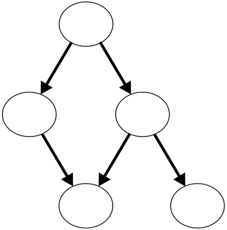
Bayesian Networks: Representation and Inference

Probabilistic Graphical Models

L. Enrique Sucar, INAOE



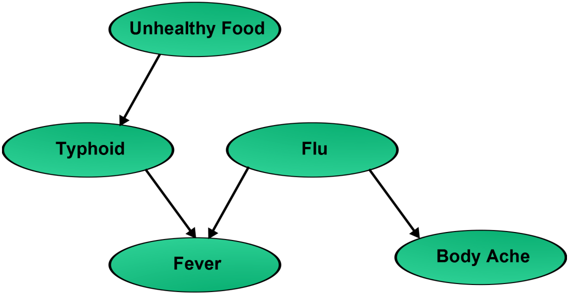
Outline

1. Introduction
2. Representation
   - Structure
   - Parameters
3. Inference
   - Probability propagation
4. References

Introduction

- Bayesian networks are directed graphical models that represent the joint distribution of a set of random variables
- In this graphs, the nodes represent random variables and the arcs direct dependencies between variables
- The structure of the graph encodes a set of conditional independence relations between the variables

Example



- *Fever* is independent of *Body ache* given *Flu* (common cause)
- *Fever* is independent of *Unhealthy food* given *Typhoid* (indirect cause)
- *Typhoid* is independent of *Flu* when *Fever* is NOT known (common effect). Knowing Fever makes Typhoid and Flu dependent

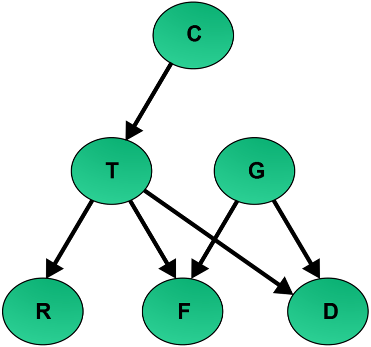
Introduction

- In addition to the structure, a Bayesian network considers a set of local parameters, which are the conditional probabilities for each variable given its parents in the graph
- The joint probability of all the variables in the network can be represented based on these local parameters; this usually implies an important saving in the number of required parameters
- Given a Bayesian network we can answer several probabilistic queries. For instance, for the previous example: What is the probability of Fever given Flu? Which is more probable, Typhoid or Flu, given Fever and Unhealthy food?

Bayesian Networks

- A Bayesian network (BN) represents the joint distribution of a set of n (discrete) variables, $X_1, X_2, \ldots, X_n$, as a directed acyclic graph (DAG) and a set of conditional probability tables (CPTs)
- Each node, that corresponds to a variable, has an associated CPT that contains the probability of each state of the variable given its parents in the graph
- The structure of the network implies a set of conditional independence assertions, which give power to this representation

An example



- Joint distribution:
  $P(C,T,G,R,F,D) =$
  $P(C)P(G)P(T \mid C)P(R \mid T)P(F \mid T,G)P(D \mid T,G)$

Conditional Independence Assertions

- The conditional independence assertions implied by the structure of a BN should correspond to the conditional independence relations of the joint probability distribution, and vice versa
- If X is conditionally independent of Z given Y:
  - In the probability distribution: $P(X|Y,Z) = P(X|Y)$.
  - In the graph: $I < X \mid Y \mid Z >$.

D-Separation

- Conditional independence assertions can be verified directly from the structure of a BN using a criteria called *D–separation*
- 3 basic BN structures for 3 variables and 2 arcs:
    - Sequential: $X \rightarrow Y \rightarrow Z$.
    - Divergent: $X \leftarrow Y \rightarrow Z$.
    - Convergent: $X \rightarrow Y \leftarrow Z$.
- In the first two cases, X and Z are conditionally independent given Y, however in the third case this is not true

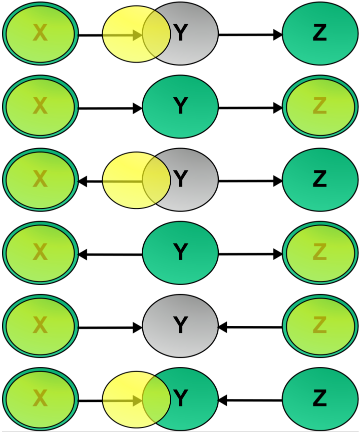
D-Separation

- Given a graph G, a set of variables A is conditionally independent of a set B given a set C, if there is no trajectory in G between A and B such that:
  1. All convergent nodes are or have descendants in C.
  2. All other nodes are outside C.

Bayes Ball

- Consider that we have a path from node X to Z with Y in the middle - Y is shaded if it is known (instantiated), otherwise it is not shaded
- We *throw a ball* from X to Z, if the ball arrives to Z then X and Z are NOT independent given Y:
  1. If Y is sequential or divergent and is not shaded, the ball goes through.
  2. If Y is sequential or divergent and it is shaded, the ball is blocked.
  3. If Y is convergent and not shaded, the ball is blocked.
  4. If Y is convergent and shaded, the ball goes through.

Bayes Ball



Contours

- *Markov assumption*: any node X is conditionally independent of all nodes in G that are not descendants of X given its parents in the graph, $Pa(X)$
- The structure of a BN can be specified by the parents of each variable; thus the set of parents of a variable X is known as the *contour* of X
- Given this condition and using the chain rule, we can specify the joint probability distribution of the set of variables in a BN as the product of the conditional probability of each variable given its parents

Markov Blanket

- The *Markov Blanket* of a node X, $MB(X)$, is the set of nodes that make it independent of all the other nodes in G, that is $P(X \mid G - X) = P(X \mid MB(X))$
- For a BN, the Markov blanket of X is:
  - the parents of X,
  - the sons of X,
  - and other parents of the sons of X.

Mappings

- Given a probability distribution P of $\mathbf{X}$, and its graphical representation G, there must be a correspondence between the conditional independence in P and in G - mappings:

  D-Map: all the conditional independence relations in P are satisfied (by D-Separation) in G.

  I-Map: all the conditional independence relations in G are true in P.

  P-Map: or perfect map, it is a D-Map and an I-Map.

- It is not always possible to have a *perfect mapping* of the independence relations between the graph (G) and the distribution (P), so we settle for what is called a *Minimal I–Map*: all the conditional independence relations implied by G are true in P, and if any arc is deleted in G this condition is lost

Independence Axioms

- Given some conditional independence relations between subsets of random variables, we can derive other conditional independence relations axiomatically
- *Independence axioms*:

Symmetry: $I(X, Z, Y) \rightarrow I(Y, Z, X)$

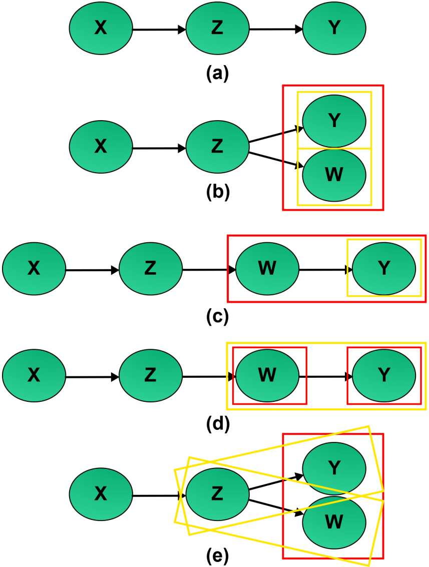
Decomposition:
$$I(X, Z, Y \cup W) \rightarrow I(X, Z, Y) \wedge I(X, Z, W)$$

Weak Union: $I(X, Z, Y \cup W) \rightarrow I(X, Z \cup W, Y)$

Contraction:
$$I(X, Z, Y) \wedge I(X, Z \cup Y, W) \rightarrow I(X, Z, Y \cup W)$$

Intersection: $I(X, Z \cup W, Y) \wedge I(X, Z \cup Y, W) \rightarrow$
$I(X, Z, Y \cup W)$

Graphically



(a)

(b)

(c)

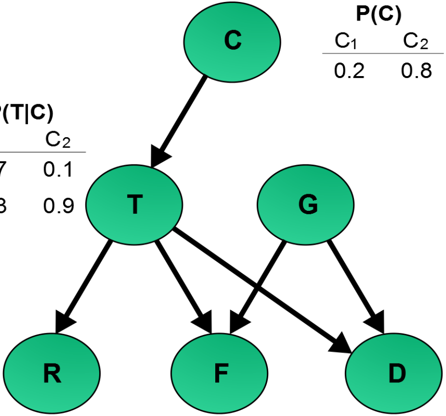
(d)

(e)

CPTs

- In the case of a BN, the parameters are the conditional probabilities of each node given its parents in the graph
- If we consider discrete variables:
    - Root nodes: vector of marginal probabilities.
    - Other nodes: conditional probability table (CPT) of the variable given its parents in the graph.

Example



P(C)

C_1	C_2
0.2	0.8

P(T|C)

	C_1	C_2
T_1	0.7	0.1
T_2	0.3	0.9

P(F|T,G)

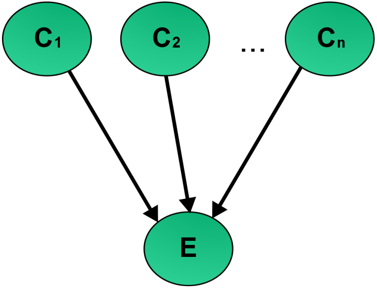
	T_1,F_1	T_1,F_2	T_2,F_1	T_2,F_2
F_1	0.8	0.6	0.5	0.1
F_2	0.2	0.4	0.5	0.9

Canonical Models

- Canonical models represent the relations between a set of random variables for particular interactions using few parameters
- There are several classes of canonical models, the most common are the *Noisy OR* and *Noisy AND* for binary variables, and their extensions for multivalued variables, *Noisy Max* and *Noisy Min*, respectively
- For example, consider a variable that represents a disease, D. In the case of the binary canonical models it has two values, *True* and *False*. For a multivalued model, it could be defined as $D \in \{False, Mild, Intermediate, Severe\}$, such that these values follow a predefined order

Noisy-OR

- The Noisy OR model is applied when several variables or *causes* can produce an *effect* if any one of them is *True*, and as more of the *causes* are true, the probability of the effect increases



Conditions

- The following two conditions must be satisfied for a Noisy OR canonical model to be applicable:

  Responsibility: the effect is false if all the possible causes are false.

  Independence of exceptions: if an effect is the manifestation of several causes, the mechanisms that inhibit the occurrence of the effect under one cause are independent of the mechanisms that inhibit it under the other causes.

- The probability that the effect E is inhibited (it does not occur) under cause C_i is defined as:

$$q_i = P(E = \mathit{False} \mid C_i = \mathit{True}) \qquad (1)$$

Parameters

- The parameters in the CPT for a Noisy OR model can be obtained using the following expressions when all the *m* causes are *True*:

$$P(E = False \mid C_1 = True, ...C_m = True) = \prod_{i=i}^{m} q_i \quad (2)$$

$$P(E = True \mid C_1 = True, ...C_m = True) = 1 - \prod_{i=i}^{m} q_i \quad (3)$$

- If *k* of *m* causes are *True*, then $P(E = False \mid C_1 = True, ...C_k = True) = \prod_{i=i}^{k} q_i$, so that if all the causes are *False* then the effect is *False* with a probability of one

Example

C_1	0	0	0	0	1	1	1	1
C_2	0	0	1	1	0	0	1	1
C_3	0	1	0	1	0	1	0	1
$P(E = 0)$	1	0.1	0.1	0.01	0.1	0.01	0.01	0.001
$P(E = 1)$	0	0.9	0.9	0.99	0.9	0.99	0.99	0.999

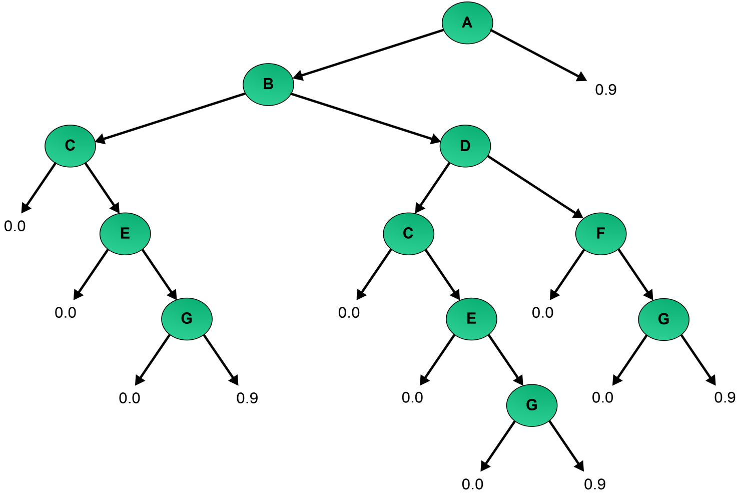
Decision Trees

- An alternative representation is based on the observation that in the probability tables for many domains, the same probability values tend to be repeated several times in the same table
- A *decision tree* (DT) could be used for representing a CPT in a compact way:
  Each internal node corresponds to a variable in the CPT, and the branches from a node correspond to the different values a variable can take. The leaf nodes in the tree represent the different probability values. A trajectory from the root to a leaf, specifies a probability value for the corresponding variables–values in the trajectory

Example - CPT

A	B	C	D	E	F	G	X
T	T/F	T/F	T/F	T/F	T/F	T/F	0.9
F	T	T/F	T	T/F	T	T	0.9
F	T	T/F	T	T/F	T	F	0.0
F	T	T/F	T	T/F	F	T/F	0.0
F	T	T	F	T	T/F	T	0.9
F	T	T	F	T	T/F	F	0.0
F	T	T	F	F	T/F	T/F	0.0
F	T	F	F	T/F	T/F	T/F	0.0
F	F	T	T/F	T	T/F	T	0.9
F	F	T	T/F	T	T/F	F	0.0
F	F	T	T/F	F	T/F	T/F	0.0
F	F	F	T/F	T/F	T/F	T/F	0.0

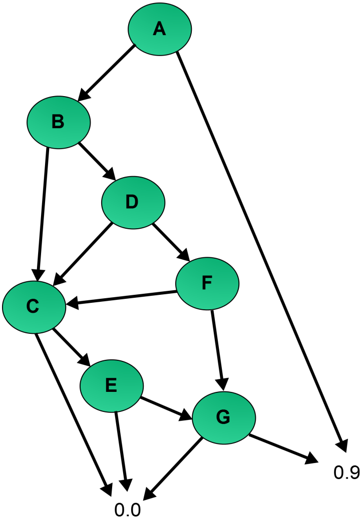
Example - DT



Decision Diagram

- A *decision diagram* (DD) extends a DT by considering a directed acyclic graph structure, such that it is not restricted to a tree
- This avoids the need to duplicate repeated probability values in the leaf nodes, and in some cases provides an even more compact representation

Example - DD



Probabilistic inference

- Probabilistic inference consists in *propagating* the effects of certain evidence in a Bayesian network to estimate its effect on the unknown variables
- There are basically two variants of the inference problem in BNs:
    - *Single query inference*: obtaining the posterior probability of a single variable, H, given a subset of known (instantiated) variables, $\mathbf{E}$, that is, $P(H \mid \mathbf{E})$
    - *Conjunctive query inference*: consists in calculating the posterior probability of a set of variables, $\mathbf{H}$ given the evidence, $\mathbf{E}$, that is, $P(\mathbf{H} \mid \mathbf{E})$

Inference algorithms:

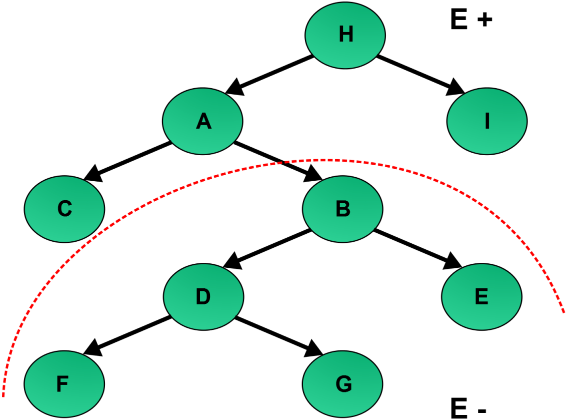
1. Probability propagation (Pearl's algorithm).
2. Variable elimination.
3. Conditioning.
4. Junction tree.
5. Stochastic simulation.

Complexity

- In the worst case the inference problem is *NP-hard* for Bayesian networks
- There are efficient (polynomial) algorithms for certain types of structures (singly connected networks)
- For other structures it depends on the connectivity of the graph.
- In many applications, the graphs are *sparse* and in this case there are inference algorithms which are very efficient

Probability propagation in trees

- Given that the BN has a tree structure, any node divides the network into two independent subtrees



Basic equations

- Given certain evidence, **E** (subset of instantiated variables), the posterior probability for a value *i* of any variable *B*, can be obtained by applying the Bayes rule:

$$P(Bi|\mathbf{E}) = P(Bi)P(\mathbf{E}|Bi)/P(\mathbf{E}) \tag{4}$$

- We can separate the evidence into:

  E-: Evidence in the tree rooted in *B*.
  E+: All other evidence.

- Then:

$$P(Bi|\mathbf{E}) = P(Bi)P(\mathbf{E}-, \mathbf{E}+|Bi)/P(\mathbf{E}) \tag{5}$$

Basic equations

- Given that **E**$+$ and **E**$-$ are independent, by applying the Bayes rule again, we obtain:

$$P(Bi|\mathbf{E}) = \alpha P(Bi|\mathbf{E}+)P(\mathbf{E}-|Bi) \tag{6}$$

Where α is a normalization constant.

- We define the following terms:

$$\lambda(Bi) = P(\mathbf{E}-|Bi) \tag{7}$$

$$\pi(Bi) = P(Bi|\mathbf{E}+) \tag{8}$$

- Then:

$$P(Bi|\mathbf{E}) = \alpha\pi(Bi)\lambda(Bi) \tag{9}$$

Propagation algorithm

- The computation of the posterior probability of any node *B* is decomposed into two parts: (i) the evidence coming from the sons of *B* in the tree (λ), and the evidence coming from the parent of *B*, (π)
- We can think of each node *B* in the tree as a simple processor that stores its vectors $\pi(B)$ and $\lambda(B)$, and its conditional probability table, $P(B \mid A)$
- The evidence is propagated via a message passing mechanism, in which each node sends the corresponding messages to its parent and sons in the tree

Messages

- A message sent from node B to its parent A:

$$\lambda_B(Ai) = \sum_j P(B_j \mid A_i)\lambda(B_j) \tag{10}$$

- A message sent from node B to its son S_k:

$$\pi_k(Bi) = \alpha\pi(B_j)\prod_{l \neq k}\lambda_l(B_j) \tag{11}$$

where l refers to each one of the sons of B

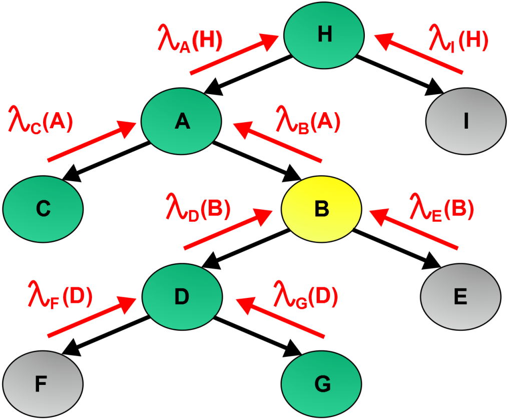
Combination and Propagation

- Each node can receive several λ messages, which are combined via a term by term multiplication for the λ messages received from each son:

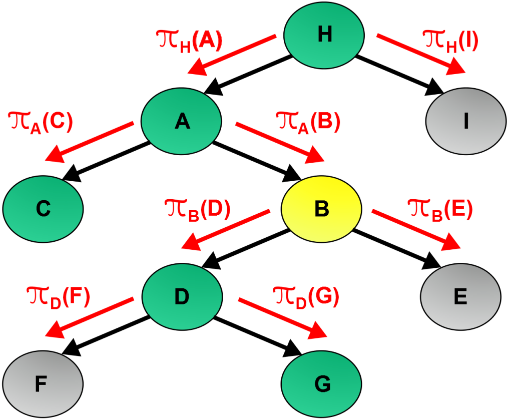
$$\lambda(Ai) = \prod_{j=1}^{m} \lambda_{Sj}(Ai) \tag{12}$$

- The propagation algorithm starts by assigning the evidence to the known variables, and then propagating it through the message passing mechanism until the root of the tree is reached for the λ messages, and the leaves are reached for the π messages

Bottom-up propagation



Top-down propagation

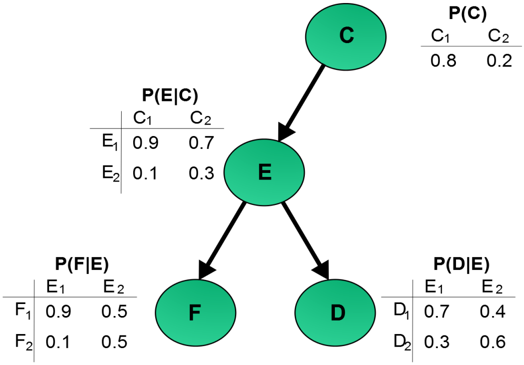


Initial Conditions

Leaf nodes: If not known, $\lambda = [1, 1, ..., 1]$ (a uniform distribution). If known, $\lambda = [0, 0, ..., 1, ..., 0]$ (one for the assigned value and zero for all other values).

Root node: If not known, $\pi = P(A)$ (prior marginal probability vector). If known, $\pi = [0, 0, ..., 1, ..., 0]$ (one for the assigned value and zero for all other values).

Propagation - example



P(C)

C_1	C_2
0.8	0.2

P(E|C)

	C_1	C_2
E_1	0.9	0.7
E_2	0.1	0.3

P(F|E)

	E_1	E_2
F_1	0.9	0.5
F_2	0.1	0.5

P(D|E)

	E_1	E_2
D_1	0.7	0.4
D_2	0.3	0.6

- Consider that the only evidence is $F = \textit{false}$ - initial conditions for the leaf nodes are: $\lambda_F = [1, 0]$ and $\lambda_D = [1, 1]$ (no evidence)

Example - λ propagation

- Multiplying the λ vectors by the corresponding CPTs:

$$\lambda_F(E) = [1,0][\begin{array}{c} 0.9, 0.5 \\ 0.1, 0.5 \end{array}] = [0.9, 0.5]$$

$$\lambda_D(E) = [1,1][\begin{array}{c} 0.7, 0.4 \\ 0.3, 0.6 \end{array}] = [1, 1]$$

- Then, $\lambda(E)$ is obtained by combining the messages from its two sons:

$$\lambda(E) = [0.9, 0.5] \times [1, 1] = [0.9, 0.5]$$

- Propagation to its parent, C:

$$\lambda_E(C) = [0.9, 0.5][\begin{array}{c} 0.9, 0.7 \\ 0.1, 0.3 \end{array}] = [0.86, 0.78]$$

Example - π propagation

- Given that C is not instantiated, $\pi(C) = [0.8, 0.2]$
- Propagate to its son, E, which also corresponds to multiplying the π vector by the corresponding CPT:

$$\pi(E) = [0.8, 0.2][\begin{array}{c} 0.9, 0.7 \\ 0.1, 0.3 \end{array}] = [0.86, 0.14]$$

- We now propagate to its son D; however, given that E has another son, F, we also need to consider the λ message from this other son, thus:

$$\pi(D) = [0.86, 0.14] \times [0.9, 0.5][\begin{array}{c} 0.7, 0.4 \\ 0.3, 0.6 \end{array}] = [0.57, 0.27]$$

Example - posterior probabilities

- Given the λ and π vectors for each unknown variable, we just multiply them term by term and then normalize to obtain the posterior probabilities:

$$P(C) = [0.86, 0.2] \times [0.86, 0.78] = \alpha[0.69, 0.16]$$

$$= [0.815, 0.185]$$

$$P(E) = [0.86, 0.14] \times [0.9, 0.5] = \alpha[0.77, 0.07]$$

$$= [0.917, 0.083]$$

$$P(D) = [0.57, 0.27] \times [1, 1] = \alpha[0.57, 0.27] = [0.67, 0.33]$$

Analysis

- The time complexity to obtain the posterior probability of all the variables in the tree is proportional to the *diameter* of the network (the number of arcs in the trajectory from the root to the most distant leaf).
- The message passing mechanism can be directly extended to polytrees, as these are also singly connected networks. In this case, a node can have multiple parents, so the λ messages should be sent from a node to all its parents
- The propagation algorithm only applies to singly connected network

Book

Sucar, L. E, *Probabilistic Graphical Models*, Springer 2015 – Chapter 7

Additional Reading (1)

- Cooper, G.F.: The Computational Complexity of Probabilistic Inference Using Bayesian Networks. Artificial Intelligence. 42, 393–405 (1990)
- Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press, New York (2009)
- Díez, F.J., Druzdzel, M.J.: Canonical Probabilistic Models for Knowledge Engineering. Technical Report CISIAD-06-01. Universidad Nacional de Educación a Distancia, Spain (2007)
- Neapolitan, R. E.: Probabilistic Reasoning in Expert Systems. John Wiley & Sons, New York (1990)
- Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)