Institut de la finance structurée et des instruments dérivés de Montréal
Montreal Institute of Structured Finance and Derivatives

L'Institut bénéficie du soutien financier de l'Autorité des marchés financiers ainsi que du ministère des Finances du Québec

Note technique

NT 13-02

Bayesian Analysis of Consumer Default Using Reversible Jump MCMC

Juillet 2013

Cette note technique a été rédigée par Philippe d'Astous

Bayesian Analysis of Consumer Default Using Reversible Jump MCMC*

Philippe d'Astous†

July 25, 2013

Abstract

I present Bayesian inference and model selection through a detailed example of the Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm, using a count data application. I use an adaptive Metropolis-Hastings algorithm in conjunction with RJMCMC to improve parameter convergence. After reviewing the Bayesian paradigm and the theory underlying model selection and the RJMCMC algorithm, I implement the technique on a bank-level dataset of consumer loan default realizations.

Keywords. Bayesian Methods, Reversible Jump Markov Chain Monte Carlo, Adaptive Metropolis-Hastings, Count data, Hurdle model, Consumer default.

In this note, I am interested in modeling payment default realizations on consumer loans using a cross-sectional dataset provided by a Canadian institution. The dependent variable used in the analysis is the number of missed payments on the loan, which is the count variable of interest. Each missed payment increases the count variable by one. My focus is on model selection; I propose different specifications and I use Bayesian model selection to calculate posterior probabilities of the competing models.

Count data models have a wide range of applications in economics, finance and insurance. Among many examples, such models have been used in health economics for the number of hospital visits in a year, in insurance for the number of accidents or claims filed during a given period and in finance for the number of late payments on a loan. The Poisson regression has been used extensively to model such variables. It is a convenient underlying distribution for non-negative integer stochastic processes but its well known equidispersion property is often too restrictive for the observed data.

*I am grateful to Richard Luger and Rusty Tchernis for helpful discussions and to Georges Dionne for providing the dataset used. I acknowledge financial support from SSHRC. Comments are welcome.

†Risk Management and Insurance Department, J. Mack Robinson College of Business, Georgia State University. Contact: pdastous1@gsu.edu

In a seminal paper, Cameron and Trivedi (1986) present many extensions to the Poisson regression model, allowing to relax some of its strong assumptions. For example, a more flexible functional form can be obtained by adding a random component to the individual means, increasing heterogeneity across observations. Assuming that this random component follows a Gamma distribution leads to a Negative Binomial density function for the dependent variable, which allows for overdispersion in the data. One can also cope with clusters of observations at certain realizations (for example, excess zeros) or truncation by using hurdle models and truncated distributions (see Dionne and Vanasse (1989) and Dionne, Artís, and Guillén (1996) for some applications).

With these different modeling possibilities, it becomes interesting to compare the estimation performance using different assumptions for the underlying population distribution. In the classical approach to model selection, competing models are estimated independently and are compared using different measures and statistics (for example, Adjusted R^2, Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) or Likelihood Ratio tests). Several researchers have challenged this approach on different grounds (see, for example, Hansen (2005)). One of the argument against adopting the classical approach is that it neglects model uncertainty, that is, the probability that a given model is the right one. Furthermore, comparing non-linear or non-nested models complicates the analysis (see, for example, Vuong (1989)).

Another option for this problem is to adopt a Bayesian framework where it is straightforward to extend the inference to account for model uncertainty and non-nested models. Using the posterior marginal likelihood, one can compute Bayes factors and posterior model probabilities which can, in turn, be used as a model selection criteria or in Bayesian model averaging. I thus use Markov Chain Monte Carlo (MCMC) methods to jointly estimate the uncertainty about parameters and competing model probabilities by extending the state space of the Markov Chain to allow for a model indicator. This technique, which was first introduced by Green (1995) under the term Reversible Jump Markov Chain Monte Carlo (RJMCMC), allows direct inferential access to the posterior model probabilities. Estimates of the marginal likelihoods of each model are directly obtained after a run of the algorithm, whereas independent estimation of the models by MCMC usually requires additional numerical computations.

The RJMCMC algorithm should be seen as a generalization of the Metropolis-Hastings (MH) algorithm. In the standard MH algorithm, the objective is to construct a Markov Chain which has as its invariant distribution the posterior of the parameters of interest. In doing so, the MH algorithm sweeps through the parameter space by proposing a candidate value for the parameter at each iteration and accepting it with a probability derived using

the detailed balance condition. In RJMCMC, the estimation of the parameters *within* each model follows the same procedure. The major difference comes from an additional step in the algorithm allowing to jump *across* different competing models during the computation. The proof of convergence of RJMCMC is surprisingly similar to the one used in the basic MH procedure. Both algorithms rely on the detailed balance condition but in the case of Reversible Jump, one needs to consider a slightly generalized state space for the Markov Chain.

Since the seminal paper of Green (1995), researchers have been active in applying trans-dimensional Markov chains to various problems.[1] According to Hastie and Green (2012),

> "[...] around 45% of the articles citing Green (1995) are in statistics and probability, about 28% in biology, genetics and medicine [...], about 20% in computer science and engineering and about 15% in other disciplines ranging from archaeometry through management science to water resources research."

Some applications in business are in finance (*e.g.,* GARCH model selection: Vrontos, Dellaportas, and Politis (2000); high-frequency data: Centanni and Minozzo (2006)) and in insurance (*e.g.,* claim count: Ntzoufras, Katsis, and Karlis (2005); aggregate loss: Ausín, Vilar, Cao, and González-Fragueiro (2011)).

I introduce a simple but rich example allowing to illustrate the technique and to highlight the importance of model selection in the context of consumer default. The count variable ranges from 0 to 6 and represents the number of missed payments on the loan. For example, a borrower who is less than 29 days late on his loan payment has a realized count of 0 since a billing cycle is assumed to be 30 days. The default status increases in the length of delinquency up to a value of 6, representing a borrower who has missed 6 payments on the loan, or equivalently, who is 180 days late on his loan payment.

I estimate the different count data models and their respective posterior probability for underlying distributions assumed to be either Poisson or Negative Binomial and allowing or not for an excess realization of zeros through hurdle models. I compare two pairs of models: the Poisson and Negative Binomial regression models and the Poisson Hurdle and Negative Binomial Hurdle regression models. The Negative Binomial specification allows for overdispersion in the data (as compared to the Poisson model) but it does not specifically take into account a possible excess realization of zeros, which is why the hurdle specifications are also considered.

I find that the Negative Binomial model has a higher posterior probability than the Poisson model when I do not account for excess zero realizations through hurdle models.

[1] See Sisson (2005) for a review on the recent progress on RJMCMC.

The additional parameter in the Negative Binomial model captures the overdispersion of the data which is very severe considering that 98.99% of the borrowers have a default status of 0. However, when I extend the analysis to hurdle models, the Poisson and Negative Binomial specifications have roughly the same estimated posterior model probabilities. This shows that once the zero realizations are modeled correctly, the residual overdispersion in the truncated count distribution is negligible. The results highlight the importance of the hurdle specification when there is a cluster of realizations around a realization of the count variable.

The structure of the paper is as follows. In Section 1, I present the different count data models used in the analysis. In Section 2, I introduce the Bayesian paradigm and model selection. In Section 3, I present the theory underlying the RJMCMC algorithm. In section 4, I detail the algorithm used in the loan repayment application. In section 5, I present the data and the implementation. In Section 6, I give the results and I conclude in Section 7.

1 Count Data Models

I briefly introduce the two classical count data models used in the analysis with their hurdle extensions, and I refer the reader to Cameron and Trivedi (1998) for an extensive discussion on count data regression models. The likelihood under the assumption that the count variable is independent and identically distributed following a Poisson model of parameter λ is

$$L^P(\boldsymbol{Y}|\boldsymbol{\lambda}) = \prod_{i=1}^{n} \frac{\lambda_i^{Y_i}}{Y_i!} \exp(-\lambda_i).$$

Introducing covariates is done through the parametrization of the conditional mean by

$$\mathbb{E}[Y_i|\boldsymbol{X}_i] = \lambda_i = \exp(\boldsymbol{X}_i'\boldsymbol{\beta}), \tag{1}$$

where $\boldsymbol{X}_i$ is the vector of covariates and $\boldsymbol{\beta}$ is its associated vector of parameters. Hence, conditioning on $\boldsymbol{\lambda}$ also implies conditioning on $\boldsymbol{X}$ in what follows. As it is well known, for this distribution the mean and variance are both given by λ. However, many datasets used in practice do not satisfy this equidispersion property. Among extensions proposed in the literature, a convenient way to allow for more flexibility is to introduce a random component in the expected mean such that

$$\mathbb{E}[Y_i|\boldsymbol{X}_i] = \lambda_i \epsilon_i.$$

Under this specification, the mean is derived from a deterministic function, λ_i but has a random component, ϵ_i, allowing for increased heterogeneity across observations. Note that for $\mathbb{E}[\epsilon_i] = 1$, the conditional mean remains equal to λ_i. When ϵ_i follows a Gamma distribution, the density of the data can be shown to be Negative Binomial (see Cameron and Trivedi (1998) for the derivation). The likelihood under the assumption that the count variable is independent and identically distributed following a Negative Binomial model of parameters λ and κ is then

$$L^{NB}(\boldsymbol{Y}|\boldsymbol{\lambda}, \kappa) = \prod_{i=1}^{n} \frac{\lambda_i^{Y_i}}{Y_i!} \frac{\Gamma(1/\kappa + Y_i)}{\Gamma(1/\kappa)(1/\kappa + \lambda_i)^{Y_i}} (1 + \kappa\lambda_i)^{-1/\kappa}.$$

It can be shown that the mean and variance of this distribution are respectively given by λ and $\lambda(1 + \kappa\lambda)$. The model has $\kappa > 0$ as a parametric restriction. Covariates are introduced through the mean parameter, as in equation (1).

Both the Poisson and Negative Binomial models can be extended to allow for an excess realization of zeros (as compared to that predicted by the theoretical distribution). In the hurdle extension, the idea is to model the zero realizations with a binary outcome model and to model the strictly positive realizations with a truncated distribution. The number of parameters are thus doubled. I normalize $\kappa = 0$ in the binary outcome model so that, in both cases, the probabilities of a zero realization and of a strictly positive realization are given by

$$\begin{aligned} \mathbb{P}[Y_i = 0|\boldsymbol{X}_i] &= (1 + \lambda_i^H)^{-1} \\ 1 - \mathbb{P}[Y_i = 0|\boldsymbol{X}_i] &= 1 - (1 + \lambda_i^H)^{-1}. \end{aligned}$$

These probabilities can be derived directly from the Poisson distribution and they are equivalent to a logit model in classical estimation. The likelihood for the hurdle part of the model is given by

$$L_1^H(\boldsymbol{Y}|\boldsymbol{\lambda}^H) = \prod_{i=1}^{n} \mathbb{P}[Y_i = 0|\boldsymbol{X}_i]^{(1-\mathbb{I}_{y_i})} (1 - \mathbb{P}[Y_i = 0|\boldsymbol{X}_i])^{\mathbb{I}_{y_i}},$$

where $\mathbb{I}_{y_i}$ is the indicator that $Y_i > 0$. When $Y_i > 0$, the count variable is modeled with truncated-at-zero Poisson or Negative Binomial distributions, where

$$L_2^P(\boldsymbol{Y}|Y_i > 0, \boldsymbol{\lambda}) = \prod_{i=1}^{n} \frac{\exp^{-\exp(\lambda_i)} \exp(\lambda_i Y_i)}{Y_i![1 - \exp^{-\exp(\lambda_i)}]},$$

is the likelihood for the truncated-at-zero Poisson, and,

$$L_2^{NB}(\boldsymbol{Y}|Y_i > 0, \boldsymbol{\lambda}, \kappa) = \prod_{i=1}^{n} \frac{\lambda_i^{Y_i}}{Y_i!} \frac{\Gamma(1/\kappa + Y_i)}{\Gamma(1/\kappa)(1/\kappa + \lambda_i)^{Y_i}} \left((1 + \lambda_i \kappa)^{-1/\kappa} - 1\right)^{-1},$$

is the likelihood for the truncated-at-zero Negative Binomial. For a given model, the likelihood is then the product of the hurdle part and the truncated distribution part. For example, the likelihood of the Poisson Hurdle model is given by,

$$L^{PH}(\boldsymbol{Y}|\boldsymbol{\lambda}) = L_1^H(\boldsymbol{Y}|\boldsymbol{\lambda}^H) L_2^P(\boldsymbol{Y}|Y_i > 0, \boldsymbol{\lambda}).$$

To implement these models in a Bayesian framework, priors on each parameters must be specified. The prior on κ is chosen to be

$$\pi(\kappa) = \frac{1}{\Gamma(\alpha_\kappa)\beta^{\alpha_\kappa}} \kappa^{\alpha_\kappa - 1} \exp\left(\frac{-\kappa}{\beta_\kappa}\right), \tag{2}$$

following Hastie and Green (2012), that is, Gamma distributed of parameters $(\alpha_\kappa, \beta_\kappa)$. For the covariates, the prior on each of the β_j is chosen to be Normal distributed so that the joint prior density is given by

$$\pi(\boldsymbol{\beta}) = \prod_{j=0}^{J} \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left[-\frac{1}{2\sigma_j^2}(\beta_j - b_j)^2\right]. \tag{3}$$

In Bayesian econometrics, the prior density for the covariates is often chosen to be Normal since it insures conjugacy in the linear regression model. Although conjugacy is not a requirement when using the MH algorithm, I follow this standard.

2 Bayesian Model Selection

For a concise notation, let $\boldsymbol{\theta}$ represent the parameters of interest in the model under consideration (*i.e.*, for Poisson $\boldsymbol{\theta}_1 = \boldsymbol{\beta}_1$, whereas for Negative Binomial $\boldsymbol{\theta}_2 = (\boldsymbol{\beta}_2, \kappa)$). For a given model, the joint density of the vectors of model parameters $(\boldsymbol{\theta})$ and of observed data $(\boldsymbol{Y})$ can be written as

$$p(\boldsymbol{Y}, \boldsymbol{\theta}) = p(\boldsymbol{Y}) p(\boldsymbol{\theta}|\boldsymbol{Y}) = \pi(\boldsymbol{\theta}) L(\boldsymbol{Y}|\boldsymbol{\theta}).$$

This simple formulation provides a decomposition in which the prior density of the parameters, $\pi(\boldsymbol{\theta})$, their posterior density, $p(\boldsymbol{\theta}|\boldsymbol{Y})$, and the likelihood function, $L(\boldsymbol{Y}|\boldsymbol{\theta})$, are made

explicit. Rearranging, or equivalently, using Bayes' theorem,

$$p(\boldsymbol{\theta}|\boldsymbol{Y}) = \frac{\pi(\boldsymbol{\theta})L(\boldsymbol{Y}|\boldsymbol{\theta})}{p(\boldsymbol{Y})} \propto \pi(\boldsymbol{\theta})L(\boldsymbol{Y}|\boldsymbol{\theta}), \tag{4}$$

where the proportionality is useful since $p(\boldsymbol{Y})$ does not depend on model parameters. Under the Bayesian paradigm, inference on the parameters is then achieved through the posterior density, on which a Markov Chain can be constructed.[2]

2.1 Model Uncertainty

To introduce model uncertainty in the Bayesian framework, let $k = 1, 2, ...,$ be model indicators for each of the competing models, indexed in a countable set $\mathcal{K}$. Each model is assumed to have a parameter vector $\boldsymbol{\theta}_k$ of dimension n_k, possibly varying across models. In model selection, the interest is to treat $(k, \boldsymbol{\theta}_k)$ as a joint unknown, for which, using Baye's theorem, the posterior density can be written as

$$p(k, \boldsymbol{\theta}_k|\boldsymbol{Y}) = \frac{\pi(k, \boldsymbol{\theta}_k)L(\boldsymbol{Y}|k, \boldsymbol{\theta}_k)}{\sum_{k' \in \mathcal{K}} \int \pi(k', \boldsymbol{\theta}'_{k'})L(\boldsymbol{Y}|k', \boldsymbol{\theta}_{k'})d\boldsymbol{\theta}'_{k'}},$$

and can be factorised as

$$p(k, \boldsymbol{\theta}_k|\boldsymbol{Y}) = \underbrace{p(k|\boldsymbol{Y})}_{\text{Posterior model probability}} \quad \underbrace{p(\boldsymbol{\theta}_k|k, \boldsymbol{Y}).}_{\text{Model-specific parameter posterior}}$$

This factorisation allows to break down the estimation results in two parts: a posterior model probability and a model-specific parameter posterior. The marginal likelihood of model k is then derived as

$$p(\boldsymbol{Y}|k) = \int L(\boldsymbol{Y}|k, \boldsymbol{\theta}_k)\pi(\boldsymbol{\theta}_k|k)d\boldsymbol{\theta}_k. \tag{5}$$

Except for a few special cases, the marginal likelihood is not available in closed form solution and must be evaluated numerically via Monte Carlo simulations. It is a crucial part of model selection since it is used in the determination of posterior model probabilities. Specifically, the posterior model probability of model k is defined as

$$p(k|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|k)\pi(k)}{\sum_{k_i \in \mathcal{K}} p(\boldsymbol{Y}|k_i)\pi(k_i)}.$$

[2]See Casella and George (1992) and Chib and Greenberg (1995) for detailed presentations of the Gibbs sampling and Metropolis-Hastings algorithms, two of the most widely used inference methods in the Bayesian paradigm.

Posterior probabilities for each model considered could then be used in Bayesian averaging (see, for example Hoeting, Madigan, Raftery, and Volinsky (1999)).

In the current application, models are compared two by two with $k = 1, 2$ respectively representing the Poisson and Negative Binomial specifications (or their respective hurdle extension). The ratio of two competing models' posterior probabilities is the posterior model odds and is defined as

$$PO_{12} = \frac{p(k=1|\boldsymbol{Y})}{p(k=2|\boldsymbol{Y})} = BF_{12}\frac{\pi(k=1)}{\pi(k=2)},$$

where the Bayes factor is given by

$$BF_{12} = \frac{p(\boldsymbol{Y}|k=1)}{p(\boldsymbol{Y}|k=2)}.$$

When the prior model probabilities are the same across competing models, the Bayes factor is equal to the posterior odds. One way to compare the evidence of model 1 against model 2 is to compute either the Bayes factor or the posterior model odds of the two models. For example, high values of BF_{12} provide strong evidence for model 1 against model 2. As will be made clear, the RJMCMC algorithm allows to estimate the posterior model probabilities in a run of the algorithm without requiring additional numerical estimations.

3 Reversible Jump Algorithm

I present the Reversible Jump Markov Chain Monte Carlo algorithm following the notation and derivation used in Hastie and Green (2012). The algorithm was first introduced by Green (1995) and provides a convenient way to implement Bayesian model selection. The particularity of this technique, which makes it favorable to model selection, is that it provides a framework where Markov Chain Monte Carlo simulations can be built over more general state spaces than the traditional Metropolis-Hastings algorithm. In particular, it allows to compare models with a different number of parameters without additional computational burden. It also avoids the numerical calculation of the integral in the marginal likelihood of equation (5).

Following the intuition of MCMC, the objective is to construct a Markov chain that has as its invariant distribution the posterior of interest, that is $(k, \boldsymbol{\theta}_k) \sim p(k, \boldsymbol{\theta}_k|\boldsymbol{Y})$. Whereas the state space of the Markov chain usually consists in the state space of the parameter vector in the traditional MH algorithm, in RJMCMC one needs additionally to consider the

given model as part of the state. A given state is thus represented by

$$(k, \boldsymbol{\theta}_k) = (k, \boldsymbol{\theta}_{k,1}, \boldsymbol{\theta}_{k,2}, ..., \boldsymbol{\theta}_{k,n_k})$$

on a state space

$$\mathcal{X} = \cup_{k \in \mathcal{K}} \left(\{k\} \times \mathcal{X}_k \right),$$

where, for each k, $\mathcal{X}_k \subset \mathbb{R}^{n_k}$. This allows to have direct inferential access to the model indicator k in each state, which proves to be crucial in analysing the output of the algorithm. A typical RJMCMC algorithm has the following structure:

1. Randomize the initial parameter values, including the model state.

2. Within the current model, run a Metropolis-Hastings algorithm to update the parameters.

3. Propose a jump to a different model, accepted with a given probability.

4. If the jump is accepted, switch to the new model, if not, stay in the current model.

5. Redo steps 2 to 4 iteratively.

As the structure of the algorithm shows, implementation is closely related to the Metropolis-Hastings algorithm with the difference that a second kind of acceptance probability, as shown by Step 4, must be considered. An interesting feature of the technique is that the number of times that a model has been visited in a given run is an estimate of the posterior probability of the model. This allows the researcher to either select the most plausible model given the data or to use Bayesian model averaging in making predictions.

The key trick to achieve smooth transition across vectors of different dimensions is to construct a diffeomorphism transformation, that is, a transformation that allows to keep track of the same number of parameters in each state. That way, the dimension-jumping across models does not introduce any additional computational burden.

Consider for example an initial state x that has a different dimension than the proposed forward state x'. The idea is to generate r random numbers u from a density g at the current state and to use a deterministic function h mapping the current state to the proposed one, *i.e.,* $(x', u') = h(x, u)$. For example, let the dimensions of x, x', u and u' be n, n', r and r'. Assume that the function h maps the current x to the forward state x' whereas the reverse move is mapped by the inverse function h'. In this case, a diffeomorphism transformation is one in which $n + r = n' + r'$.

Let the across-model moves m be indexed in a countable set $\mathcal{M}$. Since the detailed balance condition needs ultimately to be achieved, a given move consists both of the forward

move $x \to x'$ and its reverse $x' \to x$. The parameter $j_m(x)$ exogeneously determines with what probability an across-model move will be attempted at state x. The "detailed balance condition" holds if and only if

$$\int p(x) j_m(x) g_m(u) \alpha_m(x, x') dx du = \int p(x') j_m(x') g_m(u') \alpha_m(x', x) dx' du'. \tag{6}$$

One should appreciate the similarity between condition (6) and the usual condition satisfying detailed balance in the MH algorithm. Solving this equation actually proceeds in the same way as the proof of convergence for the MH algorithm; α_m is set such as to force the equality for each move considered. The major difference is now that the solution to this equality involves a change of variable in the function h. The acceptance probability can be shown to be $\alpha_m(x, x') = \min\{1, A_m(x, x')\}$, where

$$A_m(x, x') = \frac{p(x')}{p(x)} \frac{j_m(x')}{j_m(x)} \frac{g'_m(u')}{g_m(u)} \left| \frac{\partial(\boldsymbol{\theta}'_{k'}, u')}{\partial(\boldsymbol{\theta}_k, u)} \right|,$$

and where the last term is the Jacobian of the transformation. With this probability, one can now derive all that is needed to implement the algorithm.

4 Algorithm

In this section, I give details on the procedure used to compare the different models. I compare models two by two, that is, I compare the Poisson model to the Negative Binomial model and I compare the Poisson hurdle model to the Negative Binomial Hurdle model. For each pair of models compared, I follow the same two-step procedure, akin to an adaptive MH procedure. I start by estimating each model individually to get a first estimate of the parameters before implementing the RJMCMC algorithm in the second step of the procedure. I detail the comparison between the Poisson and Negative Binomial regression models since the extension to the hurdle specifications is trivial when replacing the likelihood functions according to the derivations presented in Section 1.

In the first step, I run a *random walk* MH algorithm on each individual model for $r = 50,000$ iterations, from which I discard the first $b = 20,000$ iterations as burn-in. For a general parameter vector $\boldsymbol{\theta}$, this is done by iterating on the following:

1. At iteration r, generate a proposed $\boldsymbol{\theta}^r$ from $\boldsymbol{\theta}^r = \boldsymbol{\theta}^{r-1} + \epsilon$, where $\epsilon \sim N(0, \Omega)$ and $\boldsymbol{\theta}^{r-1}$ is the $(r-1)^{\text{th}}$ MCMC iterate of $\boldsymbol{\theta}$.

2. Accept $\boldsymbol{\theta}^r$ with probability $p = \min\left(1, \frac{p(\boldsymbol{\theta}^r)}{p(\boldsymbol{\theta}^{r-1})}\right)$, otherwise set $\boldsymbol{\theta}^r = \boldsymbol{\theta}^{r-1}$.

The function p is the one given by equation (4). Note that an additional accept/reject step is needed for the Negative Binomial estimation since $\kappa > 0$ is a parametric restriction of the model. Alternatively, the model could be reparameterized using $\log(\kappa)$.

In this first step, Ω is chosen to be diagonal, with variances tuned to obtain an acceptance rate between 25% and 45%. After discarding the first 20,000 iterations, I estimate the sample mean μ_θ and the sample covariance matrix Ω_θ of the chain. This gives an accurate picture of the parameters and their interaction, which I then use in the second step of the procedure.

In the second step, I run the RJMCMC algorithm, using an *independent* MH algorithm for the *within*-model part of the algorithm, using the calibrated values of μ_θ and Ω_θ. Before presenting the *across*-model part of the algorithm, note that for a general parameter vector $\boldsymbol{\theta}$, the *within*-model independent MH algorithm is done by iterating on the following:

1. At iteration r, generate a proposed $\boldsymbol{\theta}^r$ from $\boldsymbol{\theta}^r = \mu_\theta + \epsilon$, where $\epsilon \sim N(0, \Omega_\theta)$.
2. Accept $\boldsymbol{\theta}^r$ with probability $p = \min\left(1, \frac{p(\boldsymbol{\theta}^r)g(\boldsymbol{\theta}^{r-1})}{p(\boldsymbol{\theta}^{r-1})g(\boldsymbol{\theta}^r)}\right)$, where g is the Normal density, otherwise set $\boldsymbol{\theta}^r = \boldsymbol{\theta}^{r-1}$.

Note how the proposed parameter values are generated using the estimates from the first step of the procedure (μ_θ and Ω_θ). The main advantage is a better convergence of the parameters and thus a more accurate comparison of both models in the RJMCMC algorithm. This two-step procedure is equivalent to using an adaptive MH algorithm for the *within*-model part of the RJMCMC algorithm.

In detailing the *across*-model part of the RJMCMC algorithm, let $k = 1$ and $k = 2$ correspond respectively to the Poisson and Negative Binomial models, such that

$$\begin{aligned} k = 1 &: \; Y_i \sim \text{Poisson}(\lambda_i); \; \boldsymbol{\theta}_1 = \boldsymbol{\beta}_1, \text{ and,} \\ k = 2 &: \; Y_i \sim \text{NegBin}(\lambda_i, \kappa); \; \boldsymbol{\theta}_2 = (\boldsymbol{\beta}_2, \kappa), \end{aligned}$$

where λ_i is specified as in equation (1). The posterior density can be concisely written as

$$p(k, \boldsymbol{\theta}_k | \boldsymbol{Y}) \propto \begin{cases} \pi(k=1)\pi(\boldsymbol{\theta}_1 | k=1) L^P(\boldsymbol{Y} | \boldsymbol{\theta}_1) & \text{for } k = 1, \\ \pi(k=2)\pi(\boldsymbol{\theta}_2 | k=2) L^{NB}(\boldsymbol{Y} | \boldsymbol{\theta}_2) & \text{for } k = 2, \end{cases}$$

with $\pi(\boldsymbol{\theta}_1 | k = 1)$ given by equation (3) and $\pi(\boldsymbol{\theta}_2 | k = 2)$ given by multiplying equations (2) and (3).

When the proposed move is a jump from model 1 to model 2, the current state $x = (1, \boldsymbol{\theta}_1)$ has no equivalent for the parameter κ. The idea is then to generate a dummy variable that will be mapped into a proposed value for κ in model 2. To this end, the strategy is to

generate $u \sim N(0, \sigma)$ and to set $x' = (2, \boldsymbol{\theta}_2')$ where

$$\boldsymbol{\theta}_2' = (\boldsymbol{\beta}_2', \kappa') = h(\boldsymbol{\beta}_2, u) = (\boldsymbol{\beta}_2, \mu \exp(u)).$$

Hence, the vector of parameters $\boldsymbol{\beta}_2$ is maintained across jumps but κ is generated as a lognormal random variable to compensate the fact that it has no equivalent in model 1. The Jacobian of the transformation is straightforward to calculate and yields $\mu \exp(u)$.

Note how the calibration exercise is crucial to implement an efficient probability of *across*-model jump : a good calibration will have values of κ generated close to the real value, since this will affect the probability that an *across*-model jump is accepted. A bad calibration will have values of κ generated far from the real value, affecting the proposed posterior and making it harder to accept a jump from the Poisson model to the Negative Binomial model. Using an adaptive MH algorithm improves convergence since the RJMCMC algorithm is based on parameters estimated from the model in the first stage.

For a jump from model 2 to model 1, there is no need to generate a random variable since the parameter space is actually reduced. It suffices to use the inverse transformation h^{-1} to transform the variables back to the Poisson model. In other words, set $(\theta_1, u) = \big(\boldsymbol{\beta}_1', log(\kappa'/\mu)\big)$. The Jacobian of the transformation is then $\frac{1}{\kappa'}$. Note that in this example, the parameter κ is only meaningful when considering the Negative Binomial model.

The probability of jumping across models can now be fully described. Particularly, for the move from 1 to 2 the probability of accepting the jump is given by $\min\{1, A_{1,2}\}$ where

$$A_{1,2} = \frac{p(2, \boldsymbol{\theta}_2'|\boldsymbol{Y})}{p(1, \boldsymbol{\theta}_1|\boldsymbol{Y})} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-u^2}{2\sigma^2}\right] \right\}^{-1} \mu \exp(u)$$

whereas for the move from 1 to 2 the probability of accepting the jump is given by $\min\{1, A_{2,1}\}$ where

$$A_{1,2} = \frac{p(1, \boldsymbol{\theta}_1|\boldsymbol{Y})}{p(2, \boldsymbol{\theta}_2'|\boldsymbol{Y})} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-\big(\log(\kappa'/\mu)\big)^2}{2\sigma^2}\right] \right\} \frac{1}{\kappa'}.$$

In all cases, the initial model states and parameter values are chosen randomly. The exogenous probability of attempting a model jump, $j_m(x)$, is set to 0.5. The priors are the same across all models considered and are set to $\kappa \sim G(0.001, 100)$ and $\beta_j \sim N(0, 10)$. The priors are chosen to be diffused and do not convey much information about the true parameter values. The estimates should therefore be close to those obtained through a classical estimation using Maximum Likelihood, as shown in the results of Section 6. The parameter μ in the transformation function for an *across*-model jump is set to 0.005, and the random variable u is generated according to $u \sim N(0, 0.05)$.

5 Data

The dataset consists of a cross-section of observations on consumer loans provided by a Canadian financial institution. After cleaning it from missing information and outliers, it includes 37,160 observations extracted on December 31st 2007. The loans are not mortgage related and are relatively small, the average (standard deviation) face value of the loans is 13,910.24 (9,320.60) with a remaining balance of 5,823.60 (7,090.62).

The dependent variable in the analysis is the number of missed payments on the loan. Its definition and frequency are reported in Table 1. The mean (standard deviation) value for the number of missed payments is 0.02322 (0.2761). 98.99% of the sample borrowers are under 29 days late on their loan payment which does not count as a missed payment (a billing cycle is assumed to be 30 days) and hence have a missed payment count of 0.

As control variables, I use a variable indicating that a loan is unproductive (since a late loan might be considered productive or not by the bank, depending on information not necessarily available in the database provided) and a variable indicating the highest number of missed payments on the loan since its origination. I choose these two variables because a loan's unproductive status captures soft information potentially available to the banker, while the worst past delinquency sends a strong signal of future delinquency.[3] Only 0.6% of the loans are considered unproductive and the average (standard deviation) of the worst number of missed payments on the loans is 0.03889 (0.5836).

Table 1: Frequency Table for Missed Payment

# Missed Payments	Definition	Frequency	Percent	Cumulative
0	Under 29 days late	36,784	98.99	98.99
1	30 to 59 days late	165	0.44	99.43
2	60 to 89 days late	80	0.22	99.65
3	90 to 119 days late	53	0.14	99.79
4	120 to 149 days late	31	0.08	99.87
5	150 to 179 days late	27	0.07	99.95
6	180 to 209 days late	20	0.05	100
Total		37,160	100	

[3]Other control variables, such as demographics (for example, sex, age, income or dependents), did not improve the quality of the estimation. Such variables are collected at loan origination and are not updated at time of default. An ideal situation would be to have a panel dataset to follow the evolution of the missed payments over time.

6 Results

In this Section, I present the results of the application to consumer loan payment defaults. The Reversible Jump algorithm is convenient because the posterior model probability estimates are directly available from the output of the simulation so they do not require additional numerical computations. For a given model k, they are estimated as

$$\hat{p}(k|\boldsymbol{Y}) = \frac{1}{r-b}\sum_{i=b+1}^{r}\mathbb{I}_i(k),$$

where the indicator function $\mathbb{I}_i(k) = 1$ if the model is equal to k for observation i and is equal to zero otherwise. In the case considered, since the prior model probabilities are set equal to 0.5, the posterior odds and Bayes factors are equal. They are estimated as

$$\widehat{PO}_{12} = \widehat{BF}_{12} = \frac{\hat{p}(k=1|\boldsymbol{Y})}{\hat{p}(k=2|\boldsymbol{Y})}.$$

6.1 Poisson and Negative Binomial Comparison

When comparing the Poisson and Negative Binomial regression models, the RJMCMC algorithm yields a probability of 0.9999 that the true underlying distribution is Negative Binomial. Although this may seem extreme, the reason is that the overdispersion is so severe in the data when the excess zeros are not modeled that it is virtually impossible to accept the Poisson model against the Negative Binomial model. This can directly be linked to the estimated value for κ which is 2.7249. Recall that the Poisson specification implicitly assumes a value of 0 for κ. This shows that the excess zero realizations in the data induce an extreme overdispersion which cannot be captured by the Poisson regression model.

In Table 2, I present two sets of results. The first two columns are estimates from the Bayesian RJMCMC algorithm. I report the posterior mean of each parameter along with its standard deviation and in brackets, I report the 95% credible interval which is found by calculating the 2.5% and 97.5% quantiles of the posterior mean distribution. The two last columns are estimates from the classical MLE estimation and are reported for comparison purposes. I report each estimated parameter with its standard error and the 95% confidence interval in brackets. Although both sets of results are very similar, the classical method does not provide estimated posterior model probabilities, which is the big advantage of the RJMCMC algorithm in this case. Note that the overdispersion parameter κ is only present conditional on being in the Negative Binomial specification since it is implicitly assumed to be 0 in the Poisson model.

Table 2: Poisson and Negative Binomial Comparison

	Bayesian (RJMCMC)		Classical (MLE)	
	Poisson	NegBin	Poisson	NegBin
Constant $\pm\,\sigma$	-4.9098 ± 0.0614 [-5.0328, -4.7825]	-4.9308 ± 0.0716 [-5.0545, -4.8126]	-4.9007 ± 0.0603 [-5.0189, -4.7825]	-4.9316 ± 0.0618 [-5.0527, -4.8104]
Unproductive $\pm\,\sigma$	5.5454 ± 0.0838 [5.3730, 5.7007]	5.0558 ± 0.1504 [4.7650, 5.3460]	5.5353 ± 0.0803 [5.3779, 5.6926]	5.0626 ± 0.1456 [4.7772, 5.3480]
Worst Default $\pm\,\sigma$	0.0549 ± 0.0072 [0.0401, 0.0688]	0.2553 ± 0.0374 [0.1853, 0.3297]	0.0554 ± 0.0073 [0.0410, 0.0697]	0.2519 ± 0.0361 [0.1812, .3227]
$(\kappa \mid k=2) \pm \sigma$	-	2.7249 ± 0.4331 [1.9540, 3.6488]	-	2.6511 ± 0.4166 [1.9484, 3.6073]
$\hat{p}(k=1\mid \boldsymbol{Y})$	0.0001			
$\hat{p}(k=2\mid \boldsymbol{Y})$	0.9999			

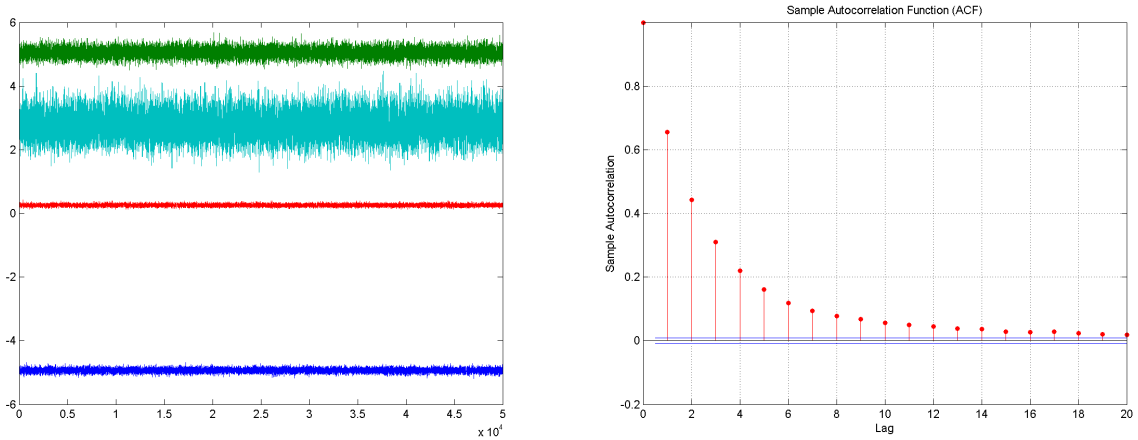
The first column presents the estimates of the Poisson model as estimated during the first step of the procedure, after the initial 20,000 burn-in period since the model has not been visited enough times during the RJMCMC algorithm to provide adequate estimates of the parameters. For the same reason, in Figure 1a, only the output of RJMCMC chain for the Negative Binomial model is presented.

For both models, the variables indicating that a loan is unproductive and the worst default status since loan origination are positive, which shows that both variables have a positive impact on the current number of missed payments on the loan. The standard deviation of each parameter are small enough to conclude to a statistically significant effect.

Figure 1a presents the output of the RJMCMC chain for the Negative Binomial regression model. Since the parameters have been estimated in the first step of the procedure with a random walk MH algorithm, convergence in the second step is achieved from the first iteration and does not require a burn-in period. These chains can each be plotted into the posterior distribution of their respective parameter. Figure 1b presents the autocorrelation function of the κ as example. The dependence in the output chain shrinks rapidly, as is the case for all of the estimated parameters.

A posterior model probability of 0.9999 yields a Bayes factor near infinity, which is an extreme indication that the Negative Binomial model is preferred to the Poisson model. However, since the excess zero realizations are not modeled in this comparison, they produce an artificial amount of overdispersion that could be modeled more adequately. For this reason, I consider the extension to hurdle models in the next section, allowing me to focus on residual overdispersion, after the excess zero realizations are taken into account properly.

Figure 1: Convergence of the RJMCMC Output
(Negative Binomial)



(a) Chain Output

(b) Autocorrelation of the κ Chain

6.2 Poisson Hurdle and Negative Binomial Hurdle Comparison

When comparing the Poisson Hurdle and Negative Binomial Hurdle regression models, the RJMCMC algorithm provides an estimated posterior probability of 0.4838 that the true underlying model is Poisson and an estimated posterior probability of 0.5162 that the true underlying model is Negative Binomial. In this case, Bayesian averaging would be roughly equivalent to putting 50% weight on the estimates of each model. This yields Bayes factors of $\widehat{BF}_{12} = 0.9372$ and $\widehat{BF}_{21} = 1.0670$, which are very low and provide no significant evidence that one model outperforms the other.

The interesting feature of the results is that, conditional on modeling the zero realizations correctly, the Negative Binomial model no longer appears more probable than the Poisson model in the data, which shows that the excess zeros were causing the overdispersion found in Section 6.1.

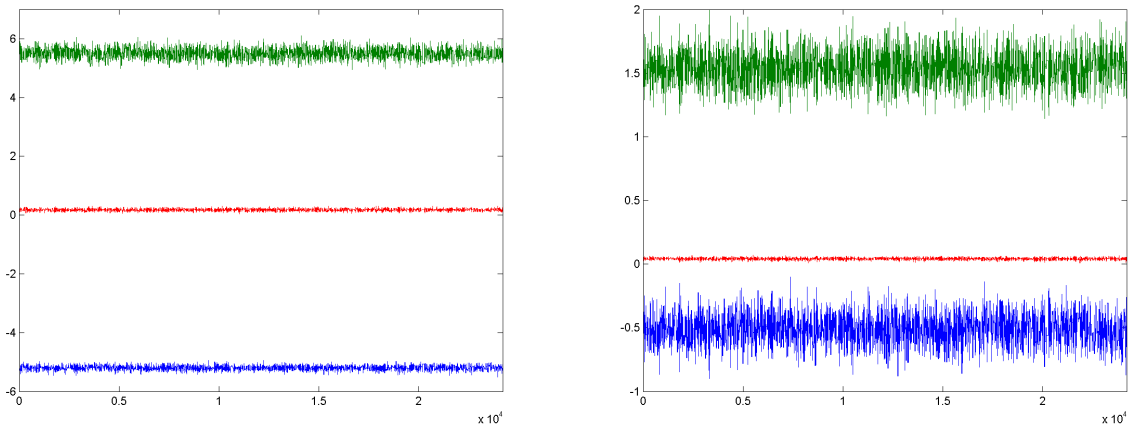
In table 3, I report the results from the Bayesian estimation and I omit the comparison with the MLE estimation since the results are very close, as was the case previously. Columns 1 and 3 present the results for the binary outcome models and columns 2 and 4 respectively present the results for the truncated Poisson and Negative Binomial models. In both cases, the parameters of the hurdle estimations should be interpreted as a binary outcome model, that is, as the decision between paying the loan on time (the 0 outcome) or missing at least one payment (the 1 outcome). Both hurdle equations are estimated in the same way (as derived in Section 1), which explains why their results are very similar. The truncated part of the model is interpreted as the usual truncated count regressions.

In every case, the variables indicating that a loan is unproductive and the worst default status since loan origination are positive, just like in the previous results. The estimated posterior mean of the overdispersion parameter κ is now 0.0098 with a standard deviation of 0.0094 which does not provide evidence of significant overdispersion in the truncated part of the default realizations. In fact, the estimated value for κ is very close to the assumption that $\kappa = 0$ in the Poisson model. This also explains why the estimated parameters for the truncated Poisson and Negative Binomial models are so similar.

Table 3: Hurdle Models Comparison

	Poisson		Negative Binomial	
	Hurdle	Truncated	Hurdle	Truncated
Constant $\pm\,\sigma$	-5.1986 ± 0.0865 [-5.3412, -5.0590]	-0.5123 ± 0.1108 [-0.7400, -0.3010]	-5.1671 ± 0.01174 [-5.3581, -4.9733]	-0.4945 ± 0.1221 [-0.7284, -0.2562]
Unproductive $\pm\,\sigma$	5.5016 ± 0.1768 [5.1677, 5.8397]	1.5413 ± 0.1185 [1.3108, 1.7882]	5.2991 ± 0.2013 [4.9184, 5.6855]	1.5261 ± 0.1693 [1.1922, 1.8507]
Worst Default $\pm\,\sigma$	0.1778 ± 0.0367 [0.1102, 0.2516]	0.0419 ± 0.0112 [0.0265, 0.0565]	0.2026 ± 0.0525 [0.1016, 0.3058]	0.0410 ± 0.0144 [0.0119, 0.0694]
$(\kappa \mid k=2) \pm \sigma$	-	-	-	0.0098 ± 0.0094 [0.0021, 0.0293]
$\hat{p}(k=1\mid \boldsymbol{Y})$	0.4838			
$\hat{p}(k=2\mid \boldsymbol{Y})$	0.5162			
$\widehat{BF}_{12}$	0.9372			
$\widehat{BF}_{21}$	1.0670			

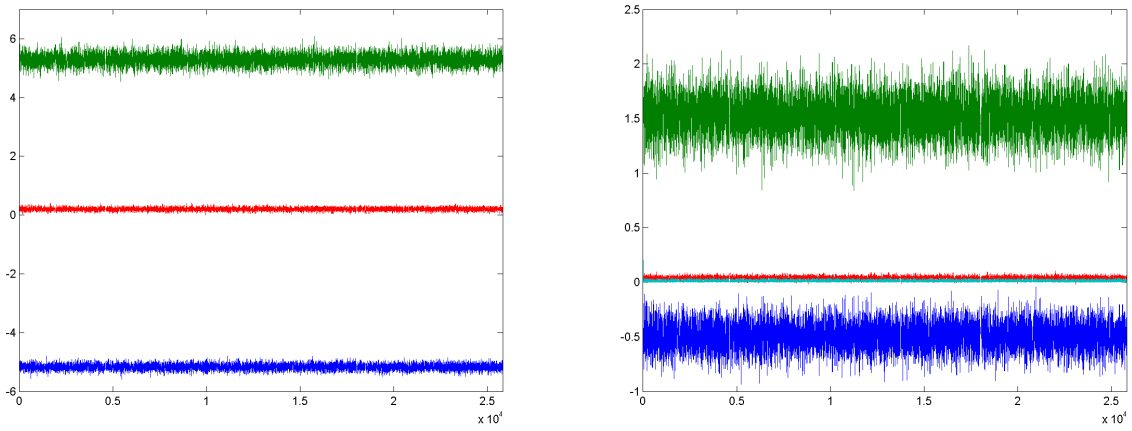
Figure 2: Convergence Diagram of the RJMCMC Output
(Poisson Hurdle)



(a) Binary Outcome Model

(b) Truncated Poisson Model

Figure 3: Convergence Diagram of the RJMCMC Output
(Negative Binomial Hurdle)



(a) Binary Outcome Model

(b) Truncated Negative Binomial Model

Figures 2 and 3 present the outputs of the RJMCMC algorithm conditional on the model visited. Since each model has been visited about half of the time, the chains consist of about 25,000 points each. Since the parameters have been estimated in the first step of the procedure with a random walk MH algorithm, convergence in the RJMCMC algorithm was achieved from the first iteration and did not require a burn-in period. The autocorrelation functions were omitted since they are qualitatively similar to the one presented in Figure 1b.

7 Conclusion

In this note, I present the Reversible Jump MCMC algorithm of Green (1995). The comparison between the Poisson and Negative Binomial models provides an ideal example of model selection for which the number of parameters varies across models. This simple example and its application to bank-level data highlights the importance of correctly choosing an underlying regression model.

When comparing the Poisson and Negative Binomial regression models for consumer loan default, the latter is found to be the most probable specification. In fact, in a run of the RJMCMC algorithm, it is virtually impossible to reject the Negative Binomial specification for the Poisson model. The estimated overdispersion parameter is 2.72, which strongly contrasts with the assumption that it is 0 in the Poisson model.

However, when introducing hurdle specifications in the comparison between the Poisson and Negative Binomial models, both of the model are equiprobable. Indeed, after modeling the excess zeros through a binary outcome model, the estimated overdispersion parameter in

the Negative Binomial model is approximately 0 which shows that there is no overdispersion in the truncated part of the count realization.

I use an adaptive version of the RJMCMC algorithm which helps to improve convergence of the parameters. Although the RJMCMC algorithm provides an easy way to estimate posterior model probabilities, calibrating the *across*-model jump in an optimal manner can sometimes be difficult. By first estimating each of the models separately using a random walk MH algorithm, I get reasonable estimates of the parameters of each models, which can then be used in an independent chain MH algorithm. This proves to be both useful to increase the convergence speed and to provide the estimates of a model that has not been visited sufficiently in the RJMCMC algorithm.

References

Ausín, M.C., J.M. Vilar, R. Cao, and C. González-Fragueiro, 2011, Bayesian analysis of aggregate loss models, *Mathematical Finance* 21.

Cameron, A. Colin, and Pravin K. Trivedi, 1986, Econometric models based on count data: Comparisons and applications of some estimators and tests, *Journal of Applied Econometrics* 1.

——— , 1998, Regression analysis of count data, *Econometric Sciety Monographs No. 30.*

Casella, George, and Edward I. George, 1992, Explaining the gibbs sampler, *The American Statistician* 46.

Centanni, S., and M Minozzo, 2006, Estimation and filtering by reversible jump mcmc for a doubly stochastic poisson model for ultra-high-frequency financial data, *Statistical Modelling* 6.

Chib, Siddhartha, and Edward Greenberg, 1995, Understanding the metropolis-hastings algorithm, *The American Statistician* 49.

Dionne, Georges, Manuel Artís, and Montserrat Guillén, 1996, Count data models for a credit scoring system, *Journal of Empirical Finance* 3.

Dionne, Georges, and Charles Vanasse, 1989, A generalization of automobile insurance rating models: The negative binomial distribution with a regression component, *Astin Bulletin* 19.

Green, Peter J., 1995, Reversible jump markov chain monte carlo computation and bayesian model determination, *Biometrika* 82.

Hansen, Bruce E., 2005, Challenges for econometric model selection, *Econometric Theory* 21.

Hastie, David I., and Peter J. Green, 2012, Model choice using reversible jump markov chain monte carlo, *Stastistica Neerlandica* 66.

Hoeting, Jennifer A., David Madigan, Adrian E. Raftery, and Chris T. Volinsky, 1999, Bayesian model averaging: A tutorial, *Statistical Science* 14.

Ntzoufras, Ioannis, Athanassios Katsis, and Dimitris Karlis, 2005, Bayesian assessment of the distribution of insurance claim counts using reversible jump mcmc, *North American Actuarial Journal* 9.

Sisson, Scott A., 2005, Transdimensional markov chains: A decade of progress and future perspectives, *Journal of the American Statistical Association* 100.

Vrontos, I.D., P. Dellaportas, and D.N. Politis, 2000, Full bayesian inference for garch and egarch models, *Journal of Business & Economic Statistics* 18.

Vuong, Quang H., 1989, Likelihood ratio tests for model selection and non-nested hypotheses, *Econometrica* 57.