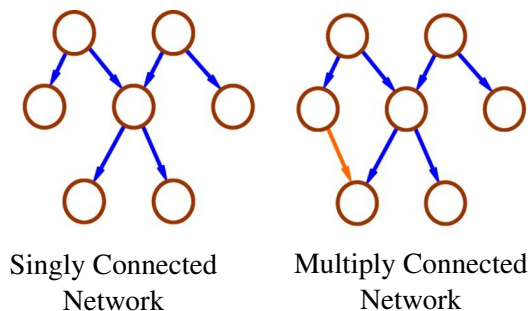


Lecture 4: Probability Propagation in Singly Connected Bayesian Networks

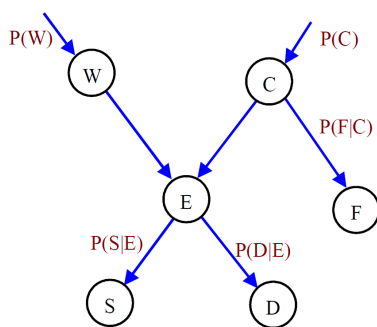
Singly Connected Networks

So far we have looked at several cases of probability propagation in networks. We now want to formalise the methods to produce a general purpose algorithm. We will restrict the treatment to singly connected networks, which are those that have at most one (undirected) path between two variables. The reason for this, which will become obvious in due course, is to ensure that the algorithm will terminate. As a further restriction, we will allow nodes to have at most two parents. This is primarily to keep the algebra simple. It is possible to extend the treatment to cope with any number of parents.



Multiple Parents

Up until now our networks have been trees, but in general they need not be. In particular, we need to cope with the possibility of multiple parents. Multiple parents can be thought of as representing different possible causes of an outcome. For example, the eyes in a picture could be caused by other image features, such as other animals with similarly shaped eyes, or by other objects entirely sharing the same geometric model that we are using. Suppose in a given application we have a data base containing pictures of owls and cats. The intrinsic characteristics of the eyes may be similar, and so there could be two causes for the E variable.



Unfortunately, with multiple parents our conditional probabilities become more complex. For the eye node we have to consider the conditional distribution:

$$P(E|W\&C)$$

which is a conditional probability table with an individual probability value for each possible combination of c_i & w_j and each e_k . The link matrix for multiple parents now takes the form:

$$P(E|W\&C) = \begin{bmatrix} P(e_1|w_1\&c_1) & P(e_1|w_1\&c_2) & P(e_1|w_2\&c_1) & P(e_1|w_2\&c_2) \\ P(e_2|w_1\&c_1) & P(e_2|w_1\&c_2) & P(e_2|w_2\&c_1) & P(e_2|w_2\&c_2) \\ P(e_3|w_1\&c_1) & P(e_3|w_1\&c_2) & P(e_3|w_2\&c_1) & P(e_3|w_2\&c_2) \end{bmatrix}$$

We would, of course not expect any cases to occur when both W and C were true in this particular example. However, we can still treat C and W as being independent events for which the special case $w_1\&c_1$, implying both an owl and a cat never occurs. Assuming that $w_1\&c_1$ never occurs in our data we need to set the values: $P(e_1|w_1\&c_1)$, $P(e_2|w_1\&c_1)$ and $P(e_3|w_1\&c_1)$ equal to 1/3, though this is not important as they will never affect any λ or π message.

π messages

We noted in the last lecture that it is important to ensure that when we pass evidence around a tree we do not count it more than once. When we are calculating the π messages that will be sent to E from C and W, we

do not use the λ messages that E has sent to C and W . Thus for example if $\pi_E(C)$ is the π message that C sends to E , it only takes into account the prior evidence for C and the λ message sent to C from the node F . We can see that it would be wrong to use the λ message from E . If we did so we would include it twice in calculating the overall evidence for E . This would bias our decision in favour (in this case) of the nodes S and D . Of course, if we want to calculate the probability of an Owl or a Cat, using all the evidence available then we need to use the λ evidence from E to calculate the final probability distribution over W and/or C . In order to make it quite clear we will reiterate following notational conventions introduced in the last lectures.

- $P'(C)$ will be used only for the final probability distribution over node C after all evidence has been propagated.
- $\pi_E(C)$ is the evidence for node C excluding any λ evidence from E . This is called the π message from C to E .

Remembering that the π messages need not be normalised, we can write the scalar equations:

$$P'(C) = \alpha \pi_E(C) \lambda_E(C)$$

$$\pi_E(C) = P'(C) / \lambda_E(C)$$

In order to send a π message in the case of a multiple parent we must calculate a joint distribution of the evidence over C & W . To do this we assume that variables C and W are independent so that we can write:

$$\pi_E(W \& C) = \pi_E(W) \times \pi_E(C)$$

which is a scalar equation with variables C and W . This independence assumption is fine in the case of singly connected networks (those without loops) since there cannot be any other path between C and W through which evidence can be propagated. However, if there was, for example a common ancestor of W and C the assumption would no longer hold. We will discuss this point further later in the course.

The joint evidence over C and W can be written as a vector of dimension four.

$$\begin{aligned} \pi_E(W \& C) &= [\pi_E(w_1 \& c_1), \pi_E(w_1 \& c_2), \pi_E(w_2 \& c_1), \pi_E(w_2 \& c_2)] \\ &= [\pi_E(w_1)\pi_E(c_1), \pi_E(w_1)\pi_E(c_2), \pi_E(w_2)\pi_E(c_1), \pi_E(w_2)\pi_E(c_2)] \end{aligned}$$

And the π evidence for the node E can be found by multiplying this joint distribution by the link matrix:

$$[\pi(e_1) \ \pi(e_2) \ \pi(e_3)] = \begin{bmatrix} P(e_1|w_1 \& c_1) & P(e_1|w_1 \& c_2) & P(e_1|w_2 \& c_1) & P(e_1|w_2 \& c_2) \\ P(e_2|w_1 \& c_1) & P(e_2|w_1 \& c_2) & P(e_2|w_2 \& c_1) & P(e_2|w_2 \& c_2) \\ P(e_3|w_1 \& c_1) & P(e_3|w_1 \& c_2) & P(e_3|w_2 \& c_1) & P(e_3|w_2 \& c_2) \end{bmatrix} \begin{bmatrix} \pi_E(w_1 \& c_1) \\ \pi_E(w_1 \& c_2) \\ \pi_E(w_2 \& c_1) \\ \pi_E(w_2 \& c_2) \end{bmatrix}$$

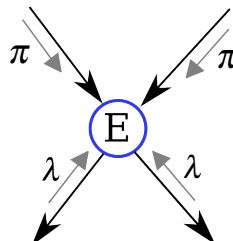
or in vector form:

$$\pi(E) = P(E|W \& C) \pi_E(W \& C)$$

We now can finally compute a probability distribution over the states of E . As before all we do is multiply the λ and π evidence together and normalise.

$$P'(e_i) = \alpha \lambda(e_i) \pi(e_i)$$

α is the normalising constant.



λ messages

Calculating λ evidence to send to multiple parents is harder than the single parent case. Before we introduced the W node the λ message from node E was calculated as:

$$\begin{aligned}\lambda_E(c_1) &= \lambda(e_1)P(e_1|c_1) + \lambda(e_2)P(e_2|c_1) + \lambda(e_3)P(e_3|c_1) \\ \lambda_E(c_2) &= \lambda(e_1)P(e_1|c_2) + \lambda(e_2)P(e_2|c_2) + \lambda(e_3)P(e_3|c_2)\end{aligned}$$

we can write this as a scalar equation parameterised by i , the state of C

$$\lambda_E(c_i) = \sum_j P(e_j|c_i)\lambda(e_j)$$

The use of the subscript indicates that λ_E is the part of the λ evidence that comes from node E . It is called the λ message from E to C . The full λ evidence for C takes into account node F as well: ($\lambda(C) = \lambda_E(C)\lambda_F(C)$). We also found it convenient to write the λ message in vector form:

$$\lambda_E(C) = \lambda(E)P(E|C)$$

However, things are now more complex, since we no longer have a simple link matrix $P(E|C)$, and the λ message to C needs to take into account the node W .

One way to deal with the problem is to calculate a joint λ message for $W \& C$. We can do this using a similar equation to the vector form given above:

$$\lambda_E(W \& C) = \lambda(E)P(E|W \& C)$$

If we do this we have a λ message for each joint state of C and W :

$$\lambda_E(W \& C) = [\lambda(w_1 \& c_1), \lambda(w_1 \& c_2), \lambda(w_2 \& c_1), \lambda(w_2 \& c_2)]$$

So we now need to use the evidence for W to separate out the evidence for C from the joint λ message:

$$\lambda_E(c_i) = \sum_j \pi_E(w_j)\lambda(w_j \& c_i)$$

The identical computation can be made estimating the link matrix $P(E|C)$, from $P(E|W \& C)$ given the evidence for W excluding E ($\pi_E(W)$). This estimate is only valid given the current state of the network. Each element of the matrix can be computed using the scalar equation:

$$P(e_i|c_j) = P(e_i|c_j \& w_1)\pi_E(w_1) + P(e_i|c_j \& w_2)\pi_E(w_2)$$

or more generally:

$$P(e_i|c_j) = \sum_k P(e_i|c_j \& w_k)\pi_E(w_k)$$

where k ranges over the states of W .

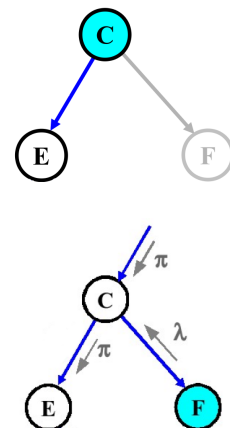
The λ message can be written directly as a single scalar equation:

$$\lambda_E(c_j) = \sum_k \pi_E(w_k) \sum_i P(e_i|c_j \& w_k)\lambda(e_i)$$

Instantiation and Dependence

We have already noted that instantiating a node with children, means that any λ message sent by a child node is ignored. This means that all the children of an instantiated node are independent. Any change of the probability distribution of any of them cannot change the others. For example, when C is instantiated any change in F will not effect E . We call this property “conditional independence” since the children are independent given that the parent is in a known state (or condition).

However, if the parent node is not instantiated then a change to one of its children will change all the other children, since it changes the π message from the parent to each child. Thus, if C is not instantiated a change in F will change the value of E .



An initially surprising result is that for a child with multiple parents the opposite is true. That is to say if a node with multiple parents is not instantiated then its parents are independent. However, if a node with multiple parents is instantiated, then there is dependence between its parents. This becomes clear when we look at the equation for the λ message to joint parents.

$$\lambda_E(c_j) = \sum_k \pi_E(w_k) \sum_i P(e_i | c_j \& w_k) \lambda(e_i)$$

For the case where E has no evidence we can write $\lambda(e_i) = 1$ for all i . Thus the second sum will evaluate to 1 for any j and k , since we are summing the columns of the link matrix. thus the equation reduces to:

$$\lambda_E(c_j) = \sum_k \pi_E(w_k)$$

Thus $\lambda_E(c_j)$ will evaluate to the same number for any j hence expressing no evidence. This is not so if we have λ evidence for E .

The Operating Equations for Probability Propagation

In order to propagate probabilities in Bayesian networks we need just five equations.

Operating Equation 1: The λ message

The λ message from C to A is given by:

$$\lambda_C(a_i) = \sum_{j=1}^m \pi_C(b_j) \sum_{k=1}^n P(c_k | a_i \& b_j) \lambda(c_k)$$

In the case where we have a single parent (A) this reduces to:

$$\lambda_C(a_i) = \sum_{k=1}^n P(c_k | a_i) \lambda(c_k)$$

and for the case of the single parent we can use the simpler matrix form:

$$\lambda_C(\mathbf{A}) = \lambda(\mathbf{C})\mathbf{P}(\mathbf{C}|\mathbf{A})$$

The matrix form for multiple parents relates to the joint states of the parents.

$$\lambda_C(\mathbf{A}\&\mathbf{B}) = \lambda(\mathbf{C})\mathbf{P}(\mathbf{C}|\mathbf{A}\&\mathbf{B})$$

It is necessary to separate the λ evidence for the individual parents with a scalar equation of the form:

$$\lambda_C(a_i) = \sum_j \pi_C(b_j) \lambda_C(a_i \& b_j)$$

Operating Equation 2: The π Message

If C is a child of A , the π message from A to C is given by:

$$\pi_C(a_i) = \begin{cases} 1 & \text{if } A \text{ is instantiated for } a_i \\ 0 & \text{if } A \text{ is instantiated but not for } a_i \\ P'(a_i)/\lambda_C(a_i) & \text{if } A \text{ is not instantiated} \end{cases}$$

Operating Equation 3: The λ evidence

If C is a node with n children D_1, D_2, \dots, D_n , then the λ evidence for C is:

$$\lambda(c_k) = \begin{cases} 1 & \text{if } C \text{ is instantiated for } c_k \\ 0 & \text{if } C \text{ is instantiated but not for } c_k \\ \prod_i \lambda_{D_i}(c_k) & \text{if } C \text{ is not instantiated} \end{cases}$$

Operating Equation 4: The π evidence

If C is a child of two parents A and B the π evidence for C is given by:

$$\pi(c_k) = \sum_{i=1}^l \sum_{j=1}^m P(c_k|a_i \& b_j) \pi_C(a_i) \pi_C(b_j)$$

This can be written in matrix form using as follows:

$$\pi(\mathbf{C}) = \mathbf{P}(\mathbf{C}|\mathbf{A}\&\mathbf{B})\pi_{\mathbf{C}}(\mathbf{A}\&\mathbf{B})$$

where

$$\pi_C(a_i \& b_j) = \pi_C(a_i) \pi_C(b_j)$$

The single parent matrix equation is:

$$\pi(\mathbf{C}) = \mathbf{P}(\mathbf{C}|\mathbf{A})\pi_{\mathbf{C}}(\mathbf{A})$$

Operating Equation 5: The posterior probability

If C is a variable the (posterior) probability of C based on the evidence received is written as:

$$P'(c_k) = \alpha \lambda(c_k) \pi(c_k)$$

where α is chosen to make $\sum_k P'(c_k) = 1$

Message Passing Summary

New evidence enters a network when a variable is instantiated, ie when it receives a new value from the outside world. When this happens the posterior probabilities of each node in the whole network must be re-calculated. This is achieved by message passing. When the λ or π evidence for a node changes it must inform some of its parents and its children as follows:

For each parent to be updated

 Update the parents λ message array

 Set a flag to indicate that the parent must be re-calculated

For each child to be updated

 Update the child's π message array

 Set a flag to indicate that the child must be re-calculated

The network reaches a steady state when there are no more nodes to be re-calculated. This condition will always be reached in singly connected networks. Multiply connected networks will not necessarily reach a steady state. We are now in a position to set out the algorithm. There are four steps that are required.

1. Initialisation

The network is initialised to a state where no nodes are instantiated. That is to say all we have in the way of information is the prior probabilities of the root nodes, and the conditional probability link matrices.

- 1 All λ messages and λ values are set to 1 (there is no λ evidence)
- 2 All π messages are set to 1
- 3 For all root nodes the π values are set to the prior probabilities, eg, for all states of R : $\pi(r_i) = P(r_i)$
- 4 Post and propagate the π messages from the root nodes using downward propagation

2. Upward Propagation

This is the way evidence is passed up the tree.

if a node C receives λ message from one child

if C is not instantiated

- { 1 Compute the new $\lambda(C)$ value (Operating Equation 3)
- 2 Compute the new posterior probability $P'(C)$ (Operating Equation 5)
- 3 Post a λ message to all C 's parents
- 4 Post a π message to C 's other children
- }

Notice that if a node is instantiated, then we know its value exactly, and the evidence from its children has no effect.

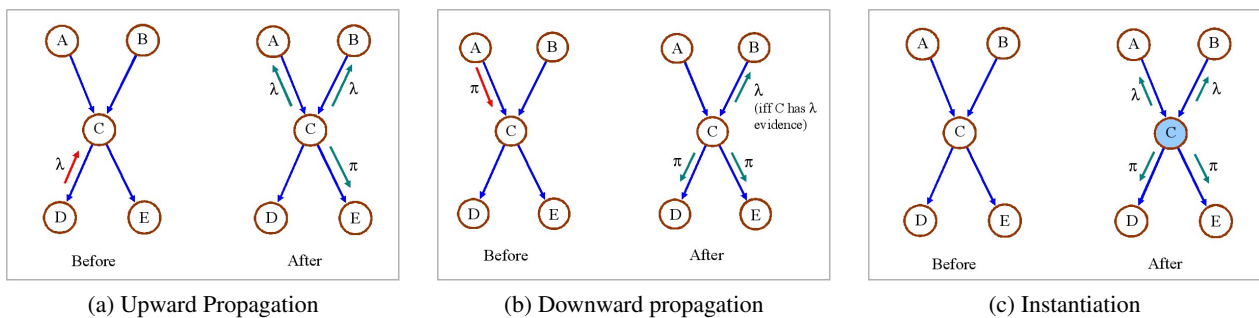


Figure 1: Message Passing

3. Downward Propagation

if a variable C receives a π message from one parent:

if C is not instantiated

- { 1 Compute a new value for array $\pi(C)$ (Operating Equation 4)
- 2 Compute a new value for $P(C)$ (Operating Equation 5)
- 3 Post a π message to each child
- }

if (there is λ evidence in C)

Post a λ message to the other parents

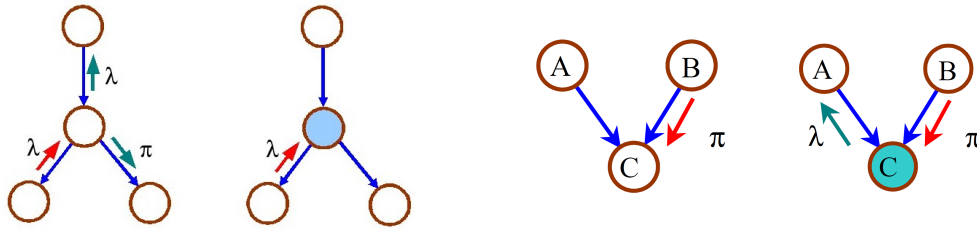
Instantiation

If variable C is instantiated for state c_k

- 1 for all $j \neq k$ set $P'(c_j) = 0$
- 2 set $P'(c_k) = 1$
- 3 Compute $\lambda(C)$ (Operating Equation 3)
- 4 Post a λ message to each parent of C
- 5 Post a π message to each child of C

Blocked Paths

If a node is instantiated it will block some message passing between other nodes. For example, there is clearly no sense in sending a λ or π message to an instantiated node, as the evidence will be ignored since we know the state of the node. This is the case for diverging paths.



However, converging paths (multiple parents) behave differently. Converging paths are blocked when there is no λ evidence on the child node, but unblocked when there is λ evidence, or the node is instantiated. This becomes clear when we look at operating equation 1.

$$\lambda_C(a_i) = \sum_{j=1}^m \pi_C(b_j) \sum_{k=1}^n P(c_k | a_i \& b_j) \lambda(c_k)$$

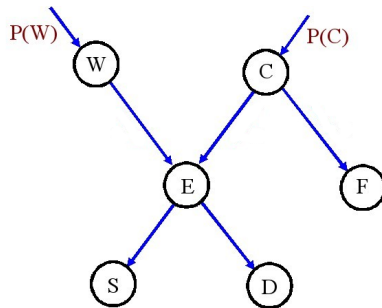
Given that node C has no λ evidence we can write $\lambda(C) = \{1, 1, 1, \dots, 1\}$ and substituting this into operating equation 1 we get:

$$\lambda_C(a_i) = \sum_{j=1}^m \pi_C(b_j) \sum_{k=1}^n P(c_k | a_i \& b_j)$$

The second summation now evaluates to 1 for any value of i and j , since we are summing a probability distribution. Thus:

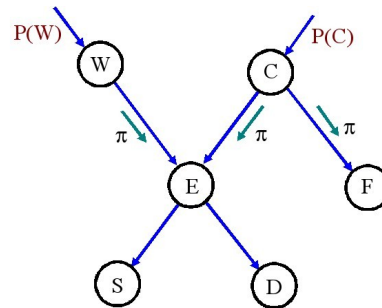
$$\lambda_C(a_i) = \sum_{j=1}^m \pi_C(b_j)$$

The sum is independent of i and hence the value of $\lambda_C(a_i)$ is the same for all values of i , that is to say there is no λ evidence sent to A . Finally, we give an example of a sequence of message passing in our owl and pussy cat example.



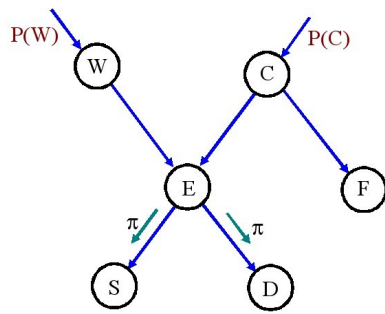
1. Initialisation

The only evidence initially is the prior probabilities of the the roots.



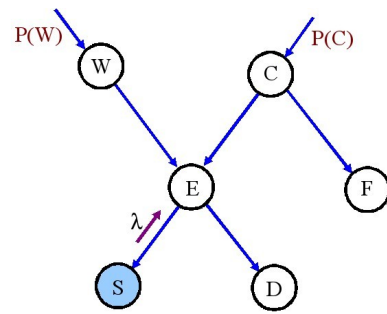
1. Initialisation

The priors become π messages and propagate downwards.



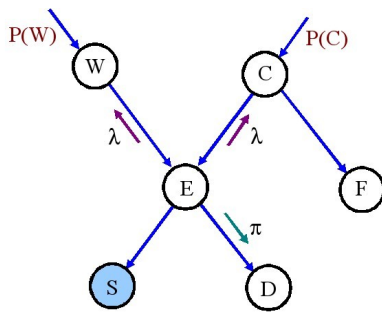
2. Initialisation (cont.)

E has no λ evidence so it just sends a π message to its children. Then the network reaches a steady state.



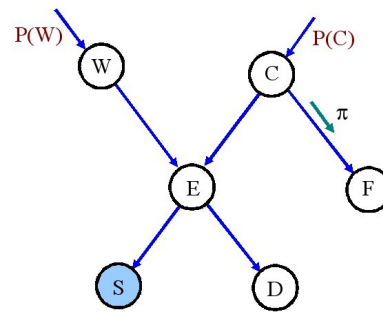
3. Instantiation

A new measurement is made for node S . We instantiate it and λ message propagate upwards.



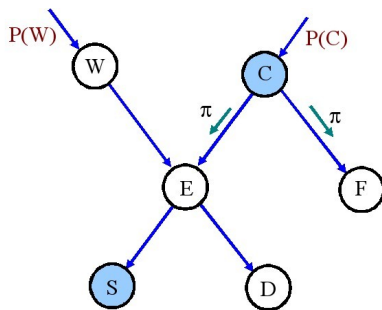
4. Propagation

E receives the λ message from S and recomputes its λ evidence and its π message to D . It sends messages everywhere except to S .



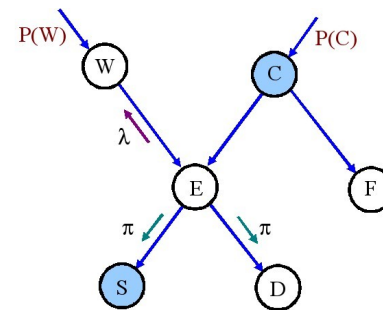
5. Propagation (cont.)

W and C are updated. Their posterior probabilities are calculated. The π message from C to F has changed, so that will be updated too. The π message is sent to F , it updates its posterior probability and the network reaches a steady state.



6. More Evidence

C is now instantiated and sends π messages to its children.



7. Propagation

E has λ evidence so it will send a λ message to W . The π message it sends to D will require D to update its posterior probability, but the π message sent to S will be blocked as S has been instantiated, and will not change.