Bayesian Networks

ETICS 2017

Pierre-Henri Wuillemin (LIP6–UPMC)

pierre-henri.wuillemin@lip6.fr

POI

- BNs in a nutshell
- Foundations of BNs
- Learning BNs
- Inference in BNs
- Graphical models for copula
- pyAgrum
  `http://agrum.gitlab.io/pages/pyagrum-installation.html`
  `http://webia.lip6.fr/~phw/Docs/tdsBN.zip`

Nutshell

Inference in discrete probabilistic model

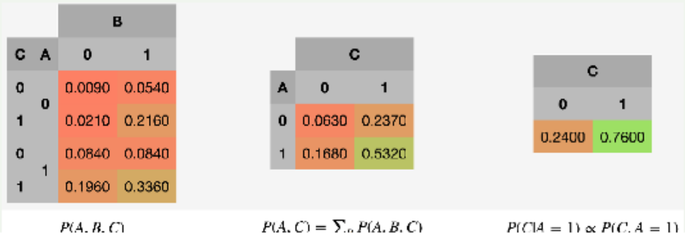
Inference

Let A, B be (discrete) random variables,
Inference consists in computing the distribution of A given observations on B.

$$P(A|B=\epsilon_b) = P(A|\epsilon_b) = \frac{P(\epsilon_b|A)\cdot P(A)}{P(\epsilon_b)} \propto P(\epsilon_b|A)\cdot P(A) = P(A,\epsilon_b)$$

Probabilistic complex systems : ⚠ $P(X_1, \cdots, X_n)$,

Inference in complex system : $P(X_i|X_j=\epsilon_j) \propto P(X_i,\epsilon_j)$



		B	
C	A	0	1
0	0	0.0090	0.0540
1		0.0210	0.2160
0	1	0.0840	0.0840
1		0.1960	0.3360

$P(A, B, C)$

	C	
A	0	1
0	0.0630	0.2370
1	0.1680	0.5320

$P(A, C) = \sum_B P(A, B, C)$

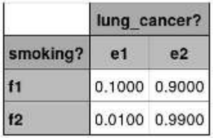
C	
0	1
0.2400	0.7600

$P(C|A=1) \propto P(C, A=1)$

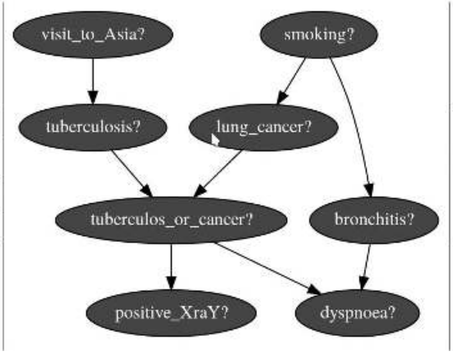
Bayesian networks : definition

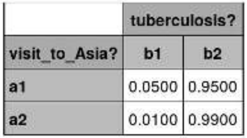
Definition (Bayesian Network (BN))

A Bayesian network is a joint distribution over a set of random (discrete) variables. A Bayesian network is represented by a directed acyclic graph (DAG) and by a conditional probability table (CPT) for each node $P(X_i|parents_i)$



	lung_cancer?	
smoking?	**e1**	**e2**
f1	0.1000	0.9000
f2	0.0100	0.9900



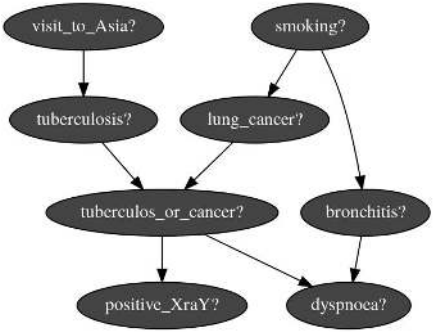


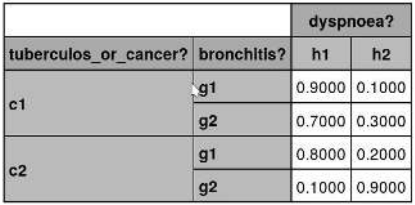
	tuberculosis?	
visit_to_Asia?	**b1**	**b2**
a1	0.0500	0.9500
a2	0.0100	0.9900

Bayesian networks : definition

Definition (Bayesian Network (BN))

A Bayesian network is a joint distribution over a set of random (discrete) variables. A Bayesian network is represented by a directed acyclic graph (DAG) and by a conditional probability table (CPT) for each node $P(X_i | parents_i)$





		dyspnoea?	
tuberculos_or_cancer?	bronchitis?	h1	h2
c1	g1	0.9000	0.1000
	g2	0.7000	0.3000
c2	g1	0.8000	0.2000
	g2	0.1000	0.9000

Bayesian networks : definition

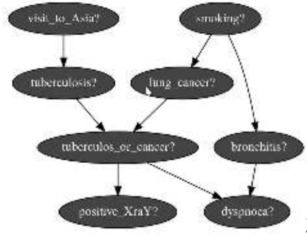
Definition (Bayesian Network (BN))

A Bayesian network is a joint distribution over a set of random (discrete) variables. A Bayesian network is represented by a directed acyclic graph (DAG) and by a conditional probability table (CPT) for each node $P(X_i|parents_i)$

Factorization of the joint distribution in a BN

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

$P(A, S, T, L, O, B, X, D)$



$P(A) \cdot P(S) \cdot P(T|A) \cdot P(L|S) \cdot P(O|T, L) \cdot P(B|S) \cdot P(X|O) \cdot P(D|O, B)$

($2^8 = 256$ parameters)

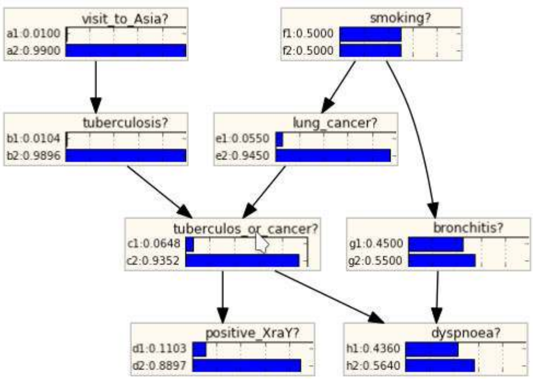
($2+2+4+4+8+4+4+8 = 32$ parameters)

Bayesian networks : definition

Definition (Bayesian Network (BN))

A Bayesian network is represented by a directed acyclic graph (DAG) and by a conditional probability table (CPT) for each node $P(X_i | parents_i)$

Inference : $P(\textbf{dyspnoea})$ **?**

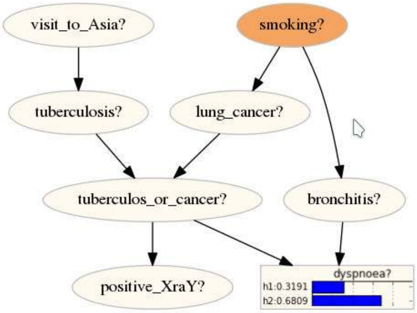


Bayesian networks : definition

Definition (Bayesian Network (BN))

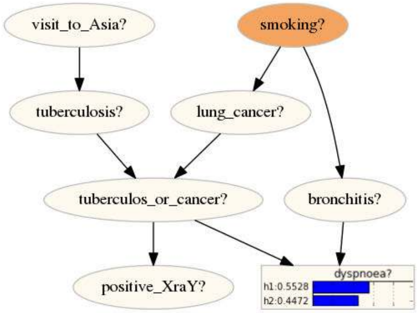
A Bayesian network is represented by a directed acyclic graph (DAG) and by a conditional probability table (CPT) for each node $P(X_i|parents_i)$

Inference : $P(\mathbf{dyspnoea}|\mathbf{smoking})$ **?**



Inference in 14.29ms

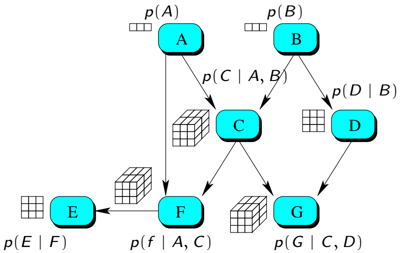
$P(dyspnoea|smoking = 1)$



Inference in 11.35ms

$P(dyspnoea|smoking = 0)$

BN and probabilistic inference



diagnostic : $P(A|F)$

- *diagnostic*
- *reliability*
- *classification*

prediction $P(E|B, A)$

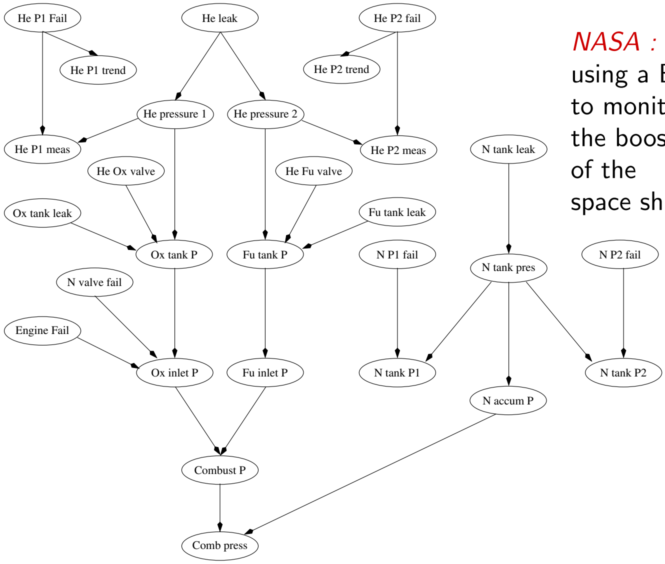
- *Process simulation (modelisation)*
- *forecasting (dynamics, etc.)*
- *Behavioral analysis (bot, intelligent tutoring system)*

Others tasks

- *Most Probable Case* : $\arg\max P(\mathfrak{X}|D)$
- *Sensitivity analysis, Informational analysis (mutual information), etc.*
- *Decision process, Troubleshooting* : $\arg\max \frac{P(.)}{C(.)}$

Application 1 : diagnostic

Diagnostic @ NASA

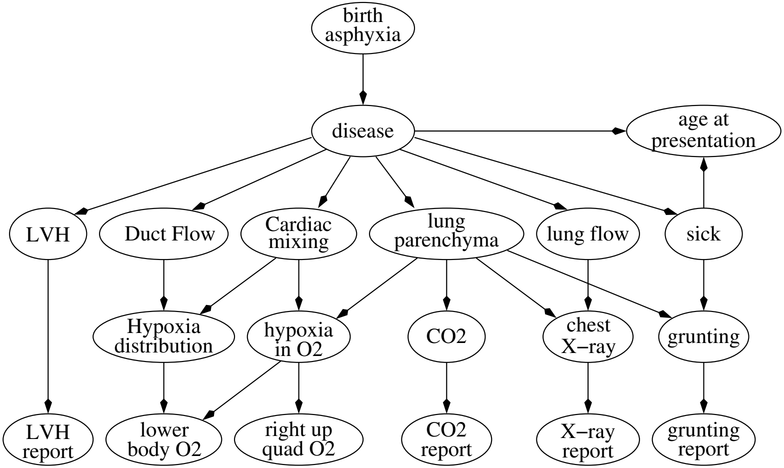


NASA :
using a BN
to monitor
the boosters
of the
space shuttle

Application 2 : medical diagnosis

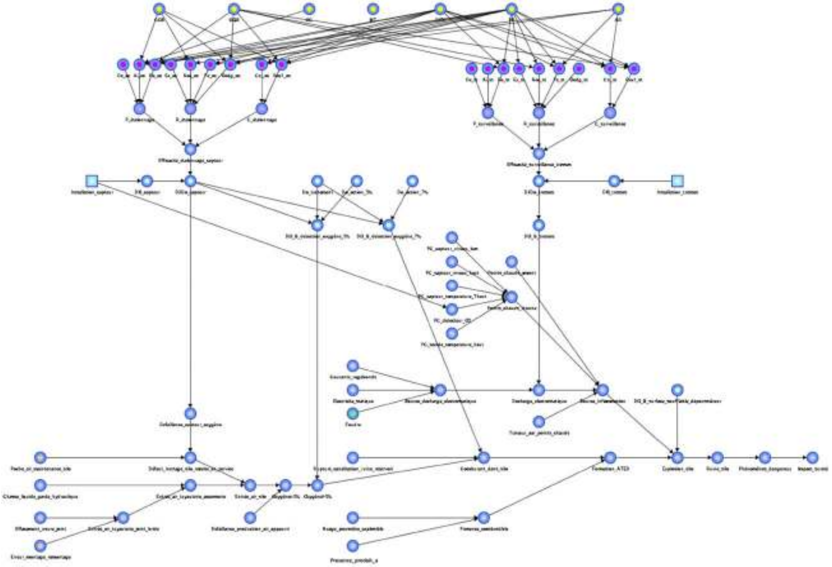
the Great Ormond Street hospital for sick children

Diagnosis of the causes of cyanosis or heart attack in the child just after birth.



Application 3 : risk analysis

Risk modelisation using BN : **modular approach**.



Application 4 : Bayesian classification

X (dimension d, features) and Y (dimension 1, *often binary but not necessarily*). Using a database $\Pi_a = (x^{(k)}, y^{(k)})_{k \in \{1,\cdots,N\}}$ (supervised learning), one can estimate the joint distribution $P(X, Y)$.

Classification

For a vector x, values of X, the goal is to predict the class (value of Y) : $\widehat{y}$.

1 Maximum of the likelihood (ML)

$$\widehat{y} = \arg\max_y P(x|y)$$

2 Maximum a posteriori (MAP)

$$\widehat{y} = \arg\max_y P(y|x) = \arg\max_y P(y) \cdot P(x|y)$$

Those distributions may be hard to estimate.
$P(X|Y)$ may induce more parameters than $|\Pi_a|$! !

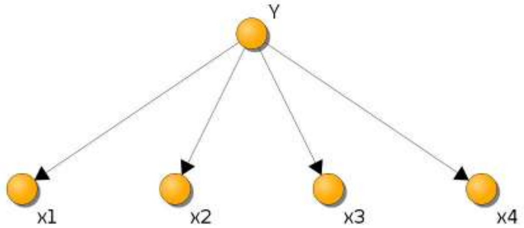
Bayesian classification (2) : naive Bayes

How to compute $P(X|Y)$?

Naive Bayes classifier

if we assume $\forall k \neq l, X_k \perp\!\!\!\perp X_l \,|\, Y$ then $P(x, y) = P(y) \cdot \prod_{k=1}^{d} P(x_k|y)$

Very strong assumption ! In most cases, it is an approximation. However, this approximation often gives good results.

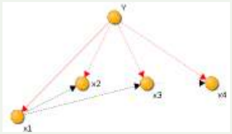


- Parameters estimation : trivial (if no missing values)
- ML : $\prod_{k=1}^{d} P(x_k|y)$ ...
- MAP : $P(y|x_1, \cdots, x_d)$: inference in the BN !

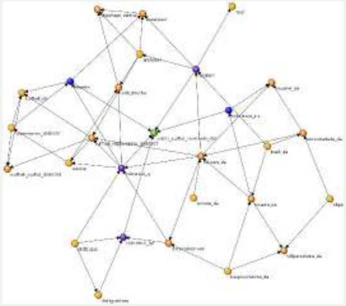
Bayesian classification (3) : more complex models

TAN : Tree-Augmented Naive Models

Every variable X_i can have Y and another parent among X (only one !).



Complete Bayesian network



In a BN including Y and (X_i), infer $P(Y|X_1, \cdots, X_n)$.
Note : There is no need to all X_i : Markov Blanket $MB(.)$)

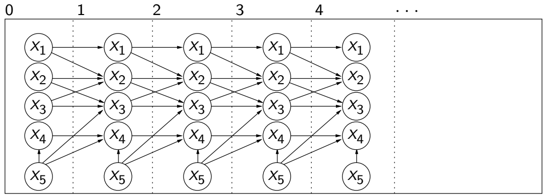
$$P(Y|X) = P(Y|MB(Y))$$

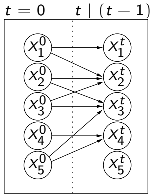
Application 5 : dynamic Bayesian networks

dBN (dynamic BN)

A dynamic BN is a BN where variables are indexed by the time t and by i : $X^t = \{X_1^t, \cdots, X_N^t\}$ and verifies :

- Markov order 1 : $P(X^t|X^0, \cdots, X^{t-1}) = P(X^t|X^{t-1})$,
- Homogeneity : $P(X^t|X^{t-1}) = \cdots = P(X^1|X^0)$.





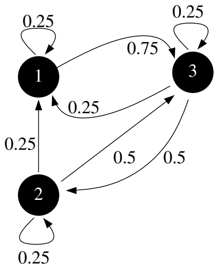
2-TBN

A dBN is characterized by :

- initial conditions ($P(X^0)$)
- the relations between $t-1$ and t (*timeslice*).

2TBN (2 timeslice BN) allows to specify a dBN of size T by starting with X^0 and copying T times the pattern $(t|t-1)$.

dBN and Markov chain



Markov chain

- A state variable (X^n) (at time n).
- Parameters :
  - Initial condition : $P(X^O)$
  - Transition model : $P(X^n|X^{n-1})$

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

Equivalent dynamic Bayesian Network :

dBN : $$\begin{tikzcd} X^0 \arrow[r] & X^1 \arrow[r] & X^2 \arrow[r] & X^3 \arrow[r] & \cdots \end{tikzcd}$$

2TBN : $$\begin{tikzcd} X^0 \arrow[r] & X^n \end{tikzcd}$$

dBN and Hidden Markov Model

HMM

- A state variable (X^n) (at time n).
- An observation variable (Y^n)
- Parameters :
  - Initial condition : $P(X^O)$
  - Transition model : $P(X^n|X^{n-1})$
  - Observation model : $P(Y^n|X^n)$

Equivalent dynamic Bayesian Network :

dBN :

$$\begin{tikzcd}
X^0 \arrow[r] \arrow[d] & X^1 \arrow[r] \arrow[d] & X^2 \arrow[r] \arrow[d] & X^3 \arrow[r] \arrow[d] & {} \\
Y^0 & Y^1 & Y^2 & Y^3 & \cdots
\end{tikzcd}$$

2TBN :

$$\begin{tikzcd}
X^0 \arrow[r] \arrow[d] & X^n \arrow[d] \\
Y^0 & Y^n
\end{tikzcd}$$

Foundations

Joint probability, Factorization, conditional independence

- A joint probability is expensive in memory

$\boxed{K^3}$ $p(X, Y, Z)$

- A joint probability can be factorized (*chain rule*)

$\boxed{K^3}$ $p(X, Y, Z) = p(X) \cdot p(Y \mid X) \cdot p(Z \mid X, Y)$ $\boxed{K + K^2 + K^3}$

- **Conditional independence is the key**.

With $X \perp\!\!\!\perp Y$ and $Z \perp\!\!\!\perp X, Y$:

$\boxed{K^3}$ $p(X, Y, Z) = p(X) \cdot p(Y) \cdot p(Z)$ $\boxed{3K}$

Goal : how to describe the list of all conditional independence in a joint probability :

$$\{U, V, W \subset X \text{ with } U \perp\!\!\!\perp V \,|\, W\}$$

Independence model

More generally, this ternary relation between subsets is called separability.

Definition (Independence model and separability)

Let X be a finite set, let $\mathfrak{I} \subset \mathcal{P}(X) \times \mathcal{P}(X) \times \mathcal{P}(X)$. i.e. a list of triplets of subsets of X.
$\mathfrak{I}$ is called an independence model.
$\forall U, V, W \subset X$, U and V are separated by W ($\ll U \oplus V \mid W \gg_{\mathfrak{I}}$) if and only if $(U, V, W) \in \mathfrak{I}$.
i.e. $\mathfrak{I}$ is the list of all 'separations' found in X.

Relation between $\mathfrak{I}$ and p

$\mathfrak{I}_p = \{(U, V, W) \in\subset \mathcal{P}(X) \times \mathcal{P}(X) \times \mathcal{P}(X), U \perp\!\!\!\perp V \mid W\}$ is an independence model.

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

Reciprocally, if $\mathfrak{I}$ is an independence model for X set of random variables, can we found P a distribution that verifies this list of conditional independence ?

Semi-graphoid and graphoid

Definition (semi-graphoid)

An independence model $\mathfrak{I}$ is a semi-graphoid if and only if $\forall A, B, S, P \subset \mathfrak{X}$:

1. *trivial independence* $\ll A \perp\!\!\!\perp \emptyset \mid S \gg_{\mathfrak{I}}$
2. *Symmetry* $\ll A \perp\!\!\!\perp B \mid S \gg_{\mathfrak{I}} \Rightarrow \ll B \perp\!\!\!\perp A \mid S \gg_{\mathfrak{I}}$
3. *Decomposition* $\ll A \perp\!\!\!\perp (B \cup P) \mid S \gg_{\mathfrak{I}} \Rightarrow \ll A \perp\!\!\!\perp B \mid S \gg_{\mathfrak{I}}$
4. *Weak union* $\ll A \perp\!\!\!\perp (B \cup P) \mid S \gg_{\mathfrak{I}} \Rightarrow \ll A \perp\!\!\!\perp B \mid (S \cup P) \gg_{\mathfrak{I}}$
5. *Contraction* $\left\{\begin{matrix} \ll A \perp\!\!\!\perp B \mid (S \cup P) \gg_{\mathfrak{I}} \\ \ll A \perp\!\!\!\perp P \mid S \gg_{\mathfrak{I}} \end{matrix}\right\} \Rightarrow \ll A \perp\!\!\!\perp (B \cup P) \mid S \gg_{\mathfrak{I}}$

Definition (graphoid)

An independence model $\mathfrak{I}$ is a graphoid if and only if $\forall A, B, S, P \subset \mathfrak{X}$:

$\mathfrak{I}$ is a semi-graphoid

6. *Intersection* $\left\{\begin{matrix} \ll A \perp\!\!\!\perp B \mid (S \cup P) \gg_{\mathfrak{I}} \\ \ll A \perp\!\!\!\perp P \mid (S \cup B) \gg_{\mathfrak{I}} \end{matrix}\right\} \Rightarrow \ll A \perp\!\!\!\perp (B \cup P) \mid S \gg_{\mathfrak{I}}$

These axioms create a strong structure inside the independence model.

Semi-graphoid and graphoid - representation of the axioms

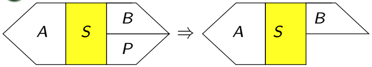
3 Decomposition $$\ll A \perp\!\!\!\perp (B \cup P) \mid S \gg_{\mathfrak{I}} \quad \Rightarrow \quad \ll A \perp\!\!\!\perp B \mid S \gg_{\mathfrak{I}}$$

4 Weak union $$\ll A \perp\!\!\!\perp (B \cup P) \mid S \gg_{\mathfrak{I}} \quad \Rightarrow \quad \ll A \perp\!\!\!\perp B \mid (S \cup P) \gg_{\mathfrak{I}}$$

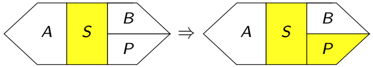
5 Contraction $$\left\{ \begin{matrix} \ll A \perp\!\!\!\perp B \mid (S \cup P) \gg_{\mathfrak{I}} \\ \ll A \perp\!\!\!\perp P \mid S \gg_{\mathfrak{I}} \end{matrix} \right\} \quad \Rightarrow \quad \ll A \perp\!\!\!\perp (B \cup P) \mid S \gg_{\mathfrak{I}}$$

6 Intersection $$\left\{ \begin{matrix} \ll A \perp\!\!\!\perp B \mid (S \cup P) \gg_{\mathfrak{I}} \\ \ll A \perp\!\!\!\perp P \mid (S \cup B) \gg_{\mathfrak{I}} \end{matrix} \right\} \quad \Rightarrow \quad \ll A \perp\!\!\!\perp (B \cup P) \mid S \gg_{\mathfrak{I}}$$

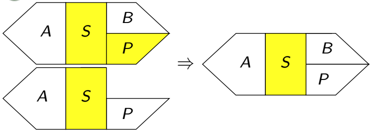
3 Decomposition



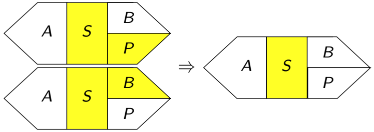
4 Weak union



5 Contraction



6 Intersection

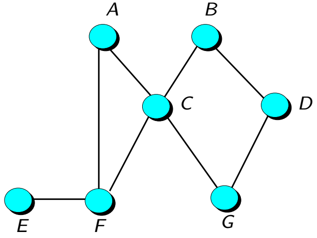


Semi-graphoid and graphoid

Theorem (probability and graphoid)

$\mathfrak{I}_p$ is a semi-graphoid.
If p positive then $\mathfrak{I}_p$ is a graphoid.

Is there any other kind of ternary relation that is a graphoid ?



Theorem (Undirected graph and graphoid)

Let $G = (X, E)$ an undirected graph,
$\forall U, V, W \subset X, \langle U \mid W \mid V \rangle_G$ if and only if every path from a node in U to a node in V includes a node in W.
$\{\langle U \mid W \mid V \rangle_G, U, V, W \subset X\}$ is a graphoid.

Graphical model

$$\boxed{P(X),. \perp\!\!\!\perp .|.} \longleftrightarrow \boxed{\mathfrak{I} \text{ Independence model}} \longleftrightarrow \boxed{G = (X, E), \langle .\,|.\,|.\rangle_G}$$

Definition (Graphical model)

A graphical model is a joint probability distribution $P(X)$ which uses a graph $G = (X, E)$ to represent its conditional independence using separation in the graph.

$$\boxed{P(X),. \perp\!\!\!\perp .|.} \leftarrow ? \rightarrow \boxed{\mathfrak{I} \text{ Independence model}} \leftarrow ? \rightarrow \boxed{G = (X, E), \langle .\,|.\,|.\rangle_G}$$

Definition (I-map, D-map, P-map, graph-isomorphism)

let $G = (X, E)$ a graph and a distribution $p(X)$.

G is a Dependency-map for p $\Leftrightarrow \quad (X \perp\!\!\!\perp Y | Z)_p \Rightarrow \langle X\,|\,Z\,|\,Y\rangle_G$.

G is a Independency-map for p $\Leftrightarrow \quad (X \perp\!\!\!\perp Y | Z)_p \Leftarrow \langle X\,|\,Z\,|\,Y\rangle_G$.

G is a Perfect-map for p $\Leftrightarrow \quad (X \perp\!\!\!\perp Y | Z)_p \Leftrightarrow \langle X\,|\,Z\,|\,Y\rangle_G$.

$p(X)$ is graph-isomorph if and only if $\exists G = (X, E)$ P-map for $p(X)$.

- The empty graph $(X, \emptyset)$ is a **D-map** for all $p(X)$.
- The complete graph is a **I-map** for all distribution $p(X)$.

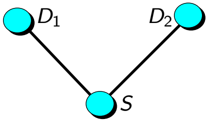
Undirected graphical model : example 1

example 1

Let $p(D_1, D_2, S)$ be the joint probability distribution with D_1 and D_2 two (independent) dice and $S = D_1 + D_2$.

(in)Dependence in example 1

- $D_1 \not\perp\!\!\!\perp S$ and $D_2 \not\perp\!\!\!\perp S$
- $D_1 \perp\!\!\!\perp D_2$ but $D_1 \not\perp\!\!\!\perp D_2 | S$



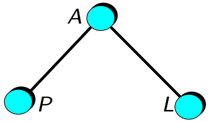
Undirected graphical model : example 2

example 2

In a database of users, a strong correlation has been found between$L$ the ability to read and P the size of the shoes.
This correlation is explained by the fact that a third variable A (the age) can be < 5.

(in)Dependence in example 2

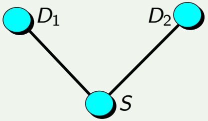
- $L \not\perp\!\!\!\perp A$ and $P \not\perp\!\!\!\perp A$
- $L \not\perp\!\!\!\perp P$ but $L \perp\!\!\!\perp P | A$



Directed graphical model

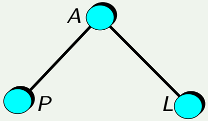
example 1

$D_1 \perp\!\!\!\perp D_2$ but $D_1 \not\perp\!\!\!\perp D_2 | S$



example 2

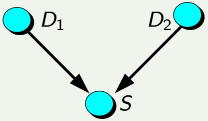
$L \not\perp\!\!\!\perp P$ but $L \perp\!\!\!\perp P | A$



Giving more information on the graph by adding the orientation.

directed example 1

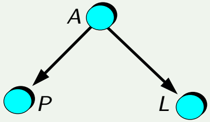
$D_1 \perp\!\!\!\perp D_2$ but $D_1 \not\perp\!\!\!\perp D_2 | S$



collider or V-structure

directed example 2

$L \not\perp\!\!\!\perp P$ but $L \perp\!\!\!\perp P | A$



We need a separation criterion for directed graphs : d-separation.

Directed graphical model and d-separation

Let $C = (x_i)_{i \in I}$ be an undirected path in $\overrightarrow{G} = (X, E)$.
x_i is a C-collider if C contains : $x_{i-1} \rightarrow x_i \leftarrow x_{i+1}$.

Definition (Active path)

Let $Z \subset X$. C is an active path wrt Z if $\forall x_i \in C$:

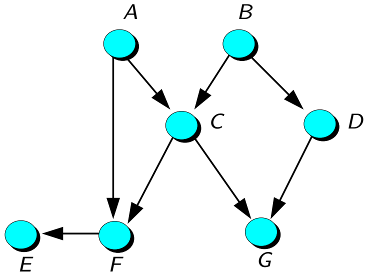
- *If x_i is a C-collider Then x_i or one of its descendant belongs to Z.*
- *Else x_i does not belong to Z.*

If a path is not active wrt Z, it is blocked by Z.

Definition (d-separation)

Let $\overrightarrow{G} = (X, E)$ a directed graph,
$\forall (U, V, W) \subset X$, U is d-separated from V by W in $\overrightarrow{G}$ ($\langle U \mid W \mid V \rangle_{\overrightarrow{G}}$) if and only if every undirected path from U to V is blocked by W.

Sampling some d-separation



Bayesian network and Markov properties

Definition (Global Markov Property)

$\overrightarrow{G}$ *satisfies the GMP for* $p \Leftrightarrow \forall A, B, S \subset \mathfrak{X}$,

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

i.e. $\overrightarrow{G}$ is an I-map for p.

Definition (Locale Markov Property)

$\overrightarrow{G}$ *satisfies the LMP for* $p \Leftrightarrow \forall X_i \in X$,

$$\{X_i\} \perp\!\!\!\perp \mathrm{nd}\,(X_i) \,|\, \Pi_{X_i}.$$

where $\mathrm{nd}\,(X_i)$ represents the non-descendant nodes of X_i

Bayesian network and Markov properties (2)

Theorem

(when $p(X)$ positive)

$$GMP \iff LMP$$

Definition (Bayesian network, graphical Factorization)

Let $p(X)$ be a directed graphical model with the graph $\overrightarrow{G} = (X, E)$. $p(X)$ is a Bayesian network if and only if $\overrightarrow{G}$ is an I-map for p.

Theorem

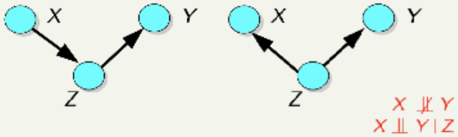
Let $p(X)$ be a Bayesian network with the graph $\overrightarrow{G} = (X, E)$,

$$p(X) = \prod_{X_i \in X} p\left(X_i \mid \Pi_{X_i}\right)$$

Markov Equivalence class

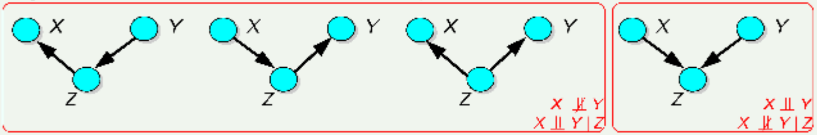
Markov Equivalence

Two Bayesian networks are equivalent if their graphs represent the same independence model.



Definition (Markov Equivalence class)

The Markov equivalence class for a Bayes net represented by $\overrightarrow{G}$ is the set of all Bayesian networks that are Markov equivalent to $\overrightarrow{G}$.


X
Y
Z
X
Y
Z
X
Y
Z
$X \not\perp\!\!\!\perp Y$
$X \perp\!\!\!\perp Y \mid Z$
X
Y
Z
$X \perp\!\!\!\perp Y$
$X \not\perp\!\!\!\perp Y \mid Z$

Causality

Causality and probability

Conditional probabilities do not allow to deal with causality. They even create paradox.

Simpson's paradox

Impact analysis : does a certain drug help to cure the patients ?

With the values : c^1 (cured) / d^1 (with drug) / d^0 (no drug) / M,W (man, woman).

- $P(c^1|d^1) = 0.575 > P(c^1|d^0) = 0.5$ the drug helps ...
- $P(c^1|d^1, M) = 0.7 < P(c^1|d^0, M) = 0.8$ ... except if the patient is a man ...
- $P(c^1|d^1, W) = 0.2 < P(c^1|d^0, W) = 0.4$ ... or a woman.

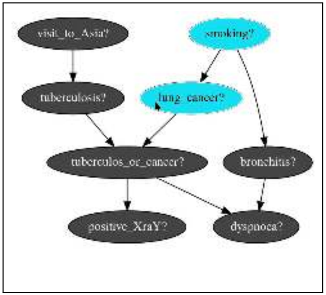
⚠ The conditional probability $P(c^1|d^1)$ is observational and is not relevant, one wants to give the drug and not to observe : intervention on d : $P(c^1| \hookrightarrow d^1)$

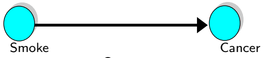
Conditioning by intervention

Let I be the state of the light switch, cause of L : 'is there light in the room ?'.

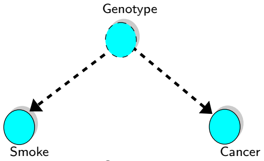
- With observational conditioning $P(L|I)$ and $P(I|L)$ (no distinction between cause and effect),
- $P(L| \hookrightarrow I) = P(L|I)$
- $P(I| \hookrightarrow L) = P(I)$.

Causal Bayesian network

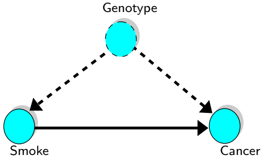




$$P(C \mid\hookrightarrow S) = P(C \mid S)$$

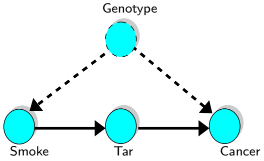


$$P(C \mid\hookrightarrow S) = P(C)$$



$P(C \mid\hookrightarrow S)$ unknown !

We do not know how to differentiate the causal impact of the 2 causes from the observation.

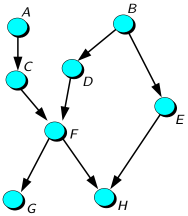


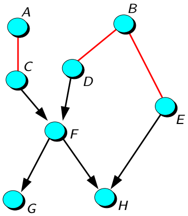
$P(C \mid\hookrightarrow S)$ computable ! Do-Calculus

Markov equivalence class, essential graph

Definition (essential graph)

A Markov equivalence class can be represented by a partially directed acyclic graph (PDAG) : the essential graph.
In an essential graph, an arc is directed if every Bayesian network in the equivalence class have the same arc.





The essential graph can be build from a BN by removing orientation of the arcs that can be reversed without creating or removing a collider.

⚠ if the BN is causal, the essential graph indicates the causal arc that can be learned from a dataset. Undirected arc is an indication for a (possible) latent cause.

Learning

What can we learn ?

Learning in Bayesian network

The goal of a learning algorithm is to estimate from a dataset and from prior :

- the structure of the Bayesian network (is X parent of Y ?)
- the parameters of the Bayesian network ($P(X = 0 \mid Y = 1)$?)

The dataset can be :

- complete,
- incomplete (missing values).

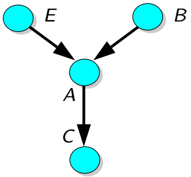
The prior knowledge can be for instance :

- (part of) the structure of the BN,
- The probability distribution for certain variables, etc.

Therefore 4 main classes of learning algorithms in BN :
"Learning of {parameters |structure} with {complete |incomplete} data".

Learning parameters with complete data

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



Let Θ be the set of all parameters for the model and $L(\Theta : D)$ the likelihood :

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

Learning parameters with complete data (2)

Let us rename E, B, A, C with $n = 4$, $(X_i)_{1 \le i \le n}$,

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

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

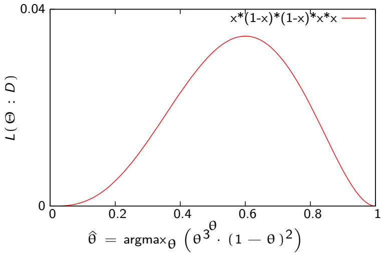
The estimation of the parameters can be decomposed into the estimation of the different conditional probability table for each node.
No need for only one global dataset : heterogeneous learning.

Maximization of the likelihood for a single variable

Let X be a binary variable. With $\theta = P(X = 1)$:

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

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



Generalization

For X random variable which can take the values $(1, \cdots, r)$,
with $\Theta_X = (\theta_1, \cdots, \theta_r)$ where $\theta_i = P(X = i)$,
and $N_i = \#_D(X = i)$ number of occurrences of i in the dataset D,

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

Maximizing the likelihood in a Bayesian network

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

- $L(\Theta : D) = \prod_{i=1}^{n} L_i(\Theta_i : D) = \prod_{i=1}^{n} \prod_{m=1}^{M} P(X_i = k_m \mid Pa_i = j_m, \Theta_i)$
$$L(\Theta : D) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}$$
- $LL(\Theta : D) = \sum_{i=1}^{n} \sum_{m=1}^{M} \log P(X_i \mid Pa_i, \Theta_i) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \theta_{ijk}$
- $\sum_k \theta_{ijk} = 1$ then $\theta_{ijr_i} = 1 - \sum_{k=1}^{r_i - 1} \theta_{ijk}$ hence
$$LL(\Theta : D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left(\sum_{k}^{r_i - 1} N_{ijk} \log \theta_{ijk} + N_{ijr_i} \log \left(1 - \sum_{k=1}^{r_i - 1} \theta_{ijk} \right) \right)$$
- We are looking for $\widehat{\Theta}$ that maximizes $L(\Theta : D)$ and then $LL(\Theta : D)$:
$$\text{i.e. } \widehat{\Theta} \text{ tel que } \forall i, \forall j, \forall k, \frac{\partial LL(\Theta : D)}{\partial \theta_{ijk}} \left(\widehat{\Theta} \right) = \frac{N_{ijk}}{\hat{\theta}_{ijk}} - \frac{N_{ijr_i}}{1 - \sum_{k=1}^{r_i - 1} \hat{\theta}_{ijk}} = \frac{N_{ijk}}{\hat{\theta}_{ijk}} - \frac{N_{ijr_i}}{\hat{\theta}_{ijr_i}} = 0$$
- Finally, $\frac{N_{ijr_i}}{\hat{\theta}_{ijr_i}} = \frac{N_{ij1}}{\hat{\theta}_{ij1}} = \cdots = \frac{N_{ij(r_i - 1)}}{\hat{\theta}_{ij(r_i - 1)}}$ (and $\sum_k \hat{\theta}_{ijk} = 1$) :

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

Bayesian prediction

The a posteriori distribution of Θ is $P(\Theta \mid D)$.

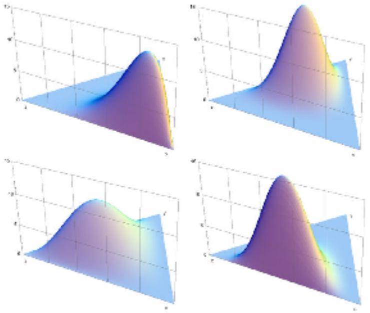
$$P(\Theta \mid D) \propto P(D \mid \Theta) \cdot P(\Theta) = L(\Theta : D) \cdot P(\Theta)$$

Goal to take into account a *prior* on Θ in order to integrate experts knowledge or to stabilize the estimation if the dataset is too small.
The Dirichlet distribution is the conjugate prior of the categorical distribution and multinomial distribution.

Dirichlet Distribution :
$f(p_1, \cdots, p_K; \alpha_1, \cdots \alpha_K) \propto \prod_{i=1}^{K} x_i^{\alpha_i - 1}$
(where $\sum_i p_i = 1$)
f can be interpret as :
$P\,(P(X = i) = p_i \quad \#_{X=i} = \alpha_i - 1)$

If the prior $P(\Theta)$ is a Dirichlet distribution :

$$P(\Theta) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}} \quad 1$$



[Wikipedia] Clockwise from top left : α $(6, 2, 2), (3, 7, 5), (6, 2, 6), (2, 3, 4)$.

Bayesian prediction (2)

From :

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

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

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 : expectation a posteriori

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

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

Parameters learning with complete data

With N_{ijk} the count of occurrences in the dataset where variable X_i has the value k and its parents have the values (t-uple) j, With α_{ijk} the parameters of a Dirichlet prior.

Parameters estimation

Two main solutions :

- MLE (Maximum Likelihood Estimation)

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

- Bayesian estimation (with Dirichlet *prior)*

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

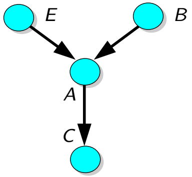
$$\widehat{\theta}^{EAP}_{ijk} = \widehat{\theta}_{\{x_i=k|pa_i=j\}} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}$$

- *Prior* important when $N_{ijk} \to 0$: no occurrence in the dataset.
- All these estimations are equivalent when $N_{ijk} \to \infty$

⚠ α_{ijk} are often hard to find $\Rightarrow$Laplace smoothing : $\alpha_{ijk} = constant (= 1)$

Learning parameters with missing values

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



$D = D^o \cup D^h$ respectively observed data and unobserved.

Typology for incomplete dataset
With $\mathcal{M}_{il} = d_l^i \in D^h$

- 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).

EM for BNs

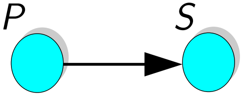
Expectation-Maximization for BNs

Repeat until convergence

Step E : Estimate $N_{ijk}^{(t+1)}$ from $P(X_i|Pa_i, \theta_{ijk}^t)$

inference in the BN with parameters θ_{ijk}^t

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



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

Parameters

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

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

EM in a BN : example

0. **Initialisation**
We have to choose a initial value for each parameters : $\theta^{(0)}_{S|P=o} = 0.3, \quad \theta^{(0)}_{S|P=n} = 0.4$

1. Step E using $\theta^{(0)}$

		$P(S \mid P=o)$		$P(S \mid P=n)$	
P	S	$S=o$	$S=n$	$S=o$	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

Step M

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

2. Step E using $\theta^{(1)}$

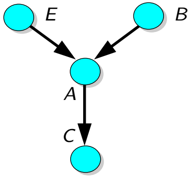
		$P(S \mid P=o)$		$P(S \mid P=n)$	
P	S	$S=o$	$S=n$	$S=o$	S=n
o	?	0.433	0.567	0	0
n	?	0	0	0.2	0.8
o	n	0	1	0	0
n	n	0	0	0	1
o	o	1	0	0	0
	N^*	1.433	1.567	0.2	1.8

Step M

$$\theta^{(2)}_{S|P=o} = \frac{1.433}{1.433+1.567} = 0.478 \quad \text{and} \quad \theta^{(2)}_{S|P=n} = \frac{0.2}{0.2+1.8} = 0.1$$

3. etc.
$(\theta^{(t)}_{S|P=o} \rightarrow 0.5 \quad \text{and} \quad \theta^{(t)}_{S|P=n} \rightarrow 0)$

EM in BNs



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

What is the step E ?

- Replace the ? by $P(A \mid B = 0, C = 1, E = 0)$
- $\Rightarrow$**inference in the BN with the parameters** Θ^t

Learning parameters with missing values

- EM converges to a local optimum,
- Sensibility to initial parameters
- Each step E can be expensive (as an inference in a complex BN)

Structural learning with complete data

- **Goal** : learning the arcs of the graph from data.
- **Theoretically** : χ^2 test plus enumeration of all the possible models : OK
- **In practice** : many different problems but above all :

Set of Bayesian networks (Robinson, 1977)

The number of different possible structures for n random variables is super-exponential.

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

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

An algorithm 'brute-force' is not feasible. The space of Bayesian networks it too large : $NS(10) \approx 4.2 \cdot 10^{18}$!

Structural learning - introduction

General picture of structural learning

- Identification of symmetrical relation (independence) + orientation
  - algorithm IC/PC
  - algorithm IC*/FCI

  Important for causal models.
- Local search
  - In the (very large) space of structures,
  - Greedy algorithms maximizing a score (entropy, AIC, BIC, MDL,BD, BDe, BDeu, $\cdots$).

Identification of symmetrical relation

Statistically, the relation that can be tested between variables are symmetrical : correlation or independence.
However, once these symmetrical relations have been found, other tests (conditional independence) can lead to the discovery of colliders which force some orientation.

Main principle for (IC, IC*, PC, FCI)

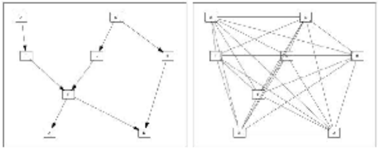
1. Build an undirected graph based on symmetrical relation statistically found (χ^2, correlation, mutual information, etc.) :
   - Add edges from the empty graph.
   - Remove edges from the complete graph.
2. Identify colliders and add the implied orientations .
3. Finalize the orientations without creating any other colliders (in order to stay in the same Markov equivalence class.

Major drawback : a very large number of statistic tests is needed. Each test is not robust in the size of the dataset.

Example : PC

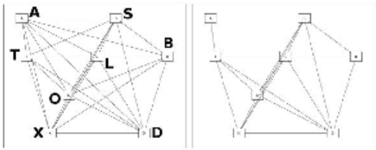
- Let P be a BN. We generate a dataset of 5000 lines compatible with P.[1]

Etape 0 : Graphe non orienté reliant tous les nœuds.



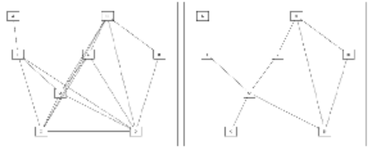
- Using χ^2 test, every possible marginal independence ($X \perp\!\!\!\perp Y$) is tested. Then every remaining possible conditional independence ($X \perp\!\!\!\perp Y \mid Z$) are tested. Then with 2 parents, etc.

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



We find : $A \perp\!\!\!\perp S$, $L \perp\!\!\!\perp A$, $B \perp\!\!\!\perp A$, $O \perp\!\!\!\perp A$, $X \perp\!\!\!\perp A$, $D \perp\!\!\!\perp A$, $T \perp\!\!\!\perp S$, $L \perp\!\!\!\perp T$, $O \perp\!\!\!\perp B$, $X \perp\!\!\!\perp B$.

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



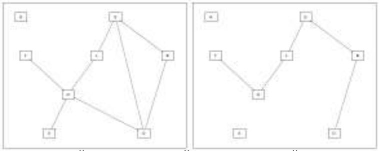
We find : $T \perp\!\!\!\perp A \mid O$, $O \perp\!\!\!\perp S \mid L$, $X \perp\!\!\!\perp S \mid L$, $B \perp\!\!\!\perp T \mid S$, $X \perp\!\!\!\perp T \mid O$, $D \perp\!\!\!\perp T \mid O$, $B \perp\!\!\!\perp L \mid S$, $X \perp\!\!\!\perp L \mid O$, $D \perp\!\!\!\perp L \mid O$, $D \perp\!\!\!\perp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

1. from Philippe Leray

Example : PC

- We continue with χ^2 conditioned on 2 variables.

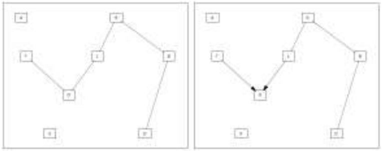
Etape 1c : Suppression des ind. conditionnelles d'ordre 2



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

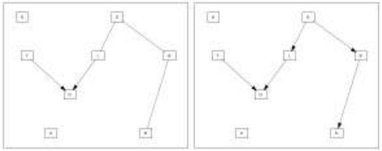
- Looking for colliders, propagation of orientation constraints, orientation of the last edges in the Markov equivalence class.

Etape 2 : Recherche des V-structures



We find : $T \not\perp\!\!\!\perp L$ and $T \perp\!\!\!\perp L \mid O$

Etape 4 : Instanciation du PDAG



Orientation in the same Markov equivalence class.

- Conclusion : with 5000 lines, PC has lost many information (noisy χ^2 tests).

Structural learning with local search

Local search

A local search is composed of :

- a space of all possible solutions (search space),
- a neighborhood defined by elementary transformations of a solution.The neighbors of a solution are the solutions that can be obtained by the application of an elementary transformation.
- a score (heurisitic) that evaluates the quality of a solution.

From an initial solution, the local search then produces a sequence of solutions such that every solution in the sequence has a better score than the precedent solutions in the sequence (*Greedy Search*).

Local search in Bayesian networks

- space of Bayesian networks (huge)
- The score (see next slide)
- The initial solution (empty Bayesian network for instance)
- Elementary operations : add/remove/reverse an arc

Scores

Properties for a score

Let D a dataset, T the graph of the current solution and Θ its parameters.A score must satissfy :

1. **Likelihood** : The solution must explain the data ($\max L(T, \Theta : D)$).
2. **Occam's razor** : The score must prefer simple graphs for T rather than complex ones ($\min Dim(T)$).
3. **Local consistency** : Adding a useful arc should increase the score.
4. **Score equivalence** : Two Markov-equivalent Bayesian networks should have the same score.

Definition ($Dim(T)$)

The dimension of a Bayesian network is its number of free parameters.

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

where r_i is the size of the variable X_i and q_i is the number of configurations for the parents of X_i.

Maximizing the likelihood is not a good score : it leads to a complete graph : overfitting.

Some scores (1) : AIC/BIC

Key idea : Maximizing likelihood but minimizing the dimension of the Bayesian network.

score AIC (Akaike, 70)

- Akaike Information Criterion

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

score BIC (Schwartz, 78)

- Bayesian Information Criterion

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

Minimum Description Length (Rissanen,78)

The MDL score consists in consider the compacity of the representation as a good indicator for the quality of the solution.

score MDL (Lam and Bacchus, 93)

- Minimum Description Length

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

where arcs_T is the set of arcs in T, c is the number of bits needed to represent a parameter.

Bayesian Dirichlet score Equivalent

With a Bayesian score, we want to maximize the joint probability of T and D :

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

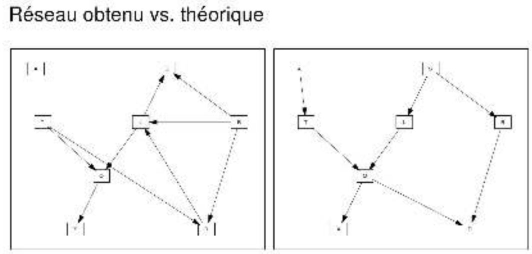
With some independence hypothesis and a Dirichlet *prior* :

score BDe

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

Recherche locale : Example

With the same dataset.



Major drawbacks : ● The algorithm may be trapped in zones where all the neighborhood has the same score.
● The algorithm may stop in local minima.

Solutions

Meta-heuristic :

- Random restart (not only from empty Bayesian networks)
- TABU-search (force the algorithm to find new solutions)
- Simulated annealing (accept from time to time structures with decreasing score)

Learning trees

In order to reduce the state space, this algorithm can find only one parent for each random variables.
This may be an over-simplification of the model, but learning tree brings :

- a mathematically beautiful solution (find the global optimum),
- a small number of parameters ($\Rightarrow$minimize the risk of overfitting).

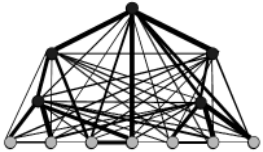
Basic idea : decomposition of the log-likelihood

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

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

Learning optimal tree

- $\forall X, Y$, compute $LL(Y \leftarrow X)$
- Find the tree (forest) that maximize $LL(T)$.
  Max Spanning Tree Algorithm – $O(n^2 \cdot log(n))$



Inference

Inference in a Bayesian network

- Elementary operations on a joint probability :

Marginalization	$\sum_{y} P(x, y \mid z) = p(x \mid z)$
Total sum	$\sum_{y} P(y \mid z) = 1$
Decomposition	$P(x, y \mid z) = P(x \mid y, z) \cdot P(y \mid z)$
Chain rule	$P(X_1, \cdots, X_n) = \prod_{i=1}^{n} P\left(X_i \mid X_1, \cdots, X_{i-1}\right)$
Independence	$X \perp\!\!\!\perp Y \mid Z \Rightarrow P(x \mid y, z) = P(x \mid z)$
Bayes rule	$P(x \mid y, z) \propto P(y \mid x, z) \cdot P(x \mid z)$

- In a Bayesian network :

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

Hard and soft evidence

Let $P(X_1, \cdots, X_n)$ be a Bayesian network, let ϵ be an event.

$$P(X_1, \cdots, X_n|\epsilon) \propto P(\epsilon|X_1, \cdots, X_n) \cdot P(X_1, \cdots, X_n)$$

Evidence in a Bayesian network

Let assume that $\exists(\epsilon_i)_{I \subset \{1, \cdots, n\}}$ s.t. :

- $\epsilon = \bigcap_{i \in I} \epsilon_i$
- $\forall i \in I, \epsilon_i \perp\!\!\!\perp X_1, \cdots, X_{i-1}, X_{i+1}, \cdots, X_n \,|\, X_i$

$$\text{then } P(X_1, \cdots, X_n, \epsilon) = \prod_{i=1}^{n} P(x_i|\pi_i) \cdot \prod_{i \in I} P(\epsilon_i|X_i)$$

- if $P(\epsilon_i|X_i)$ contains a 1 and many 0, ϵ_i is called a hard evidence. ϵ_i is the event "X_i takes a certain value".
- otherwise, ϵ_i is called a soft evidence.

ϵ_i acts like 'virtual child' of X_i. Evidence will not appear in the following since they belong to the general framework.

Inference in a Bayesian network (1) : $P(D)$?

$$\begin{tikzcd}
 & & C \\
A \arrow[r] & B \arrow[ur] \arrow[dr] & \\
 & & D
\end{tikzcd}$$

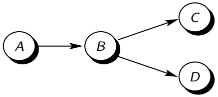
$$
\begin{aligned}
P(d) &= \sum_a \sum_b \sum_c P(a, b, c, d) \\
&= \sum_a \sum_b \sum_c P(a) \cdot P(b \mid a) \cdot P(c \mid b) \cdot P(d \mid b) \\
&= \sum_a \sum_b P(a) \cdot P(b \mid a) \cdot P(d \mid b) \cdot \underbrace{\left(\sum_c P(c \mid b) \right)}_{=1} \\
&= \sum_b \underbrace{P(d \mid b)}_{\text{in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

Inference in a Bayesian network (2) : $P(D \mid \underline{a})$?

$$\begin{tikzcd}
 & & C \\
A \arrow[r] & B \arrow[ur] \arrow[dr] & \\
 & & D
\end{tikzcd}$$

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

Inference in a Bayesian network (3) : $P(C|\overline{d})$?

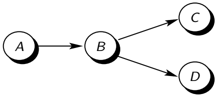


$$P(c \mid \overline{d}) = \sum_a \sum_b \sum_d P(a \mid \overline{d}) \cdot P(b \mid a, \overline{d}) \cdot P(c \mid b, \overline{d}) \cdot P(d \mid b, \overline{d})$$
$$= \sum_b \underbrace{P(c \mid b)}_{\text{in } C} \cdot \underbrace{\sum_a P(a \mid \overline{d}) \cdot P(b \mid a, \overline{d})}_{P(b|\overline{d})}$$

Bayes rule for $P(b \mid \overline{d})$

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

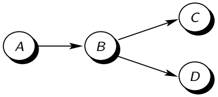
Inference in a Bayesian network (4) : $P(A|\overline{d})$?



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

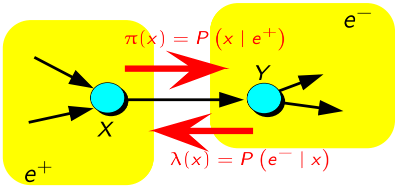
$$\begin{aligned} P(\overline{d} \mid a) &= \sum_b P(\overline{d} \mid a, b) \cdot \underbrace{P(b \mid a)}_{\text{in } B} \\ &= \sum_b \underbrace{P(\overline{d} \mid b)}_{\text{in } D} \cdot \underbrace{P(b \mid a)}_{\text{in } B} \end{aligned}$$

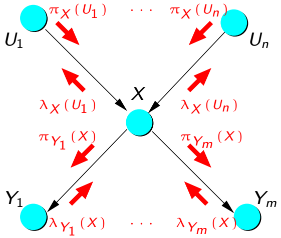
Inference in a Bayesian network (5) : $P(B \mid \overline{c}, \underline{a})$?



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

Inference in polytree





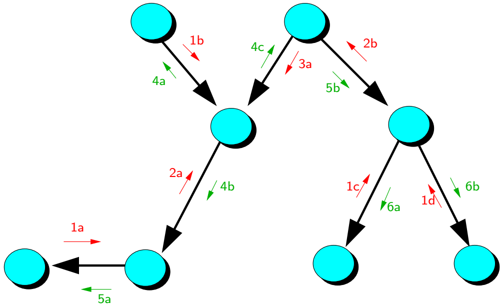
If a node has n neighbors (parents or children), he must know $n-1$ messages in order to send its message to the last neighbor.

Once all messages have been sent $(2 \cdot |E|)$, every node knows all the information received by the graph.

If a node has n neighbors (parents or children), he must know n messages in order to compute its own posterior.

A first algorithm for inference in polytree

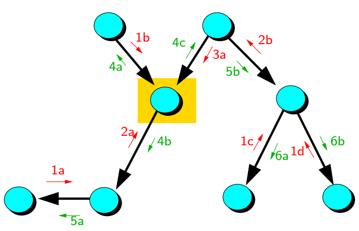
- Propagation : repeat
  - for all node N with n neighbors,
    - if N received $n-1$ messages then N can send the message to the last neighbor.
    - if N received n messages then N can send all its messages.
- until all messages have been sent
- Every node can compute its posterior.

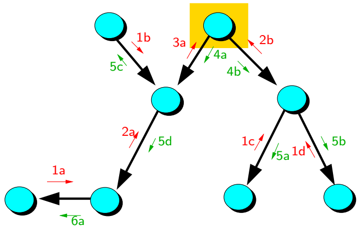


Complexity

$O(|E|)$

Inference in polytree (2)



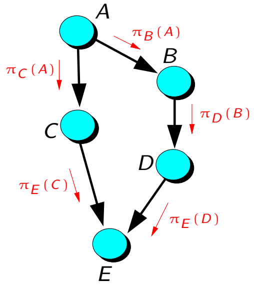


Centralized version

- Selection of a root
- Absorption
  Each node send the message to the root (as soon as he can).
- **Intégration**
  The root has received all its messages from its neighbors and can send all the messages to its neighbors.
- Diffusion
  When a node receives its message from the root, it sends all the remaining messages.

Problem with message passing algorithms in DAG

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



Message passing algorithm :

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. $P(E) = \sum_{D,C} P(E \mid C, D) \cdot \pi_E(C) \cdot \pi_E(D)$

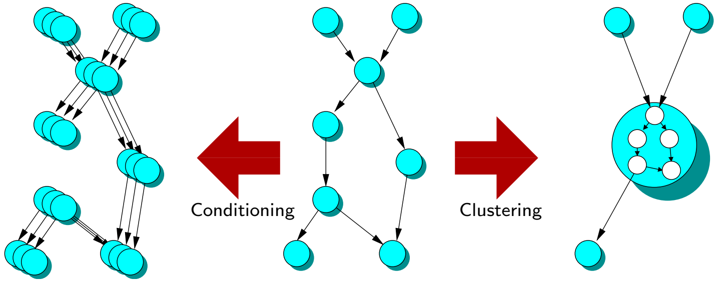
$$P(E) \neq \sum_{A,B,C,D,A'} P(E \mid C, D) \cdot P(C \mid A) \cdot P(A) \cdot P(D \mid B) \cdot P(B \mid A') \cdot P(A')$$

Inference in DAG

Message passing algorithms need a polytree.

From DAG to polytree

- Conditioning : cutting arcs in the graph.
- Clustering : Merging nodes in the graph.

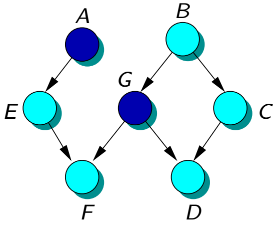


Conditioning

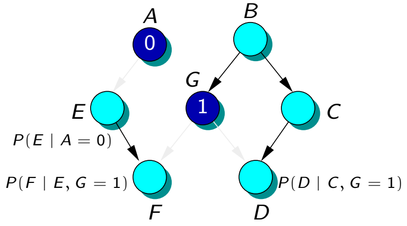
Conditioning a Bayesian network

G a Bayesian network over the set of random variables V, $S \subset V$ and s an instantiation of S.
The conditioned Bayesian network $G^{[S=s]}$ is the Bayesian network obtained by :

- removing arcs from every node in S
- if the node Y has parent(s) in S then $p(Y|\Pi_Y)$ is changed in $p(Y|s,\Pi_Y)$.



Graph G



Conditioned graph $G^{[A=0,G=1]}$

An inference in G with evidence $S = s$ is equivalent to an inference in $G^{[S=s]}$ with no evidence.

S is called a cutset.

Inference by conditioning

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

Inference by conditioning

With G Bayesian network,

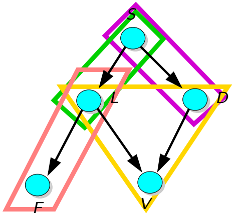
1. find $S \subset V$ cutset such that $G^{[S]}$ is a polytree.
2. $\forall s$ instantiation of S,
    - Compute $P_s(x) = P(x \mid s)$ in $G^{[S]}$.
    - Compute $p_s = P(s)$ (secondary result of the last inference).
3. $P(x) = \sum_s (p_s \cdot P_s(x))$

If $\#_i$ is the size of S_i, the inference computes $\prod_i \#_i$ inferences in $G^{[S]}$.

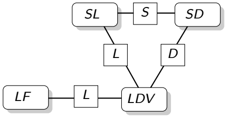
$$\Rightarrow O(k^{|S|} \cdot |E^{[S]}|)$$

Finding the minimal cutset is NP-complete.

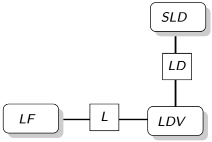
Clustering : Junction tree algorithm



$$
\begin{aligned}
&P(s,l,d,f,v)\\
&= P(s)\cdot P(l\mid s)\cdot P(d\mid s)\cdot P(f\mid l)\cdot P(v\mid d,l)\\
&= f(s,l)\cdot g(s,d)\cdot h(l,f)\cdot k(d,l,v)\\
&= j(s,l,d)\cdot h(l,f)\cdot k(d,l,v)
\end{aligned}
$$



Junction graph



Junction tree

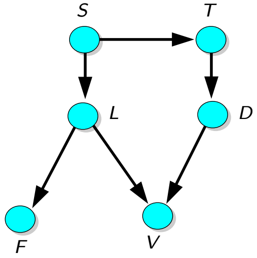
How to build a junction tree

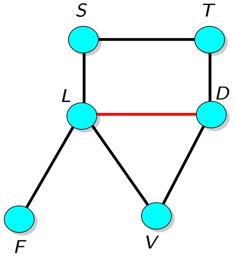
Idea : create an undirected graph from the Bayesian network. The cliques of this undirected graph will be the nodes of the junction tree.

The CPTs $P(X \mid Parent_X)$ of the Bayesian network indicate necessary clusters.

Moralization

The moral graph of a BN is the undirected skeleton of the Bayesian network to which edges between parents of the same node are added.





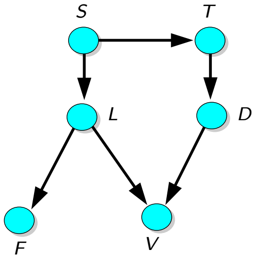
How to build a junction tree (2)

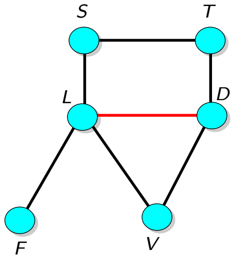
To insure the existence of the junction tree, the undirected graph must be triangulated : every cycle with length>3 must have be chordal.

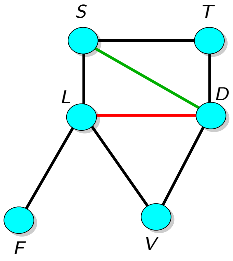
Triangulation

To triangulate a graph : variables elimination :

- Iteratively remove all nodes in the graphs.
- When a node is removed, add edge between every pair of its neighbors
- Start to eliminate nodes that will not create new edge (no neighbor, only one neighbor, neigbours already connected).







Elimination order : F, V, T, S, L, D.

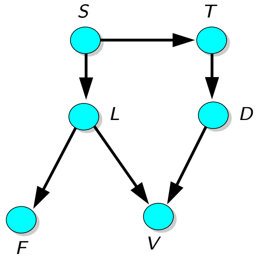
To find the best elimination order is NP-complete.

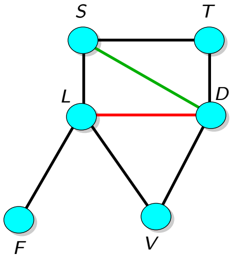
How to build a junction tree (3)

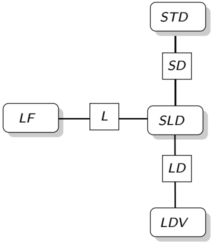
The nodes of the junction tree are the cliques of the moralized and triangulated graph, what about its edges ?

Junction tree

- Choose an order in the cliques (C_i).
- for each C_i, find the clique C_j ($j < i$) that maximize $|C_i \cup C_j$.
- Add an edge between C_i and C_j.
- Create the separator $S_{ij} = C_i \cup C_j$.

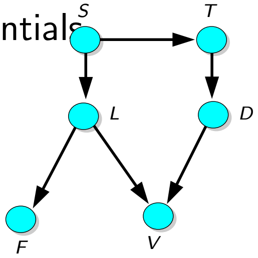


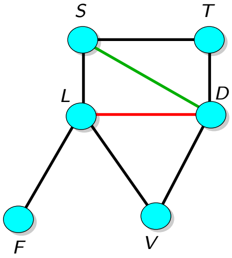


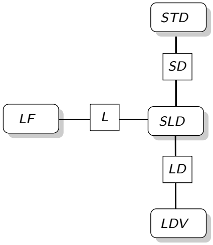


The junction tree is not unique for a Bayesian network.

Potentials







Decomposition of $P(V)$ following the cliques :

$$P(v) = \underbrace{P(s)\cdot P(l \mid s)}_{\phi_1(d,l,s)} \cdot \underbrace{P(t \mid s)\cdot P(d \mid t)}_{\phi_2(s,t,d)} \cdot \underbrace{P(f \mid l)}_{\phi_3(f,l)} \cdot \underbrace{P(v \mid l,d)}_{\phi_4(v,l,d)}$$

Another decomposition :

$$P(v) = \frac{P(s,t,d)\cdot P(s,l,d)\cdot P(l,f)\cdot P(l,d,v)}{P(s,d)\cdot P(l)\cdot P(l,d)}$$

Factorization of P in a Junction Tree

$$P(v) = \frac{\prod_i \Phi_{C_i}(c_i)}{\prod_{i<j} \Phi_{S_{ij}}(s_{ij})}$$

Propagation on potentials in the junction tree

goal : Transform the potential of all cliques ($\mathcal{C}$) and separators ($\mathcal{S}$) into joint probability of their variables.

Message passing in the Junction Tree

- **Initialization** :
  $\forall C_i \in \mathcal{C}$,
  $$\Psi^0_{C_i} = \prod_{X \in C_i, X \notin C_j, j<i} P(X \mid \Pi_X) \qquad \text{(and the evidence)}$$
  $\forall S \in \mathcal{S}$, $\Psi^0_S = 1$ (constant function).
- **Root selection** : choose a clique as a root for the propagation
- **Collect** : send all messages from all cliques to the root (a clique can send a message if it receives all the others).
- **Distribution** : send all messages from the root to all cliques

Message from a clique C_i to a clique C_j : update $\Psi^t_{C_j}$ from $\Psi^{t+1}_{C_i}$:

$$\Psi^{t+1}_{S_{ij}}(s) = \sum_{C_i \setminus S_{ij}} \Psi^{t+1}_{C_i}(c) \qquad \text{and} \qquad \Psi^{t+1}_{C_j} = \Psi^t_{C_j} \cdot \frac{\Psi^{t+1}_{S_{ij}}}{\Psi^t_{S_{ij}}}$$

- There still is only a linear complexity in number of arcs, but the clique and the Ψ_C may be very huge.
- This algorithm is the most used for exact inference in Bayesian network.

Copulas Bayesian Network

Copulas

Let $X = \{X_1, \cdots, X_n\}$ be a set of continuous random variables, its joint CDF $F(x) = P(X \leq x)$, its density of probability $p(x)$: $p(x) = \frac{\partial^n F}{\partial x_1 \cdots \partial x_n}(x_1, \cdots, x_n)$.

Copula

$C(u_1, \cdots, u_n) =$ CDF for the variable $U_1, \cdots, U_n$ uniformly distributed on $[0, 1]$

Sklar,1959 $\forall F$, $\exists C$ copula s.t. $F(x_1, \cdots, x_n) = C(F(x_1), \cdots, F(x_n))$: $C(u) = F(F_1^{-1}(u_1), \cdots, F_1^{-1}(u_n))$.

Unicity if continuity and $p > 0$

For the density : $p(x) = c(F_1(x_1), \cdots, F_n(x_n)) \prod_{i=1}^{n} p_i(x_i)$.

C integrate the structural complexity of the relation between variables and do not take into account their marginal behavior.

Copula $C(X)$ rarely used/computed/learned if $n > 10$

Copula Bayesian Networks

Let G be a Bayesian network on the density of probability :

$$p(X_1, \cdots, X_n) = \prod_{i=1}^{n} p(X_i|\pi_i)$$

Let c be a copula for p, we define, for each node X_i,

$$R_c(X_i, \pi_i) = \frac{c(F(X_i), F(P_1), \cdots, F(P_{k_i}))}{\frac{\partial^{k_i}}{\partial F(P_1)\cdots\partial F(P_{K_i})} c(1, F(p_1), \cdots, F(p_{K_i}))}$$

where $k_i = |\pi_i|$ (if $k_i = 0, Rc(X_i, \pi_i) = 1$).

$$p(X_1, \cdots, X_n) = \prod_{i=1}^{n} R_{c_i}(X_i, \pi_i) \cdot p(X_i)$$

Reciprocally, if $\{c_i\}_{i=1}^{n}$ are copulas (> 0) for each $(X_i, \pi_c i)$, then $\prod_{i=1}^{n} R_{c_i}(X_i, \pi_i)$ defines a copula for X.

$$c(F(X_1), \cdots, F(X_n)) = \prod^{n} R_{c_i}(X_i, \pi_i)$$

Copula Bayesian Network - Learning and inference

Definition (Copula Bayesian Network)

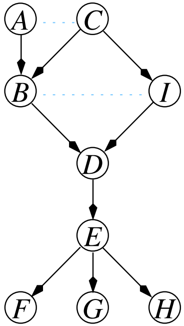
A CBN is described by a triplet (G, Θ_C, Θ_p) where G is a DAG over X, Θ_C is a set of copulas $c(X_i, \pi_i)$, Θ_p is the set of marginals $p(X_i)$.
Then the CBN has a joint density of probability $p(X)$ factorised by :

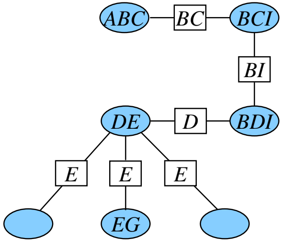
$$p(X_1, \cdots, X_n) = \prod_{i=1}^{n} R_{c_i}(X_i, \pi_i) \cdot p(X_i)$$

The same properties of locality hold for CBN as for BN.

- Large reduction of the number of parameters
- Parameter learning can be made locally on X_i, π_i
  - not the same dataset for each R_c.
  - not the same type of copula for each X_i, π_i
- Structural learning could use the same kind of algorithms than classical BN.
- Inference is simpified : simulation, Rosenblatts' transformation (etc)

Copula Junction Tree





$$p(X) = \frac{\prod_{C \in clique(G)} p(C)}{\prod_{S \in separator(G)} p(S)}$$

Let $c_S(S)$ be copula for all $S \in$ Junction Tree (clique or separator, with some restriction for consistency).

$$c(X) = \frac{\prod_{C \in clique(G)} c_C(C)}{\prod_{S \in separator(G)} c_S(S)} \text{ is a copula for } p(X)$$

Non parametric independence test : $Y \perp\!\!\!\perp Z | X$?

- Bouezmarni and al, 2009.
- Based on Hellinger distance between a joint copula and the product of marginal copulas,

$$H = \int_{[0,1]^{d+2}} \left(1 - \sqrt{\frac{c_{XY}(\boldsymbol{x}, y)c_{XZ}(\boldsymbol{x}, z)}{c_{XYZ}(\boldsymbol{x}, y, z)}}\right)^2 c_{XYZ}(\boldsymbol{x}, y, z) d\boldsymbol{x} dy dz,$$

- Every copula is approximated by a non parametric Bernstein's copula on a dataset of size N :

$$\widehat{H} \approx \frac{1}{N} \sum_{i=1}^{N} \left(1 - \sqrt{\frac{\widehat{c}_{XY}(\bar{F}_X(\boldsymbol{x}_i), F_Y(y_i))\widehat{c}_{XZ}(\bar{F}_X(\boldsymbol{x}_i), F_Z(z_i))}{\widehat{c}_{XYZ}(\bar{F}_X(\boldsymbol{x}_i), F_Y(y_i), F_Z(z_i))}}\right)^2.$$

$Y \perp\!\!\!\perp Z \mid X$?

[Bouezmarni and al, 2009] shows that under $H0 : [Y \perp\!\!\!\perp Z \mid X]$, $T \sim \mathcal{N}(0,1)$:

$$T \equiv \frac{N k^{-(d+2)/2}}{\sigma}\left(4H - N^{-1} C_1 k^{(d+2)/2} - N^{-1} B_1 k^{(d+1)/2} - N^{-1} B_2 k^{(d-\overset{1}{\cancel{2}})/2} - N^{-1} B_3 k^{d/2}\right)$$

with $k = \sqrt{d}$ (arbitrary) and :

$$C_1 = 2^{-(d+2)} \pi^{(d+2)/2}, \qquad \sigma = \sqrt{2}(\pi/4)^{(d+2)/2},$$

$$B_1 = -2^{-d} \pi^{(d+1)/2} + \frac{1}{N} \sum_{i=1}^{N} \frac{\prod_{j=1}^{d+1} (4\pi g_j^{(i)} (1 - g_j^{(i)}))^{-1/2}}{c_{XY}(g_1^{(i)}, \cdots, g_{d+1}^{(i)})},$$

$$B_2 = -2^{-d} \pi^{(d+1)/2} + \frac{1}{N} \sum_{i=1}^{N} \frac{4\pi (g_{d+2}^{(i)} (1 - g_{d+2}^{(i)}))^{-1/2} \prod_{j=1}^{d} (4\pi g_j^{(i)} (1 - g_j^{(i)}))^{-1/2}}{c_{XZ}(g_1^{(i)}, \cdots, g_d^{(i)}, g_{d+2}^{(i)})},$$

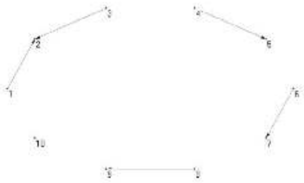
$$B_3 = 2^{-(d-1)} \pi^{-d/2} \frac{1}{N} \sum_{i=1}^{N} \frac{c_X(g_1^{(i)}, \cdots, g_d^{(i)})}{\sqrt{\prod_{j=1}^{d} g_j^{(i)} (1 - g_j^{(i)})}},$$

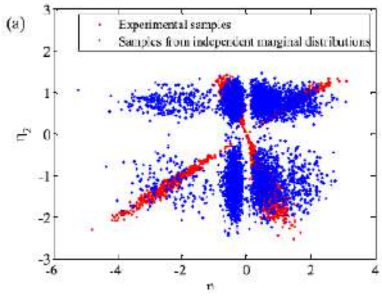
Structural Learning with CI test : PC Algorithm

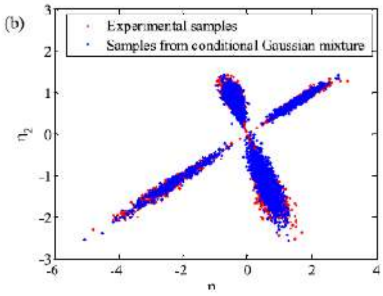
Algorithm 1 Construction of an undirected graphical model using CI tests.

```
• Start with a complete undirected graph $G = (V, E)$ where V is the node set and E is the edge set.
• Set conditional independence test order $n = 0$.
repeat
  for $Y \in V$ do
    for $Z \in Adjacencies(Y)$ do
      for $S \subseteq Adjacencies(Y) \setminus \{Z\}$ and $|S| = n$ do
        if $Y \perp Z | S$ then
          remove edge that connects Y and Z from E, and update the undirected graph $G$.
        end if
      end for
    end for
  end for
  $n = n + 1$
until $|Adjacencies(Y) \setminus \{Z\}| < n$ or $n = n_{max}$
```

Early results

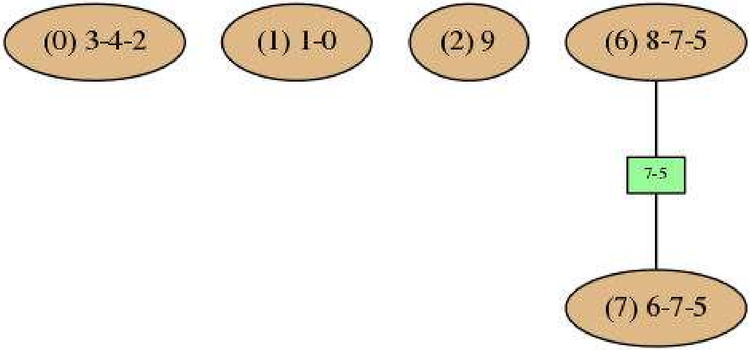






continuous-PC

Structural identification : `increasing_blocks`, 10 variables, N=10000, 27 minutes.



Conclusion :

- CBN proposes to explore the structure inside joint copulas,
- CBN may increase the number of dimensions for a joint copula,
- CBN may ease the different inference algorithms,
- CBN can be automatically learned from data using a non parametric CI test.