COMP9418: Advanced Topics in Statistical Machine Learning

Approximate Inference by Sampling

Instructor: Gustavo Batista

University of New South Wales

Sampling

- *Stochastic sampling* is a method for estimating probabilities
  - It works by measuring the frequency of events according to a simulation
  - We can efficiently simulate situations according to their probability of occurrence by operating on the corresponding network

- Basic idea
  - Draw n samples from a sampling distribution P'
  - Compute an approximate probability
  - Show this converges to the true probability P

- Why sample?
  - Inference: getting a sample is faster than computing the right answer

Monte Carlo Simulation

- We first simulate a random sample $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n$ from the underlying distribution $P(\boldsymbol{X})$
  - Evaluate a function at each instantiation of the sample, $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n$
  - Compute the arithmetic average known as the *sample mean*
  - We can also compute the *sample variance*

$$Av_n(f) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}^i)$$

$$S_n^2(f) \stackrel{\text{def}}{=} \frac{1}{n-1}\sum_{i=1}^{n}\left(f(\boldsymbol{x}^i) - Av_n(f)\right)^2$$

Sampling

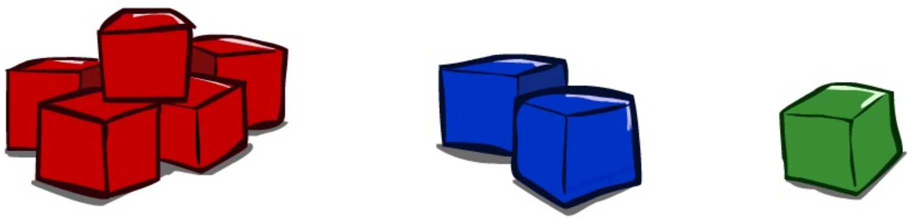
- Sampling from given distribution
  - Step 1: Get sample u from uniform distribution over [0, 1)
    - E.g. random() in python
  - Step 2: Convert this sample u into an outcome for the given distribution by having each outcome associated with a sub-interval of [0,1) with sub-interval size equal to probability 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
- E.g, after sampling 8 times:



Sampling

- Forward Sampling

- Rejection Sampling

- Likelihood Weighting

- Gibbs Sampling

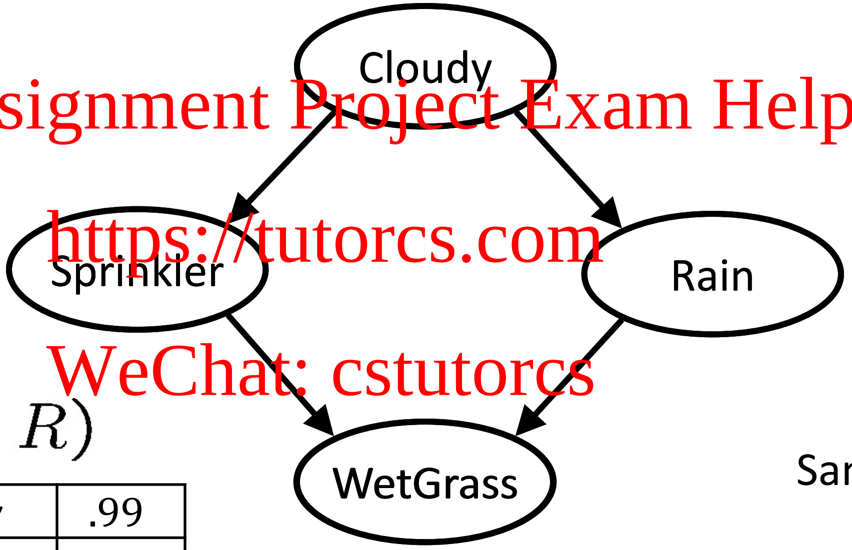
Forward Sampling

$P(C)$

c	.5
$\bar{c}$	.5

$P(S|C)$

c	s	.1
	$\bar{s}$	.9
$\bar{c}$	s	.5
	$\bar{s}$	.5



$P(R|C)$

c	r	.8
	$\bar{r}$	.2
$\bar{c}$	r	.2
	$\bar{r}$	.8

$P(W|S,R)$

s	r	w	.99
		$\bar{w}$	.01
	$\bar{r}$	w	.90
		$\bar{w}$	.10
$\bar{s}$	r	w	.90
		$\bar{w}$	.10
	$\bar{r}$	w	.01
		$\bar{w}$	.99

Samples:

Simulate Bayesian Network: Simulate_BN

Input: Network N with variables $\boldsymbol{X}$ inducing a distribution P

Output: one instance Σ

$\pi \leftarrow$ topological order of the nodes in N

$\Sigma \leftarrow \top$ # Trivial instantiation

for $i = 1$ **to** n **do** # Network has n variables

  $X \leftarrow$ variable at position i in order π

  $\boldsymbol{u} \leftarrow$ value of X's parents in instantiation Σ

  $x \leftarrow$ value of X sampled according to $P(X|\boldsymbol{u})$

  $\Sigma \leftarrow \Sigma, x$ # Append value x to Σ

return Σ

Simulate Bayesian Network

Input: event α, sample size n, Network N

Output: sample mean for f_α

$p \leftarrow 0$

for $i = 1$ **to** n **do**

  $\boldsymbol{x}^i \leftarrow$ Simulate_BN(N) # Simulate the Bayesian network

  $p \leftarrow p + f_\alpha(\boldsymbol{x}^i)$

return p/n

Example

- Get several samples from the Bayesian network:

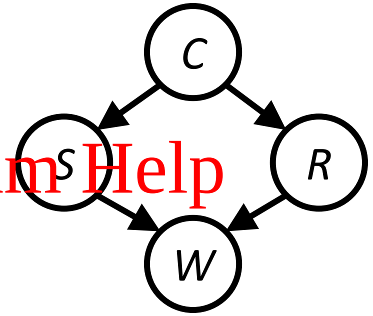
  $c, \bar{s}, r, w$

  c, s, r, w

  $\bar{c}, s, r, \bar{w}$

  $c, \bar{s}, r, w$

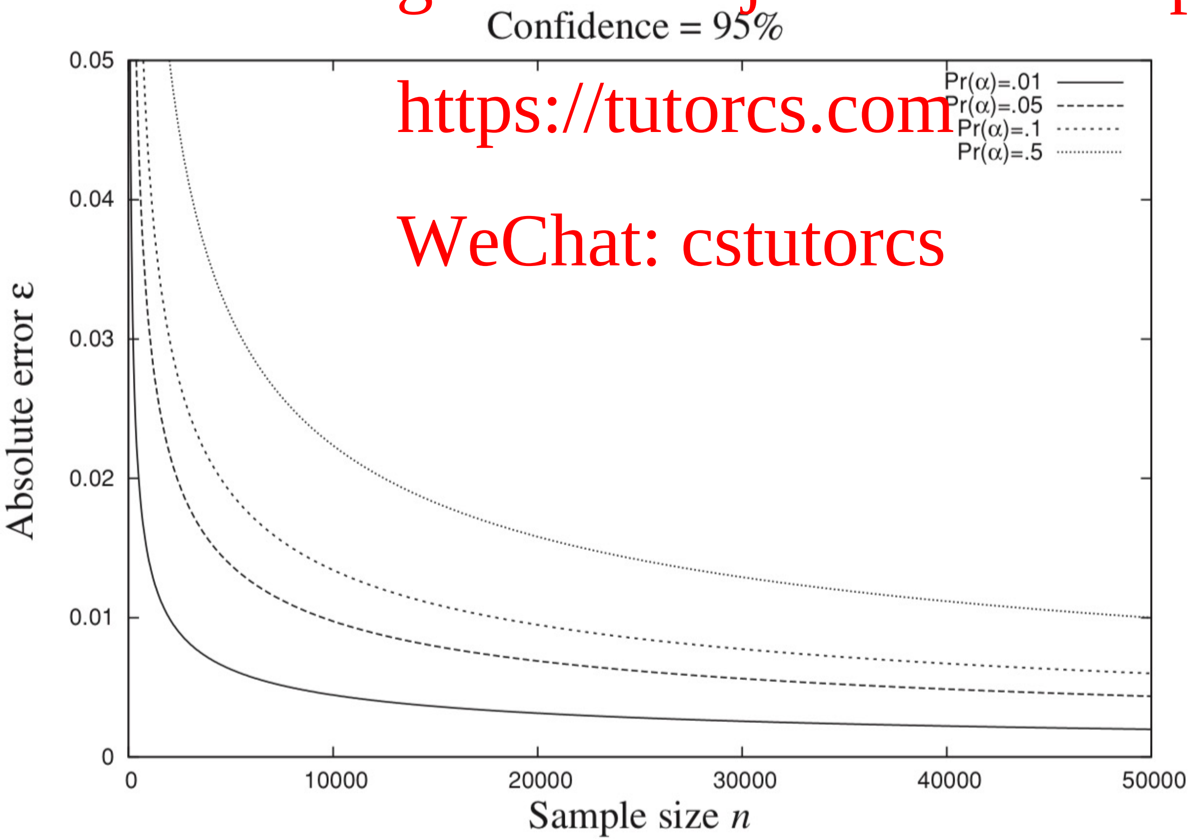
  $\bar{c}, \bar{s}, \bar{r}, w$



- If we want to know $P(W)$
  - We have counts $< w: 4, \bar{w}: 1 >$
  - Normalize to get $P(W) \ = < w: 0.8, \bar{w}: 0.2 >$
  - This will get closer to the true distribution with more samples
  - Can estimate anything else, too
  - What about $P(C|\bar{w})$? $P(C|\, r, w)$? $P(C|\bar{r}, \bar{w})$?
  - Fast: can use fewer samples if less time (what's the drawback?)

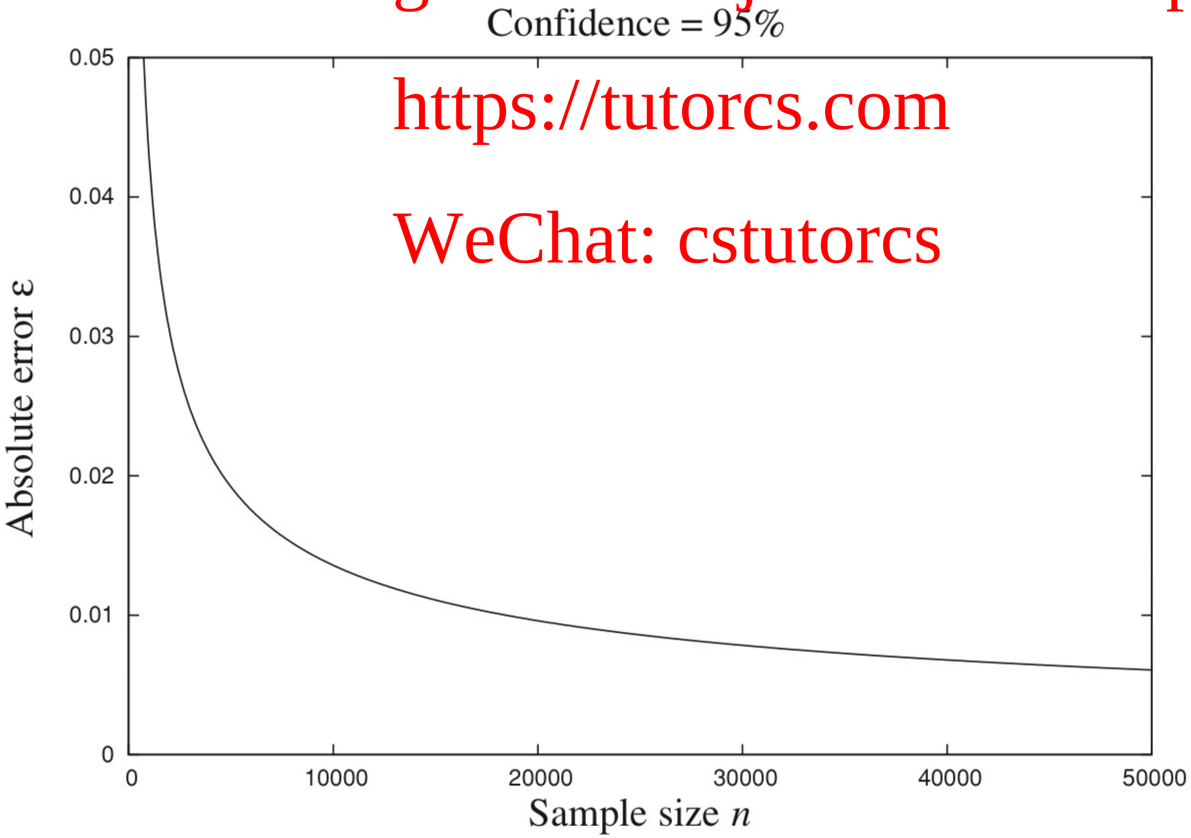
How Many Samples?

- Bounds for independent sampling, for an estimator
  - Chebyshev bound: $P(|Av_n(f_\alpha) - P(\alpha)| \leq \epsilon) \geq 1 - \frac{P(\alpha)P(\overline{\alpha})}{n\epsilon^2}$



How Many Samples?

- Bounds for independent sampling, for an estimator
  - Hoeffding bound: $P(|Av_n(f_\alpha) - P(\alpha)| \leq \epsilon) \geq 1 - 2e^{-2n\epsilon^2}$

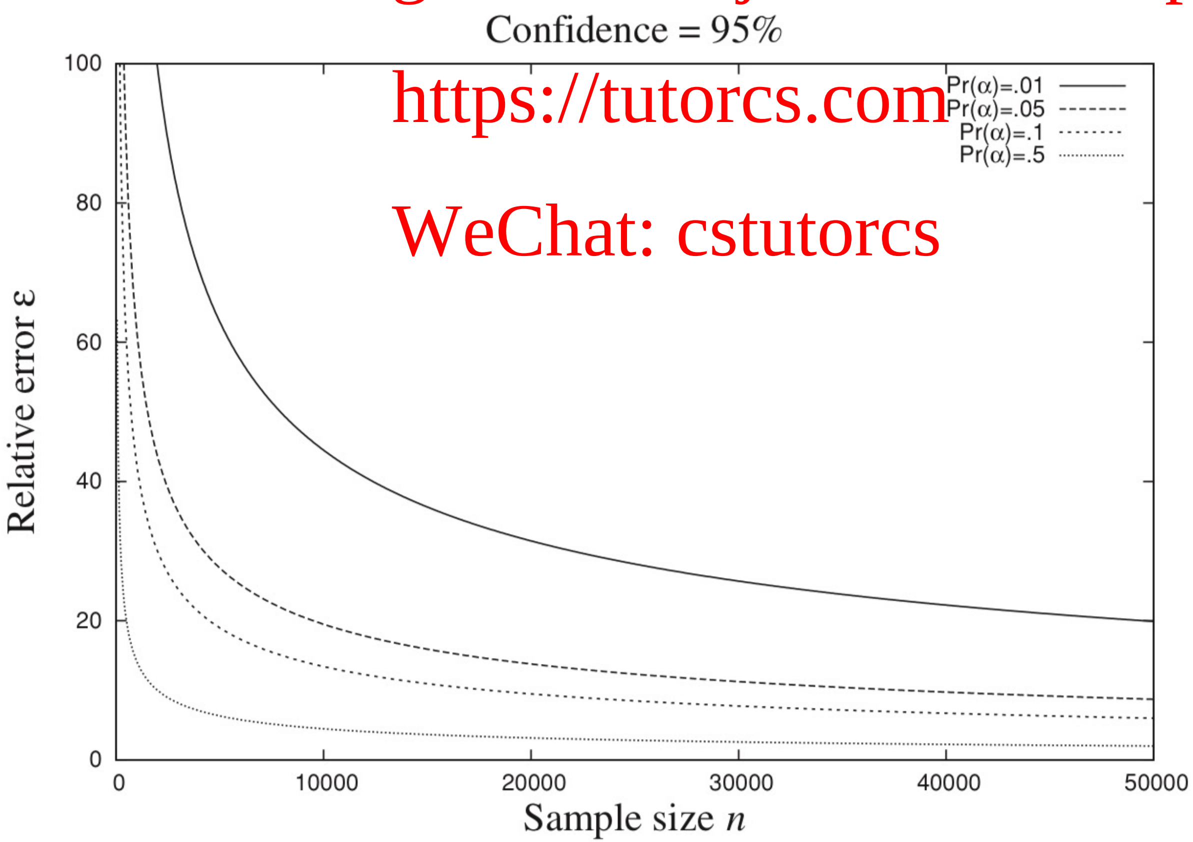


How Many Samples?

- We can define the relative error as

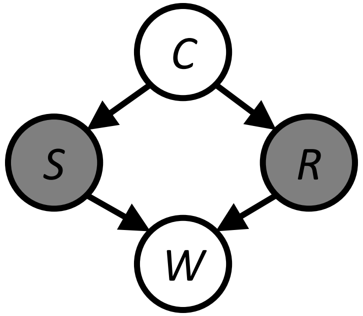
$$\frac{|Av_n(f_\alpha) - P(\alpha)|}{P(\alpha)}$$

  - Chebyshev bound: $P\left(\frac{|Av_n(f_\alpha)-P(\alpha)|}{P(\alpha)} \le \epsilon\right) \ge 1 - \frac{P(\bar{\alpha})}{n\epsilon^2 P(\alpha)}$



Estimating a Conditional Probability

- Consider now the problem of estimating a conditional probability $\mathrm{P}(\alpha|\beta)$
  - With a distribution $P(.)$ induced by a Bayesian network
  - However, sampling from the distribution $\mathrm{P}(.|\beta)$ is generally hard

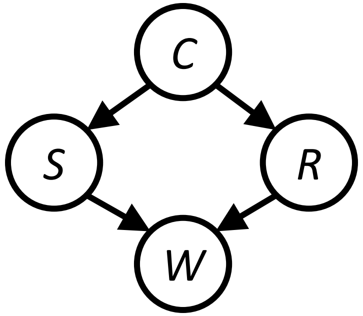


- We typically cannot efficiently generate a sequence of independent instantiations $\boldsymbol{x}^1, \dots, \boldsymbol{x}^n$
  - Where the probability of generating instantiation $\boldsymbol{x}^i$ is $P(\boldsymbol{x}^i|\beta)$

Rejection Sampling

- Suppose say we want $P(C|r)$
  - We can compute this using the Bayes conditioning
  - Count C outcomes, but ignore (reject) samples which do not have $R = r$
  - This is called rejection sampling
  - It is essentially, the same as forward sampling

$$P(\alpha|\beta) = \frac{P(\alpha, \beta)}{P(\beta)}$$



$c, \bar{s}, r, w$

c, s, r, w

$\bar{c}, s, r, \bar{w}$

$c, \bar{s}, r, w$

$\bar{c}, \bar{s}, \bar{r}, w$

Rejection Sampling: Simulate_BN

Input: Network N with variables $\boldsymbol{X}$ inducing a distribution P, evidence $\boldsymbol{e}$

Output: one instance Σ consistent with $\boldsymbol{e}$ or $\top$

$\pi \leftarrow$ topological order of the nodes in N

$\Sigma \leftarrow \top$ # Trivial instantiation

for $i = 1$ **to** n **do** # Network has n variables

    $X \leftarrow$ variable at position i in order π

    $\boldsymbol{u} \leftarrow$ value of X's parents in instantiation Σ

    $x \leftarrow$ value of X sampled according to $P(X|\boldsymbol{u})$

    if x is not consistent with $\boldsymbol{e}$

        return $\top$ # Reject, no sample is generated

    $\Sigma \leftarrow \Sigma, x$ # Append value x to Σ

return Σ

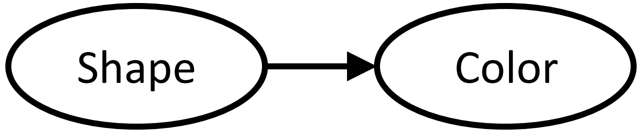
How many samples?

- Rejection sampling is a form of forward sampling
  - Hoeffding and Chebyshev bounds still hold
  - But now n is the number of samples kept

- If goal is to estimate $P(X|\boldsymbol{e})$
  - Expected fraction of examples kept ~ $P(\boldsymbol{e})$
  - For large Bayesian networks and lots of evidence $P(\boldsymbol{e})$ will be tiny
  - The number of examples kept decreases exponentially with number of observed variables

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 P anymore!
  - Solution: weight by probability of evidence given parents

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

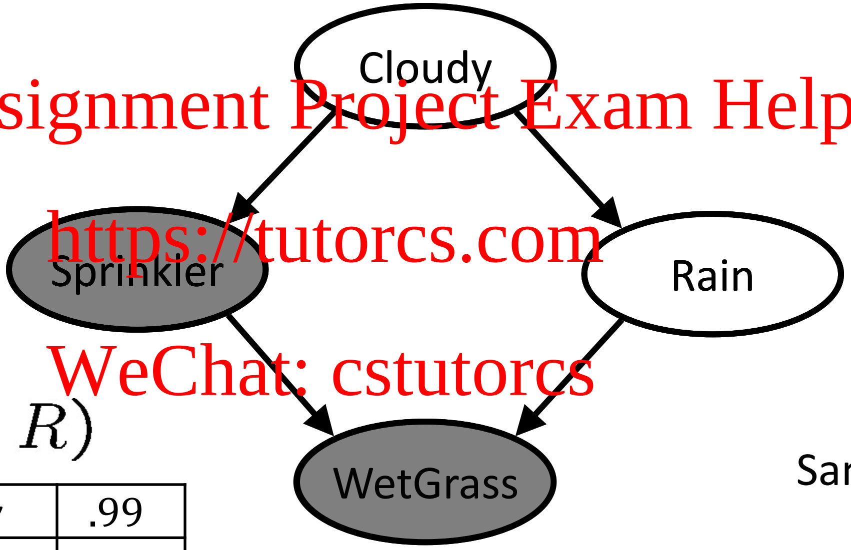
Likelihood Weighting

$P(C)$

c	.5
$\bar{c}$	.5

$P(S|C)$

c	s	.1
	$\bar{s}$	.9
$\bar{c}$	s	.5
	$\bar{s}$	.5



$P(R|C)$

c	r	.8
	$\bar{r}$	.2
$\bar{c}$	r	.2
	$\bar{r}$	.8

$P(W|S,R)$

s	r	w	.99
		$\bar{w}$	.01
	$\bar{r}$	w	.90
		$\bar{w}$	.10
$\bar{s}$	r	w	.90
		$\bar{w}$	.10
	$\bar{r}$	w	.01
		$\bar{w}$	.99

Samples:

Likelihood Weighting

Input: Network N with variables $\boldsymbol{X}$ inducing a distribution P, evidence $\boldsymbol{e} \in \boldsymbol{E}$

Output: one instance Σ and associated weight w

$\pi \leftarrow$ topological order of the nodes in N

$\Sigma \leftarrow \top$ # Trivial instantiation

$w \leftarrow 1$ # Initial weight for Σ

for $i = 1$ **to** n **do** # Network has n variables

    $X \leftarrow$ variable at position i in order π

    $\boldsymbol{u} \leftarrow$ value of X's parents in instantiation Σ

    if $X \in \boldsymbol{E}$

        $x \leftarrow e_i$

        $w \leftarrow w \times P(\mathrm{x}|\boldsymbol{u})$

    else

        $x \leftarrow$ value of X sampled according to $P(X|\boldsymbol{u})$

    $\Sigma \leftarrow \Sigma, x$ # Append value x to Σ

return Σ, w

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 (C is not more likely to get a value matching the evidence)
- We would like to consider evidence when we sample every variable
  - Gibbs sampling
  - This sampling method will allow us to sample Markov Networks

Gibbs Sampling

- *Procedure:* keep track of a full instantiation $x_1, x_2, \dots, x_n$. Start with an 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.

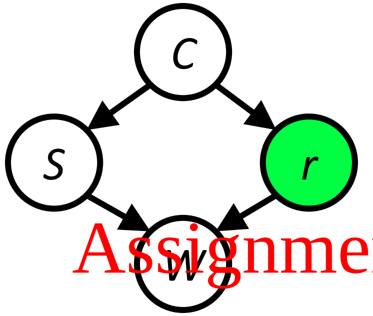
- *Property:* in the limit of repeating this infinitely many times the resulting sample is coming from the correct distribution

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

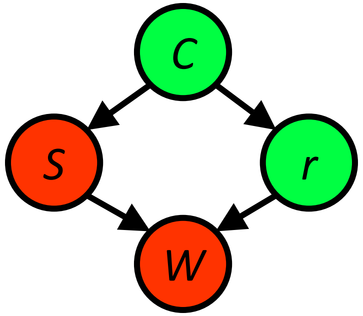
- In contrast: likelihood weighting only conditions on upstream evidence, and hence weights obtained in likelihood weighting can sometimes be very small. Sum of weights over all samples is indicative of how many “effective” samples were obtained, so we want high weights

Gibbs Sampling Example: $P(S|r)$

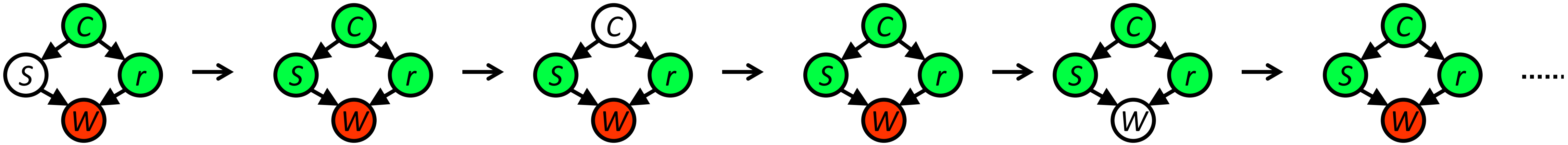
- Step 1: Fix evidence
  - $R = r$



- Step 2: Initialize other variables



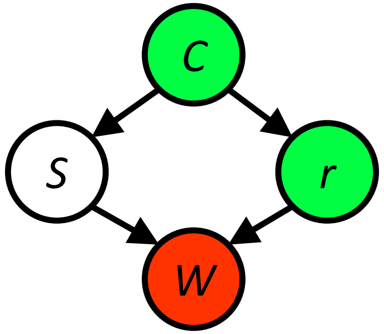
- Steps 3: Repeat
  - Choose a non-evidence variable X
  - Resample X from $P(X \mid \text{all other variables})$



Sample from $P(S|c, \bar{w}, r)$ Sample from $P(C|s, \bar{w}, r)$ Sample from $P(W|s, c, r)$

Efficient Resampling of One Variable

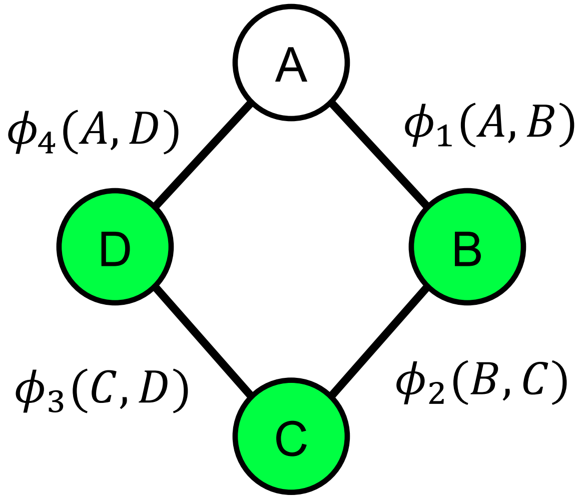
- Sample from $P(S|c,r,\bar{w})$



- 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 Network Example

- Sample from $P(A|b,c,d)$



Gibbs Sampling and Markov Chain

- 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

  - Gibbs sampling is a Markov chain

  - We start with a sample from an arbitrary distribution P_0 and approximate a stationary distribution π

Markov Chain and Sampling

- Our goal is to compute $P(\boldsymbol{x})$
  - However, P may be too hard to sample directly
  - For instance, we need to sample from $P(\boldsymbol{X}|\boldsymbol{e})$ but $P(\boldsymbol{e})$ is very small
- We construct a Markov chain T
  - Whose unique stationary distribution is P
  - Therefore, T should be irreducible (regular)
- We start sampling x_0 from P_0
  - But we cannot use these initial samples
  - Since they are not from the stationary distribution π

Markov Chain and Sampling

- Continue sampling from the transition probability $P(X_t \mid X_{t-1})$

- We want to use samples that are sampled from the distribution close to P
  - But, at early iteration P_t is usually far from P
  - Therefore, we need to wait for the chain to converge to the stationary distribution

- We say that we want to start collecting samples after the chain has run long enough to "mix"
  - P_t is close to the stationary distribution π

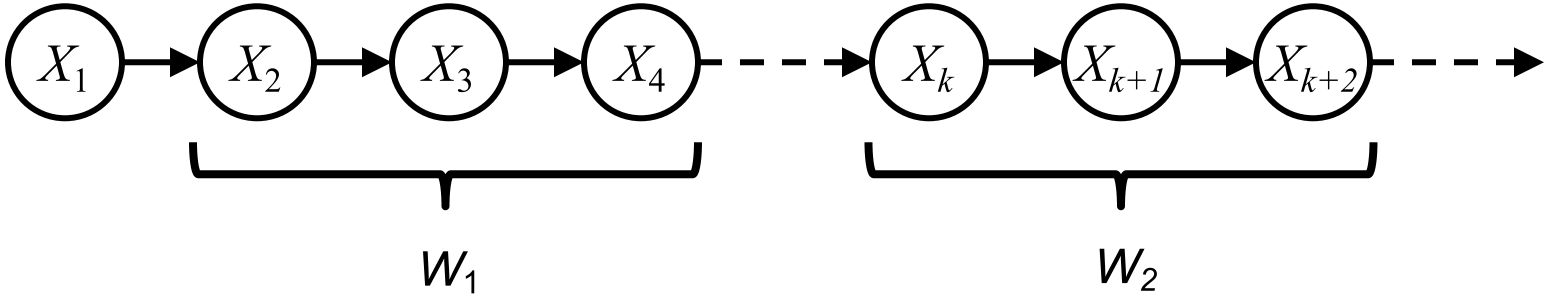
"Mixing" Chains

- The term "mix" is used to denote that the chain is close to π
  - We want to start sample only after the chain has mixed
  - However, how do we know if a chain has mixed?

- Unfortunately, we usually do not know
  - The practice is to run several tests

  - If no evidence of a non-mixing chain is found, we assume the chain has mixed

- How do we know a chain has not mixed?
  - Compare chain statistics in different windows within a single run of the chain
  - Across different runs initialized differently (recommended)



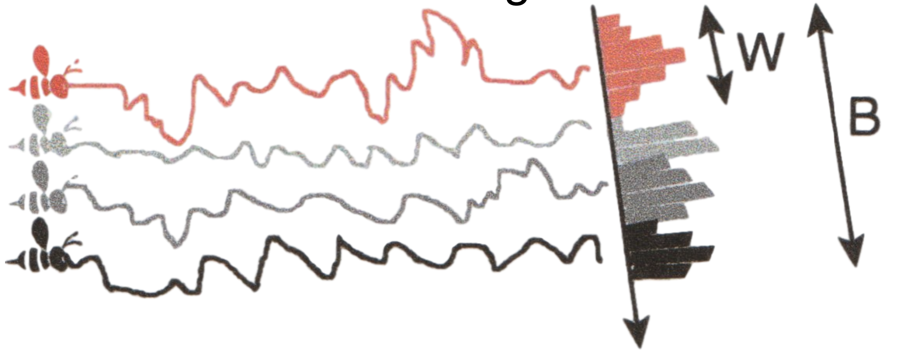
Measuring Convergence

- One popular method is to compare within (W) and between (B) chain variance
  - $\bar{\theta}_j = (1/n) \sum_{i=1}^{n} f_\alpha(x_{i,j})$
  - $\bar{\theta} = (1/c) \sum_{j=1}^{c} \bar{\theta}_j$
  - $\sigma_j^2 = (1/n-1) \sum_{i=1}^{n} (f_\alpha(x_{i,j}) - \bar{\theta}_j)^2$
  - $W = (1/c) \sum_{j=1}^{c} \sigma_j^2$
  - $B = (n/c-1) \sum_{j=1}^{c} (\bar{\theta}_j - \bar{\theta})^2$

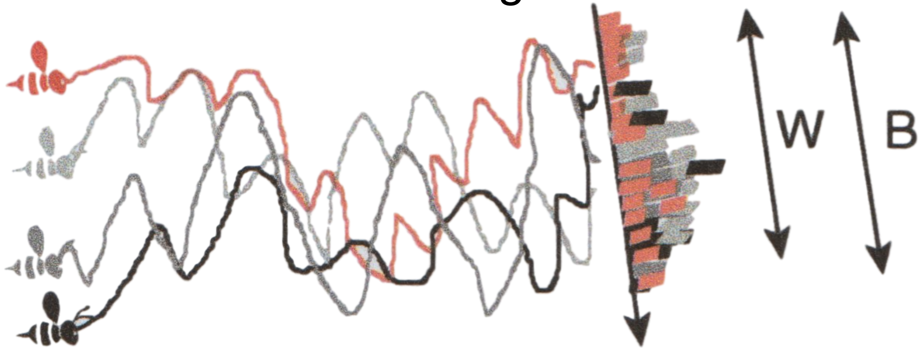
- Then we calculate
  - $\hat{R} = \sqrt{\frac{W + 1/n(B-W)}{W}}$
  - Initially $\hat{R} \gg 1$
  - Estimate $\hat{R}$ across several parameters and stop when all parameters satisfy $\hat{R} < 1.1$

If $W \rightarrow B$ or $n \rightarrow \infty$ then numerator $\rightarrow W$

Non-mixing



Mixing



Chain Samples

- Once the chain mixes, all samples are from the stationary distribution
  - We should use all samples x_t for $t > T_{mix}$
- Nearby samples are correlated
  - The examples are not independent
  - The effective sample size is smaller than independent sampling
- The faster the chain mixes, the less correlated are the samples
  - Slow convergence indicates we move slowly in the space

MCMC Algorithm *Burn-in*

Input: Network N with variables $\boldsymbol{X}$ inducing distribution P**,** evidence $\boldsymbol{e} \in \boldsymbol{E}$, number of chains c

Output: instances $\Sigma_{i,j}$

$i \leftarrow 0$

for $j = 1$ **to** c **do**

$\Sigma_{i,j} \leftarrow$ a complete instantiation of variables $\boldsymbol{X} - \boldsymbol{E}$ # Samples from $P(\boldsymbol{X}|\boldsymbol{e})$ if possible

while not mixed

for $j = 1$ **to** c **do**

$X \leftarrow$ a variable chosen randomly from $\boldsymbol{X} - \boldsymbol{E}$

$x \leftarrow$ value of X sampled according to $P(X|\Sigma_{i-1,j} - X, \boldsymbol{e})$

$\Sigma_{i,j} \leftarrow \Sigma_{i-1,j} \oplus x$ # Change value x in $\Sigma_{i-1,j}$

$i \leftarrow i + 1$

compute convergence criterion over windows of Σ such as $\hat{R}$

return $\Sigma_{i,j}$ for each chain j

MCMC Algorithm Sampling

Input: Network N with variables $\boldsymbol{X}$ inducing distribution P, evidence $\boldsymbol{e} \in \boldsymbol{E}$, number of chains c, instances Σ_j

Output: set of instances D

$D \leftarrow \emptyset$

while not sufficient samples

    for $j = 1$ **to** c **do**

        $X \leftarrow$ a variable chosen randomly from $\boldsymbol{X} - \boldsymbol{E}$

        $x \leftarrow$ value of X sampled according to $P(X|\Sigma_j - X, \boldsymbol{e})$

        $\Sigma_j \leftarrow \Sigma_j \oplus x$ # Change value x in Σ_j

        $D \leftarrow D \cup \Sigma_j$

return D

Gibbs Sampling Revisited

- Target distribution $P(\boldsymbol{X})$
  - Gibbs distribution
  - It can represent equally well normalized or unnormalized factors

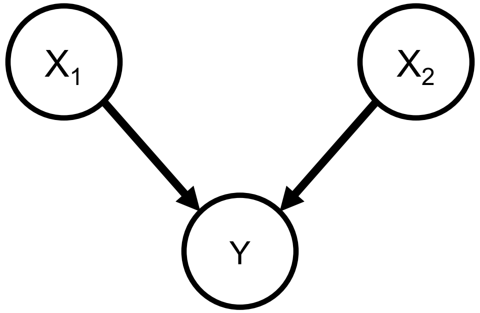
- Markov chain state space

  - Complete assignments $\boldsymbol{x}$ from $\boldsymbol{X}$

  - The state space is exponentially large in the number of variables
- Transition model given starting state $\boldsymbol{x}$:
  - For each variable X_i
  - Sample $x_i \sim P(X_i | \boldsymbol{x} - X_i, \boldsymbol{e})$

Gibbs Chain and Regularity

- Is the Gibbs chain irreducible (regular)?
  - Not always
  - However, if all factors are positive, Gibbs chain is regular



$$Y = X_1 \,\mathrm{XOR}\, X_2$$

X_1	X_2	Y	$P(X_1,X_2,Y)$
0	0	0	0.25
0	0	1	0
0	1	0	0
0	1	1	0.25
1	0	0	0
1	0	1	0.25
1	1	0	0.25
1	1	1	0

Conclusion

- Sampling are powerful techniques for approximate inference
  - Forward, rejection and likelihood sampling require traversing the network in topological order
    - Applicable to Bayesian Networks
  - MCMC works with complete states and therefore do not require node ordering
    - Applicable to Bayesian and Markov Networks
- However, these techniques have limitations
  - MCMC has parameters/design choices, may have slow convergence and it is difficult to tell when the chains mixes
  - Gibbs sampling is computational efficient but typically slow to mix
- Task
  - Read chapter 15