Using Reversible Jump MCMC to Account for Model Uncertainty

Brian M. Hartman and Jeffrey D. Hart

Texas A&M University

Abstract

When fitting a model to any data, there is some uncertainty about which model is best. Green (1995) quantifies this uncertainty through the Reversible Jump Markov Chain Monte Carlo (RJMCMC) method. When using the RJMCMC method in a regime-switching situation, the chain determines the optimal number of regimes by jumping between various possibilities. This method gives each model its posterior probability of being the best. After an overview of the methodology, we apply it to various datasets and discuss the applications in modern actuarial science.

1 Introduction

Regime-switching models are widely applicable. How can one determine the optimal number of regimes? Currently, there are many methods for determining which model is best (AIC, BIC, r^2_{adj}, etc.) or if one model is significantly better than another (Full vs. Reduced F Tests, etc.). Wouldn't it be great if there was a method which gives the probability that a certain model is best? The Reversible Jump Markov Chain Monte Carlo gives us exactly that.

There are two major advantages to having actual probabilities. First, we are able to better explain our results. Under the other methods we would just have to say that the model we chose is the best. We cannot really say how sure we are of that statement. With probabilities we can say, 'There is a 95% chance that model x is the best.' Clients outside of modeling can clearly understand that statement. Additionally, we can include the probabilities in our results through Bayesian model averaging. If we are interested in the conditional tail expectation (CTE) of a certain project 20 years in the future and we find that the cash flows follow model X with probability 0.8 and model Y with probability 0.2, we can run 800 simulations of model X and 200 simulations of model Y to get a prediction of the distribution of the 20 year CTE.

2 Regime-Switching Lognormal Model

Regime-switching models are very flexible. They consist of an underlying Markov process which switches between states or regimes at random. The Markov property implies that the distribution of any state is only dependent upon the state immediately before it. This process determines the distribution of the variable of interest. For example, suppose one is interested in the distribution of auto accidents. The number of accidents could be modeled with two different Poisson(λ) distributions. One could have a small λ for clear weather and the other a larger λ for when it rains. The weather would be the underlying Markov process (assuming the weather today is only dependent upon the weather yesterday).

In the previous example, one is able to observe the value of the Markov process at each time point. Often, one is unable to observe those values. What if the distribution of accidents is determined by the current state of the world? When the world is tumultuous or unsteady, people drive more aggressively and accidents are more common. There is no way to observe if the world is currently in a bad or good state. The process becomes a hidden Markov process as described in Rabiner and Juang (1986).

In financial modeling, a popular model is the lognormal model. Hull (2006) states that, the price of a stock at time T, under certain assumptions and given its current price, is lognormally distributed. We can expand the lognormal model using the regime-switching framework. We assume that the parameters μ and σ are determined by the Markov process. Following the suggestion in Hardy (2003), the underlying process could be the state of the economy. The two-regime case can have one lognormal distribution with a high mean and low variance for when the economy is good, and one with a low mean and high variance for when the economy is bad. This is a hidden process.

By keeping this Markov process hidden we do not need to specify what each regime means. While the good/bad economy scenario is logical, the ability of the model to make predictions is not adversely affected if it is not true.

3 Reversible Jump Methodology

To find the distribution of parameters in a model, we can use Bayes rule,

$$P(\boldsymbol{\theta}|\mathbf{y}) = \frac{\mathbf{f}(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\mathbf{P}(\mathbf{y})}.$$

It seems to be rather simple, but let us expand it:

$$P(\mathbf{y}) = \int \mathbf{f}(\mathbf{y}|\boldsymbol{\tau})\pi(\boldsymbol{\tau}) \Rightarrow \mathbf{f}(\boldsymbol{\theta}|\mathbf{y}) = \frac{\mathbf{f}(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int \mathbf{f}(\mathbf{y}|\boldsymbol{\tau})\pi(\boldsymbol{\tau})}.$$

Often, the integral in the denominator does not have a closed form solution. Markov Chain Monte Carlo (MCMC) methods were developed to allow one to draw samples from the posterior distribution of the parameters. If enough samples are drawn, a good approximation of the posterior can be obtained. Unfortunately, the MCMC framework only works in models with a constant number of parameters. The reversible jump methodology allows one to work with likelihoods of varying dimension.

There are six basic steps in the RJMCMC algorithm outlined in Waagepetersen and Sorensen (2001).

1. Select a starting value $X_1 = (M_1, Z_1)$ where M_i is the model index at iteration i and Z_i is the parameter vector of length n_{m_i}.

2. Generate a proposal value X_p.

3. Satisfy the reversibility and dimension matching conditions.

4. Calculate the acceptance probability.

5. If accepted $X_2 = X_p$, otherwise $X_2 = X_1$.

6. Repeat steps 2-5.

Notice that most of the steps are identical to MCMC methods. The main difference is that in each iteration a *model* is proposed along with its parameters.

3.1 Selecting a Starting Value

While this may seem like a difficult task, other estimation methods (MLE, MOM) can be used to find suitable starting values. Information from other experts can also be used. Luckily, even if the starting value is poor, the algorithm will eventually produce acceptable values.

3.2 Generating a Proposal Value

We need to generate $X_p = (m_p, z_p)$. The parameter vector z_p is generated by applying a deterministic mapping to the previous z and to a random component U. We can express it as $z_p = g_{mm_p}(z, U)$, where U is a random vector on $\mathbb{R}^{n_{mm_p}}, n_{mm_p} \geq 1$, which has density $q_{mm_p}(z, \cdot)$ on $\mathbb{R}^{n_{mm_p}}$, and $g_{mm_p} : \mathbb{R}^{n_m + n_{mm_p}} \rightarrow \mathbb{R}^{n_{m_p}}$ is a deterministic mapping.

Arguably, the most difficult task in setting up an RJMCMC algorithm is trying to specify the function that determines the proposal value (for a more comprehensive treatment of the possibilities, please see Brooks et al. (2003)). We will divide the methodology into two broad categories, independent and dependent proposals.

3.2.1 Dependent Proposals

With dependent proposals, the values in the current model are related to the values in the proposal model. For example, if the current model is an exponential distribution with parameter θ and the proposed distribution is the $Normal(\mu, \sigma^2)$. The first few moments of the two distributions could be matched. The mean of an exponential is θ and the variance is θ^2. The proposal values could be $(\mu_p, \sigma_p^2) = (\theta + y_1, \theta^2 + y_2)$ where y_i are $Normal(0, \tau^2)$ random variables. When going from the normal model to the exponential model, let $\theta_p = y_3 + (\mu + \sqrt{\sigma^2})/2$.

This method works well in the simplified example outlined above, but there is a problem when it is applied to an RSLN model. When proposing a model with more regimes than the current model, there are an infinite number of parameter values which will have the same mean and variance. If one tries to match higher moments as well, the more complex model will be unduly restricted. To choose which set of parameters to use, one needs to make some assumptions about the model. Unfortunately, the effectiveness of the RJMCMC sampler is highly dependent upon those assumptions. Therefore, we recommend not using dependent proposals.

3.2.2 Independent Proposals

While we often try to stay away from maximum likelihood estimates (MLEs) in Bayesian analysis, they do have some nice properties. For our purposes, we can exploit the fact that MLEs are asymptotically normal. For each model, we find the MLEs through numerical optimization. We also find a numerically estimated Hessian matrix, $\hat{H}$. Then, for each new parameter draw, we draw from a multivariate normal distribution with its mean equal to the parameter estimates and the covariance matrix equal to $-\hat{H}^{-1}$. In this way, the parameter values in the proposed model are not related to the parameter values in the current model. It is unnecessary to define functions to compare the parameter values between models. This method also simplifies many of the forthcoming steps.

The major drawback to this method is computational instability. In working with regime-switching models, the MLE of a switching probability could be 0 (when we have specified too many regimes). Since this value is on the boundary, the numerical estimate of the Hessian matrix can be very poor, or not even positive definite. Luckily, an MLE of 0 implies the model has too many regimes and we can likely disregard it.

3.3 Necessary Conditions

There are two main conditions which need to be satisfied for the chain to converge properly. The first is the condition of reversibility, which is stated as:

$$P(M_n = m, Z_n \in A_m, M_{n+1} = m_p, Z_{n+1} \in B_{m_p}) = P(M_n = m_p, Z_n \in B_{m_p}, M_{n+1} = m, Z_{n+1} \in A_m)$$

This is just a complicated way of saying that the proposal function must be invertible. This inverse function makes it possible to move from the proposed parameters back to the current parameters. In practice, this condition is almost always satisfied.

The other condition is dimension matching, and follows from the condition above:

$$n_m + n_{mm_p} = n_{m_p} + n_{m_pm}$$

This states that the number of parameters in the current model plus the number of random elements necessary to move from your current model to the proposed model must equal the number of parameters in your proposed model plus the number of random elements required to move from your proposed model back to the current model. This ensures that $f_m(z)q_{mm_p}(z,u)$ and $f_{m_p}(z_p)q_{m_pm}(z_p,u_p)$ are joint densities on spaces of equal dimension. Note that $f_m(z)$ is the likelihood of the data under model m with parameters z and $q_{mm_p}(z,u)$ is the proposal density, moving from model m to m_p with current parameters z and random elements u.

Using the independent proposals described above ensures that both conditions are satisfied. The reversibility condition is satisfied because the proposal distribution is not dependent upon the previous parameter values. The dimension matching condition is satisfied because $n_{m_p m} = n_m$ and $n_{mm_p} = n_{m_p}$.

3.4 Acceptance Probability

The general formula for the acceptance probability is

$$\alpha_{mm_p} = min\left(1, \frac{p_{m_p} f_{m_p}(z_p) p_{m_p m} q_{m_p m}(z_p, u_p)}{p_m f_m(z) p_{mm_p} q_{mm_p}(z, u)} \left| \frac{\partial g_{mm_p}(z, u)}{\partial z \partial u} \right| \right).$$

When using independent proposals it is easy to check that the Jacobian at the end is equal to 1. Therefore, the acceptance probability is

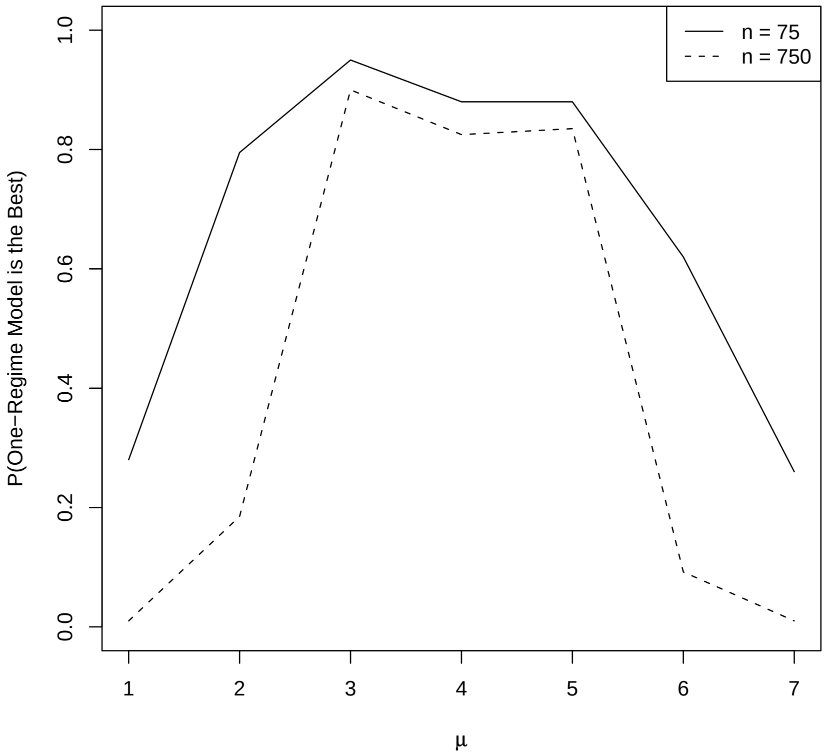
$$\alpha_{mm_p} = min\left(1, \frac{p_{m_p} f_{m_p}(z_p) p_{m_p m} q_{m_p m}(z_p, u_p)}{p_m f_m(z) p_{mm_p} q_{mm_p}(z, u)}\right)$$

A Uniform(0,1) random variable U is generated and if $U \leq \alpha_{mm_p}$ the proposed move is accepted. If $U > \alpha_{mm_p}$, the chain remains in the current state. This process is continued until the desired number of draws is obtained.

4 Simulation Study

Here we investigate whether the algorithm proposed in section 2 works well in a practical setting. In a first simulation we set $\mu_2 = 4$, $\sigma_1 = \sigma_2 = 1$, $p_{12} = 0.3$, and $p_{21} = 0.7$, and allowed μ_1 to vary from 1 to 7. We generated 200 samples of size 75 at each of the values for μ_1. For each sample, we ran 11,000 iterations of the RJMCMC chain (1,000 for burn-in and 10,000 for analysis). We recorded the posterior probability of the one-regime model being the best obtained from the final 10,000 iterations. We repeated the process with samples of size 750. The results are described on the top of the next page in figure 1.

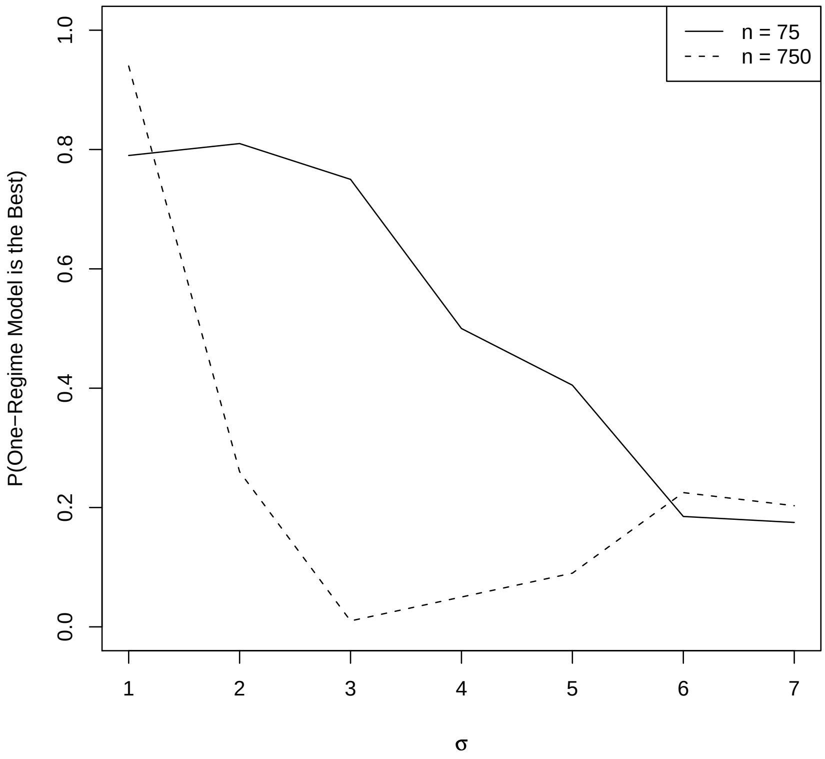
Figure 1
Sensitivity to Changes in the Mean



The method seems to do a reasonably good job. When $\mu_1 = \mu_2 = 4$ we would expect the probability of the one-regime model being the best to be high. As μ_1 moves in either direction the probability drops quickly. Also, as expected, the probability drops quicker with the larger sample size. This makes sense because larger samples contain more information about the underlying distribution and can detect smaller differences in mean values.

The method does a good job detecting differences in means, but it will be more difficult to detect differences in standard deviations. For this simulation, we set $\mu_1 = \mu_2 = 4$, $\sigma_2 = 1$, $p_{12} = 0.3$, and $p_{21} = 0.7$, and let σ_1 vary from 1 to 7. The results are described on the top of the next page in figure 2.

Figure 2
Sensitivity to Changes in the Standard Deviation



The method performed very well. There is a definite negative relationship as σ_1 grows farther from $\sigma_2 = 1$. The larger sample size greatly outperformed the smaller size.

5 S&P 500 Application

Now we can apply the algorithm to some real data. We obtained total return data on the S&P 500 Index from January 1991 to March 2008 from Standard and Poor's (2008). We started with the one-, two- and three-regime RSLN models. When we tried to optimize the parameters in the three-regime model, we were unable to get a sensible Hessian matrix. When looking at the MLEs of the parameters, we noticed that the third regime had an unconditional probability of zero. The probability of moving to the third regime from the first or second regime is zero and the probability of moving from the third regime is 1. The regime will never occur, making it a two-regime model. We removed that option from the analysis and compared the one-regime model to the two-regime model. The posterior probability that the two-regime model is very close to one because a one-regime model was never accepted in the 10,000 iterations.

6 Future Work

We would like to do a few things to improve upon this project. First, we would like to improve the stability of the independent proposals. This may be done by changing the proposal method or improving the Hessian estimation method. Second, we would like to include other models (ARCH, GARCH, SV, etc.). Finally, we would like to try some other methods, including those of Chib and Jeliazkov (2001) and Phillips and Smith (1996).

References

Brooks, S. P., P. Guidici, and G. O. Roberts (2003). Efficient construction of reversible jump markov chain monte carlo proposal distributions. *J. R. Statist. Soc. B 65*(1), 3–55.

Casella, G. and E. I. George (1992). Explaining the gibbs sampler. *American Statistician 46*, 167–174.

Chib, S. and E. Greenberg (1995). Understanding the metropolis-hastings algorithm. *American Statistician 49*, 327–335.

Chib, S. and I. Jeliazkov (2001, March). Marginal likelihood from the metropolis-hastings output. *Journal of the American Statistical Association 96*(453), 270–281.

Green, P. J. (1995, December). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika 82*(4), 711–732.

Hardy, M. (2003). *Investment Guarantees: Modeling and Risk Management for Equity Linked Life Insurance*. John Wiley and Sons.

Hastings, W. K. (1970). Monte carlo methods using markov chains and their applications. *Biometrika 57*, 97–109.

Hull, J. C. (2006). *Options, Futures and Other Derivatives* (6 ed.). Prentice Hall.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics 21*, 1087–1091.

Phillips, D. B. and A. F. M. Smith (1996). Bayesian model comparison via jump diffusions. In D. J. S. Walter R. Gilks, Sylvia Richardson (Ed.), *Markov Chain Monte Carlo in Practice: Interdisciplinary Statistics*, pp. 215–240. CRC Press.

Rabiner, L. and B. Juang (1986, Jan). An introduction to hidden markov models. *ASSP Magazine 3*(1), 4–16.

Standard and Poor's (2008). `http://www2.standardandpoors.com/spf/xls/index/MONTHLY.xls`.

Waagepetersen, R. and D. Sorensen (2001). A tutorial on reversible jump mcmc with a view toward applications in qtl-mapping. *Iternational Statistical Review 69*(1), 49–61.