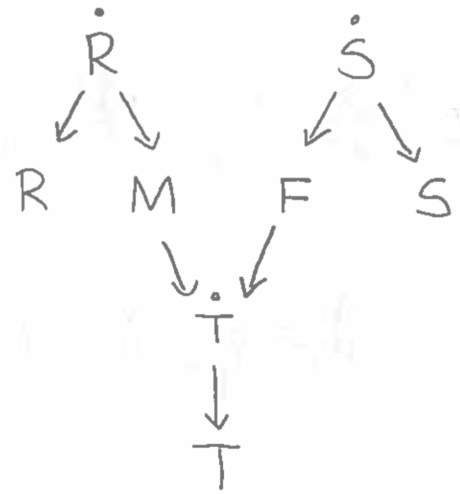
Exact Inference in Bayesian Networks

① Consider the following Bayesian network for blood types:



where:

- $\dot{R}, \dot{S}, \dot{T}$ are the blood genotypes for Rhonda, Sam, and Tim ($\in \{AA, AB, AO, BB, BO, OO\}$).
- R, S, T are the blood types for Rhonda, Sam, and Tim ($\in \{A, B, AB, O\}$)
- M, F are the genes passed to Tim by Rhonda (his Mother) and Sam (his Father). $M, F \in \{A, B, O\}$.

② What if we want to compute the probability of Rhonda having blood type AB, given that we know Tim has blood type A?

EXACT INFERENCE

③ We could try reasoning about it intuitively...

well, if Tim has type A,
then Rhonda has genotype AA, AB, AO, or BO...
and I guess AA is more likely than AB,
but AB is more likely than BO...

But it's a bit much. Plus what about harder questions like: what is the probability of Rhonda having blood type AB, given we know Tim has blood type A, AND Tim's cousin on his father's side has blood type O?

④ We need a method to automate complicated probability questions.

This problem is called <u>probabilistic inference</u>.

Note that $P(R=AB \mid T=A) = \dfrac{P(R=AB, T=A)}{P(T=A)}$, so if we can answer joint queries like $P(R=AB, T=A)$ and $P(T=A)$, then we can immediately also answer conditional queries

⑤ On the surface, it doesn't look that hard to solve:

$$P(R=AB, T=A) = \sum_{\dot{r}}\sum_{\dot{s}}\sum_{\dot{t}}\sum_{m}\sum_{f}\sum_{s} P(R=AB, T=A, \dot{r}, \dot{s}, \dot{t}, m, f, s)$$

But this expression has $6 \cdot 6 \cdot 6 \cdot 3 \cdot 3 \cdot 4 = 7776$ terms. While a computer could handle that, I certainly can't. Plus, as the number of variables in the Bayesian network increases, the number of terms grows exponentially.

⑥ We can begin by replacing the joint distribution by the factored representation assumed by the Bayesian network.

$P(r, t)$

$$= \sum_{\dot{r}}\sum_{\dot{s}}\sum_{\dot{t}}\sum_{m}\sum_{f}\sum_{s} P(\dot{r})P(\dot{s})\underbrace{P(r|\dot{r})}_{\text{this is just a function } g_3(r,\dot{r}) = P(r|\dot{r})}P(m|\dot{r})P(f|\dot{s})P(s|\dot{s})\underbrace{P(\dot{t}|m,f)}_{\text{this is just a function } g_6(\dot{t},m,f) = P(\dot{t}|m,f)}P(t|\dot{t})$$

(7) So $P(r,t)$ is just a sum of products:

$P(r,t)$

$$= \sum_{\dot{r}} \sum_{\dot{s}} \sum_{\dot{t}} \sum_{m} \sum_{f} \sum_{s} g_1(\dot{r}) g_2(\dot{s}) g_3(r,\dot{r}) g_4(m,\dot{r}) g_5(f,\dot{s}) g_6(s,\dot{s}) g_7(\dot{t},m,f) g_8(t,\dot{t})$$

How can we compute this efficiently?

(8) The first thing to notice is we can arrange the summations in whatever order we want, e.g.

$P(r,t)$

$$= \sum_{m} \sum_{f} \sum_{\dot{t}} \sum_{\dot{r}} \sum_{\dot{s}} \sum_{s} g_1(\dot{r}) g_2(\dot{s}) g_3(r,\dot{r}) g_4(m,\dot{r}) g_5(f,\dot{s}) g_6(s,\dot{s}) g_7(\dot{t},m,f) g_8(t,\dot{t})$$

(9) Now, we can move the innermost summation inward:

$P(r,t)$

$$= \sum_{m} \sum_{f} \sum_{\dot{t}} \sum_{\dot{r}} \sum_{\dot{s}} g_1(\dot{r}) g_2(\dot{s}) g_3(r,\dot{r}) g_4(m,\dot{r}) g_5(f,\dot{s}) g_7(\dot{t},m,f) g_8(t,\dot{t}) \sum_{s} g_6(s,\dot{s})$$

Note that $\sum_{s} g_6(s,\dot{s})$ is a function of $\dot{s}$, so we can define a new function $h_1(\dot{s}) = \sum_{s} g_6(s,\dot{s})$ and rewrite:

$$P(r,t) = \sum_{m} \sum_{f} \sum_{\dot{t}} \sum_{\dot{r}} \sum_{\dot{s}} g_1(\dot{r}) g_2(\dot{s}) g_3(r,\dot{r}) g_4(m,\dot{r}) g_5(f,\dot{s}) g_7(\dot{t},m,f) g_8(t,\dot{t}) h_1(\dot{s})$$

We have "eliminated" the summation over variable S (at least, we're hiding the fact that it exists.)

(10) We can continue to do this for each summation:

$$P(r,t) = \sum_m \sum_f \sum_{\dot{t}} \sum_{\dot{r}} g_1(\dot{r}) g_3(r,\dot{r}) g_4(m,\dot{r}) g_7(\dot{t},m,f) g_8(t,\dot{t}) \sum_{\dot{s}} g_2(\dot{s}) g_5(f,\dot{s}) h_1(\dot{s})$$

$$= \sum_m \sum_f \sum_{\dot{t}} \sum_{\dot{r}} g_1(\dot{r}) g_3(r,\dot{r}) g_4(m,\dot{r}) g_7(\dot{t},m,f) g_8(t,\dot{t})\, h_2(f)$$

$$\left[\text{letting } h_2(f) = \sum_{\dot{s}} g_2(\dot{s}) g_5(f,\dot{s}) h_1(\dot{s})\right]$$

$$= \sum_m \sum_f \sum_{\dot{t}} g_7(\dot{t},m,f) g_8(t,\dot{t}) h_2(f) \sum_{\dot{r}} g_1(\dot{r}) g_3(r,\dot{r}) g_4(m,\dot{r})$$

$$= \sum_m \sum_f \sum_{\dot{t}} g_7(\dot{t},m,f) g_8(t,\dot{t}) h_2(f) h_3(r,m)$$

$$\left[\text{letting } h_3(r,m) = \sum_{\dot{r}} g_1(\dot{r}) g_3(r,\dot{r}) g_4(m,\dot{r})\right]$$

$$= \sum_m \sum_f h_2(f) h_3(r,m) \sum_{\dot{t}} g_7(\dot{t},m,f) g_8(t,\dot{t})$$

$$= \sum_m \sum_f h_2(f) h_3(r,m) h_4(t,m,f)$$

$$\left[\text{letting } h_4(t,m,f) = \sum_{\dot{t}} g_7(\dot{t},m,f) g_8(t,\dot{t})\right]$$

$$= \sum_m h_3(r,m) \sum_f h_2(f) h_4(t,m,f)$$

$$= \sum_m h_3(r,m) h_5(t,m)$$

$$\left[\text{letting } h_5(t,m) = \sum_f h_2(f) h_4(t,m,f)\right]$$

$$= h_6(r,t)$$

$$\left[\text{letting } h_6(r,t) = \sum_m h_3(r,m) h_5(t,m)\right]$$

EXACT INFERENCE

(11) What have we done? Well, we have laid out a strategy for computing $P(r,t)$.

	num operations
- $\forall \dot{s}$, compute $h_1(\dot{s}) = \sum_{s} g_6(s,\dot{s})$	$\lvert S\rvert \cdot \lvert \dot{S}\rvert$
- $\forall f$, compute $h_2(f) = \sum_{\dot{s}} g_2(\dot{s}) g_5(f,\dot{s}) h_1(\dot{s})$	$\lvert \dot{S}\rvert \cdot \lvert F\rvert$
- $\forall r, m$, compute $h_3(r,m) = \sum_{\dot{r}} g_1(\dot{r}) g_3(r,\dot{r}) g_4(m,\dot{r})$	$\lvert \dot{R}\rvert \cdot \lvert R\rvert \cdot \lvert M\rvert$
- $\forall t, m, f$, compute $h_4(t,m,f) = \sum_{\dot{t}} g_7(\dot{t},m,f) g_8(t,\dot{t})$	$\lvert \dot{T}\rvert \cdot \lvert T\rvert \cdot \lvert M\rvert \cdot \lvert F\rvert$
- $\forall t, m$, compute $h_5(t,m) = \sum_{f} h_2(f) h_4(t,m,f)$	$\lvert F\rvert \cdot \lvert T\rvert \cdot \lvert M\rvert$
- $\forall r, t$, compute $h_6(r,t) = \sum_{m} h_3(r,m) h_5(t,m)$	$\lvert M\rvert \cdot \lvert R\rvert \cdot \lvert T\rvert$

Then return $P(r,t) = h_6(r,t)$.

Notice that:

$$|S|\cdot|\dot{S}| + |\dot{S}|\cdot|F| + |\dot{R}|\cdot|R|\cdot|M| + |\dot{T}|\cdot|T|\cdot|M|\cdot|F| + |F|\cdot|T|\cdot|M| + |M|\cdot|R|\cdot|T|$$

is much better than directly computing the sum of products, which is:

$$|R|\cdot|\dot{R}|\cdot|S|\cdot|\dot{S}|\cdot|T|\cdot|\dot{T}|\cdot|M|\cdot|F|$$

(414 operations, versus 124416 operations)

(12) If we execute this strategy, then we can get the probability that Rhonda has blood type AB and her son has type A.

h_1:

$\dot{s}$	$h_1(\dot{s})$
AA	1
AB	1
AO	1
BB	1
BO	1
OO	1

h_2:

f	$h_2(f)$
A	$\frac{1}{3}$
B	$\frac{1}{3}$
O	$\frac{1}{3}$

h_3:

r	m	$h_3(r,m)$
AB	A	$\frac{1}{12}$
AB	B	$\frac{1}{12}$
AB	O	0

h_4:

t	m	f	$h_4(t,m,f)$
A	A	A	1
A	A	B	0
A	A	O	1
A	B	A	0
A	B	B	0
A	B	O	0
A	O	A	1
A	O	B	0
A	O	O	0

h_5:

t	m	$h_5(t,m)$
A	A	$\frac{2}{3}$
A	B	0
A	O	$\frac{1}{3}$

h_6:

r	t	$h_6(r,t)$
AB	A	$\frac{1}{18}$

$P(R=AB, T=A) = \frac{1}{18}$!

We can use the same technique to compute $P(T=A) = \frac{1}{3}$. This gives us the probability that Rhonda's blood type is AB given Tim's blood type is A:

$$P(R=AB \mid T=A) = \frac{P(R=AB, T=A)}{P(T=A)} = \frac{1}{6}$$

EXACT INFERENCE

(13) To analyze the running time of this "variable elimination" algorithm, let d be the maximum size of any variable domain:

Compute all values of	num ops	
$h_1(\dot{s})$	$\lvert S\rvert \cdot \lvert \dot{S}\rvert \leq d^2$	$\leq d^4$
$h_2(f)$	$\lvert \dot{S}\rvert \cdot \lvert F\rvert \leq d^2$	$\leq d^4$
$h_3(r,m)$	$\lvert \dot{R}\rvert \cdot \lvert R\rvert \cdot \lvert M\rvert \leq d^3$	$\leq d^4$
$h_4(t,m,f)$	$\lvert \dot{T}\rvert \cdot \lvert T\rvert \cdot \lvert M\rvert \cdot \lvert F\rvert \leq d^4$	$\leq d^4$
$h_5(t,m)$	$\lvert F\rvert \cdot \lvert T\rvert \cdot \lvert M\rvert \leq d^3$	$\leq d^4$
$h_6(r,t)$	$\lvert M\rvert \cdot \lvert R\rvert \cdot \lvert T\rvert \leq d^3$	$\leq d^4$
		$\leq n \cdot d^4$

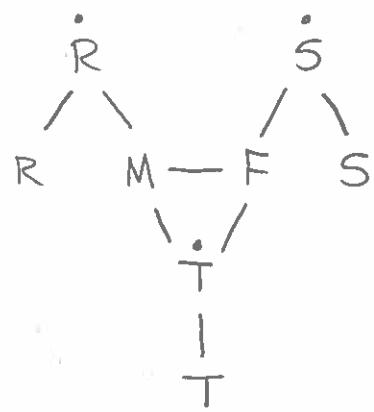
$\uparrow$ number of variables

(14) If we let w be the maximum number of variables in our h-functions, then the running time is

$$O(n \cdot d^{w+1})$$

This quantity w is called the <u>width</u> of the elimination order.

(15) It can be convenient to view this graphically. Create an undirected graph whose vertices are the variables. There is an edge between two vertices iff they appear in the same g-function:



$$P(r|t) = \sum_m \sum_f \sum_{\dot{t}} \sum_{\dot{r}} \sum_{\dot{s}} \sum_s g_1(\dot{r}) g_2(\dot{s}) g_3(r,\dot{r}) g_4(m,\dot{r}) g_5(f,\dot{s}) g_6(s,\dot{s}) g_7(\dot{t},m,f) g_8(t,\dot{t})$$

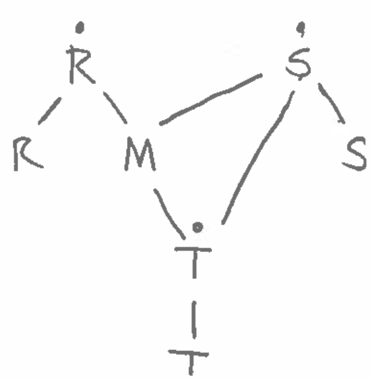
---

(16) Suppose we choose to first eliminate F:

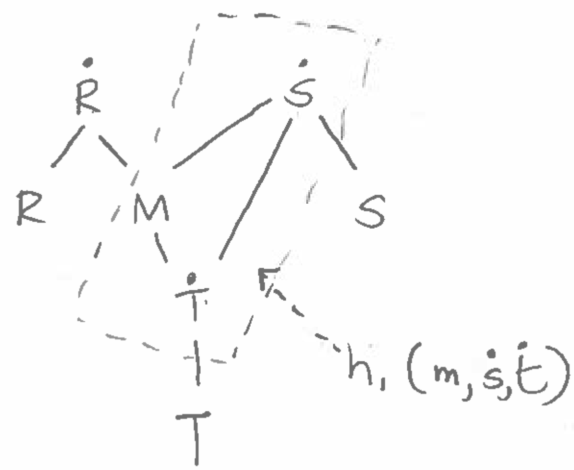
$$P(r|t) = \sum_m \sum_{\dot{t}} \sum_{\dot{r}} \sum_{\dot{s}} \sum_s g_1(\dot{r}) g_2(\dot{s}) g_3(r,\dot{r}) g_4(m,\dot{r}) g_6(s,\dot{s}) g_8(t,\dot{t}) h_1(m,\dot{s},\dot{t})$$

where $h_1(m,\dot{s},\dot{t}) = \sum_f g_5(f,\dot{s}) g_7(\dot{t},m$

We create an h-function over $m, \dot{s}, \dot{t}$. Graphically, these are all the nodes that F was connected to. The new graph becomes:



(17) Note that "eliminating" F corresponds (graphically) to removing F and connecting all the nodes it was connected to. The size of this new <u>clique</u> corresponds to the number of variables in the new h-function:

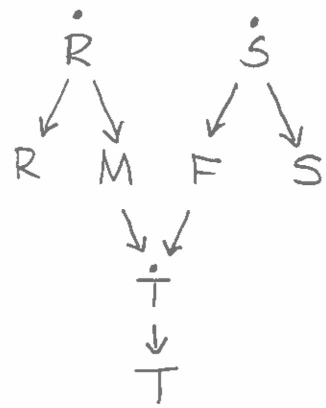


(18) Therefore, we can compute the width of an elimination order as a graph algorithm:

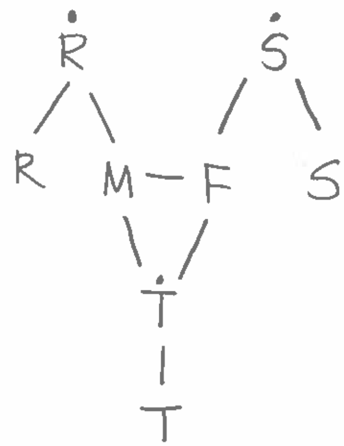
- → let $G = (V, E)$, where V corresponds to the variables and two vertices are connected if they co-occur in a g-function.
- → $w = 0$
- → iterate through $V_1, \ldots, V_n$:
  - → $w = \max(w, \text{num vertices } V_i \text{ is connected to})$
  - → remove V_i from G and connect all its neighbors
- → return w

(19) So given a Bayesian network, we can predict the runtime of probabilistic inference based solely on its graph structure, and the domain size of its variables.

(i) Take the network:



(ii) Create an undirected graph where two vars are connected if they co-appear in a conditional probability function



this corresponds to connecting all parents of each node

this is called a moral graph

(iii) Compute the width w of your selected elimination order (as described in (18))

(iv) The runtime of variable elimination is $O(n \cdot d^{w+1})$, where n is the number of variables, and d is the maximum domain size.

(20) This means that certain Bayesian networks enjoy good performance, like the following:

$$\begin{tikzcd}
H_1 \arrow[r] \arrow[d] & H_2 \arrow[r] \arrow[d] & H_3 \arrow[r] \arrow[d] & \cdots \arrow[r] & H_K \arrow[d] \\
O_1 & O_2 & O_3 & & O_K
\end{tikzcd}$$

which is sometimes called a Hidden Markov Model and has an elimination order of width 1 (i.e. $O_1, H_1, O_2, H_2 \ldots$)

Thus the runtime of variable elimination is $O(k \cdot d^2)$. If the variables are all binary (or constant-size domain), then this is $O(k)$.

(21) Unfortunately, others have guaranteed poor performance, like this grid network:

$$\begin{tikzcd}
X_{11} \arrow[r] \arrow[d] & X_{12} \arrow[r] \arrow[d] & \cdots \arrow[r] & X_{1m} \arrow[d] \\
X_{21} \arrow[r] \arrow[d] & X_{22} \arrow[r] \arrow[d] & \cdots \arrow[r] & X_{2m} \arrow[d] \\
\vdots \arrow[d] & \vdots \arrow[d] & & \vdots \arrow[d] \\
X_{m1} \arrow[r] & X_{m2} \arrow[r] & \cdots \arrow[r] & X_{mm}
\end{tikzcd}$$

whose best width is $O(m)$, thus the runtime of VE is $O(m^2 \cdot d^{m+1})$