Introduction to graphical models :

Bayesian Networks and Hidden Markov Models
Part I

June 2018

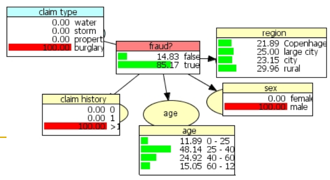
Laurence Likforman-Sulem
IMT/Telecom ParisTech/University Paris-Saclay
likforman@telecom-paristech.fr

Overview

- Part I : Graphical Models
  - Bayesian Networks
  - Dynamic Bayesian Networks (DBN)
  - link with HMMs (Hidden Markov Models)
- Part II : Hidden Markov Models
  - discretes, continuous
  - generative models
  - décoding, training

applications

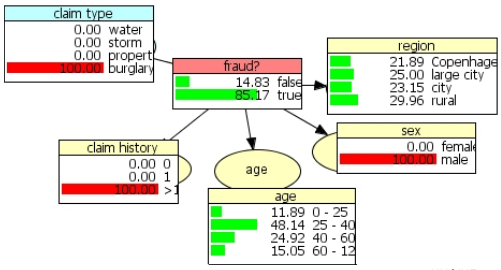
- HMMs and Dynamic Bayesian Networks (DBN)
  - Speech recognition
  - Handwriting recognition
  - Recognition of objects, faces in videos,…
  - Natural Language processing
- Static Bayesian networks
  - Fraud detection (ex: HUGIN)
  - diagnosis



observed variable

Application: fraud detection

From observation variable(s), infer values of remaining variables



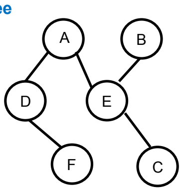
observations

source: HUGINExpert

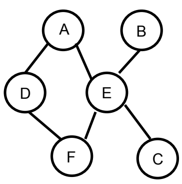
Graphical Models : undirected graphs

- tree : single path between 2 nodes
- multi-connected graph : several paths between nodes

Tree



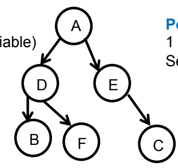
multi-connected graph



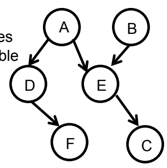
Graphical Models : directed graphs

- Relations : sons, siblings, parents, descendants, ancestors

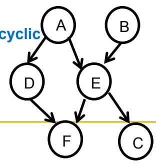
Simple Tree
(1 parent/variable)



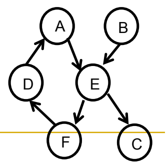
Poly-tree
1 path between 2 nodes
Several parents /variable



multi-connected acyclic Graph



Cyclic Graph

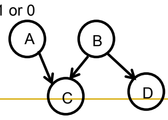


Bayesian network: definition

- Bayesian network: (G, θ)
- G=(V, E) directed acyclic graph (DAG)
  - V : variables (nodes),
  - E : edges: relations between variables (influence of one variable over another)
- θ : parameters (conditional probability tables)
  - Ex: P(X | parents(X)) or P(X) if X has no parent
- Graph : factories joint probability of all variables
  - training: less parameters 1+1+4+2=8 instead of 2^4-1=15
  - Inference : computational complexity reduced

A=1 or 0

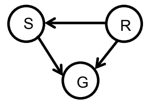
$$P(A,B,C,D) = P(A)P(B)P(C|A,B)P(D|B)$$



Bayesian network: example

[from Wikipedia BN]

- G: grass is wet (1 ou 0)
- S: sprinkler on (1 ou 0)
- R: rain during past night (1 ou 0)



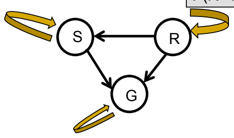
- S→ G : G is a consequence of S
- R→ S: R influences whether the sprinkler is on.
  (if it has rained, you do not need to open the sprinkler)
- R→ G: R influences whether the grass is wet
- Conversely: knowing the value of G «modifies the belief » in S and R : P(R=1|G=1) > P(R=1))

Bayesian network: parameters

- Conditional Probability Tables or Distributions (CPT and CPD) : P(X| parents(X))

- G: grass wet (1 or 0)
- S: sprinkler on (1 or 0)
- R: rain during night (1 or 0)

P(R=0)=0.8
P(R=1)=0.2



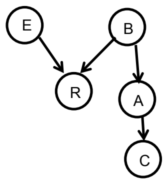
R	P(S=1|R)	P(S=0| R)
0	0.4	0.6
1	0.01	0.99

S	R	P(G=0| S,R)	P(G=1| S,R)
1	1	0.01	0.99
1	0	0.1	0.9
0	1	0.2	0.8
0	0	1.0	0.0

Bayesian network: conditional independence

- A node is independent of its non-descendants given its parents

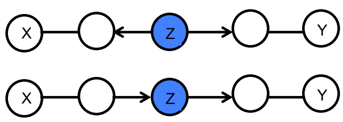
- Graph: binary variables A, B, C, E, R
- C is independent of R, B, E given A



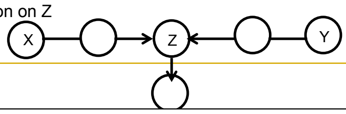
[from Nir Friedman]

conditional independence : d-separation

- A node X is independent from a node Y given a set of observed variables E (E d-separates X and Y)
- if all non directed paths between X and Y are blocked by a variable Z such as :
- Z is in E and there is a chain or divergent connexion on Z

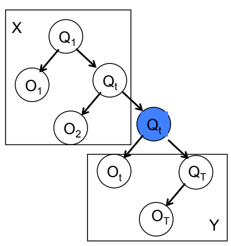


- Z is not observed, nor its descendants, and there is a convergent connexion on Z



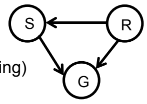
d-separation : set of nodes

- set of nodes X and set of nodes Y are d-separated given observed node Qt



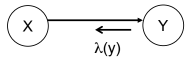
Bayesian network: inference

- computing P(variable(s) | observed variable(s))
- observed variables : « evidence » or observation
- Inference from observations
- inférence algorithms
  - Exact: algorithms trees, polytrees (message passing)
  - stochastic : sampling



Inference in chains: backward variable

- 2 nodes (1 observed)



$E=\{Y=y_2\}$

- Parameters : P(X=x) with dom{X}={x1,x2}, and $P_{Y|X}$ with dom{Y}={y1,y2, y3},

$$P_{Y|X}=\begin{bmatrix} P(Y=y_1|X=x_1) & P(Y=y_2|X=x_1) & P(Y=y_3|X=x_1) \\ P(Y=y_1|X=x_2) & P(Y=y_2|X=x_2) & P(Y=y_3|X=x_2) \end{bmatrix}$$

- backward variable λ:

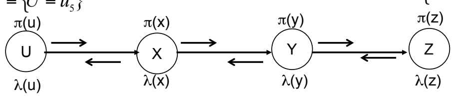
$$\lambda(y)=P(E|Y=y)=[\,0 \quad 1 \quad 0\,]^T$$
$$\lambda(x)=P(E|X=x)$$
$$\lambda(x)=P_{Y|X}\bullet\lambda(y)=\begin{bmatrix} P(Y=y_2|X=x_1) \\ P(Y=y_2|X=x_2) \end{bmatrix}$$

λ(x) computed from « message » λ(y) sent by Y to X

$$P(X,E)=P(E=e|X=x)P(X=x)=\lambda(x)P(X=x)$$

inference in chains

$E^+=\{U=u_5\}$ $E^-=\{Z=z_4\}$



4 nodes (2 are observed) (Z and U), observations E^+ and E^-:
dom{Z}={z_1, z_2, z_3, z_4}
dom{U}={u_1, u_2, u_3, u_4 , u_5}

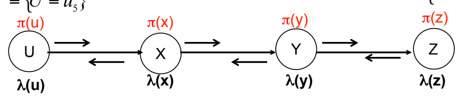
backward variable $\lambda(z)=P(E^-|Z=z)=[\,0 \quad 0 \quad 0 \quad 1\,]^T$

$$\lambda(y)=P_{Z|Y}\bullet\lambda(z)$$
$$\text{and}\quad \lambda(x)=P_{Y|X}\bullet\lambda(y)$$

λ(y) computed from « message » λ(z) sent to Y by Z
Then λ(x) computed from « message » λ(y) sent to X by Y

inference in chains : forward variable

$E^+=\{U=u_5\}$ $E^-=\{Z=z_4\}$



4 nodes (2 observed)

forward variable $\pi(x)=P(X=x,E^+)$

π(x) computed from message π(u) sent by U to child X

$$\pi(u)=[\,0 \quad 0 \quad 0 \quad 0 \quad 1]$$
$$\pi(x)=\pi(u)\bullet P_{X|U}$$
$$\pi(y)=\pi(x)\bullet P_{Y|X}$$

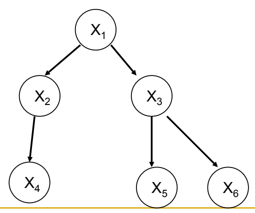
$$\boxed{\begin{aligned} P(X=x,E)&=\lambda(x)\pi(x) \\ P(X=x|E)&\propto\lambda(x)\pi(x) \end{aligned}}$$

normalisation factor: 1/P(E)

$$P(E)=\sum_{x\in dom\{X\}}\lambda(x)\pi(x)$$

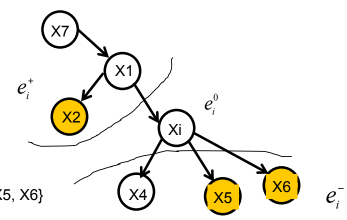
inference in trees

- observed variables :X_1=2, X_4=1, X_5=1, X_6=2
- hidden variables (states) : X_2, dom{X_2}={3,4,5} X_3, dom{X_3}={6,7}



inference in trees

- e_i^- : observed variables in subtrees rooted in Xi's children
- e_i^0 : observed value of Xi (if Xi observed)
- e_i^+: all other, observed variables
- E={e_i^0, e_i^+, e_i^- }



observations: E={X2, X5, X6}

inférence in trees

- for node Xi (hidden)
  - variable λ (backward) $\lambda(x_i) = P(e_i^0, e_i^- | X_i = x_i)$
  - variable π forward $\pi(x_i) = P(e_i^+, X_i = x_i)$

$$P(E, X_i = x_i) = P(e_i^0, e_i^-, e_i^+, X_i = x_i)$$

$$\boxed{P(X_i = x_i, E) = \lambda(x_i)\pi(x_i)}$$

$$P(E) = \sum_{x_i \in dom\{X_i\}} \lambda(x_i)\pi(x_i)$$

$$P(X_i = x_i | E) = \frac{\lambda(x_i)\pi(x_i)}{\sum_{x_i \in dom\{X_i\}} \lambda(x_i)\pi(x_i)} \propto \lambda(x_i)\pi(x_i)$$

Message passing inference algorithm

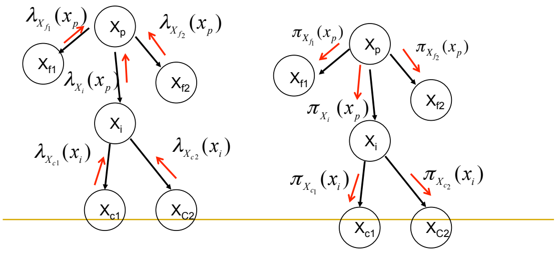
message λ from Xi's children to Xi: $\lambda_{X_{c1}}(x_i)$

message λ from Xi à to single parent Xp: $\lambda_{X_i}(x_p)$

message λ from Xi's siblings to parent Xp: $\lambda_{X_{f2}}(x_p)$

message π from parent Xp to child Xi $\pi_{X_i}(x_p)$

message π sent by Xi to its children $\pi_{X_{c1}}(x_i)$



Inference algorithm : computing λ

- from leaves to root
  - children X_{ci} send message to parent Xi.
    $\lambda_{X_{c1}}(x_i) = P_{X_{c1}|Xi} \bullet \lambda(x_{c1})$
  - IF X_i leaf node observed $\lambda(x_i) = [0 \quad 1 \quad 0]$ (1 at observatio position)
  - IF X_i leaf node non observed: $\lambda(x_i) = [1 \quad 1 \quad 1]$
  - if X_i hidden node (not leaf) : combine children λ messages
    $$\lambda(x_i) = \prod_{Y:children(Xi)} \lambda_Y(x_i) \qquad \lambda(x_i) = \lambda_{X_{c1}}(x_i)\, \lambda_{X_{c2}}(x_i)$$
  - si X_i observed node (not leaf) : $\lambda(x_i) = \prod_{Y:children(Xi)} \lambda_Y(x_i)$, for xi=corresponding to observation, else 0

Inference algorithm : computing π

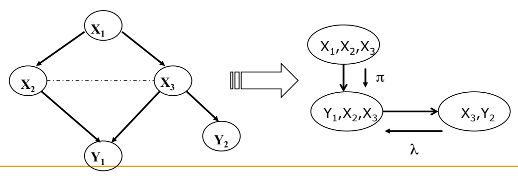
- from root to leaves
- X_i observed $\pi(x_i) = [0 \quad 1 \quad 0]$
- X_i root, non observed $\pi(x_i) = P(X_i = x_i)$
  - else: use message π : $\pi_{Xi}(x_p)$ sent to Xi by unique parent Xp:
    $$\pi_{X_i}(x_p) = \pi(x_p) \prod_{Z,\text{ siblings of } Xi} \lambda_Z(x_p)$$
  - compute $\pi(x_i) = \pi_{Xi}(x_p) P_{Xi|Xp}$
  - Xi sends messages π to each child

(combining messages λ of siblings of X_i, to parent X_p and π of X_i as a parent)

$$\pi_{X_{c1}}(x_i) = \prod_{X_f \text{ siblings of } X_{ci}} \lambda_{X_f}(x_i)\, \pi(x_i) = \prod_{X_f} \lambda_{X_f}(x_i)\, \pi_{X_i}(x_p)\, P_{X_i|X_p}$$

inference in a DAG

- convert graph into tree structure (junction tree)
- inference algorithm : junction tree [Jensen, 96] [Zweig, 2003]
  - moralisation (connect parents)
  - triangulation (cliques construction)
  - connect cliques

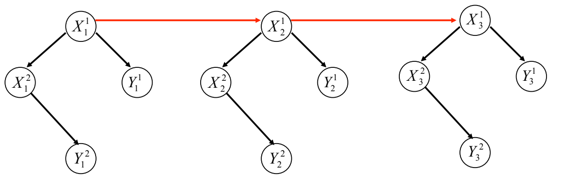


training Bayesian networks

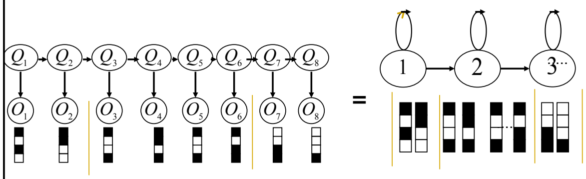
- training with complete data, known structure
  - parameter estimation $P(variable | Parents)$
  - maximum likelihood estimation
- training with incomplete data, known structure
  - EM algorithm or gradient descent or stochastic approaches MCMC (Gibbs sampling)
- training with complete data, unknown structure
  - greedy algorithms
- training with uncomplete data, unknown structure
  - EM + greedy algorithms

Dynamic Bayesian networks (DBN)

- Extension of static Bayesian networks to dynamic networks
- Stationary processes
  - structure & parameters do not vary through time
  - several state variables and observation variables at each time step.



HMM= special case of DBN



- HMM: Hidden Markov Model
- DBN: tree
- 1 state variable + 1 observation variable at each time step t

$(Q_t)_{1 \le t \le T}$: state variable (hidden)

$(O_t)_{1 \le t \le T}$: observation variable generated by state variable

références

- Wikipedia BN: http://en.wikipedia.org/wiki/Bayesian_network
- A. W. Moore, *Bayes Nets for representing and reasoning about uncertainty,* 2001
  - http://www.cs.cmu.edu/%7Eawm/tutorials
- S. Davis, A. Moore Bayesian networks: independencies and inference
  - http://www.cs.cmu.edu/~awm/tutorials
- K. Murphy, BayesNet Toolbox for matlab https://code.google.com/p/bnt/
- N. Friedman, D. Koller, Learning Bayesian Networks from data
- G. Zweig, Speech Recognition with Dynamic Bayesian Networks, Phd thesis, 1998, Univ. of California, Berkeley,
- P. Leray, Réseaux Bayésiens-Apprentissage de la structure
  - http://asi.insa-rouen.fr/enseignants/~pleray/RB2003/2-ApprentissageStructure.pdf
- P. Naïm , P-H. Wuillemin , P. Leray , O. Pourret , A. Becker, Les réseaux Bayésiens, Eyrolles, 2007.
- M. Sigelle, Bases de la Reconnaissance des Formes: Chaînes de Markov et Modèles de Markov Cachés, chapitre 7, Polycopié Telecom ParisTech, 2012.
- J. Pearl, Probabilistic Reasoning in intelligent systems: networks of plausible inference networks, 1988.
- L. Likforman-Sulem, E. Barney Smith, Reconnaissance des Formes: théorie et pratique sous matlab, Ellipses, TechnoSup, 2013.