Inference in Bayesian Networks

BMI/CS 576
www.biostat.wisc.edu/bmi576/
Mark Craven
craven@biostat.wisc.edu
Fall 2011

The inference task in Bayesian networks

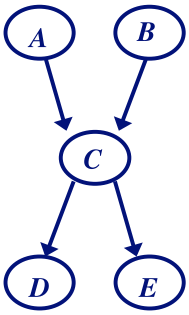
Given: values for some variables in the network (*evidence*), and a set of *query* variables

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 $\neg a$ 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 (b, $\neg b$, c, $\neg c$)

Inference by enumeration

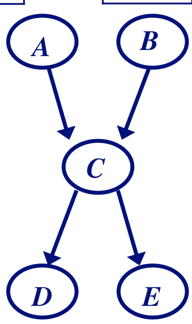
$$P(a \mid d,e) = \alpha \sum_b \sum_c P(a)P(b)P(c \mid a,b)P(d \mid c)P(e \mid c)$$

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

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

$$\begin{array}{lr} \quad\;\; A \quad\;\; B \quad\;\; C \quad\;\; D \quad\;\; E & \\ = \alpha \times 0.1 \times (0.1 \times 0.8 \times 0.4 \times 0.9 + & b, c \\ 0.1 \times 0.2 \times 0.3 \times 0.2 + & b, \neg c \\ 0.9 \times 0.5 \times 0.4 \times 0.9 + & \neg b, c \\ 0.9 \times 0.5 \times 0.3 \times 0.2) & \neg b, \neg c \end{array}$$

Inference by enumeration

- now do equivalent calculation for $P(\neg a \mid d, e)$
- 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 (a.k.a factors)

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

D	*E*	*P(G)*
t	t	0.8
t	f	0.5
f	t	0.2
f	f	0.7

$\Rightarrow$

d	e	g	.8
d	e	$\neg g$	.2
d	$\neg e$	g	.5
d	$\neg e$	$\neg g$	.5
$\neg d$	e	g	.2
$\neg d$	e	$\neg g$	.8
$\neg d$	$\neg e$	g	.7
$\neg d$	$\neg e$	$\neg g$	.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	$\neg b$	.2
$\neg a$	b	.5
$\neg a$	$\neg b$	.8

$\times$

b	c	.2
b	$\neg c$	.4
$\neg b$	c	.3
$\neg b$	$\neg c$	.5

=

a	b	c	.02
a	b	$\neg c$	.04
a	$\neg b$	c	.06
a	$\neg b$	$\neg c$	.10
$\neg a$	b	c	.10
$\neg a$	b	$\neg c$	.20
$\neg a$	$\neg b$	c	.24
$\neg a$	$\neg b$	$\neg c$	.40

a	b	.1
a	$\neg b$	.5
$\neg a$	b	.4
$\neg a$	$\neg b$	.1

$\times$

a	b	.8
a	$\neg b$	.7
$\neg a$	b	.9
$\neg a$	$\neg b$	.8

=

a	b	.08
a	$\neg b$	.35
$\neg a$	b	.36
$\neg a$	$\neg b$	.08

Marginalizing and normalizing potentials

- can also marginalize (sum out a variable) potentials

a	b	.1
a	$\neg b$	.5
$\neg a$	b	.2
$\neg a$	$\neg b$	.7

$\Rightarrow \sum_B$

a	.6
$\neg a$	.9

- and normalize them

a	b	.1
a	$\neg b$	.5
$\neg a$	b	.2
$\neg a$	$\neg b$	.7

$\Rightarrow$

a	b	.067
a	$\neg b$	.333
$\neg a$	b	.133
$\neg a$	$\neg b$	.467

Key observation

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

a	b	.1
a	$\neg b$	.2
$\neg a$	b	.2
$\neg a$	$\neg b$	.3

$\times$

b	c	.1
b	$\neg c$	.4
$\neg b$	c	.3
$\neg b$	$\neg c$	.1

$=$

b	c	a	.01
b	c	$\neg a$	.02
b	$\neg c$	a	.04
b	$\neg c$	$\neg a$	.08
$\neg b$	c	a	.06
$\neg b$	c	$\neg a$	.09
$\neg b$	$\neg c$	a	.02
$\neg b$	$\neg c$	$\neg a$	.03

$\Rightarrow \sum_A$

b	c	.03
b	$\neg c$	.12
$\neg b$	c	.15
$\neg b$	$\neg c$	.05

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

a	b	.1
a	$\neg b$	.2
$\neg a$	b	.2
$\neg a$	$\neg b$	.3

$\Rightarrow \sum_A$

b	.3
$\neg b$	.5

$\times$

b	c	.1
b	$\neg c$	.4
$\neg b$	c	.3
$\neg b$	$\neg c$	.1

$=$

b	c	.03
b	$\neg c$	.12
$\neg b$	c	.15
$\neg b$	$\neg c$	.05

Variable elimination

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

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

- 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

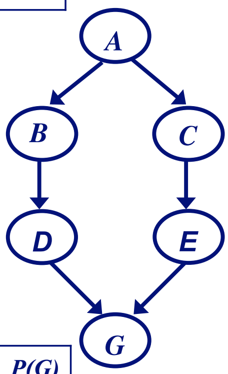
Variable elimination example

- suppose the query is $P(G \mid c)$
- 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	$P(G)$
t	t	0.8
t	f	0.5
t	t	0.2
t	f	0.7

Variable elimination: eliminating A

1. multiply all potentials that involve A

a	.2
$\neg a$	.8

×

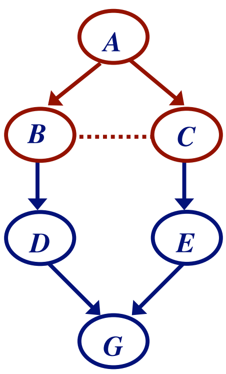
a	b	.8
a	$\neg b$	.2
$\neg a$	b	.3
$\neg a$	$\neg b$	.7

×

a	c	.1
a	$\neg c$	.9
$\neg a$	c	.6
$\neg a$	$\neg c$	.4

=

b	c	a	.016
b	c	$\neg a$	.144
b	$\neg c$	a	.144
b	$\neg c$	$\neg a$	.096
$\neg b$	c	a	.004
$\neg b$	c	$\neg a$	.336
$\neg b$	$\neg c$	a	.036
$\neg b$	$\neg c$	$\neg a$	.224



$P(G \mid c)$

2. eliminate A by summing out

b	c	.16
b	$\neg c$	.24
$\neg b$	c	.34
$\neg b$	$\neg c$	.26

Variable elimination: eliminating B

1. multiply potentials involving B

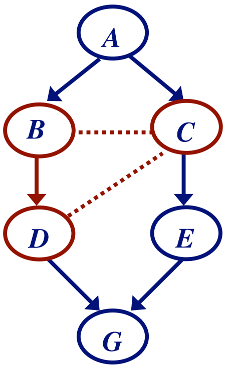
b	d	.4
b	$\neg d$	.6
$\neg b$	d	.7
$\neg b$	$\neg d$	.3

×

b	c	.16
b	$\neg c$	.24
$\neg b$	c	.34
$\neg b$	$\neg c$	.26

=

c	d	b	.064
c	d	$\neg b$	.238
c	$\neg d$	b	.096
c	$\neg d$	$\neg b$	.102
$\neg c$	d	b	.096
$\neg c$	d	$\neg b$	.182
$\neg c$	$\neg d$	b	.144
$\neg c$	$\neg d$	$\neg b$	.078



$$P(G \mid c)$$

2. eliminate B by summing out

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

Variable elimination: eliminating C

1. multiply potentials involving C

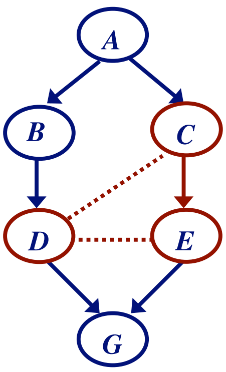
c	d	.302
c	$\neg d$	.198
$\neg c$	d	.278
$\neg c$	$\neg d$	.222

×

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



$$P(G \mid c)$$

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

Variable elimination: eliminating *D*

1. multiply potentials involving *D*

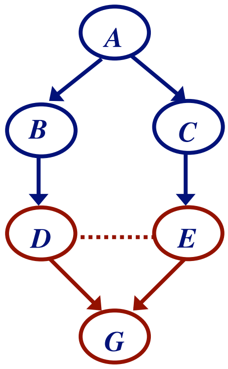
d	e	.151
d	$\neg e$	.151
$\neg d$	e	.099
$\neg d$	$\neg e$	.099

×

e	g	d	.1
e	g	$\neg d$	.4
e	$\neg g$	d	.9
e	$\neg g$	$\neg d$	.6
$\neg e$	g	d	.3
$\neg e$	g	$\neg d$	.9
$\neg e$	$\neg g$	d	.7
$\neg e$	$\neg g$	$\neg d$	.1

=

e	g	d	.015
e	g	$\neg d$	.040
e	$\neg g$	d	.136
e	$\neg g$	$\neg d$	.059
$\neg e$	g	d	.045
$\neg e$	g	$\neg d$	.089
$\neg e$	$\neg g$	d	.106
$\neg e$	$\neg g$	$\neg d$	.010



$P(G \mid c)$

2. sum out *D*

e	g	.055
e	$\neg g$	.195
$\neg e$	g	.134
$\neg e$	$\neg g$	.116

Variable elimination: eliminating *E*

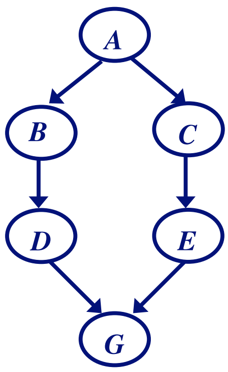
1. eliminate *E*

e	g	.055
e	$\neg g$	.195
$\neg e$	g	.134
$\neg e$	$\neg g$	.116

$\Rightarrow$

g	.189
$\neg g$	.311

$\Downarrow$



$P(G \mid c)$

2. normalize to get final answer

g	.378
$\neg g$	.622

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)

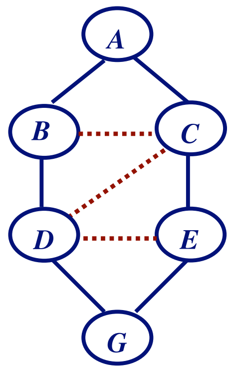
Variable elimination

- 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 $\boldsymbol{Y}$ that X appears with in potentials
- eliminating X transforms graph by adding edges between all variables in $\boldsymbol{Y}$

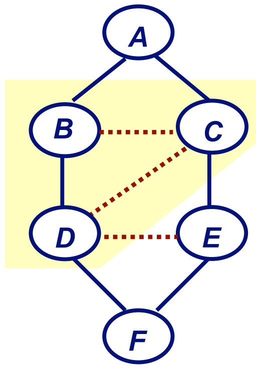
for the ordering A, B, C, D, E we get the dashed edges added



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 sizes 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 $f(G,X)$

$\pi(k) = X$

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 – these 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