MCMC using Reparametrization in Dynamic GLM

Jihwan Kim Taikyoung Kim

December 04, 2006

1 Introduction

The use of Markov chain Monte Carlo techniques can be used to deal with the dynamic models for exponential family observations. Metropolis-Hastings algorithms are employed to construct the chains in this paper. The general form of dynamic generalized linear models are introduced and the parameters in the model are reparametrized via systematic disturbances. And the posterior distribution for these disturbances is calculated in order to carry out Bayesian inference about dynamic models using MCMC. Finally, the parameters will be estimated. One example is provided for this method.

2 Dynamic Generalized Linear Model

Generally, the dynamic GLM consits of two different equations. The one is observation equation, the other system equation.

The obserbation equation for dynamic GLM can be written as,

$$f(y_t|\theta_t) \propto \exp\{\frac{y_t\theta_t + b(\theta_t)}{\phi_t}\}$$

and the means are $\mu_t = E(y_t|\theta_t) = b^{'}(\theta_t)$. The parameters are related to these means via the link function $\eta_t = g(\mu_t) = X_t\beta_t$. ϕ_t is assumed to be known.

The system equation for this dynamic system is,

$$\beta_t = \rho\beta_{t-1} + w_t, \quad w_t \sim N(0, W)$$

We assume taht β_1 has a prior of $N(a, R)$. This is the evolution of the model.

2.1 Reparametrization and Posterior distribution for system disturbances

By setting $w_1 = \beta_1$ and recursive method, $\beta_t = \sum_{j=1}^{t} \rho^{t-j} w_j, \ \ (t = 2, 3, \cdots, n)$.

Now, μ_t and θ_t are the functions of $w_1, w_2, \cdots, w_n$, since $g(\mu_t) = X_t\beta_t = X_t \sum_{j=1}^{t} \rho^{t-j} w_j$. That is, β_t is reparametrized with new parameters $w_1, w_2, \cdots, w_n$.

The above dynamic model with this new reparametrization is rewritten as,

$$f(y_t|w_1, \cdots, w_t) \propto \exp\{\frac{y_t\theta_t(w_t) + b(\theta_t(w_t))}{\phi_t}\}$$

The fully conditional posterior distirbution of these new parameters $w_1, w_2, \cdots, w_n$ needs to be calculated for using MCMC techniques. The hyperparameters, a, R, W, are assumed to be given.

$$\begin{aligned}\pi(w_1, w_2, \cdots, w_n|y_1, \cdot, y_n, a, R, W) &\propto \Pi_{t=1}^{n} f(y_t|w_1, \cdots, w_t) f_{w_1}(w_1) \\ &\times \ f_{w_2,\cdots,w_n}(w_2, \cdots, w_n) f_W(W)\end{aligned}$$

The samples for $w_1, w_2, \cdots, w_n$ are generated from the above target density with the proper proposal distribution using Metropolis-Hasting algorithm.

3 Metropolis-Hastings Algorithm

At each time t, the next state y_{t+1} is chosen by first sampling a candidate point $\tilde{y}$ from a proposal distribution $q(\cdot|y_t)$. Here, $q(\cdot|y_t)$ might be multivariate normal distribution with mean y and a fixed covariance matrix. The candidate point $\tilde{y}$ is then accepted with probability $\alpha(y, \tilde{y})$, where

$$\alpha(y, \tilde{y}) = min\{1, \frac{\pi(\tilde{y})q(y|\tilde{y})}{\pi(y)q(\tilde{y}|y)}\}$$

The next state becomes $y_{t+1} = \tilde{y}$, if the candidate point $\tilde{y}$ is accepted. If the candidate point $\tilde{y}$ is rejected, $y_{t+1} = y_t$, that is, the chain does not move.

Thus, the Metropolis-Hasting algorithm as follows :

- **Initialize** y_0 and **set** $t = 0$.
- **Repeat** {
  Sample a point $\tilde{y}$ from $q(\cdot|y_t)$
  Sample a uniform$(0,1)$ random variable U
  If $U \leq \alpha(y_t, \tilde{y})$, set $y_{t+1} = \tilde{y}$
  Otherwise, set $y_{t+1} = y_t$
  Increment t
  }

Note that the proposal distribution $q(\cdot|y)$ can have various forms and the stationary distribution of the chain will be $\pi(\cdot)$. The transition kernel for the Metropolis-Hastings algorithms is

$$p(y_{t+1}|y_t) = q(y_{t+1}|y_t)\alpha(y_t, y_{t+1}) + I(y_{t+1} = y_t)[1 - \int q(\tilde{y}|y_t)\alpha(y_t, \tilde{y})dy]$$

,where $I(\cdot)$ is the indicator function.

Using $\alpha(y, \tilde{y}) = min\{1, \frac{\pi(\tilde{y})q(y|\tilde{y})}{\pi(y)q(\tilde{y}|y)}\}$, we can derive the following.

$$\pi(y_{t+1}) = \int \pi(y_t)p(y_{t+1}|y_t)dy_t$$

Thus, once a sample from the stationary distribution has been obtained, all subsequent samples will be from that distribution.

Although any proposal function $q(\cdot|y)$ will ultimately deliver samples from the target density $\pi(\cdot)$, the rate of convergence to the stationary distribution will depend crucially on the relationship between $q(\cdot|y)$ and $\pi(\cdot)$. Therefore, for computational efficiency, $q(\cdot|y))$ should be chosen so that it can be easily sampled and evaulated. One possible proposal density is symmetric proposal density. The follwoing example constructs the chain using the symmetric proposal density.

4 Example

The observations of this example are the average numbers of a dishwasher sold from the first quarter in 1978 to the fourth quarter in 1985 (total 32 quarters). The Poisson time series model, $y_t \sim Poisson(\mu_t)$, is employed for this dataset.

$$\begin{aligned} f(y_t|\mu_t) &= \frac{e^{-\mu_t}\mu_t^{y_t}}{y_t!} \\ &\propto exp(y_t \log \mu_t - \mu_t) \\ &\propto exp(y_t\theta_t + b(\theta_t)) \end{aligned}$$

Let $\theta_t = \log \mu_t = \eta_t$ and $b(\theta_t) = -\mu_t = -e^{\theta_t}$.

The Poisson distribution has only one single parameter and if no regression structure is assumed, then the predictor is simplified to $\eta_t = \beta_t$.

This dynamic model may be compledted with the structure of a system equation.

$$\beta_t = \rho\beta_{t-1} + w_t, \ \ w_t \sim N(0, \sigma^2)$$

If $\rho = 1$, this model suggests the random walk structure of a system equation. We considered only the case of $\rho = 1$ in the example. $\beta_1 = w_1$ and the prior of β_1 is $N(a, R)$. As mentioned in the previous section, $\beta_t = \sum_{j=1}^{t} w_j$, $(t = 2, 3, \cdots, n)$ and μ_t and θ_t are the functions of $w_1, w_2, \cdots, w_n$. The parameter β_t is completely determined by $\beta_1 = w_1$ and $w_2, \cdots, w_t$.

Finally,

$$f(y_t|w_1, \cdots, w_t, a, R, \sigma^2) \propto \exp(y_t \sum_{j=1}^{t} w_j - e^{\sum_{j=1}^{t} w_j})$$

Next, Let's calculate the fully conditional posterior distribution for $w_1, w_2, \cdots, w_n$.

$$\begin{aligned} \pi(w_1, w_2, \cdots, w_n|y_1, \cdot, y_n, a, R, \sigma^2) &\propto \Pi_{t=1}^{n} f(y_t|w_1, \cdots, w_t) f_{w_1}(w_1) f_{w_2,\cdots,w_n}(w_2, \cdots, w_n) \\ &\propto \Pi_{t=1}^{n} \exp(y_t \sum_{j=1}^{t} w_j - e^{\sum_{j=1}^{t} w_j}) \frac{1}{\sqrt{R}} \exp(-\frac{(w_1 - a)^2}{2R}) \cdot \\ &\quad (\frac{1}{\sigma^2})^{\frac{n-1}{2}} \exp(-\frac{\sum_{t=2}^{n} w_t^2}{2\sigma^2}) \end{aligned}$$

While the prior distributions for $w_t's$ are independent in dynamic GLM, the resulting posterior distribution is not independent. By using Metropolis-Hasting algorithm, the samples for $w_1, w_2, \cdots, w_n$ are generated from the above target density.

The symmetric proposal in the Metropolis-Hasting Algorithm is the simplest case to generate the new candidates for sampling. Since $q(\tilde{w}, w) = q(w, \tilde{w})$, where $\tilde{w}$ is the candidate for updating w, the acceptance function reduces to

$$\alpha(w, \tilde{w}) = min\{1, \frac{\pi(\tilde{w})}{\pi(w)}\}$$

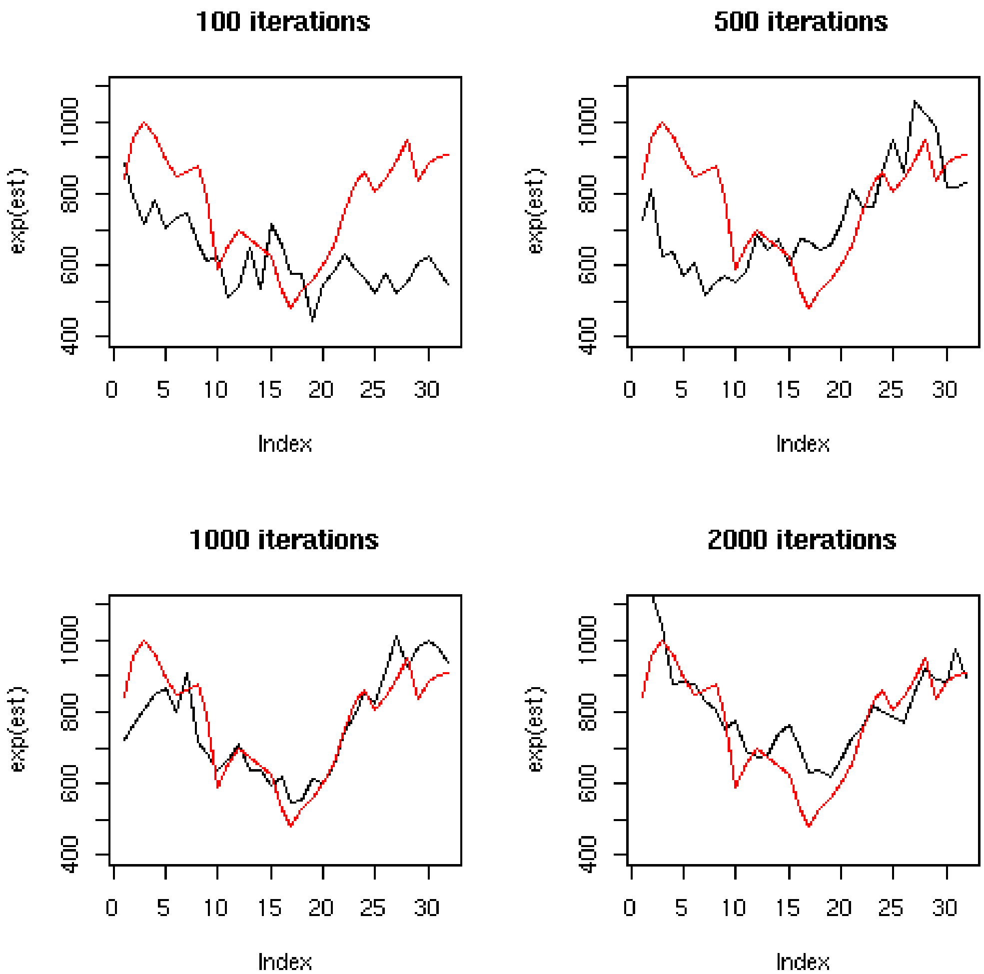
The log of the density ratio $\log \pi(\tilde{w}) - \log \pi(w)$ is calculated as the following.

$$\begin{aligned}\log \pi(\tilde{w}) - \log \pi(w) \quad &\propto \quad \sum_{t=1}^{n}(y_t \sum_{j=1}^{t} \tilde{w}_j - e^{\sum_{j=1}^{t} \tilde{w}_j}) - \sum_{t=1}^{n}(y_t \sum_{j=1}^{t} w_j - e^{\sum_{j=1}^{t} w_j}) \\ &-\{\frac{(\tilde{w}_1 - a)^2}{2R} - \frac{(w_1 - a)^2}{2R}\} \\ &-\{\frac{\sum_{t=2}^{n} \tilde{w}_t{}^2}{2\sigma^2} - \frac{\sum_{t=2}^{n} w_t^2}{2\sigma^2}\}\end{aligned}$$

This provides computationally cheapest way to obtain samples from the target density.

Based on the above density ratio, the Metropolis-Hasting updating was performed.

The sampled disturbances $w_t's, \ (t = 1, 2, \cdots, 32)$ are used to estimate the parameters $\beta_t's, \ (t = 1, 2, \cdots, 32)$. The Figure 1 is the plot for estimated numbers and actual numbers of dishwasher sold at each quarter with 100, 500, 1000 and 2000 iterations. The red line is the actual data and the black line is the estimated numbers. With 1000 iterations, the estimated numbers seem to be fitting better to the real data points.



5 Discussion

We considered only one case of proposal density. But the other proposal densities, like random walk proposal or univariate normal proposal, could be considered. Also, the convergence rates could be compared using the differnet proposal functions.

In this paper, we didn't carry out convergence diagnostics for the sampling. For diagnostics, we can use either `BOA` package in R or CODA (Convergence Diagnosis and Output Analysis) software.

A R Code

```
#22s:166 Final Project
#Dynamic GLM
#Sales data of home appliances
#From 1st quarter 1978 to 4th quarter 1985

data<-read.table("~/courses/22s166/project/salesdata",head=T)
dishwasher<-data[,2]

sales<-function(y,R,sigma,N,n=1) {
z<-matrix(0,ncol=32,nrow=N)
nw<-double(32);w<-double(32);a<-log(y[1])
w[2:32]<-rnorm(31,0,sigma);w[1]<-a
for(i in 1:N) {
for(j in 1:n) {
 nw[2:32]<-rnorm(31,0,sigma)
 nw[1]<-rnorm(1,w[1],R)
 u<-double(32); nu<-double(32)
 for(k in 1:32) {
   u[k]<-y[k]*sum(w[1:k])-exp(sum(w[1:k]))
   nu[k]<-y[k]*sum(nw[1:k])-exp(sum(nw[1:k]))}
   logDensRatio<-sum(nu)-sum(u)-0.5*(1/sigma)*(sum(nw^2)-sum(w^2))-
     0.5*(1/R)*((nw[1]-a)^2-(w[1]-a)^2)
   if(is.finite(logDensRatio) && log(runif(1)) <logDensRatio) w<-nw}
 z[i,]<-w
}
z
}

b<-sales(dishwasher,1,.2,1000)
N<-length(b[,1]); k<-double(N-1)
        for(m in 2:N) if(b[,1][m] != b[,1][m-1]) k[m-1]<-1
f<-matrix(0,ncol=32,nrow=N)
for(i in 1:(N-1)) if(k[i]==1) f[i,]<-b[i+1,]
ff<-matrix(f[f != 0],ncol=32)
di<-dim(ff)[1]
```

```
theta<-matrix(0,ncol=32,nrow=di)
for(i in 1:di) {
for(j in 1:32){
theta[i,j]=sum(ff[i,][1:j])
}}

est<-apply(theta,2,mean)
plot(exp(est),type='l',ylim=c(400,1100))
lines(dishwasher,type='l',col='red')
```

References

[1] Dani Gamerman(1997) Markov Chain Monte Carlo,1st ed. CHAPMAN & HALL/CRC

[2] Dani Gamerman(1998), Markov chain Monte Carlo for Dynamic Generalised Linear Models, *Biometrika.* Vol.85. No.1, 215-227.

[3] W. R. Gilks, S. Richardson and D. J. Spiegelhalter (1995), Markov Chain Monte Carlo in Practice, London: Chapman & Hall