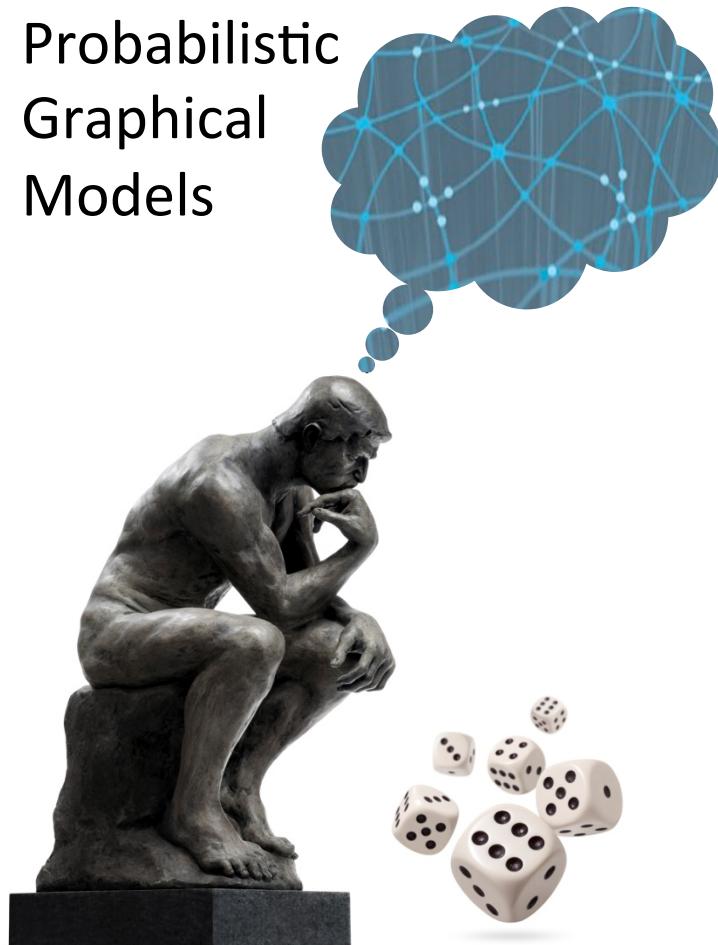


Probabilistic
Graphical
Models



Inference

Sampling Methods

Simple
Sampling

Sampling-Based Estimation

$\mathcal{D} = \{x[1], \dots, x[M]\}$ sampled IID from P
independent, identically distributed

If $P(X=1) = p = E_P[I_{X=1}]$

fraction of 1's



Estimator for p : $T_{\mathcal{D}} = \frac{1}{M} \sum_{m=1}^M I_{x[m]=1}$

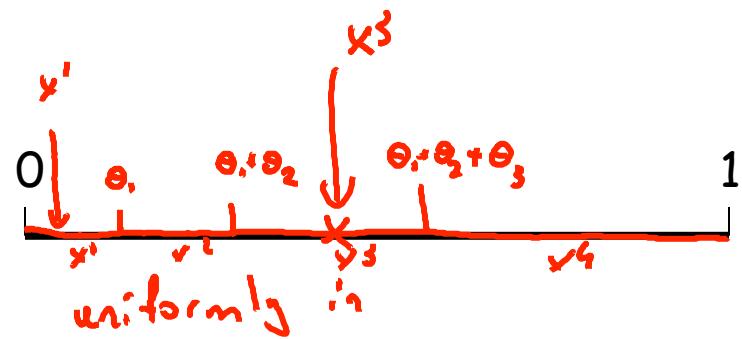
More generally, for any distribution P , function f :

$$E_P[f] \approx \frac{1}{M} \sum_{m=1}^M f(x[m])$$

indicator function f on samples
empirical expectation

Sampling from Discrete Distribution

$$\text{Val}(\underline{X}) = \{x^1, \dots, x^k\} \quad P(x^i) = \theta^i$$



Sampling-Based Estimation

Hoeffding Bound:

$$P_{\mathcal{D}}(T_{\mathcal{D}} \notin [p - \epsilon, p + \epsilon]) \leq 2e^{-2M\epsilon^2}$$

additive
samples
is ϵ -away from p
estimator
prob of sample set
a bad sample set

$$T_{\mathcal{D}} = \frac{1}{M} \sum_{m=1}^M X[m]$$

Chernoff Bound:

$$P_{\mathcal{D}}(T_{\mathcal{D}} \notin [p(1 - \epsilon), p(1 + \epsilon)]) \leq 2e^{-Mpe^2/3}$$

multiplication

Sampling-Based Estimation

Hoeffding Bound:

$$P_{\mathcal{D}}(T_{\mathcal{D}} \notin [p - \epsilon, p + \epsilon]) \leq 2e^{-2M\epsilon^2} < \delta$$

$$T_{\mathcal{D}} = \frac{1}{M} \sum_{m=1}^M X[m]$$

For additive bound ϵ on error with probability $> 1-\delta$:

$$M \geq \frac{\ln(2/\delta)}{2\epsilon^2}$$

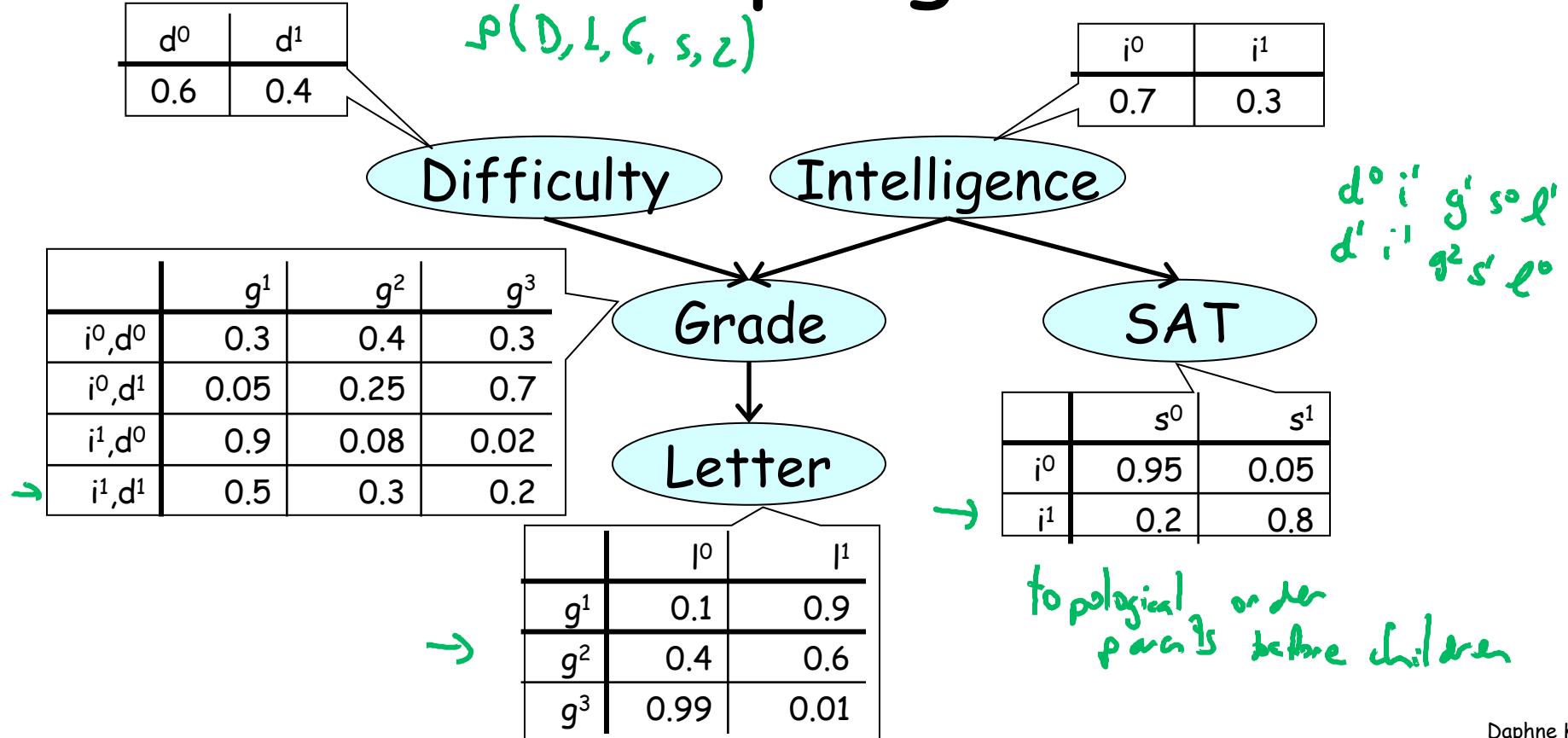
Chernoff Bound:

$$P_{\mathcal{D}}(T_{\mathcal{D}} \notin [p(1 - \epsilon), p(1 + \epsilon)]) \leq 2e^{-Mp\epsilon^2/3}$$

For multiplicative bound ϵ on error with probability $> 1-\delta$:

$$M \geq 3 \frac{\ln(2/\delta)}{p\epsilon^2}$$

Forward sampling from a BN



Forward Sampling for Querying

- Goal: Estimate $P(Y=y)$
 - Generate samples from BN
 - Compute fraction where $Y=y$

For additive bound ϵ on error with probability $> 1-\delta$: $M \geq \frac{\ln(2/\delta)}{2\epsilon^2}$

For multiplicative bound ϵ on error with probability $> 1-\delta$: $M \geq 3 \frac{\ln(2/\delta)}{P(y)\epsilon^2}$

Queries with Evidence

- Goal: Estimate $P(\underline{Y=y} \mid \underline{E=e})$
 - Rejection sampling algorithm
 - Generate samples from BN
 - Throw away all those where $\underline{E \neq e}$
 - Compute fraction where $\underline{Y=y}$
- remaining samples
are sampled
from P(Y=y | E=e)*

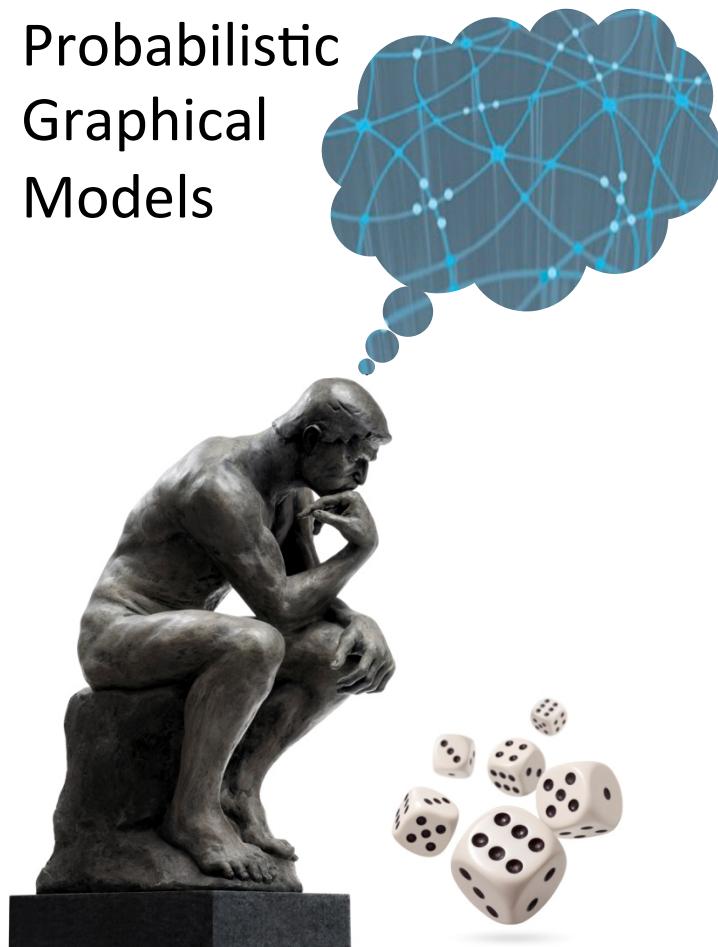
Expected fraction of samples kept $\sim P(e)$

samples needed grows exponentially
with # of observed variables

Summary

- Generating samples from a BN is easy
- (ε, δ) -bounds exist, but usefulness is limited:
 - Additive bounds: useless for low probability events
 - Multiplicative bounds: # samples grows as $1/P(y)$
- With evidence, # of required samples grows exponentially with # of observed variables
- Forward sampling generally infeasible for MNs

Probabilistic
Graphical
Models

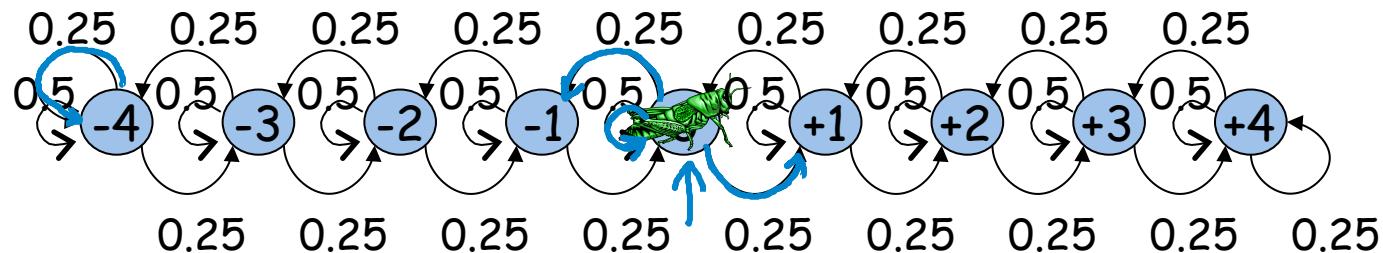


Inference

Sampling Methods

Markov Chain
Monte Carlo

Markov Chain

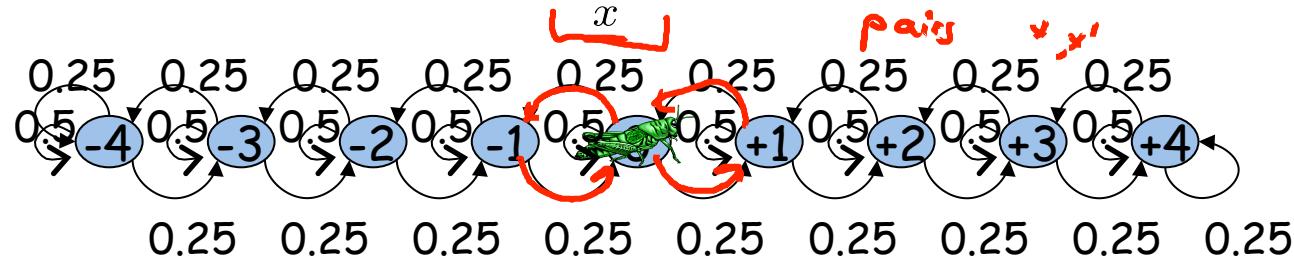


- A Markov chain defines a probabilistic transition model $T(x \rightarrow x')$ over states x :
 - for all x :

$$\sum_{x'} T(x \rightarrow x') = 1$$

Temporal Dynamics

$$P^{(t+1)}(X^{(t+1)} = x') = \sum P^{(t)}(X^{(t)} = x) T(x \rightarrow x')$$

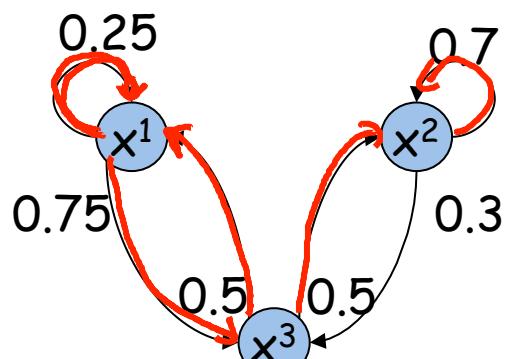


	-2	-1	0	+1	+2
$P^{(0)}$	0	0	1	0	0
$P^{(1)}$	0	.25	.5	.25	0
$P^{(2)}$	<u>$.25^2 =$</u> .0625	$2 \times (.5 \times .25) = .25$	<u>$.5^2 + 2 \times .25^2 = .375$</u>	$2 \times (.5 \times .25) = .25$	<u>$.25^2 =$</u> .0625

Stationary Distribution

$$P^{(t)}(x') \approx P^{(t+1)}(x') = \sum_x P^{(t)}(x) T(x \rightarrow x')$$

$$\pi(x') = \sum_x \pi(x) T(x \rightarrow x')$$



$$\underline{\pi(x^1)} = \underline{0.25\pi(x^1)} + \underline{0.5\pi(x^3)}$$

$$\underline{\pi(x^2)} = \underline{0.7\pi(x^2)} + \underline{0.5\pi(x^3)}$$

$$\pi(x^3) = 0.75\pi(x^1) + 0.3\pi(x^2)$$

$$\pi(x^1) = 0.2$$

$$\pi(x^2) = 0.5$$

$$\pi(x^3) = 0.3$$

$$\pi(x^1) + \pi(x^2) + \pi(x^3) = 1$$

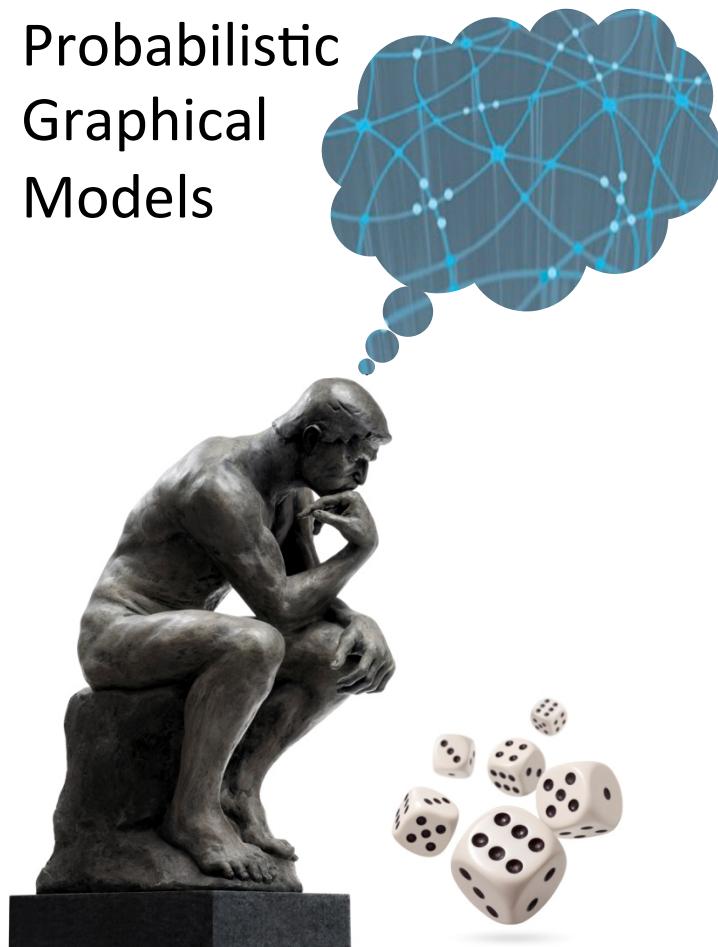
Regular Markov Chains

- A Markov chain is regular if there exists k such that, for every x, x' , the probability of getting from x to x' in exactly k steps is > 0
- Theorem: A regular Markov chain converges to a unique stationary distribution regardless of start state

Regular Markov Chains

- A Markov chain is regular if there exists k such that, for every x, x' , the probability of getting from x to x' in exactly k steps is > 0
 \leftarrow distance between furthest x, x'
- Sufficient conditions for regularity:
 - Every two states ^{x, y'} are connected with path of prob > 0
 - For every state, there is a self-transition

Probabilistic
Graphical
Models



Inference

Sampling Methods

Using a
Markov Chain

Using a Markov Chain

- Goal: compute $P(x \in S)$
 - but P is too hard to sample from directly
- Construct a Markov chain T whose unique stationary distribution is P
- Sample $x^{(0)}$ from some $P^{(0)}$
- For $t = 0, 1, 2, \dots$
 - Generate $x^{(t+1)}$ from $T(x^{(t)} \rightarrow x')$

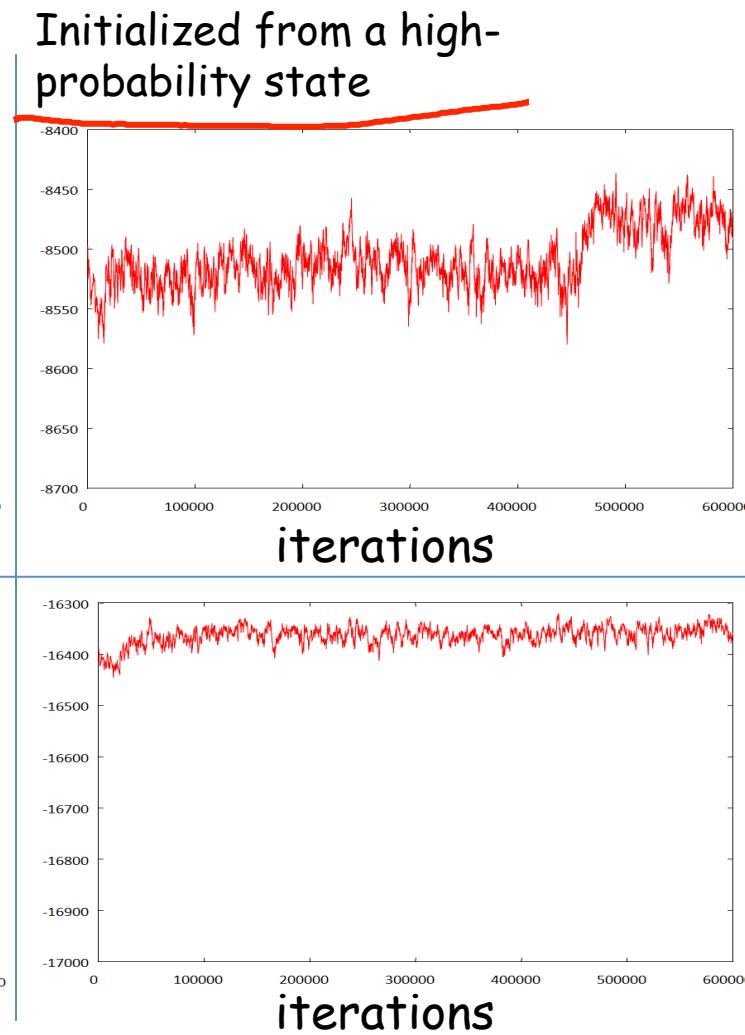
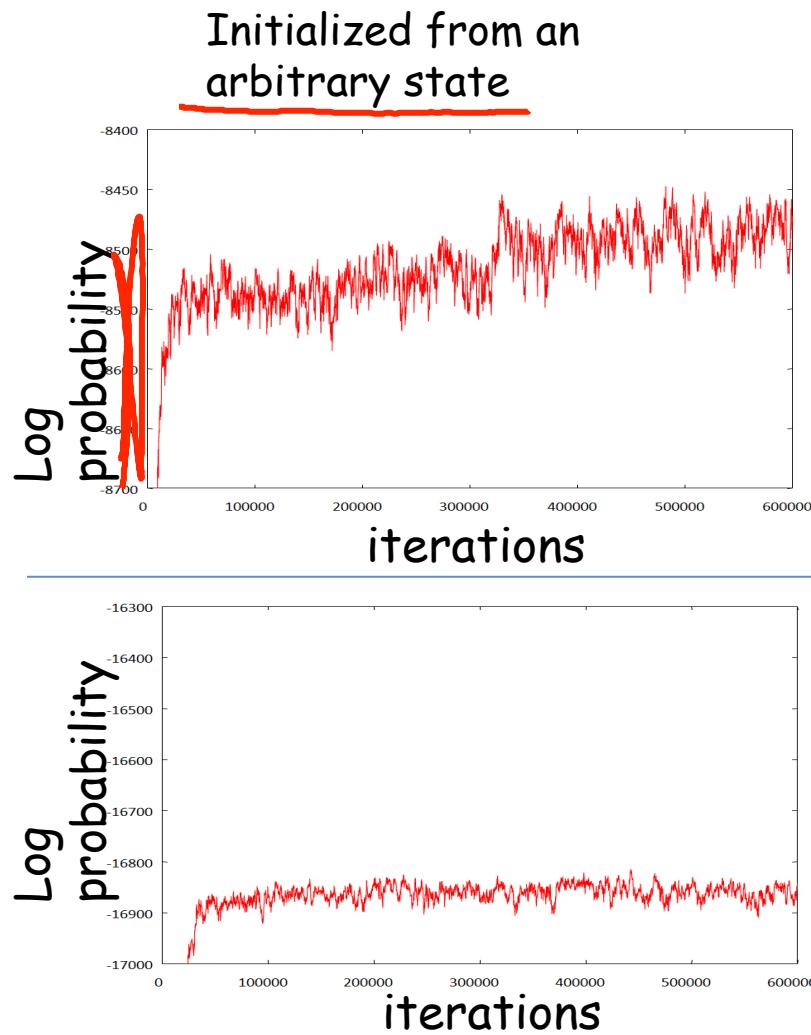
Using a Markov Chain

- We only want to use samples that are sampled from a distribution close to P
- At early iterations, $P^{(t)}$ is usually far from P
- Start collecting samples only after the chain has run long enough to "mix" $P^{(t)}$ close enough to P

Mixing

- How do you know if a chain has mixed or not?
 - In general, you can never “prove” a chain has mixed
 - But in many cases you can show that it has NOT
- How do you know a chain has not mixed?
 - Compare chain statistics in different windows within a single run of the chain
 - and across different runs initialized differently





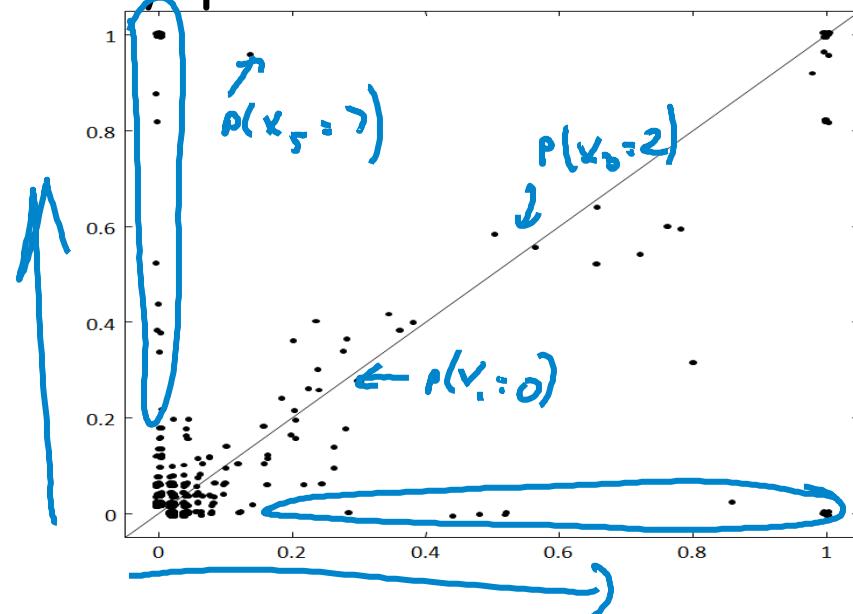
Mixing?

Maybe

NO

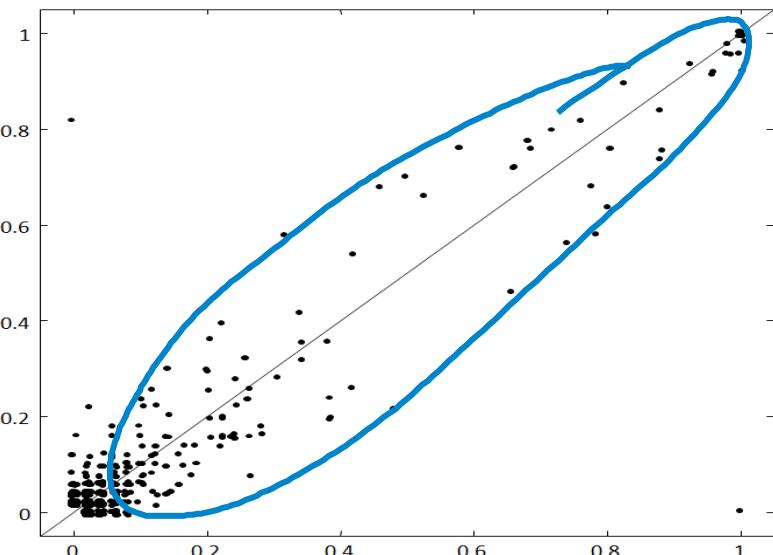


- Each dot is a statistic (e.g., $P(x \in S)$)
- x -position is its estimated value from chain 1
- y -position is its estimated value from chain 2



Mixing?

NO



Maybe

Using the Samples

- Once the chain mixes, all samples $x^{(t)}$ are from the stationary distribution π
 - So we can (and should) use all $x^{(t)}$ for $t > T_{\text{mix}}$
- However, nearby samples are correlated!
 - So we shouldn't overestimate the quality of our estimate by simply counting samples not IID
- The faster a chain mixes, the less correlated (more useful) the samples

MCMC Algorithm Summary I

- For $c=1, \dots, C$
 - Sample $x^{(c,0)}$ from $P^{(0)}$
- Repeat until mixing
 - For $c=1, \dots, C$
 - Generate $\underline{x}^{(c,t+1)}$ from $T(\underline{x}^{(c,t)} \rightarrow x')$
 - Compare window statistics in different chains to determine mixing
 - $t := t+1$

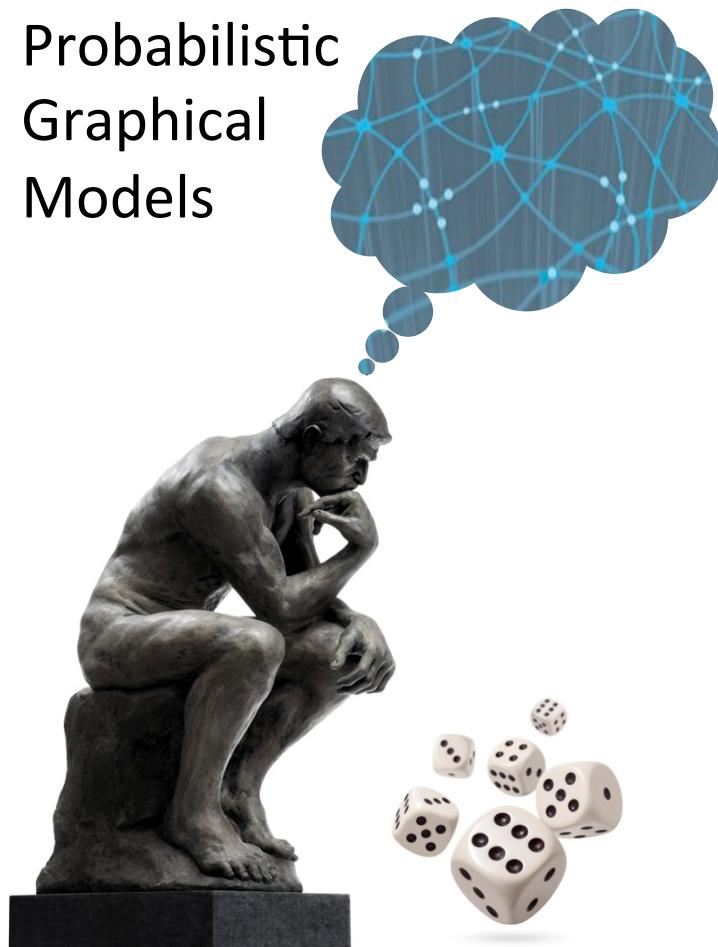
MCMC Algorithm Summary II

- Repeat until sufficient samples
 - $D := \emptyset$
 - For $c=1, \dots, C$
 - Generate $x^{(c, t+1)}$ from $T(x^{(c, t)} \rightarrow x')$
 - $D := D \cup \{x^{(c, t+1)}\}$
 - $t := t+1$
- Let $D = \{x[1], \dots, x[M]\}$
- Estimate $E_P[f] \approx \frac{1}{M} \sum_{m=1}^M f(x[m])$

Summary

- Pros:
 - Very general purpose
 - Often easy to implement
 - Good theoretical guarantees as $t \rightarrow \infty$
- Cons:
 - Lots of tunable parameters / design choices
 - Can be quite slow to converge
 - Difficult to tell whether it's working

Probabilistic
Graphical
Models



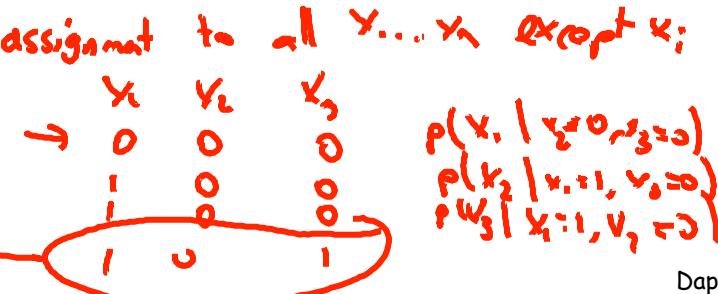
Inference

Sampling Methods

MCMC for
PGMs: The
Gibbs Chain

Gibbs Chain

- Target distribution $P_\Phi(X_1, \dots, X_n)$
- Markov chain state space: complete assignments x to X = $\{X_1, \dots, X_n\}$
- Transition model given starting state x :
 - For $i=1, \dots, n$
 - Sample $x_i \sim P_\Phi(X_i | x_{-i})$ assignment to all $x_1 \dots x_n$ except x_i
 - Set $x' = x$ ←



Daphne Koller

$$P(D | i^0, g^1, L^0, S^1)$$

	d^0	d^1
i^0	0.6	0.4
i^1		

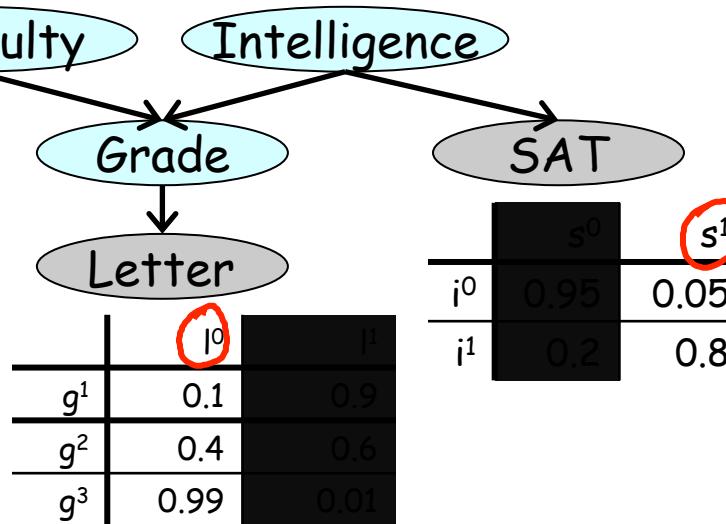
	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

$$P(G | d^1, i^1, L^0, S^1)$$

Example

$$P(i^1 | d^1, g^1, L^0, S^1)$$

	i^0	i^1
i^0	0.7	0.3
i^1		



$$P_i(d^1, i^1, G | s^1, S^1)$$

$$\begin{aligned} d^0 & i^0 & g^1 \\ d^1 & i^0 & g^1 \\ d^1 & i^1 & g^1 \\ d^1 & i^1 & g^3 \end{aligned}$$

$\hookrightarrow d^1, i^1, g^3$

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

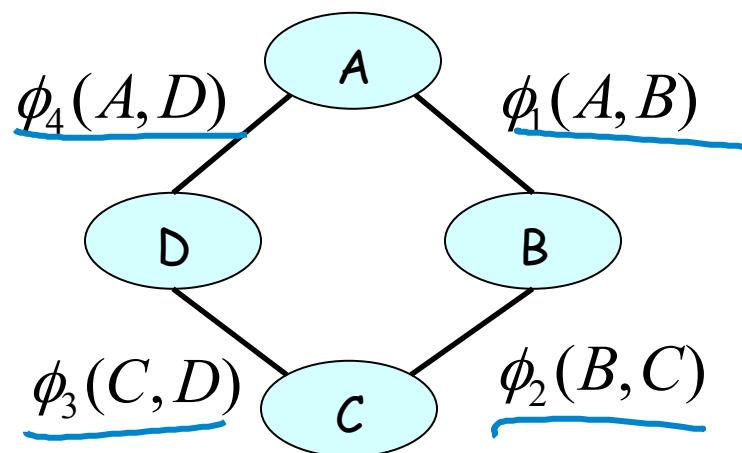
Computational Cost

- For $i=1, \dots, n$
 - Sample $x_i \sim P_\Phi(X_i | x_{-i})$

$$\underline{P_\Phi(X_i | x_{-i})} = \frac{P_\Phi(\underline{X_i}, \underline{x_{-i}})}{P_\Phi(\underline{x_{-i}})} = \frac{\cancel{\tilde{P}_\Phi(X_i, x_{-i})}}{\cancel{\tilde{P}_\Phi(x_{-i})}}$$

complete assignment
product at factors

Another Example



$$P_{\Phi}(A | b, c, d) = \frac{\tilde{P}_{\Phi}(a, b, c, d)}{\sum_{A'} \tilde{P}_{\Phi}(A', b, c, d)}$$

$$\frac{\phi_1(A, b)\phi_2(b, c)\phi_3(c, d)\phi_4(A, d)}{\sum_{A'} \phi_1(A', b)\phi_2(b, c)\phi_3(c, d)\phi_4(A', d)}$$

normalizing constant
 $\propto \phi_1(a, b) \phi_4(a, d)$

factors that involve A

Computational Cost Revisited

- For $i=1, \dots, n$
 - Sample $x_i \sim P_\Phi(X_i | \mathbf{x}_{-i})$

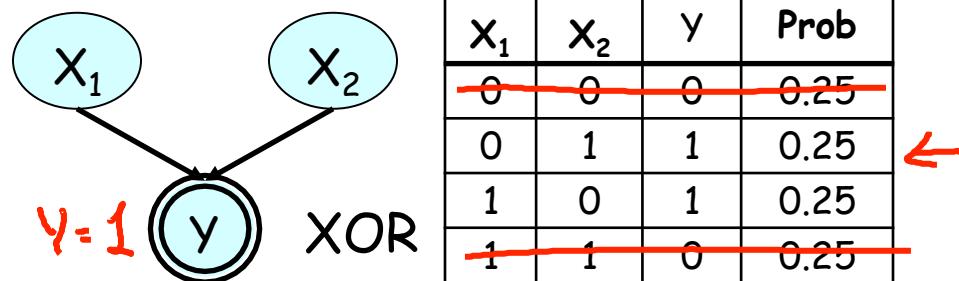
$$P_\Phi(\underline{X_i} | \underline{\mathbf{x}_{-i}}) = \frac{P_\Phi(X_i, \mathbf{x}_{-i})}{P_\Phi(\mathbf{x}_{-i})} = \frac{\tilde{P}_\Phi(X_i, \mathbf{x}_{-i})}{\tilde{P}_\Phi(\mathbf{x}_{-i})}$$

only x_i and
its neighbors

$$\propto \prod_{j: X_i \in \text{Scope}[C_j]} \phi_j(X_i, \mathbf{x}_{j,-i})$$

factored + hat
involve x_i

Gibbs Chain and Regularity

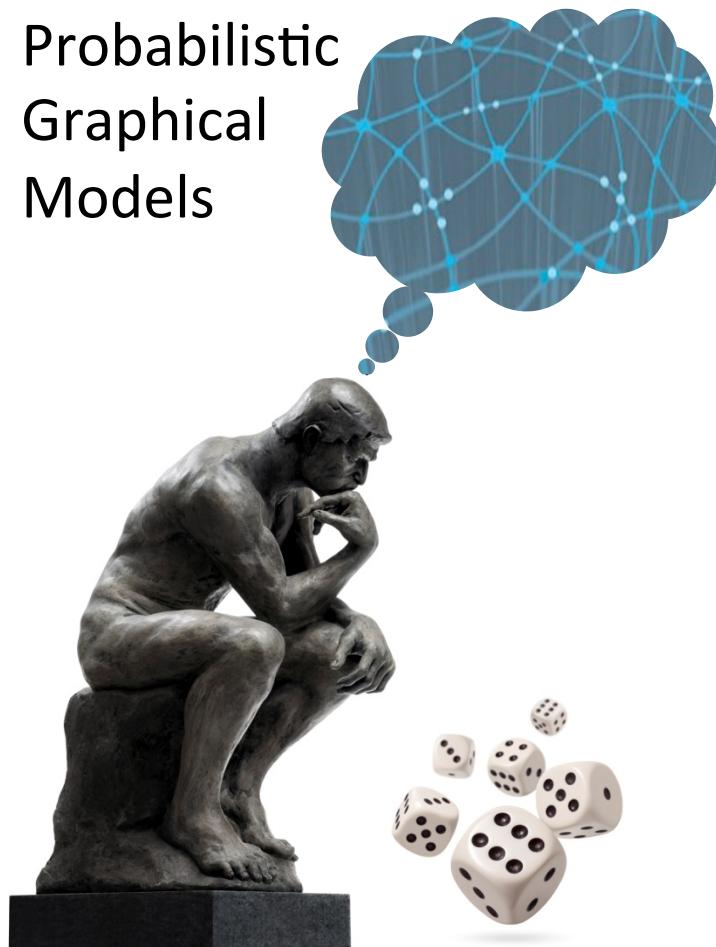


- If all factors are positive, Gibbs chain is regular
- However, mixing can still be very slow

Summary

- Converts the hard problem of inference to a sequence of "easy" sampling steps
- Pros:
 - Probably the simplest Markov chain for PGMs
 - Computationally efficient to sample
- Cons:
 - Often slow to mix, esp. when probabilities are peaked
 - Only applies if we can sample from product of factors

Probabilistic
Graphical
Models

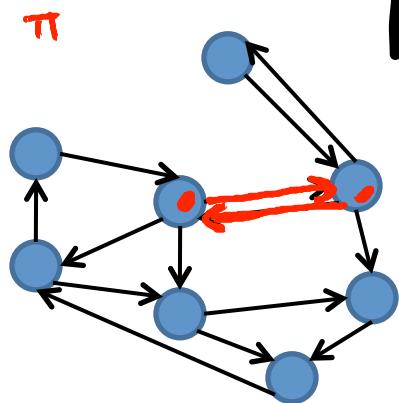


Inference

Sampling Methods

Metropolis-
Hastings
Algorithm

Reversible Chains



$$\boxed{\pi(x)T(x \rightarrow x')} = \boxed{\pi(x')T(x' \rightarrow x)}$$

detailed balance

Theorem: If detailed balance holds, and T is regular, then T has a unique stationary distribution $\underline{\pi}$

Proof:

$$\sum_x \pi(x)T(x \rightarrow x') = \sum_x \pi(x')T(x' \rightarrow x) = \underline{\pi(v)} \cdot \underbrace{\sum_x T(x \rightarrow x')}_1$$

$\rightarrow \sum_x \pi(x)T(x \rightarrow x') = \pi(x')$ definition of π

Metropolis Hastings Chain

Proposal distribution $Q(x \rightarrow x')$



Acceptance probability: $A(x \rightarrow x')$

- At each state x , sample x' from $Q(x \rightarrow x')$
- Accept proposal with probability $A(x \rightarrow x')$
 - If proposal accepted, move to x'
 - Otherwise stay at x

$$T(x \rightarrow x') = Q(x \rightarrow x') A(x \rightarrow x') \quad \text{if } x' \neq x$$

$$T(x \rightarrow x) = Q(x \rightarrow x) + \sum_{x' \neq x} Q(x \rightarrow x') (1 - A(x \rightarrow x'))$$

Acceptance Probability

$$\underline{\pi(x)T(x \rightarrow x')} = \underline{\pi(x')T(x' \rightarrow x)}$$

construct A s.t. \leftarrow holds for Q, π

$$x \neq x' \quad \pi(x)Q(x \rightarrow x')\underline{A(x \rightarrow x')} = \pi(x')Q(x' \rightarrow x)\underline{A(x' \rightarrow x)}$$

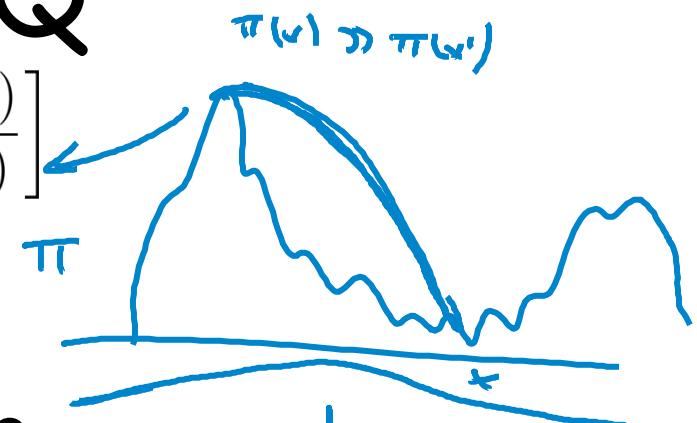
$$\begin{aligned} A(x \rightarrow x') &= p \\ A(x' \rightarrow x) &= 1 \end{aligned}$$

$$\frac{A(x \rightarrow x')}{A(x' \rightarrow x)} = \boxed{\frac{\pi(x')Q(x' \rightarrow x)}{\pi(x)Q(x \rightarrow x')} = p^{-1}}$$

$$\boxed{A(x \rightarrow x') = \min \left[1, \frac{\pi(x')Q(x' \rightarrow x)}{\pi(x)Q(x \rightarrow x')} \right]}$$

Choice of Q

$$A(x \rightarrow x') = \min \left[1, \frac{\pi(x')Q(x' \rightarrow x)}{\pi(x)Q(x \rightarrow x')} \right]$$



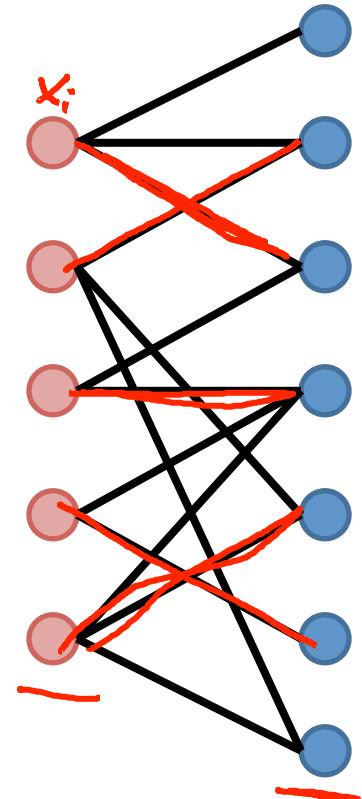
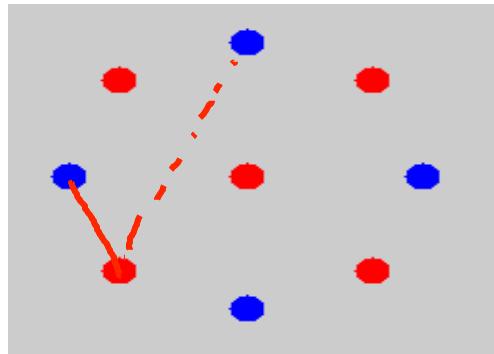
- Q must be reversible:
 - $Q(x \rightarrow x') > 0 \Rightarrow Q(x' \rightarrow x) > 0$
- Opposing forces
 - Q should try to spread out, to improve mixing
 - But then acceptance probability often low

MCMC for Matching

$X_i = j$ if i matched to j

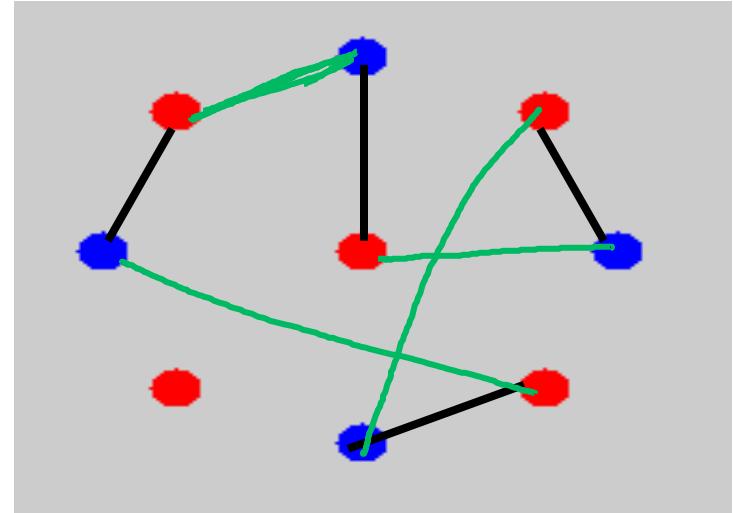
$$P(X_1 = v_1, \dots, X_4 = v_4) \propto$$

$$\begin{cases} \exp\left(-\sum_i \text{dist}(i, v_i)\right) & \text{if every } X_i \text{ has different value} \\ 0 & \text{otherwise} \end{cases}$$



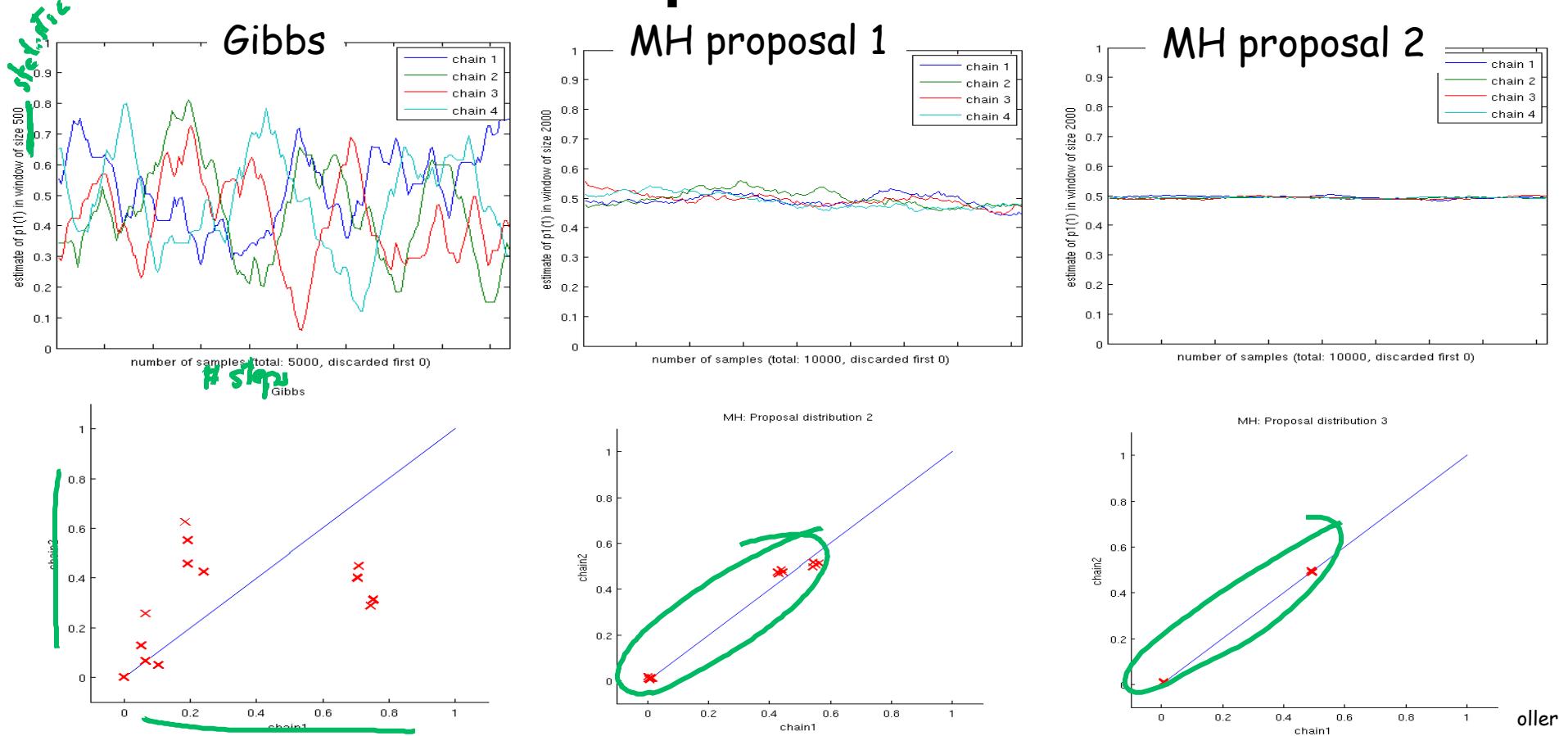
Daphne Koller

MH for Matching: Augmenting Path



- 1) randomly pick one variable X_i
 - 2) sample X_i , pretending that all values are available
 - 3) pick the variable whose assignment was taken (conflict), and return to step 2
- When step 2 creates no conflict, modify assignment to flip augmenting path

Example Results



Summary

- MH is a general framework for building Markov chains with a particular stationary distribution
 - Requires a proposal distribution
 - Acceptance computed via detailed balance
- Tremendous flexibility in designing proposal distributions that explore the space quickly
 - But proposal distribution makes a big difference
 - and finding a good one is not always easy