Parameter Estimation for Tweedie Distribution with MCMC Approach

Topic Presentation

Tairan Ye

4/12/2018

University of Connecticut

Summary

1. Markov Chain Monte Carlo
2. Tweedie Distribution
3. Parameter Estimation for Tweedie Model with MCMC Approach

Markov Chain Monte Carlo

Introduction

- Fact:
  Your desired parameters are not in the analytical form or extremely hard to solve out.
- Solution:
  The sampling algorithms, like Markov Chain Monte Carlo
- Reason:
  By constructing a Markov chain, we can sample the desired distribution by observing the chain after a number of iterations. The sample distribution will match more closely to the actual desired distribution as the iterations increase.

Example: Monte Carlo Integration

General problem:
Evaluating $\mathbf{E}[h(X)] = \int h(x)\pi(x)dx$ can be difficult.
However, if we can draw samples:

$$X^{(1)}, X^{(2)}, ..., X^{(N)} \sim \pi(x) \tag{1}$$

Then, we can estimate:

$$\mathbf{E}[h(X)] \approx \frac{1}{N} \sum_{t=1}^{N} h(X^{(t)}) \tag{2}$$

This is Monte Carlo (MC) Integration.

Markov Chain

A Markov chain is generated by sampling:

$$X^{(t+1)} \sim p(x|x^{(t)}), t = 1, 2, ... \tag{3}$$

Here p is the transition kernel.
So $X^{(t+1)}$ depends only on $X^{(t)}$, rather than $X^{(0)}$, $X^{(1)}$, $X^{(2)}$, ..., $X^{(t-1)}$
We can summarize as following:

$$p(X^{(t+1)}|X^{(t)}, X^{(t-1)}, ..., X^{(0)}) = p(X^{(t+1)}|X^{(t)}) \tag{4}$$

An example of Markov Chain

A first order auto-regressive process with lag-1 auto-correlation 0.5.

$$X^{(t+1)}|X^{(t)} \sim N(0.5x^{(t)}, 1) \tag{5}$$

A simulation is conducted as following, which contains two different starting points:

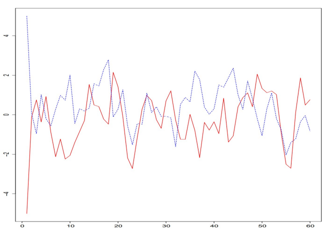

Figure 1: Simulation with two starting points

The chains seemed to have forgotten their starting positions.

Sampling Algorithm Overview

The most fundamental MCMC sampling algorithm are:

- Gibbs Sampling
    1. Require for the analytic form of the full conditional distribution.
    2. High-efficiency – the sequence will converge with little iterations.
- Metropolis-Hasting Algorithm
    1. More flexible – no need for the deduction of full conditional distribution.
    2. Some challenges for setting up the proposal distribution – experience needed to control the accept rate between 20% to 30%.

Full Conditional Distribution and Gibbs Sampling

- Full conditional distributions:
  The distributions $p(\theta|\sigma^2, y_1, ..., y_n)$ and $p(\sigma^2|\theta, y_1, ..., y_n)$ are called the full conditional distributions of σ^2 and θ respectively, as they are each a conditional distribution of a parameter given everything else.
- Gibbs Sampler:
  Given a current state of the parameters $\phi^{(s)} = \{\theta^{(s)}, \sigma^{2(s)}\}$, we generate a new state as follows:

  1. sample $\theta^{(s+1)} \sim p(\theta|\sigma^{2(s)}, y_1, ..., y_n)$;
  2. sample $\sigma^{2(s+1)} \sim p(\sigma^2|\theta^{(s+1)}, y_1, ..., y_n)$;
  3. update $\phi^{(s+1)} = \{\theta^{(s+1)}, \sigma^{2(s+1)}\}$

Generalized Gibbs Sampler

Suppose you have a vector of parameters $\phi = \{\phi_1, ..., \phi_p\}$, and your information about ϕ is measured with $p(\phi) = p(\phi_1, ..., \phi_p)$. For example, in the normal model $\phi = \{\theta, \sigma^2\}$, and the probability measure of interest is $p(\theta, \sigma^2 | y_1, ..., y_n)$. Given a starting point $\phi^{(0)} = \{\phi_1^{(0)}, ..., \phi_p^{(0)}\}$, the Gibbs sampler generates $\phi^{(s)}$ from $\phi^{(s-1)}$ as follows:

- sample $\phi_1^{(s)} \sim p(\phi_1 | \phi_2^{(s-1)}, \phi_3^{(s-1)}, ..., \phi_p^{(s-1)})$
- sample $\phi_2^{(s)} \sim p(\phi_2 | \phi_1^{(s)}, \phi_3^{(s-1)}, ..., \phi_p^{(s-1)})$

  ......
- sample $\phi_p^{(s)} \sim p(\phi_p | \phi_1^{(s)}, \phi_2^{(s)}, ..., \phi_{p-1}^{(s)})$

Markov Property: In this sequence, $\phi^{(s)}$ depends on $\phi^{(0)}, ..., \phi^{(s-1)}$ only through $\phi^{(s-1)}$, i.e. $\phi^{(s)}$ is conditionally independent of $\phi^{(0)}, ..., \phi^{(s-2)}$ given $\phi^{(s-1)}$.

Metropolis Algorithm

The Metropolis algorithm proceeds by sampling a proposal value θ^* nearby the current value $\theta^{(s)}$ using a symmetric proposal distribution $J(\theta^*|\theta^{(s)})$.
Usually $J(\theta^*|\theta^{(s)})$ is very simple, with samples from $J(\theta^*|\theta^{(s)})$ being near $\theta^{(s)}$ with high probability. Examples include:

- $J(\theta^*|\theta^{(s)}) \sim \mathbf{U}(\theta^{(s)} - \delta, \theta^{(s)} + \delta)$
- $J(\theta^*|\theta^{(s)}) \sim \mathbf{N}(\theta^{(s)}, \delta^2)$

Metropolis Algorithm

The algorithm is as following:

1. Sample $\theta^* \sim J(\theta^* | \theta^{(s)})$
2. Compute the acceptance ratio:

$$r = \frac{p(\theta^* | y)}{p(\theta^{(s)} | y)} = \frac{p(y | \theta^*) p(\theta^*)}{p(y | \theta^{(s)}) p(\theta^{(s)})} \tag{6}$$

3. Update

$$\theta^{(s+1)} = \begin{cases} \theta^*, p = min(r, 1) \\ \theta^{(s)}, p = 1 - min(r, 1) \end{cases} \tag{7}$$

- Step 3 can be accomplished by sampling $u \sim \mathbb{U}(0, 1)$ and setting $\theta^{(s+1)} = \theta^*$ if $u < r$ and setting $\theta^{(s+1)} = \theta^{(s)}$ otherwise.
- In many cases, computing the ratio r directly can be numerically unstable, a problem that often can be remedied by computing the logarithm of r:

Metropolis Hastings Algorithm

Metropolis Hastings Algorithm is the generalized and multi-parameters version of Gibbs sampling and Metropolis algorithm. Let's consider a simple example where our target probability distribution is $p_0(u, v)$, a bivariate distribution for two random variables U and V.

1. Update U:
   - Sample $u^* \sim J_u(u^*|u^{(s)}, v^{(s)})$
   - Compute the acceptance ratio:

$$r = \frac{p_0(u^*, v^{(s)})}{p_0(u^{(s)}, v^{(s)})} * \frac{J_u(u^{(s)}|u^*, v^{(s)})}{J_u(u^*|u^{(s)}, v^{(s)}} \tag{8}$$

   - Update

$$u^{(s+1)} = \begin{cases} u^*, p = min(r, 1) \\ u^{(s)}, p = 1 - min(r, 1) \end{cases} \tag{9}$$

   - Sampling $u \sim \mathbb{U}(0, 1)$ and setting $u^{(s+1)} = u^*$ if $u < r$ and setting $u^{(s+1)} = u^{(s)}$ otherwise.

Metropolis Hastings Algorithm

After updating U, we continue to update V:

1. Update V:
   - Sample $v^* \sim J_v(v^*|u^{(s+1)}, v^{(s)})$
   - Compute the acceptance ratio:

$$r = \frac{p_0(u^{(s+1)}, v^*)}{p_0(u^{(s+1)}, v^{(s)})} * \frac{J_v(v^{(s)}|u^{(s+1)}, v^*)}{J_v(v^*|u^{(s+1)}, v^{(s)})} \tag{10}$$

   - Update

$$v^{(s+1)} = \begin{cases} v^*, p = min(r, 1) \\ v^{(s)}, p = 1 - min(r, 1) \end{cases} \tag{11}$$

   - Sampling $u \sim \mathbb{U}(0, 1)$ and setting $v^{(s+1)} = v^*$ if $u < r$ and setting $v^{(s+1)} = v^{(s)}$ otherwise.

Proposal Distribution

The proposal distribution is the most "challenging" part for conducting the Metropolis algorithm.

- Without restriction:
  Normal distribution: $\theta^* \sim \mathbf{N}(\theta_{t-1}, \sigma_p^2)$
- With restriction:
  Random walk: $f(\theta^*) = f(\theta_{t-1}) + \epsilon$ and $\epsilon \sim \mathbf{N}(0, 1)$
  Example:
  $\theta > 0$, then apply *log* function.
  θ^* can be generated from *lognormal*$(log(\theta_{t-1}), \sigma_p^2)$

Combination

In complex models, it is often the case that conditional distributions are available for some parameters but not for others. In these situations we can combine Gibbs and Metropolis-type proposal distributions to generate a Markov chain to approximate the joint distribution of all of the parameters.

More to talk about

1. Stationary:

   As $t \rightarrow \infty$, the Markov chain converges to its stationary distribution.

   Technique: trace plot, effective sample size (ESS)

   Package: 'coda' in R

2. Burn-in:

   Typically, we burn-in the initial simulation values since they are most likely nonstationary, and study the remaining part.

3. Gap:

   In order to avoid the dependency, we need to pick up the simulation values by gaps.

4. Chains:

   Also, we will build more than 1 chain to make sure the sampler is representative for the distribution. By the way, the calculation of deviance information criterion (DIC) requires at least 2 chains.

Example

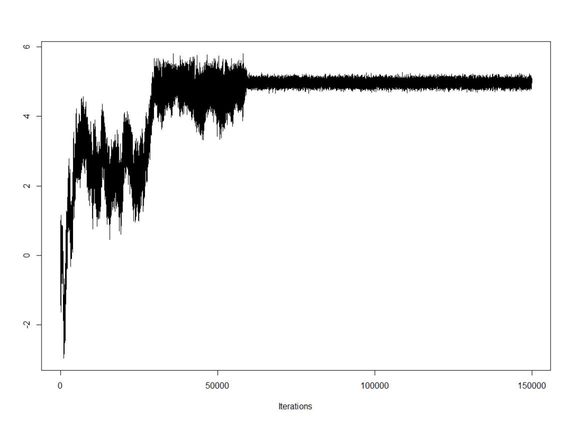

Figure 2: The trace plot of the raw MCMC

Example

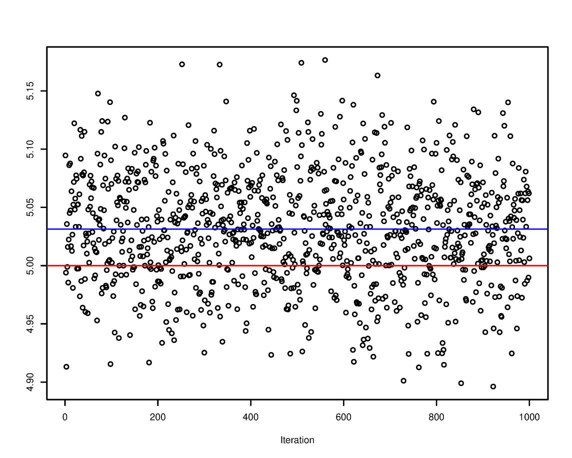

Figure 3: The trace plot of the updated sequence

Software

- WinBUGS (Bayesian inference Using Gibbs Sampling)
- JAGS (Just Another Gibbs Sampler)
- Stan – with linkage of most of data science softwares via multiply platforms
- R packages: rjags, coda, mcmc, BRugs, etc.

Tweedie Distribution

Tweedie Distribution Overview

Tweedie distribution can be defined as $Tw(\mu, \phi, p)$.

- μ is the expected mean for the Tweedie distribution.
- ϕ is called the dispersion parameter.
- p, or power, controls what distribution it belongs to.

Property:

- A special case of exponential dispersion models.
- $Variance = \phi\mu^p$.
- Most interested in $p \in (1, 2)$, which is defined as compound Poisson-gamma distribution.

Compound Poisson-gamma distribution

Due to its special property – mass at zero and continue otherwise, the Tweedie model with $p \in (1,2)$, or the compound Poisson-gamma distribution, is popular in insurance industry.

Denote the total auto insurance claim in dollar is Y, the frequency of claim is N, and the severity of each claim is Z. Then, we have:

$$Y = \sum_{n=1}^{N} Z_n \tag{12}$$

$$N \sim Poisson(\lambda) \tag{13}$$

$$Z \sim gamma(\alpha, \beta) \tag{14}$$

$$N \perp\!\!\!\perp Z \tag{15}$$

Compound Poisson-gamma distribution

We say $Y \sim Tw(\mu, \phi, p)$. Also:

$$\lambda\alpha\beta = \mu \tag{16}$$

$$\lambda\alpha(\alpha+1)\beta^2 = \phi\mu^p \tag{17}$$

$$\tag{18}$$

We can solve:

$$\lambda = \frac{\mu^{2-p}}{\phi(2-p)} \tag{19}$$

$$\alpha = \frac{2-p}{p-1} \tag{20}$$

$$\beta = \phi(p-1)\mu^{p-1} \tag{21}$$

Compound Poisson-gamma distribution

The probability function for the Tweedie distribution with a power $p \in (1, 2)$ is:

1. For $y = 0$:

$$\begin{aligned} \mathbb{P}[Y = 0] &= \exp\{-\lambda\} \\ &= \exp\left\{-\frac{\mu^{2-p}}{\phi(2-p)}\right\} \end{aligned} \tag{22}$$

2. For $y > 0$:

$$\begin{aligned} f_Y(y) &= e^{-\frac{y}{\beta}} e^{-\lambda} \sum_{n=1}^{\infty} \frac{1}{\Gamma(n\alpha)\beta^{n\alpha}} y^{n\alpha - 1} \frac{\lambda^n}{n!} \\ &= \exp\left\{-\frac{y}{\phi(p-1)\mu^{(p-1)}} - \frac{\mu^{(2-p)}}{\phi(2-p)}\right\} \frac{1}{y} \sum_{j=1}^{\infty} \frac{\left\{\frac{1}{2-p}\left(\frac{y}{\phi(p-1)}\right)^{\frac{2-p}{p-1}}\right\}^j}{\phi^j \Gamma\left(\frac{(2-p)j}{p-1}\right) j!} \end{aligned} \tag{23}$$

Parameter Estimation for Tweedie Model with MCMC Approach

Model

In order to study the key factors that affects the auto insurance claims, we can set up:

$$log(\mu_s) = \mathbf{X_s}\gamma + \epsilon_{random} \tag{24}$$

$$\epsilon_{random} \sim \mathbf{N}(0, \sigma^2) \tag{25}$$

where s indicates for different location; $\mathbf{X_s}$ contains the information about policyholders, including personal information as well as vehicle information; γ is the factors, which we are interested in; ϵ covers all other information we don't discuss here.
Therefore, the parameters we are going to analyze are: μ_s, γ, σ^2, ϕ, and p.

Why MCMC Approach

1. Advantage:
   - No need to involve huge amount of mathematical computation
   - Provide a good estimate for ϕ and p
2. Disadvantage:
   - Sampling algorithm, especially Metropolis Hasting, consumers great energy
   - In order to achieve the optimal accept rate, the proposal distribution adjustment sometimes is annoying

Model

Apply the MCMC to estimate these parameters:

```
for $m$ in $1 : M$ do
    Update $\gamma$, $\sigma^2$, $\phi$, and $p$
    for $s$ in $1 : S$ do
        Update $\mu_s$
    end
end
```

Simulation Study - Univariate Case

Suppose we are studying the auto insurance claims in two different locations, each with 200 customers respectively. Let's begin with univariate case, that is only one factor we need to consider for the auto insurance claims.

Parameter	True Value	Our Estimate
γ	5	5.03
σ^2	1	1.03
ϕ	1	0.98
p	1.5	1.48

Table 1: Summary of the simulation study

Simulation Study - Univariate Case

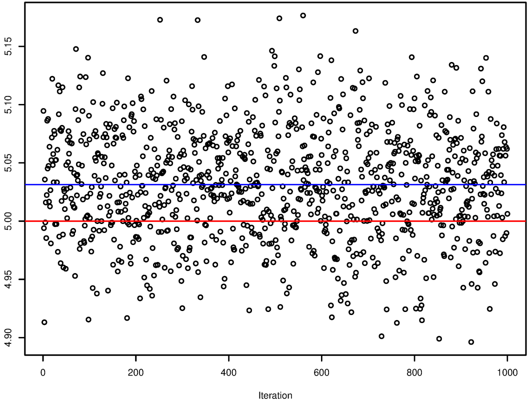

Figure 4: The MCMC result for the γ

Simulation Study- Univariate Case

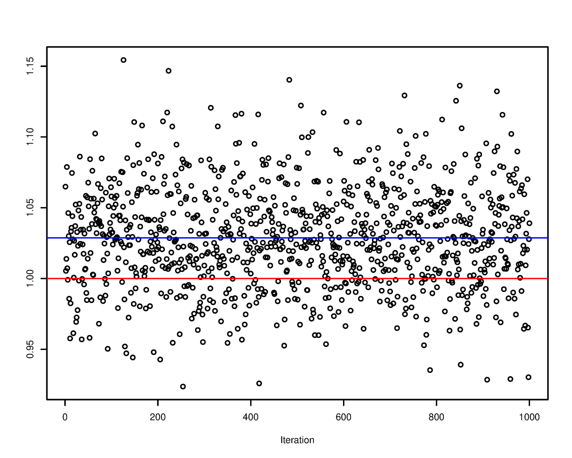

Figure 5: The MCMC result for the σ^2