Introduction to Markov Chain Monte Carlo

- Monte Carlo: sample from a distribution
  - to estimate the distribution
  - to compute max, mean
- Markov Chain Monte Carlo: sampling using "local" information
  - Generic "problem solving technique"
  - decision/optimization/value problems
  - generic, but not necessarily very efficient

Based on - Neal Madras: Lectures on Monte Carlo Methods; AMS 2002

Lecture Outline

- Markov Chains notation & terminology
  - fundamental properties of Markov Chains
- Sampling from prob. distributions using MCMC
  - uniform
  - desired target distribution
- Problem solving using MCMC
  - optimization
- Relevance to Bayesian Networks

Markov Chains
Notation & Terminology

- Countable (finite) **state space** Ω (e.g. **N**)
- Sequence of **random variables** $\{X_t\}$ on Ω for $t = 0,1,2,...$

- Definition: $\{X_t\}$ is a **Markov Chain** if

$$P[X_{t+1} = y \mid X_t = x_t, \ldots, X_0 = x_0] = P[X_{t+1} = y \mid X_t = x_t]$$

- Notation: $P[X_{t+1} = i \mid X_t = j] = p_{ji}$
  - time-homogeneous

Markov Chains Examples

- Markov Chain
  - Drawing a number from {1,2,3} with replacement. X_t= last number seen at time t

- NOT a Markov Chain
  - Drawing a number from {1,2,3} WITHOUT replacement. X_t= last number seen at time t

Markov Chains
Notation & Terminology

- Let $P = (p_{ij})$ – **transition probability matrix**
  - dimension $|\Omega| x |\Omega|$
- Let $\pi_t(j) = P[X_t = j]$
  - π_0 **– initial probability distribution**
- Then

$$\pi_t(j) = \sum_i \pi_{t-1}(i) p_{ij} = (\pi_{t-1} P)(j) = (\pi_o P^t)(j)$$

- Example: graph vs. matrix representation

Markov Chains
Fundamental Properties

- Theorem:
  - Under some conditions (irreducibility and aperiodicity), the limit $lim_{t \to \infty} P^t_{ij}$ exists and is **independent** of *i*; call it $\pi(j)$. If Ω is finite, then

    $$\sum_j \pi(j) = 1 \text{ and } (\pi P)(j) = \pi(j)$$

    and such π is a **unique** solution to $xP=x$ (π is called a **stationary distribution**)

- Nice: no matter where we start, after some time, we will be in any state *j* with probability ~ $\pi(j)$

DEMO

Markov Chains Fundamental Properties

- Proposition:
  - Assume a Markov Chain with discrete state space Ω. Assume there exist positive distribution π on Ω ($\pi(i)>0$ and $\sum_i \pi(i) = 1$) and for every i,j:

  $$\pi(i)p_{ij} = \pi(j)p_{ji} \text{ (detailed balance property)}$$

  then π is the stationary distribution of P
- Corollary:
  - If transition matrix P is symmetric and Ω finite, then the stationary distribution is $\pi(i)=1/|\Omega|$

DEMO

Markov Chain Monte Carlo

- Random Walk on $\{0,1\}^m$
  - $\Omega=\{0,1\}^m$
  - generate chain: pick $J\in\{1,\ldots,m\}$ uniformly at random and set $X_t=(z_1,\ldots,1-z_J,\ldots,z_m)$ where $(z_1,\ldots,z_m)=X_{t-1}$
- Markov Chain Monte Carlo basic idea:
  - Given a prob. distribution π on a set Ω, the problem is to generate random elements of Ω with distribution π. MCMC does that by constructing a Markov Chain with stationary distribution π and simulating the chain.

MCMC: Uniform Sampler

- Problem: sample elements uniformly at random from set (large but finite) Ω
- Idea: construct an irreducible symmetric Markov Chain with states Ω and run it for sufficient time
  - by Theorem and Corollary, this will work
- Example: generate uniformly at random a feasible solution to the Knapsack Problem

MCMC: Uniform Sampler Example Knapsack Problem

- Definition
  - Given: m items and their weight w_i and value v_i, knapsack with weight limit b
  - Find: what is the most valuable subset of items that will fit into the knapsack?
- Representation:
  - $z=(z_1,\ldots,z_m)\in\{0,1\}^m$, z_i means whether we take item i
  - feasible solutions $\Omega = \{ z\in\{0,1\}^m ; \sum_i w_i z_i \leq b\}$
  - problem: maximize $\sum_i v_i z_i$ subject to $z\in\Omega$

MCMC Example: Knapsack Problem

- Uniform sampling using MCMC: given current $X_t=(z_1,\ldots,z_m)$, generate X_{t+1} by:

  (1) choose $J \in \{1,\ldots,m\}$ uniformly at random

  (2) flip z_J, i.e. let $y = (z_1,\ldots,1-z_J,\ldots,z_m)$

  (3) if y is feasible, then set $X_{t+1} = y$, else set $X_{t+1} = X_t$

- Comments:
  - P_{ij} is symmetric $\Rightarrow$ uniform sampling
  - how long should we run it?
  - can we use this to find a “good” solution?

MCMC Example: Knapsack Problem

- Can we use MCMC to find good solution?
  - Yes: keep generating feasible solutions uniformly at random and remember the best one seen so far.
    - this may take very long time, if number of good solutions is small
  - Better: generate "good" solutions with higher probability => sample from a distribution where "good" solutions have higher probabilities

$$\pi(z) = C^{-1} \exp\left(\sum_i v_i z_i \right)$$

MCMC: Target Distribution Sampler

- Let Ω be a countable (finite) state space
- Let Q be a symmetric transition prob. matrix
- Let π be **any** prob. distribution on Ω s.t. $\pi(i)>0$
  - the **target distribution**

- we can define a new Markov Chain $\{X_i\}$ such that its stationary distribution is π
  - this allows to sample from Ω according to π

MCMC: Metropolis Algorithm

- Given such Ω, π, Q creates a new MC $\{X_t\}$:

  (1) choose "proposal" y randomly using Q

  $$P[Y=j \mid X_t = i] = q_{ij}$$

  (2) let $\alpha = \min\{1, \pi(Y)/\pi(i)\}$ (acceptance probability)

  (3) accept y with probability α, i.e. $X_{t+1}=Y$ with prob. α, $X_{t+1}=X_t$ otherwise

- Resulting p_{ij}:

$$p_{ij}=q_{ij}\min\{1, \pi(j)/\pi(i)\} \text{ for } i \neq j$$

$$p_{ii} = 1 - \sum_{j\neq i} p_{ij}$$

MCMC: Metropolis Algorithm

- Proposition (Metropolis works):
  - The p_{ij}'s from Metropolis Algorithm satisfy detailed balance property w.r.t π i.e. $\pi(i)p_{ij} = \pi(j)p_{ji}$

  $\Rightarrow$ the new Markov Chain has a stationary distr. π

- Remarks:
  - we only need to know *ratios* of values of π
  - the MC might converge to π exponentially slowly

MCMC: Metropolis Algorithm Knapsack Problem

- Target distribution:

$$\pi(z) = C_\beta^{-1} \exp(\beta \sum_i v_i z_i)$$

- Algorithm:

  (1) choose $J \in \{1,\ldots,m\}$ uniformly at random

  (2) let $y = (z_1,\ldots,1-z_J,\ldots,z_m)$

  (3) if y is not feasible, then $X_{t+1} = X_t$

  (4) if y is feasible, set $X_{t+1} = y$ with prob. α, else $X_{t+1} = X_t$
  where $\alpha = \min\{1, \exp(\beta \sum_i v_i (y_i - z_i)\}$

MCMC: Optimization

- Metropolis Algorithm theoretically works, but:
  - needs large β to make "good" states more likely
  - its convergence time may be exponential in β

$\Rightarrow$ try changing β over time

- Simulated Annealing
  - for Knapsack Problem: $\alpha = \min\{1, \exp(\beta(t) \sum_i v_i (y_i - z_i)\}$
  - $\beta(t)$ increases slowly with time (e.g. $=\log(t)$, $=(1.001)^t$)

MCMC: Simulated Annealing

- General optimization problem: maximize function $G(z)$ on all feasible solutions Ω
  - let Q be again symmetric transition prob. matrix on Ω

- Simulated Annealing is Metropolis Algorithm with

$$p_{ij}=q_{ij}\min\{1, \exp(\beta(t) [G(j)-G(i)]) \} \text{ for } i \neq j$$

$$p_{ii}= 1 - \sum_{j\neq i} p_{ij}$$

- Effect of $\beta(t)$: exploration vs. exploitation trade-off

MCMC: Gibbs Sampling

- Consider a **factored** state space
  - $z \in \Omega$ is a vector $z=(z_1,\ldots,z_m)$
  - notation: $z_{-i} = (z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_m)$
- Assume that target π is s.t. $P[Z_i | z_{-i}]$ is known
- Algorithm:

  (1) pick a component $i \in \{1,\ldots,m\}$

  (2) sample value of z_i from $P[Z_i | z_{-i}]$, set $X_t=(z_1,\ldots,z_m)$
- A special case of generalized Metropolis Sampling (Metropolis-Hastings)

MCMC: Relevance to Bayesian Networks

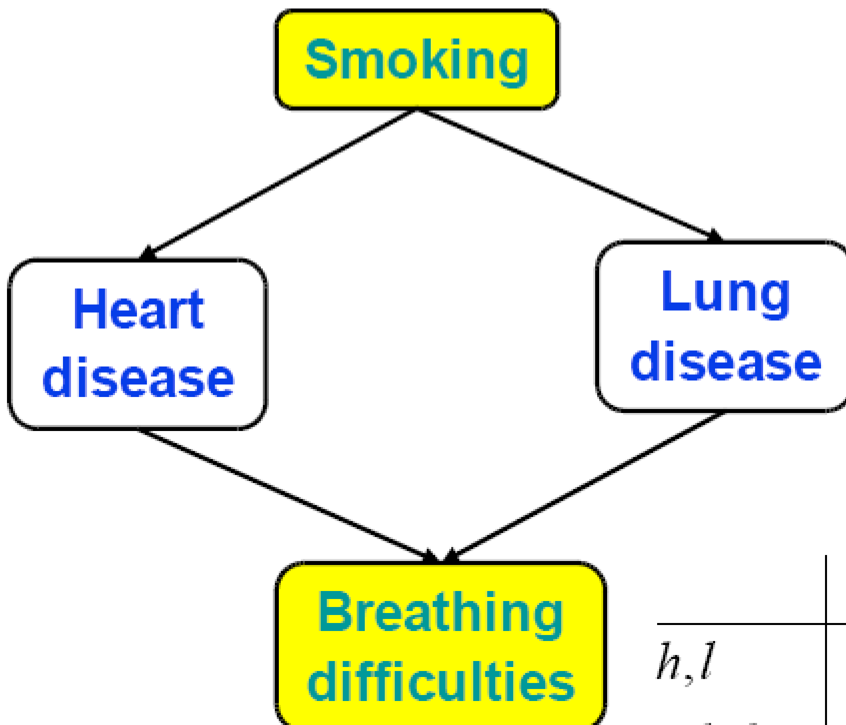
- In Bayesian Networks, we know

$$P[Z_i | z_{-i}] = P[Z_i \mid \text{MarkovBlanket}(Z_i)]$$

- BN Inference Problem: compute $P[Z_i = z_i | \boldsymbol{E} = \boldsymbol{e}]$
  - Possible solution:
    (1) sample from worlds according to $P[Z = z | \boldsymbol{E} = \boldsymbol{e}]$
    (2) compute fraction of those worlds where $Z_i = z_i$
  - Gibbs Sampler works:
    - let $\pi(z) = P[Z = z | \boldsymbol{E} = \boldsymbol{e}]$, then $P[Z_i | z_{-i}]$ satisfies detailed balance property w.r.t $\pi(z) \Rightarrow \pi(z)$ is stationary

MCMC: Inference in BN Example

$P(s)$	$P(\neg s)$
.4	.6



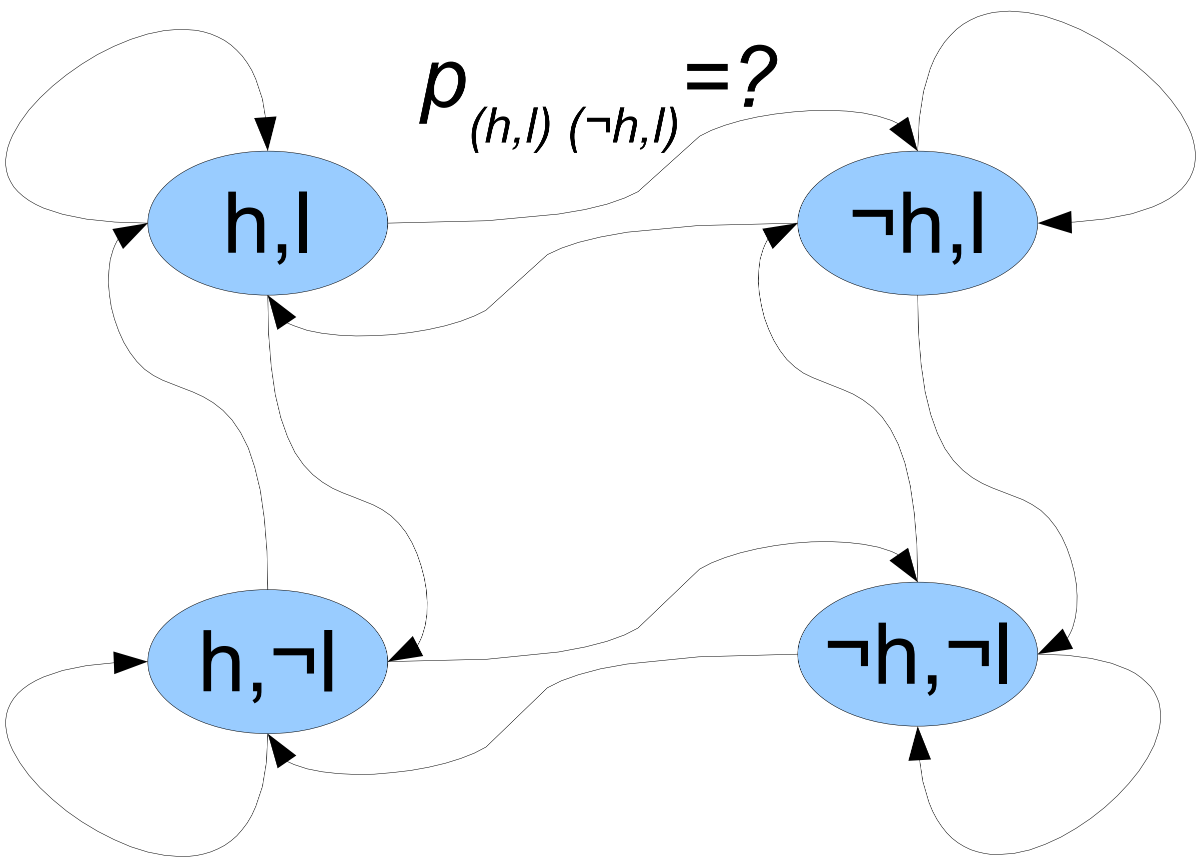
	$P(h)$	$P(\neg h)$
s	.4	.6
$\neg s$	.3	.7

	$P(l)$	$P(\neg l)$
s	.7	.3
$\neg s$	.2	.8

	$P(b)$	$P(\neg b)$
h,l	.9	.1
$\neg h,l$	.7	.3
$h,\neg l$	.5	.5
$\neg h,\neg l$	.2	.8

P[H | S=true, B=true]

MCMC: Inference in BN Example



Smoking and Breathing difficulties are fixed

MCMC: Inference in BN Example

- $P[z_i \mid \mathrm{MB}(Z_i)] \propto P[z_i|Par(Z_i)] \prod_{Y \in Chld(Z)} P[y|Par(Y)]$

- $p_{(h,l)\,(\neg h,l)} = P[h \text{ gets picked}].P[\neg h|MB(H)]$

  $= ½.P[\neg h|l,s,b]$

  $= ½.\alpha P[\neg h|s].P[b|\neg h,l]$