Bayes' Nets

Dan Klein
CS121 Winter 2000-2001

1

What are they?

- A tool for reasoning probabilistically.
- A way of compactly representing joint probability functions.
- A graph over a set of random variables, along with a bunch of conditional probabilities.

2

Random Variables

- A *random variable* is a function from outcomes to real numbers. (But we'll be sloppy.)
- Think of a random variables as *indicators*.
- In our examples:
  - H will indicate whether or not I'm happy.
  - F will indicate whether or not I got a free lunch.

3

Joint Probability Distributions

- A *joint probability distribution* is an assignment of probabilities to outcomes, or to settings of the random variables.
  - Example : P(H=y, F=y) = 2/8
- Could encode this into a table:

	Food = N	Food = Y	Total
Happy = N	3/8	0	(3/8)
Happy = Y	3/8	2/8	(5/8)
Total	(6/8)	(2/8)	

4

Probabilistic Reasoning

- Queries like:
  - If there's free food, how likely is it that I'm happy? P(H=y| F=y)
  - If I'm happy, how likely is it that there's free food? P(F=y|H=y)
  - If there's food, what's the most likely state of my happiness?
  - Can already do some basic reasoning straight from our table (reasoning by enumeration).

5

Probabilistic Reasoning

	Food = N	Food = Y	Total
Happy = N	3/8	0	(3/8)
Happy = Y	3/8	2/8	(5/8)
Total	(6/8)	(2/8)	

P(F=n) $= \Sigma_{h \quad H}$ P(F=n, H=h) = 3/8+3/8 = 3/4

P(F=n | H=y) = P(F=n, H=y) / P(H=y)

= P(F=n, H=y) / $\Sigma_{f \quad F}$ P(F=f, H=y)

= 3/8 / (3/8 + 2/8)

= 3/5

6

Why Bayes' Nets

- Imagine a table over the following random variables:
  - H (whether I'm happy)
  - F (whether there's free food)
  - G (whether my car has enough gas)
  - W (whether the weather is nice)
  - And a bunch more…
- Building complete tables won't work for distributions – too big!
  - Getting enough data to fill all the entries is impractical.
  - Even storing the tables themselves is impractical!
- Bayes' nets can solve this problem by exploiting independencies.

7

Definition

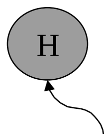
- A *Bayes' Net* is a directed, acyclic graph over a set of random variables.
- For each variable *X*, *parents*(*X*) are the variables which point to *X* in the graph.
- For each variable *X*, we have a *conditional probability table* (CPT) which specifies P(*X*|*parents*(X)).

8

First Bayes' Net

Bayes' Net over H

P(H)



Happy	
No	0.5
Yes	0.5

CPT for P(H)

Full table over H

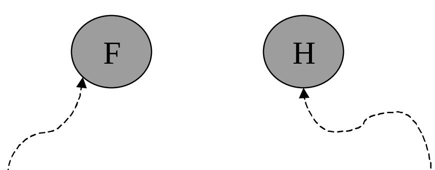
P(H)

Happy	
No	0.5
Yes	0.5

Second Bayes' Net

Bayes' Net over {F, H}

P(F,H) = P(F)P(H)



Food	
No	3/4
Yes	1/4

P(F)

Happy	
No	1/2
Yes	1/2

P(H)

Table over {F, H}

P(F,H)

H/F	No	Yes
No	3/8	1/8
Yes	3/8	1/8

The BN here says that H and F are independent! If so, we save space over the table.

Trivial Reasoning

We can immediately answer queries like:

P(F|H=y) = P(F) =

No	1/4
Yes	3/4

Since in this model, F and H are independent, reasoning is trivial.

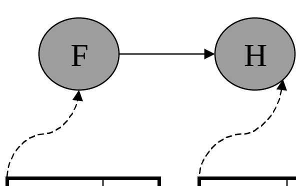
Note that by assuming independence, we lose the ability to represent some distributions (like the correct one where free food makes me happy!)

11

Third Bayes' Net

Bayes' Net over {F, H}

P(F,H) = P(F)P(H|F)



Food	
No	3/4
Yes	1/4

P(F)

Happy	F=No	F=Yes
No	1/2	0
Yes	1/2	1

P(H|F)

Table over {F, H}

P(F,H)

		Food	
		No	Yes
Happy	No	3/8	0
	Yes	3/8	2/8

Here, we didn't save any space, we just ended up saying writing the table another way.

12

Some Reasoning Examples

If we ask for P(H|F=y), life is easy, we have the answer sitting in our CPT:

No	0
Yes	1

If we ask for P(F|H=y), life is harder. We have to send the information in the "wrong" direction. We can simplify matters by asking for P(F,H=y) instead: (Why?)

P(F, H=y) = P(H=y|F)P(F) =

No	1/2
Yes	1

No	3/4
Yes	1/4

=

No	3/8
Yes	2/8

Pointwise product

13

Reasoning In General

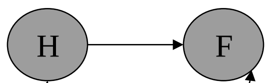
- In general, reasoning is of the form:
  - P(*Q*|*E)* where
    - *Q* is a set of *query* variables.
    - *E* is a set of *evidence* variables (with their values!)
    - Also, we have *Y*, the set of all remaining variables
  - What do we want then?
    - A |*Q*|-dimensional table giving the probability distribution P(*Q*|*E)*
- We can always break P(*Q*|*E)* into:
    - Finding P(*Q*,*E)* (which is a table)
    - Finding P(*E)* (which is just a number)
    - Dividing the table pointwise by P(*E*)

14

Fourth Bayes' Net

Bayes' Net over {F, H}

P(F,H) = P(H)P(F|H)



Happy	
No	3/8
Yes	5/8

P(H)

Food	H=No	H=Yes
No	1	3/5
Yes	0	2/5

P(F|H)

Table over {F, H}

P(F,H)

		Food	
		No	Yes
Happy	No	3/8	0
	Yes	3/8	2/8

How is this different from the last BN? Note that arrows mean dependence, not necessarily causation!

15

What the BN Means

- A Bayes' Net is an encoding of a joint distribution.
- The arrows encode which variables can depend directly on which other variables.
- The arrows are not necessarily *causal*, but it's often useful to think of them that way.

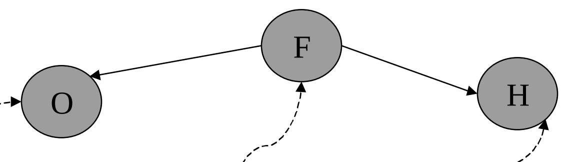
16

Fifth Bayes' Net

Bayes' Net over {O, H, F}

O = Other people happy.

P(O,H,F) = P(F)P(H|F)P(O|F)



Others	F=No	F=Yes
No	3/4	1/2
Yes	1/4	1/2

P(O|F)

Food	
No	3/4
Yes	1/4

P(F)

Happy	F=No	F=Yes
No	1/2	0
Yes	1/2	1

P(H|F)

17

Fifth Bayes' Net

- In this network,
  - O and H are not independent: If I'm happy, that's evidence that there's free food which influences the happiness of others.
  - However, once I know whether there's free food, my happiness is just a random process that has nothing to do with others' happiness.
- Notationally,
  - O and H are not (necessarily) independent, so we do not have I(O;H)
  - I(O;H|F) … O and H are *conditionally independent* given a value for F.

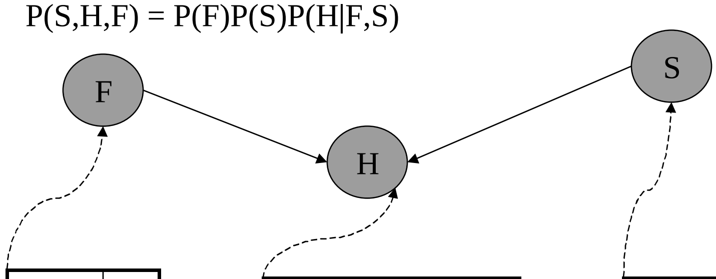
18

Sixth Bayes' Net

Bayes' Net over {S, H, F}

S = I got to sleep in.

P(S,H,F) = P(F)P(S)P(H|F,S)



Food	
No	3/4
Yes	1/4

P(F)

	F=No	F=Yes
S=No	1/2	1
S=Yes	3/4	1

P(H=1|F,S)

Sleep	
No	7/8
Yes	1/8

P(S)

19

Sixth Bayes' Net

- In this network,
  - If you know nothing about my happiness, then S and F are independent. They don't depend on each other in any way. We have I(S;F)
  - However, if you find out that I'm happy, it increases your belief that I got free food, and also increases your belief that I got to sleep in. Moreover, given F=y, knowing S *does* change the distribution over F (see next slide). Thus, we don't have I(S;F|H).
  - Something to ponder: In a given network, how can you tell what the independence relations are?

20

Explaining Away

- One interested kind of reasoning which Bayes' nets support is *intercausal* reasoning. For example, in the last network:
  - Before we know anything about H, P(F=y) = 1/4 and P(S=y) = 1/8.
  - If we know H=y, then P(F=y|H=y) goes up to .3855 and P(S=y|H=y) goes up to .1566
  - If we then discover that S=y, then P(F=y|H=y,S=y) drops to .3077
- This is often referred to as *explaining away* because once we find an explanation for an observation, the observation is partially explained and competing explanations become less likely.

21

Factorizations and BNs

- Given some set of random variables {A,B,C,D..} we can always write:

  P(A,B,C,D,..) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)…
- Or we can write:

  P(A,B,C,D,..) = P(D)P(B|D)P(C|D,B)P(A|D,B,C)…
- If we pick a good order to factorize, we can exploit independencies:

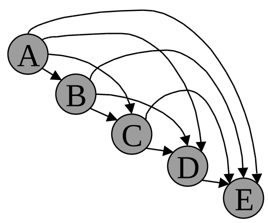
  P(S,F,H) = P(S) P(F|S) P(H|F,S) is just

  P(S,F,H) = P(S) P(F) P(H|F,S) since I(F;S)
- Different orders and independences give us different BNs.

22

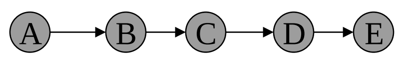
Factorizations and BNs

- Any distribution can be encoded in a BN like:



What factorization does this correspond to?

- There's only a savings over a complete table if there are useful independencies to exploit, such as:

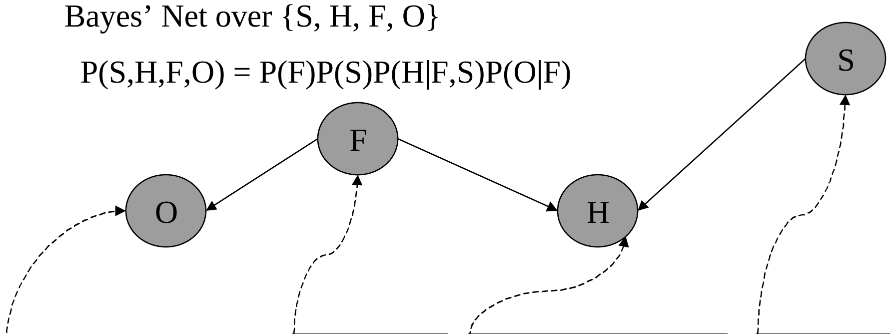


23

Seventh Bayes' Net

Bayes' Net over {S, H, F, O}

P(S,H,F,O) = P(F)P(S)P(H|F,S)P(O|F)



Others	F=No	F=Yes
No	3/4	1/2
Yes	1/4	1/2

P(O|F)

Food	
No	3/4
Yes	1/4

P(F)

	F=No	F=Yes
S=No	1/2	1
S=Yes	3/4	1

P(H=Yes|F,S)

Sleep	
No	7/8
Yes	1/8

P(S)

24

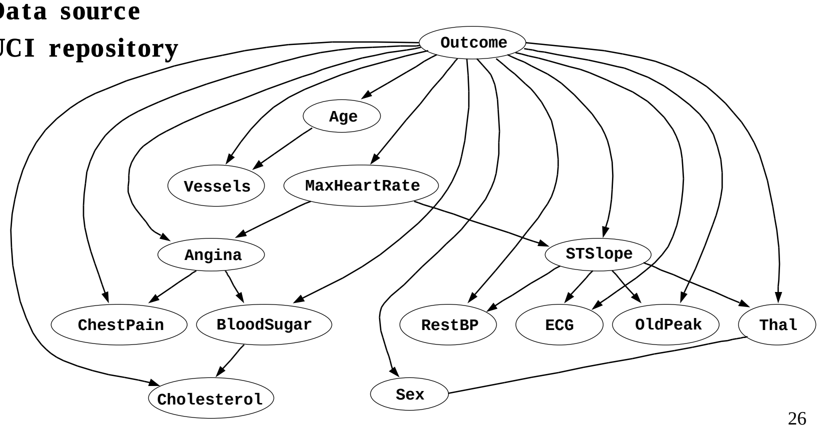
Applications

- Consumer modeling
  - Amazon customers / recommending purchases
  - TV viewers / Nielsen demographics
- User interfaces
  - NASA mission control
- Reasoning systems
  - Medical diagnosis
  - Microsoft help wizards

25

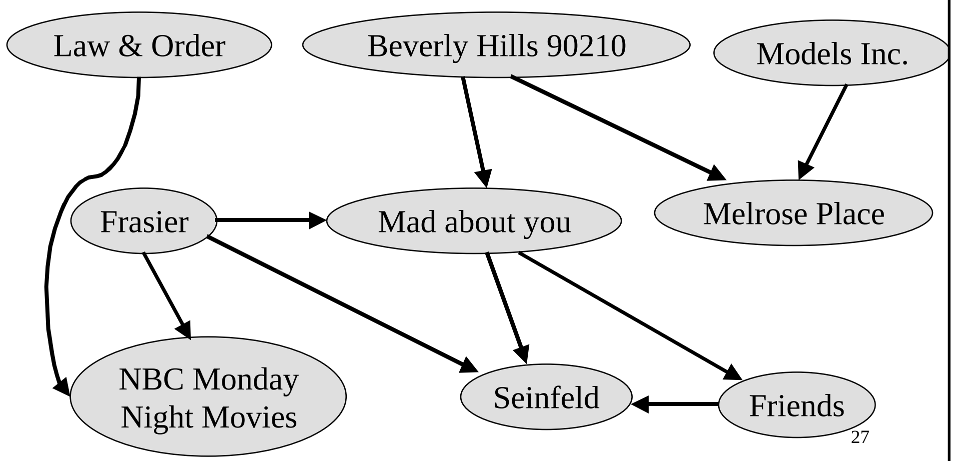
37th Bayes' Net

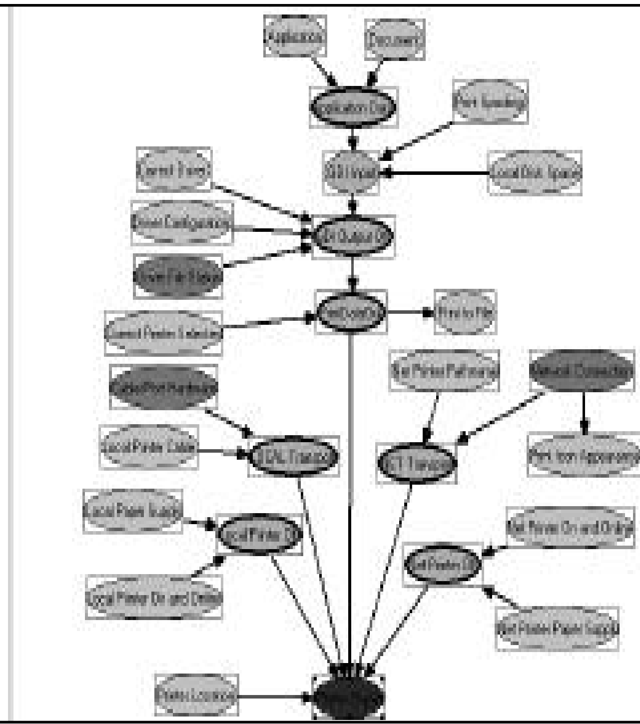
Heart disease
Accuracy = 85%
Data source
UCI repository



26

Nielsen data:
Portion of learned BN





What's Ahead?

- We want to be able to:
  - Answer queries for BNs.
    - Many algorithms for doing this, we'll discuss one called *variable elimination.*
  - Figure out what independence relations hold inside a BN.
    - The key notion is *d-separation* and lets us tell what nodes in a BN are necessarily independent of what other nodes given a certain evidence set.

29

Bayes' Nets II

CS121 Winter 2000-2001

30

Bayes' Nets II

- Naïve Bayes as a Bayes' Net
- Recap: How a BN encodes a joint distribution
- Two reasoning algorithms
  - Enumeration
  - Variable Elimination

31

NB for Text Categorization

- We have a random variable C for the category of a document, whose values are document categories.
- We have a random variable W for each word in the document, whose values are the words in our vocabulary.
- We *want* to know the most likely class given the words:
  $c = \text{argmax}_c \, P(C| W_1, W_2, .. W_n)$
- It's enough to find: *Why?*
  $c = \text{argmax}_c \, P(C, W_1, W_2, .. W_n)$
- *But* we can't store or estimate $P(C, W_1, W_2, .. W_n)$ directly.
- So we assume $I(W_i, W_j \mid C)$
- This lets us factorize $P(C, W_1, W_2, .. W_n)$ as $P(C)P(W_1|C)P(W_2|C)\ldots P(W_n| C)$
- We can estimate each $P(W_i|C)$ much more easily.

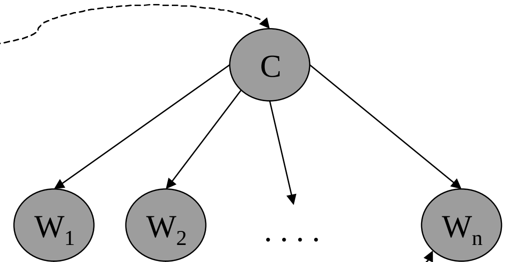
32

The BN for NB

$$P(C,W_1,W_2,\ldots W_n) = P(C)P(W_1|C)P(W_2|C)\ldots P(W_n|C)$$

Category	
Sports	1/4
Business	3/4

P(C)



Word x	C=Sports	C=Busn
goal	1/100	1/1000
stock	1/1000	1/50
…	…	…
the	1/10	1/10

$P(W_x|C)$

33

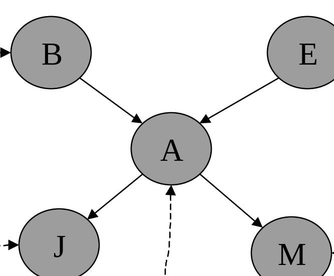
Burglary Network

Bayes' Net over {B, E, A, J, M}

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

Burglary	
No	.99
Yes	.01

P(B)



Earthquake	
No	.999
Yes	.001

P(E)

JohnCalls	A=No	A=Yes
No	.95	.10
Yes	.05	.90

P(J|A)

	B=No	B=Yes
E=No	.001	.94
E=Yes	.29	.95

P(A=Yes|B,E)

MaryCalls	A=No	A=Yes
No	.99	.10
Yes	.01	.90

P(M|A)

34

Finding an entry of the joint

- Let's say we want to know how likely the "perfect day" is:

  P(B=n,E=n,A=n,M=y,J=y) =

  $\prod_X$ P(X|*parents*(X)) =

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

  .99*.999*.999*.05*.01 ≈ .0049

35

Answering a Query by Enumeration

- Let's say we want to know how likely is it that there's a burglary given that both Mary and John call.
- This is the query P(B=y|M=y, J=y).
- We first find P(B=y, M=y, J=y)
- This is the sum of all the matching entries in the joint…

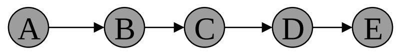
  P(B=y, M=y, J=y) = $\Sigma_{e,a}$ P(B=y, e, a, M=y, J=y)

  … so we have to sum 4 terms from the joint, one for each setting of the variables not in our query. If there are *n* binary variables, this means 2^n time just to sum them! (But it works.)
- How many entries to sum to find P(M=y, J=y)?

36

Answering queries faster!

- Take a simple chain BN:

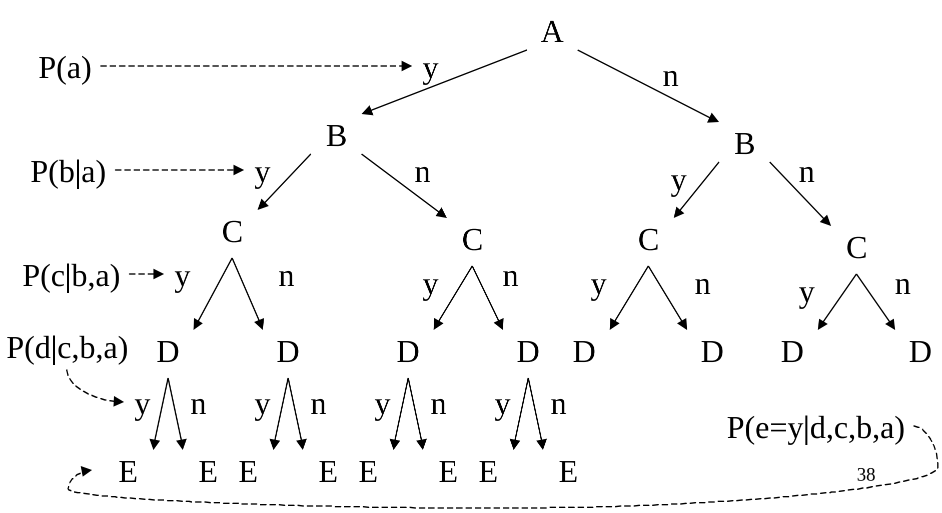


P(A,B,C,D,E) = P(A)P(B|A)P(C|B)P(D|C)P(E|D)

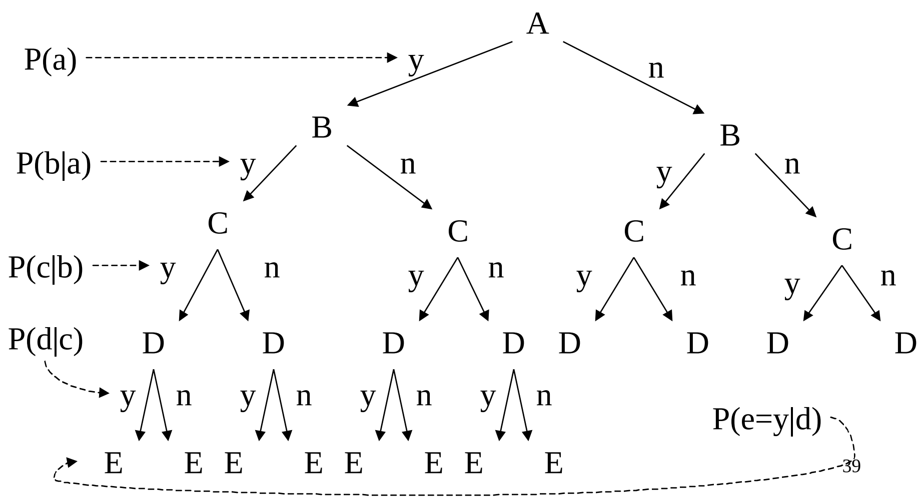
- Consider the query P(A|E=y). As usual, we'll break this up into P(A, E=y) and P(E=y).
  - Question: is there a quick way to figure out P(E=y) once we know P(A, E=y)?
- We can write this as:

$$
\begin{aligned}
P(A=a,E=y) &= \Sigma_{b,c,d}\, P(a,b,c,d,E=y) \\
&= \Sigma_{b,c,d}\, P(a)\, P(b|a)\, P(c|b)\, P(d|c)\, P(E=y|d) \\
&= P(a)\, \Sigma_{b,c,d}\, P(b|a)\, P(c|b)\, P(d|c)\, P(E=y|d) \\
&= P(a)\, \Sigma_b\, P(b|a)\, \Sigma_c\, P(c|b)\, \Sigma_d\, P(d|c)\, P(E=y|d)
\end{aligned}
$$

The resulting computation…

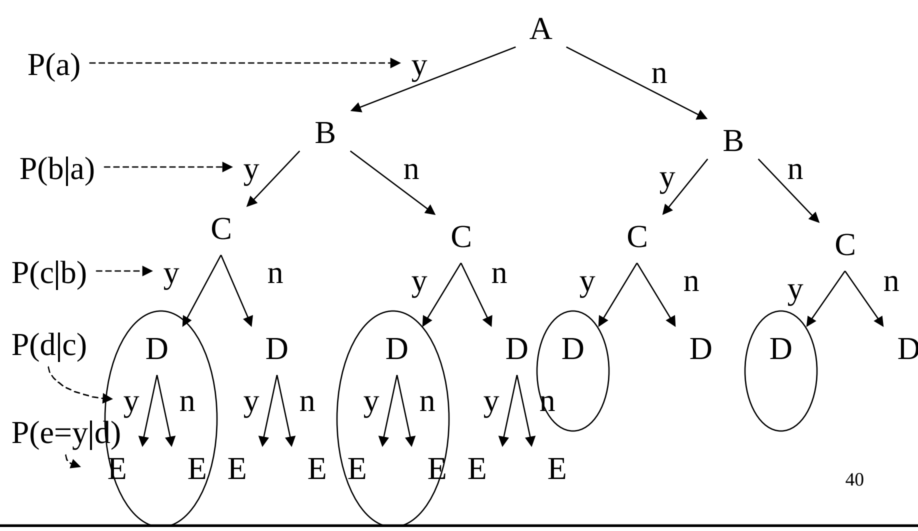


…with independences marked…



39

…and some redundant work circled



40

Variable Elimination

- The main idea of variable elimination is to *never do work twice*.
- We do the work *bottom-up*, or *inside-out,* rather than *top-down.*
- We store results that we will need again in tables called *factors*.
- We will create one factor per variable at the time we *eliminate* that variable.

41

The Variable Elimination Algorithm

start off with one factor for each CPT

while there is some (childless) variable

  pick any (childless) variable *X*

  take all factors $\{F_i\}$ which mention *X*

  create a new factor *G* by

    combining the F_i

      Create a table with a dimension for each variable in mentioned $\{F_i\}$, and fill in each entry by pointwise multiplication.

    if *X* is evidence, remove all entries in the table which don't match the observed value of *X*

    if *X* is a query variable, do nothing

    if *X* is an "other" variable, sum out over *X*

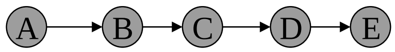
  remove the factors $\{F_i\}$, add the factor *G*

if the conditional is desired, normalize the final factor

42

Chain BN Example

- Let's go back to our chain network:



P(A,B,C,D,E) = P(A)P(B|A)P(C|B)P(D|C)P(E|D)

- We can write P(A,E=y) as

  $P(a)\ \Sigma_b\ P(b|a)\ \Sigma_c\ P(c|b)\ \Sigma_d\ P(d|c)\ P(e{=}y|d)$

P(e=y|d) determined by d

P(e=y|c) determined by c

P(e=y|b) determined by b

P(e=y|a) determined by a

P(e=y,a) the answer! determined by a

43

Chain BN Example

Let's say:

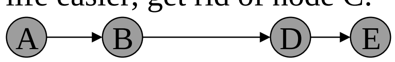
A is I set my alarm

B is I wake up in time

D is I get to work in time

E is I have to work late

- To make life easier, get rid of node C:



P(A,B,D,E) = P(A)P(B|A)P(D|B)P(E|D)

- Initial factors (one per node):

P(A) =

A	P(a)
y	3/4
n	1/4

P(B|A) =

B	P(b|A=y)	P(b|A=n)
y	1/2	1/3
n	1/2	2/3

P(D|B) =

D	P(d|B=y)	P(d|B=n)
y	2/3	1/4
n	1/3	3/4

P(E|D) =

E	P(e|D=y)	P(e|D=n)
y	1/4	1/2
n	3/4	1/2

44

Chain BN Example

Factors after:
P(A)
P(B|A)
P(D|B)
~~P(E|D)~~
P(E=y|D)

- Eliminating E:
  - Take all factors mentioning E and combine them pointwise:

P(E|D)

E	P(d|D=y)	P(d|D=n)
y	1/4	1/2
n	3/4	1/2

Just one, so no multiplications to do!

  - Since E is an evidence variable, select the portion of the result which fits the evidence:

P(E=y|D)

E	P(d|D=y)	P(d|D=n)
y	1/4	1/2

  - Remove the original factor from the factor list.

45

Chain BN Example

Factors after:
P(A)
P(B|A)
~~P(D|B)~~
~~P(E=y|D)~~
P(E=y|B)

- Eliminating D
  - Take all factors mentioning D and combine them pointwise:

P(E=y|D) P(D|B) → P(E=y,D|B)

E	P(e|D=y)	P(e|D=n)
y	1/4	1/2

D	P(d|B=y)	P(d|B=n)
y	2/3	1/4
n	1/3	3/4

E	D	P(e,d|B=y)	P(e,d|B=n)
y	y	_*2/3 = 1/6	_*_ = 1/16
y	n	_*1/3 = 1/6	_*3/4 = 3/8

  - Since D is neither evidence nor a query variable, we sum it out:

P(E=y,D|B) → P(E=y|B)

E	D	P(e,d|B=y)	P(e,d|B=n)
y	y	1/6	1/16
y	n	1/6	3/8

E	P(e|B=y)	P(e|B=n)
y	1/6+1/6 = 1/3	1/16+3/8 = 7/16

  - And we remove the input factors from the list.

46

Factors after:

P(A)

~~P(B|A)~~

~~P(E=y|B)~~

P(E=y|A)

Chain BN Example

- Eliminating B
  - Take all factors mentioning B and combine them pointwise:

P(E=y|B) P(B|A) → P(E=y,B|A)

E	P(e|B=y)	P(e|B=n)
y	1/3	7/16

B	P(b|A=y)	P(b|A=n)
y	1/2	1/3
n	1/2	2/3

E	B	P(e,b|A=y)	P(e,b|A=n)
y	y	1/3*1/2 = 1/6	1/3*1/3=1/9
y	n	7/16*1/2=7/32	7/16*2/3=7/24

- Since B is neither evidence nor a query variable, we sum it out:

P(E=y,B|A) → P(E=y|A)

E	B	P(e,b|A=y)	P(e,b|A=n)
y	y	1/6	1/9
y	n	7/32	7/24

E	P(e|A=y)	P(e|A=n)
y	1/6+7/32= 37/96	1/9+7/24= 29/72

- And we remove the input factors from the list.

47

Factors after:

~~P(A)~~

~~P(E=y|A)~~

P(E=y,A)

Chain BN Example

- Eliminating A
  - Take all factors mentioning A and combine them pointwise:

P(E=y|A) P(A) → P(E=y,A)

E	P(e|A=y)	P(e|A=n)
y	37/96	29/72

A	P(a)
y	3/4
n	1/4

E	A	P(e,a)
y	y	37/96*3/4 = .29
y	n	29/72*1/4 = .10

- Since A is a query variable, we do *not* sum it out.
- We delete the original factors.
- There are no variables left to eliminate, and we are left with a single factor which contains P(E=y,A).
- We can *normalize* it to add to one, giving us P(A|E=y). →

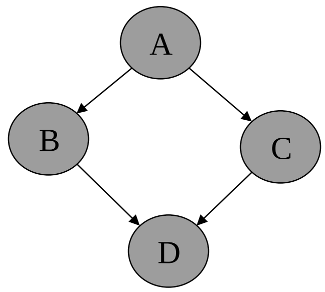
E	A	P(a|e)
y	y	.29/(.29+.1) = .73
y	n	.1/(.29+.1) = .27

I'm probably working late then, if I forget to set the alarm!

48

Next Time

- Next time, we'll do another example, with a loop:



- And discuss *d-separation*.

49

Bayes' Nets III

CS121 Winter 2000-2001

50

Bayes' Nets III

- Variable Elimination
- d-Separation
- Loose ends

51

Variable Elimination

- We want P(Q|e):
  - Start with CPTs
  - Process each variable
  - End up with a factor which represents P(Q,e)
  - Normalize to get P(Q|e)
- Remember: factors
  - Store results so we don't have to do an work twice.
  - Are *tables*, not just single numbers.

52

The Variable Elimination Algorithm

start off with one factor for each CPT

while there is some (childless) variable

- pick any (childless) variable *X*
- take all factors $\{F_i\}$ which mention *X*
- create a new factor *G* by
  - combining the F_i
    - Create a table with a dimension for each variable in mentioned $\{F_i\}$, and fill in each entry by pointwise multiplication.
  - if *X* is evidence, remove all entries in the table which don't match the observed value of *X*
  - if *X* is a query variable, do nothing
  - if *X* is an "other" variable, sum out over *X*
- remove the factors $\{F_i\}$, add the factor *G*

if the conditional is desired, normalize the final factor

53

Loop Example

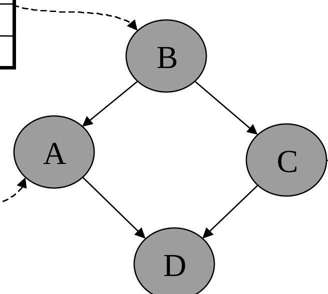
Let's say:

B is a burglary

A is my alarm goes off

C is a neighbor calls the police

D is the police show up



B	P(b)
y	.01
n	.99

A	P(a|B=y)	P(a|B=n)
y	.8	.01
n	.2	.99

C	P(c|B=y)	P(c|B=n)
y	.4	.1
n	.6	.9

D	P(d|A=y,C=y)	P(d|A=y,C=n)	P(d|A=n,C=y)	P(d|A=n,C=n)
y	.99	.7	.9	.01
n	.01	.3	.1	.99

54

Loop Example

- Query: P(B|D=y)
  - Note that we already know the prior P(B)
  - We expect P(B=y|D=y) >> P(B=y)
- Initial factors:
  - P(B)
  - P(A|B)
  - P(C|B)
  - P(D|A,C)

B	P(b)
y	.01
n	.99

55

Factors to start:
P(B)
P(A|B)
P(C|B)
P(D|A,C)

Loop Example

Factors after:
P(B)
P(A|B)
P(C|B)
~~P(D|A,C)~~
P(D=y|A,C)

- Elimination order: D,A,C,B (arbitrary)
- Eliminating D
  - Take all factors mentioning D (so just P(D|A,C)) and combine:

D	P(d|A=y,C=y)	P(d|A=y,C=n)	P(d|A=n,C=y)	P(d|A=n,C=n)
y	.99	.7	.9	.01
n	.01	.3	.1	.99

  - D is evidence so restrict the result to the entries consistent with the evidence:

D	P(d|A=y,C=y)	P(d|A=y,C=n)	P(d|A=n,C=y)	P(d|A=n,C=n)
y	.99	.7	.9	.01

P(D=y|A,C)

56

Factors before:
P(B)
P(A|B)
P(C|B)
P(D=y|A,C)

Loop Example

Factors after:
P(B)
~~P(A|B)~~
P(C|B)
~~P(D=y|A,C)~~
P(D=y|B,C)

- Eliminating A
  - Take all factors mentioning A and combine:

P(A|B)

A	P(a|B=y)	P(a|B=n)
y	.8	.01
n	.2	.99

P(D=y|A,C)

D	P(d|A=y,C=y)	P(d|A=y,C=n)	P(d|A=n,C=y)	P(d|A=n,C=n)
y	.99	.7	.9	.01

P(D=y,A|B,C)

D	A	P(d,a|B=y,C=y)	P(d,a|B=y,C=n)	P(d,a|B=n,C=y)	P(d,a|B=n,C=n)
y	y	.8*.99 = .792	.8*.7 = .56	.01*.99 = .0099	.01*.7 = .007
y	n	.2*.9 = .18	.2*.01 = .002	.99*.9 = .891	.99*.01 = .0099

Factors before:
P(B)
P(A|B)
P(C|B)
P(D=y|A,C)

Loop Example

Factors after:
P(B)
~~P(A|B)~~
P(C|B)
~~P(D=y|A,C)~~
P(D=y|B,C)

- Still eliminating A
  - Since A is a hidden (non-evidence, non-query) variable, sum it out:

P(D=y,A|B,C)

D	A	P(d,a|B=y,C=y)	P(d,a|B=y,C=n)	P(d,a|B=n,C=y)	P(d,a|B=n,C=n)
y	y	.8*.99 = .792	.8*.7 = .56	.01*.99 = .0099	.01*.7 = .007
y	n	.2*.9 = .18	.2*.01 = .002	.99*.9 = .891	.99*.01 = .0099

P(D=y|B,C)

D	P(d|B=y,C=y)	P(d|B=y,C=n)	P(d|B=n,C=y)	P(d|B=n,C=n)
y	.792+.18 = .972	.56+.002 = .562	.0099+.891 = .901	.007+.0099 = .0169

Factors before:
P(B)
P(C|B)
P(D=y|B,C)

Loop Example

Factors after:
P(B)
~~P(C|B)~~
~~P(D=y|B,C)~~
P(D=y|B)

- Eliminating C
  - Take all factors mentioning C and combine:

P(C|B)

C	P(c|B=y)	P(c|B=n)
y	.4	.1
n	.6	.9

P(D=y|B,C)

D	P(d|B=y,C=y)	P(d|B=y,C=n)	P(d|B=n,C=y)	P(d|B=n,C=n)
y	.972	.562	.901	.0169

P(D=y,C|B)

D	C	P(d,c|B=y)	P(d,c|B=n)
y	y	.4*.972 = .388	.1*.901 = .0901
y	n	.6*.562 = .3372	.9*.0169 = .01521

59

Factors before:
P(B)
P(C|B)
P(D=y|B,C)

Loop Example

Factors after:
P(B)
~~P(C|B)~~
~~P(D=y|B,C)~~
P(D=y|B)

- Still eliminating C
  - Since C is a hidden (non-evidence, non-query) variable, sum it out:

P(D=y,C|B)

D	C	P(d,c|B=y)	P(d,c|B=n)
y	y	.388	.0901
y	n	.3372	.01521

P(D=y|B)

D	P(d|B=y)	P(d|B=y)
y	.388+.3372 = .7252	.0901+.01521 = .10531

60

Factors before:
P(B)
P(D=y|B)

Factors after:
~~P(B)~~
~~P(D=y|B)~~
P(D=y,B)

Loop Example

- Eliminating B
  - Take all factors mentioning B and combine:

P(B)

B	P(b)
y	.01
n	.99

P(D=y|B)

D	P(d|B=y)	P(d|B=y)
y	.7252	.10531

P(D=y,B)

D	B	P(d,b)
y	y	.01*.7252 = .007252
y	n	.99*.10531 = .1052

61

Factors before:
P(B)
P(D=y|B)

Factors after:
~~P(B)~~
~~P(D=y|B)~~
P(D=y,B)

Loop Example

- Still eliminating B
  - Since B is a query variable, do nothing.
- Normalizing at the end:
  - Sum up the entries of P(D=y,B) to get P(D=y)

P(D=y,B)

D	B	P(d,b)
y	y	.007252
y	n	.1052

sum

P(D=y)

.007252+.1052 = .1125

normalize

P(B|D=y)

B	P(b|D=y)
y	.007252/.1125 = .064
n	.1052/.1125 = .936

The answer!

62

Loop Example

- Now we know P(B|D=y) and we've known P(B) all along:

P(B)

B	P(b)
y	.01
n	.99

P(B|D=y)

B	P(b|D=y)
y	.064
n	.936

- Knowing that the police came makes it six times as likely that there was a burglary, but since the influences are so weak (they probably wouldn't be for real) and since the prior chance of a burglary is low, even when the police show, it's still probably *not* because of a burglary!

63

Another Elimination Order

- Let's do the order {B,C,A,D}, no numbers this time.
- Eliminate B first:

Eliminating B

Initital factors: P(B) P(A|B) P(C|B) P(D|C,A)

Combine on B

P(B), P(A|B), P(C|B) combine to P(A,C,B)

B is query, do nothing

Factors after B: P(A,C,B) P(D|C,A)

64

Another Elimination Order

- Eliminate C next:

Eliminating C

Initital factors: P(A,C,B) P(D|C,A)

Combine on C

P(A,C,B) and P(D|C,A) combine to P(A,B,C,D)

C is hidden, sum out C

P(A,B,C,D) summed out over C gives P(A,B,D)

Factors after C: P(A,B,D)

65

Another Elimination Order

- Eliminate A next:

Eliminating A

Initital factors: P(A,B,D)

Combine on A

Nothing to do! Just P(A,B,D)

A is hidden, sum out A

P(A,B,D) summed out over A gives P(B,D)

Factors after A: P(B,D)

66

Another Elimination Order

- Eliminate D last:

Eliminating D

Initital factors: P(B,D)

Combine on D

Nothing to do! Just P(B,D)

D is evidence, restrict on D

P(B,D) restricted to D=y gives P(B,D=y)

Factors after D: P(B,D=y)

Another Elimination Order

- We end up with P(B,D=y) again.
- Normalize to get P(B|D=y).

- We *will* get the same answer!
- *But* we do more work this way:
  - We had to create the entire joint P(A,B,C,D) this time.
  - We didn't get to restrict to our evidence until the end this time.

- Different orders can result in different amounts of work, but not different results.

Influence

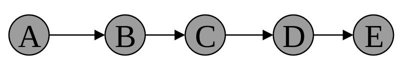
- Finding out that D=y changed our belief distribution over B (not a surprise!).
- *However*, nowhere in our BN does it say that A and D are dependent.
- We want to be able to tell when evidence will or will not influence other variables.

- In particular, we want to know whether variables X and Y are (necessarily) independent given a set E of evidence variables… in other words I(X;Y|E).

69

Paths of Influence

- In the chain network:



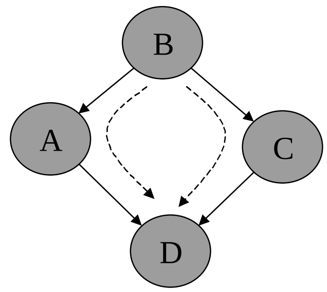
  – Every variable can influence every other variable via the intervening variables.
    - For example, A will not (generally) be independent of E.

  – If we observe a variable between A and E, then they become independent.
    - For example, I(A;E|D) if observing A has any effect on our beliefs about E, it is only because observing A changed our beliefs about D. There's no *other* dependence from A to E than the one mediated by D, so A and E are *separated* by the evidence D

70

Paths of Influence

- In general, evidence of a variable X can influence the distribution over a variable Y if there's an *active path* between them.
- In the loop network:
  - As we saw, evidence about D can influence B (in our example, P(B) was not the same as P(B|D=y))
  - This influence can flow either along B → A → D or B → C → D. So observing A doesn't by itself make B and D independent, but observing C *and* A does.



71

Paths of Influence

- There may be a lot of (undirected) paths *p* connecting two variables X and Y.
- We say that a path *p* between X and Y is an *active path* (given evidence E) if influence can flow along *p* (given E). If there are no active paths between X and Y (given E), then we will know that I(X;Y|E).
- Active paths are symmetric, i.e., influence can't flow only one way along a given path.
- As we have seen, whether or not a path is active *does* depend on the given evidence.

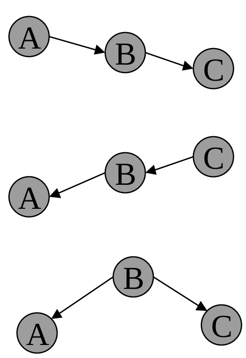
72

d-Separation

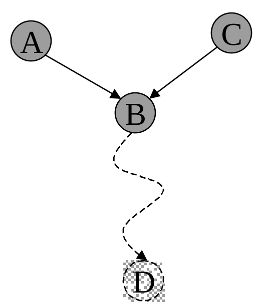
- If, given E, there is no active path between X and Y, we say that E *d-separates* X and Y, or d-sep(X;Y|E).
    - Think “data” or “direction-dependent” separation for d-separation!
- You can prove that if d-sep(X;Y|E) then I(X;Y|E).
    - But not the other direction!
- Now we have to be able to figure out if two nodes are d-separated by evidence.
    - We can check each path to see if it’s active.
    - To check a path, we only have to check each “link” in the path.

Types of Paths

- There are four kinds of “links” in a path:



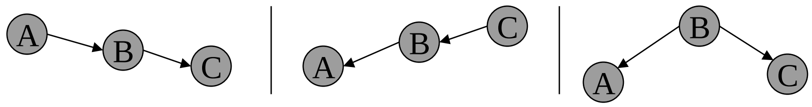
Here, influence flows from A to C through B. If B is unobserved, then this link is active. If B is observed, it breaks this link.



Here, influence cannot normally flow between A and C. They are two causes of a common effect B. However, once we observe B (or any descendant D of B), the causes “compete” and this path activates.

The Easy Three

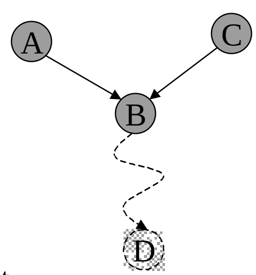
- Three of the link types are intuitively simple:



- In each case, A can influence C via B. If B is not evidence, then this link is active. If B *is* evidence, then A can no longer influence C via B and the link is no longer active.
  – Classic Example: (which case(s) does this correspond to above?)
    - Smoking causes cancer but only because of tar build-up in the lungs.
    - If you know that someone has tar in their lungs, it doesn't matter how it got there as far as lung cancer goes (maybe they work in a tar plant).
    - Whether or not they smoke is now irrelevant to whether or not they get cancer.

75

v-Structures

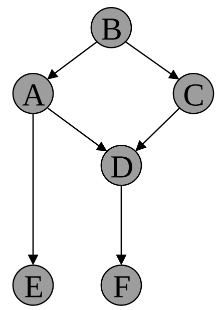


- The fourth case is less obvious:
  – A-B-C is called a v-structure.
  – v-structures work roughly backwards from the other three cases.
- Here, A and C are causes of the same effect.
  – If we do *not* know anything about B, then A and C won't influence each other… this link starts off inactive.
    - Burglaries and earthquakes both set off my alarm, but if I know nothing about the alarm, then burglaries an earthquakes are independent events.
  – If we *do* know something about B, then A and C can compete as explanations of our knowledge about B, and this link activates.
    - Once the alarm goes off, a burglary and an earthquake compete to explain the alarm.
    - B does not itself have to be observed, and descendent D of B will do.

76

d-Separation Example

- In this network:
  - Given no evidence
    - Is I(A;C)?
      - Is the A-B-C path active?
      - Is the A-D-C path active?
    - Is I(A;F)?
  - Given B
    - Is I(A;C|B)?
  - Given D and B
    - Is I(A;C|D,B)?
  - Some more to try:
    - What evidence sets separate B and F?
    - What evidence sets separate B and E?
    - What evidence sets separate C and E?



B is burglary
A is alarm
C is neighbors call police
D is police at my door
E is alarm company calls
F is a report gets filed

77

Loose Ends

- What do BNs get us?
  - A compact way of specifying a joint distribution.
  - We specify simple and *local* behavior of variables but then we get to reason about global interactions!
  - A uniform, well-founded, well-understood framework for combining arbitrary evidence and performing arbitrary queries.
- Bayes' Nets and Learning
- Dynamic Bayes' Nets, HMMs, etc.

78