# Evaluation d'algorithmes d'apprentissage de structure pour les réseaux bayésiens

# Evaluation of structure learning algorithms for bayesian networks

O. Francois, Ph. Leray

INSA Rouen - Laboratoire PSI - FRE CNRS 2645
BP 08 - Av. de l'Université, 76801 St-Etienne du Rouvray Cedex
{Olivier.Francois, Philippe.Leray}@insa-rouen.fr

## Résumé

*Les réseaux bayésiens sont un formalisme de raisonnement probabiliste de plus en plus utilisé en classification pour des problèmes de fouille de données. Dans certaines situations, 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. De nombreuses méthodes d'apprentissage automatique de structure ont été proposées ces dernières années, proposant soit de simplifier l'espace de recherche (naïf augmenté, K2), soit d'utiliser une heuristique de recherche dans l'espace des réseaux bayésiens (recherche gloutonne). La plupart de ces méthodes ne marchent qu'avec des variables complètement observées, alors que d'autres prennent en compte les variables partiellement observées (SEM).*
*Nous avons testé les principales méthodes existantes sur deux grandes séries de problèmes. La première série est destinée à évaluer la précision des méthodes en essayant de retrouver différents graphes connus à partir de données. Ces expériences nous permettent aussi d'observer la robustesse des méthodes dans différentes situations (importance du nombre d'exemples disponibles, détection de relations "faibles" entre les variables, etc...). La seconde série d'expériences est destinée à tester l'efficacité de ces méthodes dans plusieurs cas d'utilisation en classification.*
*Ces expériences nous permettent de comparer des méthodes de complexité différentes. Elles nous offrent aussi la possibilité d'étudier certains problèmes d'initialisation pour lesquels nous proposons une approche simple permettant selon les algorithmes de stabiliser la méthode ou de converger plus rapidement vers la solution.*

## Mots Clef

Réseaux Bayésiens, Apprentissage de structure, Raisonnement Probabiliste, Classification, Aide à la Décision.

## Abstract

*Bayesian networks are a formalism for probabilistic reasonning that is more and more used for classification task in data-mining. In some situations, the network structure is given by an expert, otherwise, retrieving it from a database is a NP-hard problem, notably because of the search space complexity. In the last decade, a lot of methods were introduced to learn the network structure automatically, by simplifying the search space (augmented naive bayes, K2) or by using an heuristic in this search space (greedy search). Most of these methods deal with completely observed data, but some others can deal with incomplete data (SEM).*
*We have experimented most of the main methods with two series of tests. The first experiment tries to evaluate their precision in graph retrieval. The results enable us to study the robustness of these methods in weak relation detection, or to study their behavior according to the number of examples. The last experiment consists in studying their effectiveness in several classification problems. We also study the initialisation problem that appears in most of those methods and propose a solution to simplify it.*

## Keywords

Bayesian Networks, Structure Learning, Probabilistic Reasoning, Classification.

# 1 Introduction

Les réseaux bayésiens sont un formalisme de raisonnement probabiliste introduit par [20], [21], [17], [18].

**Définition 1** *$\mathcal{B} = (\mathcal{G}, \theta)$ est un réseau bayésien si $\mathcal{G} = (X, E)$ est un graphe acyclique dirigé (*DAG*) 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 $\mathcal{G}$.*

Un réseau bayésien $\mathcal{B}$ représente donc une distribution de probabilité sur $X$ qui admet la loi jointe suivante :

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

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 en classification si les interactions entre les différents critères peuvent être modélisés par des relations de probabilités conditionnelles.
Si 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 d'un réseau bayésien n'est pas simple, principalement à cause du fait que l'espace de recherche est de taille super-exponentielle en fonction du nombre de variables.
Nous allons commencer par introduire quelques notions générales concernant la structure des réseaux bayésiens, la façon de leur associer un score 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 certaines techniques prometteuses pour l'apprentissage de structure des réseaux bayésiens.

# 2 Généralités

La première idée pour trouver la meilleure structure d'un réseau bayésien (mais aussi la plus naïve), est de parcourir tous les graphes possibles, de leur associer un score, puis de choisir le graphe ayant le score le plus élevé. Robinson [27] 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.

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

Ce qui donne $r(3) = 25$, $r(5) = 29281$, $r(10) \simeq 4,2 \times 10^{18}$. Comme l'équation 2 est super-exponentielle, il est impossible d'effectuer un parcours exhaustif de l'espace des graphes acycliques dirigés (DAG) en un temps raisonnable dès que le nombre de nœuds dépasse 7 ou 8. Les méthodes d'apprentissage de structure utilisent alors des heuristiques de recherche.
Pour que cette recherche soit efficace, si le parcours de l'ensemble des DAG s'effectue avec des opérateurs du type *ajout* ou *suppression* d'arcs, il est avantageux d'avoir un score calculable localement pour limiter les calculs à la variation du score entre deux DAG voisins.

**Définition 2** *Un score $S$ est dit* décomposable *s'il peut s'écrire comme une somme ou un produit de termes qui sont fonctions 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 :*

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

**Définition 3** *Deux* DAG *sont dits* équivalents au sens de Markov *s'ils* encodent *les mêmes lois jointes.*

Verma et Pearl [31] ont montré que deux DAG sont équivalents si et seulement si ils ont le même squelette (*i.e.* le même support d'arêtes) et les mêmes structures en V (du type $A \rightarrow C \leftarrow B$). Il peut alors être intéressant d'associer le même score à toutes les structures qui sont équivalentes.

**Définition 4** *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 [28] et a la forme suivante :

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

où $D$ est notre base d'exemples, $\theta^{MV}$ est la distribution des paramètres du réseau $\mathcal{B}$ obtenue par *maximum de vraisemblance* et où $Dim(\mathcal{B})$ est la dimension de $\mathcal{B}$ 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 la matrice $\mathbb{P}(X_i/Pa(X_i))$ il faudra $Dim(X_i, \mathcal{B})$ paramètres avec

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

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

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

Le score BIC de l'équation 3 est la somme d'un terme de vraisemblance du réseau par rapport au données, et d'un terme qui pénalise les structures complexes. Ce score est *score équivalent* car deux graphes équivalents ont la même vraisemblance (ils représentent les mêmes indépendances conditionnelles) et la même complexité.
En utilisant des scores ayant cette propriété, il devient possible de rechercher des structures directement dans l'espace des équivalents de Markov. Cet espace a des propriétés intéressantes : 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 convergera plus facilement.

# 3 Algorithmes

## 3.1 Algorithme PC, recherche de causalité

L'algorithme PC a été introduit par Spirtes, Glymour et Scheines [29] en 1993. (Pearl et Verma [26] ont proposé un algorithme similaire à la même époque).
Cette méthode utilise un test statistique pour estimer s'il y a indépendance conditionnelle entre deux variables et construire la structure du graphe. En pratique, on démarre avec un graphe complètement connecté, en retirant une arête pour chaque indépendance conditionnelle détectée. Ensuite les arêtes sont orientées lorsque des V-structures sont détectées.
Une amélioration récente de PC a été introduite par [6] (nommée BN-PC-B).

## 3.2 Arbre de recouvrement maximal

Chow et Liu [9] ont proposé une méthode dérivée de la recherche de l'arbre de recouvrement de poids maximal (MWST, *maximum weight spanning tree*). Cette méthode associe un poids à chaque arête potentielle $A$–$B$ de l'arbre. Ce poids peut être l'*information mutuelle* entre les variables $A$ et $B$ comme proposé par [9], ou encore la variation du score local lorsqu'on choisit $B$ comme parent de $A$ [16]. Une fois cette matrice de poids définie, il suffit d'utiliser un des algorithmes standards de résolution du problème de l'arbre couvrant de poids maximum comme l'algorithme de Kruskal ou celui de Prim [11], et d'orienter l'arbre non dirigé obtenu en choisissant une racine puis en parcourant l'arbre par une recherche en profondeur.

## 3.3 Structures de Bayes naïves

Le classifieur de Bayes naïf peut être vu comme un réseau bayésien dont la structure contient uniquement des arcs du nœud classe $C$ vers les autres nœuds $X_i$ (variables observées). Cette structure est liée à 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 structure de Bayes naïve suppose que les observations sont indépendantes conditionnellement à la classe. Cette hypothèse peut être levée en utilisant un classifieur de Bayes naïf *augmenté* [19, 13] où les observations peuvent être reliées selon une structure plus ou moins compliquée. Nous utilisons plus précisément comme structure augmentée le meilleur arbre reliant les observations (TANB, *Tree Augmented Naive Bayes*), arbre obtenu grâce à l'algorithme MWST [15].

## 3.4 Algorithme K2

Le principe 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 $\mathbb{P}(\mathcal{G}, D)$, [10] a donné le résultat suivant.

**Théorème 1** *Soit $D$ une base de $N$ exemples et soit $\mathcal{G}$ une structure de réseau sur $X$. De plus, si $pa_{ij}$ est la $j^{ieme}$ instantiation de $Pa(X_i)$, $N_{ijk}$ le nombre d'exemples où $X_i$ a la valeur $x_{ik}$ et $Pa(X_i)$ est instantié en $pa_{ij}$, et $Nij = \sum_{k=1}^{r_i} N_{ijk}$ alors*

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

*avec*

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

*où $\mathbb{P}(\mathcal{G})$ est la probabilité* a priori *de la structure $\mathcal{G}$.*

L'équation 6 peut être vue comme une mesure de qualité de la structure $\mathcal{G}$ par rapport aux données et est appelée *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.

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

[10] propose ensuite de simplifier l'espace de recherche en ordonnant les nœuds. Un nœud ne peut alors être parent d'un autre que s'il précède ce dernier dans l'ordre fixé. L'espace de recherche devient alors l'ensemble des réseaux bayésiens respectant cet ordre d'énumération.
L'algorithme K2 teste l'ajout de parents selon l'ordre fixé. Le premier nœud ne peut pas posséder de parent. Pour les nœuds suivants, on choisit l'ensemble de parents qui obtient le meilleur score bayésien. [16] a montré que le score bayésien n'est 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 ou le score MDL [4] tous deux *score équivalents*.

## 3.5 Recherche gloutonne

L'algorithme de recherche gloutonne (GS, *greedy search*) est très répandu en optimisation. Il part d'un premier graphe, en définit un voisinage, puis associe un score à chaque graphe du voisinage. Le meilleur graphe est alors choisi pour l'itération suivante.
Dans le cas des réseaux bayésiens, le voisinage d'un DAG est défini par l'ensemble des graphes obtenus en ajoutant, supprimant, ou en retournant un arc du graphe courant.
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.
Cet algorithme calculant le score de graphes voisins, il est nécessaire d'utiliser un score décomposable pour n'avoir à calculer que la variation locale du score entraînée par la suppression, l'ajout ou le retournement d'un arc et non les scores globaux de ces deux graphes. Pour les expérimentations, cette méthode utilisera le score BIC.

## 3.6 Recherche gloutonne dans l'espace des équivalents de Markov

De récents travaux montrent l'intérêt de parcourir l'espace des équivalents de Markov (cf définition 3) au lieu de l'espace des réseaux bayésiens. [24] a montré que cet espace avait de meilleures propriétés que l'espace des réseaux bayésiens dans lequel une recherche gloutonne utilisant un score *score équivalent* peut converger vers un maximum local au lieu de trouver la structure optimale. Ces notions ont été mises en œuvre par [7, 5, 1] dans de nouvelles méthodes d'apprentissage de structure. [8] 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 (représentants des classes d'équivalence de Markov) en deux phases (une phase d'ajout d'arcs, puis une phase de suppression).

### 3.7 Algorithme EM structurel

Cet algorithme (SEM, *Structural EM*) a été introduit par [12]. Il est basé sur le principe de l'algorithme *Expectation Maximisation* (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, dont la convergence à été prouvée par [12], part d'une structure initiale pour estimer la distribution de probabilité des variables *cachées* ou *manquantes* grâce à l'algorithme EM classique. L'espérance du score par rapport à ces variables *cachées* est ensuite calculée pour tous les réseaux bayésiens du voisinage afin de choisir la structure suivante.

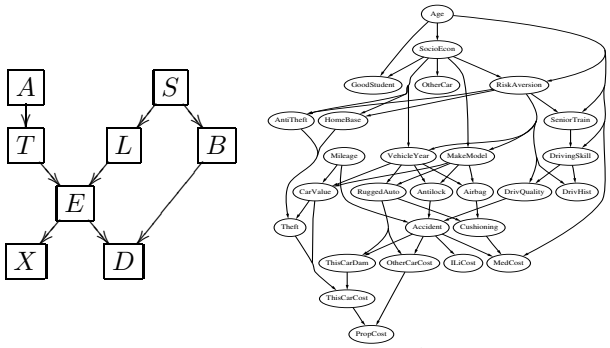
### 3.8 Problèmes d'initialisation

La plupart des méthodes proposées précédemment ont des problèmes d'initialisation. Par exemple, K2 dépend fortement de l'ordre d'énumération fixé. Suivant une recommandation de [16] nous proposons d'utiliser l'arbre rendu par l'algorithme MWST (qui est rapide) pour générer cet ordre. La difficulté se réduit alors au choix du nœud racine de l'arbre. Ceci peut être fait de manière aléatoire, ou en choisissant le nœud classe pour des tâches de classification. Il suffit alors d'utiliser l'ordre topologique de l'arbre orienté comme initialisation de K2. Nous appellerons "K2+T" cet algorithme. Nous proposons aussi l'algorithme "K2-T" utilisant l'ordonnancement inverse des nœuds pour lequel la classe peut être interprétée comme une *conséquence* plutôt qu'une *cause*.
La recherche gloutonne peut aussi être initialisée à l'aide d'un DAG particulier. Si ce DAG n'est pas fourni par un expert, nous proposons aussi d'utiliser l'arbre obtenu par MWST comme point de départ de la recherche gloutonne au lieu d'un graphe non connecté, ce qui donne un algorithme que nous appellerons "GS+T".

## 4 Expérimentation

### 4.1 Implémentation des algorithmes

Nous avons utilisé Matlab avec la boîte à outils BNT (Bayes Net Toolbox [25]) ainsi que le package *Apprentissage de Structure* que nous proposons sur le site français de BNT [22].
Les algorithmes utilisés dans les expériences qui suivent sont PC (recherche de causalité), MWST (arbre de recouvrement minimal), K2 (avec deux initialisations aléatoires), K2+T (K2 avec l'ordonnancement des nœuds issu de MWST), K2-T (K2 avec un ordonnancement des nœuds inverse de celui fourni par MWST), GS (en commençant avec une structure vide), GS+T (en commençant avec l'arbre fourni par MWST), GES (recherche gloutonne dans l'espace des équivalents de Markov) et SEM (recherche gloutonne avec prise en compte des données manquantes, en commençant avec une structure vide). Nous avons aussi utilisé NB (structure de Bayes naïve) et TANB (structure de Bayes naïve augmentée par un arbre) pour des tâches de classification.



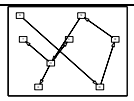
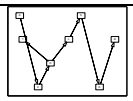
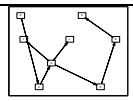
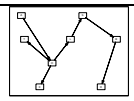
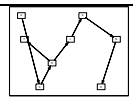
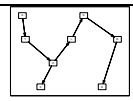
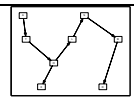
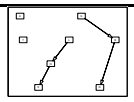
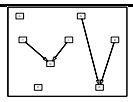
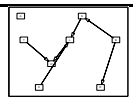
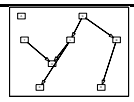
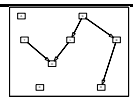
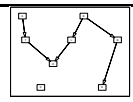
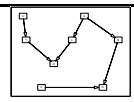
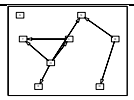
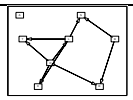
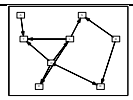
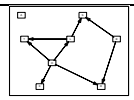
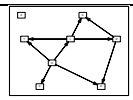
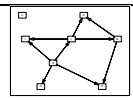
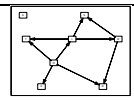
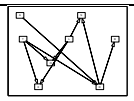
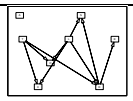
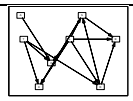
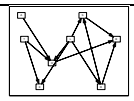
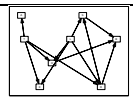
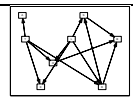
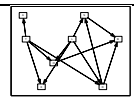
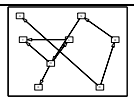
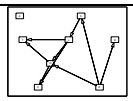
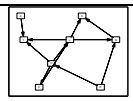
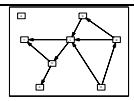
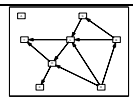
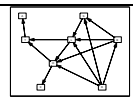
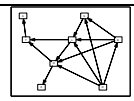
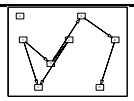
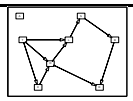
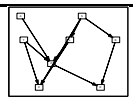
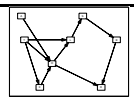
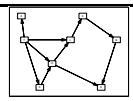

(a) ASIA (b) INSURANCE

FIG. 1 – Réseaux de référence

### 4.2 Retrouver une structure connue

**Réseaux de référence et critères d'évaluation.**
Nous allons utiliser deux réseaux bayésiens dont la structure est déjà connue. Le premier réseau (ASIA) provient du diagnostic de la dyspnée introduit par Lauritzen et Spiegelhalter [21] (voir figure 1.a). Les huit nœuds sont binaires. Notons à propos de l'arc entre $A$ et $T$ que la probabilité *a priori* que $A$ soit vraie est très faible et que l'influence de $A$ sur $T$ est également très faible.
Le second réseau utilisé (INSURANCE) possède 27 nœuds (voir figure 1.b) et est téléchargeable sur [14].
A partir de ces réseaux bayésiens, nous avons généré des bases d'exemples de taille variable pour tester l'influence du nombre d'exemples sur les résultats. Ces mêmes bases ont aussi été vidées aléatoirement de $20\%$ de leurs valeurs pour tester l'algorithme SEM.
Pour comparer les résultats issus de ces différents algorithmes, nous allons utiliser la distance d'édition définie par le nombre minimum d'opérations nécessaire pour transformer le graphe obtenu en celui d'origine (les opérateurs sont l'ajout, le renversement ou la suppression d'un arc). Remarquons ici que le renversement d'un arc est considéré comme un opérateur indépendant et non comme la succession de deux opérations.
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.

**Résultats et interprétations.**
*Influence du nombre d'exemples*
La figure 2 nous révèle que l'algorithme MWST est peu sensible à la variation de la taille de la base de données. Il donne rapidement un graphe proche du graphe d'origine en dépit d'un espace de recherche assez pauvre (l'espace des arbres).
L'heuristique PC donne également de bons résultats avec peu d'arcs oubliés ou mal placés.
La méthode K2, très souvent utilisée dans la littérature, est très rapide mais reste cependant trop sensible à l'initiali-

| **ASIA** | 250 | 500 | 1000 | 2000 | 5000 | 10000 | 15000 |
|---|---|---|---|---|---|---|---|
| MWST |  |  |  |  |  |  |  |
| | 9 ;-68837 | 10 ;-69235 | 8 ;-68772 | 6 ;-68704 | 7 ;-68704 | 3 ;-68694 | 3 ; -68694 |
| PC |  |  |  |  |  |  |  |
| | 8 ;-55765 | 7 ;-66374 | 6 ;-61536 | 7 ;-56386 | 6 ;-63967 | 5 ;-63959 | 6 ;-70154 |
| K2 |  |  |  |  |  |  |  |
| | 8 ;-68141 | 7 ;-67150 | 6 ;-67152 | 6 ;-67147 | 6 ;-67106 | 6 ;-67106 | 6 ;-67106 |
| K2(2) |  |  |  |  |  |  |  |
| | 11 ;-68643 | 11 ;-68089 | 11 ;-67221 | 10-67216 ; | 9 ;-67129 | 9 ;-67129 | 9 ;-67129 |
| K2+T |  |  |  |  |  |  |  |
| | 10 ;-68100 | 8 ;-68418 | 9 ;-67185 | 8 ;-67317 | 8 ;-67236 | 10 ;-67132 | 10 ;-67132 |
| K2-T |  |  |  |  |  |  |  |
| | 7 ;-68097 | 6 ;-67099 | 6 ;-67112 | 7 ;-67105 | 6 ;-67091 | 5 ;-67091 | 5 ;-67091 |
| GS |  |  |  |  |  |  |  |
| | 4 ;-67961 | 9 ;-68081 | 2 ;-67093 | 5 ;-67096 | 7 ;-67128 | 9 ;-67132 | 8 ;-67104 |
| GS+T |  |  |  |  |  |  |  |
| | 9 ;-68096 | 6 ;-68415 | 2 ;-67093 | 7 ;-67262 | 2 ;-67093 | 2 ;-67093 | 1 ;-67086 |
| GES |  |  |  |  |  |  |  |
| | 4 ;-68093 | 6 ;-68415 | 5 ;-67117 | 2 ;-67094 | 0 ;-67086 | 0 ;-67086 | 0 ;-67086 |
| SEM |  |  |  |  |  |  |  |
| | 10 ;-83615 | 9 ;-81837 | 2 ;-67093 | 8 ;-67384 | 4 ;-67381 | 5 ;-67108 | 4 ;-67381 |

| **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 |
| 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 |

FIG. 2 – Distance d'édition au graphe d'origine et score BIC (divisés par 100 et arrondis dans le cas d'INSURANCE) pour plusieurs méthodes d'apprentissage de structure (en lignes) et pour plusieurs tailles de bases d'exemples (en colonnes).

sation. La figure 2 donne les résultats de K2 avec deux ordonnancements choisis aléatoirement ("ELBXASDT" et "TALDSXEB" dans le cas ASIA). Pour un ordre fixé, K2 trouve toujours le même graphe. Par contre, en changeant d'ordonnancement, le graphe final change radicalement. Cela s'observe aussi pour le réseau INSURANCE, 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.

L'algorithme GS est lui aussi robuste face à la variation de la taille de la base d'exemples, surtout s'il est initialisé avec l'arbre obtenu par MWST.

La méthode GES donne de bons résultats quel que soit la taille de la base d'exemples. Pour un grand nombre de données, les résultats de GES sont meilleurs que ceux obtenu par une recherche gloutonne classique du point de vue du score. Par contre, pour INSURANCE ils sont nettement moins bons pour la distance d'édition et même lorsque l'on les compare à ceux de GS+T.

L'algorithme SEM trouve des structures très proches de la structure originale (avec $20\%$ de données manquantes) quelle que soit la taille de la base d'apprentissage. Notons que cette méthode obtient une distance d'édition assez élevée car elle trouve souvent une structure mal orientée. Il est difficile de savoir si une mauvaise orientation est une erreur grave ou non (par exemple en faisant disparaître une V-structure). Pour essayer de résoudre ce problème, nous travaillons actuellement à une mesure d'édition travaillant directement sur les classes d'équivalence de Markov.

*Découverte de dépendances faibles*

Pour la plupart des méthodes, l'arc $A$–$T$ du réseau bayésien ASIA n'a pas été trouvé. Les différentes méthodes y parviennent quelquefois lorsque la base d'exemples est suffisamment grande.

Ce phénomène s'explique pour la plupart des méthodes à base de score : l'ajout de cet arc ne permet pas d'augmenter le score global du réseau bayésien car l'augmentation de la vraisemblance est faible par rapport à la baisse du terme de pénalité associé à la dimension du réseau.

## 4.3 Recherche d'un bon réseau bayésien pour la classification

**Bases d'exemples et critère d'évaluation.**

ASIA : Nous avons utilisé la base de $2000$ exemples générée pour l'expérience précédente pour l'apprentissage, et la base de $1000$ exemples comme base de test.

HEART : Cette base d'exemples, issue du projet Statlog [30, 23], est une base de diagnostic médical avec $14$ variables (les variables continues ont été discrétisées). Elle contient $270$ exemples séparés en $189$ pour l'apprentissage et $81$ pour le test.

AUSTRALIAN : Cette base d'exemples, disponible sur [23], 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.

THYROID : Cette base d'exemples, disponible sur [3], est une base de diagnostic médical dont nous avons utilisé $22$ variables (parmi $29$) : $15$ variables discrètes, $6$ continues qui ont été discrétisés, et la classe. Elle contient $2800$ exemples d'apprentissage et $972$ de test.

CHESS : Cette base de données est également disponible sur [3] (King+Rook versus King+Pawn). Il s'agit de prédire si les blancs peuvent gagner une partie d'échec à partir de la description de la position courante. Il y a $37$ variables ($36$ observations et la classe) mesurées sur $3196$ exemples répartis en $2200$ pour l'apprentissage et $996$ pour le test.

Evaluation

Le critère de comparaison entre les méthodes est le taux de bonne classification mesuré sur les données de test, avec un intervalle de confiance à $\alpha\%$ proposé par [2] (cf éq. 8).

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

où $N$ est le nombre d'exemples, $T$ le taux de bonne classification et $Z_\alpha = 1.96$ pour $\alpha = 95\%$.

**Résultats et interprétations.**

Les performances et les intervalles de confiance des différents classifieurs obtenus sont résumés dans la table 1 et comparés avec un classifieur de type k-plus-proches-voisins ($k = 9$). Notons que le *memory crash* obtenu par l'algorithme PC pour des problèmes de taille moyenne est du à l'implémentation actuelle de l'algorithme. Spirtes *et al.* [29] proposent une heuristique qui permet de remédier à ce problème.

Pour des problèmes de classification simples comme ASIA, un réseau bayésien naïf donne d'aussi bons résultats qu'une structure plus évoluée, ou qu'un autre algorithme comme les kPPV. Par contre, l'arbre obtenu par MWST fait aussi bien, voire mieux que ce réseau bayésien naïf. Il parait donc judicieux de ne pas se priver de cette méthode qui, à un moindre coût, donne des résultats au moins aussi bons que le réseau bayésien naïf qui est régulièrement utilisé lorsque la structure du réseau n'est pas connue. Par contre, de manière surprenante, la structure naïve augmentée par un arbre ne donne pas de meilleurs résultats sur ces exemples. Même si cette méthode permet de relâcher la contrainte d'indépendance conditionnelle des observations, les réseaux bayésiens obtenus possèdent un nombre élevé d'arcs et ainsi un nombre important de paramètres à estimer.

Pour des problèmes plus compliqués comme THYROID et CHESS, l'apprentissage de structure permet d'obtenir des performances supérieures à celles du réseau naïf.

Contrairement aux expériences précédentes sur la recherche de structure, l'initialisation de l'algorithme K2 par l'arbre obtenu par MWST ne permet pas d'améliorer les résultats en classification. Par contre, elle aide à stabiliser la méthode en se démarchant du choix de l'ordonnancement initial des nœuds.

| | ASIA | HEART | AUTRALIAN | THYROID | CHESS |
|---|---|---|---|---|---|
| Nvar, Napp, Ntest | 8, 2000, 1000 | 14, 189, 81 | 15, 500, 190 | 22, 2800, 972 | 37, 2200, 996 |
| NB | 86,5% [84,2 ;88,5] | 87,6% [78,7 ;93,2] | 87,9% [82,4 ;91,8] | 95,7% [94,2 ;96,9] | 86,6% [84,3 ;88,6] |
| TANB | 86,5% [84,2 ;88,5] | 81.5% [71,6 ;88,5] | 86,3% [80,7 ;90,5] | 95,4% [93,8 ;96,6] | 86,4% [84,0 ;88,4] |
| MWST-BIC | 86,5% [84,2 ;88,5] | 86,4% [77,3 ;92,3] | 87,4% [81,8 ;91,4] | 96,8% [95,4 ;97,8] | 89,5% [87,3 ;91,3] |
| MWST-MI | 86,5% [84,2 ;88,5] | 82,7% [73,0 ;89,5] | 85,8% [80,1 ;90,1] | 96,1% [94,6 ;97,8] | 89,5% [87,3 ;91,3] |
| PC | 84,6% [82,2 ;86,8] | 85,2% [75,7 ;91,3] | 86,3% [80,7 ;90,5] | memory crash | memory crash |
| K2 | 86,5% [84,2 ;88,5] | 83,9% [74,4 ;90,4] | 83,7% [77,8 ;88,3] | 96,3% [94,9 ;97,4] | 92,8% [90,9 ;94,3] |
| K2+T | 86,5% [84,2 ;88,5] | 81,5% [71,6 ;88,5] | 84,2% [78,3 ;88,8] | 96,3% [94,9 ;97,4] | 92,6% [90,7 ;94,1] |
| K2-T | 86,5% [84,2 ;88,5] | 76,5% [66,2 ;84,5] | 85,8% [80,1 ;90,1] | 96,1% [94,6 ;97,2] | 93,0% [91,2 ;94,5] |
| GS | 86,5% [84,2 ;88,5] | 85,2% [75,8 ;91,4] | 86,8% [81,3 ;91,0] | 96.2% [94,7 ;97,3] | 94,6% [93,0 ;95,9] |
| GS+T | 86,2% [83,9 ;88,3] | 82,7% [73,0 ;89,5] | 86,3% [80,7 ;90,5] | 95,9% [94,4 ;97,0] | 92,8% [90,9 ;94,3] |
| GES | 86,5% [84,2 ;88,5] | 85,2% [75,8 ;91,4] | 84,2% [78,3 ;88,8] | 95,9% [94,4 ;97,0] | 93,0% [91,2 ;94,5] |
| SEM | 86,5% [84,2 ;88,5] | 80.2% [70,2 ;87,5] | 74,2% [67,5 ;80,0] | 96.2% [94,7 ;97,3] | 89,2% [87,1 ;91,0] |
| kPPV | 86,5% [84,2 ;88,5] | 85,2% [75,8 ;91,4] | 80,5% [74,3 ;85,6] | 98,8% [97,8 ;99,4] | 94,0% [92,3 ;95,4] |

TAB. 1 – Pourcentage de bonne classification en test et intervalle de confiance à $95\%$ pour des classifieurs obtenus par plusieurs méthodes d'apprentissage de structure.

Il est surprenant de voir que l'algorithme GS ne parvient pas à trouver de réseau bayésien donnant de meilleurs résultats malgré son parcours plus complet de l'espace des DAG. Cela peut s'expliquer par la taille de l'espace de recherche et par le nombre important d'optima locaux dans cet espace.
Sur le plan théorique, la méthode GES est la méthode à base de score la plus évoluée. Lors des expériences précédentes elle permettait de trouver une structure ayant un très bon score. Sur nos problèmes de classification, les performances sont cependant 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 les performances équivalentes à celles apprises avec toutes les données.
Les classifieurs obtenus donnent souvent des résultats comparables aux kPPV. De plus, il faut noter qu'un réseau bayésien permet aussi d'effectuer d'autres opérations comme l'inférence sur d'autres nœuds, l'interprétation de la structure obtenue, ou la 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 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. Ici, nous avons d'abord montré qu'une "bonne" recherche de structure permet d'obtenir des résultats équivalents à un algorithme tel que les kPPV. Chose surprenante, des méthodes très simples comme la structure de Bayes naïf ou MWST permettent d'obtenir des résultats aussi bons que des méthodes plus complexes.
L'adaptation des méthodes existantes à la prise en compte de données manquantes est importante pour pouvoir traiter des problèmes réels. L'algorithme EM structurel permet déjà cela pour une recherche gloutonne dans l'espace des réseaux bayésiens. Le même principe pourrait être appliqué à MWST, permettant d'obtenir très rapidement une structure simple possédant de bonnes propriétés, ou à GES pour la prise en compte des données manquantes pour les méthodes parcourant l'espace des équivalents de Markov.

# Références


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

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

[3] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. http ://www.ics.uci.edu/∼mlearn/MLRepository.html.

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

[5] R. Castelo and T. Kocka. Towards an inclusion driven learning of bayesian networks. Technical Report UU-CS-2002-05, Institute of information and computing sciences, University of Utrecht, 2002.

[6] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning Bayesian networks from data : An information-theory based approach. *Artificial Intelligence*, 137(1–2) :43–90, 2002.

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

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

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

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

[11] T. Cormen, C. Leiserson, and R. Rivest. *Introduction à l'algorithmique*. Dunod, 1994.

[12] N. Friedman. The Bayesian structural EM algorithm. In Gregory F. Cooper and Serafín Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 129–138, San Francisco, July 24–26 1998. Morgan Kaufmann.

[13] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3) :131–163, 1997.

[14] N. Friedman, M. Goldszmidt, D. Heckerman, and S. Russell. 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, 1997. http ://www.cs.huji.ac.il/labs/compbio/Repository/networks.html.

[15] D. Geiger. An entropy-based learning algorithm of bayesian conditional trees. In *Uncertainty in Artificial Intelligence : Proceedings of the Eighth Conference (UAI-1992)*, pages 92–97, San Mateo, CA, 1992. Morgan Kaufmann Publishers.

[16] D. Heckerman, D. Geiger, and M. Chickering. Learning Bayesian networks : The combination of knowledge and statistical data. In Ramon Lopez de Mantaras and David Poole, editors, *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 293–301, San Francisco, CA, USA, July 1994. Morgan Kaufmann Publishers.

[17] F.V. Jensen. *Introduction to Bayesian Networks*. Springer Verlag, 1996.

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

[19] E. Keogh and M. Pazzani. Learning augmented bayesian classifiers : A comparison of distribution-based and classification-based approaches. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, pages 225–230, 1999.

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

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

[22] P. Leray, S. Guilmineau, G. Noizet, and O. Francois. French BNT site, 2003. http ://bnt.insa-rouen.fr/.

[23] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine Learning*, *Neural and Statistical Classification*, 1994. http ://www.amsta.leeds.ac.uk/ charles/statlog/.

[24] P. Munteanu and M. Bendou. The EQ framework for learning equivalence classes of bayesian networks. In *First IEEE International Conference on Data Mining (IEEE ICDM)*, pages 417–424, San José, 2001.

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

[26] J. Pearl and T.S. Verma. A theory of inferred causation. In James F. Allen, Richard Fikes, and Erik Sandewall, editors, *KR'91 : Principles of Knowledge Representation and Reasoning*, pages 441–452, San Mateo, California, 1991. Morgan Kaufmann.

[27] R.W. Robinson. Counting unlabeled acyclic digraphs. In C. H. C. Little, editor, *Combinatorial Mathematics V*, volume 622 of *Lecture Notes in Mathematics*, pages 28–43, Berlin, 1977. Springer.

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

[29] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. The MIT Press, 2 edition, 2000.

[30] A. I. Sutherland and R. J. Henery. Statlog - an ESPRIT projecy for the comparison of statistical and logical learning algorithms. *New Techniques and Technologies for Statistics*, February 1992.

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