Inference in Bayesian Networks

BMI/CS 576
www.biostat.wisc.edu/bmi576/
Colin Dewey
cdewey@biostat.wisc.edu
Fall 2008

The Inference Task in Bayesian Networks

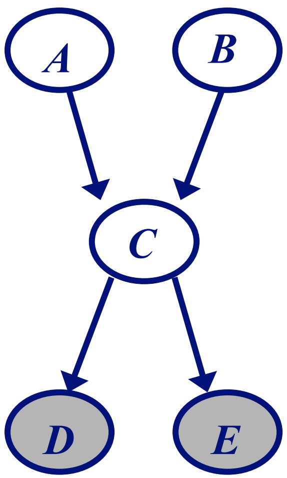
Given: values for some variables in the network (*evidence*),
and a set of *query* variables

Do: compute the posterior distribution over the query variables

- variables that are neither evidence variables nor query variables are *hidden* variables
- the BN representation is flexible enough that any set can be the evidence variables and any set can be the query variables

Inference by Enumeration

- suppose we have the following BN with 5 Boolean variables
- we use ***a*** to denote ***A***=true, and **¬*a*** to denote ***A***=false
- suppose we're given the query: $P(a \mid d, e)$
  "probability *a* is true given that *d* and *e* are true"
- from the graph structure we can compute this by:



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

normalizing constant

sum over possible values for B and C variables

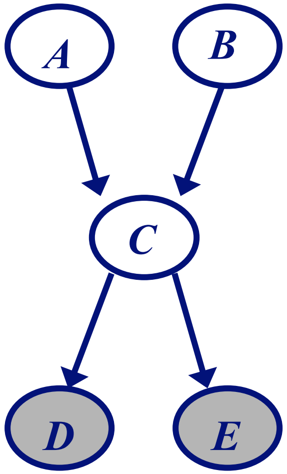
Inference by Enumeration

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

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

P(A)
0.1

P(B)
0.1



A	*B*	*P(C)*
T	T	0.8
T	F	0.5
F	T	0.2
F	F	0.7

C	*P(D)*
T	0.4
F	0.3

C	*P(E)*
T	0.9
F	0.2

A *B* *C* *D* *E*

$$= \alpha \times 0.1 \times (0.1 \times 0.8 \times 0.4 \times 0.9 + \qquad b, c$$

$$0.1 \times 0.2 \times 0.3 \times 0.2 + \qquad b, \neg c$$

$$0.9 \times 0.5 \times 0.4 \times 0.9 + \qquad \neg b, c$$

$$0.9 \times 0.5 \times 0.3 \times 0.2) \qquad \neg b, \neg c$$

Inference by Enumeration

- now do equivalent calculation for P($\neg a$ | d, e)
- and normalize

$$\alpha \sum_b \sum_c P(a)P(b)P(c \mid a,b)P(d \mid c)P(e \mid c) +$$

$$\alpha \sum_b \sum_c P(\neg a)P(b)P(c \mid \neg a,b)P(d \mid c)P(e \mid c) = 1$$

Exact Inference

- *inference by enumeration* is an *exact* method (i.e. it computes the exact answer to a given query)
- it requires summing over a joint distribution whose size is exponential in the number of variables
- can we do exact inference efficiently in large networks?
  - in many cases, yes
  - key insight: save computation by pushing sums inward

Potentials

- we can represent a CPT or a full joint distribution in this format

D	*E*	*P(F)*
T	T	0.8
T	F	0.5
F	T	0.2
F	F	0.7

$\Rightarrow$

d	*e*	*f*	.8
d	*e*	¬ *f*	.2
d	¬ *e*	*f*	.5
d	¬ *e*	¬ *f*	.5
¬ *d*	*e*	*f*	.2
¬ *d*	*e*	¬ *f*	.8
¬ *d*	¬ *e*	*f*	.7
¬ *d*	¬ *e*	¬ *f*	.3

- when we (temporarily) ignore meaning of table, we call it a *potential*: a table of non-negative values with no further constraints

Multiplying Potentials

- domain of (variables in) result is the union of domains of input potentials
- for each cell in result, multiply all input cells that agree on variable settings

a	*b*	.1
a	¬ *b*	.2
¬ *a*	*b*	.5
¬ *a*	¬ *b*	.8

×

b	*c*	.2
b	¬ *c*	.4
¬ *b*	*c*	.3
¬ *b*	¬ *c*	.5

=

a	*b*	*c*	.02
a	*b*	¬ *c*	.04
a	¬ *b*	*c*	.06
a	¬ *b*	¬ *c*	.10
¬ *a*	*b*	*c*	.10
¬ *a*	*b*	¬ *c*	.20
¬ *a*	¬ *b*	*c*	.24
¬ *a*	¬ *b*	¬ *c*	.40

a	*b*	.1
a	¬ *b*	.5
¬ *a*	*b*	.4
¬ *a*	¬ *b*	.1

×

a	*b*	.8
a	¬ *b*	.7
¬ *a*	*b*	.9
¬ *a*	¬ *b*	.8

=

a	*b*	.08
a	¬ *b*	.35
¬ *a*	*b*	.36
¬ *a*	¬ *b*	.08

Marginalizing and Normalizing Potentials

- can also marginalize (sum out a variable) potentials

a	*b*	.1
a	¬ *b*	.5
¬ *a*	*b*	.2
¬ *a*	¬ *b*	.7

$\Rightarrow$ $\sum_B$

a	.6
¬ *a*	.9

- and normalize them

a	*b*	.1
a	¬ *b*	.5
¬ *a*	*b*	.2
¬ *a*	¬ *b*	.7

$\Rightarrow$

a	*b*	.067
a	¬ *b*	.333
¬ *a*	*b*	.133
¬ *a*	¬ *b*	.467

Key Observation

$$\sum_A (P_1 \times P_2) = \left(\sum_A P_1 \right) \times P_2 \quad \text{if } A \text{ is not in } P_2$$

a	*b*	.1
a	¬ *b*	.2
¬ *a*	*b*	.2
¬ *a*	¬ *b*	.3

×

b	*c*	.1
b	¬ *c*	.4
¬ *b*	*c*	.3
¬ *b*	¬ *c*	.1

=

b	*c*	*a*	.01
b	*c*	¬ *a*	.02
b	¬ *c*	*a*	.04
b	¬ *c*	¬ *a*	.08
¬ *b*	*c*	*a*	.06
¬ *b*	*c*	¬ *a*	.09
¬ *b*	¬ *c*	*a*	.02
¬ *b*	¬ *c*	¬ *a*	.03

$\Rightarrow \sum_A$

b	*c*	.03
b	¬ *c*	.12
¬ *b*	*c*	.15
¬ *b*	¬ *c*	.05

- can marginalize (sum out *A*) before multiplying in this case, resulting in a smaller intermediate table

a	*b*	.1
a	¬ *b*	.2
¬ *a*	*b*	.2
¬ *a*	¬ *b*	.3

$\Rightarrow \sum_A$

b	.3
¬ *b*	.5

×

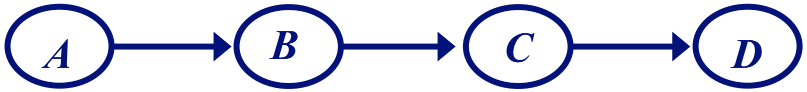
b	*c*	.1
b	¬ *c*	.4
¬ *b*	*c*	.3
¬ *b*	¬ *c*	.1

=

b	*c*	.03
b	¬ *c*	.12
¬ *b*	*c*	.15
¬ *b*	¬ *c*	.05

Variable Elimination

- consider answering the query $P(D)$ with following network



- this entails computing

$$P(D) = \sum_a \sum_b \sum_c P(a)P(b \mid a)P(c \mid b)P(D \mid c)$$

- but we can do this more efficiently by pushing the sums in

$$P(D) = \sum_c P(D \mid c) \sum_b P(c \mid b) \sum_a P(a)P(b \mid a)$$

Variable Elimination Procedure

given: Bayes net, evidence for variables in $\boldsymbol{V}$, query variable Q

initial potentials are the CPTs

repeat until only query variable remains

    choose a variable X to eliminate

        multiply all potentials that contain X

        if X is not in $\boldsymbol{V}$

            sum X out; replace potential by new result

        else

            remove X based on evidence

multiply remaining potentials for Q

normalize remaining potential to get final distribution over Q

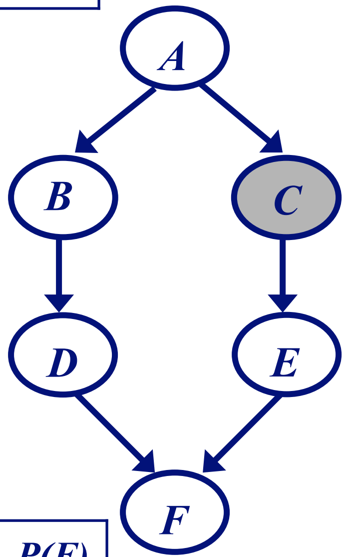
Variable Elimination Example

- suppose the query is $P(F \mid c)$
- let's eliminate variables in order *A, B, C, D, E*

P(A)
0.2

A	P(B)
T	0.8
F	0.3

A	P(C)
T	0.1
F	0.6



B	P(D)
T	0.4
F	0.7

C	P(E)
T	0.5
F	0.9

D	E	P(F)
T	T	0.8
T	F	0.5
F	T	0.2
F	F	0.7

Variable Elimination: Eliminating *A*

1. multiply all potentials that involve ***A***

a	.2
¬ *a*	.8

×

a	*b*	.8
a	¬ *b*	.2
¬ *a*	*b*	.3
¬ *a*	¬ *b*	.7

×

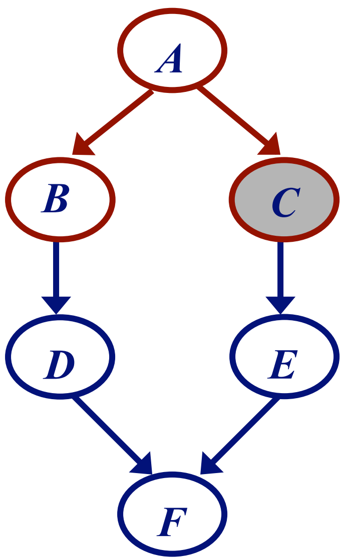
a	*c*	.1
a	¬ *c*	.9
¬ *a*	*c*	.6
¬ *a*	¬ *c*	.4

=

b	*c*	*a*	.016
b	*c*	¬ *a*	.144
b	¬ *c*	*a*	.144
b	¬ *c*	¬ *a*	.096
¬ *b*	*c*	*a*	.004
¬ *b*	*c*	¬ *a*	.336
¬ *b*	¬ *c*	*a*	.036
¬ *b*	¬ *c*	¬ *a*	.224

2. eliminate ***A*** by summing out

b	*c*	.16
b	¬ *c*	.24
¬ *b*	*c*	.34
¬ *b*	¬ *c*	.26



$$P(F \mid c)$$

Variable Elimination: Eliminating *B*

1. multiply potentials involving ***B***

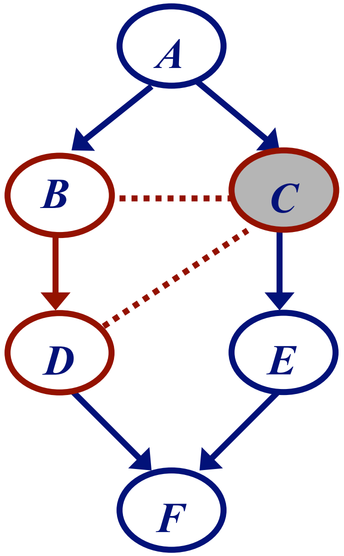
b	*d*	.4
b	¬ *d*	.6
¬ *b*	*d*	.7
¬ *b*	¬ *d*	.3

×

b	*c*	.16
b	¬ *c*	.24
¬ *b*	*c*	.34
¬ *b*	¬ *c*	.26

=

c	*d*	*b*	.064
c	*d*	¬ *b*	.238
c	¬ *d*	*b*	.096
c	¬ *d*	¬ *b*	.102
¬ *c*	*d*	*b*	.096
¬ *c*	*d*	¬ *b*	.182
¬ *c*	¬ *d*	*b*	.144
¬ *c*	¬ *d*	¬ *b*	.078



$$P(F \mid c)$$

2. eliminate ***B*** by summing out

c	*d*	.302
c	¬ *d*	.198
¬ *c*	*d*	.278
¬ *c*	¬ *d*	.222

Variable Elimination: Eliminating *C*

1. multiply potentials involving ***C***

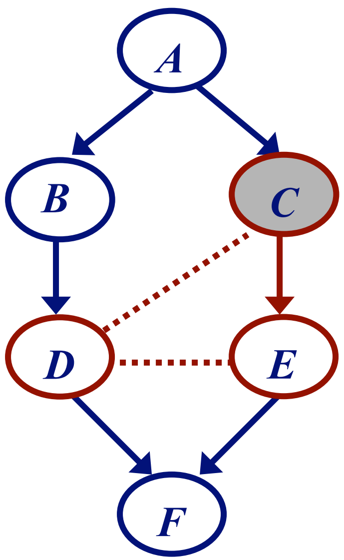
c	d	.302
c	$\neg d$	.198
$\neg c$	d	.278
$\neg c$	$\neg d$	.222

×

c	e	.5
c	$\neg e$	.5
$\neg c$	e	.9
$\neg c$	$\neg e$	.1

=

c	d	e	.151
c	d	$\neg e$	.151
c	$\neg d$	e	.099
c	$\neg d$	$\neg e$	.099
$\neg c$	d	e	.250
$\neg c$	d	$\neg e$	.028
$\neg c$	$\neg d$	e	.200
$\neg c$	$\neg d$	$\neg e$	.022



$$P(F \mid c)$$

2. we have evidence for ***C***, so eliminate values corresponding to $\neg c$

Variable Elimination: Eliminating *D*

1. multiply potentials involving ***D***

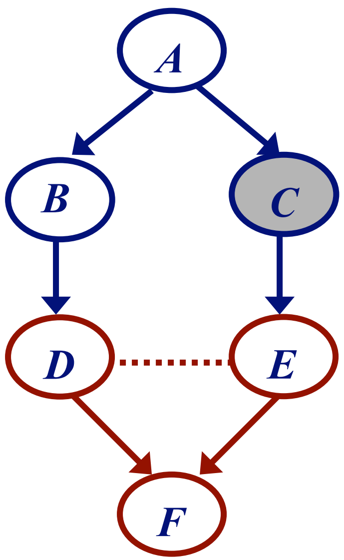
d	*e*	.151
d	¬ *e*	.151
¬ *d*	*e*	.099
¬ *d*	¬ *e*	.099

×

e	*f*	*d*	.1
e	*f*	¬ *d*	.4
e	¬ *f*	*d*	.9
e	¬ *f*	¬ *d*	.6
¬ *e*	*f*	*d*	.3
¬ *e*	*f*	¬ *d*	.9
¬ *e*	¬ *f*	*d*	.7
¬ *e*	¬ *f*	¬ *d*	.1

=

e	*f*	*d*	.015
e	*f*	¬ *d*	.040
e	¬ *f*	*d*	.136
e	¬ *f*	¬ *d*	.059
¬ *e*	*f*	*d*	.045
¬ *e*	*f*	¬ *d*	.089
¬ *e*	¬ *f*	*d*	.106
¬ *e*	¬ *f*	¬ *d*	.010



$P(F \mid c)$

2. sum out ***D***

e	*f*	.055
e	¬ *f*	.195
¬ *e*	*f*	.134
¬ *e*	¬ *f*	.116

Variable Elimination: Eliminating *E*

1. eliminate ***E***

e	***f***	**.055**
e	**¬ *f***	**.195**
¬ *e*	***f***	**.134**
¬ *e*	**¬ *f***	**.116**

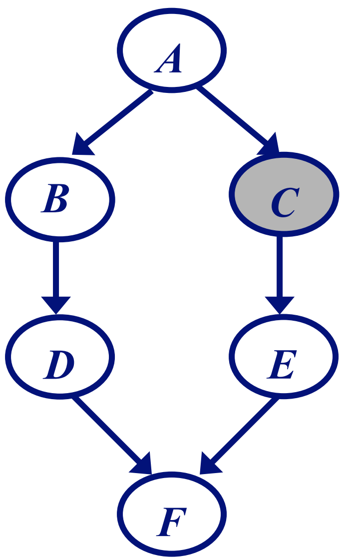
⇒

f	**.189**
¬ *f*	**.311**

⇓

2. normalize to get final answer

f	**.378**
¬ *f*	**.622**



$P(F \mid c)$

Variable Elimination

Inference by Enumeration

- multiply all CPTs in Bayes net (any order)
- eliminate some cells based on evidence
- marginalize down to query variable (sum out others in any order)

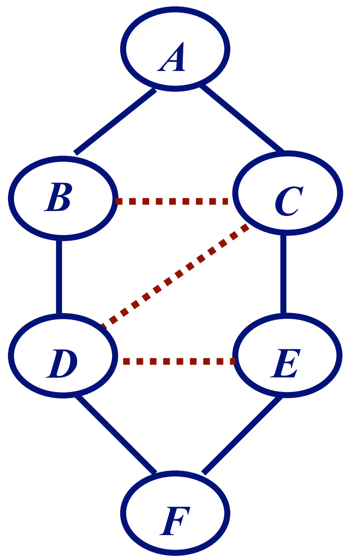
Variable Elimination

- multiply CPTs, but sum out a variable once we've included all tables that use it
- i.e. push the sums in as far as we can

Variable Elimination as a Graph Transformation

- initial graph is undirected version of Bayes net
- when we eliminate a variable *X*, we create a potential that contains *X* and the *set* of variables ***Y*** it appears with in potentials
- eliminating *X* transforms graph by adding edges between all variables in ***Y***

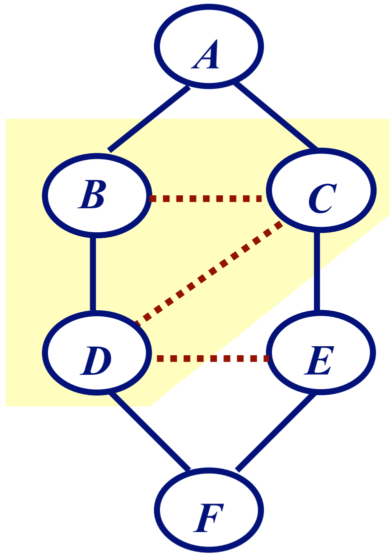
for the ordering *A, B, C, D, E* we get the dashed edges added



Variable Elimination as a Graph Transformation

- the cliques of this graph correspond to the intermediate result potentials we calculated

c	*d*	*b*	.064
c	*d*	¬ *b*	.238
c	¬ *d*	*b*	.096
c	¬ *d*	¬ *b*	.102
¬ *c*	*d*	*b*	.096
¬ *c*	*d*	¬ *b*	.182
¬ *c*	¬ *d*	*b*	.144
¬ *c*	¬ *d*	¬ *b*	.078



- the size of the potentials correspond to the size of the cliques

Choosing the Elimination Ordering

- keep in mind this *induced graph* depends on the ordering
- the time complexity is dominated by the largest potential we have to compute (largest clique in induced graph)
- therefore, we want to find an ordering that leads to an induced graph with with a small largest clique
- finding the optimal ordering is NP-hard, but there are some reasonable heuristics

Greedy Elimination Ordering

given: a Bayes net represented as an undirected graph G,
an evaluation function f

initialize all nodes in G as unmarked

for $k = 1$ to number of nodes

select unmarked variable X that minimizes $f(G,X)$

$\pi(X) = k$

add edges in G between all unmarked neighbors of X

mark X

return: variable ordering π

one reasonable choice for f is **min-fill**: the number of edges that need to be added to the graph due to its elimination

Comments on BN Inference

- in general, the Bayes net inference problem is NP-hard (or harder)
- exact inference is tractable for many real-world graphical models, however
- there are many methods for approximate inference – get an answer which is "close"
- in general, the approximate inference problem is NP-hard as well
- approximate inference works well for many real-world graphical models, however