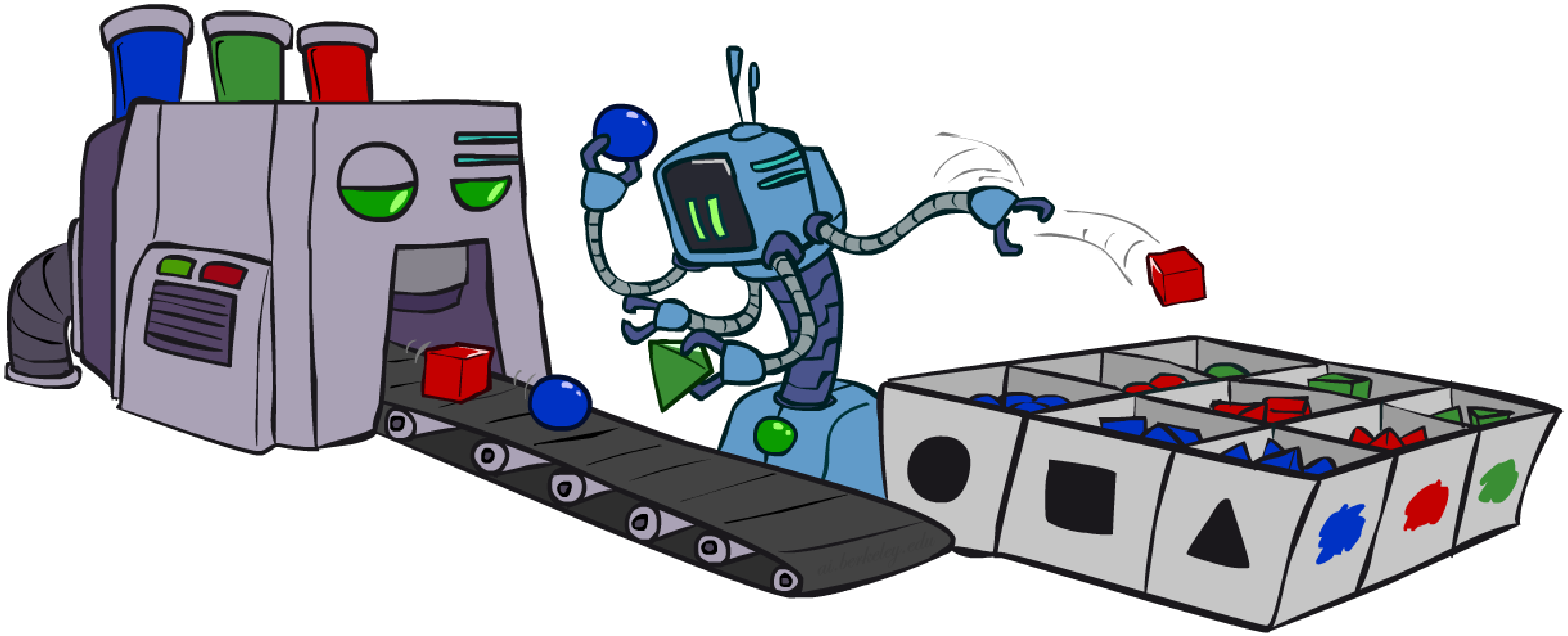
CS 6300: Artificial Intelligence

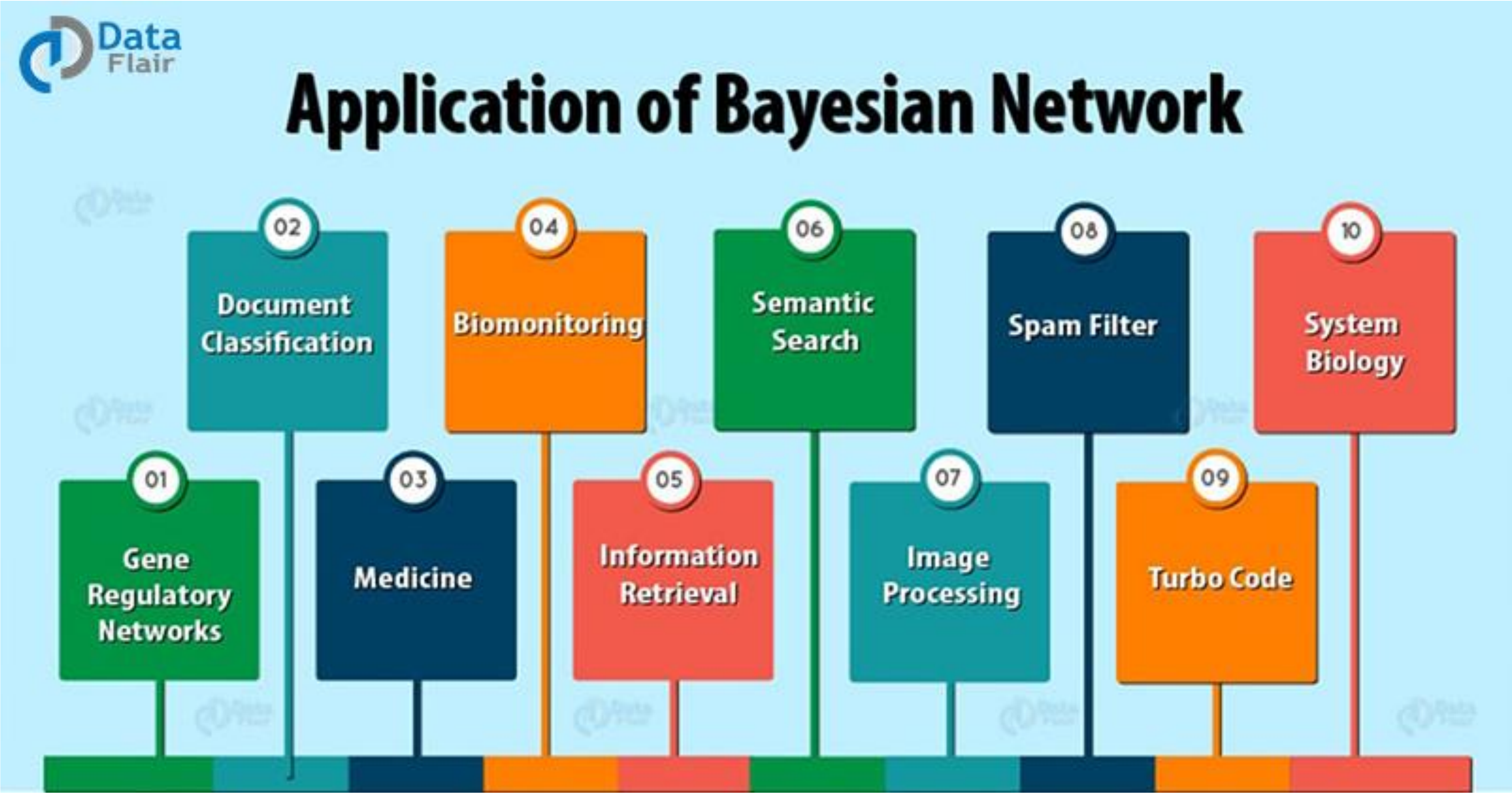
Bayes’ Nets: Sampling



Instructor: Daniel Brown --- University of Utah

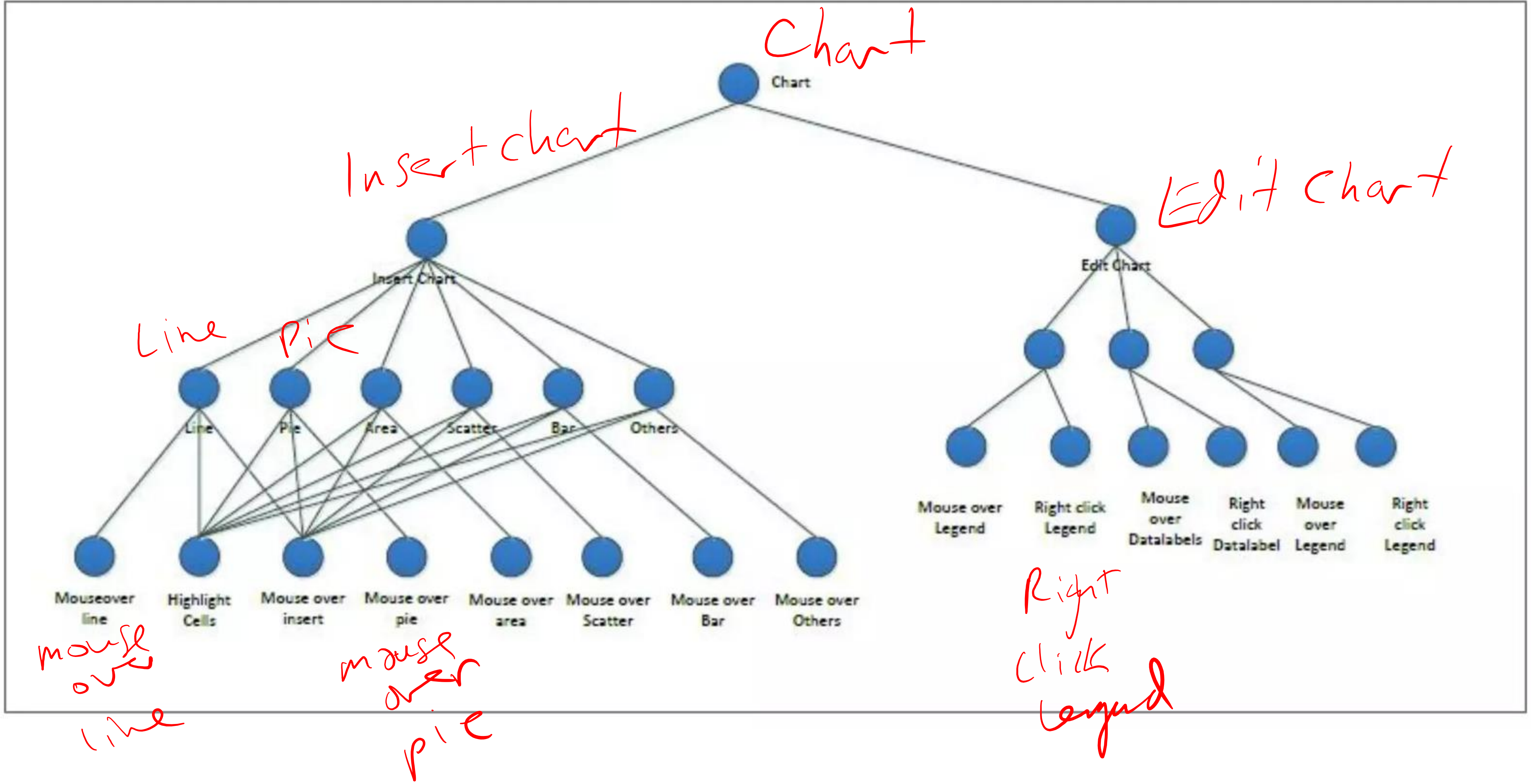
[Based on slides created by Dan Klein and Pieter Abbeel http://ai.berkeley.edu.]

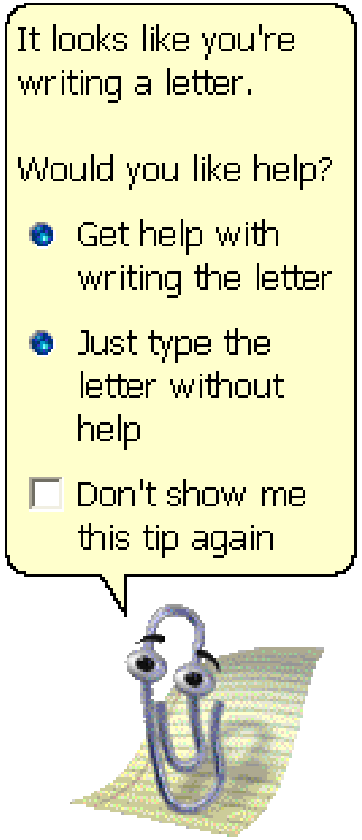
Where are Bayes’ Nets used?



https://data-flair.training/blogs/bayesian-network-applications/

(Old-school) Bayes' Net Application: Clippy





It looks like you're writing a letter.

Would you like help?

● Get help with writing the letter

● Just type the letter without help

□ Don't show me this tip again

Bayes’ Net Application: Spam Filtering

- Input: an email
- Output: spam/ham

- Setup:
  - Get a large collection of example emails, each labeled “spam” or “ham”
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future emails

- Features: The attributes used to make the ham / spam decision
  - Words: FREE!
  - Text Patterns: $dd, CAPS
  - Non-text: SenderInContacts
  - …

✗

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

✗

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES
FOR ONLY $99

✓

Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

Bayes’ Net Application: Spam Filtering

- Model-based approach
  - Build a model (e.g. Bayes’ net) where both the label and features are random variables
  - Instantiate any observed features
  - Query for the distribution of the label conditioned on the features

- Challenges
  - What structure should the BN have?
  - How should we learn its parameters?

Naïve Bayes

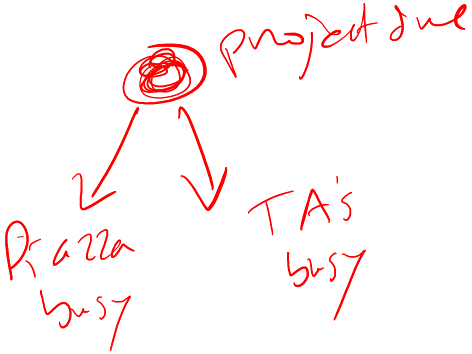
- A general Naive Bayes model:

|Y| parameters

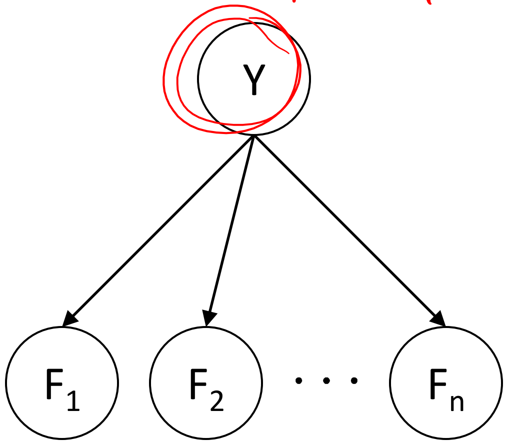
$$P(\mathsf{Y}, \mathsf{F}_1 \dots \mathsf{F}_n) = P(\mathsf{Y}) \prod_i P(\mathsf{F}_i|\mathsf{Y})$$

$|Y|$ x $|F|^n$ values

n x |F| x |Y| parameters



label = {spam, ham}



- We only have to specify how each feature depends on the class
- Total number of parameters is *linear* in n
- Model is very simplistic, but often works anyway
- Assumes all features are independent effects of the label Y (very Naïve, but very efficient)

Inference for Naïve Bayes

"free"

$p(\text{spam})\, P(f_1 | \text{spam}) \ldots$

$p(\text{ham})\, P(\text{"free"} | \text{ham}) \ldots$

- Goal: compute posterior distribution over label variable Y
  - Step 1: get joint probability of label and evidence for each label

K = # classes = 2 for Spam/Ham

$$P(Y|f_1 \ldots f_n) \propto \begin{bmatrix} P(y_1, f_1 \ldots f_n) \\ P(y_2, f_1 \ldots f_n) \\ \vdots \\ P(y_k, f_1 \ldots f_n) \end{bmatrix} \Rightarrow \frac{\begin{bmatrix} P(y_1) \prod_i P(f_i|y_1) \\ P(y_2) \prod_i P(f_i|y_2) \\ \vdots \\ P(y_k) \prod_i P(f_i|y_k) \end{bmatrix}}{P(f_1 \ldots f_n)}$$

$$= \frac{P(y, f_1 \ldots f_n)}{p(f_1 \ldots f_n)}$$

+

  - Step 2: sum to get probability of evidence

$$\Downarrow$$

  - Step 3: normalize by dividing Step 1 by Step 2

$$P(Y|f_1 \ldots f_n)$$

General Naïve Bayes

- What do we need in order to use Naïve Bayes?
  - Inference method (we just saw this part)
    - Start with a bunch of probabilities: P(Y) and the $P(F_i|Y)$ tables
    - Use standard inference to compute $P(Y|F_1 \ldots F_n)$
    - Nothing new here
  - Estimates of local conditional probability tables
    - P(Y), the prior over labels
    - $P(F_i|Y)$ for each feature (evidence variable)
    - These probabilities are collectively called the *parameters* of the model and denoted by θ
    - Up until now, we assumed these appeared by magic, but…
    - …they typically come from some training data we collect

Spam Filtering with Bag of Words Assumption

- Model: $P(Y, W_1 \ldots W_n) = P(Y) \prod_i P(W_i|Y)$
- Bag of Words: Assumes each word position is identically distributed (ignores ordering)
- What are the parameters?

$P(Y)$

ham :	0.66
spam:	0.33

$P(W|\text{spam})$

the :	0.0156
to :	0.0153
and :	0.0115
of :	0.0095
you :	0.0093
a :	0.0086
with:	0.0080
from:	0.0075
...	

$P(W|\text{ham})$

the :	0.0210
to :	0.0133
of :	0.0119
2002:	0.0110
with:	0.0108
from:	0.0107
and :	0.0105
a :	0.0100
...	

$P(y|f_1 \ldots f_n) = P(y) \prod_i P(f_i|y)$

Spam Example

$\Rightarrow$ log space $\log P(y) + \sum \log P(f_i|y)$

$P(\text{Gary}|y))$

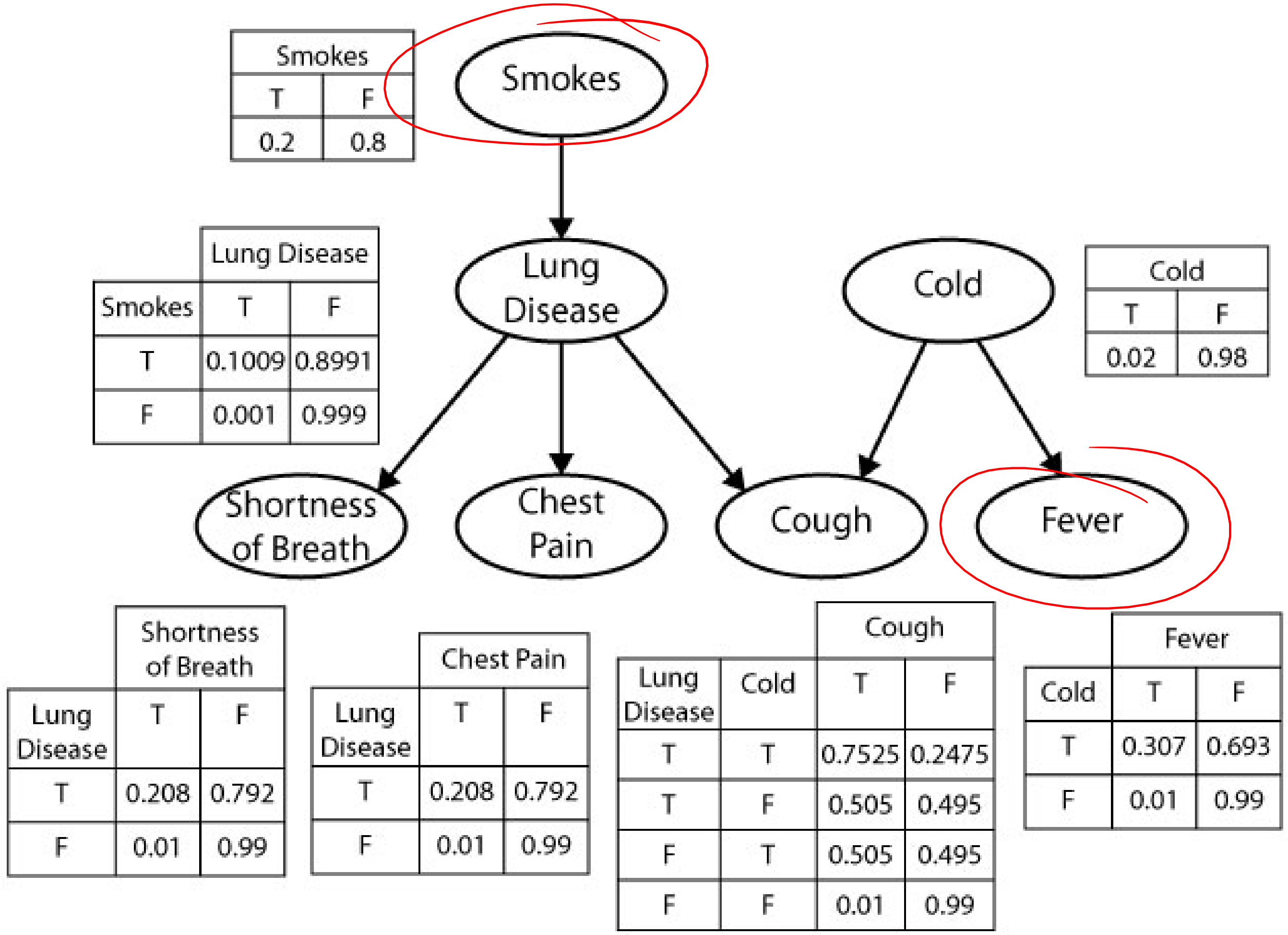
Word	P(w|spam)	P(w|ham)	Tot Spam	Tot Ham
(prior)	0.33333	0.66666	-1.1	-0.4

log P

$$\frac{\exp(-76)}{\exp(-76)+\exp(-86.5)}$$

P(spam | w) = 98.9

Medical Diagnosis



Smokes	
T	F
0.2	0.8

	Lung Disease	
Smokes	T	F
T	0.1009	0.8991
F	0.001	0.999

Cold	
T	F
0.02	0.98

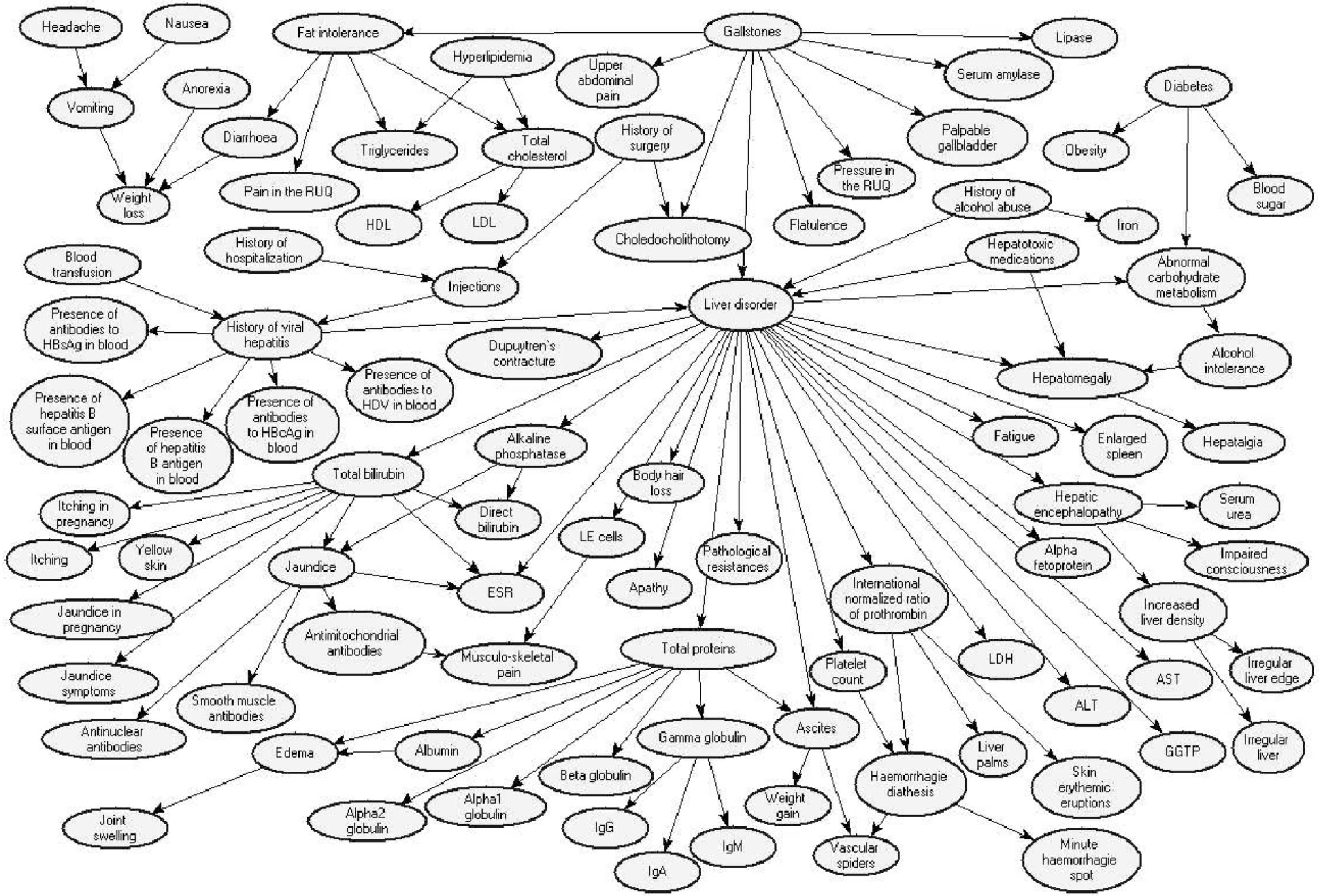
	Shortness of Breath	
Lung Disease	T	F
T	0.208	0.792
F	0.01	0.99

	Chest Pain	
Lung Disease	T	F
T	0.208	0.792
F	0.01	0.99

		Cough	
Lung Disease	Cold	T	F
T	T	0.7525	0.2475
T	F	0.505	0.495
F	T	0.505	0.495
F	F	0.01	0.99

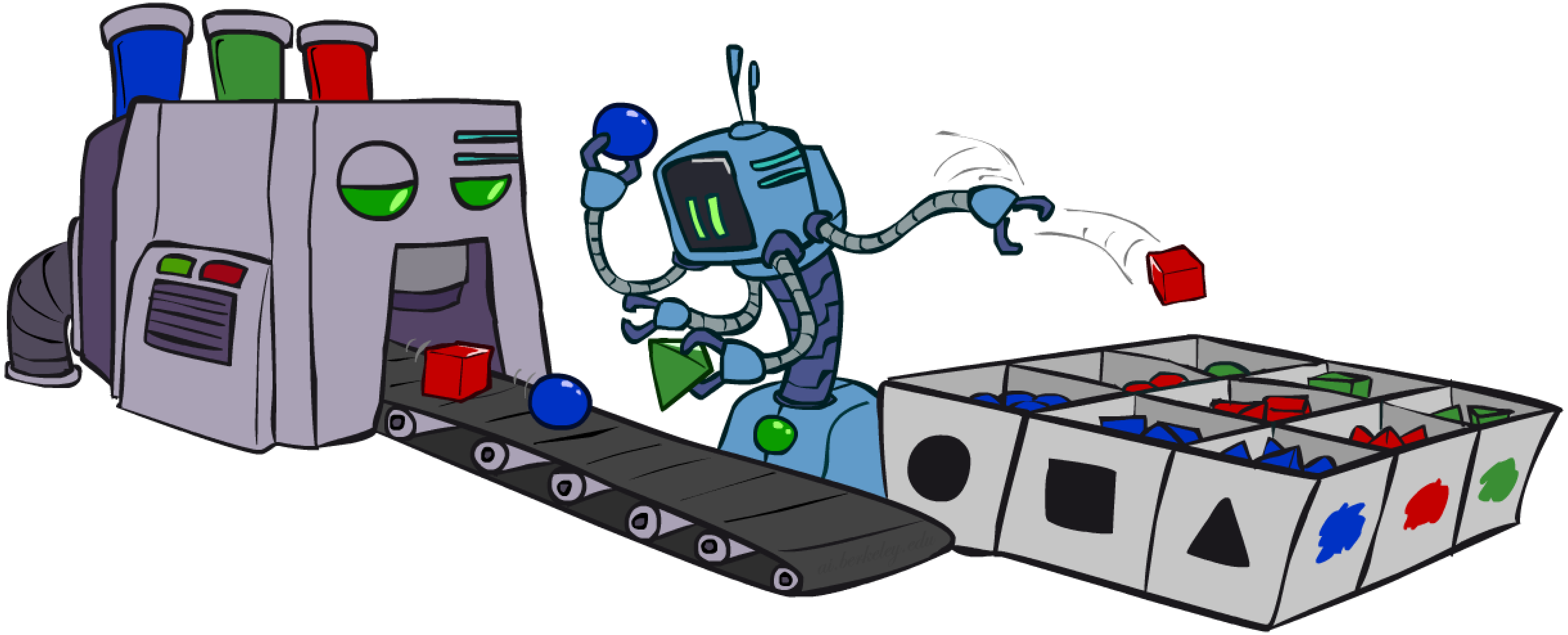
	Fever	
Cold	T	F
T	0.307	0.693
F	0.01	0.99

Medical Diagnosis



CS 6300: Artificial Intelligence

Bayes' Nets: Sampling



Instructor: Daniel Brown --- University of Utah

[Based on slides created by Dan Klein and Pieter Abbeel http://ai.berkeley.edu.]

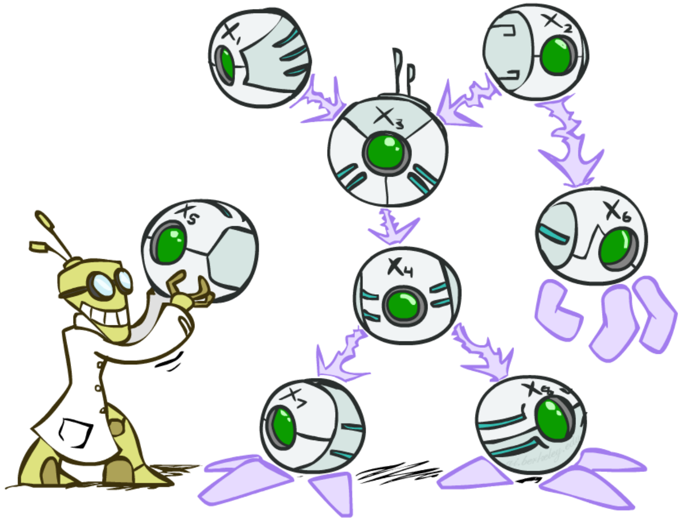
Bayes' Net Representation

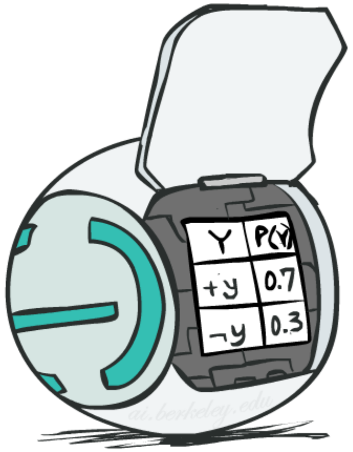
- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over X, one for each combination of parents' values

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

- 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:

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



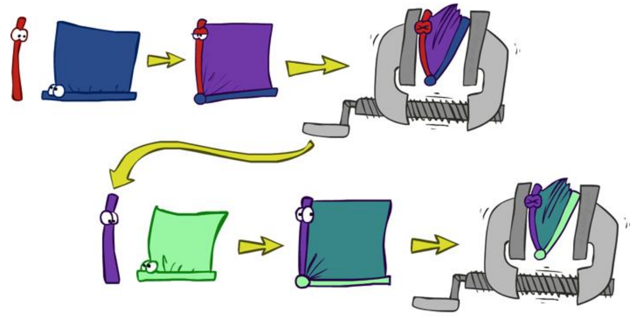


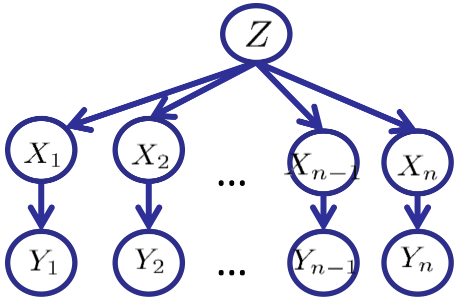
Bayes' Nets

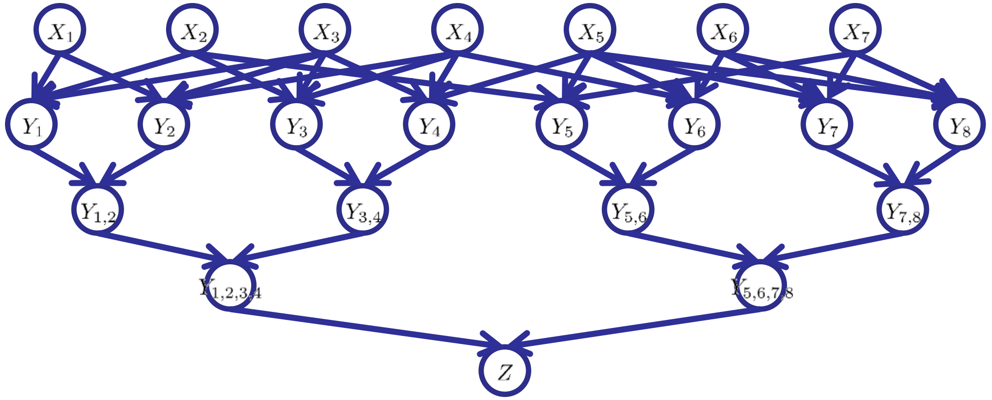
- ✓ Representation
- ✓ Conditional Independences
- Probabilistic Inference
  - ✓ Enumeration (exact, exponential complexity)
  - ✓ Variable elimination (exact, worst-case exponential complexity, often better)
  - ✓ Inference is NP-complete
  - Sampling (approximate)

Variable Elimination

- Interleave joining and marginalizing
- d^k entries computed for a factor over k variables with domain sizes d
- Ordering of elimination of hidden variables can affect size of factors generated
- Worst case: running time exponential in the size of the Bayes' net







Approximate Inference: Sampling

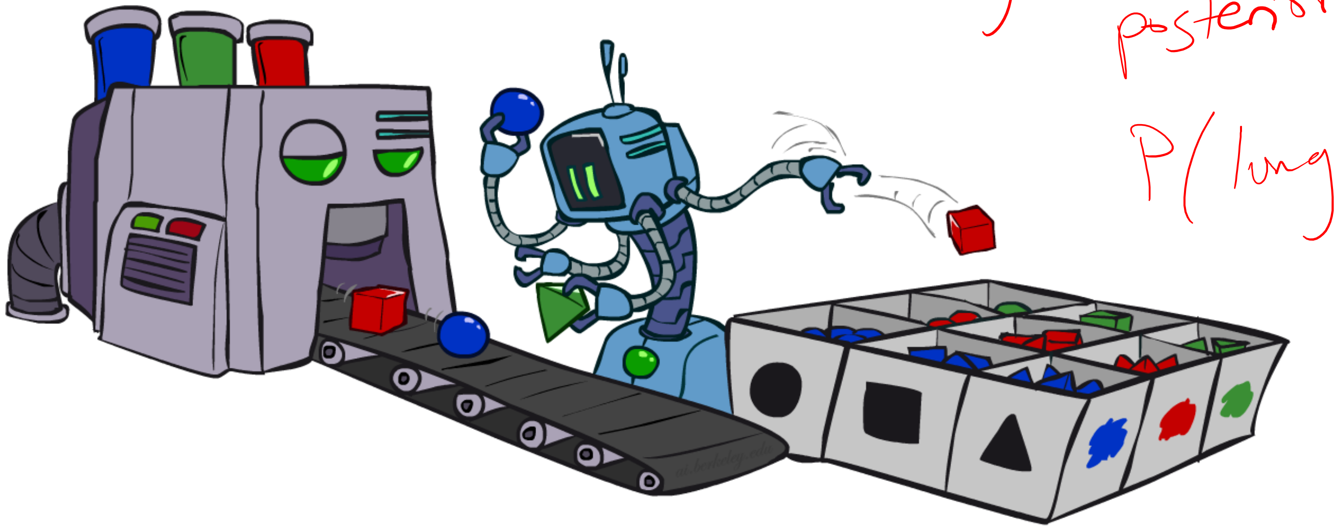
- Sampling is a lot like repeated simulation
  - Predicting the weather, basketball games, …
- Basic idea
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior probability
  - Show this converges to the true probability P

- 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)

P(lung cancer) prior

posterior

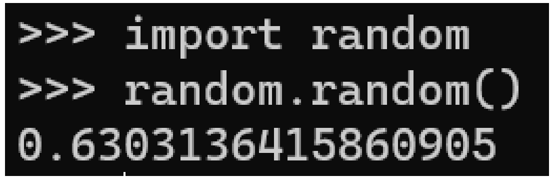
P(lung cancer | smoker, fever)



Sampling

- Sampling from given distribution
  - Step 1: Get sample *u* from uniform distribution over [0, 1)

Assume



```
>>> import random
>>> random.random()
0.6303136415860905
```

  - Step 2: Convert this sample *u* into an outcome for the given distribution by having each outcome associated with a sub-interval of [0,1) with sub-interval size equal to probability of the outcome

- Example

$u \sim U[0,1]$

0.329....

0.9

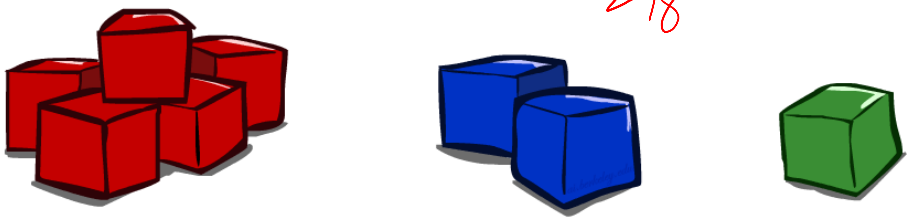
C	P(C)
red	0.6
green	0.1
blue	0.3

$0 \leq u < 0.6, \rightarrow C = red$

$0.6 \leq u < 0.7, \rightarrow C = green$

$0.7 \leq u < 1, \rightarrow C = blue$

  - If random() returns *u* = 0.83, then our sample is *C* = blue
  - E.g, after sampling 8 times:



5/8

2/8

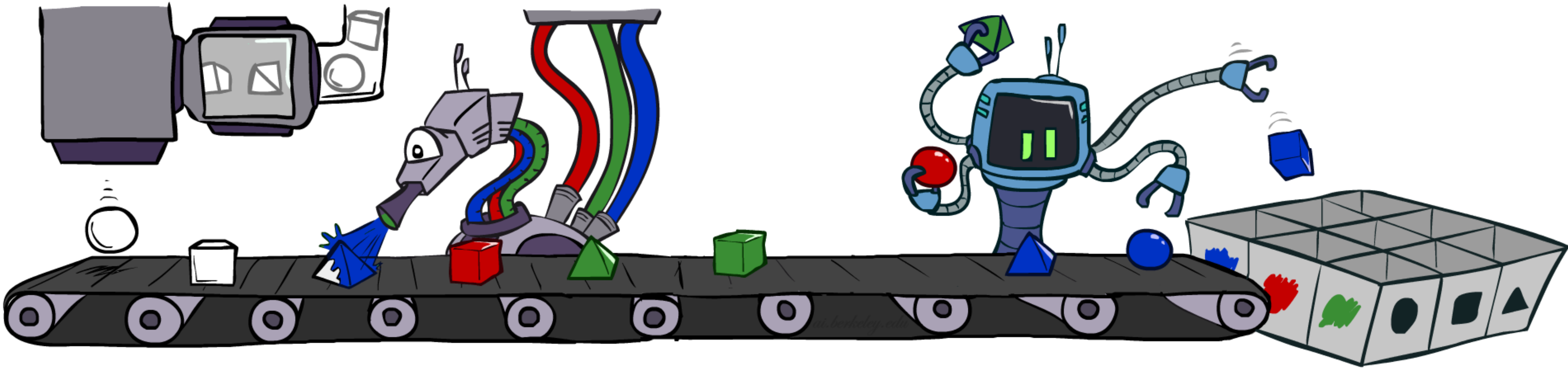
1/8

Sampling in Bayes’ Nets

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

Prior Sampling

- Sample a bunch of samples using the conditional probability tables.
- Then use these samples to compute any desired probabilistic query.



Prior Sampling

0.623

samples from U[0,1]

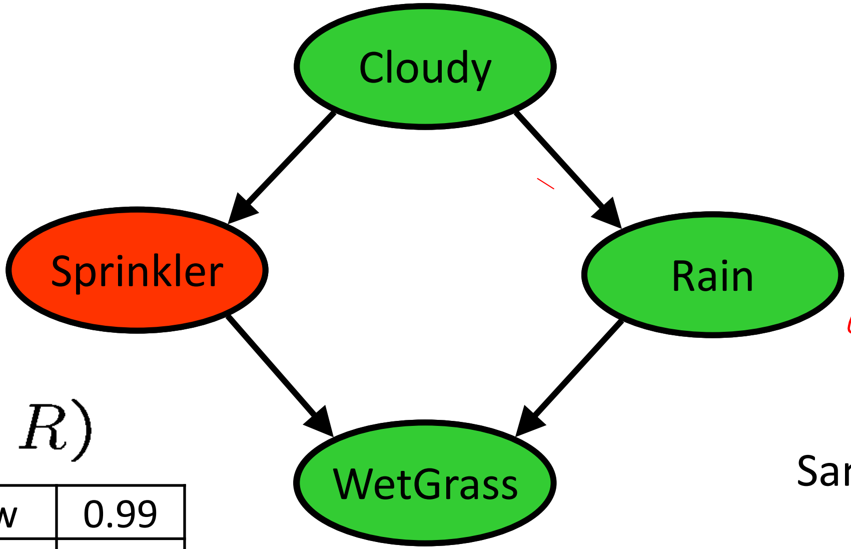
0.4, 0.3, 0.72, 0.012

$P(C)$

+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 ~P(C,S,R,W)

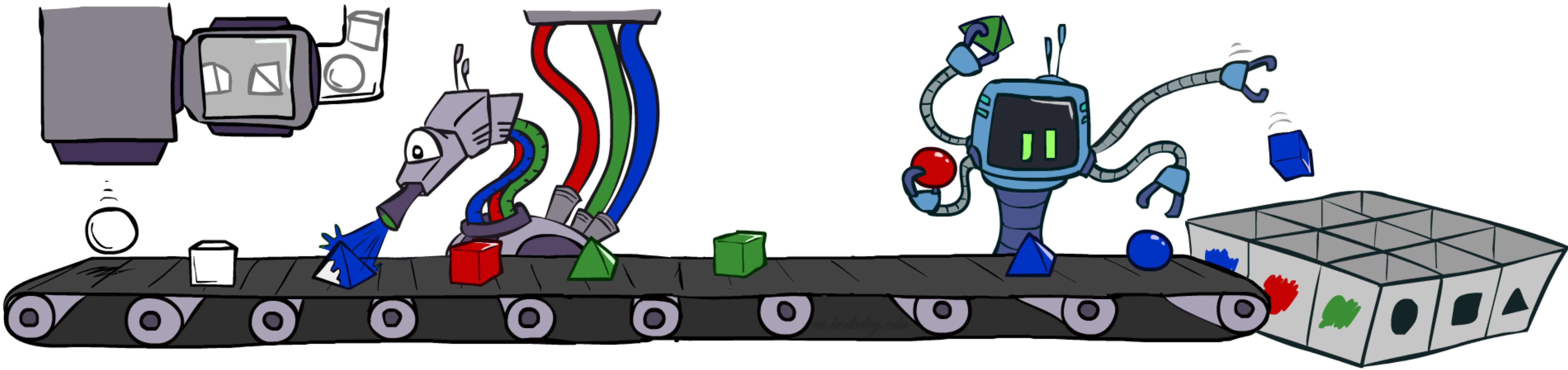
-c, +s, -r, +w

…

Never need to actually compute the full joint distribution!

Prior Sampling

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



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

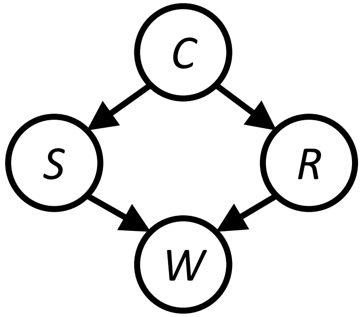
- Let the number of samples of an event be $N_{PS}(x_1 \ldots x_n)$

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

- I.e., the sampling procedure is consistent

Example

- We' ll get a bunch of samples from the BN:
  - +c, -s, +r, +w
  - +c, +s, +r, +w
  - -c, +s, +r, -w
  - +c, -s, +r, +w
  - -c, -s, -r, +w



$$P(+c \mid +w) = \frac{P(+c, +w)}{P(+w)} = \frac{3/\cancel{5}}{4/\cancel{5}} = 3/4$$

- What is P(W)?
  - We have counts <+w:4, -w:1>
  - Normalize to get P(W) = <+w:0.8, -w:0.2>
  - This will get closer to the true distribution with more samples
  - Can estimate anything else, too
  - Practice: What about P(C| +w)?   P(C| +r, +w)? P(C| -r, -w)?

$P(+c \mid +w) = 3/4$ $P(+c \mid +r, +w) = 3/3$

Example

- We’ll get a bunch of samples from the BN:

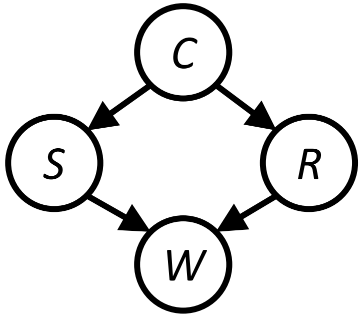
  +c, -s, +r, +w

  +c, +s, +r, +w

  -c, +s, +r, -w

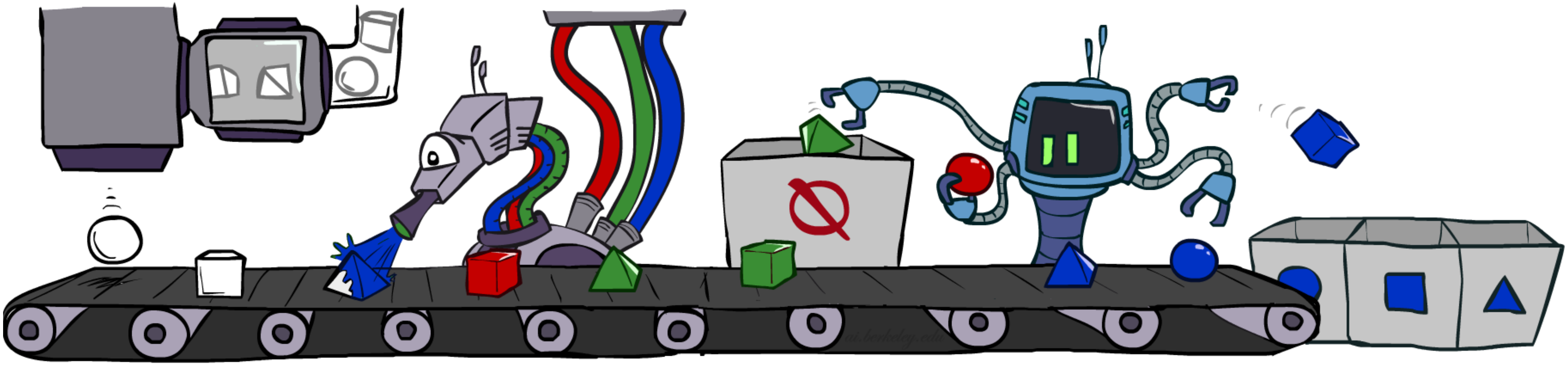
  +c, -s, +r, +w

  -c, -s, -r, +w



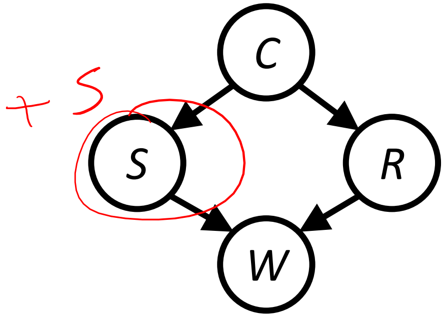
- What is P(W)?
  - We have counts <+w:4, -w:1>
  - Normalize to get P(W) = <+w:0.8, -w:0.2>
  - This will get closer to the true distribution with more samples
  - Can estimate anything else, too
  - Practice: What about P(C| +w)? P(C| +r, +w)? P(C| -r, -w)?
  - Fast: can use fewer samples if less time (what’s the drawback?)

Rejection Sampling



Rejection Sampling – early quitting

- 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 (i.e., correct in the limit)



+c, -s, ~~+r, +w~~ stop

+c, +s, +r, +w

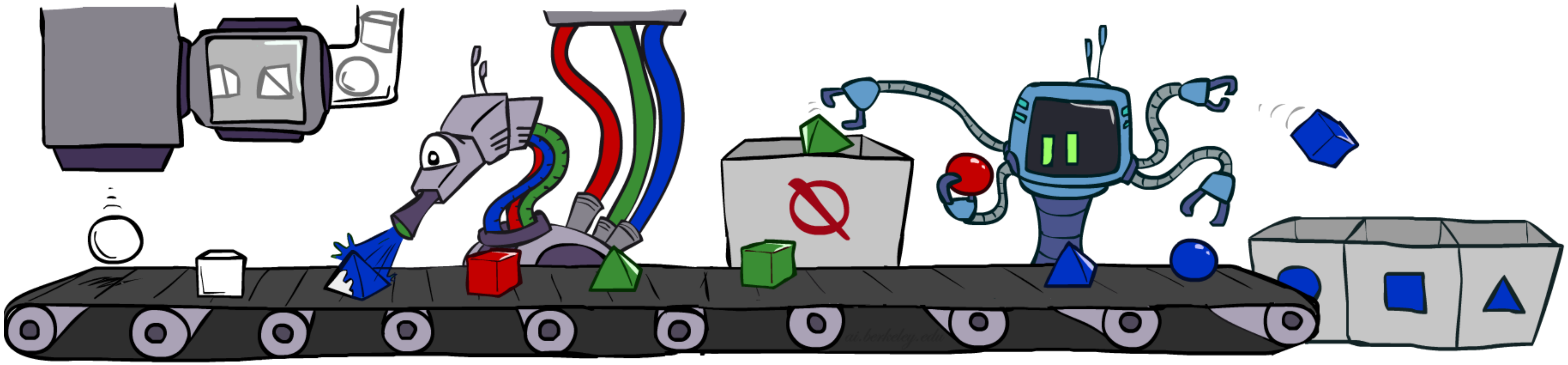
-c, +s, +r, -w

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

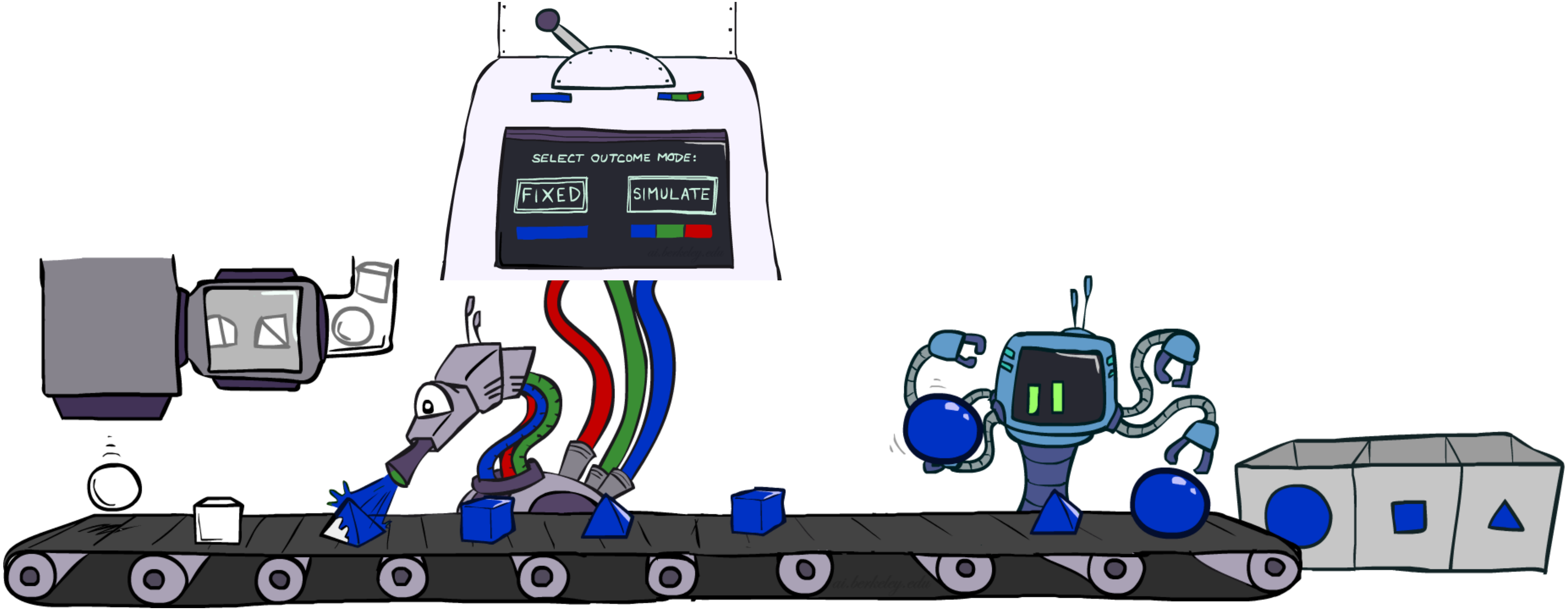
-c, -s, ~~-r, +w~~

Rejection Sampling

```
▪ IN: 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, and no sample is generated in this cycle
▪ Return $(x_1, x_2, \ldots, x_n)$
```

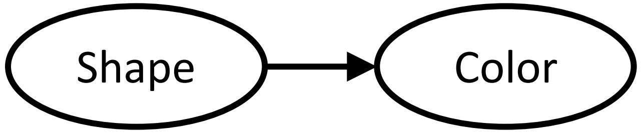


Likelihood Weighting

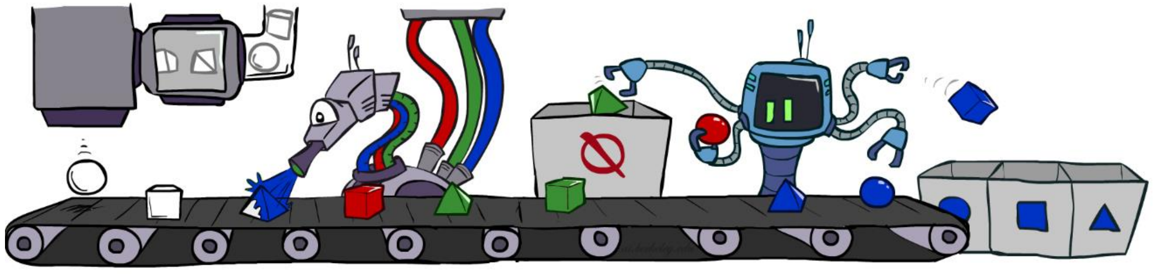


Likelihood Weighting

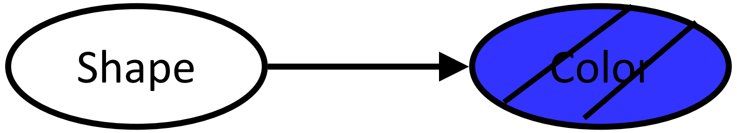
- Problem with rejection sampling:
  - If evidence is unlikely, rejects lots of samples
  - Evidence not exploited as you sample
  - Consider P(Shape|blue)



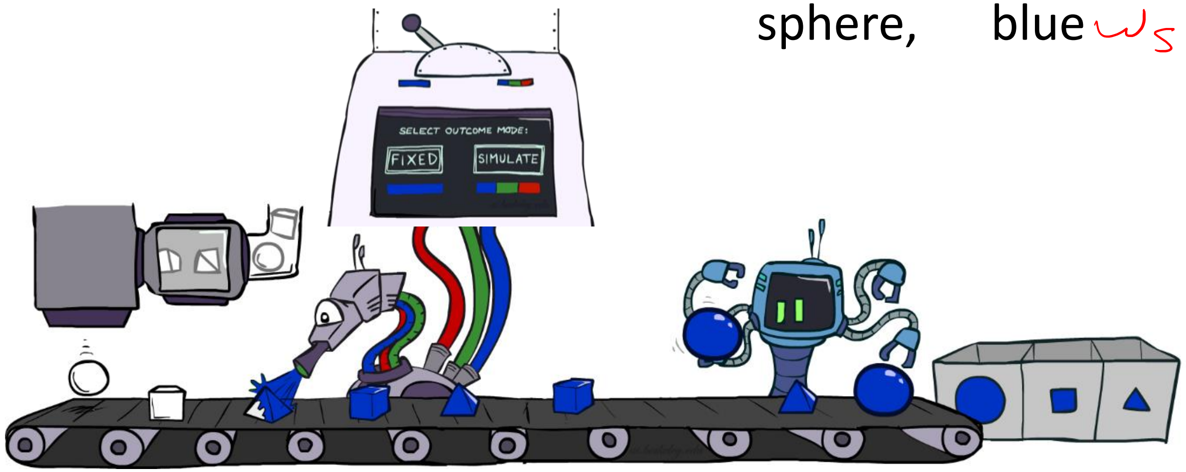
~~pyramid, green~~
~~pyramid, red~~
sphere, blue
~~cube, red~~
~~sphere, green~~



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



pyramid, blue w_1
pyramid, blue w_2
sphere, blue w_3
cube, blue w_4
sphere, blue w_5



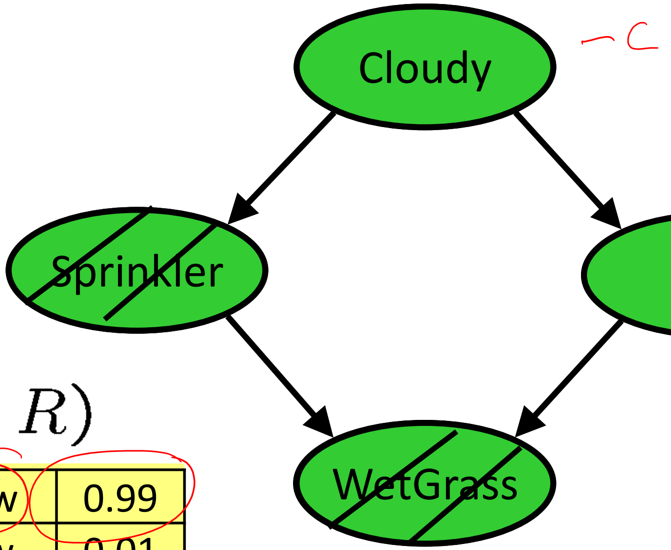
Likelihood Weighting

$P(C)$

+c	0.5
-c	0.5

0.27, 0.672

0.721, 0.01



$P(S|C)$

+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

+r

$P(W|S,R)$

+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

… -c, +s, -r, +w

P(evidence | parents)

$w_1 = 1.0 \times 0.1 \times 0.99$

$w_2 = 1.0 \times 0.5 \times 0.9$

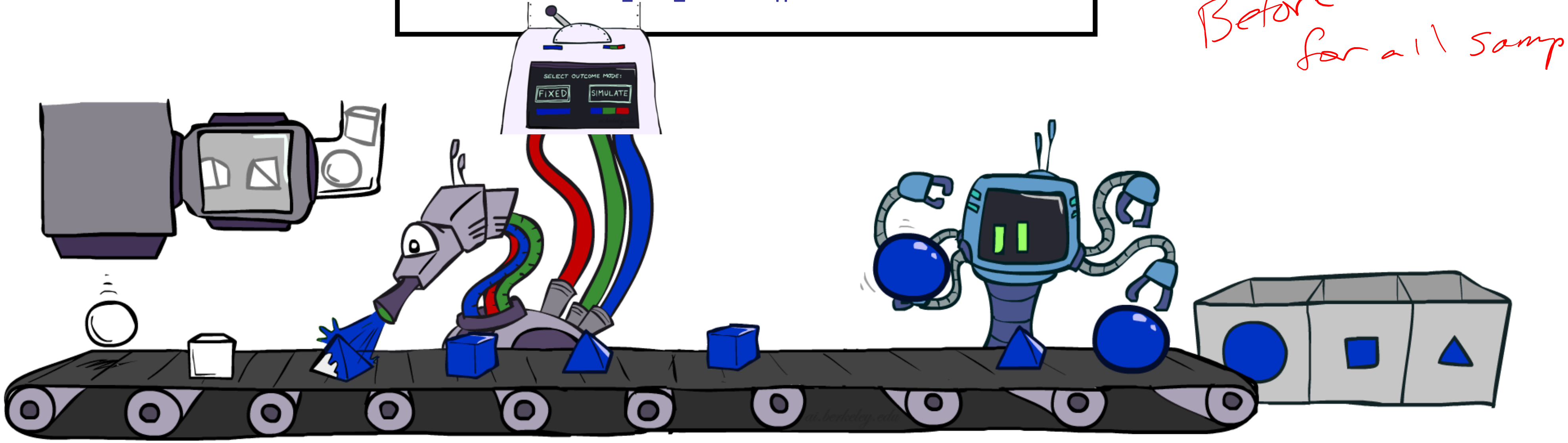
Likelihood Weighting

```
IN: 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 ($x_1$, $x_2$, …, $x_n$), w
```

Now each sample doesn't count as 1.0 but has a weight. Need to take a weighted average.

P(Q|Evidence) = Sum(weights of samples consistent with Query) / Total Weight of All samples.

Before assume w = 1 for all samples



Likelihood Weighting

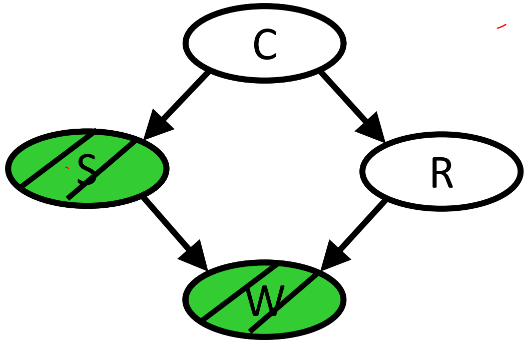
$P(X_1 \ldots X_n) = \prod_i P(x_i | \text{parents})$

- 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

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

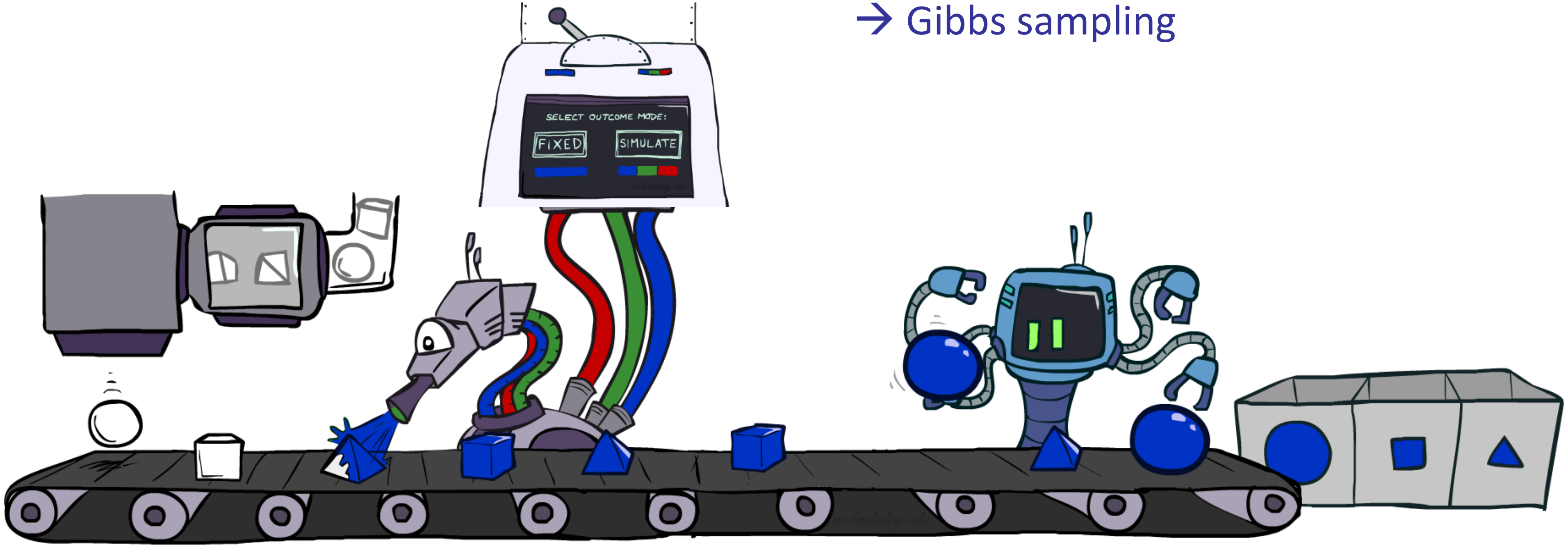
$$= P(\mathbf{z}, \mathbf{e})$$

Likelihood Weighting

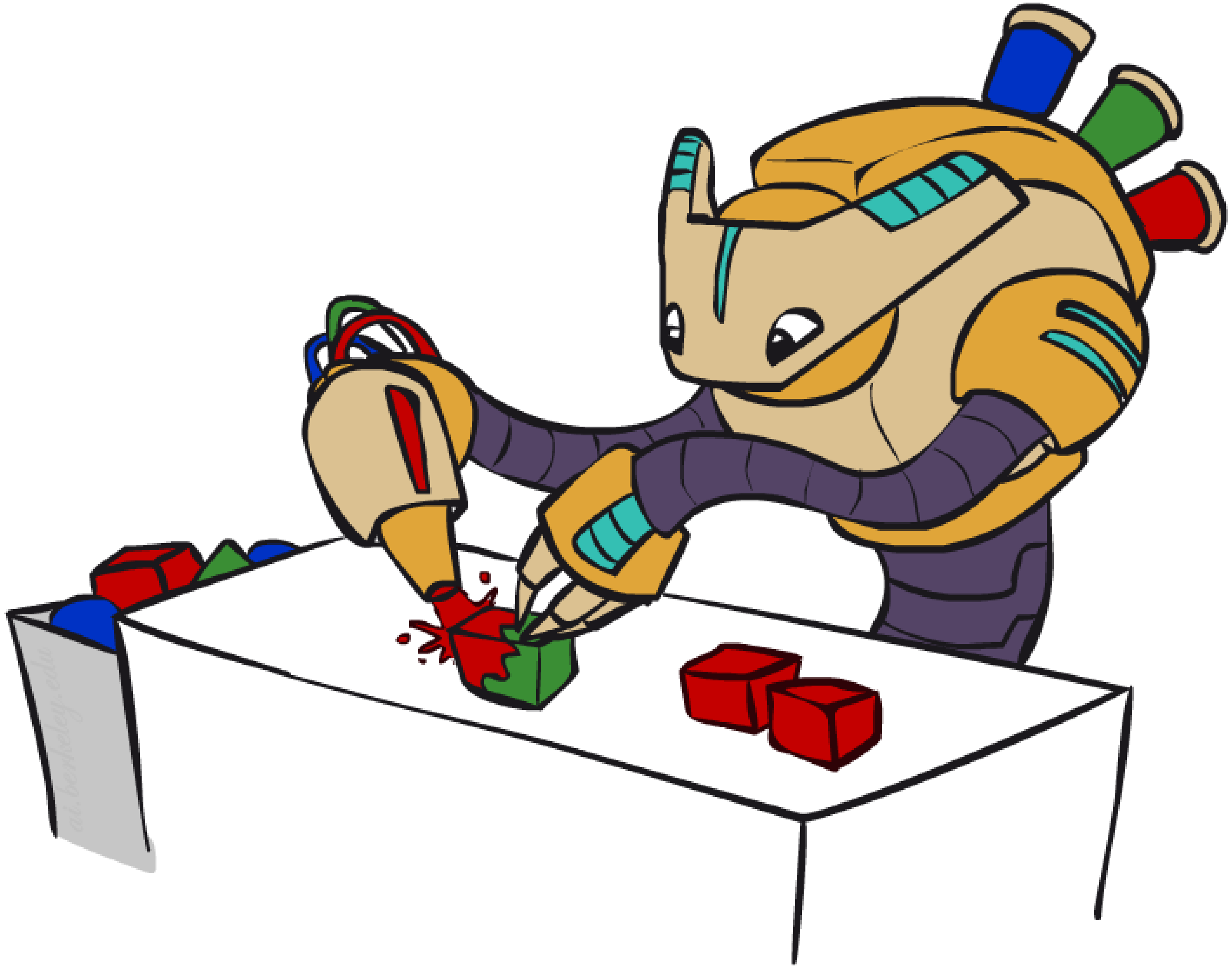
- Likelihood weighting is good
  - We have taken evidence into account as we generate the sample
  - E.g. here, W’ s value will get picked based on the evidence values of S, R
  - More of our samples will reflect the state of the world suggested by the evidence

- Likelihood weighting doesn’t solve all our problems (why?)
  - Evidence influences the choice of downstream variables, but not upstream ones (C isn’t more likely to get a value matching the evidence)
- We would like to consider evidence when we sample every variable

  → Gibbs sampling



Gibbs Sampling



Gibbs Sampling

- *Procedure:* keep track of a full instantiation x_1, x_2, …, 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 (infinite in theory).

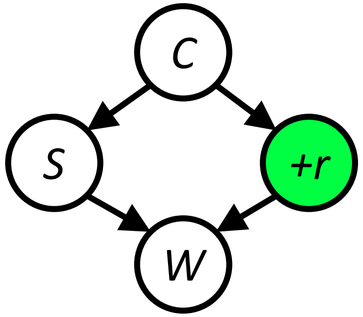
- *Property:* in the limit of repeating this infinitely many times the resulting sample is coming from the correct distribution. No need to weight!

- *Rationale*: both upstream and downstream variables condition on evidence.

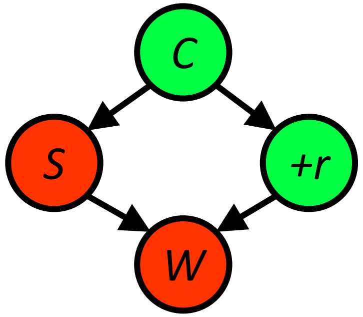
- In contrast: likelihood weighting only conditions on upstream evidence, and hence weights obtained in likelihood weighting can sometimes be very small. Sum of weights over all samples is indicative of how many "effective" samples were obtained, so want high weight.

Gibbs Sampling Example: P(S | +r)

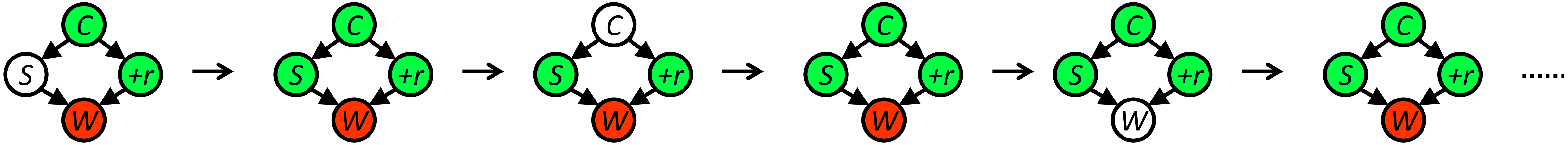
- Step 1: Fix evidence
  - R = +r



- Step 2: Initialize other variables
  - Randomly



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



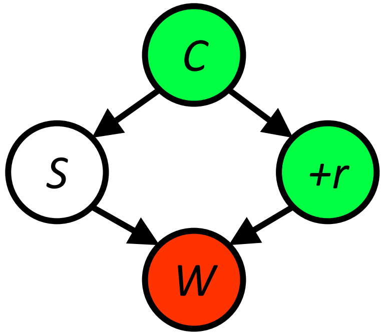
Sample from $P(S|+c,-w,+r)$ Sample from $P(C|+s,-w,+r)$ Sample from $P(W|+s,+c,+r)$

Gibbs Sampling

- 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)

Efficient Resampling of One Variable

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



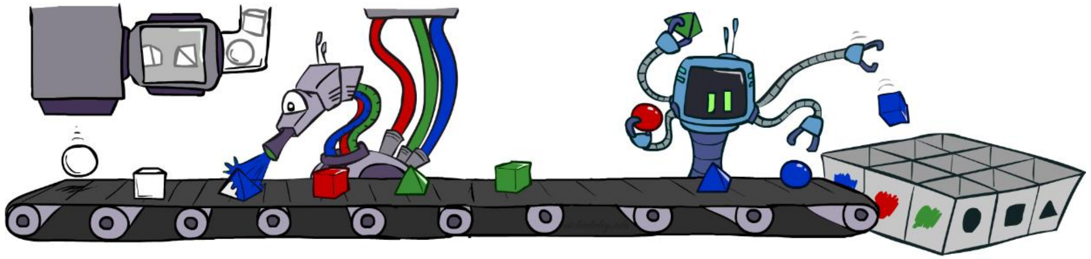
$$
\begin{aligned}
P(S|+c,+r,-w) &= \frac{P(S,+c,+r,-w)}{P(+c,+r,-w)} \\
&= \frac{P(S,+c,+r,-w)}{\sum_s P(s,+c,+r,-w)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{\sum_s P(+c)P(s|+c)P(+r|+c)P(-w|s,+r)} \\
&= \frac{\cancel{P(+c)}P(S|+c)\cancel{P(+r|+c)}P(-w|S,+r)}{\cancel{P(+c)}\cancel{P(+r|+c)}\sum_s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

Lots of things cancel

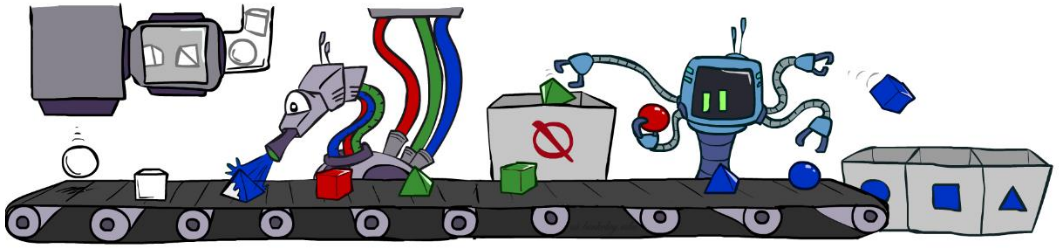
- Many things cancel out – only CPTs with S remain!
- More generally: only CPTs that have resampled variable need to be considered, and joined together

Bayes’ Net Sampling Summary

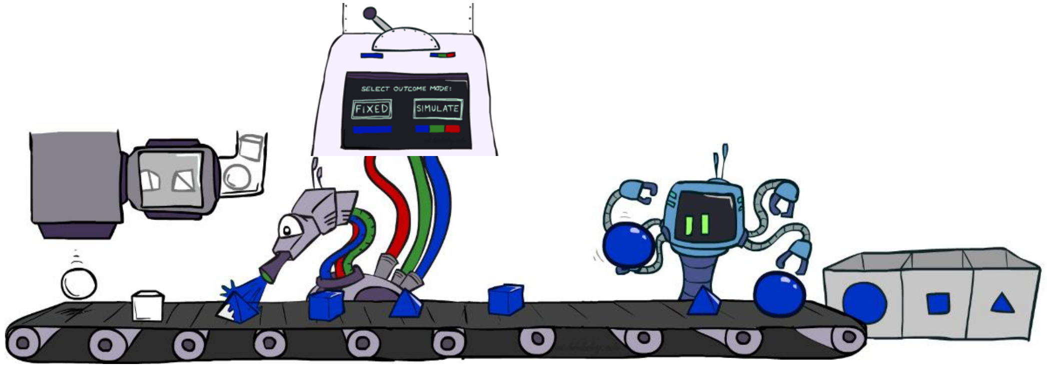
- Prior Sampling P



- Rejection Sampling P(Q | e)



- Likelihood Weighting P(Q | e)



- Gibbs Sampling P(Q | e)

Further Notes 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