Report on 'Laplace Expansions in Markov Chain Monte Carlo Algorithm'

ZiXia Huang

2024-02-29

1. Main Research Problem

Complex hierarchical models lead to a complicated likelihood and then, in a Bayesian analysis, to complicated posterior distributions. To obtain Bayes estimates such as the posterior mean or Bayesian confidence regions, in most of the cases it is necessary to simulate the posterior distribution using a method such as an MCMC algorithm. These algorithms often becomes time-consuming as the number of observations increases, especially when the latent variables are considered.

The main research topic of the paper is to improve the convergence of the MCMC algorithm when having a hierarchical model with a large number of parameters. The researcher propose to decrease the number of parameters to simulate at each iteration to improve the convergence of the most commonly used Gibbs sampler by using a Laplace approximation on the nuisance parameters.

The paper also provides a theoretical study of the impact that such an approximation has on the target posterior distribution. The paper shows that the stationary distribution of the Markov chain generated by such improved algorithm based on Laplace approximation gets close to the true posterior distribution as the numbers of observation N goes to infinity.

Finally the paper provides a simulation study on comparing the converge rate and global computational time between the traditional Gibbs algorithm and the Laplace algorithm. It turns out that this proposed Laplace algorithm converges to the stationary distribution more rapidly and has a much more shorter computational time.

Therefore, this algorithm could be used in many applied studies where the large computation time is a real problem, as an improvement of the classical Gibbs algorithm in MCMC method.

2. Classical Gibbs algorithm and Proposed Laplace algorithm

2.1 The Classical Gibbs algorithm

Suppose we have a hierarchical model with the following joint probability density:

$$f(X|\theta_S, S)\pi(\theta_S|S)p(S|\lambda)h(\lambda)$$

Where X is the vector of observations, $S \in \mathbf{S}$ is a vector of parameters,typically latent variables or at least high dimensional, $\lambda \in L$ is the parameter associated with S, and $\theta_S \in \Theta$ is the nuisance parameter.Here we consider (λ, S) as the parameter of interest, since the marginal distributions of λ and S can be obtained from the joint distribution.

A Classical Gibbs algorithm would run a chain on (λ, S, θ_S) to obtain a sample from the posterior distribution of (λ, S) in the following way:

1.$\lambda^t \sim h(\lambda|S^{t-1})$,

2.$S^t \sim p(S|X, \lambda^t, \theta_S^{t-1})$,

3.$\theta_S^t \sim \pi(\theta_S|X, S^t, \lambda^t)$.

As the chain converges eventually, we get the stationary distribution of the estimation of the parameters of interests.

2.2 The Proposed Laplace algorithm

Instead of running a chain on (λ, S, θ_S), we can run a chain only on the parameters of interests (λ, S) by integrating out the nuisance parameter θ_S, using:

$$g(X|S) = \int_\theta f(X|\theta_S, S)\pi(\theta_S|S)d\theta_S$$

However in most of the cases this integration cannot be obtained analytically but can be approximated by using a Laplace expansion. Now the new sampling scheme, named as "Proposed Laplace algorithm", is then:

1.$\lambda^t \sim h(\lambda|S^{t-1})$,

2.$S^t \sim \hat{p}(S|X, \lambda^t) \propto \hat{g}(X|S)p(S|\lambda^t)$,

where $\hat{g}(X|S)$ denotes the Laplace approximation of $g(X|S)$.

3. A simplied model for the proposed Laplace algorithm

3.1 Basic idea of the simplified model

In the paper, researches have considered the complete model of the general case which covers situations that can be very different, for instance latent variable models in finite state spaces with discrete or continuous latent variables or curve estimation via free knot splines. Here, in this report we will simplified the model by assuming we know all the prior distribution of the parameters, and the nuisance parameter θ_S has only one dimension.

Then in the following section, we derive an analytical form of the Laplace approximation for the posterior distribution. And discuss a way to find the local maximizer of the negative joint log-likelihood function.

3.2 Approximate the posterior using Laplace expansion

The basic idea in Laplace approximation is to use Taylor expansion to obtain an approximate integral that admits a closed form solution. Here we consider the second-order Taylor expansion of $h(x)$ around the point $\hat{x}$ is:

$$h(x) \approx h(\hat{x}) + h^{(1)}(\hat{x})(x - \hat{x}) + \frac{1}{2}h^{(2)}(\hat{x})(x - \hat{x})^2,$$

where $h^{(n)}(\hat{x})$ indicates the n^{th} derivative of h, evaluated at $\hat{x}$.

In order to guarantee the Laplace approximation has a good quality in a region containing a significant amount of mass contributing to the integral, the choice of $\hat{x}$ is a local maximum of $-h(x)$. Then we have $h^{(1)}(\hat{x}) = 0$.

Then our approximate integral is:

$$\begin{aligned}\int exp(-h(x))dx &\approx \int exp\{-(h(\hat{x}) + \frac{1}{2}h^{(2)}(\hat{x})(x-\hat{x})^2)\}dx \\ &= exp(-h(\hat{x})) \int exp\{-\frac{(x-\hat{x})^2}{2h^{(2)}(\hat{x})^{-1}}\}dx \\ &\approx exp(-h(\hat{x})) \times (\frac{2\pi}{h^{(2)}(\hat{x})})^{1/2}\end{aligned}$$

Let's keep the basic setting of the parameters of interests and nuisance parameters in section 2. X is the vector of observations, $S \in \mathbf{S}$ is a vector of parameters, $\lambda \in L$ is the parameter associated with S, (λ, S) are the parameters of interests, $\theta_S \in \Theta$ is the nuisance parameter.

Now we use the Laplace approximation mentioned above to derive an analytical solution to the marginal distribution when we trying to integrate the nuisance parameter θ_S out:

$$g(X|S) = \int_\theta f(X|\theta_S, S)\pi(\theta_S|S)d\theta_S$$

Denote $\hat{\theta}_s$ as the properly chosen local maximizer of the joint negative log-likelihood function, with J equals to the second order derivative of the joint-likelihood function evaluated at $\hat{\theta}_S$, also denote the negative joint log-likelihood $-\sum_{i=1}^N logf(X_i|\theta_S, S)$ as $l_N(\theta_S)$.Using the Laplace approximation:

$$\begin{aligned}g(X|S) = \int_\theta \prod_{i=1}^N f(X_i|\theta_S, S)\pi(\theta_S|S)d\theta_S &= \int_\theta exp\{-l_N(\theta_S)\}\pi(\theta_S|S)d\theta_S \\ &\approx \pi(\hat{\theta}_S) \int_\theta exp\{-l_N(\hat{\theta}_S) - \frac{(\theta_S - \hat{\theta_S})^2}{2J^{-1}}\}d\theta_S \\ \approx \pi(\hat{\theta}_S)exp\{-l_N(\hat{\theta}_S)\} \times (2\pi)^{d/2}J^{-1/2} = (2\pi)^{d/2}J^{-1/2}\pi(\hat{\theta}_S)\prod_{i=1}^N f(X_i|\hat{\theta}_S)\end{aligned}$$

Where d is the dimension of θ_S, here we consider the simplest case where $d = 1$.Therefore, $\hat{g}(X|S) = (2\pi)^{1/2}J^{-1/2}\pi(\hat{\theta}_S)\prod_{i=1}^N f(X_i|\hat{\theta}_S)$, now we have as the limiting target distribution: $\hat{\pi}(\lambda, S|X) \propto \hat{g}(X|S)p(S|\lambda)h(\lambda)$.The simplified proposed Laplace algorithm has thus the following structure: at the tth iteration of the Gibbs sampler,

1.$\lambda^t \sim h(\lambda|X, S^{t-1})$,

2.$S^t \sim \hat{p}(S|X, \lambda^t) \propto \hat{g}(X|S)p(S|\lambda^t)$.

3.3 Finding the local maximizer by Newton's method

To use the simplified model of the proposed Laplace algorithm, we have to expand the negative log-likelihood centered at the local maximizer of θ_S, where in particular we have the first order derivative equals to 0.

We will then implement Newton's method, a second-order iterative algorithm that converges quickly to a local mode if initialized in a close enough region. Given an initial estimate of the parameter $\theta_S^{(0)}$, the algorithm proceeds by iterative updating

$$\theta_S^{(t+1)} \leftarrow \theta_S^{(t)} - [h^{(2)}(\theta_S^{(t)})]^{-1} \times h^{(1)}(\theta_S^{(t)})$$

Therefore, our simplified model mainly contains two parts: First we use Newton's method to approximate a local maximizer $\hat{\theta}_S$, Secondly we using the Laplace expansion to approximate the posterior distribution $p(S|X, \lambda^t)$ in Gibbs sampler, and estimate the parameter of interest S.

4. Code the simplified Laplace algorithm

```
library(numDeriv)
library(fields)
library(geoR)
library(splancs)
library(ggplot2)
```

4.1 Derive the posterior disribution

Now we focus on the hierarchical model with the following joint probability density:

$$f(X|\theta_S, S)\pi(\theta_S|S)p(S|\lambda)h(\lambda)$$

Suppose we know that $f(X|\theta_S, S)$ is a normal distribution with $N(\theta_S, S)$; $\pi(\theta_S|S)$ is a exponential distribution $exp(S)$; $p(S|\lambda = \{\alpha, \beta\})$ is a gamma distribution $gamma(\alpha, \beta)$ and finally $\alpha \sim unif(0, S)$, $\beta \sim unif(0, S)$.

Therefore at tth iteration we have:

$\lambda^t \sim unif(0, S^{(t-1)})$

$$p(S|X, \lambda^t, \theta_S^{t-1}) \propto f(X|\theta_S, S)\pi(\theta_S|S)p(S|\lambda) \propto (S^{(t-1)})^{\alpha^t - 1 - n/2} e^{-\frac{\sum (X_i - \theta_S^{(t-1)})^2}{2S^{(t-1)}}} e^{-\beta^t S^{(t-1)}}$$

$$\pi(\theta_S|X, S^t, \lambda^t) \propto f(X|\theta_S, S)\pi(\theta_S|S) \propto (S^t)^{1-n/2} e^{-\frac{\sum (X_i - \theta_S^{(t-1)})^2}{2S^t}} e^{-\theta^{(t-1)} S^t}$$

4.2 Code for Classical Gibbs Algorithm

For the traditional Gibbs sampling algorithm, the last two posterior distribution is rather complicated so we use Metropolis-Hastings algorithm with log-normal and normal proposal to generate samples.

```
#Traditional Gibbs algorithm
N <- 40000
theta_S <- vector(length = N)
S <- vector(length = N)
alpha <- vector(length = N)
beta <- vector(length = N)
M_0 <- function(theta_S_0, S_0, alpha_0, beta_0,n,x){
i <- 1
alpha[1] <- alpha_0
beta[1] <- beta_0
theta_S[1] <- theta_S_0
S[1] <- S_0
while(i < N+1){
   #generate lambda with its posterior distribution
  alpha[i+1] <- runif(1,min = 0, max = S[i])
```

```
  beta[i+1] <- runif(1,min = 0, max = S[i])
  #generate S by Metropolis Hastings
  candidate <- rlnorm(1,meanlog = 2,sdlog = 1)
  u <- runif(1,min=0,max=1)
  test <- n/2*log(S[i]/candidate)+alpha[i+1]*log(candidate/S[i])+(beta[i+1]+theta_S[i])*(S[i]-candidate)
  p=min(1,exp(test))
  if(u <= p){
    S[i+1] <- candidate
    }else{
    S[i+1] <- S[i]
  }
#generate theta_S by Metropolis Hastings
 candidate <- rnorm(1,mean=0,sd=1)
  u <- runif(1,min=0,max=1)
  test <- (theta_S[i]-candidate)*S[i+1]+(sum(x)*(candidate-theta_S[i])-candidate^2+theta_S[i]^2)/2*S[i+1]
  p=min(1,exp(test))
  if(u <= p){
    theta_S[i+1] <- candidate
    }else{
    theta_S[i+1] <- theta_S[i]
  }
  i=i+1
}
return(S)
}
```

4.3 Code for Simplified Laplace Algorithm

For the simplified Laplace algorithm, we first find the first and second order derivative of $L = f(X|\theta_S, S)\pi(\theta_S|S) \propto e^{-\frac{\sum(X_i-\theta_S)^2}{2S}}e^{-S\theta_S}$.

$\frac{dL}{d\theta_S} = (\frac{\sum X_i - n\theta_S}{S} - S)(e^{-\frac{\sum(X_i-\theta_S)^2}{2S}-S\theta_S})$,

$\frac{d^2L}{d^2\theta_S} = [(\frac{\sum X_i - n\theta_S}{S} - S)^2 - \frac{n}{S}](e^{-\frac{\sum(X_i-\theta_S)^2}{2S}-S\theta_S})$

Therefore, the iterative update via Newton's method is $\theta_S^{(t+1)} \leftarrow \theta_S^{(t)} - \frac{\frac{\sum X_i - n\theta_S^{(t)}}{S} - S}{(\frac{\sum X_i - n\theta_S^{(t)}}{S} - S)^2 - \frac{n}{S}}$

```
#Newton's method with 20 iteration cycles
newton <- function(init,S,x,n,maxiter=20){
  par <- init
  for(i in 1:maxiter){
    par <- par- ((sum(x)-n*par)/S-S)/(((sum(x)-n*par)/S-S)^2-n/S)
  }
  return(par)
}
```

Now we have the Laplace approximation of the posterior distribution:

$S^t \sim \hat{p}(S|X, \lambda^t) \propto J^{-1/2}S^{\alpha^t - \frac{n}{2}}e^{-(\beta^t+\hat{\theta}_S)S}e^{-\frac{\sum(X_i-\hat{\theta}_S)^2}{2S}}$

The complexity reduced comparing to the original posterior distribution, but it still does not have an analytical form, so we are using Metropolis-Hastings algorithm with log-normal proposal to generate samples from it.

```
#Simplified Laplace algorithm

M_L <- function(theta_S_0, S_0, alpha_0, beta_0,n,x){
i <- 1
alpha[1] <- alpha_0
beta[1] <- beta_0
S[1] <- S_0
#First find a local maximizer using Newton's method
hat_theta <- newton(theta_S_0,S_0,x,n)
#Second carry out the Gibbs sampler
while(i < N+1){
  #generate lambda with its posterior distribution
  alpha[i+1] <- runif(1,min = 0, max = S[i])
  beta[i+1] <- runif(1,min = 0, max = S[i])
  #generate S by Metropolis Hastings
  candidate <- rlnorm(1,meanlog = 2,sdlog=1)
  u <- runif(1,min=0,max=1)
  test <- (alpha[i+1]-n/2)*log(candidate/S[i])+(beta[i+1]+hat_theta)*(S[i]-candidate)+(sum((x-hat_theta
  p=min(1,exp(test))
  if(u <= p){
    S[i+1] <- candidate
    }else{
    S[i+1] <- S[i]
    }
  i <- i+1
}
return(S)
}
```

5. Simulation Study

Now we carry out a simulation study with the posterior distribution we described in section 3, with the function we derived for the Classical Gibbs algorithm and Simplified Laplace algorithm.

Suppose the true value of the parameters of interest $S = 4$, and the true value of the nuisance parameter $\theta_S = 8$.

```
#Simulate 100 observed data X
set.seed(12)
x <- rnorm(100, mean = 8, sd=2)
```

Perform the two algorithm to estimate the parameters with initial value:

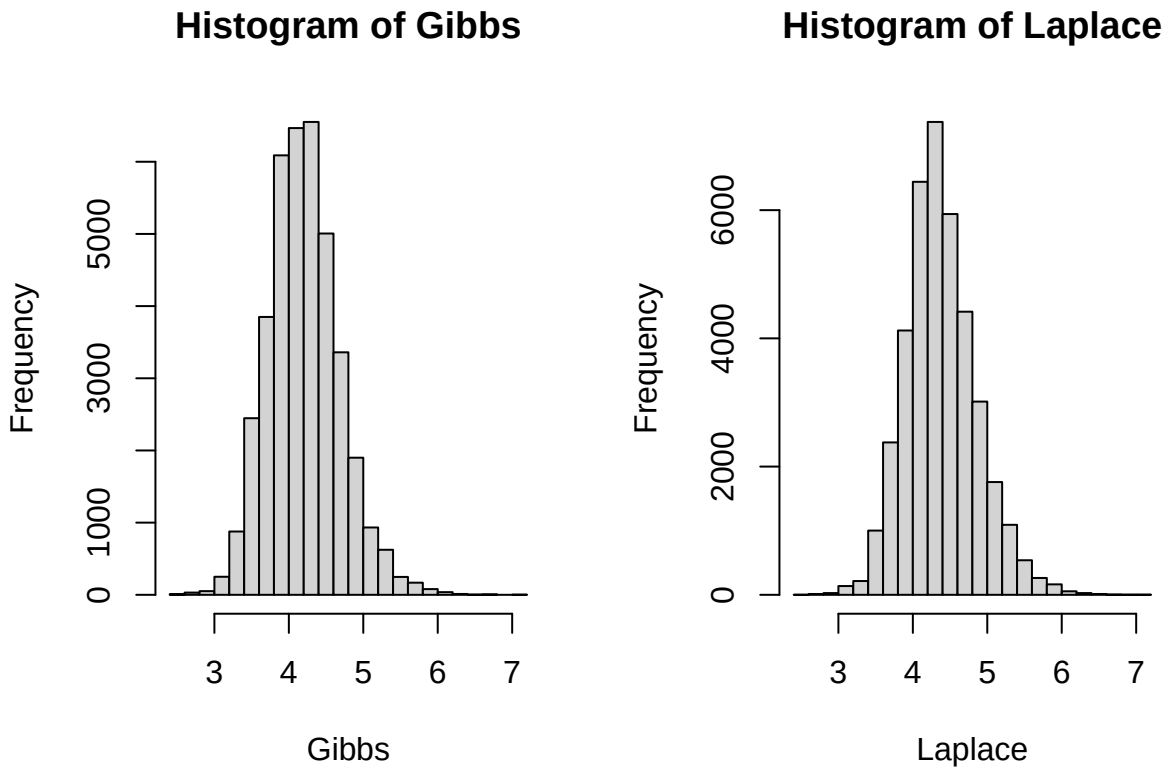
$$\lambda^{(0)} = \{\alpha^{(0)}, \beta^{(0)}\} = \{0.5, 0.5\},$$
$$S^{(0)} = 1,$$
$$\theta_S^{(0)} = 6$$

First we use the Classical Gibbs algorithm:

```
Gibbs <- M_0(6,1,0.5,0.5,100,x)
G <- Gibbs
#burn in
Gibbs <- Gibbs[-(1:1000)]
```

Then we use the Simplified Laplace algorithm:

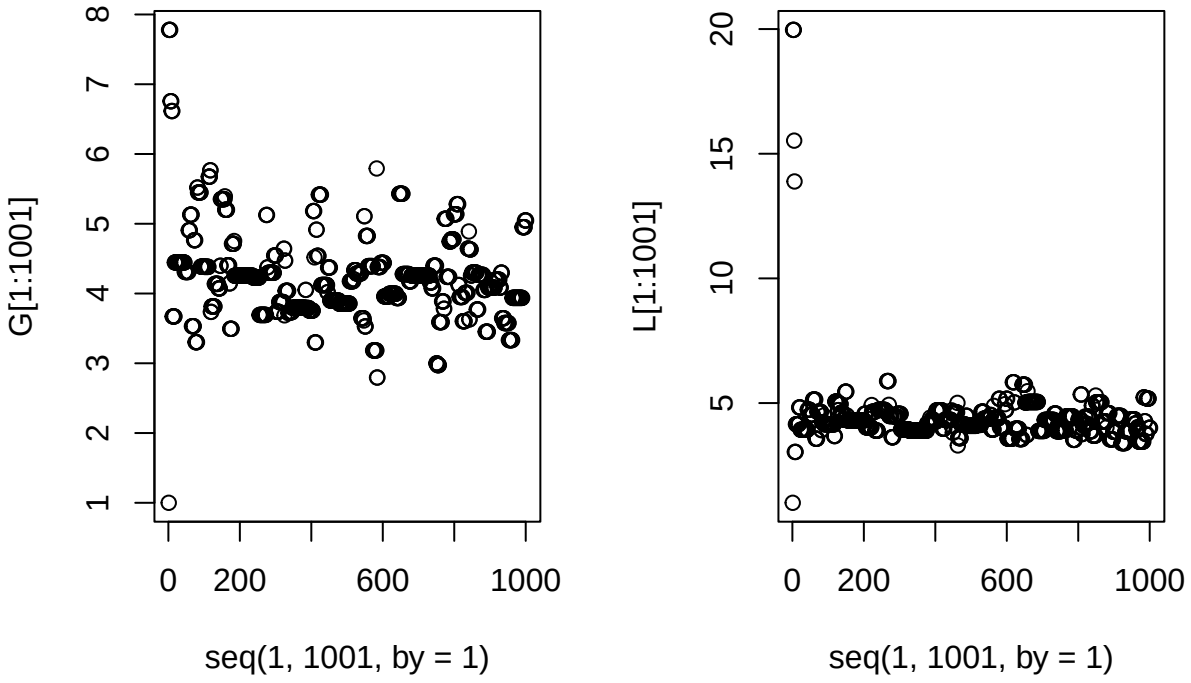
```
N <- 40000
S <- vector(length = N)
alpha <- vector(length = N)
beta <- vector(length = N)
Laplace <- M_L(6,1,0.5,0.5,100,x)
L <- Laplace
#burn in
Laplace <- Laplace[-(1:1000)]
#plot the histogram of the results
par(mfrow=c(1,2))
hist(Gibbs)
hist(Laplace)
```



The histograms from using Classical Gibbs algorithm and simplified Laplace algorithm suggest that with a reasonable number of observations, the posterior distribution is very well approximated by the simplified Laplace algorithm. But Classical Gibbs algorithm provides a narrower 95 percent credible interval for S which is $[4.58, 4.67]$, comparing to the one using simplified Laplace algorithm which is $[4.39, 5.21]$.

Moreover, the chain converges more rapidly to the stationary distribution with Laplace algorithm according to the paper. Now we further explore if it is still the case in our simulation study.

```
par(mfrow=c(1,2))
plot(x=seq(1,1001,by=1),y=G[1:1001])
plot(x=seq(1,1001,by=1),y=L[1:1001])
```



Show the scatter plots of first 1000 iterations of the Classical Gibbs algorithm and the simplified Laplace algorithm. Those plots suggest that the chain indeed converges more rapidly to the stationary distribution using Laplace algorithm.

6. Summary

The paper's main contribution is to propose an algorithm, that simulates from an approximate posterior density when integrating nuisance parameter out, by using a Laplace approximation at each iteration of a Classical Gibbs algorithm.

In this report, we simplified this proposed algorithm by assuming the prior distribution are simple and the nuisance parameter which we are integrating out has only one dimension. By those assumptions we then derived the specific formulas for each posterior distribution and applied Laplace approximation.

Trying to reproduce the paper's main contribution, our simulation study yields the same conclusion that even with just a reasonable number of observations, the Laplace algorithm approximates the posterior distribution pretty well. Additionally, the chain in the sampling algorithm converges more rapidly to the stationary distribution with Laplace approximation.

Therefore, this algorithm could be used in many applied studies where the large computation time is a real problem, as an improvement of the classical Gibbs algorithm in MCMC method.