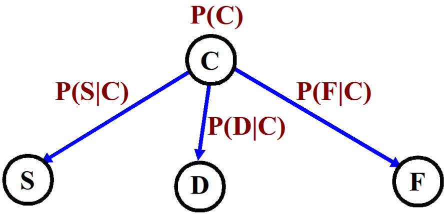
Lecture 3

Evidence and Message Passing

The story so far:

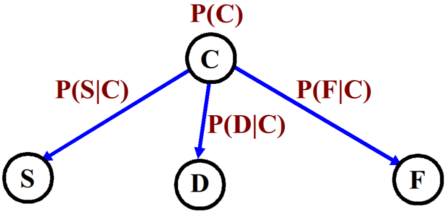
Naive Bayesian networks express Bayes' theorem for conditionally independent variables:

$$P(C|S\&D\&F) = \alpha P(C)P(S|C)P(D|C)P(F|C)$$



Variable	Interpretation	Type	Value
C	Cat	Discrete (2 states)	c_1, c_2
S	Eye separation	Discrete (7 states)	$s_1, s_2, s_3, s_4, s_5, s_6, s_7$
D	Eye difference	Discrete (4 states)	d_1, d_2, d_3, d_4
F	Fur colour	Discrete (20 states)	$f_1, f_2 \cdots f_{20}$

The story so far:

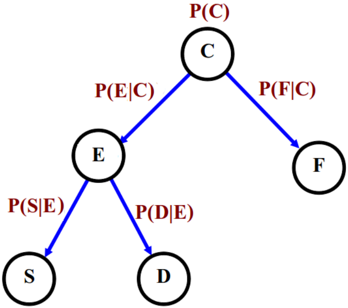


Bayes theorem allows us to separate the evidence into two parts:

- Prior: $P(C)$
- Likelihood: $P(S|C), P(D|C), P(F|C)$

Both types of evidence are available to us, either through established knowledge or from measurements.

The story so far:

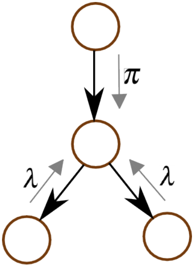


- Tree structured networks represent the data more accurately.
- Probabilities are again computed by Bayes' theorem, but the calculations are tricky.
- We need an algorithm that will extend easily to large networks.

now read on . . .

Evidence

To deal with intermediate nodes we need to generalise the method of calculating probabilities. At any intermediate node we have:



- Evidence from its descendents λ evidence.
- Evidence from its parent(s): π evidence.
- λ evidence generalises the concept of Likelihood
- π evidence generalises the concept of prior probability.

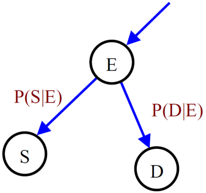
The Nature of Evidence

- Evidence is simply unnormalised probability. The evidence for the states of a variable need not sum to one (though they can do).
- We write it as a vector $\boldsymbol{\lambda}(\boldsymbol{E}) = [\lambda(e_1), \lambda(e_2), \lambda(e_3)]$
- We combine it by multiplication $\varepsilon(e_i) = \lambda(e_i) \times \pi(e_i)$
- Using evidence simplifies the equations since we do not need the normalising constant α.

Calculating λ Evidence

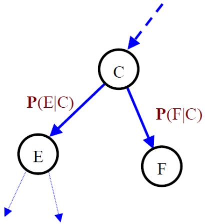
- For the simple case where leaf nodes are instantiated we can find the λ evidence for the immediate parents directly from the link matrices.
- Each child node provides a λ message to its parent, and the λ evidence for the parent is calculated by multiplying the messages together.
- In our cat network, if $S = s_4$ and $D = d_2$ the λ evidence for node E is calculated as follows:

$$\lambda(e_1) = P(s_4|e_1) \times P(d_2|e_1)$$
$$\lambda(e_2) = P(s_4|e_2) \times P(d_2|e_2)$$
$$\lambda(e_3) = P(s_4|e_3) \times P(d_2|e_3)$$



Calculating λ Evidence

- When we pass evidence from an intermediate node to its parents we must take into account the evidence we have for each state of the child node.
- The λ message is calculated by weighting the conditional probabilities from the link matrix by the evidence for the child node.



$$\lambda(c_1) = [\lambda(e_1)P(e_1|c_1) + \lambda(e_2)P(e_2|c_1) + \lambda(e_3)P(e_3|c_1)]P(f_3|c_1)$$

The Conditioning Equation

In general we use the "conditioning equation", so called because we are calculating probabilities according to the current conditions (ie measured values for some of the variables).

$$\lambda(c_i) = \prod_{(children)} \sum_j \lambda(h_j) P(h_j|c_i)$$

The states of the parent node are indexed by i and the states of the child node are indexed by j

Instantiation and Evidence

In the simple case, for leaf nodes we have a known state for that node. We defined the eye separation measure as having seven states, and if we make a measurement we set the evidence for one state to 1 and the others to 0.

State	Range	$\lambda(s_i)$
s_1	$[below - 1.5]$	0
s_2	$[-1.5 \cdots -0.75]$	0
s_3	$[-0.75 \cdots -0.25]$	0
s_4	$[-0.25 \cdots 0.25]$	0
s_5	$[0.25 \cdots 0.75]$	1
s_6	$[0.75 \cdots 1.5]$	0
s_7	$[above 1.5]$	0

Conditioning at the Leaf Nodes

If we apply the conditioning equation to instantiated leaf nodes we get the same result as selecting the conditional probabilities from the link matrices.

$$\lambda(c_i) = \prod_{(children)} \sum_j \lambda(h_j) P(h_j|c_i)$$

Calculating the evidence for the E node we have:

$$\lambda(e_i) = (\textstyle\sum_j \lambda(s_j) P(s_j|e_i))(\textstyle\sum_k \lambda(d_k) P(d_k|e_i))$$

and if S is instantiated to s_4 and D to d_2, then only $\lambda(s_4)$ and $\lambda(d_2)$ are non zero and the right hand side reduces to:

$$\lambda(e_i) = P(s_4|e_i) \times P(d_2|e_i)$$

Virtual Evidence

Sometimes, when we make a measurement it is possible to express uncertainty about it by distributing the evidence values. For example, instead of setting $\lambda(s_5) = 1$ we could use a Gaussian distribution:

State	Range	$\lambda(s_i)$
s_1	$[below - 1.5]$	0
s_2	$[-1.5 \cdots -0.75]$	0
s_3	$[-0.75 \cdots -0.25]$	0.08
s_4	$[-0.25 \cdots 0.25]$	0.3
s_5	$[0.25 \cdots 0.75]$	0.5
s_6	$[0.75 \cdots 1.5]$	0.1
s_7	$[above 1.5]$	0.02

This makes our level of uncertainty about the measurement explicit.

Virtual Evidence requires Conditioning

If we use virtual evidence at the leaf nodes then we calculate the evidence sent to the parent using the condidioning equation.

$$\lambda(c_i) = \prod_{(children)} \sum_j \lambda(h_j) P(h_j|c_i)$$

So in the event that we have virtual evidence for S, but D is instantiated to d_2 we find that:

$$\lambda(e_i) = (\textstyle\sum_j \lambda(s_j) P(s_j|e_i)) \times P(d_2|e_i)$$

No Evidence

Sometimes, we may not have data for a node. In this case each state must be given the same λ value, and for convenience we choose this to be 1.

State	Range	$\lambda(s_i)$
s_1	$[below - 1.5]$	1
s_2	$[-1.5 \cdots -0.75]$	1
s_3	$[-0.75 \cdots -0.25]$	1
s_4	$[-0.25 \cdots 0.25]$	1
s_5	$[0.25 \cdots 0.75]$	1
s_6	$[0.75 \cdots 1.5]$	1
s_7	$[above 1.5]$	1

No Evidence and the Conditioning Equation

The conditioning equation still works if we have no evidence. For example if there is no evidence on node S, but we have evidence on D we get:

$$\lambda(e_i) = (\sum_j \lambda(s_j) P(s_j|e_i))(\sum_k \lambda(d_k) P(d_k|e_i))$$

$$\lambda(e_i) = (\sum_j P(s_j|e_i))(\sum_k \lambda(d_k) P(d_k|e_i))$$

$$\lambda(e_i) = \sum_k \lambda(d_k) P(d_k|e_i)$$

The evidence from S evaluates to 1 for every state of E.

Problem Break

Given the following virtual evidence, write down an expression for the λ evidence for state e_1 of E (which has two children S and D)

s_1	1
s_2	0.2
s_3	0
s_4	0
s_5	0
s_6	0
s_7	0

d_1	0
d_2	0
d_3	0.5
d_4	1

Solution

The general conditioning equation states:

$$\lambda(c_i) = \prod_{(children)} \sum_j \lambda(h_j) P(h_j|c_i)$$

so for this case:

$$\lambda(e_1) = (P(s_1|e_1) + 0.2P(s_2|e_1)) \times (0.5P(d_3|e_1) + P(d_4|e_1))$$

Upward Propagation

- Our cat network can be used as a Bayesian classifier. In this case we have one hypothesis node, which is the root node C.
- All the evidence is propagated upwards, using the conditioning equation for all instances.
- At the root node the λ evidence is multiplied by the prior probability of the root and we then have a probability distribution from which we can classify the picture as “cat” or “not cat”.

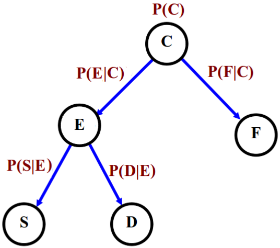
The Conditioning Equation in Vector form

- We have been using the scalar form of the conditioning equation in order to demonstrate how it works.
- In practice it is much easier to calculate λ messages using a vector equation thus:

$$\boldsymbol{\lambda}_{\mathbf{S}}(\boldsymbol{E}) = \boldsymbol{\lambda}(\boldsymbol{S}) P(S|E)$$

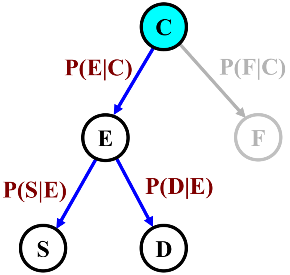
The Cat Example Again

- Propagating the λ evidence up the tree allows us to collect all the evidence there is about the C node.
- However, suppose we want to reason about the E node (which is not one of our measured variables)? There is evidence for E that comes from its parent C.



Case 1: C is instantiated

- Suppose we measure node C and it turns out that there is a cat in the picture.
- If node C is in state C_1 (cat=true) then $P(C) = [1, 0]$ and its children (in particular F in this case) cannot affect its value.



Looking at the Link Matrix

From the fundamental law of probability we know that

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

and since C is in a known state ($\boldsymbol{P'(C)} = [1,0]$) we can write:

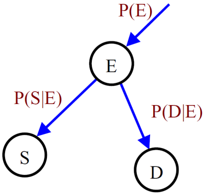
$$P(e_i\&c_2) = P(e_i|c_j)0 = 0$$
$$P(e_i\&c_1) = P(e_i|c_1)1 = P(e_i|c_1)$$

Which means that we can calculate all values of $\boldsymbol{P(E)}$ from the matrix equation:

$$\boldsymbol{P}(\boldsymbol{E}) = \begin{bmatrix} P(e_1|c_1) & P(e_1|c_2) \\ P(e_2|c_1) & P(e_2|c_2) \\ P(e_3|c_1) & P(e_3|c_2) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} P(e_1|c_1) \\ P(e_2|c_1) \\ P(e_3|c_1) \end{bmatrix}$$

Simplified Network

- The previous equation allows us to calculate a probability distribution over the states of E using only the evidence from its parent, node C.
- We can treat this value of $P(E)$ as if it were a prior probability of E.
- To find the posterior probability of E we need to multiply it by $\lambda(E)$ and normalise.
- It is as if we have a simplified network:



Case 2: C is not instantiated but has evidence from other sources

- In the more general case we might not be able to instantiate C, but we could have evidence about its state.
- Some of that evidence may have originated from E, but, if we are to send evidence to E we are interested only in the evidence that comes from elsewhere.
- In this example that is the evidence from F and the prior probability (evidence) for C.

A π Message from C to E

Suppose for a given picture we calculate the λ evidence for C from F as:

$$\boldsymbol{\lambda_F(C)} = [\lambda_F(c_1), \lambda_F(c_2)] = [0.3, 0.2]$$

and the prior probability of C is:

$$P(C) = [0.6, 0.4]$$

the total evidence for C, excluding any evidence from E, is found by multiplying the individual evidence values:

$$\boldsymbol{\pi_E(C)} = [0.3 \times 0.6, 0.2 \times 0.4] = [0.18, 0.08]$$

This is the π message to E from C. It is not $P'(C)$ since it does not contain the evidence from E.

A π Message from C to E

The π evidence for E, which is written as $\boldsymbol{\pi(E)}$ is found as follows:

$$\boldsymbol{\pi(E)} = P(E|C)\boldsymbol{\pi_E(C)}$$

$$\boldsymbol{\pi(E)} = \begin{bmatrix} P(e_1|c_1) & P(e_1|c_2) \\ P(e_2|c_1) & P(e_2|c_2) \\ P(e_3|c_1) & P(e_3|c_2) \end{bmatrix} \begin{bmatrix} 0.18 \\ 0.08 \end{bmatrix} = \begin{bmatrix} 0.18P(e_1|c_1) + 0.08P(e_1|c_2) \\ 0.18P(e_2|c_1) + 0.08P(e_2|c_2) \\ 0.18P(e_3|c_1) + 0.08P(e_3|c_2) \end{bmatrix}$$

Scalar equation for the π Evidence

There is also a scalar form of the equation for π evidence:

π evidence sent to one child

Conditional probability from link matrix

$$\pi(ei) = \sum_j \left[P(ei|cj) \left\{ \overbrace{\pi(cj) \underbrace{\prod_k \lambda_k(cj)}_{\text{Evidence from parent's other children}}}^{\text{Other evidence for the parent}} \right\} \right]$$

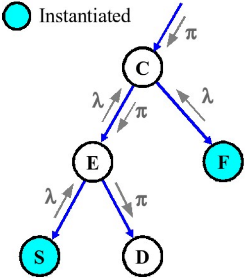
Sum taken over the parent's states

Evidence from parent's ancestors

The less said about it the better!

Generality of Propagation in Trees

- Now that we can propagate probability both up and down a tree making inferences is completely flexible.
- We can instantiate any subset of the nodes and calculate the probability distribution over the states of the other nodes.



Magnitude and Evidence

- Remember that the magnitude of the evidence is not relevant. It is the relative magnitudes of the evidence for the states of a node that carries the information.
- We could at any point normalise evidence to turn it into a probability distribution, but doing this would make no difference to the outcome of the probability propagation.

Normalisation of Evidence in a Tree

For example, to calculate the π message from C to E we could first calculate the posterior probability of C:

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

and then divide out the evidence from E

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

The magnitudes of $\pi_E(C)$ would be different, but the final result for $P'(E)$ would be the same as using the previous calculation.

Prior and π, Likelihood and λ

- We have now a uniform way of propagating probabilities at any node in a network.
- The π evidence that comes from the parent is equivalent to the prior probability at the root of a tree.
- The λ evidence is equivalent to likelihood information that comes from the subtree below a node.
- This means that probabilities can be calculated by a simple message passing algorithm.

Incorporating more Nodes

- One of the best features of Bayesian Networks is that we can incorporate new nodes as the data becomes available.
- Recall that we had information from the computer vision process as to how likely the extracted circles were.
- This could simply be treated as another node.

Adding a Node doesn't change the rest of the Network

If we add this new node we only need to find one new conditional probability matrix. All the others remain unchanged.