



Probabilistic Graphical Models, Classification, pyAgrum

Pierre-Henri WUILLEMIN, & Clara CHARON & Mahdi HADJ ALI

LIP6 & ex-LIP6

05/09/2023



Introduction

Réseaux bayésiens (discrets)

Definition

Inference

Applications

Use Cases

Learning Bayesian Networks

aGrUM/pyAgrum



DISCLAIMER



DISCLAIMER



- ▶ Niveaux de connaissances



DISCLAIMER



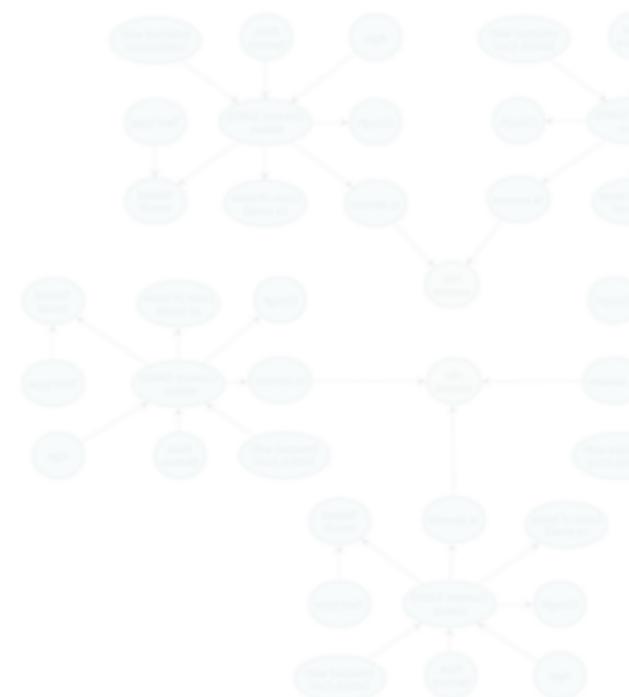
- ▶ Niveaux de connaissances
- ▶ Hypothèse : classification \in ppcc



DISCLAIMER



- ▶ Niveaux de connaissances
- ▶ Hypothèse : classification \in ppcc
- ▶ Classification versus BN



DISCLAIMER



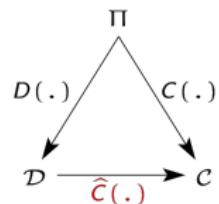
- ▶ Niveaux de connaissances
- ▶ Hypothèse : classification \in ppcc
- ▶ Classification versus BN
- ▶ ! Franglish in transparents



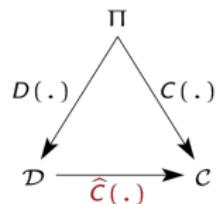
Classification in one slide



Classification in one slide



Classification in one slide

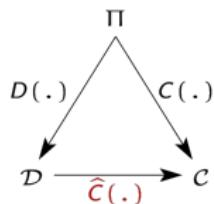


Classification (probabiliste)

$$\triangleright \hat{y} = \hat{C}(x) \approx C(D^{-1}(x)) = f(x)$$



Classification in one slide

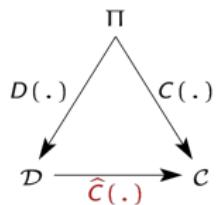


Classification (probabiliste)

- $\hat{y} = \hat{C}(x) \approx C(D^{-1}(x)) = f(x)$
- $P(x|y) = f(x)$



Classification in one slide

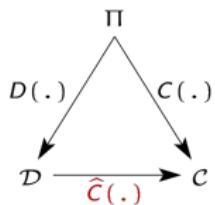


Classification (probabiliste)

- ▶ $\hat{y} = \hat{C}(x) \approx C(D^{-1}(x)) = f(x)$
- ▶ $P(x|y) = f(x)$
 - ▶ $\hat{y}_{ML} = \hat{C}(x) = \arg \max_y P(x|y)$
 - ▶ $\hat{y}_{MAP} = \hat{C}(x) = \arg \max_y P(y|x)$

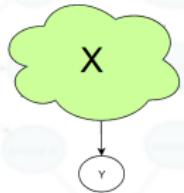


Classification in one slide

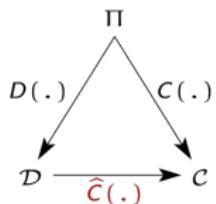


Classification (probabiliste)

- ▶ $\hat{y} = \hat{C}(x) \approx C(D^{-1}(x)) = f(x)$
- ▶ $P(x|y) = f(x)$
 - ▶ $\hat{y}_{ML} = \hat{C}(x) = \arg \max_y P(x|y)$
 - ▶ $\hat{y}_{MAP} = \hat{C}(x) = \arg \max_y P(y|x)$

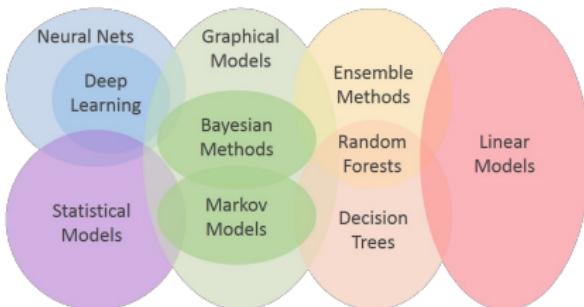
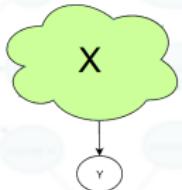


Classification in one slide

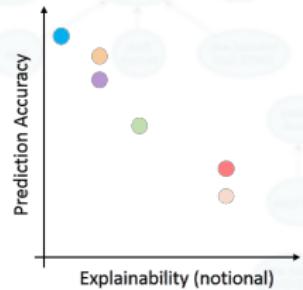


Classification (probabiliste)

- ▶ $\hat{y} = \hat{C}(x) \approx C(D^{-1}(x)) = f(x)$
- ▶ $P(x|y) = f(x)$
 - ▶ $\hat{y}_{ML} = \hat{C}(x) = \arg \max_y P(x|y)$
 - ▶ $\hat{y}_{MAP} = \hat{C}(x) = \arg \max_y P(y|x)$



Source : Data Science Lab - SIA partners



Paradoxe de Simpson - data



Paradoxe de Simpson - data



Patient	Drug	Gender	
0	Sick	With	F
1	Sick	Without	M
2	Healed	Without	M
3	Healed	With	F
4	Sick	With	M
...
1495	Healed	Without	M
1496	Healed	With	F
1497	Sick	With	F
1498	Healed	Without	F
1499	Sick	With	F

1500 rows × 3 columns

		Gender	
Patient	Drug	F	M
Healed	With	396	29
	Without	153	218
Sick	With	185	139
	Without	43	337

Paradoxe de Simpson - data



Patient	Drug	Gender	
0	Sick	With	F
1	Sick	Without	M
2	Healed	Without	M
3	Healed	With	F
4	Sick	With	M
...
1495	Healed	Without	M
1496	Healed	With	F
1497	Sick	With	F
1498	Healed	Without	F
1499	Sick	With	F

1500 rows × 3 columns

		Gender	
Patient	Drug	F	M
Healed	With	396	29
	Without	153	218
Sick	With	185	139
	Without	43	337

```
4 # pyAgrum
5 learner=gum.BNLearner("simpson.csv")
6 learner.pseudoCount(["Gender", "Drug", "Patient"])
```

Paradoxe de Simpson - calculs



Paradoxe de Simpson - calculs



		Patient	
Drug	Gender	Healed	Sick
With	F	0.2460	0.1093
	M	0.0200	0.1013
Without	F	0.1187	0.0220
	M	0.1540	0.2287

$P(\text{Patient}, \text{Gender}, \text{Drug})$



Paradoxe de Simpson - calculs



		Patient	
Drug	Gender	Healed	Sick
With	F	0.2460	0.1093
	M	0.0200	0.1013
Without	F	0.1187	0.0220
	M	0.1540	0.2287

$P(Patient, Gender, Drug)$

`p.normalize()`



Paradoxe de Simpson - calculs



		Patient	
Drug	Gender	Healed	Sick
With	F	0.2460	0.1093
	M	0.0200	0.1013
Without	F	0.1187	0.0220
	M	0.1540	0.2287

$P(Patient, Gender, Drug)$

`p.normalize()`

	Patient	
Drug	Healed	Sick
With	0.5580	0.4420
Without	0.5210	0.4790

$P(Patient|Drug)$



Paradoxe de Simpson - calculs



		Patient	
Drug	Gender	Healed	Sick
With	F	0.2460	0.1093
	M	0.0200	0.1013
Without	F	0.1187	0.0220
	M	0.1540	0.2287

$P(Patient, Gender, Drug)$

```
p.normalize()
```

		Patient	
Drug	Healed	Sick	
With	0.5580	0.4420	
Without	0.5210	0.4790	

$P(Patient|Drug)$

```
p.margSumOut("Gender")/p.margSumIn("Drug")
```

Paradoxe de Simpson - calculs



		Patient	
Drug	Gender	Healed	Sick
With	F	0.2460	0.1093
	M	0.0200	0.1013
Without	F	0.1187	0.0220
	M	0.1540	0.2287

$P(Patient, Gender, Drug)$

```
p.normalize()
```

		Patient	
Drug		Healed	Sick
With		0.5580	0.4420
Without		0.5210	0.4790

$P(Patient|Drug)$

```
p.margSumOut("Gender")/p.margSumIn("Drug")
```

$$0.5580 > 0.5210$$



Paradoxe de Simpson - calculs



		Patient	
Drug	Gender	Healed	Sick
With	F	0.2460	0.1093
	M	0.0200	0.1013
Without	F	0.1187	0.0220
	M	0.1540	0.2287

$P(Patient, Gender, Drug)$

```
p.normalize()
```

	Patient	
Drug	Healed	Sick
With	0.5580	0.4420
Without	0.5210	0.4790

$P(Patient|Drug)$

```
p.margSumOut("Gender")/p.margSumIn("Drug")
```

$$\begin{aligned} 0.5580 &> 0.5210 \\ \Rightarrow \textit{With} &\succ \textit{Without} \end{aligned}$$

Paradoxe de Simpson - calculs



		Patient	
Drug	Gender	Healed	Sick
With	F	0.2460	0.1093
	M	0.0200	0.1013
Without	F	0.1187	0.0220
	M	0.1540	0.2287

$P(Patient, Gender, Drug)$

```
p.normalize()
```

		Patient	
Drug	Healed	Sick	
With	0.5580	0.4420	
Without	0.5210	0.4790	

$P(Patient|Drug)$

```
p.margSumOut("Gender")/p.margSumIn("Drug")
```

$$0.5580 > 0.5210 \\ \Rightarrow \text{With} \succ \text{Without}$$

		Patient	
Drug	Gender	Healed	Sick
With	F	0.6923	0.3077
	M	0.1648	0.8352
Without	F	0.8436	0.1564
	M	0.4024	0.5976

$P(Patient|Gender, Drug)$

Paradoxe de Simpson - calculs



		Patient	
Drug	Gender	Healed	Sick
With	F	0.2460	0.1093
	M	0.0200	0.1013
Without	F	0.1187	0.0220
	M	0.1540	0.2287

$P(Patient, Gender, Drug)$

```
p.normalize()
```

		Patient	
Drug	Healed	Sick	
With	0.5580	0.4420	
Without	0.5210	0.4790	

$P(Patient|Drug)$

```
p.margSumOut("Gender")/p.margSumIn("Drug")
```

$$0.5580 > 0.5210 \\ \Rightarrow \text{With} \succ \text{Without}$$

		Patient	
Drug	Gender	Healed	Sick
With	F	0.6923	0.3077
	M	0.1648	0.8352
Without	F	0.8436	0.1564
	M	0.4024	0.5976

$P(Patient|Gender, Drug)$

```
p/p.margSumOut("Patient")
```

Paradoxe de Simpson - calculs



		Patient	
Drug	Gender	Healed	Sick
With	F	0.2460	0.1093
	M	0.0200	0.1013
Without	F	0.1187	0.0220
	M	0.1540	0.2287

$P(Patient, Gender, Drug)$

```
p.normalize()
```

		Patient	
Drug	Healed	Sick	
With	0.5580	0.4420	
Without	0.5210	0.4790	

$P(Patient|Drug)$

```
p.margSumOut("Gender")/p.margSumIn("Drug")
```

$$\begin{aligned} 0.5580 &> 0.5210 \\ \Rightarrow \textit{With} &\succ \textit{Without} \end{aligned}$$

		Patient	
Drug	Gender	Healed	Sick
With	F	0.6923	0.3077
	M	0.1648	0.8352
Without	F	0.8436	0.1564
	M	0.4024	0.5976

$P(Patient|Gender, Drug)$

```
p/p.margSumOut("Patient")
```

$$0.6923 < 0.8436$$

\Rightarrow sauf si F



Paradoxe de Simpson - calculs



		Patient	
Drug	Gender	Healed	Sick
With	F	0.2460	0.1093
	M	0.0200	0.1013
Without	F	0.1187	0.0220
	M	0.1540	0.2287

$P(Patient, Gender, Drug)$

```
p.normalize()
```

		Patient	
Drug	Healed	Sick	
With	0.5580	0.4420	
Without	0.5210	0.4790	

$P(Patient|Drug)$

```
p.margSumOut("Gender")/p.margSumIn("Drug")
```

$$\begin{aligned} 0.5580 &> 0.5210 \\ \Rightarrow \textit{With} &\succ \textit{Without} \end{aligned}$$

		Patient	
Drug	Gender	Healed	Sick
With	F	0.6923	0.3077
	M	0.1648	0.8352
Without	F	0.8436	0.1564
	M	0.4024	0.5976

$P(Patient|Gender, Drug)$

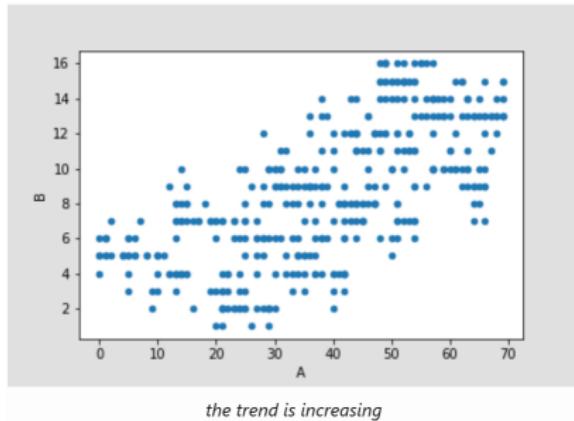
```
p/p.margSumOut("Patient")
```

$$\begin{aligned} 0.6923 &< 0.8436 \\ \Rightarrow \text{sauf si } F & \\ 0.1648 &< 0.4024 \\ \Rightarrow \text{et sauf si } H & \end{aligned}$$

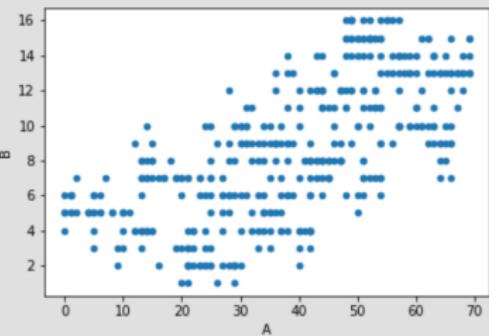
Paradoxe de Simpson - autre version



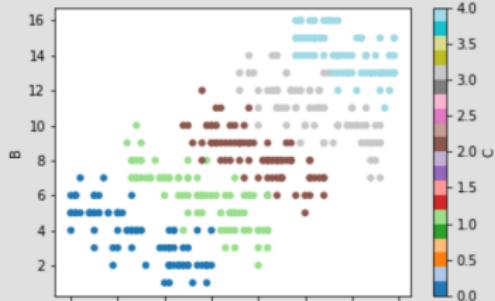
Paradoxe de Simpson - autre version



Paradoxe de Simpson - autre version



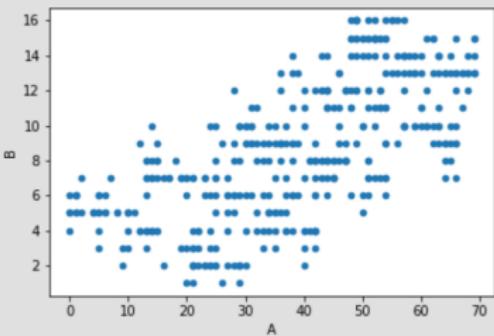
the trend is increasing



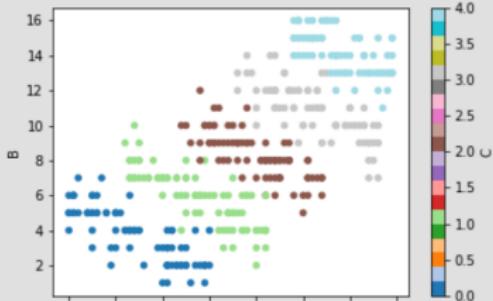
the trend is decreasing for any value for C !

Conclusions sur Simpson

Paradoxe de Simpson - autre version



the trend is increasing

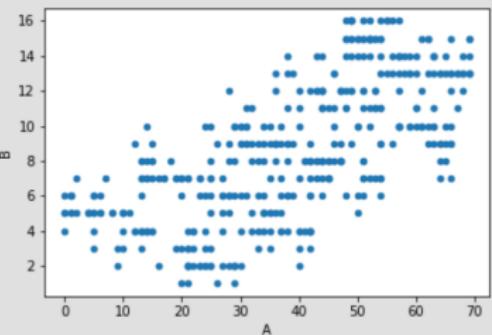


the trend is decreasing for any value for C !

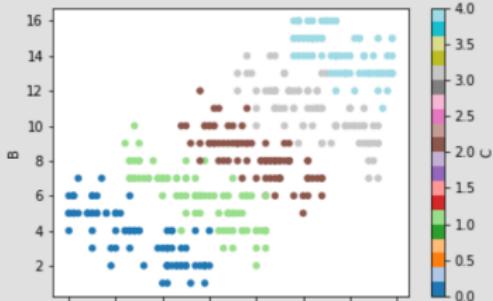
Conclusions sur Simpson

Quoi ?

Paradoxe de Simpson - autre version



the trend is increasing



the trend is decreasing for any value for C !

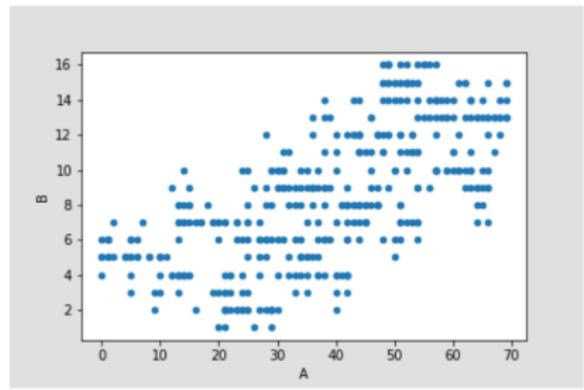
Conclusions sur Simpson



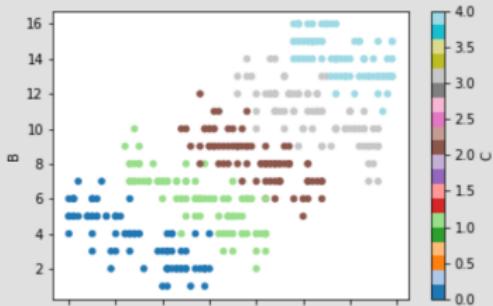
Quoi ?



Paradoxe de Simpson - autre version



the trend is increasing



the trend is decreasing for any value for C !

Conclusions sur Simpson

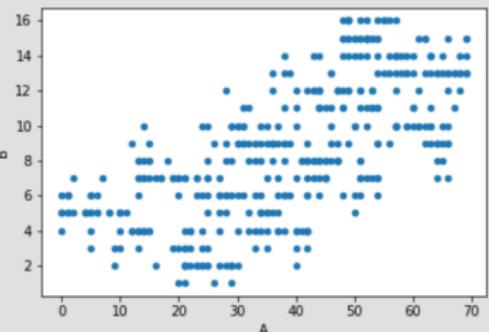


Quoi ?

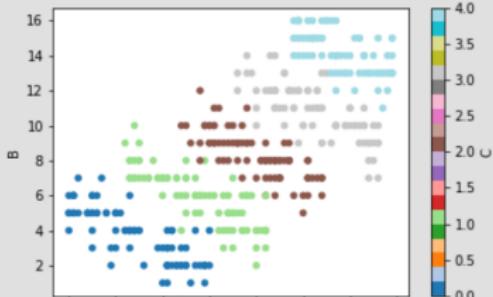
Comment ?



Paradoxe de Simpson - autre version



the trend is increasing



the trend is decreasing for any value for C !

Conclusions sur Simpson

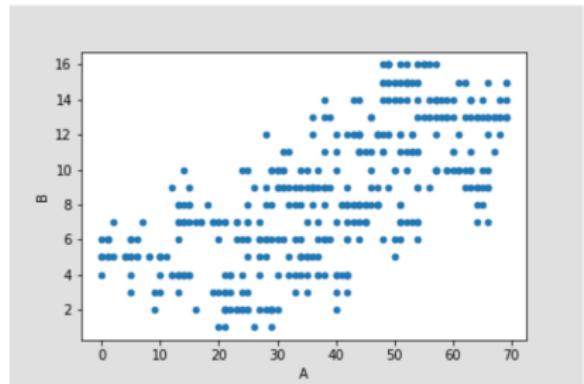


Quoi ?

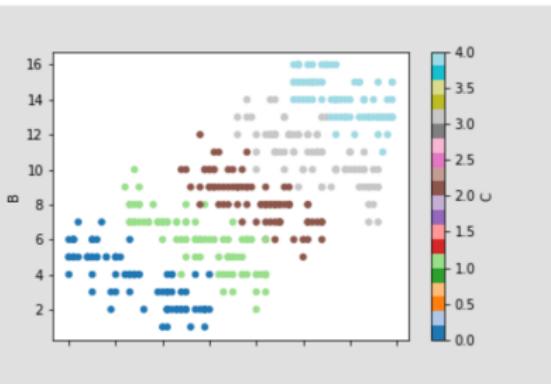


Comment ?

Paradoxe de Simpson - autre version



the trend is increasing



the trend is decreasing for any value for C !

Conclusions sur Simpson



Quoi ?

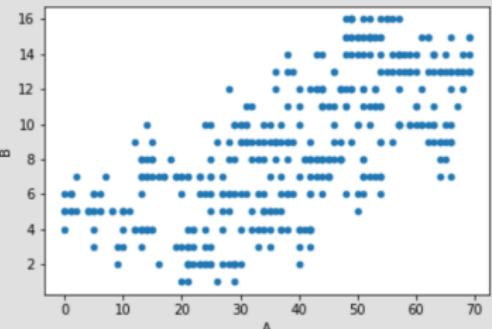


Comment ?

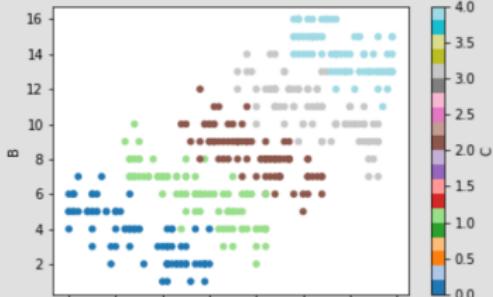
Et donc ?



Paradoxe de Simpson - autre version



the trend is increasing



the trend is decreasing for any value for C !

Conclusions sur Simpson



Quoi ?



Comment ?



Et donc ?



Introduction

Réseaux bayésiens (discrets)

Definition

Inference

Applications

Use Cases

Learning Bayesian Networks

aGrUM/pyAgrum



(Simple) Probabilistic Model





Joint Probabilistic Model

With A, C discrete random variables,





Joint Probabilistic Model

With A, C discrete random variables,

- ▶ Joint Probability :





Joint Probabilistic Model

With A, C discrete random variables,

- ▶ Joint Probability : $P(A, C)$





Joint Probabilistic Model

With A, C discrete random variables,

- ▶ Joint Probability : $P(A, C)$
- ▶ Marginal Probability :





Joint Probabilistic Model

With A, C discrete random variables,

- ▶ Joint Probability : $P(A, C)$
- ▶ Marginal Probability : $P(A) = \sum_C P(A, C)$





Joint Probabilistic Model

With A, C discrete random variables,

- ▶ Joint Probability : $P(A, C)$
- ▶ Marginal Probability : $P(A) = \sum_C P(A, C)$
- ▶ Conditional Probability :





Joint Probabilistic Model

With A, C discrete random variables,

- ▶ Joint Probability : $P(A, C)$
- ▶ Marginal Probability : $P(A) = \sum_C P(A, C)$
- ▶ Conditional Probability : $P(C|A) = \frac{P(A, C)}{P(A)}$





Joint Probabilistic Model

With A, C discrete random variables,

- ▶ Joint Probability : $P(A, C)$
- ▶ Marginal Probability : $P(A) = \sum_C P(A, C)$
- ▶ Conditional Probability : $P(C|A) = \frac{P(A, C)}{P(A)}$

Inference



Joint Probabilistic Model

With A, C discrete random variables,

- ▶ Joint Probability : $P(A, C)$
- ▶ Marginal Probability : $P(A) = \sum_C P(A, C)$
- ▶ Conditional Probability : $P(C|A) = \frac{P(A, C)}{P(A)}$

Inference

Inference consists in computing the distribution of one variable (C) given observations on some of the others (A).



Joint Probabilistic Model

With A, C discrete random variables,

- ▶ Joint Probability : $P(A, C)$
- ▶ Marginal Probability : $P(A) = \sum_C P(A, C)$
- ▶ Conditional Probability : $P(C|A) = \frac{P(A, C)}{P(A)}$

Inference

Inference consists in computing the distribution of one variable (C) given observations on some of the others (A).

$$P(C|\textcolor{red}{A} = \epsilon_a) = P(C|\epsilon_a)$$



Joint Probabilistic Model

With A, C discrete random variables,

- ▶ Joint Probability : $P(A, C)$
- ▶ Marginal Probability : $P(A) = \sum_C P(A, C)$
- ▶ Conditional Probability : $P(C|A) = \frac{P(A, C)}{P(A)}$

Inference

Inference consists in computing the distribution of one variable (C) given observations on some of the others (A).

$$P(C|\mathbf{A} = \epsilon_a) = P(C|\epsilon_a) = \frac{P(\epsilon_a|C) \cdot P(C)}{P(\epsilon_a)} = \frac{P(C, \epsilon_a)}{P(\epsilon_a)}$$



Joint Probabilistic Model

With A, C discrete random variables,

- ▶ Joint Probability : $P(A, C)$
- ▶ Marginal Probability : $P(A) = \sum_C P(A, C)$
- ▶ Conditional Probability : $P(C|A) = \frac{P(A, C)}{P(A)}$

Inference

Inference consists in computing the distribution of one variable (C) given observations on some of the others (A).

$$P(C|A = \epsilon_a) = P(C|\epsilon_a) = \frac{P(\epsilon_a|C) \cdot P(C)}{P(\epsilon_a)} = \frac{P(C, \epsilon_a)}{P(\epsilon_a)} \propto P(C, \epsilon_a)$$



Joint Probabilistic Model

With A, C discrete random variables,

- ▶ Joint Probability : $P(A, C)$
- ▶ Marginal Probability : $P(A) = \sum_C P(A, C)$
- ▶ Conditional Probability : $P(C|A) = \frac{P(A, C)}{P(A)}$

Inference

Inference consists in computing the distribution of one variable (C) given observations on some of the others (A).

$$P(C|A = \epsilon_a) = P(C|\epsilon_a) = \frac{P(\epsilon_a|C) \cdot P(C)}{P(\epsilon_a)} = \frac{P(C, \epsilon_a)}{P(\epsilon_a)} \propto P(C, \epsilon_a)$$

(Simple) Probabilistic Model (2)



Inference

Inference consists in computing the distribution of one variable (C) given observations on some of the others (A).

$$P(C|A = \epsilon_a) = P(C|\epsilon_a) = \frac{P(\epsilon_a|C) \cdot P(C)}{P(\epsilon_a)} = \frac{P(C, \epsilon_a)}{P(\epsilon_a)} \propto P(C, \epsilon_a)$$

		C	
		0	1
A	0	0.4235	0.0793
	1	0.3929	0.1043

C	
0	1
0.3929	0.1043
0.3929	0.1043

C	
0	1
0.7902	0.2098
0.7902	0.2098

$P(A, C)$

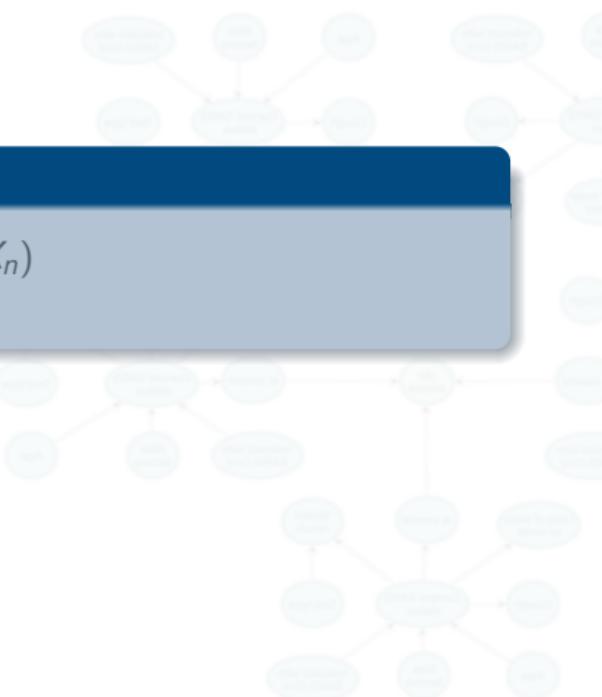
$P(A = 1, C)$

$P(C|A = 1)$



Probabilistic complex systems

$$P(X_1, \dots, X_n)$$





Probabilistic complex systems

⚠ $P(X_1, \dots, X_n)$

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



Probabilistic complex systems

⚠ $P(X_1, \dots, X_n)$

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

Complex Probabilistic Model



Probabilistic complex systems

⚠ $P(X_1, \dots, X_n)$

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

$$P(X_i|\epsilon_j) = \frac{P(X_i, \epsilon_j)}{P(\epsilon_j)} \propto \sum_{k \notin \{i,j\}} P(\dots, X_i, \dots, \epsilon_j, \dots)$$

		D		
		B	C	
		A	0	1
0	0	0	0.3383	0.0566
1	0	1	0.0227	0.0059
0	1	0	0.0511	0.0193
1	1	0	0.0069	0.0002
0	0	0	0.1104	0.0185
1	0	1	0.2098	0.0543
0	1	0	0.0167	0.0063
1	1	0	0.0793	0.0020

C		
A	0	1
0	0.4235	0.0793
1	0.3929	0.1043

C		
	0	1
0	0.3929	0.1043
1	0.7902	0.2098

$$P(A, B, C, D)$$

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

$$P(A = 1, C)$$

$$P(C | A = 1)$$



Probabilistic complex systems

⚠ $P(X_1, \dots, X_n)$

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

$$P(X_i | \epsilon_j) = \frac{P(X_i, \epsilon_j)}{P(\epsilon_j)} \propto \text{⚠} \sum_{k \notin \{i, j\}} P(\dots, X_i, \dots, \epsilon_j, \dots)$$

⚠ Combinatorial explosion, curse of dimensionality !



Probabilistic complex systems

⚠ $P(X_1, \dots, X_n) \Rightarrow O(k^n)$ in space

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

$P(X_i | \epsilon_j) = \frac{P(X_i, \epsilon_j)}{P(\epsilon_j)} \propto \sum_{k \notin \{i, j\}} P(\dots, X_i, \dots, \epsilon_j, \dots) \Rightarrow O(k^n)$ in time

⚠ Combinatorial explosion, curse of dimensionality !



Bayesian Network – Model





▶ Definition (Bayesian Network (BN))





► 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)



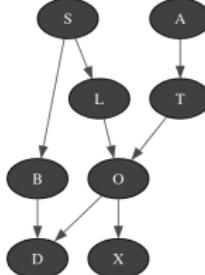


► Definition (Bayesian Network (BN))

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)

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





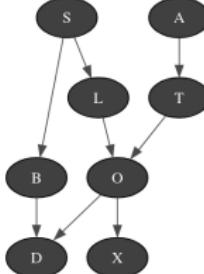
► Definition (Bayesian Network (BN))

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)

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

($2^8 = 256$ parameters)





▶ Definition (Bayesian Network (BN))

A Bayesian network is a joint distribution over a set of random (discrete) variables.

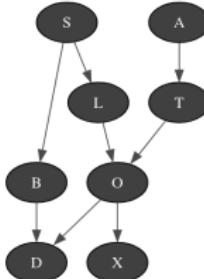
A Bayesian network is represented by a directed acyclic graph (DAG) and by a conditional probability table (CPT) for each node X_i : $P(X_i|\text{parents}_i)$.

Factorization of the joint distribution in a BN

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

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

($2^8 = 256$ parameters)





▶ Definition (Bayesian Network (BN))

A Bayesian network is a joint distribution over a set of random (discrete) variables.

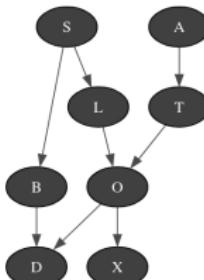
A Bayesian network is represented by a directed acyclic graph (DAG)

Factorization of the joint distribution in a BN

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

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

$$(2^8 = 256 \text{ parameters})$$



$$\begin{aligned} &P(A) \cdot P(S) \cdot P(T|A) \cdot P(L|S) \cdot \\ &P(O|T, L) \cdot P(B|S) \cdot P(X|O) \cdot \\ &P(D|O, B) \end{aligned}$$



▶ Definition (Bayesian Network (BN))

A Bayesian network is a joint distribution over a set of random (discrete) variables.

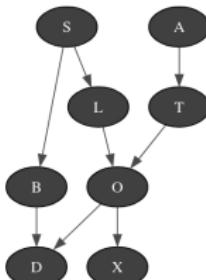
A Bayesian network is represented by a directed acyclic graph (DAG)

Factorization of the joint distribution in a BN

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

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

$$(2^8 = 256 \text{ parameters})$$



$$\begin{aligned} & P(A) \cdot P(S) \cdot P(T|A) \cdot P(L|S) \cdot \\ & P(O|T, L) \cdot P(B|S) \cdot P(X|O) \cdot \\ & P(D|O, B) \\ & (2+2+4+4+8+4+4+8 = 32 \text{ parameters}) \end{aligned}$$



▶ 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
 $P(X_i|\text{parents}_i)$

Inference : $P(\mathbf{D})$?

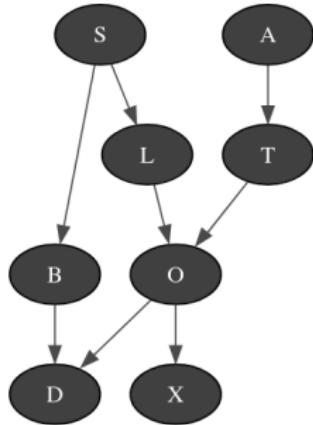




▶ Definition (Bayesian Network (BN))

A Bayesian network is represented by a directed acyclic graph (DAG) and by a conditional probability table (CPT) for each node
 $P(X_i|\text{parents}_i)$

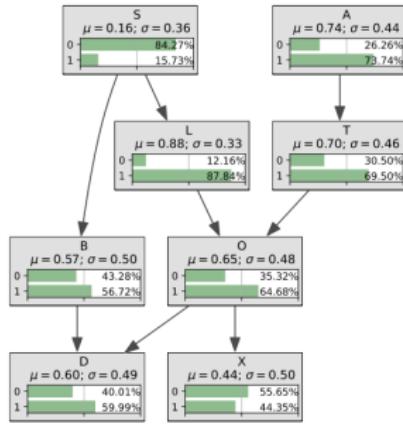
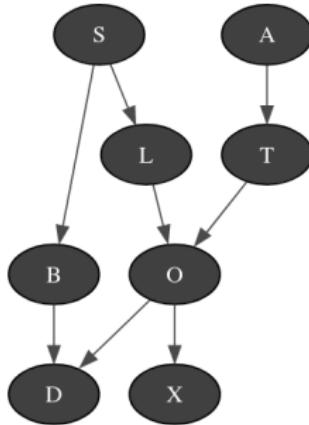
Inference : $P(\mathbf{D})$?



▶ 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
 $P(X_i|\text{parents}_i)$

Inference : $P(\mathbf{D})$?





▶ 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
 $P(X_i|\text{parents}_i)$

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

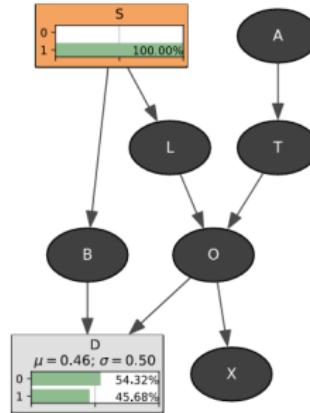
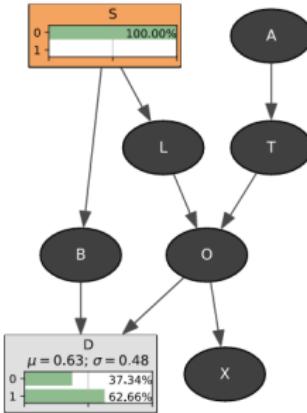




▶ Definition (Bayesian Network (BN))

A Bayesian network is represented by a directed acyclic graph (DAG) and by a conditional probability table (CPT) for each node
 $P(X_i|\text{parents}_i)$

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



Un exemple de modélisation





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





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 autre chose).





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 autre chose).
- Un séjour récent en **Asie** augmente les chances de tuberculose, tandis que **fumer** augmente les risques de cancer des poumons.





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 autre chose).
- 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.





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 autre chose).
- 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.





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 autre chose).
- 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.

Variables and relations :





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 autre chose).
- 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.

Variables and relations :

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



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 autre chose).
- 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.

Variables and relations :

- 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



Probabilités :





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 autre chose).
- 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.

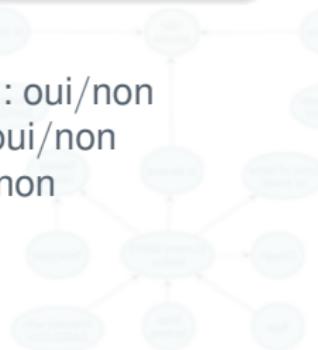
Variables and relations :

- 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



Probabilités :

- $P(T|A = \text{oui}) = ?$





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 autre chose).
- 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.

Variables and relations :

- 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



Probabilités :

- $P(T|A = \text{oui}) = ?$
- $P(T|A = \text{non}) = ?$





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 autre chose).
- 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.

Variables and relations :

- 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



Probabilités :

- $P(T|A = \text{oui}) = ?$
- $P(T|A = \text{non}) = ?$
- $P(C|F) = ?$

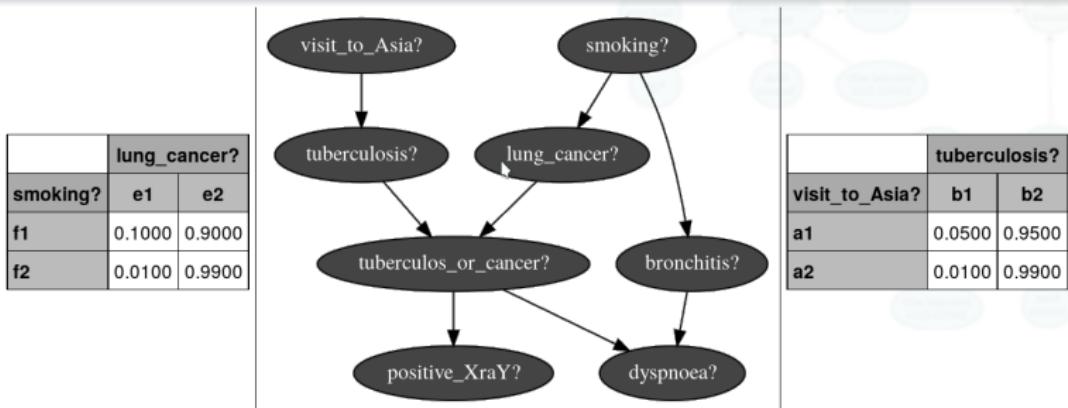


Un exemple



example 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 autre chose).
- 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.

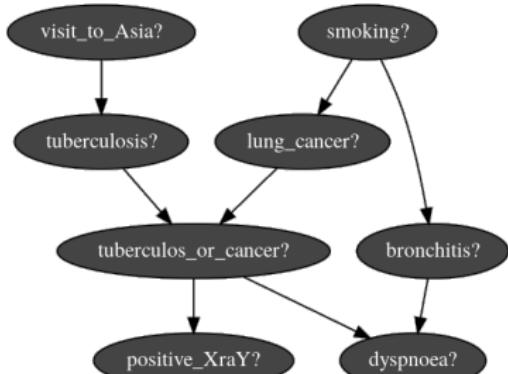


Un exemple



example 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 autre chose).
- 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.



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



▶ Definition (Réseau bayésien (BN))

Un réseau bayésien est représenté par un graphe orienté sans circuit (DAG) and par des distribution de probabilités conditionnelles : $P(X_i|\text{parents}_i)$

Inférences probabilistes : $P(\text{dyspnée}|\text{smoking})$?

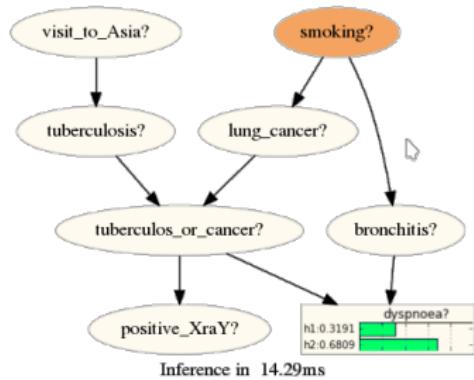




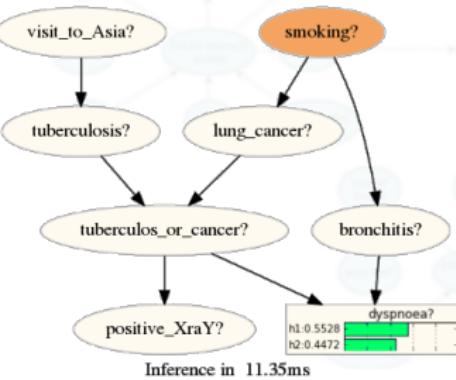
▶ Definition (Réseau bayésien (BN))

Un réseau bayésien est représenté par un graphe orienté sans circuit (DAG) and par des distribution de probabilités conditionnelles : $P(X_i|\text{parents}_i)$

Inférences probabilistes : $P(\text{dyspnée}|\text{smoking})$?

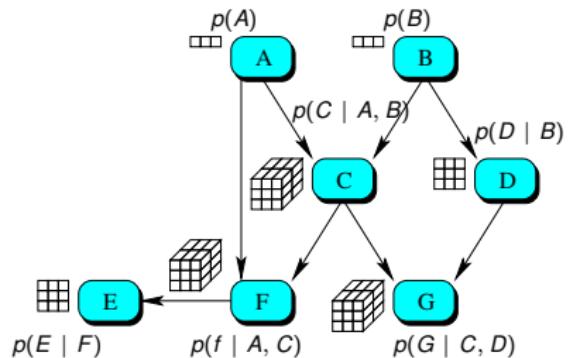


$$P(\text{dyspnée}|\text{smoking} = 1)$$



$$P(\text{dyspnée}|\text{smoking} = 0)$$

BN and probabilistic inference

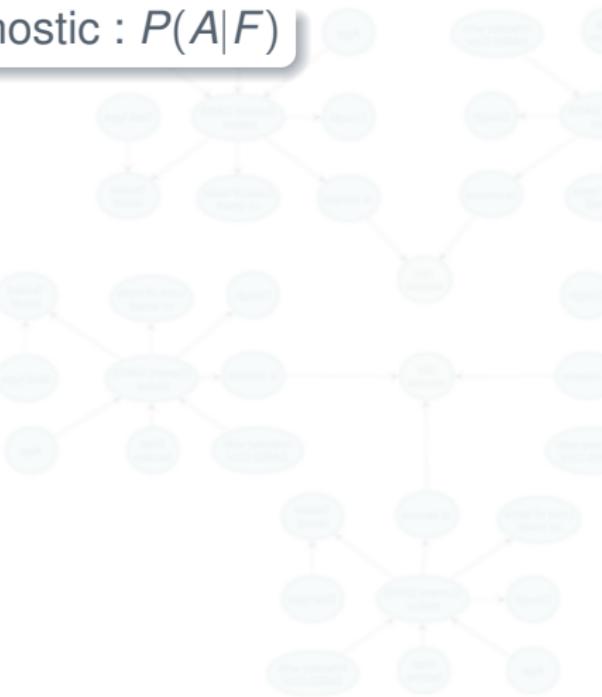
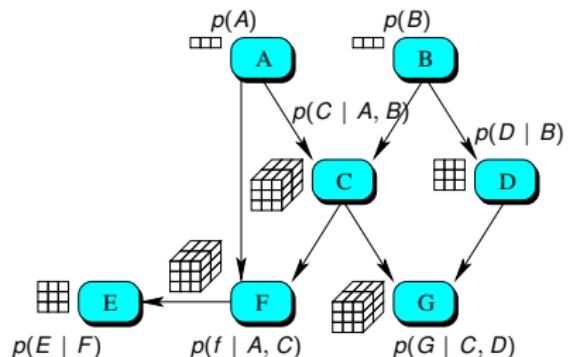


BN and probabilistic inference



17

diagnostic : $P(A|F)$

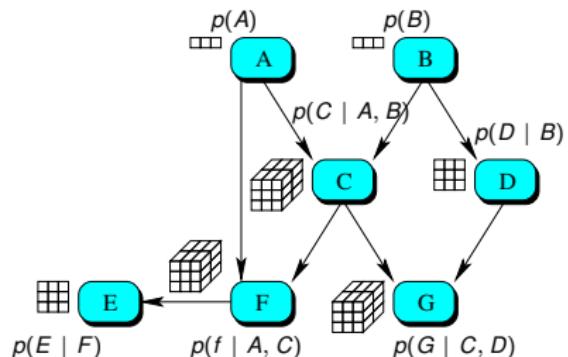


BN and probabilistic inference

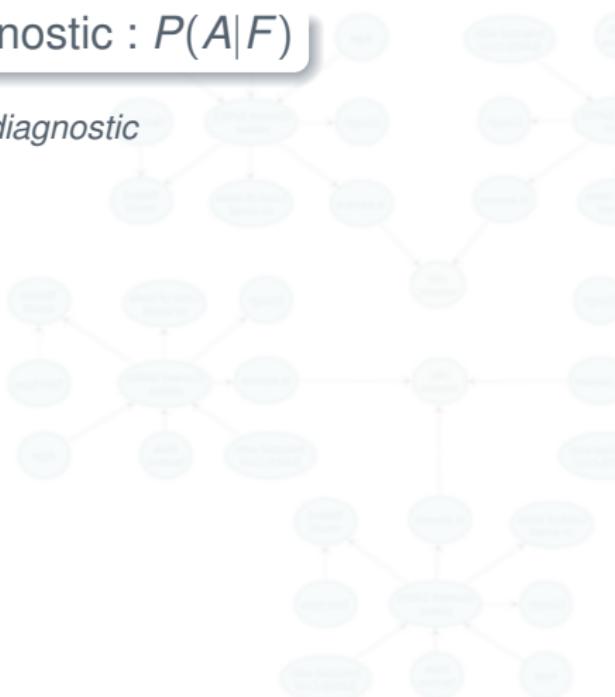


17

diagnostic : $P(A|F)$



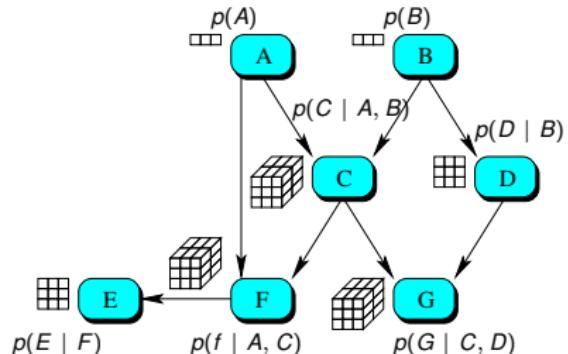
● diagnostic



BN and probabilistic inference



diagnostic : $P(A|F)$



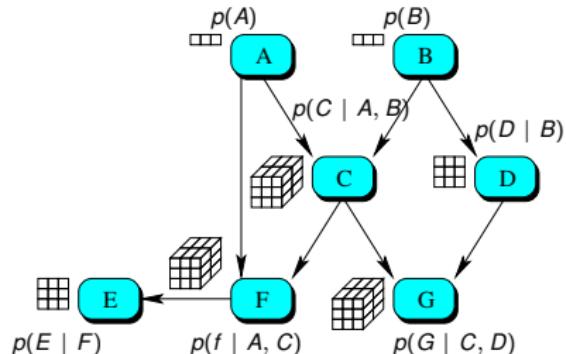
- diagnostic
- reliability



BN and probabilistic inference



diagnostic : $P(A|F)$



- diagnostic
- reliability
- classification



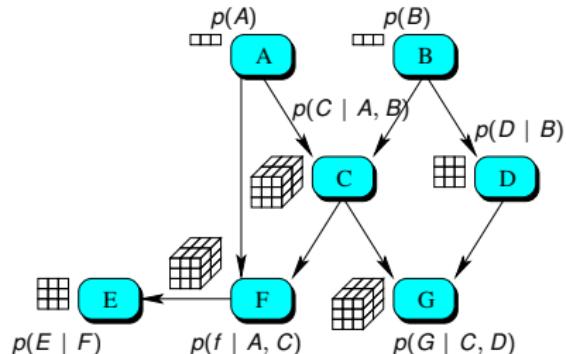
BN and probabilistic inference



diagnostic : $P(A|F)$

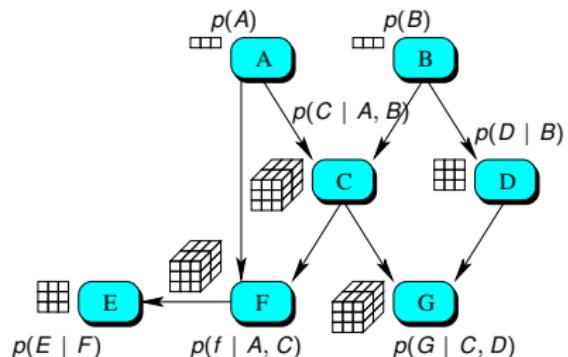
- diagnostic
- reliability
- classification

prediction $P(E|B, A)$





diagnostic : $P(A|F)$



- diagnostic

- reliability

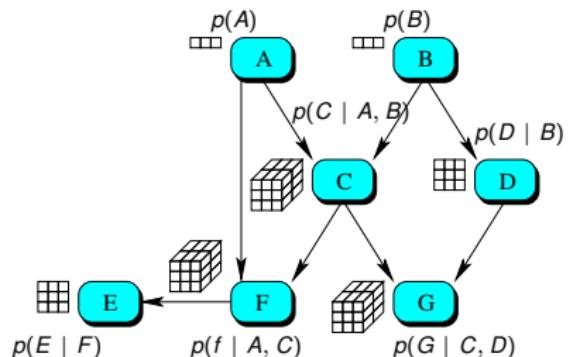
- classification

prediction $P(E|B, A)$

- Process simulation (modélisation)



diagnostic : $P(A|F)$



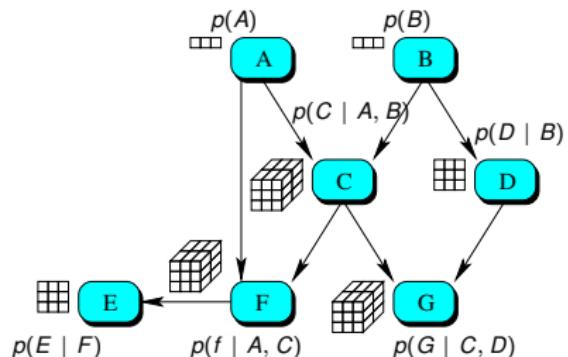
- diagnostic
- reliability
- classification

prediction $P(E|B, A)$

- Process simulation (modélisation)
- forecasting (dynamics, etc.)



diagnostic : $P(A|F)$



- diagnostic

- reliability

- classification

prediction $P(E|B, A)$

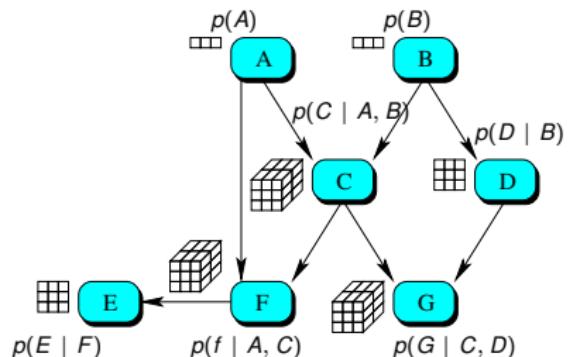
- Process simulation (modélisation)

- forecasting (dynamics, etc.)

- Behavioral analysis (bot, intelligent tutoring system)



diagnostic : $P(A|F)$



- diagnostic

- reliability

- classification

prediction $P(E|B, A)$

- Process simulation (modélisation)

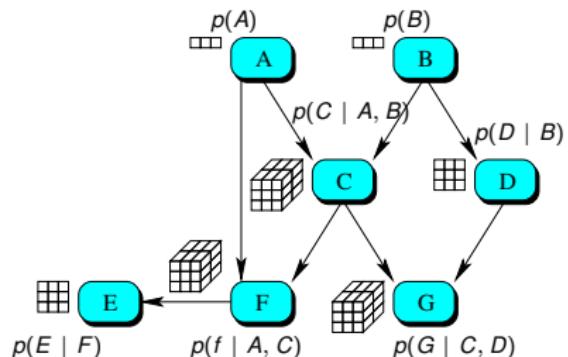
- forecasting (dynamics, etc.)

- Behavioral analysis (bot, intelligent tutoring system)

Others tasks



diagnostic : $P(A|F)$



- diagnostic

- reliability

- classification

prediction $P(E|B, A)$

- Process simulation (modélisation)

- forecasting (dynamics, etc.)

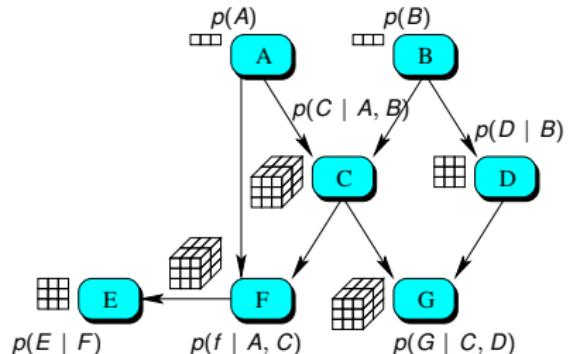
- Behavioral analysis (bot, intelligent tutoring system)

Others tasks

- Most Probable Case : $\arg \max P(\mathfrak{X}|D)$



diagnostic : $P(A|F)$



- diagnostic

- reliability

- classification

prediction $P(E|B, A)$

- Process simulation (modélisation)

- forecasting (dynamics, etc.)

- Behavioral analysis (bot, intelligent tutoring system)

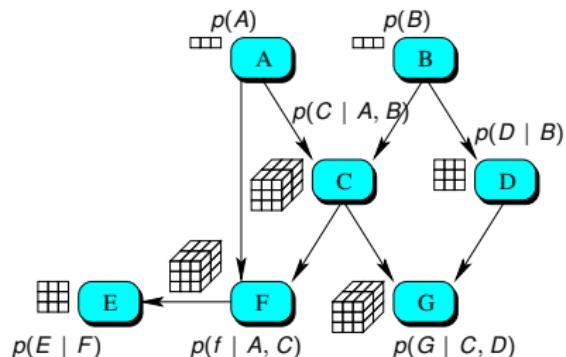
Others tasks

- Most Probable Case : $\arg \max P(\mathfrak{X}|D)$

- Sensitivity analysis, Informational analysis (mutual information), etc.



diagnostic : $P(A|F)$



- diagnostic

- reliability

- classification

prediction $P(E|B, A)$

- Process simulation (modélisation)

- forecasting (dynamics, etc.)

- Behavioral analysis (bot, intelligent tutoring system)

Others tasks

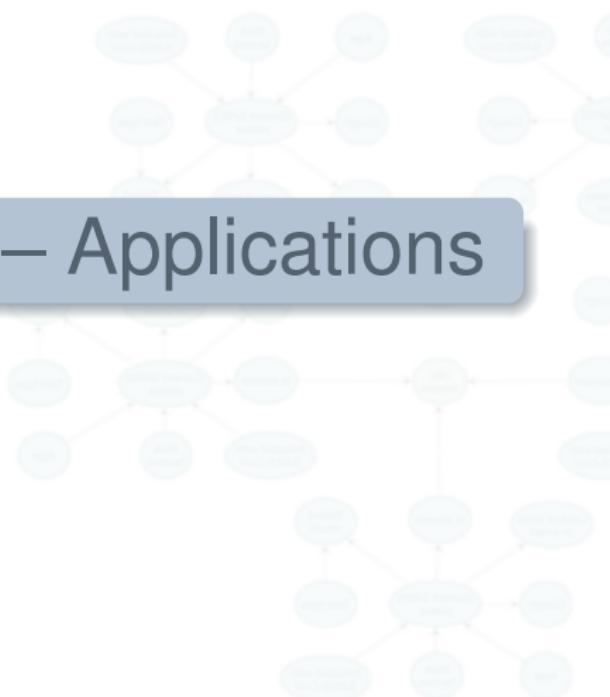
- Most Probable Case : $\arg \max P(\mathfrak{X}|D)$

- Sensitivity analysis, Informational analysis (mutual information), etc.

- Decision process, Troubleshooting : $\arg \max \frac{P(\cdot)}{C(\cdot)}$



Bayesian Network – Applications



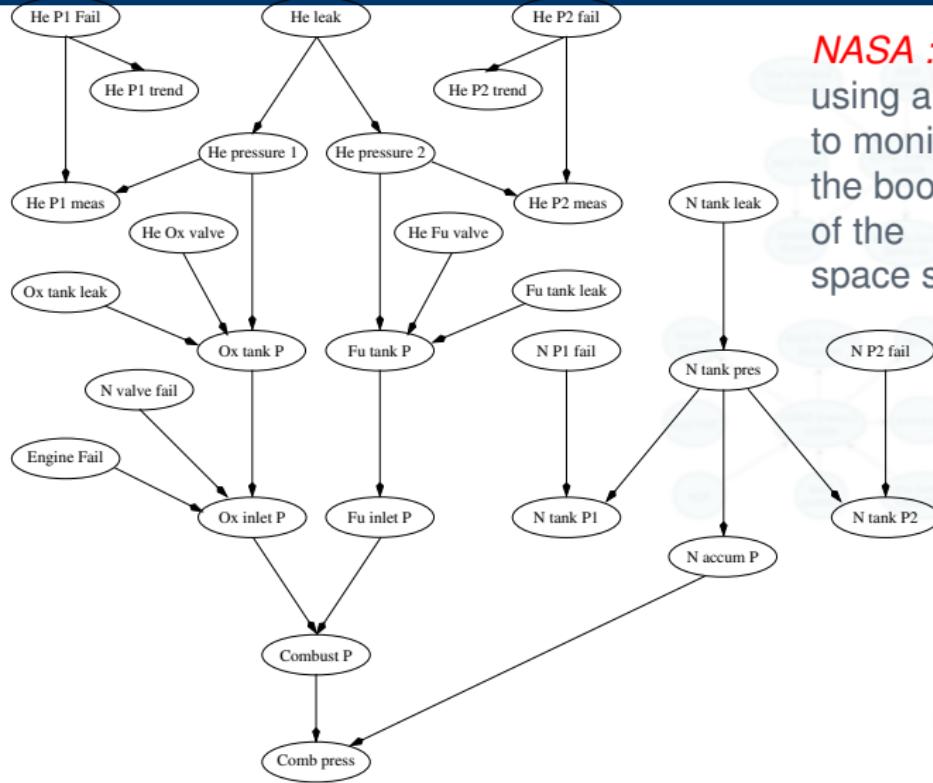
Application 1 : diagnostic

Diagnostic @ NASA



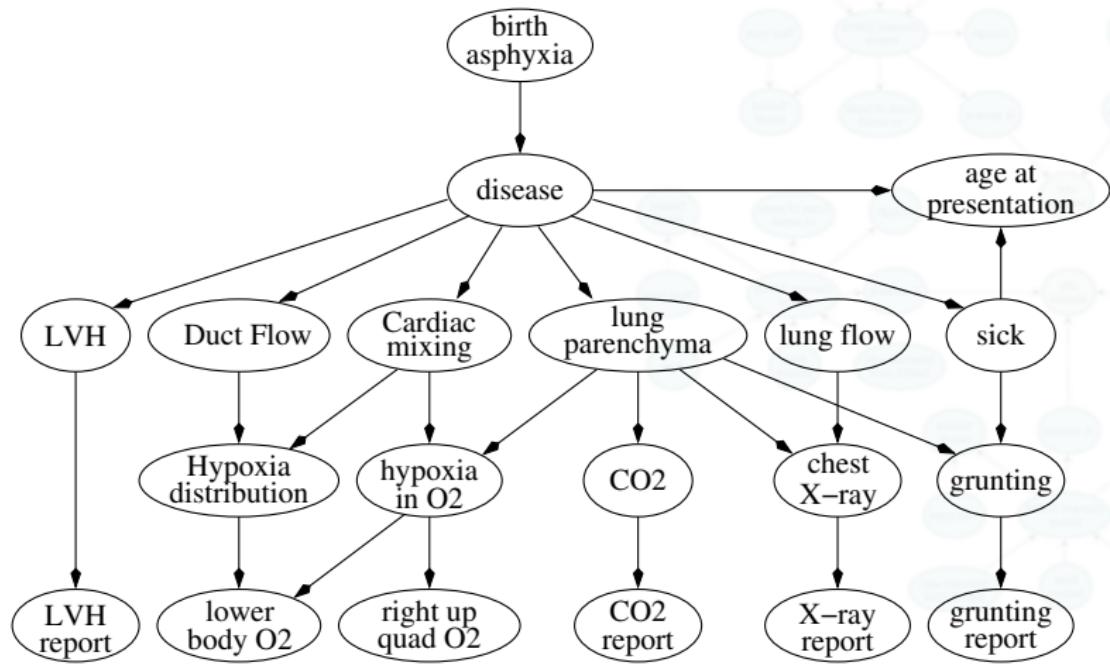
19

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





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



Application 3 : risk analysis



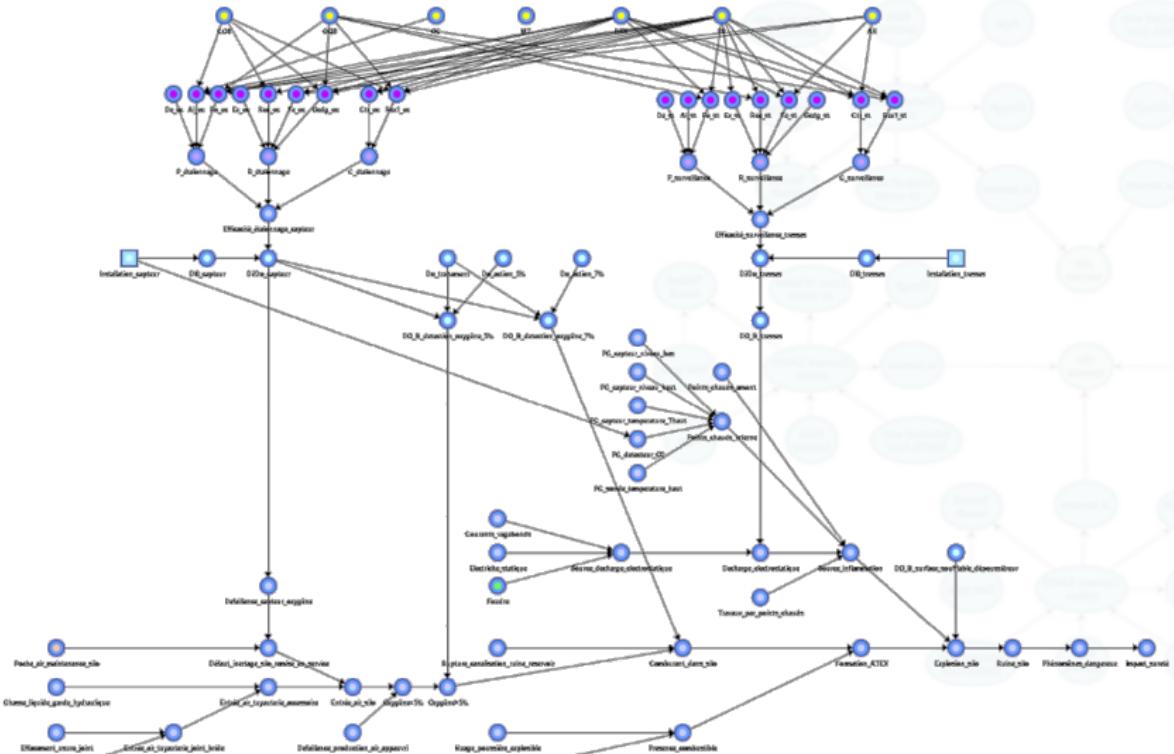
Risk modeling using BN : **modular approach.**



Application 3 : risk analysis



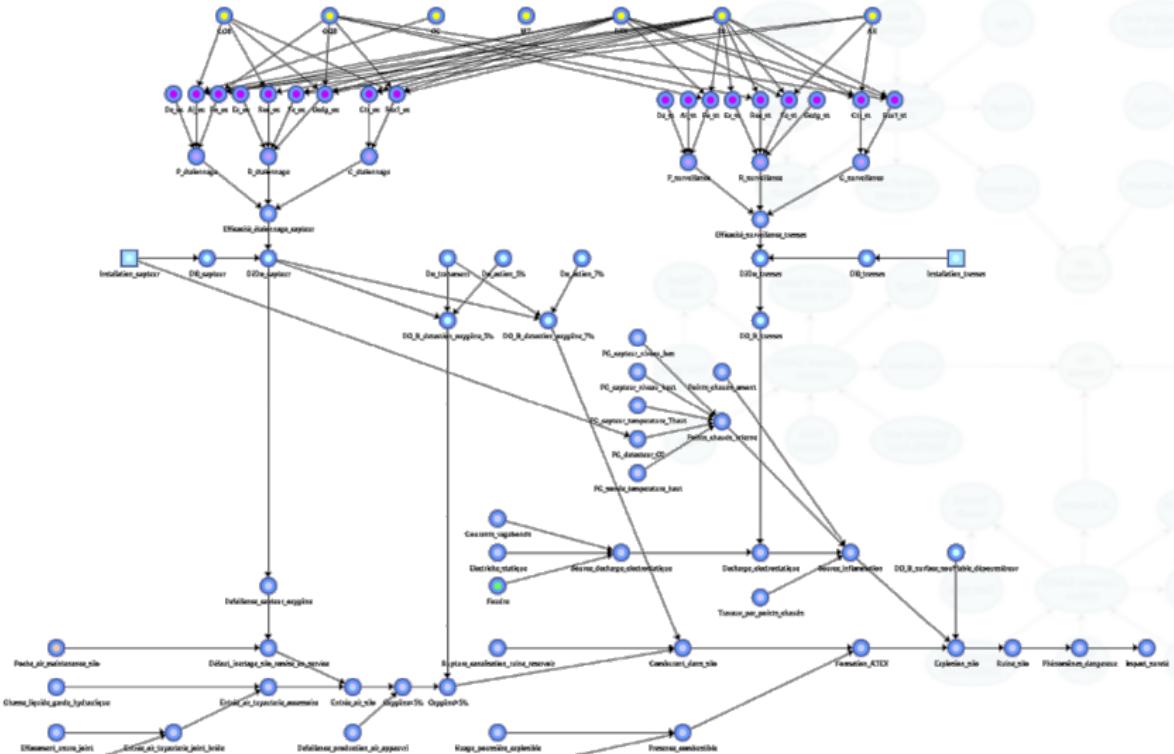
Risk modeling using BN : modular approach.



Application 3 : risk analysis



Risk modeling using BN : modular approach.



Classification probabiliste : modèles complexes

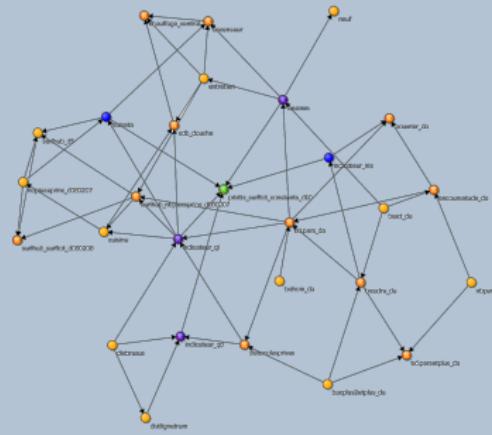




Réseau bayésien pour la classification

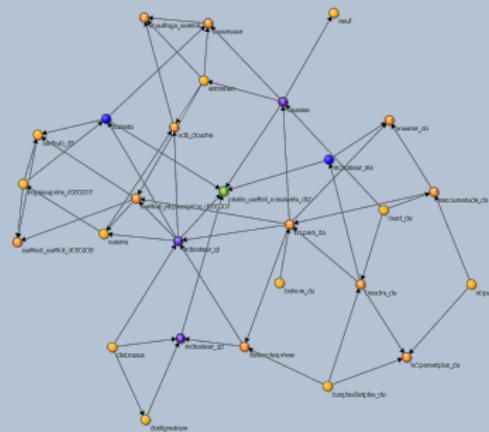


Réseau bayésien pour la classification





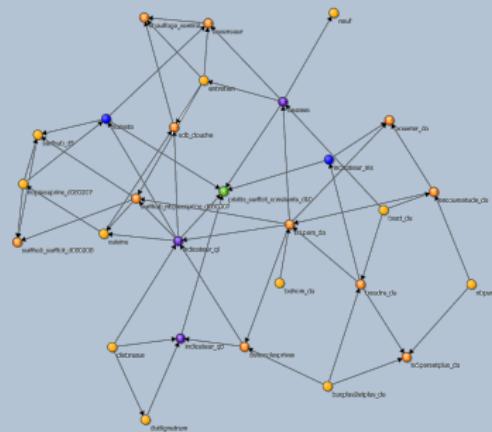
Réseau bayésien pour la classification



Dans un BN composé de Y et (X_i) , calculer $P(Y | X_1, \dots, X_n)$.



Réseau bayésien pour la classification

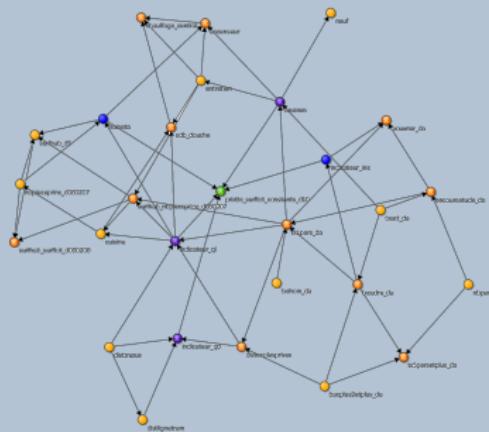


Dans un BN composé de Y et (X_i) , calculer $P(Y | X_1, \dots, X_n)$.

Note : on n'a pas besoin de tous les X_i :



Réseau bayésien pour la classification

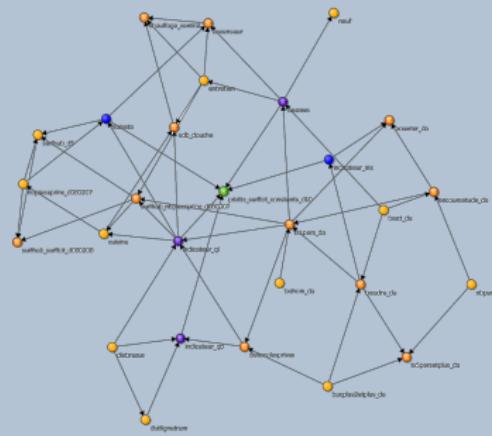


Dans un BN composé de Y et (X_i) ,
calculer $P(Y | X_1, \dots, X_n)$.

Note : on n'a pas besoin de tous les X_i : **Markov Boundary MB**



Réseau bayésien pour la classification



Dans un BN composé de Y et (X_i) , calculer $P(Y | X_1, \dots, X_n)$.

Note : on n'a pas besoin de tous les X_i : **Markov Boundary MB**

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

Modèle probabiliste dédié à la classification





Classifieur bayésien naïf





Classifieur bayésien naïf

$$\forall k \neq l, X^k \perp\!\!\!\perp X^l | Y$$

et





Classifieur bayésien naïf

$$\forall k \neq l, X^k \perp\!\!\!\perp X^l | Y$$

et

$$P(x, y) = P(y) \cdot \prod_{k=1}^d P(x^k | y)$$



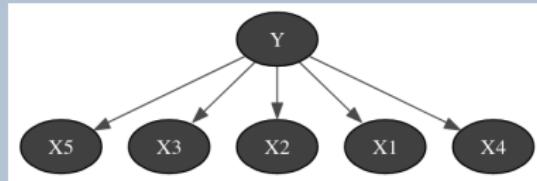


Classifieur bayésien naïf

$$\forall k \neq l, X^k \perp\!\!\!\perp X^l | Y$$

et

$$P(x, y) = P(y) \cdot \prod_{k=1}^d P(x^k | y)$$



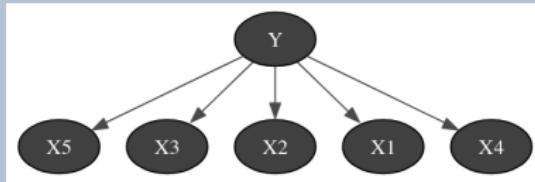


Classifieur bayésien naïf

$$\forall k \neq l, X^k \perp\!\!\!\perp X^l | Y$$

et

$$P(x, y) = P(y) \cdot \prod_{k=1}^d P(x^k | y)$$



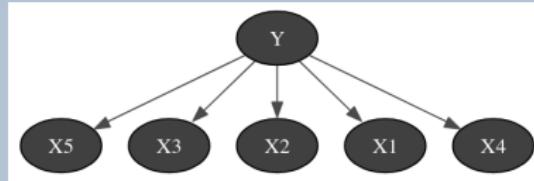


Classifieur bayésien naïf

$$\forall k \neq l, X^k \perp\!\!\!\perp X^l | Y$$

et

$$P(x, y) = P(y) \cdot \prod_{k=1}^d P(x^k | y)$$



Tree-Augmented Naive Models

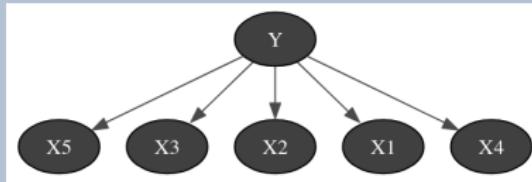


Classifieur bayésien naïf

$$\forall k \neq l, X^k \perp\!\!\!\perp X^l | Y$$

et

$$P(x, y) = P(y) \cdot \prod_{k=1}^d P(x^k | y)$$



Tree-Augmented Naive Models

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

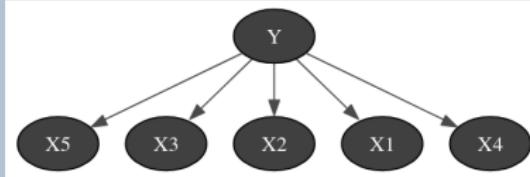


Classifieur bayésien naïf

$$\forall k \neq l, X^k \perp\!\!\!\perp X^l | Y$$

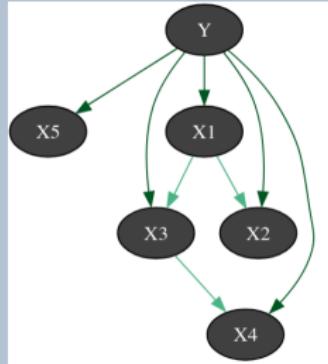
et

$$P(x, y) = P(y) \cdot \prod_{k=1}^d P(x^k | y)$$



Tree-Augmented Naive Models

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



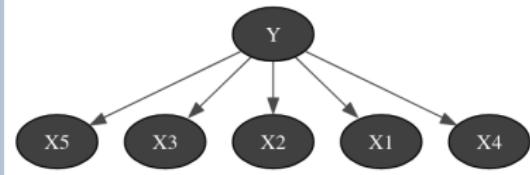


Classifieur bayésien naïf

$$\forall k \neq l, X^k \perp\!\!\!\perp X^l | Y$$

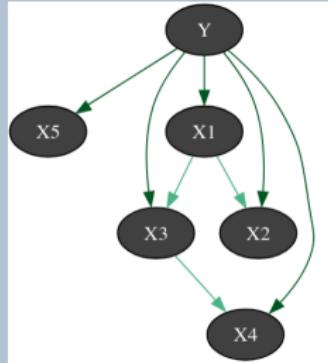
et

$$P(x, y) = P(y) \cdot \prod_{k=1}^d P(x^k | y)$$



Tree-Augmented Naive Models

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



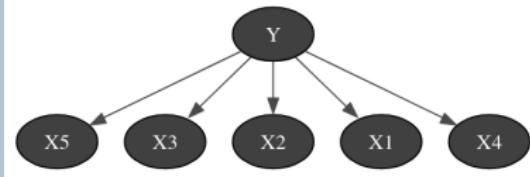


Classifieur bayésien naïf

$$\forall k \neq l, X^k \perp\!\!\!\perp X^l | Y$$

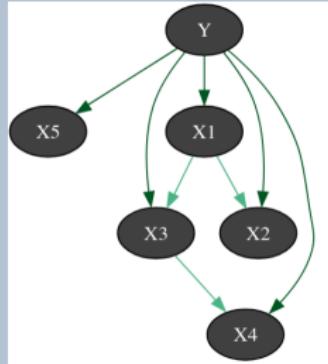
et

$$P(x, y) = P(y) \cdot \prod_{k=1}^d P(x^k | y)$$



Tree-Augmented Naive Models

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





Introduction

Réseaux bayésiens (discrets)

Definition

Inference

Applications

Use Cases

Learning Bayesian Networks

aGrUM/pyAgrum





Apprentissage dans les réseaux bayésiens

L'apprentissage a pour but d'**estimer**,





Apprentissage dans les réseaux bayésiens

L'apprentissage a pour but d'**estimer**, à partir d'une **base de données**





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 ?)





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 ($P(X = 0 \mid Y = 1)$?)





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 :





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**.





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 ;





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**,





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.





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

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.



Apprentissage des paramètres, données complètes

Résumé



25

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 :

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)

Apprentissage des paramètres, données complètes

Résumé



25

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)

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

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)

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

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

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)

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

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

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

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)

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

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

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

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

Apprentissage des paramètres, données complètes

Résumé



25

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)

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

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

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

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

- *A priori* important quand $N_{ijk} \rightarrow 0$: pas de cas dans la base.

Apprentissage des paramètres, données complètes

Résumé



25

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)

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

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

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

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

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



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)



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**



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 $\alpha_{ij} = \sum_k \alpha_{ijk}$



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 N_{ij} : elle détermine l'influence a l'*a priori* sur la loi.



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 N_{ij} : elle détermine l'influence a l'*a priori* sur la loi.

- ajustement de Laplace (smoothing)



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 N_{ij} : elle détermine l'influence a l'*a priori* sur la loi.

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



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 N_{ij} : elle détermine l'influence a l'*a priori* sur la loi.

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

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



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** $\hat{\theta}_{ijk} \approx \frac{N_{ijk} + \alpha_{ijk}}{N_{ij} + \alpha_{ij}}$ avec $\alpha_{ij} = \sum_k \alpha_{ijk}$

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

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

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

- **actualisation de Ney-Essen**



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 N_{ij} : elle détermine l'influence a l'*a priori* sur la loi.

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

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.



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 N_{ij} : elle détermine l'influence a l'a priori sur la loi.

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

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.

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

Apprentissage avec données manquantes : EM pour les BNs





EM dans les BNs





EM dans les BNs

Répéter jusqu'à convergence





EM dans les BNs

Répéter jusqu'à convergence

Étape E : Estimer $N_{ijk}^{(t+1)}$ à partir des $P(X_i | Pa_i, \theta_{ijk}^t)$



EM dans les BNs

Répéter jusqu'à 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

EM dans les BNs

Répéter jusqu'à 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 : $\theta_{ijk}^{t+1} = \frac{N_{ijk}^{(t+1)}}{N_{ij}^{(t+1)}}$



EM dans les BNs

Répéter jusqu'à 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 : $\theta_{ijk}^{t+1} = \frac{N_{ijk}^{(t+1)}}{N_{ij}^{(t+1)}}$



Apprentissage de la structure, données complètes



- ▶ But :



Apprentissage de la structure, données complètes



28

- ▶ **But :** obtenir automatiquement une structure de réseau bayésien à partir de données.



Apprentissage de la structure, données complètes



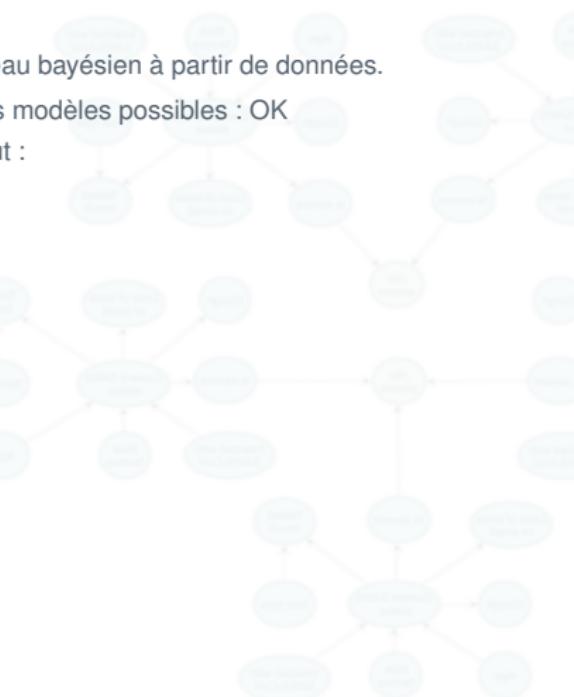
- ▶ **But :** obtenir automatiquement une structure de réseau bayésien à partir de données.
- ▶ **En théorie :** Test du χ^2 plus énumération de tous les modèles possibles : OK



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 :





- ▶ **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 acyclic digraphs*. In Lecture Notes in Mathematics : Combinatorial Mathematics V

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



Tableau général de l'apprentissage





Tableau général de l'apprentissage

Recherche de relation symétrique + orientation (*causalité*)





Tableau général de l'apprentissage

Recherche de relation symétrique + orientation (*causalité*)

- ▶ algorithme **IC/PC**
- ▶ algorithme **IC*/FCI**





Tableau général de l'apprentissage

Recherche de relation symétrique + orientation (*causalité*)

- ▶ algorithme **IC/PC**
- ▶ algorithme **IC*/FCI**

Recherche heuristique (score)





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





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**),





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





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, ...**).



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, ...**).

Classe d'équivalence de Markov



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, ...**).

Classe d'équivalence de Markov

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



Tableau général de l'apprentissage

Recherche de relation symétrique + orientation (*causalité*)

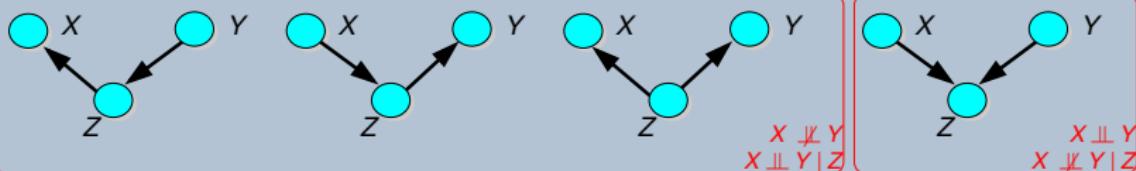
- ▶ algorithme IC/PC
- ▶ algorithme IC*/FCI

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, ...).

Classe d'équivalence de Markov

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

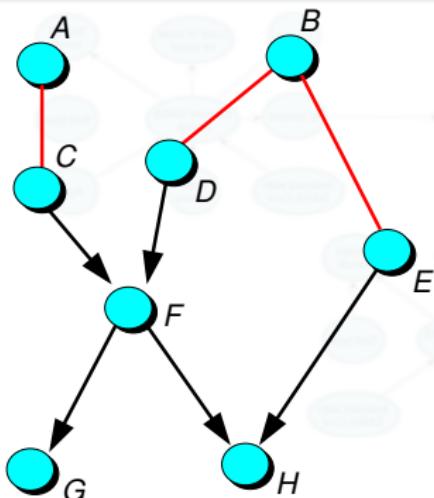
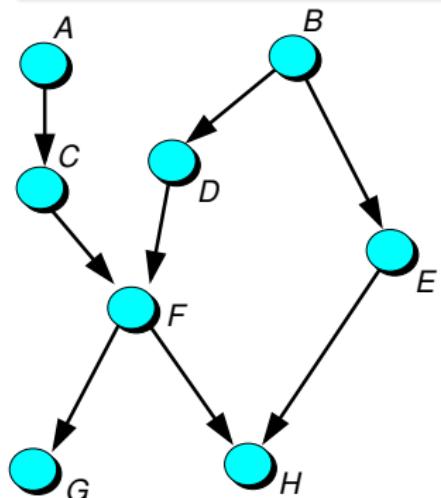




Classe d'équivalence de Markov, graphe essentiel

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



31

En terme statistique, les relations testables sont symétriques : **corrélation ou indépendance entre variables aléatoires.**



Recherche de relation symétrique



31

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.

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)



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) :





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.





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.





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. Déetecter les V-structures et les orientations qu'elles impliquent.





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. Déetecter les V-structures et les orientations qu'elles impliquent.
3. Finaliser les orientations en restant dans la même classe d'équivalence de Markov.





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. Déetecter les V-structures et les orientations qu'elles impliquent.
3. Finaliser les orientations en restant dans la même classe d'équivalence de Markov.





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. Déetecter 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 :





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)

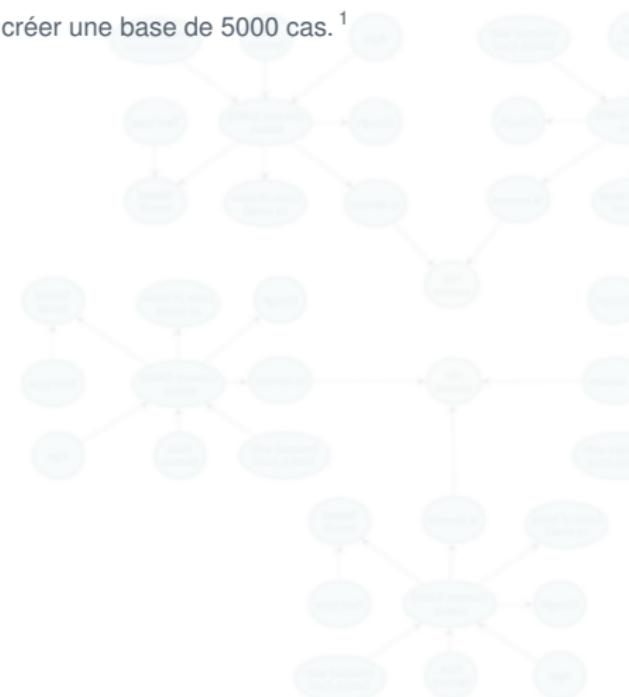
1. Construire le graphe (non orienté) des relations de dépendance trouvées statistiquement (χ^2 ou autre) :
 - ▶ Ajouter des arêtes à partir du graphe vide.
 - ▶ Retirer des arêtes à partir du graphe complet.
2. Déetecter les V-structures et les orientations qu'elles impliquent.
3. Finaliser les orientations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



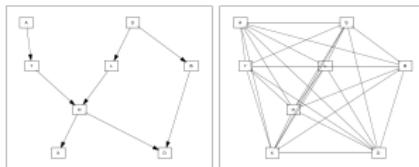
1. Exemple de Philippe Leray

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.



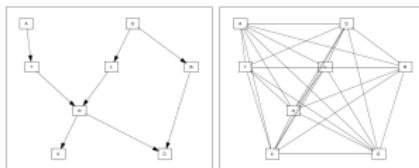
1. Exemple de Philippe Leray

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 ($X \perp\!\!\!\perp Y | Z$).

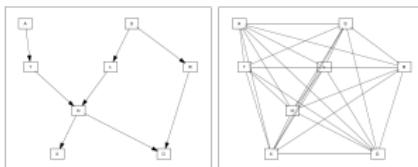
1. Exemple de Philippe Leray

Exemple PC



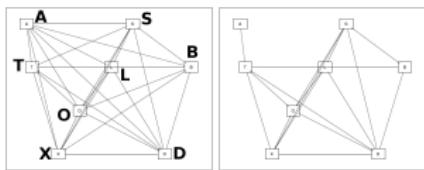
- Soit un réseau bayésien (à gauche) qui a permis de créer une base de 5000 cas.¹

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 ($X \perp\!\!\!\perp Y | Z$).

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



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

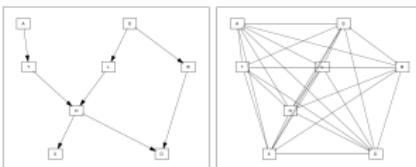
1. Exemple de Philippe Leray

Exemple PC



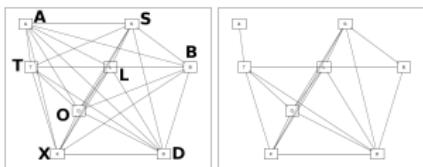
- Soit un réseau bayésien (à gauche) qui a permis de créer une base de 5000 cas.¹

Etape 0 : Graphe non orienté reliant tous les nœuds.



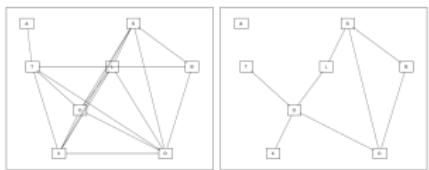
- Par des χ^2 , on teste toutes les indépendances marginales ($X \perp\!\!\!\perp Y$) puis les indépendances par rapport à une variable ($X \perp\!\!\!\perp Y | Z$).

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

1. Exemple de Philippe Leray

Exemple PC



- On continue les χ^2 d'ordre supérieur

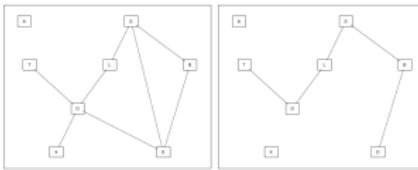


Exemple PC



- On continue les χ^2 d'ordre supérieur

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



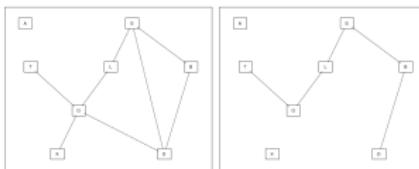
On trouve : $D \perp\!\!\!\perp S | (L, B)$, $X \perp\!\!\!\perp O | (T, L)$, $D \perp\!\!\!\perp O | (T, L)$.

Exemple PC



- On continue les χ^2 d'ordre supérieur

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



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

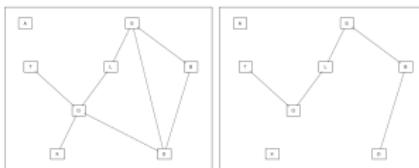
- Recherche des V-Structure, propagation des contraintes d'orientations puis orientations des dernières arêtes en restant Markov-équivalent.

Exemple PC



- On continue les χ^2 d'ordre supérieur

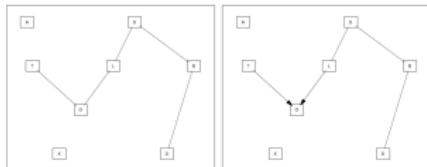
Etape 1c : Suppression des ind. conditionnelles d'ordre 2



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

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

Etape 2 : Recherche des V-structures



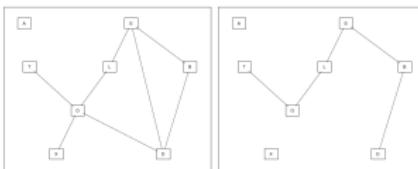
On trouve : $T \not\perp\!\!\!\perp L$ et $T \perp\!\!\!\perp L | O$

Exemple PC



- On continue les χ^2 d'ordre supérieur

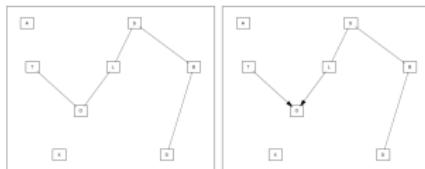
Etape 1c : Suppression des ind. conditionnelles d'ordre 2



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

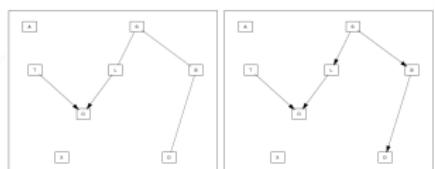
- Recherche des V-Structure, propagation des contraintes d'orientations puis orientations des dernières arêtes en restant Markov-équivalent.

Etape 2 : Recherche des V-structures



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

Etape 4 : Instantiation du PDAG



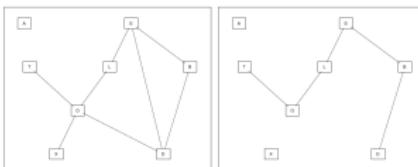
Orientation sans nouvelle V-structure

Exemple PC



- On continue les χ^2 d'ordre supérieur

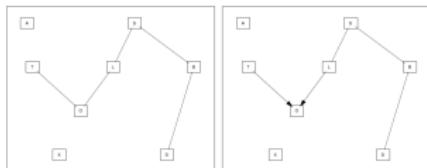
Etape 1c : Suppression des ind. conditionnelles d'ordre 2



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

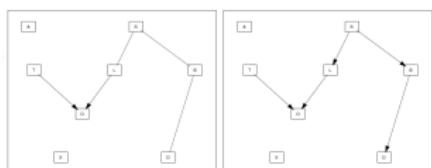
- Recherche des V-Structure, propagation des contraintes d'orientations puis orientations des dernières arêtes en restant Markov-équivalent.

Etape 2 : Recherche des V-structures



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

Etape 4 : Instantiation du PDAG



Orientation sans nouvelle V-structure

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

Recherche locale à base de scores



La recherche exhaustive des relations d'indépendances est inatteignable (nombre de tests prohibitifs, quantité de données nécessaires trop importantes, etc.).



Recherche locale à base de scores



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.



Recherche locale à base de scores



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



Recherche locale à base de scores



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

Recherche locale

Recherche locale à base de scores



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

Recherche locale

- Soit un espace de recherche,



Recherche locale à base de scores



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. 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





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. 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).





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. 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 (heuristique) calculable localement.





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. 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 (heuristique) 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*).





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. 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 (heuristique) 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 à base de scores



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. 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 (heuristique) 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)

Recherche locale à base de scores



34

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. 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 (heuristique) 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 : voir slides suivants

Recherche locale à base de scores



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. 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 (heuristique) 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 : voir slides suivants
- ▶ Soit une structure initiale

Recherche locale à base de scores



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. 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 (heuristique) 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 : voir slides suivants
- ▶ Soit une structure initiale
- ▶ Les opérations de base



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. 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 (heuristique) 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 : voir slides suivants
- ▶ Soit une structure initiale
- ▶ Les opérations de base : ajout/suppression/modification d'un arc (dans le domaine de validité)



Propriétés des scores





Propriétés des scores

Soient D la base de donnée, T la topologie du réseau bayésien candidat et Θ ses paramètres.





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 :





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 :





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 ($\max L(T, \Theta : D)$).





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** :





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 ($\min Dim(T)$).





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 ($\min Dim(T)$).



Quelques scores (1) : AIC/BIC



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



Quelques scores (1) : AIC/BIC



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

- Akaike Information Criterion (Akaike, 70)





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

- Akaike Information Criterion (Akaike, 70)

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





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

- Akaike Information Criterion (Akaike, 70)

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

- Bayesian Information Criterion (Schartz, 78)

Quelques scores (1) : AIC/BIC



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

- **Akaike Information Criterion (Akaike, 70)**

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

- **Bayesian Information Criterion (Schartz, 78)**

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

Quelques scores (1) : AIC/BIC



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

- Akaike Information Criterion (Akaike, 70)

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

- Bayesian Information Criterion (Schartz, 78)

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

- Minimum Description Length (Lam and Bacchus, 93)

Quelques scores (1) : AIC/BIC



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

- **Akaike Information Criterion (Akaike, 70)**

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

- **Bayesian Information Criterion (Schartz, 78)**

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

- **Minimum Description Length (Lam and Bacchus, 93)**

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



Quelques scores (1) : AIC/BIC



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

- **Akaike Information Criterion (Akaike, 70)**

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

- **Bayesian Information Criterion (Schartz, 78)**

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

- **Minimum Description Length (Lam and Bacchus, 93)**

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

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 (1) : AIC/BIC



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

- **Akaike Information Criterion (Akaike, 70)**

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

- **Bayesian Information Criterion (Schartz, 78)**

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

- **Minimum Description Length (Lam and Bacchus, 93)**

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

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

- **Bayesian Dirichlet score Equivalent**



Quelques scores (1) : AIC/BIC



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

- **Akaike Information Criterion (Akaike, 70)**

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

- **Bayesian Information Criterion (Schartz, 78)**

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

- **Minimum Description Length (Lam and Bacchus, 93)**

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

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

- **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 : Greedy Hill Climbing



Recherche locale : Greedy Hill Climbing



Algorithme implémentant exactement ce qui est défini précédemment.

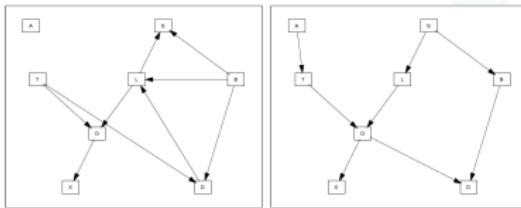


Recherche locale : Greedy Hill Climbing



Algorithme implémentant exactement ce qui est défini précédemment.

Réseau obtenu vs. théorique

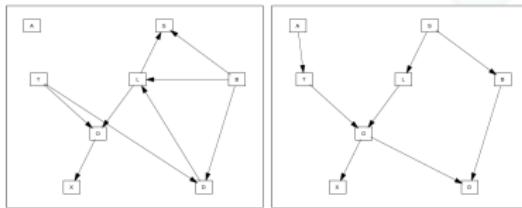


Recherche locale : Greedy Hill Climbing



Algorithme implémentant exactement ce qui est défini précédemment.

Réseau obtenu vs. théorique



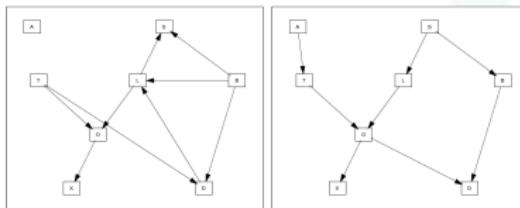
L'algorithme peut être bloqué sur des 'plateaux' et/ou converger vers des minima locaux.

Recherche locale : Greedy Hill Climbing



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

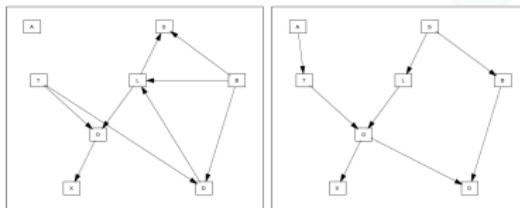
Principalement des méthodes de méta-heurisitiques :

Recherche locale : Greedy Hill Climbing



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 :

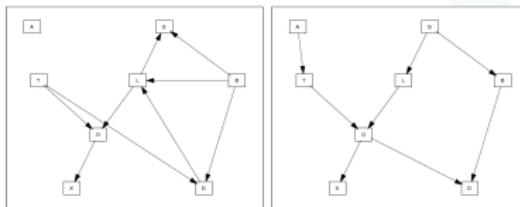
- ▶ Random restart

Recherche locale : Greedy Hill Climbing



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 :

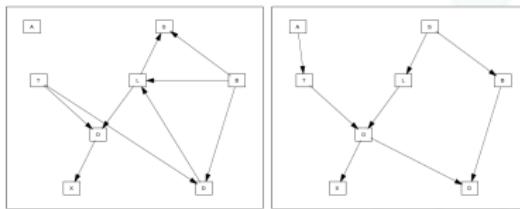
- ▶ Random restart
- ▶ TABU-search (liste des K dernières structures à éviter)

Recherche locale : Greedy Hill Climbing



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)

Retour sur le paradoxe de Simpson



38

```
learner=gum.BNLearn("simpson.csv")
bn=learner.learnBN()
gnb.sideBySide(bn,*[bn cpt(x) for x in bn.nodes()])]
```



		Drug	
Gender	With	Without	
F	0.7675	0.2325	
M	0.2468	0.7532	

Gender		
	F	M
	0.5080	0.4920

		Patient	
Drug	Gender	Healed	Sick
With	F	0.6683	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

```
ie=gum.LazyPropagation(bn)
gnb.sideBySide(ie.evidenceImpact(target="Patient",evs="Drug"),ie.evidenceImpact(target="Patient",evs=["Drug","Gender"]))
```

		Patient	
Drug	Healed	Sick	
With	0.5567	0.4433	
Without	0.4911	0.5089	

		Patient	
Drug	Gender	Healed	Sick
With	F	0.6683	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

Retour sur le paradoxe de Simpson



38

```
learner=gum.BNLearn("simpson.csv")
bn=learner.learnBN()
gnb.sideBySide(bn,[bn cpt(x) for x in bn.nodes()])]
```



		Drug	
		With	Without
Gender	F	0.7675	0.2325
	M	0.2468	0.7532

Gender		
	F	M
	0.5080	0.4920

		Patient	
		Healed	Sick
Drug	F	0.6683	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

```
ie=gum.LazyPropagation(bn)
gnb.sideBySide(ie.evidenceImpact(target="Patient",evs="Drug"),ie.evidenceImpact(target="Patient",evs=["Drug","Gender"]))
```

		Patient	
		Healed	Sick
Drug	With	0.5567	0.4433
	Without	0.4911	0.5089

		Patient	
		Healed	Sick
Drug	With	0.6683	0.3317
	Without	0.1982	0.8018
Gender	With	0.7793	0.2207
	Without	0.3993	0.6007

Conclusions sur Simpson

Retour sur le paradoxe de Simpson



38

```
learner=gum.BNLearn("simpson.csv")
bn=learner.learnBN()
gnb.sideBySide(bn,[bn cpt(x) for x in bn.nodes()])]
```



		Drug	
		With	Without
Gender	F	0.7675	0.2325
	M	0.2468	0.7532

Gender		
	F	M
	0.5080	0.4920

		Patient	
		Healed	Sick
Drug	F	0.6683	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

```
ie=gum.LazyPropagation(bn)
gnb.sideBySide(ie.evidenceImpact(target="Patient",evs="Drug"),ie.evidenceImpact(target="Patient",evs=["Drug","Gender"]))
```

		Patient	
		Healed	Sick
Drug	With	0.5567	0.4433
	Without	0.4911	0.5089

		Patient	
		Healed	Sick
Drug	With	0.6683	0.3317
	Without	0.1982	0.8018
Gender	With	0.7793	0.2207
	Without	0.3993	0.6007

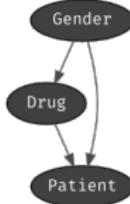
Conclusions sur Simpson

Retour sur le paradoxe de Simpson



38

```
learner=gum.BNLearn("simpson.csv")
bn=learner.learnBN()
gnb.sideBySide(bn,*[bn cpt(x) for x in bn.nodes()])]
```



		Drug	
		With	Without
Gender	F	0.7675	0.2325
	M	0.2468	0.7532

Gender		
	F	M
	0.5080	0.4920

		Patient	
		Healed	Sick
Drug	F	0.6683	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

```
ie=gum.LazyPropagation(bn)
gnb.sideBySide(ie.evidenceImpact(target="Patient",evs="Drug"),ie.evidenceImpact(target="Patient",evs=["Drug","Gender"]))
```

		Patient	
		Healed	Sick
Drug	With	0.5567	0.4433
	Without	0.4911	0.5089

		Patient	
		Healed	Sick
Drug	With	0.6683	0.3317
	Without	0.1982	0.8018
Gender	With	0.7793	0.2207
	Without	0.3993	0.6007

Conclusions sur Simpson

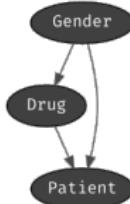
Quoi ?

Retour sur le paradoxe de Simpson



38

```
learner=gum.BNLearner("simpson.csv")
bn=learner.learnBN()
gnb.sideBySide(bn,*[bn cpt(x) for x in bn.nodes()])]
```



		Drug	
		With	Without
Gender	F	0.7675	0.2325
	M	0.2468	0.7532

Gender	
F	M
0.5080	0.4920

		Patient	
Drug	Gender	Healed	Sick
With	F	0.6682	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

```
ie=gum.LazyPropagation(bn)
gnb.sideBySide(ie.evidenceImpact(target="Patient",evs="Drug"),ie.evidenceImpact(target="Patient",evs=["Drug","Gender"]))
```

Patient	
Drug	Healed Sick
With	0.5567 0.4433
Without	0.4911 0.5089

		Patient	
Drug	Gender	Healed	Sick
With	F	0.6683	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

Conclusions sur Simpson



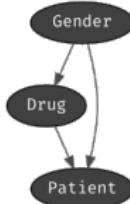
Quoi ?

Retour sur le paradoxe de Simpson



38

```
learner=gum.BNLearner("simpson.csv")
bn=learner.learnBN()
gnb.sideBySide(bn,*[bn cpt(x) for x in bn.nodes()])]
```



		Drug	
		With	Without
Gender	F	0.7675	0.2325
	M	0.2468	0.7532

Gender		
	F	M
	0.5080	0.4920

		Patient	
		Healed	Sick
Drug	F	0.6683	0.3317
	M	0.1982	0.8018
With	F	0.7793	0.2207
	M	0.3993	0.6007

```
ie=gum.LazyPropagation(bn)
gnb.sideBySide(ie.evidenceImpact(target="Patient",evs="Drug"),ie.evidenceImpact(target="Patient",evs=["Drug","Gender"]))
```

Patient		
Drug	Healed	Sick
With	0.5567	0.4433
Without	0.4911	0.5089

		Patient	
		Healed	Sick
Drug	F	0.6683	0.3317
	M	0.1982	0.8018
With	F	0.7793	0.2207
	M	0.3993	0.6007

Conclusions sur Simpson



Quoi ?

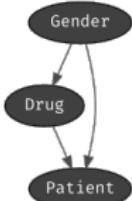
Comment ?

Retour sur le paradoxe de Simpson

38



```
learner=gum.BNLearn("simpson.csv")
bn=learner.learnBN()
gnb.sideBySide(bn,[bn.cpt(x) for x in bn.nodes()])
```



		Drug	
		With	Without
Gender	F	0.7675	0.2325
	M	0.2468	0.7532

Gender	
F	0.5080
M	0.4920

		Patient	
		Healed	Sick
Drug	F	0.6683	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

```
ie=gum.LazyPropagation(bn)
gnb.sideBySide(ie.evidenceImpact(target="Patient",evs="Drug"),ie.evidenceImpact(target="Patient",evs=["Drug","Gender"]))
```

		Patient	
		Healed	Sick
Drug	With	0.5567	0.4433
	Without	0.4911	0.5089

		Patient	
		Healed	Sick
Drug	F	0.6683	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

Conclusions sur Simpson



Quoi ?



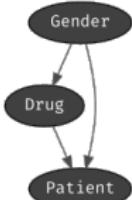
Comment ?

Retour sur le paradoxe de Simpson

38



```
learner=gum.BNLearn("simpson.csv")
bn=learner.learnBN()
gnb.sideBySide(bn,[bn.cpt(x) for x in bn.nodes()])
```



		Drug	
		With	Without
Gender	F	0.7675	0.2325
	M	0.2468	0.7532

Gender	
F	0.5080
M	0.4920

		Patient	
		Healed	Sick
Drug	F	0.6683	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

```
ie=gum.LazyPropagation(bn)
gnb.sideBySide(ie.evidenceImpact(target="Patient",evs="Drug"),ie.evidenceImpact(target="Patient",evs=["Drug","Gender"]))
```

		Patient	
		Healed	Sick
Drug	With	0.5567	0.4433
	Without	0.4911	0.5089

		Patient	
		Healed	Sick
Drug	F	0.6683	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

Conclusions sur Simpson



Quoi ?



Comment ?

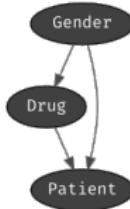
Et donc ?

Retour sur le paradoxe de Simpson



38

```
learner=gum.BNLearner("simpson.csv")
bn=learner.learnBN()
gnb.sideBySide(bn,{bn cpt(x) for x in bn.nodes()})
```



		Drug	
Gender	With	Without	
F	0.7675	0.2325	
M	0.2468	0.7532	

Gender		
	F	M
F	0.5080	0.4920
M		

		Patient	
Drug	Gender	Healed	Sick
With	F	0.6683	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

```
ie=gum.LazyPropagation(bn)
gnb.sideBySide(ie.evidenceImpact(target="Patient",evs="Drug"),ie.evidenceImpact(target="Patient",evs=["Drug","Gender"]))
```

		Patient	
Drug	Healed	Sick	
With	0.5567	0.4433	
Without	0.4911	0.5089	

		Patient	
Drug	Gender	Healed	Sick
With	F	0.6683	0.3317
	M	0.1982	0.8018
Without	F	0.7793	0.2207
	M	0.3993	0.6007

Conclusions sur Simpson



Quoi ?



Comment ?



Et donc ?

What do we gain with graphical Models ?



What do we gain with graphical Models ?



- ▶ A *compact model* : gains in time and space (tractable **exact inference, approximated inference, sampling**), etc.

What do we gain with graphical Models ?



- ▶ A *compact model* : gains in time and space (tractable **exact inference, approximated inference, sampling**), etc.
- ▶ A *learnable model*



What do we gain with graphical Models ?



- ▶ A *compact model* : gains in time and space (tractable **exact inference, approximated inference, sampling**), etc.
- ▶ A *learnable model*
- ▶ A *qualitative knowledge discovery from data*,

What do we gain with graphical Models ?



- ▶ A *compact model* : gains in time and space (tractable **exact inference, approximated inference, sampling**), etc.
- ▶ A *learnable model*
- ▶ A *qualitative knowledge discovery* from data,
 - ▶ Validation,
 - ▶ Prediction,
 - ▶ Explicability, etc.



What do we gain with graphical Models ?



- ▶ A *compact model* : gains in time and space (tractable **exact inference, approximated inference, sampling**), etc.
- ▶ A *learnable model*
- ▶ A *qualitative knowledge discovery* from data,
 - ▶ Validation,
 - ▶ Prediction,
 - ▶ Explicability, etc.
- ▶ And even, a possible *causal approach* from data.

What do we gain with graphical Models ?



- ▶ A *compact model* : gains in time and space (tractable **exact inference, approximated inference, sampling**), etc.
- ▶ A *learnable model*
- ▶ A *qualitative knowledge discovery* from data,
 - ▶ Validation,
 - ▶ Prediction,
 - ▶ Explicability, etc.
- ▶ And even, a possible *causal approach* from data.
- ▶ However,

What do we gain with graphical Models ?



- ▶ A *compact model* : gains in time and space (tractable **exact inference, approximated inference, sampling**), etc.
- ▶ A *learnable model*
- ▶ A *qualitative knowledge discovery* from data,
 - ▶ Validation,
 - ▶ Prediction,
 - ▶ Explicability, etc.
- ▶ And even, a possible *causal approach* from data.
- ▶ However,  still a NP-hard problem !



Introduction

Réseaux bayésiens (discrets)

Definition

Inference

Applications

Use Cases

Learning Bayesian Networks

aGrUM/pyAgrum





Un peu d'histoire

- ▶ The idea ? Build a set of common C++ codes for research team on PGM





Un peu d'histoire

- ▶ The idea ? Build a set of common C++ codes for research team on PGM
- ▶ Became public and Open Source on GitLab in 2016





Un peu d'histoire

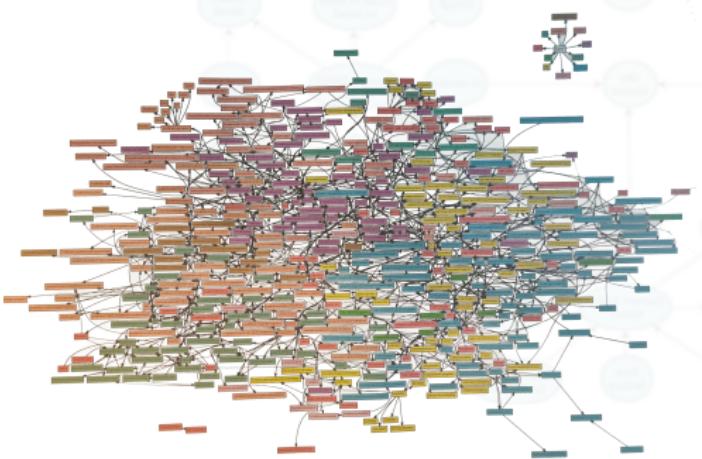
- ▶ The idea ? Build a set of common C++ codes for research team on PGM
- ▶ Became public and Open Source on GitLab in 2016
- ▶ 6000+ commits later thanks to 27 direct contributors...





Un peu d'histoire

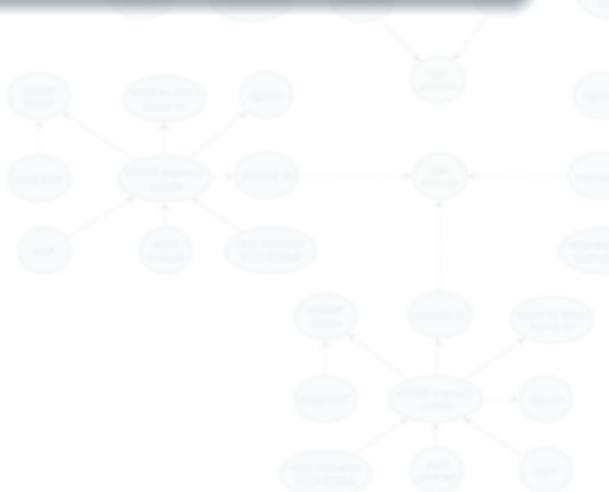
- ▶ The idea ? Build a set of common C++ codes for research team on PGM
- ▶ Became public and Open Source on GitLab in 2016
- ▶ 6000+ commits later thanks to 27 direct contributors...





aGrUM

Provides mainly low-level routines and components for PGM's algorithms but also high level components.





aGrUM

Provides mainly low-level routines and components for PGM's algorithms but also high level components.

- ▶ Optimized implementation of core data structures
- ▶ Efficient implementation of state of the art algorithms
- ▶ High level components
- ▶ Built-in tools to ensure memory leak free code

```
319 // diff the column types
320 const auto& database = score.database_.databaseTable();
321 const std::size_t nb_vars = database.numberOfVariables();
322 const std::vector< gum::learning::DBTranslatedValueType > col_types(
323     nb_vars, value::gum::learning::DBTranslatedValueType::DISCRETE);
324
325 // create the bootstrap estimator
326 DBRowGenerator<CompleteRow> generator_bootstrap(col_types);
327 DBRowGeneratorSet<> gensem_bootstrap;
328 gensem_bootstrap.insertGenerator(generator_bootstrap);
329 DBRowGeneratorParser<> parser_bootstrap(database.handler(),
330                                         gensem_bootstrap);
331 std::unique_ptr<ParamEstimator> parser_estimator_bootstrap(
332     createParamEstimator_( & parser_bootstrap, take_into_account_score));
333
334 // create the EM estimator
335 BayesNet< GUM_SCALAR > dummy_bn;
336 DBRowGenerator< GUM_SCALAR > generator_EM(col_types, dummy_bn);
337 DBRowGenerator<> gen_EM = generator_EM;
338 DBRowGeneratorSet<> gensem_EM;
339 gensem_EM.insertGenerator(generator_EM);
340 DBRowGeneratorParser<> parser_EM(database.handler(), gensem_EM);
341 std::unique_ptr<ParamEstimator> parser_estimator_EM(
342     createParamEstimator_( & parser_EM, take_into_account_score));
```



aGrUM

Provides mainly low-level routines and components for PGM's algorithms but also high level components.

- ▶ Optimized implementation of core data structures
- ▶ Efficient implementation of state of the art algorithms
- ▶ High level components
- ▶ Built-in tools to ensure memory leak free code

☞ sophisticated but with a difficult learning curve

```
310 // diff the column types
311 const int64_t database = score_database..._databaseTable();
312 const std::size_t nb_vars = database.numberOfVariables();
313 const std::vector< gum::learning::DBTranslatedValueType > col_types(
314     nb_vars, value_gum::learning::DBTranslatedValueType::DISCRETE);
315
316 // create the bootstrap estimator
317 DBRowGenerator<CompleteRow> generator_bootstrap(col_types);
318 generator_bootstrap.set(gmset_bootstrap);
319 gmset_bootstrap.insertGenerator(generator_bootstrap);
320 DBRowGeneratorParser<*> parser_bootstrap(database.handler(),
321                                     gmset_bootstrap);
322 std::unique_ptr< ParseEstimator > parse_estimator_bootstrap(
323     createParseEstimator_( & parser_bootstrap, take_into_account_score));
324
325 // create the EM estimator
326 BayesNet< GUM_SCALAR > dummy_bn;
327 DBRowGenerator< GUM_SCALAR > generator_EM(col_types, dummy_bn);
328 DBRowGenerator<*> gen_EM = generator_EM; // fix for g++-4.4
329 DBRowGeneratorSet<*> gmset_EM;
330 gmset_EM.insertGenerator(parser_EM);
331 DBRowGeneratorParser<*> parser_EM(database.handler(), gmset_EM);
332 std::unique_ptr< ParseEstimator > parse_estimator_EM(
333     createParseEstimator_( & parser_EM, take_into_account_score));
```



pyAgrum

Simplified high-level API in an easier language (Python)





pyAgrum

Simplified high-level API in an easier language (Python)

- ▶ easy to use





pyAgrum

Simplified high-level API in an easier language (Python)

- ▶ easy to use and **very** fast (compare to other python libraries)





pyAgrum

Simplified high-level API in an easier language (Python)

- ▶ easy to use and **very** fast (compare to other python libraries)
- ▶ Specific modules : dBN, CTBN, Causal reasoning





pyAgrum

Simplified high-level API in an easier language (Python)

- ▶ easy to use and **very** fast (compare to other python libraries)
- ▶ Specific modules : dBN, CTBN, Causal reasoning
- ▶ Interaction with pandas, scikit-learn, matplotlib, etc.





pyAgrum

Simplified high-level API in an easier language (Python)

- ▶ easy to use and **very** fast (compare to other python libraries)
- ▶ Specific modules : dBN, CTBN, Causal reasoning
- ▶ Interaction with pandas, scikit-learn, matplotlib, etc.
- ▶ LGPL+MIT dual license

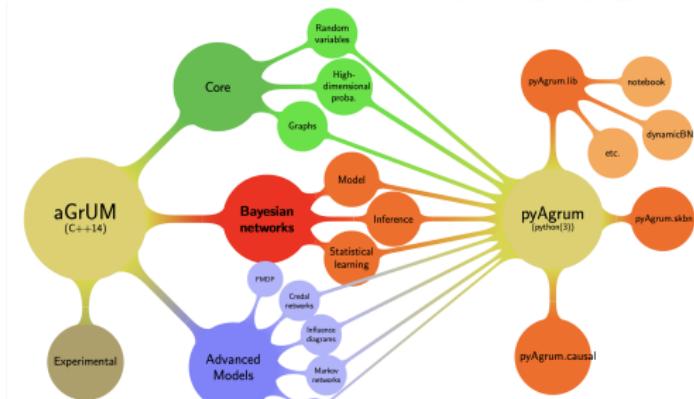




pyAgrum

Simplified high-level API in an easier language (Python)

- ▶ easy to use and **very** fast (compare to other python libraries)
- ▶ Specific modules : dBN, CTBN, Causal reasoning
- ▶ Interaction with pandas, scikit-learn, matplotlib, etc.
- ▶ LGPL+MIT dual license





To sum up

- ☞ Using SWIG to wrap aGrUM





To sum up

- ☞ Using SWIG to wrap aGrUM
- ☞ Access to low-level methods and high-level tools





To sum up

- ☞ Using SWIG to wrap aGrUM
- ☞ Access to low-level methods and high-level tools
- ☞ Cross platform and cross Python





To sum up

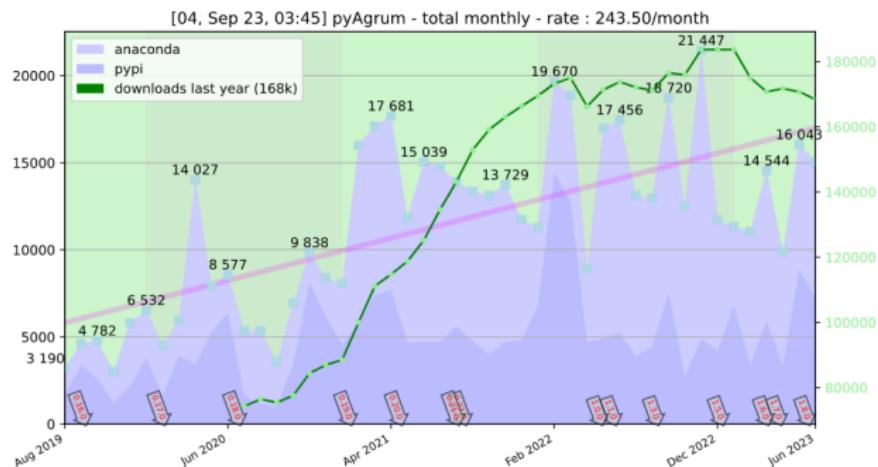
- ☞ Using SWIG to wrap aGrUM
- ☞ Access to low-level methods and high-level tools
- ☞ Cross platform and cross Python
- ☞ Hundreds of unit tests





To sum up

- ☞ Using SWIG to wrap aGrUM
- ☞ Access to low-level methods and high-level tools
- ☞ Cross platform and cross Python
- ☞ Hundreds of unit tests





Model	Domain	Features
	▶ Input/Output	<input checked="" type="checkbox"/> bif/bifxml/dsl/net/uai/ o3prm formats
	▶ Exact Inference	<input checked="" type="checkbox"/> Variable Elimination, Shafer-Shenoy Inference, Lazy Propagation <input checked="" type="checkbox"/> Marginal targets, joint targets <input checked="" type="checkbox"/> Optimized Relevance Reasoning <input checked="" type="checkbox"/> Incremental inference
	▶ Approximated Inference	<input checked="" type="checkbox"/> Gibbs Sampling, Weighted Sampling, Importance Sampling <input checked="" type="checkbox"/> Loopy Belief Propagation <input checked="" type="checkbox"/> Gibbs, Weighted, Importance LoopySampling
● Bayesian Network	▶ Parameter Learning	<input checked="" type="checkbox"/> Pure max-Likelihood, Laplace, Dirichlet <input checked="" type="checkbox"/> Multiple score <input checked="" type="checkbox"/> Parametric EM for missing values.
	▶ Structural Learning	<input checked="" type="checkbox"/> score-based learning : Greedy Hill-Climbing, local search with tabu-list, K2 <input checked="" type="checkbox"/> information-based learning : 3off2, mlic (with latent confounder variable discovery) <input checked="" type="checkbox"/> Graphical constraints (forced arcs, forbidden arcs, initial structures, partial order, possible arcs)
	▶ Algorithms	<input checked="" type="checkbox"/> Exact and approximated distance/divergence between BNs (KL, Bhattacharya, Hellinger) <input checked="" type="checkbox"/> Mutual information, entropy <input checked="" type="checkbox"/> Simulation (generation of csv files) <input checked="" type="checkbox"/> Markov Blanket, essential graph etc.
● Markov network	▶ Input/Output	<input checked="" type="checkbox"/> uai
	▶ Inference	<input checked="" type="checkbox"/> Shafer-Shenoy
● Influence Diagram	▶ Input/Output	<input checked="" type="checkbox"/> bifxml
	▶ Inference	<input checked="" type="checkbox"/> Junction Trees
● Probabilistic Relational Model	▶ Input/output	<input checked="" type="checkbox"/> O3PRM language parser
	▶ Exact inference	<input checked="" type="checkbox"/> Structured Variable Elimination (SVE)
● Credal Networks	▶ Approximated inference	<input checked="" type="checkbox"/> GL2U, MC Sampling
	▶ Input	
● FMDP	▶ Planning	<input checked="" type="checkbox"/> SVI, SPUDD
	▶ Multi-Valued Decision Diagram	<input checked="" type="checkbox"/> SPUnDD



Applications

Used in many applications with partners :

- ▶ during PhDs and internships : IBM / Teranga / Airbus / ANR ...





Applications

Used in many applications with partners :

- ▶ during PhDs and internships : IBM / Teranga / Airbus / ANR ...
- ▶ in collaboration in industrial projects





Applications

Used in many applications with partners :

- ▶ during PhDs and internships : IBM / Teranga / Airbus / ANR ...
- ▶ in collaboration in industrial projects
- ▶ by students in many courses

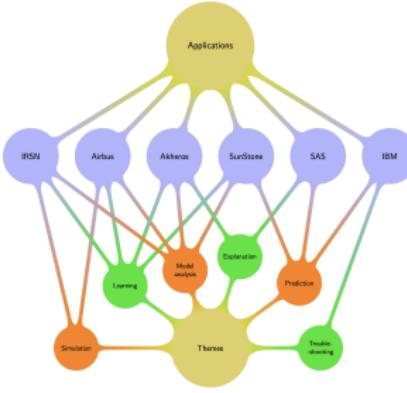




Applications

Used in many applications with partners :

- ▶ during PhDs and internships : IBM / Teranga / Airbus / ANR ...
- ▶ in collaboration in industrial projects
- ▶ by students in many courses





Pypi

Target : Python >3.7

Platform : win32, win64, Linux, OSX (Intel, M1)

```
pip install pyagrum
```

Conda

Target : Python >3.7

Platform : win64, Linux, OSX

```
conda install -c conda-forge pyagrum
```

+ Docker images, Binder ...