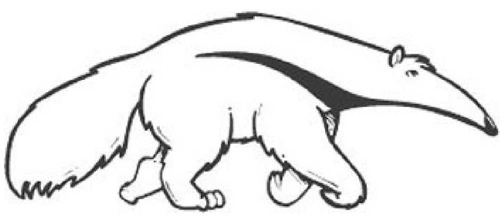
Algorithms for Causal Probabilistic Graphical Models

Class 4:

Sampling & Monte Carlo Methods

Athens Summer School on AI
July 2024

Prof. Rina Dechter
Prof. Alexander Ihler

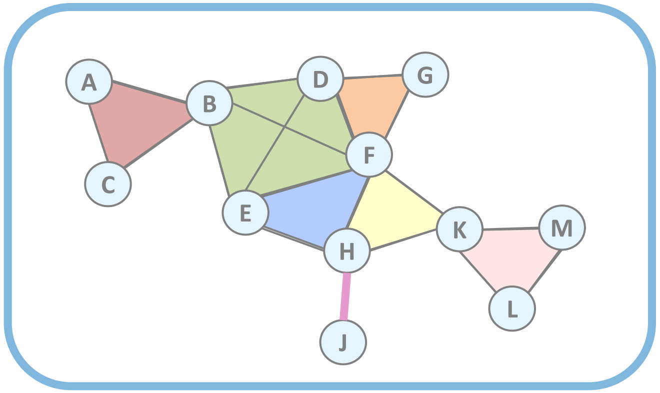


BREN:ICS INFORMATION AND COMPUTER SCIENCES

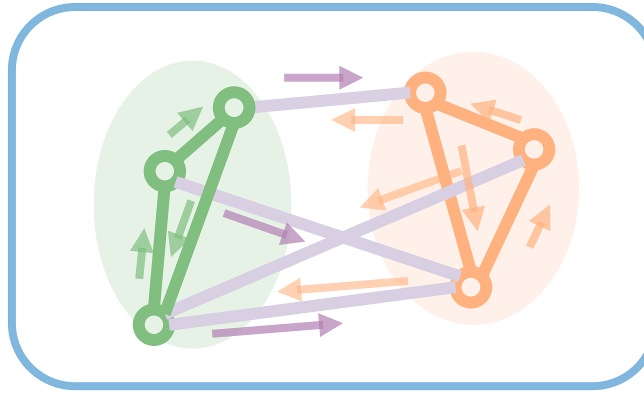
UNIVERSITY of CALIFORNIA IRVINE

Outline of Lectures

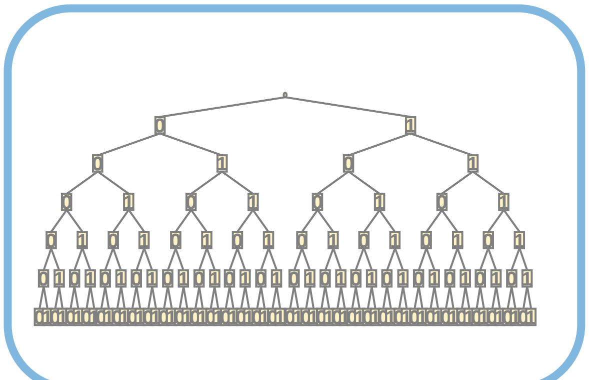
Class 1: Introduction & Inference



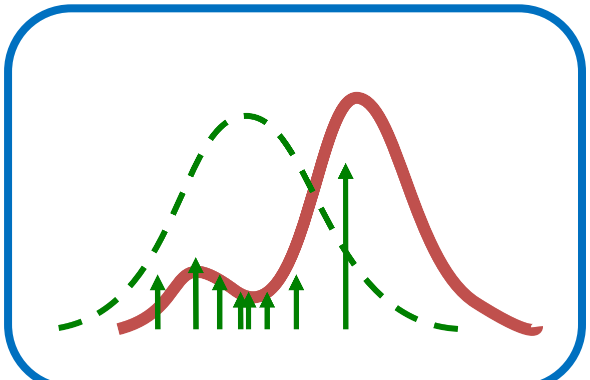
Class 2: Bounds & Variational Methods



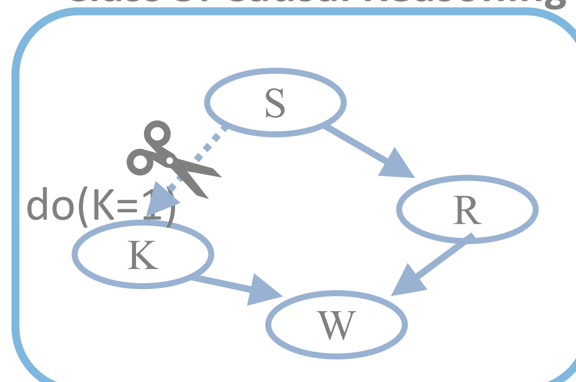
Class 3: Search Methods



Class 4: Monte Carlo Methods



Class 5: Causal Reasoning



Outline

- Monte Carlo: Basics
- Importance Sampling
- Stratified & Abstraction Sampling
- Markov Chain Monte Carlo
- Integrating Inference and Sampling

Graphical models

$A \in \{0,1\}$
$B \in \{0,1\}$
$C \in \{0,1\}$

$f_{AB}(A,B), \quad f_{BC}(B,C)$

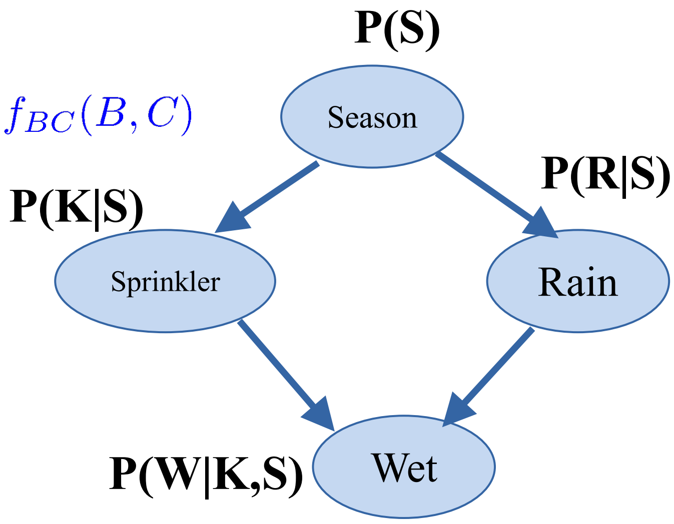
A *graphical model* consists of:

$X = \{X_1, \dots, X_n\}$ -- variables

$D = \{D_1, \dots, D_n\}$ -- domains (we'll assume discrete)

$F = \{f_{\alpha_1}, \dots, f_{\alpha_m}\}$ -- functions or "factors"

and a *combination operator*



The *combination operator* defines an overall function from the individual factors,

e.g., "*" : $P(S,K,R,W) = P(S) \cdot P(K|S) \cdot P(R|S) \cdot P(W|K,S)$

Notation:

Discrete Xi values called "states"

"Tuple" or "configuration": states taken by a set of variables

"Scope" of f: set of variables that are arguments to a factor f

often index factors by their scope, e.g., $f_\alpha(X_\alpha), \quad X_\alpha \subseteq X$

Probabilistic Reasoning Problems

- Exact inference time, space exponential in induced width
- Use randomness to help?

Max-Inference:	$f(x^*) = \max_x \prod_\alpha f_\alpha(x_\alpha)$
Sum-Inference: (e.g., causal effects)	$Z = \sum_x \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MMAP):	$f_M(x_M^*) = \max_{x_M} \sum_{x_S} \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MEU): (e.g., decisions, planning)	$\text{MEU} = \max_{D_1,\ldots,D_m} \sum_{X_1,\ldots X_n} (\prod_{P_i \in P} P_i) \times (\sum_{r_i \in R} r_i)$

(stochastic search)

(Monte Carlo)

(Monte Carlo Tree Search)

Harder

Monte Carlo estimators

- Most basic form: empirical estimate of probability

$$\mathbb{E}[u(x)] = \int p(x)u(x) \;\approx\; U = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim p(x)$$

- Relevant considerations
  - Able to sample from the target distribution p(x)?
  - Able to evaluate p(x) explicitly, or only up to a constant? $p(x|e) = \frac{p(x,e)}{p(e)}$

- "Any-time" properties
  - Unbiased estimator, $\mathbb{E}[U] = \mathbb{E}[u(x)]$

    or asymptotically unbiased, $\mathbb{E}[U] \to \mathbb{E}[u(x)]$ as $m \to \infty$

  - Variance of the estimator decreases with m

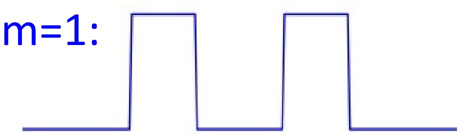
Monte Carlo estimators

- Most basic form: empirical estimate of probability

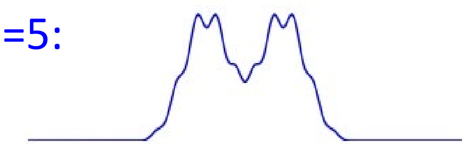
$$\mathbb{E}[u(x)] = \int p(x) u(x) \;\approx\; U = \frac{1}{m} \sum_i u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim p(x)$$

- Central limit theorem
  - p(U) is asymptotically Gaussian:

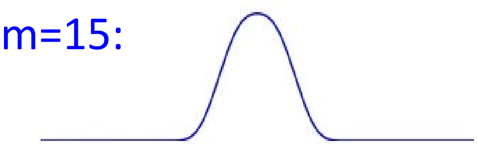
m=1:



m=5:

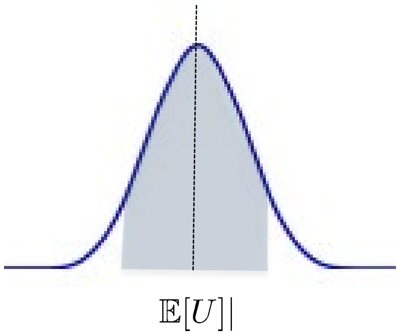


m=15:



- Finite sample confidence intervals
  - If u(x) or its variance are bounded, e.g., $u(x^{(i)}) \in [0,1]$ probability concentrates rapidly around the expectation:

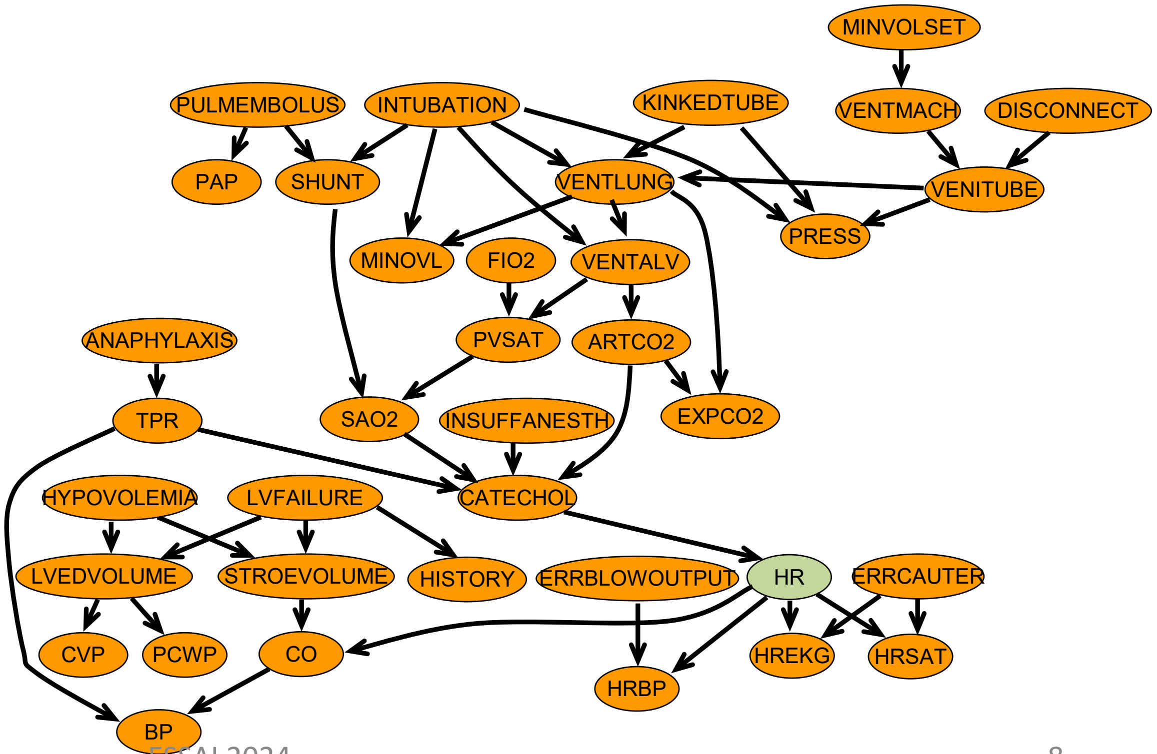
$$\Pr\big[\, |U - \mathbb{E}[U]| > \epsilon \,\big] \le O\big(\, \exp(-m\epsilon^2) \,\big)$$



Example: Alarm network

[Beinlich et al., 1989]

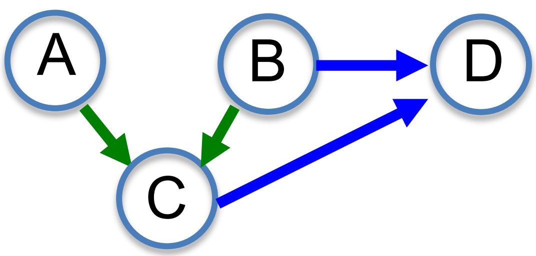
- Estimate p(HR=1)?
  - Implicitly defined by model's other probabilities
  - But, easy to estimate p(X) from samples!
  - And, samples are easy to generate!
  - Draw values for any roots; then their children…



Sampling in Bayes nets [e.g., Henrion 1988]

- No evidence: "causal" form makes sampling easy
  - Follow variable ordering defined by parents
  - Starting from root(s), sample downward
  - When sampling each variable, condition on values of parents

$$p(A, B, C, D) = p(A)\ p(B)\ p(C \mid A, B)\ p(D \mid B, C)$$



Sample:

$$a \sim p(A)$$
$$b \sim p(B)$$
$$c \sim p(C \mid A = a, B = b)$$
$$d \sim p(D \mid C = c, B = b)$$

Bayes nets with evidence

- Estimating the probability of evidence, P[E=e]:

$$\mathrm{P}[E=e] = \mathbb{E}[\,\mathbb{1}[E=e]\,] \approx \; U = \frac{1}{m}\sum_i \mathbb{1}[\tilde{e}^{(i)} = e]$$

  – Finite sample bounds: u(x) 2 [0,1]   [e.g., Hoeffding]

$$\Pr\Big[|U - \mathbb{E}[U]| > \epsilon\Big] \le 2\exp(-2m\epsilon^2)$$

What if the evidence is unlikely? P[E=e]=1e-6) could estimate U = 0!

  – Relative error bounds   [Dagum & Luby 1997]

$$\Pr\Big[\frac{|U - \mathbb{E}[U]|}{\mathbb{E}[U]} > \epsilon\Big] \le \delta \quad \text{if} \quad m \ge \frac{4}{\mathbb{E}[U]\epsilon^2}\log\frac{2}{\delta}$$

Algorithm: Forward sampling

- Easy to draw samples from Bayes nets:

Algorithm 1 Forward sampling (no evidence)

Order o such that if X_j is a child of X_i, then $o[i] < o[j]$.
for $j = 1 \ldots m$ **do**
    for $i = o[1] \ldots o[n]$ **do**
        Sample $x_i^{(j)} \sim p(X_i | X_{pa_i} = x_{pa_i}^{(j)})$
Estimate $\hat{p}(X_i = a) = \#\{x_i^{(j)} = a\} \,/\, m$

- Samples can be used to estimate any expectation:

$$\mathbb{E}_p[F(x)] = \int p(x) F(x) \approx \frac{1}{m} \sum_j F(x^{(j)}) \qquad x^{(j)} \sim p(x)$$

  - Example: Pr(Xi = a) = E[1[Xi=a]]

Bayes nets with evidence

- Estimating the probability of evidence, P[E=e]:

$$\mathrm{P}[E=e] = \mathbb{E}[\,\mathbb{1}[E=e]\,] \approx \; U = \frac{1}{m}\sum_i \mathbb{1}[\tilde{e}^{(i)} = e]$$

  – Finite sample bounds: u(x) 2 [0,1] [e.g., Hoeffding]

$$\Pr\Big[|U - \mathbb{E}[U]| > \epsilon\Big] \le 2\exp(-2m\epsilon^2)$$

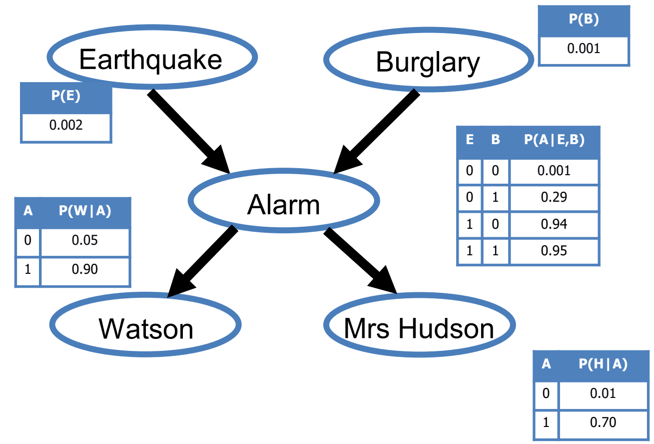
What if the evidence is unlikely? P[E=e]=1e-6) could estimate U = 0!

  – Relative error bounds [Dagum & Luby 1997]

$$\Pr\Big[\frac{|U - \mathbb{E}[U]|}{\mathbb{E}[U]} > \epsilon\Big] \le \delta \quad \text{if} \quad m \ge \frac{4}{\mathbb{E}[U]\epsilon^2}\log\frac{2}{\delta}$$

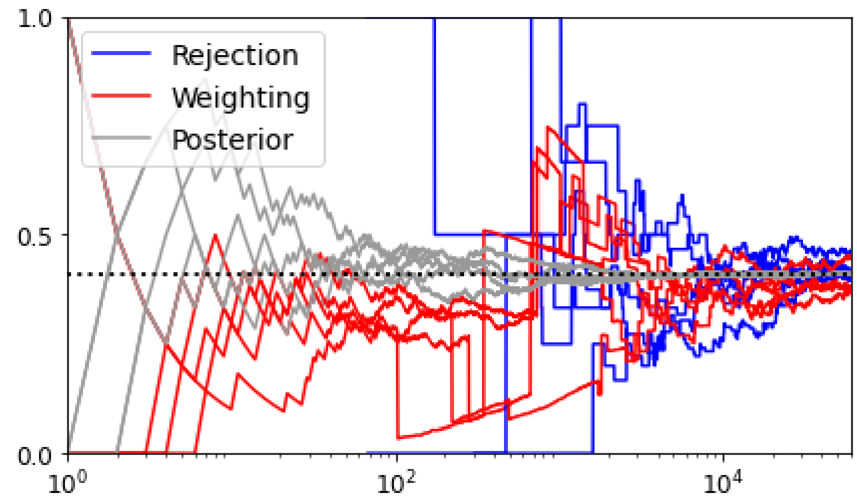
Ex: Burglary Model

What is p(E|W=1)?



- Rejection sampling
  - Discard many samples with W=0
- "Likelihood weighting"
  - Just "set" W=1
  - Now sampling E=0,W=1 too often!
  - Weight samples to adjust
- Want to draw E=1 more often!
  - Exact sampling: use inference

  (same work as just finding the answer?)



Exact sampling via inference

- Draw samples from P[X|E=e] directly?
  - Model defines un-normalized p(X_1,…,E=e)
  - Build (oriented) tree decomposition & sample

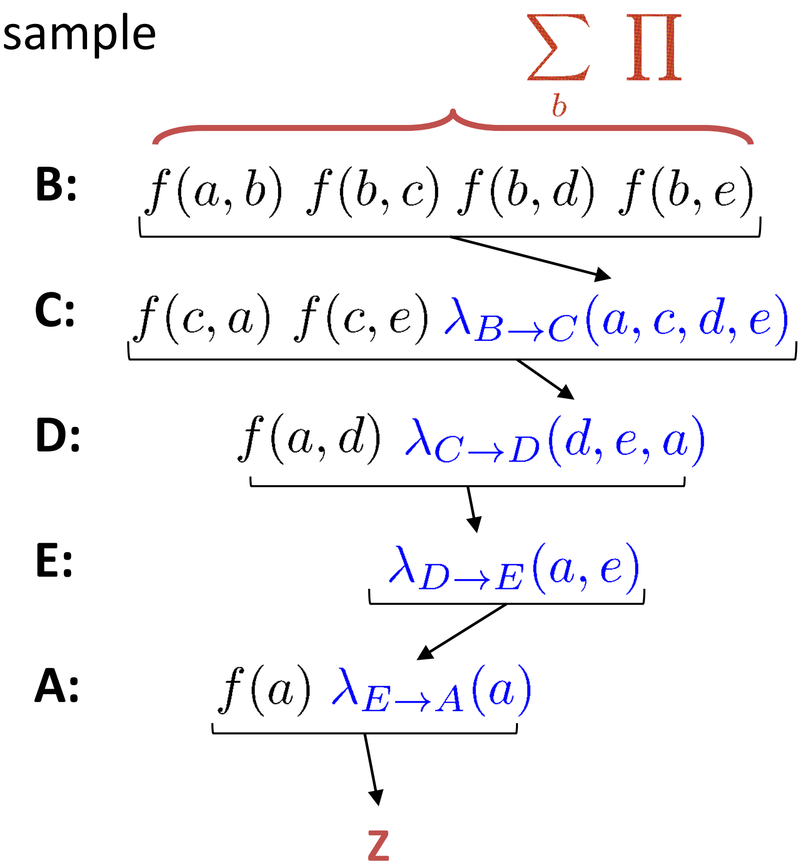
$$\tilde{\mathbf{b}} \sim f(\tilde{a}, b) \cdot f(b, \tilde{c}) \cdot f(b, \tilde{d}) \cdot f(b, \tilde{e}) / \lambda_{B \to C}$$

$$\tilde{\mathbf{c}} \sim f(c, \tilde{a}) \cdot f(c, \tilde{e}) \cdot \lambda_{B \to C}(\tilde{a}, c, \tilde{d}, \tilde{e}) / \lambda_{C \to D}$$

$$\tilde{\mathbf{d}} \sim f(\tilde{a}, d) \cdot \lambda_{B \to D}(d, \tilde{e}) / \lambda_{D \to E}(\tilde{a}, \tilde{e})$$

$$\tilde{\mathbf{e}} \sim \lambda_{D \to E}(\tilde{a}, e) / \lambda_{E \to A}(\tilde{a})$$

$$\tilde{\mathbf{a}} \sim p(A) = f(a) \cdot \lambda_{E \to A}(a) / Z$$



Downward message normalizes bucket;
ratio is a conditional distribution

Work: O(exp(w)) to build distribution
O(n d) to draw each sample

Outline

- Monte Carlo: Basics
- **Importance Sampling**
- Stratified & Abstraction Sampling
- Markov Chain Monte Carlo
- Integrating Inference and Sampling

Importance Sampling

- Basic empirical estimate of probability:

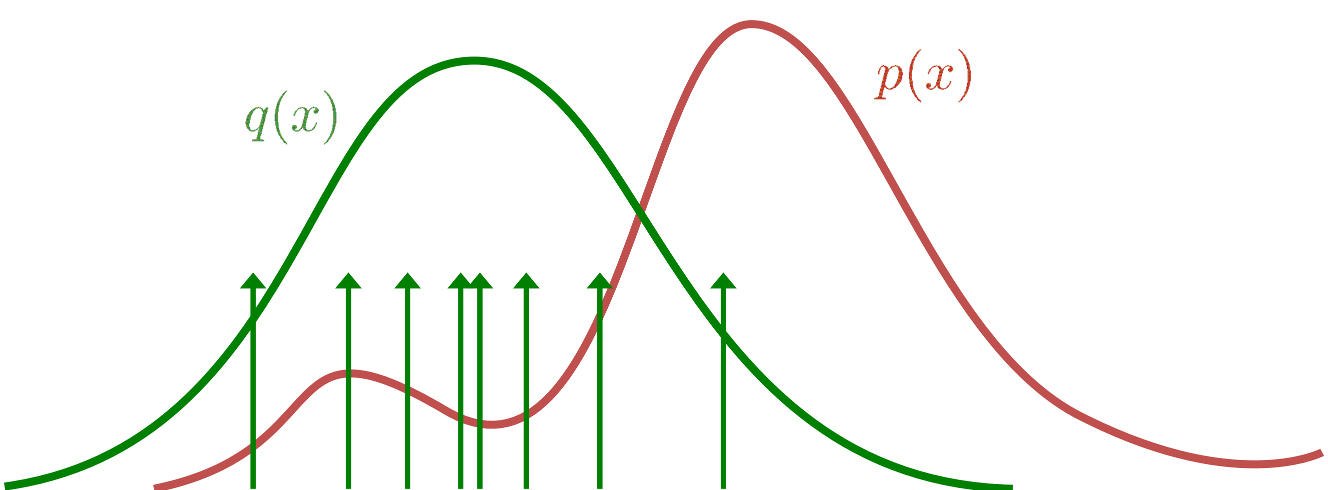
$$\mathbb{E}[u(x)] = \int p(x)u(x) \approx \hat{u} = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim p(x)$$

What if we can't sample from p(.) easily?

- Importance sampling:

$$\int p(x)u(x) = \int q(x)\frac{p(x)}{q(x)}u(x) \approx \frac{1}{m}\sum_i \frac{p(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})}u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim q(x)$$

q(.): easy to sample from



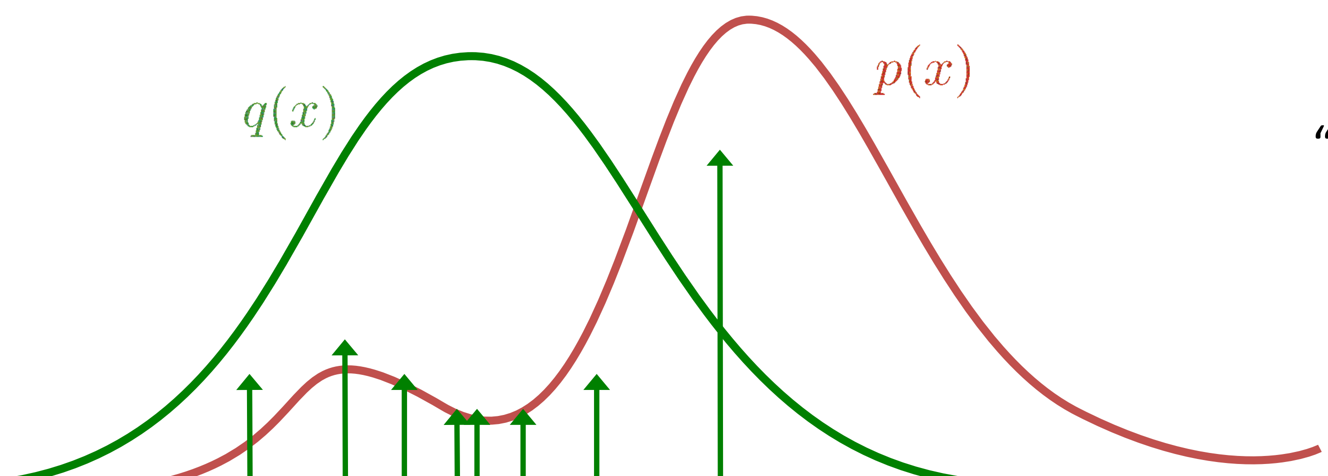
Importance Sampling

- Basic empirical estimate of probability:

$$\mathbb{E}[u(x)] = \int p(x)u(x) \;\approx\; \hat{u} = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim p(x)$$

- Importance sampling:

$$\int p(x)u(x) = \int q(x)\frac{p(x)}{q(x)}u(x) \approx \frac{1}{m}\sum_i \frac{p(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})}u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim q(x)$$



"importance weights"

$$w^{(i)} = \frac{p(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})}$$

IS for common queries

- Partition function / Probability of Evidence

$$Z = \sum_x f(x) = \sum_x q(x) \frac{f(x)}{q(x)} = \mathbb{E}_q\left[\frac{f(x)}{q(x)}\right] \approx \frac{1}{m} \sum w^{(i)}$$

  - Unbiased; only requires evaluating unnormalized function f(x)

$$w^{(i)} = \frac{f(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})}$$

- General expectations wrt p(x|E) / p(x,E) = f(x)?
  - E.g., conditional marginal probabilities, etc.

$$\mathbb{E}_p[u(x)] = \sum_x u(x) \frac{f(x)}{Z} = \frac{\mathbb{E}_q[u(x)f(x)/q(x)]}{\mathbb{E}_q[f(x)/q(x)]} \approx \frac{\sum u(\tilde{x}^{(i)}) w^{(i)}}{\sum w^{(i)}}$$

Estimate separately

"self-normalized" IS: only asymptotically unbiased…

Importance Sampling

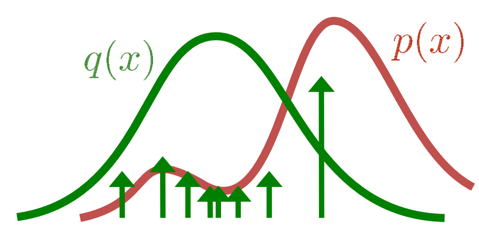
- Importance sampling:

$$\int p(x)u(x) = \int q(x)\frac{p(x)}{q(x)}u(x) \approx \frac{1}{m}\sum_i \frac{p(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})}u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim q(x)$$

- IS is unbiased and fast if q(.) is easy to sample from

- IS can be lower variance if q(.) is chosen well
  - Ex: q(x) puts more probability mass where u(x) is large
  - Optimal: q(x) / |u(x) p(x)|

- IS can also give poor performance
  - If q(x) << u(x) p(x): rare but very high weights!
  - Then, empirical variance is also unreliable!
  - For guarantees, need to analytically bound weights / variance…



Choosing a proposal

[Liu, Fisher, Ihler 2015]

- Can use WMB upper bound to define a proposal q(x):

$$\tilde{\mathbf{b}} \sim w_1\, q_1(b|\tilde{a},\tilde{c}) \;+\; w_2\, q_2(b|\tilde{d},\tilde{e})$$

Weighted mixture:

use minibucket 1 with probability w_1

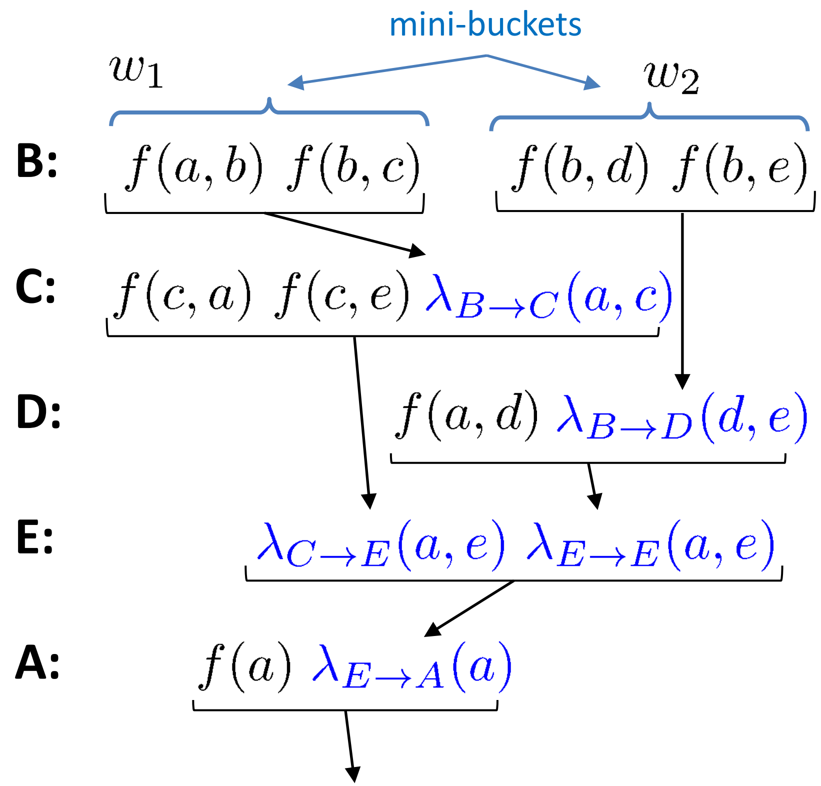
or, minibucket 2 with probability $w_2 = 1 - w_1$

where

$$q_1(b|a,c) = \left[\frac{f(a,b)\cdot f(b,c)}{\lambda_{B\to C}(a,c)}\right]^{\frac{1}{w_1}}$$

⋮

$$\tilde{\mathbf{a}} \sim q(A) = f(a)\cdot \lambda_{E\to A}(a)/U$$



U = upper bound

Key insight: provides bounded importance weights!

$$0 \le \frac{F(x)}{q(x)} \le U \qquad \forall x$$

WMB-IS Bounds

[Liu, Fisher, Ihler 2015]

- Finite sample bounds on the average

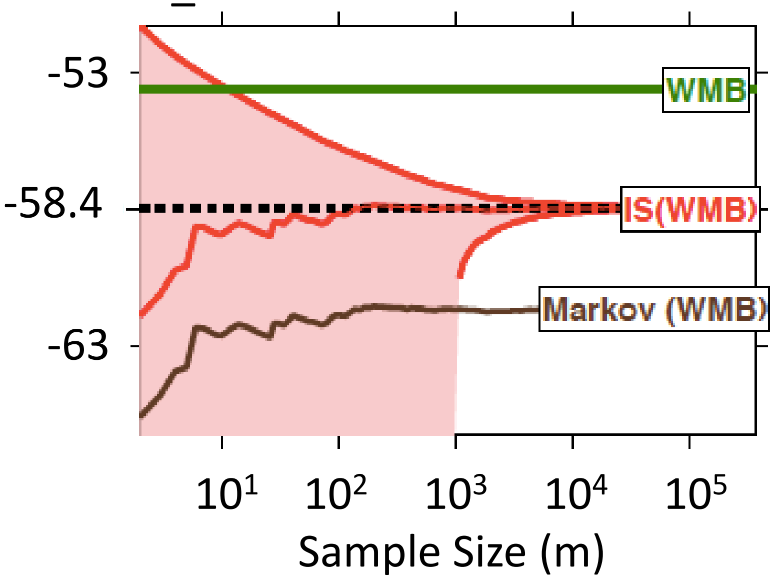
$$\Pr\left[|\hat{Z} - Z| > \epsilon\right] \leq 1 - \delta$$

$$\epsilon = \sqrt{\frac{2\hat{V}\log(4/\delta)}{m}} + \frac{7\,U\,\log(4/\delta)}{3(m-1)}$$

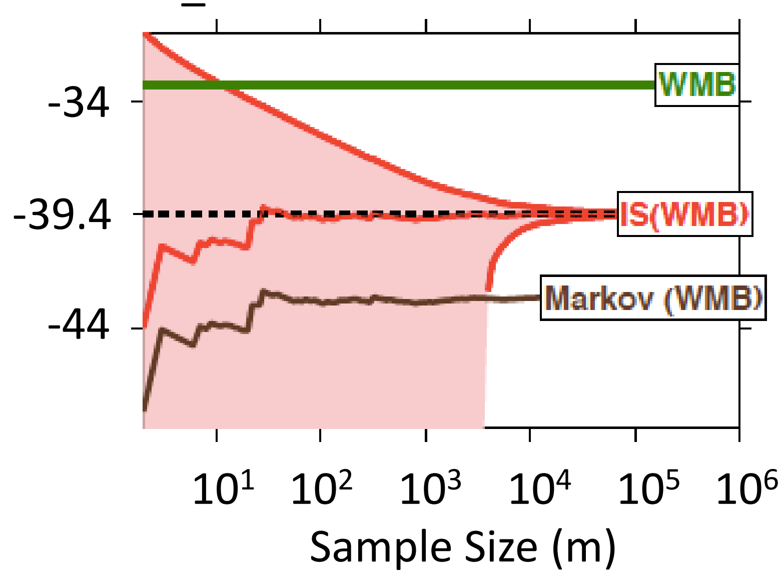
"Empirical Bernstein" bounds

- Compare to forward sampling
  - Works well if evidence "not too unlikely") not too much less likely than U

BN_6

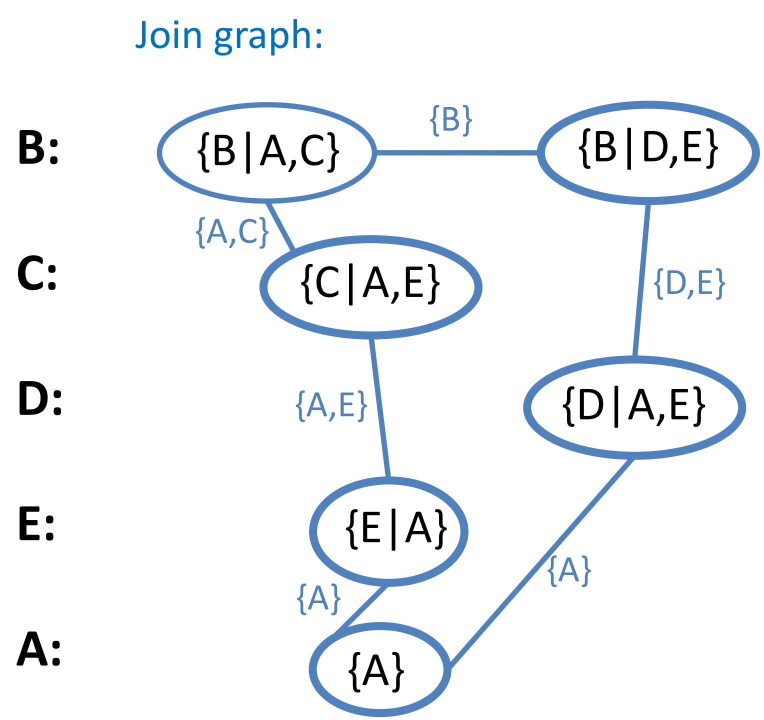


BN_11



Other choices of proposals

- Belief propagation
  - BP-based proposal [Changhe & Druzdzel 2003]
  - Join-graph BP proposal [Gogate & Dechter 2005]
  - Mean field proposal [Wexler & Geiger 2007]



Other choices of proposals

- Belief propagation
  - BP-based proposal [Changhe & Druzdzel 2003]
  - Join-graph BP proposal [Gogate & Dechter 2005]
  - Mean field proposal [Wexler & Geiger 2007]

- Adaptive importance sampling
  - Use already-drawn samples to update q(x)
  - Rates v_t and ´$_t$ adapt estimates, proposal
  - Ex:

    [Cheng & Druzdzel 2000]

    [Lapeyre & Boyd 2010]

    …
  - Lose "iid"-ness of samples

Adaptive IS

Initialize $q_0(x)$

for $t = 0 \ldots T$ **do**

  Draw $\tilde{X}_t = \{\tilde{x}^{(i)}\} \sim q_t(x)$

  $U_t = \frac{1}{m_t} \sum f(\tilde{x}^{(i)}) / q_t(\tilde{x}^{(i)})$

  $\hat{U} = (1 - v_t)\hat{U} + v_t U_t$

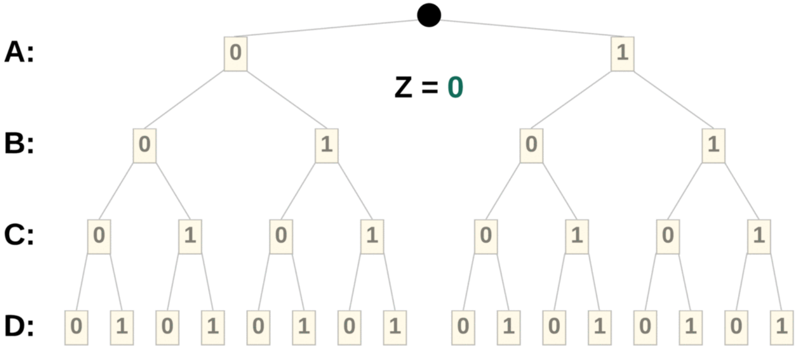
  $q_{t+1} = (1 - \eta_t) q_t + \eta_t q^*(X_t)$

Outline

- Monte Carlo: Basics
- Importance Sampling
- **Stratified & Abstraction Sampling**
- Markov Chain Monte Carlo
- Integrating Inference and Sampling

Systematic Search vs Sampling

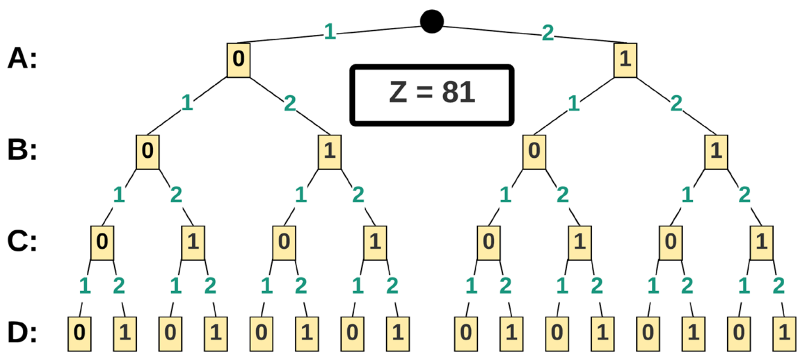
Systematic Search



- Enumerate states
- Every stone turned
- No stone turned more than once

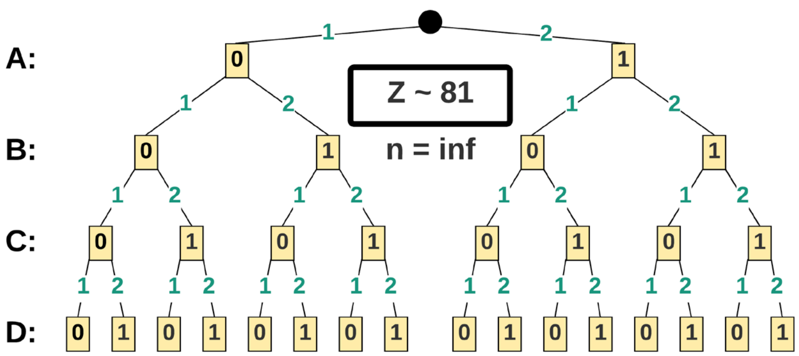
Systematic Search vs Sampling

Systematic Search



- Enumerate states
- Every stone turned
- No stone turned more than once

Importance Sampling

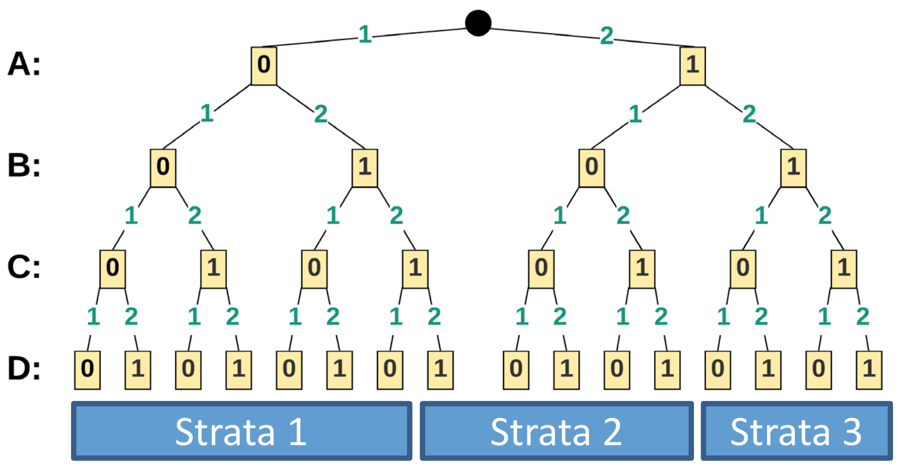


- Exploit "typicality" via randomization
- Concentration inequalities

Stratified Sampling

[Knuth, 1975; Chen, 1992; Rizzo, 2007]

- Organize states into groups ("strata")
  - Enumerate over strata
  - Importance sampling within each strata
- Reduces estimate variance
- Intermediate
  - Part search, part sampling
- "Ensemble" Monte Carlo
  - Draw multiple samples together
  - Samples are anti-correlated

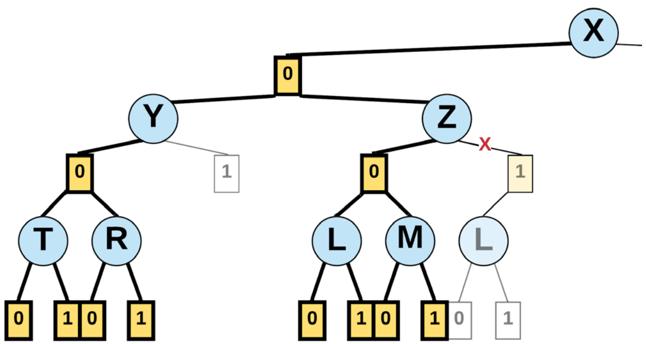


Abstraction Sampling

[Broka et al. 2018, Kask et al. 2020, Pezeshki et al. 2024]

- View ensemble of samples as a search sub-tree
  - Draw probe level by level
  - Use stratified sampling at each stage
- Exploit AND/OR search tree structure
  - Probe compactly represents many states
- Abstraction function defines strata
  - An area of ongoing development

AND/OR Abstraction Probe:
11 nodes representing
16 joint configurations



Outline

- Monte Carlo: Basics
- Importance Sampling
- Stratified & Abstraction Sampling
- **Markov Chain Monte Carlo**
- Integrating Inference and Sampling

MCMC Sampling

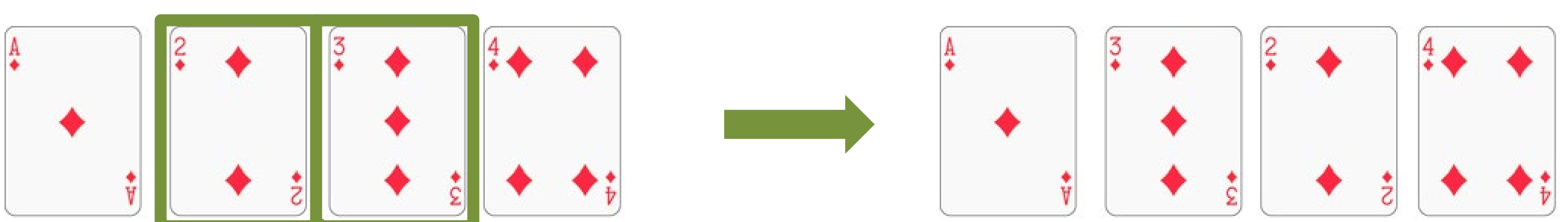
- Recall: Basic empirical estimate of probability:

$$\mathbb{E}[u(x)] = \int p(x)u(x) \approx \hat{u} = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim p(x)$$

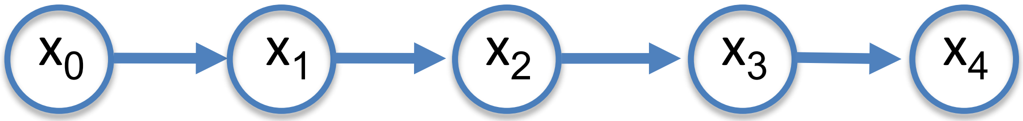
What if we can't sample from p(.) easily?

- Can we design a procedure to sample from p(x) anyway?

- Example: card shuffling
  - Want: a uniform distribution over card deck orders. How?
  - Create a "process" that converges to the right distribution
  - Ex: pick two cards at random & swap them with probability 1/2:
    - How do we know this will converge to the right distribution?

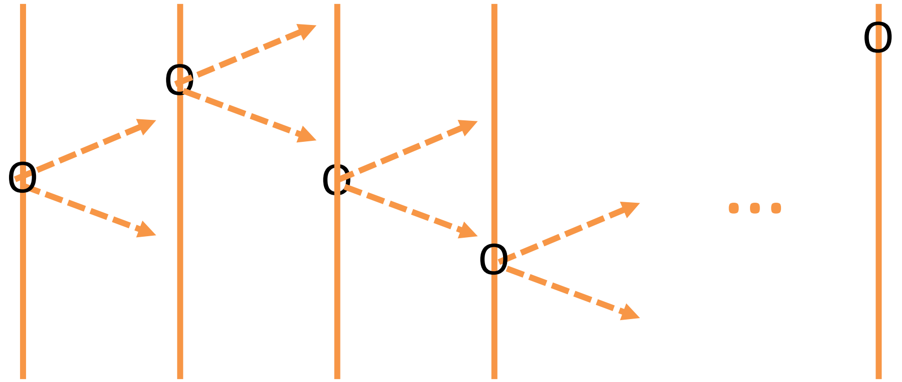


Markov Chains



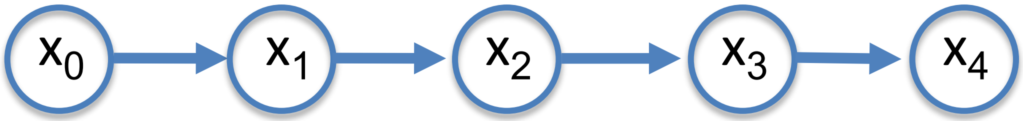
- Temporal model
  - State at each time t
  - "Markov property": state at time t depends only on state at t-1
  - "Homogeneous" (in time): $p(X_t \mid X_{t-1}) = T(X_t \mid X_{t-1})$ does not depend on t

- Example: random walk
  - Time 0: $x_0 = 0$
  - Time t: $x_t = x_{t-1}$ § 1

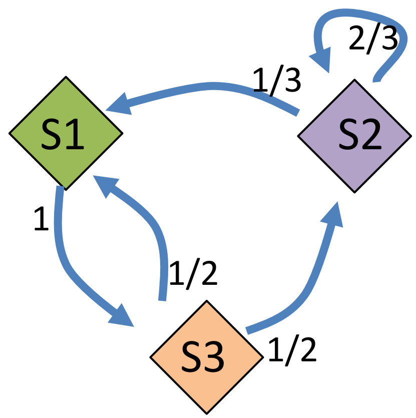


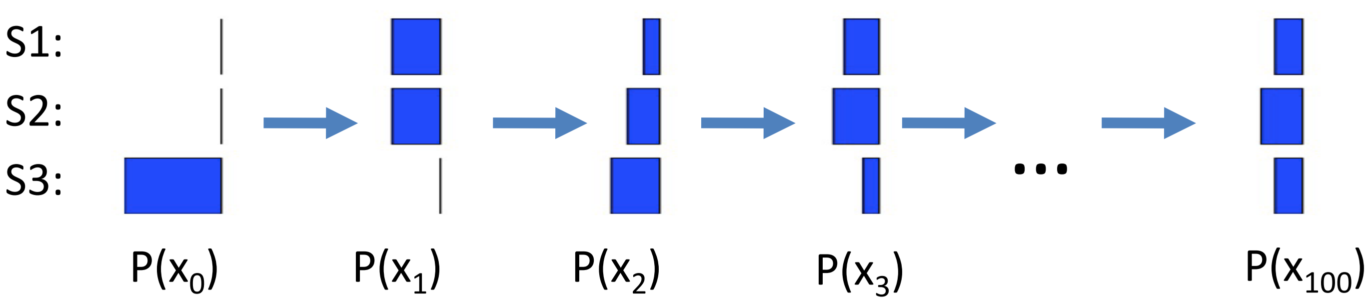
Markov Chains

- Temporal model
  - State at each time t
  - "Markov property": state at time t depends only on state at t-1
  - "Homogeneous" (in time): $p(X_t \mid X_{t-1}) = T(X_t \mid X_{t-1})$ does not depend on t



- Example: finite state machine
  - Time 0: x_0 = S3
  - Ex: S3 ! S1 ! S3 ! S2 ! …
  - What is $p(x_t)$? Does it depend on x_0?

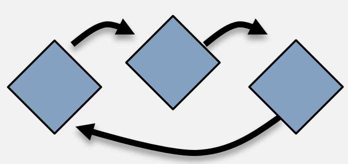




Stationary distributions

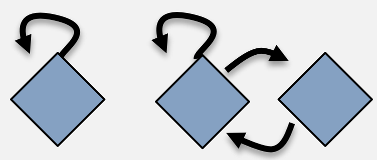
- Stationary distribution $s(x)$: $s(x_{t+1}) = \sum_{x_t} p(x_{t+1} \mid x_t) s(x_t)$
- $p(x_t)$ becomes independent of $p(x_0)$?
- Sufficient conditions for s(x) to exist and be unique:
  - (a) p(. | .) is acyclic: $\gcd\{t \;:\; \Pr[x_t = s_i \mid x_0 = s_i] > 0\} = 1$
  - (b) p(. | .) is irreducible: $\forall i, j \; \exists t \;:\; \Pr[x_t = s_i \mid x_0 = s_j] > 0$

Ex: not (a)



s(x) may not exist

Ex: not (b)



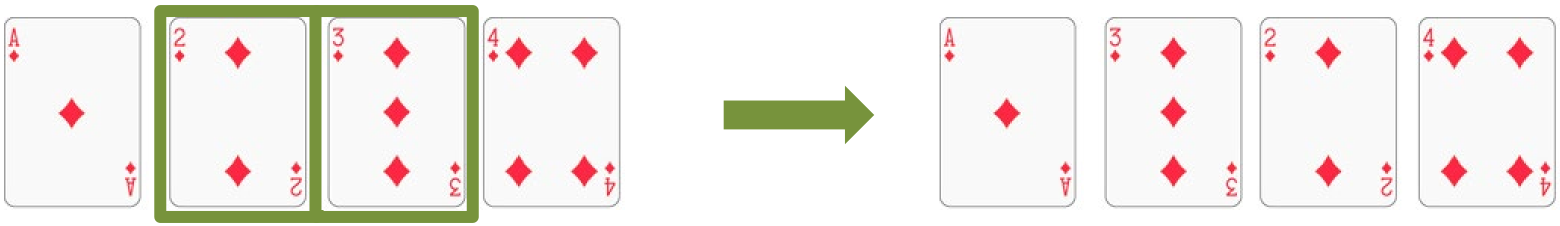
multiple s(x) exist

Without both (a) & (b), long-term probabilities may depend on the initial distribution

Stationary distributions

- Uniqueness of the stationary distribution is powerful

- Recall: simple shuffling

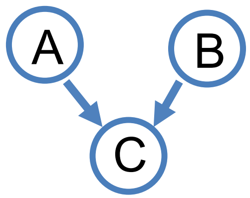


- Irreducible?
  - Yes: there is a path between any two orderings
- Acyclic?
  - Yes: if there is a path of length L, there is also one of length L+1, L+2, …

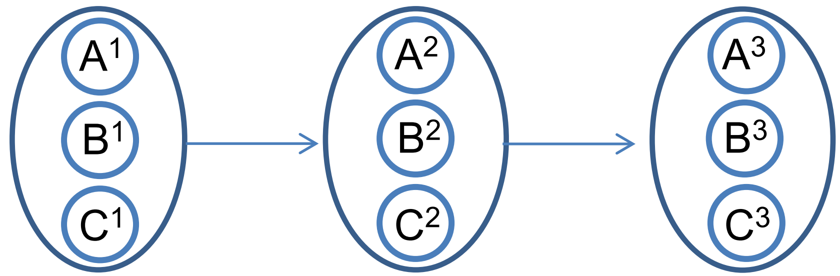
- So, the stationary distribution is unique!
  - Now just show that “uniform over orders” is a stationary dist…

Markov Chain Monte Carlo

- Method for generating samples from an intractable p(x)
  - Create a Markov chain whose stationary distribution equals p(x)



State "x":
Complete config.
of target model



  - Sample $x^{(1)} \ldots x^{(m)}$; $x^{(m)} \sim p(x)$ if m sufficiently large
  - Two common methods:

- **Metropolis sampling**
  - Propose a new point x' using q(x' | x) ; depends on current point x
  - Accept with carefully chosen probability, a(x',x)

- Gibbs sampling
  - Sample each variable in turn, given values of all the others

Metropolis-Hastings

- At each step, propose a new value $x' \sim q(x'|x)$
- Decide whether we should move there
  - If $p(x') > p(x)$, it's a higher probability region (good)
  - If $q(x|x') < q(x'|x)$, it will be hard to move back (bad)

  - Accept move with a carefully chosen probability:

$$a(x', x) = \min\left[1 \, , \, \frac{p(x')q(x|x')}{p(x)q(x'|x)}\right]$$

Probability of "accepting" the move from x to x'; otherwise, stay at state x.

Ratio p(x') / p(x) means that we can substitute an unnormalized distribution f(x) if needed

  - The resulting transition probability $T(x'|x) = q(x'|x)\, a(x', x)$ has *detailed balance* with p(x), a sufficient condition for stationarity

Detailed balance in Markov chains

- Detailed balance: $s(x')\,T(x|x') \;=\; s(x)\,T(x'|x)$
  - Mass moving from i to j at steady-state equals mass moving from j to i
  - A sufficient condition for s(.) to be the stationary dist.

$$\sum_x s(x')\,T(x|x') = s(x') \;=\; \sum_x s(x)\,T(x'|x)$$

- Metropolis-Hastings:
  - Transition depends on propose & accept: $T(x'|x) = q(x'|x)\,a(x',x)$

$$\Longrightarrow \quad p(x')\,q(x|x')\,a(x,x') = p(x)\,q(x'|x)\,a(x',x)$$

$$\Longrightarrow \quad \frac{a(x',x)}{a(x,x')} = \frac{p(x')\,q(x|x')}{p(x)\,q(x'|x)}$$

If less than 1: assign to a(x' , x)
greater than 1: assign to a(x , x')

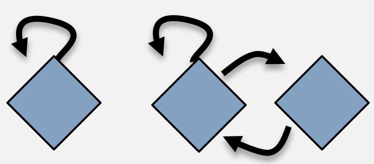
$$\Longrightarrow \quad a(x',x) = \min\left[1\;,\;\frac{p(x')q(x|x')}{p(x)q(x'|x)}\right]$$

Mixing Rate

- How quickly do approach the stationary distribution?
  - Rate to get a sample from p(x)
  - Rate of independent samples (forget previous value)

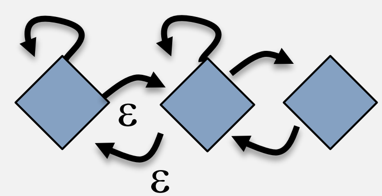
- Depends on the transitions of the Markov chain

Not irreducible



Multiple s(x) exist

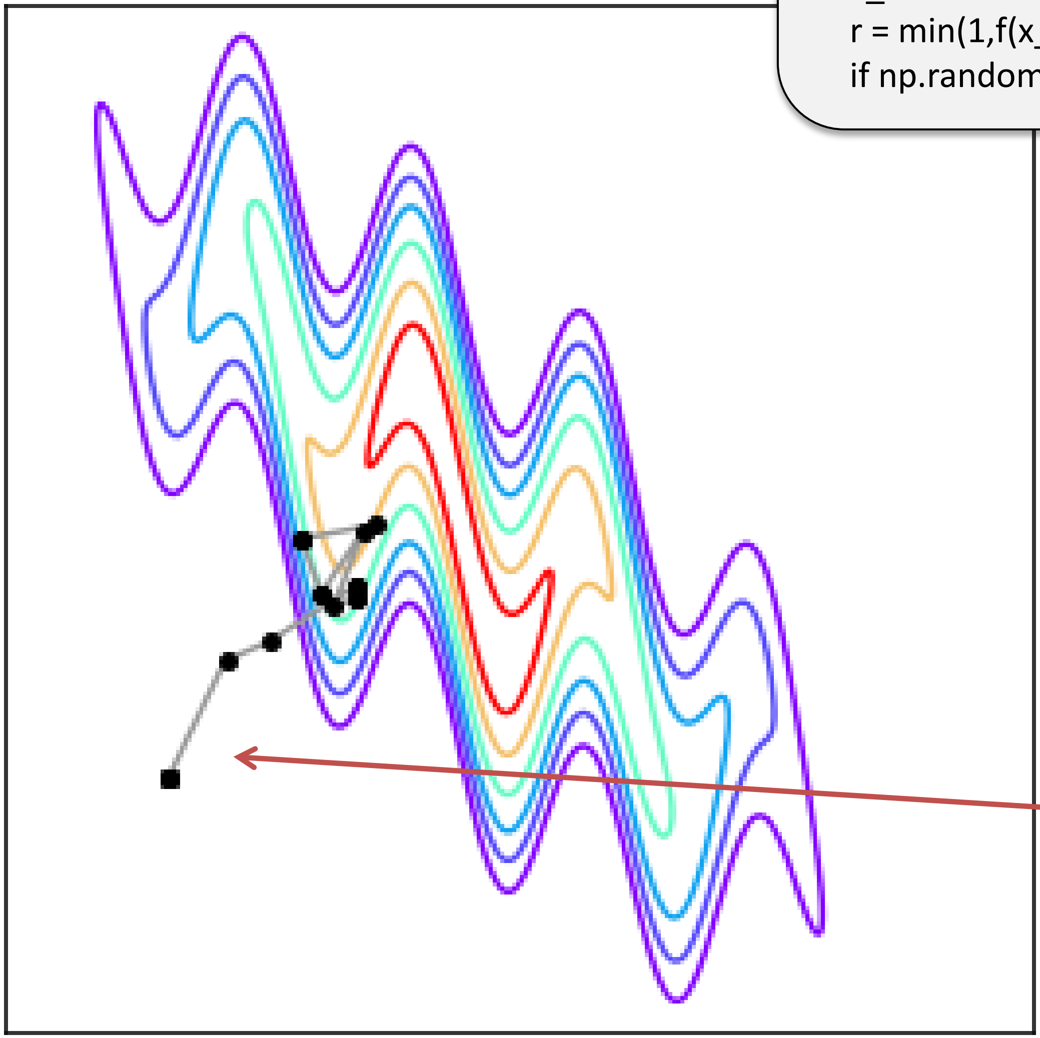
"Barely" irreducible



Unique s(x), but slow!

MCMC Example

T = 25



Metropolis-Hastings (symmetric proposal)

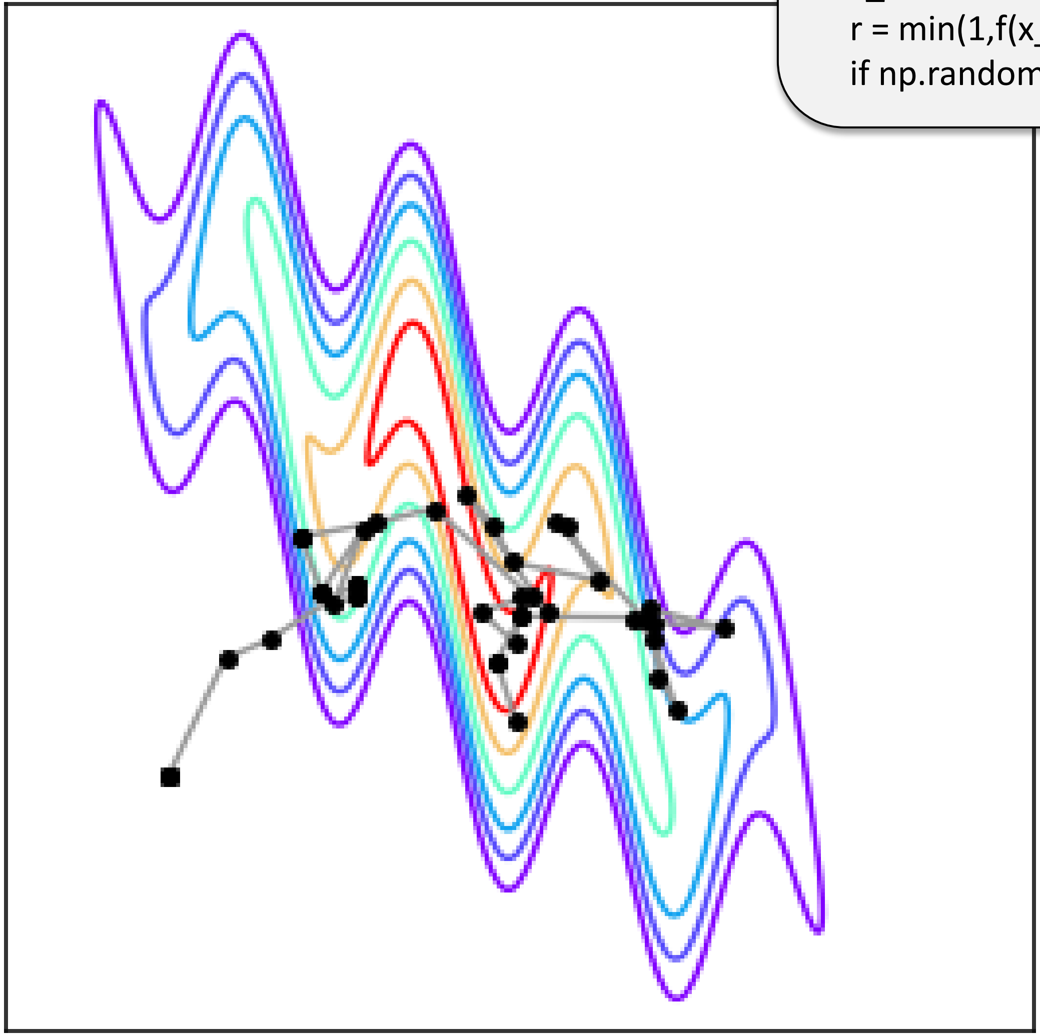
```
f = lambda X: ...        % define f(x) / p(x), target
x = np.zeros((1,2)); % set or sample initial state
for t in range(T):       % simulate Markov chain:
    x_ = x + .5*np.random.randn(1,2)    % propose move
    r = min(1,f(x_)/f(x))                % compute acceptance
    if np.random.rand() < r: x = x_      % sample acceptance
```

Early samples depend on initialization

"Burn in"; may discard these samples

MCMC Example

T = 50

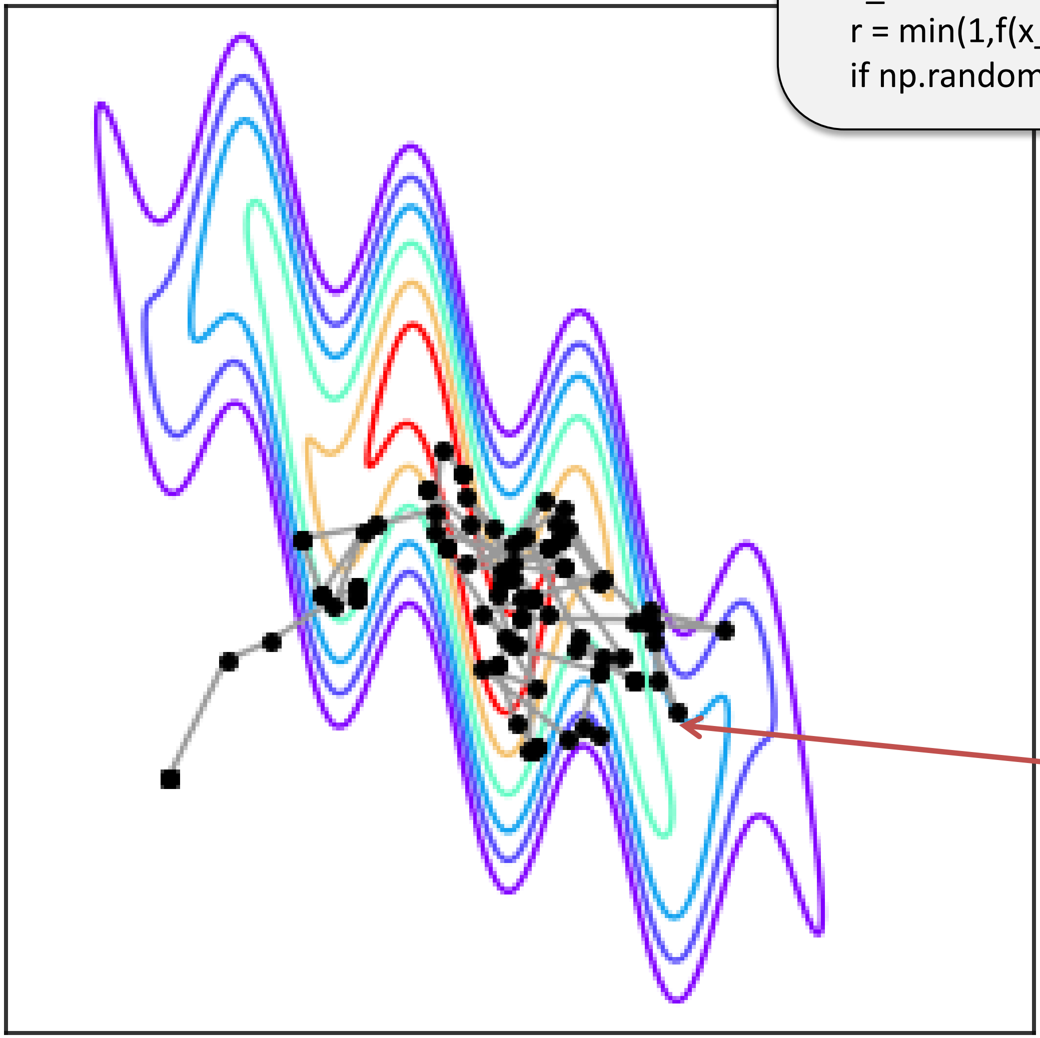


Metropolis-Hastings (symmetric proposal)

```
f = lambda X: ...        % define f(x) / p(x), target
x = np.zeros((1,2));     % set or sample initial state
for t in range(T):       % simulate Markov chain:
    x_ = x + .5*np.random.randn(1,2)    % propose move
    r = min(1,f(x_)/f(x))               % compute acceptance
    if np.random.rand() < r: x = x_     % sample acceptance
```

MCMC Example

T = 100



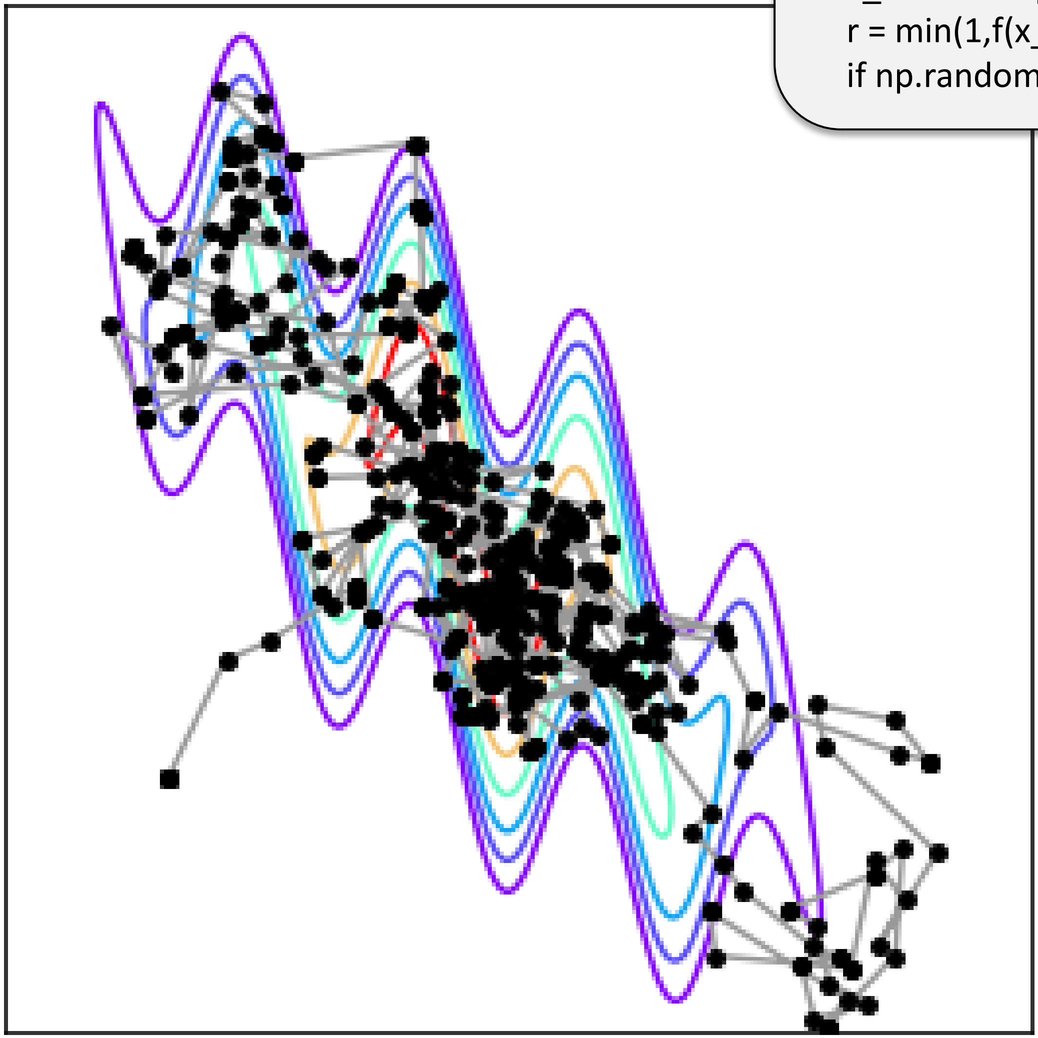
Metropolis-Hastings (symmetric proposal)

```
f = lambda X: ...       % define f(x) / p(x), target
x = np.zeros((1,2)); % set or sample initial state
for t in range(T):     % simulate Markov chain:
   x_ = x + .5*np.random.randn(1,2)     % propose move
   r = min(1,f(x_)/f(x))                 % compute acceptance
   if np.random.rand() < r: x = x_       % sample acceptance
```

Samples correlated
in time
(not independent)

MCMC Example

T = 500

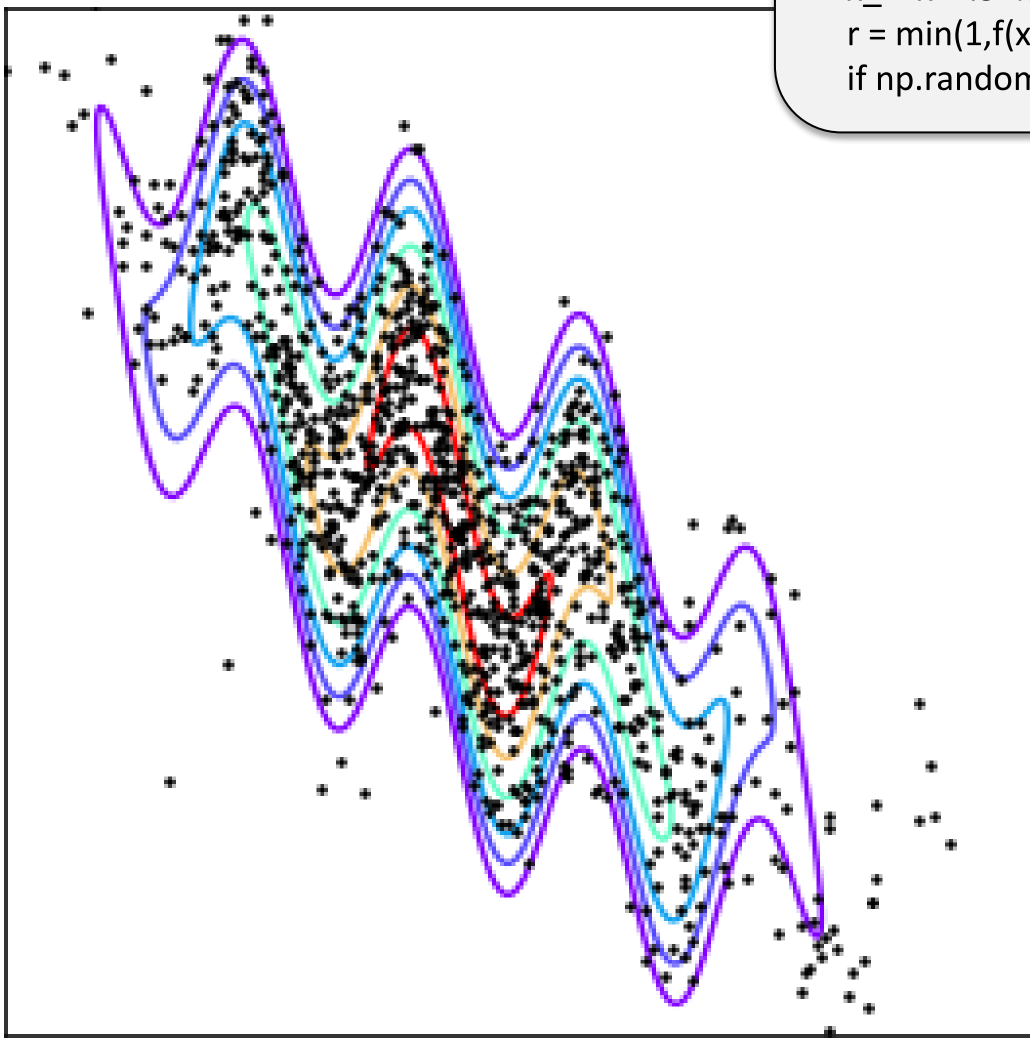


Metropolis-Hastings (symmetric proposal)

```
f = lambda X: ...        % define f(x) / p(x), target
x = np.zeros((1,2)); % set or sample initial state
for t in range(T):       % simulate Markov chain:
    x_ = x + .5*np.random.randn(1,2)     % propose move
    r = min(1,f(x_)/f(x))                 % compute acceptance
    if np.random.rand() < r: x = x_       % sample acceptance
```

MCMC Example

T = 10000 (subsampled by 10)



Metropolis-Hastings (symmetric proposal)

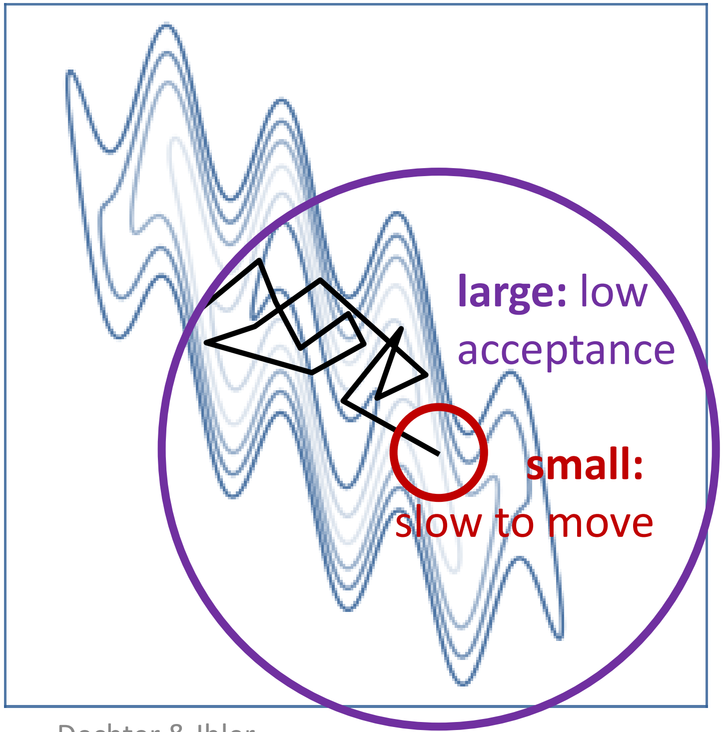
```
f = lambda X: ...       % define f(x) / p(x), target
x = np.zeros((1,2)); % set or sample initial state
for t in range(T):     % simulate Markov chain:
   x_ = x + .5*np.random.randn(1,2)     % propose move
   r = min(1,f(x_)/f(x))                 % compute acceptance
   if np.random.rand() < r: x = x_       % sample acceptance
```

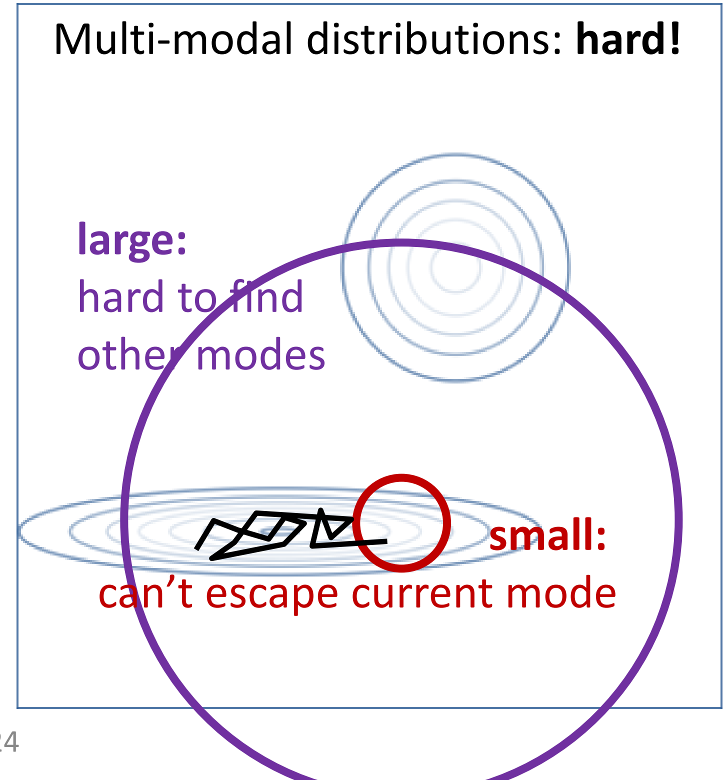
Asymptotically, samples will represent p(x)

May choose to “decimate” (keep only every kth sample), for memory/storage reasons

Mixing behavior

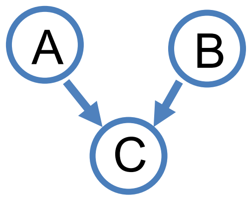
- What makes MCMC mix slowly?
- Transition proposal is:
  - too small? Can't change the state much!
  - too large? Try states with low probability; reject: same state!



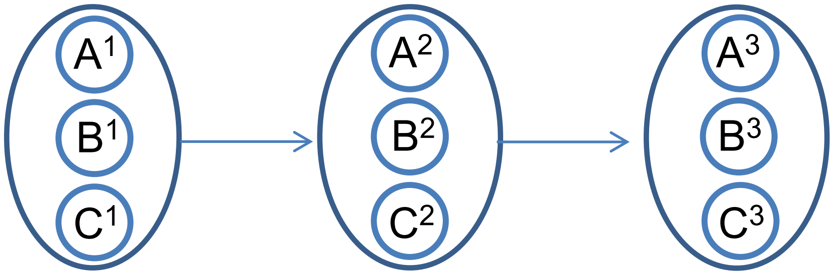


Markov Chain Monte Carlo

- Method for generating samples from an intractable p(x)
  - Create a Markov chain whose stationary distribution equals p(x)



State “x”:
Complete config.
of target model



  - Sample $x^{(1)} \ldots x^{(m)}$; $x^{(m)} \sim p(x)$ if m sufficiently large
  - Two common methods:

- Metropolis sampling
  - Propose a new point x' using q(x' | x) ; depends on current point x
  - Accept with carefully chosen probability, a(x',x)
- **Gibbs sampling**
  - Sample each variable in turn, given values of all the others

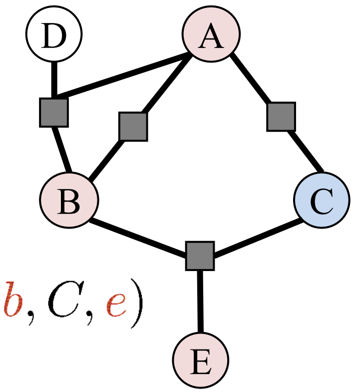
Gibbs sampling

[Geman & Geman 1984]

- Proceed in rounds
  - Sample each variable in turn given all the others' most recent values:

$$x'_0 \sim p(X_0 | x_1, x_2, x_3)$$
$$x'_1 \sim p(X_1 | x'_0, x_2, x_3)$$
$$x'_2 \sim p(X_2 | x'_0, x'_1, x_3)$$
$$\vdots$$

$$c \sim p(C \mid \ldots) \propto f(a, C)\, f(b, C, e)$$



  - Conditional distributions depend only on the Markov blanket
  - Easy to see that p(x) is a stationary distribution:

$$\sum_{x_1} p(x'_1 | x_2 \ldots x_n) p(x_1, \ldots x_n) = p(x'_1 | x_2 \ldots x_n) p(x_2, \ldots x_n) = p(x'_1, x_2 \ldots x_n)$$

Advantages:
No rejections
No free parameters (q)

Disadvantages:
"Local" moves
May mix slowly if vars strongly correlated
(can fail with determinism)

MCMC and Common Queries

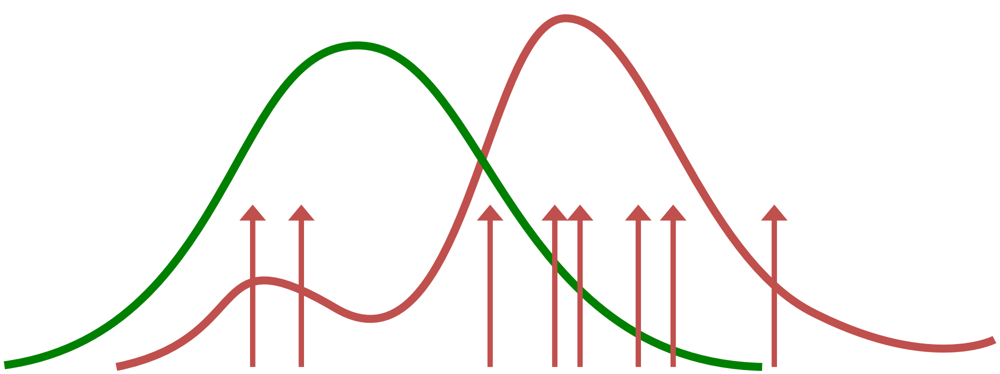
- MCMC generates samples (asymptotically) from p(x)

- Estimating expectations is straightforward

$$\mathbb{E}[u(x)] = \int p(x)u(x) \;\approx\; \hat{u} = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \{\, x^{(i)} \,\} \sim p(x)$$

- Estimating the partition function

$$\frac{1}{Z} = \int_x p_0(x)\,\frac{1}{Z} = \int_x p_0(x)\,\frac{p(x)}{f(x)}$$



MCMC and Common Queries

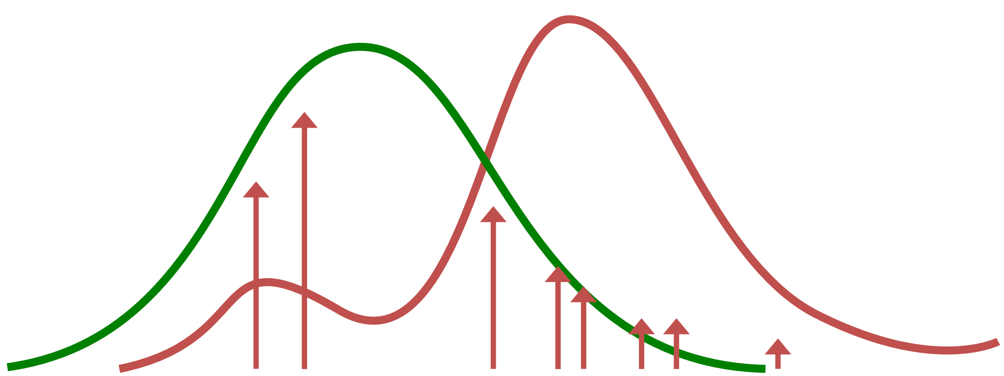
- MCMC generates samples (asymptotically) from p(x)

- Estimating expectations is straightforward

$$\mathbb{E}[u(x)] = \int p(x)u(x) \approx \hat{u} = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \{x^{(i)}\} \sim p(x)$$

- Estimating the partition function

$$\frac{1}{Z} = \int_x p_0(x)\,\frac{1}{Z} = \int_x p_0(x)\,\frac{p(x)}{f(x)} \approx \frac{1}{n}\sum_i \frac{p_0(x^{(i)})}{f(x^{(i)})}$$



"Reverse" importance sampling

$$\hat{Z}_{ris} = \left[\frac{1}{n}\sum_i \frac{p_0(x^{(i)})}{f(x^{(i)})}\right]^{-1}$$

Ex: Harmonic Mean Estimator
[Newton & Raftery 1994; Gelfand & Dey, 1994]

$f(x) = p(D|\theta)p(\theta) \qquad p_0(x) = p(\theta)$

MCMC

- Samples from p(x) asymptotically (in time)
  - Samples are not independent
- Rate of convergence ("mixing") depends on
  - Proposal distribution for MH
  - Variable dependence for Gibbs
- Good choices are critical to getting decent performance
- Difficult to measure mixing rate; lots of work on this

- Usually discard initial samples ("burn in")
  - Not necessary in theory, but helps in practice
- Average over rest; asymptotically unbiased estimator

$$\mathbb{E}[u(x)] = \int p(x)u(x) \approx \hat{u} = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim p(x)$$

Monte Carlo

Importance sampling

- i.i.d. samples
- Unbiased estimator
- Bounded weights provide finite-sample guarantees

- Samples from Q
- Good proposal: close to p but easy to sample from

- Reject samples with zero-weight

MCMC sampling

- Dependent samples
- Asymptotically unbiased
- Difficult to provide finite-sample guarantees

- Samples from ¼ P(X|e)
- Good proposal: move quickly among high-probability x

- May not converge with deterministic constraints

Outline

- Monte Carlo: Basics
- Importance Sampling
- Stratified & Abstraction Sampling
- Markov Chain Monte Carlo
- **Integrating Inference and Sampling**

Estimating with samples

- Suppose we want to estimate $p(X_i \mid E)$

- Method 1: histogram (count samples where $X_i = x_i$)

$$P(X_i = x_i | E) \approx \frac{1}{m} \sum_t \mathbb{1}[\tilde{x}_i^{(t)} = x_i] \qquad \tilde{x}^{(t)} \sim p(X|E)$$

- Method 2: average probabilities

$$P(X_i = x_i | E) \approx \frac{1}{m} \sum_t p(x_i \,|\, \tilde{x}_{\neg i}^{(t)}) \qquad \tilde{x}^{(t)} \sim p(X|E)$$

Converges faster! (uses all samples)

Rao-Blackwell Theorem: [e.g., Liu et al. 1995]

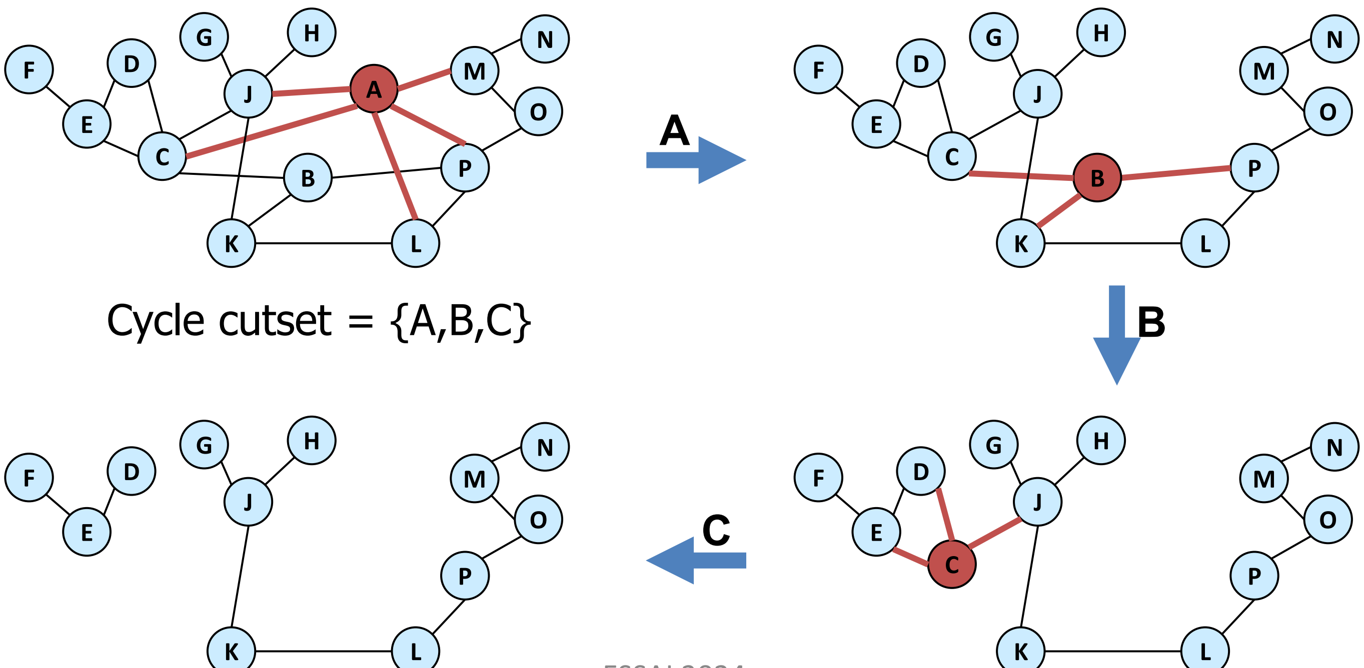
Let $X = (X_S, X_T)$, with joint distribution $p(X_S, X_T)$, to estimate $\mathbb{E}[u(X_S)]$

Then, $\mathrm{Var}\Big[\mathbb{E}[u(X_S)|X_T]\Big] \;\leq\; \mathrm{Var}\Big[u(X_S)\Big]$

Weak statement, but powerful in practice! Improvement depends on X_S, X_T

Cutsets

- Exact inference:
  - Computation is exponential in the graph's induced width
- "w-cutset": set C, such that $p(X_{:C} \mid X_C)$ has induced width w
  - "cycle cutset": resulting graph is a tree; w=1

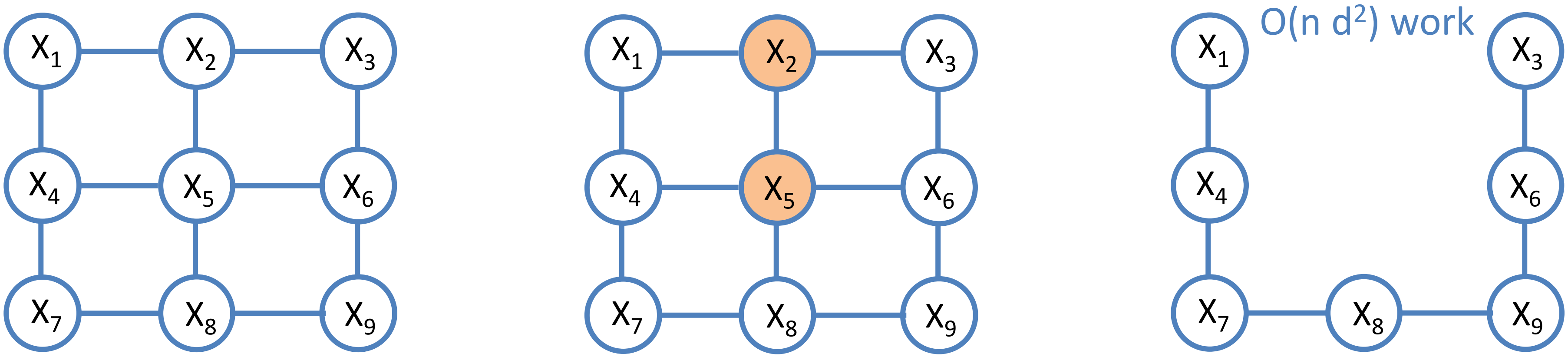


Cutset Importance Sampling

[Gogate & Dechter 2005, Bidyuk & Dechter 2006]

- Use cutsets to improve estimator variance
  - Draw a sample for a w-cutset X_C
  - Given X_C, inference is O(exp(w))

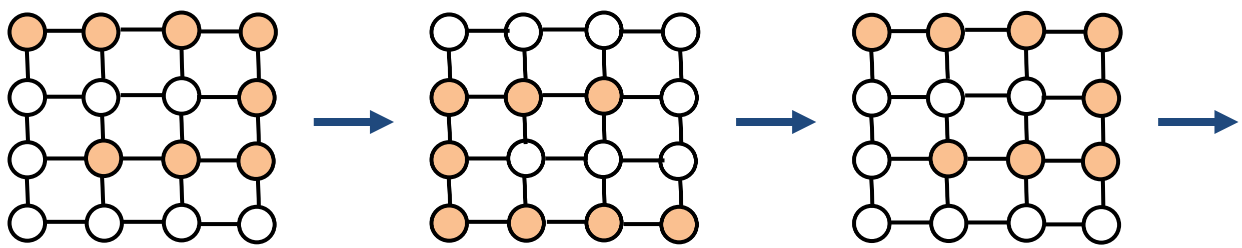
$F(X) = \prod f_{ij}(X_i, X_j)$ $\quad$ $\tilde{x}_2^{(i)}, \tilde{x}_5^{(i)} \sim q(X_2, X_5)$ $\quad$ $F(\tilde{x}_2^{(i)}, \tilde{x}_5^{(i)})$



(Use weighted sample average for X_C; weighted average of probabilities for $X_{:C}$)

Using Inference in Gibbs sampling

- "Blocked" Gibbs sampler
  - Sample several variables together

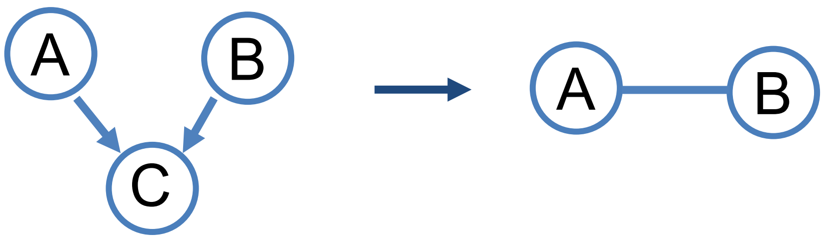


$x_A^{(1)} \sim p(X_A | x_B^{(0)})$ $x_B^{(1)} \sim p(X_B | x_A^{(1)})$ $x_A^{(2)} \sim p(X_A | x_B^{(1)})$ …

  - Cost of sampling is exponential in the block's induced width
  - Can significantly improve convergence (mixing rate)
  - Sample strongly correlated variables together

Using Inference in Gibbs sampling

- "Collapsed" Gibbs sampler
  - Analytically marginalize some variables before / during sampling

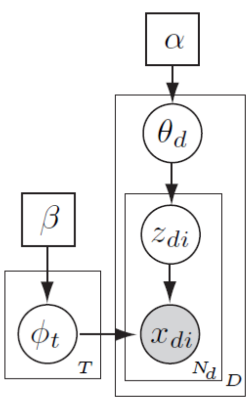


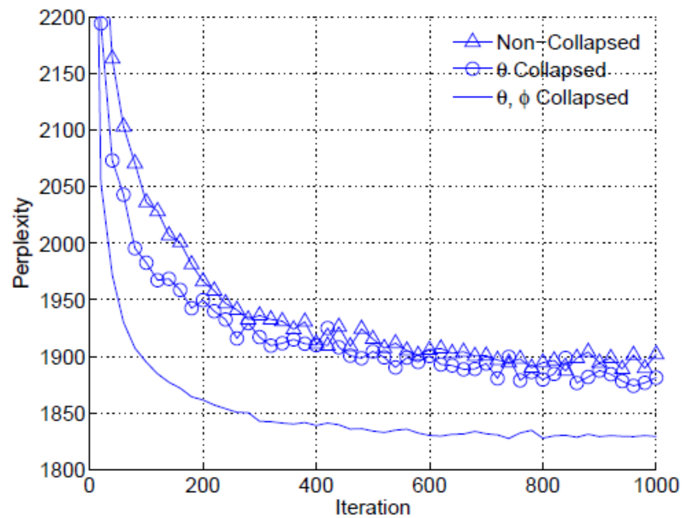
$$a^{(1)} \sim \sum_c p(A, c \mid b^{(0)})$$

$$b^{(1)} \sim \sum_c p(B, c \mid a^{(1)})$$

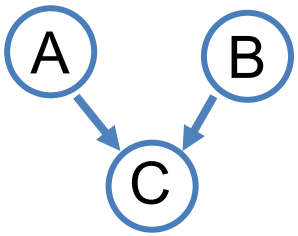
$$\vdots$$

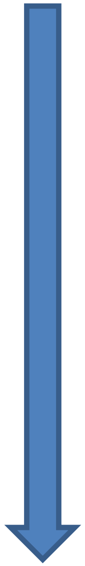
  - Ex: LDA "topic model" for text





Using Inference in Gibbs Sampling





- Standard Gibbs:
  $p(A \mid b, c) \rightarrow P(B \mid a, c) \rightarrow P(C \mid a, b)$ (1)
- Blocking:
  $p(A \mid b, c) \rightarrow P(B, C \mid a)$ (2)
- Collapsed:
  $p(A \mid b) \rightarrow P(B \mid a)$ (3)

Summary: Monte Carlo methods

- Stochastic estimates based on sampling
  - Asymptotically exact, but few guarantees in the short term
- Importance sampling
  - Fast, potentially unbiased
  - Performance depends on a good choice of proposal q
  - Bounded weights can give finite sample, probabilistic bounds
- Stratified & Abstraction Sampling
  - Ensemble of samples drawn together can reduce variance
- MCMC
  - Only asymptotically unbiased
  - Performance depends on a good choice of transition distribution
- Incorporating inference
  - Use exact inference within sampling to reduce variance