Change-point models

with Hamiltonian and reversible jump
Markov chain Monte Carlo

March 2018

1 Introduction

In change-point analysis, one assumes that a given time series has been generated by an underlying stochastic process, and the goal is to determine when and how the rate of the process might have changed during the recorded interval. Here, we assume that the interval can be partitioned into a number of subintervals, over each of which the rate remains constant.

Specifically, the focus here is on reimplementing Green's solution for the Poisson-process change-point problem, which was used to demonstrate the practicality of reversible jump Markov chain Monte Carlo (RJMC).[1] Instead of relying on Gibbs sampling for within-model moves, however, we will make use of Hamiltonian Monte Carlo (HMC).[2]

2 Model description

Let $\mathbf{y} = (y_1, \ldots, y_n)$ be a sequence of observed data points within a time window $[0, L]$. We assume that the points are generated by a non-homogeneous Poisson process with rate function $r(t)$, which implies that the log-likelihood of r (following (12)) is:

$$\begin{aligned}\log \mathrm{P}(\mathbf{y} \mid L, r) &= \log \left[\exp\left(-\int_0^L r(s)\, ds \right) \prod_{j=1}^{n} r(y_j) \right] \\ &= \sum_{j=1}^{n} \log r(y_j) - \int_0^L r(s)\, ds.\end{aligned}$$

As mentioned above, we assume here that r is a step function with a finite number of steps; say at positions $s_1, s_2 \ldots, s_k \in (0, L)$ such that $s_i < s_{i+1}$.

[1] Peter Green, *Reversible jump Markov chain Monte Carlo computation and Bayesian model determination*, 1995.

[2] Radford Neal, *MCMC Using Hamiltonian Dynamics*, in *Handbook of Markov Chain Monte Carlo*, 2011.

For the sake of convenience, we let $s_0 = 0$ and $s_{k+1} = L$. Then if the value of r over the interval $[s_i, s_{i+1})$ is $h_i = r(s_i)$, and the number of data points within that same interval is n_i, the above log-likelihood becomes:

$$\log \mathrm{P}(\mathbf{y} \mid k, \mathbf{s}, \mathbf{h}) = \sum_{i=0}^{k} n_i \log h_i - \sum_{i=0}^{k} (s_{i+1} - s_i) h_i. \tag{1}$$

We will be sampling, approximately, from the distribution $\mathrm{P}(k, \mathbf{s}, \mathbf{h} \mid \mathbf{y})$, which is related to the likelihood function by Bayes' rule:

$$\mathrm{P}(k, \mathbf{s}, \mathbf{h} \mid \mathbf{y}) = \mathrm{P}(\mathbf{y} \mid k, \mathbf{s}, \mathbf{h})\, \mathrm{P}(\mathbf{s}, \mathbf{h} \mid k)\, \mathrm{P}(k)/Z,$$

where $Z = \mathrm{P}(\mathbf{y})$ is an intractable normalisation constant. Note that for the sampling to be valid, there can be only one normalisation factor; that is, Z cannot depend on k or any of the other parameters. In particular, an unnormalised prior $\mathrm{P}(\mathbf{s}, \mathbf{h} \mid k) \cdot W$ may only be used if its absent normalisation constant W is independent of the model degree k.

2.1 Priors

If it is our belief that simpler rate functions are somewhat more likely to have generated the data than more intricate ones, we might make use of a Poisson distribution as a prior for the number of change points k:

$$\mathrm{P}(k) = e^{-\lambda} \frac{\lambda^k}{k!}, \quad \lambda > 0. \tag{2}$$

In line with the remarks of the previous paragraph, we may even restrict k to some finite set $\{0, \ldots, K\}$ with no change to (2), because the resulting normalisation factor is independent of any of the model parameters.

The Poisson prior is used by Green, but one could conceivably motivate the use of more relaxed prior distributions as well—for example the logarithmic distribution $\mathrm{P}(k) \sim \lambda^k/k$, or even the uniform distribution $\mathrm{P}(k) \sim 1$.

When it comes to the step prior $\mathrm{P}(\mathbf{s}, \mathbf{h} \mid k)$, there is considerably less freedom, since any normalisation constants will almost invariably depend on the vector length k. Green's implementation assumes the independence of $\mathbf{s}$ and $\mathbf{h}$, and makes use of the even-numbered order statistics of $2k + 1$ uniform variables, and independent gamma random variables, for $\mathbf{s}$ and $\mathbf{h}$ respectively. It might be worth noting that if one were to use the distribution of the order statistics of k uniform random variables, the resulting prior would itself be uniform, because the relevant density is simply the probability of k specific points appearing in any order:

$$\mathrm{P}(\mathbf{s} \mid k) = k! \prod_{i=1}^{k} \mathrm{P}(s_i) = \frac{k!}{L^k}.$$

The distribution of the even-numbered statistics in a sample of size $2k+1$, however, also accounts for a single arrival in-between every pair of points, and ends up favouring evenly-spaced step positions:

$$\begin{aligned} \mathrm{P}(\mathbf{s} \mid k) &= (2k+1)! \prod_{i=0}^{k} \frac{s_{i+1} - s_i}{L} \prod_{i=1}^{k} \mathrm{P}(s_i) \\ &= \frac{(2k+1)!}{L^{2k+1}} \prod_{i=0}^{k} (s_{i+1} - s_i). \end{aligned} \tag{3}$$

A straightforward prior distribution for the height vector $\mathbf{h}$ is one in which the values $h_0, \ldots, h_k$ are assumed to be independent and identically distributed according to some standard distribution, e.g., the gamma distribution $\Gamma(\alpha, \beta)$:

$$\mathrm{P}(\mathbf{h} \mid k) = \prod_{i=0}^{k} \Gamma(\alpha, \beta) = \left(\frac{\beta^\alpha}{\Gamma(\alpha)} \right)^k \exp\left(-\beta \sum_{i=0}^{k} h_i \right) \prod_{i=0}^{k} h_i^{\alpha-1}. \tag{4}$$

Other continuous distributions could be used in a similar way.

3 The Markov chain Monte Carlo process

At any given stage of the sampling process, there are a few possible moves that can be made: the creation or deletion of a single change point, which affects the dimensionality k of the model; and an update to $\mathbf{s}$ and $\mathbf{h}$, which does not. In a Markov chain Monte Carlo simulation, disparate moves such as these can be combined to form a valid transition, as long as the combination does not violate the reversibility of the underlying chain. In particular, applying moves in a fixed sequence or with set probabilities is allowed.

Strictly speaking, the creation and deletion moves that are available for different values of k are not equivalent, so there are in fact many more than three or four moves that make up the transition—possibly a countable infinite number of them. It is perfectly fine, however, to let the availability—or probability—of any single move depend on the simulation's current state $(k, \mathbf{s}, \mathbf{h})$.

This means that when k is, say, four, we can restrict the set of possible creation moves to the addition of a fifth change point, and ignore moves that take k from five to six, etc. There are thus only three available moves for most values of k: the creation move $k \mapsto k+1$, the deletion $k \mapsto k-1$, and the Hamiltonian update of $(\mathbf{s}, \mathbf{h})$. All other moves are assigned probability zero. The only exceptions to this are when $k = 0$, at which point no deletion move is available, and $k = K$ (if a maximum is set), where no creation may occur.

In this implementation, each transition will involve a within-model update followed either by a creation or a deletion move. Both the within-model

and cross-model moves are simply proposals, and are not guaranteed to be accepted (although one expects the chance of acceptance for Hamiltonian moves to be suitably high). When it comes to the reversible jump moves, Green uses probabilities b_k and d_k to choose between creation and deletion, respectively, subject to:

$$b_k \, \mathrm{P}(k) = d_{k+1} \, \mathrm{P}(k+1),$$

where $\mathrm{P}(k)$ is as in (2).

4 Hamiltonian moves

Hamiltonian Monte Carlo leverages the gradient of the log-likelihood function to propose moves that are both non-random (in that they do not resemble those of a random walk) and highly likely to be accepted. The idea behind to HMC is to extend the state space with additional momentum variables—one for each parameter being sampled—and perform a simulation of Hamiltonian dynamics on the combined state space.

The Hamiltonian is an energy function that combines the potential energy of position variables q with the kinetic energy of momentum variables p, usually additively:

$$H(q, p) = U(q) + K(p).$$

In our case, the canonical distribution of the potential energy function is the posterior distribution we're interested in sampling from (an energy function is related to its canonical distribution by a negative logarithm, plus any convenient constant). For step heights, say:

$$U(\mathbf{h} \mid \mathbf{s}, k, \mathbf{y}) = -\log \mathrm{P}(\mathbf{h} \mid \mathbf{s}, k, \mathbf{y}) - \log Z_U, \tag{5}$$

We are free to define the kinetic energy function as we choose, but it is common to do so such that its canonical distribution is a multivariate Gaussian with independent components, each of which has mean zero and its own variance m_i:

$$K(p) = \sum_{i=0}^{k} \frac{p_i^2}{2m_i}. \tag{6}$$

Since the Hamiltonian is an energy function as well, it has a canonical distribution of its own:

$$\mathrm{P}(q, p) = \exp(-U(q)) \exp(-K(p))/Z_H,$$

in which the position and momentum variables q and p are seen to be independent. This is a key property of HMC: it means that if we sample from $\mathrm{P}(q, p)$ and discard the sampled momenta, we are left with a set of

positions that is itself a valid sample from the canonical disitribution of U, i.e., the posterior distribution we're interested in.

Like any other implementation of the general Metropolis-Hastings algorithm, HMC involves proposing a new state (q', p') and accepting it with probability

$$\begin{aligned}\pi &= \min(1, \mathrm{P}(q', p') / \mathrm{P}(q, p)) \\ &= \min(1, \exp(U(q) - U(q') + K(p) - K(p'))).\end{aligned}$$

This new state is reached via Hamiltonian dynamics, which proceed with time according to Hamilton's equations:

$$\begin{aligned}\frac{dq_i}{dt} &= \frac{\partial H}{\partial p_i}, \\ \frac{dp_i}{dt} &= -\frac{\partial H}{\partial q_i}.\end{aligned} \tag{7}$$

Theoretically, these dynamics preserve energy, so that $H(q, p) = H(q', p')$, and, in particular, $\pi = 1$. Of course in practice they must be followed approximately—in a series of discrete steps—and it is this discretisation that introduces change (hopefully small) into the energy function, necessitating the acceptance step.

4.1 Hamilton's equations

The elements needed to implement HMC are a Hamiltonian H and its partial derivatives, as in (7). The first consideration to make is that of which posterior should be sampled from—that is, which parameters should be updated by the move. In our case, the options are the step locations $\mathbf{s}$, their heights $\mathbf{h}$, or the combined vector $(\mathbf{s}, \mathbf{h})$. (It will turn out that the step heights are the most amenable of these to HMC, but it is worth considering all three cases.)

The posteriors for the three vectors mentioned above are:

$$\begin{aligned}\mathrm{P}(\mathbf{s} \mid \mathbf{h}, k, \mathbf{y}) &= \mathrm{P}(\mathbf{y} \mid k, \mathbf{s}, \mathbf{h})\, \mathrm{P}(\mathbf{s} \mid k) / Z_s, \\ \mathrm{P}(\mathbf{h} \mid \mathbf{s}, k, \mathbf{y}) &= \mathrm{P}(\mathbf{y} \mid k, \mathbf{s}, \mathbf{h})\, \mathrm{P}(\mathbf{h} \mid k) / Z_h, \\ \mathrm{P}(\mathbf{s}, \mathbf{h} \mid k, \mathbf{y}) &= \mathrm{P}(\mathbf{y} \mid k, \mathbf{s}, \mathbf{h})\, \mathrm{P}(\mathbf{s} \mid k)\, \mathrm{P}(\mathbf{h} \mid k) / Z_k.\end{aligned}$$

Recall, however, that the Hamiltonion function $H(q, p)$ can be modified using any constant (relative to the state variables (q, p)) we deem convenient, because such a constant influences neither the partial derivatives nor the acceptance probability α. This implies that we may simply use the full posterior $\mathrm{P}(\mathbf{s}, \mathbf{h} \mid \cdot)$ in all three of the above cases, since $\mathrm{P}(\mathbf{h} \mid k)$ and $\mathrm{P}(\mathbf{s} \mid k)$ can be viewed as constants when updating only $\mathbf{s}$ or $\mathbf{h}$, respectively.

Combining the likelihood and prior expressions from Section 2, this posterior is:

$$\begin{aligned}\mathrm{P}(\mathbf{s}, \mathbf{h} \mid k, \mathbf{y}) \sim \exp\left(-\sum_{i=0}^{k}(s_{i+1} - s_i)h_i\right) \prod_{i=0}^{k} h_i^{n_i} \\ \times \exp\left(-\beta \sum_{i=0}^{k} h_i\right) \prod_{i=0}^{k} h_i^{\alpha-1} \prod_{i=0}^{k}(s_{i+1} - s_i),\end{aligned}$$

which implies that we may use the common potential energy function

$$U(\mathbf{s}, \mathbf{h}) = \sum_{i=0}^{k}\Big((s_{i+1} - s_i + \beta)h_i - (n_i + \alpha - 1)\log h_i - \log(s_{i+1} - s_i)\Big).$$

The second of Hamilton's equations—the partial derivative of H with respect to the position variables $\mathbf{s}$ and $\mathbf{h}$—follows in two parts:

$$\begin{aligned}\frac{\partial H}{\partial s_i} &= h_{i-1} - h_i + \frac{1}{s_{i+1} - s_i} - \frac{1}{s_i - s_{i-1}}, && i = 1, \ldots, k, \\ \frac{\partial H}{\partial h_i} &= (s_{i+1} - s_i + \beta) - (n_i + \alpha - 1)/h_i, && i = 0, \ldots, k.\end{aligned}$$

These equations, combined with the kinetic energy function given in (6), are enough for a first implementation of HMC[3]—for example with the variances of the momenta all set to $m_i = 1$.

4.2 Constrained parameters

Such a direct approach, however, has a few issues, the first of which is straightforward: the discrete application of Hamilton's equations (usually via so-called 'leapfrog' steps) does not respect any constraints placed on the state variables. In our case, each step height h_i is non-negative, and each location is restricted to the interval $(0, L)$, but it is quite possible for momentum to carry a variable across one of these boundaries if the step size is sufficiently large. Moreover, many of our computations require the step locations to be ordered, but HMC as we have currently implemented it is by no means guaranteed to respect such an ordering, since two locations may have their order reversed by a single leapfrog step.

A useful way of avoiding boundary issues is to transform the position variables to an unconstrained space before applying HMC. For example, instead of working with step heights $h_i \geq 0$, we might apply HMC to the element-wise logarithm of $\mathbf{h}$, since $\log h_i$ is an unrestricted real number. For variables that are bounded both above and below, such as $s_i \in (0, L)$, one can first transform them to the open unit interval and then apply the

[3]See Figure 2 of Neal, *MCMC Using Hamiltonian Dynamics*.

logit function $\log(x/(1-x))$, which maps $(0,1)$ onto the real line. The unconstrained step heights and locations are then, respectively:

$$\begin{aligned} \hat{h}_i = \log(h_i) &\implies h_i = \exp\big(\hat{h}_i\big), \\ \hat{s}_i = \log\left(\frac{s_i}{L - s_i}\right) &\implies s_i = \frac{L}{1 + \exp(-\hat{s}_i)}, \end{aligned} \tag{8}$$

since the inverse of the logit function is the logistic function $1/(1+e^{-x})$.

One can in fact go further than this and constrain each step height s_i to the interval (s_{i-1}, s_{i+1}) using a similar procedure[4], however this makes the resulting expressions for Hamilton's equations rather more technical, since each s_i is then a linear combination of $s_1, \ldots, s_{i-1}$. This dependency is not an issue when transforming between s_i and $\hat{s}_i$, but rather when computing derivatives, since once must now compute not only $ds_i/d\hat{s}_i$ for each i, but $ds_i/d\hat{s}_j$ for every $j < i$ as well. In our opinion it is simpler—and possibly more efficient—to simply check that the step heights are sorted after each leapfrog step, and reorder them when necessary.

In any case, variable transformations such as those for $\hat{h}_i$ and $\hat{s}_i$ must be accounted for in the density function in which they are used. Specifically, if the density function of X is $p(x)$, then that of the transformed variable $f(X)$ is:

$$p(f^{-1}(y)) \left| \frac{df^{-1}}{dy}(y) \right|.$$

For multivariate distributions, the determinant of the Jocabian matrix is needed instead; however if the transformed variables are independent—each depending on a unique untransformed variable—then the Jacobian matrix is diagonal, and its derivative is simply the product of the element-wise derivatives.

This implies that if we wish to deal with both $\hat{\mathbf{h}}$ and $\hat{\mathbf{s}}$ in an HMC move (which is only really necessary if we are updating step locations and heights simultaneously), the required posterior density would be:

$$\mathrm{P}(\hat{\mathbf{s}}, \hat{\mathbf{h}} \mid k, \mathbf{y}) = \mathrm{P}(\mathbf{s}, \mathbf{h} \mid k, \mathbf{y}) \prod_{i=0}^{k} \frac{dh_i}{d\hat{h}_i} \prod_{i=1}^{k} \frac{ds_i}{d\hat{s}_i}.$$

Here, $\mathbf{s}$ and $\mathbf{h}$ should be understood to mean the constrained values derived from the elements of $\hat{\mathbf{s}}$ and $\hat{\mathbf{h}}$. Also, if one were only updating, say, step heights $\hat{\mathbf{h}}$, the locations could simply be left in their constrained form $\mathbf{s}$, and the product involving location derivatives omitted.

[4] See Section 33 of the Stan Modeling Language user manual for more on transformations of constrained variables.

The derivatives involved in the above density follow from (8):

$$\frac{dh_i}{d\hat{h}_i} = \exp\bigl(\hat{h}_i\bigr) = h_i,$$
$$\frac{ds_i}{d\hat{s}_i} = \frac{L\exp(-\hat{s}_i)}{\bigl(1+\exp(-\hat{s}_i)\bigr)^2} = s_i(L-s_i)/L,$$

and the final posterior distribution satisfies:

$$\mathrm{P}(\hat{\mathbf{s}},\hat{\mathbf{h}} \mid k,\mathbf{y}) \sim \exp\left(-\sum_{i=0}^{k}(s_{i+1}-s_i+\beta)h_i\right)\prod_{i=0}^{k} h_i^{n_i+\alpha}$$
$$\times \prod_{i=0}^{k}(s_{i+1}-s_i)\prod_{i=1}^{k} s_i(L-s_i).$$

Once again, this leads to a relatively simple potential energy function that can be used as part of the HMC algorithm:

$$\begin{aligned} U(\mathbf{s},\mathbf{h}) = &\sum_{i=0}^{k}\Bigl((s_{i+1}-s_i+\beta)h_i - (n_i+\alpha)\log h_i - \log(s_{i+1}-s_i)\Bigr) \\ &- \sum_{i=1}^{k}\log\bigl(s_i(L-s_i)\bigr). \end{aligned} \tag{9}$$

The partial derivatives of the Hamiltonian with respect to the position variables are then:

$$\begin{aligned} \frac{\partial H}{\partial \hat{s}_i} &= \left(h_{i-1} - h_i + \frac{1}{s_{i+1}-s_i} - \frac{1}{s_i - s_{i-1}} - \frac{L-2s_i}{s_i(L-s_i)}\right)\frac{ds_i}{d\hat{s}_i}, \\ \frac{\partial H}{\partial \hat{h}_i} &= (s_{i+1}-s_i+\beta)h_i - (n_i+\alpha). \end{aligned} \tag{10}$$

4.3 Performance remarks

As alluded to earlier, the step heights $\mathbf{h}$ are more amenable to HMC than the locations $\mathbf{s}$ or the combined vector $(\mathbf{s},\mathbf{h})$. This is primarily due to the presence of data counts n_i in the potential energy function U, because every time s_i is shifted across an arrival time y_j, both n_{i-1} and n_i are changed by one, and U by (roughly) $\log h_i$. This implies that U is discontinuous, relative to $\mathbf{s}$, at each datum y_j.

This does not mean that HMC will not work—the algorithm only requires the partial derivatives to exist on a region of probability one, and the set of points at which discontinuities occur here are those states $(\mathbf{s},\mathbf{h})$ for which $s_i = y_j$ for some i and j (a region of probability zero). At all other points, the data counts n_i may simply be treated as constants (as we have done in deriving (10)).

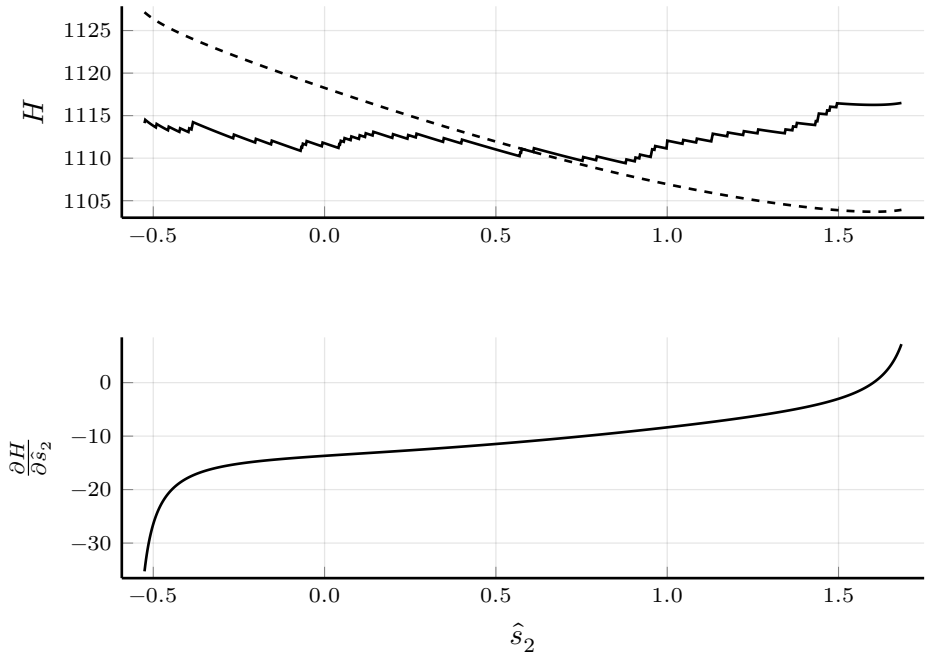

Figure 1: An example of the issue caused by the discontinuities in $U(\hat{\mathbf{s}}, \hat{\mathbf{h}})$: the first plot shows the value of the Hamiltonian as a single step location s_2 is varied, while the second shows its partial derivative. The dashed line in the first plot is the integral of this derivative—i.e., what H would look like were it smooth. This example was generated from Green's coal-mining data set, with k set to three.

The problem, however, is that the discontinuities that arise in H are significant enough to influence its overall behaviour, which is then reasonably likely to be at odds with its partial derivatives; see Figure 1. Since the leapfrog steps that make up an HMC move are guided by these derivative functions, it is reasonable to expect the algorithm's performance in exploring the state space to suffer when the change implied by the derivatives diverges from the Hamiltonian's real behaviour.

In addition to this, recall that it is the discrepancies that arise from applying Hamilton's equations in discrete steps that reduce the acceptance probability of an HMC move. The step-like nature of H simply adds to this error, since when $\mathbf{s}$ is close to a discontinuity, the change to H that arises from updating the position variables can no longer be accounted for by $\partial H / \partial \mathbf{s}$ alone, no matter how small the step size.

For these reasons, HMC updates of step locations appear to be particularly sensitive to the chosen leapfrog step size and count, with moves involving large sizes or many steps reducing the acceptance rate significantly.

(In the case of the coal-mining disaster data set, for example, we settled on a step size and count of .006 and 5, respectively.)

Since the step heights $\mathbf{h}$ do not suffer from the issues outlined above, it makes sense to update them independently of the step locations. This is because when the variables are combined, the low acceptance rate of the locations hinders the progress of the entire sampling process, and thus the step heights are updated less frequently than can be achieved when they are sampled separately.

5 Reversible jump moves

Apart from the location and height updates described above, there are two reversible-jump moves that alter the number of change points (steps) during the simulation, allowing it to sample over the extended state space $(k, \mathbf{s}, \mathbf{h})$. Although these 'birth' and 'death' moves have been implemented exactly as described in Green's paper, it might be worth describing them here as well.

Each birth move involves the creation of a new change point at a location chosen randomly from $[0, L]$, say $s^* \in [s_i, s_{i+1})$. If accepted, such a move would create two new intervals $[s_i, s^*)$ and $[s^*, s_{i+1})$, and replace the step height h_i with two new heights h'_i and h'_{i+1}. It is here where the so-called 'dimension matching' constraint of reversible jump MCMC is applied: we are free to propose values for the new parameters (s^*, h'_i, h'_{i+1}) however we wish, as long as the same (but inverted) method is used in the corresponding reverse move (in this case, the 'death' move).

Generally, this constraint is satisfied by defining an invertible mapping on the larger of the two state spaces, and augmenting the smaller space with random numbers as necessary. In this case, we wish the values of the new step heights to be informed by the one being replaced; Green makes use of a weighted geometric mean relationship:

$$(s^* - s_i) \log h'_i + (s_{i+1} - s^*) \log h'_{i+1} = (s_{i+1} - s_i) \log h_i. \tag{11}$$

A single random value u can be combined with h_i before applying the mapping, and here that number is used to determine the ratio between the new heights:

$$\frac{h'_{i+1}}{h'_i} = \frac{1-u}{u},$$

where $u \in [0, 1]$. The first new height, for example, is given by:

$$h'_i = h_i \Big(\frac{u}{1-u} \Big)^{(s_{i+1} - s^*)/(s_{i+1} - s_i)}.$$

When performing a death move, the replacement height h_i must be computed according to (11).

The acceptance probability for a birth move is the ratio of the new and old posterior probabilities, multiplied by a proposal ratio to account for the required moves, as well as a Jacobian factor that arises from the implicit change of variables that is required to compare the two states:

$$\pi = \min\left(1, \frac{\mathrm{P}(k+1, \mathbf{s}', \mathbf{h}' \mid \mathbf{y})}{\mathrm{P}(k, \mathbf{s}, \mathbf{h} \mid \mathbf{y})} \frac{d_{k+1} L}{b_k(k+1)} \left| \frac{\partial(\mathbf{s}', \mathbf{h}')}{\partial(\mathbf{s}, s^*, \mathbf{h}, u)} \right| \right),$$

Note that the proposal ratio includes factors L and $k+1$ to account for the choice of u and a change point to remove, respectively. The posterior ratio can be split into a product of two terms; a prior ratio:

$$\begin{aligned} \frac{\mathrm{P}(k+1)}{\mathrm{P}(k)} \frac{\mathrm{P}(\mathbf{s}', \mathbf{h}' \mid k+1)}{\mathrm{P}(\mathbf{s}, \mathbf{h} \mid k)} = \frac{\lambda}{k+1} \times \frac{(2k+2)(2k+3)}{L^2} \frac{(s^* - s_i)(s_{i+1} - s^*)}{s_{i+1} - s_i} \\ \times \exp(\beta(h'_i + h'_{i+1} - h_i)) \left(\frac{h'_i h'_{i+1}}{h_i} \right)^{\alpha - 1}, \end{aligned}$$

and a likelihood ratio $\mathrm{P}(\mathbf{y} \mid k+1, \mathbf{s}', \mathbf{h}') / \mathrm{P}(\mathbf{y} \mid k, \mathbf{s}, \mathbf{h})$, which is more easily computed via its logarithm:

$$\begin{aligned} \log\left(\frac{\mathrm{P}(\mathbf{y} \mid k+1, \mathbf{s}', \mathbf{h}')}{\mathrm{P}(\mathbf{y} \mid k, \mathbf{s}, \mathbf{h})} \right) = n'_i \log\left(\frac{h'_i}{h_i} \right) - (s^* - s_i)(h'_i - h_i) \\ + n'_{i+1} \log\left(\frac{h'_{i+1}}{h_i} \right) - (s_{i+1} - s^*)(h'_{i+1} - h_i). \end{aligned}$$

The last term in the acceptance ratio is the absolute value of the determinant of the Jacobian matrix for the *inverse* transformation from $(\mathbf{s}', \mathbf{h}')$ to $(\mathbf{s}, \mathbf{h}, s^*, u)$. Letting $r = (s_{i+1} - s^*)/(s_{i+1} - s_i)$:

$$\begin{aligned} \frac{\partial(\mathbf{s}', \mathbf{h}')}{\partial(\mathbf{s}, s^*, \mathbf{h}, u)} &= \begin{vmatrix} \dfrac{\partial h'_i}{\partial h_i} & \dfrac{\partial h'_i}{\partial u} \\ \dfrac{\partial h'_{i+1}}{\partial h_i} & \dfrac{\partial h'_{i+1}}{\partial u} \end{vmatrix} \\ &= \begin{vmatrix} \left(\dfrac{u}{1-u} \right)^r & h_i \left(\dfrac{u}{1-u} \right)^{r-1} \dfrac{r}{(1-u)^2} \\ \left(\dfrac{u}{1-u} \right)^{r-1} & h_i \left(\dfrac{u}{1-u} \right)^{r-2} \dfrac{r-1}{(1-u)^2} \end{vmatrix} \\ &= \left| -h_i \frac{u^{2r-2}}{(1-u)^{2r}} \right| \\ &= h_i \frac{\left(u \cdot u^{r-1} + (1-u) u^{r-1} \right)^2}{(1-u)^{2r}} \\ &= \frac{\left(h'_i + h'_{i+1} \right)^2}{h_i}. \end{aligned}$$

These are all of the expressions required for the implementation of both reversible-jump moves—the acceptance ratio for the death move being simply the inverse of that given for the birth move.

Apart from allowing us to sample from the full posterior $\mathrm{P}(k, \mathbf{s}, \mathbf{h} \mid \mathbf{y})$, one of the most important reasons for incorporating reversible-jump moves into an MCMC simulation is the effect they can have on state mixing: especially in the case of difficult simulations (such as that of step locations via HMC), these dimension-changing moves appear to increase the rate of convergence noticeably. (On the coal-mining data set, it proved more useful to enable RJMCMC moves and condition on $k = 3$, discarding irrelevant samples, than to perform a simulation with k fixed at 3.)

6 The Poisson process

A stationary Poisson process is characterised by what is known as the *strong renewal property*: at any fixed time—including event, or *arrival* times—the probabilistic process must restart, continuing independently of the past. Because this holds for arrival times, the sequence $(X_1, X_2, \dots)$ of inter-arrival times is independent and identically distributed. The fact that it holds for fixed times as well implies that the process is *memoryless*: if $X \in [0, \infty)$ denotes the inter-arrival time, then for all $s, t \in [0, \infty)$,

$$\mathrm{P}(X > t + s \mid X > s) = \mathrm{P}(X > t).$$

The memoryless property is actually strong enough to determine the distribution function of a random variable—in this case that of X.

Lemma 1. *The inter-arrival time X of a Poisson process follows an exponential distribution; i.e., for some $r \in (0, \infty)$, the distribution function of X is*

$$F(t) = 1 - e^{-rt}, \quad t \in [0, \infty),$$

and its probability density function is

$$f(t) = re^{-rt}, \quad t \in [0, \infty).$$

The function f is decreasing on $[0, \infty)$. The mean and variance of X are $1/r$ and $1/r^2$ respectively.

The parameter r is the *rate* of both the Poisson process and the above exponential distribution; its inverse $1/r$ is the *scale* parameter. In particular, arrivals occur at an average rate of r per unit of time.

The sequence of inter-arrival times determines another sequence of interest: that of actual arrival times $(T_1, \dots)$. We have:

$$T_n = \sum_{i=1}^{n} X_i, \quad n \in \mathbb{N}.$$

The distribution of the nth arrival time T_n is also determined by that of X.

Lemma 2. *The nth arrival time T_n of a Poisson process with rate r follows a gamma distribution with shape parameter n and rate r, and probability density function*

$$f_n(t) = r^n \frac{t^{n-1}}{(n-1)!}, \quad t \in [0, \infty).$$

The mean and variance of T_n are n/r and n/r^2 respectively.

The final parameter of interest is the counting process $(N_t, t \geq 0)$, in which N_t denotes the number of arrivals in the interval $(0, t]$.

Lemma 3. *The number of arrivals before time t in a Poisson process of rate r follows a Poisson distribution with parameter rt. The relevant probability mass function is*

$$\mathrm{P}(N_t = n) = e^{-rt} \frac{(rt)^n}{n!}.$$

Both the mean and variance of N_t are equal to rt.

The Poisson process can also be characterised in terms of increments of the counting process: increments are *stationary*, in that if $s < t$, then $N_t - N_s$ and N_{t-s} have the same distribution; as well as *independent*, so that if $t_1 < t_2 < \cdots$, the sequence $(N_{t_1}, N_{t_2 - t_1}, \dots)$ is an independent one.

6.1 Non-homogeneous Poisson processes

In a non-homogeneous Poisson process, the rate parameter r is no longer constant, but a function $r(t)$ of time. In this context, it is useful to rephrase both the counting process and the property of independent increments in terms of measurable subsets of the real half-line, and to define a so-called *mean function* m. Let $N(A)$ denote the number of arrivals in a measurable subset $A \subseteq [0, \infty)$, and set

$$m(A) = \int_A r(s)\, ds.$$

Definition 1. Let N be the counting process (measure) of a stochastic process in time, and $(A_1, A_2, \dots)$ a countable sequence of disjoint, measurable subsets of $[0, \infty)$. Then the process is a non-homogeneous Poisson process with rate function $r(t)$ if $(N(A_1), N(A_2), \dots)$ is a sequence of independent random variables, and if $N(A)$ has a Poisson distribution with parameter $m(A)$ for any measurable $A \subseteq [0, \infty)$.

In particular, the number of arrivals that fall within a given time window A has the probability mass function

$$\mathrm{P}(N(A) = n) = e^{-m(A)} \frac{m(A)^n}{n!},$$

and, conditioned on a single arrival in the time window A, the probability of that arrival occurring at time s has the density function

$$f(s) = \frac{r(s)}{m(A)}.$$

Because disjoint windows of time can be considered independently, the two equations above can be used to derive the probability of any set of observed data points—i.e., the likelihood of the model parameters r. If n arrivals $\mathbf{y} = (y_1, \ldots, y_n)$ are observed within an interval $[0, L]$, the relevant likelihood function is (letting $y_0 = 0$):

$$\begin{aligned} \mathrm{P}(\mathbf{y} \mid L, r) &= \mathrm{P}(N[y_n, L] = 0) \cdot \prod_{j=1}^{n} \mathrm{P}(N[y_{j-1}, y_j) = 1) f(y_j) \\ &= e^{-m[y_n, L]} \cdot \prod_{j=1}^{n} e^{-m[y_{j-1}, y_j)} r(y_j) \\ &= e^{-m(A)} \prod_{j=1}^{n} r(y_j). \end{aligned} \tag{12}$$