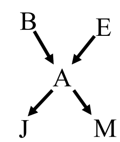
Lecture 18: Uncertainty 3

Victor R. Lesser
CMPSCI 683
Fall 2010

P(Burglary |JohnCalls) – using joint probability distribution

- **Diagnostic inferences** (from effects to causes).
  - Given that ***JohnCalls***, infer that *P(Burglary | JohnCalls)* = 0.016

  P(B,J) = normalized Sum (E,A,M) P(B,e,a,J,m)



P(B,E,A,J,M)= P(B)P(E)P(A|B,E) P(J|A)P(M|A)

- The neighbors (John, Mary) promise to call you at work when they hear the alarm
  - John always calls when he hears the alarm, but confuses alarm with phone ringing (and calls then also)
  - Mary likes loud music and sometimes misses alarm!
  - Assumption: John and Mary don't perceive burglary directly; they do not feel minor earthquakes

P(Burglary |JohnCalls) – using inference

Rearranging conditional probability expression to exploit CPTs in belief network

- *Bayes Rule* — *k*P(J|B)*P(B)*
- *Marginalization* — $k*Sum_A\ P(J,Alarm|B)*P(B)$
- $P(s_i,s_j|d) = P(s_i|\ s_j,d)\ P(s_j|d)$ — $k*Sum_A\ P(J|A,B)*P(A|B)\ *P(B)$
- Case 1: a node is conditionally independent of non-descendants given its parents — $k*Sum_A\ P(J|A)*P(A|B)\ *P(B)$
- *Marginalization* — $k*Sum_A\ P(J|A)*Sum_E P(A,E|B)\ *P(B)$
- $P(s_i,s_j|d)=P(s_i|\ s_j,d)\ P(s_j|d)$ — $k*Sum_A\ P(J|A)*Sum_E P(A|B,E)*P(E|B)\ *P(B)$
- *case 1 P(E|B)=P(E)* — $k*Sum_A\ P(J|A)*Sum_E P(A|B,E)*P(E)\ *P(B)$

Can read everything off the CPT's

Topics for this Lecture

- Construction of Belief Network
- Inference in Belief Networks
  - Variable Elimination

Benefits of belief networks

- Individual “design” decisions are understandable: causal structure and conditional probabilities.
- BNs encode conditional independence, without which probabilistic reasoning is hopeless.
- Can do inference even in the presence of missing evidence.

Constructing belief networks

Loop:

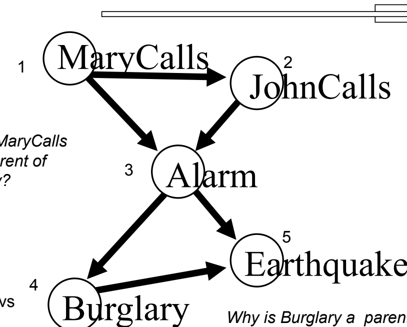
- Pick a variable X_i to add to the graph.
- Find (*minimal*) set of parents (previous nodes created) such that $P(X_i|\text{Parents}(X_i)) = P(X_i|X_{i-1}, X_{i-2}, \ldots, X_1)$
  - *Conditional Dependence vs Causality*
- Draw arcs from Parents(X_i) to X_i.
- Specify the CPT: $P(X_i|\text{Parents}(X_i))$.

Constructing belief networks cont.

Properties of the algorithm:

- Graph is always acyclic.
- No redundant information => consistency with the axioms of probability.
- Network structure/compactness depends on the ordering of the variables.

Example: Ordering M,J,A,B,E



Why is MaryCalls not a parent of Burglary?

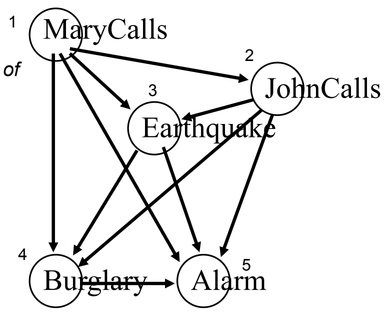
Conditional Dependence vs Causality

Why is Burglary a parent of Earthquake?

Example: Ordering M,J,E,B,A

Ordering Affects Size of CPTs?

Why is Marycalls a parent of Burglary in this ordering?



d-separation: Direction-Dependent Separation

- Network construction
  - Conditional independence of a node and its predecessors, given its parents
  - The absence of a link between two variables does not guarantee their independence
- Effective inference needs to exploit *all available conditional independences*
  - Which set of nodes X are conditionally independent of another set Y, given a set of evidence nodes E
    - P(X,Y|E) = P(X|E) • P(Y|E); P(X|Y,E)= P(X|E); P(Y|X,E)= P(Y|E)
  - Limits propagation of information
  - Comes directly from structure of network

d-separation

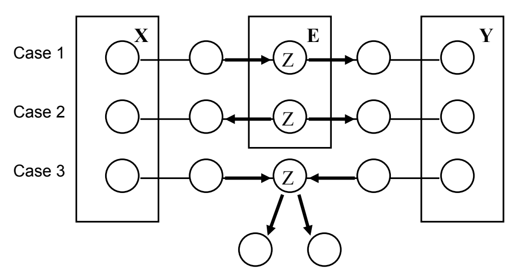
Definition: If X, Y and E are three disjoint subsets of nodes in a DAG, then E is said to d-separate X from Y if *every undirected path from X to Y is blocked by E*. A path is blocked if it contains a node Z such that:

(1) Z has one incoming and one outgoing arrow; or

(2) Z has two outgoing arrows; or

(3) Z has two incoming arrows and neither Z nor any of its descendants is in E.

d-separation cont.



Directionality of links from X or Y to immediate predecessor or successor of Z not important; In case 3, notice E is outside of Z

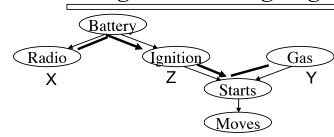
d-separation cont.

- Property of belief networks: if X and Y are d-separated by E, then X and Y are conditionally independent given E.
- An "if-and-only-if" relationship between the graph and the probabilistic model cannot always be achieved.
  - *The graph may not represent all possible conditional independent relations*

d-separation example- case 1

(1) Z has one incoming and one outgoing arrow

Z in E has one incoming and one outgoing arrow



Whether there is *Gas*[G] in the car and whether the car *Radio* [R] plays are independent given evidence about whether the *SparkPlugs* fire [Ignition] (*case 1*).

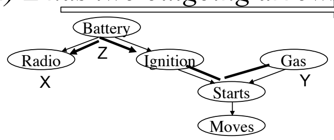
P(R,G|I) = P(R|I) · P(G|I)

P(G|I,R) = P(G|I); P(R|I, G) = P(R|I)

d-separation example- case 2

(2) Z has two outgoing arrows

Z in E has two outgoing arrows



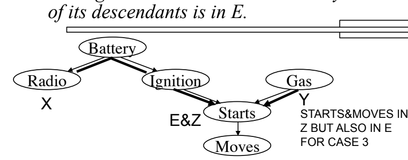
Gas and *Radio* are conditionally-independent if it is known if the *Battery* [B]works (*case2*).

P(R|B,G) = P(R|B); P(G|B,R)=P(G|B)

d-separation example - case 3

(3) Z has two incoming arrows and neither Z nor any of its descendants is in E.

Z has two incoming arrows and neither Z nor any of its descendants is in E



Gas and *Radio* are *independent given no evidence at all*.

P(Gas|Radio)=P(Gas); P(Radio|Gas)=P(Radio);

But they are *dependent given evidence about whether the car Starts*.

P(Gas|Radio,Start) ~= P(Gas|Start)

For example, if the car does not start, then the radio playing is increased evidence that we are out of gas. *Gas* and *Radio* are also dependent given evidence about whether the car *Moves*, because that is enabled by the car starting {does not fit into any of the 3 cases}.

Inference in Belief Networks

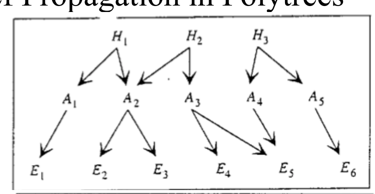
- BNs are fairly expressive and easily engineered representation for knowledge in probabilistic domains.
- They facilitate the development of inference algorithms.
- They are particularly suited for parallelization
- Current inference algorithms are efficient and can solve large real-world problems.

Network Features Affecting Efficiency of Reasoning

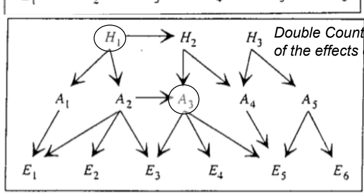
- Topology (trees, singly-connected, sparsely-connected, DAGs).
- Size (number of nodes).
- Type of variables (discrete, cont, functional, noisy-logical, mixed).
- Network dynamics (static, dynamic).

Belief Propagation in Polytrees

Polytree belief network, where nodes are singly connected
•***Exact inference, Linear in size of network***

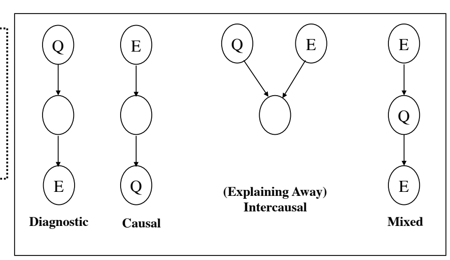


Multiconnected belief network. This is a DAG, but not a polytree.
•*Exact inference, Worst case NP-hard*



Reviewing Alternative Reasoning in Belief Networks

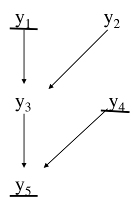
Simple examples of 4 patterns of reasoning that can be handled by belief networks. *E* represents an evidence variable; *Q* is a query variable.



P(Q/E) =?

Reasoning Directly in Belief Networks: Calculation in Polytree with Evidence Above

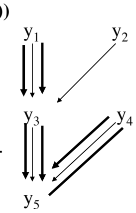
- **What is p(Y5|Y1,Y4)**
  - Define in terms of CPTs = p(Y5,Y4,Y3,Y2,Y1)
  - p(Y5|Y3,Y4)p(Y4)p(Y3|Y1,Y2)p(Y2)p(Y1)
  - p(Y5|Y1,Y4)= p(Y5,Y1,Y4)/p(Y1,Y4)
  - Use cpt to sum over missing variables
  - p(Y5,Y1,Y4)= Sum(Y2,Y3) p(Y5,Y4,Y3,Y2,Y1)
  - assuming variables take on only truth or falsity.



- **p(Y5|Y1,Y4) = p(Y5,Y3|Y1, Y4) + P(Y5, ~Y3|Y1,Y4)**
  - **Connect to parents of Y5 not already part of expression, by marginalization**
- **= SUM(Y3) p(Y5,Y3|Y1,Y4)**

Continuation of Example Above

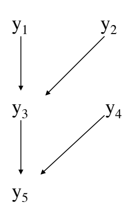
- **= SUM(Y3)(p(Y5|Y3, Y1, Y4) * p(Y3| Y1, Y4))**
  - **$P(s_i,s_j|d) = P(s_i| s_j,d)\ P(s_j|d)$**
- **= SUM(Y3) p(Y5|Y3, Y4) * p(Y3| Y1, Y4)**
  - **Y1 conditionally independent of Y5 given Y3,**
  - **Y3 represents all the contributions of Y1 to Y5**
  - **Case 1: a node is conditionally independent of non-descendants given its parents**



- **= SUM(Y3) p(Y5|Y3, Y4) * p(Y3|Y1)**
  - **Y4 conditionally independent of Y3 given Y1**
  - **Case 3: Y1 (E) not a descendant of Y5 (Z) which d-separates Y3 (X) and Y4 (Y); thus p(Y3| Y1, Y4) = p(Y3|Y1)**

Continuation of Example Above

- **= SUM(Y3) p(Y5|Y3, Y4) * (Sum (Y2)p(Y3, Y2 |Y1))**
  - **Connect to parents of Y3 not already part of expression**
- **= SUM(Y3) p(Y5|Y3, Y4) *(Sum (Y2) p(Y3| Y1,Y2) * p(Y2|Y1))**
  - **$p(s_i,s_j|d) = p(s_i| s_j,d)\ p(s_j|d)$; product rule**

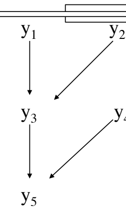


- **= SUM(Y3) p(Y5|Y3, Y4) *(SUM(Y2) p(Y3| Y1,Y2)*p(Y2))**
  - **Y2 independent of Y1; p(Y2/Y1)=p(Y2)**
  - **Definition of Baysean network**

What is p(Y5|Y1,Y4)

Reasoning Directly in Belief Networks: Calculation in Polytree with Evidence Below

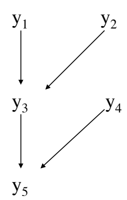
- **What is p(Y1|Y5)**
  - **p(Y1|Y5)=p(Y1,Y5)/p(Y5)**
  - **p(Y1,Y2,Y3,Y4,Y5) = in terms of cpt**
  - **p(Y5|Y3,Y4)p(Y3|Y1,Y2)p(Y1)p(Y2)p(Y4)**



- **p(Y1|Y5) = p(Y5|Y1)p(Y1)/p(Y5)**
  - **Bayes Rule**
- **=K * p(Y5|Y1)p(Y1)**

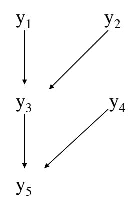
Continuation of Example Below

- =K * p(Y5|Y1)p(Y1)
- = K * *(SUM(Y3) p(Y5|Y3)p(Y3|Y1))* p(Y1)
  - Connect to Y3 parent of Y5 not already part of expression
  - $P(s_i | s_j)$ = SUM(d)$P(s_i | s_j, d)$ $P(d | s_j)$
  - Y1 conditionally independent of Y5 given Y3; case 1
  - p(Y5|Y3,Y1)= p(Y5|Y3)
- = K * (SUM(Y3) *(SUM(Y4)p(Y5|Y3,Y4)p(Y4|Y3))*p(Y3|Y1)) p (Y1)
  - Connect to Y4 parent of Y5 not already part of expression
  - $P(s_i | s_j)$ = SUM(d)$P(s_i | s_j, d)$ $P(d | s_j)$
- = K * (SUM(Y3) (SUM(Y4)p(Y5|Y3,Y4)p(Y4))p(Y3|Y1)) p (Y1)
  - Y4 independent of Y3; p(Y4|Y3)= p(Y4)



Continuation of Example Below

- **= K * (SUM(Y3) (SUM(Y4)p(Y5|Y3,Y4)p (Y4))*p(Y3|Y1))* p(Y1)**
- **= K * (SUM(Y3) (SUM(Y4)p(Y5|Y3,Y4)p (Y4))*(SUM(Y2)p(Y3|Y1,Y2)p(Y2|Y1)))* p(Y1)**
  - **Connect to Y2 parent of Y3 not already part of expression**
  - **$P(s_i | s_j)$ = SUM(d)$P(s_i | s_j, d)$ $P(d | s_j)$**
- **= K * (SUM(Y3) (SUM(Y4)p(Y5|Y3,Y4)p (Y4))(SUM(Y2)p(Y3| Y1,Y2)p(Y2))) p(Y1)**
  - **Y2 independent of Y1**
  - **Expression that can be calculated from cpt**



Evidence Above and Below for Polytrees

If there is evidence both above and below P(Y3|Y5,Y2)

we separate the evidence into above, ε^+, and below, ε^-, portions and use a version of Bayes' rule to write

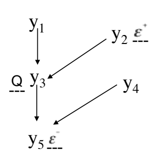
$$p(Q \mid \varepsilon^+, \varepsilon^-) = \frac{p(\varepsilon^- \mid Q, \varepsilon^+)\, p(Q \mid \varepsilon^+)}{p(\varepsilon^- \mid \varepsilon^+)}$$

we treat $\frac{1}{p(\varepsilon^- \mid \varepsilon^+)} = k_2$ as a normalizing factor and write

$$p(Q \mid \varepsilon^+, \varepsilon^-) = k_2\, p(\varepsilon^- \mid Q, \varepsilon^+)\, p(Q \mid \varepsilon^+)$$

Q d-separates ε^- from ε^+, so

$$p(Q \mid \varepsilon^+, \varepsilon^-) = k_2\, p(\varepsilon^- \mid Q)\, p(Q \mid \varepsilon^+)$$



P(Y3|Y5,Y2)=P(Y5|Y3)P(Y3|Y2)

We calculate the first probability in this product as part of the top-down procedure for calculating $p(Q \mid \varepsilon^-)$. The second probability is calculated directly by the bottom-up procedure.

Announcement

- Material on Variable Elimination was not covered in class and will not be on final examination

Variable Elimination

- Can remove a lot of re-calculation/multiplications in expression
- **K * (SUM(Y3) (SUM(Y4)p(Y5|Y3,Y4)p(Y4))(***SUM(Y2)p(Y3|Y1,Y2)p(Y2)))* **p(Y1)**
- **Summations over each variable are done only for those portions of the expression that depend on variable**
- **Save results of inner summing to avoid repeated calculation**
  - **Create Intermediate Functions**
  - **F-Y2(Y3,Y1)=** *(SUM(Y2)p(Y3|Y1,Y2)p(Y2))*

Variable elimination

General idea:

- Write query in the form

$$P(X_1, \boldsymbol{e}) = \sum_{x_k} \cdots \sum_{x_3} \sum_{x_2} \prod_i P(x_i \mid pa_i)$$

- Iteratively
  - Move all irrelevant terms outside of innermost sum
  - Perform innermost sum, getting a new term
  - Insert the new term into the product

Variable elimination in chains*

- Consider the following chain:

$$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$$

- We know that

$$P(e) = \underline{\sum_d \sum_c \sum_b \sum_a} P(a,b,c,d,e)$$ Marginalization

- By chain decomposition, we get

$$P(e) = \sum_d \sum_c \sum_b \sum_a P(a,b,c,d,e)$$ Product Rule in Reverse order

$$= \sum_d \sum_c \sum_b \sum_a P(a)P(b \mid a)P(c \mid b)P(d \mid c)P(e \mid d)$$

Elimination in chains

$$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$$

- Rearranging terms ...

$$P(e) = \sum_d \sum_c \sum_b \sum_a P(a)P(b \mid a)P(c \mid b)P(d \mid c)P(e \mid d)$$

$$= \sum_d \sum_c \sum_b P(c \mid b)P(d \mid c)P(e \mid d) \sum_a P(a)P(b \mid a)$$

Elimination in chains

$$\cancel{A} \rightarrow B \rightarrow C \rightarrow D \rightarrow E$$

- Now we can perform innermost summation

$$P(e) = \sum_d \sum_c \sum_b P(c \mid b)P(d \mid c)P(e \mid d)\underbrace{\sum_a P(a)P(b \mid a)}$$
$$= \sum_d \sum_c \sum_b P(c \mid b)P(d \mid c)P(e \mid d)p(b)$$

Elimination in chains*

$$\cancel{A} \rightarrow \cancel{B} \rightarrow C \rightarrow D \rightarrow E$$

- Rearranging and then summing again, we get

$$P(e) = \sum_d \sum_c \sum_b P(c \mid b)\overbrace{P(d \mid c)P(e \mid d)}p(b)$$
$$= \sum_d \sum_c P(d \mid c)P(e \mid d)\underbrace{\sum_b P(c \mid b)p(b)}$$
$$= \sum_d \sum_c P(d \mid c)P(e \mid d)p(c)$$

Elimination in chains with evidence

- Similarly, we understand the backward pass

$$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$$

- We write the query in explicit form

$$P(a,e) = \sum_b \sum_c \sum_d P(a,b,c,d,e)$$
$$= \sum_b \sum_c \sum_d P(a)P(b \mid a)P(c \mid b)P(d \mid c)P(e \mid d)$$

Elimination in chains with evidence

$$A \rightarrow B \rightarrow C \rightarrow \cancel{D} \rightarrow E$$

- Eliminating *d*, we get

$$P(a,e) = \sum_b \sum_c \sum_d \overbrace{P(a)P(b \mid a)P(c \mid b)}P(d \mid c)P(e \mid d)$$
$$= \sum_b \sum_c P(a)P(b \mid a)P(c \mid b)\underbrace{\sum_d P(d \mid c)P(e \mid d)}$$
$$= \sum_b \sum_c P(a)P(b \mid a)P(c \mid b)P(e \mid c)$$

P(e|c) = Sum(d) P(e,d|c), P(e,d|c)=P(e|c,d)P(e|d), P(e|c,d)=P(e|c) d-sep

Elimination in chains with evidence

- Eliminating c, we get

$$P(a,e) = \sum_b \sum_c P(a)P(b \mid a)P(c \mid b)P(e \mid c)$$
$$= \sum_b P(a)P(b \mid a)\underbrace{\sum_c P(c \mid b)P(e \mid c)}$$
$$= \sum_b P(a)P(b \mid a)p(e \mid b)$$

Elimination in chains with evidence

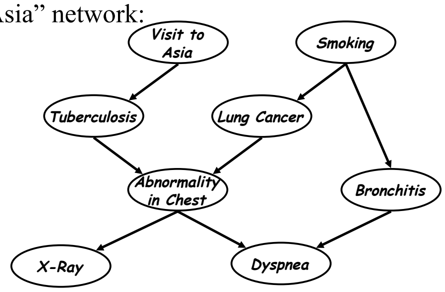
- Finally, we eliminate b

$$P(a,e) = \sum_b P(a)P(b \mid a)p(e \mid b)$$
$$= P(a)\underbrace{\sum_b P(b \mid a)p(e \mid b)}$$
$$= P(a)P(e \mid a)$$

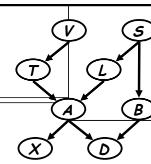
s_j=e, s_i=b, d=a
$P(s_i|d)$ = Sum(j) $P(s_i,s_j|d)$
$P(s_i,s_j|d)=P(s_i|s_j,d)P(s_j|d)$
P(e|b,a)=P(e|b) d-sep

A more complex example

- "Asia" network:



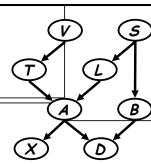
- We want to compute $P(d)$
- Need to eliminate: v,s,x,t,l,a,b



Initial factors

$$P(v)P(s)P(t \mid v)P(l \mid s)P(b \mid s)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$

- We want to compute $P(d)$
- Need to eliminate: v,s,x,t,l,a,b



Initial factors

$$\underline{P(v)}P(s)\underline{P(t \mid v)}P(l \mid s)P(b \mid s)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$

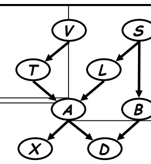
Eliminate: v

Compute: $$f_v(t) = \sum_v P(v)P(t \mid v)$$

$$\Rightarrow \underline{f_v(t)}P(s)P(l \mid s)P(b \mid s)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$

Note: $f_v(t) = P(t)$
In general, result of elimination is not necessarily a probability term

- We want to compute $P(d)$
- Need to eliminate: s,x,t,l,a,b



- Initial factors

$$P(v)P(s)P(t \mid v)P(l \mid s)P(b \mid s)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$
$$\Rightarrow f_v(t)\underline{P(s)}\underline{P(l \mid s)}\underline{P(b \mid s)}P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$

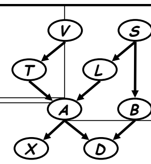
Eliminate: s

Compute: $$f_s(b,l) = \sum_s P(s)P(b \mid s)P(l \mid s)$$

$$\Rightarrow f_v(t)\underline{f_s(b,l)}P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$

Summing on s results in a factor with two arguments $f_s(b,l)$
In general, result of elimination may be a function of several variables

- We want to compute $P(d)$
- Need to eliminate: x,t,l,a,b



- Initial factors

$$P(v)P(s)P(t \mid v)P(l \mid s)P(b \mid s)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$
$$\Rightarrow f_v(t)P(s)P(l \mid s)P(b \mid s)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$
$$\Rightarrow f_v(t)f_s(b,l)P(a \mid t,l)\underline{P(x \mid a)}P(d \mid a,b)$$

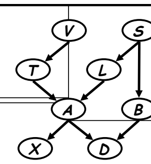
Eliminate: x

Compute: $$f_x(a) = \sum_x P(x \mid a)$$

$$\Rightarrow f_v(t)f_s(b,l)\underline{f_x(a)}P(a \mid t,l)P(d \mid a,b)$$

Note: $f_x(a) = 1$ for all values of a !! {P(x|a) + P(not x|a) =1}

- We want to compute $P(d)$
- Need to eliminate: t,l,a,b



- Initial factors

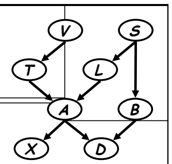
$$P(v)P(s)P(t \mid v)P(l \mid s)P(b \mid s)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$
$$\Rightarrow f_v(t)P(s)P(l \mid s)P(b \mid s)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$
$$\Rightarrow f_v(t)f_s(b,l)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$
$$\Rightarrow \underline{f_v(t)}f_s(b,l)f_x(a)\underline{P(a \mid t,l)}P(d \mid a,b)$$

Eliminate: t

Compute: $$f_t(a,l) = \sum_t f_v(t)P(a \mid t,l)$$

$$\Rightarrow f_s(b,l)f_x(a)\underline{f_t(a,l)}P(d \mid a,b)$$

- We want to compute $P(d)$
- Need to eliminate: l,a,b



- Initial factors

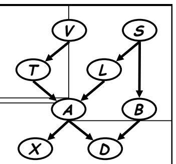
$$P(v)P(s)P(t\mid v)P(l\mid s)P(b\mid s)P(a\mid t,l)P(x\mid a)P(d\mid a,b)$$
$$\Rightarrow f_v(t)P(s)P(l\mid s)P(b\mid s)P(a\mid t,l)P(x\mid a)P(d\mid a,b)$$
$$\Rightarrow f_v(t)f_s(b,l)P(a\mid t,l)P(x\mid a)P(d\mid a,b)$$
$$\Rightarrow f_v(t)f_s(b,l)f_x(a)P(a\mid t,l)P(d\mid a,b)$$
$$\Rightarrow \underline{f_s(b,l)}f_x(a)\underline{f_T(a,l)}P(d\mid a,b)$$

Eliminate: l

Compute: $$f_l(a,b) = \sum_l f_s(b,l)f_T(a,l)$$

$$\Rightarrow \underline{f_l(a,b)}f_x(a)P(d\mid a,b)$$

- We want to compute $P(d)$
- Need to eliminate: a,b



- Initial factors

$$P(v)P(s)P(t\mid v)P(l\mid s)P(b\mid s)P(a\mid t,l)P(x\mid a)P(d\mid a,b)$$
$$\Rightarrow f_v(t)P(s)P(l\mid s)P(b\mid s)P(a\mid t,l)P(x\mid a)P(d\mid a,b)$$
$$\Rightarrow f_v(t)f_s(b,l)P(a\mid t,l)P(x\mid a)P(d\mid a,b)$$
$$\Rightarrow f_v(t)f_s(b,l)f_x(a)P(a\mid t,l)P(d\mid a,b)$$
$$\Rightarrow f_s(b,l)f_x(a)f_t(a,l)P(d\mid a,b)$$
$$\Rightarrow \underline{f_l(a,b)}\underline{f_x(a)}\underline{P(d\mid a,b)} \Rightarrow \underline{f_a(b,d)} \Rightarrow \underline{f_b(d)}$$

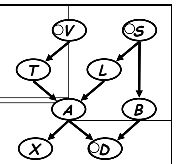
Eliminate: a and b:

$$f_a(b,d) = \sum_a f_l(a,b)f_x(a)p(d\mid a,b) \quad f_b(d) = \sum_b f_a(b,d)$$

Variable elimination

- To summarize, variable elimination can be represented as a sequence of **rewriting** operations
- Actual computations are done in elimination steps
- The overall amount of computation depends on order of elimination

Dealing with evidence



- We start by writing the factors:

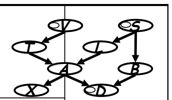
$$P(v)P(s)P(t\mid v)P(l\mid s)P(b\mid s)P(a\mid t,l)P(x\mid a)P(d\mid a,b)$$

- Since we know that $V = t$, we don't need to eliminate V
- Instead, we can replace the factors $P(V)$ and $P(T|V)$ with

$$f_{P(V)} = P(V = t) \quad f_{p(T|V)}(T) = P(T \mid V = t)$$

- These "select" the appropriate parts of the original factors given the evidence
- Note that $f_{p(V)}$ is a constant, and thus does not appear in elimination of other variables

Dealing with evidence



- Given evidence $V = t, S = f, D = t$
- Compute $P(L, V = t, S = f, D = t)$

- Initial factors, after setting evidence:

$$f_{P(v)} f_{P(s)} f_{P(t|v)}(t) f_{P(l|s)}(l) f_{P(b|s)}(b) P(a \mid t,l) P(x \mid a) f_{P(d|a,b)}(a,b)$$

- Eliminating x, we get

$$f_{P(v)} f_{P(s)} f_{P(t|v)}(t) f_{P(l|s)}(l) f_{P(b|s)}(b) P(a \mid t,l) f_x(a) f_{P(d|a,b)}(a,b)$$

- Eliminating t, we get

$$f_{P(v)} f_{P(s)} f_{P(l|s)}(l) f_{P(b|s)}(b) f_t(a,l) f_x(a) f_{P(d|a,b)}(a,b)$$

- Eliminating a, we get

$$f_{P(v)} f_{P(s)} f_{P(l|s)}(l) f_{P(b|s)}(b) f_a(b,l)$$

- Eliminating b, we get

$$f_{P(v)} f_{P(s)} f_{P(l|s)}(l) f_b(l)$$

Incremental Updating of BN: Pearl's message passing algorithm – Simple Chains

$e^+ \rightarrow T \rightarrow \cdots \rightarrow U \rightarrow X \rightarrow Y \rightarrow \cdots \rightarrow e^-$

$e = \{e^+, e^-\}$

e^+ Represents the "causal" evidence

e^- Represents the "evidential" evidence

Need to compute Bel(x)

Simple Chains cont.

$$
\begin{aligned}
Bel(x) &= P(x \mid e^+ e^-) \\
&= \frac{P(e^- \mid x\, e^+) P(x \mid e^+)}{P(e^- \mid e^+)} && \text{Bayes rule} \\
&= \alpha P(e^- \mid x\, e^+) P(x \mid e^+) && \text{Normalization} \\
&= \alpha P(e^- \mid x) P(x \mid e^+) && x \text{ d-sep } e^+ e^- \\
&= \alpha \cdot \lambda(x) \cdot \pi(x) && \text{Case 1 } e^+ \text{..-->x---> …} e^-
\end{aligned}
$$

The λ(x) and π(x) Messages

λ(x) represents the degree to which x might explain the evidential support. -- $P(e^-|X)$

π(x) represents the direct causal support for x. -- $P(X|e^+)$

Both λ(x) and π(x) can be calculated in terms of the λ and π values of the neighbors of x.

$e^+ \rightarrow T \rightarrow \cdots \rightarrow \underset{\pi}{U} \rightarrow X \rightarrow \underset{\lambda}{Y} \rightarrow \cdots \rightarrow e^-$

Computing λ(x) based on λ(y)

$e^+ \rightarrow T \rightarrow \cdots \rightarrow U \rightarrow X \rightarrow Y \rightarrow \cdots \rightarrow e^-$

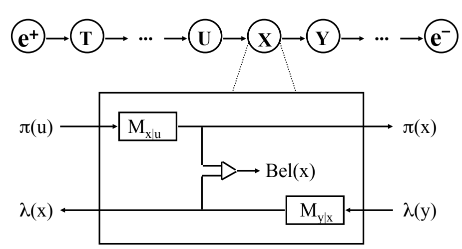
$$\begin{aligned}
\lambda(x) &= P(e^- \mid x) \\
&= \sum_y P(e^- \mid x, y) P(y \mid x) \\
&= \sum_y P(e^- \mid y) P(y \mid x) \qquad y \text{ d-sep } x, e^- \\
&= \sum_y \lambda(y) P(y \mid x) \\
&= \lambda(y) \bullet M_{y|x}
\end{aligned}$$

Computing π(x) based on π(u)

$e^+ \rightarrow T \rightarrow \cdots \rightarrow U \rightarrow X \rightarrow Y \rightarrow \cdots \rightarrow e^-$

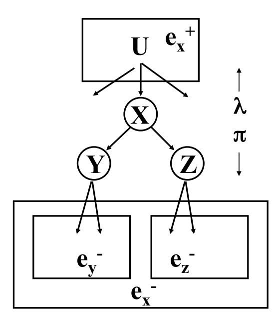
$$\begin{aligned}
\pi(x) &= P(x \mid e^+) \\
&= \sum_u P(x \mid u\, e^+) P(u \mid e^+) \\
&= \sum_u P(x \mid u) P(u \mid e^+) \qquad u \text{ d-sep } x, e^+ \\
&= \sum_u P(x \mid u) \pi(u) \\
&= \pi(u) \bullet M_{x \mid u}
\end{aligned}$$

Update scheme for chains

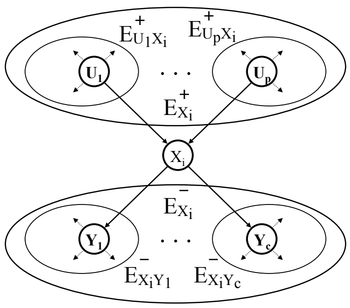


Belief Propagation in Trees

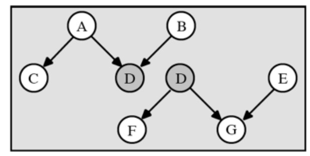
- Each node must combine the impact of λ-messages from several children.
- Each node must distribute a separate π-message to each child.



Propagation in Polytrees



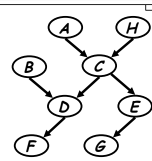
Polytrees



- In a polytree, there is just one (undirected path) between any two nodes. (avoids issue of double counting of evidence)
- A typical node D divides a polytree in two disconnected polytrees.
- Need to consider only evidence at the boundary of the graph.
  - A does not affect B so no messages need to go from A to B through D
  - F does not affect E so no messages need to go from F to E through G

Polytrees

Recall that a polytree is a network where there is at most one path from one variable to another



- **Theorem:** Inference in a polytree is linear in the size of the network

Heuristics for node ordering

- **Maximum cardinality search**: number the nodes from 1 to n, in increasing order, always assigning the next number to the vertex having the largest set of previously numbered neighbors. The elimination order is from n to 1.
- **Minimum discrepancy search**: at each point, eliminate the node that causes the fewest edges to be added to the induced graph.
- **Minimum size/weight search**: at each point, eliminate the node that causes the smallest clique to be created, where "small" is measured either in terms of number of nodes or number of entries in the factor.

Next Lecture

- Inference in Multiply Connected BNs