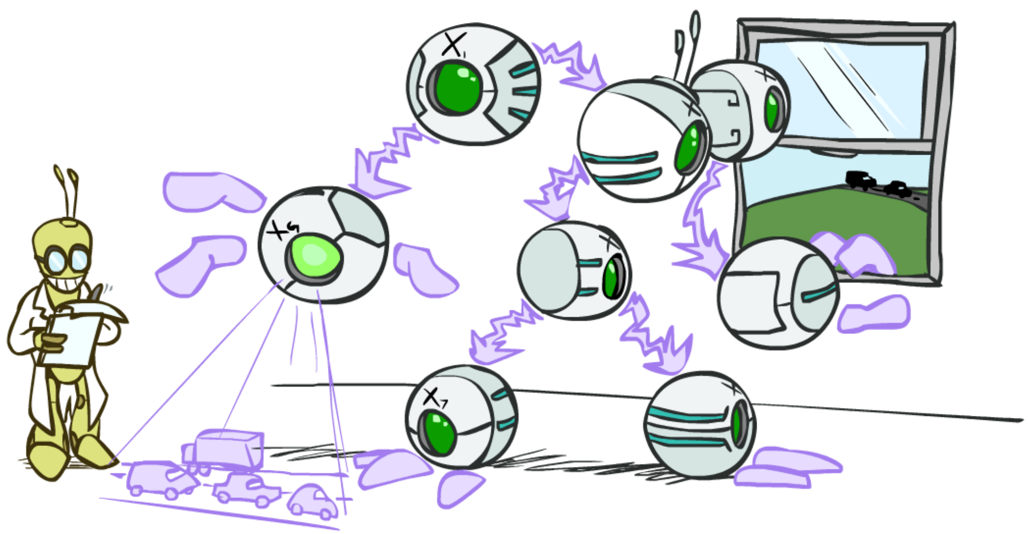
CS 188: Artificial Intelligence

Bayes' Nets: Inference



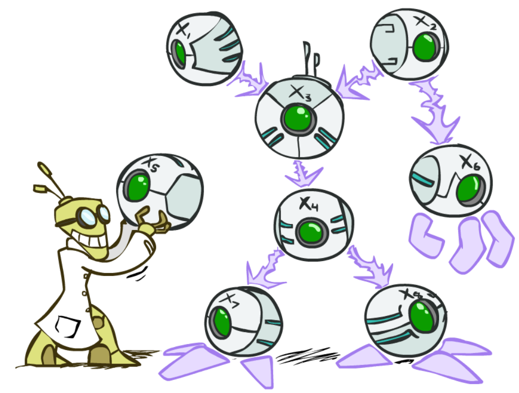
Pieter Abbeel and Dan Klein

University of California, Berkeley

Bayes' Net Representation

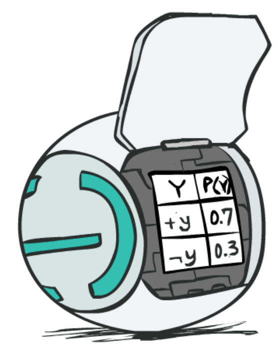
- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over X, one for each combination of parents' values

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



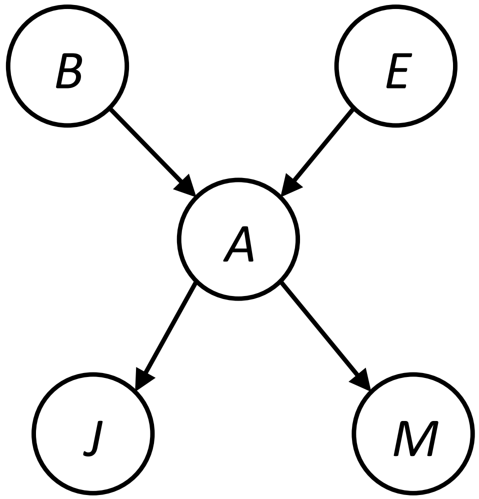
- Bayes' nets implicitly encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

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

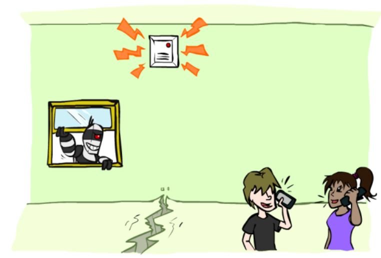


Example: Alarm Network

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

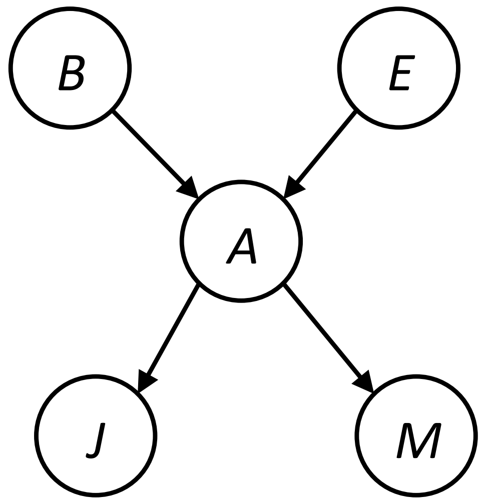
A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

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

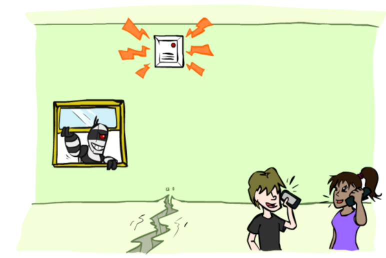
$$P(+b, -e, +a, -j, +m) =$$
$$P(+b)P(-e)P(+a|+b, -e)P(-j|+a)P(+m|+a) =$$

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

$$P(+b, -e, +a, -j, +m) =$$
$$P(+b)P(-e)P(+a| + b, -e)P(-j| + a)P(+m| + a) =$$
$$0.001 \times 0.998 \times 0.94 \times 0.1 \times 0.7$$

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

Inference

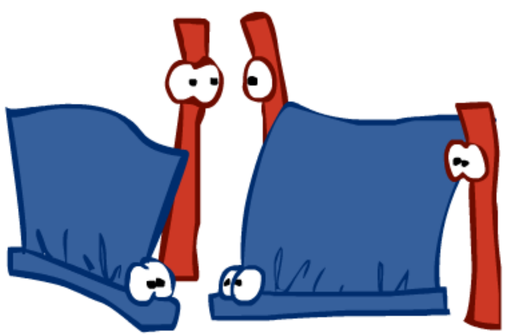
- Inference: calculating some useful quantity from a joint probability distribution

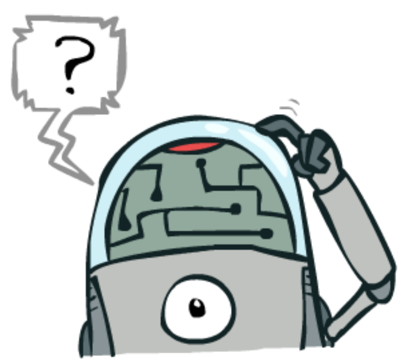
- Examples:
  - Posterior probability

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

  - Most likely explanation:

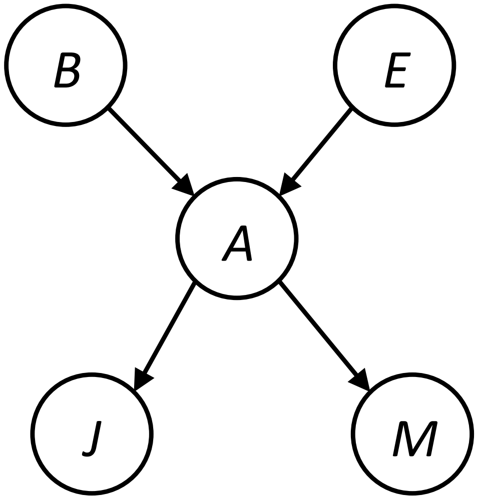
    $$\text{argmax}_q \; P(Q = q|E_1 = e_1 \ldots)$$





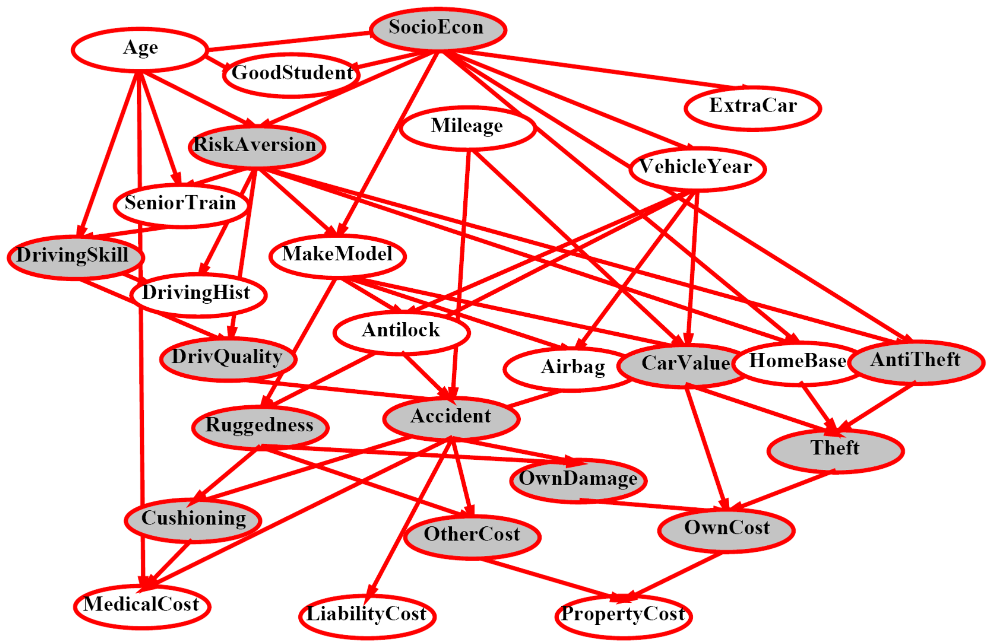
Inference by Enumeration

- Given unlimited time, inference in BNs is easy
- Recipe:
  - State the marginal probabilities you need
  - Figure out ALL atomic probabilities you need
  - Calculate and combine them
- Example: $P(+b|+j,+m) = \dfrac{P(+b,+j,+m)}{P(+j,+m)}$



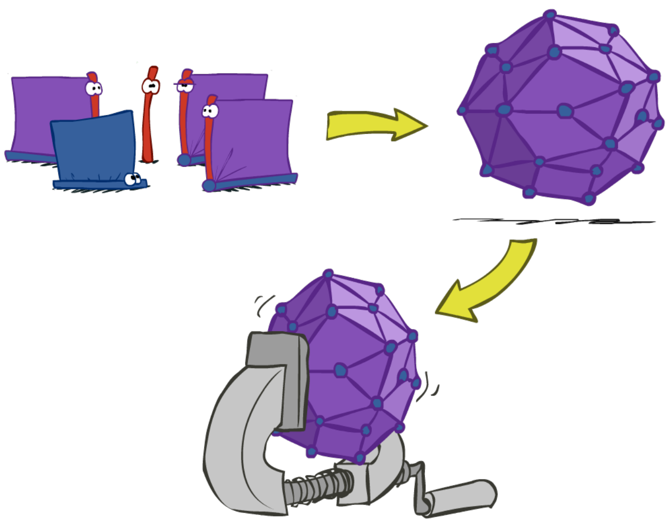
$$
\begin{aligned}
P(+b,+j,+m) = \quad & P(+b)P(+e)P(+a|+b,+e)P(+j|+a)P(+m|+a)+ \\
& P(+b)P(+e)P(-a|+b,+e)P(+j|-a)P(+m|-a)+ \\
& P(+b)P(-e)P(+a|+b,-e)P(+j|+a)P(+m|+a)+ \\
& P(+b)P(-e)P(-a|+b,-e)P(+j|-a)P(+m|-a)
\end{aligned}
$$

Inference by Enumeration?

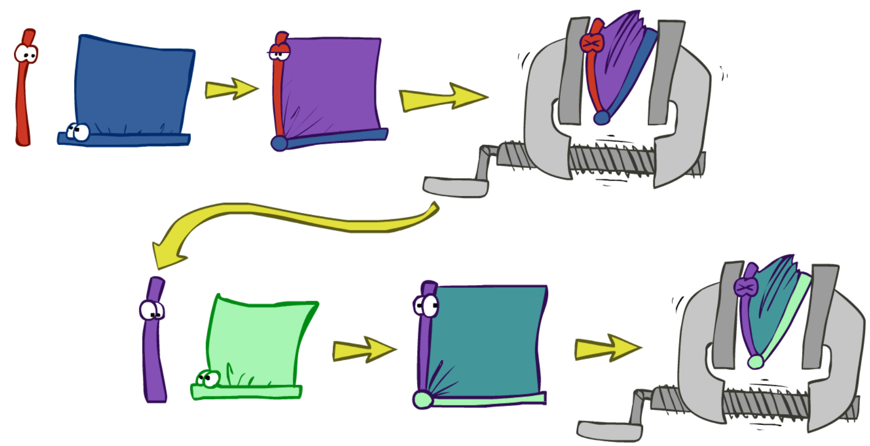


Inference by Enumeration vs. Variable Elimination

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

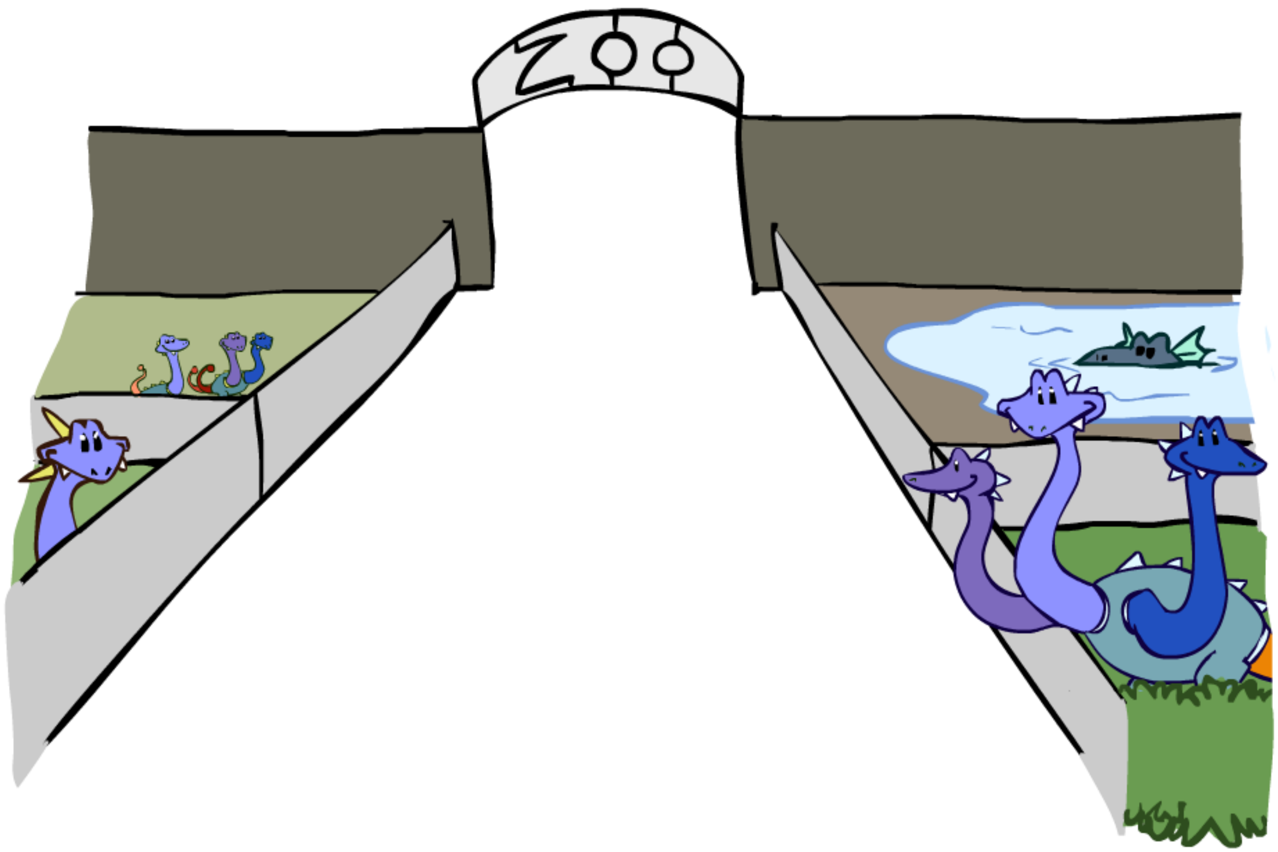


- Idea: interleave joining and marginalizing!
  - Called "Variable Elimination"
  - Still NP-hard, but usually much faster than inference by enumeration



  - First we'll need some new notation: factors

Factor Zoo



Factor Zoo I

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

$$P(T,W)$$

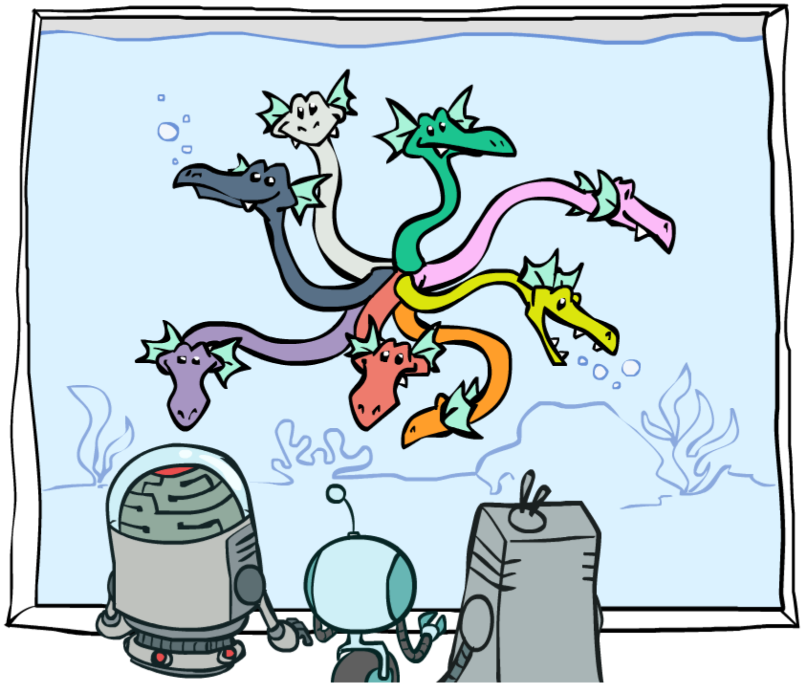
T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

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

$$P(cold,W)$$

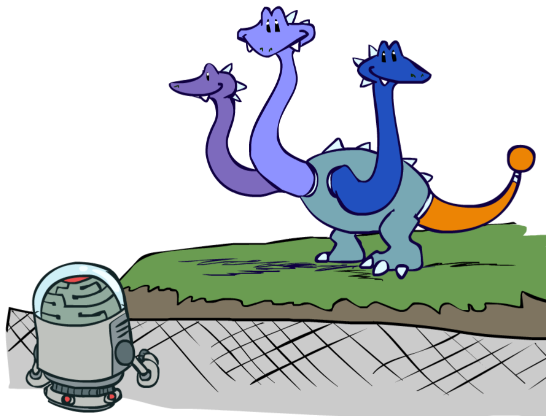
T	W	P
cold	sun	0.2
cold	rain	0.3

- Number of capitals = dimensionality of the table



Factor Zoo II

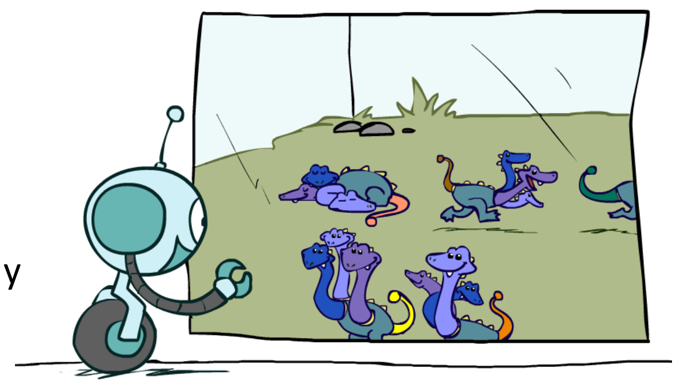
- Single conditional: P(Y | x)
  - Entries P(y | x) for fixed x, all
  - Sums to 1



$P(W|cold)$

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

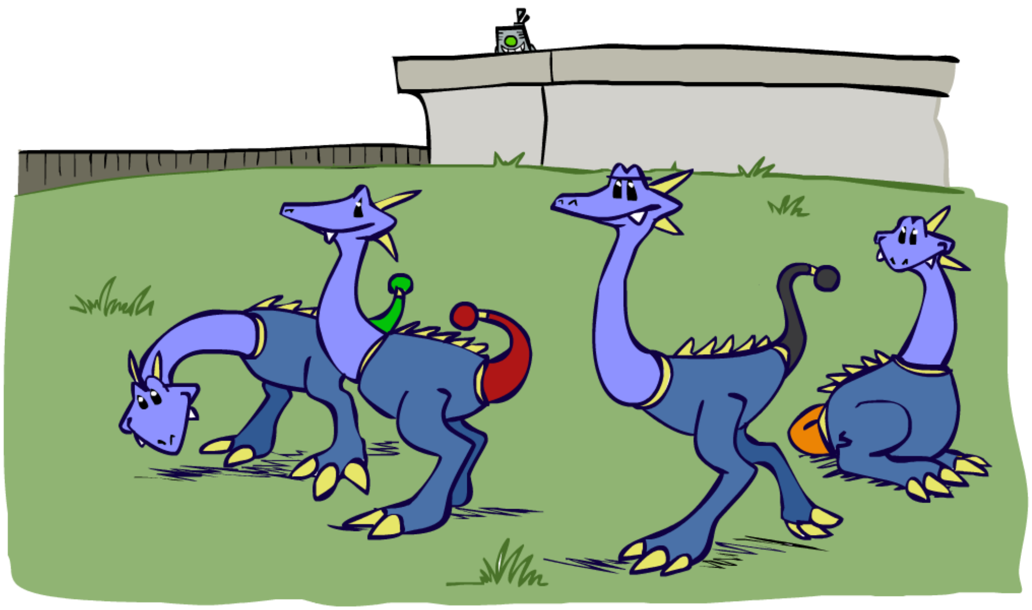
- Specified family: P(y | X)
  - Entries P(y | x) for fixed y, but for all x
  - Sums to … who knows!

$P(rain|T)$

T	W	P
hot	rain	0.2
cold	rain	0.6

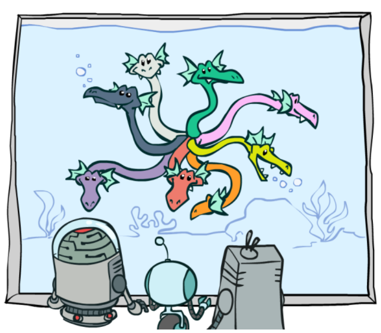
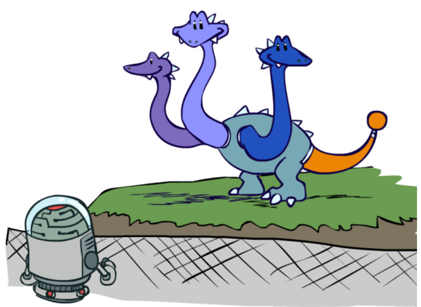
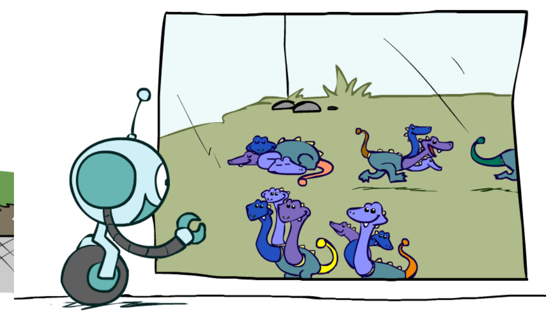
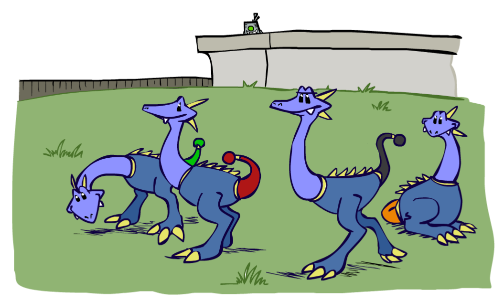
$P(rain|hot)$

$P(rain|cold)$



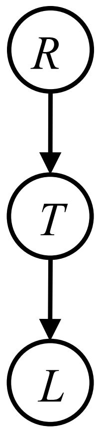
Factor Zoo Summary

- 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 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!



$P(R)$

+r	0.1
-r	0.9

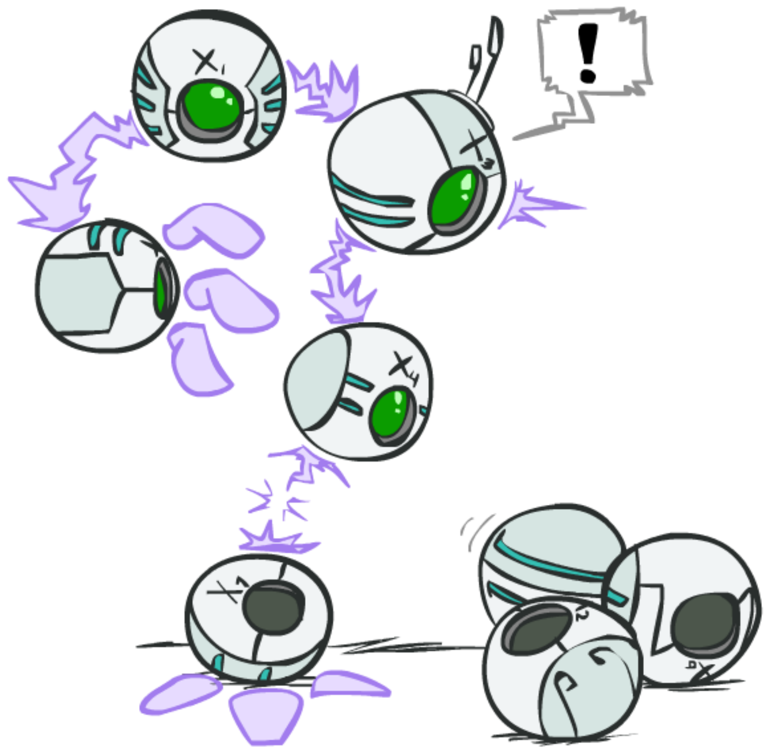
$P(T|R)$

+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

Variable Elimination (VE)



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

$P(R)$

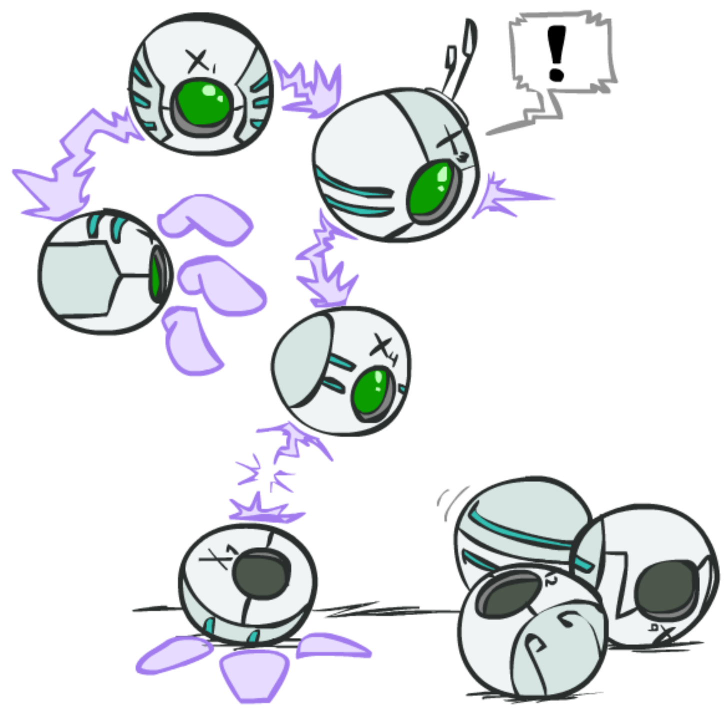
+r	0.1
-r	0.9

$P(T|R)$

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

$P(+\ell|T)$

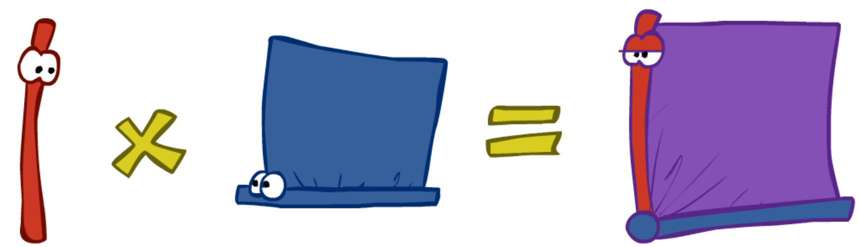
+t	+l	0.3
-t	+l	0.1



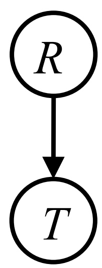
- VE: Alternately join factors and eliminate variables

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



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

$P(R)$

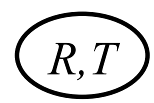
+r	0.1
-r	0.9

$P(T|R)$

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

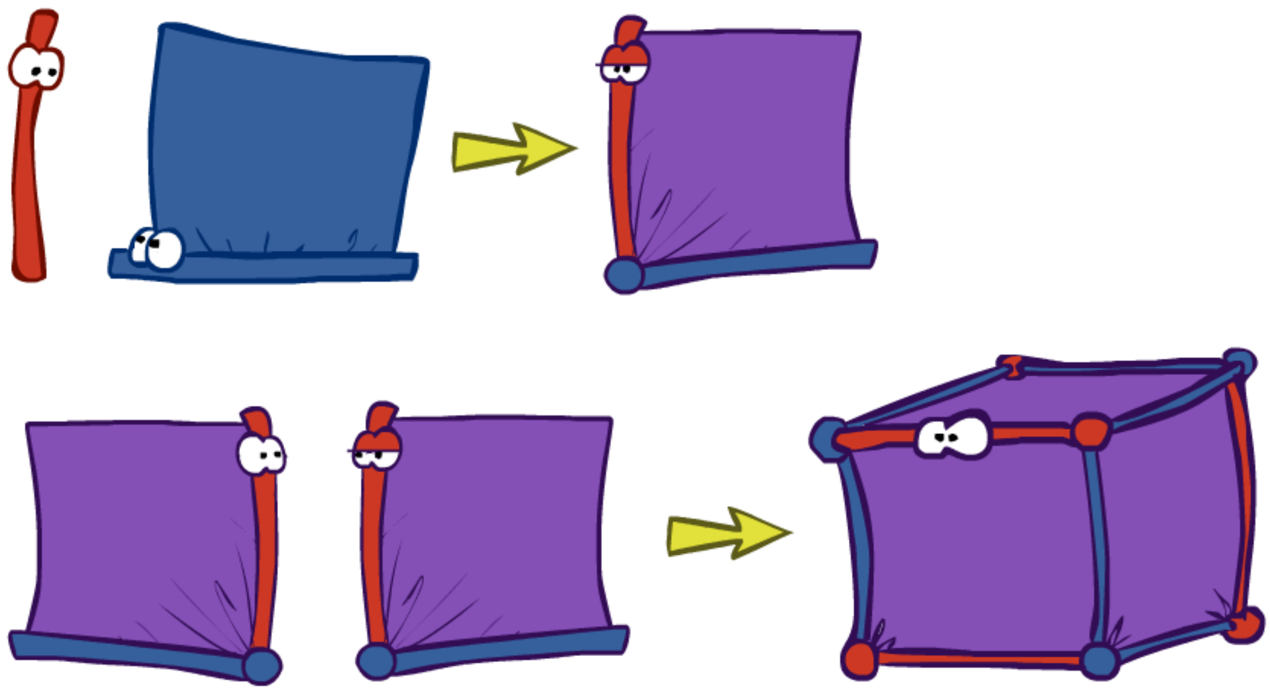
$P(R,T)$

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

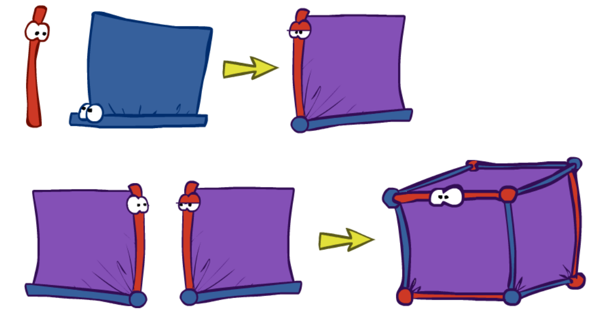


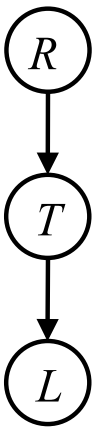
- Computation for each entry: pointwise products $\forall r, t: \quad P(r,t) = P(r) \cdot P(t|r)$

Example: Multiple Joins



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

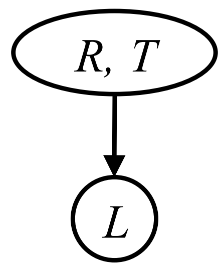
Join R

$P(R, T)$

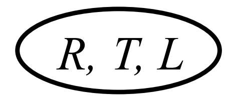
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

$P(L|T)$

+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
  - A projection operation
- Example:

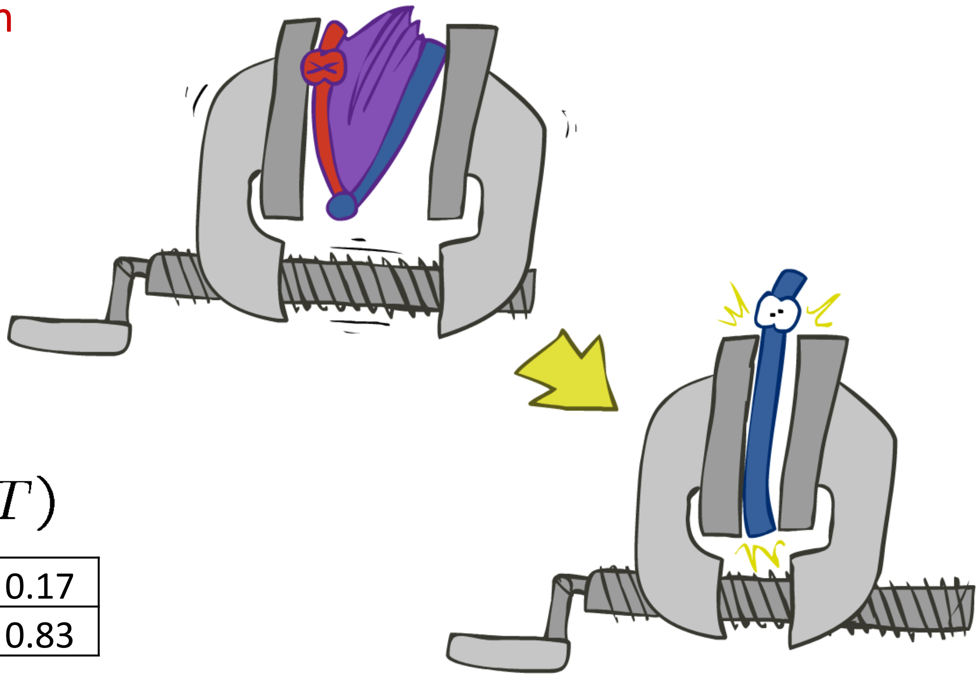
$$P(R,T)$$

+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

$P(R, T, L)$

+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 $\rightarrow$

T, L

$P(T, L)$

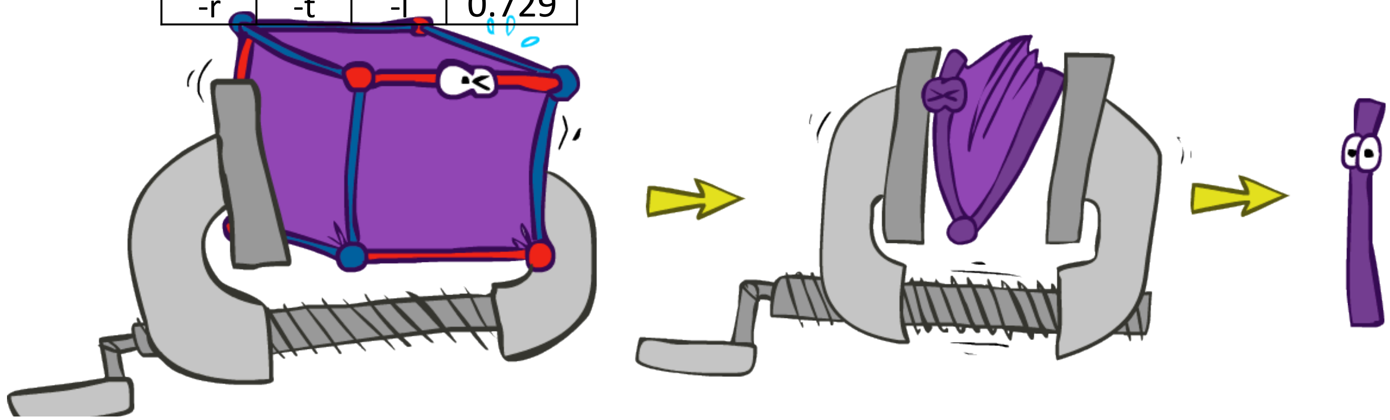
+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

Sum out T $\rightarrow$

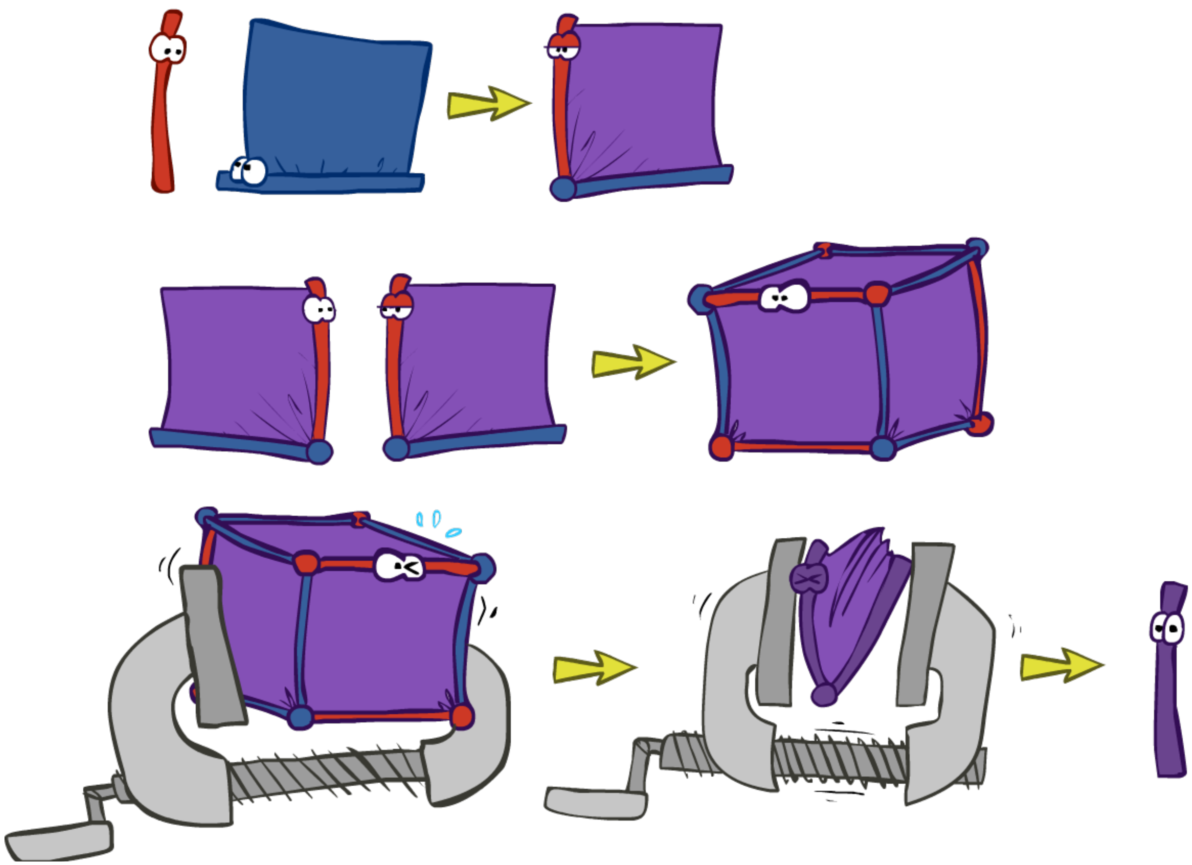
L

$P(L)$

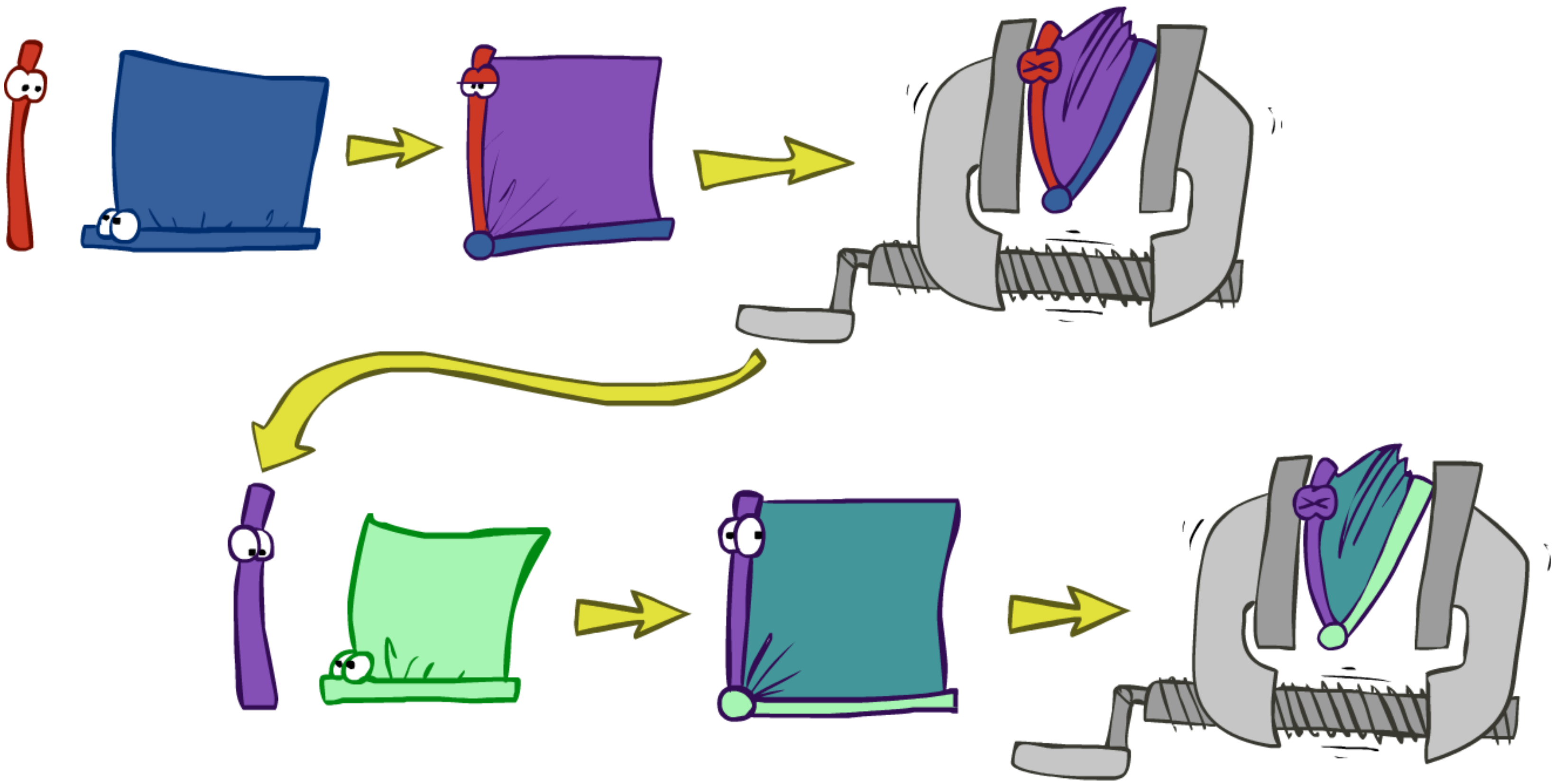
+l	0.134
-l	0.886



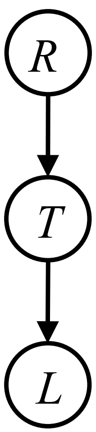
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)



Marginalizing Early (= Variable Elimination)



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

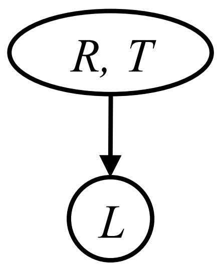
$P(L|T)$

+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



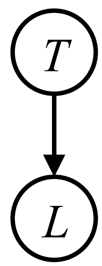
$P(L|T)$

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

Sum out R

$P(T)$

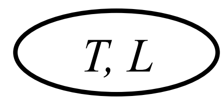
+t	0.17
-t	0.83



$P(L|T)$

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

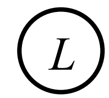
Join T



$P(T,L)$

+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

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:

$P(+r)$

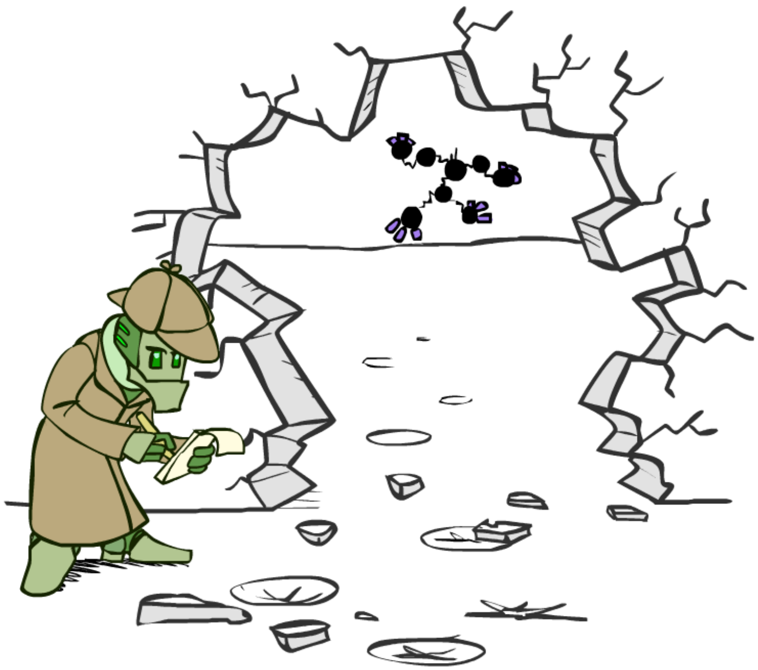
+r	0.1

$P(T|+r)$

+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:

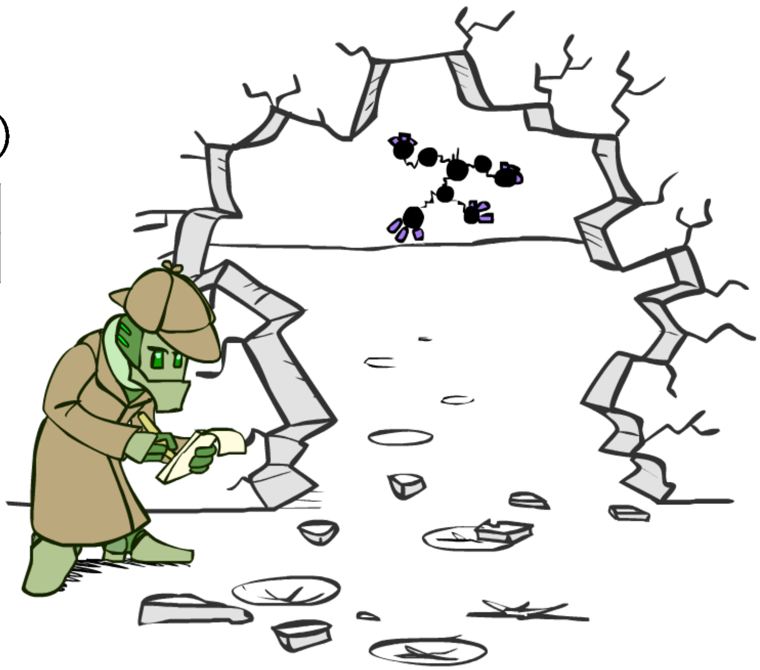
$P(+r, L)$

+r	+l	0.026
+r	-l	0.074

Normalize

$P(L | +r)$

+l	0.26
-l	0.74

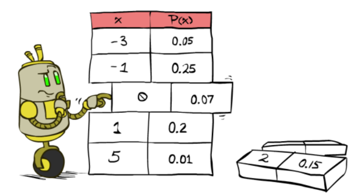


- To get our answer, just normalize this!

- That's it!

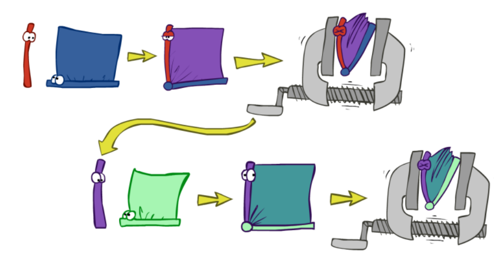
General Variable Elimination

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



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

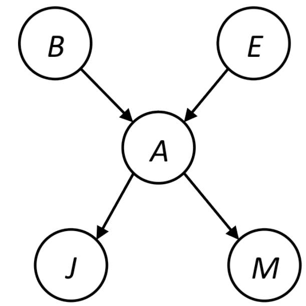
- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H



- Join all remaining factors and normalize

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

Example



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

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

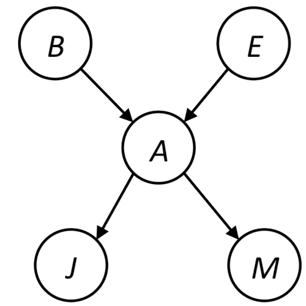
Choose A

$$\begin{array}{l} P(A|B,E) \\ P(j|A) \\ P(m|A) \end{array} \xrightarrow{\times} P(j,m,A|B,E) \xrightarrow{\sum} P(j,m|B,E)$$

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

Example

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



Choose E

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

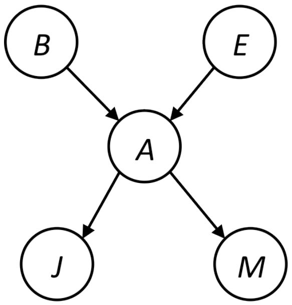
$$\boxed{P(B) \qquad P(j,m|B)}$$

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

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



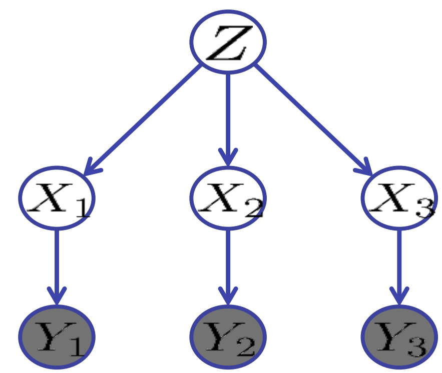
$P(B)$	$P(E)$	$P(A\|B,E)$	$P(j\|A)$	$P(m\|A)$

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

All we are doing is exploiting uwy + uwz + uxy + uxz + vwy + vwz + vxy +vxz = (u+v)(w+x)(y+z) to improve computational efficiency!

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:

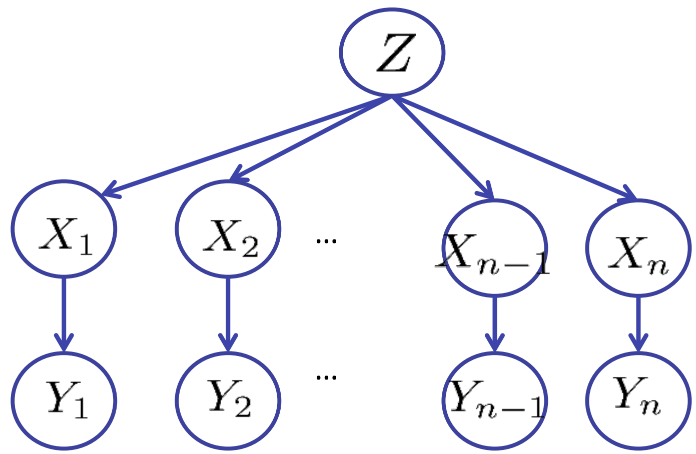
$$f_4(y_1, y_2, y_3, X_3) = P(y_3|X_3)f_3(y_1, y_2, X_3).$$

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

Computational complexity critically depends on the largest factor being generated in this process. Size of factor = number of entries in table. In example above (assuming binary) all factors generated are of size 2 --- as they all only have one variable (Z, Z, and X_3 respectively).

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: $Z, X_1, \ldots, X_{n-1}$ and $X_1, \ldots, X_{n-1}, Z$. 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

- The computational and space complexity of variable elimination is determined by the largest factor
- The elimination ordering can greatly affect the size of the largest factor.
  - E.g., previous slide's example 2^n vs. 2
- Does there always exist an ordering that only results in small factors?
  - No!

Worst Case Complexity?

- CSP:

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

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

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

$$\ldots$$

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

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

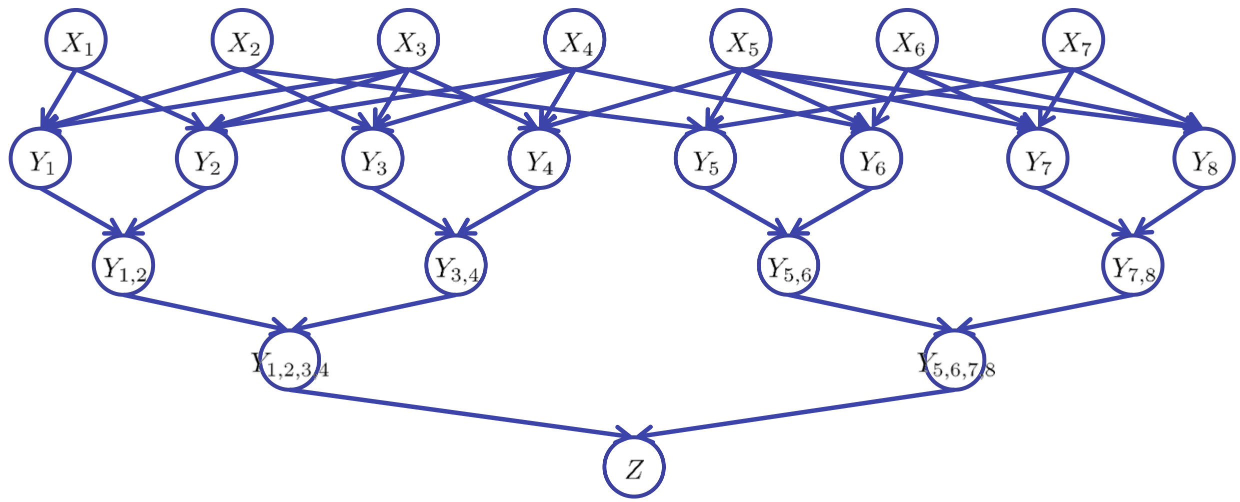
$$\ldots$$

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

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

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

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



- 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
- 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!

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