Overview

1. Probabilistic Reasoning/Graphical models
2. Importance Sampling
3. Markov Chain Monte Carlo: Gibbs Sampling
4. Sampling in presence of Determinism
5. Rao-Blackwellisation
6. AND/OR importance sampling

Overview

1. Probabilistic Reasoning/Graphical models
2. Importance Sampling
3. Markov Chain Monte Carlo: Gibbs Sampling
4. Sampling in presence of Determinism
5. Cutset-based Variance Reduction
6. AND/OR importance sampling

Probabilistic Reasoning; Graphical models

- Graphical models:
  - Bayesian network, constraint networks, mixed network
- Queries
- Exact algorithm
  - using inference,
  - search and hybrids
- Graph parameters:
  - tree-width, cycle-cutset, w-cutset

Queries

- **Probability of evidence (or partition function)**

$$P(e) = \sum_{X-\mathrm{var}(e)} \prod_{i=1}^{n} P(x_i \mid pa_i)|_e \qquad Z = \sum_X \prod_i \psi_i(C_i)$$

- **Posterior marginal (beliefs):**

$$P(x_i \mid e) = \frac{P(x_i, e)}{P(e)} = \frac{\sum_{X-\mathrm{var}(e)-X_i} \prod_{j=1}^{n} P(x_j \mid pa_j)|_e}{\sum_{X-\mathrm{var}(e)} \prod_{j=1}^{n} P(x_j \mid pa_j)|_e}$$

- **Most Probable Explanation**

$$\bar{\mathbf{x}}^* = \arg\max_{\bar{\mathbf{x}}} P(\bar{\mathbf{x}}, \mathbf{e})$$

Approximation

- Since inference, search and hybrids are too expensive when graph is dense; (high treewidth) then:

- Bounding inference:
    - mini-bucket and mini-clustering
    - Belief propagation

- **Bounding search:**
    - **Sampling**

- Goal: an anytime scheme

Overview

1. Probabilistic Reasoning/Graphical models
2. Importance Sampling
3. Markov Chain Monte Carlo: Gibbs Sampling
4. Sampling in presence of Determinism
5. Rao-Blackwellisation
6. AND/OR importance sampling

Outline

- Definitions and Background on Statistics
- Theory of importance sampling
- Likelihood weighting
- State-of-the-art importance sampling techniques

A sample

- Given a set of variables X={X_1,...,X_n}, a sample, denoted by S^t is an instantiation of all variables:

$$S^t = (x_1^t, x_2^t, ..., x_n^t)$$

How to draw a sample ?
Univariate distribution

- Example: Given random variable X having domain {0, 1} and a distribution P(X) = (0.3, 0.7).
- Task: Generate samples of X from P.
- How?
  - draw random number $r \in [0, 1]$
  - If ($r < 0.3$) then set $X=0$
  - Else set $X=1$

How to draw a sample? Multi-variate distribution

- Let X={X_1,..,X_n} be a set of variables
- Express the distribution in product form

$$P(X) = P(X_1) \times P(X_2 \mid X_1) \times \ldots \times P(X_n \mid X_1, \ldots, X_{n-1})$$

- Sample variables one by one from left to right, along the ordering dictated by the product form.
- Bayesian network literature: Logic sampling

Sampling for Prob. Inference Outline

- **Logic Sampling**
- Importance Sampling
  - Likelihood Sampling
  - Choosing a Proposal Distribution
- Markov Chain Monte Carlo (MCMC)
  - Metropolis-Hastings
  - Gibbs sampling
- Variance Reduction

Logic Sampling: No Evidence (Henrion 1988)

Input: Bayesian network

$X= \{X_1,\ldots,X_N\}$, N- #nodes, T - # samples

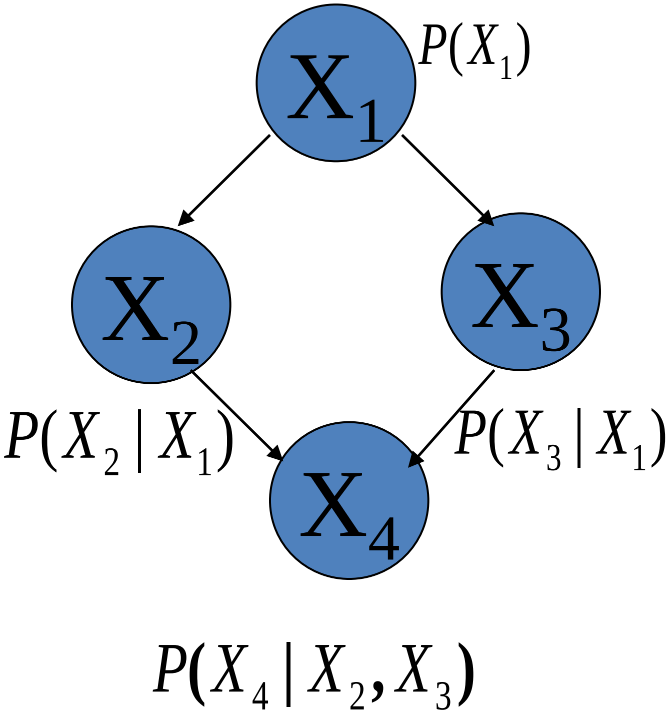
Output: T samples

Process nodes in topological order – first process the ancestors of a node, then the node itself:

```
For t = 0 to T
    For i = 0 to N
        $X_i \leftarrow$ sample $x_i^t$ from $P(x_i \mid pa_i)$
```

Logic sampling (example)

$$P(X_1, X_2, X_3, X_4) = P(X_1) \times P(X_2 \mid X_1) \times P(X_3 \mid X_1) \times P(X_4 \mid X_2, X_3)$$



No Evidence

```
// generate sample $k$
1. Sample $x_1$ from $P(x_1)$
2. Sample $x_2$ from $P(x_2 \mid X_1 = x_1)$
3. Sample $x_3$ from $P(x_3 \mid X_1 = x_1)$
4. Sample $x_4$ from $P(x_4 \mid X_2 = x_{2,} X_3 = x_3)$
```

Logic Sampling w/ Evidence

Input: Bayesian network

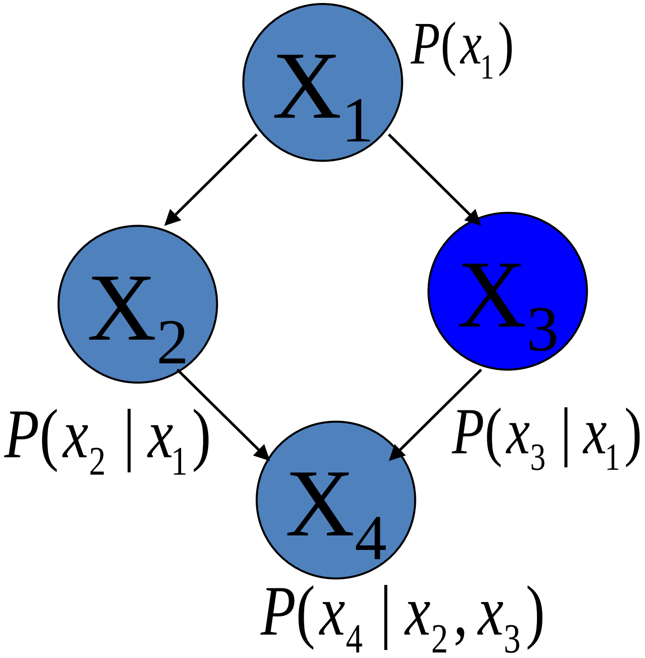
$X = \{X_1,\ldots,X_N\}$, N- #nodes

E – evidence, T - # samples

Output: T samples consistent with E

```
1. For t=1 to T
2.     For i=1 to N
3.         $X_i \leftarrow$ sample $x_i^t$ from $P(x_i \mid pa_i)$
4.         If $X_i$ in E and $X_i \neq x_i$, reject sample:
5.             Goto Step 1.
```

Logic Sampling (example)



$\text{Evidence} : X_3 = 0$

// generate sample k

1. Sample x_1 from $P(x_1)$
2. Sample x_2 from $P(x_2 | x_1)$
3. Sample x_3 from $P(x_3 | x_1)$
4. If $x_3 \neq 0$, reject sample and start from 1, otherwise
5. Sample x_4 from $P(x_4 | x_{2,} x_3)$

Expected value and Variance

Expected value: Given a probability distribution P(X) and a function g(X) defined over a set of variables X = {X_1, X_2, … X_n}, the expected value of g w.r.t. P is

$$E_P[g(x)] = \sum_x g(x)P(x)$$

Variance: The variance of g w.r.t. P is:

$$Var_P[g(x)] = \sum_x \left[g(x) - E_P[g(x)]\right]^2 P(x)$$

Monte Carlo Estimate

- **Estimator:**
  - An estimator is a function of the samples.
  - It produces an estimate of the unknown parameter of the sampling *distribution.*

Given i.i.d. samples $\mathrm{S}^1, \mathrm{S}^2, \ldots \mathrm{S}^\mathrm{T}$ drawn from P,

the Montecarlo estimate of $\mathrm{E}_P[\mathrm{g}(\mathrm{x})]$ is given by:

$$\hat{g} = \frac{1}{T} \sum_{t=1}^{T} g(S^t)$$

Example: Monte Carlo estimate

- Given:
  - A distribution P(X) = (0.3, 0.7).
  - g(X) = 40 if X equals 0

    = 50 if X equals 1.
- Estimate $E_P[g(x)]=(40x0.3+50x0.7)=47$.
- Generate k samples from P: 0,1,1,1,0,1,1,0,1,0

$$\hat{g} = \frac{40 \times \# samples(X = 0) + 50 \times \# samples(X = 1)}{\# samples}$$

$$= \frac{40 \times 4 + 50 \times 6}{10} = 46$$

Outline

- Definitions and Background on Statistics
- **Theory of importance sampling**
- Likelihood weighting
- State-of-the-art importance sampling techniques

Importance sampling: Main idea

- Express query as the expected value of a random variable w.r.t. to a distribution Q.
- Generate random samples from Q.
- Estimate the expected value from the generated samples using a monte carlo estimator (average).

Importance sampling for P(e)

$Let\ Z = X \setminus E,$

Let Q(Z) be a (proposal) distribution, satisfying

$P(z,e) > 0 \Rightarrow Q(z) > 0$

Then, we can rewrite P(e) as :

$$P(e) = \sum_{z} P(z,e) = \sum_{z} P(z,e) \frac{Q(z)}{Q(z)} = E_Q\left[\frac{P(z,e)}{Q(z)}\right] = E_Q[w(z)]$$

MonteCarlo estimate **:**

$$\hat{P}(e) = \frac{1}{T}\sum_{t=1}^{T} w(z^t), \text{where } z^t \leftarrow Q(Z)$$

Properties of IS estimate of P(e)

- **Convergence:** by law of large numbers

$$\hat{P}(e) = \frac{1}{T}\sum_{i=1}^{T} w(z^i) \xrightarrow{a.s.} P(e) \text{ for T} \to \infty$$

- **Unbiased.**

$$E_Q[\hat{P}(e)] = P(e)$$

- **Variance:**

$$Var_Q\left[\hat{P}(e)\right] = Var_Q\left[\frac{1}{T}\sum_{i=1}^{N} w(z^i)\right] = \frac{Var_Q[w(z)]}{T}$$

Properties of IS estimate of P(e)

- Mean Squared Error of the estimator

$$MSE_Q\left[\hat{P}(e)\right] = E_Q\left[\left(\hat{P}(e) - P(e)\right)^2\right]$$

$$= \left(P(e) - E_Q[\hat{P}(e)]\right)^2 + Var_Q\left[\hat{P}(e)\right]$$

$$= Var_Q\left[\hat{P}(e)\right]$$

$$= \frac{Var_Q[w(x)]}{T}$$

This quantity enclosed in the brackets is zero because the expected value of the estimator equals the expected value of g(x)

Estimating $P(X_i|e)$

Let $\delta_{x_i}(z)$ be a dirac - delta function, which is 1 if z contains x_i and 0 otherwise.

$$P(x_i \mid e) = \frac{P(x_i, e)}{P(e)} = \frac{\sum_z \delta_{x_i}(z) P(z,e)}{\sum_z P(z,e)} = \frac{E_Q\left[\frac{\delta_{x_i}(z) P(z,e)}{Q(z)}\right]}{E_Q\left[\frac{P(z,e)}{Q(z)}\right]}$$

Idea : Estimate numerator and denominator by IS.

$$\text{Ratio estimate}: \overline{P}(x_i \mid e) = \frac{\hat{P}(x_i, e)}{\hat{P}(e)} = \frac{\sum_{k=1}^{T} \delta_{x_i}(z^k) w(z^k, e)}{\sum_{k=1}^{T} w(z^k, e)}$$

Estimate is biased : $\mathrm{E}\left[\overline{P}(x_i \mid e)\right] \neq P(x_i \mid e)$

Properties of the IS estimator for $P(X_i|e)$

- Convergence: By Weak law of large numbers

$$\overline{P}(x_i \mid e) \rightarrow P(x_i \mid e) \text{ as T} \rightarrow \infty$$

- Asymptotically unbiased

$$\lim_{T \rightarrow \infty} E_P[\overline{P}(x_i \mid e)] = P(x_i \mid e)$$

- Variance
  - Harder to analyze
  - Liu suggests a measure called “Effective sample size”

Generating samples from Q

- No restrictions on “how to”
- Typically, express Q in product form:
  - $Q(Z)=Q(Z_1)xQ(Z_2|Z_1)x....xQ(Z_n|Z_1,..Z_{n-1})$
- Sample along the order $Z_1,..,Z_n$
- Example:
  - $Z_1 \Leftarrow Q(Z_1)=(0.2,0.8)$
  - $Z_2 \Leftarrow Q(Z_2|Z_1)=(0.1,0.9,0.2,0.8)$
  - $Z_3 \Leftarrow Q(Z_3|Z_1,Z_2)=Q(Z_3)=(0.5,0.5)$

Outline

- Definitions and Background on Statistics
- Theory of importance sampling
- **Likelihood weighting**
- State-of-the-art importance sampling techniques

Likelihood Weighting

(Fung and Chang, 1990; Shachter and Peot, 1990)

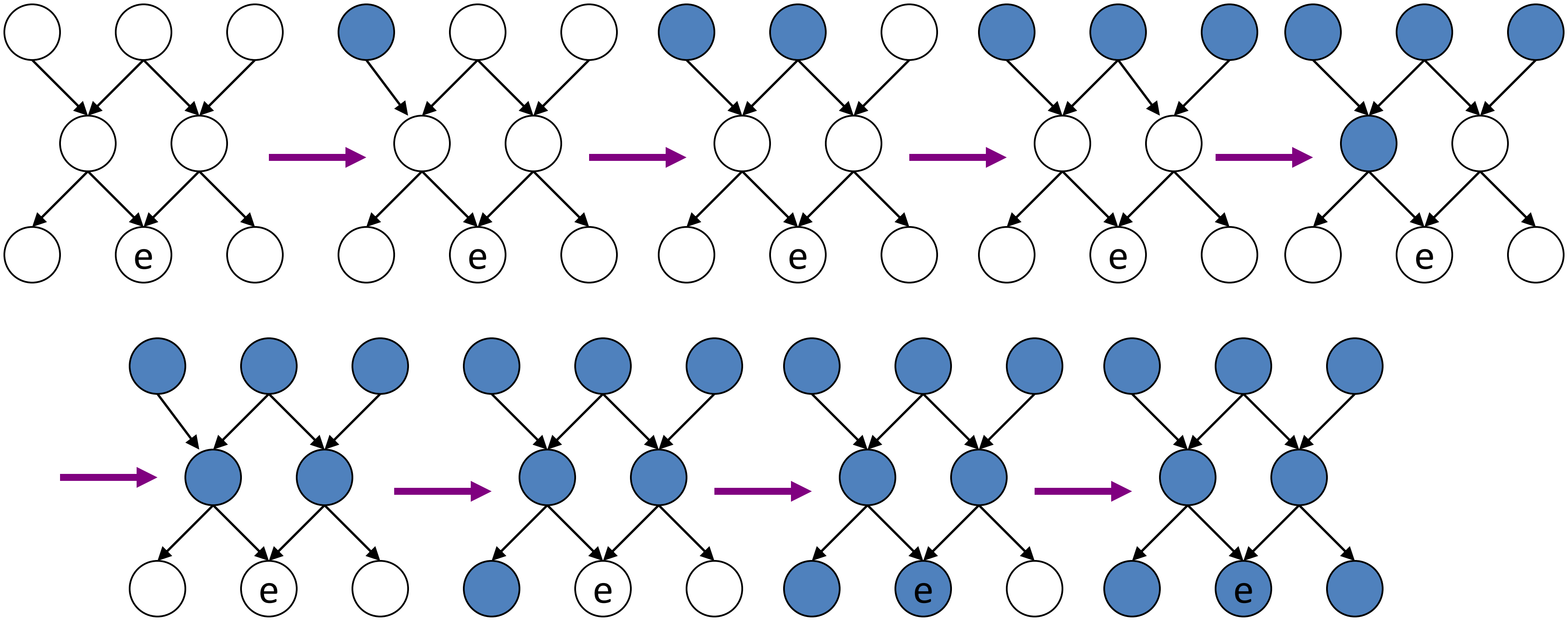
Is an instance of importance sampling!

“Clamping” evidence+
logic sampling+
weighing samples by evidence likelihood

Works well for *likely evidence!*

Likelihood Weighting: Sampling

Sample in topological order over **X** !



Clamp evidence, Sample $x_i \leftarrow P(X_i|pa_i)$, $P(X_i|pa_i)$ is *a look-up* in CPT!

Likelihood Weighting: Proposal Distribution

$$Q(X \setminus E) = \prod_{X_i \in X \setminus E} P(X_i \mid pa_i, e)$$

Notice: Q is another Bayesian network

Example :

Given a Bayesian network : $P(X_1, X_2, X_3) = P(X_1) \times P(X_2 \mid X_1) \times P(X_3 \mid X_1, X_2)$ and

Evidence $X_2 = x_2$.

$Q(X_1, X_3) = P(X_1) \times P(X_3 \mid X_1, X_2 = x_2)$

Weights :

Given a sample : $x = (x_1, .., x_n)$

$$w = \frac{P(x,e)}{Q(x)} = \frac{\cancel{\prod_{X_i \in X \setminus E} P(x_i \mid pa_i}, e) \times \prod_{E_j \in E} P(e_j \mid pa_j)}{\cancel{\prod_{X_i \in X \setminus E} P(x_i \mid pa_i, e)}}$$

$$= \prod_{E_j \in E} P(e_j \mid pa_j)$$

Likelihood Weighting: Estimates

Estimate P(e): $\hat{P}(e) = \frac{1}{T}\sum_{t=1}^{T} w^{(t)}$

Estimate Posterior Marginals:

$$\hat{P}(x_i \mid e) = \frac{\hat{P}(x_i, e)}{\hat{P}(e)} = \frac{\sum_{t=1}^{T} w^{(t)} g_{x_i}(x^{(t)})}{\sum_{t=1}^{T} w^{(t)}}$$

$g_{x_i}(x^{(t)}) = 1$ if $\mathrm{x_i} = x_i^t$ and equals zero otherwise

Likelihood Weighting

- Converges to exact posterior marginals
- Generates Samples Fast
- Sampling distribution is close to prior (especially if E $\subset$ Leaf Nodes)
- Increasing sampling variance

$\Rightarrow$Convergence may be slow

$\Rightarrow$Many samples with $P(x^{(t)})=0$ rejected

Outline

- Definitions and Background on Statistics
- Theory of importance sampling
- Likelihood weighting
- **Error estimation**
- State-of-the-art importance sampling techniques

Bounds on the Absolute Error

The absolute error of an estimate $\mathrm{Av}_n(\breve{\alpha})$

is the absolute difference it has with the true probability $\mathrm{Pr}(\alpha)$ we are trying to estimate.

For any $\epsilon > 0$, we have

$$\mathbb{P}\Big(|\mathrm{Av}_n(\breve{\alpha}) - \mathrm{Pr}(\alpha)| < \epsilon\Big) \geq 1 - \frac{\mathrm{Pr}(\alpha)\mathrm{Pr}(\neg\alpha)}{n\epsilon^2}$$

The estimate $\mathrm{Av}_n(\breve{\alpha})$ computed by direct sampling will fall within the interval $(\mathrm{Pr}(\alpha) - \epsilon, \mathrm{Pr}(\alpha) + \epsilon)$ with probability at least $1 - \mathrm{Pr}(\alpha)\mathrm{Pr}(\neg\alpha)/n\epsilon^2$

Bounds on the Absolute Error

A sharper bound which does not depend on the probability $\Pr(\alpha)$

Hoeffding's inequality

Let $\mathrm{Av}_n(f)$ be a sample mean, where the function f has expectation μ and values in $\{0, 1\}$. For any $\epsilon > 0$, we have:

$$\mathbb{P}\Big(|\mathrm{Av}_n(f) - \mu| \le \epsilon\Big) \ge 1 - 2e^{-2n\epsilon^2}$$

For any $\epsilon > 0$, we have:

$$\mathbb{P}\Big(|\mathrm{Av}_n(\breve{\alpha}) - \Pr(\alpha)| \le \epsilon\Big) \ge 1 - 2e^{-2n\epsilon^2}$$

The estimate $\mathrm{Av}_n(\breve{\alpha})$ computed by direct sampling will fall within the interval $(\Pr(\alpha) - \epsilon, \Pr(\alpha) + \epsilon)$ with probability at least $1 - 2e^{-2n\epsilon^2}$

Bounds on the Relative Error

For any $\epsilon > 0$, we have:

$$\mathbb{P}\Big(\frac{|\mathrm{Av}_n(\breve{\alpha}) - \mathrm{Pr}(\alpha)|}{\mathrm{Pr}(\alpha)} \leq \epsilon\Big) \geq 1 - 2e^{-2n\epsilon^2 \mathrm{Pr}(\alpha)^2}$$

Require the probability $\mathrm{Pr}(\alpha)$ (or some lower bound on it).

Bounds on the Relative Error

The relative error of an estimate $\mathrm{Av}_n(\breve{\alpha})$

$$\frac{|\mathrm{Av}_n(\breve{\alpha}) - \Pr(\alpha)|}{\Pr(\alpha)}$$

The bound on the absolute error becomes tighter as the probability of an event becomes more extreme. Yet, the corresponding bound on the relative error becomes looser as the probability of an event becomes more extreme.

Example

For an event with probability .5 and a sample size of 10000, there is a 95% chance that the absolute error is $\approx$ 4.5%. However, for the same confidence level, the relative error increases to $\approx$ 13.4% if the event has probability .1, and increases again to $\approx$ 44.5% if the event has probability .01

Outline

- Definitions and Background on Statistics
- Theory of importance sampling
- Likelihood weighting
- **State-of-the-art importance sampling techniques**

Proposal selection

- One should try to select a proposal that is as close as possible to the posterior distribution.

$$Var_Q\left[\hat{P}(e)\right] = \frac{Var_Q[w(z)]}{T} = \frac{1}{N}\sum_{z\in Z}\left(\frac{P(z,e)}{Q(z)} - P(e)\right)^2 Q(z)$$

$$\frac{P(z,e)}{Q(z)} - P(e) = 0, \text{ to have a zero-variance estimator}$$

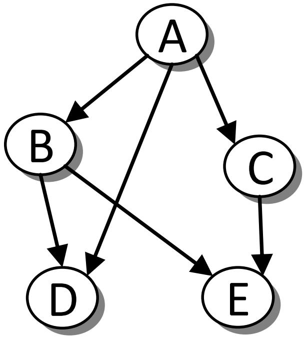
$$\therefore \frac{P(z,e)}{P(e)} = Q(z)$$

$$\therefore Q(z) = P(z\,|\,e)$$

Perfect sampling using Bucket Elimination

- Algorithm:
  - Run Bucket elimination on the problem along an ordering o=(X_N,..,X_1).
  - Sample along the reverse ordering: (X_1,..,X_N)
  - At each variable X_i, recover the probability $P(X_i|x_1,\ldots,x_{i-1})$ by referring to the bucket.

Bucket Elimination



Query: $P(a \mid e=0) \propto P(a, e=0)$ Elimination Order: d,e,b,c

$$P(a,e=0) = \sum_{c,b,e=0,d} P(a)P(b \mid a)P(c \mid a)P(d \mid a,b)P(e \mid b,c)$$

$$= P(a)\sum_{c} P(c \mid a)\sum_{b} P(b \mid a)\sum_{e=0} P(e \mid b,c)\sum_{d} P(d \mid a,b)$$

Original Functions

D: $P(d \mid a,b)$

E: $P(e \mid b,c)$

B: $P(b \mid a)$

C: $P(c \mid a)$

A: $P(a)$

Messages

$$\boxed{f_D(a,b)} = \sum_{d} P(d \mid a,b)$$

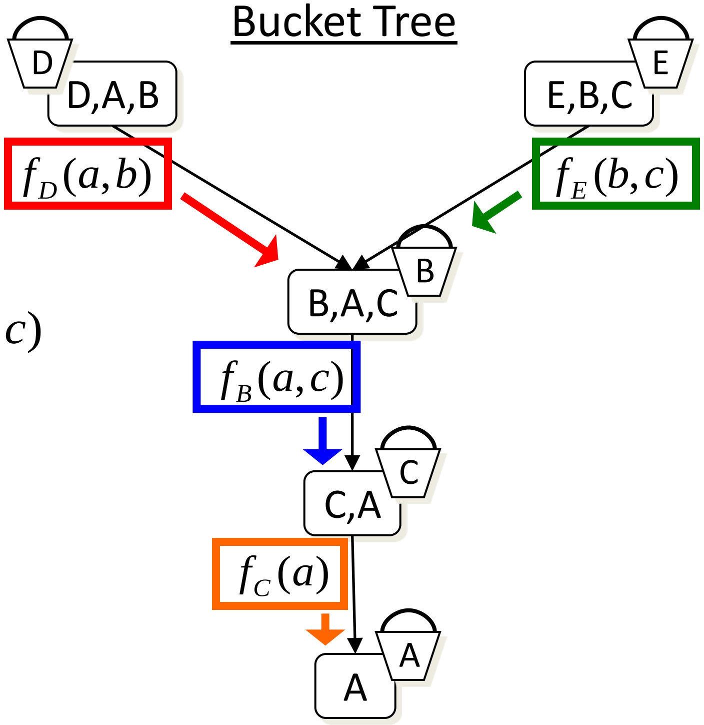
$$\boxed{f_E(b,c)} = P(e=0 \mid b,c)$$

$$\boxed{f_B(a,c)} = \sum_{b} P(b \mid a)\, f_D(a,b)\, f_E(b,c)$$

$$\boxed{f_C(a)} = \sum_{c} P(c \mid a)\, f_B(a,c)$$

$$P(a,e=0) = p(A)\, f_C(a)$$

Bucket Tree



Time and space exp(w*)

Bucket elimination (BE)

Algorithm *elim-bel* (Dechter 1996)

$\sum_b \prod$ ← Elimination operator

bucket B: $\overbrace{P(B|A) \quad P(D|B,A) \quad P(e|B,C)}^{\sum_b \prod}$

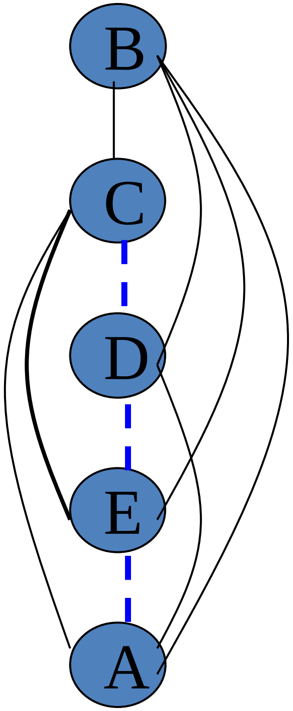
bucket C: $P(C|A)$ $\boldsymbol{h^B(A,D,C,e)}$

bucket D: $\boldsymbol{h^C(A,D,e)}$

bucket E: $\boldsymbol{h^D(A,e)}$

bucket A: $P(a)$ $\boldsymbol{h^E(a)}$

$\boldsymbol{P(e)}$



Sampling from the output of BE

(Dechter 2002)

$\text{Set } A = a, D = d, C = c \text{ in the bucket}$

$\text{Sample}: B = b \leftarrow Q(B \mid a, e, d) \propto P(B \mid a) P(d \mid B, a) P(e \mid b, c)$

bucket B: $P(B|A)$ $P(D|B,A)$ $P(e|B,C)$

bucket C: $P(C|A)$ $h^B(A,D,C,e)$

$\text{Set } A = a, D = d \text{ in the bucket}$

$\text{Sample}: C = c \leftarrow Q(C \mid a, e, d) \propto P(C \mid A) \cdot h^B(a, d, C, e$

bucket D: $h^C(A,D,e)$

$\text{Set } A = a \text{ in the bucket}$

$\text{Sample}: D = d \leftarrow Q(D \mid a, e) \propto h^C(a, D, e)$

bucket E: $h^D(A,e)$

Evidence bucket : ignore

bucket A: $P(A)$ $h^E(A)$

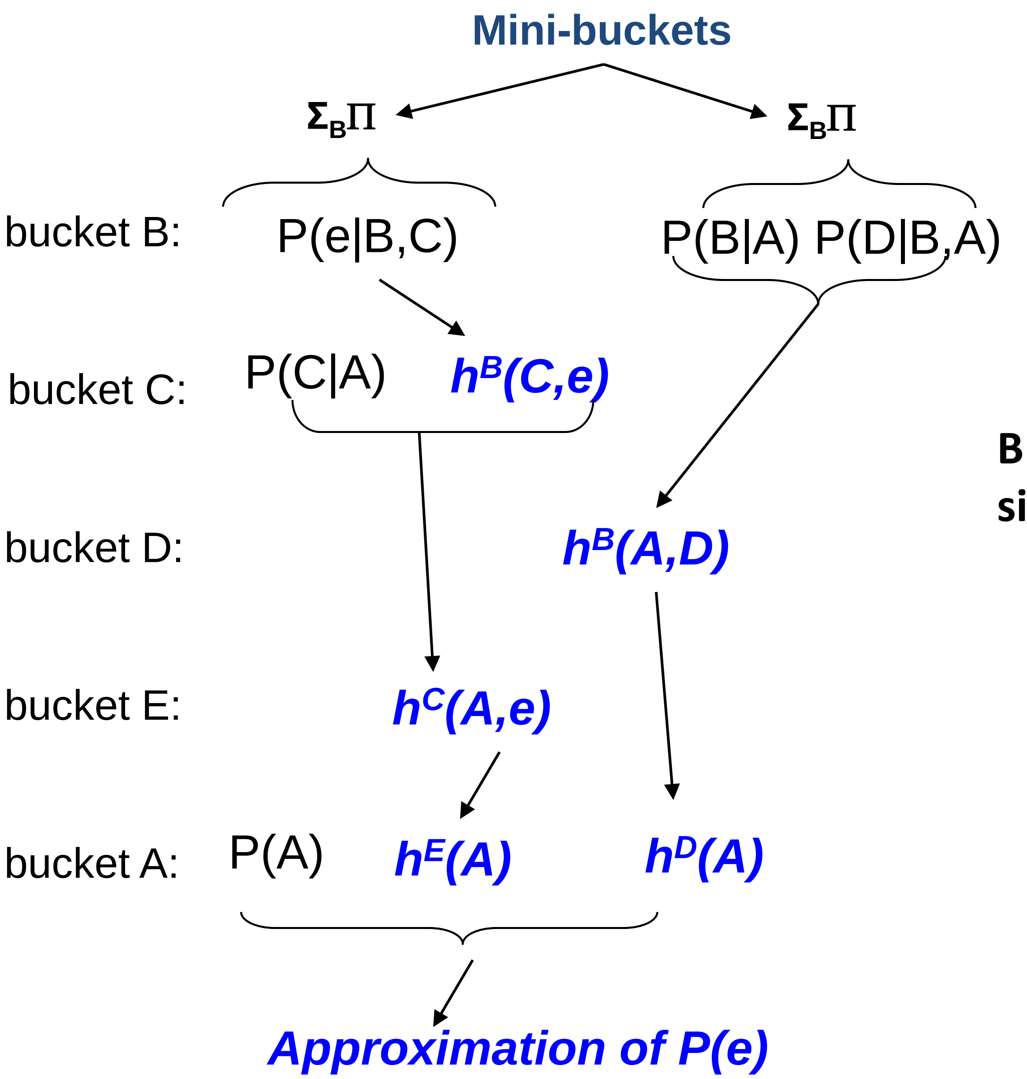
$$\mathbf{Q(A) \propto P(A) \times h^E(A)}$$

$$\textit{Sample}: \mathbf{A = a \leftarrow Q(A)}$$

Mini-buckets: “local inference”

- Computation in a bucket is time and space exponential in the number of variables involved
- Therefore, partition functions in a bucket into “mini-buckets” on smaller number of variables
- Can control the size of each “mini-bucket”, yielding polynomial complexity.

Mini-Bucket Elimination



Mini-buckets

$\Sigma_B \Pi$ $\Sigma_B \Pi$

bucket B: $P(e|B,C)$ $P(B|A)\ P(D|B,A)$

bucket C: $P(C|A)$ $h^B(C,e)$

bucket D: $h^B(A,D)$

bucket E: $h^C(A,e)$

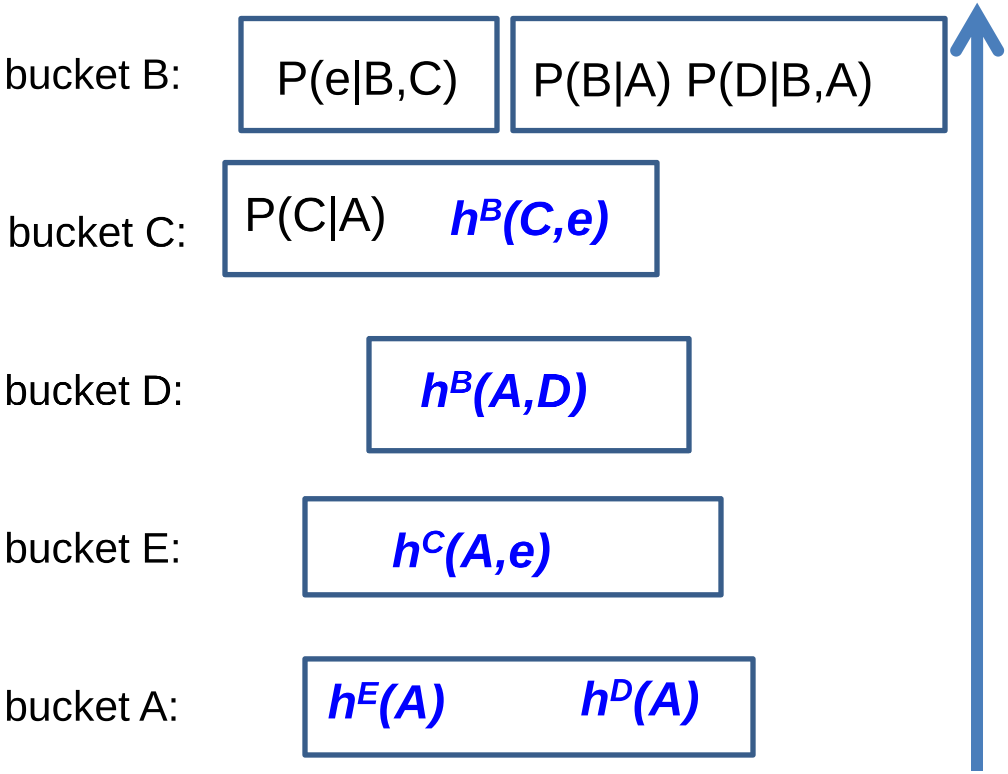
bucket A: $P(A)$ $h^E(A)$ $h^D(A)$

Approximation of $P(e)$

Space and Time constraints: Maximum scope size of the new function generated should be bounded by 2

BE generates a function having scope size 3. So it cannot be used.

Sampling from the output of MBE



bucket B: $P(e|B,C)$ $P(B|A)\ P(D|B,A)$

bucket C: $P(C|A)$ $h^B(C,e)$

bucket D: $h^B(A,D)$

bucket E: $h^C(A,e)$

bucket A: $h^E(A)$ $h^D(A)$

Sampling is same as in BE-sampling except that now we construct Q from a randomly selected “mini-bucket”

IJGP-Sampling
(Gogate and Dechter, 2005)

- Iterative Join Graph Propagation (IJGP)
  - A Generalized Belief Propagation scheme (Yedidia et al., 2002)
- IJGP yields better approximations of P(X|E) than MBE
  - (Dechter, Kask and Mateescu, 2002)
- Output of IJGP is same as mini-bucket “clusters”
- **Currently the best performing IS scheme!**

Current Research question

- Given a Bayesian network with evidence or a Markov network representing function P, generate another Bayesian network representing a function Q (from a family of distributions, restricted by structure) such that Q is closest to P.
- Current approaches
  - Mini-buckets
  - Ijgp
  - Both
- Experimented, but need to be justified theoretically.

Algorithm: Approximate Sampling

1) Run IJGP or MBE
2) At each branch point compute the edge probabilities by consulting output of IJGP or MBE

- Rejection Problem:
  – Some assignments generated are non solutions

Adaptive Importance Sampling

```
Initial Proposal $= \mathrm{Q}^1(Z) = Q(Z_1) \times Q(Z_2 \mid pa(Z_2)) \times \ldots \times Q(Z_n \mid pa(Z_n))$
$\hat{P}(E = e) = 0$
For i = 1 to k do
    Generate samples $\mathrm{z}^1, \ldots, \mathrm{z}^\mathrm{N}$ from $\mathrm{Q}^k$
    $\hat{\mathrm{P}}(\mathrm{E} = \mathrm{e}) = \hat{P}(E = e) + \frac{1}{\mathrm{N}} \sum_{j=1}^{N} w_k(z^i)$
    Update $\mathrm{Q}^{\mathrm{k+1}} = Q^k + \eta(k)\left[Q^k - Q'\right]$
End
Return $\frac{\hat{P}(E = e)}{\mathrm{k}}$
```

Adaptive Importance Sampling

- General case
- Given k proposal distributions
- Take N samples out of each distribution
- Approximate P(e)

$$\hat{P}(e) = \frac{1}{k}\sum_{j=1}^{k}\left[Avg - weight - jth - proposal\right]$$

Estimating Q'(z)

$$\mathrm{Q}^{'}(Z) = Q'(Z_1) \times Q'(Z_2 \mid pa(Z_2)) \times \ldots \times Q'(Z_n \mid pa(Z_n))$$

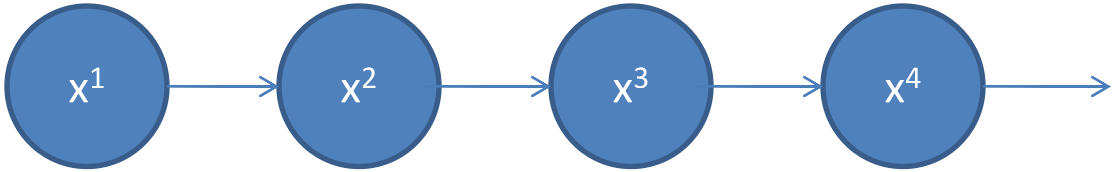
where each $\mathrm{Q}'(\mathrm{Z}_{\mathrm{i}} \mid \mathrm{Z}_1,..,\mathrm{Z}_{\mathrm{i-1}})$

is estimated by importance sampling

Overview

1. Probabilistic Reasoning/Graphical models
2. Importance Sampling
3. Markov Chain Monte Carlo: Gibbs Sampling
4. Sampling in presence of Determinism
5. Rao-Blackwellisation
6. AND/OR importance sampling

Markov Chain



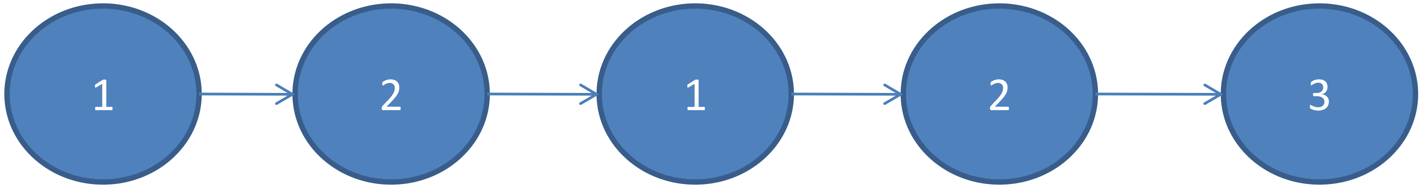
- A **Markov chain** is a discrete random process with the property that the next state depends only on the current state (**Markov Property**):

$$P(x^t \mid x^1, x^2, \ldots, x^{t-1}) = P(x^t \mid x^{t-1})$$

- If $P(X^t|x^{t-1})$ does not depend on t (**time homogeneous**) and state space is finite, then it is often expressed as a **transition function** (aka **transition matrix**) $\sum_x P(X = x) = 1$

Example: Drunkard’s Walk

- a random walk on the number line where, at each step, the position may change by +1 or −1 with equal probability

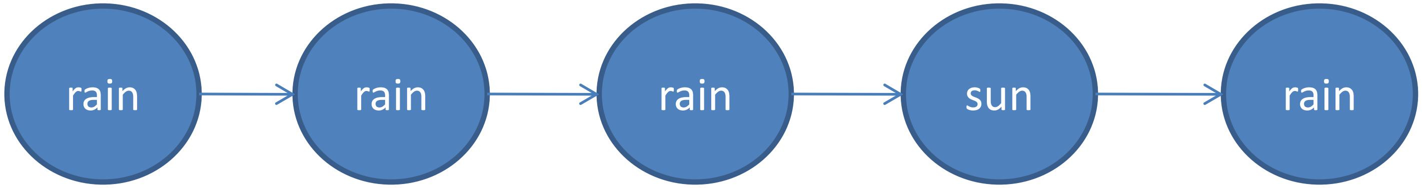


$$D(X) = \{0,1,2,\ldots\}$$

$$\begin{array}{c|cc} & P(n-1) & P(n+1) \\ \hline n & 0.5 & 0.5 \end{array}$$

transition matrix P(X)

Example: Weather Model



$$D(X) = \{rainy, sunny\}$$

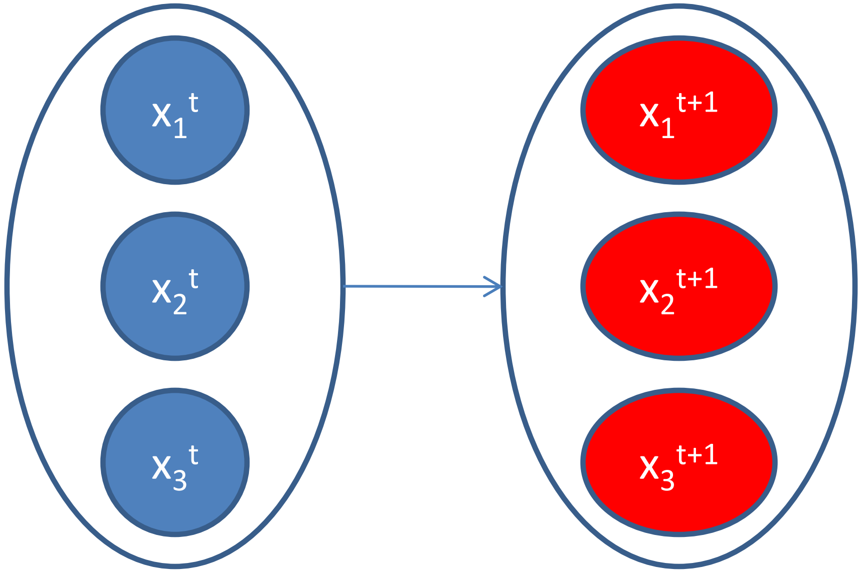
	$P(rainy)$	$P(sunny)$
$rainy$	0.9	0.1
$sunny$	0.5	0.5

transition matrix P(X)

Multi-Variable System

$$X = \{X_1, X_2, X_3\}, D(X_i) = discrete,\ finite$$

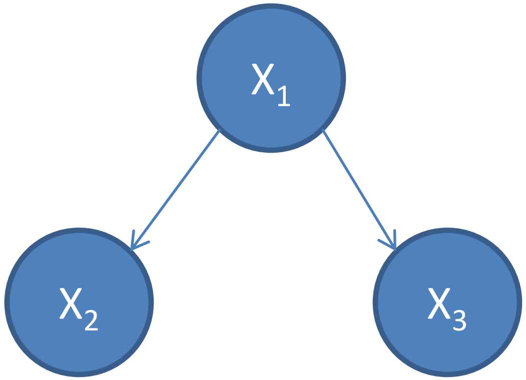
- state is an assignment of values to all the variables

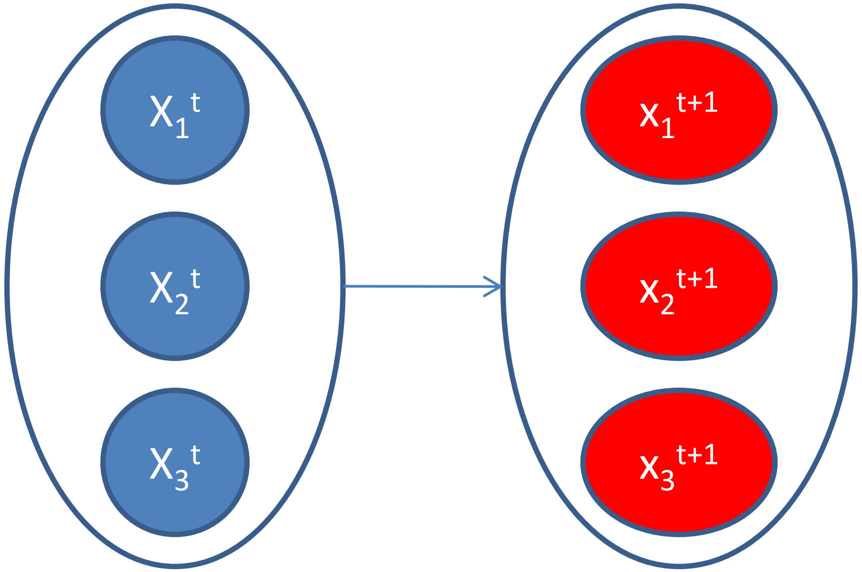


$$x^t = \{x_1^t, x_2^t, \ldots, x_n^t\}$$

Bayesian Network System

- Bayesian Network is a representation of the joint probability distribution over 2 or more variables





$$X = \{X_1, X_2, X_3\}$$

$$x^t = \{x_1^t, x_2^t, x_3^t\}$$

Stationary Distribution Existence

- If the Markov chain is time-homogeneous, then the vector π(X) is a *stationary* distribution (aka *invariant* or *equilibrium* distribution, aka “fixed point”), if its entries sum up to 1 and satisfy:

$$\pi(x_i) = \sum_{x_i \in D(X)} \pi(x_j) P(x_i \mid x_j)$$

- Finite state space Markov chain has a unique stationary distribution if and only if:
  - The chain is irreducible
  - All of its states are positive recurrent

Irreducible

- A state x is *irreducible* if under the transition rule one has nonzero probability of moving from x to any other state and then coming back in a finite number of steps
- If one state is irreducible, then all the states must be irreducible

(Liu, Ch. 12, pp. 249, Def. 12.1.1)

Recurrent

- A state x is *recurrent* if the chain returns to x with probability 1
- Let M(x) be the expected number of steps to return to state x
- State x is *positive recurrent* if M(x) is finite

The recurrent states in a finite state chain are positive recurrent .

Stationary Distribution Convergence

- Consider infinite Markov chain:

$$P^{(n)} = P(x^n \mid x^0) = P^0 P^n$$

- If the chain is both *irreducible* and *aperiodic*, then:

$$\pi = \lim_{n\to\infty} P^{(n)}$$

- Initial state is not important in the limit

"The most useful feature of a "good" Markov chain is its fast forgetfulness of its past…"

(Liu, Ch. 12.1)

Aperiodic

- Define $d(i)$ = g.c.d.{n > 0 | it is possible to go from i to i in n steps}. Here, g.c.d. means the greatest common divisor of the integers in the set. If $d(i)=1$ for $\forall i$, then chain is *aperiodic*
- *Positive recurrent, aperiodic* states are *ergodic*

Markov Chain Monte Carlo

- How do we estimate *P(X)*, e.g., *P(X|e)* ?
- Generate samples that form Markov Chain with stationary distribution π=*P(X|e)*
- Estimate π from samples (observed states):

visited states x^0,…,x^n can be viewed as “samples” from distribution π

$$\bar{\pi}(x) = \frac{1}{T}\sum_{t=1}^{T}\delta(x, x^t)$$

$$\pi = \lim_{T\to\infty}\bar{\pi}(x)$$

MCMC Summary

- Convergence is guaranteed in the limit
- Initial state is not important, but… typically, we throw away first K samples - “**burn-in**”
- Samples are dependent, not i.i.d.
- Convergence (*mixing rate*) may be slow
- The stronger correlation between states, the slower convergence!

Gibbs Sampling (Geman&Geman,1984)

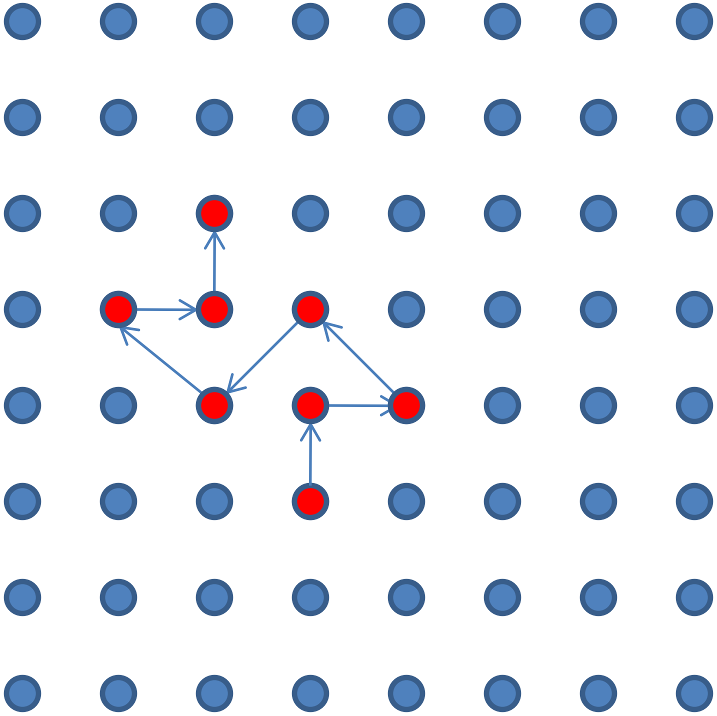
- Gibbs sampler is an algorithm to generate a sequence of samples from the **joint probability distribution** of two or more random variables
- Sample new variable value one variable at a time from the variable’s conditional distribution:

$$P(X_i) = P(X_i \mid x_1^t,.., x_{i-1}^t, x_{i+1}^t,..., x_n^t\} = P(X_i \mid x^t \setminus x_i)$$

- Samples form a Markov chain with stationary distribution *P(X|e)*

Gibbs Sampling: Illustration

The process of Gibbs sampling can be understood as a *random walk* in the space of all instantiations of X=x (remember drunkard's walk):



In one step we can reach instantiations that differ from current one by value assignment to at most one variable (assume randomized choice of variables X_i).

Ordered Gibbs Sampler

Generate sample x^{t+1} from x^t :

Process
All
Variables
In Some
Order

$$X_1 = x_1^{t+1} \leftarrow P(X_1 \mid x_2^t, x_3^t, \ldots, x_N^t, e)$$

$$X_2 = x_2^{t+1} \leftarrow P(X_2 \mid x_1^{t+1}, x_3^t, \ldots, x_N^t, e)$$

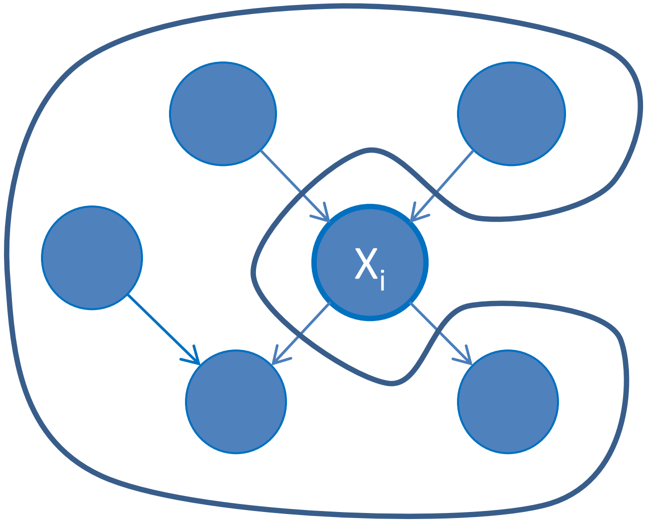
$$\ldots$$

$$X_N = x_N^{t+1} \leftarrow P(X_N \mid x_1^{t+1}, x_2^{t+1}, \ldots, x_{N-1}^{t+1}, e)$$

In short, for i=1 to N:

$$X_i = x_i^{t+1} \leftarrow \text{sampled from } P(X_i \mid x^t \setminus x_i, e)$$

Transition Probabilities in BN



Given *Markov blanket* (parents, children, and their parents), X_i is independent of all other nodes

Markov blanket:

$$markov(X_i) = pa_i \cup ch_i \cup \left(\bigcup_{X_j \in ch_j} pa_j \right)$$

$$P(X_i \mid x^t \setminus x_i) = P(X_i \mid markov_i^t):$$

$$P(x_i \mid x^t \setminus x_i) \propto P(x_i \mid pa_i) \prod_{X_j \in ch_i} P(x_j \mid pa_j)$$

Computation is linear in the size of Markov blanket!

Ordered Gibbs Sampling Algorithm (Pearl,1988)

Input: *X, E=e*

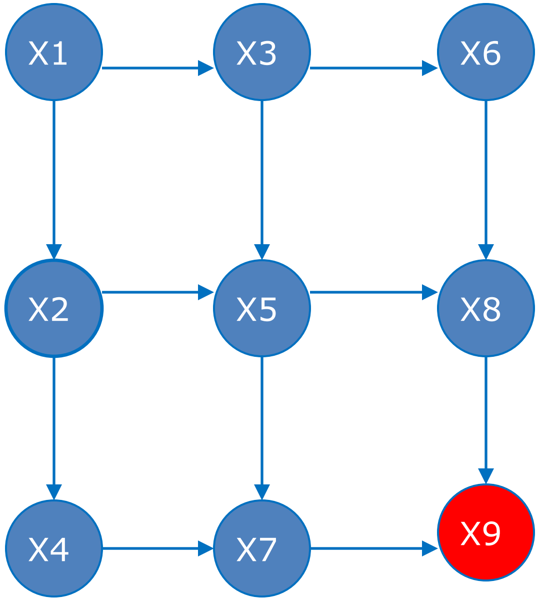
Output: *T* samples $\{x^t\}$

Fix evidence E=e, initialize x^0 at random

```
For t = 1 to T (compute samples)
    For i = 1 to N (loop through variables)
        $x_i^{t+1} \leftarrow P(X_i \mid markov_i^t)$
    End For
End For
```

Gibbs Sampling Example - BN

$$\boxed{X = \{X_1, X_2, \ldots, X_9\}, E = \{X_9\}}$$



$X_1 = x_1^0$

$X_6 = x_6^0$

$X_2 = x_2^0$

$X_7 = x_7^0$

$X_3 = x_3^0$

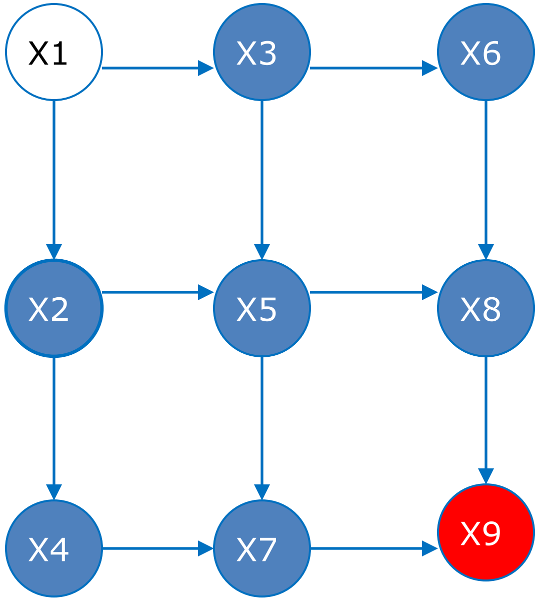
$X_8 = x_8^0$

$X_4 = x_4^0$

$X_5 = x_5^0$

Gibbs Sampling Example - BN

$$X = \{X_1, X_2, \ldots, X_9\}, E = \{X_9\}$$



$$x_1^1 \leftarrow P(X_1 \mid x_2^0, \ldots, x_8^0, x_9)$$

$$x_2^1 \leftarrow P(X_2 \mid x_1^1, \ldots, x_8^0, x_9)$$

$$\ldots$$

Answering Queries *P(x$_i$ | e) = ?*

- **Method 1**: count # of samples where $X_i = x_i$ (*histogram estimator*):

Dirac delta f-n

$$\overline{P}(X_i = x_i) = \frac{1}{T}\sum_{t=1}^{T}\delta(x_i, x^t)$$

- **Method 2**: average probability (*mixture estimator*):

$$\overline{P}(X_i = x_i) = \frac{1}{T}\sum_{t=1}^{T}P(X_i = x_i | markov_i^t)$$

- Mixture estimator converges faster (consider estimates for the unobserved values of X$_i$; prove via Rao-Blackwell theorem)

Rao-Blackwell Theorem

Rao-Blackwell Theorem: Let random variable set X be composed of two groups of variables, R and L. Then, for the joint distribution π(R,L) and function g, the following result applies

$$Var[E\{g(R) \mid L\} \leq Var[g(R)]$$

for a function of interest g, e.g., the mean or covariance (*Casella&Robert,1996, Liu et. al. 1995*).

- theorem makes a weak promise, but works well in practice!
- improvement depends the choice of R and L

Importance vs. Gibbs

Gibbs:

$$x^t \leftarrow \hat{P}(X \mid e)$$

$$\hat{P}(X \mid e) \xrightarrow{T \to \infty} P(X \mid e)$$

$$\hat{g}(X) = \frac{1}{T}\sum_{t=1}^{T} g(x^t)$$

Importance:

$$X^t \leftarrow Q(X \mid e)$$

$$\bar{g} = \frac{1}{T}\sum_{t=1}^{T} \frac{g(x^t)P(x^t)}{Q(x^t)}$$

w_t

Gibbs Sampling: Convergence

- Sample from $\overline{P}(X|e) \rightarrow P(X|e)$
- Converges iff chain is irreducible and ergodic
- Intuition - must be able to explore all states:
  - if X_i and X_j are strongly correlated, $X_i=0 \leftrightarrow X_j=0$, then, we cannot explore states with $X_i=1$ and $X_j=1$
- All conditions are satisfied when all probabilities are positive
- Convergence rate can be characterized by the second eigen-value of transition matrix

Gibbs: Speeding Convergence

Reduce dependence between samples (autocorrelation)

- Skip samples
- Randomize Variable Sampling Order
- Employ blocking (grouping)
- Multiple chains

Reduce variance (cover in the next section)

Blocking Gibbs Sampler

- Sample several variables **together, as a block**
- **Example:** Given three variables X,Y,Z, with domains of size 2, group Y and Z together to form a variable $W=\{Y,Z\}$ with domain size 4. Then, given sample (x^t,y^t,z^t), compute next sample:

$$x^{t+1} \leftarrow P(X \mid y^t, z^t) = P(w^t)$$

$$(y^{t+1}, z^{t+1}) = w^{t+1} \leftarrow P(Y, Z \mid x^{t+1})$$

+ Can improve convergence greatly when two variables are strongly correlated!

- Domain of the block variable grows exponentially with the #variables in a block!

Gibbs: Multiple Chains

- Generate M chains of size K
- Each chain produces independent estimate P_m:

$$\overline{P}_m(x_i \mid e) = \frac{1}{K}\sum_{t=1}^{K} P(x_i \mid x^t \setminus x_i)$$

- Estimate $P(x_i|e)$ as average of $P_m(x_i|e)$:

$$\hat{P}(\bullet) = \frac{1}{M}\sum_{i=1}^{M} P_m(\bullet)$$

Treat P_m as independent random variables.

Gibbs Sampling Summary

- Markov Chain Monte Carlo method

(Gelfand and Smith, 1990, Smith and Roberts, 1993, Tierney, 1994)

- Samples are **dependent**, form Markov Chain
- Sample from $\overline{P}(X \mid e)$ which **converges** to $\overline{P}(X \mid e)$
- Guaranteed to converge when all *P > 0*
- Methods to improve convergence:
  - Blocking
  - Rao-Blackwellised

Overview

1. Probabilistic Reasoning/Graphical models
2. Importance Sampling
3. Markov Chain Monte Carlo: Gibbs Sampling
4. Sampling in presence of Determinism
5. Rao-Blackwellisation
6. AND/OR importance sampling

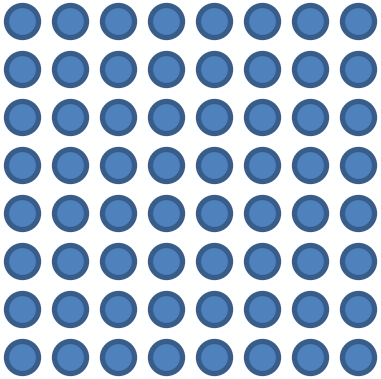
Sampling: Performance

- Gibbs sampling
  - Reduce dependence between samples
- Importance sampling
  - Reduce variance
- Achieve both by **sampling a subset of variables** and integrating out the rest (reduce dimensionality), aka **Rao-Blackwellisation**
- Exploit graph structure to manage the extra cost

Smaller Subset State-Space

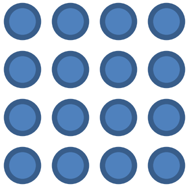
- Smaller state-space is easier to cover

$X = \{X_1, X_2, X_3, X_4\}$



$D(X) = 64$

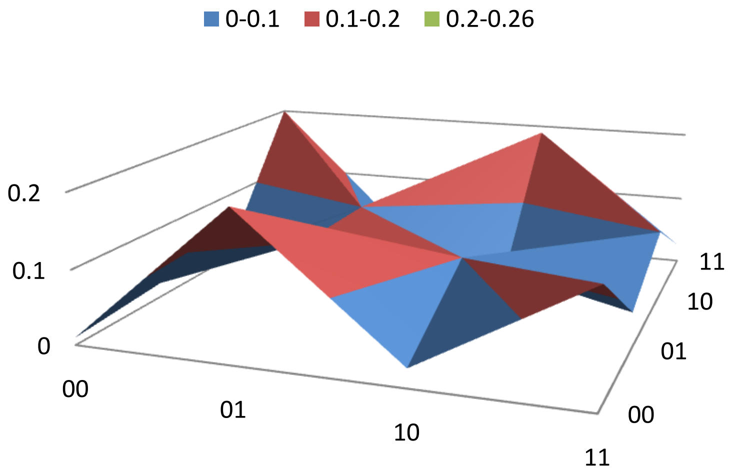
$X = \{X_1, X_2\}$



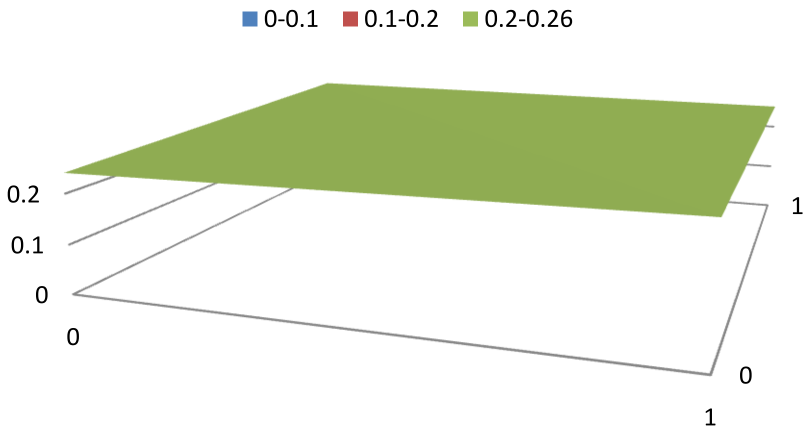
$D(X) = 16$

Smoother Distribution

$P(X_1,X_2,X_3,X_4)$



$P(X_1,X_2)$



Speeding Up Convergence

- Mean Squared Error of the estimator:

$$MSE_Q\left[\overline{P}\right] = BIAS^2 + Var_Q\left[\overline{P}\right]$$

- In case of unbiased estimator, BIAS=0

$$MSE_Q[\hat{P}] = Var_Q[\hat{P}] = \left(E_Q\left[\hat{P}\right]^2 - E_Q[P]^2 \right)$$

- Reduce variance $\Rightarrow$ speed up convergence !

Rao-Blackwellisation

$$X = R \bigcup L$$

$$\hat{g}(x) = \frac{1}{T}\{h(x^1) + \cdots + h(x^T)\}$$

$$\tilde{g}(x) = \frac{1}{T}\{E[h(x) \mid l^1] + \cdots + E[h(x) \mid l^T]\}$$

$$Var\{g(x)\} = Var\{E[g(x) \mid l]\} + E\{\mathrm{var}[g(x) \mid l]\}$$

$$Var\{g(x)\} \geq Var\{E[g(x) \mid l]\}$$

$$Var\{\hat{g}(x)\} = \frac{Var\{h(x)\}}{T} \geq \frac{Var\{E[h(x) \mid l]\}}{T} = Var\{\tilde{g}(x)\}$$

Liu, Ch.2.3

Rao-Blackwellisation

"Carry out analytical computation as much as possible" - Liu

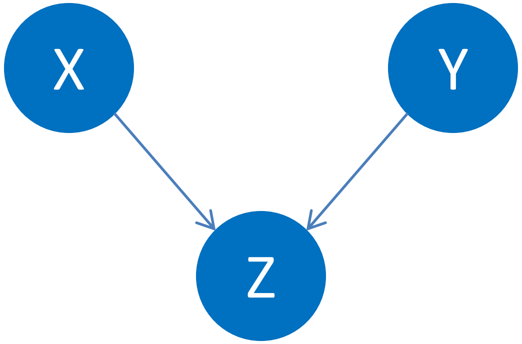
- X=R∪L
- Importance Sampling:

$$Var_Q\{\frac{P(R,L)}{Q(R,L)}\} \geq Var_Q\{\frac{P(R)}{Q(R)}\}$$

Liu, Ch.2.5.5

- Gibbs Sampling:
  - autocovariances are lower (less correlation between samples)
  - if X_i and X_j are strongly correlated, $X_i=0 \leftrightarrow X_j=0$, only include one fo them into a sampling set

Blocking Gibbs Sampler vs. Collapsed



Faster Convergence

- Standard Gibbs:

$$P(x \mid y, z), P(y \mid x, z), P(z \mid x, y) \tag{1}$$

- Blocking:

$$P(x \mid y, z), P(y, z \mid x) \tag{2}$$

- Collapsed:

$$P(x \mid y), P(y \mid x) \tag{3}$$

Collapsed Gibbs Sampling

Generating Samples

Generate sample c^{t+1} from c^t :

$$C_1 = c_1^{t+1} \leftarrow P(c_1 \mid c_2^t, c_3^t, ..., c_K^t, e)$$

$$C_2 = c_2^{t+1} \leftarrow P(c_2 \mid c_1^{t+1}, c_3^t, ..., c_K^t, e)$$

...

$$C_K = c_K^{t+1} \leftarrow P(c_K \mid c_1^{t+1}, c_2^{t+1}, ..., c_{K-1}^{t+1}, e)$$

In short, for i=1 to K:

$$C_i = c_i^{t+1} \leftarrow \textbf{sampled from } P(c_i \mid c^t \setminus c_i , e)$$

Collapsed Gibbs Sampler

Input: $C \subset X$, *E=e*

Output: *T* samples $\{c^t\}$

Fix evidence E=e, initialize c^0 at random

```
1.  For t = 1 to T (compute samples)
2.      For i = 1 to N (loop through variables)
3.          $c_i^{t+1} \leftarrow P(C_i \mid c^t \backslash c_i)$
4.      End For
5.  End For
```

Calculation Time

- Computing $P(c_i| c^t \backslash c_i, e)$ is more expensive (requires inference)
- Trading #samples for smaller variance:
  - – generate more samples with higher covariance
  - – generate fewer samples with lower covariance
- Must control the time spent computing sampling probabilities in order to be time-effective!

Exploiting Graph Properties

Recall… computation time is *exponential in the adjusted induced width* of a graph

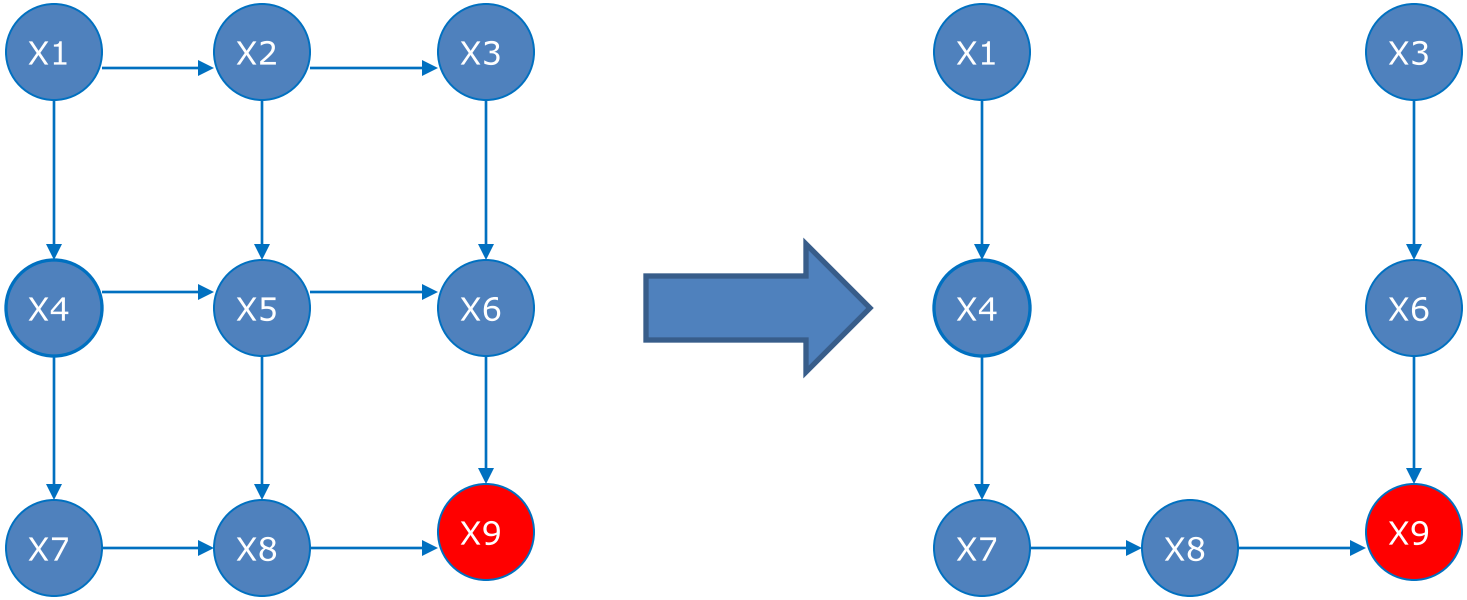
- **w-cutset** is a subset of variable s.t. when they are observed, induced width of the graph is w
- when sampled variables form a **w-cutset** , inference is exp(w) (e.g., using *Bucket Tree Elimination*)
- **cycle-cutset** is a special case of w-cutset

Sampling w-cutset $\Rightarrow$ **w-cutset sampling!**

What If C=Cycle-Cutset ?

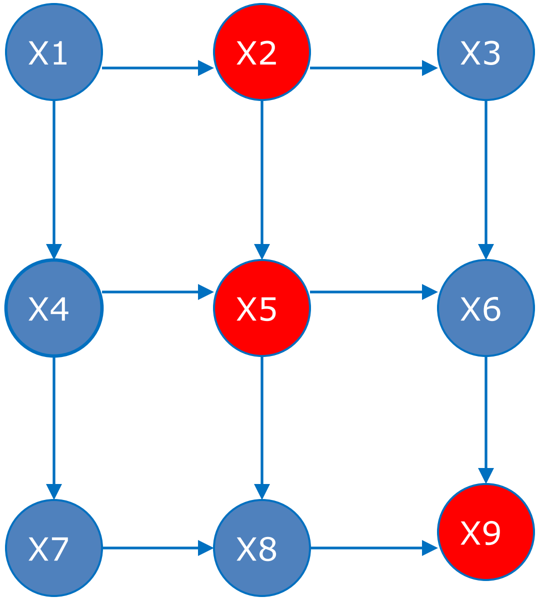
$$\boxed{c^0 = \{x_2^0, x_5^0\}, E = \{X_9\}}$$

$P(x_2,x_5,x_9)$ – can compute using Bucket Elimination



$P(x_2,x_5,x_9)$ – computation complexity is O(N)

Computing Transition Probabilities



Compute joint probabilities:

$$BE : P(x_2 = 0, x_3, x_9)$$

$$BE : P(x_2 = 1, x_3, x_9)$$

Normalize:

$$\alpha = P(x_2 = 0, x_3, x_9) + P(x_2 = 1, x_3, x_9)$$

$$P(x_2 = 0 \mid x_3) = \alpha P(x_2 = 0, x_3, x_9)$$

$$P(x_2 = 1 \mid x_3) = \alpha P(x_2 = 1, x_3, x_9)$$

Cutset Sampling-Answering Queries

- Query: $\forall c_i \in C$, $P(c_i \mid e)$=? same as Gibbs:

$$\hat{P}(c_i|e) = \frac{1}{T}\sum_{t=1}^{T} P(c_i \mid c^t \setminus c_i, e)$$

computed while generating sample t using bucket tree elimination

- Query: $\forall x_i \in X \setminus C$, $P(x_i \mid e)$=?

$$\overline{P}(x_i|e) = \frac{1}{T}\sum_{t=1}^{T} P(x_i \mid c^t, e)$$

compute after generating sample t using bucket tree elimination

Cutset Sampling vs. Cutset Conditioning

- Cutset Conditioning

$$P(x_i|e) = \sum_{c \in D(C)} P(x_i \mid c,e) \times \boxed{P(c \mid e)}$$

- Cutset Sampling

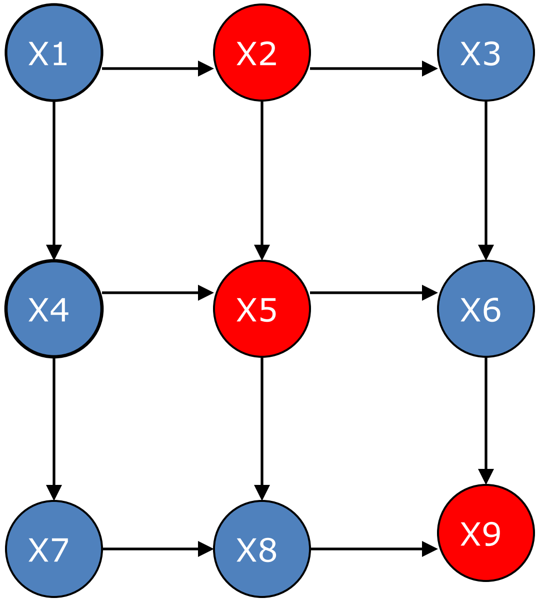
$$\overline{P}(x_i|e) = \frac{1}{T} \sum_{t=1}^{T} P(x_i \mid c^t, e)$$

$$= \sum_{c \in D(C)} P(x_i \mid c,e) \times \frac{count(c)}{T}$$

$$= \sum_{c \in D(C)} P(x_i \mid c,e) \times \boxed{\overline{P}(c \mid e)}$$

Cutset Sampling Example

Estimating $P(x_2|e)$ for sampling node X_2 :



$$x_2^1 \leftarrow P(x_2 | x_5^0, x_9)$$ Sample 1

$$\ldots$$

$$x_2^2 \leftarrow P(x_2 | x_5^1, x_9)$$ Sample 2

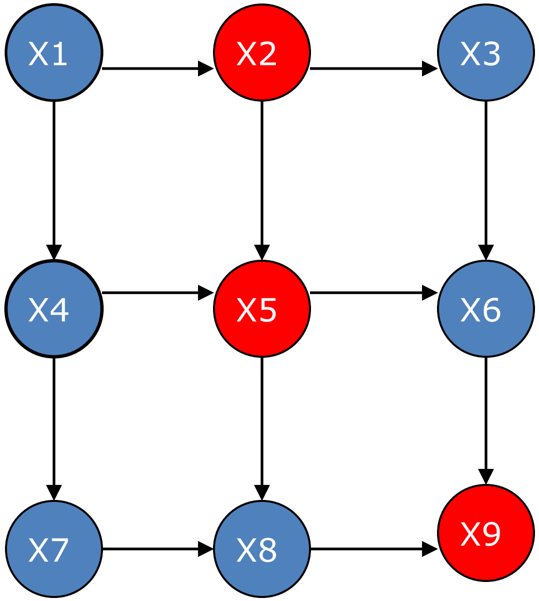
$$\ldots$$

$$x_2^3 \leftarrow P(x_2 | x_5^2, x_9)$$ Sample 3

$$\overline{P}(x_2 | x_9) = \frac{1}{3}\begin{bmatrix} P(x_2 | x_5^0, x_9) \\ + P(x_2 | x_5^1, x_9) \\ + P(x_2 | x_5^2, x_9) \end{bmatrix}$$

Cutset Sampling Example

Estimating $P(x_3 | e)$ for non-sampled node X_3 :



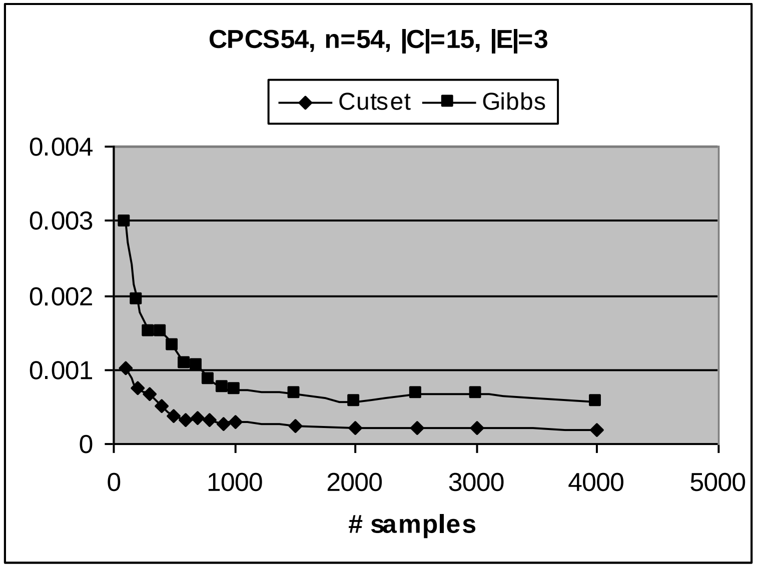
$$c^1 = \{x_2^1, x_5^1\} \Rightarrow P(x_3 | x_2^1, x_5^1, x_9)$$

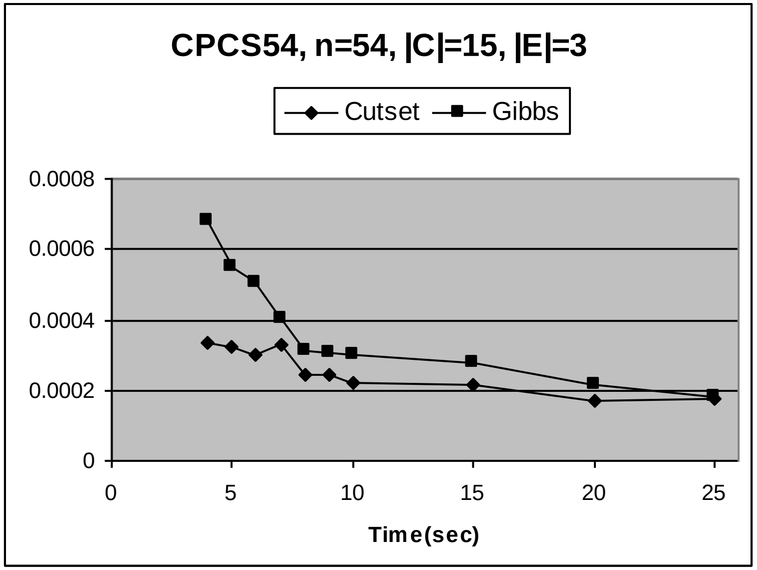
$$c^2 = \{x_2^2, x_5^2\} \Rightarrow P(x_3 | x_2^2, x_5^2, x_9)$$

$$c^3 = \{x_2^3, x_5^3\} \Rightarrow P(x_3 | x_2^3, x_5^3, x_9)$$

$$\boxed{P(x_3 | x_9) = \frac{1}{3}\left[\begin{array}{l} P(x_3 | x_2^1, x_5^1, x_9) \\ + P(x_3 | x_2^2, x_5^2, x_9) \\ + P(x_3 | x_2^3, x_5^3, x_9) \end{array}\right]}$$

CPCS54 Test Results



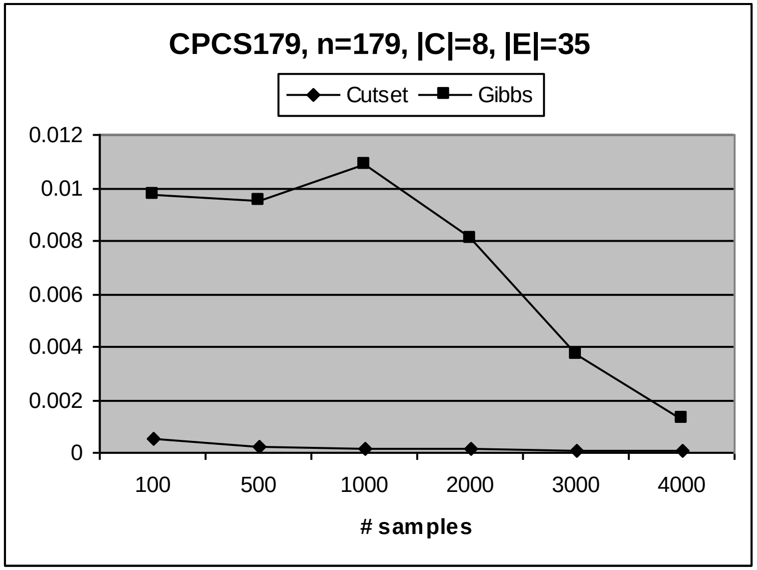


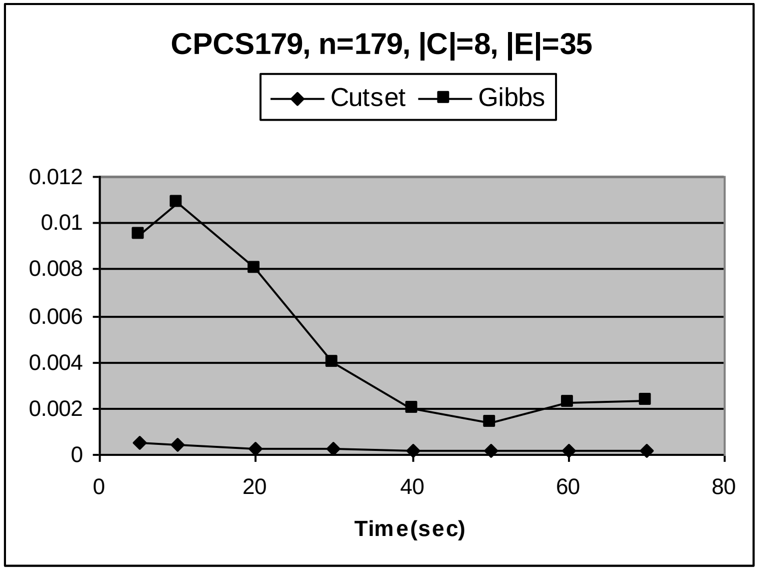
MSE vs. #samples (left) and time (right)

Ergodic, |X|=54, D(X_i)=2, |C|=15, |E|=3

Exact Time = 30 sec using Cutset Conditioning

CPCS179 Test Results

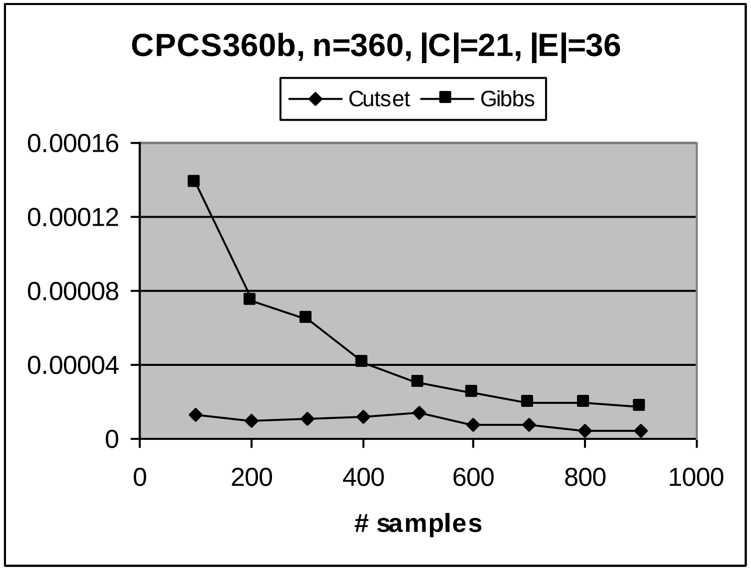


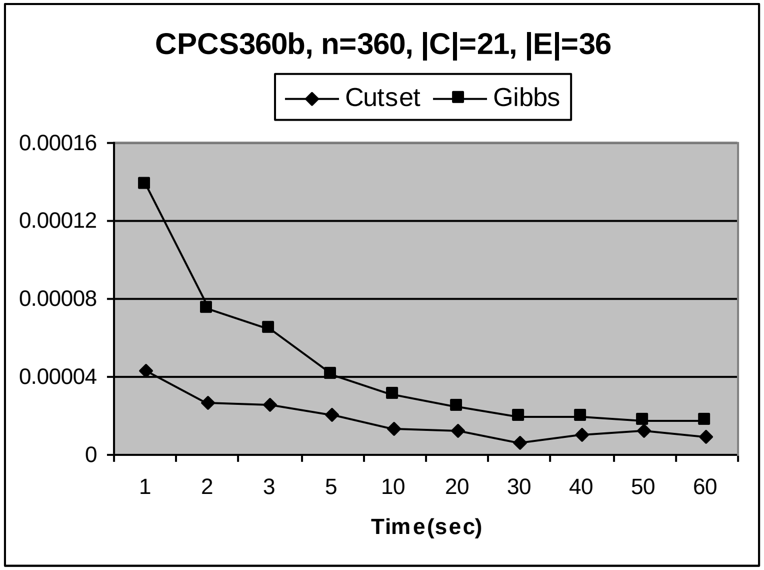


MSE vs. #samples (left) and time (right)
Non-Ergodic (1 deterministic CPT entry)
|X| = 179, |C| = 8, 2<= $D(X_i)$<=4, |E| = 35

Exact Time = 122 sec using Cutset Conditioning

CPCS360b Test Results





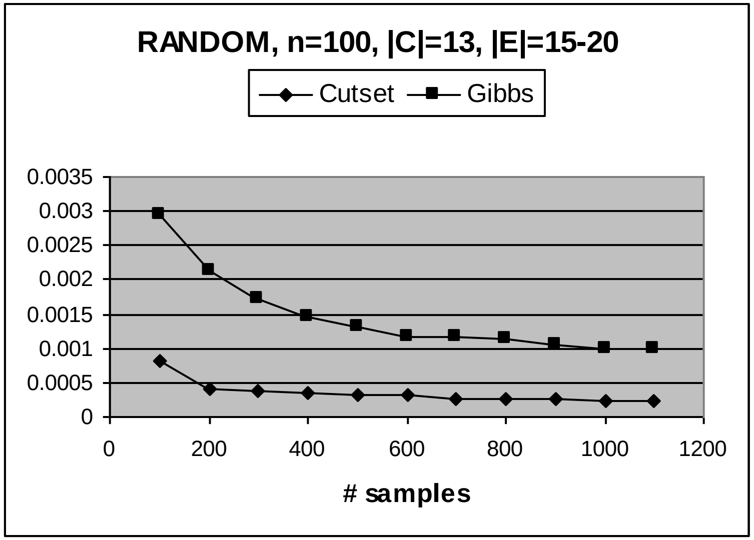
MSE vs. #samples (left) and time (right)

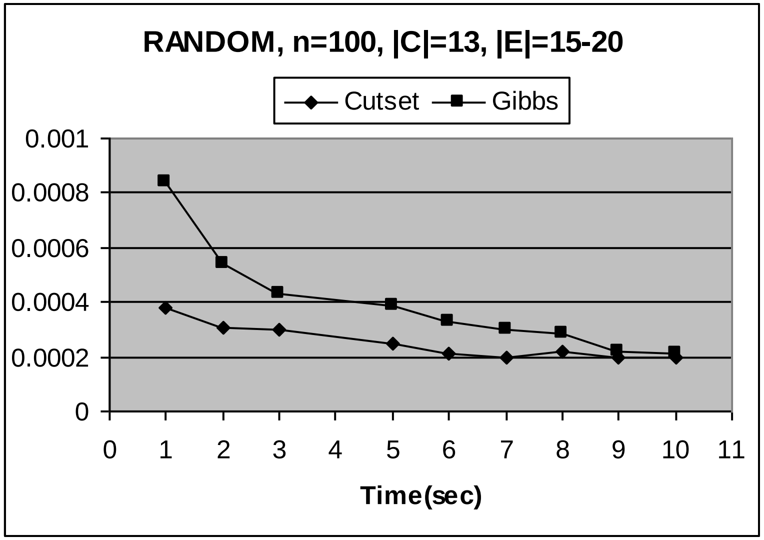
Ergodic, |X| = 360, $D(X_i)$=2, |C| = 21, |E| = 36

Exact Time > 60 min using Cutset Conditioning

Exact Values obtained via Bucket Elimination

Random Networks





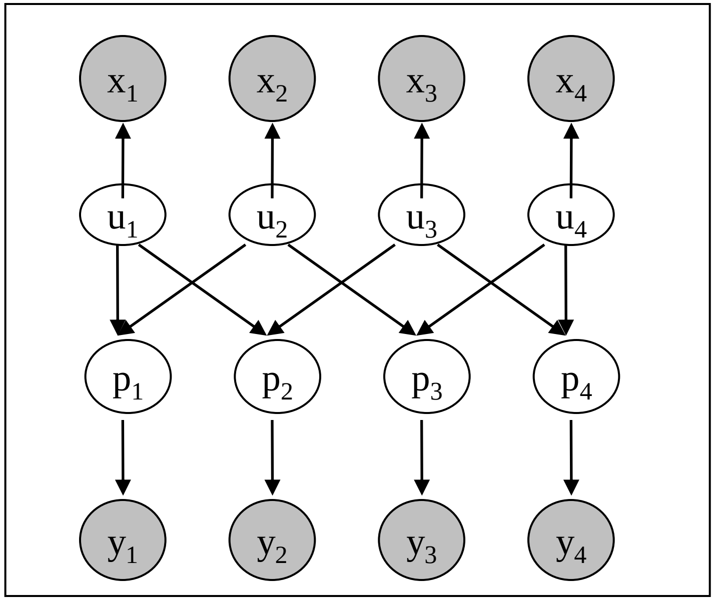
MSE vs. #samples (left) and time (right)

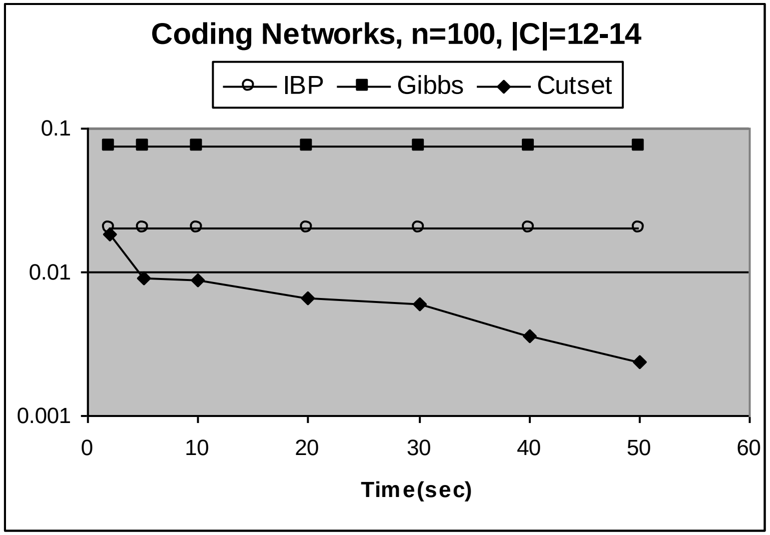
|X| = 100, D(X_i) =2,|C| = 13, |E| = 15-20

Exact Time = 30 sec using Cutset Conditioning

Coding Networks

Cutset Transforms Non-Ergodic Chain to Ergodic





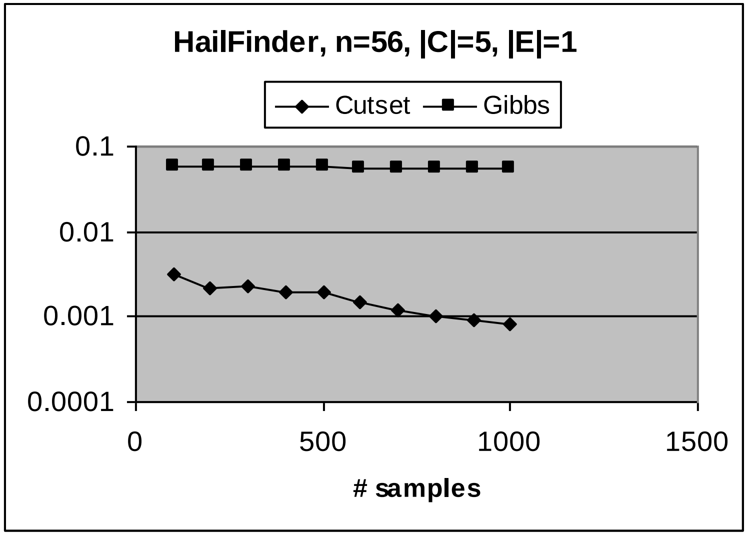
MSE vs. time (right)

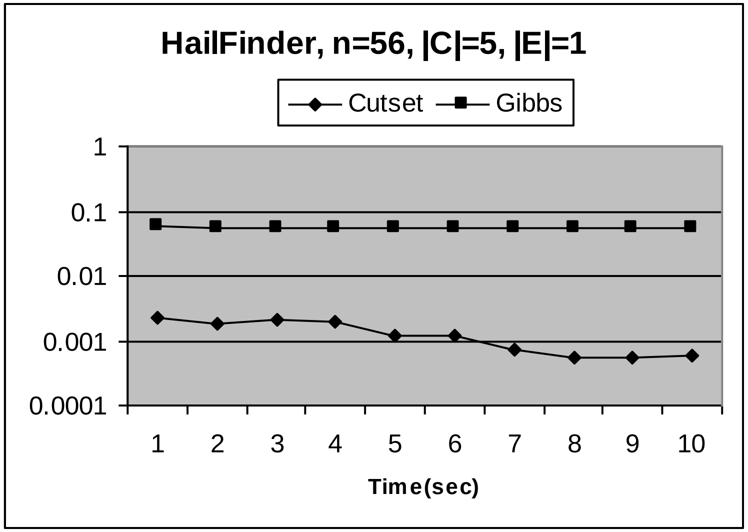
Non-Ergodic, |X| = 100, D(X_i)=2, |C| = 13-16, |E| = 50

Sample Ergodic Subspace U={U_1, U_2,…U_k}

Exact Time = 50 sec using Cutset Conditioning

Non-Ergodic Hailfinder



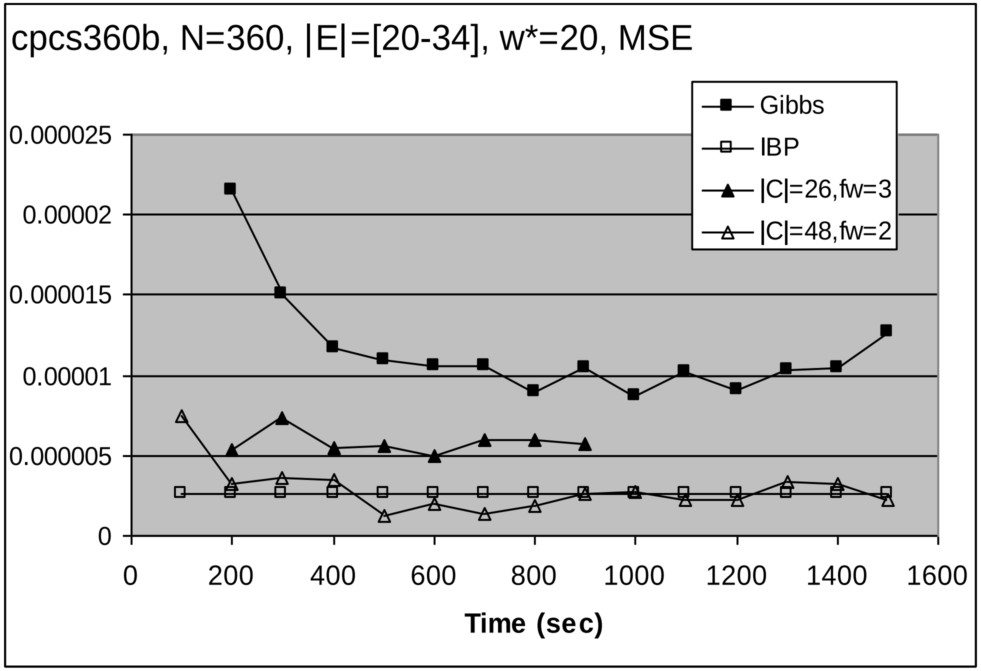


MSE vs. #samples (left) and time (right)

Non-Ergodic, |X| = 56, |C| = 5, 2 <=D(X_i) <=11, |E| = 0

Exact Time = 2 sec using Loop-Cutset Conditioning

CPCS360b - MSE



MSE vs. Time

Ergodic, |X| = 360, |C| = 26, $D(X_i)=2$

Exact Time = 50 min using BTE

Cutset Importance Sampling

(Gogate & Dechter, 2005) and (Bidyuk & Dechter, 2006)

- Apply Importance Sampling over cutset C

$$\hat{P}(e) = \frac{1}{T}\sum_{t=1}^{T}\frac{P(c^t, e)}{Q(c^t)} = \frac{1}{T}\sum_{t=1}^{T} w^t$$

where *P(c^t,e)* is computed using Bucket Elimination

$$\overline{P}(c_i \mid e) = \alpha \frac{1}{T}\sum_{t=1}^{T}\delta(c_i, c^t) w^t$$

$$\overline{P}(x_i \mid e) = \alpha \frac{1}{T}\sum_{t=1}^{T} P(x_i \mid c^t, e) w^t$$

Likelihood Cutset Weighting (LCS)

- Z=Topological Order{C,E}
- Generating sample t+1:

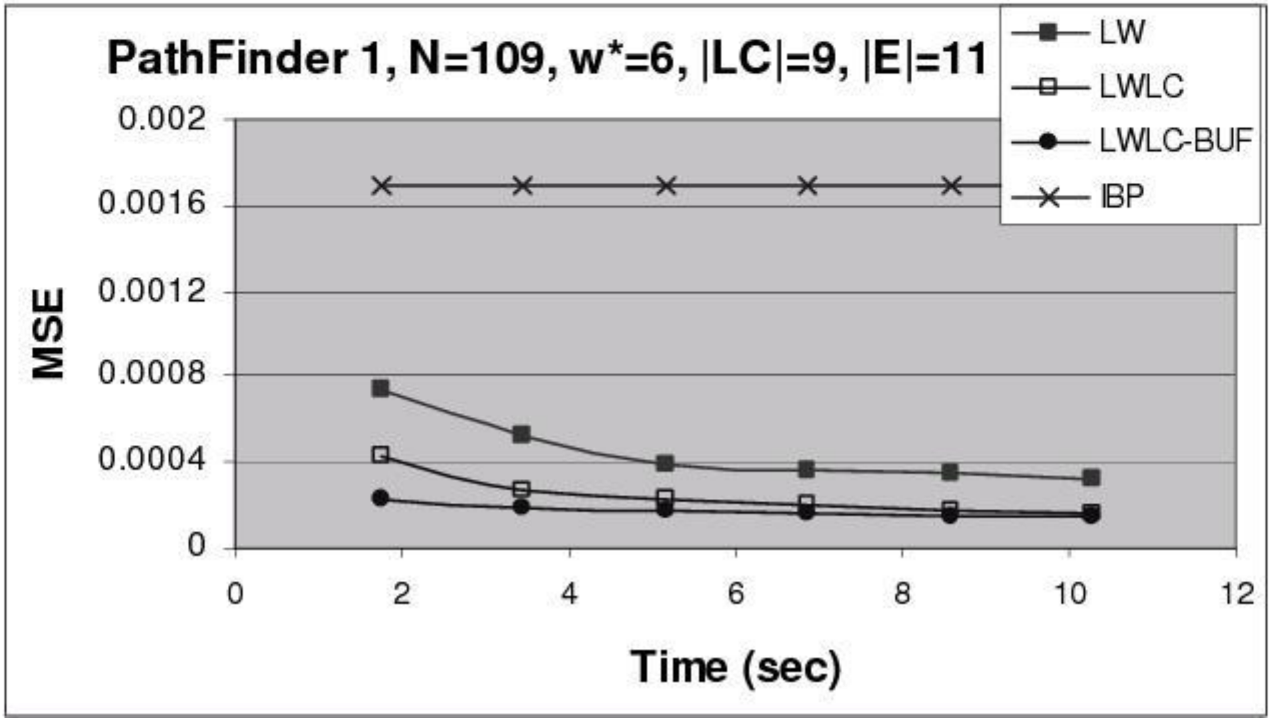
```
For $Z_i \in Z$ do :
    If $Z_i \in E$
        $z_i^{t+1} = z_i, z_i \in e$
    Else
        $z_i^{t+1} \leftarrow P(Z_i \mid z_1^{t+1}, \ldots, z_{i-1}^{t+1})$
    End If
End For
```

- computed while generating sample t using bucket tree elimination

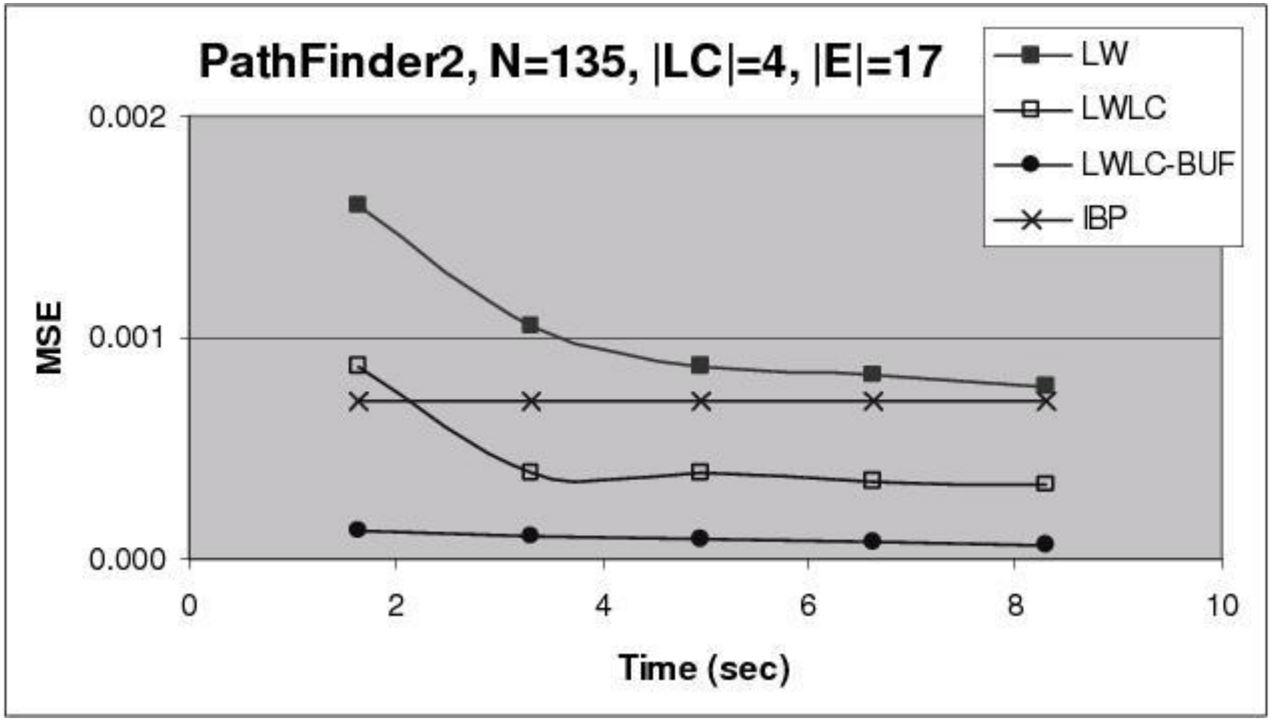
- can be memoized for some number of instances K (based on memory available

$KL[P(C|e), Q(C)] \leq KL[P(X|e), Q(X)]$

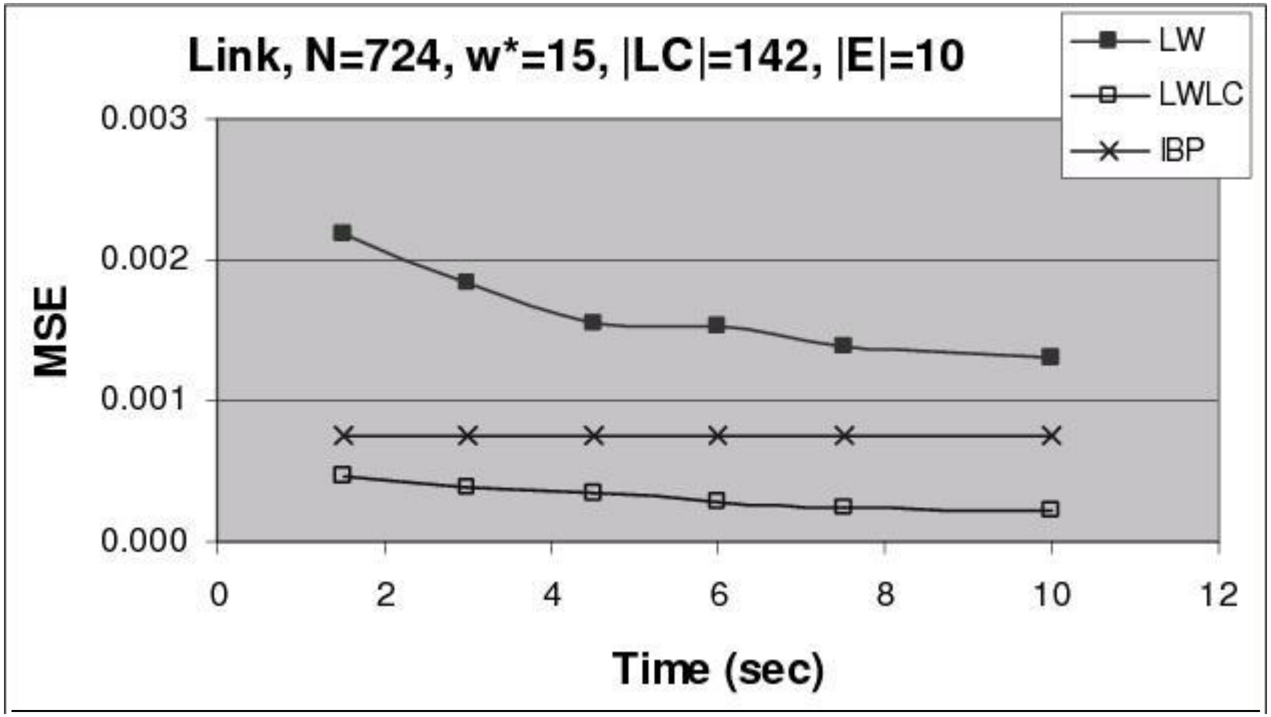
Pathfinder 1



Pathfinder 2



Link


Link, N=724, w*=15, |LC|=142, |E|=10
LW
LWLC
IBP
MSE
0.003
0.002
0.001
0.000
0
2
4
6
8
10
12
Time (sec)

Summary

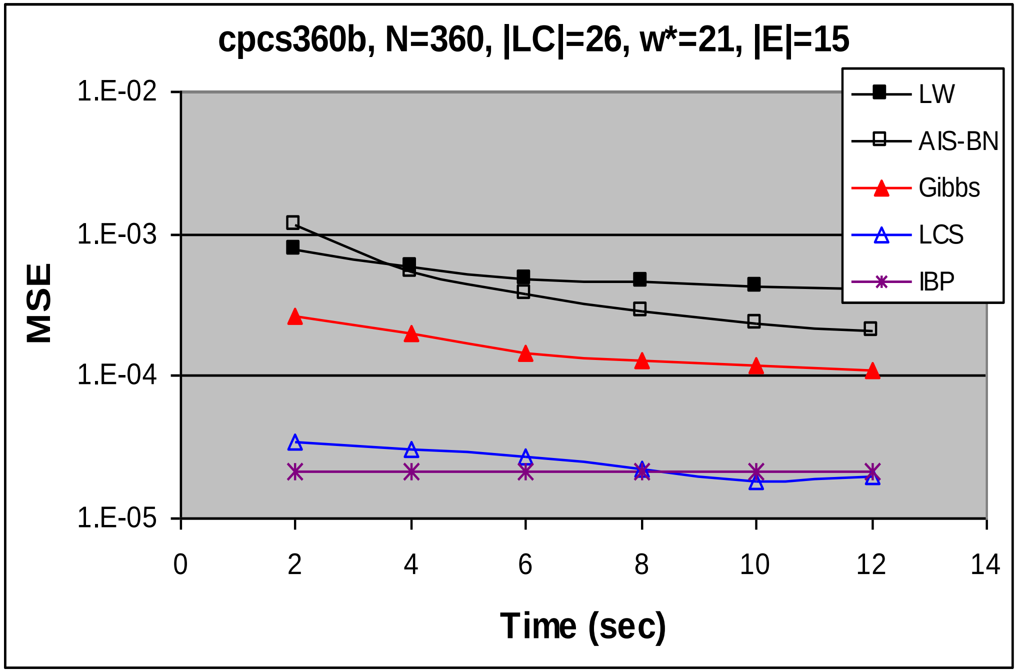
Importance Sampling

- i.i.d. samples
- Unbiased estimator
- Generates samples fast
- Samples from Q
- Reject samples with zero-weight
- Improves on cutset

Gibbs Sampling

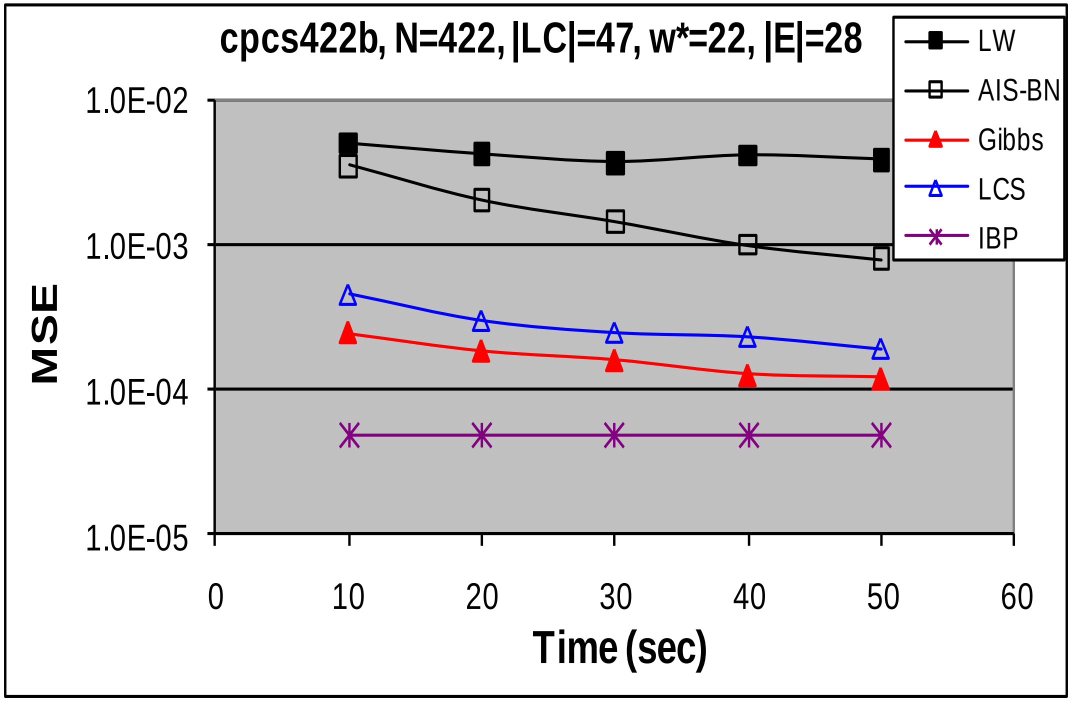
- Dependent samples
- Biased estimator
- Generates samples slower
- Samples from $\overline{P}(X|e)$
- Does not converge in presence of constraints
- Improves on cutset

CPCS360b



LW – likelihood weighting
LCS – likelihood weighting on a cutset

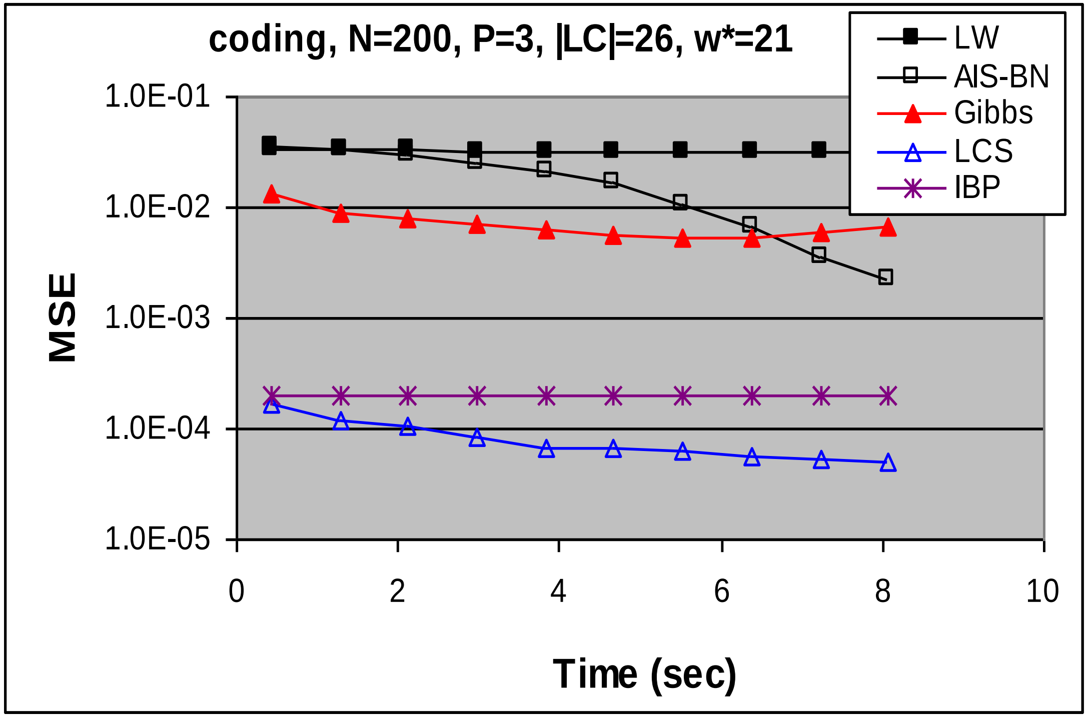
CPCS422b



LW – likelihood weighting

LCS – likelihood weighting on a cutset

Coding Networks



LW – likelihood weighting

LCS – likelihood weighting on a cutset