Bachelor Workgroup Essay

Roos Kilburn
Supervisor: prof. dr. M.A. Grzegorczyk

31 January 2023

Contents

1 Introduction **3**

2 Background **4**
2.1 Monte Carlo methods 4
2.2 Markov chains 5
2.3 Bayesian statistics 6
2.4 Markov chain Monte Carlo algorithms 7
2.5 Reversible-jump Markov chain Monte Carlo 8

3 Existing Literature **9**

4 My contribution **13**

1 Introduction

Markov chain Monte Carlo (MCMC) algorithms are a powerful tool in computational statistics. They let us sample from distributions that we only know up to some normalization constant, without having to analytically solve often intractable integrals. This is particularly useful in Bayesian statistics, where we deal with posterior distributions that we only know up to proportionality.

For my thesis I will be researching convergence and jump proposal mechanisms for reversible jump Markov chain algorithms applied to linear regression models. Where regular Markov chain Monte Carlo algorithms only jump across parameters in a fixed model, reversible jump Markov chain Monte Carlo also jumps across-model, adding in and removing variables from the model as it jumps. This means the algorithm is also exploring which model is best in predicting the data.

However, this does bring some difficulties in designing a proposal mechanism, since across-model jumps don't have the same concept of proximity that within-model jump proposals use. Without a proper jump proposal mechanism these algorithms may converge impractically slow.

In this essay I will go over the background of Markov chain Monte Carlo methods, talk about the applications of these methods to Bayesian statistics and go over the literature that will be my starting point for my thesis. I will close with what I hope to add to the field.

2 Background

I will start with a brief overview of Monte Carlo methods and Markov chains to set the stage for Markov chain Monte Carlo algorithms.

2.1 Monte Carlo methods

Monte Carlo methods are programs that let us approximate quantities through repeated random sampling. Let's look at a simple example to illustrate what that might look like:

Say we are interested in the area of a circle with a certain radius. One way of approximating this area is by taking a box that contains the circle, and then randomly sampling points in the box. By taking the proportion of samples that happen to lie inside the circle to the total sample size, we can get an idea of the proportion between area of the circle and the square. Since the area of the square is known to us, we can now approximate the area of the circle, or any arbitrary shape that fits in the square.

While Monte Carlo methods are approximations, they are generally constructed such that their limit will approach the desired quantity as we increase the sample size.

Monte Carlo methods are often used when more direct methods of computation are difficult. While integrating over a circle is quite simple, it is quite common to run into integrals that are intractable to solve analytically. This is particularly common in Bayesian statistics, where as we will see, the posterior distribution has an integral term which can be quite complicated to solve analytically.

2.2 Markov chains

A Markov Chain is a sequence of events or states, where the probability of transitioning from the current state to the next is only dependent on the current state.

We define a discrete Markov chain as a stochastic process $\{X_t\}_{t=1,2,3,...}$ with the discrete state space S, which fulfills the Markov property:

$$P(X_t = x_t | X_{t-1} = x_{t-1}, ..., X_1 = x_1) = P(X_t = x_t | X_{t-1} = x_{t-1})$$

for all states $x_1, ..., x_t \in S$

Often represented by nodes that represent states, and arrows with the probability of transitioning to another state.

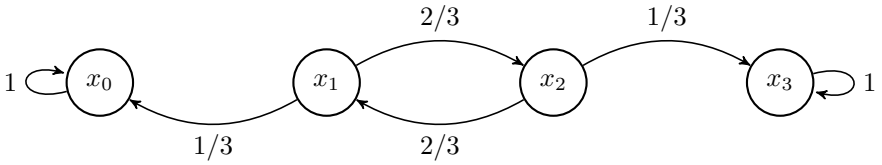

Figure 1: Simple Markov chain

We can sample from a Markov chain by choosing a starting node and moving through the states according to the transition probabilities. By taking count of the various states, we could get an idea where in the chain most time is spent. In the above example it is easy to see that we will get stuck in either x_0 or x_3 quite quickly. For our purposes we would like to avoid this, only working with Markov chains that satisfy certain properties.

First off, a Markov chain is called *irreducible* if any state can be reached from any other state in a finite number of steps with positive probability, i.e., if for all $i, j \in S$ there exists $t > 0$ such that $P(X_t = j | X_1 = i) > 0$.

Secondly, Markov chains are *periodic* if some states can only be reached in a certain order. If they are not periodic we call them aperiodic. A Markov chain is aperiodic if for each state $i \in S$ there is a natural number n_i such that for all $t \geq n_i$:

$$P(X_t = i | X_1 = i) > 0$$

If a discrete Markov chain with finite state space is both irreducible and aperiodic we call it *ergodic*. Ergodic Markov chains are nice because they guarentee that there exists a unique stationary distribution. Meaning that repeated sampling from the chain will result in a predictable distribution of time spent in the various states. This property is key in developing Markov chain Monte Carlo algorithms.

2.3 Bayesian statistics

Before getting in to Markov chain Monte Carlo methods, let's take a step back to see why they are so useful for Bayesian statistics.

In Bayesian statistics we are often interested in posterior distributions $p(\theta|x_1, ..., X_n)$[6]. We have some data $x_1, ..., x_n$ and we are not just interested in the value of θ which is most likely to generate our measured data, but we want to study the whole distribution of possible values our parameters can take.

From Bayes' law we get:

$$\begin{aligned} p(\theta|x_1, ..., x_n) &= \frac{p(x_1, ..., x_n|\theta)p(\theta)}{p(x_1, ..., x_n)} \\ &= \frac{p(x_1, ..., x_n|\theta)p(\theta)}{\int_\Theta p(x_1, ..., x_n, \theta)d\theta} \\ &= \frac{p(x_1, ..., x_n|\theta)p(\theta)}{\int_\Theta p(x_1, ..., x_n)p(\theta)d\theta} \end{aligned}$$

While we generally know the likelihood $p(x_1, ...x_n|\theta)$ and the prior $p(\theta)$, the marginal term in the numerator can be tricky. The integral can often be intractable to solve analytically, especially as the dimension of the parameter space increases.

But notice that the marginal is only a constant, we know the posterior distribution is proportional to the likelihood times prior:

$$p(\theta|x_1, ..., x_n) \propto p(x_1, ..., x_n|\theta)p(\theta)$$

Now this is where Markov chain Monte Carlo techniques can help us sample from this posterior distribution without us having to find an exact expression for it.

Our goal is to set up an ergodic Markov chain that transitions between states in such a way that it's stationary distribution lines up exactly with the posterior distribution we are interested in. This would mean that sampling from the Markov chain should be indistinguishable from sampling from the posterior distribution,when our sample size is sufficiently large. Then from these Markov samples we can approximate all kinds of quantities we are interested in such as the mean and variance of the posterior distribution.
This is exactly what Markov chain Monte Carlo algorithms are designed to do.

2.4 Markov chain Monte Carlo algorithms

So we have a goal in mind, we want to create a Markov chain with a stationary distribution that happens to line up exactly with a posterior distribution we are interested in. How do we go about doing this?

To start answering this question we consider the following lemma. Here T is the matrix of probabilities $T_{i,j}$ that represent the probability of transitioning from state i to j:

lemma: Given a discrete ergodic Markov chain with finite state space $S = 1, ..., k$ and transition matrix T. If a distribution $\pi = (p(1), ..., p(k))$ fulfills for all $i, j \in S$ with $p(i) > 0$ and $T_{j,i} > 0$, the so-called "Equation of Detailed Balance":

$$\frac{T_{i,j}}{T_{j,i}} = \frac{p(j)}{p(i)}$$

then π is the stationary distribution of this Markov chain.

The basic idea of any Markov chain Monte Carlo algorithm is to find a transition rule that satisfies the equation of detailed balance and also involve what we know about the posterior distribution in such a way that the stationary distribution of the Markov chain lines up with it.

While I will avoid going into the technical details in this essay, it is useful to go over what the Metropolis-Hasting algorithm is doing.

Starting at some arbitrary state i, the algorithm will propose a possible next state j. It does this by sampling from some symmetric distribution and adding it to a parameter of the current state. Then it will either accept or reject the proposed state with the following probability:

$$A(i, j) = min\{1, \frac{p(j)}{p(i)} \cdot \frac{Q(j, i)}{Q(i, j)}\}$$

Where $Q(i, j)$ represents the probability the move is proposed. It will do this for each parameter, jumping through the parameter space, preferring moves from lower to higher probability.

It can be shown that the transition probability of this algorithm satisfies the equation of detailed balance, and furthermore that the stationary distribution of this Markov chain will line up with the posterior distribution we are interested in. Which is exactly what we want.

For my thesis I plan to expand this theory section with some extra care, to explore and show the proofs that are the foundation of MCMC algorithms.

Looking at the acceptance probability, it does make intuitive sense. When proposing moves we accept proportionally to the likelihood of the new state to the old, spending more time in states with high likelihood and less time in states with low likelihood.

2.5 Reversible-jump Markov chain Monte Carlo

The reversible jump markov chain monte carlo sampler[5] lets us do MCMC sampling in which the dimension of the parameter space varies as well as the value of parameters. This means that the simulation will explore not just the parameters of a single model, but also which models are best at predicting the data.

While this is a very powerful method, without a smart proposal mechanism these algorithms may explore the space of parameters and models very slowly. So what makes a good proposal?

When proposing a new state in a Markov chain Monte Carlo algorithm, it is desirable that the acceptance rate is not too high or too low. When working with conventional MCMC algorithms like the Metropolis-Hastings algorithm, the next state takes the form of $x_{t-1} + \mu$ where x_{t-1} is the previous state and μ is distributed by some symmetric distribution, i.e., uniformly or Gaussian. If we choose μ's range to be too small, we may find that most proposals get accepted, and the parameter space is explored very slowly due to this small step size. But when we choose a very large range of values for μ we may find that very few proposals get accepted at all, because if we are in a region of high probability, far off proposals will have a minuscule acceptance rate. Both of these extremes cause very slow convergence rates.

While we can use the concept of proximity to tune acceptance rates for conventional MCMC, the idea of proximity between models is trickier. Designing reversible-jump MCMC algorithms that achieve reasonable acceptance rates can often prove quite difficult. So developing proposal mechanisms for RJ-MCMC methods is still an active area of research.

3 Existing Literature

Here is an overview of the existing literature surrounding reversible jump Markov chain Monte Carlo methods, focusing on proposal mechanisms and convergence behavior. For each paper I have included the relevant part of its' abstract, and a personal comment on the utility of the papers for my research.

On Bayesian Analysis of Mixtures with an Unknown Number of Components (with discussion)[8]

> New methodology for fully Bayesian mixture analysis is developed, making use of reversible jump Markov chain Monte Carlo methods that are capable of jumping between the parameter subspaces corresponding to different numbers of components in the mixture. A sample from the full joint distribution of all unknown variables is thereby generated, and this can be used as a basis for a thorough presentation of many aspects of the posterior distribution. The methodology is applied here to the analysis of univariate normal mixtures, using a hierarchical prior model that offers an approach to dealing with weak prior information while avoiding the mathematical pitfalls of using improper priors in the mixture context.

This is one of the first papers after the introduction of the RJ-MCMC method in 1995 to attempt improving between-model proposals. Seeing the initial attempts to improving these mechanisms will hopefully show me in what ways I can approach the problem.

Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions[1]

> (. . .) We consider mechanisms for guiding the choice of proposal (for RJ-MCMC methods). The first group of methods is based on an analysis of acceptance probabilities for jumps. Essentially, these methods involve a Taylor series expansion of the acceptance probability around certain canonical jumps and turn out to have close connections to Langevin algorithms. The second group of methods generalizes the reversible jump algorithm by using the so-called saturated space approach. These allow the chain to retain some degree of memory so that, when proposing to move from a smaller to a larger model, information is borrowed from the last time that the reverse move was performed. (. . .)

This paper introduces several proposal mechanisms for RJ-MCMC. Especially interesting is the approach where the chain retains some degree of memory to

make better proposals, while usually the chain is essentially memory-less. It will be interesting to see how they preserve (or lose) the theoretical guarantees that usually make sure the stationary distribution exists and aligns with the target distribution.

Model Choice using Reversible Jump Markov Chain Monte Carlo[7]

> We review the across-model simulation approach to computation for Bayesian model determination, based on the reversible jump Markov chain Monte Carlo method. Advantages, difficulties and variations of the methods are discussed. We also discuss some limitations of the ideal Bayesian view of the model determination problem, for which no computational methods can provide a cure. Some key words: Across-model sampling, Bayes factors, Bayesian model determination, poste- rior model probabilities, transdimensional inference, variable dimension problems,

This paper provides a valuable overview of the research area. Particularly useful for me are the sections on the challenges of implementing RJ-MCMC methods, ways of improving proposals and convergence diagnostics.

An efficient interpolation technique for jump proposals in reversible-jump Markov chain Monte Carlo calculations[4]

> (...) Here, we demonstrate an interpolation technique that uses samples from single-model MCMCs to propose intermodel jumps from an approximation to the single-model posterior of the target parameter space. The interpolation technique, based on a kD-tree data structure, is adaptive and efficient in modest dimensionality. We show that our technique leads to improved convergence over naive jumps in an RJMCMC, and compare it to other proposals in the literature to improve the convergence of RJMCMCs. (...)

Exploring single-model MCMC to inform jumps across-model seems like a smart approach of informing across-model jumps. I am curious to see how easily this method can be adapted and how effective it is.

On a Selection of Advanced Markov Chain Monte Carlo Algorithms for Everyday Use: Weighted Particle Tempering, Practical Reversible Jump, and Extensions[2]

> (...) The humble goal of this work is to bring to the table a few more highly versatile and robust, yet easily-tuned algorithms. Specifically, we introduce weighted particle tempering, a parallelizable MCMC procedure that is adaptable to large computational resources. We also explore and develop a highly practical implementation of reversible jump, the most generalized form of Metropolis-Hastings. Finally, we combine these two algorithms into reversible jump weighted particle tempering, and apply it on a model and dataset that was partially collected by the author and his collaborators, halfway around the world. It is our hope that by introducing, developing, and exhibiting these algorithms, we can make a reasonable contribution to the ever-growing body of MCMC research.

The practical implementation of RJ-MCMC developed by the authors will be valuable in writing my own omplementation. While interesting, their parallelizable MCMC producedure, and the extenstion for RJ-MCMC will be out of the scope of my thesis.

Bayesian Approach to Variable Selection in Linear Regression Model[3]

> (...) The paper presents Bayesian approach to the variable selection (or more generally of model choice) in the linear regression model. There are several practical solutions in this approach. The one used here will be the Reversible Jump Markov Chain Monte Carlo algorithm. The article shows an application of this method with the usage of the so-called data prior which utilizes part of the data to produce an informative prior.

This paper provides me with an example of the RJ-MCMC algorithm for the linear regression model. This will be a good starting point to study when working on my own implementation.

Convergence diagnostics for Markov chain Monte Carlo[9]

> (...) Two critical questions MCMC practictioners need to adress are where to start and stop a simulation. Although a great amount of research has gone into establishing convergence criteria and stopping rules with sound theoretical foundation, in practice, MCMC users often decide convergence by applying empirical diagnostic tools. This review article discusses the most widely used MCMC convergence diagnostic tools. Some recently proposed stopping rules with fi
>
> rm theoretical footing are also presented. The convergence diagnostics and stopping rules are illustrated using three detailed examples.

This paper is on convergence diagnostics in regular MCMC methods, some of these tools may be able to be adapted for RJ-MCMC. So having a relatively recent overview of the most commonly used diagnostic tools will be valuable when considering methods of monitoring convergence for my own algorithms.

4 My contribution

Now what do I hope to contribute to the field of research. I wish to gain a deeper understanding of the reversible-jump Markov chain Monte Carlo algorithm, and it's general applicability. While conventional MCMC algorithms have been seeing wide application across any fields, RJ-MCMC algorithms have seen relatively little use, possibly due to the impression they are hard to implement.

I will be implementing a reversible-jump Markov chain Monte Carlo algorithm for a linear regression model. I will be comparing the different proposal mechanisms that have been developed for jumping across-model, measuring convergence rates using available convergence diagnostics.

As a result, I would be happy to have a practical implementation of a reversible-jump Markov chain Monte Carlo algorithm, and have concrete data on the strength and weaknesses of the various across-model proposal mechanisms using convergence criteria as metrics. Ideally, I would write on the insights I have gained into the pragmatic realities surrounding the algorithm, it's applicability to various problems, why it may not see as much use currently and if it will in the future.

References

[1] S. P. Brooks, P. Giudici, and G. O. Roberts. "Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65.1 (2003), pp. 3–39. DOI: https://doi.org/10.1111/1467-9868.03711. eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/1467-9868.03711. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.03711.

[2] Marcos Arantes Carzolio. "On a selection of advanced Markov chain Monte Carlo algorithms for everyday use: Weighted particle tempering, practical reversible jump, and extensions". PhD thesis. Virginia Tech, 2016.

[3] Jan Coufal and Jiřı Tobıšek. "Bayesian Approach to Variable Selection in Linear Regression Model." In: *Ekonomické Listy* 2 (2018).

[4] W. M. Farr, I. Mandel, and D. Stevens. "An efficient interpolation technique for jump proposals in reversible-jump Markov chain Monte Carlo calculations". In: *Royal Society Open Science* 2.6 (June 2015), p. 150030. DOI: 10.1098/rsos.150030. URL: https://doi.org/10.1098%2Frsos.150030.

[5] Peter J Green. "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination". In: *Biometrika* 82.4 (1995), pp. 711–732.

[6] Marco Grzegorczyk. "Lecture notes Statistical Reasoning". In: *Statistical Reasoning course at the Rijksuniversiteit Groningen* (2022).

[7] David I. Hastie and Peter J. Green. "Model choice using reversible jump Markov chain Monte Carlo". In: *Statistica Neerlandica* 66.3 (2012), pp. 309–338. DOI: https://doi.org/10.1111/j.1467-9574.2012.00516.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9574.2012.00516.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9574.2012.00516.x.

[8] Sylvia. Richardson and Peter J. Green. "On Bayesian Analysis of Mixtures with an Unknown Number of Components (with discussion)". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 59.4 (1997), pp. 731–792. DOI: https://doi.org/10.1111/1467-9868.00095. eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/1467-9868.00095. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00095.

[9] Vivekananda Roy. "Convergence diagnostics for markov chain monte carlo". In: *Annual Review of Statistics and Its Application* 7 (2020), pp. 387–412.