复旦大学大数据学院
School of Data Science, Fudan University
魏忠钰

Bayes' Nets: Inference

May 10th, 2017

Bayes' Nets

- Representation
- Conditional Independences
- Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Sampling (approximate)

Inference

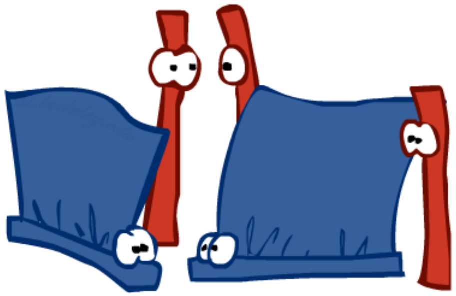
- Inference: calculating some useful quantity from a joint probability distribution

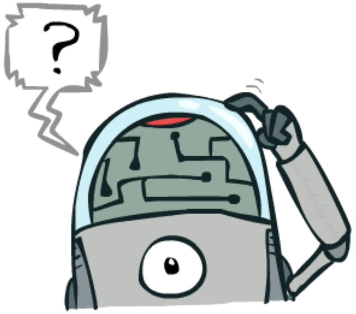
- Examples:
  - Posterior probability

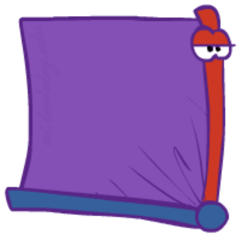
$$P(Q|E_1 = e_1, \ldots E_k = e_k)$$

  - Most likely explanation:

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







Inference by Enumeration

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

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

All variables

- 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
- Step 3: Normalize

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

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

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

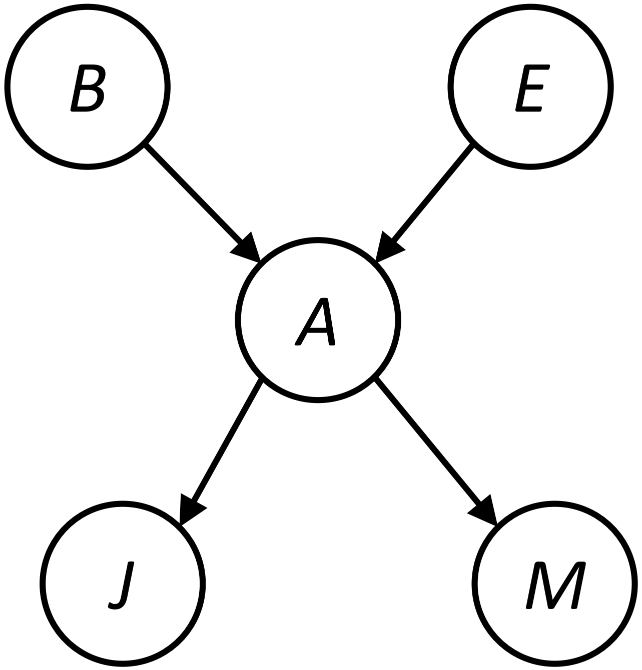
Inference by Enumeration in Bayes' Net

- Given unlimited time, inference in BNs is easy

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

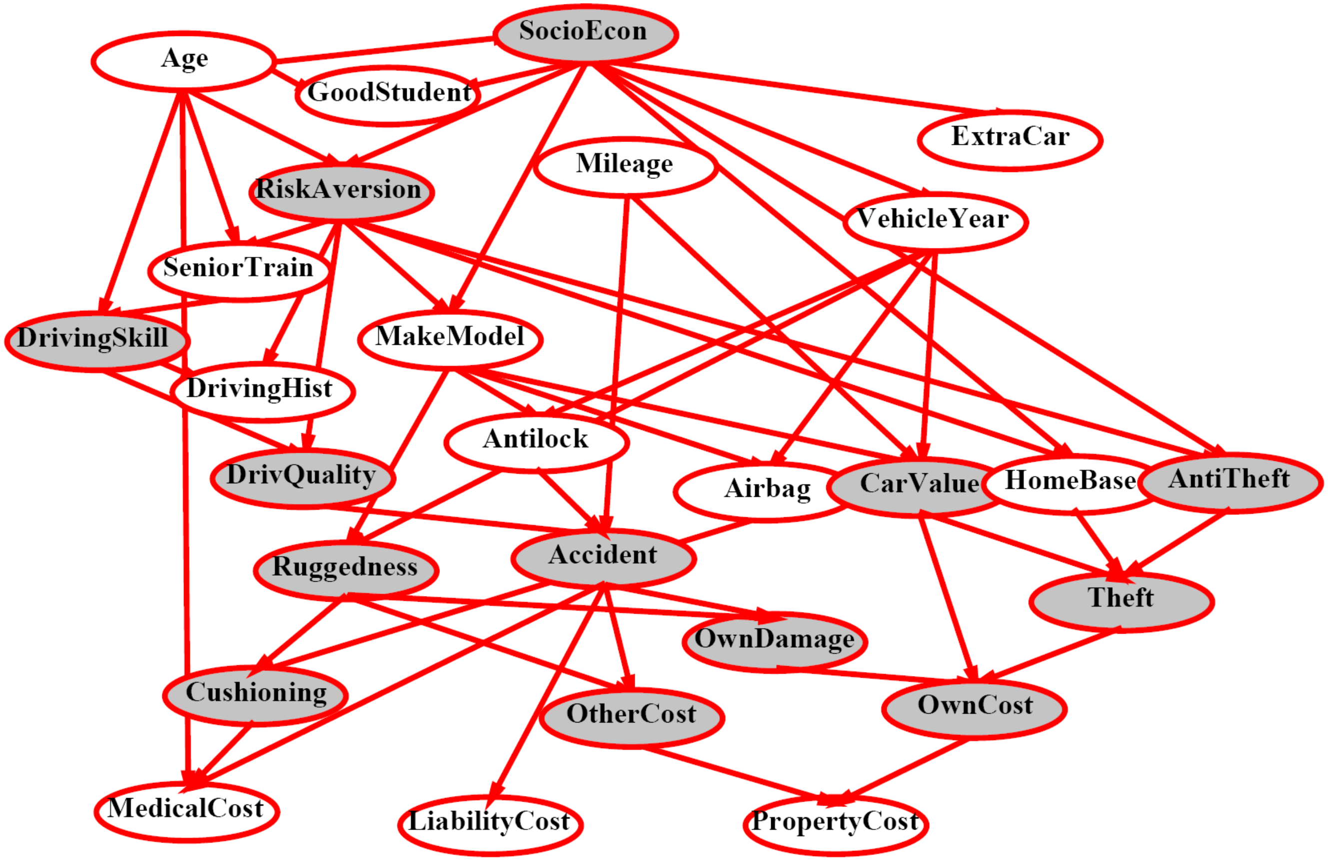
$$= \sum_{e,a} P(B, e, a, +j, +m)$$

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



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

Inference by Enumeration?

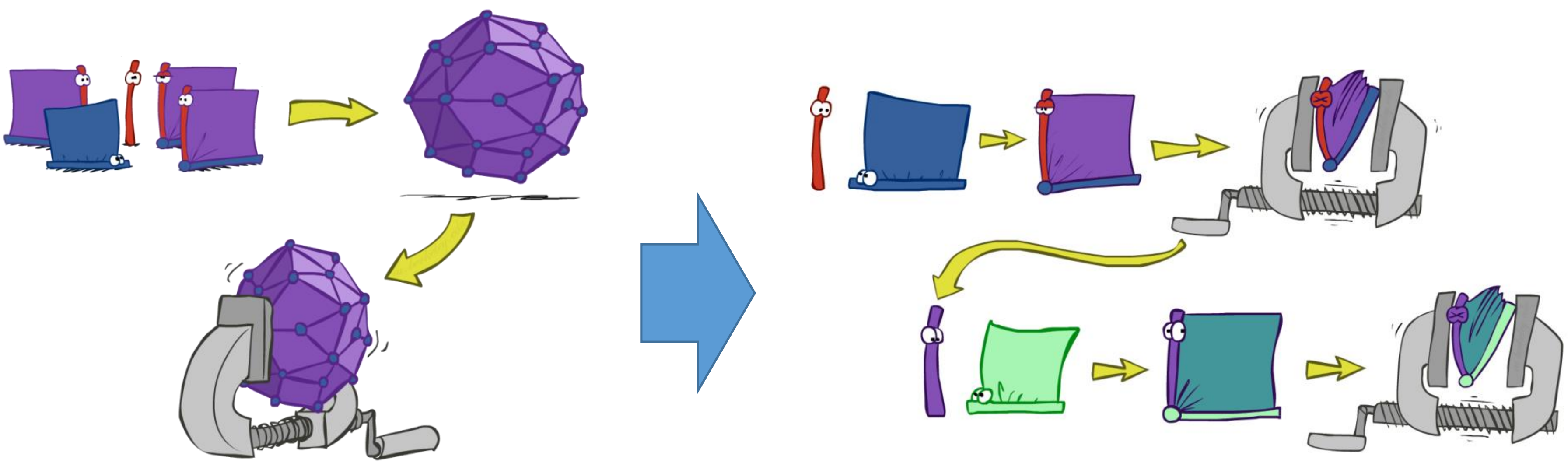


$$P(Antilock|observed\ variables) = ?$$

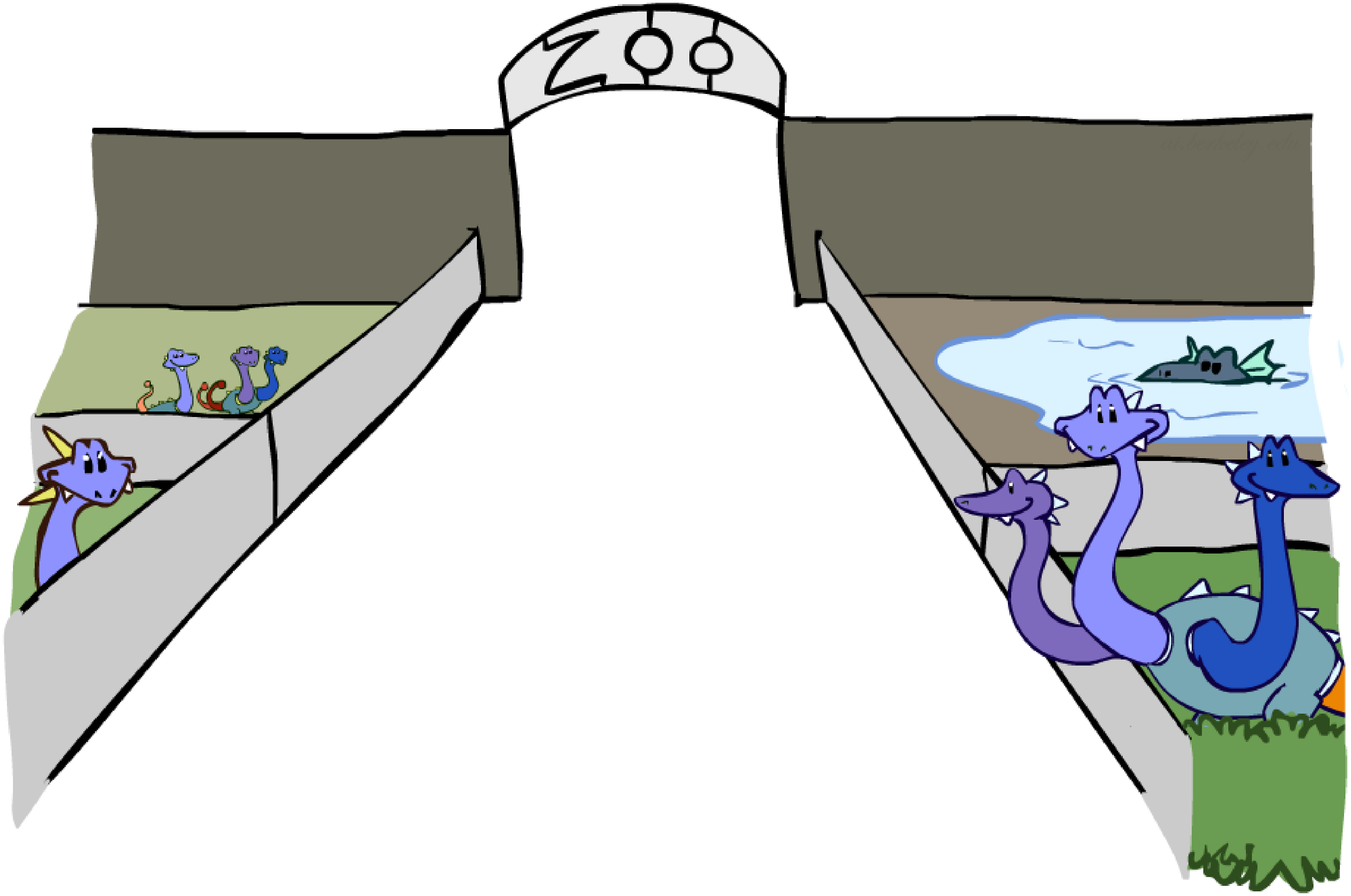
Inference by Enumeration vs. Variable Elimination

- 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"
  - Same computation complexity, 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 y
  - Sums to 1

$P(W|cold)$

T	W	P
cold	sun	0.4
cold	rain	0.6

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

$P(W|T)$

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

$P(W|hot)$

$P(W|cold)$

Factor Zoo III

- Specified family: P(y | X)
  - Entries P(y | x) for fixed y, but for all x
  - Sums to ?

$$P(rain|T)$$

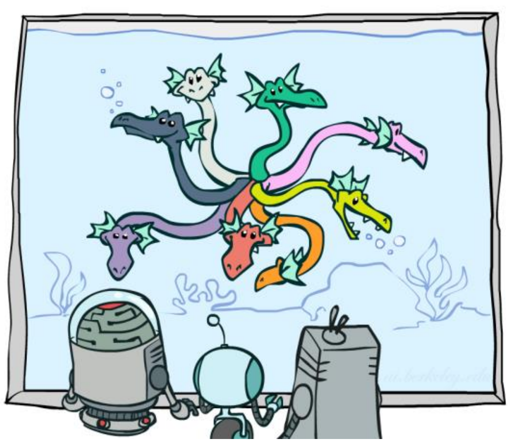
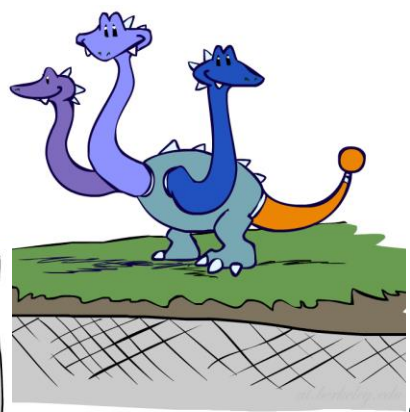
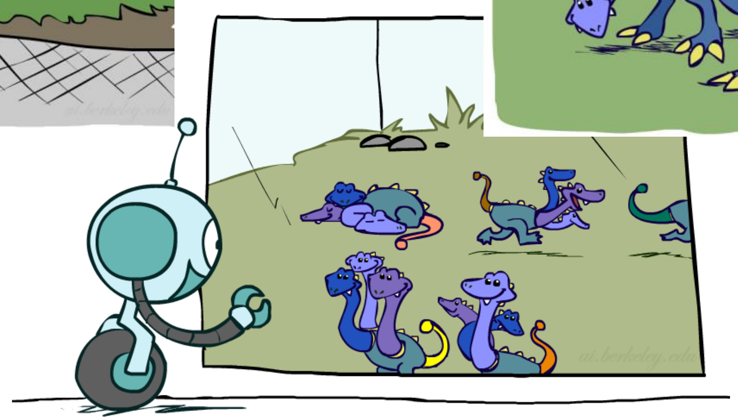
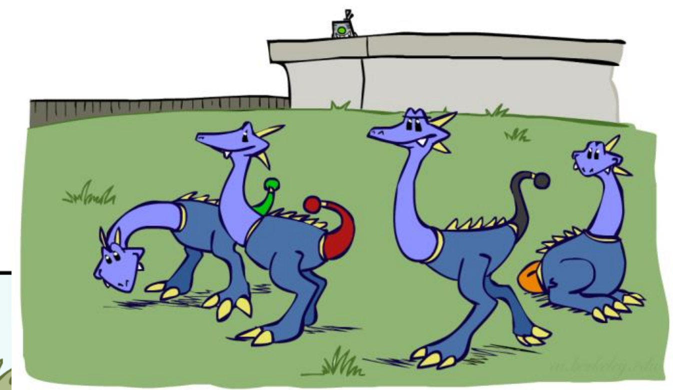
T	W	P
hot	rain	0.2
cold	rain	0.6

$$P(rain|hot)$$

$$P(rain|cold)$$

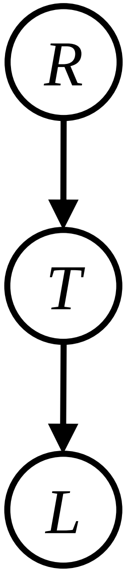
Factor Zoo Summary

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

Example: Traffic Domain

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



$P(R)$

+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

$$P(L) = ?$$

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

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

Inference by Enumeration: Procedural Outline

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

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

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

$P(L|T)$

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

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

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

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

$P(+\ell|T)$

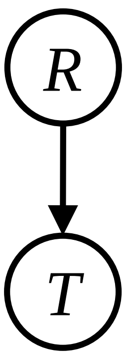
+t	+l	0.3
-t	+l	0.1

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

Operation 1: Join Factors

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

- Example: Join on R

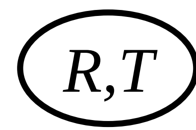


$$P(R) \times P(T|R) \rightarrow P(R,T)$$

+r	0.1
-r	0.9

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

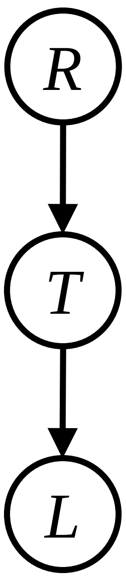
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81



  - Computation for each entry: pointwise products

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

Example: Multiple Joins



$P(R)$

+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

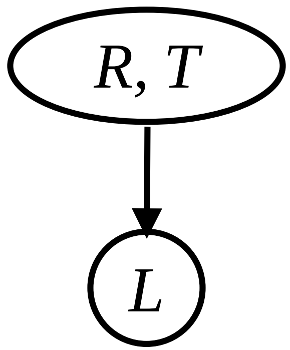
$P(L|T)$

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

Join R

$P(R,T)$

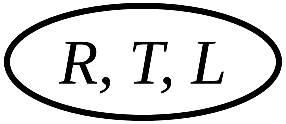
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81



$P(L|T)$

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

Join T

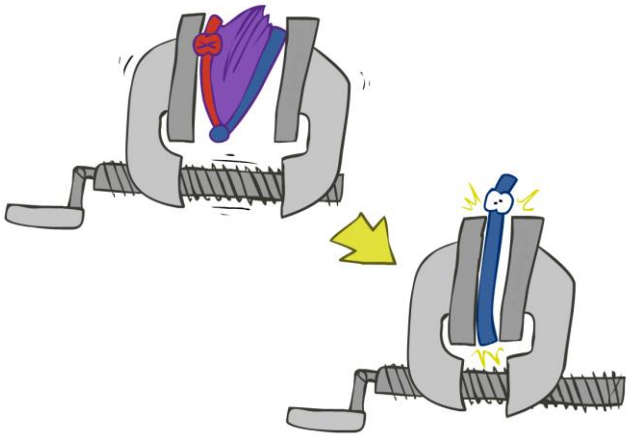


$P(R,T,L)$

+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

$\xrightarrow{\text{sum } R}$

$P(T)$

+t	0.17
-t	0.83

Multiple Elimination

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

$P(R, T, L)$

Sum out R

T, L

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

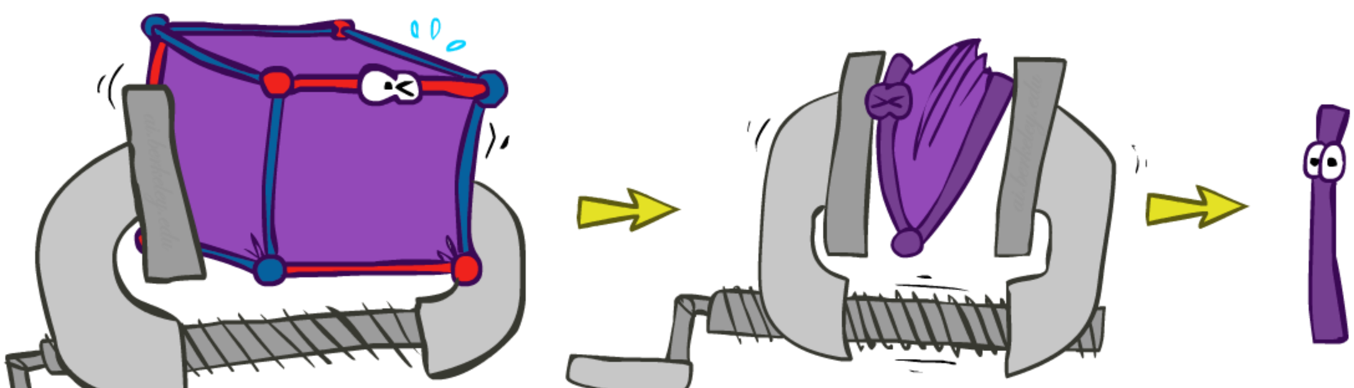
$P(T, L)$

Sum out T

L

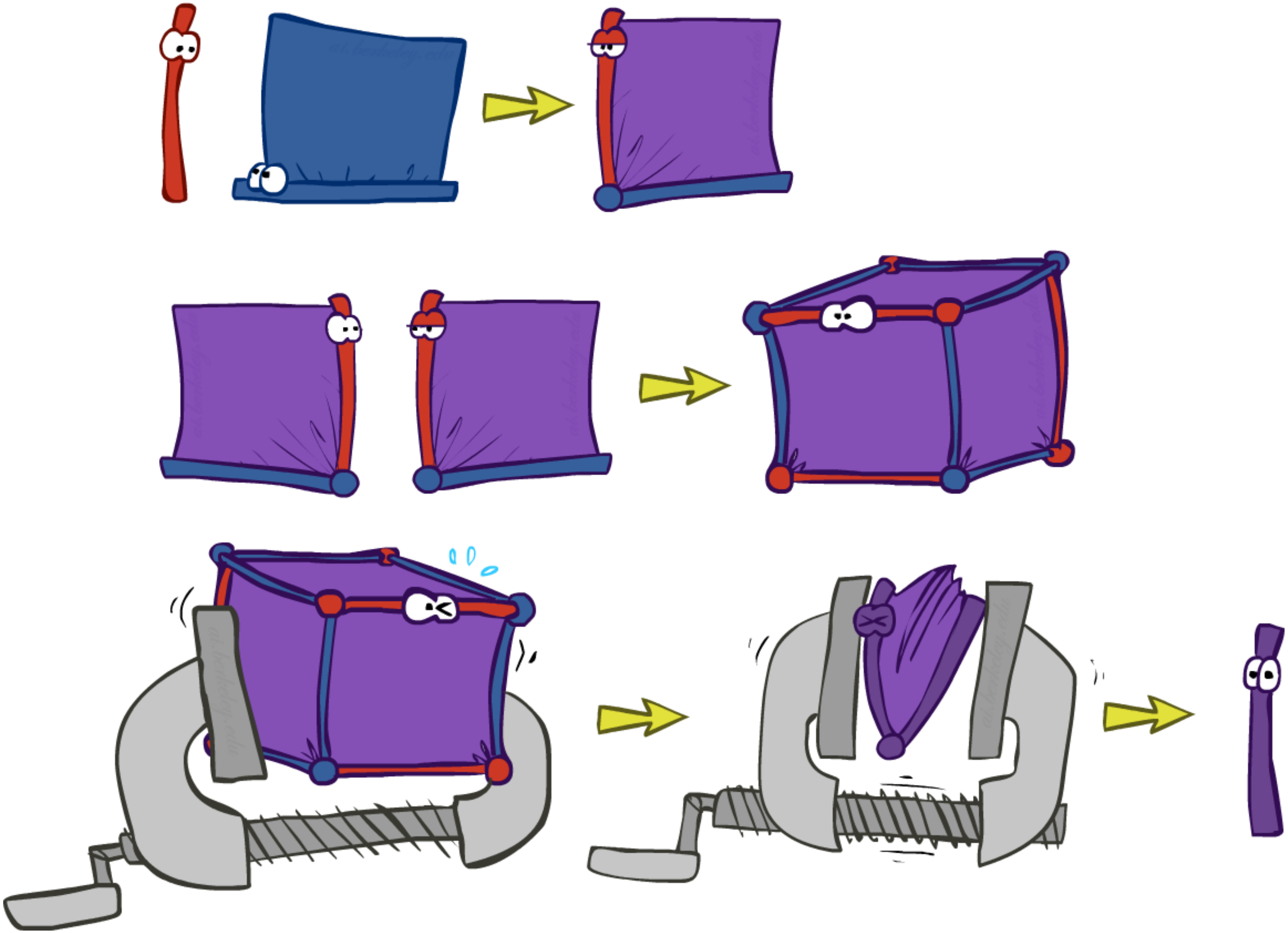
+l	0.134
-l	0.886

$P(L)$

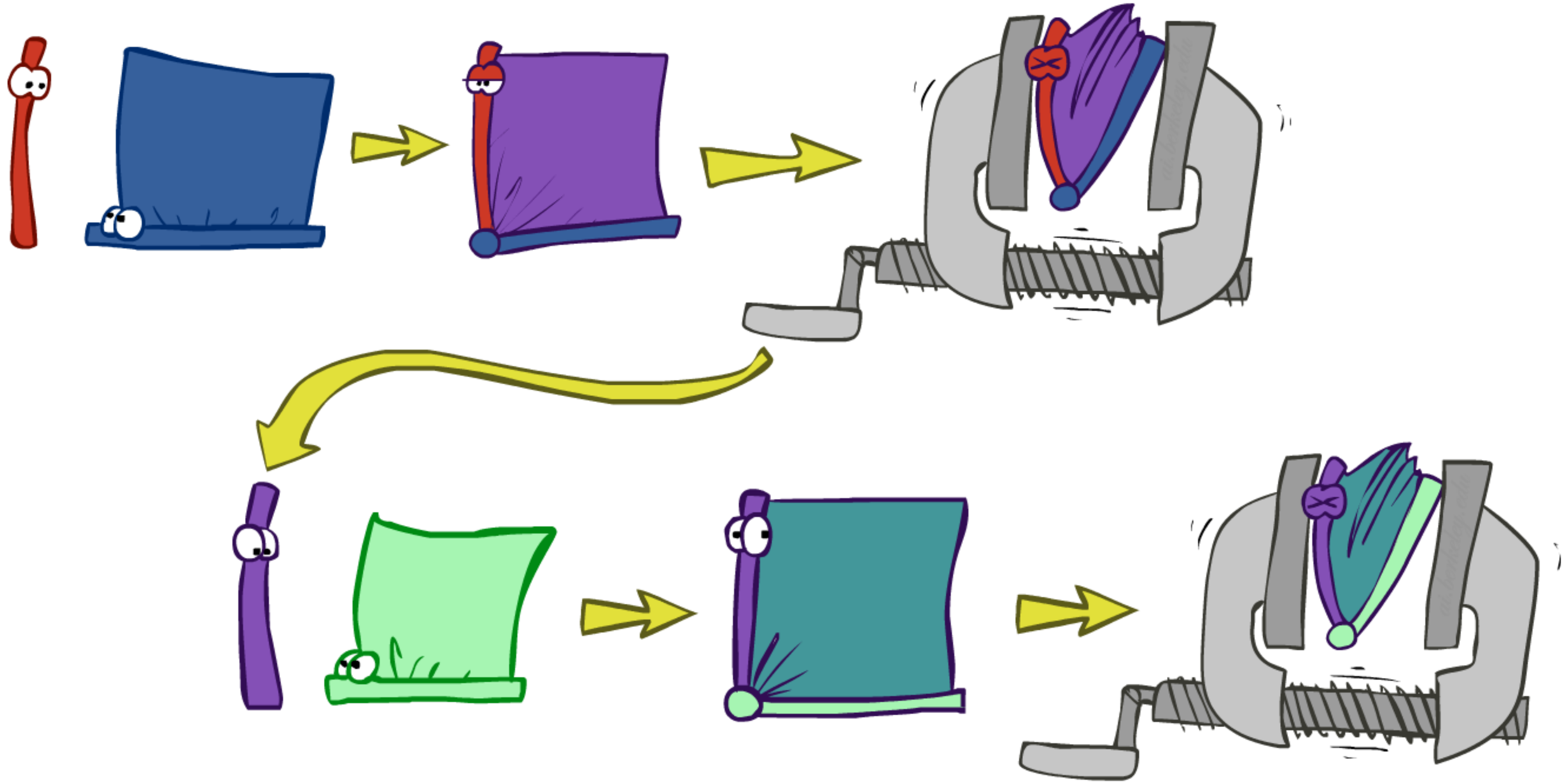


Inference by Enumeration

Thus Far: Multiple Join, Multiple Eliminate

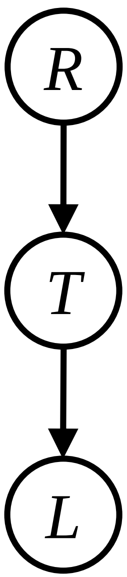


Marginalizing Early (= Variable Elimination)



Traffic Domain

$$P(L) = ?$$



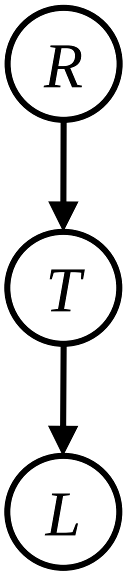
- Inference by Enumeration

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

- Variable Elimination

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

Marginalizing Early! (aka VE)



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

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

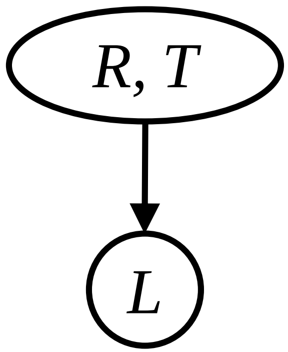
$P(L|T)$

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

Join R

$P(R,T)$

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



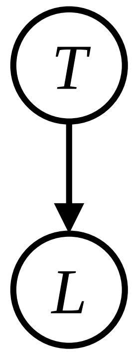
$P(L|T)$

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

Sum out R

$P(T)$

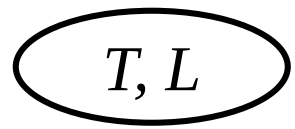
+t	0.17
-t	0.83



$P(L|T)$

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

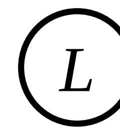
Join T



$P(T,L)$

+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	0.866

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

$$P(L | +r)$$

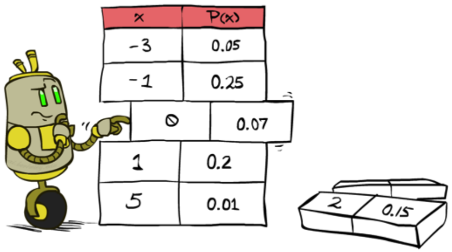
+l	0.26
-l	0.74

- To get our answer, just normalize this!

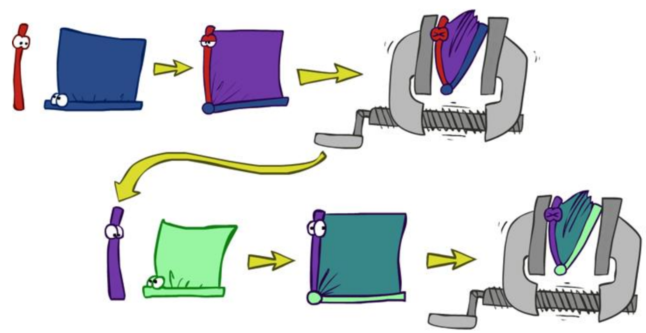
General Variable Elimination

- 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

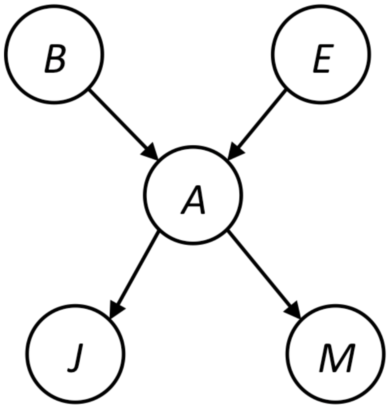
$$P(B|j,m) \propto P(B,j,m)$$

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

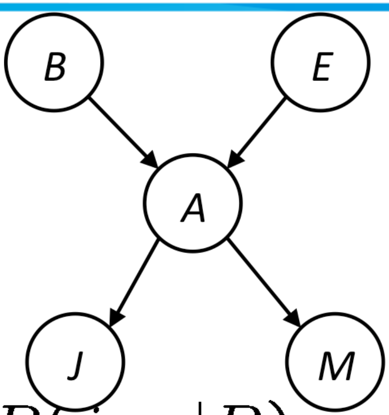
Choose A

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

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



Example



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

Choose E

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

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

Finish with B

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

Same Example in Equations

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

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

$$\begin{aligned}
P(B|j,m) &\propto P(B,j,m) \\
&= \sum_{e,a} P(B,j,m,e,a) \\
&= \sum_{e,a} P(B)P(e)P(a|B,e)P(j|a)P(m|a) \\
&= \sum_{e} P(B)P(e)\sum_{a} P(a|B,e)P(j|a)P(m|a) \\
&= \sum_{e} P(B)P(e)f_1(B,e,j,m) \\
&= P(B)\sum_{e} P(e)f_1(B,e,j,m) \\
&= P(B)f_2(B,j,m)
\end{aligned}$$

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

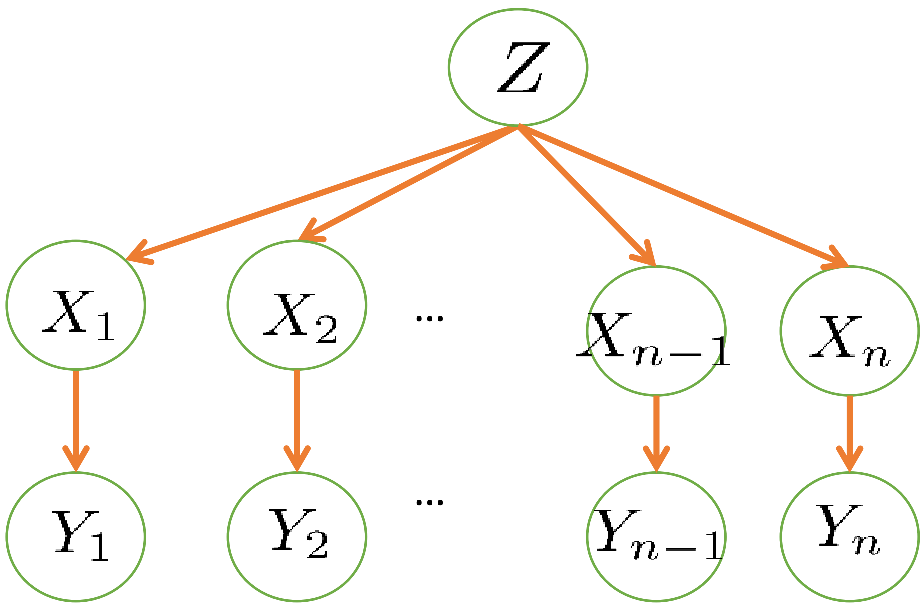
joining on e, and then summing out gives f_2

All we are doing is exploiting uwy + uwz + uxy + uxz + vwy + vwz + vxy +vxz

= (u+v)(w+x)(y+z) to improve computational efficiency!

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 versus 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!

Polytrees

- A polytree is a directed graph with no undirected cycles
- For poly-trees you can always find an ordering that is efficient
- Cut-set conditioning for Bayes’ net inference（Clustering algorithm）
  - Choose set of variables such that if removed only a polytree remains

Bayes' Nets

- Representation
- Conditional Independences
- Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Sampling (approximate)