ST509 Computational Statistics

Lecture 11: Metropolis-Hasting Algorithm

Seung Jun Shin

Department of Statistics
Korea University

E-mail: sjshin@korea.ac.kr

Bayesian Inference I

- Do we really need to compute the expectation in the data analysis?
- Frequentists focus on sample estimates only.
- But Bayesians do not.

Bayesian Inference II

- Probability starts from relative frequency.
- Consider the following scenarios.
  - S1 A musician claim that, given a pair of music sheets, she can identify which one was Haydn or Mozart.
  - S2 In the tea example, she claim that a lady can tell whether the tea or the milk was added first to a cup.
  - S3 One of my friend claim that he can predict whether heads is up or not when flipping a fair coin.
- For all scenarios, they had 8 correct answers out 10 trials!

Bayesian Inference III

- Classical (Frequentist) approach gives an identical estimate of θ as

$$p = \frac{x}{n} = \frac{8}{10} = .8$$

- For (S1), "1" seems okay since a musician is an expert in classic music.
- On the other hand, we still cannot believe "1" for (S3) since no one can predict the random event perfectly.

- Bayesian distinguishes (S1) – (S3)
- Estimator should be based not only on the data but also prior information related to the data.

Bayesian Inference IV

- Bayesian Approach:
  - Prior Distribution: $\pi(\boldsymbol{\theta})$ - prior information about $\boldsymbol{\theta}$
  - Data Distribution or Likelihood: $L(\mathbf{x} \mid \boldsymbol{\theta})$ - probability of obtaining the current data $\mathbf{x}$ at $\boldsymbol{\theta}$.
  - Posterior Distribution: $f(\boldsymbol{\theta} \mid \mathbf{x})$ - updated belief after $\mathbf{x}$ observed.
- Posterior distribution can be computed by Bayes Theorem:

$$f(\boldsymbol{\theta} \mid \mathbf{x}) = \frac{f(\mathbf{x}, \boldsymbol{\theta})}{f(\mathbf{x})} = \frac{L(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int L(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

- We often simply state

$$\underbrace{f(\boldsymbol{\theta} \mid \mathbf{x})}_{\text{Posterior}} \propto \underbrace{L(\mathbf{x} \mid \boldsymbol{\theta})}_{\text{Likelihood}} \times \underbrace{\pi(\boldsymbol{\theta})}_{\text{Prior}}$$

since the denominator is constant in terms of $\boldsymbol{\theta}$.

Bayesian Inference V

- Bayes estimator minimizes the posterior risk.

$$\min_{T(X)} E_{\boldsymbol{\theta}|\mathbf{X}} \{T(\mathbf{X}) - \boldsymbol{\theta}\}^2$$

- The Bayes estimator is

$$E(\boldsymbol{\theta} \mid \mathbf{X}) = \int \boldsymbol{\theta} f(\boldsymbol{\theta} \mid \mathbf{X}) d\boldsymbol{\theta}.$$

- Integration is essential in Bayesian!

Bayesian Inference VI

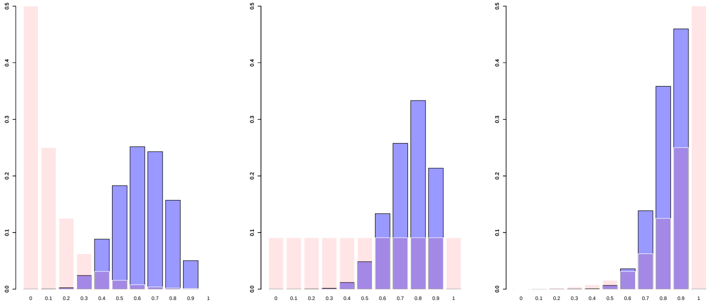
- $X \sim B(10, \theta)$ and we observe $x = 8$ and thus $\hat{\theta} = 0.8$.
- 

Figure: Posterior (blue) distributions, $f(\theta \mid X = 8)$ for three different prior (red) distributions. (decreasing/uniform/ increasing). Bayesian approach yields different estimates for different priors.

Bayesian Inference VII

- (Beta-Binomial) Suppose $X_1, \cdots, X_n \overset{iid}{\sim} \text{Binomial}(k, \theta)$:

$$L(x \mid \theta) = \binom{k}{x} \theta^x (1-\theta)^{k-x}$$

- For Bayesian inference, we assume the following Beta prior on θ:

$$\theta \sim \text{Beta}(\alpha, \beta), \qquad \pi(\theta) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1} (1-\theta)^{\beta-1}$$

- The posterior distribution is

$$\begin{aligned} f(\theta \mid \mathbf{X}) &= C \times \theta^{\sum_{i=1}^n X_i} (1-\theta)^{N - \sum_{i=1}^n X_i} \times \theta^{\alpha-1} (1-\theta)^{\beta-1} \\ &\propto \theta^{\sum_{i=1}^n X_i + \alpha - 1} (1-\theta)^{N - \sum_{i=1}^n X_i + \beta - 1} \end{aligned}$$

where $N = nk$.

$$\theta \mid \mathbf{X} \sim \text{Beta}\left(\sum_{i=1}^n X_i + \alpha, N - \sum_{i=1}^n X_i + \beta \right)$$

Bayesian Inference VIII

- Conjugate pair: A pair of prior and likelihood so that the corresponding posterior is the same distribution as the prior (with different parameters).
- Popular conjugate pairs include
  - Beta – Binomial for θ;
  - Gamma – Poisson for μ;
  - Normal – Normal for μ;
  - Inverse Gamma – Normal for σ^2.

Bayesian Inference IX

- However, in practice, we often encounter a much much more complicating form of posterior!
- Bayesian inference requires to compute

$$E_{\boldsymbol{\theta}|\mathbf{x}}(h(\boldsymbol{\theta})) = \int h(\boldsymbol{\theta}) f(\boldsymbol{\theta} \mid \mathbf{x}) d\theta$$

  where

$$f(\boldsymbol{\theta} \mid \mathbf{x}) = \frac{L(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int L(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}) d\boldsymbol{\theta}}$$

- An analytical computation is often impossible.
- Classical MC approach is also not possible since generating random samples from $f(\boldsymbol{\theta} \mid \mathbf{x})$ is notorious.

Markov Chain Theory I

- A Markov Chain $\{X^{(t)}\}$ is a sequence of dependent random variables

$$X^{(0)}, X^{(1)}, X^{(2)}, \cdots, X^{(t)}, \cdots$$

  such that

$$X^{(t+1)} \mid X^{(t)}, \cdots, X^{(0)} \stackrel{\mathcal{D}}{=} X^{(t+1)} \mid X^{(t)}$$

- This is known as Markov Property.
- The conditional probability of $X^{(t)} \mid X^{(t-1)}$ is called a transition kernel or a Markov kernel:

$$X^{(t+1)} \mid X^{(t)}, \cdots, X^{(0)} \sim K(X^{(t)}, K^{(t+1)})$$

ex. A simple random walk Markov chain satisfies

$$X^{(t+1)} = X^{(t)} + e_t$$

where $e_t \sim N(0, 1)$, independently of $X^{(t)}$.

- Thus $K(X^{(t)}, X^{(t+1)})$ corresponds to $N(X^{(t)}, 1)$.

Markov Chain Theory II

- Suppose there exists a probability distribution f such that if $X^{(t)} \sim f$ then $X^{(t+1)} \sim f$, we call it a stationary distribution.
- The stationary distribution f satisfies

$$\int K(x,y)f(x)dx = f(y)$$

Markov Chain Theory III

- Irreducible: No matter the starting value $X^{(0)}$, the sequence has a positive probability of eventually reaching any states. (i.e., all states communicate)
- Periodic: all states are not periodic.
- Recurrent: The sequence visit any states infinitely many times.
- An irreducible, aperiodic, recurrent (ergodic) Markov chain has a unique stationary distribution, which is also the limiting distribution.

Markov Chain Theory IV

- Suppose a kernel K produces an ergodic Markov chain with stationary distribution f, and $\{X^{(t)}, t = 1, \cdots, T\}$ are a Markov Changes generated from the kernel K. Then $\{X^{(t)}, t = 1, \cdots, T\}$ can be viewed as simulations from f.
- (SLLN for Markov Chain) We have

$$\frac{1}{T}\sum_{t=1}^{T} h(X^{(t)}) \quad \rightarrow \quad E_f[h(X)]$$

which is known as Ergodic Theorem.

Metropolis-Hasting Algorithm I

- Given a target density f, we like to build a Markov Kernel K with stationary distribution f and then generate a Markov chain $\{X^{(t)}\}$ to evaluate the integral via Ergodic Theorem.
- It is unclear how to obtain K for a target density f.
- Miraculously, there exist methods for deriving such kernels!
- Metropolis-Hasting algorithm is one canonical example!

Metropolis-Hasting Algorithm II

- (MH algorithm) Given $x^{(t)}$
  1. Generate $Y_t \sim q(y \mid x^{(t)})$
  2. Take

     $$X^{(t+1)} = \begin{cases} Y_t, & \text{with probability } \rho(x^{(t)}, Y_t) \\ x^(t), & \text{with probability } 1 - \rho(x^{(t)}, Y_t) \end{cases}$$

     where

     $$\rho(x, y) = \min\left\{ \frac{f(y)}{f(x)} \frac{q(x \mid y)}{q(y \mid x)}, 1 \right\}.$$

- The distribution q is called the instrumental/proposal/candidate distribution.
- The probability $\rho(x, y)$ is called the MH acceptance probability.

Metropolis-Hasting Algorithm III

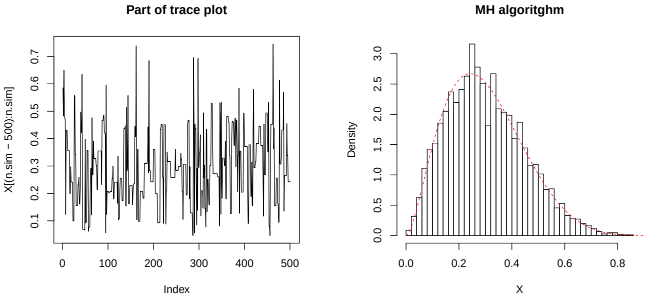
- Acceptance rate is defined by

$$\bar{\rho} = \frac{1}{T} \sum_{t=1}^{T} \rho(X^{(t)}, Y_t).$$

  and used to evaluate the performance of the algorithm.

Metropolis-Hasting Algorithm IV

ex Simulation from $\text{Beta}(2.7, 6.3)$ via MH algorithm:



(a) Trace plot (b) Histogram

Metropolis-Hasting Algorithm V

- MCMC and exact sampling outcomes look identical, but MCMC samples are correlated.
- This means that the quality of the sample is necessarily degraded and thus we need more simulations to achieve the same precision.

Metropolis-Hasting Algorithm VI

- MH algorithm depends only on the ratios

$$f(t_t)/f(x^{(t))} \qquad \text{and} \qquad q(x^{(t)} \mid y_t)/q(y_t \mid x^{(t)})$$

  and thus independent of the normalizing constants!

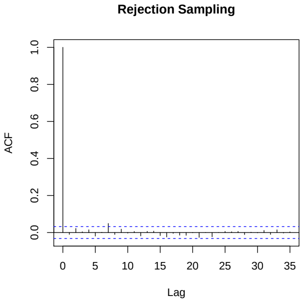
Metropolis-Hasting Algorithm VII

- Independent MH algorithm employs q to be independent of $x^{(t)}$.
  1. Generate $Y_t \sim q(y)$
  2. Take

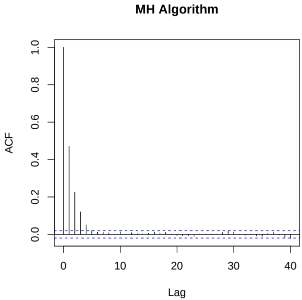
$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability } \min\left\{ \frac{f(Y_t)}{f(x^{(t)})} \frac{q(x^{(t)})}{q(Y_t)}, 1 \right\} \\ x^{(t)} & \text{otherwise} \end{cases}$$

Metropolis-Hasting Algorithm VIII

- Independent MH is similar to the rejection sampling, but keeps the samples when rejected (which makes the samples are correlated).



(c) Rejection Sampling



(d) MH algorithm

Metropolis-Hasting Algorithm IX

- Random Walk MH algorithm:
  1. Generate $Y_t \sim q(y - x^{(t)})$
  2. Take

$$X^{(t+1)} = \begin{cases} Y_t, & \text{with probability } \min\left\{\frac{f(Y_t)}{f(x^{(t)})}, 1\right\} \\ x^{(t)}, & \text{otherwise} \end{cases}$$

- A natural choice is

$$Y_t = x^{(t)} + \epsilon, \qquad \epsilon \sim N(0, \delta^2)$$

Metropolis-Hasting Algorithm X

- (Likelihood) Bayesian Logistic Regression

$$y_i \sim \text{Bernoulli}\{p(\mathbf{x}_i)\},$$

where

$$\log\left\{\frac{p(\mathbf{x}_i)}{1-p(\mathbf{x}_i)}\right\} = \boldsymbol{\beta}^T\mathbf{x}_i$$

- (Prior) Normal prior

$$\boldsymbol{\beta} \sim N_p(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta),$$

with a given set of hyper-parameter $(\boldsymbol{\mu}_\beta, \Sigma_\beta)$.

- Posterior is proportional to

$$f(\boldsymbol{\beta} \mid \text{Data}) \propto \prod_{i=1}^{n} p(\mathbf{x}_i)_i^y\{1-p(\mathbf{x}_i)\}^{1-y_i} \times \exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_\beta)^T\boldsymbol{\Sigma}_\beta^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_\beta)\right\}$$

Metropolis-Hasting Algorithm XI

```
mcmc.logit <- function(x, y, init, n.sample = 10000, step = 0.3){
  post.beta <- matrix(0, n.sample, p)
  ac.ratio <- rep(0, n.sample)

  prior.m <- 10
  prior.s <- 1000 # for vague prior

  # intialize
  post.beta[1,] <- beta <- init
  eta <- x %*% beta
  pi <- exp(eta)/(1 + exp(eta))

  log.prior <- sum(dnorm(beta, prior.m, prior.s, log = T))
  log.like  <- sum(y * log(pi) +  (1 - y) * log(1 - pi))

  iter <- 2
  for (iter in 1:n.sample){

    # candidate
    beta.new <- beta + rnorm(p, 0, step)
    eta.new <- x %*% beta.new
    pi.new  <- exp(eta.new)/(1 + exp(eta.new))
```

Metropolis-Hasting Algorithm XII

```
      # prior
      log.prior.new <- sum(dnorm(beta.new, prior.m, prior.s, log = T))

      # liklihood
      log.like.new <- sum(y * log(pi.new) + (1 - y) * log(1 - pi.new))

      # ratio
      temp <- exp((log.like.new + log.prior.new) - (log.like + log.prior))
      rho <- min(1, temp)

      if (runif(1) < rho) {
        ac.ratio[iter] <- 1
        beta <- beta.new
        log.prior <- log.prior.new
        log.like  <- log.like.new
        eta <- x %*% beta
        pi <- exp(eta)/(1 + exp(eta))
      }
      post.beta[iter,] <- beta
    }

    obj <- list(posterior = post.beta, acpt.ratio = mean(ac.ratio))
    return(obj)
  }
```

Metropolis-Hasting Algorithm XIII