MADI

Apprentissage dans les réseaux Bayésiens

Pierre-Henri WUILLEMIN

DESIR
LIP6
pierre-henri.wuillemin@lip6.fr

Apprendre quoi ?

Apprentissage dans les réseaux bayésiens

L'apprentissage a pour but d'estimer, à partir d'une base de données et de connaissances a priori :

- La structure du réseau bayésien (X parent de Y ?)
- Les paramètres du réseau bayésien ($P(X = 0 \mid Y = 1)$?)

La base de données peut être :

- complète,
- incomplète.

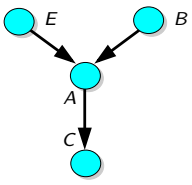
Les connaissances a priori sont très variables ; par exemple :

- structure du BN connue,
- Loi a priori pour certaines variables, etc.

Ce qui donne 4 cadres principaux de l'apprentissage dans les réseaux Bayésiens : "Apprentissage de {paramètres |structure} avec données {complètes |incomplètes}".

Apprentissage des paramètres, données complètes

$$D : \begin{bmatrix} d_1^A & d_1^B & d_1^C & d_1^E \\ \dots & \dots & \dots & \dots \\ V & F & F & V \\ \dots & \dots & \dots & \dots \\ d_M^A & d_M^B & d_M^C & d_M^E \end{bmatrix}$$



En appelant Θ l'ensemble des paramètres du modèle et $L(\Theta : D)$ la vraisemblance :

$$\begin{aligned} L(\Theta : D) &= P(D \mid \Theta) \\ &= \prod_{m=1}^{M} P(d_m \mid \Theta) && \text{(échantillons indépendants, identiquement distribués)} \\ &= \prod_{m=1}^{M} P(E = d_m^E, B = d_m^B, A = d_m^A, C = d_m^C \mid \Theta) \end{aligned}$$

Apprentissage des paramètres, données complètes (2)

En renommant E, B, A, C par $n = 4$, $(X_i)_{1 \leq i \leq n}$,

$$\begin{aligned} L(\Theta : D) &= \prod_{m=1}^{M} P(X_1 = d_m^1, X_2 = d_m^2, \cdots, X_n = d_m^n \mid \Theta) \\ &= \prod_{m=1}^{M} \prod_{i=1}^{n} P(X_i \mid Pa_i, \Theta) \\ &= \prod_{i=1}^{n} \prod_{m=1}^{M} P(X_i \mid Pa_i, \Theta_i) \end{aligned}$$

$L(\Theta : D) = \prod_{i=1}^{n} L_i(\Theta_i : D)$

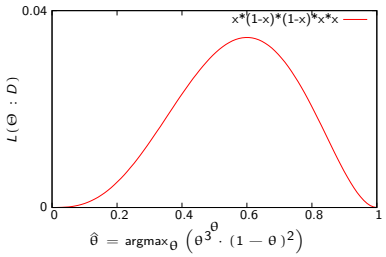
L'estimation des paramètres d'un réseau bayésien se décomposent en l'estimation des paramètres de chaque loi de probabilité conditionnelle

Maximisation de la vraisemblance (MLE)

Soit une variable binaire X. Avec $\theta = P(X = 1)$:

$$\Theta = \{\theta, 1 - \theta\}$$
$$D = (1, 0, 0, 1, 1)$$
$$L(\Theta : D) = \prod_{m} P(X = d_m \mid \Theta)$$

Ici : $L(\Theta : D) = \theta \cdot (1 - \theta) \cdot (1 - \theta) \cdot \theta \cdot \theta$.



Généralisation pour une variable multinomiale

Pour X v.a. de valeurs $(1, \cdots, r)$,
avec $\Theta_X = (\theta_1, \cdots, \theta_r)$ où $\theta_i = P(X = i)$,
et $N_i = \#_D(X = i)$ nombre d'occurence de i dans D,

$$L(\Theta_X : D) = \prod_{i=1}^{r} \theta_i^{N_i} \qquad \text{et} \qquad \widehat{\Theta}_X = \text{argmax}_{\Theta_X} \left(L(\Theta_X : D) \right)$$

Maximum de vraisemblance dans un réseau bayésien

$\theta_{ijk} = P(X_i = k \mid Pa_i = j)$, $N_{ijk} = \#_D(X_i = k, Pa_i = j)$, $k \in \{1 \cdots r_i\}$, $j \in \{1 \cdots q_i\}$

- $L(\Theta : D) = \prod_{i=1}^{n} L_i(\Theta_i : D) = \prod_{i=1}^{n} \prod_{m=1}^{M} P(X_i = k_m \mid Pa_i = j_m, \Theta_i)$

$$L(\Theta : D) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}$$

- $LL(\Theta : D) = \sum_{i=1}^{n} \sum_{m=1}^{M} \log P(X_i \mid Pa_i, \Theta_i) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \theta_{ijk}$
- On sait que $\sum_k \theta_{ijk} = 1$ soit $\theta_{ijr_i} = 1 - \sum_{k=1}^{r_i - 1} \theta_{ijk}$ d'où

$$LL(\Theta : D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left(\sum_{k}^{r_i - 1} N_{ijk} \log \theta_{ijk} + N_{ijr_i} \log \left(1 - \sum_{k=1}^{r_i - 1} \theta_{ijk} \right) \right)$$

- On cherche $\widehat{\Theta}$ maximisant $L(\Theta : D)$ et donc $LL(\Theta : D)$:

$$\text{i.e. } \widehat{\Theta} \text{ tel que } \forall i, \forall j, \forall k, \frac{\partial LL(\Theta : D)}{\partial \theta_{ijk}} \left(\widehat{\Theta} \right) = \frac{N_{ijk}}{\widehat{\theta}_{ijk}} - \frac{N_{ijr_i}}{1 - \sum_{k=1}^{r_i - 1} \widehat{\theta}_{ijk}} = \frac{N_{ijk}}{\widehat{\theta}_{ijk}} - \frac{N_{ijr_i}}{\widehat{\theta}_{ijr_i}} = 0$$

- Finalement, $\frac{N_{ijr_i}}{\widehat{\theta}_{ijr_i}} = \frac{N_{ij1}}{\widehat{\theta}_{ij1}} = \cdots = \frac{N_{ij(r_i - 1)}}{\widehat{\theta}_{ij(r_i - 1)}}$ (et $\sum_k \widehat{\theta}_{ijk} = 1$) d'où

$$\forall k \in \{1, ..., r_i\}, \widehat{\theta}_{ijk} = \frac{N_{ijk}}{N_{ij}} \qquad \text{avec } N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$$

Prédiction bayésienne

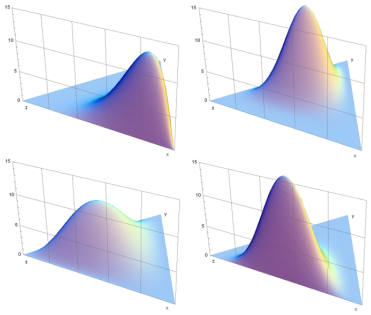
Θ suit une distribution $P(\Theta \mid D)$.

$$P(\Theta \mid D) \propto P(D \mid \Theta) \cdot P(\Theta) = L(\Theta : D) \cdot P(\Theta)$$

Cette méthode permet de prendre en compte un *a priori* sur Θ ; pour intégrer des connaissances d'expert ou pour rendre plus stable les estimations avec un petit échantillon D.

Distribution de Dirichlet :
$f(p_1, \cdots, p_K; \alpha_1, \cdots \alpha_K) \propto \prod_{i=1}^{K} x_i^{\alpha_i - 1}$
(où $\sum_i p_i = 1$)
Intuitivement, f se lit comme :
$$P\left(P(X = i) = p_i \mid \#_{X=i} = \alpha_i - 1\right)$$

En supposant que l'a priori $P(\Theta)$ soit une distribution de Dirichlet :
$$P(\Theta) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk} - 1}$$



[Wikipedia] Clockwise from top left :
$\alpha = (6, 2, 2), (3, 7, 5), (6, 2, 6), (2, 3, 4)$

Prédiction bayésienne (2)

À partir de :
- $P(\Theta \mid D) \propto L(\Theta : D) \cdot P(\Theta)$
- $P(\Theta) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk} - 1}$
- $L(\Theta : D) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}$

$$P(\Theta \mid D) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk} + \alpha_{ijk} - 1}$$

MAP : maximum a posteriori

$$\widehat{\Theta}^{\text{MAP}} = \arg\max_{\Theta} P(\Theta \mid D)$$

$$\hat{\theta}_{ijk}^{\text{MAP}} = \frac{N_{ijk} + \alpha_{ijk} - 1}{\sum_k (N_{ijk} + \alpha_{ijk} - 1)}$$

EAP : espérance a posteriori

$$\widehat{\Theta}^{\text{EAP}} = \int_{\Theta} \Theta \cdot P(\Theta \mid D)\, d\Theta$$

$$\hat{\theta}_{ijk}^{\text{EAP}} = \frac{N_{ijk} + \alpha_{ijk}}{\sum_k (N_{ijk} + \alpha_{ijk})}$$

Apprentissage des paramètres, données complètes - résumé

Avec N_{ijk} le nombre de fois où la variable X_i a pris la valeur k et ses parents la valeur (t-uple) j et α_{ijk} les paramètres d'un a priori de Dirichlet.

Estimation des paramètres

Deux méthodes possibles pour l'estimation des paramètres :

- MLE (Maximum Likelihood Estimation)

$$\widehat{\theta}_{ijk} = \widehat{\theta}_{\{x_i = k \mid pa_i = j\}} = \frac{N_{ijk}}{N_{ij}}$$

- Estimation bayésienne (avec *a priori de Dirichlet)*

$$\widehat{\theta}_{ijk}^{MAP} = \widehat{\theta}_{\{x_i = k \mid pa_i = j\}} = \frac{\alpha_{ijk} + N_{ijk} - 1}{\alpha_{ij} + N_{ij} - r_i}$$

$$\widehat{\theta}_{ijk}^{EAP} = \widehat{\theta}_{\{x_i = k \mid pa_i = j\}} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}$$

- *A priori* important quand $N_{ijk} \to 0$: pas de cas dans la base.
- Les estimations sont consistantes et équivalentes quand $N_{ijk} \to \infty$

Apprentissage des paramètres, données complètes - peu de données

Des correctifs 'pragmatiques' ont été proposés dans le cas où peu de données rendaient l'estimation des paramètres fragiles.

Ajustement des paramètres (éviter les 0)

- **a priori de Dirichlet** $\widehat{\theta}_{ijk} \approx \frac{N_{ijk}+\alpha_{ijk}}{N_{ij}+\alpha_{ij}}$ avec $\alpha_{ij} = \sum_k \alpha_{ijk}$

PS- α_{ij} est à comparer à Nij : elle détermine l'influence a l'*a priori* sur la loi.

- **ajustement de Laplace** $\widehat{\theta}_{ijk} \approx \frac{N_{ijk}+1}{N_{ij}+|X_i|}$

PS- revient au cas précédent avec $\alpha_{ijk} = 1$: a priori uniforme, influence faible.

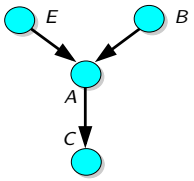
- **actualisation de Ney-Essen**

On retire à tout x une valeur fixe δ et on répartit uniformément la somme collectées.

$$D_{ij} = \sum_k \min(N_{ijk}, \delta) \qquad \text{et} \qquad \widehat{\theta}_{ijk} \approx \frac{N_{ijk} - \min(N_{ijk}, \delta) + \frac{D_{ij}}{|X_i|}}{N_{ij}}$$

Apprentissage des paramètres, données incomplètes

$$D : \begin{bmatrix} d_1^A & d_1^B & d_1^C & d_1^E \\ \cdots & \cdots & \cdots & \cdots \\ V & F & ? & V \\ V & F & ? & V \\ ? & F & ? & V \\ \cdots & \cdots & \cdots & \cdots \\ d_M^A & d_M^B & d_M^C & d_M^E \end{bmatrix}$$



$D = D^o \cup D^h$ respectivement données observées et données manquantes.

Typologie des données incomplètes
En notant $\mathcal{M}_{il} = P(d_l^i \in D^h)$

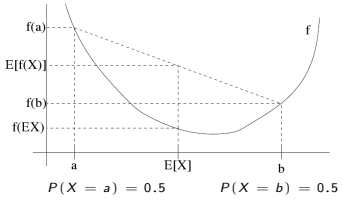
- MCAR :$P(\mathcal{M} \mid D) = P(\mathcal{M})$ (Missing Completely At Random).
- MAR : $P(\mathcal{M} \mid D) = P(\mathcal{M} \mid D^o)$ (Missing At Random).
- NMAR :$P(\mathcal{M} \mid D)$ (Not Missing At Random).

Inégalités de Jensen

Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction convexe ($\forall x, f''(x) > 0$)

Théorème (Jensen's inequality)

$$\mathbb{E}(f(X)) \geq f(\mathbb{E}(X))$$



- Si f est strictement convexe,
  $\mathbb{E}(f(X)) = f(\mathbb{E}(X)) \Rightarrow X = \mathbb{E}(X) = cst$.
- Extension sur les fonctions vectorielles.
- Forme finie du théorème :

$$\frac{\sum_i a_i \cdot f(x_i)}{\sum_i a_i} \geq f\left(\frac{\sum_i a_i \cdot x_i}{\sum_i a_i}\right)$$

- Forme finie pour les fonctions concaves (ici avec $\log$) :

$$\frac{\sum_i a_i \cdot \log(x_i)}{\sum_i a_i} \leq \log\left(\frac{\sum_i a_i \cdot x_i}{\sum_i a_i}\right)$$

Démonstration de l'inégalité de Jensen

$$f\left(\sum_{i=1}^{n}\lambda_i x_i\right) \leq \sum_{i=1}^{n}\lambda_i f(x_i)$$

- par récurrence : si $n=1$: trivial, si $n=2$: convexité
- $$\begin{aligned} f\left(\sum_{i=1}^{n+1}\lambda_i x_i\right) &= f\left(\lambda_{n+1}x_{n+1}+\sum_{i=1}^{n}\lambda_i x_i\right) \\ &= f\left(\lambda_{n+1}x_{n+1}+(1-\lambda_{n+1})\sum_{i=1}^{n}\frac{\lambda_i}{1-\lambda_{n+1}}x_i\right) \\ &\leq \lambda_{n+1}f(x_{n+1})+(1-\lambda_{n+1})f\left(\sum_{i=1}^{n}\frac{\lambda_i}{1-\lambda_{n+1}}x_i\right) \\ &\leq \lambda_{n+1}f(x_{n+1})+(1-\lambda_{n+1})\sum_{i=1}^{n}\frac{\lambda_i}{1-\lambda_{n+1}}f(x_i) \\ &= \lambda_{n+1}f(x_{n+1})+\sum_{i=1}^{n}\lambda_i f(x_i) = \sum_{i=1}^{n+1}\lambda_i f(x_i) \end{aligned}$$

Algorithme EM : approximation de la log-vraisemblance

On se place dans le cadre de la maximisation de la (log) vraisemblance.

$$LL(\Theta : D) = \sum_{m=1}^{M}\log P(d_m \mid \Theta)$$

Il nout faut faire apparaître $P(d_m^o, d_m^h)$ donc :

$$= \sum_{m=1}^{M}\log \sum_{d_m^h} P(d_m^o, d_m^h \mid \Theta)$$

Soit $Q_m(d_m^h)$ une loi de probabilités quelconque des variables d_m^h :

$$= \sum_{m=1}^{M}\log \sum_{d_m^h} Q_m(d_m^h)\cdot\frac{P(d_m^o, d_m^h \mid \Theta)}{Q_m(d_m^h)}$$

Enfin, en utilisant la concavité de $\log$ et l'inégalité de Jensen :

$$LL(\Theta : D) \geq \sum_{m=1}^{M}\sum_{d_m^h} Q_m(d_m^h)\cdot\log\frac{P(d_m^o, d_m^h \mid \Theta)}{Q_m(d_m^h)}$$

Algorithme EM

$$LL(\theta : D) \geq \sum_{m=1}^{M}\sum_{d_m^h} Q_m(d_m^h)\cdot\log\frac{P(d_m^o, d_m^h \mid \Theta)}{Q_m(d_m^h)}$$

Que choisir pour Q_m ?

Pour atteindre l'égalité dans Jensen : $\frac{P(d_m^o, d_m^h \mid \Theta)}{Q_m(d_m^h)} = c$ (constante en fonction des d_m^h)

Autrement dit :

$$Q_m(d_m^h) \propto P(d_m^o, d_m^h \mid \Theta)$$

i.e. :

$$Q_m(d_m^h) = P(d_m^h \mid d_m^o, \Theta)$$

D'où l'idée d'un algorithme itératif :

Soit Θ^0 une version initiale des paramètres
Répéter jusque convergence :

- Étape E(xpectation) :
  $Q_m^{(t+1)}(d_m^h) = P(d_m^h \mid d_m^o, \Theta^t)$
  $LL(\Theta : d_m) = \sum_{d_m^h} Q_m^{(t+1)}(d_m^h)\cdot\log\frac{P(d_m^o, d_m^h \mid \Theta)}{Q_m^{(t+1)}(d_m^h)}$
- Étape M(aximisation) :
  $\Theta^{t+1} = \arg\max_{\Theta}\sum_{m=1}^{M} LL(\Theta : d_m)$

Convergence de EM

On prouvera la monotonie de EM $LL(\Theta^{(t+1)} : D) \geq LL(\Theta^{(t)} : D)$

Démonstration :
On sait que (Jensen, puisque c'est vrai pour n'importe quel Q_m) :

$$LL(\Theta : D) \geq \sum_{m=1}^{M} \sum_{d_m^h} Q_m^{t+1}(d_m^h) \cdot \log \frac{P(d_m^o, d_m^h|\Theta)}{Q_m^{t+1}(d_m^h)}$$

En particulier, pour Θ^{t+1} :

$$LL(\Theta^{t+1} : D) \geq \sum_{m=1}^{M} \sum_{d_m^h} Q_m^{t+1}(d_m^h) \cdot \log \frac{P(d_m^o, d_m^h|\Theta^{t+1})}{Q_m^{t+1}(d_m^h)}$$

Mais Θ^{t+1} a été choisi comme $\arg\max$

$$\geq \sum_{m=1}^{M} \sum_{d_m^h} Q_m^{t+1}(d_m^h) \cdot \log \frac{P(d_m^o, d_m^h|\Theta^{t})}{Q_m^{t+1}(d_m^h)}$$

Le membre droit est exactement $LL(\Theta^t : D) = \sum_{m=1}^{M} LL(\Theta^t : D_m)$. D'où

$$LL(\Theta^{t+1} : D) \geq LL(\Theta^t : D) \quad \blacksquare$$

Deux inconvénients majeurs à cet algorithme dans le cas général :

- Sensibilité à Θ^0
- Piège dans des optimas locaux.

EM pour les BNs

EM dans les BNs

Répéter jusque convergence

Étape E : Estimer $N_{ijk}^{(t+1)}$ à partir des $P(X_i|Pa_i, \theta_{ijk}^t)$
inférence dans le BN de paramètres θ_{ijk}^t

Étape M : $\theta_{ijk}^{t+1} = \frac{N_{ijk}^{(t+1)}}{N_{ij}^{(t+1)}}$

P	S
o	?
n	?
o	n
n	n
o	o

Paramètres à estimer :
- $P(P) = [\theta_P \quad 1 - \theta_P]$
- $P(S \mid P = o) = [\theta_{S|P=o} \quad 1 - \theta_{S|P=o}]$
- $P(S \mid P = n) = [\theta_{S|P=n} \quad 1 - \theta_{S|P=o}]$

Par MLE : $\theta_P = \frac{3}{5}$

EM dans un BN : exemple

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

1 Étape E selon $\theta^{(0)}$

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

Étape M

$$\theta_{S|P=o}^{(1)} = \frac{1.3}{1.3+1.7} = 0.433 \quad \text{et} \quad \theta_{S|P=n}^{(1)} = \frac{0.4}{0.4+1.6} = 0.2$$

2 Étape E selon $\theta^{(1)}$

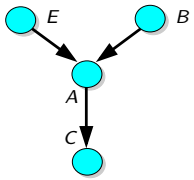
Pluie	Seine	$P(S \mid P = o)$ $S = o$	$S = n$	$P(S \mid P = n)$ $S = o$	S=n
o	?	0.433	0.567	0	0
n	?	0	0	0.2	0.8
o	n	0	1	0	0
n	n	0	0	0	1
o	o	1	0	0	0
	N^*	1.433	1.567	0.2	1.8

Étape M

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

3 etc. ($\theta_{S|P=o}^{(t)} \to 0.5 \quad$ et $\quad \theta_{S|P=n}^{(t)} \to 0$)

EM pours les BNs – cas général



	A	B	C	E
...				
1325	?	0	1	0
...				

Comment faire l'étape E ?

- Remplacer le ? par $P(A \mid B = 1, C = 1, E = 0)$
- $\Rightarrow$**inférence dans le BN avec les paramètres** Θ^t

Apprentissage de la structure, données complètes

- **But** : obtenir automatiquement une structure de réseau bayésien à partir de données.
- **En théorie** : Test du χ^2 plus énumération de tous les modèles possibles : OK
- **En pratique** : Beaucoup de problème mais avant tout :

Espace des réseaux bayésiens (Robinson, 1977)

Le nombre de structures possibles pour n nœuds est super-exponentiel.

$$NS(n) = \begin{cases} 1 & , n \leq 1 \\ \sum_{i=1}^{n} (-1)^{i+1} \cdot C_i^n \cdot 2^{i \cdot (n-i)} \cdot NS(n-1) & , n > 1 \end{cases}$$

Robinson (1977) *Counting unlabelled acyclid digraphs.* In Lecture Notes in Mathematics : Combinatorial Mathematics V

La recherche exhaustive n'est pas possible. L'espace est bien trop grand : $NS(10) \approx 4.2 \cdot 10^{18}$!

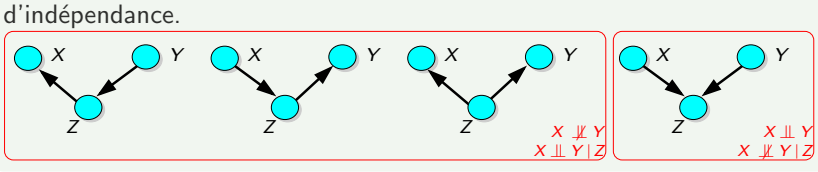
Apprentissage de structure - introduction

Tableau général de l'apprentissage

- Recherche de relation symétrique + orientation (*causalité*)
  - algorithme IC/PC
  - algorithme IC*/FCI
- Recherche heuristique (score)
  - Dans l'espace des structures (BN ou équivalent de Markov),
  - algorithmes essayant de maximiser un score (entropie, AIC, BIC, MDL,BD, BDe, BDeu, $\cdots$).

Classe d'équivalence de Markov

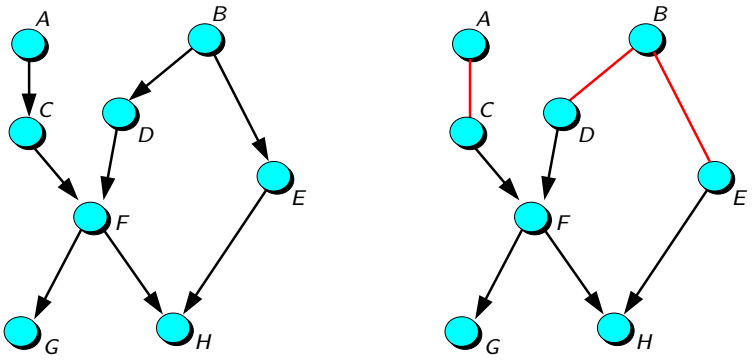
Deux réseaux bayésiens sont équivalents si ils représentent le même modèle d'indépendance.



Classe d'équivalence de Markov, graphe essentiel

Définition (classe d'équivalence de Markov, graphe essentiel)

Une classe d'équivalence de Markov est l'ensemble de réseaux bayésiens qui sont tous équivalents.
Elle peut être représentée par le graphe sans circuit partiellement orienté qui a la même structure que tous les réseaux équivalents, mais pour lequel les arcs réversibles (n'appartenant pas à des V-structures, ou dont l'inversion ne génère pas de V-structure) sont remplacés par des arêtes (non orientées) : le graphe essentiel.



Recherche de relation symétrique

En terme statistique, les relations testables sont symétriques : corrélation ou indépendance entre variables aléatoires.
Par contre, une fois des relations 2 à 2 trouvées, il s'agit de tester certaines indépendances conditionnelles (V-structure) qui forcent les orientations.

Principe de base (IC, IC*, PC, FCI)

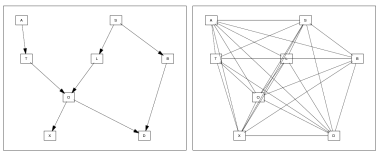
1. Construire le graphe (non orienté) des relations de dépendance trouvées statistiquement (χ^2 ou autre) :
   - Ajouter des arêtes à partir du graphe vide.
   - Retirer des arêtes à partir du graphe complet.
2. Détecter les V-structures et les orientations qu'elles impliquent.
3. Finaliser les orientations en restant dans la même classe d'équivalence de Markov.

Écueils principaux :un très grand nombre de tests d'indépendances, chaque test étant très sensible au nombre de données disponibles.

Exemple PC

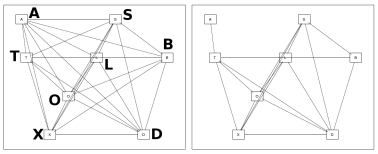
- Soit un réseau bayésien (à gauche) qui a permis de créer une base de 5000 cas. [1]

Etape 0 : Graphe non orienté reliant tous les nœuds.



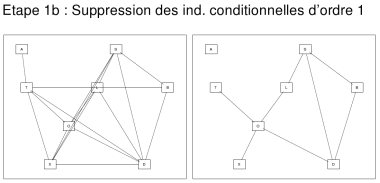
- Par des χ^2, on teste toutes les indépendances marginales ($X \perp\!\!\!\perp Y$) puis les indépendances par rapport à une variable ($X \perp\!\!\!\perp Y \mid Z$).

Etape 1a : Suppression des ind. conditionnelles d'ordre 0



On trouve : $A \perp\!\!\!\perp S$, $L \perp\!\!\!\perp A$, $B \perp\!\!\!\perp A$, $O \perp\!\!\!\perp A$, $X \perp\!\!\!\perp A$, $D \perp\!\!\!\perp A$, $T \perp\!\!\!\perp S$, $L \perp\!\!\!\perp T$,$O \perp\!\!\!\perp B$, $X \perp\!\!\!\perp B$.

Etape 1b : Suppression des ind. conditionnelles d'ordre 1



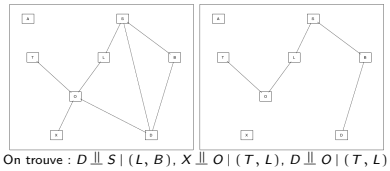
On trouve : $T \perp\!\!\!\perp A \mid O$, $O \perp\!\!\!\perp S \mid L$, $X \perp\!\!\!\perp S \mid L$, $B \perp\!\!\!\perp T \mid S$, $X \perp\!\!\!\perp T \mid O$, $D \perp\!\!\!\perp T \mid O$, $B \perp\!\!\!\perp L \mid S$, $X \perp\!\!\!\perp L \mid O$, $D \perp\!\!\!\perp L \mid O$, $D \perp\!\!\!\perp X \mid O$.

1. Exemple de Philippe Leray

Exemple PC

- On continue les χ^2 d'ordre supérieur

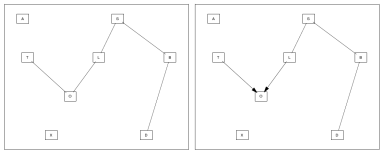
Etape 1c : Suppression des ind. conditionnelles d'ordre 2



On trouve : $D \perp\!\!\!\perp S \mid (L, B)$, $X \perp\!\!\!\perp O \mid (T, L)$, $D \perp\!\!\!\perp O \mid (T, L)$.

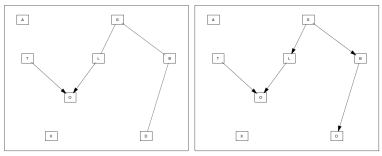
- Recherche des V-Structure, propagation des contraintes d'orientations puis orientations des dernières arêtes en restant Markov-équivalent.

Etape 2 : Recherche des V-structures



On trouve : $T \not\perp\!\!\!\perp L$ et $T \perp\!\!\!\perp L \mid O$

Etape 4 : Instanciation du PDAG



Orientation sans nouvelle V-structure

- Conclusion : avec 5000 cas, PC perd des informations sur des χ^2 faussés.

Algorithmes dirigés par une heuristique

La recherche exhaustive des relations d'indépendances est inatteignable (nombre de tests prohibitifs, quantité de données nécessaires trop importantes, etc.). Donc utilisation d'une heuristique permettant de quantifier l'adéquation d'une structure à une base de données.

Propriétés des scores

Soient D la base de donnée, T la topologie du réseau bayésien candidat et Θ ses paramètres.Pour qu'un score (une fonction calculée sur un réseau bayésien) soit considéré comme une bonne heuristique, on peut lui demander :

1. **Vraisemblance** : Coller le mieux aux données ($\max L(T, \Theta : D)$).
2. **Rasoir d'Occam** : Privilégier les topologies T simples aux topologies complexes ($\min Dim(T)$).
3. **Consistance locale** : Ajouter un arc 'utile' devrait augmenter le score. Ajouter un arc 'inutile' devrait diminuer le score.
4. **Score équivalence** : Deux réseaux bayésiens Markov-équivalents devraient avoir le même score.
5. **Décomposition locale** : Calculer la modification du score par l'ajout/retrait d'un arc ne doit pas imposer de re-calculer tout le score mais seulement une partie, locale à l'arc modifié.

Précision sur la décomposition locale

Décomposition

On dira qu'un score $Q(T, \Theta, D)$ est décomposable si $\exists \{q_i, \forall i$ nœud de $T\}$ famille de fonctions telle que

$$Q(T, \Theta, D) = \sum_i q_i(i, pa(i), D[i, pa(i)])$$

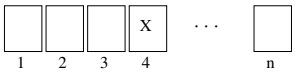
Les fonctions q_i dépendent du nœud i, des parents de i et de la partie de la base de donnée qui correspond à ces nœuds.

Il est alors clair que rajouter ou supprimer un arc revient à modifier un seul q_i : le calcul peut se faire de manière locale. Les méthodes de *recherche locales* sont alors utilisables.

Les scores utilisés pour l'apprentissage de réseau bayésien vérifient, en pratique, toutes ces propriétés.

Digression culturelle : information et entropie statistique

Soit une expérience de probabilité simple : un gain se trouve dans une des boites numérotées de 1 à n. Il y a équiprobabilité d'occurence des n positions pour le gain.



Définition (Le nombre d'information H – HARTLEY, 1928)

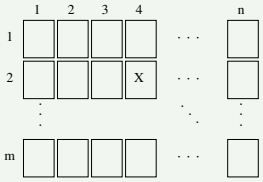
Le nombre d'information $H(n)$ est la quantité d'information reçue en apprenant où se trouve le gain. Elle est équivalente à la quantité d'incertitude *expérimentée au début de l'expérience (sans connaissance).*

$H(n)$ doit nécessairement avoir quelques propriétés.
Par exemple, $H(1) = 0$

Propriétés de H

Quelques propriétés de H

1. $H(1) = 0$
2. Arbitrairement, $H(2) = 1$
3. Monotonie : $H(n) \leq H(n+1)$ (n augmente $\Rightarrow$l'incertitude grandit.)
4. $H(n \cdot m)$?



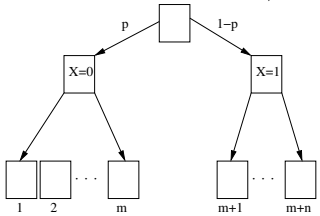
Additivité : $H(n \cdot m) = H(n) + H(m)$

Théorème

$H(n) = \log_2(n) = -\log_2(\frac{1}{n})$ vérifie ces conditions et est la seule si on considère n et m rationnels en 4.

Entropie de Shannon (1948)

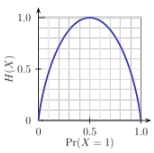
Plutôt que définir l'information apportée par le résultat d'une expérience, il s'agit de mesurer la *quantité moyenne d'information contenue dans une loi de probabilité.*
Soit une v.a. binaire X tel que $P(X = 0) = p$, p rationnel : $p = \frac{m}{m+n}$.



- Quantité d'information par la position du gain parmi $m + n$: $H(m + n) = log_2(m + n)$
- Quantité d'information par $X = 0$: $H(m + n) - H(m)$ (position du gain parmi m superflue)
- Quantité d'information par $X = 1$: $H(m + n) - H(n)$ (position du gain parmi n superflue)
- En moyenne : $p(H(m + n) - H(m)) + (1 - p)(H(m + n) - H(n)) = -p \log_2(p) - (1 - p) \log_2(1 - p)$

Définition (Entropie de Shannon)

$$h(p_1, \cdots, p_n) = -\sum_i p_i \cdot \log_2(p_i)$$



Entropie statistique expérimentale (1/3)

- X = variable aléatoire $\approx$ 1 caractère dans une phrase
- domaine de $X = \{x_1, \ldots, x_{16}\}$

Encodage «classique d'une phrase» :

- 16 valeurs possibles $\Longrightarrow$ codage sur 4 bits

  $\Longrightarrow$ phrase de 10 caractères = 40 bits

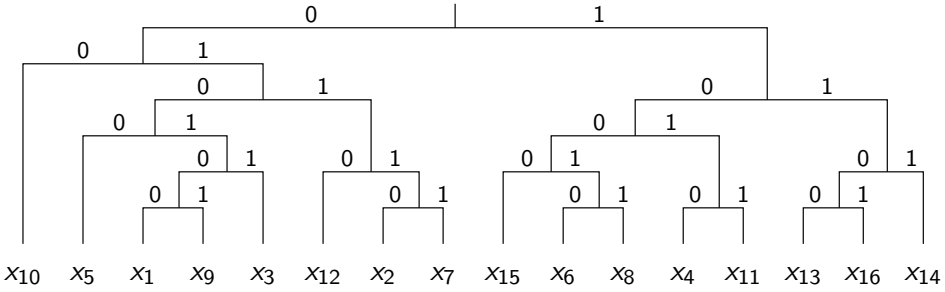
- Faire mieux : minimisation de l'espérance du nombre de bits $\Longrightarrow$ compression

Proba d'apparition des caractères ($\times 100$)															
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
1	3	2	6	5	4	2	3	1	21	6	5	9	17	7	8

Entropie statistique expérimentale (2/3)

Proba d'apparition des caractères ($\times 100$)															
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
1	3	2	6	5	4	2	3	1	21	6	5	9	17	7	8

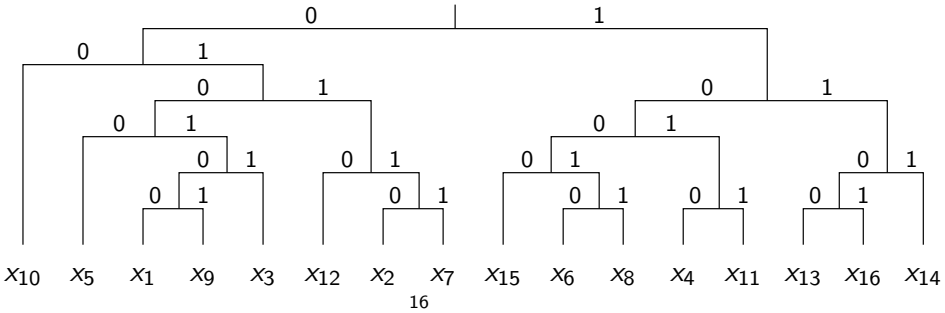
Codage de Huffman



$\Longrightarrow$ par exemple $x_3 = 01011$

Entropie statistique expérimentale (3/3)

X_i	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
nb bits	6	5	5	4	4	5	5	5	6	2	4	4	4	3	4	4
proba	1	3	2	6	5	4	2	3	1	21	6	5	9	17	7	8



Espérance du nombre de bits $L = \sum_{i=1}^{16} p_i |\text{nb bits } x_i| = 3,59 < 4$

$$H(X) \leq L \leq H(X) + 1$$

Interprétation de la divergence de Kullback-Leibler

Soient P et Q deux distribution sur le même espace X. Comment les comparer ?

Divergence de Kullback-Leibler (entropie relative)

$$D_{KL}(P\|Q) = \sum_x P(x) \log_2 \frac{P(x)}{Q(x)}$$

- $D_{KL}(P\|Q) \geq 0$
- $D_{KL}(P\|Q) = 0 \iff P = Q$
- $D_{KL}(P\|Q) \neq D_{KL}(Q\|P)$

$$D_{KL}(P\|Q) = \sum_x P(x) \log_2 P(x) - \sum_x P(x) \log_2 Q(x) = H(P,Q) - H(P)$$

Cette mesure s'interprète comme la différence moyenne du nombre de bits nécessaires au codage d'échantillons de P selon que le codage est choisi optimal pour la distribution P ou Q.

Entropie conditionnelle et dimension d'un réseau bayésien

- L'entropie statistique, due à Claude Shannon, est une fonction mathématique qui correspond à la quantité d'information contenue ou délivrée par une source d'information. Elle est telle que plus la source est redondante, moins elle contient d'information au sens de Shannon.

Plus un réseau bayésien sera complet, plus son entropie sera grande.

Entropie conditionnelle dans un réseau bayésien

$$H(T,D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} -\frac{N_{i,j,k}}{N_{i,j}} \cdot log_2 \left(\frac{N_{i,j,k}}{N_{i,j}} \right)$$

où r_i est le nombre de valeurs de X_i et $q_i = \prod_{j \in pa_i} r_j$ est le nombre de configuration des parents de X_i.

On peut prouver que $log_2 L(\Theta^{MV}, T : D) = -N \cdot H(T,D)$: **Maximiser la vraisemblance va avoir tendance à produire des réseaux bayésiens complet.**

- La dimension d'un réseau bayésien va être donnée par le nombre de paramètres nécessaires à l'instantiation de toutes les lois conditionnelles ($= |\Theta|$). En notant que pour la loi marginale d'une variable multinomiale X_i, il faut $r_i - 1$ paramètres, il est aisé de trouver que :

$$Dim(T) = \sum_i ((r_i - 1) \cdot q_i)$$

Quelques scores (1) : AIC/BIC

Idée de base : Il faut maximiser la vraisemblance tout en minimisant la dimension.

score AIC (Akaike, 70)

- Akaike Information Criterion

$$\text{Score}_{\text{AIC}}(T,D) = log_2 L(\Theta^{MV}, T : D) - Dim(T)$$

score BIC (Schwartz, 78)

- Bayesian Information Criterion

$$\text{Score}_{\text{BIC}}(T,D) = log_2 L(\Theta^{MV}, T : D) - \frac{1}{2} \cdot Dim(T) \cdot log_2 N$$

Quelques scores (2) : MDL (Rissanen,78)

MDL consiste à considérer la compacité de la représentation du modèle comme un bon critère de la qualité de ce modèle. Étonamment, ce critère est équivalent au critère BIC ci-dessus.
Il s'agit donc de minimiser la taille de la représentation, composée de :

- la représentation du modèle,
- la représentation des données sous forme de paramètres du modèle.

score MDL (Lam and Bacchus, 93)

- Minimum Description Length

$$\text{Score}_{\text{MDL}}(T, D) = log_2 L(\Theta^{\text{MV}}, T : D) - |\text{arcs}_T| \cdot log_2 N - c \cdot Dim(T)$$

où arcs_T est l'ensemble des arcs du graphe, c est le nombre de bits nécessaire à la représentation d'un paramètre.

Quelques scores (3) : BDe

Avec un critère bayésien, il s'agirait simplement de maximiser la probabilité jointe de T et D :

$$\begin{aligned} P(T, D) &= \int_{\Theta} P(D \mid \Theta, T) \cdot P(\Theta \mid T) \cdot P(T) d\Theta \\ &= P(T) \cdot \int_{\Theta} L(\Theta, T : D) \cdot P(\Theta \mid T) d\Theta \end{aligned}$$

Avec des hypothèses d'indépendances, un *a priori* de Dirichlet bien choisi, on obtient :

score BDe

- Bayesian Dirichlet score Equivalent

$$\text{Score}_{\text{BDe}}(T, D) = P(T) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{i,j})}{\Gamma(N_{i,j} + \alpha_{i,j})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{i,j,k} + \alpha_{i,j,k})}{\Gamma(\alpha_{i,j,k})}$$

Recherche locale à base de scores

L'algorithme de recherche locale est un algorithme générique qui ne demande que quelques hypothèses de base :

Recherche locale

- Soit un espace de recherche,
- Soit une notion de voisinage définie par des opérations élémentaires (les voisins d'un élément sont les points atteignables par l'application d'une opération élémentaire à cet élément).
- Soit un score (heurisitique) calculable localement.
- La recherche locale est alors une séquence de voisins tels qu'à partir du point initial, tout élément ultérieur de la séquence augmente le score. (*Greedy Search*).

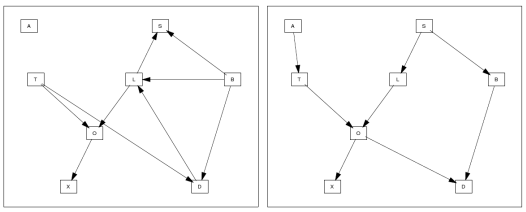
Recherche locale dans les réseaux bayésiens

- L'espace est l'espace des réseaux bayésiens (énorme)
- Le score est l'un des scores précédénts
- Soit une structure initiale
- Les opérations de base : ajout/suppression/modification d'un arc (dans le domaine de validité)

Recherche locale : Greedy Search

Algorithme implémentant exactement ce qui est défini précédemment.

Réseau obtenu vs. théorique



L'algorithme peut être bloqué sur des 'plateaux' et/ou converger vers des minima locaux.

Solutions

Principalement des méthodes de méta-heurisitiques :

- Random restart
- TABU-search (liste des K dernières structures à éviter)
- Simulated annealing (accepter des structures diminuant le score avec un seuil diminuant au cours du temps)

Recherche locale : Diminution de l'espace de recherche

S'il existe un ordre dans les nœuds, tel qu'il ne soit pas possible d'avoir des arcs rétrogrades, alors il y a diminution de la taille de l'espace de recherche.

Taille de l'espace de recherche avec ordre sur les nœuds

$$NS'(n) = 2^{\frac{n\cdot(n-1)}{2}}$$

Algorithme K2

- Réseau initial sans arcs
- Opération élémentaire : ajouter un arc de j à $i > j$.
- Greedy algorithm sur le score BD(e).
- Limite sur le nombre de parents maximum.

Problème principal : algorithme très dépendant de l'ordre.

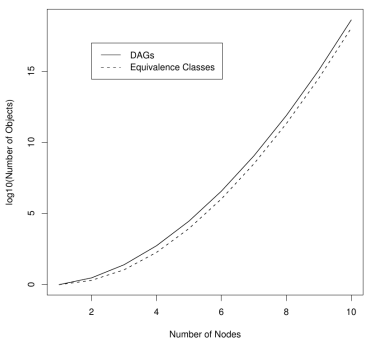
Réduction de l'espace de recherche : équivalents de Markov

Comme la recherche locale peut tomber dans des minima locaux, l'idée est de s'affranchir d'une partie de ces minima en changeant d'espace pour l'espace des classes d'équivalence de Markov.

Greedy Equivalence Search

L'espace des classes d'équivalences (notés les graphes essentiels) a une structure. On peut définir des opérateurs élémentaires et donc mener une recherche locale.

- Avantage : Pas de plage de score équivalence.
- Pas avantage : La taille de l'espace de recherche est sensiblement la même (ratio asymptotique de 3.7).



Réduction de l'espace de recherche : recherche dans les arbres

Cette recherche se limite aux BNs dans lesquels chaque nœud a au plus un parent.
Malgré une simplification (trop) grande du modèle, les arbres apportent :

- une solution élégante mathématiquement (optimisation globale),
- un nombre de paramètres minimum (minimise le risque de sur-apprentissage).

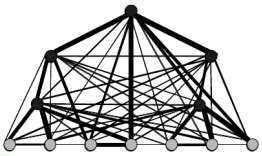
La décomposabilité du score donne :

$$LL(T) = \sum_i LL_i(i, pa(i)) = \sum_{X \to Y} LL(Y \leftarrow X) + K$$

avec $LL(Y \leftarrow X) = LL_Y(Y, X) - LL_Y(Y, \emptyset)$

Recherche de l'arbre de score maximal

- $\forall X, Y$, calculer $LL(Y \leftarrow X)$
- Trouver l'arbre (ou la forêt) de poids maximal.
  Max Spanning Tree Algorithm – $O(n^2 \cdot log(n))$



Limites de la gourmandise – Studeny, 2005

Soit un réseau bayésien G sur n variables aléatoires **binaires**.

- On note i une variable aléatoire, r_i le nombre de ses modalités, p_i le nombre de ses parents, j une configuration possible de ses p_i nœuds parents.
  Montrer que q_i le nombre de configurations des nœuds parents d'un nœud de G est forcément une puissance de 2. Quelle forme prend $Dim(G)$?
- Soit D une base de données complète sur ces n variables aléatoires **binaires**. On note N le nombre de cas total de la base, N_{ij} le nombre de cas où les parents de i ont la configuration j et N_{ijk} le nombre de cas où les parents de i ont la configuration j et i a la valeur k.
  Donner la formule générale du score BIC en fonction des paramètres N, N_{ij}, N_{ijk}, q_i et r_i.
  Dans un second temps, simplifier au maximum cette formule pour le cas de ce réseau bayésien et de ces n variables **binaires**.
- Soit la base de donnée suivante :

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

  Calculer le score BIC du graphe G^0 sans aucun arc.
  Calculer le score BIC du graphe G' où il n'existe qu'un arc. Le choix de cet arc est-il important ?
  Calculer le score BIC du graphe G^* : $A \to B \leftarrow C$.
- On propose, comme algorithme d'apprentissage de la strucutre, un algorithme gourmand qui part de G^0 et qui, utilisant le score BIC, améliore sa structure incrémentalement par ajout d'arcs successifs. Que pouvez-vous dire de cet algorithme dans le cas de notre réseau bayésien et de notre base de données ?
  Qu'en concluez-vous ?

References I

Robert G. Cowell, A. Philip Dawid, Steffen L. Lauritzen, and David J. Spiegelhalter, *Probabilistic networks and expert systems*, Springer-Verlag, New York, NY, 1999.

Finn V. Jensen and Thomas D. Nielsen, *Bayesian networks and decision graphs, second edition*, Springer, New York, NY, 2007.

D. Koller and N. Friedman, *Probabilistic Graphical Models : Principles and Techniques*, MIT Press, 2009.

P. Naïm, P.H. Wuillemin, P. Leray, O. Pourret, and A. Becker, *Réseaux bayésiens*, Eyrolles, 2011.

J. Pearl, *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*, Morgan Kaufmann, 1988.