M.I. Jaime Alfonso Reyes ´Cortés

BAYESIAN NETWORKS (CONT.)

INFERENCE IN BAYESIAN NETWORKS

- The basic task for any probabilistic inference system is to compute the posterior probability distribution for a set of query nodes, given values for some evidence nodes.
- This task is called **belief updating or probabilistic inference.**

EXACT INFERENCE IN CHAINS

- Two node network
- A two node network $X \rightarrow Y$
- If there is evidence about the parent node, say X = x, then the posterior probability (or belief) for Y can be read straight from the value in CPT $P(Y|X = x)$
- If there is evidence about the child node, say Y = y, then the inference task of updating the belief for X is done using

$$
\begin{aligned}
Bel(X = x) &= P(X = x|Y = y) \\
&= \frac{P(Y = y|X = x)P(X = x)}{P(Y = y)} \\
&= \alpha P(x)\lambda(x)
\end{aligned}
$$

where

$$\alpha = \frac{1}{P(Y = y)}$$

- *P*(*x*) is the prior and $\lambda(x) = P(Y = y|\ X = x)$ is the **likelihood**
- Note that we don't need to know the prior for the evidence. Since the beliefs for all the values of X must sum to one (due to the Total Probability) , we can compute α as a **normalizing constant**

- Three node chain $X \to Y \to Z$
- If we have evidence about the root node, X=x, updating in the same direction as the arcs involves the simple application of the chain rule, using the independencies implied in the network

$$Bel(Z) = P(Z|X = x) = \sum_{Y=y} P(Z|Y)P(Y|X = x)$$

$$P(C|A) = P(C|B)P(B|A) + P(C|\neg B)P(\neg B|A)$$

- If we have evidence about the leaf node, Z=z, the diagnostic inference to obtain Bel(X) is done with the application of Bayes' Theorem and the chain rule

$$
\begin{aligned}
Bel(X=x) &= P(X=x|Z=z) \\
&= \frac{P(Z=z|X=x)P(X=x)}{P(Z=z)} \\
&= \frac{\sum_{Y=y} P(Z=z|Y=y,X=x)P(Y=y|X=x)P(X=x)}{P(Z=z)} \\
&= \frac{\sum_{Y=y} P(Z=z|Y=y)P(Y=y|X=x)P(X=x)}{P(Z=z)} \quad (Z \perp\!\!\!\perp X|Y) \\
&= \alpha P(x)\lambda(x)
\end{aligned}
$$

where

$$\lambda(x) = P(Z=z|X=x) = \sum_{Y=y} P(Z=z|Y=y)P(Y=y|X=x)$$

- **Exact inference in polytrees**
- **polytree (or "forest")**
- Polytrees have at most one path between any pair of nodes; hence they are also referred to as **singly-connected networks**
- Assume X is the query node, and there is some set of evidence nodes **E** (not including X)
- The task is to update Bel(X) by computing $P(X|\mathbf{E})$
- The local belief updating for X must incorporate evidence from all other parts of the network

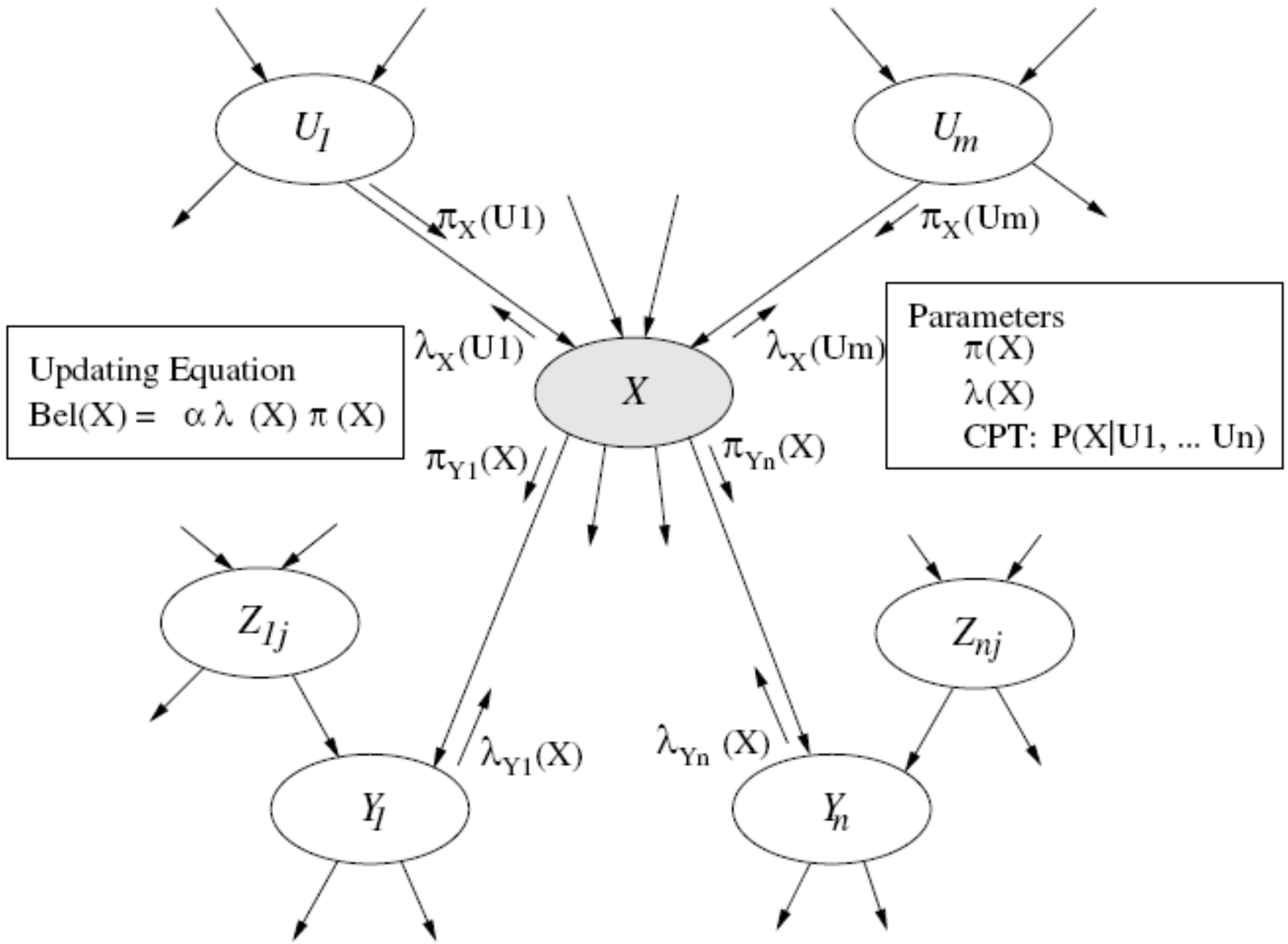


FIGURE 3.1

A generic polytree showing how local belief updating of node X is achieved through incorporation of evidence through its parents (the U_i) and children (the Y_j). Also shown are the message passing parameters and messages.

- evidence can be divided into:

- The **predictive support** for X, from evidence nodes connected to X through its parents, $U_1, \ldots U_m$; and
- The **diagnostic support** for X, from evidence nodes connected to X through its children $Y_1, \ldots Y_n$.

ALGORITHM 3.1

Kim and Pearl's Message Passing Algorithm

This algorithm requires the following three types of parameters to be maintained.

- *The current strength of the predictive support* π *contributed by each incoming link* $U_i \to X$*:*

  $\pi_X(U_i) = P(U_i | E_{U_i \setminus X})$

  where $E_{U_i \setminus X}$ *is all evidence connected to* U_i *except via* X.
- *The current strength of the diagnostic support* λ *contributed by each outgoing link* $X \to Y_j$*:*

  $\lambda_{Y_j}(X) = P(E_{Y_j \setminus X} | X)$

  where $E_{Y_j \setminus X}$ *is all evidence connected to* Y_j *through its parents except via* X.
- *The fixed CPT* $P(X | U_i, \ldots, U_n)$ *(relating* X *to its immediate parents).*

These parameters are used to do local belief updating in the following three steps, which can be done in any order.

(Note: in this algorithm, x_i *means the* i*th state of node* X*, while* $u_1 \ldots u_n$ *is used to represent an instantiation of the parents of* X*,* $U_1 \ldots U_n$*, in the situations where there is a summation of all possible instantiations.)*

1. **Belief updating**.

Belief updating for a node X is activated by messages arriving from either children or parent nodes, indicating changes in their belief parameters.

When node X is activated, inspect $\pi_X(U_i)$ (messages from parents), $\lambda_{Y_j}(X)$ (messages from children). Apply with

$$Bel(x_i) = \alpha\lambda(x_i)\pi(x_i) \tag{3.1}$$

where,

$$\lambda(x_i) = \begin{cases} 1 & \text{if evidence is } X = x_i \\ 0 & \text{if evidence is for another } x_j \\ \prod_j \lambda_{Y_j}(x_i) & \text{otherwise} \end{cases} \tag{3.2}$$

$$\pi(x_i) = \sum_{u_1,\ldots,u_n} P(x_i|u_1,\ldots,u_n) \prod_i \pi_X(u_i) \tag{3.3}$$

and α is a normalizing constant rendering $\sum_{x_i} Bel(X = x_i) = 1$.

2. **Bottom-up propagation**.

Node X computes new λ messages to send to its parents.

$$\lambda_X(u_i) = \sum_{x_i} \lambda(x_i) \sum_{u_k : k \neq i} P(x_i | u_1, \ldots, u_n) \prod_{k \neq i} \pi_X(u_k) \tag{3.4}$$

3. **Top-down propagation**.

Node X computes new π messages to send to its children.

$$\pi_{Y_j}(x_i) = \begin{cases} 1 & \text{if evidence value } x_i \text{ is entered} \\ 0 & \text{if evidence is for another value } x_j \\ \alpha[\prod_{k \neq j} \lambda_{Y_k}(x_i)] \sum_{u_1,\ldots,u_n} P(x_i|u_1,\ldots,u_n) \prod_i \pi_X(u_i) \\ = \frac{\alpha Bel(x_i)}{\lambda_{Y_j}(x_i)} \end{cases} \tag{3.5}$$

First, equation (3.2) shows how to compute the $\lambda(x_i)$ parameter. Evidence is entered through this parameter, so it is 1 if x_i is the evidence value, 0 if the evidence is for some other value x_j, and is the product of all the λ messages received from its children if there is no evidence entered for X. The $\pi(x_i)$ parameter (3.3) is the product of the CPT and the π messages from parents.

The λ message to one parent combines (i) information that has come from children via λ messages and been summarized in the $\lambda(X)$ parameter, (ii) the values in the CPT and (iii) any π messages that have been received from any other parents.

The $\pi_{Y_j}(x_i)$ message down to child Y_j is 1 if x_i is the evidence value and 0 if the evidence is for some other value x_j. If no evidence is entered for X, then it combines (i) information from children other than Y_j, (ii) the CPT and (iii) the π messages it has received from its parents.

The algorithm requires the following initializations (i.e., before any evidence is entered).

- Set all λ values, λ messages and π messages to 1.
- Root nodes: If node W has no parents, set $\pi(W)$ to the prior, $P(W)$.

The message passing algorithm can be used to compute the beliefs for all nodes in the network, even before any evidence is available.

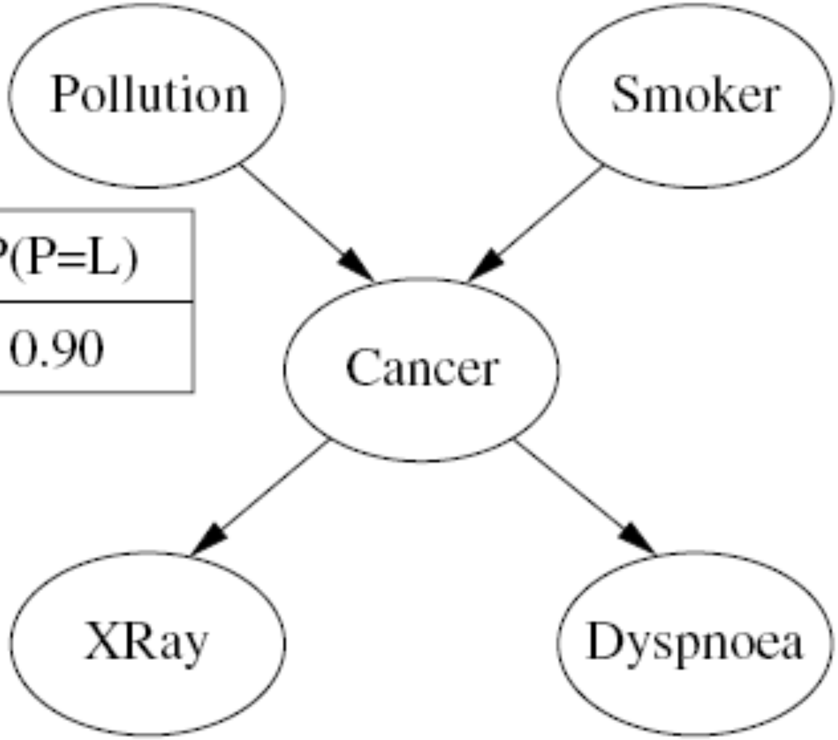
When specific evidence $W = w_i$ is obtained, given that node W can take values $\{w_1, w_2, \ldots, w_n\}$,

- Set $\lambda(W) = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$ with the 1 at the ith position.

This π/λ notation used for the messages is that introduced by Kim and Pearl and can appear confusing at first. Note that the format for both types of messages is $\pi_{Child}(Parent)$ and $\lambda_{Child}(Parent)$. So,

- π messages are sent in the direction of the arc, from parent to child, hence the notation is $\pi_{Receiver}(Sender)$;
- λ messages are sent from child to parent, against the direction of the arc, hence the notation is $\lambda_{Sender}(Receiver)$.

EJEMPLO



P(S=T)
0.30

P(P=L)
0.90

P	S	P(C=T|P,S)
H	T	0.05
H	F	0.02
L	T	0.03
L	F	0.001

C	P(X=pos|C)
T	0.90
F	0.20

C	P(D=T|C)
T	0.65
F	0.30

Node P(S)=0.3	No Evidence	Reasoning Case Diagnostic D=T	Predictive S=T	Intercausal C=T	C=T S=T	Combined D=T S=T
Bel(P=high)	0.100	0.102	0.100	0.249	0.156	0.102
Bel(S=T)	0.300	0.307	1	0.825	1	1
Bel(C=T)	0.011	0.025	0.032	1	1	0.067
Bel(X=pos)	0.208	0.217	0.222	0.900	0.900	0.247
Bel(D=T)	0.304	1	0.311	0.650	0.650	1
P(S)=0.5						
Bel(P=high)	0.100	0.102	0.100	0.201	0.156	0.102
Bel(S=T)	0.500	0.508	1	0.917	1	1
Bel(C=T)	0.174	0.037	0.032	1	1	0.067
Bel(X=pos)	0.212	0.226	0.311	0.900	0.900	0.247
Bel(D=T)	0.306	1	0.222	0.650	0.650	1

REFERENCIAS

- Kevin B. Korb, Ann E. Nicholson. Bayesian Artificial Intelligence, CHAPMAN & HALL/CRC. England, 2004.