les premiers à être en rupture avec ceux des philosophes médiévaux le précédant, y compris le réalisme aristotélicien de Thomas D'Aquin. William luttait contre la pratique habituelle (toujours d'actualité) de décrire la nature en ayant recours à des abstractions non testables, tentant d'être aussi proche de la réalité physique que possible quelque soit le domaine étudié.
Ce principe a été énoncé comme suit :

"*Pluralitas non est ponenda sine neccesitate*"
"*Frustra fit per plura quod potest fieri per pauciora*"
"*Non sunt multiplicanda entia praeter necessitatem*" [i]

Ce texte peut se traduire en : "*La pluralité ne devrait pas être posée sans nécessité*", "*c'est en vain que l'on fait avec beaucoup ce qui peut être fait avec un petit nombre*", "*les entités ne doivent pas être multipliées sans nécessité*". De tels points de vue ont été partagés par de nombreux auteurs dont Aristote en 350 avant notre ère qui a déclaré : "*la nature prend toujours le chemin le plus court possible*", ou encore : "*le plus limité, s'il est adéquat, est toujours préférable*". De nombreux savants ont adopté ou réinventé le rasoir d'Occam, par exemple, en voici une variante moderne : "*Ce qui peut être fait avec peu d'hypothèses sera fait en vain avec plus*", ou encore, Isaac Newton, qui énonça la règle : "*Nous n'avons pas à accepter plus de causes des choses naturelles que celles qui sont à la fois vraies et suffisantes pour expliquer ces choses*". De même, plus récemment, Einstein disait : "*Tout devrait être fait le plus simplement possible, ce qui ne veut pas dire de façon simplette*". Ce principe est également appelé « principe de simplicité », « principe de parcimonie », ou encore « principe d'économie ». Le principe du rasoir d'Occam consiste à ne pas multiplier les hypothèses au-delà du nécessaire, et en d'autres termes, à ***privilégier l'hypothèse la plus simple*** tant que cela reste compatible avec les observations.

Dans le cadre des réseaux bayésiens, nous privilégierons le modèle le moins complexe, c'est à dire, celui possédant le moins de paramètres.

7.1.4 Score décomposable

En supposant que nous disposions d'une méthode qui parcourt l'espace des DAG en effectuant des opérations du type *ajout* ou *suppression* d'arcs, il est nécessaire de réduire le nombre de calculs utilisés pour l'évaluation du score. Pour cela, posséder un score calculable localement, permet de n'estimer que la variation de ce score entre deux structures voisines, au lieu de le recalculer entièrement pour la nouvelle structure.

Définition 7.1.2 *Un score S est dit* décomposable *(ou* local*) s'il peut être écrit comme une somme ou un produit de mesures dont chacune n'est fonction seulement que d'un nœud et de l'ensemble de ses parents. En clair, si n est le nombre de nœuds du graphe, le score doit avoir une des formes suivantes :*

$$S(\mathscr{B}) = \sum_{i=1}^{n} s(X_i, Pa(X_i)) \qquad ou \qquad S(\mathscr{B}) = \prod_{i=1}^{n} s(X_i, Pa(X_i)) \tag{7.4}$$

Tous les scores que nous allons introduire par la suite peuvent se décomposer localement.

[i] Cette dernière assertion est en fait héritée d'un élève de William d'Occam.

7.1.5 Score équivalent

Nous avons vu dans la section 3.4 que certains réseaux bayésiens différents mais avec le même squelette permettaient de modéliser la même décomposition de la loi jointe de l'ensemble des variables. De tels réseaux sont alors équivalents pour l'expressivité, ainsi que pour la complexité. Le rasoir d'Occam ne permet alors pas d'en un préférer par rapport à l'autre. Il serait alors intéressant d'associer une même valeur du score à toutes les structures équivalentes.

Définition 7.1.3 *Un score qui associe une même valeur à deux graphes équivalents est dit* équivalent.

Certains scores ne sont pas *équivalents* comme par exemple le score BD qui est issu de la vraisemblance marginale $\mathbb{P}(\mathbf{D}|\mathcal{G})$. Par contre, la vraisemblance $\mathbb{P}(\mathbf{D}|\theta,\mathcal{G})$ est score équivalente mais ne respecte pas le *principe de parcimonie*.

7.2 La vraisemblance marginale : la mesure bayésienne

7.2.1 Le score bayésien

La première manière d'associer un score à une structure est de calculer la vraisemblance marginale des données par rapport à cette structure. L'équation 5.4 montre comment exprimer la vraisemblance d'un réseau bayésien à partir de ses paramètres. Cette équation ne peut pas être utilisée ici, car nous n'avons qu'une structure sans paramètre, et il est inimaginable d'intégrer cette formule par rapport à toutes les valeurs des paramètres possibles en pratique.

Pour calculer la probabilité d'une structure conditionnellement à des données, il est possible d'utiliser la remarque suivante :

$$\frac{\mathbb{P}(\mathcal{G}_1|\mathbf{D})}{\mathbb{P}(\mathcal{G}_2|\mathbf{D})} = \frac{\frac{\mathbb{P}(\mathcal{G}_1,\mathbf{D})}{\mathbb{P}(\mathbf{D})}}{\frac{\mathbb{P}(\mathcal{G}_2,\mathbf{D})}{P(\mathbf{D})}} = \frac{\mathbb{P}(\mathcal{G}_1,\mathbf{D})}{\mathbb{P}(\mathcal{G}_2,\mathbf{D})} \tag{7.5}$$

Et, pour calculer $\mathbb{P}(\mathcal{G},\mathbf{D})$, Cooper & Hersovits (1992) ont donné le résultat suivant pour les variables discrètes :

Théorème 7.2.1 *Soit X l'ensemble des variables aléatoires $\{X_1,...,X_n\}$, $n \geqslant 1$, où chaque X_i peut prendre les valeurs $\{x_{i1},...,x_{ir_i}\}$, $r_i \geqslant 1$, $i = 1,...n$. Soient $\mathbf{D}$ la base de données et N le nombre de cas dans $\mathbf{D}$, et soit $\mathcal{G}$ la structure du réseau sur X. De plus, soient pa_{ij} la j^{ieme} instanciation de $Pa(X_i)$, et N_{ijk} le nombre de cas dans $\mathbf{D}$ où X_i a la valeur x_{ik} et que $Pa(X_i)$ est instancié en pa_{ij}. Si $Nij = \sum_{k=1}^{r_i} N_{ijk}$ alors*

$$\mathbb{P}(\mathcal{G},\mathbf{D}) = \mathbb{P}(\mathcal{G})\mathbb{P}(\mathbf{D}|\mathcal{G}) \text{ avec } \mathbb{P}(\mathbf{D}|\mathcal{G}) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\prod_{k=1}^{r_i}N_{ijk}! \tag{7.6}$$

où $\mathbb{P}(\mathcal{G})$ représente la probabilité a priori *affectée à la structure $\mathcal{G}$.*

Grâce à ce résultat, il est possible de calculer la vraisemblance $L(\mathcal{G}|\mathbf{D}) = \mathbb{P}(\mathbf{D}|\mathcal{G})$ de la structure $\mathcal{G}$ vis-à-vis des cas disponibles dans $\mathbf{D}$.

La formule 7.6 peut encore être adaptée en présence d'*a priori* de Dirichlet et donne l'équation 7.7.

$$\mathbb{P}(\mathcal{G},\mathbf{D}) = \mathbb{P}(\mathcal{G})\mathbb{P}(\mathbf{D}|\mathcal{G}) \text{ avec } \mathbb{P}(\mathbf{D}|\mathcal{G}) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{(\alpha_{ij}+r_i-1)!}{(N_{ij}+\alpha_{ij}+r_i-1)!}\prod_{k=1}^{r_i}\frac{(\alpha_{ijk}+N_{ijk})!}{\alpha_{ijk}!} \tag{7.7}$$

Par ailleurs, si nous supposons avoir une distribution *a priori* uniforme sur l'ensemble des graphes, le rapport des deux lois jointes pour chaque structure et les données est le même que le rapport de la distribution des données conditionnellement aux données.

$$\text{si } \mathbb{P}(\mathcal{G}_1) = \mathbb{P}(\mathcal{G}_2) \quad \text{alors} \quad \frac{\mathbb{P}(\mathcal{G}_1|\mathbf{D})}{\mathbb{P}(\mathcal{G}_2|\mathbf{D})} = \frac{\mathbb{P}(\mathcal{G}_1,\mathbf{D})}{\mathbb{P}(\mathcal{G}_2,\mathbf{D})} = \frac{\mathbb{P}(\mathbf{D}|\mathcal{G}_1)}{\mathbb{P}(\mathbf{D}|\mathcal{G}_2)} \tag{7.8}$$

Dans ce cas, la variation de la vraisemblance des structures par rapport aux données est la même que la variation de la masse de la loi jointe des structures et des données. Il est donc possible de faire de l'apprentissage de structure en utilisant le score dit *bayésien*, noté BD, qui n'est autre que la probabilité des données conditionnellement à la structure. Ce score est local, et est défini par l'équation 7.9.

$$BD(\mathcal{G}|\mathbf{D}) = \mathbb{P}(\mathbf{D}|\mathcal{G}) = \prod_{i=1}^{n} bd_i(X_i, Pa(X_i)|\mathbf{D}) \tag{7.9}$$

où le score local bd_i vaut

$$bd(X_i, Pa(X_i)|\mathbf{D}) = \prod_{j=1}^{q_i} \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{7.10}$$

7.2.2 Les autres scores bayésiens

Le score BDe

Le score BD est simple à utiliser, mais il possède le désavantage de ne pas être un *score équivalent*. Néanmoins, Heckerman, Geiger & Chickering (1994) ont adapté ce score pour en obtenir de nouveaux, très proches, et qui possèdent cette propriété.

Supposons que nous possédions deux structures équivalentes, $\mathcal{G}_1$ et $\mathcal{G}_2$, alors la vraisemblance de ces deux structures est la même : $\mathbb{P}(\mathbf{D}|\mathcal{G}_1) = \mathbb{P}(\mathbf{D}|\mathcal{G}_2)$. Si, par ailleurs, nous supposons que les probabilités *a priori* de celles-ci sont strictement positives et égales alors $\mathbb{P}(\mathbf{D},\mathcal{G}_1) = \mathbb{P}(\mathbf{D},\mathcal{G}_2)$ (voir équation 7.5). Dans ce cas, admettons que l'hypothèse suivante soit vérifiée.

> **Hypothèse d'équivalence de vraisemblance** : Si $\mathcal{G}_1$ et $\mathcal{G}_2$ sont deux structures équivalentes de probabilités *a priori* non négatives alors $\mathbb{P}(\Theta|\mathcal{G}_1) = \mathbb{P}(\Theta|\mathcal{G}_2)$.

Cette hypothèse supplémentaire nécessite l'introduction de contraintes supplémentaires sur les exposants de Dirichlet α_{ijk} telles que

$$\alpha_{ijk} = N' \times \mathbb{P}(X_i = k, Pa(X_i) = j|\mathcal{G}_c) \tag{7.11}$$

où $\mathcal{G}_c$ est le graphe complètement connecté et où N' est un nombre de pseudo-exemples supplémentaires défini par l'utilisateur. La spécialisation du score BD qui s'en suit est appelée le score BDe et est définie par l'équation 7.12. Dans ce cas les valeurs des α_{ijk} ne sont plus nécessairement entières et il faut donc remplacer la fonction factorielle de l'équation 7.10 par la fonction gamma (Γ).

$$\mathbb{P}(\mathcal{G},\mathbf{D}) = \mathbb{P}(\mathcal{G})\mathbb{P}(\mathbf{D}|\mathcal{G}) \text{ avec } \mathbb{P}(\mathbf{D}|\mathcal{G}) = \prod_{i=1}^{n}\prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma((\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})} \tag{7.12}$$

Heckerman, Geiger & Chickering (1994) montrent que le score BDe est un *équivalent*.

Le score *BDeu*

Sur ce principe, il est possible de proposer de nombreuses adaptations du score *BD*. Buntine (1991) en a proposé une qui utilise l'*a priori* sur les paramètres suivant.

$$\alpha_{ijk} = \frac{N'}{r_i q_i} \tag{7.13}$$

L'adaptation du score utilisant ces *a priori* est nommée le score *BDeu*.

Le score bayésien généralisé $BD\gamma$

L'introduction d'un hyperparamètre γ permet de généraliser le score bayésien.

$$BD\gamma(\mathcal{G}, \mathbf{D}) = \mathbb{P}(\mathcal{G})\mathbb{P}(\mathbf{D}|\mathcal{G}) = \prod_{i=1}^{n}\prod_{j=1}^{q_i} \frac{\Gamma(\gamma N_{ij} + \alpha_{ij})}{\Gamma((\gamma+1)N_{ij} + \alpha_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma((\gamma+1)N_{ijk} + \alpha_{ijk})}{\Gamma(\gamma N_{ijk} + \alpha_{ijk})} \tag{7.14}$$

La fonction de score de l'équation 7.14 a été introduite par Borgelt & Kruse (2002). Elle permet de passer du score bayésien avec $\gamma = 1$ à l'entropie conditionnelle quand γ tend vers $+\infty$.

7.3 Les principaux scores pondérés

Dans cette section, nous allons introduire les mesures *AIC*, *BIC* et *MDL* qui sont l'adaptation par un critère de score aux réseaux bayésiens de principes proches du rasoir d'Occam et basés sur la notion d'information.

7.3.1 Approximation du score bayésien par Laplace

Les développements suivants peuvent être trouvés en détail dans Chickering (1996). Soit $\Phi_{ijk} = \log(\frac{\theta_{ijk}}{\theta_{ij1}})$, les paramètres naturels. Ces paramètres permettent d'obtenir de meilleures approximations lors d'utilisation d'approximation car ils sont plus stables numériquement. Par ailleurs, ils sont indépendants. Ceci simplifie les expressions en dérivées partielles, et en particulier, les approximations au sens des développements limités de Taylor (MacKay (1998)).

Approchons $\mathbb{P}(\Phi|\mathbf{D}, \mathcal{G}) \propto \mathbb{P}(\mathbf{D}|\mathcal{G}, \Phi)\mathbb{P}(\Phi|\mathcal{G})$ à l'aide de lois gaussiennes multi-variées. Appelons $LP(\Phi) = \ln(\mathbb{P}(\mathbf{D}|\mathcal{G}, \Phi)\mathbb{P}(\Phi|\mathcal{G}))$ et $\widehat{\Phi^{MAP}}$ la configuration qui maximise LP (le maximum *a posteriori* (MAP) de $\mathbb{P}(\Phi|\mathbf{D}, \mathcal{G})$).

Un développement de Taylor au second ordre donne

$$\begin{aligned} LP(\Phi) = \quad & LP(\widehat{\Phi^{MAP}}) + {}^t(\Phi - \widehat{\Phi^{MAP}})LP'(\widehat{\Phi^{MAP}}) \\ & + \tfrac{1}{2}{}^t(\Phi - \widehat{\Phi^{MAP}})A(\Phi - \widehat{\Phi^{MAP}}) + \mathcal{O}(||\Phi - \widehat{\Phi^{MAP}}||^2) \end{aligned} \tag{7.15}$$

avec

$$LP' = {}^t\Big(\frac{\partial}{\partial\Phi_{1q_1r_1}}(LP), \cdots, \frac{\partial}{\partial\Phi_{nq_nr_n}}(LP)\Big)$$

où $LP'(\widehat{\Phi^{MAP}}) = 0$ et où le Hessien au point $\widehat{\Phi^{MAP}}$ est

$$A = \Big(\Big(\frac{\partial^2}{\partial\Phi_{ijk}\partial\Phi_{i'j'k'}}(-LP)\Big)\Big)_{\substack{1 \leqslant i,i' \leqslant n \\ 1 \leqslant j,j' \leqslant q_i \\ 1 \leqslant k,k' \leqslant r_i}}$$

En appliquant cela à notre problème, nous obtenons

$$\mathbb{P}(\mathbf{D}|\mathscr{G}) = \int \mathbb{P}(\mathbf{D}|\mathscr{G},\Phi)\mathbb{P}(\Phi|\mathscr{G})d\Phi = \int \exp(LP(\Phi))d\Phi$$

$$\mathbb{P}(\mathbf{D}|\mathscr{G}) \simeq \int \exp(LP(\widehat{\Phi^{MAP}})\exp(\frac{1}{2}{}^t(\Phi - \widehat{\Phi^{MAP}})A(\Phi - \widehat{\Phi^{MAP}}))d\Phi$$

La méthode d'approximation de Laplace donne

$$\int \exp(\frac{1}{2}{}^t(\Phi - \widehat{\Phi^{MAP}})A(\Phi - \widehat{\Phi^{MAP}}))d\Phi \simeq \sqrt{2\pi^d|A^{-1}|}$$

où d est le nombre de paramètres Φ indépendants (*c-à-d* la dimension du réseau) alors

$$\mathbb{P}(\mathbf{D}|\mathscr{G}) \simeq \mathbb{P}(\mathbf{D}|\mathscr{G},\widehat{\Phi^{MAP}})\mathbb{P}(\widehat{\Phi^{MAP}}|\mathscr{G})\sqrt{2\pi^{Dim(\mathscr{B})}|A^{-1}|}$$

Et l'approximation de Laplace vaut alors

$$\ln(\mathbb{P}(\mathbf{D}|\mathscr{G})) \simeq \ln(\mathbb{P}(\mathbf{D}|\mathscr{G},\widehat{\Phi^{MAP}})) + \ln(\mathbb{P}(\widehat{\Phi^{MAP}}|\mathscr{G})) - \frac{1}{2}\ln(|A|) + \frac{Dim(\mathscr{B})}{2}\ln(2\pi) \quad (7.16)$$

L'équation 7.16 peut alors être utilisée en tant qu'approximation du score bayésien. Remarquons que dans ce score, il reste alors le terme qui dépend du Hessien qui suppose une bonne connaissance de la fonction à évaluer, ce qui n'est pas toujours le cas.

Nous allons à présent voir comment, à partir de cette approximation, en obtenir d'autres qui ne dependent plus de la dérivée seconde de la fonction à évaluer.

7.3.2 Les scores basés sur des critères informatifs

Le score *AIC*

Le critère d'information de Akaike (1973), nommé *AIC* pour *Akaike Information Criterion*, est un critère issu de principe généraux énoncés dans Akaike (1970). Il permet alors de rechercher un compromis entre le sur-apprentissage et le sous-apprentissage.

En effet, lorsque la vraisemblance est utilisée à la fois pour l'apprentissage des paramètres puis pour l'estimation du modèle, cela introduit un biais. Alors que si l'on utilise la vraisemblance pour estimer les paramètres, puis le critère *AIC* pour associer un score au modèle, cette méthode n'est plus biaisée et évite le sous-apprentissage. Par ailleurs, ce critère permet de rechercher le modèle le plus parcimonieux pour représenter les données, il évite donc le sur-apprentissage. Le critère *AIC* donne une estimation de la distance entre le modèle qui s'adapte le mieux aux données et le mécanisme (réel) qui les a produites[ii].

$$AIC(\mathscr{G},\mathbf{D}) = -2\log\mathbb{P}(\mathbf{D}|\mathscr{G},\widehat{\theta^{MV}}) + 2Dim(\mathscr{G}) \quad (7.17)$$

où $\widehat{\theta^{MV}}$ est la configuration des paramètres obtenue par *maximum de vraisemblance* pour la structure $\mathscr{G}$.

Le score *AIC* est décomposable localement et équivalent.

[ii] Le facteur -2 est présent pour des raisons historiques.

Le score BIC

Akaike (1979) a alors repris les principes de Akaike (1970) et a proposé le critère BIC. Ce critère adapté au réseaux bayésiens donne la fonction de score de l'équation 7.18, que nous nommerons, malgré tout, BIC.

$$BIC(\mathscr{G}, \mathbf{D}) = \log \mathbb{P}(\mathbf{D}|\mathscr{G}, \widehat{\theta^{MV}}) - \frac{Dim(\mathscr{G})}{2} \log N \tag{7.18}$$

Le score BIC est décomposable localement et

$$BIC(\mathscr{B}, \mathbf{D}) = \sum_{i=1}^{n} bic(X_i, Pa(X_i), \mathbf{D}) \tag{7.19}$$

où (*cf.* 7.1)

$$bic(X_i, Pa(X_i), \mathbf{D}) = \log \mathbb{P}(\mathbf{D}_{\mathbf{X_i}, \mathbf{Pa}(\mathbf{X_i})}|\mathscr{G}, \widehat{\theta_i^{MV}}) - \frac{Dim(X_i, \mathscr{B})}{2} \log N \tag{7.20}$$

Par ailleurs, le score BIC est équivalent.

Approximation $BIC - MV$ **de l'approximation de Laplace**

Prenons les termes qui croissent avec n dans l'équation 7.16 : par exemple $\ln(|A|)$ croît comme $Dim(\mathscr{B}) \ln(N)$, et pour la taille de la base de cas N assez grande, $\widehat{\phi^{MAP}}$ (le MAP de $\mathbb{P}(\Phi|\mathbf{D}, \mathscr{G})$) peut être approché par $\widehat{\Phi^{MV}} = \arg max_\Phi \mathbb{P}(\mathbf{D}|\mathscr{G}, \Phi)$, le maximum de vraisemblance (MV) de $\mathbb{P}(\mathbf{D}|\mathscr{G}, \Phi)$ (ou le MAP en présence d'*a priori*).

$$\ln \mathbb{P}(\mathbf{D}|\mathscr{G}) \simeq \ln \mathbb{P}(\mathbf{D}|\mathscr{G}, \widehat{\Phi^{MV}}) - \frac{Dim(\mathscr{B})}{2} \ln(N) \tag{7.21}$$

Nous retrouvons alors ici la même approximation que celle obtenue à partir des critères issus des principes BIC.

Autres critères de score

Le critère AIC pénalise les modèles ayant trop de paramètres, néanmoins cette pénalisation n'est pas suffisante et il a été montré expérimentalement que ce critère engendrait le choix de modèles sur-paramétrisés (*cf.* Shibata (1976)). De nombreuses autres adaptations du critère AIC ont été proposées. Par exemple, Schwartz (1978) et Rissanen (1978) ont amélioré le critère AIC et ont introduit un critère qu'ils ont nommé SIC. Hurvich & Tsai (1989) trouvaient que le critère BIC n'etait pas assez précis asymptôtiquement et ont proposé le *Akaike Information Corrected Criterion*.

$$AICC(\mathscr{G}, \mathbf{D}) = \ln \mathbb{P}(\mathbf{D}|\mathscr{G}, \widehat{\theta^{MV}}) - 2Dim(\mathscr{G}) \frac{N}{N - Dim(\mathscr{G}) - 1} \tag{7.22}$$

Ce dernier est une estimation au second ordre des principes de Akaïke. Tant que la base de cas n'est pas assez grande par rapport au nombre de paramètres à évaluer, il est conseillé d'utiliser $AICC$ plutôt que le score AIC classique.

Il est imaginable d'utiliser n'importe quel score, du moment, que son utilisation est justifiée par notre objectif. Il serait également possible d'utiliser le score ICL (pour *Integrated Completed likelyhood*) introduit par Biernacki, Celeux & Govaert (1998). Ce score est essentiellement un score BIC sauf qu'il est de plus pénalisé par l'entropie moyenne estimée.

7.3.3 La longueur de description minimum

Le principe de longueur de description minimale a été introduit par Rissanen (1978). Ce principe dit que pour modéliser le plus fidèlement des données, il faut minimiser la taille de la description qui contient à la fois la longueur de codage du modéle et la longueur de codage des données en utilisant ce modèle. De manière similaire au score AIC, ce principe a été adapté au cadre des réseaux bayésiens par Bouckaert (1993) pour donner le score MDL.

$$MDL(\mathcal{G},\mathbf{D}) = \ln \mathbb{P}(\mathcal{G}) - N \cdot \mathbb{E}\big[\ln\big(P(\mathbf{D}|\mathcal{G})\big)\big] + \frac{Dim(\mathcal{G})}{2}\ln N \tag{7.23}$$

Comme le score BIC, le score MDL est une approximation asymptotique de la vraisemblance marginale respectant le principe de parsimonie. Soient $\widehat{\theta_{ijk}}$, les valeurs obtenues par maximum de vraisemblance comme il est décrit sur les pages 52 et suivantes. La vraisemblance de la structure correspondante est alors connue et son entropie vaut

$$\begin{aligned} H(\mathbf{D}|\mathcal{G}) &= -\mathbb{E}\big[\ln\big(P(\mathbf{D}|\mathcal{G})\big)\big] = -\mathbb{E}\big[\ln\big(\prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i}\widehat{\theta_{ijk}}\big)\big] \\ &= -\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}\mathbb{E}\big[\ln(\widehat{\theta_{ijk}})\big] = -\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}\frac{N_{ijk}}{N}\ln\widehat{\theta_{ijk}} \end{aligned} \tag{7.24}$$

En nous inspirant de la formule 5.6, nous obtenons la formulation de l'entropie en fonction de la vraisemblance du graphe.

$$H(\mathbf{D}|\mathcal{G}) = -\frac{LL(\mathcal{G}|\mathbf{D})}{N} = -\frac{\ln\big(\mathbb{P}(\mathbf{D}|\mathcal{G})\big)}{N} \tag{7.25}$$

$$-MDL(\mathcal{G},\mathbf{D}) = \ln\left(\frac{\mathbb{P}(\mathbf{D}|\mathcal{G})}{\mathbb{P}(\mathcal{G})}\right) - \frac{Dim(\mathcal{G})}{2}\ln N \tag{7.26}$$

Or la longueur de description est à minimiser, en considérant la réécriture du score MDL de l'équation 7.26, nous nous apercevons que l'expression de ce critère $-MDL$ est très proche du score BIC, en particulier avec un *a priori* uniforme sur l'ensemble des structures la principale différence est dans les termes $\mathbb{P}(\mathbf{D}|\mathcal{G})$ et $\mathbb{P}(\mathbf{D}|\mathcal{G},\widehat{\theta^{MV}})$ qui sont asymptotiquement équivalents.

Le score MDL est décomposable localement et équivalent.

Une autre formulation a été donnée par Lam & Bacchus (1993) :

$$MDL_2(\mathcal{G},\mathbf{D}) = \ln \mathbb{P}(\mathbf{D}|\widehat{\Theta},\mathcal{G}) - Na(\mathcal{G})\cdot\ln N - c\cdot Dim(\mathcal{G}) \tag{7.27}$$

où $Na(\mathcal{G})$ est le nombre d'arcs dans le graphe $\mathcal{G}$ et c le nombre de *nats* (ou bits) utilisé pour stocker chaque paramètre numérique.

Plus récemment, Yun & Keong (2004) ont proposé une autre formulation du score MDL pour l'apprentissage de réseaux bayésiens.

$$\begin{aligned} IMDL(\mathcal{G},\mathbf{D}) = \;& \ln(n) + \sum_{i=1}^{n}\ln(r_i) - N\sum_{i=1}^{n}\big(H(X_i|Pa(X_i))\big) \\ & + \frac{1}{2}\ln(N)\Big[\sum_{\theta_{ijk}\neq 1} q_i(r_i-1) + \sum_{\theta_{ijk}=1} q_i\Big] \end{aligned} \tag{7.28}$$

L'avantage de cette formulation est que de nombreux termes ne dépendent pas de la structure $\mathcal{G}$ et n'ont donc pas besoin d'être implémentés en pratique.

7.4 Autres mesures adaptées à la classification

Supposons à présent que nous devions faire face à une problématique de classification. Appelons C, la variable aléatoire représentant la classe d'une observation de la base d'exemples $\mathbf{D}$, où $(\{x_1, \cdots, x_n\}_j, c_j)$ est la j-ème entrée de la base d'exemples. Notons κ le nombre de classes, alors la variable C peut prendre ses valeurs parmi $\{1, \ldots, \kappa\}$.

Pour une tâche de classification, nous voulons minimiser l'erreur en généralisation $\varepsilon(f) = \int\int_{\mathcal{X}\times\{1,\ldots,k\}} \text{Coût}(x,c)\mathbb{P}(x,c)\,\mathrm{d}x\mathrm{d}c$, où $\text{Coût}(x,c)$ est le coût de prédire la classe c pour l'entrée x (typiquement 0 s'il s'agit de la bonne classe, et 1 sinon) et/ou le risque $\mathcal{R}(f) = \mathbb{P}(f(x) \neq c)$, où f est la fonction que réalise notre classifieur.

Lors d'un apprentissage classique, nous allons chercher à approcher la distribution jointe $\mathbb{P}(X, C)$ entre les variables d'observation $X = (X_1, \cdots, X_n)$ et l'attribut représentant la classe C.

Le classifieur idéal étant $f^*(x) = arg\max_{1\leqslant c\leqslant\kappa}(\mathbb{P}(C = c|X = x))$ en supposant qu'une entrée n'appartienne qu'à une seule classe. Si nous nous concentrons sur une tâche de classification, nous devons alors simplement approcher au mieux la distribution $\mathbb{P}(C|X)$. Pour ce faire, il peut être plus judicieux d'utiliser la vraisemblance classifiante plutôt que la vraisemblance. Celle-ci permet alors de construire des modèles discriminants (qui séparent au mieux les classes) au détriment de modèles génératifs (qui modélisent au mieux la loi jointe).

Comme nous l'avons vu précédemment la vraisemblance s'écrit (pour tout DAG $\mathcal{G}$)

$$L(\mathcal{G}|\mathbf{D}) = \mathbb{P}(\mathbf{D}|\mathcal{G}) = \prod_{j=1}^{N} \mathbb{P}(\{x_1, \cdots, x_n\}_j, c_j|\mathcal{G})$$

Or lors d'une tâche de classification nous allons évaluer la probabilité de chaque classe c : $\mathbb{P}(c|x_1, \cdots, x_n)$. Il faut donc que cette probabilité ait été bien apprise. Pour des considérations de stabilité numérique, nous pourrions donc être amenés à maximiser la vraisemblance de la classe par rapport aux données d'observation

$$\mathbb{P}(x_1, \cdots, x_n|C = c, \mathcal{G}) \propto \frac{\mathbb{P}(C = c|x_1, \cdots, x_n, \mathcal{G})}{\sum_{x_1,\cdots,x_n} \ldots \sum \mathbb{P}(x_1, \cdots, x_n, C = c|\mathcal{G})} \tag{7.29}$$

La *vraisemblance classifiante* (où *vraisemblance conditionnelle*) définie dans Scott & Symons (1971) et Celeux & Govaert (1992) est donnée par

$$L_c(\mathcal{G}|\mathbf{D}) = \prod_{j=1}^{N} \mathbb{P}\big(\{x_1, \cdots, x_n\}_j|C = c_j, \mathcal{G}\big) \tag{7.30}$$

Calculer cette valeur revient à partitionner la base d'exemples en fonction des valeurs de la classe, et faire le produit des différentes vraisemblances à valeur de la classe fixées.

Contrairement au cas de la vraisemblance marginale, la solution des paramètres $\widehat{\theta}_{ijk}$ maximisant la vraisemblance conditionnelle n'a pas de forme fermée. D'un point de vue pratique, maximiser cette vraisemblance conditionnelle nécessite donc plus de calculs que de maximiser la vraisemblance marginale puisqu'il faut avoir recours à des techniques d'optimisation (de type descente de gradient par exemple)

Par ailleurs, la vraisemblance classifiante n'est pas décomposable localement à cause du terme au dénominateur de l'équation 7.29. D'un point de vue pratique, il n'est donc pas possible d'utiliser un système de cache pour les scores locaux. Cette méthode est donc bien plus lourde à utiliser.

Greiner, X., Bin & Wei (2005) proposent de maximiser une autre log-vraisemblance conditionnelle LCL pour effectuer un apprentissage discriminant.

$$LCL(\mathscr{G}|\mathbf{D}) = \frac{1}{N}\sum_{j=1}^{N} \ln\left(\mathbb{P}(c_j|\{x_1,\cdots,x_n\}_j,\mathscr{G})\right) \tag{7.31}$$

Ils utilisent alors une méthode de type descente de gradient qui est très coûteuse lorsque le nombre d'attributs augmente et qui est donc difficilement conjugable avec un apprentissage de structure. Grossman & Domingos (2004) proposent alors une méthode mixte consistant en l'utilisation de cette vraisemblance conditionnelle pour sélectionner les structures tandis que les paramètres sont estimés par maximum de vraisemblance.

Remarquons par ailleurs que la log-vraisemblance conditionnelle LCL et la log-vraisemblance classifiante $LL_c = \log(L_c)$ sont reliées par l'équation 7.32 qui est déduite de l'équation 7.29.

$$LL_c(\mathscr{G}|\mathbf{D}) = N \cdot LCL(\mathscr{G}|\mathbf{D}) + \sum_{j=1}^{N} \ln\left(\frac{\sum_{c=1}^{\kappa}\mathbb{P}(\{x_1,\cdots,x_n\}_j, C=c|\mathscr{G})}{\sum_{x_1,\cdots,x_n}\ldots\sum\mathbb{P}(x_1,\cdots,x_n,C=c_j|\mathscr{G})}\right) \tag{7.32}$$

8

Méthodes de recherche de structure à base de score

"In the 1960s and 1970s, students frequently asked : Which kind of representation is best ? I usually replied that we'd need more research...But now I would reply : To solve really hard problems, we'll have to use several different representations. This is because each particular kind of data structure has its own virtues and deficiencies, and none by itself would seem adequate for all the different functions involved with what we call common sense."

Marvin Lee Minsky, né en 1927.

Sommaire

8.1 Les heuristiques sur l'espace de recherche 88
8.1.1 Arbre de poids maximal 88
8.1.2 Structure de Bayes Naïve augmentée 88
8.1.2.1 Structure de Bayes naïve 88
8.1.2.2 Structure de Bayes naïve augmentée 88
8.1.2.3 Structure de Bayes naïve augmentée par un arbre 89
8.1.2.4 Classifieur naïf augmenté par une forêt : FAN . . . 89
8.1.3 Algorithme K2 89
8.1.3.1 Méthode générale 89
8.1.3.2 Deux propositions d'ordonnancement : K2+T et K2-T 90
8.2 Les heuristiques sur la méthode de recherche 91
8.2.1 Principe de la recherche gloutonne dans l'espace des DAG . 91
8.2.2 Différentes initialisations 92
8.2.2.1 Initialisations classiques 92
8.2.2.2 Notre initialisation : GS+T 92
8.2.3 Méthode GES 93
8.2.4 Les méthodes incrémentales 95
8.2.5 Les méthodes mixtes 95
8.3 Les méthodes non-déterministes 95
8.3.1 Utilisation de MCMC 95
8.3.2 Utilisation d'algorithmes évolutionnaires 95
8.3.3 Autres heuristiques d'optimisation 96
8.4 Les méthodes mixtes 96
8.5 L'apprentissage de réseaux bayésiens hybrides 96

8.1 Les heuristiques sur l'espace de recherche

La recherche exhaustive n'étant pas une méthode réalisable en pratique lorsque le nombre de nœuds est élevé (voir page 14), il faut utiliser des heuristiques de recherche. Les premières méthodes présentées ici recherchent le meilleur graphe dans un espace de recherche plus petit, tandis que les algorithmes suivants vont effectuer un parcours heuristique de l'espace des solutions (DAG ou CPDAG).

8.1.1 Arbre de poids maximal

Une première heuristique peut être adaptée des travaux de Chow & Liu (1968). Ces derniers ont proposé une méthode dérivée de la recherche de l'arbre de recouvrement de poids maximal (*maximal weight spanning tree* ou MWST). Cette méthode s'applique aussi à la recherche de structure d'un réseau bayésien en fixant un poids à chaque arête potentielle A–B de l'arbre. Ce poids peut être par exemple l'*information mutuelle* entre les variables A et B comme proposé par Chow & Liu (1968), ou encore la variation du score local lorsqu'on choisit B comme parent de A comme proposé par Heckerman, Geiger & Chickering (1994). Une fois cette matrice de poids définie, il suffit d'utiliser un des algorithmes standards de résolution du problème de l'arbre de poids maximal comme l'algorithme de Kruskal ou celui de Prim. L'arbre non dirigé retourné par cet algorithme doit ensuite être dirigé en choisissant une racine puis en parcourant l'arbre par une recherche en profondeur. La racine peut être choisie soit aléatoirement, soit à l'aide de connaissance *a priori*, soit en prenant la variable représentant la classe pour des problèmes de classification.

8.1.2 Structure de Bayes Naïve augmentée

8.1.2.1 Structure de Bayes naïve

Pour une problématique de classification, la *structure de Bayes naïve* a prouvé expérimentalement qu'elle était capable de donner de bons résultats. L'hypothèse de base de ce modèle est de supposer que toutes les observations soient indépendantes les unes des autres conditionnellement à la variable classe, ce qui revient à une simplification de la loi jointe de l'équation 2.6.

Même si le réseau bayésien naïf suppose des hypothèses très contraignantes, Domingos & Pazzani (1997) ont montré que celui-ci est optimal pour modéliser des relations conceptuelles conjonctives et disjonctives.

8.1.2.2 Structure de Bayes naïve augmentée

L'hypothèse simplificatrice utilisée dans le classifieur de Bayes naïf est largement non vérifiée en pratique. Il existe différentes techniques pour assouplir cette hypothèse. Elles consistent toutes en l'ajout de dépendances conditionnelles entre les observations. Nous obtenons alors une structure dite de *Bayes naïve augmentée*.

Domingos & Pazzani (1997) ont montré que le classifieur de Bayes naïf ne nécessite pas l'hypothèse d'indépendance des attributs pour être optimal pour une fonction de coût en 0-1 et donc que les classifieurs de Bayes naïfs augmentés sont également optimaux dans ce cas.

Sahami (1999) propose ainsi l'algorithme KDB (pour *k-limited Dependence Bayesian classifiers*) pour apprendre des classifieurs de Bayes limités dans le fait qu'un nombre

maximum de parents par nœud doit être fixé. Cette méthode est basée sur des calculs d'information mutuelle.

8.1.2.3 Structure de Bayes naïve augmentée par un arbre

Il est possible de construire une sous-structure optimale sur les observations en adaptant la méthode de recherche de l'arbre de poids maximal. Pour cela, la fonction de score doit être modifiée de manière à obtenir une fonction de score *conditionnellement* à la variable classe. Puis en reliant le nœud représentant la classe à tous les sommets de cette structure, nous obtenons *la structure de Bayes naïve augmentée par un arbre* (TANB pour *Tree Augmented Naive Bayes*, Geiger (1992) et Friedman, Geiger & Goldszmidt (1997)). La méthodologie sera developpée dans un contexte plus général en section 12.2.1.

Cette technique permet d'améliorer les résultats obtenus avec une structure de Bayes naïve classique, mais reste exclusivement réservée aux problématiques de classification.

8.1.2.4 Classifieur naïf augmenté par une forêt : FAN

Forêt optimale

En ajoutant un critère d'arrêt prématuré en fonction d'une valeur minimum d'augmentation du score dans les algorithmes de recherche de l'arbre couvrant de poids maximal, il est possible de construire une forêt (voir définition C.1.8). Ce critère peut par exemple être un pourcentage du score courant jugé significatif, ou encore simplement de choisir seulement des valeurs positives car le score *BIC* d'une arête peut être négatif si l'augmentation de la vraisemblance est plus faible que le terme de pénalité. Le graphe ne sera pas connexe et aura donc une structure de forêt.

Classifieur naïf augmenté par une forêt : FAN

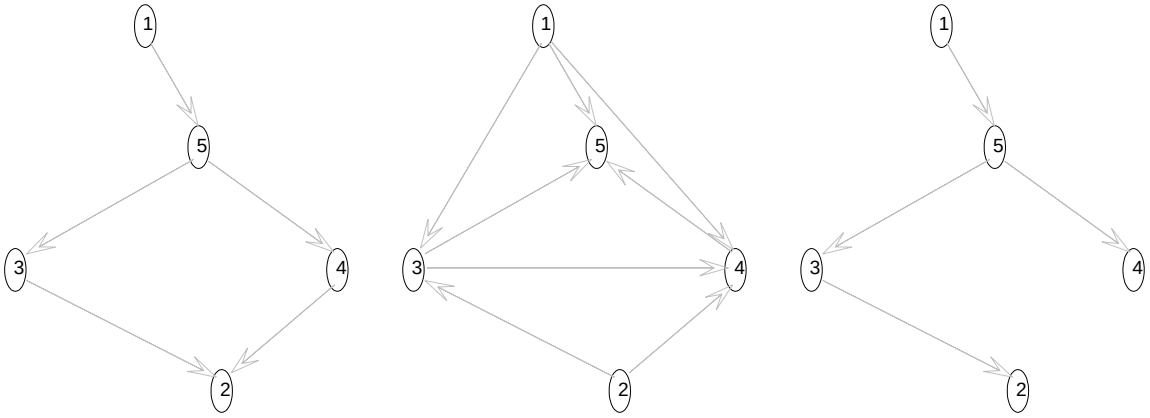
De même, en utilisant l'algorithme pour trouver une forêt optimale sur l'ensemble des observations et en adaptant la matrice de poids comme dans la section 8.1.2.3, nous obtiendrons l'algorithme FAN qui sélectionne le classifieur de Bayes naïf augmenté par une forêt optimale selon le score et le critère qui auront été choisis.

8.1.3 Algorithme K2

8.1.3.1 Méthode générale

Une autre méthode efficace a été proposée par Cooper & Hersovits (1992). Cette méthode nommée K2 est à présent largement utilisée bien qu'elle possède l'inconvénient de demander un ordre d'énumération en paramètre d'entrée. L'idée de l'algorithme K2 est de maximiser la probabilité de la structure sachant les données dans l'espace de DAG respectant cet ordre d'énumération. Cette technique utilise originellement le score bayésien de l'équation 7.9.

L'algorithme K2 teste alors l'ajout de parents de la manière suivante : le premier nœud ne peut pas posséder de parents et pour les nœuds suivants, l'ensemble des parents choisi est le sous-ensemble de nœuds qui augmente le plus le score parmi l'ensemble des nœuds le précédant dans l'ordre d'énumération. L'espace de recherche est ainsi réduit à l'ensemble des DAG respectant cet ordre.



(a) Graphe original et obtenu avec l'algorithme K2 initialisé avec un ordre topologique issu du résultat de MWST.

(b) Graphe obtenu par K2 avec l'ordre 1, 2, 3, 4, 5

(c) Graphe obtenu par MWST

FIG. 8.1 : *K2 : apport de l'initialisation par MWST : illustration sur un exemple jouet.*

Heckerman, Geiger & Chickering (1994) ont montré que le score bayésien n'était pas *score équivalent* et en ont proposé une variante, BDe, qui corrige cela en utilisant un *a priori* spécifique sur les paramètres du réseau bayésien. Il est aussi possible d'utiliser le score BIC (*score équivalent*) ou n'importe quel autre *score décomposable* au lieu du score bayésien. Bouckaert (1993) a également proposé une variante à l'algorithme K2 qui utilise le score MDL.

L'hypothèse *a priori* de l'ordre d'énumération est très forte et rechercher le meilleur ordre pour les variables est un problème NP-difficile. Singh & Valtorta (1993) proposent une méthode générique pour trouver un ordre d'énumération à partir de tests d'indépendance conditionnelle avant d'utiliser l'algorithme K2. Comme pour l'algorithme PC, les tests doivent être faits avec des ensembles de conditionnement de taille croissante, et en pratique, nous sommes alors souvent amenés à limiter la taille de ces ensembles pour limiter le nombre de tests. Friedman & Koller (2000) utilisent des MCMC pour échantillonner l'espace des énumérations possibles des variables, puis cherchent la meilleure structure correspondant à cet ordre grâce à l'algorithme K2. De manière similaire, Larrañaga, Kuijpers, Murga & Yurramendi (1996) utilisent un algorithme génétique pour trouver un ordre d'énumération et de Campos & Huete (1999) utilisent une méthode de recuit simulé et des algorithmes évolutionnaires.

8.1.3.2 Deux propositions d'ordonnancement : K2+T et K2-T

Nous proposons d'exploiter l'arbre retourné par l'algorithme MWST pour générer un ordre d'énumération des variables. Cela nous donne une heuristique d'initialisation à moindre coût puisque MWST possède l'avantage d'être très rapide. Pour des tâches de classification, il est possible d'utiliser le nœud classe comme racine de l'arbre obtenu par MWST, puis de prendre l'ordre d'énumération des nœuds issu de l'arbre trouvé par MWST pour obtenir un ordonnancement des nœuds qui servira à l'algorithme K2. Nous appellerons "K2+T" l'algorithme K2 utilisant cet ordonnancement des nœuds.

Cette initialisation, qui permet de diminuer les problèmes lors de l'apprentissage de structure liés au phénomène évoqué en page 37, est visible sur la figure 8.1. Nous

avons utilisé le graphe de la figure 8.1(a) pour générer 5000 exemples, puis nous avons effectué un apprentissage de structure avec l'algorithme K2 pour obtenir le graphe de la figure 8.1(b). Nous nous apercevons que nous obtenons une structure très différente de la structure originale, mais qui possède tout de même la propriété d'être une carte d'indépendances minimale du problème. Si, par ailleurs, nous effectuons un apprentissage de structure avec la méthode de recherche de l'arbre de poids maximal, nous obtenons la structure illustrée sur la figure 8.1(c). En utilisant un ordre topologique de ce graphe (par exemple $1, 5, 3, 2, 4$ ou $1, 5, 4, 3, 2$) pour initialiser l'algorithme K2, le graphe original (8.1(a)) est alors retrouvé.

Par ailleurs, il peut être avantageux d'interpréter le nœud classe comme une *conséquence* plutôt que comme une *cause*. Dans ce cas, nous regarderons aussi ce qui se passe lorsque l'ordonnancement est l'*inverse* de celui qui est proposé par MWST. Nous appellerons "K2-T" l'algorithme K2 utilisant cet ordonnancement *inverse* des nœuds. Ces différentes propositions d'initialisation ainsi que leur évaluation expérimentale ont fait l'objet de publications : François & Leray (2004a) et François & Leray (2004b).

Récemment, Teyssier & Koller (2005) ont proposé un algorithme pour évaluer l'ordre optimal d'énumération des variables. La technique utilisée est alors une méthode gloutonne sur l'espace des ordres. Cette première phase est de complexité élevée. Une fois qu'elle est effectuée, il est ensuite possible d'utiliser une technique du type K2 pour évaluer un réseau bayésien.

8.2 Les heuristiques sur la méthode de recherche

8.2.1 Principe de la recherche gloutonne dans l'espace des DAG

Une méthode plus générale, puisqu'elle recherche dans l'espace *complet* des DAG, consiste en l'utilisation d'une recherche gloutonne (ou GS comme *greedy search*). Elle prend un graphe de départ, définit un voisinage de ce graphe, puis associe un score à chaque graphe du voisinage. Le meilleur graphe est alors choisi comme point de départ de l'itération suivante.

L'intérêt de cette méthode réside dans la définition du voisinage à utiliser. Une recherche gloutonne demande de nombreuses itérations avant de converger. Plus le voisinage sera grand, plus la méthode se rapproche d'une méthode exhaustive, moins il y aura d'itérations mais plus elle sera longue car il faudra calculer de nombreux scores. Plus le voisinage sera petit et plus la méthode a de chance de converger vers un optimum local rapidement. En utilisant un voisinage contenant toutes les structures se différenciant du graphe de l'itération courante par l'ajout, le retrait ou l'inversion d'un arc nous pouvons utiliser la décomposabilité d'une mesure de score à notre avantage. En effet dans ce cas, nous n'aurons plus besoin de recalculer le score entièrement pour chaque graphe du voisinage, mais il sera seulement nécessaire de mettre à jour le score en fonction de l'opération effectuée comme il est indiqué dans la table 8.1 où s représente le score local.

Cette méthode étant itérative, il faut alors choisir une initialisation.

Friedman, Nachman & Peér (1999) ont alors proposé une méthode d'apprentissage de structure gloutonne lorsque le nombre de nœuds est très grand. Ils proposent alors

Opérateur	insertion $X_i \rightarrow X_j$	suppression $X_i \rightarrow X_j$	inversion $X_i \rightarrow X_j$
Variation du score	$+\ s(X_j, Pa(X_j) \cup \{X_i\})$ $-\ s(X_j, Pa(X_j))$	$+\ s(X_j, Pa(X_j) \setminus \{X_i\})$ $-\ s(X_j, Pa(X_j))$	$+\ s(X_j, Pa(X_j) \setminus \{X_i\})$ $-\ s(X_j, Pa(X_j))$ $+\ s(X_i, Pa(X_i) \cup \{X_j\})$ $-\ s(X_i, Pa(X_i))$

TAB. 8.1 : *Variation du calcul du score dans l'espace des* DAG *selon l'opérateur.*

de restreindre l'espace de recherche pour éviter de prendre en considération des réseaux candidats qui ne sont pas raisonnables. L'ensemble restreint des candidats pour l'itération suivante est alors construit de manière à ce que seules les variables étant fortement dépendantes peuvent être voisines les unes des autres.

Moore & Wong (2003) ont proposé un opérateur supplémentaire pour effectuer une recherche gloutonne. Cet opérateur consiste en le choix d'un nœud cible, le retrait des arcs adjacents à ce nœud, puis la réinsertion de nouveaux arcs pour ce nœud. Moore & Wong (2003) ont alors montré que l'utilisation de cet opérateur permettait de réduire considérablement les temps de calculs d'une recherche gloutonne tout en donnant de meilleurs résultats finaux.

8.2.2 Différentes initialisations

La recherche gloutonne est sensible à l'initialisation. Habituellement, il est recommandé d'effectuer plusieurs recherches gloutonnes à partir de différents points de départ pour éviter de retenir le résultat pour lequel la recherche gloutonne s'est arrêtée sur un *mauvais* optimum local.

8.2.2.1 Initialisations classiques

Nous noterons GS+0, une recherche gloutonne initialisée par un graphe vide (complètement déconnecté). De plus, nous noterons GS+C, une recherche gloutonne initialisée par une chaîne aléatoire. Cette initialisation par une chaîne est une recommandation de Friedman (1997) pour la méthode SEM qui sera décrite dans la section 12.1.1, qui est une méthode gloutonne qui fonctionne à partir de bases de données incomplètes.

8.2.2.2 Notre initialisation : GS+T

Sur le même principe que pour les méthodes K2+T et K2-T, nous proposons également de recourir à l'arbre obtenu par MWST comme point de départ de l'algorithme GS, ce qui donne la méthode que nous appellerons "GS+T".

Cette proposition d'initialisation ainsi que son évaluation expérimentale ont fait l'objet de publications : François & Leray (2004a) et François & Leray (2004b).

La construction de la structure arborescente repose sur la liaison des variables les plus corrélées entre elles. Nous choisissons alors de lier les variables ayant l'information mutuelle (ou la variation du score *BIC*) la plus élevée. Comme le graphe final aura une structure arborescente, toutes les variables seront deux à deux marginalement dépendantes (sans conditionnement). Néanmoins, les dépendances conditionnelles, représentées par l'intermédiaire de V-structures, ne seront alors pas présentes dans ce graphe.

Mais, comme les variables les plus corrélées seront rapprochées les unes des autres, une première partie du travail que doit effectuer un algorithme glouton sera effectuée. Les itérations successives de la méthode gloutonne vont ensuite ajouter d'autres dépendances, en particulier des dépendances conditionnelles, via l'ajout de nouveaux arcs et via l'inversion d'arcs pour créer des V-structures.

8.2.3 Adaptation de la recherche gloutonne à l'espace des représentants des classes d'équivalence de Markov

Principe

Des travaux récents (Chickering (2002a), Castelo & Kocka (2002) et Auvray & Wehenkel (2002)) montrent qu'il peut être plus profitable de travailler dans l'espace des CPDAG (représentant des classes d'équivalence de Markov) plutôt que dans l'espace des DAG. En effet cet espace possède moins de plateaux pour la fonction de score, car de nombreux DAG avec des scores égaux sont représentés par un seul CPDAG. Il est donc plus aisé d'avoir une propriété de convergence dans cet espace, malgré sa taille, qui reste super-exponentielle (voir page 40).

Une méthode de recherche gloutonne dans l'espace des équivalents de Markov nommée GES pour *Greedy Equivalent Search* a été introduite par Meek (1997). Elle consiste en deux étapes itératives. La première étape construit un graphe pas à pas en parcourant l'espace des équivalents de Markov. la seconde étape consiste en le retrait des arcs superflus éventuels.

L'optimalité de cet algorithme a été prouvée et dépend de la *conjecture de Meek*.

Première phase

Lors de la première phase de cet algorithme, nous allons commencer avec un graphe vide. Avec l'utilisation d'un score et la définition d'un voisinage du graphe courant dans l'espace de CPDAG, nous allons choisir le voisin qui augmente le plus le score pour l'itération suivante. Bien évidemment le score utilisé doit être *score équivalent*. Le voisinage est défini comme suit. Tous les graphes dirigés sans circuit qui différent d'une instanciation du CPDAG représentant la classe d'équivalence courante par l'ajout ou le retournement d'un arc sont considérés. Puis, en prenant le quotient de l'ensemble d'arrivée par la relation d'équivalence de Markov et en retirant la classe courante si elle est présente, nous obtenons un voisinage. Une illustration est donnée sur la figure 8.2.

Cette technique gloutonne permet de passer d'une carte de dépendances (*D-map*) à une carte de dépendances plus grande (c'est-à-dire qui représente strictement plus de dépendances). Ce processus continue jusqu'à ce qu'il n'y ait plus de cartes de dépendances qui augmentent le score.

Deuxième phase

Cette phase commence par construire le voisinage inférieur de ce graphe. Pour cela, il faut construire toutes les instanciations du CPDAG courant. Tous les graphes qui différent d'une instanciation par le retrait d'un arc sont considérés. Puis comme dans la phase précédente, le quotient de l'ensemble d'arrivée par la relation d'équivalence de Markov est considéré. En pratique, cela revient donc à prendre l'ensemble des CPDAG construits à partir de cet ensemble de DAG. Nous choisissons ensuite le graphe de ce voisinage qui augmente le plus le score, tant qu'il en existe.

Cette phase permet de passer d'une carte d'indépendances pour le problème à une autre carte d'indépendances plus petite (c'est-à-dire qui représente strictement moins d'indépendances conditionnelles). Une fois les deux phases traversées, nous obtenons donc une carte d'indépendances minimale pour le problème.

Conjecture de Meek

L'optimalité de cet algorithme a été prouvé en partie par Meek (1997), modulo la conjecture de Meek qui s'énonce de la manière suivante :

Théorème 8.2.1 *Soient $\mathcal{G}$ et $\mathcal{H}$, deux* DAG *tels que $\mathcal{H}$ est une carte d'indépendances de $\mathcal{G}$, c'est-à-dire que toute indépendance représentée par la structure de $\mathcal{H}$ est également représentée par la structure de $\mathcal{G}$. Alors, il existe une suite finie d'additions et de retournements d'arcs qui, appliquée à $\mathcal{G}$, possède les propriétés suivantes*

1) après chaque changement, $\mathcal{G}$ reste un DAG *et $\mathcal{H}$ est toujours une carte d'indépendances de $\mathcal{G}$,*

2) après tous les changements, $\mathcal{G} = \mathcal{H}$.

Kočka, Bouckaert & Studený (2001) ont en partie démontré cette conjecture, qui a finalement été entièrement prouvée par Chickering (2002b). Donc, lorsque nous sommes en présence d'une carte d'indépendance pour un problème, il est toujours possible d'obtenir une carte d'indépendances minimale pour ce problème en effectuant des retraits et des inversions d'arcs. Ce résultat permet donc, lorsque le problème est supposé représentable par un graphe dirigé sans cycles, d'obtenir de manière sûre une carte d'indépendances pour un problème à partir d'une *D-map*.

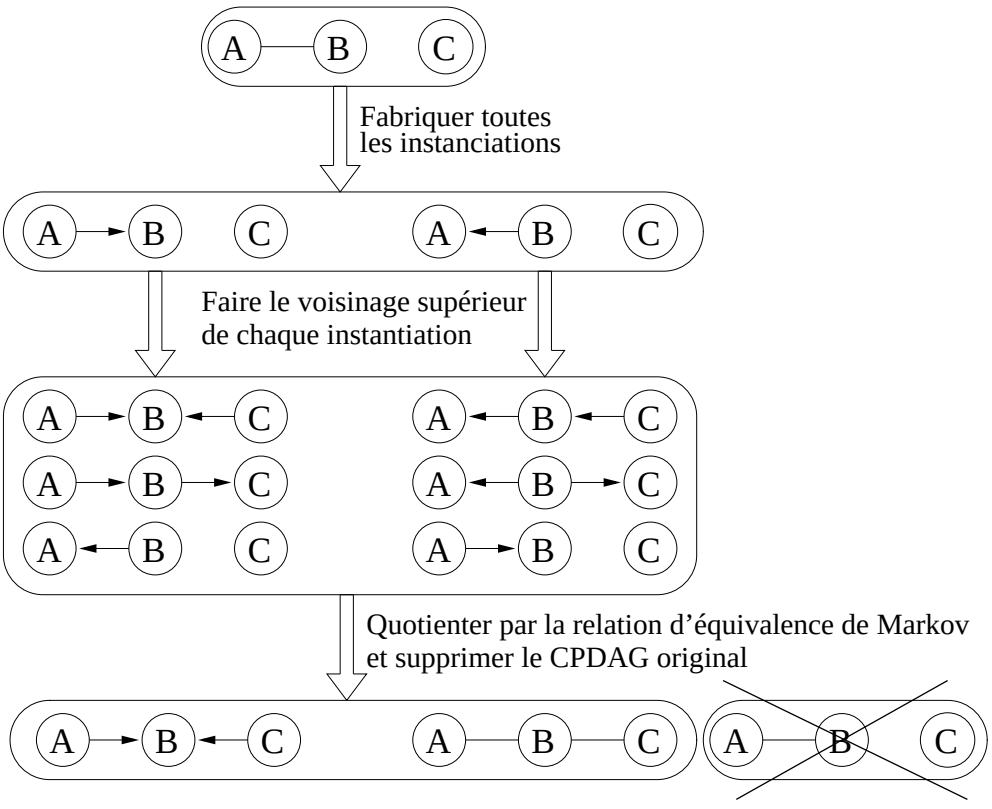


FIG. 8.2 : *Voisinage supérieur au sens des équivalents de Markov d'un graphe simple pour la première phase de l'algorithme GES*

Contrairement aux méthodes gloutonnes classiques, il n'est pas justifié d'utiliser différents points de départ pour cette méthode étant donné que Chickering (2002b) et Meek (1997) ont prouvé l'optimalité de cette méthode lorsqu'elle est initialisée à l'aide d'une structure vide.

8.2.4 Les méthodes incrémentales

En se basant sur des techniques d'apprentissage déterministes, Roure i Alcobé (2004) a proposé une technique pour effectuer un apprentissage de réseaux bayésiens incrémentalement à partir des méthodes MWST, K2 ou d'une recherche gloutonne.

La plupart des méthodes d'apprentissage incrémentales supposent que les données sont stationnaires. Plus récemment, Nielsen & Nielsen (2006) ont proposé un algorithme permettant d'apprendre une structure de manière incrémentale à partir de données non-stationnaires et de mettre à jour cette structure pour l'adapter à la distribution *dynamique* du flux de données.

8.2.5 Les méthodes mixtes

Il est bien sûr imaginable de tirer profit de chacune de ces méthodes, voire même de parcourir différents espaces de recherche au cours de l'apprentissage. Gonzales & Jouve (2006) proposent par exemple une méthode originale en trois phases. La première phase est proche d'un algorithme K2, donc dans l'espace des graphes dirigés respectant un ordre d'énumération. La seconde phase est proche d'une recherche gloutonne dans l'espace des graphes non-orientés. Et la troisième phase, est alors proche de la phase de descente de l'algorithme GES, donc dans l'espace des équivalents de Markov.

8.3 Les méthodes non-déterministes

8.3.1 Utilisation de MCMC

Il est possible d'utiliser une implémentation des chaînes de Markov cachées (Markov Chain Monte Carlo (MCMC)), par exemple l'algorithme de Metropolis-Hastings pour rechercher dans l'espace de tous les DAG (Murphy (2001)). L'idée de base de cette méthode est alors de créer des exemples à partir de $\mathbb{P}(\mathbf{D}|\mathscr{G})$ puis, durant chaque itération un nouveau graphe $\mathscr{G}'$ est choisi si une variable aléatoire uniforme prend une valeur plus grande que le facteur de Bayes $\frac{\mathbb{P}(\mathbf{D}|\mathscr{G}')}{\mathbb{P}(\mathbf{D}|\mathscr{G})}$ (ou d'une version pondérée de ce facteur) où $\mathbf{D}$ est la base d'origine augmentée des nouveaux exemples.

Citons également Friedman & Koller (2000) qui utilisent des MCMC pour échantillonner l'espace des énumérations possibles des variables, puis cherchent la meilleure structure correspondant à cet ordre grâce à l'algorithme K2.

8.3.2 Utilisation d'algorithmes évolutionnaires

Pour les problèmes où l'espace de recherche est grand, il est toujours possible d'avoir recours à une heuristique évolutionnaire. Récemment, Wong, Lee & Leung (2004) et Cotta & Muruzabál (2004) ont utilisé ce type de méthodes pour effectuer de l'apprentissage de structure. Muruzábal & Cotta (2004) ont ensuite étendu la méthode de Cotta & Muruzabál (2004) à l'espace des équivalents de Markov.

Il est également possible de diviser l'apprentissage en deux phases, une première phase consistant à trouver un ordre d'énumération des nœuds compatible avec le problème, puis une deuxième phase dans laquelle cet ordre est utilisé pour créer le graphe dirigé. Citons alors Larrañaga, Poza, Yurramendi, Murga & C. Kuijpers (1996) qui utilisent un algorithme évolutionnaire pour effectuer la première phase, puis une technique similaire à l'algorithme K2 pour la deuxième phase.

Delaplace, Brouard & Cardot (2006) ont également récemment proposé une méthode pour apprendre la structure d'un réseau bayésien en utilisant un algorithme génétique.

8.3.3 Autres heuristiques d'optimisation

Bon nombre de méthodes non déterministes peuvent être adaptées pour l'apprentissage de structure. Par exemple, de Campos, Fernández-Luna, Gámez & Puerta (2002) utilisent un algorithme d'optimisation de type 'colonie de fourmis' pour effectuer l'apprentissage de structure de réseaux bayésiens. Ils utilisent l'heuristique d'optimisation par colonie de fourmis pour effectuer une recherche dans l'espace des ordres sur les variables puis donnent un graphe à l'aide de l'algorithme K2, et utilisent cette heuristique également directement dans l'espace des DAG.

8.4 Les méthodes mixtes

Il existe donc deux grands types d'approches pour effectuer de la recherche de structure : à partir de tests statistiques, ou en utilisant une fonction de score. Néanmoins, nous pouvons imaginer des méthodes qui tirent parti des avantages respectifs de ces deux types d'approches.

Par exemple, l'algorithme BENEDICT proposé par Acid & de Campos (2001) est une méthode gloutonne dans l'espace des DAG respectant un ordre d'énumération. L'hybridation provient de la construction de la fonction de score qui est la somme des entropies de Kullback-Leiber de toutes les indépendances conditionnelles (avec ensemble de conditionnement minimal) codées par le graphe et listées selon l'ordre d'énumération.

Dash & Druzdzel (1999) utilisent également une méthode de recherche de structure mixte qu'ils nomment EGS pour *essential graph search*. Cette méthode est basée sur l'algorithme PC, mais elle fait varier l'ordre d'énumération des nœuds ainsi que la puissance du test (*significance level*) de manière itérative et aléatoire jusqu'à ce qu'elle n'arrive plus à augmenter le score (*bayésien*) de la solution pendant n itérations. Cette méthode rend alors un CPDAG.

8.5 L'apprentissage de réseaux bayésiens hybrides

Les méthodes présentées ci-dessus sont principalement dédiées à l'apprentissage de réseaux bayésiens pour lesquels tous les nœuds sont supposés discrets. Davies & Moore (2000a) ont proposé une méthode pour effectuer l'apprentissage de réseaux bayésiens hybrides contenant des nœuds discrets et des nœuds dont les distributions de probabilités sont supposées continues et être de la forme d'un mélange de gaussiennes.

Plus récemment, Cobb & Shenoy (2006) ont introduit une technique pour apprendre des structures de réseaux bayésiens avec à la fois des nœuds discrets et des nœuds continus dont les distributions de probabilités sont supposées être de la forme de mélanges d'exponentielles tronquées.

9

Expérimentations

"Il est dans la probabilité que mille choses arrivent qui sont contraires à la probabilité."

Henry Louis Mencken (1880 - 1956)

Sommaire

9.1 Recherche d'une bonne structure 98
9.1.1 Algorithmes utilisés 98
9.1.2 Réseaux tests et techniques d'évaluation 98
Réseaux de tests 98
Score *BIC* 99
Divergence de Kullback-Leibler 99
Distance d'édition 99
Temps de calcul 100
9.1.3 Résultats et interprétations 101
9.1.3.1 Influence de la taille de la base 101
9.1.3.2 Distance à la distribution de base 104
9.1.3.3 Stabilité vis-à-vis de la base d'exemples 104
9.1.3.4 Reconnaissance des dépendances faibles 104
9.1.3.5 Choix du score 105
9.1.3.6 Choix de l'initialisation 106
9.1.3.7 Duel inter-méthodes 108
9.2 Recherche d'un bon classifieur 108
9.2.1 Algorithmes utilisés et techniques d'évaluation 108
9.2.2 Résultats et interprétations 110

9.1 Recherche d'une bonne structure

9.1.1 Algorithmes utilisés

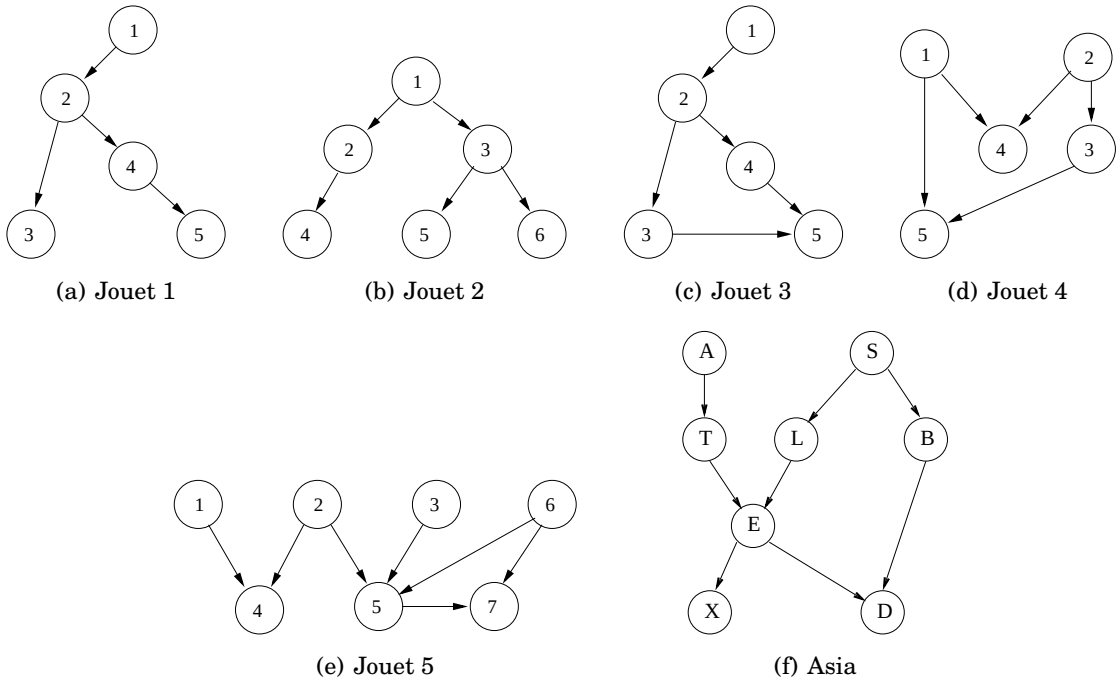
Nous avons utilisé Matlab, et plus précisément la Bayes Net Toolbox (Murphy (2004)) qui fournit déjà certaines méthodes présentées ci-dessous. Le site Leray, Guilmineau, Noizet, François, Feasson & Minoc (2003) en propose une introduction et des tutoriels en français. Le code des fonctions mises en œuvre dans ces expérimentations est mis à disposition par l'intermédiare du *Structure Learning Package* décrit dans Leray & François (2004b).

Nous avons testé les algorithmes suivants :

- PC, utilisant des tests du χ^2 (section 6.3),
- BNPC, utilisant l'information mutuelle conditionnelle (section 6.5),
- MWST avec l'information mutuelle (MWST-im) et le score *BIC* (MWST-bic) (section 8.1.1),
- K2 initialisé aléatoirement avec le score *BD* (meilleur résultat sur 5 lancements) (section 8.1.3),
- K2+T avec le score *BD* (racine de l'arbre aléatoire) (section 8.1.3.2),
- K2-T avec le score *BD* (racine de l'arbre aléatoire) (section 8.1.3.2),
- GS-0 initialisé avec la structure vide, scores *BD* et *BIC* (section 8.2.1),
- GS-C initialisé avec une chaîne, scores *BD* et *BIC* (section 8.2.2.1),
- GS+T initialisé par un arbre, score *BD* (*init.* MWST-im) et score *BIC* (MWST-bic) (section 8.2.2.2),
- GES avec les scores *BD* et *BIC* (section 8.2.3).

9.1.2 Réseaux tests et techniques d'évaluation

Réseaux de tests



(a) Jouet 1 (b) Jouet 2 (c) Jouet 3 (d) Jouet 4

(e) Jouet 5 (f) Asia

FIG. 9.1 : *Les réseaux de tests.*

Pour cette première série d'expériences, nous allons utiliser des réseaux bayésiens connus qui sont illustrés sur la figure 9.1. Ceux-ci vont alors être utilisés pour générer des bases d'exemples de différentes tailles par échantillonnage (voir section 4.2.1).

Nous utilisons les bases d'exemples ainsi créées pour tester différentes méthodes d'apprentissage de structure. Pour une taille fixée, nous générons 20 bases d'exemples sur lesquelles nous effectuons chaque algorithme d'apprentissage. Les réseaux obtenus sont alors comparés au réseau original à l'aide de différentes mesures d'évaluation.

Score *BIC*

La première mesure est simplement le score *BIC* moyen des 20 réseaux obtenus sur une base d'exemples de tests (qui est toujours la même pour un problème et qui ne sert qu'à cela). En plus de ce score *BIC*, l'écart-type des différentes mesures est mentionné. Le score *BIC* est négatif, et plus il est proche de zéro, plus il est possible de considérer que la méthode permet d'obtenir des réseaux bayésiens modélisant bien les données. Par ailleurs, plus la valeur de l'écart-type des scores est faible, plus il est possible de considérer la méthode comme stable par rapport à de légères variations de la base d'apprentissage. Sur certaines mesures, il se peut que les résultats aient un meilleur score que le réseau original qui a servi à générer les données (cela n'arrive que très rarement sur les moyennes). Pour ces cas, nous pouvons par exemple considérer qu'un léger surapprentissage a eu lieu ou encore que le modèle obtenu arrive à coder une loi très similaire à la loi sous-jacente au réseau original, et cela avec moins de paramètres. Néanmoins ces cas restent exceptionnels.

Divergence de Kullback-Leibler

La deuxième mesure est la divergence de Kullback-Leiber (moyenne) entre les lois codées par les réseaux obtenus par rapport à la loi codée par le réseau original qui est introduite en section 6.2.

Pour calculer le gain d'information entre deux réseaux bayésiens $\mathscr{B}^1 = (\mathscr{G}^1, \Theta^1)$ et $\mathscr{B}^2 = (\mathscr{G}^2, \Theta^2)$, il suffit d'adapter la formule de l'équation 6.7.

$$KL(\mathscr{B}^1 || \mathscr{B}^2) = \sum_{i=1}^{n} \sum_{j^1=1}^{q_i^1} \sum_{k=1}^{r_i} \theta_{ijk}^1 \log \left(\frac{\theta_{ijk}^1}{\mathbb{P}(X_i = k | Pa^1(X_i) = j^1, \mathscr{G}^2, \Theta^2)} \right) \tag{9.1}$$

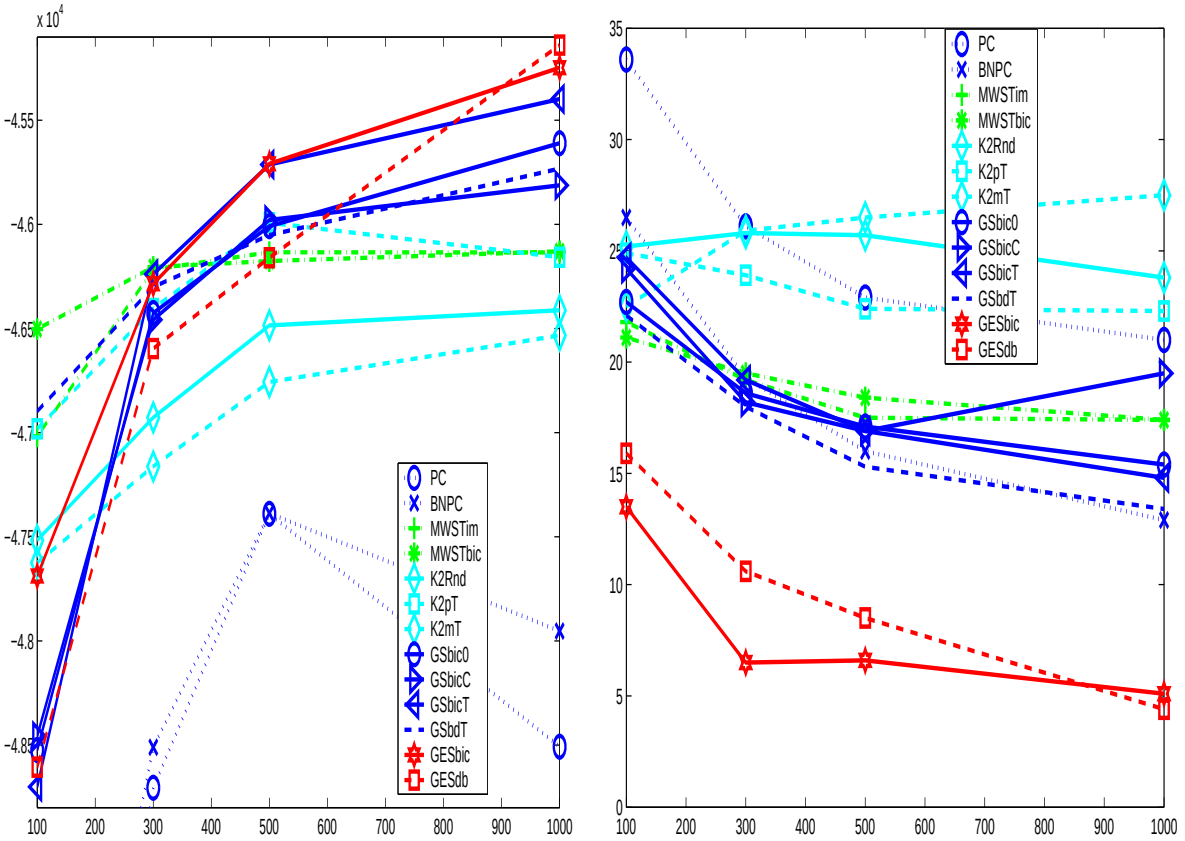
où $Pa^1(X_i)$ représente l'ensemble des parents de la variable X_i dans la structure $\mathscr{G}^1$.

Si les structures $\mathscr{G}^1$ et $\mathscr{G}^2$ sont identiques, il suffit de remplacer le terme du dénominateur par $\theta_{ij^1k}^2$, sinon, il faut effectuer une inférence dans le réseau $\mathscr{B}^2$.

Cette information n'est pas redondante avec le score *BIC*, puisque dans ce dernier la complexité du réseau est prise en compte et masque la précision du résultat obtenu. Par exemple, pour deux réseaux ayant deux scores *BIC* égaux, si l'un possède une valeur de la distance de Kullback-Leiber plus faible, cela signifie qu'il approche mieux la loi jointe sous-jacente au réseau original mais qu'il est plus complexe. Dans ce cas, le réseau possède des paramètres (donc des arcs) supplémentaires et *il se peut* qu'il soit alors moins précis du point de vue de l'identification des relations d'indépendance conditionnelle. Ceci peut alors être corroboré par la distance d'édition.

Distance d'édition

La troisième mesure est la distance d'édition (moyenne) des graphes obtenus. Cette mesure est peu pertinente avec la notion d'équivalence de Markov (voir section 3.4), il



(a) Evolution de la somme des scores *BIC* moyens sur l'apprentissage des réseaux 9.1(b), 9.1(c), 9.1(d), 9.1(e), 9.1(f)..

(b) Evolution de la somme des distances d'édition moyennes pour les mêmes problèmes.

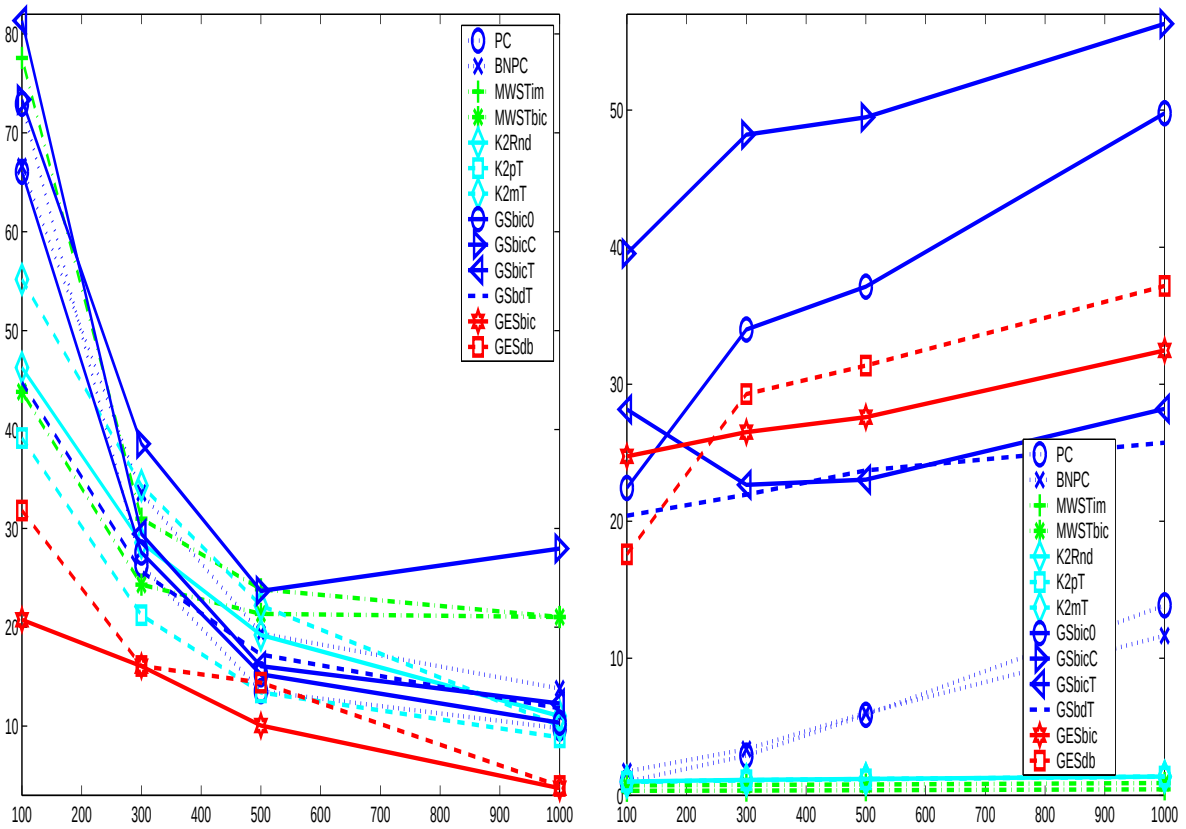
FIG. 9.2 : *Quelques courbes d'évolution de score BIC et de la distance d'édition en fonction de la taille de la base d'exemples, les résultats présentés sont des moyennes sur 20 lancements.*

est aisé d'obtenir deux réseaux équivalents pour lesquels la distance d'édition est tout de même éloignée. Mais en considérant cette valeur de concert avec le score *BIC* d'une part, ou la distance de Kullback-Leiber d'autre part, il est possible de se rendre compte si les différentes méthodes permettent d'obtenir des réseaux bayésiens où soit une distribution représentant les données correctement est retrouvée, soit une structure où les bonnes relations d'indépendance conditionnelles sont retrouvées, soit les deux.

Insistons sur le fait que la distance d'édition (resp. le score *BIC*) n'est pas un critère suffisant pour comparer plusieurs structures. En effet, le renversement d'un arc réversible n'a pas d'incidence sur le score alors qu'il modifie la distance. De même, si une V-structure n'est pas correctement détectée, il peut y avoir création d'une clique qui est alors à la distance d'édition de 2 ou 3 de la V-structure mais qui possède tout de même un bon score local.

Temps de calcul

La dernière mesure rapportée est le temps de calcul moyen des différentes méthodes. Celui-ci permet d'avoir une comparaison empirique des complexités des implémentations des différents algorithmes, mais n'est néanmoins présenté qu'à titre indicatif.



(a) Evolution de la somme des divergences de Kullback-Leiber moyennes sur l'apprentissage des réseaux 9.1(b), 9.1(c), 9.1(d), 9.1(e), 9.1(f).

(b) Evolution de la somme des temps de calculs moyens pour les mêmes problèmes.

FIG. 9.3 : *Quelques courbes d'évolution de la divergence de Kullback-Leiber moyenne.*

9.1.3 Résultats et interprétations

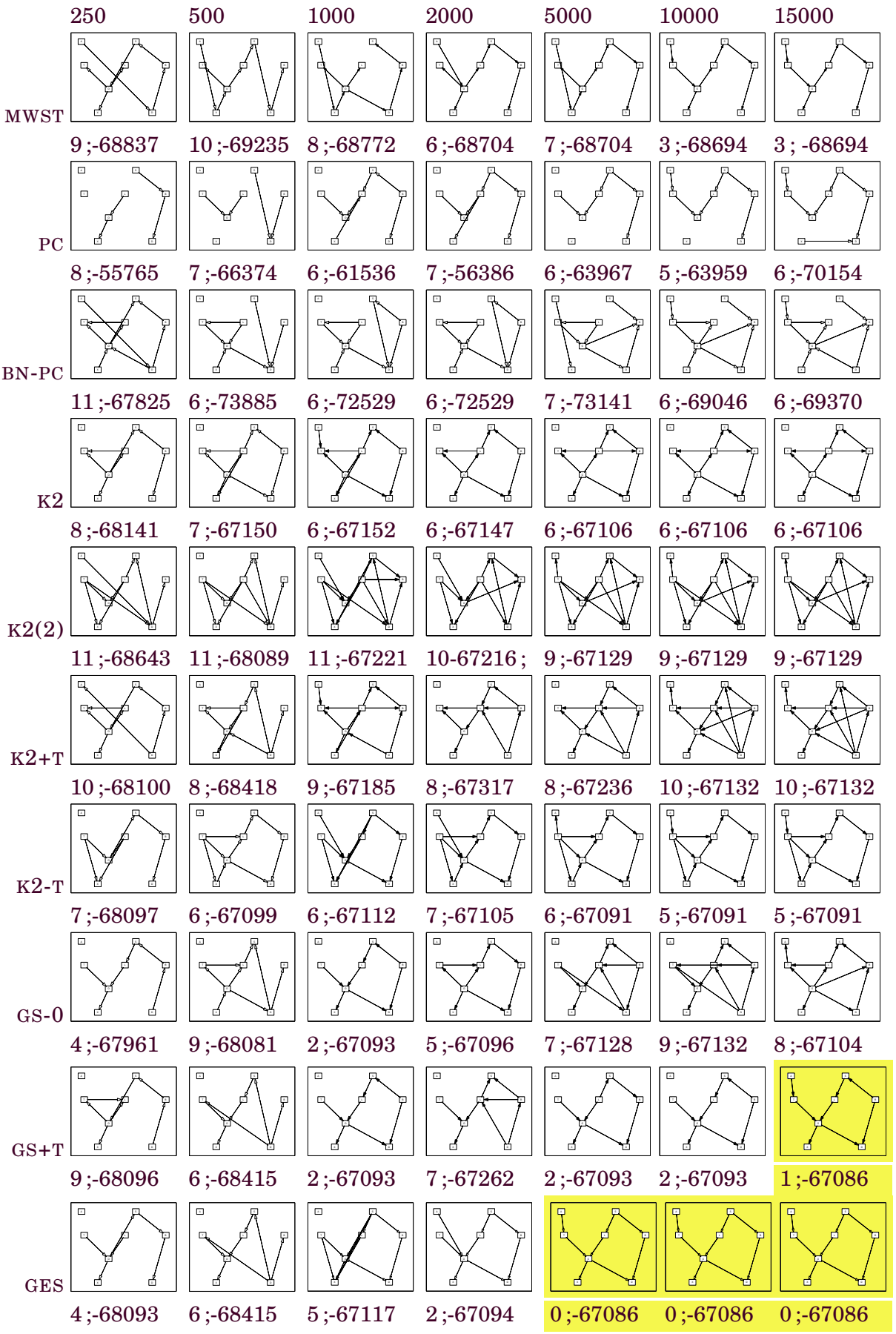
Les résultats sont donnés en détails dans l'annexe F.1 et toutes les figures qui sont présentées dans cette section n'en sont qu'un résumé.

9.1.3.1 Influence de la taille de la base

L'augmentation du score est faible dès que nous possédons plus de 500 comme il est possible de le voir sur la figure 9.2(a) (et a fortiori entre 1000 et 2000 points, voir annexe F.1). De plus, la diminution des distances d'édition n'est plus vraiment visible au-delà de 300 cas en apprentissage sur la figure 9.2(b). Les méthodes d'apprentissage de structure testées semblent très stables lorsque la taille de la base d'exemples varie.

Nous pouvons cependant remarquer que la méthode MWST (courbes vertes) est celle qui donne les meilleurs résultats lorsque le nombre d'exemples est très faible (100 points ici). De plus, c'est également celle qui donne les résultats les plus stables. En cela, elle se laisse surpasser par les autres méthodes lorsque la base d'exemples devient plus grande. Ces dernières peuvent alors profiter de leur espace de recherche plus riche et du volume d'exemples disponibles pour découvrir des relations plus subtiles entre les variables.

Comme nous pouvons le voir sur les tables 9.1 et 9.4 où les résultats de la méthode K2 sont donnés avec deux ordonnancements différents ("ELBXASDT" et "TALDSXEB" pour ASIA). Ces résultats, très visuels pour ASIA, permettent alors de se rendre compte de la



TAB. 9.1 : ASIA *(figure 9.1(f)) : réseaux, distances d'édition et scores BIC obtenus pour différentes méthodes (lignes) et différentes tailles de base d'exemples (colonnes). Les graphes correspondant à la structure d'origine sont surlignés en jaune. Les scores BIC ont été calculés à partir d'une base de 20000 exemples de tests.*

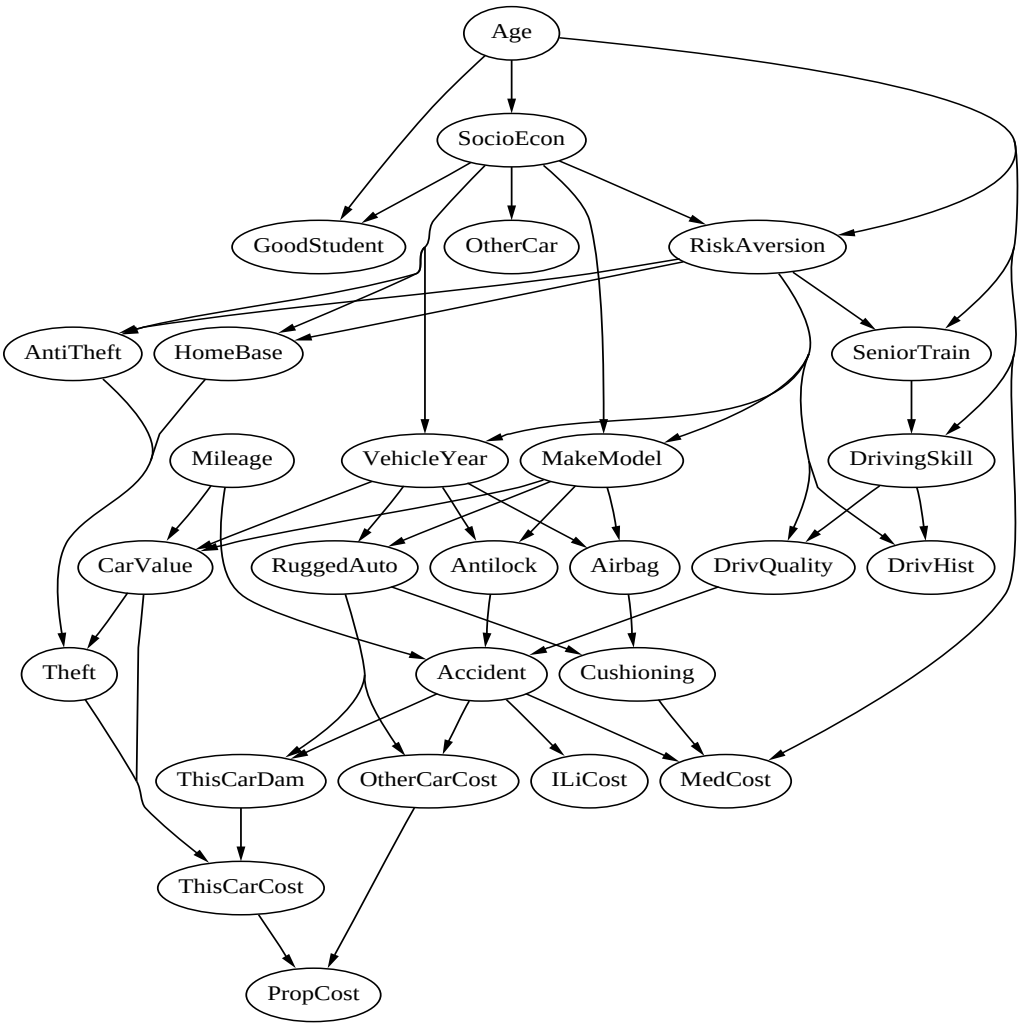


FIG. 9.4 : *Structure du réseau bayésien* INSURANCE *utilisé pour les expérimentations reportées sur la table 9.2.*

INSURANCE	250	500	1000	2000	5000	10000	15000
MWST	**37** ;-3373	**34** ;-3369	36 ;-3371	35 ;-3369	34 ;-3369	34 ;-3369	34 ;-3369
K2	56,-3258	62 ;-3143	60 ;-3079	64 ;-3095	78 ;-3092	82 ;-3080	85 ;-3085
K2(2)	**26** ;-3113	**22** ;-2887	**20** ;-2841	**21** ;-2873	**21** ;-2916	**18** ;-2904	**22** ;-2910
K2+T	42 ;-3207	40 ;-3009	42 ;-3089	44 ;-2980	47 ;-2987	51 ;-2986	54 ;-2996
K2-T	55 ;-3298	57 ;-3075	57 ;-3066	65 ;-3007	70 ;-2975	72 ;-2968	73 ;-2967
MCMC*	50 ;-3188	44 ;-2967	46 ;-2929	40 ;-2882	50 ;-2905	51 ;-2898	54 ;2892
GS	**37** ;-3228	39 ;-3108	**30** ;-2944	33 ;-2888	**29** ;-2859	25 ;-2837	28 ;-2825
GS+T	43 ;-3255	**35** ;-3074	**28** ;-2960	**26** ;-2906	33 ;-2878	**19** ;-2828	**21** ;-2820
GES	43 ;-2910	41 ;-2891	39 ;-2955	41 ;-2898	38 ;-2761	38 ;-2761	38 ;-2752

TAB. 9.2 : *Distance d'édition au graphe d'origine et score* BIC *(divisés par 100 et arrondis) pour le réseau* INSURANCE *de la figure 9.4 et pour plusieurs méthodes d'apprentissage de structure (en lignes) et pour plusieurs tailles de bases d'exemples (en colonnes). Les scores BIC ont été calculés à partir d'une base de 30000 exemples de tests.*
* *Comme la méthode d'apprentissage de structure utilisant la technique de Chaînes de Monte-Carlo (*MCMC*) n'est pas déterministe, les résultats donnés sont des moyennes sur cinq exécutions.*

variabilité des graphes obtenus inter-méthodes ou intra-méthodes. Pour le réseau INSURANCE, nous nous apercevons que la méthode MWST donne des résultats similaires quelle que soit la taille de la base d'exemples. La méthode qui donne le plus de variations est la méthode GS qui voit le score *BIC* augmenter de 9% lorsque nous passons d'une base de 500 exemples à une base de 10000 exemples et la méthode GS+T pour laquelle le score *BIC* augmente de 13% lorsque la base passe de 250 exemples à 15000 exemples.

9.1.3.2 Distance à la distribution de base

Observons à présent la représentation de la loi plutôt que le graphe lui-même. Les différents graphes obtenus permettent, après un apprentissage de structure, de trouver des lois de probabilité proches des lois originales. Plus la base d'exemples est grande et plus l'apprentissage des paramètres est précis (voir figure 9.3(a)). Les résultats obtenus par la méthode MWST sont, soit, très bons lorsque le graphe original est proche d'un arbre, soit, nettement moins bons que ceux obtenus par d'autres méthodes lorsque le graphe original contient de nombreuses V-structures. Les tables de probabilités conditionnelles obtenues avec cette méthode sont peu complexes, donc simples à apprendre, mais ne peuvent cependant pas représenter des lois conditionnelles complexes.

Comme les divergences de Kullback-Leiber diminuent alors que les distances d'édition restent stables (figure 9.2(b)), nous pouvons très bien imaginer que plus d'information concernant la loi est codée dans les tables de probabilités conditionnelles, plutôt que dans la structure du graphe.

9.1.3.3 Stabilité vis-à-vis de la base d'exemples

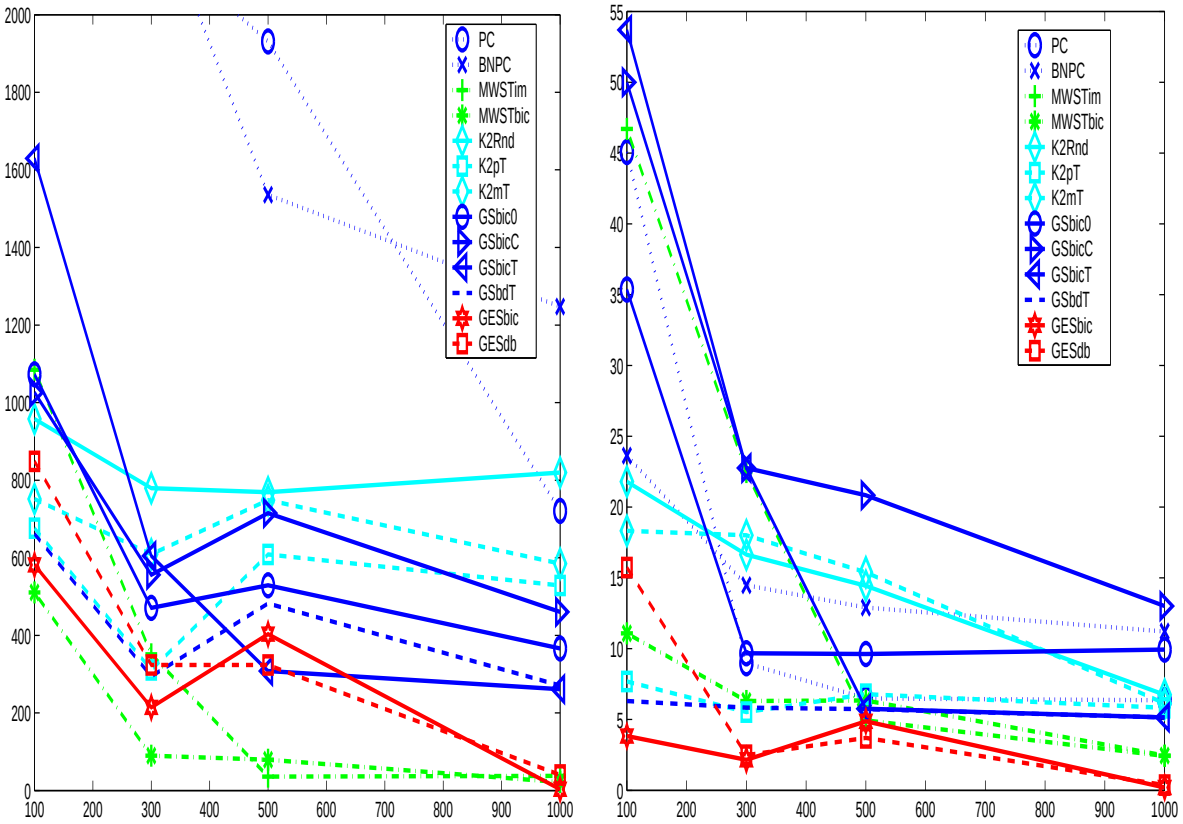
Comme nous l'avons vu précédemment, les méthodes sont plutôt stables en moyenne lorsque la taille de la base d'exemples augmente. Cela n'est alors pas toujours le cas à taille de bases d'exemples fixée.

Contrairement à ce qu'il est possible de penser, lorsque la taille de la base d'exemples augmente, les variances des résultats pour les scores *BIC* reportées en figure 9.5(a) ne dimimuent pas. Ceci peut être dû au fait que la base d'exemples était trop petite et que les variations restaient importantes. Néanmoins, ces résultats sont souvent très bons lorsque la base atteint les tailles de 1000 ou 2000 exemples, cette variation peut être dûe au fait que l'apprentissage de la structure est très dépendant des premiers arcs trouvés, Or ces premiers arcs sont alors très variables pour une même méthode.

En particulier, la variance des résultats pour les divergences de Kullback-Leiber reportées en figure 9.5(b) ont tendance à diminuer, mais la variance des méthodes basées sur l'algorithme K2 reste élevée quelle que soit la taille de la base d'exemples. Comme nous pouvons le voir sur les tables 9.1 et 9.4 où les résultats de la méthode K2 sont donnés avec deux ordonnancements différents ("ELBXASDT" et "TALDSXEB" pour ASIA). Ces résultats, très visuels pour ASIA, permettent alors de se rendre compte de la variabilité des graphes obtenus inter-méthodes ou intra-méthodes.

9.1.3.4 Reconnaissance des dépendances faibles

Les méthodes d'identification de structure ne sont pas toujours capables de retrouver des relations de dépendances de "poids faible". Par exemple, dans les réseaux bayésiens



(a) Evolution de la somme des variances de score *BIC* sur l'apprentissage des réseaux 9.1(b), 9.1(c), 9.1(d), 9.1(e), 9.1(f).

(b) Evolution de la somme des variances de la divergence KL sur l'apprentissage des réseaux 9.1(b), 9.1(c), 9.1(d), 9.1(e), 9.1(f).

FIG. 9.5 : *Variances des résultats de score BIC et de la divergence de KL sur 20 apprentissages.*

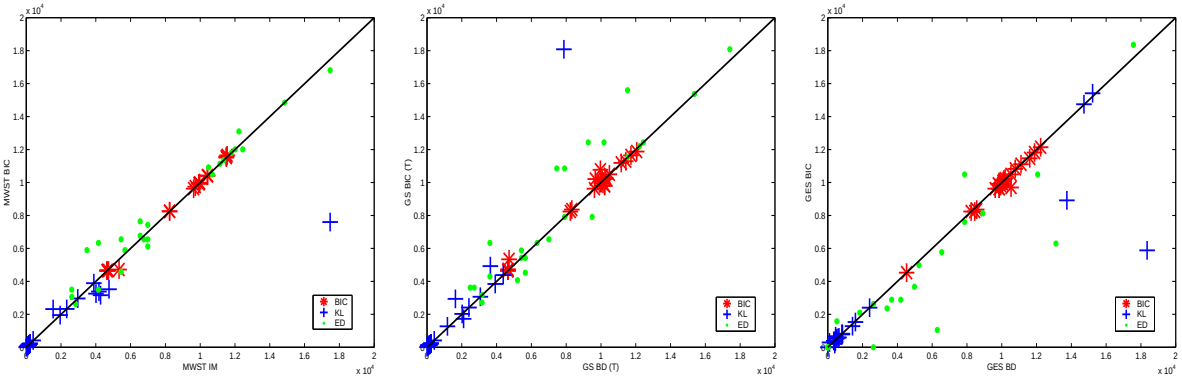
ASIA de la figure 9.1(f), le nœud A possède un état de probabilité *a priori* faible. De plus cet état a une influence également faible sur le nœud T. Les résultats qui avaient été obtenus sont représentés sur la table 9.1.

Pour la plupart des méthodes, cet arc entre A et T n'a pas été retrouvé. Cependant, il arrive que les algorithmes MWST, PC, K2-T et GES y parviennent lorsque la base de données est suffisamment grande. Pour les méthodes à base de score comme GS, l'ajout de cet arc ne peut se faire que s'il permet d'augmenter le score du réseau bayésien. Même si la découverte de ce lien permet d'augmenter la vraisemblance du réseau bayésien, cela n'est pas suffisant par rapport à l'augmentation du terme de pénalité associé à la dimension du réseau bayésien.

9.1.3.5 Choix du score

Comme nous pouvons le voir sur la figure 9.6(a), utiliser l'information mutuelle pour la méthode MWST ou bien le score *BIC*, sont deux techniques qui se valent pour le score *BIC* et la KL-divergence car tous les points (rouges et bleus hormis un) sont très proches de la bissectrice. Nous voyons également que du point de vue de la distance d'édition (points verts), il semble légèrement plus avantageux d'utiliser l'information mutuelle.

Pour les méthodes gloutonnes, il paraît plus intéressant d'utiliser le score bayésien pour la méthode gloutonne simple. En effet, sur la figure 9.6(b), nous observons que de nombreux points sont repoussés dans la partie de la méthode GS-bic, ce qui signi-



(a) Duel MWST avec l'information mutuelle en abscisses et MWST avec le score *BIC* en ordonnées.

(b) Duel GS+T avec le score *BD* en abscisses et GS+T avec le score *BIC* en ordonnées.

(c) Duel GES avec le score *BD* en abscisses et GES avec le score *BIC* en ordonnées.

FIG. 9.6 : *Quelques duels : les points en rouge représentent l'opposé des scores BIC, ceux en bleu, la divergence de KL, et ceux en vert, la distance d'édition. Comme il ne s'agit que de valeurs à minimiser, la méthode qui possède le moins de points de son coté est estimée meilleure.*

fie que la méthode GS-bd obtient des scores plus faibles (ou égaux) donc meilleurs (ou équivalents) pour toutes les mesures ici présentées (opposé du score *BIC* en rouge, KL-divergence en bleu et distance d'édition en vert).

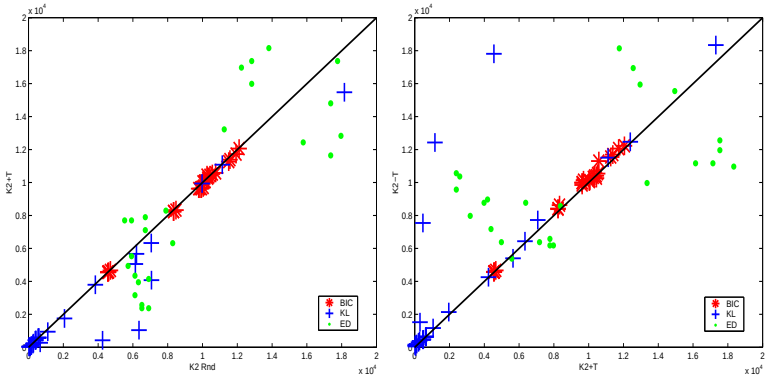
Pour la recherche gloutonne dans les équivalents de Markov, la différence est moins marquée. Au contraire, il semblerait qu'il soit préférable d'utiliser le score *BIC* pour cette méthode et seulement du point de vue de la distance d'édition (points verts de la figure 9.6(c)).

Remarquons qu'utiliser le score bayésien ou le score *BIC* a peu d'influence sur le score *BIC* final de la structure, ce qui est logique car le score *BIC* est une approximation du score bayésien (voir section 7.3) Lorsque le score bayésien est utilisé, le nombre d'arcs de la structure résultat est souvent plus elevé. En effet, l'augmentation de la vraisemblance même faible n'est alors pas compensée par un terme de pénalité.

9.1.3.6 Choix de l'initialisation

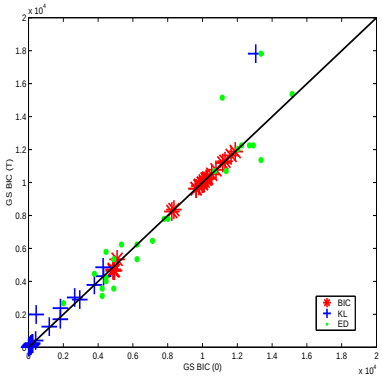
Comme nous pouvons le voir sur les figures 9.7(a) et 9.7(b), la méthode K2 initialisée par l'ordre de l'arbre rendu par MWST (K2+T) l'emporte légèrement face aux méthodes K2 initialisées aléatoirement et K2-T pour les critères de distance d'édition et de divergence de Kullback-Leiber (les scores *BIC* sont identiques). De plus, la figure 9.5 montre qu'une telle initialisation permet de réduire la variance des résultats significativement, que ce soit du point de vue de la divergence de Kullback-Leiber ou du score *BIC*. Par ailleurs, cette initialisation n'a aucune influence sur le temps de calcul global de la méthode.

Dans la section 8.1.3.2, nous proposions deux initialisations pour K2. Les résultats des tables 9.1 et 9.2 nous montrent les caractéristiques des réseaux obtenus par K2 pour deux initialisations aléatoires. Nous voyons que les graphes obtenus avec les différentes initialisations de K2 sont très différents, tant du point de vue de la structure (table 9.1) que des scores *BIC* des structures obtenues (table 9.2). Pour le réseau ASIA, la stratégie K2-T semble plus payante que la stratégie K2+T, cela est dû au fait que la classe (le

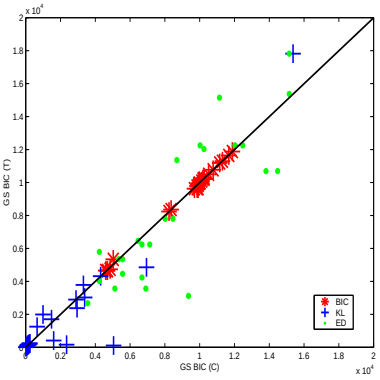


(a) Duel K2(R)-K2+T

(b) Duel K2+T-K2-T



(c) Duel GS+0-GS+T



(d) Duel GS+C-GS+T

FIG. 9.7 : *Duels sur le type d'initialisation. Le '+0' signifie initialisation vide, le '+C' signifie initialistion par une chaîne et le '(R)', initialisation aléatoire. La première méthode est en abcisses et la seconde en ordonnées.*

nœud D représentant l'*insuffisance respiratoire*) est une cause des autres facteurs (*fumeur, tuberculose, etc*). Par contre, pour le réseau INSURANCE la stratégie K2+T a donné de meilleurs résultats. Par ailleurs nous constatons sur les figures 9.2 et 9.3(a) que la méthode K2+T permet d'obtenir de meilleurs résultats que la méthode K2. La méthode K2-T n'ayant alors pas l'air d'être une initialisation pertinente pour K2. Elle donne de moins bons résultats que le meilleur des résultats parmi 5 lancements de K2 avec des initialisations aléatoires là où K2+T donne statistiquement des résultats de meilleure qualité.

Dans la section 8.2.2.2, nous proposions d'utiliser la structure obtenue par MSWT comme initialisation de l'algorithme glouton GS. Contrairement aux méthodes K2+T et K2-T pour lesquelles nous ne constatons aucune différence au niveau du temps de calcul, les temps d'exécution de la méthode GS lorsqu'elle est initialisée par une structure vide, par une chaîne ou par l'arbre rendu par MWST sont très différents. Autant au niveau des résultats ces techniques d'initialisation se valent (figures 9.7(c), 9.7(d) et 9.6(b)), autant il est étonnant de voir que l'initialisation de la méthode gloutonne par une chaine est une technique qui allonge considérablement les temps de calculs, alors que l'initialisation par l'arbre rendu par MWST permet d'accélérer la méthode, les temps sont alors équivalents avec le score bayésien ou le score BIC, et sont même inférieurs à ceux obtenus par la méthode GES (qui obtient tout de même des résultats de meilleure qualité, voir les figures 9.2(a), 9.2(b), 9.3(a)).

Pour résumer, utiliser la structure arborescente (ou son ordre topologique) permet d'obtenir de meilleurs résultats et plus stables avec la méthode K2 sans changer la vitesse d'exécution et permet d'obtenir des résultats similaires avec la méthode GS mais en diminuant significativement le temps d'exécution de calcul.

9.1.3.7 Duel inter-méthodes

Après avoir observé quelles étaient les meilleures techniques d'initialisation ainsi que les scores les plus favorables pour chaque méthode, regardons à présent quelles sont les meilleures techniques d'apprentissage de structure.

Sur la figure 9.8(a), nous voyons que la méthode PC permet d'obtenir de meilleurs résultats sur la divergence de Kullback-Leiber (points bleus), tandis qu'elle semble plus mauvaise du point de vue de la distance d'édition (points verts). Les deux méthodes se valent pour le score *BIC* (points rouges).

La figure 9.8(b) nous montre que MWST semble permettre d'obtenir de meilleurs résultats que PC selon le score *BIC* et la distance d'édition, mais est moins bonne pour la distance de Kullback-Leiber. Ceci est confirmé par les figures 9.8(c), 9.8(d), 9.8(e),9.8(f), qui nous montrent que systématiquement, MWST est plus mauvais du point de vue de la divergence de KL. Cela est certainement dû au fait que la structure d'arbre ne permet pas à un nœud d'avoir plusieurs parents, et donc il n'y a pas de grandes tables de probabilités conditionnelles pour les réseaux bayésiens issus de cette méthode. Donc le résultat possède moins de paramètres, ceux-ci sont donc appris plus précisément lorsque la base est de faible taille, mais ne permet pas d'encoder des lois compliquées.

Sur les figures 9.8(g), 9.8(h), 9.8(i), nous nous apercevons que la méthode BNPC, obtient systématiquement des résultats plus mauvais du point de vue du score *BIC* (points rouges de son coté) que les autres méthodes considérées. Il est néanmois difficile de dire si cette méthode est plus mauvaise que les méthode K2 ou GS. Comme les nuages de points sont écartés, nous pouvons alors dire que cette méthode obtient tout de même de meilleurs résultats avec certains exemples pour les autre mesures d'évaluation.

Finalement nous nous apercevons sur les figures 9.8(f), 9.8(i), 9.8(j), 9.8(k) que la méthode GES donne systématiquement des meilleurs résultats que les autres méthodes, et cela quel que soit le critère d'évaluation (si ce n'est le temps de calcul, voir figure 9.3(b)). Ceci vérifie bien le fait que cette méthode possède un caractère optimal.

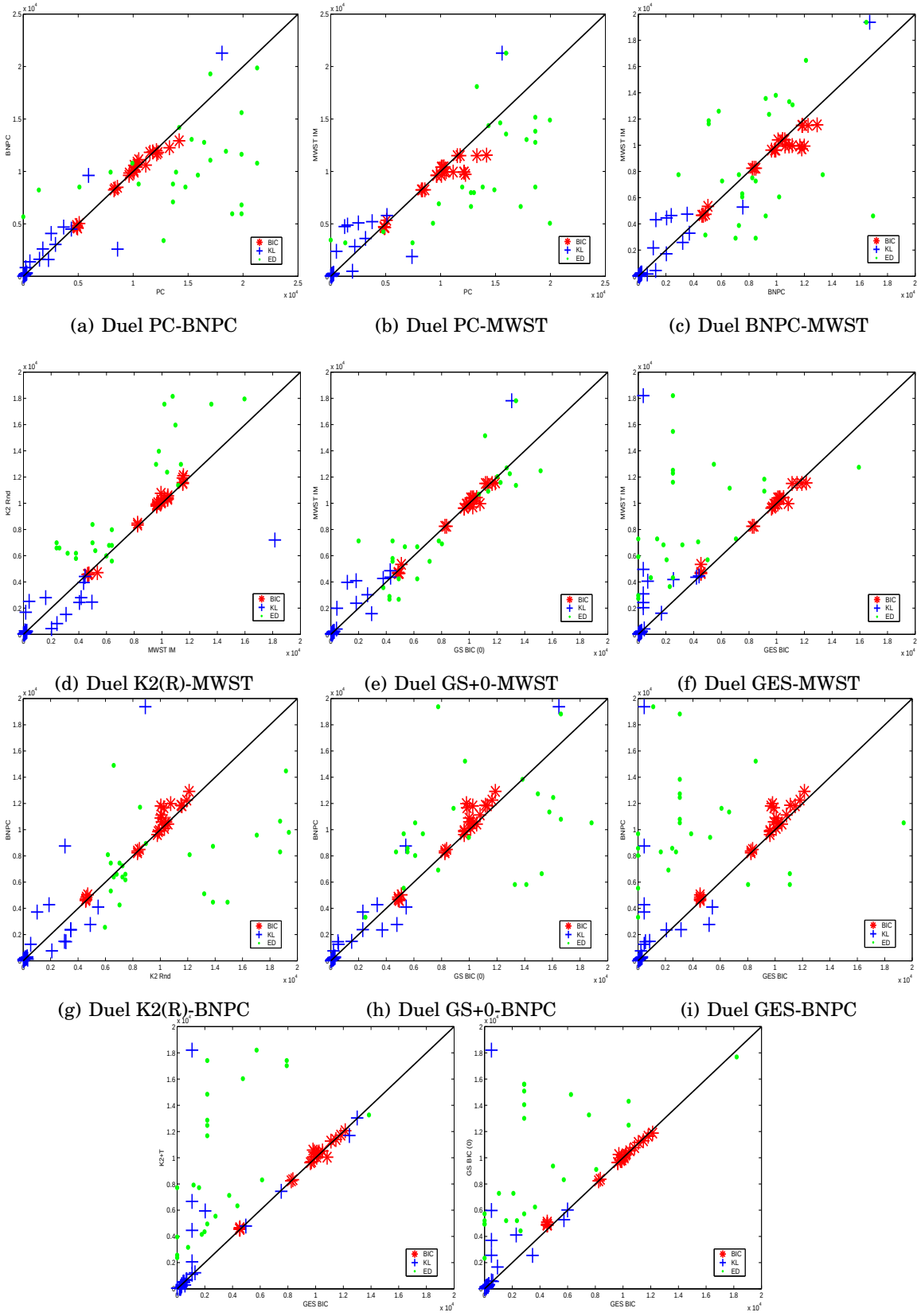
9.2 Recherche d'un bon classifieur

9.2.1 Algorithmes utilisés et techniques d'évaluation

Le critère de comparaison entre les méthodes est ici le taux de bonne classification mesuré sur les données de tests.

Nous considérons également le score *BIC* sur les bases d'exemples de tests comme critère d'évaluation de la pertinence de la structure obtenue (Tous les résultats sont visibles en détails dans l'annexe F.2). Le temps de calcul est également indiqué à titre indicatif.

Comme critère supplémentaire, nous considérons une grandeur appellée la *puissance BIC*, qui est la capacité d'une méthode à se classer parmi les meilleures méthodes du point de vue du score *BIC*. Cette *puissance BIC* est calculée comme étant la somme sur les bases de tests des rapports des scores *BIC* obtenus sur les différentes bases d'exemples par la méthode divisée par le meilleur score *BIC* obtenu sur chaque base



(a) Duel PC-BNPC (b) Duel PC-MWST (c) Duel BNPC-MWST

(d) Duel K2(R)-MWST (e) Duel GS+0-MWST (f) Duel GES-MWST

(g) Duel K2(R)-BNPC (h) Duel GS+0-BNPC (i) Duel GES-BNPC

(j) Duel GES-K2+T (k) Duel GES-GS+0

FIG. 9.8 : *Duels inter-méthodes, les étoiles rouges représentent l'opposé des scores BIC, les signes plus bleus, la divergence de KL et les points verts, la distance d'édition. Comme ces grandeurs sont à minimiser, la méthode qui a le moins de points dans sa moitié est estimée meilleure. La première méthode est en abscisse et la seconde en ordonnée.*

de tests. En clair, si une méthode obtient 1, c'est qu'il s'agit de la méthode qui obtient systématiquement le meilleur score *BIC*. Par contre, si sa puissance *BIC* vaut $0,5$, cela signifie qu'elle obtient, en moyenne, un score *BIC* qui vaut la moitié de celui obtenu par la meilleure méthode pour chaque base.

Pour les tests, nous avons considéré le réseau bayésien naïf (NB) ainsi que les méthodes d'apprentissage de structure suivantes :

- TAN-im classifieur de bayes naïf augmenté par une arbre avec l'information mutuelle,
- TAN-bic classifieur de bayes naïf augmenté par une arbre avec les score *BIC*,
- MWST-im avec l'information mutuelle,
- MWST-bic avec les score *BIC*,
- K2 initialisé aléatoirement (meilleur résultat obtenu parmi 5 lancements) avec le score *BD*,
- K2+T avec le score *BD* (classe comme racine de l'arbre),
- K2-T avec le score *BD* (classe comme racine de l'arbre),
- PC,
- BNPC,
- GS-0 initialisé avec la structure vide, scores *BD* et *BIC*,
- GS-C initialisé avec une chaîne, scores *BD* et *BIC*,
- GS+T initialisé avec l'arbre rendu par MWST, score *BD* (*init.* MWST-IM) et score *BIC* (*init.* MWST-BIC),
- GES avec le score *BIC*.

Les résultats sont synthétisés sur la figure 9.9.

9.2.2 Résultats et interprétations

Les taux de classification pour chaque méthode et pour chaque base d'exemples décrites en annexe E sont disponibles en annexe (tableaux F.5 et F.6).

Comme nous pouvons le voir sur la figure 9.9, seule la moitié des méthodes d'apprentissage de structure obtiennent de meilleurs résultats en classification que le réseau bayésien naïf. Les premières à la dépasser sont les arbres augmentés d'une structure arborescente. Ces méthodes obtiennent presque 1% de bonnes classifications supplémentaires en moyenne là où les structures obtenues par MWST obtiennent en moyenne $1,5\%$ de taux de bonne classification en moins. Ceci est dû au fait que pour une tâche de classification, plus le nœud représentant la classe aura de voisins, plus la classification prendra en compte d'informations.

Ensuite, nous pouvons voir que la méthode K2+T avec un seul lancement, obtient des résultats identiques (voire meilleurs) que K2 lorsque l'on garde son meilleur résultat parmi 5 lancements. De plus nous observons également, que l'initialisation avec l'ordre inverse de l'ordre rendu pas MWST est plutôt une mauvaise initialisation, les résultats descendent alors au niveau de ceux obtenus par la méthode PC. Nous pouvons remarquer que les résultats obtenus par MWST sont équivalents à ceux obtenus pas K2 en classification. Cela est surprenant, et lancer K2 avec l'ordre issu de MWST permet d'obtenir une initialisation de K2 efficace, mais en moyenne, ne depasse pas le pouvoir classifiant de l'arbre obtenu par MWST.

Les méthodes d'identification de dépendances conditionnelles obtiennent les résultats en classification les plus bas, tandis que les méthodes gloutonnes en obtiennent de meilleurs. Initialiser une méthode gloutonne par une chaîne est alors une stratégie plutôt mauvaise si l'on veut effectuer une tâche de classification. Par contre, l'initialiser

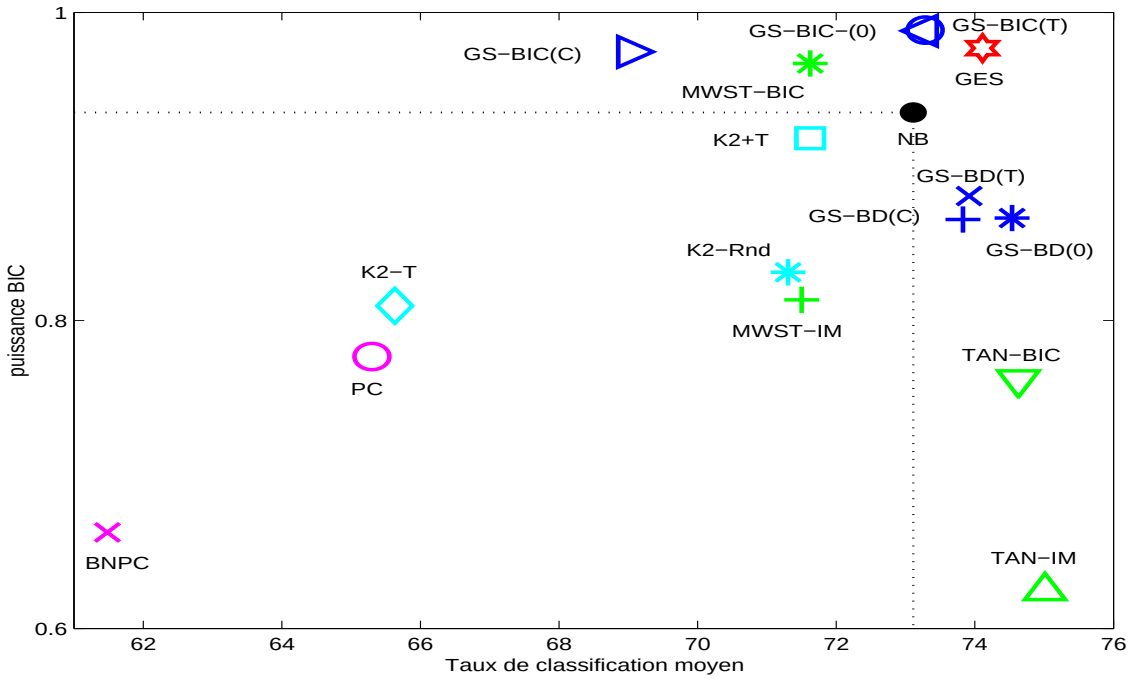


FIG. 9.9 : *Schéma représentant la moyenne du taux de classification pour chaque méthode sur les bases d'exemples utilisées. En ordonnées, nous inscrivons à titre indicatif la puissance BIC, c'est-à-dire la capacité d'une méthode à obtenir un bon score BIC.*

avec l'arbre rendu par MWST permet d'obtenir des résultats similaires à une initialisation avec le graphe vide, mais permet un gain de temps de calcul. A titre indicatif, les temps de calcul moyens étant alors (en secondes) : $MWST\,im = 1.25$, $MWST\,bic = 1.74$, $TANB\,im = 1.13$, $TANB\,bic = 1.55$, $K2\,Rnd = 4.43$, $K2 + T = 3.69$, $K2 - T = 6.00$, $PC = 146.96$, $BNPC = 30.08$ [i], $GS\,bic = 249.85$, $GS + C\,bic = 223.93$, $GS + T\,bic = 95.27$, $GS\,bd = 319.46$, $GS + C\,bd = 340.06$, $GS + T\,bd = 193.69$, $GES = 222.73$. Nous voyons alors que l'initialisation par un arbre permet de diviser les temps de calcul d'une recherche gloutonne tandis que l'initialisation par une chaîne provoque une perte de temps avec le score BD.

Pour ce qui est des méthodes obtenant de bons scores BIC, nous nous aperçevons que ce ne sont pas toujours les mêmes que celles qui obtiennent de bons scores de classification. Par exemple, les méthodes à base de classifieur de Bayes naïf augmenté par un arbre, obtiennent de très bons résultats en classification, mais de très mauvais scores BIC tandis qu'une recherche gloutonne initialisée par une chaîne aléatoire permet d'obtenir de bons scores BIC mais de mauvais résultats en classification. Ceci est certainement dû au fait, que le nœud représentant la classe n'est alors pas positionné initialement en voisin des 2 nœuds avec lesquels les corrélations sont les plus fortes, et la recherche gloutonne n'arrive alors apparemment pas à récupérer cette erreur d'initialisation.

Il est tout de même remarquable de voir que cette fois-ci, initialiser K2 avec l'arbre rendu par MWST permet d'obtenir des structures avec de bons scores BIC.

[i] Les valeurs pour les méthodes PC et BNPC sont seulement données à partir des exemples sur lesquels elles n'ont pas provoquées de *crash* mémoire

Le score bayésien permet d'obtenir de bons taux de classification lorsqu'il est utilisé par une méthode gloutonne. Cela est certainement dû au fait que les structures obtenues ont de nombreux arcs. Elles permettent alors de représenter plus finement de nombreuses lois. Une recherche gloutonne avec le score BIC ne permet pas autant de précision pour la représentation de la loi, par contre, la structure obtenue est alors plus lisible et moins complexe, elle possède alors un meilleur score BIC.

Par ailleurs, il est surprenant de voir que le réseau bayésien naïf obtient en moyenne un score BIC élevé. Les structures obtenues par TANB sont plus complexes et ont donc un score beaucoup plus bas, même si à la fois l'arbre rendu par MWST et la structure naïve ont tous deux de bons scores.

Les méthodes les plus efficaces restent cependant GES et les méthodes de recherche gloutonnes. Pour une tâche de classification, il semble préférable d'utiliser le score bayésien. Les méthodes gloutonnes ayant de grandes complexités, de très bons compromis restent alors MWST et K2+T pour le score BIC, et, la structure naïve et la méthode TANB pour le taux de classification.

Dans ces expérimentations, nous avons appris les paramètres des différents réseaux bayésiens avec le maximum de vraisemblance classique. Néanmoins, si nous avions utilisé la vraisemblance classifiante, nous aurions certainement obtenu de meilleurs résultats en classification car cette méthode permet d'apprendre plus fidèlement les probabilités conditionnelles à la classe qui vont être utilisées pour la tâche de classification.

Toutefois, les différences entre les différentes méthodes d'apprentissage de structure auraient vraisemblablement été les mêmes si nous avions utilisé cet autre critère pour toutes les méthodes. Par ailleurs, une étude récente menée par Pernkopf & Bilmes (2005) montre qu'utiliser une approche discriminative pour apprendre les paramètres ou la structure d'un classifieur de Bayes (naïf ou augmenté par un arbre), permet d'augmenter le taux de classification moyen d'environ $0,5\%$ (pour 25 bases d'exemples testées). Néanmoins, la méthode la plus performante reste alors de choisir d'utiliser le taux de classification comme fonction objectif, et dans ce cas, utiliser la vraisemblance semble être plus efficace que d'utiliser la vraisemblance conditionnelle. De plus, lors de l'apprentissage de multinets, les résultats de Pernkopf & Bilmes (2005) montrent qu'utiliser la vraisemblance ou la vraisemblance classifiante donne des résultats similaires.

Dans cette section, nous avons été amenés à tester les méthodes décrites dans le tableau 9.3 pour une problématique d'identification de structure et une problématique de classification. Les méthodes ont alors été classées à titre indicatif en fonction de leurs performances pour différents critères dans le tableau 9.4.

Méthodes	scores	initialisations	classification
PC	χ^2		
BNPC	information mutuelle conditionnelle		
MWST	information mutuelle, *BIC*	chaîne aléatoire	racine=classe
K2	*BD*	ordre aléatoire, ordre issu de MWST et inversé	classe : 1*er* ou dernier nœud
GS	*BD* et *BIC*	structure vide, chaîne aléatoire, résultat de MWST	
GES	*BIC*		
TAN	information mutuelle, *BIC*		spécifique

TAB. 9.3 : *Méthodes testées dans le chapitre* 2.

Méthodes	Ind.Cond.		MWST		K2			GS				GES	
	PC	BNPC	IM	BIC	Rnd	+T	-T	BIC(0)	BIC(C)	BIC(T)	BD(T)	BIC	BD
Performances BIC	13	12	8	9	10	7	11	**5**	6	**2**	**4**	**1**	**3**
Performances KL	9	10	12	**5**	6	**3**	7	8	13	11	**4**	**1**	**2**
Perf. distance d'édition	11	**4**	7	8	12	10	13	**5**	9	5	**3**	**1**	**2**
Stabilité BIC	12	13	**2**	**1**	11	6	10	7	9	8	**5**	**3**	**4**
Stabilité KL	6	10	9	**3**	11	**5**	11	7	13	8	**4**	**1**	**2**
Temps de calcul	7	6	**1**	**2**	**3**	**3**	**5**	12	13	9	8	10	11

Méthodes		ind.Cond.		MWST		TAN		K2			GS-BIC			GS-BD			GES
	NB	PC	BNPC	IM	BIC	IM	BIC	Rnd	+T	-T	(0)	(C)	(T)	(0)	(C)	(T)	BIC
Classification	9	16	17	10	12	**1**	**2**	13	11	15	7	14	7	**3**	6	**5**	**4**

TAB. 9.4 : *Classements des méthodes d'apprentissage de structure testées pour différents critères.*

Conclusion

Dans cette partie, nous avons introduit différentes méthodes d'apprentissage de structure, et comparé ces dernières empiriquement.

Il existe principalement deux techniques d'apprentissage de structure. Le premier type se consacre à trouver les (in)dépendances conditionnelles entre les variables par le biais de tests statistiques. Cette approche permet d'estimer des structures de réseaux bayésiens qui peuvent être interprétées en terme de causalité. Seulement comme de nombreuses relations de causalité ne peuvent être identifiées sans l'ajout de tests supplémentaires (en utilisant des données d'expérimentation Meganck, Leray & Manderick (2006)), ces méthodes ne rendent comme résultat qu'un graphe partiellement orienté, qui doit alors être orienté pour devenir un réseau bayésien.

Le seconde type d'approches consiste en l'utilisation d'une fonction de score. L'interprétabilité du graphe obtenu est alors issue de l'interprétabilité du score. En utilisant le score BD, nous obtenons des structures permettant de coder correctement la loi jointe. Seulement les graphes obtenus sont alors souvent très connectés et de nombreuses indépendances conditionnelles ne sont pas découvertes. Il est alors possible d'utiliser des scores qui sont composés d'un terme de vraisemblance et d'un terme de pénalité, dans ce cas les réseaux sont moins denses et plus interprétables, néanmoins les performances en classification pure sont légèrement moins bonnes. En effet pour la classification avec des données complètement observées, seules les variables de la frontière de Markov de la variable classe sont prises en compte. Or, comme les graphes sont moins denses, la variable classe possède moins de voisins, et moins d'informations sont alors prises en compte pour la classification.

Que l'on choisisse le premier ou le second type d'approche, il faut faire des compromis pendant l'apprentissage car l'espace de recherche est beaucoup trop vaste. Les compromis peuvent alors être sur la donnée d'un nombre maximum de parents, sur le choix d'un ordre d'énumération, sur la restriction de la recherche à des structures plus simples, ou sur l'utilisation des techniques issues de la recherche opérationnelle pour trouver rapidement un optimum dont il n'est alors pas garanti qu'il soit *global*.

Les différents choix à faire doivent alors être guidés par l'objectif que nous nous donnons. En effet, nous devons nous poser les questions pour cerner quelle interprétabilité nous voulons du résultat, ainsi que quel niveau de performance ou de généralisation nous comptons obtenir. Par exemple, si nous voulons un outil qui servira surtout d'outil de calcul et que la base d'exemples y figure le plus précisément possible sans se soucier de la généralisation du résultat, le score bayésien paraît être une bonne solution s'il est combiné à un algorithme

K2 par exemple. A contrario, si nous nous intéressons plus particulièrement à un modèle interprétable, il faudra utiliser des mesures de score pénalisées.

Nous avons alors vu que la méthode GES permettait d'obtenir de meilleurs résultats que les autres techniques testées sur des données synthétiques, tant au niveau de la qualité des réseaux bayésiens obtenus que sur de la stabilité des résultats qu'elle rend.

La méthode MWST donne, quant à elle, des résultats particulièrement stables. Elle permet également d'obtenir des réseaux ayant de bons scores *BIC*, néanmoins, la restriction de son espace de recherche ne permet pas d'obtenir de KL-divergences satisfaisantes. Cette méthode permettant tout de même d'obtenir des résultats compétitifs lorsque le nombre d'exemples de la base est faible.

La méthode K2+T est significativemement plus efficace que la méthode K2 pour tous les critères observés. Il n'y a cependant pas de différence significative entre K2-T et K2 si ce n'est un apport de stabilité.

La méthode GS+T conserve la qualité des résultats obtenus avec GS mais permet de réduire considérablement les temps de calcul de cette méthode itérative.

Finalement, la méthode TAN, qui est spécifique à la classification, est une légère variante de la méthode MWST qui donne, quant à elle, les meilleures performances en classification que toutes les méthodes que nous avons testées.

Les perspectives de ce travail sont en premier lieu de tester plus de méthodes d'apprentissage sur des problèmes plus difficiles et sur davantage de bases de cas réelles. Puis d'adapter ces méthodes aux données mixtes, ou encore, à la prise en compte de connaissance *a priori*. Cette connaissance peut être dûe à un expert.

Par exemple, nous pourrions choisir de conserver un arc fixe durant l'apprentissage. Dans ce cas, il faut pouvoir adapter le calcul du score. Remarquons qu'un travail similaire est effectué dans la méthode de recherche du classifieur de Bayes naïf augmenté par un arbre, en effet tous les arcs partant du nœud classe vers les autres nœuds sont fixes et le calcul du score est changé en conséquence (dans ce cas, cela est plus simple car cela reste symétrique étant donné que tous les nœuds sont reliés à la classe).

Il serait également possible de prendre en compte de l'information issue du problème lui même. Par exemple, s'il est connu qu'une variable cachée est pertinente pour le problème, la prendre en considération. Ou encore, s'il est connu que le problème possède un aspect dynamique, peut-être pourrions nous adapter ces méthodes à l'apprentissage de la structure *intra-tranche* et/ou *inter-tranches*.

Pour de nombreux problèmes, il peut être avantageux de prendre en compte ou découvrir des variables latentes. Ces variables n'étant pas observées, il faut alors pouvoir effectuer de l'apprentissage de structure lorsque la base d'exemples est incomplète. Ce point est alors l'objet du chapitre suivant.

Troisième partie

Apprentissage de Structure à partir de données incomplètes

Introduction

De nos jours, de plus en plus de bases de cas sont disponibles, or de nombreuses bases sont incomplètes. Les raisons pour lesquelles certaines valeurs peuvent être manquantes sont multiples. Par exemple, un instrument de mesure peut être en panne durant une période. Une mesure peut alors ne pas être réalisable lorsque le système entre dans un état particulier (en répondant ”non” à une question d'un questionnaire, il arrive souvent que certaines questions suivantes ne soient pas pertinentes) ou à cause d'interférences extérieures (comment maintenir la surveillance d'une étoile lorsque le ciel est nuageux). Ou il se peut alors simplement que l'opérateur ait oublié de reporter une donnée, ou que la donnée reportée ne soit pas lisible.

Néanmoins, quand nous voulons construire un modèle à partir d'une base d'exemples incomplète, il est souvent possible de le faire en n'utilisant que les exemples complètement observés de la base. Seulement, avec cette approche, nous n'avons plus la même quantité d'informations pour apprendre un modèle. Par exemple, considérons que nous soyons en présence d'une base contenant 2000 exemples sur 20 attributs, mais dont la probabilité qu'une valeur soit manquante vaille 20%. Dans ce cas, nous n'avons seulement en moyenne que 23 cas complètement observés. En généralisant cet exemple, nous voyons que le problème des données incomplètes ne peut pas être ignoré.

Dans cet exemple, nous avons considéré le cas particulier où la probabilité *a priori* qu'une valeur soit manquante ne dépendait de rien.

Comme nous l'avons vu en section 5.3.1 il existe trois types de données manquantes : MCAR, MAR et NMAR mis en évidence par Rubin (1976). Ces différents types sont différenciés par les caractéristiques du processus qui a généré les données. En particulier, en fonction de quelles variables dépend la probabilité qu'une entrée soit manquante.

Lorsque nous effectuons de la recherche d'informations dans une base de cas incomplète, il peut être utile de connaître ou d'identifier le processus qui a généré les données. Par exemple, si nous n'avons pas pris une mesure à cause des intempéries, il est clair que cette valeur est manquante au hasard. Par contre, si nous n'avons pas pris une mesure car la question n'était plus pertinente, ayant répondu ”non” à une question précédente, cette absence de mesure est une information sur l'état du système (en particulier si la réponse négative à la question précédente est elle-même manquante également).

Que faire d'une valeur manquante ? La première idée qui vient à l'esprit est de simplement supprimer les exemples qui ne sont pas complets (étude des

cas complets, voir page 56), seulement cette idée n'est pas satisfaisante lorsque le taux de valeurs manquantes est élevé.

La deuxième idée pourraît être de remplacer une valeur manquante par un état supplémentaire à ajouter à la variable. Seulement ce type de modélisation introduit souvent un biais.

En utilisant une technique d'apprentissage de type EM, le temps de calcul croît exponentiellement par rapport au nombre de valeurs manquantes. La modélisation par un état supplémentaire permet donc de diminuer le nombre de valeurs manquantes et donc de diminuer le temps de calcul.

Voyons à présent quand une telle modélisation peut être fondée.

Comment détecter un processus qui a généré des valeurs manquantes ? La modélisation par un état supplémentaire ne peut se faire que dans certains cas particuliers. Il faut être en présence d'une base d'exemples générée par un processus NMAR. Seulement, le cas NMAR contient le cas MAR, et il faut être capable de différencier les valeurs qui sont manquantes de manière aléatoire de celles qui sont manquantes en présence d'un état particulier du système.

Pour cela, nous pouvons, par exemple, considérer les probabilités suivantes

$$\mathbb{P}(X_i = \text{'manquant'} | \mathbf{D}, X_j = x_j^{k_j}, j \in J) \tag{1}$$

pour $i \in \{1 \dots n\}$, $k_j \in \{1, \dots, r_j\}$ et $J \subset \{1 \dots n\} \setminus \{i\}$.

En pratique, comme le nombre d'exemples disponibles décroit lorsque la cardinalité de J augmente et comme la cardinalité de $2^{\{1 \dots n\} \setminus \{i\}}$ augmente de manière exponentielle avec le nombre de variables n, nous devons nous limiter aux ensembles J de faible cardinalité (par exemple $|J| < \frac{\ln N}{2}$).

Ensuite, si pour un certain ensemble de X_j et pour une certaine configuration de ces variables, la probabilité de l'équation 1 est élevée (proche de 1) nous pouvons estimer que l'ajout d'un état pour la variable X_i au lieu de la considérer manquante est pertinent dans ce cas. Par contre, si cette probabilité est faible, ou s'il n'y a que trop peu d'exemples de cette configuration représentés dans $\mathbf{D}$, cette modélisation n'est pas pertinente.

Méthodes classiques pour traiter les données manquantes :

Comme pour l'apprentissage de structure à partir de bases de données complètes, il existe deux techniques principales : les méthodes à base de recherche de dépendances conditionnelles, et celles à base de score.

Pour les premières, comme les tests statistiques ne peuvent être effectués à partir de données incomplètes, il faut recourir soit à l'effacement des entrées incomplètes de la base d'exemples, soit à la substitution des valeurs manquantes par des valeurs admissibles.
Les techniques les plus populaires pour traiter les bases d'exemples incomplètes sont :

Étude des cas complets : La première méthode pour traiter des bases de données incomplètes est de simplement ignorer les exemples qui possèdent au moins

une variable non observée. Si les données manquantes sont aléatoirement distribuées (*i.e.* base d'exemples MCAR) alors le retrait des exemples incomplets permet de conserver une base telle que l'estimateur de maximum de vraisemblance avec cette base sera sans biais. Néanmoins, si ce n'est pas le cas, cette méthode n'est pas recommandée car elle introduit alors un biais. Par ailleurs, lorsqu'il y a de nombreuses données manquantes, cette approche n'est pas satisfaisante car le nombre de cas complets peut alors être très faible ou nul (cf. pages 56 et suivantes). Cette méthode n'est donc efficace que lorsque le pourcentage de données manquantes est faible et leur distribution relativement uniforme.

Étude des cas disponibles (*pairwise data deletion*)**:** Lorsque nous effectuons un test sur un nombre limité de variables, il est possible de ne considérer que les cas où ces variables sont complètement observées. Comme pour l'étude des cas complets, cette approche introduit un biais lorsque les données ne sont pas aléatoirement distribuées.

Substitution par la moyenne : Remplacer les valeurs manquantes par la moyenne (ou le mode) de la variable correspondante calculée sur les cas complets de cette variable est une approche qui possède l'avantage de travailler avec la base entière. Néanmoins les principaux désavantages sont, d'une part que ce type de substitution fait artificiellement décroître les variations de score et plus il y a des données manquantes plus cette variation sera faible, et d'autre part, la substitution par les valeurs moyennes peut considérablement changer les corrélations entre les variables.

Méthode du maximum de vraisemblance : Après identification des cas les plus semblables à l'exemple ayant des valeurs manquantes, nous substituons cette valeur par la valeur la plus vraisemblable. Une manière de faire cela est d'utiliser un algorithme de type *Expectation Maximization* (EM).

Multiple imputation : De manière similaire à la *méthode du maximum de vraisemblance*, il s'agit de remplacer les valeurs manquantes par d'autres valeurs. Sauf que l'imputation multiple utilise des valeurs réelles appropriées à partir des données brutes pour compléter la base de données existante. Typiquement, cinq à dix bases de données sont créées de cette façon. Ces matrices de données sont ensuite analysées en utilisant une méthode appropriée d'analyse statistique et en traitant ces bases comme si elles étaient des bases de données complètes. Les multiples résultats de ces analyses sont alors combinés dans un unique modèle récapitulatif simple.

Toutes ces méthodes sont particulièrement adaptées aux données MAR (et en particulier MCAR). Par la suite nous allons nous concentrer sur les méthodes basées sur l'étude des cas complets, l'étude des cas disponibles et principalement sur les principes de l'algorithme EM.

Après un rappel détaillé sur le fonctionnement de l'algorithme EM, nous allons voir comment calculer un score pour un modèle à partir d'une base de cas incomplète. Puis, suite à un état de l'art des différentes techniques d'identification de structure de réseau bayésien à partir de données incomplètes, nous allons introduire un algorithme qui travaille dans l'espace des arbres, qui est optimal

et avec une complexité faible pour découvrir l'arbre qui représente au mieux la base d'exemples.

Nous finirons par comparer l'algorithme SEM introduit par Friedman (1997) avec notre implémentation et nous observerons ce qui se passe lorsque l'algorithme SEM est initialisé avec l'arbre optimum que notre méthode découvre. Nous comparerons également ces différentes techniques avec leur implémentation qui n'utilise que les *cas complets* et les *cas disponibles* de la base d'exemples pour évaluer la pertinence de l'utilisation d'une méthode gourmande en temps de calcul comme l'algorithme EM.

10

Rappel sur l'algorithme EM

Les devises Shadok, Jacques Rouxel (1931-2004)

Sommaire

10.1 L'algorithme *Expectation-Maximisation* **124**
10.1.1 Introduction 124
10.1.2 Développement de la méthode 124
Maximum de vraisemblance avec données incomplètes 124
Remarques préliminaires 124
10.1.3 *Expectation-Maximisation* 126
10.1.4 Convergence 127
10.2 Les adaptations de l'algorithme EM **127**
10.2.1 EM généralisé 127
10.2.2 EM pour la classification 127
10.2.3 EM incrémental 127
10.2.4 Monte-Carlo EM 127
10.2.5 EM variationnel 128

10.1 L'algorithme *Expectation-Maximisation*

10.1.1 Introduction

L'algorithme EM est une méthode itérative pour évaluer le maximum de vraisemblance en présence de données incomplètes Dempster, Laird & Rubin (1977) et McLachlan & Krishnan (1996). Sa convergence est dûe à la convexité de la log-vraisemblance.

Soit f une fonction convexe sur l'intervale I (i.e. $f(\lambda x_1 + (1-\lambda)x_2) \leqslant \lambda f(x_1) + (1-\lambda)f(x_2)$). Si $x_1, \ldots, x_n \in I$ et $\lambda_1, \ldots, \lambda_n \geqslant 0$ avec $\sum_i \lambda_i = 1$, alors par généralisation de cette définition de la connexité nous obtenons :

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leqslant \sum_{i=1}^{n} \lambda_i f(x_i) \tag{10.1}$$

Comme ln est concave, $-\ln$ est convexe et nous avons

$$\ln \sum_{i=1}^{n} \lambda_i x_i \geqslant \sum_{i=1}^{n} \lambda_i \ln(x_i) \tag{10.2}$$

Cette propriété très intéressante va nous permettre d'obtenir une borne inférieure pour l'estimation du logarithme d'une somme. ce qui nous permettra de prouver la convergence de l'algorithme EM. Mais commençons par introduire cet algorithme.

10.1.2 Développement de la méthode

Maximum de vraisemblance avec données incomplètes :

Soit $\mathbf{D}$, un vecteur aléatoire issu d'un modèle paramétrique. Nous voulons trouver les paramètres $\widehat{\theta^{MV}}$ tels que $\mathbb{P}(\mathbf{D}|\widehat{\theta^{MV}})$ soit maximum. $\widehat{\theta^{MV}}$ est alors appelé l'*estimation du maximum de vraisemblance*. Pour estimer $\widehat{\theta^{MV}}$, nous devons donc introduire la fonction de log-vraisemblance qui est considérée comme une fonction de la variable θ définie par

$$L(\theta) = \ln\Big(\mathbb{P}(\mathbf{D}|\theta)\Big) \tag{10.3}$$

Comme la fonction ln est strictement croissante la valeur qui maximise $\mathbb{P}(\mathbf{D}|\theta)$ maximise également $L(\theta)$.

Soit $\mathbf{D} =< \mathbf{O}, \mathbf{H} >= (d_{li})_{m\times n}$ la base d'exemples, avec $\mathbf{O}$ et $\mathbf{H}$ représentant respectivement les parties complètes et manquantes de $\mathbf{D}$. Lorsque les données sont incomplètes, estimer $\widehat{\theta^{MV}}$ revient à maximiser l'intégrale sur les données manquantes de l'équation 10.4.

$$L(\theta) = \ln\Big(\sum_{\mathbf{H}} \mathbb{P}(\mathbf{O},\mathbf{H}|\theta)\Big) = \ln\Big(\sum_{\mathbf{H}} \mathbb{P}(\mathbf{O}|\mathbf{H},\theta)\mathbb{P}(\mathbf{H}|\theta)\Big) \tag{10.4}$$

L'évaluation d'une telle somme est alors exponentielle en le nombre de données manquantes.

Remarques préliminaires :

Si nous voulons améliorer une estimation θ_n par une nouvelle valeur θ telle que $L(\theta) \geqslant L(\theta_n)$. Nous voulons de plus maximiser la différence de l'équation 10.5 à chaque étape.

$$L(\theta) - L(\theta_n) = \ln \mathbb{P}(\mathbf{D}|\theta) - \ln \mathbb{P}(\mathbf{D}|\theta_n) \tag{10.5}$$

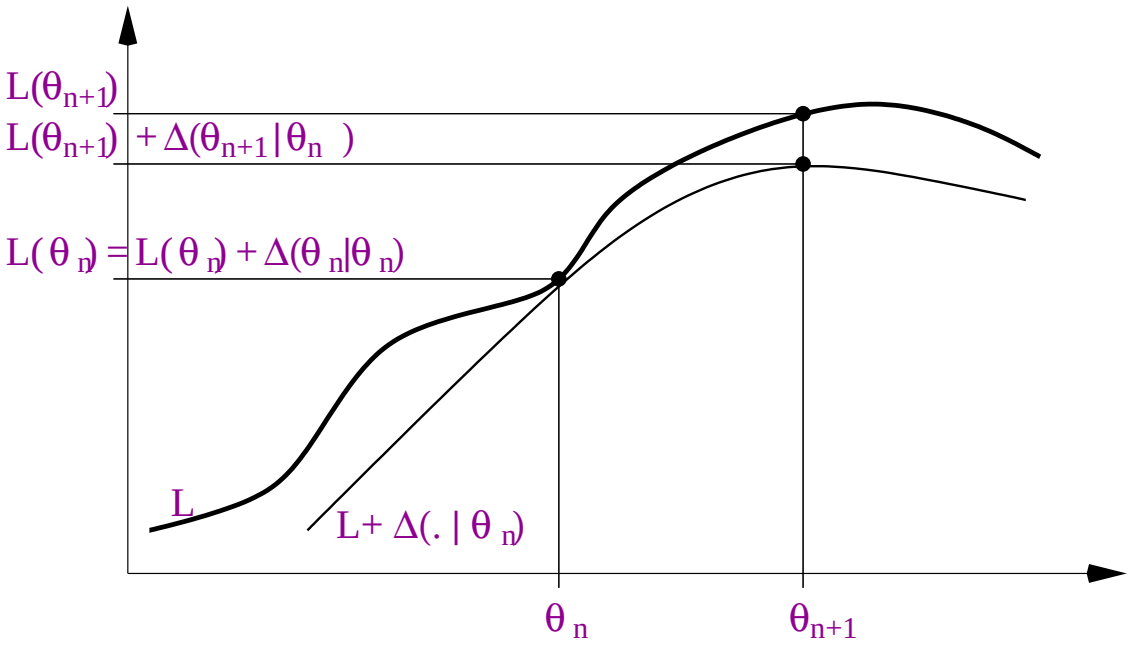


FIG. 10.1 : *Illustration d'une itération de l'algorithme EM.*

Le principal problème de cette technique est alors l'estimation de $\mathbb{P}(\mathbf{D}|\theta)$ lorsque certaines valeurs de $\mathbf{D}$ sont manquantes.
Nous pouvons réécrire l'équation 10.5 en utilisant l'équation 10.4.

$$L(\theta) - L(\theta_n) = \ln\left(\sum_{\mathbf{H}} \mathbb{P}(\mathbf{D}|\mathbf{H},\theta)\mathbb{P}(\mathbf{H}|\theta)\right) - \ln \mathbb{P}(\mathbf{D}|\theta_n) \tag{10.6}$$

Comme $\mathbb{P}(\mathbf{H}|\mathbf{D},\theta)$ est une mesure de probabilité, l'équation 10.2 va être applicable.

$$\begin{aligned} L(\theta) - L(\theta_n) &= \ln\left(\sum_{\mathbf{H}} \mathbb{P}(\mathbf{D}|\mathbf{H},\theta)\mathbb{P}(\mathbf{H}|\theta)\frac{\mathbb{P}(\mathbf{H}|\mathbf{D},\theta)}{\mathbb{P}(\mathbf{H}|\mathbf{D},\theta)}\right) - \ln \mathbb{P}(\mathbf{D}|\theta_n) \\ &\geqslant \sum_{\mathbf{H}} \left(\mathbb{P}(\mathbf{H}|\mathbf{D},\theta)\ln\left(\frac{\mathbb{P}(\mathbf{D}|\mathbf{H},\theta)\mathbb{P}(\mathbf{H}|\theta)}{\mathbb{P}(\mathbf{H}|\mathbf{D},\theta)}\right)\right) - \ln \mathbb{P}(\mathbf{D}|\theta_n) \\ &\geqslant \sum_{\mathbf{H}} \mathbb{P}(\mathbf{H}|\mathbf{D},\theta)\ln\left(\frac{\mathbb{P}(\mathbf{D}|\mathbf{H},\theta)\mathbb{P}(\mathbf{H}|\theta)}{\mathbb{P}(\mathbf{H}|\mathbf{D},\theta)\mathbb{P}(\mathbf{D}|\theta_n)}\right) \end{aligned}$$

Nommons cette dernière somme $\Delta(\theta|\theta_n)$ alors

$$L(\theta) \geqslant L(\theta_n) + \Delta(\theta|\theta_n) \tag{10.7}$$

De plus

$$\begin{aligned} \Delta(\theta_n|\theta_n) &= \sum_{\mathbf{H}} \mathbb{P}(\mathbf{H}|\mathbf{D},\theta_n)\ln\left(\frac{\mathbb{P}(\mathbf{D}|\mathbf{H},\theta_n)\mathbb{P}(\mathbf{H}|\theta_n)}{\mathbb{P}(\mathbf{H}|\mathbf{D},\theta_n)\mathbb{P}(\mathbf{D}|\theta_n)}\right) \\ &= \sum_{\mathbf{H}} \mathbb{P}(\mathbf{H}|\mathbf{D},\theta_n)\ln\left(\frac{\mathbb{P}(\mathbf{D},\mathbf{H}|\theta_n)}{\mathbb{P}(\mathbf{D},\mathbf{H}|\theta_n)}\right) \\ &= 0 \end{aligned}$$

Donc, pour toute valeur de θ et de θ_n, $L(\theta) \geqslant L(\theta_n) + \Delta(\theta|\theta_n)$ et $L(\theta_n) = L(\theta_n) + \Delta(\theta_n|\theta_n)$ et il est possible de représenter une itération de l'algorithme EM, développé dans le paragraphe suivant, par le schéma de la figure 10.1.

10.1.3 *Expectation-Maximisation*

La méthode EM est un algorithme itératif pour maximiser $L(\theta)$ de telle manière qu'à chaque itération nous obtenons une estimation θ_n meilleure que la précédente. Chaque itération contient deux étapes : celle d'Espérance et celle de Maximisation.

Nous venons de voir que le fait d'améliorer $L(\theta_n) + \Delta(\theta|\theta_n)$ améliore également $L(\theta)$, donc pour l'itération suivante nous allons choisir le θ qui maximise $L(\theta_n) + \Delta(\theta|\theta_n)$. Nous aurons alors l'assurance d'améliorer $L(\theta)$ (voir la figure 10.1).

$$\begin{aligned}
\theta_{n+1} &= arg\max_{\theta}\left\{L(\theta_n) + \Delta(\theta|\theta_n)\right\} \\
&= arg\max_{\theta}\left\{L(\theta_n) + \sum_{\mathbf{H}}\mathbb{P}(\mathbf{H}|\mathbf{D},\theta)\ln\left(\frac{\mathbb{P}(\mathbf{D}|\mathbf{H},\theta)\mathbb{P}(\mathbf{H}|\theta)}{\mathbb{P}(\mathbf{H}|\mathbf{D},\theta_n)\mathbb{P}(\mathbf{D}|\theta_n)}\right)\right\} \\
&= arg\max_{\theta}\left\{\sum_{\mathbf{H}}\mathbb{P}(\mathbf{H}|\mathbf{D},\theta_n)\ln\left(\mathbb{P}(\mathbf{D}|\mathbf{H},\theta)\mathbb{P}(\mathbf{H}|\theta)\right)\right\} \\
&= arg\max_{\theta}\left\{\sum_{\mathbf{H}}\mathbb{P}(\mathbf{H}|\mathbf{D},\theta_n)\ln\left(\frac{\mathbb{P}(\mathbf{D},\mathbf{H},\theta)}{\mathbb{P}(\mathbf{H},\theta)}\frac{\mathbb{P}(\mathbf{H},\theta)}{\mathbb{P}(\theta)}\right)\right\} \\
&= arg\max_{\theta}\left\{\sum_{\mathbf{H}}\mathbb{P}(\mathbf{H}|\mathbf{D},\theta_n)\ln\mathbb{P}(\mathbf{D},\mathbf{H}|\theta)\right\} \\
\theta_{n+1} &= arg\max_{\theta}\left\{\mathbb{E}_{\mathbf{H}|\mathbf{D},\theta_n}\left(\ln\mathbb{P}(\mathbf{D},\mathbf{H}|\theta)\right)\right\} \qquad (10.8)
\end{aligned}$$

Dans l'équation 10.8, les deux phases apparaissent. L'algorithme EM consiste donc en

1) *Étape E* : la détermination de l'espérance conditionnelle $\mathbb{E}_{\mathbf{H}|\mathbf{D},\theta_n}\left(\ln\mathbb{P}(\mathbf{D},\mathbf{H}|\theta)\right)$,

2) *Étape M* : la maximisation de cette expression pour θ.

Qu'avons-nous gagné de passer de la maximisation de $L(\theta)$ à celle de $L(\theta_n) + \Delta(\theta|\theta_n)$? Si nous avions eu à maximiser $L(\theta)$, nous aurions obtenu

$$\begin{aligned}
\widehat{\theta^{MV}} &= arg\max_{\theta}\left\{L(\theta)\right\} \\
&= arg\max_{\theta}\left\{\ln\left(\sum_{\mathbf{H}}\mathbb{P}(\mathbf{D}|\mathbf{H},\theta)\mathbb{P}(\mathbf{H}|\theta)\right)\right\} \\
\widehat{\theta^{MV}} &= arg\max_{\theta}\left\{\mathbb{E}_{\mathbf{H}|\theta}(\mathbb{P}(\mathbf{D}|\mathbf{H},\theta)\right\} \qquad (10.9)
\end{aligned}$$

Or dans l'équation 10.9, nous ne connaissons pas la loi $\mathbf{H}|\theta$, alors que si nous effectuons la maximisation de $L(\theta_n) + \Delta(\theta|\theta_n)$, la loi de $\mathbf{H}|\mathbf{D},\theta_n$ est connue et donc l'équation 10.8 est alors soluble.

L'algorithme EM donne également un cadre pour estimer les variables manquantes quand nous le désirons puisqu'à chaque itération il est possible d'en avoir une estimation considérant l'hypothèse courante sur les paramètres θ_n.

10.1.4 Convergence

Soit θ_{n+1} la configuration des paramètres qui maximisent la différence $\Delta(\theta|\theta_n)$.

Comme $\Delta(\theta_n|\theta_n) = 0$ et vu comment est choisie θ_{n+1}, nous avons $\Delta(\theta_{n+1}|\theta_n) \geqslant 0$. Donc, à chaque itération, la vraisemblance $L(\theta)$ ne décroît pas.

A l'itération n, la valeur θ_n maximise $L(\theta_n) + \Delta(\theta|\theta_n)$, et dans ce cas nous avons $L(\theta_n) + \Delta(\theta_n|\theta_n) = L(\theta_n)$. Les fonctions L et Δ sont différentiables en θ_n, cela signifie que θ_n est un point stationnaire de L. Ce point stationnaire peut alors être soit un optimal local, soit un point selle dans quelques rares situations.

10.2 Les adaptations de l'algorithme EM

Vu le succès de la méthode EM, de nombreuses implémentations et adaptations de cette méthode ont été proposées.

10.2.1 EM généralisé

Choisir θ_{n+1} en maximisant $\Delta(\theta|\theta_n)$ permet de maximiser l'augmentation de la vraisemblance $L(\theta)$ à chaque étape et donc de 'converger' plus rapidement.

Néanmoins, lorsqu'il est difficile d'obtenir le maximum de $\Delta(\theta|\theta_n)$, il est possible de se contenter de choisir θ_{n+1} tel que $\Delta(\theta_{n+1}|\theta_n) \geqslant \Delta(\theta_n|\theta_n)$. Dans ce cas, l'augmentation de la vraisemblance n'est pas optimisée à chaque itération mais la convergence vers un point de stationnarité existera toujours (même preuve que précédemment) et il devra alors être effectué plus d'itérations pour y arriver.

Cette méthode est connue comme étant l'algorithme *EM généralisé*. Même, si en théorie, cette méthode converge moins rapidement (en nombre d'itérations), en pratique elle peut être plus rapide (en temps de calcul). En effet, il est possible d'économiser bon nombre de calculs à l'étape de maximisation, puisqu'au lieu d'évaluer toutes les valeurs pour choisir la plus grande, il est possible de se restreindre à un sous-ensemble (par exemple un voisinage) ou simplement, prendre la première valeur qui augmente la vraisemblance pour l'itération suivante.

10.2.2 EM pour la classification

L'algorithme CEM, introduit par Celeux & Govaert (1992), est une adaptation de l'algorithme EM pour la classification. Celui-ci maximise alors la vraisemblance classifiante (section 7.4) plutôt que la vraisemblance.

10.2.3 EM incrémental

L'inconvénient de l'algorithme EM est qu'il doit être effectué hors-ligne. Or si de nouveaux exemples deviennent disponibles, il devient nécessaire de réeffectuer le calcul.

Cohen, Bronstein & Cozman (2001) ont introduit une méthode basée sur l'algorithme EM appelée *Voting EM* permettant un apprentissage de paramètres de réseaux bayésiens de manière incrémentale.

10.2.4 Monte-Carlo EM

Lorsque la méthode EM demande des calculs d'intégrales qui sont parfois insurmontables (ou de sommes contenant un nombre exponentiel de termes dans le cas discret ici

présenté), il est possible d'utiliser des méthodes de Monte-Carlo pour leur évaluation. La méthode *Monte-Carlo Expectation Maximization* (MCEM) introduite par Wei & Tanner (1990) est une extension de la méthode EM qui utilise une méthode de Monte-Carlo pour évaluer ces intégrales/sommes. En pratique, il faut que la base soit suffisamment grande pour que l'estimation soit fiable.

10.2.5 EM variationnel

Comme dans le cas précédent, lorsque l'évaluation des intégrales de l'algorithme EM est difficile, il est possible d'utiliser des méthodes variationnelles pour effectuer une estimation. Nous sommes alors en présence d'un algorithme dit de EM variationnel comme l'on introduit Neal & Hinton (1998) et Beal & Ghahramani (2003).

11

Fonction de scores avec données incomplètes

"Les règles des probabilités sont en défaut lorsqu'elles proposent, pour trouver l'enjeu, de multiplier la somme espérée par la probabilité du cas qui doit faire gagner cette somme."

Jean le Rond d'Alembert (1717-1783)

Sommaire

11.1 Adapter les scores pour les bases incomplètes 130
Utilisation des exemples complets 130
Utilisation des exemples disponibles 130
Utilisation des méthodes de remplacement 130
11.2 Approximation d'un score avec des bases incomplètes 131
11.2.1 L'approximation de *Cheeseman et Stutz* 131
Approximation de Laplace de *Cheeseman et Stutz* . . 131
Approximation BIC-MAP de *Cheeseman et Stutz* . . . 131
11.2.2 Méthode d'évaluation générique de score 132
11.2.2.1 La méthode 132
11.2.2.2 Exemple avec le score *BIC* 132
11.2.2.3 Exemple avec le score *BD* 133

11.1 Adaptation des scores pour des bases incomplètes

Soit $V = \{X_1, \cdots, X_n\}$ un ensemble des variables aléatoires, $\mathbf{D}_c$ une base de m tirages de V indépendants et identiquement distribués.

Supposons par ailleurs que l'on ne possède alors qu'une version incomplète $\mathbf{D}$ de la base $\mathbf{D}_c$, celle-ci peut se décomposer en

$$\mathbf{D} = [[\mathbf{X}_i^l]]_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant l \leqslant m}} = [\mathbf{O}, \mathbf{H}]$$

ou $\mathbf{O}$ est l'ensemble des variables $\mathbf{X}_i^l$ observées et $\mathbf{H}$ l'ensemble des variables $\mathbf{X}_i^l$ cachées.

Ici nous voudrions évaluer le score bayésien qui est défini par $BD(\mathcal{G}, \mathbf{D}) = \mathbb{P}(\mathcal{G}, \mathbf{D}) = \mathbb{P}(\mathcal{G})\mathbb{P}(\mathbf{D}|\mathcal{G})$ à partir de bases d'exemples incomplètes. En présence de données incomplètes, nous avons

$$\mathbb{P}(\mathbf{D}^l|\mathcal{G}) = \mathbb{P}(\mathbf{O}^l|\mathcal{G}) = \sum_{\mathbf{H}^l} \mathbb{P}(\mathbf{O}^l, \mathbf{H}^l|\mathcal{G})$$

pour le l-ième exemple de la base. En considérant que les exemples de la base $\mathbf{D}$ sont *i.i.d*, nous obtenons

$$\mathbb{P}(\mathbf{D}|\mathcal{G}, \Theta) = \prod_{l=1}^{m} \Big(\sum_{\mathbf{H}^l} \mathbb{P}(\mathbf{O}^l, \mathbf{H}^l|\mathcal{G}, \Theta) \Big) \tag{11.1}$$

Le nombre de terme de l'équation 11.1 croît exponentiellement avec le nombre de variables non observées. La complexité de l'évaluation de cette probabilité est donc exponentielle par rapport au nombre de valeurs manquantes dans la base d'exemples. En pratique, ceci n'est donc pas utilisable, on va donc devoir avoir recours à une méthode d'approximation.

Utilisation des exemples complets :

Une première approximation de cette intégrale serait donc de dire qu'elle ne doit pas être éloignée de celle de l'équation 7.6 évaluée seulement sur les exemples complets de la base. Cette approximation n'est seulement justifiée que quand le pourcentage de données incomplètes est faible.

Utilisation des exemples disponibles :

Une seconde approximation simple de l'équation 11.1 serait de dire qu'il s'agit de n'utiliser que les exemples disponibles pour calculer chaque terme. En pratique, cela revient donc à utiliser une équation identique à celle de l'équation 7.6, mais où les N_{ijk} seront évalués sur la base d'exemples incomplète de la manière suivante : pour calculer les N_{ijk}, les exemples de la base où X_i et $Pa(X_i)$ sont complètement observés sont conservés pour effectuer le comptage.

Utilisation des méthodes de remplacement :

De manière analogue, il est aisé d'imaginer d'effectuer un remplacement par la valeur médiane ou le mode (et non moyenne dans le cas discret) et de faire le comptage ensuite. Nous nous apercevons alors immédiatement que cette méthode va fortement biaiser les résultats de comptage en faveur de cette valeur.

Bien sur une méthode de substitution plus avancée peut être utilisée également. Dans ce cas, le biais sera moins important (penser aux méthodes *hot deck imputation* ou d'imputation par régression).

Observons à présent d'autres techniques *moins naïves* pour évaluer cette intégrale.

11.2 Approximation d'un score avec des bases incomplètes

11.2.1 L'approximation de *Cheeseman et Stutz*

Cette méthode d'approximation a été introduite par Cheeseman & Stutz (1996). Elle consiste en l'utilisation d'une complétion $\mathbf{D}_c$ de la base incomplète $\mathbf{D}$. Il est alors toujours possible d'écrire l'équation suivante.

$$\mathbb{P}(\mathbf{D}|\mathcal{G}) = \mathbb{P}(\mathbf{D}_c|\mathcal{G})\frac{\mathbb{P}(\mathbf{D}|\mathcal{G})}{\mathbb{P}(\mathbf{D}_c|\mathcal{G})} \tag{11.2}$$

puis, il vient

$$\mathbb{P}(\mathbf{D}|\mathcal{G}) = \mathbb{P}(\mathbf{D}_c|\mathcal{G})\frac{\int \mathbb{P}(\mathbf{D},\Theta|\mathcal{G})d\Theta}{\int \mathbb{P}(\mathbf{D}_c,\Theta|\mathcal{G})d\Theta} \tag{11.3}$$

Il reste alors à évaluer l'intégrale $\int \mathbb{P}(\mathbf{D},\Theta|\mathcal{G})d\Theta$. Celle-ci peut alors être approchée par une méthode de maximum *a posteriori*.

Approximation de Laplace de *Cheeseman et Stutz* :

En utilisant la formule 7.16, nous pouvons donner l'approximation suivante pour la formule de Cheeseman et Stutz.

$$\begin{aligned}\ln(\mathbb{P}(\mathbf{D}|\mathcal{G})) \simeq \ln(\mathbb{P}(\mathbf{D}_c|\mathcal{G})) \quad & -\ln(\mathbb{P}(\mathbf{D}_c,\widehat{\theta'^{MAP}}|\mathcal{G})) + \frac{1}{2}\ln(|A'|) \\ & +\ln(\mathbb{P}(\mathbf{D},\widehat{\theta^{MAP}}|\mathcal{G})) - \frac{1}{2}\ln(|A|)\end{aligned} \tag{11.4}$$

Les valeurs θ'^{MAP} et A' sont évaluées sur la base d'exemples complète $\mathbf{D}_c$ tandis que les valeurs θ^{MAP} et A sont évaluées sur la base d'exemples incomplète $\mathbf{D}$

Approximation BIC-MAP de la formule de *Cheeseman et Stutz* :

Considérons la base $\mathbf{D}_c$ qui est une complétion de $\mathbf{D}$ telle que les statistiques suffisantes N'_{ijk} de $\mathbf{D}_c$ sont égales aux statistiques suffisantes N_{ijk} de $\mathbf{D}$ calculées par maximum *a posteriori*. Soient $\widehat{\Phi}$ les paramètres naturels évalués par maximum *a posteriori* sur $\mathbf{D}$ alors d'après la formule 7.21, nous avons

$$\begin{aligned}\ln\mathbb{P}(\mathbf{D}|\mathcal{G}) \simeq \ln\mathbb{P}(\mathbf{D}_c|\mathcal{G}) \quad & -\ln\mathbb{P}(\mathbf{D}_c|\mathcal{G},\widehat{\Phi}') + \frac{dim(\mathcal{G}|\mathbf{D}_c)}{2}\ln N \\ & +\ln\mathbb{P}(\mathbf{D}|\mathcal{G},\widehat{\Phi}) - \frac{dim(\mathcal{G}|\mathbf{D})}{2}\ln N\end{aligned} \tag{11.5}$$

Dans ce cas, Geiger & Heckerman (1996) ont montré que $dim(\mathcal{G}|\mathbf{D}) = dim(\mathcal{G}|\mathbf{D}')$. De plus, comme les statistiques essentielles des deux bases d'exemples sont égales nous avons $\widehat{\Phi}' = \widehat{\Phi}$ donc

$$\ln\mathbb{P}(\mathbf{D}|\mathcal{G}) \simeq \ln\mathbb{P}(\mathbf{D}_c|\mathcal{G}) - \ln\mathbb{P}(\mathbf{D}_c|\mathcal{G},\widehat{\Phi}) + \ln\mathbb{P}(\mathbf{D}|\mathcal{G},\widehat{\Phi}) \tag{11.6}$$

Cette méthode permet donc d'évaluer $\mathbb{P}(\mathbf{D}|\mathcal{G})$ à partir d'une complétion de la base de cas de manière simple : il suffit juste de s'assurer que les statistiques essentielles de la base incomplète soient conservées dans la base complète.

11.2.2 Méthode d'évaluation générique de score à partir d'une base d'exemples incomplète

Pour nos expérimentations, nous utiliserons des scores du type

$$\ln \mathbb{P}(\mathbf{D}|\mathcal{G}) \simeq BIC(\mathcal{B}, \mathbf{D}) = \ln \mathbb{P}(\mathbf{D}|\mathcal{G}, \widehat{\Theta^{MAP}}) - \frac{1}{2} Dim(\mathcal{B}) \log N \tag{11.7}$$

Il reste cependant à voir comment calculer le terme $\mathbb{P}(\mathbf{D}|\mathcal{G}, \widehat{\Theta^{MAP}})$ lorsque la base $\mathbf{D}$ est incomplète. Donnons à présent une méthode générique permettant de calculer un score à partir d'une base d'exemples incomplète.

11.2.2.1 La méthode

Nous avons vu comment évaluer les paramètres d'un réseau bayésien à partir d'une base d'exemples incomplète en section 5. Nous allons maintenant voir comment faire de même pour un critère de score.

Soit $S(\mathcal{M}|\mathbf{D}_c)$ une fonction de score pour un modèle $\mathcal{M}$ en fonction d'une base complète $\mathbf{D}_c$. Alors, il est possible de considérer le score de ce même modèle avec une base de données incomplète $\mathbf{D}$.

$$Q^S(\mathcal{M}|\mathbf{D}) = \mathbb{E}_{\mathbf{H}\sim\mathbb{P}(\mathbf{H})}(S(\mathcal{M}|\mathbf{O}, \mathbf{H})) \tag{11.8}$$

Or, nous n'avons pas accès à la loi $\mathbb{P}(\mathbf{H})$. Il faut donc l'approcher à partir d'un modèle de représentation de $\mathbf{D}$.

Supposons, à présent, la donnée d'un modèle $\mathcal{M}^0$ supposé générateur de $\mathbf{D}$, alors il est possible de faire l'approximation suivante

$$Q^S(\mathcal{M}|\mathbf{D}) \approx Q^S(\mathcal{M} : \mathcal{M}^0|\mathbf{D}) = \mathbb{E}_{\mathbf{H}\sim\mathbb{P}(\mathbf{H}|\mathcal{M}^0)}(S(\mathcal{M}|\mathbf{O}, \mathbf{H}))$$

c'est-à-dire

$$Q^S(\mathcal{M} : \mathcal{M}^0|\mathbf{D}) = \sum_{\mathbf{H}} S(\mathcal{M}|\mathbf{O}, \mathbf{H}) \mathbb{P}(\mathbf{H}|\mathcal{M}^0) \tag{11.9}$$

Or, maintenant, nous avons accès à $\mathbb{P}(\mathbf{H}|\mathcal{M}^0)$ puisque $\mathcal{M}^0$ est fixé.

Cette méthode nous permet, à partir d'une fonction de score $S(\mathcal{M}|\mathbf{D}_c)$ quelconque, de créer une fonction de score $Q^S(\mathcal{M} : \mathcal{M}^0|\mathbf{D})$ qui donne un résultat (approché car un modèle est rarement exact) sur des bases d'exemples incomplètes. Dans les chapitres suivantes, la base d'exemples sera implicite et la notation simplifiée en $Q^S(\mathcal{M} : \mathcal{M}^0)$.

Par ailleurs, ce score possède la particularité de conserver les propriétés de *décomposabilité* (linéarité de l'espérance) et de *score équivalence* du score S.

Le grand avantage de cette méthode est qu'elle s'incorpore parfaitement bien dans un algorithme EM pour lequel l'évaluation du score est de plus en plus fine au fur et à mesure que le modèle est précis.

Regardons à présent comment se décline le score BIC rappelé en équation 11.7 pour une base d'exemples incomplète.

11.2.2.2 Exemple avec le score BIC

Il est possible d'adapter le score de l'équation 11.7 aux bases d'exemples incomplètes comme décrit dans la section 11.2.2.1, ce qui donne

$$Q^{BIC}(\mathcal{B} : \mathcal{B}^0|\mathbf{O}, \mathbf{H}) = \mathbb{E}_{\mathbf{H}\sim\mathbb{P}(\mathbf{H}|\mathcal{G}^0,\theta^0)}\big(BIC(\mathcal{B}, \mathbf{O}, \mathbf{H}) \mathbb{P}(\mathbf{H}|\mathcal{G}^0, \theta^0)\big) \tag{11.10}$$

Or le score BIC est décomposable donc le score Q^{BIC} également (le score local bic est défini dans l'équation 7.20).

$$Q^{BIC}(\mathcal{B} : \mathcal{B}^0|\mathbf{O},\mathbf{H}) = \mathbb{E}_{\mathbf{H}\sim\mathbb{P}(\mathbf{H}|\mathcal{G}^0,\theta^0)} \sum_{i=1}^{n} \left(bic(X_i, Pa(X_i),\mathbf{O},\mathbf{H})\mathbb{P}(\mathbf{H}|\mathcal{G}^0,\theta^0)\right)$$

Par la linéarité de l'espérance, nous obtennons

$$Q^{BIC}(\mathcal{B} : \mathcal{B}^0|\mathbf{O},\mathbf{H}) = \sum_{i=1}^{n} q^{bic}(X_i, Pa(X_i) : \mathcal{G}^0,\theta^0|\mathbf{O},\mathbf{H})$$

avec

$$q_i^{bic}(X_i, Pa(X_i) : \mathcal{G}^0,\theta^0|\mathbf{O},\mathbf{H}) = \mathbb{E}_{\mathbf{H}\sim\mathbb{P}(\mathbf{H}|\mathcal{G}^0,\theta^0)}\left[bic(X_i, Pa(X_i),\mathbf{O},\mathbf{H})\mathbb{P}(\mathbf{H}|\mathcal{G}^0,\theta^0)\right]$$

En utilisant les propriétés de l'espérance on trouve donc

$$q_i^{bic}(X_i, Pa(X_i) : \mathcal{G}^0,\theta^0|\mathbf{D}) = \mathbb{E}_{\mathbf{H}\sim\mathbb{P}(\mathbf{H}|\mathcal{G}^0,\theta^0)}\Big[\log\left(\mathbb{P}\left(\mathbf{D}_{|Pa(X_i)\cup\{X_i\}}| < X_i, Pa(X_i) >, \widehat{\theta^{MV}_{X_i|Pa(X_i)}}\right)\right) - \frac{Dim(X_i, Pa(X_i))}{2}\log N\Big]$$

$$q_i^{bic}(X_i, Pa(X_i) : \mathcal{G}^0,\theta^0|\mathbf{D}) = \sum_{X_i}\sum_{Pa(X_i)} \mathbb{E}_{\mathbf{H}\sim\mathbb{P}(\mathbf{H}|\mathcal{G}^0,\theta^0)}\left[N_{X_i,Pa(X_i)}\right]\cdot\log(\widehat{\theta^{MV}_{X_i|Pa(X_i)}}) - \frac{Dim(X_i, Pa(X_i))}{2}\log N \tag{11.11}$$

Nous avons ici l'expression locale du score BIC adaptée aux bases d'exemples incomplètes.

En pratique, si l'on utilise un algorithme EM, il est possible de profiter des boucles pour mettre à jour les statistiques essentielles $N^*_{X_i,Pa(X_i)} = \mathbb{E}_{\mathbf{H}\sim\mathbb{P}(\mathbf{H}|\mathcal{G}^0,\theta^0)}[N_{X_i,Pa(X_i)}]$.

11.2.2.3 Exemple avec le score BD

Le score BD est défini par l'équation 7.9 et donne $\mathbb{P}(\mathbf{D}|\mathcal{G}^0)$. Nous voulons prendre l'espérance du score bayésien pour construire une version de ce score à partir de bases incomplètes.

$$Q(\mathcal{G} : \mathcal{G}^0) = \mathbb{E}_{\mathbf{H}\sim\mathbb{P}(\mathbf{H}|\mathbf{O},\mathcal{G}^0)}\left[\mathbb{P}(\mathbf{H},\mathbf{O},\mathcal{G})\right] \tag{11.12}$$

Ce qui est défini par

$$Q(\mathcal{G} : \mathcal{G}^0) = \sum_{\mathbf{H}} \mathbb{P}(\mathbf{H}|\mathbf{O},\mathcal{G}^0)\mathbb{P}(\mathbf{H},\mathbf{O},\mathcal{G}) \tag{11.13}$$

où $\mathbb{P}(\mathbf{H},\mathbf{O},\mathcal{G})$ est donné par la formule 7.6 page 78 et où $\mathbb{P}(\mathbf{H}|\mathbf{O},\mathcal{G}^0)$ peut être approché par évalué par $\mathbb{P}(\mathbf{H}|\mathcal{G}^0,\theta^{MV})$, avec θ^{MV} les paramètres obtenus par maximum de vraisemblance pour la structure $\mathcal{G}^0$ et la base d'exemples incomplète $\mathbf{O}$. Typiquement, cette évaluation est faite grâce à l'algorithme EM paramétrique de la page 57.

Nous obtenons donc l'approximation du score bayésien à partir d'une base d'exemples incomplète de l'équation 11.14.

$$Q(\mathcal{G} : \mathcal{G}^0) = \sum_{\mathbf{H}} \mathbb{P}(\mathbf{H}|\mathcal{G}^0,\widehat{\theta^{0^{MV}}})\mathbb{P}(\mathbf{H},\mathbf{O},\mathcal{G}) \tag{11.14}$$

Observons maintenant comment utiliser cette méthode de calcul de score pour mettre en oeuvre une méthode d'identification de structure de réseaux bayésiens à partir d'une base de cas incomplète.

12

Méthodes à base de score

"Le hasard : l'enchaînement des effets dont nous ignorons les causes."

Claude Adrien Helvétius (1715-1771)

Sommaire

12.1 Méthodes existantes 136
12.1.1 *Structural Expectation-Maximisation* 136
12.1.1.1 AMS-EM : *alternative model selection* 136
Principe 136
12.1.1.2 Bayesian Structural EM 137
12.1.2 Utilisation des MCMC 138
12.1.3 *Hybrid Independence Test* 138
The Robust Bayesian Estimator 138
12.2 Méthodes proposées 138
12.2.1 MWST-EM 138
12.2.2 Variantes pour les problèmes de classification 140
12.2.2.1 Classifieur de Bayes naïf augmenté par un arbre . 140
12.2.2.2 Classifieur naïf augmenté par une forêt : FAN-EM 140
12.2.2.3 Travaux connexes 141
12.2.3 Une proposition d'initialisation pour SEM 141
12.2.4 Passer à l'espace des équivalents de Markov 142

12.1 Méthodes existantes

12.1.1 *Structural Expectation-Maximisation*

Les principes de l'algorithme EM, que nous avons utilisés pour l'apprentissage des paramètres d'un réseau bayésien en section 5.3.2, peuvent également être utilisés pour l'apprentissage de structure. Néanmoins, comme nous allons le voir par la suite, la méthode va devoir être adaptée.

12.1.1.1 AMS-EM : *alternative model selection*

Les premiers travaux qui ont utilisé une approche de type EM pour effectuer de l'apprentissage de structure sont dûs à Friedman (1997). Détaillons la méthode AMS-EM proposée.

Principe :

Une adaptation simple de l'algorithme EM reviendrait à utiliser une méthode similaire à celle décrite dans l'algorithme 3.

ALGORITHME 3 : EM générique pour l'apprentissage de structure

1: **Init :** $i = 0$
   Choix aléatoire ou utilisant une heuristique pour sélectionner le réseau bayésien initial $(\mathcal{G}^0, \Theta^0)$
2: **Répéter**
3: $\quad i = i + 1$
4: $\quad (\mathcal{G}^i, \Theta^i) = \underset{\mathcal{G},\Theta}{\operatorname{argmax}} \; Q(\mathcal{G}, \Theta : \mathcal{G}^{i-1}, \Theta^{i-1})$
5: **Jusqu'à** $|Q(\mathcal{G}^i, \Theta^i : \mathcal{G}^{i-1}, \Theta^{i-1}) - Q(\mathcal{G}^{i-1}, \Theta^{i-1} : \mathcal{G}^{i-1}, \Theta^{i-1})| \leqslant \epsilon$

L'étape de maximisation de l'algorithme (étape 4) devrait être menée dans l'espace produit des structures-paramètres $\{\mathcal{G}, \Theta\}$. Cela revient donc à trouver le meilleur réseau bayésien (la meilleure structure *et* les meilleurs paramètres). En pratique, cela n'est pas possible et deux étapes distinctes doivent être faites. Une dans l'espace des DAG (équation 12.1)[i], puis une dans l'espace des paramètres (équation 12.2).

$$\mathcal{G}^i = \underset{\mathcal{G}}{\operatorname{argmax}} \; Q(\mathcal{G}, \bullet : \mathcal{G}^{i-1}, \Theta^{i-1}) \tag{12.1}$$

$$\Theta^i = \underset{\Theta}{\operatorname{argmax}} \; Q(\mathcal{G}^i, \Theta : \mathcal{G}^{i-1}, \Theta^{i-1}) \tag{12.2}$$

où $Q(\mathcal{G}, \Theta : \mathcal{G}^*, \Theta^*)$ est l'espérance du score d'un réseau bayésien $< \mathcal{G}, \Theta >$ obtenue en utilisant la distribution des données $P(\mathbf{D}|\mathcal{G}^*, \Theta^*)$.

Remarquons que la maximisation de l'équation 12.1 dans l'espace des DAG nous conduit à nouveau au problème initial : trouver la meilleure structure dans un espace de taille *super-exponentielle*. Néanmoins, grâce à l'algorithme EM *généralisé* (voir section 10.2.1), nous savons qu'il est possible de chercher une meilleure solution que la solution actuelle plutôt que de choisir la meilleure dans l'espace tout entier, sans affecter les propriétés de convergence de l'algorithme. Friedman (1997) propose alors de chercher le

[i] La notation $Q(\mathcal{G}, \bullet : \dots)$ utilisée dans l'équation 12.1 signifie $\mathbb{E}_\Theta[Q(\mathcal{G}, \Theta : \dots|\mathbf{D})]$ pour une approche bayésienne ou $Q(\mathcal{G}, \widehat{\Theta^{MV}} : \dots|\mathbf{D})$ pour une approche par maximum de vraisemblance, nous ometrons le $\mathbf{D}$ pour ne pas alourdir les notations.

ALGORITHME 4 : EM détaillé pour l'apprentissage de structure

1: **Init :** $fini = faux$, $i = 0$
Choix aléatoire ou utilisant une heuristique pour sélectionner le réseau bayésien initial $(\mathcal{G}^0, \Theta^0)$
2: **Répéter**
3: $j = 0$
4: **Répéter**
5: $\Theta^{i,j+1} = \underset{\Theta}{\text{argmax}} \; Q(\mathcal{G}^i, \Theta : \mathcal{G}^i, \Theta^{i,j})$
6: $j = j + 1$
7: **Jusqu'à** convergence $(\Theta^{i,j} \rightarrow \Theta^{i,j^o})$
8: **Si** $i = 0$ ou $|Q(\mathcal{G}^i, \Theta^{i,j^o} : \mathcal{G}^{i-1}, \Theta^{i-1,j^o}) - Q(\mathcal{G}^{i-1}, \Theta^{i-1,j^o} : \mathcal{G}^{i-1}, \Theta^{i-1,j^o})| > \epsilon$ **Alors**
9: $\mathcal{G}^{i+1} = \underset{\mathcal{G} \in \mathcal{V}_{\mathcal{G}^i}}{\text{argmax}} \; Q(\mathcal{G}, \bullet : \mathcal{G}^i, \Theta^{i,j^o})$
10: $\Theta^{i+1,0} = \underset{\Theta}{\text{argmax}} \; Q(\mathcal{G}^{i+1}, \Theta : \mathcal{G}^i, \Theta^{i,j^o})$
11: $i = i + 1$
12: **Sinon**
13: $fini = vrai$
14: **Fin Si**
15: **Jusqu'à** $fini = vrai$

graphe pour l'itération suivante dans un voisinage $\mathcal{V}_{\mathcal{G}}$ défini comme étant l'ensemble des DAG qui diffèrent de la structure courante $\mathcal{G}$ par l'*ajout*, le *retrait* ou l'*inversion* d'un arc (*cf.* section 8.2.1).

Pour la recherche dans l'espace des paramètres de l'équation 12.2, Friedman (1997) propose de répéter l'opération plusieures fois avec des initialisations *intelligentes*. Cette étape revient alors à utiliser l'algorithme EM paramétrique (algorithme 2, page 57). Cet algorithme représente alors les étapes 4 à 7 de l'algorithme 4. Une itération de l'algorithme EM paramétrique est alors effectuée en utilisant l'initialisation *intelligente* donnée par l'étape 10 de l'algorithme 4. Dans cette étape, les paramètres optimaux de l'itération précédente sont utilisés pour initialiser les paramètres de la structure maximisant le score.

La fonction de score utilisée est une fonction du type espérance d'un score (*BIC* ou *MDL*) tel qu'il est décrit en section 11.2.2.

12.1.1.2 Bayesian Structural EM

Friedman (1998) propose ensuite une version bayésienne de son algorithme AMS-EM qu'il nomme BS-EM pour *Bayesian Structural EM*. Cette méthode optimise le score bayésien plutôt qu'un score asymptotiquement équivalent (de type *BIC*/*MDL*). La fonction de score utilisée est celle de la formule 11.13.

Les méthodes AMS-EM et BS-EM peuvent être considérées comme des recherches gloutonnes utilisant un algorithme EM paramétrique à chaque itération. Il s'agit donc d'une méthode itérative, contenant une autre méthode itérativue à l'intérieur de chacune de ses itérations. Le temps de calcul de telles méthodes est donc forcement assez élevé.

Par la suite nous regrouperons les méthodes AMS-EM et BS-EM sous le sigle SEM pour *Structural-EM*.

12.1.2 Utilisation des MCMC

Myers, Laskey & Lewitt (1999) ont introduit une heuristique nommée EMCMC pour *evolutionary Markov Chain of Monte Carlo* et l'utilisent pour effectuer de l'apprentissage de structure à partir de données incomplètes. Leur technique utilise les avantages des algorithmes évolutionnaires pour effectuer du croisement et de la mutation des modèles contenus dans une population, et ceux de l'algorithme de Metropolis-Hasting pour générer cette population. La base d'exemples est alors remplie et les modèles évalués à l'aide de la mesure BDe (*cf.* section 7.2.2).

12.1.3 *Hybrid Independence Test*

Dash & Druzdzel (2003) introduisent une méthode qu'ils nomment PC*, dont la procédure de recherche de structure est la même que celle de l'algorithme PC (section 6.3). Par contre, les tests statistiques sont basés sur une méthode originale nommée *Hybrid Independence Test* qui évalue les statistiques essentielles de la base d'exemples et qui utilise alors ces statistiques essentielles pour effectuer les tests statistiques dont a besoin une méthode comme PC.

Pour calculer les statistiques essentielles $\widehat{N}_{X,Y|\{Z_1,\cdots,Z_n\}}$, un 'petit' réseau bayésien où tous les nœuds de $\{X, Z_1, \cdots, Z_n\}$ sont connectés vers le nœud Y est construit. Puis un apprentissage des paramètres est effectué pour ce petit réseau bayésien, par exemple en utilisant des MCMC ou à l'aide de l'algorithme EM paramétrique quand la base est incomplète, il est donc possible de considérer des *a priori* de Dirichlet.

Ces statistiques essentielles vont ensuite servir pour tester si $X \perp\!\!\!\perp Y|\{Z_1, \cdots, Z_n\}$ à l'aide d'une technique de type χ^2. Comme souvent l'ordre des tests statistiques est restreint, alors le réseau bayésien à apprendre ne contient pas beaucoup de nœuds, et l'apprentissage par une méthode itérative du type EM peut être effectué en un temps raisonnable.

The Robust Bayesian Estimator Une méthode d'évaluation des paramètres d'un réseau bayésien à partir de bases d'exemples incomplètes a été introduite par Ramoni & Sebastiani (2000) et Sebastiani & Ramoni (2001). Cette méthode pourrait être utilisée dans le cadre de l'algorithme PC* car elle permet une évaluation des paramètres plus robuste que les approches classiques (EM, MCMC) lorsque les données manquantes suivent des processus NMAR. Durant l'apprentissage, les bornes inférieures et supérieures des paramètres sont conservées pour permettre d'obtenir une valeur plus fidèle lors de l'évaluation finale.

12.2 Méthodes proposées

12.2.1 MWST-EM

Pour cette méthode, nous proposons de rechercher la structure arborescente optimale reliant toutes les variables (François & Leray (2005b) et Leray & François (2005)). Dans ce cas, la première étape de l'algorithme 4 ne change pas. Le choix d'une chaîne

ALGORITHME 5 : MWST-EM

1: **Init :** $fini = faux$, $i = 0$
Choix aléatoire ou utilisant une heuristique pour sélectionner le réseau bayésien initial $(\mathcal{T}^0, \Theta^0)$
2: **Répéter**
3: $j = 0$
4: **Répéter**
5: $\Theta^{i,j+1} = \underset{\Theta}{\text{argmax}} \; Q(\mathcal{T}^i, \Theta : \mathcal{T}^i, \Theta^{i,j})$
6: $j = j + 1$
7: **Jusqu'à** convergence $(\Theta^{i,j} \rightarrow \Theta^{i,j^o})$
8: **Si** $i = 0$ ou $|Q(\mathcal{T}^i, \Theta^{i,j^o} : \mathcal{T}^{i-1}, \Theta^{i-1,j^o}) - Q(\mathcal{T}^{i-1}, \Theta^{i-1,j^o} : \mathcal{T}^{i-1}, \Theta^{i-1,j^o})| > \epsilon$ **Alors**
9: $\mathcal{T}^{i+1} = \underset{\mathcal{T}}{\text{argmax}} \; Q(\mathcal{T}, \bullet : \mathcal{T}^i, \Theta^{i,j^o})$
10: $\Theta^{i+1,0} = \underset{\Theta}{\text{argmax}} \; Q(\mathcal{T}^{i+1}, \Theta : \mathcal{T}^i, \Theta^{i,j^o})$
11: $i = i + 1$
12: **Sinon**
13: $fini = vrai$
14: **Fin Si**
15: **Jusqu'à** $fini = vrai$

orientée reliant toutes les variables comme le propose Friedman (1997) semble encore plus adaptée puisqu'une chaîne est un cas particulier de structure arborescente.

Le changement principal intervient à l'étape 9 de l'algorithme 5. Nous ne proposons plus de choisir un meilleur graphe dans un voisinage, mais la structure arborescente optimale. Nous n'utilisons plus un algorithme EM généralisé dans l'espace des DAG, mais bien un algorithme EM dans l'espace des structures arborescentes. Comme nous travaillons alors dans l'espace complet à chaque itération, la méthode résultante converge plus rapidement vis-à-vis du nombre d'itérations.

De manière analogue aux cas des base d'exemples complètes (section 8.1.1), pour trouver la structure qui maximise la fonction de score Q, nous construisons la matrice des variations du score BIC lorsqu'un nœud est choisi comme parent d'un autre ou non, avec comme référence le modèle de l'itération courante $< \mathcal{T}^i, \Theta^i >$ pour avoir accès à la distribution des données manquantes $\mathbb{P}(\mathbf{H}|\mathcal{T}^i, \Theta^i)$. Cette matrice (symétrique avec des $-\infty$ sur la diagonale) est donnée par l'équation 12.3.

$$\left[M_{ij}\right]_{1 \leqslant i < j \leqslant n} = \left[q_i^{bic}(X_i, Pa(X_i) = X_j, \Theta_{Xi|X_j}) - q_i^{bic}(X_i, Pa(X_i) = \emptyset, \Theta_{Xi})\right] \tag{12.3}$$

où le score local q_i^{bic} est défini par l'équation 11.11.

Nous utilisons alors un algorithme du type de celui de Kruskal (1956) (voir l'algorithme 6, il serait également possible d'utiliser un algorithme similaire proposé indépendamment par Prim, Dijkstra où Jarník (1930)) pour trouver l'arbre de poids maximal à partir de cette matrice. Il ne reste plus qu'à choisir une racine pour orienter cet arbre en effectuant un parcours en profondeur pour obtenir la structure arborescente que nous recherchons.

La symétrie de la matrice de scores est due au choix d'un score équivalent (voir le théorème 3.4.1) étant donnée qu'il n'est pas possible qu'une structure arborescente contienne de V-structure.

ALGORITHME 6 : méthode de Kruskal

Classer les arêtes par ordre de mesure décroissante $(u_1, u_2, u_3, ..., u_l)$
$T = \{\}$, $i = 1$
Tant que $|T| < N - 1$ **Faire**
  Si $T \cup u_i$ est sans cycle **Alors**
    ajouter u_i à T.
  Sinon
    $i = i + 1$
  Fin Si
Fin Tant que

12.2.2 Variantes pour les problèmes de classification

12.2.2.1 Classifieur de Bayes naïf augmenté par un arbre

Comme dans le cas des données complètes (voir section 8.1.2.3), il est aisé d'apprendre un *classifieur de Bayes naïf augmenté par un arbre*[ii]. La technique pour obtenir une structure de Bayes naïve augmentée par un arbre (François & Leray (2006b)) est très proche de celle pour obtenir une structure arborecente obtimale puisqu'elle consiste en la construction d'une structure arborescente optimale sur l'ensemble des observations, puis en la liaison de la classe à toutes les observations.

En pratique, il faut cependant faire attention au fait qu'*a posteriori* le nœud représentant la classe va être un parent de tous les autres nœuds donc la matrice (toujours avec des $-\infty$ sur la diagonale mais également sur la ligne et la colonne représentant la variable classe) de score que nous allons construire est donnée par l'équation 12.4. De même que dans le cas des données complètes, la méthode TAN-EM est issue de la méthode MWST-EM et permet d'assouplir les hypothèses faites par une structure de Bayes Naïve (voir section 8.1.2).

$$\left[M_{ij}^{q}\right]_{\substack{1 \leqslant i,j \leqslant n \\ i \neq j, i \neq C, j \neq C}} = \left[q_i^{BIC}\left(X_i, Pa(X_i) = \{C, X_j\}, \Theta_{Xi|X_jC} : \mathcal{T}^*, \Theta^*\right) - q_i^{BIC}\left(X_i, Pa(X_i) = \{C\}, \Theta_{Xi|C} : \mathcal{T}^*, \Theta^*\right) \right] \quad (12.4)$$

où le score local q_i^{BIC} est défini par l'équation 11.11.

En utilisant ensuite un algorithme pour obtenir l'arbre couvrant maximal (voir l'algorithme 6 de Kruskal (1956) ou celui de Jarník (1930)) puis en connectant le nœud classe à tous les autres nœuds, nous obtenons alors la structure de Bayes naïve augmentée par un arbre recherchée.

12.2.2.2 Classifieur naïf augmenté par une forêt : FAN-EM

En utilisant un critère d'arrêt prématuré dans les algorithmes de recherche de l'arbre couvrant de poids maximal, il est possible de construire une forêt (voir section 8.1.2.4).

De même, en utilisant l'algorithme MWST-EM sur les observations et en adaptant la matrice de poids comme dans la section 12.2.2.1 et en utilisant un critère d'arrêt prématuré dans l'algorithme 6, nous obtennons l'algorithme FAN-EM qui sélectionne le classifieur de bayes naïf augmenté par une forêt optimale.

[ii] Cette appellation est en fait un abus de langage issu de la littérature anglo-saxonne, car en français nous devrions dire *classifieur de Bayes naïf augmenté par une structure arborescente*.

12.2.2.3 Travaux connexes

Meila-Predoviciu (1999) applique l'algorithme MWST et le principe de l'algorithme EM, mais dans un autre cadre, celui de l'apprentissage de mélanges de structures arborescences. Dans son travail, la base d'exemples est complète mais une nouvelle variable est introduite pour prendre en compte l'importance à donner à chaque arborescence du mélange. Cette variable n'étant pas mesurée, elle utilise alors l'algorithme EM pour déterminer ses paramètres.

Peña, Lozano & Larrañaga (2002) proposent un changement à l'intérieur de l'algorithme SEM pour adapter cette approche plus spécifiquement à l'apprentissage de réseaux bayésiens plus efficaces pour une probématique de catégorisation (*clustering*). Dans ce cas, seule la variable classe n'est jamais mesurée, ce qui est un cas très particulier de données incomplètes.

Greiner & Zhou (2002) proposent de maximiser la vraisemblance conditionnelle lors de l'apprentissage des paramètres des réseaux bayésiens. Ils appliquent alors leur méthode à des bases de données incomplètes suivant de processus MCAR en utilisant l'*étude des cas disponibles* pour trouver le meilleur classifieur de Bayes naïf augmenté par un arbre.

Cohen, Cozman, Sebe, Cirelo & Huang (2004) traitent des classifieurs de Bayes naïfs augmentés par un arbre et de l'algorithme EM pour les données partiellement étiquettées. Dans leur travail, seulement la variable représentant la classe peut être manquante alors que n'importe quelle variable peut être partiellement observée dans notre algorithme TAN-EM.

Karčiauskas (2005) propose une technique pour apprendre des modèles hiérarchiques latents sous forme d'arbre. Dans cette approche, les nœuds observables sont les feuilles de l'arbre et la structure arborescente est construite en ajoutant des variables cachées parentes de certains sous-ensembles de variables observées.

12.2.3 Une proposition d'initialisation pour SEM

Comme dans le cadre des données complètement observées avec la recherche gloutonne GS (section 8.2.1), nous proposons d'observer ce qui se passe lorsque nous initialisons l'algorithme glouton SEM de Friedman (1997) par l'arbre obtenu par la méthode MWST-EM (François & Leray (2006b) et Leray & François (2005)). La méthode résultante est nommée SEM+T.

La méthode SEM est une technique gloutonne (donc itérative) qui utilise un algorithme EM (donc itératif) à l'intérieur de chacune de ses itérations. Elle requiert donc un temps de calcul important. Comme nous avons pu l'observer avec des bases de cas complètes, initialiser une recherche gloutonne dans l'espace des DAG par une structure arborescente optimale permet de réduire les temps de calcul tout en permettant également d'obtenir occasionnellement de meilleurs résultats que sans cette initialisation *intelligente*. Initialiser la méthode SEM par le résultat obtenu par MWST-EM pourrait donc permettre de réduire considérablement les temps de calcul, voire d'améliorer les résultats comme nous le verrons dans le chapitre 14.

12.2.4 Passer à l'espace des équivalents de Markov

Puisque SEM≃GS+EM, il est possible d'adapter l'algorithme GES introduit par Chickering (2002b) aux bases d'exemples incomplètes et, en suivant le même principe, créer une méthode GES-EM≃GES+EM.

La méthode GES est une méthode gloutonne dans l'espace des équivalents de Markov. Comme il n'est pas possible d'associer un score directement à une classe d'équivalence de Markov, nous devons à chaque itération effectuer les étapes suivantes :

- choisir un DAG représentant de la classe d'équivalence de l'itération courante,
- associer un score à ce DAG,
- prendre la classe d'équivalence de Markov du DAG possédant le meilleur score pour l'itération suivante.

Il faut de plus faire attention d'utiliser un score équivalent pour être certain qu'il y ait unicité du score d'une classe d'équivalence de Markov.

Comme un DAG représentant la classe de Markov est disponible à chaque itération (ce que fait la méthode de Chickering (2002b), voir section 8.2.3). En adaptant la méthode de calcul de score par celle introduite en section 11.2.2.1 et en l'appliquant à ce DAG, nous obtenons une méthode pour effectuer de l'apprentissage de structure dans l'espace des équivalents de Markov à partir de bases d'exemples incomplètes.

13

Génération de données incomplètes

"Qui témoigne de plus de force que l'homme
qui ne prend le hasard ni pour un dieu,
comme le fait la masse des gens,
ni pour une cause fluctuante."

Lettre à Ménécée
Épicure (341-270 avant JC)

Sommaire

13.1 Introduction 144
13.2 Échantillonnage à partir de réseaux bayésiens 144
13.3 Génération aléatoire de structures de réseaux bayésiens . . 145
13.4 Notations et hypothèses préliminaires 146
13.4.1 Hypothèses pour les situations MCAR et MAR 146
13.4.2 L'approche générale 147
13.5 Notre modélisation des processus MCAR 148
13.5.1 Le modèle 148
13.5.2 Identification des paramètres 149
13.5.3 Comment construire les β_i aléatoirement ? 149
13.5.4 Un exemple simple 149
13.5.5 Situation générale 150
13.6 Notre modélisation des mécanismes MAR 150
13.6.1 Le modèle 150
13.6.2 Identification des paramètres 151
13.7 Modélisations de processus de génération de données NMAR 152
13.8 Extension possible 153

13.1 Introduction

Le test des méthodes d'apprentissage de structure avec données incomplètes est une partie coûteuse en temps de calcul. Parfois, pour créer un protocole de tests, il faut développer une stratégie bien particulière. La phase laborieuse de notre protocole de tests a été de trouver des bases d'exemples incomplètes disponibles. Sur internet, peu sont effectivement disponibles, et celles qui le sont, ne représentent que peu de mécanismes mis en jeu par rapport aux situations MCAR, MAR et NMAR possibles.

Nous avons donc été amenés à concevoir une méthode permettant de générer des bases d'exemples incomplètes car la première méthode pour établir une confiance empirique envers une méthode particulière reste la multiplication des tests pour un nombre maximum de situations de tests possibles. Bien sûr, l'idéal est que ces différentes situations soient disponibles ou, au pire, générables automatiquement.

A travers les années, de nombreuses méthodes on été proposées pour générer des données de tests automatiquement.
Ces méthodes sont alors divisées en trois classes Ferguson & Korel (1996) :

- la génération de données aléatoires Thévenod-Fosse & Waeselynck (1993) et Deason, Brown, Chang & Cross II (1991),
- la génération de données orientée par une structure ou un chemin Ramamoorthy, Ho & W.T. (1976) et Korel (1990) (par exemple, utilisant des méthodes de statisfaction de contraintes DeMillo & Offutt (1991)),
- la génération de données orientée par un but Korel & Al-Yami (1996) (par exemple, en utilisant des algorithmes génétiques Pargas, Harrold & Peck (1999)).

Notre approche peut certainement être classée parmi les méthodes de génération de données aléatoires puisque nous allons effectuer un échantillonnage à partir d'une distribution de probabilités prédéfinie par un réseau bayésien.

De nos jours, les algorithmes de *Machine Learning* doivent être testés et comparés dans différents contextes. Comme de plus en plus de méthodes d'apprentissage peuvent prendre en compte des bases de données incomplètes, il faut être capable de générer un ensemble de bases incomplètes qui couvrent un spectre très large de types de données sur un spectre très large de taux de données manquantes également.

Notre méthode a pour but de modéliser les processus de génération de données incomplètes en utilisant le formalisme des réseaux bayésiens pour automatiquement générer des bases d'exemples avec différents pourcentages de données manquantes.

Les réseaux bayésiens peuvent être utilisés comme modèles génératifs, c'est pourquoi nous avons décidé d'utiliser ce formalisme pour la génération de données de tests. Détaillons, à présent, comment ils peuvent être utilisés pour l'échantillonnage de distribution de probabilités.

13.2 Échantillonnage à partir de réseaux bayésiens

L'idée de base des méthodes stochastiques est d'utiliser la connaissance disponible sur la distribution de probabilités pour générer automatiquement des exemples suivant cette loi.

L'échantillonnage probabiliste logique (*Probabilistic logic sampling* Henrion (1988)) est la méthode la plus simple et la première proposée pour effectuer de l'échantillonnage à partir de réseaux bayésiens. Cet échantillonnage est basé sur une notion de rejet.

Quand une distribution $\mathbb{P}$ n'est pas parfaitement connue, mais qu'une fonction $\mathbb{Q}$ telle que $1 \leqslant \frac{\mathbb{P}}{\mathbb{Q}} \leqslant M$ est connue, où M est une borne connue. Alors il est choisi de conserver l'exemple généré avec la probabilité $\frac{\mathbb{P}(\mathcal{X})}{M \cdot \mathbb{Q}(\mathcal{X})}$. Donc, plus M est grand, moins les exemples ont des chances d'être acceptés.
L'*importance sampling* décrit dans Shachter & Peot (1989) et Cheng & Druzdzel (2000) est proche de l'échantillonnage précédent sauf que la fonction $\mathbb{Q}$ est mise à jour pour actualiser les tables de probabilités conditionnelles (CPT) périodiquement pour permettre à la distribution évaluée de graduellement s'approcher de la distribution initiale. Pour cette méthode, un exemple est accepté avec la probabilité $\min\left(1, \frac{\mathbb{P}(x)\mathbb{Q}(x|x^{t-1})}{\mathbb{P}(x^{t-1})\mathbb{Q}(x^{t-1}|x)}\right)$.

Une autre famille de méthodes d'échantillonnages stochastiques est formée des méthodes dites de *Chaînes de Markov de Monte-Carlo* (MCMC). Cette famille est constituée de l'échantillonnage de Gibbs, de l'échantillonnage de Metropolis et des méthodes hybrides de Monte-Carlo (Geman & Geman (1984) et Gilks, Richardson & Spiegelhalter (1996)). Une fois appliquées aux réseaux bayésiens (Pearl (1987) et Chavez & Cooper (1990) et York (1992)), ces approches déterminent la distribution d'échantillonnage d'une variable à partir de ses tirages précédents sachant sa *couverture de Markov*.

L'échantillonnage de Gibbs est une adaptation particulière de l'échantillonnage de Metropolis-Hastings qui est utilisable lorsque nous avons un espace d'états factorisé. Cela est particulièrement adapté aux réseaux bayésiens, puisque ces derniers séparent l'espace d'états localement pour chaque variable conditionnellement à ses variables parentes

L'idée est simplement de passer d'un état (instanciation de variable) à un autre itérativement. L'algorithme peut être brièvement décrit par l'algorithme 7

ALGORITHME 7 L'échantillonneur de Gibbs.

1: Choisir une première variable,
2: échantillonner cette variable en utilisant sa CPT,
3: retourner en première étape jusqu'à ce que toutes les variables soient instanciées.

Pour l'échantillonneur de Gibbs, le rapport de rejet est alors $r = \frac{\mathbb{P}(x)\mathbb{Q}(x|x^{t-1})}{\mathbb{P}(x^{t-1})\mathbb{Q}(x^{t-1}|x)} = 1$ comme ici $\mathbb{Q} = \mathbb{P}$ (nous n'utilisons que les CPT qui sont parfaitement connues), et donc, tous les exemples sont conservés.

Nous proposons d'utiliser l'échantillonneur de Gibbs avec ces petites variations dans sa première étape. Le choix de la variable se fait en priorité parmi l'ensemble de nœuds racines du réseau bayésien. Si ces variables sont déjà toutes instanciées, le choix se fera parmi les nœuds qui ont leur ensemble de parents complètement instanciés (ce qui existe toujours).

13.3 Génération aléatoire de structures de réseaux bayésiens

Comme la méthode proposée a pour objectif de pouvoir générer de nombreuses bases d'exemples issues de mécanismes de génération de données variables, il serait appréciable de pouvoir générer de nombreux réseaux bayésiens aléatoirement. Par exemple, Xiang & Miller (1999) ou Ide, Cozman & Ramos (2004) ont proposé des méthodes pour faire cela à partir de chaînes de Markov.

13.4 Notations et hypothèses préliminaires

Soient n et m des entiers naturels et soient $\mathcal{X}_1^1, \ldots, \mathcal{X}_n^1, \mathcal{X}_1^2, \ldots, \mathcal{X}_n^m$, $n \times m$ variables aléatoires qui suivent les distributions $\mathfrak{X}_i^j$ respectivement, pour $1 \leqslant i \leqslant n$ et $1 \leqslant j \leqslant m$. Supposons la mise à disposition de la base d'exemples

$$\mathbf{D} = [[\mathbf{x}_i^j]]_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant m}}$$

La base $\mathbf{D}$ est alors une instanciation du vecteur aléatoire $\mathcal{D} = (\mathcal{X}_1^1, \ldots, \mathcal{X}_n^1, \mathcal{X}_1^2, \ldots, \mathcal{X}_n^m)$. Ce vecteur $\mathcal{D}$ suit alors la distribution $\mathfrak{D} = (\mathfrak{X}_1^1, \ldots, \mathfrak{X}_n^1, \mathfrak{X}_1^2, \ldots, \mathfrak{X}_n^m)$. Soient θ, les paramètres de la loi $\mathfrak{D}$, et $\mathbf{x}_i^j$, l'instanciation de $\mathcal{X}_i^j$ dans $\mathbf{D}$.

Soit $\mathcal{R} = (\mathcal{R}_1^1, \ldots, \mathcal{R}_n^1, \mathcal{R}_1^2, \ldots, \mathcal{R}_n^m)$ le vecteur aléatoire où les variables aléatoires $\mathcal{R}_i^j$ prennent la valeur $\mathbf{1}$ si $\mathcal{X}_i^j$ est dit manquant et la valeur $\mathbf{0}$ si $\mathcal{X}_i^j$ est observé.

Le vecteur aléatoire $\mathcal{R}$ suit la distribution $\mathfrak{R} = (\mathfrak{R}_1^1, \ldots, \mathfrak{R}_n^1, \mathfrak{R}_1^2, \ldots, \mathfrak{R}_n^m)$. Soit μ, les paramètres de la distribution $\mathfrak{R}$.

Soit $\mathbf{R} = [[\mathbf{r}_i^j]]_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant m}}$ la matrice où $\mathbf{r}_i^j = \mathbf{0}$ si $\S_i^j$ est observé et $\mathbf{1}$ sinon.

Finalement, soient $\mathbf{O} = \{\mathbf{x}_i^j\}_{\{(i,j)|\mathbf{r}_i^j=\mathbf{0}\}}$, la partie observée de la base $\mathbf{D}$ et $\mathbf{H} = \{\mathbf{x}_i^j\}_{\{(i,j)|\mathbf{r}_i^j=\mathbf{1}\}}$, la partie cachée de $\mathbf{D}$ alors

$$\mathbf{D} = \{\mathbf{O}, \mathbf{H}\}$$

Remarquons que dans ces notations, la base d'exemples $\mathbf{D}$ est toujours complète. Les variables dans $\mathbf{H}$ sont mesurées dans $\mathbf{D}$, mais "oubliées" quand les variables $\mathcal{R}$ prennent la valeur $\mathbf{1}$. Définissons maintenant $\mathbf{D}_{mes} = \{\mathbf{O}, \mathbf{H} = \textit{manquant}\}$, la base d'exemples incomplète "réelle" ayant pour seules valeurs observées, les valeurs dans $\mathbf{O}$.

Notre approche va échantillonner $\mathbf{D}$ puis $\mathbf{R}$ à partir des distributions $\mathfrak{D}$ et $\mathfrak{R}$ que nous proposons de modéliser par un réseau bayésien. La sortie de la méthode sera alors $\mathbf{D}_{mes}$, qui est construite en prennant seulement les valeurs $\mathbf{x}_i^j$ dans $\mathbf{D}$ telles que $\mathbf{r}_i^j = \mathbf{0}$.

Rappelons brièvement les trois mécanismes de données manquantes mis en évidence par Rubin (1976) (voir section 5.3.1). Nous avons donc l'équation suivante

$$\mathbb{P}(\mathcal{O}, \mathcal{H}, \mathcal{R}|\theta, \mu) = \mathbb{P}(\mathcal{O}, \mathcal{H}|\theta) \cdot \mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu) \tag{13.1}$$

Puisque $\mathcal{R}$ ne dépend pas de θ et $\mathcal{D}$ ne dépend pas de μ. Et voici les trois mécanismes possibles :

- **MCAR** quand $\mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu) = \mathbb{P}(\mathcal{R}|\mu)$.
- **MAR** quand $\mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu) = \mathbb{P}(\mathcal{R}|\mathcal{O}, \mu)$.
- **NMAR** quand $\mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu)$ ne se simplifie pas.

13.4.1 Hypothèses pour les situations MCAR et MAR

Souvent, l'hypothèse que les exemples de la base sont indépendants et identiquement distribués (*i.i.d*) est faite. Dans ce cas, les exemples $\mathbf{x}^j = (\mathbf{x}_1^j, \cdots, \mathbf{x}_n^j)$ ne doivent pas dépendre les uns des autres, et doivent suivre la même distribution de probabilités. Cela peut se traduire par les hypothèses suivantes :

H1 : *"si $j \neq l$, les variables aléatoires $\mathcal{X}_i^j$ et $\mathcal{X}_k^l$ sont indépendantes"*

H2 : *"pour tout j, les distributions $\mathfrak{X}_i^j$ sont identiques"*

Par la suite, nous oublierons donc le 'j', et appellerons $\mathfrak{X}_i$ cette unique distribution de

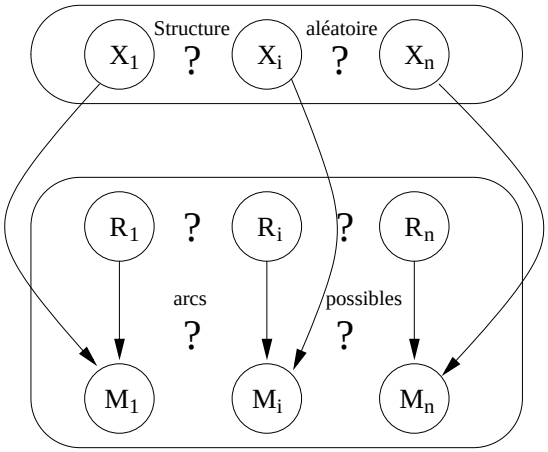


FIG. 13.1 : *Réseau bayésien générique pour la génération de bases d'exemples incomplètes.*

probabilités. Nous ferons de même pour les variables aléatoires $\mathcal{X}_i$ quand le contexte est clair.

Pour être cohérent, le même type d'hypothèse doit également être fait sur les distributions $\mathcal{R}_i^j$. Si ce n'est pas le cas, et si les mécanismes d'élimination de variables peuvent varier au court du temps, nous ne pourrons pas affirmer être en présence de données *i.i.d.* .

H3 : *"si $j \neq l$, les variables aléatoires $\mathcal{R}_i^j$ et $\mathcal{R}_k^l$ sont indépendantes"*

H4 : *"pour tout j, les distributions $\mathfrak{R}_i^j$ sont identiques"*

Par la suite, nous oublierons donc également l'indice 'j' pour les distributions $\mathfrak{R}_i$, et pour les variables $\mathcal{R}_i$ quand le contexte est clair.

Comme la base d'exemples que nous voulons créer est *i.i.d*, le fait qu'une donnée soit manquante ne doit pas avoir d'influence sur les autres valeurs de la base à des temps différents. D'où la dernière hypothèse :

H5 : *"si $j \neq l$, les variables aléatoires $\mathcal{X}_i^j$ et $\mathcal{R}_k^l$ sont indépendantes"*

Dans la section 13.7, des pistes seront données pour assouplir certaines de ces hypothèses dans les situations NMAR.

13.4.2 L'approche générale

L'approche que nous proposons ici s'appuie sur le formalisme des réseaux bayésiens Supposons que nous possédons un modèle génératif $\mathfrak{X}$ qui peut être utilisé pour générer une base de données complète. Ce modèle peut alors être représenté par le réseau bayésien contenu dans la boîte supérieure de la figure 13.1. Selon les hypothèses H1 et H2, il ne contient alors que les variables $\mathcal{X}_i$ qui sont représentées par les nœuds X_i.

Nous devons alors ajouter une variable $\mathcal{R}_i$ pour chaque variable $\mathcal{X}_i$ qui indiquera si cette dernière est observée ou non. De même, ces variables vont être représentées par les nœuds R_i.

L'exemple en sortie de cette approche sera la valeur des nœuds M_i qui sont évalués en utilisant seulement les valeurs des X_i et des R_i correspondantes.

L'obtention de bases d'exemples MCAR ou MAR dépendra de la manière de connecter les nœuds R_i, X_i et M_i mais également de la manière dont nous allons remplir les tables de probabilités conditionnelles car certaines indépendances peuvent naître de ces tables (notamment lorsqu'il y a des probabilités nulles).

Comme, pour le j-ième exemple, les variables $\mathcal{M}_i$ prennent la valeur de $\mathcal{X}_i$ si $\mathcal{R}_i = \mathbf{0}$ où la valeur *manquant* si $\mathcal{R}_i = \mathbf{1}$, les tables de probabilités des nœuds $\mathcal{M}_i$ sont connues et décrites dans la table 13.1 où s_i représente la taille de la $\mathcal{X}_i$.

$\mathbf{M}_i$ / $\mathbf{X}_i, \mathbf{R}_i$	$\mathbf{M}_i = v_1$	$\mathbf{M}_i = v_2$	$\mathbf{M}_i = v...$	$\mathbf{M}_i = v_{si}$	$\mathbf{M}_i = manquant$
$\mathbb{P}(\mathbf{M}_i \mid \mathbf{X}_i = v_1, \mathbf{R}_i = 0)$	1	0	0	0	0
$\mathbb{P}(\mathbf{M}_i \mid \mathbf{X}_i = v_2, \mathbf{R}_i = 0)$	0	1	0	0	0
$\mathbb{P}(\mathbf{M}_i \mid \mathbf{X}_i = v..., \mathbf{R}_i = 0)$	0	0	1	0	0
$\mathbb{P}(\mathbf{M}_i \mid \mathbf{X}_i = v_{s_i}, \mathbf{R}_i = 0)$	0	0	0	1	0
$\mathbb{P}(\mathbf{M}_i \mid \mathbf{X}_i = v_1, \mathbf{R}_i = 1)$	0	0	0	0	1
$\mathbb{P}(\mathbf{M}_i \mid \mathbf{X}_i = v_2, \mathbf{R}_i = 1)$	0	0	0	0	1
$\mathbb{P}(\mathbf{M}_i \mid \mathbf{X}_i = v..., \mathbf{R}_i = 1)$	0	0	0	0	1
$\mathbb{P}(\mathbf{M}_i \mid \mathbf{X}_i = v_{s_i}, \mathbf{R}_i = 1)$	0	0	0	0	1

TAB. 13.1 : *Table de probabilité pour $\mathbb{P}(M_i|X_i, R_i)$ pour des mécanismes MCAR et MAR.*

Remarquons que le fait d'utiliser des valeurs de probabilités nulles dans cette table introduit des indépendances. Par exemple, ici, le fait que $\mathcal{M}_i = manquant$ ne dépend que de la valeur **1** pour $\mathcal{R}_i$ et nullement de la valeur $\mathcal{X}_i$.

Les deux prochaines sections vont décrire plus en détails notre modélisation des processus MCAR et MAR.

13.5 Notre modélisation des processus MCAR

13.5.1 Le modèle

Jusqu'à présent, la plupart des méthodes proposées pour la génération de bases d'exemples MCAR éliminaient des variables avec une probabilité fixe α. Nous proposons ici une méthode plus générale qui autorise que les différentes variables aient des probabilités d'être manquantes différentes.

Comme $\mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu) = \mathbb{P}(\mathcal{R}|\mu)$ pour les processus MCAR et avec l'hypothèse H3, nous avons

$$\mathbb{P}(\mathcal{R}|\mu) = \prod_{j=1}^{m} \mathbb{P}(\mathcal{R}_1^j, \cdots, \mathcal{R}_n^j|\mu) \tag{13.2}$$

Mais il n'est pas possible d'avoir une décomposition plus fine étant donné que nous ne savons pas quelles peuvent être les dépendances ou indépendances entre les variables $\mathcal{R}_i^j$, pour j fixé.

En revanche, avec l'hypothèse H4, les variables $\mathcal{R}_i^j$ ne seront jamais dépendantes les unes des autres pour des valeurs de j différentes.

Ainsi quand nous sommes en présence d'un mécanisme MCAR, il est possible d'avoir des règles du type : "si $\mathcal{X}_i$ n'est pas observée alors $\mathcal{X}_k$ est manquante également".

Donc, les variables $\mathcal{M}_i$ dépendent des variables $\mathcal{R}_i$ et $\mathcal{X}_i$, pour un i fixé, mais les variables $\mathcal{R}_i$ peuvent être dépendantes les unes des autres quand i varie. La manière dont nous allons alors représenter les mécanismes MCAR est alors illustrée sur la figure 13.2.

Dans la prochaine section, nous allons mettre en évidence les paramètres qui doivent être fixés, puis proposer une manière de les fixer pour pouvoir créer automatiquement un générateur de *modèles générateurs* de bases d'exemples incomplètes.

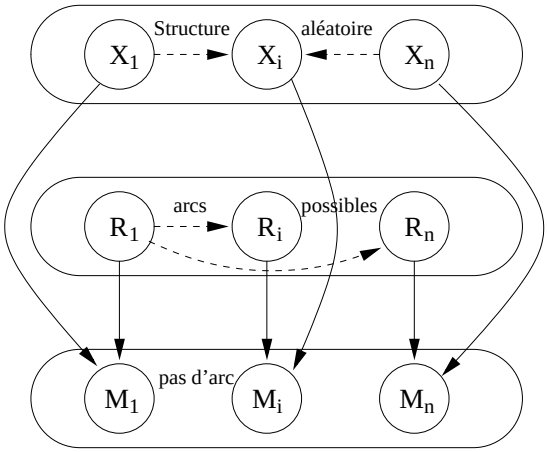


FIG. 13.2 : *Une modélisation de mécanismes MCAR par réseaux bayésiens.*

Ensuite, nous étudierons comment créer un modèle pour les mécanismes MCAR simples, puis plus généraux.

13.5.2 Identification des paramètres

Le seul paramètre que l'utilisateur final doit absolument fixer est le taux de valeurs manquantes moyen. Appelons α ce taux. Avec nos notations, nous avons donc

$$\mathbb{E}(\overline{\mathcal{R}}) = \alpha\,, \qquad \text{où} \qquad \overline{\mathcal{R}} = \frac{1}{n \cdot m} \sum_{i,j} \mathcal{R}_i^j \tag{13.3}$$

Les hypothèses H3 et H4 donnent alors $\mathbb{E}(\overline{\mathcal{R}}) = \dfrac{1}{n} \sum_{i=1}^{n} \mathbb{E}(\mathcal{R}_i^1)$.

De plus, comme $\mathbb{E}(\mathcal{R}_i^1) = \sum_{\mathbf{r} \in \{0,1\}} \Big(\mathbf{r} \cdot \mathbb{P}(\mathcal{R}_i^1 = \mathbf{r}) \Big) = \mathbb{P}(\mathcal{R}_i^1 = \mathbf{1})$, il reste

$\alpha = \dfrac{1}{n} \sum_{i=1}^{n} \mathbb{P}(\mathcal{R}_i^1 = \mathbf{1})$. Par la suite, nous noterons $\beta_i = \mathbb{P}(\mathcal{R}_i^1 = \mathbf{1})$ la probabilité que la variable $\mathcal{X}_i$ soit manquante et

$$\alpha = \frac{1}{n} \sum_{i=1}^{n} \beta_i \tag{13.4}$$

13.5.3 Comment construire les β_i aléatoirement ?

Soit α un nombre réel entre $[0, 1]$. Supposons que nous voulions générer un vecteur aléatoire $\beta = [\beta_1 \dots \beta_n]$ tel que chaque β_i appartient à l'ensemble $[0, 1]$ et tel que l'ensemble des β_i satisfasse la contrainte de l'équation 13.4.

De manière analogue à la transformation 'softmax', nous devons simplement générer une suite aléatoire $\Gamma = (\gamma_1, \cdots, \gamma_n)$ de nombres réels, puis construire les β_i en utilisant la formule de l'équation 13.5.

$$\beta_i = \frac{\alpha \cdot n \cdot \exp \gamma_i}{\sum_{i=1}^{n} \exp \gamma_i} \qquad (\forall i) \tag{13.5}$$

13.5.4 Un exemple simple

Supposons que nous voulions créer un mécanisme MCAR vérifiant l'hypothèse supplémentaire suivante (seulement valable dans cette sous-section) :

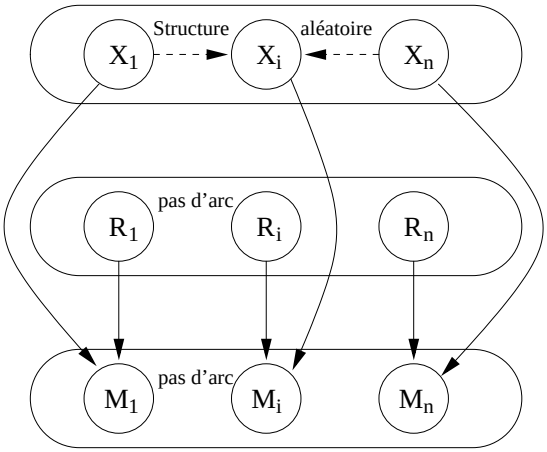


FIG. 13.3 : *Un exemple simple de modélisation de mécanisme MCAR à l'aide de réseau bayésien.*

"Les variables aléatoires $\mathcal{R}_i$ et $\mathcal{R}_k$ sont marginalement indépendantes si $i \neq k$".

Dans ce cas, nous pouvons représenter le modèle par le réseau bayésien de la figure 13.3.

Pour remplir les tables de probabilités *a priori* des nœuds R_i, il faut générer un vecteur de probabilité $\beta = (\beta_1, \cdots, \beta_n)$ respectant la contrainte de l'équation 13.4. Alors il ne reste plus qu'à construire les tables suivantes et les associer aux nœuds correspondants.

$$\mathbb{P}(\mathcal{R}_i) = [1 - \beta_i\,,\ \beta_i] \tag{13.6}$$

13.5.5 Situation générale

En pratique, il peut être utile de modéliser des procédés de génération de données MCAR plus génériques. Dans ce cas, des arcs entre les nœuds R_i doivent être créés. Le modèle résultant est illustré par la figure 13.2.

La méthode précédente proposait de générer aléatoirement les tables de probabilités $\mathbb{P}(\mathcal{R}_i)$, c'est-à-dire des nœuds n'ayant pas de parents. La méthode utilisée pour déterminer les tables de probabilités *conditionnelles* $\mathbb{P}(\mathcal{R}_i|\mathcal{R}_1, \cdots, \mathcal{R}_{j\neq i}, \cdots, \mathcal{R}_n)$ est proche de celle exposée précédemment et va être présentée en section 13.6.2 puisqu'elle sera également utilisée pour les mécanismes MAR.

13.6 Notre modélisation des mécanismes MAR

13.6.1 Le modèle

Pour les processus de génération de données MAR, les nœuds représentant l'absence ou non d'une variable ne peuvent plus être indépendants des valeurs observées (rappelons nous que $\mathbb{P}(\mathcal{R}|\mathcal{O}, \mathcal{H}, \mu) = \mathbb{P}(\mathcal{R}|\mathcal{O}, \mu)$). Comme nous pouvons le voir sur la figure 13.4, les liens entre les variables M_i et les variables R_i permettent de représenter ces dépendances.

De plus, il n'est plus utile de mettre des liens entre les variables R_i car les dépendances entre ces variables proviennent à présent du fait que les variables R_i dépendent de certaines variables M_i et que les variables M_i sont dépendantes d'autres variables R_i (de manière très forte à la vue de la table de l'équation 13.4.2). Dans ce cas si une variable est manquante ($\mathrm{R}_i = \mathbf{1}$) alors $\mathrm{M}_i = \mathit{manquant}$ et donc
$\mathbb{P}(\mathrm{R}_{i'} = \mathbf{1}|\mathrm{R}_i = \mathbf{1}) = \mathbb{P}(\mathrm{R}_{i'} = \mathbf{1}|\mathrm{M}_i = \mathit{manquant})$.

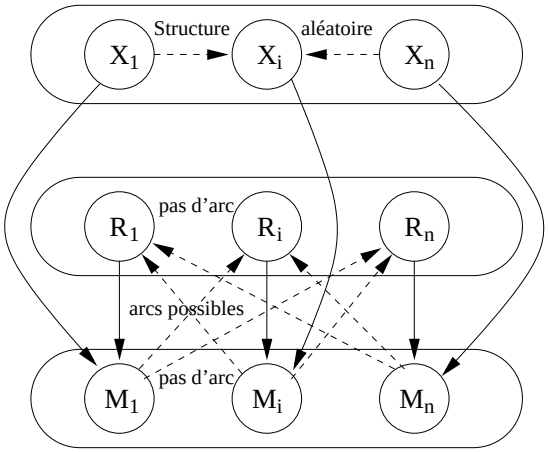


FIG. 13.4 : *Un réseau bayésien pour la modélisation de mécanismes MAR.*

Avec ce modèle, nous avons juste à ajuster la probabilité que $\mathrm{R}_i = \mathbf{1}$ connaissant les valeurs des variables M_i dont elle dépend. C'est-à-dire que nous devons fixer des valeurs (aléatoirement ou non) pour la matrice $\mu = [[[\mu_{ibk}]]]_{1 \leqslant i \leqslant n,\ \mathbf{b} \in \{\mathbf{0},\mathbf{1}\},\ \mathbf{k} \in K_i}$ où

$$\mu_{ibk} = \mathbb{P}(\mathcal{R}_i = \mathbf{b} | \mathcal{P}a(\mathrm{R}_i) = \mathbf{k}) \tag{13.7}$$

avec $1 \leqslant i \leqslant n$, $\mathbf{b} \in \{\mathbf{0},\mathbf{1}\}$ et $\mathbf{k} \in K_i$ où K_i est l'ensemble de toutes les configurations possibles pour le vecteur aléatoire $\mathcal{P}a(\mathrm{R}_i)$.

13.6.2 Identification des paramètres

Rappelons que $\beta_i = \mathbb{P}(\mathcal{R}_i = \mathbf{1})$, alors $\beta_i = \sum_{\mathbf{k}} \mathbb{P}(\mathcal{R}_i = \mathbf{1}, \mathcal{P}a(\mathrm{R}_i) = \mathbf{k})$ et la formule de Bayes donne

$$\beta_i = \sum_{\mathbf{k}} \mu_{i\mathbf{1}k} \cdot \xi_{ik} \tag{13.8}$$

où $\xi_{ik} = \mathbb{P}(\mathcal{P}a(\mathrm{R}_i) = \mathbf{k})$. Toutes les valeurs des ξ sont obtenues par inférence dans le réseau bayésien original (de la boite supérieure) car nous pouvons réécrire ξ_{ik} comme $\xi_{ik} = \mathbb{P}(\{\mathrm{X}_i = \mathbf{x}_i\}_{\{i|\mathrm{M}_i \neq manquant\}} | \mathcal{P}a(\mathrm{R}_i) = \mathbf{k})$ et n'utiliser que les valeurs observées ($\neq$ *manquant*) de l'ensemble des parents de R_i comme ceux-ci sont tous des M_i dont l'ensemble d'états admissibles est le même que celui des X_i (plus l'état *manquant* mis de coté ici).

Nous pouvons donc simplement utiliser une méthode proche de celle exposée en section 13.5.3 pour générer les paramètres μ_{i1k} aléatoirement. La formule de l'équation 13.5 devient alors celle de l'équation 13.10.

$$\mu_{i1k} = \frac{\beta_i}{\xi_{ik}} \frac{\exp \gamma_{ik}}{\sum_{k \in K_i} \exp \gamma_{ik}}\ ,\ 1 \leqslant i \leqslant n\ ,\ \mathbf{k} \in K_i\ . \tag{13.9}$$

ou encore,

$$\mu_{i1k} = \frac{n\alpha}{\xi_{ik}} \frac{\exp \gamma_{ik}}{\sum_{i=1}^{n} \sum_{k \in K_i} \exp \gamma_{ik}}\ ,\ 1 \leqslant i \leqslant n\ ,\ \mathbf{k} \in K_i\ . \tag{13.10}$$

Ceci est vrai car

$$\begin{aligned}\alpha &= \frac{1}{n}\sum_{i=1}^{n}\beta_i = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{k\in K_i}\mu_{i\mathbf{1}k}\cdot\xi_{ik}\right)\\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{k\in K_i}\frac{n\alpha}{\xi_{ik}}\cdot\frac{\exp\gamma_{ik}}{\sum_{i=1}^{n}\sum_{k\in K_i}\exp\gamma_{ik}}\cdot\xi_{ik}\right)\end{aligned}$$

Maintenant, supposons que le nœud R_i possède seulement le nœud $\mathrm{M}_{i'}$ comme parent. La table de probabilités conditionnelles $\mathbb{P}(\mathcal{R}_i|\mathcal{P}a(\mathrm{R}_i))$ est alors complètement déterminée par

$$^t\begin{bmatrix}\mu_{i01},\cdots,\mu_{i0k},\cdots,\mu_{i0s_i},1-\beta_i\\ \mu_{i11},\cdots,\mu_{i1k},\cdots,\mu_{i1s_i},\quad\beta_i\end{bmatrix} \tag{13.11}$$

où s_i est la taille du nœud X_i et où $\mu_{i0k} = 1-\mu_{i1k}$.

Ici, il est possible d'utiliser la formule 13.10 pour toutes les configurations en excluant celles où l'unique parent est *manquant* car prendre en considération un état supplémentaire de probabilité fixé *a priori* à β_i n'a aucune influence sur le résultat comme β_i est la moyenne des valeurs.

Il y a de nombreuses manières de créer des mécanismes NMAR et dans le prochain paragraphe nous allons donner quelques pistes pour faire cela.

13.7 Modélisations de processus de génération de données NMAR

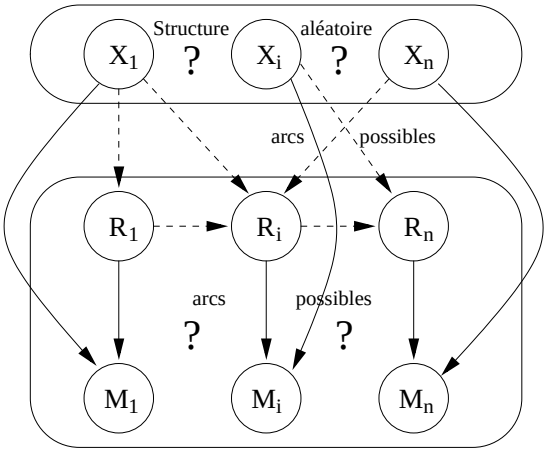


FIG. 13.5 : *Un réseau bayésien pour la modélisation de mécanismes MAR* i.i.d.

Tous les processus de génération de données qui ne sont pas MAR sont NMAR (rappelons que MCAR est un cas particulier de MAR).

La première manière de créer un processus de génération de données NMAR est simplement de reprendre ce qui a été fait dans le cas MAR, mais de construire les tables de probabilités différemment : soit en changeant les tables des nœuds M_i, soit en ne respectant pas les règles énoncées précédemment. Comme pour avoir un processus NMAR, la probabilité qu'une valeur soit manquante peut dépendre des valeurs des variables observées, nous proposons d'utiliser un modèle proche de celui illustré sur la figure 13.5.

Une autre technique consiste alors en l'introduction de variables supplémentaires, ces variables doivent alors avoir une influence sur les nœuds R_i, qu'elle soit directe ou

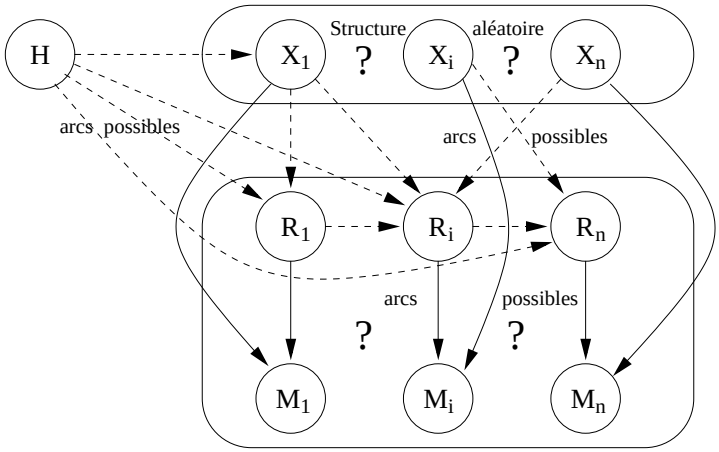


FIG. 13.6 : *Utiliser une variable* cachée *pour modéliser de mécanismes MAR* i.i.d.

indirecte. Un exemple utilisant une variable dite latente (par analogie car elle ne sera jamais observée par l'utilisateur qui utilisera la base d'exemples) est illustré en figure 13.6.

Le principal problème des situations NMAR est que, en pratique, les bases de données ne sont pas *i.i.d*. Le nombre de mécanismes NMAR pour construire des bases d'exemples pseudo-réelles est alors infini et dépend de nombreux facteurs externes.

Une autre solution pour créer de telles données (non *i.i.d.*) est alors de représenter les données comme étant dépendantes du temps, et donc des autres variables à des instants précédents. Pour cela, il est possible d'avoir recours au formalisme des réseaux bayésiens dynamiques (section 2.2.4). Par exemple, en connectant les variables $\mathcal{R}_i(t)$ aux variables $\mathcal{R}_i(t+1)$.

Il est alors possible d'imaginer mixer ces différentes approches comme illustré sur la figure 13.7.

13.8 Extension possible

La méthode présentée ici pour les variable discrètes peut être étendue aux variables continues en utilisant des modèles conditionnels gaussiens pour les variables $\mathcal{X}$, et des fonctions *softmax* pour les distributions de probabilités conditionnelles $\mathbb{P}(\mathcal{R}_i|\mathcal{X}_j,\mathcal{R}_k)$.

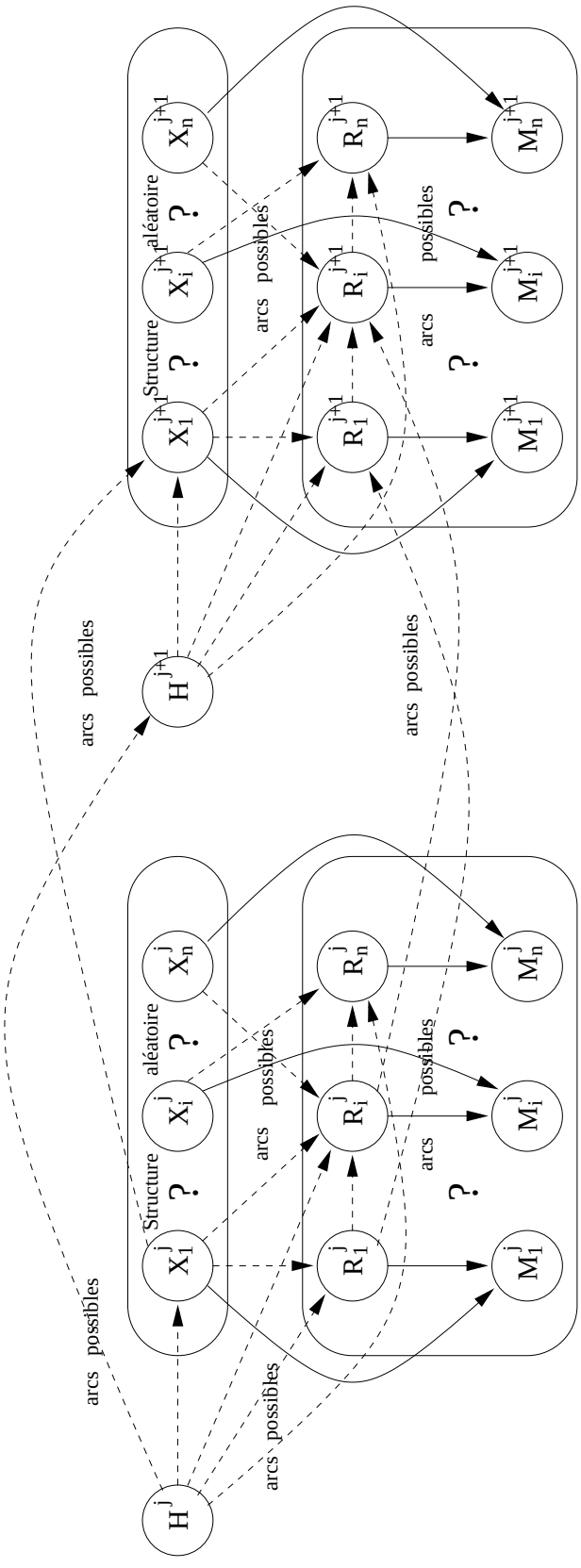


FIG. 13.7 : *Un réseau bayésien pour la modélisation de mécanismes MAR non* i.i.d.

14

Expérimentations

"Les questions les plus importantes de la vie ne sont en fait,

pour la plupart, que des problèmes de probabilités."

Pierre-Simon de Laplace (1749-1827)

Sommaire

14.1 Recherche d'une bonne structure 156
14.1.1 Algorithmes utilisés 156
14.1.2 Réseaux tests et techniques d'évaluation 156
Réseaux tests 156
Critères d'évaluation 157
14.1.3 Résultats et interprétations 157
14.1.3.1 Etude des cas complets contre méthode EM 157
14.1.3.2 Etude des cas disponibles contre méthode EM . . . 157
14.1.3.3 Stabilité en fonction de la taille de la base 160
14.1.3.4 Stabilité en fonction du taux de données manquantes 162
14.1.3.5 Duels entre les différentes méthodes gloutonnes . . 162
14.2 Recherche d'un bon classifieur 164
14.2.1 Techniques utilisées et techniques d'évaluation 164
14.2.2 Résultats et interprétations 164

14.1 Recherche d'une bonne structure

14.1.1 Algorithmes utilisés

Le code des fonctions mises en œuvre dans ces expérimentations est mis à disposition par l'intermédiare du *Structure Learning Package* décrit dans Leray & François (2004b).

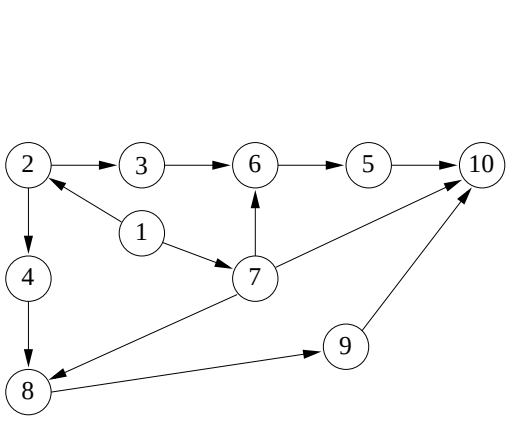
Nous avons testé les algorithmes suivants :

- MWST-EM avec une racine aléatoire et le score Q^{BIC} (section 12.2.1),
- SEM initialisé avec une chaîne, score Q^{BIC} (section 12.1.1),
- SEM+T initialisé par une structure arborescente optimale, score Q^{BIC} (section 12.2.3),

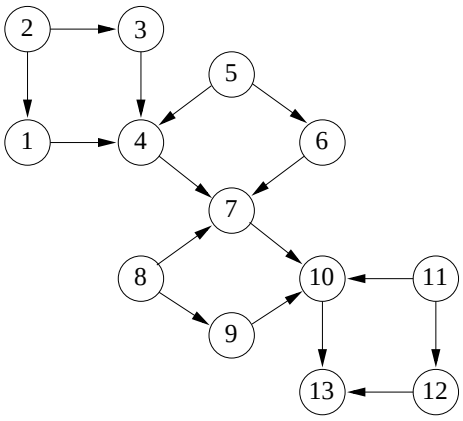
et comparé à leur version où l'apprentissage est effectué avec les exemples complets (CCA, nommées respectivement MWST-CCA, GS-CCA et GS+T-CCA), et leur version ou l'apprentissage est effectué avec les exemples disponibles (ACA, nommées respectivement MWST-ACA, GS-ACA et GS+T-ACA).

14.1.2 Réseaux tests et techniques d'évaluation

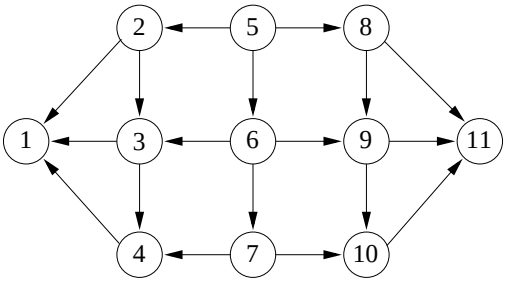
Réseaux tests



(a) Jouet 6



(b) Jouet 7



(c) Jouet 8

FIG. 14.1 : *Les réseaux de tests supplémentaires.*

Pour cette première série d'expériences, nous allons utiliser des réseaux bayésiens connus qui sont illustrés sur les figures 9.1 et 14.1. Ces réseaux vont alors être utilisés pour générer des bases d'exemples de différentes tailles par échantillonnage (voir chapitre 13).

Nous utilisons les bases d'exemples ainsi créées pour tester différentes méthodes d'apprentissage de structure. Pour une structure fixée, nous générons 10 bases d'exemples

MCAR et 10 bases d'exemples MAR avec des valeurs des paramètres diférentes pour chaque valeur de taille de la base d'exemples (100, 200, 400, 600, 1000 et 2000 exemples). Nous exécutons ensuite chaque algorithme d'apprentissage sur chacune de ces bases qui sont alors au nombre de 9 réseaux $\times 10$ valeurs des paramètres $\times 4$ taux de données manquantes $\times 6$ tailles $\times 2$ types de données manquantes $= 4320$ bases d'exemples. Les réseaux obtenus sont alors comparés au réseau original à l'aide de différentes mesures d'évaluation.

Critères d'évaluation

Les critères d'évaluation sont le score *BIC* et la divergence de Kullback-Leibler. Ceux-ci sont décrits plus en détails dans la section 9.1.2.

14.1.3 Résultats et interprétations

14.1.3.1 Etude des cas complets contre méthode EM

La table 14.1 nous montre les différences obtenues pour les scores *BIC* entre la méthode EM et les méthodes CCA et ACA. A première vue, nous observons qu'il n'y a pas de différences remarquables entre les résultats obtenus avec les données MCAR et ceux obtenus avec des données MAR.

Nous observons que dans le cas de l'étude des cas complets (CCA), la méthode a été capable (plus de 10 exemples disponibles) de ne traiter que 95% des bases avec 20% de données manquantes, 77% des bases avec 30% de données manquantes, 49% des bases avec 40% de données manquantes et 24% des bases avec 50% de données manquantes. Ici les exemples traités contiennent de 5 à 12 variables, or le taux d'exemples complètement observés baisse lorsque le nombre de variables augmente. Une telle technique ne permet donc pas de traiter des bases d'exemples ayant beaucoup d'attributs efficacement.

Les résultats obtenus sont ici toujours plus mauvais que ceux obtenus avec la méthode EM du point de vue de la divergence de Kullback-Leiber car la grande majorité des points sont au dessus de la bissectrice (critère à minimiser) quel que soit le taux de données manquantes.

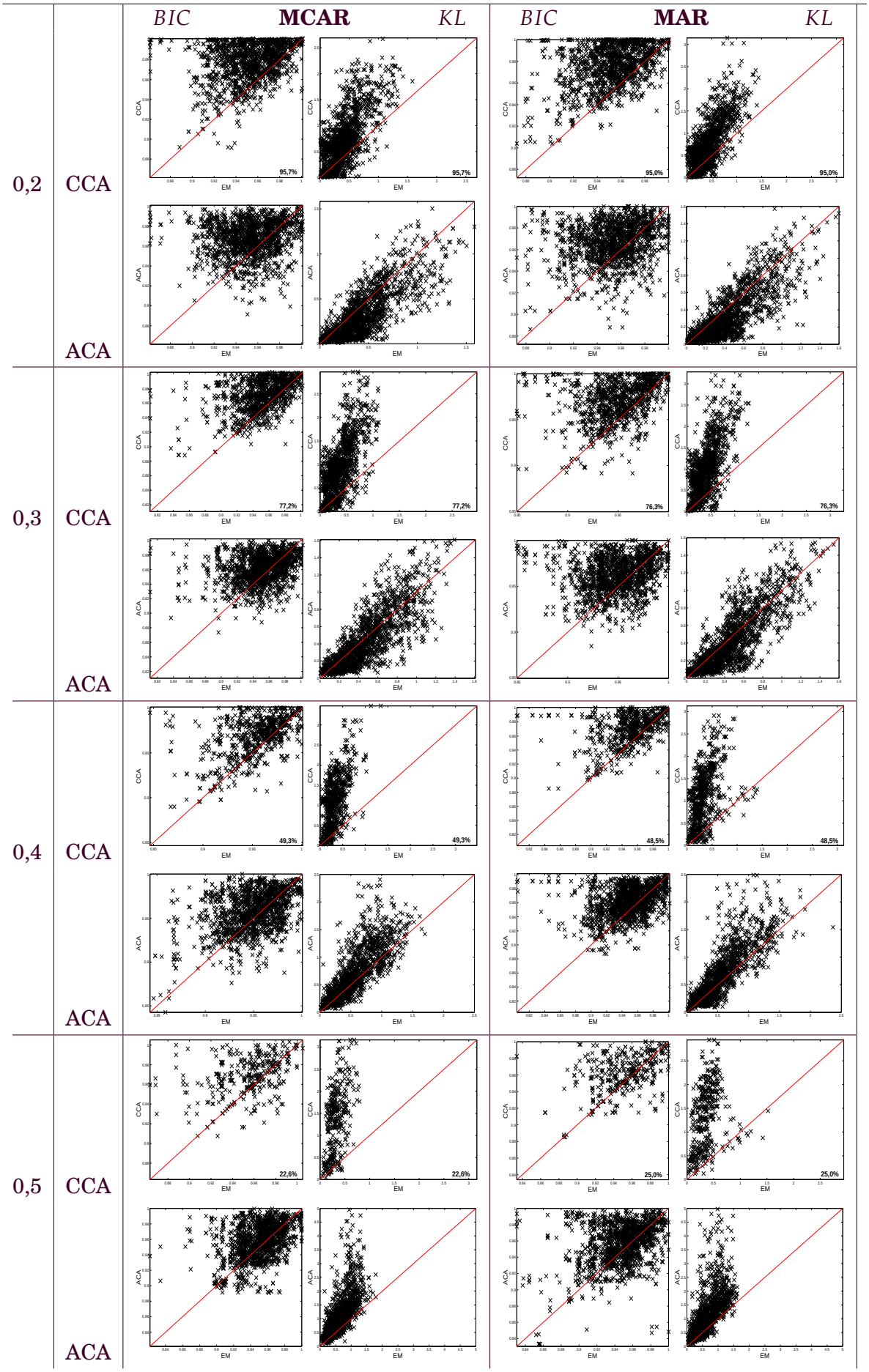
Néanmoins, il est étonnant de voir que cette méthode permet régulièrement d'obtenir de meilleurs scores *BIC* qu'un apprentissage de la structure avec EM et cela, même lorsque le taux de données incomplètes dépasse les 40%.

Cela peut être dû au fait que le nombre d'arcs découverts doit être plus faible car, étant donné que moins d'exemples sont disponibles, l'augmentation de la vraisemblance due à l'ajout d'un arc est alors, elle aussi, plus faible (durant l'apprentissage bien sûr car les scores *BIC* reportés sont tous calculés de la même manière sur une *grande* base de test complète) et donc plus facilement absorbée par l'augmentation du terme de pénalité.

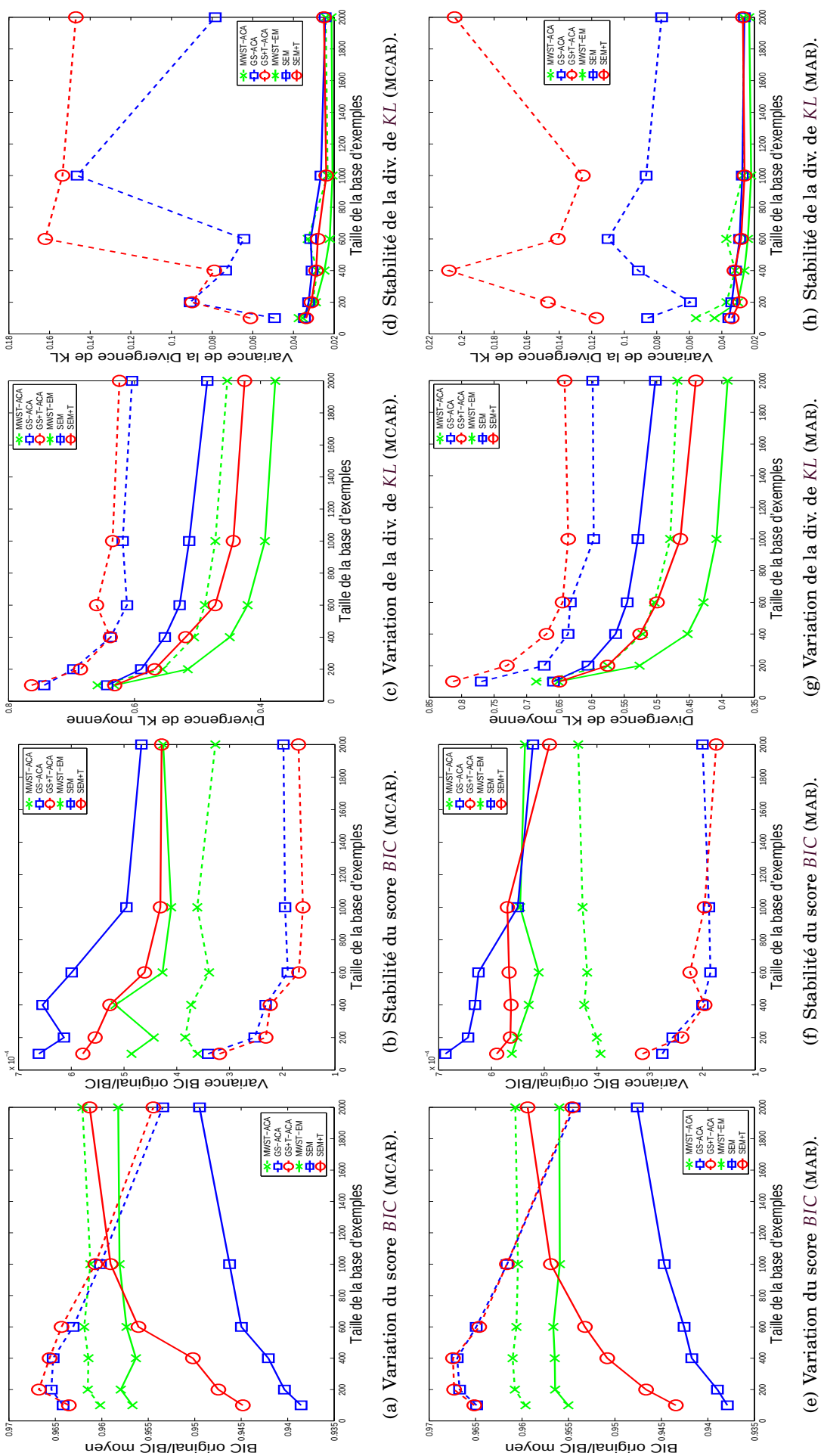
Cette méthode permet probablement d'obtenir des résultats corrects pour la divergence de Kullback-Leiber avec de faibles taux de valeurs manquantes (de l'ordre de 5% ou moins). Cela n'a pas été testé ici car la méthode CCA peut aisément être remplacée par la méthode ACA en pratique.

14.1.3.2 Etude des cas disponibles contre méthode EM

La méthode d'étude des cas disponibles est, quant à elle, bien plus efficace. Nous observons qu'elle obtient également des résultats similaires avec des données MCAR et MAR. Non seulement, cette méthode permet d'obtenir régulièrement de meilleurs scores



TAB. 14.1 : *Score (BIC réseau original)/BIC et divergence de KL pour la méthode EM contre les méthodes CCA et ACA. La méthode EM est toujours en abscisses. Le pourcentage pour la méthode ACA indique le nombre de bases traitées (plus de 10 exemples complètement observés).*



(a) Variation du score *BIC* (MCAR). (b) Stabilité du score *BIC* (MCAR). (c) Variation de la div. de *KL* (MCAR). (d) Stabilité de la div. de *KL* (MCAR).

(e) Variation du score *BIC* (MAR). (f) Stabilité du score *BIC* (MAR). (g) Variation de la div. de *KL* (MAR). (h) Stabilité de la div. de *KL* (MAR).

FIG. 14.2 : *Stabilité en fonction de la taille de la base d'exemples pour des données manquantes MCAR et MAR.*

BIC qu'avec un apprentissage utilisant EM, mais elle permet également d'obtenir de meilleures divergences de Kullback-Leiber lorsque le taux de données manquantes est inférieur à 35%. Ceci est visible dans la table 14.1 mais également sur les figures 14.2 et 14.3 où nous voyons alors la méthode ACA prendre l'avantage avec 30% de données manquantes, mais le perdre avec 40%.

Cette méthode reste donc une technique de choix pour évaluer des probabilités en présence de bases de données incomplètes lorsque le taux de données manquantes est inférieur à 40% (figures 14.3(a),14.3(e),14.3(c) et 14.3(g)) ou lorsque la base d'exemples possèdent moins de 1500 cas (figures 14.2(a) et 14.3(e)). Néanmoins, nous pouvons voir dans la table 14.1 que cette méthode obtient de moins bons résultats que la méthode EM du point de vue de la divergence de Kullback-Leiber lorsque le taux de données manquantes dépasse les 40%.

14.1.3.3 Stabilité en fonction de la taille de la base d'exemples

La figure 14.2 illustre les variations du score *BIC* et de la divergence de Kullback-Leiber lorsque la taille de la base d'exemples varie de 100 à 2000 cas. Les méthodes d'apprentissage utilisant les exemples disponibles sont visibles en traits pointillés, et celles utilisant la méthode EM sont représentées en traits pleins.

A la comparaison des deux lignes, nous pouvons remarquer que le fait que les données soient manquantes avec un mécanisme d'élimination MCAR (lignes supérieures des figures 14.2 et 14.3) ou MAR (lignes inférieures des figures) a peu d'influence sur les résultats.

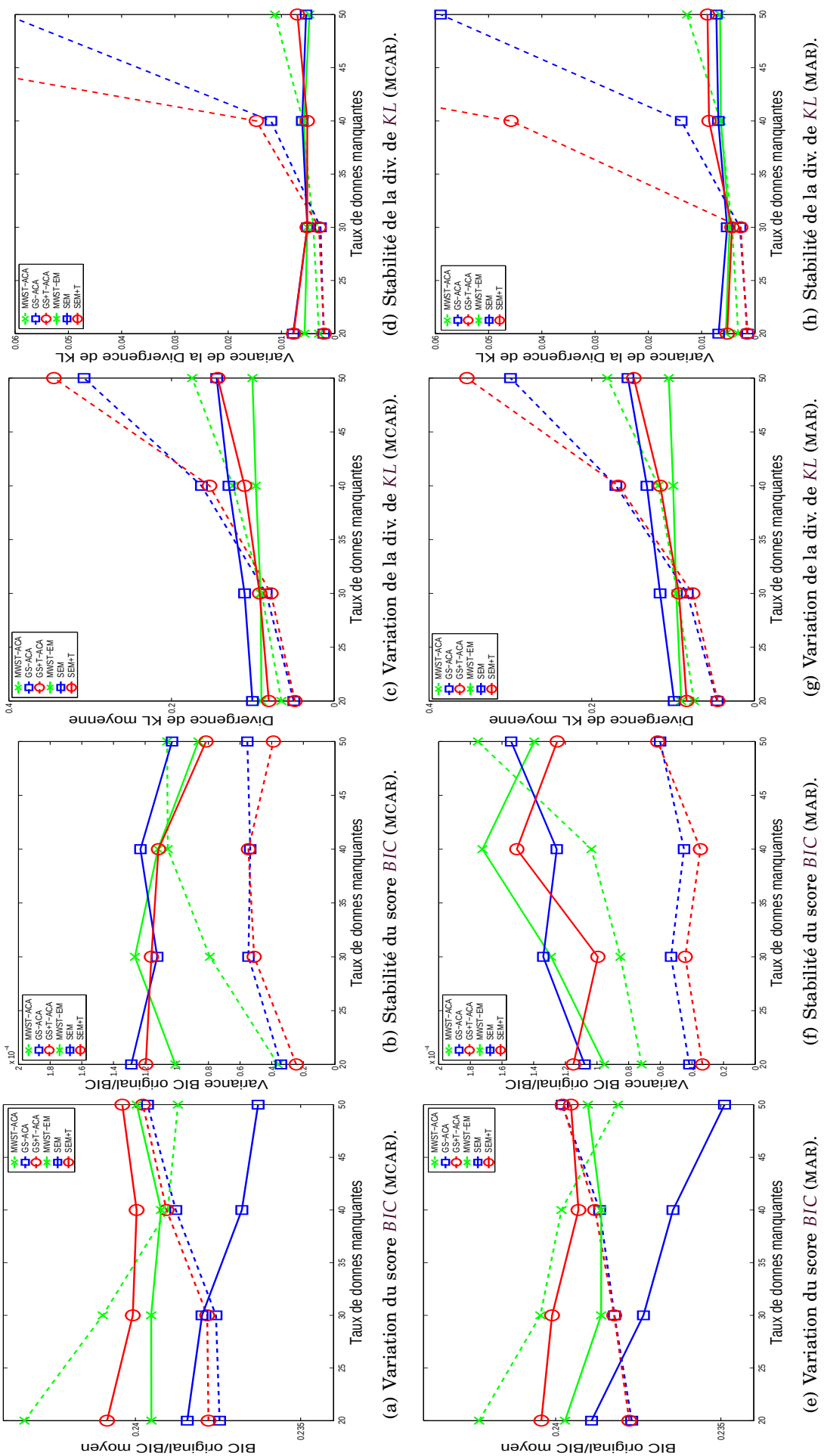
Lorsque la taille de la base d'exemples augmente, les méthodes utilisant l'algorithme EM voient leur score *BIC* augmenter légèrement. Ces méthodes sont alors très stables par rapport au nombre d'exemples, mais elles sont légèrement plus efficaces quand la base croît (figures 14.2(a) et 14.2(e)).

Par ailleurs, il est surprenant de voir que les méthodes utilisant l'étude des cas disponibles voient alors leurs performances décroîtrent lorsque la taille de la base d'exemples augmente. L'utilisation de ACA a l'avantage d'obtenir de meilleurs résultats lorsque la base est de petite taille, mais l'utilisation de EM permet alors d'être plus efficace lorsque la taille de la base d'exemples croît.

Nous remarquons de plus que l'utilisation de ACA permet d'obtenir des résultats plus stables du point de vue du score *BIC* (figures 14.2(b) et 14.2(f)).

Du point de vue de la divergence de Kullback-Leiber, le constat est tout autre. Nous observons sur les figures 14.2(c) et 14.2(g) que les réseaux obtenus en utilisant la méthode EM sont significativement meilleurs en moyenne que ceux obtenus avec une étude des exemples disponibles pour les méthodes SEM et SEM+T. Remarquons que la méthode MWST-ACA est la seule méthode directe testée ici, toutes les autres méthodes étant itératives. Les variances des divergences de Kullback-Leiber obtenues mènent à la même conclusion (figures 14.2(d) et 14.2(h)).

Les variances des résultats de toutes les méthodes utilisant l'algorithme EM sont équivalentes et restent les mêmes quelle que soit la taille de la base d'exemples (figures 14.2(b),14.2(f),14.2(d) et 14.2(h)), la méthode MWST-EM étant néanmoins légèrement plus stable.



(a) Variation du score *BIC* (MCAR). (b) Stabilité du score *BIC* (MCAR). (c) Variation de la div. de *KL* (MCAR). (d) Stabilité de la div. de *KL* (MCAR).

(e) Variation du score *BIC* (MAR). (f) Stabilité du score *BIC* (MAR). (g) Variation de la div. de *KL* (MAR). (h) Stabilité de la div. de *KL* (MAR).

FIG. 14.3 : *Stabilité en fonction du taux de données manquantes MCAR ou MAR.*

Ces variances sont par ailleurs comparables que les données soient MCAR ou MAR.

La méthode MWST-EM reste très stable du point de vue du score *BIC* en fonction de la taille de la base d'exemples (figures 14.2(a) et 14.2(e)). Cette méthode obtient alors de meilleurs résultats que les recherches gloutonnes utilisant EM lorsque la base d'exemples possède moins de 600 exemples. La méthode SEM obtient ensuite de meilleurs résultats lorsque la base d'exemples contient plus de 600 cas.

Sur ces petits exemples, la méthode MWST-EM permet d'obtenir de plus faibles divergences de Kullback-Leiber en moyenne quelle que soit la taille de la base d'exemples.

14.1.3.4 Stabilité en fonction du taux de données manquantes

La figure 14.3 illustre la stabilité des méthodes d'apprentissage de structure à partir de données incomplètes testées en fonction du taux de données manquantes qui varie de 20% à 50%.

Cette fois encore nous observons que les méthodes utilisant l'algorithme EM obtiennent des scores *BIC* qui dépendent peu du taux de données manquantes (figures 14.3(a) et 14.3(e)). De manière suprenante, nous nous apercevons que les méthodes gloutonnes SEM-ACA et SEM+T-ACA utilisant l'analyse des exemples disponibles voient leurs résultats s'améliorer lorsque le nombre de données manquantes augmente.

La méthode MWST-ACA est la méthode qui obtient les meilleurs scores *BIC* lorsque le taux de données incomplètes est faible. Ceci est dû au bon compromis entre la complexité du réseau bayésien obtenu et sa bonne vraisemblance.

Néanmoins, cette fois, les méthodes utilisant ACA obtiennent des résultats meilleurs du point de vue de la divergence de Kullback-Leiber lorsque le taux de données incomplète est sous les 35% mais ces résultats deviennent alors rapidement très mauvais (figures 14.3(c) et 14.3(g)) et instables (figures 14.3(d) et 14.3(h)) lorsque le taux continue d'augmenter.

Les méthodes utilisant EM restent stables lorsque les données sont MCAR (figure 14.3(d)) mais deviennent également instables, mais moins rapidement que les méthodes utilisant ACA, lorsque les données sont MAR (figure 14.3(h))

Les divergences de Kullback-Leiber moyennes obtenues par les méthodes MWST-EM, SEM et SEM+T sont très similaires. La méthode MWST-EM, qui obtient des résultats de qualité comparable lorsque le taux de données manquantes est faible, devient ensuite moins bonne lorsque le taux de données manquantes augmente (figure 14.3(g)).

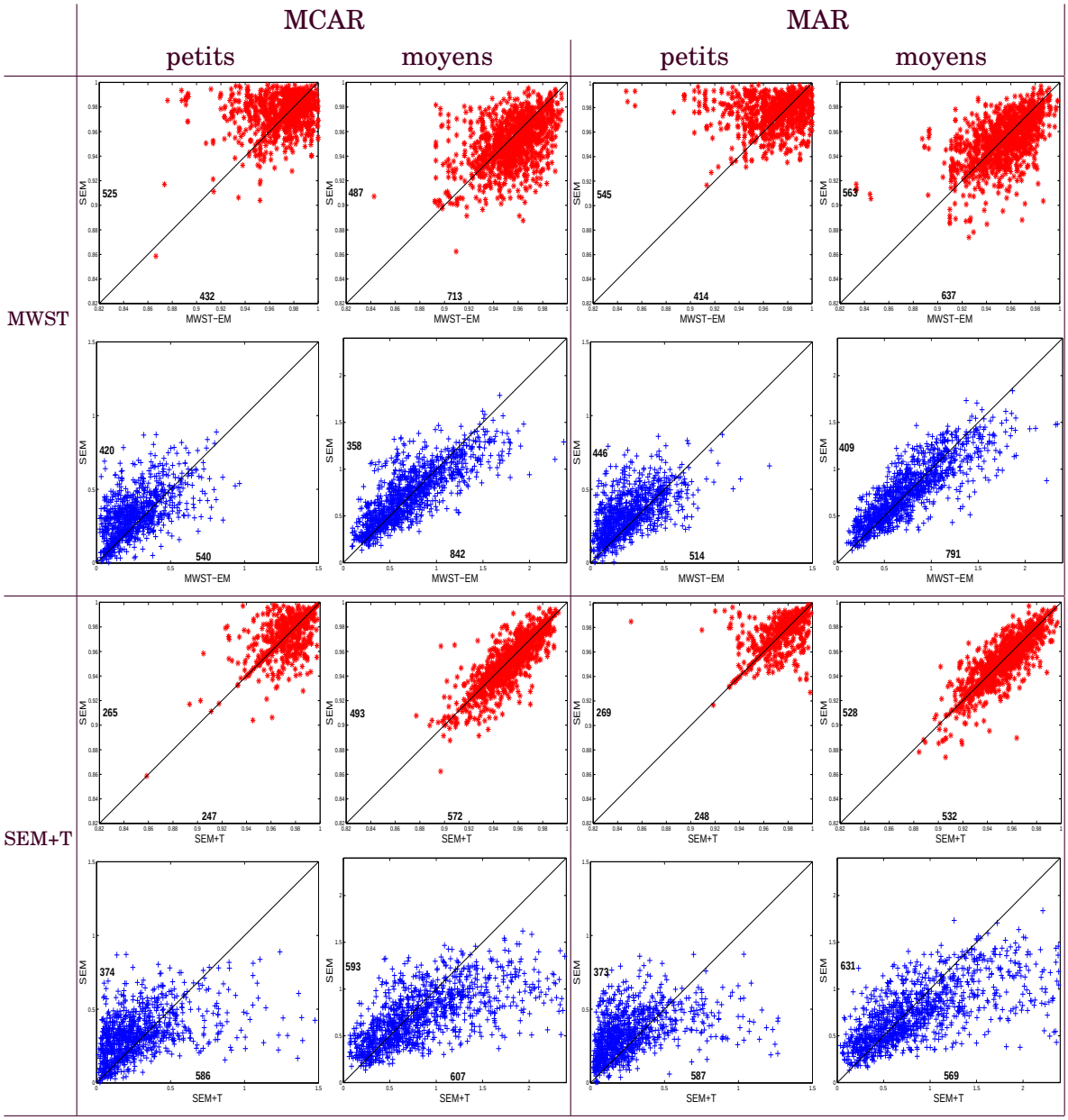
Du point de vue du score *BIC*, la méthode SEM+T est cependant meilleures que SEM. Elles sont alors moins stables que les méthodes utilisant ACA mais restent très efficaces.

Remarquez bien que ce constat est à inverser pour le critère de la divergence de KL pour lequel les méthodes utilisant EM sont plus efficaces lorsque le taux de données manquantes est supérieur à 35%.

14.1.3.5 Duels entre les différentes méthodes gloutonnes

Dans la table 14.2, nous pouvons observer quelques duels entre les méthodes MWST et SEM d'une part, et les méthodes SEM+T et SEM d'autre part.

Nous remarquons que sur les *petits exemples*, c'est-à-dire ceux possédant une structure proche d'une structure arborescente, la méthode MWST-EM parvient à obtenir plus régulièrement des scores *BIC* et des divergence de Kullback-Leiber meilleurs que la méthode SEM, et cela, que les données soient MCAR ou MAR.



TAB. 14.2 : *Duels MWST-EM vs SEM et SEM+T vs SEM : Score* $(BIC\ réseau\ original)/BIC$ *(en rouge, ligne supérieure) et divergence de* KL *(en bleu, ligne inférieure) pour les problèmes de petite difficulté (Jouet1, Jouet2, Jouet3, Jouet4) et ceux de difficulté moyenne (Jouet 5, Asia, Jouet6, Jouet7, Jouet8). Les chiffres représentent le nombre de bases pour lesquelles la méthode a pris l'avantage strictement.*

Ce bon résultat est partagé par la méthode SEM+T qui bat la méthode SEM sur les petits exemples. Néanmoins, aucune différence significative n'est à noter pour les réseaux de tailles moyennes.

Par ailleurs, nous pouvons remarquer que la méthode MWST-EM bat la méthode SEM sur les problèmes de tailles moyennes très régulièrement. Cela est certainement dû au fait que les réseaux obtenus sont de faible complexité. Nous remarquons également qu'elle bat régulièrement la méthode SEM sur la divergence de Kullbak-Leiber, cela est doit être dû au nombre important de petites bases d'exemples (de moins de 600 points) par rapport au nombre total de bases testées.

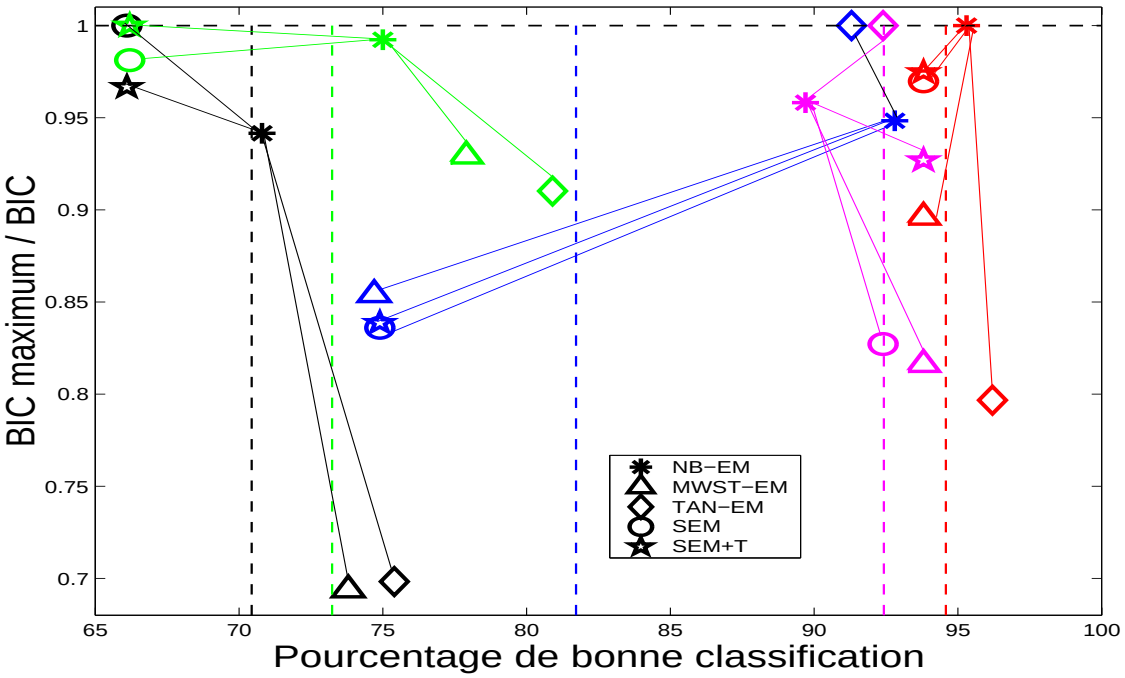


FIG. 14.4 : *Schéma représentant le taux de classification pour chaque méthode testée sur les bases d'exemples utilisées. En ordonnées, nous inscrivons à titre indicatif la puissance BIC, et en abscisses, le taux de classification sur les bases de tests. Chaque couleur représente une base (les points sont reliés) et les moyennes de classification sur chaque base sont indiqués par les traits verticaux.*

14.2 Recherche d'un bon classifieur

14.2.1 Techniques utilisées et techniques d'évaluation

Nous avons testé les algorithmes suivants en classification :

- NB-EM, qui consiste en un apprentissage EM paramétrique (algorithme 2 page 57) pour une structure de Bayes naïve,
- MWST-EM avec comme racine le nœud classe et le score Q^{BIC} (section 12.2.1),
- TAN-EM avec le score Q^{BIC} (section 12.2.2.1)
- SEM initialisé avec une chaîne et utilisant le score Q^{BIC} (section 12.1.1),
- SEM+T, c'est-à-dire SEM initialisé par la structure obtenue par MWST-EM (section 12.2.3).

Pour les expériences, nous avons utilisé des bases d'exemples incomplètes disponibles sur le site de l'UCI de Blake & Merz (1998). Ces bases sont décrites dans l'annexe E et possèdent entre 17 et 28 attributs pour des tailles d'apprentissage entre 90 et 5416 exemples et des taux d'exemples incomplets allant de $8,4\%$ à 88%.

Pour l'évaluation, nous avons considéré le taux de bonne classification sur des bases de test ainsi que la puissance *BIC* (score *BIC* de la méthode ayant permis d'obtenir le meilleur score *BIC* divisé par le score *BIC* de la méthode testée) des modèles obtenus.

14.2.2 Résultats et interprétations

Les résultats sont synthétisés sur la figure 14.4 et la table 14.3. Ils sont également donnés en détail dans la table F.7 de l'annexe F.3.

méthodes	NB-EM	MWST-EM	TAN-EM	SEM	SEM+T
indice de temps de calcul moyen	1.00	2.57	2.56	48.89	33.00
taux de classification moyen	84.72	82.80	87.24	78.68	78.96

TAB. 14.3 : *Temps de calcul et taux de classification moyens obtenus sur les problèmes considérés.*

Le classifieur de Bayes naïf donne toujours de bons résultats. Les autres méthodes peuvent permettre d'obtenir de meilleures performances en classification mais cela n'est pas systématique comme nous pouvons le voir sur la figure 14.4. Mais là où chaque exécution de NB-EM donne les mêmes résultats, les méthodes TAN-EM et SEM nécessitent une initialisation : la racine du sous-arbre pour TAN-EM et la chaîne de départ pour SEM. Nous avons choisi de ne garder que les meilleurs résultats pour ces méthodes. Néanmoins, nous avons remarqué que la méthode TAN-EM est très stable quant au taux de classification final, alors que la méthode SEM donne des résultats plus variables.

La méthode MWST-EM peut obtenir de bons résultats en classification même si, dans la structure finale, le nœud classe n'est connecté qu'à deux autres nœuds. Cette méthode fait donc office de sélection de variables en prenant d'avantage en compte les deux variables les plus pertinentes pour l'état de la variable classe, puis les suivantes lorsque ces dernières ne sont pas observées. Cette stratégie s'est avérée gagnante sur la base d'exemples HOUSE.

Pour nos tests, TAN-EM a toujours donné de très bons taux de classification par rapport aux autres méthodes d'apprentissage testées. Même si la méthode TAN-EM ne permet pas systématiquement d'obtenir de bons score BIC étant donné la complexité des structures obtenues, elle obtient tout de même deux fois le meilleur score *BIC*. Par ailleurs, nous avons remarqué qu'elle permet d'approcher particulièrement bien la distribution sous-jacente des bases d'exemples malgré la contrainte très forte sur la forme de la structure à obtenir (tableau F.7).

Par ailleurs, nous pouvons nous rendre compte à la lecture de la table 14.3 que les méthodes TAN-EM et MWST-EM ont des complexités très proches et très faibles comparativement à la méthode SEM. La méthode SEM+T permet d'économiser un tiers du temps de calcul pour des performances similaires. Il apparaît que la méthode TAN-EM permet d'obtenir les meilleurs taux de classification moyens dans le temps record de deux ou trois EM paramétriques là où la méthode SEM en effectue presqu'une cinquantaine pour des problèmes possédant environ 22 attributs.

Conclusion

Dans cette partie, nous avons proposé une technique d'apprentissage de structure à partir de bases d'exemples incomplètes dans l'espace des structures arborescentes (MWST-EM) et l'avons comparée à la principale technique d'apprentissage de structure existante dans ce cadre (SEM). Nous avons également proposé d'utiliser le résultat de MWST-EM pour initialiser SEM. Nous avons alors comparé ces méthodes d'apprentissage avec leur version utilisant les exemples complets (CCA) et leur version utilisant les exemples disponibles (ACA).

Nous avons également proposé un processus de modélisation et de génération de données manquantes MCAR et MAR ce qui nous a permis de tester les méthodes sur ces deux types de données incomplètes.

Nous avons alors remarqué que CCA n'est pas une bonne méthode dès que le taux de données incomplètes est supérieur à 10%. D'une part, cette méthode est limitée dans son application car il est fréquent qu'une base d'exemples contienne trop peu de cas complets pour que l'apprentissage soit efficace.

Par contre, les différences entre l'utilisation de ACA et de EM sont plus fines. En résumé, nous dirons que la méthode ACA mène à des résultats ayant de meilleurs scores BIC et la méthode EM mène à des résultats avec de très bonnes divergences de Kullback-Leiber. Plus précisément, la méthode ACA est étonnamment efficace lorsque le nombre d'exemples de la base est faible tandis que la méthode EM, bien qu'itérative, devient plus efficace lorsque le nombre de cas augmente. La méthode EM est alors très stable en fonction du taux de données incomplètes alors que la méthode ACA, même si elle permet d'obtenir de bons résultats lorsque ce taux est faible, devient rapidement instable lorsque ce taux augmente.

Toutes les méthodes testées se comportant de manière similaire, que les données manquantes soient MCAR ou MAR.

Notre proposition MWST-EM donne les meilleurs résultats lorsque la taille de la base d'exemples est faible. La méthode SEM prend l'avantage lorsque la taille devient grande. Ce résultat est dû au fait que la méthode MWST-EM obtient des résultats très stables dans toutes situations alors que la qualité des résultats obtenus par SEM augmente régulièrement avec la taille de la base.

La méthode MWST-EM permet également d'obtenir des résultats similaires à ceux obtenus par SEM lorsque le taux de données manquantes est élevé. Les structures obtenues étant moins complexes, donc possédant moins de paramètres, il est alors possible d'apprendre ces paramètres plus finement.

L'utilisation de la structure obtenue par MWST-EM pour initialiser SEM n'est étonnamment pas aussi efficace que lorsque les données sont complètes. Sur les données synthétiques aucune différence significative n'a été remarquée. La méthode SEM semble être finalement plus efficace. Par contre, sur les problèmes de classification, nous avons tout de même constaté un gain de temps de calcul important pour la méthode SEM+T par rapport à la méthode SEM, avec des performances similaires.

Quatrième partie

Conclusion et Perspectives

15

Conclusion

Dans la première partie de ce manuscrit, nous avons mis en place le cadre des modèles graphiques probabilistes avant de décrire les différentes variantes de modèles graphiques existantes. Que ces derniers soient orientés, dynamiques, causaux, ou construits spécifiquement pour la décision, chacun peut être amené à les utiliser pour la modélisation. Nous avons ensuite passé en revue les principales méthodes d'inférence et d'apprentissage des paramètres associées aux réseaux bayésiens.

Dans la deuxième partie, nous avons étudié les différentes techniques d'apprentissage de structure de réseaux bayésiens à partir de données complètes. Nous avons exposé les principales méthodes d'identification de structure à base de tests d'indépendance conditionnelle. Ensuite, après avoir introduit la notion de score pour les réseaux bayésiens, nous avons présenté les principales méthodes *état de l'art* utilisant ces scores. Après avoir remarqué que certaines méthodes souffraient d'un problème d'initialisation, nous avons proposé une technique utilisant une structure arborescente optimale pour résoudre ce problème. Seules des comparaisons ponctuelles de ces algorithmes existent, nous avons donc décidé de comparer de nombreuses techniques d'apprentissage de structure de réseaux bayésiens et nos propositions d'initialisation expérimentalement à la fois sur des problèmes de découverte de structures connues et sur des problèmes de classification.

Nous en avons alors déduit que la méthode GES permettait effectivement d'obtenir de meilleures performances que les autres méthodes. La méthode de recherche d'une structure arborescente optimale MWST est particulièrement stable en toute situation. Cette stabilité lui permet notamment d'obtenir les meilleurs résultats lorsque la taille de la base d'exemples est faible. Nous avons également observé que notre proposition d'utiliser le résultat de cette méthode pour initialiser d'autres techniques d'apprentissage comme K2 ou GS permettait effectivement de rendre ces méthodes plus stables tout en les rendant soit plus efficaces, soit plus rapides pour des performances comparables. Nous avons de plus observé que sur des tâches de classification, l'utilisation de méthodes à base d'identification de dépendances conditionnelles est peu performante tandis qu'une méthode très rapide comme l'utilisation d'une variante de MWST pour obtenir une structure de Bayes naïve augmentée permet d'obtenir les meilleures performances en classification en utilisant un temps d'apprentissage record.

Toutes ces techniques d'apprentissage étant dédiées aux bases de cas complètes spécifiquement, la troisième partie de nos travaux a consisté en l'adaptation de certaines de ces méthodes à l'apprentissage de structure de réseaux bayésiens à partir de bases d'exemples incomplètes. Après avoir assimilé les principes de l'algorithme EM et leur adaptation au calcul de score de structure par rapport à une base incomplète, nous avons proposé une technique permettant d'apprendre une structure de réseau bayésien arborescente à partir de bases d'exemples incomplètes.

Pour tester cette méthode, nous avons proposé un formalisme de modélisation permettant de générer artificiellement des bases d'exemples incomplètes vérifiant les hypothèses MCAR ou MAR. Cette technique permet également de générer certaines formes de données NMAR ; Après avoir généré de nombreuses bases d'exemples, nous avons comparé notre méthode MWST-EM à la technique *état de l'art* SEM. Cette technique souffrant également de difficultés d'initialisation, nous avons également proposé d'utiliser le résultat de notre méthode MWST-EM pour initialiser SEM.

Nous avons aussi comparé ces différentes techniques d'apprentissage utilisant l'algorithme EM à des techniques similaires utilisant les exemples complets (CCA) ou les exemples disponibles (ACA). Les expérimentations faites ont alors permis de conclure que toutes les méthodes testées sont robustes, que le type des données manquantes soit MCAR ou MAR. Nous nous sommes également rendu compte que les méthodes utilisant CCA n'étaient pas aussi performantes que celles utilisant ACA ou EM pour les taux de données manquantes testés. Nous avons alors été étonnés de remarquer l'efficacité des méthodes utilisant CCA qui permettent de modéliser correctement la loi jointe lorsque le taux de données incomplètes est faible, mais qui deviennent rapidement instables lorsque celui-ci augmente.

Par ailleurs, contrairement au cas des données complètes, utiliser le résultat de MWST-EM pour initialiser SEM apporte peu à cette dernière méthode si ce n'est un gain de temps de calcul pour des performances égales lorsque les données sont de grande dimension. Par contre, la méthode MWST-EM étant particulièrement stable, elle permet d'obtenir les meilleurs résultats lorsque la taille de la base d'exemples est faible. Elle est ensuite dépassée par la méthode SEM qui utilise un espace de parcours plus général. Néanmoins l'adaptation de la méthode MWST à l'appentissage d'une structure de Bayes naïve augmentée à partir de bases d'exemples incomplètes a permis d'obtenir les modèles avec les plus forts pouvoirs discriminants et cela, avec un temps de calcul très faible pour une méthode itérative.

Durant cette thèse, nous avons mis en évidence les cas où l'utilisation d'une structure arborescente permet d'obtenir de bons résultats pour la modélisation, pour l'initialisation d'autres méthodes d'apprentissage ou pour la classification, que les données d'apprentissage soient complètement observées ou non.

16

Perspectives

Perspectives à court terme

Ces dernières années, différentes techniques d'apprentissage de structure de réseaux bayésiens à partir de bases d'exemples complètes ont vu le jour. Nous avons implémenté et testé certaines de ces méthodes. Il reste néanmoins à tester de nombreuses autres techniques, en particulier parmi les méthodes d'apprentissage de structure à partir de recherche d'indépendance conditionnelle, et étendre ces dernières aux bases d'exemples incomplètes. Une extension de la phase de tests pourrait alors être considérée, pour, par exemple, tester ces différentes méthodes avec des données de plus grande dimension et des problèmes de classification plus variés.

Les perspectives possibles en rapport avec le développement de la méthode MWST-EM sont de passer de l'espace des arbres à des espaces plus généraux comme ceux des forêts ou des équivalents de Markov.

Utiliser une structure de forêt possède l'avantage de pouvoir déconnecter certaines variables des autres, au contraire d'une approche utilisant une structure arborescente. Elle pourra donc plus facilement identifier les variables trop bruitées ou non pertinentes pour le problème traité. Une telle représentation par une structure de forêt pourrait s'avérer être une meilleure base de départ pour initialiser une méthode gloutonne comme SEM et cela reste à tester.

Nous pourrions également comparer cette dernière méthode à une extension de l'algorithme GES aux bases d'exemples incomplètes.

Par ailleurs, augmenter une structure de Bayes naïve par une forêt permet d'incorporer des indépendances entre les observations qui ne peuvent pas exister lorsque la structure naïve est augmentée par une structure arborescente.

La méthode EM demande de nombreux calculs et sa convergence n'est prouvée que dans le cadre des données incomplètes vérifiant les hypothèses MCAR ou MAR. L'utilisation de méthodes plus robustes (*cf.* Ramoni & Sebastiani (2000)) dans le cadre plus géréral des données manquantes NMAR doit être considérée pour traiter plus efficacement les problèmes réels.

Il faudrait également adapter la plupart des méthodes d'apprentissage aux bases de données mixtes, ces dernières représentant la majorité des bases disponibles réelles. Des extensions utilisant des distributions gaussiennes (ou des mixtures de gausiennes : Davies & Moore (2000b)) ont déjà été explorées, mais dans la majorité des cas, il s'agit de traiter exclusivement des données continues.

Des extensions aux données mixtes utilisant des distributions à base d'exponentielles tronquées ou des fonctions noyaux peuvent aussi voir le jour.

De manière connexe, nous pourrions mettre en oeuvre une extension de la modélisation des processus de génération de données incomplètes que nous avons proposée aux données continues, puis au données mixtes.

Pour nos tests, nous avons considéré l'hypothèse de suffisance causale et n'avons donc pas cherché à détecter des variables latentes. Les modèles utilisant des variables cachées ont prouvé leur efficacité dans de multiples domaines. Certaines approches considèrent des apprentissages de modèles hiérarchiques avec variables cachées comme dans Zhang (2002). Mais peu de techniques proposent d'effectuer un apprentissage d'un modèle général utilisant abondamment les variables cachées, comme par exemple l'utilisation d'un opérateur qui transformerait une variable en un duo (variable, variable cachée) comme cela est fait dans les chaînes de Markov cachées.

Perspectives à moyen et long termes

Nous avons vu qu'il existe de nombreuses variations des réseaux bayésiens. Par exemple, les diagrammes d'influence, ou leur version dynamique, les processus de décision markoviens, sont adaptés aux problématiques liées à la décision. Dans ces modèles, il existe également des nœuds d'action où l'utilisateur choisit une décision, et les nœuds utilités, où le coût d'une décision est évalué. Nous pourrions adapter les méthodes d'apprentissage de structure à l'apprentissage automatique des liens entre les variables décisions, les variables aléatoires et les variables utilités.

Dans un objectif de modélisation temporelle, les chaînes de Markov et autres réseaux bayésiens dynamiques (avec ou sans variables latentes) sont des outils très puissants de modélisation et d'inférence. Ces modèles sont très complexes, leur compléxité étant fonction du nombre de variables et du nombre de tranches de temps utilisées. Ces modèles nécessitent alors l'utilisation d'hypothèses particulières pour leur apprentissage. En particulier, dans le cadre des 2TBN (RB à deux tranches), le modèle a une partie statique modélisant les dépendances entre les variables à un instant donné et possède également une partie dynamique pour modéliser les dépendances temporelles. Les techniques d'apprentissage doivent donc être adaptées à ces apprentissages de structure *intra-tranche* et *inter-tranches*.

La plupart des techniques actuelles proposent de n'apprendre qu'une structure représentant des indépendances conditionnelles entre les variables. Néanmoins dans de nombreux problèmes réels, des dépendances causales entre les différents attributs doivent être considérées. Meganck, Leray & Manderick (2006) ont commencé à étudier cette problématique.

Il peut également être intéressant de considérer des phénomènes de dépendances (probabilistes ou causales), non plus entre les attributs, mais entre les états des attributs. Comme l'ont fait remarquer Cheng & Greiner (2001), lorsqu'une variable (par exemple la classe) entre dans un état particulier, les dépendances entre les autres variables peuvent être modifiées. Ils proposent alors de construire différents réseaux en fonction des différentes valeurs de cet attribut.

Seulement, leur méthode ne permet pas de lier les états entre eux. Par ailleurs, il peut parfois arriver que la conjonction de l'état de deux variables ait une influence très forte sur l'état d'une tierce variable, or, avec les méthodes d'apprentissage actuelles, ce cas n'est pas correctement appris car seules sont observées les relations entre les variables et non entre les états de ces variables. Aussem et al. (2006) ont ouvert la voie pour étudier cette problématique. Dans l'exemple précédent, la notion de causalité est importante, or, les techniques actuelles se concentrent sur les relations probabilistes. La spécialisation de ces méthodes à l'identification de réseaux bayésiens causaux est une piste naissante. Pour traiter ce type de phénomènes, nous pouvons imaginer utiliser des variables bouléennes pour représenter la présence des différents états de chaque variable, seulement, une telle technique augmenterait de manière trop importante le nombre de variables. Nous pourrions également imaginer d'utiliser un nouvel opérateur permettant de séparer une variable en deux, la variable originale et une spécialisation de cette variable sur certains états qui lui seraient alors liés, de manière à pouvoir plus spécifiquement retrouver les liens entre certains états de cette variable à d'autres variables et seulement ces états.

Jusqu'à présent nous n'avons considéré que les réseaux bayésiens ; or, de nombreux modèles probabilistes ne sont pas forcément complètement orientés. En particulier les champs de Markov qui sont de plus en plus utilisés. Les idées de certaines méthodes d'apprentissage de réseaux bayésiens pourraient sûrement être adaptées à l'apprentissage de modèles non orientés.

Par ailleurs, le formalisme des modèles graphiques semi-orientés possède maintenant quelques méthodes d'inférences mais aucune méthode d'apprentissage spécifique n'a (à ma connaissance) été proposée pour ces modèles jusqu'à présent.

Références Bibliographiques

J'aimerais conclure en disant que
comme toute création n'est pas un travail mais un patchwork,
n'est pas une influence mais influencée,
n'est pas originale mais un amalgame d'originalités,
qu'il faut être contre la propriété intellectuelle.
Un retour en arrière (certains disent une avancée)
sur la notion de propriété,
qui gangrène la communauté humaine depuis plusieurs millénaires,
est souhaitable.

J'aime à penser que, un jour,
la connaissance et la culture, au moins,
seront pleinement libres et accessibles.

Les références bibliographiques sont suivies des pages sur lesquelles elles sont citées.

Acid, S. & de Campos, L. (2001). A hybrid methodology for learning belief networks : BENEDICT. *International Journal of Approximate Reasoning*, *27*, 235–262. 96

Aeberhard, S., Coomans, D., & de Vel, O. (1992). Comparison of classifiers in high dimensional settings. Technical Report 92-02, Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland. 213

Akaike, H. (1970). Statistical predictor identification. *Ann. Inst. Statist. Math.*, *22*, 203–217. 81, 82

Akaike, H. (1973). Information theory and extension of the maximum likelihood principle. Dans B.N.Petrov & (eds.), F. (Eds.), *Proceedings of the 2nd International Symposium of Information Theory*, (pp. 267–281)., Akademiai Kiado, Budapest. 81

Akaike, H. (1979). A bayesian extention of the minimum AIC procedure of autoregressive model. *Biometrika*, *66*(1), 237–242. 82

Alimoglu, F. & Alpaydin, E. (1996). Methods of combining multiple classifiers based on different representations for pen-based handwriting recognition. Dans *Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN 96)*, Istanbul, Turkey. 212

Andersen, S., Olesen, K., Jensen, F., & Jensen, F. (1989). Hugin - a shell for building bayesian belief universes for expert systems. *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, *1*, 1080–1085. http://www.hugin.com/. 210

Anderson, B. & Moore, J. (1979). *Optimal Filtering*. Prentice-Hall. 20

Anderson, S. A., Madigan, D., & Perlman, M. D. (2001). Alternative markov property for chain graphs. *Scandinavian journal of statistics*, *28*, 33–86. 27

Androutspoulos, I., Palouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., & Stamatopoulos, P. (2000). Learning to filter spam e-mail : A comparaison of a naive bayesian and a memory-based approach. Technical Report DEMO 2000/5, Dept. of Informatics, University of Athens. 22

Asmussen, S., Nerman, O., & Olsson, M. (1996). Fitting phase-type dist via the EM algorithm. *Scandinavian Journal of Statistics*, *23*, 419–441. 19

Aussem, A., Kebaili, Z., Corbex, M., & Marchi, F. (2006). Apprentissage de la structure de réseaux bayésien à partir des motifs fréquents corrélés : application à l'identification des facteurs environnementaus du cancer du nasopharynx. Dans *Actes des journée Extraction et Gestion de Connaissances (EGC'06), Revue des Nouvelles Technologies de l'Information (RNTI-E-6)*, (pp. 651–662). Cépaduès-Editions. 73, 175

Auvray, V. & Wehenkel, L. (2002). On the construction of the inclusion boundary neighbourhood for markov equivalence classes of bayesian network structures. Dans Darwiche, A. & Friedman, N. (Eds.), *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-02)*, (pp. 26–35)., S.F., Cal. Morgan Kaufmann Publishers. 93

Bach, F. & Jordan, M. (2005). Discriminative training of hidden markov models for multiple pitch tracking, , 2005. Dans *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, (pp. 489–492). 18

Badea, L. (2003). Inferring large gene networks from microarray data : a constraint-based approach. Dans *IJCAI-2003 Workshop on Learning Graphical Models for Computational Genomics*. 72

Bangsø, O. & Wuillemin, P.-H. (2000). Object oriented bayesian networks a framework for top-down specification of large bayesian networks and repetitive structures. Technical Report CIT-87.2-00-obphw1, Hewlett-Packard Laboratory for Normative Systems, Aalborg University. 16

Bauer, E., Koller, D., & Singer, Y. (1997). Update rules for parameter estimation in bayesian networks. Dans *Proceedings of Uncertainty in Artificial Intelligence (UAI'97)*, (pp. 3–13). 57

Bayes, T. (1764). An essay toward solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, *53*, 370–418. [Fascimile available online : the original essay with an introduction by his friend Richard Price http://www.stat.ucla.edu/history/essay.pdf]. 9, 13

Beal, M. & Ghahramani, Z. (2003). The variational bayesian EM algorithm for incomplete data : with application to scoring graphical model structures. *Bayesian Statistics*, *7*, 453–464. Oxford University Press. 128

Bellot, D. (2002). *Fusion de données avec des réseaux bayésiens pour la modélisation des systèmes dynamiques et son application en télémédecine*. PhD thesis, Université Henri Poincaré, Nancy 1, France. 18

Ben Amor, N., Benferhat, S., & Mellouli, K. (2002). Un algorithme de propagation pour les réseaux possibilistes basés sur le conditionnement ordinal. Dans *13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et d'Intelligence Artificielle (RFIA 2002)*, (pp. 339–348)., Angers, France. 6, 25

Benferhat, S., Dubois, D., Kaci, S., & Prade, H. (2001). Graphical readings of possibilisitc logic bases. Dans Morgan Kaufmann (Ed.), *17th Conference Uncertainty in Artificial Intelligence (UAI01)*, (pp. 24–31)., Seattle. 6

Bennani, Y. & Bossaert, F. (1996). Predictive neural networks for traffic disturbance detection in the telephone network. Dans *Proceedings of IMACS-CESA'96*, Lille, France. 219, 222

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). Dans *Journal of Royal Statistical Society*, volume 36 of *series B*, (pp. 192–326). 26

Biernacki, C., Celeux, G., & Govaert, G. (1998). Assessing a mixture model for clustering with integrated classification likelyhood. Technical Report rapport de recherche 3521, INRIA. 82

Binder, J., Murphy, K., & Russell, S. (1997). Space-efficient inference in dynamic probabilistic networks. Dans *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI'97)*, (pp. 1292–1296). Morgan Kaufmann Publishers. 18

Blake, C. & Merz, C. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html. 164, 211

Bohanec, M. & Rajkovic, V. (1990). Expert system for decision making. *Sistemica*, *1*(1), 145–157. 211

Borgelt, C. & Kruse, R. (2002). *Graphical Models - Methods for Data Analysis and Mining*. John Wiley & Sons, Chichester, United Kingdom. 80

Bøttcher, S. & Dethlefsen, C. (2004). Deal : A package for learning bayesian networks. http://www.r-project.org/gR/. 209

Bouckaert, R. R. (1993). Probabilistic network construction using the minimum description length principle. *Lecture Notes in Computer Science*, *747*, 41–48. 83, 90

Bouillaut, L., François, O., Leray, P., & Aknin, P. (2007). Statavaries : Analyse statistique du traitement du rail cassé pour les ligne fer du réseau ratp. Contrat d'étude StatAvaries INRETS-LTN / RATP.

Bouleau, N. (1999). Modèles probabilistes ou modèles déterministes, le cas du changement global. Matapli-SMAI n^{o}58. 197

Boutilier, C., Dean, T., & Hanks, S. (1999). Decision-theoretic planning : structural assumptions and computational leverage. *Journal of Artificial Intelligence research*, *11*, 1–94. 21

Bradski, G. (2004). Open source probabilistic network library. Systems Technology Labs, Intel. http://www.intel.com/research/mrl/pnl/. 209

Brown, L., Tsamardinos, I., & Aliferis, C. (2005). A comparison of novel and state-of-the-art polynomial bayesian network learning algorithms. Dans *Proceedings of the Twentieth National Conference on Artificial Intelligence*, (pp. 739–745)., Pittsburgh, Pennsylvania. AAAI Press, Menlo Park, California. 72

Buntine, W. (1991). Theory refinement on bayesian networks. Dans D'Ambrosio, B., Smets, P., & Bonissone, P. (Eds.), *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, (pp. 52–60)., San Mateo, CA, USA. Morgan Kaufmann Publishers. 80

Cartwright, N. (1979). Causal laws and effective strategies. *NoÃ »s journal, Special Issue on Counterfactuals and Laws*, *13*(4), 419–437. 200

Castelo, R. & Kocka, T. (2002). Towards an inclusion driven learning of bayesian networks. Technical Report UU-CS-2002-05, Institute of information and computing sciences, University of Utrecht. 93

Castillo, E., Gutiérrez, J. M., , & Hadi, A. S. (1996). A new method for symbolic inference in bayesian networks. *Networks*, *28*, 31–43. 45

Celeux, C. & Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, *14*(3), 315–332. 84, 127

Chavez, M. & Cooper, G. (1990). A randomized approximation algorithm for probabilistic inference on bayesian belief networks. *Networks*, *20*(5), 661–685. 145

Cheeseman, P. & Stutz, J. (1996). Bayesian classification (AUTOCLASS) : Theory and results. Dans U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 153–180). AAAI Press/MIT Press. 131

Cheng, J. & Druzdzel, M. (2000). Ais-bn : An adaptive importance sampling algorithm for evidential reasoning in large bayesian networks. *Journal of Artificial Intelligence Research*, *13*, 155–188. 145

Cheng, J. & Greiner, R. (2001). Learning bayesian belief network classifiers : Algorithms and system. Dans *Proceedings of the Canadian Conference on AI 2001*, volume 2056, (pp. 141–151). 22, 174

Cheng, J., Greiner, R., Kelly, J., Bell, D., & Liu, W. (2002). Learning Bayesian networks from data : An information-theory based approach. *Artificial Intelligence*, *137*(1-2), 43–90. 72, 73

Cheng, J., Hatzis, C., Hayashi, H., Krogel, N., Morishita, S., Page, D., & Sese, J. (2001). Kdd cup 2001 repport. the awards ceremony of the 7th ACM SIGKDD 2001, pp 47-64 (San Francisco, CA). http://www.cs.ualberta.ca/~jcheng/bnsoft.htm. 209

Chernoff, H. & Lehmann, E. (1954). The use of maximum likelihood estimates in χ^2 tests for goodness-of-fit. *Annals of Mathematical Statistics*, *25*, 579–586. 68

Chickering, D. (1996). Learning equivalence classes of bayesian network structures. Dans Horvitz, E. & Jensen, F. (Eds.), *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI-96)*, (pp. 150–157)., San Francisco. Morgan Kaufmann Publishers. 40, 80

Chickering, D. (2002a). Learning equivalence classes of bayesian-network structures. *Journal of machine learning research*, *2*, 445–498. 93

Chickering, D. (2002b). Optimal structure identification with greedy search. *Journal of Machine Learning Research*, *3*, 507–554. 94, 95, 142

Chow, C. & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, *14*(3), 462–467. 70, 88

Cobb, B. & Shenoy, P. (2005). Inference in hybrid bayesian networks with mixtures of truncated exponentials. *International Journal of Approximate Reasoning*, *41(3)*, 257–286. 38

Cobb, B. & Shenoy, P. (2006). Approximating probability density functions in hybrid bayesian networks with mixtures of truncated exponentials. *Statistics and Computing*, *16*(3), 293–308. 96

Cohen, I., Bronstein, A., & Cozman, F. (2001). Adaptive online learning of bayesian network parameters. Technical Report HPL-2001-156, Hewlett Packard Labs. 127

Cohen, I., Cozman, F. G., Sebe, N., Cirelo, M. C., & Huang, T. S. (2004). Semisupervised learning of classifiers : Theory, algorithms, and their application to human-computer interaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26(12)*, 1553–1568. 141

Cooper, G. (1992). The computational complexity of probabilistic inference using bayesian belief networks. *Artificial intelligence*, *42*, 393–347. 45

Cooper, G. & Hersovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Maching Learning*, *9*, 309–347. 78, 89

Cotta, C. & Muruzabál, J. (2004). On the learning of bayesian network graph structures via evolutionary programming. Dans Lucas, P. (Ed.), *Proceedings on the Second European Workshop on Probabilistic graphical Models*, (pp. 65–72)., The Nederlands. 95

Darwiche, A. (2000). A differential approach to inference in bayesian networks. Dans *Proceedings of Uncertainty in Artificial Intelligence*, (pp. 123–132). 45

Darwiche, A. & Provan, G. (1997). Query dags : A practical paradigm for implementing belief-network inference. *Journal of Artificial Intelligence Research*, *6*, 147–176. 45

Dash, D. & Druzdzel, M. (1999). A hybrid anytime algorithm for the construction of causal models from sparse data. Dans *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI99)*, (pp. 142–149). 71, 96

Dash, D. & Druzdzel, M. (2003). Robust independence testing for constraint-based learning of causal structure. Proceedings of The Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI03), pp 167-174. 138

Davies, S. & Moore, A. (2000a). Mix-nets : Factored mixtures of gaussians in bayesian networks with mixed continuous and discrete variables. Dans Boutilier, C. & Goldszmidt, M. (Eds.), *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-00)*, (pp. 168–175)., SF, CA. Morgan Kaufmann Publishers. 96

Davies, S. & Moore, A. (2000b). Mixnets : Learning bayesian networks with mixtures of discrete and continuous attributes. Dans *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*. AAAI Press. 173

de Campos, L., Fernández-Luna, J., Gámez, J., & Puerta, J. (2002). Ant colony optimization for learning bayesian networks. *International Journal of Approximate Reasoning*, *31*, 291–311. 96

de Campos, L. & Huete, J. (1999). Approximating causal orderings for bayesian networks using genetic algorithms and simulated annealing. Technical Report DECSAI-990212, Research Group of Uncertainty Treatment in Artificial Intelligence, University of Granada. 90

Dean, T. & Kanazawa, K. (1989). A model for reasonning about persistence and causation. *Computationnal Intelligence, 5(3)*, *1*, 142–150. 18

Deason, W., Brown, D., Chang, K., & Cross II, J. (1991). A rule-based software test data generator. *IEEE transactions on Knowledge and Data Engineering*, *3(1)*, 108–117. 144

Dechter, R. (1998). Bucket elimination : a unifying framework for structure-driven inference. Technical report, Dept. of Computer and Information Science, University of California, Irvine, USA. 44

Delaplace, A., Brouard, T., & Cardot, H. (2006). Two evolutionary methods for learning bayesian network structures. Dans *International Conference on Computational Intelligence and Security (CIS 2006)*, Guangzhou, China. Springer Verlag. 96

Dembo, A. & Zeitouni, O. (1986). Parameter estimation of partially observed continuous time stochastic processes via the em algorithm. *Stochastic Processes and their Applications*, *23*, 91–113. 19

DeMillo, R. & Offutt, J. (1991). Constraint-based automatic test data generation. *IEEE Transactions on Software Engineering*, *17(9)*, 900–910. 144

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, *B 39*, 1–38. 56, 124

Diard, J. (2003). *La carte bayésienne – Un modèle probabiliste hiérarchique pour la navigation en robotique mobile*. Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, France. 30

Domingos, P. (1997). Why does bagging work ? a bayesian account and its implications. Dans in D. Heckerman, H. Mannila, D. P. & R. Uthurusamy, A. P. (Eds.), *Proceedings of the third international conference on Knowledge Discovery and Data Mining*, (pp. 155– 158). 58

Domingos, P. & Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, *29*, 103–130. 88

Dor, D. & Tarsi, M. (1992). A simple algorithm to construct a consistent extension of a partially oriented graph. Technical Report R-185, Cognitive Systems Laboratory, UCLA Computer Science Department. 40

Drakos, N. & Moore, R. (1998). Javabayes, bayesian networks in java : User manual and download. Available at URL : http://www-2.cs.cmu.edu/~javabayes/Home/. 209

Dugad, R. & Desai, B. (1996). A tutorial on hidden markov models. *Published Online, http ://vision.ai.uiuc.edu/dugad/*. 18

El Matouat, F., Colot, O., Vannoorenberghe, P., & Labiche, J. (2000). From continuous to discrete variables for bayesian network classifiers. Dans *Conference on Systems, Man and Cybernetics, IEEE-SMC*, (pp. 2800–2805)., Nashville, USA. 63

Ferguson, R. & Korel, B. (1996). The chaining approach for software test data generation. *ACM TOSEM*, *5(1)*, 63–86. 144

Fine, S., Singer, Y., & Tishby, N. (1998). The hierarchical hidden markov model : Analysis and application. *Machine Learning*, *1*, 32–41. 20

François, O. & Leray, P. (2003). Etude comparative d'algorithmes d'apprentissage de structure dans les réseaux bayésiens. Dans Florence Dupin De Saint-Cyr (Ed.), *RJCIA2003 - 6emes Rencontres Nationales des Jeunes Chercheurs en Intelligence Artificielle, ISBN : 2-7061-1143-7*, (pp. 167–180). Presses Universitaires de Grenoble.

François, O. & Leray, P. (2004a). Etude comparative d'algorithmes d'apprentissage de structure dans les réseaux bayésiens. *Journal électronique d'intelligence artificielle*, *5*(39), 1–19. 9, 91, 92

François, O. & Leray, P. (2004b). Evaluation d'algorithmes d'apprentissage de structure pour les réseaux bayésiens. Dans *14ieme Congrès francophone de Reconnaissance des formes et d'Intelligence artificielle*, (pp. 1453–1460). 9, 91, 92

François, O. & Leray, P. (2005a). Applications of hierarchical latent classes models and bayesian structure learning from incomplete datasets. second Workshop on Applying Graphical Models (WAGM 2005), Aalborg, Danemark.

François, O. & Leray, P. (2005b). Apprentissage de structure dans les réseaux bayésiens et données incomplètes. Dans *Actes des journées Extraction et Gestion de Connaissances (EGC'05), Revue des Nouvelles Technologies de l'Information (RNTI-E-3), ISBN : 2-85428-677-4*, volume 1, (pp. 127–132)., Paris, France. Cépaduès-Editions. 10, 138

François, O. & Leray, P. (2006a). Apprentissage de structure à partir de données incomplètes et application à la classification. Troisièmes Journées Francophones sur les Réseaux Bayésiens (JFRB'06), Valenciennes, France.

François, O. & Leray, P. (2006b). Learning the tree augmented naive bayes classifier from incomplete datasets. Dans *Proceedings of the Third European Workshop on Probabilistic Graphical Models (PGM'06), ISBN : 80-86742-14-8*, (pp. 91–98)., Prague, Czech Republic. 10, 140, 141

François, O. & Leray, P. (2006c). Sur la robustesse de l'apprentissage de réseaux bayésiens à partir de bases d'exemples incomplètes. Septième congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2006), Lille, France.

François, O. & Leray, P. (2006d). Sur l'apprentissage de la structure réseaux bayésiens à partir de bases d'exemples incomplètes. Deuxièmes journées des Rencontres Inter-Associations : La classification et ses applications (RIAs 2006), Lyon, France.

François, O. & Leray, P. (2007). Incomplete datasets generation using the bayesian networks formalism. Dans *Proceedings of the International Joint Conferences on Neural Networks (IJCNN 2007)*, Orlando, Florida.

Freund, Y. & Schapire, R. (1996). Experiments with a new boosting algorithm. Dans *International Conference on Machine Learning*, (pp. 148–156). 58

Frey, P. & Slate, D. (1991). Letter recognition using Holland-style adaptive classifiers. *Machine Learning*, *6*(2), 161–182. 212

Friedman, H., Nachman, I., & Peér, D. (1999). Learning bayesian network structure from massive datasets : The "sparse candidate" algorithm. Dans *Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI'99)*. 91

Friedman, N. (1997). Learning belief networks in the presence of missing values and hidden variables. Dans *Proceedings of the 14th International Conference on Machine Learning*, (pp. 125–133). Morgan Kaufmann. 92, 122, 136, 137, 139, 141

Friedman, N. (1998). The bayesian structural EM algorithm. Dans Cooper, G. F. & Moral, S. (Eds.), *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, (pp. 129–138)., San Francisco. Morgan Kaufmann. 137

Friedman, N. & Elidan, G. (1999). LibB for windows/linux programs. http://www.cs.huji.ac.il/labs/compbio/LibB/. 209

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). bayesian network classifiers. *Machine Learning*, *29*(2-3), 131–163. 22, 89

Friedman, N. & Koller, D. (2000). Being bayesian about network structure. Dans Boutilier, C. & Goldszmidt, M. (Eds.), *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-00)*, (pp. 201–210)., SF, CA. Morgan Kaufmann Publishers. 90, 95

Frydenberg, M. (1990). The chain graph markov property. *Scandinavian Journal of Statistics*, *17*, 333–353. 27

Gales, M. (1999). Semi-tied covariance matrices for hidden markov models. Dans *IEEE Transaction on speech and Audio Processing*, volume 7, (pp. 272–281). 20

Gebhardt, J. & Kruse, R. (1995). Learning possibilistic networks from data. Dans *5th International Workshop on Artificial Intelligence and Statistics*, (pp. 233–244)., New York, USA. Springer. 25

Geiger, D. (1990). *Graphoids : A Qualitative Framework for Probabilistic Inference*. PhD thesis, UCLA Cognitive Systems Laboratory. 33

Geiger, D. (1992). An entropy-based learning algorithm of bayesian conditional trees. Dans *Uncertainty in Artificial Intelligence : Proceedings of the Eighth Conference (UAI-1992)*, (pp. 92–97)., San Mateo, CA. Morgan Kaufmann Publishers. 89

Geiger, D. & Heckerman, D. (1996). Knowledge representation and inference in similarity networks and bayesian multinets. *Artificial Intelligence*, *82*(1-2), 45–74. 131

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*(6), 721–741. 26, 47, 57, 145

Gilks, W., Richardson, S., & Spiegelhalter, D. (1996). *Markov Chain Monte Carlo in Practice*. Interdisciplinary Statistics. Chapman & Hall. 46, 57, 145

Gillies, D. (2000). *Philosophical Theories of Probability*. Routledge, King's College London. 193

Gillispie, S. & Lemieux, C. (2001). Enumerating markov equivalence classes of acyclic digraph models. Dans *Uncertainty in Artificial Intelligence : Proceedings of the Seventeenth Conference (UAI-2001)*, (pp. 171–177)., San Francisco, CA. Morgan Kaufmann Publishers. 40

Godenberg, A. & Moore, A. (2004). Tractable learning of large bayes net structures from sparse data. Dans *21st International Conference on Machine Learning*, (pp. 44–52)., Banff, Alberta, Canada. ACM Press, ISBN : 1-58113-828-5. 73

Gonzales, C. & Jouve, N. (2006). Learning bayesian networks structure using markov network. Dans Studený, M. & Vomel, J. (Eds.), *Proceedings of the Third European Workshop on Probabilistic Graphical Models (PGM'06)*, number ISBN : 80-86742-14-8, (pp. 147–154)., Prague, Czech Republic. 95

Greiner, R., X., S., Bin, S., & Wei, Z. (2005). Structural extension to logistic regression : Discriminative parameter learning of belief net classifiers. *Machine Learning Journal special issue on Probabilistic Graphical Models for Classification*, *59*, 297–322. 85

Greiner, R. & Zhou, W. (2002). Structural extension to logistic regression. Dans *Proceedings of the Eighteenth Annual National Conference on Artificial Intelligence (AAI02)*, (pp. 167–173)., Edmonton, Canada. 141

Grossman, D. & Domingos, P. (2004). Learning bayesian network classifiers by maximizing conditional likelyhood. Dans *Proceedings of the 21st International Conference on Machine Learning*, (pp. 361–368). ACM Press. 85

Guo, H., Boddhireddy, P., & Hsu, W. (2004). An aco algorithm for the most probable explanation problem. Dans *Australian Conference on Artificial Intelligence*, (pp. 778–790). 44

Guo, H. & Hsu, W. (2001). A survey of algorithms for real-time bayesian network inference. (unpublished). 48

Hádjek, A. (2003). Interpretation of probability. *invited contribution to the stanford encyclopédia of philosophy, ed. E.Zalta*, *1*, . http://plato.stanford.edu/entries/probability-interpret/. 193

Hamilton, J. (1994). *Time Series Analysis*. Princeton University Press. 20

Heckerman, D. (1999). A tutorial on learning with bayesian network. Dans M. I. Jordan (Ed.), *Learning in Graphical Models* (pp. 301–354). Kluwer Academic Publishers, Boston. 56

Heckerman, D., Geiger, D., & Chickering, M. (1994). Learning Bayesian networks : The combination of knowledge and statistical data. Dans de Mantaras, R. L. & Poole, D. (Eds.), *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, (pp. 293–301)., San Francisco, CA, USA. Morgan Kaufmann Publishers. 79, 88, 90

Henrion, M. (1988). Propagating uncertainty in Bayesian networks by probabilistic logic sampling. Dans J. F. Lemmer & L. M. Kanal (Eds.), *Uncertainty in Artificial Intelligence 2* (pp. 149–163). Amsterdam : Elsevier Science Publishing Company. 144

Henrion, M. (1990). An introduction to algorithms for inference in belief nets. Dans *Proceedings of the 5th Annual Conference on Uncertainty in Artificial Intelligence (UAI-90)*, New York, NY. Elsevier Science Publishing Company, Inc. 44, 48

Hitchcock, C. (1993). A generalized probabilistic theory of causal relevance. *Synthese*, *97*, 335–364. 200

Hitchcock, C. (Fall 2002). Probabilistic causation. Dans E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy*. . 198

Howard, R. & Matheson, J. (1981). Influence diagrams. *In Howard, R. and Matheson, J., editors, Readings on the Principles and Applications of Decision Analysis*, *volume II*, 721–762. 17

Humphreys, P. (1985). Why propensities cannot be probabilities. *The Philosophical Review 94*, *1*, 557–570. 194

Hurvich, C. & Tsai, C. (1989). Regression and times series model selection in small samples. *Biometrika*, *76*, 297–307. 82

Ide, J., Cozman, F., & Ramos, F. (2004). Generating random bayesian networks with constraints on induced width. Dans *European Conference on Artificial Intelligence (ECAI)*, (pp. 323–327). IOS Press, Amsterdan. 145

Jarník, V. (1930). O jistem problemu minimalnim. *raca Moravske Prirodovedecke Spolecnosti (in Czech)*, *6*, 57–63. 139, 140

Jensen, F. (1996). *An introduction to Bayesian Networks*. Taylor and Francis, London, United Kingdom. 13

Jensen, F. (2001). *Bayesian Networks and Decision Graphs*. Springer Verlag series : Statistics for Engineering and Information Science, ISBN : 0-387-95259-4. 17

Jensen, F., Lauritzen, S., & Olesen, K. (1990). bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quaterly*, *4*, 269–282. 42, 43, 44

Johnson, W. E. (1921). Logic. *Cambridge University Press*. 193

Jordan, M., Ghahramani, Z., Jaakkola, T., & Saul, L. (1998). An introduction to variational methods for graphical models. Dans M. I. Jordan (Ed.), *Learning in Graphical Models* (pp. 105–161). Kluwer Academic Publishers, Boston. 47

Jordan, M. I. (1998). *Learning in Graphical Models*. The Netherlands : Kluwer Academic Publishers. 13

Kadie, C., Hovel, D., & Horvitz, E. (2001). Msbnx : A component-centric toolkit for modeling and inference with bayesian networks. Microsoft Research Technical Report MSR-TR-2001-67. http://www.research.microsoft.com/adapt/MSBNx/. 210

Kaelbling, L., Littman, M., & Cassandra, A. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*, 99–134. 20

Kanazawa, K., Koller, D., & Russell, S. (1995). Stochastic simulation algorithms for dynamic probabilistic networks. Dans Besnard, P. & editors, S. H. (Eds.), *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI'95)*, (pp. 346–351)., San francisco, USA. Morgan Kauffmann Publishers. 18

Karčiauskas, G. (2005). *Learning with Hidden Variables : A Parameter Reusing Approach for Tree-Structured Bayesian Network*. PhD thesis, Departement of Computer Science, Aalborg University. 141

Keren, D. (2002). Painter recognition using local features and Naive Bayes. Dans *Proceedings on the International Conference on Pattern Recognition*, (pp. 474–477). 22

Keynes, J. M. (1921). A treatise on probability. *Macmillan and Co*. 193

Kim, J. & Pearl, J. (1983). A computational model for combined causal and diagnostic reasoning in inference systems. Dans *Proceedings IJCAI-83*, (pp. 190–193)., Karlsruhe, Germany. 42

Kim, J. & Pearl, J. (1987). Convice ; a conversational inference consolidation engine. *IEEE Trans. on Systems, Man and Cybernetics*, *17*, 120–132. 13

Kjærulff, U. (1993). Approximation of Bayesian networks through edge removals. Research Report IR-93-2007, The Machine Intelligence Group, Aalborg University. 48

Koller, D. & Pfeffer, A. (1997). Oject-oriented bayesian network. Dans *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, (pp. 302–313). 16

Korb, K. (2001). Machine learning as philosophy of science. Dans *ECML-PKDD-01 workshop on Machine Learning as Experimental Philosophiy of Science*. 4

Korel, B. (1990). Automated software test data generation. *IEEE transations on Software Engineering*, *16(8)*, 870–879. 144

Korel, B. & Al-Yami, A. (1996). Assertion-oriented automated test data generation. Dans *Proceedings of the 18th International Conferance on Software Engineering (ICSE)*, volume 16(8), (pp. 71–80). 144

Kočka, T., Bouckaert, R., & Studenỳ, M. (2001). On characterization inclusion of bayesian networks. Dans Breese, J. & Koller, D. (Eds.), *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence.*, (pp. 261–268). Morgan Kaufmann. 94

Kruskal, J. (1956). On the shortest spanning subtree of a graph and traveling salesman problem. Dans *Proceedings of the American Mathematical Society 7*, (pp. 48–50). 139, 140

Kullback, S. (1952). An application of information theory to multivariate analysis. *Annals of mathematics and Statistics*, *23*, 88–102. 70

Kurgan, L., Cios, K., Tadeusiewicz, R., Ogiela, M., & Goodenday, L. (2001). Knowledge discovery approach to automated cardiac spect diagnosis. *Artificial Intelligence in Medicine*, *23*(2), 149–169. 212

Lam, W. & Bacchus, F. (1993). Using causal information and local measures to learn bayesian networks. Dans Heckerman, D. & Mamdani, A. (Eds.), *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, (pp. 243–250)., San Mateo, CA, USA. Morgan Kaufmann Publishers. 83

Laplace, P. S. (1814). A philosophical essay on probabilities. *English edition 1951, New York, Dover Publications Inc*. 193, 197

Larrañaga, P., Kuijpers, C., Murga, R., & Yurramendi, Y. (1996). Learning bayesian network structures by searching the best order ordering with genetic algorithms. *IEEE Transactions on System, Man and Cybernetics*, *26*, 487–493. 90

Larrañaga, P., Poza, M., Yurramendi, Y., Murga, R., & C. Kuijpers, C. (1996). Structure learning of bayesian networks by genetic algorithms : A performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18(9)*, 912–926. 96

Laskey, K. (2006). First-order bayesian logic. George Mason University Department of Systems Engineering and Operations Research. http://ite.gmu.edu/~klaskey/papers/Laskey_FOBL.pdf. 21

Lauritzen, S. (1996). *Graphical Models*. Oxford : Clarendon Press. 47

Lauritzen, S. & Spiegelhalter, D. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B*, *50*(2), 157–224. 13, 42, 44

Lauritzen, S. L., Badsberg, J. H., BÃ¸ttcher, S. G., Dalgaard, P., Dethlefsen, C., Edwards, D., Eriksen, P. S., Gregersen, A. R., HÃ¸jsgaard, S., & Kreiner, S. (2004). gr - graphical models in r. Aalborg, DENMARK : Aalborg University Department of Mathematical Sciences. Available from URL : http://www.math.auc.dk/gr/. 209

Lauritzen, S. L. & Wermuth, N. (1989). Graphical models for associations between variables, some of which are quantitative and some qualitative. *Annals of Statistics*, *17*, 31–57. 27, 38

Lawrence, N. (2000). *Variational Infererence in Probabilistic Models*. PhD thesis, University of Cambridge, U.K. 47

Lecoutre, M.-P., Clément, E., & B., L. (2004). Failure to construct and transfer correct representations across probability problems. *Psychological Reports*, *94*, 151–162. 197

Leray, P. (1998). *Apprentissage et Diagnostic de Systemes Complexes : Réseaux de Neurones et Réseaux Bayesiens. Application À La Gestion En Temps Réel Du Trafic Téléphonique Français*. PhD thesis, Université Paris 6.

Leray, P. & François, O. (2004a). Réseaux bayésiens pour la classification - méthodologie et illustration dans le cadre du diagnostic médical. *Revue d'Intelligence Artificielle, ISBN : 2-7462-0912-8*, *18*(2/2004), 169–193.

Leray, P. & François, O. (2004b). BNT structure learning package : Documentation and experiments. Technical Report 2004/PhLOF, Laboratoire PSI, INSA de Rouen. http://bnt.insa-rouen.fr. 98, 156, 210, 219, 222

Leray, P. & François, O. (2005). Bayesian Network Structural Learning and Incomplete Data. Dans *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR 2005), Espoo, Finland*, (pp. 33–40). 10, 138, 141

Leray, P., Guilmineau, S., Noizet, G., François, O., Feasson, E., & Minoc, B. (2003). French BNT site. http://bnt.insa-rouen.fr/. 98, 210

Levin, I. P. (1999). *Relating statistics and experimental design*. Thousand Oaks, CA : Sage Publications.Quantitative Applications in the Social Sciences series 125. 68

Li, Z. & D'Ambrosio, B. (1994). Efficient inference in bayes nets as a combinatorial optimization problem. *International Journal of Approximate Reasoning*, *11*(1), 55–81. 45

Lim, T.-S., Loh, W.-Y., & Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, *40(3)*, 203–228. 211, 213

MacKay, D. (1998). Choice of basis for Laplace approximation. *Machine Learning*, *33*(1), 77–86. 80

Maes, S. (2005). *Multi-Agent Causal Models : Inference and Learning*. PhD thesis, Vrije Universiteit Brussel, Faculteit Wetenschappen, DINF, Computational Modeling Lab. 24

Maes, S., Meganck, S., & Leray, P. (2007). An integral approach to causal inference with latent variables. Technical report, Laboratoire PSI, INSA de Rouen. 24, 25

Maes, S., Meganck, S., & Manderick, B. (2005). Identification of causal effects in multi-agent causal models. Dans *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2005)*, (pp. 178–182)., Innsbruck. 16, 24

Margaritis, D. (2003). *Learning Bayesian Network Model Structure from Data*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh. 73

Maybeck, P. S. (1979). *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. Academic Press, Inc. (copyright now owned by Navtech Seminars & GPS Supply). http://www.cs.unc.edu/~welch/media/pdf/maybeck_ch1.pdf. 20

Mazer, E., Miribel, J.-F., Bessière, P., Lebeltel, O., Mekhnacha, K., & Ahuactzin, J.-M. (2004). Probayes : Mastering uncertainty. http://www.probayes.com/. 209

McLachlan, G. & Krishnan, T. (1996). The EM algorithm and extensions. John Wiley & Sons, New York. 124

Meek, C. (1997). *Graphical Models : Selecting causal and statistical models*. PhD thesis, Carnegie Mellon University. 93, 94, 95

Meganck, S., Leray, P., Maes, S., & Manderick, B. (2006). Apprentissage des réseaux bayésiens causaux à partir de données d'observation et d'expérimentation. Dans *15e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle, RFIA 2006*. 200

Meganck, S., Leray, P., & Manderick, B. (2006). Learning causal bayesian networks from observations and experiments : A decision theoretic approach. Dans *Modelling Decisions in Artificial Intelligence (MDAI'06)*. 24, 115, 174

Meila-Predoviciu, M. (1999). *Learning with Mixtures of Trees*. PhD thesis, MIT. http://people.csail.mit.edu/people/mmp/thesis.html. 141

Mekhnacha, K., Ahuactzin, J.-M., Bessière, P., Mazer, E., & Smail, L. (2006). A unifying framework for exact and approximate bayesian inference. Technical Report RR-5797, Rapport de recherche de l'INRIA - Rhone-Alpes , Equipe : E-MOTION. http://www.inria.fr/rrrt/rr-5797.html. 45

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). *Machine Learning*, *Neural and Statistical Classification*. http://www.amsta.leeds.ac.uk/~charles/statlog/, http://www.liacc.up.pt/ML/statlog/datasets/. 211

Moore, A. & Wong, W.-K. (2003). Optimal reinsertion : A new search operator for accelerated and more accurate bayesian network structure learning. Dans Fawcett, T. & Mishra, N. (Eds.), *Proceedings of the 20th International Conference on Machine Learning (ICML '03)*, (pp. 552–559)., Menlo Park, California. AAAI Press. 92

Moral, S., Rumí, R., & Salmerón, A. (2001). Mixtures of truncated exponentials in hybrid bayesian networks. *Lecture Notes in Computer Science*, *2143*, 156–167. 38

Munteanu, P., Jouffe, L., & Wuillemin, P. (2001). Bayesia lab. http://www.bayesia.com/. 209

Murphy, K. (2001). Active learning of causal bayes net structure. Technical report, UC Berkeley. 95

Murphy, K. (2002). *Dynamic bayesian Networks : Representation, Inference and Learning*. PhD thesis, University of california, Berkeley. 18, 20

Murphy, K. (2004). Bayes net toolbox v5 for matlab. Cambridge, MA : MIT Computer Science and Artificial Intelligence Laboratory. http://bnt.sourceforge.net. 98, 209, 210

Murphy, K., Weiss, Y., & Jordan, M. (1999). Loopy belief propagation for approximate inference : An empirical study. Dans *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, (pp. 467–475)., San Francisco, CA. Morgan Kaufmann Publishers. 48

Muruzábal, J. & Cotta, C. (2004). A primer on the evolution of equivalence classes of bayesian network structures. *Lecture Notes in Computer Science*, *3242*, 612–621. Parallel Problem Solving From Nature VIII, Springer-Verlag, Berlin. 95

Myers, J., Laskey, K., & Lewitt, T. (1999). Learning bayesian network from incomplete data with stochastic search algorithms. Dans *the Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI99)*. 138

Myllymäki, P., Silander, T., Tirri, H., & Uronen, P. (2002). B-course : A web-based tool for bayesian and causal data analysis. International Journal on Artificial Intelligence Tools, **11**(3), p. 369-387. http://b-course.hiit.fi/. 210

Naïm, P., Wuillemin, P.-H., Leray, P., Pourret, O., & Becker, A. (2004). *Réseaux bayésiens*. Eyrolles, ISBN : 2-212-11137-1. 13, 52

Neal, R. & Hinton, G. (1998). A view of the EM algorithm that justifies incremental, sparse and other variants. Dans M. I. Jordan (Ed.), *Learning in Graphical Models* (pp. 355–368). Kluwer Academic Publishers, Boston. 56, 128

Netica (1998). by Norsys Software Corp. http://www.norsys.com/. 210

Nielsen, S. & Nielsen, T. (2006). Adapting bayes network structures to non-stationary domains. Dans Studený, M. & Vomlel, J., I. .-.-.-. (Eds.), *Third European Workshop on Probabilistic Graphical Models*, (pp. 223–230)., Prague, Czech Republic. 95

Nijman, M., Akay, E., Wiegerinck, W., & Nijmegen, S. (2002). Bayesbuilder : A tool for constructing and testing bayesian networks. Available at URL : http://www.snn.kun.nl/nijmegen/index.php3?page=31. 210

Nodelman, U. & Horvitz, E. (2003). Ctbns for inferring users' presence and activities with extentions for modeling and evaluation. Technical Report MSR-TR-2003-97, Microsoft Research. 21

Nodelman, U., Koller, D., & Shelton, C. (2005). Expectation propagation for continuous time bayesian networks. Dans *Proceedings of the Twenty-first Conference on Uncertainty in AI (UAI)*, (pp. 431–440). 21

Nodelman, U., Shelton, C., & Koller, D. (2005). Expectation maximization and complex duration distributions for continuous time bayesian networks. Dans *Proceedings of the Twenty-first Conference on Uncertainty in AI (UAI)*, (pp. 421–430). 21

Olave, M., Rajkovic, V., & Bohanec, M. (1989). An application for admission in public school systems. *Expert Systems in Public Administration*, *1*, 145–160. Elsevier Science Publishers (North Holland). 212

Ollivier, Y. (1997). Soirée philo. http://www.eleves.ens.fr/home/ollivier/philo/ph970117.htm. 200

Pargas, R., Harrold, M., & Peck, R. (1999). Test-data generation using genetic algorithms. *Journal of Software testing, Verification and reliability*, *9*, 263–282. 144

Peña, J., Lozano, J., & Larrañaga, P. (2002). Learning recursive bayesian multinets for data clustering by means of constructive induction. *Machine Learning*, *47 :1*, 63–90. 22, 141

Pearl, J. (1985). bayesian networks : a model of self-activated memory for evidential reasoning. Technical Report 850021 (R-43), UCLA Computer Science Department Technical Report, and in Cognitive Science Society, UC Irvine, 329-334. 13, 42, 48

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Journal of Artificial Intelligence*, *29*, 241–288. 42

Pearl, J. (1987). Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, *32*(2), 245–258. 145

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference.* Morgan Kaufmann, second edition in 1991. 5, 13, 30, 48, 194

Pearl, J. (2000). *Causality : Models, Reasoning, and Inference*. Cambridge, England : Cambridge University Press, ISBN : 0-521-77362-8. 24, 30, 72

Pearl, J. & Paz, A. (1985). Graphoids : A graph based logic for reasonning about relevance relations. Technical Report 850038 (R-53-L), Cognitive Systems Laboratory, University of California, Los Angeles. 33, 34

Pearl, J. & Verma, T. (1991). A theory of inferred causation. Dans Allen, J. F., Fikes, R., & Sandewall, E. (Eds.), *KR'91 : Principles of Knowledge Representation and Reasoning*, (pp. 441–452)., San Mateo, California. Morgan Kaufmann. 71

Pearson, K. (1892). *The Grammar of Science*. Dover Publications 2004 edition, ISBN : 0486495817. 68

Pernkopf, F. & Bilmes, J. (2005). Discriminative versus generative parameter and structure learning of bayesian network classifiers. Dans *22nd International Conference on Machine Learning (ICML'05)*, (pp. 657–664)., Bonn, Germany. ACM Press, ISBN : 1-59593-180-5. 112

Perry, B. P. & Stilson, J. A. (2002). Bn-tools : A software toolkit for experimentation in bbns (student abstract). In Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002), Edmondon, Alberta, CANADA, pp. 963-964. Menlo Park, CA : AAAI Press. Avalaible at URL : http://bnj.sourceforge.net/. 209

Poole, D. (1993). Average-case analysis of a search algorithm for estimating prior and posterior probabilities in bayesian networks with extreme probabilities. Dans *Proceeding of the thirteenth International Joint Conference on Artificial Intelligence*, (pp. 606–612). 48

Popper, K. (1957). The propensity interpretation of the calculus of probability and the quantum theory. *S. Körner (ed.), The Colston Papers, 9*, 65–70. 193

Putterman, M. (1994). Markov decision processes : Discrete stochastic dynamic programming. Jhon Wiley & Sons, Inc. 20

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. Dans *Proceedings of the IEEE Transactions*, volume 77(2) of *ASSP series*, (pp. 257–285). 20

Rabiner, L. R. & Juang, B. H. (1986). An introduction to hidden markov models. *IEEE ASSP Magazine*, *1*, 4–16. 20

Ramamoorthy, C., Ho, S., & W.T., C. (1976). On the automated generation of program test data. *IEEE Transactions on software Engineering*, *2(4)*, 193–300. 144

Ramoni, M. & Sebastiani, P. (2000). Robust learning with missing data. *Machine Learning*, *45*, 147–170. 58, 138, 173

Ramsey, F. P. (1926). Truth and probability. *in Foundations of Mathematics and other Essays, R. B. Braithwaite (ed.), Routledge & P. Kegan , 1931, 156-198 ; reprinted in Studies in Subjective Probability, H. E. Kyburg, Jr. and H. E. Smokler (eds.), 2nd ed., R. E. Krieger Publishing Company, 1980, 23-52 ; reprinted in Philosophical Papers, D. H. Mellor (ed.) Cambridge University Press, 1990*. 193

Richardson, T. & Spirtes, P. (2002). Ancestral graph Markov models. Technical Report 375, Dept. of Statistics, University of Washington. 25

Rissanen, J. (1978). Modelling by shortest data description. *Automatica*, *14*, 465–471. 82, 83

Robert, C. (1994). *The bayesian Choice : a decision-theoretic motivation*. New York : Springer. 53

Robert, C. & Castela, G. (2004). *Monte Carlo Statistical Methods*. Springer texts in statistics. 46

Robinson, R. W. (1977). Counting unlabeled acyclic digraphs. Dans Little, C. H. C. (Ed.), *Combinatorial Mathematics V*, volume 622 of *Lecture Notes in Mathematics*, (pp. 28–43)., Berlin. Springer. 14, 64

Roure i Alcobé, J. (2004). *Incremental Methods for Bayesian Network Structure Learning*. PhD thesis, Departament de Llenguatges i Sistemes d'Informació, Universitat Politècnica de Catalunya. 95

Rovira, K., Lecoutre, M.-P., B., L., & Poitevineau, J. (2003). Interprétations intuitives du hasard et degré d'expertise en probabilité. *In A. Vom Hofe, H. Chardin, J.-L. Bernaud & D. Guédon (Eds), Psychologie Différentielle : Recherches et réflexions. Rennes : Presses Universitaires de Rennes*, *1*, 167–171. 197

Rubin, D. (1976). Inference and missing data. *Biometrika*, *63*, 581–592. 54, 119, 146

Rubin, D. (1981). The bayesian boostrap. *Ann. Statistics*, *9*, 130–134. 58

Rubin, D. (1987). Multiple imputation for nonresponse in surveys. *New York : Wiley*. 58

Sahami, M. (1999). *Using Machine Learning to improove Information access*. PhD thesis, Dept. of computer science of Stanford University. 88

Sahani, M. (1999). *Latent Variable Models for Neural Data Analysis*. PhD thesis, California Institute of Technology, Pasadena, California. 18

Sangüesa, R. & Cortés, U. (1997). Learning causal networks from data : a survey and new algorithm for recovering possibilistic causal networks. *AI Communications*, *10*, 31–61. 25

Santos, E. & Hussein, A. (2004). Comparing case-based bayesian network and recursive bayesian multi-net classifiers. Dans *Proceedings of the International Conference on Artificial Intelligence, IC-AI '04, Volume 2 & Proceedings of the International Conference on Machine Learning ; Models, Technologies & Applications, MLMTA '04, June 21-24, 2004, Las Vegas, Nevada, USA*, (pp. 627–633). 23

Schwartz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464. 82

Scott, A. & Symons, M. (1971). Clustering methods based on likelyhood ration criteria. *Biometrics*, *27*, 387–397. 84

Sebastiani, P. & Ramoni, M. (2001). Bayesian selection of decomposable models with incomplete data. Journal of the American Statistical Association, 96, No. 456, pp 1375-1386. 58, 138

Sebastiani, P., Ramoni, M., & Crea, A. (1999). Profiling your customers using bayesian networks : A tutorial exercise and the bayesware. KDD Cup 99. http://bayesware.com/. 210

Sebe, N., Lew, M., Cohen, I., Garg, A., & T.S., H. (2002). Emotion recognition using a Cauchy Naive Bayes Classifier. Dans *Proceedings of the International Conference on Pattern Recognition*, (pp. 17–20). 22

Shachter, R. (1998). Bayes-ball : The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). Dans Cooper, G. & Moral, S. (Eds.), *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, (pp. 480–487)., San Francisco. Morgan Kaufmann. 42

Shachter, R. & Kenley, C. (1989). Gaussian influence diagrams. *Management Science*, *35*, 527–550. 17

Shachter, R. & Peot, M. (1989). Simulation approaches to general probabilistic inference on belief networks. Dans *Proceedings of Uncertainty in Artificial Intelligence 5*, (pp. 221–231). Elsevier Science Publishing Company. 145

Shachter, R. D. (1986). Intelligent probabilistic inference. Dans Kanal, L. N. & Lemmer, J. F. (Eds.), *Uncertainty in Artificial Intelligence*, (pp. 371–382)., Amsterdam. North-Holland. 44

Shachter, R. D., D'Ambrosio, B., , & DelFabero, B. (1990). Symbolic probabilistic inference in belief networks. Dans *Proceedings of the Eighth National Conference on Artificial Intelligence*, (pp. 126–131). 45

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton : Princeton University Press. 5

Shibata, R. (1976). Selection of the order of an autoregressive model by the Akaike's Information Criterium. *Biometrika*, *63*, 117–126. 82

Singh, M. & Valtorta, M. (1993). An algorithm for the construction of Bayesian network structures from data. Dans *Proceedings of Ninth Conference on Uncertainty in Artificial Intelligence,* Washington, DC, (pp. 259–265). Morgan Kaufmann. 37, 90

Singh, S., Littman, M., Jong, N., Pardoe, D., & Stone, P. (2003). Learning predictive state representation. Dans *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, (pp. 712–719). 21

Skyrms, B. (1980). Causal necessity : A pragmatic investigation of the necessity of laws. New Haven : Yale University Press. 200

Smail, L. (2004). *Algorithmique pour les Réseaux Bayésiens et leurs Extensions.* PhD thesis, Université de Marne-La-Vallée. 28, 45

Smith, J., Everhart, J., Dickson, W., Knowler, W., & Johannes, R. (1988). Using the adap learning algorithm to forecast the onset of diabetes mellitus. Dans *the Symposium on Computer Applications and Medical Care*, (pp. 261–265). IEEE Computer Society Press. 211

Spiegelhalter, D. (1986). Probabilistic prediction in patient management and clinical trials. *Statistics in Medecine*, *5*, 421–433. 22

Spiegelhalter, D. J. & Lauritzen, S. L. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, *20*, 579–605. 56

Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, prediction, and search*. Springer-Verlag. 30, 57, 71

Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, Prediction, and Search* (2 ed.). The MIT Press, ISBN : 0-262-19440-6. 30, 72

Spirtes, P., Glymour, C., & Scheines, R. (2004). The tetrad project : Causal models and statistical data. pittsburgh. PA : Carnegie Mellon University Department of Philosophy. Available from URL : http://www.phil.cmu.edu/projects/tetrad/. 209

Spirtes, P., Meek, C., & Richardson, T. (1999). An algorithm for causal inference in the presence of latent variables and selection bias. Dans *Computation, Causation, and Discovery* (pp. 211–252). Menlo Park, CA : AAAI Press. 24

Steinsky, B. (2003). Enumeration of labeled chain graphs and labeled essential directed acyclic graphs. *Discrete Math.*, *270*, 267–278. 40

Suppes, P. (1970). *A Probabilistic Theory of Causality*. Amsterdam : North-Holland. 198

Takikawa, M., d'Ambrosio, B., & Wright, E. (2002). Real-time inference with large scale temporal bayes nets. Dans *Proceedings of the Seventheenth Conference of Uncertainty in Artificial Intelligence*, (pp. 477–484)., San Mateo. Morgan Kaufmann. 18

Teyssier, M. & Koller, D. (2005). Ordering-based search : A simple and effective algorithm for learning bayesian networks. Dans *Twenty-first Conference on Uncertainty in AI (UAI'05)*, (pp. 584–590). 91

Thévenod-Fosse, P. & Waeselynck, H. (1993). Statemate : Applied to statistical software testing. Dans *ACM SIGSOFT, Proceedings of the 1993 International Symposium on Software Testing ans Analysis*, volume Software Enginering Notes 23(2), (pp. 78–81). 144

Thrun, S. (1991). The MONK's problems - A performance comparison of different learning algorithms. Technical Report CS-CMU-91-197, Carnegie Mellon University. others contributors : J.Bala, E.Bloedorn, I.Bratko, B.Cestnik, J.Cheng, K.De Jong, S.Dzeroski, S.Fahlman, D.Fisher, R.Hamann, K.Kaufman, S.Keller, I.Kononenko, J.Kreuziger, R.Michalski, T.Mitchell, P.Pachowicz, Y.Reich, H.Vafaie, W.Van de Weldea, W.Wenzel, J.Wnek et J.Zhang. 212

Tian, J. & Pearl, J. (2002a). A general identification condition for causal effects. Dans *Proceedings of the Eighteenth National Conference on Artificial Intelligence, AAAI Press / The MIT Press : Menlo Park, CA, 567-573, August 2002*, (pp. 567–573)., Menlo Park, CA. AAAI Press/The MIT Press. also printed in UCLA Cognitive Systems Laboratory Technical Report (R-290-A), April 2001. 24

Tian, J. & Pearl, J. (2002b). On the identification of causal effets. Technical Report R-290-L, UCLA C.S. Lab. 24

Tsamardinos, I., Brown, L., & Aliferis, C. (2005). The max-min hill-climbing bayesian network structure learning algorithm. Technical Report DSL-TR-05-01, Vanderbuilt University. http://www.dsl-lab.org. 73, 209

Venn, J. (1876). The logic of chance. *2nd ed., Macmillan and co, reprinted, New York*. 193

Verma, T. & Pearl, j. (1988). Causal networks : Semantics and expressiveness. Dans *in Proceedings, 4th Workshop on Uncertainty in Artificial Intelligence*, (pp. 352–359). UCLA Cognitive Systems Laboratory Technical Report 870032 (R-65), June 1987, , Minneapolis, MN, Mountain View, CA, . Also in R. Shachter, T.S. Levitt, and L.N. Kanal (Eds.), Uncertainty in AI 4, Elsevier Science Publishers, 69-76, 1990. 35, 36

Verma, T. & Pearl, J. (1990). Equivalence and synthesis of causal models. Dans *Proceedings Sixth Conference on Uncertainty and Artificial Intelligence*, (pp. 255–268)., San Francisco. Morgan Kaufmann. 39

Wainwright, M. & Jordan, M. (2003). Graphical models, exponential families, and variational inference. Technical report, Departement of Statistics, University of California, Berkeley. 47

Wei, G. & Tanner, M. (1990). A monte-carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the Americam Statistical Association*, *85(411)*, 699–704. 128

Wermuth, N. & Lauritzen, S. L. (1990). On substantive research hypotheses, conditional independence graphs and graphical chain models (with discussion). *Journal of the Royal Statistical Society*, *52*(B), 21–72. 27

Wiener, N. (1948). Cybernetics or control and communication in the animal and the machine. Hermann et Cie et Cambridge (Mass.), The MIT Press. 70

Wong, M., Lee, S., & Leung, K. (2004). Data mining of bayesian networks using cooperative coevolution. *Decision Support Systems*, *38, issue 3*, 451–472. 95

Wothke, W. & Arbuckle, J. (1996). Full-information missing data analysis with amos. *Softstat '95 : Advances in statistical software*, *5*. Stuttgart, Germany : Lucius and Lucius. 58

Wright, S. (1934). The method of path coefficients. *Annals of Mathematics and Statistics*, *5*, 161–215. 33

Xiang, Y. (1999). WebWeavR-III. http://snowhite.cis.uoguelph.ca/faculty_info/yxiang/ww3/. 209

Xiang, Y. (2002). Probabilistic reasoning in multiagent systems : A graphical models approach. *Cambridge University Press 2002*. 16

Xiang, Y. & Miller, T. (1999). A well-behaved algorithms for simulating dependence structure of bayesian networks. *International Journal of Applied Mathematics*, *1*, 923–932. 145

Yehezkel, R. & Lerner, B. (2005). Recursive autonomy identification for bayesian network structure learning. Dans *The 10th International Workshop on Artificial Intelligence & Statistics, AISTATS 2005*, (pp. 429–436)., Barbados. 73

York, J. (1992). Use of the Gibbs sampler in expert systems. *Artificial Intelligence*, *56*, 115–130. 145

Yun, Z. & Keong, K. (2004). Improved MDL score for learning of Bayesian networks. Dans *Proceedings of the International Conference on Artificial Intelligence in Science and Technology, AISAT 2004*, (pp. 98–103). 83

Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems, 1978*, *1*, 3–28. 5, 25

Zhang, J. (2006). *Causal Inference and Reasoning in Causally Insufficient Systems*. PhD thesis, Departement of Philosophy, Carnegie Mellon University. 25, 72

Zhang, J. & Spirtes, P. (2005). A characterization of markov equivalence classes for ancestral graphical models. Technical Report 168, Dept. of Philosophy, Carnegie-Mellon University. 25

Zhang, N. & Poole, D. (1994). A simple approach to bayesian network computations. Dans *In Proceedings of the tenth Canadian Conference on Artificial Intelligence*, (pp. 171–178). 44

Zhang, N. L. (2002). Hierarchical latent class models for cluster analysis. Dans *Proceedings of AAAI'02*. 174

Zhou, L., Feng, J., & Sears, A. (2005). Applying the naive bayes classifier to assist user in detectind speech recognition errors. Dans *Proceedings of the 38th Hawaï International Conference on System Science*. 22

A

Historique des probabilités et raisonnement probabiliste

A.1 Différentes interprétations des probabilités

Il existe différentes manières de définir et/ou d'interpréter les probabilités. Voici les cinq principales interprétations des probabilités Hádjek (2003) et Gillies (2000).

Probabilités classiques : introduites par Laplace (1814), et reprises dans les travaux de Pascal, Bernoulli, Huygens, et Leibniz. Il s'agit ici d'attribuer des probabilités sans aucune observation, ou en présence d'observations supposées symétriquement équilibrées et parfaitement représentatives. L'idée de base est de dire que la masse de probabilité est partagée de manière égale par tous les résultats possibles, cette interprétation n'est donc réservée qu'aux ensembles finis.

Interprétation logique : introduite par Johnson (1921) et Keynes (1921), et très proche des probabilités classiques. Ici, la masse de probabilité n'est pas équirépartie. Soit il n'existe pas d'observations, soit il y a des observations et il n'est plus obligatoire de les supposer symétriquement équilibrées, il faut alors choisir la distribution de la masse.

Interprétation fréquentiste : une version simple du fréquentisme, qui est appelée *fréquentisme fini*, donne des probabilités aux évènements ou attributs pour un ensemble d'exemples de référence de la manière suivante Venn (1876) : *La probabilité d'un évènement (ou d'un attribut) pour un ensemble de référence est sa fréquence relative d'apparition dans cet ensemble de référence*. Il est alors possible d'imaginer les probabilités comme la limite de cette définition lorsque la taille de l'ensemble de référence tend vers l'infini. La limite de cette interprétation est qu'un évènement n'est pas toujours aisément reproductible (élections, expériences onéreuses, *etc*).

Interprétation propensionniste (ou objectivisme) : il s'agit ici de dire que les probabilités sont dans la nature Popper (1957). La probabilité est considérée comme une propriété physique d'un type donné de situation. Lorsqu'une expérience n'est pas reproductible, il peut être demandé à des agents d'estimer cette probabilité. Le principe est alors que les créances *raisonnables* essayent d'identifier les "chances" de sorte que, si un agent raisonnable connaît la propension des résultats donnés, leurs degrés de croyance soient identiques.

Interprétation bayésienne (ou subjectiviste) : dans cette théorie, les probabilités sont vues comme des degrés de croyance Ramsey (1926). Le problème est que la définition d'une probabilité n'est alors plus unique mais devient très subjective.

La formule d'inversion de Thomas Bayes (équation 1.1) peut s'énoncer dans toutes ces interprétations, sauf celle *propensionniste* Humphreys (1985). Par la suite nous supposerons que nous sommes en présence de probabilités fréquentistes ou bayésiennes.

Cette formule permet de mettre à jour les croyances/probabilités sur une hypothèse **H** en présence de l'observation **e**. Une nouvelle information sur **H** est obtenue. Celle-ci peut être utilisée pour mettre à jour notre croyance en une nouvelle hypothèse, et ainsi de suite. Ce principe de base sera utilisé pour l'inférence dans les réseaux bayésiens comme nous le verrons dans la section 4.

Les probabilités pouvant aussi bien représenter des croyances que des valeurs issues de statistiques, il faut alors "faire attention à la notion de subjectivisme" de sorte ***que les croyances soient raisonnables***.
Voici un exemple simple sur le type d'incohérences qui peuvent émerger lorsque les agents ne sont pas raisonnables. En particulier, si le problème est mal posé, il devient difficile d'estimer intuitivement les probabilités avec une approche subjective. Dans ce cas, le raisonnement probabiliste issu de la formule d'inversion de Bayes, même s'il est exact donnera des résultats faux !

A.2 Le paradoxe du changement de porte

Nous allons définir le paradoxe du changement de porte (*the Monty Hall problem*[i]) dont une variante est également appelée **Le paradoxe du prisonnier** et résolu formellement dans Pearl (1988) qui l'utilise pour expliquer pourquoi l'*inférence est contre-intuitive mais exacte*.

Le paradoxe du changement de porte :

Un candidat d'une fameuse émission de télévision arrive à la scène finale. Pour déterminer son gain, il doit choisir entre 3 portes. Derrière deux d'entre elles, il y a des lots sans grande valeur, mais derrière la dernière, il s'agit du gros lot. Le candidat désigne par exemple la première porte, la porte A. Le présentateur ouvre alors une porte qui ne cache pas le gros lot, par exemple la porte B, il dit alors "Voulez-vous changer de porte ?".
A la place du candidat, que feriez-vous ?

```
Si vous ne changez pas de porte, vous faites comme la majorité des gens..
..et vous avez tort !
```

Dans ce cas, pensez-vous que les deux portes restantes ont toutes les deux 50% de chance de cacher le gros lot ?
Ce n'est pas le cas, en effet, avec la stratégie de ne jamais changer de porte, il est clair que vous n'avez juste qu'une chance sur trois de gagner le gros lot, puisque il y a une chance sur trois simplement que vous désigniez la bonne porte au départ. Avec la stratégie de toujours changer de porte :

- si le candidat avait choisi dès le départ la bonne porte, il perd immanquablement. Ce cas se produit avec une probabilité de $\frac{1}{3}$.
- si le candidat avait choisi une mauvaise porte, ce qui arrive avec une probabilité de $\frac{2}{3}$, le présentateur est obligé de montrer l'autre porte ayant un lot de consolation. La troisième porte, qui est choisie quand on change, est alors le gros lot.

[i] Ce paradoxe a été mis à la mode dans les années 1990 grâce à un jeu télévisé américain, Let's make a deal (le Bigdil en France). Dans ce jeu, le candidat choisissait au terme du jeu entre 3 portes, avec 1 cabriolet et 2 chèvres. La controverse est née d'un article dans une revue allemande militant en faveur du changement de porte, signé Marilyn vos Savant, qui a la particularité d'avoir le plus grand QI jamais mesuré (228). De grands scientifiques auraient été contre cette idée, et lui auraient envoyé des lettres d'insultes...

Nous avons donc deux chances sur trois de gagner avec un changement de porte systématique, contre une sur trois sans changement ! Tout l'intérêt des probabilités conditionnelles est dans ce problème :

Lorsque de l'information est apportée, il faut en tenir compte.

Dans ce cas, l'observation apportée n'apporte pas de nouvelle information sur la porte choisie. Elle apporte cependant de l'information sur la porte non choisie.

Une autre leçon à retenir de ces exemples est que les probabilités ne sont pas toujours intuitives. Avoir une approche exclusivement subjective peut poser des difficultés pour l'évaluation de probabilités, particulièrement en présence de conditionnement. Dans ce cas, la majorité des gens aurait dit que les deux portes restantes avaient une chance sur deux de cacher le gros lot.

Dans la section 5.2.2, nous allons voir comment mêler subjectivisme et fréquentisme à l'aide d'*a priori* particuliers sur les probabilités à évaluer.

Où y-a-t'il de l'information apportée ?

Que se passe t'il si l'on remplace l'ouverture par le présentateur, par une ouverture aléatoire ?

Si le présentateur ouvre une porte entièrement au hasard, c'est-à-dire avec une chance d'ouvrir celle ou il y a le gros lot. cela ne chance plus rien pour les probabilités et si par (mal)chance, il ouvre une porte avec un petit lot, la probabilité que le gros lot soit derrière chacune des deux portes restantes est alors de 50%.

Supposons à présent que le candidat ait choisi la porte A et regardons où sont les changements dans l'inférence. Supposons également que le présentateur ait ouvert la porte B complètement au hasard et qu'elle cache un petit lot.

Soit G_A la variable aléatoire représentant "la porte A cache le gros lot", et I_B celle représentant "la porte B cache un petit lot", alors il est possible d'écrire

$$\mathbb{P}(G_A = vrai | I_B = vrai) = \frac{\mathbb{P}(I_B = vrai | G_A = vrai)\mathbb{P}(G_A = vrai)}{\mathbb{P}(I_B = vrai)} = \frac{1 \times 1/3}{2/3} = \frac{1}{2}$$

A priori, le fait d'avoir vu que la porte B ne cachait pas le gros lot ne change pas la probabilité qu'il soit derrière A ou C ?

En effet, $\mathbb{P}(I_B = vrai | G_A = vrai)$, car il n'y a bien qu'un seul gros lot, ici supposé derriére A, mais si nous l'avions supposé derrière C cela ne changeait pas cette probabilité.

Y-a t'il une erreur ?

Non, il n'y a pas d'erreur car le fait que le présentateur ouvre la porte B n'apporte donc aucune information sur A et C et l'incertitude reste totale concernant les deux portes restantes, la probabilibité que B cache le gros lot tombe a 0 et celles des deux autres augmentent de manière équitable pour qu'il y ait conservation d'une distribution de probabilité.

A présent, ne raisonnons plus sur les implications mais sur la connaissance que nous avons, en particulier sur la connaissance que le présentateur possède.

Soit I'_B la variable aléatoire représentant "le présentateur AFFIRME que la porte B ne cache pas le gros lot sachant que la porte A a été choisie par le joueur (et le prouve en l'ouvrant)" alors le calcul devient :

$$\mathbb{P}(G_A = vrai | I'_B = vrai) = \frac{\mathbb{P}(I'_B = vrai | G_A = vrai)\mathbb{P}(G_A = vrai)}{\mathbb{P}(I'_B = vrai)} = \frac{1/2 \times 1/3}{1/2} = \frac{1}{3}$$

La différence entre ces deux calculs, qui sont tous deux exacts, vient du fait que I'_B implique I_B et que $\mathbb{P}(I'_B = vrai) = \frac{1}{2}$ tandis que $\mathbb{P}(I_B = vrai) = \frac{2}{3}$.

Ici, $\mathbb{P}(I'_B = vrai | G_A = vrai)$ vaut bien $\frac{1}{2}$, car le présentateur sachant que le lot est derrière A, il aurait pu choisir d'ouvrir C, remarquons ici que si le gros lot avait été derrière C, cela n'est plus symétrique est il est tout naturel que cette perte de symétrie se retrouve dans les probabilités.

Si l'on évalue la probabilité que "C cache le gros lot", G_C, alors dans le premier cas on obtient :

$$\mathbb{P}(G_C = vrai | I_B = vrai) = \frac{\mathbb{P}(I_B = vrai | G_C = vrai)\mathbb{P}(G_C = vrai)}{\mathbb{P}(I_B = vrai)} = \frac{1 \times 1/3}{2/3} = \frac{1}{2}$$

tandis que dans le second on obtient

$$\mathbb{P}(G_C = vrai | I'_B = vrai) = \frac{\mathbb{P}(I'_B = vrai | G_C = vrai)\mathbb{P}(G_C = vrai)}{\mathbb{P}(I'_B = vrai)} = \frac{1 \times 1/3}{1/2} = \frac{2}{3}$$

Ici encore, les deux calculs sont exacts, mais le deuxième est le plus intéressant car le fait que le présentateur ouvre la porte B n'apporte pas d'information sur le fait que A cache le gros lot mais sur le fait que C cache le gros lot, car dans tous les cas le présentateur n'ouvrira que B ou C , c'est-à-dire une porte qui n'aura pas été choisie par le candidat. Dans ce cas, c'est donc la probabilité que C cache le gros lot qui augmente.

Ce qu'il faut voir ici, c'est que l'inférence issue de la formule d'inversion de Bayes est toujours exacte, seulement elle ne peut donner des résultats intéressants que lorsque que nous avons une bonne compréhension du problème posé et que nous considérons alors les bonnes variables aléatoires.

En d'autres termes, l'inférence répond correctement à la question posée encore faut-il avoir réussi à poser la question qui donne bien la réponse que l'on cherche...

Une autre leçon à tirer est que l'impact d'une nouvelle information ne peut pas être évaluée si l'on ne regarde que les implications de celle-ci.

A.3 Hasard et probabilités

Un problème de chaussettes : Le biais d'équiprobabilité

On tire au hasard (à l'aveugle) une paire de chaussettes dans un tiroir qui contient deux chaussettes rouges et deux chaussettes vertes.

On considère les résultats suivants :

- Résultat 1 : on obtient une paire de chaussettes appareillées (deux rouges ou deux vertes)
- Résultat 2 : on obtient une paire de chaussettes dépareillées (une rouge et une verte)

Pensez-vous qu'il y a :

1) plus de chances d'obtenir le résultat 1

2) plus de chances d'obtenir le résultat 2

3) autant de chances d'obtenir les deux résultats

Répondez le plus spontanément possible (sans calcul).

`La réponse correcte est` : 2) plus de chances d'obtenir le résultat 2.

Il y a plus de chances d'obtenir une paire de chaussettes dépareillées (une rouge et une verte).

Si on numérote les chaussettes dans le tiroir par paire : $R_1R_2; V_1V_2$ il y a 6 tirages différents possibles : $R_1R_2; R_1V_1; R_1V_2; R_2V_1; R_2V_2; V_1V_2$, donc quatre chances sur six d'obtenir une paire de chaussettes dépareillées.

Si vous avez répondu 3) autant de chances d'obtenir les deux résultats, vous faites partie de la grande majorité et vous êtes sujet au *biais d'équiprobabilité* : "Lorsque la situation est présentée comme une situation *de hasard*, il existe chez une majorité de sujets, un modèle cognitif implicite selon lequel des événements à caractère aléatoire sont *par nature* équiprobables" Lecoutre, Clément & B. (2004).

Comme vous pouvez le remarquer, c'est également ce biais d'équiprobabilité qui vaut dans le *paradoxe du changement de porte*.

Probabilités et opinions :

Les personnes non habituées à manipuler les probabilités ont souvent des difficultés à se les représenter. Les décideurs, par exemple, ne peuvent pas se permettre de prendre une décision lorsqu'il reste un risque, même si sa probabilité est extrêmement faible.

«*L'homme de la rue, donc aussi le journaliste et l'homme politique, a une compréhension affective du hasard : dès lors qu'un phénomène relève des probabilités, il y a une sorte de maladie qui fait que tout est possible. Dans un sens favorable ou défavorable. L'incertain plane et brouille tout ce qu'il touche. On joue au loto malgré des probabilités de gains très faibles et des taxes dissuasives parce que "on ne sait jamais" et on refuse de faire vacciner ses enfants par crainte d'effets secondaires non confirmés parce que "on ne sait jamais". J'ai vainement tenté, avec un collègue économiste, de faire comprendre que la notion de risque n'était pas scalaire, qu'il y avait derrière au moins le couple d'une probabilité et d'une amplitude, peine perdue.*» Bouleau (1999)

Hasard ou Hasard ?

Considérez les deux évènements suivants :

1) Le fait de constituer une paire de chaussettes 'assorties' à partir d'un tirage à l'aveugle de deux chaussettes d'un tiroir qui contient deux paires de chaussettes différentes
2) Le fait qu'une graine mise en terre germe

Est-ce que, selon vous le hasard intervient ou non dans chacun de ces deux événements ? Il n'y a bien entendu pas de "bonne réponse" !
Trois groupes de sujets ont été interrogés par Rovira, Lecoutre, B. & Poitevineau (2003) : des collégiens, des psychologues et des mathématiciens. Une large majorité des sujets a le même avis pour le premier événement : ils répondent qu'il fait intervenir le hasard parce que "il est possible de calculer 'facilement' une probabilité". Cette majorité est toutefois plus faible chez les psychologues que chez les autres sujets.

Au contraire les sujets sont divisés pour le second item (graine). Deux conceptions principales sont observées : soit le hasard intervient parce qu'*un raisonnement probabiliste est en jeu*, soit le hasard n'intervient pas parce qu'*il existe une grande part de déterminisme* ou parce que *des facteurs causaux peuvent être identifiés*.

Une spécificité des mathématiciens est qu'un certain nombre d'entre eux se réfèrent explicitement à deux sortes de hasard Laplace (1814) :

1) un hasard "mathématique" (quand il est facile de calculer une probabilité)
2) un hasard "par ignorance" (quand il n'est pas facile de calculer une probabilité faute d'un modèle probabiliste standard disponible).

Néanmoins, même lorsque des facteurs causaux peuvent être identifiés, il n'est pas toujours possible de les identifier tous (pour la graine, pensons à la température, l'humidité, mais aussi la richesse du sol, *etc*).

Dans le cas où il n'est pas possible d'identifier toutes les *causes*, utiliser le hasard (et donc les probabilités) reste un moyen efficace de modéliser un système complexe (voire chaotique). Dans l'exemple de la graine, nous voyons qu'après un échange avec un botaniste il sera possible de créer un *réseau bayésien*. Ce dernier fournira une probabilité sur les chances qu'a cette graine de germer, même si cet évènement est en soit plus déterministe qu'aléatoire.

Cause/Effet ou co-variation :

Les concepts de probabilité et de causalité sont intimement liés malgré leurs différences. Ils sont tous deux utilisés pour représenter l'enchaînement des événements or ils restent très difficile à définir. Comme nous l'avons vu en page 193, il existe différentes définitions des probabilités, qui sont parfois incohérentes entre-elles. De même, nous pouvons imaginer plusieurs définitions pour la causalité, aucune n'étant réellement satisfaisante. La notion de causalité étant incertaine, elle peut se produire ou non en fonction des circonstances, elle a alors un caractère statistique. D'ou l'idée d'utiliser les probabilités pour la définir. Par exemple, prenons la définition introduite par Suppes (1970) qui dit qu'un événement A produit au temps t' est une cause d'un événement B produit au temps t si : 1) $t' < t$ 2) $\mathbb{P}(B) > 0$ 3) $\mathbb{P}(B|A) > \mathbb{P}(B)$

Cette définition est intéressante puisqu'elle incorpore le caractère contingent de la notion de causalité. Des exemples simples peuvent être trouvés dans la vie de tous les jours : *fumer* peut *provoquer un cancer du poumon*. Le problème de cette définition est qu'elle ne permet pas de distinguer la causalité de la corrélation simple. En effet, un événement peut augmenter la probabilité d'un autre sans en être la cause. Nous pouvons alors donner le contre exemple classique suivant : *la probabilité d'un orage en présence d'une indication 'orageuse' sur le baromètre est plus grande qu'en son absence*, or selon la définition précédente, nous en déduirions que l'indication du baromètre est une cause de l'orage !

La causalité implique la corrélation, mais la réciproque n'est pas vraie.

Depuis longtemps, un des objectifs des statistiques est d'essayer d'identifier des phénomènes de cause à effet, et de différencier alors les causes des effets.

Hitchcock a mis en ligne une bonne introduction sur les moyens actuels utilisés pour définir la causalité Hitchcock (2002). Bien qu'aucune définition ne soit philosophiquement satisfaisante[ii], il nous est néanmoins possible d'utiliser cette notion de manière intuitive.

Même sans définition formelle de la causalité, nous pouvons essayer d'identifier des causes et des effets. Cet objectif, pourtant intuitif, n'est pas si simple qu'il n'y parait. En particulier lorsque toute l'information nécessaire n'est pas disponible. Donnons à présent un autre exemple, similaire à celui de l'orage ci-dessus, ainsi qu'une explication possible du phénomène mis en exergue.

Prenons l'exemple de deux variables corrélées : soit M la variable représentant le fait qu'un individu soit malade ou pas, et D celle représentant le fait que des microbes se développent dans son organisme.

[ii] Du moins en 2002, selon Hitchcock (2002)

`Pasteur`, dans la "microbiologie des agressions"[iii], a dit que le développement des microbes provoque la maladie (D implique M). `Tissot`, dans la "microbiologie des mutations", a dit que le fait d'être malade engendre le développement des microbes (M implique D). Puis, quelques années plus tard, `Bernard`, dans la théorie de la "microbiologie des résistances", faisait intervenir une troisième variable H représentant la sensibilité de l'organisme au moment donné (encore appelé *encrassement humoral*). Il dit ensuite que cette grandeur influe sur le fait que d'être malade et sur le fait que les microbes ont la possiblité de se développer dans un organisme affaibli (H implique M et H implique D).

Dans ce cas, la variable H est ce que nous appellerons une *variable latente*, c'est-à-dire une variable qui est importante pour identifier les phénomènes de dépendances causales qui ne sont jamais observées. De plus, les variables M et D sont toujours corrélées lorsque la valeur de H est inconnue.

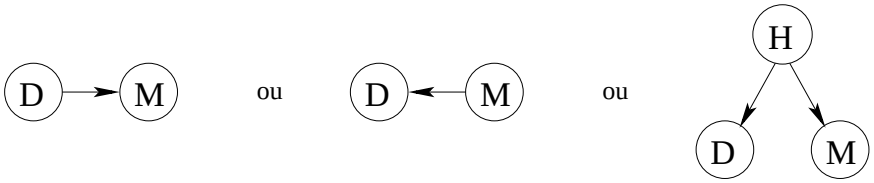


FIG. A.1 : *Quelle orientation choisir ?*

De manière générale, il est difficile de différencier la cause de l'effet lorsque l'on ne fait qu'observer ces deux variables. Dans ce cas, nous n'observons que la corrélation entre ces deux variables. Néanmoins, lorsque que nous faisons intervenir une troisième variable corrélée aux deux premières, il devient plus simple d'identifier les causes et les effets. Nous verrons cela plus en détail dans la section 3.4. Pour l'instant, essayons de définir la causalité.

Causalité et hasard :

Les notions de causalité et de hasard restent très liées puisque la notion de causalité modifie la notion de hasard, voire la fait disparaître. Lorsqu'un évènement possède plusieurs modalités et en réalise une aléatoirement, nous disons que c'est une variable aléatoire. Ensuite, l'étude des fréquences relatives des résultats passés de cette variable ou les degrés de croyance en les résultats possibles, suivant l'interprétation des probabilités qui est utilisée, permet de définir une probabilité.

Une probabilité n'est alors pas une prévision mais le mélange d'une constatation et d'une opinion.

La causalité est alors un phénomène de corrélation qui va permettre de réactualiser cette constatation ou de modifier notre opinion. Mais, cela n'est pas simplement une corrélation, car la notion de causalité est dynamique, c'est-à-dire que parmi ces deux entités nous dirons qu'une est cause de l'autre, si les deux sont corrélées et si elle précède l'autre[iv].

Avec l'habitude, le raisonnement humain est alors capable d'anticiper l'effet à la vue de la cause. Ce schéma de penser provient alors d'une adéquation avec la réalité[v].

[iii] Que les spécialistes m'excusent pour cette comparaison bien trop caricaturale.

[iv] Cette définition est la plus communément admise même si elle n'est pas satisfaisante (voir page 198).

[v] Puisque cette manière de penser est vraisemblablement un mécanisme Darwinien. Cela ne veut pas forcément dire qu'elle est optimale : penser à la théorie des jeux, nous sommes alors en présence d'un équilibre.

Pour identifier la causalité, nous devons observer une répétition. Néanmoins, la causalité existe sans une telle observation de répétition.

Il est alors possible de trouver de la causalité dans des domaines très variables. Il existe par exemple la causalité *physique*, qui est la plus usuelle et qui touche aux phénomènes physiques, la causalité *psychologique*, qui est liée à la décision et aux objectifs qui ont été fixés, et la causalité *logique*, qui intègre des notions plus formelles comme l'utilisation de règles (prémisses, conclusions, nécessité, potentialité).

Si nous appelons *cause* un évènement qui précède occasionnellement l'effet, nous pouvons faire de nouveau le parallèle avec la notion de hasard.

La causalité devient alors un phénomène qui modifie les chances d'apparition d'un phénomène. Cette première forme de hasard est celle qui est alors relative au manque de connaissance de l'état du monde (*hasard par ignorance*).

La deuxième forme de hasard qui peut être présentée serait celle qui est plus intrinsèque. Dans ce cas aucune connaissance supplémentaire sur l'état passé de l'univers ne peut permettre de modifier les chances d'apparition d'un évènement. Si ce type de hasard existe, nous dirons que nous sommes en présence d'un *univers non déterministe*.[vi]

Le hasard n'est alors pas obligatoirement un évènement dont nous ne pouvons pas reconnaître les causes. Par exemple, il est possible de donner le résultat d'un jet de dé grâce à la hauteur, l'angle, la force du lancer, le champ gravitationnel, etc. Mais en pratique, ce lancé étant fortement déterministe, mais également fortement chaotique, il est plus simple (et plus efficace) d'utiliser alors une modélisation probabiliste, car nous ne pouvons faire de mesures assez précises de ces grandeurs.

Si l'on arrive à identifier parfaitement les variables d'intérêt ainsi que leurs liaisons avec le phénomène en question, ce phénomène est alors considéré comme ne relevant plus du hasard.

Remarquons que la plupart du temps, les relations de causalité retrouvées à l'aide de méthodes automatiques ne seront présentes qu'à titre indicatif.

En effet, comme l'ont fait remarquer Meganck, Leray, Maes & Manderick (2006), pour être certains que nous sommes bien en présence de relations causales, il est souvent nécessaire d'effectuer des tests supplémentaires. Ces tests supplémentaires sont appelés des tests expérimentaux, car par exemple le font régulièrement les biologistes, ils consistent à tester si la relation de causalité subsiste lorsque le contexte varie. Ils permettent alors à la fois d'orienter la relation mais également d'être assuré que les deux variables considérées ne sont pas dépendantes d'un facteur extérieur (une variable latente). Ces tests ont certainement été inspirés par la définition de la causalité suivante, introduite par Cartwright (1979) et Skyrms (1980), reprise et comparée à d'autres définitions par Hitchcock (1993).

$$A \text{ est une cause de } B \text{ si } \mathbb{P}(B|A,T) > \mathbb{P}(B|\neg A,T), \text{pour toute situation de test } T.$$

Dans tous les cas, même si nous ne sommes pas certains qu'il s'agit bien de liens causaux, les variables considérées seront bien corrélées et l'inférence dans le réseau bayésien résultant donnera toujours de bons résultats, en particulier s'il n'y a pas de variables latentes.

[vi] Une alternative serait de dire que dans ce cas, l'univers est déterministe, mais la pensée humaine incapable d'identifier les mécanismes de détermination Ollivier (1997).

B

Notions de probabilités

B.1 Rappels de probabilités

Soient Ω un espace d'observables et $\mathscr{A}$ une tribu d'évènements sur Ω.
$(\Omega, \mathscr{A})$ est un espace probabilisable.

Définition B.1.1 (probabilité) *Une application $\mathbb{P} : \mathscr{A} \mapsto [0,1]$ est dite probabilité sur l'espace probabilisable $(\Omega, \mathscr{A})$ si elle vérifie les axiomes suivants :*

- $\mathbb{P}(\Omega) = 1$,
- *pour toute suite dénombrable $(A_1, A_2, \cdots)$ d'évènements de $\mathscr{A}$ qui sont deux à deux disjoints, la série $\sum_k \mathbb{P}(A_k)$ converge et a pour somme $\mathbb{P}(\bigcup_k A_k)$.*

$(\Omega, \mathbb{P})$ *est alors appelé* espace probabilisé.

Définition B.1.2 (variable aléatoire) *Nous appelons variable aléatoire X, toute fonction d'un espace probabilisable $(\Omega, \mathscr{A})$ vers un autre $(\mathscr{X}, \mathscr{B})$ telle que pour tout événement de $\mathscr{B}$, son image réciproque par X soit un événement de $\mathscr{A}$.*

Par la suite, toutes les définitions et théorèmes qui seront énoncés le seront à partir de variables aléatoires, mais il est bien évident que des énoncés analogues existent avec des événements.

Soient $X_1, \cdots, X_n$, Y, Z des variables aléatoires définies sur leurs tribus d'événements respectives (non nommées ici) et à valeurs dans $\mathscr{X}_1, \cdots, \mathscr{X}_n, \mathscr{Y}$ et $\mathscr{Z}$. Soit X la variable aléatoire $(X_1, \cdots, X_n)$ de $(\Omega, \mathscr{A})$ vers le produit cartésien $(\mathscr{X} = \mathscr{X}_1 \times \cdots \times \mathscr{X}_n,\ \mathscr{B})$[i].
L'ensemble $\{A \in \mathscr{A} | X(A) = x\}$ forme l'évènement que nous noterons $\{X = x\}$ ou encore $X = x$ ou encore simplement x pour raccourcir les notations et lorsque le contexte est clair. Nous utiliserons des notations similaires pour x_i, y et z.

Définition B.1.3 (probabilité conditionnelle) *Soit $y \in \mathscr{Y}$ telle que $\mathbb{P}(Y = y) \neq 0$ alors nous appelons* probabilité conditionnelle *à $Y = y$ la fonction $\mathbb{P}(.|y)$ qui à $x \in \mathscr{X}$ associe* [ii]

$$\mathbb{P}(x|y) = \frac{\mathbb{P}(x, y)}{\mathbb{P}(y)}$$

[i] Typiquement, si l'un des $\mathscr{X}_i$ est un ensemble non dénombrable de $\mathbb{R}$ nous prendrons la tribu borélienne pour $\mathscr{B}$, sinon, lorsque tous les $\mathscr{X}_i$ sont dénombrables nous prendrons $\mathscr{B} = 2^{\mathscr{X}}$.

[ii] Il est également possible de rencontrer les notations suivantes $\mathbb{P}_y(x) = \mathbb{P}_{Y=y}(X = x) = \mathbb{P}(X = x | Y = y) = \frac{\mathbb{P}(\{X=x\} \cup \{Y=y\})}{\mathbb{P}(Y=y)} = \frac{\mathbb{P}(X=x, Y=y)}{\mathbb{P}(Y=y)}$ mais, par la suite, nous nous conformerons aux notations introduites tant qu'il n'y a aucune ambiguïté.

Proposition B.1.1 *$\mathbb{P}(.|y) : \mathscr{X} \to [0,1]$ est une probabilité, mais $\mathbb{P}(.|.) : \mathscr{X} \times \mathscr{Y} \to [0,1]$ et $\mathbb{P}(x|.) : \mathscr{Y} \to [0,1]$ n'en sont pas.*

De plus, si pour tout y, $\mathbb{P}(Y = y) \neq 0$ alors la définition précédente existe toujours et nous noterons :

$$\mathbb{P}(X|Y) = \frac{\mathbb{P}(X,Y)}{\mathbb{P}(Y)}$$

Définition B.1.4 (loi jointe) *Nous appellerons* loi jointe *de l'ensemble de variables aléatoires $X_1, \cdots, X_n$, la fonction n-aire suivante :*

$$\begin{array}{rccc} \mathbb{P}(X_1, \cdots, X_n) : & \mathscr{X}_1 \times \cdots \times \mathscr{X}_n & \longrightarrow & [0,1] \\ & (x_1, \cdots, x_n) & \longmapsto & \mathbb{P}(x_1, \cdots, x_n) \end{array}$$

Cette loi jointe est alors une distribution de probabilité sur $(\mathscr{X}_1 \times \cdots \times \mathscr{X}_n, \mathscr{B})$.

Théorème B.1.2 (Théorème de Bayes généralisé)

$$\mathbb{P}(x_1, \cdots, x_n) = \mathbb{P}(x_1) \cdot \mathbb{P}(x_2|x_1) \cdot \mathbb{P}(x_3|x_1, x_2) \cdots \mathbb{P}(x_n|x_1, \cdots, x_{n-1}) \tag{B.1}$$

pour tout évènement $(x_1, \cdots, x_n) = \bigcap_{i=1}^{n} \{X_i = x_i\}$.

Propriété B.1.3 (marginalisation) *Nous avons $\mathbb{P}(y) = \int_{x \in \mathscr{X}} \mathbb{P}(y|x)\mathbb{P}(x)dx$. Par généralisation pour tout $y \in \mathscr{Y}$, et par abus de langage, nous noterons souvent $\mathbb{P}(Y) = \sum_X \mathbb{P}(Y|X)\mathbb{P}(X)$*

Définition B.1.5 (Espérance, variance, écart-type) *Nous appellerons* espérance *de la variable X, la valeur $\mathbb{E}(X) = \sum_X X \cdot \mathbb{P}(X)$, sa* variance*, la valeur $\mathbb{V}(X) = \mathbb{E}((X - \mathbb{E}(X))^2)$ et son* écart-type*, le nombre $\sigma(X) = \sqrt{\mathbb{V}(X)}$.*

B.2 Indépendance conditionnelle

La base du processus de représentation de la connaissance dans les réseaux bayésiens réside dans les notions de probabilité conditionnelle et d'indépendance conditionnelle.

Définition B.2.1 (indépendance) *Deux variables aléatoires X et Y sont dites (marginalement) indépendantes (noté $X \perp\!\!\!\perp Y$) si le fait que X se réalise ne donne pas d'infomation sur Y et réciproquement. On a donc $\mathbb{P}(X|Y) = \mathbb{P}(X)$ et $\mathbb{P}(Y|X) = \mathbb{P}(Y)$.*

Dans ce cas, la loi jointe de X et Y vaut $\mathbb{P}(X,Y) = \mathbb{P}(X)\mathbb{P}(Y)$.

Proposition B.2.1 *Quelles que soient les fonctions f et g, X et Y sont indépendantes est équivalente à $\mathbb{E}(f(X)g(Y)) = \mathbb{E}(f(X))\mathbb{E}(g(Y))$.*

Définition B.2.2 (Corrélation) *Deux variables aléatoires X et Y sont dites corrélées si*

$$\mathbb{E}(X \cdot Y) \neq \mathbb{E}(X) \cdot \mathbb{E}(Y)$$

Deux variables aléatoires indépendantes sont non-corrélées, mais la réciproque est fausse.

Définition B.2.3 (indépendance conditionnelle) *Soient trois variables aléatoires X, Y et Z. Alors X est dite indépendante à Y conditionnellement à Z (noté $X \perp\!\!\!\perp Y|Z$) si*

$$\mathbb{P}(x, y|z) = \mathbb{P}(x|z) \times \mathbb{P}(y|z)$$

pour les valeurs de x, y et z telles que $\mathbb{P}(X = x) \neq 0$, $\mathbb{P}(Y = y) \neq 0$ et $\mathbb{P}(Z = z) \neq 0$.

Théorème B.2.2 (formule d'inversion de Bayes) *Pour toutes valeurs de x et y telles que $\mathbb{P}(X = x) \neq 0$ et $\mathbb{P}(Y = y) \neq 0$, nous avons*

$$\mathbb{P}(x|y) = \frac{\mathbb{P}(y|x) \times \mathbb{P}(x)}{\mathbb{P}(y)} \tag{B.2}$$

En effet, car $\mathbb{P}(x, y) = \mathbb{P}(x)\mathbb{P}(y|x) = \mathbb{P}(y)\mathbb{P}(x|y)$.

Ce type de condition sera noté par la suite $\mathbb{P}(X|Y) = \frac{\mathbb{P}(Y|X) \times \mathbb{P}(X)}{\mathbb{P}(Y)}$ sans se soucier des valeurs de probabilités éventuellement nulles.

C

Notions de graphe

Nous allons à présent préciser le vocabulaire[i] utilisé dans la théorie des graphes. Soit n un entier naturel non nul.

C.1 Graphes non-orientés

Définition C.1.1 (graphe non-orienté, graphe semi-orienté) *Soit $X = \{X_1, \cdots, X_n\}$ un ensemble fini de nœuds, également nommés* points *ou* sommets*, et $\mathscr{E}$ une application de $X \times X$ vers $\{0,1\}$*[ii]*. Nous appellerons* graphe semi-orienté *et noterons $\mathscr{G} = (X, \mathscr{E})$ le couple constitué par l'ensemble X et l'application $\mathscr{E}$.*
Un graphe est dit non-orienté *(ou* non-dirigé*) si $(\mathscr{E}(X_i, X_j) = 1 \Rightarrow \mathscr{E}(X_j, X_i) = 1)$,*
$1 \leqslant i, j \leqslant n$.

Nous allons illustrer les notions suivantes à partir du graphe non-orienté de la figure C.1 que nous nommerons $\mathscr{G}$.

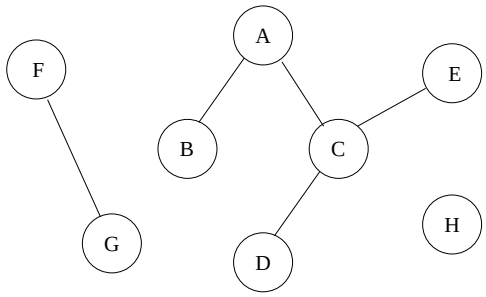


FIG. C.1 : *Un exemple de graphe non-orienté, nommé $\mathscr{G}$.*

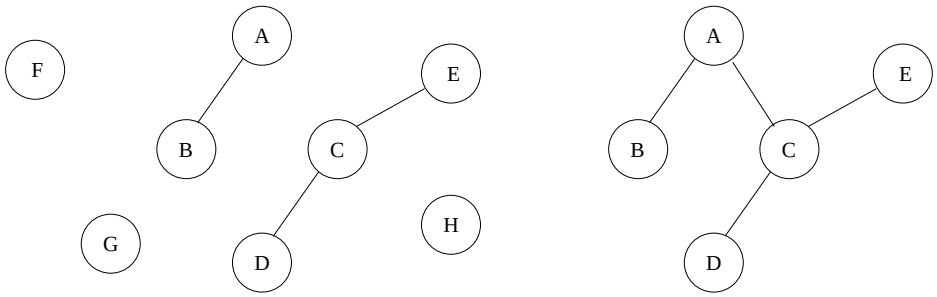
Définition C.1.2 (nœuds adjacents) *Deux sommets X_i et X_j sont dits* adjacents *si $(\mathscr{E}(X_i, X_j) = 1$ ou $\mathscr{E}(X_j, X_i) = 1)$.*
Nous noterons $Adj(X_i)$ l'ensemble des sommets adjacents à X_i.

[i] Attention ce vocabulaire diffère parfois de celui utilisé par les anglo-saxons, par exemple, ces derniers ne font pas de différence entre chemin et chaîne qu'ils traduisent par *path*, néanmoins, ils utilisent parfois *oriented path* quand le contexte n'est pas clair.

[ii] Le premier élément d'un graphe est toujours un ensemble de nœuds, par contre, pour le deuxième élément il est également possible de trouver une application Γ de X vers 2^X telle que $\Gamma(X_i) = \{Y \in X | \mathscr{E}(X_i, Y) = 1\}$, ou encore simplement un ensemble des *arcs* $\{(X_i, X_j) | \mathscr{E}(X_i, X_j) = 1\}$. Toutes ces définitions sont équivalentes.

Les sommets A et B sont adjacents dans le graphe $\mathscr{G}$ (figure C.1) et $Adj(A) = \{B, C\}$.

Définition C.1.3 (graphe partiel, sous-graphe) *Si $\mathscr{E}'$ est une application de $X \times X$ vers $\{0,1\}$ telle que $\mathscr{E}' \leqslant \mathscr{E}$ alors le graphe $\mathscr{G}' = (X, \mathscr{E}')$ sera dit graphe partiel de $\mathscr{G}$.*
Un sous-graphe de $\mathscr{G} = (X, \mathscr{E})$ est un graphe $\mathscr{G}' = (X', \mathscr{E}|_{X' \times X'})$ où X' est un sous-ensemble de sommets de X.



(a) Exemple de sous-graphe de $\mathscr{G}$.

(b) Exemple de graphe partiel de $\mathscr{G}$.

FIG. C.2 : *Exemple de sous-graphe et de graphe partiel.*

Un graphe partiel de $\mathscr{G}$ est illustré en figure C.2(a), et un exemple de sous-graphe est illustré en figure C.2(b).

Définition C.1.4 (arête) *Lorsque $\mathscr{E}(X_i, X_j) = 1$ et $\mathscr{E}(X_j, X_i) = 1$, nous dirons que nous sommes en présence d'une* arête*, que nous noterons graphiquement* (X_i)——(X_j).

Définition C.1.5 (chaîne, cycle) *Une* chaîne *est une suite d'arêtes dans laquelle chaque arête est reliée à la précédente et à la suivante par une extrémité commune. Pour simplifier les notations, une chaîne peut également être définie par la suite des sommets qu'elle rencontre.*
Un cycle *est une chaîne finie de longueur supérieure à 2 pour laquelle ses sommets initiaux et finaux sont identiques.*

Par exemple, la chaîne $(BA, AC, CE) = (BACE)$ et la longueur de cette chaîne est 3.

Définition C.1.6 (graphe connexe) *Un graphe est dit* connexe *si tout couple de sommets distincts du squelette de ce graphe peut être relié par une chaîne.*

Le sous-graphe de la figure C.2(b) est connexe.

Définition C.1.7 (graphe complet, clique) *Un graphe est dit* complet[iii] *lorsqu'il existe toujours une arête entre deux sommets quelconques.*
Une clique *d'un graphe $\mathscr{G} = (X, \mathscr{E})$ est un sous-ensemble de sommet $C \subset X$ tel que le* sous-graphe *$\mathscr{G}' = (C, \mathscr{E}|_{C \times C})$ de $\mathscr{G}$ relativement à l'ensemble de sommets C est complet.*

Le sous-graphe de $\mathscr{G}$ formé des deux nœuds F et G est complet.

[iii] Les notions de *graphe complet* et de *clique* sont spécifiques aux graphes non orientés, néanmoins, nous utiliserons également ces termes pour un graphe orienté lorsque le squelette de celui-ci sera complet.

Définition C.1.8 (arbre, forêt) *Un* arbre[iv] *est un graphe connexe sans cycle. Son nombre d'arêtes est alors égal au nombre de sommets moins un.*
Une forêt *est un graphe sans cycle. Les composantes connexes d'une forêt sont donc des arbres.*

Le graphe $\mathscr{G}$ est une forêt.

Définition C.1.9 (frontière et couverture de Markov) *La* frontière de Markov *d'un nœud X_i est l'ensemble des sommets adjacents à ce dernier et est notée $F(X_i) = Adj(X_i)$.*
La couverture de Markov *d'un nœud X_i est $M(X_i) = F(X_i) \cup \{X_i\}$.*

La frontière de Markov du nœud C est l'ensemble $\{A, D, E\}$ tandis que la couverture de Markov du nœud C est l'ensemble $\{A, C, D, E\}$.

C.2 Graphes orientés

Définition C.2.1 (graphe orienté) *Un graphe est dit* orienté *(ou* dirigé[v]*) s'il vérifie de plus $(\mathscr{E}(X_i, X_j) = 1 \Rightarrow \mathscr{E}(X_j, X_i) = 0)$, $1 \leqslant i, j \leqslant n$.*

Nous allons illustrer les notions suivantes à partir du graphe orienté de la figure C.3 que nous nommerons $\mathscr{D}$.

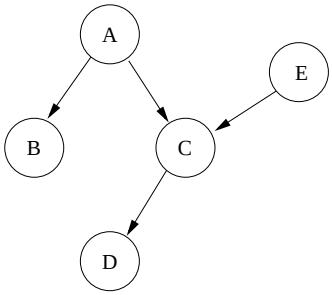


FIG. C.3 : *Un exemple de graphe orienté, nommé $\mathscr{D}$.*

Définition C.2.2 (arc, origine, extrémité) *Un* arc *est un couple de nœuds (X_i, X_j) vérifiant $\mathscr{E}(X_i, X_j) = 1$ et $\mathscr{E}(X_j, X_i) = 0$, le nœud X_i sera alors appelé l'*origine *de l'arc, et X_j son* extrémité*. Pour la représentation graphique, une flèche sera tracée de X_i vers X_j, comme ceci* $(X_i) \longrightarrow (X_j)$.

L'arc AB est d'origine A et d'extrémité B.

Définition C.2.3 (parent, nœud fils, descendant, ancêtre) *Dans ce cas, le nœud X_i sera également appelé le* parent *(ou le prédécesseur) de X_j et X_j sera dit l'*enfant *ou le nœud* fils *(ou le successeur) de X_j. L'ensemble des parents du nœud X_i sera noté $Pa(X_i)$ et l'ensemble de ses enfants $Enf(X_i)$*
Un nœud est dit descendant *d'un autre s'il appartient à l'ensemble itéré des successeurs de celui-ci. L'ensemble des descendants du nœud X_i sera noté $Desc(X_i)$.*
Un nœud est dit ancêtre *d'un autre s'il appartient à l'ensemble itéré des prédécesseurs de celui-ci.*

[iv] Un arbre non orienté est nommé *tree* par les anglo-saxons, mais un arbre orienté est appelé *polytree*. Remarquons par ailleurs que la définition d'*arbre* est non dépendante de l'orientation, par exemple, un arbre peut être semi-orienté.

[v] aussi appelé simplement *digraph* en anglais.

Nous avons $Enf(A) = \{B, C\}$, $Desc(A) = \{B, C, D\}$ et $Pa(C) = \{A, E\}$ dans $\mathscr{D}$.

Définition C.2.4 (racine, feuille) *Un sommet n'ayant pas de parent sera appelé une* racine *(ou une* source*). Un sommet n'ayant pas de fils sera appelé une* feuille*.*

Le nœud E est une racine et le nœud D est une feuille.

Définition C.2.5 (chemin, circuit) *Pour un graphe $\mathscr{G} = (X, \mathscr{E})$, on appelle* chemin *une suite $(u_1, \cdots, u_k)$ d'arcs telle que l'extrémité de chaque arc coïncide avec l'origine du suivant. Un* circuit *est un chemin pour lequel le sommet initial et le sommet terminal sont identiques.*

L'entier naturel non nul k est alors appelé la *longueur* du chemin.

Un chemin peut également être identifié par la suite des nœuds $(X_1, \cdots, X_{k+1})$ par lesquels il passe.
Le graphe $\mathscr{D}$ ne possède pas de circuit, et ACE est un de ses chemins.

Définition C.2.6 (arborescence) *Une* arborescence[vi] *est un graphe orienté sans circuit disposant d'un sommet racine unique. Remarquons que le graphe non orienté sous-jacent est un arbre.*

Le graphe $\mathscr{D}$ n'est pas une arborescence car il possède deux racines (A et E).

Définition C.2.7 (arête, squelette) *Lorsque nous utiliserons le terme d'*arête *pour un graphe orienté $\mathscr{G} = (X, \mathscr{E})$, c'est que nous prendrons en considération le graphe non orienté qu'il induit par symétrisation de l'application $\mathscr{E}$.*
Le squelette *d'un graphe $\mathscr{G} = (X, \mathscr{E})$ est alors le graphe $\mathscr{S} = (X, \mathscr{E}')$ où $\mathscr{E}'(X_i, X_j) = \max(\mathscr{E}(X_i, X_j), \mathscr{E}(X_j, X_i)) = \mathscr{E}'(X_j, X_i)$.*

Le squelette d'une arborescence est alors un arbre, mais remarquons ici que le squelette de $\mathscr{D}$ est également un arbre, celui de la figure C.2(b).

Dans le cadre orienté, la frontière de markov d'un sommet peut s'écrire

$$F(X_i) = Pa(X_i) \cup Desc(X_i)$$

[vi] Une arborescence est nommée *rooted tree* par les anglo-saxons.

D

Solutions logiciels

D.1 Différentes solutions existantes

Il existe différents logiciels qui permettent l'utilisation de réseaux bayésiens. Ceux-ci supportent l'inférence, mais ne proposent alors que peu de moteurs d'inférence en général. Ils permettent également de faire de l'apprentissage des paramètres, et proposent parfois quelques méthodes pour effectuer l'apprentissage de structure. Voici une liste (non-exhaustive) de logiciels permettant de manipuler les réseaux bayésiens.

- **gR** Lauritzen, Badsberg, BÃ¸ttcher, Dalgaard, Dethlefsen, Edwards, Eriksen, Gregersen, HÃ¸jsgaard & Kreiner (2004) : Le projet *Graphical models in R* regroupe plusieurs projets consistant en la mise en œuvre de modèles graphiques avec le langage R. Par exemple, l'implémentation de DEAL Bøttcher & Dethlefsen (2004) fait partie de gR.
- **BNT** Murphy (2004) : La *Bayes Net Toolbox* pour Matlab, est issue du travail de Kevin Murphy. Elle propose de nombreuses fonctions pour utiliser les réseaux bayésiens, et en particulier les réseaux bayésiens dynamiques.
- **PNL** Bradski (2004) : La *Probabilistic Network Library* est un projet *open source* mené par la société Intel. Cette bibliothèque contient de nombreuses fonctions dans le langage C, certaines sont des traductions des fonctions de la *bayes Net Toolbox*.
- **BNJ** Perry & Stilson (2002) : Le *Bayesian Network tools in Java* est un projet Java sous licence GPL.
- **TETRAD** Spirtes, Glymour & Scheines (2004) : Il s'agit d'un projet de l'équipe de Peter Spirtes.
- **Causal explorer** Tsamardinos, Brown & Aliferis (2005) : Une toolbox pour Matlab.
- **LibB** Friedman & Elidan (1999) : Il s'agit d'un projet de l'équipe de Nir Friedman.
- **BNPC** Cheng, Hatzis, Hayashi, Krogel, Morishita, Page & Sese (2001) : Il s'agit d'un projet de l'équipe de Jie Cheng.
- **Web WeavR** Xiang (1999) : Il s'agit d'un projet de l'équipe de Yang Xian.
- **JavaBayes** Drakos & Moore (1998) : Il s'agit d'un projet de l'équipe de Fabio Gagliardi Cozman.
- **ProBayes** Mazer, Miribel, Bessière, Lebeltel, Mekhnacha & Ahuactzin (2004) : Produit issu des travaux de l'équipe d' Émmanuel Mazer.
- **BayesiaLab** Munteanu, Jouffe & Wuillemin (2001) : Produit issu des travaux de l'équipe de Paul Munteanu.

- **Hugin** Andersen, Olesen, Jensen & Jensen (1989) : Depuis 1989, *Hugin Expert A/S* est une des sociétés phares proposant un logiciel permettant de manipuler les réseaux bayésiens.
- **Netica** Netica (1998) : *Norsys Software Corp.* est la concurente directe de Hugin depuis 1995.
- **BayesWare** Sebastiani, Ramoni & Crea (1999) : Produit de la société *Bayesware*.
- **MSBNx** Kadie, Hovel & Horvitz (2001) : Un projet gratuit de l'équipe de *Microsoft Research*.
- **B-Course** Myllymäki, Silander, Tirri & Uronen (2002) : Permet d'avoir accès à des outils de recherche de structure de réseaux bayésiens directement en pages ouèbes.
- **Bayes Builder** Nijman, Akay, Wiegerinck & Nijmegen (2002) : Le moteur d'inférence utilisé pour le projet PROMEDAS (*PRObabilistic MEdical Diagnostic Advisory System*).

D.2 SLP : Le *Structure Learning Package* pour BNT

Toutes les fonctions issues de ces travaux de doctorat, ainsi que de nombreuses autres sont disponibles en ligne[i] sur le site français de la *Bayes Net Toolbox* Leray, Guilmineau, Noizet, François, Feasson & Minoc (2003) sous la forme du *Structure Learning Package* Leray & François (2004b), une bibliothèque de fonction d'apprentissage de structure pour la *Bayes Net Toolbox* Murphy (2004), également disponible librement.

A présent, listons les principales fonctions disponibles dans ce *Structure Learning Package*.

- add_SLP
- \datas (répertoire contenant toutes les bases d'exemples utilisées ainsi que les réseaux bayésiens utilisés pour générer des bases)
- \documentation
- \examples (répertoire contenant des scripts de tests pour certaines fonctions)
- learn_struct_EM
- learn_struct_ges
- learn_struct_gs
- learn_struct_hc
- learn_struct_mwst
- learn_struct_mwst_EM
- learn_struct_pdag_bnpc
- learn_struct_tan
- learn_struct_tan_EM
- mk_naive_struct
- CPT_from_bnet
- bnt_to_mat
- classification_evaluation
- compute_bnet_nparams
- confiance_interval
- cpdag_to_dag
- dag_to_cpdag
- discretization
- editing_dist
- find_nodes_in_undirected_component
- gener_MAR_net
- gener_MCAR_net
- gener_data_from_bnet_miss
- gener_empty_cache
- hist_ic
- histc_ic
- inference
- isdag
- ismemberclique
- mat_to_bnt
- mk_nbrs_of_dag_topo
- mk_nbrs_of_pdag_add
- mk_nbrs_of_pdag_del
- pdag_to_dag
- cond_indep_chisquare
- cond_mutual_info_score
- kl_divergence
- mutual_info_score
- score_add_to_cache
- score_dags
- score_family
- score_find_in_cache
- score_init_cache

[i] http://banquiseasi.insa-rouen.fr/projects/bnt-slp/

E

Bases d'exemples utilisées

Toutes les bases d'exemples utilisées ici sont disponibles en ligne soit sur Blake & Merz (1998), soit sur Michie, Spiegelhalter & Taylor (1994).

E.1 Bases d'exemples complètes disponibles

Australian :
Cette base consiste en une évaluation de la possibilité de crédit accordée aux clients australiens d'une certaine banque en fonction de 14 attributs. Elle contient 690 exemples qui ont été séparés en 500 pour l'apprentissage et 190 pour le test.

Car :
Cette base est issue d'un simple modèle de décision hiérarchique utilisé originalement dans Bohanec & Rajkovic (1990). Elle possède 6 attributs comme par exemple le prix ou le nombre de portes d'un véhicule. Il s'agit de classer un véhicule parmi quatre classes qui sont : non-acceptable, acceptable, bien, très bien. La base contient 1728 entrées, qui ont été séparées en 1152 pour l'apprentissage et 576 pour le test.

Contrasep :
Voici un extrait d'une enquête nationale indonésienne sur la prédominance des méthodes de contraception Lim, Loh & Shih (2000). Les exemples sont des femmes mariées qui ne sont pas enceintes. L'objectif est de choisir une méthode de contraception adaptée pour ces femmes en fonction de leurs caractéristiques socio-économique et démographique. Les neuf attributs sont par exemple l'âge, le niveau d'étude, le nombre d'enfants ou encore la religion. Cette base contient 1473 exemples, les 850 premiers étant utilisés pour l'apprentissage, et les 623 derniers pour le test.

Diabetes :
La base *Pima Indians Diabetes Database* contient 768 exemples répartis en deux classes. Chaque exemple est décrit par 7 attributs. La base a été séparée en une base d'apprentissage contenant les 400 premiers exemples et une base de tests contenant les 368 derniers exemples Smith, Everhart, Dickson, Knowler & Johannes (1988). Les exemples représentent des femmes indiennes de plus de 21 ans. A partir d'attributs comme le nombre de fois qu'elles ont été enceintes, le taux de glucose ou encore la pression sanguine, il faut déterminer si elles sont atteintes de diabète.

German :
A partir de 24 attributs représentant le profil financier d'un individu, nous devons déterminer si la banque peut accorder ou non un crédit. La base contient 1000 exemples, nous avons pris les 600 permiers pour l'apprentissage, et les 400 autres pour le test.

Heart :
Il s'agit ici de prédire la présence ou l'absence de maladie du coeur à partir de 13 attributs représentant l'état physiologique du patient. Cette base contient 270 exemples, nous avons pris les 150 premiers pour l'apprentissage et les 120 derniers pour le test.

Letter :
Il s'agit d'une base d'exemples qui a été créée à partir d'images de lettres alphabétiques manuscrites. Elle contient 16 attributs comme la position et la hauteur de la lettre, mais aussi les moyennes ou les variances des pixels en x et en y Frey & Slate (1991). La variable représentant la classe peut donc prendre 26 valeurs différentes et la base contient 15000 exemples d'apprentissage et 5000 de test.

***The Monk's Problems* :**
Les problèmes MONK ont été la base de la première journée de comparaison des algorithmes d'apprentissage. Les résultats sont résumés dans Thrun (1991) La base contient initialement 432 exemples décris par six attributs (dont la classe) et sert de base de test. Pour l'apprentissage, le premier problème contient les exemples de la base de tests vérifiant $(X_1 = X_2)$ ou $(X_5 = 1)$, le deuxième contient les exemples vérifiant exactement deux des assertions suivantes : $\{X_1 = 1, X_2 = 1, X_3 = 1, X_4 = 1, X_5 = 1, X_6 = 1\}$, et la troisème base d'apprentissage contient les exemples vérifiant $(X_5 = 3 \text{ et } X_4 = 1)$ ou $(X_5 \neq 4 \text{ et } X_2 \neq 3)$, et 5% de bruit ont été ajoutés à cette base d'apprentissage.

Nursery :
Cette base est issue d'un modèle de décision hiérarchique initialement développé pour classer les écoles maternelles. La décision finale dépend de la structure familiale : ses ressources, son niveau social, sa santé Olave, Rajkovic & Bohanec (1989). Cette base contient 12960 exemples. Nous en avons pris 8500 pour l'apprentissage et 4460 pour le test. Les cinq classes vont de la forte priorité d'inscrire l'enfant dans une école maternelle à la non-recommandation.

Pen :
Cette base contient une collection de 250 chiffres rédigés par 44 scripteurs différents. La base d'apprentissage est constituée des exemples issus de 30 scripteurs et est de taille 7494. La base de tests est constituée des exemples issus de 14 autres scripteurs et regroupe 3498 exemples. Il s'agit donc d'identifier un chiffre de zéro à neuf à partir de 16 caractéritiques numériques issues de mesures lors de l'écriture (pression, vitesse, coordonnées, *etc*) Alimoglu & Alpaydin (1996).

Spect :
Cette base décrit le diagnostic de faiblesse cardiaque (classe binaire, normal et anormal) à partir d'informations extraites d'images issues d'un *Single Proton Emission Computed Tomography* (SPECT) Kurgan, Cios, Tadeusiewicz, Ogiela & Goodenday (2001). La base

contient 267 descriptions de patients sur 22 attributs binaires. Elle a été séparée en une base de 80 exemples pour l'apprentissage et une de 187 exemples pour le test.

Segment :
Les exemples de cette base sont issus d'une base de sept images d'extérieurs, les 7 classes : briques, ciel, herbe, *etc*. Ces images ont été segmentées à la main en region de 3×3 pixels. La base contient 2310 exemples décrits par 19 attributs continus qui ont été discrétisés. Nous avons pris les 1400 premiers pour l'apprentissage, et les 910 restants pour le test.

Tae :
Cette base consiste en une évaluation des performances de 151 enseignants du département de statistiques de l'université de Wisconsin-Madison Lim, Loh & Shih (2000). Les scores sont séparés en trois catégories : bas, normal, haut. Cette base contient 6 attributs (dont la classe). Nous avons pris les 100 premiers exemples pour l'apprentissage et les 51 derniers pour le test.

Wine :
Cette base contient les résultats d'une analyse chimique de vins produits dans la même région d'Italie, mais par trois producteurs différents (les 3 classes). Cette analyse détermine les quantités de 13 constituants retrouvés dans chaque type de vin (alcool, magnésium, phenols, *etc*) Aeberhard, Coomans & de Vel (1992). A partir de cette base, une base d'apprentissage contenant 80 exemples a été créée, ainsi qu'une autre de 98 exemples pour le test.

Zoo :
Cette base contient 16 valeurs binaires et une numérique. Il s'agit de reconnaître de quelle famille (parmi 7) est un animal à partir de cette description. La base contient 101 exemples, 60 ont été pris pour l'apprentissage, et 41 pour le test.

E.2 Bases d'exemples incomplètes disponibles

Hepatitis :
Cette base contient une étude menée sur 155 patients. 20 grandeurs ont étés reportées, et il s'agit de dire si un patient atteint de l'hépatite est mort. La distribution sur la classe est 32 patients morts et 123 vivants.

Horse :
Cette base contient 28 grandeurs physiques liées à la santé du cheval. Tous ces chevaux sont atteints de la colique du cheval et il faut reconnaître s'ils ont été traités avec ou sans chirurgie. Elle possède 300 exemples d'apprentissage et 68 exemples de test.

House :
Il s'agit ici d'essayer de reconnaître un démocrate d'un républicain lors d'un vote au congrès américain. Ces individus devaient répondre par oui ou par non à 16 questions politiques alors d'actualité. Bon nombre de politicien sont restés vagues et n'ont alors donné ni réponse positive ni réponse négative à certaines questions. Cette base contient alors les réponses de 435 personnes dont 267 démocrates et 168 républicains.

Mushrooms :
Cette base contient la description de 23 espèces de champignons des familles *Agaricus* et *Lepiota*. Chaque espèce peut alors être classée comestible ou dangeureuse. Elle contient alors la descrition de 8124 champignons sur 22 caractéristiques (dont 4208 commestibles).

Soybean :
Nous devons ici identifier à quelle famille appartiennent différentes graines de soja en fonction de leur description sur 35 attributs. La base contient 376 exemples d'apprentissage pour 376 exemples de tests. Il y a alors 19 classes à identifier.

Thyroid :
C'est une base de diagnostic médical pour laquelle nous avons utilisé 22 variables (parmi 29) : 15 variables discrètes, 6 continues qui ont été discrétisées, et la classe. Il y a 2800 exemples d'apprentissage et 972 de tests.

F

Tableaux de résultats

F.1 Expérimentations à partir de bases complètes générées

Dans cette section sont présentés les résultats d'identification de structure à partir de bases de données complètes générées à partir de réseaux bayésiens connus. Le contenu d'une cellule est décrit dansle cadre suivant.

Score BIC $\pm$ écart-type ; Distance d'édition
Divergence de Kullback-Leiber $\pm$ écart-type ; Temps de calcul

Remarquons que sur les tableaux de résultats les cellules contiennent ces informations pour plusieurs tailles de la base d'exemples.

Les distances d'édition et la divergence de Kullback-Leiber sont données par rapport au réseau bayésien d'origine qui est connu pour cette première phase d'expérimentations. Le temps de calcul est en secondes et est donné à titre indicatif.

Toutes les valeurs données sont des moyennes sur 20 lancements, chacun sur une base d'apprentissage différente qui a servi pour toutes les méthodes.

bases	tailles					Indépendances conditionnelles		MWST		K2		
	N	score	app	test	moy	PC	BNPC	IM	BIC	Rnd	+T	-T
Jouet1	5	-8236.9	100	2000	3	-8569.4±80.20 ; 4.9	-8490.7±206.60 ; 3.5	**-8244.6**±2.21 ; 3.0	-8256.8±66.42 ; 3.1	-8473.4±203.28 ; 3.0	-8325.1±153.41 ; 2.8	-8628.5±194.06 ; 2.7
						0.6363±0.1291 ; 0.16	0.5863±0.1196 ; 0.33	0.7200±0.1200 ; 0.06	0.7170±0.1179 ; 0.12	0.7097±0.1485 ; 0.15	0.7355±0.1358 ; 0.15	0.7306±0.1594 ; 0.15
Jouet1	5	-8236.9	300	2000	3	-8357.1±42.45 ; 3.5	-8368.7±1.67 ; 3.8	-8241.9±0.91 ; 1.9	-8242.0±0.06 ; 2.9	-8339.3±165.54 ; 2.9	**-8238.9**±0.71 ; 2.5	-8420.5±159.28 ; 3.2
						0.6983±0.0752 ; 0.33	0.7012±0.0745 ; 0.41	0.7097±0.0780 ; 0.06	0.7060±0.0748 ; 0.13	0.7090±0.0850 ; 0.15	0.7080±0.0750 ; 0.16	0.7021±0.0724 ; 0.16
Jouet1	5	-8236.9	500	2000	3	-8264.0±38.58 ; 1.8	-8240.8±1.36 ; 3.0	-8242.1±1.36 ; 1.6	-8241.9±0.09 ; 2.7	-8337.9±151.11 ; 3.1	**-8238.8**±0.00 ; 2.2	-8384.9±133.83 ; 3.6
						0.7416±0.0519 ; 0.63	0.7552±0.0527 ; 0.39	0.7491±0.0551 ; 0.07	0.7443±0.0504 ; 0.13	0.7513±0.0529 ; 0.17	0.7457±0.0524 ; 0.17	0.7341±0.0657 ; 0.17
Jouet2	6	-9570.6	100	2000	4	-12263.4±1903.80 ; 7.5	-11808.8±2087.77 ; 7.0	-9719.4±136.29 ; 1.9	-9747.5±253.45 ; 1.6	-10040.7±360.14 ; 3.1	-9702.2±115.56 ; 1.6	-10138.3±205.28 ; 4.0
						20.0832±17.7653 ; 0.30	6.0529±6.1116 ; 0.36	5.1167±0.5448 ; 0.08	7.6651±5.0777 ; 0.17	8.9093±9.8829 ; 0.24	5.1340±0.5359 ; 0.23	20.0817±12.4040 ; 0.23
Jouet2	6	-9570.6	300	2000	4	-10081.9±391.08 ; 2.8	-9931.9±211.62 ; 3.5	**-9624.9**±0.00 ; 1.2	**-9624.9**±0.00 ; 1.6	-9938.0±231.28 ; 3.3	**-9624.9**±0.00 ; 1.2	-9999.7±163.18 ; 4.8
						5.3711±1.1395 ; 0.69	3.7473±2.2587 ; 0.77	1.3125±0.1204 ; 0.08	1.3639±0.0857 ; 0.18	8.0059±10.8015 ; 0.26	1.3125±0.1204 ; 0.25	14.0097±13.4529 ; 0.26
Jouet2	6	-9570.6	500	2000	4	-9658.5±77.00 ; 0.5	-9848.7±25.11 ; 2.9	**-9624.9**±0.00 ; 1.2	**-9624.9**±0.00 ; 1.4	-9853.7±214.60 ; 3.5	**-9624.9**±0.00 ; 1.2	-9977.1±154.90 ; 5.3
						0.5264±0.0434 ; 1.24	1.9397±1.5809 ; 0.93	0.5331±0.0430 ; 0.08	0.5240±0.0492 ; 0.18	5.3658±8.3299 ; 0.25	0.5331±0.0430 ; 0.24	8.5020±10.0073 ; 0.25
Jouet2	6	-9570.6	1000	2000	4	**-9624.9**±0.00 ; 0.0	**-9624.9**±0.00 ; 2.0	**-9624.9**±0.00 ; 1.3	**-9624.9**±0.00 ; 1.2	-9782.3±181.51 ; 3.3	**-9624.9**±0.00 ; 1.3	-9847.8±119.49 ; 5.2
						-0.0853±0.0235 ; 2.61	-0.1168±0.0222 ; 0.87	-0.0937±0.0314 ; 0.10	-0.0892±0.0271 ; 0.20	0.4046±0.9095 ; 0.28	-0.0937±0.0314 ; 0.27	1.2792±1.1275 ; 0.29
Jouet3	5	-11162.5	100	2000	4	-14188.6±236.25 ; 7.0	-12918.6±886.56 ; 5.5	**-11552.9**±139.67 ; 3.2	-11645.8±72.85 ; 2.8	-12105.5±252.24 ; 4.0	-12057.2±259.74 ; 4.2	-12226.0±203.43 ; 4.3
						10.2442±3.0784 ; 0.16	10.5284±2.9030 ; 0.19	14.1345±3.5277 ; 0.06	10.4477±3.5509 ; 0.11	14.0454±3.5177 ; 0.13	13.9695±3.5181 ; 0.13	14.0595±3.4523 ; 0.13
Jouet3	5	-11162.5	300	2000	4	-13303.6±816.70 ; 6.5	-12260.3±689.46 ; 4.2	**-11508.3**±0.00 ; 2.5	-11576.5±63.30 ; 2.1	-11927.2±229.60 ; 4.2	-11724.2±67.72 ; 3.2	-12057.2±163.87 ; 4.4
						6.8496±1.9186 ; 0.33	7.1074±1.7067 ; 0.48	13.8190±3.3019 ; 0.06	11.1836±3.8211 ; 0.12	12.5782±2.9889 ; 0.15	12.5350±3.2853 ; 0.14	12.9736±2.6811 ; 0.14
Jouet3	5	-11162.5	500	2000	4	-11525.7±400.71 ; 4.8	-11822.3±98.13 ; 3.1	-11508.3±0.00 ; 2.5	-11557.9±62.33 ; 3.0	-11578.0±250.45 ; 3.5	-11388.7±298.38 ; 2.1	-11692.6±257.71 ; 4.5
						4.1625±1.1965 ; 0.76	6.1023±1.3453 ; 1.15	13.2623±2.5456 ; 0.07	10.7573±3.9524 ; 0.12	8.8987±4.8490 ; 0.17	7.9756±5.0985 ; 0.17	8.7046±4.2400 ; 0.17
Jouet3	5	-11162.5	1000	2000	4	-11783.6±268.54 ; 3.7	-11865.0±66.82 ; 3.1	-11508.3±0.00 ; 2.6	-11508.3±0.00 ; 2.7	-11506.5±241.59 ; 3.2	-11268.5±200.87 ; 2.0	-11546.3±244.45 ; 4.4
						3.4560±0.6182 ; 1.91	3.7994±0.6093 ; 2.63	12.8625±1.1991 ; 0.07	12.8502±1.2337 ; 0.14	7.7430±5.1681 ; 0.19	6.3731±5.3011 ; 0.19	6.0863±4.5519 ; 0.19
Jouet4	5	-10071.0	100	2000	3	**-10307.9**±277.14 ; 5.6	-10425.8±230.92 ; 3.4	-10432.5±31.48 ; 3.1	-10419.8±51.97 ; 3.0	-10566.8±144.43 ; 3.4	-10514.8±135.99 ; 3.6	-10539.1±140.25 ; 3.2
						0.0453±0.0919 ; 0.13	0.0829±0.0907 ; 0.32	0.1292±0.1188 ; 0.05	0.1225±0.0964 ; 0.11	0.0924±0.1035 ; 0.15	0.1275±0.0978 ; 0.14	0.1009±0.0978 ; 0.14
Jouet4	5	-10071.0	300	2000	3	**-10093.4**±59.26 ; 4.8	-10581.3±553.76 ; 2.5	-10410.6±18.86 ; 3.0	-10402.9±0.00 ; 3.5	-10429.0±149.15 ; 3.0	-10440.4±74.30 ; 3.9	-10413.3±125.24 ; 3.3
						0.0339±0.0474 ; 0.49	0.0493±0.0762 ; 0.92	0.1049±0.0704 ; 0.06	0.1163±0.0706 ; 0.11	0.0552±0.0575 ; 0.16	0.0942±0.0748 ; 0.16	0.0712±0.0554 ; 0.16
Jouet4	5	-10071.0	500	2000	3	-10181.2±245.81 ; 5.2	-10431.0±304.11 ; 3.0	-10410.6±18.86 ; 3.2	-10402.9±0.00 ; 3.4	-10359.3±122.63 ; 3.4	-10336.9±101.64 ; 4.0	-10396.6±156.05 ; 3.1
						0.0020±0.0476 ; 0.93	0.0879±0.0520 ; 1.90	0.1205±0.0635 ; 0.06	0.1310±0.0512 ; 0.12	0.0785±0.0537 ; 0.17	0.0919±0.0661 ; 0.16	0.0789±0.0615 ; 0.17
Jouet4	5	-10071.0	1000	2000	3	**-10074.1**±0.00 ; 4.5	-10095.7±70.87 ; 1.2	-10408.1±15.84 ; 3.2	-10402.9±0.00 ; 3.0	-10300.5±164.29 ; 2.8	-10263.1±88.70 ; 3.9	-10312.0±109.54 ; 3.1
						0.0060±0.0306 ; 2.04	0.0118±0.0339 ; 4.35	0.1085±0.0399 ; 0.08	0.1150±0.0298 ; 0.14	0.0430±0.0486 ; 0.20	0.0558±0.0453 ; 0.19	0.0537±0.0423 ; 0.20
Jouet5	7	-9221.2	100	2000	4	-10510.1±811.43 ; 7.5	-11121.2±774.61 ; 3.8	**-9973.1**±72.31 ; 5.6	**-9973.1**±72.31 ; 6.0	-10096.7±121.29 ; 5.7	-10043.8±107.75 ; 6.7	-10055.6±141.38 ; 5.0
						0.2323±0.1292 ; 0.28	0.1419±0.1421 ; 0.68	0.4388±0.1641 ; 0.09	0.4311±0.1573 ; 0.21	0.3162±0.2150 ; 0.29	0.3941±0.1313 ; 0.29	0.2612±0.1715 ; 0.29
Jouet5	7	-9221.2	300	2000	4	-10363.2±798.21 ; 7.0	-10886.3±693.35 ; 4.1	**-9942.7**±9.95 ; 5.7	**-9942.7**±9.95 ; 5.5	-10023.6±131.55 ; 6.5	-10039.3±136.68 ; 8.1	-10101.0±119.23 ; 5.6
						0.1448±0.0752 ; 1.06	0.1635±0.0501 ; 0.71	0.0040±0.0683 ; 0.10	0.0060±0.0702 ; 0.22	0.0830±0.1547 ; 0.34	0.1270±0.0943 ; 0.34	0.0086±0.1281 ; 0.34
Jouet5	7	-9221.2	500	2000	4	-11149.5±1100.50 ; 7.0	-10601.6±949.33 ; 2.4	-9939.3±9.85 ; 5.2	-9939.3±9.85 ; 5.2	-10112.0±146.88 ; 6.2	-10069.0±184.75 ; 8.6	-10119.3±149.37 ; 5.6
						0.2367±0.0944 ; 2.28	0.2483±0.1011 ; 1.29	0.1343±0.0703 ; 0.11	0.1404±0.0767 ; 0.24	0.0451±0.1602 ; 0.38	0.0281±0.0881 ; 0.38	0.1071±0.1250 ; 0.38
Jouet5	7	-9221.2	1000	2000	4	-12147.0±345.04 ; 7.0	-11669.8±936.27 ; 2.1	-9935.5±5.69 ; 4.8	-9935.5±5.69 ; 5.0	-10260.2±206.10 ; 6.5	-10446.6±215.60 ; 8.8	-10263.2±92.42 ; 6.3
						0.2759±0.0415 ; 5.72	0.3067±0.0483 ; 2.26	0.2783±0.0568 ; 0.13	0.2795±0.0526 ; 0.27	0.2061±0.1385 ; 0.46	0.1310±0.0750 ; 0.47	0.2201±0.0869 ; 0.45
Jouet5	7	-9221.2	2000	2000	4	-12119.5±467.16 ; 6.7	-11987.2±648.17 ; 2.1	-9934.1±4.26 ; 4.9	-9934.1±4.26 ; 4.8	-10756.1±839.59 ; 7.0	-10584.6±236.06 ; 9.2	-11302.9±2535.24 ; 5.5
						0.3575±0.0254 ; 11.15	0.3743±0.0277 ; 6.50	0.3600±0.0357 ; 0.17	0.3599±0.0335 ; 0.34	0.3157±0.0635 ; 0.61	0.2463±0.0459 ; 0.59	0.3389±0.0448 ; 0.58

TAB. F.1 : *Résultats pour les exemples jouets 1 à 5 et pour les méthodes PC, BNPC, MWST-im, MWST-bic, K2(R), K2+T et K2-T.*

bases	tailles					GS-BIC			GS-BD	GES	
	N	score	app	test	moy	0	chaîne	+T	+T	BD	BIC
Jouet1	5	-8236.9	100	2000	3	-8357.6±165.69 ; 2.4	-8357.6±165.69 ; 3.0	-8357.6±165.69 ; 2.8	-8325.1±153.41 ; 2.8	-8559.2±113.50 ; 1.9	-8357.2±165.93 ; 1.4
						0.7348±0.1334 ; 3.90	0.7372±0.1320 ; 3.87	0.7421±0.1329 ; 2.99	0.7336±0.1358 ; 2.80	0.7632±0.1222 ; 2.56	0.7268±0.1393 ; 2.31
Jouet1	5	-8236.9	300	2000	3	-8239.1±0.98 ; 2.2	-8239.1±0.98 ; 2.5	-8239.1±0.98 ; 2.4	-8239.1±0.98 ; 2.5	-8446.0±173.57 ; 1.6	-8239.1±0.98 ; 1.1
						0.7064±0.0726 ; 4.51	0.7073±0.0727 ; 4.95	0.7087±0.0741 ; 3.23	0.7076±0.0745 ; 2.65	0.7419±0.0843 ; 2.90	0.7025±0.0723 ; 2.60
Jouet1	5	-8236.9	500	2000	3	**-8238.8**±0.00 ; 1.7	**-8238.8**±0.00 ; 2.5	**-8238.8**±0.00 ; 2.0	**-8238.8**±0.00 ; 2.5	**-8238.8**±0.00 ; 1.0	**-8238.8**±0.00 ; 1.0
						0.7427±0.0505 ; 4.67	0.7442±0.0505 ; 4.82	0.7441±0.0495 ; 3.46	0.7464±0.0521 ; 2.92	0.7413±0.0507 ; 2.93	0.7413±0.0507 ; 2.66
Jouet2	6	-9570.6	100	2000	4	-10247.4±495.29 ; 2.8	-10257.5±481.44 ; 3.2	-10207.6±513.64 ; 2.8	-9673.1±100.19 ; 1.6	-10534.8±455.80 ; 2.4	-9698.9±130.47 ; 0.4
						9.5419±7.5359 ; 8.17	9.3672±7.6415 ; 11.96	9.3780±7.5211 ; 5.70	5.2182±0.3444 ; 4.12	16.5932±11.9909 ; 5.99	5.3176±0.0718 ; 7.05
Jouet2	6	-9570.6	300	2000	4	-9635.0±45.32 ; 1.9	-9700.6±124.77 ; 2.3	**-9624.9**±0.00 ; 1.6	**-9624.9**±0.00 ; 1.2	-9851.2±12.15 ; 1.0	**-9624.9**±0.00 ; 0.0
						1.3194±0.1280 ; 10.12	5.2106±8.6834 ; 14.57	1.3286±0.1330 ; 4.52	1.3282±0.1137 ; 3.67	1.4380±0.0552 ; 8.02	1.3824±0.0253 ; 7.34
Jouet2	6	-9570.6	500	2000	4	-9637.1±29.89 ; 2.2	-9858.2±189.35 ; 3.1	**-9624.9**±0.00 ; 1.6	**-9624.9**±0.00 ; 1.1	-9854.5±19.93 ; 1.0	**-9624.9**±0.00 ; 0.0
						0.4935±0.0476 ; 10.13	7.6121±8.3531 ; 12.66	0.5129±0.0528 ; 4.28	0.5286±0.0502 ; 3.67	0.5826±0.0540 ; 7.88	0.5378±0.0299 ; 7.01
Jouet2	6	-9570.6	1000	2000	4	-9633.0±25.11 ; 1.9	-9911.7±19.30 ; 4.2	**-9624.9**±0.00 ; 1.4	**-9624.9**±0.00 ; 1.4	**-9624.9**±0.00 ; 0.0	**-9624.9**±0.00 ; 0.0
						0.0888±0.0265 ; 12.27	15.9248±2.2545 ; 14.57	0.0899±0.0251 ; 5.71	0.0958±0.0329 ; 3.79	0.0853±0.0235 ; 8.92	0.0853±0.0235 ; 7.88
Jouet3	5	-11162.5	100	2000	4	-11880.9±189.44 ; 3.5	-11880.9±189.44 ; 3.6	-11880.9±189.44 ; 3.5	-12057.2±259.74 ; 4.2	-12236.7±173.39 ; 3.4	-12137.9±231.58 ; 3.1
						13.9846±3.5652 ; 3.59	13.9646±3.5838 ; 4.23	13.9896±3.5828 ; 3.01	13.9894±3.5160 ; 2.82	13.7607±3.3054 ; 2.28	13.9344±3.4928 ; 1.82
Jouet3	5	-11162.5	300	2000	4	-11644.1±112.47 ; 3.2	-11675.1±82.59 ; 2.9	-11644.1±112.47 ; 2.9	-11724.2±67.72 ; 3.1	-11884.2±150.44 ; 2.5	-11816.8±130.55 ; 2.2
						12.2622±3.6182 ; 4.24	10.7514±3.5408 ; 4.24	12.2686±3.5715 ; 2.65	12.5280±3.2848 ; 2.54	13.3120±1.9993 ; 2.94	13.3304±1.9909 ; 2.50
Jouet3	5	-11162.5	500	2000	4	-11293.3±235.50 ; 2.0	-11308.6±258.86 ; 1.9	**-11293.3**±235.50 ; 1.8	-11428.9±272.59 ; 2.3	-11607.1±110.36 ; 1.3	-11465.8±303.01 ; 0.9
						5.9270±4.3253 ; 5.30	4.8106±2.5375 ; 4.50	5.5106±3.3798 ; 3.16	6.7136±3.2982 ; 2.79	12.4229±3.2318 ; 3.72	8.0595±4.6689 ; 3.07
Jouet3	5	-11162.5	1000	2000	4	-11168.6±146.41 ; 2.0	-11153.2±56.07 ; 3.0	-11188.6±164.01 ; 1.9	-11168.6±146.41 ; 1.6	**-11108.6**±0.00 ; 0.0	**-11108.6**±0.00 ; 0.0
						3.8632±4.1720 ; 6.21	2.1214±0.0991 ; 4.85	4.0592±3.9108 ; 3.41	3.7536±3.8792 ; 2.67	2.1731±0.0477 ; 4.16	2.1731±0.0477 ; 3.64
Jouet4	5	-10071.0	100	2000	3	-10531.2±111.02 ; 3.6	-10515.0±116.86 ; 3.8	-10498.7±105.89 ; 3.5	-10482.1±119.10 ; 3.5	-10527.9±96.02 ; 2.0	-10494.0±135.44 ; 1.9
						0.1190±0.1069 ; 3.81	0.1303±0.1077 ; 5.37	0.1419±0.0987 ; 2.36	0.1170±0.0847 ; 2.28	0.0949±0.0795 ; 2.96	0.0839±0.0709 ; 2.65
Jouet4	5	-10071.0	300	2000	3	-10296.4±89.29 ; 2.8	-10253.5±93.64 ; 2.4	-10271.4±91.08 ; 2.4	-10274.1±65.22 ; 2.4	-10197.4±92.78 ; 0.7	-10235.3±69.45 ; 0.8
						0.0673±0.0567 ; 5.02	0.0580±0.0577 ; 6.27	0.0632±0.0539 ; 2.97	0.0592±0.0789 ; 2.45	0.0163±0.0608 ; 3.56	0.0331±0.0596 ; 3.17
Jouet4	5	-10071.0	500	2000	3	-10203.9±58.26 ; 2.0	-10187.2±54.33 ; 1.9	-10213.6±65.14 ; 2.6	-10230.7±90.39 ; 2.4	**-10121.5**±84.23 ; 0.2	-10178.4±96.77 ; 0.6
						0.0674±0.0447 ; 5.77	0.0661±0.0430 ; 6.80	0.0693±0.0493 ; 3.64	0.0606±0.0453 ; 2.59	0.0178±0.0528 ; 3.89	0.0380±0.0571 ; 3.34
Jouet4	5	-10071.0	1000	2000	3	-10097.7±36.90 ; 0.9	-10126.2±99.79 ; 1.6	-10091.7±31.56 ; 1.2	-10080.6±17.72 ; 1.4	**-10074.1**±0.00 ; 0.0	**-10074.1**±0.00 ; 0.0
						0.0208±0.0429 ; 7.66	0.0253±0.0433 ; 8.36	0.0171±0.0396 ; 4.72	0.0110±0.0354 ; 3.53	0.0061±0.0306 ; 4.36	0.0061±0.0306 ; 3.79
Jouet5	7	-9221.2	100	2000	4	-10773.7±115.37 ; 6.8	-10773.7±115.37 ; 6.8	-10773.7±115.37 ; 6.9	**-9968.4**±122.40 ; 5.1	-10773.7±115.37 ; 6.7	-10830.2±81.03 ; 7.0
						0.0896±0.1443 ; 6.74	0.0935±0.1426 ; 17.78	0.0996±0.1488 ; 17.04	0.2983±0.1597 ; 11.06	0.0764±0.1332 ; 1.91	0.2583±0.1700 ; 8.81
Jouet5	7	-9221.2	300	2000	4	-9987.2±90.59 ; 5.7	-9977.6±85.60 ; 5.6	-9968.0±79.08 ; 5.5	-10019.0±144.89 ; 4.5	-10131.0±60.02 ; 5.0	-10080.7±11.26 ; 2.4
						0.0169±0.0881 ; 14.28	0.0284±0.0927 ; 22.65	0.0342±0.0875 ; 12.47	0.0804±0.0976 ; 13.16	0.0323±0.1468 ; 10.32	0.1680±0.0630 ; 9.09
Jouet5	7	-9221.2	500	2000	4	-9999.2±98.01 ; 5.5	-9941.4±53.40 ; 5.4	**-9929.4**±0.00 ; 5.5	-10116.3±112.01 ; 4.1	-10044.6±100.81 ; 4.6	-9913.0±0.00 ; 4.0
						0.1511±0.0947 ; 15.25	0.1920±0.0785 ; 24.78	0.1802±0.0571 ; 11.90	0.1480±0.1198 ; 14.53	0.0889±0.1408 ; 11.42	0.2641±0.1030 ; 9.76
Jouet5	7	-9221.2	1000	2000	4	-9832.2±50.54 ; 4.8	-9923.9±111.60 ; 6.2	**-9835.4**±49.70 ; 4.8	-10199.6±90.42 ; 3.5	-9799.3±32.01 ; 3.0	-9913.0±0.00 ; 4.0
						0.4107±0.0552 ; 22.06	0.3252±0.1229 ; 26.99	0.4121±0.0619 ; 14.30	0.2444±0.0962 ; 15.60	0.3983±0.0520 ; 15.29	0.2976±0.0777 ; 12.76
Jouet5	7	-9221.2	2000	2000	4	-9791.0±14.06 ; 5.1	-9849.4±89.65 ; 6.5	**-9787.1**±14.19 ; 4.8	-10217.4±91.81 ; 3.3	-9943.4±155.73 ; 3.0	-9813.0±37.62 ; 2.9
						0.4215±0.0394 ; 34.97	0.4011±0.0560 ; 43.52	0.4168±0.0418 ; 24.02	0.3217±0.0510 ; 19.75	0.3092±0.0622 ; 14.64	0.3480±0.0569 ; 16.36

TAB. F.2 : *Résultats pour les exemples jouets 1 à 5 et pour les méthodes GS-bic(0), GS-bic(C), GS-bic(T), GS-bd(T), GES-bd, GES-bic.*

bases	tailles					Indépendances conditionnelles		MWST		K2		
	N	score	app	test	moy	PC	BNPC	IM	BIC	Rnd	+T	-T
asia	8	-4516.0	100	2000	2	-5090.7±161.63 ; 6.0	-5029.9±123.09 ; 6.8	-5339.6±705.10 ; 8.0	-4716.4±59.91 ; 7.7	-4706.9±79.80 ; 9.0	-4663.9±57.34 ; 8.8	-4665.7±60.93 ; 6.0
						42.3272±24.0113 ; 0.12	49.8078±14.3848 ; 0.20	57.7544±42.3417 ; 0.05	25.1337±2.1991 ; 0.12	22.8981±8.0789 ; 0.20	19.5161±3.4025 ; 0.20	20.6757±2.1957 ; 0.19
asia	8	-4516.0	300	2000	2	-4862.5±133.07 ; 5.0	-4851.2±169.16 ; 5.0	-4730.4±322.19 ; 6.8	-4657.6±16.23 ; 6.8	-4609.8±37.98 ; 8.8	-4575.0±32.06 ; 7.5	-4590.6±37.34 ; 7.8
						13.8987±5.7907 ; 0.33	22.5079±10.3860 ; 0.48	15.7322±18.9120 ; 0.05	11.6330±2.2580 ; 0.13	7.8237±2.6373 ; 0.22	7.1480±1.9516 ; 0.21	7.2568±1.6920 ; 0.20
asia	8	-4516.0	500	2000	2	-4874.8±107.19 ; 5.4	-4683.8±159.20 ; 4.6	-4651.0±7.09 ; 5.4	-4651.0±7.09 ; 5.4	-4581.6±34.22 ; 9.1	**-4572.3**±23.64 ; 6.5	**-4570.7**±30.83 ; 8.0
						8.6028±5.1138 ; 0.67	10.9873±9.8284 ; 0.74	9.7946±2.2183 ; 0.05	9.7946±2.2183 ; 0.13	4.8414±1.0472 ; 0.23	4.7816±1.4987 ; 0.22	4.7969±0.9567 ; 0.22
asia	8	-4516.0	1000	2000	2	-4878.6±107.12 ; 5.8	-4697.7±173.82 ; 4.5	-4657.8±16.00 ; 5.5	-4657.8±16.00 ; 5.5	**-4562.5**±26.09 ; 8.0	**-4554.0**±23.38 ; 6.3	**-4564.2**±19.07 ; 8.5
						5.9479±5.6450 ; 1.58	9.5611±10.4968 ; 1.54	7.6991±1.1063 ; 0.06	7.6991±1.1063 ; 0.15	2.6073±0.4790 ; 0.24	2.2078±0.3686 ; 0.25	2.4075±0.3293 ; 0.23
asia	8	-4516.0	2000	2000	2	-4941.3±13.50 ; 6.0	-4598.8±67.64 ; 3.9	-4654.9±7.04 ; 5.1	-4654.9±7.04 ; 5.1	**-4555.9**±16.88 ; 8.8	**-4547.0**±16.93 ; 5.9	**-4563.0**±16.03 ; 9.1
						1.4175±0.0388 ; 4.09	3.2293±4.2797 ; 2.95	6.4567±0.6619 ; 0.08	6.4567±0.6619 ; 0.17	**1.3946**±0.3074 ; 0.28	**1.1919**±0.2596 ; 0.28	**1.3181**±0.2476 ; 0.26

TAB. F.3 : *Résultats pour l'exemple asia et pour les méthodes PC, BNPC, MWST-im, MWST-bic, K2(R), K2+T et K2-T.*

bases	tailles					GS-BIC			GS-BD	GES	
	N	score	app	test	moy	0	chaîne	+T	+T	BD	BIC
asia	8	-4516.0	100	2000	2	-5090.7±161.63 ; 6.0	-5029.9±123.09 ; 6.8	-5339.6±705.10 ; 8.0	-4716.4±59.91 ; 7.7	**-4532.5**±8.07 ; 1.4	**-4527.2**±3.09 ; 1.1
						42.3272±24.0113 ; 0.12	49.8078±14.3848 ; 0.20	57.7544±42.3417 ; 0.05	25.1337±2.1991 ; 0.12	**1.2846**±0.2203 ; 4.45	**1.1560**±0.0221 ; 4.41
asia	8	-4516.0	300	2000	2	-4862.5±133.07 ; 5.0	-4851.2±169.16 ; 5.0	-4730.4±322.19 ; 6.8	-4657.6±16.23 ; 6.8	**-4532.5**±8.07 ; 1.4	**-4527.2**±3.09 ; 1.1
						13.8987±5.7907 ; 0.33	22.5079±10.3860 ; 0.48	15.7322±18.9120 ; 0.05	11.6330±2.2580 ; 0.13	**1.2846**±0.2203 ; 4.44	**1.1560**±0.0221 ; 4.41
asia	8	-4516.0	500	2000	2	-4874.8±107.19 ; 5.4	-4683.8±159.20 ; 4.6	-4651.0±7.09 ; 5.4	-4651.0±7.09 ; 5.4	**-4532.5**±8.07 ; 1.4	**-4527.2**±3.09 ; 1.1
						8.6028±5.1138 ; 0.67	10.9873±9.8284 ; 0.74	9.7946±2.2183 ; 0.05	9.7946±2.2183 ; 0.13	**1.2846**±0.2203 ; 4.45	**1.1560**±0.0221 ; 4.42
asia	8	-4516.0	1000	2000	2	-4878.6±107.12 ; 5.8	-4697.7±173.82 ; 4.5	-4657.8±16.00 ; 5.5	-4657.8±16.00 ; 5.5	**-4532.5**±8.07 ; 1.4	**-4527.2**±3.09 ; 1.1
						5.9479±5.6450 ; 1.58	9.5611±10.4968 ; 1.54	7.6991±1.1063 ; 0.06	7.6991±1.1063 ; 0.15	**1.2846**±0.2203 ; 4.44	**1.1560**±0.0221 ; 4.40
asia	8	-4516.0	2000	2000	2	-4941.3±13.50 ; 6.0	-4598.8±67.64 ; 3.9	-4654.9±7.04 ; 5.1	-4654.9±7.04 ; 5.1	**-4532.5**±8.07 ; 1.4	**-4527.2**±3.09 ; 1.1
						1.4175±0.0388 ; 4.09	3.2293±4.2797 ; 2.95	6.4567±0.6619 ; 0.08	6.4567±0.6619 ; 0.17	**1.2846**±0.2203 ; 4.44	**1.1560**±0.0221 ; 4.41

TAB. F.4 : *Résultats pour l'exemple asia et pour les méthodes GS-bic(0), GS-bic(C), GS-bic(T), GS-bd(T), GES-bd, GES-bic.*

F.2 Expérimentations à partir de bases complètes disponibles pour la classification

Dans cette section sont présentés les résultats d'identification de structure à partir de bases de données complètes décrites dans la section E.

Le contenu d'une cellule est alors

Taux de classification [Intervale de confiance à 95%]
Score BIC ; Temps de calcul

Les taux de bonne classification mesuré sur des données de tests sont données avec un intervale de confiance à α% calculé à l'aide de l'équation F.1 et proposé par Bennani & Bossaert (1996).

$$I(\alpha, N) = \frac{T + \frac{Z_\alpha^2}{2N} \pm Z_\alpha \sqrt{\frac{T(1-T)}{N} + \frac{Z_\alpha^2}{4N^2}}}{1 + \frac{Z_\alpha^2}{N}} \tag{F.1}$$

où N est le nombre d'exemples, T, le taux de bonne classification du classifieur et $Z_\alpha = 1,96$ pour $\alpha = 95\%$.

Le temps de calcul est en secondes et est donné à titre indicatif.

Toutes les bases d'exemples ont été entièrement discrétisées. Les nouvelles bases sont disponibles dans le *Structure Learning Package* (Leray & François (2004b)).

bases	tailles						MWST		TANB		K2		
	N	class	app	test	moy	NB	IM	BIC	IM	BIC	Rnd	+T	-T
australian	15	2	400	290	7	84.1 [79.5;87.9]	84.1 [79.5;87.9]	**85.2** [80.6;88.8]	84.1 [79.5;87.9]	83.1 [78.4;87.0]	84.5 [79.9;88.2]	84.8 [80.2;88.5]	84.5 [79.9;88.2]
						-5055.2; 0.0	-7008.2; 0.6	-4975.3; 1.3	-9643.5; 0.6	-5378.7; 1.1	-5132.2; 1.9	-5069.4; 2.0	-5056.5; 2.2
car	7	4	1152	576	4	83.5 [80.3;86.3]	84.0 [80.8;86.8]	83.0 [79.7;85.8]	91.5 [88.9;93.5]	**93.2** [90.8;95.0]	77.1 [73.5;80.3]	83.0 [79.7;85.8]	73.8 [70.0;77.2]
						-4702.0; 0.0	-4702.0; 0.1	-4691.5; 0.3	-4997.0; 0.1	-4956.9; 0.1	-4704.2; 0.4	-4700.5; 0.4	-4715.5; 0.4
contrasep	10	3	860	623	5	43.3 [39.5;47.3]	41.1 [37.3;45.0]	40.0 [36.2;43.9]	**46.9** [43.0;50.8]	43.7 [39.8;4.6]	40.0 [36.2;43.9]	41.1 [37.3;45.0]	34.5 [30.9;38.3]
						-6888.5; 0.0	-6791.2; 0.3	-6662.5; 0.6	-7893.0; 0.2	-7002.4; 0.2	-6674.5; 0.9	-6638.1; 0.9	-6668.5; 0.8
diabetes	9	2	400	368	13	75.0 [70.3;79.2]	77.4 [72.9;81.4]	75.0 [70.3;79.2]	73.9 [69.2;78.1]	72.8 [68.1;77.1]	77.4 [72.9;81.4]	77.4 [72.9;81.4]	77.4 [72.9;81.4]
						-6155.7; 0.0	-9466.2; 0.3	-6155.7; 0.4	-13621.9; 0.3	-10313.0; 0.3	-5986.0; 0.5	-5986.0; 0.4	-5986.0; 0.5
german	25	2	600	400	6	73.2 [68.7;77.4]	69.2 [64.6;73.6]	**73.8** [69.2;77.8]	**73.5** [68.9;77.6]	**74.0** [69.5;78.1]	69.2 [64.6;73.6]	70.2 [65.6;74.5]	69.2 [64.6;73.6]
						-9578.8; 0.0	-15630.4; 1.8	-9107.4; 5.0	-23876.2; 1.3	-9796.6; 4.5	-8963.7; 8.8	-8926.3; 9.1	-8962.9; 8.8
heart	14	2	150	120	6	**86.7** [79.4;91.6]	79.2 [71.1;85.5]	83.3 [75.7;88.9]	77.5 [69.2;84.1]	81.7 [73.8;87.6]	**86.7** [79.4;91.6]	**84.2** [76.6;89.6]	76.7 [68.3;83.3]
						-2063.9; 0.0	-3316.3; 0.4	-2060.8; 1.0	-4981.0; 0.4	-2293.4; 0.9	-1981.5; 1.6	-1991.6; 1.3	-1998.6; 1.5
letter	17	26	15 k	5 k	16	73.5 [72.3;74.7]	74.9 [73.6;76.0]	74.2 [72.9;75.4]	**86.3** [85.3;87.3]	**86.4** [85.4;87.4]	60.0 [58.6;61.3]	74.9 [73.6;76.0]	36.2 [34.9;37.6]
						-184078.4; 0.0	-166417.4; 11.8	-166018.1; 10.6	-513872.5; 16.8	-513871.9; 3.5	-209247.3; 24.6	-166417.4; 17.8	-194376.7; 23.0
monks 1	6	2	124	432	3	71.3 [66.9;75.4]	68.1 [63.5;72.3]	70.8 [66.4;74.9]	95.4 [93.0;97.0]	71.3 [66.8;75.4]	66.7 [62.1;70.9]	66.7 [62.1;70.9]	75.0 [70.7;78.9]
						-2897.7; 0.0	-2912.9; 0.1	-2897.7; 0.2	-2875.7; 0.1	-2897.7; 0.1	-2879.5; 0.3	-2879.5; 0.2	-2873.4; 0.3
monks 2	6	2	169	432	3	46.3 [41.6;51.0]	45.8 [41.2;50.5]	45.1 [40.5;49.9]	49.1 [44.4;53.8]	49.1 [44.4;53.8]	50.0 [45.3;54.7]	50.0 [45.3;54.7]	50.0 [45.3;54.7]
						-2897.7; 0.0	-2931.1; 0.1	-2897.7; 0.2	-3025.1; 0.1	-2958.4; 0.2	-2957.4; 0.2	-2957.4; 0.2	-2957.4; 0.2
monks 3	6	2	122	432	3	55.6 [50.8;60.2]	55.6 [50.8;60.2]	55.6 [50.8;60.2]	55.6 [50.8;50.2]	**55.8** [51.1;60.4]	55.6 [50.8;60.2]	55.6 [50.8;60.2]	55.6 [50.8;60.2]
						-2897.7; 0.0	-2912.9; 0.1	-2897.7; 0.2	-2897.7; 0.1	-2958.4; 0.1	-2879.5; 0.3	-2879.5; 0.2	-2897.7; 0.3
nursery	9	5	8500	4460	4	91.2 [90.4;92.0]	91.2 [90.4;92.0]	91.0 [90.1;91.8]	**93.7** [92.9;94.4]	91.8 [90.9;92.6]	91.1 [90.3;91.9]	92.2 [91.3;92.9]	90.2 [89.2;91.0]
						-43997.8; 0.0	-43997.8; 0.9	-43985.3; 1.5	-4.4878.9; 0.5	-4.4180.2; 0.4	-44120.9; 2.2	-44234.1; 2.4	-46932.8; 2.1
pen	17	10	7494	3498	10	83.0 [81.7;84.2]	82.3 [81.0;83.5]	82.3 [81.0;83.5]	**92.2** [91.2;93.1]	**92.1** [91.1;93.0]	85.0 [83.7;86.1]	90.7 [89.7;91.7]	73.7 [72.2;75.1]
						-98768.1; 0.0	-97233.4; 4.6	-97233.4; 5.8	-125259.8; 4.9	-125259.9; 1.9	-148501.7; 16.4	-117625.3; 13.6	-147498.8; 18.7
segment	20	7	1400	910	21	90.0 [87.9;91.8]	89.2 [87.0;91.1]	89.8 [87.6;91.6]	91.5 [89.6;93.2]	91.9 [89.9;93.5]	**94.0** [92.2;95.3]	**93.3** [91.5;94.7]	82.9 [80.3;85.2]
						-37105.4; 0.0	-56395.7; 2.9	-35277.6; 3.6	-285624.0; 2.7	-150681.3; 3.2	-97261.5; 19.8	-84892.6; 16.1	-184820.7; 50.9
SPECT	23	2	80	187	2	**64.7** [57.6;71.2]	57.8 [50.6;64.6]	57.8 [50.6;64.6]	**63.1** [56.0;69.7]	**63.1** [56.0;69.7]	**65.2** [58.2;71.7]	56.1 [49.0;63.1]	57.8 [50.6;64.6]
						-2654.6; 0.0	-2361.2; 0.3	-2361.2; 1.4	-2415.0; 0.2	-2415.0; 1.2	-2338.8; 3.6	-2375.4; 3.3	-2311.2; 3.3
tae	6	3	100	51	11	37.3 [25.3;51.0]	31.4 [20.3;45.0]	31.4 [20.3;45.0]	17.7 [9.5;30.3]	31.4 [20.3;45.0]	29.4 [18.7;43.0]	15.7 [8.2;28.0]	29.4 [18.7;43.0]
						-753.4; 0.0	-2116.6; 0.1	-680.0; 0.1	-5518.1; 0.1	-1083.8; 0.1	-11482.1; 0.4	-11900.0; 0.4	-11502.0; 0.4
wine	14	3	80	98	7	91.5 [82.8;96.1]	91.5 [82.8;96.1]	91.5 [82.8;96.1]	**93.0** [84.6;97.0]	**93.0** [84.6;97.0]	90.1 [81.0;95.1]	91.5 [82.8;96.1]	63.4 [51.8;73.6]
						-1456.9; 0.0	-2133.3; 0.4	-1456.9; 1.0	-4937.0; 0.4	-3255.4; 0.9	-1573.1; 1.5	-1377.2; 1.5	-1632.1; 1.6
zoo	17	7	60	41	2	**92.7** [80.6;97.5]	**92.7** [80.6;97.5]	87.8 [74.5;94.7]	90.2 [77.5;96.1]	**92.7** [80.6;97.5]	90.2 [77.5;96.1]	90.2 [77.5;96.1]	85.4 [71.6;93.1]
						-460.3; 0.0	-461.1; 0.3	-408.9; 1.6	-921.9; 0.3	-657.0; 1.4	-3253.6; 5.3	-623.8; 4.1	-7584.3; 5.1

TAB. F.5 : *Résultats en classification pour les méthodes Naïve Bayes, MWST-im, MWST-bic, TAN-im, TAN-bic, K2(R), K2+T, K2-T.*

bases	tailles					Indépendances conditionnelles		Gready						GES
	N	class	app	test	moy	PC	BNPC	BIC	+C	+T	BD	+C	+T	BIC
australian	15	2	400	290	7		84.1 [79.4 ;87.9]	84.5 [79.9;88.2]	84.5 [79.9;88.2]	84.5 [79.9;88.2]	83.8 [79.1;87.5]	80.7 [75.8;84.2]	84.5 [79.9;88.2]	84.5 [79.9;88.2]
							-4962.0 ; 201.8	-4876.5; 59.4	-4899.9; 98.9	-4876.5; 69.1	-5069.8; 63.2	-5043.9; 97.8	-5067.3; 104.0	-4875.9; 42.3
car	7	4	1152	576	4	83.5 [80.2;86.4]	**92.4**[89.9;94.3]	81.4 [78.0;84.4]	81.4 [78.0;84.4]	81.4 [78.0;84.4]	83.3 [80.1;86.1]	83.3 [80.1;86.1]	83.0 [79.7;85.8]	81.4 [78.0;84.4]
						-5803; 4.8	-8483; 3.6	-4675.4; 3.9	-4675.4; 5.2	-4675.4; 3.6	-4689.3; 4.7	-4689.3; 5.6	-4700.5; 2.9	-4675.4; 2.0
contrasep	10	3	860	623	5	42.2 [38.3;46.2]	38.8 [35.0;42.8]	41.1 [37.3;45.0]	41.1 [37.3;45.0]	41.1 [37.3;45.0]	41.1 [37.3;45.0]	41.1 [37.3;45.0]	41.1 [37.3;45.0]	41.1 [37.3;45.0]
						-22604; 12.4	-87082; 18.1	-6638.1; 16.9	-6648.0; 22.2	-6638.1; 8.7	-6638.1; 16.7	-6676.0; 19.9	-6638.1; 10.1	-6638.1; 8.3
diabetes	9	2	400	368	13			77.4 [72.9;81.4]	77.4 [72.9;81.4]	77.4 [72.9;81.4]	77.4 [72.9;81.4]	77.4 [72.9;81.4]	77.4 [72.9;81.4]	77.4 [72.9;81.4]
								-5986.0; 4.2	-5986.0; 12.5	-5986.0; 11.2	-5986.0; 4.3	-5986.0; 10.9	-5986.0; 9.3	-5986.0; 2.3
german	25	2	600	400	6			70.5 [65.9;74.8]	70.2 [65.6;74.5]	70.5 [65.9;74.8]	70.2 [65.6;74.5]	70.2 [65.6;74.5]	70.5 [65.9;74.8]	71.5 [66.8;75.8]
								-8886.6; 1149.7	-8888.1; 1598.5	-8886.6; 575.4	-8887.6; 1255.5	-8893.4; 1560.3	-8915.7; 1642.5	-8905.3; 876.8
heart	14	2	150	120	6		69.2 [60.4;76.8]	**84.2** [76.6;89.6]	84.2 [76.6;89.6]	84.2 [76.6;89.6]	84.2 [76.6;84.2]	84.2 [76.6;89.6]	84.2 [76.6;89.6]	84.2 [76.6;89.6]
							-3423; 105.2	-1986.1; 41.3	-1991.6; 70.8	-1986.1; 37.1	-1991.6; 37.4	-1986.1; 73.9	-1986.1; 56.9	-2033.8; 20.0
letter	17	26	15 k	5 k	16			74.2 [72.9;75.4]	24.9 [23.7;26.0]	74.2 [72.9;75.4]	74.9 [73.6;76.0]	69.7 [68.5;71.7]	74.9 [73.6;76.0]	74.8 [73.5;75.9]
								-166226.7; 1696.3	-184109.6; 503.9	-166018.1; 352.8	-181121.1; 2039.6	-206258.4; 1639.3	-166417.4; 274.7	-189554.6; 518.2
monks 1	6	2	124	432	3	75 [70.7;78.9]	**100** [99.1;100]	**100** [99.1;100]	**100** [99.1;100]	**100** [99.1;100]	**100** [99.1;100]	**100** [99.1;100]	**100** [99.1;100]	**100** [99.1;100]
						-2873; 0.1	-2760; 0.4	-2760.4; 2.4	-2760.4; 3.6	-2766.5; 3.2	-2766.5; 2.4	-2766.5; 3.4	-2766.5; 2.8	-2760.4; 2.2
monks 2	6	2	169	432	3	50.0 [45.3;54.7]	50.0 [45.3;54.7]	50.0 [45.3;54.7]	50.0 [45.3;54.7]	50.0 [45.3;54.7]	50.0 [45.3;54.7]	50.0 [45.3;54.7]	50.0 [45.3;54.7]	50.0 [45.3;54.7]
						-2957; 0.1	-2976; 0.3	-2957.4; 1.4	-2957.4; 3.6	-2957.4; 3.6	-2957.4; 1.4	-2957.4; 3.6	-2957.4; 3.6	-2957.4; 0.6
monks 3	6	2	122	432	3	55.6 [50.8;60.2]	55.6 [50.8;60.2]	55.6 [50.8;60.2]	55.6 [50.8;60.2]	55.6 [50.8;60.2]	55.6 [50.8;60.2]	55.6 [50.8;60.2]	55.6 [50.8;60.2]	55.6 [50.8;60.2]
						-2898.7; 0.1	-2898.7; 0.3	-2879.5; 2.1	-2897.7; 4.1	-2879.5; 2.9	-2897.7; 2.1	-2879.5; 4.0	-2879.5; 2.7	-2897.7; 1.5
nursery	9	5	8500	4460	4	90.1 [89.2;90.9]	30.6 [29.3;32.0]	90.8 [89.9;91.6]	90.9 [90.0;91.7]	90.8 [89.9;91.6]	90.5 [89.6;91.3]	90.5 [89.6;91.3]	92.0 [91.2;92.8]	90.9 [89.9;91.7]
						-46933; 125.1	-526330; 56.8	-43847.2; 56.9	-43817.4; 83.8	-43847.2; 32.3	-44757.5; 58.1	-44757.5; 77.7	-44234.1; 36.3	-43817.0; 18.0
pen	17	10	7494	3498	10			83.0 [81.8;84.3]	75.6 [74.1;77.2]	83.0 [81.8;84.3]	90.5 [89.5;91.4]	90.4 [89.4;91.3]	91.0 [90.0;91.9]	86.4 [85.2;87.5]
								-99534.5; 1039.6	-108477.3; 676.6	-99534.5; 303.6	-119650; 1484.2	-118817; 1590.4	-117914; 731.8	-105400; 285.3
segment	20	7	1400	910	21			88.8 [86.6;88.8]	90.4 [88.4;90.4]	88.8 [86.6;90.7]	**93.1** [91.2;93.1]	92.6 [90.8;92.6]	92.5 [90.6;94.1]	92.3 [90.3;93.9]
								-34115.9; 439.4	-35647.1; 791.7	-34115.9; 332.9	-102981.6; 732.4	-95048.4; 922.8	-78531.4; 496.4	-54592; 1659
SPECT	23	2	80	187	2	**64.7** [57.6 ;71.2]	57.8 [50.6;64.6]	57.8 [50.6;64.6]	57.8 [50.6;64.6]	57.8 [50.6;64.6]	54.5 [47.4;61.2]	56.7 [49.5;63.5]	56.1 [49.0;63.1]	57.8 [50.6;64.6]
						-2654; 1500	-2400; 0.3	-2339.2; 299.5	-2267.8; 358.6	-2355.1; 75.6	-2394.7; 370.1	-2359.8; 752.6	-2373.6; 101.0	-2346.8; 871.9
tae	6	3	100	51	11	**51.0** [37.7;64.2]	29.4 [18.7;43.0]	31.4 [20.3;45.0]	31.4 [20.3;45.0]	31.4 [20.3;45.0]	31.4 [20.3;45.0]	31.4 [20.3;45.0]	15.7 [8.2;28.0]	31.4 [20.3;45.0]
						-138170; 0.7	-674.2; 0.3	-577.9; 1.6	-577.9; 2.8	-577.9; 1.8	-12350.0; 2.7	-12350.0; 2.4	-11900.0; 2.6	-577.8; 1.1
wine	14	3	80	98	7		59.2 [49.2;68.4]	87.3 [77.6;93.2]	87.3 [77.6;93.2]	87.3 [77.6;93.2]	91.5 [82.8;95.5]	88.7 [79.3;93.6]	**93.0** [84.6;97.0]	87.3 [77.6;93.2]
							-14668; 0.25	-1377.9; 36.6	-1377.9; 87.4	-1377.9; 45.8	-1507.0; 44.8	-1600.0; 69.6	-1437.3; 56.8	-1377.9; 41.6
zoo	17	7	60	41	2	75.6 [60.6;86.2]	70.7 [55.5;82.4]	87.8 [74.5;94.7]	70.7 [55.5;82.4]	87.8 [74.5;94.7]	**95.1** [83.9;98.7]	**92.6** [80.5;97.5]	**95.1** [83.9;98.7]	87.8 [74.5;94.7]
						-487.1; 120.2	-365.4; 63.9	-407.5; 145.8	-429.2; 154.5	-407.5; 45.9	-1055.0; 269.6	-843.9; 267.0	-1312.7; 339.5	-494.4; 103.5

TAB. F.6 : *Résultats en classification pour les méthodes PC, BNPC, GS-bic(0), GS-bic(C), GS-bic(T), GS-bd(0), GS-bd(C), GS-bd(T) et GES-bic.*

F.3 Expérimentations à partir de bases d'exemples incomplètes disponibles pour la classification

Dans cette section sont présentés les résultats d'identification de structure à partir de bases de données complètes décrites dans la section E.

Le contenu d'une cellule est alors

Taux de classification [Intervale de confiance à 95%]
Score *BIC*
Vraisemblance du modèle ; Temps de calcul

Les taux de bonne classification mesuré sur des données de tests sont données avec un intervale de confiance à α% calculé à l'aide de l'équation F.1 et proposé par Bennani & Bossaert (1996).

Le temps de calcul est en secondes et est donné à titre indicatif.

Toutes les bases d'exemples ont été entièrement discrétisées. Les nouvelles bases sont alors disponibles dans le *Structure Learning Package* (Leray & François (2004b)).

Bases	N	Napp	Ntest	#C	%I	NB-EM	MWST-EM	TAN-EM	SEM	SEM+T
Hepatitis	20	90	65	2	8.4	70.8 [58.8 ;80.5] -1410.9 -1224.2 ; 29.5	73.8 [62.0 ;83.0] -1914.8 **-1147.6** ; 90.4	**75.4** [63.6 ;84.2] -1902.4 **-1148.7** ; 88.5	66.1 [54.0 ;76.5] **-1328.5** -1211.5 ; 1213.1	66.1 [54.0 ;76.5] -1374.4 -1207.9 ; 1478.5
Horse	28	300	300	2	88.0	75 [63.5 ;83.8] -5934.2 -5589.1 ; 227.7	77.9 [66.7 ;86.2] -6337.5 **-5199.6** ; 656.1	**80.9** [69.9 ;88.5] -6469.5 -5354.4 ; 582.2	66.2 [54.3 ;76.3] -6001.4 -5348.3 ; 31807	66.2 [54.3 ;76.3] **-5888.6** -5318.2 ; 10054
House	17	290	145	2	46.7	89.7 [83.6 ;93.7] -2297.0 -2203.4 ; 110.3	**93.8** [88.6 ;96.7] -2696.6 -2518.0 ; 157.0	92.4 [86.9 ;95.8] **-2200.8** **-2022.2** ; 180.7	92.4 [86.9 ;95.8] -2660.5 -2524.4 ; 1732.4	**93.8** [88.6 ;96.7] -2374.4 -2195.8 ; 3327.2
Mushrooms	23	5416	2708	2	30.5	**92.8** [91.7 ;93.8] -98682.1 -97854 ; 2028.9	74.7 [73.0 ;73.4] -109579.8 -108011 ; 6228.2	91.3 [90.2 ;92.4] **-93573.6** **-87556** ; 5987.4	74.9 [73.2 ;76.5] -111932.3 -111484 ; 70494	74.9 [73.2 ;76.5] -111591.1 -110828 ; 59795
Thyroid	22	2800	972	2	29.9	95.3 [93.7 ;96.5] **-40352.0** -39348 ; 1305.6	93.8 [92.1 ;95.2] -45036.5 -38881 ; 3173.0	**96.2** [94.7 ;97.3] -50649.7 **-38350** ; 3471.4	93.8 [92.1 ;95.2] -41613.3 **-38303** ; 17197	93.8 [92.1 ;95.2] -41400.1 -39749 ; 14482

TAB. F.7 : *Première ligne : meilleur taux de classification atteint sur* 10 *exécutions (en %) sur les bases d'exemples de tests et leur intervalle de confiance, pour les méthodes suivantes NB-EM, MWST-EM, SEM, TAN-EM et SEM+T. Seconde ligne : Score* BIC *obtenus. Troisème ligne : log-vraisemblance estimée sur les base d'exemples de tests et les temps de calculs (en secondes) pour le réseau avec le meilleur taux de classification. Les* 6 *premières colonnes indiquent le nom de la base et certaines de ses propriétés : nombre d'attributs, taille de la base d'apprentissage, taille de la base de tests, nombre de classes et pourcentage d'entrées incomplètes.*

Liste des figures

2.1 Quatre types de modèles graphiques . . . 12
2.2 La structure en V, l'avantage des modèles dirigés . . . 12
2.3 Le carré, l'avantage des modèles non dirigés . . . 13
2.4 Un exemple de RBMA . . . 16
2.5 Un exemple de diagramme d'influence . . . 17
2.6 Un exemple de modèle de Markov caché . . . 18
2.7 Un exemple de chaîne de Markov . . . 18
2.8 Un réseau bayésien dynamique associé à une chaîne de Markov . . . 19
2.9 Un exemple de chaîne de Markov cachée . . . 20
2.10 Différents types de modélisations avec variables latentes. . . . 25
2.11 Un exemple de champ de Markov . . . 26
2.12 Les modèles graphiques . . . 27
2.13 Un champ de Markov sous forme de DAG . . . 28
2.14 Un exemple de RB de niveau deux renseigné . . . 28

3.1 Illustration de la complétude causale . . . 31
3.2 Deux chemins d'influence causale . . . 31
3.3 La V-structure . . . 32
3.4 Modèle divergent . . . 32
3.5 Illustration d'un nœud *puits* . . . 35
3.6 Trois cartes d'indépendances minimales . . . 37
3.7 Exemple de table de probabilités conditionnelles . . . 39
3.8 Trois structures équivalentes encodant $A \perp\!\!\!\perp B|C$. . . . 39

8.1 K2 : apport de l'initialisation par MWST . . . 90
8.2 Voisinage supérieur au sens de Markov d'un graphe simple . . . 94

9.1 Les réseaux de tests. . . . 98
9.2 évolution de score BIC en fonction de la taille de la base . . . 100
9.3 évolution de la divergence de Kullback-Leiber . . . 101
9.4 Structure du réseau bayésien INSURANCE . . . 103
9.5 évolution des variances du score BIC et de la divergence de KL . . . 105
9.6 Quelques duels sur les critères de score . . . 106
9.7 Duels sur le type d'initialisation . . . 107
9.8 Duels inter-méthodes . . . 109
9.9 Moyenne du taux de classification . . . 111

10.1 Une itération EM . . . 125

13.1 Réseau bayésien générique pour la génération de bases incomplètes . . . 147

13.2 Une modélisation de mécanismes MCAR par réseaux bayésiens. 149
13.3 Un exemple simple de modélisation de mécanisme MCAR 150
13.4 Un réseau bayésien pour la modélisation de mécanismes MAR. 151
13.5 Un réseau bayésien pour la modélisation de mécanismes MAR *i.i.d.* 152
13.6 Utiliser une variable *cachée* pour modéliser de mécanismes MAR *i.i.d.* . . . 153
13.7 Un réseau bayésien pour la modélisation de mécanismes MAR non *i.i.d.* . . 154

14.1 Les réseaux de tests supplémentaires. 156
14.2 Stabilité en fonction de la taille de la base d'exemples MCAR et MAR 159
14.3 Stabilité en fonction de la taux de données manquantes 161
14.4 Taux de classification, données incomplètes 164

A.1 Quelle orientation choisir ? 199

C.1 Un exemple de graphe non-orienté, nommé $\mathscr{G}$. 205
C.2 Exemple de sous-graphe et de graphe partiel. 206
C.3 Un exemple de graphe orienté, nommé $\mathscr{D}$. 207

Index

χ^2, 68

ACA, 56, 121, 130
adjacence, 205
AIC, 81
Akaike Information Criterion, 81
    corrected, 82
algorithme de Metropolis-Hastings, 46
alternative model selection, 136
AMS-EM, 136
AMS-EM+T, 141
ancêtre, 207
Apprentissage
    de structure
        indépendances conditionnelles, 67
        méthodes à base de score, 87
        scores, 75
    des paramètres, 51
        avec *a priori*, 53
        données incomplètes, 54
        données manquantes, 56, 137
        sans a priori, 51
Approximate bayesian bootstrap, 58
arête, 206, 208
arborescence, 208
arbre, 207
arbre de jonction, 42, 43
arc, 207
    réversible, 39
automates stochastiques, 19
Available Cases Analysis, 56

Bagging, 58
base incomplète, 54
Bayes naïf, 22, 88
    augmenté, 88
        par un arbre, 89
        par une forêt, 89, 140
    augmentée, 22
Bayes Net Power Constructor, 72
Bayesian Multi-Nets, 22
Bayesian Network Augmented Naive Bayes, 22
Bayesian Structural EM, 137
biais d'équiprobabilité, 196
BIC, 82
BMN, 22
BNAN, 22
BNPC, 72
Boosting, 58
BS-EM, 137
Bucket Elimination, 44

carte d'indépendances, 35
    minimale, 36
carte de dépendances
    minimale, 37
carte parfaite, 38
Case-Based Bayesian Networks, 23
causalité, 198, 199
cause/effet, 198
CBBN, 23
CCA, 56, 120, 130
CEM, 127
chaîne, 206
    active, 34
    bloquée, 34
    convergente, 34
    divergente, 34
    en série, 34
Chaîne de Monte-Carlo, 18, 46
Chaînes de Markov, 18
    cachées, 20
Chain Graphs, 27
Champs de Markov, 26
Cheeseman et Stutz, 131
chemin, 208
circuit, 208
classification, 89, 108, 140, 164
classifieur de Bayes Naïf, 88
clique, 206
co-variation, 198

complétude causale, 30
Complete Cases Analysis, 56
completed-PDAG, 40
condition de Markov, 30
Connaissance, 7
constraints based search, 67
corde, 27
corrélation, 202
Couverture de Markov, 207
Cut-Set conditioning, 42
cycle, 206

D-map, 37
d-séparation, 35
DBN, 17
descendant, 207
diagramme d'influence, 17
dimension
    d'un RB, 76
Dirichlet, 50
distribution de Dirichlet, 50
données manquantes, 54

Ecart-type, 202
Echantillonnage, 144
    de Gibbs, 47
    de Gibbs, 57
    de Gibbs., 145
    hypothése, 146
Elimination de variables, 44
EM, 124
    classification, 127
    incrémental, 127
    Monte-Carlo, 127
    paramétrique, 56, 137
    variationnel, 128
EMCMC, 138
enfant, 207
ensemble de coupe, 42
Equivalents de Markov, 39
espérance, 202
    a posteriori, 53
Espace probabilisé, 201
Etude des cas complets, 56
Etude des cas disponibles, 56
evolutionary Markov Chain of Monte Carlo, 138

factoriser, 36
Faithfullness, 31
FAN, 89, 140
FAN-EM, 140
FCI, 71
feuille, 208
Fidélité, 31
fils, 207
Filtre
    bayésien, 19
    de Kalman, 20
forêt, 207
Formule d'inversion de Bayes, 9, 203
Formule de Robinson, 14
Frontière de Markov, 207
Full Information Maximum Likelihood, 58

Génération
    d'exemples, 147
GES, 93
GES-EM, 142
Gibbs sampler, 47, 145
Gibbs sampling, 57
Graphe
    complet, 206
    connexe, 206
    cordal, 27
    essentiel, 40
    Mixtes, 27
    moral, 43
        triangulé, 43
    rôle, 8
    sous-graphe, 206
Graphes, 205
    orienté, 205
    orientés, 207
    partiel, 206
Graphes ancestraux maximum, 25
Graphoïde, 33
    semi-, 33
Greedy Equivalent Search, 93
Greedy Search, 91
GS, 91
GS+T, 92

HMM, 20
Hot deck imputation, 56
Hybrid Independence Test, 138
Hypothése
    Echantillonnage, 146

génération d'exemple, 146
MAR, 146
MCAR, 146
Hypothèse
apprentissage de structure, 65
apprentissage des paramètres, 50, 79
de Dirichlet, 51
indépendance des exemples, 50
indépendance paramétrique, 51
modularité de la vraisemblance, 51

I-map, 35
IC, 71
IC*, 71
ICL, 82
imprécis, 6
Imputation
multiple, 58, 130
par régression, 56
incertain, 6
Indépendance, 202
conditionnelle, 202
inférence
approchée, 45
causale, 24
symbolique, 45
information mutuelle, 70
conditionnelle, 71
Integrated Completed likelyhood, 82
intelligence artificielle, 4
inversion d'arcs, 44

junction tree, 43

K2, 89
K2+T, 90
K2-T, 90
Kullback-Leibler
divergence, 70, 99

Likelihood Ratio Test, 69
log-vraisemblance
d'un RB, 52
Logiciels
B-Course, 210
Bayes Builder, 210
BayesiaLab, 209
BayesWare, 210
BNJ, 209
BNPC, 209
BNT, 209
Causal explorer, 209
gR, 209
Hugin, 210
JavaBayes, 209
LibB, 209
MSBNx, 210
Netica, 210
PNL, 209
Probayes, 209
SLP, 210
Tetrad, 209
Web WeavR, 209
loi jointe, 202
longueur de description minimale, 83
Loopy belief propagation, 48
LRT, 69

méthodes d'apprentissage mixtes, 95, 96
méthodes différentielles, 45
méthodes variationnelles, 47
Machine Learning, 4
MAG, 25
MAR, 55
modélisation, 150
marginalisation, 202
Markov Chains Monte Carlo, 18, 46
Max-Min Hill Climbing, 72
Maximal Anestral Graph, 25
maximum
a posteriori, 52, 54
de vraisemblance, 51, 56
MCAR, 54
modélisation, 148
MCEM, 128
MCMC, 18, 95, 138
MDL, 83
MDP, 20
messages locaux, 42
mesure bayésienne, 78
Metropolis-Hastings, 46
minimum description length, 83
Missing At Random, 55
Missing Completely At Random, 54
Modèles
de Gibbs, 26
markovien, 23
semi-markoviens, 24
semi-orientés, 27

Modèles de Markov, 18
Modèles graphiques, 29
Modes Conditionnels Itérés, 47
Monte Carlo EM, 128
moralisation, 43
Most Probable Explanation, 44
motifs fréquents corrélés, 73
MPE, 44
MRF, 26
Multiple imputation, 121
MWST, 138
MWST-EM, 138

NMAR, 55
    modélisation, 152
Non Missing At Random, 55

Occam
    rasoir d', 77
orientation, 9

P-map, 38
PAG, 40, 74
paradoxe
    du changement de porte, 194
    du prisonnier, 194
paramètres
    définition, 14
    EM, 56, 137
    locaux, 65
parents, 207
PC, 71, 138
PC*, 138
PDAG, 74
    complet, 40
    instantiable, 40
PMMS, 72
Polynomial Max-Min Skeleton, 72
POMDP, 21
Probabilités, 193, 197
    avantages, 5
    classiques, 193
    définition, 201
    interprétation bayésienne, 193
    interprétation fréquentiste, 193
    interprétation logique, 193
    interprétation propensionniste, 193
Probabilités conditionnelle, 201
processus de décision markovien, 20
    partiellement observables, 21
processus de Markov continu, 19
puits, 34

QFCI, 72
Query DAGs concept, 45

Réseaux bayésiens, 13
    à temps continu, 21
    causaux, 23
    définition, 13
    de niveau 2, 28
    discrets finis, 13
    dynamiques, 17
    génération, 145
    multi-agents, 16
    multi-entités, 21
    orientés objet, 16
    pourquoi ?, 7
racine, 208
RAI, 73
rapport de vraisemblance, 69
Raw maximum likelihood, 58
RBE, 57, 138
RBMA, 16
RBME, 21
RBMN, 22
RBOO, 16
recherche
    à base de score, 87
    d'indépendances conditionnelles, 67
recherche de masse, 48
recherche gloutonne, 91, 93
recuit simulé, 47
Recursive Bayesian Multi-Nets, 22
Recusive Autonomy Identification, 73
représentablilité, 38
restrictions successives, 45
Robot, 4
Robust Bayesian Estimator, 57, 138

Séparation, 34
    directionnelle, 35
score, 75
    AIC, 81
    BD, 78
        donneés incomplètes, 133
    BDγ, 80
    BDe, 79

BDeu, 80
BIC, 82, 131
donneés incomplètes, 132
KL, 99
MDL, 83
bayésien, 78
généralisé, 80
données manquantes, 130
IM, 70
KL, 70
méthodes, 87
Vraisemblance, 78
SEM, 136
SEM+T, 141
Simplification du réseau, 48
simulated annealing, 47
SMCM, 24
squelette, 208
Structural Expectation-Maximisation, 136
structure de Bayes Naïf, 88
Structure Learning Package, 210
substitution, 121
Substitution par la moyenne, 56
Symbolic Probabilistic Inference, 45
Système expert, 7

TAN, 89, 140
TAN-EM, 140
test du khi-deux, 68
Théorème de Bayes
généralisé, 202
Three Phase Dependency Analysis, 72
TPDA, 72
Tree augmented Naïve bayes, 22
triangulation, 43

V-structure, 34
Variable aléatoire
définition, 201
latente, 199
variance, 202
VEM, 128
Voting EM, 127
vraisemblance, 79
classifiante, 84
conditionnelle, 84
d'un RB, 51
marginale, 78
maximum de, 52