REDES BAYESIANAS: APRENDIZAJE, INFERENCIA Y APLICACIONES

Concha Bielza

Computational Intelligence Group
Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid

Escuela de Verano de Inteligencia Artificial (EVIA 2016)

17 de junio de 2016

Outline

- BASICS
- INFERENCE
- LEARNING

Basics of Bayesian networks

- Conditional independence
- u-separation
- Bayesian networks: formal definition

Reasoning under uncertainty

Advantages of BNs

- Explicit representation of the uncertain knowledge
  - Graphical, intuitive, closer to a world repres.
- Deal with uncertainty for reasoning *and* decision-making
- Founded on probability theory, provide a clear semantics and a sound theoretical foundation
- Manage many variables
- Both data and experts can be used to construct the model
- Current and huge development
- Support the expert; do not try to replace him

Conditional independence

Modularity

- The joint probability distribution (JPD) (global model) is specified via marginal and conditional distributions (local models), taking into account conditional independence relationships among variables

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | X_1, \ldots, X_{i-1})$$ *(chain rule)*

2^n-1 parameters (complete dependence)

some conditional independencies

n parameters (mutual independence)

Conditional independence

Independence and conditional independence

- Independence (marginal) $P(x, y) = P(x)P(y)$ ⟷ $P(x|y) = P(x)$

  sets of vars

- Conditional independence of *X* and *Y* given *Z*

  3 disjoint sets of variables

$$P(x|y, z) = P(x|z)$$

for all possible values x,y,z

Intuitively, whenever *Z=z*, the information *Y=y* does not influence on the probability of *x*

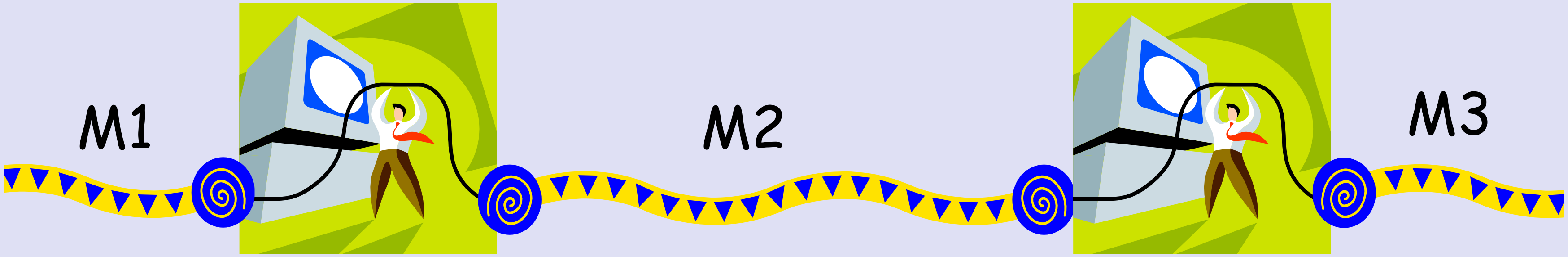
Notation: $I_P(X, Y|Z)$

Conditional independence

Example

- Send a message M1 through a transmitter. It is received as M2 and it is then sent through other transmitter. It is received finally as M3.

  Transmitters have noise that modifies messages



➡ M1 and M3 are dependent without any knowledge $\neg I_P(M1, M3|\emptyset)$

➡ M1 and M3 are independent given M2 $I_P(M1, M3|M2)$

Further factorizing the JPD

Factorization via c.i.

- About $P(X_i|X_1, \ldots, X_{i-1})$:
  - Domain knowledge usually allows one to identify a subset $pa(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$ such that
    - Given $pa(X_i)$, X_i is independent of all variables in $\{X_1, \ldots, X_{i-1}\} \setminus pa(X_i)$, i.e.

$$P(X_i|X_1, \ldots, X_{i-1}) = P(X_i|pa(X_i))$$

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|pa(X_i))$$

Joint distribution factorized

- The number of parameters might be substantially reduced

Bayesian Networks

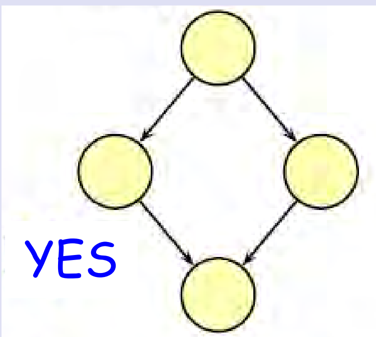
Informal definition: two components

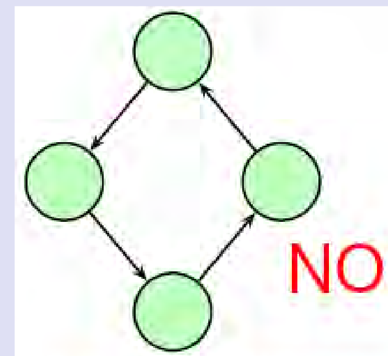
- Qualitative part: a directed acyclic graph (DAG)

  Nodes = variables

  Arcs = direct dependence relations (otherwise it indicates absence of direct dependence; there may be indirect dependencies and independencies)

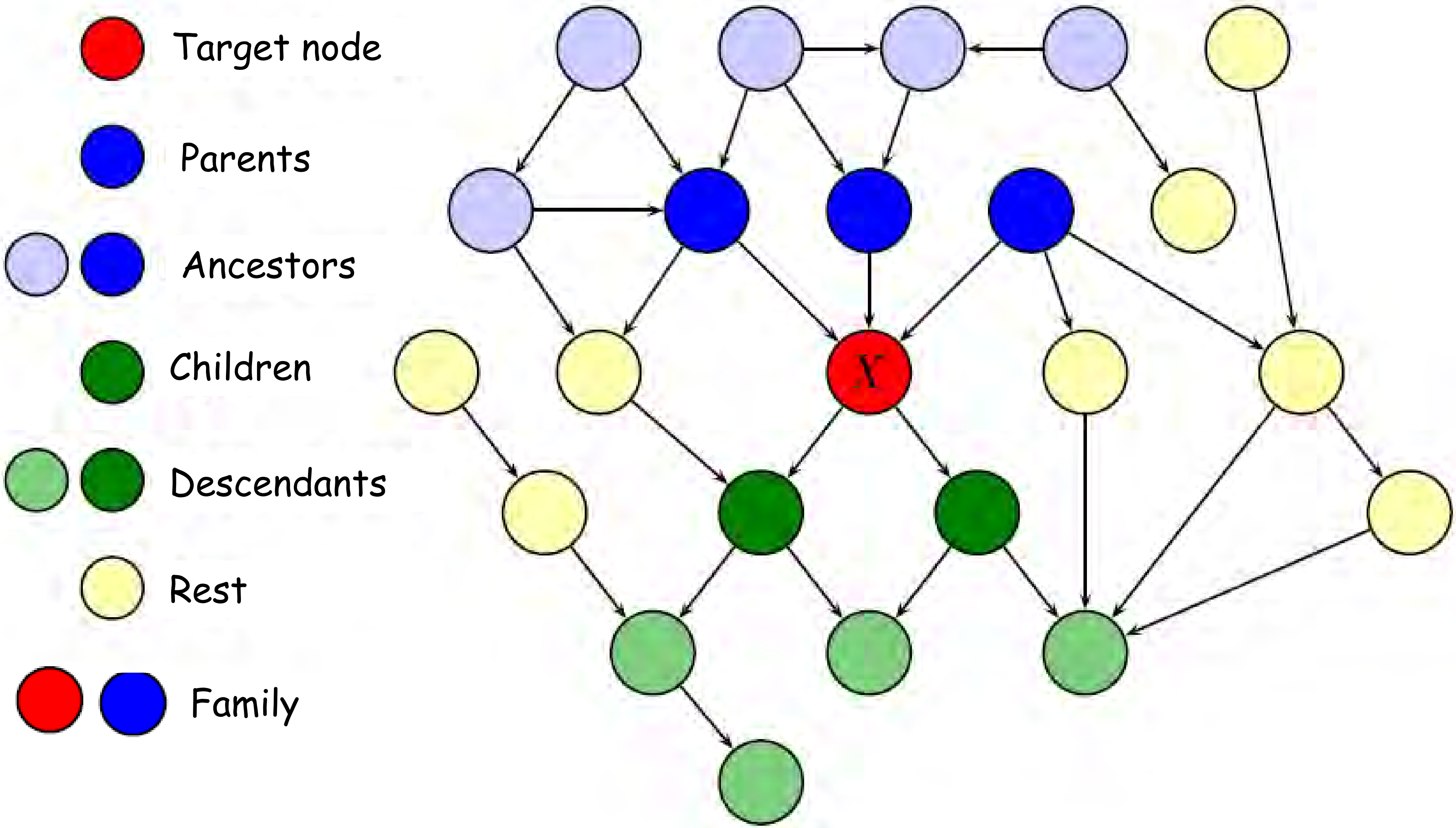
  → Not necessarily causality





- Quantitative part: a set of conditional probabilities that determine a unique JPD

Bayesian Networks: nodes



BNs: arcs (types of independence)

Independencies in a BN

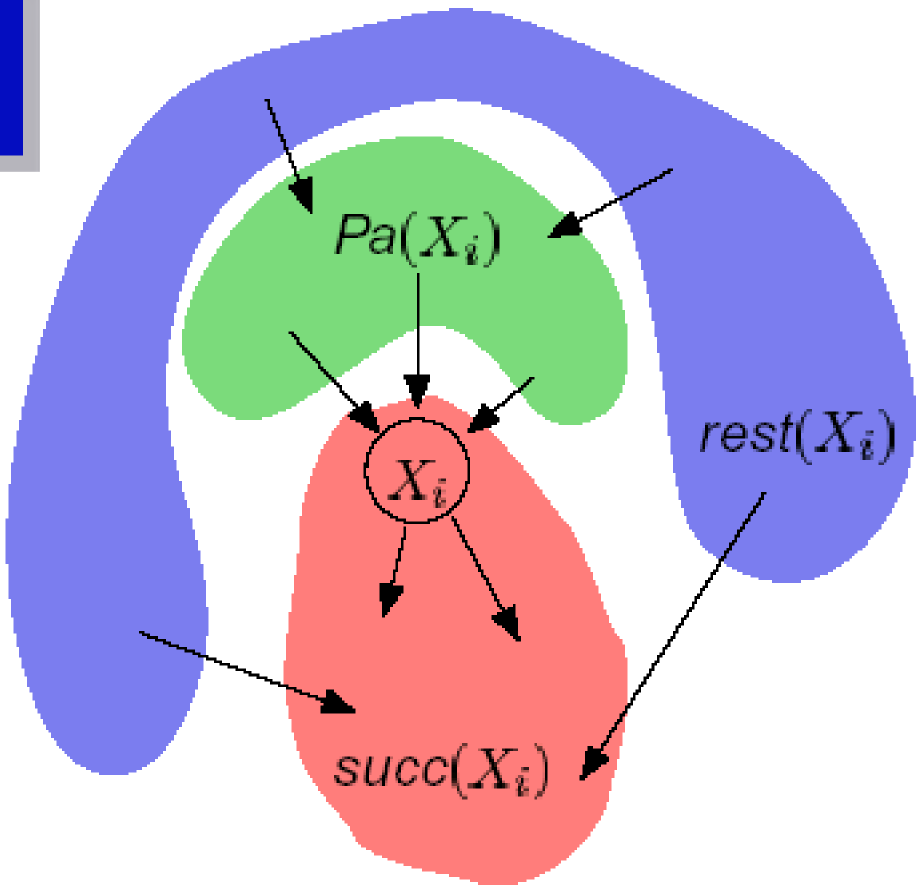
- A BN represents a set of independencies
- Distinguish:
  - Basic independencies: we should take care of verifying them when constructing the net
  - Derived independencies: from the previous independencies, by using the properties of the independence relations

Check them by means of the u-separation (or d-separation) criterion

Basic independencies

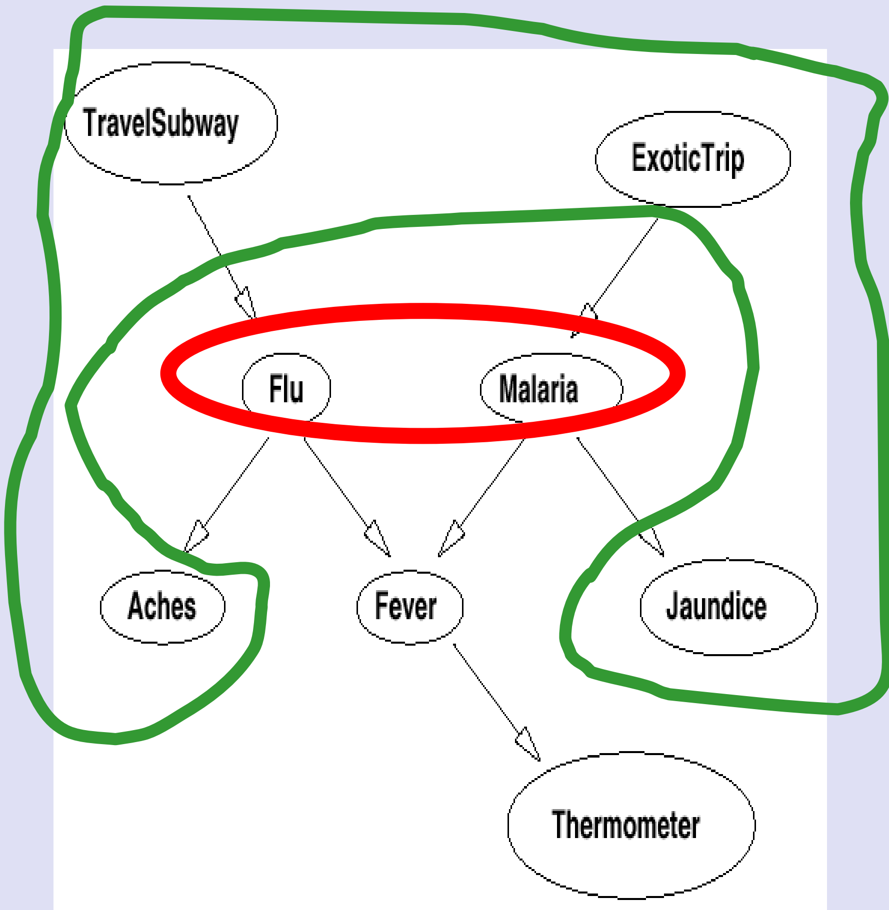
Basic independence: Markov condition

X_i is c.i. of its non-descendants, given its parents $Pa(X_i)$

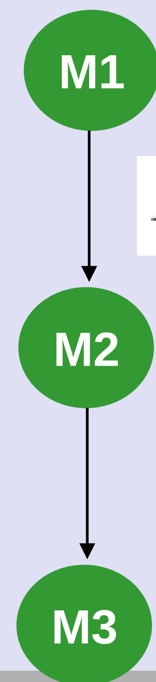


Basic independencies

Examples



Fever is c.i. of **Jaundice** given **Malaria** and **Flu**



$$I_P(M1, M3|M2)$$

Markov condition and JPD factorization

Quantitative part

- Use the chain rule and the Markov condition

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | X_1, \ldots, X_{i-1})$$

$$I_P(X, \text{non-desc} | Pa(X))$$

- Let $X_1,\ldots,X_n$ be an ancestral ordering (parents appear before their children in the sequence). It always exists (DAG)

- Using that ordering in the chain rule, in $\{X_1, \ldots, X_{i-1}\}$ there are non-descendants of X_i, and we have

$$P(X_i | X_1, \ldots, X_{i-1}) = P(X_i | pa(X_i))$$

Markov condition and JPD factorization

Quantitative part

- Therefore, we can recover the JPD by using the following factorization:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | pa(X_i))$$

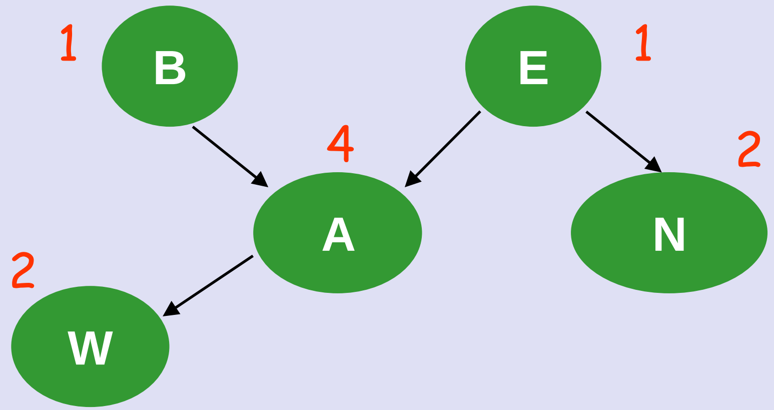
⬇

MODEL CONSTRUCTION EASIER:

- Only store local distributions at each node
- Fewer parameters to assign and more naturally
- Inference easier (reasoning)

Example of savings

With all binary variables:



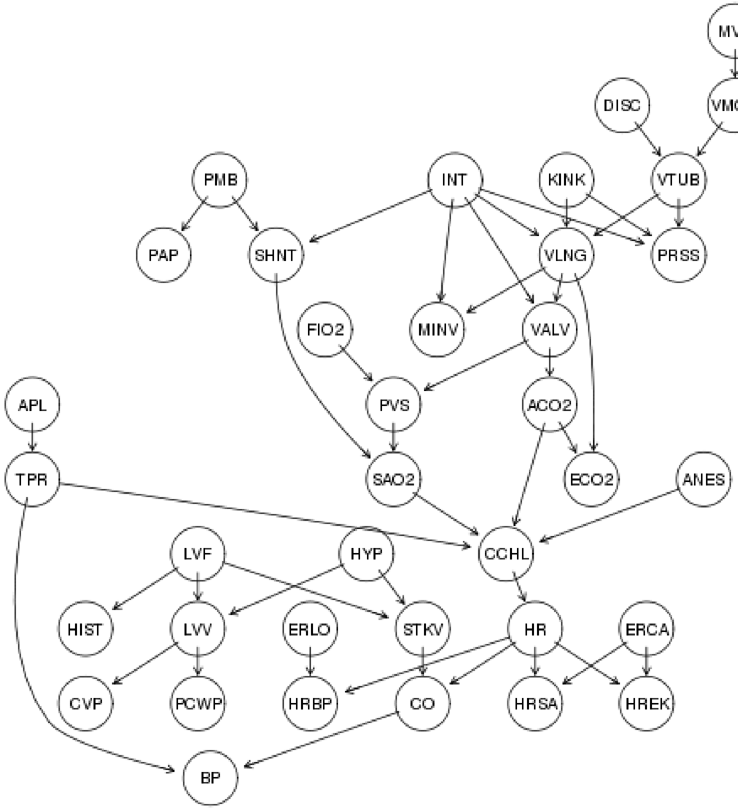
- $32=2^5-1$ probabilities for the JPD
- 10 with the factorization in the BN:

$$P(B,E,A,N,W) = P(W|A)P(A|B,E)P(N|E)P(B)P(E)$$

Example of savings

BN *Alarm* for monitoring ICU patients

- 2^{37} probabilities for the JPD vs. 509 in BN



Independencies derived from u-separation

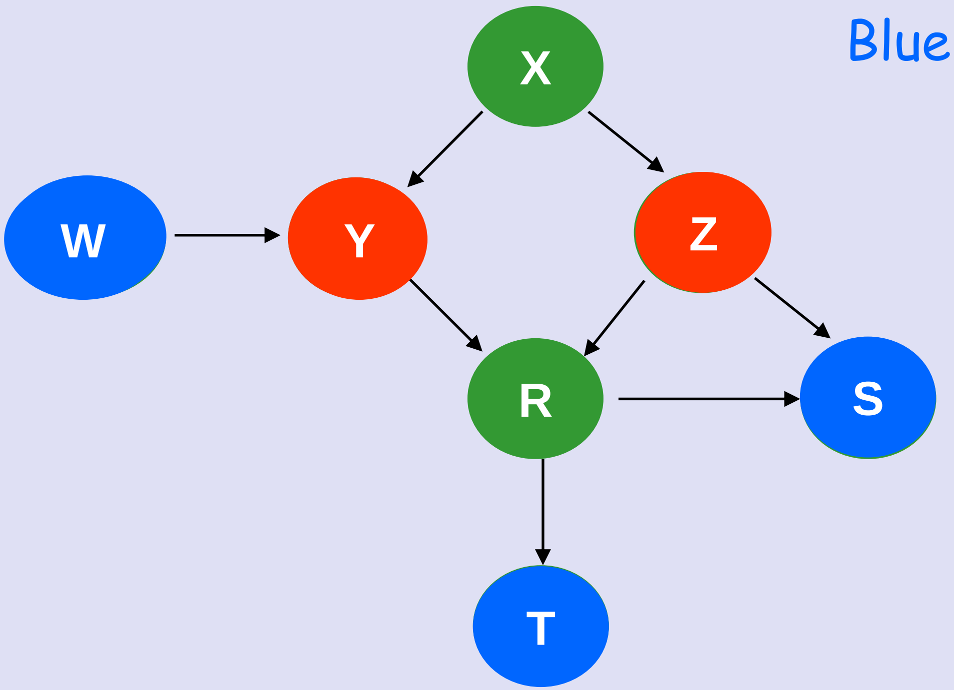
u-separation

$X \perp_G Y | Z$

- Obtain the minimum graph containing **X**,**Y**,**Z** and their ancestors (ancestral graph)
- The subgraph obtained is moralized (add a link between parents with children in common) and remove direction of arcs
- **Z** u-separates **X** and **Y** whenever **Z** is in all paths between **X** and **Y**

Independencies derived from u-separation

u-separation



Blue u-separated by red?

$W \perp S \mid \{Y,Z\}$?

$W \perp T \mid Y$?

Joining the two parts

Theorem [Verma and Pearl'90, Neapolitan'90]

- Let P be a prob. distribution of the variables in V and G=(V,E) a DAG.
  (G,P) holds the Markov condition iff

$$\mathbf{X} \perp_{\mathbf{G}} \mathbf{Y} | \mathbf{Z} \Longrightarrow I_P(\mathbf{X}, \mathbf{Y} | \mathbf{Z}) \quad \forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq V$$

u-separation defined by G

c.i. defined by P

disjoint

- Graph G represents all dependencies of P
- Some independencies of P may be not identified by d-separation in G

Definition of BN

Formal definition

- Let P be a JPD over V={X_1,…,X_n}.

  A BN is a tuple (G,P), where G=(V,E) is a DAG such that:
  - Each node of G represents a variable of V
  - The Markov condition is held (taking an ancestral ordering)
  - Each node has associated a local prob. $P(X_i|pa(X_i))$, distrib. such that

$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} P(X_i|pa(X_i))$$

  - u-separated variables in the graph are independent (G is a minimal I-map of P)

Building a BN

Expert / data / both

- Manual with the aid of an expert in the domain

Causal mechanisms —modelisation→ Causal graph —probabilities→ Bayesian net

  - Build it in the causal direction: BNs simpler and efficient

- Learning from a database

Database —algorithm→ Bayesian net

- A combination (experts → structure; database → probabilities)

Inference in Bayesian networks

Types of queries

Exact inference:

- Brute-force computation
- Variable elimination algorithm
- Message passing algorithm

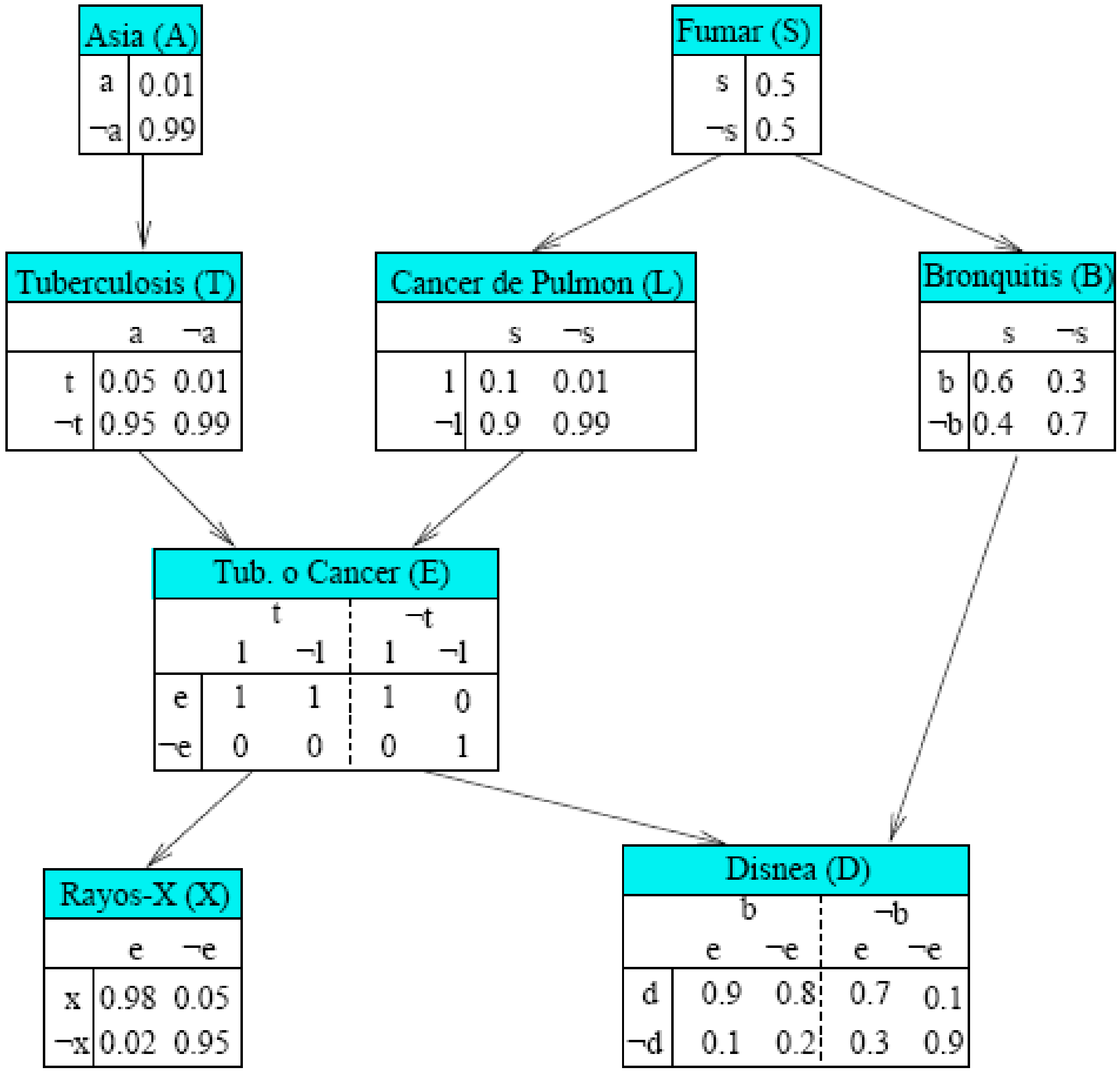
Approximate inference:

- Probabilistic logic sampling

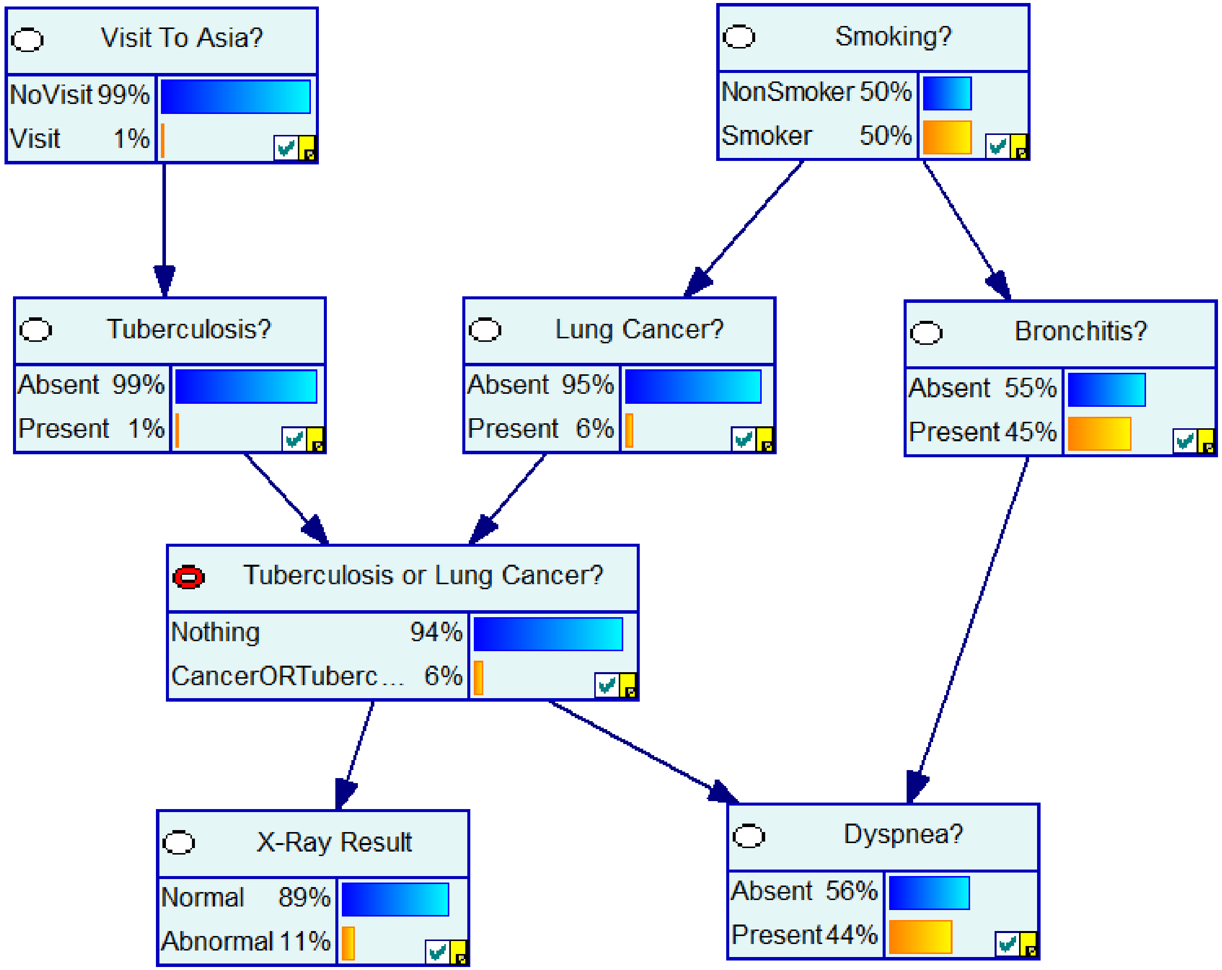
Example: Asia BN [Lauritzen & Spiegelhalter'88]

- Physician wants to diagnose her patients w.r.t. 3 diseases
  - Tuberculosis
  - Lung cancer
  - Bronchitis
- Causes or risk factors:
  - Recent Visit to Asia increases the chances of Tuberculosis
  - Smoking is a risk factor for both Lung cancer and Bronchitis
- Symptoms:
  - Dyspnea (shortness-of-breath) may be due to Tuberculosis, Lung cancer, Bronchitis, none of them, or more than one of them
  - Chest X-Ray. Neither symptom discriminates between Lung cancer and Tuberculosis

Example: Asia BN [Lauritzen & Spiegelhalter'88]

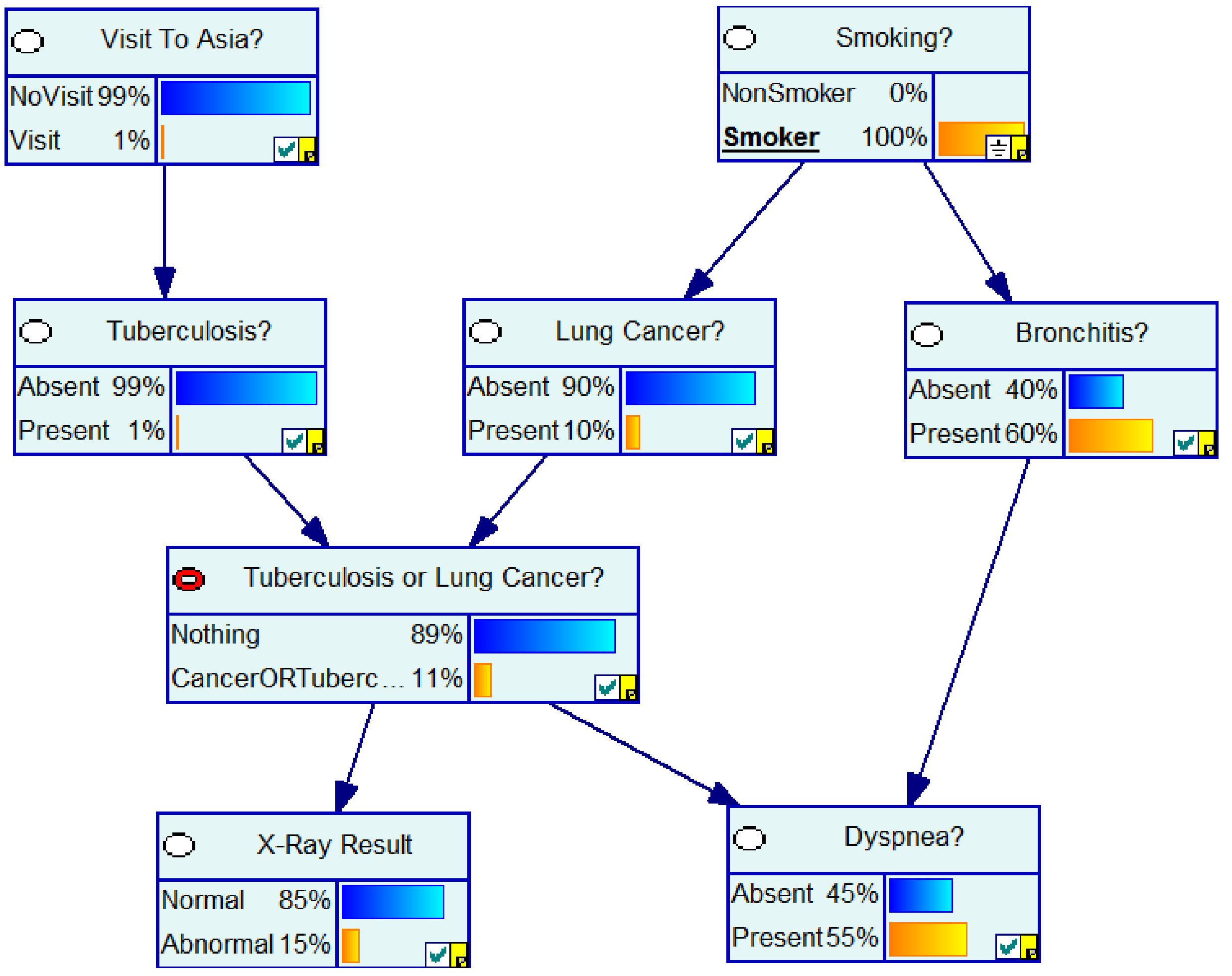


P(X)?

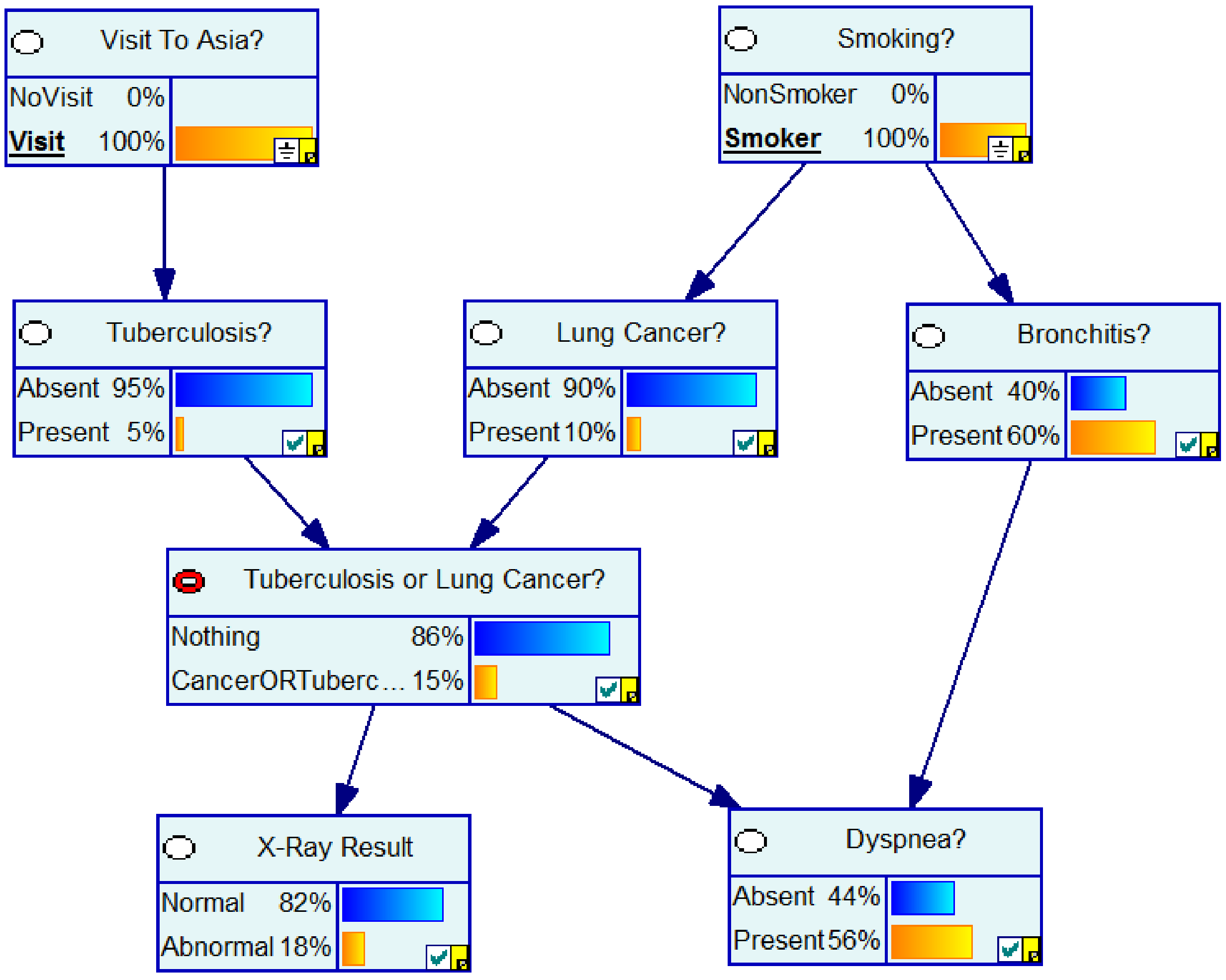


Visit To Asia?
NoVisit 99%
Visit 1%
Smoking?
NonSmoker 50%
Smoker 50%
Tuberculosis?
Absent 99%
Present 1%
Lung Cancer?
Absent 95%
Present 6%
Bronchitis?
Absent 55%
Present 45%
Tuberculosis or Lung Cancer?
Nothing 94%
CancerORTuberc... 6%
X-Ray Result
Normal 89%
Abnormal 11%
Dyspnea?
Absent 56%
Present 44%

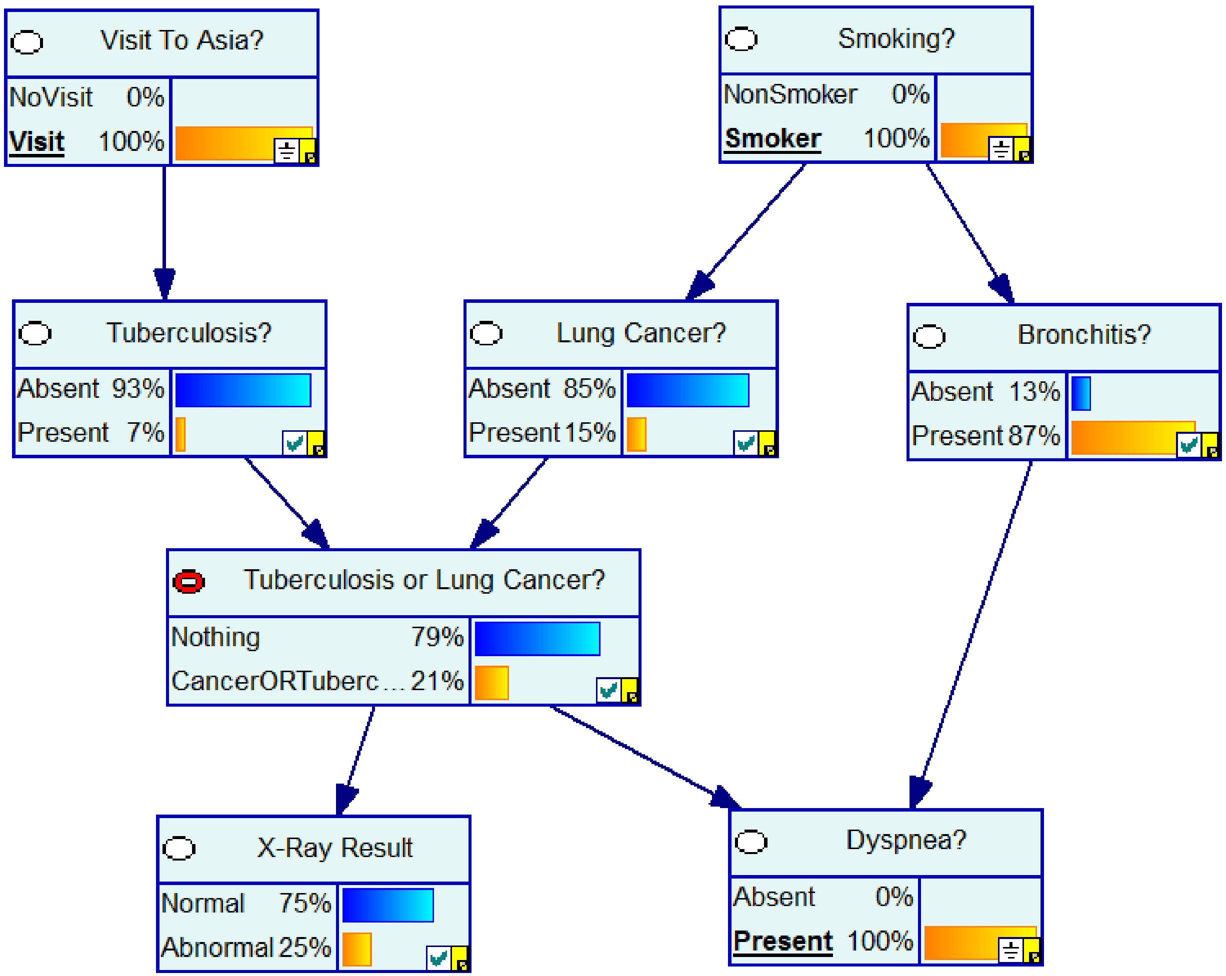
P(X|Smoker=yes)?



P(X|Asia=yes,Smoker=yes)?



P(X|Asia=yes,Smoker=yes,Dyspnea=yes)?



Types of queries

Queries: posterior probabilities

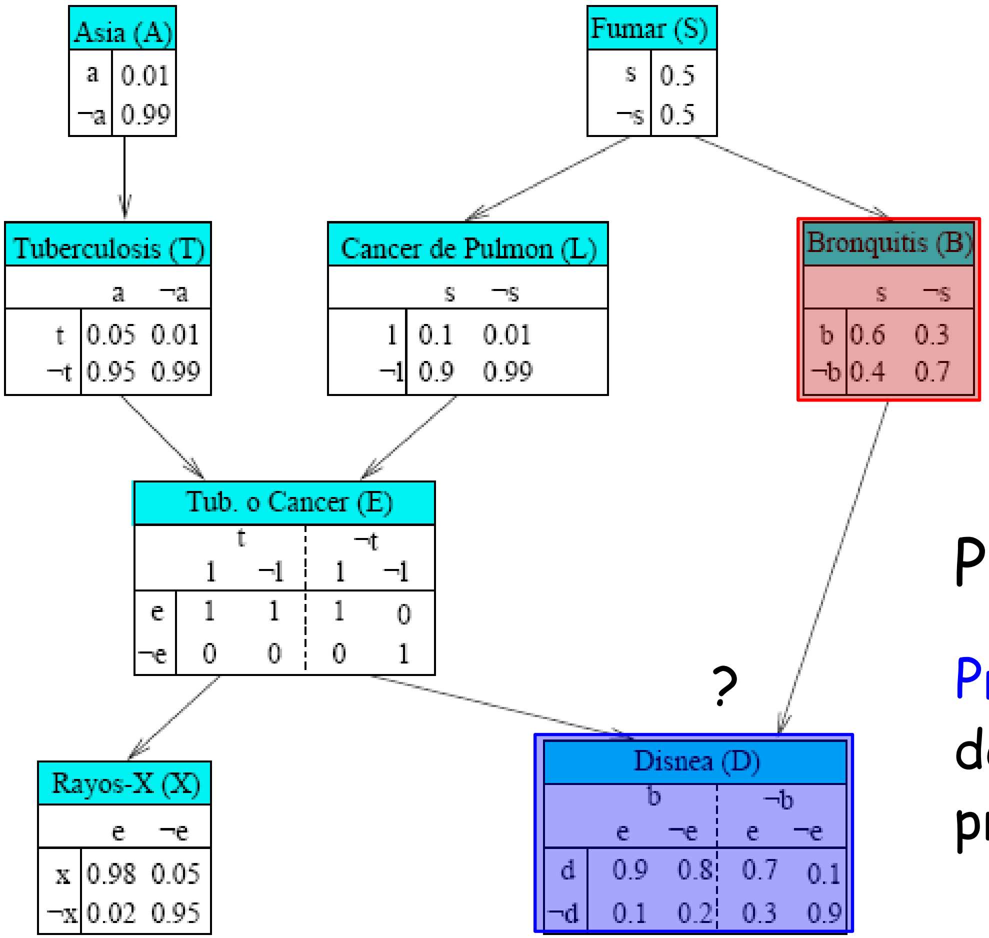
- Given some evidence e (observations),
  - Posterior probability of a target variable(s) X :

$$P(X|e)$$

X → Vector

answer queries about P

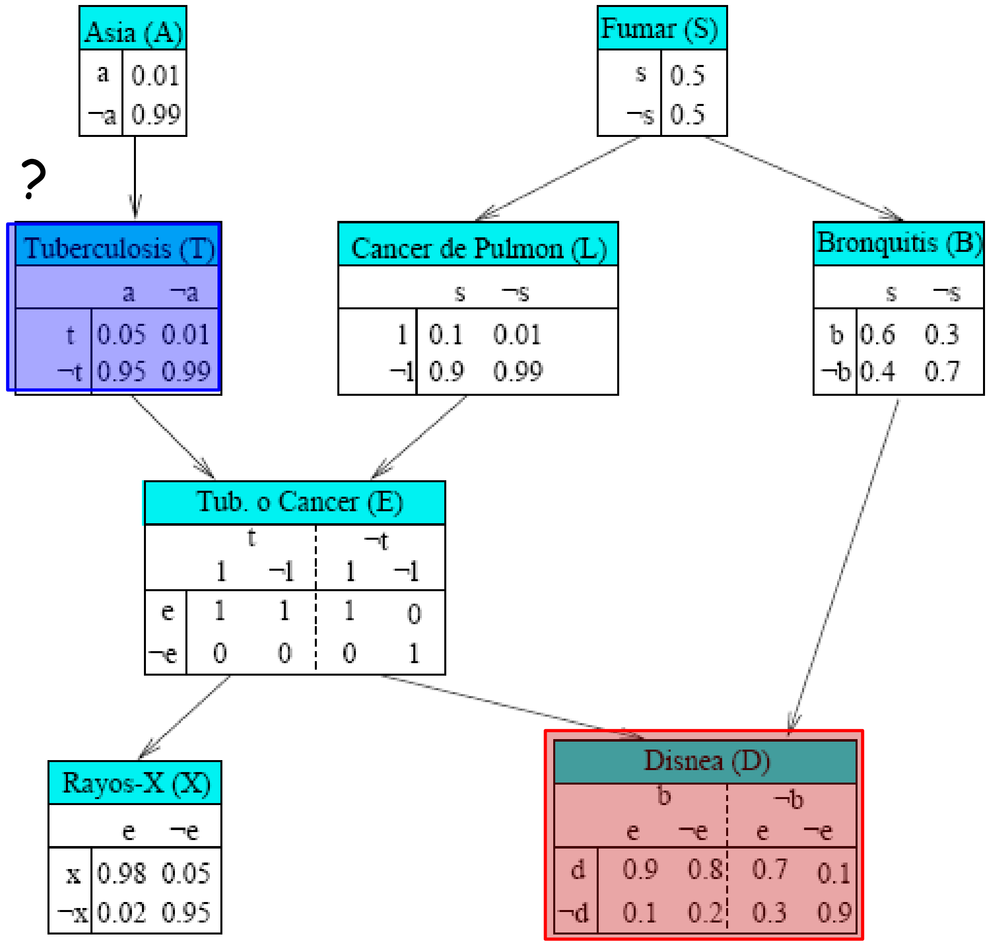
Other names: probability propagation, belief updating or revision…

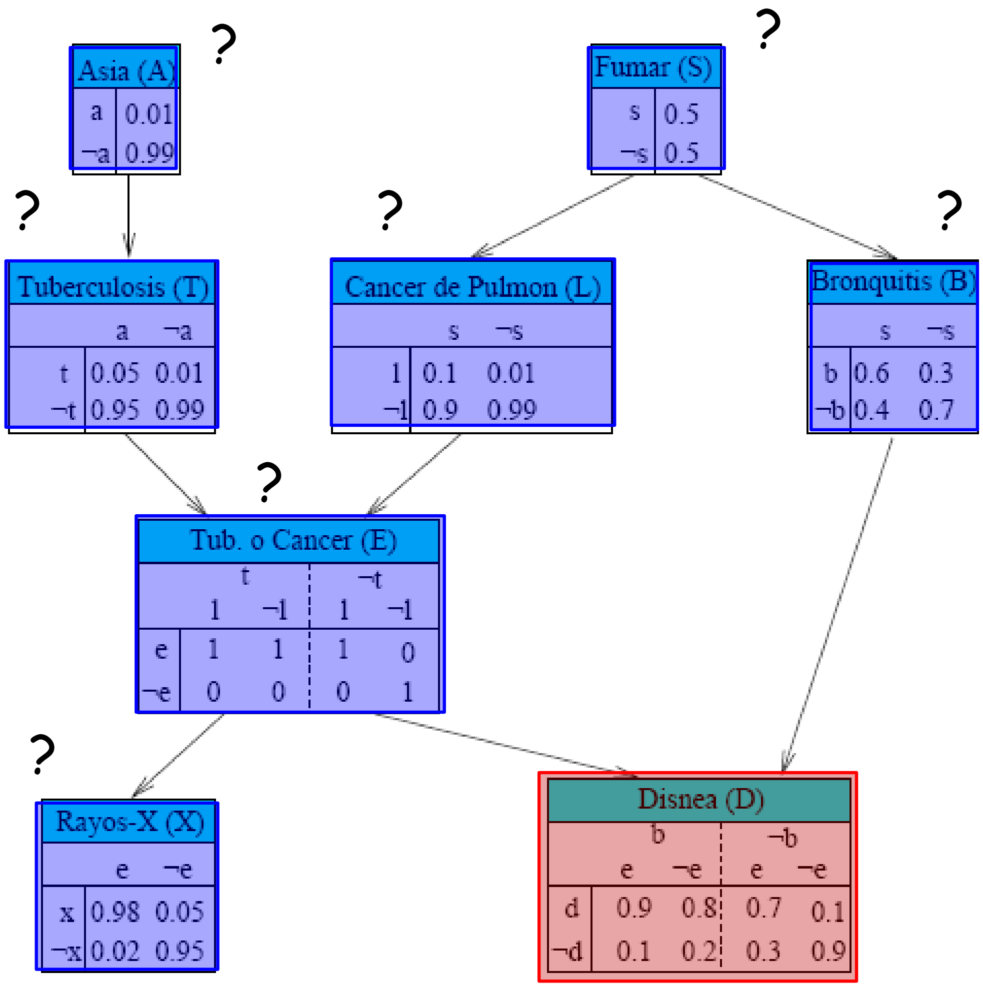


P(D|Bronquitis=yes)?

Predictive reasoning or deductive (causal inference): predict effects

P(T|Dyspnea=yes)? Diagnostic reasoning (diagnostic inference): diagnose the causes



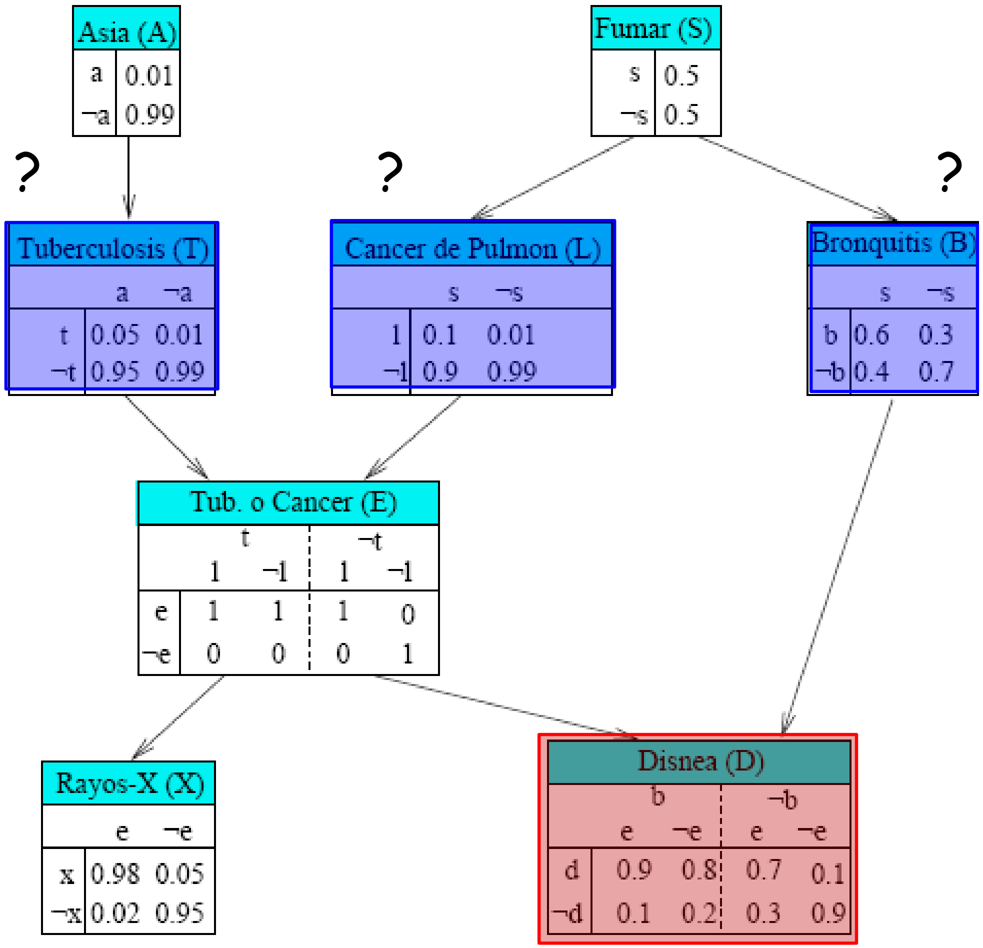


Max a posteriori (MAP)

(abductive inference):
event that best explains the evidence

Total (or MPE)

$$(x_1, \ldots, x_n) \text{ such that } \max P(x_1, \ldots, x_n|\mathbf{e})$$



Max a posteriori (MAP)

(abductive inference):

event that best explains the evidence

Partial

$$(x_1, \ldots, x_l) \text{ such that } \max P(x_1, \ldots, x_l | \mathbf{e})$$

Types of queries

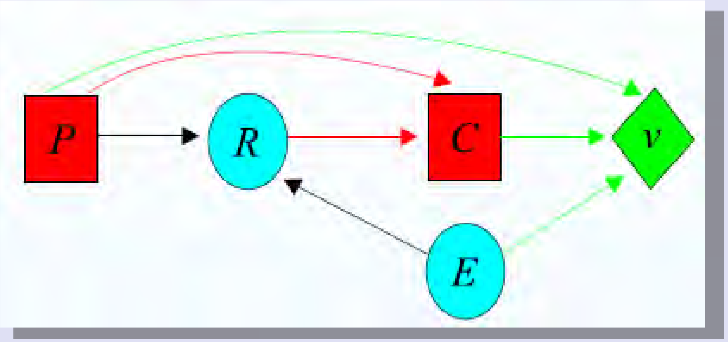
Classification

- Use MPE to:
  - Find most likely label, given the evidence

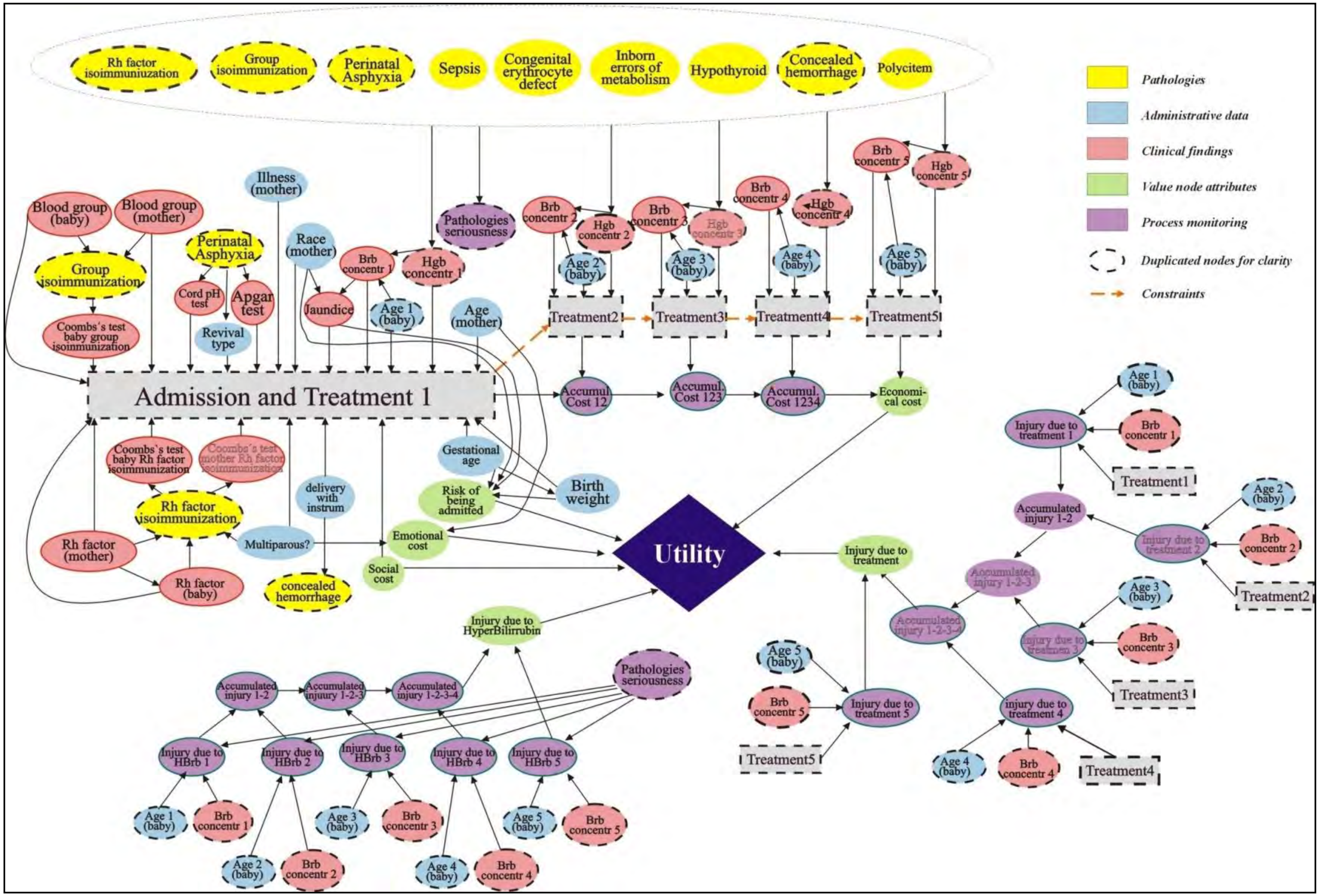
$$\max_c P(c \mid x_1,\dots,x_n)$$

Decision-making

- Optimal decisions (of maximum expected utility), with influence diagrams

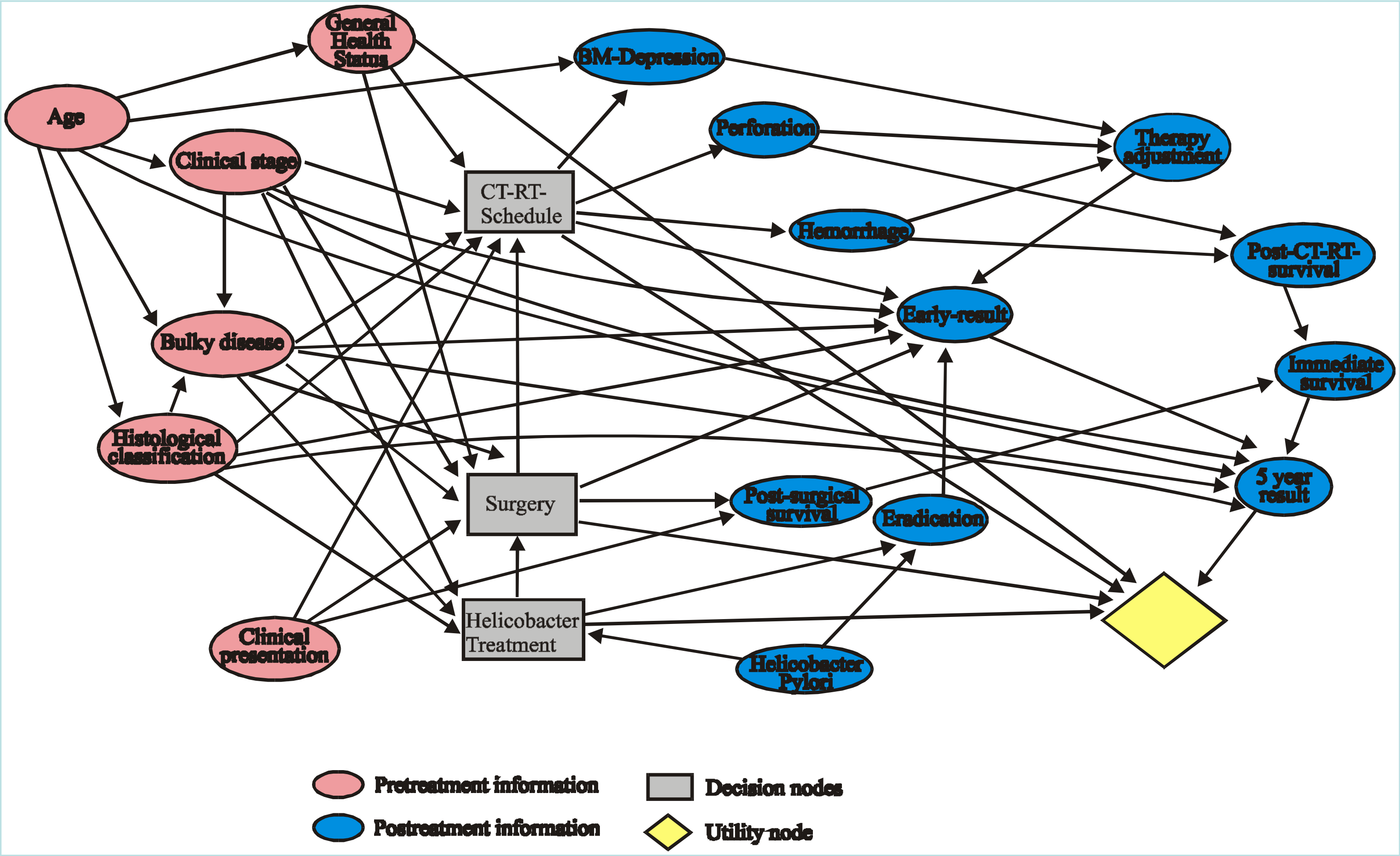


Examples: medicine (jaundice)



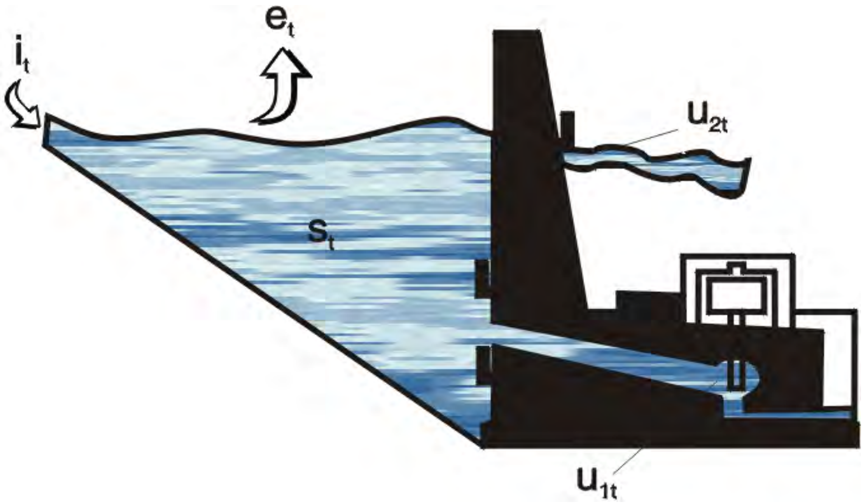
Gómez, M., Bielza, C., Fernández del Pozo, J.A., Ríos-Insua, S. (2007).
A graphical decision-theoretic model for neonatal jaundice. ***Medical Decision Making***, 27(3), 250-265

Examples: medicine (gastric lymphoma)



Bielza, C., Fernández del Pozo, J.A., Lucas, P. (2008).
Explaining clinical decisions by extracting regularity patterns. ***Decision Support Systems***, 44, 397-408

Examples: reservoir management



- Objectives: energy + water supply

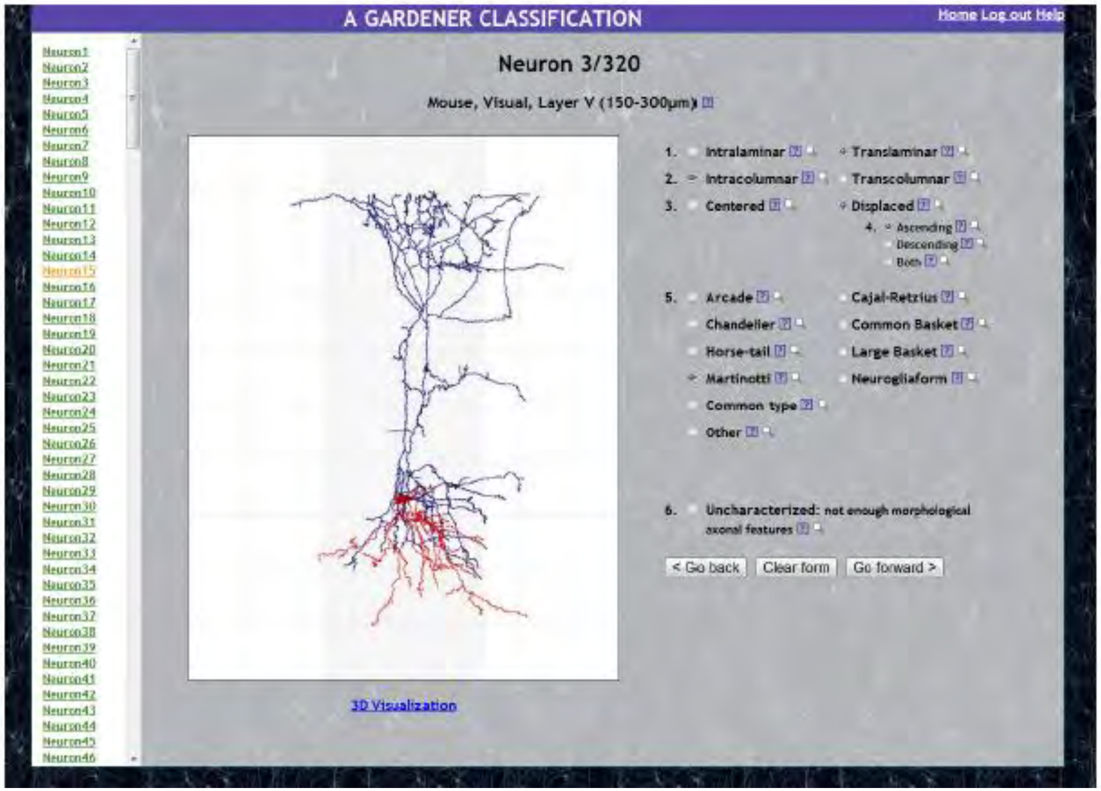
  Lake Kariba: Nearly 70% of the electricity is *consumed*

  Cahora Bassa: generated energy is *sold* to South Africa

Ríos Insua, D., Salewicz, K.A., Müller, P., Bielza, C. (1997) Bayesian methods in reservoir operations: the Zambezi river case. In *The Practice of Bayesian Analysis*, 107–130

Examples: neuroscience

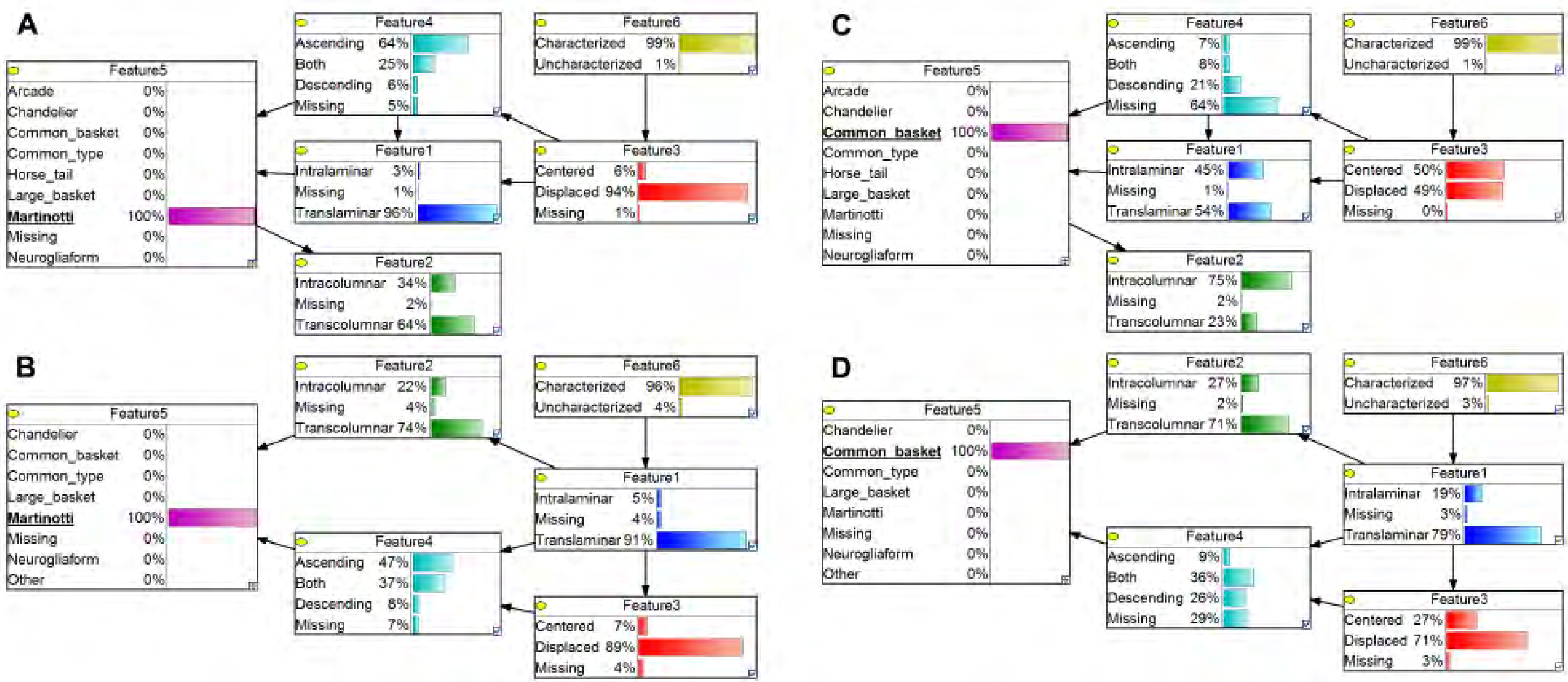
A 'gardener' classification of neurons



DeFelipe, J., Lopez-Cruz, P.L., Benavides-Piccione, R., Bielza, C., Larrañaga, P. *et al.* (2013). New insights into the classification and nomenclature of cortical GABAergic interneurons. ***Nature Reviews Neuroscience***, 14(3), 202-216

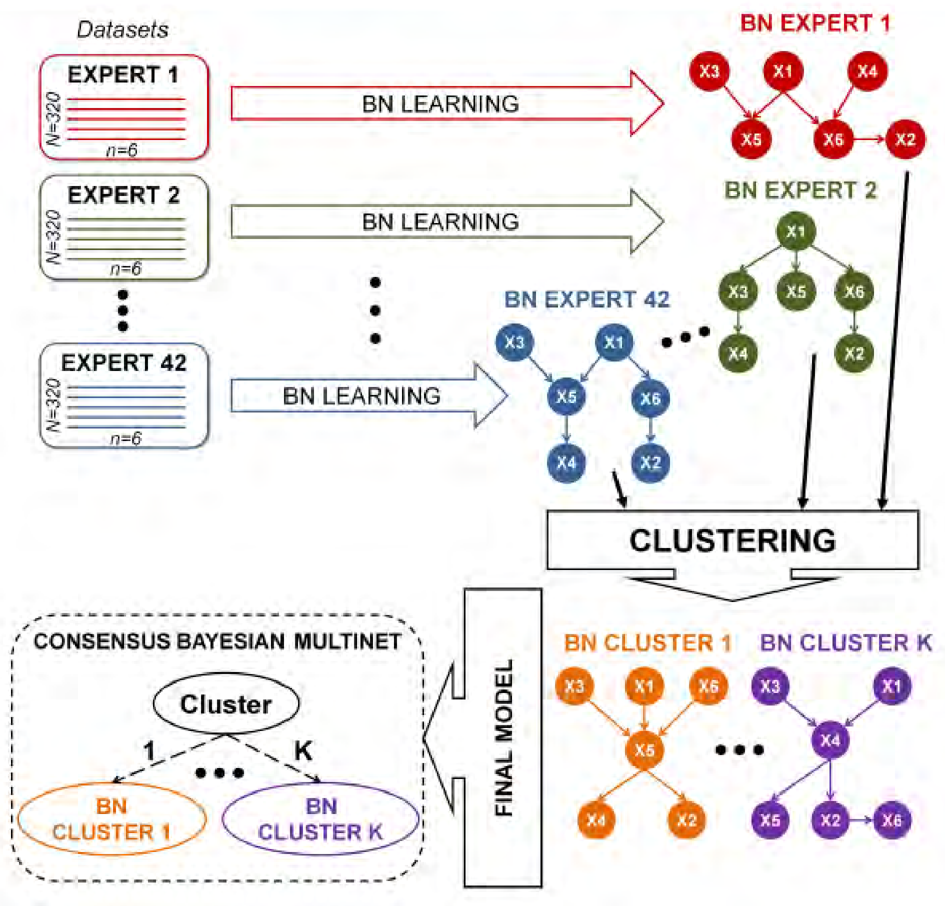
Examples: neuroscience

A Bayesian network learnt for each expert



Examples: neuroscience

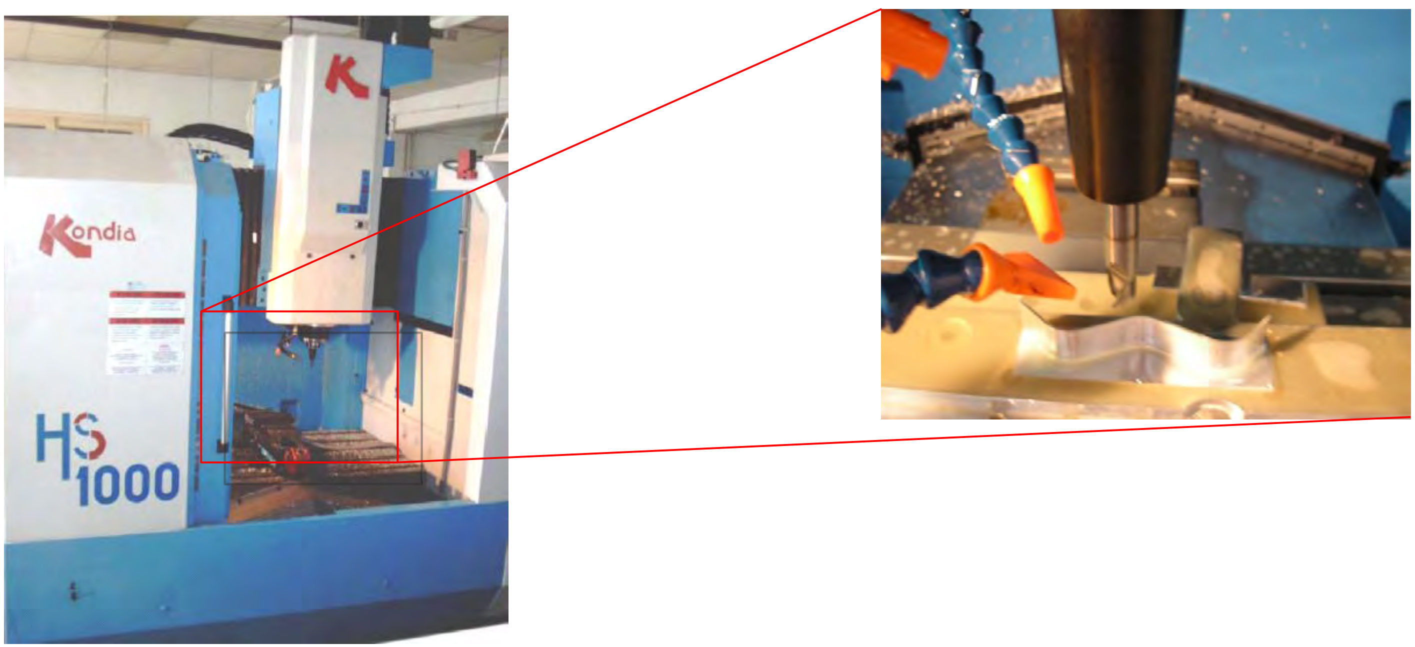
Inducing a consensus Bayesian multinet from a set of expert opinions



Lopez-Cruz, P.L., Larrañaga, P., J. DeFelipe, Bielza, C. (2014). Bayesian network modeling of the consensus between experts: An application to neuron classification. ***International Journal of Approximate Reasoning***, 55(1), 3-22

Examples: industry (high-speed machining)

- How to online guarantee a *good surface roughness*
  - Cutting parameters: spindle speed, cutting force, feed rate, cutting depth…
  - Tool variables: number of teeth (flutes), tool diameter…



Correa, M., Bielza, C., Ramírez, M. de J., Alique, J.R. (2008) A Bayesian network model for surface roughness prediction in the machining process. ***International Journal of Systems Science***, 39(12), 1181-1192

Exact inference [Pearl'88; Lauritzen & Spiegelhalter'88]

Brute-force computation of P(X|e)

- Conceptually simple but computationally complex
- For a BN with n variables:

$$P(X_i) = \sum_{X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_n} \prod_{j=1}^{n} P(X_j|Pa(X_j))$$

Brute-force approach

- But this amounts to computing the JPD, often very inefficient and even intractable computationally
- CHALLENGE: Without computing the JDP, exploit the factorization encoded by the BN and the distributive law (local computations)

Exact inference

Improving brute-force

- Use the JPD factorization and the distributive law

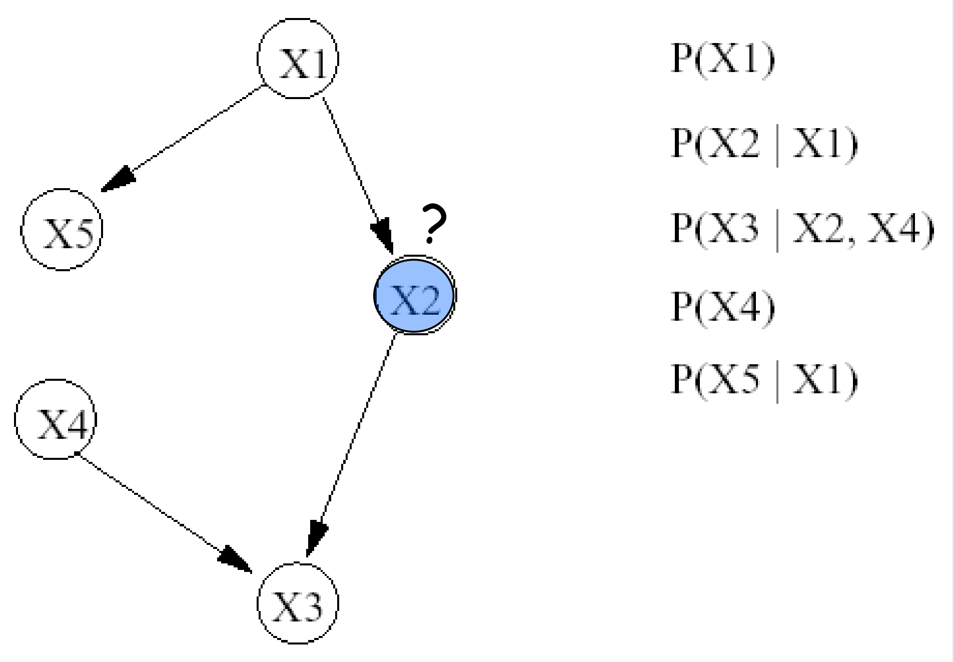


Table with 32 inputs (JPD)
(if binary variables)

$$P(X_2) = \sum_{X_1, X_3, X_4, X_5} \{P(X_1) \cdot P(X_2|X_1) \cdot P(X_3|X_2, X_4) \cdot P(X_4) \cdot P(X_5|X_1)\}$$

Exact inference

Improving brute-force

- Arrange computations effectively, moving some additions
  - over X_5 and X_3:

$$= \sum_{X_1, X_4} \left\{ \left(\underbrace{\sum_{X_5} P(X_5|X_1)}_{f_1(X_1)} \right) \cdot P(X_1) \cdot P(X_2|X_1) \cdot \left(\underbrace{\sum_{X_3} P(X_3|X_2, X_4)}_{f_2(X_2, X_4)} \right) \cdot P(X_4) \right\}$$

Biggest table with 8 (like the BN)

  - over X_4:

$$= \sum_{X_1} \left\{ \left(\underbrace{\sum_{X_5} P(X_5|X_1)}_{f_1(X_1)} \right) \cdot P(X_1) \cdot P(X_2|X_1) \cdot \left[\underbrace{\sum_{X_4} \left(\underbrace{\sum_{X_3} P(X_3|X_2, X_4)}_{f_2(X_2, X_4)} \right) \cdot P(X_4)}_{f_3(X_2)} \right] \right\}$$

Exact inference

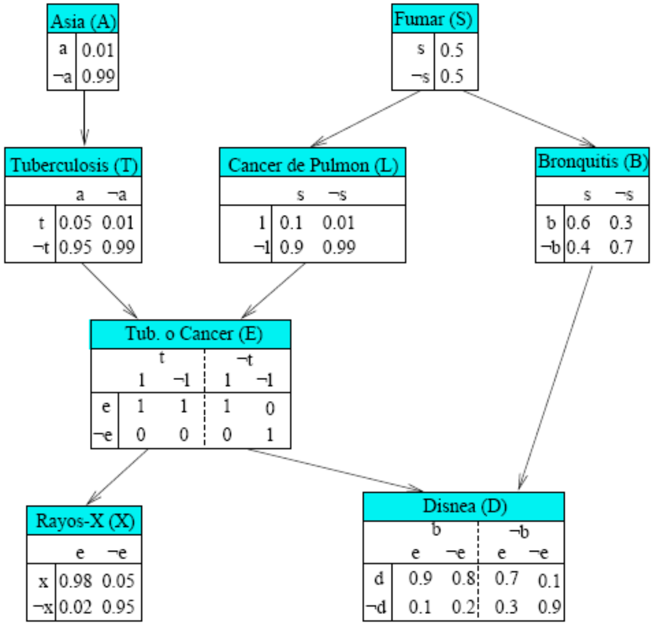
Variable elimination (VE) algorithm

- Wanted: $P(X_i|\mathbf{e})$ → ONE variable
- A list with all functions of the problem $\{f_1, \ldots, f_n\}$
- Select an elimination order σ of all variables (except i)
- For each X_k from σ, if F is the set of functions that involve X_k:
  - Delete F from the list
  - Compute $f' = \sum_{X_k} (\prod_{f \in F} f)$
  - Add f' to the list
- Output: combination (multiplication) of all functions in the current list

Eliminate X_k = combine all the functions that contain this variable and marginalize out X_k

Repeat the algorithm for each target variable

Example with Asia network; P(D)?



Brute-force approach

- Compute P(D) by brute-force:

$$P(d)=\sum_x\sum_b\sum_e\sum_l\sum_t\sum_s\sum_a P(a,s,t,l,e,b,x,d)$$

- Complexity is exponential in the size of the graph (n×number of states for each variable)

Example with Asia network: VE

$\sigma_1 = T, S, E, A, L, B, X.$

1 $\mathcal{L} = \{f_A(A), \underbrace{f_T(T,A)}, f_S(S), f_L(L,S), f_B(B,S), \underbrace{f_E(E,T,L)}, f_X(X,E), f_D(D,E,B)\}$. **Delete T.**

$$g_1(A,E,L) = \sum_T (f_T(A,T) \times f_E(E,T,L))$$

not necessarily a probability term

size = 16

2 $\mathcal{L} = \{f_A(A), \underbrace{f_S(S), f_L(L,S), f_B(B,S)}, f_X(X,E), f_D(D,E,B), g_1(A,E,L)\}$. **Delete S.**

$$g_2(L,B) = \sum_S (f_S(S) \times f_L(L,S) \times f_B(B,S))$$

size = 8

3 $\mathcal{L} = \{f_A(A), \underbrace{f_X(X,E), f_D(D,E,B), g_1(A,E,L)}, g_2(L,B)\}$. **Del. E**

$$g_3(X,D,B,A,L) = \sum_E (f_X(X,E) \times f_D(D,E,B) \times g_1(A,E,L))$$

size = 64

Example with Asia network: VE

4 $\mathcal{L} = \{\underbrace{f_A(A)}, g_2(L,B), \underbrace{g_3(X,D,B,A,L)}\}$. **Delete A** size = 32

$$g_4(X,D,B,L) = \sum_A (f_A(A) \times g_3(X,D,B,A,L))$$

5 $\mathcal{L} = \{\underbrace{g_2(L,B), g_4(X,D,B,L)}\}$. **Delete L.** size = 16

$$g_5(X,D,B) = \sum_L g_2(L,B) \times g_4(X,D,B,L)$$

6 $\mathcal{L} = \{\underbrace{g_5(X,D,B)}\}$. **Delete B.** size = 8

$$g_6(X,D) = \sum_B g_5(X,D,B)$$

7 $\mathcal{L} = \{\underbrace{g_6(X,D)}\}$. **Delete X.** size = 8

4

$$g_7(D) = \sum_X g_6(X,D)$$

8 return normalize($g_7(D)$)

elimination order $\sigma_1 = A, X, T, S, L, E, B$ → Size = 8

Message passing algorithm

Basic operations for a node

- Ask info(i,j): Target node i asks info to node j. Does it for all neighbors j. They do the same until there are no nodes to ask
- Send-message(i,j): Each node sends a message $M^{i \to j}$ to the node that asked him the info… until reaching the target node
- A message is defined over the intersection of domains, F_i and F_j, of f_i and f_j:

$$M^{i \to j} = \sum_{X \notin F_i \cap F_j} f_i \cdot \left(\prod_{k \neq j} M^{k \to i} \right)$$

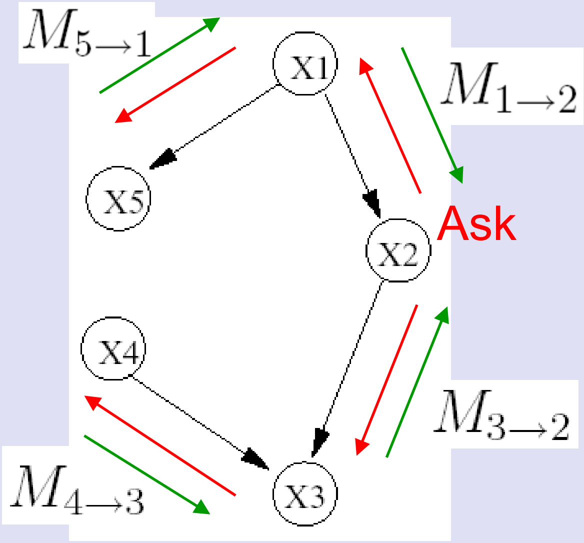
- And finally, we calculate locally at each node i:

Target combines all received info with his info and marginalize over the target variable

$$P(X_i|\mathbf{e}) = \text{normalize}\left[\sum_{X_j \neq X_i} \left(f_i \cdot \prod_{k \in \text{neighbours}(X_i)} M^{k \to i} \right) \right]$$

Message passing algorithm

Procedure for $P(X_2)$

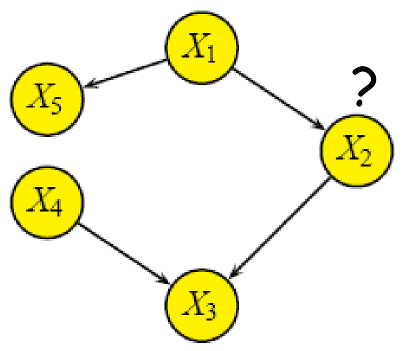


CollectEvidence

Exact inference

VE as a message passing algorithm

- Direct correspondence:



Mess. $$P(X_2) = \sum_{X_1} c_2 M^{1\to 2} M^{3\to 2}$$

VE $$= \sum_{X_1} \left\{ \underbrace{\left(\sum_{X_5} P(X_5|X_1) \right)}_{f_1(X_1)} \cdot P(X_1) \cdot P(X_2|X_1) \cdot \underbrace{\left[\sum_{X_4} \underbrace{\left(\sum_{X_3} P(X_3|X_2,X_4) \right)}_{f_2(X_2,X_4)} \cdot P(X_4) \right]}_{f_3(X_2)} \right\}$$

$M^{5\to 1}$ $M^{1\to 2}$ $M^{3\to 2}$ $M^{4\to 3}$

Message passing algorithm

Computing prob. $P(X_i|e)$ of all (unobserved) variables i at a time

- Rerun this for *each* node: many messages *repeated*!
- Or, we can use 2 rounds of messages as follows:
  - Select a node as a root (or pivot)
  - Ask or collect evidence: leaves → root *(messages in downward direction)*. As VE.
  - Distribute evidence: root → leaves *(upward direction)*
  - Calculate marginal distributions at each node by local computation, i.e. using its incoming messages
- Enables to compute the posteriors of all variables in twice the time it takes to compute that of **one** single variable

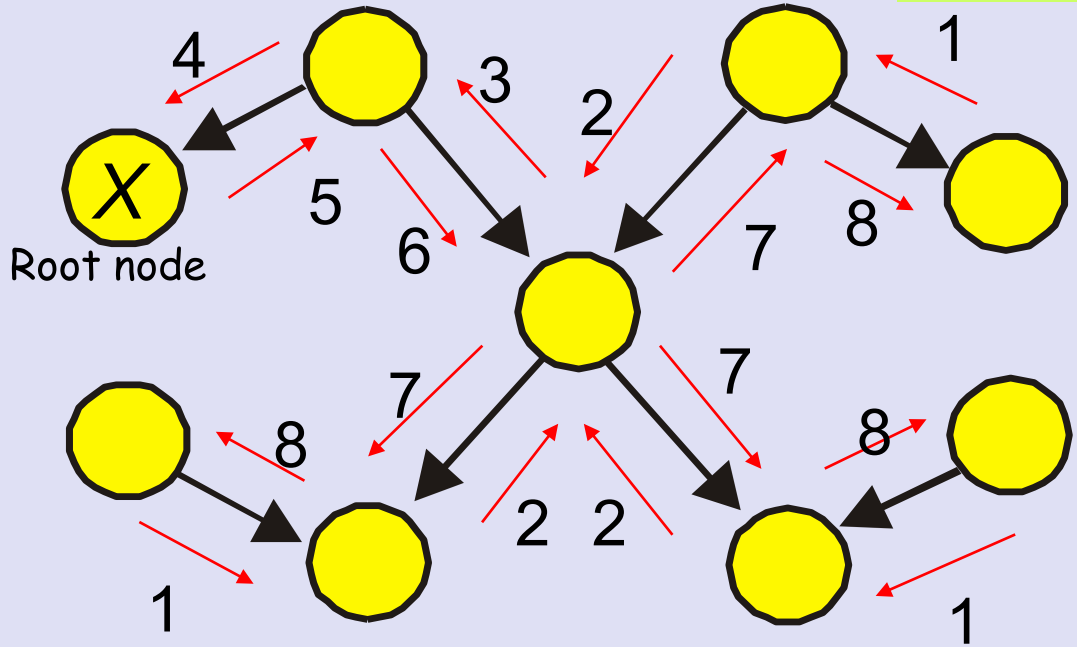
Message passing algorithm

First sweep:

CollectEvidence

Second sweep:

DistributeEvidence

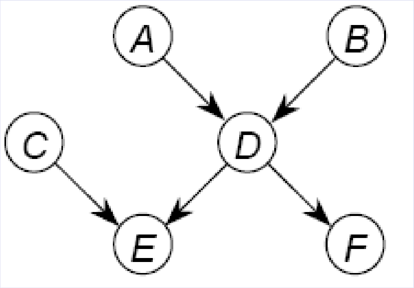


Exact inference

Complexity of exact inference in BNs

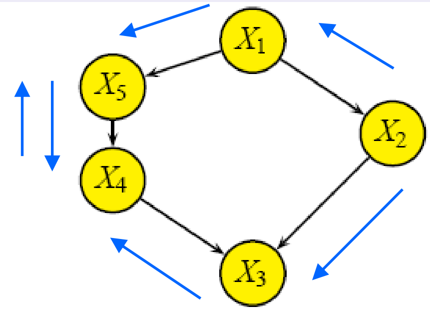
- In general BNs, exact inference is NP-complete [Cooper 1990]
- In BN without loops (cycles in the underlying undirected graph) -**polytrees**-, inference is easy (polynomial)

Polytree=DAG without loops



There is only one path between any pair of nodes =singly connected graph

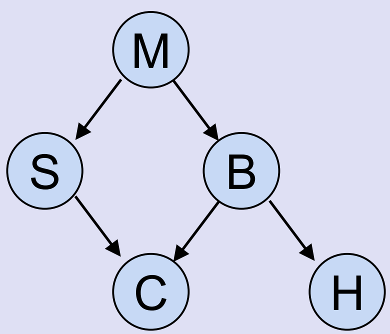
Exact inference



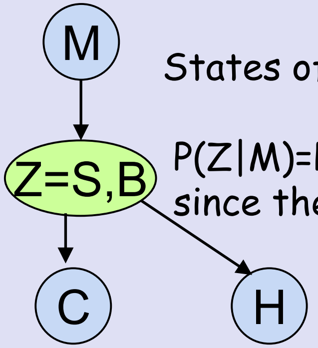
Multiply-connected BNs

Alternative: clustering methods [Lauritzen & Spiegelhalter'88]

- Transform the BN into an auxiliary representation (clique tree or junction tree) by merging nodes and removing loops



Create a new node Z, that combines S and B



States of Z: {tt,ft,tf,ff}

P(Z|M)=P(S|M)P(B|M) since they are c.i. given M

P(H|Z)=P(H|B) since H c.i. of S given B

***Metastatic cancer** (M) is a possible cause of brain tumors (B) and an explanation for increased total serum calcium (S). In turn, either of these could explain a patient falling into a coma (C). Severe headache (H) is also associated with brain tumors.*

Approximate inference

Stochastic simulation

- Uses the network to generate a large number of cases (full instantiations) from the network distribution
- $P(X_i|\mathbf{e})$ is estimated using these cases by counting observed frequencies in the samples. By the Law of Large Numbers, the estimate converges to the exact probability as more cases are generated
- Approximate inference in BNs within an arbitrary tolerance or accuracy is NP-hard
  - In practice, if **e** is not too unlikely, convergence is quickly

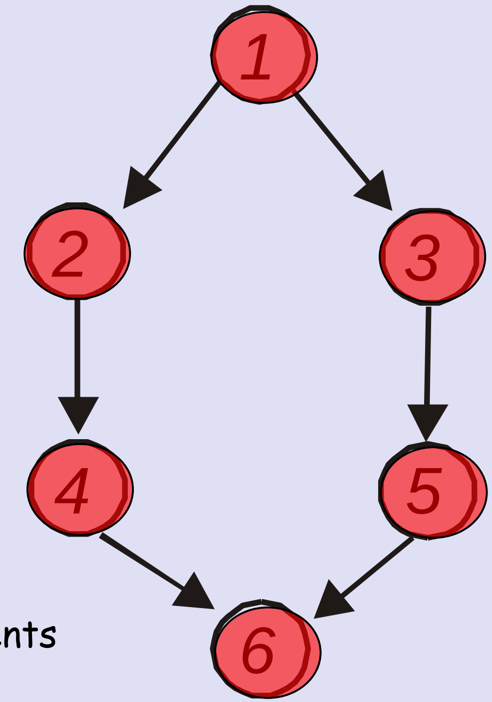
- P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60:141–153, 1993.

Approximate inference

Probabilistic logic sampling [Henrion'88]

- Given an ancestral ordering of the nodes (parents before children), generate from X once we have generated from its parents (i.e. from the root nodes down to the leaves)

When all the nodes have been visited, we have a case, an instantiation of all the nodes in the BN



Use conditional prob. given the known values of the parents

- Repeat and use the observed frequencies to estimate $P(X_i|\mathbf{e})$

Approximate inference

Probabilistic logic sampling

- Suppose we obtain the following samples:

(0,1,1,1,1,1), (0,1,0,1,1,1), (1,0,0,1,1,1), (0,0,1,1,1,0), (1,1,1,1,0,0)

- Then: $$\hat{p}(X_1 = 0) = \frac{3}{5}$$

- With evidence, e.g. X_2=1, we discard the third and fourth samples and we would repeat until having a sample of size 5 as desired

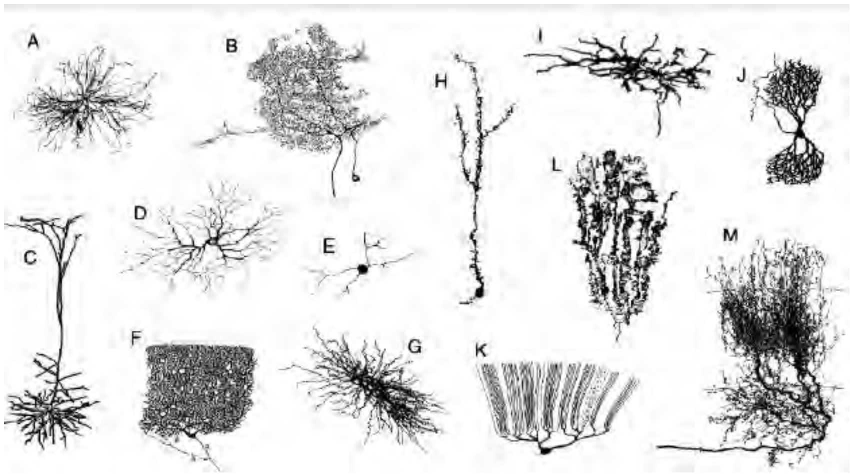
(0,1,1,1,1,1), (0,1,0,1,1,1), (1,1,0,0,1,1), (1,1,1,1,1,0), (1,1,1,1,0,0)

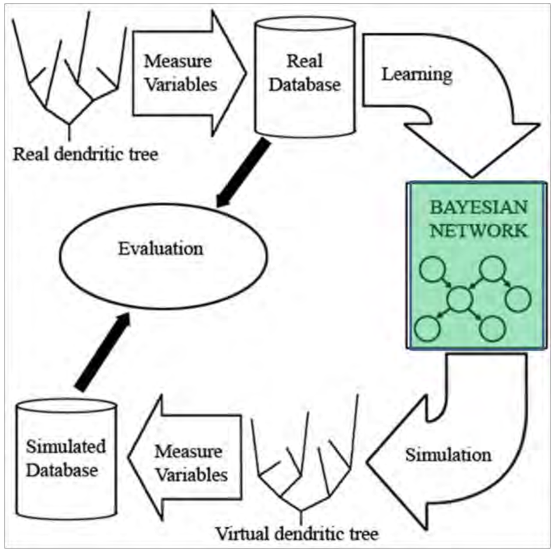
$$\hat{p}(X_1 = 0 | X_2 = 1) = \frac{2}{5}$$

Examples: neuroscience

Models and simulation of 3D dendritic tree morphology

- How and why vastly different shapes arise is still largely unknown
- Understanding how formed in the brain, their normal function and why they are often malformed in neurological diseases or under the effects of some drugs (cocaine, morphine)

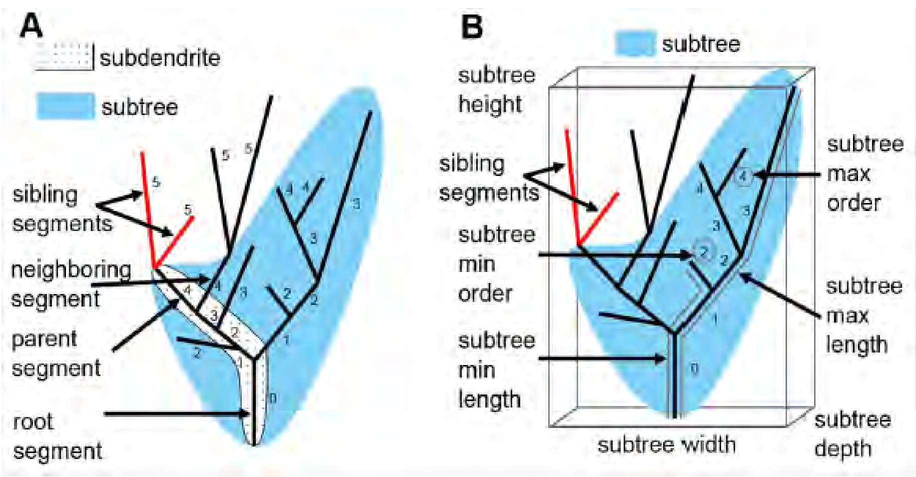


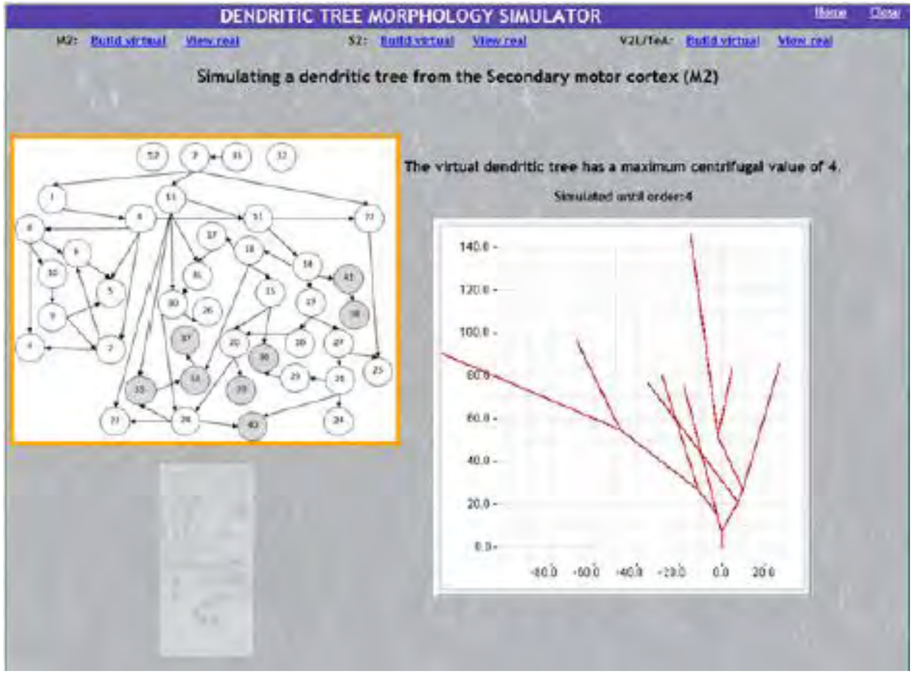


Lopez-Cruz, P.L., Bielza, C., Larrañaga, P., Benavides-Piccione, R. & DeFelipe, J. (2011).
Models and simulation of 3D neuronal dendritic trees using Bayesian networks. ***Neuroinformatics***, 9(4), 347-369

Examples: neuroscience

Models and simulation of 3D dendritic tree morphology





Resources

On the web

- BN repositories:
  http://www.cs.huji.ac.il/site/labs/compbio/Repository/
  http://genie.sis.pitt.edu/index.php/network-repository
  http://www.bnlearn.com/bnrepository/

- Much information:
  http://www.cs.ualberta.ca/~greiner/bn.html#applic

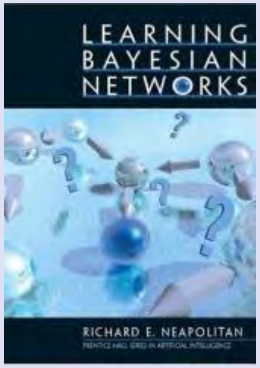
- Coursera (D. Koller @ Stanford): “Probabilistic graphical models”: https://class.coursera.org/course/pgm

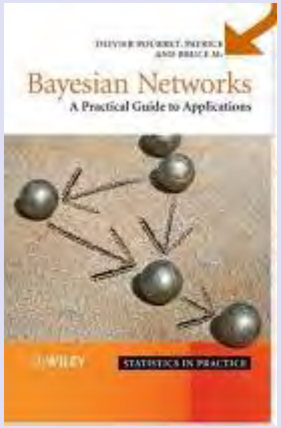
Texts

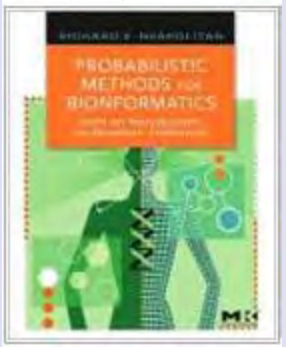
- E. Castillo, J.M. Gutiérrez, A.S. Hadi (1997) *Expert Systems and Probabilistic Network Models*. Springer
- R.G. Cowell, A.P. Dawid, S.L. Lauritzen, D.J. Spiegelhalter (1999) *Probabilistic Networks and Expert Systems*. Springer
- F.V. Jensen, T. Nielsen (2007) *Bayesian Networks and Decision Graphs*. Springer
- K.B. Korb, A. Nicholson (2004) *Bayesian Artificial Intelligence*. Chapman and Hall
- R. Neapolitan (2004) *Learning Bayesian Networks*. Prentice Hall
- U. Kjaerulff, A. Madsen (2008) *Probabilistic Networks and Influence Diagrams*. Available at http://www.cs.aau.dk/~uk/papers/pgm-book-I-05.pdf
- J. Pearl (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann
- Proceedings of the most important related conference: *Uncertainty in Artificial Intelligence*. http://www.auai.org
- D. Koller, N. Friedman (2009) *Probabilistic Graphical Models*, The MIT Press
- A. Darwiche (2009) *Modeling and Reasoning with BNs*, Cambridge U.P.

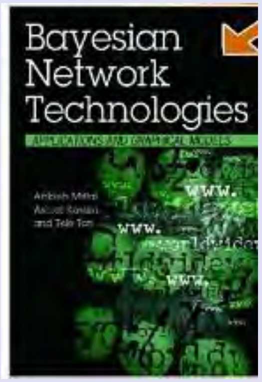
Books with applications

- Some in Neapolitan (2004)
- Many more in Mittal and Kassim (2007)
- ...and in Pourret et al. (2008)
- In Bioinformatics field, Neapolitan (2009)









Important groups/conferences

- European Worshop PGM (2002-)
- Uncertainty in AI (1985-)

Software

http://www.cs.ubc.ca/~murphyk/Bayes/bnsoft.html
http://www.cs.iit.edu/~mbilgic/classes/fall10/cs595/tools.html

www.hugin.com/

Software

www.bayesia.com

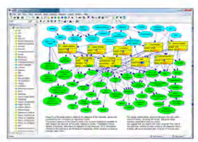
See "Applications overview"

Software

GeNIe at www.bayesfusion.com

BAYESFUSION, LLC

Data Analytics, Mathematical Modeling, Decision Support



GeNIe Modeler

GeNIe is a graphical user interface (GUI) to SMILE and allows for interactive model building and learning. It is written for the Windows environment but can be also used on Mac OS and Linux under Windows emulators. Learn more about GeNIe Modeler.

QGeNIe Modeler

QGeNIe is a rapid model development interface that allows for fast prototyping of decision models, useful especially in applications such as strategic planning. QGeNIe can be also used to develop rapidly the first, approximate version of a model that can be translated to GeNIe format for further refinement and development.

SMILE Engine

SMILE (Structural Modeling, Inference, and Learning Engine) is a reasoning engine for graphical models, such as Bayesian networks, influence diagrams, and structural equation models. Technically, it is a library of C++ classes that can be embedded into user applications. SMILE is fully portable and available for most computing platforms. We offer wrappers for SMILE that make it possible to use it from Java, .NET, and other development platforms.

SMILE Discovery Module

SMILE Learn is a model discovery module, allowing for learning Bayesian networks from data and causal discovery from within user's custom applications. Full functionality of SMILE Learn is accessible from GeNIe Modeler as well.

Software

code.google.com/p/bnt/

Bayes Net Toolbox for Matlab

Written by Kevin Murphy, 1997--2002. Last updated: 19 October 2007. As on January 2014, a copy of this is available at https://github.com/bayesnet/bnt

MATH BY MATLAB

- Major Features
- Examples of supported Models
- Download zip file
- Installation
- How to use the toolbox
- Subscribe to the BNT Email List
- Invited Paper on BNT published in Computing Science and Statistics, 2001.
- Other Bayes net software
- A brief introduction to Bayesian Networks
- Terms and conditions of use (GNU Library GPL)
- Why do I give the code away?
- Changelog
- Why MATLAB?
- Acknowledgements
- How do I contribute changes to the code?

Software

www.openmarkov.org/ (UNED)

OpenMarkov

An open source tool for Probabilistic Graphical Models

Software

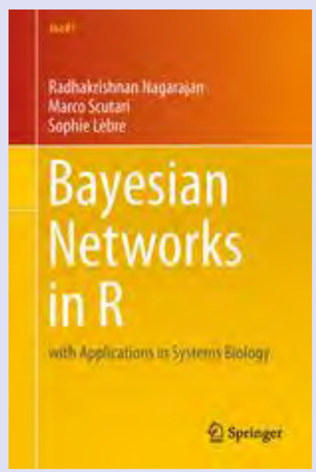
reasoning.cs.ucla.edu/samiam/

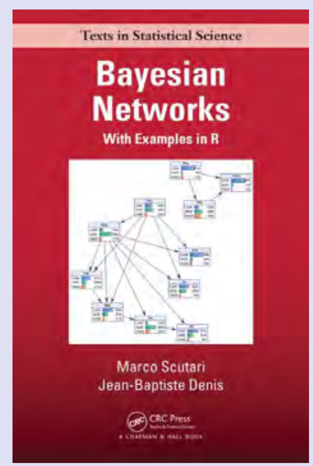
Software

www.r-project.org/

`bnlearn, deal, pcalg, catnet, mugnet, bnclassify` → learning

`gRbase, gRain` → inference





REDES BAYESIANAS: APRENDIZAJE, INFERENCIA Y APLICACIONES

Concha Bielza

Computational Intelligence Group
Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid

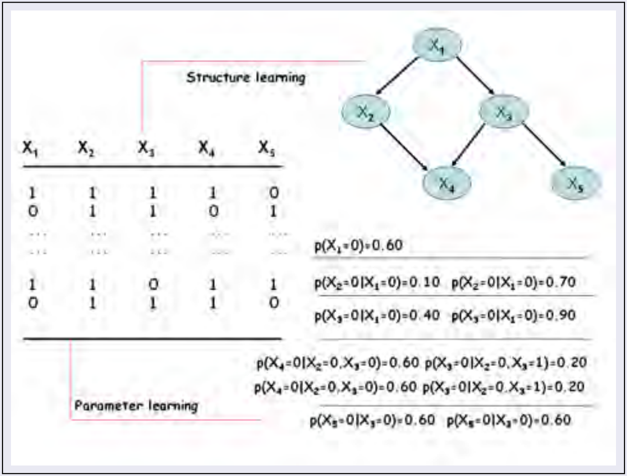
Madrid, 17 de junio de 2016

Outline

1. **Learning associations from data**
   - Learning parameters
   - Learning structures
2. **Bayesian classifiers**
   - From naive Bayes to multinets
   - Applications
3. **Conclusions**

From data to Bayesian networks

Learning structure and parameters



Outline

Maximum likelihood estimation of parameters

- $P(X_i = x_i^k \mid \boldsymbol{pa}_i^j) = \theta_{ijk}$, $i = 1, ..., n; j = 1, ..., q_i; k = 1, ..., r_i$
- N_{ij} number of cases in D where configuration $\boldsymbol{pa}_i^j$ has been observed
- N_{ijk} number of cases in D where simultaneously $X_i = x_i^k$ and $\boldsymbol{Pa}_i = \boldsymbol{pa}_i^j$ have been observed ($N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$)

$$\textit{likelihood} \quad L(D : \boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}$$

- For each variable X_i and configuration $\boldsymbol{pa}_i^j$ of $\boldsymbol{Pa}_i$

$$\widehat{\theta}_{ijk}^{\mathrm{ML}} = \frac{N_{ijk}}{N_{ij}}$$

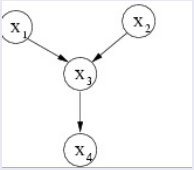
- Laplace estimator for sparse data ($N_{ij} = 0$, or unlikely $\mathbf{pa}_i^j$ or $X_i = x_i^k$)

$$\widehat{\theta}_{ijk}^{\mathrm{Lap}} = \frac{N_{ijk} + 1}{N_{ij} + r_i}$$

Maximum likelihood estimation of parameters

Parameters θ_{ijk}: example

Four variables: X_1, X_3 and X_4 with two possible values, and X_2 with three possible values



Local probabilities

$\boldsymbol{\theta}_1$	=	$(\theta_{1-1}, \theta_{1-2})$	$P(x_1^1)$, $P(x_1^2)$
$\boldsymbol{\theta}_2$	=	$(\theta_{2-1}, \theta_{2-2}, \theta_{2-3})$	$P(x_2^1)$, $P(x_2^2)$, $P(x_2^3)$
$\boldsymbol{\theta}_3$	=	$(\theta_{311}, \theta_{321}, \theta_{331},$	$P(x_3^1 \vert x_1^1, x_2^1)$, $P(x_3^1 \vert x_1^1, x_2^2)$, $P(x_3^1 \vert x_1^1, x_2^3)$,
		$\theta_{341}, \theta_{351}, \theta_{361},$	$P(x_3^1 \vert x_1^2, x_2^1)$, $P(x_3^1 \vert x_1^2, x_2^2)$, $P(x_3^1 \vert x_1^2, x_2^3)$,
		$\theta_{312}, \theta_{322}, \theta_{332},$	$P(x_3^2 \vert x_1^1, x_2^1)$, $P(x_3^2 \vert x_1^1, x_2^2)$, $P(x_3^2 \vert x_1^1, x_2^3)$,
		$\theta_{342}, \theta_{352}, \theta_{362})$	$P(x_3^2 \vert x_1^2, x_2^1)$, $P(x_3^2 \vert x_1^2, x_2^2)$, $P(x_3^1 \vert x_1^2, x_2^3)$,
$\boldsymbol{\theta}_4$	=	$(\theta_{411}, \theta_{421}, \theta_{412}, \theta_{422})$	$P(x_4^1 \vert x_3^1)$, $P(x_4^1 \vert x_3^2)$, $P(x_4^2 \vert x_3^1)$, $P(x_4^2 \vert x_3^2)$

Factorisation of the JPD:

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2)P(x_3|x_1, x_2)P(x_4|x_3)$$

variable X_i	possible values r_i	parent variables $\boldsymbol{Pa}_i$	possible values of the parents q_i
X_1	2	$\emptyset$	0
X_2	3	$\emptyset$	0
X_3	2	$\{X_1, X_2\}$	6
X_4	2	$\{X_3\}$	2

Bayesian estimation

- Parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$ are modeled with a random variable
- $f(\boldsymbol{\theta}|\mathcal{G})$: the prior about possible values of $\boldsymbol{\theta}$
- Posterior: $f(\boldsymbol{\theta}|\mathcal{D}, \mathcal{G}) \propto p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{G}) f(\boldsymbol{\theta}|\mathcal{G})$
- Summarize the posterior by using mean or mode (MAP):

$$\widehat{\boldsymbol{\theta}}^{\mathrm{Ba}} = \int \boldsymbol{\theta} f(\boldsymbol{\theta}|\mathcal{D}, \mathcal{G}) d\boldsymbol{\theta}, \qquad \widehat{\boldsymbol{\theta}}^{\mathrm{Ba}} = \arg \max_{\boldsymbol{\theta}} f(\boldsymbol{\theta}|\mathcal{D}, \mathcal{G})$$

- For parameters $\boldsymbol{\theta}_{ij} = (\theta_{ij1}, \ldots, \theta_{ijr_i})$, if $(\boldsymbol{\theta}_{ij}|\mathcal{G}) \sim Dir(\alpha_{ij1}, \ldots, \alpha_{ijr_i})$, then $(\boldsymbol{\theta}_{ij}|\mathcal{D}, \mathcal{G}) \sim Dir(\alpha_{ij1} + N_{ij1}, \ldots, \alpha_{ijR_i} + N_{ijr_i})$ and hence the posterior mean is

$$\widehat{\theta}_{ijk}^{\mathrm{Ba}} = \frac{N_{ijk} + \alpha_{ijk}}{N_{ij} + \alpha_{ij}}$$

  where $\alpha_{ij} = \sum_{k'=1}^{r_i} \alpha_{ijk'}$, called **equivalent sample size**
- Laplace estimates: a particular case of Bayesian estimation, with $\alpha_{ijk} = 1, \forall k$ (**flat Dirichlet**, equivalent to a uniform distribution)

Learning structures

Two types of methods

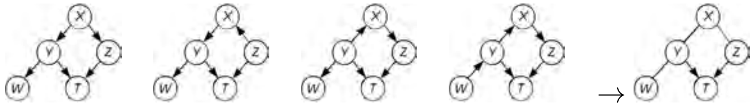
- Based on detecting conditional independencies (constrained-based methods)
  - First: study dependence/independence relationships among the variables by means of statistical tests
  - Second: try to find the structure (or structures) that represents the most (or all) of these relationships
- Based on score + search
  - They try to find the structure that best "fit" the data
  - They need:
    - A score (metric or evaluation function) in order to measure the goodness of each candidate structure
    - A search method (heuristic) to explore in an intelligent manner the space of possible solutions
    - Several types of spaces can be considered

Testing conditional independencies

PC algorithm (Spirtes et al. 1993)

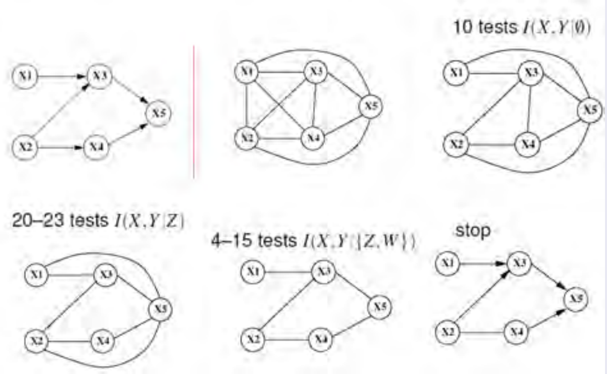
0) Start from the complete undirected graph
1) Produce the skeleton via edge elimination by hypothesis testing. If for some **S**, $I_p(X_i, X_j|\mathbf{S})$ holds, edge $X_i - X_j$ can be removed
   (c.i. $\leftrightarrow$ u-separ., is assumed)
2) Identify v-structures
3) Try to orient the edges to have the completed partially DAG
   (CPDAG or essential graph, the Markov equivalence class of DAGs)

Markov equivalent: Same skeleton, same v-structures (inmoralities)

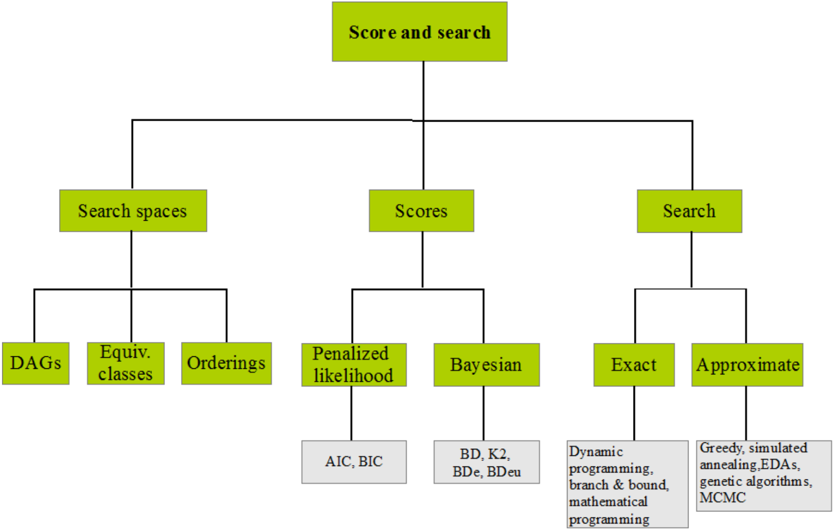


Testing conditional independencies

PC algorithm (Spirtes et al. 1993). Example with $t = 2$



Score+search approaches



Score+search approaches

Score metrics. Log-likelihood

- Log-likelihood of the data:

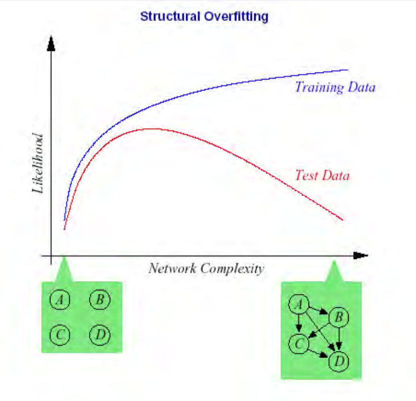
$$\log P(D : \mathcal{G}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \log(\theta_{ijk})^{N_{ijk}}$$

- Estimated log-likelihood:

$$\log P(D : \mathcal{G}, \widehat{\boldsymbol{\theta}}^{\mathrm{ML}}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}}$$

Score+search approaches

Score metrics. Log-likelihood



Likelihood of the data increases monotonically with the complexity of the model (structural overfitting)

Score+search approaches

Score metrics. Penalized log-likelihood

- Avoid overfitting penalizing the complexity of the BN in the log-likelihood :

$$\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}} - dim(\mathcal{G}) pen(N)$$

  - $dim(\mathcal{G}) = \sum_{i=1}^{n} q_i(r_i - 1)$, model dimension
  - $pen(N) \geq 0$, penalization function
    - $pen(N) = 1$: Akaike's information criterion (AIC)
    - $pen(N) = \frac{1}{2} \log N$: Bayesian information criterion (BIC). Its calculation is equivalent to the minimum description length (MDL) criterion

Score+search approaches

Score metrics. Bayesian approach

- Try to obtain the structure with maximum a posteriori probability given the data, that is, $\arg\max_{\mathcal{G}} P(\mathcal{G}|D)$
- Using Bayes' formula:

$$P(\mathcal{G}|D) \propto P(D|\mathcal{G})P(\mathcal{G})$$

- $P(\mathcal{G})$: the prior distribution over structures
- If $P(\mathcal{G})$ is uniform ($\max P(\mathcal{G}|D) \equiv \max P(D|\mathcal{G})$), i.e., the structure with maximum marginal likelihood
- $P(D|\mathcal{G})$: the marginal likelihood of the data
- $P(D|\mathcal{G}) = \int P(D|\mathcal{G}, \boldsymbol{\theta}) f(\boldsymbol{\theta}|\mathcal{G}) d\boldsymbol{\theta}$
  - $P(D|\mathcal{G}, \boldsymbol{\theta})$: likelihood of the data given the BN (structure + parameters)
  - $f(\boldsymbol{\theta}|\mathcal{G})$: prior distribution over the parameters

Score+search approaches

Score metrics. Bayesian approach: BD and K2 scores

- If $f(\theta|\mathcal{G})$ follows a Dirichlet distribution, we have a closed formula for $P(D|\mathcal{G})$

$$P(D|\mathcal{G}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

Bayesian Dirichlet (BD) score

- If $\alpha_{ijk} = 1, \forall i, j, k$ (flat Dirichlet or uniform distribution):

$$P(D|\mathcal{G}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!$$

K2 metric

Score+search approaches

K2 algorithm

- An ordering between the nodes is assumed
- An upper bound is set on the number of parents for any node
- For every node, X_i, K2 searches for the set of parent nodes that maximizes:

$$g(X_i, \boldsymbol{Pa}_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!$$

- K2 assumes initially that a node does not have parents
- At each step K2 incrementally adds the parent whose addition provides the best value for $g(X_i, \boldsymbol{Pa}_i)$
- K2 stops when adding a single parent to any node cannot increase $g(X_i, \boldsymbol{Pa}_i)$
- K2 is a greedy algorithm

Score+search approaches

Different spaces for the search

- Space of DAGs

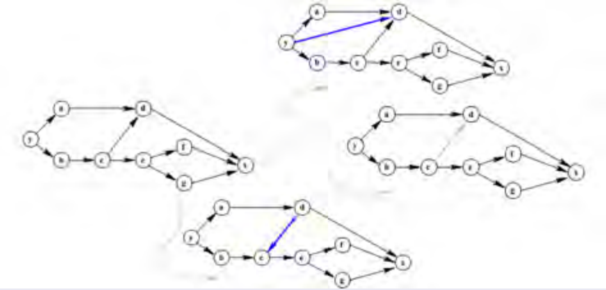
$$d(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} d(n-i); \quad d(0) = 1; \quad d(1) = 1$$

- Space of equivalence classes
  - # DAGs $\approx$ 3.7 # CPDAGs (moderate gain)
  - Scores: score equivalent
- Ordering between the variables: cardinality of the search space $n!$

Score+search approaches

Search algorithms. Local search. Algorithm B

- Local operators: add, remove and reverse an arc
- Efficient search due to the decomposability of the most usual metrics (AIC, BIC, BD, K2,...)



Search algorithms. Genetic algorithms

- Each individual represents a DAG structure (binary representation)

Outline

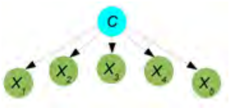
Supervised classification

Supervised: From labelled data to classification models

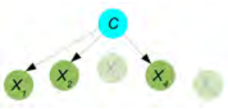
Predictor variables (attributes) and one labelled (class) variable:

	X_1	$\ldots$	X_n	C
$(\boldsymbol{x}^{(1)}, c^{(1)})$	$x_1^{(1)}$	$\ldots$	$x_n^{(1)}$	$c^{(1)}$
$(\boldsymbol{x}^{(2)}, c^{(2)})$	$x_1^{(2)}$	$\ldots$	$x_n^{(2)}$	$c^{(2)}$
$\ldots$		$\ldots$		$\ldots$
$(\boldsymbol{x}^{(N)}, c^{(N)})$	$x_1^{(N)}$	$\ldots$	$x_n^{(N)}$	$c^{(N)}$
$\boldsymbol{x}^{(N+1)}$	$x_1^{(N+1)}$	$\ldots$	$x_n^{(N+1)}$	???

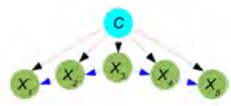
Different architectures



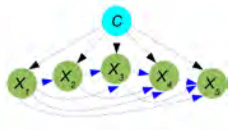
Naive Bayes



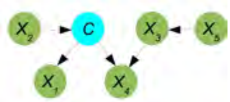
Selective naive Bayes



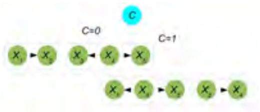
TAN



k-dependence

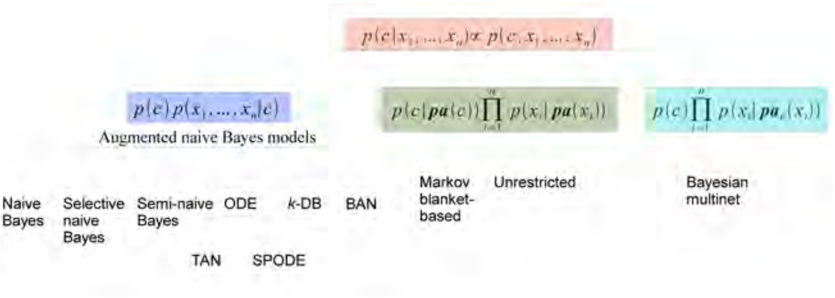


Unrestricted



Bayesian multinet

Different architectures

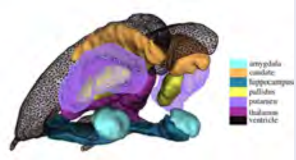


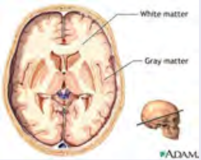
Bielza, Larrañaga (2014). Discrete Bayesian network classifiers: A survey. *ACM Computing Surveys* 47, 1, Article 5

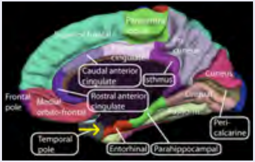
Outcome prediction after epilepsy surgery

Armañanzas, Alonso-Nanclares, DeFelipe-Oroquieta, Kastanauskaite, de Sola, DeFelipe, Bielza, Larrañaga (2013). Machine learning approach for the outcome prediction of temporal lobe epilepsy surgery. *PLoS ONE*, 8(4), e62819 (2009)

Dementia development in Parkinson’s disease

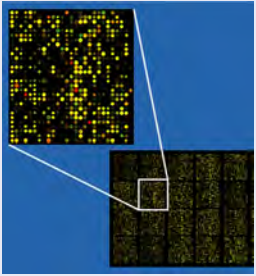


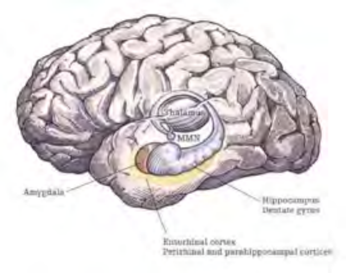




Morales, Vives-Gilabert, Gómez-Ansón, Bengoetxea, Larrañaga, Bielza, Pagonabarraga, Kulisevsky, Corcuera-Solano, Delfino (2012). Predicting dementia development in Parkinson’s disease using Bayesian network classifiers. *Psychiatry Research: NeuroImaging*, 213, 92-98

Alzheimer's disease and DNA microarrays

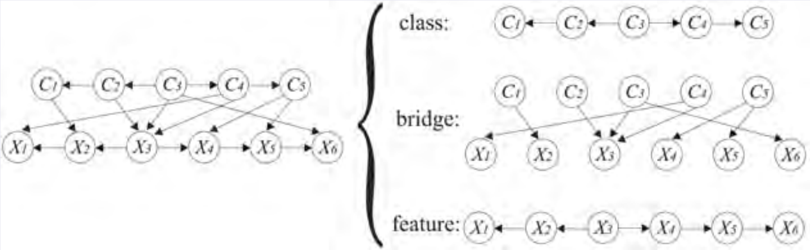




Armañanzas, Bielza, Larrañaga (2012). Ensemble transcript interaction networks: A case study on Alzheimer's disease. *Computer Methods and Programs in Biomedicine*, 108, 1, 442 - 450

Multi-dimensional classification with Bayesian networks

	X_1	...	X_m	C_1	...	C_d
$(\boldsymbol{x}^{(1)}, c^{(1)})$	$x_1^{(1)}$	...	$x_m^{(1)}$	$c_1^{(1)}$	...	$c_d^{(1)}$
$(\boldsymbol{x}^{(2)}, c^{(2)})$	$x_1^{(2)}$	...	$x_m^{(2)}$	$c_1^{(2)}$	...	$c_d^{(2)}$
...		...			...	
$(\boldsymbol{x}^{(N)}, c^{(N)})$	$x_1^{(N)}$	...	$x_m^{(N)}$	$c_1^{(N)}$	...	$c_d^{(N)}$
$\boldsymbol{x}^{(N+1)}$	$x_1^{(N+1)}$	...	$x_m^{(N+1)}$	???	...	???



Bielza, Li, Larrañaga (2011). Multi-dimensional classification with Bayesian networks. *International Journal of Approximate Reasoning*, 52, 705 - 727

Multi-dimensional classification for genotypic predictors of HIV type 1 drug resistance

Borchani, Bielza, Toro, Larrañaga (2013). Learning multi-dimensional Bayesian network classifiers using Markov blankets: A case study in the prediction of HIV-1 reverse transcriptase and protease inhibitors. *Artificial Intelligence in Medicine*, 57(3), 219-229

Multi-dimensional classification for EQ-5D health states from PDQ-39 in Parkinson's disease

PDQ-39

PDQ-39 captures patients perception of his illness covering 8 dimensions:

1. Mobility
2. Activities of daily living
3. Emotional well-being
4. Stigma
5. Social support
6. Cognitions
7. Communication
8. Bodily discomfort

Multi-dimensional classification for EQ-5D health states from PDQ-39 in Parkinson's disease

EQ-5D

EQ-5D is a generic measure of health for clinical and economic appraisal

Mobility
I have no problems in walking about
I have some problems in walking about
I am confined to bed

Self-care
I have no problems with self-care
I have some problems washing and dressing myself
I am unable to wash and dress myself

Usual activities (eg. work, study, housework, family or leisure activities)
I have no problems with performing my usual activities
I have some problems with performing my usual activities
I am unable to perform my usual activities

Pain/discomfort
I have no pain or discomfort
I have moderate pain or discomfort
I have extreme pain or discomfort

Anxiety/depression
I am not anxious or depressed
I am moderately anxious or depressed
I am extremely anxious or depressed

Multi-dimensional classification for EQ-5D health states from PDQ-39 in Parkinson's disease

Mapping PDQ-39 to EQ-5D

PDQ_1	PDQ_2	...	...	PDQ_{39}	EQ_1	EQ_2	EQ_3	EQ_4	EQ_5
3	1	...	...	3	1	3	3	2	1
2	3	...	...	2	1	1	2	3	2
5	2	...	...	4	1	3	3	1	2
...	...	...	...	...	...	...	...	...	...
4	4	...	...	3	3	1	2	3	2
4	4	...	...	3	3	1	2	3	2
5	5	...	...	4	2	3	2	3	3

$$h : (PDQ_1, ..., PDQ_{39}) \to (EQ_1, ..., EQ_5)$$

Borchani, Bielza, Martínez-Martín, Larrañaga (2012). Markov blanket-based approach for learning multi-dimensional Bayesian network classifiers: An application to predict the European quality of life-5Dimensions (EQ-5D) from the 39-item Parkinson's disease questionnaire (PDQ- 39), *Journal of Biomedical Informatics*, 45, 1175-1184

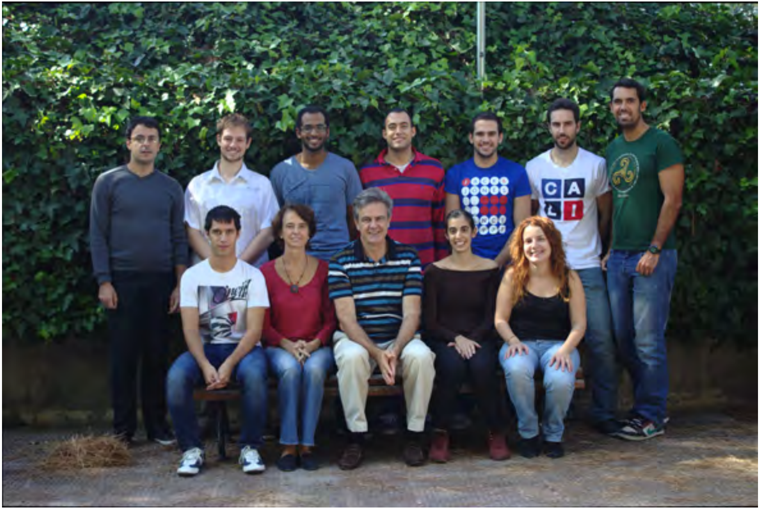
Outline

Conclusions

Bayesian networks and Bayesian classifiers

- Based on probability theory
- Theoretical properties
- Knowledge discovery
- Intuitive models
- Reasoning as inference propagation
- Simulation from the model
- Competitive results in accuracy

Acknowledgments



CIG members at UPM

REDES BAYESIANAS: APRENDIZAJE, INFERENCIA Y APLICACIONES

Concha Bielza

Computational Intelligence Group
Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid

Madrid, 17 de junio de 2016