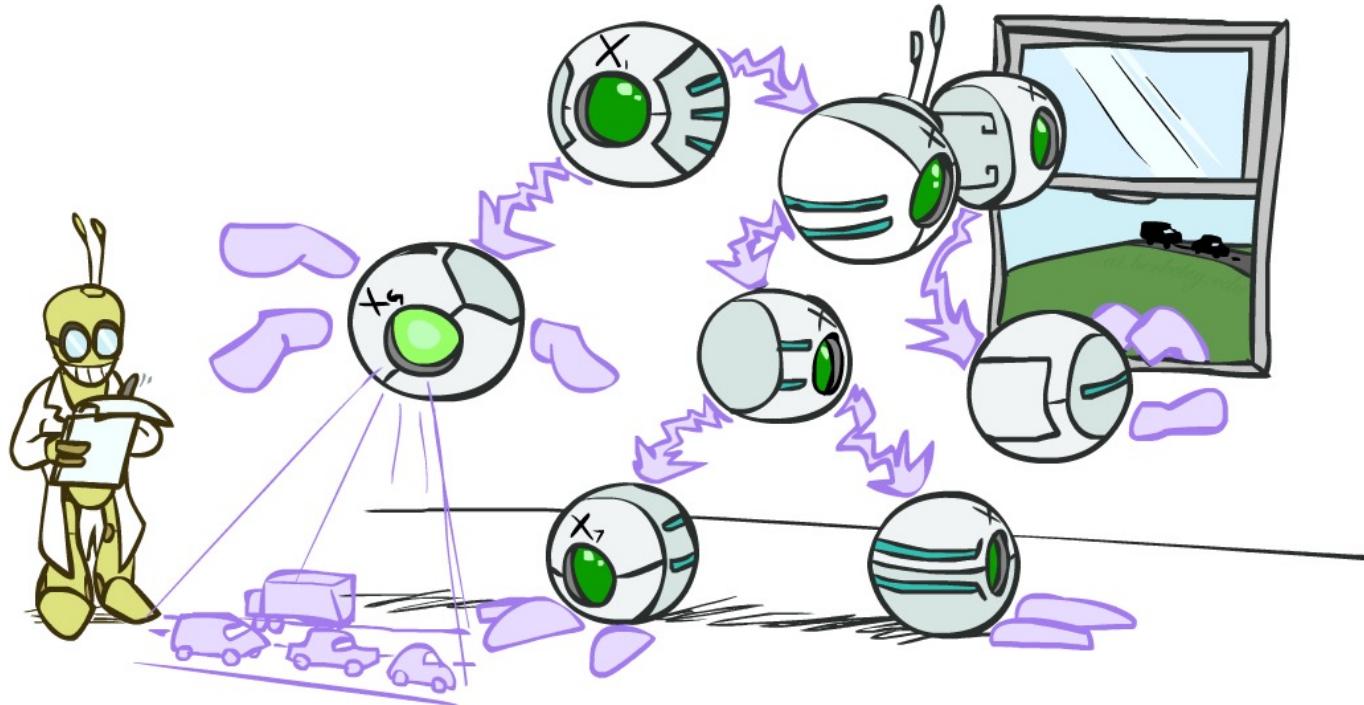


CS 3317: Artificial Intelligence

Bayes Nets: Inference



Instructors: Panpan Cai

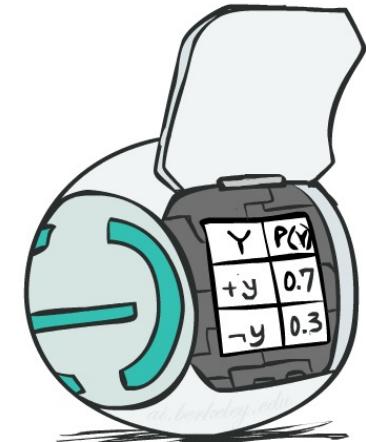
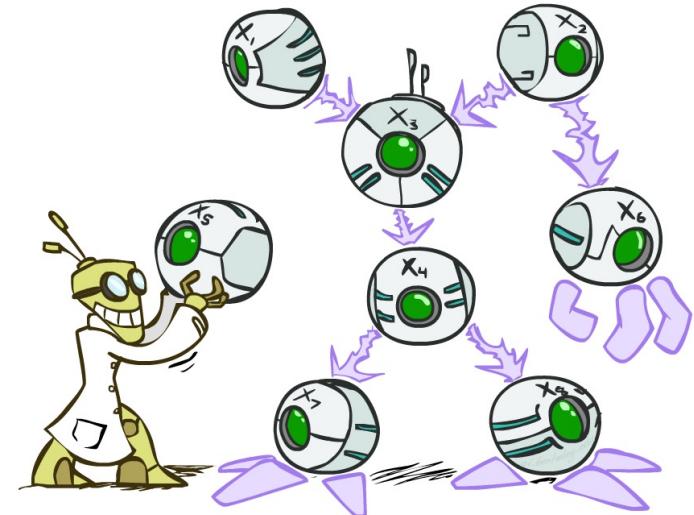
[Slides adapted from UC Berkeley CS188]



Bayes Net Representation

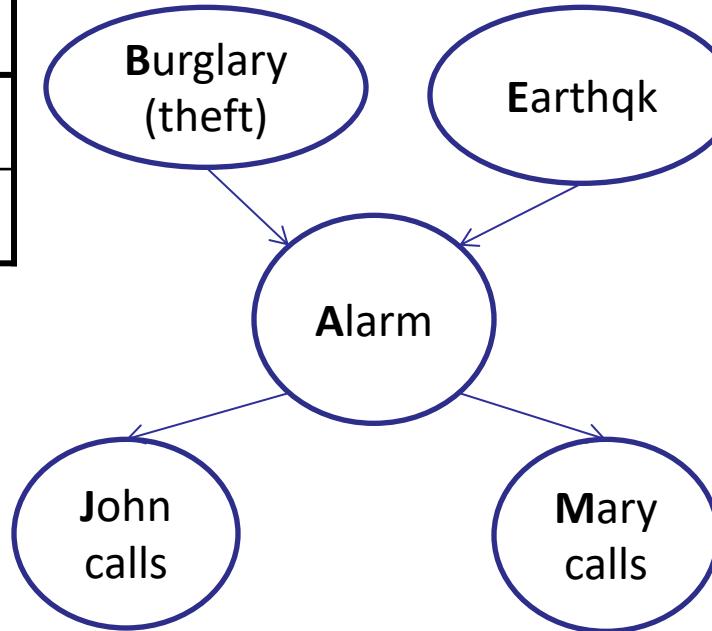
- A directed, acyclic graph, with node = random variable
- Each node stores a conditional probability table (CPT)
 - A collection of conditional distributions over X , one for each combination of parents' values
$$P(X|a_1 \dots a_n)$$
- Bayes nets implicitly encode joint distributions
 - As a product of local conditional distributions
 - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

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



Example: Alarm Network

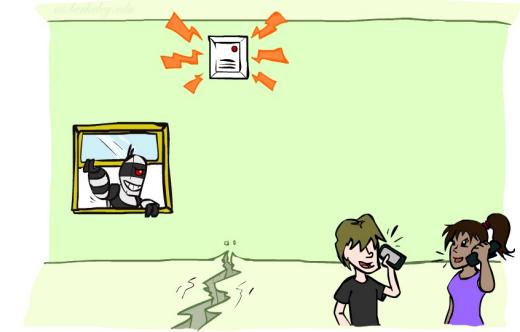
B	P(B)
+b	0.001
-b	0.999



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

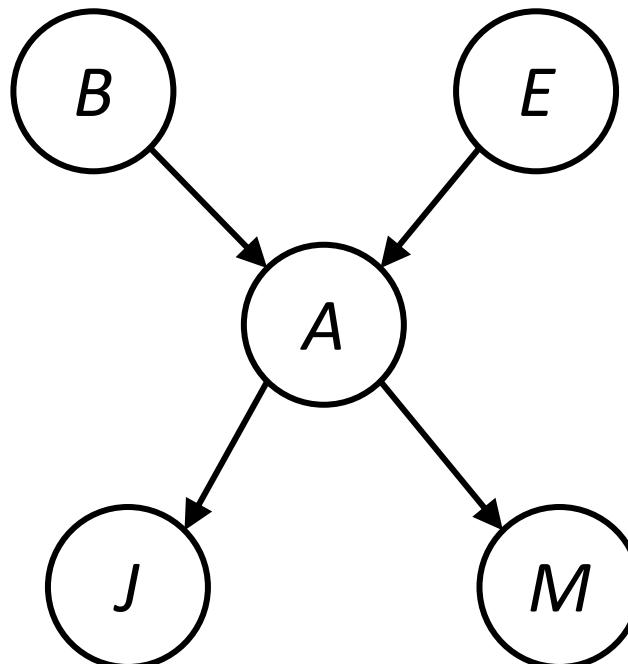
E	P(E)
+e	0.002
-e	0.998



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

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

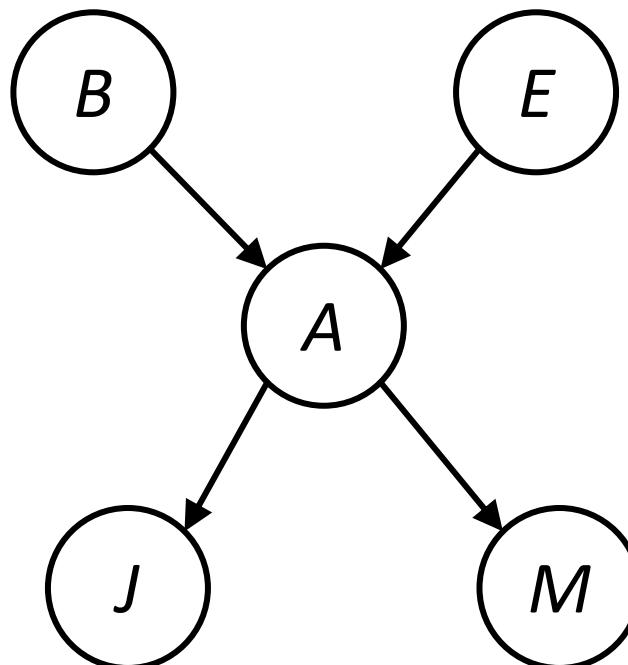
$$P(+b, -e, +a, -j, +m) =$$



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

Example: Alarm Network

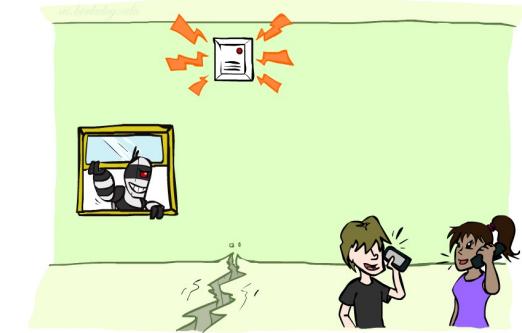
B	P(B)
+b	0.001
-b	0.999



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



$$\begin{aligned}
 P(+b, -e, +a, -j, +m) &= \\
 P(+b)P(-e)P(+a|+b, -e)P(-j|+a)P(+m|+a) &= \\
 = 0.001 \times 0.998 \times 0.94 \times 0.1 \times 0.7
 \end{aligned}$$

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

Bayes Nets



Representation



Probabilistic Inference

- Enumeration (exact, exponential complexity)

- Variable elimination (exact, worst-case

- exponential complexity, often better)

- Probabilistic inference is NP-complete

- Conditional Independences

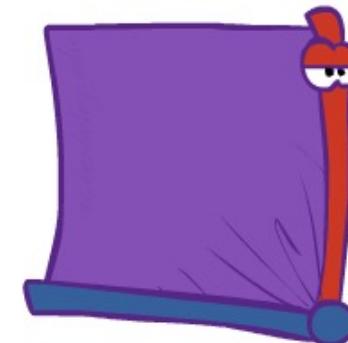
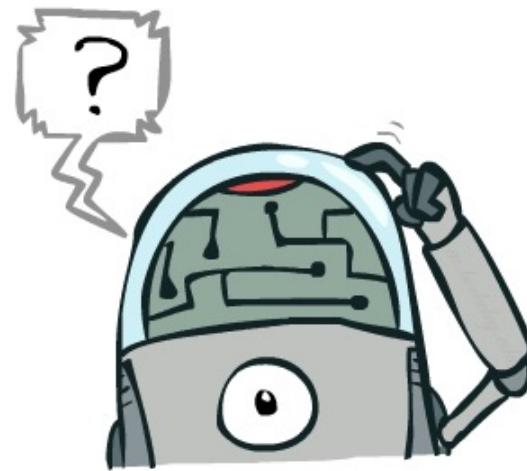
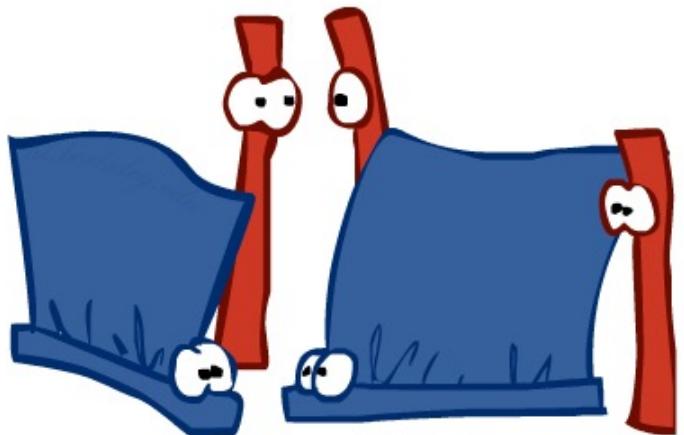
- Sampling

- Learning from data

Inference

- Inference: calculating some useful quantity from a joint probability distribution
- $P(\text{high score} \mid \text{attendance, homework, review})$
- $P(\text{cold} \mid \text{cough, fever, sore throat})$
vs.
- $P(\text{COVID} \mid \text{cough, fever, sore throat})$

- Examples:
 - Posterior distribution
$$P(Q \mid E_1 = e_1, \dots, E_k = e_k)$$
 - Most likely explanation:
$$\operatorname{argmax}_q P(Q = q \mid E_1 = e_1, \dots)$$



Recap: Inference by Enumeration

- General case:

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

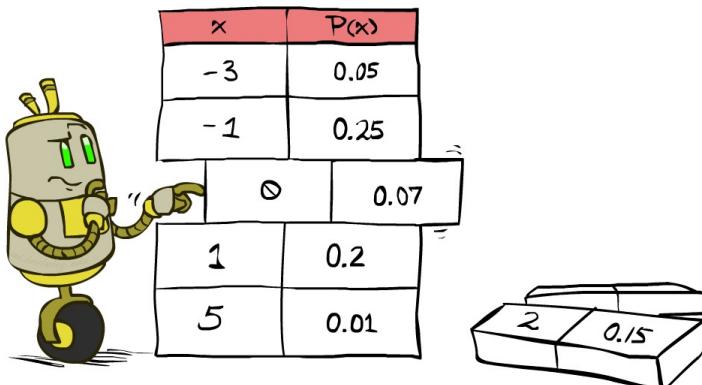
X_1, X_2, \dots, X_n
All variables

- We want:

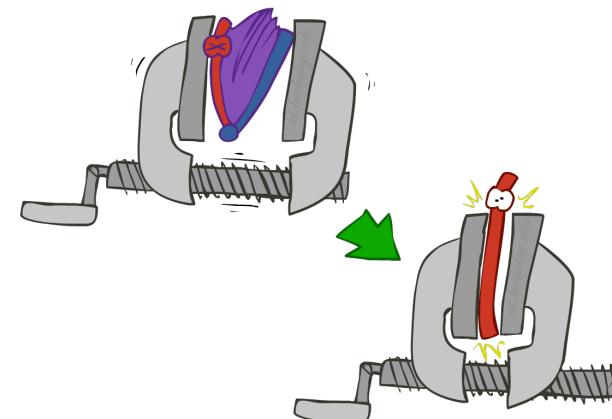
* Works fine with
multiple query
variables, too

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

- Step 1: Select the entries consistent with the evidence



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



- Step 3: Normalize

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

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

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

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

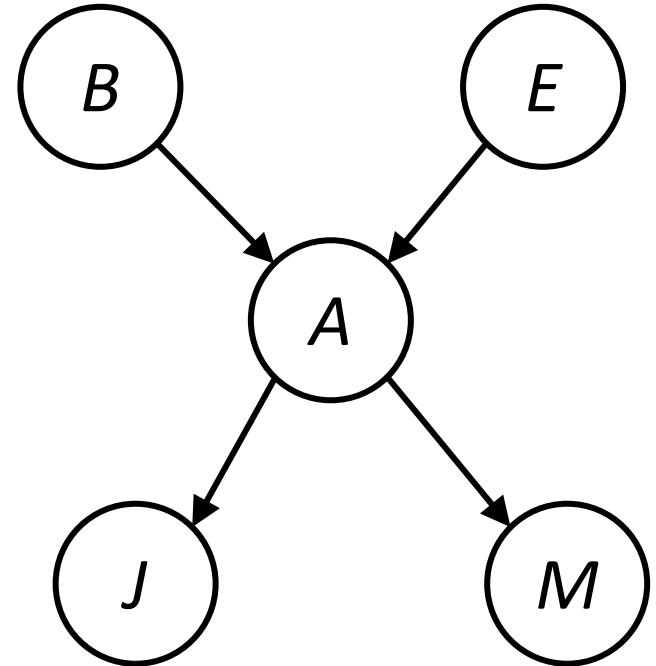
Inference by Enumeration in Bayes Nets

- Given unlimited time, inference in BNs is easy

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

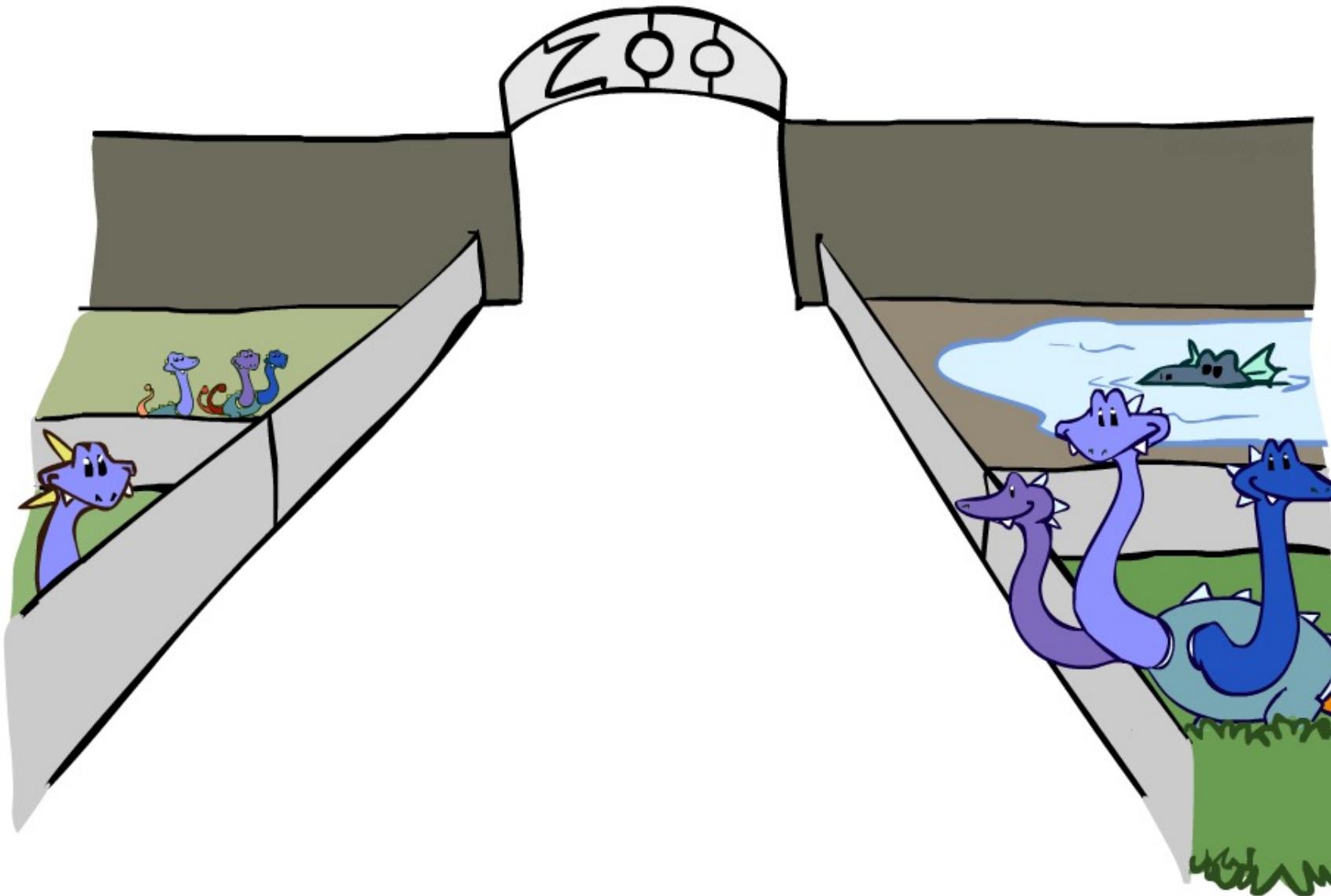
$$= \sum_{e,a} P(B,e,a,+j,+m)$$

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



- Intermediate results are not normal distributions, simply call them *factors*.

Factor Zoo



Factor Zoo I

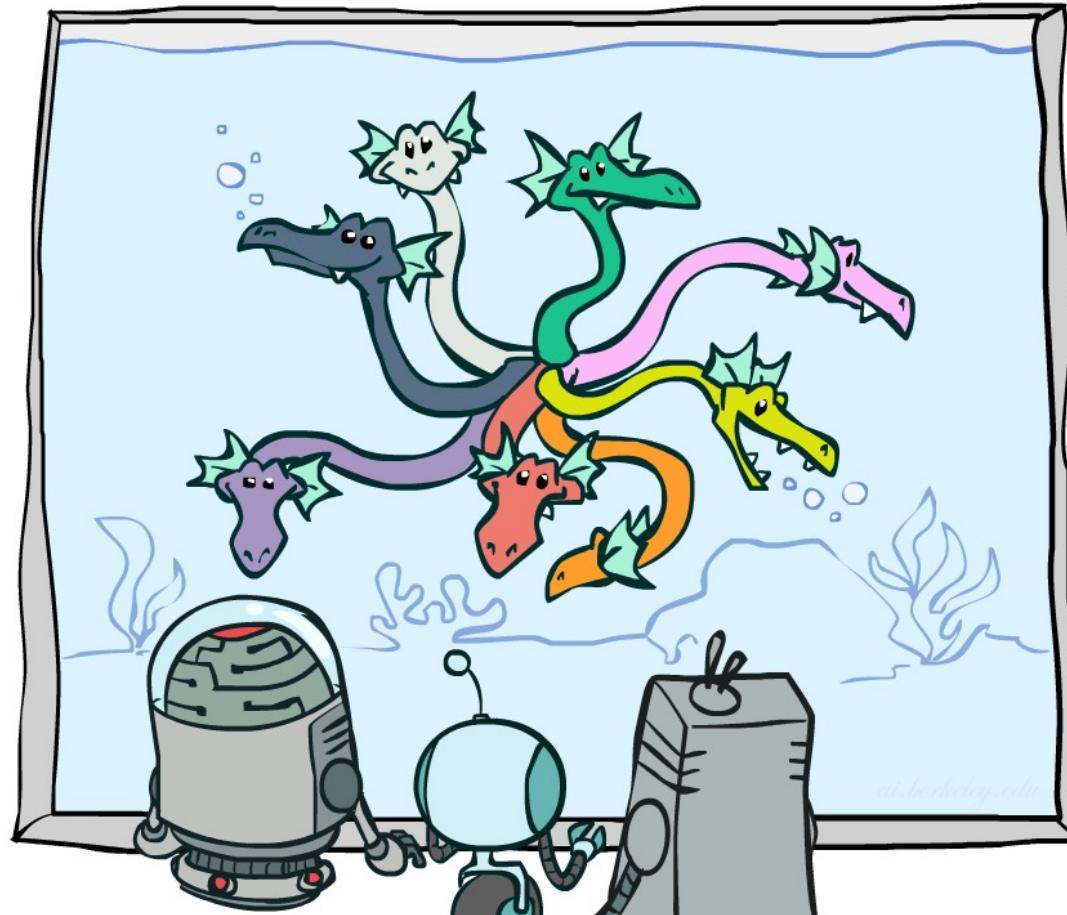
- Joint distribution: $P(X, Y)$
 - Entries $P(x,y)$ for all x, y
 - Sums to 1
- Selected joint: $P(x, Y)$
 - A slice of the joint distribution
 - Entries $P(x,y)$ for fixed x , all y
 - Sums to $P(x)$
- Number of capitals = dimensionality of the table

$P(T, W)$

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

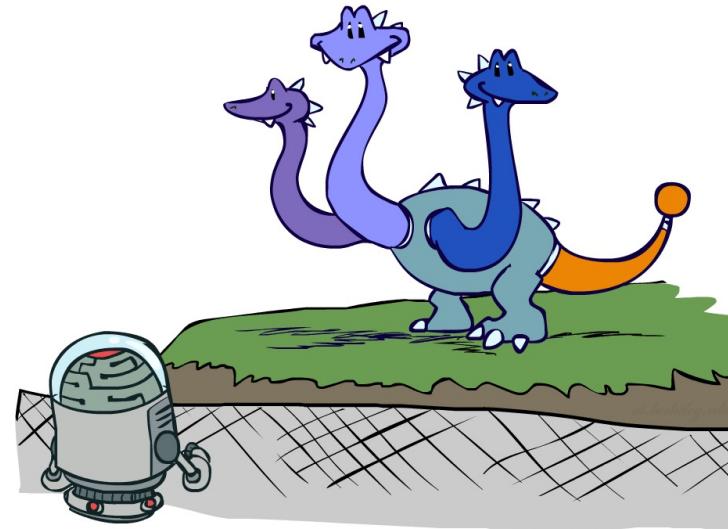
$P(\text{cold}, W)$

T	W	P
cold	sun	0.2
cold	rain	0.3



Factor Zoo II

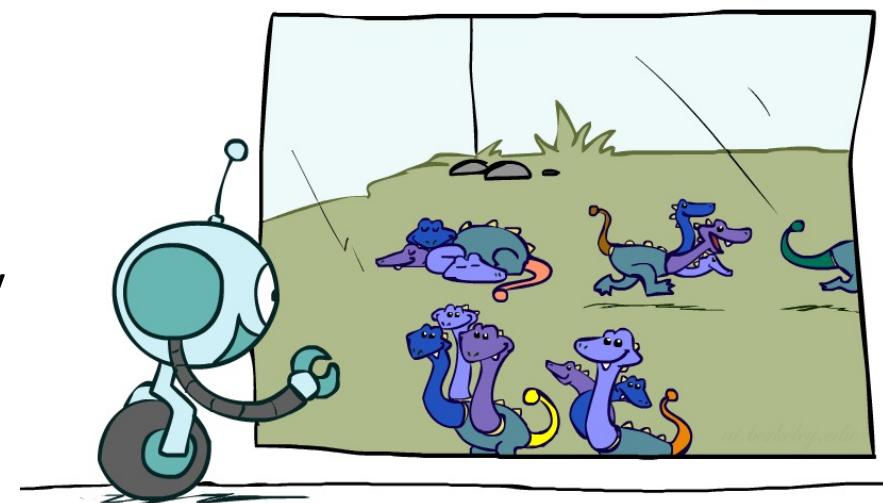
- Single conditional: $P(Y | x)$
 - Entries $P(y | x)$ for fixed x , all y
 - Sums to 1



- Family of conditionals:

$P(Y | X)$

- Multiple conditionals
- Entries $P(y | x)$ for all x, y
- Sums to $|X|$



$P(W | cold)$

T	W	P
cold	sun	0.4
cold	rain	0.6

$P(W | T)$

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

$P(W | hot)$

$P(W | cold)$

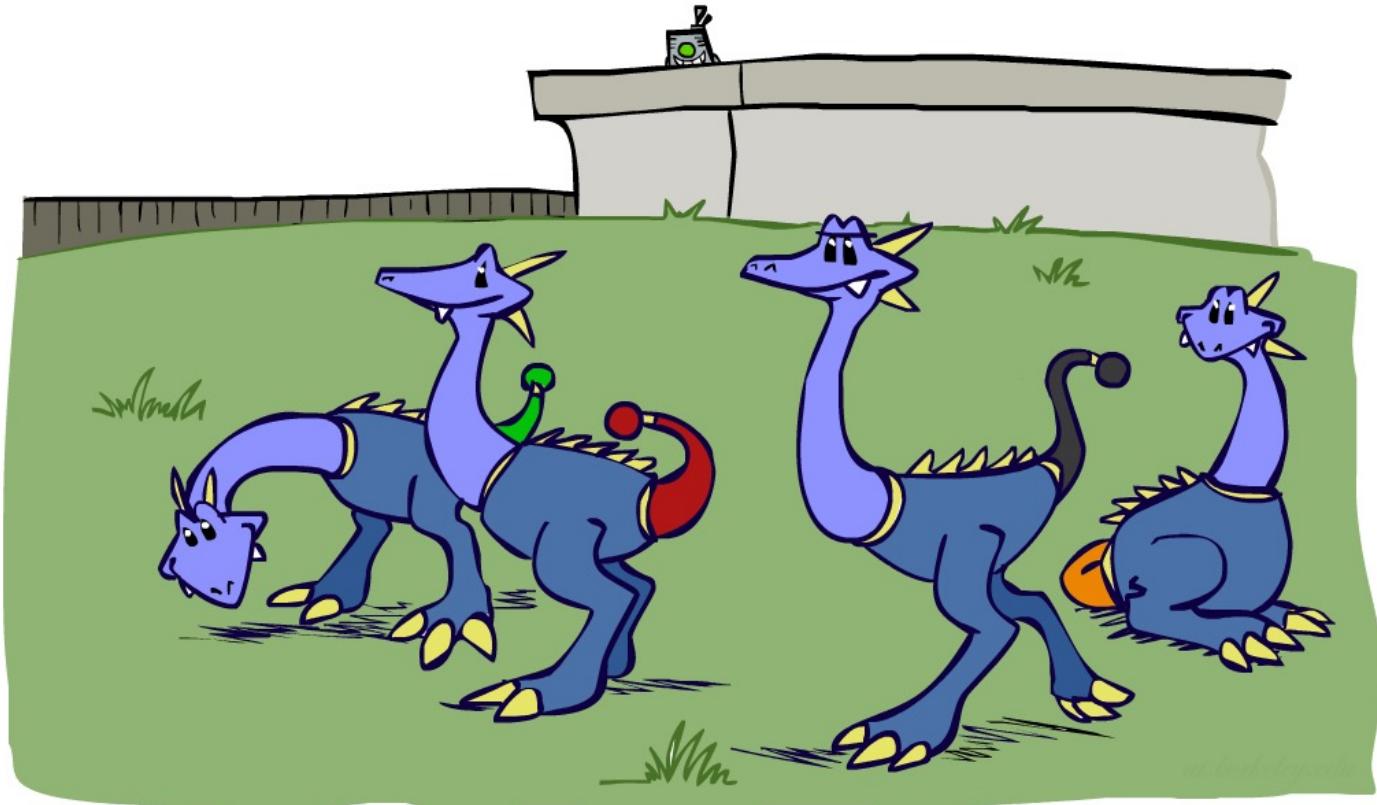
Factor Zoo III

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

$P(rain|T)$

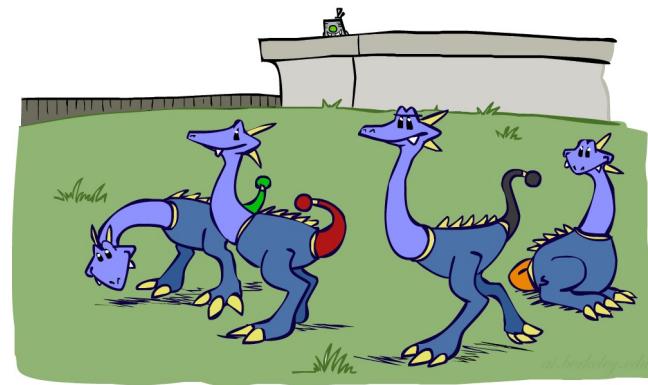
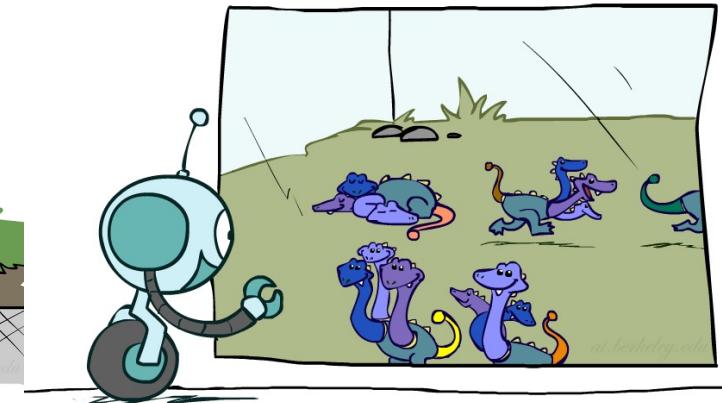
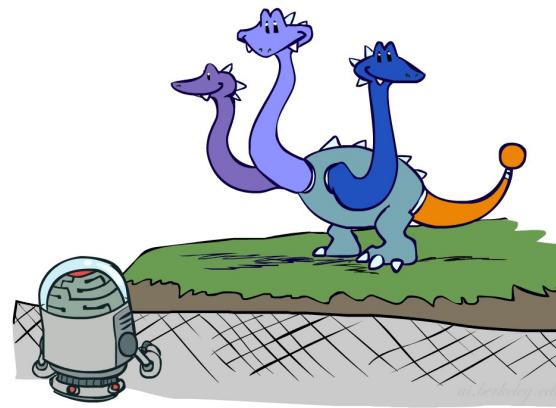
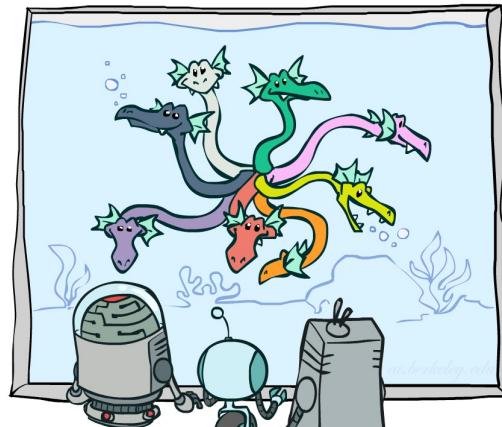
T	W	P
hot	rain	0.2
cold	rain	0.6

$$\left. \begin{array}{l} P(rain|hot) \\ P(rain|cold) \end{array} \right\}$$



Factor Zoo Summary

- In general, when we write $P(Y_1 \dots Y_N | X_1 \dots X_M)$:
 - It is a “factor,” a multi-dimensional array / table
 - Each entry has a probability value $P(y_1 \dots y_N | x_1 \dots x_M)$
 - Any lower-case X or Y is a dimension missing from the array (value fixed)
 - Any capital X or Y is an active dimension (enumerating all values)



Example: Traffic Domain

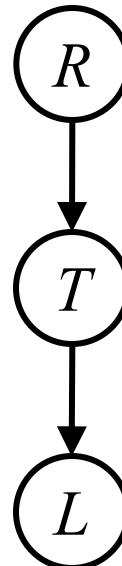
- Random Variables

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

$$P(L) = ?$$

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

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



$$P(R)$$

+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: Assembly line on Factors

- 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

- Instantiate by known values

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

$$P(R)$$

+r	0.1
-r	0.9

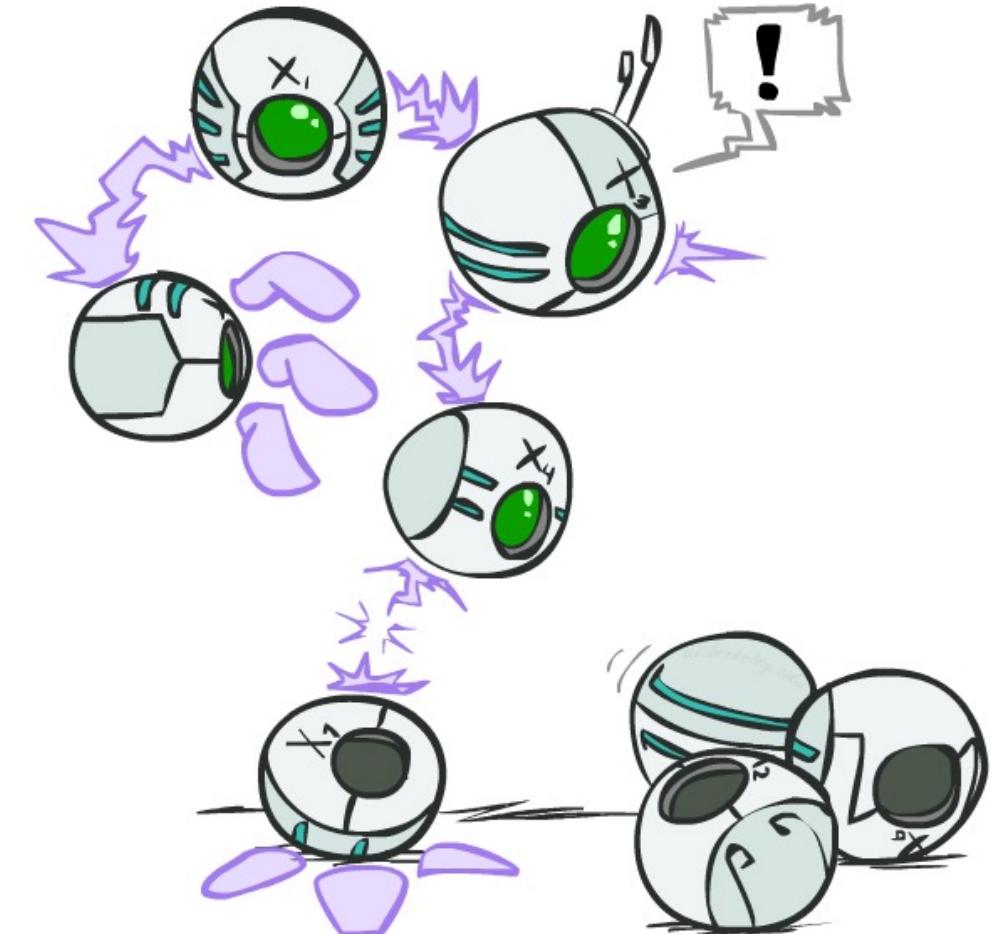
$$P(T|R)$$

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

$$P(+\ell|T)$$

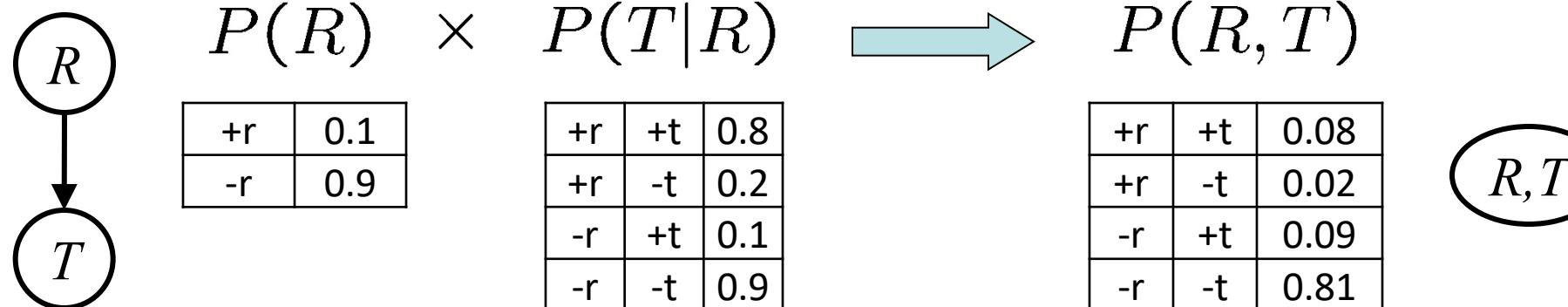
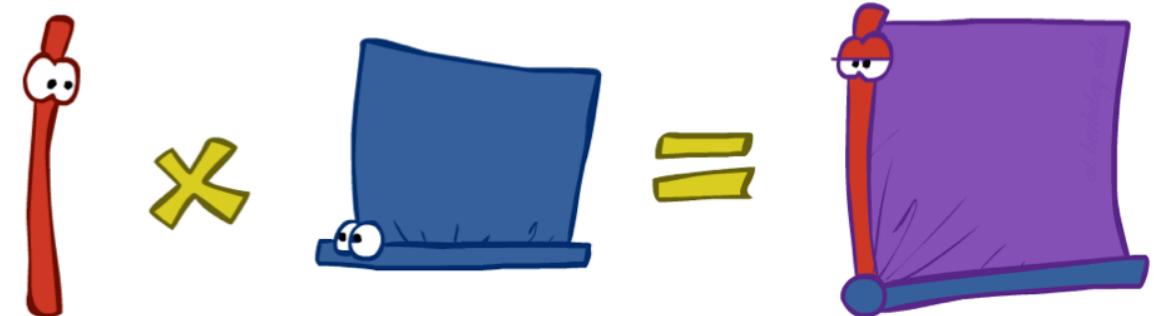
+t	+l	0.3
-t	+l	0.1

- Procedure: Join all factors, then sum out all hidden variables



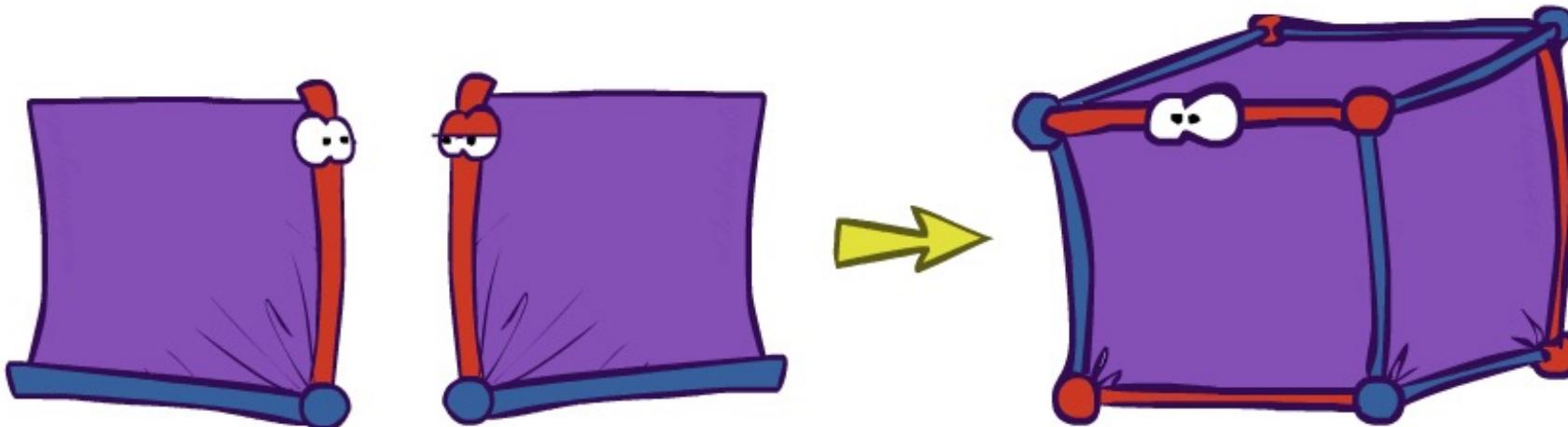
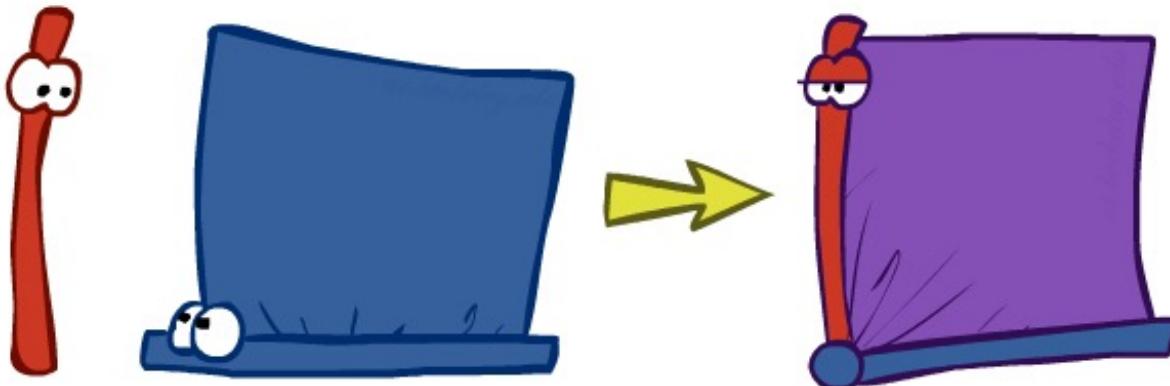
Operation 1: Join Factors

- First basic operation: **joining factors**
 - Get *all* factors involving the joining variable
 - Build a new factor over the *union* of variables involved
 - like a database join, not matrix multiplication
- Example: Join *on R*

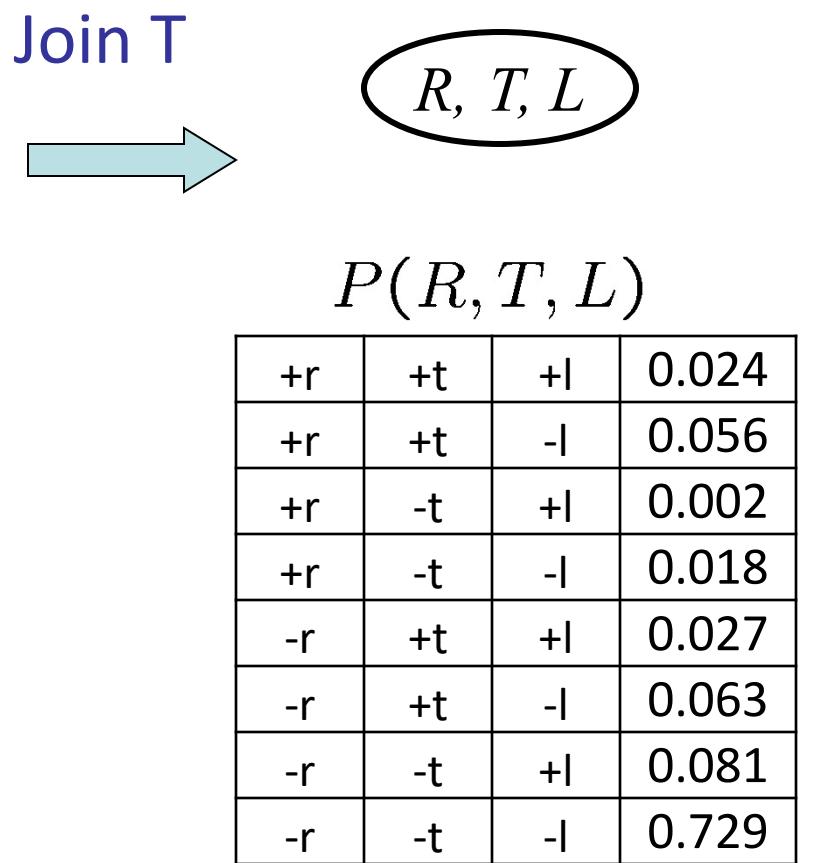
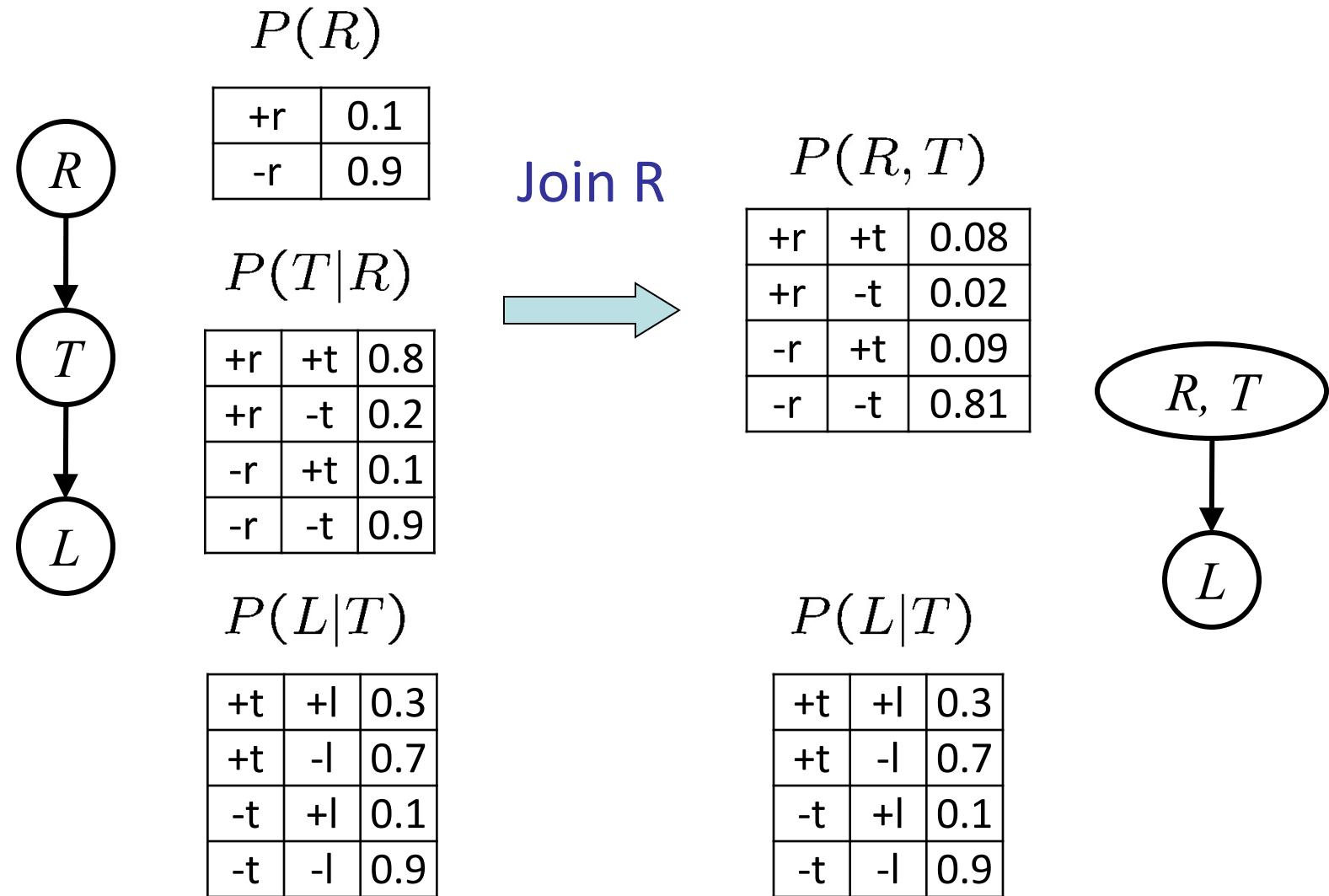
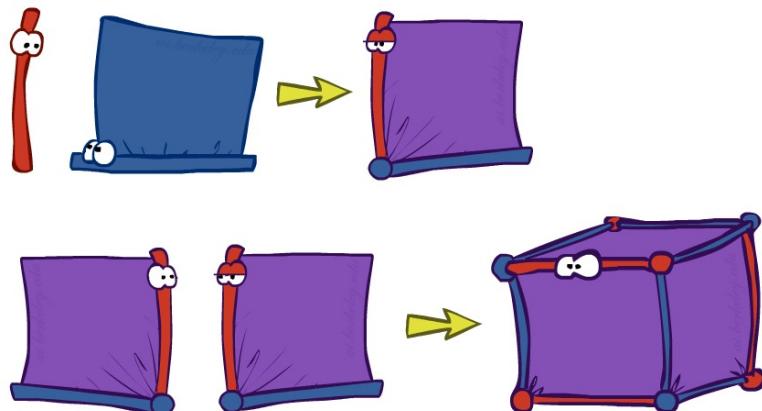


- Computation for each entry: point-wise products $\forall r, t : P(r, t) = P(r) \cdot P(t|r)$

Example: Multiple Joins



Example: Multiple Joins



Operation 2: Eliminate

- Second basic operation: **marginalization**
- Take **one** factor and sum out a hidden variable
 - Shrinks the factor to a lower-dim one
- Example:

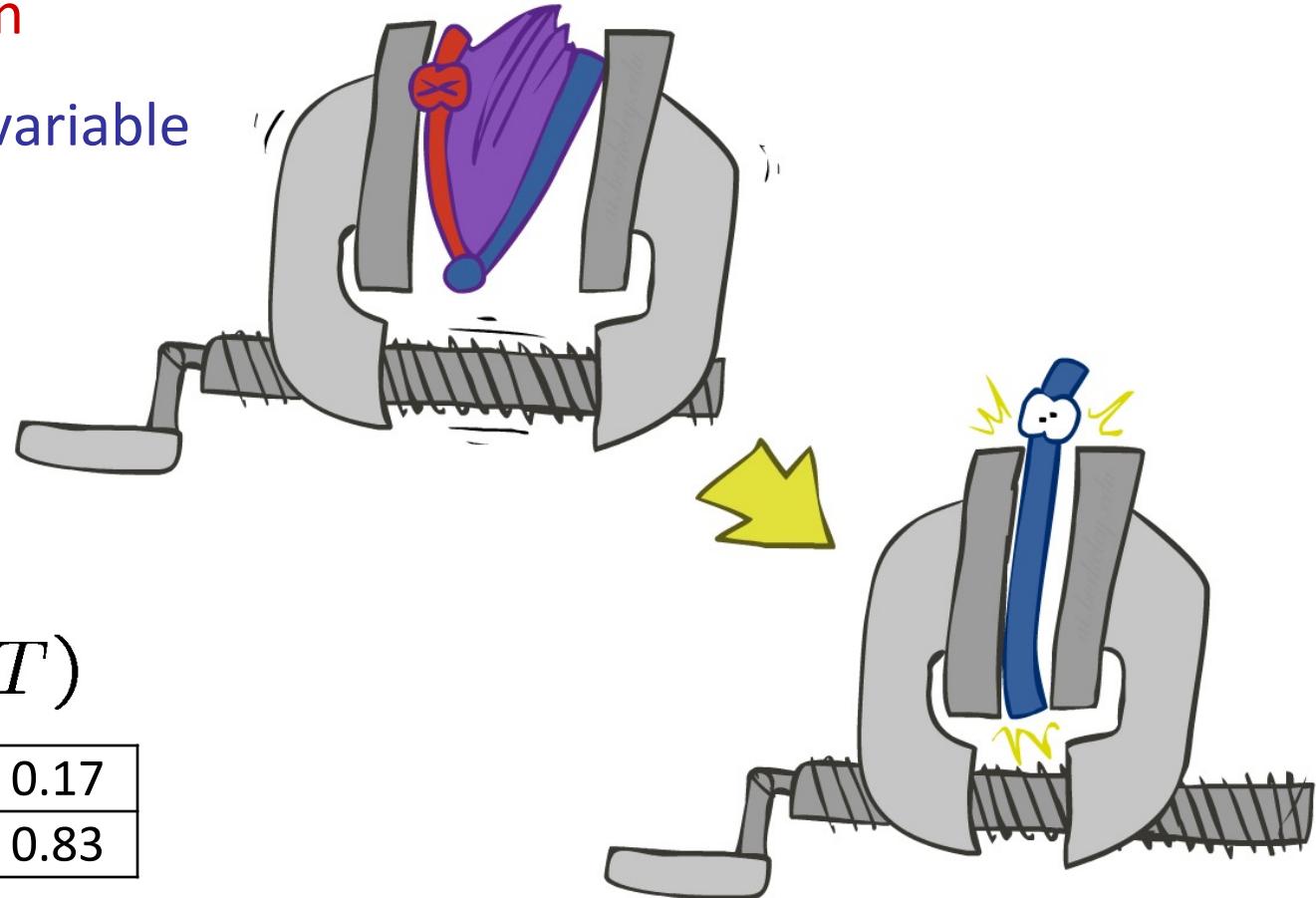
$P(R, T)$		
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

sum R



$P(T)$

+t	0.17
-t	0.83



Multiple Elimination

$P(R, T, L)$

$+r$	$+t$	$+l$	$P(R, T, L)$
$+r$	$+t$	$-l$	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

R, T, L

Sum
out R

T, L

Sum
out T

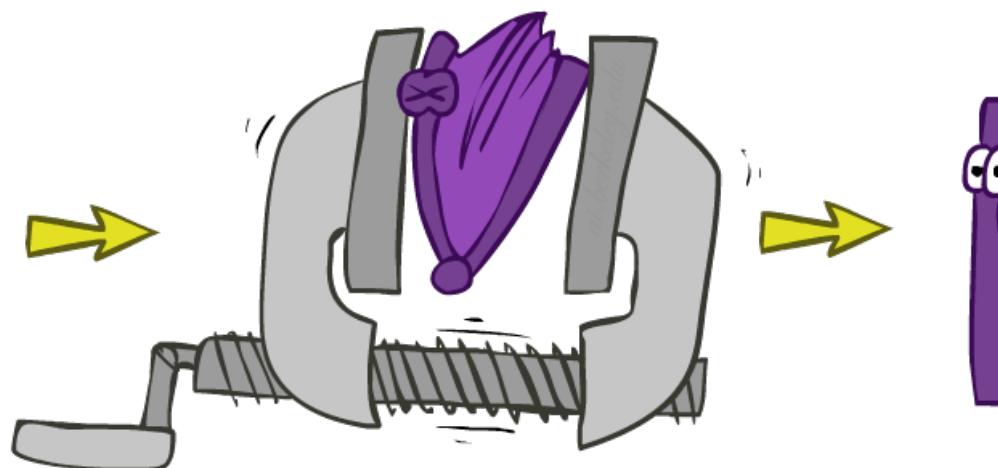
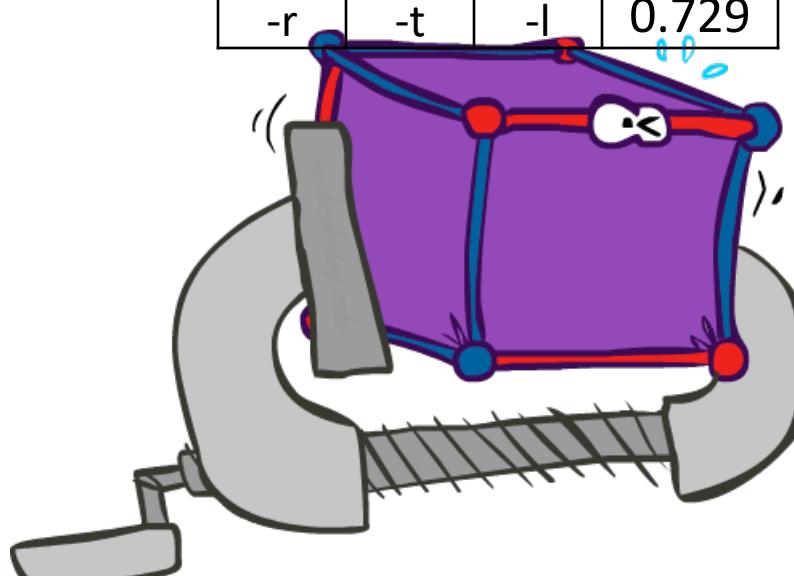
L

$P(T, L)$

$+t$	$+l$	$P(T, L)$
$+t$	$+l$	0.051
$+t$	$-l$	0.119
$-t$	$+l$	0.083
$-t$	$-l$	0.747

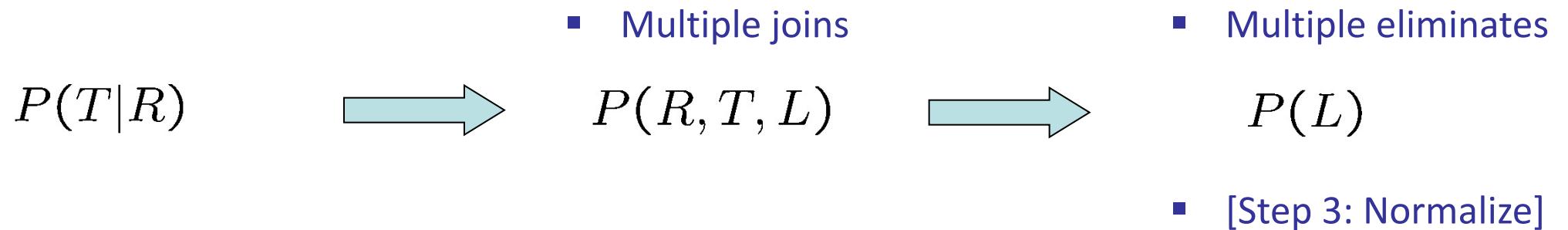
$P(L)$

$+l$	0.134
$-l$	0.866



Multiple Join, Multiple Eliminate (= Inference by Enumeration)

$$P(R)$$



$$P(L|T)$$

Reference: Inference by Enumeration

- General case:

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

$$\left. \begin{array}{l} E_1 \dots E_k = e_1 \dots e_k \\ Q \\ H_1 \dots H_r \end{array} \right\} X_1, X_2, \dots, X_n$$

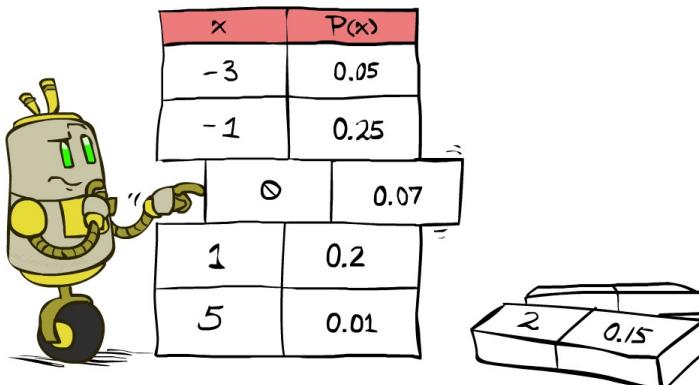
All variables

- We want:

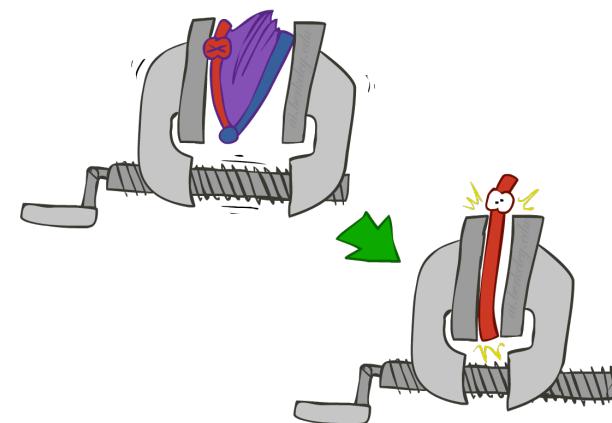
$$P(Q|e_1 \dots e_k)$$

* Works fine with multiple query variables, too

- Step 1: Select the entries consistent with the evidence



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



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

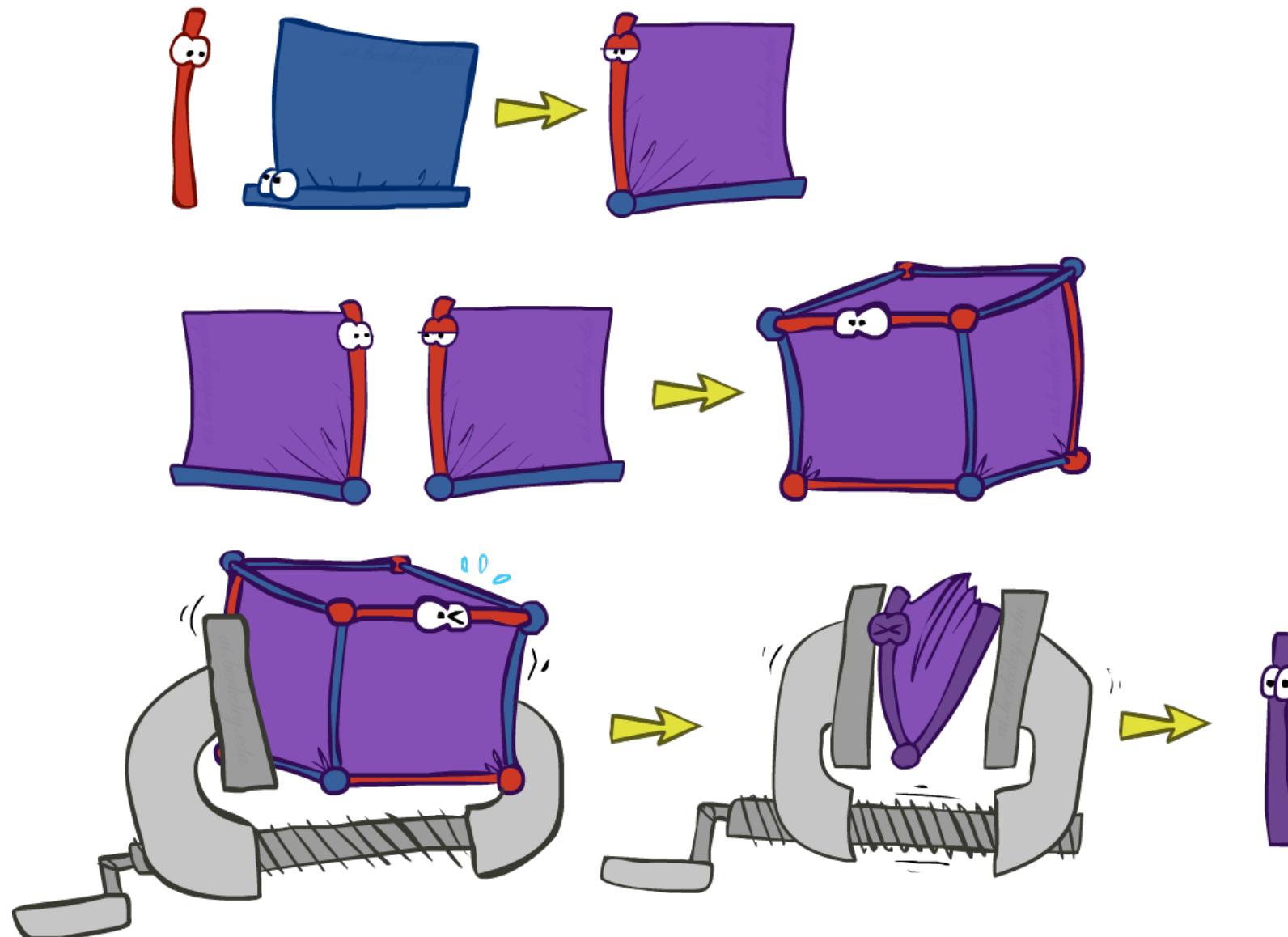
$$Z = \sum_q P(Q, e_1 \dots e_k)$$

$$P(Q, e_1 \dots e_k) = \sum_{h_1 \dots h_r} \underbrace{P(Q, h_1 \dots h_r, e_1 \dots e_k)}_{\text{Multiple joins}}$$

■ Multiple eliminations X_1, X_2, \dots, X_n

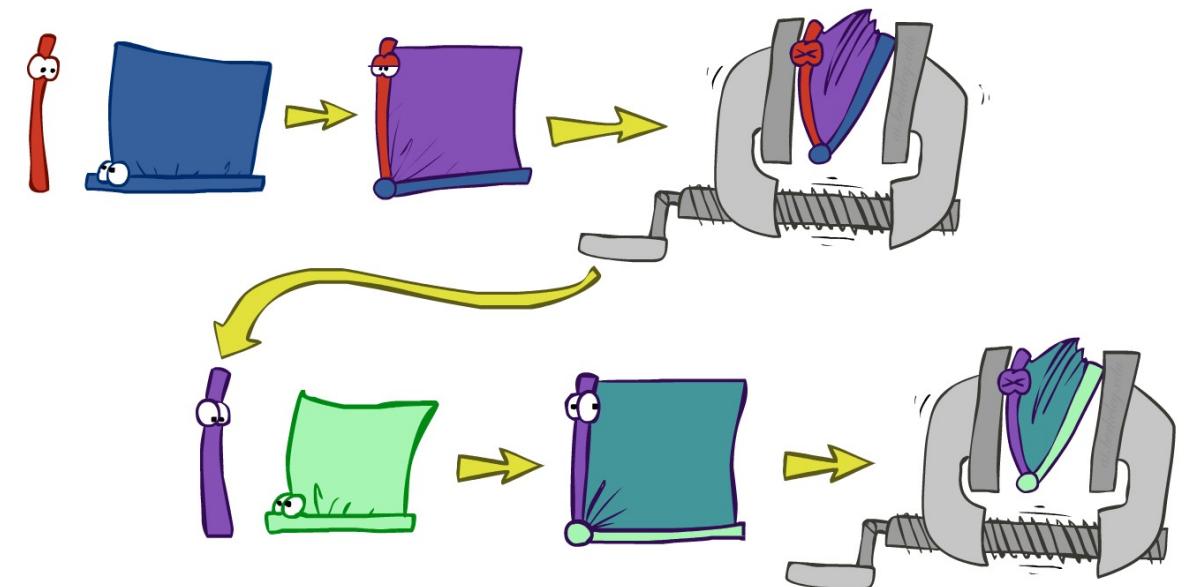
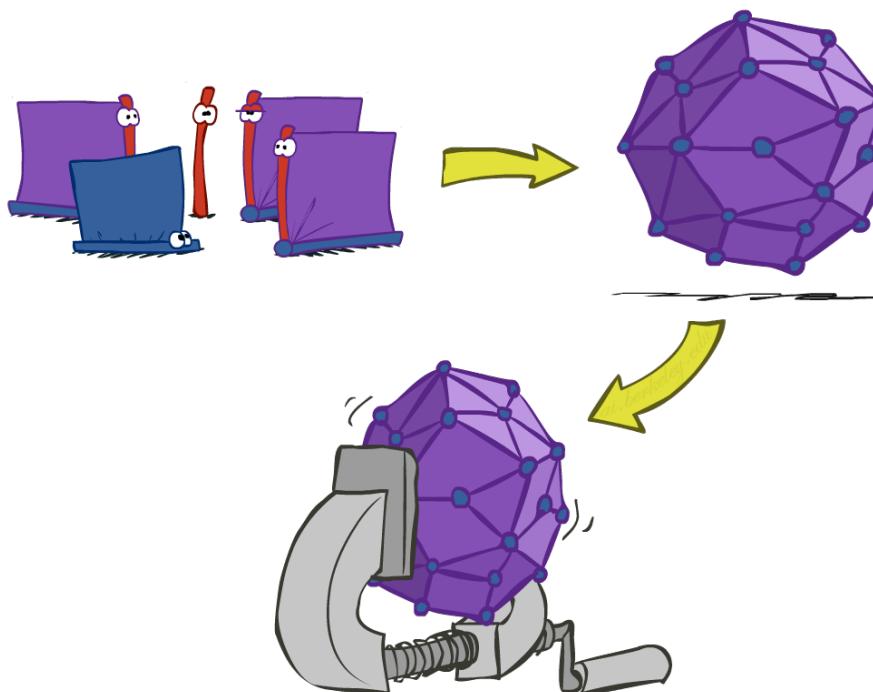
$$P(Q|e_1 \dots e_k) = \frac{1}{Z} P(Q, e_1 \dots e_k)$$

Multiple Join, Multiple Eliminate (= Inference by Enumeration)

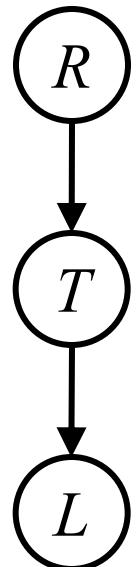


Inference by Enumeration vs. Variable Elimination

- 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



Traffic Domain



$$P(L) = ?$$

- Inference by Enumeration

$$= \sum_t \sum_r P(L|t) P(r) P(t|r)$$

Join on r

Join on t

Eliminate r

Eliminate t

- Variable Elimination

$$= \sum_t P(L|t) \sum_r P(r) P(t|r)$$

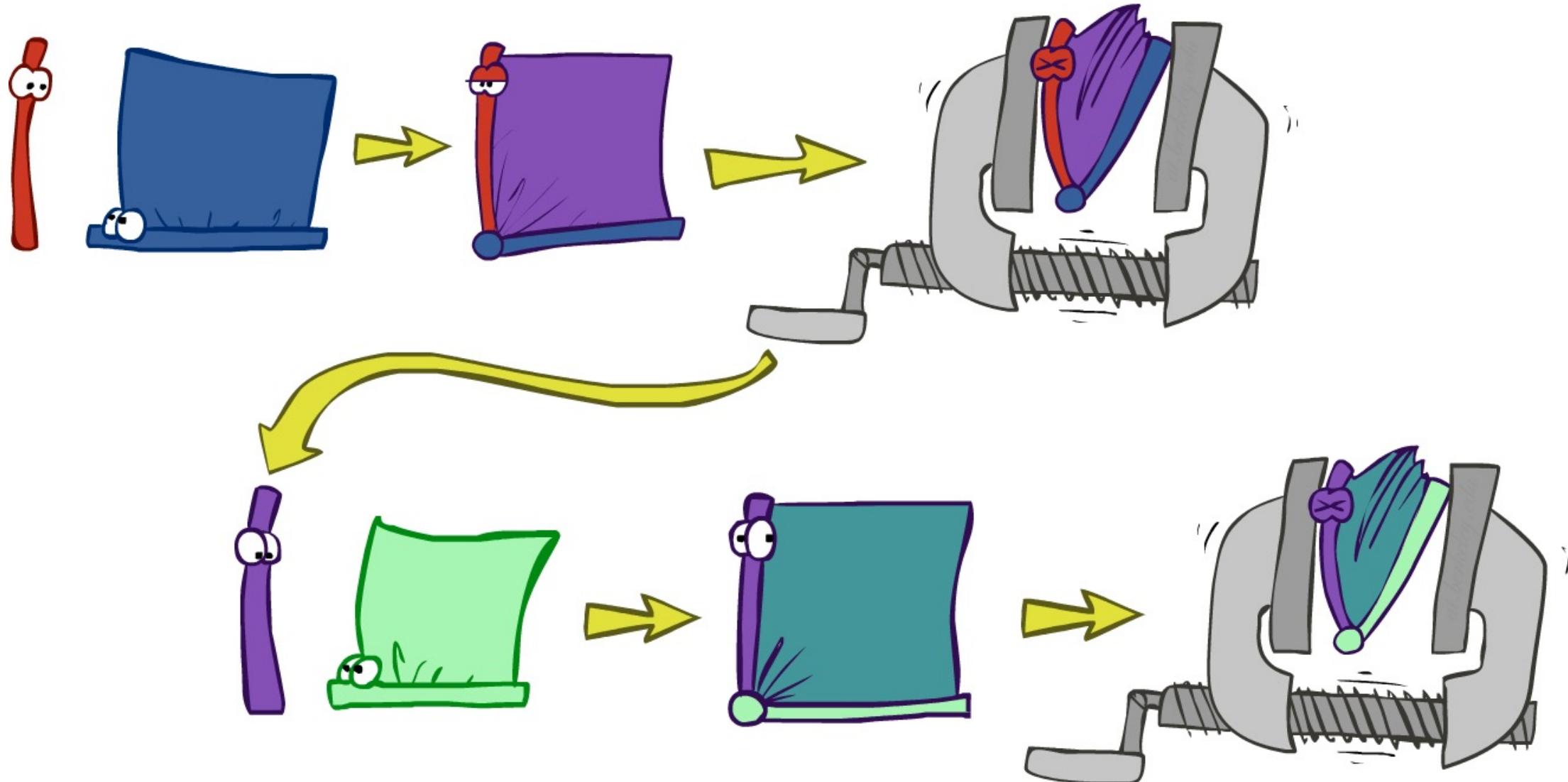
Join on r

Eliminate r

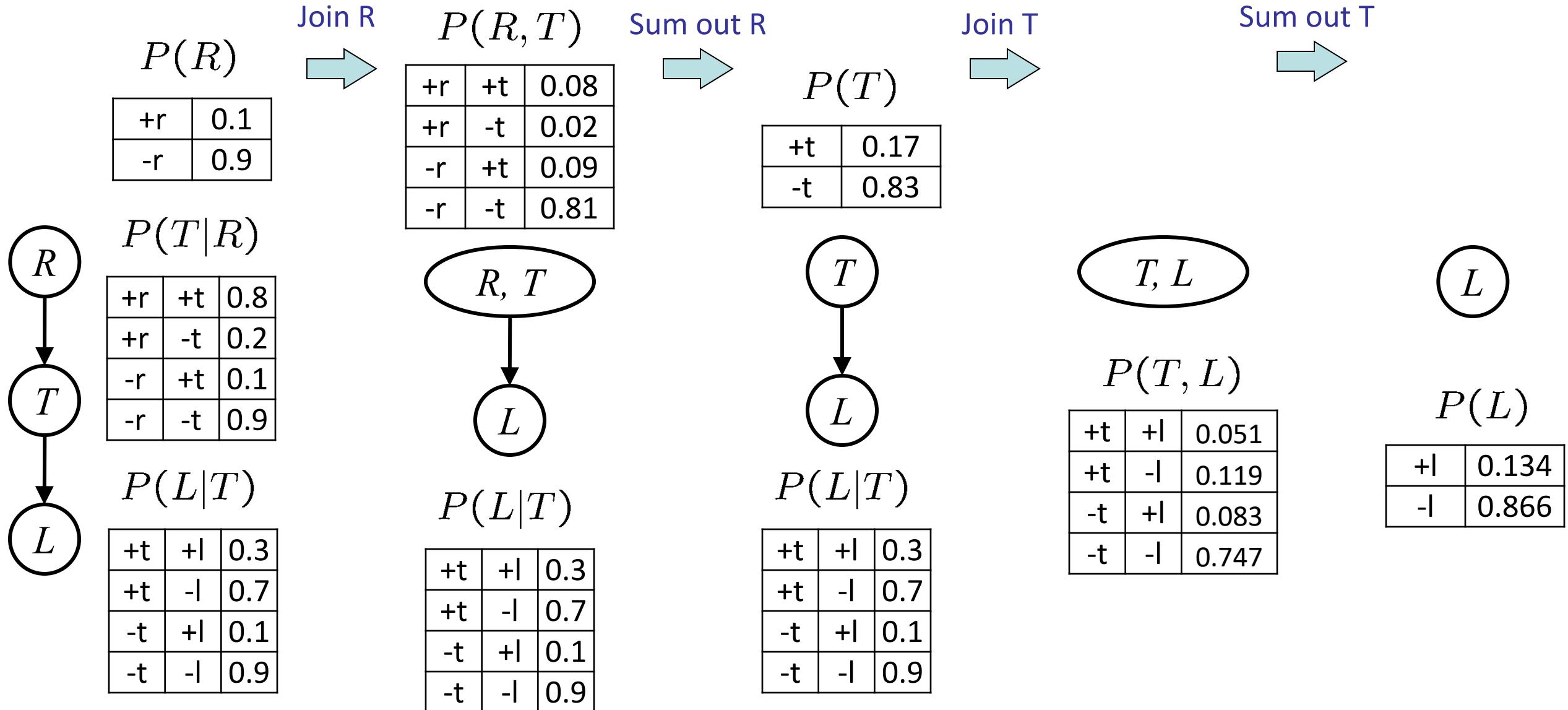
Join on t

Eliminate t

Marginalizing Early (= Variable Elimination)



Marginalizing Early! (aka VE)



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
----	-----

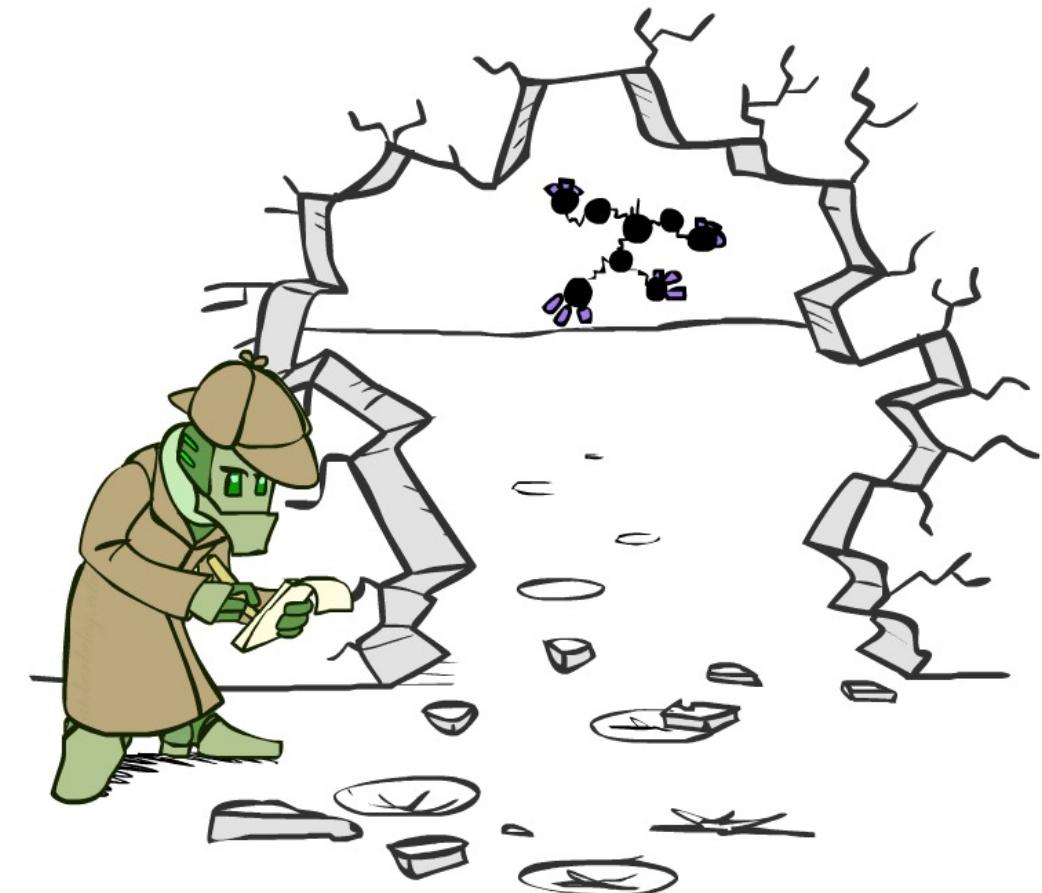
$$P(T|+r)$$

+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 would end up with:

$$P(+r, L)$$

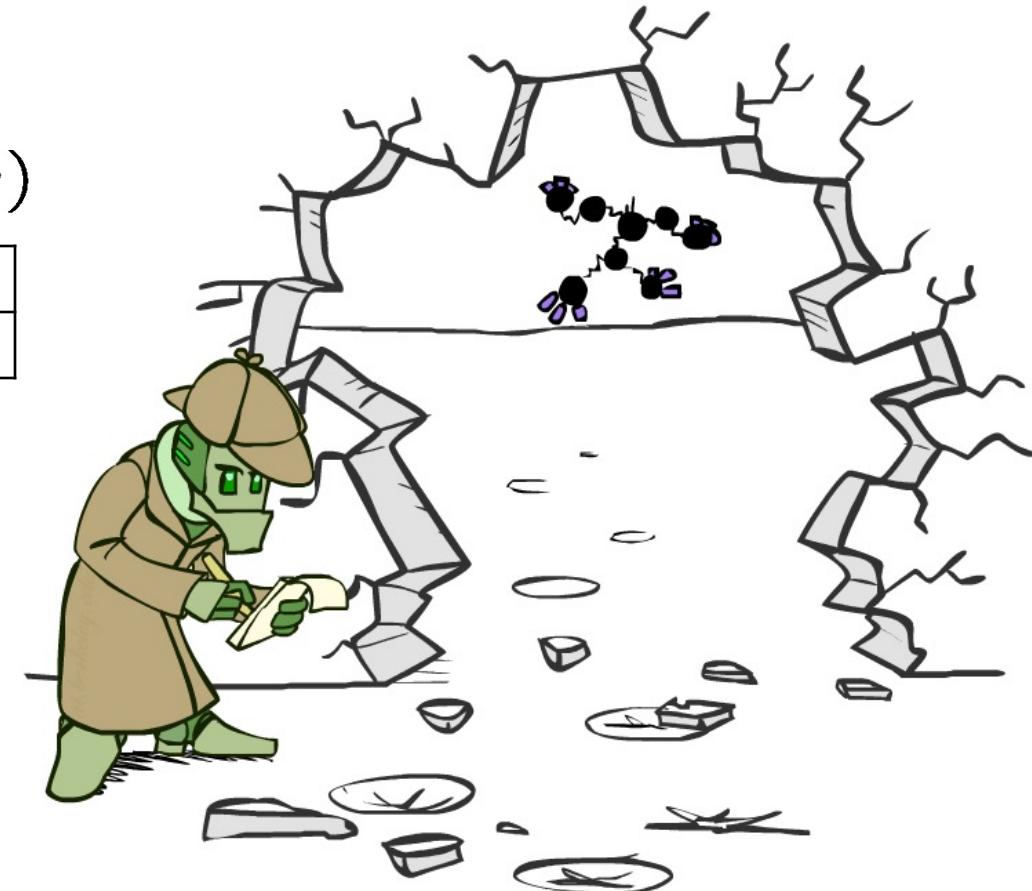
+r	+l	0.026
+r	-l	0.074

Normalize

$$P(L | +r)$$

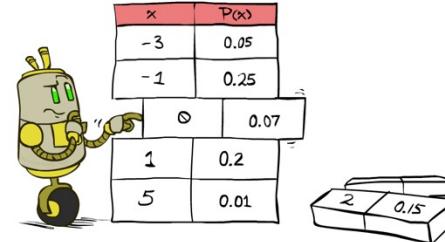
+l	0.26
-l	0.74

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

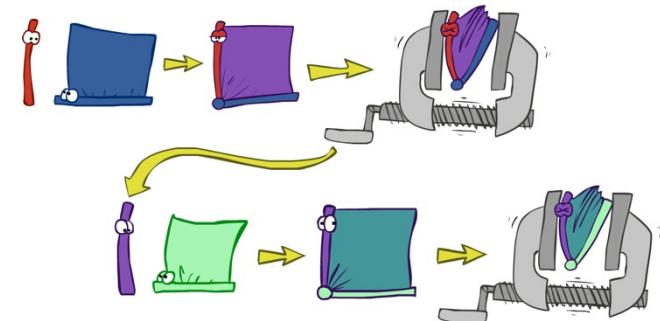


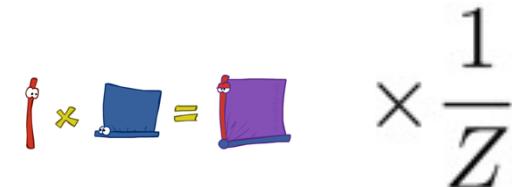
General Variable Elimination

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



x	P(x)
-3	0.05
-1	0.25
0	0.07
1	0.2
5	0.01
2	0.15




$$\left(\times \text{ [blue card]} = \text{ [purple card]} \right) \times \frac{1}{Z}$$

Why does this reduce computation?

- Consider $\mathbf{uwy + uwz + uxy + uxz + vwy + vwz + vxy + vxz}$
 - 16 multiplies, 7 adds
 - Lots of repeated subexpressions!
- Rewrite as $(\mathbf{u+v})(\mathbf{w+x})(\mathbf{y+z})$
 - 2 multiplies, 3 adds
- Sum early reduces computation!

Example: Alarm Net (Math)

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

$P(B)$	$P(E)$	$P(A B, E)$	$P(j A)$	$P(m A)$
--------	--------	-------------	----------	----------

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

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

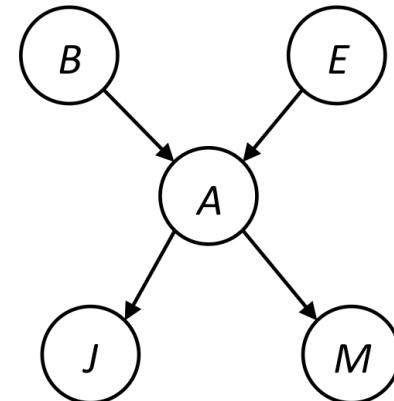
$$= \sum_{e,a} P(B)P(e)P(a|B, e)P(j|a)P(m|a)$$

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

$$= \sum_e P(B)P(e)f_1(j, m|B, e)$$

$$= P(B) \sum_e P(e)f_1(j, m|B, e)$$

$$= P(B)f_2(j, m|B)$$



marginal can be obtained from joint by summing out

use Bayes' net joint distribution expression

use $x^*(y+z) = xy + xz$

joining on a, and then summing out gives f_1

use $x^*(y+z) = xy + xz$

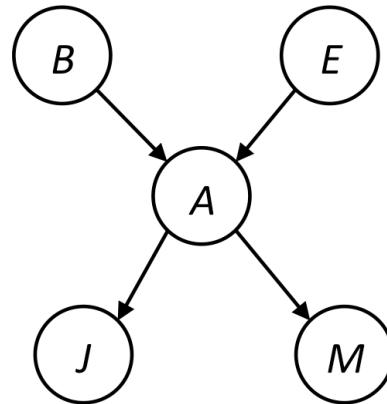
joining on e, and then summing out gives f_2

All we are doing is exploiting $uw\bar{y} + uw\bar{z} + u\bar{x}\bar{y} + u\bar{x}\bar{z} + v\bar{w}\bar{y} + v\bar{w}\bar{z} + v\bar{x}\bar{y} + v\bar{x}\bar{z} = (u+v)(w+x)(y+z)$ to improve computational efficiency!

Example: Alarm Net (Factors)

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

$P(B)$	$P(E)$	$P(A B, E)$	$P(j A)$	$P(m A)$
--------	--------	-------------	----------	----------

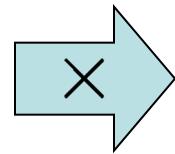


Choose A

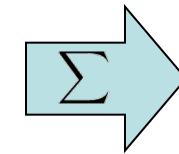
$$P(A|B, E)$$

$$P(j|A)$$

$$P(m|A)$$



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



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

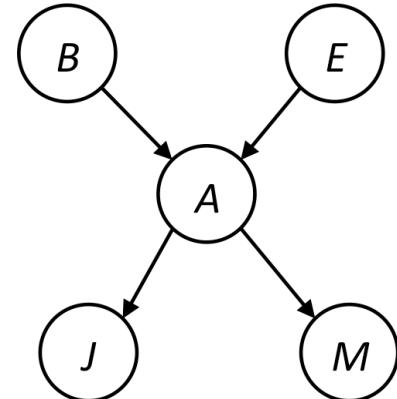
$P(B)$	$P(E)$	$P(j, m B, E)$
--------	--------	----------------

Example: Alarm Net (Factors) II

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

Choose E

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



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

Finish with B

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

Order matters

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

Initial:

$P(Z), P(X_1|Z), P(X_2|Z), P(X_3|Z), P(y_1|X_1), P(y_2|X_2), P(y_3|X_3)$

Choose X_1 :

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

Choose X_2 :

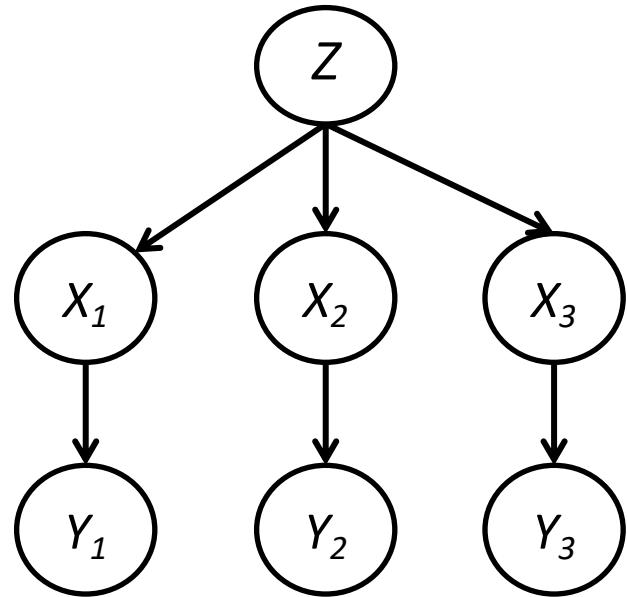
$P(Z), P(X_3|Z), P(y_3|X_3), f_1(y_1|Z), f_2(y_2|Z)$

Choose Z :

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

Joint on X_3 :

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



- Largest factor generated has dimension
▪ Size = 2 (assuming binary)

Order matters

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

Initial:

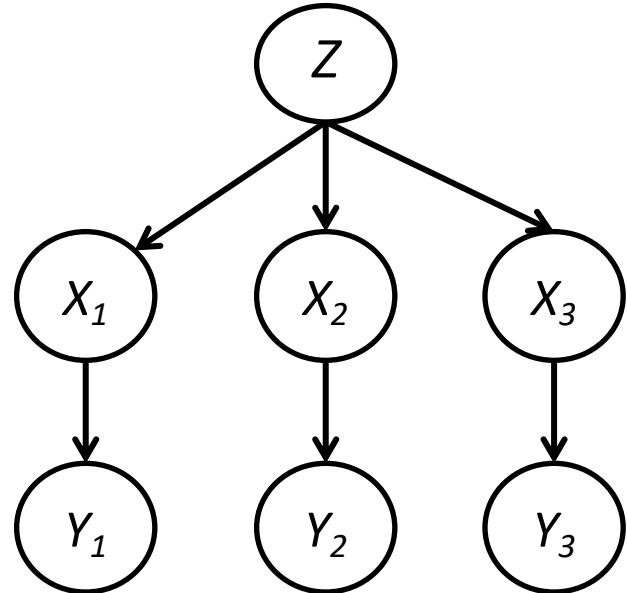
$P(Z), P(X_1|Z), P(X_2|Z), P(X_3|Z), P(y_1|X_1), P(y_2|X_2), P(y_3|X_3)$

Choose Z :

Choose X_1 :

Choose X_2 :

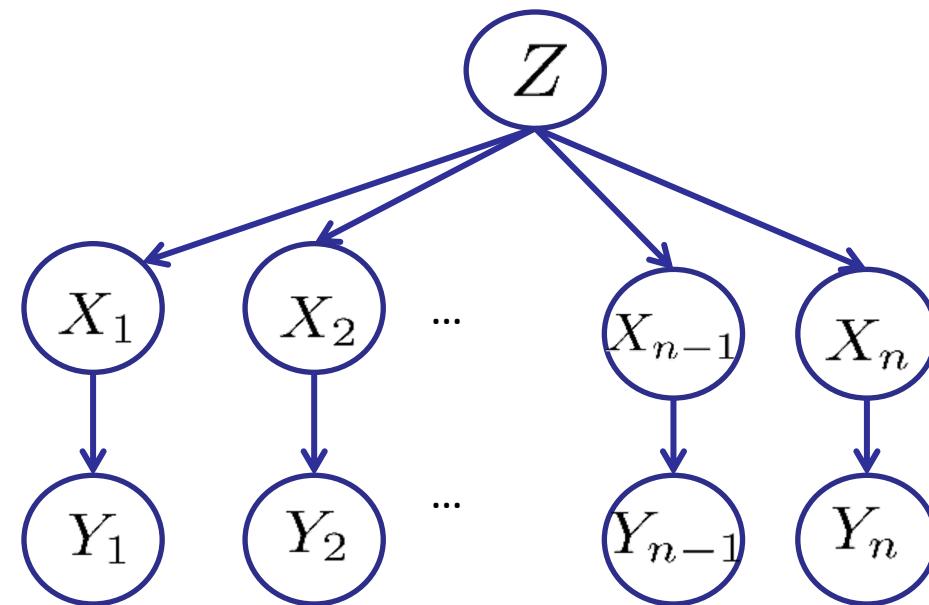
Joint on X_3 :



- Largest factor has n dimensions
 - $n = \# \text{ leaves}$
 - Size = 2^n (assuming binary)

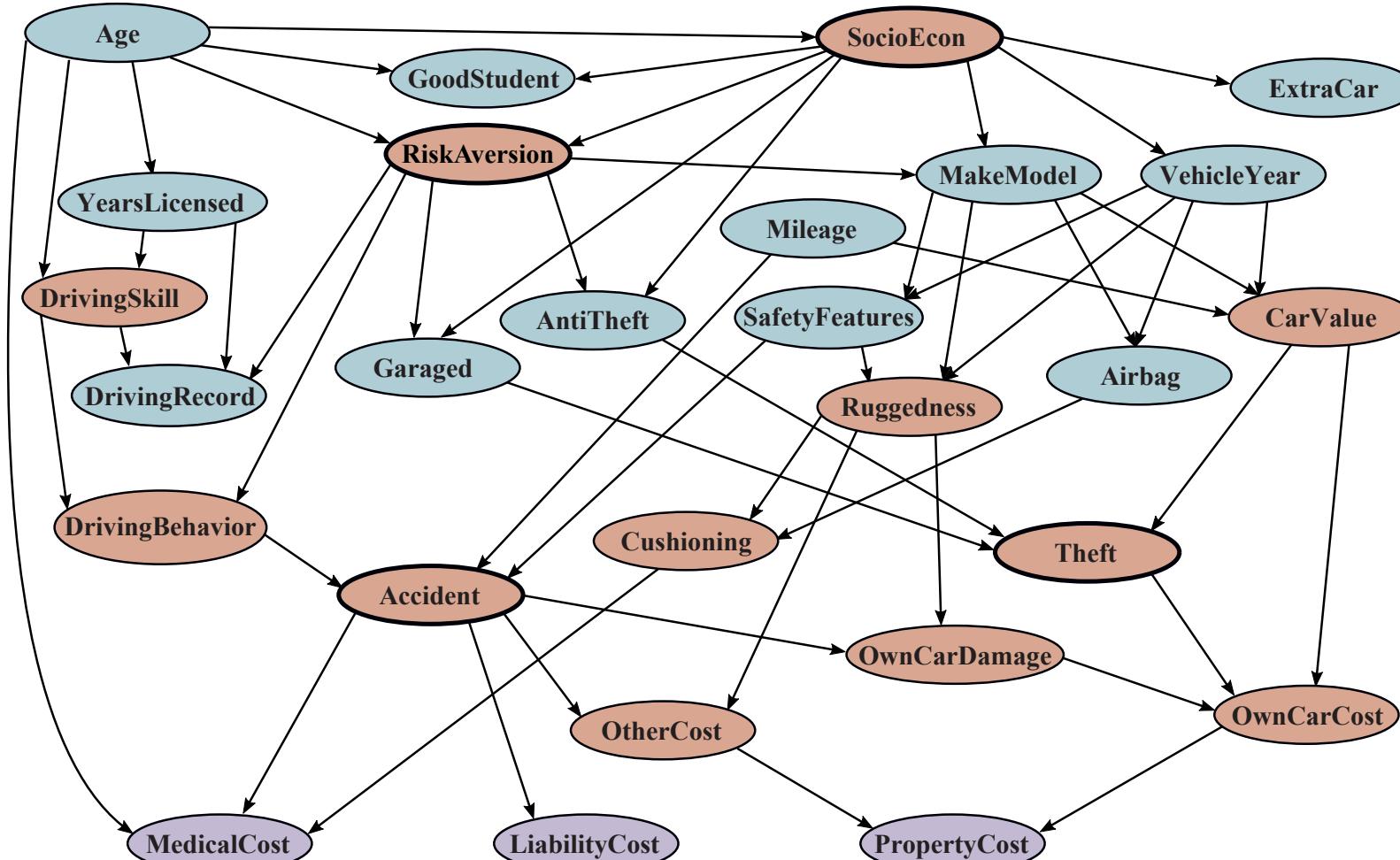
Order matters

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



- Answer: 2^n versus 2 (assuming binary)
 - Ordering can greatly affect efficiency!

Example: Car Insurance Net



27 Variables

Enumeration: 227M operations

Elimination: 221K operations

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: assign variables so that the sentence is true

Clause need be true

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

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

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

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

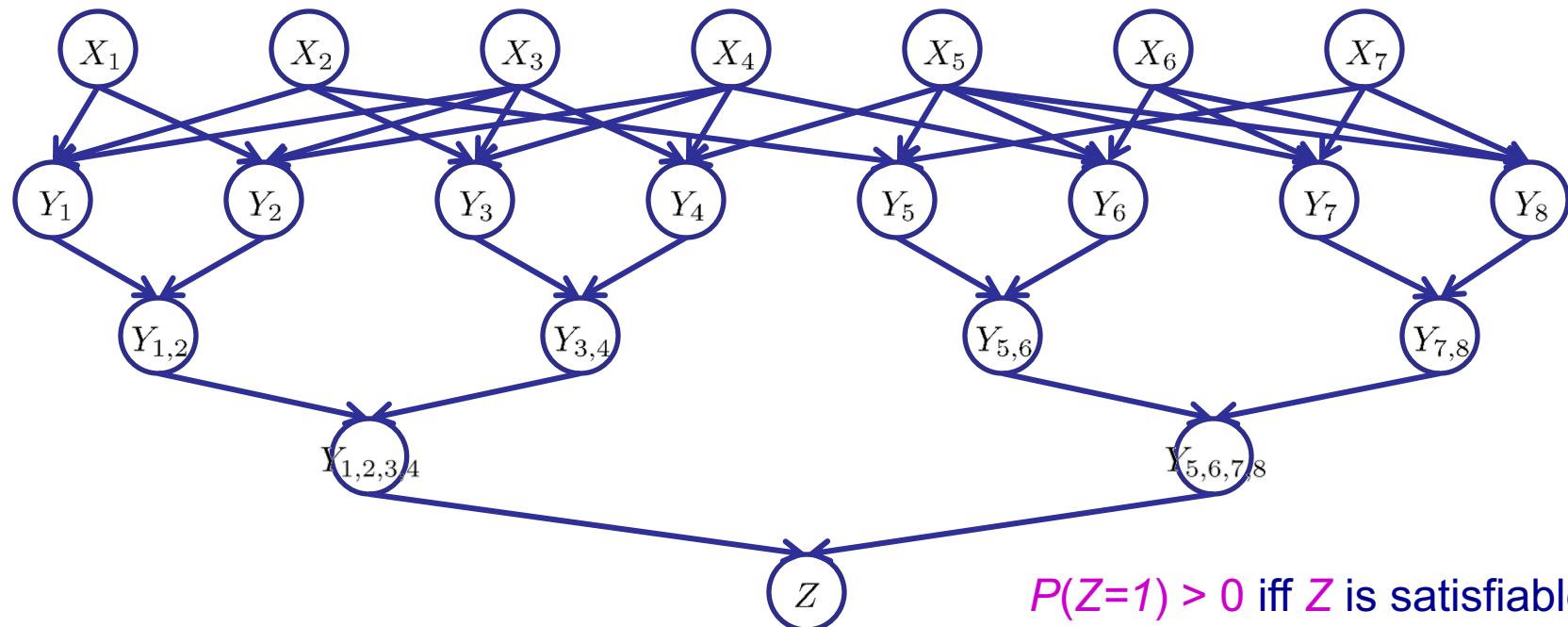
$$Y_{1,2} = Y_1 \wedge Y_2$$

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

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

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

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

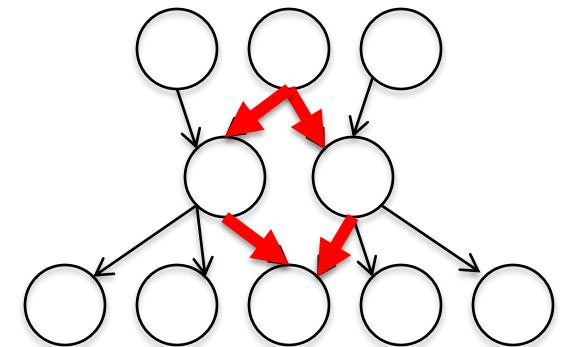
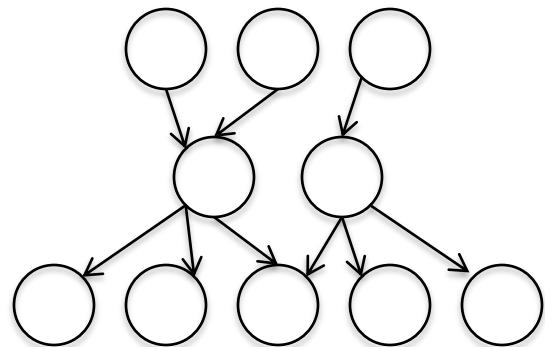
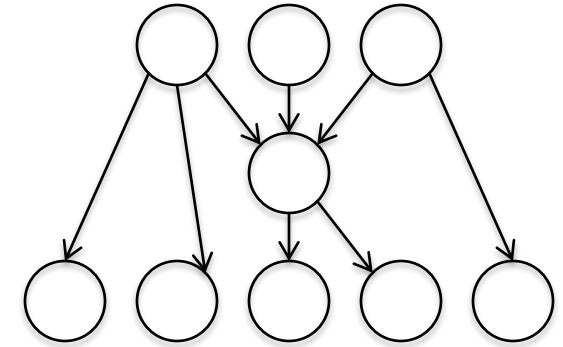
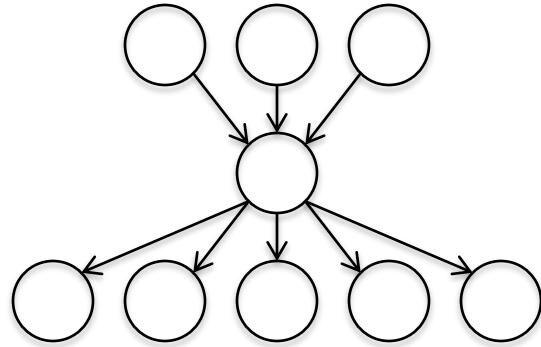


$$P(Z=1) > 0 \text{ iff } Z \text{ is satisfiable}$$

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

“Easy” Structures: Polytrees

- A polytree is a directed acyclic graph whose underlying undirected graph is a tree
- For polytrees you can always find an ordering that is efficient
 - Try it in homework!



Bayes Nets

✓ Representation

✓ Probabilistic Inference

- Enumeration (exact, exponential complexity)
- Variable elimination (exact, worst-case exponential complexity, often better)
- Probabilistic inference is NP-complete

✓ Conditional Independences

- Sampling
- Learning from data