Markov Chain Monte Carlo

- an overview and the idea of indirect estimation

Anders Sjögren

January 12, 2005

Preface

This paper is a report for the graduate course in Markov chains held at Chalmers University of Technology during the autumn of 2004 by professor Olle Häggström. The idea of this report is for the student (i.e. the author) to take a peak at the field of Markov Chain Monte Carlo (MCMC) and summarise some concepts. The reader is assumed to have basic knowledge about Markov chain theory in general, even though some concepts are summarised in the introduction.

During the phase of literature research for the report, some ideas of extension of the standard use of MCMC arose. Curiosity made the author (i.e. me) pursue the ideas a little further and include it in the report. Since the available time for writing the report was limited, too little litterature research has been performed which makes it highly possible that this extension is well known or even standard. However, looking at the problem on my own proved to get some brain exercise and deepened my understanding of the foundation of MCMC, which I guess course work is for...

The background material for this paper is limited to [Gilks] and [Häggström] and the report is therefore heavily influenced by those pieces of work.

1 Brief introduction to Markov Chains

In this paper, the kind of Markov Chains focused on can be described by: a sequence of random variables (indexed by time) with the property that the future of the sequence depends only on the current state (=value) of the sequence. The duration of each visit of a state must be strictly positive. Such a Markov chain can be described by it's state space, a transition kernel on the state space and a (possibly state dependent) jump frequency. When the state space is discrete, the transition kernel becomes a transition matrix, where for each pair of current and candidate next state the probability is given of jumping to the candidate next state given the current state.

An important result for the use of Markov chains in Monte Carlo simulations is that: for a Markov chain, $\{X_t\}$, with certain technical properties[1], the distribution of the taken value at time t given the state at time 0, $\pi_{X_t|X_0}$, converges to an unique distribution solely determined by the Markov chain

[1] for example all finite state space Markov chains that are irreducible and aperiodic

itself and not depending on the starting value of the chain:

$$\lim_{t\to\infty} \pi_{X_t|X_0} \to \pi_X \ .$$

This distribution, π_X, is then known as the stationary distribution of the Markov chain.

2 Markov Chain Monte Carlo foundations

In the original type of Monte Carlo simulations, a distribution π is approximated by the empirical distribution $\hat{\pi}$ of a sample of independent observations , $X_1, \ldots, X_n$, from π, i.e.:

$$\hat{\pi}(A) = \frac{1}{n}\sum_{i=1}^{n} I_A(x_i) \ .$$

Perhaps the most common application is estimation of the expected value of π:

$$\mathbb{E}(\pi) = \int x d\pi(x) \hat{=} \int x d\hat{\pi}(x) = \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{x} \ ,$$

where $\hat{=}$ denotes "estimated by".

The application of Markov chains to Monte Carlo estimation comes when a random sample from π is hard to create directly. One of the most significant applications of Markov chain Monte Carlo (MCMC) is in the estimation of the posterior distribution of the parameters in a Bayesian model, i.e. the distribution of the parameters, Θ, conditional on the observed data, D. According to Bayes formula we have:

$$\pi_{\Theta|D} = \frac{\pi_{D|\Theta}\pi_\Theta}{\pi_D} = \frac{\pi_{D|\Theta}\pi_\Theta}{\int \pi_{D|\Theta}(d|\theta)\pi_\Theta(\theta)d\theta} \ .$$

However, the integral $\int \pi_{D|\Theta}(d|\theta)\pi_\Theta(\theta)d\theta$ might be inconvenient to compute, even numerically. Nevertheless, for any two values, θ_1 and θ_2, for the parameters we can calculate the ratio:

$$\frac{\pi_{\Theta|D}(\theta_1|d)}{\pi_{\Theta|D}(\theta_2|d)} = \frac{\pi_{D|\Theta}(d|\theta_1)\pi_\Theta(\theta_1)}{\pi_{D|\Theta}(d|\theta_2)\pi_\Theta(\theta_2)} \ .$$

Intuitively, one would think that walking around in the parameter space taking these probability ratios into account, it could be possible to get a

stationary distribution that equals $\pi_{\Theta|D}(\cdot|d)$. This is in fact possible, as will be shown in the next section.

So, as suggested above, in some situations where we want to estimate (a function of) a distribution, π, but are unable to sample from π to Monte Carlo estimate it, we are in fact able to construct a Markov chain with its unique stationary distribution being π. As noted previously, if we start the chain in an arbitrary state it will eventually end up[2] having distribution π. At that point we can observe the state of the chain and get an observation distributed according to π. Two main problems are left to deal with though: convergence and independence.

2.1 Convergence to stationary distribution

Even though one is promised that the distribution of the Markov chain will converge to π eventually, no bound is set on how long one does have to wait until the chain has converged well enough to π.

For smaller (discrete) examples this problem is possible to solve through calculating how the probability mass spreads from the state at X_0 to the other states as time passes, getting a clear indication of when equilibrium has been achieved. However, this demands keeping the entire state space in memory and making calculations for all the states at each time point. Therefore the problems at hand have to be very small. Also, this way of using the probability mass results in an estimate of π directly, making sampling the chain unnecessary.

Another path is to start several independent realisations of the chain at different starting states. Then when the estimates of π stemming from the different realisations converge (in some sense), one can hope that this is because they have all converged to π. However, there is no guarantee that this is the case. For example, the way of choosing starting states for the different chains might be biased so that some areas of the state space are never explored. Intuitively, if two parts of the state space have a border with small crossing probability and the starting point distribution is strongly biased toward one of the parts, the multiple chains could well converge to the conditional distribution of the favoured part of the state space. This example points our the important role that the Markov chain has in not only having the correct stationary distribution, but also in "allowing easy access" to all

[2]given that the technical conditions are met

relevant parts of the state space.

The time before convergence has been established is referred to as the *burn-in time*. Since the chain prior to that point in time is not distributed according to π, that part of the data is most often discarded and not used when estimating π.

A third path which avoids the need for determining the burn-in time is the Propp-Wilson algorithm. Here the idea is to trace one realisation of a Markov chain backward in time to find a negative timepoint, t, where no matter which state is selected, the same state s_0 will be reached at time 0. This would imply that if the chain was started infinitely far ago, no matter which state it would pass at time t, s_0 would be reached at time 0. Since s_0 would then be the result after an infinite run of the chain, s_0 is an observation from the stationary distribution π. However, the algorithm demands that the realisation from all states are traced forward, typically for a long while back in time. This typically makes the approach infeasible for large state spaces.

A trick that solves the state complexity of the Propp-Wilson algorithm for certain cases is the idea of "Sandwiching". In these cases, opposite extreme states exist in the sense that if they meet at time 0, one can conclude all other states will also meet then. The problems are of course that most problems don't have as extreme cases as demanded above and even if they do, one would have to trace back for a very long period of time before a point is found where the opposite extremes will eventually meet.

2.2 Independence in the sample

Consider a Markov chain $\{X_t\}$ produced with the goal of sampling independently from the stationary distribution π. Assume that we have reached a time, t, where the chain has close enough distribution to π. X_t then clearly has distribution close to π. However, to estimate π well enough for realistic situations, we must draw a large sample from π. A problem arising when $\{X_t\}$ stems from a Markov chain rather than being independently drawn from π is that $\{X_t\}$ are typically dependent. As we will see later, for some ways of constructing $\{X_t\}$[3], X_t and X_{t+1} only differ in at most one dimension. In reality, the severeness of this problem depends heavily on the usage of the estimate $\hat{\pi}$ of π. The expectation estimate of common functions of the state space seems to be regarded as rather robust against this problem.

[3]the Gibbs sampler

However, one could want to estimate a property of π that is more reliant on the independence in the sample.

One way of getting a much less dependent sequence of variables is to perform *thinning*. This means that after observing X at one timepoint t, one waits a while before observing X again. If we wait infinitely long before observing X again, we know that its distribution will converge to π. However, for reasonable ways of constructing the chain and for reasonable demands on the independence of the thinned sequence, not that long waiting times should be needed.

In [Implementing MCMC, pp 115, Gilks] a binary function f of the state space of $\{X_t\}$ is of interest. It is used to create a family of sequences $Z_t^k = f(X_{tk})$ where k is a thinning parameter. To determine a k giving an independent sample, an information content approach is taken. For different values of k, the likelihood ratio test is performed between the first and second order Markov chains adapted to Z_t^k. When k is small, the second order chain provides much better likelihoods, while for k large enough for $X_t k$ and $X_{(t+1)k}$ to be independent, they should provide an equal amount of information. When the Markov chain is adapted to the same data that is subsequently used to produce a likelihood, overfitting makes the second order chain always having higher likelihood than the first order one. Therefore, a Bayesian information criterion (BIC) is used to determine if for a fixed k the increase in likelihood when going from a first to a second order chain is due to overfitting or if there really is a dependence in the sequence that is captured by the second order chain. The lowest k for which the first order chain is selected in favor of the second order one is then selected as the thinning parameter to use. In order to estimate k, a pilot run is used.

Additional to the direct goal of decreasing dependence, thinning can be used to save space. For a fixed amount of available space, in effect determining the length of the analysed sequence, a more independent sequence contains more information than a more dependent one.

3 Markov Chain Monte Carlo algorithms

In the previous section it was implied that it might be possible to construct a Markov chain with a stationary distribution, π, which is infeasible to sample from directly. In this section we will show examples of this.

3.1 The Metropolis-Hastings algorithm

In this algorithm which was generalised by Hastings in 1970 to extend the results of Metropolis (1953), one starts out with a proposal transition kernel $q(\cdot|X_t)$. This kernel would form a Markov chain in its own, possibly with a stationary distribution. However, q does not in general directly generate the stationary distribution π. Instead, q is modified in a clever yet simple way to generate a Markov chain with the sought stationary distribution. This is performed through modifying the probability of staying in the current state at any jumping moment. Loosely speaking, the chain tends to jump easily from states that would be overrepresented by q alone, while it tends to stay longer at underrepresented states. The algorithm is as follows:

1. Given the current state at time t, $X_t = x_t$, generate a proposal for time $t+1$: $X'_{t+1} \sim q(\cdot|X_t)$.

2. Now X'_{t+1} is accepted with probability $\min(1, \frac{\pi(X'_{t+1})q(X|X'_{t+1})}{\pi(X)q(X'_{t+1}|X)})$. If it is accepted, $X_{t+1} = X'_{t+1}$, otherwise X does not change, i.e. $X_{t+1} = X_t$.

Now, it is easy to show that the distribution π would make the chain reversible, i.e.: $q(X_{t+1}|X_t)\pi(X_t) = q(X_t|X_{t+1})\pi(X_{t+1})$. Since a distribution that is reversible for a Markov chain is a stationary distribution for it, the chain above will have π as stationary distribution. However, q must be irreversible or else isolated parts of the state space exist that could be never reached, i.e. the stationary distribution will not be unique and the chain will in general not converge to π. For the same reason, q must be able to reach all parts of the state space having non-zero probability mass under π.

Since in the previous section it was noted that care should be taken on how to construct the Markov chain in addition to the demand of having π as unique stationary distribution, care must now be taken on the choice of q. The best way of choosing q depends heavily on the nature of π, for example what relation exists among the dimensions of π. Some examples of different methods are described below.

3.2 Gibbs sampling

In situations where mutual and possibly complex dependencies exist among the dimensions of π, *Gibbs sampling* provides a way of getting a good proposal

kernel q. Here, the proposal by q is always accepted and the transition kernel is therefore given directly below.

Gibbs sampling begins with choosing one dimension to update, either by random (*random sweep Gibbs sampling*) or in turn (*systematic sweep Gibbs sampling*). This dimension, v_t, is then sampled according to the conditional distribution of π given that the other dimensions remain the same:

$$X_{t+1}^{(v_t)} \sim \pi_{X^{(v_t)}|X^{(-v_t)}}(\cdot|X_t^{(-v_t)}) \qquad X_{t+1}^{(-v_t)} = X_t^{(-v_t)} \, .$$

Worth noting is that for π with independent dimensions, one sweep through all dimensions is enough to get independence between the random variables in the Markov chain.

4 The hard-core model

In this section, an example is walked through showing a situation where a Gibbs sampler is natural to use. The major part of this section is borrowed directly from [Häggström]. This section primarily serves as an introduction to the next section where some problems are stated and solved in a straight forward way and to the section after that, where an alternative route is taken to solve those problems.

The hard core model was originally used to model physical conditions where two particles with non-zero width cannot exist arbitrarily close to each other. To form the state space, a graph $G = (V, E)$ is used. The vertexes describe where the centre of particles may be located and an edge between two vertexes describe that those vertexes cannot simultaneously contain particle centres. The state space for the Markov chain is then $\{0, 1\}^V$. Each member of the state space is called a configuration. The probabilistic model is that all configurations that are feasible, i.e. that do not contain 1:s at both ends of any edge in the graph, are equally probable. Hence each configuration ξ has probability $\frac{1}{Z_G}$ if it is feasible and 0 if it is not, where Z_G is the number of feasible configurations for the graph G.

Suppose now that we want to compute the estimated number of ones in the configurations: $\mathbb{E}(n(\xi))$. Even for moderately sized graphs, such as a 10 by 10 two-dimensional grid, the number of feasible configurations is daunting and the feasible configurations are hard to enumerate. Consequently, exact computation of the expected value is not a viable options. Direct Monte Carlo simulation through sampling from π_ξ is not readily performed either,

since the feasible configurations are hard to sample uniformly from. One solution is instead to perform MCMC using random sweep Gibbs sampling:

1. X_0 is chosen in some fashion among the feasible states (=configurations).

2. Since all feasible configurations are equally likely, the conditional distribution of one dimension (=vertex) given the others is $(\frac{1}{2}, \frac{1}{2})$ if a 1 at that vertex is feasible given the configuration of the other vertexes and $(1, 0)$ otherwise.

So for v_t randomly selected in V, if a change in vertex v_t (from 0 to 1) of X_t would yield it infeasible, then X_{t+1} is set to X_t. If the change is feasible on the other hand, $X_{t+1}^{(v_t)}$ is selected uniformly in $\{0, 1\}$, while $X_{t+1}^{(-v_t)}$ remains $X_t^{(-v_t)}$.

Through running a Markov chain as described above, discarding a burn-in sequence and possibly thinning the chain (as described in section 2), we can achieve an estimate of π_ξ. Using this, we can estimate the sought entity:

$$\mathbb{E}(n(\xi)) = \int n(\xi) d\pi_\xi(\xi) \hat{=} \int n(\xi) d\hat{\pi}_\xi(\xi) = \sum_{i=1}^{m} n(X_i) \cdot \frac{1}{m}$$

A generalised hard core model which allows for different "packing intensities" is introduced in problem 7.4 in [Häggström]. Here a parameter λ is introduced and the relative probability of two feasible states ξ_1 and ξ_2 is:

$$\frac{\pi_\xi^\lambda(\xi_1)}{\pi_\xi^\lambda(\xi_2)} = \frac{\lambda^{n(\xi_1)}}{\lambda^{n(\xi_2)}} = \lambda^{n(\xi_1)-n(\xi_2)} .$$

From this we can see that the corresponding random sweep Gibbs sampler would be based on the following: The conditional distribution of one dimension (=vertex) given the others is $(\frac{1}{1+\lambda}, \frac{\lambda}{1+\lambda})$ if a 1 at that vertex is feasible given the configuration of the other vertexes and $(1, 0)$ otherwise.

Comparing to the standard hard-core model, the tendency of the chain to jump to denser states is increased if $\lambda > 1$ and decreased if $\lambda < 1$.

5 Some problems on the hard-core model

In this section, a few problems on the hard-core model are introduced and solutions are outlined. In the next section a more refined approach to the solutions is introduced.

Suppose that we are able to make physical measurements of particle counts in a situation that can be modelled by the hard-core model with a given graph $G = (V, E)$. We first want to test whether $\lambda = 1$, i.e. if the standard hard-core model is suitable. If that hypothesis is rejected, i.e. $\lambda \neq 1$, we want to maximum likelihood estimate λ in the general hard-core model.

5.1 Testing the hypothesis $\lambda = 1$ using one measurement

We want to test the hypothesis $\lambda = 1$ through getting a p-value for one measurement, Y, of the particle count. This is arguably a weak way of testing, since only one measurement is used, but it will do to provide the idea.

For the sake of brevity let us assume that the alternative hypothesis is that λ is lower than 1, making it a one way test:

$$H_0 : \lambda = 1 \qquad H_A : \lambda < 1 \, .$$

We want to perform the test through examining:

$$p = \mathbb{P}(Y \leq y | \lambda = 1) = \pi_n^{G,1}([0, y]) \, ,$$

where $\pi_n^{G,\lambda}$ is the distribution of the number of 1:s in the hard-core model with parameter λ.

But, we can estimate $\pi_n^{G,\lambda}$ by using the Gibbs sampler MCMC algorithm introduced in the last section. Suppose that $\{X_i\}$ is the MCMC sequence with the burn-in sequence discarded and thinning performed in a proper way to make the X_i:s nearly independent. Further, let $Z_i = n(X_i)$ denote the particle counts in the MCMC sequence. $\pi_n^{G,\lambda}$ can then be estimated by:

$$\hat{\pi}_n^{G,\lambda}(A) = \frac{1}{n} \sum_{i=1}^{n} I_A(n(x_i)) = \frac{1}{n} \sum_{i=1}^{n} I_A(z_i) \, .$$

This would yield the following estimate of p:

$$\hat{p} = \hat{\pi}_n^{G,\lambda}([0, y]) = \frac{1}{m} \sum_{i=1}^{m} I_{[0,y]}(z_i) \, ,$$

that is the proportion of observed counts in the sequence that are at least as low as y. For independent $\{Z_i\}$, $\hat{p}$ is distributed according to a scaled binomial distribution:

$$\hat{p} \sim \frac{1}{m} \mathrm{Bin}(m, p)$$

One potential problem is that if p is very low, only a small proportion of $\{Z_i\}$ will fall in $[o, y]$ and a large length, m, of the sequence is necessary to get a precise estimate of p. One way of reporting the uncertainty of the p value, is for example to say that it is less than 0.05 with $(1 - \alpha) \cdot 100\%$ certainty, where the latter uncertainty comes from the MCMC estimation of the null-distribution.

5.2 Maximum likelihood estimation of λ

Suppose now that $\lambda = 1$ was rejected above and that the general hard-core model will be used instead. In order to use the general model, λ must be given a value[4]. To perform this, a series of independent physical measurements, $D = \{d_i\}$, are produced. One way of estimating λ is then through maximising the likelihood function:

$$\begin{aligned}
\hat{\lambda} &= \mathrm{argmax}_\lambda l(\lambda | X = D) \\
&= \mathrm{argmax}_\lambda \, \mathbb{P}(X = d | \lambda = \lambda) \\
&= \mathrm{argmax}_\lambda \prod_{i=1}^{m} \pi_n^{G,\lambda}(d_i) \, ,
\end{aligned}$$

where π is estimated by MCMC as in the previous section.

Since no analytical form of the likelihood function is available, one has to resort to numerical search for the maximum value. Since each new value of λ requires a completely new run of the MCMC algorithm with different jumping probabilities, this will demand substantial computing energy[5] even to get a rough estimate of λ.

6 An indirect estimation approach

In section 5 above, in some sense brute force was applied to solve both problems. In the first problem of estimating the p-value, the (thinned) chain

[4] not taking in account the Bayesian approach of giving it an a priori distribution

[5] i.e. computing power integrated over time

had to be very long to get a precise estimate in the case of a very small p-value. In the second problem, a series of MCMC runs had to be performed to get estimates of the distributions of the number of 1:s, each run with a different value of the parameter λ.

Below, a different methodology is suggested to estimate the distributions $\pi_n^{G,\lambda}$ indirectly, giving the user the freedom to (1) increase the precision when estimating the p-value and (2) to estimate the likelihood functions from a common run of the Markov chain, allowing for many more λ:s to be searched through much more rapidly by the numerical optimisation procedure.

In both examples the results of the following section is used.

6.1 The basic idea

If a distribution π on a space S is to be estimated, an alternative distribution π' is estimated through a Markov chain $(X_1, \ldots, X_n)$. Now, for each X_i a ratio r_{X_i} is recorded, accounting for the bias at that state:

$$r_{x_i} = \frac{\pi(x_i)}{\pi'(x_i)} \, .$$

Let us expand that notation. Assume the ratio $\frac{\pi}{\pi'}$ to be "piecewise constant", meaning that for any subset of the state space, $A \subseteq S$ there exist a finite partitioning

$$A = A_1 \cup \ldots \cup A_m \quad A_i \neq A_j \text{ for } i \neq j \, ,$$

such that:

$$\forall i \in (1, \ldots, m) : \forall A_1', A_2' \subseteq A_i : \frac{\pi(A_1')}{\pi'(A_1')} = \frac{\pi(A_2')}{\pi'(A_2')} \, .$$

Denote the ratio $r_B = \frac{\pi(B)}{\pi'(B)}$, where the ratio has to be constant over B for it to be defined. For brevity, $r_{\{x\}}$ is often denoted r_x for $x \in S$.

Now, the estimation of $\pi(A)$ for any $A \subseteq S$ can be performed through :

$$\begin{aligned}
\pi(A) = \pi(\bigcup_{i=1}^{m} A_i) &= \sum_{i=1}^{m} \pi(A_i) = \sum_{i=1}^{m} \frac{\pi(A_i)}{\pi'(A_i)} \pi'(A_i) \\
&\hat{=} \sum_{i=1}^{m} r_{A_i} \hat{\pi'}(A_i) = \sum_{i=1}^{m} r_{A_i} \frac{1}{n} \sum_{j=1}^{n} I_{A_i}(x_j) \\
&= \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{n} r_{A_i} I_{A_i}(x_j) = \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{n} r_{x_j} I_{A_i}(x_j) \\
&= \frac{1}{n} \sum_{j=1}^{n} r_{x_j} \sum_{i=1}^{m} I_{A_i}(x_j) = \frac{1}{n} \sum_{j=1}^{n} r_{x_j} I_A(x_j) \ .
\end{aligned}$$

A problem with this approach is that $\hat{\pi}$ is not bound to $[0, 1]$. For example, for $n = 1$, $\hat{\pi}(S) = r_1$ which for all underrepresented states is greater than 1. This problem stems from the fact that $\frac{\pi(A_i)}{\pi'(A_i)}$ is known while $\pi'(A_i)$ is estimated.

In the previous part of the report, the distribution has been estimated by the proportion of elements in the chain falling in the argument, i.e. $\hat{\pi}(A) = \frac{1}{n} \sum_{i=1}^{n} I_A(x_i)$. A direct approach to indirectly estimate π through a sequence $(X_1, \ldots, X_n)$ from π' would analogously be:

$$\hat{\pi}(A) = \frac{1}{\sum_{i=1}^{n} r_{x_i}} \sum_{i=1}^{n} r_{x_i} I_A(x_i) \ ,$$

i.e. a *weighted proportion*. This estimate is bounded by $[0, 1]$. Additionally, it is sufficient to know scaled ratios $r'_{x_i} = C \cdot r_{x_i}$ to perform the estimation.

The two methods above do converge when $n \to \infty$. First:

$$\mathbb{E}\left[r_X\right] = \int r_x \pi'(x) dx = \int \frac{\pi(x)}{\pi'(x)} \pi'(x) dx = \int \pi(x) dx = 1 \ ,$$

then, the law of large numbers proves that the ratio of the estimates converges to 1 almost surely.

In some situations the ratios $r_{x_j} = \frac{\pi(x_j)}{\pi'(x_j)}$ are not known directly. Instead for any pair (x', x'') of members of the sequence, the relative ratio

$$R_{x',x''} := \frac{r_{x'}}{r_{x''}}$$

is available. When using the weighted proportion method above, we can be satisfied with learning proportional ratios $r'_{x_j} = C \cdot r_{x_j}$, since the constant C is cancelled out in the ratio. In that case, knowing the R:s is in fact sufficient. Choosing $C = \frac{1}{r_{x_1}}$,

$$r'_{x_j} = \frac{1}{r_{x_1}} r_{x_j} = R_{x_j,x_1} \, .$$

6.2 Testing the hypothesis $\lambda = 1$ using one measurement (using indirect estimation)

A problem with the estimation procedure in section 5.1 is that the only a small proportion of the elements in the Markov chain have a count of 1:s below the observed count y. Therefore the chain has to be run for a long time to get good precision in the estimate of the sought p-value:

$$p = \mathbb{P}(Y \leq y | \lambda = 1) = \pi_n^{G,1}([0,y]) \hat{=} \hat{\pi}_n^{G,1}([0,y]) = \frac{1}{n} \sum_{i=1}^{n} I_{[0,y]}(z_i) \, ,$$

where Z_i is the number of 1:s in the element X_i of the Markov chain.

Using the indirect estimation outlined above (section 6.1), an alternative chain could be run creating modified amounts of 1:s below and above y. Assuming independence in the thinned chain of length n, the variance of the original estimate would be[6] :

$$\mathrm{Var}(\hat{p}) = \mathrm{Var}(\frac{1}{n} \mathrm{Bin}(n,p)) = \frac{1}{n^2} \cdot np(1-p) = \frac{p(1-p)}{n} \, .$$

If an alternative distribution, π'_n, would be used with a different chance of getting no more than y 1:s, $p' = \hat{\pi}'_n([0,y])$, with a known ratio:

$$r = r_{\{\xi : n(\xi) \in [0,y]\}} = \frac{p}{p'} \, ,$$

the estimate of p would be:

$$p = rp' = r\pi'_n([0,y]) \Rightarrow \hat{p}_a = r\hat{\pi}'_n([0,y]) = r\frac{1}{n} \sum_{i=1}^{n} I_{[0,y]}(z'_i) \sim \frac{r}{n} \mathrm{Bin}(n, \frac{p}{r}) \, ,$$

[6]The dependence decreases as the amount of thinning increases (cf section 2.2). The approximation is good if $\frac{\mathrm{Var}(\sum U_i)}{n \, \mathrm{Var}(U_i)} \approx 1$, i.e. $\sum_{i \neq j} \mathrm{Cov}(U_i, U_j) \ll n \, \mathrm{Var}(U)$, where U_i are the $0-1$ random variables $U_i = I_{[0,y]}(n(X_i))$.

where Z_i' is the number of 1:s in the state X_i' in the alternative Markov chain having stationary distribution π_n' and where $\hat{p}_a$ is the estimate of p stemming from the alternatively distributed chain. Now the variance of the alternative estimate $\hat{p}_a$ is:

$$\mathrm{Var}(\hat{p}_a) = \mathrm{Var}\left(\frac{r}{n}\,\mathrm{Bin}(n, \frac{p}{r})\right) = \frac{r^2}{n^2} \cdot n\frac{p}{r}(1 - \frac{p}{r}) = \frac{rp(1 - \frac{p}{r})}{n}\,. \tag{1}$$

In what cases will the alternative estimation result in a lower variance of the estimate?

$$\begin{aligned}\mathrm{Var}(\hat{p}_a) \leq \mathrm{Var}(\hat{p}) \Leftrightarrow \frac{rp(1 - \frac{p}{r})}{n} \leq \frac{p(1-p)}{n} \Leftrightarrow r(1 - \frac{p}{r}) \leq 1 - p \\ \Leftrightarrow r \leq 1 \Leftrightarrow p \leq p'\,,\end{aligned}$$

i.e. the bigger inflation of the p-value in the alternative distribution the better.

Now, it is difficult to create an alternative distribution with a given ratio $r = \frac{p}{p'} = \frac{\pi_n^{G,1}([0,y])}{\pi_n'([0,y])}$. Instead we can define a distribution, π_n', to be estimated by selecting a ratio of ratios:

$$R = \frac{\pi_n'([0,y])}{\pi_n'((y,\infty))} \Big/ \frac{\pi_n^{G,1}([0,y])}{\pi_n^{G,1}((y,\infty))}\,,$$

then use Gibbs sampling to create a sequence $\{X_i\}$ with π_n' as stationary distribution and finally use the weighted proportion method from section 6.1 to estimate the p-value:

$$\begin{aligned}\hat{p}_a = \hat{\pi}_n^{G,1}([0,y]) &= \frac{1}{\sum_{i=1}^n r_{x_i}} \sum_{i=1}^n r_{x_i} I_{[0,y]}(x_i) = \{N := \sum_{i=1}^n I_{[0,y]}(x_i)\} \\ &= \frac{N}{N + (n-N)\cdot R}\,,\end{aligned} \tag{2}$$

where $N \sim \mathrm{Bin}(n, p')$.

Since the formula for the variance of $\hat{p}_a$ in equation 2 is not as simple as of the scaled binomial one in equation 1 and since p' is only determined implicitly through R ($R = \frac{1/p'-1}{1/p-1}$), an optimal R would need prior information on p. However, if one is most interested if p exceeds a certain threshold, p_0, such as 0.01, R could be chosen to give minimal variance for $p = p_0$ since for p much greater or less the accuracy is not as important. An optimal R is not shown here even for p being known, but it should be possible to obtain by simulation if not directly. However, it is not included in this report due to time constraints.

6.3 Maximum likelihood estimation of λ (using indirect estimation)

In section 5.2, λ was estimated through maximising the likelihood function:

$$\begin{aligned}\hat{\lambda} &= \text{argmax}_\lambda l(\lambda|X=x) \\ &= \text{argmax}_\lambda \, \mathbb{P}(X=x|\lambda=\lambda) \\ &= \text{argmax}_\lambda \prod_{i=1}^{m} \pi_n^{G,\lambda}(x_i) \,,\end{aligned}$$

where $\pi_n^{G,\lambda}$ was estimated with one MCMC run for each examined value of λ. One obvious problem in this setting is that it might only be possible to examine a few values of λ, since estimating $\pi_n^{G,\lambda}$ can be time consuming.

Taking the indirect estimation approach, $\pi_n^{G,1}$ $(\lambda = 1)$ can be treated as the alternative distribution that has been estimated. Using the notation of section 6.1:

$$r_s := \frac{\pi_n^{G,\lambda}(s)}{\pi_n^{G,1}(s)} \qquad r'_s := \frac{r_s}{r_{s^0}} \,,$$

where s^n is defined as an arbitrary state with n 1:s, r'_s can be shown to be:

$$r'_s = \lambda^{n(s)} \,,$$

for an arbitrary state s.

This can be shown by induction:

- Base case (s^0):

$$r'_{s^0} = \frac{r_{s^0}}{r_{s^0}} = 1 = \lambda^0$$

- Induction step (assume for s^i, show for s^{i+1}):

$$r'_{s^{i+1}} = \frac{r_{s^{i+1}}}{r_{s^0}} = \frac{r_{s^{i+1}}}{r_{s^i}} \Big/ \frac{r_{s^i}}{r_{s^0}} = \lambda\lambda^i = \lambda^{i+1} \,,$$

since:

$$\begin{aligned}\frac{r_{s^{i+1}}}{r_{s^i}} &= \frac{\pi_n^{G,\lambda}(s^{i+1})/\pi_n^{G,1}(s^{i+1})}{\pi_n^{G,\lambda}(s^i)/\pi_n^{G,1}(s^i)} \\ &= \frac{\pi_n^{G,\lambda}(s^{i+1})}{\pi_n^{G,\lambda}(s^i)} \Big/ \frac{\pi_n^{G,1}(s^{i+1})}{\pi_n^{G,1}(s^i)} =_1 \lambda/1 \\ &= \lambda \,,\end{aligned}$$

where $=_1$ follows by the definition of $\pi_n^{G,\lambda}$.

Now, since r'_{x_i} are known, the weighted proportion method from section 6.1 can be used to estimate $\pi_n^{G,\lambda}$ from the estimate of $\pi_n^{G,1}$. Therefore, the numerical maximisation of the likelihood function do not need to perform a novel MCMC run for each examined value of λ. Therefore, more steps of the maximisation procedure can be performed in less time, hopefully producing a λ with higher likelihood.

One issue with this approach is that most of the "emphasis" of the original Markov chain will be on states that are typical in the distribution for $\lambda = 1$. In an extreme example, where $\lambda \ll 1$, the observed number of particles, D, might typically be very low, while the Markov chain typically has a very small proportion of elements with such a low count of 1:s. Therefore, estimates such as $\pi_n^{G,1}(s^0)$ underlying $\pi_n^{G,\lambda}(s^0)$ will be uncertain. These circumstances should be watched out for and if needed, a novel alternative MCMC run should be performed with a $\lambda = \lambda_0$ closer to the preliminary estimate produced from $\pi_n^{G,1}$. $r'_s = (\lambda/\lambda_0)^{n(s)}$ would then be the correcting factor to use.

7 Bibliography

[Gilks] Gilks, W., Richardson, S. & Spiegelhalter, D. (1996) *Markov Chain Monte Carlo in Practice*, Chapman & Hall, London.

[Häggström] Häggström, O. (2002) *Finite Markov Chains and Algorithmic Applications*, Cambridge University Press, Cambridge.