Sampling Techniques for Probabilistic and Deterministic Graphical models

ICS 276, Spring 2018

Bozhena Bidyuk

Rina Dechter

Reading" Darwiche chapter 15, related papers

Algorithms for Reasoning with graphical models

Slides Set 11(part b):

Sampling Techniques for Probabilistic and Deterministic Graphical models

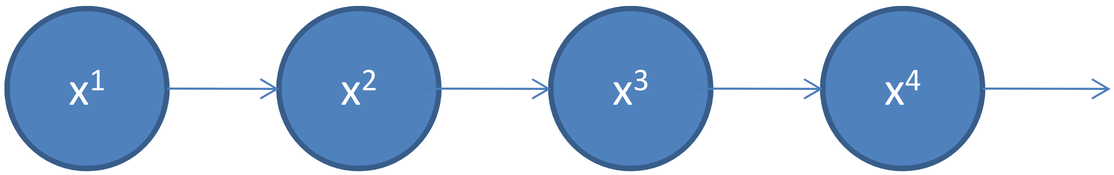
Rina Dechter

(Reading" Darwiche chapter 15, cutset-sampling paper posted)

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



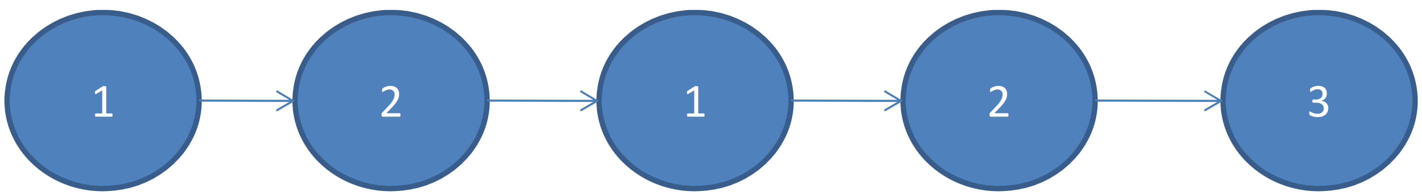
- A **Markov chain** is a discrete random process with the property that the next state depends only on the current state (**Markov Property**):

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

- 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

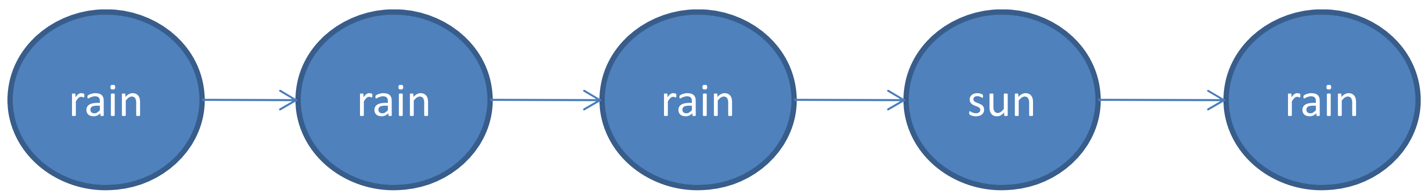


$$D(X) = \{0,1,2,\ldots\}$$

	$P(n-1)$	$P(n+1)$
n	0.5	0.5

transition matrix P(X)

Example: Weather Model



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

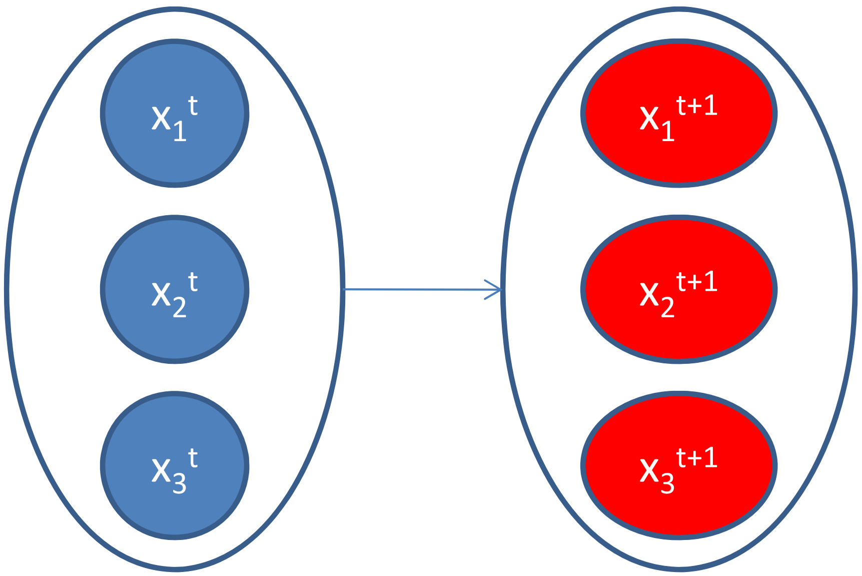
	$P(rainy)$	$P(sunny)$
$rainy$	0.9	0.1
$sunny$	0.5	0.5

transition matrix P(X)

Multi-Variable System

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

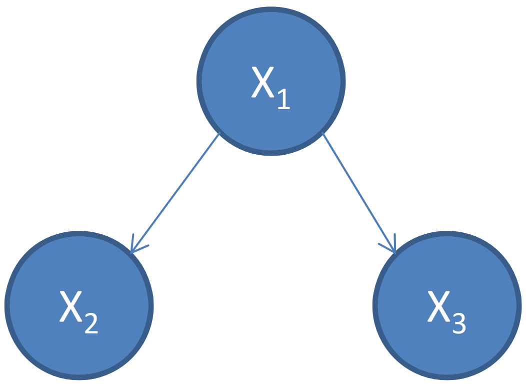
- state is an assignment of values to all the variables

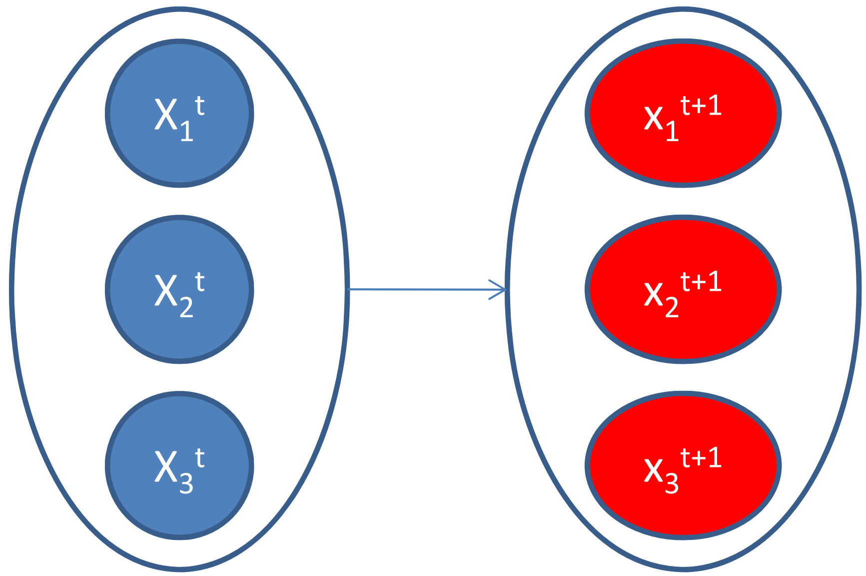


$$x^t = \{x_1^t, x_2^t, \ldots, x_n^t\}$$

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 $\pi = P(X|e)$
- Estimate π from samples (observed states):

visited states $x^0, \ldots, x^n$ can be viewed as "samples" from distribution π

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

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

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)

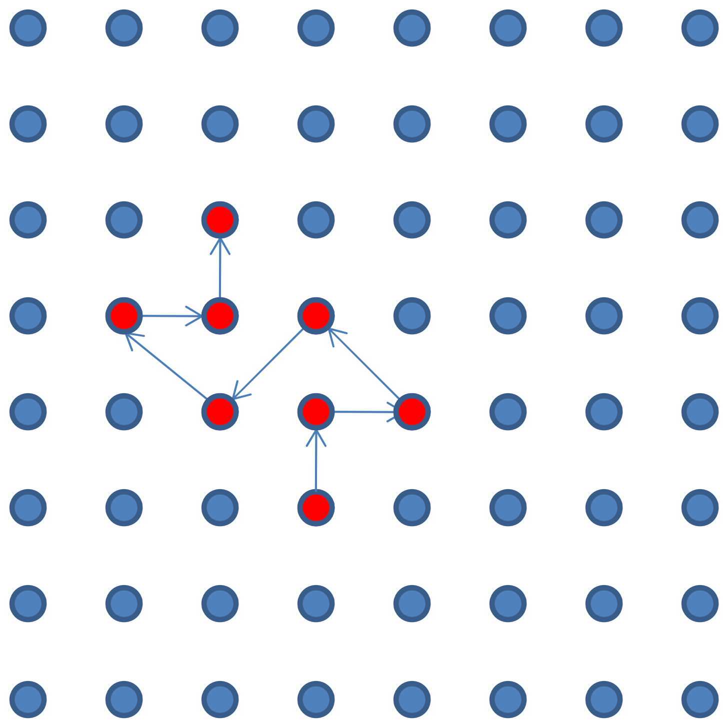
- Gibbs sampler is an algorithm to generate a sequence of samples from the **joint probability distribution** of two or more random variables
- Sample new variable value one variable at a time from the variable's conditional distribution:

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

- 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

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

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

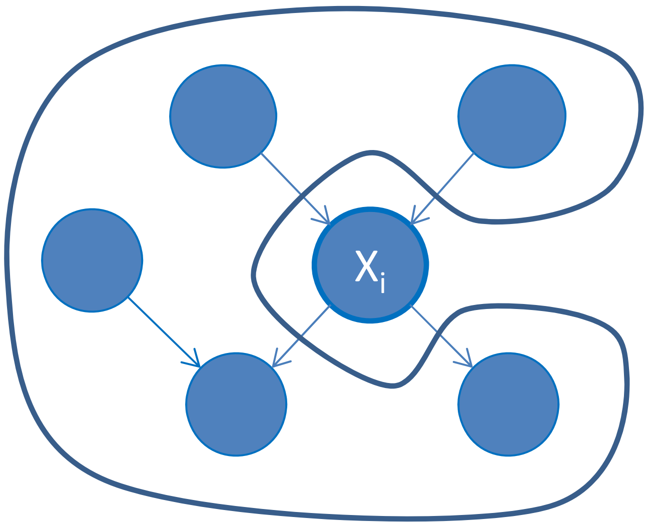
…

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

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:

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

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

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

Computation is linear in the size of Markov blanket!

Ordered Gibbs Sampling Algorithm (Pearl,1988)

Input: *X, E=e*

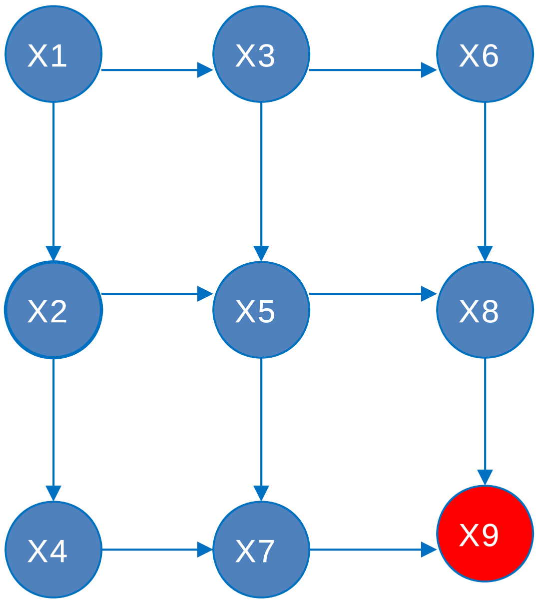
Output: *T* samples $\{x^t\}$

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

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

Gibbs Sampling Example - BN

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



$$X_1 = x_1^0$$

$$X_6 = x_6^0$$

$$X_2 = x_2^0$$

$$X_7 = x_7^0$$

$$X_3 = x_3^0$$

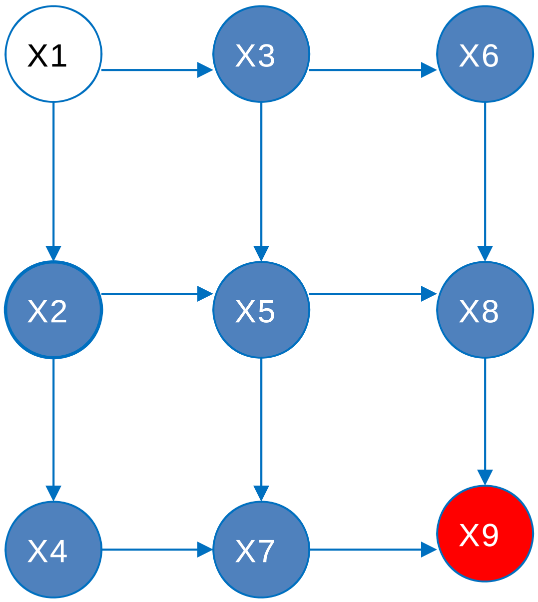
$$X_8 = x_8^0$$

$$X_4 = x_4^0$$

$$X_5 = x_5^0$$

Gibbs Sampling Example - BN

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



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

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

...

Answering Queries $P(x_i \mid e) = ?$

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

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

Dirac delta f-n

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

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

- 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

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

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 on 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 $\bar{P}(X|e) \to P(X|e)$
- 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

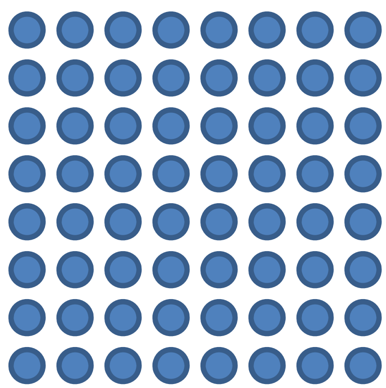
Sampling: Performance

- Gibbs sampling
  - Reduce dependence between samples
- Importance sampling
  - Reduce variance
- Cutset sampling 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

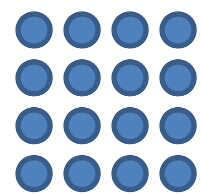
- Smaller state-space is easier to cover

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



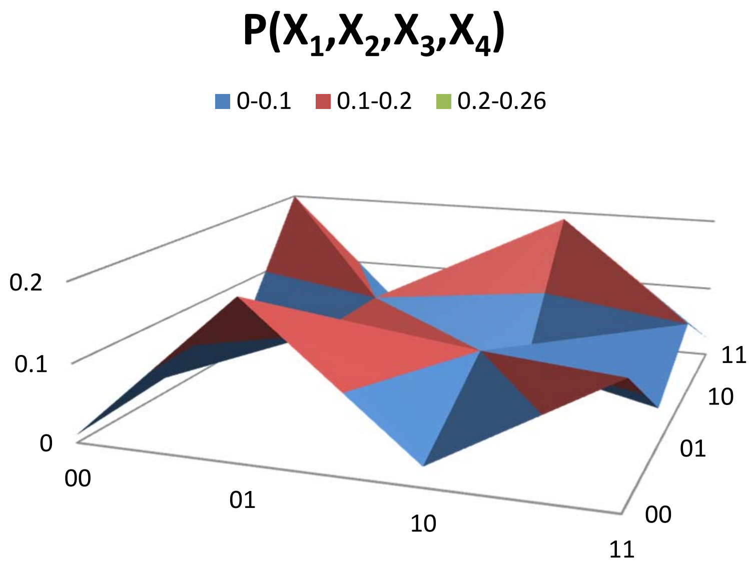
$D(X) = 64$

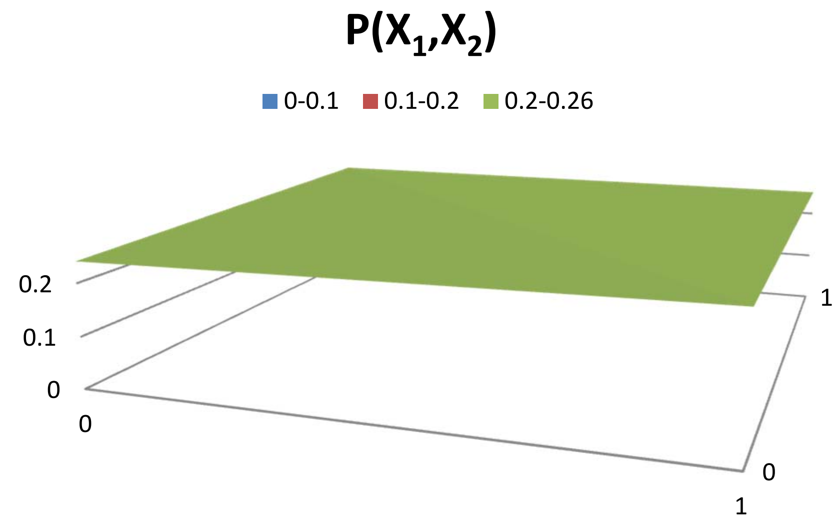
$X = \{X_1, X_2\}$



$D(X) = 16$

Smoother Distribution





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

$$X = R \cup L$$

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

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

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

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

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

Liu, Ch.2.3

Rao-Blackwellisation

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

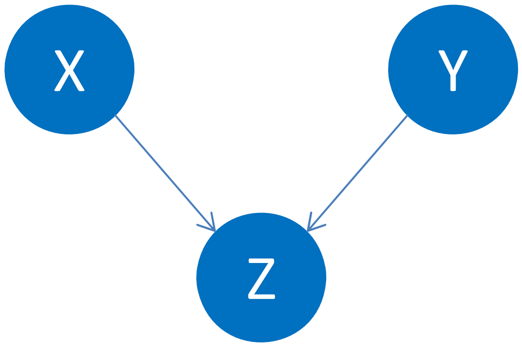
- $X = R \cup L$
- Importance Sampling:

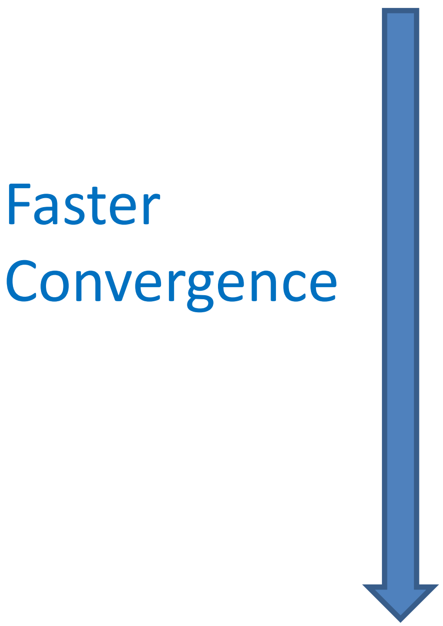
$$Var_Q\{\frac{P(R,L)}{Q(R,L)}\} \geq Var_Q\{\frac{P(R)}{Q(R)}\}$$

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 of them into a sampling set

Blocking Gibbs Sampler vs. Collapsed





- Standard Gibbs:

  $$P(x \mid y, z), P(y \mid x, z), P(z \mid x, y) \quad (1)$$

- Blocking:

  $$P(x \mid y, z), P(y, z \mid x) \quad (2)$$

- Collapsed:

  $$P(x \mid y), P(y \mid x) \quad (3)$$

Collapsed Gibbs Sampling

Generating Samples

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

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

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

...

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

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

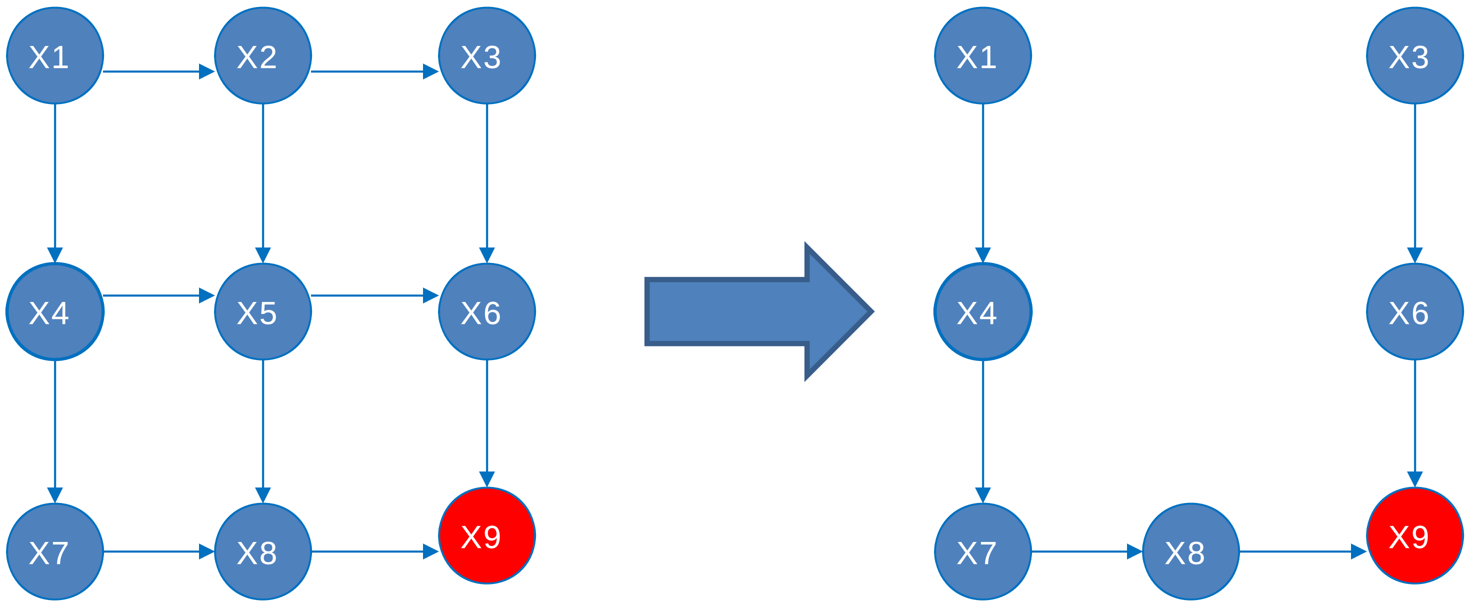
- **w-cutset** is a subset of variable s.t. when they are observed, induced width of the graph is w
- when sampled variables form a **w-cutset** , inference is exp(w) (e.g., using *Bucket Tree Elimination*)
- **cycle-cutset** is a special case of w-cutset

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

What If C=Cycle-Cutset ?

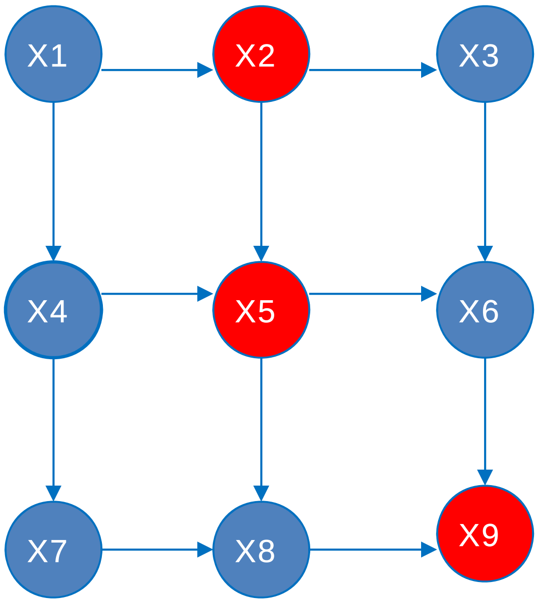
$$c^0 = \{x_2^0, x_5^0\}, E = \{X_9\}$$

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



$P(x_2,x_5,x_9)$ – 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

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

- Cutset Sampling

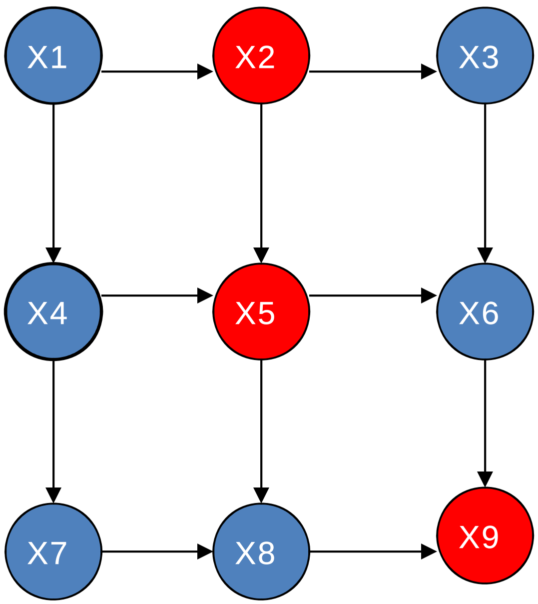
$$\overline{P}(x_i|e) = \frac{1}{T} \sum_{t=1}^{T} P(x_i \mid c^t, e)$$

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

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

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

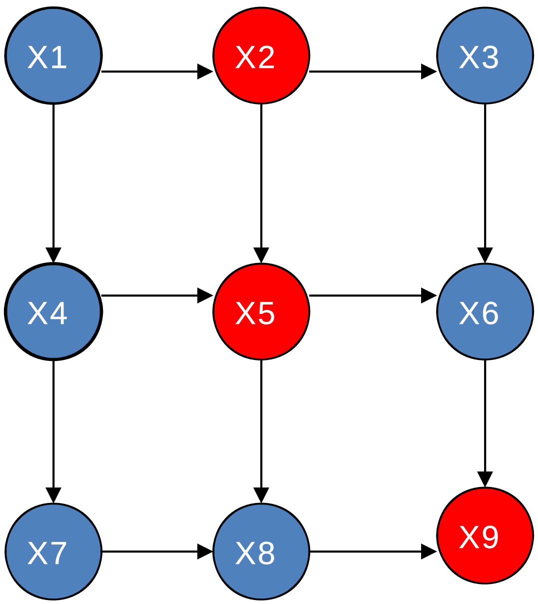
$$\ldots$$

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

$$\overline{P}(x_2 \mid x_9) = \frac{1}{3}\left[\begin{array}{l} P(x_2 | x_5^0, x_9) \\ + P(x_2 | x_5^1, x_9) \\ + P(x_2 | x_5^2, x_9) \end{array}\right]$$

Cutset Sampling Example

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



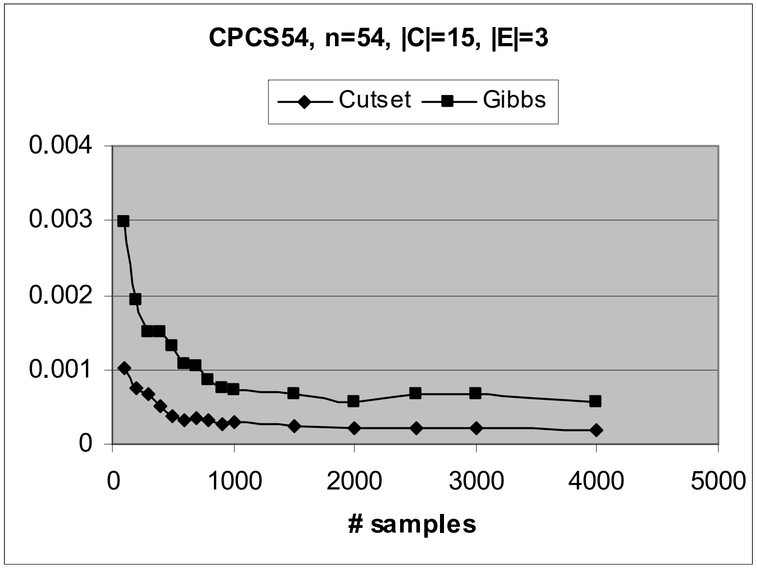
$$c^1 = \{x_2^1, x_5^1\} \Rightarrow P(x_3 \mid x_2^1, x_5^1, x_9)$$

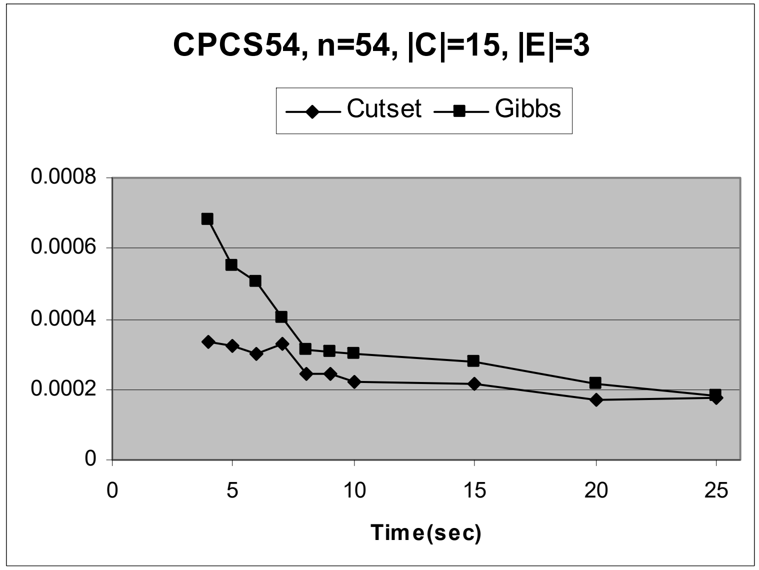
$$c^2 = \{x_2^2, x_5^2\} \Rightarrow P(x_3 \mid x_2^2, x_5^2, x_9)$$

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

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

CPCS54 Test Results



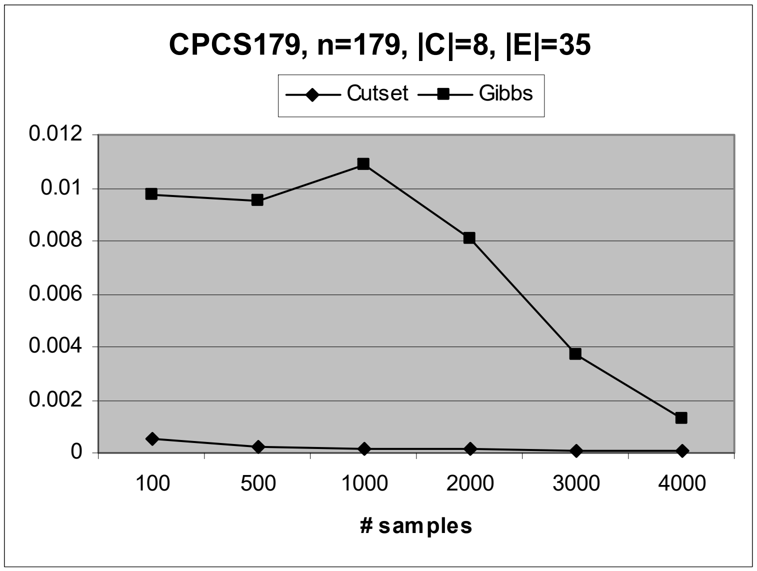


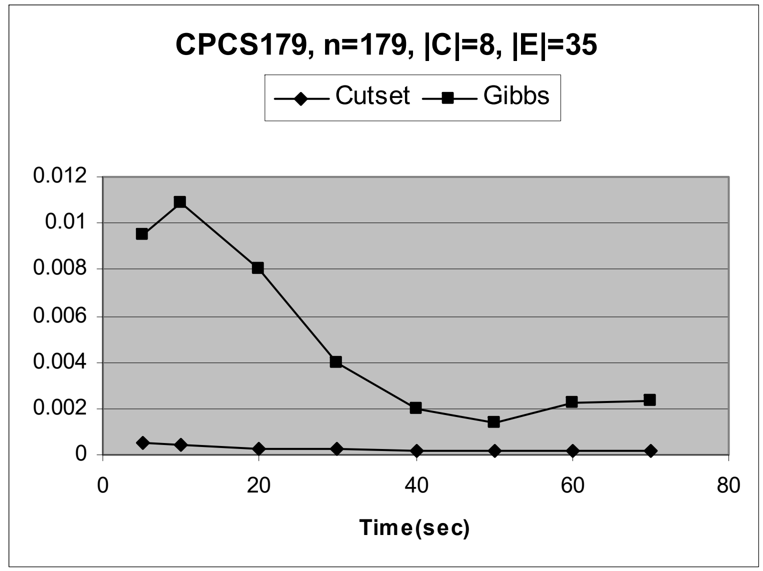
MSE vs. #samples (left) and time (right)

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

Exact Time = 30 sec using Cutset Conditioning

CPCS179 Test Results

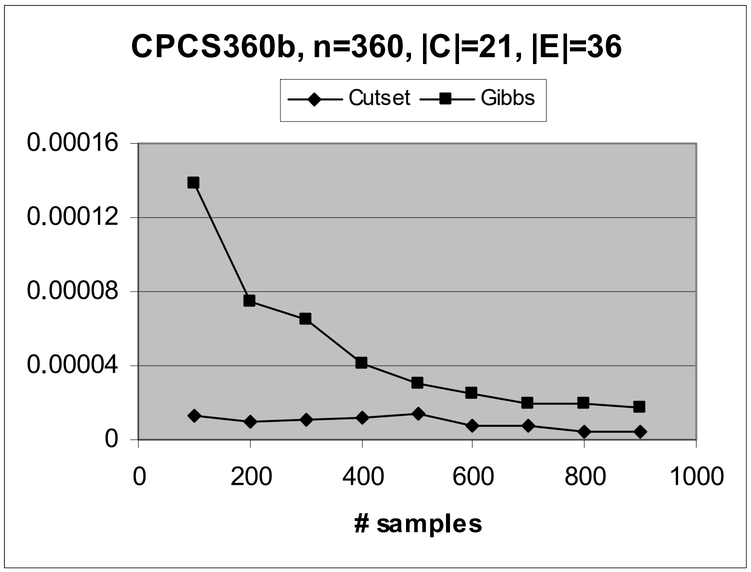


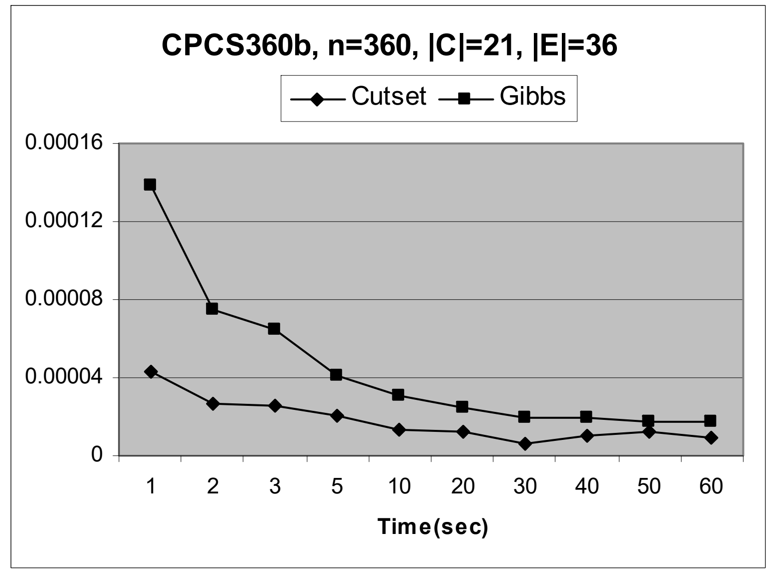


MSE vs. #samples (left) and time (right)
Non-Ergodic (1 deterministic CPT entry)
|X| = 179, |C| = 8, 2<= $D(X_i)$<=4, |E| = 35

Exact Time = 122 sec using Cutset Conditioning

CPCS360b Test Results





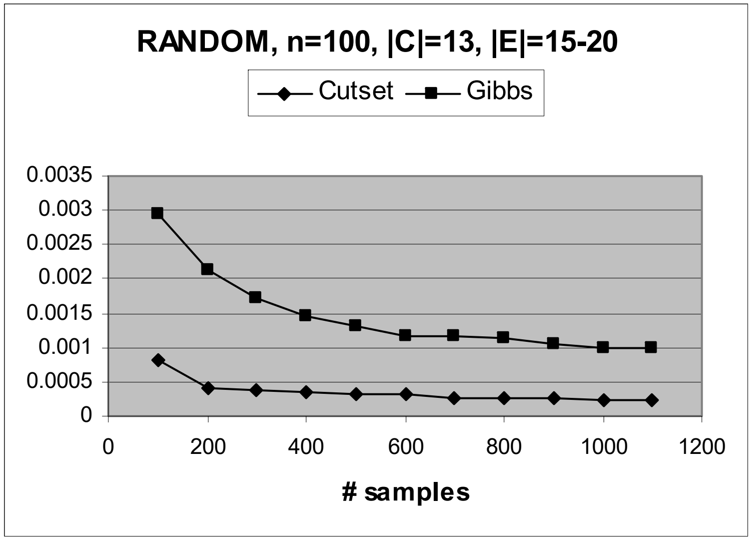
MSE vs. #samples (left) and time (right)

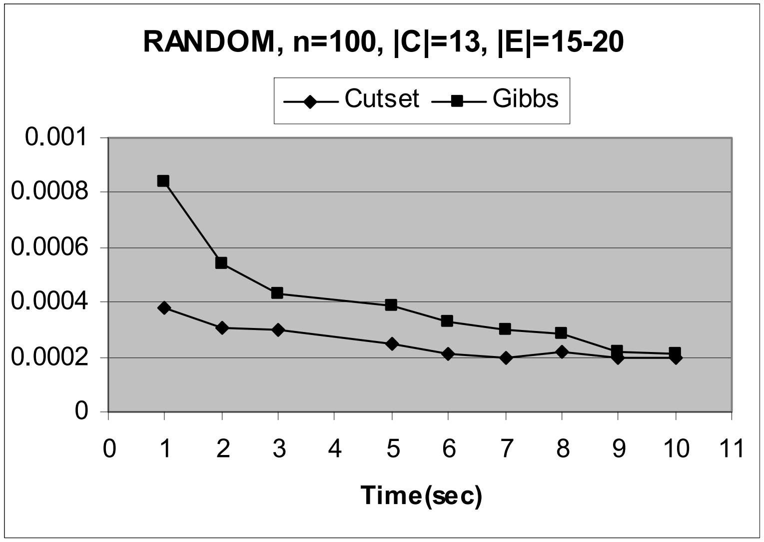
Ergodic, |X| = 360, $D(X_i)$=2, |C| = 21, |E| = 36

Exact Time > 60 min using Cutset Conditioning

Exact Values obtained via Bucket Elimination

Random Networks





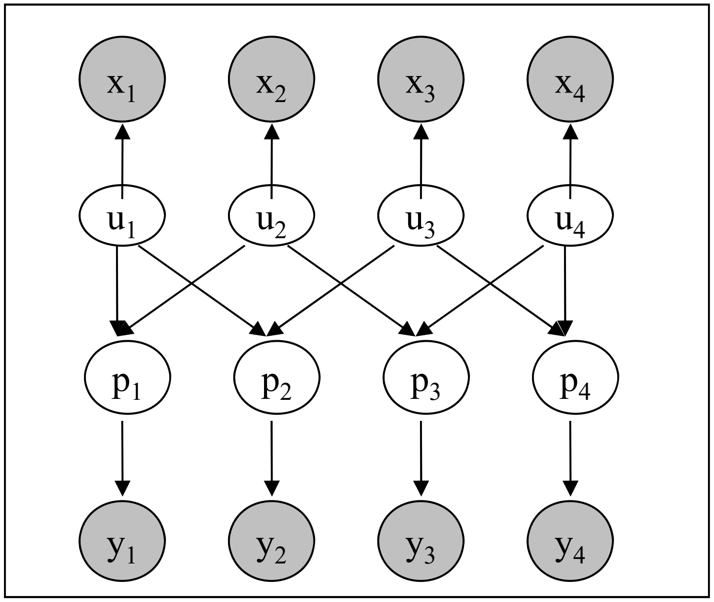
MSE vs. #samples (left) and time (right)

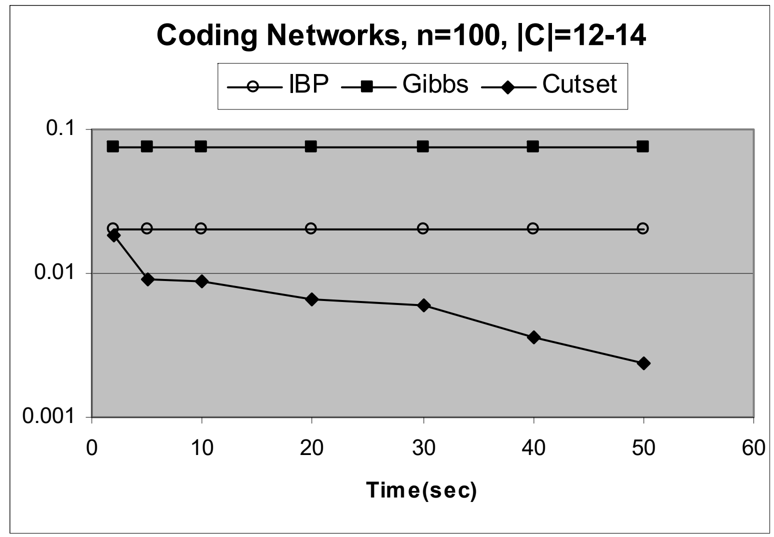
$|X| = 100$, $D(X_i) = 2$, $|C| = 13$, $|E| = 15\text{-}20$

Exact Time = 30 sec using Cutset Conditioning

Coding Networks

Cutset Transforms Non-Ergodic Chain to Ergodic





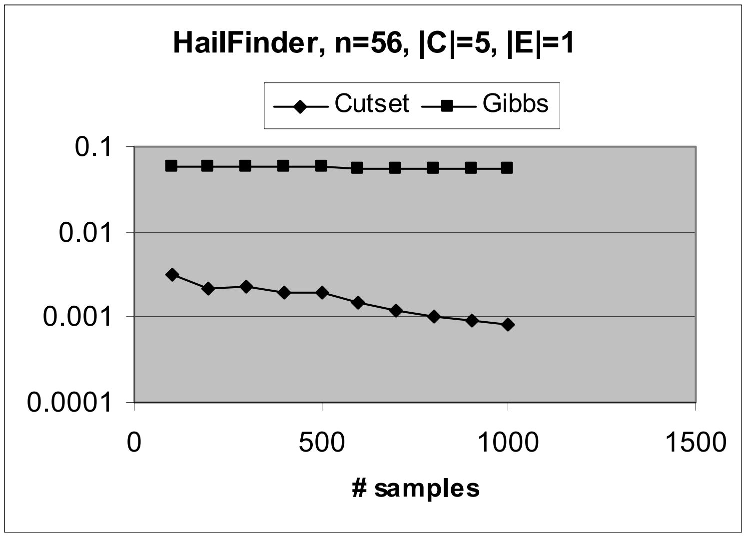
MSE vs. time (right)

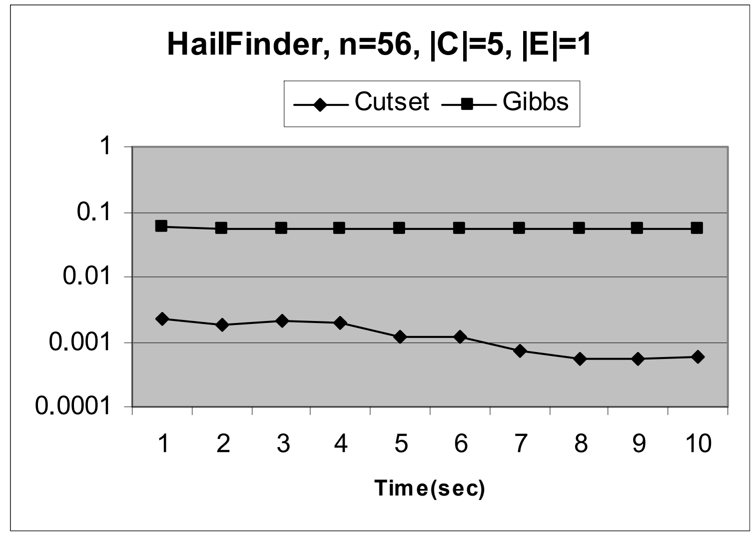
Non-Ergodic, |X| = 100, $D(X_i)=2$, |C| = 13-16, |E| = 50

Sample Ergodic Subspace $U=\{U_1, U_2, \ldots U_k\}$

Exact Time = 50 sec using Cutset Conditioning

Non-Ergodic Hailfinder



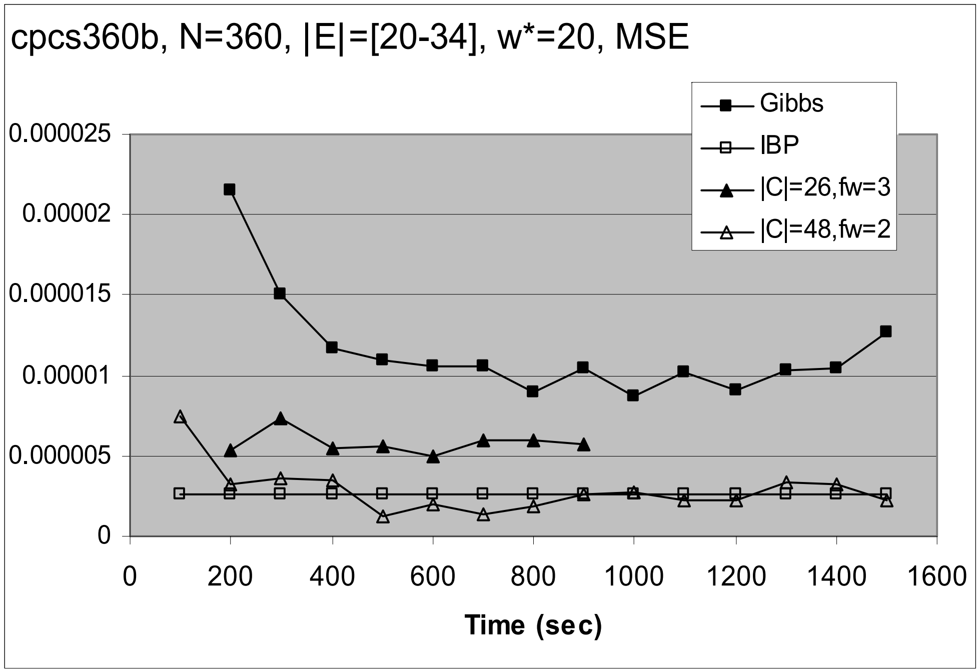


MSE vs. #samples (left) and time (right)

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:

$$\text{For } Z_i \in Z \text{ do}:$$

$$\text{If } Z_i \in E$$

$$z_i^{t+1} = z_i, z_i \in e$$

$$\text{Else}$$

$$z_i^{t+1} \leftarrow P(Z_i \mid z_1^{t+1}, \ldots, z_{i-1}^{t+1})$$

$$\text{End If}$$

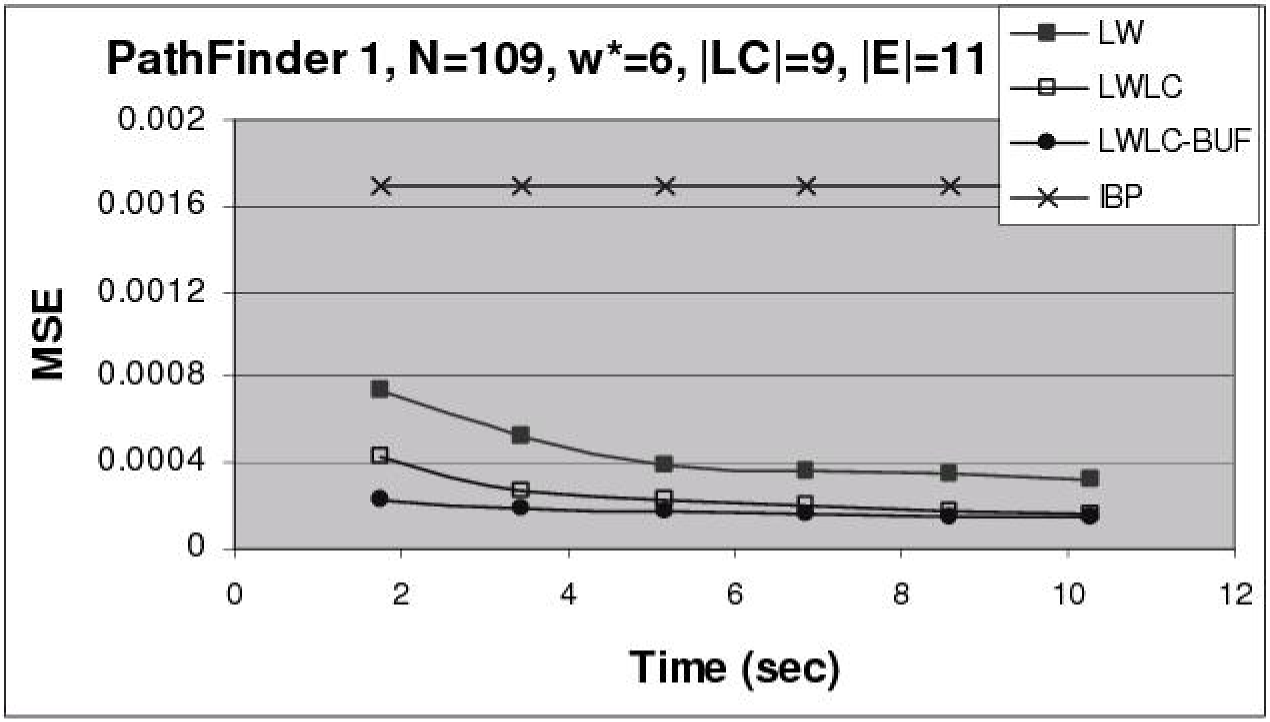
$$\text{End For}$$

- computed while generating sample t using bucket tree elimination

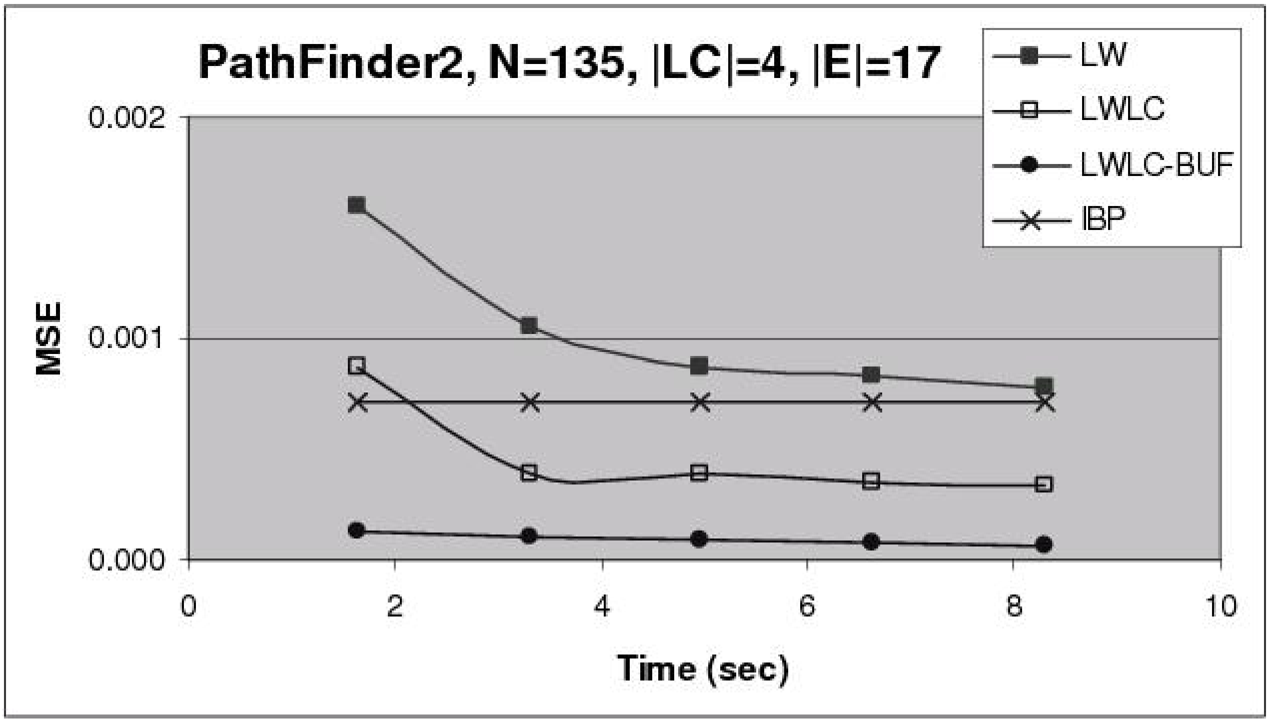
- can be memoized for some number of instances K (based on memory available

KL[P(C|e), Q(C)] ≤ KL[P(X|e), Q(X)]

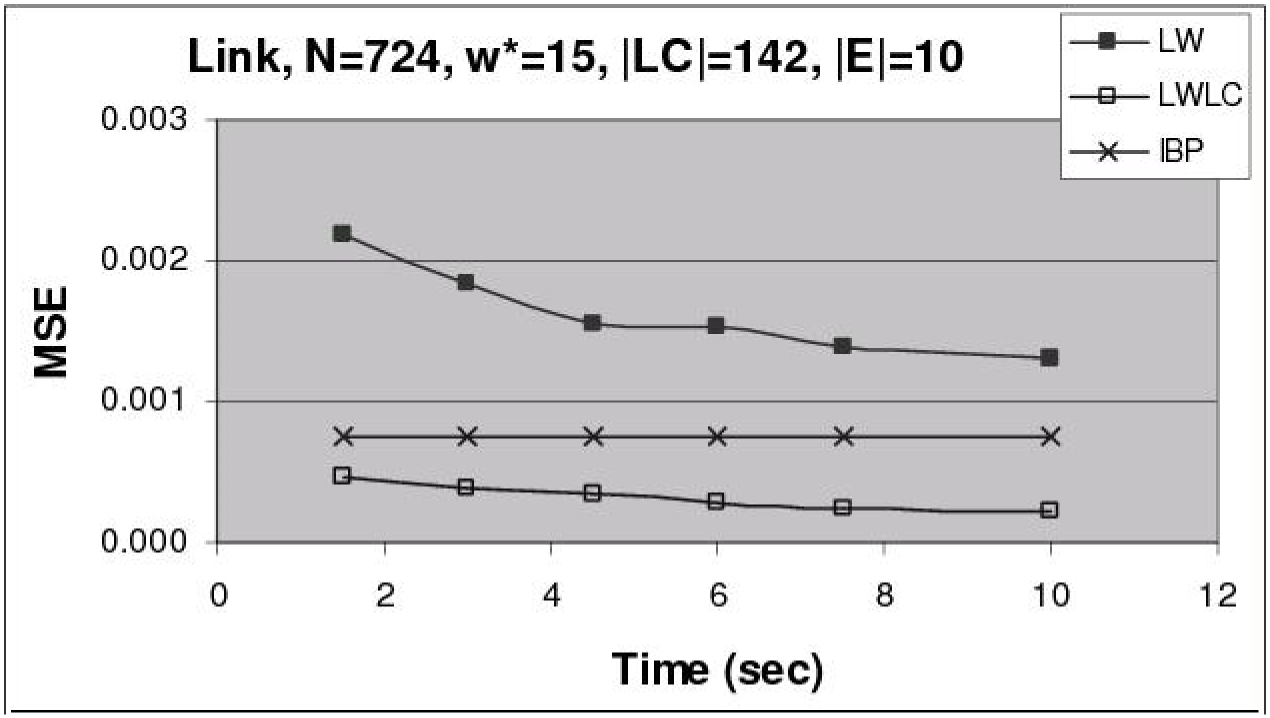
Pathfinder 1



Pathfinder 2



Link



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

Summary

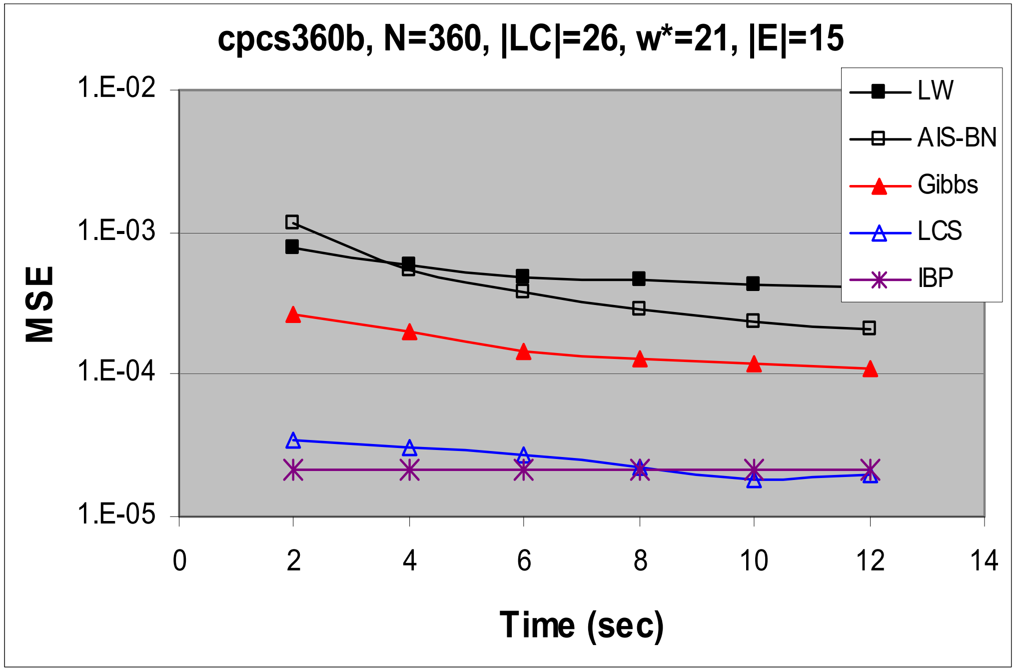
Importance Sampling

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

Gibbs Sampling

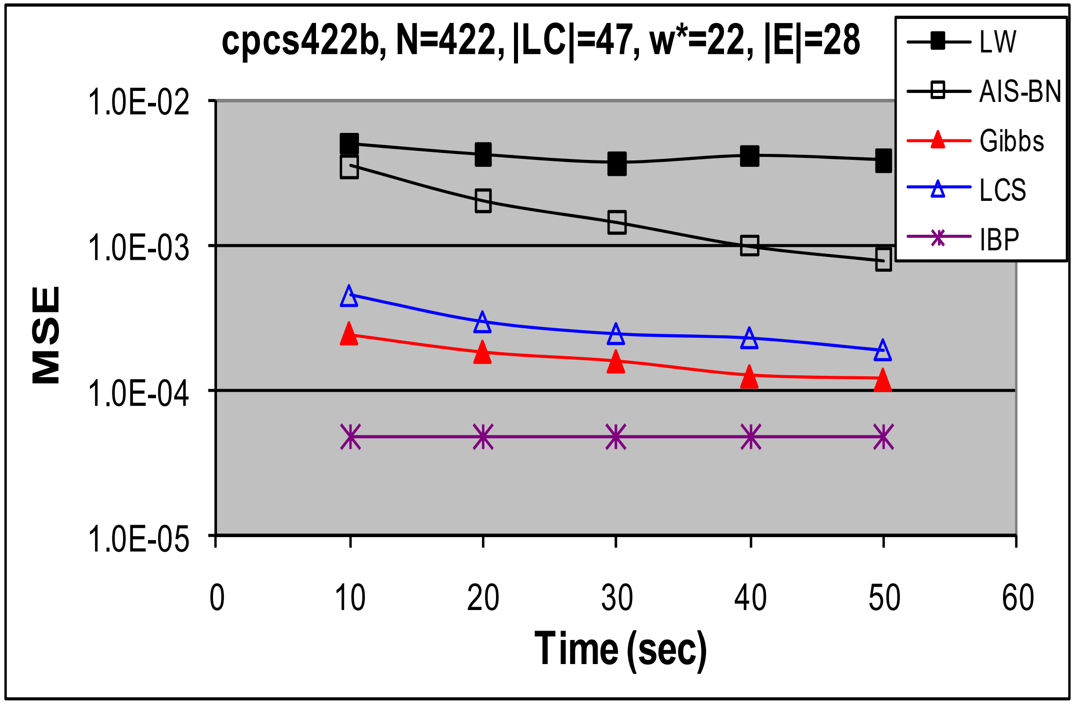
- Dependent samples
- Biased estimator
- Generates samples slower
- Samples from $\overline{P}(X|e)$
- Does not converge in presence of constraints
- Improves on cutset

CPCS360b



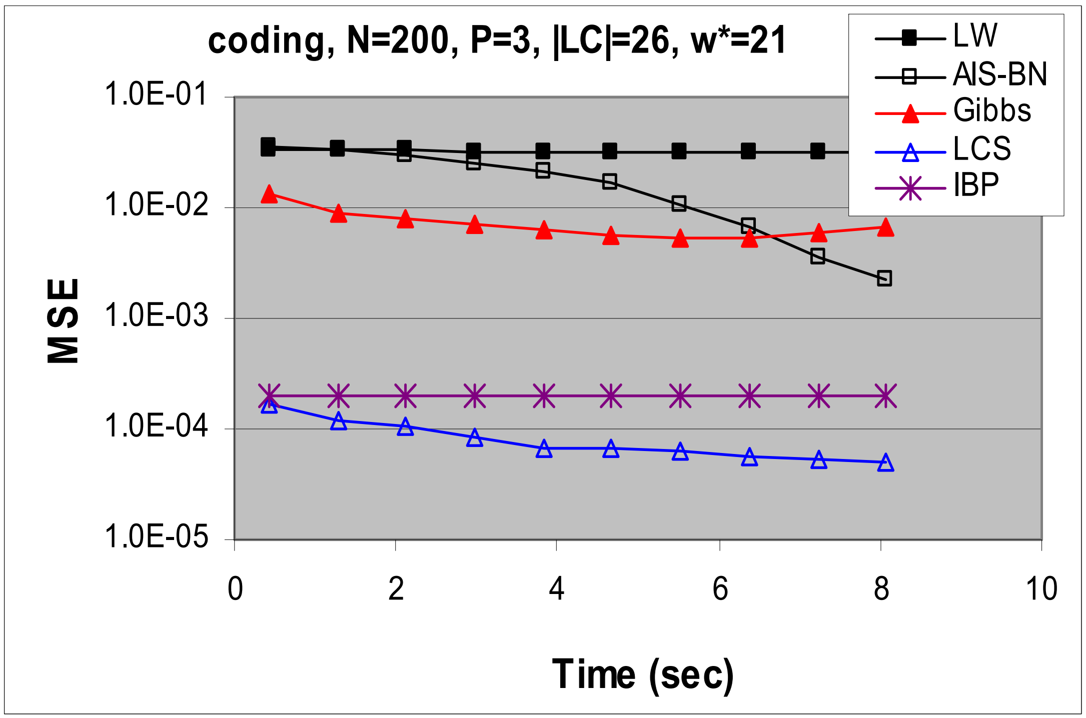
LW – likelihood weighting
LCS – likelihood weighting on a cutset

CPCS422b



LW – likelihood weighting
LCS – likelihood weighting on a cutset

Coding Networks



LW – likelihood weighting
LCS – likelihood weighting on a cutset

Rao-Blackwell Sampling

- Given a Bayesian network over disjoint variables $\mathbf{X}$ and $\mathbf{Y}$
- Goal is to estimate the probability of some event α, $\Pr(\alpha)$
- Assume $\Pr(\alpha|\mathbf{y})$ can be computed efficiently for any instantiation $\mathbf{y}$
- Rao-Blackwell sampling can exploit this fact to reduce the variance, by sampling from the distribution $\Pr(\mathbf{Y})$ instead the full distribution $\Pr(\mathbf{X}, \mathbf{Y})$

Rao-Blackwell Sampling

Rao-Blackwell sampling:

1. Draw a sample $\mathbf{y}^1, \ldots, \mathbf{y}^n$ from the distribution $\Pr(\mathbf{Y})$
2. Compute $\Pr(\alpha|\mathbf{y}^i)$ for each sampled instantiation $\mathbf{y}^i$
3. Estimate the probability $\Pr(\alpha)$ using the average

$$(1/n) \sum_{i=1}^{n} \Pr(\alpha|\mathbf{y}^i)$$

Will generally have a smaller variance than direct sampling.

Rao-Blackwell Sampling

The Rao-Blackwell (RB) function for event α and distribution $\Pr(\mathbf{X}, \mathbf{Y})$

maps each instantiation $\mathbf{y}$ into $[0, 1]$ as follows:

$$\ddot{\alpha}(\mathbf{y}) \stackrel{def}{=} \Pr(\alpha|\mathbf{y})$$

If our sample is $\mathbf{y}^1, \ldots, \mathbf{y}^n$, and if we use Monte Carlo simulation to estimate the expectation of the RB function $\ddot{\alpha}(\mathbf{Y})$, then our estimate will simply be the sample mean:

$$\mathrm{Av}_n(\ddot{\alpha}) = \frac{1}{n} \sum_{i=1}^{n} \Pr(\alpha|\mathbf{y}^i)$$