CES – Data Scientist

Réseaux bayésiens, modèles graphiques

Pierre-Henri Wuillemin

LIP6 – Université Paris 6, France

Présentation des Modèles Graphiques dans MADI

1. Modèle graphique probabiliste : les réseaux Bayésiens
2. Inférence exacte dans les BNs
3. Inférence approchée dans les BNs
4. Apprentissages dans les BNs

Plan du cours 1

modèles graphiques pour les probabilités

1. Rappels sur les probabilités
2. Modèles décomposables, modèles factorisables
3. Modèles d'indépendance, modèles graphiques
4. Définition des réseaux bayésiens
5. Exemple de BNs et quelques applications
6. Utilisations et généralisations

Rapides rappels : indépendances (conditionnelles)

Soit X, Y, Z trois variables aléatoires (ou groupes de variables)

➥ si $P(Y) > 0$, $P(X|Y) = \dfrac{P(X,Y)}{P(Y)}$

$$\Longrightarrow P(X,Y) = P(X|Y)P(Y)$$

$$\begin{aligned}\Longrightarrow P(X,Y,Z) &= P(X|Y,Z)P(Y,Z) \\ &= P(X|Y,Z)P(Y|Z)P(Z)\end{aligned}$$

Indépendance marginale

$$\begin{aligned}X \perp\!\!\!\perp Y \text{ si et seulement si } & P(X,Y) = P(X)P(Y) \\ \text{si et seulement si } & P(X \mid Y) = P(X)\end{aligned}$$

Indépendance conditionnelle

$$\begin{aligned}X \perp\!\!\!\perp Y \,|\, Z \text{ si et seulement si } & P(X,Y \mid Z) = P(X \mid Z)P(Y \mid Z) \\ \text{si et seulement si } & P(X \mid Y,Z) = P(X \mid Z)\end{aligned}$$

Modèle probabiliste complexe

La représentation probabiliste d'un système est caractérisé par un univers Ω où chaque $\omega \in \Omega$ est un état du système.
Exemple : Un dé est caractérisé par $\Omega_{\texttt{dé}} = \{1, 2, 3, 4, 5, 6\}$.

Un système complexe est caractérisé par un univers Ω de grande taille.
Exemple : $\Omega_{\texttt{voiture}}$?

Définition (Modèle décomposable)

Un modèle probabiliste (sur Ω)est décomposable lorsqu'il existe une famille $\mathfrak{X} = (X_i)_{i<n}$ de variables aléatoires sur Ω telle que chaque $\omega \in \Omega$ est caractérisé de manière unique par les valeurs $(X_i(\omega))_{i<n}$.

Exemple : $\omega_{\texttt{voiture}} =$ `(Vitesse=55, Phare=Eteint, Pneu_gauche=dégonflé,...).`

Modèle probabiliste (2)

Modèle probabiliste complexe

Dans un modèle décomposable, une probabilité sur Ω sera donc représentée par une loi jointe des variables de $\mathfrak{X}$.

$$\forall \omega \in \Omega, p(\omega) = p\left(X_1 = X_1(\omega), X_2 = X_2(\omega), \cdots, X_n = X_n(\omega)\right)$$

Explosion combinatoire : Si toutes les variables sont binaires, un système factorisé en n variables nécessitent $\approx 2^n$ valeurs !

La factorisation peut-elle permettre d'améliorer la compacité ? Grâce à l'indépendance conditionnelle ! !

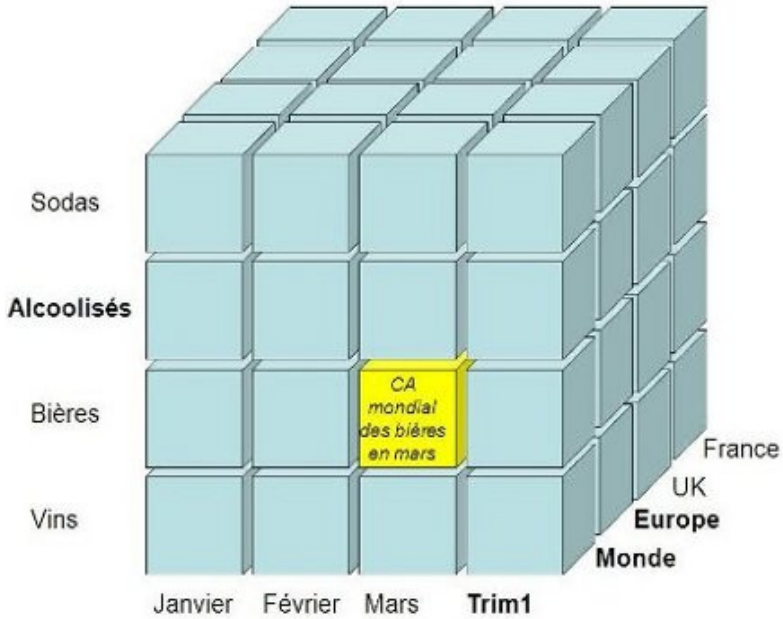
2^3 $\quad p(X, Y, Z) = p(X) \cdot p(Y \mid X) \cdot p(Z \mid X, Y)$ $\quad 2 + 2^2 + 2^3$

Avec $X \perp\!\!\!\perp Y$ et $Z \perp\!\!\!\perp X, Y$:

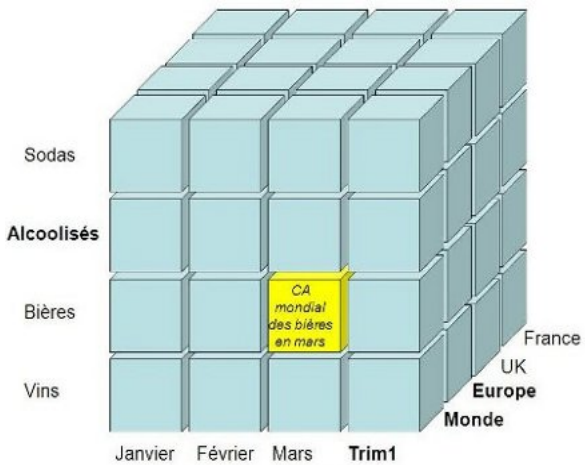
2^3 $\quad p(X, Y, Z) = p(X) \cdot p(Y) \cdot p(Z)$ $\quad 2 + 2 + 2$

Modèles complexes factorisable

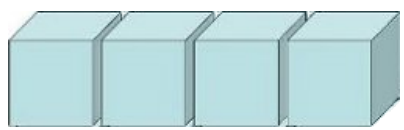
		Janvier	Février	Mars	Trim1
France	Bières	70	70	80	220
	Vins	100	110	90	300
	Total	**170**	**180**	**170**	**520**
UK	Bières	250	220	240	710
	Vins	50	40	60	150
	Total	**300**	**260**	**300**	**860**
Total Europe		***470***	***440***	***470***	***1380***



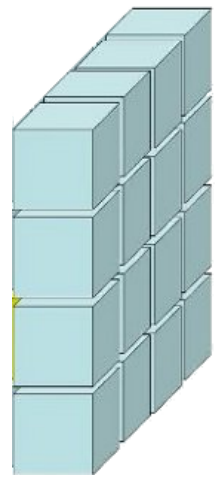
Comment voir dans ce modèle que $Mois \perp\!\!\!\perp \{Pays, Boisson\} | ?$



$=$



$*$



Modèle d'indépendances

Soit notre loi jointe $p(X_1, \cdots, X_n)$ (hyper-cube). Il existe des indépendances conditionnelles testables (χ^2) dans cette loi. Comment les représenter aisément ?
Il serait intéressant de fournir un outil basé sur les variables aléatoires du modèle, qui permettrait de manipuler les indépendances conditionnelles de manière plus naturelle qu'indirectement dans l'hyper-cube de la loi jointe :
Quel objet manipule-t-on lorsqu'on parle de l'**ensemble** des indépendances condionnelles de $p(\,)$?.

Définition (Séparabilité)

Soit $\mathfrak{I} \subset \mathcal{P}(\mathfrak{X}) \times \mathcal{P}(\mathfrak{X}) \times \mathcal{P}(\mathfrak{X})$. On nomme $\mathfrak{I}$ un modèle d'indépendance.
$\forall U, V, W \subset \mathfrak{X}$, on dit que U et V sont séparés par W ($\ll U \diamond V \mid W \gg_{\mathfrak{I}}$) si et seulement si $(U, V, W) \in \mathfrak{I}$.

Relation entre $\mathfrak{I}$ et p

L'ensemble $\mathfrak{I}_p = \{(U, V, W) \in \subset \mathcal{P}(\mathfrak{X}) \times \mathcal{P}(\mathfrak{X}) \times \mathcal{P}(\mathfrak{X}), U \perp\!\!\!\perp V \mid W\}$ est un modèle d'indépendance.

$$U \perp\!\!\!\perp V \mid W \Longleftrightarrow \ll U \diamond V \mid W \gg_{\mathfrak{I}_p}$$

et ... réciproquement ? ...

Semi-graphoïde et graphoïde - définitions

Définition (semi-graphoïde)

Un modèle d'indépendance $\mathfrak{I}$ est un semi-graphoïde s'il satisfait $\forall A, B, S, P \subset \mathfrak{X}$:

1. *Indépendance triviale* $\quad \ll A \diamond \emptyset \mid S \gg_{\mathfrak{I}}$
2. *Symétrie* $\quad \ll A \diamond B \mid S \gg_{\mathfrak{I}} \Rightarrow \ll B \diamond A \mid S \gg_{\mathfrak{I}}$
3. *Décomposition* $\quad \ll A \diamond (B \cup P) \mid S \gg_{\mathfrak{I}} \Rightarrow \ll A \diamond B \mid S \gg_{\mathfrak{I}}$
4. *Union faible* $\quad \ll A \diamond (B \cup P) \mid S \gg_{\mathfrak{I}} \Rightarrow \ll A \diamond B \mid (S \cup P) \gg_{\mathfrak{I}}$
5. *Contraction* $\quad \left\{ \begin{array}{c} \ll A \diamond B \mid (S \cup P) \gg_{\mathfrak{I}} \\ \ll A \diamond P \mid S \gg_{\mathfrak{I}} \end{array} \right\} \Rightarrow \ll A \diamond (B \cup P) \mid S \gg_{\mathfrak{I}}$

Définition (graphoïde)

Un modèle d'indépendance $\mathfrak{I}$ est un graphoïde s'il satisfait $\forall A, B, S, P \subset \mathfrak{X}$:

$\mathfrak{I}$ *est un semi-graphoïde*

6. *Intersection* $\quad \left\{ \begin{array}{c} \ll A \diamond B \mid (S \cup P) \gg_{\mathfrak{I}} \\ \ll A \diamond P \mid (S \cup B) \gg_{\mathfrak{I}} \end{array} \right\} \Rightarrow \ll A \diamond (B \cup P) \mid S \gg_{\mathfrak{I}}$

Semi-graphoïde et graphoïde - représentation des axiomes

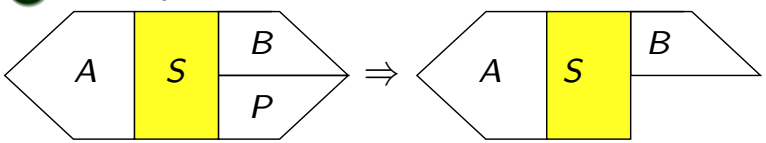
3. Décomposition $\quad \ll A \diamond (B \cup P) \mid S \gg_{\mathfrak{I}} \Rightarrow \ll A \diamond B \mid S \gg_{\mathfrak{I}}$

4. Union faible $\quad \ll A \diamond (B \cup P) \mid S \gg_{\mathfrak{I}} \Rightarrow \ll A \diamond B \mid (S \cup P) \gg_{\mathfrak{I}}$

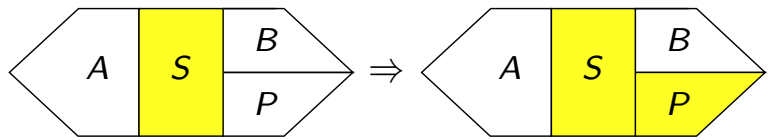
5. Contraction $\quad \left\{ \begin{array}{c} \ll A \diamond B \mid (S \cup P) \gg_{\mathfrak{I}} \\ \ll A \diamond P \mid S \gg_{\mathfrak{I}} \end{array} \right\} \Rightarrow \ll A \diamond (B \cup P) \mid S \gg_{\mathfrak{I}}$

6. Intersection $\quad \left\{ \begin{array}{c} \ll A \diamond B \mid (S \cup P) \gg_{\mathfrak{I}} \\ \ll A \diamond P \mid (S \cup B) \gg_{\mathfrak{I}} \end{array} \right\} \Rightarrow \ll A \diamond (B \cup P) \mid S \gg_{\mathfrak{I}}$

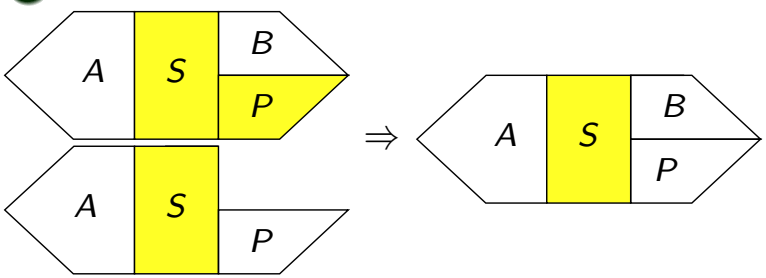
3 Décomposition



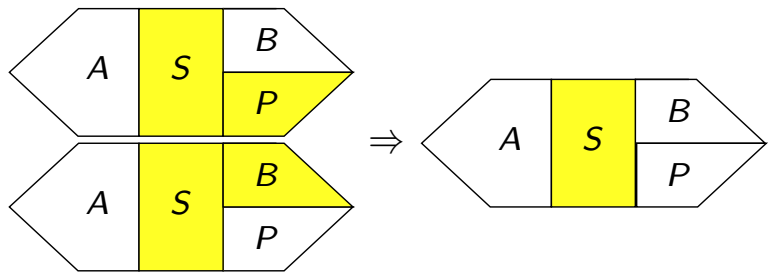
4 Union faible



5 Contraction



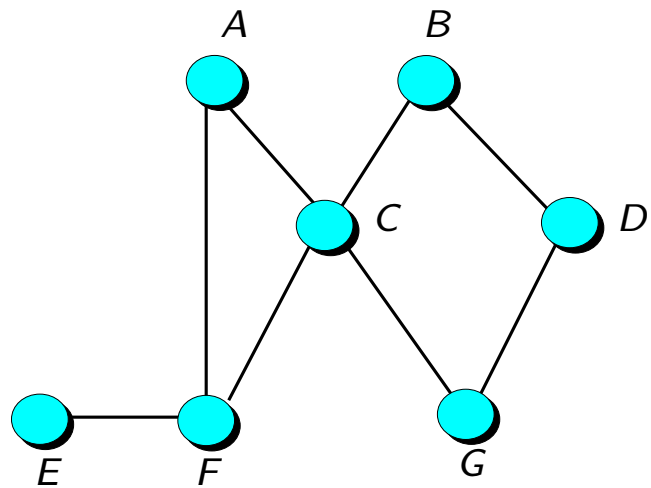
6 Intersection



Semi-graphoïde et graphoïde - Utilisation

Théorème (loi de probabilité et graphoïde)

$\mathfrak{I}_p$ possède une structure de semi-graphoïde.
Si $p > 0$ alors $\mathfrak{I}_p$ possède une structure de graphoïde.



Théorème (graphe non orienté et graphoïde)

Soit un graphe $G = (\mathfrak{X}, \mathfrak{E})$,
$\forall U, V, W \subset X, \langle U \mid W \mid V \rangle_G$ indique que toute chaîne d'un nœud de U vers un nœud de V contient forcément un nœud de W.
Alors $\{\langle U \mid W \mid V \rangle_G, U, V, W \subset X\}$ possède une structure de graphoïde.

Et bien voilà ! ! ! On va utiliser un graphe pour représenter les indépendances condionnelles de p ! ! !

Modèle graphique

Définition (Modèle graphique)

Un modèle graphique est un modèle probabiliste factorisé qui se sert d'un graphe entre les variables aléatoires pour représenter des indépendances conditionnelles.

Est-ce que ça se passe bien ? Peut-on toujours avoir $(X \perp\!\!\!\perp Y \mid Z)_p \Leftrightarrow \langle X \mid Z \mid Y \rangle_G$?

Définition (I-map, D-map, P-map, graphe-isomorphisme)

soit $G = (\mathfrak{X}, \mathfrak{E})$ un graphe et une loi de probabilité p.

G est une $D_{ependency}$-map de p ssi $(X \perp\!\!\!\perp Y \mid Z)_p \Rightarrow \langle X \mid Z \mid Y \rangle_G$.

G est une $I_{ndependency}$-map de p ssi $(X \perp\!\!\!\perp Y \mid Z)_p \Leftarrow \langle X \mid Z \mid Y \rangle_G$.

G est une P_{erfect}-map de p ssi $(X \perp\!\!\!\perp Y \mid Z)_p \Leftrightarrow \langle X \mid Z \mid Y \rangle_G$.

Un loi de probabilité p est dite graphe-isomorphe si et seulement s'il existe un graphe G qui soit une P-map de p.

- Le graphe vide, sans arc est une **D-map** de toute distribution p.
- Le graphe complet est une **I-map** de toute distribution p.

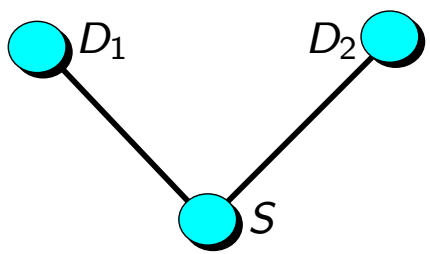
Modèle graphique non orienté : exemple 1

exemple 1

Soit un modèle probabiliste du système composé de 3 variables aléatoires : le tirage de deux dé D_1, D_2 et $S = D_1 + D_2$ qui est la somme des tirages des 2 dés.

indépendances et dépendances du modèle de l'exemple 1

- $D_1 \not\perp\!\!\!\perp S$ et $D_2 \not\perp\!\!\!\perp S$
- $D_1 \perp\!\!\!\perp D_2$ mais $D_1 \not\perp\!\!\!\perp D_2 | S$



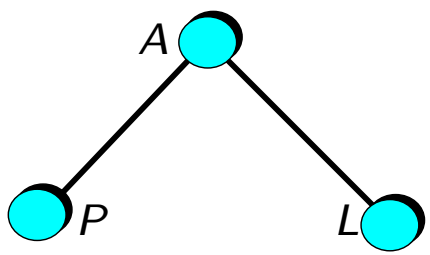
Modèle graphique non orienté : exemple 2

exemple 2

Dans un sondage, on s'aperçoit qu'il y a une forte corrélation entre l'aptitude à lire d'un individu et sa pointure...
On s'aperçoit rapidement que l'âge de l'individu est la variable qui explique cette corrélation bizarre.

indépendances et dépendances du modèle de l'exemple 2

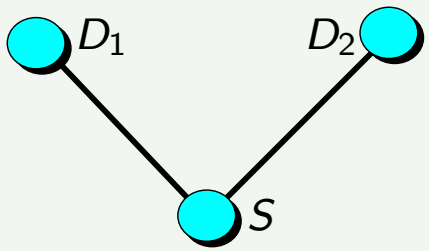
- $L \not\perp\!\!\!\perp A$ et $P \not\perp\!\!\!\perp A$
- $L \not\perp\!\!\!\perp P$ mais $L \perp\!\!\!\perp P \,|\, A$



Modèle graphique orienté

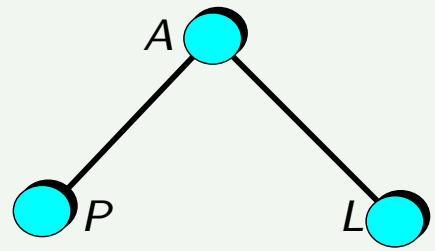
exemple 1

$D_1 \perp\!\!\!\perp D_2$ mais $D_1 \not\perp\!\!\!\perp D_2 \,|\, S$



exemple 2

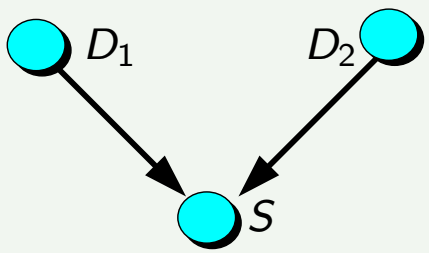
$L \not\perp\!\!\!\perp P$ mais $L \perp\!\!\!\perp P \,|\, A$



Lever l'ambiguïté en ajoutant de l'information qualitative sur les arcs : l'orientation.

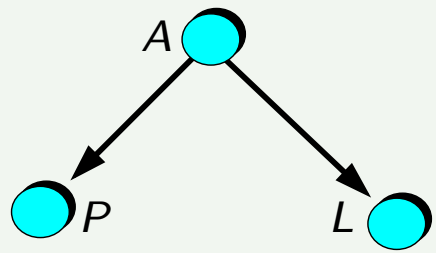
exemple 1 orienté

$D_1 \perp\!\!\!\perp D_2$ mais $D_1 \not\perp\!\!\!\perp D_2 \,|\, S$



exemple 2 orienté

$L \not\perp\!\!\!\perp P$ mais $L \perp\!\!\!\perp P \,|\, A$



Reste à donner un critère de séparation sur les graphes orientés : la d-séparation.

Modèle graphique orienté et d-séparation

Soit une chaîne $C = (x_i)_{i \in I}$ dans un graphe orienté $\overrightarrow{G}$. On dira que x_i est un puits de la chaîne C (ou C-puits) s'il est du type : $x_{i-1} \longrightarrow x_i \longleftarrow x_{i+1}$.

Définition (Chaîne active, bloquée)

Soit une chaîne $C = (x_i)_{i \in I}$ dans $\overrightarrow{G}$ et Z un sous-ensemble de nœuds de $\overrightarrow{G}$. C est une chaîne active par rapport à Z si :

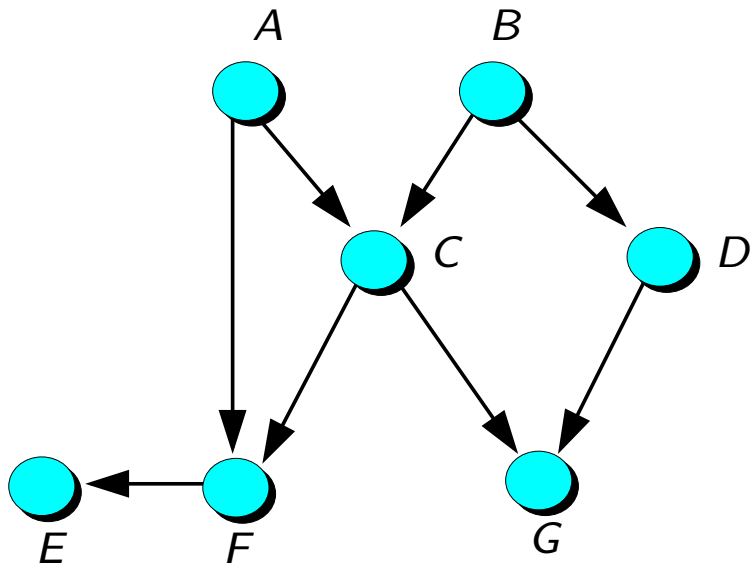
- *Tout C-puits a l'un de ses descendants ou lui-même dans Z.*
- *Aucun élément de C qui n'y est pas un C-puits n'appartient à Z.*

Une chaîne non active par rapport à Z est dite bloquée par Z.

Définition (d-séparation)

Soit $\overrightarrow{G} = (\mathfrak{X}, \mathfrak{E})$ un graphe orienté,
$\forall (X, Y, Z) \subset \mathfrak{X}$, X est d-séparé de Y par Z dans $\overrightarrow{G}$ ($\langle X \mid Z \mid Y \rangle_{\overrightarrow{G}}$) si et seulement si toute chaîne d'un élément de X vers un élément de Y est bloquée par Z.

Exemple de d-séparation



Réseau bayésien et propriétés de Markov

Définition (réseau bayésien)

Soit un graphe $\overrightarrow{G}$, muni de la d-séparation. Si $\overrightarrow{G}$ est I-map d'une loi p alors $\overrightarrow{G}$ est un réseau bayésien pour p.

Définition (Propriété de Markov globale)

$\overrightarrow{G}$ *vérifie la PMG pour* $p \Leftrightarrow \forall A, B, S \subset \mathfrak{X}$,

$$\langle A \,|\, S \,|\, B \rangle_{\overrightarrow{G}} \Rightarrow A \perp\!\!\!\perp B \,|\, S.$$

i.e. $\overrightarrow{G}$ est une I-map pour p.

Définition (Propriété de Markov locale)

$\overrightarrow{G}$ *vérifie la PML pour* $p \Leftrightarrow \forall x \in \mathfrak{X}$,

$$\{x\} \perp\!\!\!\perp \mathrm{nd}(x) \,|\, \Pi_x.$$

où $\mathrm{nd}(x)$ représente les nœuds non descendants de x et Π_x ses parents.

Réseau bayésien et propriétés de Markov (2)

Théorème

$$PMG \iff PML$$

Un graphe est un réseau bayésien pour p si et seulement si chaque nœud est indépendant de ses non-descendants, conditionnellement à ses parents pour p.

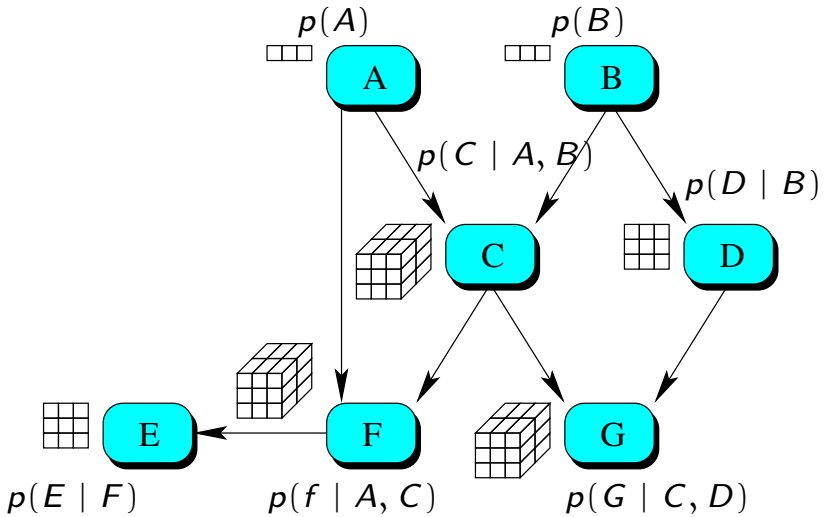
(quand la loi est positive)

Théorème (Factorisation récursive)

Soit $\overrightarrow{G} = (\mathfrak{X}, \mathfrak{E})$ *un réseau bayésien pour p, alors :*

$$p(\mathfrak{X}) = \prod_{X \in \mathfrak{X}} p\left(X \mid \Pi_X\right)$$

réseau bayésien : exemple et définition



$$p(A, B, C, D, E, F, G,) = ?$$
$$p(A) \cdot p(B)$$
$$\cdot p(C \mid A, B) \cdot p(D \mid B) \cdot p(F \mid A, C)$$
$$\cdot p(E \mid F) \cdot p(G \mid C, D)$$

Tout se passe comme si l'information était localisée dans les nœuds !

$P(A, B, C, D, E, F, G)$: $3^7 = 2187$ paramètres vs 105 paramètres dans le BN !

Définition (Réseau bayésien (BN))

Un réseau bayésien est une représentation compacte d'une distribution de probabilité sur un ensemble de variables aléatoires. Il s'appuie sur un graphe orienté sans circuit (DAG) pour représenter son modèle d'indépendance. La décomposition de la loi jointe suivant le graphe s'écrit :

$$P(\mathfrak{X}) = \prod_i P(X_i \mid \Pi_i)$$

1er exemple de construction d'un RB (1/6)

Exemple de la dyspnée (Lauritzen & Spiegelhalter (88))

La dyspnée peut être engendrée par une tuberculose, un cancer des poumons, une bronchite, par plusieurs de ces maladies, ou bien par aucune.
Un séjour récent en Asie augmente les chances de tuberculose, tandis que fumer augmente les risques de cancer des poumons. Des rayons X permettent de détecter une tuberculose ou un cancer.
Un patient éprouve des difficultés à respirer. Dans quelle mesure peut-on dire qu'il est atteint de dyspnée ?

Variables aléatoires :

- D : dyspnée : oui/non
- C : cancer : oui/non
- A : Asie : oui/non
- R : rayons X : positif/négatif
- T : tuberculose : oui/non
- B : bronchite : oui/non
- F : fumer : oui/non

1er exemple de construction d'un RB (2/6)

Exemple de la dyspnée (Lauritzen & Spiegelhalter (88))

La dyspnée peut être engendrée par une tuberculose, un cancer des poumons, une bronchite, par plusieurs de ces maladies, ou bien par aucune. Un séjour récent en Asie augmente les chances de tuberculose, tandis que fumer augmente les risques de cancer des poumons. Des rayons X permettent de détecter une tuberculose ou un cancer. Un patient éprouve des difficultés à respirer. Dans quelle mesure peut-on dire qu'il est atteint de dyspnée ?

$$P(D, R, T, C, B, A, F) = P(D|R, T, C, B, A, F) \times P(R, T, C, B, A, F)$$

Or $P(D|R, T, C, B, A, F) = P(D|T, C, B)$

$$\Longrightarrow P(D, R, T, C, B, A, F) = P(D|T, C, B) \times P(R, T, C, B, A, F)$$

1er exemple de construction d'un RB (3/6)

Exemple de la dyspnée (Lauritzen & Spiegelhalter (88))

La dyspnée peut être engendrée par une tuberculose, un cancer des poumons, une bronchite, par plusieurs de ces maladies, ou bien par aucune. Un séjour récent en Asie augmente les chances de tuberculose, tandis que fumer augmente les risques de cancer des poumons. Des rayons X permettent de détecter une tuberculose ou un cancer. Un patient éprouve des difficultés à respirer. Dans quelle mesure peut-on dire qu'il est atteint de dyspnée ?

$$P(D, R, T, C, B, A, F) = P(D|T, C, B) \times P(R, T, C, B, A, F)$$

or $P(R, T, C, B, A, F) = P(R|T, C, B, A, F) \times P(T, C, B, A, F)$

et $P(R|T, C, B, A, F) = P(R|T, C)$

$$\Longrightarrow P(D, R, T, C, B, A, F) = P(D|T, C, B) \times P(R|T, C) \times P(T, C, B, A, F)$$

1er exemple de construction d'un RB (4/6)

Exemple de la dyspnée (Lauritzen & Spiegelhalter (88))

La dyspnée peut être engendrée par une tuberculose, un cancer des poumons, une bronchite, par plusieurs de ces maladies, ou bien par aucune. Un séjour récent en Asie augmente les chances de tuberculose, tandis que fumer augmente les risques de cancer des poumons. Des rayons X permettent de détecter une tuberculose ou un cancer. Un patient éprouve des difficultés à respirer. Dans quelle mesure peut-on dire qu'il est atteint de dyspnée ?

$$P(D,R,T,C,B,A,F) = P(D|T,C,B) \times P(R|T,C) \times P(T,C,B,A,F)$$

$$P(T|C,B,A,F) = P(T|A)$$

$$P(D,R,T,C,B,A,F) = P(D|T,C,B) \times P(R|T,C) \times P(T|A) \times P(C,B,A,F)$$

.....

$$= P(D|T,C,B) \times P(R|T,C) \times P(T|A) \times P(C|F) \times P(B|F) \times P(A) \times P(F)$$

1er exemple de construction d'un RB (5/6)

$$P(D,R,T,C,B,A,F) = P(D|T,C,B) \times P(R|T,C) \times P(T|A) \times P(C|F) \times P(B|F) \times P(A) \times P(F)$$

Si toutes les variables ont 10 valeurs possibles :

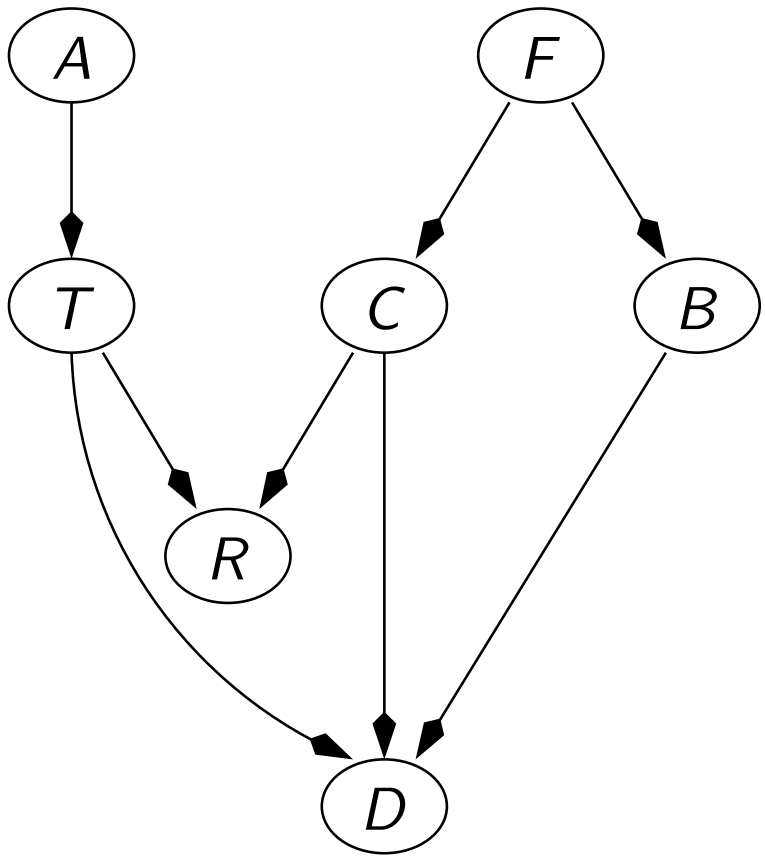
$P(D,R,T,C,B,A,F)$ nécessite une table de 10^7 éléments

formule décomposée nécessite :

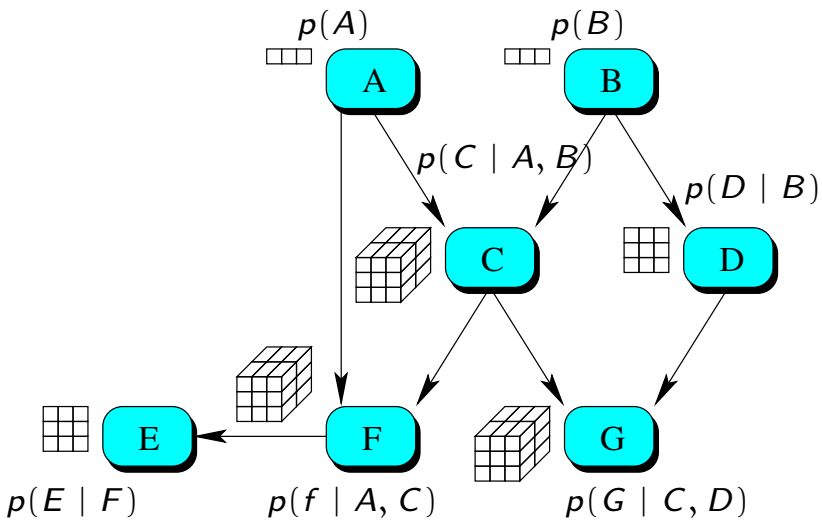
$10000 + 1000 + 3 \times 100 + 2 \times 10 = 11320$ éléments

1er exemple de construction d'un RB (6/6)

$$P(D,R,T,C,B,A,F) =$$
$$P(D|T,C,B) \times P(R|T,C) \times P(T|A) \times P(C|F) \times P(B|F) \times P(A) \times P(F)$$



Utilisations des BNs : inférence probabiliste



diagnostic : $P(A \mid F)$

- *diagnostic de panne*
- *sûreté de fonctionnement*
- *filtrage de spams*

prédiction $P(E \mid B, A)$

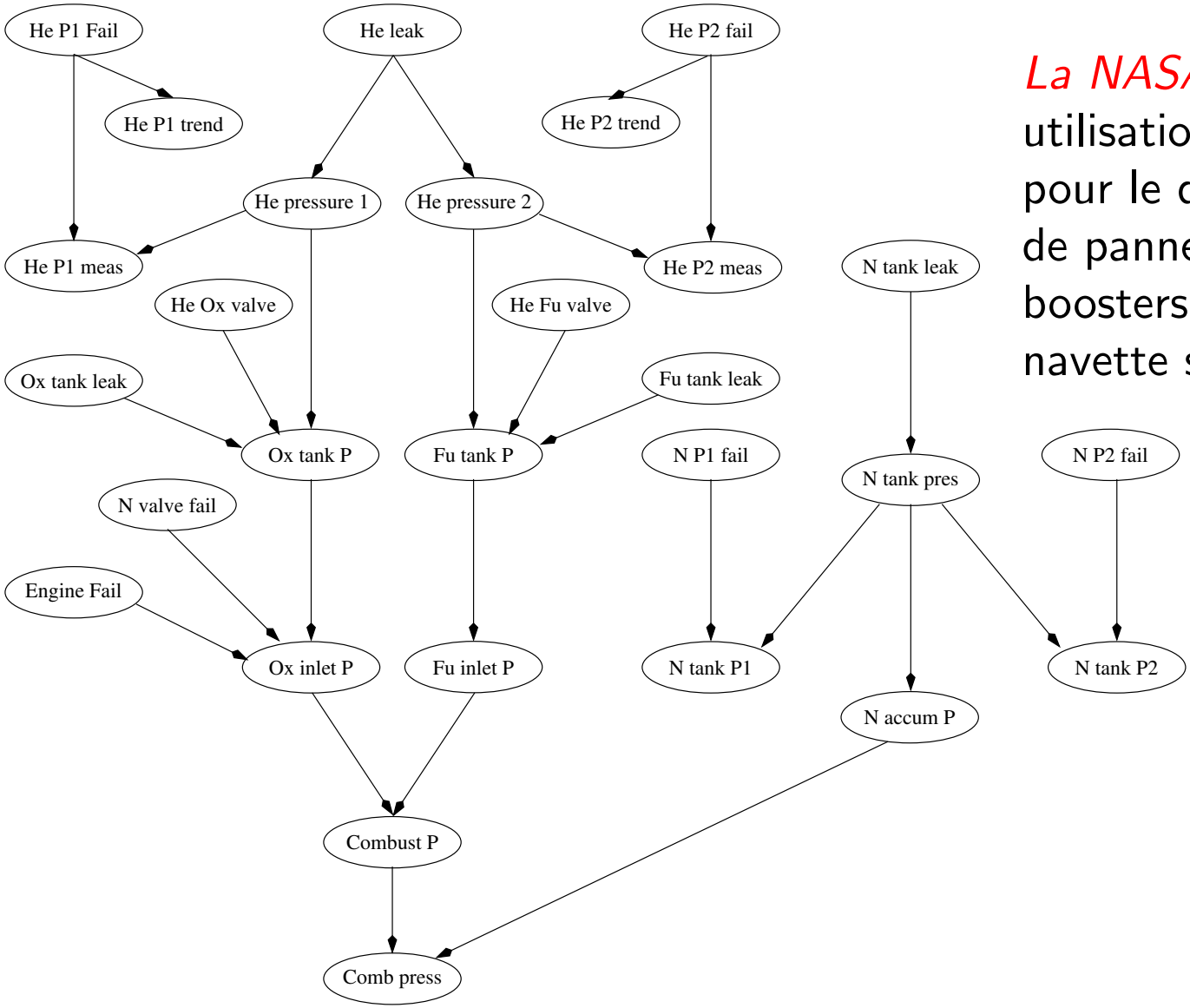
- *Simulation de process (industriels)*
- *prévisions boursières*
- *modélisation de joueurs*

Autres tâches

- *Cas le plus probable :* $\arg\max P(\mathfrak{X} \mid D)$
- *Analyse de sensibilité, information mutuelle, etc.*
- *Troubleshooting :* $\arg\max \frac{P(.)}{C(.)}$

Application 1 : Diagnostic de panne

Diagnostic de panne à la NASA

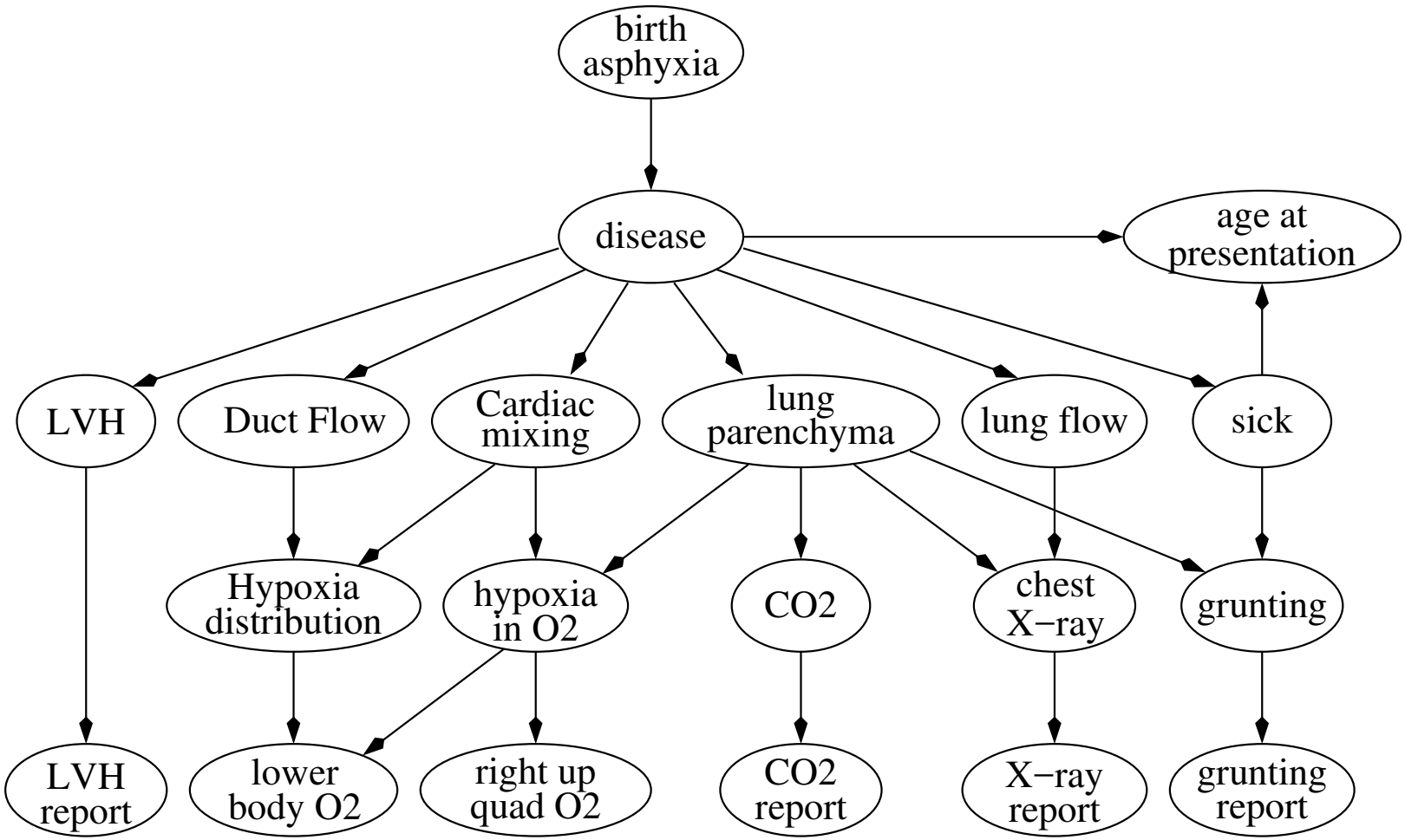


La NASA :
utilisation d'un RB pour le diagnostic de pannes des boosters de la navette spatiale

Application 2 : Diagnostic médical

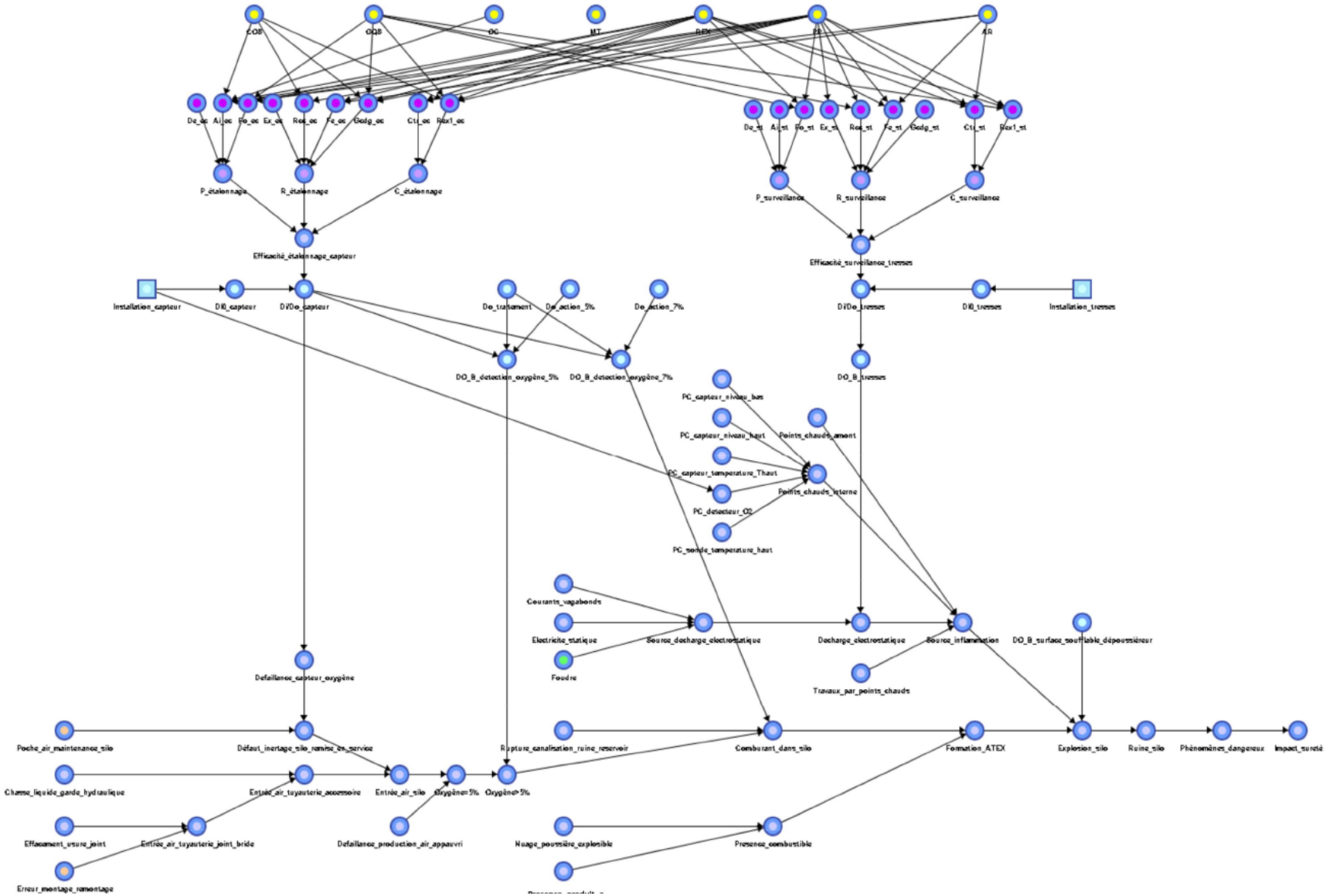
Le Great Ormond Street hospital for sick children

Aide au diagnostic des causes d'une cyanose ou d'une crise cardiaque chez le nourrisson juste après sa naissance.



Application 3 : analyse du risque

Modélisation des phénomèes de risque par réseau bayésien : **approche modulaire**.



Application 4 : classification bayésienne

Soient deux v.a. X (de dimension d) discrète et Y (de dimension 1) discrète (*pas forcément binaire*).
Sur une base d'apprentissage (supervisé) Π_a, on peut estimer les probabilités par des fréquences pour $P(X, Y)$.

Classification

Pour une instantiation x de X, on cherche à prédire sa classe (valeur de Y) : $\widehat{y}$.

1 Maximum de vraisemblance (ML)

$$\widehat{y} = \arg\max_{y_i} P(x \mid y_i)$$

2 Maximum a posteriori (MAP)

$$\widehat{y} = \arg\max_{y_i} P(y_i \mid x) = \arg\max_{y_i} P(y_i) \cdot P(x \mid y_i)$$

D'après la règle de Bayes, $P(Y \mid X) \propto P(X \mid Y) \cdot P(Y)$, on comprend que l'intérêt du MAP est de prendre en compte un *a priori* sur la fréquence de chaque classe.

Il peut être difficile d'obtenir ces distributions.
Particulièrement : $P(X \mid Y)$ peut demander beaucoup d'observation ! !

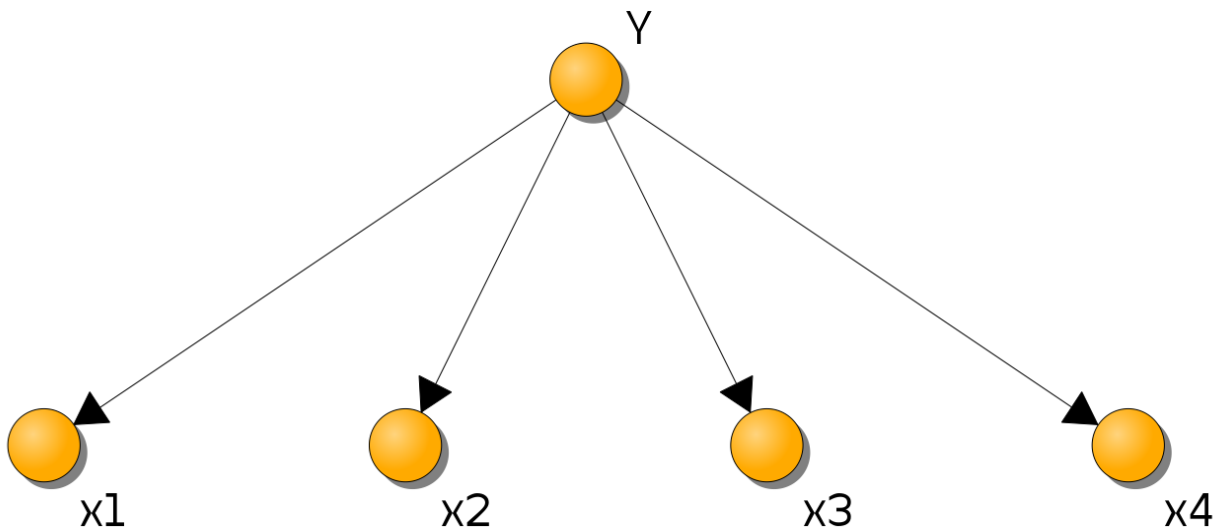
classification bayésienne (2) : Naif Bayes

Comment calculer $P(X \mid Y)$?

Classifieur bayésien naïf

$$\forall k \neq l, X^k \perp\!\!\!\perp X^l \mid Y \qquad \text{et} \qquad P(x, y) = P(y) \cdot \prod_{k=1}^{d} P(x^k \mid y)$$

Cette hypothèse est très forte. Elle a peu de chance de s'avouer exacte dans un cas réel. Néanmoins cette approximation donne des résultats souvent satisfaisants.

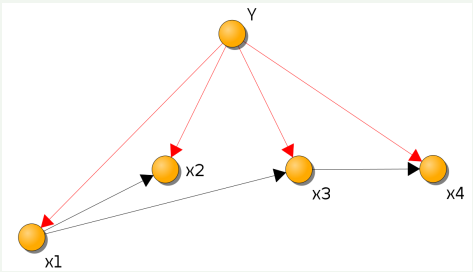


- Estimation des paramètres : trivial (si Π_a sans valeurs manquantes)
- ML : $\prod_{k=1}^{d} P(x^k \mid y)$ ...
- MAP : $P(y \mid x_1, \cdots, x_d)$: inférence dans le BN !

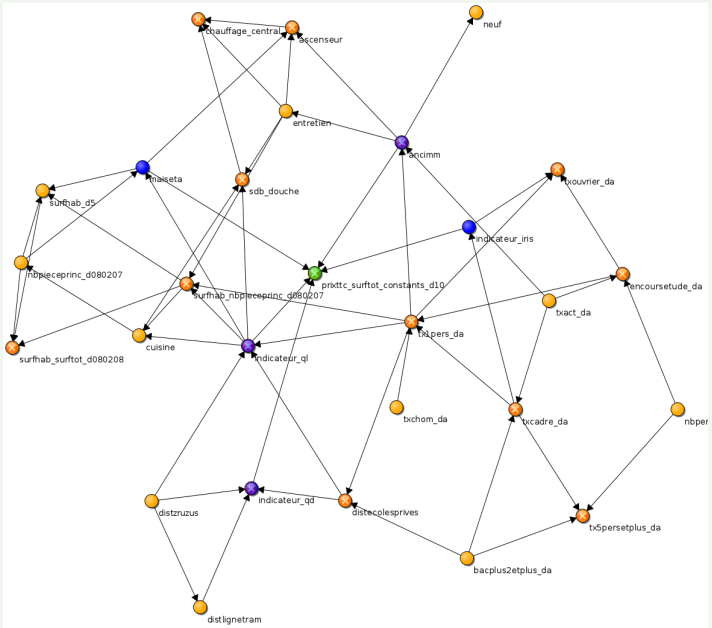
Classification bayésienne (3) : modèles plus complexes

Tree-Augmented Naive Models

Toute variable X_i peut avoir un parent autre que Y (mais un seul !).



Réseau bayésien complet



Dans un BN composé de Y et (X_i), calculer $P(Y \mid X_1, \cdots, X_n)$.
Note : on n'a pas besoin de tous les X_i : Markov Blanket $MB(.)$).

$$P(Y \mid X) = P(Y \mid MB(Y))$$

Application 5 : modèles séquentiels

données séquentielles

Un ensemble de données séquentielles est un ensemble de données dont la séquence est porteur de sens.

- $\{PressionPneuGauche, PressionPneuDroit, NiveauBatterie\}$ est un ensemble de variables dont une instantiation n'est pas un ensemble de données séquentielles.
- $\{Euro_{1999}, Euro_{2000}, Euro_{2001}, Euro_{2002}\}$ est un ensemble de variables dont une instantiation est un ensemble de données séquentielles.

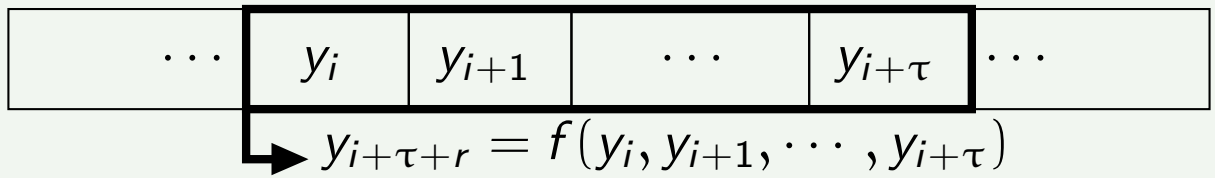
Un grand nombre d'applications peuvent se présenter sous la forme d'un modèle de données séquentielles :

- Données temporelles
  - Reconnaissance de la parole
  - Données sismiques
  - Données financières
  - ...
- Données générées par un processus mono-dimensionnel
  - Bio-séquences
  - ...

Prédiction de données séquentielles

Approches classiques

Principe : Prédire directement en fonction d'une fenêtre de valeurs.

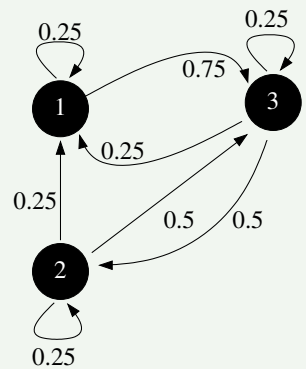


- Modèles linéaires : ARIMA (*auto-regressive integrated moving average*), ARMAX (*auto-regressive moving average exogenous variables*), etc.
- Modèles non linéaires : réseaux de neurones, arbres de décisions, etc.

inconvénients : Fenêtre limitée , peu adaptable (connaissance a priori ?) , mauvais comportement quand Y est multidimensionnel.

Modèle direct probabiliste : chaîne de Markov

- Une variable d'état discrète (X^n) (à l'instant n).
- Paramètres du modèle :
  - Condition initiale : $P(X^O)$
  - Modèle de transition : $P(X^n \mid X^{n-1})$

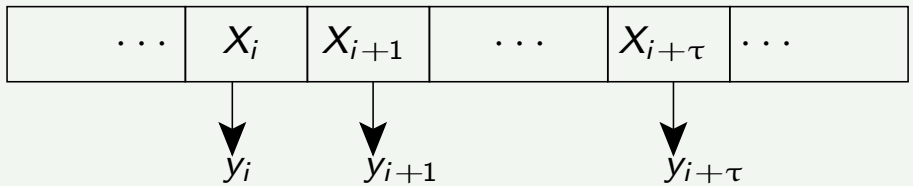


Dans le cas des CdM : fenêtre de taille 1

Représentation par espace d'états

Modèles à espace d'états (*state-space models*)

Principe : Les données temporelles (y_t) sont générées par un système dont l'état X_t évolue dans le temps et est non observé.



Une représentation par espace d'état implique une représentation d'une connaissance incertaine : l'état du système. Cette connaissance incertaine sera **probabilisée** ici.

On notera X_i ou X_t la (ou les) variable(s) aléatoire(s) représentant l'espace d'état.
On notera Y_i ou Y_t la (ou les) variable(s) aléatoire(s) représentant les observations.

Modèles connus à espace d'état

- HMM : Modèle de Markov caché
- KFM : filtre de Kalman
- dBN : réseaux bayésiens dynamiques

On peut voir les CdM, les HMMs et les KFMs comme des cas particulier de dBN. Voir plus loin.

Représentation par espace d'états (2)

Principes des modèles à espace d'états

- Les variables d'état forment une chaîne de Markov :
  - $P(X_1)$: les conditions initiales
  - $P(X_{t+1} \mid X_t)$ les probabilité de transition d'état.
- Les variables d'observations dépendent des variables d'état :
  - $P(Y_t \mid X_t)$ fonctions (probabilistes) d'observation.
- Le modèle doit vérifier certaines hypothèses :
  - Propriété de Markov sur X_t : $P(X_{t+1} \mid X_1, \cdots, X_t) = P(X_{t+1} \mid X_t)$
  - Propriété de Markov conditionnelle sur Y_t : $P(Y_t \mid X_t, Y_{t-1}) = P(Y_t \mid X_t)$
  - Invariance séquentielle (homogénéité) : $P(X_{t+1} \mid X_t)$ et $P(Y_t \mid X_t)$ ne dépendent pas de t.

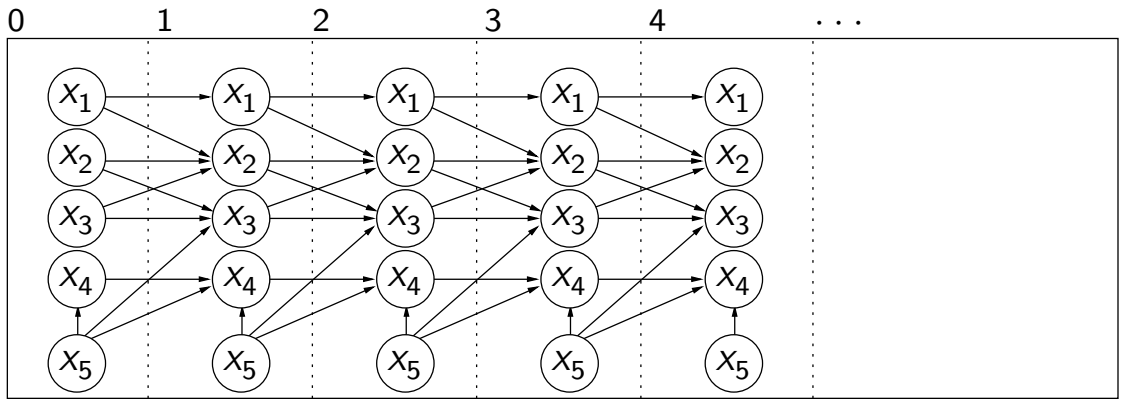
- HMM : X_t est un vecteur de variables aléatoires discrètes.
- KFM : X_t est un vecteur de variables aléatoires continues.
- dBN : Généralisation du modèle.

Les réseaux bayésiens dynamiques

dBN (dynamic BN)

Un réseau bayésien dynamique est un réseau bayésien dont les variables sont indicées par le temps t et par i : $X^{(t)} = X_1^{(t)}, \cdots, X_N^{(t)}$ et dont la distribution vérifie certaines propriétés :

- Markov ordre 1 :
  $P(X^{(t)} \mid X^{(0)}, \cdots, X^{(t-1)}) = P(X^{(t)} \mid X^{(t-1)})$,
- Homogénéité :
  $P(X^{(t)} \mid X^{(t-1)}) = \cdots = P(X^{(1)} \mid X^{(0)})$.



- L'adjectif *dynamique* n'est pas forcément bien choisi (puisqu'il y a homogénéité). *Temporel* ou *séquentiel* eût été de meilleur goût.
- Formellement, un réseau bayésien dynamique peut être considéré comme virtuellement infini.
- La définition ci-dessus est celle d'un dBN du premier ordre (t ne dépend que de $t-1$). On pourrait, bien évidemment, définir des dBNs d'ordre supérieur.
- On remarque que d'après la définition, les arcs d'un dBN vont de $X^{(t-1)}$ à X^t ou restent dans le même X^t (le même *timeslice*).

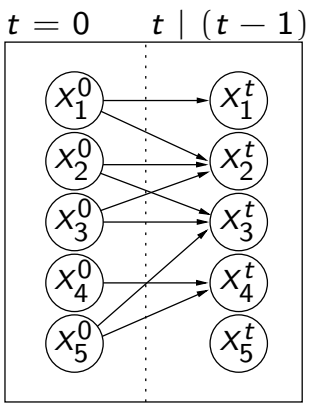
Les réseaux bayésiens dynamiques (2-TBN)

2-TBN

Un réseau bayésien dynamique est défini

- par les conditions initiales ($P(X^{(0)})$)
- par les relations entre des variables à l'instant $t-1$ et ces même variables à l'instant t (*timeslice*).

Cette représentation, appelée 2TBN (2 timeslice BN) permet de modéliser un BN virtuellemet infini qui en est le développement dans le temps, à partir d'un instant 0.

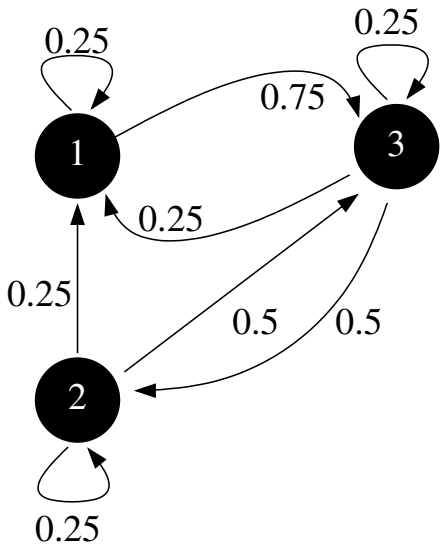


$P(x_1^{(t)}, \ldots, x_5^{(t)} \mid x_1^{(t-1)}, \ldots, x_5^{(t-1)})$

1024 contre 4+16+16+8+2=46 ! !

$$= P(x_1^{(t)} \mid x^{(t-1)}) \\ P(x_2^{(t)} \mid x_1^{(t-1)}, x_2^{(t-1)}, x_3^{(t-1)}) \\ P(x_3^{(t)} \mid x_2^{(t-1)}, x_3^{(t-1)}, x_4^{(t-1)}) \\ P(x_4^{(t)} \mid x_4^{(t-1)}, x_5^{(t-1)}) \\ P(x_5^{(t)})$$

Chaîne de Markov et réseaux bayésiens dynamiques

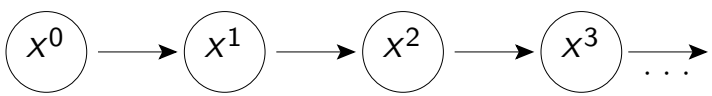


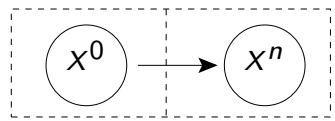
$$P(X^n \mid X^{n-1}) = \begin{pmatrix} 0.25 & 0 & 0.75 \\ 0.25 & 0.25 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{pmatrix}$$

Chaîne de Markov

- Une variable d'état discrète (X^n) (à l'instant n).
- Paramètres du modèle :
  - Condition initiale : $P(X^O)$
  - Modèle de transition : $P(X^n \mid X^{n-1})$

Réseau bayésien dynamique équivalent :

dBN : 

2TBN : 

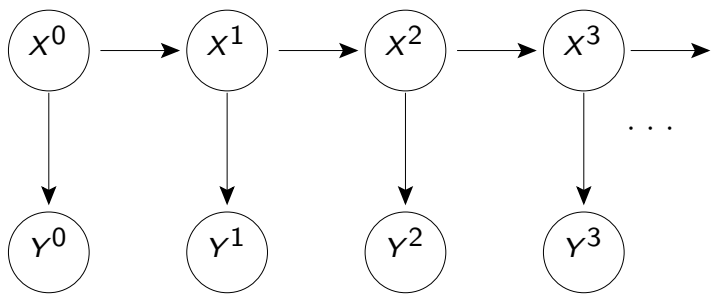
HMM et réseaux bayésiens dynamiques

HMM simple

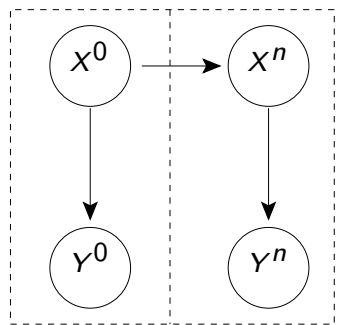
- Une variable d'état discrète (X^n) (à l'instant n).
- Une variable d'observation discrète (Y^n)
- Paramètres du modèle :
  - Condition initiale : $P(X^O)$
  - Modèle de transition : $P(X^n \mid X^{n-1})$
  - Modèle d'observation : $P(Y^n \mid X^n)$

Ce qui donne, modélisé comme un réseau bayésien dynamique :

dBN :



2TBN :



Inférences dans les réseaux bayésiens dynamiques

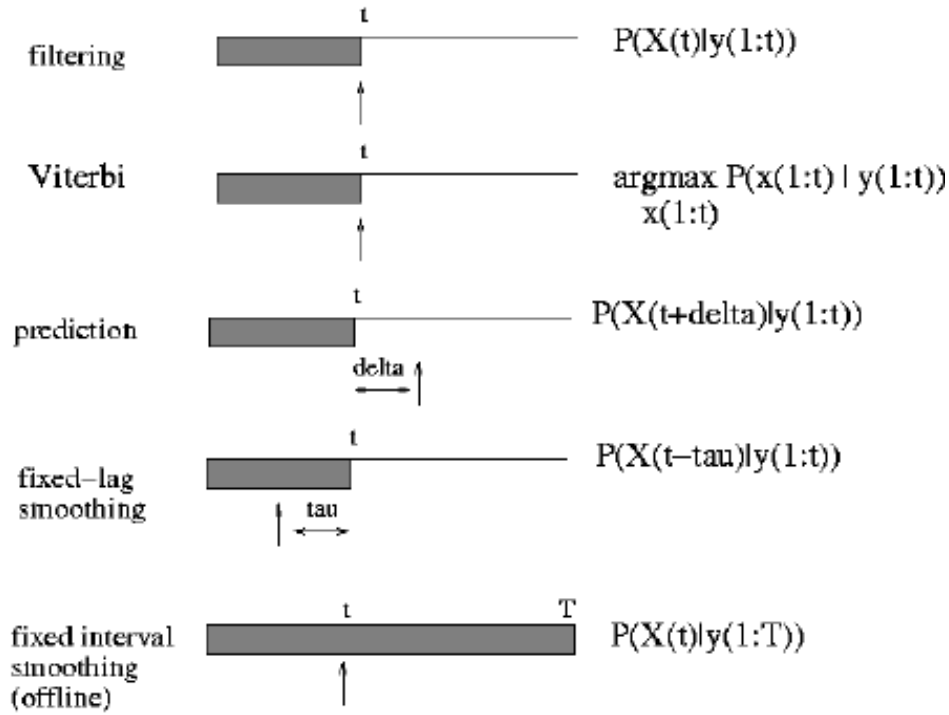
- A priori, très complexe : nombre de nœuds importants
- A priori, très complexe : “causes” communes dans un passé (lointain)

Complexité

NP-difficile

$P(X_t^i \mid y_{[1:r]})$?

- $r = t$: *Filtering*
- $r > t$: *Smoothing*
- $r < t$: *Prediction*
- MPE : *Viterbi*



rappels rapide : MDP

- Temps : t
- État : $S_t \in \mathcal{S}$, état à l'instant t ;
- Action : $a_t \in \mathcal{A}$, action à l'instant t ;
- Récompense : $r_t \in \mathbb{R}$, récompense à l'instant t ;
- Politique : π est une fonction de $\mathcal{S}$ dans $\mathcal{A}$:à chaque état, on associe une action à effectuer ($\pi(s_t) = a_t$).

Dynamique d'un processus markovien

Le système représenté évolue dans le temps, en fonctions de la dynamique propre du système et de la séquence de décisions de l'agent.
La dynamique est non-déterminisite et donc représentée par une probabilité de transition d'état à état, dépendant de l'action effectuée.

$$p(s_{t+1} \mid s_t, a_t) = p_{a_t}(s_{t+1} \mid s_t)$$

Trouver π^* : Value Iteration (VI) ou Policy Iteration (PI)

Rappels de l'algorithme Policy Iteration

Algorithme 6: Algorithme d'itération de la politique modifié - Critère γ-pondéré

initialiser $V_0 \in \mathcal{V}$ tel que $LV_0 \geq V_0$
$flag \leftarrow 0$
$n \leftarrow 0$
répéter
  pour $s \in S$ **faire**
    $\pi_{n+1}(s) \in \text{argmax}_{a \in A} \left\{ r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) V_n(s') \right\}$
    $(\pi_{n+1}(s) = \pi_n(s)$ si possible$)$
    $V_n^0(s) = \max_{a \in A} \left\{ r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) V_n(s') \right\}$
  $m \leftarrow 0$
  si $\| V_n^0 - V_n \| < \epsilon$ **alors** $flag \leftarrow 1$
  sinon
    répéter
      pour $s \in S$ **faire**
        $V_n^{m+1}(s) = r(s, \pi_{n+1}(s)) + \gamma \sum_{s' \in S} p(s' \mid s, \pi_{n+1}(s)) V_n^m(s')$
      $m \leftarrow m + 1$
    jusqu'à $\| V_n^{m+1} - V_n^m \| < \delta$
    $V_{n+1} \leftarrow V_n^m$
    $n \leftarrow n + 1$
jusqu'à $flag = 1$
retourner V_n, π_{n+1}

(origine : Frédérick Garcia, INRA, 2001)

Problèmes de calculs des probabilités de transition

Algorithme 6: Algorithme d'itération de la politique modifié - Critère γ-pondéré

initialiser $V_0 \in \mathcal{V}$ tel que $LV_0 \geq V_0$
$flag \leftarrow 0$
$n \leftarrow 0$
répéter
  pour $s \in S$ **faire**
    $\pi_{n+1}(s) \in \text{argmax}_{a \in A} \{r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) V_n(s')\}$
    $(\pi_{n+1}(s) = \pi_n(s)$ si possible$)$
    $V_n^0(s) = \max_{a \in A} \{r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) V_n(s')\}$
  $m \leftarrow 0$
  si $\| V_n^0 - V_n \| < \epsilon$ **alors** $flag \leftarrow 1$
  sinon
    répéter
      pour $s \in S$ **faire**
        $V_n^{m+1}(s) = r(s, \pi_{n+1}(s)) + \gamma \sum_{s' \in S} p(s' \mid s, \pi_{n+1}(s)) V_n^m(s')$
      $m \leftarrow m + 1$
    jusqu'à $\| V_n^{m+1} - V_n^m \| < \delta$
    $V_{n+1} \leftarrow V_n^m$
    $n \leftarrow n + 1$
jusqu'à $flag = 1$
retourner V_n, π_{n+1}

Si l'état est factorisé

- Comment calculer $P(s_{t+1} \mid s_t, a)$?
  Dans un espace factorisé, on peut noter $s = (x_1, \ldots, x_n)$.
  On utilise le $'$ pour indiquer le futur

- Comment calculer $P(x'_1, \ldots, x'_n \mid x_1, \ldots, x_n, a)$?
  En supposant chaque variable binaire, pour chaque action possible,
  Hyper-matrice à $2^{(2n)}$ paramètres ! ! Explosion combinatoire ! !

Curse of dimensionality (Bellman 1961)

Curse of dimensionality refers to the exponential growth of hypervolume as a function of dimensionality.

- Une solution pour limiter les dégâts :

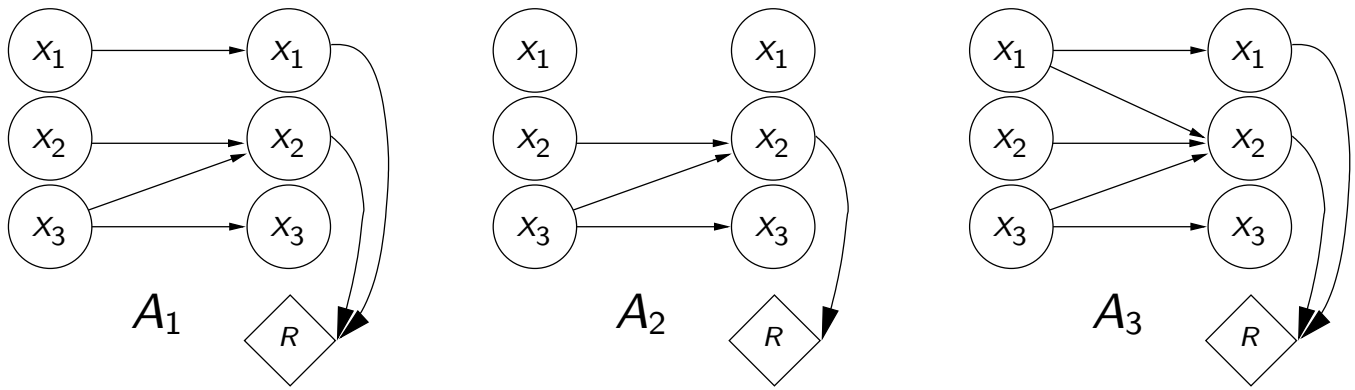
les réseaux bayésiens dynamiques

les FMDPs

Factored MDPs

Les FMDPs sont des MDPs dont l'espace d'état est factorisé. Cette factorisation permet de représenter les transitions comme des DBNs.

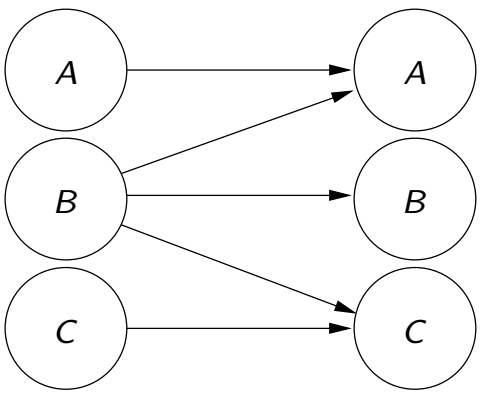
- La structure du DBN dépendant de l'action choisie.
- La récompense est une fonction de l'action et de l'ensemble des variables composants l'état. Mais se simplifie souvent (ne dépend que de certains des variables de l'état).

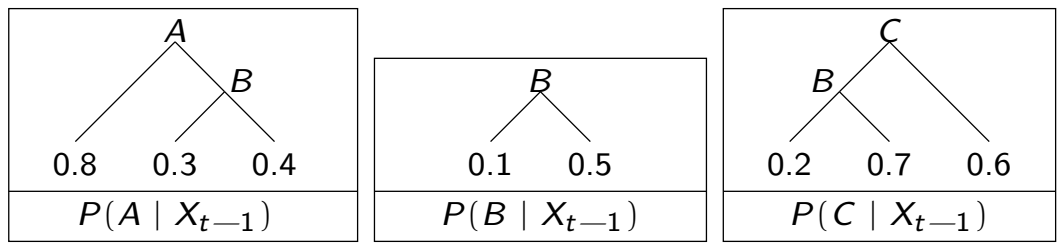


- Pour notre problème, l'inférence est facile : on connaît **complètement** $t-1$ et on veut t.

Améliorer la représentation ?

Peut on encore compacter la représentation ? OUI !
En utilisant les “symétries contextuelles”, c’est-à-dire en utilisant une représentation des CPTs sous la forme d’arbre.





Il faut alors mettre à jour les algorithmes pour travailler sur les arbres (cf. SVI et SPI) :

SVI, SPI

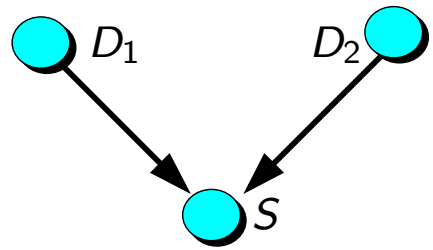
Boutilier, C., Dearden, R., & Goldszmidt, M. **(2000)**. *Stochastic Dynamic Programming with Factored Representations*. Artificial Intelligence, 121.

Annexe : Autres modèles que les BNs ?

Modèle d'indépendance

- $D_1 \perp\!\!\!\perp D_2$

(donc $D_1 \not\perp\!\!\!\perp D_2 | S$)

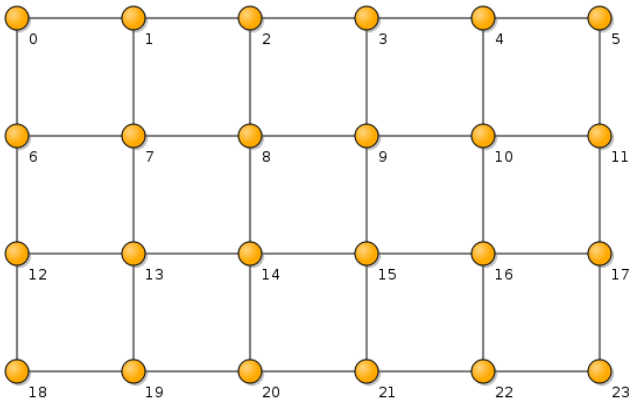


Modèle d'indépendance

- $A \perp\!\!\!\perp B | \{C, D\}$
- $C \perp\!\!\!\perp D | \{A, B\}$

?

Réseaux de Markov



- Séparation dans les réseaux de Markov : séparation dans les graphes non-orientés.
- Propriété de Markov locale dans les réseaux de Markov :
  $X \perp\!\!\!\perp \text{non-voisin}(X) \,|\, \text{voisin}(X)$

Définition (Réseau de Markov)

Un réseau de Markov est une représentation compacte d'une distribution de probabilité sur un ensemble de variables aléatoires. Il s'appuie sur un graphe non-orienté pour représenter son modèle d'indépendance.
La décomposition de la loi jointe suivant le graphe s'écrit :

$$P(\mathfrak{X}) = \frac{1}{Z} \cdot \prod_{C \in clique(G)} \Phi(C)$$

CES – Data Scientist

Inférences probabilistes dans les réseaux bayésiens

Pierre-Henri WUILLEMIN

DESIR
LIP6
pierre-henri.wuillemin@lip6.fr

Une base de données

Soit une base de données présentée sous la forme d'un fichier tabulaire comportant 4 colonnes.

A	B	C	D
true	false	false	true
true	false	true	true
false	true	false	false
true	true	false	true
true	false	false	false
...	...	...	...

- Il y a répétition d'évènements donc fréquences calculables donc représentable par un modèle probabiliste
- Chaque évènement est identifié par la liste des valeurs des variables A à D : modèle probabiliste factorisé
- Peut-on représenter ce système par un réseau bayésien ?

Vers un réseau bayésien (1) : χ^2

Pour **construire** un réseau bayésien (différent de **apprendre**), il faut isoler les indépendances conditionnelles dans ce modèle probabiliste factorisé : le χ^2 !

Soit X et Y deux v.a. binaires,

si $X \perp\!\!\!\perp Y$ alors $\quad \forall i,j,\ p(X=i, Y=j) = p(X=i) \cdot p(Y=j)$

Dans le cadre d'un test expérimental, on ne peut avoir que des estimations fréquentistes des probabilités :

si $X \perp\!\!\!\perp Y$ alors $\forall i,j,\ p(X=i, Y=j) = \frac{n_{ij}}{n} = p(X=i) \cdot p(Y=j) = \frac{n_{i.}}{n} \cdot \frac{n_{.j}}{n}$

Tester l'indépendance de X et Y revient donc à comparer $\frac{n_{ij}}{n}$ et $\frac{n_{i.}}{n} \cdot \frac{n_{.j}}{n}$.

Définition (χ^2 d'écart à l'indépendance)

$$d^2 = \sum_{i=1}^{r} \sum_{j=1}^{s} \frac{\left(n_{ij} - \frac{n_{i.} \cdot n_{.j}}{n}\right)^2}{\frac{n_{i.} \cdot n_{.j}}{n}}$$

alors $d^2 \leq n \cdot min(s-1, r-1)$ *suit une loi du* χ^2.

Vers un réseau bayésien (2) : tableau de contingence

Première étape donc : calculer les n_{ij} : le tableau de contingence.
En l'occurence, pour notre problème, ce sont des n_{ijkl} qu'il faut calculer (4 variables).

En supposant une base de données de 1000 expériences, on trouve :

		A=True		A=False	
		B=True	B=False	B=True	B=False
C=True	D=True	7	77	2	58
	D=False	5	307	2	230
C=False	D=True	65	19	22	14
	D=False	43	77	14	58

On peut vérifier que $7 + 77 + 2 + 58 + 5 + 307 + 2 + 230 + 65 + 19 + 22 + 14 + 43 + 77 + 14 + 58 = 1\,000$

Vers un BN (3) : $A \perp\!\!\!\perp B$?

		$\overline{a}$		$\underline{a}$	
		b	$\underline{b}$	b	$\underline{b}$
c	$\overline{d}$	7	77	2	58
	$\underline{d}$	5	307	2	230
$\overline{c}$	$\overline{d}$	65	19	22	14
	$\underline{d}$	43	77	14	58

Vers un BN (4) : $A \perp\!\!\!\perp C \mid B$?

		$\overline{a}$		$\underline{a}$	
		b	$\underline{b}$	b	$\underline{b}$
c	$\overline{d}$	7	77	2	58
	$\underline{d}$	5	307	2	230
$\overline{c}$	$\overline{d}$	65	19	22	14
	$\underline{d}$	43	77	14	58

Vers un BN (5) : liste d'indépendances

- $A \perp\!\!\!\perp C \,|\, B$
- $A \perp\!\!\!\perp D \,|\, B$
- $C \perp\!\!\!\perp D \,|\, B$

Un réseau bayésien

		$\overline{a}$		$\underline{a}$	
		b	$\underline{b}$	b	$\underline{b}$
$\overline{c}$	$\overline{d}$	7	77	2	58
	$\underline{d}$	5	307	2	230
$\underline{c}$	$\overline{d}$	65	19	22	14
	$\underline{d}$	43	77	14	58

Calcul dans un réseau bayésien

- Les quelques manipulations de base :

Marginalisation	$\sum_y P(x, y \mid z) = p(x \mid z)$
Somme totale	$\sum_y P(y \mid z) = 1$

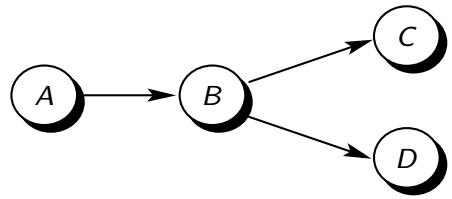
Décomposition	$P(x, y \mid z) = P(x \mid y, z) \cdot P(y \mid z)$
Chain rule	$P(X_1, \cdots, X_n) = \prod_{i=1}^{n} P\left(X_i \mid X_1, \cdots, X_{i-1}\right)$

Indépendance	$X \perp\!\!\!\perp Y \mid Z \Rightarrow P(x \mid y, z) = P(x \mid z)$
Loi de Bayes	$P(x \mid y, z) \propto P(y \mid x, z) \cdot P(x \mid z)$

- Dans un réseau bayésien :

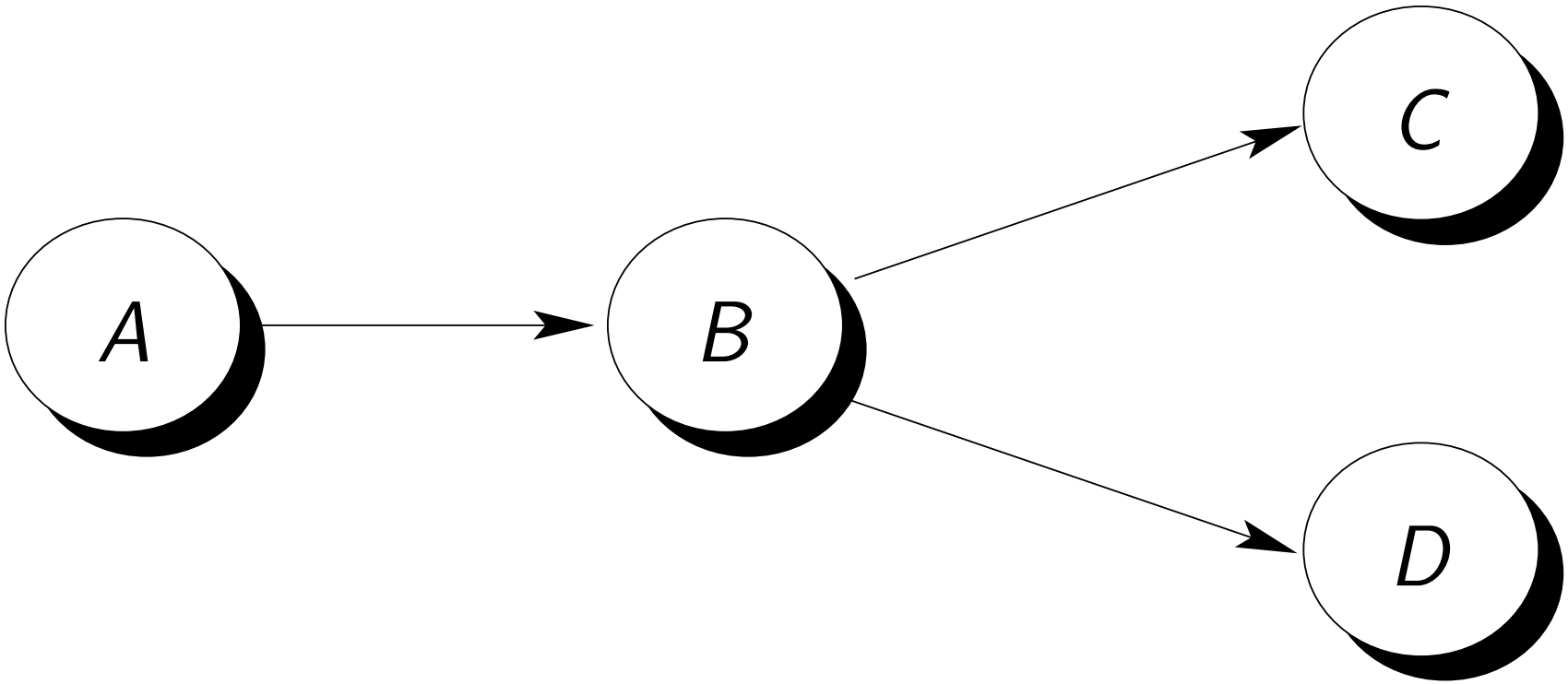
Markov local	$P(X_1, \cdots, X_n) = \prod_{i=1}^{n} P\left(X_i \mid \text{parents}(X_i)\right)$

Calcul dans un réseau bayésien (1) : $P(D)$?

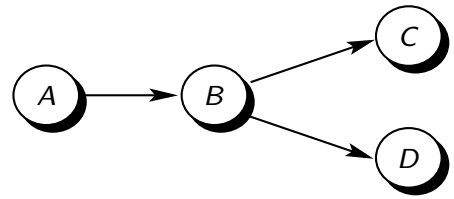


		$\overline{a}$		$\underline{a}$	
		b	$\underline{b}$	b	$\underline{b}$
$\overline{c}$	$\overline{d}$	7	77	2	58
	$\underline{d}$	5	307	2	230
$\underline{c}$	$\overline{d}$	65	19	22	14
	$\underline{d}$	43	77	14	58

$$
\begin{aligned}
P(d) &= \sum_a \sum_b \sum_c P(a,b,c,d) \\
&= \sum_a \sum_b \sum_c P(a) \cdot P(b \mid a) \cdot P(c \mid b) \cdot P(d \mid b) \\
&= \sum_a \sum_b P(a) \cdot P(b \mid a) \cdot P(d \mid b) \cdot \underbrace{\left(\sum_c P(c \mid b) \right)}_{=1} \\
&= \sum_b \underbrace{P(d \mid b)}_{\text{en } D} \cdot \left(\sum_a \underbrace{P(a)}_{\text{en } A} \cdot \underbrace{P(b \mid a)}_{\text{en } B} \right)
\end{aligned}
$$

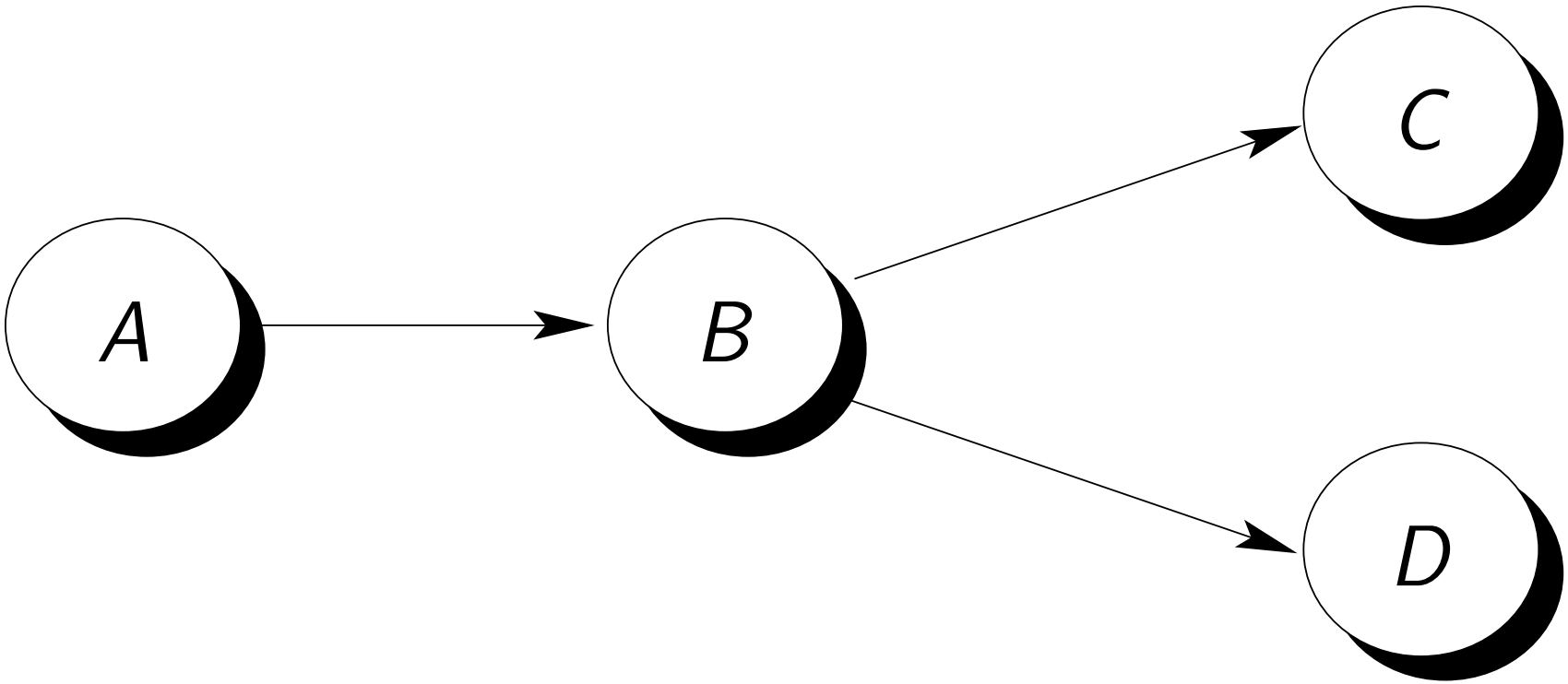

A
B
C
D

Calcul dans un réseau bayésien (2) : $P(D \mid \underline{a})$?

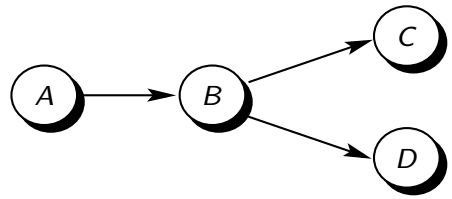


		$\overline{a}$		$\underline{a}$	
		b	$\underline{b}$	b	$\underline{b}$
$\overline{c}$	$\overline{d}$	7	77	2	58
	$\underline{d}$	5	307	2	230
$\underline{c}$	$\overline{d}$	65	19	22	14
	$\underline{d}$	43	77	14	58

$$
\begin{aligned}
P(d \mid \underline{a}) &= \sum_a \sum_b \sum_c P(a, b, c, d \mid \underline{a}) \\
&= \sum_a \sum_b \sum_c P(a \mid \underline{a}) \cdot P(b \mid a, \underline{a}) \cdot P(c \mid b, \underline{a}) \cdot P(d \mid b, \underline{a}) \\
&= \sum_a \sum_b P(a \mid \underline{a}) \cdot \underbrace{P(b \mid a, \underline{a})}_{\underline{a} \perp\!\!\!\perp B|A} \cdot \underbrace{P(d \mid b, \underline{a})}_{A \perp\!\!\!\perp D|B} \cdot \underbrace{\left(\sum_c P(c \mid b, \underline{a}) \right)}_{=1} \\
&= \sum_b \underbrace{P(d \mid b)}_{\text{en } D} \cdot \left(\sum_a \underbrace{P(a \mid \underline{a})}_{\text{en } A} \cdot \underbrace{P(b \mid a)}_{\text{en } B} \right)
\end{aligned}
$$


A
B
C
D

Calcul dans un réseau bayésien (3) : $P(C|\overline{d})$?



		$\overline{a}$		$\underline{a}$	
		b	$\underline{b}$	b	$\underline{b}$
$\overline{c}$	$\overline{d}$	7	77	2	58
	$\underline{d}$	5	307	2	230
$\underline{c}$	$\overline{d}$	65	19	22	14
	$\underline{d}$	43	77	14	58

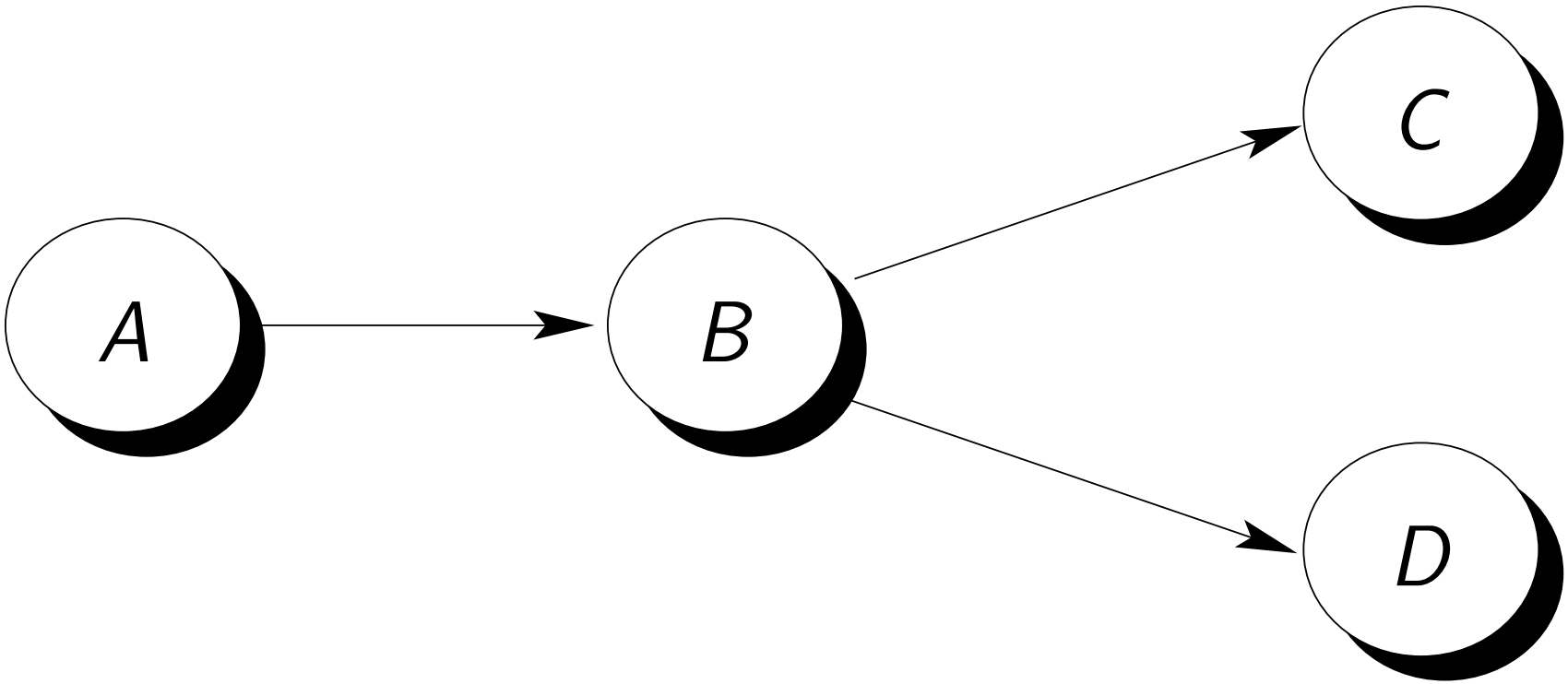
$$P(c \mid \overline{d}) = \sum_a \sum_b \sum_d P(a \mid \overline{d}) \cdot P(b \mid a, \overline{d}) \cdot P(c \mid b, \overline{d}) \cdot P(d \mid b, \overline{d})$$

$$= \sum_b \underbrace{P(c \mid b)}_{\text{en } C} \cdot \underbrace{\sum_a P(a \mid \overline{d}) \cdot P(b \mid a, \overline{d})}_{P(b|\overline{d})}$$

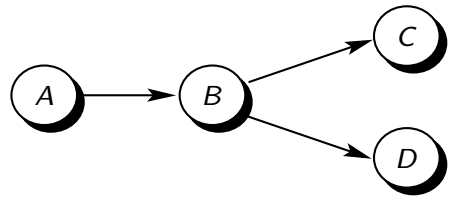
Utilisation de la loi de Bayes pour $P(b \mid \overline{d})$

$$P(b \mid \overline{d}) \propto P(\overline{d} \mid b) \cdot p(b)$$

$$\propto \underbrace{P(\overline{d} \mid b)}_{\text{en } D} \cdot \sum_a \underbrace{p(b \mid a)}_{\text{en } B} \cdot \underbrace{P(a)}_{\text{en } A}$$


A
B
C
D

Calcul dans un réseau bayésien (4) : $P(A|\overline{d})$?

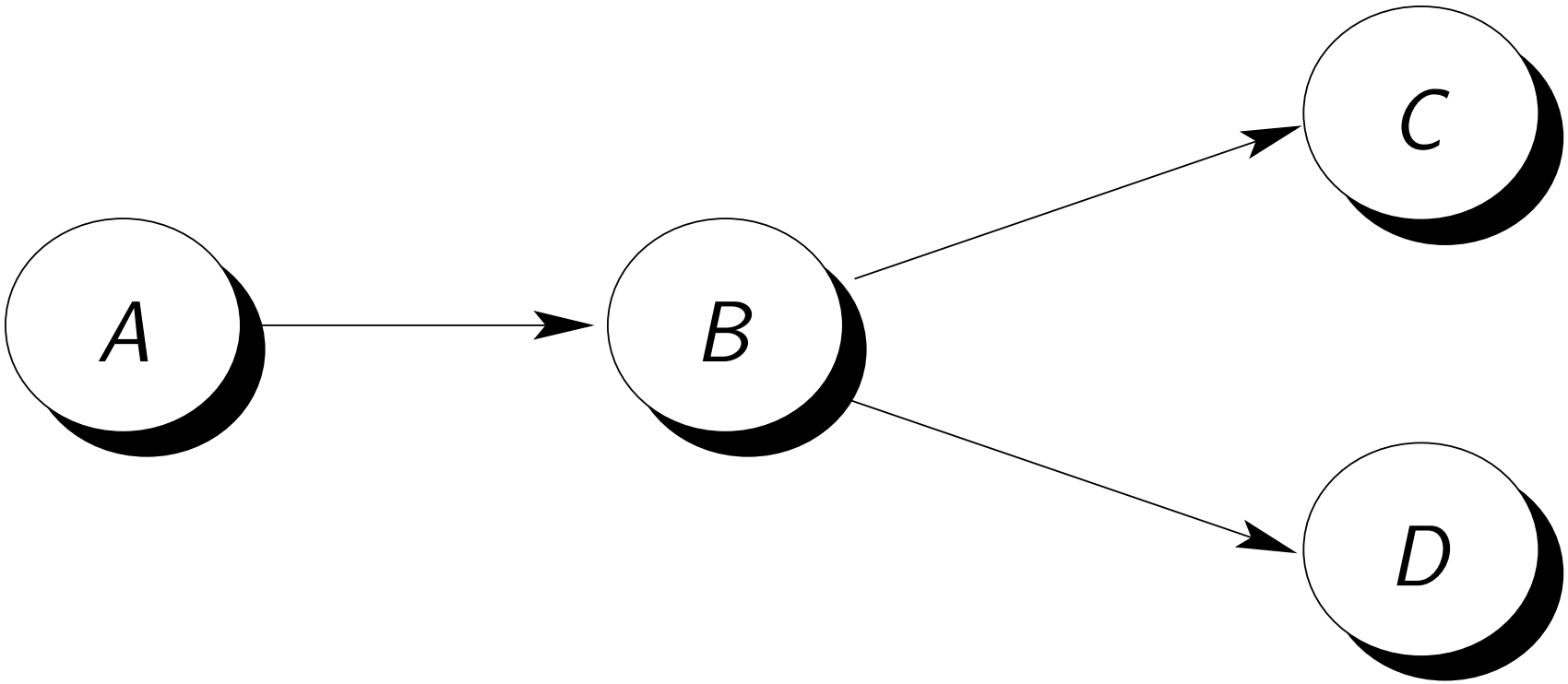


		$\overline{a}$		$\underline{a}$	
		b	$\underline{b}$	b	$\underline{b}$
$\overline{c}$	$\overline{d}$	7	77	2	58
	$\underline{d}$	5	307	2	230
$\underline{c}$	$\overline{d}$	65	19	22	14
	$\underline{d}$	43	77	14	58

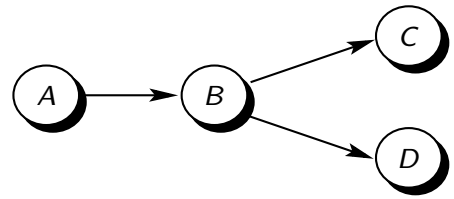
$$P(a \mid \overline{d}) \propto P(\overline{d} \mid a) \cdot \underbrace{P(a)}_{\text{en } A}$$

$$P(\overline{d} \mid a) = \sum_b P(\overline{d} \mid a, b) \cdot \underbrace{P(b \mid a)}_{\text{en } B}$$

$$= \sum_b \underbrace{P(\overline{d} \mid b)}_{\text{en } D} \cdot \underbrace{P(b \mid a)}_{\text{en } B}$$


A
B
C
D

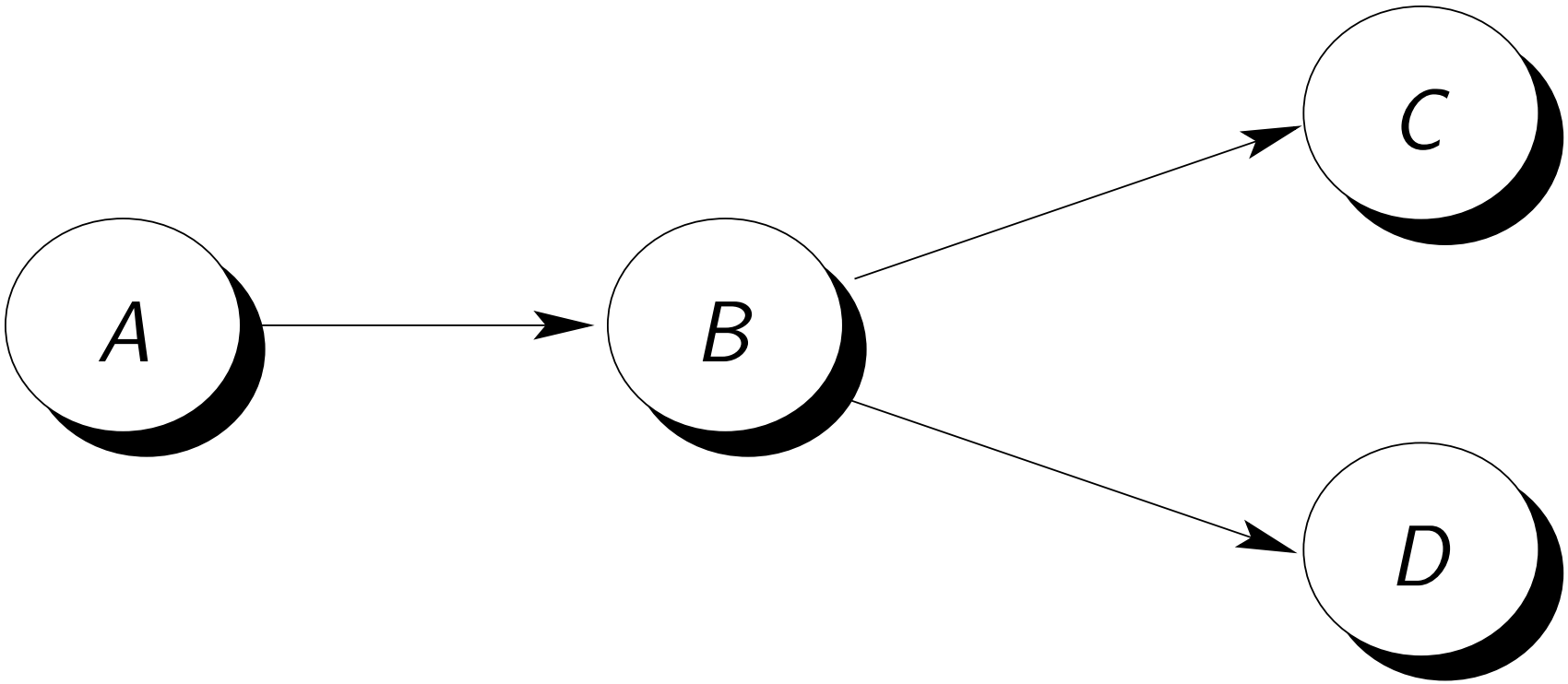
Calcul dans un réseau bayésien (5) : $P(B \mid \overline{c}, \underline{a})$?



		$\overline{a}$		$\underline{a}$	
		b	$\underline{b}$	b	$\underline{b}$
$\overline{c}$	$\overline{d}$	7	77	2	58
	$\underline{d}$	5	307	2	230
$\underline{c}$	$\overline{d}$	65	19	22	14
	$\underline{d}$	43	77	14	58

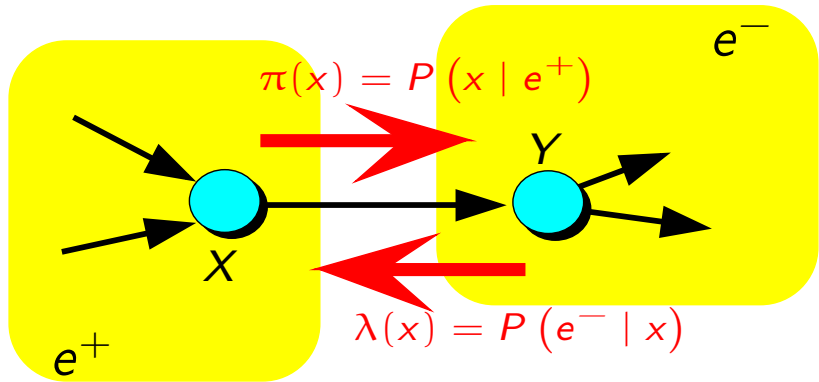
$$P(b \mid \overline{c}, \underline{a}) \propto P(\overline{c} \mid b, \underline{a}) \cdot P(b \mid \underline{a})$$

$$\propto \underbrace{P(\overline{c} \mid b)}_{\text{en } C} \cdot \sum_{a} \underbrace{p(b \mid a)}_{\text{en } B} \cdot \underbrace{P(a \mid \underline{a})}_{\text{en } A}$$

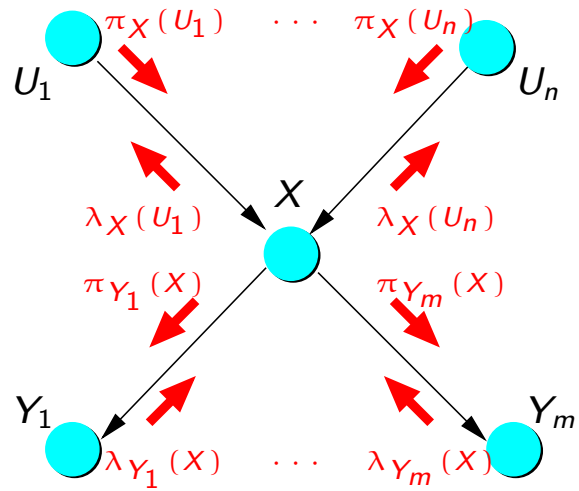

A
B
C
D

Inférence dans les poly-arbres (graphes orientés sans cycle)

Les messages transitent de nœuds en nœuds dans le sens (π-messages) ou en remontant (λ-messages) les arcs du graphe.



Une propagation de l'ensemble des messages sur la structure du graphe permet à tous les nœuds de connaître l'état global (toute l'information).

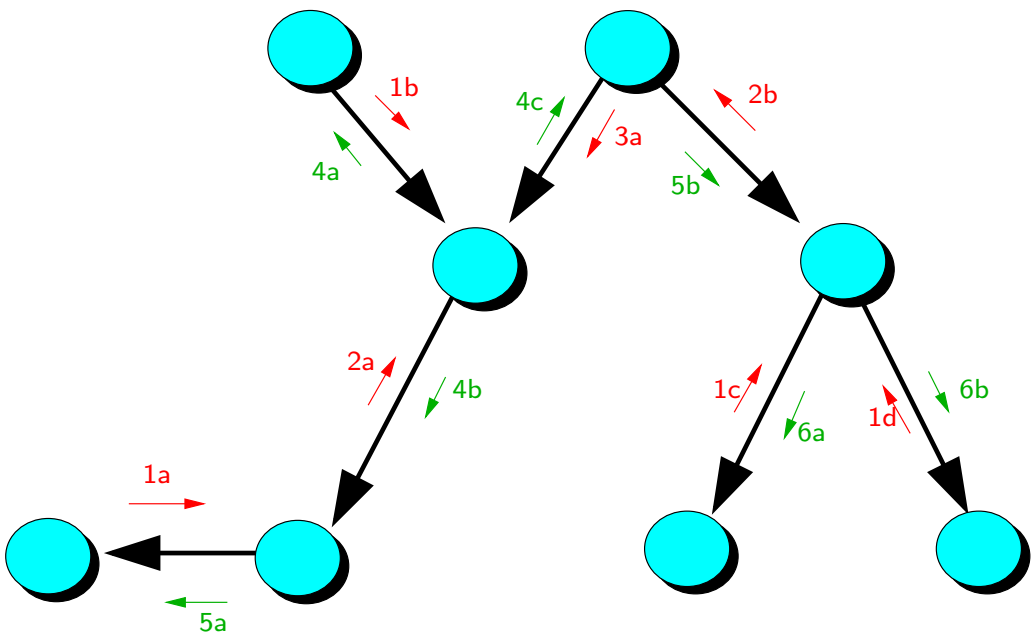


Si un nœud à n parents, il doit connaître les messages issus de $n-1$ de ses voisins pour pouvoir envoyer le message vers son $n^{\text{ème}}$ voisin.

Algorithme de propagation de messages (version simplifiée sans information)

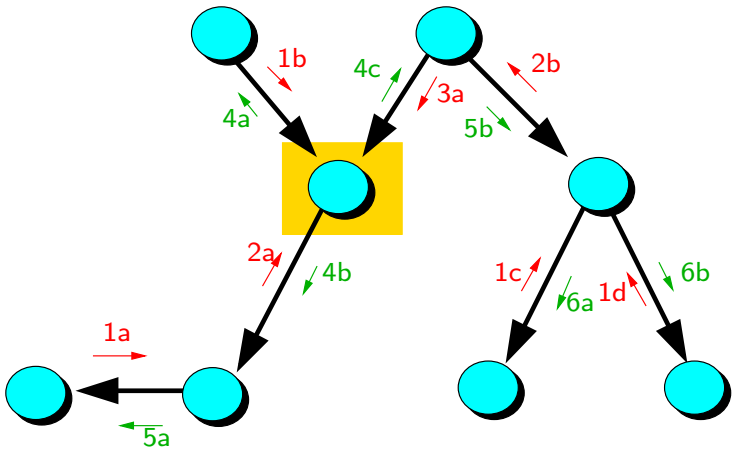
Version asynchrone

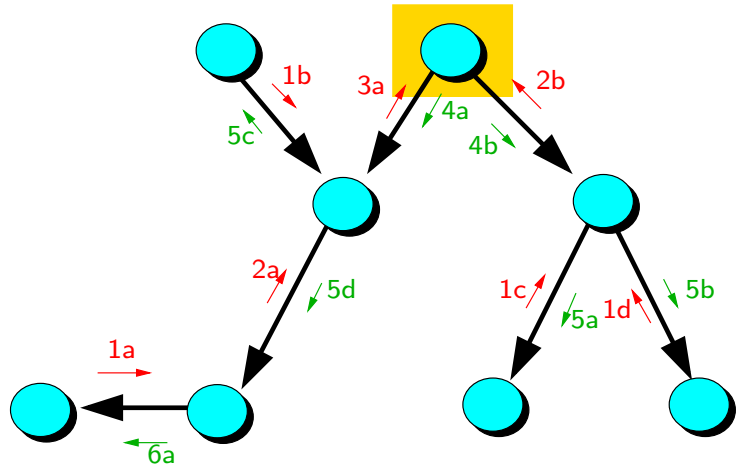
- Initialisation
  - les nœuds sans parents peuvent envoyer leurs messages π.
  - les nœuds sans enfants peuvent envoyer leurs messages λ.
- Propagation : pour chaque nœud (avec n voisins),
  - si $n - 1$ messages reçus alors envoi du message vers le $n^{\text{ème}}$ voisin.
  - si n messages reçus alors envoi des messages restants à envoyer vers tous les voisins (et calcul de sa loi a posteriori).



On note que la complexité de cet algorithme est proportionnelle au nombre d'arcs dans le graphe.

Algorithme de propagation de messages (2)





Version centralisée

- Choix d'une racine
- Absorption

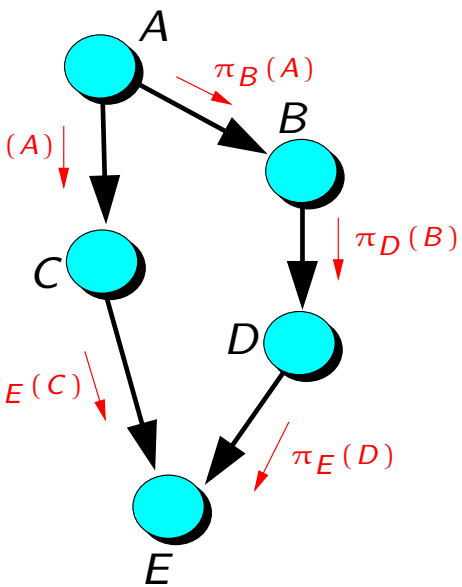
  Tout nœud envoie son message vers la racine dès qu'il le peut.
- Intégration

  La racine a reçu les messages de tous ses voisins. Elle envoie donc ses messages vers tous ses voisins.
- Diffusion

  Tout nœud ayant reçu le message venant de la racine a reçu tous ses messages et, donc, peut envoyer tous ses autres messages.

Problèmes dans un graphe avec cycles (mais toujours sans circuit)

$$P(E) = \sum_{A,B,C,D} P(A) \cdot P(B \mid A) \cdot P(C \mid A) \cdot P(D \mid B) \cdot P(E \mid C, D)$$



En utilisant l'algorithme des messages :

1. $\pi_B(A) = P(A)$
2. $\pi_C(A) = P(A)$
3. $\pi_D(B) = \sum_A P(B \mid A) \cdot \pi_B(A)$
4. $\pi_E(D) = \sum_B P(D \mid B) \mid \pi_D(B)$
5. $\pi_E(C) = \sum_A P(C \mid A) \cdot \pi_C(A)$
6. $P(E) = \sum_{D,C} P(E \mid C, D) \cdot \pi_E(C) \cdot \pi_E(D)$

$$P(E) \neq \sum_{*} P(E \mid C, D) \cdot P(C \mid A) \cdot P(A) \cdot P(D \mid B) \cdot P(B \mid A') \cdot P(A')$$

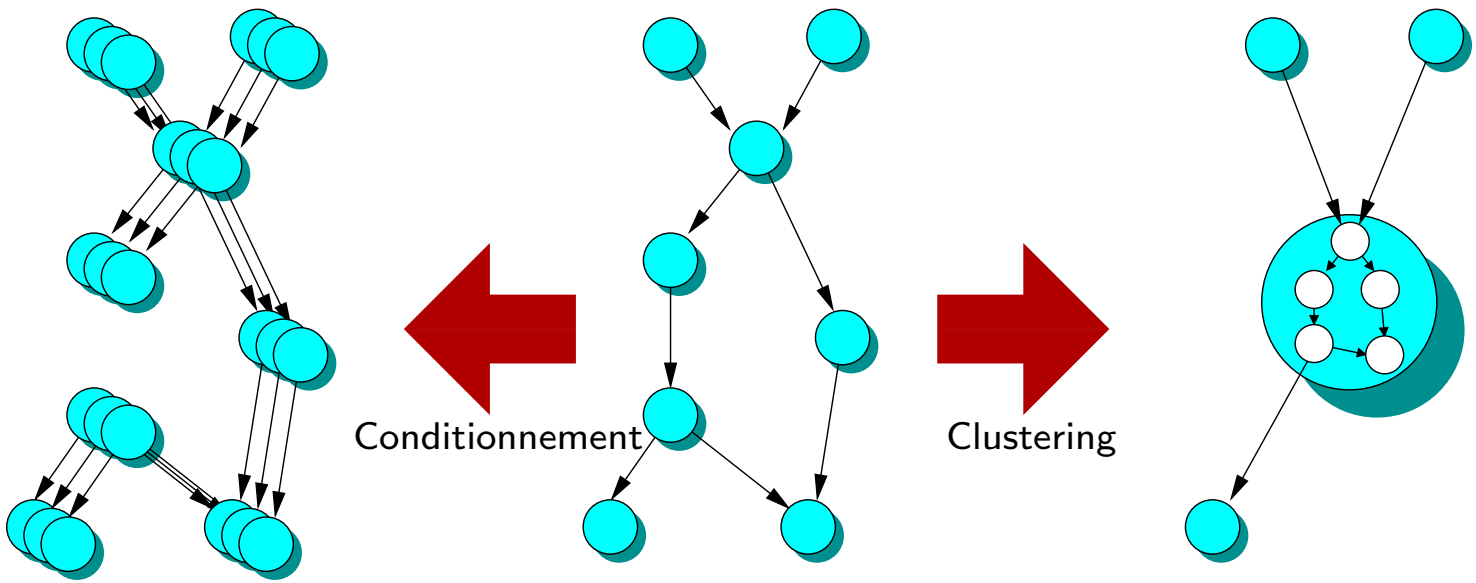
Propager dans des graphes avec cycles (mais toujours sans circuit) ?

La propagation par messages qui permet ne fonctionne que dans un graphe sans cycles.

Se ramener à un graphe sans cycle

Méthodes multiples, par exemple (pour les plus connues) :

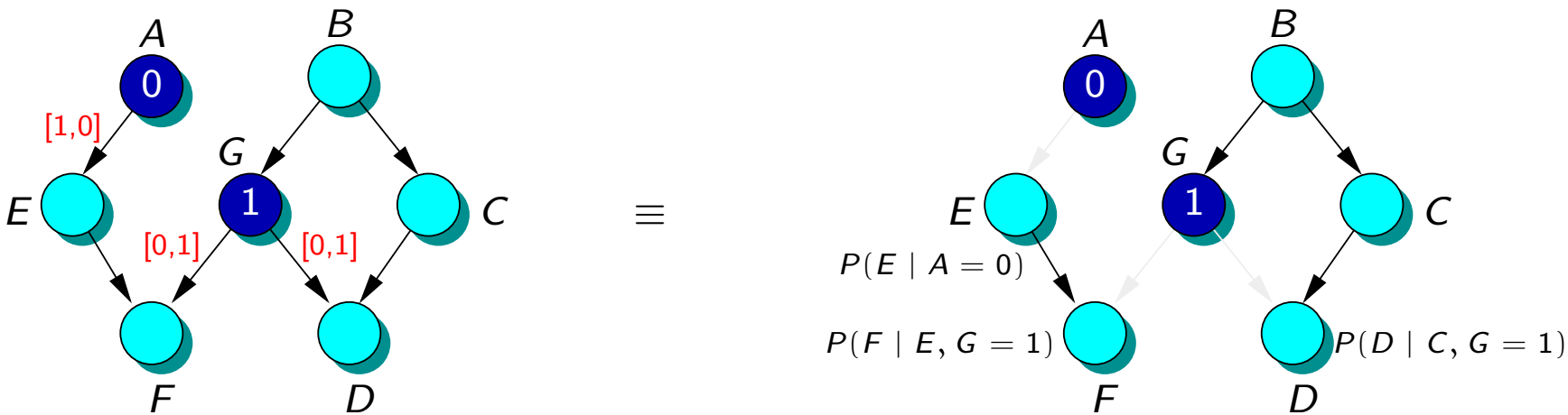
- Conditioning : Couper le graphe (retirer des arcs) jusqu'à obtenir un graphe sans cycle.
- Clustering : Regrouper (fusionner) les nœuds jusqu'à obtenir un graphe sans cycle.



Conditioning : méthode coupe-cycle

Conditionnement par S

Soit un réseau bayésien sur l'ensemble de variables V. Soient $S \subset V$ et s une instanciation des variables de S. Lors de la propagation de l'information s, tous les messages π issus d'une variable de S seront déterministes. Cette propagation est alors équivalente à la propagation dans un réseau bayésien où les arcs issus des nœuds de S seraient supprimés.



Ensemble de coupe

On appelle ensemble de coupe, un ensemble S de nœuds qui permet de supprimer les cycles du réseau bayésien.

Conditioning : méthode coupe-cycle (2)

On rappelle que $\forall S \subset V$, on a la propriété :

$$\forall x \in V, P(x) = \sum_s P(x \mid s).P(s)$$

Algorithme du coupe-cyle (global)

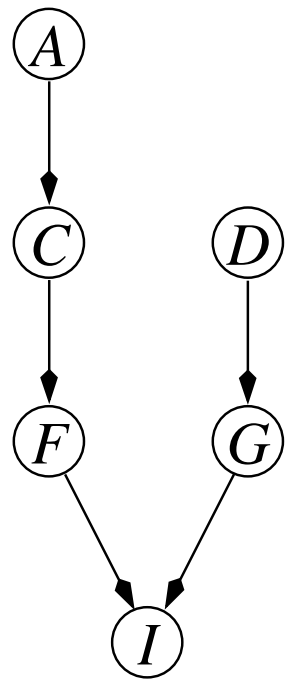
Soit un réseau bayésien G,

1. Soit $S = \{S_1, \cdots, S_n\}$ un ensemble de coupe sur G
2. $\forall s$ instanciation de S,
   - Calculer $p_s = P(s)$.
   - Calculer $P_s(x) = P(x \mid s)$ dans le réseau sans cycle.
3. $P(x) = \sum_s (p_s \cdot P_s(x))$

Si on note $\#_i$ le nombre de modalité de la variable S_i, cet algorithme consiste donc à calculer $\prod_i \#_i$ inférences dans un graphe sans cycle.

Osons le calcul des probabilités a priori

$$P(A, C, D, F, G, I) = P(A)P(C|A)P(F|C)P(D)P(G|D)P(I|F, G)$$



Calcul de $P(I)$?

Shafer-Shenoy brut

$$P(A,C,D,F,G,I) = P(A)P(C|A)P(F|C)P(D)P(G|D)P(I|F,G)$$

$$P(I) = \sum_G \left(\sum_F \left(\sum_D \left(\sum_C \left(\sum_A P(A,C,D,F,G,I) \right) \right) \right) \right)$$

$$\sum_A P(A,C,D,F,G,I) = \underbrace{\left(\sum_A P(A)P(C|A) \right)}_{P(C)} P(F|C)P(D)P(G|D)P(I|F,G)$$

$$\sum_C \sum_A P(A,C,D,F,G,I) = \underbrace{\left(\sum_C P(C)P(F|C) \right)}_{P(F)} P(D)P(G|D)P(I|F,G)$$

Dissection du produit de deux probabilités

$$P(A,B|C) = \begin{pmatrix} \overbrace{0,15}^{c_1} & \overbrace{0,18}^{c_2} & \overbrace{0,07}^{c_1} & \overbrace{0,56}^{c_2} \\ 0,15 & 0,12 & 0,63 & 0,14 \end{pmatrix} \begin{matrix} b_1 \\ b_2 \end{matrix} = \underbrace{\begin{pmatrix} \overbrace{0,5}^{c_1} & \overbrace{0,6}^{c_2} & \overbrace{0,1}^{c_1} & \overbrace{0,8}^{c_2} \\ 0,5 & 0,4 & 0,9 & 0,2 \end{pmatrix}}_{P(B|A,C)} \begin{matrix} b_1 \\ b_2 \end{matrix} \times \underbrace{\begin{pmatrix} \overbrace{0,3}^{a_1} & \overbrace{0,7}^{a_2} \end{pmatrix}}_{P(A)}$$

$$P(I,C|B) = \begin{pmatrix} \overbrace{0,48}^{c_1} & \overbrace{0,08}^{c_2} & \overbrace{0,48}^{c_1} & \overbrace{0,08}^{c_2} \\ 0,12 & 0,32 & 0,12 & 0,32 \end{pmatrix} \begin{matrix} i_1 \\ i_2 \end{matrix} = \overbrace{\begin{pmatrix} 0,8 & 0,2 \\ 0,2 & 0,8 \end{pmatrix}}^{P(I|C)} \begin{matrix} i_1 \\ i_2 \end{matrix} \times \overbrace{\begin{pmatrix} 0,6 & 0,4 \end{pmatrix}}^{P(C)}$$

Shafer-Shenoy graphique (1/6)

Séquence d'élimination

A *C* *D* *F* *G*

$$P(A,C,D,F,G,I) = P(A)P(C|A)\,P(F|C)P(D)P(G|D)P(I|F,G)$$

$$\begin{array}{cccccc} A & AC & FC & D & GD & IFG \\ P(A) & P(C|A) & P(F|C) & P(D) & P(G|D) & P(I|F,G) \end{array}$$

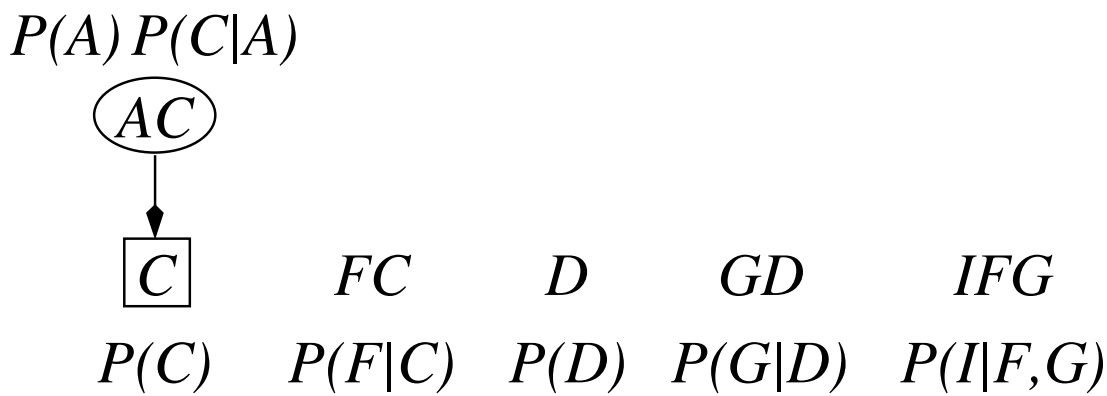
$$\text{somme sur } A \Longrightarrow P(C) = \sum_{A} P(A)P(C|A)$$

Shafer-Shenoy graphique (2/6)

Séquence d'élimination

A C D F G

$$P(C,D,F,G,I) = P(C)P(F|C)\,P(D)P(G|D)P(I|F,G)$$



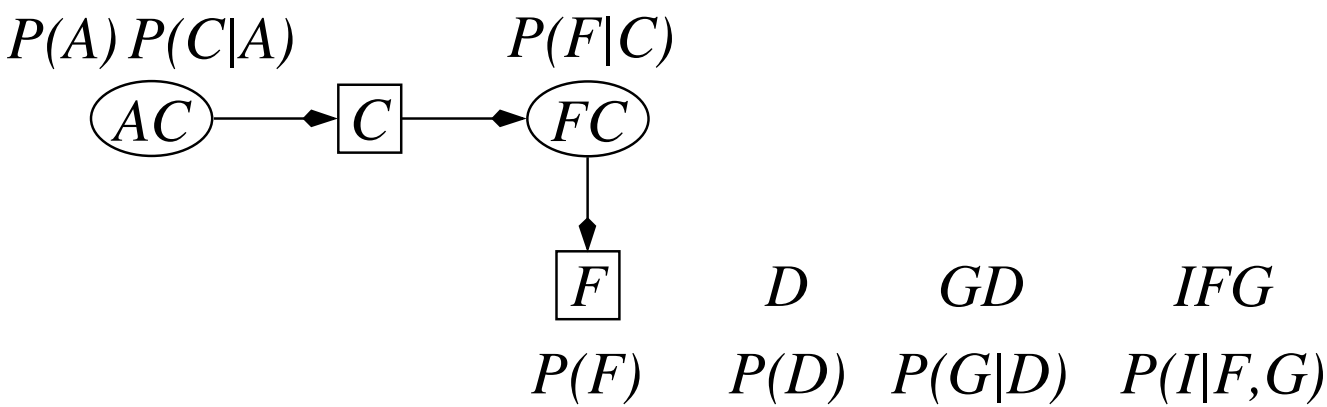
somme sur $C \Longrightarrow P(F) = \sum_{C} P(C)P(F|C)$

Shafer-Shenoy graphique (3/6)

Séquence d'élimination

$A\ C\ D\ F\ G$

$$P(D,F,G,I) = P(F)\ P(D)P(G|D)\ P(I|F,G)$$



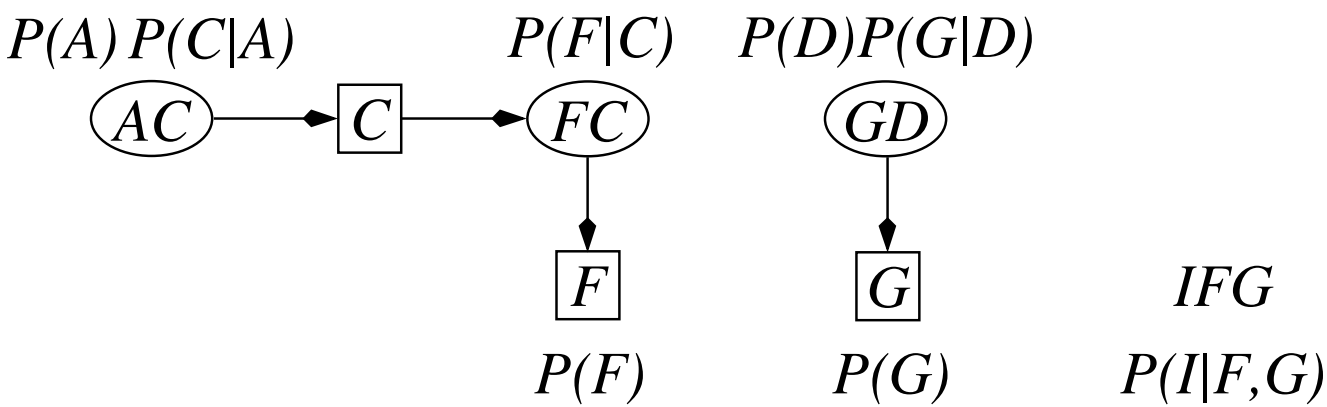
$$\text{somme sur } D \Longrightarrow P(G) = \sum_{D} P(D)P(G|D)$$

Shafer-Shenoy graphique (4/6)

Séquence d'élimination

$A\ C\ D\ \boxed{F}\ G$

$$P(F,G,I) = \boxed{P(F)P(I|F,G)}\ P(G)$$



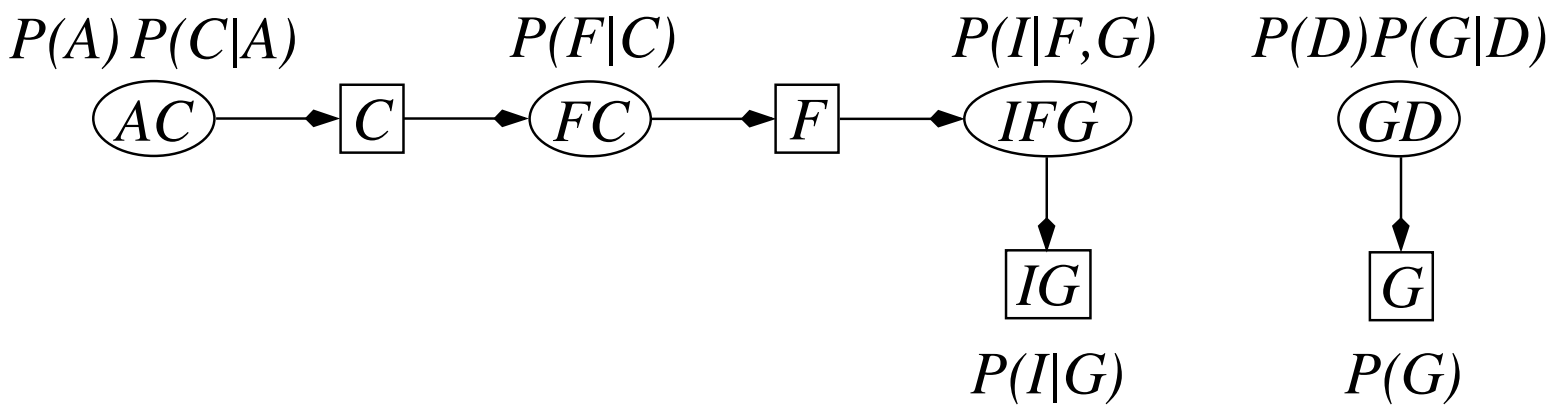
$$\text{somme sur } F \implies P(I|G) = \sum_F P(F)P(I|F,G)$$

Shafer-Shenoy graphique (5/6)

Séquence d'élimination

$A\ C\ D\ F\ \boxed{G}$

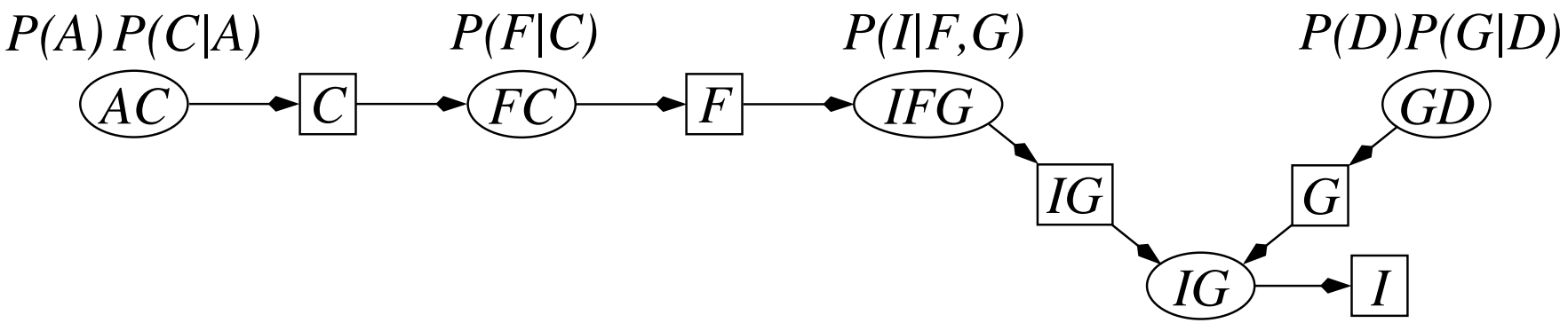
$P(G, I) = \boxed{P(I|G)P(G)}$



$$\text{somme sur } G \Longrightarrow P(I) = \sum_{G} P(G)P(I|G)$$

Shafer-Shenoy graphique (6/6)

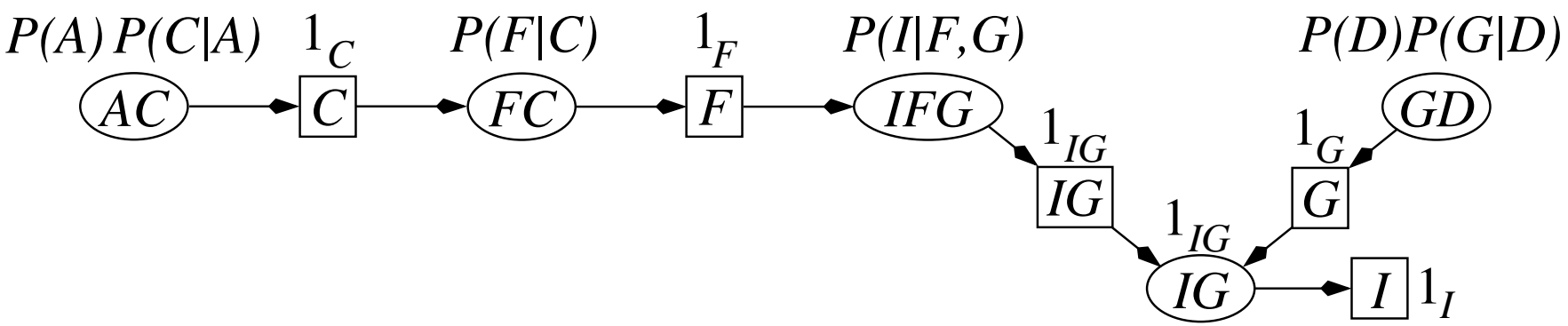
Le graphe final obtenu par Shafer-Shenoy



Algorithme de Shafer-Shenoy

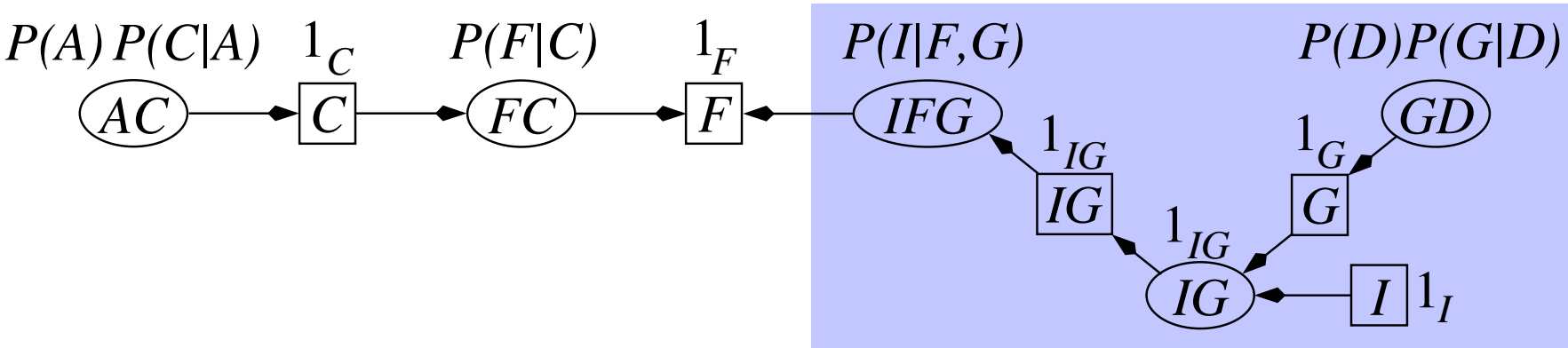
1. Se donner une séquence d'élimination des nœuds
   $\Longrightarrow$ *join tree*,
2. propager les impacts dans le sens des flèches :
   - dans les ellipses (cliques), on effectue des multiplications,
   - dans les rectangles (séparateurs), on effectue des additions (projections).

Quelques remarques sur Shafer-Shenoy (1/3)

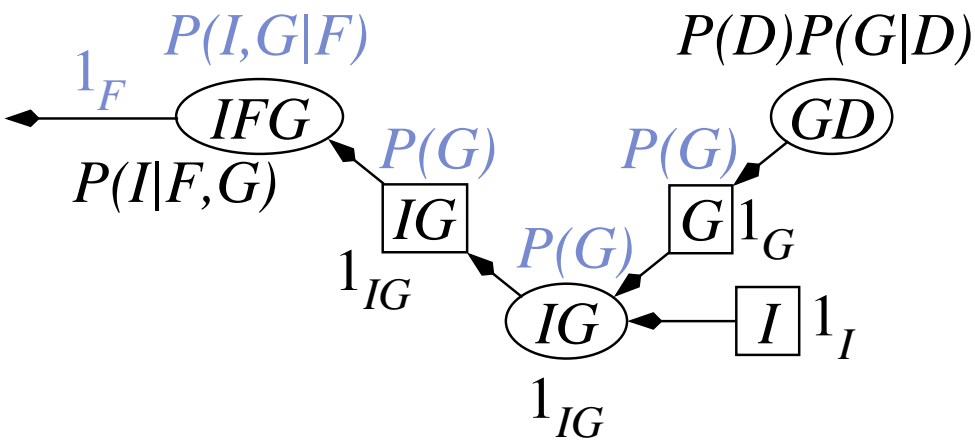


1. Loi jointe = produit des fonctions des cliques et des séparateurs : $P(A, C, D, F, G, I) = P(A)P(C|A)P(F|C)P(D)P(G|D)P(I|F, G)$.
2. Élimination récursive des cliques et séparateurs «externes» $\Longrightarrow$ le produit des fonctions des cliques et des séparateurs restants = loi jointe des variables restantes.

Quelques remarques sur Shafer-Shenoy (2/3)



$$P(A) \times P(C|A) \times P(F|C) \times 1_C \times 1_F = P(A, C, F).$$



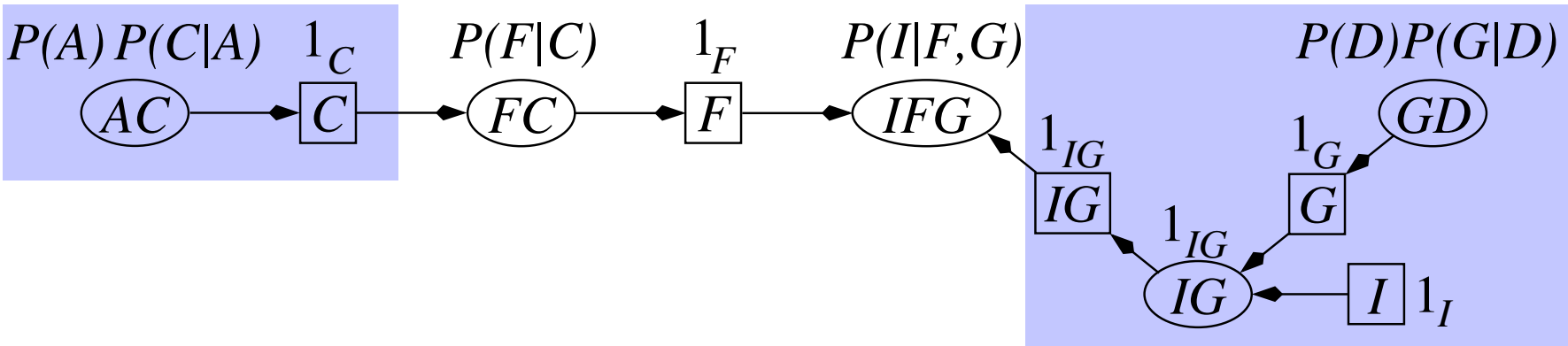
séparateur G : $\sum_D P(D)P(G|D)$

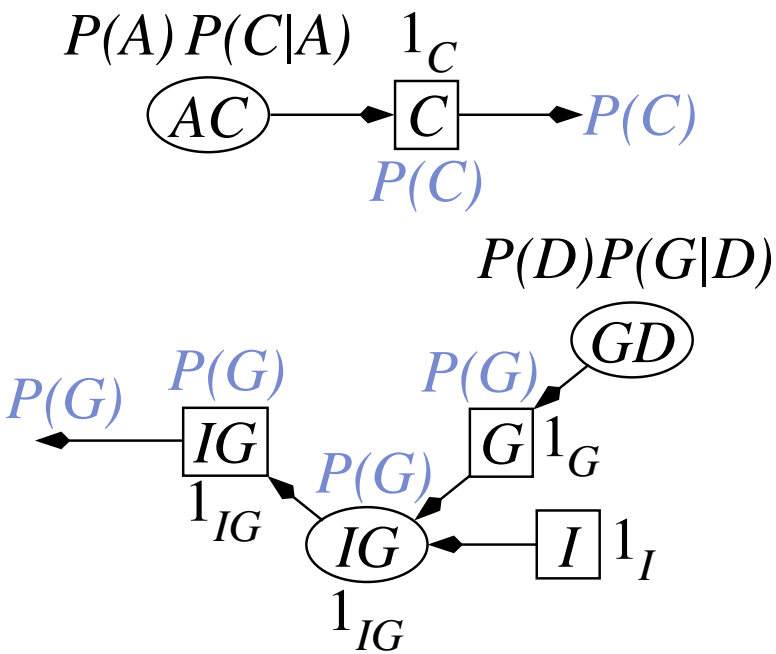
clique IG : $P(G) \times 1_{IG} \times 1_G \times 1_I$

clique IFG : $\sum_{IG} P(I|F,G) \times P(G)$

$\Longrightarrow$ en sortie de IFG : 1_F

Quelques remarques sur Shafer-Shenoy (3/3)





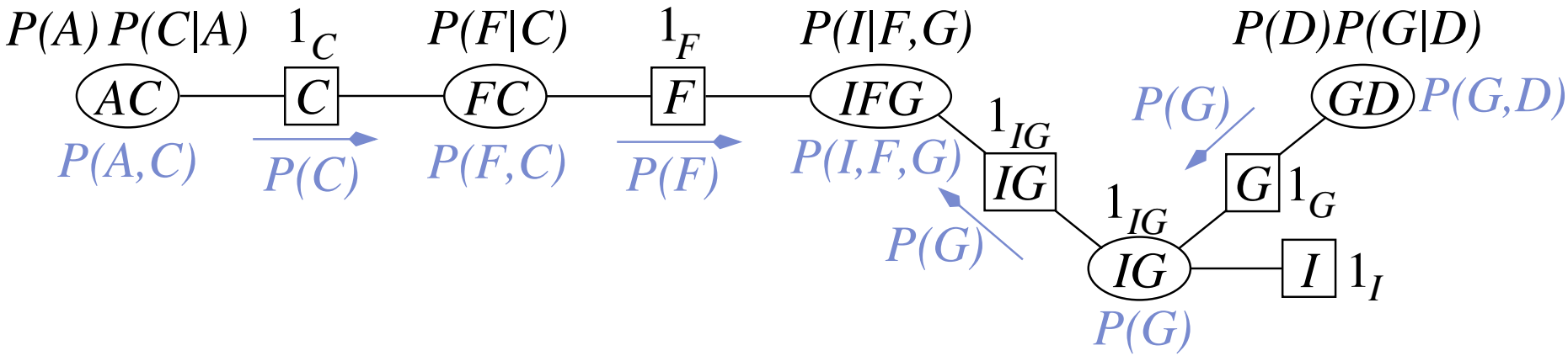
Produit des cliques et séparateurs restants :

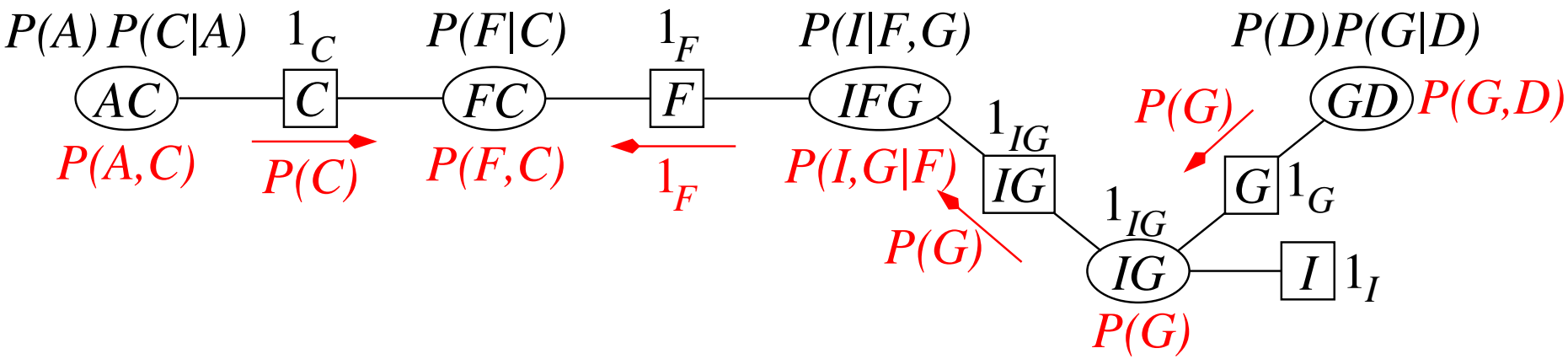
$P(C)\,P(F|C)1_F P(I|F,G)\,P(G)$
$= P(I,C,F,G)$

Conclusion : on peut utiliser le même graphe pour calculer toutes les probabilités marginales

C'est pas le deux en un, mais le tout en deux (1/3)

En bleu : les calculs de $P(I, F, G)$, en rouge, ceux de $P(F, C)$





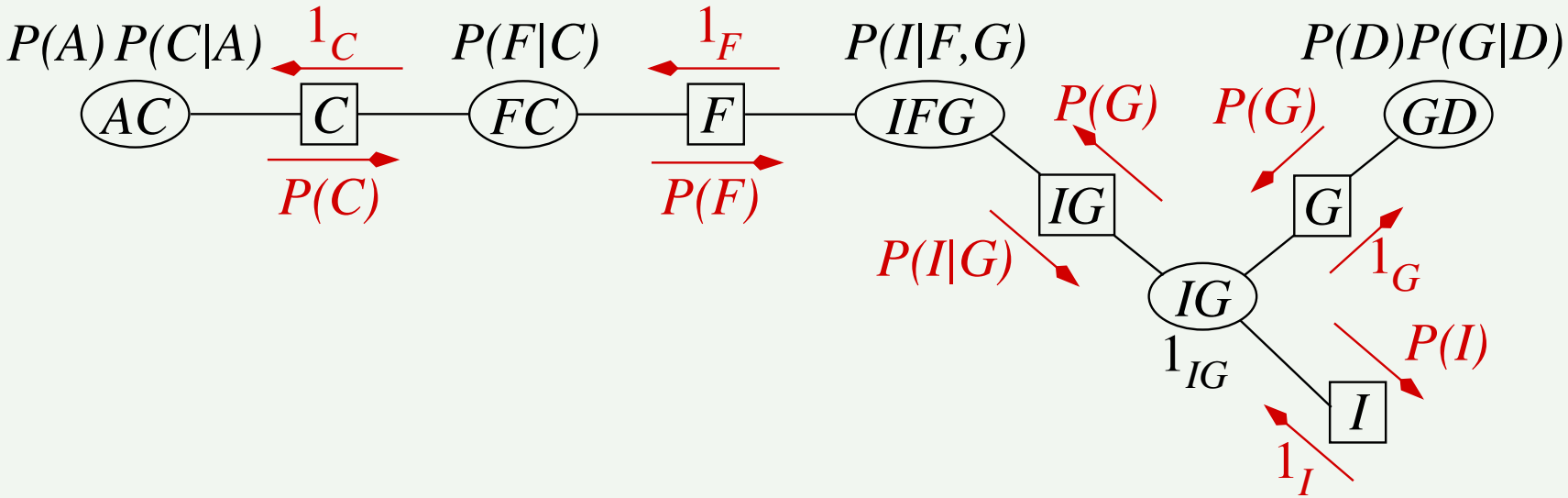
C'est pas le deux en un, mais le tout en deux (2/3)

Algorithme de Shafer-Shenoy

1. Chaque séparateur contient deux messages initialisés à 1, un en direction de chaque clique voisine.
2. Chaque nœud du join tree envoie des messages vers ses voisins en respectant les deux règles suivantes :
    1. avant d'envoyer un message vers son voisin X, le nœud Y attend que tous ses autres voisins lui aient envoyé leur message.
    2. le message d'un nœud Y vers son voisin X est le produit de tous les messages reçus par Y, à l'exception de celui envoyé par X, et de la table stockée par Y, le tout marginalisé sur X (c'est-à-dire sommé sur les variables de $Y \backslash X$).

C'est pas le deux en un, mais le tout en deux (3/3)

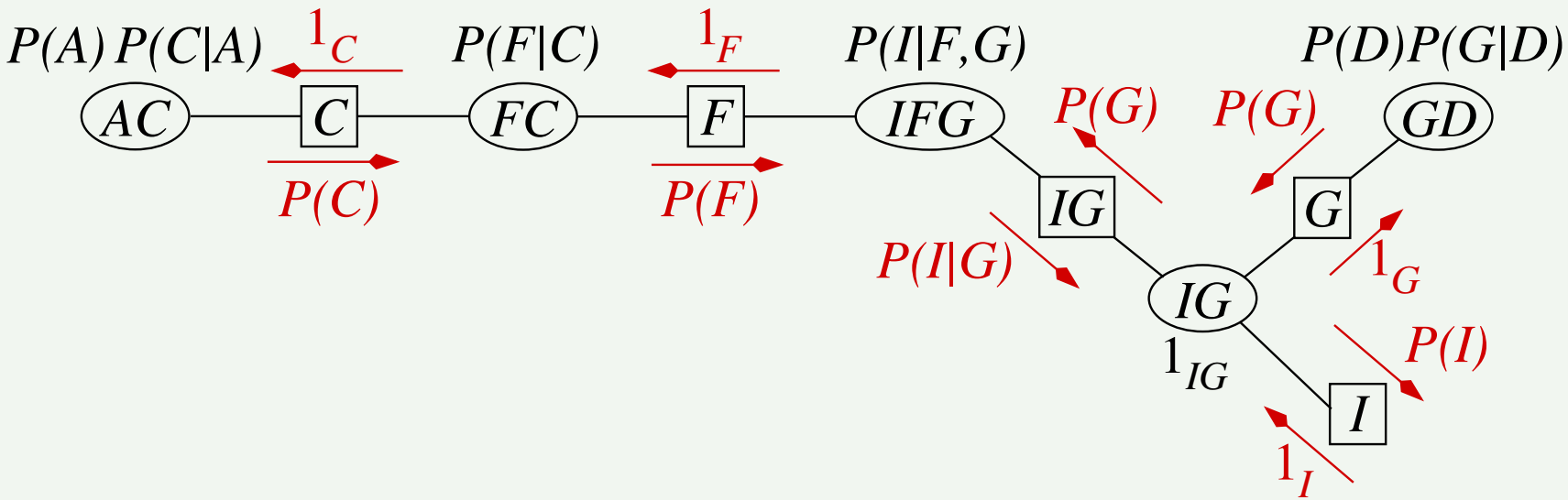
Algorithme de Shafer-Shenoy



À la fin de l'algorithme, pour tout nœud X, le produit de la table stockée en X par l'ensemble des messages envoyés à X est la probabilité jointe des variables de X.

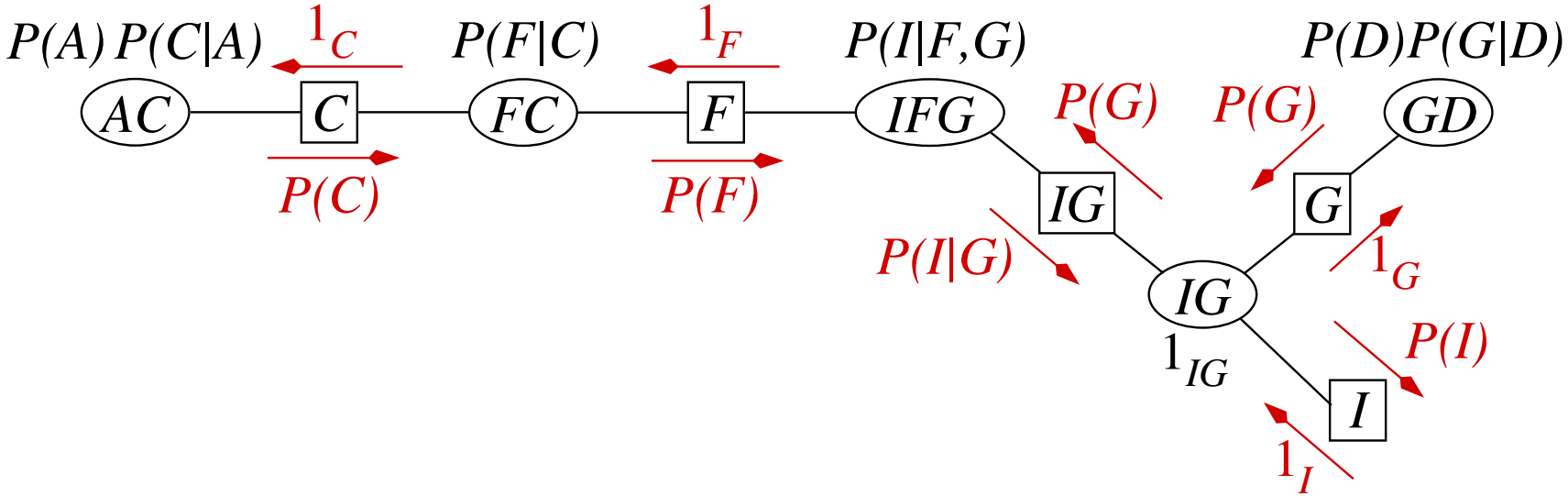
Shafer-Shenoy : ce qu'il faut retenir

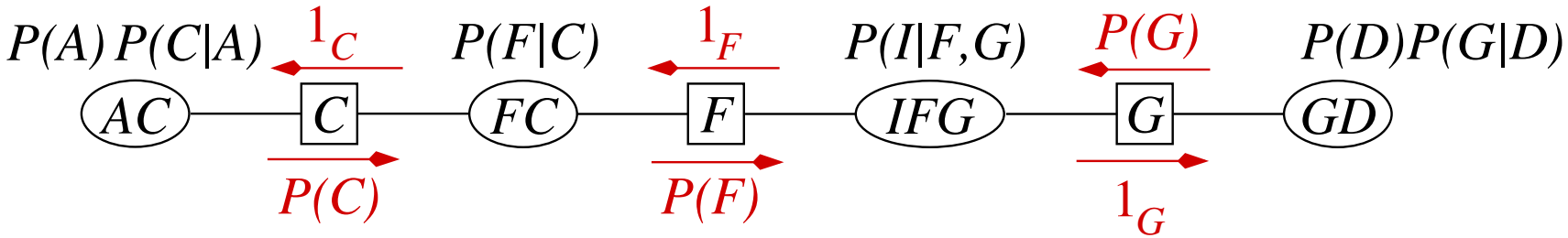
Algorithme de Shafer-Shenoy



- cliques = ellipses, séparateurs = rectangles
- algorithme par envoi de messages
- opérateurs = $+$ sur les séparateurs, $\times$ sur les cliques

Les arbres de jonction





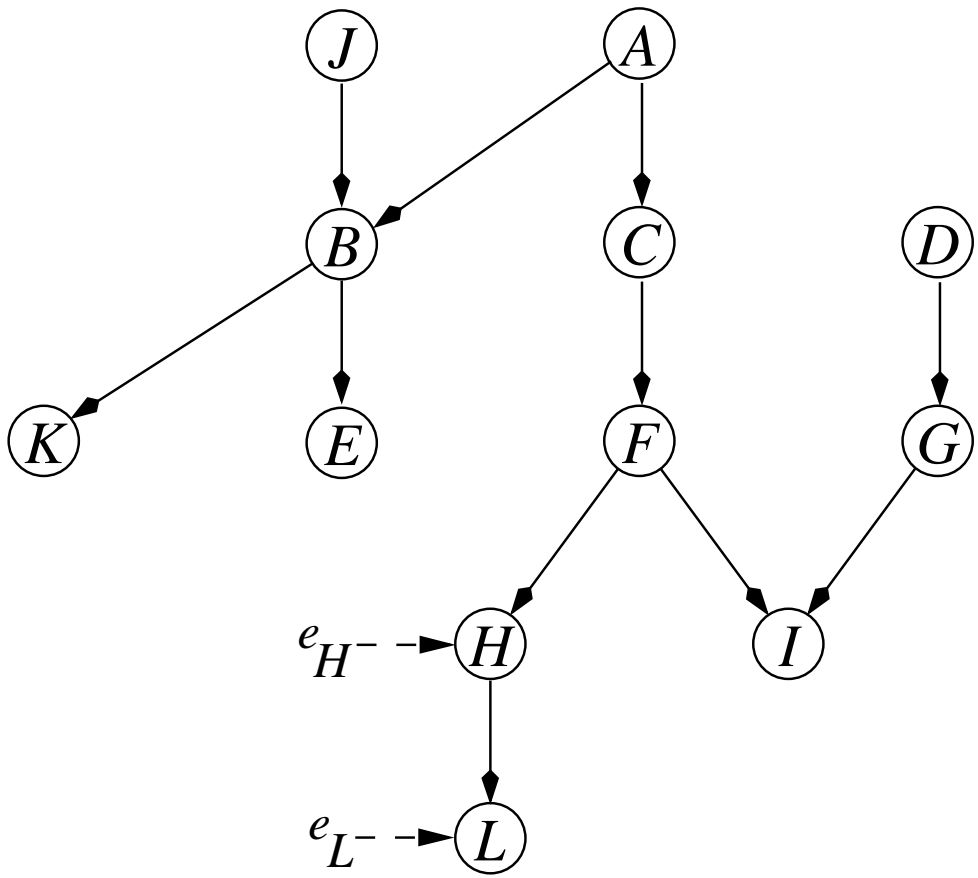
arbre de jonction : suppression des cliques incluses dans d'autres cliques

Soyons observateur : les probabilités a posteriori

Nous observons des informations e_H sur H et e_L sur L.

On veut connaître $P(I|e_H, e_L)$

$\Longrightarrow$ on va calculer $P(I, e_H, e_L)$ car c'est plus simple



Quelles sont ces informations ?

Teneur des informations

informations = observations :

$e_L = \ll L$ ne peut plus prendre les valeurs l_1 et $l_4 \gg$

Entrée de ces observations : $P(e_L|L)$

Vous avez dit calcul de $P(e_L|L)$? (1/2)

observation : $e_L = \ll$ Le capteur m'indique que L ne peut plus prendre les valeurs l_1 et $l_4 \gg$

$$P(e_L|L) = \begin{array}{|c|l} \hline 0 & l_1 \\ \hline 1 & l_2 \\ \hline 1 & l_3 \\ \hline 0 & l_4 \\ \hline \end{array}$$

dimension de $P(e_L|L) = |L|$

Vous avez dit calcul de $P(e_L|L)$? (2/2)

observation : $e_L =$ « Le capteur m'indique que L a pris la valeur l_2. Mais je n'ai pas totalement confiance en ce capteur. Je pense que

- il y a 90% de chances qu'il indique l_2 lorsque $L = l_2$, mais
- il y a également 10% de chances qu'il indique l_2 lorsque $L = l_1$ ou $L = l_3$. »

$$P(e_L|L) = \begin{array}{|c|l} \hline 0.1 & l_1 \\ \hline 0.9 & l_2 \\ \hline 0.1 & l_3 \\ \hline 0 & l_4 \\ \hline \end{array}$$

Hypothèses et conséquences (1/4)

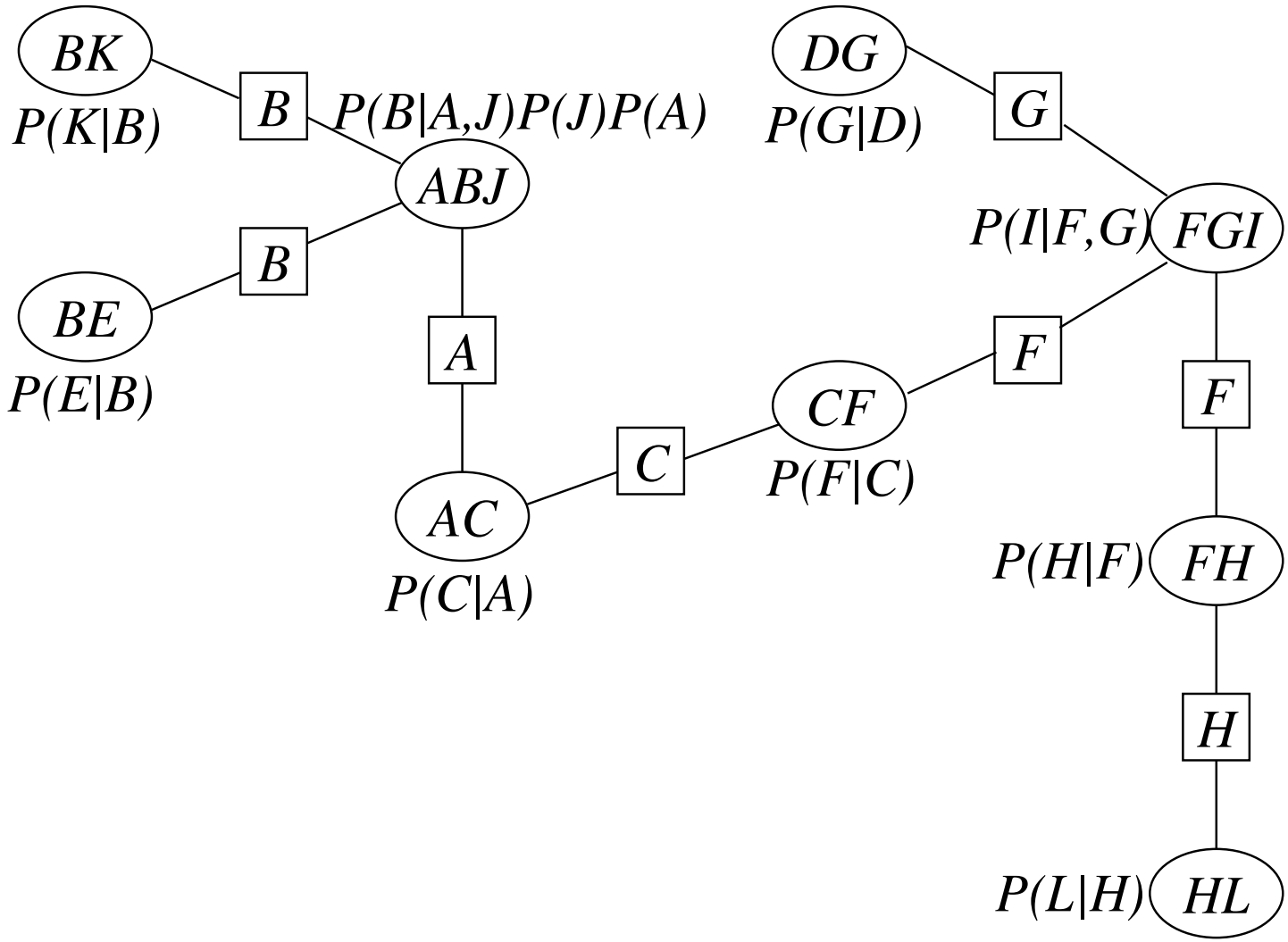
Hypothèse

Toute information e_X sur un nœud X est indépendante du reste du réseau bayésien conditionnellement à X.

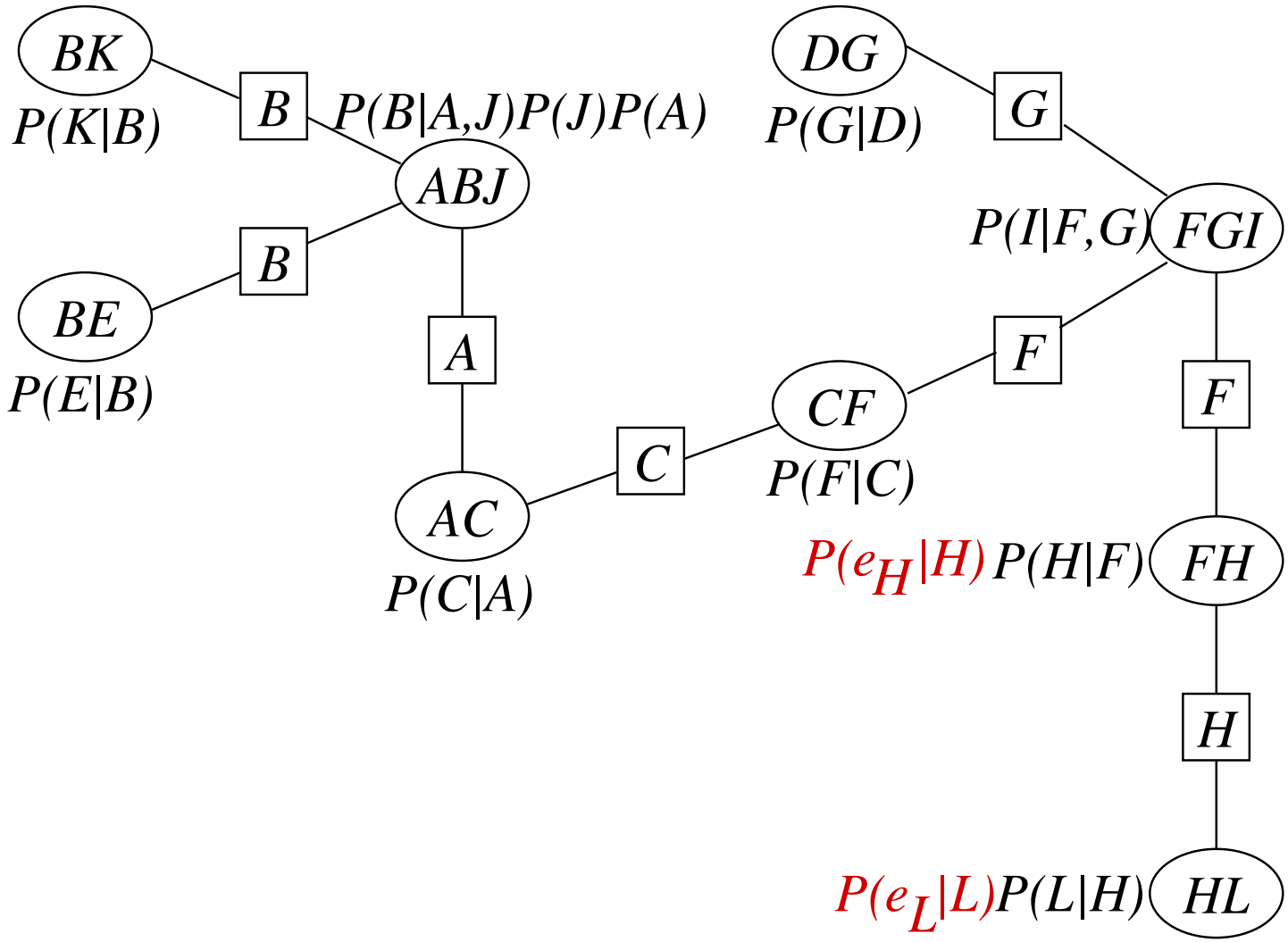
Conséquence

$$
\begin{aligned}
&P(A, B, C, D, E, F, G, H, I, J, K, L, e_H, e_L) = \\
&\qquad P(e_H|A, B, C, D, E, F, G, H, I, J, K, L, e_L) \times \\
&\qquad P(e_L|A, B, C, D, E, F, G, H, I, J, K, L) \times \\
&\qquad P(A, B, C, D, E, F, G, H, I, J, K, L) \\
&= P(e_H|H)P(e_L|L)P(A, B, C, D, E, F, G, H, I, J, K, L)
\end{aligned}
$$

Hypothèses et conséquences (2/4)



Hypothèses et conséquences (3/4)



Hypothèses et conséquences (4/4)

L'algorithme de Shafer-Shenoy permet donc de calculer dans la clique FGI : $P(F, G, I, e_H, e_L)$

$$\Longrightarrow P(I, e_H, e_L) = \sum_{F,G} P(F, G, I, e_H, e_L)$$

$$\Longrightarrow P(I|e_H, e_L) = \frac{P(I, e_H, e_L)}{P(e_H, e_L)}$$

Or $P(e_H, e_L) = \sum_{I} P(I, e_H, e_L)$

Donc $P(I|e_H, e_L) = \frac{P(I, e_H, e_L)}{\sum_{I} P(I, e_H, e_L)}$.

Résumé sur l'algorithme de Shafer-Shenoy

Algorithme de Shafer-Shenoy

Pour calculer une proba *a posteriori* $P(Y|e)$:

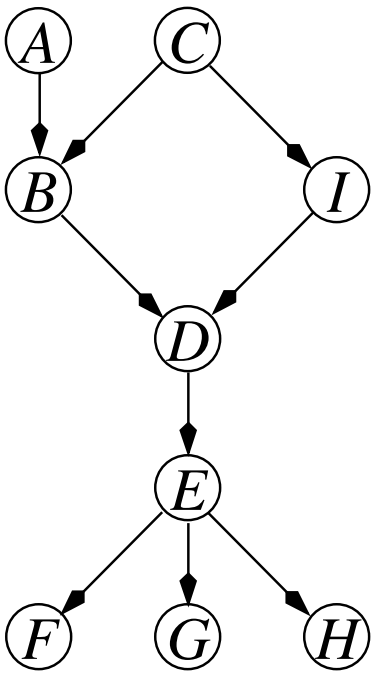
1. construire l'arbre de jonction
2. insérer les probas conditionnelles du réseau bayésien dans les cliques
3. insérer des tables contenant uniquement des 1 dans les séparateurs
4. pour chaque information e_X, calculer $P(e_X|X)$ et insérer cette proba dans une clique contenant X
5. envoyer les messages dans l'arbre de jonction
6. choisir une clique contenant Y, faire le produit de sa table de proba par tous les messages qui lui ont été envoyés, puis sommer sur toutes les variables $\neq Y \Longrightarrow P(Y, e)$
7. normaliser $P(Y, e) \Longrightarrow P(Y|e)$

Petit guide pratique sur Shafer-Shenoy

Quelques références

1. G. Shafer (1996) *Probabilistic expert systems*, Society for Industrial and Applied Mathematics.
2. P.P. Shenoy, G. Shafer (1990) *Axioms for probability and belief-function propagation*, Uncertainty in Artificial Intelligence 4, pp.169–198.
3. P.P. Shenoy (1997) *Binary join trees for computing marginals in the Shenoy-Shafer architecture*, International Journal of Approximate Reasoning 17, pp.1–25.
4. V. Lepar, P.P. Shenoy (1998) *A Comparison of Lauritzen-Spiegelhalter, Hugin and Shenoy-Shafer Architectures for Computing Marginals of Probability Distributions*, Proceedings of UAI-98, pp.328–337.

Exemple à l'usage des étudiants studieux (1/9)



$$P(A) = \begin{matrix} a_1 & a_2 \\ (0{,}3 & 0{,}7) \end{matrix}$$

$$P(C) = \begin{matrix} c_1 & c_2 \\ (0{,}6 & 0{,}4) \end{matrix}$$

$$P(B|A,C) = \begin{matrix} a_1 & & a_2 & \\ c_1 & c_2 & c_1 & c_2 \\ \multicolumn{4}{c}{\begin{pmatrix} 0{,}5 & 0{,}6 & 0{,}1 & 0{,}8 \\ 0{,}5 & 0{,}4 & 0{,}9 & 0{,}2 \end{pmatrix}} \end{matrix} \begin{matrix} b_1 \\ b_2 \end{matrix}$$

$$P(I|C) = \begin{matrix} c_1 & c_2 \\ \begin{pmatrix} 0{,}8 & 0{,}2 \\ 0{,}2 & 0{,}8 \end{pmatrix} \end{matrix} \begin{matrix} i_1 \\ i_2 \end{matrix}$$

$$P(D|B,I) = \begin{matrix} b_1 & & b_2 & \\ i_1 & i_2 & i_1 & i_2 \\ \multicolumn{4}{c}{\begin{pmatrix} 0{,}1 & 0{,}4 & 0{,}7 & 0{,}8 \\ 0{,}9 & 0{,}6 & 0{,}3 & 0{,}2 \end{pmatrix}} \end{matrix} \begin{matrix} d_1 \\ d_2 \end{matrix}$$

$$P(E|D) = \begin{matrix} d_1 & d_2 \\ \begin{pmatrix} 0{,}4 & 0{,}3 \\ 0{,}5 & 0{,}4 \\ 0{,}1 & 0{,}3 \end{pmatrix} \end{matrix} \begin{matrix} e_1 \\ e_2 \\ e_3 \end{matrix}$$

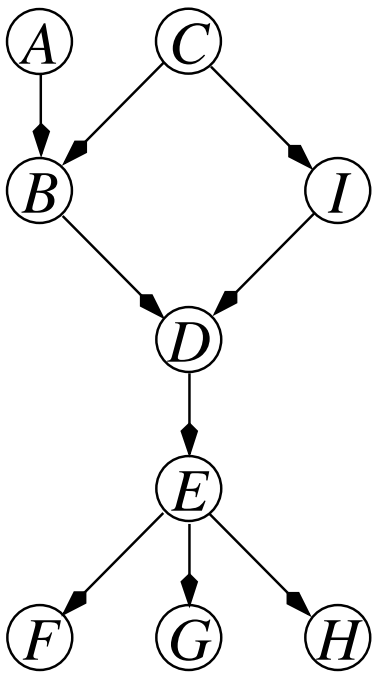
$$P(F|E) = \begin{matrix} e_1 & e_2 & e_3 \\ \begin{pmatrix} 0{,}2 & 0{,}7 & 0{,}4 \\ 0{,}8 & 0{,}3 & 0{,}6 \end{pmatrix} \end{matrix} \begin{matrix} f_1 \\ f_2 \end{matrix}$$

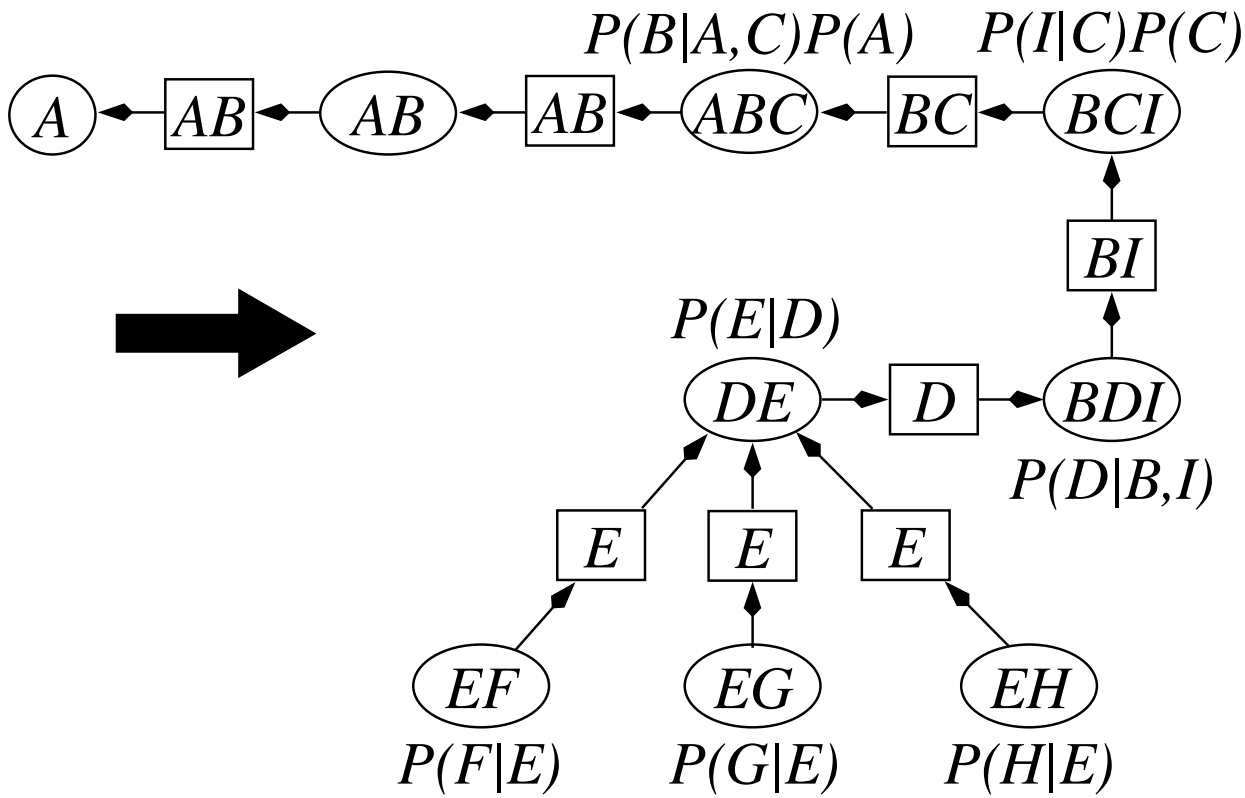
$$P(G|E) = \begin{matrix} e_1 & e_2 & e_3 \\ \begin{pmatrix} 0{,}6 & 0{,}1 & 0{,}3 \\ 0{,}4 & 0{,}9 & 0{,}7 \end{pmatrix} \end{matrix} \begin{matrix} g_1 \\ g_2 \end{matrix}$$

$$P(H|E) = \begin{matrix} e_1 & e_2 & e_3 \\ \begin{pmatrix} 0{,}5 & 0{,}6 & 0{,}2 \\ 0{,}5 & 0{,}4 & 0{,}8 \end{pmatrix} \end{matrix} \begin{matrix} h_1 \\ h_2 \end{matrix}$$

Exemple à l'usage des étudiants studieux (2/9)

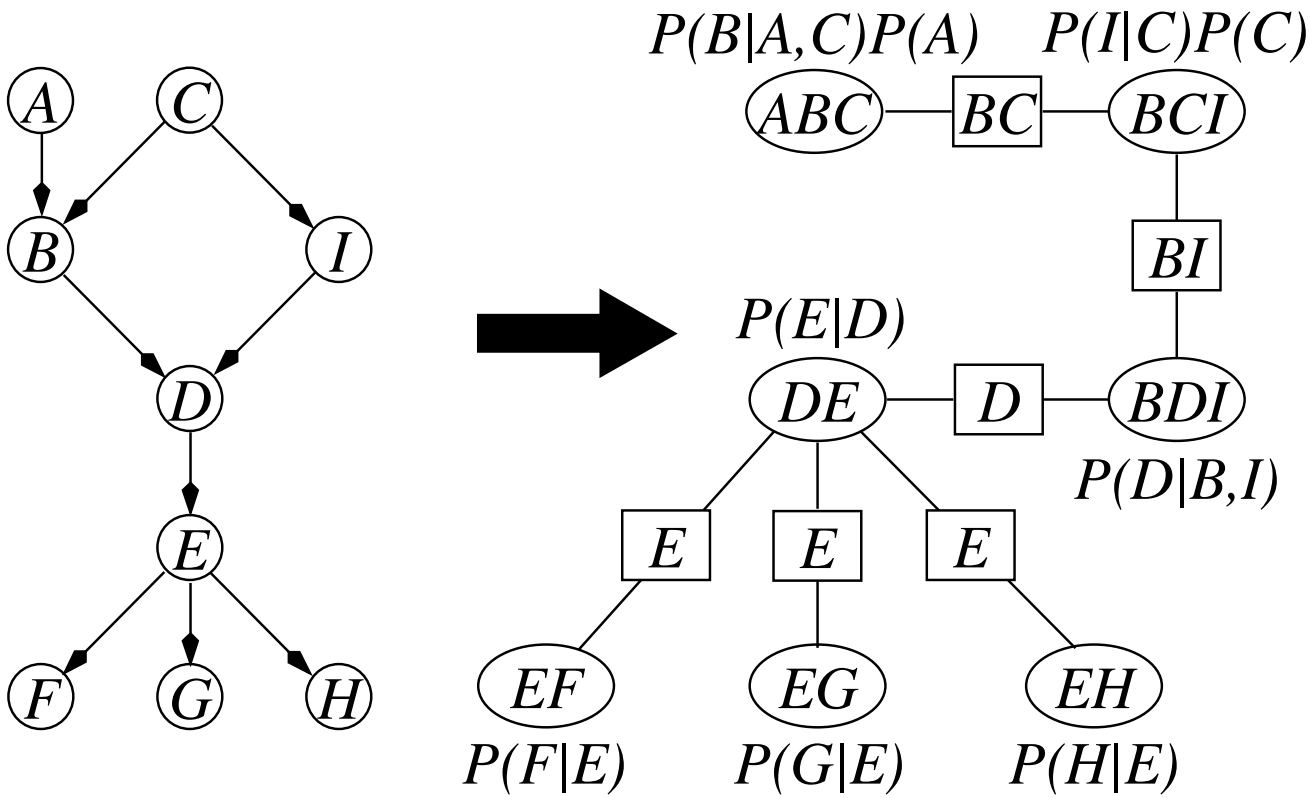
Séquence d'élimination : $F, H, G, E, D, I, C, B, A$



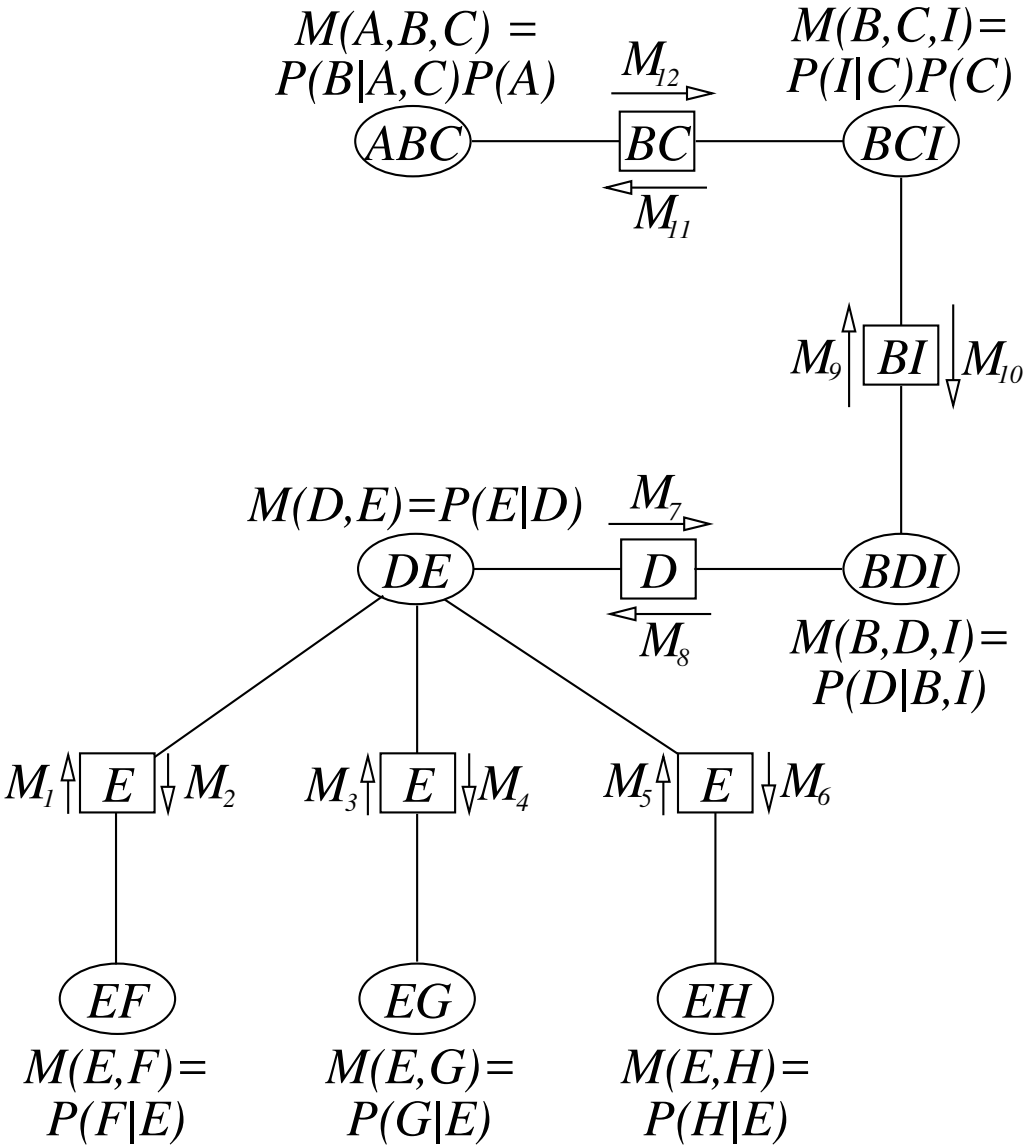


Exemple à l'usage des étudiants studieux (3/9)

Séquence d'élimination : $F, H, G, E, D, I, C, B, A$



Exemple à l'usage des étudiants studieux (4/9)



initialisation

$$M(A,B,C) = \begin{array}{c} \overbrace{}^{a_1}\overbrace{}^{a_2} \\ c_1\ c_2\ c_1\ c_2 \end{array} \begin{pmatrix} 0{,}15 & 0{,}18 & 0{,}07 & 0{,}56 \\ 0{,}15 & 0{,}12 & 0{,}63 & 0{,}14 \end{pmatrix} \begin{array}{l} b_1 \\ b_2 \end{array}$$

$$M(B,C,I) = \begin{array}{c} \overbrace{}^{b_1}\overbrace{}^{b_2} \\ c_1\ c_2\ c_1\ c_2 \end{array} \begin{pmatrix} 0{,}48 & 0{,}08 & 0{,}48 & 0{,}08 \\ 0{,}12 & 0{,}32 & 0{,}12 & 0{,}32 \end{pmatrix} \begin{array}{l} i_1 \\ i_2 \end{array}$$

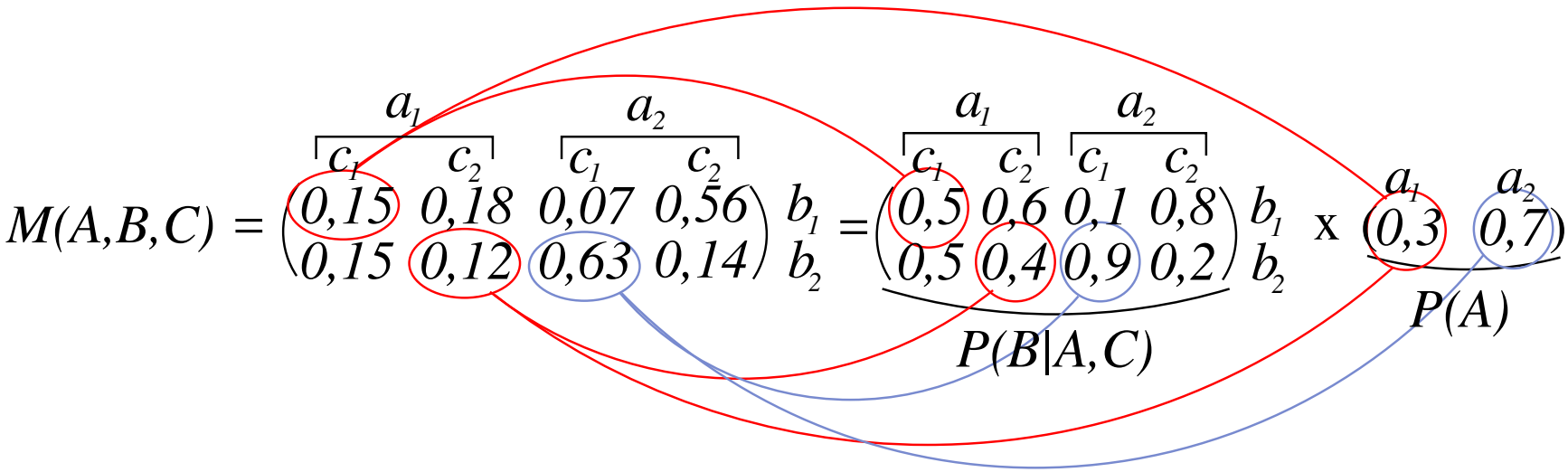
$$M_1 = M_2 = M_3 = M_4 = M_5 = M_6 = \begin{array}{c} e_1\ e_2\ e_3 \\ (1\ \ 1\ \ 1) \end{array}$$

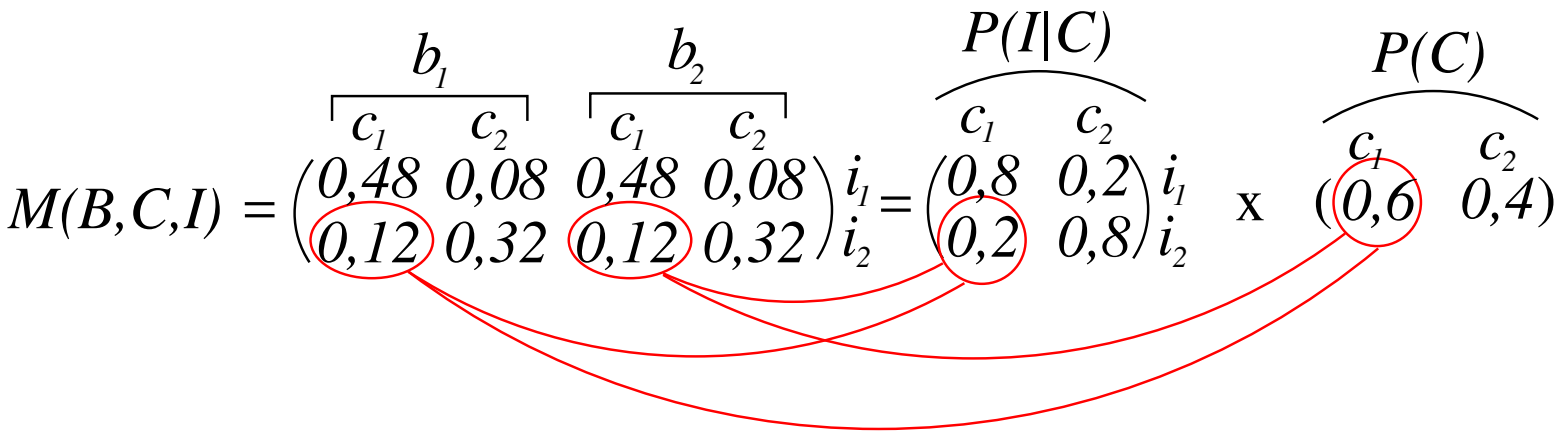
$$M_7 = M_8 = \begin{array}{c} d_1\ d_2 \\ (1\ \ 1) \end{array}$$

$$M_9 = M_{10} = \begin{array}{c} i_1\ i_2 \\ \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{array} \begin{array}{l} b_1 \\ b_2 \end{array}$$

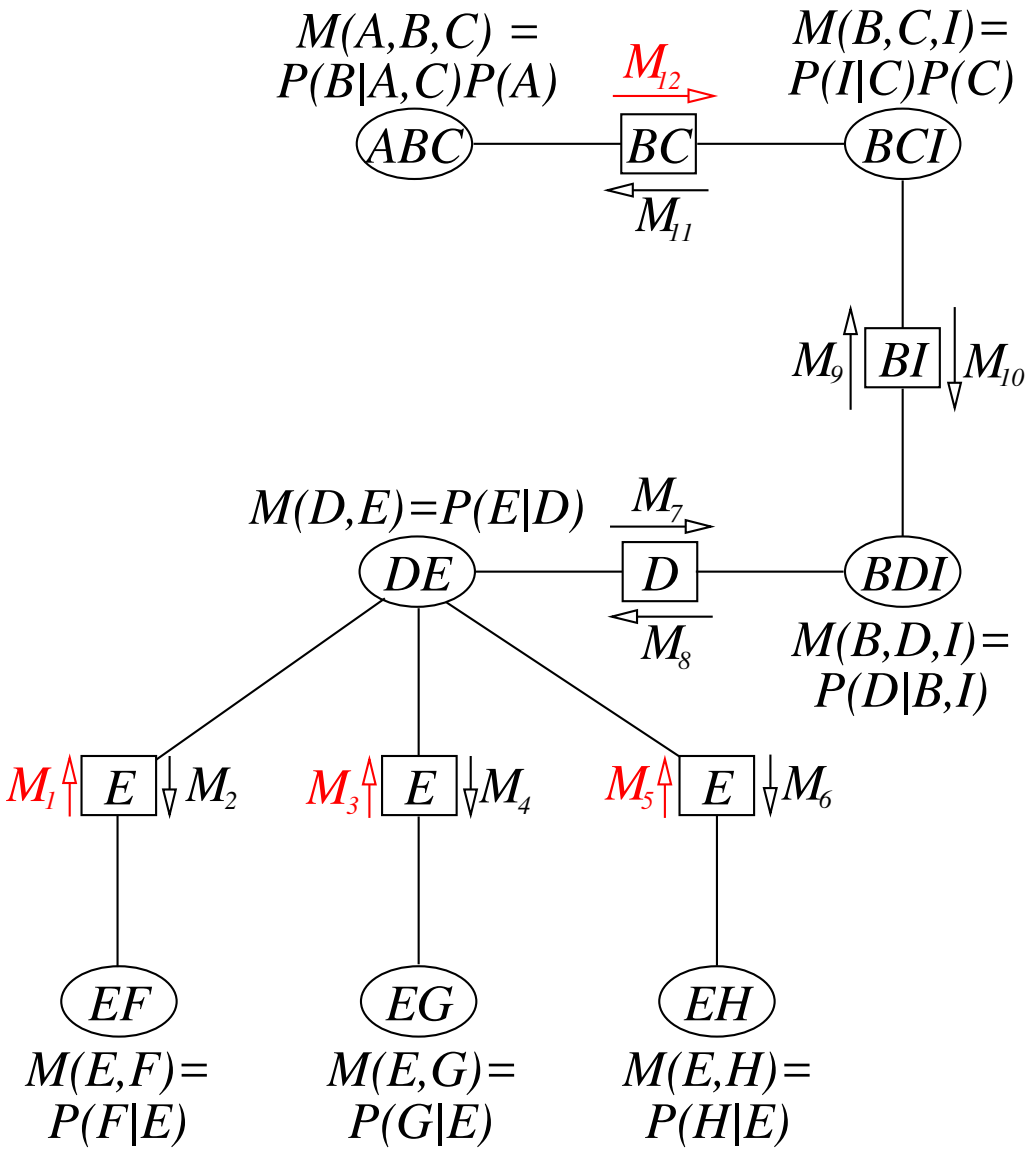
$$M_{11} = M_{12} = \begin{array}{c} c_1\ c_2 \\ \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{array} \begin{array}{l} b_1 \\ b_2 \end{array}$$

Rappel : dissection du produit de deux probabilités





Exemple à l'usage des étudiants studieux (5/9)



collecte: les feuilles envoient des messages

$$M_1 := \mathrm{Proj}_E M(E,F) = \mathrm{Proj}_E \begin{pmatrix} e_1 & e_2 & e_3 \\ 0{,}2 & 0{,}7 & 0{,}4 \\ 0{,}8 & 0{,}3 & 0{,}6 \end{pmatrix} \begin{matrix} f_1 \\ f_2 \end{matrix}$$

$$:= (1 \quad 1 \quad 1) \quad (e_1\ e_2\ e_3)$$

$$M_3 := \mathrm{Proj}_E M(E,G) = \mathrm{Proj}_E \begin{pmatrix} e_1 & e_2 & e_3 \\ 0{,}6 & 0{,}1 & 0{,}3 \\ 0{,}4 & 0{,}9 & 0{,}7 \end{pmatrix} \begin{matrix} g_1 \\ g_2 \end{matrix}$$

$$:= (1 \quad 1 \quad 1) \quad (e_1\ e_2\ e_3)$$

$$M_5 := \mathrm{Proj}_E M(E,H) = \mathrm{Proj}_E \begin{pmatrix} e_1 & e_2 & e_3 \\ 0{,}5 & 0{,}6 & 0{,}2 \\ 0{,}5 & 0{,}4 & 0{,}8 \end{pmatrix} \begin{matrix} h_1 \\ h_2 \end{matrix}$$

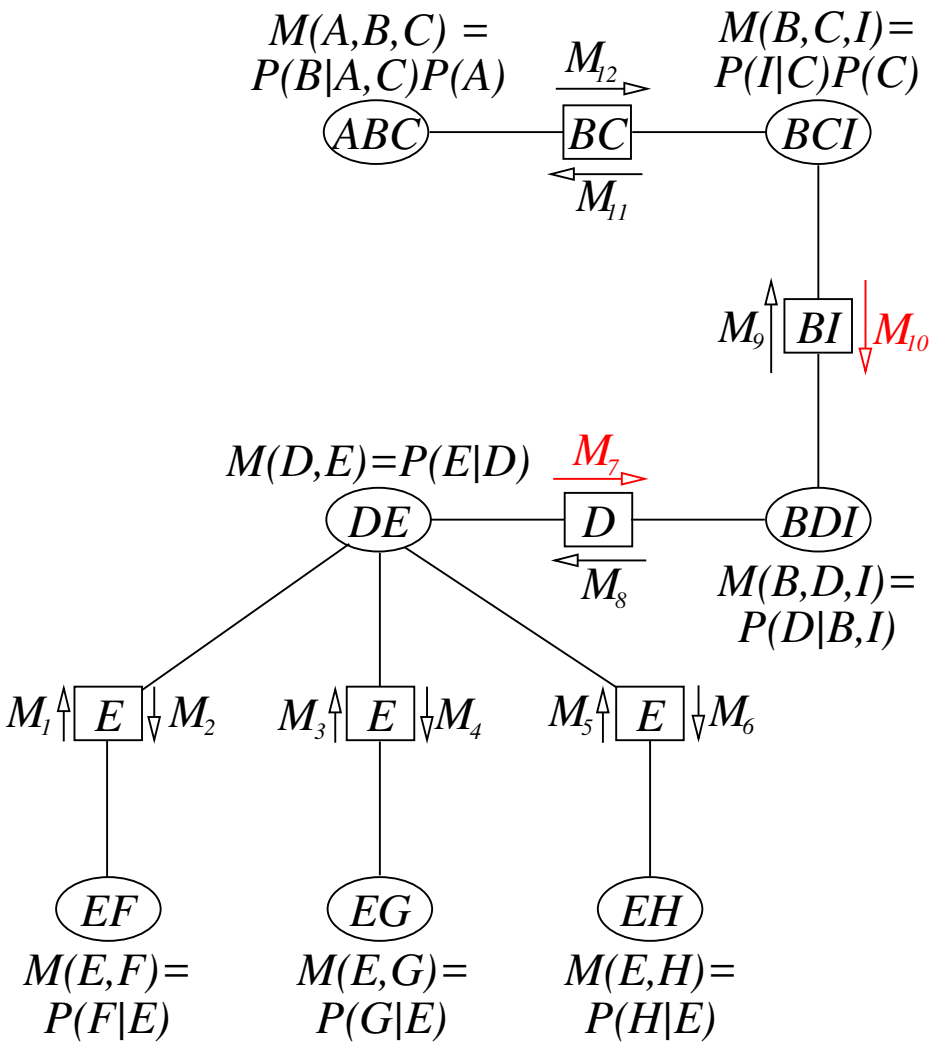
$$:= (1 \quad 1 \quad 1) \quad (e_1\ e_2\ e_3)$$

$$M_{12} := \mathrm{Proj}_{BC} M(A,B,C)$$

$$:= \mathrm{Proj}_{BC} \begin{pmatrix} \overbrace{c_1 \quad c_2}^{a_1} & \overbrace{c_1 \quad c_2}^{a_2} \\ 0{,}15 & 0{,}18 & 0{,}07 & 0{,}56 \\ 0{,}15 & 0{,}12 & 0{,}63 & 0{,}14 \end{pmatrix} \begin{matrix} b_1 \\ b_2 \end{matrix}$$

$$:= \begin{pmatrix} c_1 & c_2 \\ 0{,}22 & 0{,}74 \\ 0{,}78 & 0{,}26 \end{pmatrix} \begin{matrix} b_1 \\ b_2 \end{matrix}$$

Exemple à l'usage des étudiants studieux (6/9)



collecte (suite)

$$M_7 := \mathrm{Proj}_{D}\left(M(D,E) \times M_1 \times M_3 \times M_5 \right)$$

$$:= \mathrm{Proj}_{D} \begin{array}{c} \begin{matrix} d_1 & d_2 \end{matrix} \\ \begin{pmatrix} 0{,}4 & 0{,}3 \\ 0{,}5 & 0{,}4 \\ 0{,}1 & 0{,}3 \end{pmatrix} \end{array} \begin{matrix} e_1 \\ e_2 \\ e_3 \end{matrix} = \begin{matrix} d_1 & d_2 \end{matrix} \begin{pmatrix} 1 & 1 \end{pmatrix}$$

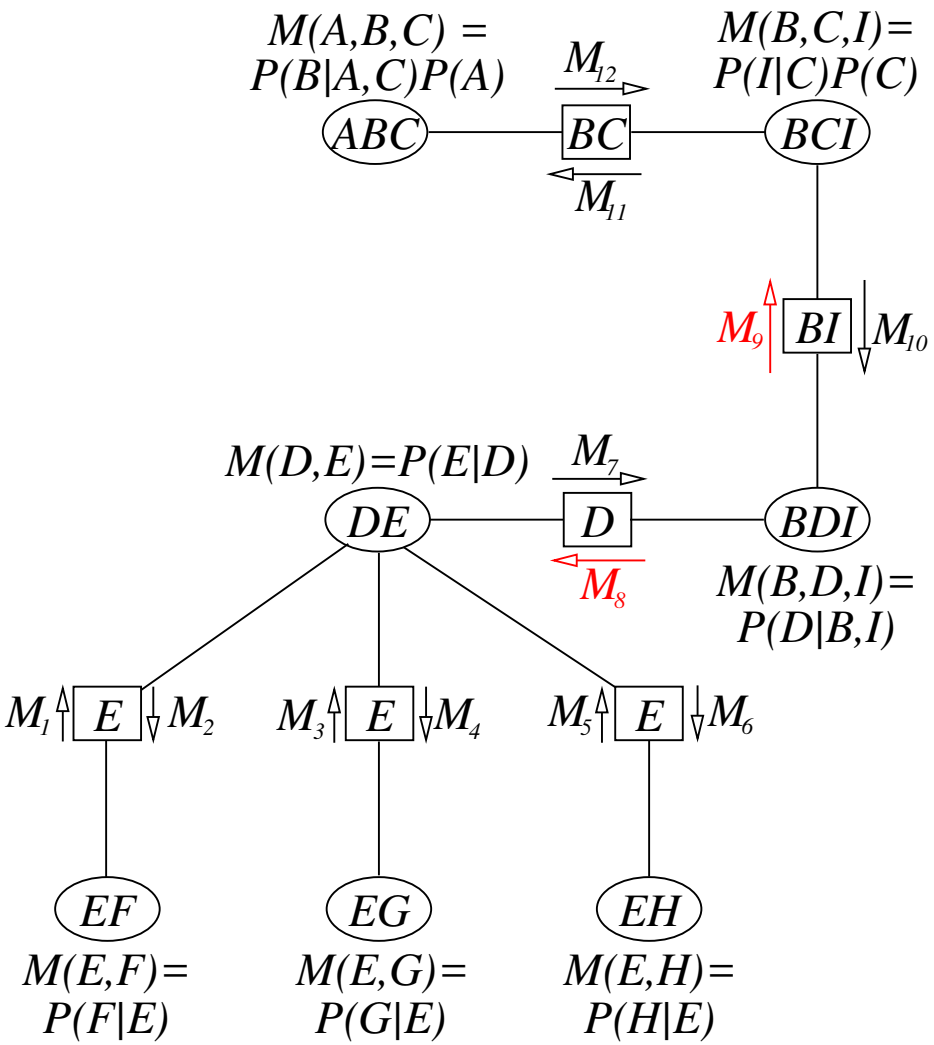
$$M_{10} := \mathrm{Proj}_{BI}\left(M(B,C,I) \times M_{12} \right)$$

$$:= \mathrm{Proj}_{BI} \left[\begin{array}{c} \overbrace{\begin{matrix} c_1 & c_2 \end{matrix}}^{b_1} \quad \overbrace{\begin{matrix} c_1 & c_2 \end{matrix}}^{b_2} \\ \begin{pmatrix} 0{,}48 & 0{,}08 & 0{,}48 & 0{,}08 \\ 0{,}12 & 0{,}32 & 0{,}12 & 0{,}32 \end{pmatrix} \end{array} \begin{matrix} i_1 \\ i_2 \end{matrix} \times \begin{array}{c} \begin{matrix} c_1 & c_2 \end{matrix} \\ \begin{pmatrix} 0{,}22 & 0{,}74 \\ 0{,}78 & 0{,}26 \end{pmatrix} \end{array} \begin{matrix} b_1 \\ b_2 \end{matrix} \right]$$

$$:= \mathrm{Proj}_{BI} \begin{array}{c} \overbrace{\begin{matrix} c_1 & c_2 \end{matrix}}^{b_1} \quad \overbrace{\begin{matrix} c_1 & c_2 \end{matrix}}^{b_2} \\ \begin{pmatrix} 0{,}1056 & 0{,}0592 & 0{,}3744 & 0{,}0208 \\ 0{,}0264 & 0{,}2368 & 0{,}0936 & 0{,}0832 \end{pmatrix} \end{array} \begin{matrix} i_1 \\ i_2 \end{matrix}$$

$$:= \begin{array}{c} \begin{matrix} b_1 & b_2 \end{matrix} \\ \begin{pmatrix} 0{,}1648 & 0{,}3952 \\ 0{,}2632 & 0{,}1746 \end{pmatrix} \end{array} \begin{matrix} i_1 \\ i_2 \end{matrix}$$

Exemple à l'usage des étudiants studieux (7/9)



début de la diffusion

$$M_8 := \mathrm{Proj}_{D}(M(B,D,I) \times M_{10})$$

$$:= \mathrm{Proj}_{D}\left[\begin{array}{cccc} \overbrace{i_1 & i_2}^{b_1} & \overbrace{i_1 & i_2}^{b_2} & \\ 0{,}1 & 0{,}4 & 0{,}7 & 0{,}8 \\ 0{,}9 & 0{,}6 & 0{,}3 & 0{,}2 \end{array}\begin{array}{c} d_1 \\ d_2 \end{array} \times \begin{array}{cc} b_1 & b_2 \\ 0{,}1648 & 0{,}3952 \\ 0{,}2632 & 0{,}1746 \end{array}\begin{array}{c} i_1 \\ i_2 \end{array}\right]$$

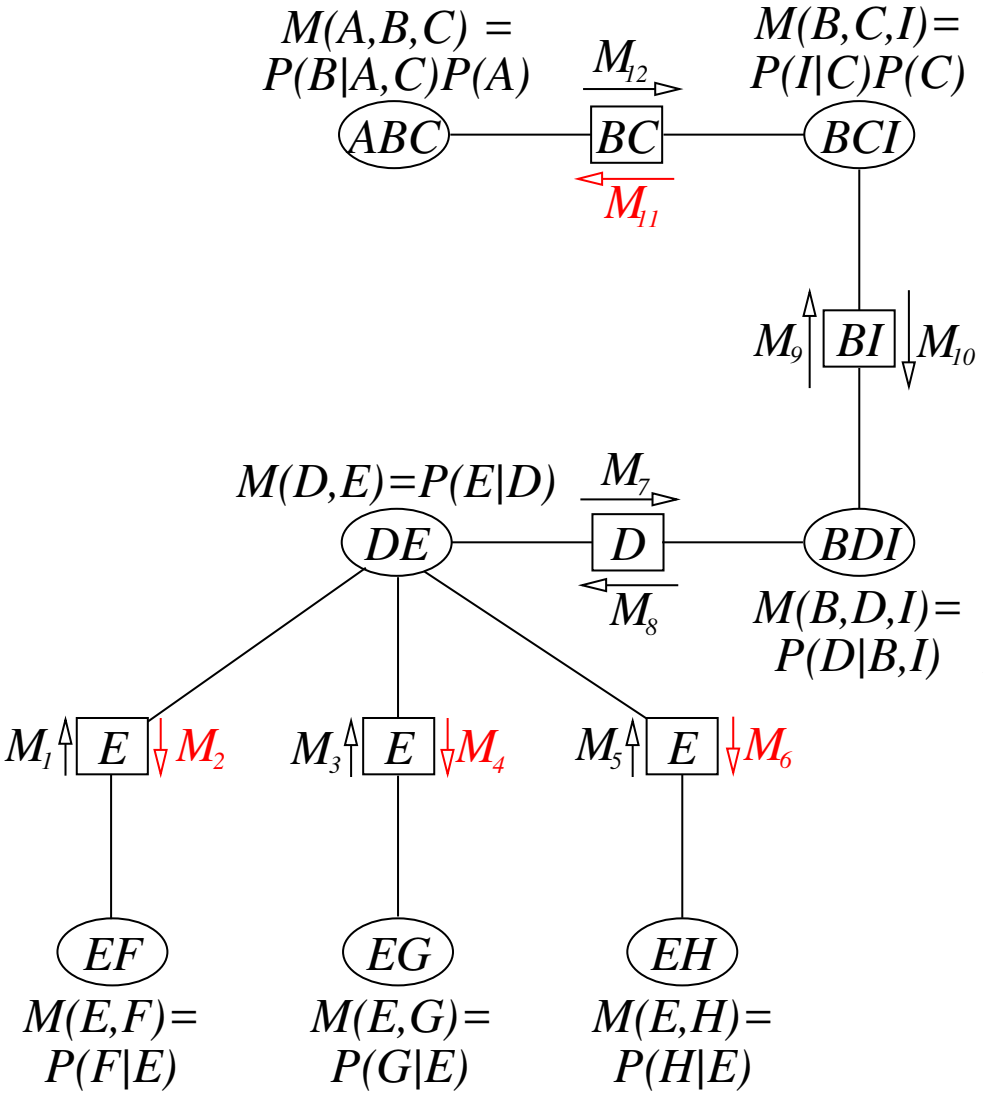
$$:= \mathrm{Proj}_{D}\begin{array}{cccc} \overbrace{i_1 & i_2}^{b_1} & \overbrace{i_1 & i_2}^{b_2} & \\ 0{,}01648 & 0{,}10528 & 0{,}27664 & 0{,}13968 \\ 0{,}14832 & 0{,}15792 & 0{,}11856 & 0{,}03492 \end{array}\begin{array}{c} d_1 \\ d_2 \end{array}$$

$$:= \begin{array}{c} 0{,}53808 \\ 0{,}45972 \end{array}\begin{array}{c} d_1 \\ d_2 \end{array}$$

$$M_9 := \mathrm{Proj}_{BI}(M(B,D,I) \times M_7)$$

$$:= \mathrm{Proj}_{BI}\left[\begin{array}{cccc} \overbrace{i_1 & i_2}^{b_1} & \overbrace{i_1 & i_2}^{b_2} & \\ 0{,}1 & 0{,}4 & 0{,}7 & 0{,}8 \\ 0{,}9 & 0{,}6 & 0{,}3 & 0{,}2 \end{array}\begin{array}{c} d_1 \\ d_2 \end{array} \times \begin{array}{cc} d_1 & d_2 \\ 1 & 1 \end{array}\right] = \begin{array}{cc} i_1 & i_2 \\ 1 & 1 \\ 1 & 1 \end{array}\begin{array}{c} b_1 \\ b_2 \end{array}$$

Exemple à l'usage des étudiants studieux (8/9)



fin de la diffusion

$$M_2 := \mathrm{Proj}_{E}(M(D,E) \times M_3 \times M_5 \times M_8)$$

$$:= \mathrm{Proj}_{E}\left[\begin{array}{c} \begin{array}{cc} d_1 & d_2 \end{array} \\ \begin{pmatrix} 0{,}4 & 0{,}3 \\ 0{,}5 & 0{,}4 \\ 0{,}1 & 0{,}3 \end{pmatrix} \begin{array}{c} e_1 \\ e_2 \\ e_3 \end{array} \end{array} \times \begin{array}{c} \\ \begin{pmatrix} 0{,}53808 \\ 0{,}45972 \end{pmatrix} \begin{array}{c} d_1 \\ d_2 \end{array} \end{array} \right]$$

$$:= \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\ := \begin{pmatrix} 0{,}35314 & 0{,}45292 & 0{,}19171 \end{pmatrix}$$

$$M_4 := \mathrm{Proj}_{E}(M(D,E) \times M_1 \times M_5 \times M_8) := M_2$$

$$M_6 := \mathrm{Proj}_{E}(M(D,E) \times M_1 \times M_3 \times M_8) := M_2$$

$$M_{11} := \mathrm{Proj}_{BC}(M(B,C,I) \times M_9)$$

$$:= \mathrm{Proj}_{BC}\left[\begin{array}{c} \overbrace{\begin{array}{cc} c_1 & c_2 \end{array}}^{b_1} \quad \overbrace{\begin{array}{cc} c_1 & c_2 \end{array}}^{b_2} \\ \begin{pmatrix} 0{,}48 & 0{,}08 & 0{,}48 & 0{,}08 \\ 0{,}12 & 0{,}32 & 0{,}12 & 0{,}32 \end{pmatrix} \begin{array}{c} i_1 \\ i_2 \end{array} \end{array} \times \begin{array}{c} \begin{array}{cc} i_1 & i_2 \end{array} \\ \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{array}{c} b_1 \\ b_2 \end{array} \end{array} \right]$$

$$:= \begin{array}{cc} c_1 & c_2 \end{array} \\ := \begin{pmatrix} 0{,}60 & 0{,}40 \\ 0{,}60 & 0{,}40 \end{pmatrix} \begin{array}{c} b_1 \\ b_2 \end{array}$$

Exemple à l'usage des étudiants studieux (9/9)

$$M_{11} \times M(A,B,C) = \begin{array}{c} \\ b_1 \\ b_2 \end{array}\!\!\begin{array}{c} \begin{array}{cc} c_1 & c_2 \end{array} \\ \begin{pmatrix} 0,60 & 0,40 \\ 0,60 & 0,40 \end{pmatrix} \end{array} \times \begin{array}{c} \begin{array}{cc} \multicolumn{2}{c}{a_1} & \multicolumn{2}{c}{a_2} \\ \hline c_1 & c_2 & c_1 & c_2 \end{array} \\ \begin{pmatrix} 0,15 & 0,18 & 0,07 & 0,56 \\ 0,15 & 0,12 & 0,63 & 0,14 \end{pmatrix} \end{array} \begin{array}{c} \\ b_1 \\ b_2 \end{array}$$

$$= \begin{array}{c} \\ b_1 \\ b_2 \end{array}\!\!\begin{array}{c} \begin{array}{cc} \multicolumn{2}{c}{a_1} & \multicolumn{2}{c}{a_2} \\ \hline c_1 & c_2 & c_1 & c_2 \end{array} \\ \begin{pmatrix} 0,090 & 0,072 & 0,042 & 0,224 \\ 0,090 & 0,048 & 0,378 & 0,056 \end{pmatrix} \end{array}$$

1. Petit exercice de calcul mental : calculez la projection de la matrice ci-dessus sur BC.
2. Petit exercice de déduction : à quoi correspond cette matrice ?
3. Ultime exercice à l'usage de l'élite (et des biens nantis) : calculez le produit $M_{11} \times M_{12}$.
4. Un dernier pour la route : dites ≪bizarre, bizarre, comme c'est bizarre≫, et expliquez pourquoi c'est bizarre.

2. Existe-t-il d'autres

algorithmes d'inférence ?

De Shafer-Shenoy à Lazy propagation

si $P(A,B,C,D,E) = P(A)P(B)P(C|A,B)P(D)P(E|C,D)$

alors $P(B,C,D,E) = \sum_A P(A)P(B)P(C|A,B)P(D)P(E|C,D)$

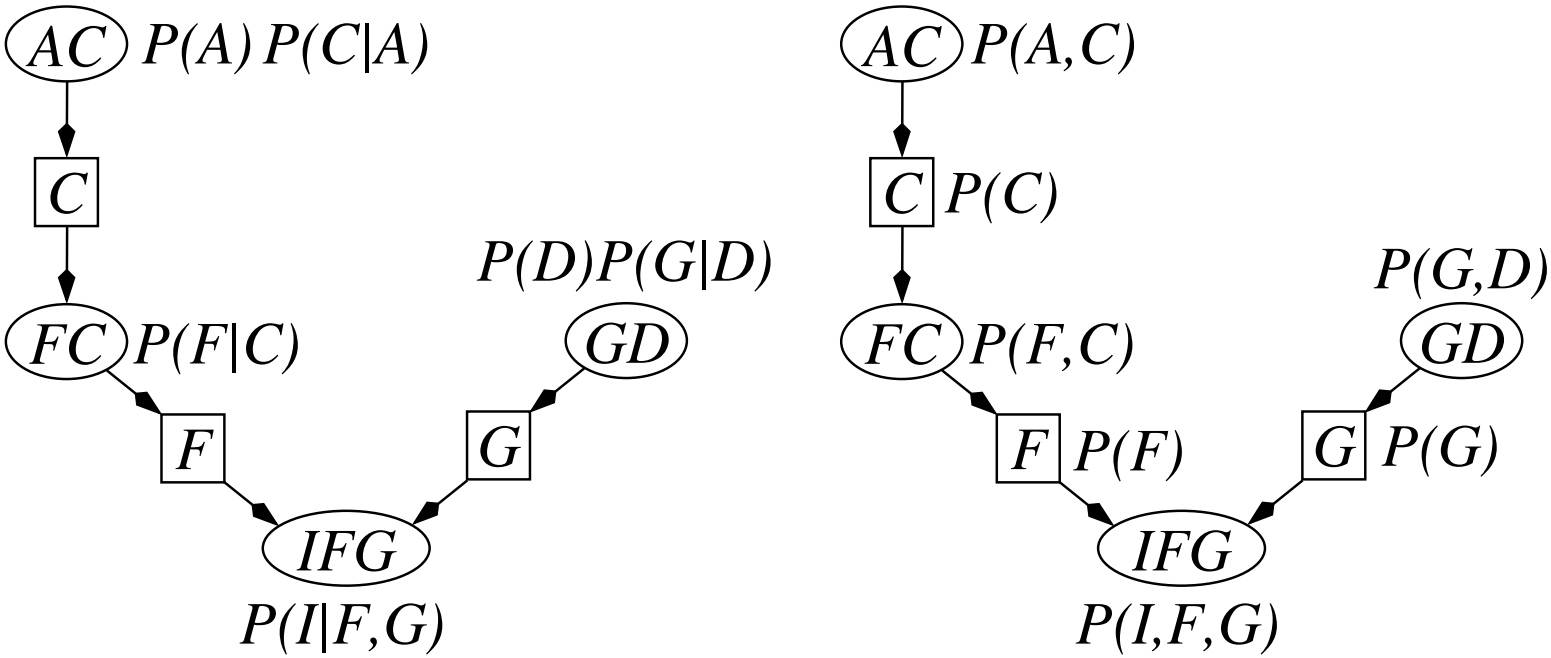
$$= \left(\sum_A P(A)P(C|A,B)\right) P(B)P(D)P(E|C,D)$$

$\Longrightarrow$ on a intérêt à ne pas effectuer les produits avant les sommes

Lazy propagation

Principe : garder les produits sous forme de listes et n'effectuer les multiplications que lorsque c'est nécessaire.

De Shafer-Shenoy à Jensen



Shafer-Shenoy : élimination de $A \Longrightarrow P(C, e_A) = \sum_A P(A, e_A) P(C|A)$

élimination de $F \Longrightarrow P(F, C, e_A) = P(F|C) P(C, e_A)$

Jensen : élimination de $F \Longrightarrow P(F, C, e_A) = \frac{P(F, C)}{P(C)} P(C, e_A)$

Petit guide sur Jensen et Lazy Propagation

Quelques références

1. S.L. Lauritzen, D.J. Spiegelhalter (1988) *Local compu- tations with probabilities on graphical structures and their application to expert systems(with discussion)*, Journal of the Royal Statistical Society, *Series B*, 50, pp.157–224.
2. F.V. Jensen, S.L. Lauritzen, K.G. Olesen (1990) *Bayesian Updating in Causal Probabilistic Networks by Local Computations*, Comp. Stat. Quarterly, 4, pp.269–282.
3. A.L. Madsen, F.V. Jensen (1998) *Lazy Propagation in Junction Trees*, Proceedings d'UAI-98.
4. A.L. Madsen, F.V. Jensen (1999) *Lazy Propagation : A Junction Tree Inference Algorithm Based on Lazy Evaluation*, Artificial Intelligence, 113, pp.203–245.

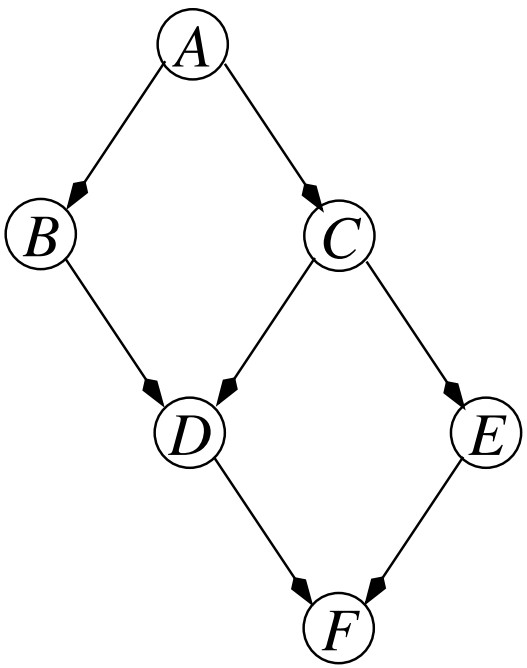
3. Comment construire l'arbre de jonction

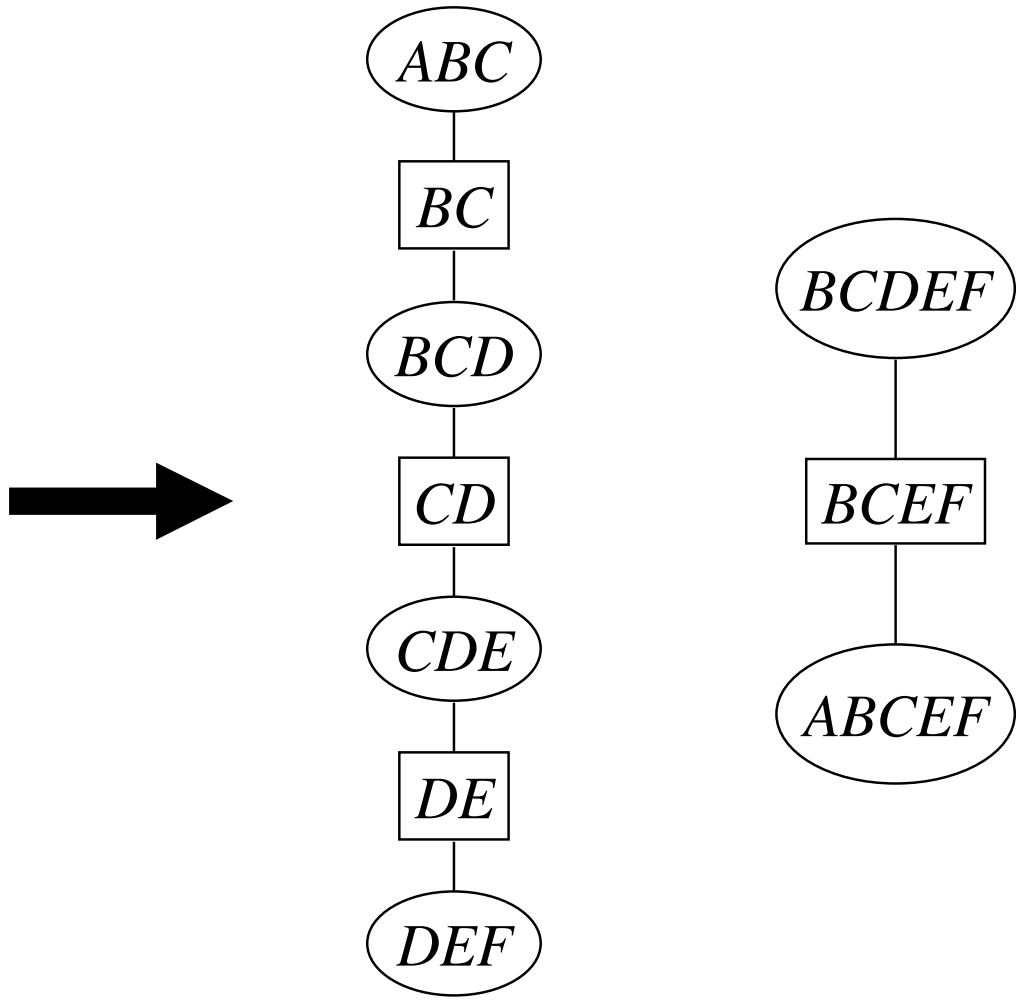
ou la triangulation par l'exemple

Toutes les séquences d'élimination ne sont pas égales

Séquence 1 : A, B, C, F, D, E

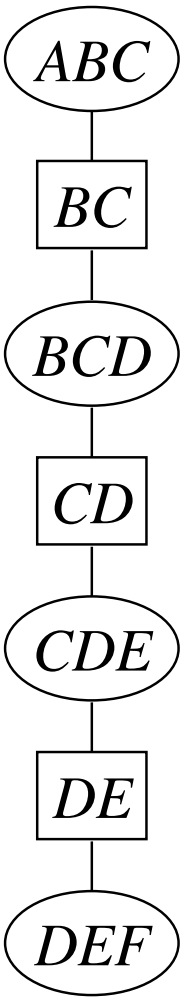
Séquence 2 : D, C, A, E, B, F





séquence 1 *séquence 2*

De la propriété d'intersection courante



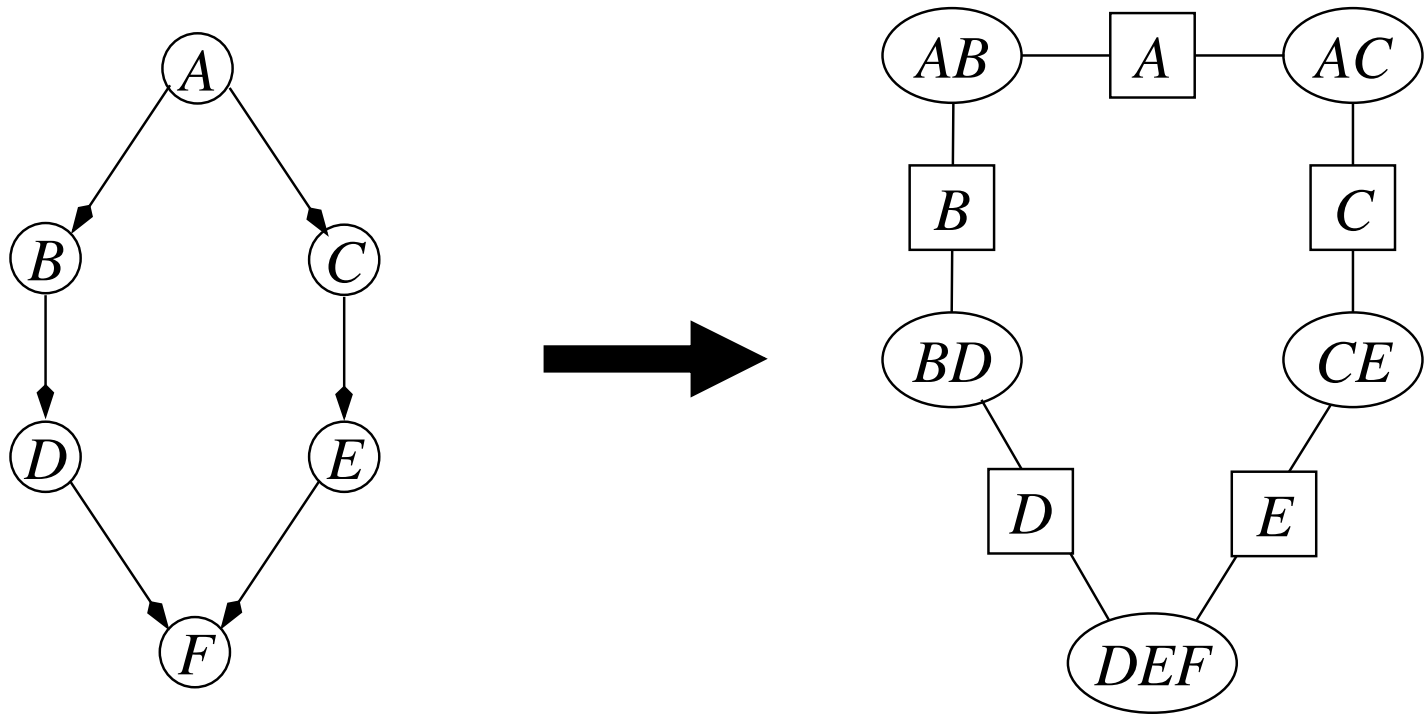
Propriété d'intersection courante

Soient C_1 et C_2 deux cliques quelconques de l'arbre de jonction et soit $S = C_1 \cap C_2 \neq \emptyset$. Alors sur toute chaîne reliant C_1 et C_2, les cliques et séparateurs contiennent S

Théorème

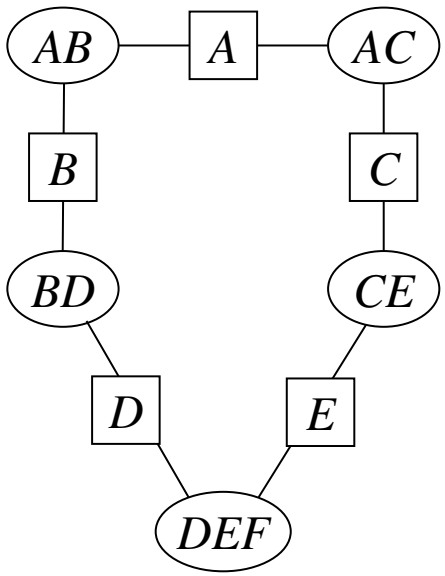
L'algorithme de Shafer-Shenoy fonctionne avec n'importe quel graphe sans cycle vérifiant la propriété d'intersection courante (ce que l'on appelle un *join tree*)

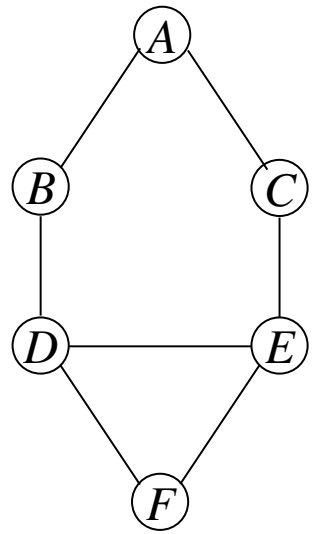
Un graphe de jonction avec cycles



Si l'on n'y prend garde, des cycles peuvent exister bien que la propriété d'intersection courante soit vérifiée

Graphe de jonction et triangulation (1/2)



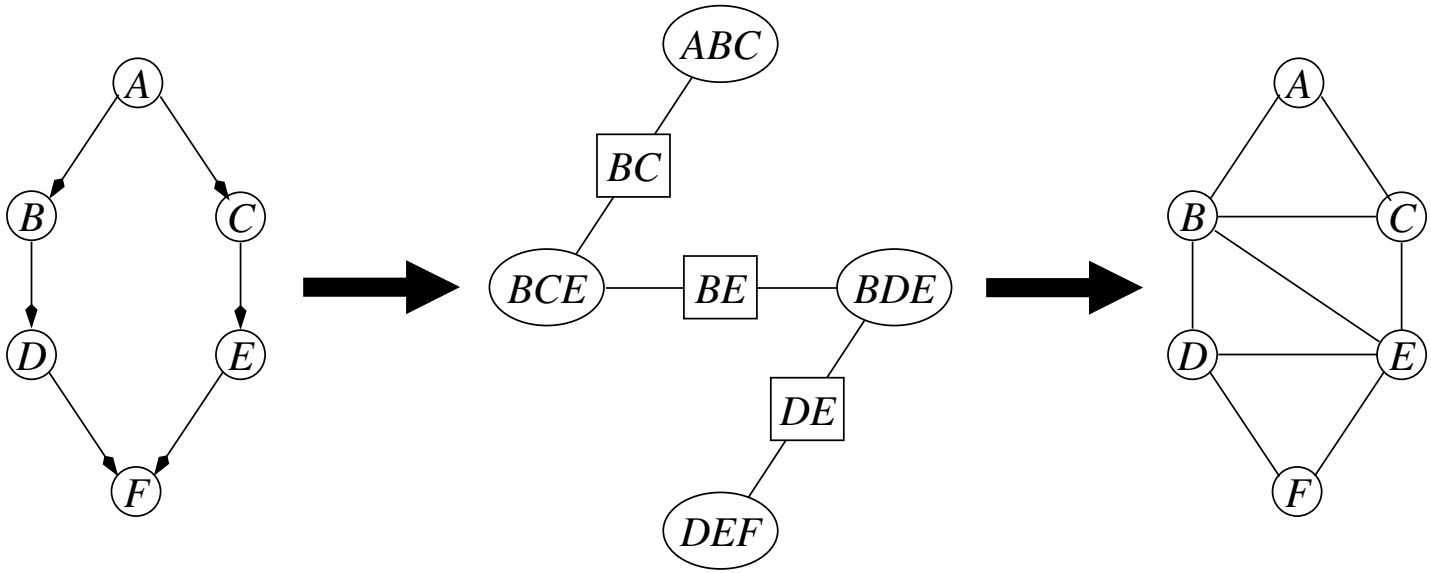


Triangulation

Un graphe non orienté est triangulé si et seulement si, pour tout cycle de longueur 4 ou plus, il existe une corde, c'est-à-dire une arête reliant deux nœuds non consécutifs du cycle

Exemple : le graphe ci-dessus n'est pas triangulé car le cycle A, B, D, E, C, A ne comporte pas de corde

Graphe de jonction et triangulation (2/2)



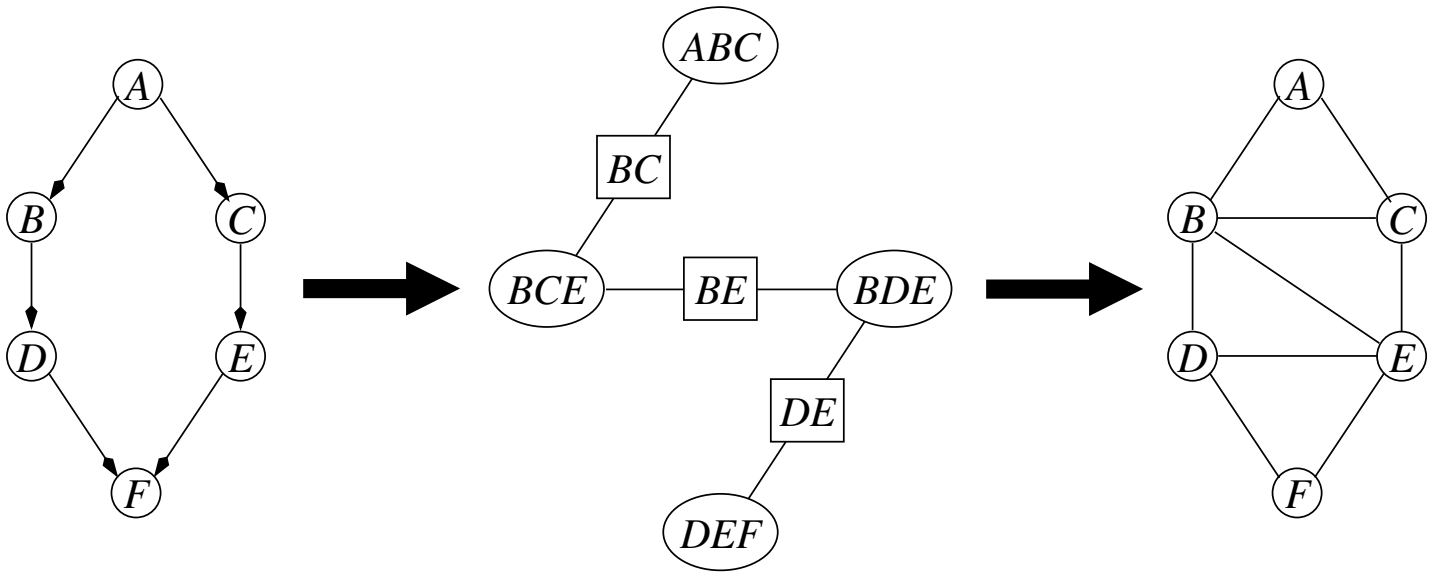
Proposition

il y a équivalence entre les deux assertions :

1. Le graphe de jonction est acyclique
2. Le graphe non orienté correspondant est triangulé

$\Longrightarrow$ pour trouver un «bon» arbre de jonction, il faut trouver une «bonne» triangulation

Un peu de morale, ça ne fait pas de mal



Moralisation

relier tous les parents d'un même nœud, puis supprimer les orientations $\Longrightarrow$ le graphe moral.

$\Longrightarrow$ les cliques pourront contenir l'ensemble des probabilités conditionnelles de la décomposition de la loi jointe

Recherche des triangulations optimales (1/2)

Proposition (Rose 1970)

Un graphe non orienté est triangulé si et seulement si l'application des deux règles suivantes permet d'éliminer tous les nœuds X_i du graphe sans rajouter une seule arête :

1. on rajoute des arêtes entre tous les voisins du nœud X_i que l'on veut éliminer (on forme une clique)
2. on supprime X_i ainsi que les arêtes qui lui sont adjacentes du graphe

$\Longrightarrow$ pour créer un join tree, il suffit de partir d'un graphe non orienté et d'appliquer, avec une certaine séquence d'élimination, les deux points ci-dessus

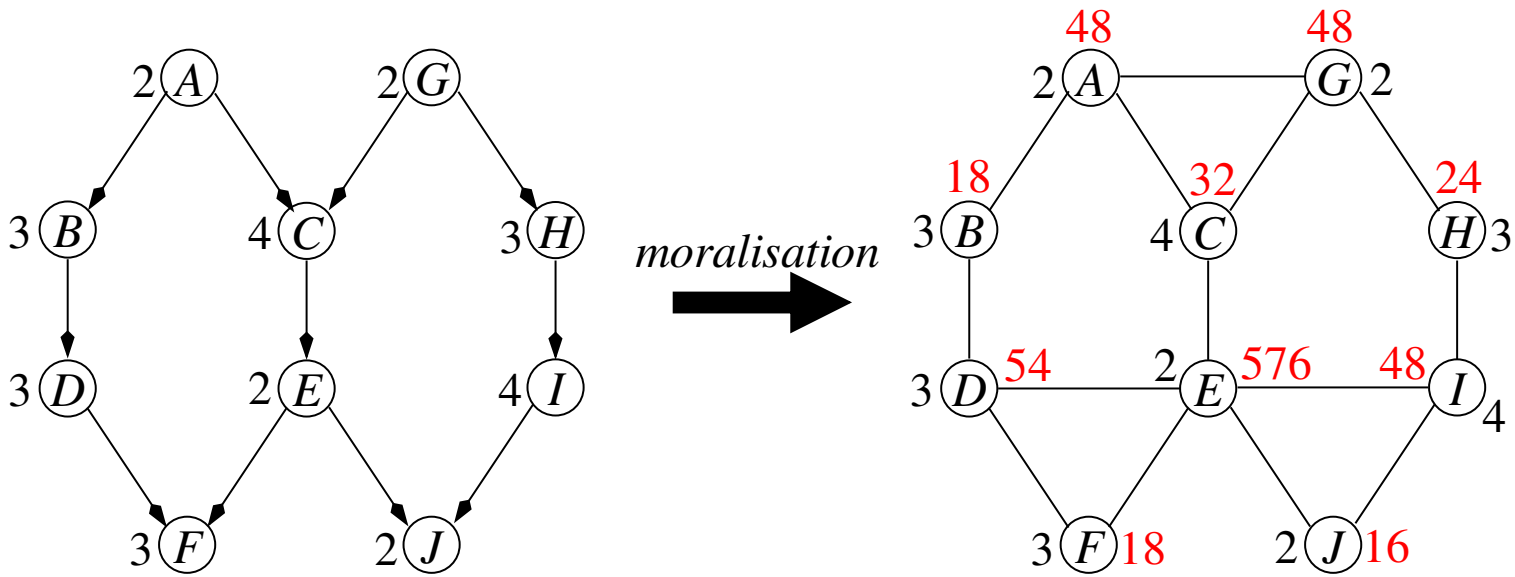
Mellouli (87) : Tout join tree «optimal» peut être construit à partir d'une séquence d'élimination

Recherche des triangulations optimales (2/2)

- *Arnborg et al. (87) :* trouver la triangulation optimale est NP-difficile $\Longrightarrow$ essayer de trouver des heuristiques
- *Kjærulff (90) :* un algorithme glouton rapide et efficace :
  Soit un graphe non orienté (moral) $G = (X, E)$, $X = \{X_1, \ldots, X_n\}$
  1. Associer à chaque X_i un ≪poids≫ égal au produit des modalités de X_i et de ses voisins
  2. éliminer le nœud X_i dont le poids est minimal (i.e., relier tous ses voisins de manière à former une clique C_i puis éliminer X_i et ses arêtes adjacentes)
  3. mettre à jour les poids des nœuds restants

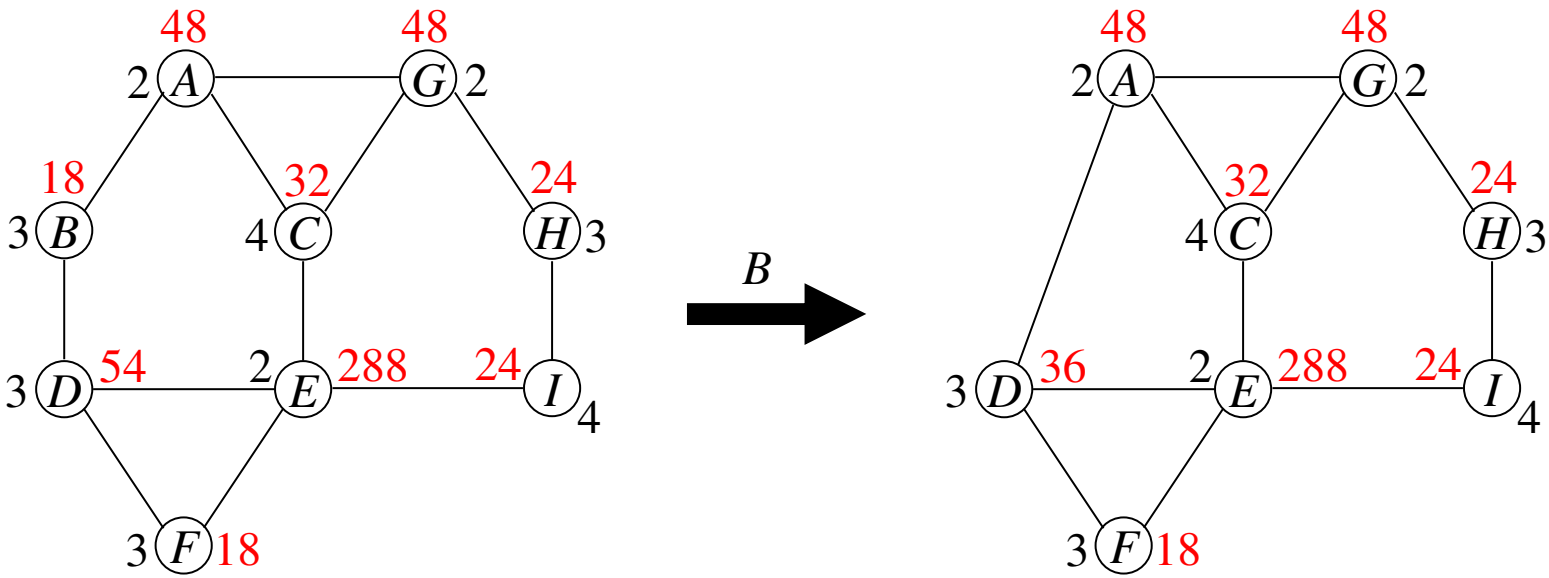
  $\Longrightarrow$ les C_i sont les cliques (ellipses) du join tree
- *van den Eijkhof & Bodlaender (2002) :* "safe reductions"
  $\Longrightarrow$ élimination de variables avec garantie d'optimalité
- Autres algorithmes : Becker & Geiger (96) ; Shoiket & Geiger (87)

Exemple de création de join tree (1/5)



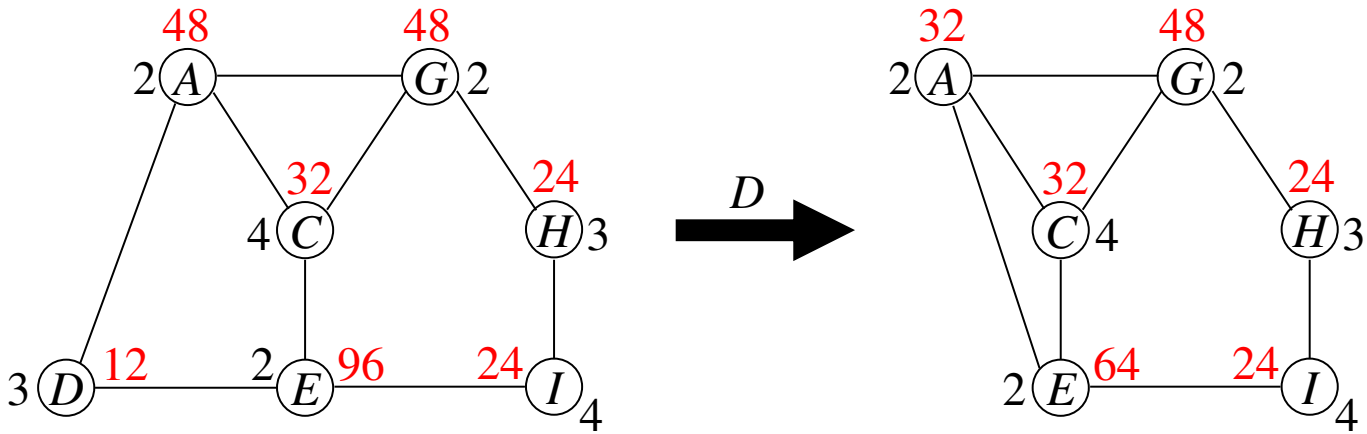
Variable à éliminer : $J \Longrightarrow$ clique EIJ

Exemple de création de join tree (2/5)



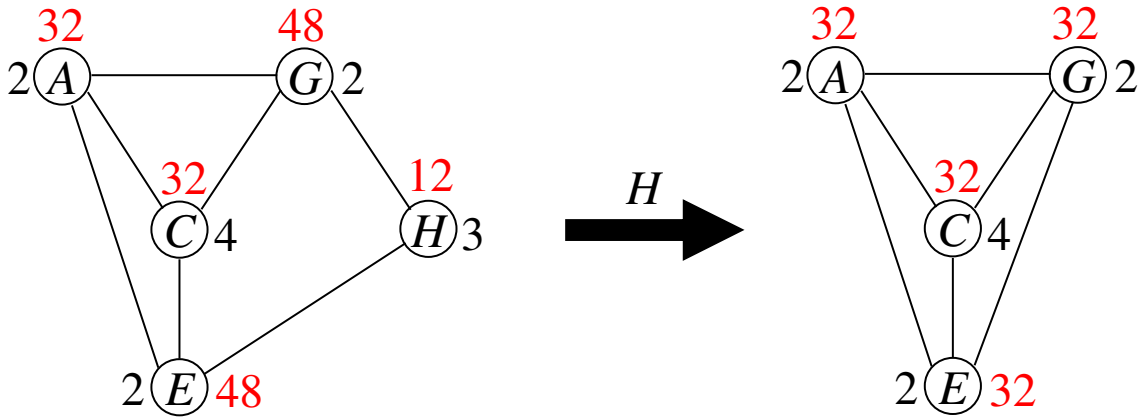
première variable à éliminer : $B \Longrightarrow$ clique ABD
deuxième variable à éliminer : $F \Longrightarrow$ clique DEF

Exemple de création de join tree (3/5)



première variable à éliminer : $D \Longrightarrow$ clique ADE
deuxième variable à éliminer : $I \Longrightarrow$ clique EHI

Exemple de création de join tree (4/5)

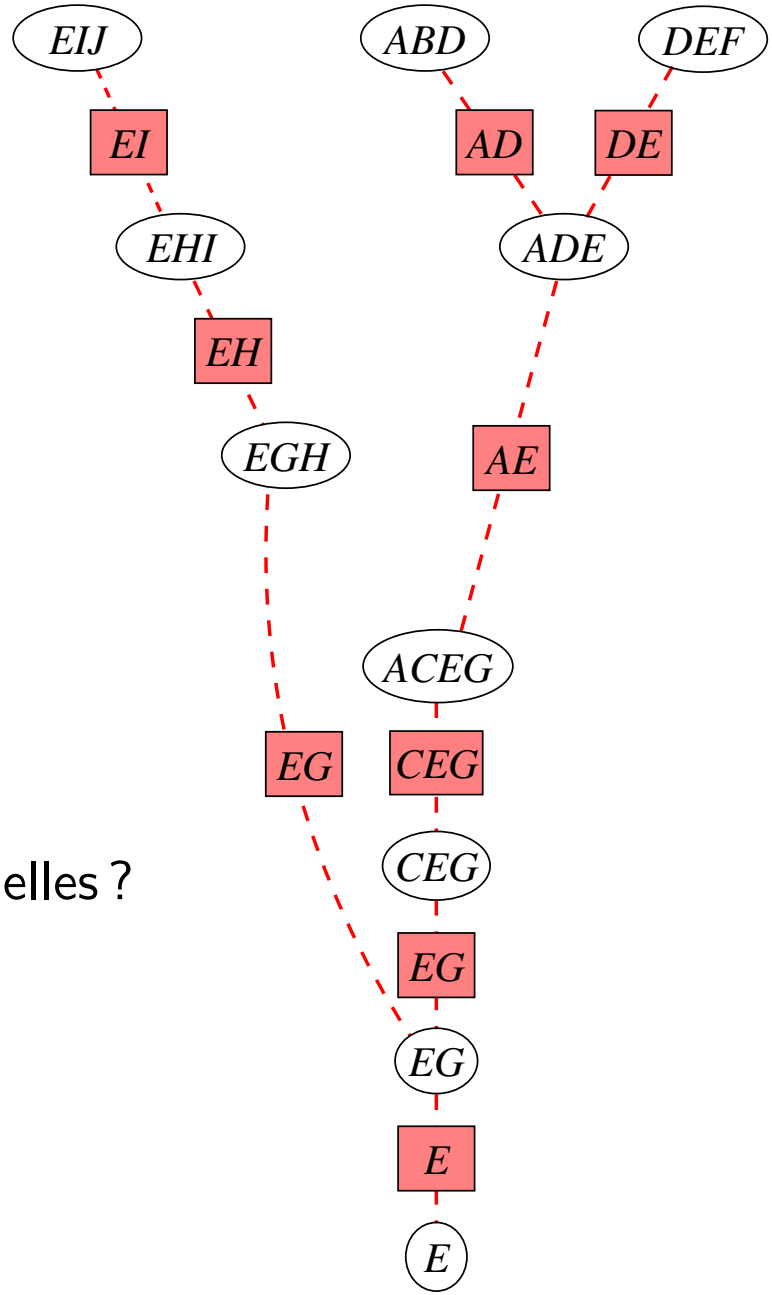


- première variable à éliminer : $H \Longrightarrow$ clique EGH
- deuxième variable à éliminer : $A \Longrightarrow$ clique $ACEG$
- puis les autres variables peuvent être éliminées dans n'importe quel ordre puisqu'elles appartiennent toutes à la même clique : $C \Longrightarrow$ clique CEG
  $G \Longrightarrow$ clique EG
  $E \Longrightarrow$ clique E

Exemple de création de join tree (5/5)

Ensemble des cliques selon leur ordre de création (avec la variable dont l'élimination a créé la clique) :
EIJ (*J*), *ABD* (*B*), *DEF* (*F*), *ADE* (*D*), *EHI* (*I*), *EGH* (*H*), *ACEG* (*A*), *CEG* (*C*), *EG* (*G*), *E* (*E*)

Problème : comment relier les cliques entre elles ?



Des cliques vers l'arbre d'élimination (1/2)

Définition de l'arbre d'élimination

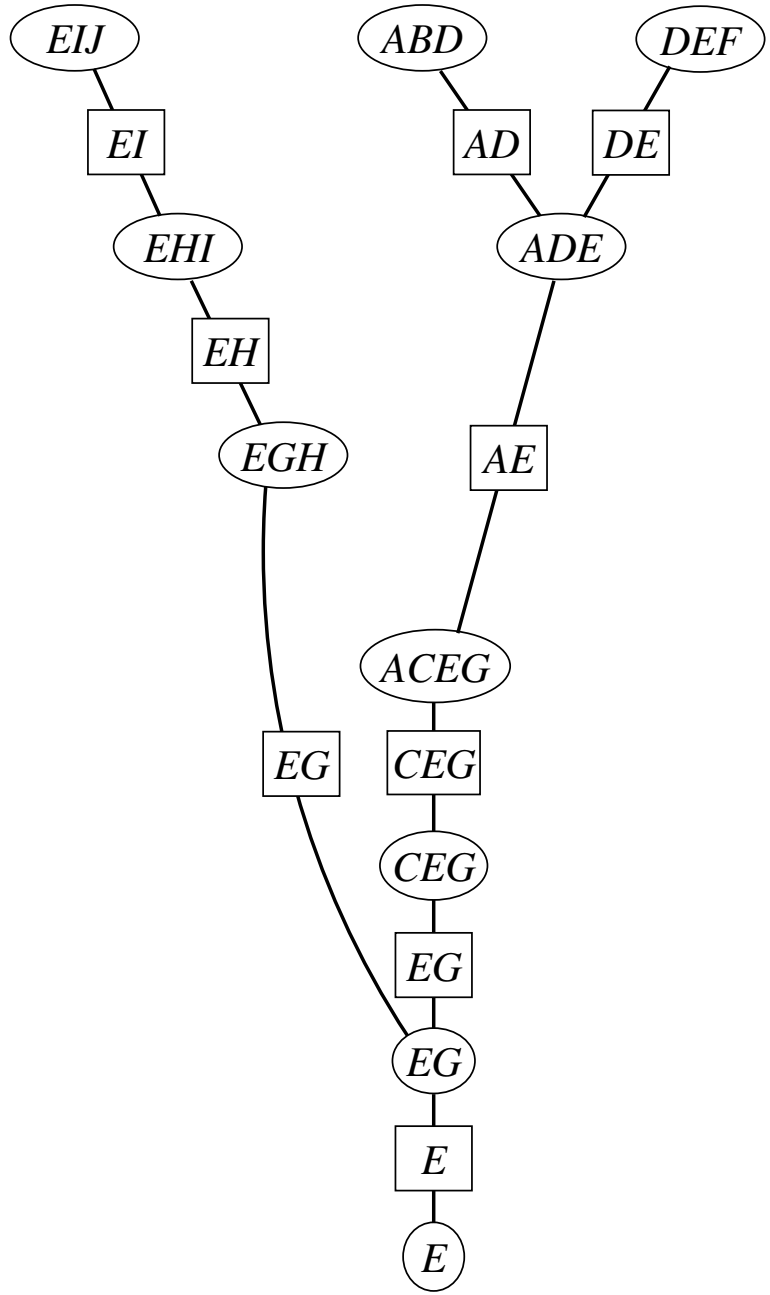
- Soit $\sigma : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$ la permutation telle que les variables X_i sont éliminées dans l'ordre $X_{\sigma(1)}, \ldots, X_{\sigma(n)}$
- Pour tout i, soit $D_{\sigma(i)}$ la clique créée au moment où $X_{\sigma(i)}$ est éliminée
- Arbre d'élimination : graphe $\mathcal{G} = (\mathcal{D}, \mathcal{E})$, où :
  - $\mathcal{D} = \{D_{\sigma(i)} : i \in \{1, \ldots, n\}\}$,
  - $\mathcal{E} = \{(D_{\sigma(i)}, D_{\sigma(j)}) :$
    $1 \leq i < n, j = min\{k \neq i : X_{\sigma(k)} \in D_{\sigma(i)}\}\}$

$\Longrightarrow$ Si l'on trie les nœuds X_i à l'intérieur des cliques selon leur ordre d'élimination, alors :
on relie $D_{\sigma(i)} = \{X_{\sigma(i)}, X_{\sigma(j)}, \ldots\}$ à $D_{\sigma(j)}$.

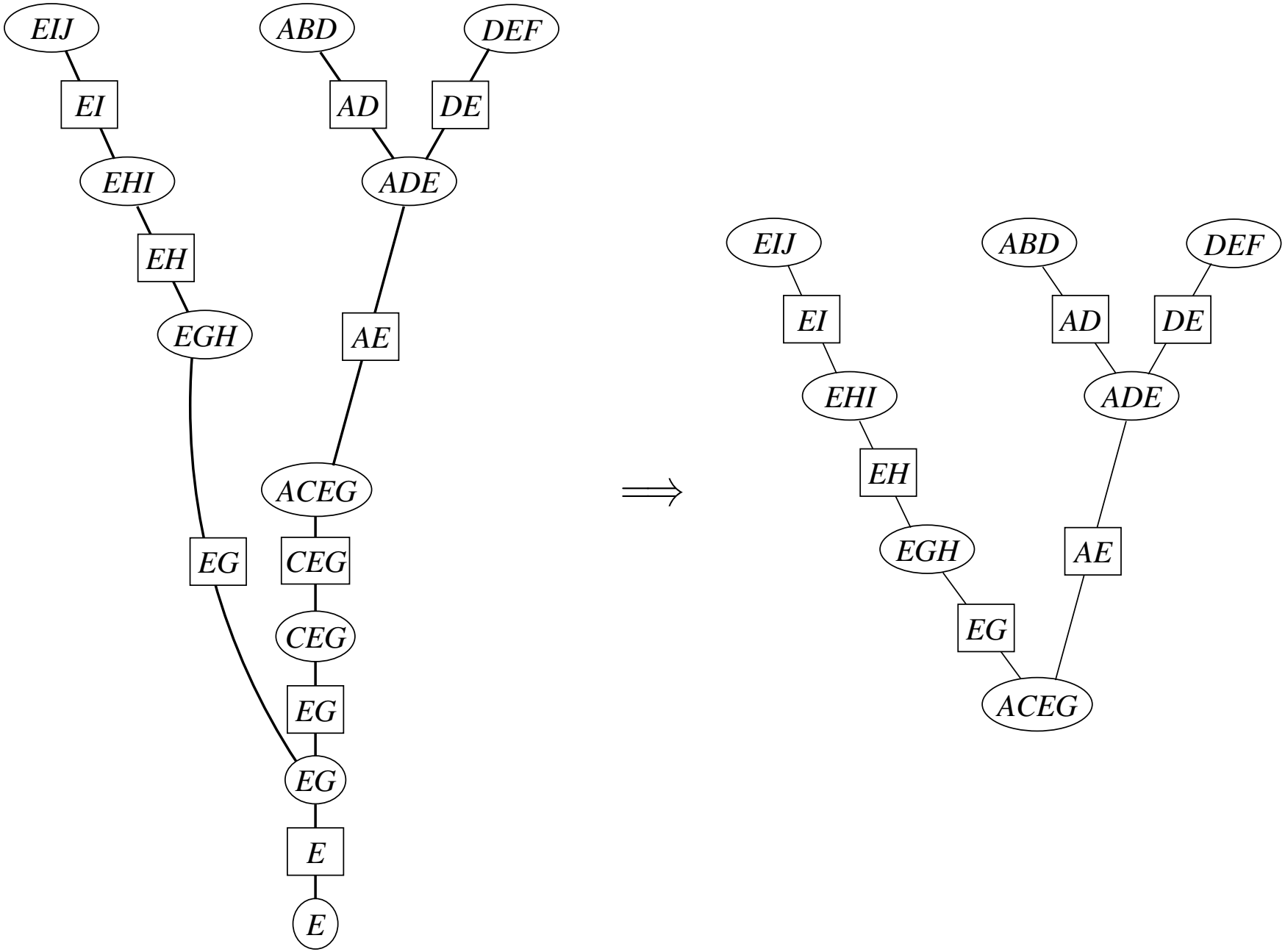
Des cliques vers l'arbre d'élimination (2/2)

- Arbre d'élimination : $\mathcal{G} = (\mathcal{D}, \mathcal{E})$, où :
  - $\mathcal{D} = \{D_{\sigma(i)} : i \in \{1, \ldots, n\}\}$,
  - $\mathcal{E} = \{(D_{\sigma(i)}, D_{\sigma(j)}) : 1 \leq i < n,$
    $j = min\{k \neq i : X_{\sigma(k)} \in D_{\sigma(i)}\}\}$

- Ensemble des cliques selon leur ordre de création (avec la variable dont l'élimination a créé la clique) :
  EIJ (*J*), *ABD* (*B*), *DEF* (*F*), *ADE* (*D*), *EHI* (*I*), *EGH* (*H*), *ACEG* (*A*), *CEG* (*C*), *EG* (*G*), *E* (*E*)



De l'arbre d'élimination vers l'arbre de jonction (1/2)



Propriétés de l'arbre d'élimination

Propriétés

$\mathcal{D} = \{D_{\sigma(i)} : i \in \{1, \ldots, n\}\}$,
$\mathcal{E} = \{(D_{\sigma(i)}, D_{\sigma(j)}) : 1 \leq i < n, j = min\{k \neq i : X_{\sigma(k)} \in D_{\sigma(i)}\}\}$

1. L'arbre d'élimination est un arbre
2. Il vérifie la propriété d'intersection courante
3. Soit $D_{\sigma(j)}$ un enfant de $D_{\sigma(i)}$, alors $|D_{\sigma(j)}| \geq |D_{\sigma(i)}| - 1$
4. Soient $D_{\sigma(i)}$ et $D_{\sigma(j)}$ les parents de $D_{\sigma(k)}$, alors $D_{\sigma(i)} \not\subset D_{\sigma(j)}$ et $D_{\sigma(j)} \not\subset D_{\sigma(i)}$
5. Soit $D_{\sigma(j)}$ un enfant de $D_{\sigma(i)}$, alors $D_{\sigma(j)} \subset D_{\sigma(i)} \iff |D_{\sigma(j)}| = |D_{\sigma(i)}| - 1$
6. Soit $D_{\sigma(j)}$ un enfant de $D_{\sigma(i)}$ tel que $D_{\sigma(j)} \not\subset D_{\sigma(i)}$, alors il n'existe pas d'ancêtre $D_{\sigma(k)}$ de $D_{\sigma(i)}$ tel que $D_{\sigma(j)} \subset D_{\sigma(k)}$

De l'arbre d'élimination vers l'arbre de jonction (2/2)

Algorithme pour obtenir un arbre de jonction

01 créer l'arbre d'élimination $\mathcal{G} = (\mathcal{D}, \mathcal{E})$
02 marquer à **false** tous les arcs de $\mathcal{E}$
03 **pour** i variant de n à 1 **faire**
04 **si** il existe $D_{\sigma(j)}$ parent de $D_{\sigma(i)}$ tel que l'arc $(D_{\sigma(j)}, D_{\sigma(i)})$ est non marqué et $|D_{\sigma(i)}| = |D_{\sigma(j)}| - 1$ **alors**
05 **pour** tous les autres parents $D_{\sigma(k)}$ de $D_{\sigma(i)}$ **faire**
06 créer dans $\mathcal{G}$ un arc $(D_{\sigma(k)}, D_{\sigma(j)})$
07 marquer cet arc à `true`
08 **fait**
09 **si** $D_{\sigma(i)}$ a un enfant $D_{\sigma(k)}$ **alors**
10 créer dans $\mathcal{G}$ un arc $(D_{\sigma(j)}, D_{\sigma(k)})$
11 **finsi**
12 supprimer $D_{\sigma(i)}$ ainsi que ses arcs adjacents
13 **fait**

A la fin de l'algorithme ci-dessus, $\mathcal{G}$ est un arbre de jonction.

Améliorations de l'algorithme d'élimination

- Au lieu de choisir le nœud à éliminer en fonction du poids, choisir, quand c'est possible, un nœud appartenant à une seule clique («simplicial rule», bodlaender (02)) ;
- Sous certaines contraintes, choisir un nœud presque simplicial (il manque une seule arête pour former une clique) ;
- Autres règles de réduction optimales (Buddies rule, Extended cube rule...) ;
- Suppression des arêtes de triangulation superflues (cf. Kjærulff (90)) — nécessite de recalculer la séquence d'élimination (par exemple par maximum cardinality search) ;
- Optimisation de l'arbre de jonction par modification des adjacences (cf. Jensen & Jensen (94)).

Références bibliographiques (1/2)

Quelques références

- Kjærulff, U (1990) *Triangulation of graphs – Algorithms giving small total state space*, technical report.
- Kjærulff, U (1991) *Optimal decomposition of probabilistic networks by simulated annealing*, Statistics and Computing, Vol 2, pp7-17.
- Becker, A & Geiger, D (1996) *A sufficiently fast algorithm for finding close to optimal junction trees*, Proceedings d'UAI-96.
- van den Eijkhof, F & Bodlaender A (2002) *Safe reduction rules for weighted treewidth*, Proceedings of the 28th International Workshop on Graph-Theoretic Concepts in Computer Science, Lecture Notes in Computer Science, vol 2573, pp176–185.

Références bibliographiques (2/2)

Quelques références

- Jensen, F.V. & Jensen, F. (1994) *Optimal junction trees* Proceedings d'UAI-94.
- Shoikhet, K & Geiger, D (1997) *Finding optimal triangulations via minimal vertex separators*, Proccedings de AAAI-97.
- Leimer, H.-G. (1993) *Optimal decomposition by clique separators*, Discrete Mathematics, vol 113, pp99-123.
- Olesen, K & Madsen, A (1999) *Maximal prime decomposition of Bayesian networks*, technical report.
- Flores, J & Gámez, J & Olesen, K (2003) *Incremental compilation of Bayesian networks*, Proceedings d'UAI-03.

CES – Data Scientist

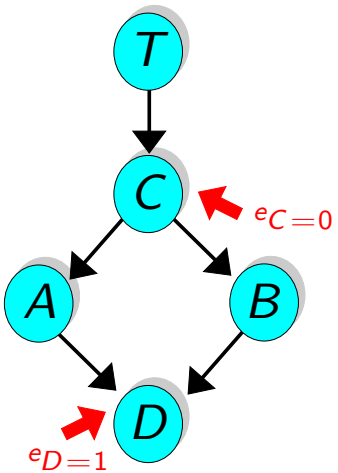
Inférence approchée dans les réseaux Bayésiens

Pierre-Henri WUILLEMIN

DESIR
LIP6
pierre-henri.wuillemin@lip6.fr

Limites de l'inférence exacte : calcul de $P(X \mid e)$

Les réseaux bayésiens sont un outil qui permet d'aggrandir de manière considérable la famille des loi jointes d'un grand nombre de variables aléatoires que l'on peut traiter (informatiquement).



Question : Y a-t-il une limite ?

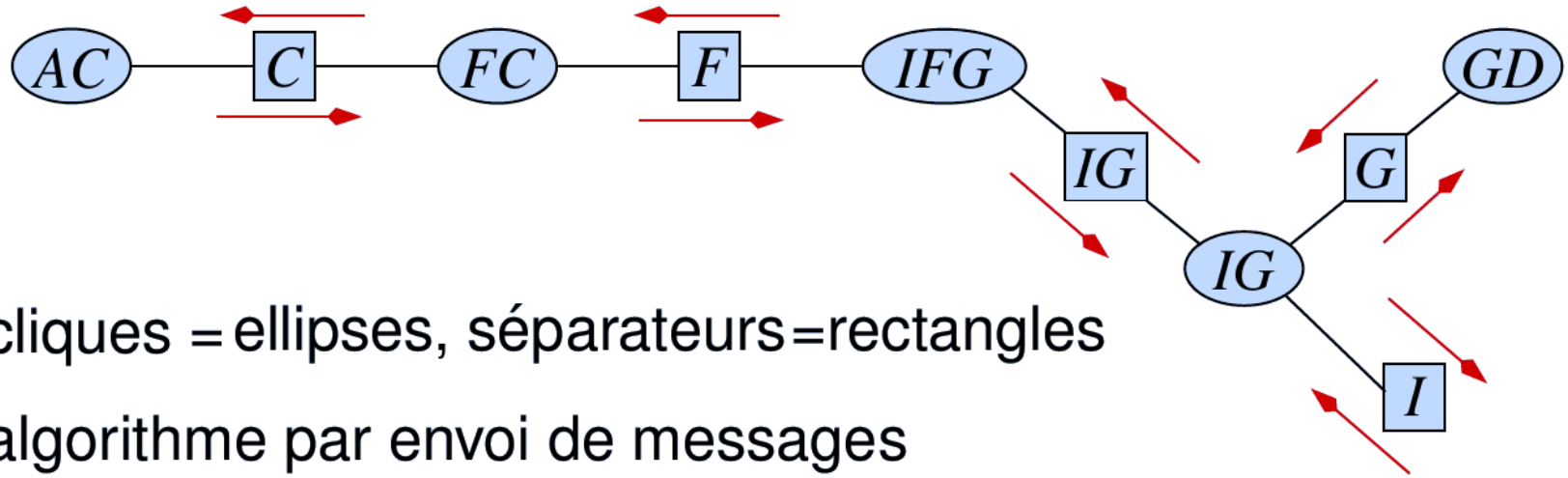
- Occupation mémoire : un nœud avec beaucoup de parents nécessite beaucoup de paramètres pour $P(X \mid \Pi_X)$.
- Temps de traitement : Peut-on toujours mener les calculs sur un BN ?

Complexité de l'inférence dans les BNs.

- L'inférence exacte dans un réseau bayésien est NP-difficile .
- L'inférence approchée dans un réseau bayésien est NP-difficile.

1. Cooper, G. F., 1990. *The computational complexity of probabilistic inference using Bayesian belief networks*. Artif. Intell. 42, 393-405.
2. Dagum, P. & Luby, M., 1993. *Approximating probabilistic inference in Bayesian belief networks is NP-hard*. Artif. Intell. 60, 141–153.

Limites de l'inférence exacte : calcul de $P(X \mid e)$



- cliques = ellipses, séparateurs=rectangles
- algorithme par envoi de messages

$$P(X) = \frac{1}{Z} \prod_{C \in JT} \Phi_C(C)$$

Complexité de l'inférence dans les BNs.

D'une manière générale, la complexité en espace et en temps d'un algorithme d'inférence exact est exponentielle en la *treewidth* du graphe : $O\left(n^2 \cdot e^{maxW}\right)$.

treewidth $\approx$ taille de la plus grande clique.

Limites de l'inférence exacte (3)

En pratique, il est extrêmement facile de construire des réseaux bayésiens qui ne seront pas traitables :

- Un nœud avec un grand nombre de parents.
- Un BN de très grande taille.
  - dynamic BN,
  - Enrichissement du language des BNs pour la représentation de la répétition de motifs : OOBN,
  - etc.
- Des variables aléatoires discrètes avec un grand nombre de valeurs possibles.
- etc.

Dans les applications, on rencontre facilement ce genre de BN. D'où la nécessité d'une inférence approchée.

Inférence approchée dans les BNs

Un algorithme de calcul approché fournit un solution raisonnable dans un temps raisonnable.

Inférences approchées dans les BNs

Pour les BNs, on peut distinguer deux familles principales de méthodes approchées :

- Simplification ou relaxation des algorithmes d'inférence exacte,
- Inférences basées sur la simulation.

Inférence approchée par simulation

Inférence approchée basée sur la simulation

Les inférences basées sur la simulation approchent la loi jointe recherchée par l'inférence grâce à la génération d'un grand nombre d'instances de cette loi. On appelle parfois ces instances des **particules**.

D'où deux grandes questions :

- Qu'est ce qu'on met exactement dans une particule ?
- Comment générer ces particules ?

Méthode de Monte Carlo

Définition (Simulation)

"step by step the probabilities of separate events are merged into a composite picture which gives an approximate but workable answer to the problem"

The Monte Carlo Method, D.D. McCracken, Scientific American, 1955

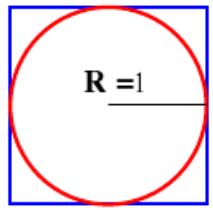
Monte Carlo *[1947 – von Neumann and Ulam (Los Alamos Scientific Laboratory)]*
Projet consistant à utiliser des nombres aléatoires pour simuler des séquences complexes d'évènements :

Simulation de la diffusion des neutrons dans un matériau fissile.

roulette : méthode bien connue de génération de nombres aléatoires.

Autres utilisations fréquentes : Aiguille de Buffon (1777), équations elliptiques ou paraboliques (diffusion), systèmes linéaires, optimisations (recuit), finance, go ...

Un exemple : approximation de π



Méthode : on jette des cailloux dans le carré.
Hypothèse : $NbJets \propto Surface$

d'où

$$\frac{NbJets_{Cercle}}{NbJets_{Total}} = \frac{\pi \cdot R^2}{(2 \cdot R)^2} = \frac{\pi}{4}$$

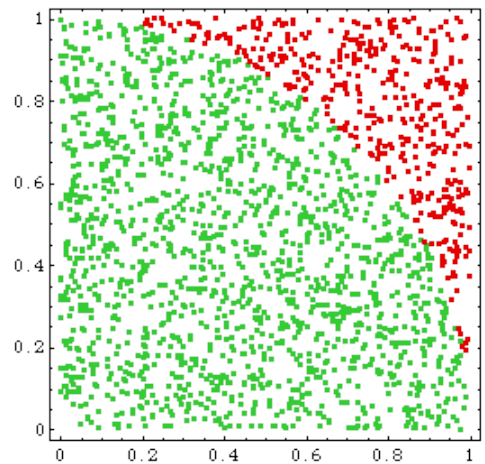
6 jets dans le cercle sur 10 jets en tout $\Rightarrow \hat{\pi}_{10} = 2.4$

89 jets dans le cercle sur 100 jets en tout $\Rightarrow \hat{\pi}_{100} = 3.57$

750 jets dans le cercle sur 1000 jets en tout $\Rightarrow \hat{\pi}_{1000} = 3$

7852 jets dans le cercle sur 10000 jets en tout $\Rightarrow \hat{\pi}_{10000} = 3.1408$

Autrement dit, on choisit aléatoirement un point du carré, et on vérifie (par *distance* ≤ 1) si il est dans le cercle. Puis on itère.



simulation avec 4000 jets

Un exemple : approximation de π - formalisation

En notant $(x_i, y_i)_{i \leq N}$ les positions des N jets successifs.

Soit la fonction indicatrice du disque dans le carré : $\mathbf{1}_{\bigcirc}(x, y) = \begin{cases} 1 \text{ si } x^2 + y^2 \leq 1 \\ 0 \text{ sinon} \end{cases}$

Le calcul précédent revient donc à calculer :

$$\frac{\sum_{i \leq N} \mathbf{1}_{\bigcirc}(x_i, y_i)}{N}$$

Qu'est-on en train d'estimer par une telle méthode ?

La solution idéale du problème serait de tester tous les points du carré pour connaître exactement la fraction de ceux-ci appartenant au cercle :

- Le "nombre" de point du carré : $\int_{(x,y)\in\square} dx\, dy$
- Le "nombre" de point du cercle : $\int_{(x,y)\in\square} \mathbf{1}_{\bigcirc}(x, y) dx\, dy$
- En introduisant une loi p uniforme sur $\square$ (changement de mesure) :
  $\int_{(x,y)\in\square} p(x, y) dx\, dy = 1$ et $\int_{(x,y)\in\square} \mathbf{1}_{\bigcirc}(x, y) p(x, y) dx\, dy$

On estime donc $\int_{(x,y)\in\square} \mathbf{1}_{\bigcirc}(x, y) p(x, y) dx\, dy$ par $\frac{\sum_i \mathbf{1}_{\bigcirc}(x_i, y_i)}{N}$.

Monte Carlo en statistique bayésienne

- Les méthodes de Monte Carlo proposent donc une simulation stochastique pour le calcul d'intégrales (ou d'équations différentielles).
- Il s'avère que l'intégration (ou l'équivalent discret : **la somme**) est une opération fondamentale dans les statistiques (et particulièrement dans la statistique bayésienne) à partir de :

$$posterior \propto L(ikelihood) \times P(rior)$$

- Calculer la constante de normalisation : $\int L \times P$ car $posterior = \frac{L \times P}{\int L \times P}$
- Marginaliser une distribution jointe : $P(x_2) = \int P(x_1, x_2) dx_1$
- Statistiques sur une distribution : $E_P(f) = \int f(x) P(x) dx$
  - Moyenne de P : $f(x) = x$
  - Moment d'ordre 2 de P : $f(x) = x^2$
  - $P(A)$: $f(X) = \mathbf{1}_A$

Synthèse et résultats théoriques sur Monte Carlo

Supposons que nous voulions calculer $\mu = E_P(f) = \int f(x)P(x)dx$.
S'il n'y a pas de résultats analytiques, la méthode de Monte Carlo propose d'utiliser une suite $(X_i)_{i \leq N}$ d'observations de variables aléatoires, i.i.d, suivant la loi P et d'estimer μ par :

$$\widehat{\mu}_N = \frac{1}{N} \sum_{i \leq N} f(X_i)$$

Loi forte des grands nombres

Si $E(|X|) < \infty$ alors
(presque sûrement)

$$\lim_{N \to \infty} \frac{1}{N} \sum_{i \leq N} X_i = E(X)$$

Théorème de la Limite Centrale

Soit $S_n = \sum_{i=1}^{n} X_i$ et $\mu_n = \frac{S_n}{n}$, avec les X_i v.a. indépendante, à variance finie. Alors

$$S_n \underset{n \to \infty}{\longrightarrow} \mathcal{N}\left(n \cdot \mu; n \cdot \sigma^2\right) \text{ ou } \frac{\mu_n - \mu}{\sigma/\sqrt{n}} \underset{n \to \infty}{\longrightarrow} \mathcal{N}(0;1)$$

Propriétés Monte Carlo

$$\widehat{\mu}_N \underset{N \to \infty}{\longrightarrow} \mu = \int f(x)P(x)dx \qquad \widehat{\sigma}_N^2 = \frac{1}{N-1} \sum_{i \leq N} \left[f(X_i) - \widehat{\mu}_N\right]^2 \qquad \frac{\widehat{\mu}_N - \mu}{\widehat{\sigma}_N/\sqrt{N}} \sim \mathcal{N}(0;1)$$

Monte Carlo : convergence

Estimation de la variance

$$s^2 = \frac{1}{M-1} \sum_{i=1}^{M} \left(f(x_i) - \hat{f}\right)^2 = \frac{\sigma^2}{M}$$

où σ^2 est la variance de P

Rappels : intervalles de confiance

- $[\hat{f} - s, \hat{f} + s]$: 66%
- $[\hat{f} - 2 \cdot s, \hat{f} + 2 \cdot s]$: 95%
- $[\hat{f} - 3 \cdot s, \hat{f} + 3 \cdot s]$: 99%

MonteCarlo dans les BNs : forward sampling

Dans un BN, on veut estimer $\forall i, P(X_i \mid e)$ à partir du BN et de e.

Les questions sont donc :

- Quelles particules pour calculer toutes ces distributions marginales ?
- Comment générer ces particules ?
- Combien en générer ?

Particules dans un BN

Afin de pouvoir calculer toutes les lois marginales dans un BN, une particule doit être une instance de l'ensemble des variables du BN.

On reconstruit une base de données à partir d'un BN : processus inverse de l'apprentissage.

Forward Sampling : génération des particules

Il est aisé de générer une particule (une valeur) d'une loi marginale :

Particule mono-dimensionnelle

Générer une particule pour une loi $P(X)$ consiste à tirer une valeur de X en suivant la distribution $P(X)$.

Comment faire pour un BN ?
Soit p la particule à générer et $p_{<X_i>}$ la valeur dans p de la variable X_i.

$p_{<X>}$ dans un BN

- pour les variables sans parent, la procédure ci-dessus permet de fournir certaines composantes de la particule.
- pour les variables avec parents, si $\forall X_j \in \Pi_X$, $p_{<X_j>}$ a déjà été tiré, il suffit de tirer $p_{<X>}$ suivant la loi $P(X \mid X_j = p_{<X_j>}, \forall X_j \in \Pi_X)$.

Génération d'une particule dans un BN

Générer une particule dans un BN revient donc à itérer la procédure ci-dessus sur l'ensemble des variables dans un ordre topologique.

Forward Sampling : calculer $P(X_i = x)$ par simulation

Inférence approché par forward sampling

Soit $(p_k)_{k \in D}$ l'ensemble des particules générés,

$$P(X_i = x) \approx \hat{P}_D(X_i = x) = \frac{1}{|D|} \sum_{k \in D} \mathbf{1}_{X_i = x}(p_k)$$

où $\mathbf{1}_{X_i = x}(p) = \begin{cases} 1 & \text{si} p_{<X_i>} = x \\ 0 & \text{sinon} \end{cases}$

Complexité de l'algorithme :

$$O(|D| \cdot |\text{BN}| \cdot (\max_{i \in |\text{BN}|} |\Pi_{X_i}| + \log \max_{i \in |\text{BN}|} |X_i|))$$

Forward Sampling : $|D|$ pour estimer $P(X_i = x)$?

Inégalité de Hoeffding [analyse de l'erreur absolue]

Soit $\mathcal{D} = \{X_1, \cdots, X_M\}$ M variables de Bernoulli indépendantes avec une même probabilité de succès p. Soit $T_{\mathcal{D}} = \frac{1}{M} \sum_m X_m$.

$$P(T_{\mathcal{D}} > p + \epsilon) \leq e^{-2M\epsilon^2}$$
$$P(T_{\mathcal{D}} < p - \epsilon) \leq e^{-2M\epsilon^2}$$

Pour le forward sampling,

$$P(|\hat{P}_D(X_i = x) - P(X_i = x)| > \epsilon) \leq 2e^{-2|M|\epsilon^2}$$

Estimation à ϵ près avec un degré de confiance $1 - \delta$

Il faut $2e^{-|M|\epsilon^2} < \delta$, soit :

$$M_a(\epsilon, \delta) \geq \frac{\ln \frac{2}{\delta}}{2\epsilon^2}$$

Forward Sampling : $|D|$ pour estimer $P(X_i = x)$? (2)

Inégalité de Chernoff [analyse de l'erreur relative]

Soit $\mathcal{D} = \{X_1, \cdots, X_M\}$ M variables de Bernoulli indépendantes avec une même probabilité de succès p. Soit $T_{\mathcal{D}} = \frac{1}{M} \sum_m X_m$.

$$P\left(T_{\mathcal{D}} > p(1+\epsilon)\right) \leq e^{\frac{-Mp\epsilon^2}{3}}$$
$$P\left(T_{\mathcal{D}} < p(1-\epsilon)\right) \leq e^{\frac{-Mp\epsilon^2}{3}}$$

Pour le forward sampling,

$$P\left(\hat{P}_D(X_i = x) \notin P(X_i = x) \cdot (1 \pm \epsilon)\right) \leq 2e^{-|M| \cdot P(X_i = x) \cdot \epsilon^2 / 3}$$

Estimation à ϵ près avec un degré de confiance $1 - \delta$

Il faut $2e^{-|M| \cdot P(X_i = x) \cdot \epsilon^2} < \delta$, soit :

$$M_r(\epsilon, \delta) \geq 3 \frac{\ln \frac{2}{\delta}}{P(X_i = x) \cdot \epsilon^2}$$

Forward Sampling : $P(X_i = x \mid e)$?

Première idée

$$P(X \mid e) = \frac{P(X, e)}{P(e)} \Rightarrow P(X \mid e) \approx \hat{P}_D(X \mid e) = \frac{\hat{P}_D(X, e)}{\hat{P}_D(e)}$$

Problème : $P(e)$ souvent petit : estimation difficile et demande une grande taille de D.

Seconde idée : *Rejection Sampling*

Générer des particules suivant $P(\cdot \mid e)$ plutôt que suivant $P(\cdot)$:

1. Générer une particule p comme dans le Forward Sampling
2. Si $p_{<e>}$ n'est pas compatible avec e, rejeter la particule

$$P(X_i = x \mid e) \approx \hat{P}_D(X_i = x \mid e) = \frac{1}{|D|} \sum_{k \in D} \mathbf{1}_{X_i = x}(p_k)$$

Problème : $P(e)$ souvent petit $\Rightarrow$ beaucoup de rejets : difficultés pour D de grande taille.

$P(X_i = x \mid e)$: *Likelihood Weighting*

Rejection sampling n'est vraiment pas efficace. Au lieu de rejeter tant de particules, pourquoi ne pas les pondérer par leur vraisemblance suivant e ?

Likelihood Weighting algorithm

1. Générer une particule p suivant Forward Sampling.
2. Forcer $p_{<e>} \leftarrow e$
3. Associer à p le poids $w_p = \prod_{e_i \in e} P(e_i \mid \Pi_{e_i})$

Les particules ne sont pas cohérentes !

$w_p \neq P(e \mid (p \setminus e))$

w_p est la probabilité du tirage successif des e_i dans le processus de *sampling*.

Estimation par *Likelihood Weighting*

$$P(X_i = x \mid e) \approx \hat{P}_D(X_i = x \mid e) = \frac{\sum_{k \in D} w_{p_k} \cdot \mathbf{1}_{X_i = x}(p_k)}{\sum_{k \in D} w_{p_k}}$$

C'est bien une généralisation du Forward Sampling si $e = \emptyset$.

Généralisons encore : Importance Sampling

$P(X_i = x)$: *Importance Sampling*

Rappel : On veut estimer $E_P(f)$ par $\frac{1}{M}\sum_m f(p_m)$ où les p_m sont des particules générées suivant la loi P.
Que faire si la loi P contient des "zones de raretés" rendant difficile l'estimation ?

Sampling distribution

Une loi Q est une loi d'échantillonnage pour $P \iff \forall x, P(x) > 0 \Rightarrow Q(x) > 0$

Comment utiliser Q pour établir une approximation de $E_P(f)$?

Unnormalized (ou Unweighting) Importance Sampling

$$E_P(f) = E_Q\left(f \cdot \frac{P}{Q}\right)$$

$$\text{alors } E_P(f) \approx \frac{1}{M}\sum_m f(p_m) \cdot \frac{P(p_m)}{Q(p_m)} \text{ où les } p_m \text{ sont générées suivant } Q$$

preuve : $E_Q(f\frac{P}{Q}) = \sum_x Q(x) \cdot f(x) \cdot \frac{P(x)}{Q(x)} = \sum_x f(x) \cdot P(x) = E_P(f)$

$P(X_i = x \mid e)$: *Importance Sampling* (2)

On utilise $E_P(f) = E_Q\left(f \cdot \frac{P}{Q}\right)$ pour échantillonner.

Problème : Si la loi cherchée est $P(X \mid e)$, on ne connait pas la loi ! Il faudrait la calculer pour $\frac{P}{Q}$ ...

Supposons $\tilde{P}$ connue vérifiant $P \propto \tilde{P}$ ($\exists Z, P = \frac{1}{Z}\tilde{P}$, $Z = \sum_x \tilde{P}(x)$).

$$E_P(f) = \sum_x Q(x) \cdot f(x) \cdot \frac{P(x)}{Q(x)} = \frac{1}{Z} \sum_x Q(x) \cdot f(x) \cdot \frac{\tilde{P}(x)}{Q(x)} = \frac{\sum_x Q(x) \cdot f(x) \cdot \frac{\tilde{P}(x)}{Q(x)}}{\sum_x \tilde{P}(x)} = \frac{\sum_x Q(x) \cdot f(x) \cdot \frac{\tilde{P}(x)}{Q(x)}}{\sum_x Q(x) \cdot \frac{\tilde{P}(x)}{Q(x)}}$$

$$E_P(f) = \frac{E_Q\left(f \frac{\tilde{P}}{Q}\right)}{E_Q\left(\frac{\tilde{P}}{Q}\right)}$$

Importance Sampling

En nommant $\omega(x) = \frac{\tilde{P}(x)}{Q(x)}$,

$$E_P(f) \approx \frac{\sum_m f(p_m) \cdot \omega(p_m)}{\sum_m \omega(p_m)} \text{ où les } p_m \text{ sont générées suivant } Q.$$

Importance sampling : calcul de petites probabilités

Soit $Z \sim \mathcal{N}(0,1)$ (loi normale centrée réduite,fonction de densité $\phi(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$), calculer :

$$p_a = P(Z \geq a) = \int_a^{+\infty} \phi(t)dt = E_Z(I_{(Z \geq a)})$$

Monte Carlo

Tirer $p_1, \cdots, p_M$ en suivant $\mathcal{N}(0,1)$ afin de calculer

$$\hat{p}_a = \frac{1}{M} \sum_i I_{(Z \geq a)}(p_i)$$

Pour $a = 5$, on a déjà : $p_a = 2.87 \cdot 10^{-7}$! Environ 1 sur 3.5 millions de $I_{(Z \geq a)}(p_i)$ vaut 1...

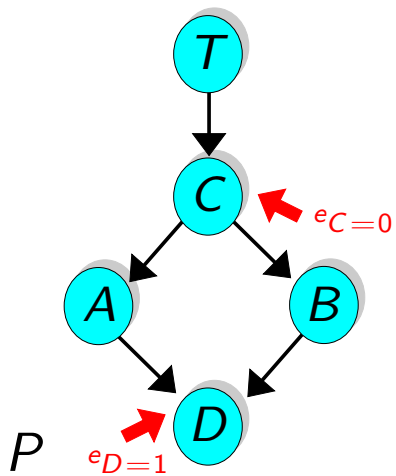
Importance sampling

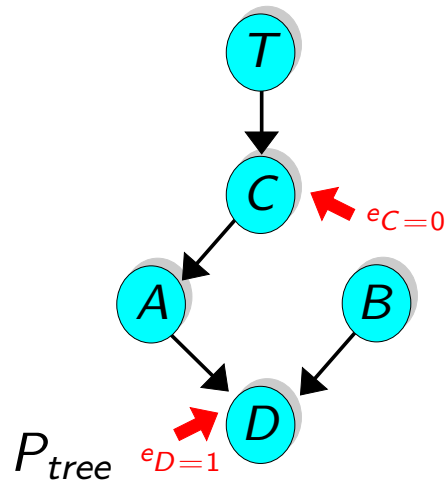
Tirer $p_1, \cdots, p_M$ en suivant $\mathcal{N}(a,1)$ afin de calculer

$$\hat{p}_a^{[IS]} = \frac{1}{M} \sum_i I_{(Z \geq a)}(p_i) \frac{\phi(p_i)}{\phi(p_i - a)}$$

$P(X_i = x \mid e)$: *Importance Sampling* (3)

$$E_P(f) \approx \frac{\sum_m f(p_m) \cdot \omega(p_m)}{\sum_m \omega(p_m)} \text{ où } \begin{cases} \text{les } p_m \text{ sont générées suivant } Q. \\ \omega(x) = \frac{\tilde{P}(x)}{Q(x)} \end{cases}$$



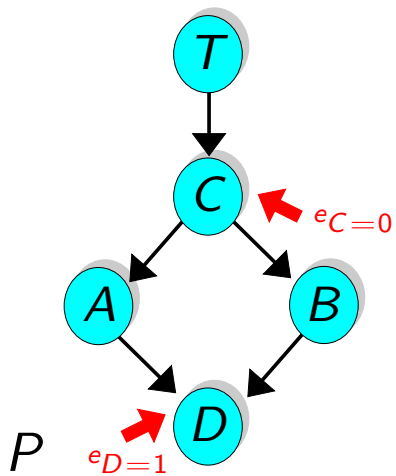


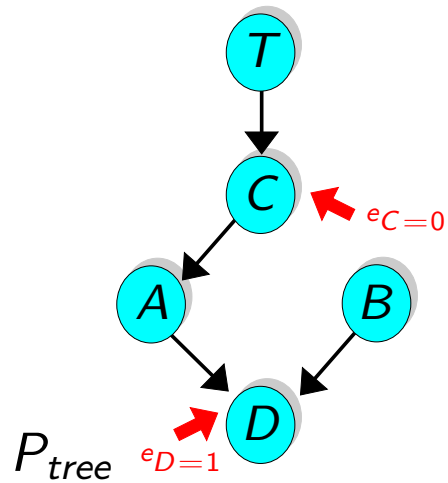
Importance sampling dans un réseau bayésien

- $Q(X) = P_{tree}(X \mid e)$ calculable en temps polynomial (Pearl, 88)
- $\tilde{P}(X) = P(X, e)$ donc $\tilde{P}(p_m)$ calculable en temps polynomial.

$P(X_i = x \mid e)$: *Importance Sampling* (3)

$$E_P(f) \approx \frac{\sum_m f(p_m) \cdot \omega(p_m)}{\sum_m \omega(p_m)} \text{ où } \begin{cases} \text{les } p_m \text{ sont générées suivant } Q. \\ \omega(x) = \frac{\tilde{P}(x)}{Q(x)} \end{cases}$$

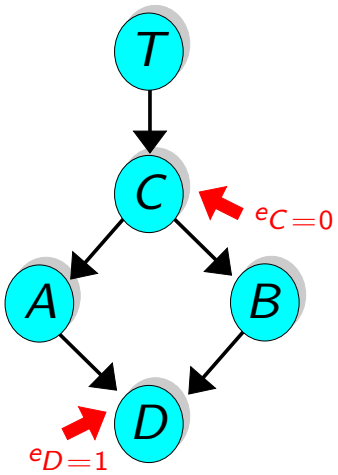




Importance sampling dans un réseau bayésien

- $Q(X) = P_{tree}(X \mid e)$ calculable en temps polynomial (Pearl, 88)
- $\tilde{P}(X) = P(X, e)$ donc $\tilde{P}(p_m)$ calculable en temps polynomial.

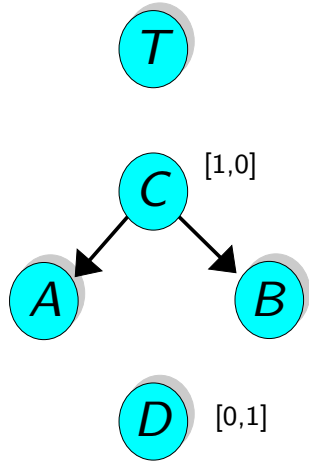
Likelihood Weighting vs. Importance Sampling



BN contextualisé

Soit un BN B et une information e, on appelle 'BN contextualisé' le BN B_e construit ainsi,

- $\forall X \in B, X \notin e, \Pi_{X[B_e]} = \Pi_{X[B]}$, de même loi.
- $\forall X \in B, X \in e, \Pi_{X[B_e]} = \emptyset$, $P_{B_e}(X) = \delta_e$.



Équivalence LW et IS

Pour calculer $P(X \mid e)$, l'algorithme LW correspond exactement à IS avec comme *Sampling Distribution* la loi du BN contextualisé par e.

Estimation de l'erreur

Pour estimer la probabilité $P(X_i = x \mid e)$ par LW pour une erreur relative ϵ avec un degré de confiance $1 - \delta$, il faut itérer tant que :

$$\sum_p w_p < \frac{4(1+\epsilon)}{\epsilon^2} \ln \frac{2}{\delta} \cdot u^{|e|}$$ où u est un majorant des paramètres du BN.

Limites des méthodes de MonteCarlo

Une limitation assez évidente de ces algorithmes est leur dépendance à la position des observations : une observation placée à une racine va être facilement prise en compte alors qu'une observation à une feuille sera beaucoup plus mal gérée. Ceci est principalement dû à la nécessité du tirage d'un échantillon dans un ordre topologique ... Comment faire autrement ?

Pour estimer $\mu = \int f(x)P(x)dx$, la méthode de Monte Carlo propose d'utiliser une suite $(X_i)_{i \leq N}$ d'observations de v.a., i.i.d, suivant la loi π et d'estimer μ par $\widehat{\mu}_N = \frac{1}{N} \sum_{i \leq N} f(X_i)$.

Que faire si il n'est pas possible d'obtenir des variables aléatoires i.i.d suivant π ?

Principe de MCMC : Monte Carlo Markov Chain

Il s'agirait de construire une suite (X_i) de variables aléatoires qui seraient (presque) i.i.d , suivant π. Il serait alors possible d'utiliser la méthode de Monte Carlo pour estimer μ.

Chaîne de Markov

Plus fréquemment qu'une relation fonctionnelle entre les (X_n), on peut étudier des relations d'indépendances conditionnelles entre ces différentes variables aléatoires.

Propriété de Markov

Un processus stochastique vérifie la propriété de Markov (d'ordre 1) si et seulement si :

$$P(X_n \mid X_0, \cdots, X_{n-1}) = P(X_n \mid X_{n-1})$$

(lorsque cette probabilité a un sens : i.e. $P(X_0, \cdots, X_{n-1}) > 0$)

Définition (Chaîne de Markov)

Une chaîne de Markov est un processus stochastique vérifiant la propriété de Markov (d'ordre 1).

Chaîne de Markov homogène

Propriété (Homogénéité)

Une chaîne de Markov est dite homogène si

$$\forall n > 0, P(X_n \mid X_{n-1}) = P(X_1 \mid X_0)$$

Définition (Probabilité et matrice de transition)

Soit $(X_n)_{n\in\mathbb{N}}$ une chaîne de Markov homogène, alors

- *la probabilité de transition de i à j est $p_{ij} = P(X_n = j \mid X_{n-1} = i)$*
- *la matrice de transition P est la matrice des $(p_{ij})_{i,j\in S}$ (si S est fini).*

Définition (Graphe de transition)

Si $(X_n)_{n\in\mathbb{N}}$ une chaîne de Markov homogène (S fini), alors
le graphe de transition est un graphe $G = (S, E)$ orienté qui vérifie :
$(i \to j) \in E \iff p_{ij} \neq 0$

Étude en régime permanent

Ce qui nous intéresse ici est le comportement de la chaîne de Markov si on laisse se dérouler le processus durant un temps très important.
Que peut-on dire de la position du système ? Suit-il une loi de probabilité particulière ?
En notant $\pi^{(n)}$ le vecteur de probabilité du système à l'instant n, on se rappelle que :

$$\pi^{(n+1)} = \pi^{(n)} \cdot P = \pi^{(0)} \cdot P^n$$

Définition (distribution de probabilité invariante)

Une distribution de probabilité est invariante pour la chaîne de Markov si et seulement si elle s'écrit comme le vecteur π et :

$$\pi = \pi \cdot P$$

i.e. : π est un vecteur propre de P^T pour la valeur propre 1

En supposant que $(\pi^{(n)})_{n\in\mathbb{N}}$ converge vers π^* alors :

$$\pi^* = lim_{n\to\infty}\pi^{(n)} = \pi^{(0)} \cdot lim_{n\to\infty}P^n = \pi^{(0)} \cdot P^*$$

Propriété

$(\pi^{(n)})_{n\in\mathbb{N}}$ converge vers π^* indépendamment de $\pi^{(0)}$ si et seulement si $lim_{n\to\infty}P^{(n)} = P^*$, matrice dont toutes les lignes sont égales entre elles (et égalent à π^*).

Ergodicité

Définition (Chaîne de Markov ergodique)

Une chaîne de Markov est ergodique si et seulement si elle est irréductible, apériodique et récurente positive.

Théorème (théorème ergodique)

Une chaîne de Markov ergodique est telle que $(\pi^{(n)})_{n\in\mathbb{N}}$ *converge, quelque soit* $\pi^{(0)}$*, vers* π^* *vérifiant :*

$$\begin{cases} \pi^* \cdot P = \pi^* \\ \pi^* \cdot 1 = 1 \end{cases}$$

De plus, $$\pi^*_j = \frac{1}{M_j}$$

Autrement dit, la proportion des instants où la chaîne se trouve dans l'état j *tend vers* π^*_j *avec probabilité 1. Pour presque toutes les trajectoires,* **la moyenne temporelle est identique à la moyenne spatiale**.

Limites de Monte Carlo et solution

Pour estimer $\mu = \int f(x)P(x)dx$, la méthode de Monte Carlo propose d'utiliser une suite $(X_i)_{i \leq N}$ d'observations de v.a., i.i.d, suivant la loi π et d'estimer μ par $\widehat{\mu}_N = \frac{1}{N} \sum_{i \leq N} f(X_i)$.

Que faire si il n'est pas possible d'obtenir des variables aléatoires i.i.d suivant π ?

Principe de MCMC : Monte Carlo Markov Chain

Il s'agirait de construire une suite (X_i) de variables aléatoires qui seraient (presque) i.i.d , suivant π. Il serait alors possible d'utiliser la méthode de Monte Carlo pour estimer μ.
Une Chaîne de Markov (à temps discret), ergodique, de loi stationnaire π ($\pi = \pi \cdot P$) est un processus stochastique qui permet de générer une telle suite (X_i).

PS : il faut un "certain nombre" d'itérations pour qu'une chaîne de Markov s'approche de la convergence (c'est à dire $P(X_t) \approx \pi$). Cette période (ou burn-in) passée, on peut considérer que $X_t \perp\!\!\!\perp X_{t+1}$, puisque les 2 v.a. suivent la même loi π.

MCMC : Monte Carlo Markov Chain

Changement de point de vue

- Quand on étudie les MC(TD), à partir d'une matrice de transition P, il s'agit de trouver la distribution stationnaire π.
- Pour les MCMC, étant donnée une loi π, il s'agit de construire une MC(TD) convergent vers cette loi π.

Comment construire cette chaîne de Markov ?

Définition (Algorithme de Metropolis-Hastings – 1953)

Soit les lois $q(X \mid Y)$ lois candidates ou instrumentales, on construit alors
$\alpha(X, Y) = \min\left(1, \frac{\pi(Y)q(X|Y)}{\pi(X)q(Y|X)}\right)$

- *Soit x_t la position courante du processus stochastique*
- *Itérations :*
  1. *Proposer un candidat y suivant la loi $q(. \mid x_t)$*
  2. *Calculer $\alpha(x_t, y)$*
  3. *Avec la probabilité $\alpha(x_t, y)$, $x_{t+1} = y$, sinon $x_{t+1} = x_t$*
- *les $(x_t)_{m \leq t \leq N}$ forment une suite de v.a. i.i.d utilisables pour une approximation MC.*

MCMC - suite

1. Proposer un candidat y suivant la loi $q(. \mid x_t)$
2. Calculer $\alpha(x_t, y)$
3. Avec la probabilité $\alpha(x_t, y)$, $x_{t+1} = y$: acceptation, sinon $x_{t+1} = x_t$: rejet

Les étapes 1 et 3 sont indépendantes, on peut donc calculer la probabilité de transition par :

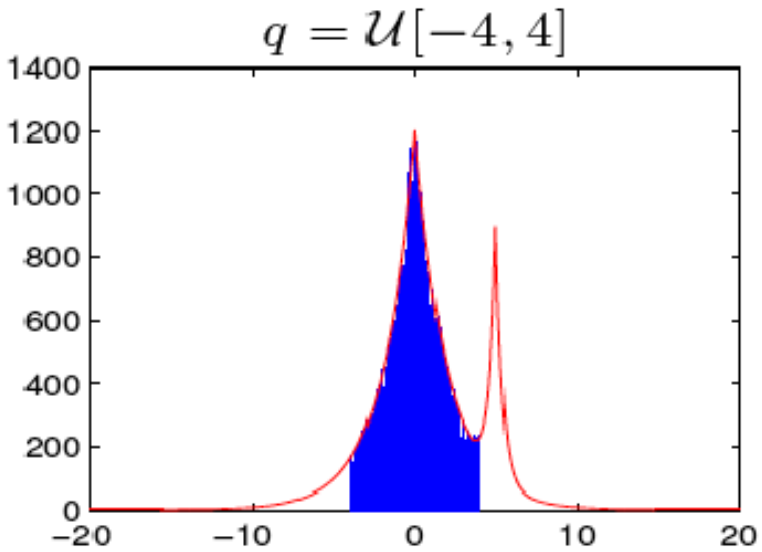
$$\begin{cases} P(X_{t+1} = y \mid X_t = x) = q(y \mid x) \cdot \alpha(x, y) & \forall x \neq y \\ P(X_{t+1} = x \mid X_t = x) = 1 - \sum_{y \neq x} P(X_{t+1} = y \mid X_t = x) \end{cases}$$

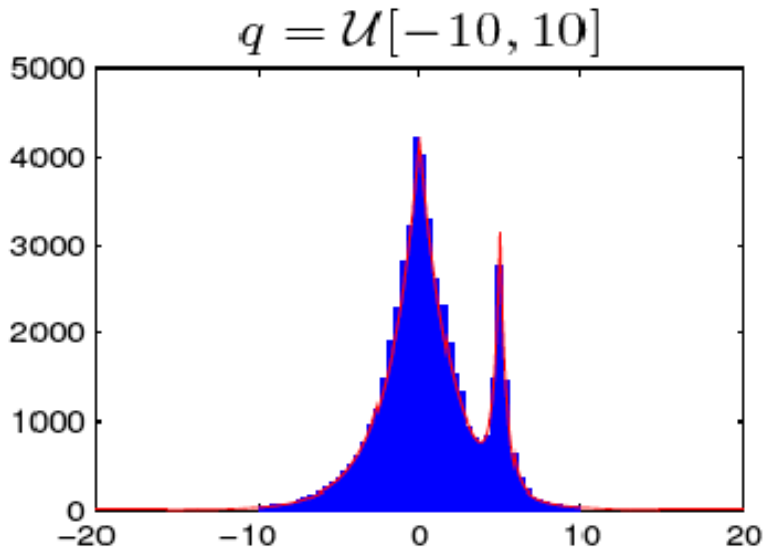
Cette chaîne de Markov est irréductible et apériodique en fonction de $q(x \mid y)$ et $\alpha(x, y)$. Quelle est son point fixe ?

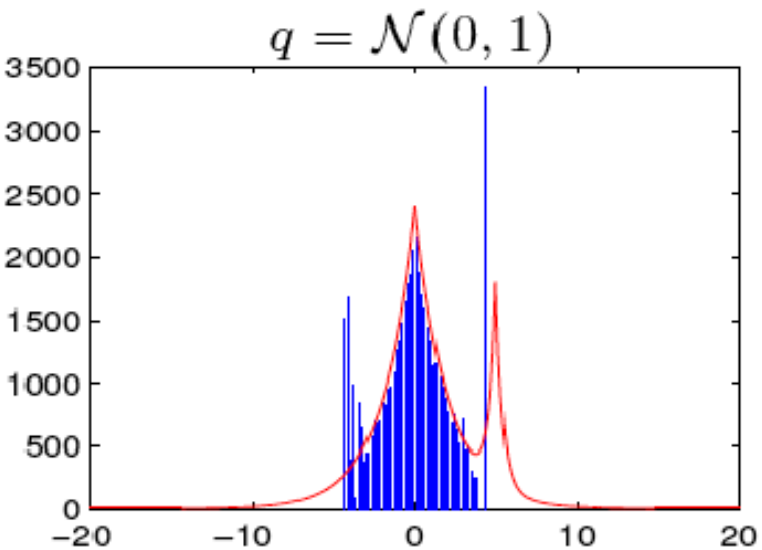
$\alpha(X, Y) = \min\left(1, \frac{\pi(Y)q(X|Y)}{\pi(X)q(Y|X)}\right)$
$\Rightarrow \pi(X_t)q(X_{t+1} \mid X_t)\alpha(X_t, X_{t+1}) = \pi(X_{t+1})q(X_t \mid X_{t+1})\alpha(X_{t+1}, X_t)$
$\Rightarrow \pi(X_t)P(X_{t+1} \mid X_t) = \pi(X_{t+1})P(X_t \mid X_{t+1})$
En sommant sur X_t : $\sum_x \pi(X_t = x)P(X_{t+1} \mid X_t = x) = \pi(X_{t+1}) \Rightarrow \pi \cdot P = \pi$

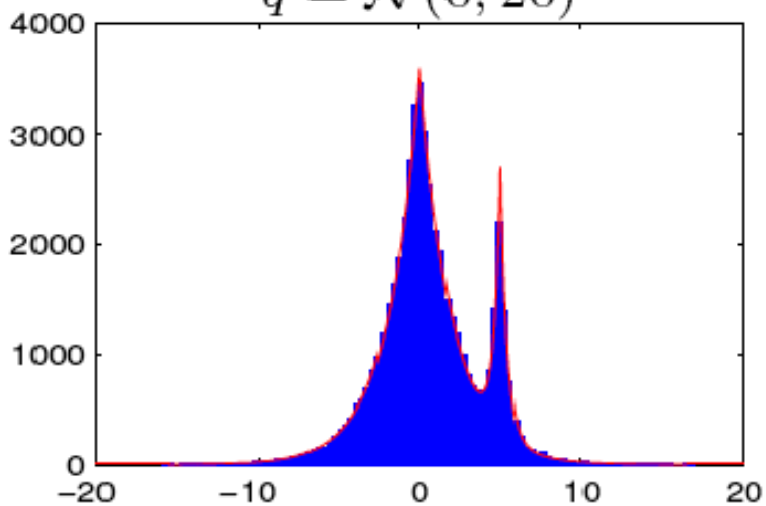
Si cette CM(TD) converge, c'est vers π ■

Influence de $q(x \mid y)$









Une méthode MCMC dans les BNs : Gibbs sampling

Le Gibbs sampling est une méthode MCMC qui prend une forme très simple dans les BNs :

Itération $t+1$ du Gibbs Sampling

soit p^t la particule échantillonnée à t, on note $-X$ toutes les variables du BN sauf X,

$$\exists X \in \text{ BN}, \begin{cases} p^{t+1}_{<X>} \text{ est tirée suivant la distribution } P(X \mid p^t_{<-X>}) \\ p^{t+1}_{<-X>} = p^t_{<-X>} \end{cases}$$

rappel : La couverture de Markov d'un nœud est constitué de ses parents, de ses enfants et des parents de ses enfants.

En fait, calculer $P(X \mid p^t_{<-X>})$ ne nécessite que la couverture de Markov (Markov Blanket) de X car, dans un BN, $X \perp\!\!\!\perp -X \mid MB(X)$.

Inférence approchée par simplification

Approximation de la distribution à calculer

Constat : calculer $P(X \mid e)$ est trop difficile.
But : trouver une loi $Q(X)$ plus facile et "proche" de $P(X \mid e)$.

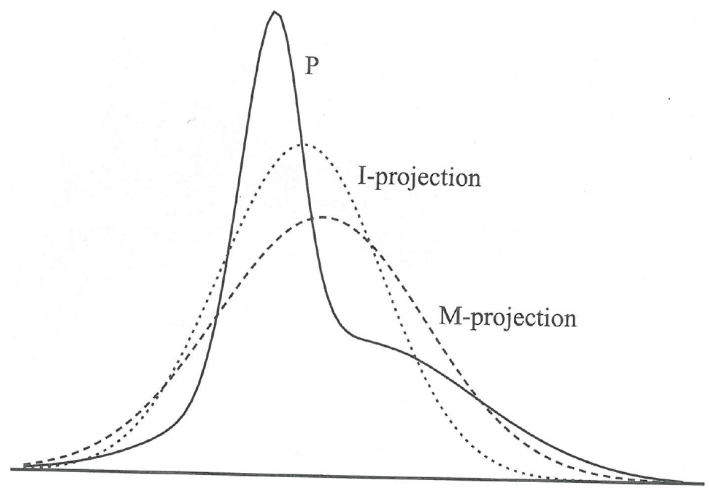
Soit $\mathfrak{Q}$ un ensemble (convexe) de distributions faciles,
il s'agit de trouver $\min_{Q \in \mathfrak{Q}}(\text{distance}(Q,P))$ en utilisant $D_{KL}(P\|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$!

D_{KL} n'est pas symétrique ! !

I-projection et M-projection

Moment projection : $Q_M = \arg\min_{Q \in \mathfrak{Q}} D_{KL}(P\|Q)$

Information projection : $Q_I = \arg\min_{Q \in \mathfrak{Q}} D_{KL}(Q\|P)$



- M-projection préfère rendre probable tout x où $P(x) > 0$,
- I-projection privilégie les grandes probabilités de P

I-projection et M-projection

M-projection

Moment projection : $Q_M = \arg\min_{Q \in \mathfrak{Q}} D_{KL}(P\|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$

- Supporté par P : la "vraie" loi,
- M-projection est certainement le calcul le plus correct,
- Mais nécessite $P(x)$: on ne sait pas calculer !

I-projection

Information projection : $Q_I = \arg\min_{Q \in \mathfrak{Q}} D_{KL}(Q\|P) = \sum_x Q(x) \log \frac{Q(x)}{P(x)}$

- Supporté par Q : la "fausse" loi simplifiée,
- I-projection a tendance à sur-évaluer les probas fortes,
- Mais nécessite $Q(x)$: on devrait savoir calculer !

Calcul de la plus proche distribution factorisée

On suppose $P(X, Y)$ connue. On veut trouver la plus proche distribution où les 2 variables sont indépendantes.
On cherche donc $Q(X, Y) = Q(X) \cdot Q(Y)$ la plus proche de P.

Paramètres à estimer :

Fonction à minimiser :

Sous les contraintes :

Calcul de la plus proche distribution factorisée (2)

Lagrangien :

Dérivées partielles w.r.t. θ_{x^k} :

Point selle = dérivées nulles :

Valeur de λ_x **(resp.** λ_y**)** :

Valeur de θ_x **(resp.** θ_y**)** :

Calcul du KL pour la I-projection

$$\begin{aligned} D_{KL}(Q\|P) &= \sum_x Q(x) \log \frac{Q(x)}{P(x)} \\ &= \sum_x Q(x) \log Q(x) - \sum_x Q(x) \log P(x) \\ &= -H(Q) + H(Q,P) \end{aligned}$$

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

- $H(Q)$: Entropy
- $H(Q,P) = E_Q(\log P)$: Cross Entropy

J. E. Shore and R. W. Johnson, *Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy*, Information Theory, IEEE Transactions on, vol. 26, no. 1, pp. 26-37, Jan. 1980.

Mean Field Approximation

On veut calculer : $P(X) = \frac{1}{Z} \prod_{C \in JT} \Phi_C(C)$

les observations sont intégrées dans les $\Phi_C(C)$.

Mean Field : $Q(x)$ au plus simple

$$Q(x) = \prod_{X_i} Q_i(X_i)$$

Entropy : facile à calculer

$H(Q) = \sum_x Q(x) \log Q(x) = \sum_i \sum_{X_i} Q_i(X_i) \log Q_i(X_i)$

Cross Entropy : facile à calculer (à une translation de $-\log Z$ près)

$\mathrm{H}(Q, P) = \sum_x Q(x) \log P(x) = \sum_{C \in JT} \sum_C \prod_{X_i \in C} Q_i(X_i) \log \Phi_C(C) - \log Z$

Minimisation de $D_{KL}(Q||P)$

Dans un cas général, on a : $D_{KL}(Q||P) = H(Q,P) - H(Q)$

$$= -\mathbb{E}_Q(\log P) + \mathbb{E}_Q(\log Q)$$

$$= \mathbb{E}_Q(\log Q) - \sum_C \mathbb{E}_Q(\log \Phi_C) + \log Z$$

On appelle fonction d'énergie :
$\mathcal{F}[P,Q] = \sum_C \mathbb{E}_Q(\log \Phi_C) - \mathbb{E}_Q(\log Q) = \sum_C \mathbb{E}_Q(\log \Phi_C) - H(Q)$

$\log Z = D_{KL}(Q||P) + \mathcal{F}[P,Q]$, constante dans l'inférence en cours.

Conséquences

- $\min D_{KL}(Q||P) \iff \max \mathcal{F}[P,Q]$
- Comme $D_{KL}(Q||P) \geq 0$ et tend (au mieux) vers 0, $\log Z \geq \mathcal{F}[P,Q]$ qui tend au mieux vers $\log Z$
- Trouver Q le plus proche de P correspond trouver Q pour que $\mathcal{F}[P,Q]$, borne inférieure de $\log Z$, en soit le plus proche possible $\Rightarrow$ Estimation de Z.

Inférence approchée comme une optimisation

Mean Field inference

$$\begin{aligned} \max \quad & \mathcal{F}[P, \quad (Q_1, \cdots, Q_n)] \\ \text{s.c.} \quad & \forall j, \sum_{X_j} Q_j(X_j) = 1 \\ & \forall j, Q_j(X_j) \geq 0 \end{aligned}$$

Optimisation sous contrainte d'une fonction dérivable $\Rightarrow$ multiplicateurs de Lagrange $: L(Q;\lambda) = \mathcal{F}[P,(Q_1,\cdots,Q_n)] + \sum_j \lambda_j \left(\sum_{X_j} Q_j(X_j) - 1\right)$

La caractérisation du point selle donne :

Solution de l'approximation Mean Field

Q est un point stationnaire pour Mean Field $\iff$

$$Q(X_j = x_j) = \frac{1}{Z_j} \exp \sum_{C \ni X_j} \mathbb{E}_Q(\log \Phi_C(., x_j))$$

Il n'y a pas unicité du point stationnaire.

Algorithme Mean Field

MeanField(Φ, Q_0)

- $ATraiter = X$
- $TantQueATraiter \neq \emptyset$
  - choisir $X_j \in ATraiter$
  - Mise à jour de Q_j

$$Q^{t+1}(X_j = x_j) = \exp \sum_{C \ni X_j} \mathbb{E}_{Q^t}(\log \Phi_C(., x_j))$$

  - Normaliser Q_j^{t+1}
  - $ATraiter \leftarrow ATraiter \setminus \{X_j\}$
  - Si $Q^t \neq Q^{t+1}$ Alors $ATraiter \leftarrow ATraiter \cup voisins(X_j)$
- $FinTantQue$

Certitude de convergence pour cet algorithme, mais pas forcément vers la loi P recherchée.

Mean Field et méthode variationnelles

Variationnelle = généralisation de Mean Field

Idée : utiliser une famille $\mathfrak{Q}$ moins naïve que le produit de marginales.
Donc proposer pour Q une structure calculable simplement mais plus complexe. (arbre, ...).

For more :
Probabilistic Graphical Models : Principles and Techniques, Daphne Koller and Nir Friedman, MIT Press, 2009

CES – Data Scientist

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.

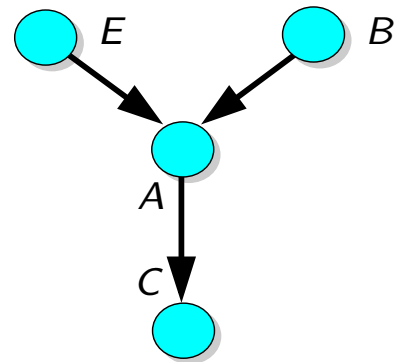
Les connaissances a priori sont très variables ; par exemple :

- 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 :

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

Apprentissage des paramètres, données complètes (2)

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

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

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

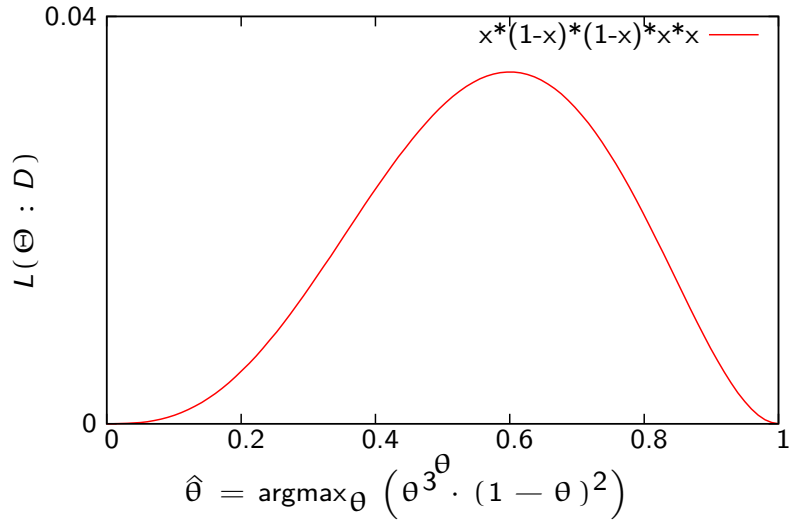
L'estimation des paramètres d'un réseau bayésien se décomposent en l'estimation des paramètres de chaque loi de probabilité conditionnelle

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 $P(\Theta \mid D)$.

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

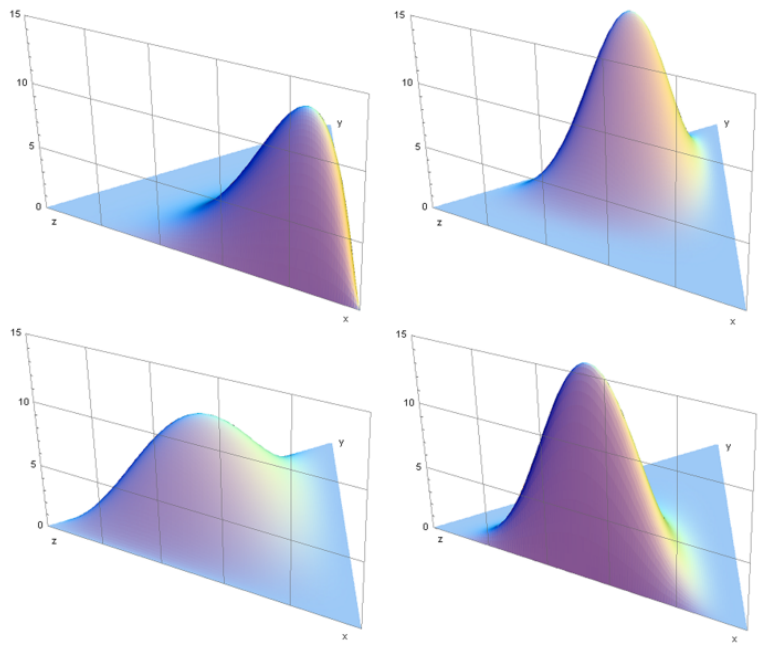
Cette méthode permet de prendre en compte un *a priori* sur Θ ; pour intégrer des connaissances d'expert ou pour rendre plus stable les estimations avec un petit échantillon D.

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)

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

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

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

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

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

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.

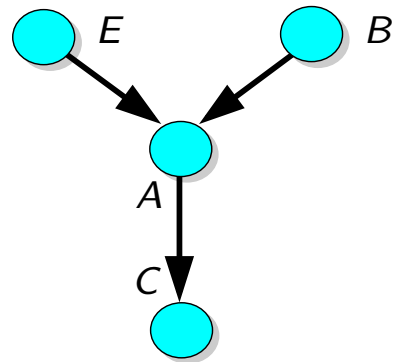
- **actualisation de Ney-Essen**

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

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



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

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

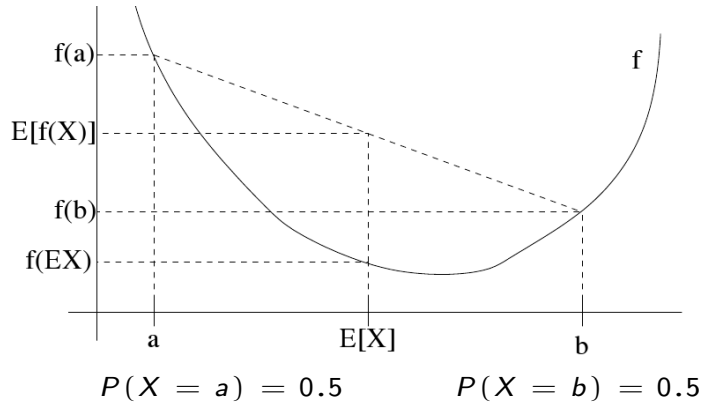
- MCAR :$P(\mathcal{M} \mid D) = P(\mathcal{M})$ (Missing Completely At Random).
- MAR : $P(\mathcal{M} \mid D) = P(\mathcal{M} \mid D^o)$ (Missing At Random).
- NMAR :$P(\mathcal{M} \mid D)$ (Not Missing At Random).

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
- $$f\left(\sum_{i=1}^{n+1}\lambda_i x_i\right) = f\left(\lambda_{n+1}x_{n+1} + \sum_{i=1}^{n}\lambda_i x_i\right)$$

$$= f\left(\lambda_{n+1}x_{n+1} + (1-\lambda_{n+1})\sum_{i=1}^{n}\frac{\lambda_i}{1-\lambda_{n+1}}x_i\right)$$

$$\leq \lambda_{n+1}f(x_{n+1}) + (1-\lambda_{n+1})f\left(\sum_{i=1}^{n}\frac{\lambda_i}{1-\lambda_{n+1}}x_i\right)$$

$$\leq \lambda_{n+1}f(x_{n+1}) + (1-\lambda_{n+1})\sum_{i=1}^{n}\frac{\lambda_i}{1-\lambda_{n+1}}f(x_i)$$

$$= \lambda_{n+1}f(x_{n+1}) + \sum_{i=1}^{n}\lambda_i f(x_i) = \sum_{i=1}^{n+1}\lambda_i f(x_i)$$

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

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

Deux inconvénients majeurs à cet algorithme dans le cas général :

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

EM pour les BNs

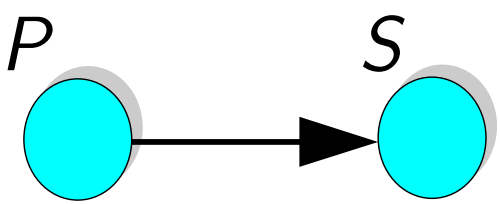
EM dans les BNs

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 :

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

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

EM dans un BN : exemple

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

1 Étape E selon $\theta^{(0)}$

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

Étape M

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

2 Étape E selon $\theta^{(1)}$

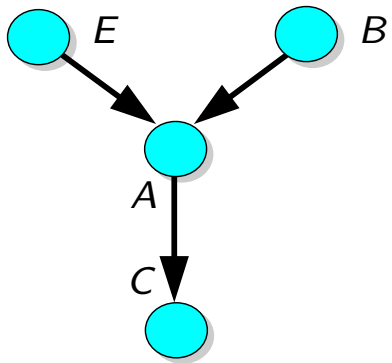
		$P(S \mid P=o)$		$P(S \mid P=n)$	
Pluie	Seine	$S=o$	$S=n$	$S=o$	S=n
o	?	0.433	0.567	0	0
n	?	0	0	0.2	0.8
o	n	0	1	0	0
n	n	0	0	0	1
o	o	1	0	0	0
	N^*	1.433	1.567	0.2	1.8

Étape M

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

3 etc. ($\theta^{(t)}_{S|P=o} \longrightarrow 0.5 \qquad$ et $\qquad \theta^{(t)}_{S|P=n} \longrightarrow 0$)

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 : $NS(10) \approx 4.2 \cdot 10^{18}$!

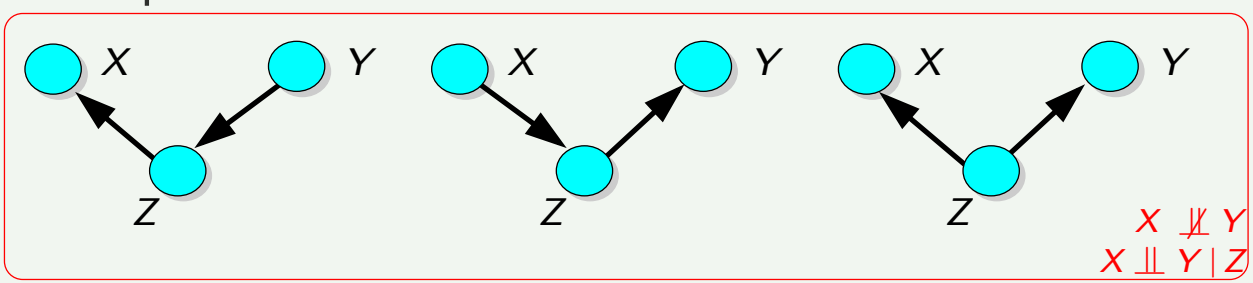
Apprentissage de structure - introduction

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

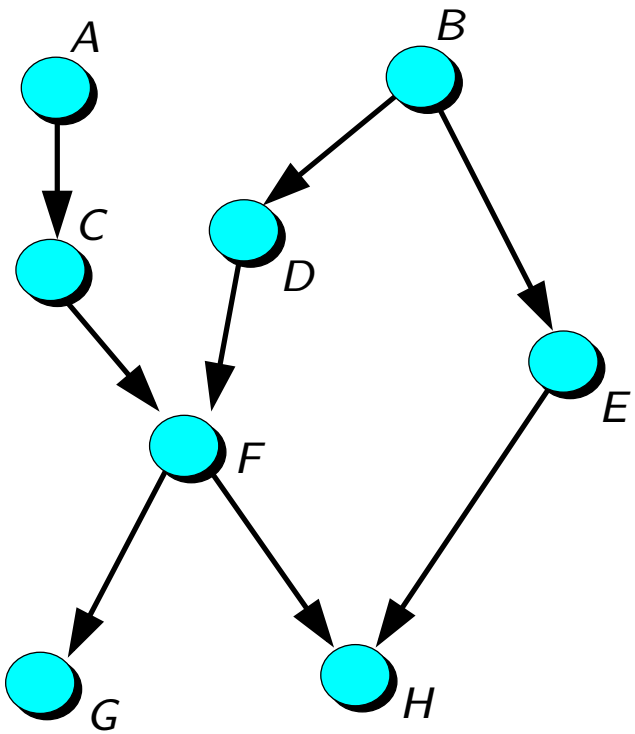
Deux réseaux bayésiens sont équivalents si ils représentent le même modèle d'indépendance.

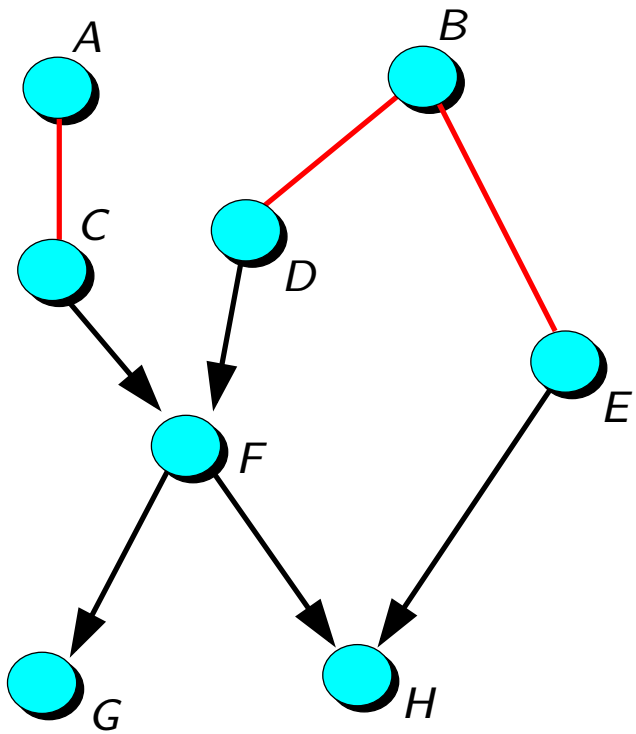


Classe d'équivalence de Markov, graphe essentiel

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)

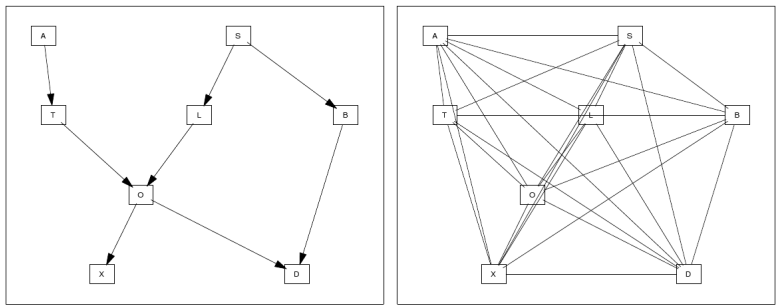
1. Construire le graphe (non orienté) des relations de dépendance trouvées statistiquement (χ^2 ou autre) :
   - Ajouter des arêtes à partir du graphe vide.
   - Retirer des arêtes à partir du graphe complet.
2. Détecter les V-structures et les orientations qu'elles impliquent.
3. Finaliser les orientations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

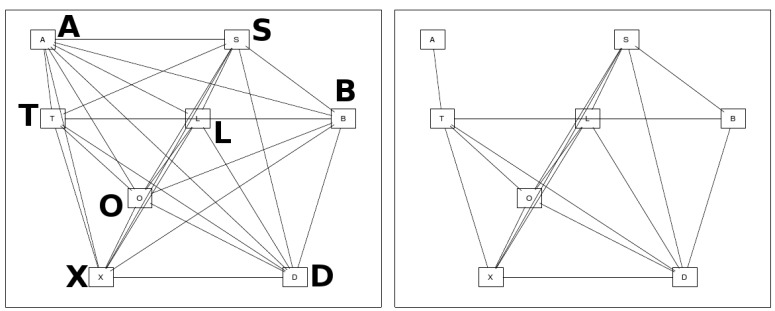
- Soit un réseau bayésien (à gauche) qui a permis de créer une base de 5000 cas. [1]

Etape 0 : Graphe non orienté reliant tous les nœuds.



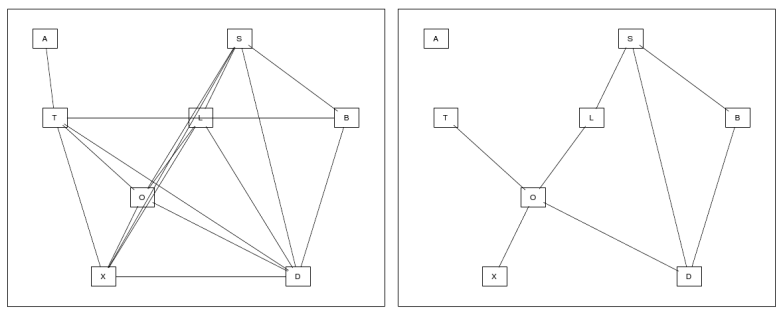
- Par des χ^2, on teste toutes les indépendances marginales ($X \perp\!\!\!\perp Y$) puis les indépendances par rapport à une variable ($X \perp\!\!\!\perp Y \mid Z$).

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



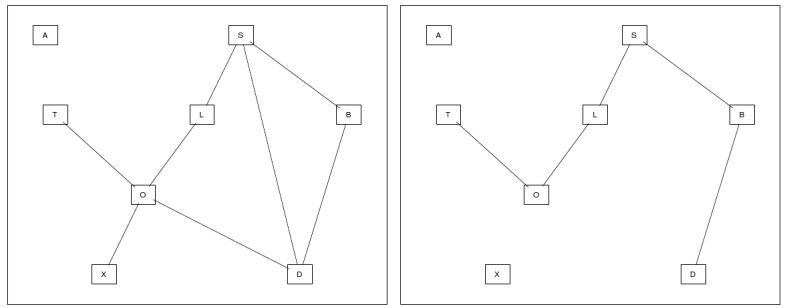
On trouve : $T \perp\!\!\!\perp A \mid O$, $O \perp\!\!\!\perp S \mid L$, $X \perp\!\!\!\perp S \mid L$, $B \perp\!\!\!\perp T \mid S$, $X \perp\!\!\!\perp T \mid O$, $D \perp\!\!\!\perp T \mid O$, $B \perp\!\!\!\perp L \mid S$, $X \perp\!\!\!\perp L \mid O$, $D \perp\!\!\!\perp L \mid O$, $D \perp\!\!\!\perp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

1. Exemple de Philippe Leray

Exemple PC

- On continue les χ^2 d'ordre supérieur

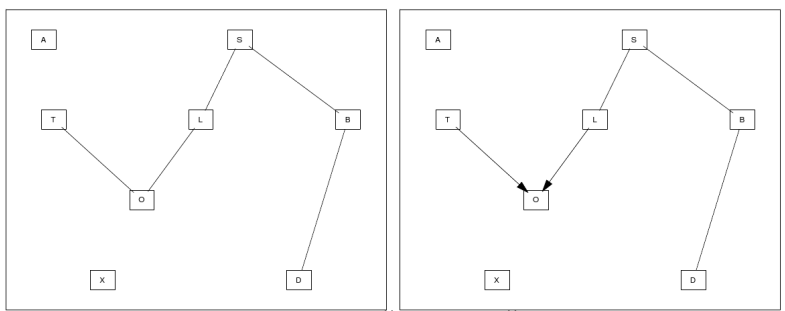
Etape 1c : Suppression des ind. conditionnelles d'ordre 2



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

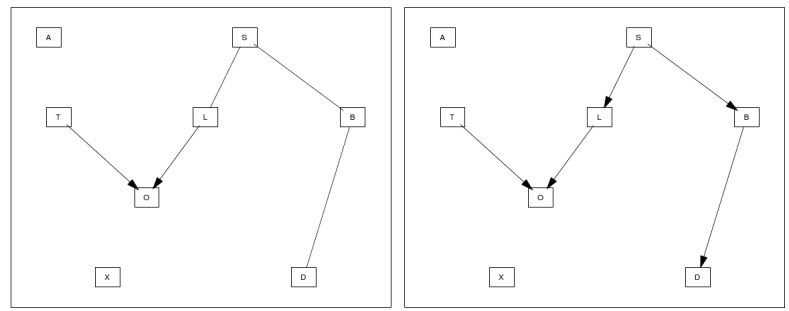
- Recherche des V-Structure, propagation des contraintes d'orientations puis orientations des dernières arêtes en restant Markov-équivalent.

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

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

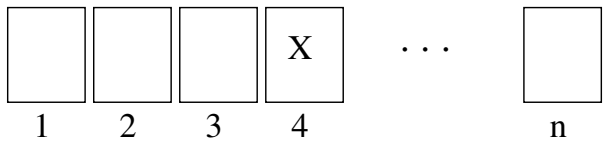
Les fonctions q_i dépendent du nœud i, des parents de i et de la partie de la base de donnée qui correspond à ces nœuds.

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)

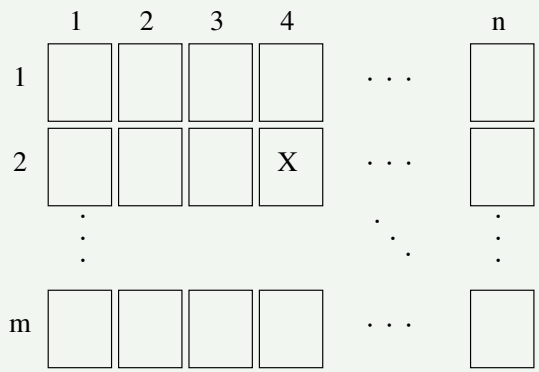
Le nombre d'information $H(n)$ est la quantité d'information reçue en apprenant où se trouve le gain. Elle est équivalente à la quantité d'incertitude *expérimentée au début de l'expérience (sans connaissance).*

$H(n)$ 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. $H(n \cdot m)$?



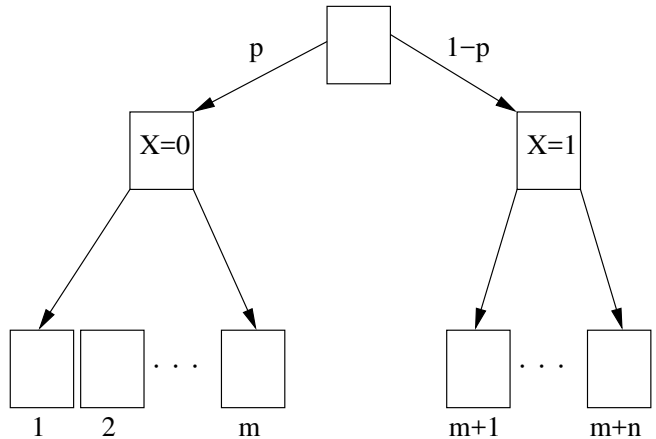
Additivité : $H(n \cdot m) = H(n) + H(m)$

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)

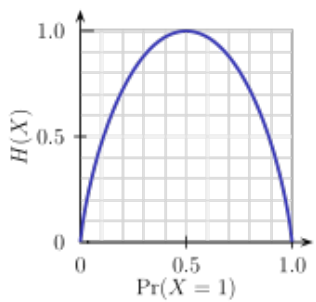
Plutôt que définir l'information apportée par le résultat d'une expérience, il s'agit de mesurer la *quantité moyenne d'information contenue dans une loi de probabilité*.
Soit une v.a. binaire X tel que $P(X = 0) = p$, p rationnel : $p = \frac{m}{m+n}$.



- 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

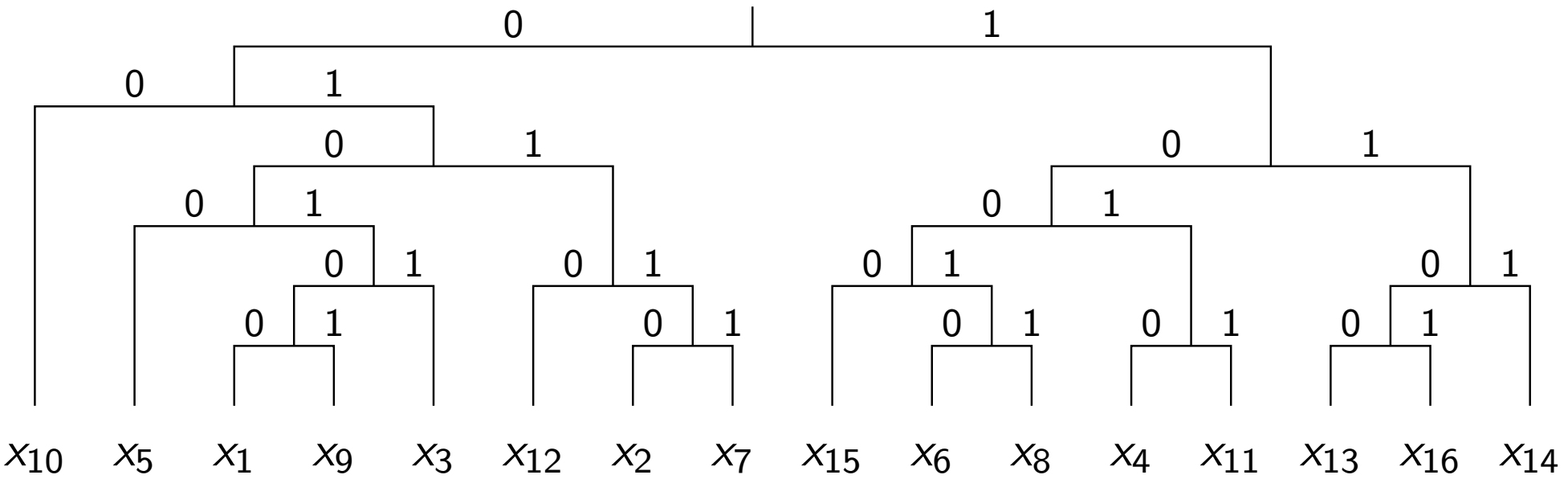
- Faire mieux : minimisation de l'espérance du nombre de bits $\Longrightarrow$ compression

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

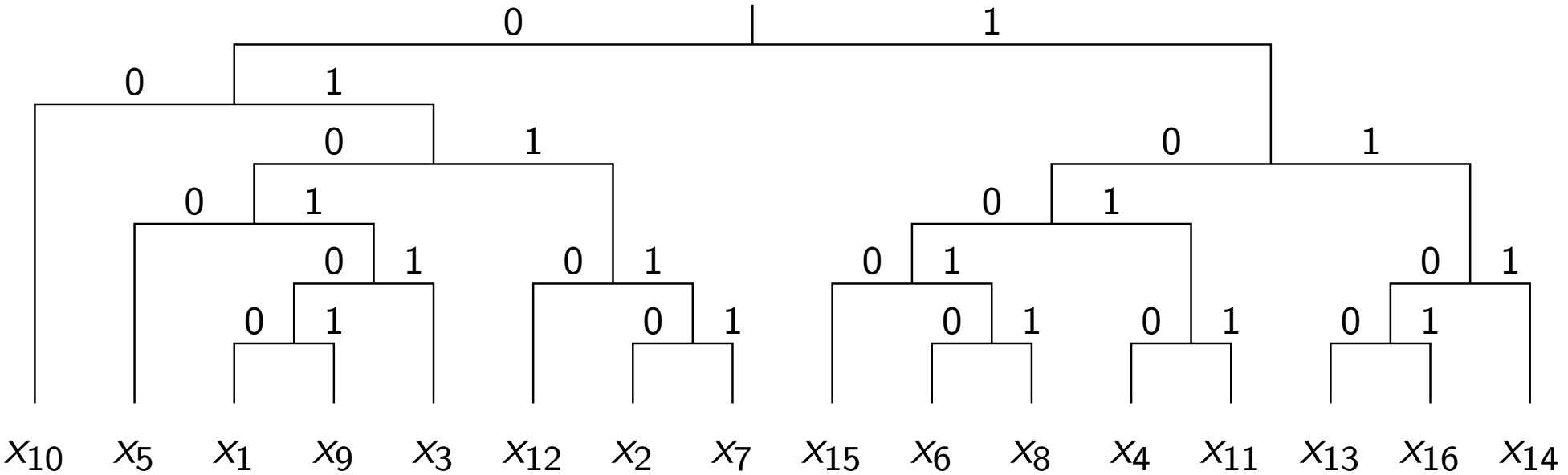
Codage de Huffman



$\Longrightarrow$ 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

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

score BIC (Schwartz, 78)

- Bayesian Information Criterion

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

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 :

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

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*).

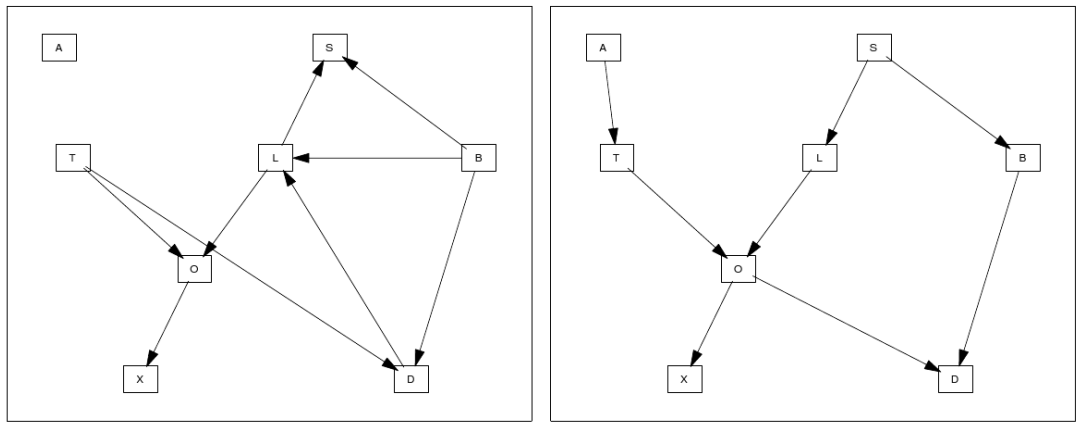
Recherche locale dans les réseaux bayésiens

- 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.

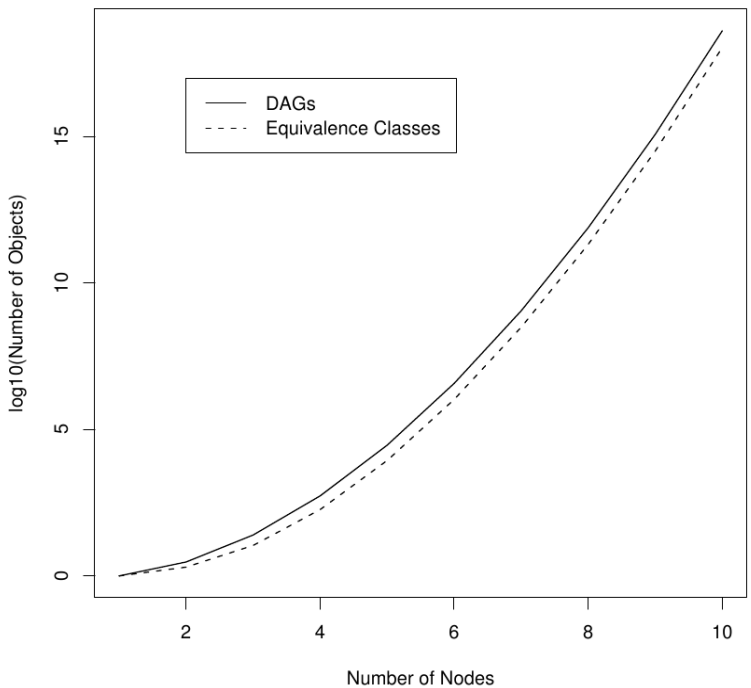
Réduction de l'espace de recherche : équivalents de Markov

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).

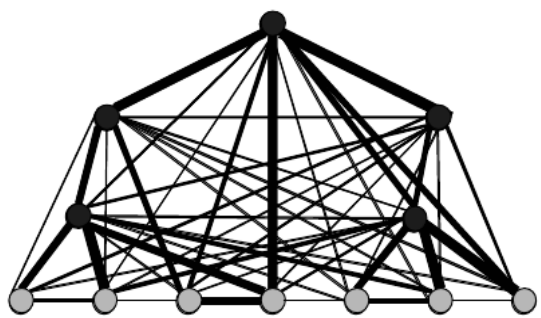
La décomposabilité du score donne :

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

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^* : $A \rightarrow B \leftarrow C$.

- 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