Sampling Techniques for Probabilistic and Deterministic Graphical models

Bozhena Bidyuk
Vibhav Gogate
Rina Dechter

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

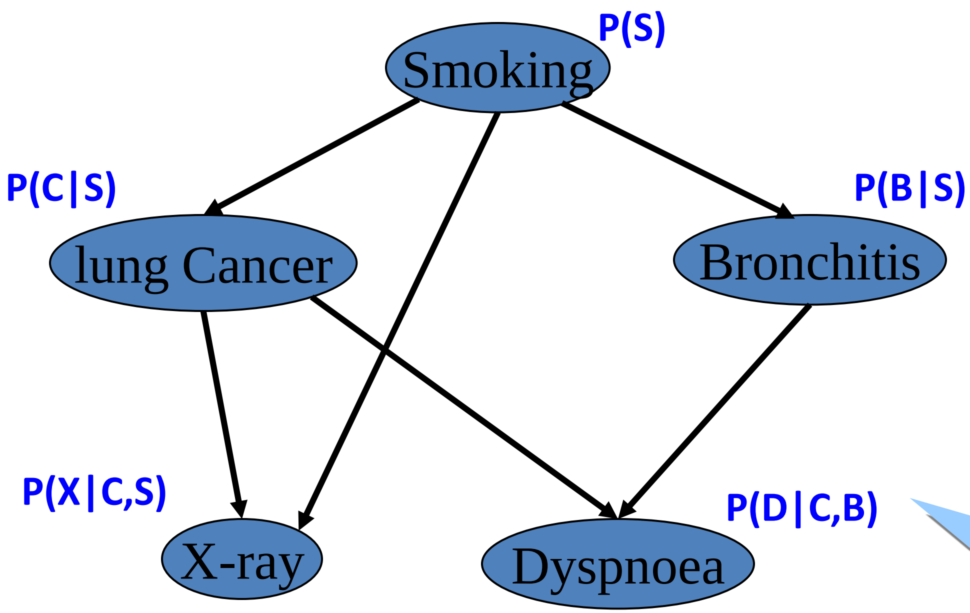
Probabilistic Reasoning; Graphical models

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

Bayesian Networks (Pearl, 1988)

CPTs : $P(X_i \mid pa(X_i))$

$$P(X) = \prod_{i=1}^{n} P(X_i \mid pa(X_i))$$



CPT:

C	B	P(D|C,B)	
0	0	0.1	0.9
0	1	0.7	0.3
1	0	0.8	0.2
1	1	0.9	0.1

$P(S, C, B, X, D) = P(S)\, P(C|S)\, P(B|S)\, P(X|C,S)\, P(D|C,B$

Belief Updating:

P (lung cancer=yes | smoking=no, dyspnoea=yes) = ?

Probability of evidence:

P (smoking=no, dyspnoea=yes) = ?

Queries

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

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

- **Posterior marginal (beliefs):**

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

- **Most Probable Explanation**

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

Constraint Networks

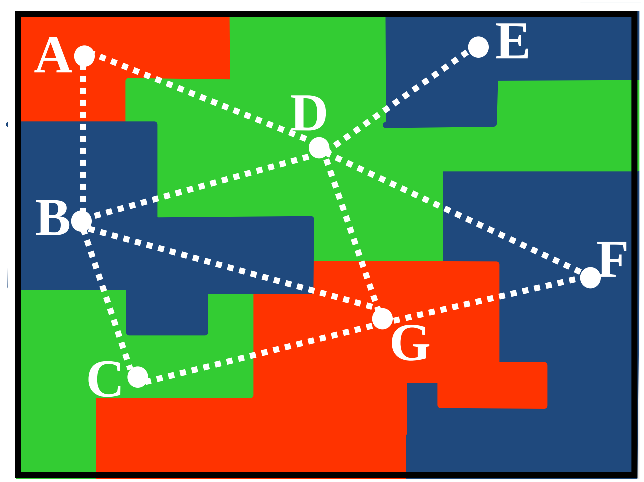
Map coloring

Variables: countries (A B C etc.)

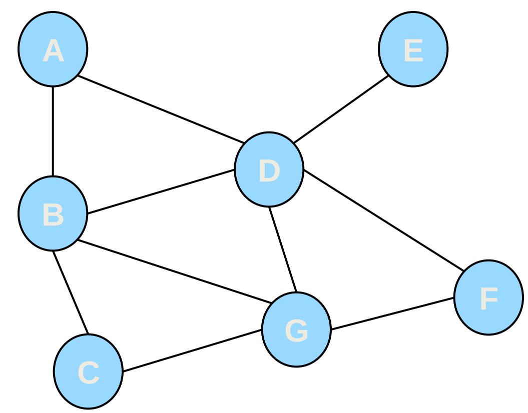
Values: colors (red green blue)

Constraints: $A \neq B$, $A \neq D$, $D \neq E$, etc.

A	B
red	green
red	yellow
green	red
green	yellow
yellow	green
yellow	red



Constraint graph

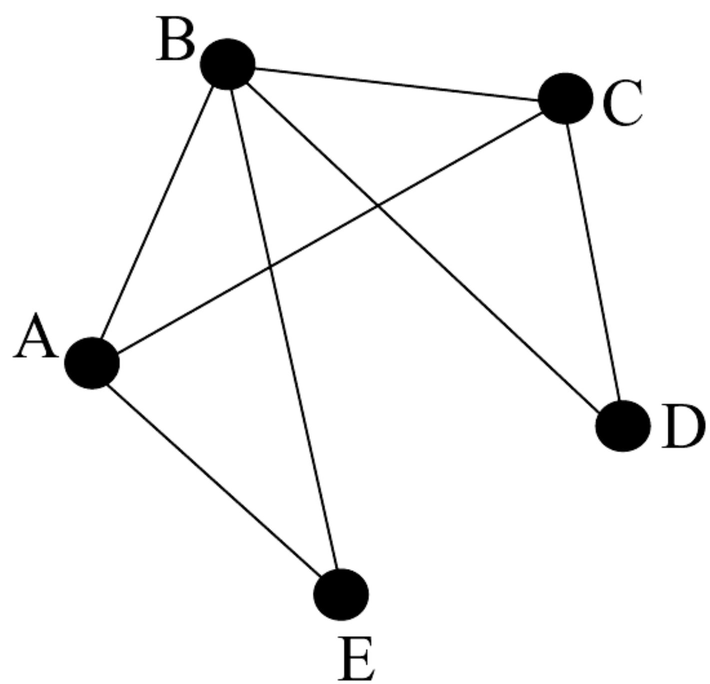


Task: find a solution

Count solutions, find a good one

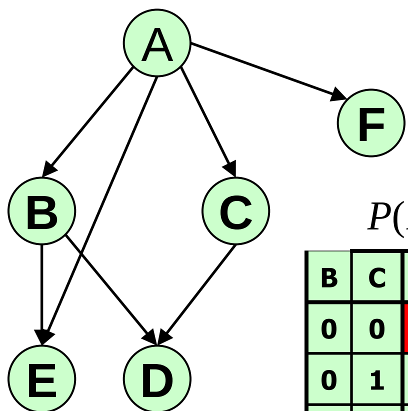
Propositional Satisfiability

$\varphi = \{(\neg C), (A \vee B \vee C), (\neg A \vee B \vee E), (\neg B \vee C \vee D)\}$.



Mixed Networks: Mixing Belief and Constraints

Belief or Bayesian Networks



$P(D|B,C)$

B	C	D=0	D=1
0	0	0	1
0	1	.1	.9
1	0	.3	.7
1	1	1	0

B=

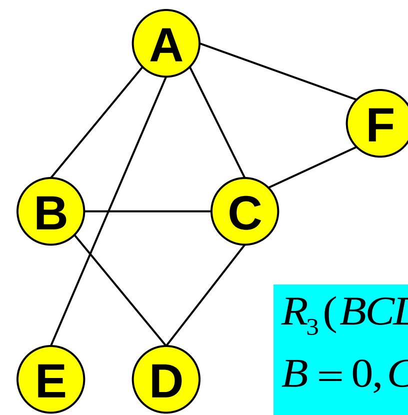
Variables: A, B, C, D, E, F

Domains : $D_A = D_B = D_C = D_D = D_E = D_F = \{0,1\}$

CPTS: $P(A), P(B|A), P(C|A), P(D|B,C)$

$P(E|A,B), P(F|A)$

Constraint Networks



$R_3(BCD)$

$B=0, C=0, D=0$ is not allowed

$B=1, C=1, D=1$ is not allowed

R=

Variables: A, B, C, D, E, F

Domains : $D_A = D_B = D_C = D_D = D_E = D_F = \{0,1\}$

Constraints: $R_1(ABC), R_2(ACF), R_3(BCD), R_4(A,E)$

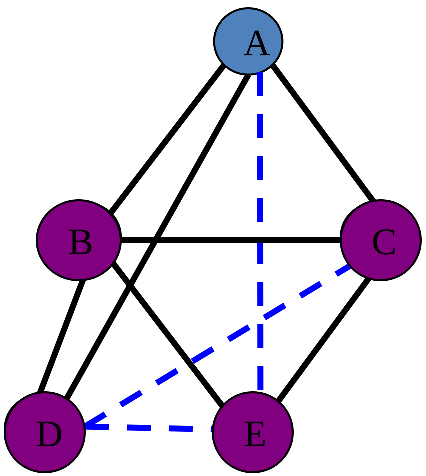
Expresses the set of solutions : $sol(R)$

Constraints could be specified externally or may occur as zeros in the Belief network

Same queries (e.g., weighted counts)

$$M = \sum_{x \in sol(R)} P_B(x)$$

Belief Updating



"Moral" graph

$$P(a|e=0) \propto P(a,e=0)=$$

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

$$P(a)\sum_{e=0}\sum_{d}\sum_{c}P(c|a)\underbrace{\sum_{b}P(b|a)P(d|b,a)P(e|b,c)}_{h^B(a,d,c,e)}$$

Variable Elimination

Bucket Elimination

Algorithm *elim-bel* (Dechter 1996)

$\sum_b \prod$ ← Elimination operator

bucket B: P(b|a) P(d|b,a) P(e|b,c)

bucket C: P(c|a) $h^B(a,d,c,e)$

bucket D: $h^C(a,d,e)$

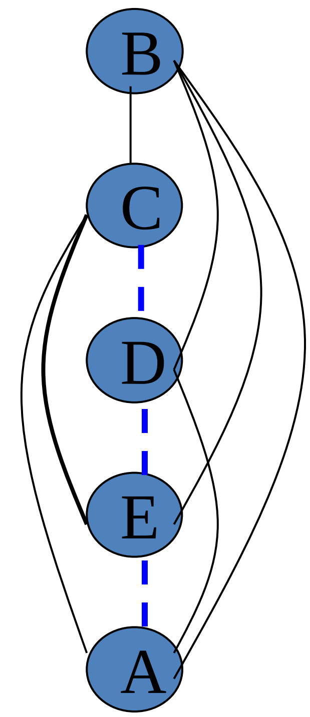
bucket E: e=0 $h^D(a,e)$

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

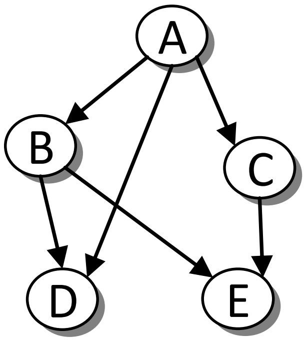
$P(a|e=0)$

W*=4

Exp(w*)



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

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

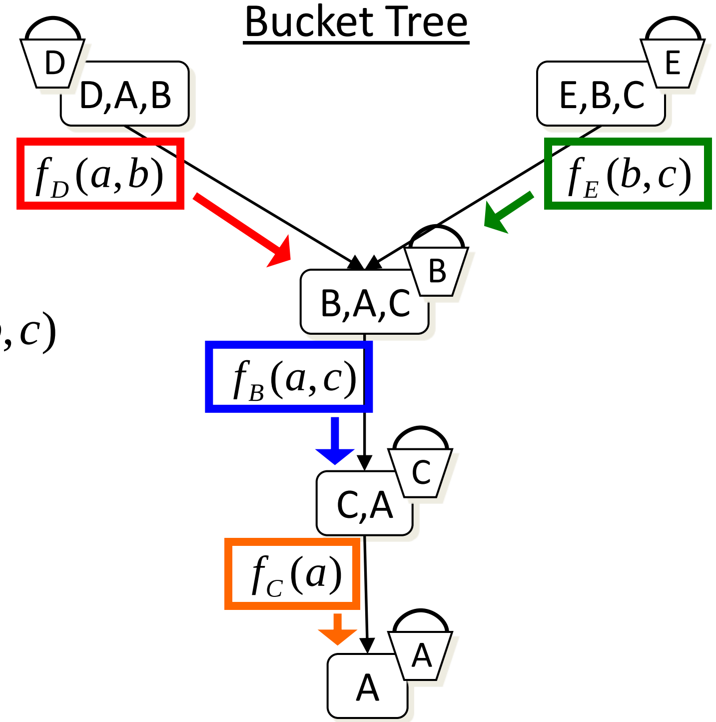
$$f_E(b,c) = P(e=0 \mid b,c)$$

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

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

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

Bucket Tree



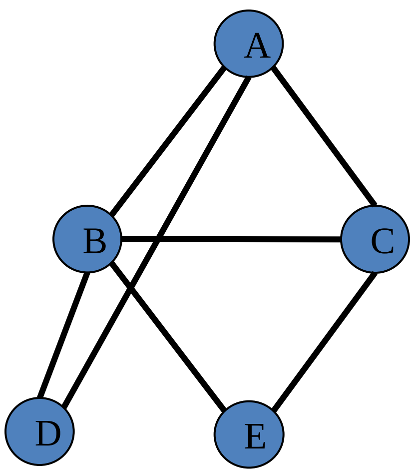
Time and space exp(w*)

Complexity of Elimination

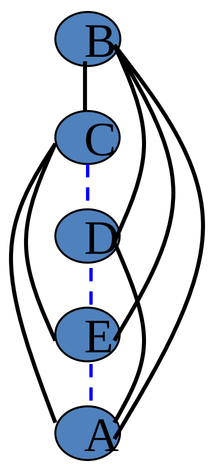
$$O(n \exp(w^*(d)))$$

$w^*(d)$ – the induced width of moral graph along ordering d

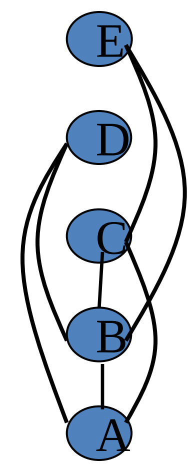
The effect of the ordering:



"Moral" graph

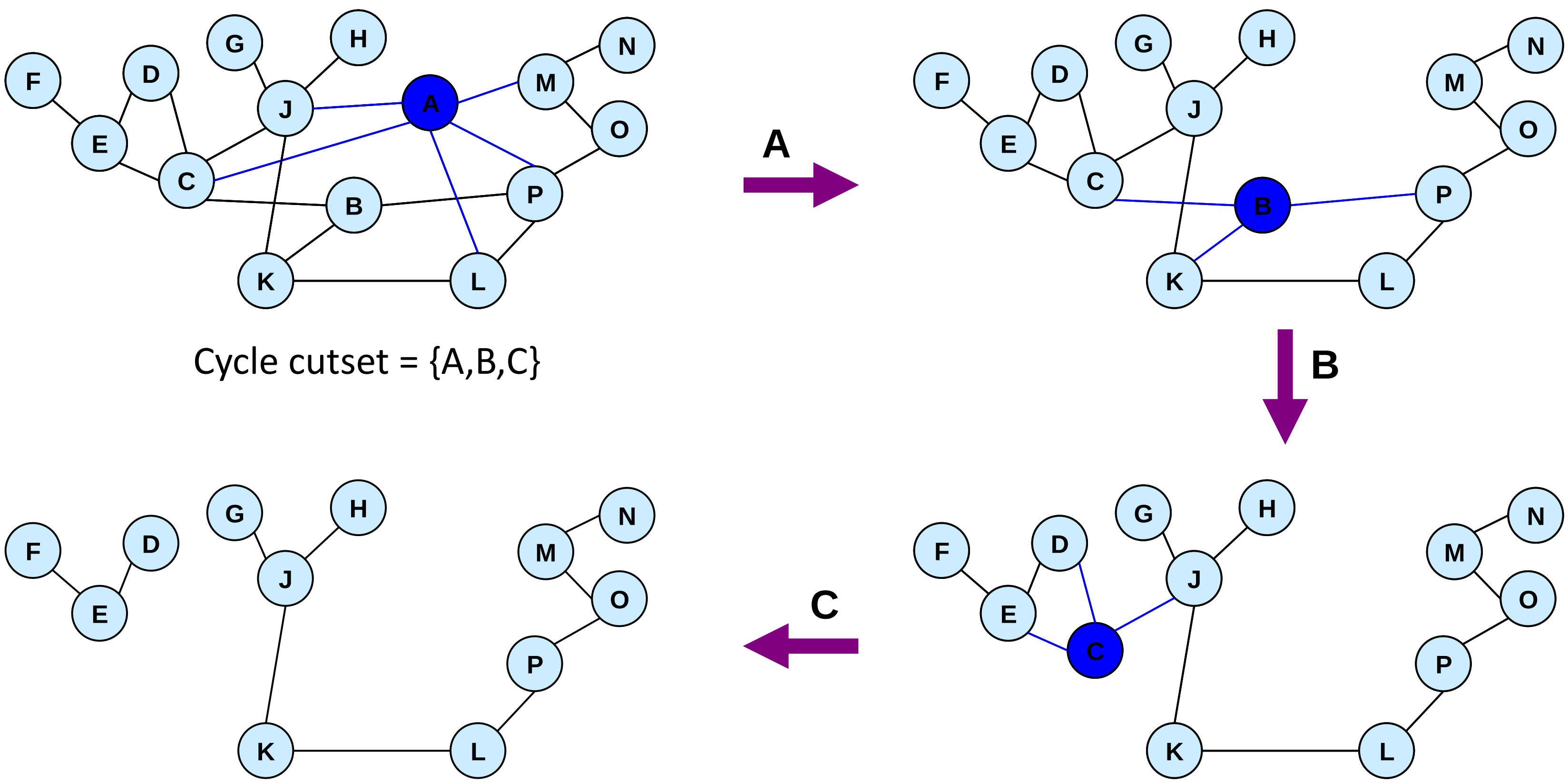


$$w^*(d_1) = 4$$



$$w^*(d_2) = 2$$

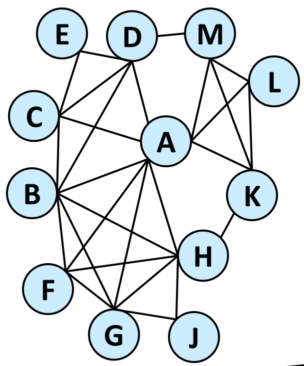
Cutset-Conditioning



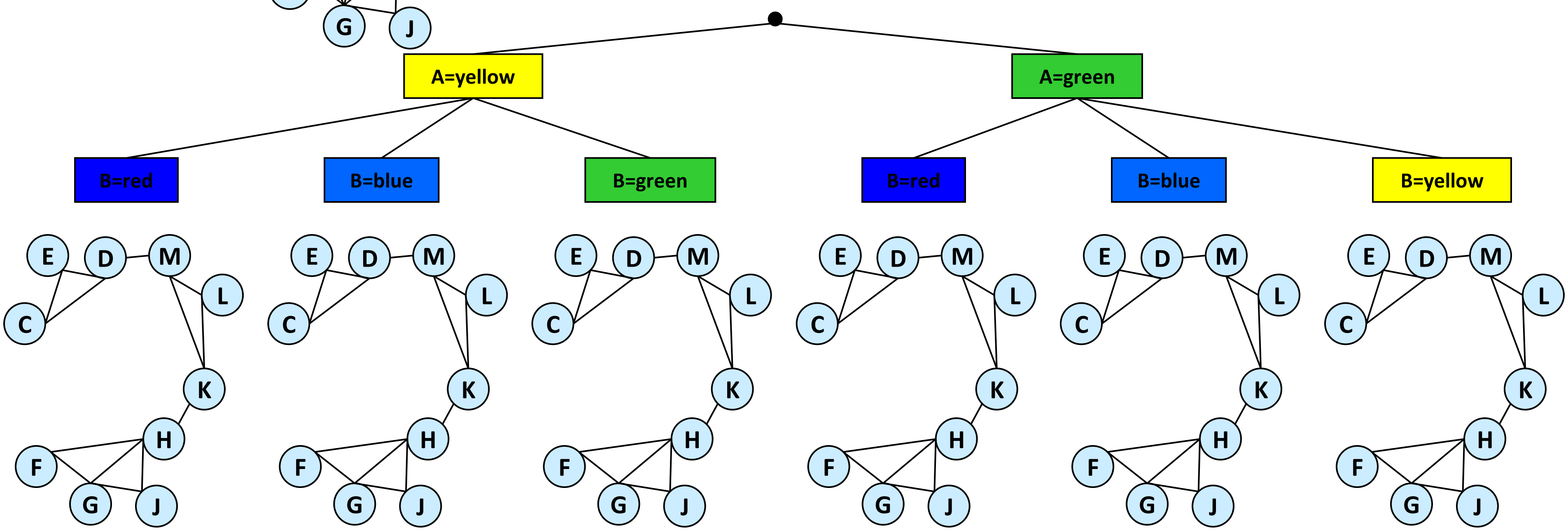
Search Over the Cutset

Space: exp(w): w is a user-controled parameter
Time: exp(w+c(w))

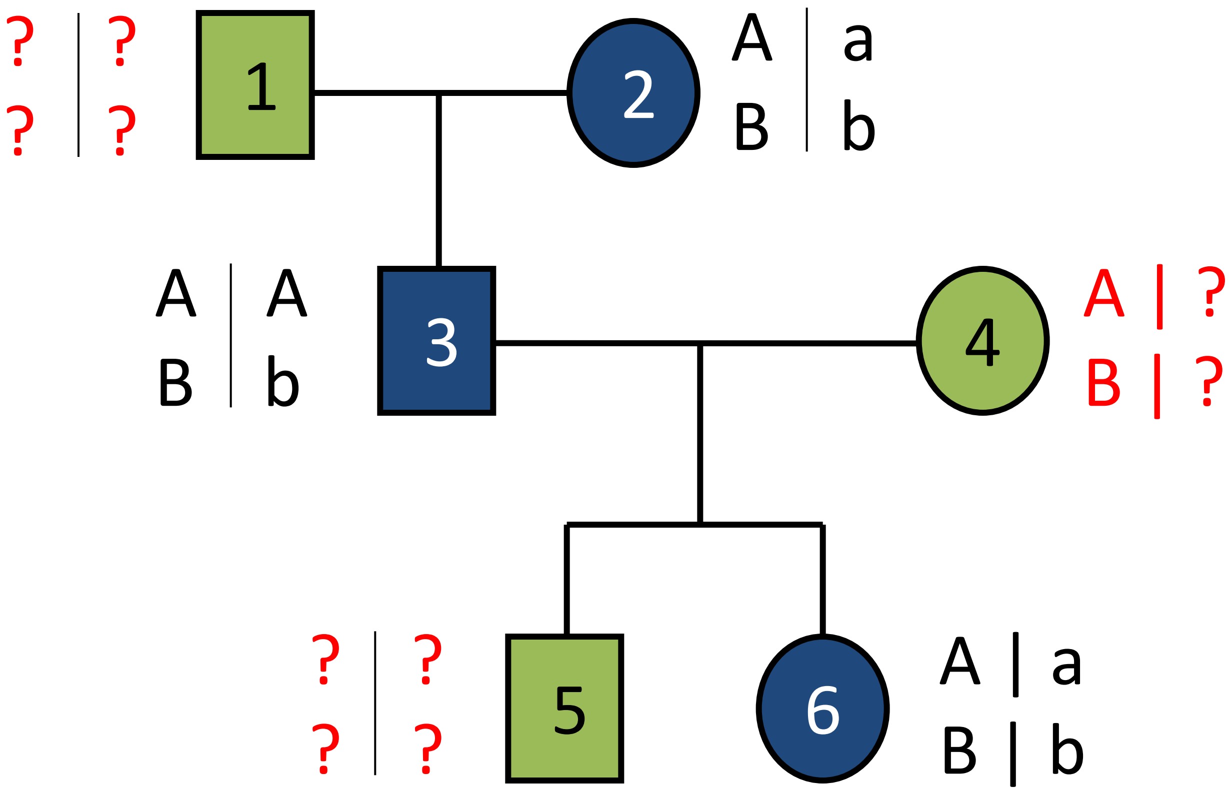
Graph Coloring problem

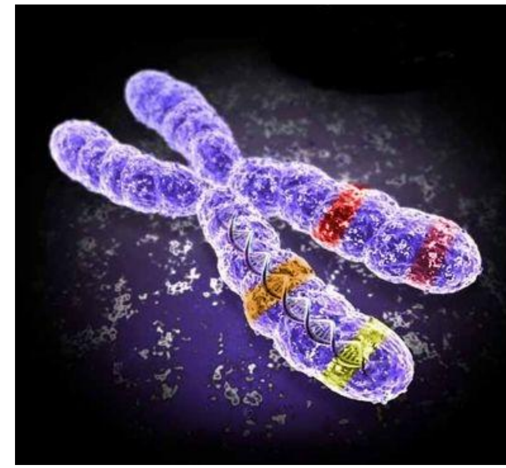


- Inference may require too much memory
- **Condition** on some of the variables



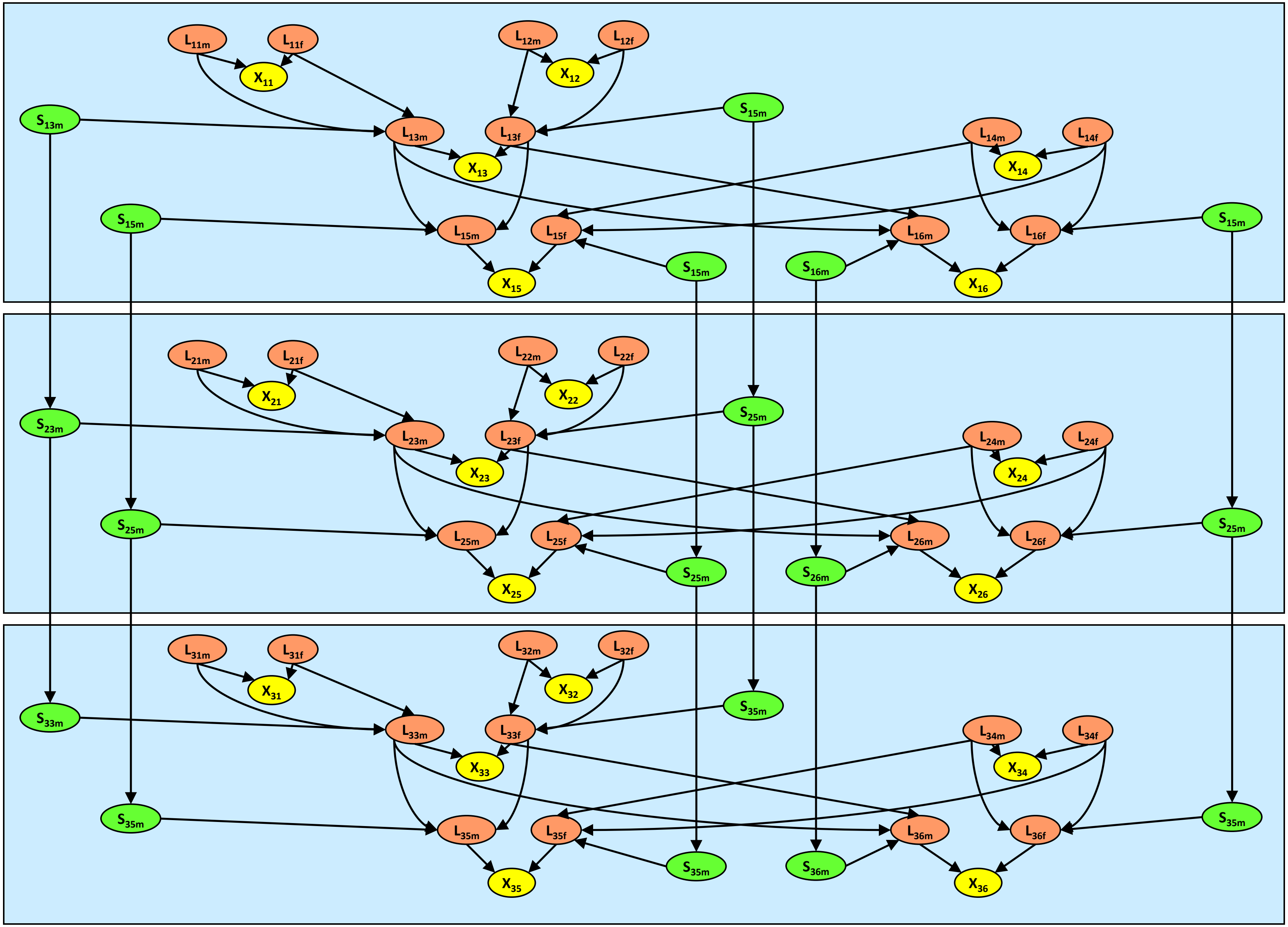
Linkage Analysis





- 6 individuals
- Haplotype: {2, 3}
- Genotype: {6}
- Unknown

Linkage Analysis: 6 People, 3 Markers

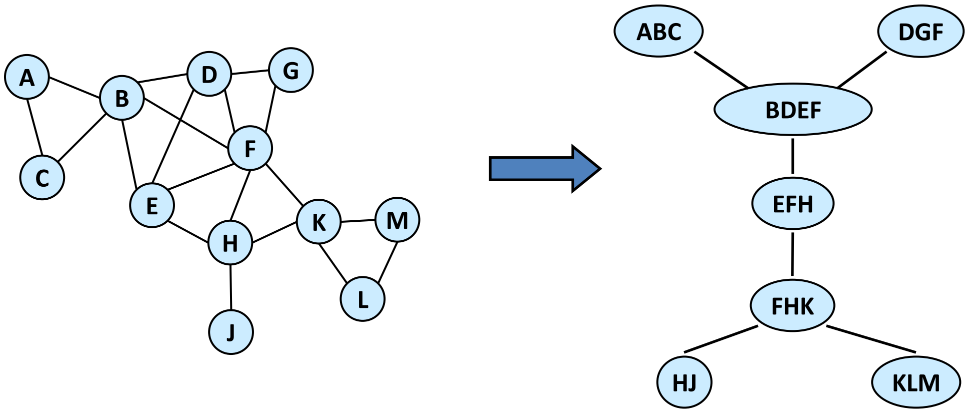


Applications

- **Determinism:** More Ubiquitous than you may think!

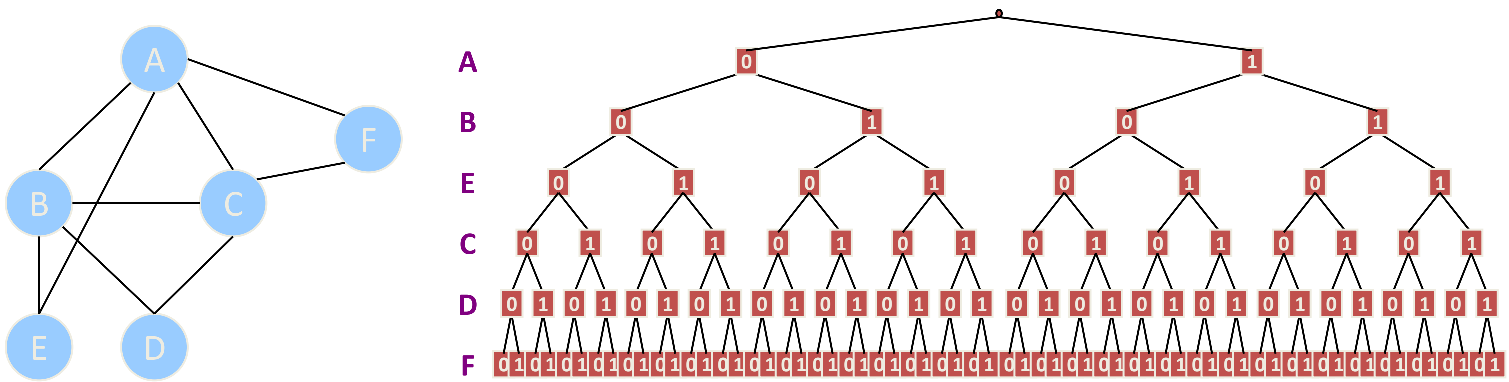
- Transportation Planning (Liao et al. 2004, Gogate et al. 2005)
  - Predicting and Inferring Car Travel Activity of individuals
- Genetic Linkage Analysis (Fischelson and Geiger, 2002)
  - associate functionality of genes to their location on chromosomes.
- Functional/Software Verification (Bergeron, 2000)
  - Generating random test programs to check validity of hardware
- First Order Probabilistic models (Domingos et al. 2006, Milch et al. 2005)
  - Citation matching

Inference vs Conditioning-Search



Inference

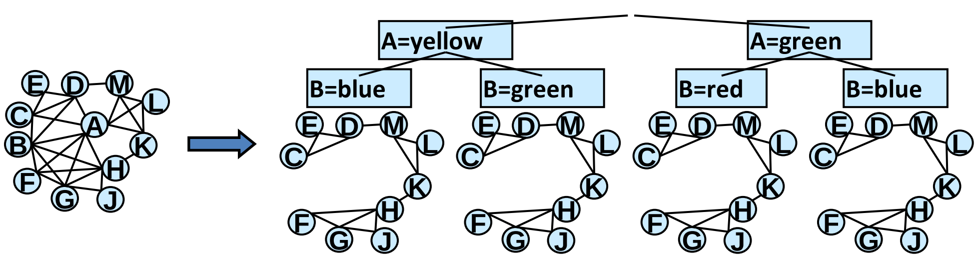
Exp(w*) time/space



Search

Exp(n) time

O(n) space



Search+inference:

Space: exp(w)

Time: exp(w+c(w))

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

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

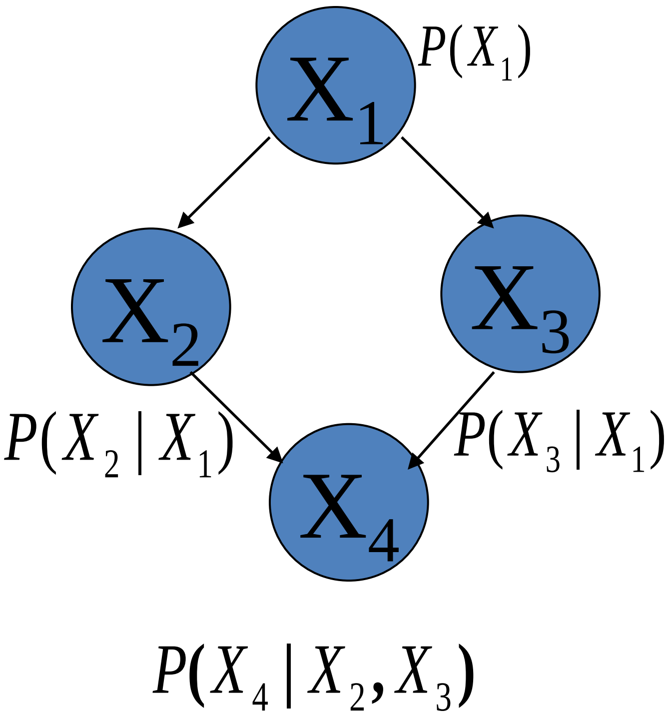
- Let $X=\{X_1,..,X_n\}$ be a set of variables
- Express the distribution in product form

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

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

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 $P(x_4 \mid X_2 = x_2, X_3 = x_3)$

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 $S^1, S^2, \ldots S^T$ drawn from P,

the Monte carlo estimate of $E_P[g(x)]$ 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

- Transform the probabilistic inference problem into the problem of computing 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.

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:

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

Monte Carlo 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: $E\left[\overline{P}(x_i \mid e)\right] \neq P(x_i \mid e)$

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”

Effective Sample size

$$P(x_i \mid e) = \sum_z g_{x_i}(z) P(z \mid e)$$

Given samples from $P(z \mid e)$, we can estimate $P(x_i \mid e)$ using :

$$\hat{P}(x_i|e) = \frac{1}{T}\sum_{j=1}^{T} g_{x_i}(z^t)$$ → **Ideal estimator**

$$Define: ESS(Q,T) = \frac{T}{1+\mathrm{var}_Q[w(z)]}$$ → **Measures how much the estimator deviates from the ideal one.**

$$\frac{Var_P[\hat{P}(x_i \mid e)]}{Var_Q[\overline{P}(x_i \mid e)]} \approx \frac{T}{ESS(Q,T)}$$

Thus T samples from P are worth ESS(Q, T) samples from Q.

Therefore, the variance of the weights must be as small as possible.

Outline

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

Likelihood Weighting: Proposal Distribution

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

Example :

Given a Bayesian network : $\mathrm{P}(\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3) = \mathrm{P}(\mathrm{X}_1) \times \mathrm{P}(\mathrm{X}_2 \mid \mathrm{X}_1) \times \mathrm{P}(\mathrm{X}_3 \mid \mathrm{X}_1, \mathrm{X}_2)$ and

Evidence $\mathrm{X}_2 = \mathrm{x}_2$.

$$\mathrm{Q}(\mathrm{X}_1, \mathrm{X}_3) = P(X_1) \times P(X_3 \mid X_1, X_2 = x_2)$$

Weights **:**

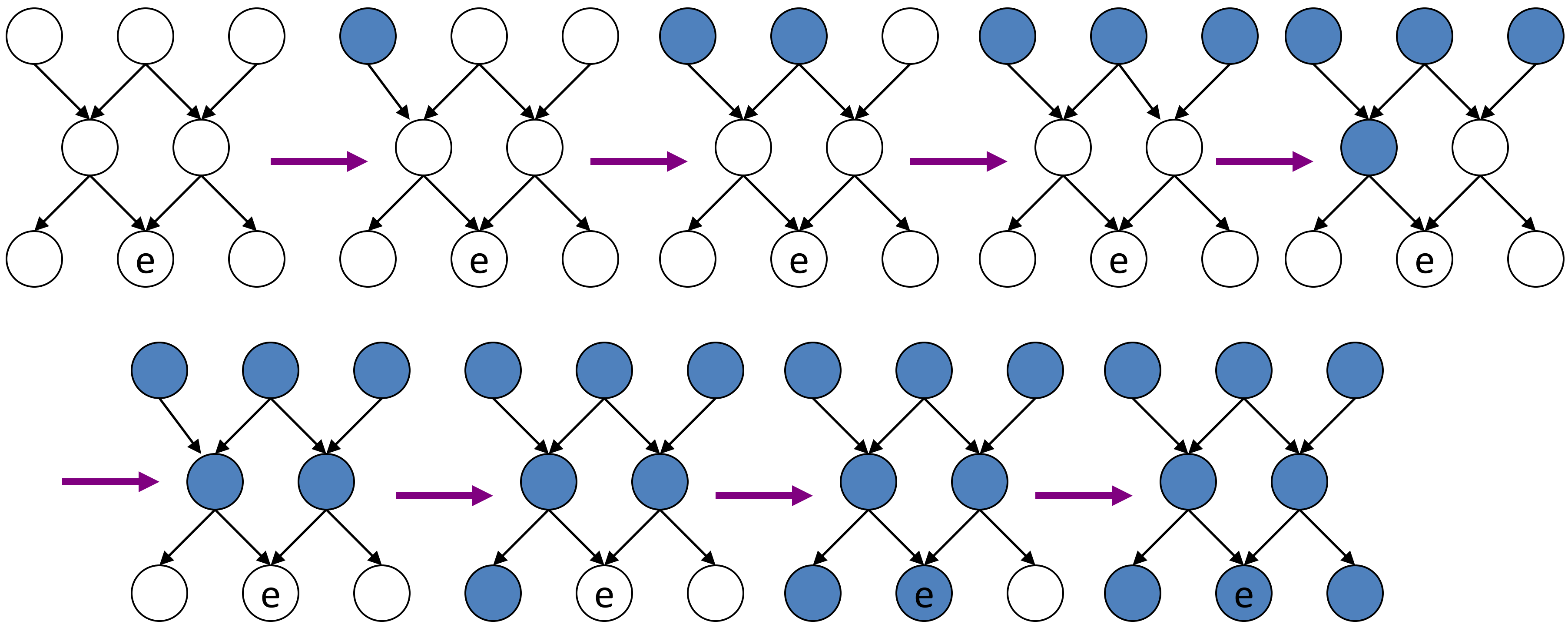
Given a sample : $x = (x_1, .., x_n)$

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

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

Likelihood Weighting: 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!

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.

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

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

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

$$\therefore Q(z) = P(z \mid e)$$

Proposal Distributions used in Literature

- AIS-BN (Adaptive proposal)
  - Cheng and Druzdzel, 2000
- Iterative Belief Propagation
  - Changhe and Druzdzel, 2003
- Iterative Join Graph Propagation (IJGP) and variable ordering
  - Gogate and Dechter, 2005

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 $P(X_i|x_1,...,x_{i-1})$ by referring to the bucket.

Bucket elimination (BE)

Algorithm *elim-bel* (Dechter 1996)

$\sum_b \prod$ ← Elimination operator

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

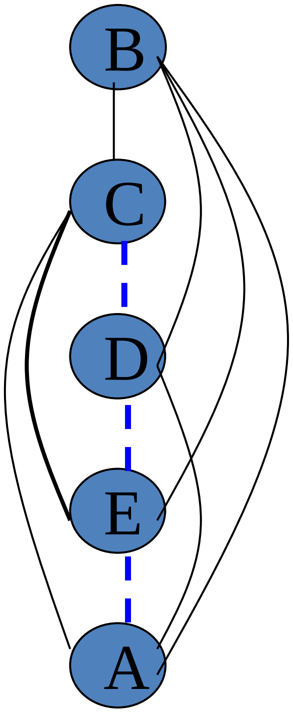
bucket C: P(C|A) $h^B(A,D,C,e)$

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

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

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

$P(e)$



Sampling from the output of BE
(Dechter 2002)

$$\text{Set } A = a, D = d, C = c \text{ in the bucket}$$
$$\text{Sample}: B = b \leftarrow Q(C \mid a, e, d) \propto P(B \mid a) P(d \mid B, a) P(e \mid b, c)$$

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

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

$$\text{Set } A = a, D = d \text{ in the bucket}$$
$$\text{Sample}: C = c \leftarrow Q(C \mid a, e, d) \propto h^B(a, d, C, e)$$

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

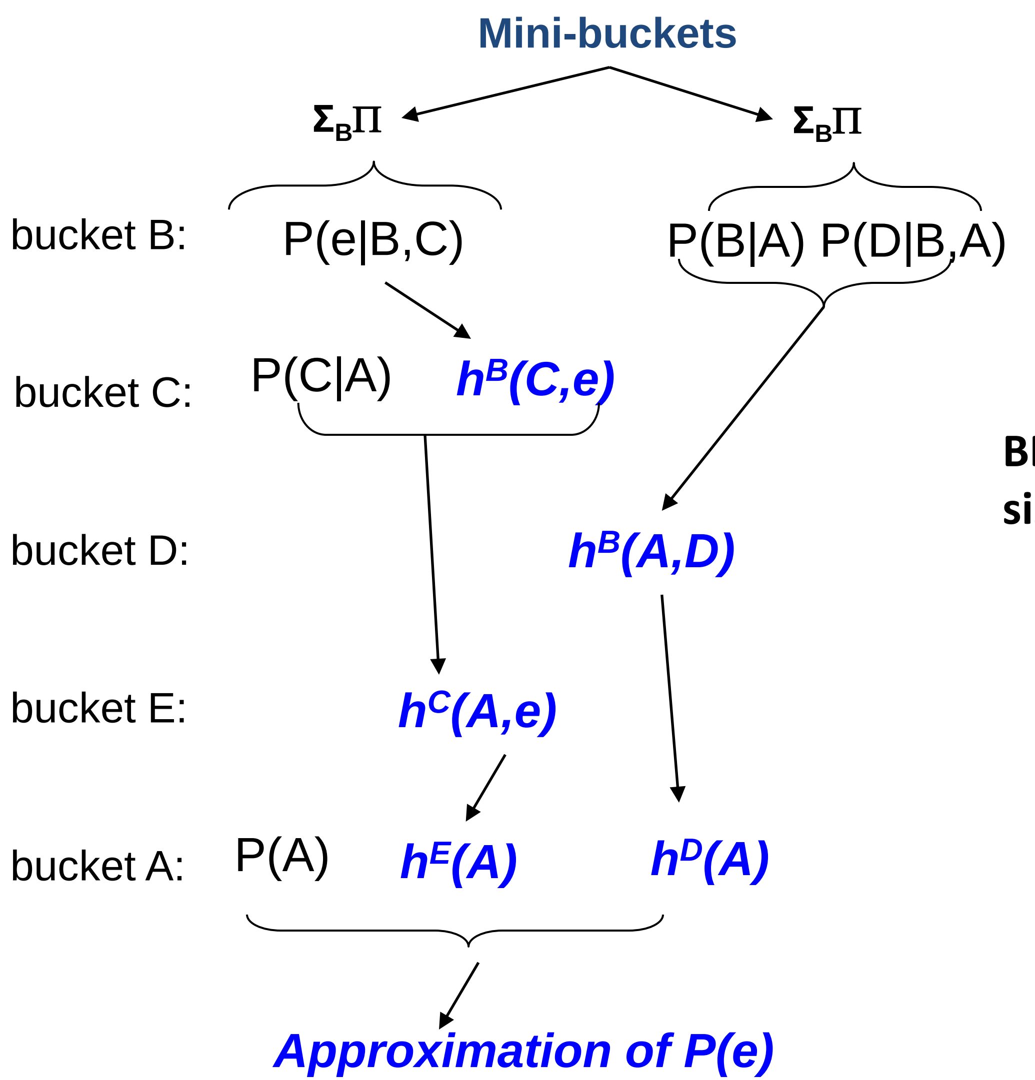
bucket A: $P(A)$ $h^E(A)$

$$\mathbf{Q(A) \propto P(A) \times h^E(A)}$$
$$\textit{Sample}: \mathbf{A = a \leftarrow Q(A)}$$

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



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!**

Adaptive Importance Sampling

$$\text{Initial Proposal} = \mathrm{Q}^1(Z) = Q(Z_1) \times Q(Z_2 \mid pa(Z_2)) \times \ldots \times Q(Z_n \mid pa(Z_n))$$

$$\hat{P}(E = e) = 0$$

$$\text{For i} = 1 \text{ to k do}$$

$$\text{Generate samples } \mathrm{z}^1, \ldots, \mathrm{z}^{\mathrm{N}} \; \textit{from} \; \mathrm{Q}^k$$

$$\hat{\mathrm{P}}(\mathrm{E} = \mathrm{e}) = \hat{P}(E = e) + \frac{1}{\mathrm{N}} \sum_{j=1}^{N} w_k(\mathrm{z}^i)$$

$$\text{Update } \mathrm{Q}^{\mathrm{k}+1} = Q^k + \eta(k)\left[Q^k - Q'\right]$$

$$End$$

$$\text{Re}\,turn \quad \frac{\hat{P}(E = e)}{\mathrm{k}}$$

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)

$$\mathrm{Q}^{'}(Z) = Q'(\mathrm{Z}_1) \times Q'(\mathrm{Z}_2 \mid pa(\mathrm{Z}_2)) \times \ldots \times Q'(\mathrm{Z}_n \mid pa(\mathrm{Z}_n))$$

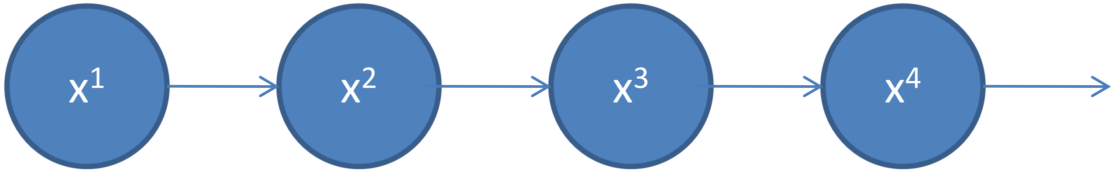
where each $\mathrm{Q}'(\mathrm{Z}_\mathrm{i} \mid \mathrm{Z}_1,..,\mathrm{Z}_{\mathrm{i}\text{-}1})$

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



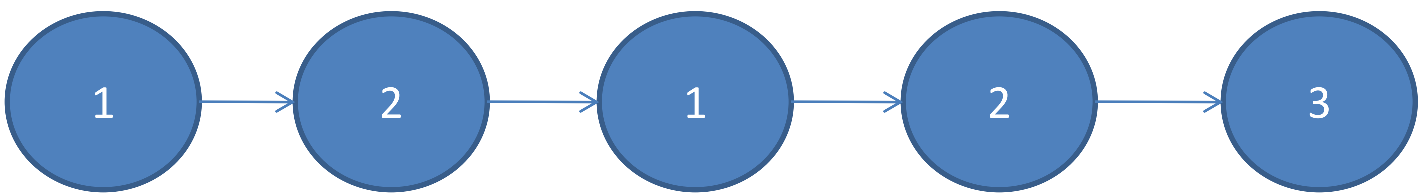
- A **Markov chain** is a discrete random process with the property that the next state depends only on the current state (**Markov Property**):

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

- 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

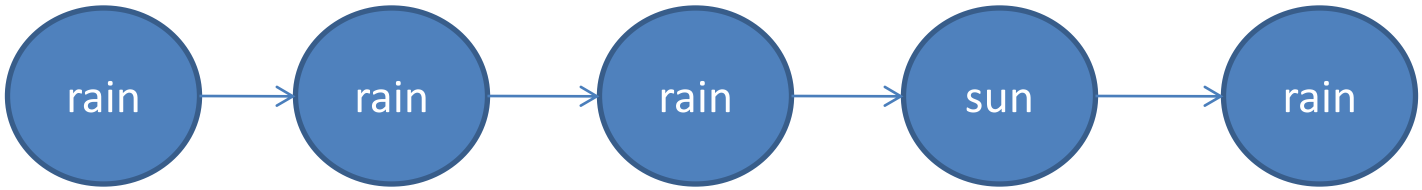


$$D(X) = \{0,1,2,\ldots\}$$

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

transition matrix P(X)

Example: Weather Model



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

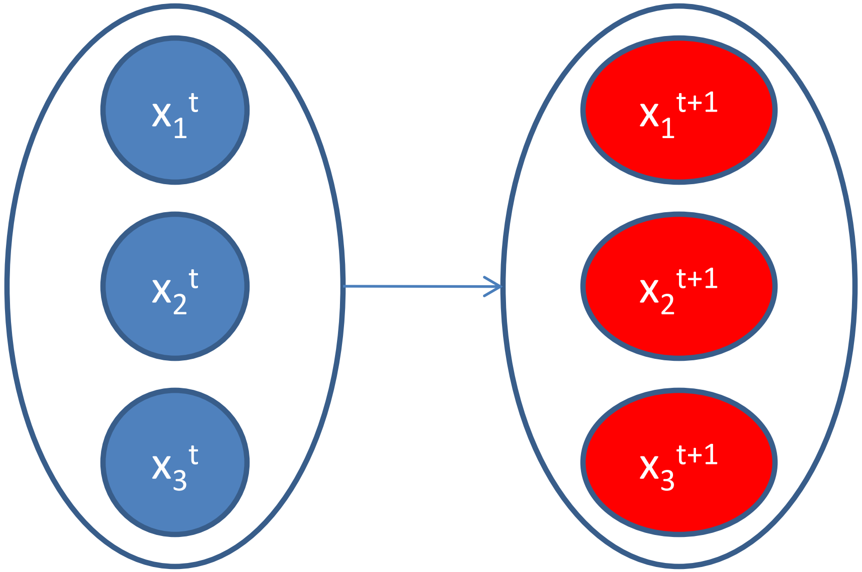
	$P(rainy)$	$P(sunny)$
$rainy$	0.9	0.1
$sunny$	0.5	0.5

transition matrix P(X)

Multi-Variable System

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

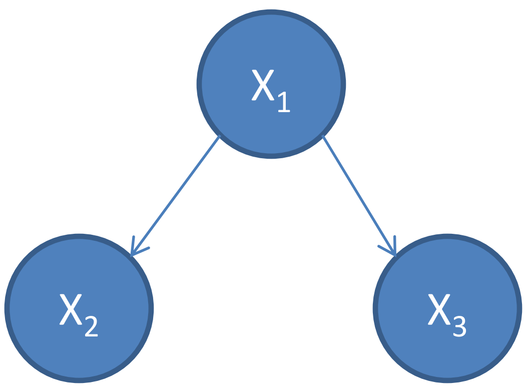
- state is an assignment of values to all the variables

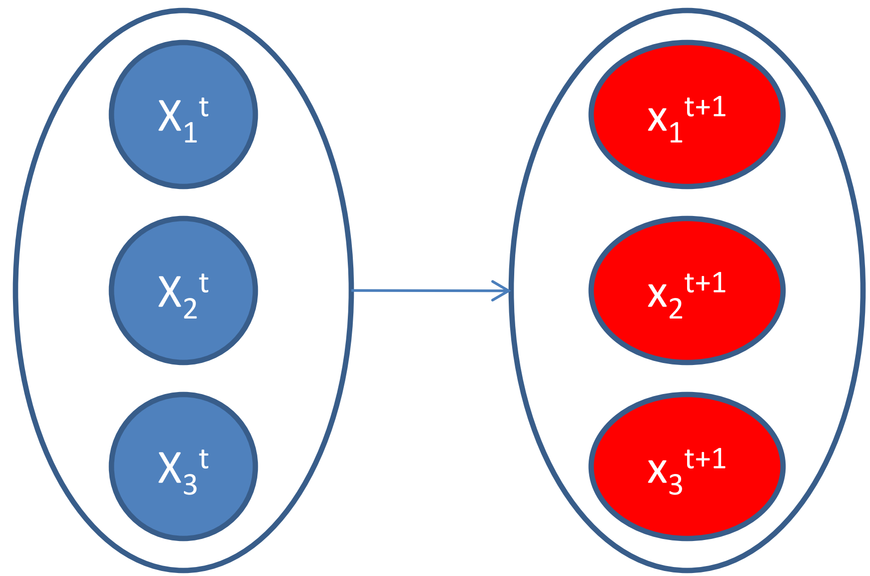


$$x^t = \{x_1^t, x_2^t, \ldots, x_n^t\}$$

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

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

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

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)

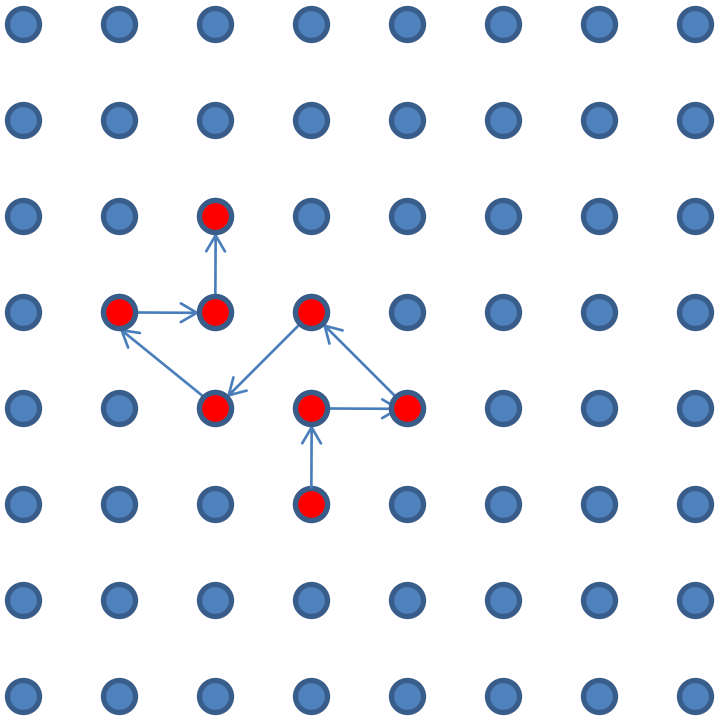
- Gibbs sampler is an algorithm to generate a sequence of samples from the **joint probability distribution** of two or more random variables
- Sample new variable value one variable at a time from the variable's conditional distribution:

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

- 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

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

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

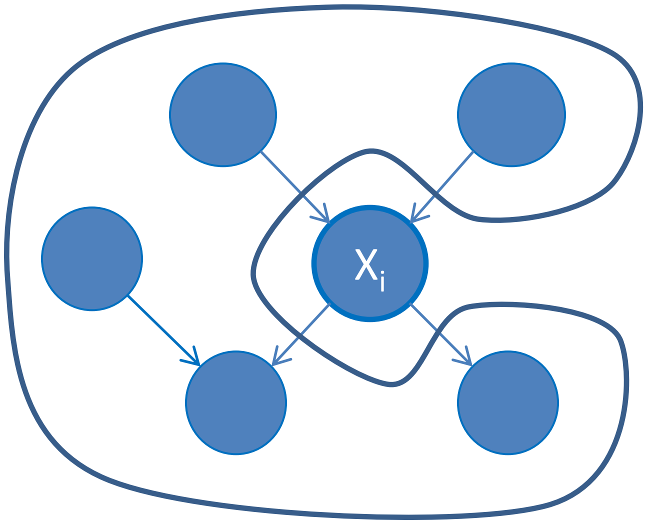
...

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

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*

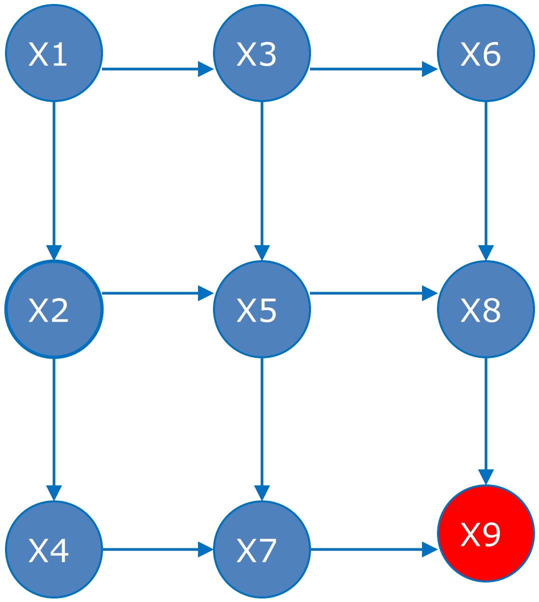
Output: *T* samples $\{x^t\}$

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

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

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



$X_1 = x_1^0$

$X_6 = x_6^0$

$X_2 = x_2^0$

$X_7 = x_7^0$

$X_3 = x_3^0$

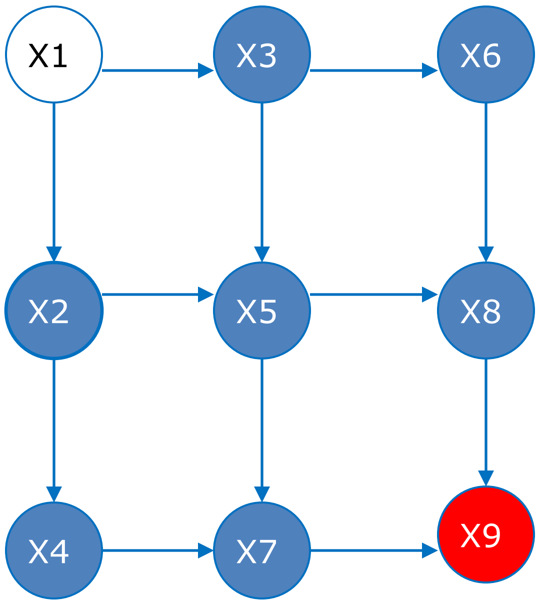
$X_8 = x_8^0$

$X_4 = x_4^0$

$X_5 = x_5^0$

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 $P(x_i | e) = ?$

- **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 $\bar{P}(X|e) \rightarrow P(X|e)$
- 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

Outline

- Rejection problem
- Backtrack-free distribution
  - Constructing it in practice
- SampleSearch
  - Construct the backtrack-free distribution on the fly.
- Approximate estimators
- Experiments

Outline

- **Rejection problem**
- Backtrack-free distribution
  - Constructing it in practice
- SampleSearch
  - Construct the backtrack-free distribution on the fly.
- Approximate estimators
- Experiments

Rejection problem

$$\hat{P}(e) = \frac{1}{N}\sum_{i=1}^{N}\frac{P(z^i, e)}{Q(z^i)}$$

- Importance sampling requirement
  - P(z,e) > 0 → Q(z)>0
- When P(z,e)=0 but Q(z) > 0, the weight of the sample is zero and it is rejected.
- The probability of generating a rejected sample should be very small.
  - Otherwise the estimate will be zero.

Rejection Problem

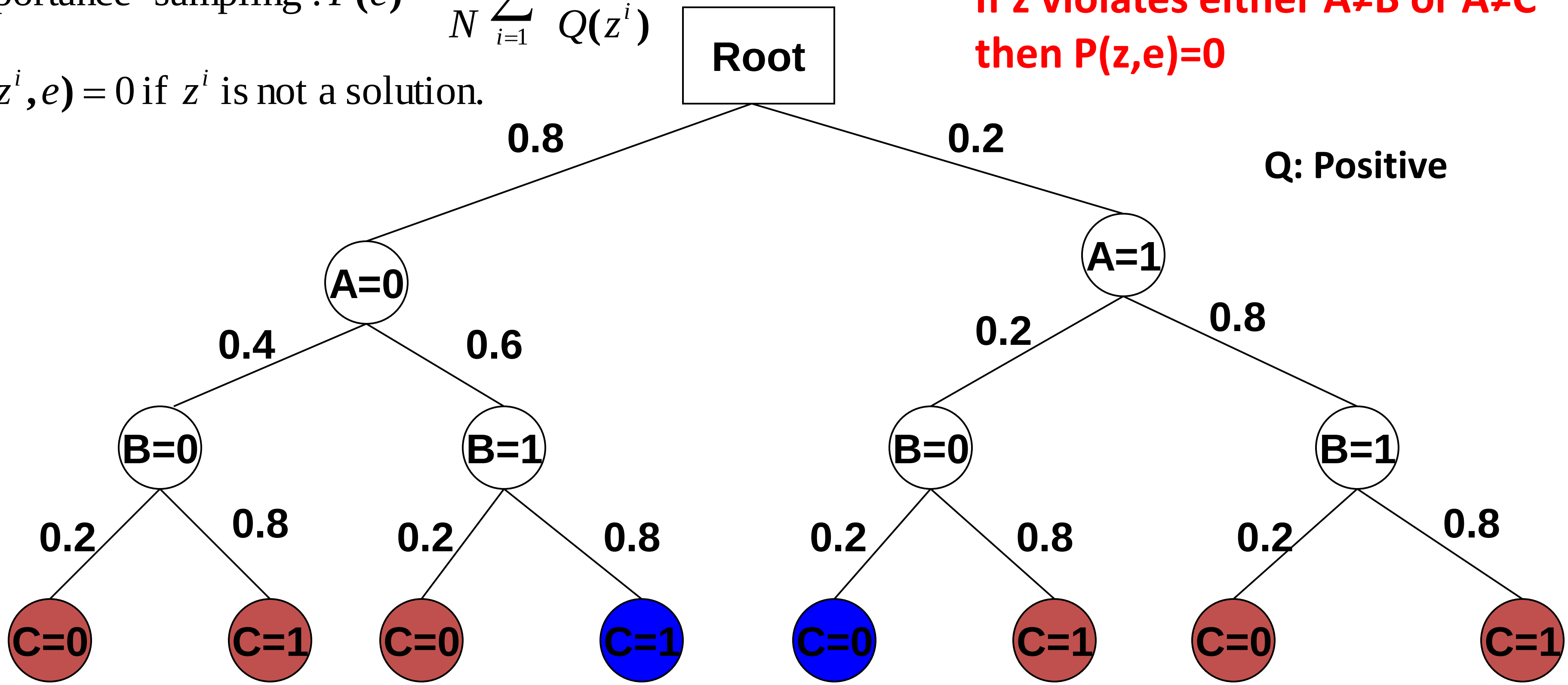
Importance sampling : $\hat{P}(e) = \frac{1}{N}\sum_{i=1}^{N}\frac{P(z^i, e)}{Q(z^i)}$

$P(z^i, e) = 0$ if z^i is not a solution.

Constraints: A≠B A≠C

If z violates either A≠B or A≠C then P(z,e)=0

Q: Positive



All Blue leaves correspond to solutions i.e. g(x) >0

All Red leaves correspond to non-solutions i.e. g(x)=0

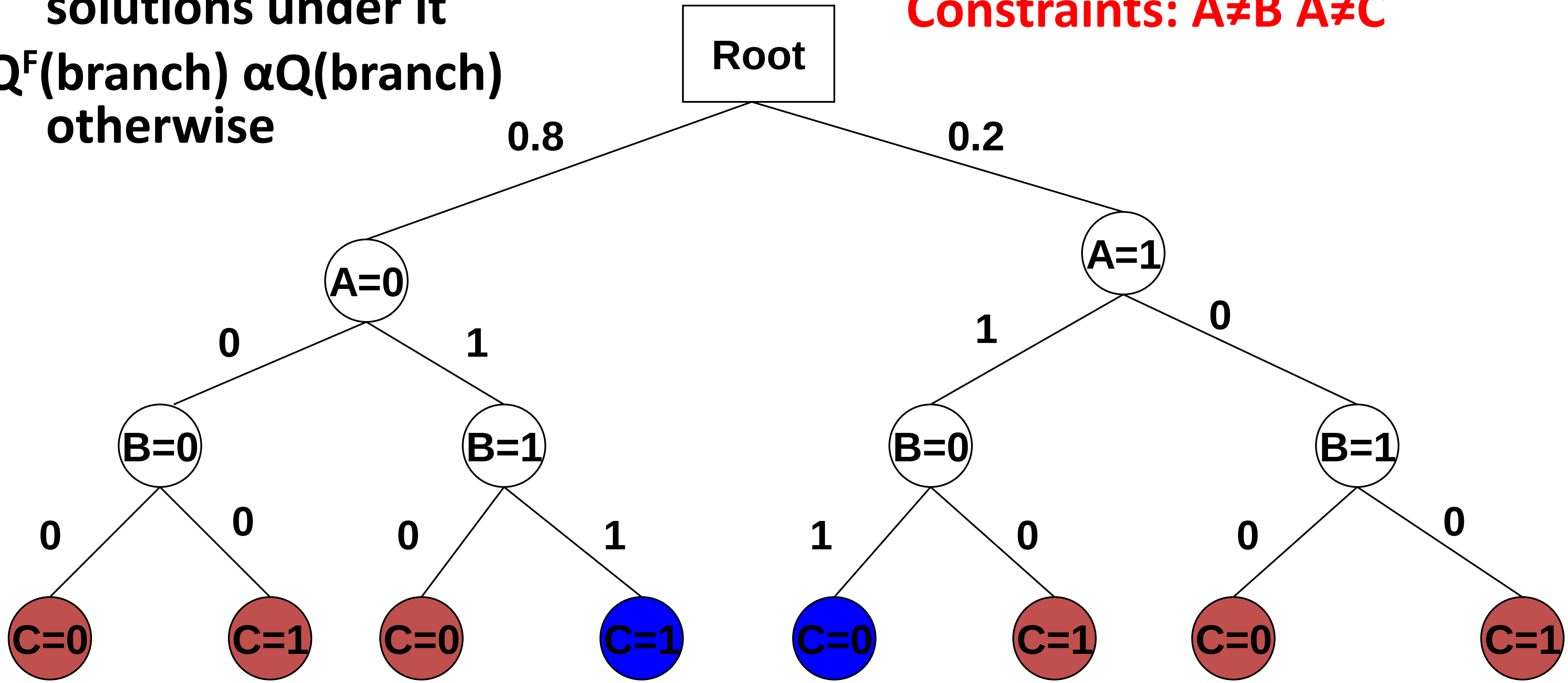
Outline

- Rejection problem
- **Backtrack-free distribution**
  - Constructing it in practice
- SampleSearch
  - Construct the backtrack-free distribution on the fly.
- Approximate estimators
- Experiments

Backtrack-free distribution: A rejection-free distribution

- **Q^F(branch)=0 if no solutions under it**
- **Q^F(branch) αQ(branch) otherwise**

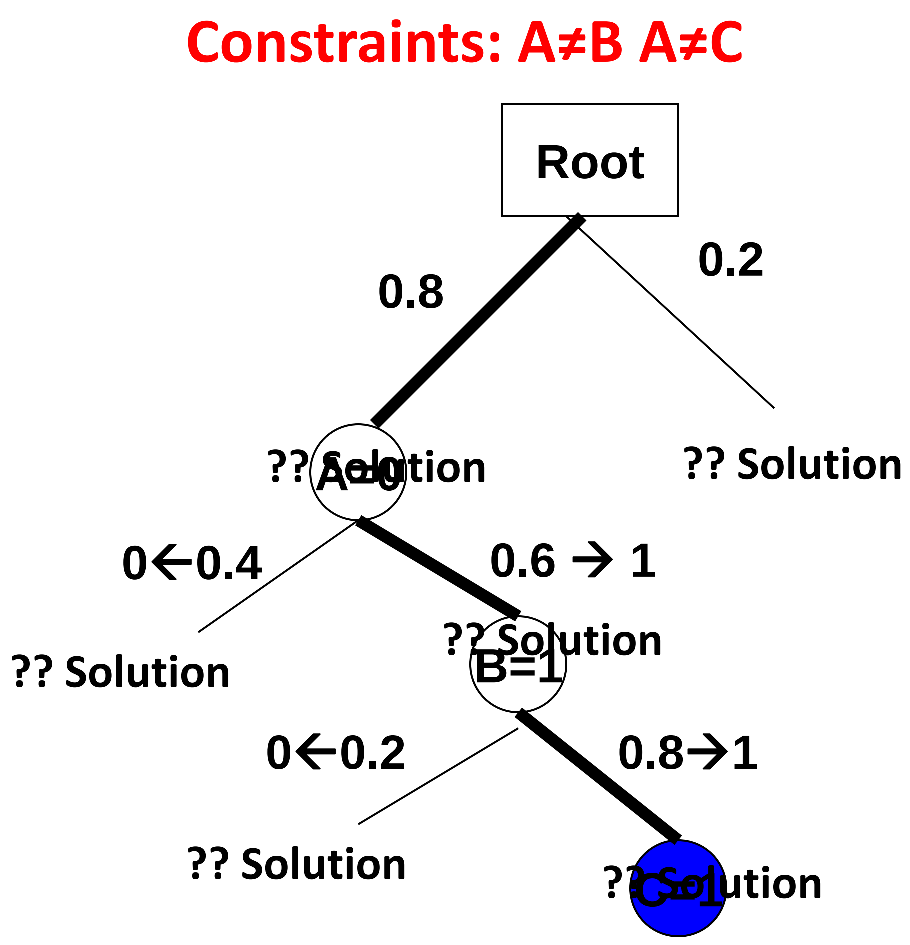
Constraints: A≠B A≠C



All Blue leaves correspond to solutions i.e. g(x) >0

All Red leaves correspond to non-solutions i.e. g(x)=0

Generating samples from Q^F



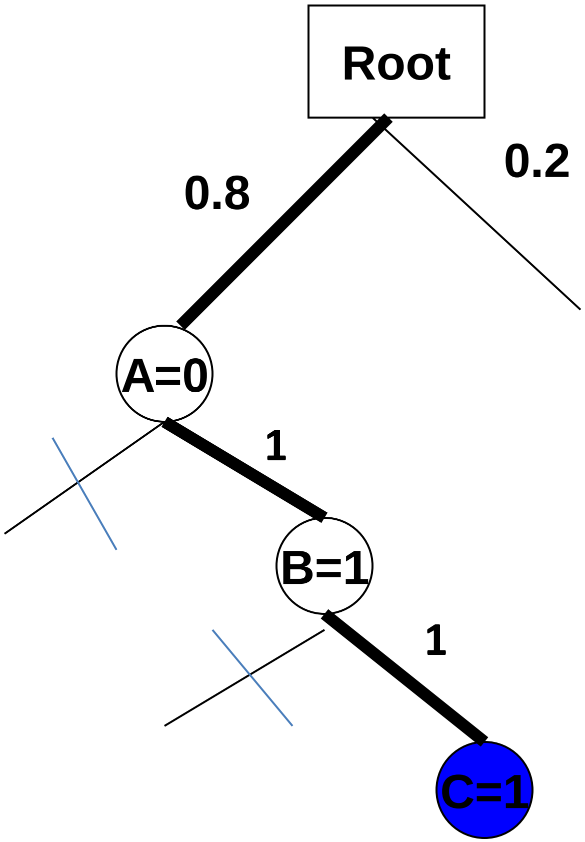
Q^F(branch)=0 if no solutions under it

Q^F(branch) αQ(branch) otherwise

- Invoke an oracle at each branch.
  - Oracle returns True if there is a solution under a branch
  - False, otherwise

Generating samples from Q^F

Constraints: A≠B A≠C



- Oracles
  - Adaptive consistency as pre-processing step
    - Constant time table look-up
    - Exponential in the treewidth of the constraint portion.
  - A complete CSP solver
    - Need to run it at each assignment.

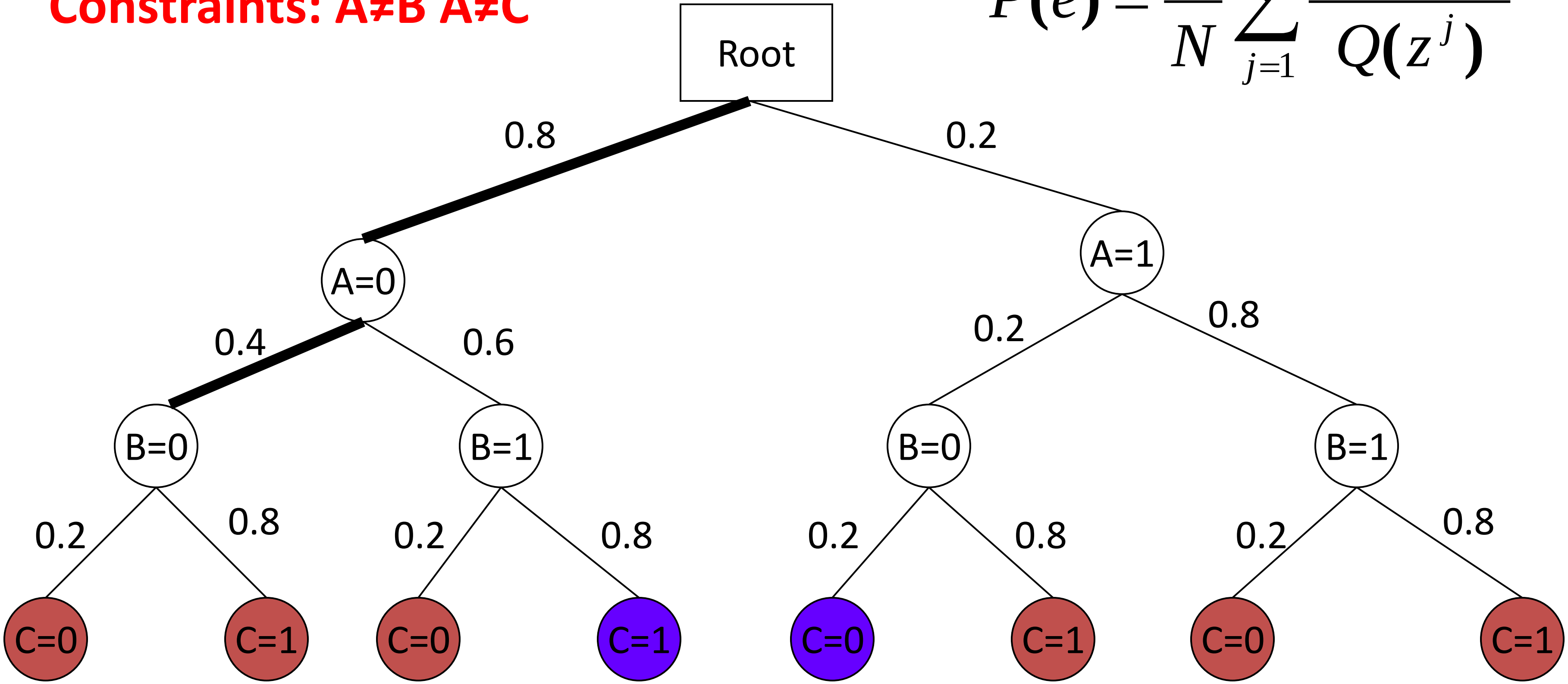
Outline

- Rejection problem
- Backtrack-free distribution
  - Constructing it in practice
- **SampleSearch**
  - **Construct the backtrack-free distribution on the fly.**
- Approximate estimators
- Experiments

Algorithm SampleSearch

$$\hat{P}(e) = \frac{1}{N}\sum_{j=1}^{N}\frac{P(z^j, e)}{Q(z^j)}$$

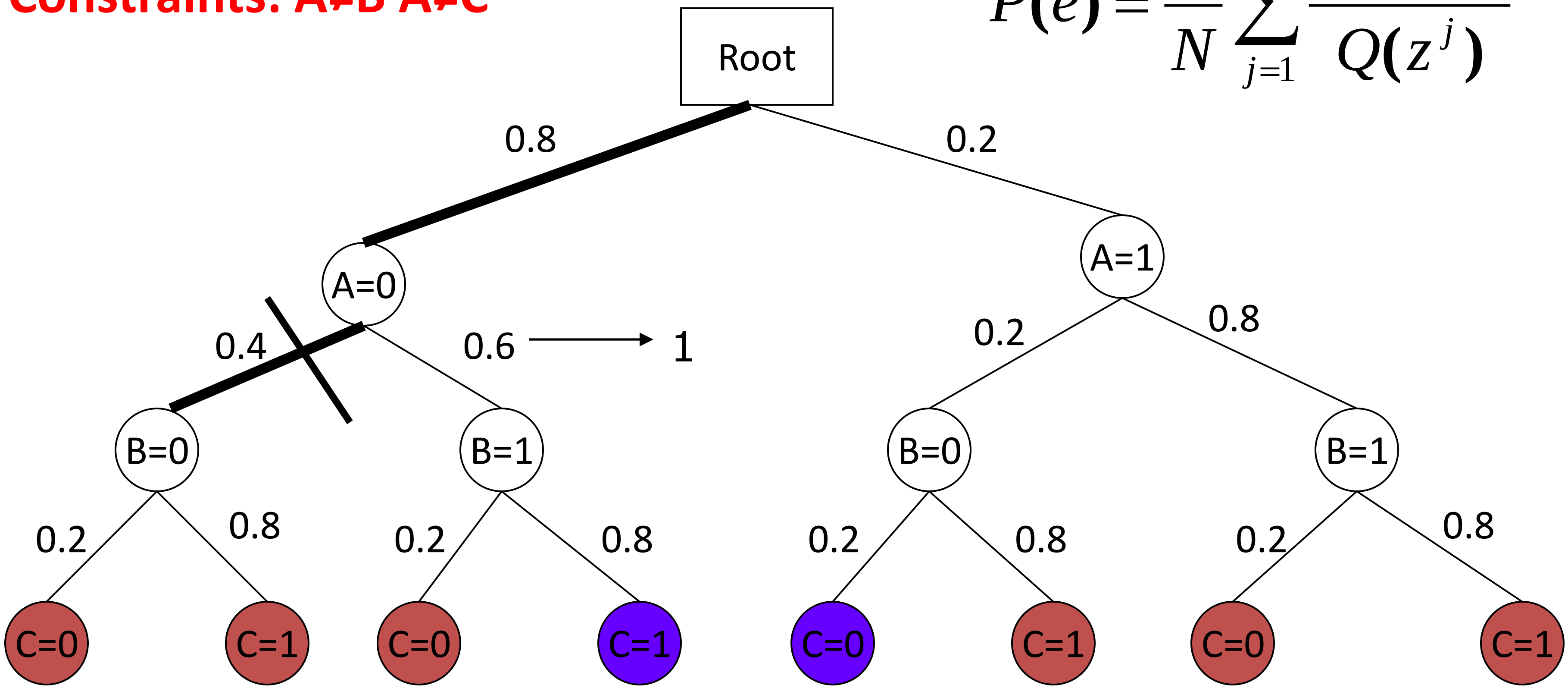
Constraints: A≠B A≠C



Algorithm SampleSearch

Constraints: A≠B A≠C

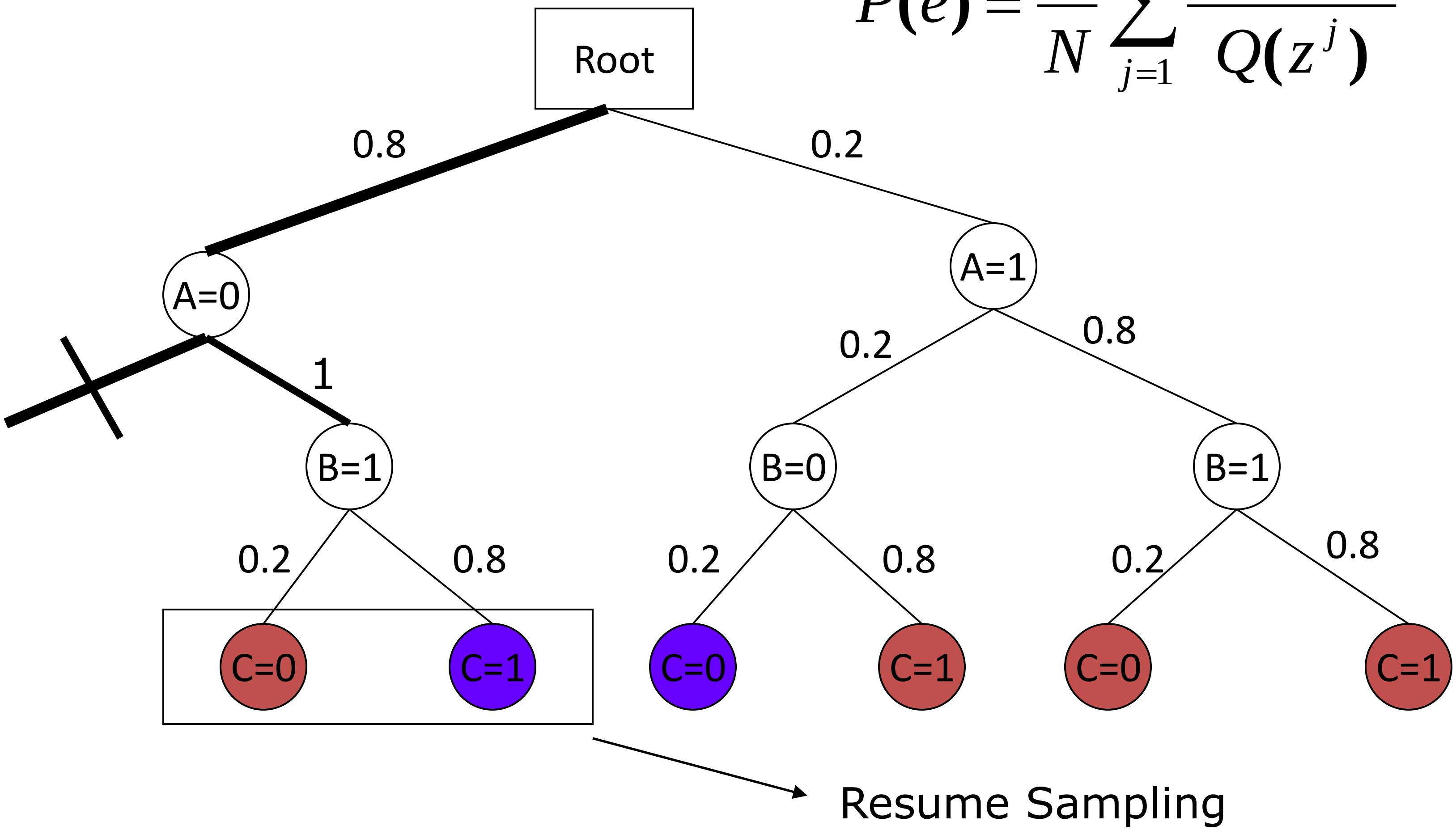
$$\hat{P}(e) = \frac{1}{N}\sum_{j=1}^{N}\frac{P(z^j, e)}{Q(z^j)}$$



Algorithm SampleSearch

Constraints: A≠B A≠C

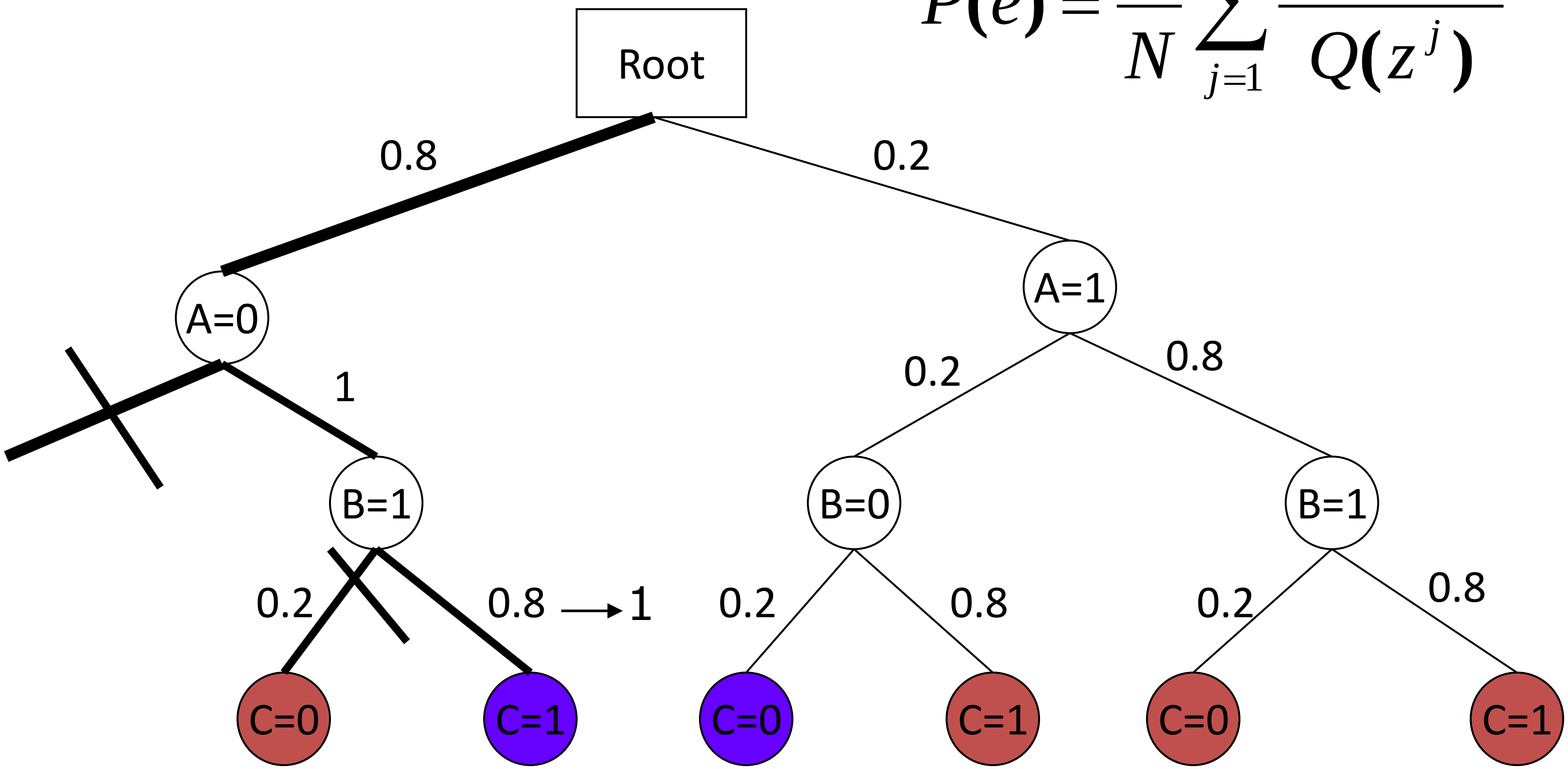
$$\hat{P}(e) = \frac{1}{N}\sum_{j=1}^{N}\frac{P(z^j, e)}{Q(z^j)}$$



Algorithm SampleSearch

Constraints: A≠B A≠C

$$\hat{P}(e) = \frac{1}{N}\sum_{j=1}^{N}\frac{P(z^j,e)}{Q(z^j)}$$



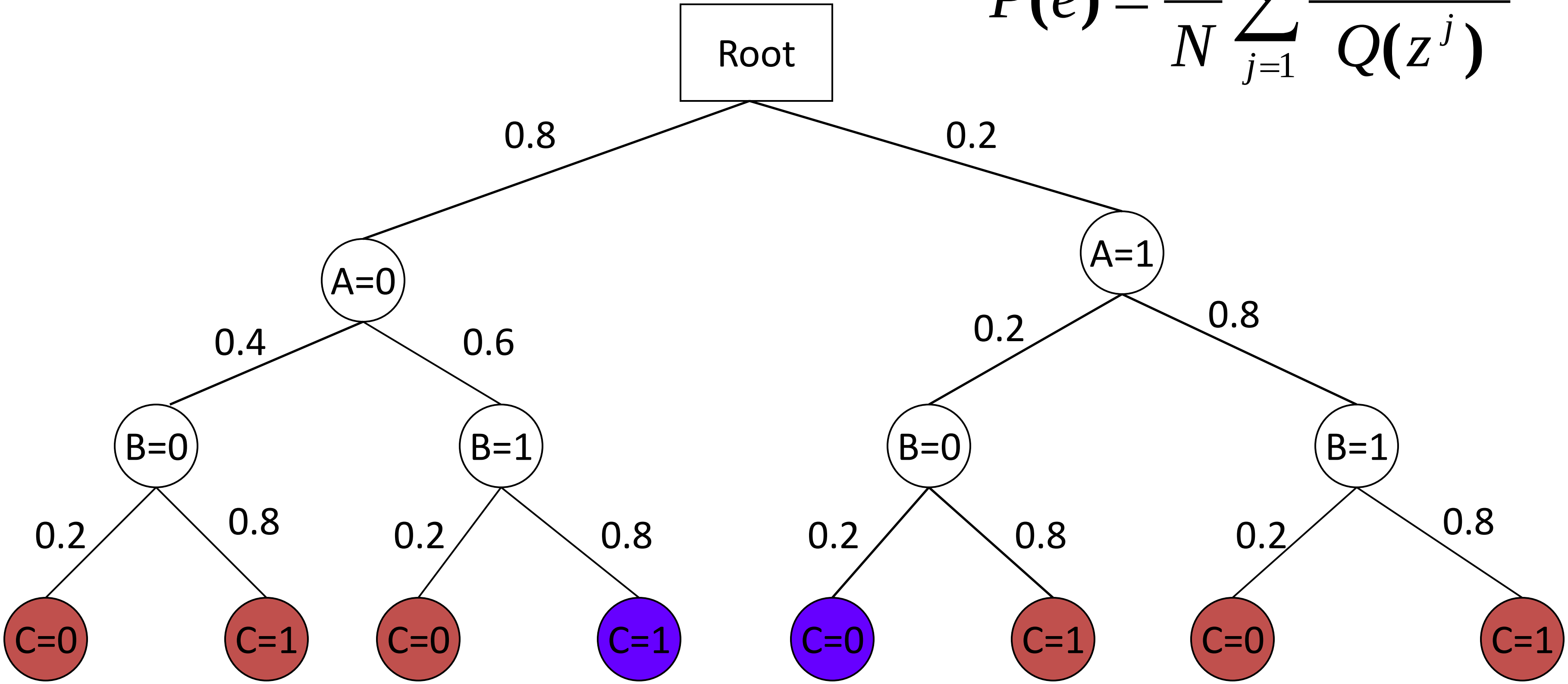
Until P(sample,e)>0

Constraint Violated

Generate more Samples

Constraints: A≠B A≠C

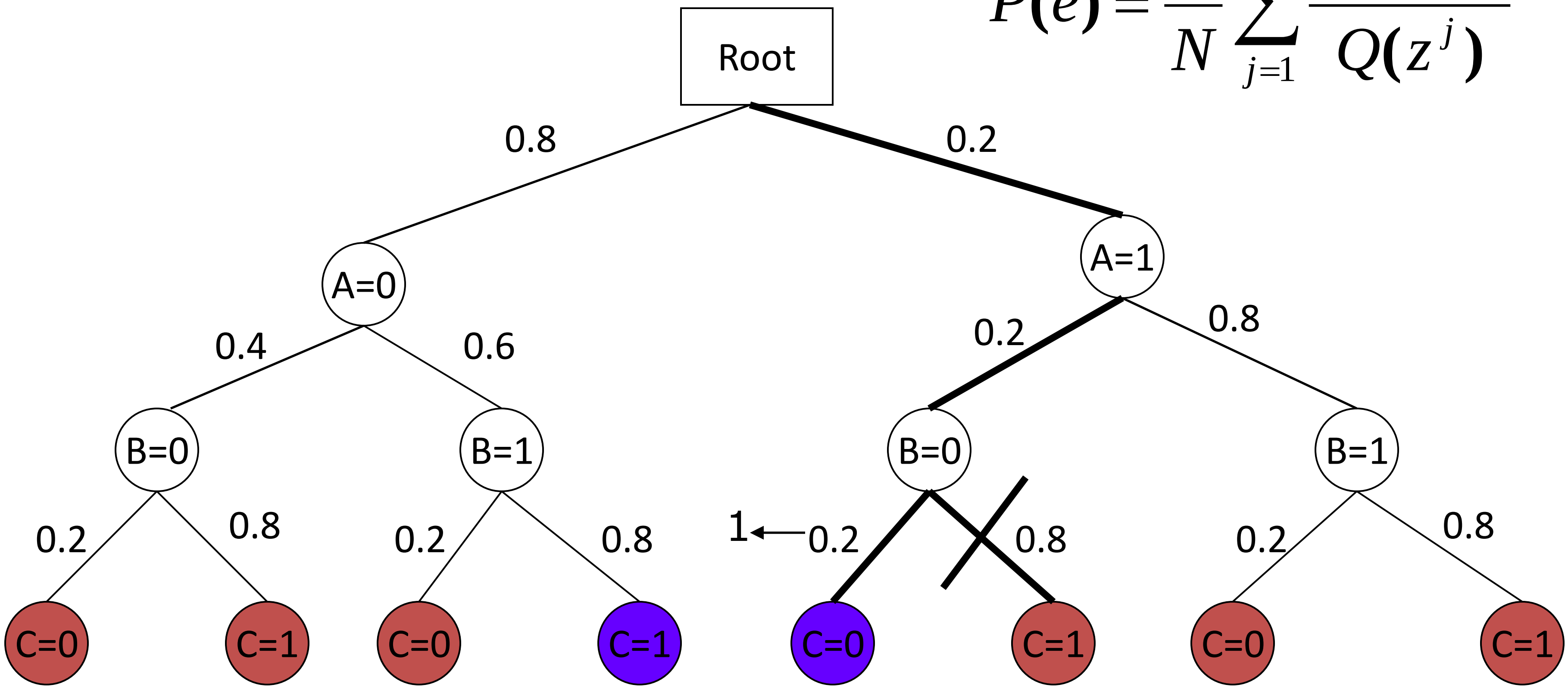
$$\hat{P}(e) = \frac{1}{N}\sum_{j=1}^{N}\frac{P(z^j, e)}{Q(z^j)}$$



Generate more Samples

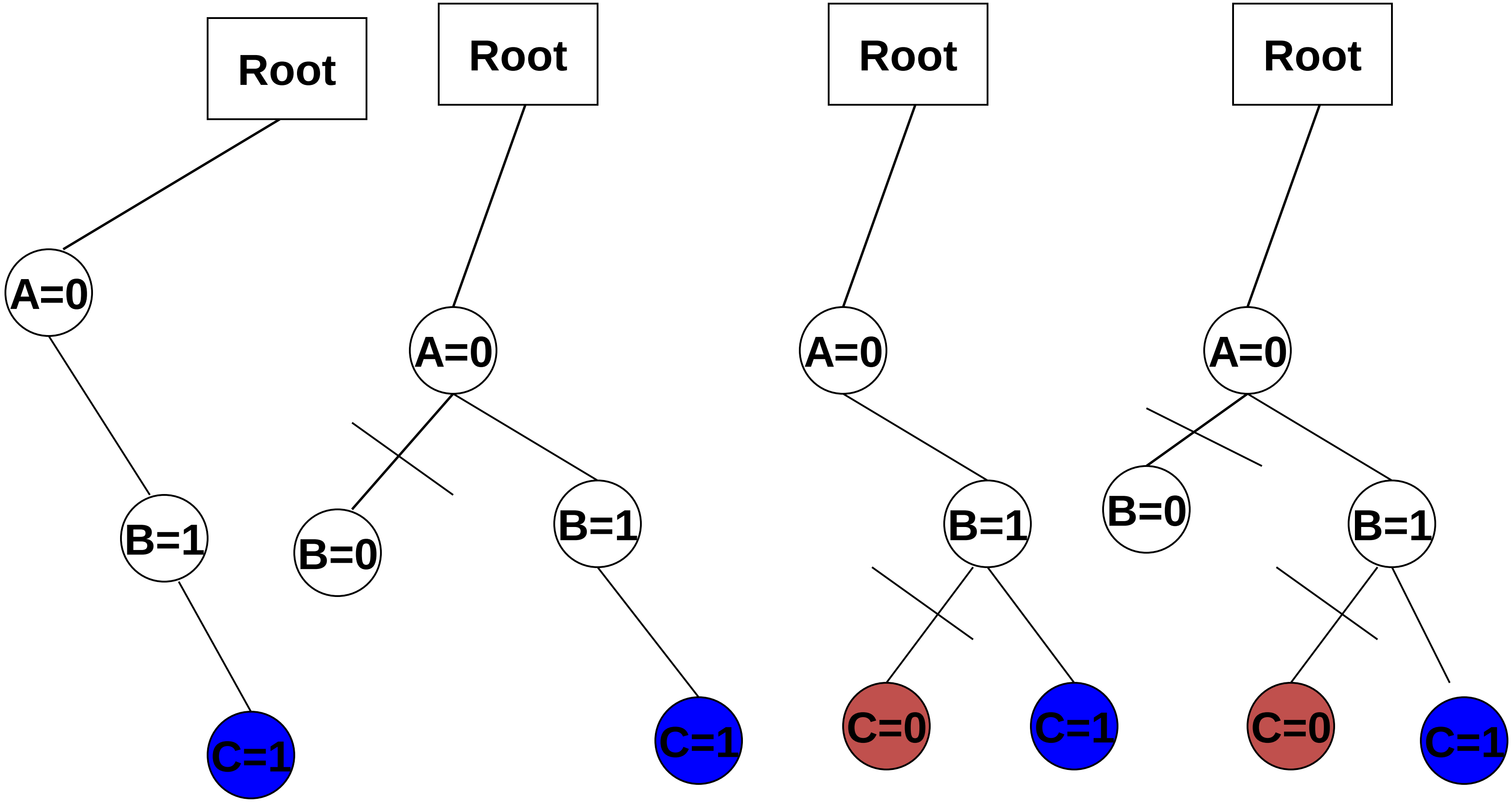
Constraints: A≠B A≠C

$$\hat{P}(e) = \frac{1}{N}\sum_{j=1}^{N}\frac{P(z^j, e)}{Q(z^j)}$$



Traces of SampleSearch

Constraints: A≠B A≠C



SampleSearch: Sampling Distribution

- Problem: Due to Search, the samples are no longer i.i.d. from Q.

$$\overline{P}(e) = \frac{1}{N}\sum_{j=1}^{N}\frac{P(z^j,e)}{Q(z^j)}, \quad E_Q\left[\overline{P}(e)\right] \neq P(e)$$

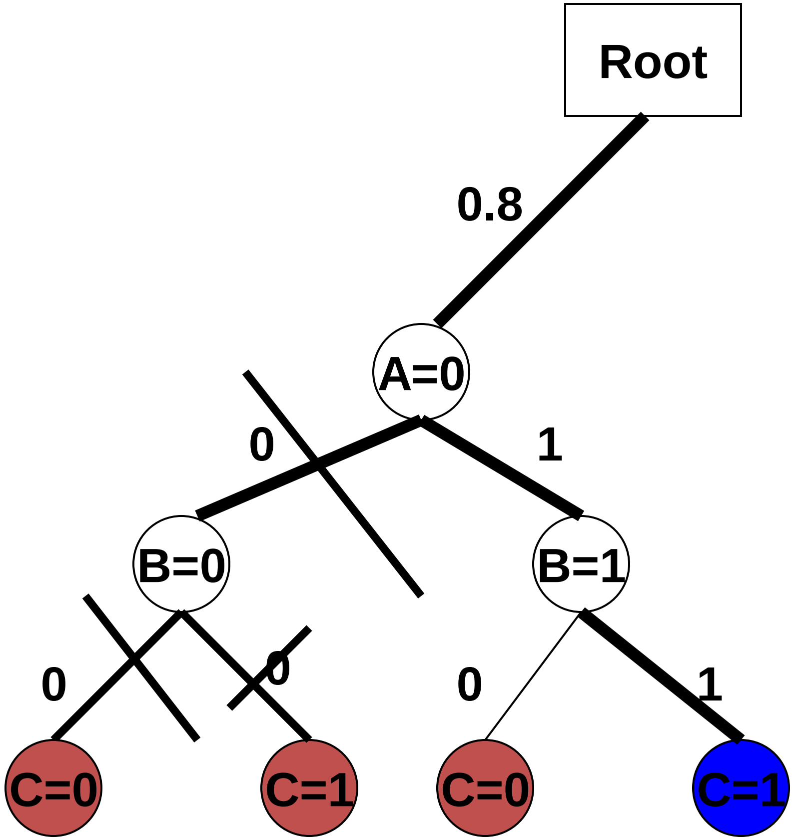
- Thm: SampleSearch generates i.i.d. samples from the **backtrack-free distribution**

$$\hat{P}_F(e) = \frac{1}{N}\sum_{j=1}^{N}\frac{P(z^j,e)}{Q^F(z^j)}, \quad E_{Q^F}\left[\hat{P}_F(e)\right] = P(e)$$

The Sampling distribution Q^F of SampleSearch

$$\hat{P}(e) = \frac{1}{N}\sum_{j=1}^{N}\frac{P(z^j, e)}{Q(z^j)}$$

Constraints: A≠B A≠C



Backtrack-free distribution

What is probability of generating A=0?

$Q^F(A=0)=0.8$

Why? SampleSearch is systematic

What is probability of generating (A=0,B=1)?

$Q^F(B=1|A=0)=1$

Why? SampleSearch is systematic

What is probability of generating (A=0,B=0)?

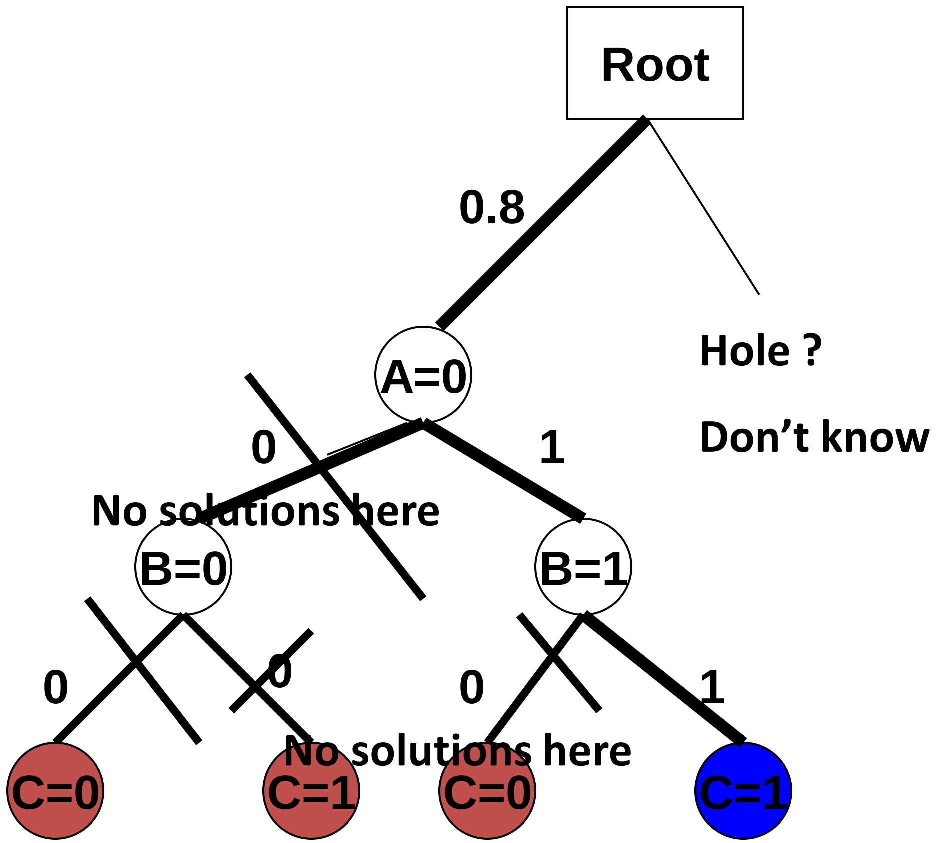
Simple: $Q^F(B=0|A=0)=0$

All samples generated by SampleSearch are solutions

Outline

- Rejection problem
- Backtrack-free distribution
  - Constructing it in practice
- SampleSearch
  - Construct the backtrack-free distribution on the fly.
- **Approximate estimators**
- Experiments

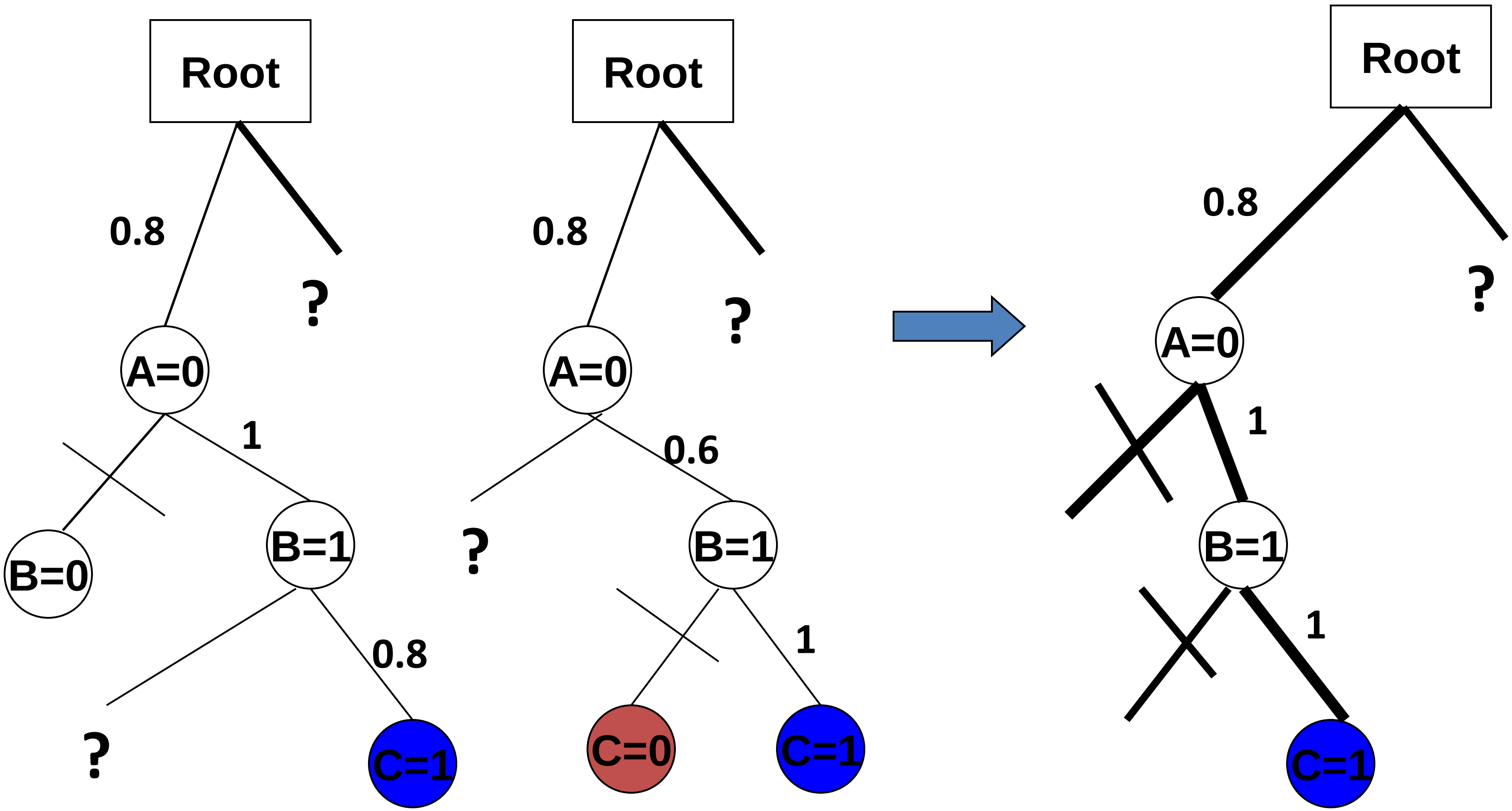
Asymptotic approximations of Q^F



- IF Hole THEN
  - U^F=Q (i.e. there is a solution at the other branch)
  - L^F=0 (i.e. no solution at the other branch)

Approximations: Convergence in the limit

- Store all possible traces



Approximations: Convergence in the limit

- **From the combined sample tree, update U and L.**

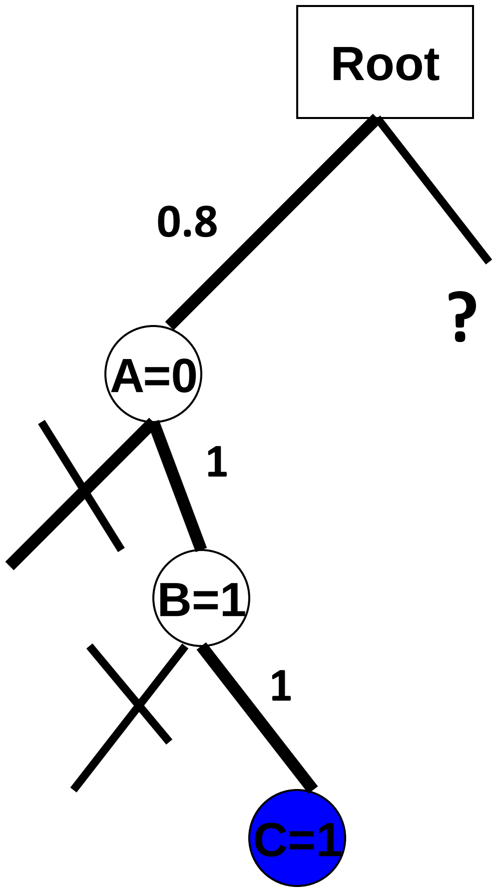
IF Hole THEN U^F_N=Q and L^F_N=0

$$\lim_{N\to\infty} E\left[\frac{P(z,e)}{U_N^F(z)}\right] = \lim_{N\to\infty} E\left[\frac{P(z,e)}{L_N^F(z)}\right] = P(e)$$

Asymptotically unbiased

Bounding : $U_N^F(z) \le Q^F(z) \le L_N^F(z)$

$$\overline{P}_F^U(e) \ge \hat{P}_F(e) \ge \overline{P}_F^L(e)$$



Upper and Lower Approximations

- Asymptotically unbiased.
- Upper and lower bound on the unbiased sample mean
- Linear time and space overhead
- Bias versus variance tradeoff
  - Bias = difference between the upper and lower approximation.

Improving Naive SampleSeach

- Better Search Strategy
  - Can use any state-of-the-art CSP/SAT solver e.g. minisat (Een and Sorrenson 2006)
    - All theorems and result hold
- Better Importance Function
  - Use output of generalized belief propagation to compute the initial importance function Q (Gogate and Dechter, 2005)

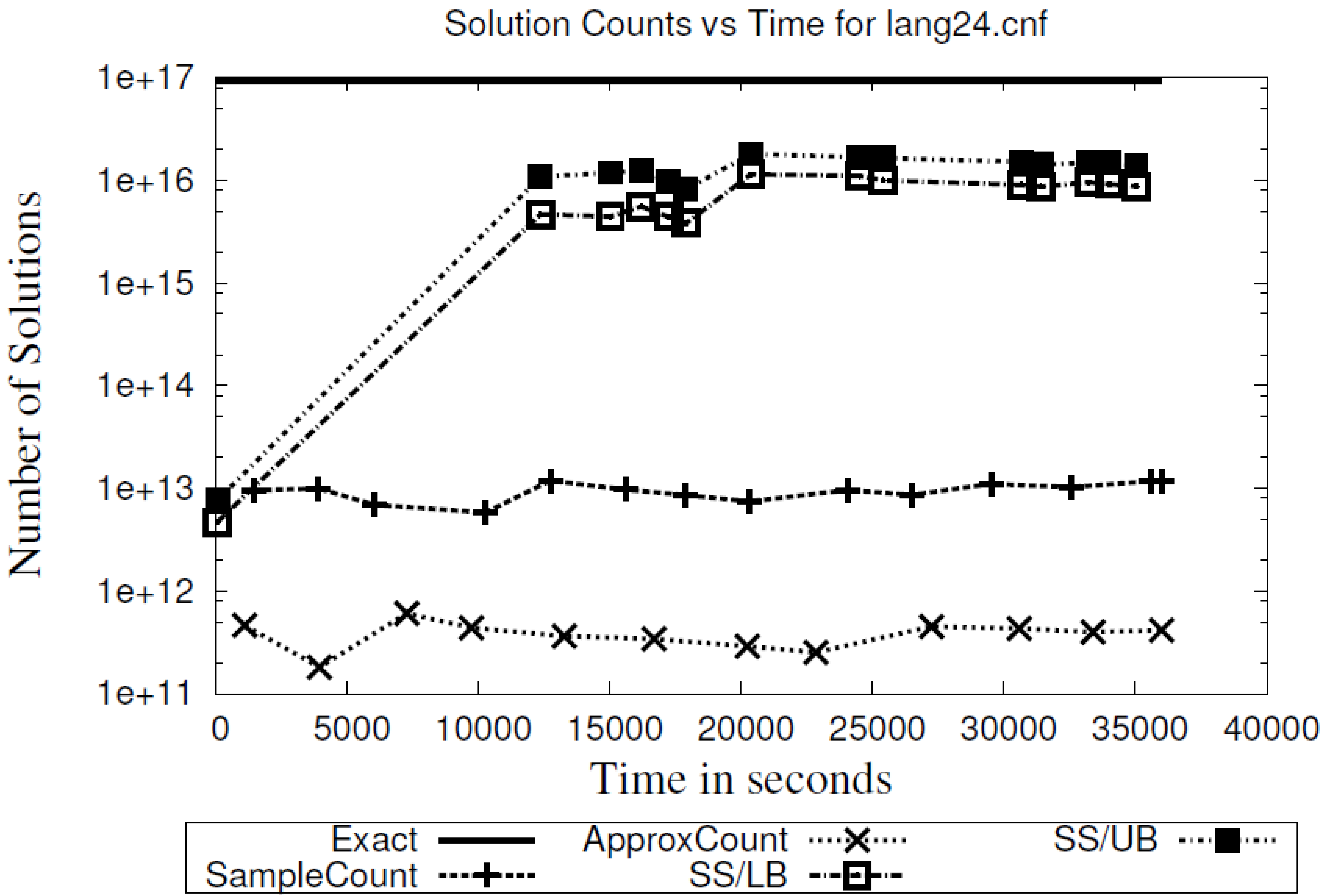
Experiments

- Tasks
  - Weighted Counting
  - Marginals
- Benchmarks
  - Satisfiability problems (counting solutions)
  - Linkage networks
  - Relational instances (First order probabilistic networks)
  - Grid networks
  - Logistics planning instances
- Algorithms
  - SampleSearch/UB, SampleSearch/LB
  - SampleCount (Gomes et al. 2007, SAT)
  - ApproxCount (Wei and Selman, 2007, SAT)
  - RELSAT (Bayardo and Peshoueshk, 2000, SAT)
  - Edge Deletion Belief Propagation (Choi and Darwiche, 2006)
  - Iterative Join Graph Propagation (Dechter et al., 2002)
  - Variable Elimination and Conditioning (VEC)
  - EPIS (Changhe and Druzdzel, 2006)

Results: Solution Counts Langford instances

Problem	$\langle n, k, c, w \rangle$	Exact	Sample Count	Approx Count	REL SAT	SS /LB	SS /UB
lang12	$\langle 576, 2, 13584, 383 \rangle$	2.16E+5	1.93E+05	2.95E+04	**2.16E+05**	2.16E+05	2.16E+05
lang16	$\langle 1024, 2, 32320, 639 \rangle$	6.53E+08	5.97E+08	8.22E+06	6.28E+06	**6.51E+08**	6.99E+08
lang19	$\langle 1444, 2, 54226, 927 \rangle$	5.13E+11	9.73E+10	6.87E+08	8.52E+05	**6.38E+11**	7.31E+11
lang20	$\langle 1600, 2, 63280, 1023 \rangle$	5.27E+12	1.13E+11	3.99E+09	8.55E+04	2.83E+12	**3.45E+12**
lang23	$\langle 2116, 2, 96370, 1407 \rangle$	7.60E+15	7.53E+14	3.70E+12	X	4.17E+15	**4.19E+15**
lang24	$\langle 2304, 2, 109536, 1535 \rangle$	9.37E+16	1.17E+13	4.15E+11	X	8.74E+15	**1.40E+16**
lang27	$\langle 2916, 2, 156114, 1919 \rangle$		4.38E+16	1.32E+14	X	**2.41E+19**	2.65E+19

Time Bound: 10 hrs


Solution Counts vs Time for lang24.cnf
Number of Solutions
1e+17
1e+16
1e+15
1e+14
1e+13
1e+12
1e+11
0
5000
10000
15000
20000
25000
30000
35000
40000
Time in seconds
Exact
ApproxCount
SS/UB
SampleCount
SS/LB

Results: Probability of Evidence Linkage instances (UAI 2006 evaluation)

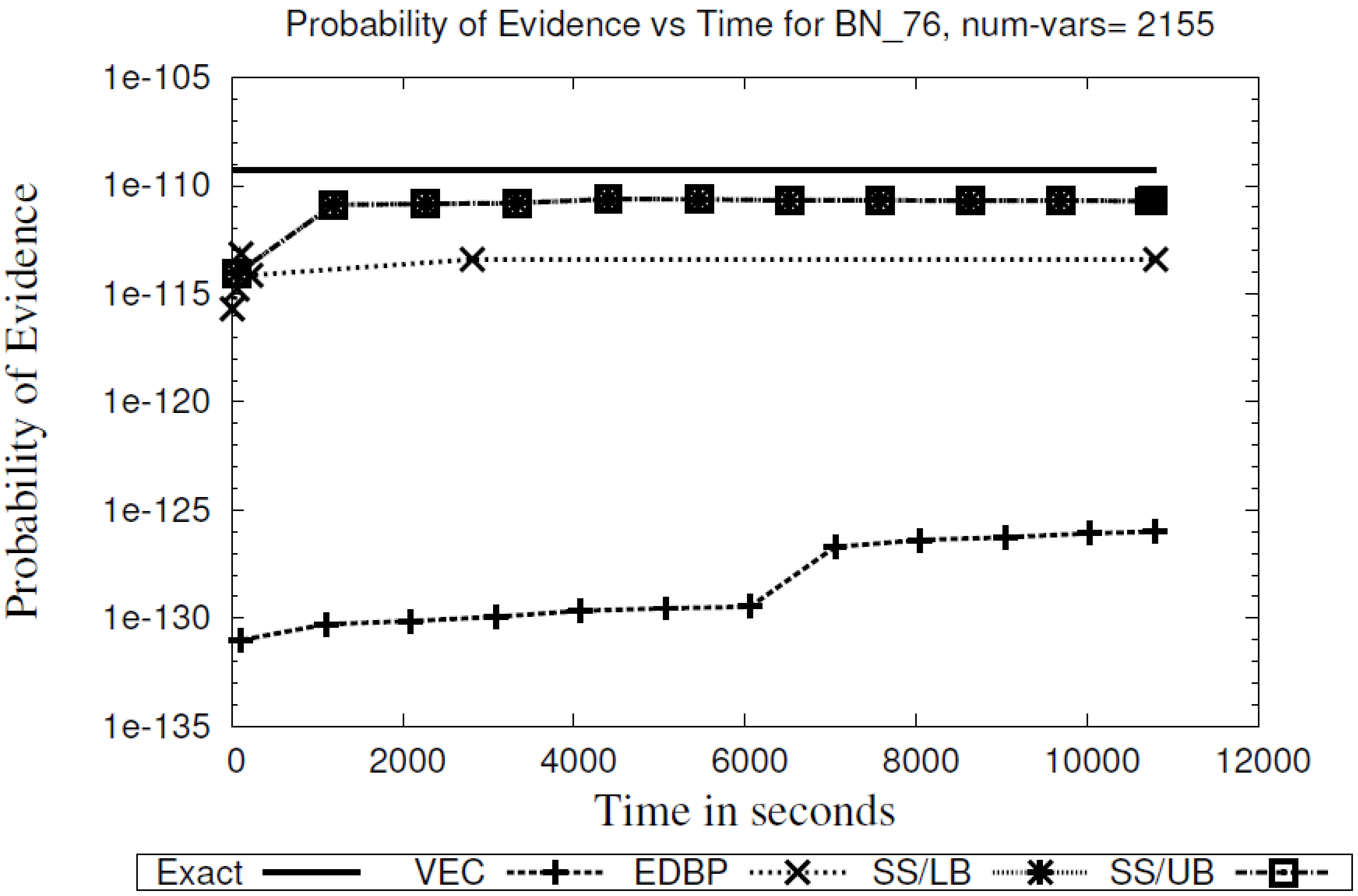
Problem	$\langle n, k, e, w \rangle$	Exact	VEC	EDBP	SS/LB	SS/UB
BN_69	$\langle 777, 7, 78, 47 \rangle$	5.28E-054	1.93E-61	2.39E-57	**3.00E-55**	3.00E-55
BN_70	$\langle 2315, 5, 159, 87 \rangle$	2.00E-71	7.99E-82	6.00E-79	**1.21E-73**	1.21E-73
BN_71	$\langle 1740, 6, 202, 70 \rangle$	5.12E-111	7.05E-115	1.01E-114	**1.28E-111**	1.28E-111
BN_72	$\langle 2155, 6, 252, 86 \rangle$	4.21E-150	1.32E-153	9.21E-155	**4.73E-150**	4.73E-150
BN_73	$\langle 2140, 5, 216, 101 \rangle$	2.26E-113	6.00E-127	2.24E-118	**2.00E-115**	2.00E-115
BN_74	$\langle 749, 6, 66, 45 \rangle$	3.75E-45	3.30E-48	5.84E-48	**2.13E-46**	2.13E-46
BN_75	$\langle 1820, 5, 155, 92 \rangle$	5.88E-91	5.83E-97	3.10E-96	**2.19E-91**	2.19E-91
BN_76	$\langle 2155, 7, 169, 64 \rangle$	4.93E-110	1.00E-126	3.86E-114	**1.95E-111**	1.95E-111

Time Bound: 3 hrs

Results: Probability of Evidence Linkage instances (UAI 2008 evaluation)

Problem	$\langle n, k, e, w \rangle$	Exact	SS/LB	SS/UB	VEC	EDBP
pedigree18	$\langle 1184, 1, 0, 26 \rangle$	7.18E-79	7.39E-79	7.39E-79	**7.18E-79***	**7.18E-79***
pedigree1	$\langle 334, 2, 0, 20 \rangle$	7.81E-15	7.81E-15	7.81E-15	**7.81E-15**	**7.81E-15***
pedigree20	$\langle 437, 2, 0, 25 \rangle$	2.34E-30	2.31E-30	2.31E-30	**2.34E-30***	6.19E-31
pedigree23	$\langle 402, 1, 0, 26 \rangle$	2.78E-39	2.76E-39	2.76E-39	**2.78E-39***	1.52E-39
pedigree25	$\langle 1289, 1, 0, 38 \rangle$	1.69E-116	**1.69E-116**	1.69E-116	**1.69E-116***	**1.69E-116***
pedigree30	$\langle 1289, 1, 0, 27 \rangle$	1.84E-84	1.90E-84	1.90E-84	**1.85E-84***	**1.85E-84***
pedigree37	$\langle 1032, 1, 0, 25 \rangle$	2.63E-117	1.18E-117	1.18E-117	**2.63E-117***	5.69E-124
pedigree38	$\langle 724, 1, 0, 18 \rangle$	5.64E-55	3.80E-55	3.80E-55	**5.65E-55***	8.41E-56
pedigree39	$\langle 1272, 1, 0, 29 \rangle$	6.32E-103	6.29E-103	6.29E-103	**6.32E-103***	**6.32E-103***
pedigree42	$\langle 448, 2, 0, 23 \rangle$	1.73E-31	1.73E-31	1.73E-31	**1.73E-31***	8.91E-32
pedigree19	$\langle 793, 2, 0, 23 \rangle$		**6.76E-60**	**6.76E-60**	1.597E-60	3.35E-60
pedigree31	$\langle 1183, 2, 0, 45 \rangle$		**2.08E-70**	**2.08E-70**	1.67E-76	1.34E-70
pedigree34	$\langle 1160, 1, 0, 59 \rangle$		**3.84E-65**	**3.84E-65**	2.58E-76	4.30E-65
pedigree13	$\langle 1077, 1, 0, 51 \rangle$		**7.03E-32**	**7.03E-32**	2.17E-37	6.53E-32
pedigree40	$\langle 1030, 2, 0, 49 \rangle$		**1.25E-88**	**1.25E-88**	2.45E-91	7.02E-17
pedigree41	$\langle 1062, 2, 0, 52 \rangle$		**4.36E-77**	**4.36E-77**	4.33E-81	1.09E-10
pedigree44	$\langle 811, 1, 0, 29 \rangle$		**3.39E-64**	**3.39E-64**	2.23E-64	7.69E-66
pedigree51	$\langle 1152, 1, 0, 51 \rangle$		**2.47E-74**	**2.47E-74**	5.56E-85	6.16E-76
pedigree7	$\langle 1068, 1, 0, 56 \rangle$		**1.33E-65**	**1.33E-65**	1.66E-72	2.93E-66
pedigree9	$\langle 1118, 2, 0, 41 \rangle$		**2.93E-79**	**2.93E-79**	8.00E-82	3.13E-89

Time Bound: 3 hrs



Probability of Evidence vs Time for BN_76, num-vars= 2155
Probability of Evidence
Time in seconds
1e-105
1e-110
1e-115
1e-120
1e-125
1e-130
1e-135
0
2000
4000
6000
8000
10000
12000
Exact
VEC
EDBP
SS/LB
SS/UB

Results on Marginals

- Evaluation Criteria

$$Exact: P(x_i) \quad Approximate: A(x_i)$$

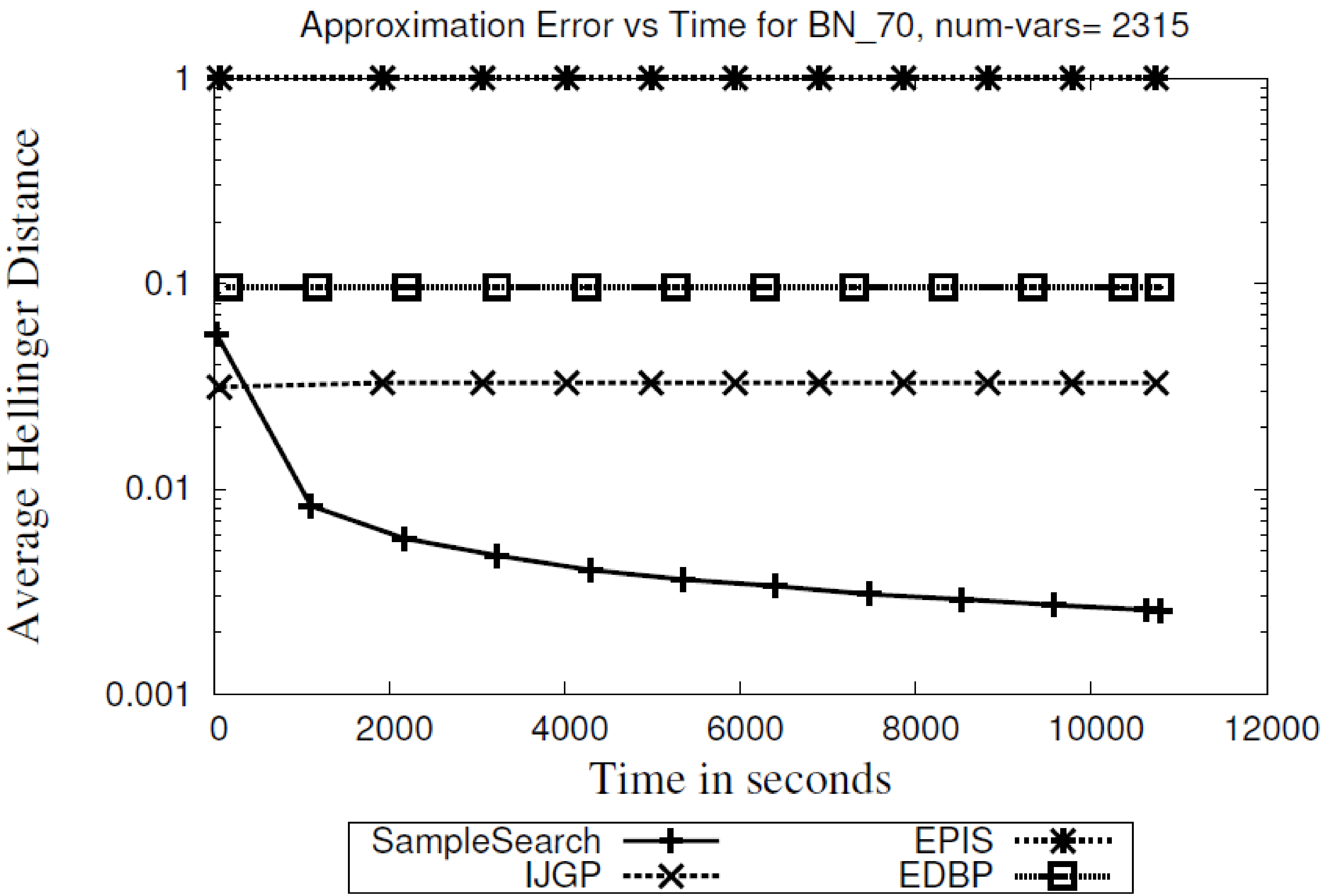
$$Hellinger\ distance = \frac{\sum_{i=1}^{n} \frac{1}{2} \sum_{x_i \in D_i} \left(\sqrt{P(x_i)} - \sqrt{A(x_i)}\right)^2}{n}$$

- Always bounded between 0 and 1
- Lower Bounds the KL distance
- When probabilities close to zero are present KL distance may tend to infinity.

Results: Posterior Marginals
Linkage instances (UAI 2006 evaluation)

Problem	$\langle n, K, e, w \rangle$	SampleSearch	IJGP	EPIS	EDBP
BN_69	$\langle 777, 7, 78, 47 \rangle$	**9.4E-04**	3.2E-02	1	8.0E-02
BN_70	$\langle 2315, 5, 159, 87 \rangle$	**2.6E-03**	3.3E-02	1	9.6E-02
BN_71	$\langle 1740, 6, 202, 70 \rangle$	**5.6E-03**	1.9E-02	1	2.5E-02
BN_72	$\langle 2155, 6, 252, 86 \rangle$	**3.6E-03**	7.2E-03	1	1.3E-02
BN_73	$\langle 2140, 5, 216, 101 \rangle$	**2.1E-02**	2.8E-02	1	6.1E-02
BN_74	$\langle 749, 6, 66, 45 \rangle$	6.9E-04	**4.3E-06**	1	4.3E-02
BN_75	$\langle 1820, 5, 155, 92 \rangle$	**8.0E-03**	6.2E-02	1	9.3E-02
BN_76	$\langle 2155, 7, 169, 64 \rangle$	**1.8E-02**	2.6E-02	1	2.7E-02

Time Bound: 3 hrs
Distance measure: Hellinger distance


Approximation Error vs Time for BN_70, num-vars= 2315
Average Hellinger Distance
1
0.1
0.01
0.001
0
2000
4000
6000
8000
10000
12000
Time in seconds
SampleSearch
IJGP
EPIS
EDBP

Summary: SampleSearch

- Manages rejection problem while sampling
  - Systematic backtracking search
- Sampling Distribution of SampleSearch is the backtrack-free distribution Q^F
  - Expensive to compute
- Approximation of Q^F based on storing all traces that yields an asymptotically unbiased estimator
  - Linear time and space overhead
  - Bound the sample mean from above and below
- Empirically, when a substantial number of zero probabilities are present, SampleSearch based schemes dominate their pure sampling counter-parts and Generalized Belief Propagation.

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

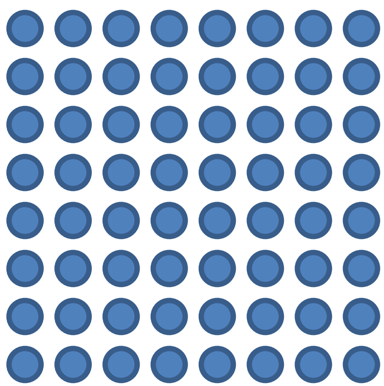
Sampling: Performance

- 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

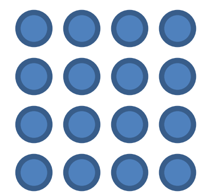
- Smaller state-space is easier to cover

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



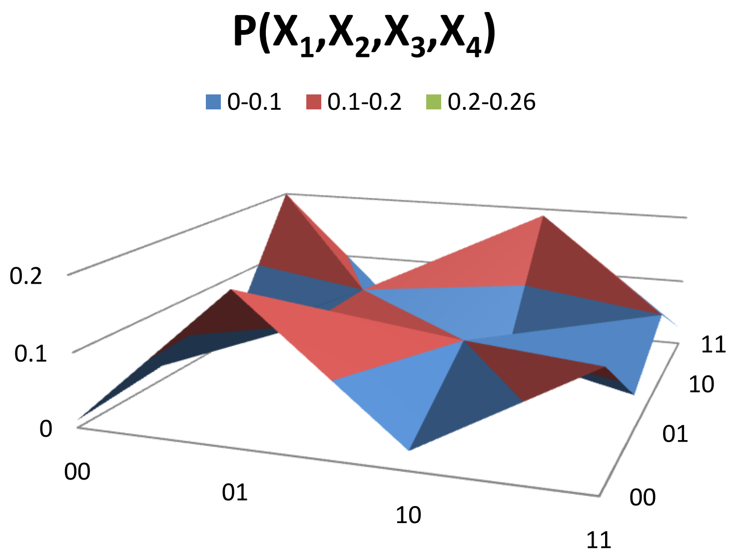
$$D(X) = 64$$

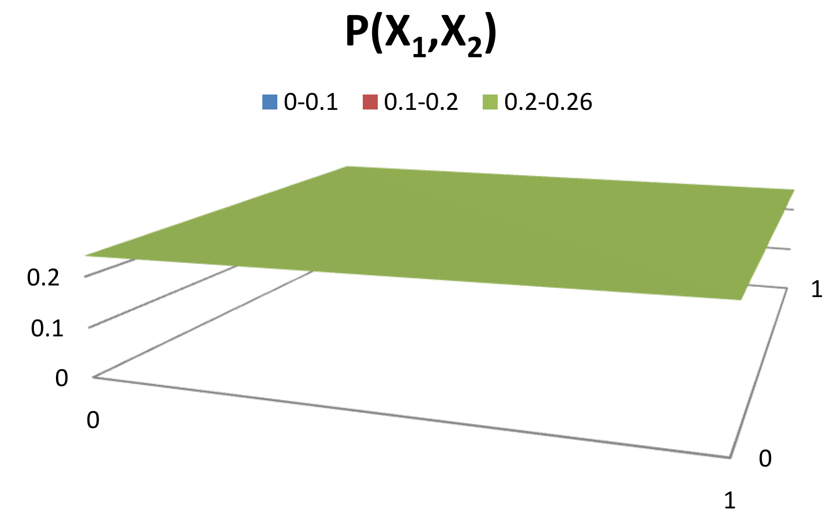
$$X = \{X_1, X_2\}$$



$$D(X) = 16$$

Smoother Distribution





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

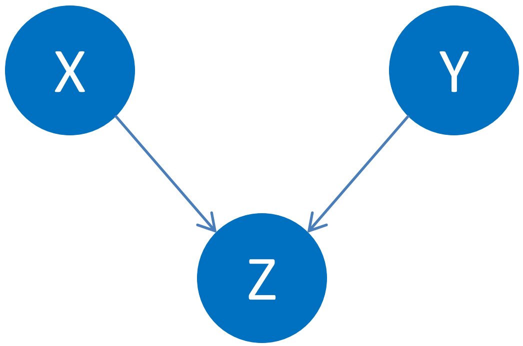
- $X = R \cup L$
- Importance Sampling:

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

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 fo them into a sampling set

Blocking Gibbs Sampler vs. Collapsed



Faster Convergence

- Standard Gibbs:

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

- 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

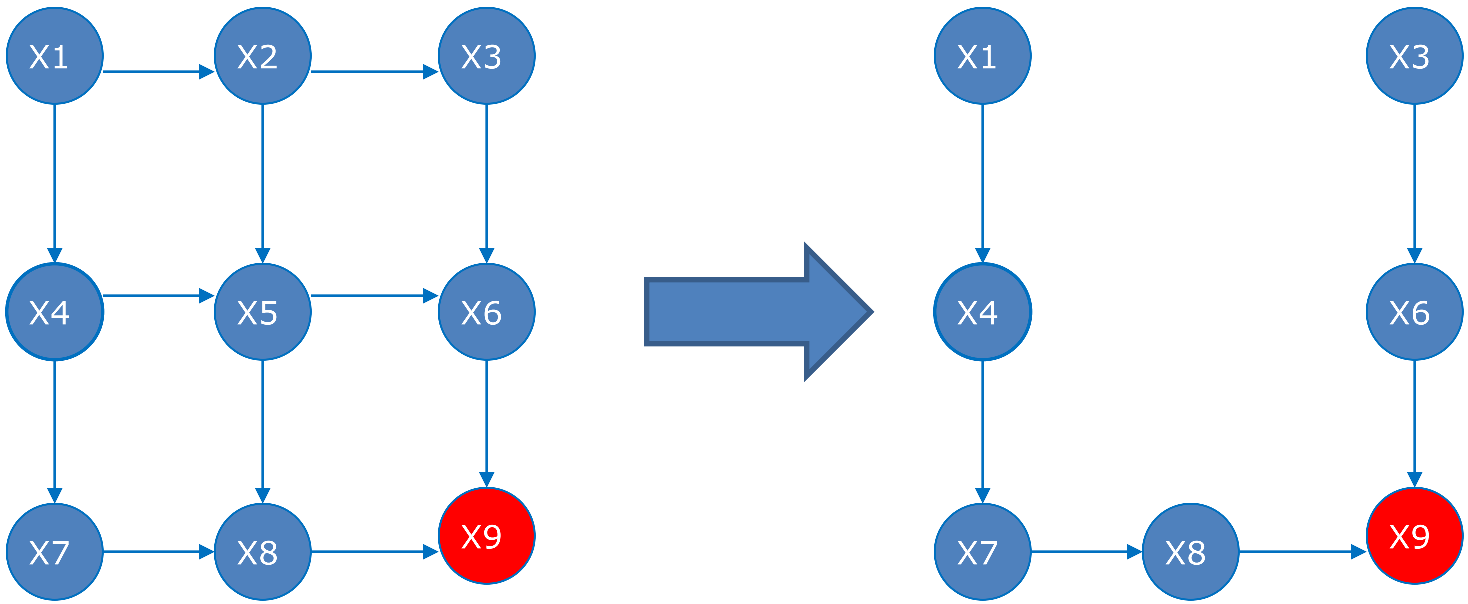
- ***w*-cutset** is a subset of variable s.t. when they are observed, induced width of the graph is w
- when sampled variables form a ***w*-cutset** , inference is exp(w) (e.g., using *Bucket Tree Elimination*)
- **cycle-cutset** is a special case of w-cutset

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

What If C=Cycle-Cutset ?

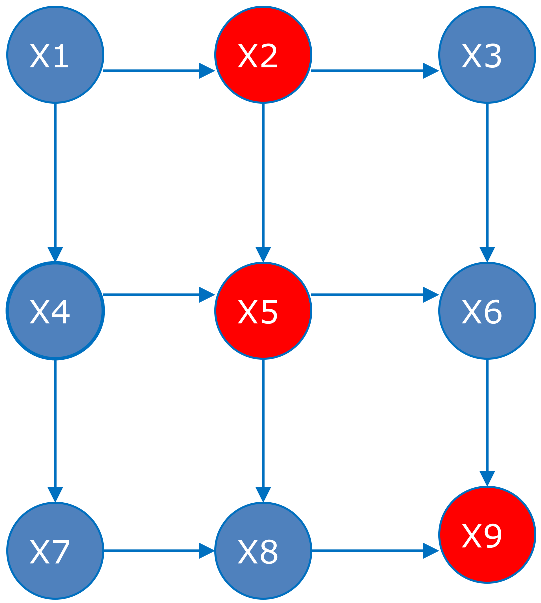
$$c^0 = \{x_2^0, x_5^0\}, E = \{X_9\}$$

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



$P(x_2,x_5,x_9)$ – 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

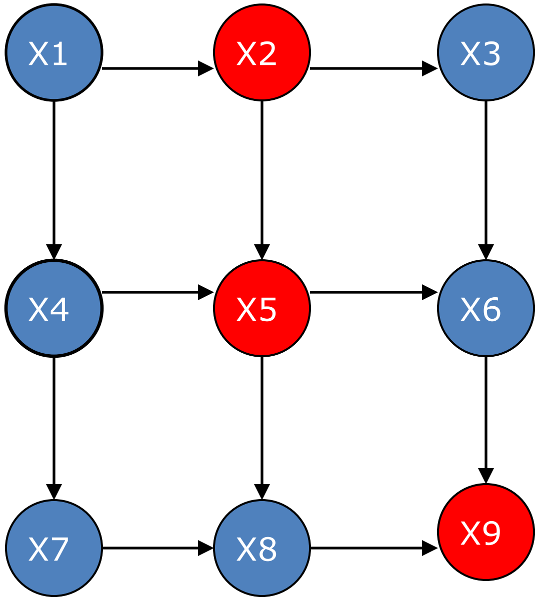
$$\overline{P}(x_i|e) = \frac{1}{T} \sum_{t=1}^{T} P(x_i \mid c^t, e)$$

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

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

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

$\dots$

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

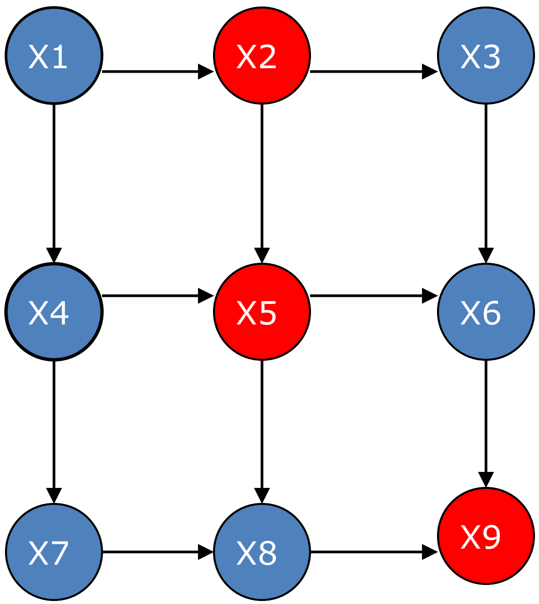
$\dots$

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



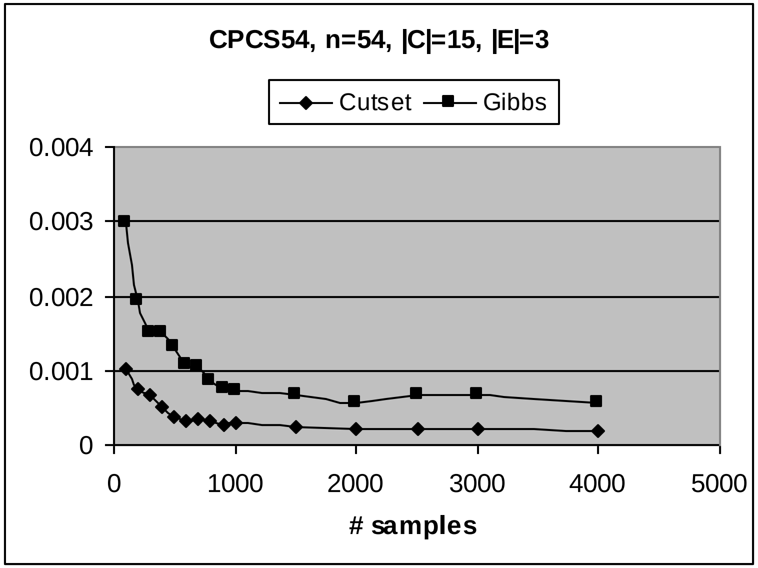
$$c^1 = \{x_2^1, x_5^1\} \Rightarrow P(x_3 | x_2^1, x_5^1, x_9)$$

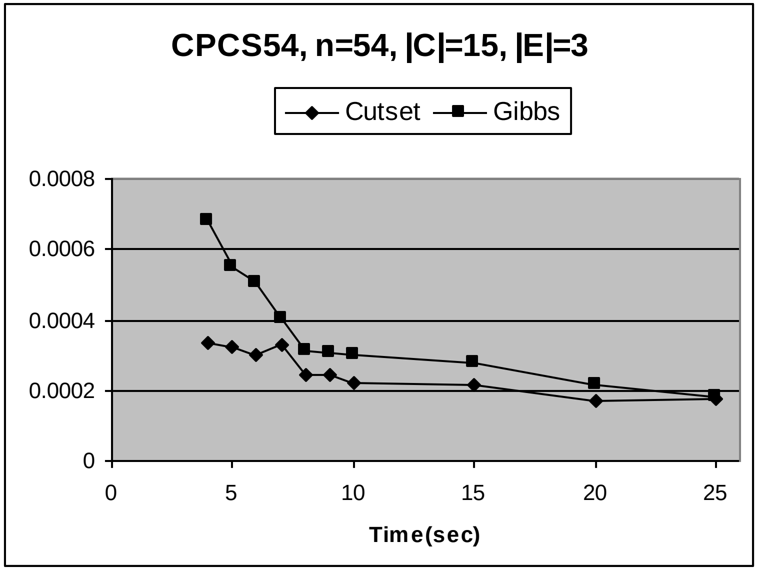
$$c^2 = \{x_2^2, x_5^2\} \Rightarrow P(x_3 | x_2^2, x_5^2, x_9)$$

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

$$P(x_3 | x_9) = \frac{1}{3}\begin{bmatrix} P(x_3 | x_2^1, x_5^1, x_9) \\ + P(x_3 | x_2^2, x_5^2, x_9) \\ + P(x_3 | x_2^3, x_5^3, x_9) \end{bmatrix}$$

CPCS54 Test Results



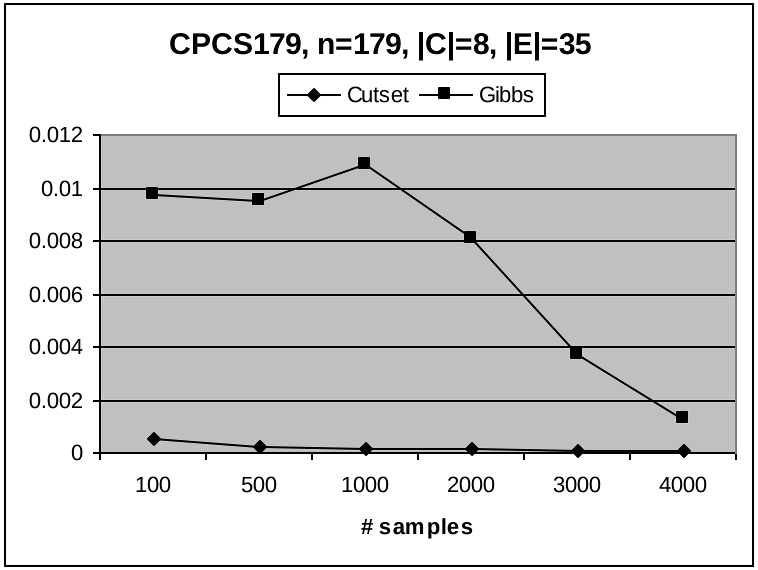


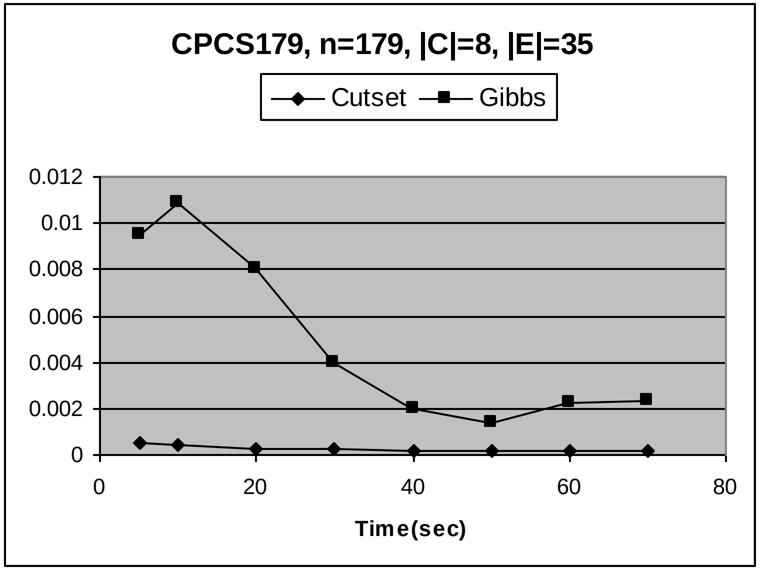
MSE vs. #samples (left) and time (right)

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

Exact Time = 30 sec using Cutset Conditioning

CPCS179 Test Results

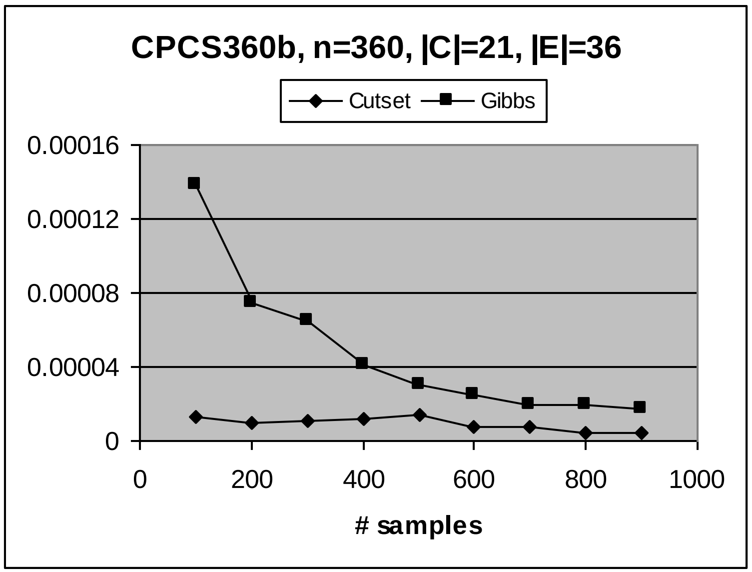


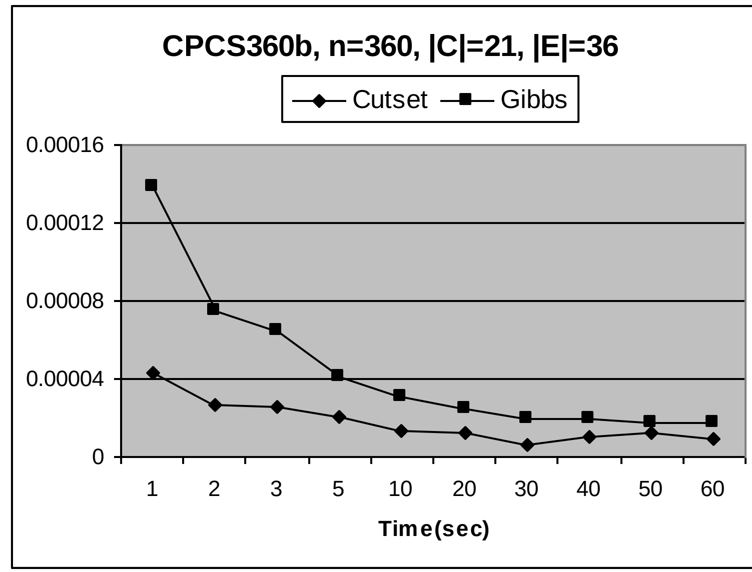


MSE vs. #samples (left) and time (right)
Non-Ergodic (1 deterministic CPT entry)
$|X| = 179$, $|C| = 8$, $2 <= D(X_i) <= 4$, $|E| = 35$

Exact Time = 122 sec using Cutset Conditioning

CPCS360b Test Results





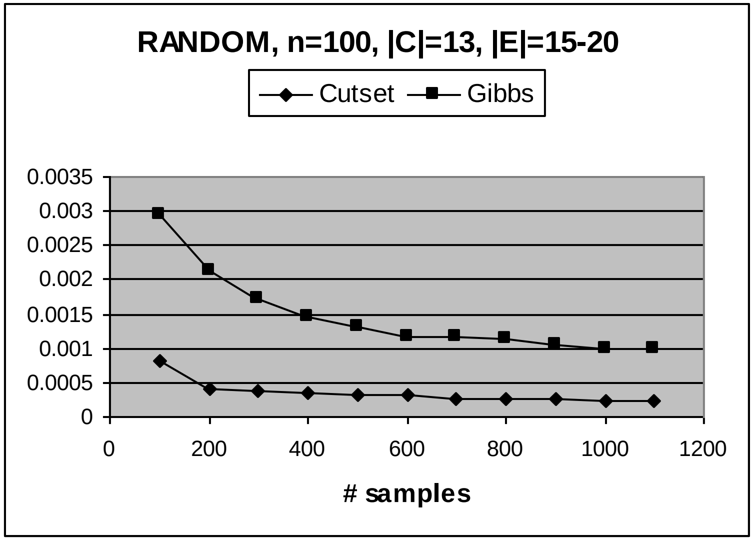
MSE vs. #samples (left) and time (right)

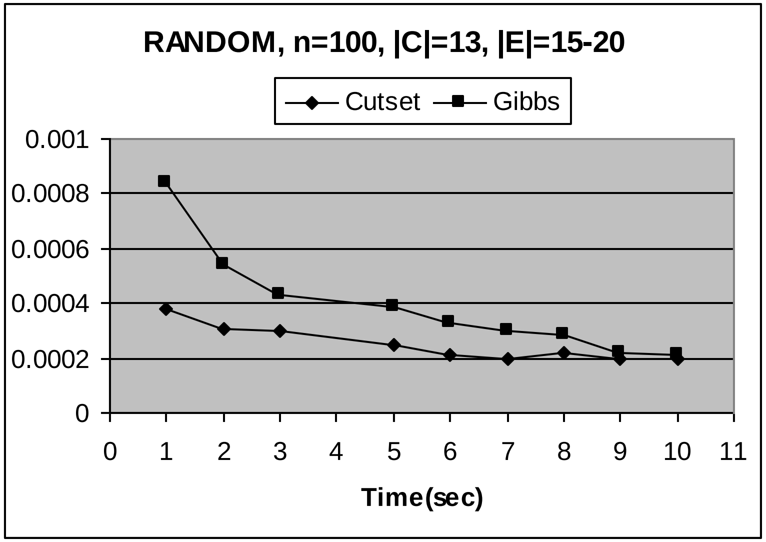
Ergodic, |X| = 360, $D(X_i)$=2, |C| = 21, |E| = 36

Exact Time > 60 min using Cutset Conditioning

Exact Values obtained via Bucket Elimination

Random Networks





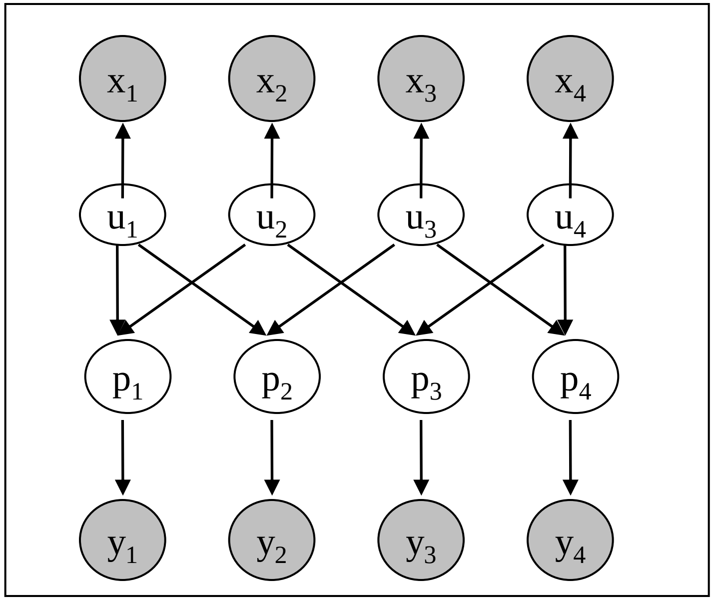
MSE vs. #samples (left) and time (right)

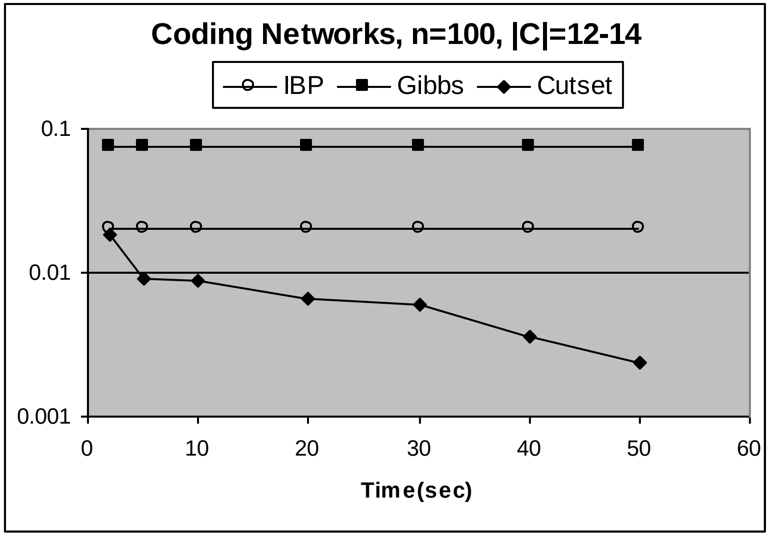
$|X| = 100$, $D(X_i) = 2$, $|C| = 13$, $|E| = 15\text{-}20$

Exact Time = 30 sec using Cutset Conditioning

Coding Networks

Cutset Transforms Non-Ergodic Chain to Ergodic





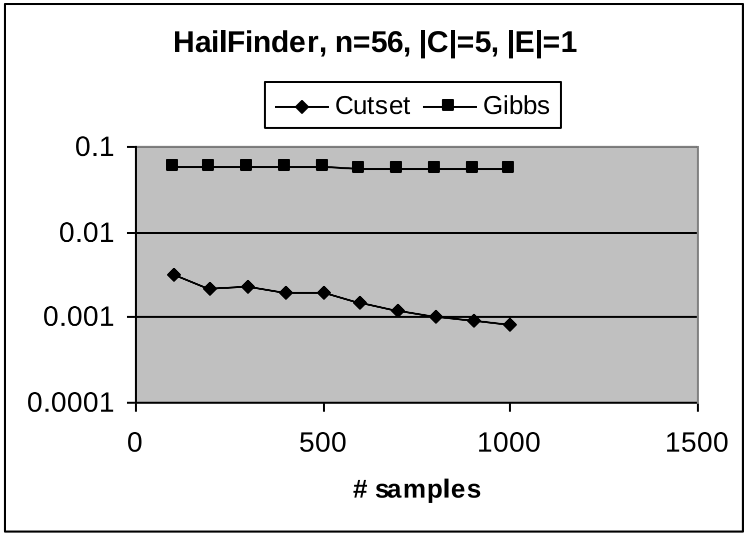
MSE vs. time (right)

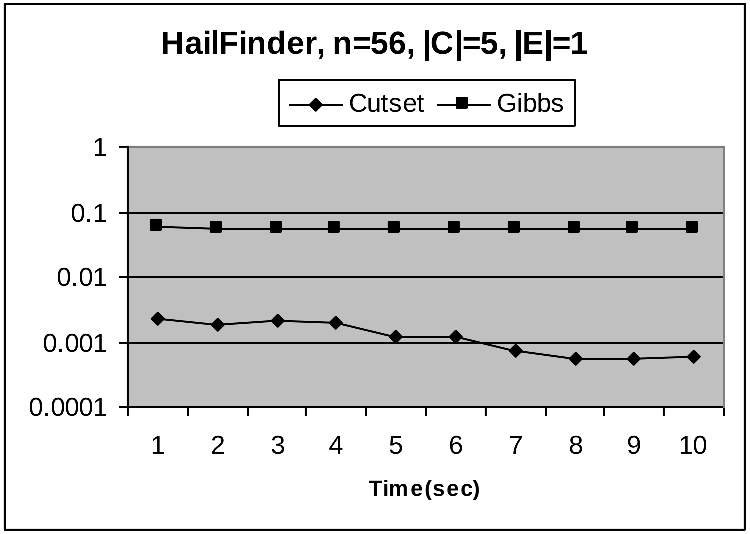
Non-Ergodic, |X| = 100, $D(X_i)=2$, |C| = 13-16, |E| = 50

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

Exact Time = 50 sec using Cutset Conditioning

Non-Ergodic Hailfinder



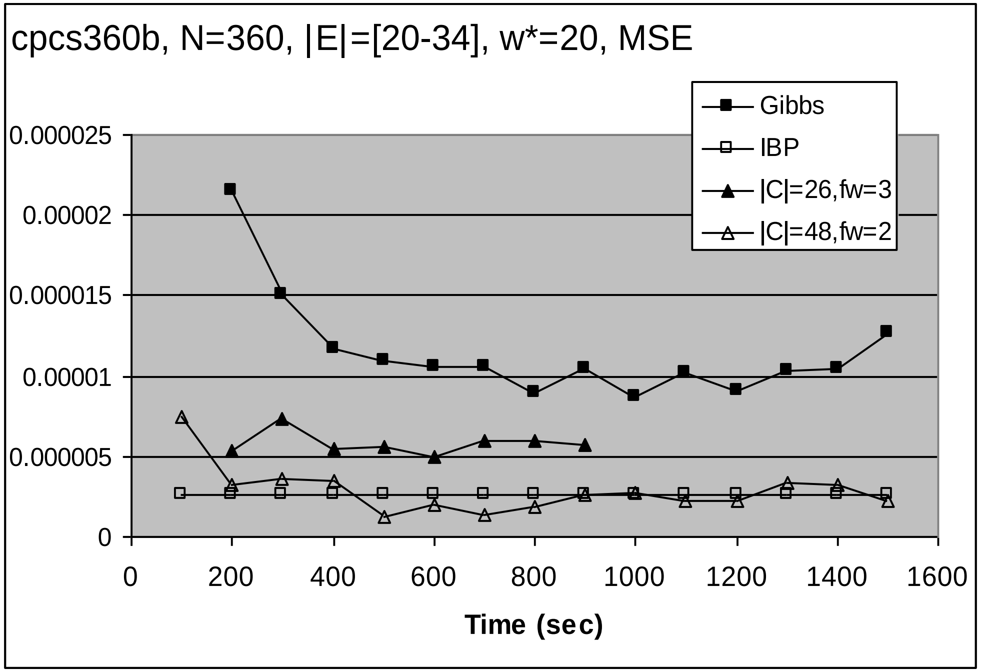


MSE vs. #samples (left) and time (right)

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:

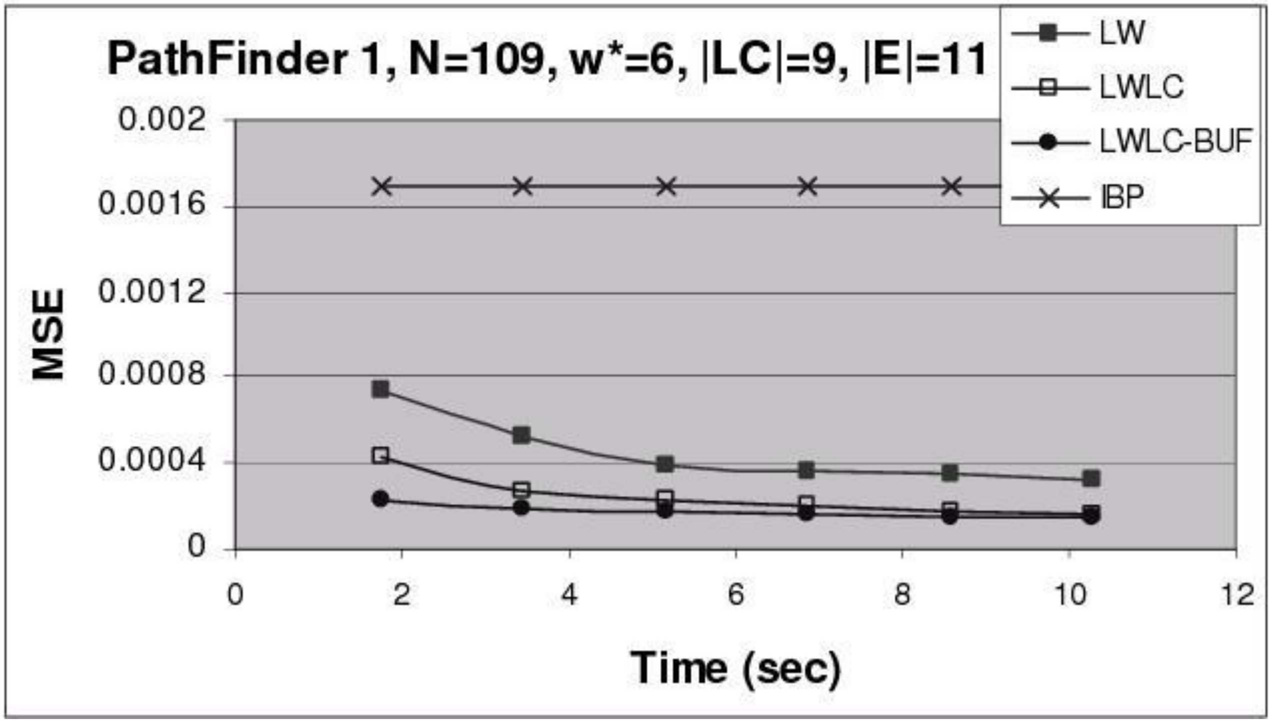
$$
\begin{aligned}
&\text{For } Z_i \in Z \text{ do}: \\
&\quad \text{If } Z_i \in E \\
&\qquad z_i^{t+1} = z_i, z_i \in e \\
&\quad \text{Else} \\
&\qquad z_i^{t+1} \leftarrow P(Z_i \mid z_1^{t+1},\ldots,z_{i-1}^{t+1}) \\
&\quad \text{End If} \\
&\text{End For}
\end{aligned}
$$

- computed while generating sample t using bucket tree elimination

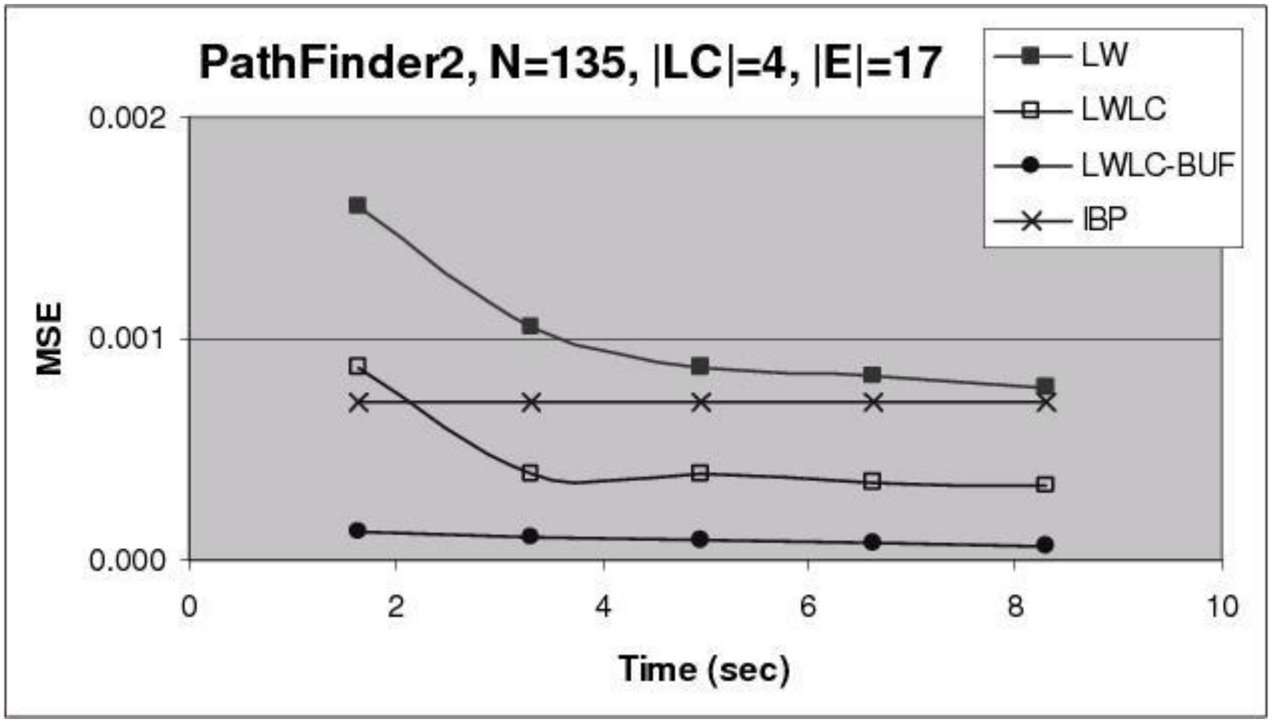
- can be memoized for some number of instances K (based on memory available

KL[P(C|e), Q(C)] ≤ KL[P(X|e), Q(X)]

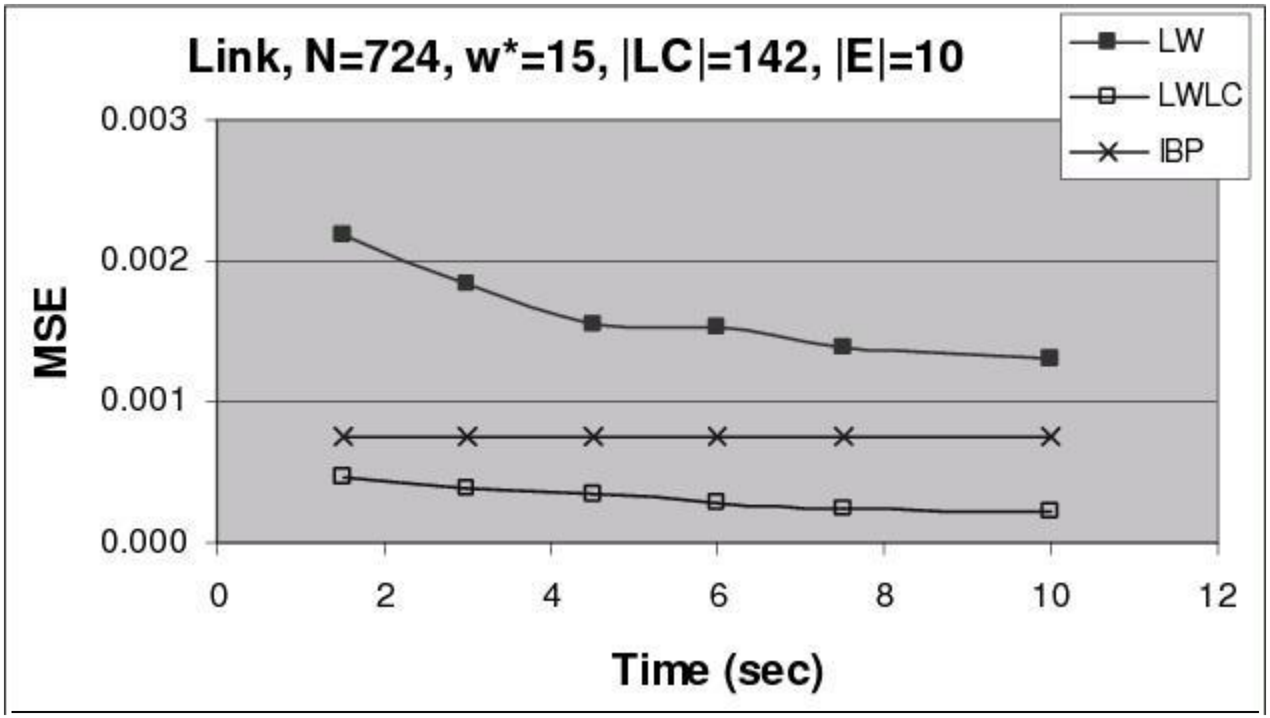
Pathfinder 1



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

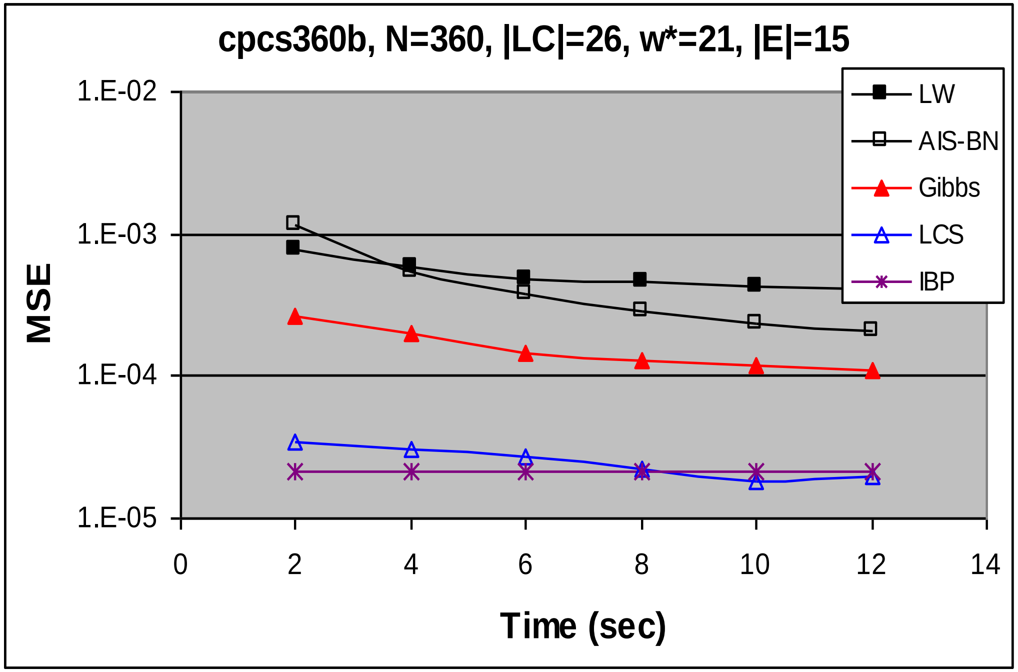
Importance Sampling

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

Gibbs Sampling

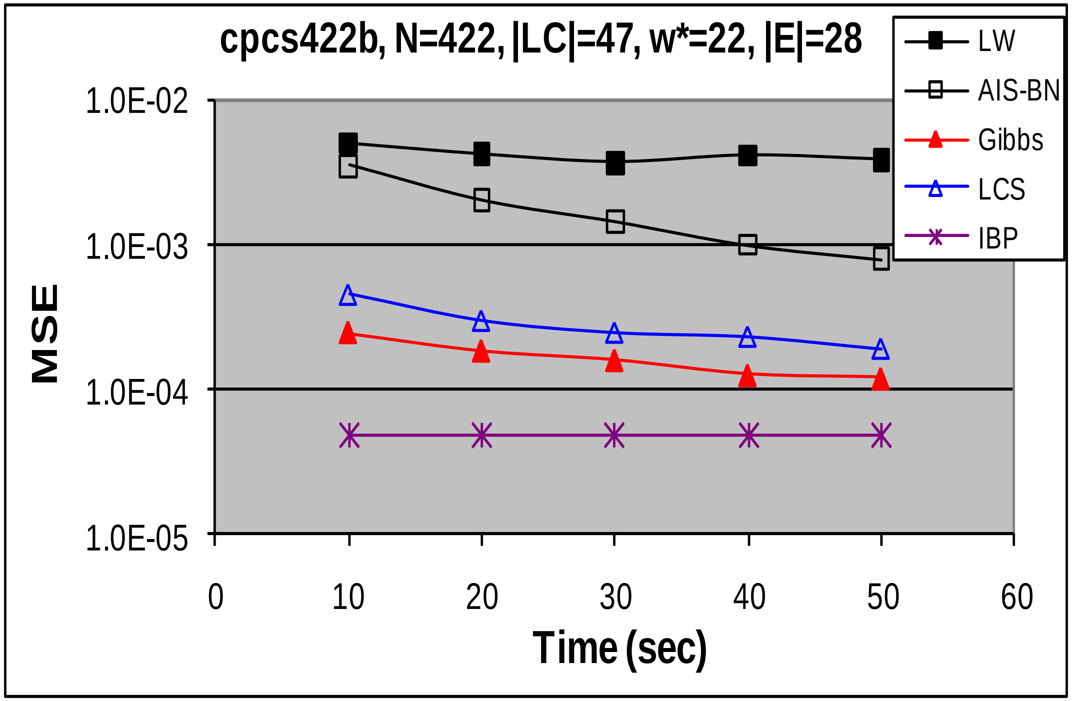
- Dependent samples
- Biased estimator
- Generates samples slower
- Samples from $\bar{P}(X|e)$
- Does not converge in presence of constraints
- Improves on cutset

CPCS360b



LW – likelihood weighting
LCS – likelihood weighting on a cutset

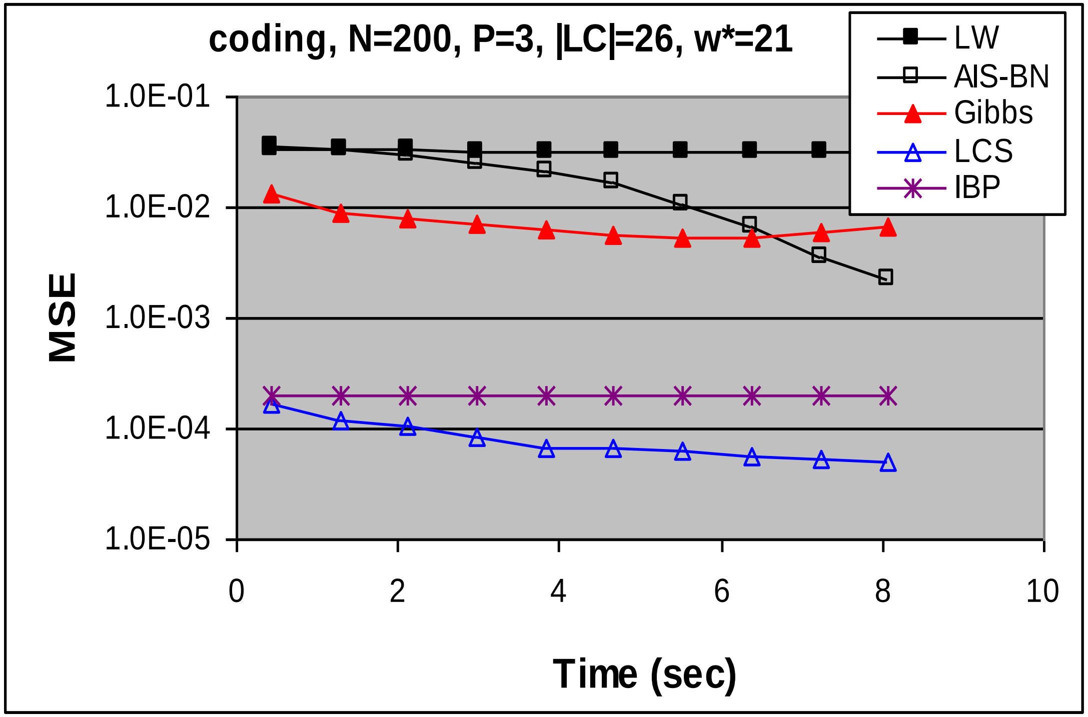
CPCS422b



LW – likelihood weighting

LCS – likelihood weighting on a cutset

Coding Networks



LW – likelihood weighting

LCS – likelihood weighting on a cutset

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

Motivation

Expected value of the number on the face of a die:

$$\frac{1+2+3+4+5+6}{6} = 3.5$$

What is the expected value of the product of the numbers on the face of "k" dice?

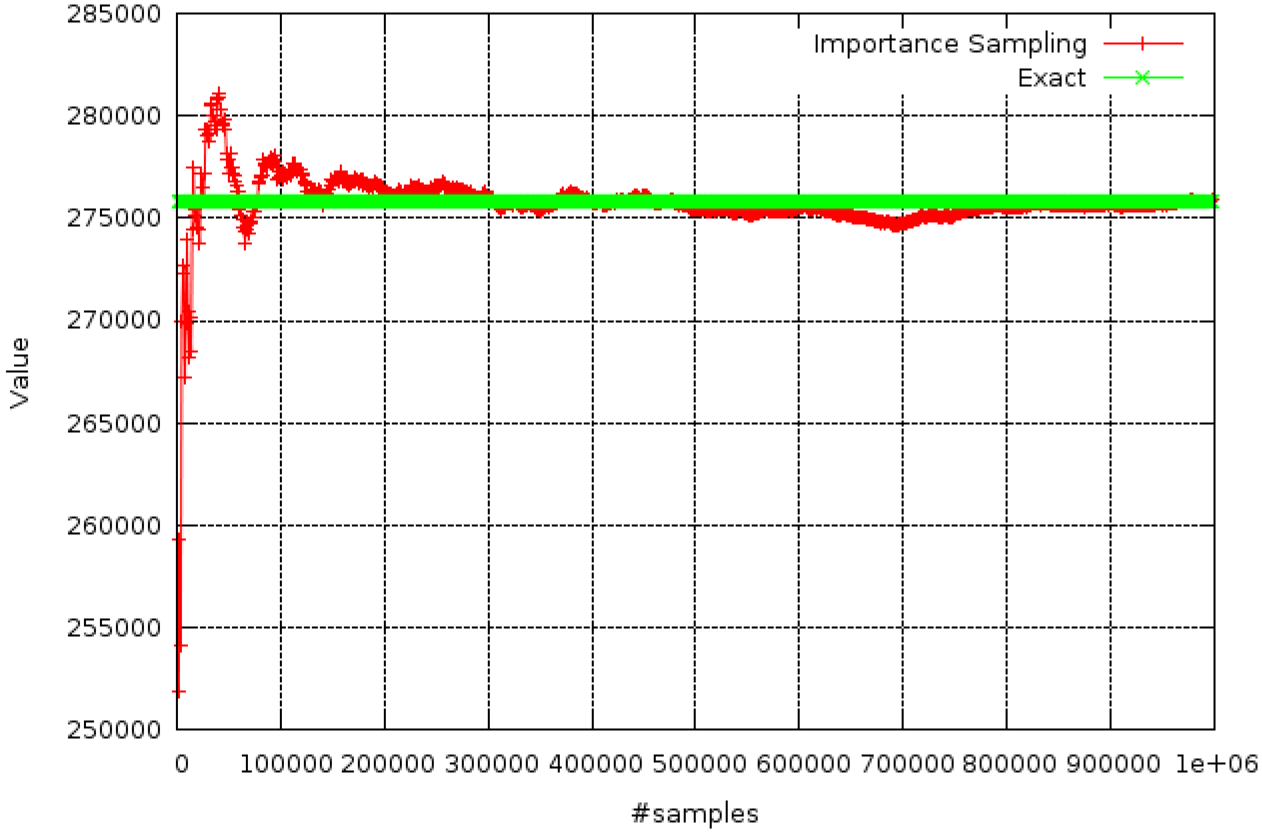
$$(3.5)^k$$

Monte Carlo estimate

- Perform the following experiment N times.
  - Toss all the k dice.
  - Record the product of the numbers on the top face of each die.
- Report the average over the N runs.

$$\hat{Z} = \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{k}(\text{the number on the face of the "}j^{\text{th}}\text{" dice in the }N^{\text{th}}\text{ run})$$

How the sample average converges?



10 dice. Exact Answer is $(3.5)^{10}$

But This is Really Dumb?

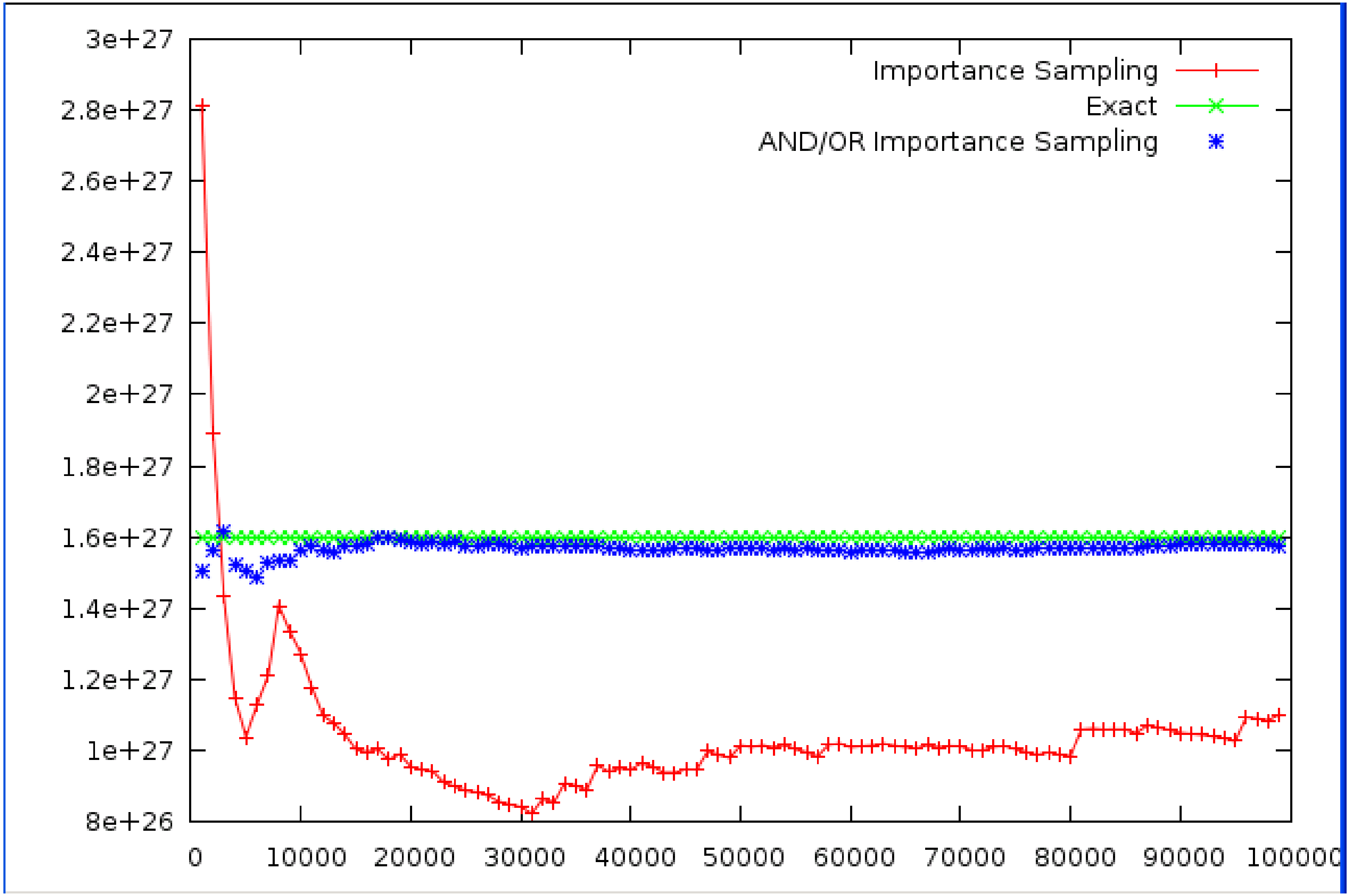
- The dice are independent.
- A better Monte Carlo estimate
  1. Perform the experiment N times
  2. For each dice record the average
  3. Take a **product of the averages**

$$\hat{Z}_{new} = \prod_{j=1}^{k} \frac{1}{N} \sum_{i=1}^{N} \textbf{(}\text{the number on the face of the "j}^{\text{th}}\text{"dice in the N}^{\text{th}}\text{ run}\textbf{)}$$

- **Conventional estimate: Averages of products.**

$$\hat{Z} = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{k} (\text{the number on the face of the "j}^{\text{th}}\text{"dice in the N}^{\text{th}}\text{ run})$$

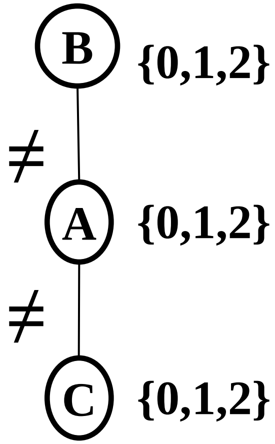
How the sample Average Converges



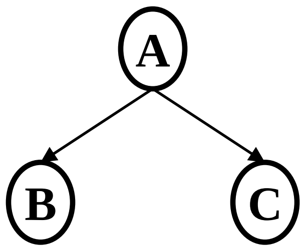
Moral of the story

- Make use of (conditional) independence to get better results
- Used for exact inference extensively
  - Bucket Elimination (Dechter, 1996)
  - Junction tree (Lauritzen and Speigelhalter, 1988)
  - Value Elimination (Bacchus et al. 2004)
  - Recursive Conditioning (Darwiche, 2001)
  - BTD (Jegou et al., 2002)
  - AND/OR search (Dechter and Mateescu, 2007)
- How to use it for sampling?
  - AND/OR Importance sampling

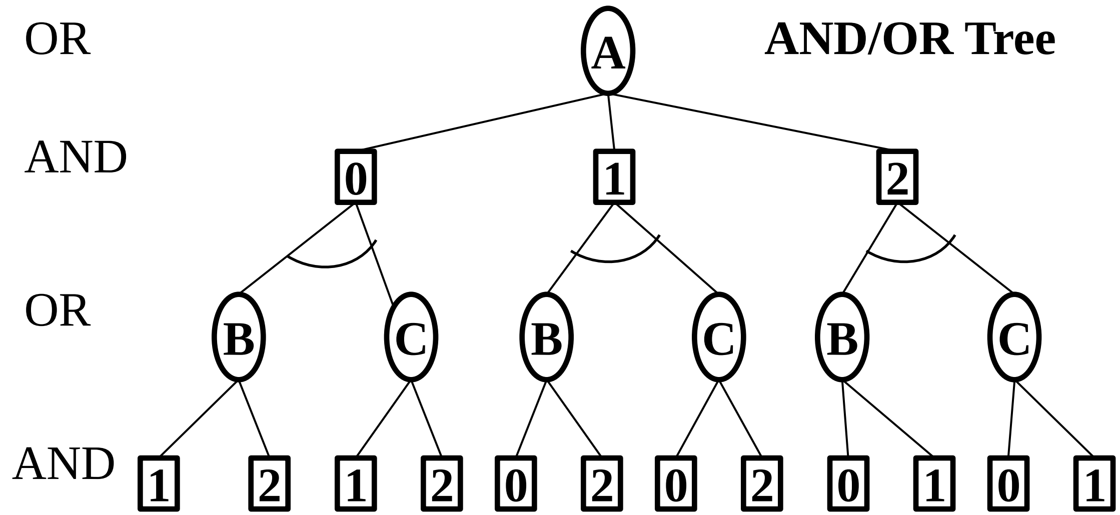
Background: AND/OR search space

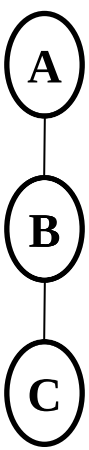


Problem

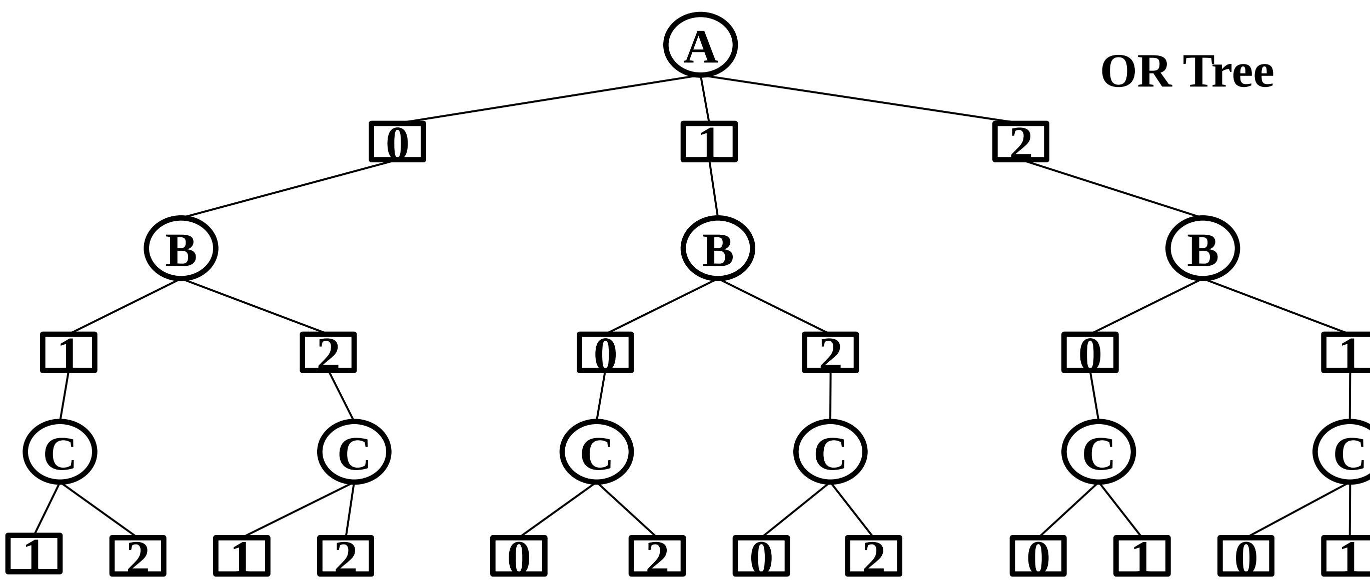


Pseudo-tree

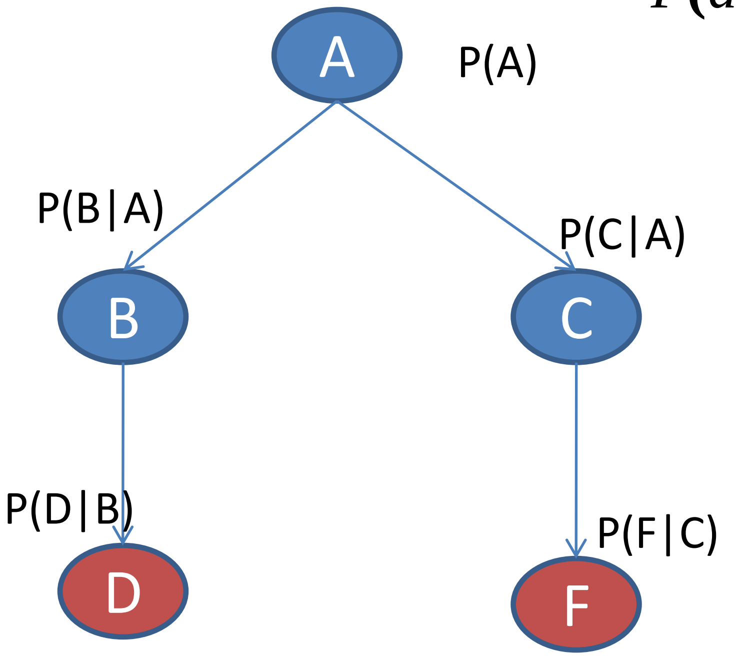




Chain Pseudo-tree

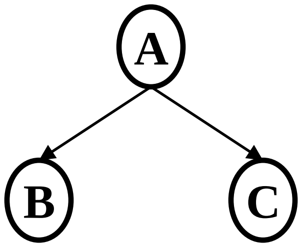


AND/OR sampling: Example



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

AND/OR Importance Sampling (General Idea)



Pseudo-tree

- Decompose Expectation

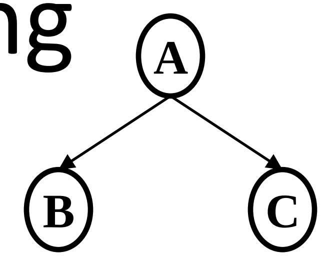
$$P(d,f)=\sum_{a,b,c}P(a)P(c|a)P(b|a)P(d|b)P(f|c)$$

$$Q(A,B,C)=Q(A)Q(B|A)Q(C|A)$$

$$P(d,f)=\sum_{a,b,c}\frac{P(a)P(c|a)P(b|a)P(d|b)P(f|c)}{Q(a)Q(b|a)Q(c|a)}Q(a)Q(b|a)Q(c|a)$$

$$=E_Q\left[\frac{P(a)P(c|a)P(b|a)P(d|b)P(f|c)}{Q(a)Q(b|a)Q(c|a)}\right]$$

AND/OR Importance Sampling (General Idea)



Pseudo-tree

- Decompose Expectation

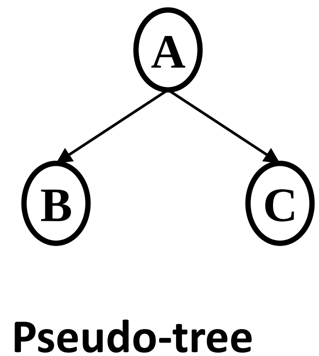
$$P(d,f)=\sum_{a,b,c}\frac{P(a)P(c|a)P(b|a)P(d|b)P(f|c)}{Q(a)Q(b|a)Q(c|a)}Q(a)Q(b|a)Q(c|a)$$

$$P(d,f)=\sum_{a}\frac{P(a)}{Q(a)}Q(a)\sum_{b}\frac{P(b|a)P(d|b)}{Q(b|a)}Q(b|a)\sum_{c}\frac{P(c|a)P(f|c)}{Q(c|a)}Q(c|a)$$

$$P(d,f)=E_Q\left[\frac{P(a)}{Q(a)}E_Q\left[\frac{P(b|a)P(d|b)}{Q(b|a)}|a\right]E_Q\left[\frac{P(c|a)P(f|c)}{Q(c|a)}|a\right]\right]$$

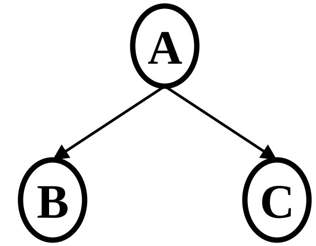
AND/OR Importance Sampling (General Idea)

$$P(d,f) = E_Q\left[\frac{P(a)}{Q(a)} E_Q\left[\frac{P(b|a)P(d|b)}{Q(b|a)} \,|\, a\right] E_Q\left[\frac{P(c|a)P(f|c)}{Q(c|a)} \,|\, a\right]\right]$$

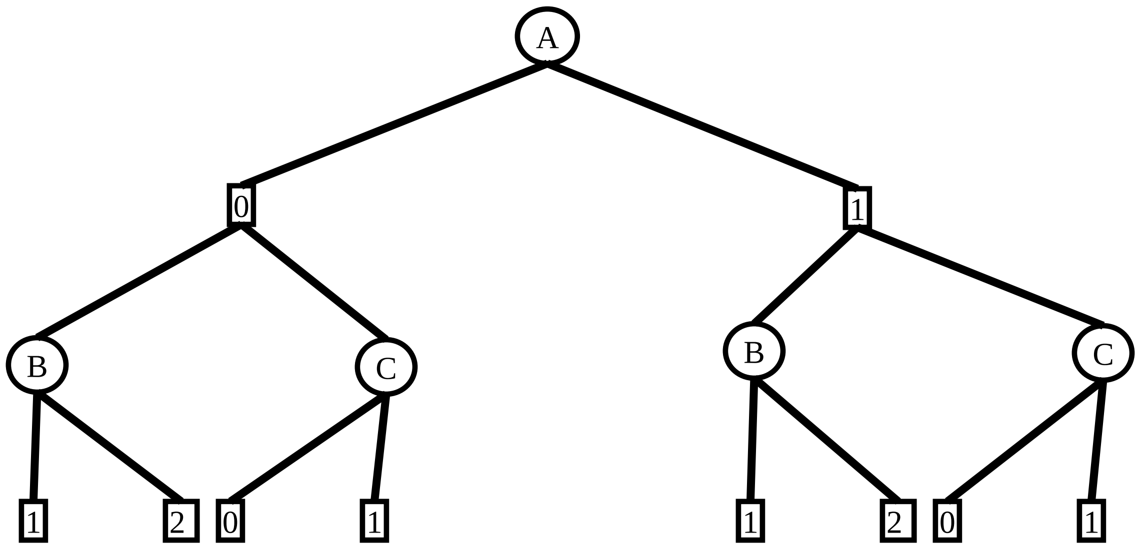


- Compute all expectations separately
- How?
  - Record all samples
  - For each sample that has A=a
    - Estimate the conditional expectations separately using the generated samples
    - Combine the results

AND/OR Importance Sampling



Pseudo-tree



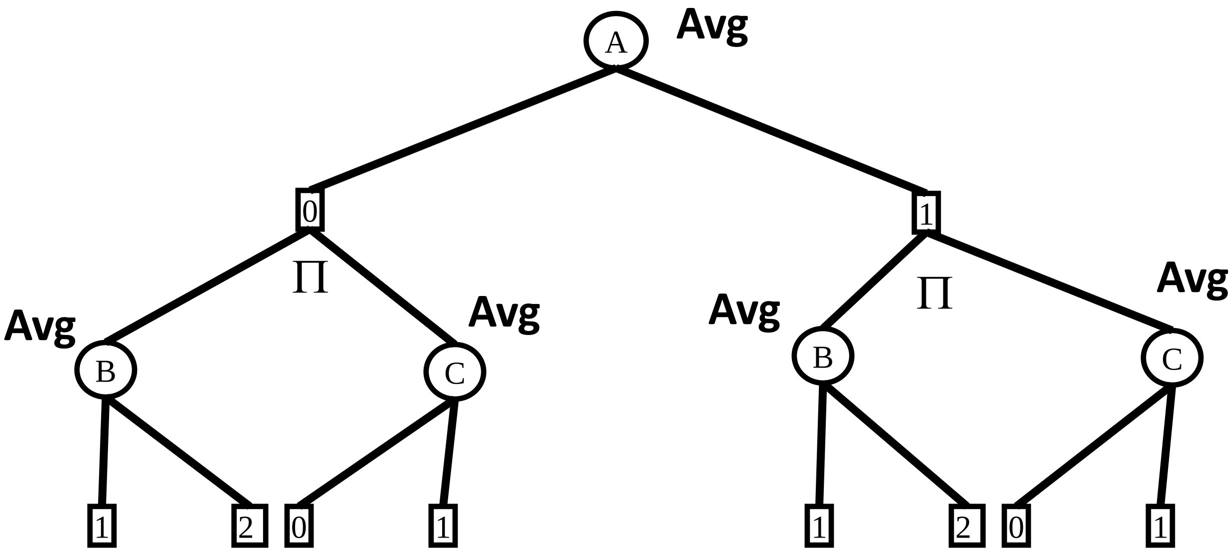
Sample #	A	B	C
1	0	1	0
2	0	2	1
3	1	1	1
4	1	2	0

$$\text{Estimate of } E\left[\frac{P(b \mid A=0)P(d \mid b)}{Q(b \mid A=0)} \mid A=0\right]$$

$$= \text{Average Weight of samples of B having A} = 0$$

$$= \frac{\mathrm{w}(\mathrm{B}=1, \mathrm{A}=0) + \mathrm{w}(\mathrm{B}=2, A=0)}{2}$$

AND/OR Importance Sampling



Sample #	Z	X	Y
1	0	1	0
2	0	2	1
3	1	1	1
4	1	2	0

All AND nodes: Separate Components. Take Product

Operator: Product

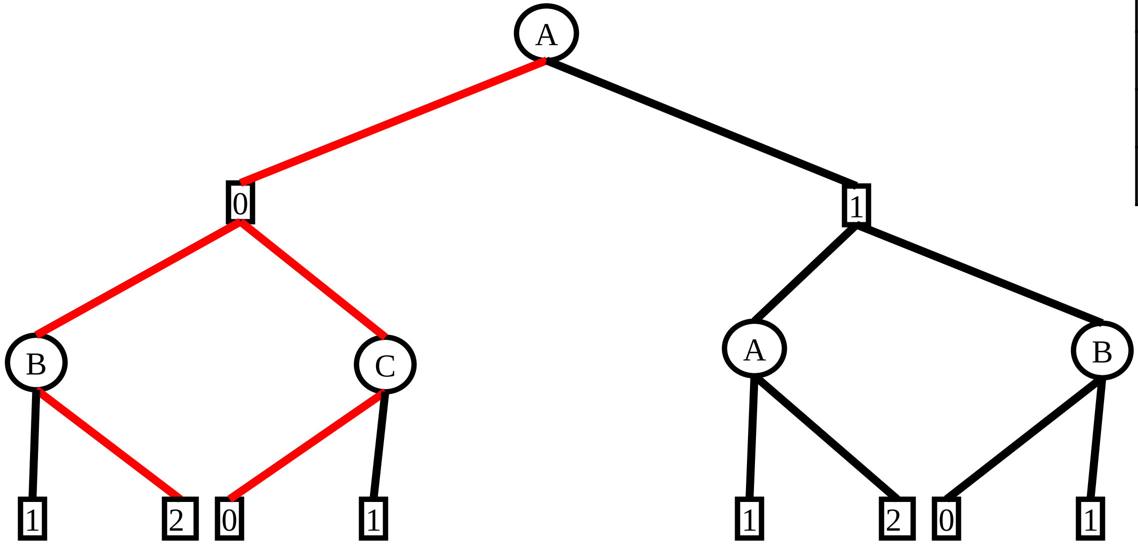
All OR nodes: Conditional Expectations given the assignment above it

Operator: Weighted Average

Algorithm AND/OR Importance Sampling

1. Construct a pseudo-tree.
2. Construct a proposal distribution along the pseudo-tree
3. Generate samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ *from Q along O.*
4. Build a AND/OR sample tree f*or the samples* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ along the ordering *O.*
5. **FOR** all leaf nodes *i of AND-OR tree do*
   1. **IF AND-node v(i)= 1 ELSE v(i)=0**
6. **FOR** every node *n from leaves to the root do*
   1. ***IF AND-node v(n)=product of children***
   2. ***IF OR-node v(n) = Average of children***
7. ***Return v(root-node)***

samples in AND/OR vs Conventional



Sample #	A	B	C
1	0	1	0
2	0	2	1
3	1	1	1
4	1	2	0

- 8 Samples in AND/OR space versus 4 samples in importance sampling
- Example: A=0, B=2, C=0 is not generated but still considered in the AND/OR space

Why AND/OR Importance Sampling

- AND/OR estimates have smaller variance.
- Variance Reduction
  - Easy to Prove for case of complete independence (Goodman, 1960)

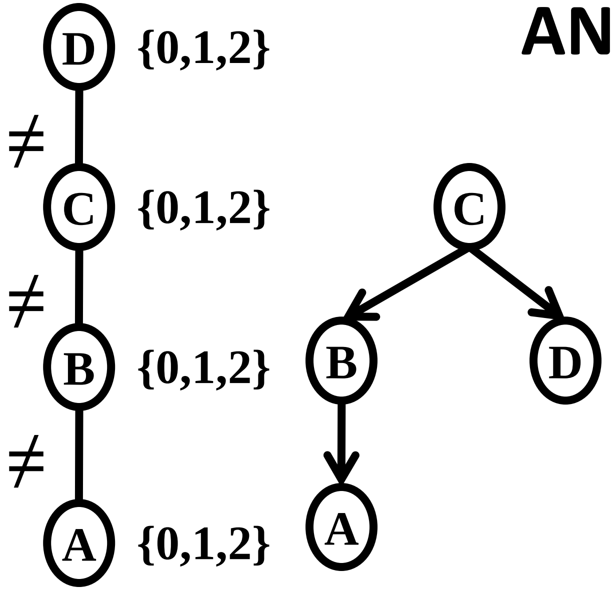
$$V[\overline{xy}] = \frac{V[x]E[y]^2}{N} + \frac{V[y]E[x]^2}{N} + \frac{V[x]V[y]}{N}, \text{not independent}$$

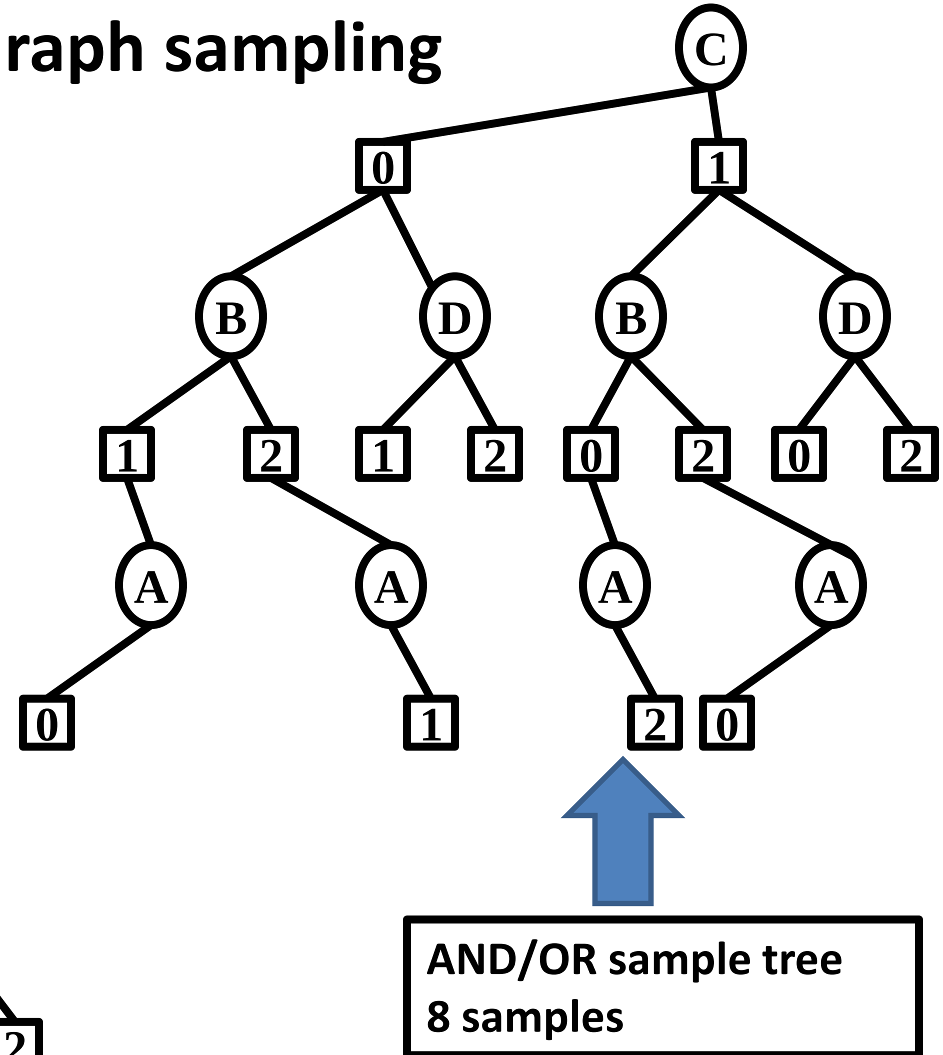
$$V[\bar{x}\bar{y}] = \frac{V[x]E[y]^2}{N} + \frac{V[y]E[x]^2}{N} + \frac{V[x]V[y]}{N^2}, \text{independent}$$

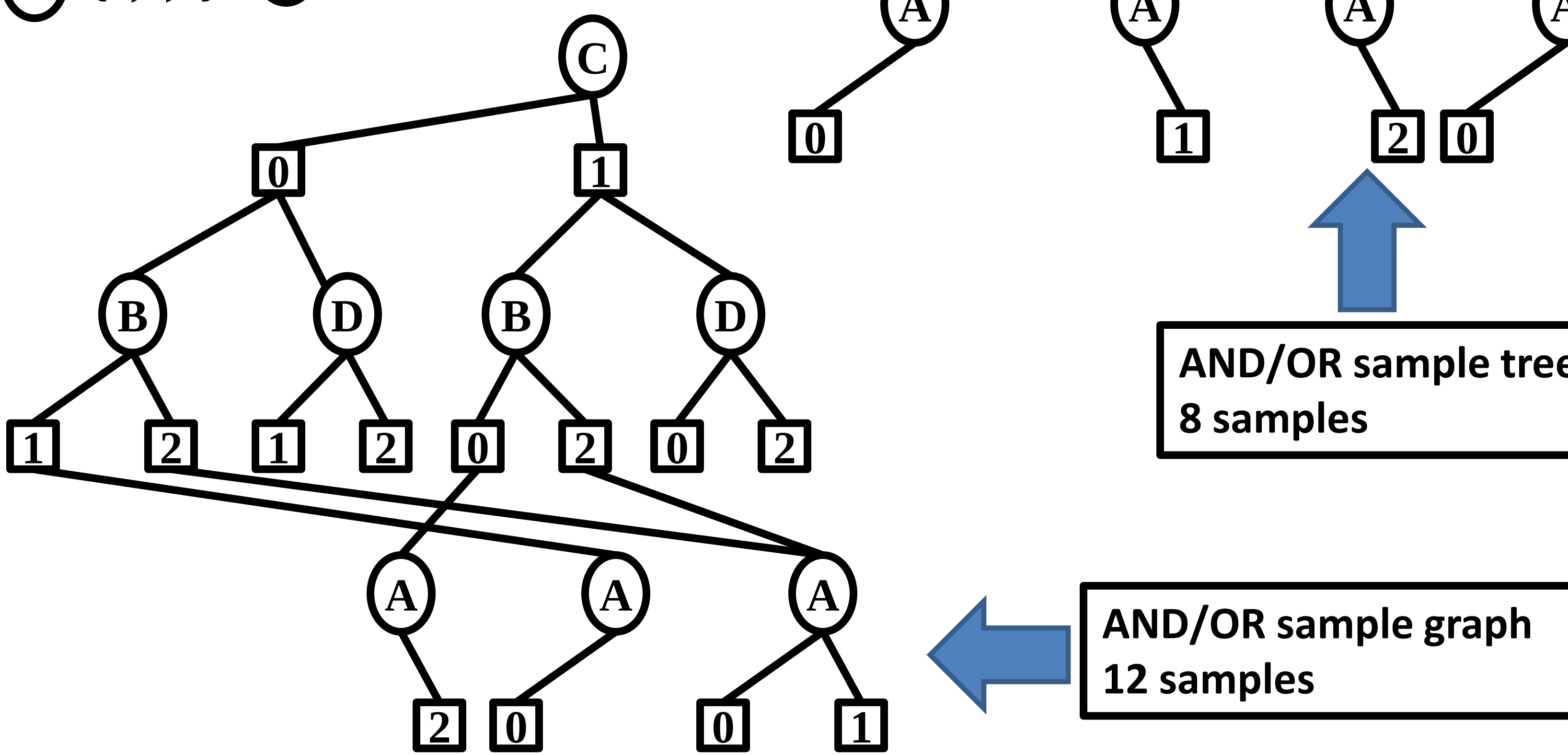
Note the squared term.

  - Complicated to prove for general conditional independence case (See Vibhav Gogate's thesis)!

AND/OR Graph sampling







Combining AND/OR sampling and w-cutset sampling

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

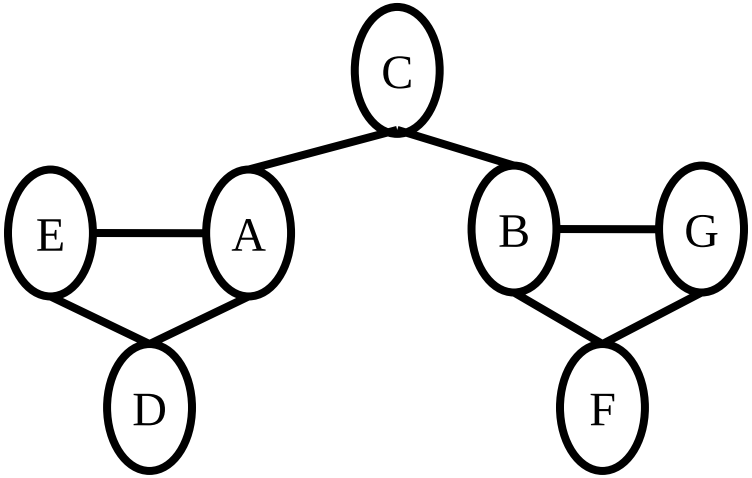
- Reduce the variance of weights
  - Rao-Blackwellised w-cutset sampling (Bidyuk and Dechter, 2007)
- Increase the number of samples; **kind of**
  - AND/OR Tree and Graph sampling (Gogate and Dechter, 2008)
- Combine the two

Algorithm AND/OR w-cutset sampling

Given an integer constant w

1. Partition the set of variables into K and R, such that the treewidth of R is bounded by w.
2. AND/OR sampling on K
   1. Construct a pseudo-tree of K and compute Q(K) consistent with K
   2. Generate samples from Q(K) and store them on an AND/OR tree
3. Rao-Blackwellisation (Exact inference) at each leaf
   1. For each leaf node of the tree compute Z(R|g) where g is the assignment from the leaf to the root.
4. Value computation: Recursively from the leaves to the root
   1. At each AND node compute product of values at children
   2. At each OR node compute a weighted average over the values at children
5. Return the value of the root node

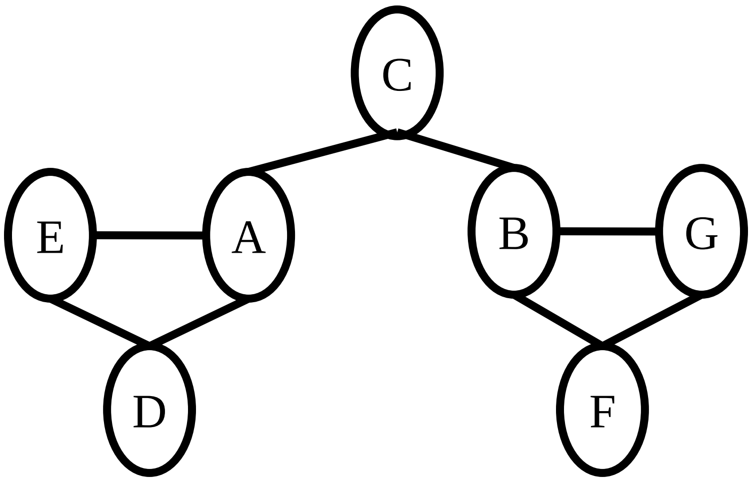
AND/OR w-cutset sampling: Step 1: Partition the set of variables



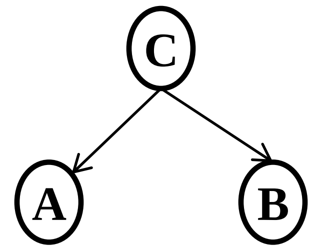
Graphical model

Practical constraint: Can only perform exact inference if the treewidth is bounded by 1.

AND/OR w-cutset sampling:
Step 2: AND/OR sampling over {A,B,C}

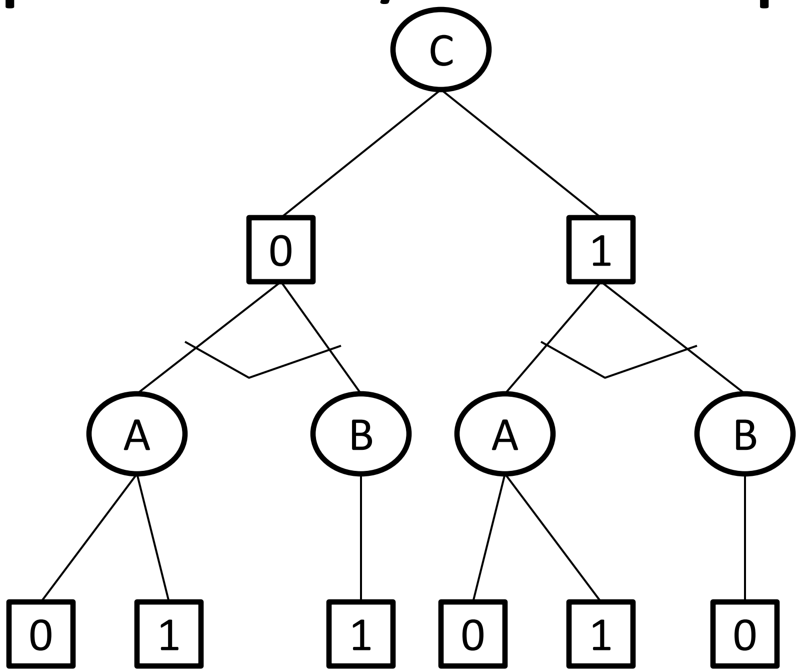


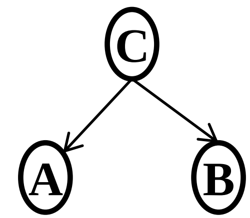
Graphical model



Pseudo-tree

AND/OR w-cutset sampling: Step 2: AND/OR sampling over {A,B,C}

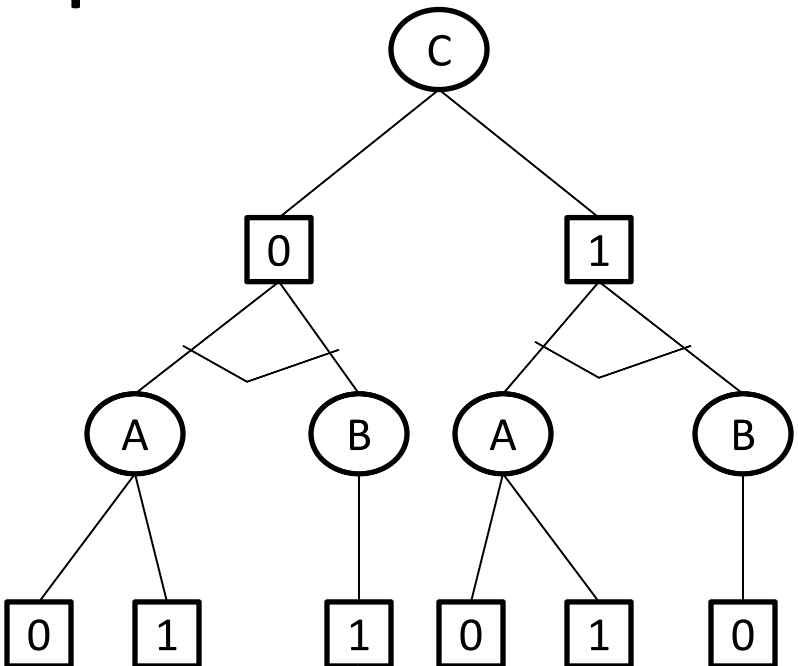


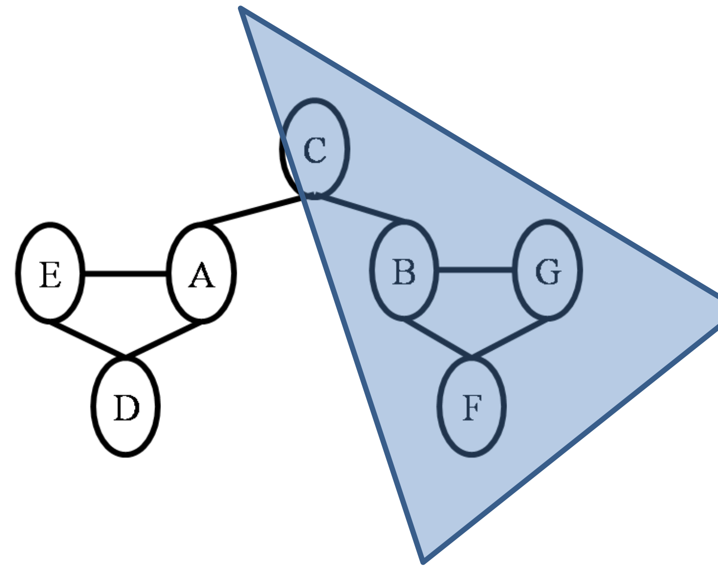


Pseudo-tree

Samples: (C=0,A=0,B=1), (C=0,A=1,B=1), (C=1,A=0,B=0), (C=1,A=1,B=0)

AND/OR w-cutset sampling: Step 3: Exact inference at each leaf

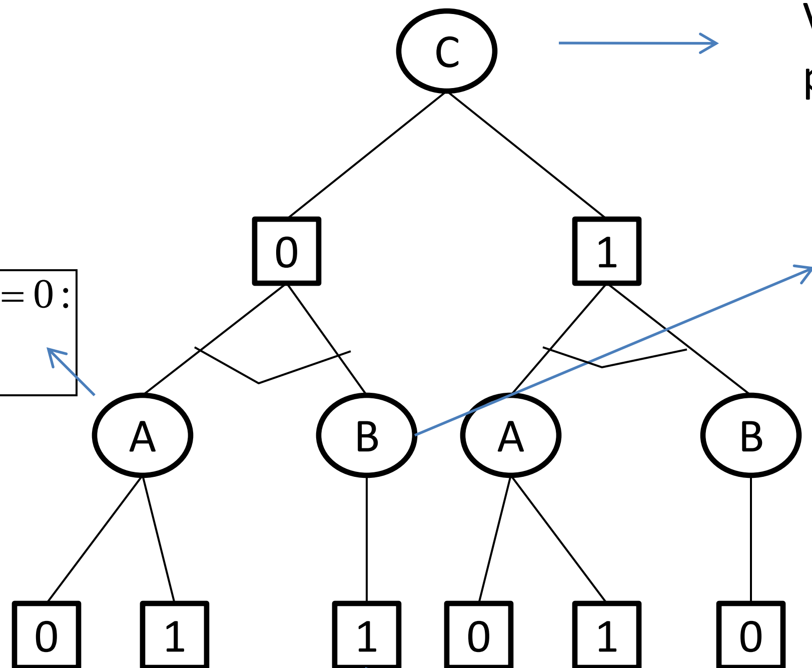




Value of $B=1 | C=0$:

$$v(B=1 | c=0) = \sum_{f \in F} \sum_{g \in G} H(C=0, g) H(B=1, g) H(g, f) H(B=1, f)$$

AND/OR w-cutset sampling: Step 4: Value computation



Value of C: Estimate of the partition function

Value of B given C = 0:
$= \hat{E}[B, F, G \mid C = 0]$

Value of A given C = 0:
$= \hat{E}[A, D, E \mid C = 0]$

Value of B = 1 given C = 0:

$$v(B=1 \mid C=0) = \sum_{f \in F} \sum_{g \in G} H(C=0, g) H(B=1, g) H(g, f) H(B=1, f)$$

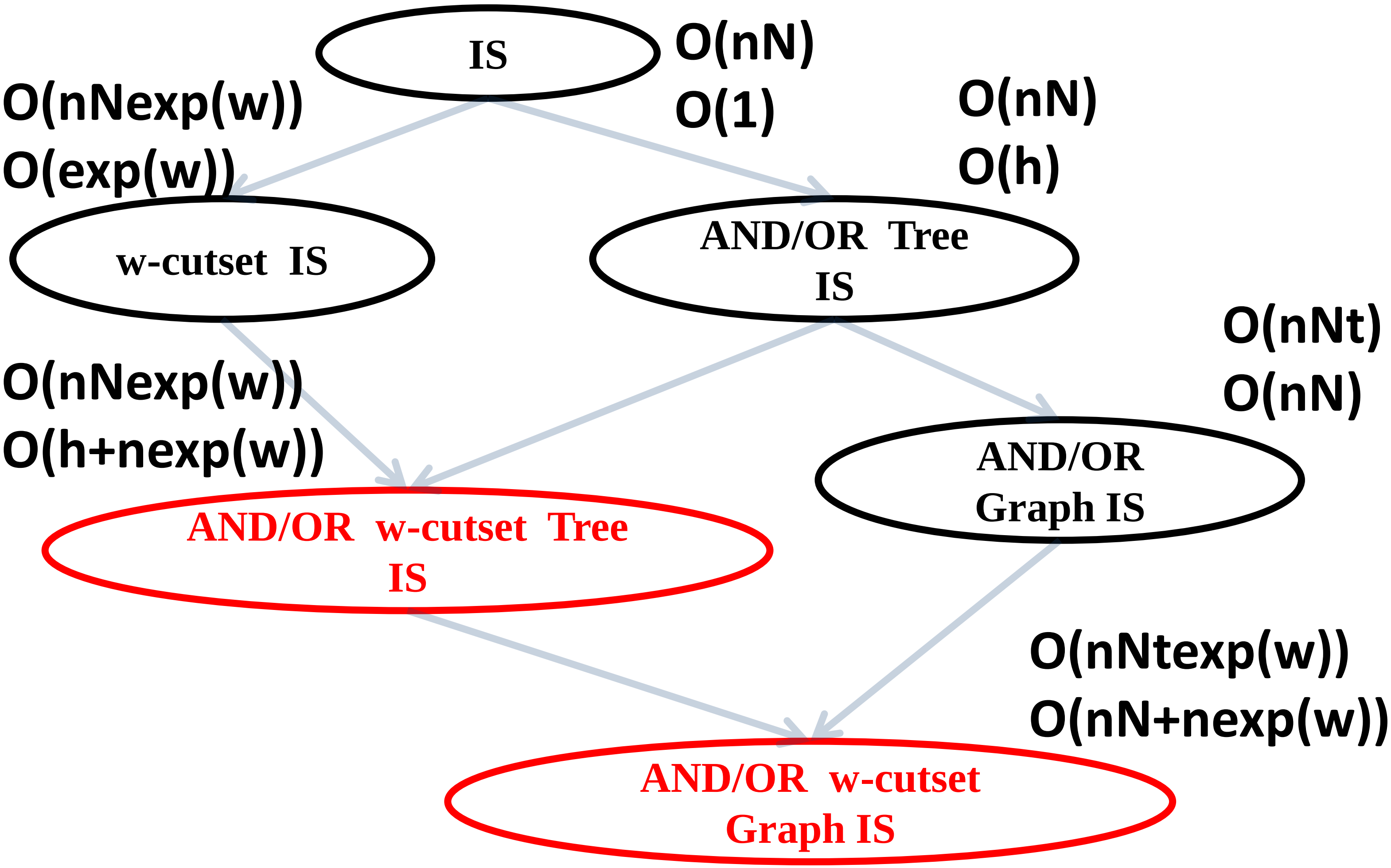
Properties and Improvements

- Basic underlying scheme for sampling remains the same
  - The only thing that changes is what you estimate from the samples
  - Can be combined with any state-of-the-art importance sampling technique
- Graph vs Tree sampling
  - Take full advantage of the conditional independence properties uncovered from the primal graph

AND/OR w-cutset sampling Advantages and Disadvantages

- Advantages
  - Variance Reduction
  - Relatively fewer calls to the Rao-Blackwellisation step due to efficient caching (Lazy Rao-Blackwellisation)
  - Dynamic Rao-Blackwellisation when context-specific or logical dependencies are present
    - Particularly suitable for Markov logic networks (Richardson and Domingos, 2006).
- Disadvantages
  - Increases time and space complexity and therefore fewer samples may be generated.

Take away Figure: Variance Hierarchy and Complexity



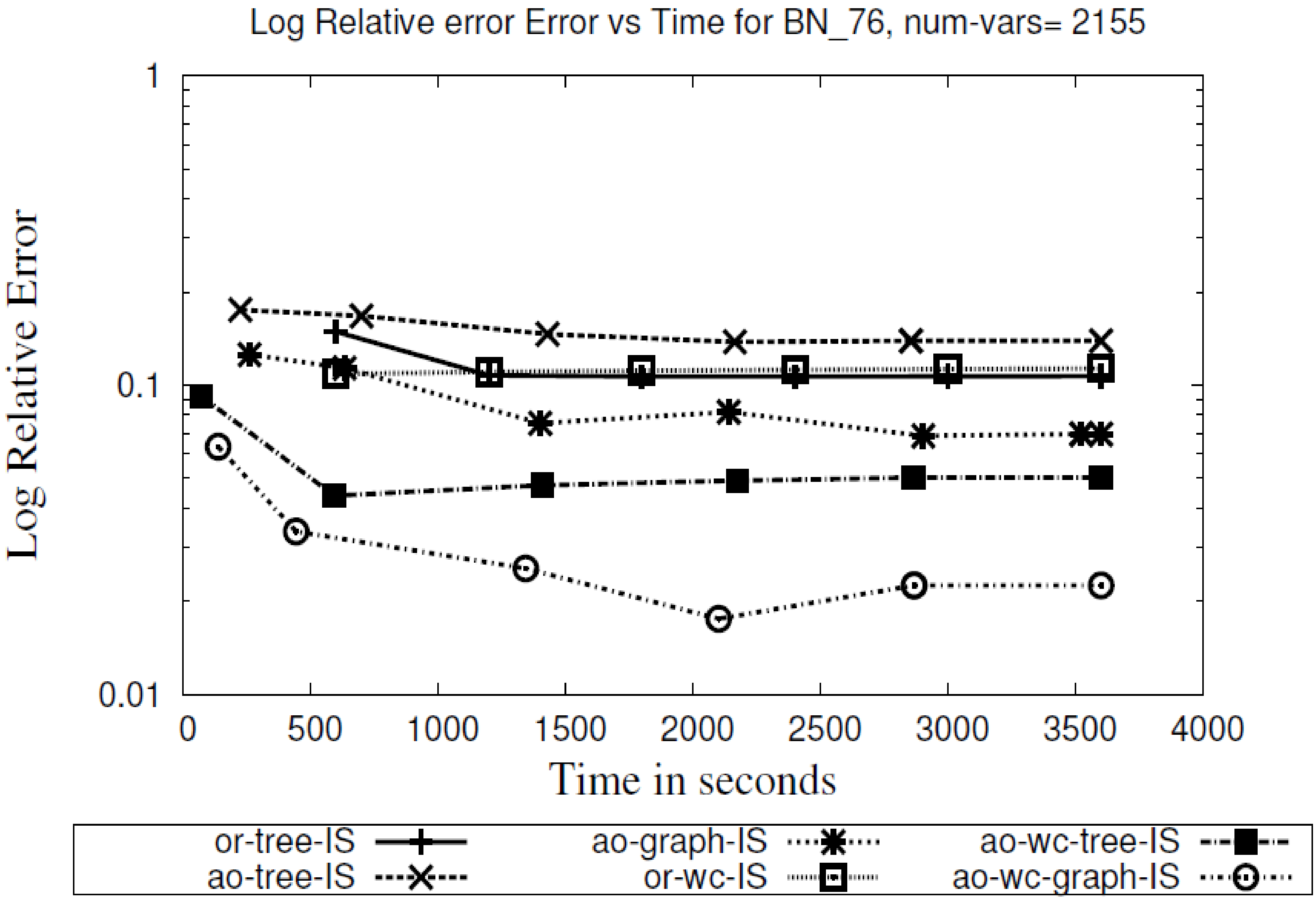
Experiments

- Benchmarks
  - Linkage analysis
  - Graph coloring
- Algorithms
  - OR tree sampling
  - AND/OR tree sampling
  - AND/OR graph sampling
  - w-cutset versions of the three schemes above

Results: Probability of Evidence Linkage instances (UAI 2006 evaluation)

Problem	$\langle n, k, E, t^*, c \rangle$	Exact	or-tree-IS Δ	ao-tree-IS Δ	ao-graph-IS Δ	or-wc-tree-IS Δ	ao-wc-tree-IS Δ	ao-wc-graph-IS Δ
BN_69.uai	$\langle 777, 7, 78, 47, 59 \rangle$	5.28E-54	2.26E-02	2.46E-02	2.43E-02	2.42E-02	2.34E-02	**4.22E-03**
BN_70.uai	$\langle 2315, 5, 159, 87, 98 \rangle$	2.00E-71	6.32E-02	7.25E-02	5.12E-02	8.18E-02	5.36E-02	**2.62E-02**
BN_71.uai	$\langle 1740, 6, 202, 70, 139 \rangle$	5.12E-111	6.74E-02	5.51E-02	2.35E-02	8.58E-02	**9.46E-03**	1.21E-02
BN_72.uai	$\langle 2155, 6, 252, 86, 88 \rangle$	4.21E-150	3.19E-02	4.61E-02	2.46E-03	6.12E-02	**1.41E-03**	2.63E-03
BN_73.uai	$\langle 2140, 5, 216, 101, 149 \rangle$	2.26E-113	1.18E-01	1.12E-01	4.55E-02	1.58E-01	**3.54E-02**	3.95E-02
BN_74.uai	$\langle 749, 6, 66, 45, 72 \rangle$	3.75E-45	5.34E-02	4.31E-02	2.87E-02	8.08E-02	2.83E-02	**2.76E-02**
BN_75.uai	$\langle 1820, 5, 155, 92, 131 \rangle$	5.88E-91	4.47E-02	8.15E-02	4.73E-02	7.28E-02	4.20E-02	**7.60E-03**
BN_76.uai	$\langle 2155, 7, 169, 64, 239 \rangle$	4.93E-110	1.07E-01	1.39E-01	6.95E-02	1.13E-01	5.03E-02	**2.26E-02**
BN_77.uai	$\langle 1020, 9, 135, 22, 97 \rangle$	6.88E-79	1.06E-01	9.38E-02	8.26E-02	1.24E-01	6.75E-02	**3.27E-02**

Time Bound: 1hr

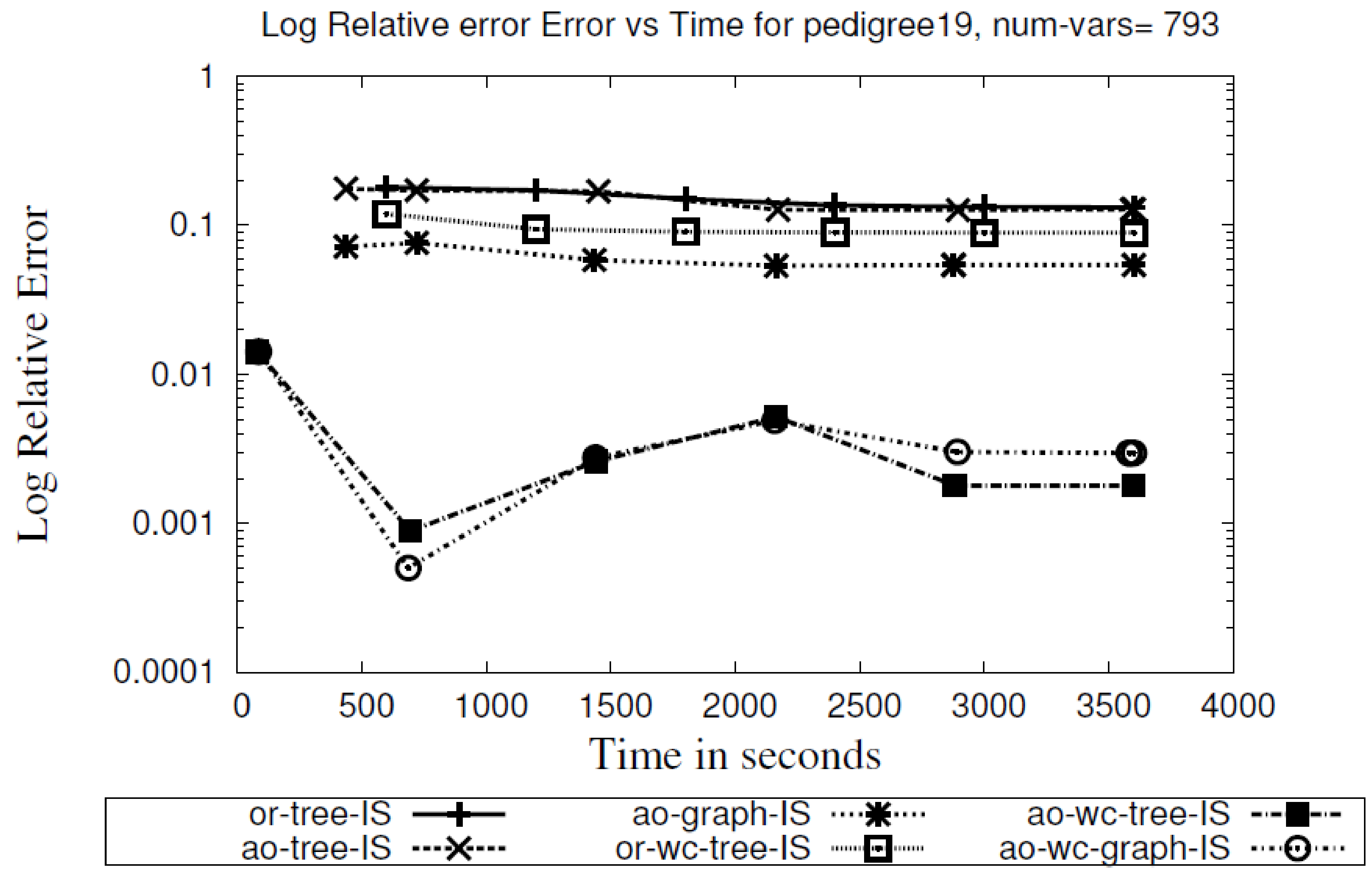


Log Relative error Error vs Time for BN_76, num-vars= 2155
Log Relative Error
1
0.1
0.01
0
500
1000
1500
2000
2500
3000
3500
4000
Time in seconds
or-tree-IS
ao-tree-IS
ao-graph-IS
or-wc-IS
ao-wc-tree-IS
ao-wc-graph-IS

Results: Probability of Evidence Linkage instances (UAI 2008 evaluation)

Problem	$\langle n, k, E, t^*, w \rangle$	Exact	or-tree-IS Δ	ao-tree-IS Δ	ao-graph-IS Δ	or-wc-tree-IS Δ	ao-wc-tree-IS Δ	ao-wc-graph-IS Δ
pedigree18.uai	$\langle 1184, 1, 0, 26, 72 \rangle$	4.19E-79	3.17E-02	3.44E-02	3.20E-03	4.30E-02	3.49E-04	**3.02E-04**
pedigree19.uai	$\langle 793, 2, 0, 23, 102 \rangle$	1.59E-60	1.32E-01	1.28E-01	5.41E-02	8.92E-02	**1.79E-03**	2.97E-03
pedigree1.uai	$\langle 334, 2, 0, 20, 27 \rangle$	7.81E-15	2.18E-03	1.90E-03	1.73E-04	3.15E-05	**7.61E-06**	1.13E-05
pedigree20.uai	$\langle 437, 2, 0, 25, 33 \rangle$	2.34E-30	1.52E-01	1.56E-01	2.12E-03	6.93E-02	**9.17E-04**	1.18E-03
pedigree23.uai	$\langle 402, 1, 0, 26, 29 \rangle$	2.00E-40	**2.62E-02**	2.74E-02	2.90E-02	2.82E-02	2.88E-02	2.88E-02
pedigree37.uai	$\langle 1032, 1, 0, 25, 36 \rangle$	2.63E-117	2.46E-02	3.50E-03	3.24E-03	1.45E-02	**3.00E-03**	3.01E-03
pedigree38.uai	$\langle 724, 1, 0, 18, 45 \rangle$	5.64E-55	4.08E-02	1.40E-02	1.25E-02	1.69E-02	8.91E-03	**8.79E-03**
pedigree39.uai	$\langle 1272, 1, 0, 29, 42 \rangle$	6.32E-103	8.67E-02	5.11E-02	1.72E-03	1.89E-02	2.31E-04	**2.13E-04**
pedigree42.uai	$\langle 448, 2, 0, 23, 50 \rangle$	1.73E-31	4.29E-03	1.94E-03	5.06E-04	1.11E-03	3.53E-05	**3.17E-05**
pedigree31.uai	$\langle 1183, 2, 0, 45, 118 \rangle$		1.09E-01	1.31E-01	4.15E-02	8.34E-02	**0.00E+00**	2.93E-04
pedigree34.uai	$\langle 1160, 1, 0, 59, 104 \rangle$		2.12E-01	1.47E-01	8.37E-02	8.09E-02	4.83E-04	**0.00E+00**
pedigree13.uai	$\langle 1077, 1, 0, 51, 98 \rangle$		3.93E-01	3.93E-01	5.66E-02	9.11E-02	1.51E-04	**0.00E+00**
pedigree41.uai	$\langle 1062, 2, 0, 52, 95 \rangle$		1.12E-01	5.06E-02	8.23E-04	5.04E-02	**0.00E+00**	3.15E-04
pedigree44.uai	$\langle 811, 1, 0, 29, 64 \rangle$		3.16E-02	3.08E-02	2.27E-03	1.90E-02	**0.00E+00**	4.63E-06
pedigree51.uai	$\langle 1152, 1, 0, 51, 106 \rangle$		9.22E-02	6.39E-02	2.26E-02	4.31E-02	9.35E-05	**0.00E+00**
pedigree7.uai	$\langle 1068, 1, 0, 56, 90 \rangle$		7.86E-02	9.98E-02	2.31E-02	4.61E-02	4.38E-04	**0.00E+00**
pedigree9.uai	$\langle 1118, 2, 0, 41, 80 \rangle$		3.29E-02	3.19E-02	**0.00E+00**	8.25E-02	9.74E-03	1.01E-02

Time Bound: 1hr



Log Relative error Error vs Time for pedigree19, num-vars= 793
Log Relative Error
1
0.1
0.01
0.001
0.0001
0
500
1000
1500
2000
2500
3000
3500
4000
Time in seconds
or-tree-IS
ao-tree-IS
ao-graph-IS
or-wc-tree-IS
ao-wc-tree-IS
ao-wc-graph-IS

Results: Solution counting Graph coloring instance

Problem	$\langle n, k, E, t^*, c \rangle$	Exact	or-tree-IS Δ	ao-tree-IS Δ	ao-graph-IS Δ	or-wc-tree-IS Δ	ao-wc-tree-IS Δ	ao-wc-graph-IS Δ
4-coloring1.uai	$\langle 400, 2, 0, 71, 309 \rangle$		3.82E-03	4.05E-03	4.51E-03	6.00E-03	2.35E-03	**0.00E+00**
4-coloring2.uai	$\langle 400, 2, 0, 95, 315 \rangle$		1.23E-02	9.54E-03	7.64E-03	3.38E-02	3.63E-02	**0.00E+00**
4-coloring3.uai	$\langle 800, 2, 0, 144, 617 \rangle$		2.86E-03	4.58E-03	2.32E-03	2.41E-02	2.38E-02	**0.00E+00**
4-coloring4.uai	$\langle 800, 2, 0, 191, 620 \rangle$		2.13E-02	5.06E-03	2.19E-02	1.79E-02	4.69E-03	**0.00E+00**
4-coloring5.uai	$\langle 1200, 2, 0, 304, 925 \rangle$		2.98E-02	2.81E-02	5.85E-02	5.70E-02	3.89E-02	**0.00E+00**
4-coloring6.uai	$\langle 1200, 2, 0, 338, 929 \rangle$		3.43E-02	2.72E-02	2.63E-03	3.17E-03	2.09E-03	**0.00E+00**

Time Bound: 1hr

Summary: AND/OR Importance sampling

- AND/OR sampling: A general scheme to exploit conditional independence in sampling
- Theoretical guarantees: lower sampling error than conventional sampling
- Variance reduction orthogonal to Rao-Blackwellised sampling.
- Better empirical performance than conventional sampling.