CSE 3402: Intro to Artificial Intelligence
Reasoning under Uncertainty
Value Elimination

- Readings: Russell & Norvig Chapter 13 and Chapter 14 Sec. 1 to 4 (in both 3rd and 2nd editions)

Inference in Bayes Nets

- Given a Bayes net

  $Pr(X_1, X_2, \ldots, X_n)$

  $= Pr(X_n \mid Par(X_n)) * Pr(X_{n-1} \mid Par(X_{n-1}))$
  $* \cdots * Pr(X_1 \mid Par(X_1))$
- And some evidence E = {a set of values for some of the variables} we want to compute the new probability distribution

  $Pr(X_k \mid E)$
- That is, we want to figure our Pr(X_k = d |E) for all $d \in Dom[X_k]$

Inference in Bayes Nets

- Other types of examples are, computing probability of different diseases given symptoms, computing probability of hail storms given different metrological evidence, etc.
- In such cases getting a good estimate of the probability of the unknown event allows us to respond more effectively (gamble rationally)

Inference in Bayes Nets

- In the Alarm example we have

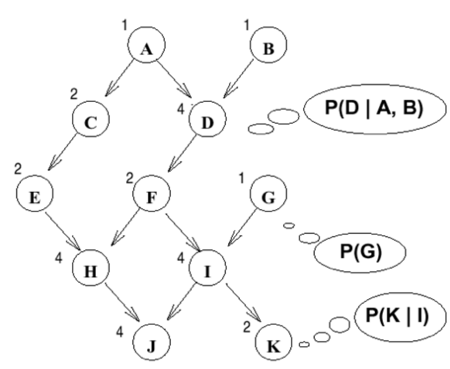
  Pr(Breakin,Earthquake,Radio,Sound) =
  Pr(Earthquake) * Pr(BreakIn) *
  Pr(Radio|Earthquake) *
  Pr(Sound|BreakIn,Earthquake)
- And, e.g., we want to compute things like Pr(BreakIn=True| Radio=false, Sound=true)

Variable Elimination

- Variable elimination uses the product decomposition and the summing out rule to compute posterior probabilities from the information (CPTs) already in the network.

Example (Binary valued Variables)

Pr(A,B,C,D,E,F,G,H,I,J,K) =
Pr(A)
x Pr(B)
x Pr(C|A)
x Pr(D|A,B)
x Pr(E|C)
x Pr(F|D)
x Pr(G)
x Pr(H|E,F)
x Pr(I|F,G)
x Pr(J|H,I)
x Pr(K|I)



Example

Pr(A,B,C,D,E,F,G,H,I,J,K) =
Pr(A)Pr(B)Pr(C|A)Pr(D|A,B)Pr(E|C)Pr(F|D)Pr(G)
Pr(H|E,F)Pr(I|F,G)Pr(J|H,I)Pr(K|I)

Say that E = {H=true, I=false}, and we want to know Pr(D|h,i) (h: H is true, -h: H is false)

1. Write as a sum for each value of D

   $\sum_{A,B,C,E,F,G,J,K}$ Pr(A,B,C,d,E,F,h,-i,J,K) = Pr(d,h,-i)

   $\sum_{A,B,C,E,F,G,J,K}$ Pr(A,B,C,-d,E,F,h,-i,J,K) = Pr(-d,h,-i)

Example

2. Pr(d,h,-i) + Pr(-d,h,-i) = Pr(h,-i)
3. Pr(d|h,-i) = Pr(d,h,-i)/Pr(h,-i)
   Pr(-d|h,-i) = Pr(-d,h,-i)/Pr(h,-i)

So we only need to compute Pr(d,h,-i) and Pr(-d,h,-i) and then normalize to obtain the conditional probabilities we want.

Example

$Pr(d,h,-i) = \sum_{A,B,C,E,F,G,J,K} Pr(A,B,C,d,E,F,h,-i,J,K)$

Use Bayes Net product decomposition to rewrite summation:

$\sum_{A,B,C,E,F,G,J,K} Pr(A,B,C,d,E,F,h,-i,J,K)$
$= \sum_{A,B,C,E,F,G,J,K} Pr(A)Pr(B)Pr(C|A)Pr(d|A,B)Pr(E|C)$
$Pr(F|d)Pr(G)Pr(h|E,F)Pr(-i|F,G)Pr(J|h,-i)$
$Pr(K|-i)$

Now rearrange summations so that we are not summing over that do not depend on the summed variable.

Example

$= \sum_{A,}\sum_{B,}\sum_{C,}\sum_{E,}\sum_{F,}\sum_{G,}\sum_{J,}\sum_{K} Pr(A)Pr(B)Pr(C|A)Pr(d|A,B)Pr(E|C)$
$Pr(F|d)Pr(G)Pr(h|E,F)Pr(-i|F,G)Pr(J|h,-i)$
$Pr(K|-i)$

$= \sum_A Pr(A) \sum_B Pr(B) \sum_C Pr(C|A)Pr(d|A,B) \sum_E Pr(E|C)$
$\sum_F Pr(F|d) \sum_G Pr(G)Pr(h|E,F)Pr(-i|F,G) \sum_J Pr(J|h,-i)$
$\sum_K Pr(K|-i)$

$= \sum_A Pr(A) \sum_B Pr(B)\, Pr(d|A,B) \sum_C Pr(C|A) \sum_E Pr(E|C)$
$\sum_F Pr(F|d)\, Pr(h|E,F) \sum_G Pr(G)\, Pr(-i|F,G) \sum_J Pr(J|h,-i)$
$\sum_K Pr(K|-i)$

Example

= $\sum_{A,}\sum_{B,}\sum_{C,}\sum_{E,}\sum_{F,}\sum_{G,}\sum_{J,}\sum_{K}$ Pr(A)Pr(B)Pr(C|A)Pr(d|A,B)Pr(E|C)
Pr(F|d)Pr(G)Pr(h|E,F)Pr(−i|F,G)Pr(J|h,−i)
Pr(K|−i)

= $\sum_A$ Pr(A) $\sum_B$ Pr(B) $\sum_C$ Pr(C|A)Pr(d|A,B) $\sum_E$ Pr(E|C)
$\sum_F$ Pr(F|d) $\sum_G$ Pr(G)Pr(h|E,F)Pr(−i|F,G) $\sum_J$ Pr(J|h,−i)
$\sum_K$ Pr(K|−i)

= $\sum_A$ Pr(A) $\sum_B$ Pr(B) Pr(d|A,B) $\sum_C$ Pr(C|A) $\sum_E$ Pr(E|C)
$\sum_F$ Pr(F|d) Pr(h|E,F)$\sum_G$ Pr(G) Pr(−i|F,G) $\sum_J$ Pr(J|h,−i)
$\sum_K$ Pr(K|−i)

Example

- Now start computing.

$\sum_A$ Pr(A) $\sum_B$ Pr(B) Pr(d|A,B) $\sum_C$ Pr(C|A) $\sum_E$ Pr(E|C)
$\sum_F$ Pr(F|d) Pr(h|E,F)$\sum_G$ Pr(G) Pr(−i|F,G)
$\sum_J$ Pr(J|h,−i)
$\sum_K$ Pr(K|−i)

$\sum_K$ Pr(K|−i) = Pr(k|−i) + Pr(−k|−i) = c_1

$\sum_J$Pr(J|h,−i) c_1 = c_1 $\sum_J$Pr(J|h,−i)
= c_1 (Pr(j|h,−i) + Pr(−j|h,−i))
= $c_1 c_2$

Example

- $\sum_A$ Pr(A) $\sum_B$ Pr(B) Pr(d|A,B) $\sum_C$ Pr(C|A) $\sum_E$ Pr(E|C)
  $\sum_F$ Pr(F|d) Pr(h|E,F)$\sum_G$ Pr(G) Pr(−i|F,G)
  $\sum_J$ Pr(J|h,−i)
  $\sum_K$ Pr(K|−i)

  $c_1 c_2$ $\sum_G$ Pr(G) Pr(−i|F,G)
  = $c_1 c_2$(Pr(g)Pr(−i|F,g) + Pr(−g)Pr(−i|F,−g))

  !!But Pr(−i|F,g) depends on the value of F, so this is not a single number.

Example

- Try the other order of summing.

$\sum_A$ Pr(A) $\sum_B$ Pr(B) Pr(d|A,B) $\sum_C$ Pr(C|A) $\sum_E$ Pr(E|C)
$\sum_F$ Pr(F|d) Pr(h|E,F)$\sum_G$ Pr(G) Pr(−i|F,G)
$\sum_J$ Pr(J|h,−i)
$\sum_K$ Pr(K|−i)

=

Pr(a) $\sum_B$ Pr(B) Pr(d|a,B) $\sum_C$ Pr(C|a) $\sum_E$ Pr(E|C)
$\sum_F$ Pr(F|d) Pr(h|E,F)$\sum_G$ Pr(G) Pr(−i|F,G)
$\sum_J$ Pr(J|h,−i)
$\sum_K$ Pr(K|−i)

\+

Pr(−a) $\sum_B$ Pr(B) Pr(d|−a,B) $\sum_C$ Pr(C|−a) $\sum_E$ Pr(E|C)
$\sum_F$ Pr(F|d) Pr(h|E,F)$\sum_G$ Pr(G) Pr(−i|F,G)
$\sum_J$ Pr(J|h,−i)
$\sum_K$ Pr(K|−i)

Example

=

$Pr(a)Pr(b)\ Pr(d|a,b)\ \sum_C Pr(C|a)\ \sum_E Pr(E|C)$
$\quad \sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$
$\quad \sum_J Pr(J|h,-i)$
$\quad \sum_K Pr(K|-i)$

+

$Pr(a)Pr(-b)\ Pr(d|a,-b)\ \sum_C Pr(C|a)\ \sum_E Pr(E|C)$
$\quad \sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$
$\quad \sum_J Pr(J|h,-i)$
$\quad \sum_K Pr(K|-i)$

+

$Pr(-a)Pr(b)\ Pr(d|-a,b)\ \sum_C Pr(C|-a)\ \sum_E Pr(E|C)$
$\quad \sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$
$\quad \sum_J Pr(J|h,-i)$
$\quad \sum_K Pr(K|-i)$

+

$Pr(-a)Pr(-b)\ Pr(d|-a,-b)\ \sum_C Pr(C|-a)\ \sum_E Pr(E|C)$
$\quad \sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$
$\quad \sum_J Pr(J|h,-i)$
$\quad \sum_K Pr(K|-i)$

Example

=

Yikes! The size of the sum is doubling as we expand each variable (into -v and v). This approach has exponential complexity.

But let's look a bit closer.

Example

$$=$$
$$Pr(a)Pr(b)\ Pr(d|a,b)\ \sum_C Pr(C|a)\ \sum_E Pr(E|C)$$
$$\sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$$
$$\sum_J Pr(J|h,-i)$$
$$\sum_K Pr(K|-i)$$

Repeated subterm

$$+$$
$$Pr(a)Pr(-b)\ Pr(d|a,-b)\ \sum_C Pr(C|a)\ \sum_E Pr(E|C)$$
$$\sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$$
$$\sum_J Pr(J|h,-i)$$
$$\sum_K Pr(K|-i)$$
$$+$$
$$Pr(-a)Pr(b)\ Pr(d|-a,b)\ \sum_C Pr(C|-a)\ \sum_E Pr(E|C)$$
$$\sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$$
$$\sum_J Pr(J|h,-i)$$
$$\sum_K Pr(K|-i)$$

Repeated subterm

$$+$$
$$Pr(-a)Pr(-b)\ Pr(d|-a,-b)\ \sum_C Pr(C|-a)\ \sum_E Pr(E|C)$$
$$\sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$$
$$\sum_J Pr(J|h,-i)$$
$$\sum_K Pr(K|-i)$$

Dynamic Programming

- If we store the value of the subterms, we need only compute them once.

Dynamic Programming

$=$
$Pr(a)Pr(b)\ Pr(d|a,b)\ \sum_C Pr(C|a)\ \sum_E Pr(E|C)$
$\quad \sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$
$\quad \sum_J Pr(J|h,-i)$
$\quad \sum_K Pr(K|-i)$ — f_1
$+$
$Pr(a)Pr(-b)\ Pr(d|a,-b)\ \sum_C Pr(C|a)\ \sum_E Pr(E|C)$
$\quad \sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$
$\quad \sum_J Pr(J|h,-i)$
$\quad \sum_K Pr(K|-i)$
$+$
$Pr(-a)Pr(b)\ Pr(d|-a,b)\ \sum_C Pr(C|-a)\ \sum_E Pr(E|C)$
$\quad \sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$
$\quad \sum_J Pr(J|h,-i)$
$\quad \sum_K Pr(K|-i)$ — f_2
$+$
$Pr(-a)Pr(-b)\ Pr(d|-a,-b)\ \sum_C Pr(C|-a)\ \sum_E Pr(E|C)$
$\quad \sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$
$\quad \sum_J Pr(J|h,-i)$
$\quad \sum_K Pr(K|-i)$

$= c_1f_1 + c_2f_1 + c_3f_2 + c_3f_2$

$c_1 = Pr(a)Pr(b)\ Pr(d|a,b)$

$c_2 = Pr(a)Pr(-b)\ Pr(d|a,-b)$

$c_3 = Pr(a)Pr(-b)\ Pr(d|a,-b)$

$c_4 = Pr(a)Pr(-b)\ Pr(d|a,-b)$

Dynamic Programming

$f_1 = \sum_C Pr(C|a)\ \sum_E Pr(E|C)$
$\quad \sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$
$\quad \sum_J Pr(J|h,-i)$
$\quad \sum_K Pr(K|-i)$

$= Pr(c|a)\ \sum_E Pr(E|c)$
$\quad \sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$
$\quad \sum_J Pr(J|h,-i)$
$\quad \sum_K Pr(K|-i)$ — Repeated subterm
$+$
$Pr(-c|a)\ \sum_E Pr(E|-c)$
$\quad \sum_F Pr(F|d)\ Pr(h|E,F) \sum_G Pr(G)\ Pr(-i|F,G)$
$\quad \sum_J Pr(J|h,-i)$
$\quad \sum_K Pr(K|-i)$

Dynamic Programming

- So within the computation of the subterms we obtain more repeated smaller subterms.
- The core idea of dynamic programming is to remember all "smaller" computations, so that they can be reused.
- This can convert an exponential computation into one that takes only polynomial time.
- Variable elimination is a dynamic programming technique that computes the sum from the bottom up (starting with the smaller subterms and working its way up to the bigger terms).

Relevant (return to this later)

- A brief aside is to also note that in the sum
  $\sum_A$ Pr(A) $\sum_B$ Pr(B) Pr(d|A,B) $\sum_C$ Pr(C|A) $\sum_E$ Pr(E|C)
  $\sum_F$ Pr(F|d) Pr(h|E,F)$\sum_G$ Pr(G) Pr(-i|F,G)
  $\sum_J$ Pr(J|h,-i)
  $\sum_K$ Pr(K|-i)

  we have that $\sum_K$ Pr(K|-i) = 1 (Why?), thus
  $\sum_J$ Pr(J|h,-i)$\sum_K$ Pr(K|-i) = $\sum_J$ Pr(J|h,-i)

  Furthermore $\sum_J$ Pr(J|h,-i) = 1.

  So we could drop these last two terms from the computation---J and K are not relevant given our query D and our evidence -i and -h. For now we keep these terms.

Variable Elimination (VE)

- VE works from the inside out, summing out K, then J, then G, …, as we tried to before.
- When we tried to sum out G

$\sum_A$ Pr(A) $\sum_B$ Pr(B) Pr(d|A,B) $\sum_C$ Pr(C|A) $\sum_E$ Pr(E|C)
$\sum_F$ Pr(F|d) Pr(h|E,F)$\sum_G$ Pr(G) Pr(−i|F,G)
$\sum_J$ Pr(J|h,−i)
$\sum_K$ Pr(K|−i)

c_1c_2 $\sum_G$ Pr(G) Pr(−i|F,G)
= c_1c_2(Pr(g)Pr(−i|F,g) + Pr(−g)Pr(−i|F,−g))

we found that Pr(−i|F,−g) depends on the value of F, it wasn't a single number.

- However, we can still continue with the computation by computing two different numbers, one for each value of F (−f, f)!

CSE 3402 W12 Fahiem Bacchus & Yves Lesperance

23

Variable Elimination (VE)

- t(−f) = c_1c_2 $\sum_G$ Pr(G) Pr(−i|−f,G)

  t(f) = c_1c_2($\sum_G$ Pr(G) Pr(−i|f,G)

  - t(−f) = c_1c_2(Pr(g)Pr(−i|−f,g) + Pr(−g)Pr(−i|−f,−g))
  - t(−f) = c_1c_2(Pr(g)Pr(−i|f,g) + Pr(−g)Pr(−i|f,−g))

- Now we sum out F

CSE 3402 W12 Fahiem Bacchus & Yves Lesperance

24

Variable Elimination (VE)

- $\sum_A$ Pr(A) $\sum_B$ Pr(B) Pr(d|A,B) $\sum_C$ Pr(C|A) $\sum_E$ Pr(E|C)
  $\sum_F$ Pr(F|d) Pr(h|E,F)$\sum_G$ Pr(G) Pr(−i|F,G)
  $\sum_J$ Pr(J|h,−i)
  $\sum_K$ Pr(K|−i)

c_1c_2 $\sum_F$ Pr(F|d) Pr(h|E,F)$\sum_G$ Pr(G) Pr(−i|F,G)

= c_1c_2(Pr(f|d) Pr(h|E,f)($\sum_G$ Pr(G) Pr(−i|f,G))
+Pr(−f|d)Pr(h|E,−f)($\sum_G$ Pr(G)Pr(−i|−f,G))

= c_1c_2 $\sum_F$Pr(F|d) Pr(h|E,F)t(F)

t(f), t(−f)

Variable Elimination (VE)

- c_1c_2(Pr(f|d) Pr(h|E,f)t(f)
  +Pr(−f|d)Pr(h|E,−f)t(−f)
- This is a function of E, so we obtain two new numbers

  s(e) = c_1c_2(Pr(f|d) Pr(h|e,f)t(f)
  +Pr(−f|d)Pr(h|e,−f)t(−f)

  s(−e) = c_1c_2(Pr(f|d) Pr(h|−e,f)t(f)
  +Pr(−f|d)Pr(h|−e,−f)t(−f)

Variable Elimination (VE)

- On summing out E we obtain two numbers, or a function of C. Then a function of B, then a function of A. On finally summing out A we obtain the single number we wanted to compute which is Pr(d,h,-i).
- Now we can repeat the process to compute Pr(-d,h,-i).
- But instead of doing it twice, we can simply regard D as an variable in the computation.
- This will result in some computations depending on the value of D, and we obtain a different number for each value of D.
- Proceeding in this manner, summing out A will yield a function of D. (I.e., a number for each value of D).

Variable Elimination (VE)

- In general, at each stage VE will be compute a table of numbers: one number for each different instantiation of the variables that are in the sum.
- The size of these tables is exponential in the number of variables appearing in the sum, e.g.,

  $\sum_F \Pr(F|D)\ \Pr(h|E,F)t(F)$

  depends on the value of D and E, thus we will obtain |Dom[D]||Dom[E]| different numbers in the resulting table.

Factors

- we call these tables of values computed by VE factors.
- Note that the original probabilities that appear in the summation, e.g., P(C|A), are also tables of values (one value for each instantiation of C and A).
- Thus we also call the original CPTs factors.

- Each factor is a function of some variables, e.g., P(C|A) = f(A,C): it maps each value of its arguments to a number.
  - A tabular representation is exponential in the number of variables in the factor.

Operations on Factors

- If we examine the inside-out summation process we see that various operations occur on factors.

- Notation: f($\underline{X}$,$\underline{Y}$) denotes a factor over the variables $\underline{X} \cup \underline{Y}$ (where $\underline{X}$ and $\underline{Y}$ are sets of variables)

The Product of Two Factors

- Let f(**X**,**Y**) & g(**Y**,**Z**) be two factors with variables **Y** in common
- The *product* of f and g, denoted h = f x g (or sometimes just h = fg), is defined:

h(**X**,**Y**,**Z**) = f(**X**,**Y**) x g(**Y**,**Z**)

f(A,B)		g(B,C)		h(A,B,C)			
ab	0.9	bc	0.7	abc	0.63	ab~c	0.27
a~b	0.1	b~c	0.3	a~bc	0.08	a~b~c	0.02
~ab	0.4	~bc	0.8	~abc	0.28	~ab~c	0.12
~a~b	0.6	~b~c	0.2	~a~bc	0.48	~a~b~c	0.12

Summing a Variable Out of a Factor

- Let f(X,**Y**) be a factor with variable X (**Y** is a set)
- We *sum out* variable X from f to produce a new factor h = Σ_X f, which is defined:

h(**Y**) = $\Sigma_{x \in Dom(X)}$ f(x,**Y**)

f(A,B)		h(B)	
ab	0.9	b	1.3
a~b	0.1	~b	0.7
~ab	0.4		
~a~b	0.6		

Restricting a Factor

- Let f(X,**Y**) be a factor with variable X (**Y** is a set)
- We *restrict* factor f *to* X=a by setting X to the value x and "deleting" incompatible elements of f' s domain . Define $h = f_{X=a}$ as: h(**Y**) = f(a,**Y**)

f(A,B)		h(B) = $f_{A=a}$	
ab	0.9	b	0.9
a~b	0.1	~b	0.1
~ab	0.4		
~a~b	0.6		

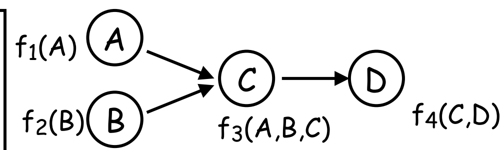
Variable Elimination the Algorithm

Given query var Q, evidence vars **E** (set of variables observed to have values **e**), remaining vars **Z**. Let F be original CPTs.

1. Replace each factor f∈F that mentions a variable(s) in **E** with its restriction $f_{E=e}$ (this might yield a "constant" factor)
2. For each Z_j—in the order given–eliminate $Z_j \in$ **Z** as follows:
   (a) Compute new factor $g_j = \sum_{Z_j} f_1 \times f_2 \times \ldots \times f_k$, where the f_i are the factors in F that include Z_j
   (b) Remove the factors f_i (that mention Z_j) from F and add new factor g_j to F
3. The remaining factors refer only to the query variable Q. Take their product and normalize to produce Pr(Q|E)

VE: Example

Factors: $f_1(A)$ $f_2(B)$ $f_3(A,B,C)$ $f_4(C,D)$
Query: P(A)?
Evidence: D = d
Elim. Order: C, B



Restriction: replace $f_4(C,D)$ with $f_5(C) = f_4(C,d)$

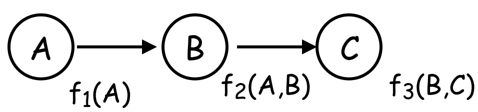
Step 1: Compute & Add $f_6(A,B) = \Sigma_C f_5(C) f_3(A,B,C)$
Remove: $f_3(A,B,C)$, $f_5(C)$

Step 2: Compute & Add $f_7(A) = \Sigma_B f_6(A,B) f_2(B)$
Remove: $f_6(A,B)$, $f_2(B)$

Last factors: $f_7(A)$, $f_1(A)$. The product $f_1(A) \times f_7(A)$ is (unnormalized) posterior. So… $P(A|d) = \alpha f_1(A) \times f_7(A)$ where $\alpha = 1/\sum_A f_1(A) f_7(A)$

Numeric Example

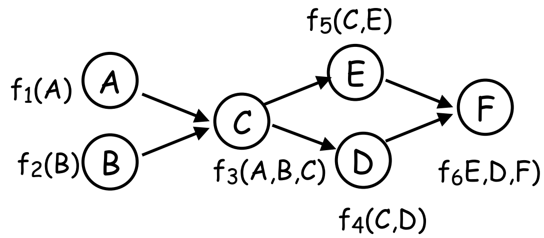
- Here's the example with some numbers



$f_1(A)$		$f_2(A,B)$		$f_3(B,C)$		$f_4(B)$ $\Sigma_A f_2(A,B) f_1(A)$		$f_5(C)$ $\Sigma_B f_3(B,C) f_4(B)$	
a	0.9	ab	0.9	bc	0.7	b	0.85	c	0.625
~a	0.1	a~b	0.1	b~c	0.3	~b	0.15	~c	0.375
		~ab	0.4	~bc	0.2				
		~a~b	0.6	~b~c	0.8				

VE: Buckets as a Notational Device

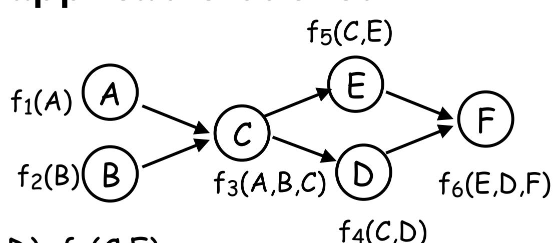
Ordering:
C,F,A,B,E,D



1. C:
2. F:
3. A:
4. B:
5. E:
6. D:

VE: Buckets—Place Original Factors in first applicable bucket.

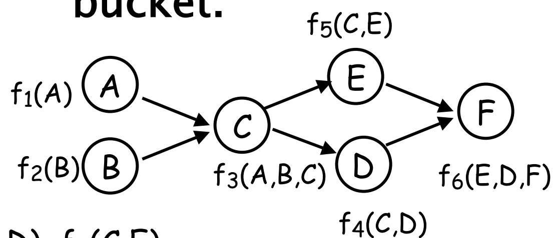
Ordering:
C,F,A,B,E,D



1. C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$
4. B: $f_2(B)$
5. E:
6. D:

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D

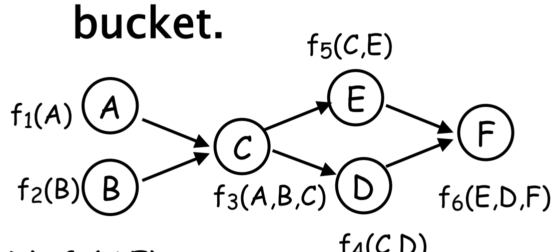


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E:
6. D:

1. $\Sigma_C\, f_3(A,B,C), f_4(C,D), f_5(C,E)$
   $= f_7(A,B,D,E)$

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D

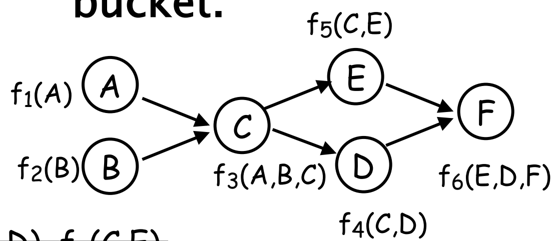


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E: $f_8(E,D)$
6. D:

2. $\Sigma_F\, f_6(E,D,F) = f_8(E,D)$

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D

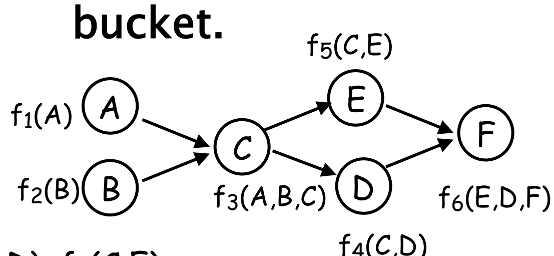


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. B: $f_2(B)$, f_9(B,D,E)
5. E: $f_8(E,D)$
6. D:

3. $\Sigma_A f_1(A), f_7(A,B,D,E)$
   $= f_9(B,D,E)$

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D

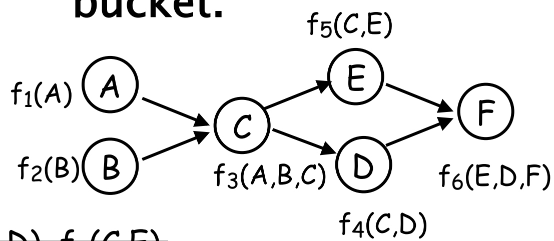


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. ~~B: $f_2(B)$, $f_9(B,D,E)$~~
5. E: $f_8(E,D)$, $f_{10}(D,E)$
6. D:

4. $\Sigma_B f_2(B), f_9(B,D,E)$
   $= f_{10}(D,E)$

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D



1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. ~~B: $f_2(B)$, $f_9(B,D,E)$~~
5. ~~E: $f_8(E,D)$, $f_{10}(D,E)$~~
6. D: $f_{11}(D)$

5. Σ_E $f_8(E,D)$, $f_{10}(D,E)$
   $= f_{11}(D)$

f_{11} is he final answer, once we normalize it.

Complexity of VE

- VE with a given elimination ordering requires $2^{O(k)}$ space where k is the number of variables in the largest factor in that elimination ordering (k is related to the *treewidth*).
- Time complexity is also $2^{O(k)}$.
- In the worst case, k is the number of variables n, so no better than storing probability for all variable assignments.
- Finding the best variable ordering is NP hard in general, but there are heuristics that work well, especially for restricted BN topologies.