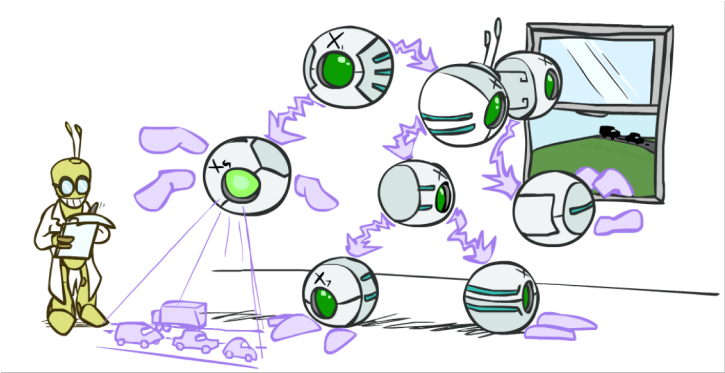
CSE 573-P: Artificial Intelligence

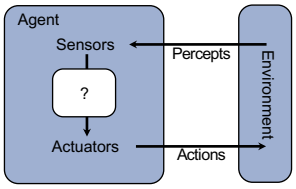
Bayes' Nets: Inference



Dan Weld

[Many slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]

Outline



- Decision Making
  - Static world, deterministic effects, full observability — A*, IDA*, heuristics
  - Multi-agent (adversarial & non-zero-sum objectives) — α-β, generalized Min-Max
  - Stochastic world, full observability, known model — MDP, VI, PI, RTDP
  - Stochastic world, full observability, unknown model — RL, Q-learning
  - Stochastic world, partial observability, known model — POMDP
  - Stochastic world, partial observability, unknown model — RL-POMDP
  - Stochastic world, partial observability, multi-agent — Dec-POMDP
- Knowledge Representation
  - Certain — propositional & FO logic
  - Uncertain — HMMs, Bayes nets, MLNs
- Ethics, Explanation, Control

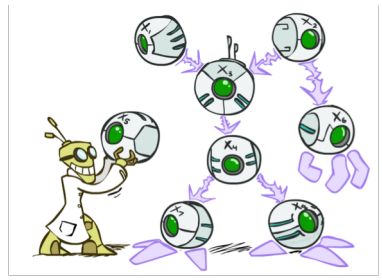
Bayes' Net Representation

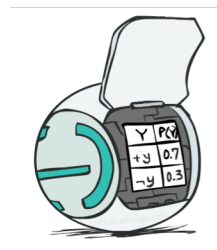
- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over X, one for each combination of parents' values

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

- 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:

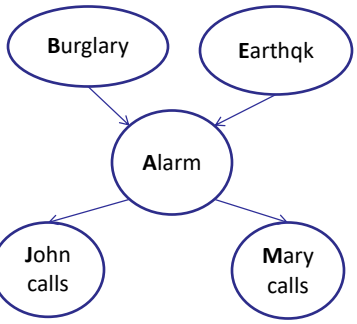
$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$





Example: Alarm Network

B	P(B)
+b	0.001
-b	0.999



E	P(E)
+e	0.002
-e	0.998

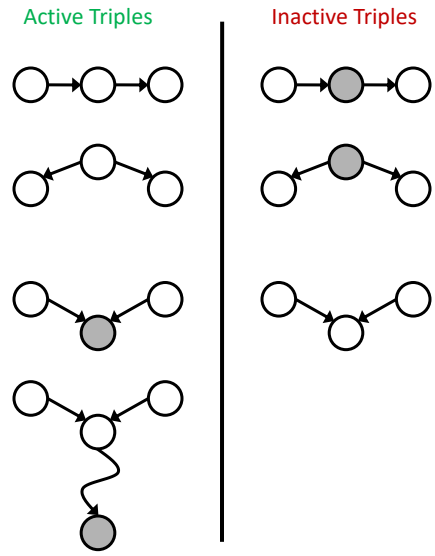
A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

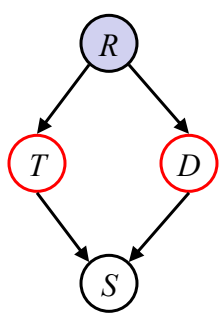
Active / Inactive Paths

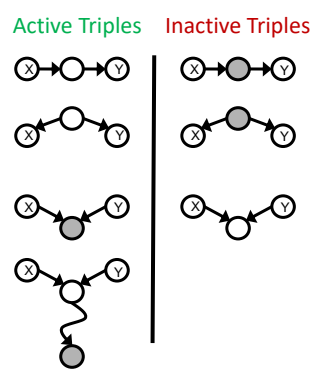
- Question: Are X and Y conditionally independent given evidence variables {Z}?
  - Yes, if X and Y "d-separated" by Z
- Consider all (undirected) paths from X to Y…
- If ***all*** paths are inactive → independence!
- A path is active if ***every*** triple is active:
  - Causal chain A → B → C where B is unobserved (either direction)
  - Common cause A ← B → C where B is unobserved
  - Common effect (aka v-structure)
    A → B ← C where B *or one of its descendants* is observed
- All it takes to block a path is a ***single*** inactive segment
  - But every ***path*** must be blocked



Example

- Variables:
  - R: Raining
  - T: Traffic
  - D: Roof drips
  - S: I'm sad
- Questions:



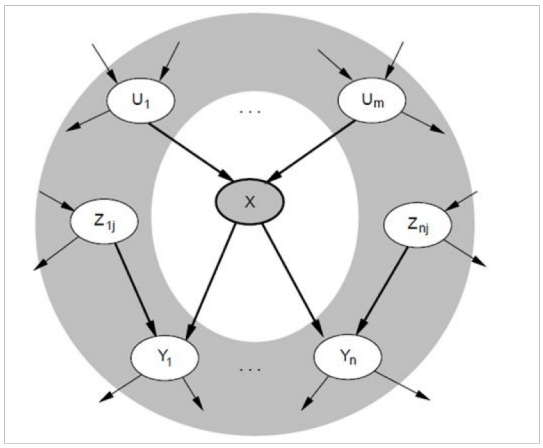


$T \perp\!\!\!\perp D$ *No*

$T \perp\!\!\!\perp D | R$ *Yes, Independent*

Markov Blanket

Markov Blanket (X) = Parents(X) ∪ Children(x) ∪Parents(Children(X))



7

Structure Implications

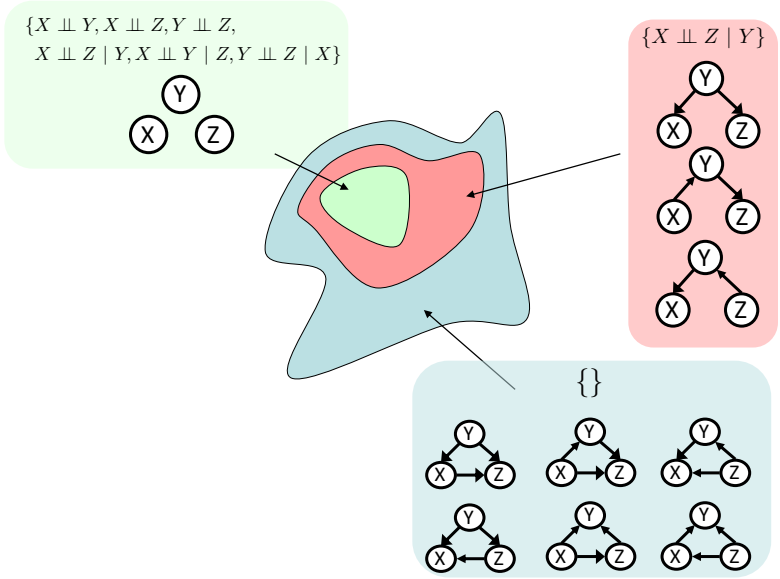
- Given a Bayes net structure, can run d-separation algorithm to build a complete list of conditional independences that are necessarily true of the form

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

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

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



Bayes' Nets

- ✔ Representation
- ✔ Conditional Independences
- Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Probabilistic inference is NP-complete
  - Sampling (approximate)
- Learning Bayes' Nets from Data

Inference

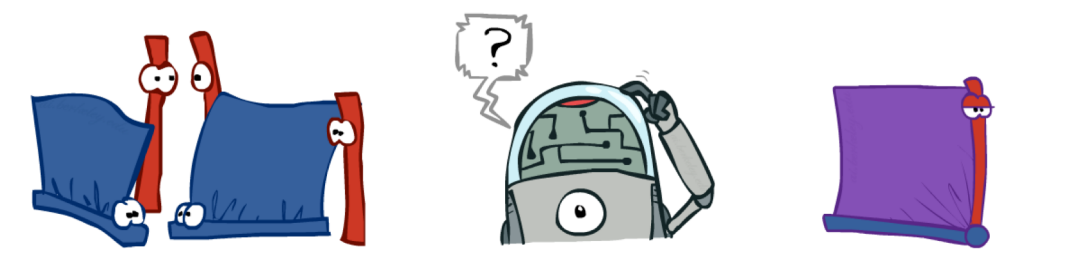
- Inference: calculating some useful quantity from a joint probability distribution

- Examples:
  - Posterior probability

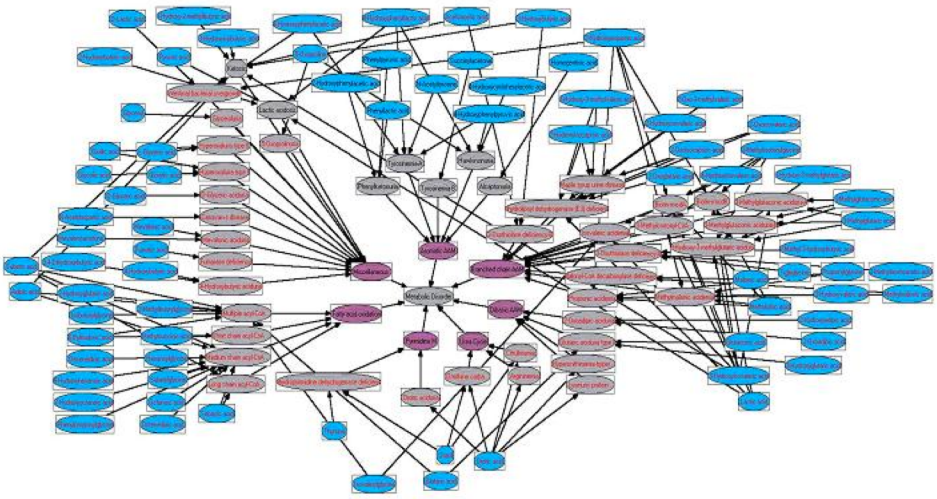
    $$P(Q|E_1 = e_1, \ldots E_k = e_k)$$

  - Most likely explanation:

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



Test for Infant Metabolic Defects



Blue ovals represent chromatographic peaks, grey ovals represent 20 metabolic diseases

Inference by Enumeration

- General case:
  - Evidence variables: $E_1 \dots E_k = e_1 \dots e_k$
  - Query* variable: Q
  - Hidden variables: $H_1 \dots H_r$

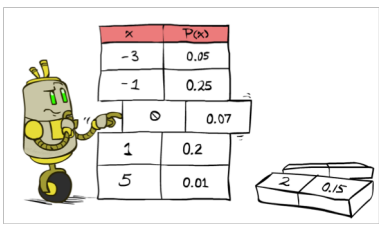
$\left.\right\}$ $X_1, X_2, \dots X_n$ *All variables*

- We want:

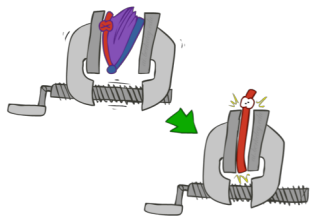
** Works fine with multiple query variables, too*

$$P(Q|e_1 \dots e_k)$$

- Step 1: Select the entries consistent with the evidence



- Step 2: Sum out H to get joint of Query and evidence



$$P(Q, e_1 \dots e_k) = \sum_{h_1 \dots h_r} \underbrace{P(Q, h_1 \dots h_r, e_1 \dots e_k)}_{X_1, X_2, \dots X_n}$$

- Step 3: Normalize

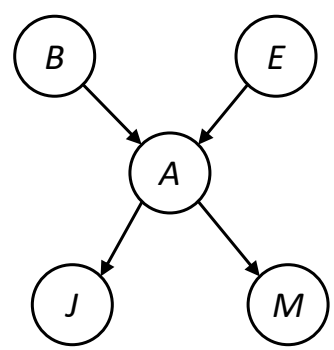
$$\times \frac{1}{Z}$$

$$Z = \sum_q P(Q, e_1 \cdots e_k)$$

$$P(Q|e_1 \cdots e_k) = \frac{1}{Z} P(Q, e_1 \cdots e_k)$$

Inference by Enumeration in Bayes' Net

- Given unlimited time, inference in BNs is easy
- Reminder of inference by enumeration by example:



$$P(B \mid +j, +m) \propto_B P(B, +j, +m)$$

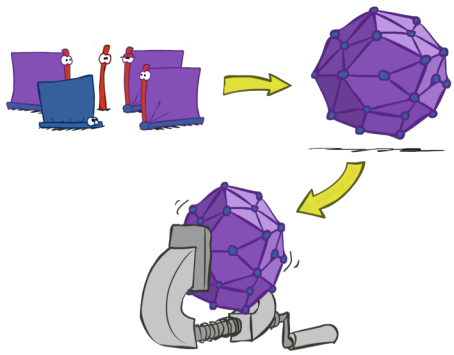
$$= \sum_{e,a} P(B, e, a, +j, +m)$$

$$= \sum_{e,a} P(B)P(e)P(a|B,e)P(+j|a)P(+m|a)$$

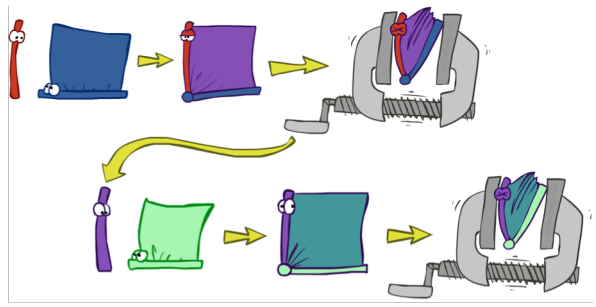
$$= P(B)P(+e)P(+a|B,+e)P(+j|+a)P(+m|+a) + P(B)P(+e)P(-a|B,+e)P(+j|-a)P(+m|-a)$$
$$P(B)P(-e)P(+a|B,-e)P(+j|+a)P(+m|+a) + P(B)P(-e)P(-a|B,-e)P(+j|-a)P(+m|-a)$$

Inference by Enumeration vs. Variable Elimination

- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out the hidden variables

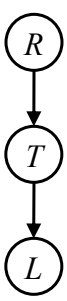


- Idea: interleave joining and marginalizing!
  - Called "Variable Elimination"
  - Still NP-hard, but usually much faster than inference by enumeration



- First we'll need some new notation: factors

Traffic Domain



$$P(L) = ?$$

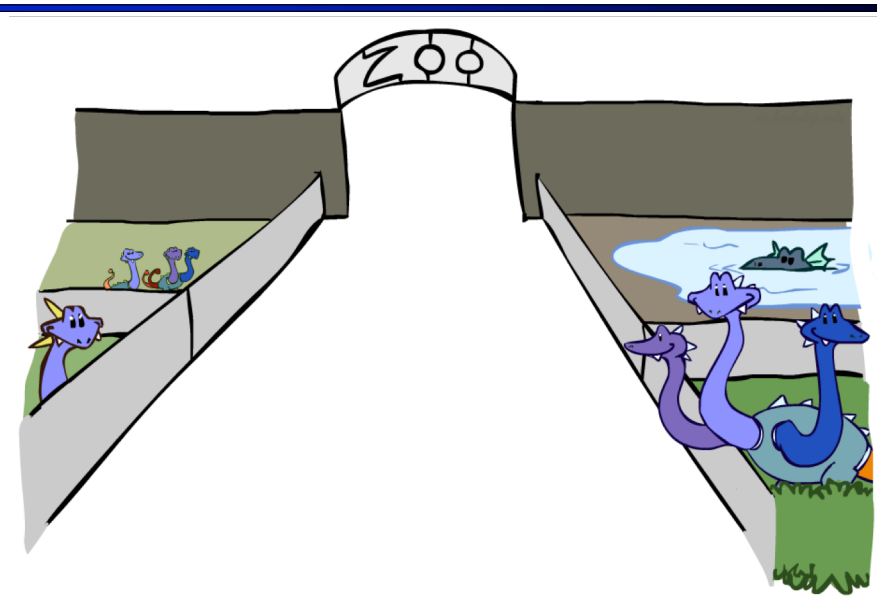
- Inference by Enumeration

$$= \underbrace{\sum_t \underbrace{\sum_r \underbrace{P(L|t)\underbrace{P(r)P(t|r)}_{\text{Join on r}}}_{\text{Join on t}}}_{\text{Eliminate r}}}_{\text{Eliminate t}}$$

- Variable Elimination

$$= \underbrace{\sum_t \underbrace{P(L|t) \underbrace{\sum_r \underbrace{P(r)P(t|r)}_{\text{Join on r}}}_{\text{Eliminate r}}}_{\text{Join on t}}}_{\text{Eliminate t}}$$

Factor Zoo



Factor Zoo I

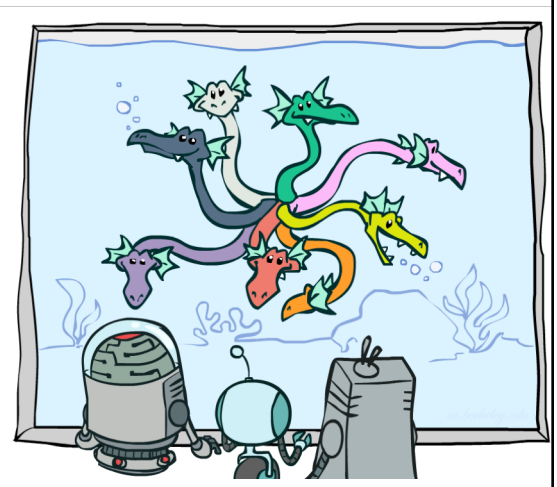
- Joint distribution: P(X,Y)
  - Entries P(x,y) for all x, y
  - Sums to 1
- Selected joint: P(x,Y)
  - A slice of the joint distribution
  - Entries P(x,y) for fixed x, all y
  - Sums to P(x)
- Number of capitals = dimensionality of the table

$P(T, W)$

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

$P(cold, W)$

T	W	P
cold	sun	0.2
cold	rain	0.3



Factor Zoo I

Two dimensions

	sun	rain
hot	0.4	0.1
cold	0.2	0.3

$P(T, W)$

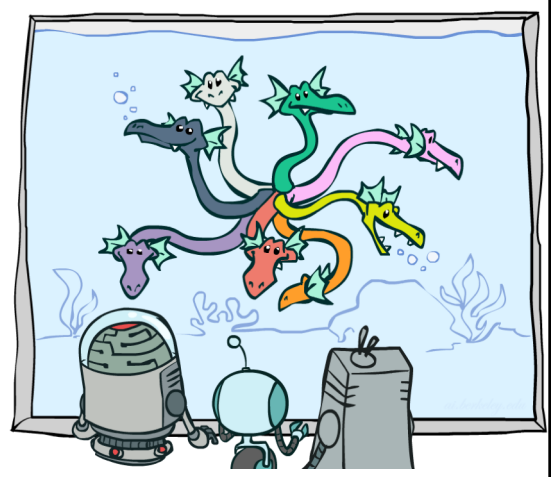
T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

One dimension

	sun	rain
cold	0.2	0.3

$P(cold, W)$

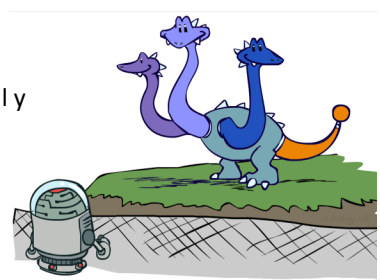
T	W	P
cold	sun	0.2
cold	rain	0.3



- Number of capitals = dimensionality of the table

Factor Zoo II

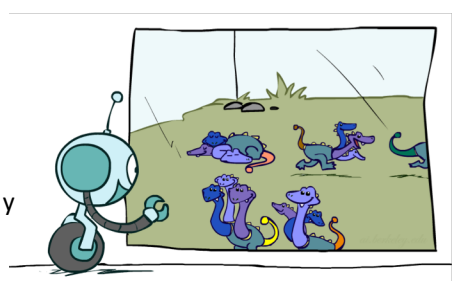
- Single conditional: P(Y | x)
  - Entries P(y | x) for fixed x, all y
  - Sums to 1



$P(W|cold)$

T	W	P
cold	sun	0.4
cold	rain	0.6

- 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

$P(W|hot)$

$P(W|cold)$

Factor Zoo III

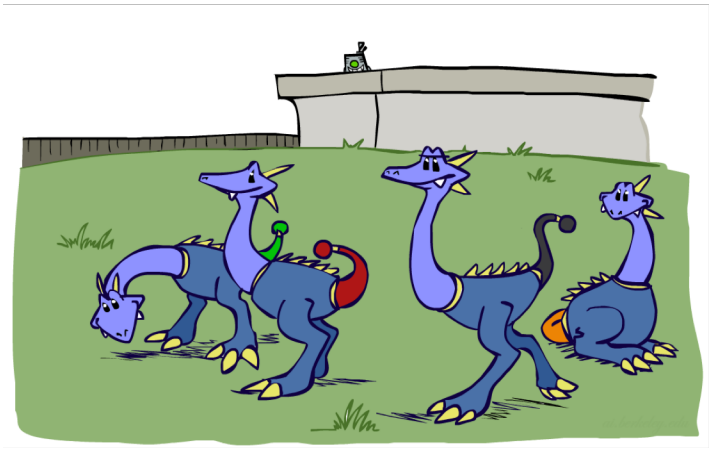
- Specified family: P(y | X)
  - Entries P(y | x) for fixed y, but for all x
  - Sums to … who knows!

$P(rain|T)$

T	W	P
hot	rain	0.2
cold	rain	0.6

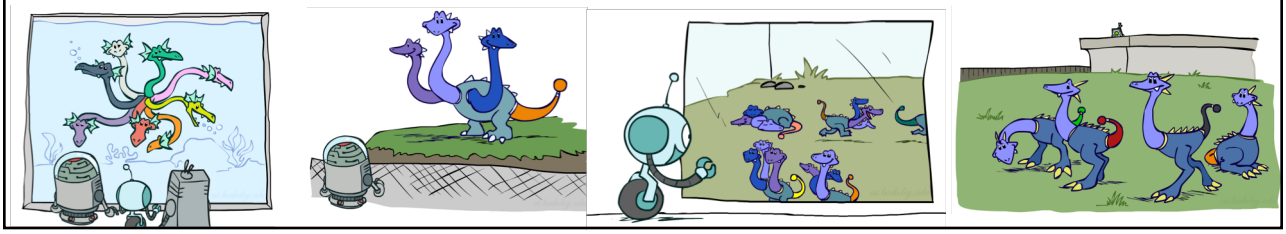
$P(rain|hot)$

$P(rain|cold)$



Factor Zoo Summary

- 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 $P(y_1 \ldots y_N \mid x_1 \ldots x_M)$
  - Any assigned (=lower-case) X or Y is a dimension missing (selected) from the array



Example: Traffic Domain

- Random Variables
  - R: Raining
  - T: Traffic
  - L: Late for class!

$$R \rightarrow T \rightarrow L$$

$$P(L) = ?$$
$$= \sum_{r,t} P(r,t,L)$$
$$= \sum_{r,t} P(r)P(t|r)P(L|t)$$

$P(R)$

+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

$P(L|T)$

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

Inference by Enumeration: Procedural 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

$P(R)$

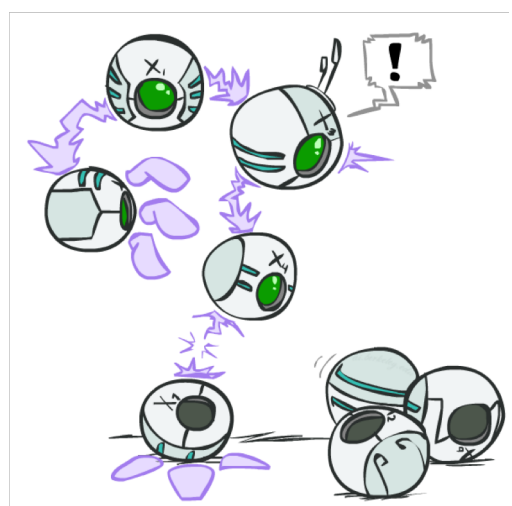
+r	0.1
-r	0.9

$P(T|R)$

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

$P(+\ell|T)$

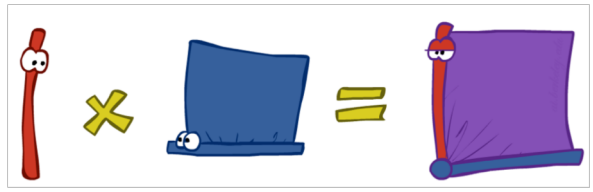
+t	+l	0.3
-t	+l	0.1



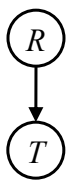
- Procedure: Join all factors, then eliminate all hidden variables

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



$$P(R) \times P(T|R) \Rightarrow P(R,T)$$

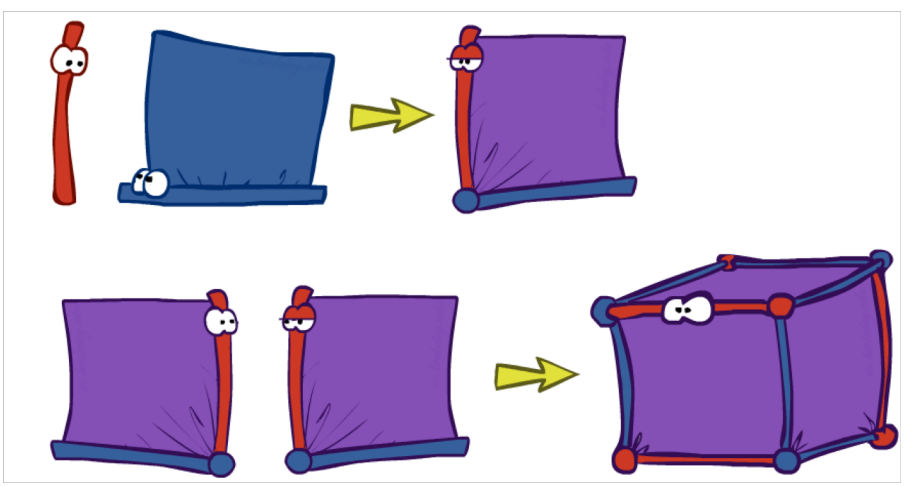
+r	0.1
-r	0.9

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

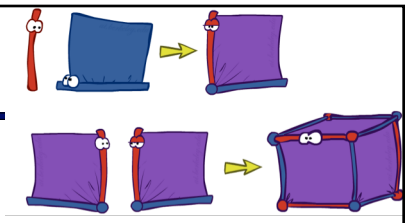
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

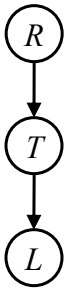
- Computation for each entry: pointwise products $\forall r, t: \quad P(r,t) = P(r) \cdot P(t|r)$

Example: Multiple Joins



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

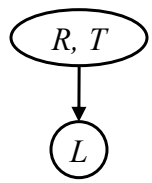
Join R

$P(R,T)$

+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



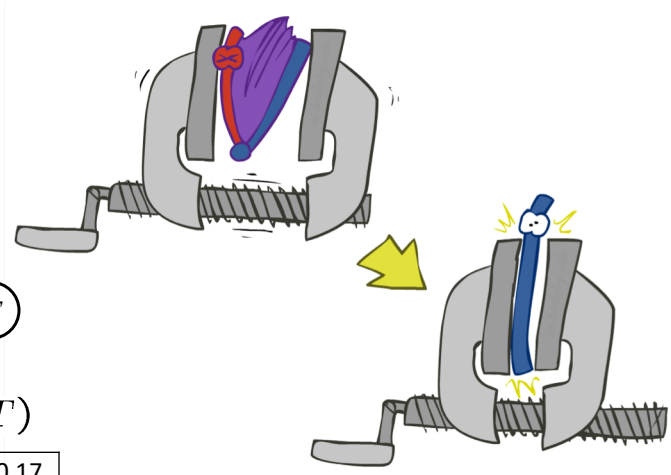
Join T

$P(R,T,L)$

+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
  - An aggregate/project 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

$P(R,T,L)$

+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

T, L

$P(T,L)$

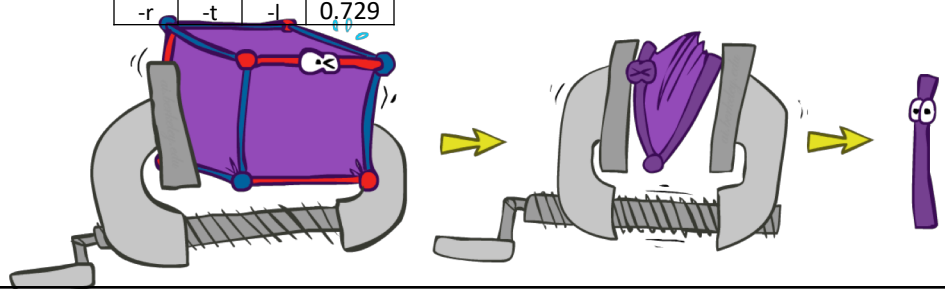
+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

Sum out T

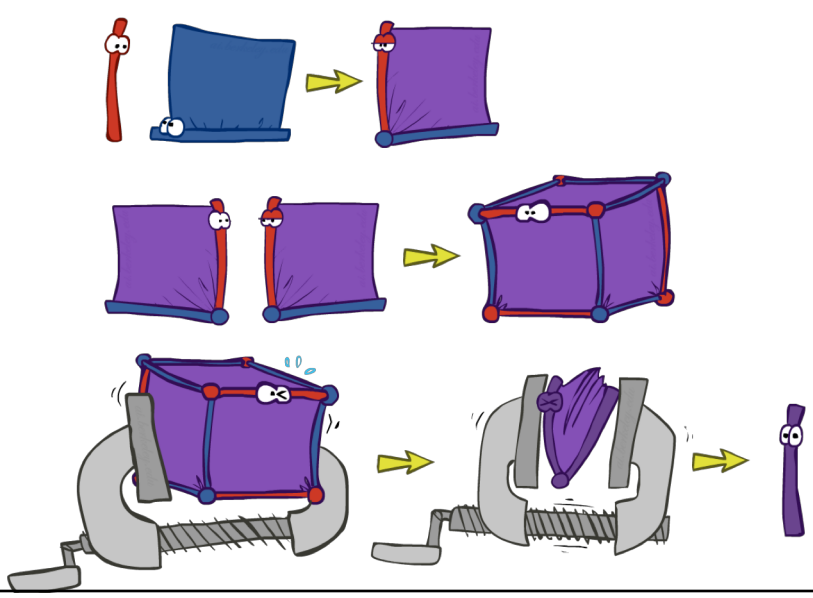
L

$P(L)$

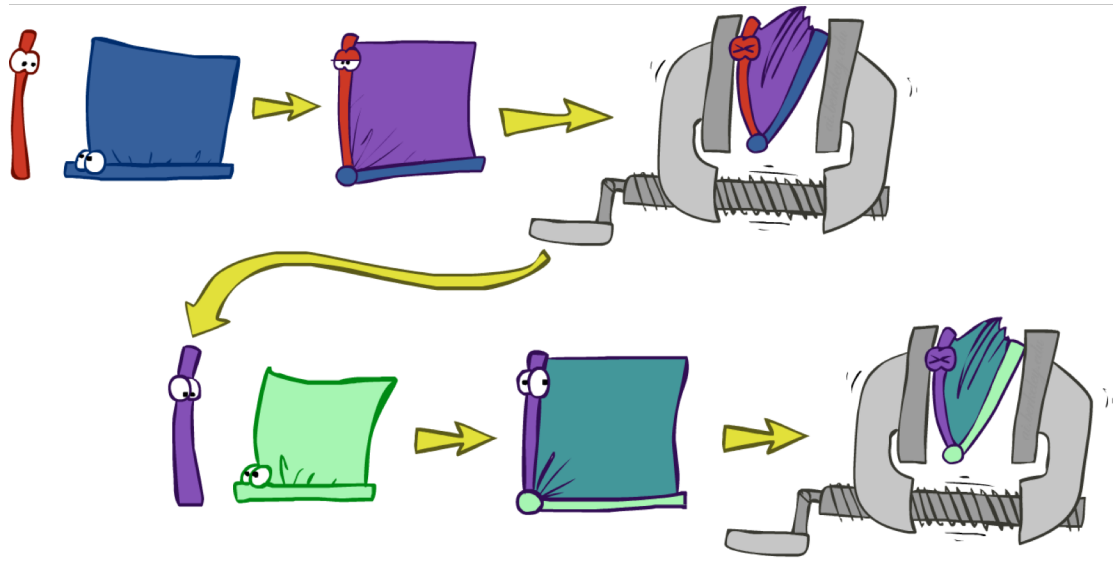
+l	0.134
-l	0.886



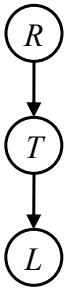
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)



Marginalizing Early (= Variable Elimination)



Traffic Domain



$$P(L) = ?$$

- Inference by Enumeration

$$= \underbrace{\sum_t \underbrace{\sum_r \underbrace{P(L|t)\underbrace{P(r)P(t|r)}_{\text{Join on r}}}_{\text{Join on t}}}_{\text{Eliminate r}}}_{\text{Eliminate t}}$$

- Variable Elimination

$$= \underbrace{\sum_t \underbrace{P(L|t)\underbrace{\sum_r \underbrace{P(r)P(t|r)}_{\text{Join on r}}}_{\text{Eliminate r}}}_{\text{Join on t}}}_{\text{Eliminate t}}$$

Marginalizing Early! (aka VE)

$P(R)$

+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

R, T

L

$P(L|T)$

+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

T

L

$P(L|T)$

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

Join T

T, L

$P(T,L)$

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

Sum out T

L

$P(L)$

+l	0.134
-l	0.866

Evidence

- If evidence, start with factors that select that evidence
  - If there is no evidence, then use these initial factors:

$P(R)$

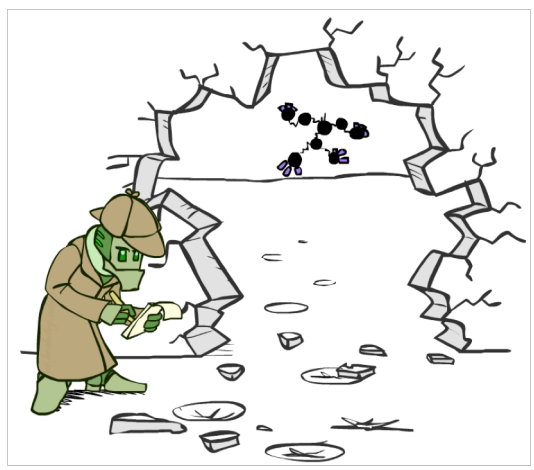
+r	0.1
-r	0.9

$P(T|R)$

+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



  - But if given some evidence, eg +r, then select for it…
  - 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

- Next do joins & eliminate, removing 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 would end up with:

$P(+r, L)$

+r	+l	0.026
+r	-l	0.074

Normalize

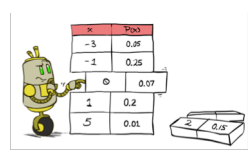
$P(L|+r)$

+l	0.26
-l	0.74

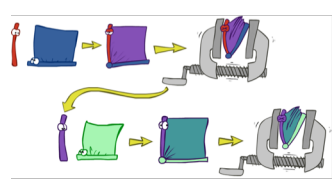
- To get our answer, just normalize this!
- That 's it!

General Variable Elimination

- Query: $P(Q|E_1 = e_1, \ldots E_k = e_k)$
- Start with initial factors:
  - Local CPTs (but instantiated by evidence)



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



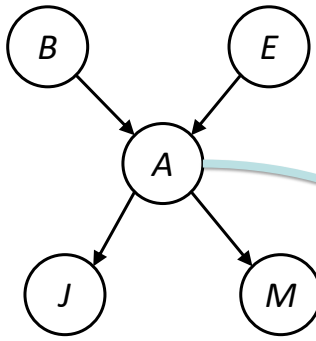
- Join all remaining factors and normalize

$$\times \frac{1}{Z}$$

Example: Alarm Network

P(B | j, m) = ?

B	P(B)
+b	0.001
-b	0.999



E	P(E)
+e	0.002
-e	0.998

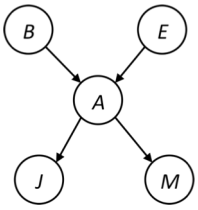
B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

Example

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



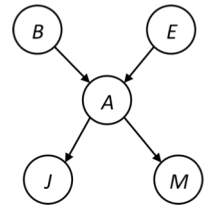
$$\boxed{P(B) \qquad P(E) \qquad P(A|B,E) \qquad P(j|A) \qquad P(m|A)}$$

Choose A

$$\begin{matrix} P(A|B,E) \\ P(j|A) \\ P(m|A) \end{matrix} \xrightarrow{\times} P(j,m,A|B,E) \xrightarrow{\Sigma} P(j,m|B,E)$$

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

Example



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

Choose E

$P(E)$

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

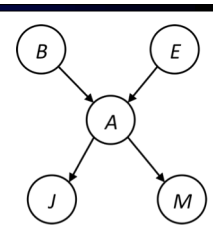
$$P(B) \qquad P(j,m|B)$$

Finish with B

$P(B)$

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

Same Example in Equations



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

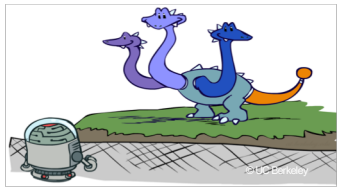
$$P(B) \qquad P(E) \qquad P(A|B,E) \qquad P(j|A) \qquad P(m|A)$$

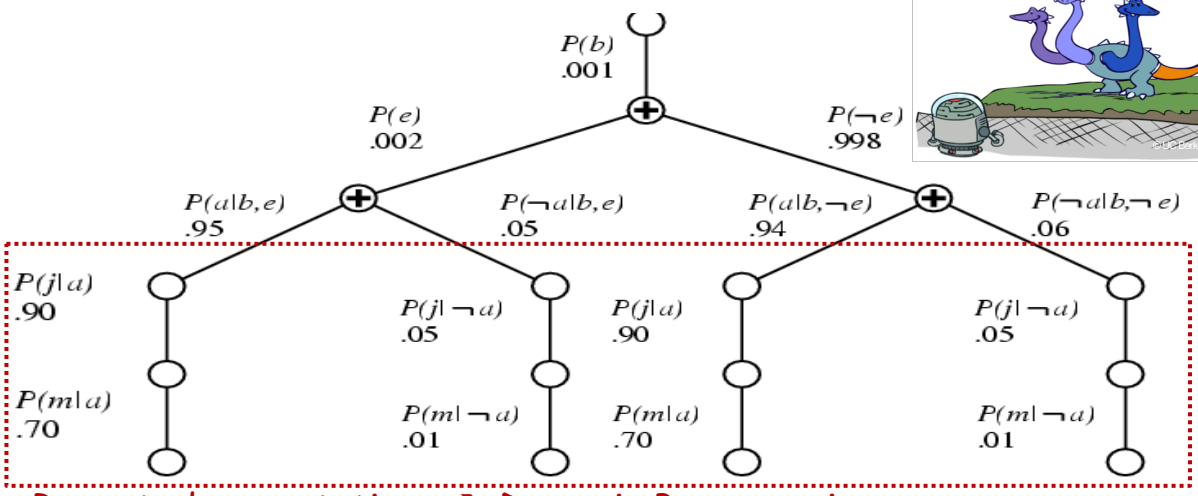
$$\begin{aligned}
P(B|j,m) &\propto P(B,j,m) \\
&= \sum_{e,a} P(B,j,m,e,a) && \text{marginal can be obtained from joint by summing out} \\
&= \sum_{e,a} P(B)P(e)P(a|B,e)P(j|a)P(m|a) && \text{use Bayes' net joint distribution expression} \\
&= \sum_{e} P(B)P(e)\sum_{a} P(a|B,e)P(j|a)P(m|a) && \text{use xy + xz= x*(y+z)} \quad \textit{do sum first} \\
&= \sum_{e} P(B)P(e)f_1(B,e,j,m) && \text{joining on a, and then summing out gives } f_1 \\
&= P(B)\sum_{e} P(e)f_1(B,e,j,m) && \text{use xy + xz = x*(y+z)} \quad \textit{do sum first} \\
&= P(B)f_2(B,j,m) && \text{joining on e, and then summing out gives } f_2
\end{aligned}$$

Simple! Exploiting uwy + uwz + uxy + uxz + vwy + vwz + vxy +vxz = (u+v)(w+x)(y+z) to reduce computation

Variable Elimination *vs.* Inference by Enumeration

$$P(b|j,m) = \alpha P(b) \sum_e P(e) \sum_a P(a|b,e)P(j|a)P(m,a)$$



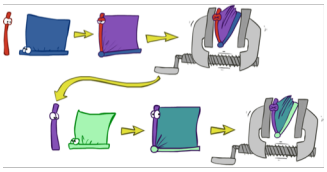


Repeated computations ➔ Dynamic Programming

Choices during Variable Elimination

- Query: $P(Q|E_1 = e_1, \ldots E_k = e_k)$
- Start with initial factors:
  - Local CPTs (but instantiated by evidence)

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



- Join all remaining factors and normalize

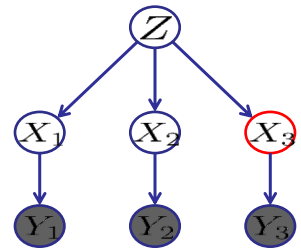
$$\times \frac{1}{Z}$$

Another Variable Elimination Example

Query: $P(X_3|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3)$

Start by inserting evidence, which gives the following initial factors:

$$p(Z)p(X_1|Z)p(X_2|Z)p(X_3|Z)p(y_1|X_1)p(y_2|X_2)p(y_3|X_3)$$



What variables could we eliminate?

Another Variable Elimination Example

Query: $P(X_3|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3)$

Start by inserting evidence, which gives the following initial factors:

$$p(Z)p(X_1|Z)p(X_2|Z)p(X_3|Z)p(y_1|X_1)p(y_2|X_2)p(y_3|X_3)$$

Eliminate X_1, this introduces the factor $f_1(Z, y_1) = \sum_{x_1} p(x_1|Z)p(y_1|x_1)$, and we are left with:

$$p(Z)f_1(Z, y_1)p(X_2|Z)p(X_3|Z)p(y_2|X_2)p(y_3|X_3)$$

Eliminate X_2, this introduces the factor $f_2(Z, y_2) = \sum_{x_2} p(x_2|Z)p(y_2|x_2)$, and we are left with:

$$p(Z)f_1(Z, y_1)f_2(Z, y_2)p(X_3|Z)p(y_3|X_3)$$

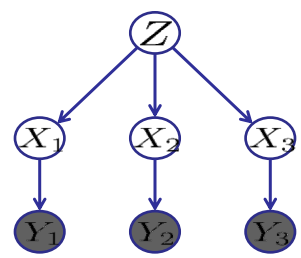
Eliminate Z, this introduces the factor $f_3(y_1, y_2, X_3) = \sum_z p(z)f_1(z, y_1)f_2(z, y_2)p(X_3|z)$, and we are left:

$$p(y_3|X_3), f_3(y_1, y_2, X_3)$$

No hidden variables left. Join the remaining factors to get:

$$f_4(y_1, y_2, y_3, X_3) = P(y_3|X_3)f_3(y_1, y_2, X_3).$$

Normalizing over X_3 gives $P(X_3|y_1, y_2, y_3)$.



What dimension are f_1, f_2 & f_3?

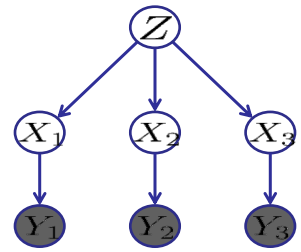
1

Another Variable Elimination Example

Query: $P(X_3|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3)$

Start by inserting evidence, which gives the following initial factors:

$$p(Z)p(X_1|Z)p(X_2|Z)p(X_3|Z)p(y_1|X_1)p(y_2|X_2)p(y_3|X_3)$$



Alternatively, suppose we start by eliminating Z:

$P(X_1 \mid Z)$

$P(X_2 \mid Z)$

$P(X_3 \mid Z)$

$\times$ $\Rightarrow$ $f_Z(X_1, X_2, X_3)$

$p(y_1 \mid X_1)$

$p(y_2 \mid X_2)$

$p(y_3 \mid X_3)$

What is the resulting factor?

What dimension is it? 3

How many entries? k^3

Another Variable Elimination Example

Query: $P(X_3|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3)$

Start by inserting evidence, which gives the following initial factors:

$$p(Z)p(X_1|Z)p(X_2|Z)p(X_3|Z)p(y_1|X_1)p(y_2|X_2)p(y_3|X_3)$$

Eliminate X_1, this introduces the factor $f_1(Z, y_1) = \sum_{x_1} p(x_1|Z)p(y_1|x_1)$, and we are left with:

$$p(Z)f_1(Z, y_1)p(X_2|Z)p(X_3|Z)p(y_2|X_2)p(y_3|X_3)$$

Eliminate X_2, this introduces the factor $f_2(Z, y_2) = \sum_{x_2} p(x_2|Z)p(y_2|x_2)$, and we are left with:

$$p(Z)f_1(Z, y_1)f_2(Z, y_2)p(X_3|Z)p(y_3|X_3)$$

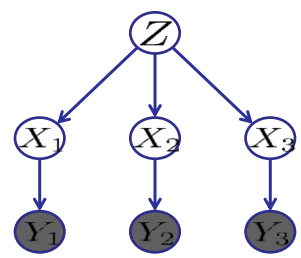
Eliminate Z, this introduces the factor $f_3(y_1, y_2, X_3) = \sum_z p(z)f_1(z, y_1)f_2(z, y_2)p(X_3|z)$, and we are left:

$$p(y_3|X_3), f_3(y_1, y_2, X_3)$$

No hidden variables left. Join the remaining factors to get:

$$f_4(y_1, y_2, y_3, X_3) = P(y_3|X_3)f_3(y_1, y_2, X_3).$$

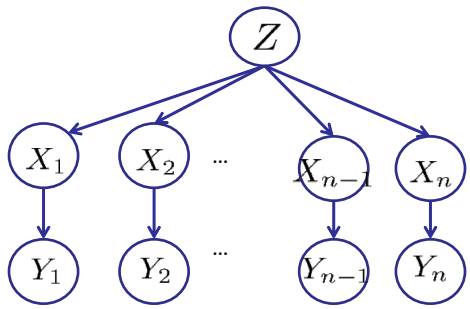
Normalizing over X_3 gives $P(X_3|y_1, y_2, y_3)$.



Computational complexity depends on the ***largest factor*** generated by the process.
Size of factor = number of entries in table.

Variable Elimination Ordering

- For the query $P(X_n|y_1,\ldots,y_n)$ work through the following two different orderings as done in previous slide: Z, X_1, …, X_{n-1} and X_1, …, X_{n-1}, Z. What is the size of the maximum factor generated for each of the orderings?



- Answer: 2^{n+1} versus 2^2 (assuming binary)
- In general: the ordering can ***greatly*** affect efficiency.

VE: Computational and Space Complexity

- Both computational & space complexity of variable elimination is determined by the largest factor
- Elimination ordering greatly affects the size of the largest factor.
  - E.g., previous slide's example 2^n vs. 2
- Does there always exist an ordering that yields small factors?
  - No!

Worst Case Complexity?

3-Satisfiability

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee \neg x_4) \wedge (x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5) \wedge (x_2 \vee x_5 \vee x_7) \wedge (x_4 \vee x_5 \vee x_6) \wedge (\neg x_5 \vee x_6 \vee \neg x_7) \wedge (\neg x_5 \vee \neg x_6 \vee x_7)$$

$P(X_i = 0) = P(X_i = 1) = 0.5$

$Y_1 = X_1 \vee X_2 \vee \neg X_3$

...

$Y_8 = \neg X_5 \vee X_6 \vee X_7$

$Y_{1,2} = Y_1 \wedge Y_2$

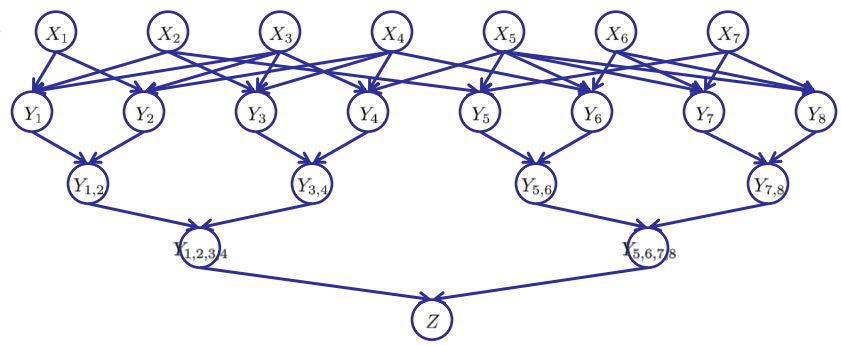
...

$Y_{7,8} = Y_7 \wedge Y_8$

$Y_{1,2,3,4} = Y_{1,2} \wedge Y_{3,4}$

$Y_{5,6,7,8} = Y_{5,6} \wedge Y_{7,8}$

$Z = Y_{1,2,3,4} \wedge Y_{5,6,7,8}$

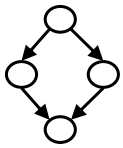


- If we can answer P(z) equal to zero or not, we answered whether the 3-SAT problem has a solution.
- Hence inference in Bayes' nets is NP-hard. No known efficient probabilistic inference in general.

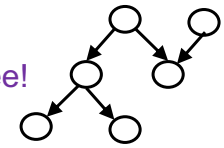
Polytrees

- A ***polytree*** is a directed graph ***with no undirected cycles***
  - *I.e. a tree (or forest) with multiple roots*

Not a poly tree!



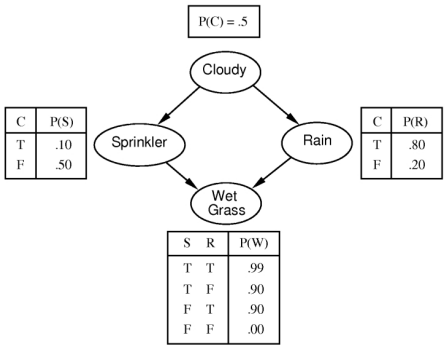
Yes! poly tree!



- For poly-trees you can always find an ordering that is efficient
  - Try it!!
- Cut-set conditioning for Bayes' net inference
  - Choose set of variables such that if removed only a polytree remains
  - Exercise: Think about how the specifics would work out!

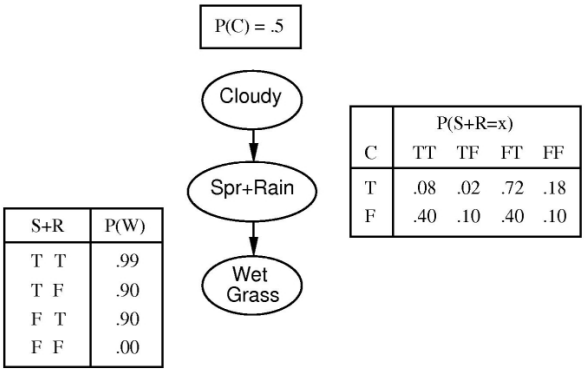
Inference in Multiply Connected Networks

- Clustering
- Cutset conditioning
- Stochastic methods
  - Rejection Sampling
  - Likelihood weighting
  - Markov Chain Monte Carlo

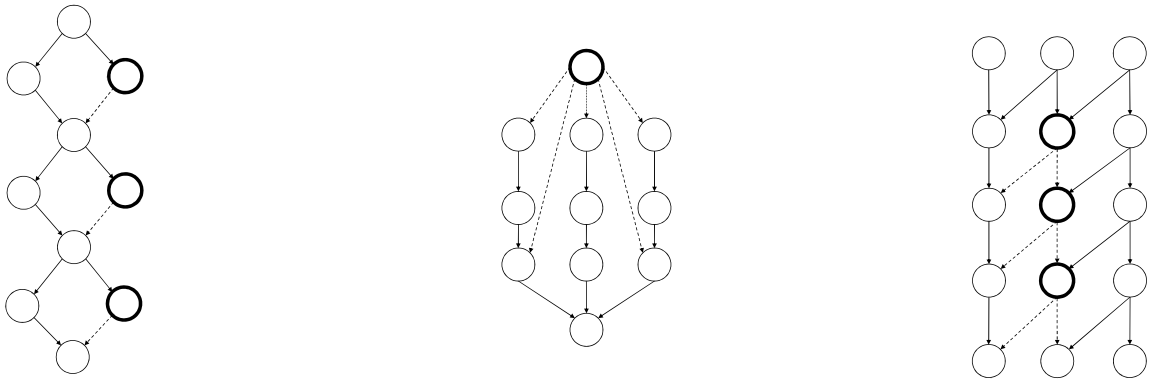


Clustering

- Merge variables, replace their CPTs by their combined CPT



A set of nodes, C, is a ***loop cutset***, if removing C yields a polytree



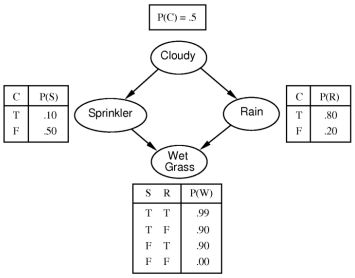
Finding smallest loop cutset is NP hard

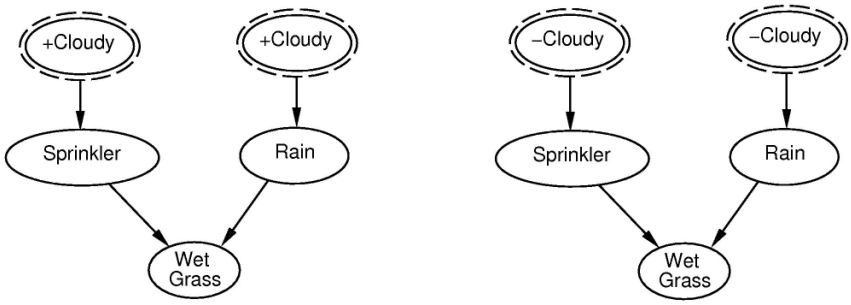
Figure from Jinbo Huang http://users.cecs.anu.edu.au/~jinbo/pr/lecture03.pdf

59

Cutset conditioning

- cutting off cycles to obtain polytree.
- Sum over all instantiations of cutset variables.





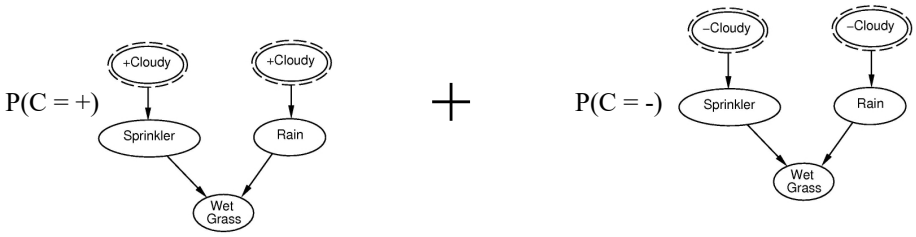
Cutset conditioning - example

$P(W|S) = \sum_{RC} P(W|S,R,C)\, P(R,C|S)$

$= \sum_{RC} P(W|S,R)\, P(R,C|S)$

$P(R,C|S) = P(R,S|C)\, P(C) / P(S) = P(R|C)\, P(S|C)\, P(C) / P(S)$

(Notice here that node C d-separates R and S)



Stochastic Inference

- It's expensive to work with the full joint distribution… whether as a table or as a Bayes Network
- Hmmm. What did we do for big HMMs?
- Approximate with Sampling Methods! (aka Monte Carlo)

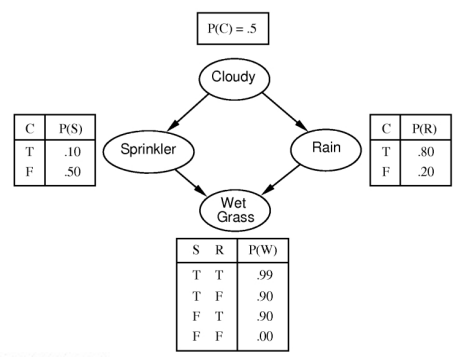
Prior sampling

Sampling algorithm that generates events from a Bayesian network

function PRIOR-SAMPLE(bn) **returns** an event sampled from the prior specified by bn
  inputs: bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  $\mathbf{x} \leftarrow$ an event with n elements
  foreach variable X_i **in** $X_1, \ldots, X_n$ **do**
    $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return x

Prior Sampling Example



1. Sample from $\mathbf{P}(Cloudy) = \langle 0.5, 0.5 \rangle$; suppose this returns $true$.
2. Sample from $\mathbf{P}(Sprinkler | Cloudy = true) = \langle 0.1, 0.9 \rangle$; suppose this returns $false$.
3. Sample from $\mathbf{P}(Rain | Cloudy = true) = \langle 0.8, 0.2 \rangle$; suppose this returns $true$.
4. Sample from $\mathbf{P}(WetGrass | Sprinkler = false, Rain = true) = \langle 0.9, 0.1 \rangle$; suppose this returns $true$.

In this case, PRIOR-SAMPLE returns the event $[true, false, true, true]$.

Approximating true distribution

With enough samples, perfect modeling is possible

$$\lim_{N\to\infty} \frac{N_{PS}(x_1,\ldots,x_n)}{N} = S_{PS}(x_1,\ldots,x_n) = P(x_1,\ldots,x_n)$$

Rejection Sampling

Compute P(X | e)

- Use *PriorSample* (S_{PS}) and create *N* samples
- Denote N_e number of samples consistent with *e (namely E = e)*
- Denote N_{ex} number of samples consistent with *E = e)* AND X = x
- P(X | e) can be computed from N_{ex} / N_e

Example

- P(Rain | Sprinkler = true)
- Use Bayes Net to generate 100 samples
  - Suppose 73 have *Sprinkler=false*
  - Suppose 27 have *Sprinkler=true*
    - 8 have *Rain=true*
    - 19 have *Rain=false*
- P(Rain | Sprinkler=true) = 8 / 19

Rejection sampling

```
function REJECTION-SAMPLING(X, e, bn, N) returns an estimate of P(X|e)
   inputs: X, the query variable
           e, observed values for variables E
           bn, a Bayesian network
           N, the total number of samples to be generated
   local variables: N, a vector of counts for each value of X, initially zero

   for j = 1 to N do
      x ← PRIOR-SAMPLE(bn)
      if x is consistent with e then
         N[x] ← N[x]+1 where x is the value of X in x
   return NORMALIZE(N)
```

Problems with Rejection Sampling

Standard deviation of the error in probability is proportional to 1/sqrt(n), where n is the number of samples consistent with evidence

As problems become complex, number of samples consistent with evidence becomes small and it becomes harder to construct accurate estimates

Likelihood Weighting

We only want to generate samples
that are consistent with the evidence, e

- Idea: We'll sample the Bayes Net, but we won't let every random variable be sampled, some will be forced to produce a specific output

Likelihood Weighting algorithm

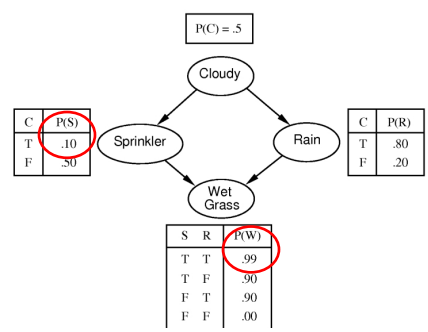
```
function LIKELIHOOD-WEIGHTING(X, e, bn, N) returns an estimate of P(X|e)
   inputs: X, the query variable
           e, observed values for variables E
           bn, a Bayesian network specifying joint distribution P(X_1, ..., X_n)
           N, the total number of samples to be generated
   local variables: W, a vector of weighted counts for each value of X, initially zero

   for j = 1 to N do
      x, w ← WEIGHTED-SAMPLE(bn, e)
      W[x] ← W[x] + w where x is the value of X in x
   return NORMALIZE(W)
```

```
function WEIGHTED-SAMPLE(bn, e) returns an event and a weight

   w ← 1; x ← an event with n elements initialized from e
   foreach variable X_i in X_1, ..., X_n do
      if X_i is an evidence variable with value x_i in e
         then w ← w × P(X_i = x_i | parents(X_i))
         else x[i] ← a random sample from P(X_i | parents(X_i))
   return x, w
```

Likelihood Weighting Example



Estimate P (Rain| sprinkler, wetGrass)

First, weight vector, w := 1.0

Consider variables in order of topological sort:

1. Sample from $\mathbf{P}(Cloudy) = \langle 0.5, 0.5 \rangle$; suppose this returns *true*.
2. *Sprinkler* is an evidence variable with value *true*. Therefore, we set
$$w \leftarrow w \times P(Sprinkler = true | Cloudy = true) = 0.1\ .$$
3. Sample from $\mathbf{P}(Rain | Cloudy = true) = \langle 0.8, 0.2 \rangle$; suppose this returns *true*.
4. *WetGrass* is an evidence variable with value *true*. Therefore, we set
$$w \leftarrow w \times P(WetGrass = true | Sprinkler = true, Rain = true) = 0.099$$

Discussion

Notice that weight is reduced according to how likely an evidence variable's output is given its parents

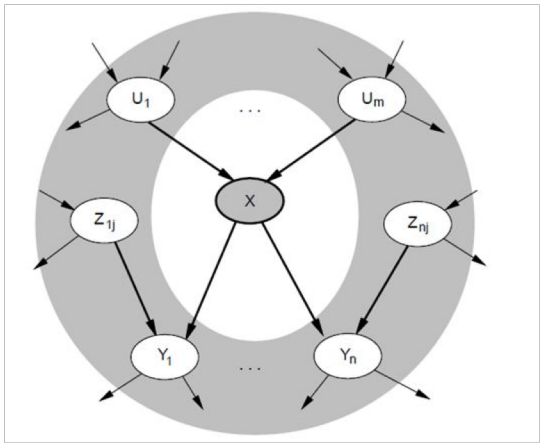
- So final probability is a function of what comes from sampling the *free* variables while constraining the evidence variables

Comparing techniques

- In rejection sampling, evidence variables are considered *after* the sampling
  - Accurate ☺
  - But slow ☹
- In likelihood weighting, attention is paid to evidence variables *before* samples are collected
  - Uses all samples ☺
  - Less accurate ☹
  - Because the sampled variables *ignore evidence* among z's non-ancestors

Reminder: Markov Blanket

Markov Blanket (X) = Parents(X) ∪ Children(x) ∪Parents(Children(X))



77

MCMC with Gibbs Sampling

- Fix the values of observed variables
- Set the values of all non-observed variables randomly
- Perform a random walk through the space of complete variable assignments.
  On each move:
  1. Pick a variable X
  2. Calculate Pr(X=true | Markov blanket)
  3. Set X to true with that probability
- Repeat many times.
- Frequency with which any variable X is true is it's posterior probability.

Converges to true posterior when frequencies stop changing significantly

stable distribution, mixing

78

Example

- See https://www.youtube.com/watch?v=a_08GKWHFWo
  - Starting at 4:30 they give an example (animation) for the case of a bivariate normal. The probability distribution is continuous, but it still illustrates the idea

79

Bayes' Nets

- ✔ Representation
- ✔ Conditional Independences
- Probabilistic Inference
  - ✔ Enumeration (exact, exponential complexity)
  - ✔ Variable elimination (exact, worst-case exponential complexity, often better)
  - ✔ Inference is NP-complete
  - ✔ Sampling (approximate)
- Learning Bayes' Nets from Data