Étude Comparative d'Algorithmes d'Apprentissage de Structure dans les Réseaux Bayésiens

Olivier François, Philippe Leray

Étude Comparative d'Algorithmes d'Apprentissage de Structure dans les Réseaux Bayésiens

Olivier François, Philippe Leray

Etude comparative d'algorithmes d'apprentissage de structure dans les réseaux bayésiens

Article · January 2004

2 authors:

Olivier C.H. François
Université Gustave Eiffel
33 PUBLICATIONS **345** CITATIONS
SEE PROFILE

Philippe Leray
University of Nantes
215 PUBLICATIONS **1,400** CITATIONS
SEE PROFILE

Étude Comparative d'Algorithmes d'Apprentissage de Structure dans les Réseaux Bayésiens

Olivier François et Philippe Leray

Laboratoire Perception, Systèmes, Information - FRE CNRS 2645
BP08, av. de l'université, 76801 Saint-Etienne-du-Rouvray - Cedex
{olivier.francois,philippe.leray}@insa-rouen.fr

13 juillet 2004

Résumé : Les réseaux bayésiens sont un formalisme de raisonnement probabiliste de plus en plus utilisé pour des tâches aussi diverses que le diagnostic médical, la fouille de texte ou encore la robotique. Dans certains cas, la structure du réseau bayésien est fournie *a priori* par un expert. Par contre, la détermination de cette structure à partir de données est une problématique NP-difficile pour laquelle de nombreuses méthodes d'apprentissage automatique ont été proposées ces dernières années (recherche d'un arbre optimal, recherche gloutonne, prise en compte de données incomplètes, etc). Après un bref rappel des principales méthodes existantes, nous proposons deux séries de tests destinés tout d'abord à évaluer la précision de ces méthodes en essayant de retrouver un graphe connu, et ensuite à tester leur efficacité face à des problèmes de classification. Les résultats obtenus nous permettent d'étudier par exemple la robustesse des méthodes pour la détection de relations "faibles" entre les variables, ou encore leur comportement en fonction du nombre d'exemples.

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

1 Introduction

Les réseaux bayésiens sont un formalisme de raisonnement probabiliste introduit en autre par [Kim & Pearl, 1987, Lauritzen & Speigelhalter, 1988, Jensen, 1996, Jordan, 1998, Naïm *et al.*, 2004].

Définition 1 $\mathcal{B} = (\mathcal{G},\theta)$ *est un* `réseau bayésien` *si* $\mathcal{G} = (X,E)$ *est un graphe acyclique dirigé dont les sommets représentent un ensemble de variables aléatoires* $X = \{X_1, \cdots, X_n\}$*, et si* $\theta_i = [\mathbb{P}(X_i/X_{Pa(X_i)})]$ *est la matrice des probabilités conditionnelles du nœud* i *connaissant l'état de ses parents* $Pa(X_i)$ *dans* $\mathcal{G}$*.*

Une hypothèse imposée par la théorie des réseaux bayésiens est que pour chaque variables X_i, l'ensemble de variables $Pa(X_i)$ doit être tel que X_i est conditionnellement indépendants à X_j $(j \neq i)$ sachant $Pa(X_i)$ (noté $X_i \perp X_j | Pa(X_i)$).

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

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

Cette décomposition de la loi jointe permet 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.

Lorsque la structure du réseau bayésien n'est pas fournie *a priori* par un expert, il est possible d'en faire l'apprentissage à partir d'une base de données. La recherche de structure de réseaux bayésiens n'est pas simple, principalement à cause de la taille super-exponentielle de l'espace de recherche en fonction du nombre de variables.

Nous allons commencer par introduire quelques notions générales sur la structure des réseaux bayésiens, la façon d'associer un score à cette structure et les propriétés intéressantes de ces scores. Ensuite nous détaillerons les méthodes de recherche de structure les plus couramment utilisées, de la recherche de la causalité, au parcours heuristique de l'espace des réseaux bayésiens avec les différents problèmes d'initialisation que cela pose. Nous comparerons alors ces méthodes grâce à deux séries de tests. La première série de tests concerne la capacité des méthodes à retrouver une structure connue. L'autre série de tests permet d'évaluer l'efficacité de ces méthodes à trouver un bon réseau bayésien pour des problèmes de classification en utilisant éventuellement certaines connaissances *a priori* sur la tâche à résoudre. Nous conclurons alors sur les avantages et inconvénients des différentes méthodes utilisées et évoquerons plusieures perspectives.

2 Généralités

La première idée pour trouver la meilleure structure d'un réseau bayésien, est de parcourir tous les graphes possibles, de leur associer un score, puis de choisir le graphe ayant le score le plus élevé. [Robinson, 1977] a montré que $r(n)$, le nombre de structures différentes pour un réseau bayésien possédant n nœuds, est donné par la formule de récurrence de l'équation 2.

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

Ce qui donne $r(1) = 1, r(2) = 3, r(3) = 25, r(5) = 29281, r(10) \simeq 4{,}2 \times 10^{18}$.

Comme l'équation 2 est super-exponentielle, il est impossible d'effectuer un parcours exhaustif en un temps raisonnable dès que le nombre de nœuds dépasse 7 ou 8. La plupart des méthodes d'apprentissage de structure utilisent alors une heuristique de recherche dans l'espace des graphes acycliques dirigés (DAG).

De plus, si le parcours de l'espace des DAG s'effectue avec des opérateurs du type *ajout* ou *suppression* d'arcs, il est nécessaire de réduire le nombre de calculs utilisés pour l'évaluation des scores en se servant d'un score calculable localement afin de n'estimer que la variation de ce score entre deux voisins.

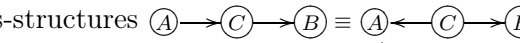
Définition 2 *Un score S est dit* `décomposable` *s'il peut être écrit comme une somme ou un produit de mesures qui sont fonction seulement du nœud et de ses parents. En clair, si n est le nombre de nœuds du graphe, le score doit avoir une des formes suivantes:*

$$S(\mathcal{B}) = \sum_{i=1}^{n} s(X_i, pa(X_i)) \quad ou \quad S(\mathcal{B}) = \prod_{i=1}^{n} s(X_i, pa(X_i))$$

La définition 1 nous dit qu'à un réseau bayésien correspond une décomposition de la loi jointe, mais cette relation n'est pas bijective, d'où la notion d'équivalence définie par:

Définition 3 *Deux* DAG *sont* équivalents au sens de Markov *(noté $\equiv$) s'ils* encodent *le même décomposition de la loi jointe.*

Définition 4 *Une sous-structure du type*  *est appelée une* V-structure *(de nœud puits C).*

Dans cette structure particulière, $A \not\perp B \,|\, C$, c'est à dire que si l'etat du nœud C est donné, la connaissance du nœud A a une influence sur le nœud B contrairement aux 3 autres sous-structures  qui encodent l'indépendance conditionnelle suivante $A \perp B \,|\, C$.

[Verma & Pearl, 1990] ont montré que deux DAG sont équivalents si et seulement si ils ont le même squelette et les mêmes *V-structures.*

Définition 5 *Un arc est dit* `réversible` *s'il n'intervient pas dans une V-structure et s'il ne crée ni cycle ni nouvelle V-structure lorsque que son orientation est changée.*

Définition 6 *Le* graphe acyclique partiellement dirigé *(*PDAG*) obtenu en conservant les arcs non réversibles et en transformant les arcs réversibles en arêtes (non dirigées) est appelé* ***graphe essentiel*** *(ou* Completed-PDAG*). Il représente sans ambiguïté la classe d'équivalence de Markov à laquelle appartient le* DAG *initial.*

Par exemple, les trois structures précédentes peuvent etre représentées pas le `graphe essentiel` suivant : Ⓐ—Ⓑ—Ⓒ

Définition 7 *Un* PDAG *est* ***instantiable*** *s'il est le représentant d'une classe d'équivalence de Markov.*

Il peut alors être intéressant d'associer le même score à toutes les structures équivalentes.

Définition 8 *Un score qui associe une même valeur à deux graphes équivalents est dit* ***score équivalent****.*

Par exemple, le score BIC est à la fois *décomposable*, et *score équivalent*. Il est issu de principes énoncés dans [Schwartz, 1978] et a la forme suivante :

$$BIC(\mathcal{B},D) = \log \mathbb{P}(D|\mathcal{B},\theta^{MV}) - \frac{1}{2} Dim(\mathcal{B}) \log N \tag{3}$$

où D est notre base d'exemples, θ^{MV} est la distribution des paramètres obtenue par *maximum de vraisemblance* pour le réseau $\mathcal{B}$, et où $Dim(\mathcal{B})$ est la dimension du réseau bayésien, définie comme suit.

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 $r_i - 1$ donc pour représenter $\mathbb{P}(X_i/Pa(X_i))$ il faudra $Dim(X_i,\mathcal{B})$ paramètres avec

$$Dim(X_i,\mathcal{B}) = (r_i - 1)q_i \qquad \text{et avec} \qquad q_i = \prod_{X_j \in Pa(X_i)} r_j \tag{4}$$

où q_i est le nombre de configurations possibles pour les parents de X_i. La dimension du réseau $\mathcal{B}$ est alors définie par

$$Dim(\mathcal{B}) = \sum_{i=1}^{n} Dim(X_i,\mathcal{B}) \tag{5}$$

Le score BIC est la somme d'un terme de vraisemblance du réseau par rapport au données, et d'un terme qui pénalise les réseaux complexes. Ce score est *score équivalent* car deux graphes équivalents ont la même vraisemblance (ils représentent la même décomposition de la loi jointe) et la même complexité.

En utilisant des scores ayant cette propriété, il devient possible de faire de la recherche de structure dans l'espace des équivalents de Markov. Cela s'avère être un choix judicieux, car là où un algorithme à base de score dans l'espace des DAG peut boucler sur plusieurs réseaux équivalents, la même méthode utilisée dans l'espace des équivalents sera soit à l'optimum (local), soit pourra trouver un représentant d'une classe d'équivalence qui augmente le score (cf [Munteanu & Bendou, 2002]).

3 Les algorithmes

De nombreuses méthodes sont proposées dans la littérature. Nous avons choisi d'implémenter des méthodes représentatives des grandes familles existantes.

3.1 L'algorithme PC, recherche de causalité

L'algorithme PC [Spirtes *et al.*, 2000] a été introduit par Spirtes, Glymour et Scheines en 1993. Il utilise un test statistique 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, l'arc correspondant est retiré. Un algorithme de principe similaire (IC) a été introduit à la même époque par [Pearl & Verma, 1991].

3.2 L'algorithme BN-PC

Cette autre méthode de recherche de causalité (nommée BN-PC-B [Cheng *et al.*, 2002]) effectue la plupart de son travail dans l'espace des graphes non orientés. Elle est constituée de trois phases. La première phase consiste à rechercher une structure initiale arborescente (avec un principe décrit dans le prochain paragraphe). La deuxième phase recherche des ensembles de conditionnement entre les variables pour décider s'il faut relier les nœuds correspondants. 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 donc à l'aide d'une heuristique orientant ce graphe.

3.3 L'arbre de poids maximal

[Chow & Liu, 1968] 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 [Heckerman *et al.*, 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 aléatoirement ou à l'aide de connaissance *a priori*, ou encore en prenant la variable représentant la classe pour des problèmes de classification.

3.4 L'algorithme K2

L'idée de l'algorithme K2 est de maximiser la probabilité de la structure sachant les données. Pour cela, la probabilité d'une structure conditionnellement à des données peut être calculée en utilisant la remarque suivante :

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

Pour calculer $\mathbb{P}(\mathcal{G},D)$, [Cooper & Hersovits, 1992] ont donné le résultat suivant :

Théorème 1 *Soient D la base de données et N le nombre de cas dans D, et soit $\mathcal{G}$ la structure du réseau sur X. De plus, soient pa_{ij} la j^{ieme} instantiation de $Pa(X_i)$, et N_{ijk} le nombre de cas dans D où X_i a la valeur x_{ik} et que $Pa(X_i)$ est instantié en pa_{ij}. Si $Nij = \sum_{k=1}^{r_i} N_{ijk}$ alors*

$$\mathbb{P}(\mathcal{G},D) = \mathbb{P}(\mathcal{G})\mathbb{P}(D|\mathcal{G}) \text{ avec } \mathbb{P}(D|\mathcal{G}) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\prod_{k=1}^{r_i}N_{ijk}! \tag{6}$$

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

L'équation 6 peut être vue comme une mesure de qualité du réseau par rapport aux données et est appelée la *mesure bayésienne*.

En supposant un *a priori* uniforme sur les structures, la qualité d'un jeu de parents pour un nœud fixé pourra donc être mesurée par le score local de l'équation 7.

$$s(X_i,Pa(X_i)) = \prod_{j=1}^{q_i}\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\prod_{k=1}^{r_i}N_{ijk}! \tag{7}$$

Ensuite en imposant un ordre sur les nœuds, de manière à ce qu'un nœud ne puisse être parent d'un autre que s'il précède celui-ci dans cet ordre, il est alors possible de réduire l'espace de recherche, qui devient alors l'ensemble des réseaux bayésiens respectant cet ordre d'énumération. L'algorithme K2 teste l'ajout de parents en respectant cet ordre. Le premier nœud ne peut pas posséder de parents et pour les nœuds suivants, l'ensemble de parents choisi est celui qui augmente le plus la *mesure bayésienne*.

[Heckerman *et al.*, 1994] a montré que le score bayésien n'était pas *score équivalent* et en a 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*) au lieu du score bayésien. [Bouckaert, 1993] a également proposé une variante à l'algorithme K2 qui utilise le score MDL, lui aussi *score équivalent*.

Citons [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.

3.5 La recherche gloutonne

L'algorithme de recherche gloutonne (ou GS comme *greedy search*) est très répandu en optimisation. Il 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.

Dans notre cas, le voisinage d'un DAG est défini par l'ensemble de tous les DAG obtenus soit en ajoutant, soit en supprimant, soit en retournant un arc du graphe d'origine. L'algorithme s'arrête lorsque le graphe obtenu réalise un maximum (local) de la fonction de score. L'espace de recherche est alors l'espace *complet* des DAG contrairement aux méthodes précédentes.

Puisque cet algorithme va calculer le score des graphes voisins, il faut utiliser un score décomposable pour n'avoir à calculer que la variation de scores locaux entraînée par la suppression ou l'ajout d'un arc et non les scores globaux de ces graphes.

3.6 La recherche gloutonne dans l'espace des équivalents de Markov

Comme nous l'avons vu dans la section 2, il peut être avantageux de définir le voisinage d'un graphe en tenant compte des équivalents au sens de Markov Des travaux très récents [Chickering, 2002a, Castelo & Kocka, 2002, Auvray & Wehenkel, 2002] montrent qu'il peut être plus profitable de travailler dans l'espace des CPDAG (représentants des classes d'équivalence de Markov) plutôt que dans l'espace des DAG. [Chickering, 2002b] a proposé l'algorithme Greedy Equivalent Search (GES) et montré l'optimalité de cette méthode qui effectue une recherche gloutonne dans l'espace des CPDAG en deux phases (une phase d'ajout d'arcs, puis une phase de suppression).

3.7 L'algorithme Structural-EM

Cette méthode a été introduite par [Friedman, 1997]. Elle est basée sur le principe de l'algorithme EM et permet de traiter des bases d'exemples incomplètes sans avoir à ajouter une nouvelle modalité (*variable non mesurée*) à chaque nœud.

Cette méthode itérative part d'une structure initiale pour estimer la distribution de probabilité des variables *cachées* ou *manquantes* grâce à l'algorithme EM classique. L'espérance d'un score par rapport à ces variables *cachées* est ensuite calculée pour tous les réseaux bayésiens du voisinage afin de choisir la structure suivante. La convergence générale de cet algorithme a été prouvée dans [Friedman, 1998].

3.8 Problèmes d'initialisation

La plupart des méthodes proposées précédemment ont des problèmes d'initialisation. Par exemple, K2 dépend fortement de l'ordre d'énumération qui lui est passé en paramètre.

De part la nature de l'algorithme, deux ordres d'énumération des variables pourront mener à des résultats radicalement différents. Il est bien sûr possible d'éliminer ce problème d'initialisation en demandant cet ordre d'énumération à un expert, mais nous préférons nous placer dans le cas contraire, et essayer d'éliminer ce problème en proposant quelques heuristiques d'initialisation.

Suivant une recommendation de [Heckerman *et al.*, 1994] nous proposons d'exploiter l' arbre retourné par l'algorithme MWST pour générer cet ordre. Cela nous donne donc une heuristique d'initialisation à moindre coût puisque MWST a 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. Il est possible 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.

Les recherches gloutonnes sont également sensibles à l'initialisation. Sur le même principe, nous proposons de recourir à l'arbre obtenu par MWST comme point de départ de l'algorithme GS, ce qui donne l'algorithme que nous appellerons "GS+T".

4 Expérimentation

4.1 Recherche d'une bonne structure

4.1.1 Méthodes

Nous avons utilisé Matlab, et plus précisément la Bayes Net Toolbox [Murphy, 2001] qui fournit déjà certaines méthodes présentées ci-dessous. [Leray *et al.*, 2003] en propose une introduction, des tutoriels en français ainsi que le code des fonctions mises en œuvre dans ces expérimentations.

Nous avons mis en œuvre les algorithmes suivants: PC, BN-PC, MWST, K2 (initialisé avec deux ordres différents tirés aléatoirement), K2+T, K2-T (en prenant le nœud correspondant à la classe comme racine de l'arbre), GS (initialisé avec la structure vide et utilisant le score BIC), GS+T, GES et SEM (avec des données manquantes et initialisé avec la structure vide).

4.1.2 Réseaux tests et techniques d'évaluation

Nous allons utiliser deux réseaux bayésiens dont la structure est déjà connue. Le premier réseau provient d'un exemple de diagnostic de la dyspnée *Asia*, qui a été introduit par [Lauritzen & Speigelhalter, 1988] (voir figure 1.a). Les huit nœuds sont binaires. On peut noter que l'arc entre A et T nous dit que le fait d'avoir séjourné en Asie modifie le risque

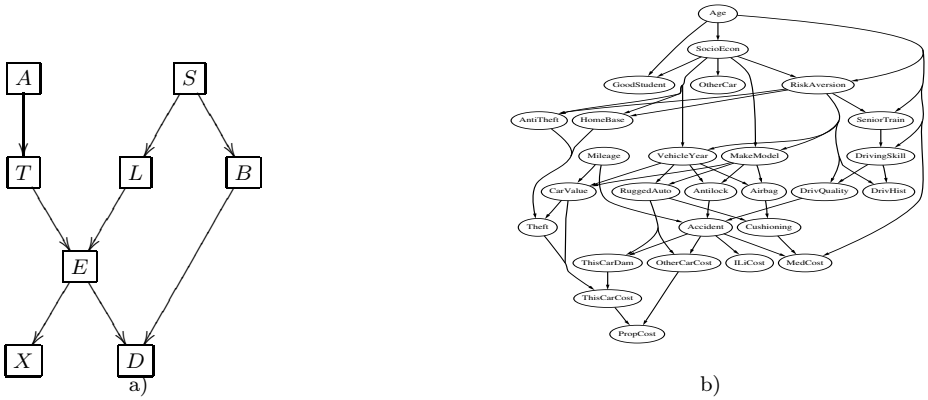


FIG. 1 – *Les réseaux originaux :* ASIA *(à gauche) et* INSURANCE *(à droite)*

d'avoir contracté une tuberculose. La probabilité *a priori* de A est très faible, tout comme l'influence de A sur T.

Le second réseau bayésien est le réseau *Insurance* avec 27 nœuds (voir figure 1.b) et est téléchargeable sur [Friedman *et al.*, 1997].

À partir de ces réseaux bayésiens, nous avons généré plusieurs bases de données de taille variable. Elles serviront à tester l'influence du nombre d'exemples sur les différentes méthodes. Ces mêmes bases ont aussi été vidées *aléatoirement* de leurs valeurs (chaque valeur ayant une probabilité de 0.2 d'être manquante) pour tester plus explicitement l'algorithme SEM.

Pour comparer les résultats issus de ces différents algorithmes, nous allons utiliser la distance d'édition. Cette distance est définie par le nombre d'opérations nécessaires pour transformer le graphe obtenu en celui d'origine (l'ajout, le renversement ou le retrait d'un arc augmente de 1 la distance d'édition). Remarquons ici que le renversement d'un arc est considéré différemment du retrait de l'arc puis de l'ajout de l'arc opposé.

Le score BIC des différents réseaux est également précisé à titre comparatif. Il est calculé à partir d'une base d'exemples supplémentaire de 30000 cas pour ASIA et de 20000 cas pour INSURANCE.

4.1.3 Résultats et interprétations

Influence de la taille de la base

Tout d'abord, l'algorithme MWST parait être peu sensible à la variation de la taille de la base de données. Il donne rapidement un graphe proche du graphe d'origine bien qu'il ne parcourt que l'espace (plus pauvre) des arbres.

L'heuristique PC donne également de bons résultats. Cette méthode construit des structures avec peu d'arcs, mais qui sont presque tous pertinents. Il faut remarquer que le score

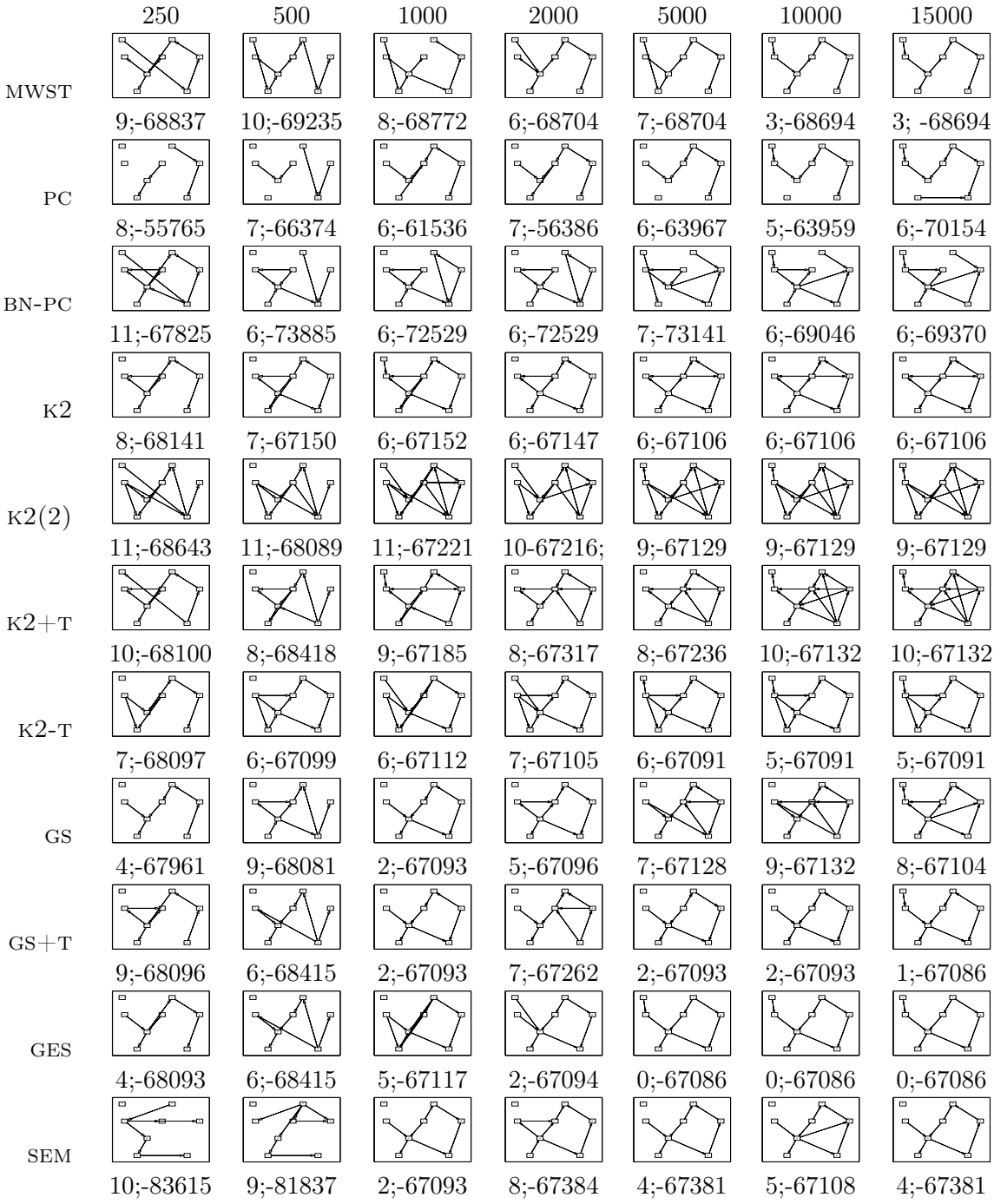


FIG. 2 – ASIA*: réseau, distance d'édition et score BIC obtenus pour différentes méthodes (lignes) et différentes tailles de base d'exemples (colonnes).*

	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
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
SEM	50;-4431	57;-4262	61;-4396	61;-4092	69;-4173	63;-4105	63;-3978

TAB. 1 – INSURANCE *: distance d'édition au graphe d'origine et score BIC (divisés par* 100 *et arrondis) en fonction de la taille de la base de données.*

BIC des structures obtenues est très haut car le terme de pénalité dû au nombre d'arcs est peu élevé.

La méthode BN-PC permet d'obtenir des graphes plus denses que la méthode PC, avec de meilleurs scores BIC, mais en ajoutant souvent des arcs supplémentaires non pertinents.

La méthode K2 est très rapide et est souvent utilisée dans la littérature. Elle reste cependant trop sensible à l'initialisation. La figure 2 donne le résultat de K2 avec deux ordonnancements différents ("ELBXASDT" et "TALDSXEB"). Pour un ordre fixé, K2 trouve des graphes très similaires quelle que soit la taille de la base. Par contre, en changeant d'ordonnancement, le graphe final change radicalement. Cela se voit également sur la table 1 où un premier ordonnancement de K2 a donné des résultats moyens alors qu'une nouvelle initialisation de l'ordre des nœuds (K2(2)) a donné d'excellents résultats.

Initialiser K2 avec un ordre issu de l'arbre rendu par MWST ne permet pas vraiment d'améliorer les résultats. Par contre cette technique à l'avantage de stabiliser l'algorithme K2 qui donne des résultats trop dépendants de son ordre d'énumération. On peut voir que sur l'exemple ASIA c'est la méthode K2-T qui a le mieux fonctionnée tandis que pour INSURANCE, c'est K2+T qui a été la plus efficace des deux.

L'algorithme GS est robuste face à la variation de la taille de la base d'exemples, surtout s'il est initialisé avec l'arbre obtenu par MWST. GS+T est d'ailleurs la méthode qui donne les résultats les plus proches du réseau original ASIA. On remarque également que pour la base d'exemple INSURANCE, les résultats issus de GS+T sont équivalents à ceux trouvés par GS du point de vue du score, mais sont meilleurs pour le nombre d'arcs en commun avec la structure originale.

La méthode GES donne de bons résultats quelle que soit la taille de la base d'exemples. Pour un grand nombre de données, les résultats de GES sont meilleurs que ceux obtenus par une recherche gloutonne classique du point de vue du score. Par contre, ils sont nettement moins bons pour la distance d'édition et même lorsque l'on les compare à ceux obtenus

à l'aide de la recherche d'arbre optimal. Néanmoins, il n'est pas primordial de retrouver exactement la même structure étant donné que des réseaux bayésiens ayant des structures équivalentes encodent la même décomposition de la loi jointe.

L'algorithme SEM trouve des réseaux très proches quelle que soit la taille de la base d'apprentissage. Cette technique permet d'obtenir de très bons résultats sur un exemple simple. Mais lorsque le nombre de nœuds augmente, la perte d'information due aux données manquantes a beaucoup plus d'influence. Il devient alors difficile d'obtenir des réseaux ayant un bon score.

Reconnaissance des dépendances faibles

Pour la plupart des méthodes, l'arc entre A et T du réseau bayésien ASIA n'a pas été trouvé. Or les algorithmes MWST, PC, K2-T et GES y arrivent lorsque la base de données est suffisamment grande. Pour les méthodes à base de score simples 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 d'un lien entre A et T 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.

4.2 Recherche d'un bon classifieur

4.2.1 Bases utilisées et techniques d'évaluation

Asia

Pour cet exemple jouet de diagnostic médical, nous avons utilisé la base de 2000 exemples générée pour l'expérience précédente pour faire l'apprentissage, et la base de 1000 exemples comme base de test.

Heart (disponible sur [Michie *et al.*, 1994])

C'est une base de diagnostic médical qui possède 13 attributs (choisis parmi 75) et la classe et dont tous les attributs continus ont été discrétisés.

La base a été séparée en 189 exemples d'apprentissage et 81 de test.

Australian (disponible sur [Michie *et al.*, 1994])

Cette base consiste en une évaluation de la possibilité de crédit accordée à un client australien en fonction de 14 attributs. Elle contient 690 cas qui ont été séparés en 500 pour l'apprentissage et 190 pour le test.

Letter (disponible sur [Michie *et al.*, 1994])

Il s'agit d'une base d'exemples qui a été crée à partir d'images de lettres alpahabétiques 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. La variable représentant la classe peut donc prendre 26 valeurs différentes et la base contient 15000 exemples d'apprentissage et 5000 de test.

Thyroid (disponible sur [Blake & Merz, 1998])

C'est une base de diagnostic médical pour laquelle nous avons utilisé 22 variables (parmi 29) : 15 variables discrètes, 6 continues qui ont été discrétisés, et la classe. Il y a 2800 exemples d'apprentissage et 972 de test. Cette base contient des données manquantes qui ont été modélisées par un état supplémentaire.

Chess (disponible sur [Blake & Merz, 1998])

La version de cette base est "Chess – King+Rook versus King+Pawn". Il s'agit de prédire si les blancs peuvent gagner la partie d'échec à partir de la description de la position courante. Il y a 3196 instances séparées en 2200 d'apprentissage et 996 de test pour 37 attributs, dont la classe.

Évaluation

Le critère de comparaison entre les méthodes est ici le taux de bonne classification mesuré sur des données de test avec un intervalle de confiance à $\alpha\%$ donné dans l'équation 8 et proposé par [Bennani & Bossaert, 1996].

$$I(\alpha,N) = \frac{T + \frac{Z_\alpha^2}{2N} \pm Z_\alpha \sqrt{\frac{T(1-T)}{N} + \frac{Z_\alpha^2}{4N^2}}}{1 + \frac{Z_\alpha^2}{N}} \tag{8}$$

où N est le nombre d'exemples, T, le taux de bonne classification du classifieur et $Z_\alpha = 1{,}96$ pour $\alpha = 95\%$.

4.2.2 Résultats et interprétations

Les tables 2 et 3 donnent les performances des réseaux bayésiens obtenus par différentes méthodes de recherche de structure ainsi que par deux autres structures typiques en classification. Dans la première structure, appelée *réseau bayésien naïf*, les arcs partent de la variable représentant la classe pour aller aux observations. Cela correspond à une hypothèse d'indépendance des observations conditionnellement à la classe, d'où la simplification de la loi jointe suivante : $\mathbb{P}(C,X_1,...,X_n) = \mathbb{P}(C)\mathbb{P}(X_1|C)...\mathbb{P}(X_n|C)$. La seconde structure correspond à un réseau bayésien naïf pour lequel on aurait rajouté des liens entre les observations pour assouplir l'hypothèse d'indépendance conditionnelle précédente. En pratique, la méthode de l'*arbre augmenté* (*tree augmented naive*, TAN) consiste à rajouter à la structure de Bayes naïve l'arbre optimal reliant les observations obtenu par MWST.

	ASIA2000	HEART	AUTRALIAN
Nvar	8	14	15
Napp, Ntest	2000, 1000	189, 81	500, 190
Bayes Naïf	86,5% [84,2 ; 88,5]	87,7% [78,7 ; 93,2]	87,9% [82,4 ; 91,8]
MWST-BIC	86,5% [84,2 ; 88,5]	81,5% [75,3 ; 86,4]	87,4% [81,8 ; 91,4]
MWST-MI	86,5% [84,2 ; 88,5]	82,7% [73,0 ; 89,5]	85,8% [80,1 ; 90,1]
TAN	84,9% [82,6 ; 87.0]	85,2% [79,4 ; 89,6]	86,3% [80,7 ; 90,5]
PC	84,6% [82,2 ; 86,8]	75,3% [68,6 ; 81,0]	86,3% [80,7 ; 90,5]
BN-PC	86,5% [84,2 ; 88,5]	80,3% [70,3 ; 87,5]	85,8% [80,1 ; 90,1]
K2	86,5% [84,2 ; 88,5]	84,0% [74,4 ; 90,4]	83,7% [77,8 ; 88,3]
K2+T	86,5% [84,2 ; 88,5]	81,5% [71,6 ; 88,5]	84,2% [78,3 ; 88,8]
K2-T	86,5% [84,2 ; 88,5]	76,5% [66,2 ; 84,5]	85,8% [80,1 ; 90,1]
GS	86,5% [84,2 ; 88,5]	85,2% [75,8 ; 91,4]	86,8% [81,3 ; 91,0]
GS+T	86,2% [83,9 ; 88,3]	82,7% [73,0 ; 89,5]	86,3% [80,7 ; 90,5]
GES	86,5% [84,2 ; 88,5]	85,2% [75,8 ; 91,4]	84,2% [78,3 ; 88,8]
SEM	86,5% [84,2 ; 88,5]	75,3% [68,6 ; 81,0]	74,2% [67,5 ; 80,0]
9PPV	86,5% [84,2 ; 88,5]	85,2% [75,8 ; 91,4]	80,5% [74,3 ; 85,6]

TAB. 2 – *Taux de classification et intervalle de confiance à* 95% *obtenus à partir des différentes méthodes sur des problèmes simples.*

Une autre méthode classique en reconnaissance des formes a également été testée, il s'agit des k plus proches voisins (kPPV) (les résultats donnés pour cette méthode ont été calculés avec $k = 9$ néanmoins toutes les valeurs de k donnent des résultats similaires).

Pour des problèmes de classification simples comme ASIA, un réseau bayésien naïf donne d'aussi bons résultats qu'une structure plus évoluée (cf figure 2 et table 2), ou qu'un autre algorithme comme les kPPV. Par contre, sur des problèmes plus complexes, l'arbre obtenu par MWST fait mieux que le réseau bayésien naïf (voir table 3). Il parait donc judicieux de ne pas se priver de cette méthode lorsque le nombre de nœuds dépasse 15 ou 20. Cette méthode donne, à un moindre coût, des résultats au moins aussi bons que ceux issus de la structure naïve, qui est très régulièrement utilisée lorsque la structure du réseau n'est pas connue *a priori*.

Il est difficile de se faire une idée sur l'algorithme TAN à la suite des résultats présentés ici car il donne de bons résultats sur certains exemples (LETTER) et de mauvais résultats sur d'autres (CHESS). Notre avis est que cette méthode perd de son efficacité lorsque le nombre de nœuds augmente. En effet, les réseaux bayésiens obtenus possèdent un nombre élevé d'arcs (notamment à cause de la structure naive) et ainsi un nombre important de paramètres à estimer.

	LETTER	THYROID	CHESS
Nvar	17	22	37
Napp, Ntest	15000, 5000	2800, 972	2200, 996
Bayes Naïf	73,5% [72,2 ; 74,7]	95,7% [94,2 ; 96,9]	86,6% [84,3 ; 88,6]
MWST-BIC	74,1% [72,9 ; 75,4]	96,8% [95,4 ; 97,8]	89,5% [87,3 ; 91,3]
MWST-MI	74,9% [73,6 ; 76,1]	96,1% [94,6 ; 97,8]	89,5% [87,3 ; 91,3]
TAN	85,3% [84,3 ; 86,3]	96,4% [95,0 ; 97,4]	86,4% [84,1 ; 88,4]
PC	memory crash	memory crash	memory crash
BN-PC	memory crash	memory crash	memory crash
K2	74,9% [73,6 ; 76,1]	96,3% [94,9 ; 97,4]	92,8% [90,9 ; 94,3]
K2+T	74,9% [73,6 ; 76,1]	96,3% [94,9 ; 97,4]	92,6% [90,7 ; 94,1]
K2-T	36,2% [34,9 ; 37,6]	96,1% [94,6 ; 97,2]	93,0% [91,2 ; 94,5]
GS	74,9% [73,6 ; 76,1]	96,2% [94,7 ; 97,3]	94,6% [93,0 ; 95,9]
GS+T	74,9% [73,6 ; 76,1]	95,9% [94,4 ; 97,0]	92,8% [90,9 ; 94,3]
GES	74,9% [73,6 ; 76,1]	95,9% [94,4 ; 97,0]	93,0% [91,2 ; 94,5]
SEM	memory crash	96,2% [94,7 ; 97,3]	89,2% [87,1 ; 91,0]
9PPV	94,8% [94,2 ; 95,5]	98,8% [97,8 ; 99,4]	94,0% [92,3 ; 95,4]

TAB. 3 – *Taux de classification et intervalle de confiance à* 95% *obtenus à partir des différentes méthodes sur des problèmes plus complexes.*

La méthode PC[1], qui donne de bons résultat d'un point de vue de la structure, ne permet pas d'obtenir de bons résultats en classification. Cela est principalement dû au fait que les graphes obtenus sont peu denses et même souvent non connexes. L'état du nœud classe n'est alors pas toujours calculé à partir de tous les attributs, mais seulement à partir des nœuds de sa composante connexe. Cette perte d'information influe alors sur le taux de classification.

La méthode BN-PC[2] permet d'obtenir des résultats en classification plus satisfaisant que l'algorithme PC sur des problèmes de faible dimension.

La méthode K2 permet d'obtenir (très rapidement) de bon résultats. Mais elle n'obtient jamais le meilleur résultat (sur nos exemples). Néanmoins, cette méthode prend nettement l'avantage sur les méthodes simples (BN, MWST, TAN) sur des problèmes complexes (cf. table 3). Dans le cadre de la recherche d'un bon classifieur, l'initialisation de l'algorithme K2 par l'arbre obtenu par MWST ne permet pas d'améliorer les résultats en classification.

1. Il faut noter que le *memory crash* correspondant à l'exécution de l'algorithme PC sur des problèmes de taille moyenne est du à l'implémentation actuelle de la méthode. [Spirtes *et al.*, 2000] propose des variantes de l'algorithme PC pouvant traiter des problèmes de plus grande taille que notre implémentation actuelle.

2. Comme pour l'algorithme PC, notre implémentation de BNPC est très gourmande en espace mémoire et ne peut traiter de problèmes ayant trop d'attributs.

Par contre, cela aide à stabiliser la méthode en se démarchant du choix de l'ordonnancement initial des nœuds.

Lorsque l'algorithme de recherche gloutonne GS réussit à battre la structure naïve, celui-ci s'approche du podium des meilleures méthodes. Ces bons résultats sont essentiellement obtenus pour des problèmes complexes où, malheureusement, le coût de calcul de GS est assez prohibitif. De plus il est étonnant de voir que l'algorithme GS initialisé avec l'arbre optimal donne de moins bon résultats que lorsque celui-ci est initialisé avec une structure vide. Cela peut s'expliquer par la taille de l'espace de recherche (cf équation 2) et par le nombre important d'optima locaux dans cet espace.

La méthode GES réussit difficilement à se faire une place dans le trio des méthodes les plus efficaces. Il s'agit pourtant de la méthode à base de score la plus évoluée. On a vu qu'elle permettait de trouver une structure ayant un très bon score. Seulement sur nos problèmes de classification les performances restent moins bonnes que celles obtenues par une recherche gloutonne classique.

L'algorithme SEM permet de traiter efficacement le problème des données manquantes en fournissant un réseau ayant de bonnes performances. Celles-ci sont souvent excellentes lorsque l'on considère que les bases d'exemples ont été vidées d'environ 20% de leurs valeurs pour l'apprentissage.

De manière générale, sur des problèmes complexes, l'apprentissage de structure permet d'obtenir des performances supérieures à celles du réseau naïf, performances qui sont par ailleurs équivalentes voir meilleures que celles des kPPV. De plus, il faut noter que le réseau bayésien obtenu permet aussi d'effectuer d'autres opérations (inférence sur d'autres nœuds, interprétation de la structure obtenue, prise en compte de données manquantes).

5 Conclusions et perspectives

Apprendre la structure d'un réseau bayésien à partir de données est un problème difficile pour lequel nous avons passé en revue les principales méthodes existantes.

Une première série de tests nous a permis d'évaluer la précision de ces méthodes en essayant de retrouver un graphe connu. Les résultats obtenus montrent qu'il est difficile de retrouver des relations "faibles" entre les variables avec peu d'exemples. L'initialisation aléatoire de la plupart des méthodes peut aussi être remplacée efficacement par une initialisation issue d'une méthode simple et rapide comme l'algorithme MWST.

La seconde série de tests nous a permis de tester l'efficacité des mêmes méthodes face à des problèmes de classification. Dans cet article, nous avons d'abord montré qu'une "bonne" recherche de structure permettait d'obtenir des résultats équivalents à un algorithme tel que les kPPV. Chose surprenante, une méthode très simple comme MWST permet souvent d'obtenir des résultats équivalents à ceux de méthodes plus complexes comme GS.

Concernant les méthodes de parcours de l'espace des réseaux bayésiens, il pourrait être judicieux de remplacer la recherche gloutonne par des techniques de parcours d'espace plus

évoluées (recuit simulé, algorithmes génétiques [Larrañaga *et al.*, 1996], etc).

L'adaptation des méthodes existantes à la prise en compte de données manquantes est importante pour pouvoir traiter des problèmes réels. L'algorithme SEM permet déjà cela pour une recherche gloutonne dans l'espace des réseaux bayésiens, mais le même principe pourrait être utilisé pour MWST, permettant d'obtenir très rapidement une structure de réseau ayant de bonnes propriétés. L'étape suivante serait la prise en compte des données manquantes pour les méthodes parcourant l'espace des équivalents de Markov.

Références

[Auvray & Wehenkel, 2002] Auvray V. & Wehenkel L. (2002). On the construction of the inclusion boundary neighbourhood for markov equivalence classes of bayesian network structures. In A. Darwiche & N. Friedman, Eds., *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-02)*, p. 26–35, S.F., Cal.: Morgan Kaufmann Publishers.

[Bennani & Bossaert, 1996] Bennani Y. & Bossaert F. (1996). Predictive neural networks for traffic disturbance detection in the telephone network. In *Proceedings of IMACS-CESA'96*, Lille, France.

[Blake & Merz, 1998] Blake C. & Merz C. (1998). UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[Bouckaert, 1993] Bouckaert R. R. (1993). Probabilistic network construction using the minimum description length principle. *Lecture Notes in Computer Science*, **747**, 41–48.

[Castelo & Kocka, 2002] Castelo R. & Kocka T. (2002). *Towards an inclusion driven learning of bayesian networks*. Rapport interne UU-CS-2002-05, Institute of information and computing sciences, University of Utrecht.

[Cheng *et al.*, 2002] 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.

[Chickering, 2002a] Chickering D. M. (2002a). Learning equivalence classes of bayesian-network structures. *Journal of machine learning research*, **2**, 445–498.

[Chickering, 2002b] Chickering D. M. (2002b). Optimal structure identification with greedy search. *Journal of Machine Learning Research*, **3**, 507–554.

[Chow & Liu, 1968] Chow C. & Liu C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, **14**(3), 462–467.

[Cooper & Hersovits, 1992] Cooper G. & Hersovits E. (1992). A bayesian method for the induction of probabilistic networks from data. *Maching Learning*, **9**, 309–347.

[François & Leray, 2003] François O. & Leray P. (2003). Etude comparative d'algorithmes d'apprentissage de structure dans les réseaux bayésiens. In F. D. de Saint-

Cyr, Ed., *RJCIA2003 - 6emes Rencontres Nationales des Jeunes Chercheurs en Intelligence Artificielle*, p. 167–180: Presses Universitaires de Grenoble.

[Friedman, 1997] Friedman N. (1997). Learning belief networks in the presence of missing values and hidden variables. In *Proc. 14th International Conference on Machine Learning*, p. 125–133: Morgan Kaufmann.

[Friedman, 1998] Friedman N. (1998). The Bayesian structural EM algorithm. In G. F. Cooper & S. Moral, Eds., *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, p. 129–138, San Francisco: Morgan Kaufmann.

[Friedman *et al.*, 1997] Friedman N., Goldszmidt M., Heckerman D. & Russell S. (1997). Challenge: What is the impact of bayesian networks on learning?, *Proceedings of the 15'th International Joint Conference on Artificial Intelligence (NIL-97)*, 10-15. `http://www.cs.huji.ac.il/labs/compbio/Repository/`.

[Friedman & Koller, 2000] Friedman N. & Koller D. (2000). Being bayesian about network structure. In C. Boutilier & M. Goldszmidt, Eds., *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-00)*, p. 201–210, SF, CA: Morgan Kaufmann Publishers.

[Heckerman *et al.*, 1994] Heckerman D., Geiger D. & Chickering M. (1994). Learning Bayesian networks: The combination of knowledge and statistical data. In R. L. de Mantaras & D. Poole, Eds., *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, p. 293–301, San Francisco, CA, USA: Morgan Kaufmann Publishers.

[Jensen, 1996] Jensen F. V. (1996). *An introduction to Bayesian Networks*. Taylor and Francis, London, United Kingdom.

[Jordan, 1998] Jordan M. I. (1998). *Learning in Graphical Models*. The Netherlands: Kluwer Academic Publishers.

[Kim & Pearl, 1987] Kim J. & Pearl J. (1987). Convice; a conversational inference consolidation engine. *IEEE Trans. on Systems, Man and Cybernetics*, **17**, 120–132.

[Larrañaga *et al.*, 1996] 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.

[Lauritzen & Speigelhalter, 1988] Lauritzen S. & Speigelhalter D. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Royal statistical Society B*, **50**, 157–224.

[Leray *et al.*, 2003] Leray P., Guilmineau S., Noizet G., Francois O., Feasson E. & Minoc B. (2003). French BNT site. `http://bnt.insa-rouen.fr/`.

[Michie *et al.*, 1994] 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/`.

[Munteanu & Bendou, 2002] Munteanu P. & Bendou M. (2002). The eq framework for learning equivalence classes of bayesian networks. In *First IEEE International Conference on Data Mining (IEEE ICDM)*, p. 417–424, San José.

[Murphy, 2001] Murphy K. (2001). The BayesNet Toolbox for Matlab, *Computing Science and Statistics: Proceedings of Interface*, 33. `http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html`.

[Naïm *et al.*, 2004] Naïm P., Wuillemin P.-H., Leray P., Pourret O. & Becker A. (2004). *Réseaux bayésiens*. Paris: Eyrolles.

[Pearl & Verma, 1991] Pearl J. & Verma T. S. (1991). A theory of inferred causation. In J. F. Allen, R. Fikes & E. Sandewall, Eds., *KR'91: Principles of Knowledge Representation and Reasoning*, p. 441–452, San Mateo, California: Morgan Kaufmann.

[Robinson, 1977] Robinson R. W. (1977). Counting unlabeled acyclic digraphs. In C. H. C. Little, Ed., *Combinatorial Mathematics V*, volume 622 of *Lecture Notes in Mathematics*, p. 28–43, Berlin: Springer.

[Schwartz, 1978] Schwartz G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, **6**(2), 461–464.

[Spirtes *et al.*, 2000] Spirtes P., Glymour C. & Scheines R. (2000). *Causation, Prediction, and Search*. The MIT Press, 2 edition.

[Verma & Pearl, 1990] Verma T. & Pearl J. (1990). Equivalence and synthesis of causal models. In *in Proceedings Sixth Conference on Uncertainty and Artificial Intelligence*, San Francisco: Morgan Kaufmann.