



Parameter estimation for a discretely observed population process under Markov-modulation

Mathisca de Gunst^a, Bartek Knapik^{a,1}, Michel Mandjes^b, Birgit Sollie^{a,*}

^a Department of Mathematics, Vrije Universiteit Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

^b Korteweg-de Vries Institute (KdVI) for Mathematics, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands

ARTICLE INFO

Article history:

Received 18 October 2018

Received in revised form 12 April 2019

Accepted 18 June 2019

Available online 26 June 2019

Keywords:

Population process

Markov modulation

Infinite server queue

Maximum likelihood estimation

EM algorithm

ABSTRACT

A Markov-modulated independent sojourn process is a population process in which individuals arrive according to a Poisson process with Markov-modulated arrival rate, and leave the system after an exponentially distributed time. A procedure is developed to estimate the parameters of such a system, including those related to the modulation. It is assumed that the number of individuals in the system is observed at equidistant time points only, whereas the modulating Markov chain cannot be observed at all. An algorithm is set up for finding maximum likelihood estimates, based on the EM algorithm and containing a forward-backward procedure for computing the conditional expectations. To illustrate the performance of the algorithm the results of an extensive simulation study are presented.

© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Population processes have been studied extensively, owing to their applicability in a broad range of areas such as biology, medicine, economics and operations research. A specific type of population process is one in which individuals arrive in a system, but once in the system they do not interfere with each other, and their sojourn times – that is, the times the individuals spend in the system – are independent. In queueing theory, this is conveniently called an *infinite-server queue*. A commonly imposed assumption is that of Poisson arrivals, but in many real life systems the arrival process is substantially more variable, for example alternating between busy and quiet periods. In such situations the *Markov-modulated Poisson process* (MMPP) is a more suitable alternative. The MMPP is a doubly stochastic Poisson process of which the rate is determined by a finite, continuous-time Markov chain, also referred to as the background process, such that the rate switches between distinct values at the jump times of the modulating Markov chain.

In this paper we consider a population process with independent sojourn times – or of the infinite server type – fed by an MMPP arrival process, and will refer to it as a Markov-modulated independent sojourn (MMIS) process. This class of models can be seen as birth–death processes under modulation, and has applications across various disciplines, see for example, Anderson and Kurtz (2015), O’Cinneide and Purdue (1986) and Schwabe et al. (2012). We note that in queueing theory this process is also known as the M/M/∞ queue in a random environment (O’Cinneide and Purdue, 1986). For more background information on this kind of stochastic models, see for example Neuts (1981, Ch. 3 and 6). We also remark that the cumulative arrival process is a counting process.

* Corresponding author.

E-mail address: b.sollie@vu.nl (B. Sollie).

¹ Current affiliation: Samsung Research and Development Institute Poland, Plac Europejski 1, 00-844 Warsaw, Poland.

We are interested in estimating the unknown parameters of this system—including those related to the modulation. We assume that the population size can be observed, but the modulating Markov chain cannot. More specifically, the number of individuals present is recorded only at equidistant points in time, which reflects the fact that in most practical situations it is infeasible to observe the process continuously in time. We further assume that the times the individuals spend in the system are independent and exponentially distributed with a constant (that is, non-modulated) rate parameter. Since in many applications the departure rate can be controlled or is otherwise known, we develop our estimation procedure for the case that this rate parameter is known. We then show that under a natural additional assumption, the departure rate parameter can be estimated together with the parameters of the arrival process in a similar way.

An analysis of the MMIS process in terms of generating functions and moments can be found in O’Cinneide and Purdue (1986), while asymptotic properties have been investigated in Anderson et al. (2016) and Blom et al. (2013); see also the recent paper (Pang and Zhou, 2017). In the setting of queueing systems several parameter estimation procedures have been proposed for the analysis of infinite-server queues with non-modulated Poisson arrivals. We mention the method of moments and the maximum likelihood procedure for estimating the arrival rate developed in Pickands and Stine (1997). Here it is assumed that all arrivals and departures are observed, without knowing which departure belongs to which arrival. Another relevant reference is Bingham and Pitts (1999), in which a parametric and a non-parametric procedure are derived for estimating the arrival rate based on continuous-time observations of the population size, as well as a non-parametric estimation method based on the idle and busy periods.

Outside the setting of queueing systems, substantial attention has been paid to parameter estimation for counting processes which are affected by a hidden background process. Because of the hidden background process, missing data are intrinsic to estimation in this context, and the EM algorithm (Dempster et al., 1977) plays an essential role. The first class of models with a hidden background process for which parameter estimation was considered, is the class of discrete-time hidden Markov models. The Baum–Welch algorithm (Rabiner and Juang, 1986; Welch, 2003), which is used for parameter estimation in hidden Markov models, is essentially an EM algorithm. Later on, various continuous-time processes have been considered as well. It has been shown how to apply the EM algorithm to the class of phase-type distributions (Asmussen et al., 1996), the class of MMPPs (Rydén, 1996; Roberts et al., 2006) and, more generally, to the class of Markov-modulated Markov processes (Ephraim and Roberts, 2009). Rydén’s EM algorithm for MMPPs can also be used for parameter estimation in Markovian arrival processes. An improved algorithm for this case has been proposed in Okamura and Dohi (2009). In this body of literature estimation is performed based on observations of the counting process, the cumulative arrival process in the MMIS system. This is a marked difference with our setting, in which we wish to learn the system’s input parameters from data on the population size. A specific case of parameter estimation for a Markovian arrival process from population size data is presented in Hautphenne and Fackrell (2014), where Markovian binary trees are considered. This is, again, a setting very different from ours.

To the best of our knowledge, parameter estimation for MMIS processes under the particular assumptions on the data that we formulated above, has not been studied so far. In all papers listed above it is assumed that the counting process can be observed continuously in time, and as a result, missing data only arise due to the hidden nature of the background process. Parameter estimation results for arrival processes based on discretely observed data are known only for the class of Markovian arrival processes (Okamura et al., 2009; Breuer and Kume, 2010), for which naturally the EM algorithm was used as well. However, in these papers the data directly concern the cumulative arrival process as their models do not include departures, whereas in our setting we only indirectly observe the *effect* of the arrivals, namely through the population size.

In our context the parameter estimation is seriously complicated by (i) the fact that the modulating Markov chain of the Markov-modulated arrival process is not observed, and (ii) that the population size is not observed continuously in time. Issue (i) entails that it is not known when the arrival rate changes value, and issue (ii) that it is not possible to deduce from the observations the number of arrivals or the number of departures between two consecutive observations. To deal with these complications we treat the modulating continuous-time Markov chain and the number of arrivals between two consecutive observations as missing data, and, making use of the EM algorithm, develop an explicit algorithm to find maximum likelihood estimates of the parameters. Our approach borrows some elements of the one proposed by Okamura et al. (2009), but the adaptation of their estimation algorithm to our setting is not straightforward. Although we end up with the same type of parameter updates as the ones they employ, the computations of these updates require major adjustments to the steps of their algorithm. In particular, we use a different method to obtain the required transition probabilities and we redefine the forward and backward vectors. The conditional expectations in the parameter updates can be expressed as integrals containing these vectors in a similar way as in Okamura et al. (2009), but the computations of these integrals now demand solving a significantly more involved system of differential equations.

The remainder of this paper is organized as follows. In Section 2 we define our statistical model, the MMIS process, and we state the estimation problem. In Section 3 we derive the estimation algorithm. We investigate the accuracy of the proposed estimation method by a simulation study in Section 4. Section 5 discusses an extension in which the departure rate is not known and estimated as well. The paper concludes with a discussion in Section 6.

2. Model and estimation

We consider an MMIS process where the arrivals follow an MMPP driven by a modulating continuous-time Markov chain of which each state corresponds to a different arrival rate value. In this section the mathematical formulation of this model and the corresponding estimation problem are presented.

2.1. Markov-Modulated independent sojourn process

The modulating continuous-time Markov chain with state space $\{1, \dots, d\}$, $d \geq 2$, that defines the state of the arrival process at time t , will be denoted by $\{X_t\}_{t \geq 0}$. Its transition rate matrix is given by $Q = (q_{ij})_{i,j=1}^d$ and its initial state distribution at $t = 0$ by $\pi = (\pi_1, \dots, \pi_d)^\top$. We define $q_i = -q_{ii} = \sum_{j \neq i} q_{ij}$ as the total rate at which the Markov chain jumps out of state i . The MMPP models the cumulative arrival process $\{A_t\}_{t \geq 0}$ with corresponding time-inhomogeneous arrival rate $\lambda(t)$. This rate stochastically alternates between d different rates $\lambda_1, \dots, \lambda_d$ in such a way that $\lambda(t) = \lambda_i$ if $X_t = i$, for $i = 1, \dots, d$. We assume that the sojourn times are independent and identically exponentially distributed with rate $\mu > 0$.

Let $\{M_t\}_{t \geq 0}$ be the population size at time t . Then $\{M_t, X_t\}_{t \geq 0}$ is a joint Markov process with corresponding transition probabilities

$$p_{ij}(m, m'; t) = \mathbb{P}(M_t = m', X_t = j | M_0 = m, X_0 = i), \quad (1)$$

for all $t \geq 0$ and $m, m' \geq 0$. For each combination of m and m' , we define for $t > 0$, the $d \times d$ transition matrix $P_t(m, m')$ containing these transition probabilities by

$$[P_t(m, m')]_{ij} = p_{ij}(m, m'; t). \quad (2)$$

We assume that the process $\{M_t\}$ is observed at $n + 1$ equidistant time points $t_k = k\Delta$ for some fixed $\Delta > 0$, $0 \leq k \leq n$, and denote the corresponding observations by m_0, \dots, m_n . These $n + 1$ observations constitute the available data set. Associated with these data, we will write M_l^k for the vector $(M_{t_l}, \dots, M_{t_k})^\top$, and m_l^k for the vector of observations $(m_l, \dots, m_k)^\top$, $0 \leq l \leq k \leq n$.

For the forthcoming analysis, it will be convenient to introduce a number of additional random variables. The indicator random variable for the event that at time $t = 0$ the background process is in state i will be denoted by B_i , that is,

$$B_i = 1_{\{X_0=i\}}, \quad i = 1, \dots, d.$$

We also define, for all $k = 1, \dots, n$ and $i = 1, \dots, d$, the following random variables corresponding to the k th interval $(t_{k-1}, t_k]$:

$$\begin{aligned} Z_i^{[k]} &= \text{the total amount of time spent in state } i \text{ by } \{X_t\}; \\ Y_{ij}^{[k]} &= \text{the total number of state transitions of } \{X_t\} \text{ from state } i \text{ to } j \text{ (} j \neq i \text{)}; \\ A_i^{[k]} &= \text{the total number of arrivals while the background process is in state } i. \end{aligned}$$

Finally, in the sequel we will write $X = \{X_t : 0 \leq t \leq t_n\}$, $Y = \sum_{k=1}^n \sum_{i=1}^d \sum_{j \neq i} Y_{ij}^{[k]}$ and $A = \{A_i^{[k]} : i = 1, \dots, d, k = 1, \dots, n\}$. We see that Y equals the total number of jumps of $\{X_t\}$ in $(0, t_n]$. Let J_1, \dots, J_Y be the corresponding jump times of $\{X_t\}$, and $J_0 = 0$. The corresponding states after the jumps will be denoted by S_0, \dots, S_Y , so that $S_l = X_{J_l}$, $l = 0, \dots, Y$.

2.2. Parameter estimation

Our goal is to estimate the unknown parameters of an MMIS process given the number of states d , and observations m_0, \dots, m_n of M_{t_0}, \dots, M_{t_n} . We consider the setting that the departure rate μ is time-independent and known, and we thus concentrate on estimating the parameter vector $\theta = (\pi_i, q_{ij}, \lambda_i : i, j \in \{1, \dots, d\}, j \neq i)^\top$. The estimate will be denoted by $\hat{\theta} = (\hat{\pi}_i, \hat{q}_{ij}, \hat{\lambda}_i : i, j \in \{1, \dots, d\}, j \neq i)^\top$.

Let $v = (1, \dots, 1)^\top$ be a vector of size d . By taking into account the background process $\{X_t\}$ at the observation times and using (1) and (2), we find that the likelihood function \mathcal{L}_0 is given by

$$\begin{aligned} \mathcal{L}_0(\theta | M_0^n) &= \mathbb{P}_\theta(M_0^n = m_0^n) = \sum_{x_0, \dots, x_n} \mathbb{P}_\theta(M_{t_0} = m_0, X_{t_0} = x_0, \dots, M_{t_n} = m_n, X_{t_n} = x_n) \\ &= \pi^\top \left(\prod_{k=1}^n P_\Delta(m_{k-1}, m_k) \right) v. \end{aligned} \quad (3)$$

Maximum likelihood estimation based on (3) is problematic, since the likelihood is expressed in terms of matrix multiplications. These matrix multiplications appear due to the fact that we observe the process $\{M_t\}$ only at discrete time points and because the process $\{X_t\}$ is unobserved. As indicated above, we will use the EM algorithm to tackle the estimation problem.

3. The algorithm

The EM algorithm starts with an initial input value θ^0 and then updates the estimate $\hat{\theta}$ iteratively. Each iteration in the EM algorithm consists of an expectation step and a maximization step, which together produce a parameter update $\hat{\theta} = (\hat{\pi}_i, \hat{q}_{ij}, \hat{\lambda}_i : i, j \in \{1, \dots, d\}, j \neq i)^\top$ from a parameter input $\theta' = (\pi'_i, q'_{ij}, \lambda'_i : i, j \in \{1, \dots, d\}, j \neq i)^\top$. The key here is

that, instead of maximizing the loglikelihood \mathcal{L}_0 based on the observed data, the loglikelihood \mathcal{L} based on a larger data set is maximized. This larger data set, called the complete data set, consists of the observed data and missing data. The missing data can be missing observations, or, as in our case, conveniently chosen unobserved data.

In Section 3.1 we describe the expectation and maximization steps and derive expressions for the parameter updates $\tilde{\pi}_i$, \tilde{q}_{ij} and $\tilde{\lambda}_i$. Sections 3.2–3.5 elaborate on how to compute these expressions explicitly. Section 3.6 summarizes the entire algorithm with which the parameter estimates can be obtained.

3.1. Parameter updates

To perform the expectation and maximization steps, we consider (A, X) as the missing data, so that (M_0^n, A, X) is the complete data set. The loglikelihood function of the complete data is then

$$\log \mathcal{L}(\theta | M_0^n, A, X) = \log \mathbb{P}_\theta(M_0^n | A, X) + \log \mathbb{P}_\theta(A, X). \quad (4)$$

For the expectation step of the EM algorithm, we have to compute

$$\begin{aligned} \mathbb{E}_{\theta'} [\log \mathcal{L}(\theta | M_0^n, A, X) | M_0^n = m_0^n] \\ = \mathbb{E}_{\theta'} [\log \mathbb{P}_\theta(M_0^n | A, X) | M_0^n = m_0^n] + \mathbb{E}_{\theta'} [\log \mathbb{P}_\theta(A, X) | M_0^n = m_0^n]. \end{aligned} \quad (5)$$

For the maximization step we compute

$$\tilde{\theta} = \arg \max_{\theta} \mathbb{E}_{\theta'} [\log \mathcal{L}(\theta | M_0^n, A, X) | M_0^n = m_0^n] = \arg \max_{\theta} \mathbb{E}_{\theta'} [\log \mathbb{P}_\theta(A, X) | M_0^n = m_0^n]. \quad (6)$$

Note that (6) follows from (5) because the first term on the right hand side of (4) – and hence also of (5) – only depends on the known departure rate μ and not on the unknown parameter θ . Hence, given A and X , the arrivaltimes in each state follow a uniform distribution by a well-known property of the homogeneous Poisson process, and therefore $\mathbb{P}_\theta(M_0^n | A, X)$ translates into a probability on the departures only.

To find the parameter update $\tilde{\theta}$ the expectation $\mathbb{E}_{\theta'} [\log \mathbb{P}_\theta(A, X) | M_0^n = m_0^n]$ needs to be computed. We first observe that

$$\mathbb{P}_\theta(A, X) = \mathbb{P}_\theta(A | X) \mathbb{P}_\theta(X). \quad (7)$$

By using the partition of the interval $(0, t_n]$ into the observation intervals $(t_{k-1}, t_k]$, $k = 1, \dots, n$, we see that

$$\mathbb{P}_\theta(A | X) = \prod_{k=1}^n \prod_{i=1}^d \frac{(\lambda_i Z_i^{[k]})^{A_i^{[k]}}}{(A_i^{[k]})!} e^{-\lambda_i Z_i^{[k]}}. \quad (8)$$

For the computation of $\mathbb{P}_\theta(X)$ we do not use this partition, but consider the entire interval $(0, t_n]$. We have

$$\mathbb{P}_\theta(X) = \pi_{s_0} \prod_{y=1}^Y \left(q_{s_{y-1}s_y} e^{-q_{s_{y-1}}} (j_y - j_{y-1}) \right) e^{-q_{s_Y} (t_n - j_Y)}. \quad (9)$$

Combining (7), (8) and (9), and rewriting the obtained expression by aggregating all terms with π_i , all terms with q_{ij} and all terms with λ_i , we find

$$\begin{aligned} \log \mathbb{P}_\theta(A, X) = & \sum_{i=1}^d \log(\pi_i) B_i + \sum_i \sum_{j \neq i} \sum_{k=1}^n \left(Y_{ij}^{[k]} \log(q_{ij}) - q_{ij} Z_i^{[k]} \right) \\ & + \sum_{i=1}^d \sum_{k=1}^n \left(A_i^{[k]} \log(\lambda_i) - \lambda_i Z_i^{[k]} \right) + \sum_{i=1}^d \sum_{k=1}^n \left(A_i^{[k]} \log(Z_i^{[k]}) - \log(A_i^{[k]})! \right). \end{aligned} \quad (10)$$

Substituting this result into (6) and solving the equation for $\tilde{\theta}$, we obtain the parameter updates

$$\begin{aligned} \tilde{\pi}_i &= \mathbb{E}_{\theta'} [B_i | M_0^n = m_0^n], \\ \tilde{q}_{ij} &= \frac{\sum_{k=1}^n \mathbb{E}_{\theta'} [Y_{ij}^{[k]} | M_0^n = m_0^n]}{\sum_{k=1}^n \mathbb{E}_{\theta'} [Z_i^{[k]} | M_0^n = m_0^n]}, \\ \tilde{\lambda}_i &= \frac{\sum_{k=1}^n \mathbb{E}_{\theta'} [A_i^{[k]} | M_0^n = m_0^n]}{\sum_{k=1}^n \mathbb{E}_{\theta'} [Z_i^{[k]} | M_0^n = m_0^n]}. \end{aligned} \quad (11)$$

In the next three sections, we further elaborate on how to compute these parameter updates explicitly.

3.2. Transition probabilities

Before the parameter updates of (11) can be computed, some preliminary steps need to be taken. First we show how to obtain approximations of the transition probability matrices defined in (2), which will be used in the next steps.

Exact computation of the transition probability matrices is not feasible, since the computation of the transition probabilities would need taking the exponent of the transition rate matrix of $\{M_t, X_t\}$, which is problematic for systems with infinite state space. We therefore approximate our MMIS process by a process in which the population size is bounded from above by a finite number C . In other words, we consider the same system as described above but now with the restriction that there can be at most C individuals in the system. New arrivals are blocked whenever the system is full. Because of this restriction, the system will behave slightly differently, but we can choose C large enough such that $\mathbb{P}(M_t = m)$ is negligible for $m > C$ for all t . The transition probabilities of the original system can then be well approximated by the transition probabilities of the process with population size bounded by C , and we can use

$$P_t(m, m') \approx P_t^C(m, m'), \quad (12)$$

where $P_t^C(m, m')$ is the analog of $P_t(m, m')$ for the system with population size bounded by C .

For the system with population size bounded by C we can find the transition probabilities by taking the exponent of its $d(C+1) \times d(C+1)$ transition rate matrix R^C . This transition rate matrix has a tridiagonal form and is given by

$$R^C = \begin{pmatrix} R_0 & R_1 & & & 0 \\ \mu I_d & R_0 - \mu I_d & \ddots & & \\ & 2\mu I_d & \ddots & R_1 & \\ 0 & & \ddots & R_0 - (C-1)\mu I_d & R_1 \\ & 0 & & C\mu I_d & Q - C\mu I_d \end{pmatrix},$$

where I_d is the $d \times d$ identity matrix, $R_1 = \text{diag}\{\lambda_1, \dots, \lambda_d\}$ and $R_0 = Q - R_1$. The t -time transition probability matrix is obtained by taking the matrix exponent of $R^C t$. Note that this probability matrix is a composition of the $d \times d$ block matrices $P_t^C(m, m')$ in which, for fixed $0 \leq m, m' \leq C$, the (i, j) th entry is equal to the transition probability $p_{ij}^C(m, m'; t)$. More specifically, let us define, for $0 \leq i \leq C$, e_i as the $d(C+1) \times d$ matrix which consists of the identity matrix I_d at the $(i+1)$ th block and zeros elsewhere. Then

$$P_t^C(m, m') = e_m^\top [e^{R^C t}] e_{m'}. \quad (13)$$

3.3. Forward and backward vectors

We will now introduce the forward and backward vectors that are involved in the EM-algorithm for the MMIS process and show how to compute them. These forward and backward vectors will be used to obtain the conditional expectations in (11).

The forward vector $f_{k,\theta}(m, u)$ is defined for $k = 0, \dots, n$, $m \geq 0$ and $0 \leq u \leq \Delta$, as the vector of length d with i th entry

$$[f_{k,\theta}(m, u)]_i = \mathbb{P}_\theta(M_0^k = m_0^k, M_{(t_k+u)^-} = m, X_{(t_k+u)^-} = i). \quad (14)$$

The backward vector $b_{k,\theta}(m, u)$ is defined for $k = 0, \dots, n-1$, $m \geq 0$ and $0 \leq u \leq \Delta$, as the vector of length d with i th entry

$$[b_{k,\theta}(m, u)]_i = \mathbb{P}_\theta(M_k^n = m_k^n | M_{(t_k-u)^+} = m, X_{(t_k-u)^+} = i). \quad (15)$$

In the above, the notation M and X with indices $(t_k + u)^-$ and $(t_k - u)^+$ indicates the values of M and X just before time $(t_k + u)$ and just after time $(t_k - u)$, respectively.

To compute $f_{k,\theta}(m, u)$ and $b_{k,\theta}(m, u)$ in an efficient way, we first consider the special cases $f_{k,\theta} = f_{k,\theta}(m_k, 0)$ and $b_{k,\theta} = b_{k,\theta}(m_k, 0)$. In view of (1) and (2) we have for $i = 1, 2$,

$$\begin{aligned} [f_{k,\theta}]_i &= \sum_{x_0, \dots, x_{k-1}} \pi_{x_0} \prod_{l=1}^k \mathbb{P}_\theta(M_{t_l} = m_l, X_{t_l} = x_l | M_{t_{l-1}} = m_{l-1}, X_{t_{l-1}} = x_{l-1}) \\ &= \left[\left(\prod_{l=1}^k P_\Delta(m_{l-1}, m_l) \right)^\top \pi \right]_i, \end{aligned}$$

and

$$\begin{aligned} [b_{k,\theta}]_i &= \sum_{x_{k+1}, \dots, x_n} \prod_{l=k+1}^n \mathbb{P}_\theta(M_{t_l} = m_l, X_{t_l} = x_l | M_{t_{l-1}} = m_{l-1}, X_{t_{l-1}} = x_{l-1}) \\ &= \left[\left(\prod_{l=k+1}^n P_\Delta(m_{l-1}, m_l) \right) v \right]_i. \end{aligned}$$

Since, for $k = 1, \dots, n$,

$$f_{k,\theta} = P_{\Delta}(m_{k-1}, m_k)^{\top} f_{k-1,\theta}, \quad \text{with initial condition } f_{0,\theta} = \pi, \quad (16)$$

and for $k = 0, \dots, n-1$,

$$b_{k,\theta} = P_{\Delta}(m_k, m_{k+1}) b_{k+1,\theta}, \quad \text{with initial condition } b_{n,\theta} = v, \quad (17)$$

we see that $f_{k,\theta}$ and $b_{k,\theta}$ can be computed recursively. After the computation of $f_{1,\theta}, \dots, f_{n,\theta}$ and $b_{0,\theta}, \dots, b_{n-1,\theta}$ by the recurrence relations (16) and (17), respectively, the general versions $f_{k,\theta}(m, u)$ and $b_{k,\theta}(m, u)$ can be computed by

$$f_{k,\theta}(m, u) = P_u(m_k, m)^{\top} f_{k,\theta},$$

and

$$b_{k,\theta}(m, u) = P_u(m, m_k) b_{k,\theta},$$

which can be evaluated using (12) and (13).

3.4. Conditional expectations

Having seen how to compute the transition probability matrices of (2) and the forward and backward vectors, which are necessary tools for the computation of the conditional expectations in (11), we are ready to derive expressions for these conditional expectations in terms of the $f_{k,\theta}(m, u)$ and $b_{k,\theta}(m, u)$. Below we will make frequent use of the definitions (14) and (15) of these forward and backward vectors.

As a start, we note that

$$\mathbb{E}_{\theta'}[B_i | M_0^n = m_0^n] = \frac{1}{\mathbb{P}_{\theta'}(M_0^n = m_0^n)} \mathbb{E}_{\theta'}[B_i \mathbf{1}_{\{M_0^n = m_0^n\}}]. \quad (18)$$

In (18) B_i can be replaced by $Y_{i,j}^{[k]}$, $Z_i^{[k]}$ or $A_i^{[k]}$, to obtain an analogous relationship for the other conditional expectations in (11). Since $\mathbb{P}_{\theta'}(M_0^n = m_0^n)$ on the right-hand side of these relations is given by (3), we only need expressions in terms of the forward and backward vectors for the expectations $\mathbb{E}_{\theta'}[B_i \mathbf{1}_{\{M_0^n = m_0^n\}}]$, $\mathbb{E}_{\theta'}[Y_{i,j}^{[k]} \mathbf{1}_{\{M_0^n = m_0^n\}}]$, $\mathbb{E}_{\theta'}[Z_i^{[k]} \mathbf{1}_{\{M_0^n = m_0^n\}}]$ and $\mathbb{E}_{\theta'}[A_i^{[k]} \mathbf{1}_{\{M_0^n = m_0^n\}}]$.

First, we observe that $\mathbb{E}_{\theta'}[B_i \mathbf{1}_{\{M_0^n = m_0^n\}}]$ is simply computed by

$$\mathbb{E}_{\theta'}[B_i \mathbf{1}_{\{M_0^n = m_0^n\}}] = \mathbb{P}_{\theta'}(X_0 = i, M_0^n = m_0^n) = \pi'_i [b_{0,\theta'}]_i. \quad (19)$$

Second, for $\mathbb{E}_{\theta'}[Y_{i,j}^{[k]} \mathbf{1}_{\{M_0^n = m_0^n\}}]$, we have that

$$\begin{aligned} & \mathbb{E}_{\theta'}[Y_{i,j}^{[k]} \mathbf{1}_{\{M_0^n = m_0^n\}}] \\ &= \int_0^{\Delta} \mathbb{P}_{\theta'}(X_{(t_{k-1}+\tau)^-} = i, X_{(t_{k-1}+\tau)^+} = j, M_0^n = m_0^n) d\tau \\ &= \int_0^{\Delta} \sum_{m=0}^{\infty} \mathbb{P}_{\theta'}(M_k^n = m_k^n | X_{(t_{k-1}+\tau)^+} = j, M_{(t_{k-1}+\tau)^+} = m) q'_{ij} \\ & \quad \mathbb{P}_{\theta'}(M_0^{k-1} = m_0^{k-1}, X_{(t_{k-1}+\tau)^-} = i, M_{(t_{k-1}+\tau)^-} = m) d\tau, \end{aligned} \quad (20)$$

where in the second step we conditioned on the population size at time τ , and used the Markov property. The last integral in (20) can be rewritten in terms of the forward and backward vectors $f_{k,\theta}(m, u)$ and $b_{k,\theta}(m, u)$, to obtain

$$\mathbb{E}_{\theta'}[Y_{i,j}^{[k]} \mathbf{1}_{\{M_0^n = m_0^n\}}] = \int_0^{\Delta} \sum_{m=0}^{\infty} [f_{k-1,\theta'}(m, \tau)]_i q'_{ij} [b_{k,\theta'}(m, \Delta - \tau)]_j d\tau. \quad (21)$$

In a similar way, we have for $\mathbb{E}_{\theta'}[Z_i^{[k]} \mathbf{1}_{\{M_0^n = m_0^n\}}]$,

$$\begin{aligned} & \mathbb{E}_{\theta'}[Z_i^{[k]} \mathbf{1}_{\{M_0^n = m_0^n\}}] \\ &= \int_0^{\Delta} \mathbb{P}_{\theta'}(X_{t_{k-1}+\tau} = i, M_0^n = m_0^n) d\tau \\ &= \int_0^{\Delta} \sum_{m=0}^{\infty} \mathbb{P}_{\theta'}(M_0^{k-1} = m_0^{k-1}, M_{t_{k-1}+\tau} = m, X_{t_{k-1}+\tau} = i) \\ & \quad \mathbb{P}_{\theta'}(M_k^n = m_k^n | M_{t_{k-1}+\tau} = m, X_{t_{k-1}+\tau} = i) d\tau. \end{aligned} \quad (22)$$

Rewriting (22) in terms of $f_{k,\theta}(m, u)$ and $b_{k,\theta}(m, u)$, we get

$$\mathbb{E}_{\theta'}[Z_i^{[k]} \mathbf{1}_{\{M_0^n = m_0^n\}}] = \int_0^\Delta \sum_{m=0}^\infty [f_{k-1,\theta'}(m, \tau)]_i [b_{k,\theta'}(m, \Delta - \tau)]_i d\tau. \quad (23)$$

Lastly, we find the expression for $\mathbb{E}_{\theta'}[A_i^{[k]} \mathbf{1}_{\{M_0^n = m_0^n\}}]$. Let \mathcal{A}_t be the event that an arrival occurs at time $t > 0$. Then

$$\begin{aligned} \mathbb{E}_{\theta'}[A_i^{[k]} \mathbf{1}_{\{M_0^n = m_0^n\}}] &= \int_0^\Delta \mathbb{P}_{\theta'}(\mathcal{A}_\tau, X_\tau = i, M_0^n = m_0^n) d\tau \\ &= \int_0^\Delta \sum_{m=0}^\infty \mathbb{P}_{\theta'}(M_0^{k-1} = m_0^{k-1}, M_{(t_{k-1}+\tau)^-} = m, X_{(t_{k-1}+\tau)^-} = i) \\ &\quad \mathbb{P}_{\theta'}(M_{(t_{k-1}+\tau)^+} = m+1, X_{(t_{k-1}+\tau)^+} = i \mid M_{(t_{k-1}+\tau)^-} = m, X_{(t_{k-1}+\tau)^-} = i) \\ &\quad \mathbb{P}_{\theta'}(M_k^n = m_k^n \mid M_{(t_{k-1}+\tau)^+} = m+1, X_{(t_{k-1}+\tau)^+} = i) d\tau, \end{aligned}$$

and rewriting in terms of $f_{k,\theta}(m, u)$ and $b_{k,\theta}(m, u)$ yields

$$\mathbb{E}_{\theta'}[A_i^{[k]} \mathbf{1}_{\{M_0^n = m_0^n\}}] = \int_0^\Delta \sum_{m=0}^\infty [f_{k-1,\theta'}(m, \tau)]_i \lambda'_i [b_{k,\theta'}(m+1, \Delta - \tau)]_i d\tau. \quad (24)$$

We note that the obtained results (21), (23) and (24) are very similar, but that there are some minor but crucial differences. The entries of the forward and backward vectors differ per expression and the variable in the backward vector is equal to $m+1$ in (24) in contrast to m in (21) and (23).

3.5. Differential equations method

In order to use Eqs. (21), (23) and (24) for computing the parameter updates of (11), we need a method to compute the integrals on their right-hand sides in an efficient way. For this we propose a differential equations method.

Because $[f_{k-1,\theta'}(m, \tau)]_i$ is negligible for $m > C$, we may truncate the sums in (21), (23) and (24) at the finite bound C . Next, we introduce, for $0 \leq m, m' \leq C$ and $t > 0$, the $d \times d$ matrix

$$G_{k,\theta}(m, m', t) = \int_0^t b_{k,\theta}(m', t - \tau) f_{k-1,\theta}(m, \tau)^\top d\tau, \quad (25)$$

so that the conditional expectations in (11) become equal to

$$\mathbb{E}_{\theta'}[Y_{i,j}^{[k]} \mid M_0^n = m_0^n] = \frac{q'_{ij}}{\mathbb{P}_{\theta'}(M_0^n = m_0^n)} \sum_{m=0}^C [G_{k,\theta'}(m, m, \Delta)]_{i,j}, \quad (26)$$

$$\mathbb{E}_{\theta'}[Z_i^{[k]} \mid M_0^n = m_0^n] = \frac{1}{\mathbb{P}_{\theta'}(M_0^n = m_0^n)} \sum_{m=0}^C [G_{k,\theta'}(m, m, \Delta)]_{i,i}, \quad (27)$$

$$\mathbb{E}_{\theta'}[A_i^{[k]} \mid M_0^n = m_0^n] = \frac{\lambda'_i}{\mathbb{P}_{\theta'}(M_0^n = m_0^n)} \sum_{m=0}^{C-1} [G_{k,\theta'}(m, m+1, \Delta)]_{i,i}. \quad (28)$$

To facilitate the computation of (26)–(28), we derive a system of differential equations for $G_{k,\theta}(m, m', t)$.

For $f_{k,\theta}(m, u)$ and $b_{k,\theta}(m, u)$ a system of differential equations can be easily obtained from the derivatives of the corresponding transition probabilities while making use of (13), (14) and (15). For the forward vector, this yields for $1 \leq m \leq C-1$,

$$\begin{aligned} \frac{d}{du} f_{k,\theta}(0, u) &= R_0^\top f_{k,\theta}(0, u) + \mu f_{k,\theta}(1, u), \\ \frac{d}{du} f_{k,\theta}(m, u) &= R_1^\top f_{k,\theta}(m-1, u) + R_0^\top f_{k,\theta}(m, u) + (m+1)\mu f_{k,\theta}(m+1, u) - m\mu f_{k,\theta}(m, u), \\ \frac{d}{du} f_{k,\theta}(C, u) &= Q^\top f_{k,\theta}(C, u) + R_1^\top f_{k,\theta}(C-1, u) - C\mu f_{k,\theta}(C, u), \end{aligned} \quad (29)$$

with initial condition $f_{0,\theta}(m_0, 0) = \pi$, and for the backward vector we find for $0 \leq m \leq C-1$,

$$\begin{aligned} \frac{d}{du} b_{k,\theta}(m, u) &= R_1 b_{k,\theta}(m+1, u) + R_0 b_{k,\theta}(m, u) + m\mu b_{k,\theta}(m-1, u) - m\mu b_{k,\theta}(m, u), \\ \frac{d}{du} b_{k,\theta}(C, u) &= Q b_{k,\theta}(C, u) + C\mu b_{k,\theta}(C-1, u) - C\mu b_{k,\theta}(C, u), \end{aligned} \quad (30)$$

with initial condition $b_{n,\theta}(m_n, 0) = v$. Furthermore, from (25) we get

$$\frac{d}{dt}G_{k,\theta}(m, m', t) = \int_0^t \frac{d}{dt}b_{k,\theta}(m', t - \tau)f_{k-1,\theta}(m, \tau)^\top d\tau + b_{k,\theta}(m', 0)f_{k-1,\theta}(m, t)^\top. \quad (31)$$

Combining the differential equations for the backward vectors from (30) and (31), we obtain, for $m = 0, \dots, C$, the following system of differential equation for $G_{k,\theta}(m, m', t)$:

$$\begin{aligned} \frac{d}{dt}G_{k,\theta}(m, m', t) &= R_1 G_{k,\theta}(m, m' + 1, t) + R_0 G_{k,\theta}(m, m', t) + m' \mu G_{k,\theta}(m, m' - 1, t) \\ &\quad - m' \mu G_{k,\theta}(m, m', t) + b_{k,\theta}(m', 0)f_{k-1,\theta}(m, t)^\top, \quad 0 \leq m' \leq C - 1 \\ \frac{d}{dt}G_{k,\theta}(m, C, t) &= Q G_{k,\theta}(m, C, t) + C \mu G_{k,\theta}(m, C - 1, t) \\ &\quad - C \mu G_{k,\theta}(m, C, t) + b_{k,\theta}(C, 0)f_{k-1,\theta}(m, t)^\top. \end{aligned} \quad (32)$$

Note that these differential equations contain the term $b_{k,\theta}(m', 0)f_{k-1,\theta}(m, t)^\top$, in which $f_{k-1,\theta}(m, t)$ depends on the variable of differentiation t . Therefore, the differential equations for the forward vectors in (29) are needed to solve the differential equations for $G_{k,\theta}(m, m', t)$.

Analyzing the system of differential equations in (32) a bit further, we observe that the system can be split into d independent systems of differential equations. For this, we consider each column of the matrix $G_{k,\theta}(m, m', t)$ separately. Let the j th column of $G_{k,\theta}(m, m', t)$ be denoted by $[G_{k,\theta}(m, m', t)]_j$. Then for each $j = 1, \dots, d$, we stack the j th columns of the matrices $G_{k,\theta}(0, m', t), \dots, G_{k,\theta}(C, m', t)$, into $d(C + 1)$ -dimensional vectors of the form

$$G_{k,\theta}(m', t)_j = \begin{pmatrix} [G_{k,\theta}(0, m', t)]_j \\ [G_{k,\theta}(1, m', t)]_j \\ \vdots \\ [G_{k,\theta}(C, m', t)]_j \end{pmatrix}.$$

From (32) it follows that

$$\frac{d}{dt}G_{k,\theta}(m', t)_j = R^C G_{k,\theta}(m', t)_j + c_{k,\theta}(t)_j, \quad (33)$$

where $c_{k,\theta}(t)_j$ is a vector containing $[f_{k-1,\theta}(m', t)]_j [b_{k,\theta}]_1$ and $[f_{k-1,\theta}(m', t)]_j [b_{k,\theta}]_2$ at its entries $2m' + 1$ and $2m' + 2$ respectively, and zeros elsewhere. We note that for each $j = 1, \dots, d$, (33) is a linear system of differential equations for which the solution is equal to

$$G_{k,\theta}(m', t)_j = \int_0^t e^{R^C(t-s)} c_{k,\theta}(s)_j ds.$$

3.6. Summarized algorithm

In Sections 3.1–3.5, we elaborated on the expectation and maximization steps to find the parameter updates. Iteratively repeating the above obtained building blocks for computing the parameter updates results in the complete algorithm for obtaining estimates of the arrival parameters. The algorithm is presented below.

Algorithm

- 1 Determine initial values $\theta^0 = (\pi_i^0, q_{ij}^0, \lambda_i^0 : i, j \in \{1, \dots, d\}, j \neq i)^\top$ and set $\theta' = \theta^0$.
- 2 Compute $f_{k,\theta'}$ for $k = 1, \dots, n - 1$ and $b_{k,\theta'}$ for $k = 0, \dots, n - 1$ by recurrence relations (16) and (17).
- 3 Compute $G_{k,\theta'}(m, m, \Delta)$ and $G_{k,\theta'}(m, m + 1, \Delta)$ for all $m = 0, \dots, C$ and $k = 1, \dots, n$ by solving differential equations (32).
- 4 Compute conditional expectations (19), (26), (27) and (28) for all $k = 1, \dots, n$.
- 5 Compute parameter updates according to (11).
- 6 If stopping criterion is not satisfied, set $\theta' = \tilde{\theta}$ and go to step 2; else stop algorithm and use final parameter update $\tilde{\theta}$ as parameter estimate $\hat{\theta} = (\hat{\pi}_i, \hat{q}_{ij}, \hat{\lambda}_i : i, j \in \{1, \dots, d\}, j \neq i)^\top$.

Remark 1. The stopping criterion for the algorithm can be chosen in different ways. As proposed in Okamura et al. (2009), a reasonable choice is to let the stopping criterion depend on the difference in the loglikelihood functions based on the observed data. In this case the stopping criterion would be given by

$$\left| \log \mathcal{L}_0(\tilde{\theta} \mid M_0^n = m_0^n) - \log \mathcal{L}_0(\theta' \mid M_0^n = m_0^n) \right| < \varepsilon, \quad (34)$$

where ε can be chosen arbitrarily small. Another possibility is to let the stopping criterion depend on the difference between the obtained parameter updates. The stopping criterion would then be

$$\frac{\|\tilde{\theta} - \theta'\|}{\|\theta'\|} < \varepsilon.$$

Remark 2. The observation times t_0, \dots, t_n are defined as equidistant time points with $t_k = k\Delta$. However, the current approach can be applied with general, non-equidistant time points as well. For this, define the sequence $\Delta_1, \dots, \Delta_n$ of lengths of the observation intervals, hence $\Delta_k = t_k - t_{k-1}$ and $t_k = \sum_{i=1}^k \Delta_i$. The parameter estimates in (11) will stay the same, but the integrals in (21), (23) and (24) will have upper bound Δ_k instead of Δ , and hence, the $G_{k,\theta'}$ matrices in (26)–(28) will depend on Δ_k instead of Δ . The algorithm in Section 3.6 will remain completely the same.

Remark 3. The main differences between the steps in the algorithm above and the one presented in Okamura et al. (2009) are the following. In our algorithm the forward and backward vectors featuring in step 2 are redefined in terms of population size (instead of arrivals), and we need a different method to obtain the transition probabilities required to compute these vectors. Additionally, the differential equations appearing in step 3 are more involved, since the $G_{k,\theta}(m, m', t)$ matrices here are defined in terms of population size as well, and therefore require a different solution method. Moreover, our model includes departures and our algorithm enables the estimation of the departure rate (see Section 5), which is not considered in Okamura et al. (2009).

Remark 4. With the algorithm not only point estimates, but also bootstrap confidence intervals can be obtained, see Section 4.1.2.

4. Simulations

In this section, we investigate the accuracy of the proposed algorithm by means of a simulation study. The algorithm was applied to several simulated data sets with varying values of the model parameters, including the time between two observations, Δ , and the sample size n . We simulated the MMIS process with two states, that is $d = 2$, and considered examples in the relevant regime where the background process is relatively slow with respect to the arrival process. If the background process is too fast, the modulated arrival process will be averaged to a Poisson process with a homogeneous rate equal to $\lambda_\infty = \pi_1 \lambda_1 + \pi_2 \lambda_2$ (see Anderson et al., 2016), and as a consequence the modulation will not be detectable from data on the population size.

The algorithm was implemented in MATLAB with the likelihood-based stopping criterion (34). Choosing the initial values is a pragmatic procedure and depends on the data setting. Here, the effect of the choice on the initial value for π is negligible, since the parameter updates for π quickly converge to a 0-1 vector. For this reason, the initial value π^0 was set to (0.5, 0.5), and the results on $\hat{\pi}$ were omitted in this section. We used a rough guess on the trace of the background process in combination with moment estimators to find the initial values for the other parameters. There is no crucial difference between the implementation of the algorithm for $d > 2$ and for $d = 2$. The analysis presented in Section 3.5 reveals that as d increases by one, only the length of the vectors $G_{k,\theta}(m', t)_i$ and the size of the matrix R^C change, both by $C + 1$. Importantly, in the vast majority of real life processes for which an MMPP is an appropriate model, the number of states is low, and typically 2.

In Section 4.1, we discuss the influence of Δ and n on the parameter estimates by varying the values of Δ and n . In Section 4.2, we explore the influence of the timescale of the background process $\{X_t\}$ on the parameter estimates by varying the values of the parameters q_1 and q_2 .

4.1. Influence of Δ and n

We considered the MMIS process with parameter values $\pi = (1, 0)^\top$, $q_1 = 0.3$, $q_2 = 0.9$, $\lambda_1 = 4$, $\lambda_2 = 18$, and $\mu = 0.6$. We simulated 100 times the complete path up to time $T = t_n = n\Delta$ of the background process $\{X_t\}$ and the corresponding population process $\{M_t\}$ with these parameter values. From this we computed for each of the 100 simulations for various values of Δ and n the realization of the data vector $(M_{t_0}, \dots, M_{t_n})$, which corresponds to the available data if one would observe the number of individuals at times t_0, \dots, t_n only. To investigate the influence of the interval length Δ , we fixed the total observation time $T = 100$, and considered $\Delta = 0.1$, $\Delta = 0.05$ and $\Delta = 0.025$. To investigate the influence of the number of observations n , we fixed $\Delta = 0.05$ and chose $n = 500$, $n = 1000$, $n = 2000$, $n = 3000$ and $n = 4000$. Because for $n = 2000$ we could use the data vectors that were already computed from the 100 simulations for the combination $T = 100$ and $\Delta = 0.05$, from each simulation with $\Delta = 0.05$ four additional data vectors had to be generated for the other values of n .

Table 1

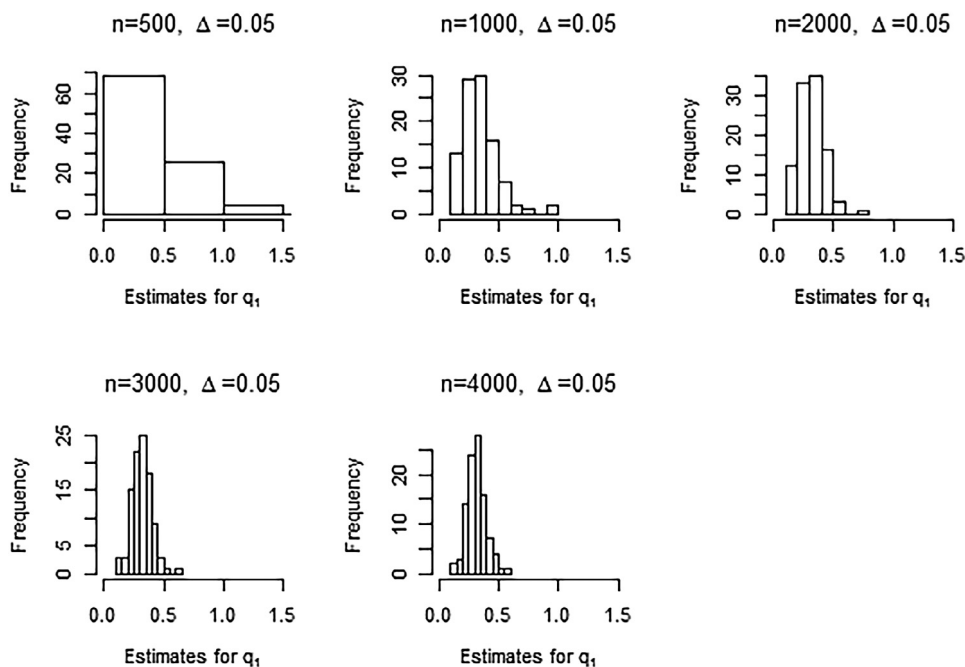
Mean of estimates of 100 data sets, with corresponding standard deviation between brackets. True parameter values: $q_1 = 0.3$, $q_2 = 0.9$, $\lambda_1 = 4$, $\lambda_2 = 18$.

		q_1	q_2	λ_1	λ_2
$n = 1000$	$\Delta = 0.1$	0.321 (0.105)	0.955 (0.370)	3.883 (0.372)	17.952 (1.528)
$n = 2000$	$\Delta = 0.05$	0.321 (0.101)	0.954 (0.346)	3.902 (0.344)	17.911 (1.416)
$n = 4000$	$\Delta = 0.025$	0.317 (0.089)	0.958 (0.320)	3.945 (0.323)	17.960 (1.346)

Table 2

Mean of estimates of 100 data sets, with corresponding standard deviation between brackets for $\Delta = 0.05$. True parameter values: $q_1 = 0.3$, $q_2 = 0.9$, $\lambda_1 = 4$, $\lambda_2 = 18$.

	q_1	q_2	λ_1	λ_2
$n = 500$	0.445 (0.441)	1.205 (0.897)	3.899 (0.847)	17.270 (3.780)
$n = 1000$	0.345 (0.152)	1.026 (0.556)	3.918 (0.483)	17.751 (2.004)
$n = 2000$	0.321 (0.101)	0.954 (0.346)	3.902 (0.344)	17.911 (1.416)
$n = 3000$	0.315 (0.083)	0.935 (0.281)	3.942 (0.266)	18.028 (1.224)
$n = 4000$	0.316 (0.076)	0.940 (0.228)	3.969 (0.236)	18.087 (1.044)

**Fig. 1.** Histograms of the obtained estimates for q_1 , with n increasing from left to right.

4.1.1. Results

The results of the first part of the study, where we considered the three different values of Δ , are presented in Table 1. This table shows for each Δ (rows) and each parameter (columns), the mean of the 100 estimates together with the corresponding standard deviation between brackets. All rows are quite similar, from which we can conclude that $\Delta = 0.1$ is already small enough to obtain estimates which lie close to the true parameter values. However, as Δ decreases there is a small decrease in the standard deviations, hence the estimates become more accurate as Δ decreases.

For the second part of the study, where we examined increasing values of n , the results are shown in Table 2 and Figs. 1–4. Table 2 contains for each sample size (rows) and for each parameter (columns), the mean values of the 100 estimates together with the corresponding standard deviation between brackets. Figs. 1–4 show histograms of the 100 estimates for the parameters q_1 , q_2 , λ_1 and λ_2 , respectively, where each figure contains five histograms corresponding to the five different values of n . We see from Table 2 that the means of the estimates lie closer to the true parameter values for larger values of n . Furthermore, the standard deviations decrease as n increases, which means that the estimates become more accurate when n gets larger. The decrease in standard deviation is also visible in the histograms. Each figure shows that the estimates are concentrated around the true parameter value, but their standard deviation clearly decreases as n becomes larger. In addition, the histograms look more and more bell-shaped, which is indicative of the distributions of the estimators becoming approximately normal when n increases.

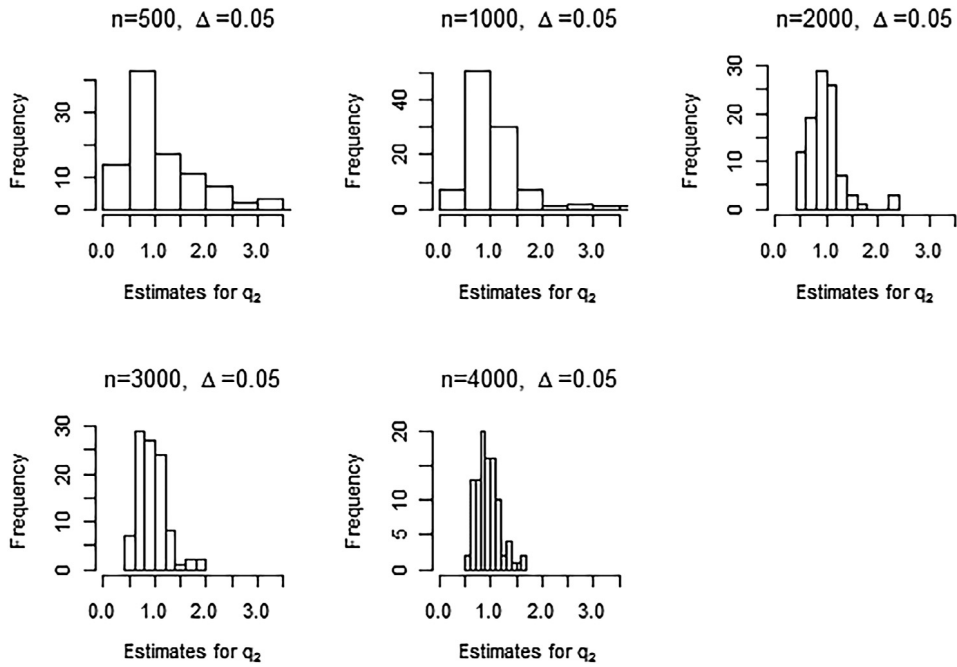


Fig. 2. Histograms of the obtained estimates for q_2 , with n increasing from left to right.

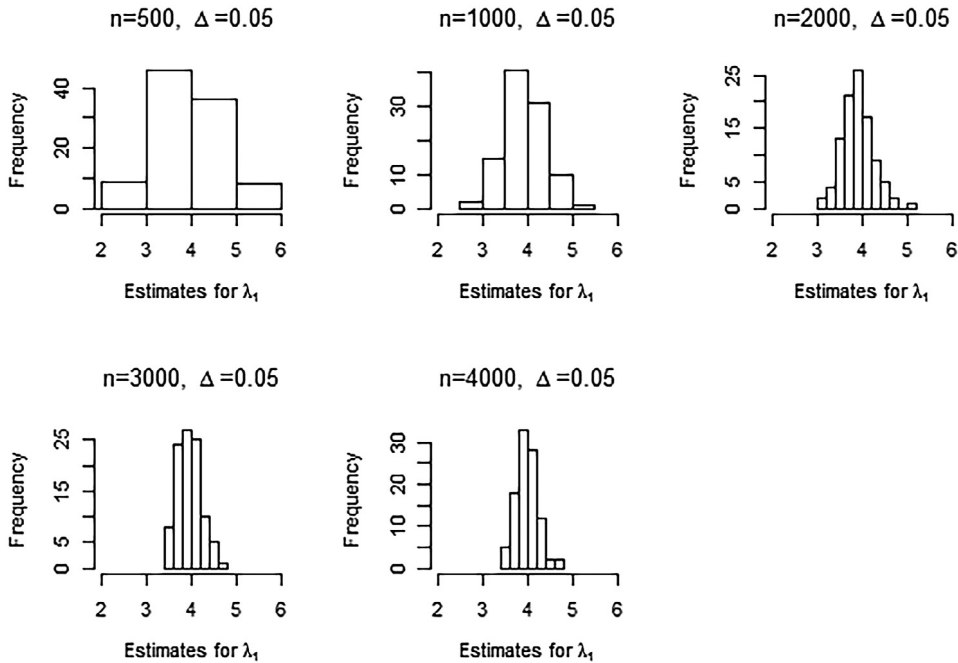


Fig. 3. Histograms of the obtained estimates for λ_1 , with n increasing from left to right.

4.1.2. Bootstrap confidence intervals

We note that this kind of simulation can also be used to construct bootstrap confidence intervals from a real data set. Suppose that a real data set is available with sample size n and interval length Δ , and that with the estimation algorithm the parameter estimates \hat{q}_1 , \hat{q}_2 , $\hat{\lambda}_1$ and $\hat{\lambda}_2$ have been obtained. Bootstrap confidence intervals for these parameters can then be computed by similar simulations as above in the following way. Choose $B > 0$ large, for example $B = 1000$, and simulate B new data sets with sample size n and interval length Δ using the parameter values \hat{q}_1 , \hat{q}_2 , $\hat{\lambda}_1$ and $\hat{\lambda}_2$. Compute

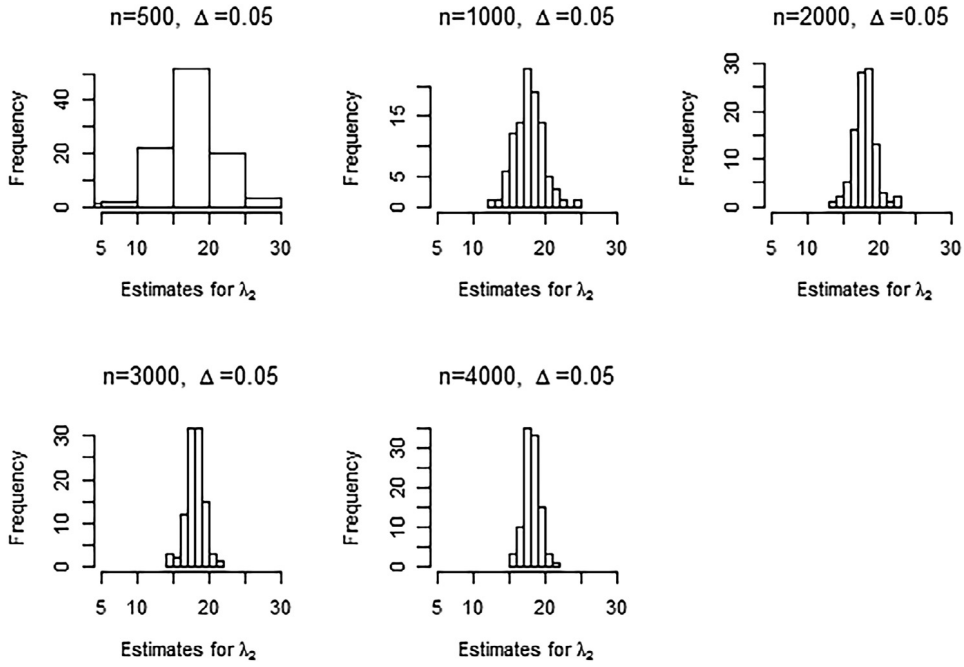


Fig. 4. Histograms of the obtained estimates for λ_2 , with n increasing from left to right.

for each simulated data set the corresponding parameter estimates using the estimation algorithm. This yields for each parameter, B bootstrap estimates. These, together with the original estimate, can then be used to construct the confidence interval. As our results above illustrate, the larger the sample size n , the more accurate the parameter estimates will be. Hence, the larger the sample size n , the smaller the confidence intervals will be.

4.2. Influence of the speed of the background process

We now fix $\Delta = 0.05$ and $n = 4000$ and consider the same parameter values as in the previous section, but speed up the background process by varying the values of q_1 and q_2 . We investigated the following four scenarios, where we kept the ratio $q_2/q_1 = 3$ fixed:

1. $q_1 = 0.1, q_2 = 0.3$;
2. $q_1 = 0.3, q_2 = 0.9$;
3. $q_1 = 0.5, q_2 = 1.5$;
4. $q_1 = 1.5, q_2 = 4.5$.

4.2.1. Results

Table 3 shows, for each setting (rows) and for each parameter (columns), the mean values of the 100 estimates together with the corresponding standard deviation between brackets. The mean estimates lie close to the true parameter values for each of the four settings. However, the standard deviations clearly increase as the speed of the background process becomes higher (q_1 and q_2 increase). The slower the background process, the easier it is for the algorithm to distinguish the two states and the more accurate the estimates become.

5. Estimation of the departure rate μ

Let the departure rate μ now be an unknown parameter which we also want to estimate. In this case the unknown parameter vector is $\theta = (\pi_i, q_{ij}, \lambda_i, \mu : i, j \in \{1, \dots, d\}, j \neq i)^T$. In many situations an appropriate assumption would be that an arriving individual does not leave the system in the same observation interval as in which it arrived. We will show that under this assumption the parameter μ can be estimated along with the other parameters of the system in a similar way to the one in Section 3.1. Again, let M_0^n be the observed vector and (A, X) be the missing data. For the purpose of this section we start with rewriting the loglikelihood function of the complete data by conditioning on X only, instead of on A and X as we did in (4). We have

$$\log \mathcal{L}(\theta | M_0^n, A, X) = \log \mathbb{P}_\theta(M_0^n, A | X) + \log \mathbb{P}_\theta(X). \quad (35)$$

Table 3

Mean of estimates of 100 data sets, with corresponding standard deviation between brackets, for four settings of the parameter values with $\Delta = 0.05$ and $n = 4000$.

	q_1	q_2	λ_1	λ_2
1.	0.108 (0.037)	0.321 (0.105)	3.967 (0.194)	17.953 (0.718)
2.	0.316 (0.076)	0.940 (0.228)	3.968 (0.236)	18.087 (1.044)
3.	0.528 (0.131)	1.555 (0.343)	3.934 (0.299)	18.036 (1.189)
4.	1.535 (0.396)	4.485 (1.150)	3.897 (0.472)	17.964 (1.738)

In Section 3.1, $\mathbb{P}_\theta(X)$ is computed, so the second term on the right hand side of (35) can be rewritten using (9). However, to find an expression for the first term, a few minor computations need to be made. We use the partition of the interval $(0, t_n]$ into the observation intervals $(t_{k-1}, t_k]$, $k = 1, \dots, n$, to obtain

$$\begin{aligned}
 & \log \mathbb{P}_\theta(M_0^n, A|X) \\
 &= \log \left(\prod_{k=1}^n \mathbb{P}_\theta(M_k, A_1^{[k]}, \dots, A_d^{[k]} | M_{k-1}, X) \right) \\
 &= \log \left(\prod_{k=1}^n \mathbb{P}_\theta(M_k | A_1^{[k]}, \dots, A_d^{[k]}, M_{k-1}, X) \cdot \mathbb{P}_\theta(A_1^{[k]} | X) \cdot \dots \cdot \mathbb{P}_\theta(A_d^{[k]} | X) \right) \\
 &= \sum_{k=1}^n \log \mathbb{P}_\theta(M_k | A_1^{[k]}, \dots, A_d^{[k]}, M_{k-1}, X) + \sum_{k=1}^n \sum_{i=1}^d \log \mathbb{P}_\theta(A_i^{[k]} | X).
 \end{aligned} \tag{36}$$

First, we note that the probability in the first term on the right hand side of (36) converts into a probability on the number of departures in the k th interval $(t_{k-1}, t_k]$. By the additional assumption, newly arrived individuals in $(t_{k-1}, t_k]$ cannot leave the system in this interval. Therefore, only individuals that are already present in the system at time t_{k-1} can leave the system in $(t_{k-1}, t_k]$. We thus have

$$\mathbb{P}_\theta(M_k | A_1^{[k]}, \dots, A_d^{[k]}, M_{k-1}, X) = \binom{M_{k-1}}{D_k} (1 - e^{-\mu\Delta})^{D_k} (e^{-\mu\Delta})^{M_{k-1} - D_k}, \quad 0 \leq D_k \leq M_{k-1}, \tag{37}$$

and zero otherwise. Here $D_k = M_{k-1} - M_k + \sum_{i=1}^d A_i^{[k]}$, the number of departures in the k th interval $(t_{k-1}, t_k]$. Next, we observe that for the second term on the right hand side of (36) it holds that

$$\mathbb{P}_\theta(A_i^{[k]} | X) = \frac{(\lambda_i Z_i^{[k]})^{A_i^{[k]}}}{(A_i^{[k]})!} e^{-\lambda_i Z_i^{[k]}}, \quad i = 1, \dots, d. \tag{38}$$

By combining (37) and (38), (36) becomes

$$\begin{aligned}
 & \log \mathbb{P}_\theta(M_0^n, A|X) \\
 &= \sum_{k=1}^n \left[\log \binom{M_{k-1}}{D_k} + D_k \log(1 - e^{-\mu\Delta}) - \mu\Delta(M_{k-1} - D_k) \right. \\
 & \quad \left. + \sum_{i=1}^d A_i^{[k]} \log(\lambda_i) + A_i^{[k]} \log(Z_i^{[k]}) - \log(A_i^{[k]}!) - \lambda_i Z_i^{[k]} \right], \quad 0 \leq D_k \leq M_{k-1}.
 \end{aligned} \tag{39}$$

Using (9), (35) and (39), we can rewrite the loglikelihood function of the complete data by aggregating all terms with π_i , all terms with q_{ij} , all terms with λ_i and all terms with μ . This yields

$$\begin{aligned}
 & \log \mathcal{L}(\theta | M_0^n, A, X) \\
 &= \sum_{i=1}^d \log(\pi_i) B_i + \sum_i \sum_{j \neq i} \sum_{k=1}^n \left(Y_{i,j}^{[k]} \log(q_{ij}) - q_{ij} Z_i^{[k]} \right) \\
 & \quad + \sum_{i=1}^d \sum_{k=1}^n \left(A_i^{[k]} \log(\lambda_i) - \lambda_i Z_i^{[k]} \right) + \sum_{i=1}^d \sum_{k=1}^n \left(A_i^{[k]} \log(Z_i^{[k]}) - \log(A_i^{[k]}!) \right) \\
 & \quad + \sum_{k=1}^n \left(\log \binom{M_{k-1}}{D_k} + D_k \log(1 - e^{-\mu\Delta}) - \mu\Delta(M_{k-1} - D_k) \right).
 \end{aligned} \tag{40}$$

It can be seen that (40) and (10) are precisely the same except for the additional last term on the right hand side of (40), which depends on μ and does not depend on the other parameters. Hence, we obtain the same parameter updates for π_i ,

Table 4

Mean of estimates of 100 data sets, with corresponding standard deviation between brackets, for $\Delta = 0.05$ and $\mu^0 = 0.1$ in the upper part and $\mu^0 = 1.7$ in the bottom part. True parameter values: $q_1 = 0.3, q_2 = 0.9, \lambda_1 = 4, \lambda_2 = 18, \mu = 0.6$.

$\mu^0 = 0.1$	q_1	q_2	λ_1	λ_2	μ
$n = 500$	0.446 (0.459)	1.200 (0.902)	3.927 (0.855)	17.380 (3.811)	0.620 (0.061)
$n = 1000$	0.346 (0.154)	1.020 (0.549)	3.940 (0.514)	17.850 (2.073)	0.615 (0.043)
$n = 2000$	0.321 (0.102)	0.949 (0.343)	3.922 (0.361)	18.016 (1.483)	0.614 (0.030)
$\mu^0 = 1.7$	q_1	q_2	λ_1	λ_2	μ
$n = 500$	0.442 (0.458)	1.196 (0.894)	3.945 (0.858)	17.417 (3.829)	0.621 (0.061)
$n = 1000$	0.342 (0.152)	1.018 (0.547)	3.954 (0.512)	17.897 (2.068)	0.616 (0.043)
$n = 2000$	0.318 (0.100)	0.945 (0.342)	3.935 (0.360)	18.051 (1.480)	0.615 (0.030)

q_{ij} and λ_i as in Section 3.1, while the additional parameter update for μ will be based on the last term on the right hand side of (40).

Let the parameter update for μ be denoted by $\tilde{\mu}$. Since not only the first four terms on the right hand side of (40), but also $\log \binom{M_{k-1}}{D_k}$ does not depend on μ , we see from (40) that

$$\begin{aligned} \tilde{\mu} &= \arg \max_{\mu} \mathbb{E}_{\theta'} [\log \mathcal{L}(\theta | M_0^n, A, X) | M_0^n = m_0^n] \\ &= \arg \max_{\mu} \mathbb{E}_{\theta'} \left[\sum_{k=1}^n \left(\log \binom{M_{k-1}}{D_k} + D_k \log(1 - e^{-\mu \Delta}) - \mu \Delta (M_{k-1} - D_k) \right) \middle| M_0^n = m_0^n \right] \\ &= \arg \max_{\mu} \left(\log(1 - e^{-\mu \Delta}) \sum_{k=1}^n \mathbb{E}_{\theta'} [D_k | M_0^n = m_0^n] - \mu \Delta \sum_{k=1}^n \mathbb{E}_{\theta'} [(M_{k-1} - D_k) | M_0^n = m_0^n] \right). \end{aligned} \quad (41)$$

Solving equation (41) for μ and using $D_k = M_{k-1} - M_k + \sum_{i=1}^d A_i^{[k]}$ yields the parameter update

$$\tilde{\mu} = \frac{1}{\Delta} \log \left(\frac{\sum_{k=1}^n m_{k-1}}{\sum_{k=1}^n m_k - \sum_{i=1}^d \sum_{k=1}^n \mathbb{E}_{\theta'} [A_i^{[k]} | M_0^n = m_0^n]} \right). \quad (42)$$

Note that this parameter update depends on $\sum_{k=1}^n \mathbb{E}_{\theta'} [A_i^{[k]} | M_0^n = m_0^n], i = 1, \dots, d$, which are already computed in updating the parameters λ_i , see (11). Hence, the parameter update of μ can be obtained in addition to the other parameter updates without much extra computational effort. The algorithm for finding maximum likelihood estimates for the new parameter vector $\theta = (\pi_i, q_{ij}, \lambda_i, \mu : i, j \in \{1, \dots, d\}, j \neq i)^T$ is the same as that in Section 3.6 with the single addition of computing $\tilde{\mu}$ according to (42) in step 5 of the algorithm.

To investigate the accuracy of the algorithm presented in this section, and for a proper comparison with the results in Section 4, we extended the simulation study of Section 4 by using the same data vectors and corresponding initial values, but adding the parameter update $\tilde{\mu}$ in (42) to the algorithm. Various initial values μ^0 for μ were taken, of which for two values, $\mu^0 = 0.1$ and $\mu^0 = 1.7$, the results are shown. Table 4 shows for each sample size (rows) and for each parameter (columns), the mean values of the 100 estimates together with the corresponding standard deviation between brackets, with $\mu^0 = 0.1$ and $\mu^0 = 1.7$ respectively. Our experiments show that the choice of μ^0 has a minor effect on the estimates, as each of the μ^0 values leads to very similar results. We see in Table 4 that the means of the estimates for μ lie close to the true parameter value and the corresponding standard deviations are small. This can be explained by the fact that the parameter μ is not affected by the modulation. We also see that the standard deviations decrease as n increases, which is also visible in Fig. 5. Here the histograms of the 100 estimates for μ are shown with increasing value of n , and $\mu^0 = 0.1$. The histograms show that the estimates for μ are concentrated around the true parameter value, and they have a bell-shape, indicating the distribution of the estimator being approximately normal. Finally, the values in Tables 2 and 4 illustrate that the estimates and corresponding standard deviations of the other parameters are barely influenced by the estimation of μ .

6. Discussion

In this paper we developed an algorithm for finding estimates of the parameters of an MMIS process. The proposed algorithm numerically approximates the maximum likelihood estimates of the parameters based on observations of the population size at equidistant time points. The algorithm is an iterative EM-type algorithm, in the spirit of the one that Okamura et al. (2009) developed. It is stressed that they did not consider departures but rather assumed discrete-time observations of the cumulative arrival process, and therefore some major adjustments to the steps in their approach were required to obtain our estimation algorithm.

We have investigated the accuracy of the proposed algorithm by means of an extensive simulation study. The results showed that the estimates are concentrated around the true parameter values and become more accurate as the sample

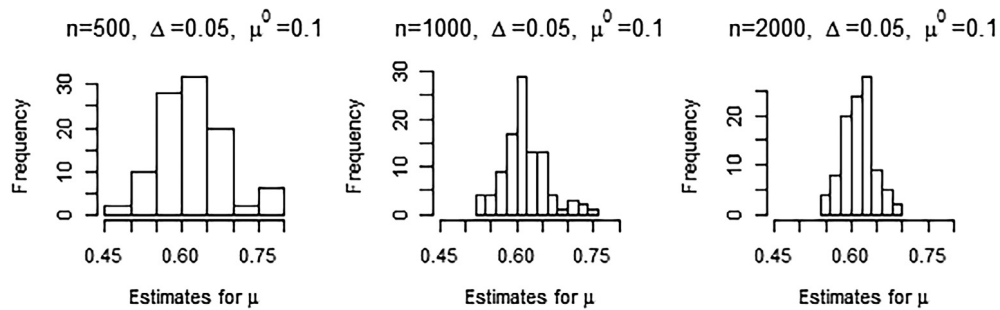


Fig. 5. Histograms of the obtained estimates for μ , with n increasing from left to right, and $\mu^0 = 0.1$.

size increases. In addition, the results indicated that for sufficiently large n the distributions of the estimators become approximately normal. Furthermore, the estimates got more accurate when the background process becomes slower, as it is easier for the algorithm to detect the different states. However, to retain a high accuracy, the sample size n , and hence $T = n\Delta$, must be large such that the background process jumps often enough during the total observation time T . Moreover, for larger d , n must also be larger to retain a similar accuracy.

The run-time of the algorithm depends on various parameters. In the first place, the run-time of a single iteration in the algorithm is linear in n . However, as n increases the algorithm is likely to converge more quickly, implying that the number of iterations required will decrease in n . In addition, the run-time of the algorithm increases in d . The run-time of one iteration in the algorithm is mainly determined by step 3 in Section 3.6, and from Section 3.5 we know that the computational effort of this step increases in d . However, as we mentioned before, Markov-modulation is typically used to model situations for which d is small (typically 2). We finally mention that the number of iterations needed for convergence of the algorithm, and hence the run-time, tends to be large if the length of the observation intervals, Δ , is large, or if the initial parameter values are far away from the true parameter values.

There are many interesting directions for future research based on our results. We note that the estimation algorithm that we developed is built on the assumption that we know the number of states d . The choice of the dimension d from the data is a model selection problem which is outside the scope of the present paper, but could be explored in a follow-up project. Another research theme could relate to generalizing the sojourn time distribution, where non-parametric estimation could be explored. One could also consider more general inter-arrival times, since for some applications exponential inter-arrival times may not be a suitable fit.

Acknowledgment

MM was supported by the NWO Gravitation program NETWORKS, grant 024002003.

References

- Anderson, D., Blom, J., Mandjes, M., Thorsdottir, H., de Turck, K., 2016. A functional central limit theorem for a Markov-modulated infinite-server queue. *Methodol. Comput. Appl. Probab.* 18, 153–168. <http://dx.doi.org/10.1007/s11009-014-9405-8>.
- Anderson, D., Kurtz, T., 2015. Stochastic Analysis of Biochemical Systems, Vol. 1.2. In: Stochastics in Biological Systems, Springer International Publishing, <http://dx.doi.org/10.1007/978-3-319-16895-1>.
- Asmussen, S., Nerman, O., Olsson, M., 1996. Fitting phase-type distributions via the EM algorithm. *Scand. J. Stat.* 23 (4), 419–441.
- Bingham, N., Pitts, S., 1999. Non-parametric estimation for the M/G/∞ queue. *Ann. Inst. Statist. Math.* 51 (1), 71–97. <http://dx.doi.org/10.1023/A:1003831118254>.
- Blom, J., Kella, O., Mandjes, M., Thorsdottir, H., 2013. Markov-modulated infinite-server queues with general service times. *Queueing Syst.* 76, 403–424. <http://dx.doi.org/10.1007/s11134-013-9368-4>.
- Breuer, L., Kume, A., 2010. An EM algorithm for Markovian arrival processes observed at discrete times. In: International GI/ITG Conference on Measurement, Modeling, and Evaluation of Computing Systems and Dependability and Fault Tolerance, Vol. 5987. Springer, pp. 242–258. http://dx.doi.org/10.1007/978-3-642-12104-3_19.
- Dempster, A., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 39 (1), 1–38.
- Ephraim, Y., Roberts, W., 2009. An EM algorithm for Markov modulated Markov processes. *IEEE Trans. Signal Process.* 57 (2), 463–470. <http://dx.doi.org/10.1109/TSP.2008.2007919>.
- Hautphenne, S., Fackrell, M., 2014. An EM algorithm for the model fitting of Markovian binary trees. *Comput. Statist. Data Anal.* 70 (20), 19–34. <http://dx.doi.org/10.1016/j.csda.2013.08.015>.
- Neuts, M., 1981. Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach. Johns Hopkins University Press, Baltimore, <http://dx.doi.org/10.1002/net.3230130219>.
- O’Cinneide, C., Purdue, P., 1986. The M/M/∞ queue in a random environment. *J. Appl. Probab.* 23 (1), 175–184. <http://dx.doi.org/10.2307/3214126>.
- Okamura, H., Dohi, T., 2009. Faster maximum likelihood estimation algorithms for Markovian arrival processes. In: IEEE, Sixth International Conference on the Quantitative Evaluation of Systems, pp. 73–82. <http://dx.doi.org/10.1109/QEST.2009.28>.
- Okamura, H., Dohi, T., Trivedi, K., 2009. Markovian arrival process parameter estimation with group data. *IEEE/ACM Trans. Netw.* 17 (4), 1326–1339. <http://dx.doi.org/10.1109/TNET.2008.2008750>.

- Pang, G., Zhou, Y., 2017. Two-parameter process limits for an infinite-server queue with arrival dependent service times. *Stochastic Process. Appl.* 127 (5), 1375–1416. <http://dx.doi.org/10.1016/j.spa.2016.08.003>.
- Pickands, J., Stine, R., 1997. Estimation for an $M/G/\infty$ queue with incomplete information. *Biometrika* 84 (2), 295–308.
- Rabiner, L., Juang, B., 1986. An introduction to hidden Markov models. *IEEE ASSP Mag.* 3 (1), 4–16. <http://dx.doi.org/10.1109/MASSP.1986.1165342>.
- Roberts, W., Ephraim, Y., Dieguez, E., 2006. On Rydén's EM algorithm for estimating MMPPs. *IEEE Signal Process. Lett.* 13 (6), 373–376. <http://dx.doi.org/10.1109/LSP.2006.871709>.
- Rydén, T., 1996. An EM algorithm for estimation in Markov modulated poisson processes. *Comput. Statist. Data Anal.* 21, 431–447. [http://dx.doi.org/10.1016/0167-9473\(95\)00025-9](http://dx.doi.org/10.1016/0167-9473(95)00025-9).
- Schwabe, A., Rybakova, K., Bruggeman, F., 2012. Transcription stochasticity of complex gene regulation models. *Bioph. J.* 103 (6), 1152–1161. <http://dx.doi.org/10.1016/j.bpj.2012.07.011>.
- Welch, L., 2003. Hidden Markov models and the Baum–Welch algorithm. *IEEE Inf. Theory Soc. Newsl.* 53 (4), 1–13.