EC 607: Bayesian Econometrics

Bayesian Computation IV

Prof. Jeremy M. Piger

Department of Economics

University of Oregon

Last Revised: March 22, 2019

1 Diagnostics for MCMC Samplers

- An MCMC algorithm will return an answer given the inputs you give it. It is thus useful to have some diagnostic tools to help us feel (more) confident that the algorithm has done what we wanted it to do.

- The primary question is whether we have taken enough draws, both pre-convergence (the burn-in period) and post convergence. In the following discussion, assume that we are sampling from a posterior distribution, $p(\theta|Y)$ using an MCMC sampler. We have $G0$ burn-in draws, followed by G post-convergence draws. There are two related questions here:

  - Did we take enough draws to allow the MCMC sampler has converged? (Is G_0 big enough?)

  - Once convergence has occurred, how many draws are needed to adequately sample the posterior distribution and estimate objects of interest? (Is G big enough?)

- While these sound like separate questions, they are often closely related. If your sampler was such that it took a long time to converge to the target density (G_0 needed to be large) then it is likely that it will also take a long time for your sampler to adequately sample the target density (G needs to be large). A rule of thumb that is often used in practice is to set $G = 10 \times G_0$. This automatically raises G as we get into situations where G_0 is required to be larger.

- Some of the diagnostics we will talk about are primarily designed to evaluate the adequacy of G_0. Others are better thought of as jointly evaluating the adequacy of G_0 and G. Before getting to specific diagnostics, it is important to recognize that none of these diagnostics are foolproof. Indeed, one can come up with examples where all of these diagnostics raise no red flags, and yet the sampler has either not converged or not properly explored the target density post convergence. These examples usually have to do with multimodal target densities.

- **Common MCMC Diagnostics:**

  1. **Trace Plots**

     A very simple graphical diagnostic is known as a **trace plot**. Consider an individual parameter, θ_i. A trace plot for θ_i is a graph where the y-axis shows the value of $\theta_i^{[g]}$ and the x-axis shows the value of $[g]$. A sampler that has converged will yield trace plots that have settled down and are sampling over a stable region, which will be the support of the posterior density for θ_i. We would plot trace plots for our burn-in period. When we see the trace plot settle down this is consistent with a burn-in period (G_0) that is large enough.

  2. **Running Mean Plot**

     Another simple graphical diagnostic is a **running mean plot**, which plots the sample mean of the draws of $\theta_i^{[g]}$ up through draw g against g. Provided $G0$ is big enough, then the sample mean should have settled down by the time $G0$ is reached and shouldn't be changing much as we add new draws beyond that. When we see the running mean plot for each parameter settle down this is consistent with a burn-in period (G_0) that is large enough.

  3. **Geweke's (1992) Statistic**

     Geweke (1992, *Bayesian Statistics*) proposes a formal statistic to assess the adequacy of G and G_0 based on running means. Suppose we have done G^* draws and we want to know if $G_0 < G^*$ is adequate for a burn-in period, leaving $G = G^* - G_0$ availabe to estimate objects of interest about the posterior density. Calculate the sample mean of θ_i for the first 10% of the draws and the final 50% of the draws. We then test whether these came from distributions with the same population mean by conducting a differences in means test. Geweke's formal statistic is given in the Koop (2003) textbook and is built into many MCMC computer packages. If we fail to reject the null hypothesis of equal means, then this is consistent with $G0$ and G being big enough. If we reject, then this suggests that either our G_0 or our G are not big enough. This test should be repeated for each θ_i.

4. **Running Multiple Chains with Overdispersed Starting Values**

   One of the more convincing checks that both G_0 and G are large enough is to demonstrate that you obtain similar results in two or more separate runs of a sampler with starting values that are far from each other. Such starting values are referred to as "overdispersed." To compare the results you could compare posterior statistics computed from each chain (such as the posterior mean, median and variance). You could also show that trace plots or running mean plots for the different chains converge to looking the same.

   A single run of your sampler, or multiple runs with the same starting values, leaves leaves open the possibility that your sampling chain is stuck sampling around the same mode of a multimodal distribution. Running multiple chains with overdispersed starting values can help guard against this possibility, since hopefully one of your chains will find this alternative mode.

   Note that if some of your chains with overdispersed starting values don't look the same as the others, this doesn't necessarily mean the other chains haven't converged. When you are trying over dispersed starting values, you can easily start the sampler in such a bad place with some starting values that convergence will take longer for those starting values, or will never practically occur. That is, it may just be that some of your chains haven't converged for their starting values.

5. **Autocorrelation Function**

   Samples taken using an MCMC sampler are not independent. Instead, they will display serial correlation. A common diagnostic is to assess how much serial correlation is present. The $j^{[th]}$ order sample autocorrelation of θ_i (calculated over the burn-in period) is:

$$r_{i,j} = \frac{\sum_{g=j+1}^{G_0} \left(\theta_i^{[g]} - \bar{\theta}_i\right)\left(\theta_i^{[g-j]} - \bar{\theta}_i\right)}{\sqrt{\sum_{g=1}^{G_0} \left(\theta_i^{[g]} - \bar{\theta}_i\right)^2}}$$

where $\bar{\theta}_i = \frac{1}{G_0} \sum_{g=1}^{G_0} \theta_i^{[g]}$ is the sample mean of $\theta_i^{[g]}$ calculated over the burn-in period. We can summarize the autocorrelations graphically using the autocorrelation function, which is a plot of $r_{i,j}$ agains j. We would expect the autocorrelations to die off as j increases, since the chain is ergodic. However, if the autocorrelations die off very slowly, then the chain will explore the parameter space very slowly. This means that more draws will need to be taken to obtain convergence than would be true if the autocorrelations were lower.

High autocorrelation is an issue not only for convergence, but if these high autocorrelations continue in the sampler post-convergence, which they likely will, then this means that you will need to sample more post-convergence as well. Highly autocorrelated post-convergence draws means that you will have long strings of "clumpy" observations that are not representative of the unconditional target density. You will need to sample more to average out these clumpy strings. One way to assess this in post-convergence samples is to compute the **Effective Sample Size**:

$$ESS = \frac{G}{1 + 2\sum_{k=1}^{\infty} r_{i,j}}$$

where $r_{i,j}$ is the j^{th} order autocorrelation computed over the *post-convergence* sample. We can deal with the ∞ term by setting the limit equal to some large number such that $r_{i,j}$ is close to zero. The intuition of the effective sample size is that it tells us how large a sample of independent draws of θ_i would need to be to contain the same amount of information about $p(\theta_i|Y)$ as was contained in our sample of G autocorrelated draws. As an example, suppose we have taken 1 million draws of a parameter that display autocorrelation consistent with an AR(1) with AR parameter of 0.99. In this case, these 1 million draws would be roughly equivalent to 5000 independent draws.

It is sometimes suggested that because of autocorrelation in MCMC draws, we should not keep every post-convergence draw of θ_i for posterior inference, but should instead only use every d^{th} draw, where d is chosen to be large enough to eliminate the correlation between draws. This process, known as **thinning**, is usually not very useful. While it will be succesful in reducing autocorrelation in the samples of θ_i, it throws out any useful information that was in the autocorrelated draws about the posterior. One place where thinning can be useful is if you run into storage constraints when saving your sampled draws.

2 Exercise: MCMC Diagnostics from Linear and Nonlinear Regression Models

In the dropbox you will find two zipped collections of files. One is called "Nonlinear Regression MH with Diagnostics.zip." The other is called "Linear Regression GS with Diagnostics.zip." These programs implement the nonlinear and linear regression models that we have used earlier, but add in the regression diagnostics discussed above. Work with these files to make sure you understand what they do and how they work. Based on these diagnostics, what do you think is an appropriate value of $G0$? Make sure to do some analysis with multiple runs from overdispersed starting values.

3 Hierarchical Priors

- Some of the remaining models we will discuss use a special kind of prior distribution known as a hierarchical prior. We will discuss hierarchical priors in general first before continuing.

- Consider a $k \times 1$ vector of parameters $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_k)'$, where the elements of α play similar roles in the model. For example, they may all be variances assigned to individual observations. Or response coefficients to the same variable across individuals in a panel.

- It is sometimes the case that k will be large relative to the amount of data we have, which will make estimation of each of these k parameters difficult. When this is true, a popular approach to setting the prior for the elements of α proceeds as follows: First, α is assumed to arise from a distribution that depends on a vector of parameters, λ, that is of much lower dimension than k:

  $$\alpha \sim p(\alpha|\lambda)$$

  where we have explicitly denoted that the prior depends on the parameters λ. One common way of doing this is to have a common prior across all α_i. For example, we might say:

  $$\alpha_i \sim N(\mu, V)$$

- If we stopped here, λ would serve as the vector of hyperparameters. However, we will instead treat λ as parameters to be estimated. Thus, the next step in setting a prior is to place a prior on λ, $p(\lambda)$.

- This prior is what is known as a **hierarchical prior**. With a hierarchical prior, a prior for one unobserved object is determined, through a hierarchy, by the prior for other unobserved objects. In particular, in this case we have a vector of unknowns, α, as well as λ. The prior for these parameters is then set as:

$$p(\alpha, \lambda) = p(\alpha|\lambda)\, p(\lambda)$$

At the top of the hierarchy is λ. Our prior for these parameters then completely determines our prior for α. In particular:

$$p(\alpha) \sim \int_{\lambda} p(\alpha|\lambda)\, p(\lambda)\, d\lambda$$

- Note that with a hierarchical prior, the elements of α lose their place as traditional parameters, and become a stochastic function of other parameters in the model λ. In other words, we are specifying a model for α as a function of some other parameters. Thus, this is a way of reducing the number of parameters in the model from k to the number of parameters in λ, but also maintaining the flexibility that allowing the elements of α to be different gives us. Of course the cost is that it places structure on the distribution of the elements of α that might not be consistent with the data. That is, the α are no longer being estimated as free parameters.

- Hierarchical priors are used frequently to place structure on flexible models with lots of parameters (e.g. random coefficients models, random effects models, etc.)

- A hierarchical prior was evident in one model we talked about already, namely the probit model. In this model, the y^* served as a high dimensional vector of unobserved objects. We modeled these as dependent on the smaller parameter vector β. Our prior for β then determined our prior y^*. Here β was at the top of the hierarchy.

4 Linear Regression with other Error Distributions: A Mixture of Normals Model

- In our discussion of the linear regression model so far, we have assumed that the vector of disturbances follows the following multivariate normal distribution:

$$\epsilon \sim N\left(0_N, h^{-1} I_N\right)$$

which is alternatively written as:

$$\epsilon_i \sim i.i.d.N\left(0, h^{-1}\right) \quad i = 1, 2, \cdots N$$

- A popular approach in Bayesian econometrics to relax this restriction is to allow ϵ_i to be determined by a **mixture of normals** distribution:

$$\epsilon_i = \sum_{j=1}^{M} D_{ij}\eta_{ij}$$

where $\eta_{ij} \sim N\left(\alpha_j, h_j^{-1}\right)$ and $D_{ij} = 0$ or 1 for $j = 1, \ldots, M$ and $\sum_{j=1}^{M} D_{ij} = 1$.

In this setup, the shock ϵ_i is drawn from one of M different normal distributions, with each potentially having a different mean and variance.

- Mixture of normals distributions are popular because a very flexible distribution can be obtained by mixing together known distributions. This flexible distribution can then be used to approximate an unknown distribution of interest. Here we will focus on a mixture of normal distributions, but in principle any distribution can be mixed.

- We can consider the case where D_{ij} is observed or unobserved. In the case where it is unobserved, then it is unknown which component the i^{th} disturbance term was drawn from. In this case, we introduce a probability that $D_{ij} = 1$, called p_j, $j = 1, \ldots, M$. In this case, the p_j become additional parameters to be estimated.

- Mixture of normal distributions are very flexible, and can be used to effectively approximate many other distributions. Thus, using a mixture of normals distribution can be an effective way to approximate an unknown distribution.

- Here we will do a particular example of a mixture of normals distribution in which there are N mixtures, and D_{ij} is known. This is popular approach for allowing for heteroskedasticity of an unknown form.

- Consider the linear regression model expressed in matrix form:

$$Y = X\beta + \epsilon$$

  To capture heteroskedasticity of an unknown form we will assume:

$$\epsilon_i = \eta_i$$
$$\eta_i \sim N\left(0, h_i^{-1}\right)$$

  The parameters are then β and $h = (h_1, h_2, \cdot\cdot\cdot, h_N)'$.

  This is a special case of the mixture of normals distribution in which $\alpha_j = 0$. Such a mixture is referred to as a **scale mixture of normals**. Also, in this example, each ϵ_i is assumed to have it's own normal distribution, so that $M = N$, and $D_{ij} = 1$ if $i = j$ and 0 otherwise.

- We will assume prior independence between β and h_1, $h_2, \cdot\cdot\cdot, h_N$:

$$p(\beta, h) = p(\beta) \prod_{i=1}^{N} p(h_i)$$

where:

$$\beta \sim N(\mu, V)$$

For h, we will use a hierarchical prior:

$$h_i \sim \text{Gamma}(m, v)$$

Here m and v are parameters to be estimated, meaning we need a prior for m and v, $p(m, v)$.

- With this hierarchical prior, we have changed the model from having N free parameters h_i, to having two free parameters m and v. Again, the benefit of this is that we have substantially reduced the number of parameters to be estimated. Again, the cost is that we have placed structure on the distribution of the h_i, namely we have modeled them as arising from the distribution $\text{Gamma}(m, v)$.

- To proceed with estimation, we need a prior for m and v. We set these as independent exponential distributions, with parameters a_m and a_v:

$$\begin{aligned} p(m, v) &= p(m)\, p(v) \\ m &\sim \text{exponential}(a_m) \\ v &\sim \text{exponential}(a_v) \end{aligned}$$

  The exponential distribution enforces the restriction that both of these parameters must be positive. The exponential distribution is a special case of a Gamma distribution. Other priors could be used without changing the sampling algorithm given below by much.

- This model can be estimated via a Metropolis-within-Gibbs Sampler with three blocks: β, h, and $\{v, m\}$. Specifically, we will sample iteratively from:

1. $p\left(\beta|m,v,h,Y,X\right)$
2. $p\left(h|\beta,m,v,Y,X\right)$
3. $p\left(m,v|\beta,h,Y,X\right)$

Note that in this sampler, the treatment of h is in the spirit of **data augmentation** as in the probit model. The N elements of h are treated as unknown objects that are sampled. These drawn values are then conditioned on in other steps, which will make those sampling steps easier.

Sampling $p\left(\beta|m,v,h,Y,X\right)$: Using Bayes Rule we have:

$$\begin{aligned}
p\left(\beta|m,v,h,Y,X\right) &\propto p\left(Y|\beta,m,v,h,X\right)p\left(\beta|m,v,h,X\right)\\
p\left(\beta|m,v,h,Y,X\right) &\propto p\left(Y|\beta,m,v,h,X\right)p\left(\beta\right)
\end{aligned}$$

where:

$$p\left(Y|\beta,m,v,h,X\right)=(2\pi)^{(-N/2)}\prod_{i=1}^{N}h_i^{1/2}\exp\left[\left(Y-X\beta\right)'\Sigma^{-1}\left(Y-X\beta\right)\right]$$

where $\Sigma^{-1}=diag\left(h_1,h_2,h_3,\ldots,h_N\right)$.

Plugging in the equation for the prior and likelihood function, and doing some rearranging, gives us:

$$\beta|m,v,h,Y,X\sim N\left(\bar{\mu},\bar{V}^{-1}\right)$$

where:

$$\begin{aligned}
\bar{V} &= V^{-1}+X'\Sigma^{-1}X\\
\bar{\mu} &= \bar{V}^{-1}\left(V^{-1}\mu+X'\Sigma^{-1}Y\right)
\end{aligned}$$

Sampling $p\left(h|\beta,m,v,Y,X\right)$:

Using Bayes Rule we have:

$$\begin{aligned}p\left(h|\beta,m,v,Y,X\right) &\propto p\left(Y|\beta,m,v,h,X\right)p\left(h|\beta,m,v,X\right)\\ p\left(h|\beta,m,v,Y,X\right) &\propto p\left(Y|\beta,m,v,h,X\right)p\left(h|m,v\right)\end{aligned}$$

Plugging in the likelihood function and the equation for the Gamma distribution that describes $p\left(h|m,v\right)$, and doing some rearranging, gives us:

$$p\left(h|\beta,m,v,Y,X\right)=\prod_{i=1}^{N}p\left(h_i|\beta,m,v,Y,X\right)$$

where:

$$h_i|\beta,m,v,Y\sim\text{Gamma}\left(\bar{m}_i,\bar{v}_i\right)$$

$$\begin{aligned}\bar{v}_i &= 1+v\\ \bar{m}_i &= \frac{\bar{v}_i}{\epsilon_i^2+\frac{v}{m}}\end{aligned}$$

where ϵ_i is the i^{th} row of $Y-X\beta$.

Sampling $p\left(m,v|\beta,h,Y,X\right)$:

Using Bayes Rule we have:

$$\begin{aligned}
p\left(m,v|\beta,h,Y,X\right) &\propto p\left(Y|\beta,m,v,h,X\right)p\left(m,v|\beta,h,X\right)\\
p\left(m,v|\beta,h,Y,X\right) &\propto p\left(m,v|h\right)\\
p\left(m,v|\beta,h,Y,X\right) &\propto p\left(h|m,v\right)p\left(m,v\right)\\
p\left(m,v|\beta,h,Y,X\right) &\propto p\left(h|m,v\right)p\left(m\right)p\left(v\right)
\end{aligned}$$

Plugging in for these and eliminating the terms that don't depend on m or v we have:

$$p\left(m,v|\beta,h,Y\right)\propto\exp\left[-\left(\frac{m}{a_m}+\frac{v}{a_v}\right)\right]\prod_{i=1}^{N}\left(\frac{2m}{v}\right)^{-\frac{v}{2}}\Gamma\left(\frac{v}{2}\right)^{-1}h_i^{\frac{v-2}{2}}\exp\left[-\frac{h_i v}{2m}\right]$$

This does not correspond to the kernel of a known distribution. However, it can be sampled using a Metropolis step. Given this equation for the kernel of the posterior, we can form the acceptance probability for proposals.

5 Exercise: Linear Regression Model with a Scale Mixture of Normals Error Distribution

In the dropbox you will find a zipped collection of files called "Linear Regression Mixture.zip." This implements the Metropolis-within-Gibbs sampler for the scale mixture of normals model outlined in the previous section. Work with these files to make sure you understand what they do. How do your prior on m and v affect the results? Add in some MCMC diagnostics to the program and assess the convergence of the sampler.