CS 5135/6035 Learning Probabilistic Models

Lecture 21: Markov Chain Monte Carlo Methods I

Gowtham Atluri

November 15, 2018

Reading Material

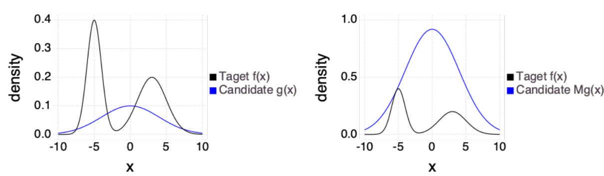
- Chapter 6. Metropolis-Hastings Algorithms
  Christian Robert and George Casella. Introducing Monte Carlo Methods with R
- Ilker Yildirim, Bayesian Inference: Metropolis-Hastings Sampling
  http://www.mit.edu/~ilkery/papers/MetropolisHastingsSampling.pdf
- Andrieu et al. An introduction to MCMC for machine learning, Machine learning, 2003.

Topics

- Limitations of sampling and integration approaches
- Intuitive idea of MCMC
- Markov Chain
  - Introduction
  - Finite state space
  - Infinite state space
  - Stationary distribution
  - Ergodicity
  - Ergodic Theorem
- Metropolis Hastings Algorithm
  - Algorithm
  - Acceptance probability
  - Example

Accept-Reject Methods

- Accept-reject methods used to sample from arbitrary distributions
- Need to choose $g(x)$ such that $Mg(x)$ is a tighter envelope
  - Non-trivial for high-dimensional problems

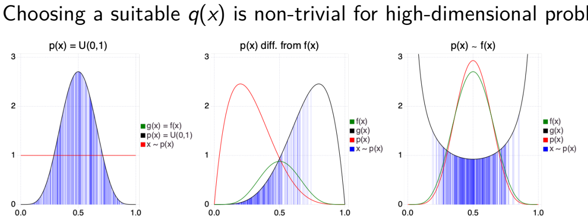


Challenges with Monte Carlo Integration

$$I(f) = \int f(\mathbf{x})d\mathbf{x} = \int g(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \mathbb{E}_{p(x)}[g(\mathbf{x})] \approx \frac{1}{n}\sum_{i=1}^{n} g(\mathbf{x}_i) = S_n$$

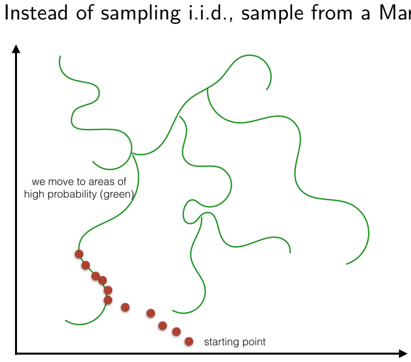
Error in Monte Carlo Integration $|\epsilon_n| \leq 1.96\sqrt{\frac{Var[g(x)]}{n}}$

- In Importance sampling, we use the factorization $\frac{f(x)}{q(x)}q(x)$ for $g(x)p(x)$
  - Error is small when we choose $q(x)$ with the same shape as $f(x)$
- Choosing a suitable $q(x)$ is non-trivial for high-dimensional problems



Idea behind Markov Chain Monte Carlo Methods

- Instead of sampling i.i.d., sample from a Markov Chain



- *Markov Chain*- where we go next depends on our current state
- *Monte Carlo* - Simulating data

Advantages/Disadvantes of MCMC

Advantages:

- applicable even when we can't directly draw samples
- works for complicated distributions in high-dimensional spaces, even when we don't know where the regions of high probability are
- relatively easy to implement
- fairly reliable

Disadvantages:

- slower than simple Monte Carlo or importance sampling (i.e., requires more samples for the same level of accuracy)
- can be very difficult to assess accuracy and evaluate convergence, even empirically

Markov Chain

- A *Markov Chain* is a **sequence of random variables** $\{x_i\} = x_0, x_1, \ldots, x_n$ such that, given the present state, future and past states are independent.

$$p(x_{n+1}|x_1, x_2, \ldots x_n) = p(x_{n+1}|x_n)$$

- In other words, conditional distribution of x_{n+1} (in future), depends only on present state x_n

Markov Chain

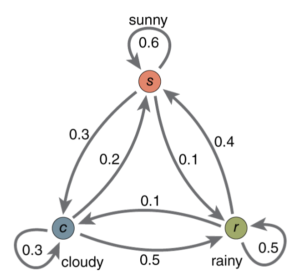
- A *Markov Chain* is a *sequence* of random variables $x_1, x_2, \ldots, x_n$ such that, given the present state, future and past states are independent

$$p(x_{n+1}|x_1, x_2, \ldots x_n) = p(x_{n+1}|x_n)$$

- In other words, conditional distribution of x_{n+1} (in future), depends only on present state x_n
- *State space* of the Markov Chain: the set from which x_i take values
  - $[s_1, s_2, \ldots, s_k]$
- *Example:* Weather over 6 days is observed as $\{$*sunny*, *sunny*, *cloudy*, *rainy*, *sunny*, *sunny*$\}$
  - State-space is [*sunny*, *cloudy*, *rainy*]

Defining a Markov Chain $\{x_i\}$

- **State space** of the Markov Chain: the set from which x_i take values
- **Initial distribution (π_0):** the distribution of x_0
- **Transition probability distribution** or **Markov kernel** $K(x_n, x_{n+1})$: conditional distribution $p(x_{n+1}|x_n)$
  - *Time-homogeneous chain* when $p(x_{n+1}|x_n)$ does not depend on n



Defining a Markov Chain $\{x_i\}$

- **State space** of the Markov Chain: the set from which x_i take values
- **Initial distribution (π_0):** the distribution of x_0
- **Transition probability distribution** or **Markov kernel** $K(x_n, x_{n+1})$: conditional distribution $p(x_{n+1}|x_n)$
  - *Time-homogeneous chain* when $p(x_{n+1}|x_n)$ does not depend on n

Example: Weather over 6 days is observed as $\{$*sunny*, *sunny*, *cloudy*, *rainy*, *sunny*, *sunny*$\}$

- State-space is [*sunny*, *cloudy*, *rainy*] (Discrete-state Markov Chain)
- Let our initial distribution be $\pi_0 = [0.8, 0.05, 0.15]$
- Let our transition probabilities be:

$$K = \left(\begin{array}{ccccc} & & & T & \\ & & Sunny & Cloudy & Rainy \\ & Sunny & 0.6 & 0.3 & 0.1 \\ T-1 & Cloudy & 0.2 & 0.3 & 0.5 \\ & Rainy & 0.4 & 0.1 & 0.5 \end{array} \right)$$

Markov Chain: probability

Example: Probability of daily weather from a Markov Chain

- State-space is [*sunny*, *cloudy*, *rainy*], initial distr. $\pi_0 = [0.8, 0.05, 0.15]$
- Transition probability distr.:

$$K = \left(\begin{array}{cccc} & Sunny & Cloudy & Rainy \\ Sunny & 0.6 & 0.3 & 0.1 \\ Cloudy & 0.2 & 0.3 & 0.5 \\ Rainy & 0.4 & 0.1 & 0.5 \end{array} \right)$$

- The probability for the second day π_1 is $\pi_1 = \pi_0 * K$, i.e.,

$$[0.8, 0.05, 0.15] * \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.3 & 0.5 \\ 0.4 & 0.1 & 0.5 \end{pmatrix} = [0.55, 0.27, 0.18]$$

- The probability for the third day is $\pi_2 = \pi_1 * K = \pi_0 * K^2$, i.e.,

$$[0.8, 0.05, 0.15] * K^2 = [0.46, 0.26, 0.28]$$

- Probability of states at n: $\pi_n = \pi_{n-1} * K = \pi_0 * K^n$

Stationary distribution: Equilibruim

$$\text{Typically,}\quad \pi_1 = \pi_0 K$$
$$\pi_2 = \pi_1 K$$
$$\pi_3 = \pi_2 K$$
$$\vdots$$
$$\pi_n = \pi_{n-1} K$$

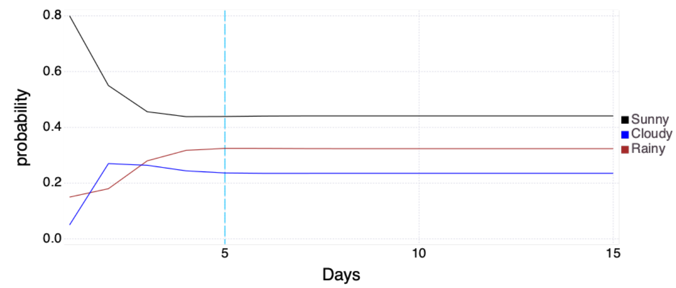
- A **stationary distribution** of a Markov Chain is a probability distribution that remains unchaged in the Markov Chain as time progresses $\pi = \pi K$
  - π can be determined by solving the set of eqns $\pi = \pi K$
  - When $\pi_n = \pi$, we say the chain reached equilibrium
- Once in equilibrium, $x_{n+1} \sim \pi$, $x_{n+2} \sim \pi, \ldots$
  - This property is leveraged by MCMC approaches

Stationary distribution

Example: Weather

- State-space is [*sunny*, *cloudy*, *rainy*], initial distr. $\pi_0 = [0.8, 0.05, 0.15]$

$$\text{Transition probability distr.:} K = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.3 & 0.5 \\ 0.4 & 0.1 & 0.5 \end{pmatrix}$$



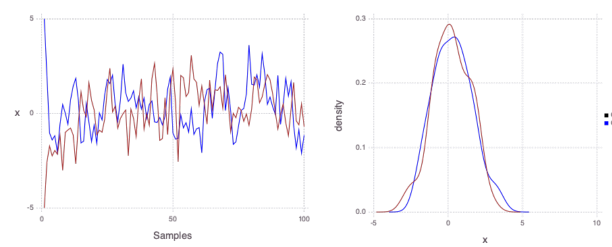
Markov Chain: Continuous States

- **State space**: $-\infty \leq x \leq \infty$ (uncountable)
- **Initial distribution (π_0):** $\pi_0(x) = \mathcal{N}(0, 3)$
- **Transition probability distribution**:$p(x_{n+1}|x_n) = K(x_n, x_{n+1}) = \mathcal{N}(x_n/2, 1)$

Markov Chain: Continuous States

- **State space**: $-\infty \leq x \leq \infty$ (uncountable)
- **Initial distribution (π_0):** $\pi_0(x) = \mathcal{N}(0, 3)$
- **Transition probability distribution**:$p(x_{n+1}|x_n) = K(x_n, x_{n+1}) = \mathcal{N}(x_n/2, 1)$
- Generating data from this Markov Chain

$$x_{n+1} \sim \mathcal{N}(x_n/2, 1), \text{ alternatively } \quad x_{n+1} = x_n/2 + \epsilon_n, \quad \epsilon_n \sim \mathcal{N}(0, 1)$$



Properties of Markov Chain

- **Irreducibility**: Every state is reachable from any other state in finite time

$$\forall s_i, s_j \in S, \exists m < \infty : p(x_{n+m} = s_j | x_n = s_i) > 0$$

  - If there is path from each state to every other, then it's irreducible.

$$K_1 = \left(\begin{array}{cccc} & A & B & C \\ A & 0.8 & 0 & 0.2 \\ B & 0 & 0.5 & 0.5 \\ C & 0.4 & 0.6 & 0 \end{array}\right) \text{(irreducible)} \quad K_2 = \left(\begin{array}{cccc} & A & B & C \\ A & 0.8 & 0.15 & 0.05 \\ B & 0.4 & 0.5 & 0.1 \\ C & 0 & 0 & 1 \end{array}\right) \text{(reducible)}$$

- When irreducibile, *Markov Kernel* K allows for free moves all over the state space
  - This property is leveraged by MCMC approaches

Reversibility

- If the joint probability of $x_0, x_1, \ldots, x_{n-1}, x_n$ is the same as that of $x_n, x_{n-1}, \ldots, x_1, x_0$, we say the markov chain is **reversible**.

$$p(x_0, x_1, \ldots, x_{n-1}, x_n) = p(x_n, x_{n-1}, \ldots, x_1, x_0)$$

- Recorded simulation of a reversible chain looks the same if it is run backwards
- A Markov chain with stationary probability π is reversible if and only if

$$\pi_i K_{ij} = \pi_j K_{ji}, \forall i, j$$

- These are called *detailed balance equations*
- If a Markov Kernel K satifies *detailed balance equations* w.r.t a distrbution π
  - Then π is a unique stationary distribution
  - This property will be leveraged in MCMC.

Recurrent & a-periodic states

- **Recurrent state**: a state that the chain began with is guaranteed to be revisited. Otherwise state s_i is *transient*
  - If all states are recurrent, stationary distribution π exists.

$$t_i = \inf(t \geq 1 : x_t = s_i) \qquad p(t_i < \infty | x_0 = s_i) = 1$$

$$K = \begin{pmatrix} & A & B & C \\ A & 0.85 & 0.15 & 0 \\ B & 0.1 & 0.9 & 0 \\ C & 0.8 & 0.2 & 0 \end{pmatrix} \text{ A is recurrent, C is transient}$$

- **Periodicity**: A state is periodic if the chain can return to it only at multiples of some integer larger than 1

$$d(s_i) = gcd\{n \in \mathbb{N}_+ : K_{ii}^n > 0\}$$

  - if $d(s_i) = 1$ the state is *a-periodic*

Ergodicity, Ergodic Theorem

- **Ergodicity**: A state is *ergodic* if it is recurrent and a-periodic
  - A Markov Chain is ergodic if all states are ergodic
- **Convergence**: Irrespective of the starting state, the probability of state s_j follows stationary distribution.
  - For an ergodic Markov chain $\lim_{t \to \infty} p(x_t = s_j | x_0 = s_i) = \pi$
- These properties (Ergodicity and Convergence) have major consequences for simulation
  - Samples from Markov Chain $x \sim \pi$
  - These will be leveraged for MCMC

Ergodic Theorem and CLT

- **Ergodic Theorem**: For a finite irreducible chain with stationary distribution π

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} h(x_t) = \mathbb{E}_\pi(h(x))$$

- This expectation is the same as the integral $\int h(x)\pi(x)dx$
  - This will be leveraged by MCMC methods
- **Central Limit Theorem**: For a finite irreducible chain with stationary distribution π

$$\frac{1}{\sqrt{n}} \Big(\sum_{t=1}^{n} (h(x_t) - \mathbb{E}_\pi(h)) \Big) \implies {}_{distr.} \mathcal{N}(0, \sigma_h^2)$$

  where $\sigma_h = \gamma_0 + 2\sum_{k=1}^{\infty} \gamma_k < \infty, \gamma_k = Cov_\pi(h(x_0), h(x_k))$
- CLT provides a bound on the error ϵ_n

Markov Chain Monte Carlo Methods

- Given a target density f
  - We build a Markov Kernel K with stationary distr. f
  - We generate a Markov Chain $\{x_t\}$ using this kernel
  - By *stationarity*, samples from the Markov Chain follow f
  - Integrals can be approximated using these samples
    - Supported by *Ergodic Theorem*
- The real effort is in contructing a kernel K such that it is associated with an arbitrary density f

Metropolis-Hastings Algorithm

- Metropolis-Hastings is the most general MCMC algorithm
  - First Introduced by Metropolis and team in 1953
    - one of the top ten most influential algorithms of the 20th century
  - Later extended by Hastings in 1970
- Three components
  1. Generate a *candidate* sample x_{cand} from a proposal distr. $q(x_i|x_{i-1})$
     - $q(x_i|x_{i-1})$ is not the Chain's Kernel K
  2. Compute acceptance probability (α) using proposal distr. q and the full joint density f
  3. Accept the candidate sample with probability α

Metropolis-Hastings Algorithm

Algorithm:

1. Initialize $x_0 \sim q$
2. **for** iteration $i = 1, 2, \ldots$ **do**
3. Propose: $x_{cand} \sim q(x_i|x_{i-1})$
4. Acceptance Prob.:

$$\alpha(x_{cand}|x_{i-1}) = min\{1, \frac{q(x_{i-1}|x_{cand})f(x_{cand})}{q(x_{cand}|x_{i-1})f(x_{i-1})}\}$$

5. $u \sim Uniform(0, 1)$
6. **if** $u < \alpha$ **then**
7. Accept the proposal $x_i \leftarrow x_{cand}$
8. **else**
9. Reject the proposal $x_i \leftarrow x_{i-1}$
10. **end if**
11. **end for**

Guidelines for choosing proposal distribution q

- Choosing a suitable $q(x, y)$ is needed to use Metropolis-Hastings algorithm
- The induced Markov chain should be irreducible, with short mixing time, to allow full coverage of the state-space
  - Support of q should include support of f ($support(f) \subset support(q)$)
- More on this later

MH Algorithm

Deriving Acceptance Probability

$$\alpha(x_{cand}|x_{i-1}) = min\{1, \frac{q(x_{i-1}|x_{cand})f(x_{cand})}{q(x_{cand}|x_{i-1})f(x_{i-1})}\}$$

- Transitions are made according to $K(x_{cand}|x_{i-1}) = q(x_{cand}|x_{i-1})\alpha(x_{cand}$ - Detailed Balance Equations

$$K(x_{i-1}|x_{cand})f(x_{cand}) = p(x_{i-1}, x_{cand}) = K(x_{cand}|x_{i-1})f(x_{i-1})$$

$$q(x_{i-1}|x_{cand})\alpha(x_{i-1}|x_{cand})f(x_{cand}) = q(x_{cand}|x_{i-1})\alpha(x_{cand}|x_{i-1})f(x_{i-1})$$

$$\frac{\alpha(x_{cand}|x_{i-1})}{\alpha(x_{i-1}|x_{cand})} = \frac{q(x_{i-1}|x_{cand})f(x_{cand})}{q(x_{cand}|x_{i-1})f(x_{i-1})} = \rho$$

MH Algorithm

$$\frac{\alpha(x_{cand}|x_{i-1})}{\alpha(x_{i-1}|x_{cand})} = \frac{q(x_{i-1}|x_{cand})f(x_{cand})}{q(x_{cand}|x_{i-1})f(x_{i-1})} = \rho$$

- How do we determine acceptance probability $\alpha(x_{cand}|x_{i-1})$?
- If $(\rho < 1)$, $\alpha(x_{cand}|x_{i-1}) = \rho$ and $\alpha(x_{cand}|x_{i-1}) = 1$
- If $(\rho < 1)$, $\alpha(x_{cand}|x_{i-1}) = \rho/2$ and $\alpha(x_{cand}|x_{i-1}) = 1/2$
  - inefficient, as more samples will be rejected
- To account for scenarios when $\rho > 1$, we use above accept. prob.
- Because this follows *detailed balance equations*, this Markov Chain will coverge to a stationary distribution i.e., our desired target function

MH Algorithm

Interpreting Acceptance Probability

$$\alpha(x_{cand}|x_{i-1}) = min\{1, \frac{q(x_{i-1}|x_{cand})f(x_{cand})}{q(x_{cand}|x_{i-1})f(x_{i-1})}\}$$

- Notice that we only need to know f upto a normalization constant

Two constraints:

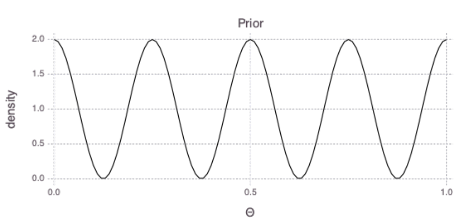
1. The sampler should tend to visit higher probability areas under the full joint density
   - Accounted for by $f(x_{cand})/f(x_{i-1})$
   - When q is symmetric, this is the only part of the acceptance prob. that remains
2. The sampler should explore the space and avoid getting stuck at one site
   - Accounted for by $q(x_{i-1}|x_{cand})/q(x_{cand}|x_{i-1})$

- Opposing forces
  - Moving too far out in the support space & too far out in f is not desired

Example: Sampling

Binomial likelihood with non-standard prior

- $y = [y_1, y_2, \ldots y_n]^T$, where $y_1, \ldots, y_n \sim Bernoulli(\theta)$;
- $S_n = \sum y_i$
- $p(\theta) = 2cos^2(4\pi\theta)$
- $p(y|\theta) = \theta^{N_H}(1-\theta)^{N_T} = \theta^2(1-\theta)^8$, using $y = \{2\ \text{Heads}, 8\text{Tails}\}$



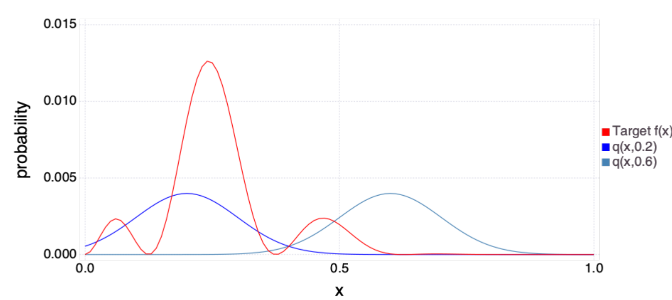
Example: MH Setup

$p(\theta|y) = f(x) = 2\theta^2(1-\theta)^8cos^2(4\pi\theta)$

Candidate/proposal distribution:

$q(x_{cand}|x) = \mathcal{N}(x, 0.1)$

```
f(x) = 2.*x.^2.*(1.-x).^8
         .*cos(4.*pi.*x).^2;
q(x,y) = pdf.(Normal.(y,0.1),x);
```

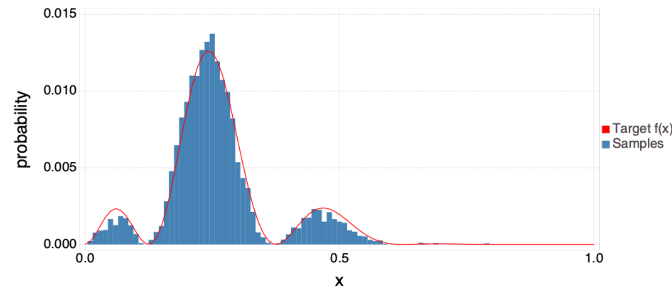


Metropolis Hastings

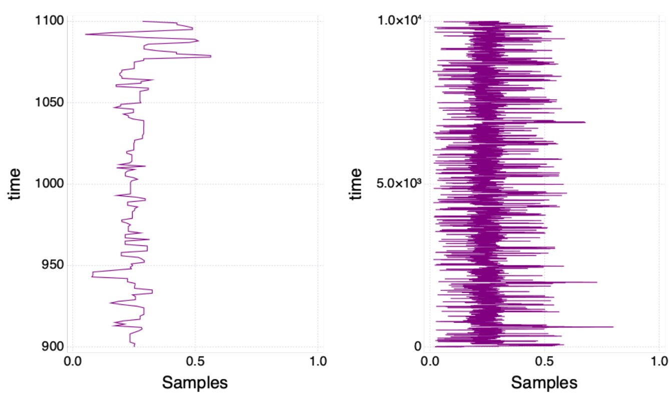
```
function metropolis_hastings(n)
    x = zeros(n); count = 1;
    x[1] = abs(rand(Normal(0,0.1)));

    while(count < n)
        x_cand = rand(Normal(x[count],0.1));
        if((x_cand<0) | (x_cand > 1)) continue; end
        rho = (q(x[count],x_cand)/
                q(x_cand,x[count]))*(f(x_cand)/f(x[count]));
        alpha = minimum([1,rho]);
        u = rand();
        count = count + 1
        if (u < alpha)
            x[count] = x_cand;
        else
            x[count] = x[count-1];
        end
    end
    return x;
end
```


Metropolis Hastings

```
samples = metropolis_hastings(10000);
plot(x = samples, Geom.histogram);
```



Metropolis Hastings: Tracing the Chain



Summary

- For complicated, high-dimensional target distributions
  - Sampling and approximating integrals is challenging
- Markov Chain
  - State-space, Initial distr., Transition probabilities
  - Stationary distrbution
  - Irreducibility, Ergodicity
  - Ergodic theorem guarantees that samples from the chain can be used to compute expectation under the standard distribution
- Markov Chain Monte Carlo methods
  - Developes a Markov Chain whose stationary distribution is the same as the target desntiy
  - We need to choose a good candidate/proposal distribution
  - Ensure that the chain converges to a stationary distribution
    - by complying with *detailed balance equations* in selecting samples