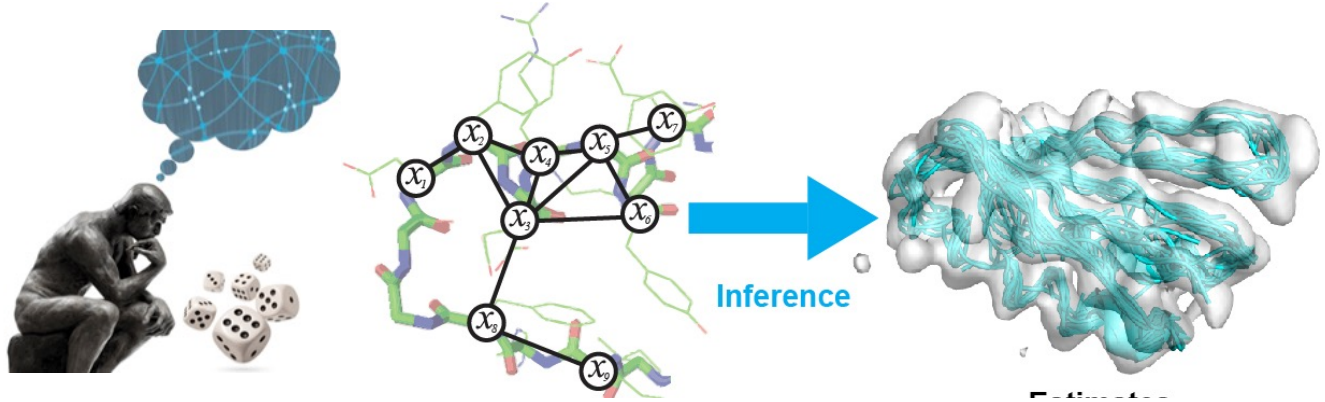
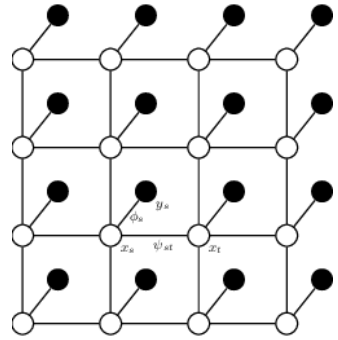
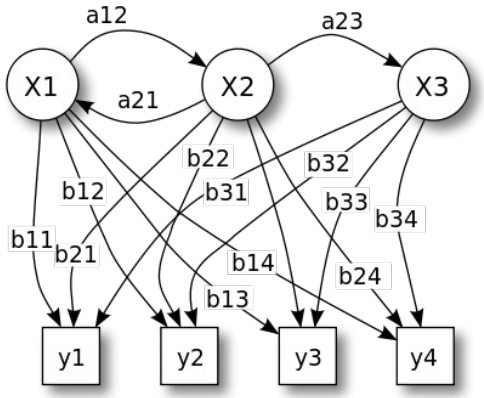
AI3603: Artificial Intelligence: Principles and Applications

Probabilistic Graphical Model II: Inference & Sampling

Yue Gao
Shanghai Jiao Tong University



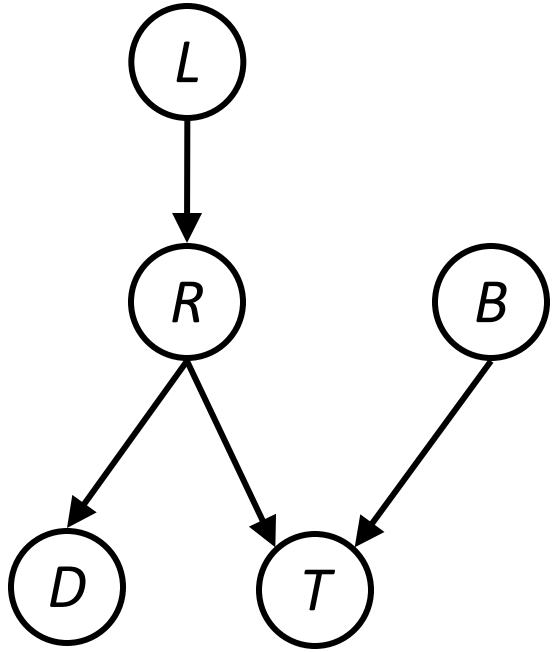




Contents

Reachability

- Recipe: shade evidence nodes, look for paths in the resulting graph
- Attempt 1: if two nodes are connected by an undirected path not blocked by a shaded node, they are conditionally independent
- Almost works, but not quite
  - Where does it break?
  - Answer: the v-structure at T doesn't count as a link in a path unless "active"

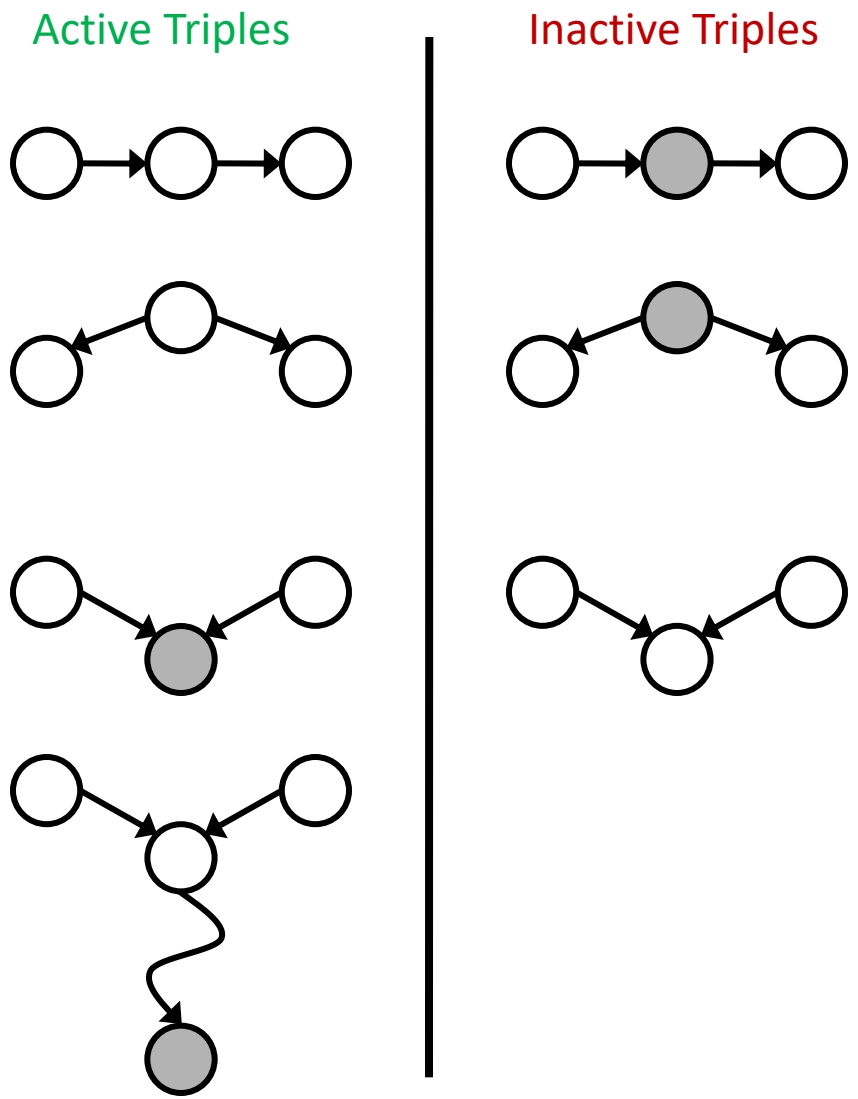


Active / Inactive Paths

- Question: Are X and Y conditionally independent given evidence variables {Z}?
  - Yes, if X and Y "d-separated" by Z
  - Consider all (undirected) paths from X to Y
  - No active paths = independence!

- A path is active if each triple is active:
  - Causal chain $A \rightarrow B \rightarrow C$ where B is unobserved (either direction)
  - Common cause $A \leftarrow B \rightarrow C$ where B is unobserved
  - Common effect (aka v-structure)
    $A \rightarrow B \leftarrow C$ where B *or one of its descendents* is observed

- All it takes to block a path is a single inactive segment



D-Separation

- Query: $X_i \perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$?
- Check all (undirected!) paths between X_i and X_j
  - If one or more active, then independence not guaranteed

$$X_i \not\perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$$

  - Otherwise (i.e. if all paths are inactive), then independence is guaranteed

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

Bayes' Nets

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

Contents

Inference

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

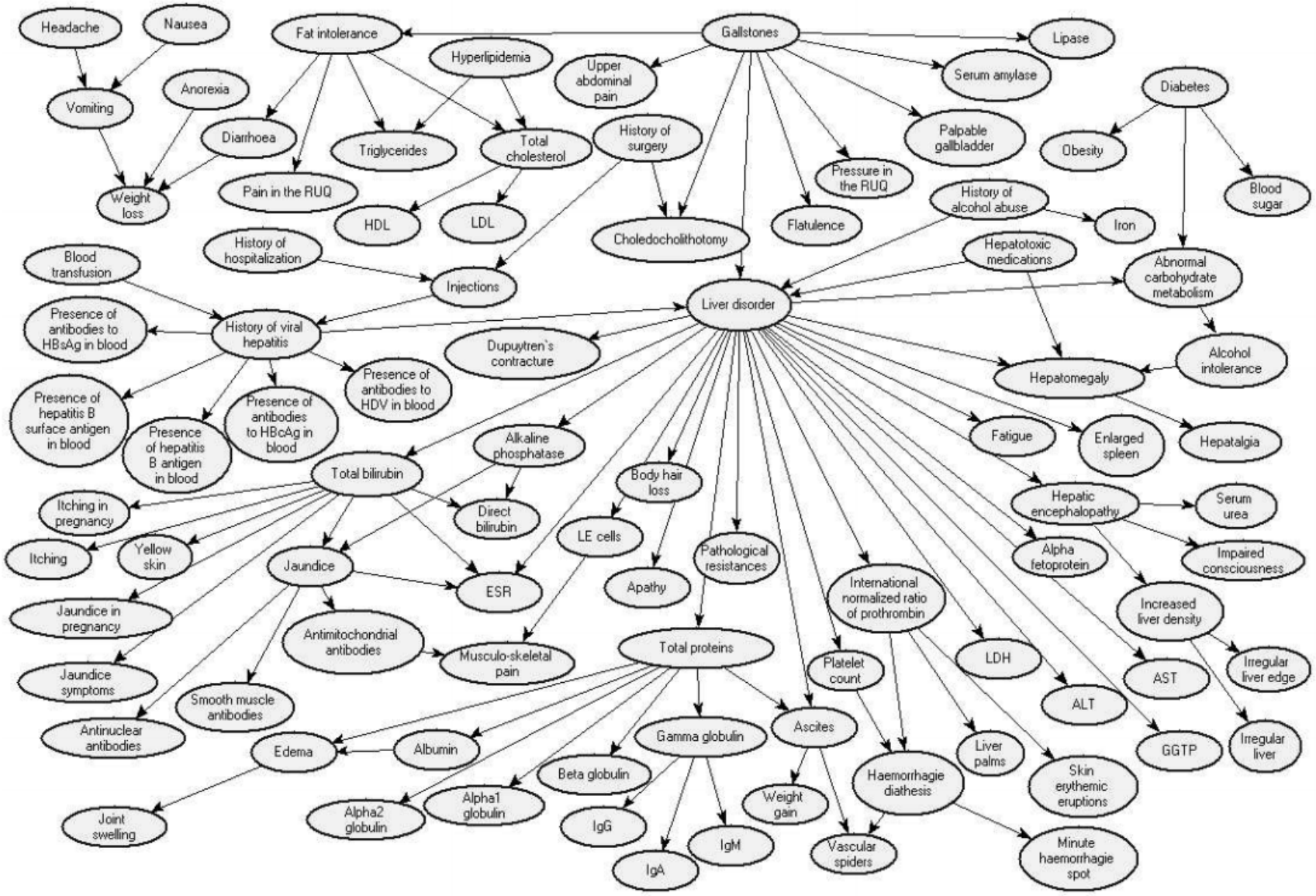
- Examples:
  - Posterior probability

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

  - Most likely explanation:

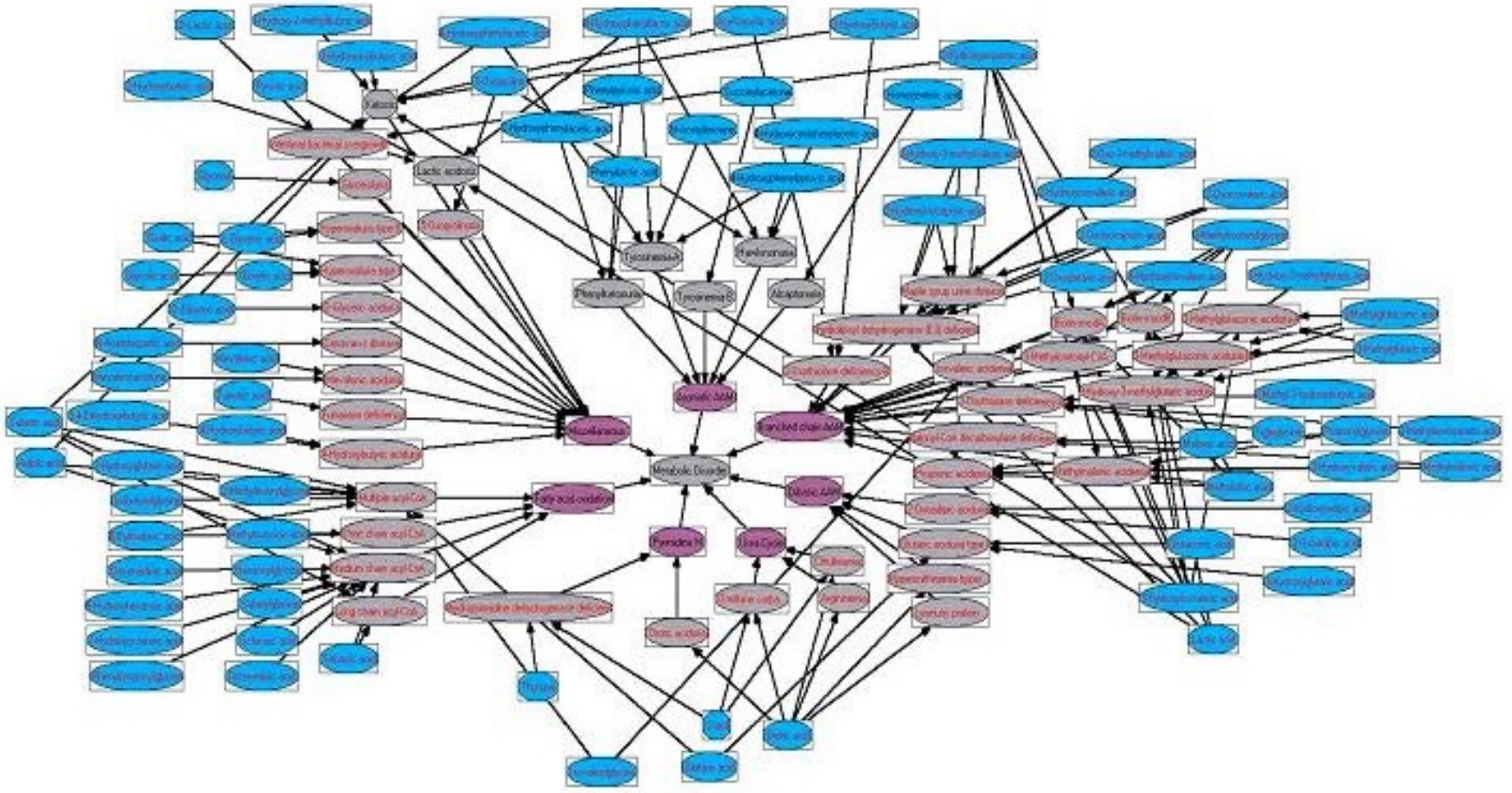
    $$\text{argmax}_q \; P(Q = q|E_1 = e_1 \ldots)$$

Liver Disorder



Onisko *et al* CBMI-99-27 (1999)

Test for Infant Metabolic Defects



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

Inference by Enumeration

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

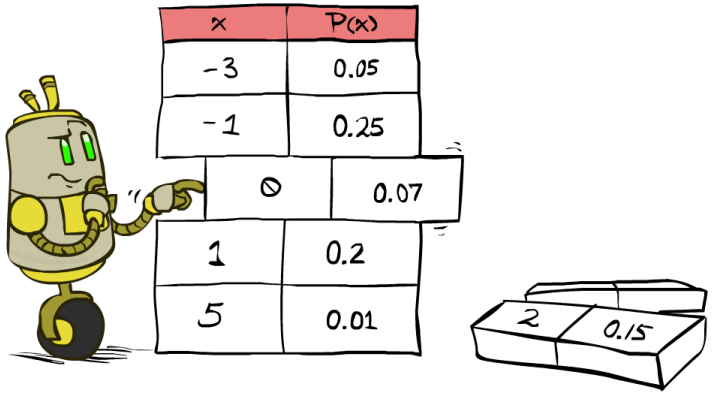
$X_1, X_2, \dots X_n$ *All variables*

- We want:

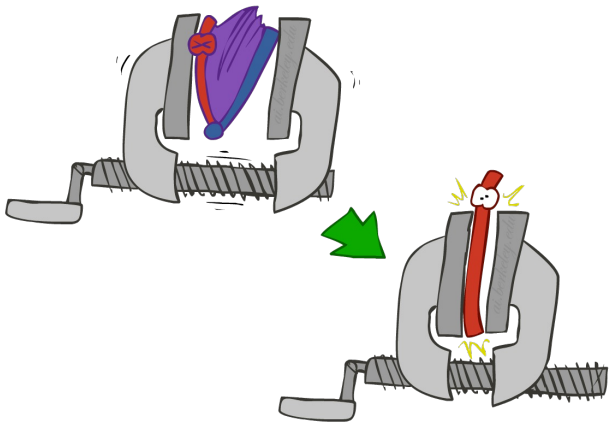
$$P(Q|e_1 \dots e_k)$$

** 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



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

- Step 3: Normalize

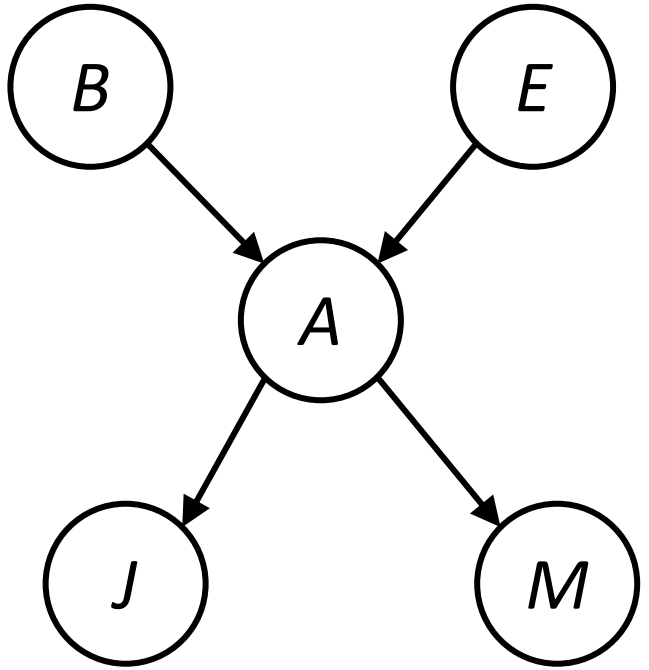
$$\times \frac{1}{Z}$$

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

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

Inference by Enumeration in Bayes' Net

- Given unlimited time, inference in BNs is easy



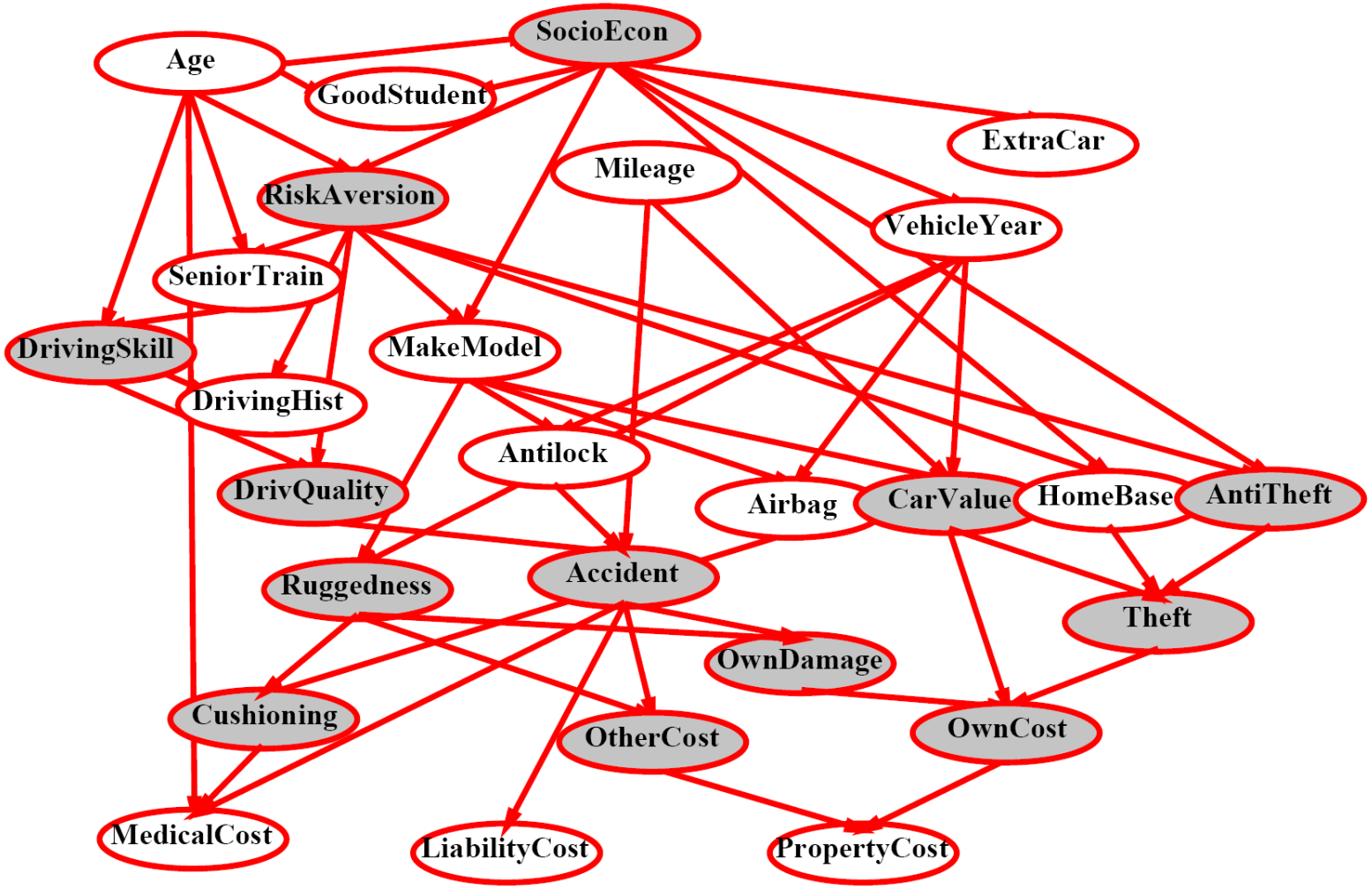
$$P(B \mid +j, +m) \propto_B P(B, +j, +m)$$

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

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

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

Inference by Enumeration?

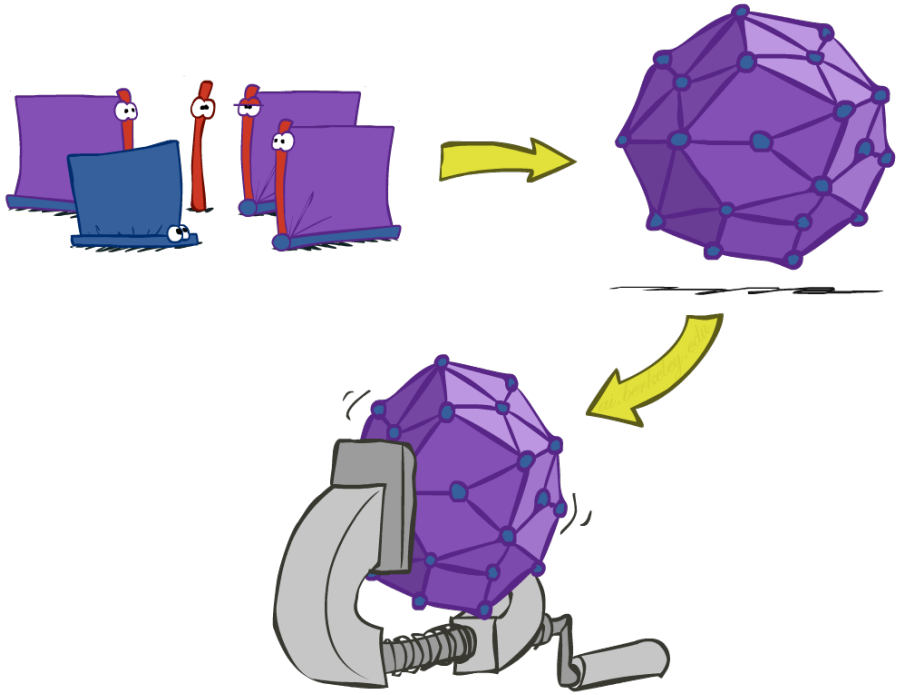


P(Antilock | age, model) = ?

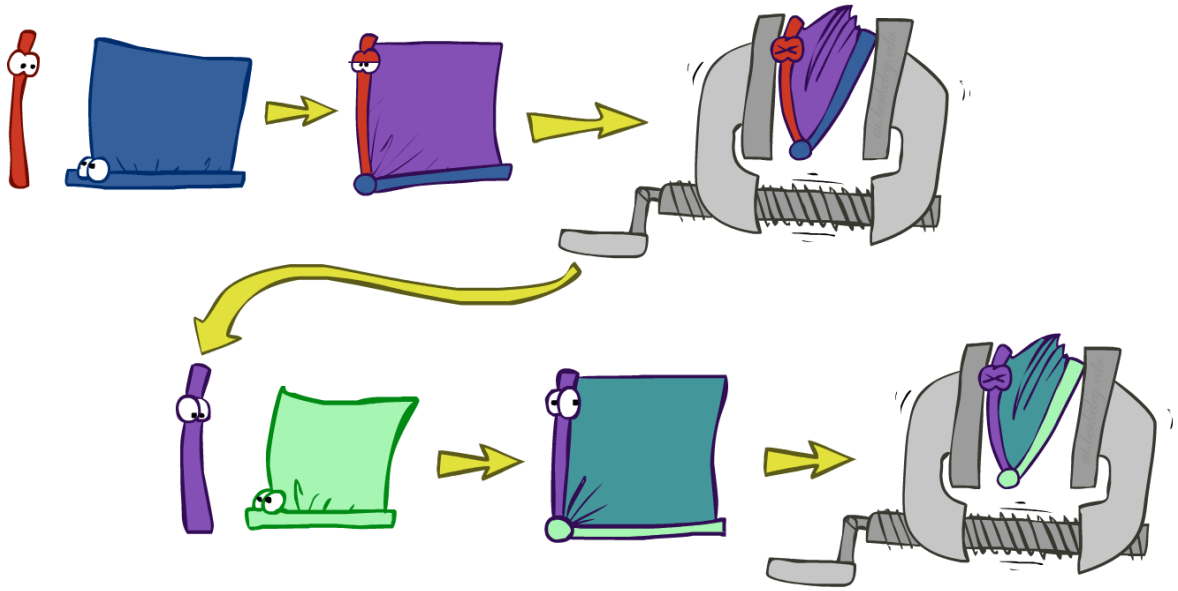
Contents

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"
  - new notation: factors
  - Still NP-hard, but usually much faster than inference by enumeration



Factor I

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

$$P(T, W)$$

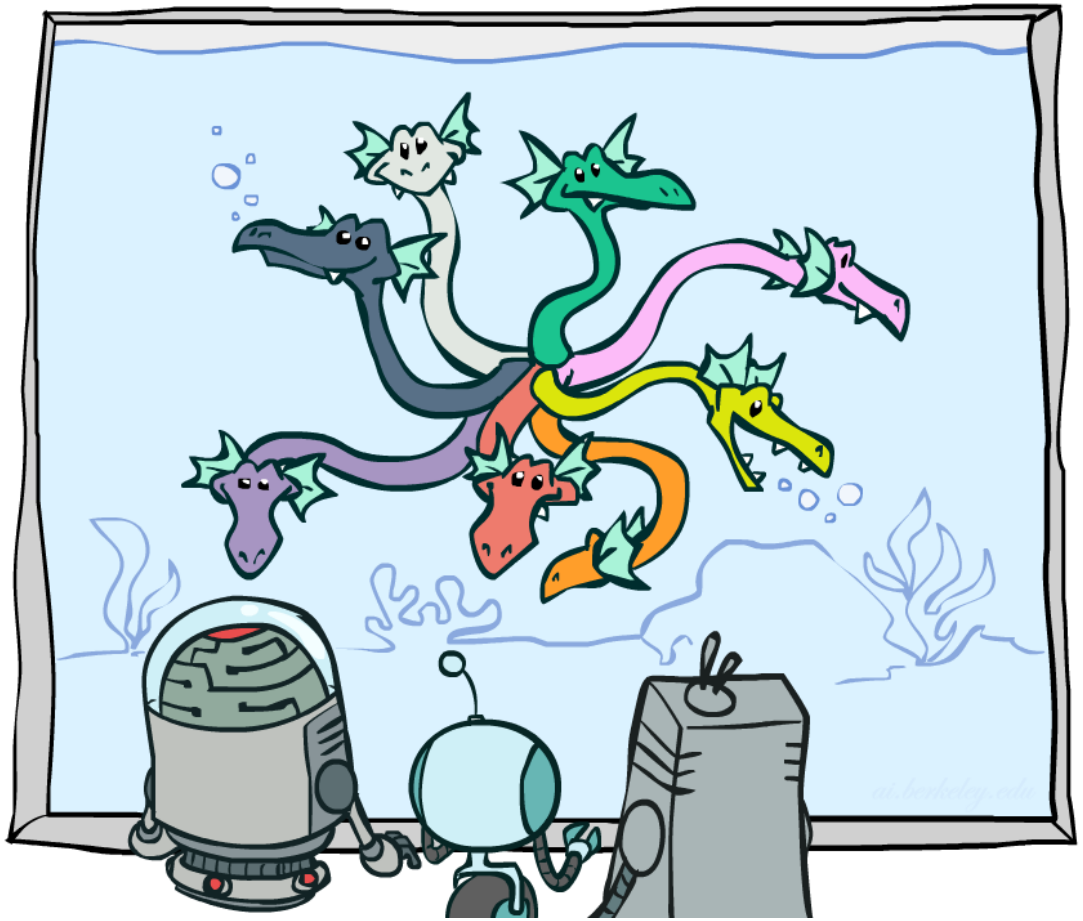
T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

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

$$P(cold, W)$$

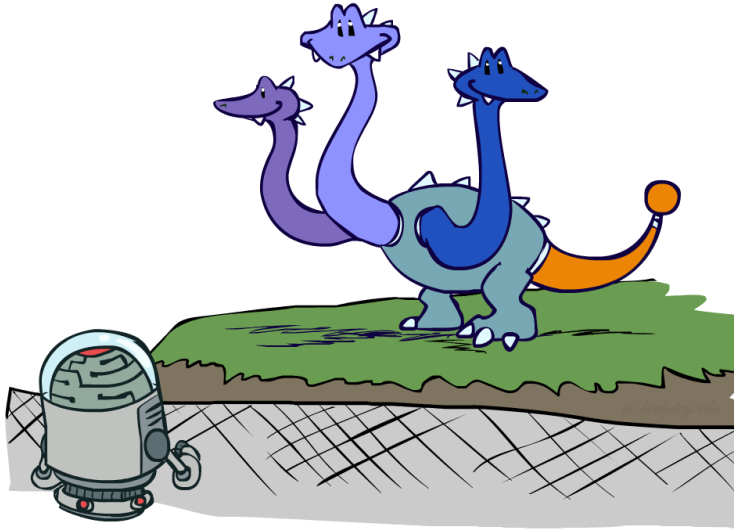
T	W	P
cold	sun	0.2
cold	rain	0.3

- Number of capitals = dimensionality of the table



Factor II

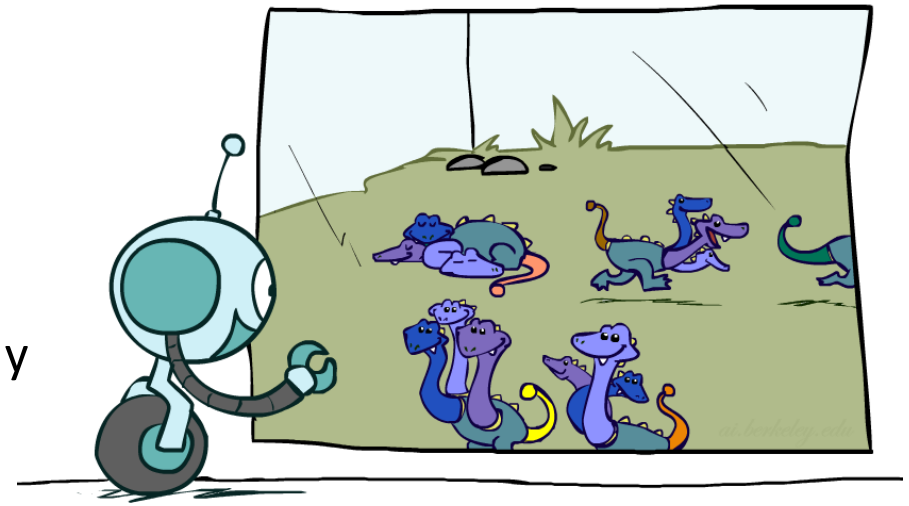
- Single conditional: P(Y | x)
  - Entries P(y | x) for fixed x, all
  - Sums to 1



$$P(W|cold)$$

T	W	P
cold	sun	0.4
cold	rain	0.6

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



$$P(W|T)$$

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

$P(W|hot)$

$P(W|cold)$

Factor III

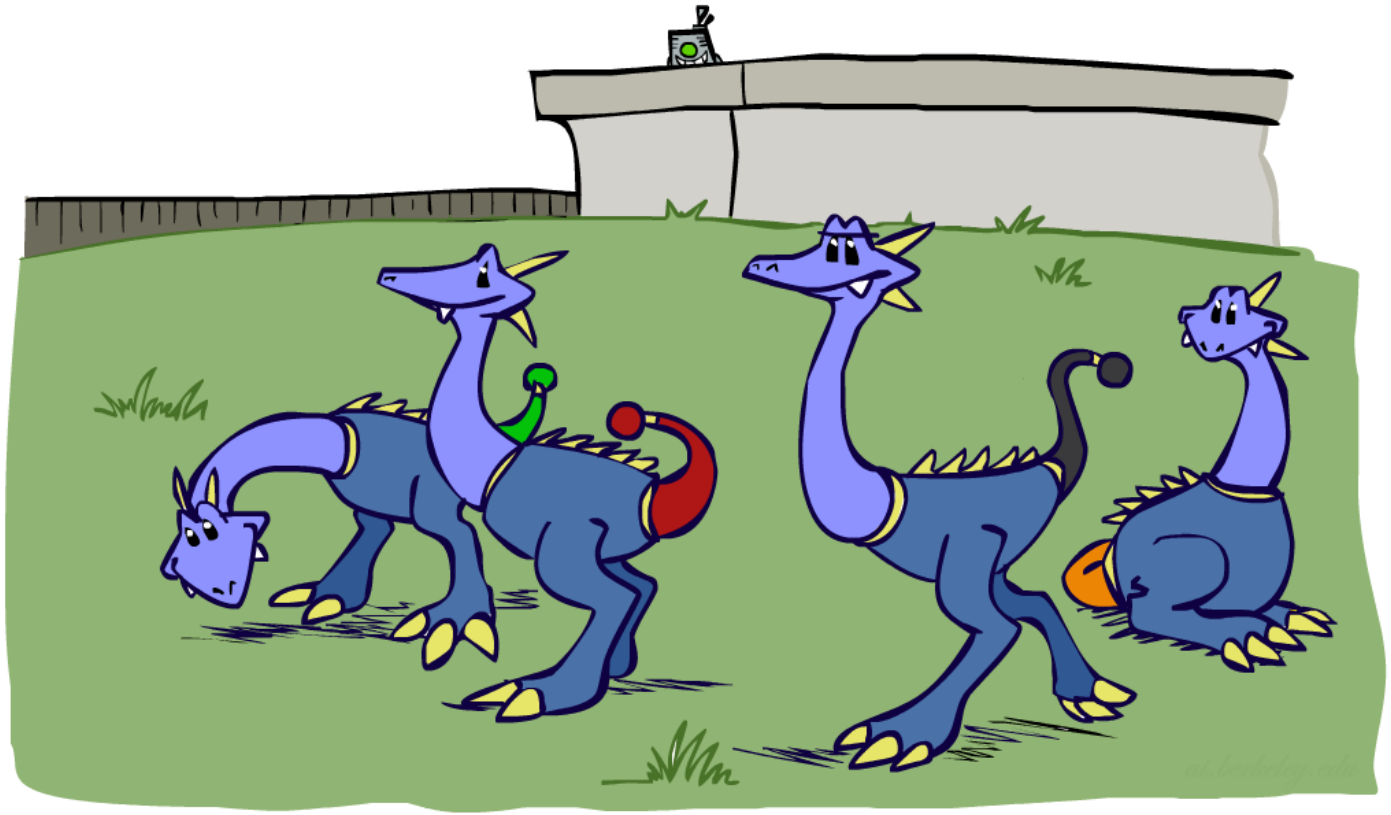
- Specified family: P(y | X)
  - Entries P(y | x) for fixed y, but for all x
  - Sums to … who knows!

$$P(rain|T)$$

T	W	P
hot	rain	0.2
cold	rain	0.6

$P(rain|hot)$

$P(rain|cold)$

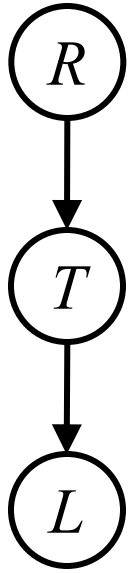


Factor Summary

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

Example: Traffic Domain

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



$$P(L) = ?$$

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

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

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

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

$P(L|T)$

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

Inference by Enumeration: Procedural Outline

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

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

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

$P(L|T)$

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

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

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

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

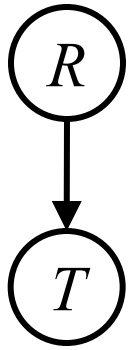
$P(+\ell|T)$

+t	+l	0.3
-t	+l	0.1

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

Operation 1: Join Factors

- First basic operation: joining factors
- Combining factors:
  - Just like a database join
  - Get all factors over the joining variable
  - Build a new factor over the union of the variables involved
- Example: Join on R



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

+r	0.1
-r	0.9

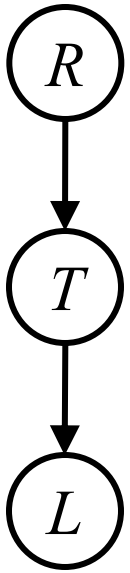
+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

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

- Computation for each entry: pointwise products

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

Example: Multiple Joins



$P(R)$

+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

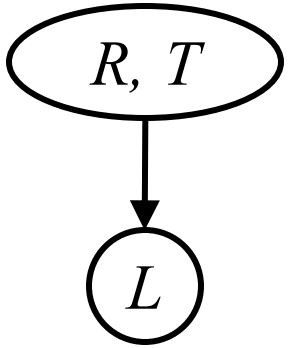
$P(R,T)$

+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

得到一个新的 factor



Join T

$P(R,T,L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729

Operation 2: Eliminate

- Second basic operation: marginalization
- Take a factor and sum out a variable
  - Shrinks a factor to a smaller one
  - A projection operation
- Example:

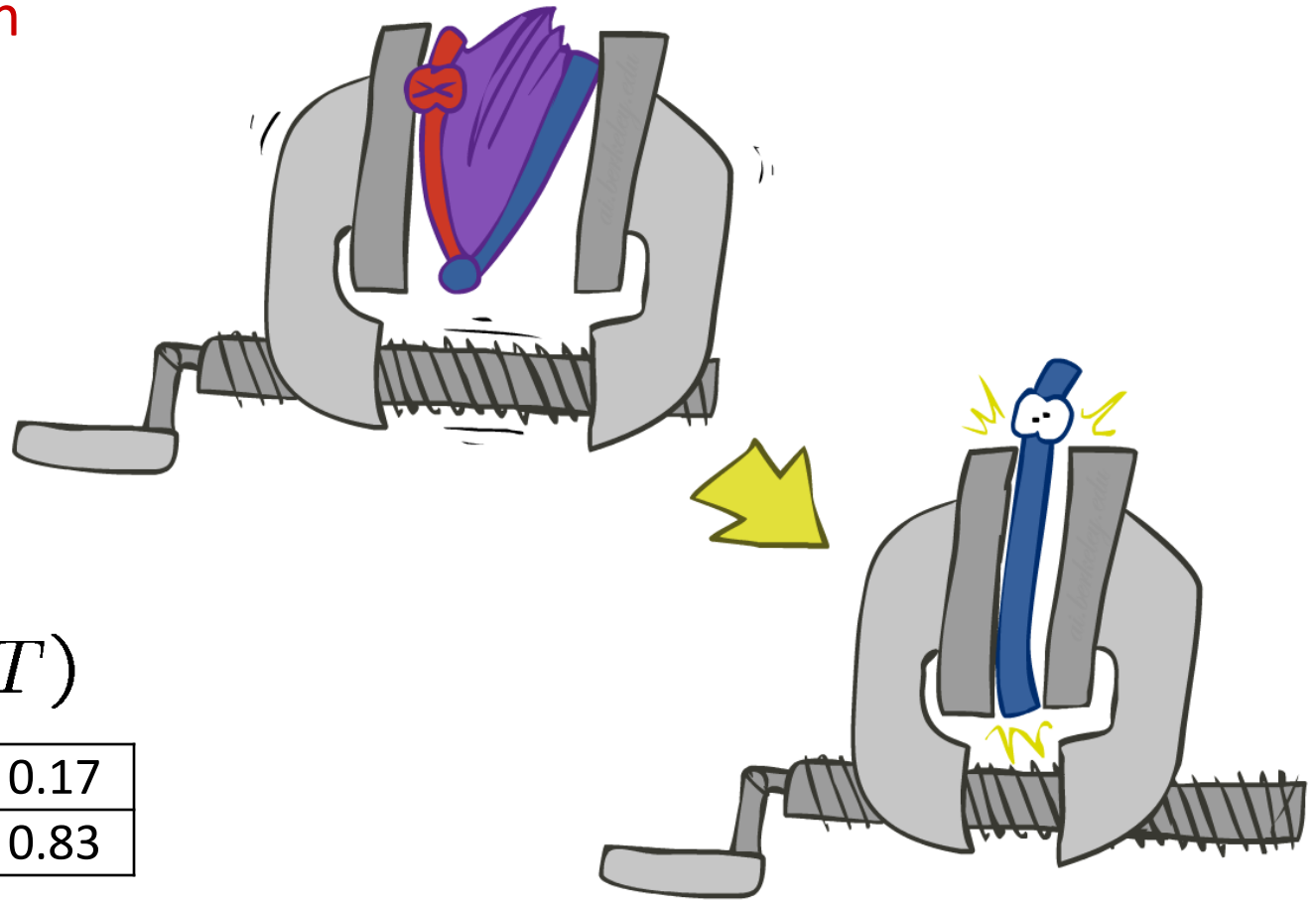
$P(R,T)$

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

sum R

$P(T)$

+t	0.17
-t	0.83



Multiple Elimination

$P(R, T, L)$

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

Inference by Enumeration

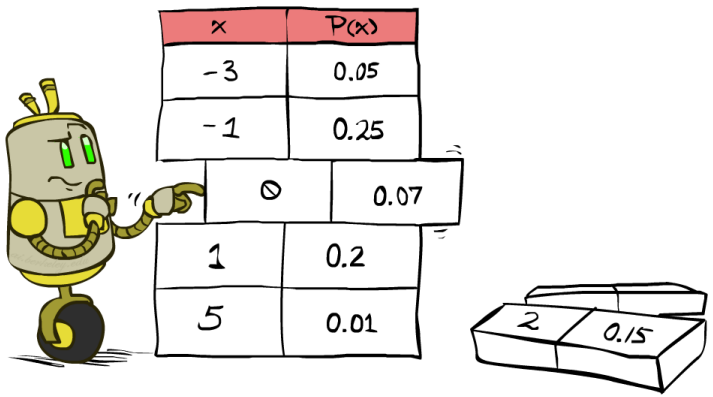
- General case:
  - Evidence variables: $E_1 \dots E_k = e_1 \dots e_k$
  - Query* variable: Q
  - Hidden variables: $H_1 \dots H_r$

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

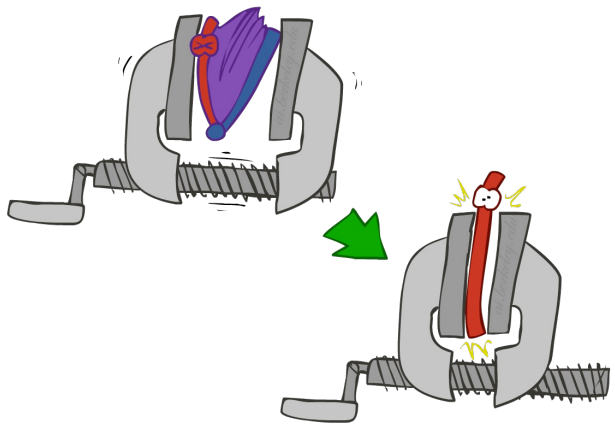
- We want: $P(Q|e_1 \dots e_k)$

** 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



- Compute joint

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

- Sum out hidden variables

- Step 3: Normalize

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

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

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

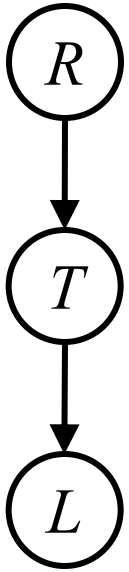
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)

$P(R)$

$P(T|R)$  $P(R,T,L)$  $P(L)$

$P(L|T)$

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

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

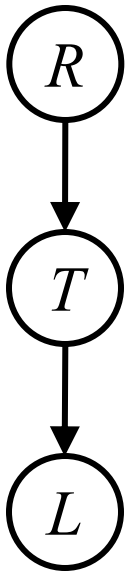
Join on r

Eliminate r

Join on t

Eliminate t

Marginalizing Early! (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

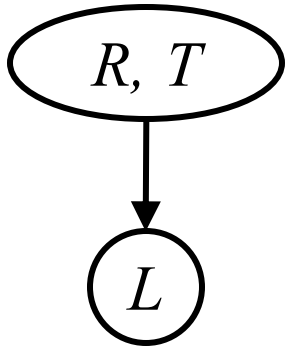
$P(L|T)$

+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



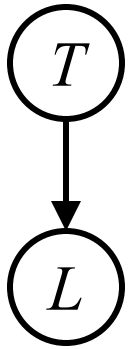
$P(L|T)$

+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



$P(L|T)$

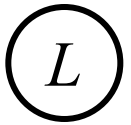
+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

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!

Inference by Enumeration

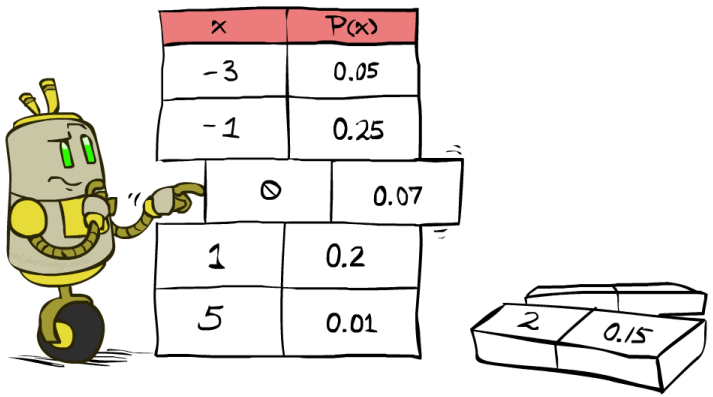
- General case:
  - Evidence variables: $E_1 \dots E_k = e_1 \dots e_k$
  - Query* variable: Q
  - Hidden variables: $H_1 \dots H_r$

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

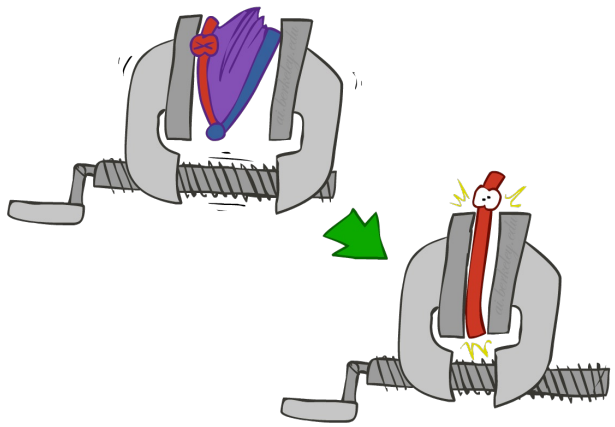
- We want: $P(Q|e_1 \dots e_k)$

** 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



- Compute joint

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

- Sum out hidden variables

- Step 3: Normalize

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

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

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

Variable Elimination

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

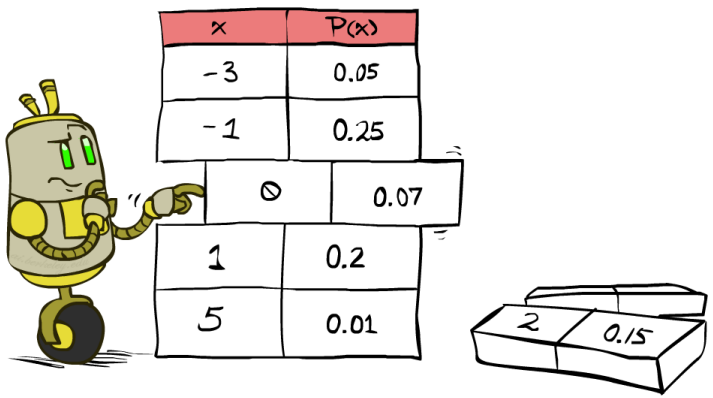
  $X_1, X_2, \dots X_n$ *All variables*

- We want:

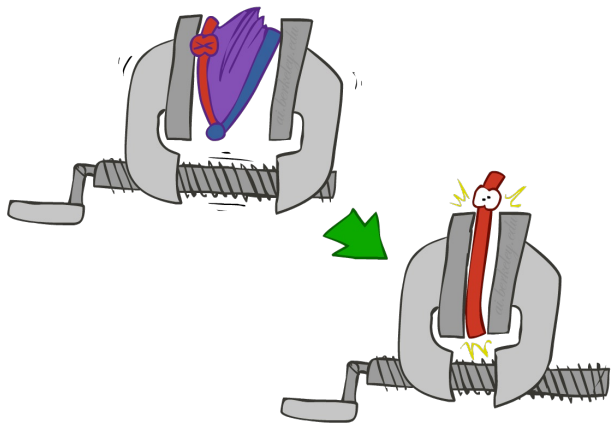
$$P(Q|e_1 \dots e_k)$$

** 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



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

- Interleave joining and summing out

- Step 3: Normalize

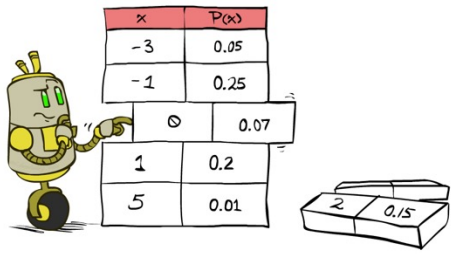
$$\times \frac{1}{Z}$$

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

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

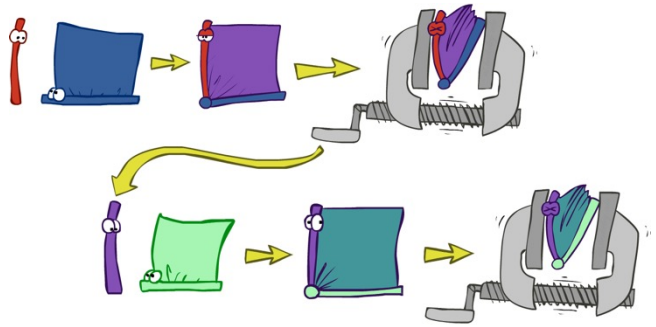
General Variable Elimination

- 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



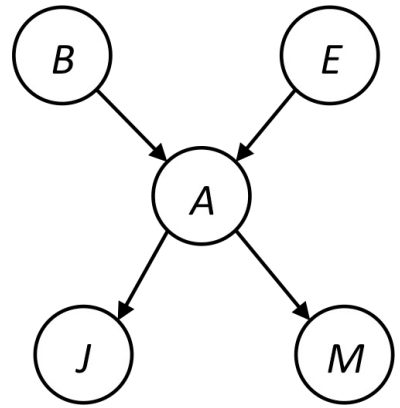
- Join all remaining factors and normalize

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

Example

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

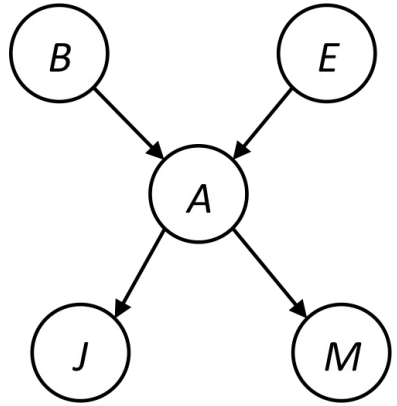
$P(B)$	$P(E)$	$P(A\|B,E)$	$P(j\|A)$	$P(m\|A)$



Example

从P(B,j,m) 得到P(B | j, m)

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



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

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

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

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

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

$$= \sum_{e} P(B)P(e)f_1(j,m|B,e)$$

$$= P(B) \sum_{e} P(e)f_1(j,m|B,e)$$

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

最后一步才 normalization

marginal can be obtained from joint by summing out

use Bayes' net joint distribution expression

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

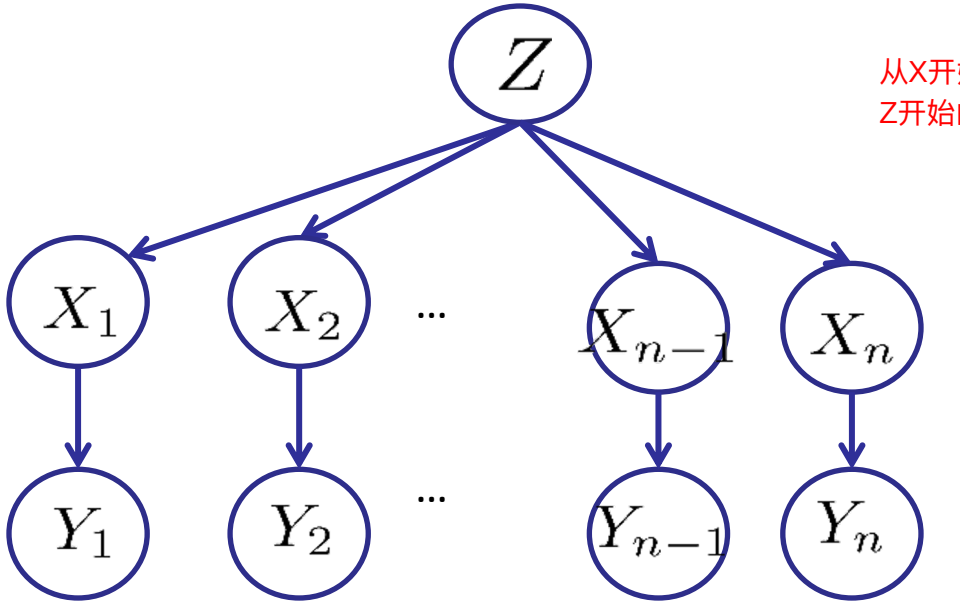
joining on a, and then summing out gives f_1

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

joining on e, and then summing out gives f_2

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?



从X开始，往Z上走，join只需要4
Z开始的话需要2的n次方

Evidence

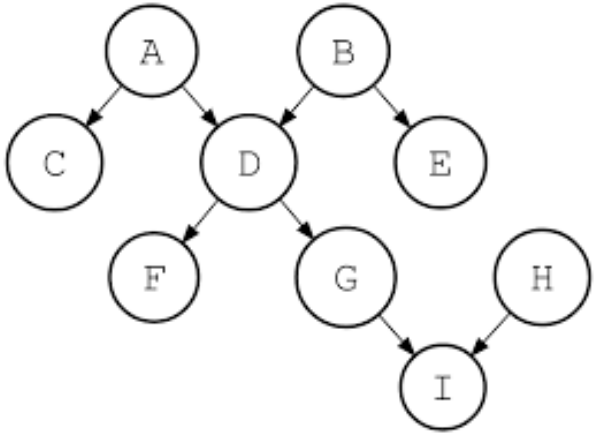
- In general: the ordering can greatly affect efficiency.

Computational and Space Complexity

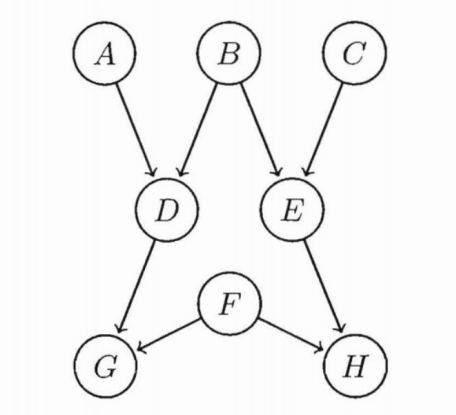
- The computational and space complexity of variable elimination is determined by the largest factor
- The elimination ordering can greatly affect the size of the largest factor.
- Does there always exist an ordering that only results in small factors?
  - No!

Polytrees

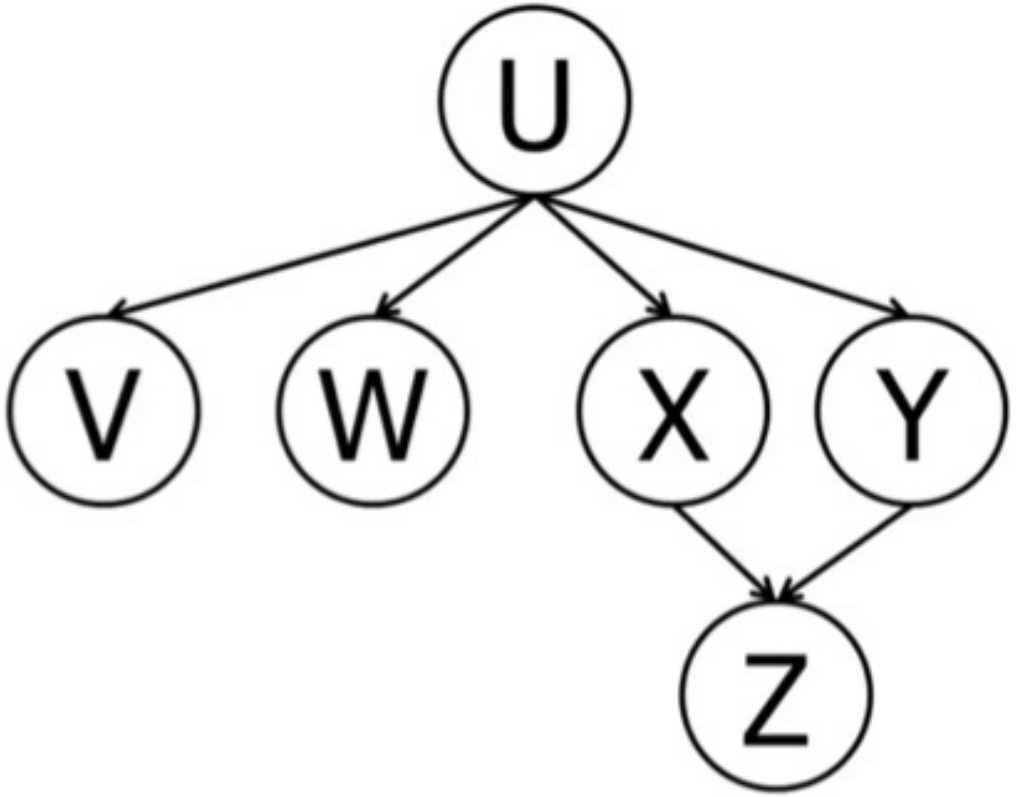
- A polytree is a directed graph with no undirected cycles



- For poly-trees you can always find an ordering that is efficient
  - Try it!!
- Cut-set conditioning for Bayes' net inference
  - Choose set of variables such that if removed only a polytree remains



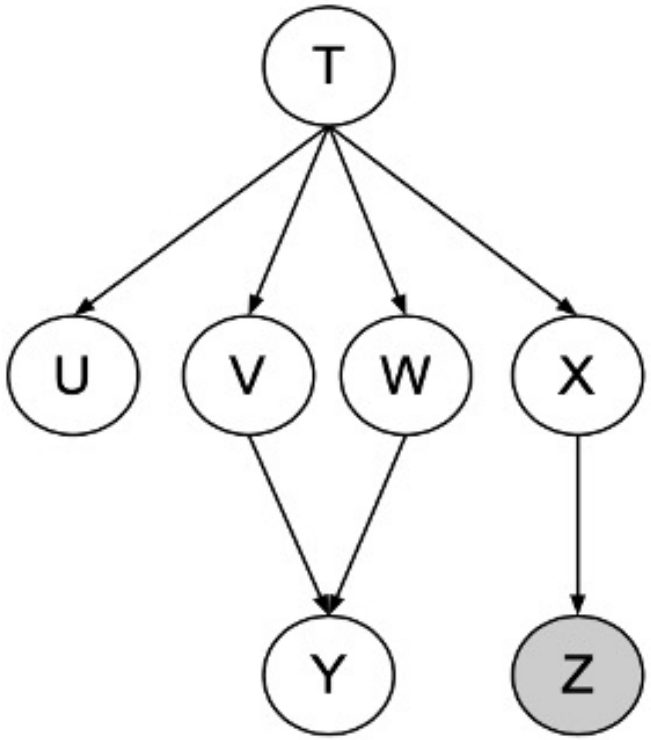
Exercise



Query: $P(V, W| + z)$

Elimination ordering: Y,X,U

Exercise



- Compute P(Y|+z)
- Elimination ordering
- X,T,U,V,W
- What is the size of the largest factor?
- Does there exist a better ordering?

Contents

Intro

- What is P(-a, +b)
- A 5/100
- B 20/100
- C 25/100
- D 125/100

$P(A, B, C)$

+a	+b	+c	4/100
+a	+b	-c	1/100
+a	-b	+c	45/100
+a	-b	-c	0/100
-a	+b	+c	20/100
-a	+b	-c	5/100
-a	-b	+c	25/100
-a	-b	-c	0/100

Intro

- What is P(-c |+a, +b)
- A 1/4
- B 1/5
- C 1/100
- D 4/100

$P(A, B, C)$

+a	+b	+c	4/100
+a	+b	-c	1/100
+a	-b	+c	45/100
+a	-b	-c	0/100
-a	+b	+c	20/100
-a	+b	-c	5/100
-a	-b	+c	25/100
-a	-b	-c	0/100

Intro

- Given these N=10 observations of the world
- What is the approximate value for

  P(-a, +b)
- A 1/10
- B 4/10
- C 5/10
- D 2.5/10

$-a, -b, +c$
$+a, -b, +c$
$-a, -b, +c$
$-a, +b, +c$
$+a, -b, +c$
$-a, +b, -c$
$-a, +b, +c$
$-a, +b, +c$
$+a, -b, +c$
$-a, +b, +c$

Counts

+a	+b	+c	0
+a	+b	-c	0
+a	-b	+c	3
+a	-b	-c	0
-a	+b	+c	4
-a	+b	-c	1
-a	-b	+c	2
-a	-b	-c	0

Intro

- Given this Bayes Net and CPT
- What is P(+a, +b, +c)
- A 1/2
- B 5/100
- C 3/100
- D 4/100

D

$P(A)$

+a	1/2
-a	1/2

$P(B|A)$

+a	+b	1/10
	-b	9/10
-a	+b	1/2
	-b	1/2

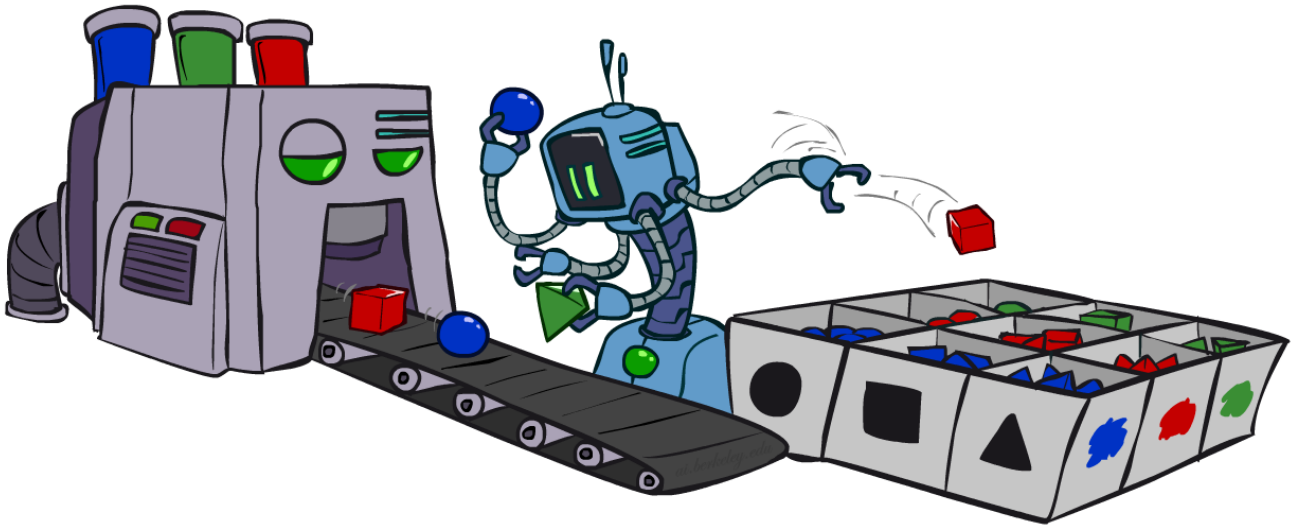
$P(C|B)$

+b	+c	4/5
	-c	1/5
-b	+c	1
	-c	0

Inference vs Sampling

Sampling

- Sampling is a lot like repeated simulation
  - Predicting the weather, basketball games, …
- Basic idea
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior probability
  - Show this converges to the true probability P
- Why sample?
  - Learning: get samples from a distribution you don't know
  - Inference: getting a sample is faster than computing the right answer (e.g. with variable elimination)



Sampling

- Sampling from given distribution
  - Step 1: Get sample *u* from uniform distribution over [0, 1)
    - E.g. random() in python
  - Step 2: Convert this sample *u* into an outcome for the given distribution by having each target outcome associated with a sub-interval of [0,1) with sub-interval size equal to probability of the outcome

- Example

C	P(C)
red	0.6
green	0.1
blue	0.3

$$0 \leq u < 0.6, \rightarrow C = red$$
$$0.6 \leq u < 0.7, \rightarrow C = green$$
$$0.7 \leq u < 1, \rightarrow C = blue$$

  - If random() returns $u = 0.83$, then our sample is C = blue

Exercise: sampling from a joint distribution

A	P(A)
+a	0.8
-a	0.2

Step 1: Draw uniformly at random from [0,1)

0.31	**0.58**	**0.04**	**0.94**	**0.67**	**0.49**

…

Step 2: Generate samples from requested sample distribution

Exercise: sampling from a joint distribution

A	P(A)
+a	0.8
-a	0.2

Step 1: Draw uniformly at random from [0,1)

0.31	**0.58**	**0.04**	**0.94**	**0.67**	**0.49**

…

Step 2: Generate samples from requested sample distribution

+a, +a, +a, -a, +a, +a

Exercise: sampling from a joint distribution

A	P(A)
a1	0.35
a2	0.15
a3	0.10
a4	0.25
a5	0.15

Step 1: Draw uniformly at random from [0,1)

0.31	**0.58**	**0.04**	**0.94**	**0.67**	**0.49**

…

Step 2: Generate samples from requested sample distribution

Exercise: sampling from a joint distribution

A	P(A)
a1	0.35
a2	0.15
a3	0.10
a4	0.25
a5	0.15

Step 1: Draw uniformly at random from [0,1)

0.31	**0.58**	**0.04**	**0.94**	**0.67**	**0.49**

…

Step 2: Generate samples from requested sample distribution

a1, a3, a1, a5, a4, a2

Exercise: sampling from a joint distribution

Step 1: Draw uniformly at random from [0,1)

0.31	**0.58**	**0.04**	**0.94**	**0.67**	**0.49**

…

Step 2: Generate samples from requested sample distribution

A	B	P(A,B)
+a	+b	0.3
+a	-b	0.1
-a	+b	0.4
-a	-b	0.2

Exercise: sampling from a joint distribution

Step 1: Draw uniformly at random from [0,1)

0.31	**0.58**	**0.04**	**0.94**	**0.67**	**0.49**	…

Step 2: Generate samples from requested sample distribution

A	B	P(A,B)
+a	+b	0.3
+a	-b	0.1
-a	+b	0.4
-a	-b	0.2

+a, -b ; -a +b; +a +b; -a,-b

Exercise: sampling from a condition distribution
Query P(B|+a)

Step 1: Draw uniformly at random from [0,1)

0.31	**0.58**	**0.04**	**0.94**	**0.67**	**0.49**

…

Step 2: Generate samples from requested sample distribution

A	B	P(B|A)
+a	+b	0.3
+a	-b	0.7
-a	+b	0.4
-a	-b	0.6

Exercise: sampling from a condition distribution
Query P(B|+a)

Step 1: Draw uniformly at random from [0,1)

0.31	**0.58**	**0.04**	**0.94**	**0.67**	**0.49**

…

Step 2: Generate samples from requested sample distribution

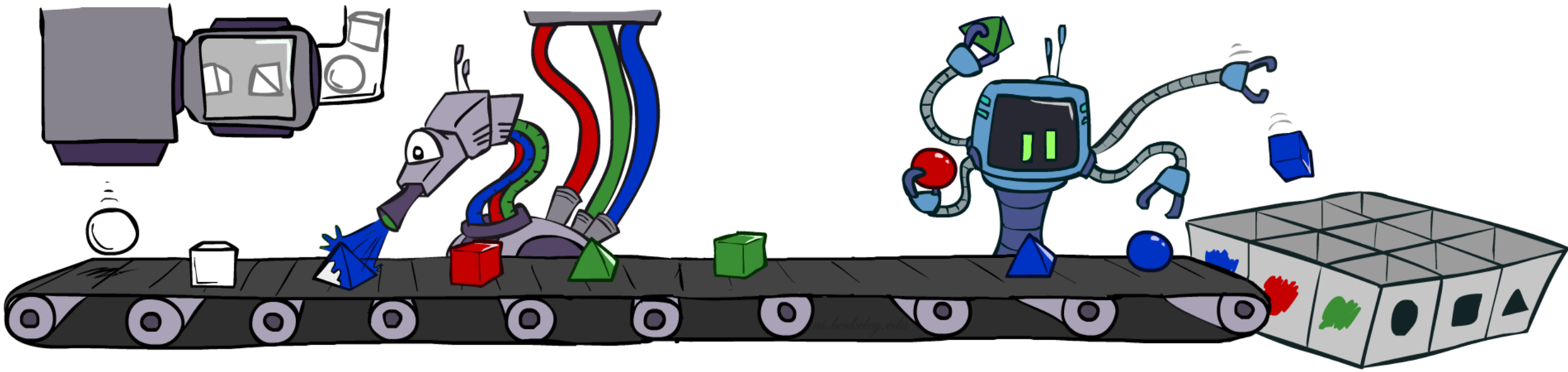
A	B	P(B|A)
+a	+b	0.3
+a	-b	0.7
-a	+b	0.4
-a	-b	0.6

-b, -b, +b, -b, -b, -b

Sampling in Bayes' Nets

- Prior Sampling
- Rejection Sampling
- Likelihood Weighting
- Gibbs Sampling

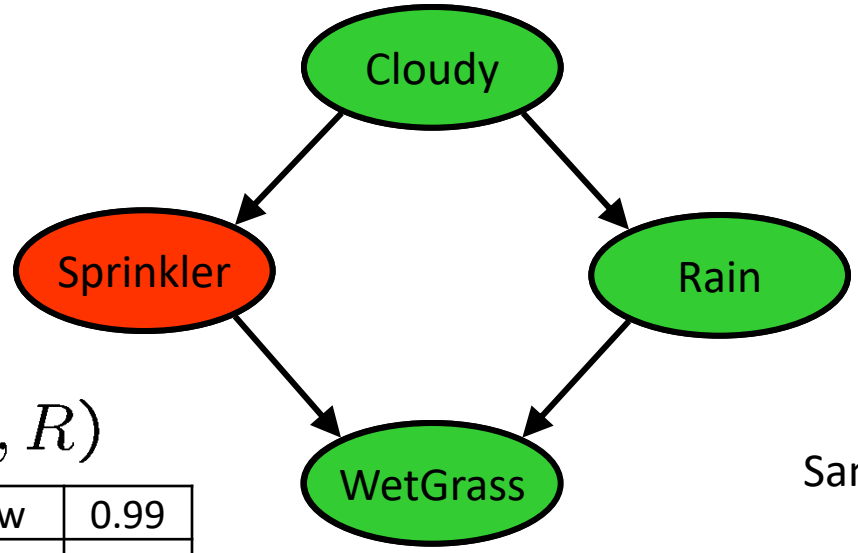
Prior Sampling



Prior Sampling

$P(C)$

+c	0.5
-c	0.5



$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5

$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

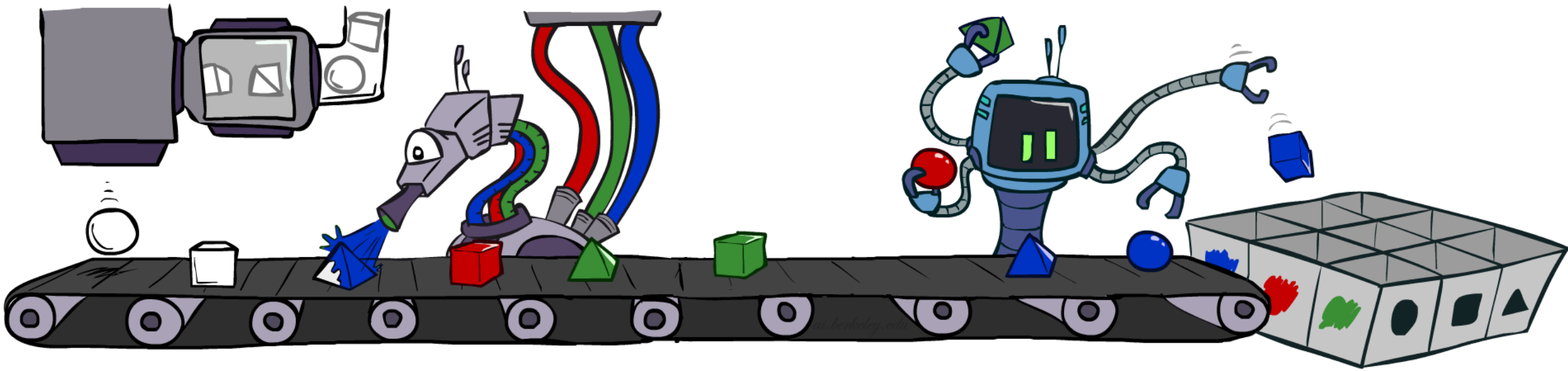
+c, -s, +r, +w

-c, +s, -r, +w

…

Prior Sampling

- For i = 1, 2, …, n
  - Sample x_i from $P(X_i \mid \text{Parents}(X_i))$
- Return $(x_1, x_2, \ldots, x_n)$



Question

- Prior Sampling: What does the value $\frac{N(+a,-b,+c)}{N}$ approximate
- A P(+a,-b,+c)
- B P(+c|+a,-b)
- C P(+c|-b)
- D P(+c)

$-a,-b,+c$
$+a,-b,+c$
$-a,-b,+c$
$-a,+b,+c$
$+a,-b,+c$
$-a,+b,-c$
$-a,+b,+c$
$-a,+b,+c$
$+a,-b,+c$
$-a,+b,+c$

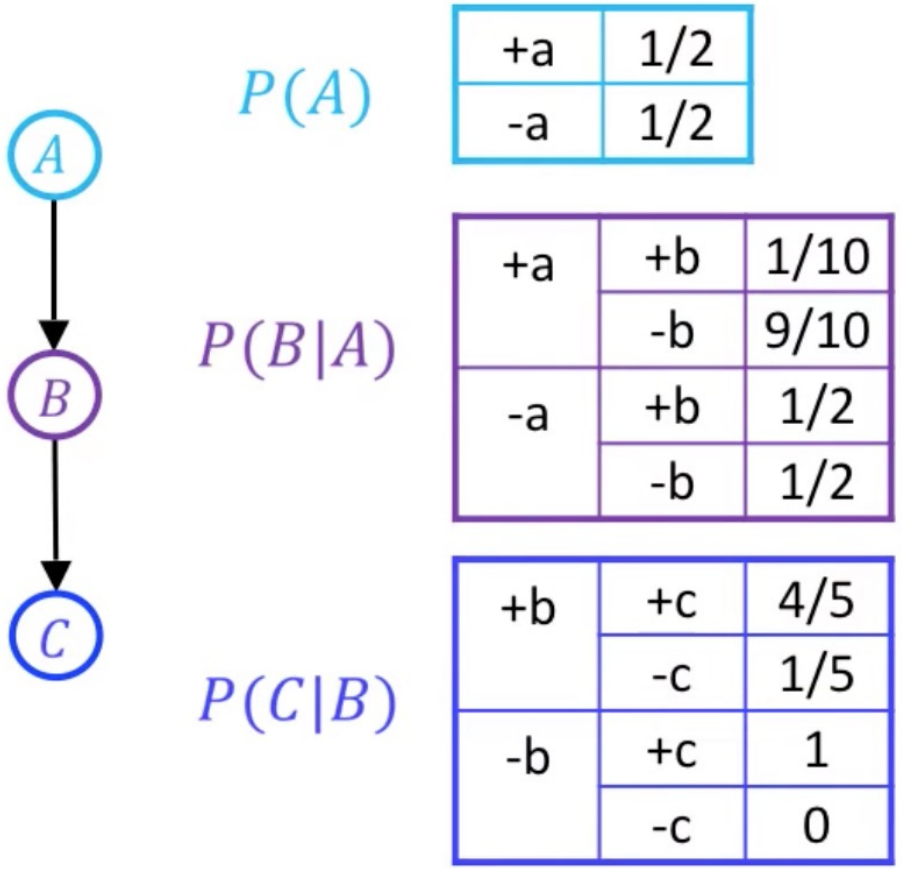
Counts

+a	+b	+c	0
+a	+b	-c	0
+a	-b	+c	3
+a	-b	-c	0
-a	+b	+c	4
-a	+b	-c	1
-a	-b	+c	2
-a	-b	-c	0

Question

- How many {-a,+b,-c} samples our of N = 1000 should we expect?
- A 1
- B 40
- C 125
- D 200
- E Other

50个



Prior Sampling

- This process generates samples with probability:

$$S_{PS}(x_1 \ldots x_n) = \prod_{i=1}^{n} P(x_i | \text{Parents}(X_i)) = P(x_1 \ldots x_n)$$

  …i.e. the BN's joint probability

- Let the number of samples of an event be $N_{PS}(x_1 \ldots x_n)$

- Then $$\begin{aligned} \lim_{N \to \infty} \hat{P}(x_1, \ldots, x_n) &= \lim_{N \to \infty} N_{PS}(x_1, \ldots, x_n)/N \\ &= S_{PS}(x_1, \ldots, x_n) \\ &= P(x_1 \ldots x_n) \end{aligned}$$

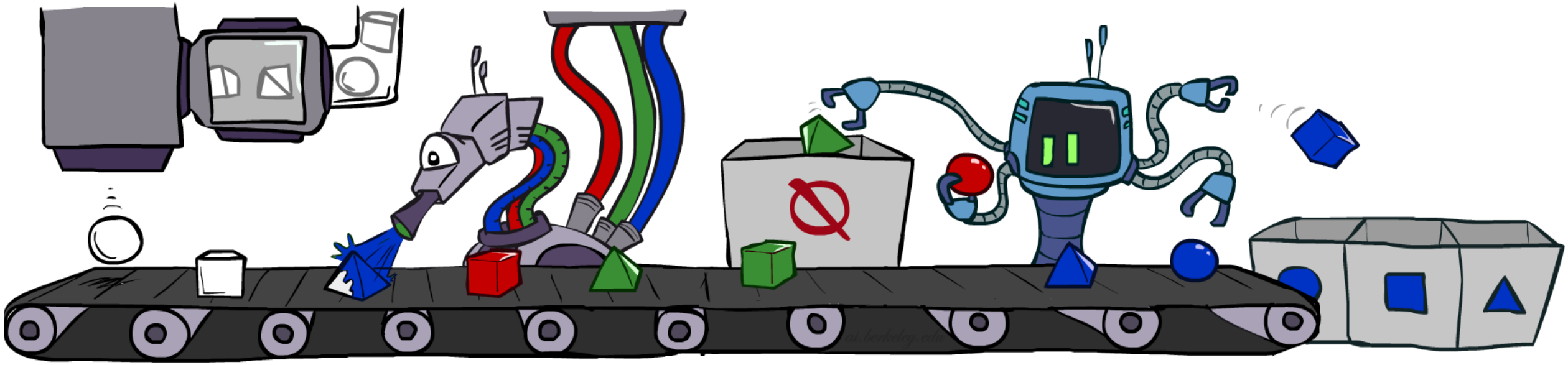
- I.e., the sampling procedure is consistent

If we know the query in advance, can we sample more efficiently?

If we know the query in advance, can we sample more efficiently?

Yes

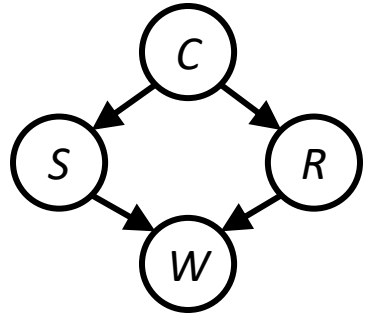
Rejection Sampling



Rejection Sampling

- Let's say we want P(C)
  - No point keeping all samples around
  - Just tally counts of C as we go

- Let's say we want P(C | +s)
  - Same thing: tally C outcomes, but ignore (reject) samples which don't have S=+s
  - This is called rejection sampling
  - It is also consistent for conditional probabilities.

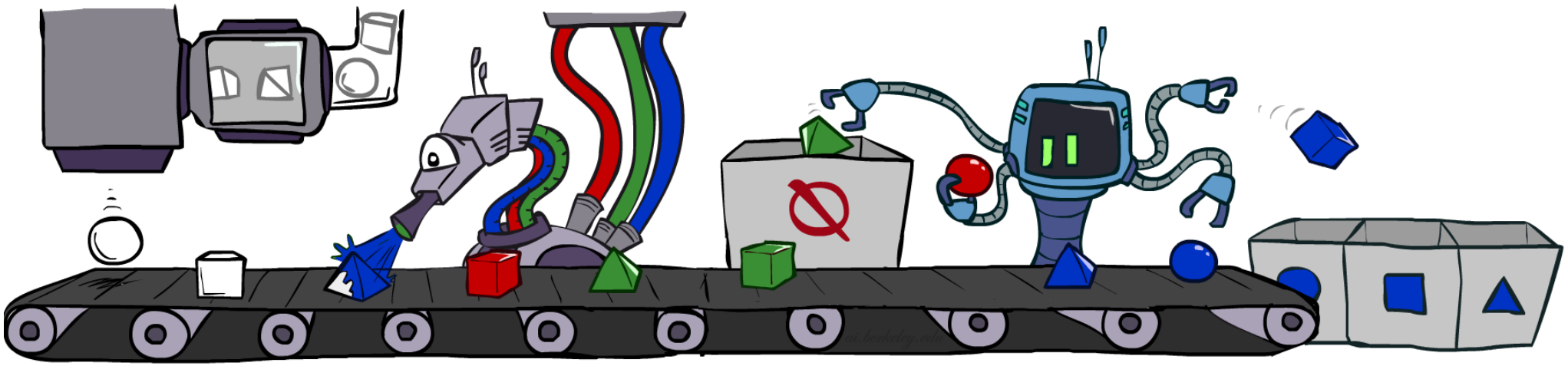


evidence不match，就quit

+c, -s, +r, +w
+c, +s, +r, +w
-c, +s, +r, -w
+c, -s, +r, +w
-c, -s, -r, +w

Rejection Sampling

- Input: evidence instantiation
- For i = 1, 2, …, n
  - Sample x_i from $P(X_i \mid \text{Parents}(X_i))$
  - If x_i not consistent with evidence
    - Reject: return – no sample is generated in this cycle
- Return $(x_1, x_2, \ldots, x_n)$



Question

- What queries can we answer with regection samples (evidence: +c)?

A P(+a,-b,+c)

B P(+a,-b|+c)

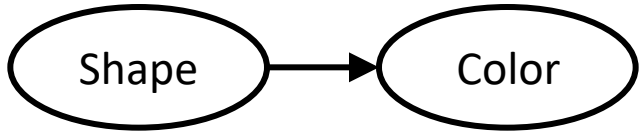
C both

D Neither

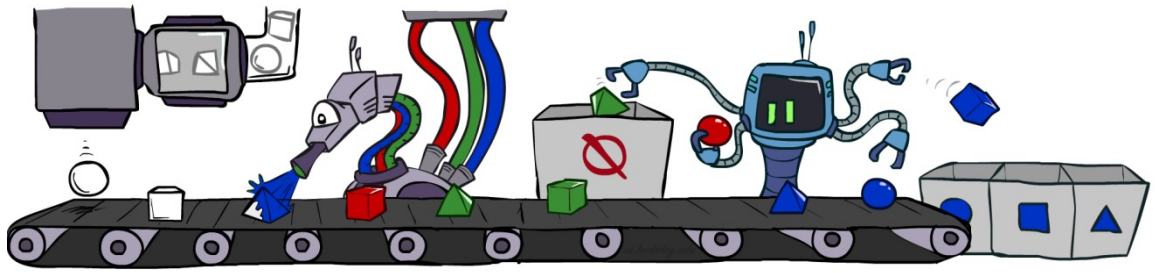
设置自己reject了多少回，就可以AB都有，否则是B

Likelihood Weighting

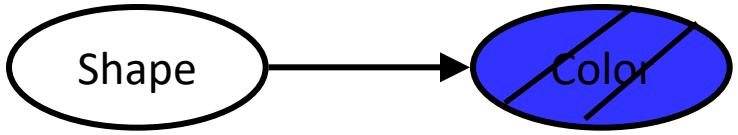
- Problem with rejection sampling:
  - If evidence is unlikely, rejects lots of samples
  - Evidence not exploited as you sample
  - Consider P(Shape | blue)



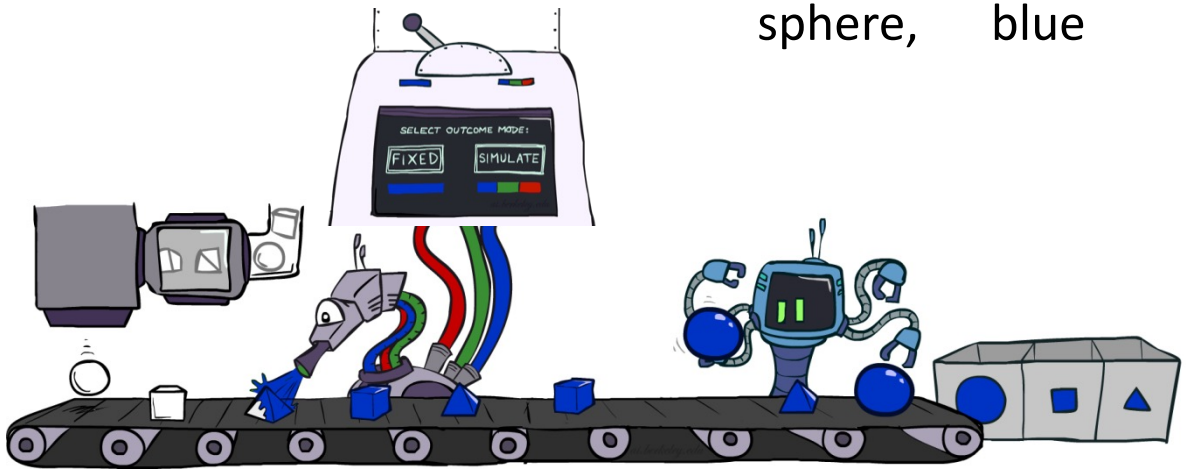
~~pyramid, green~~
~~pyramid, red~~
sphere, blue
~~cube, red~~
~~sphere, green~~



- Idea: fix evidence variables and sample the rest
  - Problem: sample distribution not consistent!
  - Solution: weight by probability of evidence given parents



pyramid, blue
pyramid, blue
sphere, blue
cube, blue
sphere, blue



Likelihood Weighting

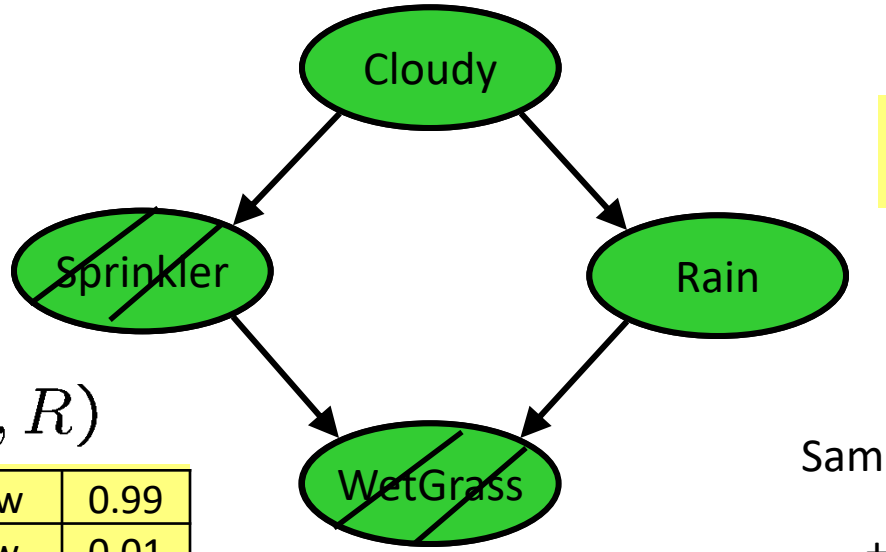
$P(C)$

+c	0.5
-c	0.5

P(R|+s,+w)

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

+c, +s, +r, +w。 0.099

…

$$w = 1.0 \times 0.1 \times 0.99$$

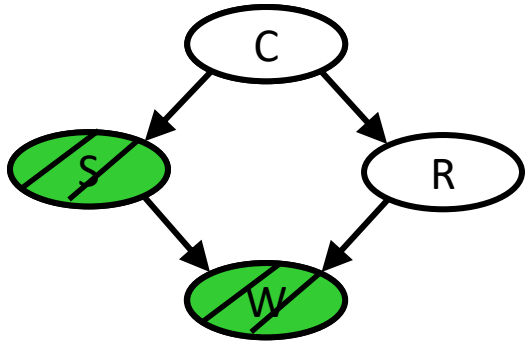
Likelihood Weighting

- Input: evidence instantiation
- w = 1.0
- for i = 1, 2, …, n
  - if X_i is an evidence variable
    - X_i = observation x_i for X_i
    - Set w = w * $P(x_i \mid Parents(X_i))$
  - else
    - Sample x_i from $P(X_i \mid Parents(X_i))$
- return $(x_1, x_2, \ldots, x_n)$, w

Likelihood Weighting

- Sampling distribution if z sampled and e fixed evidence

$$S_{WS}(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^{l} P(z_i | \text{Parents}(Z_i))$$



- Now, samples have weights

$$w(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^{m} P(e_i | \text{Parents}(E_i))$$

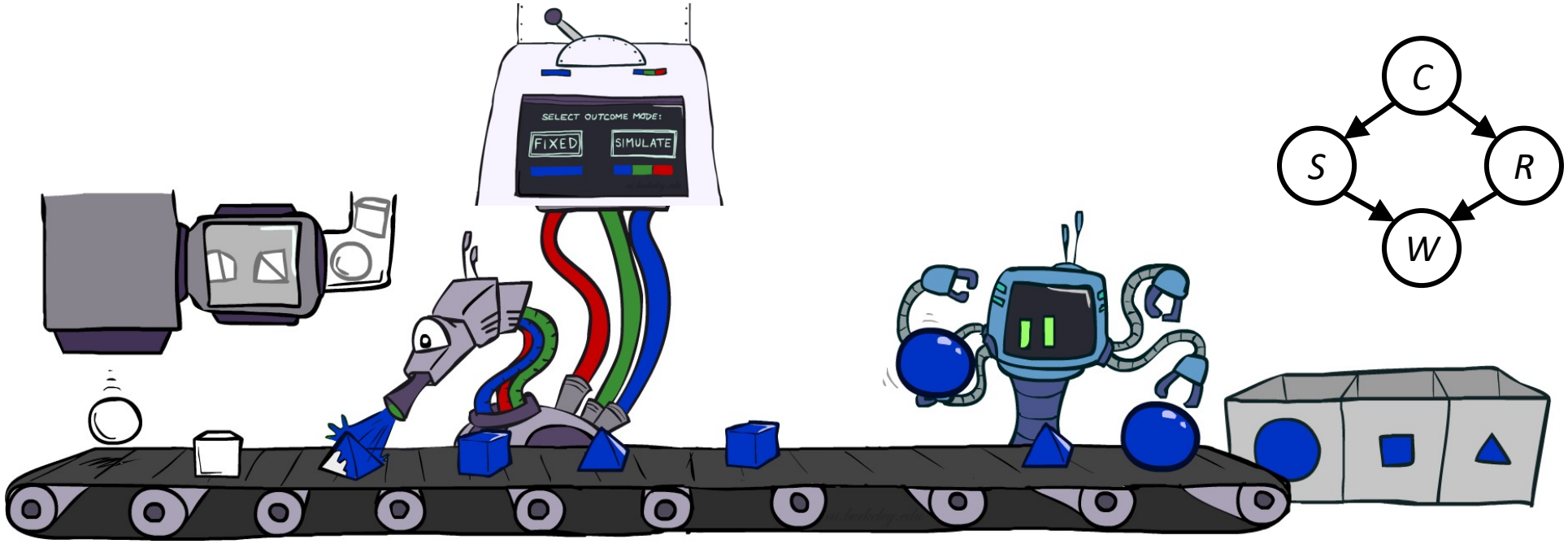
- Together, weighted sampling distribution is consistent

$$S_{\text{WS}}(z, e) \cdot w(z, e) = \prod_{i=1}^{l} P(z_i | \text{Parents}(z_i)) \prod_{i=1}^{m} P(e_i | \text{Parents}(e_i))$$

$$= P(\mathbf{z}, \mathbf{e})$$

Likelihood Weighting

- Likelihood weighting is good
  - We have taken evidence into account as we generate the sample
  - E.g. here, W's value will get picked based on the evidence values of S, R
  - More of our samples will reflect the state of the world suggested by the evidence
- Likelihood weighting doesn't solve all our problems
  - Evidence influences the choice of downstream variables, but not upstream ones
- We would like to consider evidence when we sample every variable (leads to Gibbs sampling)

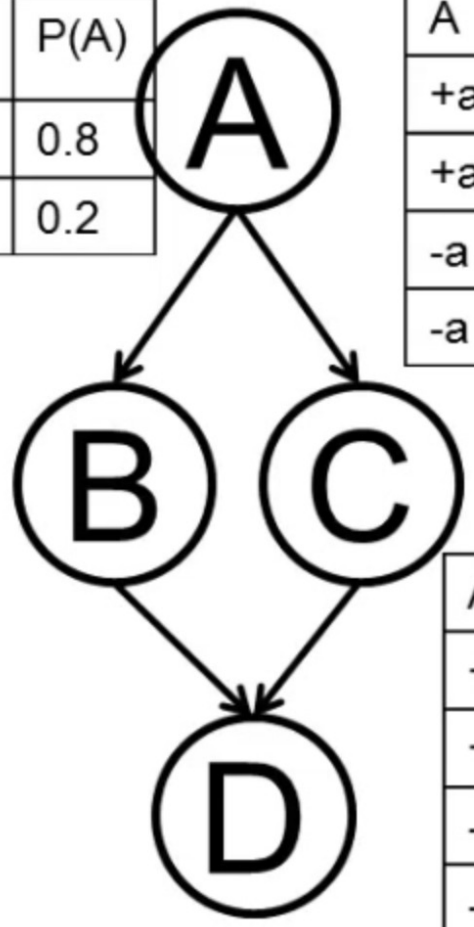


Exercise: forward sampling order of A,B,C,D

0.31	0.58	0.04	0.94	0.67	0.49	0.37	0.42	…

B	C	D	P(D|B,C)
+b	+c	+d	0.3
+b	+c	-d	0.7
+b	-c	+d	0.1
+b	-c	-d	0.9
-b	+c	+d	0.2
-b	+c	-d	0.8
-b	-c	+d	0.9
-b	-c	-d	0.1

A	P(A)
+a	0.8
-a	0.2



A	B	P(B|A)
+a	+b	0.8
+a	-b	0.2
-a	+b	0.5
-a	-b	0.5

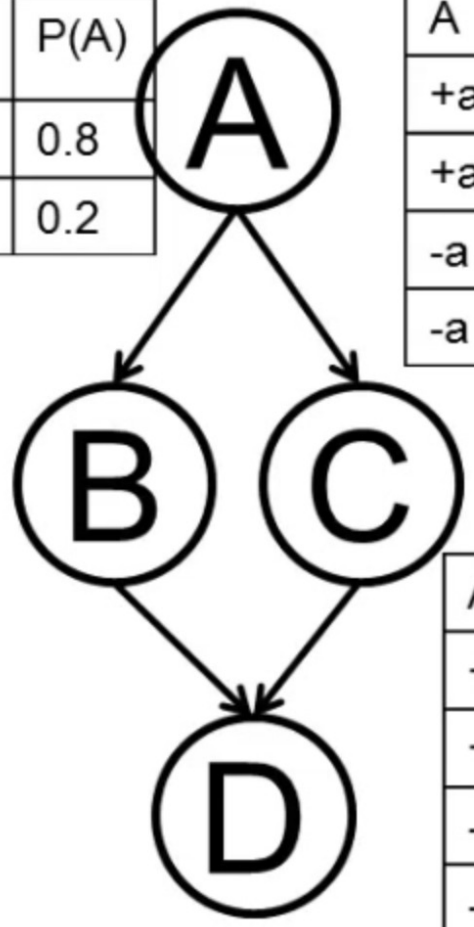
A	C	P(C|A)
+a	+c	0.7
+a	-c	0.3
-a	+c	0.1
-a	-c	0.9

Exercise: forward sampling order of A,B,C,D

0.31	**0.58**	**0.04**	**0.94**	**0.67**	**0.49**	**0.37**	**0.42**	…

B	C	D	P(D|B,C)
+b	+c	+d	0.3
+b	+c	-d	0.7
+b	-c	+d	0.1
+b	-c	-d	0.9
-b	+c	+d	0.2
-b	+c	-d	0.8
-b	-c	+d	0.9
-b	-c	-d	0.1

A	P(A)
+a	0.8
-a	0.2



A	B	P(B|A)
+a	+b	0.8
+a	-b	0.2
-a	+b	0.5
-a	-b	0.5

A	C	P(C|A)
+a	+c	0.7
+a	-c	0.3
-a	+c	0.1
-a	-c	0.9

+a, +b, +c, -d
+a, +b, +c, -d

Answering queries from samples

Samples

+a	-b	-c	-d
-a	+b	+c	-d
+a	-b	+c	-d
+a	+b	-c	-d
+a	-b	+c	+d
-a	-b	-c	+d
-a	-b	-c	-d
+a	+b	+c	-d
-a	+b	-c	-d
+a	+b	-c	+d

Queries

P(+d) 3/10

P(+a, -b)

P(-a, -b, -c, -d)

P(-c | -d)

P(+d | -a, -b)

Answering queries from samples

Samples

+a	-b	-c	-d
-a	+b	+c	-d
+a	-b	+c	-d
+a	+b	-c	-d
+a	-b	+c	+d
-a	-b	-c	+d
-a	-b	-c	-d
+a	+b	+c	-d
-a	+b	-c	-d
+a	+b	-c	+d

Queries

$P(+d)$ ~ #samples of d/ total # of samples
= 3/10

$P(+a, -b)$ ~ #samples of (+a,-b)/ total # of samples
= 3/10

$P(-a, -b, -c, -d)$ ~ #samples of joint / total # of samples
= 1/10

$P(-c \mid -d)$ ~ #samples of (query, evid)/ total # of evidence
= 4/7

$P(+d \mid -a, -b)$ ~ #samples of (query, evid)/ total # of evidence
= 1/2

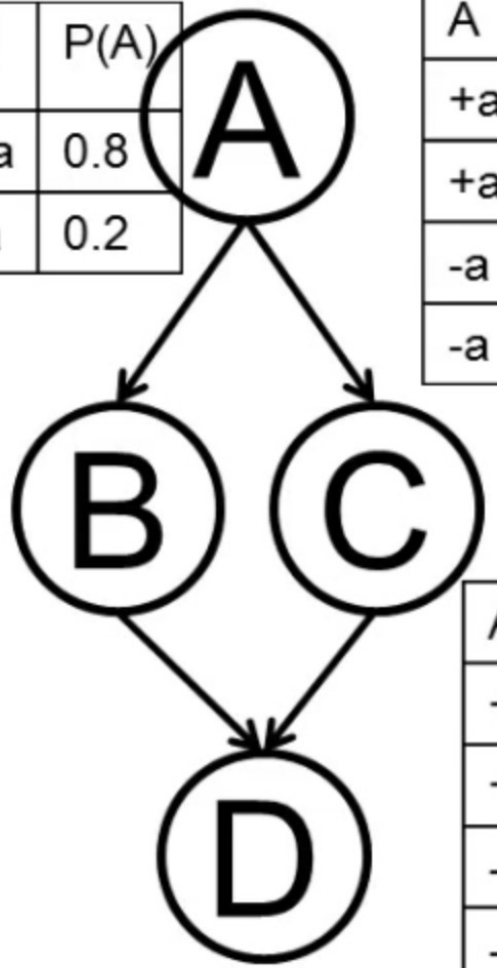
Exercise: Rejection sampling for P(-d|-b), ordering ABCD

+b，rejected　　–b　　+c　　–d

0.31	0.58	0.04	0.94	0.67	0.49	0.37	0.42	…

B	C	D	P(D|B,C)
+b	+c	+d	0.3
+b	+c	-d	0.7
+b	-c	+d	0.1
+b	-c	-d	0.9
-b	+c	+d	0.2
-b	+c	-d	0.8
-b	-c	+d	0.9
-b	-c	-d	0.1

A	P(A)
+a	0.8
-a	0.2



A	B	P(B|A)
+a	+b	0.8
+a	-b	0.2
-a	+b	0.5
-a	-b	0.5

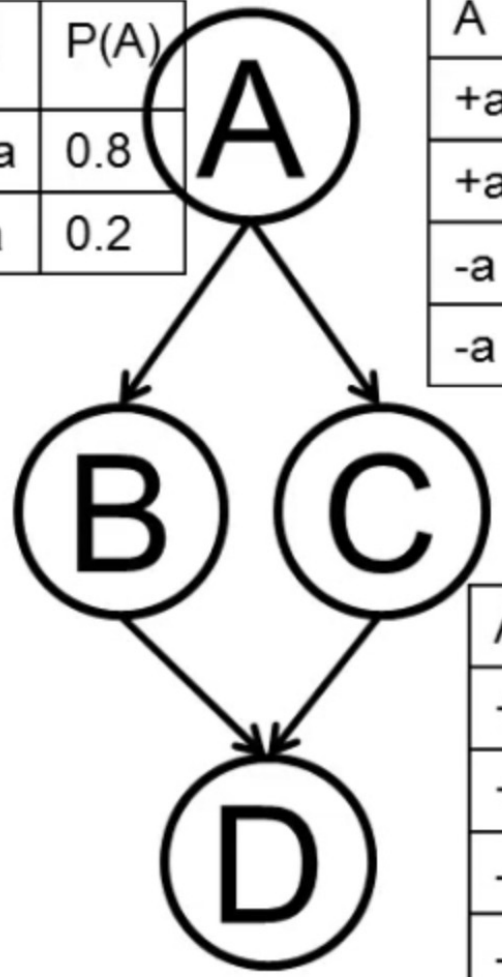
A	C	P(C|A)
+a	+c	0.7
+a	-c	0.3
-a	+c	0.1
-a	-c	0.9

Exercise: Likelihood weighting for P(-d|-a,-b), ordering ABCD

0.31	0.58	0.04	0.94	0.67	0.49	0.37	0.42	…

B	C	D	P(D|B,C)
+b	+c	+d	0.3
+b	+c	-d	0.7
+b	-c	+d	0.1
+b	-c	-d	0.9
-b	+c	+d	0.2
-b	+c	-d	0.8
-b	-c	+d	0.9
-b	-c	-d	0.1

A	P(A)
+a	0.8
-a	0.2



A	B	P(B|A)
+a	+b	0.8
+a	-b	0.2
-a	+b	0.5
-a	-b	0.5

A	C	P(C|A)
+a	+c	0.7
+a	-c	0.3
-a	+c	0.1
-a	-c	0.9

1.0 * 0.2 然后要0.31

–c, +d

Answering Queries from Weighted Samples

Samples				权重
+a	-b	-c	-d	0.3
-a	+b	+c	-d	0.4
+a	-b	+c	-d	0.1
+a	+b	-c	-d	0.3
+a	-b	+c	+d	0.4
-a	-b	-c	+d	0.1
-a	-b	-c	-d	0.2
+a	+b	+c	-d	0.5
-a	+b	-c	-d	0.7
+a	+b	-c	+d	0.8

Queries

P(+d)

P(+a, -b)

P(-a, -b, -c, -d)

P(-c | -d)

2.5

P(+d | -a, -b)

Answering Queries from Weighted Samples

Samples

+a	-b	-c	-d	0.3
-a	+b	+c	-d	0.4
+a	-b	+c	-d	0.1
+a	+b	-c	-d	0.3
+a	-b	+c	+d	0.4
-a	-b	-c	+d	0.1
-a	-b	-c	-d	0.2
+a	+b	+c	-d	0.5
-a	+b	-c	-d	0.7
+a	+b	-c	+d	0.8

Queries

$P(+d)$ ~ sum of samples of +d/ total sum of samples
= 1.3/3.8

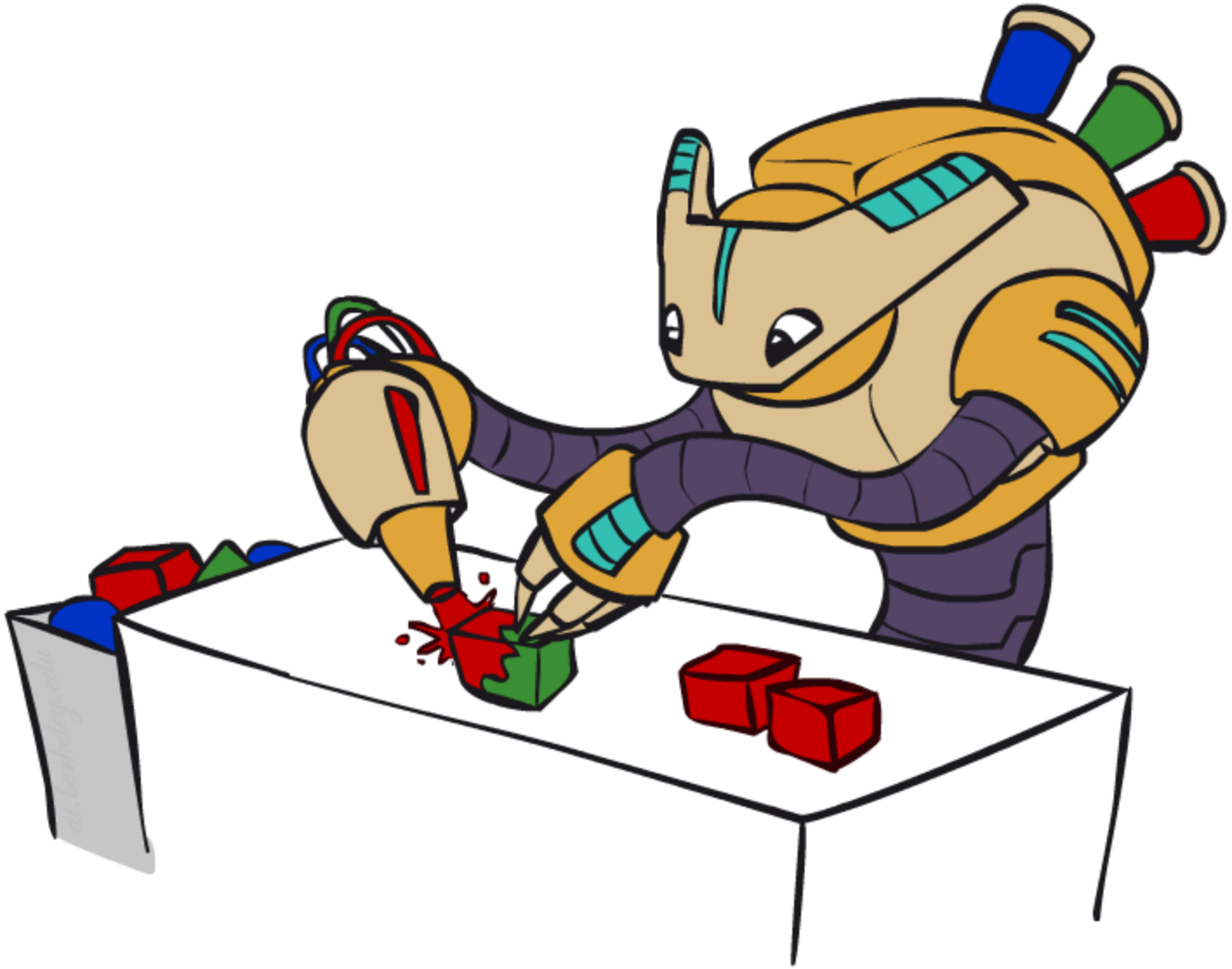
$P(+a, -b)$ ~ sum of samples of (+a,-b)/ total sum of samples
= 0.8/3.8

$P(-a, -b, -c, -d)$ = 0.2/3.8

$P(-c \mid -d)$ ~ sum of samples of (-c,-d)/ sum of samples with (-d)
= (0.3+0.3+0.2+0.7)/2.5

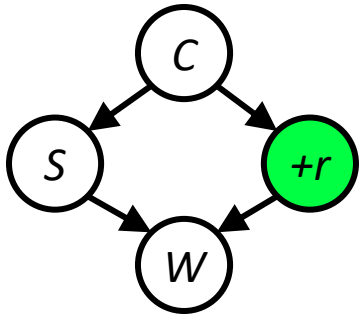
$P(+d \mid -a, -b)$ = 0.1/0.3

Gibbs Sampling

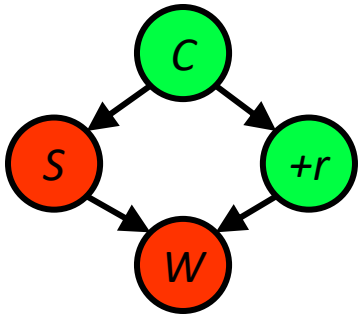


Gibbs Sampling Example: P(S | +r)

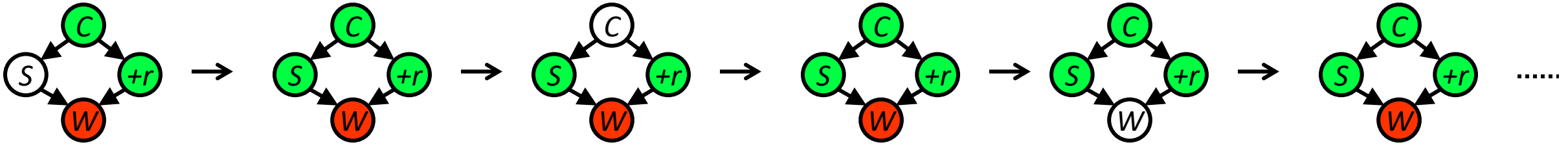
- Step 1: Fix evidence
  - R = +r



- Step 2: Initialize other variables
  - Randomly



- Steps 3: Repeat
  - Choose a non-evidence variable X
  - Resample X from P(X | all other variables)



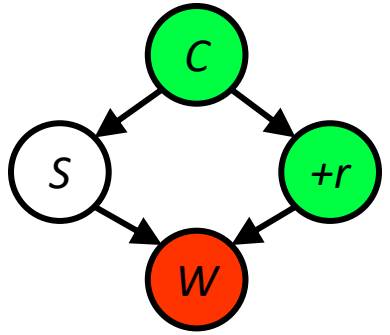
Sample from $P(S|+c,-w,+r)$ Sample from $P(C|+s,-w,+r)$ Sample from $P(W|+s,+c,+r)$

Gibbs Sampling

- *Procedure:* keep track of a full instantiation $x_1, x_2, \ldots, x_n$. Start with an arbitrary instantiation consistent with the evidence. Sample one variable at a time, conditioned on all the rest, but keep evidence fixed. Keep repeating this for a long time.
- *Property:* in the limit of repeating this infinitely many times the resulting samples come from the correct distribution (i.e. conditioned on evidence, nonzero Condition distribution).
- *Rationale*: both upstream and downstream variables condition on evidence.
- In contrast: likelihood weighting only conditions on upstream evidence, and hence weights obtained in likelihood weighting can sometimes be very small. Sum of weights over all samples is indicative of how many "effective" samples were obtained, so we want high weight.

Efficient Resampling of One Variable

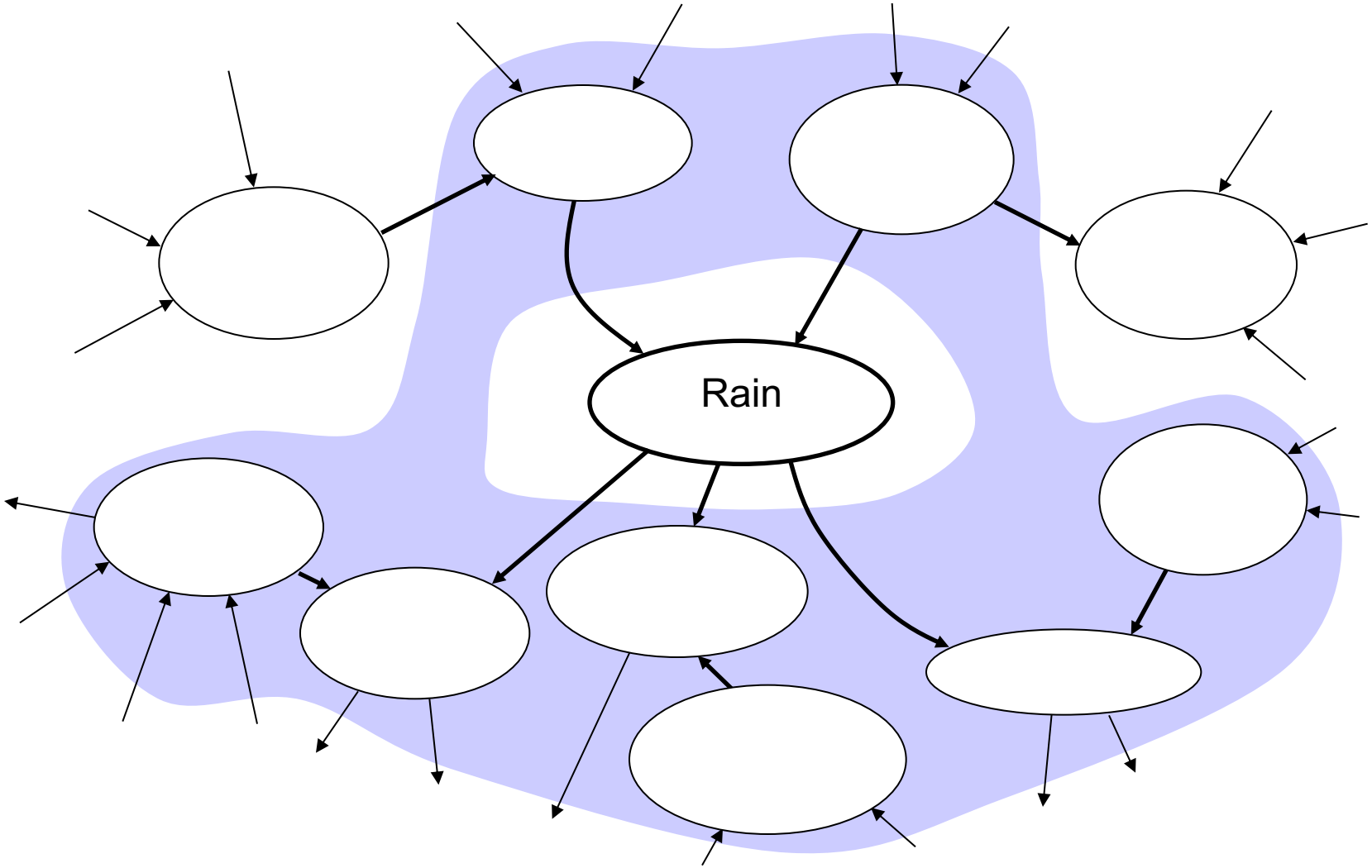
- Sample from P(S | +c, +r, -w)



$$
\begin{aligned}
P(S|+c,+r,-w) &= \frac{P(S,+c,+r,-w)}{P(+c,+r,-w)} \\
&= \frac{P(S,+c,+r,-w)}{\sum_s P(s,+c,+r,-w)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{\sum_s P(+c)P(s|+c)P(+r|+c)P(-w|s,+r)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{P(+c)P(+r|+c)\sum_s P(s|+c)P(-w|s,+r)} \\
&= \frac{P(S|+c)P(-w|S,+r)}{\sum_s P(s|+c)P(-w|s,+r)}
\end{aligned}
$$

- Many things cancel out – only CPTs with S remain!
- More generally: only CPTs that have resampled variable need to be considered, and joined together

Markov Blanket



Strongly relevant features *(Kohavi-John, 1997)* ⇔ Markov Blanket *(Tsamardinos-Aliferis, 2003)*

Gibbs Sampling

只算了少量的，没有算全局

- How is this better than sampling from the full joint?
  - In a Bayes' Net, sampling a variable given all the other variables (e.g. P(R|S,C,W)) is usually much easier than sampling from the full joint distribution
  - Only requires a join on the variable to be sampled (in this case, a join on R)
  - The resulting factor only depends on the variable's parents, its children, and its children's parents (this is often referred to as its Markov blanket)

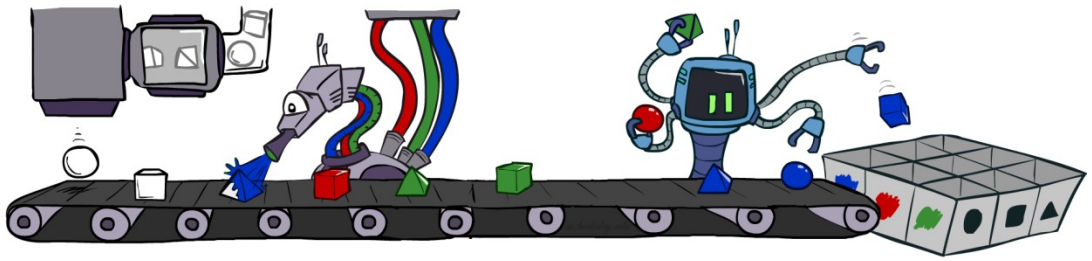
https://blog.csdn.net/step_forward_ML/article/details/80333166

Further Reading on Gibbs Sampling

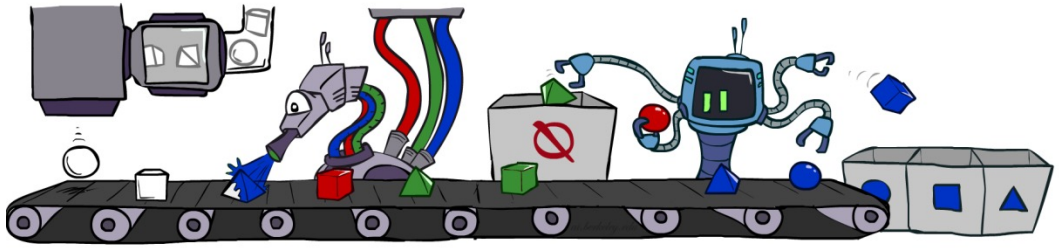
- Gibbs sampling produces sample from the query distribution P(Q | e) in limit of re-sampling infinitely often
- Gibbs sampling is a special case of more general methods called Markov chain Monte Carlo (MCMC) methods
  - Metropolis-Hastings is one of the more famous MCMC methods (in fact, Gibbs sampling is a special case of Metropolis-Hastings)
- You may read about Monte Carlo methods – they're just sampling

Bayes' Net Sampling Summary

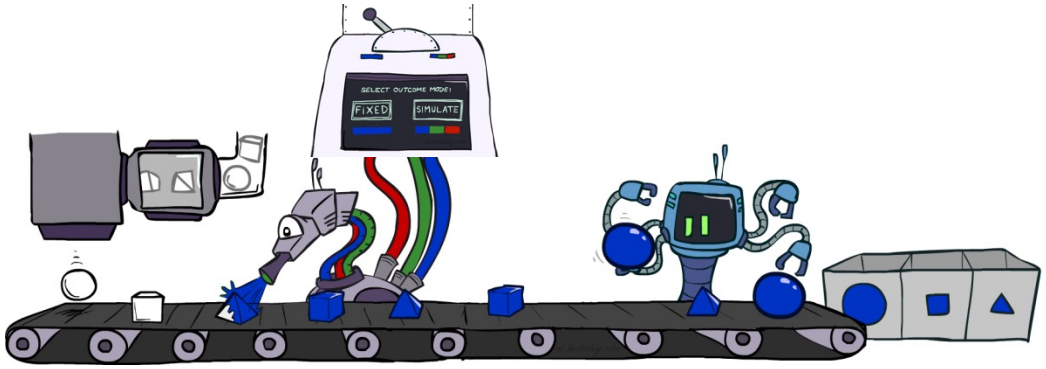
- Prior Sampling P(Q)



- Rejection Sampling P(Q | e)



- Likelihood Weighting P(Q | e)



- Gibbs Sampling P(Q | e)