Estimation of birth death processes

Steven Nguyen

Supervisors: Dr Sophie Hautphenne, Dr Brendan Patch

THE UNIVERSITY OF MELBOURNE

Problem

A birth death process (BDP) is a continuous time Markov chain which models the number of particles in a system. This number can either transition up or down one step at random according to a sequence of birth-rates λ_k and death-rates μ_k, dependent on the current number of particles k.

- My model of interest is on *logistic growth*, with μ_k and λ_k given by:

$$\lambda_k = k^2 \lambda e^{-\alpha k} \quad (1)$$
$$\mu_k = k\mu, \quad (2)$$

- These models will have the number of particles fluctuate around some *carrying capacity* when $\lambda_k \approx \mu_k$ as shown in Fig 1.
- My aim is to estimate the parameters $\boldsymbol{\theta} = (\lambda, \mu, \alpha)$ based on *discrete* observations of the population.

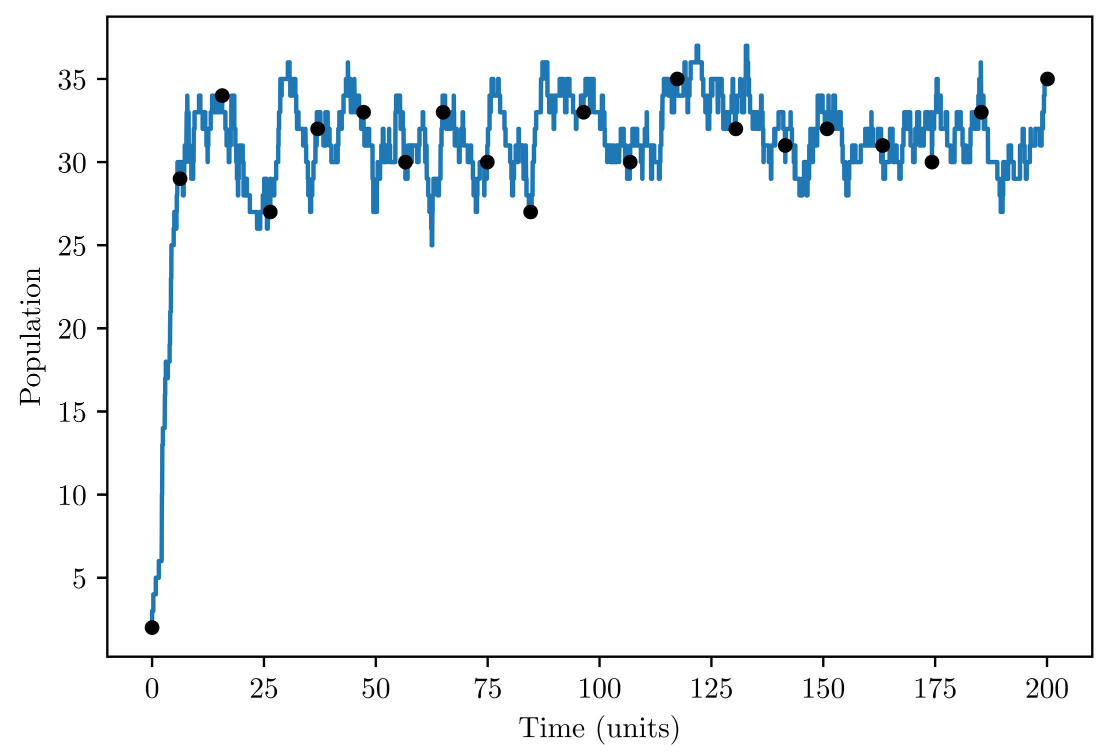

Fig. 1: A sample trajectory of the BDP with λ_k and μ_k given in (1) and (2), where $\lambda = 1$, $\mu = 0.05$, $\alpha = 0.2$. Black dots represent sampling points, which are done at regular time intervals. Here $K \approx 32$

Method

- Let k_i be the ith observation of the population in one given sample, and τ_i be the time between observation $(i-1)$ and i. Then the log-likelihood function is:

$$\ell(\boldsymbol{\theta}; \boldsymbol{\tau}, \mathbf{k}) = \sum_{i=1}^{N} \log p_{k_{i-1},k_i}(\tau_i), \quad (3)$$

  (see [1]), where each $p_{ij}(t)$ is the probability that the process transitions from population i to j in time interval t.
- To find estimates, we aim to maximise this function with respect to $\boldsymbol{\theta}$.
- However $p_{ij}(t)$ is difficult to find explicitly for general μ_k and λ_k.
- The method of interest to find $p_{ij}(t)$, by Crawford [2], first finds Laplace transform $f_{ij}(s)$ of the probability $p_{ij}(t)$ as a continued fraction given by:

$$\mathcal{L}\{p_{ij}\}(s) = f_{ij}(s) = \begin{cases} \left(\displaystyle\sum_{k=j+1}^{i} \mu_k\right) \dfrac{B_j(s)}{B_{i+1}(s)+} \dfrac{B_i(s)a_{i+2}}{b_{i+2}+} \dfrac{a_{i+3}}{b_{i+3}+}, \cdots & j \leq i, \\ \left(\displaystyle\sum_{k=i}^{j-1} \lambda_k\right) \dfrac{B_i(s)}{B_{j+1}(s)+} \dfrac{B_j(s)a_{j+2}}{b_{j+2}+} \dfrac{a_{j+3}}{b_{j+3}+}, \cdots & i \leq j, \end{cases} \quad (4)$$

where $a_k = -\lambda_{k-2}\mu_{k-1}$, $a_1 = 1$,
$b_k = s + \lambda_{k-1}, +\mu_{k-1}$, $b_1 = s + \lambda_0$,
$B_k = b_k B_{k-1} + a_k B_{k-2}$, $B_0 = 1, B_1 = b_1$,

and then numerically inverts to find p_{ij}.

Results

- Python was used to evaluate the probabilities and likelihood numerically, after which optimisation was done using the `optimparallel` package [4]. The initial value of $\boldsymbol{\theta}$ chosen for the optimisation was done by adding noise to the true value. For each trajectory we used the same initial value of $\boldsymbol{\theta}$.
- In the interest of program runtime, $N = 200$ observations were taken for each estimation routine, with sampling being done on simulations of trajectories. The initial populations were chosen randomly for each simulated trajectory between 1 and 20.
- From 100 independent trajectories, kernel density estimates of the MLEs are plotted below, with the true values of the parameters $\lambda = 0.3$, $\mu = 0.05$, $\alpha = 0.5$ indicated by the dashed blue lines.

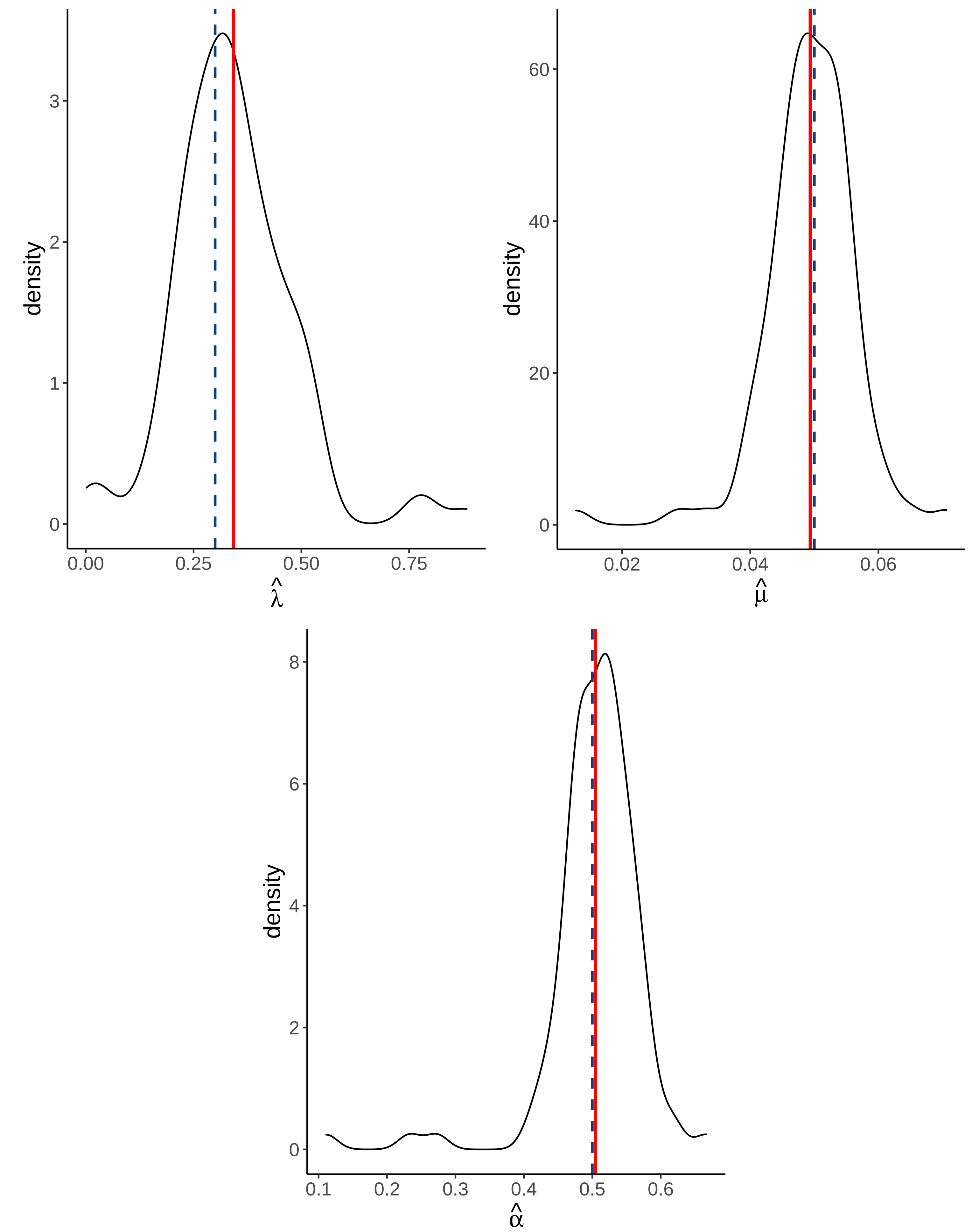

Fig. 2: Distribution of estimates for λ (top left), μ (top right) and α (bottom) for $M = 100$ samples, for true population parameters $(\lambda, \mu, \alpha) = (0.3, 0.05, 0.5)$. Red lines indicate the means of the estimates.

- The plots suggest that the MLE used is consistent.
- However, some estimates deviated substantially from the true value. This may be due to a small sample size, or poor initial guess.

An important point of discussion is the efficiency of calculation of the probabilities in (3). We would like to draw a comparison between taking a matrix exponential of the transition rate matrix Q against the Laplace method used here.

- The Laplace transform method is slightly faster when computing $p_{ij}(t)$ for observations with low populations (i.e low i, j).
- A large advantage of the matrix method is that once the matrix exponential e^{Qt} truncated at some N is computed for a given time t, we have approximations to all of the transition probabilities $p_{ij}(t)$ as $[e^{Qt}]_{ij}$ for $i, j < N$ without needing to re-evaluate e^{Qt}.

Laplace	Truncation Level	MatExp	MatExp2
262.43	N = 100	2.86	27.93
	N = 200	3.54	127.65
	N = 500	11.07	1546.61

Tab. 1: Table of CPU times (s) for finding MLEs, using different methods of finding $p_{ij}(t)$. Here MatExp requires only one computation of the matrix exponential, while MatExp2 reevaluates the matrix exponential for each probability.

- The matrix method proves to be much faster in the case when we take advantage of reading a single matrix over and over again, and still outperforms in the cases where we need to recompute, up to a certain truncation level.
- Due to the availability of efficient ways to calculate the matrix exponential, it is much simpler to implement in practice than the Laplace method of finding $p_{ij}(t)$.

Outlook: EM Approach

As an alternative to maximising (3), we can instead use the EM algorithm, which is the advocated method in [3]. It involves finding the following function (the E-step):

$$Q(\boldsymbol{\theta}; \mathbf{Y}, \boldsymbol{\theta}^{(m)}) = \sum_{k=0}^{\infty} \Big[\mathbb{E}[U_k|\mathbf{Y}, \boldsymbol{\theta}^{(m)}](\log(\lambda) - \alpha k) + \mathbb{E}[D_k|\mathbf{Y}, \boldsymbol{\theta}^{(m)}] \log(\mu) - \mathbb{E}[T_k|\mathbf{Y}, \boldsymbol{\theta}^{(m)}](\lambda k^2 e^{-\alpha k} + k\mu) \Big], \quad (5)$$

maximising over $\boldsymbol{\theta}$ to find $\boldsymbol{\theta}^{m+1}$ (the M-step), then iterating until convergence. Here, U_k, D_k represent the total number of up or down steps made at state k respectively, while T_k represents the total sojourn time at state k. $\mathbf{Y}$ represents an observed sample.

So far, I have been able to find the expectations, but we are still working on implementing the maximisation algorithm and hope to compare its accuracy against simply maximising (3).

Acknowledgements

I wish to thank my supervisors Dr Sophie Hautphenne and Dr Brendan Patch for their time and work in supervising me, as well as the School of Mathematics and Statistics for this wonderful opportunity. This experience has been quite invaluable to me, as it has allowed me to both develop my ability and grow my appreciation for scientific research.

References

[1] A. Kraus A.C. Davidson S. Hautphenne. “Parameter estimation for discretely observed linear birth-and-death processes”. In: *The International Biometric Society* (2020). DOI: 10.1111/biom.13282.

[2] Marc A. Suchard Forrest W. Crawford. “Transition probabilities for general birth–death processes with applications in ecology, genetics, and evolution”. In: *J. Math. Biol.* 65 (2011), pp. 553–580.

[3] Marc A. Suchard Forrest W. Crawford Vladimir N. Minin. “Estimation for General Birth-Death Processes”. In: *Journal of the American Statistical Association* 109.506 (2014), pp. 730–747. DOI: 10.1080/01621459.2013.866565.

[4] F. Gerber. “A parallel version of 'scipy.optimize.minimize(method='L-BFGS-B'”. In: (2020). DOI: 10.5281/zenodo.3888570.