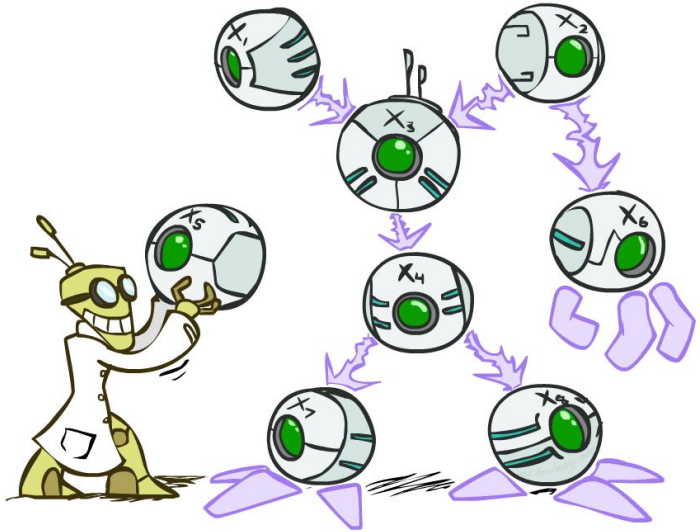
Bayes Nets

Dr. Angelica Lim
Assistant Professor
School of Computing Science
Simon Fraser University, Canada
Nov. 11, 2024

Slide Credits:Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley (ai.berkeley.edu).
Pearson 2021

Bayes Nets



[These 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.]

Conditional Independence

Conditional Independence

- ***Conditional independence*** is our most basic and robust form of knowledge about uncertain environments.

- X is conditionally independent of Y given Z if and only if:

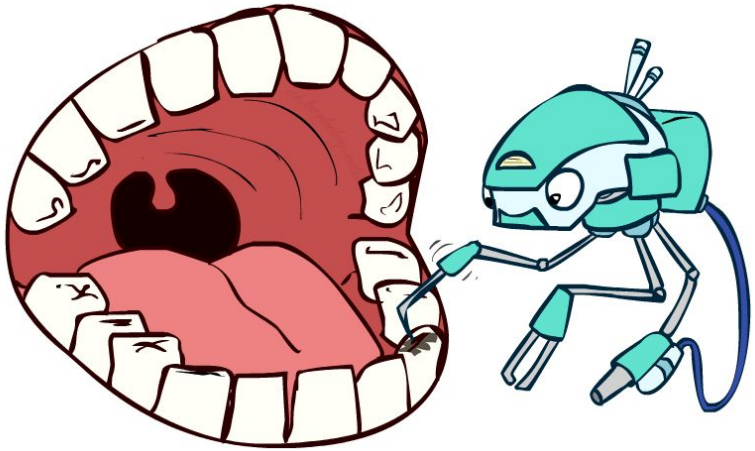
$$\forall x,y,z \qquad P(x \mid y, z) = P(x \mid z)$$

or, equivalently, if and only if

$$\forall x,y,z \qquad P(x, y \mid z) = P(x \mid z)\, P(y \mid z)$$

Conditional Independence

- P(Toothache, Cavity, Catch)
- If I have a cavity, the probability that the probe catches it doesn't depend on whether I have a toothache:
  - P(+catch | +toothache, +cavity) = P(+catch | +cavity)
- The same independence holds if I don't have a cavity:
  - P(+catch | +toothache, -cavity) = P(+catch| -cavity)
- Catch is *conditionally independent* of Toothache given Cavity:
  - P(Catch | Toothache, Cavity) = P(Catch | Cavity)

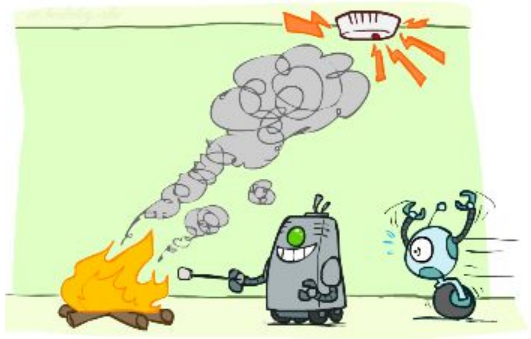


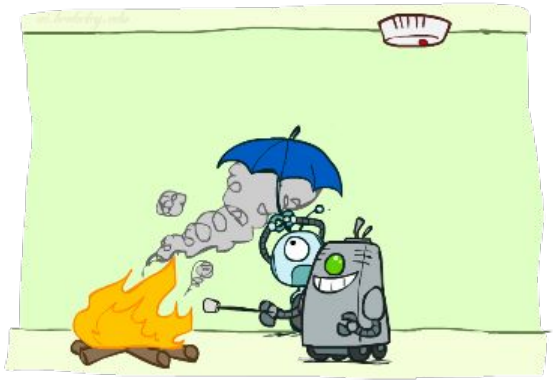
- Equivalent statements:
  - P(Toothache | Catch , Cavity) = P(Toothache | Cavity)
  - P(Toothache, Catch | Cavity) = P(Toothache | Cavity) P(Catch | Cavity)
  - One can be derived from the other easily

Conditional Independence

- What about this domain:
  - Fire
  - Smoke
  - Alarm

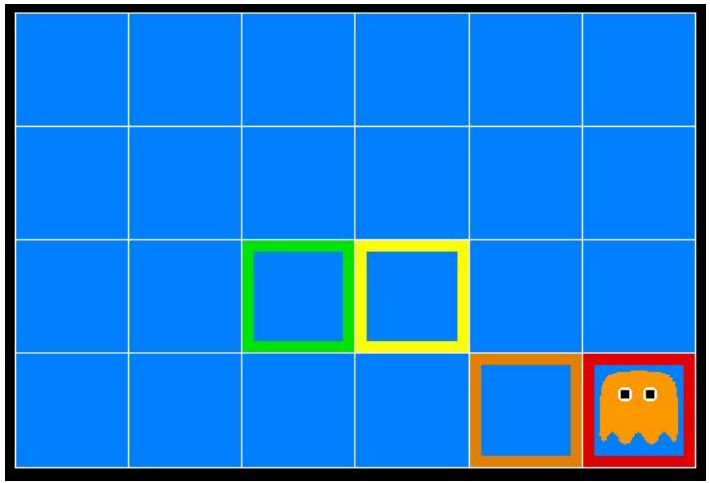
$P(A|S) = P(A|S,F)$





Ghostbusters

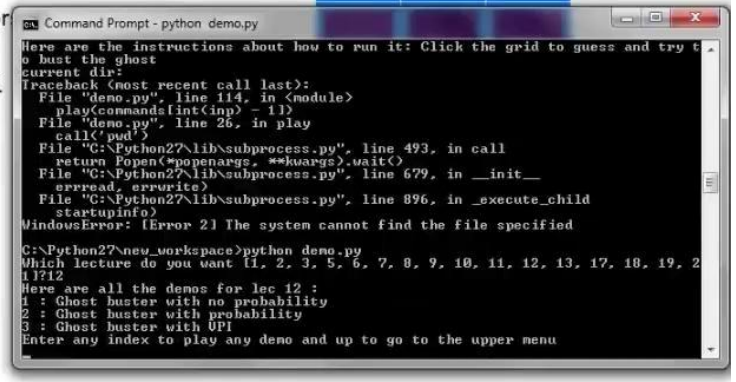
- A ghost is in the grid somewhere
- **Sensor** readings tell how close a square is to the ghost
  - On the ghost: usually red
  - 1 or 2 away: usually orange
  - 3 or 4 away: usually yellow
  - 5+ away: usually green
- Click on squares until confident of location, then "***bust***"



Video of Demo Ghostbusters with Probability

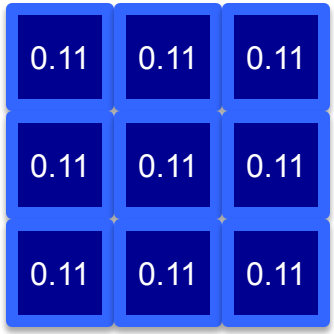
Ghostbusters, Revisited

- Let's say we have two distributions:
  - Prior distribution over ghost location: P(G)
    - Let's say this is uniform
  - Sensor reading model: P(C | G)
    - Given: we know what our sensor
    - C = color measured at (1,1)
    - E.g. P(C = yellow | G=(1,1)) = 0.1



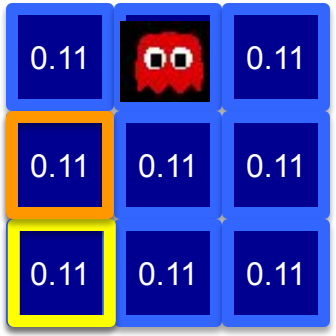
Ghostbusters model

- Variables and ranges:
  - G (ghost location) in {(1,1),…,(3,3)}
  - $C_{x,y}$ (color measured at square x,y) in {red,orange,yellow,green}



- Ghostbuster physics:
  - ***Uniform prior distribution*** over ghost location: $P(G)$
  - ***Sensor model***: $P(C_{x,y} \mid G)$ (depends only on distance to G)
    - E.g. $P(C_{1,1} = \text{yellow} \mid G = (1,1)) = 0.1$

Ghostbusters model, contd.



- $P(G, C_{1,1}, \ldots C_{3,3})$ has $9 \times 4^9 = 2{,}359{,}296$ entries!!!
- Ghostbuster independence:
  - Are $C_{1,1}$ and $C_{1,2}$ independent?
    - E.g., does $P(C_{1,1} = \text{yellow}) = P(C_{1,1} = \text{yellow} \mid C_{1,2} = \text{orange})$?
- Ghostbuster physics again:
  - $P(C_{x,y} \mid G)$ ***depends only on distance to G***
    - So $P(C_{1,1} = \text{yellow} \mid G = (2,3)) = P(C_{1,1} = \text{yellow} \mid G = (2,3), C_{1,2} = \text{orange})$
    - I.e., $C_{1,1}$ is ***conditionally independent*** of $C_{1,2}$ ***given*** G

Ghostbusters model, contd.

Apply the chain rule to decompose the joint probability model:

- $P(G, C_{1,1}, \ldots C_{3,3}) =$
  $P(G)\, P(C_{1,1} \mid G)\, P(C_{1,2} \mid G, C_{1,1})\, P(C_{1,3} \mid G, C_{1,1}, C_{1,2}) \ldots P(C_{3,3} \mid G, C_{1,1}, \ldots, C_{3,2})$

Exponential in the amount of space needed.
If each table has n entries and there are k tables,
the join will construct a table of size n^k

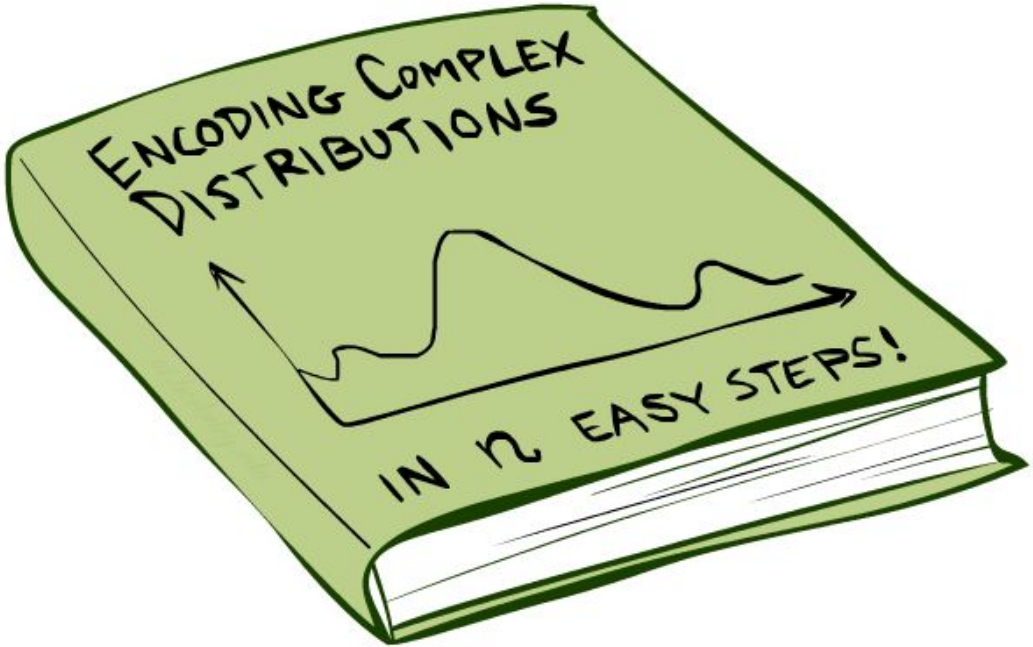
Now simplify using conditional independence:

- $P(G, C_{1,1}, \ldots C_{3,3}) = P(G)\, P(C_{1,1} \mid G)\, P(C_{1,2} \mid G)\, P(C_{1,3} \mid G) \ldots P(C_{3,3} \mid G)$

By simplifying the right hand side, the amount of space needed decreases from exponential to linear.

I.e., conditional independence properties of ghostbuster physics simplify the probability model from ***exponential*** to ***linear*** in the number of squares

Bayes Nets: Big Picture



Bayes Nets: Big Picture

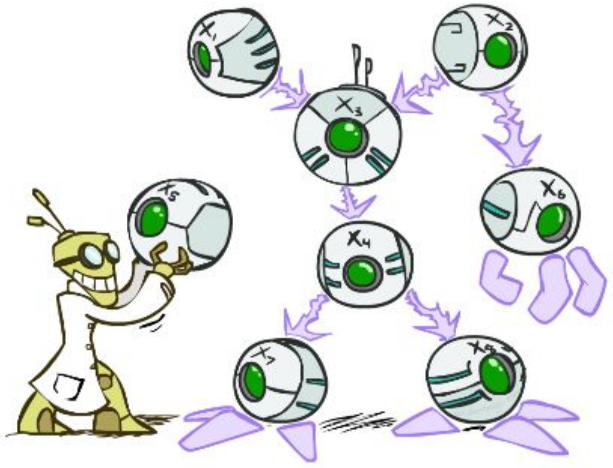
- Bayes nets: a technique for describing complex joint distributions (models) using simple, conditional distributions
  - A subset of the general class of graphical models

Use local causality/conditional independence:

- the world is composed of many variables,
- each interacting locally with a few others

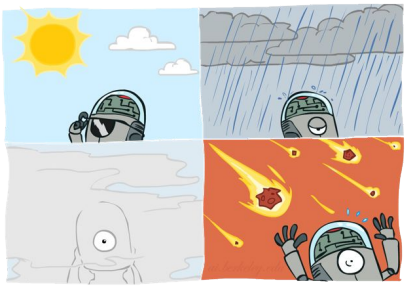
Outline

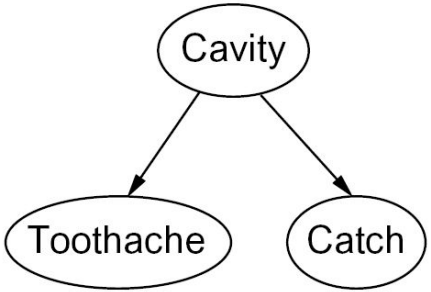
- Representation
- Exact inference
- Approximate inference

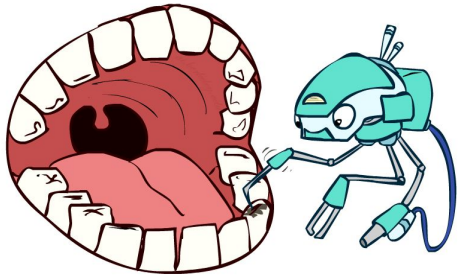


Graphical Model Notation

- **Nodes**: variables (with domains)
  - Can be assigned (observed) or unassigned (unobserved)
- **Arcs**: interactions
  - Indicate “direct influence” between variables
  - Formally: absence of arc encodes conditional independence (more later)
- For now: imagine that arrows mean direct **causation** (in general, they don’t!)

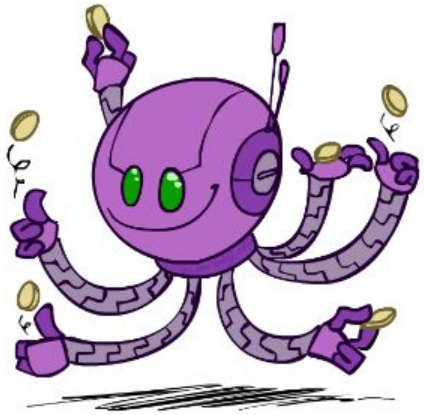






Example: Coin Flips

- *n* independent coin flips

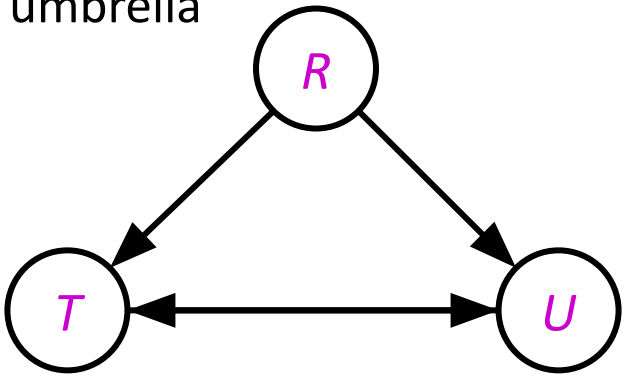


X_1 X_2 . . . X_n

- No interactions between variables: strict independence

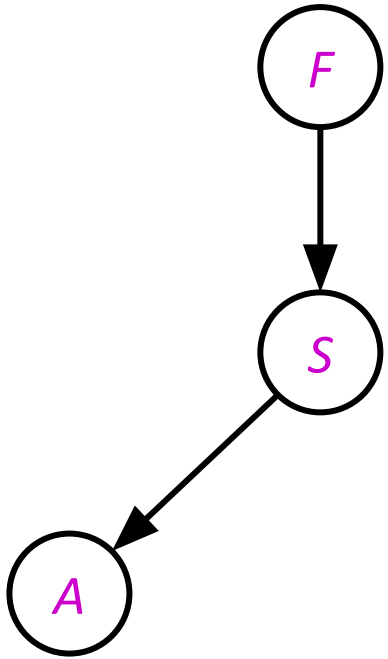
Example: Traffic

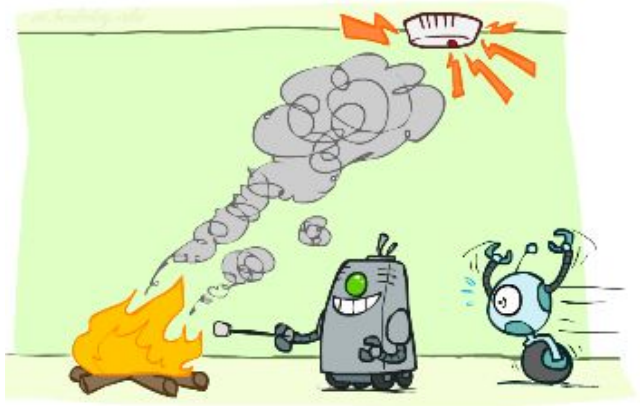
- Variables:
  - ***T***: There is traffic
  - ***U***: I'm holding my umbrella
  - ***R***: It rains

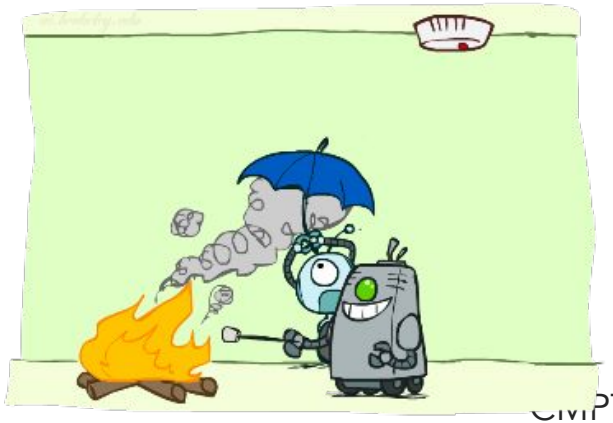


Example: Smoke alarm

- Variables:
  - ***F***: There is fire
  - ***S***: There is smoke
  - ***A***: Alarm sounds

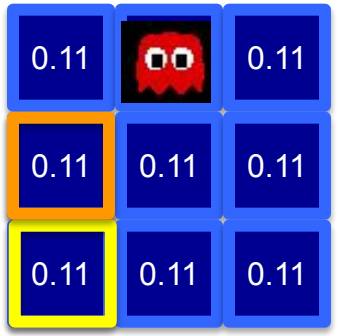




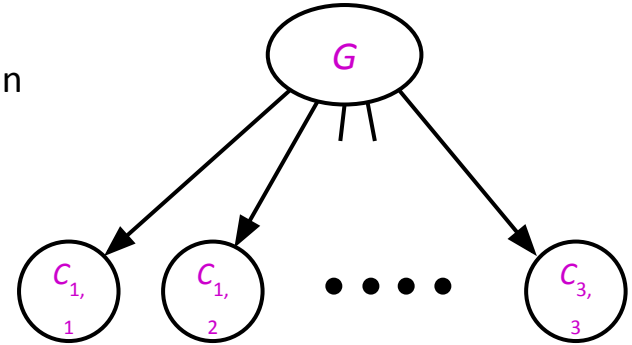


Example: Ghostbusters

- Variables:
  - G: The ghost's location
  - $C_{1,1}, \ldots C_{3,3}$:
    The observation at each location

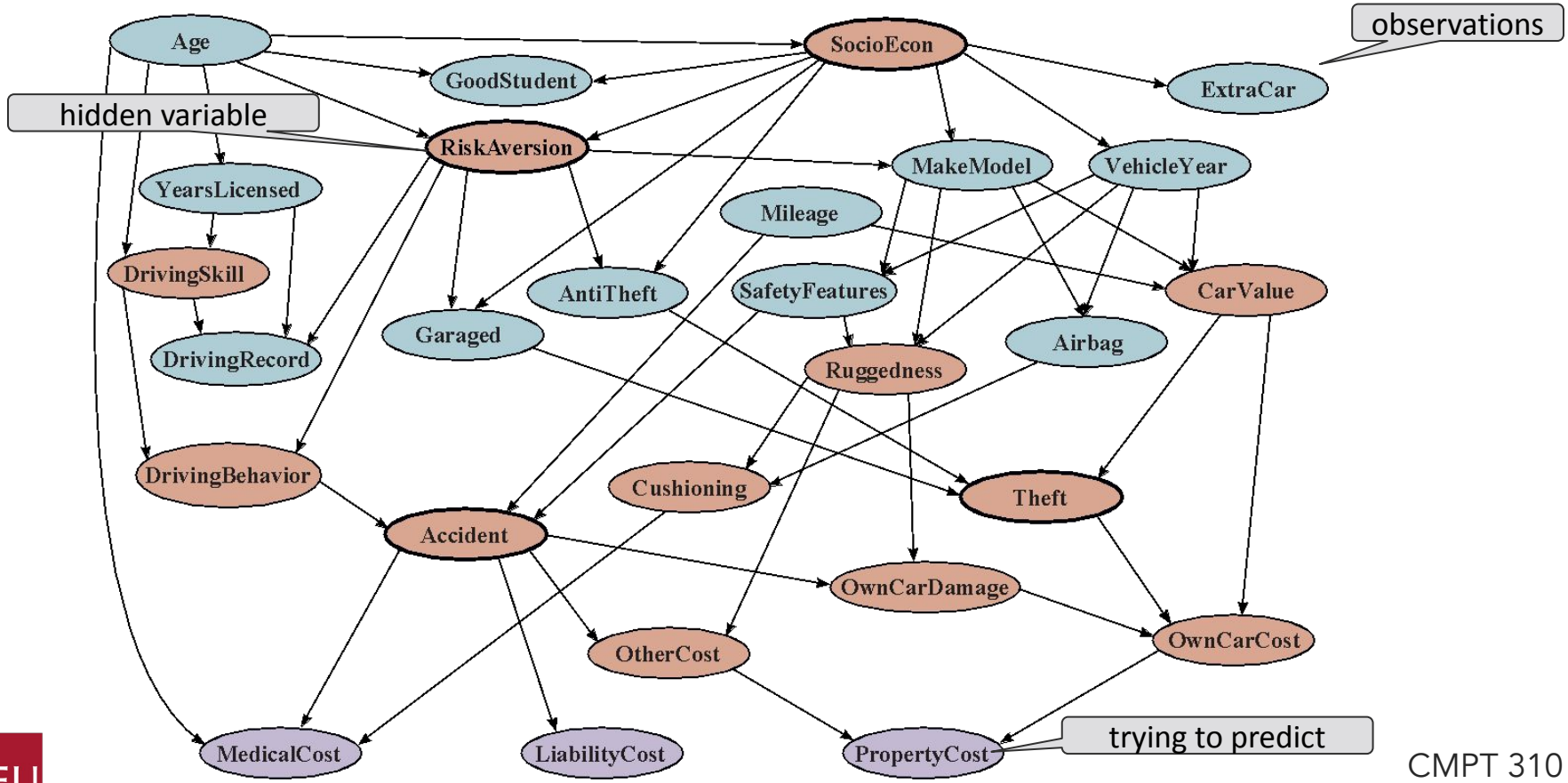


- Want to estimate:
  $P(G \mid C_{1,1}, \ldots C_{3,3})$

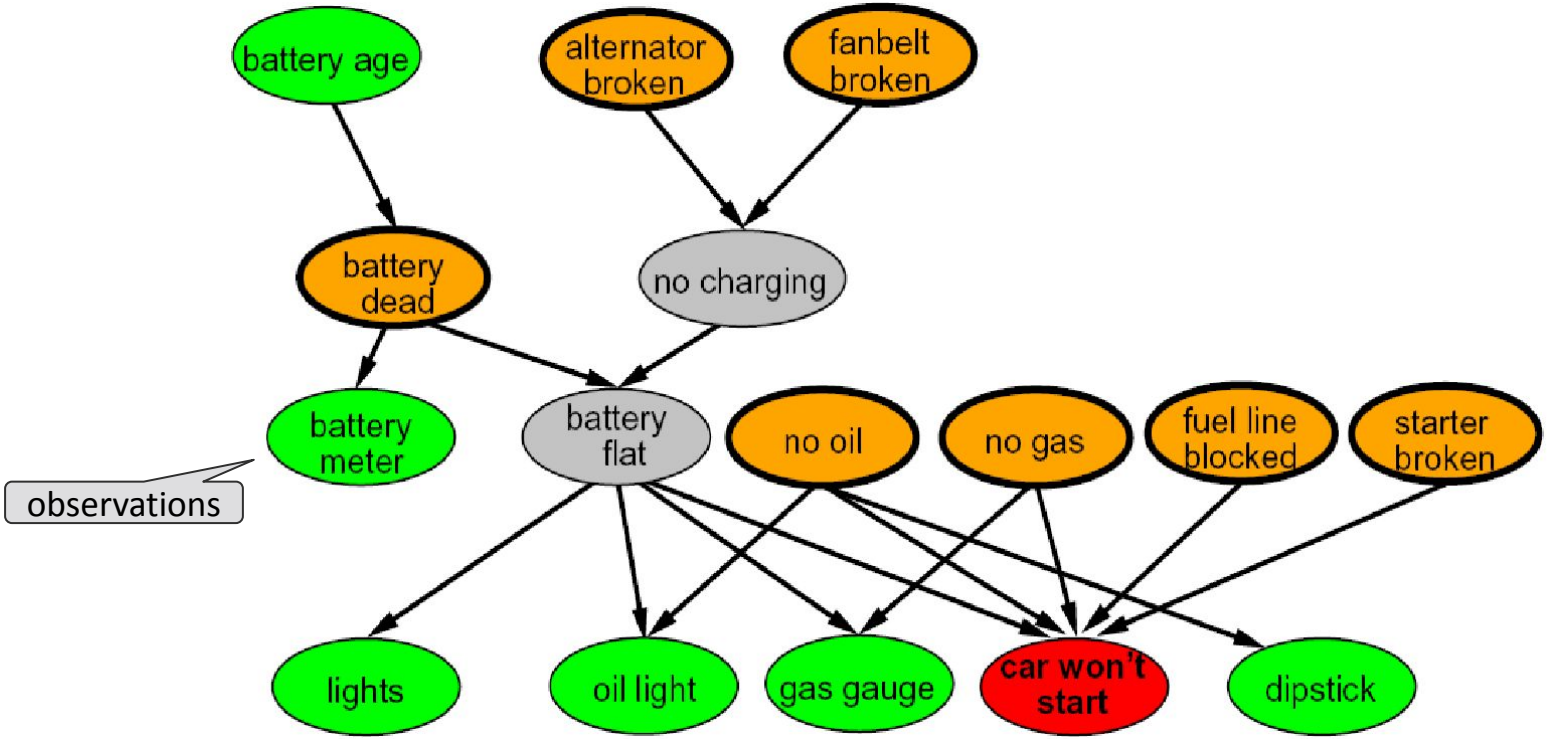


- This is called a ***Naïve Bayes*** model:
  - One discrete query variable (often called the ***class*** or ***category*** variable)
  - All other variables are (potentially) evidence variables
  - Evidence variables are all conditionally independent given the query variable

Example: Car Insurance



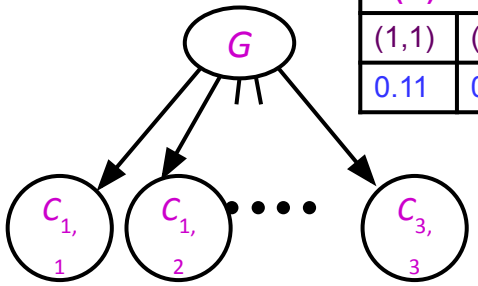
Example: Car Won't Start



Bayes Net Syntax and Semantics

Bayes Net Syntax

- A set of nodes, one per variable X_i
- A directed, acyclic graph
- A conditional distribution for each node given its ***parent variables*** in the graph
  - ***CPT*** (conditional probability table); each row is a distribution for child given values of its parents



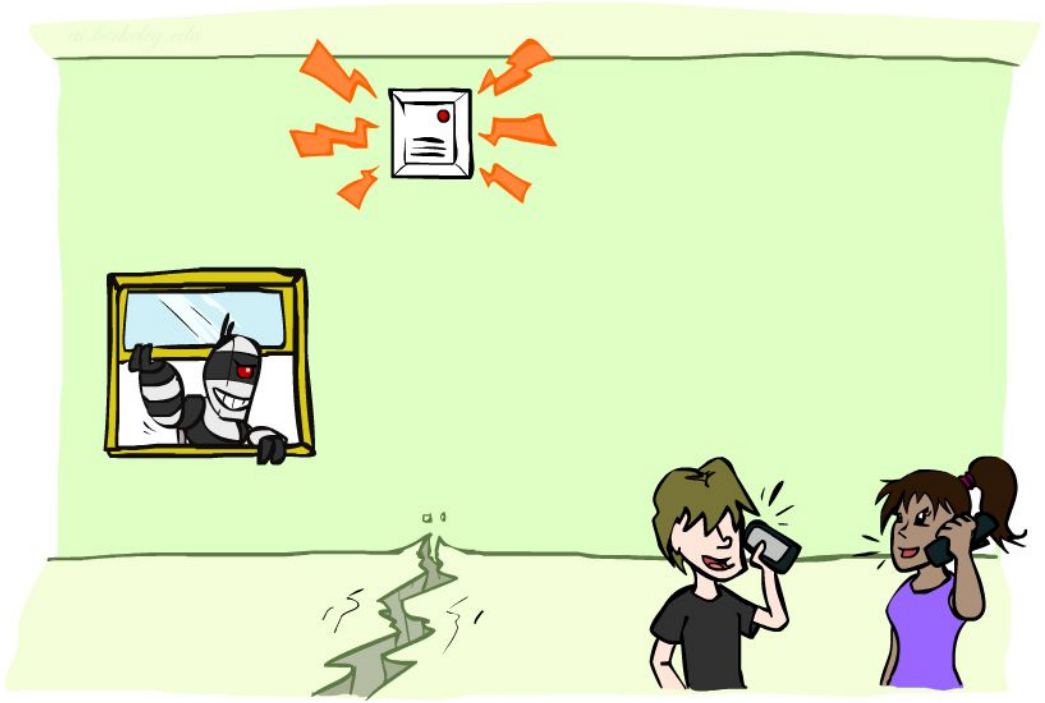
P(G)			
(1,1)	(1,2)	(1,3)	…
0.11	0.11	0.11	…

G	P($C_{1,1}$ | G)			
	g	y	o	r
(1,1)	0.01	0.1	0.3	0.59
(1,2)	0.1	0.3	0.5	0.1
(1,3)	0.3	0.5	0.19	0.01
…				

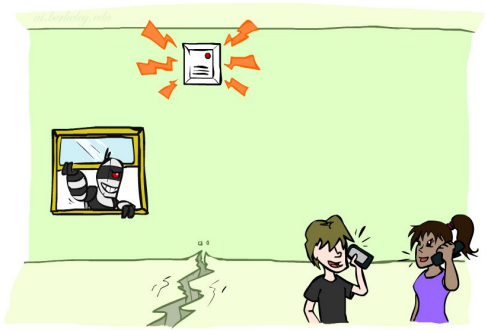
Bayes net = Topology (graph) + Local Conditional Probabilities

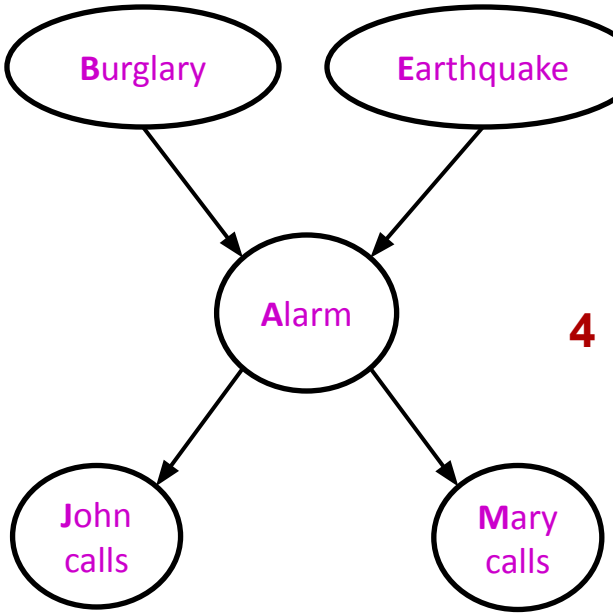
Example: Alarm Network

- Variables
  - B: Burglary
  - E: Earthquake
  - A: Alarm goes off
  - J: John calls
  - M: Mary calls



Example: Alarm Network





P(B)	
true	false
0.001	0.999

1

P(E)	
true	false
0.002	0.998

1

B	E	P(A|B,E)	
		true	false
true	true	0.95	0.05
true	false	0.94	0.06
false	true	0.29	0.71
false	false	0.001	0.999

4

A	P(J|A)	
	true	false
true	0.9	0.1
false	0.05	0.95

2

A	P(M|A)	
	true	false
true	0.7	0.3
false	0.01	0.99

2

Number of ***free parameters*** in each CPT:

Parent range sizes $d_1,\ldots,d_k$

Child range size d
Each table row must sum to 1

$(d-1) \prod_i d_i$

Bayes net global semantics

- Bayes nets encode joint distributions as product of conditional distributions on each variable:

$$P(X_1,..,X_n) = \prod_i P(X_i \mid Parents(X_i))$$

- Exploits sparse structure: number of parents is usually small

Size of a Bayes Net

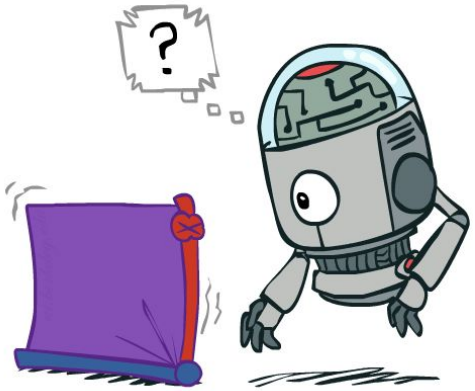
- How big is a joint distribution over N variables, each with d values?

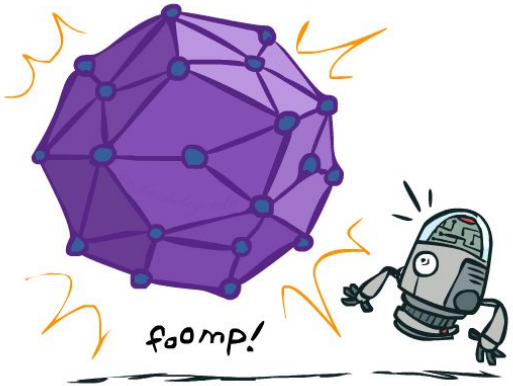
  d^N

- How big is an N-node net if nodes have at most k parents?

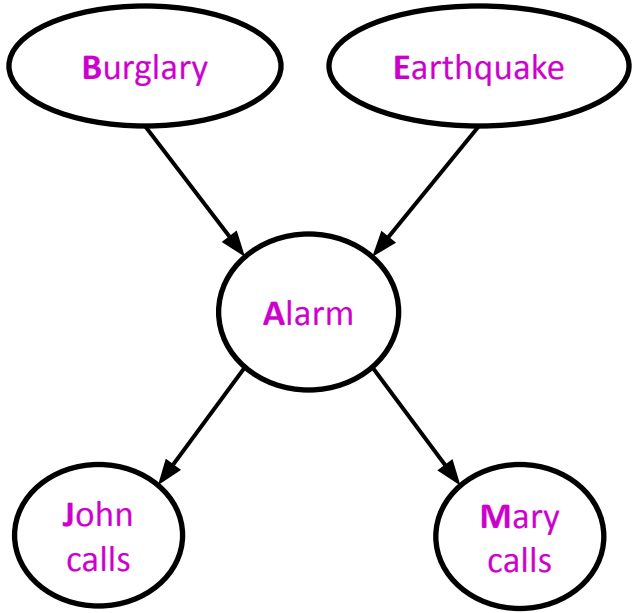
  $O(N * d^k)$

- Both give you the power to calculate $P(X_1, X_2, \ldots, X_N)$
- Bayes Nets: huge space savings with sparsity!
- Also easier to elicit local CPTs
- Also faster to answer queries (coming)





Recovering joint distribution from Bayes Net



P(B)	
true	false
0.001	0.999

P(E)	
true	false
0.002	0.998

A	P(J|A)	
	true	false
true	0.9	0.1
false	0.05	0.95

B	E	P(A|B,E)	
		true	false
true	true	0.95	0.05
true	false	0.94	0.06
false	true	0.29	0.71
false	false	0.001	0.999

A	P(M|A)	
	true	false
true	0.7	0.3
false	0.01	0.99

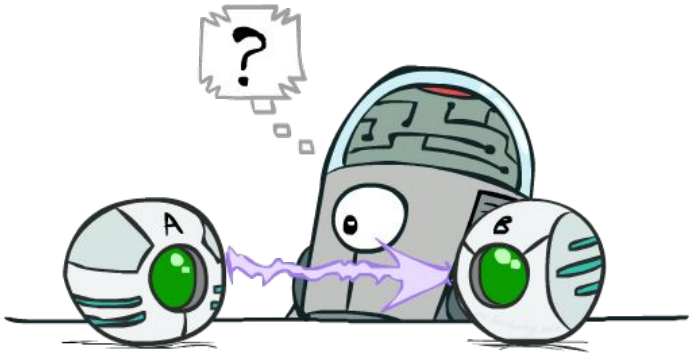
$P(b,\neg e, a, \neg j, \neg m) =$

$P(b)\ P(\neg e)\ P(a|b,\neg e)\ P(\neg j|a)\ P(\neg m|a)$

$=.001x.998x.94x.1x.3=.000028$

Causal structure in a Bayes Net is ideal, but optional

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

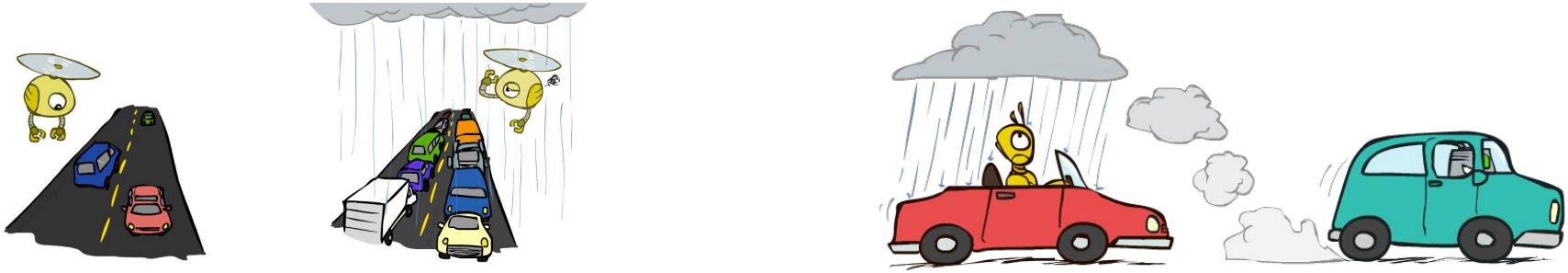


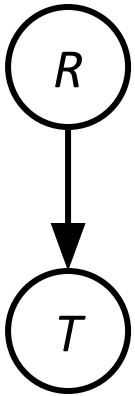
- BNs need not actually be causal
  - Sometimes no causal net exists over the domain (especially if variables are missing)
  - E.g. consider the variables *Traffic* and *Rain*
  - End up with arrows that reflect correlation, not causation

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

$$P(x_i|x_1, \ldots x_{i-1}) = P(x_i|parents(X_i))$$

Example: Traffic





$P(R)$

+r	1/4
-r	3/4

$P(T|R)$

+r	+t	3/4
	-t	1/4
-r	+t	1/2
	-t	1/2

$P(T,R)$

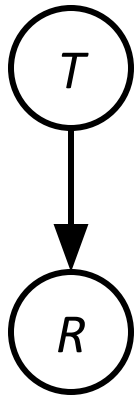
+r	+t	3/16
+r	-t	1/16
-r	+t	6/16
-r	-t	6/16

$P(T)$

+t	9/16
-t	7/16

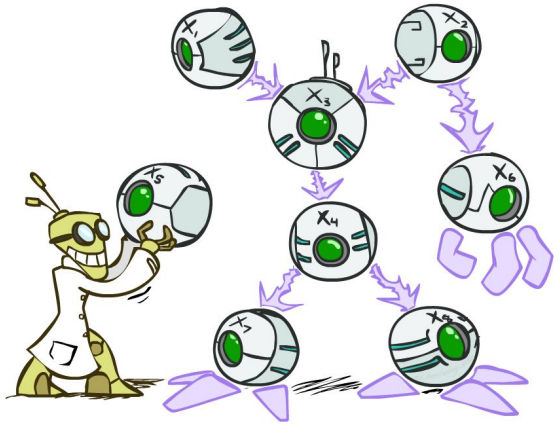
$P(R|T)$

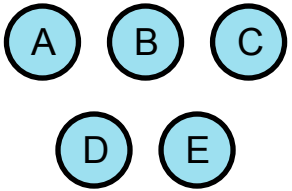
+t	+r	1/3
	-r	2/3
-t	+r	1/7
	-r	6/7



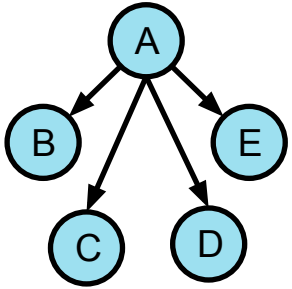
Summary

- Independence and conditional independence are important forms of probabilistic knowledge
- **Bayes net** encode **joint distributions** efficiently by taking advantage of **conditional independence**
  - Global joint probability = product of local conditionals
- Allows for flexible tradeoff between model accuracy and memory/compute efficiency

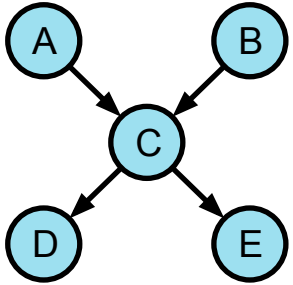




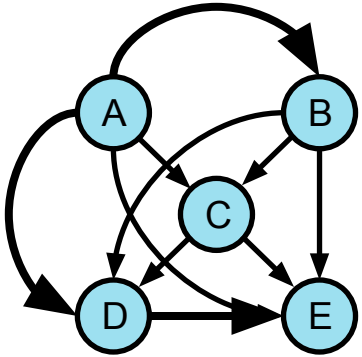
Strict Independence



Naïve Bayes

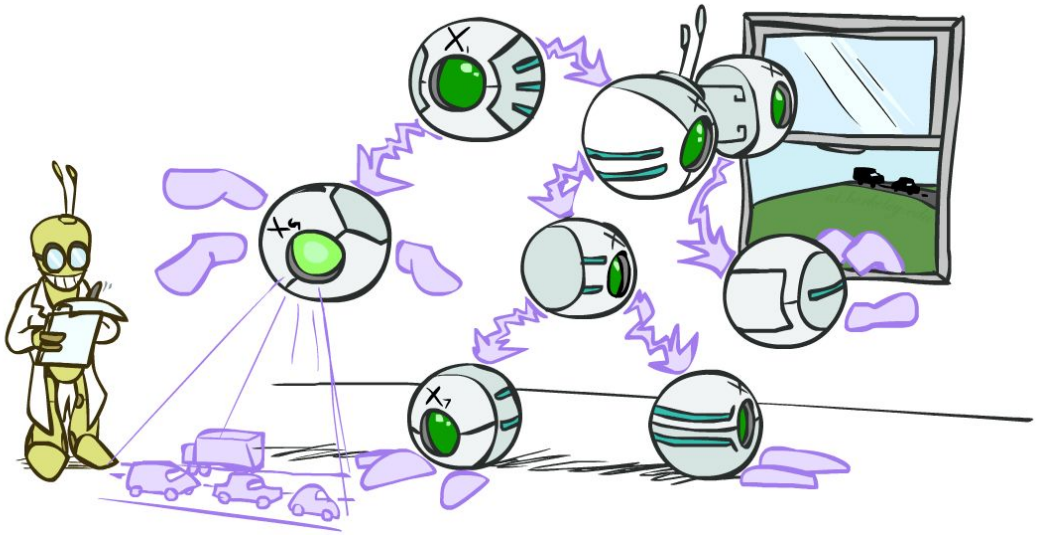


Sparse Bayes Net



Joint Distribution

Bayes Nets: Exact Inference



[These 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.]

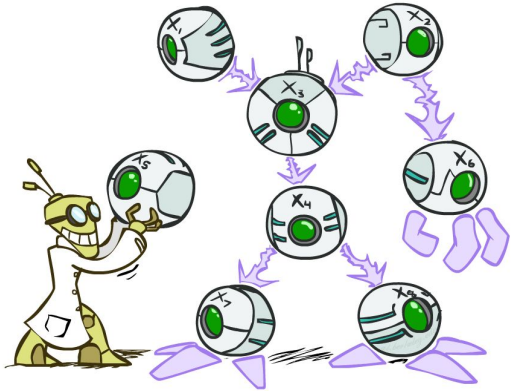
Bayes Net Representation

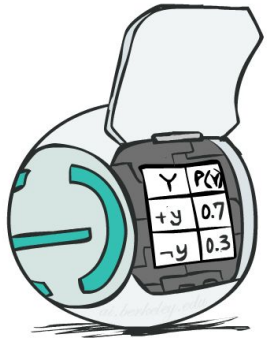
- A **directed**, **acyclic graph**, one node per **random variable**
- A **conditional probability** table (CPT) for each node
  - A collection of distributions over X, one for each combination of parents' values

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

- 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:

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





Inference

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

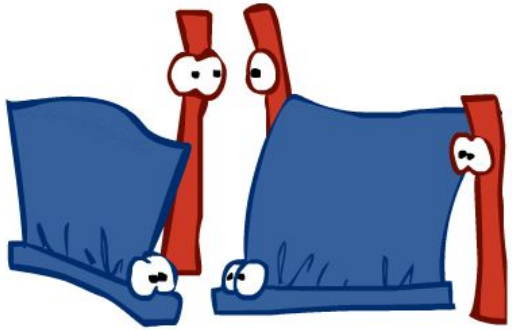
e.g. probability of spam/not spam given evidence

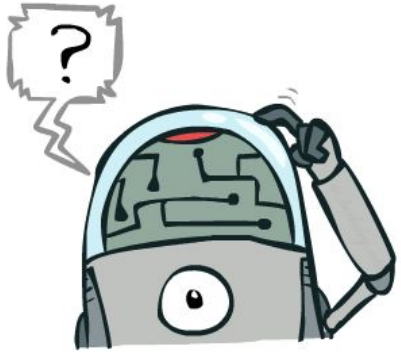
- Examples:
  - Posterior probability

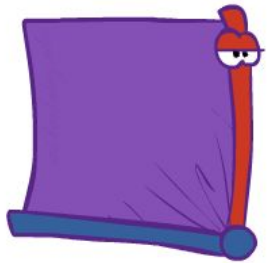
$$P(Q|E_1 = e_1, \ldots E_k = e_k)$$

  - Most likely explanation:

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







Inference by Enumeration

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

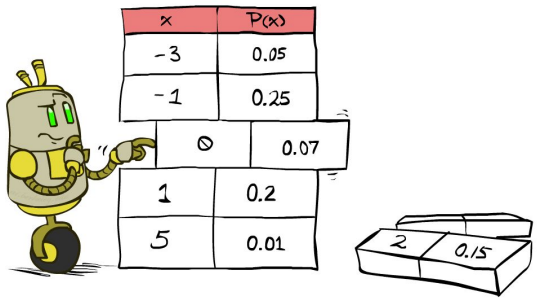
$X_1, X_2, \dots X_n$ *All variables*

** Works fine with multiple query variables, too*

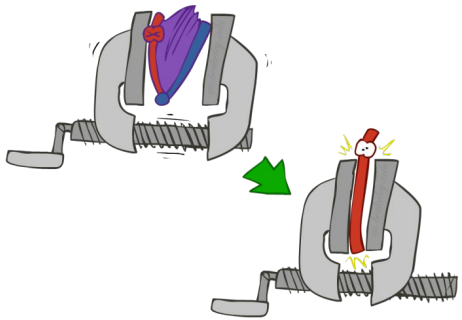
- We want:

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

- 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

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

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

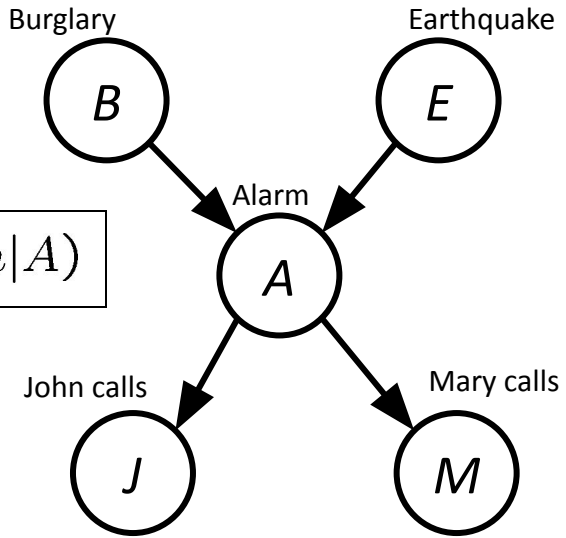
$$P(Q|e_1 \cdots e_k) = \frac{1}{Z} P(Q, e_1 \cdots e_k)$$

SFU

Inference by Enumeration using Bayes Nets

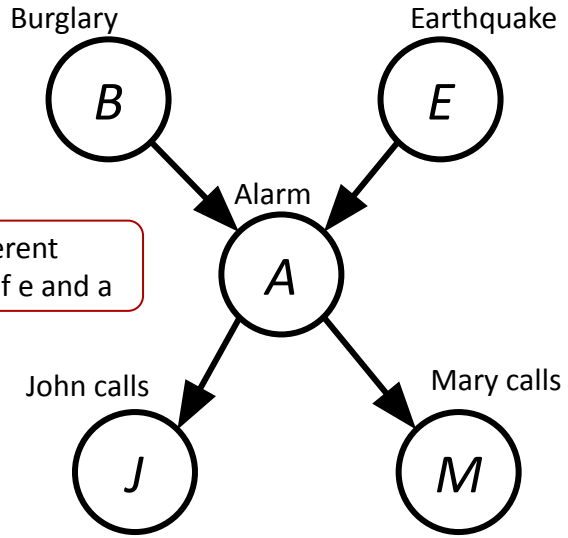
$$P(B|j,m) \propto P(B,j,m)$$

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



Inference by Enumeration using Bayes Nets

- Given unlimited time, inference in BNs is straightforward
- Inference by enumeration:



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

All the different combinations of e and a

Both John and Mary called me. What's the probability that there was a burglar?

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

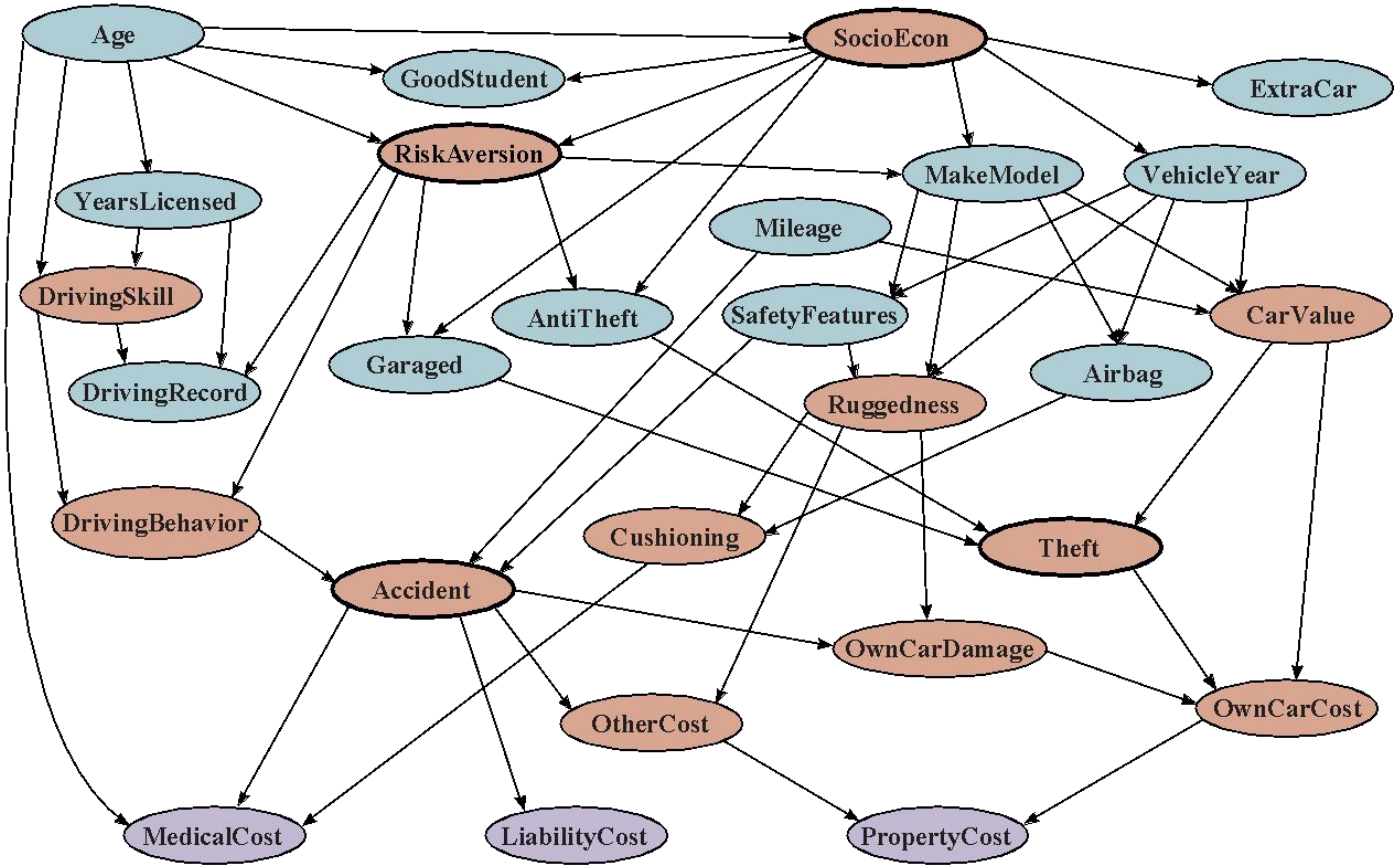
Text

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

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

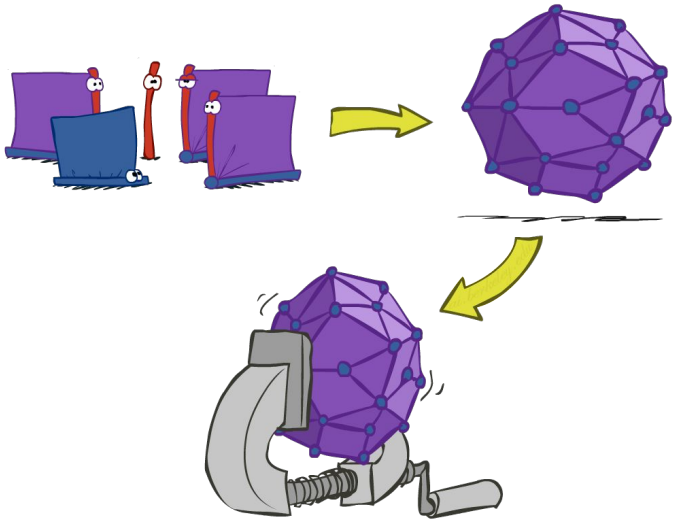
Lots of repeated subexpressions! Can we factor them out?

Inference by Enumeration?

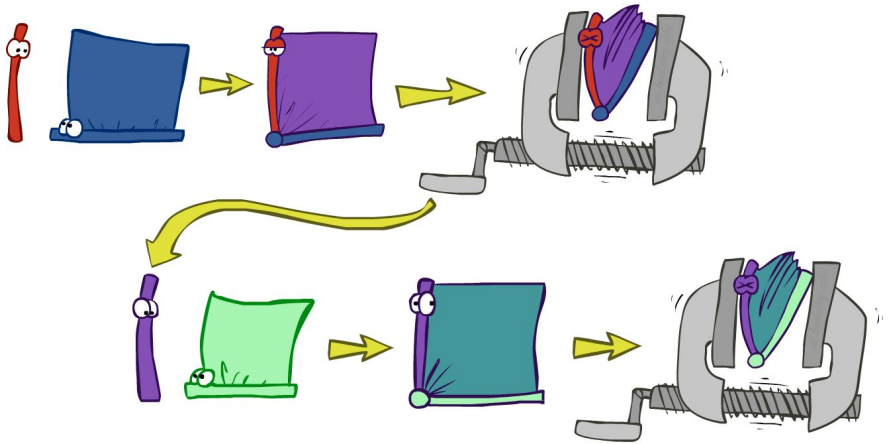


Inference by Enumeration vs. Variable Elimination

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



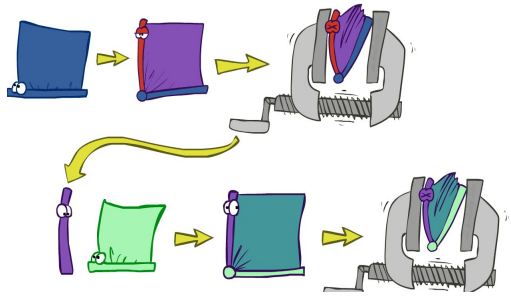
- Idea: interleave joining and marginalizing!
  - Called “Variable Elimination”
  - Still NP-hard, but usually much faster than inference by enumeration



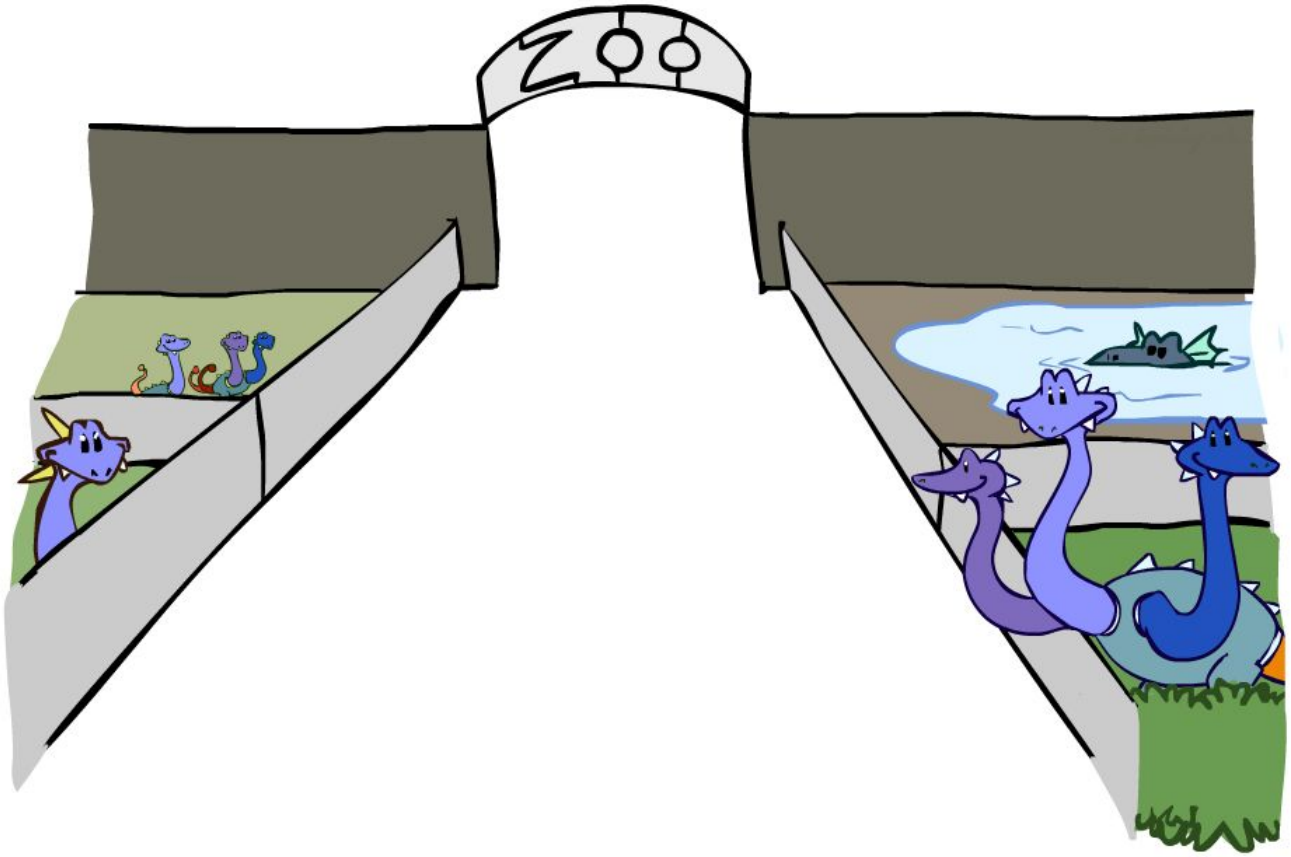
- First we’ll need some new notation: factors

Variable elimination: The basic ideas

- Consider: **uwy + uwz + uxy + uxz + vwy + vwz + vxy +vxz**
  - 16 multiplies, 7 adds
- Rewrite as: **(u+v)(w+x)(y+z)**
  - 2 multiplies, 3 adds
- Move summations inwards as far as possible
  - $P(B \mid j, m) = \alpha \sum_{e,a} P(B)\, P(e)\, P(a|B,e)\, P(j|a)\, P(m|a)$
  - $= \alpha\, P(B) \sum_{e} P(e) \sum_{a} P(a|B,e)\, P(j|a)\, P(m|a)$
- Do the calculation from the inside out
  - i.e., sum over *a* first, then sum over *e*



Factor Zoo: 5 Types of Factors



Factor Zoo I

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

Dimensionality of 2

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

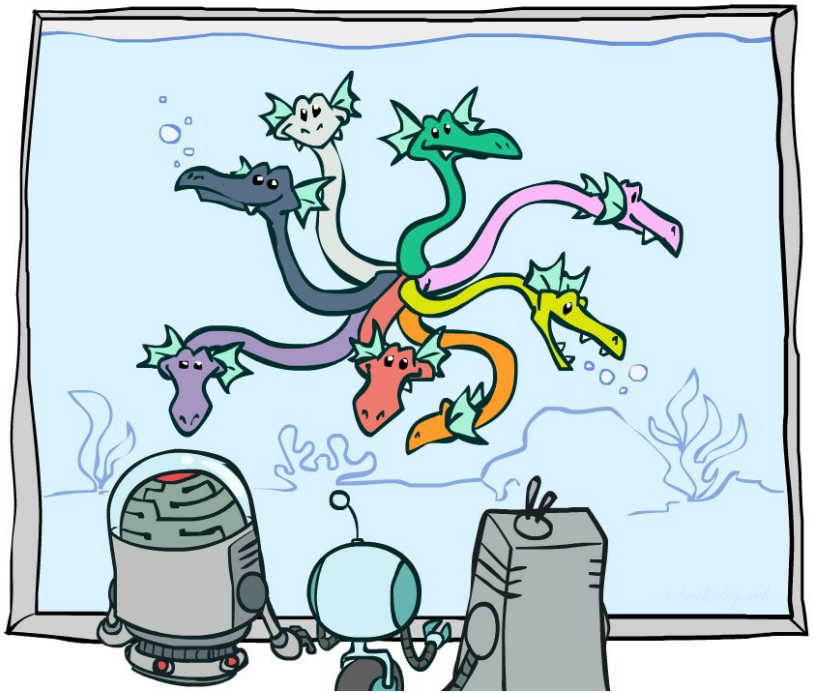
- Number of capitals = dimensionality of the table

$P(T, W)$

	W	
T	sun	rain
hot	0.4	0.1
cold	0.2	0.3

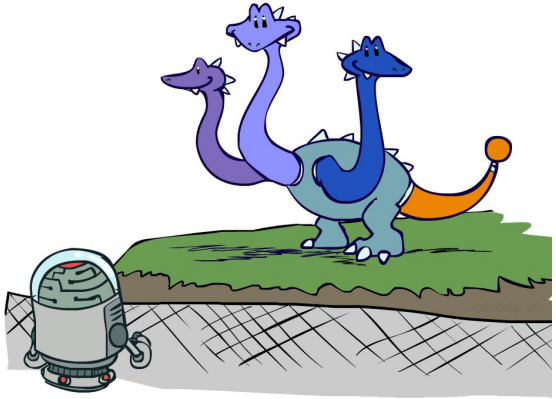
$P(cold, W)$

	W	
T	sun	rain
cold	0.2	0.3



Factor Zoo II

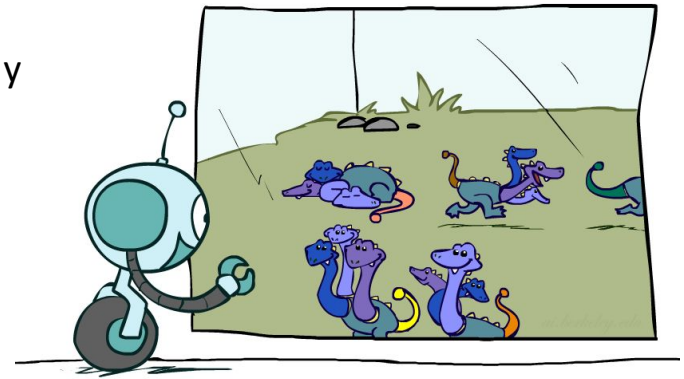
3) **Single conditional**: **P(Y | x)**
- Entries P(y | x) for fixed x, all y
- Sums to 1



$P(W|cold)$

	W	
T	sun	rain
cold	0.4	0.6

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



$P(W|T)$

	W	
T	sun	rain
hot	0.8	0.2
cold	0.4	0.6

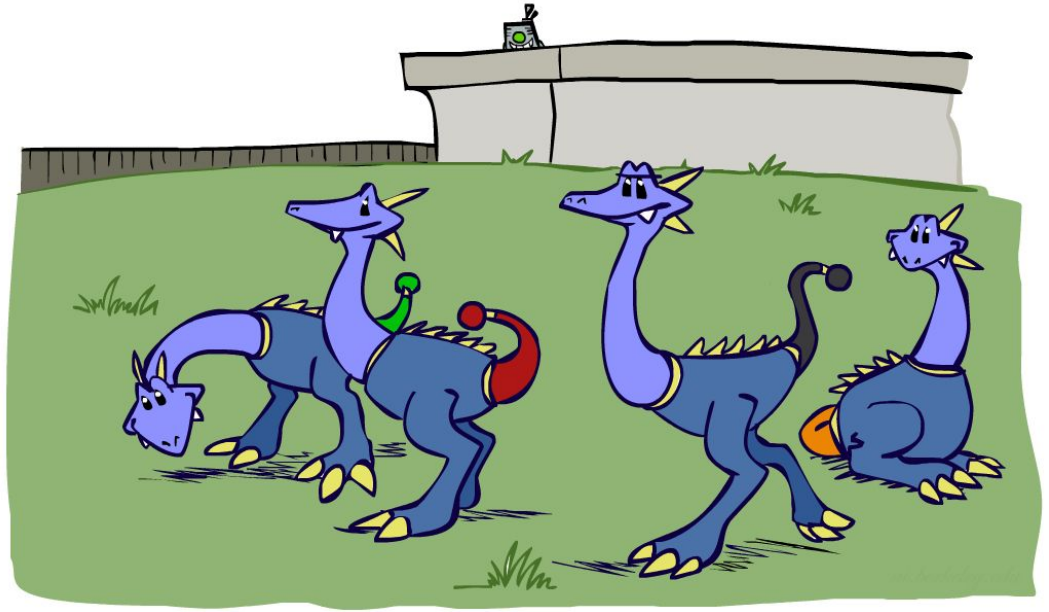
$P(W|hot)$

$P(W|cold)$

Factor Zoo III

5) **Specified family**: **P(y | X)**

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



$P(rain|T)$

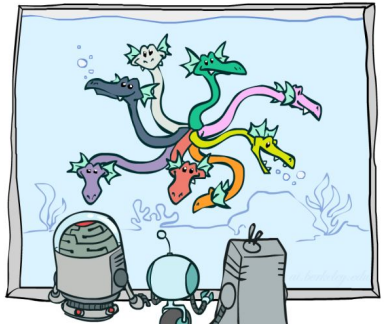
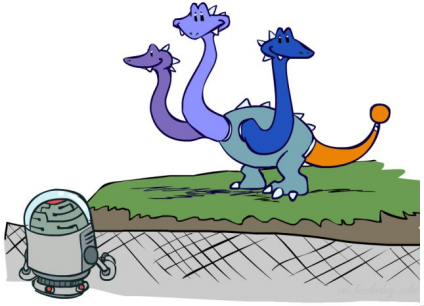
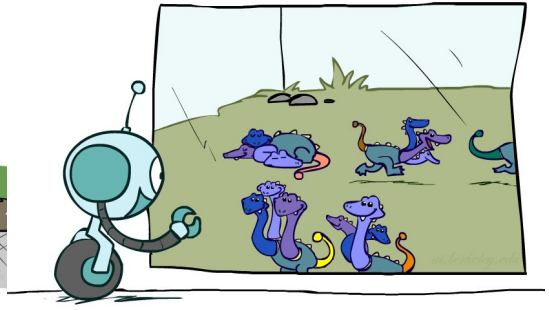
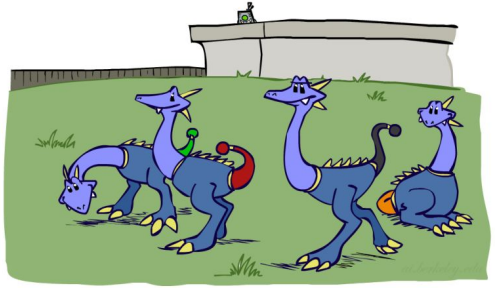
	W
T	rain
hot	0.2
cold	0.6

$P(rain|hot)$

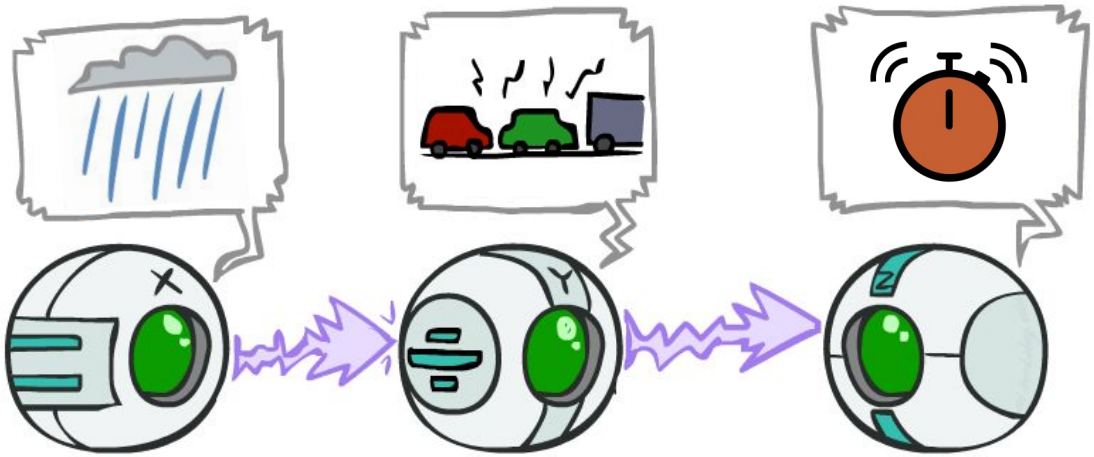
$P(rain|cold)$

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
  - Sometimes we'll write P(A,b|c,D) as $f_i(A,b,c,D)$—just another name for the same table.

Traffic Domain



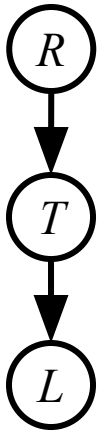
Raining

Traffic

Late for class!

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

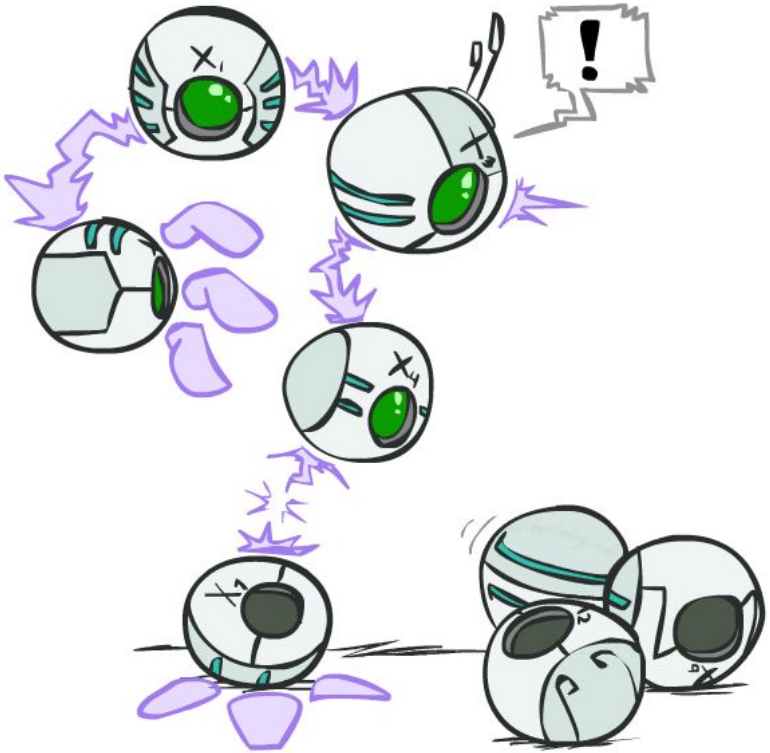
$P(T|R)$

+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

Variable Elimination (VE)



Inference by Enumeration: Procedure

- 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

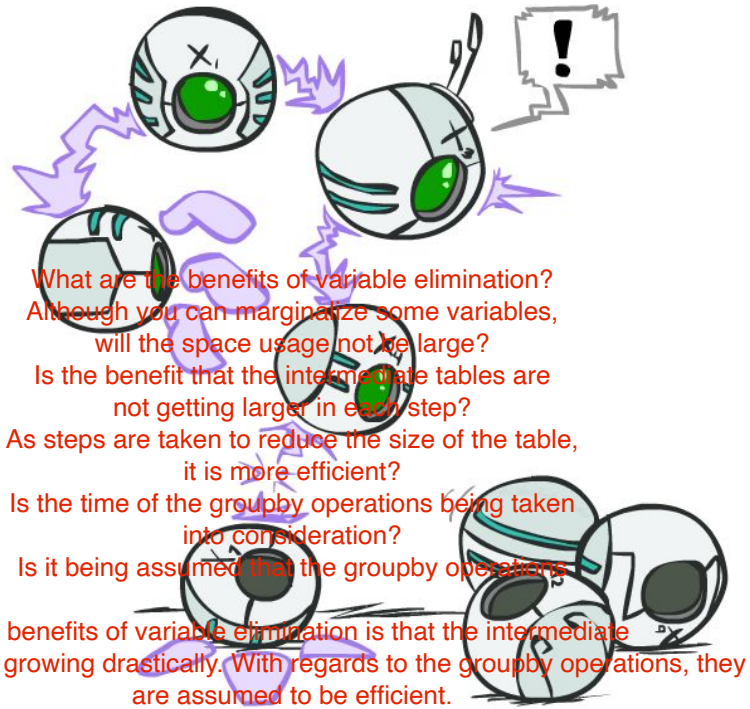
$P(T|R)$

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

$P(+\ell|T)$

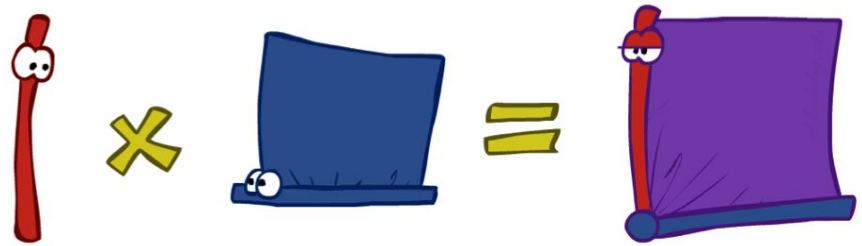
+t	+l	0.3
-t	+l	0.1

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

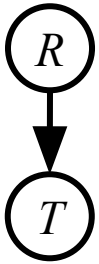


Operation 1: Join Factors

- 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

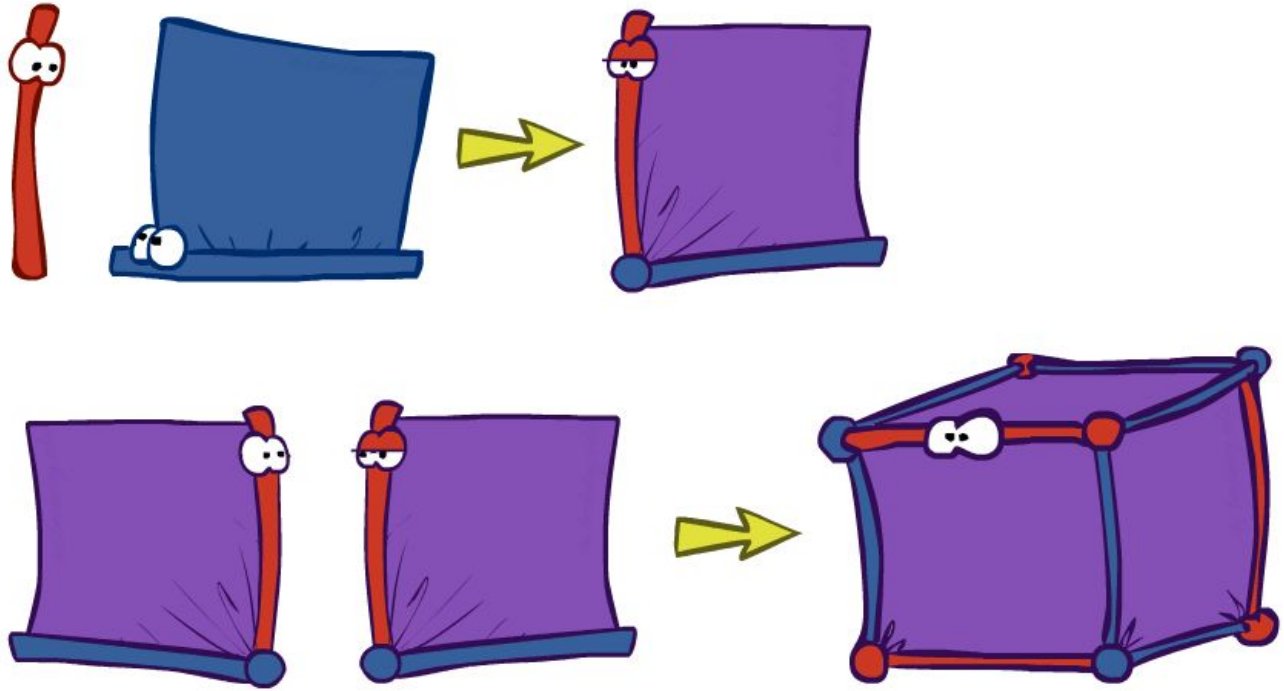
+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

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

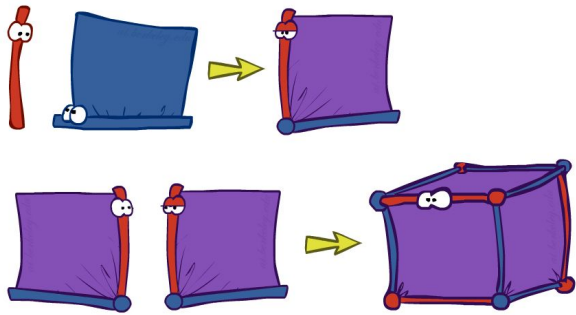
  - Computation for each entry: pointwise products

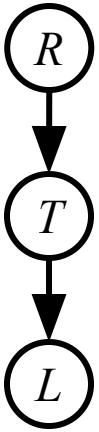
$$\forall r, t: \quad P(r,t) = P(r) \cdot P(t|r)$$

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

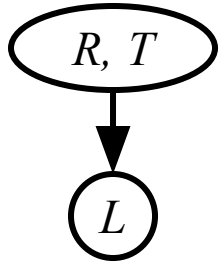
$P(L|T)$

+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
  - A projection operation

Should this not be a group-by operation?

- Example:

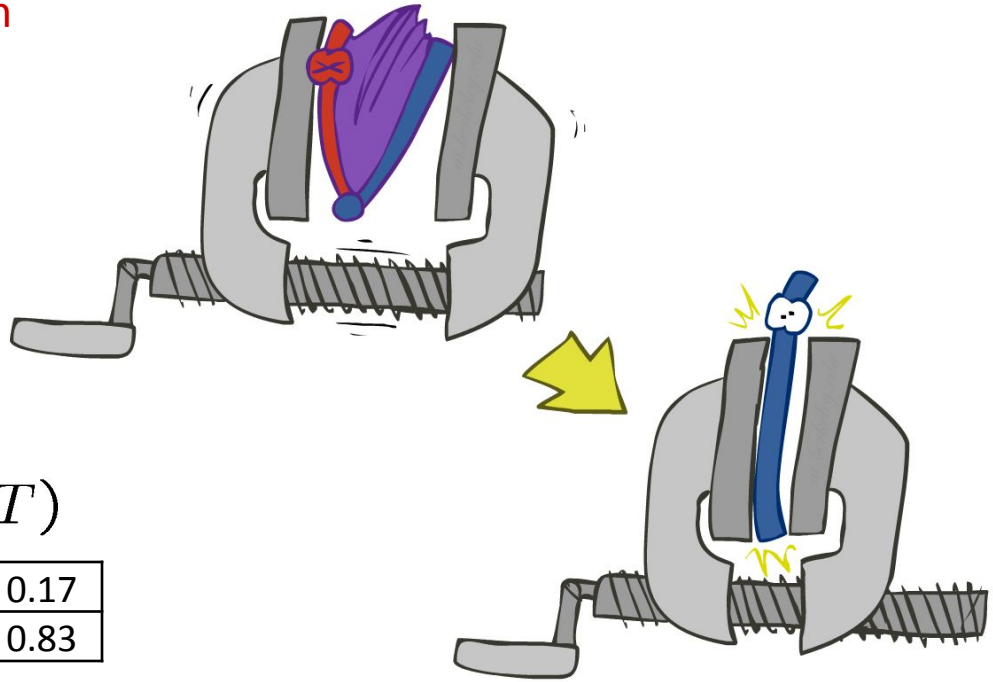
$P(R,T)$

+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

$P(R, T, L)$

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

$P(T, L)$

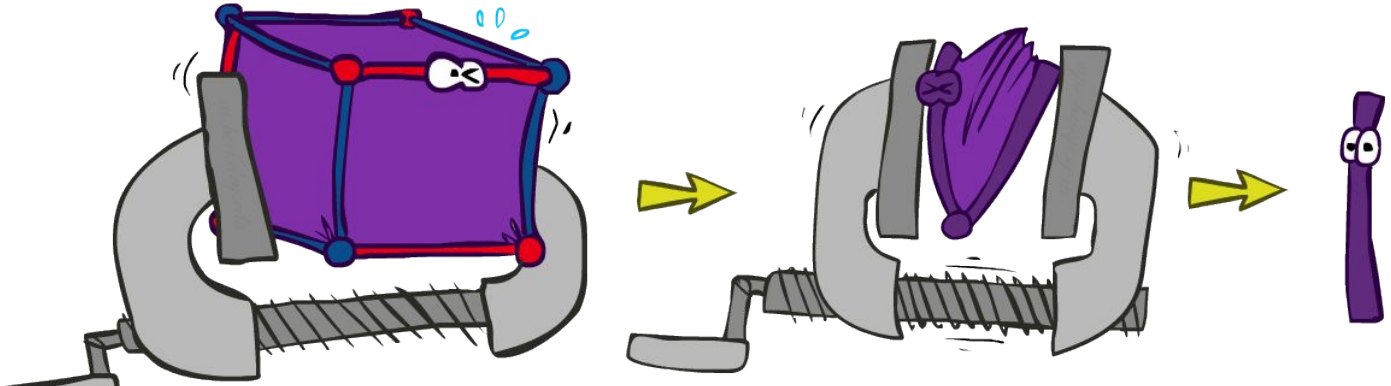
+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

Sum out T

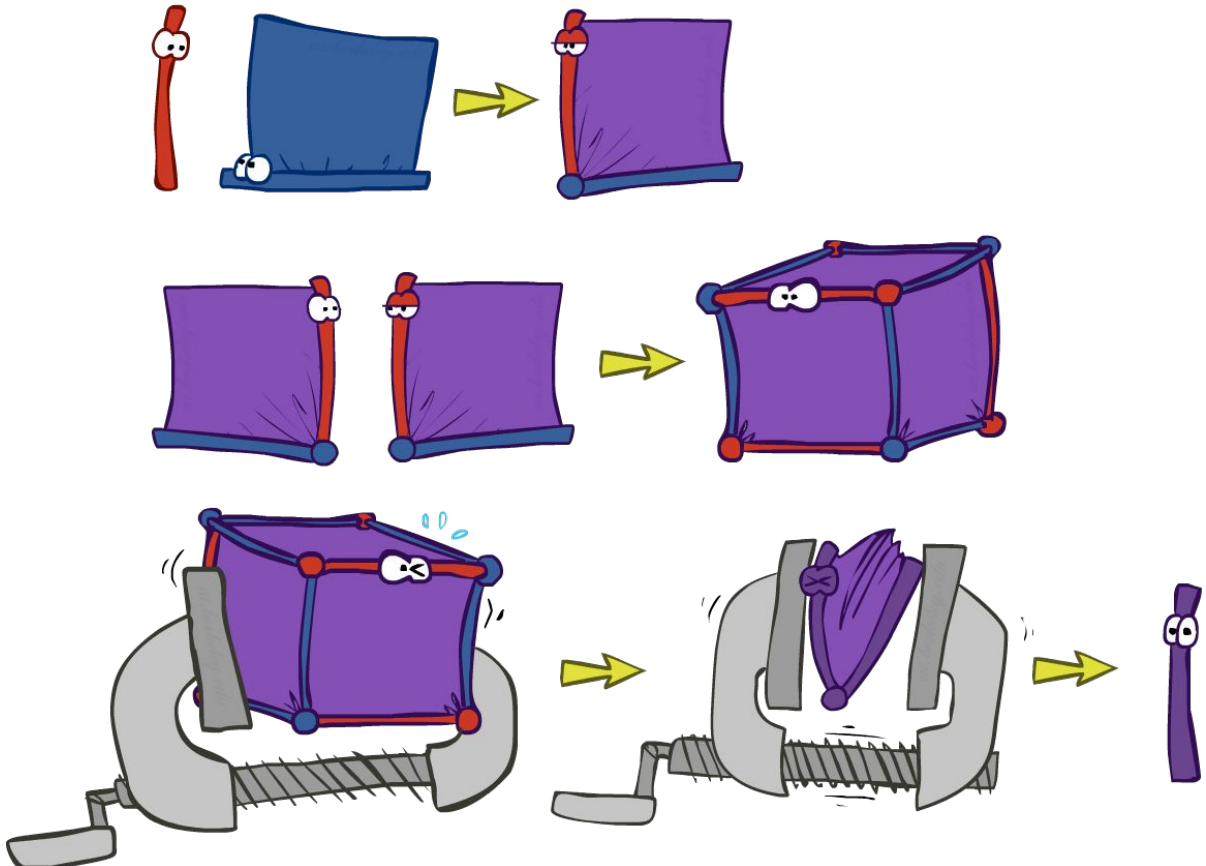
L

$P(L)$

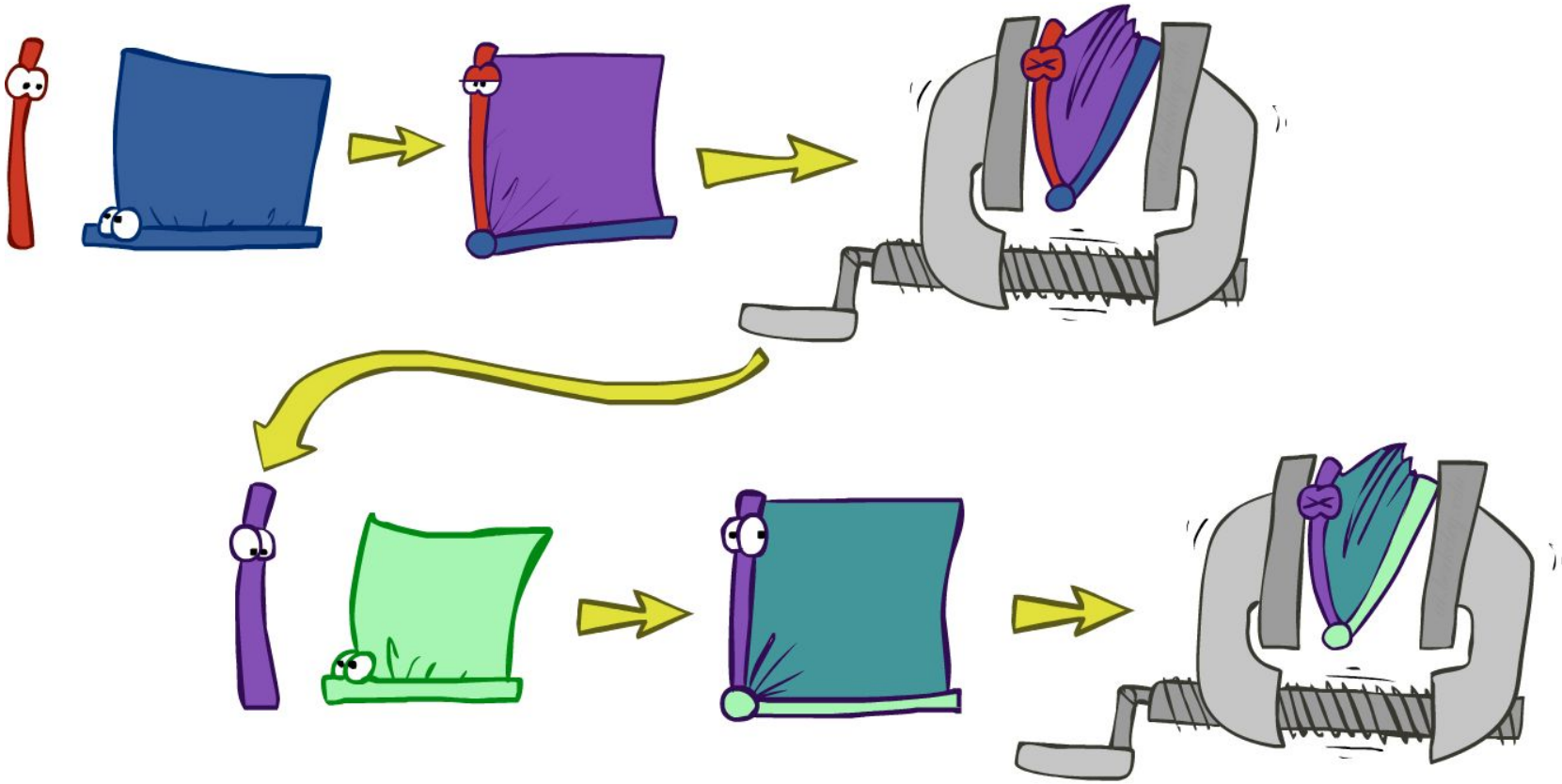
+l	0.134
-l	0.866



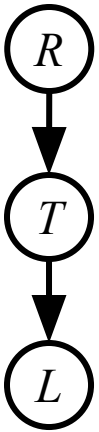
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)



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

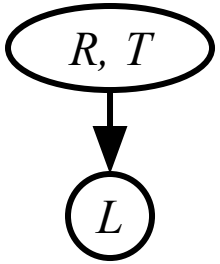
$P(L|T)$

+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



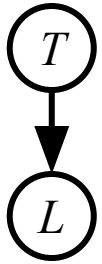
$P(L|T)$

+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

When we sum out a variable, we are left with a table with two rows. When we join that table with another table, can we upper bound the size of the table being joined by the size of the largest table that we begin with?

$P(T,L)$

+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	0.866

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:

$P(+r)$

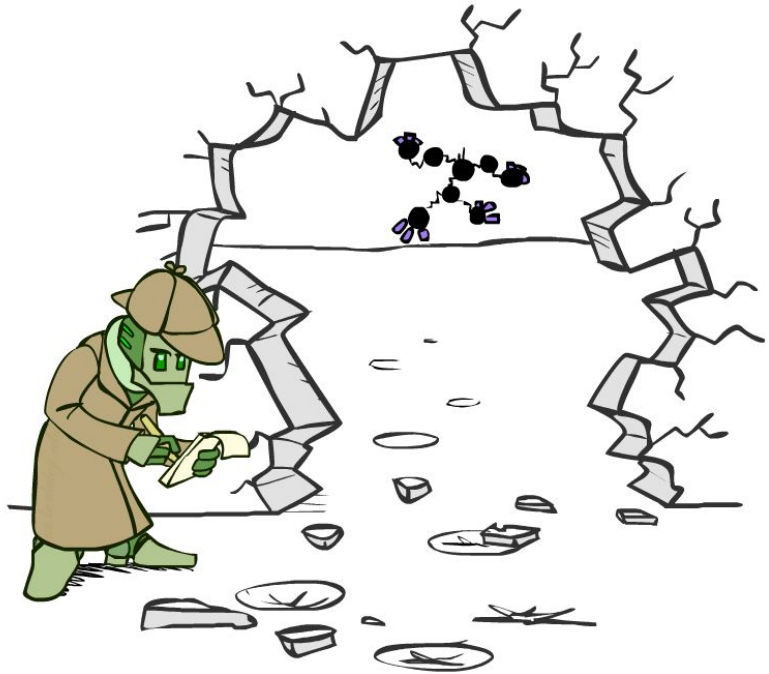
+r	0.1

$P(T|+r)$

+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:

$P(+r, L)$

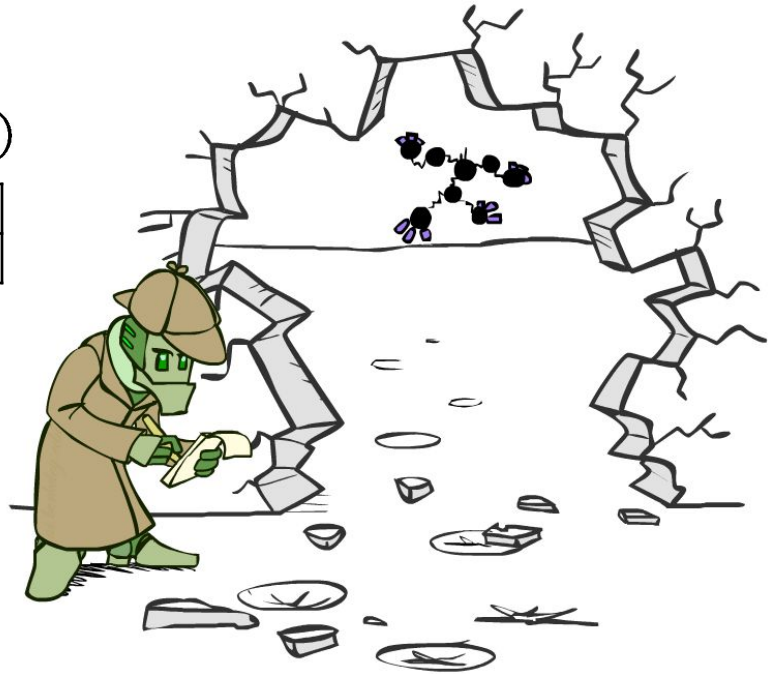
+r	+l	0.026
+r	-l	0.074

Normalize

$P(L|+r)$

+l	0.26
-l	0.74

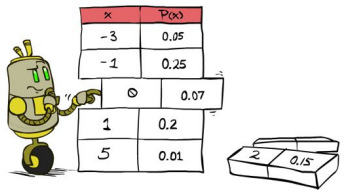
- To get our answer, just normalize this!
- That's it!



General Variable Elimination

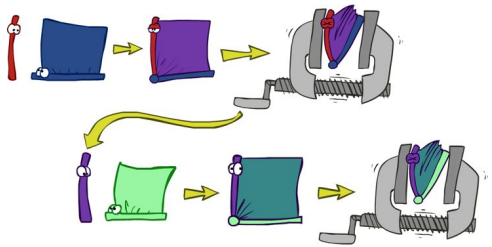
Will not be on the exam

- 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



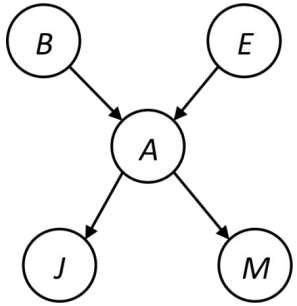
- Join all remaining factors and normalize

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

Example

Text

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



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

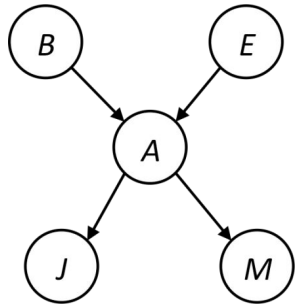
Choose A to eliminate (could also have chosen E)

$$P(A|B,E) \\ P(j|A) \\ P(m|A) \quad \Rightarrow^{\times} \quad P(j,m,A|B,E) \quad \Rightarrow^{\Sigma} \quad P(j,m|B,E)$$

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

Example

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



Choose E

$P(E)$

$P(j, m|B, E)$ $\Rightarrow \times$ $P(j, m, E|B)$ $\Rightarrow \sum$ $P(j, m|B)$

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

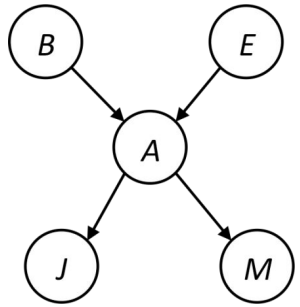
Finish with B

$P(B)$

$P(j, m|B)$ $\Rightarrow \times$ $P(j, m, B)$ $\Rightarrow$ Normalize $P(B|j, m)$

Same Example in Equations

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

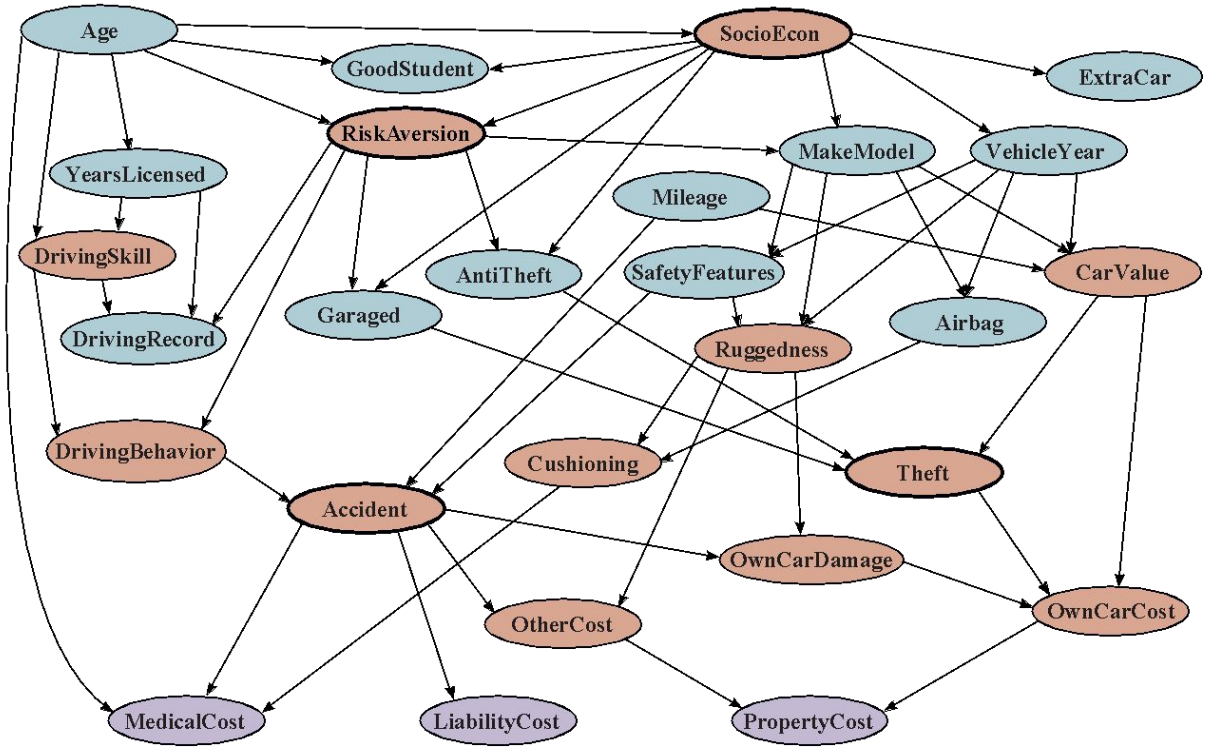


$P(B)$	$P(E)$	$P(A\|B,E)$	$P(j\|A)$	$P(m\|A)$

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

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

Example Bayes Net: Car Insurance



Enumeration: **227M** operations

Elimination: **221K** operations

Summary

- Exact inference = sums of products of conditional probabilities from the network
- Enumeration is always exponential
- Variable elimination reduces this by avoiding the recomputation of repeated subexpressions
  - Massive speedups in practice
- Exact inference is #P-hard

Maxwell Libbrecht?

Approximation methods exist