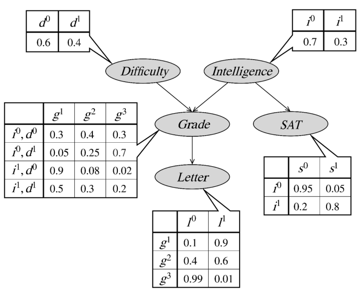
Inference on Bayes nets

CS B553
Spring 2013

Announcements

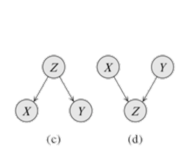
- Assignment 1 due Thursday

Last time: Bayes nets



Active trails

- A trail is between X and Y is *active* given a set of observed nodes **Z** if
  - In any "v-structure" (type (d)) below, the middle node or one of its descendants is in **Z**
  - No other node along the trail is in **Z**



d-separation

- Sets of nodes X and Y are *d-separated* with respect to nodes Z if there is no active trail between X and Y.
- **Stronger result: For almost all possible distributions, d-separation is equivalent to conditional independence.**

Solving problems with Bayes nets

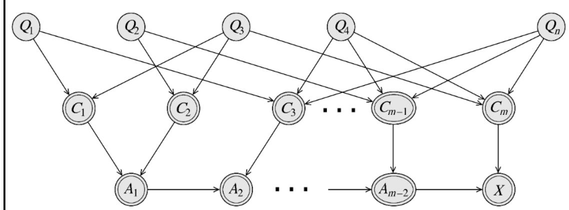
- We'd like to use Bayes nets to estimate (distributions over) variable values, given observed values for other variables
  - aka *Conditional Probability Queries:* Given a set of variables **E** and corresponding values **e**, estimate distributions over unobservable values **Y**, i.e. P(**Y** | **E**=**e**)

Sample application: Solving constraint problems

- Say someone gives us a Boolean expression like,
  $(Q_1 \text{ OR } Q_2 \text{ OR } Q_3) \text{ AND } (Q_4 \text{ OR } Q_5 \text{ OR } Q_6) \text{ AND } \ldots$
  - Where Q's may repeat and may include negation
  - We'd like to decide if there exists an assignment of boolean values to the Q's such that the expression is true
- This problem can be solved using a Bayes nets
  - Use conditional probabilities to "program" AND and OR operations

CSP with a Bayes net

- Q nodes represent variables in CSP
- C nodes have CPDs that implement "OR" operations
- A nodes have CPDs that implement "AND" operations
- CSP can be satisfied iff P(X) > 0

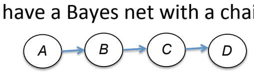


Bad news!

- We just showed a reduction from 3-SAT to Bayes net inference!
  - I.e., if we could solve Bayes net inference efficiently, then we can solve 3-SAT efficiently
  - But 3-SAT is NP-hard
  - So Bayes net inference is NP hard also
  - ☹
- Even approximate inference to within an absolute or relative error bound is NP-hard
  - ☹
  - Fortunately, for some Bayes nets inference is tractable

Marginal inference example

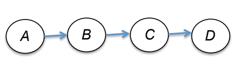
- Suppose we have a Bayes net with a chain topography:



  - How to compute P(B)? Running time to compute P(B)?
  - How to compute P(C)? Running time to compute P(C)?
  - For chain of length N, where each variables has k possible values, what is the running time to compute $P(X_N)$?
  - What if we assumed no independence assumptions between variables?

Alternative view

- What makes efficient inference possible?



  - Form of joint distribution, P(A,B,C,D)?
  - Supposing binary variables, we'd need to sum up 2^4 terms,

P(D=d^1)=

$$\begin{array}{lllll} & P(a^1) & P(b^1 \mid a^1) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\ + & P(a^2) & P(b^1 \mid a^2) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\ + & P(a^1) & P(b^2 \mid a^1) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\ + & P(a^2) & P(b^2 \mid a^2) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\ + & P(a^1) & P(b^1 \mid a^1) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\ + & P(a^2) & P(b^1 \mid a^2) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\ + & P(a^1) & P(b^2 \mid a^1) & P(c^2 \mid b^2) & P(d^1 \mid c^2) \\ + & P(a^2) & P(b^2 \mid a^2) & P(c^2 \mid b^2) & P(d^1 \mid c^2) \end{array}$$

P(D=d^2)=

$$\begin{array}{lllll} & P(a^1) & P(b^1 \mid a^1) & P(c^1 \mid b^1) & P(d^2 \mid c^1) \\ + & P(a^2) & P(b^1 \mid a^2) & P(c^1 \mid b^1) & P(d^2 \mid c^1) \\ + & P(a^1) & P(b^2 \mid a^1) & P(c^1 \mid b^2) & P(d^2 \mid c^1) \\ + & P(a^2) & P(b^2 \mid a^2) & P(c^1 \mid b^2) & P(d^2 \mid c^1) \\ + & P(a^1) & P(b^1 \mid a^1) & P(c^2 \mid b^1) & P(d^2 \mid c^2) \\ + & P(a^2) & P(b^1 \mid a^2) & P(c^2 \mid b^1) & P(d^2 \mid c^2) \\ + & P(a^1) & P(b^2 \mid a^1) & P(c^2 \mid b^2) & P(d^2 \mid c^2) \\ + & P(a^2) & P(b^2 \mid a^2) & P(c^2 \mid b^2) & P(d^2 \mid c^2) \end{array}$$

P(D=d^1)=

$$\begin{array}{lllll} & P(a^1) & P(b^1 \mid a^1) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\ + & P(a^2) & P(b^1 \mid a^2) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\ + & P(a^1) & P(b^2 \mid a^1) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\ + & P(a^2) & P(b^2 \mid a^2) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\ + & P(a^1) & P(b^1 \mid a^1) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\ + & P(a^2) & P(b^1 \mid a^2) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\ + & P(a^1) & P(b^2 \mid a^1) & P(c^2 \mid b^2) & P(d^1 \mid c^2) \\ + & P(a^2) & P(b^2 \mid a^2) & P(c^2 \mid b^2) & P(d^1 \mid c^2) \end{array}$$

Using distributive property of multiplication over addition,

P(D=d^1)= $$\begin{array}{llll} & (P(a^1)P(b^1 \mid a^1) + P(a^2)P(b^1 \mid a^2)) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\ + & (P(a^1)P(b^2 \mid a^1) + P(a^2)P(b^2 \mid a^2)) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\ + & (P(a^1)P(b^1 \mid a^1) + P(a^2)P(b^1 \mid a^2)) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\ + & (P(a^1)P(b^2 \mid a^1) + P(a^2)P(b^2 \mid a^2)) & P(c^2 \mid b^2) & P(d^1 \mid c^2) \end{array}$$

Which can be re-written as,

P(D=d^1)= $$\begin{array}{llll} & \tau_1(b^1) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\ + & \tau_1(b^2) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\ + & \tau_1(b^1) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\ + & \tau_1(b^2) & P(c^2 \mid b^2) & P(d^1 \mid c^2) \end{array}$$

where, $\tau_1(b^1) = P(a^1)P(b^1|a^1) + P(a^2)P(b^1|a^2)$

$\tau_1(b^2) = P(a^1)P(b^2|a^1) + P(a^2)P(b^2|a^2)$

$$P(D=d^1) = \tau_1(b^1)\,P(c^1 \mid b^1)\,P(d^1 \mid c^1) + \tau_1(b^2)\,P(c^1 \mid b^2)\,P(d^1 \mid c^1) + \tau_1(b^1)\,P(c^2 \mid b^1)\,P(d^1 \mid c^2) + \tau_1(b^2)\,P(c^2 \mid b^2)\,P(d^1 \mid c^2)$$

$$P(D=d^2) = \tau_1(b^1)\,P(c^1 \mid b^1)\,P(d^2 \mid c^1) + \tau_1(b^2)\,P(c^1 \mid b^2)\,P(d^2 \mid c^1) + \tau_1(b^1)\,P(c^2 \mid b^1)\,P(d^2 \mid c^2) + \tau_1(b^2)\,P(c^2 \mid b^2)\,P(d^2 \mid c^2)$$

$$P(D=d^1) = (\tau_1(b^1)P(c^1 \mid b^1) + \tau_1(b^2)P(c^1 \mid b^2))\,P(d^1 \mid c^1) + (\tau_1(b^1)P(c^2 \mid b^1) + \tau_1(b^2)P(c^2 \mid b^2))\,P(d^1 \mid c^2)$$

$$P(D=d^2) = (\tau_1(b^1)P(c^1 \mid b^1) + \tau_1(b^2)P(c^1 \mid b^2))\,P(d^2 \mid c^1) + (\tau_1(b^1)P(c^2 \mid b^1) + \tau_1(b^2)P(c^2 \mid b^2))\,P(d^2 \mid c^2)$$

$$P(D=d^1) = \tau_2(c^1)\,P(d^1 \mid c^1) + \tau_2(c^2)\,P(d^1 \mid c^2)$$

$$P(D=d^2) = \tau_2(c^1)\,P(d^2 \mid c^1) + \tau_2(c^2)\,P(d^2 \mid c^2)$$

where, $\tau_1(c^1) = \tau_1(b^1)P(c^1|b^1) + \tau_1(b^2)P(c^1|b^2)$

$\tau_1(c^2) = \tau_1(b^1)P(c^2|b^1) + \tau_1(b^2)P(c^2|b^2)$

Summary of what just happened…

- We want to compute P(D)

$$P(D) = \sum_C \sum_B \sum_A P(A,B,C,D)$$
$$P(D) = \sum_C \sum_B \sum_A P(A)P(B|A)P(C|B)P(D|C)$$
$$P(D) = \sum_C \sum_B P(C|B)P(D|C)\left(\sum_A P(A)P(B|A)\right)$$
$$P(D) = \sum_C \sum_B P(C|B)P(D|C)\tau_1(B)$$
$$P(D) = \sum_C P(D|C)\left(\sum_B P(C|B)\tau_1(B)\right)$$
$$P(D) = \sum_C P(D|C)\tau_2(C)$$

Dynamic programming

- This idea of caching intermediate results (in the form of tables τ_1 and τ_2) is called *dynamic programming*
  - General algorithmic concept
  - Other examples: Dijkstra's algorithm, string algorithms (e.g. for bioinformatics), Tower of Hanoi puzzle, …

How did we avoid exponential time?

$A \to B \to C \to D$

- Two important ingredients:
  - 1. The independence assumptions of the Bayes net allowed us to factor the joint distribution into simpler terms, each of which involved only a few variables.
  - 2. Dynamic programming let us "cache" intermediate results, avoiding re-computing them repeatedly.

More generally…

- More generally, notice that for any **sets** of random variables **A**, **B**, **C**, and **D**, and random variable $Z \notin \mathbf{U} \cup \mathbf{V}$,

$$\sum_Z P(\mathbf{U}|\mathbf{V})P(\mathbf{W}|\mathbf{X}) = P(\mathbf{U}|\mathbf{V})\sum_Z P(\mathbf{W}|\mathbf{X})$$

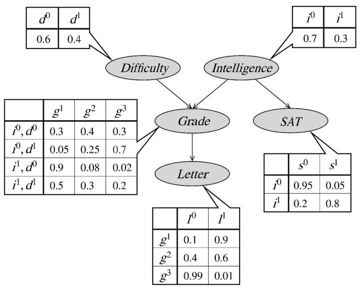
  - So, in the chain example above, this lets us do:

$$P(D) = \sum_C \sum_B \sum_A P(A)P(B|A)P(C|B)P(D|C)$$
$$= \sum_C \sum_B P(C|B)P(D|C)\left(\sum_A P(A)P(B|A)\right)$$
$$= \sum_C P(D|C)\left(\sum_B P(C|B)\left(\sum_A P(A)P(B|A)\right)\right)$$
$$= \sum_C P(D|C)\sum_B P(C|B)\sum_A P(A)P(B|A)$$

Variable elimination algorithm

1. Sort the non-query variables in an arbitrary order, Z1, Z2, … Zn
2. Initialize set of *factors* **F** to be the conditional probability distributions, P(Zi | Pa(Zi)).
3. For each i=1..n,
   a. Identify subset of factors **F'** involving Zi; these factors have some subset of variables **V** as parameters
   b. Take product of factors **F'**, parameterized by **V**
   c. Sum this product over all values of Zi, producing a new factor f parameterized by **V**-{Zi}
   d. Remove **F'** from **F**, then add f to **F**

Example



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Handling evidence

- Suppose we want to compute P(**Y** | **E**=**e**)
  - Set variables in **E** to their known values
  - Eliminate all remaining variables except for **Y,** resulting in P(**Y**, **E=e**)
  - Then marginalize over **Y** to compute P(**E**=**e**), in order to compute P(Y | E=e)