Bayesian Networks

Cédric Buche

ENIB

October 21, 2019

Bayesian Networks

2019-10-21

Bayesian Networks

Cédric Buche

ENIB

October 21, 2019

Page 1 :

The word "Bayesian" is common since the appearance of anti-spam filters of the same name. We propose, in this course, to present a powerful mathematical model: Bayesian networks. We will see how these tools upset the notion of classical reasoning, integrating a probabilistic framework. It thus becomes possible to reason about uncertainty, that is to say when one has incomplete information about a system that is not perfectly deterministic.

Page 2 :

Page 3 :

Uncertainty

- ▷ Uncertainty of events

$\Rightarrow$ event = probability

Uncertainty

▷ Uncertainty of events

⇒ event = probability

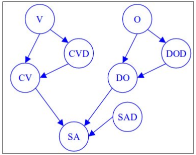
Page 4 :

The world is uncertain: indeed, can we be 100 % sure or not of the realization of an event, the veracity of a fact? Example: "it will rain tomorrow", it is certainly difficult to answer yes or no. Indeed, according to the weather today, according to the intensity of the rheumatism pain ... It can be deduced that it will rain "maybe", "surely", "certainly not" ... It to say that unconsciously, we will **associate a probability with the event** "it will rain tomorrow" which will be translated by adverbs previously mentioned. But we can probably announce that it is almost impossible to say that at 0 or 100 % =¿ we live in an uncertain world. Indeed, can I be 100% sure that tomorrow I will not have a flat tire? Can I be 100% sure that my presentation will last 2 UC? One sees it in these questions, **it is impossible to give the exact certainty of an event, on the other hand, one can quantify it, in particular thanks to the probabilities**.

$\boxed{\text{event} = \text{probability}}$

Thus, rather than reasoning with the veracity of a proposition, one can reason about the **trust given to a proposition** or the realization of an event. This trust will result in the attribution of a probability. We attribute a probability to an event in order to quantify the confidence that we have in the realization of this event.

- ▷ Probabilistic relations between facts
- ▷ Conditional reasoning, but not implicative



▷ Probabilistic relations between facts
▷ Conditional reasoning, but not implicative

Page 5 :

Let's be interested in models that allow to express probabilistic relations between sets of facts.
These relationships differ from logical relationships, they do not allow implicative but conditional reasoning. Two facts can be causally related without one involving the other. Example: in the database of an insurance company, for a majority of entries in a certain city, the items "parking ticket" are true when another item "the driver likes vegetables " is correlated. It would be too quick to conclude that the second fact involves statistically the first. A conditional probability analysis of the facts could reveal that the majority of the tickets are stuck on saturday, market day. There is indeed a common "cause" (or at least a condition of high probability), but logic does not have to intervene in this case.
Set of facts + conditional probabilities $\Rightarrow$ graphs

- $\rightarrow$ Reasoning, that is, calculating the conditional probability of any set knowing any other set
- $\rightarrow$ exploit the databases Databases = 10 thousand entries $\Rightarrow$ no expert can extract a probabilistic dependency structure alone
- $\rightarrow$ artificial learning

Conditional probability

- $X = \{true, false\}$ $P(X) = [0.2, 0.8]$
- $Y = \{small, normal, huge\}$ $P(Y) = [0.2, 0.6, 0.2]$
- $P(X = true, Y = small)$
- $P(X = true | Y = small)$

$X = true$			$X = false$		
$Y = small$	$Y = normal$	$Y = huge$	$Y = small$	$Y = normal$	$Y = huge$

Conditional probability

- $X = \{true, false\}$ $P(X) = [0.2, 0.8]$
- $Y = \{small, normal, huge\}$ $P(Y) = [0.2, 0.6, 0.2]$
- $P(X = true, Y = small)$
- $P(X = true | Y = small)$

$X = true$			$X = false$		
$Y = small$	$Y = normal$	$Y = huge$	$Y = small$	$Y = normal$	$Y = huge$

Page 6 :

The notion of conditional probability makes it possible to take into account a forecast of additional information. For example, if I randomly take a card, I is naturally to estimate that one in four chance to get a heart; but if I see a red reflection on the table, I correct my estimation to one chance out of two. This second estimatation is the probability of getting a heart knowing that the card is red. It is conditioned by the color of the card; therefore, conditional.

Application Domains

Diagnostic

Assuming a failure, a system based on Bayesian networks can determine the most likely causes that provided the problem.

Classification

Based on a number of system features, Bayesian networks will be able to categorize them.

2019-10-21

Bayesian Networks
Introduction
Application Domains

Application Domains

Diagnostic

Assuming a failure, a system based on Bayesian networks can determine the most likely causes that provided the problem.

Classification

Based on a number of system features, Bayesian networks will be able to categorize them.

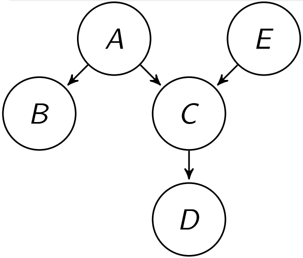
Page 7 :

Page 8 :

Causal graph

Acyclic oriented graph

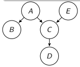
- nodes represent variables of interest of the domain
- arcs represent the relations between variables



Causal graph

Acyclic oriented graph

- ▷ nodes represent variables of interest of the domain
- ▷ arcs represent the relations between variables

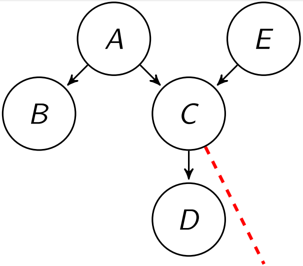


Page 9 :

Probability of the variables

Probability table

- $P(X|parents(X))$

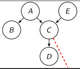


	$A, E =$			
	$True, True$	$True, False$	$False, True$	$False, False$
$C = True$				
$C = False$				

Probability of the variables

Probability table

- $P(X|parents(X))$



	$A, E =$			
	True, *True*	*True*, *False*	*False*, *True*	*False*, *False*
$C = True$				
$C = False$				

Page 10 :

Construction

- Experts
- Machine Learning (cf section 4)

Construction

- Experts
- Machine Learning (cf section 4)

Page 11 :

Example: An alarm problem

Alarm installed in stores and connected to a surveillance company

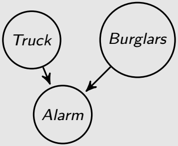
- The alarm is triggered in 2 stores, in which to send a team first?
- it is common that the passage of a truck (next to the store) triggers the alarm.
- burglars may not be detected
- alarm does not necessarily mean a theft
- there is no reason for the passage of a big truck is linked to the presence of thieves, and vice versa

Example: An alarm problem

Alarm installed in stores and connected to a surveillance company

- The alarm is triggered in 2 stores, in which to send a team first?
- it is common that the passage of a truck (next to the store) triggers the alarm.
- burglars may not be detected
- alarm does not necessarily mean a theft
- there is no reason for the passage of a big truck is linked to the presence of thieves, and vice versa

Page 12 :



Pratice

Application: object detector on a car

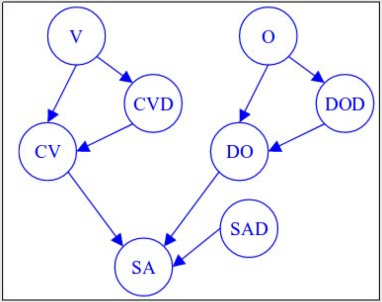
- On your new car, a device provides a warning if there is a object too close to your car. Your car uses an object detector to see the objects around your car. Considering the speed of the car and the position of objects, this warning device will trigger to warn you of danger if an object is too close to your car. Let's consider the following variables: V (speed of the car), CV (speedometer), CVD (meter Defective Speedometer), DO (Object Detector), O (Objects), DOD (Object Detector) defective), SA (warning system) and SAD (defective warning system).
  1. Draw a belief network, assuming the speedometer tends to be more inaccurate when speed increases and the object detector detects with difficulty small objects.
  2. Suppose there are only two possible values for speed (Normal or High) and that the speedometer gives the correct speed x % of the time when it works, but only y % of the time when it is defective. Then give the table of conditional probabilities for CV.

Pratice

Application: object detector on a car

- On your new car, a device provides a warning if there is a object too close to your car. Your car uses an object detector to see the objects around your car. Considering the speed of the car and the position of objects, this warning device will trigger to warn you of danger if an object is too close to your car. Let's consider the following variables: V (speed of the car), CV (speedometer), CVD (meter Defective Speedometer), DO (Object Detector), O (Objects), DOD (Object Detector) defective), SA (warning system) and SAD (defective warning system).
  1. Draw a belief network, assuming the speedometer tends to be more inaccurate when speed increases and the object detector detects with difficulty small objects.
  2. Suppose there are only two possible values for speed (Normal or High) and that the speedometer gives the correct speed x % of the time when it works, but only y % of the time when it is defective. Then give the table of conditional probabilities for CV.

Page 13 :



	V = Normale		V = Élevée	
	CVD	¬CVD	CVD	¬CVD
CV = Normale	y	x	1 - y	1 - x
CV = Élevée	1 - y	1 - x	y	x

Page 14 :

Bayesian Inference

Definitions of conditional probability

$$P(A|B) * P(B) = P(A, B) = P(B|A) * P(A)$$

Bayes theorem

- ▷ After making an observation on a variable, how will it affect the state of the other variables?
- ▷ $P(A|B) = \frac{P(B|A)*P(A)}{P(B)}$

→ Inversion of probabilities: *bottom-up* induction

Generalization

- ▷ H = Hidden, V = Visibles : $P(H|V) = \frac{P(H,V)}{\sum_H P(H,V)}$

Bayesian Inference

Definitions of conditional probability

$$P(A|B) * P(B) = P(A, B) = P(B|A) * P(A)$$

Bayes theorem

- After making an observation on a variable, how will it affect the state of the other variables?
- $P(A|B) = \frac{P(B|A)*P(A)}{P(B)}$

→ Inversion of probabilities: *bottom-up* induction

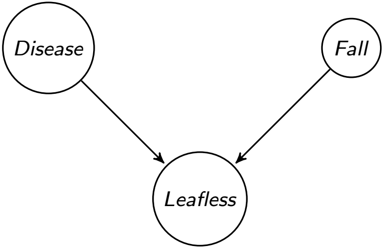
Generalization

- H = Hidden, V = Visibles : $P(H|V) = \frac{P(H,V)}{\sum_H P(H,V)}$

Page 15 :

Bayesian inference is the logical process for calculating or revising the probability of a hypothesis. This approach is governed by the use of strict rules of combination of probabilities, from which the Bayes theorem derives.

Bayesian Inference: example

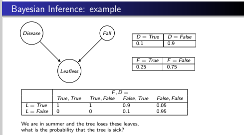


$D = True$	$D = False$
0.1	0.9

$F = True$	$F = False$
0.25	0.75

	$F, D =$			
	$True, True$	$True, False$	$False, True$	$False, False$
$L = True$	1	1	0.9	0.05
$L = False$	0	0	0.1	0.95

We are in summer and the tree loses these leaves,
what is the probability that the tree is sick?



Page 16 :

These tables permits to have all the information concerning the different variables of our graph, and it will be possible to use them. Being interested in the probability of being "sick" in a given situation, let's look at the calendar and realize that we are on July 12. We therefore deduce that we are not in fall. $P(F = 0) = 1$. Looking at the tree, we also observe that it loses its leaves. $P(L = 1) = 1$.
And one may wonder: "knowing these elements, is the tree sick?"
$P(D = 1|F = 0, L = 1) = ?$
$P(A|B) =$ what we want / possibility (we know B, we make A vary)
$P(D, F, L) = P(L|D, F) * P(F) * P(D)$

$$P(D = 1, F = 0, L = 1) = P(L = 1|F = 0, D = 1) * P(F = 0) * P(D = 1) = 0.9 * 0.75 * 0.1 = 0.0675$$

$$P(D = 0, F = 0, L = 1) = P(L = 1|F = 0, D = 0) * P(F = 0) * P(D = 0) = 0.05 * 0.75 * 0.9 = 0.03375$$

$$P(D = 1|F = 0, L = 1) = \frac{0.0675}{(0.0675 + 0.03375)} = 0.67$$

Practice

Detection of an animal disease

- ▷ In one animal population, one out of every hundred is affected by a disease.
- ▷ A test used to detect the disease is characterized by a probability of non-detection estimated at 5 % (false negative rate), and a probability of inadvertent detection equal to 1 % (false positive rate)

$\Rightarrow$ Propose a network and the associated probability tables

$\Rightarrow$ Estimate the probability of an individual being reached, knowing that the test is negative

Detection of an animal disease

- In one animal population, one out of every hundred is affected by a disease.
- A test used to detect the disease is characterized by a probability of non-detection estimated at 5 % (false negative rate), and a probability of inadvertent detection equal to 1 % (false positive rate)

⇒ Propose a network and the associated probability tables

⇒ Estimate the probability of an individual being reached, knowing that the test is negative

Page 17 :

Inference = NP-completeness problem

Méthodes exactes

- ▷ Bucket Elimination
- ▷ Message Passing (trees)
- ▷ Junction Tree

$\Rightarrow$ Problem = combinatorial explosion of these methods for strongly connected graphs

Approximate methods

- ▷ Sampling : Markov Chain Monte Carlo, ...
- ▷ Variational methods

Inference = NP-completeness problem

Méthodes exactes

- ▷ Bucket Elimination
- ▷ Message Passing (trees)
- ▷ Junction Tree
- ⇒ Problem = combinatorial explosion of these methods for strongly connected graphs

Approximate methods

- ▷ Sampling : Markov Chain Monte Carlo, . . .
- ▷ Variational methods

Page 18 :

Message Passing

Principle

- Each node sends messages to its neighbors
- The algorithm works only in the case of trees
- E = set of instantiated variables
- 2 types of messages λ and π will be used to calculate

$$\lambda(X) \propto P(Dx|X)$$

$$\pi(X) \propto P(X|Nx)$$

- and then we can show that

$$P(X|E=e) \propto \lambda(X)\pi(X)$$

Message Passing

Principle

- Each node sends messages to its neighbors
- The algorithm works only in the case of trees
- E = set of instantiated variables
- 2 types of messages λ and π will be used to calculate

$$\lambda(X) \propto P(D_X|X)$$

$$\pi(X) \propto P(X|N_X)$$

- and then we can show that

$$P(X|E=e) \propto \lambda(X)\pi(X)$$

Page 19 :

Message Passing

Messages λ

- For each child Y of X,

$$\lambda_Y(X = x) = \sum_y P(Y = y|X = x)\lambda(Y = y)$$

- How to calculate λ in each node ?
  - If X is instancied, $\lambda(X) = [001...0]$
    (the position of 1 corresponds to the value given to X)
  - else
    - if X is a leaf, $\lambda(X) = [1...1]$
    - else

$$\lambda(X = x) = \prod_{Y \in Enf(X)} \lambda_Y(X = x)$$

Message Passing

Messages λ

- For each child Y of X,

$$\lambda_Y(X = x) = \sum_y P(Y = y|X = x)\lambda(Y = y)$$

- How to calculate λ in each node ?
  - If X is instancied, $\lambda(X) = [001...0]$
    (the position of 1 corresponds to the value given to X)
  - else
    - if X is a leaf, $\lambda(X) = [1...1]$
    - else

$$\lambda(X = x) = \prod_{Y \in Enf(X)} \lambda_Y(X = x)$$

Page 20 :

Message Passing

Messages π

- For Z the only parent of X,

$$\pi_x(Z = z) = \pi(Z = z) \prod_{U \in Enf(Z) \setminus \{X\}} \lambda_u(Z = z)$$

- How to calculate π in each node ?
  - if X is instancied, $\pi(X) = [001...0]$
    (the position of 1 corresponds to the value given to X)
  - else
    - if X is a leaf, $\pi(X) = P(X)$
    - else

$$\pi(X = x) = \sum_z P(X = x | Z = z) \pi_x(Z = z)$$

Message Passing

Messages π

- For Z the only parent of X,

$$\pi_x(Z = z) = \pi(Z = z) \prod_{U \in Enf(Z) \setminus \{X\}} \lambda_u(Z = z)$$

- How to calculate π in each node ?
  - if X is instancied, $\pi(X) = [001...0]$
    (the position of 1 corresponds to the value given to X)
  - else
    - if X is a leaf, $\pi(X) = P(X)$
    - else
      $$\pi(X = x) = \sum_z P(X = x | Z = z)\pi_x(Z = z)$$

Page 21 :

Practice

Application : Juggler, the juggling robot

Juggler often drops the balls with which he juggles when his battery is low. According to previous experiences, the probability that he drops a ball when his battery is low is 0.9. On the other hand, when his battery is not weak, the probability that he drops a ball is only 0.01. Since the battery was recharged a short time ago, there is only a 5% chance that the battery will be low. A first (unreliable) vision system observes the robot and warns us when it thinks Juggler has dropped a bullet. Another system (independent of the first) acts in the same way.

1. Assuming all your variables are boolean, what variables will you choose to model this problem?
2. Propose a Bayesian B_1 network corresponding to the problem. Label the network nodes and clearly indicate the direction of the arcs between the nodes.
3. Represents the probability tables corresponding to the statement and your B_1 structure.
4. How is the quality of the two observers "coded" in the Bayesian network?

Practice

Application : Juggler, the juggling robot

Juggler often drops the balls with which he juggles when his battery is low. According to previous experiences, the probability that he drops a ball when his battery is low is 0.9. On the other hand, when his battery is not weak, the probability that he drops a ball is only 0.01. Since the battery was recharged a short time ago, there is only a 5% chance that the battery will be low. A first (unreliable) vision system observes the robot and warns us when it thinks Juggler has dropped a bullet. Another system (independent of the first) acts in the same way.

1. Assuming all your variables are boolean, what variables will you choose to model this problem?
2. Propose a Bayesian B_1 network corresponding to the problem. Label the network nodes and clearly indicate the direction of the arcs between the nodes.
3. Represents the probability tables corresponding to the statement and your B_1 structure.
4. How is the quality of the two observers "coded" in the Bayesian network?

Page 22 :

Practice

Application : the juggling robot (continuation)

We now suppose that the reliability of O_1 (the observer 1), respectively O_2 (the observer 2), is 70 % (resp. 90 %). We can illustrate the answers by the Bayesian network when virtual children are created.

1. Calculate messages *lambda* and *pi* flowing in the network when there is no added evidence. The objective is to make appear the $P(X)$, X sets of the variables of the network B_1.
2. O_1 observes that Juggler has dropped a bullet (Evidence Ev_1). What is the probability that the battery is low knowing this?
3. One adds then additional information: O_2 saw nothing (Evidence Ev_2) in the difference of O_1. What is the probability that the battery is low knowing these two informations?

Practice

Application : the juggling robot (continuation)

We now suppose that the reliability of O_1 (the observer 1), respectively O_2 (the observer 2), is 70 % (resp. 90 %). We can illustrate the answers by the Bayesian network when virtual children are created.

1. Calculate messages *lambda* and *pi* flowing in the network when there is no added evidence. The objective is to make appear the $P(X)$, X sets of the variables of the network B_1.
2. O_1 observes that Juggler has dropped a bullet (Evidence Ev_1). What is the probability that the battery is low knowing this?
3. One adds then additional information: O_2 saw nothing (Evidence Ev_2) in the difference of O_1. What is the probability that the battery is low knowing these two informations?

Page 23 :

Page 24 :

Bayesian network

- variables
- a graph between variables
- conditional probabilities

Learning

- parameters
- structure

Bayesian network
- ▷ variables
- ▷ a graph between variables
- ▷ conditional probabilities

Learning
- ▷ parameters
- ▷ structure

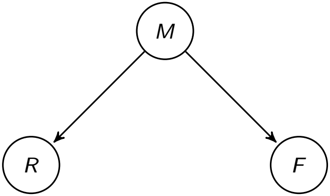
Page 25 :

	Known structure	**Unknown structure**
Data complete	Parametric statistical estimation	Discrete optimization on structures (discrete search algorithms)
Data incomplete	Parametric optimization (EM, gradient descent . . .)	Combination of methods (Structural EM, model mix)

	Known structure	**Unknown structure**
Data complete	Parametric statistical estimation	Discrete optimization on structures (discrete search algorithms)
Data incomplete	Parametric optimization (EM, gradient descent . . .)	Combination of methods (Structural EM, model mix)

Page 26 :

Known structure and complete data



$$P(M = m0) = 6/15 = 0.4$$

$$P(M = m1) = 8/15 = 0.53$$

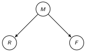
$$P(M = m2) = 1/15 = 0.07$$

$$P(F = OK|M = m0) = 1/6 = 0.17$$

$$P(F = BAD|M = m0) = 5/6 = 0.83$$

M	F	R
m0	BAD	O
m0	BAD	O
m0	BAD	O
m0	BAD	O
m0	BAD	N
m0	OK	O
m1	BAD	O
m1	BAD	N
m1	OK	O
m1	OK	N
m1	OK	O
m1	OK	N
m1	OK	O
m1	OK	N
m2	OK	N

Known structure and complete data



$$P(M = m0) = 6/15 = 0.4$$
$$P(M = m1) = 8/15 = 0.53$$
$$P(M = m2) = 1/15 = 0.07$$
$$P(F = OK|M = m0) = 1/6 = 0.17$$
$$P(F = BAD|M = m0) = 5/6 = 0.83$$

M	F	R
m0	BAD	O
m0	BAD	O
m0	BAD	O
m0	BAD	O
m0	BAD	N
m0	OK	O
m1	BAD	O
m1	BAD	N
m1	OK	O
m1	OK	N
m1	OK	O
m1	OK	N
m1	OK	O
m1	OK	N
m2	OK	N

Page 27 :

A priori of Dirichlet

- Problem : $P(F = BAD|M = m2) = 0/1$

  this configuration does not appear in our sample database

$\rightarrow$ pseudo coin toss *a priori* of $N*$ measures

Exemple

- A priori of Dirichlet on M spreaded over $m0$ and $m1$ = [50 50 0]

$$P(M = m0) = (6 + 50)/(15 + 100) = 0.487$$

$$P(M = m1) = (8 + 50)/(15 + 100) = 0.5043$$

$$P(M = m2) = (1 + 0)/(15 + 100) = 0.0087$$

- A priori of Dirichlet on $(F|M = mi) = [9\ 1]$

$$P(F = BAD|M = m2) = (0 + 1)/(1 + 10) = 0.09$$

A priori of Dirichlet

- Problem : $P(F = BAD|M = m2) = 0/1$
  this configuration does not appear in our sample database

→ pseudo coin toss *a priori* of $N*$ measures

Exemple

- A priori of Dirichlet on M spreaded over $m0$ and $m1$ = [50 50 0]

$$P(M = m0) = (6 + 50)/(15 + 100) = 0.487$$

$$P(M = m1) = (8 + 50)/(15 + 100) = 0.5043$$

$$P(M = m2) = (1 + 0)/(15 + 100) = 0.0087$$

- A priori of Dirichlet on $(F|M = mi) = [9\ 1]$

$$P(F = BAD|M = m2) = (0 + 1)/(1 + 10) = 0.09$$

Page 28 :

Known structure and incomplete data

Algorithme EM (Expectation Maximisation)

- initialize the parameters $\theta^{(0)}$
- $\boxed{\textbf{E}}$ estimate missing values from current settings $\theta^{(t)}$
  - = calculate $P(Xmanquant|Xmesures)$ in the current network
  - = make inferences
- $\boxed{\textbf{M}}$ re-estimate the parameters $\theta^{(t+1)}$ from completed data

Known structure and incomplete data

Algorithme EM (Expectation Maximisation)

- initialize the parameters $\theta^{(0)}$
- $\boxed{\textbf{E}}$ estimate missing values from current settings $\theta^{(t)}$
  - = calculate $P(Xmanquant|Xmesures)$ in the current network
  - = make inferences
- $\boxed{\textbf{M}}$ re-estimate the parameters $\theta^{(t+1)}$ from completed data

Page 29 :

- Initialization $P^{(0)}(M) = [\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\ ]$

M	F	R
m0	BAD	O
m0	BAD	O
?	BAD	O
m0	BAD	O
?	BAD	N
m0	OK	O
m1	BAD	O
m1	BAD	N
?	OK	O
m1	OK	N
m1	OK	O
m1	OK	N
m1	?	O
m1	OK	N
m2	OK	N

▷ Initialization $P^{(0)}(M) = [\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\ ]$

M	F	R
m0	BAD	O
m0	BAD	O
?	BAD	O
m0	BAD	O
?	BAD	N
m0	OK	O
m1	BAD	O
m1	BAD	N
?	OK	O
m1	OK	N
m1	OK	O
m1	OK	N
m1	?	O
m1	OK	N
m2	OK	N

Page 30 :

M	F	R	$P(M = m0)$	$P(M = m1)$	$P(M = m2)$
m0	BAD	O	1	0	0
m0	BAD	O	1	0	0
?	BAD	O	1/3	1/3	1/3
m0	BAD	O	1	0	0
?	BAD	N	1/3	1/3	1/3
m0	OK	O	1	0	0
m1	BAD	O	0	1	0
m1	BAD	N	0	1	0
?	OK	O	1/3	1/3	1/3
m1	OK	N	0	1	0
m1	OK	O	0	1	0
m1	OK	N	0	1	0
m1	?	O	0	1	0
m1	OK	N	0	1	0
m2	OK	N	0	0	1
TOTAL			5	8	2

Iteration 1

$\boxed{\textbf{E}}$

$\boxed{\textbf{M}}$

$$P^{(1)}(m0) = \frac{5}{15} = 0.333$$

$$P^{(1)}(m1) = \frac{8}{15} = 0.533$$

$$P^{(1)}(m2) = \frac{2}{15} = 0.133$$

M	F	R	$P(M=m0)$	$P(M=m1)$	$P(M=m2)$
m0	BAD	O	1	0	0
m0	BAD	O	1	0	0
?	BAD	O	1/3	1/3	1/3
m0	BAD	O	1	0	0
?	BAD	N	1/3	1/3	1/3
m0	OK	O	1	0	0
m1	BAD	O	0	1	0
m1	BAD	N	0	1	0
?	OK	O	1/3	1/3	1/3
m1	OK	N	0	1	0
m1	OK	O	0	1	0
m1	OK	N	0	1	0
m1	?	O	0	1	0
m1	OK	N	0	1	0
m2	OK	N	0	0	1
TOTAL			5	8	2

Iteration 1

$\boxed{\textbf{E}}$

$\boxed{\textbf{M}}$

$$P^{(1)}(m0) = \frac{5}{15} = 0.333$$

$$P^{(1)}(m1) = \frac{8}{15} = 0.533$$

$$P^{(1)}(m2) = \frac{2}{15} = 0.133$$

Page 31 :

M	F	R	$P(M = m0)$	$P(M = m1)$	$P(M = m2)$
m0	BAD	O	1	0	0
m0	BAD	O	1	0	0
?	BAD	O	0.333	0.533	0.133
m0	BAD	O	1	0	0
?	BAD	N	0.333	0.533	0.133
m0	OK	O	1	0	0
m1	BAD	O	0	1	0
m1	BAD	N	0	1	0
?	OK	O	0.333	0.533	0.133
m1	OK	N	0	1	0
m1	OK	O	0	1	0
m1	OK	N	0	1	0
m1	?	O	0	1	0
m1	OK	N	0	1	0
m2	OK	N	0	0	1
TOTAL			5	8.6	1.4

Iteration 2

E

M

$$P^{(2)}(m0) = \frac{5}{15} = 0.333$$

$$P^{(2)}(m1) = \frac{8.6}{15} = 0.573$$

$$P^{(2)}(m2) = \frac{1.4}{15} = 0.093$$

M	F	R	$P(M = m0)$	$P(M = m1)$	$P(M = m2)$
m0	BAD	O	1	0	0
m0	BAD	O	1	0	0
?	BAD	O	0.333	0.533	0.133
m0	BAD	O	1	0	0
?	BAD	N	0.333	0.533	0.133
m0	OK	O	1	0	0
m1	BAD	O	0	1	0
m1	BAD	N	0	1	0
?	OK	O	0.333	0.533	0.133
m1	OK	N	0	1	0
m1	OK	O	0	1	0
m1	OK	N	0	1	0
m1	?	O	0	1	0
m1	OK	N	0	1	0
m2	OK	N	0	0	1
TOTAL			5	8.6	1.4

Iteration 2

$\boxed{\textbf{E}}$

$\boxed{\textbf{M}}$

$$P^{(2)}(m0) = \frac{5}{15} = 0.333$$

$$P^{(2)}(m1) = \frac{8.6}{15} = 0.573$$

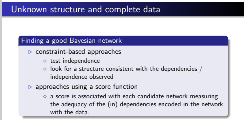
$$P^{(2)}(m2) = \frac{1.4}{15} = 0.093$$

Page 32 :

Unknown structure and complete data

Finding a good Bayesian network

- constraint-based approaches
  - test independence
  - look for a structure consistent with the dependencies / independence observed
- approaches using a score function
  - a score is associated with each candidate network measuring the adequacy of the (in) dependencies encoded in the network with the data.



Page 33 :

These two families of approaches are well founded (from a statistical point of view), that is to say that with enough data, the learning converges to a correct structure in both cases.
However, the former are sensitive to errors in independence tests, while for the latter the search for an optimal structure is an NP-hard problem. Indeed the number of graphs is + that exponential according to the number of variables (data) $\Rightarrow$ heuristics $\Rightarrow$ descent of the gradient / search taboo / simulated annealing

Unknown structure and incomplete data

Combine methods used for parameter learning and structure learning

1. for each G structure we estimate the optimal parameters P using either a gradient descent technique or the EM algorithm, then we associate a G score
2. we examine the graphs obtained from G with structure change operators and we choose the one with the highest score

Unknown structure and incomplete data

Combine methods used for parameter learning and structure learning

1. for each G structure we estimate the optimal parameters P using either a gradient descent technique or the EM algorithm, then we associate a G score
2. we examine the graphs obtained from G with structure change operators and we choose the one with the highest score

Page 34 :

Practice (1/6)

Frauds

Let's be interested in possible frauds using a credit card.
We will take into account the following variables:

- **Fraud** (marked as **F**) who can take for value true (marked as *O*) or false (marked as *N*)
- payment of the **Gaz** bill (marked as **G**) who can take for value true (marked as *O*) or false (marked as *N*)
- buy in a **Jewelery** (marked as **B**) who can take for value true (marked as *O*) or false (marked as *N*)
- **Age** (marked as **A**) who can take for value :
  - Under 30 years old (marked as <30)
  - Between 30 and 50 years old (marked as *30-50*)
  - Upper than 50 years old (marked as >50)
- **Sex** (marked as**S**) who can take for value Man (marked as *H*) or Woman (marked as *F*)

Practice (1/6)

Frauds

Let's be interested in possible frauds using a credit card.
We will take into account the following variables:

- **Fraud** (marked as **F**) who can take for value true (marked as *O*) or false (marked as *N*)
- payment of the **Gaz** bill (marked as **G**) who can take for value true (marked as *O*) or false (marked as *N*)
- buy in a **Jewelery** (marked as **B**) who can take for value true (marked as *O*) or false (marked as *N*)
- **Age** (marked as **A**) who can take for value :
  - Under 30 years old (marked as <30)
  - Between 30 and 50 years old (marked as *30-50*)
  - Upper than 50 years old (marked as >50)
- **Sex** (marked as**S**) who can take for value Man (marked as *H*) or Woman (marked as *F*)

Page 35 :

Practice (2/6)

Representation

Build the representation of the Bayesian network (concepts + links) highlighting the following relations:

- A fraud can be made to pay in jewelery or its gas bill
- The purchase in jewelery depends on the age and sex of the person

Join Distribution

Take into account the previous Bayesian network to simplify the expression of the joint probability of this problem.
Reminder: the joined probability of a system S corresponds to:
$P(S) = P(S1)P(S2/S1)P(S3/S2, S1)...P(Sn/Sn-1...S1)$
Concretely, it is a table that gives all the probabilities of all combinations of variables.

Practice (2/6)

Representation

Build the representation of the Bayesian network (concepts + links) highlighting the following relations:

- A fraud can be made to pay in jewelery or its gas bill
- The purchase in jewelery depends on the age and sex of the person

Join Distribution

Take into account the previous Bayesian network to simplify the expression of the joint probability of this problem.
Reminder: the joined probability of a system S corresponds to:
$P(S) = P(S1)P(S2/S1)P(S3/S2, S1)...P(Sn/Sn-1...S1)$
Concretely, it is a table that gives all the probabilities of all combinations of variables.

Page 36 :

Practice (3/6)

Inference

Consider the following probabilities:

P(F= O)	0.00001
P(A= <30)	0.25
P(A= 30-50)	0.4
P(S= H)	0.5
P(G= O/ F= O)	0.2
P(G= O/ F= N)	0.01
P(B= O/ F= O, A= * , S= *)	0.05
P(B= O/ F= N, A= <30 , S= H)	0.0001
P(B= O/ F= N, A= 30-50 , S= H)	0.0004
P(B= O/ F= N, A= >50 , S= H)	0.0002
P(B= O/ F= N, A= <30 , S= F)	0.0005
P(B= O/ F= N, A= 30-50 , S= F)	0.002
P(B= O/ F= N, A= >50 , S= F)	0.001

* : means all possibilities are allowed

Practice (3/6)

Inference

Consider the following probabilities:

P(F= O)	0.00001
P(A= <30)	0.25
P(A= 30-50)	0.4
P(S= H)	0.5
P(G= O/ F= O)	0.2
P(G= O/ F= N)	0.01
P(B= O/ F= O, A= * , S= *)	0.05
P(B= O/ F= N, A= <30 , S= H)	0.0001
P(B= O/ F= N, A= 30-50 , S= H)	0.0004
P(B= O/ F= N, A= >50 , S= H)	0.0002
P(B= O/ F= N, A= <30 , S= F)	0.0005
P(B= O/ F= N, A= 30-50 , S= F)	0.002
P(B= O/ F= N, A= >50 , S= F)	0.001

* : means all possibilities are allowed

Page 37 :

Practice (4/6)

Inference

1. Express the probability that a 40-year-old man buying a jewel will make a fraud using the following property:

$$P(H/V) = \frac{\sum_{H'} P(H, V, H')}{\sum_{H,H'} P(H, V, H')}$$

   With V = instantiated variables (visible), H and H '= hidden variables
2. Calculate P(F = O , A= 30-50, S = H , G = * , B = O)
3. Calculate P(F = N , A= 30-50, S = H , G = * , B = O)
4. Calculate the probability that a 40-year-old man buying a jewel will cheat using previous results

Practice (4/6)

Inference

1. Express the probability that a 40-year-old man buying a jewel will make a fraud using the following property:

$$P(H/V) = \frac{\sum_{H'} P(H,V,H')}{\sum_{H,H'} P(H,V,H')}$$

   With V = instantiated variables (visible), H and H '= hidden variables
2. Calculate P(F = O , A= 30-50, S = H , G = * , B = O)
3. Calculate P(F = N , A= 30-50, S = H , G = * , B = O)
4. Calculate the probability that a 40-year-old man buying a jewel will cheat using previous results

Page 38 :

Practice (5/6)

Learning : known structure, incomplete data

The following data are extracted from statistics representing fraud information using a bank card for the purchase of jewelery and the payment of gas bills.

G	F	A	S	B
O	N	30-50	F	O
O	?	< 30	H	N
O	N	30-50	F	N
N	N	>50	F	N
N	?	> 50	F	O
?	O	30-50	F	N

Practice (5/6)

Learning : known structure, incomplete data

The following data are extracted from statistics representing fraud information using a bank card for the purchase of jewelery and the payment of gas bills.

G	F	A	S	B
O	N	30-50	F	O
O	?	< 30	H	N
O	N	30-50	F	N
N	N	>50	F	N
N	?	> 50	F	O
?	O	30-50	F	N

Page 39 :

Exercice (6/6)

Learning : known structure, incomplete data

Let's calculate the probabilities having or not a fraud using the EM algorithm

1. Initialization: Explain the initialization of the EM $P^{(0)}(F)$ algorithm using the information P (F = O) = 0.00001
2. Iterations: complete the tables and calculate for the first 2 iterations of the EM algorithm $P^{(1)}(F)$ et $P^{(2)}(F)$

G	F	A	S	B	P(F= O)	P(F= N)
O	N	30-50	F	O		
O	?	< 30	H	N		
O	N	30-50	F	N		
N	N	>50	F	N		
N	?	> 50	F	O		
?	O	30-50	F	N		
				TOTAL		

3. Propose and explain how to put in place a stop condition

Exercice (6/6)

Learning : known structure, incomplete data

Let's calculate the probabilities having or not a fraud using the EM algorithm

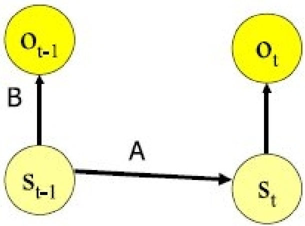
1. Initialization: Explain the initialization of the EM $P^{(0)}(F)$ algorithm using the information P (F = O) = 0.00001
2. Iterations: complete the tables and calculate for the first 2 iterations of the EM algorithm $P^{(1)}(F)$ et $P^{(2)}(F)$

G	F	A	S	B	P(F= O)	P(F= N)
O	N	30-50	F	O		
O	?	< 30	H	N		
O	N	30-50	F	N		
N	N	>50	F	N		
N	?	> 50	F	O		
?	O	30-50	F	N		
				TOTAL		

3. Propose and explain how to put in place a stop condition

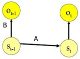
Page 40 :

Page 41 :



- $P(S_0)$ intitial state of S
- $P(O_t|S_t)$ independant of t
- $P(S_t|S_{t-1})$

- Inference : Forward-Backward = Message Passing
- Explication : Viterbi = Most Likely Explanation
- Learning : Baum&Welch = EM



- $P(S_0)$ intitial state of S
- $P(O_t|S_t)$ independant of t
- $P(S_t|S_{t-1})$

- Inference : Forward-Backward = Message Passing
- Explication : Viterbi = Most Likely Explanation
- Learning : Baum&Welch = EM

Page 42 :

Decision concept

Example

- ▷ A 12-year-old child goes to the emergency room
  - ◇ he has a stomach ache since 8h
  - ◇ he vomited once
  - ◇ he ate at the restaurant recently
  - ◇ no medical "liabilities", no treatment in progress
  - ◇ first examination: diffuse abdominal pain, mean CBC

$\Rightarrow$ Should we send the boy to be operated on appendicitis? to put it in observation? leave it go ?

Example

- ▷ A 12-year-old child goes to the emergency room
  - ◇ he has a stomach ache since 8h
  - ◇ he vomited once
  - ◇ he ate at the restaurant recently
  - ◇ no medical "liabilities", no treatment in progress
  - ◇ first examination: diffuse abdominal pain, mean CBC

⇒ Should we send the boy to be operated on appendicitis? to put it in observation? leave it go ?

Page 43 :

A decision problem has 3 components:

1. the values (the "symptoms", the observables, ...to take into account)
2. the actions (the choices proposed to the decision maker)
3. the consequences

Hypotheses :

- values, options and consequences are given
- the decision maker can arrange the consequences by order preferably: **utility function**

A decision problem has 3 components:

1. the values (the "symptoms", the observables, . . . to take into account)
2. the actions (the choices proposed to the decision maker)
3. the consequences

Hypotheses :

- values, options and consequences are given
- the decision maker can arrange the consequences by order preferably: **utility function**

Page 44 :

U utility function, defined for consequences

- ▷ $U(c_1) > U(c_2)$ if and only if the decision maker prefers the consequence c_1 to c_2
- ▷ $U(c_1) = U(c_2)$ if and only if the decoder has no preference between c_1 to c_2

Exemple

- ▷ Consequence c = achieve UV — fail UV
- ▷ $U(c_1) = 5$ ECTS credits and $U(c_2) = 0$

U does not have to be normalized, and does not represent necessarily a probability

U utility function, defined for consequences

- $U(c_1) > U(c_2)$ if and only if the decision maker prefers the consequence c_1 to c_2
- $U(c_1) = U(c_2)$ if and only if the decoder has no preference between c_1 to c_2

Exemple

- Consequence c = achieve UV — fail UV
- $U(c_1) = 5$ ECTS credits and $U(c_2) = 0$

U does not have to be normalized, and does not represent necessarily a probability

Page 45 :

How to model the reasoning?

How to find the optimal decision?

- Decision tree
  - make a decision without probabilities:
    - Maximax, Maximin, Minimax, d'Hurwicz, de Laplace
  - make a decision with probabilities:
    - Average Utility
- Influence diagram

How to model the reasoning?

How to find the optimal decision?

- Decision tree
  - make a decision without probabilities:
    - Maximax, Maximin, Minimax, d'Hurwicz, de Laplace
  - make a decision with probabilities:
    - Average Utility
- Influence diagram

Page 46 :

Example "real estate"

- Real estate investment: should we invest in
  - a residence
  - a building
  - an apartment
  - no investment
- This will depend on the state of the real estate market:
  - Important — Medium — Low
- Profit according to the decision and the state of the market

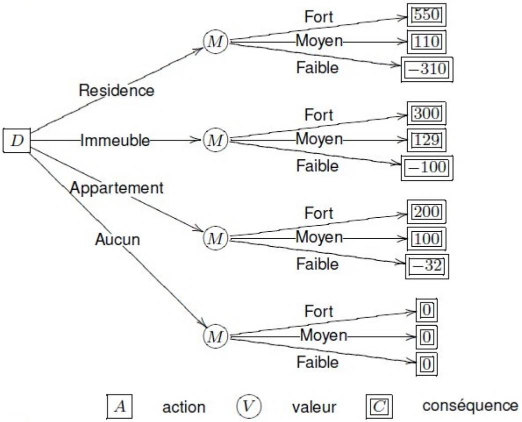
	Important	Medium	Low
Residence	550	110	-310
Building	300	129	-100
Apartment	200	100	-32
None	0	0	0

Example "real estate"

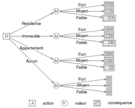
- Real estate investment: should we invest in
  - a residence
  - a building
  - an apartment
  - no investment
- This will depend on the state of the real estate market:
  - Important — Medium — Low
- Profit according to the decision and the state of the market

	Important	Medium	Low
Residence	550	110	-310
Building	300	129	-100
Apartment	200	100	-32
None	0	0	0

Decision tree



Decision tree



Page 48 :

Maximax

- The criterion of the optimistic decision maker
- "reduce" the maximum usefulness of each "value"
- choose the decision that maximizes maximum utility

Maximin

- The pessimistic decision maker criterion
- "reduce" the minimal utility of each "value"
- choose the decision that has the greatest minimum utility (the "least worst")

Maximax

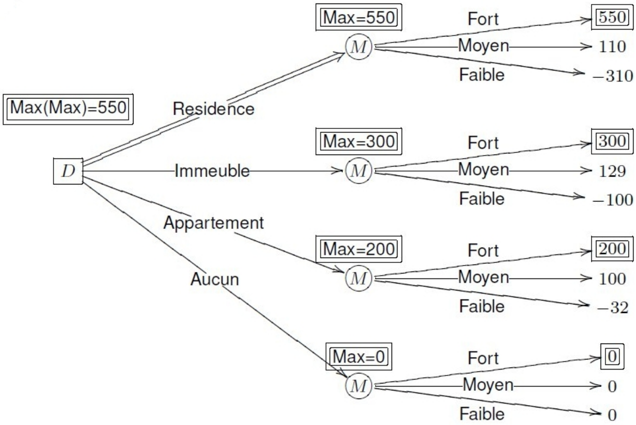
- The criterion of the optimistic decision maker
- "reduce" the maximum usefulness of each "value"
- choose the decision that maximizes maximum utility

Maximin

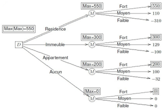
- The pessimistic decision maker criterion
- "reduce" the minimal utility of each "value"
- choose the decision that has the greatest minimum utility (the "least worst")

Page 49 :

Maximax

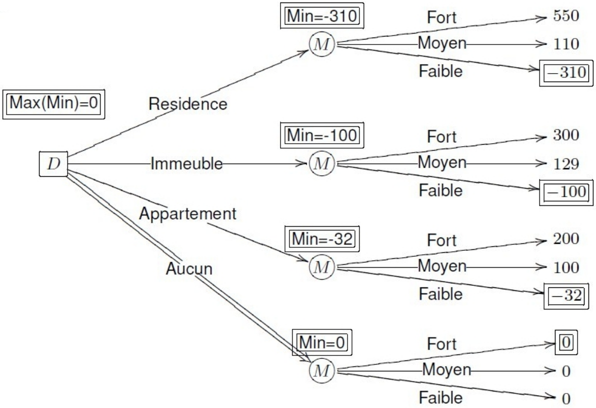

Max(Max)=550
D
Residence
Immeuble
Appartement
Aucun
Max=550
Max=300
Max=200
Max=0
M
Fort
Moyen
Faible
550
110
−310
300
129
−100
200
100
−32
0
0
0

Maximax

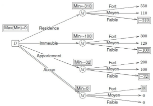


Page 50 :

Maximin


Max(Min)=0
D
Residence
Immeuble
Appartement
Aucun
Min=-310
M
Fort
Moyen
Faible
550
110
−310
Min=-100
M
Fort
Moyen
Faible
300
129
−100
Min=-32
M
Fort
Moyen
Faible
200
100
−32
Min=0
M
Fort
Moyen
Faible
0
0
0

Maximin



Page 51 :

Utilité moyenne

Expected Utility (EU)

- action a_1 is related to the consequence c by $P(c|a_1)$
- action a_2 is related to the consequence c by $P(c|a_2)$
- we will prefer a_1 à a_2 si :

$$\sum_c U(c)P(c|a_1) > \sum_c U(c)P(c|a_2)$$

$$EU(a_1) > EU(a_2)$$

Decide = choose the action that maximizes the average utility

Utilité moyenne

Expected Utility (EU)

- action a_1 is related to the consequence c by $P(c|a_1)$
- action a_2 is related to the consequence c by $P(c|a_2)$
- we will prefer a_1 à a_2 si :

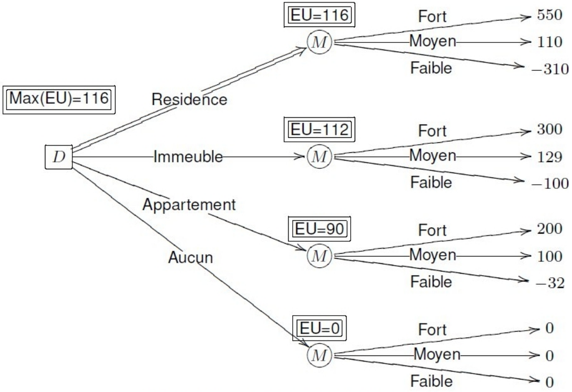
$$\sum_c U(c)P(c|a_1) > \sum_c U(c)P(c|a_2)$$

$$EU(a_1) > EU(a_2)$$

Decide = choose the action that maximizes the average utility

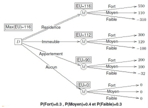
Page 52 :

Average utility



P(Fort)=0.3 , P(Moyen)=0.4 et P(Faible)=0.3

Average utility



Page 53 :

Decision tree

Pro

- ▷ Structure adapted to find the optimal solution

Cons

- ▷ The size of the tree quickly becomes huge!

$\Rightarrow$ Diagrams of influence

Decision tree

Pro

- ▷ Structure adapted to find the optimal solution

Cons

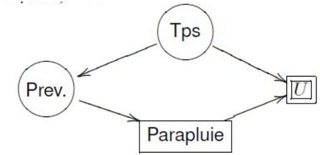
- ▷ The size of the tree quickly becomes huge!

⇒ Diagrams of influence

Page 54 :

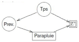
Example: Should we take his umbrella tomorrow?

- $P(Time = Rain) = 0.3$
- $P(Forecast = Rainy|Rain = Rain) = 0.6$
- $P(Forecast = Cloudy|Rain = Rain) = 0.25$
- $P(Forecast = Sun|Time = Rain) = 0.15$
- $P(Forecast = Rainy|Time = NoWindle) = 0.1$
- $P(Forecast = Cloudy|Time = NoWeather) = 0.2$
- $P(Forecast = Sun|Time = NoWind) = 0.7$
- $U(NoRain, Umbrella) = 20$
- $U(NoWater, NoWithout) = 100$
- $U(Rain, Umbrella) = 70$
- $U(Rain, NoUmbrella) = 0$



Example: Should we take his umbrella tomorrow?

- $P(Time = Rain) = 0.3$
- $P(Forecast = Rainy | Rain = Rain) = 0.6$
- $P(Forecast = Cloudy | Rain = Rain) = 0.25$
- $P(Forecast = Sun | Time = Rain) = 0.15$
- $P(Forecast = Rainy | Time = NoWindle) = 0.1$
- $P(Forecast = Cloudy | Time = NoWeather) = 0.2$
- $P(Forecast = Sun | Time = NoWind) = 0.7$
- $U(NoRain, Umbrella) = 20$
- $U(NoWater, NoWithout) = 100$
- $U(Rain, Umbrella) = 70$
- $U(Rain, NoUmbrella) = 0$



Page 55 :

Influence Diagram

- Using the formalism of Bayesian networks
- Separation of decisions into 2 families
  - tests
  - actions
    - "non-intervening" actions (eg Take Umbrella)
    - "intervening" actions: acts on certain variables
    - rule to follow: the impact of an "intervening" action can only follow the direction of the arrows

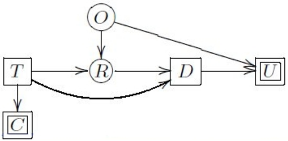
Influence Diagram

- Using the formalism of Bayesian networks
- Separation of decisions into 2 families
  - tests
  - actions
    - "non-intervening" actions (eg Take Umbrella)
    - "intervening" actions: acts on certain variables
    - rule to follow: the impact of an "intervening" action can only follow the direction of the arrows

Page 56 :

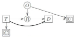
Oil Drilling Example

- Decisions
  - D = Drilling (70 Euros) — No drilling
  - T = seismic Test = Yes (10 Euros) — No
- Variables
  - O = state of the Olier = dry — wet — soaked
  - R = Reservoir tank = big — little — no trace of oil
- Utilities
  - C = test Cost
  - U = drill Utility



Oil Drilling Example

- Decisions
  - D = Drilling (70 Euros) — No drilling
  - T = seismic Test = Yes (10 Euros) — No
- Variables
  - O = state of the Olier = dry — wet — soaked
  - R = Reservoir tank = big — little — no trace of oil
- Utilities
  - C = test Cost
  - U = drill Utility



Page 57 :

Oil Drilling Example

1. Utilities
   - $U(D, O)$

	O=dry	0=wet	O=soaked
D=Yes	-70	50	200
D=No	0	0	0

   - $C(T)$

T=Yes	T=No
-10	0

2. Probabilities
   - $P(O)$

O=dry	0=wet	O=soaked
0.5	0.3	0.2

   - $P(R|O, T = No) = 1/3$
   - $P(R|O, T = Yes)$

	O=dry	0=wet	O=soaked
R = nothing	0.6	0.3	0.1
R = little	0.3	0.4	0.4
R = huge	0.1	0.3	0.5

Oil Drilling Example

1. Utilities
   - $U(D, O)$

	O=dry	0=wet	O=soaked
D=Yes	-70	50	200
D=No	0	0	0

   - $C(T)$

T=Yes	T=No
-10	0

2. Probabilities
   - $P(O)$

O=dry	0=wet	O=soaked
0.5	0.3	0.2

   - $P(R|O, T = No) = 1/3$
   - $P(R|O, T = Yes)$

	O=dry	0=wet	O=soaked
R = nothing	0.6	0.3	0.1
R = little	0.3	0.4	0.4
R = huge	0.1	0.3	0.5

Page 58 :

Oil Drilling Example

- Should we drill?

$$EU(D = No) = 0$$

$$EU(D = Yes) = \sum_{O} U(D = Yes, O)P(O)$$

$$EU(D = Yes) = 0.5 * -70 + 0.3 * 50 + 0.2 * 200 = +20$$

$$MEU(D|T = No) = Max(0, +20) = +20$$

- if we do not test, the best decision is to drill

Oil Drilling Example

- Should we drill?

$$EU(D = No) = 0$$

$$EU(D = Yes) = \sum_{O} U(D = Yes, O)P(O)$$

$$EU(D = Yes) = 0.5 * -70 + 0.3 * 50 + 0.2 * 200 = +20$$

$$MEU(D|T = No) = Max(0, +20) = +20$$

- if we do not test, the best decision is to drill

Page 59 :

Oil Drilling Example

- Should we do a seismic Test?

$$P(O|R = nothing)\alpha P(R = nothing|O)P(O) = [0.60.30.1]*[0.50.30.$$

$$P(O|R = nothing)\alpha[0.30.090.02]$$

$$P(O|R = nothing) = [0.30.090.02]/0.41 = [0.7320.220.049]$$

$$EU(D = Yes|R = nothing) = \sum_O U(D = Yes, O)P(O|R = nothing)$$

$$EU(D = Yes|R = nothing) = -30.5$$

- and continue for all the values of (D, R) to get the table $EU(D|R, T = Yes)$

Oil Drilling Example

- Should we do a seismic Test?

$$P(O|R = nothing) \alpha P(R = nothing|O)P(O) = [0.60.30.1]*[0.50.30.$$

$$P(O|R = nothing) \alpha [0.30.090.02]$$

$$P(O|R = nothing) = [0.30.090.02]/0.41 = [0.7320.220.049]$$

$$EU(D = Yes|R = nothing) = \sum_{O} U(D = Yes, O)P(O|R = nothing)$$

$$EU(D = Yes|R = nothing) = -30.5$$

- and continue for all the values of (D, R) to get the table $EU(D|R, T = Yes)$

Page 60 :

Oil Drilling Example

- Should we do a seismic Test? (end)

$$MEU(D|R, T = Yes) = [87.532.90]$$

$$MEU(D|R, T = Yes) = \sum_S MEU(D|S, T = Yes)P(R) - C(T = Yes$$

$$MEU(D|T = Yes) = +22.5$$

$$MEU(D|T = Yes) = 22.5 > MEU(D|T = No) = 20$$

- We must do the test!

Oil Drilling Example

- Should we do a seismic Test? (end)

$$MEU(D|R, T = Yes) = [87.532.90]$$

$$MEU(D|R, T = Yes) = \sum_{S} MEU(D|S, T = Yes)P(R) - C(T = Yes)$$

$$MEU(D|T = Yes) = +22.5$$

$$MEU(D|T = Yes) = 22.5 > MEU(D|T = No) = 20$$

- We must do the test!

Page 61 :

Find the optimal policy

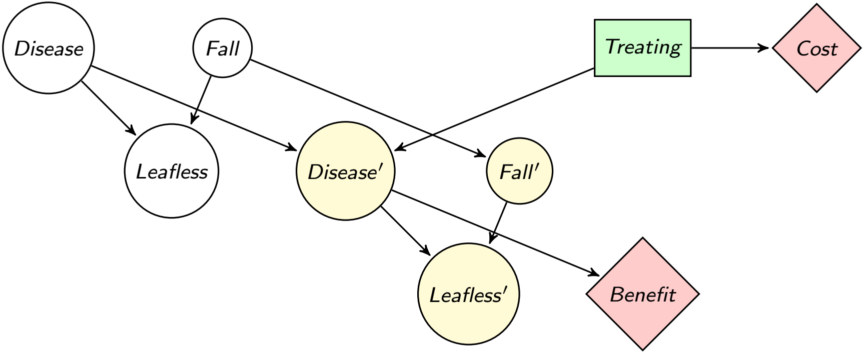
- Jensen et al. 1994 : From Influence Diagrams to Junction Trees
- Zhang 1996 : Probabilistic Inference in Influence Diagrams
- Lauritzen & Nilsson 2001 : Representing and Solving Decision Problems with Limited Information

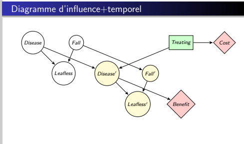
Find the optimal policy

- Jensen et al. 1994 : From Influence Diagrams to Junction Trees
- Zhang 1996 : Probabilistic Inference in Influence Diagrams
- Lauritzen & Nilsson 2001 : Representing and Solving Decision Problems with Limited Information

Page 62 :

Diagramme d'influence+temporel





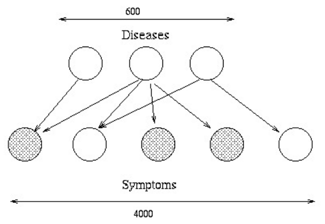
Page 63 :

The gardener's problem is that he wants to sell his apples on the market, and of course he wants to make the biggest profit. We added the three variables 'Disease', 'Autumn' and 'Loss', which is the same as the initial three but **at the time of apples harvesting**. These new nodes can therefore take the same states as the old ones.
In this model, there was a dependence between sick and sick, then between autumn and autumn. Indeed, we expect that if the tree is sick at a time t, it affects its state of health at time t + delta (t). This causality will of course depend on the size of this delta (t).
Our gardener can provide solutions to his problem. He can try to treat the tree with a treatment to cure a possible disease. If he thinks that the tree loses his leaves only because of the fall, he may not be able to treat his tree and save the money that would cost him.
So we will add another node to the tree called the decision tree. The R.B. becomes at this stage an influence diagram. The decision nodes are represented by rectangles. This new node will have two states: "yes" and "no" depending on whether the gardener is treating the tree or not. And it will be linked to Disease ', because we expect the treatment to have an influence on the future health of the tree.
Then, to determine this influence diagram, we will add some utility numbers. They will allow us in our case to calculate the cost of decision. They are represented on the figure by trapezes.
The "Cost" node gives information on the cost of treatment, while "Harvest" represents the benefits of the harvest. It depends on disease, indicating that the production of apples depends on the health of the tree. Each utility node contributes to the total usefulness of the diagram.
Then, just like in the BN For each "variable" node, define the conditional probabilities. For each utility node, define the utility table. For each decision node, do not define anything, since it is a phenomenon that is not random, which depends solely on the user.

Page 64 :

Medecine

- ▷ Diagnosis support for cardiovascular problems
- ▷ Transfusion supervision, . . .

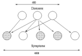


Industrie

- ▷ NASA: real-time fault diagnosis support for the propulsion systems of the Space Shuttle
- ▷ Lockheed Martin : control system of an autonomous underwater vehicle
- ▷ Ricoh : remote diagnostic assistance
- ▷ EDF : generator modeling

Medecine

- ▷ Diagnosis support for cardiovascular problems
- ▷ Transfusion supervision, ...



Industrie

- ▷ NASA: real-time fault diagnosis support for the propulsion systems of the Space Shuttle
- ▷ Lockheed Martin : control system of an autonomous underwater vehicle
- ▷ Ricoh : remote diagnostic assistance
- ▷ EDF : generator modeling

Page 65 :

Software offer

Toolbox

- Bayes Net Toolbox (BNT) for Matlab
- gR, GRAPPA, . . . for R
- BNJ, JavaBayes, . . . for Java

Non-commercial software

- Microsoft Belief Network [US]
- Genie 2/Smile [US]

Logiciels commerciaux

- Bayesia [FR]
- ProBT (inférence probabiliste) [FR]
- Hugin [DK]
- Netica [CA]

Software offer

Toolbox
- ▷ Bayes Net Toolbox (BNT) for Matlab
- ▷ gR, GRAPPA, . . . for R
- ▷ BNJ, JavaBayes, . . . for Java

Non-commercial software
- ▷ Microsoft Belief Network [US]
- ▷ Genie 2/Smile [US]

Logiciels commerciaux
- ▷ Bayesia [FR]
- ▷ ProBT (inférence probabiliste) [FR]
- ▷ Hugin [DK]
- ▷ Netica [CA]

Page 66 :

How to choose a tool?

- Limited offer
- Tasks taken into account?
    - creating a network "graphically"?
    - inference: implemented algorithms?
    - learning: missing data? structure?
- API?

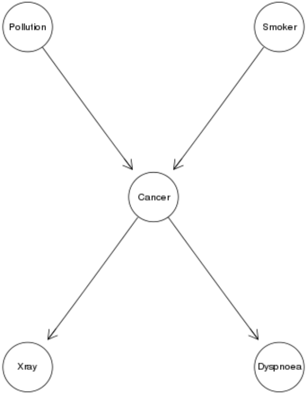
How to choose a tool?

- Limited offer
- Tasks taken into account?
  - creating a network "graphically"?
  - inference: implemented algorithms?
  - learning: missing data? structure?
- API?

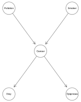
Page 67 :

Page 68 :

Example



Example



Page 69 :

In this example we will try to create the cancer (`http://www.bnlearn.com/bnrepository/#cancer`) bayesian network using pgmpy and do some simple queries on the network.

Create

```
# Starting with defining the network structure
from pgmpy.models import BayesianModel
cancer_model = BayesianModel([('Pollution', 'Cancer'),
                              ('Smoker', 'Cancer'),
                              ('Cancer', 'Xray'),
                              ('Cancer', 'Dyspnoea')])
# Now defining the parameters.
from pgmpy.factors.discrete import TabularCPD
cpd_poll = TabularCPD(variable='Pollution', variable_card=2,
                      values=[[0.9], [0.1]])
cpd_smoke = TabularCPD(variable='Smoker', variable_card=2,
                       values=[[0.3], [0.7]])
cpd_cancer = TabularCPD(variable='Cancer', variable_card=2,
                        values=[[0.03, 0.05, 0.001, 0.02],
                                [0.97, 0.95, 0.999, 0.98]],
                        evidence=['Smoker', 'Pollution'],
                        evidence_card=[2, 2])
cpd_xray = TabularCPD(variable='Xray', variable_card=2,
                      values=[[0.9, 0.2], [0.1, 0.8]],
                      evidence=['Cancer'], evidence_card=[2])
cpd_dysp = TabularCPD(variable='Dyspnoea', variable_card=2,
                      values=[[0.65, 0.3], [0.35, 0.7]],
                      evidence=['Cancer'], evidence_card=[2])
# Associating the parameters with the model structure.
cancer_model.add_cpds(cpd_poll, cpd_smoke, cpd_cancer, cpd_xray, cpd_dysp)
```

Create

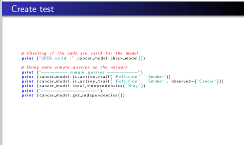
```python
# Starting with defining the network structure
from pgmpy.models import BayesianModel
cancer_model = BayesianModel([('Pollution', 'Cancer'),
                              ('Smoker', 'Cancer'),
                              ('Cancer', 'Xray'),
                              ('Cancer', 'Dyspnoea')])
# Now defining the parameters.
from pgmpy.factors.discrete import TabularCPD
cpd_poll = TabularCPD(variable='Pollution', variable_card=2,
                      values=[[0.9], [0.1]])
cpd_smoke = TabularCPD(variable='Smoker', variable_card=2,
                       values=[[0.3], [0.7]])
cpd_cancer = TabularCPD(variable='Cancer', variable_card=2,
                        values=[[0.03, 0.05, 0.001, 0.02],
                                [0.97, 0.95, 0.999, 0.98]],
                        evidence=['Smoker', 'Pollution'],
                        evidence_card=[2, 2])
cpd_xray = TabularCPD(variable='Xray', variable_card=2,
                      values=[[0.9, 0.2], [0.1, 0.8]],
                      evidence=['Cancer'], evidence_card=[2])
cpd_dysp = TabularCPD(variable='Dyspnoea', variable_card=2,
                      values=[[0.65, 0.3], [0.35, 0.7]],
                      evidence=['Cancer'], evidence_card=[2])
# Associating the parameters with the model structure.
cancer_model.add_cpds(cpd_poll, cpd_smoke, cpd_cancer, cpd_xray, cpd_dysp)
```

Page 70 :

https://github.com/pgmpy/pgmpy

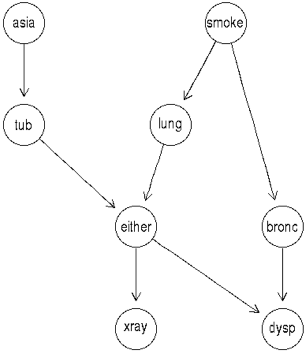
Create test

```
# Checking if the cpds are valid for the model.
print ("CPDS valid :",cancer_model.check_model())

# Doing some simple queries on the network
print ("———— simple queries ————")
print (cancer_model.is_active_trail('Pollution', 'Smoker'))
print (cancer_model.is_active_trail('Pollution', 'Smoker', observed=['Cancer']))
print (cancer_model.local_independencies('Xray'))
print ("————————————")
print (cancer_model.get_independencies())
```

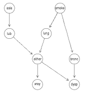



```
CPDS valid : True
———————— simple queries —————————
False
True
(Xray _|_ Dyspnoea, Pollution, Smoker | Cancer)
—————————————————
(Xray _|_ Dyspnoea, Pollution, Smoker | Cancer)
(Xray _|_ Pollution, Smoker | Dyspnoea, Cancer)
(Xray _|_ Dyspnoea, Smoker | Pollution, Cancer)
(Xray _|_ Dyspnoea, Pollution | Cancer, Smoker)
(Xray _|_ Smoker | Dyspnoea, Pollution, Cancer)
(Xray _|_ Pollution | Dyspnoea, Cancer, Smoker)
(Xray _|_ Dyspnoea | Pollution, Cancer, Smoker)
(Dyspnoea _|_ Xray, Pollution, Smoker | Cancer)
(Dyspnoea _|_ Pollution, Smoker | Xray, Cancer)
(Dyspnoea _|_ Xray, Smoker | Pollution, Cancer)
(Dyspnoea _|_ Xray, Pollution | Cancer, Smoker)
(Dyspnoea _|_ Smoker | Xray, Pollution, Cancer)
(Dyspnoea _|_ Pollution | Xray, Cancer, Smoker)
(Dyspnoea _|_ Xray | Pollution, Cancer, Smoker)
(Pollution _|_ Smoker)
(Pollution _|_ Xray, Dyspnoea | Cancer)
(Pollution _|_ Dyspnoea | Xray, Cancer)
(Pollution _|_ Xray | Dyspnoea, Cancer)
(Pollution _|_ Xray, Dyspnoea | Cancer, Smoker)
(Pollution _|_ Dyspnoea | Xray, Cancer, Smoker)
(Pollution _|_ Xray | Dyspnoea, Cancer, Smoker)
(Smoker _|_ Pollution)
```

Inference : Asia



Inference : Asia

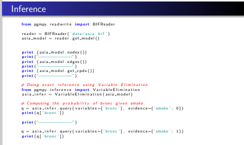


Page 72 :

We will be using the Asia network (`http://www.bnlearn.com/bnrepository/#asia`) and do some inference queries.

Inference

```
from pgmpy.readwrite import BIFReader

reader = BIFReader('data/asia.bif')
asia_model = reader.get_model()


print (asia_model.nodes())
print("--------------")
print (asia_model.edges())
print("--------------")
print (asia_model.get_cpds())
print("--------------")

# Doing exact inference using Variable Elimination
from pgmpy.inference import VariableElimination
asia_infer = VariableElimination(asia_model)

# Computing the probability of bronc given smoke.
q = asia_infer.query(variables=['bronc'], evidence={'smoke': 0})
print(q['bronc'])

print("--------------")

q = asia_infer.query(variables=['bronc'], evidence={'smoke': 1})
print(q['bronc'])
```

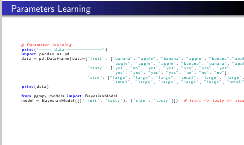


Page 73 :

```
['bronc', 'tub', 'either', 'xray', 'smoke', 'asia', 'lung', 'dysp']
——————————————
[('bronc', 'dysp'), ('tub', 'either'), ('either', 'dysp'), ('either', 'xray'),
('smoke', 'bronc'), ('smoke', 'lung'), ('asia', 'tub'), ('lung', 'either')]
——————————————
[<TabularCPD representing P(asia:2) at 0x7f6d67fecc88>,
<TabularCPD representing P(bronc:2 | smoke:2) at 0x7f6d6630b358>,
<TabularCPD representing P(dysp:2 | bronc:2, either:2) at 0x7f6d6630b3c8>,
<TabularCPD representing P(either:2 | lung:2, tub:2) at 0x7f6d6177b9b0>,
<TabularCPD representing P(lung:2 | smoke:2) at 0x7f6d410a9940>,
<TabularCPD representing P(smoke:2) at 0x7f6d410b4470>,
<TabularCPD representing P(tub:2 | asia:2) at 0x7f6d410b4710>,
<TabularCPD representing P(xray:2 | either:2) at 0x7f6d410b45c0>]
——————————————
+———————————+————————————————+
| bronc     |     phi(bronc) |
+===========+================+
| bronc_0   |         0.6000 |
+———————————+————————————————+
| bronc_1   |         0.4000 |
+———————————+————————————————+
——————————————
+———————————+————————————————+
| bronc     |     phi(bronc) |
+===========+================+
| bronc_0   |         0.3000 |
```

Parameters Learning

```
# Parameter learning
print("——— Data ———————————")
import pandas as pd
data = pd.DataFrame(data={'fruit': ["banana", "apple", "banana", "apple", "banana","appl
                                    "apple", "apple", "apple", "banana", "banana", "appl
                          'tasty': ["yes", "no", "yes", "yes", "yes", "yes", "yes",
                                    "yes", "yes", "yes", "yes", "no", "no", "no"],
                          'size': ["large", "large", "large", "small", "large", "large",
                                    "small", "large", "large", "large", "large", "small"
print(data)

from pgmpy.models import BayesianModel
model = BayesianModel([('fruit', 'tasty'), ('size', 'tasty')])  # fruit -> tasty <- size
```



Page 74 :

```
——— Data ————————
      fruit   size  tasty
0    banana  large    yes
1     apple  large     no
2    banana  large    yes
3     apple  small    yes
4    banana  large    yes
5     apple  large    yes
6    banana  large    yes
7     apple  small    yes
8     apple  large    yes
9     apple  large    yes
10   banana  large    yes
11   banana  large     no
12    apple  small     no
13   banana  small     no
```

Parameters Learning

```
# State counts
print("---- State counts ----------")
from pgmpy.estimators import ParameterEstimator
pe = ParameterEstimator(model, data)
print("\n", pe.state_counts('fruit'))  # unconditional
print("----------------")
print("\n", pe.state_counts('tasty'))  # conditional on fruit and size
print("----------------")
```

Parameters Learning

```python
# State counts
print("------ State counts ------------------")
from pgmpy.estimators import ParameterEstimator
pe = ParameterEstimator(model, data)
print("\n", pe.state_counts('fruit'))  # unconditional
print("--------------------------")
print("\n", pe.state_counts('tasty'))  # conditional on fruit and size
print("--------------------------")
```

Page 75 :

```
------ State counts ------------------

         fruit
apple        7
banana       7
--------------------------

 fruit  apple        banana
size    large  small  large  small
tasty
no        1.0    1.0    1.0    1.0
yes       3.0    2.0    5.0    0.0
--------------------------
```

Parameters Learning

```
# Maximum Likelihood Estimation
print("——— Maximum Likelihood Estimation ———————")
from pgmpy.estimators import MaximumLikelihoodEstimator
mle = MaximumLikelihoodEstimator(model, data)
print(mle.estimate_cpd('fruit'))  # unconditional
print("——————————")
print(mle.estimate_cpd('tasty'))  # conditional
print("——————————")
```

Parameters Learning

```python
# Maximum Likelihood Estimation
print("—— Maximum Likelihood Estimation ————————")
from pgmpy.estimators import MaximumLikelihoodEstimator
mle = MaximumLikelihoodEstimator(model, data)
print(mle.estimate_cpd('fruit'))  # unconditional
print("————————————")
print(mle.estimate_cpd('tasty'))  # conditional
print("————————————")
```

Page 76 :

——— Maximum Likelihood Estimation ————————

fruit(apple)	0.5
fruit(banana)	0.5

————————————

fruit	fruit(apple)	fruit(apple)	fruit(banana)	fruit(banana)
size	size(large)	size(small)	size(large)	size(small)
tasty(no)	0.25	0.3333333333333333	0.1666666666666666	1.0
tasty(yes)	0.75	0.6666666666666666	0.8333333333333334	0.0

————————————

Parameters Learning

```
# Calibrate all CPDs of `model` using MLE:
model.fit(data, estimator=MaximumLikelihoodEstimator)

#Bayesian Parameter Estimation
print("------ Bayesian Parameter Estimation ----------------")

from pgmpy.estimators import BayesianEstimator
est = BayesianEstimator(model, data)
print(est.estimate_cpd('tasty', prior_type='BDeu', equivalent_sample_size=10))
print("----------------------")
```

Parameters Learning

```python
# Calibrate all CPDs of 'model' using MLE:
model.fit(data, estimator=MaximumLikelihoodEstimator)

#Bayesian Parameter Estimation
print("——— Bayesian Parameter Estimation —————————")

from pgmpy.estimators import BayesianEstimator
est = BayesianEstimator(model, data)
print(est.estimate_cpd('tasty', prior_type='BDeu', equivalent_sample_size=10))
print("——————————————————")
```

Page 77 :

```
——— Bayesian Parameter Estimation —————————
+------------+--------------------+--------------------+--------------------+----------
| fruit      | fruit(apple)       | fruit(apple)       | fruit(banana)      
| fruit(banana)         |
+------------+--------------------+--------------------+--------------------+----------
| size       | size(large)        | size(small)        | size(large)        
| size(small)           |
+------------+--------------------+--------------------+--------------------+----------
| tasty(no)  | 0.34615384615384615 | 0.4090909090909091 | 0.2647058823529412 | 0.6428571
|
+------------+--------------------+--------------------+--------------------+----------
| tasty(yes) | 0.6538461538461539 | 0.5909090909090909 | 0.7352941176470589 | 0.3571428
+------------+--------------------+--------------------+--------------------+----------
——————————————————
```

Parameters Learning

```
# BayesianEstimator, too, can be used via the fit()-method. Full example:
import numpy as np
import pandas as pd
from pgmpy.models import BayesianModel
from pgmpy.estimators import BayesianEstimator

# generate data
data = pd.DataFrame(np.random.randint(low=0, high=2, size=(5000, 4)), columns=['A', 'B',
model = BayesianModel([('A', 'B'), ('A', 'C'), ('D', 'C'), ('B', 'D')])
model.fit(data, estimator=BayesianEstimator, prior_type="BDeu") # default equivalent_san
for cpd in model.get_cpds():
    print(cpd)
```

Parameters Learning

```python
# BayesianEstimator, too, can be used via the fit()-method. Full example:
import numpy as np
import pandas as pd
from pgmpy.models import BayesianModel
from pgmpy.estimators import BayesianEstimator

# generate data
data = pd.DataFrame(np.random.randint(low=0, high=2, size=(5000, 4)), columns=['A', 'B',
model = BayesianModel([('A', 'B'), ('A', 'C'), ('D', 'C'), ('B', 'D')])
model.fit(data, estimator=BayesianEstimator, prior_type="BDeu") # default equivalent_san
for cpd in model.get_cpds():
    print(cpd)
```

Page 78 :

```
+------+--------------------+--------------------+--------------------+------------------
| A    | A(0)               | A(0)               | A(1)
| A(1)                 |
+------+--------------------+--------------------+--------------------+------------------
| D    | D(0)               | D(1)               | D(0)
| D(1)                 |
+------+--------------------+--------------------+--------------------+------------------
| C(0) | 0.5041912745284817 | 0.5165023143489635 | 0.5082791247782378 | 0.48096849228166
+------+--------------------+--------------------+--------------------+------------------
| C(1) | 0.4958087254715184 | 0.4834976856510364 | 0.4917208752217623 | 0.51903150771833
+------+--------------------+--------------------+--------------------+------------------
+------+--------------------+--------------------+
| B    | B(0)               | B(1)               |
+------+--------------------+--------------------+
| D(0) | 0.5008102086287219 | 0.5299625468164794 |
+------+--------------------+--------------------+
| D(1) | 0.4991897913712781 | 0.4700374531835206 |
+------+--------------------+--------------------+
+------+---------------------+--------------------+
| A    | A(0)                | A(1)               |
+------+---------------------+--------------------+
| B(0) | 0.49373654335486394 | 0.4926545602938176 |
+------+---------------------+--------------------+
| B(1) | 0.5062634566451361  | 0.5073454397061824 |
```

Structure Learning

```python
# Scoring functions
print("-- Scoring functions ------")
import pandas as pd
import numpy as np
from pgmpy.estimators import BdeuScore, K2Score, BicScore
from pgmpy.models import BayesianModel
# create random data sample with 3 variables, where Z is dependent on X, Y:
data = pd.DataFrame(np.random.randint(0, 4, size=(5000, 2)), columns=list('XY'))
data['Z'] = data['X'] + data['Y']

bdeu = BdeuScore(data, equivalent_sample_size=5)
k2 = K2Score(data)
bic = BicScore(data)

model1 = BayesianModel([('X', 'Z'), ('Y', 'Z')])  # X -> Z <- Y
model2 = BayesianModel([('X', 'Z'), ('X', 'Y')])  # Y <- X -> Z

print(bdeu.score(model1))
print("-----------")
print(k2.score(model1))
print("-----------")
print(bic.score(model1))
print("-----------")
```

Structure Learning

```
# Scoring functions
print("——  Scoring functions ————")
import pandas as pd
import numpy as np
from pgmpy.estimators import BdeuScore, K2Score, BicScore
from pgmpy.models import BayesianModel
# create random data sample with 3 variables, where Z is dependent on X, Y:
data = pd.DataFrame(np.random.randint(0, 4, size=(5000, 2)), columns=list('XY'))
data['Z'] = data['X'] + data['Y']

bdeu = BdeuScore(data, equivalent_sample_size=5)
k2 = K2Score(data)
bic = BicScore(data)

model1 = BayesianModel([('X', 'Z'), ('Y', 'Z')])  # X -> Z <- Y
model2 = BayesianModel([('X', 'Z'), ('X', 'Y')])  # Y <- X -> Z

print(bdeu.score(model1))
print("————————")
print(k2.score(model1))
print("————————")
print(bic.score(model1))
print("————————")
```

Page 79 :

```
——  Scoring functions ————
−13937.643714269507
————————
−14328.379219715654
————————
−14293.685763707872
————————
```

Structure Learning

```
print(bdeu.score(model2))
print("————————")
print(k2.score(model2))
print("————————")
print(bic.score(model2))
print("————————")

print(bdeu.local_score('Z', parents=[]))
print("————————")
print(bdeu.local_score('Z', parents=['X']))
print("————————")
print(bdeu.local_score('Z', parents=['X', 'Y']))
print("————————")
```

Structure Learning

```python
print(bdeu.score(model2))
print("————————")
print(k2.score(model2))
print("————————")
print(bic.score(model2))
print("————————")

print(bdeu.local_score('Z', parents=[]))
print("————————")
print(bdeu.local_score('Z', parents=['X']))
print("————————")
print(bdeu.local_score('Z', parents=['X', 'Y']))
print("————————")
```

Page 80 :

-20896.95812733213

————————

-20923.79732546166

————————

-20941.01008687303

————————

-9261.147916770107

————————

-6990.071623800283

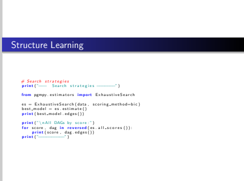
————————

-57.11667730492627

————————

Structure Learning

```
# Search strategies
print("--  Search strategies -------")

from pgmpy.estimators import ExhaustiveSearch

es = ExhaustiveSearch(data, scoring_method=bic)
best_model = es.estimate()
print(best_model.edges())

print("\nAll DAGs by score:")
for score, dag in reversed(es.all_scores()):
    print(score, dag.edges())
print("--------------")
```



Page 81 :

```
——  Search strategies ————
[('Y', 'Z'), ('X', 'Z')]

All DAGs by score:
-14293.685763707872 [('Y', 'Z'), ('X', 'Z')]
-14324.148081608948 [('Y', 'Z'), ('Y', 'X'), ('Z', 'X')]
-14324.148081608948 [('Y', 'Z'), ('Y', 'X'), ('X', 'Z')]
-14324.148081608948 [('X', 'Z'), ('X', 'Y'), ('Z', 'Y')]
-14324.148081608948 [('Y', 'Z'), ('X', 'Z'), ('X', 'Y')]
-14324.14808160895 [('Y', 'X'), ('Z', 'Y'), ('Z', 'X')]
-14324.14808160895 [('X', 'Y'), ('Z', 'Y'), ('Z', 'X')]
-16563.51091980794 [('X', 'Y'), ('Z', 'Y')]
-16566.894032093027 [('Y', 'X'), ('Z', 'X')]
-18667.801818487875 [('Z', 'Y'), ('Z', 'X')]
-18667.801818487875 [('Y', 'Z'), ('Z', 'X')]
-18667.801818487875 [('X', 'Z'), ('Z', 'Y')]
-20907.16465668687 [('Z', 'Y')]
-20907.16465668687 [('Y', 'Z')]
-20910.547768971956 [('Z', 'X')]
-20910.547768971956 [('X', 'Z')]
-20937.626974587944 [('Y', 'Z'), ('Y', 'X')]
-20937.626974587944 [('Y', 'Z'), ('X', 'Y')]
-20937.626974587947 [('Y', 'X'), ('Z', 'Y')]
-20941.01008687303 [('Y', 'X'), ('X', 'Z')]
-20941.01008687303 [('X', 'Y'), ('Z', 'X')]
-20941.01008687303 [('X', 'Z'), ('X', 'Y')]
```

Structure Learning

```
from pgmpy.estimators import HillClimbSearch
# create some data with dependencies
data = pd.DataFrame(np.random.randint(0, 3, size=(2500, 8)), columns=list('ABCDEFGH'))
data['A'] += data['B'] + data['C']
data['H'] = data['G'] - data['A']
hc = HillClimbSearch(data, scoring_method=BicScore(data))
best_model = hc.estimate()
print(best_model.edges())
print("————————————")
```

```python
from pgmpy.estimators import HillClimbSearch
# create some data with dependencies
data = pd.DataFrame(np.random.randint(0, 3, size=(2500, 8)), columns=list('ABCDEFGH'))
data['A'] += data['B'] + data['C']
data['H'] = data['G'] - data['A']
hc = HillClimbSearch(data, scoring_method=BicScore(data))
best_model = hc.estimate()
print(best_model.edges())
print("-----------")
```

Page 82 :

```
[('A', 'B'), ('A', 'C'), ('A', 'H'), ('G', 'H'), ('B', 'C')]
------------
```

Structure Learning

```
# Constraint-based Structure Learning
print("--  Constraint-based Structure Learning --------")
from pgmpy.estimators import ConstraintBasedEstimator

# (Conditional) Independence Tests
data = pd.DataFrame(np.random.randint(0, 3, size=(2500, 8)), columns=list('ABCDEFGH'))
data['A'] += data['B'] + data['C']
data['H'] = data['G'] - data['A']
data['E'] *= data['F']

est = ConstraintBasedEstimator(data)
print(est.test_conditional_independence('B', 'H'))          # dependent
print("--------------")
print(est.test_conditional_independence('B', 'E'))          # independent
print("--------------")
print(est.test_conditional_independence('B', 'H', ['A']))   # independent
print("--------------")
print(est.test_conditional_independence('A', 'G'))          # independent
print("--------------")
print(est.test_conditional_independence('A', 'G',  ['H']))  # dependent
print("--------------")
```

Structure Learning

```
# Constraint-based Structure Learning
print("--  Constraint-based Structure Learning ------")
from pgmpy.estimators import ConstraintBasedEstimator

# (Conditional) Independence Tests
data = pd.DataFrame(np.random.randint(0, 3, size=(2500, 8)), columns=list('ABCDEFGH'))
data['A'] += data['B'] + data['C']
data['H'] = data['G'] - data['A']
data['E'] *= data['F']

est = ConstraintBasedEstimator(data)
print(est.test_conditional_independence('B', 'H'))          # dependent
print("----------")
print(est.test_conditional_independence('B', 'E'))          # independent
print("----------")
print(est.test_conditional_independence('B', 'H', ['A']))   # independent
print("----------")
print(est.test_conditional_independence('A', 'G'))          # independent
print("----------")
print(est.test_conditional_independence('A', 'G',  ['H']))  # dependent
print("----------")
```

Page 83 :

```
--  Constraint-based Structure Learning ------
(710.0636405024474, 5.615634053227994e-133, True)
----------
(3.7273260674929154, 0.9772254435634241, True)
----------
(12.51036855418668, 0.999999231909619, True)
----------
(9.951958325687539, 0.969034725679817, True)
----------
(4621.0, 0.0, True)
----------
```

Structure Learning

```python
def is_independent(X, Y, Zs=[], significance_level=0.05):
    return est.test_conditional_independence(X, Y, Zs)[1] >= significance_level

print(is_independent('B', 'H'))
print("————————————")
print(is_independent('B', 'E'))
print("————————————")
print(is_independent('B', 'H', ['A']))
print("————————————")
print(is_independent('A', 'G'))
print("————————————")
print(is_independent('A', 'G', ['H']))
```

Structure Learning

```python
def is_independent(X, Y, Zs=[], significance_level=0.05):
    return est.test_conditional_independence(X, Y, Zs)[1] >= significance_level

print(is_independent('B', 'H'))
print("----------")
print(is_independent('B', 'E'))
print("----------")
print(is_independent('B', 'H', ['A']))
print("----------")
print(is_independent('A', 'G'))
print("----------")
print(is_independent('A', 'G', ['H']))
```

Page 84 :

```
False
----------
True
----------
True
----------
True
----------
False
```

Structure Learning

```
# DAG (pattern) construction
print("-- DAG (pattern) construction ---------")

skel, seperating_sets = est.estimate_skeleton(significance_level=0.01)
print("Undirected edges: ", skel.edges())
print("----------------")

pdag = est.skeleton_to_pdag(skel, seperating_sets)
print("PDAG edges:         ", pdag.edges())
print("----------------")

model = est.pdag_to_dag(pdag)
print("DAG edges:          ", model.edges())
print("----------------")

print(est.estimate(significance_level=0.01).edges())
print("----------------")

from pgmpy.independencies import Independencies
ind = Independencies(['B', 'C'],
                     ['A', ['B', 'C'], 'D'])
ind = ind.closure()  # required (!) for faithfulness
model = ConstraintBasedEstimator.estimate_from_independencies("ABCD", ind)
print(model.edges())
```

Structure Learning

```python
# DAG (pattern) construction
print("—— DAG (pattern) construction ————")

skel, seperating_sets = est.estimate_skeleton(significance_level=0.01)
print("Undirected edges: ", skel.edges())
print("—————————")

pdag = est.skeleton_to_pdag(skel, seperating_sets)
print("PDAG edges:        ", pdag.edges())
print("—————————")

model = est.pdag_to_dag(pdag)
print("DAG edges:         ", model.edges())
print("—————————")

print(est.estimate(significance_level=0.01).edges())
print("—————————")

from pgmpy.independencies import Independencies
ind = Independencies(['B', 'C'],
                     ['A', ['B', 'C'], 'D'])
ind = ind.closure()  # required (!) for faithfulness
model = ConstraintBasedEstimator.estimate_from_independencies("ABCD", ind)
print(model.edges())
```

Page 85 :

```
—— DAG (pattern) construction ————
Undirected edges: [('C', 'A'), ('E', 'F'), ('A', 'B'), ('A', 'H'), ('G', 'H')]
—————————
PDAG edges:       [('C', 'A'), ('E', 'F'), ('A', 'H'), ('F', 'E'), ('G', 'H'), ('B', 'A
—————————
DAG edges:        [('C', 'A'), ('A', 'H'), ('F', 'E'), ('G', 'H'), ('B', 'A')]
—————————
[('C', 'A'), ('A', 'H'), ('F', 'E'), ('G', 'H'), ('B', 'A')]
—————————
[('B', 'D'), ('C', 'D'), ('A', 'D')]
```

Bayesian Networks

Cédric Buche

ENIB

October 21, 2019