

1 Methods for Posterior Simulation

Let $p(\theta|y)$ be the posterior. Koop presents four methods for (posterior) simulation.

1. Monte Carlo integration: draw from $p(\theta|y)$.
2. Gibbs sampler: sequentially drawing from each of the full conditional posteriors e.g. $p(\theta_1|\theta_2, y)$ and $p(\theta_2|\theta_1, y)$.
3. Importance sampler: draw from $q(\theta)$ rather than $p(\theta|y)$ but weigh the draws appropriately.
4. Metropolis-Hastings algorithm: draw from $q(\theta)$ rather than $p(\theta|y)$ but only accept some of the draws.

Usually 1. is preferred though 3. may be better in some cases even where 1. is possible.

Methods 2. and 4. are both Markov Chain Monte Carlo (MCMC) methods that generate correlated draws and MCMC diagnostics are needed to confirm that the chain has converged.

Koop introduces these methods one per chapter in chapters 1-5.

1.1 Monte Carlo Integration - Koop pp. 8-10

Let β^s $s = 1, \dots, S$ be a collection of S random draws from $p(\beta|y)$ and let $g(\cdot)$ be any function. Then **Monte Carlo integration** uses the result that

$$\hat{g}_S = \frac{1}{S} \sum_{s=1}^S g(\beta^s) \rightarrow E[g(\beta)|y] \text{ as } S \rightarrow \infty \quad (1)$$

For example, suppose I want to calculate $\sigma_{\beta|y}^2$ but I only know $E[\beta|y]$. Since $\sigma_{\beta}^2 = E[\beta^2] - (E[\beta])^2$ I first need to estimate $E[\beta^2|y]$. This can be done given the posterior density for β , as I can draw many $\beta^{(s)}$ from this posterior and calculate

$$\hat{\beta}_S^2 = \frac{1}{S} \sum_{s=1}^S (\beta^s)^2 \rightarrow E[\beta^2|y].$$

Then compute $\hat{\sigma}_{\beta|y}^2 = \hat{\beta}_S^2 - (E[\beta|y])^2$. [Aside: Notice that in general, $g(E[\beta]) \neq E[g(\beta)]$].

1.2 The Gibbs Sampler - Koop pp. 62-64

Quite often, it is not possible to obtain the joint posterior density of all the parameters analytically. In fact, for more complicated models, this will be the rule rather than the exception.

Section 4.2 in Koop discusses what happens in the normal linear regression model with independent priors so that $p(\beta, h) = p(\beta)p(h)$. (Whereas chapter 3 had dependent priors with $p(\beta, h) = p(\beta|h)p(h)$).

In particular, assume that the independent priors for β and h are

$$\begin{aligned} p(\beta) &= N(\underline{\beta}, \underline{V}) \\ p(h) &= G(\underline{s}^{-2}, \underline{\nu}). \end{aligned}$$

Then Koop shows that the conditional posteriors for β given h and for h given β are

$$\begin{aligned} p(\beta|y, h) &\propto N(\bar{\beta}, \bar{V}) \\ p(h|y, \beta) &\propto G(\bar{s}^{-2}, \bar{\nu}) \end{aligned}$$

However the joint posterior cannot be obtained from these as

$$p(\beta, h|y) \neq p(\beta|y, h)p(h|\beta, y).$$

If we wanted to use Monte Carlo integration to compute posterior means and standard deviations, what we really need are draws from this joint posterior, not the conditional posteriors.

The **Gibbs sampler** is a method for **generating draws from the joint posterior** by **using draws of the conditional posteriors**. As we study more complex models, it will be easier to conjecture priors for blocks of the parameter vector (rather than a joint prior for all the parameters) and this will inevitably result in a corresponding set of conditional posteriors. This is when the Gibbs sampler becomes useful.

For simplicity divide a generic parameter vector θ into two blocks, θ_1 and θ_2 , although it should be mentioned that the Gibbs sampler generalizes to multiple blocks directly. In the linear regression example with independent priors $\theta_1 = \beta$, and $\theta_2 = h$. Just as in this example, assume that the conditional posteriors $p(\theta_1|\theta_2, y)$ and $p(\theta_2|\theta_1, y)$ are available.

Note that

$$p(\theta_1, \theta_2|y) = p(\theta_1|\theta_2, y)p(\theta_2|y)$$

Hence, if a draw from $p(\theta_2|y)$ were available, say θ_2^0 , then we could draw a valid θ_1^1 from the joint posterior with the conditional posterior and θ_2^0 . Similarly, if we had a valid draw, say θ_1^1 , from the joint posterior, then we could plug it into the conditional posterior of θ_2 to obtain θ_2^1 .

A summary of the Gibbs sampler algorithm is as follows:

- Step 1: pick an initial θ_2^0
- Step 2: draw once from $p(\theta_1|\theta_2^0, y)$ to obtain θ_1^1
- Step 3: draw once from $p(\theta_2|\theta_1^1, y)$ to obtain θ_2^1
- Step 4: go back to step 2 and iterate S times to generate S Monte Carlo draws.

Note that since the marginal $p(\theta_2|y)$ is unknown the Gibbs sampler instead draws from the conditional $p(\theta_1|\theta_2, y)$ (and similarly instead of $p(\theta_1|y)$ it draws from $p(\theta_1|\theta_2, y)$).

Remarks

- How to pick θ_2^0 and how sensitive are the results to the choice of this initial condition? Under *weak conditions*, θ_2^0 does not matter as the Gibbs sampler will converge to draws that do come from the joint posterior. We will see that some convergence diagnostics are based on trying alternative starting values. However, poorly choosing θ_2^0 in a chain that is highly correlated will require an enormous number of replications before convergence.
- For this reason, it is customary to drop the first S_0 draws and retain $S_1 = S - S_0$ draws. The first S_0 are called **burn-in replications** and S_0 should be large enough that we have converged.
- It is important to realize that, unlike old fashion Monte Carlo simulation, the **Gibbs sampler draws are correlated** rather than i.i.d. (hence the Gibbs sampler is an example of a Markov Chain Monte Carlo method). This can be easily seen from the fact that the s draw is based on realizations from the $s - 1$ draw. Therefore, it is usually the case that more draws are necessary to get accurate Monte Carlo integration. We need a way to determine when S_1 is big enough.

- An example where *weak conditions* may be violated is when the joint posterior has two (or more) disconnected modes. In that case, the Gibbs sampler will stay in one of the modes but not migrate to the other to obtain draws from that other mode. Fortunately, for most applications this is not an issue.
- Imagine that you wanted to calculate the posterior mean of θ_1 with the S_1 draws on θ_1 and θ_2 from the Gibbs sampler. It turns out that

$$\frac{1}{S_1} \sum^{S_1} \theta_1^s$$

while a valid Monte Carlo integration for the sample mean, is sometimes not the most efficient calculation because it disregards information contained in the draws for θ_2 . We will see this in more detail below.

- In some situations, the conditional posterior for θ_1 may be available but not the conditional posterior for θ_2 . We will see other algorithms (notably Metropolis-Hastings) that fix this problem.

1.2.1 A simple Application of the Gibbs Sampler: Generating Correlated draws for a bivariate Normal from univariate Normals

Consider the experiment of drawing from a bivariate normal density with zero means, unit variances and correlation ρ . That is

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \sim N \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}; \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right]$$

Usually we would draw two sets of univariate standard normal draws and then multiply these two vectors of draws by a matrix P that satisfies

$$P'P = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

This matrix can be easily obtained from the Cholesky decomposition of

$$\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

Instead, we could use the Gibbs sampler by noticing that for the normal

$$p(y_1|y_2) \sim N(\rho y_2, 1 - \rho^2)$$

$$p(y_2|y_1) \sim N(\rho y_1, 1 - \rho^2)$$

The algorithm would go something like this (for a given value of ρ) :

- pick y_1^0 (say $y_1^0 = 0$ since that is the mean of the joint density, usually we would not know this).
- Step 1: generate y_2^1 by drawing it from $N(\rho y_1^0, 1 - \rho^2)$
- Step 2: generate y_1^1 by drawing it from $N(\rho y_2^1, 1 - \rho^2)$
- go to Step 1 with y_1^1 and repeat $S = S_0 + S_1$ times.

I have prepared a small GAUSS program that uses the Gibbs sampler for this example. You should experiment with it for different values of ρ and check that when ρ is high, more draws are needed to achieve the same results as those obtained instead by the usual i.i.d. draws from the Monte Carlo.

1.3 Importance Sampling - Koop p. 78-81

In some cases, obtaining draws from the posterior $p(\theta|y)$ may be difficult to do. However, it may be easy to draw from another density function $q(\theta)$, which is referred to as the **importance function**. **Importance sampling** consists in using the draws from $q(\theta)$ in a Monte Carlo integration computation, but with each draw weighted appropriately so that the MC integration reflects the properties of $p(\theta|y)$.

Specifically, suppose we are interested in computing $E[g(\theta)|y]$. For example, if we are interested in the posterior variance of β , we may want to compute $E[\beta|y]^2$ and $E[\beta^2|y]$ so that $\sigma_\beta^2 = E[\beta^2|y] - (E[\beta|y])^2$. Let θ^s , $s = 1, \dots, S$ be a random sample from $q(\theta)$, then we can use weighted Monte Carlo integration as follows:

$$\hat{g}_S = \frac{\sum^S w(s)g(\theta^s)}{\sum^S w(s)} \rightarrow E[g(\theta)|y] \tag{2}$$

with weights

$$w(s) = \frac{p(\theta^s|y)}{q(\theta^s)}.$$

Notice that when we draw directly from the posterior, $q(\theta) = p(\theta|y)$ and therefore the weights are always one, so that (2) reduces to (1). Furthermore, because $w(s)$ appears in the numerator and the denominator of (2), all we need to calculate $w(s)$ are the kernels of $q(\theta)$ and $p(\theta|y)$, i.e., if $q^*(\theta) \propto q(\theta)$ and $p^*(\theta|y) \propto p(\theta|y)$ then

$$w(s) = \frac{p^*(\theta^s|y)}{q^*(\theta^s)}$$

Remarks

- The importance function $q(\theta)$ should be reasonably close to the posterior $p(\theta|y)$, otherwise we will find many cases for which $w(s)$ is practically zero. The implication of this observation is that, to get an accurate estimate \hat{g}_S , we will need a number of draws that is significantly higher than if we had been able to simulate directly from the posterior, $p(\theta|y)$.
- Finding good importance functions $q(\theta)$ becomes more difficult for high-dimensional posteriors, i.e., posteriors for a high number of parameters.
- Importance sampling provides a simple way to conduct posterior inference on **parameters subject to inequality constraints**. For example, suppose the constraints on the parameter space can be summarized by $\beta \in A$ (e.g., $\beta > 0$, $\beta_1 + \beta_2 = 1$). Further suppose you are able to draw from the posterior (analytically for special cases of the linear regression model, as we have seen, or more generally with the Gibbs sampler). Then the weights $w(s)$ can be obtained as $w(s) = 1$ if $\beta^s \in A$, 0 otherwise. Thus, for the constrain $\beta > 0$, we effectively retain those draws that are positive and discard those that are not and then compute the Monte Carlo average as usual.

1.4 The Metropolis-Hastings Algorithm - Koop pp. 92-99

The Metropolis-Hastings algorithm is a third method for posterior simulation. Like importance sampling, it uses an auxiliary density that is easy to take draws from. Rather than weighing each draw, Metropolis-Hastings weighs all draws equally but not all the draws are accepted. (This is like

the accept-reject method). In addition and unlike importance sampling, the draws from this auxiliary density depend on past values of the accepted draws (and hence this method is an MCMC algorithm). Using the same notation that we used when we discussed importance sampling, this auxiliary density $q(\theta|\theta^{s-1})$ is now called the **candidate generating density**.

We begin by discussing the **Metropolis algorithm**, which can be summarized as follows:

- Step 1: Choose an initial value of θ , say θ^0
- Step 2: Draw θ^{1*} from $q(\theta|\theta^0)$
- Step 3: Calculate the ratio

$$r = \frac{p(\theta^{1*}|y)}{p(\theta^0|y)}$$

- Step 4: If $r \geq 1$, set $\theta^1 = \theta^{1*}$, otherwise set $\theta^1 = \theta^0$ with probability r , $\theta^1 = \theta^0$ with probability $1 - r$.
- Go to step 2.

Consequently, in general the probability that θ^* is accepted is given by

$$\alpha(\theta^*, \theta^{s-1}) = \min \left[\frac{p(\theta^*|y)}{p(\theta^{s-1}|y)}, 1 \right]$$

The Metropolis-Hastings algorithm uses instead the following acceptance probability

$$\alpha(\theta^*, \theta^{s-1}) = \min \left[\frac{p(\theta^*|y)q(\theta^{s-1}|\theta^*)}{p(\theta^{s-1}|y)q(\theta^*|\theta^{s-1})}, 1 \right]$$

Remarks

- When $q(\theta^*|\theta^{s-1})$ is symmetric, then Metropolis-Hastings reduces to Metropolis.
- A Metropolis chain will sometimes show sequences of identical values when realizations are repeatedly rejected. Like all MCMC methods, for a sufficiently large number of draws, this will not be a problem.

- As with the Gibbs sampler, S_0 burn-in replications are disregarded to avoid initialization problems.
- Choosing a good candidate generating density is very important: otherwise virtually all the candidate draws will be rejected and it will take a very large number of replications to get a valid set of draws.
- As with the Gibbs sampler, it is important to check that the chain has converged and that we have a sufficient number of draws to overcome the non-iid'ness of the draws. We will see shortly what these diagnostics look like.
- There are many strategies for selecting the candidate generating density $q(\theta^*|\theta^{s-1})$. Koop has good references and we discuss here a few of the more common.

1.4.1 Common Strategies to Choose a Candidate Generating Density

The Independence Chain Metropolis-Hastings Algorithm Here the candidate density is chosen such that $q(\theta^*|\theta^{s-1}) = q(\theta^*)$. Under this choice, the acceptance probability simplifies to

$$\alpha(\theta^*, \theta^{s-1}) = \min \left[\frac{p(\theta^*|y)q(\theta^{s-1})}{p(\theta^{s-1}|y)q(\theta^*)}, 1 \right]$$

It is easy to establish how Independence Chain M-H relates to importance sampling by recalling that the importance sampling weights are

$$w(\theta = \theta^A) = \frac{p(\theta^A|y)}{q(\theta^A)}; \theta^A = \theta^*, \theta^{s-1}$$

so that the acceptance probability is

$$\alpha(\theta^*, \theta^{s-1}) = \min \left[\frac{w(\theta^*)}{w(\theta^{s-1})}, 1 \right]$$

Remarks

- Choosing $q(\theta^*|\theta^{s-1}) = q(\theta^*)$ still requires choosing a specific density from which to draw from. One approach is to rely on asymptotic results and use the normal approximation with mean and variance given by the MLE results. i.e. $q(\theta^*)$ is the $N(\hat{\theta}_{ML}, \hat{V}[\hat{\theta}_{ML}])$ density.
- In practice, it is customary to choose a t-distribution instead with low degrees of freedom so as to have a density with fatter tails. The acceptance probability will ensure that the right draws are included for those extreme events at the tails.
- Naturally, if we suspect the posterior density is defined on a limited range (such as the Gamma) or it is multimodal, it is best to use a $q(\theta^*)$ that reflects these properties. Otherwise, we may need too many replications.

Random Walk Chain Metropolis-Hastings Algorithm When it is difficult to recognize what a good approximating density might be, it is useful to choose a candidate generating density that wanders widely as the acceptance probability will retain all the draws that are valid and disregard the remainder. Accordingly, draws are generated by

$$\theta^* = \theta^{s-1} + z$$

where z is the random variable from which draws will be taken. In this particular M-H method, $q(\theta^*|\theta^{s-1}) = q(\theta^{s-1}|\theta = \theta^*)$ and hence the acceptance probability is the same as for the Metropolis algorithm, that is

$$\alpha(\theta^*, \theta^{s-1}) = \min \left[\frac{p(\theta^*|y)}{p(\theta^{s-1}|y)}, 1 \right]$$

Remarks

- It is common to use the multivariate normal as the candidate generating density of z , with mean zero, and variance covariance matrix Σ , determined by either using the MLE variance-covariance estimate (if it is easily available) or by a method that ensures that the acceptance probability is in the range 0.45 (for a one parameter problem) to 0.23 (for a high-dimensional problem).

- One method to figure out the variance-covariance matrix that will achieve these acceptance probabilities is to follow the following algorithm
 1. Set $\Sigma = cI$ and experiment with several values to c until you find one that gives acceptance probabilities in the desired range
 2. Use these draws to compute a crude estimate of Ω and set $\Sigma = c\Omega$.
 3. Try several values of c to get the acceptance probabilities in the desired range.
 4. Go to step 2.

Metropolis-within-Gibbs In practice it is often the case that some blocks of the parameter vector θ have analytic (or easy to guess) posteriors relative to the other blocks. It is acceptable to break the problem of drawing from the joint posterior of θ with the Gibbs sampler and use the Metropolis-Hastings algorithm to draw from the conditional posteriors of the sub-blocks of θ . For example, in the nonlinear regression example in chapter 5 of Koop's book, while the conditional posterior $p(\beta|h, y)$ is not available (and hence requires Metropolis-Hastings for its simulation), the conditional posterior $p(h|\beta, y)$ is still a Gamma.

1.5 MCMC Diagnostics - Koop pp. 64-68

MCMC methods rely on results that ensure that the Markov chain has a stationary distribution to which it converges to so that, even though we are obtaining draws that are correlated, in the limit these draws represent i.i.d. draws from the stationary density. For this reason, MCMC methods require both more draws than traditional Monte Carlo simulation, and diagnostics that ensure that the Markov chain is close to its stationary density. The following set of diagnostics is used to assess these properties and should be used on any of the MCMC methods described above.

1.5.1 Geweke's convergence diagnostic (CD)

The intuition of this statistic is similar to a Chow test. In a Chow test we can compare whether there are breaks in the estimate of a parameter by comparing the parameter estimates over two subsamples. Here, we split the sample

of replications and compare the Monte Carlo averages for each subsample. Hence, suppose we have S_0 burn-in replications and S_1 replications that are being used to calculate $\hat{g}(\theta)$ with Monte Carlo integration (see (1)).

- Then divide the S_1 replications into three groups, such that $S_1 = S_A + S_B + S_C$. It is recommended to set S_A as the first 10% of S_1 replications, S_B as the second 50% of S_1 replications and S_C as the remaining 40% of S_1 replications.
- Compute \hat{g}_A and \hat{g}_C using the S_A and S_C replications respectively.
- Compute the associated standard errors for \hat{g}_A and \hat{g}_C . Because the draws are correlated in any MCMC method, it is recommended that an autocorrelation robust standard error be computed. A Newey-West standard error seems a good option.

- Then the statistic

$$\frac{\hat{g}_A - \hat{g}_C}{\frac{\hat{\sigma}_A}{\sqrt{S_A}} - \frac{\hat{\sigma}_C}{\sqrt{S_C}}} \rightarrow N(0, 1)$$

under the null that the algorithm has converged.

Remarks

- The CD statistic can fail if the posterior is bimodal.

1.5.2 The Gelman-Rubin Statistic

Consider starting your MCMC algorithm with a wide array of starting values. If the MCMC algorithm has reached convergence, then the effects of the starting value should have vanished and the variability of the draws within run (or sequence) should be similar to the variability of the draws across runs. The Gelman-Rubin statistic exploits this observation. Hence, consider experimenting with m chains that are started with initial values $\theta^{0,i}$ $i = 1, \dots, m$ that are taken from different regions of the parameter space. For the run based on the i^{th} starting value, the variance of the draws within the run can be calculated as

$$\hat{\sigma}_i^2 = \frac{1}{S_1 - 1} \sum_{s=S_0+1}^{S_0+S_1} [g_{i,s} - \hat{g}^i]^2; \quad i = 1, \dots, m$$

where \hat{g}^i is calculated as usual from (1). The average within-run variance is simply

$$W = \frac{1}{m} \sum_{i=1}^m \hat{\sigma}_i^2$$

The between-runs variance can be estimated as

$$B = \frac{S_1}{m-1} \sum_{i=1}^m (\hat{g}^i - \bar{g})^2$$

where

$$\bar{g} = \frac{1}{m} \sum_{i=1}^m \hat{g}^i$$

Hence, the diagnostic

$$R = \frac{S_1 - 1}{S_1} + \frac{1}{S_1} \frac{B}{W}$$

tends to one when the chain has converged (the between-runs variance B and the average within-runs variance are the same) and will be larger than one if it has not. There is no distribution associated with this statistic and a common rule of thumb is to reject convergence when R is bigger than 1.2.

1.6 Using Samples from the posterior

Suppose you have draws on two parameters, θ_1^s, θ_2^s that you have obtained from your favorite MCMC algorithm. A natural estimate of the posterior mean of, say θ_2 is

$$\bar{\theta}_2 = \frac{1}{S_1} \sum_{s=S_0+1}^{S_0+S_1} \theta_2^s$$

It turns out that if the conditional expectation of $\theta_2|\theta_1$ is a known function $\phi(\theta_1^s, \theta_2^s)$ (as it may be if you are using the Gibbs sampler, say), then an estimate of the posterior mean based on

$$\tilde{\theta}_2 = \frac{1}{S_1} \sum_{s=S_0+1}^{S_0+S_1} \phi(\theta_1^s, \theta_2^s)$$

is a more efficient estimate of the posterior mean of θ_2 . To see the intuition behind this result, notice that in linear regression $Y = X\beta + \varepsilon$, $V[Y] \geq V[X\beta]$ and that the law of iterated expectations tells us that $E[E[Y|X]] = E[Y]$.

A similar result exists for posterior densities called Rao-Blackwell density estimates. Assuming $p(\theta_2|\theta_1)$ is available (as it would be if you are using the Gibbs sampler), then

$$p(\theta_2) = \int p(\theta_1, \theta_2) d\theta_1 = \int p(\theta_2|\theta_1)p(\theta_1) d\theta_1 = E[p(\theta_2|\theta_1)].$$