STA 630: Bayesian Inference - Chapter 4

Zeya Wang

University of Kentucky

Spring 2025

Lecture 7: Monte Carlo Methods - Outline

- Integration and Monte Carlo methods
- Acceptance-Rejection Sampling
- Importance Sampling

Integration and Monte Carlo methods

- We have already used Monte Carlo methods plenty of times!!
- It is the most used technique in statistics to perform integral approximation. Briefly, the method relies on computer simulation of random variables to produce an approximation of the integrals that converges with the number of simulations.
- Suppose we are interested in evaluating

$$E[f(\theta)] = \int f(\theta)p(\theta)d\theta$$

which can be seen as the expectation of a function, $f(\cdot)$, of a random variable θ, having density $p(\cdot)$

Approximating the integral

- Suppose we could take a sample of m values from $p(\theta)$ (that could be your posterior distribution of $p(\theta \mid x)$)

$$\theta^{(1)}, \ldots, \theta^{(m)} \overset{iid}{\sim} p(\theta)$$

for large m.

- Law of Large Numbers:

$$\begin{array}{rcll} \bar{\theta}_{MC} & = \frac{1}{m} \sum \theta^{(i)} & \rightarrow & E[\theta] \\ \overline{f(\theta)}_{MC} & = \frac{1}{m} \sum f\left(\theta^{(i)}\right) & \rightarrow & E[f(\theta)] \end{array}$$

Other examples

- Cumulative ordered values approximate $F(\theta)$ (cdf)
- Empirical distribution of the sample $\theta^{(1)}, \ldots, \theta^{(m)}$ approximates $p(\theta)$ (use histogram or kernel density estimator)
- $P(f(\theta) > c)$ approximated by proportion of samples where event $\left(f\left(\theta^{(i)}\right) > c\right)$ occurs
- Sample moments/quantiles/functions approximate true moments/quantiles/functions

Extends easily to higher dimensional parameters.

Monte Carlo variance

- If

$$\int \theta^2 p(\theta) d\theta < +\infty$$

the central limit theorem shows that

$$\frac{\bar{\theta}_{MC} - \int \theta p(\theta) d\theta}{\sqrt{\sigma^2/m}} \to N(0,1)$$

where $\sigma^2 = \text{Var}[\theta]$. We can obtain the Monte Carlo standard error by means of

$$\text{Var}\left[\theta_{MC}\right] \simeq \hat{\sigma}^2 = \frac{1}{m} \sum \left[\theta^{(i)} - \bar{\theta}_{MC}\right]^2$$

- An approximate 95% Monte Carlo confidence interval for the mean of θ is then $\bar{\theta}_{MC} \pm 1.96\sqrt{\hat{\sigma}^2/m}$, that is we expect the posterior mean of θ to be in this interval for roughly 95% of repeated MC samples.
- To increase accuracy, increase m.

Monte Carlo variance

- Similarly for $f(\theta)$, if

$$\int [f(\theta)]^2 p(\theta) d\theta < +\infty$$

the central limit theorem shows that

$$\frac{\overline{f(\theta)}_{MC} - \int f(\theta) p(\theta) d\theta}{\sqrt{\sigma_f^2 / m}} \to N(0,1)$$

where $\sigma_f^2 = \text{Var}[f(\theta)]$. We can obtain confidence bounds for $\overline{f(\theta)}_{MC}$ and estimate the asymptotic variance by means of

$$\text{Var}\left[f(\theta)_{MC}\right] \simeq \hat{\sigma}_f^2 = \frac{1}{m} \sum \left[f\left(\theta^{(i)}\right) - \overline{f(\theta)}_{MC} \right]^2$$

- An approximate 95% Monte Carlo confidence interval for the mean of $f(\theta)$ is $\overline{f(\theta)}_{MC} \pm 1.96\sqrt{\hat{\sigma}_f^2 / m}$.

Example

Suppose $x_1, \ldots, x_n$ are iid $N(\mu, \sigma^2)$ and $\pi(\mu, \sigma^2) \propto \frac{1}{\sigma^2}$. Let $\tau = 1/\sigma^2$. Recall that

$$\tau \mid x \sim \text{Gamma}\left(\frac{n-1}{2}, \frac{(n-1)s^2}{2}\right),$$

$$\mu \mid \tau, x \sim N\left(\bar{x}, \frac{1}{n\tau}\right),$$

where $s^2 = \frac{1}{n-1}\sum [x_i - \bar{x}]^2$.

Example

We can generate samples from the posterior of (μ, σ^2) as follows

1. Sample $\tau_i \sim \text{Gamma}\left(\frac{n-1}{2}, \frac{(n-1)s^2}{2}\right)$,
2. Then sample $\mu_i \sim N\left(\bar{x}, \frac{1}{n\tau_i}\right)$

$i = 1, \ldots, m$. This process creates the set $\left\{\left(\mu_i, \sigma_i^2\right), i = 1, \ldots, m\right\}$ from $p\left(\mu, \sigma^2 \mid x\right)$. To estimate, for example, the posterior mean of μ, we would use

$$\hat{E}[\mu \mid x] = \frac{1}{m} \sum \mu_i,$$

or, to obtain a 95% HPD interval for μ, we simply use the empirical 0.025 and 0.975 quantiles of the sample of μ_i values.

Summary so far

- We have seen that Monte Carlo sampling is a useful tool for sampling from distributions
- In simple models, especially with conjugate prior distributions, it is often easy to derive the posterior distribution in closed form. MC sampling is convenient (though not really necessary).
- When the posterior density does not have a recognizable form, it might be possible to factor the distribution analytically and simulate in parts, as we did for the Normal model.
- For more complicated problems, it may not possible to directly generate samples from the target distribution (e.g., non-conjugate settings and, in general, situations where we cannot sample from the joint posterior distribution)
- We will see indirect sampling schemes first and then introduce Markov chain Monte Carlo (MCMC) methods.

Indirect Methods

If we cannot directly sample from the joint posterior then we cannot use a plain Monte Carlo algorithm.
We need to search for ways to sample from then joint posterior indirectly. Several methods have been proposed in the literature for indirect sampling. We will discuss two of them.
These are

1. Acceptance-Rejection Sampling
2. Importance Sampling

Acceptance-Rejection Sampling

- Many distributions from which it is difficult, or even impossible, to directly simulate.
- It is a class of methods that only require us to know the functional form of the density f of interest only up to a multiplicative constant.
- The key to this method is to use a simpler (simulation-wise) density g, the **instrumental density** or **envelope function**, from which the simulation from the target density f is actually done.

Accept-Reject algorithm

Given a density of interest f, find a density g and a constant M such that:

$$f(x) \leq Mg(x)$$

on the support of f.

1. Generate $X \sim g, U \sim U_{[0;1]}$
2. Accept $Y = X$ if $U \leq f(X)/Mg(X)$
3. Return to 1. otherwise.

Accept-Reject algorithm

The members of $\{Y_j, j = 1, \ldots, N\}$ will then be variables from $f(\cdot)$.
Intuition suggests that M should be chosen as small as possible, so as not to waste too many samples.
This is easy to confirm. Let k denote the number of iterations required to get one acceptable candidate Y_j, then k is a geometric random variable. That is,

$$P(k = i) = (1 - p)^{i-1} p,$$

where p is the probability of acceptance.

Remarks

- We can show that the probability of acceptance

$$p = \text{Prob}\left(U < \frac{f(X)}{Mg(X)}\right) = \frac{1}{M}$$

- Since $k \sim \text{Geom}(p)$,

$$E(k) = \frac{1}{p} = M, \text{Var}(k) = \frac{1-p}{p^2} = M^2 - M$$

- Hence, M should be kept as small as possible.

Choice of g

- must be simple (uniform, triangular, normal, any density that can be obtained by inversion)
- Since the uniform distribution must be defined on a bounded support, also

$$\frac{f(x)}{Mg(x)} \text{ must be bounded}$$

Hence, $g(y)$ must have heavier tails and sharper infinite peaks than f.

How to find M and g

- Choose a class of densities for g and then find that density for which M is the smallest.
- Example generate normals from double exponential.
- We assume $f(x) = N(0,1)$ and $g(x \mid \alpha)$ is a double exponential density

$$g(x \mid \alpha) = \frac{\alpha}{2} \exp(-\alpha|x|), \quad \alpha > 0$$

In general

$$M = \sup_x \left(\frac{f(x)}{g(x)} \right)$$

How to find M and g

It is then straightforward to show that

$$\frac{f(x)}{g(x \mid \alpha)} \leq \sqrt{2/\pi}\alpha^{-1}e^{\alpha^2/2}$$

and that the minimum of this bound (in α) is obtained for $\alpha = 1$. The probability of acceptance is then $\sqrt{\pi/2e} = 0.76$ to produce one normal random variable, this A-R algorithm requires on average $1/0.76 \approx 1.3$ uniform variables.

How to find M and g

```
set.seed(1)
n = 10000
x.rs = rep(NA, n); accpt = rep(NA, n)
for(i in 1:n) {
x = rdexp(1); u = runif(1)
if(u < exp(-.5*(abs(x)-1)^2)) {
x.rs[i]=x
accpt[i]=1 }
else accpt[i]=0
}
z = rnorm(10000)
x.dir = z
qqplot(x.rs, x.dir)
```

Importance sampling

- Suppose that $f(x)$ can be approximated by some density $g(x)$, from which we can easily sample.
- The importance sampling approach is based on the principle that

$$\begin{aligned} E(h(x)) = \mu &= \int h(x) f(x) dx \\ &= \int h(x) \frac{f(x)}{g(x)} g(x) dx \\ &= \frac{\int h(x) \frac{f(x)}{g(x)} g(x) dx}{\int \frac{f(x)}{g(x)} g(x) dx} = \frac{E_g[h(x)w(x)]}{E_g[w(x)]} \end{aligned}$$

where g is called **importance sampling distribution** or **envelope** and $w(x) = f(x)/g(x)$ is the importance weight function.

Importance sampling - unstandardized weights

The previous equations suggest the following two ways to estimate $E(h(X))$ by means of a MC approach:

1. Draw $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} g$ and use the estimator

$$\hat{\mu}^*_{IS} = \frac{1}{n} \sum_{i=1}^{n} h(X_i)\, w^*(X_i)$$

where $w^*(X_i) = f(X_i)/g(X_i)$ are unstandardized weights (**importance ratios**).
Of course, it must be easy to sample from g and evaluate f, even when it is not easy to sample from f.

Importance sampling - standardized weights

2. Otherwise, we can draw $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} g$ and compute

$$\hat{\mu}_{IS} = \sum_{i=1}^{n} h(X_i)\, w(X_i)$$

where $w(X_i) = w^*(X_i) / \sum_{i=1}^{n} w^*(X_i)$ are standardized weights. This approach is particularly appealing when f is known up to a proportionality constant (e.g. posterior in Bayesian inference).

Importance sampling

$$E(h(x)) \simeq \frac{1}{n} \sum_{i=1}^{n} h(X_i)\, w^*(X_i)$$

- The importance weighting allows us to over sample a portion of the domain of the function that receives lower probability under the target distribution.
- Also, the same sample generated from g can be used repeatedly, not only for different functions $h(\cdot)$, but also for different densities $f(x)$, which is quite attractive for robustness and Bayesian sensitivity analysis.

f and g

(a) the support of g must include all the support of f
(b) the ratio $f(x)/g(x)$ must be bounded
(c) g must have heavier tails than f
(d) Since

$$E\left(\frac{f(X)}{g(X)}\right) = \int \frac{f(x)}{g(x)} g(x)dx = 1$$

and $g(x)$ has heavier tails than f, the variance of $\frac{f(X)}{g(X)}$ must be large.

If we desire to use the importance sampling approach to reduce the variance of the MC estimate, then it is better to choose $g(x)$ so that $\frac{f(X)}{g(X)}$ is large when $h(x)$ is small, and viceversa. If $g(X)$ is a poor approximation, many of the weights will be close to zero, and thus few X_i 's will dominate the sums, producing an inaccurate approximation.

Example

Suppose $X \sim \text{Exp}(\lambda), \lambda = 1$. We want to compute $E[\sqrt{(X)}] = \int \sqrt{(x)} f(x) dx$ We consider three different importance density functions.

```
set.seed(1)
samp.size=10000
lambda=1

## storage
x <- matrix(NA, samp.size, 3)
wts <- matrix(NA, samp.size, 3)

## Envelopes
#Case 1: abs(Normal(0,1))
x[,1]=abs(rnorm(samp.size))
wts[,1]=dexp(x[,1], rate=lambda)/(dnorm(x[,1])*2)
```

Example

```
#Case 2: uniform(0,1000)
x[,2]=runif(samp.size,0,1000)
wts[,2]=dexp(x[,2], rate=lambda)/(dunif(x[,2],0,1000))

#Case 3: abs(Cauchy(0,1))
x[,3]=abs(rcauchy(samp.size))
wts[,3]=dexp(x[,3], rate=lambda)/(dcauchy(x[,3])*2)

## Importance sampling estimate of E[sqrt(X)]
mu.is = rep(NA, 0)
for(i in 1:3)
mu.is[i] = sum(sqrt(x[,i])*wts[,i])/sum(wts[,i])
```

Effective Sample Size

Effective sample size to measure the efficiency of an importance sampling strategy using an envelope g. For the unstardized weights, the ESS $N(f, g)$ is

$$N(f, g) = \frac{n}{1 + \hat{V}\{w^*(x)\}},$$

where $\hat{V}\{w^*(x)\}$ is the sample variance of the $w^*(x_i)$. If the standardized weights are used, then

$$N(f, g) = \frac{n}{1 + \hat{cv}^2\{\tilde{w}(x)\}},$$

where $\hat{cv}\{\tilde{w}(x)\}$ is the sample standard deviation of the standardized importance weights divided by their sample mean. The ESS indicates that n weighted samples used in IS are worth $N(f, g)$ unweighted i.i.d. samples drawn exactly from f and used in a simple Monte Carlo estimate.

Lecture 8: Monte Carlo Methods I - Outline

- Markov Chains
- Markov Chain Monte Carlo
- Metropolis-Hastings algorithm

Summary so far

- We have seen that Monte Carlo sampling is a useful tool for sampling from distributions
- For more complicated problems, it may not possible to directly generate samples from the target distribution (e.g., non-conjugate settings and, in general, situations where we cannot sample from the joint posterior distribution)
- We have seen indirect sampling schemes
- We now introduce Markov chain Monte Carlo (MCMC) methods

Brief review of Markov Chains (discrete case)

A Markov chain is a sequence of random variables $X_1, X_2, X_3, \ldots$ with the Markov property, namely that, given the present state, the future and past states are independent. The possible values of X_i form a countable set S called the state space of the chain.

$$\Pr(X_{n+1} = x_j \mid X_n = x_n, \ldots, X_1 = x_1) = \Pr(X_{n+1} = x_j \mid X_n = x_n)$$

Properties of a Markov Chain:

1. Reducibility
2. Periodicity
3. Recurrence
4. Ergodicity

Markov Chains properties:

- **Reducibility** A state j is said to be accessible from a state i if a system started in state i has a non-zero probability of transitioning into state j at some point.

$$\Pr\left(X_n = j \mid X_0 = i\right) = p_{ij}^{(n)} > 0$$

- **Irreducibility** A chain is irreducible if all states communicate (there's a positive probability to visit all states in a finite number of steps).
- **Periodicity** A state i has period k if any return to state i must occur in multiples of k time steps. If $k = 1$, then the state is said to be aperiodic i.e. returns to state i can occur at irregular times.

Markov Chains properties:

- **Recurrence** A state i is said to be **transient** if, given that we start in state i, there is a non-zero probability that we will never return to i. A state i is called **absorbing** if it is impossible to leave this state. A state i is **recurrent** if the chain will return to the state with probability 1 at some point in the future: $P(\text{return to } i \mid X_0 = i) = 1$. This means that if the Markov chain starts in state i, it is guaranteed to eventually revisit i infinitely often (expected number of visit to i is equal to infinity).
  1. Positive Recurrent: The expected return time to state s is finite.
  2. Null Recurrent: The state is recurrent, but the expected return time is infinite.
- **Ergodicity: to what is the chain converging?**

$$\lim_{n \to \infty} \|P^n - \pi\| = 0$$

that is the distribution P^n of X_n converges to a target invariant distribution π irrespective of the initial conditions. A chain is said to be ergodic if it is irreducible, aperiodic and positive recurrent.

Ergodicity

A Markov chain is ergodic if it is:

- Irreducible (every state can be reached from any other state),
- Aperiodic (does not get trapped in cycles),
- Positive recurrent (returns to any state in finite expected time).

An ergodic Markov chain also has a unique stationary distribution because:

- Irreducibility ensures the chain cannot be decomposed into separate, disconnected subsets.
- Positive recurrence guarantees that long-run averages converge to a well-defined limit.
- Aperiodicity prevents oscillations, ensuring convergence to equilibrium.

Stationary distribution

- A Markov chain with transition matrix P will have an equilibrium distribution π if and only if $\pi = \pi P$.
- This chain is a **reversible** chain if and only if $\pi_j p_{ji} = \pi_i p_{ij}$ for all $i \neq j$. (This is also known as the **detailed balance condition**).
- Note that $\pi_j p_{ji} = \pi_i p_{ij}$ implies that $\pi = \pi P$, since $(\pi P)_j = \sum_i \pi_i p_{ij} = \sum_i \pi_j p_{ji} = \pi_j \sum_i p_{ji} = \pi_j$
- In this way, to sample from the limiting distribution π, we run a Markov Chain with transition matrix P satisfying the detailed balance condition until the chain appears to have settle down to equilibrium.

Classical LLN's and CLT's not directly applicable due to:

- Markovian dependence structure between the obs X_i,
- Non-stationarity of the sequence.

Theorem (Ergodic Theorem)

If the Markov chain (X_n) is Ergodic (irreducible, aperiodic and positive recurrent) then for any function h with $E|h| < \infty$,

$$\lim_{n \longrightarrow \infty} \frac{1}{n} \sum h(X_i) = \int h(x)\pi(x)dx$$

If the chain is also reversible $(X_{n+1} | X_{n+2} = x \sim X_{n+1} | X_n = x)$ then a "modified" version of the CLT holds.

The MCMC principle - general idea

- A Markov Chain Monte Carlo (MCMC) method for simulation from a distribution π is any method producing an ergodic Markov chain (X_n) whose stationary distribution is π.
- For an arbitrary starting value $x^{(0)}$, a chain (X_n) is generated using a transition kernel with stationary distribution π, which ensure the convergence in distribution of X_n to a random variable from π.
- In order for the Markov chain to converge to the target (stationary or equilibrium) distribution π, it must be irreducible, aperiodic and positive recurrent
- The main problem is how to construct a suitable chain; we will see some of the most used algorithms (namely, Metropolis-Hastings and Gibbs sampling) and also look into methods to assess whether the chain has reached its stationary distribution.

Metropolis-Hastings Algorithm for Inference on θ

1. Start with $\theta^{(0)}$ (for example, generated from the prior)
2. At iteration t generate $\theta^* \sim q\left(\theta^* \mid \theta^{(t-1)}\right)$ proposal distribution
3. Take

$$\theta^{(t)} = \begin{cases} \theta^* & \text{with probability} \quad \rho\left(\theta^{(t-1)}, \theta^*\right) \\ \theta^{(t-1)} & \text{with probability} \quad 1 - \rho\left(\theta^{(t-1)}, \theta^*\right) \end{cases}$$

where

$$\rho\left(\theta^{(t-1)}, \theta^*\right) = \min\left\{ \frac{\pi\left(\theta^* \mid x\right)}{\pi\left(\theta^{(t-1)} \mid x\right)} \frac{q\left(\theta^{(t-1)} \mid \theta^*\right)}{q\left(\theta^* \mid \theta^{(t-1)}\right)}, 1 \right\}$$

[The update at each iteration is accomplished by (a) sampling $u \sim \text{Uniform}(0;1)$ and (b) setting $\theta^{(t)} = \theta^*$ if $u < \rho\left(\theta^{(t-1)}, \theta^*\right)$ or $\theta^{(t)} = \theta^{(t-1)}$ otherwise]

Remarks

- The ratio $\rho\left(\theta^{(t-1)}, \theta^{*}\right)$ is called the MH **acceptance ratio**.
- A chain constructed via the MH algorithm is Markov, since $\theta^{(t)}$ depends only on $\theta^{(t-1)}$.
- Limit Behavior: With minimal regularity conditions on both π and the proposal q it can be proven that $\pi(\theta \mid x)$ is the limiting (stationary) distribution of the chain $\theta^{(t)}$ produced by the MH algorithm
- If $q(z \mid y) = q(y \mid z)$, i.e., the proposal is symmetric, the method is generally know as **Metropolis** algorithm. In such a case, the acceptance is driven only by the ratio of the two density functions

$$\frac{\pi\left(\theta^{*} \mid x\right)}{\pi\left(\theta^{(t-1)} \mid x\right)}$$

Other remarks

- The MH algorithm depends only on the ratios

$$\pi\left(\theta^* \mid x\right) / \pi\left(\theta^{(t-1)} \mid x\right) \quad \text{and} \quad q\left(\theta^{(t-1)} \mid \theta^*\right) / q\left(\theta^* \mid \theta^{(t-1)}\right)$$

  Hence, it's independent of normalizing constants. (However, those normalizing constants must be, in turn, also independent of the conditioning variables).

- In addition, a good proposal distribution would satisfy the following:
  - it should be easy to sample from it
  - it should be easy to compute the acceptance ratio
  - each proposal should go a reasonable distance in the parameter space, otherwise the random walk moves too slowly
  - the proposals are not rejected too frequently

Integral Approximation

Once the chain has reached convergence to the stationary distribution, π, we can use the realizations of the chain to approximate quantities of interest such as

$$E[h(\theta \mid x)] \approx \frac{1}{n} \sum_{t=1}^{T} h\left(\theta^{(t)}\right) \quad \text{(Ergodic Theorem)}$$

Of course, the approximation depends on whether or not the chain has reached convergence (we'll discuss this point later, for now we just assume that's the case)

Although the MH algorithm is valid for any q satisfying the mild conditions we have seen before, the **choice of the proposal** greatly affect the efficiency of the algorithm. Hence, now we proceed discussing some default choices.

Independent Metropolis Hastings

The IMH considers a proposal q which is **independent** of the current state of the chain. Then, the MH proceeds as follows:

1. Given $\theta^{(t-1)}$,
2. Generate $\theta^* \sim q\left(\theta^*\right)$
3. Take

$$\theta^{(t)} = \begin{cases} \theta^* \text{ with probability } \min\left\{\frac{\pi(\theta^*|x)}{\pi\left(\theta^{(t-1)}|x\right)}\frac{q\left(\theta^{(t-1)}\right)}{q(\theta^*)}, 1\right\} \\ \theta^{(t-1)} \text{ otherwise} \end{cases}$$

- Although the θ 's are generated independently, the resulting sample is not i.i.d. Consider, for example, the probability of acceptance, which depends on $\theta^{(t-1)}$.

Example - Mixture of two normals

Suppose we have observed data $y_1, \ldots, y_n$ from the following mixture distribution

$$\delta N\left(7, 0.5^2\right) + (1-\delta) N\left(10, 0.5^2\right)$$

We may want to obtain inference on the δ. We assume a $U(0,1)$ prior on δ. We implement a MH scheme, with an independent proposal. In particular, we compare the results for 2 different choices:

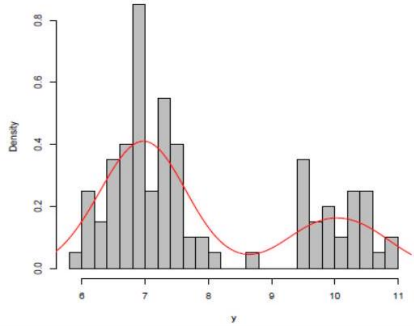
$$\delta \sim \text{Unif}(0,1) \quad \text{and} \quad \delta \sim \text{Beta}(2,10)$$

(see code in R - pay attention on how the MH is implemented)

Data inspection

```
# y is the scanned mixture data
# Plot of the data
hist(y, xlab="y", ylab="Density", main="Mixture data",
        freq=FALSE, breaks=20, col="gray")
lines(density(y, bw=0.39), lwd=2, col="red")
```

Figure: Mixture data



Define density of Normal mixture

```
# density function for the MH ratio
dmixtnorm = function(x, a, m1, m2, s1, s2)
    {
        # x=data; a=alpha;
        # m1=mean of first component; m2=mean of second cc
        # sl=SD of first component; s2=SD of second comp
        a*dnorm(x, m1, s1) + (1-a)*dnorm(x, m2, s2)
    }
m1 = 7; s1 = 0.5; m2 = 10; s2 = 0.5
```

Define density of Normal mixture

```
# case 1: Uniform[0,1]=Beta(1,1) proposal
set.seed(100)
T = 100000 ## number of MCMC iterations
nburn = 50000 ## burn-in period
alpha = rep(NA, T) ## storage for sampled values
alpha[1] = 0.5 ## initial value
accept = 0 ## acceptance count
```

Define density of Normal mixture

```
## Implementation of Metropolis-Hastings algorithm
for(t in 2:T) {
    alpha.prop = runif(1) ## proposed value
## MH acceptance ratio -- work on log-scale for num st
logR = sum(log(dmixtnorm(y, alpha.prop, m1, m2, s1, s2))) -
    sum(log(dmixtnorm(y, alpha[t-1], m1, m2, s1, s2)))
    R = exp(logR)
    u <- runif(1) ## uniform variable to determine the acceptance
    if(u < R) ## accept the new value
        { alpha[t] = alpha.prop
        accept=accept+1 ## update count }
    else{ alpha[t] = alpha[t-1] }
    }
accpt.rate = accept/T
```

Define density of Normal mixture

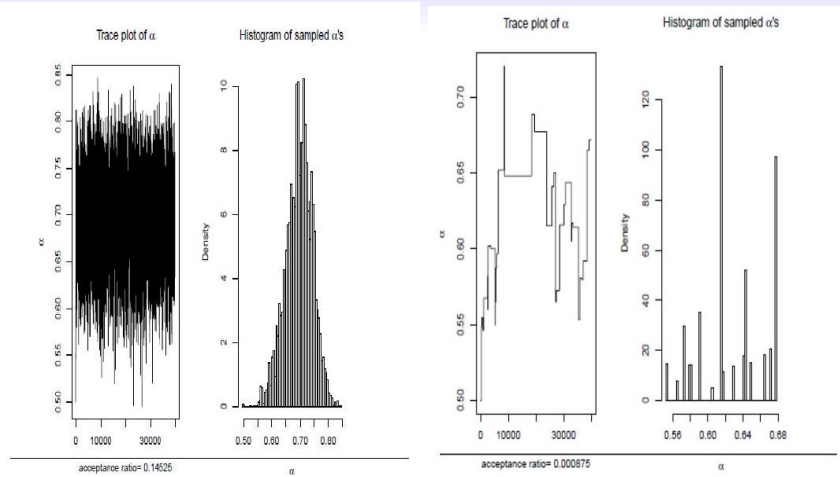


Figure: Left: Uniform(0,1) proposal distribution; Right: Beta(2,10)

Random Walk Metropolis Hastings (RWMH)

- Intuitively, the chain might be more efficient if we consider proposals that take into account the value **previously simulated** to generate the following value; that is, if we consider a local exploration of the neighborhood of the current value of the Markov Chain.
- A first choice is to simulate θ^* as

$$\theta^* = \theta^{(t-1)} + \varepsilon,$$

where $\varepsilon \sim g$ independently of $\theta^{(t-1)}$. Hence, now

$$q\left(\theta^* \mid \theta^{(t-1)}\right) = g\left(\theta^* - \theta^{(t-1)}\right)$$

Choices of g: Most common choices are:

- Uniform distributions on spheres centered at the origin
- Scaled normal distribution
- Scaled Student's t distribution

Lecture 9: Monte Carlo Methods II - Outline

- The Gibbs sampler
- Multi-Stage Gibbs sampler
- SPF Example
- MCMC diagnostic