Bayes Nets 3

Ch. 14.4

Adapted from slides kindly shared by Stuart Russell

Appreciations

- ♢ My uncle Pat, passed away yesterday after years of Alzheimer's
- ♢ Family, gathered for Thanksgiving
- ♢ Language and evidence of the past

Share some of yours?

Announcements

Project P1 grades are up on D2L, with extra credit, early bonus, late add-ons, etc.

Project P3 Reinforcement due Thu Nov 29th at 17:00

Outline

◇ Bayes Nets, Exact Inference, Variable Elimination

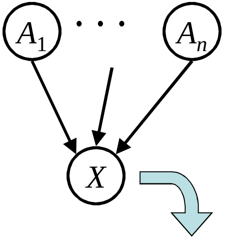
Credit to Dan Klein, Stuart Russell and Andrew Moore for most of today's slides

Bayes' Net Semantics

- A set of nodes, one per variable X
- A directed, acyclic graph
- A conditional distribution for each node
  - A collection of distributions over X, one for each combination of parents' values

$$P(X|a_1 \ldots a_n)$$

  - CPT: conditional probability table
  - Description of a noisy "causal" process



$$P(X|A_1 \ldots A_n)$$

A Bayes net = Topology (graph) + Local Conditional Probabilities

Probabilities in BNs

- For all joint distributions, we have (chain rule):

$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i | x_1, \ldots, x_{i-1})$$

- Bayes' nets **implicitly** encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i | \mathit{parents}(X_i))$$

- This lets us reconstruct any entry of the full joint
- Not every BN can represent every joint distribution
  - The topology enforces certain conditional independencies

All Conditional Independences

- Given a Bayes net structure, can run d-separation to build a complete list of conditional independences that are necessarily true of the form

$$X_i \perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$$

- This list determines the set of probability distributions that can be represented

Same Assumptions, Different Graphs?

- Can you have two different graphs that encode the same assumptions?
  - Yes!
  - Examples:

Example: Independence

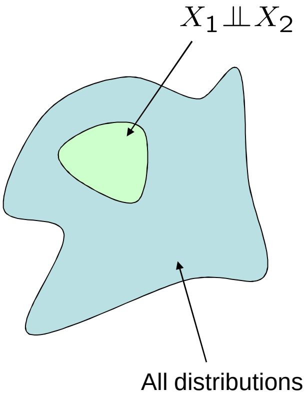
- For this graph, you can fiddle with θ (the CPTs) all you want, but you won't be able to represent any distribution in which the flips are dependent!

$P(X_1)$

h	0.5
t	0.5

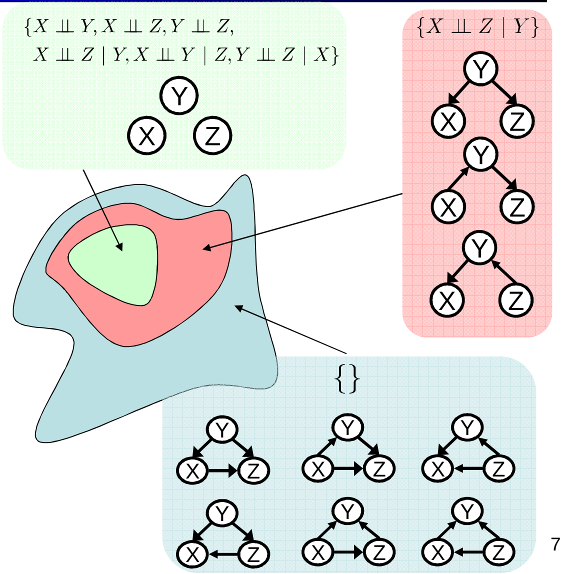
$P(X_2)$

h	0.5
t	0.5



Topology Limits Distributions

- Given some graph topology G, only certain joint distributions can be encoded
- The graph structure guarantees certain (conditional) independences
- (There might be more independence)
- Adding arcs increases the set of distributions, but has several costs
- Full conditioning can encode any distribution



Causality?

- When Bayes' nets reflect the true causal patterns:
  - Often simpler (nodes have fewer parents)
  - Often easier to think about
  - Often easier to elicit from experts

- BNs need not actually be causal
  - Sometimes no causal net exists over the domain
  - E.g. consider the variables *Traffic* and *Drips*
  - End up with arrows that reflect correlation, not causation

- What do the arrows really mean?
  - Topology may happen to encode causal structure
  - Topology only guaranteed to encode conditional independence

- *More about causality: [Causility – Judea Pearl]

Bayes Nets Representation Summary

- Bayes nets compactly encode joint distributions
- Guaranteed independencies of distributions can be deduced from BN graph structure
- D-separation gives precise conditional independence guarantees from graph alone
- A Bayes’ net’s joint distribution may have further (conditional) independence that is not detectable until you inspect its specific distribution

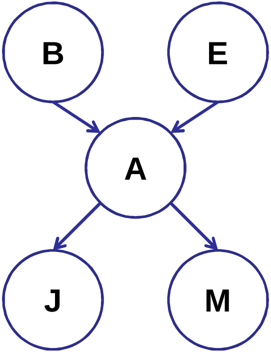
Inference

- Inference: calculating some useful quantity from a joint probability distribution
- Examples:
  - Posterior probability:

$$P(Q|E_1 = e_1, \ldots E_k = e_k)$$

  - Most likely explanation:

$$\text{argmax}_q \; P(Q = q|E_1 = e_1 \ldots)$$

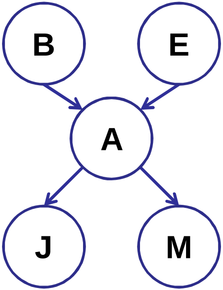


Inference by Enumeration

- Given unlimited time, inference in BNs is easy
- Recipe:
  - State the marginal probabilities you need
  - Figure out ALL the atomic probabilities you need
  - Calculate and combine them
- Example:

$$P(+b| + j, +m) =$$

$$\frac{P(+b, +j, +m)}{P(+j, +m)}$$



Example: Enumeration

- In this simple method, we only need the BN to synthesize the joint entries

$$P(+b,+j,+m) =$$

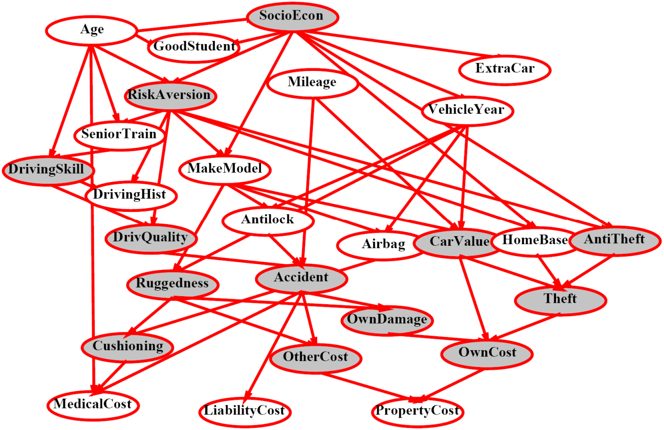
$$P(+b)P(+e)P(+a|+b,+e)P(+j|+a)P(+m|+a)+$$

$$P(+b)P(+e)P(-a|+b,+e)P(+j|-a)P(+m|-a)+$$

$$P(+b)P(-e)P(+a|+b,-e)P(+j|+a)P(+m|+a)+$$

$$P(+b)P(-e)P(-a|+b,-e)P(+j|-a)P(+m|-a)$$

Inference by Enumeration?



Variable Elimination

- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out the hidden variables
  - You end up repeating a lot of work!
- Idea: interleave joining and marginalizing!
  - Called “Variable Elimination”
  - Still NP-hard, but usually much faster than inference by enumeration
- We’ll need some new notation to define VE

Factor Zoo I

- Joint distribution: P(X,Y)
  - Entries P(x,y) for all x, y
  - Sums to 1

$$P(T,W)$$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

- Selected joint: P(x,Y)
  - A slice of the joint distribution
  - Entries P(x,y) for fixed x, all y
  - Sums to P(x)

$$P(cold,W)$$

T	W	P
cold	sun	0.2
cold	rain	0.3

Factor Zoo II

- Family of conditionals: P(X |Y)
  - Multiple conditionals
  - Entries P(x | y) for all x, y
  - Sums to |Y|

$$P(W|T)$$

T	W	P
hot	sun	0.8
hot	rain	0.2
cold	sun	0.4
cold	rain	0.6

$P(W|hot)$

$P(W|cold)$

- Single conditional: P(Y | x)
  - Entries P(y | x) for fixed x, all y
  - Sums to 1

$$P(W|cold)$$

T	W	P
cold	sun	0.4
cold	rain	0.6

Factor Zoo III

- Specified family: P(y | X)
  - Entries P(y | x) for fixed y, but for all x
  - Sums to … who knows!

$$P(rain|T)$$

T	W	P
hot	rain	0.2
cold	rain	0.6

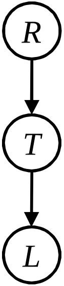
$P(rain|hot)$

$P(rain|cold)$

- In general, when we write $P(Y_1 \ldots Y_N \mid X_1 \ldots X_M)$
  - It is a "factor," a multi-dimensional array
  - Its values are all $P(y_1 \ldots y_N \mid x_1 \ldots x_M)$
  - Any assigned X or Y is a dimension missing (selected) from the array

Example: Traffic Domain

- Random Variables
  - R: Raining
  - T: Traffic
  - L: Late for class!
- First query: P(L)



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|R)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Variable Elimination Outline

- Track objects called factors
- Initial factors are local CPTs (one per node)

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- Any known values are selected
  - E.g. if we know $L = +\ell$, the initial factors are

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(+\ell|T)$

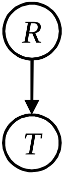
+t	+l	0.3
-t	+l	0.1

- VE: Alternately join factors and eliminate variables

Operation 1: Join Factors

- First basic operation: joining factors
- Combining factors:
  - Just like a database join
  - Get all factors over the joining variable
  - Build a new factor over the union of the variables involved

- Example: Join on R



$$P(R) \times P(T|R) \quad \longrightarrow \quad P(R,T)$$

$P(R)$:

+r	0.1
-r	0.9

$P(T|R)$:

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

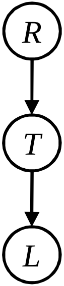
$P(R,T)$:

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

- Computation for each entry: pointwise products

$$\forall r, t : \quad P(r,t) = P(r) \cdot P(t|r)$$

Example: Multiple Joins



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

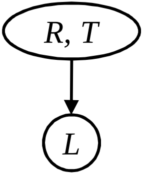
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join R

$P(R,T)$

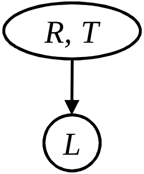
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81



$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Example: Multiple Joins



$P(R,T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join T

$P(R,T,L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729

Operation 2: Eliminate

- Second basic operation: marginalization
- Take a factor and sum out a variable
  - Shrinks a factor to a smaller one
  - A projection operation
- Example:

$P(R,T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

sum R →

$P(T)$

+t	0.17
-t	0.83

Multiple Elimination

R, T, L

T, L

L

$P(R, T, L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729

Sum out R

$P(T, L)$

+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

Sum out T

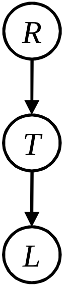
$P(L)$

+l	0.134
-l	0.886

P(L) : Marginalizing Early!

$P(R)$

+r	0.1
-r	0.9



$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

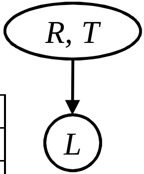
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join R

$P(R,T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81



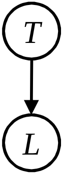
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Sum out R

$P(T)$

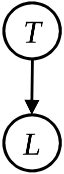
+t	0.17
-t	0.83



$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Marginalizing Early (aka VE*)



Join T

Sum out T

$P(T)$

+t	0.17
-t	0.83

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

$P(T, L)$

+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

$P(L)$

+l	0.134
-l	0.886

* VE is variable elimination

Evidence

- If evidence, start with factors that select that evidence
  - No evidence uses these initial factors:

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

  - Computing $P(L|+r)$, the initial factors become:

$P(+r)$

+r	0.1

$P(T|+r)$

+r	+t	0.8
+r	-t	0.2

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- We eliminate all vars other than query + evidence

Evidence II

- Result will be a selected joint of query and evidence
  - E.g. for P(L | +r), we'd end up with:

$$P(+r, L)$$

+r	+l	0.026
+r	-l	0.074

Normalize

$$P(L| + r)$$

+l	0.26
-l	0.74

- To get our answer, just normalize this!

- That's it!

General Variable Elimination

- Query: $P(Q|E_1 = e_1, \dots E_k = e_k)$

- Start with initial factors:
  - Local CPTs (but instantiated by evidence)

- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H

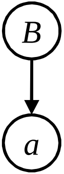
- Join all remaining factors and normalize

Variable Elimination Bayes Rule

Start / Select

$P(B)$

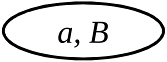
B	P
+b	0.1
¬b	0.9



$P(A|B) \rightarrow P(a|B)$

B	A	P
+b	+a	0.8
~~+b~~	~~¬a~~	~~0.2~~
¬b	+a	0.1
~~¬b~~	~~¬a~~	~~0.9~~

Join on B



$P(a, B)$

A	B	P
+a	+b	0.08
+a	¬b	0.09

Normalize

$P(B|a)$

A	B	P
+a	+b	8/17
+a	¬b	9/17

Example

$$P(B|j,m) \propto P(B,j,m)$$

$$\boxed{P(B) \qquad P(E) \qquad P(A|B,E) \qquad P(j|A) \qquad P(m|A)}$$

Choose A

$$\left.\begin{matrix} P(A|B,E) \\ P(j|A) \\ P(m|A) \end{matrix}\right. \xrightarrow{\times} P(j,m,A|B,E) \xrightarrow{\sum} P(j,m|B,E)$$

$$\boxed{P(B) \qquad P(E) \qquad P(j,m|B,E)}$$

Example

$$P(B) \qquad P(E) \qquad P(j,m|B,E)$$

Choose E

$$P(E),\; P(j,m|B,E) \xrightarrow{\times} P(j,m,E|B) \xrightarrow{\sum} P(j,m|B)$$

$$P(B) \qquad P(j,m|B)$$

Finish with B

$$P(B),\; P(j,m|B) \xrightarrow{\times} P(j,m,B) \xrightarrow{\text{Normalize}} P(B|j,m)$$

Variable Elimination

- What you need to know:
  - Should be able to run it on small examples, understand the factor creation / reduction flow
  - Better than enumeration: saves time by marginalizing variables as soon as possible rather than at the end
- We will see special cases of VE later
  - On tree-structured graphs, variable elimination runs in polynomial time, like tree-structured CSPs
  - You'll have to implement a tree-structured special case to track invisible ghosts (Project 4)