2. Stochastic Process and Markov Chain Monte Carlo

Introduction to Stochastic Process, Introduction to MCMC, MCMC Algorithms, MCMC diagnostics. Skipping measure theoretic details.

Sun Woo Lim

Mar 10, 2022

Motivation of Markov Chain Monte Carlo

We have keep been learning random sampling from a distribution.

1. Inverse CDF method literally useful when inverse function of CDF is obtainable
2. Acceptance Rejection method when the pdf is known and useful when the proposal dist'n is similar as the target dist'n
3. Change of variable technique (key point is identical in distribution!)
4. and basic chain rule of probability (product rule)

Then, how about the following cases when techniques of iid sampling does not work?

1. Case when acceptance rejection technique is very inefficient
2. Case when **the exact form of the pdf is hard (or impossible) to obtain**
    - non-conjugate posterior sampling, known only up to the normalizing constant: $p(\theta|data) = c \cdot P(\theta)p(data|\theta)$
    - Example 12a from Simulation (5th edition), Ross, S.M.

There are many cases when iid sampling from f is hard but sampling from f using MCMC (dependent sample) is easier. Most of such cases are knowing the form of f up to a normalizing constant.

MCMC idea in rough sense

Generate a Markov chain that has stationary distribution same as the target distribution so I obtain the density by histogram and probabilities by sample mean (Monte Carlo). Be careful, no law of large numbers!

Example) Grid Approximation of posterior density (Ch 10.1 in Hoff, P.D (2009))

Data : Y: Number of offspring $\in \{0, 1, \ldots, \}$, x: age of bird.

Model: Poisson Regression (GLM)

- Data Generation Process: Let $\vec{\beta} := (\beta_1, \beta_2, \beta_3)$ and $\vec{x} := (1, x, x^2)$: abuse of notation.
  $Y|X = x \sim Pois(exp(\beta^T \vec{x})) = Pois(exp(\beta_1 + \beta_2 x + \beta_3 x^2))$.
- Prior $\vec{\beta} \sim MVN(\vec{0_3}, 100 I_3)$
- Posterior $p(\vec{\beta}|X, \vec{y}) \propto p(\vec{y}|X, \vec{\beta}) \cdot p(\vec{\beta})$ but normalizing constant $p(\vec{y})$ is intractable (b/c nonconjugate model)

Algorithm (Pseudocode): For each grid pt, get log(unnormalized posterior) = log(prior) + log(lik) $\rightarrow$ exp($\cdot$) $\rightarrow$ normalize

1. Set a $100 \times 100 \times 100$ sized array of **grid** of 3 dimensional $\vec{\beta}$.

   For $i = 1, .., 100$: β_1 grid, For $j = 1, ..., 100$: β_2 grid, For $k = 1, .., 100$: β_3 grid, (triple nested for loop), repeat $2 \sim 5$.
2. $\theta_x = \beta_1[i] \cdot 1 + \beta_2[j] \cdot x + \beta_3[k] \cdot x^2$
3. $log(prior) = log[pnorm(x = \beta_1[i], mn = 0, sd = 10)] \times \ldots \times log[pnorm(x = \beta_3[k], mn = 0, sd = 10)]$
4. $log(likelihood) = sum(log(dpois(x = x, \lambda = exp(\theta_x))))$
5. $log(posterior)[i, j, k] = logprior + loglikelihood$
6. $posterior[i, j, k] = exp(log(posterior)[i, j, k]$ $, \forall i, j, k \in \{1, \ldots, 100\}$: obtain **unnormalized** posterior over all grid pts.
7. $posterior[i, j, k] = \frac{posterior[i,j,k]}{\sum posterior[i,j,k]}$: finally normalized.

✓ This is not recommended because of **curse of dimensionality**.

Basic Terms

1. **Stochastic process**: Family of random variables indexed by index set $\mathcal{I}$ is called a stochastic process.
   : S valued family of random variables $\{X(t)|t \in \mathcal{I}\}$
2. **Index Set**: $\mathcal{I}$ is called the index set, or the parameter set (common to be a set of time).
3. **State Space** "S": The set of different values that the stochastic processes can take.
   - Discrete State Space(Finite or countable) vs Continuous State Space(uncountable)
4. **Sample function = Trajectory = Path function = Path**: single outcome (realization) of a stochastic process.

Types of Stochastic Processes

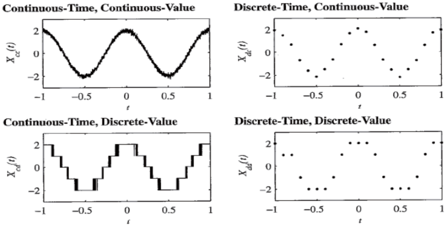


Figure: From https://www.ee.ryerson.ca/ courses/ee8103/chap4.pdf

$\mathcal{I}$	Discrete State Space	Continuous State Space
Discrete Time	Bernoulli Process,Markov Chain, Random Walk	Markov Chain,Random Walk
Continuous Time	Poisson Process, Spatial Point Process	Gaussian Process, Brownian Motion

Table: Types of stochastic processes: examples

Moments

- **Mean function** $m_X(t) := E(X_t) = \int_x x f_{X_t}(x)dx$ or $\sum_x x p_{X_t}(x)$ is a **deterministic function**
- **Autocovariance function (ACVF)** $\gamma_X(s,t) := E(X_t - m_X(t))(X_s - m_X(s))$ is a **deterministic function**
- **Autocorrelation function (ACF)** $\rho_X(s,t) := \frac{\gamma_X(s,t)}{\sqrt{\gamma_X(s,s)\gamma_X(t,t)}}$ is a **deterministic function**

Stationarity

1. Strict Stationary (Strong): The probility distribution of every collection of values $(X_{t_1}, X_{t_2}, ..., X_{t_k})$ is identical to the collection of time-shifted $(X_{t_1+h}, X_{t_2+h}, ..., X_{t_k+h})$. h is called "time-lag".
   In other words, $P[X_{t_1} \leq v_1, X_{t_2} \leq v_2, ..., X_{t_k} \leq v_k] = P[X_{t_1+h} \leq v_1, X_{t_2+h} \leq v_2, ..., X_{t_k+h} \leq v_k]$ for
   - all "number of collection of values" $k = 1, 2, \ldots$
   - for given k, all "time points" $t_1, ..., t_k$
   - for given k, all "values" $v_1, ..., v_k$
   - all "time-lag" h
2. Weak stationarity: when $E(X_1) = E(X_2) = \ldots = E(X_t)$, for all $t = 1, 2, ...$ and $Cov(X_t, X_s) = Cov(X_{t+h} = X_{s+h})$ for all "times" t and s and "lag" h.

1. Gaussian Process

A **continuous time** stochastic process $\{X_t | t \in \mathcal{I}\}$ is called **Gaussian Process** if **every finite collection** of times $t_1, ..., t_k$ follow **multivariate normal distribution**.

Facts

1. With Gaussian Process, the weak stationarity and strong stationarity is equivalent
2. Recall, when $X \sim N(\mu, \Sigma)$, $f(x) = \frac{1}{(2\pi)^{\frac{n}{2}}} exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))$ if Σ is positive definite.
3. Affine transformation of MVN, a linear combination of independent MVN's, Marginal Distribution of MVN, conditional distributions of MVN are all MVN!
4. Generalizing into GP, $X(t) \sim GP(m(t), k(t,s)$, $m(t) := E[X(t)]$, $k(s,t) := Cov(X(t), X(s))$: completely determined by mean function and covariance function

Realizations of GP is a random function, which makes GP used as prior distribution over functions.

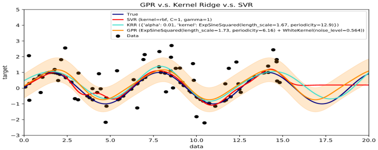


Figure: From https://en.wikipedia.org/wiki/Gaussian_process/media/File:Regressions_sine_demo.svg

2. Random Walk

A **discrete time** stochastic process defined as sums of iid random variables is called **Random walk**.

Simple Random Walk having state space of $\mathbb{Z}$ is $\{X_0, X_1, ...\}$ where $X_0 = 0, X_t = X_{t-1} + \xi_t$, where ξ_n is iid with $\xi_n = 1$ w.p p and $\xi_n = -1$ w.p $1 - p$.

1. It is a special type of Markov Chain because the transition probability only relies on previous state X_{t-1}
2. When $p = 0.5$, it is called **Symmetric Random Walk**

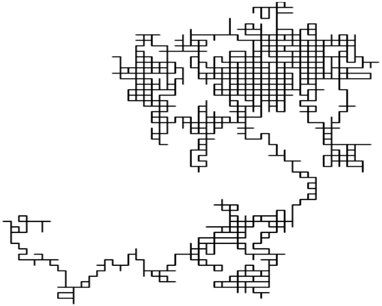


Figure: Random Walk in 2D, from https://en.wikipedia.org/wiki/Random_walk#/media/File:Random_walk_2500.svg

3. Poisson Process

A **continuous time, discrete state space** stochastic process $\{X(t)|t \geq 0\}$ is called a **counting process** if

1. $X(t) \in \{0, 1, 2, ...\}$: non negative integer valued (state space)
2. $\forall t_2 > t_1(\geq 0), X(t_2) \geq X(t_1)$: monotone increasing sequence of random variables
3. For $t_2 > t_1, X(t_2) - X(t_1)$ is the number of events occuring in $(t_1, t_2]$

Poisson process of rate λ is a type of **counting process** satisfying

1. $X(0) = 0$
2. For $0 < t_1 < t_2 < \ldots < t_n$, $X(t_1) \perp [X(t_2) - X(t_1)] \ldots \perp [X(t_n) - X(t_{n-1})]$: independent increments
3. $\forall t_2 > t_1 \geq 0$, $X(t_2) - X(t_1) \sim Poi(\lambda \cdot (t_2 - t_1))$: number of events in interval length t follows $Pois(\lambda t)$

Note) Poisson Process is a generalization of Markov Chain into continuous time.

Theorem

1. $Pr[X(t) = 0] = 1 - \lambda t + o(t)$
2. $Pr[X(t) = 1] = \lambda t + o(t)$
3. $Pr[X(t) \geq 2] = o(t)$

Example) $X(t)$: # times you collect dropped money from your home to Yonsei University, t: time from you departed.

1. The probability of finding money proportional to the interval
2. # times you collect from home to subway station indepedent from # times collect from bus station to Daewoo Hall.
3. For small interval, very less probable that you collect money twice or more.

4.Wiener Process (= Brownian Motion)

A **continuous time, continuous state space** stochastic process $\{X(t)|t \geq 0\}$ is called **Brownian Motion** if
1. $X(0) = 0$
2. $\{X(t)|t \geq 0\}$ has stationary and independent increments
3. $X(t) \sim N(0, \sigma^2 t), \forall t \geq 0$

Note) Weiner process is a generalization of Markov Chain into continuous time.

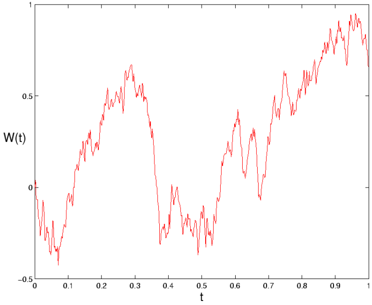


Figure: Weiner process in 1D, from https://sites.me.ucsb.edu/ moehlis/APC591/tutorials/tutorial7/node2.html

Time Homogeneous Markov Chain in Finite State Space

First, deal with Markov Chain on discrete (mostly finite) state space and then, generalize into continuous state space.

$\{X_0, X_1, ...\}$ where $X_t \in \{1, 2, ...\}$ is a Markov Chain on **countable state space** if $Pr(X_{t+1}|X_t, ..., X_0\} = Pr(X_{t+1}|X_t)$.
$\{X_0, X_1, ...\}$ where $X_t \in \{1, 2, ...N\}$ is a Markov Chain on **finite state space** if $Pr(X_{t+1}|X_t, ..., X_0\} = Pr(X_{t+1}|X_t)$.

One Step Transition Probability $p_{ij} := P(X_{t+1} = j|X_t = i)$.
Almost always, deal with **time homogeneous Markov Chain** having one step transition probability indep. from time index t.

When S is finite (finite state space), p_{ij} can be represented by **Transition Probability Matrix** $P = (P_{ij})$ which has:

1) $P_{ij} \geq 0$: Of course, thinking of $P_{ij} := P(X_{t+1} = j|X_t = i)$

2) $\sum_j P_{ij} = 1$: row sum of transition probability matrix is 1. Starting from i, the next state is in $\{1, 2, ...\}$ w.p 1.

3) Information of $p_{ij}^{(n)} := Pr(X_{t+n} = j|X_t = i)$: "n-step transition probability" contained in $P^{(n)} = P \cdot P^{(n-1)}$

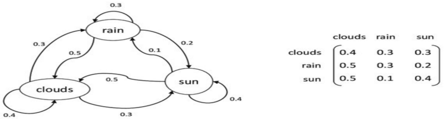


Figure: Markov Chain with S and P can be represented by **labeled directed graph**

Stationary distribution of Markov Chain

$\pi := [lim_{n\to\infty}\frac{1}{n}\sum_{t=1}^{n} I(X_t = 1), lim_{n\to\infty}\frac{1}{n}\sum_{t=1}^{n} I(X_t = 2), \ldots lim_{n\to\infty}\frac{1}{n}\sum_{t=1}^{n} I(X_t = N)]^T$ w.p 1.

In other words, π_j denotes the **long run proportion** that the Markov chain is at state j.

Considering the initial value, state more precisely as $\pi_j = lim_{n\to\infty}\frac{1}{n}\sum_{t=1}^{n} I[X_t = j|X_0 = i] w.p\ 1, \forall i$.

Taking expectation and applying Bounded convergence thm (measure theory),

$\pi_j = lim_{n\to\infty}\frac{1}{n}\sum_{t=1}^{n} Pr[X_t = j|X_0 = i]\ w.p\ 1, \forall i.$

Why need? : Goal is sampling from $X \sim (p(X = 1) = p_1, \ldots, p(X = N) = p_N)$ where independent sampling is hard.

In this case, I generate dependent samples (here, MCMC) well that the **long-run proportion** is $p_1, \ldots, p_N$ in each state.

Seem unrealistic? Refer to Example 12a from Ross, S.M, which is a combinatorial problem with N unknown.

Two properties of $\vec{\pi}$

1. $\sum \pi_i = 1$. If $|S| < \infty$ (finite state space), can use vector notation $\pi' 1 = 1$.
2. $\pi_j = \sum_{i\in S} \pi_i p_{ij}$. If $|S| < \infty$, can use vector notation $\pi' P = \pi'$.

In other words, π is a solution of above two equations. However, solution may not be unique. Need irreducibility!
(https://www.math.is.tohoku.ac.jp/ obata/student/graduate/file/2017-GSIS-ProbModel6-9.pdf)

Irreducibility

A Markov chain on discrete state space(!) is **irreducible** if $\exists t > 0 \ s.t. \ Pr[X_t = j | X_0 = i] > 0, \forall i, j \in S$.
"Wherever you are ($\forall i$), you can visit everywhere ($\forall j$) some time ($\exists \ t > 0$)"

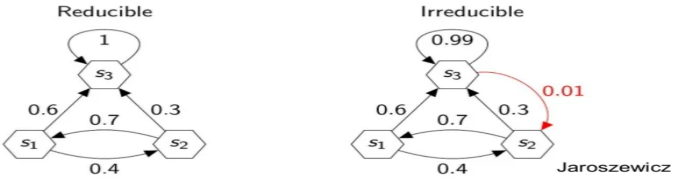


Figure: From https://www.slideshare.net/TomaszKusmierczyk/sampling-and-markov-chain-monte-carlo-techniques

Recurrence and Positive Recurrence

- **"Return time to state i when it started in i"** $\tau_{ii} := min[n \geq 1 | X_n = i | X_0 = i]$
- $f_{ii}^{(n)} := Pr[X_n = i, X_{n-1} \neq i, \ldots, X_1 \neq i | X_0 = i]$ (: prob that first recurrence to i is n^{th} step)
- State i is **recurrent** if $f_i := Pr[\tau_{ii} < \infty] = 1 \leftrightarrow \sum_{n=1}^{\infty} f_{ii}^{(n)} = 1$ and **transient** if $f_i < 1$.
- Recurrent state i is **positive recurrent** if $E[\tau_{ii}] = \sum_{n=1}^{\infty} n \cdot f_{ii}^{(n)} < \infty$ and **null recurrent** if $E[\tau_{ii}] = \infty$.
  (: expected amount of time to return to i given that starting state is i)
- Def) The Markov Chain is positive recurrent if every state in irreducible MC is positive recurrent.

Connection of irreducibility, Positive Recurrence, stationary distribution and MCMC

- Irreducibility is defined on discrete (finite or countable) state space.

- With **irreducibility** + **positive recurrence**, the Markov Chain has **unique** stationary dist'n: $\pi' P = \pi', \sum_{j \in S} \pi = 1$
- **Irreducibility** in finite state space $\rightarrow$ **positive recurrence** satisfied.

- **MCMC**: $\frac{1}{n} \sum_{t=1}^{n} g(X_t) = \sum_{j \in S} \frac{1}{n} \sum_{t=1}^{n} I[X_t = j] g(j) \xrightarrow{p} \sum_{j \in S} \pi_j g(j) = E[g(X)]$.

  : For function g, estimate $E[g(X)]$ as a consistent estimator $\frac{1}{n} \sum_{t=1}^{n} g(X_t)$

  : Approach the stationary distribution by average over time!

  : Meaningful in that I get consistent estimator not by iid sequence but dependent (MC property) sequence.

  : Valuable when it is hard to sample iid sequence $\{X_1, X_2, \ldots\}$ from π (e.g, example 12a, Simulation)

  : **All I need is forming a proper Markov Chain that has stationary probability I need.**

Aperiodic Markov Chain assists convergence to π without time averaging

For a positive recurrent and irreducible chain, approached π_j via time averaging.
Using **aperiodicity**, can approach π without it.

Def) Aperiodicity

A state j is aperiodic if for some $t \geq 0$, $d(j) := gcd[n \geq 1 | P_{jj}^n > 0] = 1$.

If all states in S are aperiodic, the Markov chain is aperiodic.

In the following figure, the leftmost has period 2 for all states, other two are aperiodic MC.

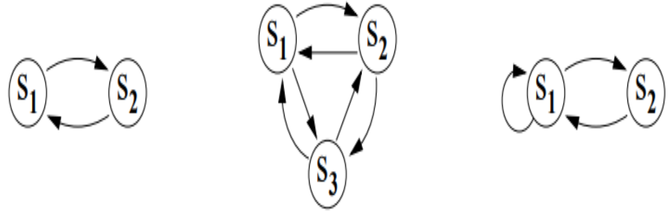


Figure: arrow: positive probability, no arrow: probability of zero. From https://pages.dataiku.com/hubfs/Dataiku%20Dec%202016/Files/lecture3.pdf

Understanding aperiodicity

Aperiodicity is easy to understand as an opposite of periodicity.

Period d means I deterministically know that return to j needs $dk, k \in \mathbb{N}$ number of steps.

I.O.W, if $X_t = j$ and j is d-periodic $(d > 1)$, I am sure that $X_{t+1} \neq j, X_{t+2} \neq j, \ldots, X_{t+d-1} \neq j$.

Be careful) State j being aperiodic does not require $P_{jj} > 0$!! ex) gcd(2,3,5,6,7,...) $= 1$

Aperiodicity helps convergence without time averaging

For **Ergodic** (irreducible, aperiodic in finite state space) Markov chain, obtain

$\pi_j = lim_{t\to\infty} Pr[X_t = j | X_0 = i], \forall i = lim_{t\to\infty} P[X_t = j], j \in S.$

$lim_{t\to\infty} Pr[X_t = j | X_0 = i]$ without time averaging is called **limiting probability**.

Time reversibility helps find π easier

Reverting the Markov Chain

Let a stationary, ergodic Markov Chain $\{X_t\}$ with stationary distribution π.

Reverting the Markov chain lead to $\{\ldots, X(t), X(t-1), X(t-2), \ldots,\}$, which is a Markov chain.

(b/c future and past independent given present $\rightarrow$ past and futre independent given present)

Def) Time reversible MC

A stationary, ergodic MC is time reversible if $\forall i \neq j \in S,\ Q_{ij} := Pr[X_t = j | X_{t+1} = i] = Pr[X_{t+1} = j | X_t = i] = P_{ij}$.

This leads to...

$Q_{ij} := Pr[X_t = j | X_{t+1} = i] = \frac{Pr[X_t = j, X_{t+1} = i]}{Pr[X_{t+1} = i]} = \frac{Pr[X_t = j] \cdot Pr[X_{t+1} = i | X_t = j]}{Pr[X_{t+1} = i]} = \frac{\pi_j P_{ji}}{\pi_i} = P_{ij} \leftrightarrow \pi_j P_{ji} = \pi_i P_{ij}$

Thm) Nonnegative numbers $\pi_1, \ldots \pi_N$ s.t. $\sum_{j \in S} \pi_j = 1$ and $\pi_i P_{ij} = \pi_j P_{ji}$ form stationary dist'n $\vec{\pi} = [\pi_1, \ldots, \pi_N]^T$

proof) $\sum_{i \in S} \pi_i p_{ij} = \sum_{i \in S} \pi_j p_{ji} = \pi_j \sum_{i \in S} p_{ji} = \pi_j$.

This with $\sum_{j \in S} \pi_j = 1$ satisfies two conditions of stationary probabilities.

Note) Almost all MC we use are irreducible, positive recurrent, aperiodic, and time reversible.

Metropolis-Hastings Algorithm in finite state space

Situation

Want to sample from pmf $\pi = \frac{1}{\sum_{j=1}^{N} b_j}[b_1, \ldots, b_N]^T$ but the normalizing constant $\frac{1}{\sum_{j=1}^{N} b_j}$ is intractable.

✓ This means I only know the target pmf (which will be stationary dist'n of MC) **up to a normalizing constant**

✓ Situation seems very unreal but how about cases of N: large and unknown? (e.g, truncation)

✓ Generalizing into continuous state space, intractable normalizing constant is very natural(posterior), so wait!

Metropolis Hastings Algorithm Idea: Now at state i. Think of irreducible proposal MC represented by transition matrix $Q = (q_{ij})$ and accept the proposal with probability α_{ij} to make the resulting chain $P = (p_{ij})$ have stationary dist'n π.

Metropolis Hastings Algorithm

Now at $X_t = i$. Generate proposal X_{t+1}^{prop} from $Pr[X_{t+1}^{prop} = j | X_t = i] = q_{ij}$.

Given $X_{t+1}^{prop} = j$ (realization), $X_{t+1} = j$ (acceptance) w.p α_{ij} or $X_{t+1} = i$ (rejection) w.p $1 - \alpha_{ij}$.

This results $\forall i \neq j \in S$, $p_{ij} = q_{ij}\alpha_{ij}$: "proposed and accepted".

$p_{ii} = q_{ii} + \sum_{k \neq i} q_{ik}(1 - \alpha_{ik})$: "propose i or propose $k(\neq i)$ and rejected"

✓ If P-chain is irreducible, it has stationary dist'n π and solve π easily by assuming time reversibility: $\pi_i p_{ij} = \pi_j p_{ji}$.

Issues

1. Q) Q is what I set. Then, what is α_{ij}?
   A) Calculate α_{ij} as an equation of 1) ratio of π's, which is ratio of b's.

2. Q) Do not know that P is irreducible, which is most important?
   A) There is sufficient condition of Q that makes P-chain irreducible addressed later.

Choice of α_{ij} assuming that P-chain is irreducible

Given the resulting P-chain is irreducible,
$\forall i \neq j \in S, \pi_i p_{ij} = \pi_j p_{ji} \leftrightarrow \pi_i q_{ij} \alpha_{ij} = \pi_j q_{ji} \alpha_{ji} \leftrightarrow b_i q_{ij} \alpha_{ij} = b_j q_{ji} \alpha_{ji} \leftrightarrow \alpha_{ij} = min(\frac{b_j q_{ji}}{b_i q_{ij}}, 1)$

✓ First equivalence: from M-H algorithm formulation addressed in previous slide.
✓ Second equivalence: from $\pi_j = \frac{1}{\sum_{k=1}^{N} b_k} b_j$: "fixed normalizing constant"
✓ Last equivalence: Do by yourself (Hint: divide cases that $\frac{\pi_j q_{ji}}{\pi_i q_{ij}}$ is bigger or smaller than 1).

Sufficient condition of Q that makes P-chain irreducible

1) Q is irreducible, $q_{ij} > 0, \forall i, j$: a **strong** sufficient condition b/c $p_{ij} > 0, \forall i, j$: "one step probability positive".

1) Q is irreducible, 2)$q_{ij} = 0 \leftrightarrow q_{ji} = 0$: a **weak** sufficient condition. Note) Q does not need to be symmetric!

- For i, j s.t. $q_{ij} > 0$, also, $q_{ji} > 0$. Then, $\alpha_{ij} > 0$. Then, both $p_{ij} > 0, p_{ji} > 0$ b/c proposed & accepted with + prob.
- For i, j s.t. $q_{ij} = 0$, also $q_{ji} = 0$. Then, $p_{ij} = p_{ji} = 0$.
  :cannot reach all states in **one** step (weaker!) But, using Q: irreducible, can reach all states in **finite** steps.

Final M-H Algorithm in finite state space

Metropolis-Hastings Algorithm

1. Choose Q with the sufficient condition above
2. Initialization: $t = 0, X_0 = i, i \in \{1, 2, \ldots, N\}$
3. Generate (sample) proposal X_{t+1}^{prop} from Q. Note that current state is i.
4. Proposal is realized as j. Accept proposal w.p $min(\frac{b_j q_{ji}}{b_i q_{ij}}, 1)$
5. $t++$ until $t = n$, n: large. Note, n: MCMC sample size vs $N = |S|$: number of states.

This looks similar as **acceptance-rejection sampling**. However, note two differences (related each other).

1. In A-R method, next value is independent from current value. In M-H Algorithm, next value is dependent from current.
2. In A-R, iterate until acceptance (so, rejection does not count as sample). In M-H, rejection counts as next sample. So, in M-H, can have a long path with same value for a long time.

Metropolis-Hastings Algorithm in continuous state space

Situation

Want to sample from pdf $\pi(x) = \frac{1}{\int_{x \in S} g(x)dx} g(x), x \in S$ but the normalizing constant $\frac{1}{\int_{x \in S} g(x)dx}$ is intractable.

✓ This means I only know the target pdf (which will be stationary dist'n of MC) **up to a normalizing constant**

✓ Now this is a very usual situation in Bayesian analysis (high dimensional, untractable integral).

Metropolis Hastings Algorithm

1. Choose $q(X^{proposal}|X)$: proposal density.
2. Initialization: $t = 0, X_0 = x_0, x_0 \in S$.
3. Generate (sample) proposal X_{t+1}^{prop} from $q(X^{proposal}|X = x_t)$
4. Proposal is realized as x_{t+1}^{prop}. Accept proposal w.p $min[\frac{\pi(x_{t+1}^{prop})q(x_t|x_{t+1}^{prop})}{\pi(x_t)q(x_{t+1}^{prop}|x_t)}, 1] = min[\frac{g(x_{t+1}^{prop})q(x_t|x_{t+1}^{prop})}{g(x_t)q(x_{t+1}^{prop}|x_t)}, 1]$
5. $t++$ until $t = n$, n: large. Note, n: MCMC sample size vs $N = |S|$: number of states.

Types of Metropolis Hastings Algorithm

1. Random Walk Metropolis

The **random walk MH** uses proposal $q(\cdot)$ using a **random walk** from the current state $X_t = x_t$.

Make proposal $X_{t+1}^{proposal} = X_t + \zeta$ where ζ is symmetric w.r.t $0 \leftrightarrow X_{t+1}^{prop}|X_t$ symmetric w.r.t. X_t

I.O.W, use $q(\cdot)$ s.t. $q(X^{proposal}|X) = q(X|X^{proposal})$.

Then, accept proposal w.p $min[\frac{\pi(x_{t+1}^{prop})q(x_t|x_{t+1}^{prop})}{\pi(x_t)q(x_{t+1}^{prop}|x_t)}, 1] = min[\frac{\pi(x_{t+1}^{prop})}{\pi(x_t)}, 1] = min[\frac{g(x_{t+1}^{prop})}{g(x_t)}, 1]$ (same normalizing const)

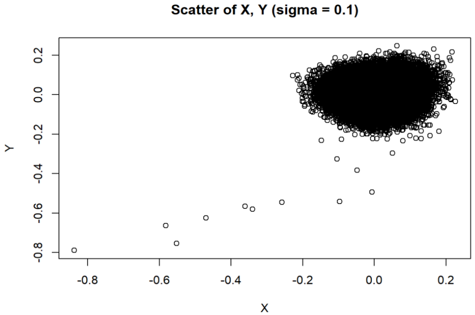
Ex) Sample from pdf proportional to $g(x) = exp(-(\frac{x}{2})^8)$ using random walk Metropolis algorithm
Similar example of sampling $N(0,1)$ R.V's is in https://bookdown.org/rdpeng/advstatcomp/metropolis-hastings.html.

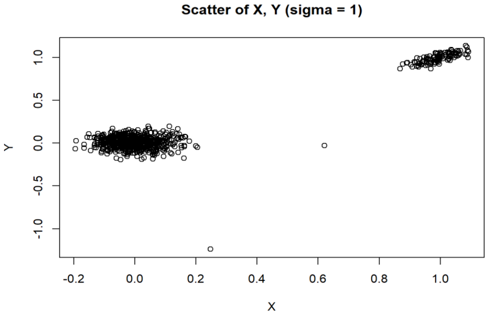
Algorithm

1. $q(x^{proposal}|x) = \frac{1}{2\delta} I[x^{proposal} \in (x-\delta, x+\delta)] \leftrightarrow X^{proposal}|X \sim Unif(X-\delta, X+\delta)$
2. Accept proposal with $min[\frac{\pi(x_{t+1}^{prop})}{\pi(x_t)}, 1] = min[\frac{g(x_{t+1}^{prop})}{g(x_t)}, 1]$

Example) Let bivariate random variable $X := (X_1, X_2)$. $f(x_1, x_2) \propto exp(-150[(x^2 - y)^2 + (x - y^2)^2])$

Proposal density $q(X^{proposal}|X) = dMVN(\vec{x^{proposal}}, mean = \vec{x}, Cov = \sigma^2 I_2) = q(X|X^{proposal})$





Importance of appropriate σ^2

1. Small σ^2 $\rightarrow$ Acceptance probability $\uparrow$, but navigate only locally.
2. Large σ^2 $\rightarrow$ Acceptance probability $\downarrow$, so, keep staying at the current position.

The **direction** of the next sample is decided randomly. Can be inefficient b/c navigating similar regions repeatedly.

2. Independence Sampler

Be careful, this is not iid sampling scheme despite the name!

Independence sampler uses $q_{X^{prop}|X}(x^{proposal}|x)$ does not depend on x.

Thus, $q_{X^{prop}|X}(x^{proposal}|x) = q(x^{prop})$

Thus, accept proposal w.p $min[\frac{\pi(x_{t+1}^{prop})q(x_t)}{\pi(x_t)q(x_{t+1}^{prop})}, 1] = min[\frac{g(x_{t+1}^{prop})q(x_t)}{g(x_t)q(x_{t+1}^{prop})}, 1]$

3. Gibbs Sampler

Situation

Want to sample from random vector $X = (X_1, \ldots, X_d)$. $X \sim \pi(\cdot)$. $\pi(x) \propto g(x)$: "knowing up to normalizing constant"

Assume $X = (X_{(1)}, \ldots, X_{(k)})$, $k \leq d$: decomposed as subvectors.

Denote $X_{(j)}$ to be the j^{th} subvector and $X_{-(j)}$ be the remainder.

Gibbs sampling used when 1)sampling from π directly: hard, 2) but sampling from full conditional $p(X_{(j)}|X_{-(j)})$: possible.

Many times, set $k = d$: each subvector is each scalar component.

Gibbs Sampler Algorithm

1. Initialization: $t = 0, X_0 = (x_1, \ldots, x_d)$
2. Updated index sampling: $i \sim Unif[1, 2, \ldots, d]$. "i" stands for index.
3. For given update index i, propose i^{th} component $X_{t+1}^{proposal}[i]$ from full conditional $p(x|x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$
4. Let x be realization of $X_{t+1}^{proposal}[i]$. Proposal is always accepted and $X_{t+1} = (x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_d)$
5. $t++$ until $t = n$, iterate from index sampling.

Gibbs Sampling Steps illustrated recursively

1. Current sample $X_t = (x_1, \dots, x_i, \dots, x_d)$.
2. Chose index i to update.
3. $X_{t+1} = (x_1, \dots, x_i^{new}, \dots, x_d)$

Gibbs Sampling: Special case of M-H Algorithm

By Gibbs Sampler Algorithm, $q(x_{t+1}|x_t) = \frac{1}{d} \cdot \pi(x_i^{new}|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)$: using $p(z,w) = p(z)p(w|z)$.

$= \frac{1}{d} \frac{\pi(x_{t+1})}{\pi(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)}$: by definition of conditional distribution.

Fitting into M-H Algorithm, Accept proposal w.p $min[\frac{\pi(x_{t+1}) \cdot q(x|x_{t+1})}{\pi(x_t) \cdot q(x_{t+1}|x)}, 1]$

$$= min[\frac{\pi(x_{t+1}) \cdot \frac{1}{d} \cdot \frac{\pi(x_t)}{\pi(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)}}{\pi(x_t) \cdot \frac{1}{d} \cdot \frac{\pi(x_{t+1})}{\pi(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)}}, 1] = min(1,1) = 1$$

✓ Gibbs Sampling is a special type of M-H Algorithm with acceptance probability 1!

Example) Generating Finite Mixture Normal with Gibbs Sampling (Hoff, Ch6.6)

We inspected how we sample finite mixture normal distribution with independent sampling.

Able to sample using Gibbs Sampling too!

"Groups" $d \in \{1, 2, 3\}$, "Means" $(\mu_1, \mu_2, \mu_3) = (-3, 0, 3)$, "Variances" $(\sigma_1^2, \sigma_2^2, \sigma_3^2) = (1/3, 1/3, 1/3)$.

```
### in the full conditional posterior of d, the sum of "prob" is not 1 (since it is unnormalized).
### However, it doesn't matter in "sample" function!  e.g, sample(1:3, prob = c(0.1, 0.2, 0.3))

#### MCMC sampling
set.seed(1)
th = 0 # initialization!!
THD.MCMC<-NULL # placeholder
S = 10000
for(s in 1:S) {
  d<-sample( 1:3 , 1, prob= w*dnorm(th,mu,sqrt(s2))   ) # full conditional post of d (p100)
  th<-rnorm(1,mu[d],sqrt(s2[d]) ) # full conditional post of theta (already provided)
  THD.MCMC<-rbind(THD.MCMC,c(th,d) )
}
#### Figure 6.5
pdf("fig6_5.pdf",family="Times",height=3.5,width=7)
par(mfrow=c(1,2),mar=c(3,3,1,1),mgp=c(1.75,.75,0))
Smax<-1000
ths<-seq(-6,6,length=1000)
plot(ths, w[1]*dnorm(ths,mu[1],sqrt(s2[1])) +
       w[2]*dnorm(ths,mu[2],sqrt(s2[2])) +
       w[3]*dnorm(ths,mu[3],sqrt(s2[3])) ,type="l" , xlab=expression(theta),
     ylab=expression( paste( italic("p("),theta,")",sep="") ),lwd=2 ,ylim=c(0,.40))
hist(THD.MCMC[1:Smax,1],add=TRUE,prob=TRUE,nclass=20,col="gray")
lines( ths, w[1]*dnorm(ths,mu[1],sqrt(s2[1])) +
         w[2]*dnorm(ths,mu[2],sqrt(s2[2])) +
         w[3]*dnorm(ths,mu[3],sqrt(s2[3])),lwd=2 )
plot(THD.MCMC[1:Smax,1],xlab="iteration",ylab=expression(theta))
dev.off()
```

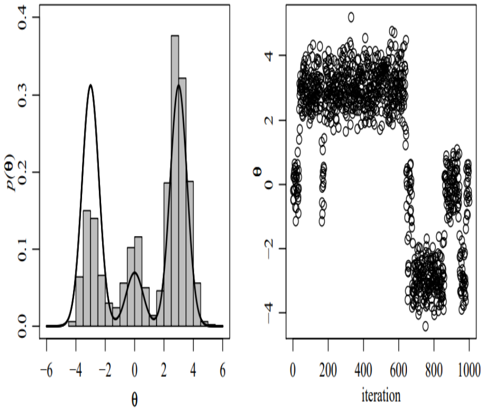


Fig. 6.5. Histogram and traceplot of 1,000 Gibbs samples.

4. Hamiltonian Monte Carlo (HMC) : algorithm used in RStan

Goal: Sample from $\pi(x)$. Many times, it is used in Bayesian analysis so, $\pi(x)$ is $p(\theta|data)$.

Background and HMC idea

Inefficiency of Metropolis algorithm: long time zig-zagging for the target dist'n (random walk behavior).
HMC: move faster to the target by suppressing random walk behavior using **momentum** concept.
Introduce new momentum variable $\rho \rightarrow$, draw from $\pi(x, \rho) = \pi(\rho|x)\pi(x)$.

Hamiltonian

$H(x, \rho) := -log\pi(x, \rho) = -log\pi(\rho|x) - log\pi(x) = T(\rho|x) + V(x) =$ "kinetic energy" + "potential energy"

HMC Algorithm

1. Initalization: $t = 0, X_0 = x_0$
2. $\rho \sim MVN(0, \Sigma)$: generate momentum. (in RStan, use $\pi(\rho|x) = \pi(\rho)$ and Σ: diagonal).
3. For small $\epsilon > 0$, repeat the following leapfrog steps L times
    1. $\rho = \rho - \frac{\epsilon}{2}\frac{\partial V}{\partial x}|_{x=x_t}$: "half step update of momentum"
    2. $x_t = x_t + \epsilon\Sigma^{-1}\rho$: "full step update of the position"
    3. $\rho = \rho - \frac{\epsilon}{2}\frac{\partial V}{\partial x}|_{x=x_t}$: "half step update of momentum" again.
4. (ρ^*, x_t^*) denotes the (ρ, x_t) after L times. $X_{t+1} = x_t^*$ w.p $min[exp(H(x, \rho) - H(x^*, \rho^*), 1]$
5. $t++$ until $t = n$, iterate from the leapfrog step.

Interpretation of the Algorithm

- HMC incorporates MCMC and deterministic differentiation $\rightarrow$ also called **hybrid MC**
- Although simulating from $\pi(x, \rho)$, ρ is only auxiliary, only interested in $\pi(x)$. ρ: for moving faster in $support(X)$.
- Proposal for the next X is related largely to ρ.
- Note that $V(x)$ is defined as "negative" $log\pi(x)$. So, want to go where $V(x)$ is small.
- **momentum update**: Since I go to x s.t. $V(x)$ is small, if the gradient is $+$, go backward and if gradient is -, go forward.
- **position update**: Move with modified ρ.

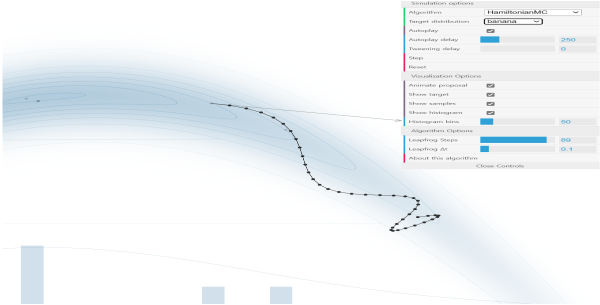


Figure: HMC demo from http://chi-feng.github.io/mcmc-demo/app.html?algorithm=HamiltonianMCtarget=banana

Example Codes of Metropolis-Hastings and Hamiltonian MC

https://stephens999.github.io/fiveMinuteStats/MH-examples1.html : Simple M-H for generating exponential dist'n

https://jonnylaw.rocks/posts/2019-02-11-metropolis_r/ : MH algorithm for bivariate normal

https://jonnylaw.rocks/posts/2019-07-31-hmc/ : HMC for bivariate normal

5. Reversible Jump MCMC

MCMC diagnostics

With proper proposal density, MCMC leads to the proper stationary distribution.
However, MCMC has downsides that all originate from correlation between samples.

- burn in: early samples highly related to x_0.
- very low convergence
- not clear when the convergence happened (true density multimodal?)

1. Representativeness : whether the samples represent the target
Get a **hint(!)** of it by plot of several paths (trace plot) with different x_0's.
Using trace plot, erase burn-in period.

Gelman-Rubin statistic : numerical diagnostic method. Calculate $\hat{R} > 1$ and if R is big, keep sampling

Idea) If convergence, variance within the chain $\approx$ variance between the chains.

m: number of MCMC to runs, n: number of sample size per chain.
$\bar{\phi_{.j}} = \frac{1}{n}\sum_{i=1}^{n}\phi_{ij}$: mean of each chain, $\bar{\phi_{..}} = \frac{1}{m}\sum_{i=1}^{m}\bar{\phi_{.j}}$: mean of all chains.
"Between sequence variance" $B := \frac{n}{m-1}\sum_{j=1}^{m}(\bar{\phi_{.j}} - \bar{\phi_{..}})^2$
"Within sequence variance" $W := \frac{1}{m}\sum_{j=1}^{m}[\frac{1}{n-1}\sum_{i=1}^{n}(\phi_{ij} - \bar{\phi_{.j}})^2]$

"Potential scale reduction" $\hat{R} := \sqrt{\frac{\frac{n-1}{n}W + \frac{1}{n}B}{W}}$. Check if $\hat{R} \approx 1$ or $>> 1$.
Numerator estimates, while denominator underestimates $Var(\phi)$ for finite n.

2. Accuracy : whether the MCMC estimate is accurate

We want MCMC estimates (mean, variance, quantiles, etc) to be accurate (i.e, small standard error!)

Principle: more "information", less standard error.

However, MCMC samples of size n gives less information than n iid samples $\because$ correlation!

Calculate **effective sample size** as a measure of 'how much information of iid sample does the chain have'.

$ESS := \frac{mn}{\sum_{t=-\infty}^{\infty} ACF(t)}$

- $ACF(t)$ denotes the autocorrelation of the MCMC sequence at lag t.
- Drastic cases: $ACF(t) = 0, \forall t \neq 0$, then $ESS = mn$, $ACF(t) = 1, \forall t$, then $ESS = 1$.
- Using $ACF(t) = ACF(-t)$ and $ACF(0) = 1$, $ESS = \frac{mn}{1+2\sum_{t=1}^{\infty} ACF(t)}$

Using ESS, can obtain **Markov Chain Standard Error (MCSE)**, that is $MCSE = \frac{stdev\ of\ a\ MC}{\sqrt{ESS}}$

References

https://www.ee.ryerson.ca/ courses/ee8103/chap4.pdf : types of stochastic processes
STAT 3124 lecture note, Taeyoung Park, Yonsei University :stochastic processes examples
https://en.wikipedia.org/wiki/Random_walk#/media/File:Random_walk_2500.svg : image
https://en.wikipedia.org/wiki/Gaussian_process/media/File:Regressions_sine_demo.svg : image
https://sites.me.ucsb.edu/ moehlis/APC591/tutorials/tutorial7/node2.html : image
https://en.wikipedia.org/wiki/Markov_chain : Markov Chain description
https://www.researchgate.net/publication/330360197_Decision_Support_Models_for_Operations_and_Maintenance_for_Offshore_Wind_Farms_A_Review/figures?lo=1 : image
https://www.slideshare.net/TomaszKusmierczyk/sampling-and-markov-chain-monte-carlo-techniques : image
https://people.engr.tamu.edu/andreas-klappenecker/csce658-s18/markov_chains.pdf : aperiodicity definition and figures
https://www.math.is.tohoku.ac.jp/ obata/student/graduate/file/2017-GSIS-ProbModel6-9.pdf : MCMC theory in finite state space
http://www.columbia.edu/ ks20/stochastic-I/stochastic-I-MCII.pdf : MCMC theory in finite state space
https://sites.pitt.edu/ super7/19011-20001/19561.pdf : MCMC theory in finite state space
https://pages.dataiku.com/hubfs/Dataiku%20Dec%202016/Files/lecture3.pdf: image
https://bookdown.org/rdpeng/advstatcomp/metropolis-hastings.html : random walk Metropolis example
Gelman, Andrew, J. B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2013. Bayesian Data Analysis. Third. London: Chapman Hall/CRC Press. : HMC description, MCMC diagnostics
https://mc-stan.org/docs/2_19/reference-manual/hamiltonian-monte-carlo.html : HMC description
Hoff, P. D. (2009). A first course in Bayesian statistical methods (Vol. 580). New York: Springer. : example, MCMC diagnostics
https://hun-learning94.github.io/posts/bayesian-ml/week3/02-mcmc-approximation-for-bayesian-posterior/ : HMC explanation
http://chi-feng.github.io/mcmc-demo/app.html?algorithm=HamiltonianMCtarget=banana : HMC demo
https://stephens999.github.io/fiveMinuteStats/MH-examples1.html : Simple M-H for generating exponential dist'n
https://jonnylaw.rocks/posts/2019-02-11-metropolis_r/ : MH algorithm for bivariate normal
https://jonnylaw.rocks/posts/2019-07-31-hmc/ : HMC for bivariate normal