Analysing partially observed continuous time and discrete state space models (First stage summary)

Ramses Mena & Stephen Walker

September 2019

The aim of this project is to develop models and find algorithms for estimating Markov time series, which are partially observed. At the start we looked at the case of continuous time process with discrete state space. The aim is to then develop the corresponding models and algorithms for continuous state spaces.

The model in discrete time is based on exponential holding times for length of stays in each state and a transition probability and it is these quantities which need to be estimated from the partially observed process. When partially observed, the likelihood is notoriously hard to deal with and the benchmark algorithm for doing this is Bladt and Sorensen (2005). One of the key ideas of this paper is a rejection algorithm which we were looking to improve on. It is well known that rejection algorithms within the context of Bayesian posterior sampling algorithms are a highly weak link in the algorithm.

A Research Associate, Qiaohui Lin, was employed during the Spring 2019 semester. The aim was to find a way to sample a very complicated density function associated with the partially observed Markov process in continuous time and with a discrete state space. The sampling of the density function is a part of an algorithm for estimating the parameters of the model which is represented by a square matrix; known as the *Generator*. The size of the matrix is precisely the number of states.

The process is observed in the following way; at specific points in time, say (τ_j) the process, denoted by $X(t)$, is known to be in state (s_j). It is the missing parts of the process, i.e. what happens in each interval of time $T_j = (t_{j-1}, t_j)$, which needs to be sampled, in order to exploit the full data likelihood.

If the process was fully observed, that is, each change of state was observed and the times at which the changes occured were known, then the estimation of the parameters is quite straightforward. By imputing the missing parts of the process, one is completing the data, and then one can exploit the simplicity of the full likelihood. However, the sampling of the missing parts of the process is non–trivial; in fact it is highly complicated. Made more so if the number of states is large.

It is a novel approach to the sampling of this density which the RA was taking on. The difficulty lies in the problem that the number of changes within each interval of time T_j is unknown and must itself be sampled. If this is done using a Markov chain Monte Carlo method (MCMC), which would be quite normal in such a scenario, then the burden lies in the fact that one would actually need a *reversible jump* MCMC. What is unknown in an interval of time T with start state s_0 and end state S_T is, k, the number of changes of state in going from s_0 to s_T, the states involved, and the time spent in each of these states. The number of states and the times within each state depend on k. So, with a reversible jump MCMC approach one needs to find a suitable transtion density taking $(k, s_1, \ldots, s_k, t_1, \ldots, t_k)$ to say $(k', s'_1, \ldots, s'_{k'}, t'_1, \ldots, t'_{k'})$ satisfying the usual reversible constraints.

This is a very complicated agenda, and even if one has a mathematically correct algorithm, it certainly does not imply it works well. Indeed, this was the problem we faced and by the end of the Spring semeter it was concluded that the idea was not feasible.

An alternative idea is a direct computation of the likelihood function, not previously attempted. This avoids all need for filling in the process in periods of time in which it is unobserved. Initial trials with simulated data suggest there is a strong possibility of this working even with a large number of states.

RA Report
Qiao Lin, May 2019

We introduce a new algorithm to estimate the generator matrix for a continuous time markov chain, a reversible jump procedure in a Gibbs framework with a likelihood regularization. Previous contribution in estimating generator matrix from a bayesian perspective has been made by Bladt and Sorensen (2005). They proposed a rejection sampling to generate complete samples of a not fully observed chain, and using the complete samples, they were able to estimate each element in the generator matrix from a conjugate family. However, as the rejection sampling becomes inefficient when the state space expands and depends heavily on the starting point, we investigate a new method to search a wider parameter space and allow more flexibility in estimating the generator matrix. (Further comparison of the two methods is discussed in the section of related work.) We first give a brief introduction of the continuous time markov chain and the generator matrix, and build our estimation model upon these well-known properties.

A continuous time markov chain (CTMC) is a stochastic process of $\{X(t), t \geq 0\}$ where t is any nonnegative real number, and $X(t)$ is a random variable of the state at time t. A homogeneous CTMC where the transition probability matrix P is denoted as:

$$P(X(s+t) = j | X(s) = i) = P(X(t) = j | X(0) = i) = P_{i,j}(t)$$

A CTMC can be presented by its transition rates, i.e., the generator matrix

G defined as

$$G_{i,j} = \lim_{h \to 0+} \frac{P_{i,j}(h) - P_{i,j}(0)}{h} = \lim_{h \to 0+} \frac{P_{i,j}(h) - I}{h}$$

Following this definition, G satisfies the property of $G_{i,j} = G_{i,i}P_{i,j}$. G also satisfies the property of each row summing to 0, written as $G_{ii} = -\sum_{j \neq i} G_{i,j}$ in a finite state space.

The Kolmogorov forward equation $P'(t) = P(t)G$ describes the relationship between transition matrix P and generator matrix G. Solving the this ODE gives us a widely recoginized equation for CTMC

$$P(t) = e^{Gt} = \sum_{l=0}^{\infty} \frac{G^l t^l}{l!}.$$

To create conjugacy of the generator matrix for the purpose of sampling, we use the uniformization trick to separate the transition intensity λ and the transition probability matrix $\tilde{P}$ and find a conditional conjugacy family for each of them.

For any continuous time markov chain X_t can be written as $Y_{N(t)}$ where $N(t)$ is poison process with intensity λ and Y is a discrete time markove chain with transition probability matrix $\tilde{P}$, where $\tilde{P} = -\lambda^{-1}G + I$. The probability from state i to state j in time t can thus be represented as

$$P_{ij}(t) = P(x(t)|x(0)) = \sum_{l=0}^{\infty} \tilde{P}_{ij}^{l} \frac{e^{-\lambda t}(\lambda t)^l}{l!}$$

It can be easily shown this equation holds with the expansion of exponentials, as in

$$\sum_{l=0}^{\infty} \tilde{P}^l \frac{e^{-\lambda t}(\lambda t)^l}{l!} = \sum_{l=0}^{\infty} (\lambda^{-1}G + I)^l \frac{e^{-\lambda t}(\lambda t)^l}{l!} = e^{-\lambda t} \sum_{l=0}^{\infty} (G + \lambda I)^l \frac{e^{-\lambda t}(\lambda t)^l}{l!}$$

$$= e^{-\lambda t} e^{G + \lambda I} t = \sum_{l=0}^{\infty} \frac{G^l t^l}{l!} = P(t)$$

From the equation above, we can separate λ and $\tilde{P}$ from the generator matrix and sample them individually. The equation also shows the number of jump l can be modeled as Poisson distributed random variable with parameter λt and thus, λ can thus be modeled by a gamma prior and a conjugate gamma posterior.

In order to sample the posterior λ and $\tilde{P}$, we need to sample the number of jumps l and the state jumped to $k_1, \ldots, k_l$ between each observed state s_i and s_j. Note number of jump l is different between each observed state.

Once l is sampled, the number of states inserted $k_1, \ldots, k_l$ depends on l, i.e the dimension of parameter space is changeable determined by l. Reversible jump is one way to solve the problem of non-deterministic parameter space.

In a resversible jump, let $z = (l, k_1, , , k_l) = (l, \mathbf{s_l})$ denote at state z we have l parameters in parameter vector $\mathbf{s_l}$, and $z' = (l', \mathbf{s_l'})$ with l' parameters in $\mathbf{s_l'}$. We want to make the jump from state z to z' and make sure this jump can be reversibly made from z' to z. At state z, to jump to state z', we propose auxiliary variable u from $g(u)$, and u' from $g'(u')$, then the jump and its reverse can be made through a function h

$$h(z, u) = (z', u')$$

$$h^{-1}(z', u') = (z, u).$$

To make h and h^{-1} both differentiable, we must have the dimension match

$$\dim(z) + \dim(u) = \dim(z') + \dim(u').$$

The jump of h from z to z' is accepted with probablity $\alpha(z, z')$ and z' back to z with probability $\alpha(z', z)$. Reversibility means

$$\int_{z,z'} \pi(z)g(u)\alpha(z, z')dxdu = \int_{z,z'} \pi(z')g'(u)\alpha(z', z)dz'du'.$$

Using the change of variable, the above equation holds when

$$\pi(z)g(u)\alpha(z, z') = \pi(z')g'(u')\alpha(z', z)\left|\frac{\partial(z', u')}{\partial(z, u)}\right|.$$

This would always hold if we set

$$\alpha(z, z') = \min\left\{1, \frac{\pi(z')g'(u')}{\pi(z)g(u)}\left|\frac{\partial(z', u')}{\partial(z, u)}\right|\right\} = \min\{1, A(z, z')\},$$

$$\alpha(z', z) = \min\left\{1, \frac{\pi(z)g(u)}{\pi(z')g'(u')}\left|\frac{\partial(z, u)}{\partial(z', u')}\right|\right\} = \min\{1, A(z', z)\}.$$

With observed states $\mathbf{y}$, $A(z, z')$ can be written as

$$A(z, z') = \frac{p(z'|y)r(z')g'(u')}{p(z|y)r(z)g(u)}\left|\frac{\partial(z', u')}{\partial(z, u)}\right|$$

where $p(z'|y)/p(z|y)$ is the likelihood ratio, and $r(z)$ is the probability of proposal to jump to z'.

Between each observed s_i, s_j, We propose the number of jumps l goes up by 1 to $l + 1$ (birth) with probability $\frac{1}{2}$, and goes down to $l - 1$ (death) also with probability $\frac{1}{2}$. Once we make the decision of birth, we randomly choose a state (k_-), between this current state (k_-) and its next one (k_+), we uniformly insert a jump k_{ins}, then the process of k_- to k_+ becomes k_- to k_{ins} to k_+. If we make the decision of death, we randomly choose a state and cancels it.

Because the dimension change is only 1 in this case, we only propose a one-dimensional auxiliary variable u in the birth step and no u' is required coming back to one dimension lower state, i.e., $\dim(z) + \dim(u) = \dim(z')$.

If we propose u uniformly between all states (we have s states in total), and $k_{ins} = u$, we have

$$(x, u) = (l, k_1, \ldots, k_-, k_+, \ldots, k_l, u),$$

and $x' = (l+1, k_1, \ldots, k_-, k_{ins}, k_+, \ldots, k_l)$. As we set $k_{ins} = u$, we have $\left|\partial x'/\partial(x,u)\right| = 1$.

Thus, for $l > 1$, the reversiblity would be satisfied when we use

$$\begin{aligned} A(z, z') &= \frac{p(z'|y)}{p(z|y)} \frac{\Pr(death)P_{alloc})}{\Pr(birth)P_{alloc}} \frac{1}{g(u)} \left|\frac{\partial z'}{\partial(z,u)}\right| \\ &= \left(\frac{\lambda t}{l+1} \frac{P_{k_-,k_{ins}} P_{k_{ins},k_+}}{P_{k_-,k_+}}\right) \frac{\frac{1}{2}\frac{1}{l+1}}{\frac{1}{2}\frac{1}{l+1}} * \frac{1}{\frac{1}{s}} * 1. \end{aligned}$$

Here $p(z|y)$ is determined by two parts, the number of jumps l and the probability of making the jump to observed status using the transition matrix. Since $l \sim \mathrm{Po}(\lambda)$,

$$p(l|\lambda) = \frac{\lambda^l e^{-\lambda}}{l!},$$

$$p(l+1|\lambda) = \frac{\lambda^{l+1} e^{-\lambda}}{(l+1)!}$$

and

$$\frac{p(l+1|\lambda)}{p(l|\lambda)} = \frac{\lambda t}{l+1}.$$

Thus

$$\frac{p(z'|y)}{p(z|y)} = \frac{\lambda t}{l+1} * \frac{P_{k_-,k_{ins}} P_{k_{ins},k_+}}{P_{k_-,k_+}}.$$

Since insertion position is random between all jumps, $P_{alloc} = \frac{1}{l+1}$ for both birth and death, as jumping up has $l+1$ space for insertion between s_i and $k_1, , k_l$ and s_j and jumping down has $l+1$ states $k_1, , k_{l+1}$ to choose to cancel. As u is uniformly proposed between all possible states, $g(u) = \frac{1}{s}$, s is the number of legal states to propose, all the states that is not same as k_- and k_+.

Similarly, for $l > 1$,

$$\begin{aligned} A(z', z) &= \frac{p(z|y)}{p(z'|y)} \frac{\Pr(birth)P_{alloc}}{\Pr(death)P_{alloc}} \frac{g(u)}{1} \left|\frac{\partial(z,u)}{\partial z'}\right| \\ &= \left(\frac{l}{\lambda} \frac{P_{k_{can-1},k_{can+1}}}{P_{k_{can-1},k_{can}} * P_{k_{can},k_{can+1}}}\right) \frac{\frac{1}{2}\frac{1}{l}}{\frac{1}{2}\frac{1}{l}} * \frac{\frac{1}{s}}{1} * 1. \end{aligned}$$

However, $l = 1$ and $l = 0$ are the two special cases for the reversible jump as at $l = 0$ there is no death proposal, it can only jump up. Thus, at $l = 0$, there is only a birth propsal with

$$A(z, z') = \left(\frac{\lambda t}{l+1} \frac{P_{k_-,k_{ins}} P_{k_{ins},k_+}}{P_{k_-,k_+}} \right) \frac{\frac{1}{2}\frac{1}{l+1}}{\frac{1}{l+1}} * \frac{1}{\frac{1}{s}} * 1, \quad l = 0.$$

The $\frac{1}{2}$ is no longer in the denominator as the birth proposal now has probability 1.

When $l = 1$, jumping up is the same as $l > 1$ cases, but jumping down has to be modified for the reversibility,

$$A(z'z) = \left(\frac{l}{\lambda} \frac{P_{k_{can-1},k_{can+1}}}{P_{k_{can-1},k_{can}} * P_{k_{can},k_{can+1}}} \right) \frac{\frac{1}{l}}{\frac{1}{2}\frac{1}{l}} * \frac{\frac{1}{s}}{1} * 1, \quad l = 1.$$

The death proposal for $l = 1$ is $l = 0$, and $l = 0$ has probability 1 to propose the reverse instead of $\frac{1}{2}$. Hence there is no $\frac{1}{2}$ in the nominator for $l = 1$.

Once the $A(z, z')$ and $A(z', z)$ is calculated at each proposal,

$$\alpha(z, z') = \min\{1, A(z, z')\} \quad \text{and} \quad \alpha(z', z) = \min\{1, A(z', z)\}$$

is determined for acceptance of the proposal and this concludes the first step, the reversible jump, in our sampling.

One problem of this strategy is that our uniform insertion of states is not informative of the transition probability matrix $\tilde{P}$. To efficiently recover $\tilde{P}$, we need the states to be more representative of the transition probability. Thus, after the reversible jump step, we do a shuffle of states using Metropolis Hastings.

With the current l and $k_1, \ldots, k_l$ from the reversible jump result, we now propose k''_1, k''_2, k''_l, with only one criterion that k''_i has to be a legal state not the same as its neighbours. We make this k''_1, k''_2, k''_l a metropolis proposal and we decide whether to accept this proposal by

$$\min\left\{1, \frac{P_{s_i,k''_1} P_{k''_2,k''_3} \ldots P_{k''_l,s_j}}{P_{s_i,k_1} P_{k_2,k_3} \ldots P_{k_l,s_j}}\right\},$$

where

$$\frac{P_{s_i,k''_1} P_{k''_2,k''_3} \ldots P_{k''_l,s_j}}{P_{s_i,k_1} P_{k_2,k_3} \ldots P_{k_l,s_j}}$$

is the likelihood $f(k'')/f(k)$ in the Metropolis acceptance ratio

$$\frac{f(k'')}{f(k)} \frac{q(k|k'')}{q(k''|k)}.$$

The proposal $q(k|k'')/q(k''|k)$ is cancelled here as the k''_1, k''_2, k''_l and k_1, k_2, k_l does not depend on each other. This Metropolis decided the sequence we insert into observed states in each iteration.

Now with the number of states $l_1, , , l_N$ and all the k sequences inserted between all observed data, our posterior λ and $\tilde{P}$ follows the posterior in a Gibbs framework of

$$\lambda | \cdot \sim \mathrm{Ga}(a + l_1 + \cdots + l_N, b + t_1 + \cdots + t_N)$$

where a, b is the prior hyperparameter $\lambda \sim \mathrm{Ga}(a, b)$. Here $t_1, \ldots, t_N$ is the time elapse between observed states s_1, s_N.

$$\tilde{P}[i,] | \cdot \sim Dir(\alpha + Ni1, \ldots, \alpha + Nis)$$

$\tilde{P}[i,]$ is the ith row of $\tilde{P}$ without the diagnol element $\tilde{P}[i, i]$, $\tilde{P}[i, i] = 0$. α is the hyperparameter of prior Dirichlet distribution of $\tilde{P}[i,]$. N_{ij} is the number of jump from state i to state j in the chain we complete in total.

Instead of updating the λ and $\tilde{P}$ using the posterior sample, we treat the posterior sample at iteration $iter + 1$, λ' and $\tilde{P}'$, as a Metroplolis-Hastings proposal for λ and $\tilde{P}$ at iteration $iter$. The acceptance ratio is

$$\min \left\{ 1, \frac{L(\lambda', \tilde{P}')}{L(\lambda, \tilde{P})} \right\},$$

where $L(\lambda, \tilde{P})$ is the likelihood of all observed data $s_1, \ldots, s_N$ depending on $(\lambda, \tilde{P})$,

$$L(\lambda, \tilde{P}) = \prod_{i=1}^{N} P_{s_i, s_{i+1}}(t) = \prod_{i=1}^{N} e^{Gt_i}[s_i, s_{i+1}] = e^{(-\lambda \tilde{P} + \lambda I)t_i}[s_i, s_{i+1}].$$

The posterior sample at iteration $iter + 1$ will not be accepted if it results in a much smaller likelihood over observed data points. In this way, we regularize the chain to move in the direction where the observed likelihood can only be improved over iterations.

One thing we have to notice is in a general Metropolis-Hastings framework, the acceptance ratio is

$$\frac{f(\lambda', \tilde{P}')}{f(\lambda, \tilde{P})} \frac{q(\lambda, \tilde{P} | \lambda', \tilde{P}')}{q(\lambda', \tilde{P}' | \lambda, \tilde{P})}$$

but we are only keeping the likelihood but omitting the proposal q part in our calculation.

We simulate a process with intensity $\lambda = 4$ and a random matrix transition matrix $\tilde{P}$ with 7 states. We observe random 200 data points in the chain, the intervals have expected λ data points omitted in between. The prior we use for λ is Gamma(4,1), and prior for each row of $\tilde{P}$ is Dirichlet(1,1,,1).

Using the MCMC described in the previous section, we are able to recover the $\lambda = 4.8$ and $\tilde{P}$ with maximum 0.1 difference elementwise to the true $\tilde{P}$.

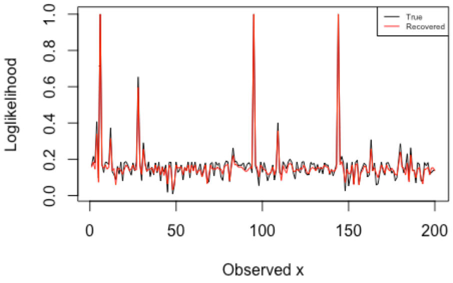

Figure 1: True and recovered Loglikelihood at observed data points in the simulated process

We compare the true likelihood $P_{s_1,s_2}(t_1)$,$P_{s_2,s_3}(t_2)$,...,$P_{s_{N-1},s_N}(t_{N-1})$ with the one we recover using estimated λ and $\tilde{P}$.

The true loglikelihood for the whole observed chain

$$L = \prod_{i}^{N} P_{s_i,s_{i+1}}(t)$$

is -385, while our recovered loglikelihood is higher by 0.9.

Bladt and Sorensen (2005) proposed a rejection sampling within a gibbs framework to estimate the generator matrix for the continuous time markov chain. If all jumps observed, the likelihood can be written as

$$L = \prod_{i}^{m} \prod_{j \neq i} G_{ij}^{N_{ij}} e^{-G_{ij} T_i}$$

m is the number of all states,T_i is the time stayed at status i, and N_{ij} is the number of jumps from i to j in the complete chain. Based on the likelihood, one conjugate family is to propose a Gamma prior $\mathrm{Ga}(\alpha_{ij}, \beta_i)$ for each of the element G_{ij} in the G matrix. So the posterior for G_{ij} is

$$p^*(G_{ij}) \propto G_{ij}^{N_{ij}+\alpha_{ij}-1} e^{-G_{ij}(T_i+\beta_i)}$$

When not all state are observed in the chain, a rejection sampling is proposed by Bladt and Sorensen (2005) and further explained by Inamura (2006). We first obtain one sample of the complete chain based on the observations $s_1, \ldots, s_n$ at time $t_1, \ldots, t_n$. At the $(n-1)$th observed state with status i at time stamp t_{n-1}, we propose a holding time Δt_1 (exponentially distributed with parameter G_{ii}), if $t_{n-1} + \Delta t_1 < t_n$, we propose an inserted unobserved state $S_1 = j$ based on the probability $-G_{ij}/-G_{ii}$. Now we start from $S_1 = j$, and propose $\Delta t_2 \exp(G_{jj})$, if $t_{n-1} + \Delta t_1 + \Delta t_2 < t_n$ we insert another $S_2 = k$ at $t_{n-1} + \Delta t_1 + \Delta t_2$ based on probability $-G_{jk}/G_{jj}$. Repeat this procedure until $t_{n-1} + \Delta t_1 + \cdots + \Delta t_l \geq t_n$, if the final state S_l when the time is up is the observed state s_n, we accepted this chain between s_{n-1} and s_n and move on to sample the chain s_n to s_{n+1}, otherwise we reject and start from s_{n-1} again. Once the chain is completed and accepted, we calculate the N_{ij} and T_i and update the G matrix by drawing each element from its posterior.

Though rejection sampling is one way of getting samples of the complete chain and repeated estimate G by each element based on the chain, it has one drawback. If the G starts from a bad prior, it may never get to the right point as the chain it proposed will always get rejected and the Gibbs sampler for estimating G will get stuck. The fact the rejection sampling does not move around and search the whole parameter space for G makes the estimate depend heavily on the starting point and the prior.

Another problem of the rejection sampling is when the observed states are far away from each other in time or the state space is large, the acceptance ratio of the chain gets very small, and the algorithm will thus be inefficient.

Technical Report
Mena & Walker, September 2019

We consider a partially observed Markov process which has a finite state space, denoted by $\mathbb{S}$, and in continuous time; i.e. a continuous time discrete state system. Also known as a continuous time Markov chain (CTMC). Such a system has many application areas, including physiscs, ecology, and neuroscience; see [5], [2] and [9], respectively. One special case of such a process would be a *Renewal Process*, see [6], for example.

If such a process $\{X(t); t \leq \tau\}$, is fully observed over time; so that at any point in time the value of the state is known, inference for the unknown parameters is straightforward. The unknown is the *Generator Matrix*, G, which is a m square matrix, also known as the intensity matrix. This matrix of zero row sums, comprises non–negative off–diagonal jump intensities and non–positive diagonal elements, which we refer to as the exponential parameters, since the holding times in each state are exponential random variables.

The generator matrix can therefore be understood as comprising a diagonal matrix D with positive entries, comprising the exponential parameters for the holding times within each state. So if there are m states, D is of dimension $m \times m$. The other unknown is a $m \times m$ stochastic matrix P comprising the transition probabilities; $P = (p_{jk})$, with $p_{jj} = 0$. So once it is that a state is to change, P determines the probabilities for the move to state. The point is that if all transition times are known, and the accompanying changes of state are also known, the likelihood function is

$$\mathbb{L}(P, D|\text{data}) = \prod_{j,k} p_{jk}^{n_{jk}} \quad \times \quad \prod_{j} d_j^{n_j}\, e^{-T_j\, d_j},$$

where n_{jk} is the number of transitions from state j to state k, n_j is the number of visits to state j, and T_j is the total time spent in state j. From here, a maximum likelihood estimator is easily available, for example, as would be a Bayesian posterior.

However, in most illustrations involving such processes, the process X is observed only at specific time points, say at times $(t_1, \ldots, t_n)$, and we write the observations as $(X(t_1), \ldots, X(t_n))$, to be shortened to $(X_1, \ldots, X_n)$. In this case, in order to construct the likelihood function, we would need $Q_t(j, k)$; the probability of moving from state j to state k within an interval of time t. The likelihood then becomes

$$\mathbb{L}(D, P|\text{data}) = \prod_{i=1}^{n} Q_{\Delta_i}(x_i, x_{i+1}), \tag{0.1}$$

where $\Delta_i = t_i - t_{i-1}$, with $t_0 = 0$, and, as we have mentioned, x_i is the state of the process at time t_i. We will see how Q depends on D and P in section 2 though it is well known textbook theory.

The problem of the partially observed process has received considerable attention within the literature, and up to date reviews can be found in [1], [4], and [7], for example. The last of these references also provides an R package implementation of various proposals. By now, it is well understood that a maximum likelihood estimation (MLE) approach has several drawbacks; i.e. the MLE might not exist. The existence issues worsen as gaps between the partially observed records increase; see [1]. Given all this, a Bayesian approach to the problem is often preferred.

Perhaps the most accepted technique is to *impute* the missing observations between the observed data. Indeed, this is the technique proposed by [1], for both of their approaches, uisng the expectation maximization (EM) algorithm and the Gibbs sampler method. However, these approaches rely on an ability to sample the missing observations – which is not an easy prospect, particularly when the number of states becomes large.

The approach we describe in this paper has we believe been overlooked in preference for the imputation procedures. However, as we shall see, it is possible to use a Bayesian approach on the likelihood function corresponding to the partially observed process. Given the likelihood is not easy to compute, but we can do it, so a direct sampling from the posterior is not possible. Therefore a Metropolis algorithm is used.

The process can be characterized by the *Generator Matrix* G which is of dimension m and has zero row sums. The background here is well trodden and to be found, for example, in section 6.9 of [3]. The generator is given by

$$G = -D + D\,P.$$

The interpretation is that if the process is in current state j, the remaining time in this state is an exponential random variable with parameter d_j and after this time the process moves to a different state k with probability p_{jk}. That X is a Markov process follows immediately from the *lack of memory* property of the exponential distribution.

The probability of moving from state j to state k, with k possibly equal to j, in an interval of time t, is given by

$$Q_t(j,k) = [\exp(t\,G)]_{(j,k)}. \tag{0.2}$$

Here the matrix $\exp(tG)$ is given by

$$\exp(tG) = \sum_{l=0}^{\infty} \frac{t^l}{l!}\,G^l \tag{0.3}$$

with $G^0 = I$, the $s \times s$ identity matrix. So $Q_t(j,k)$ is the (j,k)th element of $\exp(tG)$. For more on the theory presented here, see, for example, [3].

It is a matter of some light calculus and algebra to understand the representations involved and to demonstrate the veracity of (0.2). It is tempting to write

$$Q_t(j,k) = \exp(-tD)\,\exp(tDP)$$

and to expand $\exp(tDP)$ using the exponential series, since all the terms will now be positive. Hence, if possible, one could write

$$Q_t(j,k,l) = \exp(-tD)\frac{(tDP)^l}{l!}$$

to obtain a convenient full likelihood with missing information l. However, this “obvious” strategy fails in general since for matrices A and B, $\exp(A+B) \neq \exp(A)\,\exp(B)$, with equality only if $AB = BA$. Hence, this would require $D^2\,P = DPD$, which is clearly not true in general.

We will need to sum (0.3), to a finite number of terms, in order to approximate the $\exp(tG)$. This is standard; indeed the computer will always return an exponential value by truncating such a sum. Here we discuss how many terms are required for a given level of accuracy.

To this end, let $\|\cdot\|$ be a matrix norm which is both sub–additive and sub–multiplicative; i.e. it satisfies the triangular inequality and $\|A\,B\| \leq \|A\|\,\|B\|$. For example,

$$\|A\| = \sum_{i,j} |a_{ij}|.$$

If we truncate the sum of exponential terms at L, the error is given by

$$\mathbb{E}(L) = \left\| \sum_{l=L}^{\infty} \frac{t^l\, G^l}{l!} \right\|$$

and we want to find L so that this is upper bounded by ϵ, which would be our choice.

Now

$$\mathbb{E}(L) \leq \sum_{l=L}^{\infty} \frac{(t\|G\|)^l}{l!} = \frac{(t\|G\|)^L}{L!} \sum_{l=0}^{\infty} \frac{(t\|G\|)^l}{(l+L)!/L!}.$$

Since $(l+L)!/L! \geq l!$ we see that

$$\mathbb{E}(L) \leq \frac{(t\|G\|)^L}{L!}\, e^{t\|G\|}.$$

Hence, we find L such that this bound is less than ϵ. Note that

$$\|G\| \leq \|D\|\,(1 + \|P\|)$$

and so it is easy to find the appropriate L.

Lemma 0.1. *If*

$$L = \frac{\log(1/\epsilon) + t\|G\|(1+b)}{\log b}$$

for any $b > 1$, then

$$\frac{(t\|G\|)^L}{\Gamma(1+L)}\, e^{t\|G\|} \leq \epsilon.$$

Proof. Let $a = t\|G\|$; then

$$\Gamma(1+L)/a^L = \int_0^{\infty} (s/a)^L\, e^{-s}\, ds = \int_0^{\infty} y^L\, e^{-ay}\, a\, dy,$$

and so

$$\Gamma(1+L)/a^L > a \int_b^{\infty} y^L\, e^{-ay}\, dy > b^L\, e^{-ab}$$

for any $b > 0$. Plugging in the given value of L yields

$$\Gamma(1+L)/a^L > \exp\{\log(1/\epsilon) + a(1+b) - ab\} = e^a/\epsilon,$$

as required. □

As an illustration, we took

$$P = \begin{pmatrix} 0 & 0.3 & 0.7 \\ 0.5 & 0 & 0.5 \\ 0.8 & 0.2 & 0 \end{pmatrix} \quad \text{and} \quad D = \operatorname{diag}(1, 2, \tfrac{1}{2})$$

and approximated $\exp(G)$ with 150 terms in the exponential expansion, yielding

$$\exp\, G = \begin{pmatrix} 0.487 & 0.092 & 0.421 \\ 0.337 & 0.188 & 0.475 \\ 0.228 & 0.056 & 0.716 \end{pmatrix}.$$

This is a stochastic matrix and is correct to 3 decimal places.

The likelihood function for the given observations $(X(0)\, X(t_1), \dots, X(t_n))$ is

$$\mathcal{L}(G) = \prod_{i=1}^{n} Q_{\Delta_i}(x_{i-1}, x_i),$$

where, for short, we write $X(t_i) = x_i$, and $\Delta_i = t_i - t_{i-1}$, with $t_0 = 0$. Inference is complicated by the fact that Q is defined via an infinite sum. Maximizing the likelihood function is going to be exceptionally difficult due to the numerous constraints involved. Indeed, it appears to be a stratgey that has not been tried.

This has motivated searches for suitable latent variables. If all state changes are observed, with the time spent in each state also observed, the likelihood takes on a simple form;

$$\mathbb{L}(G) = \prod_{j \neq k} p_{jk}^{n_{jk}} \prod_{j} d_j^{n_j} \, \exp(-d_j\, T_j), \tag{0.4}$$

where T_j is the total time spent in state j, n_j is the number of times state j is visited and n_{jk} is the number of times a change from state j to state k occurs.

In order to utilize this full data likelihood via an EM algorithm, it would be necessary to sample the missing states and changes and times spent in states between known states. That is, suppose the start state is x and the end state is y and the time between these two known states is t. That is, at time 0 it is known the process is in state x and after a time t it is known the process is in state y. The probability of this is $Q_t(x, y) = [\exp(tG)]_{(x,y)}$.

To start to fill in the missing data, we denote the number of changes within this time region by k; so there will be $k + 1$ states

$$(s_0 = x, s_1, \dots, s_{k-1}, s_k = y)$$

with times in states $(t_1, \dots, t_{k+1})$; i.e. time t_j is spent in state s_j, and $\sum_{l=1}^{k+1} t_l = t$. By means of illustration, suppose we want the probability density for $k = 2$, with inbetween state s_1 and times (t_1, t_2), with $t_3 = t - t_1 - t_2$. See Fig 2. So t_1 is the time spent in state s_0 and t_2 is the time spent in s_1 and $t_3 = t - t_1 - t_2$ is the time spent in state s_2. Then

$$p(2, s_1, t_1, t_2) = d_{s_0} e^{-t_1 d_{s_0}}\, p_{s_0\, s_1}\, d_{s_1} e^{-t_2 d_{s_1}}\, p_{s_1\, s_2}\, e^{-d_{s_2}(t - t_1 - t_2)}.$$

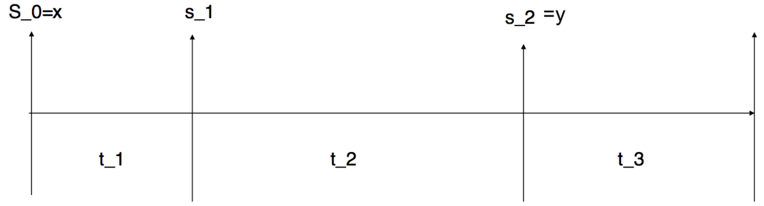

Figure 2: Observed states x and y separated by a time of length t

In general, the density to sample is given by

$$\pi(k, t_1, \ldots, t_k, s_1, \ldots, s_{k-1}|G, x, y, t) = e^{-t_{k+1} d_{s_k}} \prod_{l=1}^{k} p_{s_{l-1}\,s_l}\, d_{s_{l-1}}\, \exp(-t_l\, d_{s_{l-1}}). \tag{0.5}$$

This is not an density easy to sample. A laboured way is to sample the process forward from the start; i.e. to sample t_1 then s_1, t_2, s_2 and so on, and to accept (k, t, s) if after time t the process is at state y. This is equivalent to sampling with the contraint via a rejection algorithm. Of course, with a few states this might work adequately but will run into efficiency issues when the number of states becomes large. This idea of sampling the missing data and the full likelihood as an iterative procedure is precisely the EM algorithm of [1], who also use the idea for the Gibbs sampler.

As has just been mentioned in the previous section, a Bayesian approach using latent variables can be based on sampling π, given in (0.5), using a Markov chain such as a Gibbs sampler or a Metropolis–Hastings sampler. Hence, within a Gibbs sampling framework, we would sample iteratively

$$\pi(k, t_1, \ldots, t_k, s_1, \ldots, s_{k-1}|G, x_i, x_{i+1}, \Delta_i)$$

for $i = 1, \ldots, n-1$, and then sample

$$\pi(G|\text{full data}),$$

which is based on (0.4), suitably multiplied by the prior.

However, latent variables are and should only be introduced when the likelihood function is not directly computable. We have shown in section 2 that it is computable; and so a Bayesian analysis can proceed, via a Metropolis algorithm, for example. With such a Bayesian framework, the setting of prior distributions is straightforward; independent gamma priors can be assigned to each d_i, with shape and scale parameters both set to $\frac{1}{2}$, for example, and each row of P can be assigned a Dirichlet prior with parameters chosen so the prior is uniform on the simplex.

With the P and D as given in the first section, we sampled a process with 5000 changes of state, and from this we subsampled 1000 observations with an equal time difference between the observations as 2; i.e. $\Delta_i = 2$ for all i.

We use a Metropolis–Hastings algorithm for sampling from the posterior distribution. The proposals for the parameters are $q(d_j'|d_j)$, for each j, to be a log–normal distribution with mean $\log d_j$ and standard deviation 0.05, and $q(p_j'|p_j)$, for each j, where $p_j = (p_{jk})$, is a Dirichlet distribution with parameters (cp_{jk}) and $c = 50$. Each time a proposal is made, we recompute the likelihood function in order to determine whether the proposal is accepted. For this we summed the exponential series to 150 terms. We illustrate with the update for d_1; we sample d_1' from $q(d_1'|d_1)$ and accept this move with probability

$$\alpha = \min\left\{1, \frac{\mathcal{L}(G')\,\pi(D')}{\mathcal{L}(G)\,\pi(D)}\right\},$$

where $G' = -D' + D'P$ and $D' = (d_1', d_2, d_3)$, while $G = -D + DP$ and $D = (d_1, d_2, d_3)$. A similar and obvious procedure follows for the other parameters making up D and P.

The results are as follows: with a running of the Metropolis–Hastings chain for 100,000 iterations, we estimate

$$\widehat{P} = \begin{pmatrix} 0 & 0.22 & 0.78 \\ 0.43 & 0 & 0.57 \\ 0.73 & 0.27 & 0 \end{pmatrix} \quad \text{and} \quad \widehat{D} = \mathrm{diag}(1.08, 2.48, 0.67).$$

A trace plot of the 100,000 samples of (d_1) is presented in Fig. 3.

First, here, we construct a data set with 5 states. The exponential parameters for the D matrix are taken to be $(0.4, 1.8, 1.2, 1.6, 2.0)$, and the P matrix is obtained by each row having probabilities obtained by a uniform Dirichlet distribution. Then with these true values, a process is generated from which we extracted as data 3000 points at equally spaced intervals of time, of length 2.

A Metropolis algorithm is used to perform Bayesian analysis. The prior for the rows of P are independent uniform Dirichlet distributions and the prior for the exponential parameters are independent standard exponential. We use $2m$ Metropolis steps within each iteration; one for each of the m exponential parameters and one for each of the m rows of P. The proposal for the former is a log–normal centered on the current value with a variance of 0.05, and the proposal for the latter, i.e. a new row, is a Dirichlet distribution with parameter values set at 10 multiplied by the current row values. The ensuing chain is run for 50,000 iterations.

The subsequent posterior density estimates obtained from the sample output of the Metropolis chain is presented in Fig. 4. The means and variances are given, respectively, by

$$(0.45, 0.001) \quad (0.82, 0.007) \quad (1.10, 0.019) \quad (1.99, 0.108) \quad (2.18, 0.216).$$

Apart from the d_4, in this instance, the estimates are good. The estimates of P, which are not focused on here, are also good, for example p_{12} has a true value of 0.12 and is estimated at 0.10.

Next we considered a process with $m = 10$ states with the probability matrix generated in the same way as before and the exponential parameters as $d_j = j/4$ for $j = 1, \ldots, 10$. We took the prior distributions as before, though in this case we took the proposals for the exponential parameters to have a standard deviation of 0.1. A plot of the estimates of the parameters against the estimated values, obtained as the sample means from the Metropolis algorithm output, is provided in Fig. 5. The chain was run this time for just 5000 iterations.

Other figures, Fig. 6 and Fig. 7 show in more detail output of the Metropolis algorithm, including the posterior density estimates of the exponential parameters and some trace plots, respectively.

References

[1] M. Bladt and M. Sorensen, *Statistical inference for discretely observed Markov jump processes*, Journal of the Royal Statistical Society, Series B (2005) 67, 395–410.

[2] K. Fukaya and J. A. Royle, *Markov models for community dynamics allowing for observation error*, Ecology (2013) 94, 2670–2677.

[3] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes* (1982), Oxford University Press.

[4] R. B. Israel, J. S. Rosenthal and J. S. Wei, *Finding generators for Markov chains via empirical transition matrices, with applications to credit ratings*, Mathematical Finance (2001) 11, 245–265.

[5] N. G. Van Kampen, *Stochastic Processes in Physics and Chemistry* (2007), North–Holland.

[6] R. Pyke, *Markov renewal processes: definitions and preliminary properties*, The Annals of Mathematical Statistics (1961) 32, 1231–1242.

[7] M. Pfeuffer, *ctmcd: An R Package for Estimating the Parameters of a Continuous-Time Markov Chain from Discrete-Time Data*, The R Journal (2017) 19, 127–141.

[8] D. B. Rubin, *Bayesianly justifiable and relevant frequency calculations for the applied statistician*, Annals of Statistics (1984) 12, 1151–1172.

[9] M. Sauer and W. Stannat, *Reliability of signal transmission in stochastic nerve axon equations*, Journal of Computational Neuroscience (2016) 40, 103-111.

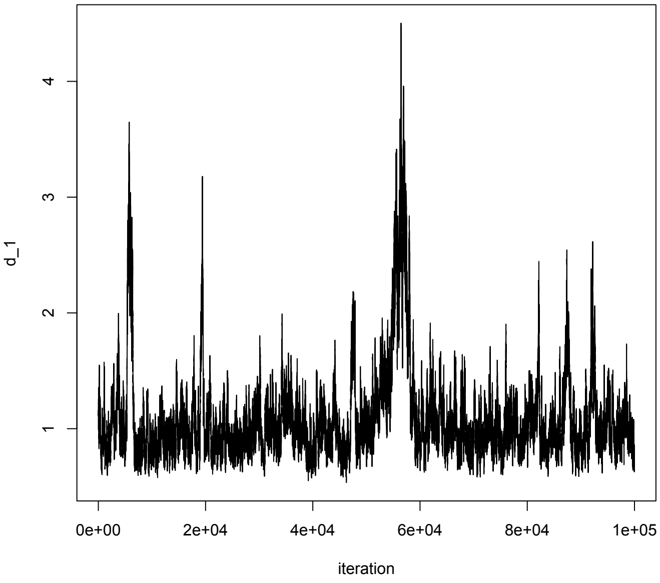

Figure 3: Trace plot of the 100,000 samples of d_1

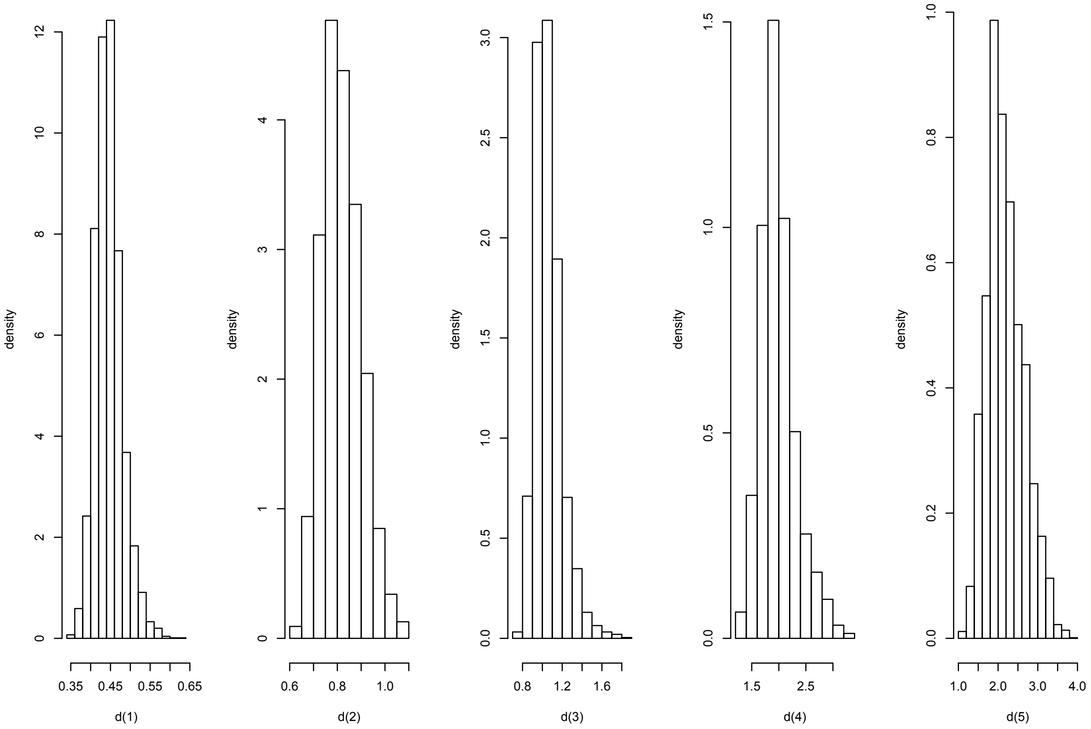

Figure 4: Histogram estimates of posterior densities for the exponential parameters d_1 to d_5

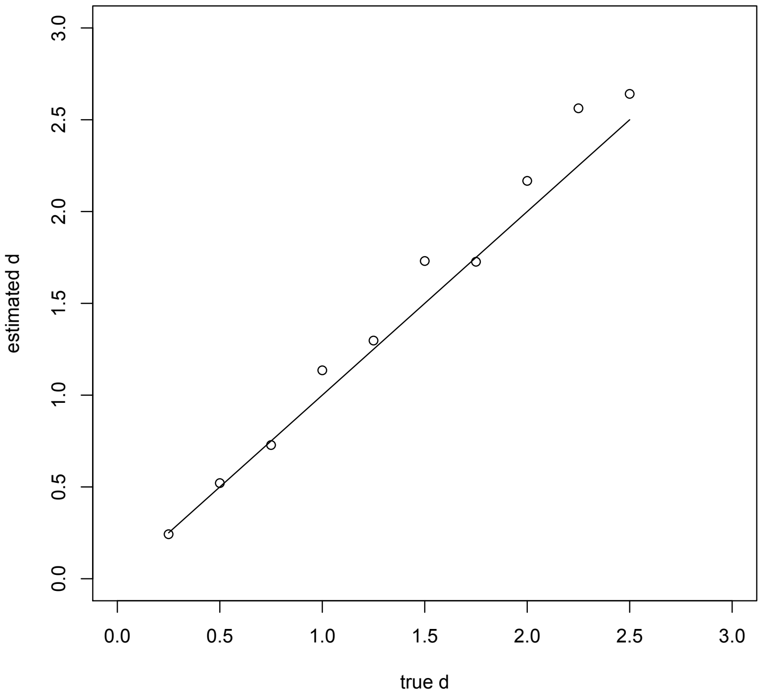

Figure 5: True exponential parameters plotted against the estimated values

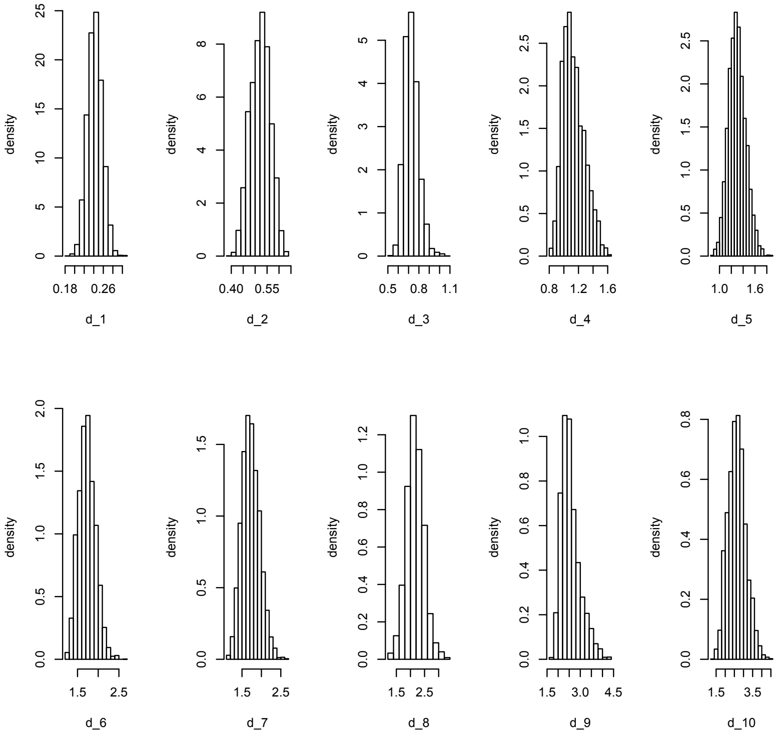

Figure 6: Histogram density estimates of the exponential parameters from Metropolis output

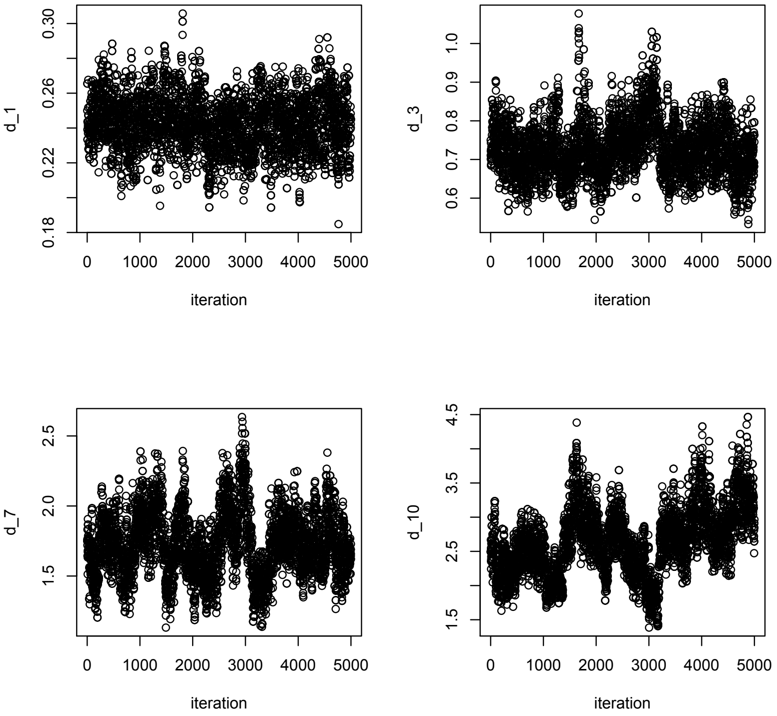

Figure 7: Trace plots of some of the exponential parameters from Metropolis output