ISyE 6416: Computational Statistics
Spring 2017

Lecture 14: Markov Chain Monte Carlo

Prof. Yao Xie

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Introduction

- so far we have assumed we can draw random variables from any desired distribution
- for some problems, we cannot do this.
- in Markov chain Monte Carlo (MCMC) we do this by sampling $x_1, \ldots, x_n$ from a Markov chain constructed so that the distribution of x_i approaches the target distribution π
- MCMC originated in physics
- primary methods: Metropolis algorithm, Gibbs sampler

Markov chain

- a Markov chain is a mathematical model for stochastic systems whose states, discrete or continuous, are governed by transition probabilities.
- the current state in a Markov chain only depends on the most recent previous states
- a Markov chain is often denoted by (Ω, ν, P) for discrete state Markov chain
  - Ω: state space
  - $P = (p_{ij})$: probability to transition from state i to state j
  - ν: a row vector contains the stationary distribution
- **equilibrium distribution** π: a row vector such that

$$\pi = \pi P$$

- in MCMC, we design a Markov chain such that its stationary distribution converges to the desired distribution

Application of Markov Chain Monte Carlo

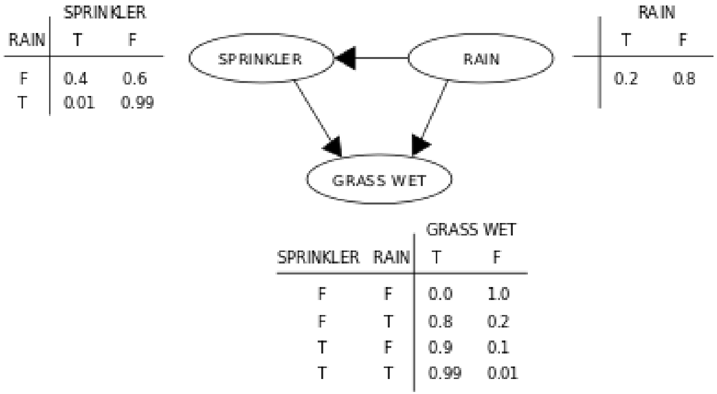
- learning: one need to compute posterior from likelihood $p(x|\theta)$ and prior distribution $p(\theta)$

  $$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{\int p(x|z)p(z)dz}$$

  $p(\theta|x)$ may not be computed in close-form
- Bayesian network: $p(x|\theta)$, $p(\theta|\rho)$, $p(\rho)$, and one has to figure out $p(\rho|x)$ for example

  $$p(\rho|x) \propto \int_\theta p(x|\theta)p(\theta|\rho)p(\rho)d\theta$$

Example: Bayesian network



	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99

RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

A simple Bayesian network with conditional probability tables.

Simple example

- let $\pi(x) = 10^x e^{-10}/x!$, $x = 0, 1, 2, \dots$
- set $x_0 = 0, 1, 2$ with probability $1/3$ for each
- set Markov chain transition probability

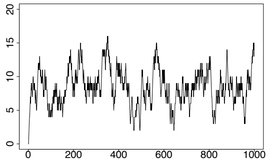
$$\mathbb{P}\{x_{i+1} = x_i - 1 | x_i\} = \begin{cases} x_i/20, & x_i \leq 9 \\ 1/2, & x_i > 9 \end{cases}$$

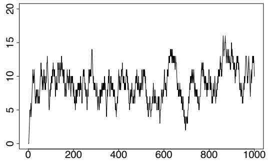
$$\mathbb{P}\{x_{i+1} = x_i | x_i\} = \begin{cases} (10 - x_i)/20, & x_i \leq 9 \\ (x_i - 9)/(2(x_i + 1)), & x_i > 9 \end{cases}$$

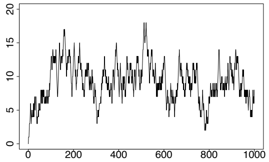
$$\mathbb{P}\{x_{i+1} = x_i + 1 | x_i\} = \begin{cases} 1/2, & x_i \leq 9 \\ 5/(x_i + 1), & x_i > 9 \end{cases}$$

(from Lecture notes by Hakon Tjelmeland)

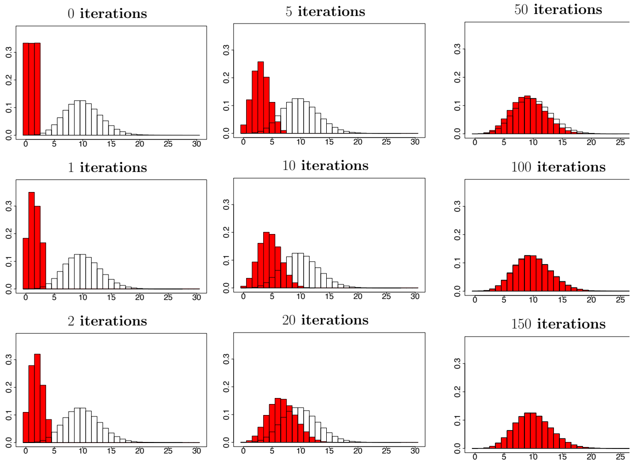
Trace plots of three runs







Convergence to target distribution



Metropolis-Hastings algorithm

- Given a prescribe equilibrium distribution π
- Come up with a proposed (arbitrary) transition probability from state i to state j: q_{ij} (it is symmetric: $q_{ij} = q_{ji}$)
- Metropolis-Hasting is a two-stage method
  - Stage 1: if the chain is currently in state i, then in the proposal stage a new state j is proposed according to q_{ij}
  - Stage 2: a random number is drawn uniformly from [0, 1] to determine whether the proposed step is actually taken.
    if the number is less than the Metropolis acceptance probability

    $$p_{ij} = \min\{1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\},$$

    then the proposed step is taken; otherwise, the proposed step is declined, and the chain remains in place.
- this will generate a Markov chain with equilibrium distribution π

Intuition

- at each iteration, the algorithm picks a candidate for the next sample value baed on the current sample value
- with some probability (based on p_{ij}), the candidate is either accepted, or rejected (in this case the current value is reused in the next iteration)
- the probability of acceptance is determined by comparing the likelihoods of the current and candidate sample values with respect to π

Proof

there are three components:

1. it suffices to verify the detailed balance $\pi_i q_{ij} p_{ij} = \pi_j q_{ji} p_{ji}$. Suppose p_{ij} is well-defined, $\pi_i > 0$ and $q_{ij} > 0$. WLOG, assume $\pi_j q_{ji}/(\pi_i q_{ij}) \leq 1$, for $j \neq i$. We have $p_{ij} = \pi_j q_{ji}/(\pi_i q_{ij})$, and $p_{ji} = 1$ (since the inverse ratio is greater than 1).

$$\pi_i q_{ij} p_{ij} = \pi_i q_{ij} \frac{\pi_j q_{ji}}{\pi_i q_{ij}} = \pi_j q_{ji} = \pi_j q_{ji} p_{ji}.$$

2. aperiodicity: acceptance-rejection allows the chain to remain in place
3. irreducibility of the chain: $\pi_i > 0$ for any i, and the chain defined by q_{ij} is irreducible

Example: Bayesian inference

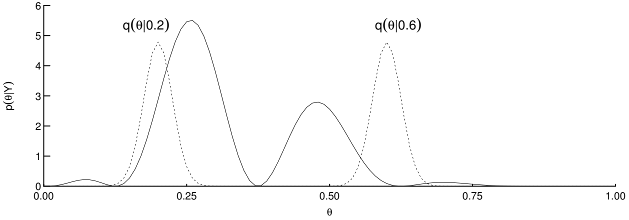
- Binomial distribution with non-standard prior
  - $Y = [Y_1, \ldots, Y_n]^T$ with $Y_1, \ldots, Y_n$ i.i.d. Bin$(1, \theta)$
  - $S_n = \sum_{i=1}^n Y_i$
  - $\pi(\theta) = 2\cos^2(4\pi\theta)$
- the posterior is

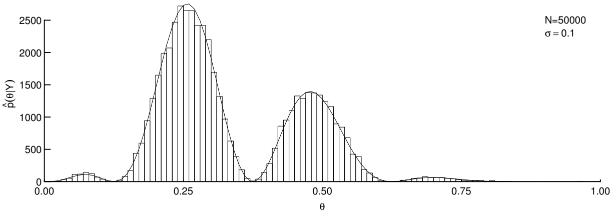
$$\pi(\theta|Y) \sim f(Y|\theta)\pi(\theta) = 2\theta^{S_n}(1-\theta)^{n-S_n}\cos^2(4\pi\theta)$$

- proposal distribution $q(\theta'|\theta) \sim \exp(-\frac{1}{2\sigma^2}(\theta-\theta')^2)$

acceptance probability

$$p(\theta, \theta') = \min\{1, \frac{\pi(\theta'|Y)q(\theta|\theta')}{\pi(\theta|Y)q(\theta'|\theta)}\}$$
$$= \min\{1, \frac{\theta'^{S_n}(1-\theta')^{n-S_n}\cos^2(4\pi\theta')}{\theta^{S_n}(1-\theta)^{n-S_n}\cos^2(4\pi\theta)}\}$$





Gibbs sampling

- Gibbs sampling is applicable when the joint distribution is not known explicitly or is difficult to sample from directly, but the conditional distribution of each variable is known and is easy to sample from.
- The Gibbs sampling algorithm generates an instance from the distribution of each variable in turn, conditional on the current values of the other variables.
- the sequence of samples constitutes a Markov chain, and the stationary distribution of that Markov chain is just the desired joint distribution.
- Gibbs sampling is particularly well-adapted to sampling the posterior distribution of a Bayesian network, since Bayesian networks are typically specified as a collection of conditional distributions.

Algorithm

- A special case of Metropolis-Hastings algorithm for Cartesian product state space: $S \times S \times \cdots \times S$ (product of m of S), with a specific form of proposed probability as follows
- each sample point $i = (i_1, \ldots, i_m)$ has m components. Gibbs sampler updates one component of i at a time.
- let i_c be a uniformly randomly chosen component.
- remaining component:

$$i_{-c} = (i_1, \ldots, i_{c-1}, i_{c+1}, \ldots, i_m).$$

- if j is a neighbor of i: identical by changing only one component i_c, then $j_{-c} = i_{-c}$

- define proposal probability for such a neighbor j

$$q_{ij} = \frac{1}{m} \frac{\pi_j}{\sum_{k:k_{-c}=i_{-c}} \pi_k}$$

  we have

$$\pi_i q_{ij} = \pi_j q_{ji}$$

- the ratio appeared in the Hastings-Metropolis algorithm is 1

Example: Ising Model

- mathematical model of ferromagnetism in statistical mechanics
- m elementary particles equally spaces around the boundary of the unit circle
- each particle c has two magnetic states: spin up $(i_c = 1)$ or spin down $(i_c = -1)$. The Gibbs distribution:

  $$\pi_i \propto \exp(\beta \sum_d i_d i_{d+1}).$$

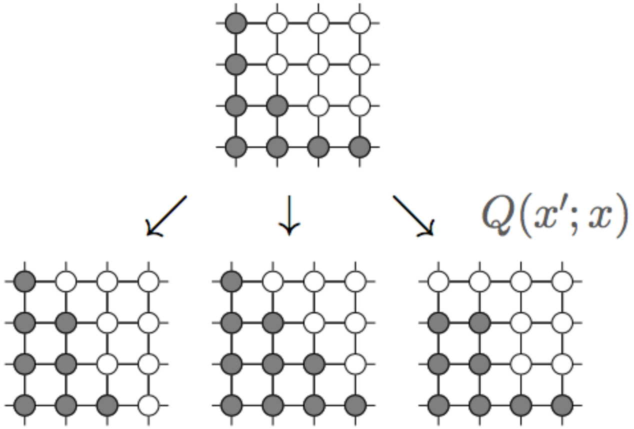
  Note $i_{m+1} = i_1$.
- for $\beta > 0$, state $(1, \ldots, 1)$ is favored and state $(1, -1, 1, -1, \ldots)$ is not.
- normalizing constant $Z = \sum_i \exp(\beta \sum_d i_d i_{d+1})$ is hard to compute, but it is not needed implementing Gibbs sampler

- from the formula: choose $j_c = -i_c$ and $j_c = i_c$ with probability

$$\begin{aligned}
&\frac{\pi_j}{\sum_{k:k_{-c}=i_{-c}} \pi_k} \\
&= \frac{\exp[\beta(j_c i_{c-1} + j_c i_{c+1})]}{\sum_{j_c} \exp[\beta(j_c i_{c-1} + j_c i_{c+1})]} \\
&= \begin{cases} \frac{\exp[\beta(-i_c i_{c-1} - i_c i_{c+1})]}{\exp[\beta(-i_c i_{c-1} - i_c i_{c+1})] + \exp[\beta(i_c i_{c-1}) + i_c i_{c+1}]}, & \text{if } j_c = -i_c \\ \frac{\exp[\beta(i_c i_{c-1} + i_c i_{c+1})]}{\exp[\beta(-i_c i_{c-1} - i_c i_{c+1})] + \exp[\beta(i_c i_{c-1}) + i_c i_{c+1}]}, & \text{if } j_c = i_c \end{cases} \\
&= \begin{cases} \frac{1}{1 + \exp[2\beta(i_c i_{c-1}) + i_c i_{c+1}]}, & \text{if } j_c = -i_c \\ \frac{\exp[\beta(i_c i_{c-1} + i_c i_{c+1})]}{\exp[\beta(-i_c i_{c-1} - i_c i_{c+1})] + \exp[\beta(i_c i_{c-1}) + i_c i_{c+1}]}, & \text{if } j_c = i_c \end{cases}
\end{aligned}$$

β: inverse temperature.

A demonstration of Ising model: http://physics.weber.edu/schroeder/software/demos/IsingModel.html


$Q(x';x)$

Simulated annealing

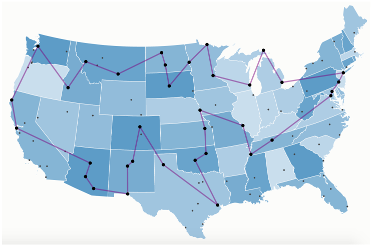
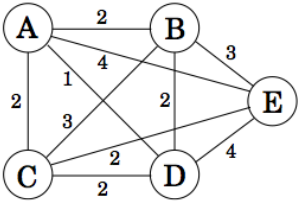
- Name: metal produced with a slow decrease in temperature (annealing) is stronger than metals produced with a fast decrease of the temperature.
- objective: finding the most probable state of a Markov chain: k, $\pi_k > \pi_i$ for all $i \neq k$
- Metropolis-Hastings acceptance probability

  $$p_{ij} = \min\{(\pi_j/\pi_i)^{1/\tau}, 1\}$$

  τ is call temperature
- if τ goes from ∞ to 0, it becomes harder to accept/make a move
- effect of τ: exploration vs. exploitation
- so far, no satisfactory theoretical justification

Simulation for solving optimization problems

- Traveling salesman problem: minimize distance traveled.

Example: solving traveling salesman problem

- a salesman must visit n towns
- given the traveling cost d_{ij} between every pair of towns i and j, in what order should he visit the towns to minimize the total cost?
- NP-complete problem: with deterministic solutions that are conjectured to increase in complexity at an exponential rate in n
- simulated annealing approach to solve this problem:
  - assign each permutation $\sigma = (\sigma_1, \ldots, \sigma_n)$ a cost

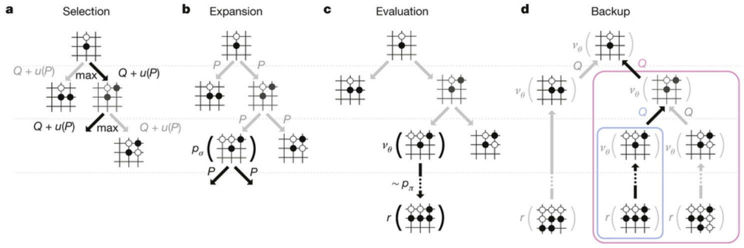
    $$c_\sigma = \sum_{i=1}^{n} d_{\sigma_i, \sigma_{i+1}}$$

  - define

    $$\pi_\sigma \propto e^{-c_\sigma}$$

    turns the problem into to finding the most probably permutation σ.

A computer chess program could be seen as trying to find the set of, say, 10 moves that produces the best evaluation function at the end.



Monte Carlo tree search in AlphaGo.