COMS W4701: Artificial Intelligence

Lecture 21: Approximate Inference in Bayes Nets

Tony Dear, Ph.D.

Department of Computer Science

School of Engineering and Applied Sciences

Today

- Forward sampling
- Likelihood weighting and importance sampling
- MCMC and Gibbs sampling

Approximate Inference: Sampling

- Exact inference to find complex posterior distributions is NP-hard
- But we do have the “components” of the posterior in the CPTs

- Idea: **Monte Carlo** methods use the Bayes net parameters (CPTs) to generate *samples* and approximate query probabilities and distributions
- Can obtain approximations very quickly if exact values are not needed

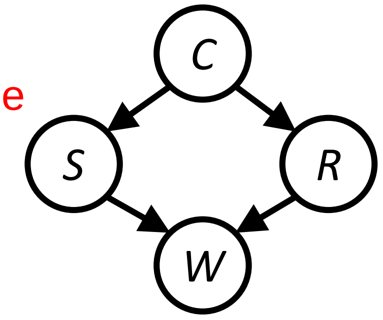
- The more samples we get, the closer we can estimate true probabilities
- Runtimes scale with number of samples rather than distribution size

Forward Sampling

- A full sample consists of a value assignment to every variable using known CPTs

- Procedure: Sample a value for each variable one at a time following topological order

- For each node, use the conditional distribution corresponding to the parents’ sampled values



1. Sample from $P(C)$. Suppose we get $+c$.

2. Sample from $P(S|+c)$. Suppose we get $+s$.

3. Sample from $P(R|+c)$. Suppose we get $-r$.

4. Sample from $P(W|+s,-r)$. Suppose we get $-w$.

- To sample from distribution with probabilities $\theta_1, \dots, \theta_k$, we can uniformly sample between 0 and 1 and pick value j if sample is in $[\sum_{i=1}^{j-1} w_i, \sum_{i=1}^{j} w_i]$

Example: Forward Sampling

- Suppose we get 5 samples:
- (+c, -s, +r, +w)
- (+c, +s, +r, +w)
- (-c, +s, +r, -w)
- (+c, -s, +r, +w)
- (-c, -s, -r, +w)

$\hat{P}(R)$

+r	0.8
-r	0.2

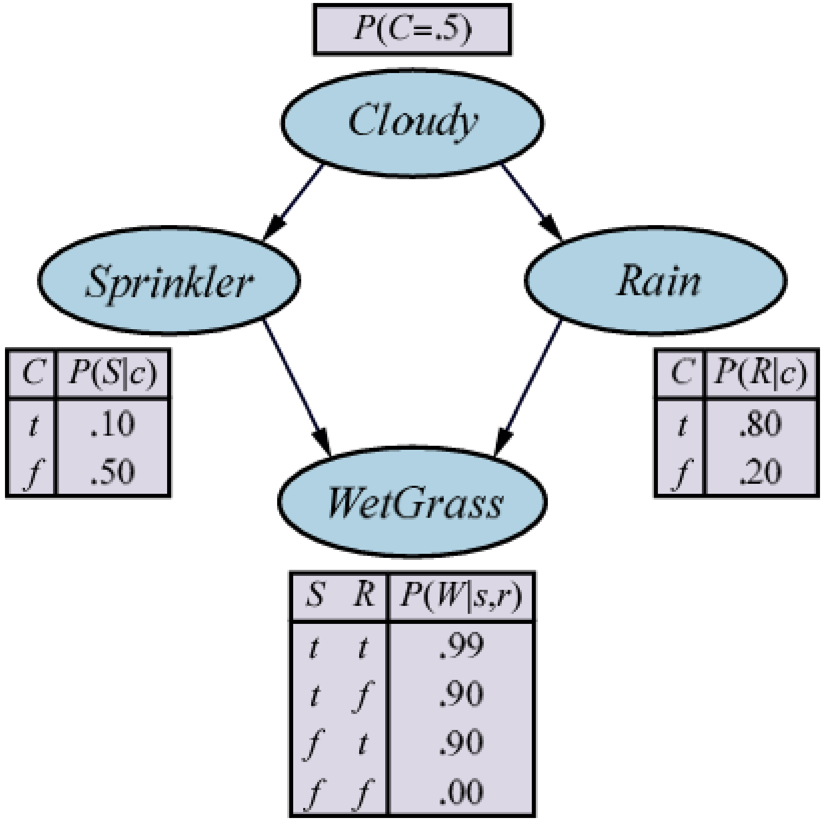
$\hat{P}(C,W)$

+c	+w	0.6
	-w	0
-c	+w	0.2
	-w	0.2

$\hat{P}(S|W)$

+w	+s	0.25
	-s	0.75
-w	+s	1
	-s	0

- We can now estimate any distributions or probability tables we want!



C	P(S|c)
t	.10
f	.50

C	P(R|c)
t	.80
f	.20

S	R	P(W|s,r)
t	t	.99
t	f	.90
f	t	.90
f	f	.00

Forward Sampling Considerations

- Applications: Rollouts in MCTS, text generation in large language models
- Probability of any sample $(x_1, \dots, x_n)$ is just $\prod_{i=1}^{n} P(x_i | parents(X_i))$

- Samples are **consistent** with the *prior* Bayes net joint distribution
- Proportion of samples approximates probability of corresponding event

- Complexity of generating a single sample is linear in Bayes net size
- Number of required samples to achieve a given error bound grows inversely with event probability (rare event -> more samples needed)

Rejection Sampling

- If we are interested in a query with *evidence* $P(X|e)$, we only need to keep samples that match all e values and *reject* the rest
- Ex: Want $P(S|+w)$, can reject all samples with $W = -w$

- Expected fraction of samples that are kept is equal to $P(e)$
- Can be very small if evidence is rare, leading to few samples kept

- Evidence probability also decreases exponentially with *number* of variables, e.g. $P(e_1, e_2, e_3)$ much less likely than $P(e_1)$

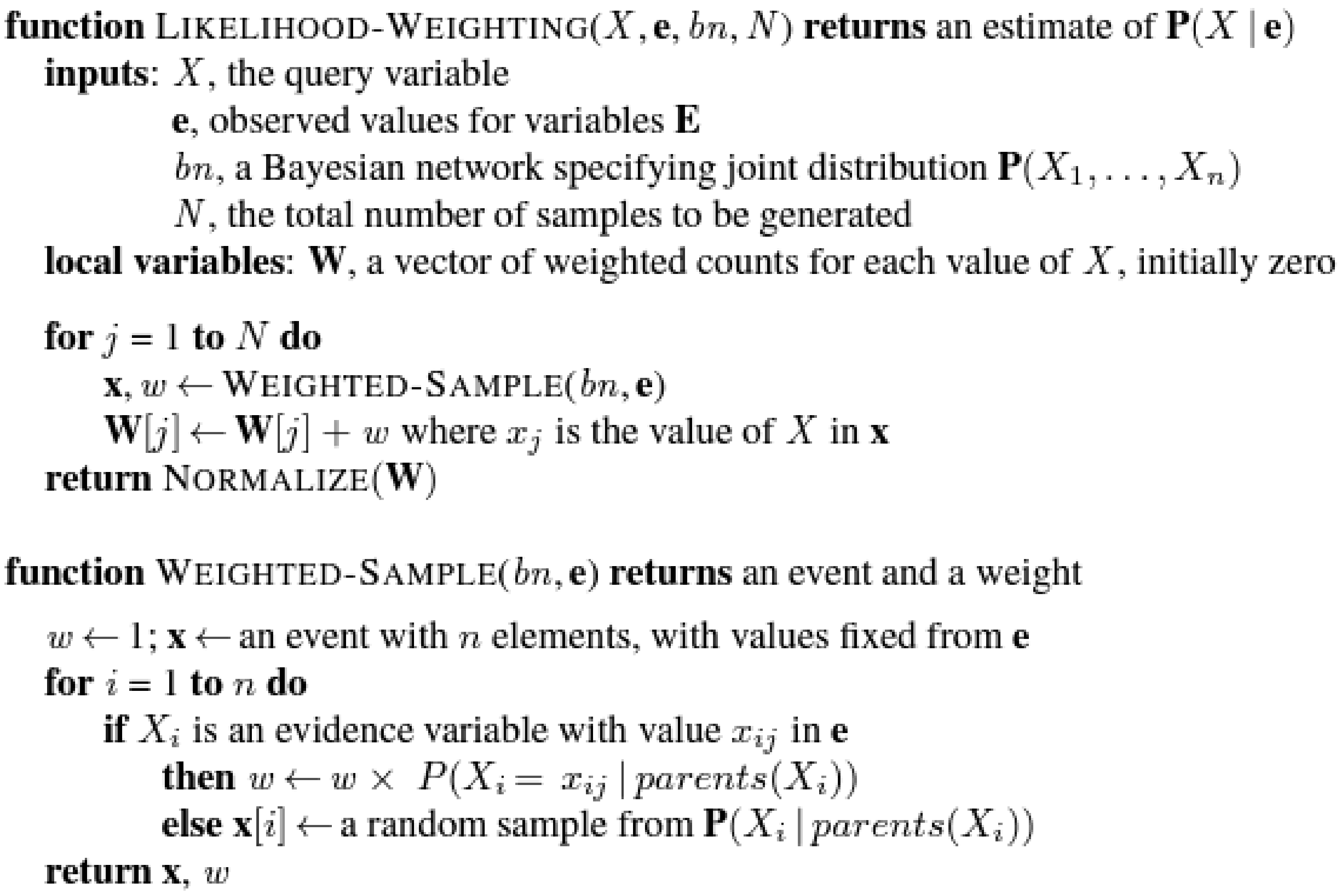
Likelihood Weighting

- Idea: **Fix** evidence variables to take on the given values
- We can keep *every* sample that we generate—no rejection

- But now samples may look very different from the prior distribution
- Idea: **Weight** each sample using likelihood of evidence to compensate

- The “count” contribution from a weighted sample is just its weight value
- We take weighted averages of sample values to estimate posterior probs

Likelihood Weighting



```
function LIKELIHOOD-WEIGHTING($X$, $\mathbf{e}$, $bn$, $N$) returns an estimate of $\mathbf{P}(X \mid \mathbf{e})$
  inputs: $X$, the query variable
          $\mathbf{e}$, observed values for variables $\mathbf{E}$
          $bn$, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$
          $N$, the total number of samples to be generated
  local variables: $\mathbf{W}$, a vector of weighted counts for each value of $X$, initially zero

  for $j$ = 1 to $N$ do
      $\mathbf{x}$, $w \leftarrow$ WEIGHTED-SAMPLE($bn$, $\mathbf{e}$)
      $\mathbf{W}[j] \leftarrow \mathbf{W}[j] + w$ where $x_j$ is the value of $X$ in $\mathbf{x}$
  return NORMALIZE($\mathbf{W}$)

function WEIGHTED-SAMPLE($bn$, $\mathbf{e}$) returns an event and a weight

  $w \leftarrow 1$; $\mathbf{x} \leftarrow$ an event with $n$ elements, with values fixed from $\mathbf{e}$
  for $i$ = 1 to $n$ do
      if $X_i$ is an evidence variable with value $x_{ij}$ in $\mathbf{e}$
          then $w \leftarrow w \times P(X_i = x_{ij} \mid parents(X_i))$
          else $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return $\mathbf{x}$, $w$
```

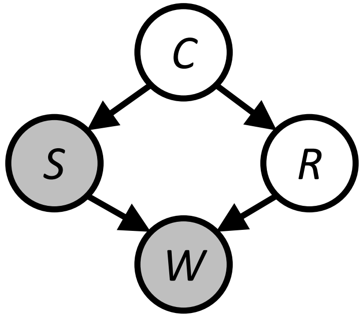
Example: Likelihood Weighting

- Suppose we want $P(C, R \mid +s, +w)$

- Fix $+s$ and $+w$; sample other variables
- Sample weights: $P(+s|c)P(+w|+s, r)$
  - (+c, +s, +r, +w) $0.1 \times .99 = .099$
  - (+c, +s, +r, +w) $0.1 \times .99 = .099$
  - (+c, +s, -r, +w) $0.1 \times 0.9 = 0.09$
  - (-c, +s, -r, +w) $0.5 \times 0.9 = 0.45$

- Probabilities are normalized weighted averages of sample values

C	P(+s|C)
+c	0.1
-c	0.5



R	P(+w|+s,R)
+r	0.99
-r	0.90

$\hat{P}(C, R \mid +s, +w)$

+c	+r	0.198
	-r	0.09
-c	+r	0
	-r	0.45

$\propto$

+c	+r	0.268
	-r	0.122
-c	+r	0
	-r	0.610

Importance Sampling

- Another perspective: We want to find the *posterior distribution* $P(X|e)$
- But samples are generated from the *prior* $\prod_i P(X_i|parents(X_i))$

- **Importance sampling**: Apply a *weight* to each sample $\boldsymbol{x}$ to correct for this mismatch:

Posterior probability

$$w(\boldsymbol{x}) = \frac{P(\boldsymbol{x}|e)}{\prod_i P(x_i|parents(X_i))} \propto \prod_j P(e_j|parents(E_j))$$

Prior probability

- In practice, this works regardless of the sampling (prior) distribution!

Likelihood Weighting Considerations

- Think of likelihood weighting as a “patch”
- The more different the prior and posterior, the more variance in the weight values

- If most evidence is *upstream* near roots, then the prior will be more similar to the posterior, and weights will provide smaller or no corrections

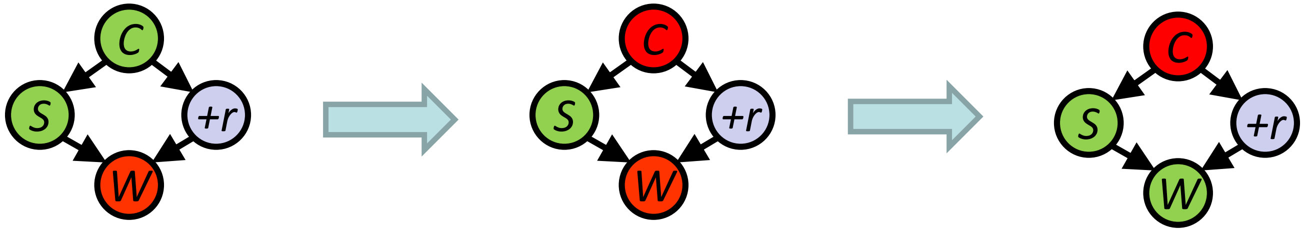
- If most evidence is *downstream* near leaves, then many samples may be irrelevant to query, with greater variety in the weights
- Estimates will be dominated by few samples with non-infinitesimal weights

Sequences of Samples

- We want evidence to affect sampling both upstream and downstream
- If most evidence at leaf nodes, prior looks very different from posterior

- New idea: Construct a *sequence* of samples s.t. successive samples come from distributions that look more and more like the posterior

- Example: Suppose $+r$ is fixed evidence
- Initial sample: $(+c, -s, +r, -w)$



Sample from $P(S \mid +c, +r, -w)$ and obtain $+s$

Sample from $P(C \mid +s, +r)$ and obtain $-c$

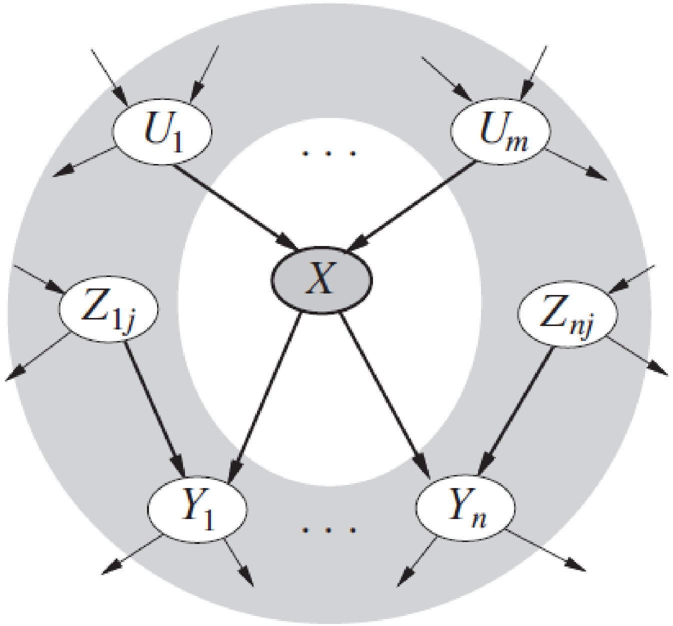
Sample from $P(W \mid +s, +r)$ and obtain $+w$

Markov Blanket

- Problem: We need to compute $P(X_i \mid all\ other\ nodes\ in\ the\ BN)$ in order to perform sampling, which seems intractable
- We can reduce these dependencies to just the *Markov blanket* of X_i

- The **Markov blanket** of X includes its parents U_i, children Y_j, children's parents Z_{kj}

- X is conditionally independent of *all other nodes* given observations of all nodes in its Markov blanket



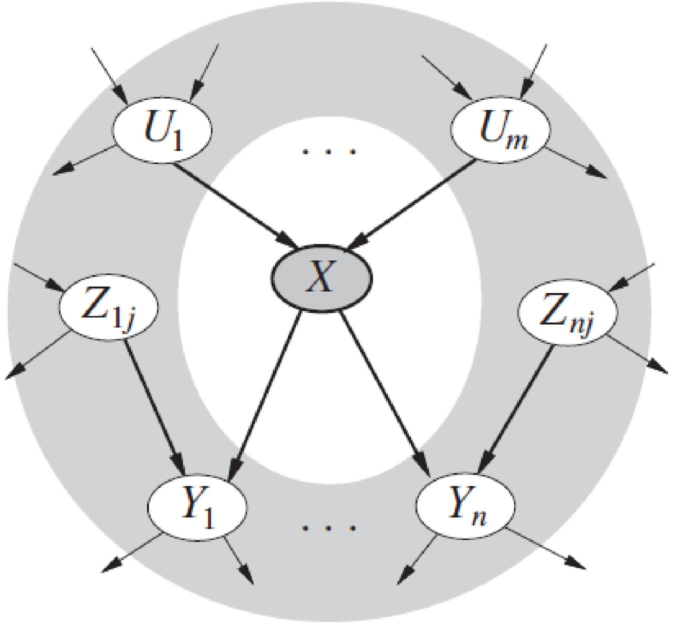
Markov Blanket

- To sample from $P(X_i|MB(X_i))$, compute $P\big(X_i, MB(X_i)\big)$ and normalize
- If X_i has n children, this is a product of $n+1$ tables, each of size $|X_i|$

$$P(X_i|mb(X_i)) \propto P(X_i|parents(X_i)) \times \prod_{Y_j} P\big(y_j|parents(Y_j)\big)$$

- We can easily compute these distributions given any combination of sample values

- We can use these distributions to *resample* any non-evidence (non-fixed) variable

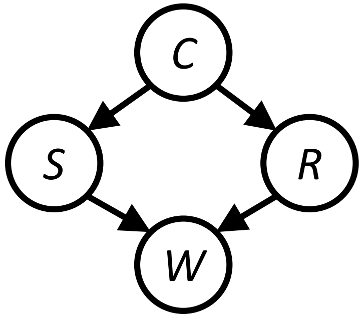


Example: Markov Blanket

C	P(+s|C)
+c	0.1
-c	0.5

C	P(C)
+c	0.5

C	P(+r|C)
+c	0.8
-c	0.2



$$P(C \mid mb(C)) = P(C \mid s, r) \propto P(C)P(s|C)P(r|C)$$

$$P(S \mid mb(S)) = P(S \mid c, r, w) \propto P(S|c)P(w|S, r)$$

$$P(R \mid mb(R)) = P(R|c, s, w) \propto P(R|c)P(w|s, R)$$

$$P(W \mid mb(W)) = P(W \mid s, r)$$

C	P(C,+s,+r)
+c	0.04
-c	0.05

=

C	P(C)
+c	0.5
-c	0.5

×

C	P(+s|C)
+c	0.1
-c	0.5

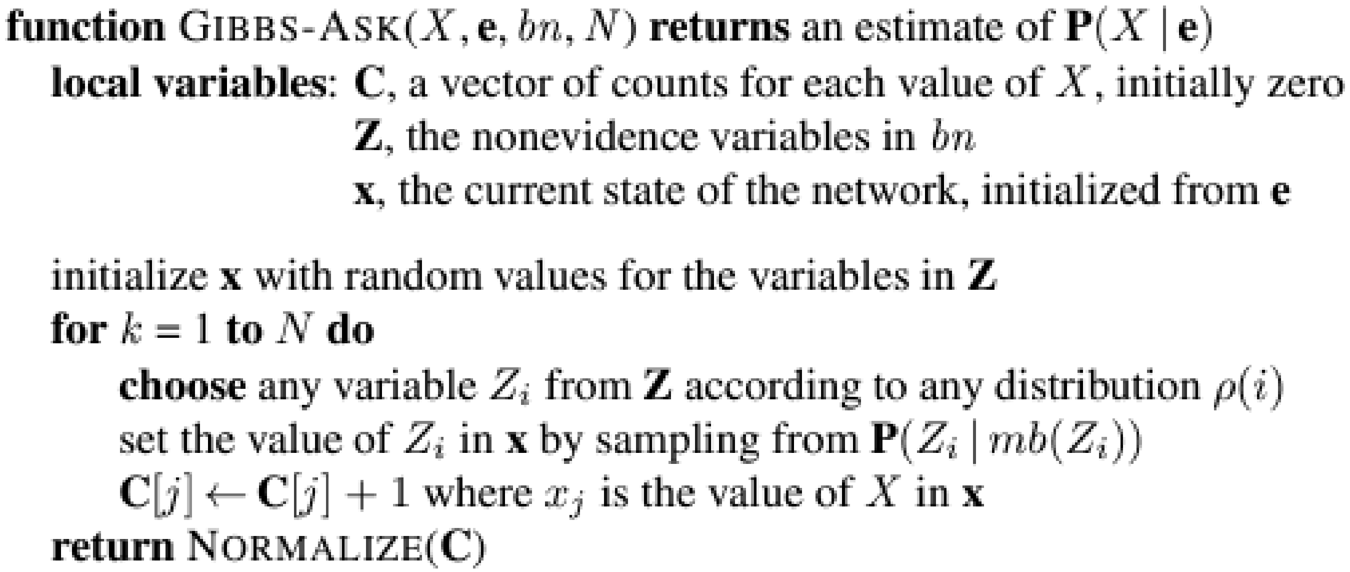
×

C	P(+r|C)
+c	0.8
-c	0.2

$$P(C|mb(C)) = (\frac{4}{9}, \frac{5}{9})$$

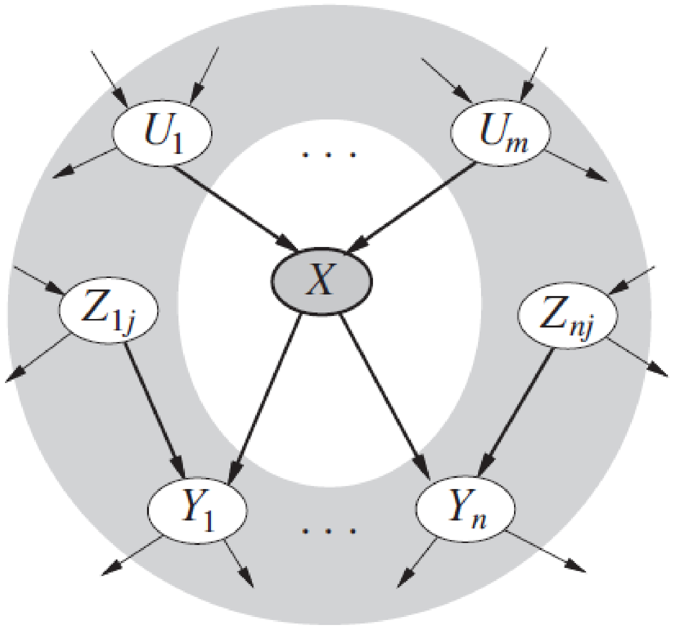
Gibbs Sampling

- **Gibbs sampling**: Generate a *sequence* of samples, where *one* non-evidence variable Z_i of the ith sample is resampled conditioned on the $(i-1)$th sample
- The resampled variable Z_i can be chosen randomly or deterministically
- Over time, the samples start to reflect the posterior that we want to estimate!



```
function GIBBS-ASK($X$, $\mathbf{e}$, $bn$, $N$) returns an estimate of $\mathbf{P}(X \mid \mathbf{e})$
  local variables: $\mathbf{C}$, a vector of counts for each value of $X$, initially zero
                   $\mathbf{Z}$, the nonevidence variables in $bn$
                   $\mathbf{x}$, the current state of the network, initialized from $\mathbf{e}$

  initialize $\mathbf{x}$ with random values for the variables in $\mathbf{Z}$
  for $k$ = 1 to $N$ do
     choose any variable $Z_i$ from $\mathbf{Z}$ according to any distribution $\rho(i)$
     set the value of $Z_i$ in $\mathbf{x}$ by sampling from $\mathbf{P}(Z_i \mid mb(Z_i))$
     $\mathbf{C}[j] \leftarrow \mathbf{C}[j] + 1$ where $x_j$ is the value of $X$ in $\mathbf{x}$
  return NORMALIZE($\mathbf{C}$)
```

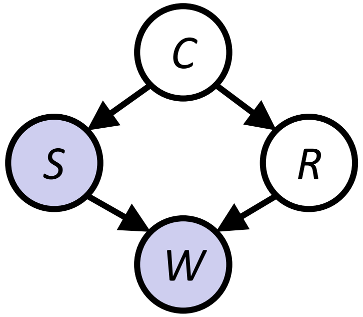


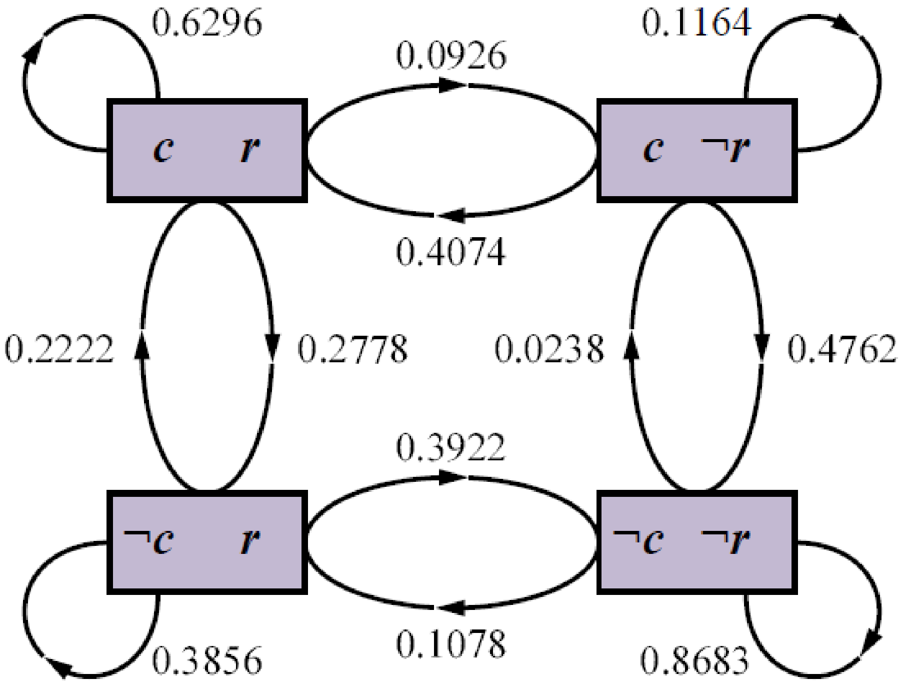
Markov Chain Monte Carlo

- Think of posterior as a *belief state*
- Jumping from sample to sample simulates a Markov chain over posterior

- Transition probabilities are *likelihoods* of obtaining new sample given current

- Can show that stationary distribution is exactly equal to posterior distribution





Gibbs Sampling Considerations

- Initial set of samples may be far from the posterior estimate
- Solution: Implement a *burn-in* period and discard the first n samples

- There are more sophisticated MCMC methods (e.g., Metropolis-Hastings)
- Different methods construct the underlying Markov chain in different ways

- MCMC are the most common methods for inference in large networks and other computational problems, e.g. solving multi-dimensional integrals
- Once compiled, can run very fast or asynchronously in parallel

Summary

- Performing inference is computationally heavy in large Bayes nets with many query and hidden variables

- Monte Carlo sampling allows us to *estimate* probability distributions
- Direct sampling methods draw samples independently
- Can also weight samples to be consistent with evidence

- MCMC methods (e.g., Gibbs sampling) treat sampling as local search
- Transitions follow a Markov chain; stationary distribution is the posterior