Noname manuscript No.
(will be inserted by the editor)

Tracking multiple moving objects in images using Markov Chain Monte Carlo

Lan Jiang · Sumeetpal S. Singh

Received: date / Accepted: date

Abstract A new Bayesian state and parameter learning algorithm for multiple target tracking (MTT) models with image observations is proposed. Specifically, a Markov chain Monte Carlo algorithm is designed to sample from the posterior distribution of the unknown time-varying number of targets, their birth, death times and states as well as the model parameters, which constitutes the complete solution to the specific tracking problem we consider. The conventional approach is to pre-process the images to extract point observations and then perform tracking, i.e. infer the target trajectories. We model the image generation process directly to avoid any potential loss of information when extracting point observations using a pre-processing step that is decoupled from the inference algorithm. Numerical examples show that our algorithm has improved tracking performance over commonly used techniques, for both synthetic examples and real florescent microscopy data, especially in the case of dim targets with overlapping illuminated regions.

1 Introduction

The multiple target tracking (MTT) problem is to infer the states or tracks of multiple moving objects from noisy measurements. The problem is very challenging because the number of targets is unknown and changes over time as it is a birth-death process. Other compounding factors include the non-linearity of both the target's motion and observation models. In applications such as radar (or sonar) tracking (Bar-Shalom and Fortmann, 1988) and Fluorescence Microscopy (Weimann et al, 2013), the measurements (or observations) are images. For example, a pixel's illumination intensity is a measure of the energy captured from nearby targets (e.g. through a reflected radar signal) and background noise. These images are usually pre-processed prior to actual tracking to extract point measurements. Each point is a spatial coordinate and is assumed to be either a noisy measurement of a target's state or spuriously generated. The latter is an artefact of the method that extracts point measurements. Converting images to point measurements is advantageous because it yields a simpler observation model and also quite significantly simplifies the design of tracking algorithms (Bar-Shalom and Fortmann, 1988; Mahler, 2007), e.g. (Weimann et al, 2013) connects the point measurements using a nearest neighbour method to form target trajectories. However, the pre-processing step can introduce information loss in the low signal-to-noise (SNR) regime, which can be near complete as the targets become more closely spaced and background noise intensifies. In low SNR it can be difficult to isolate bright regions in the image (see a frame of real data in Fig. 8) whose centres would be the candidate point measurements, and then attribute them to distinct targets. Thus, MTT algorithms that track using the images directly can be preferable (called track-before-detect (TBD) techniques) and a selection of works is (Streit et al, 2002; Rutten et al, 2005; Boers and Driessen, 2004; Punithakumar et al, 2005; Davey et al, 2007; Vo et al, 2010; Papi and Kim, 2015). These assume known model parameters but differ in how tracking is achieved (Bayesian, maximum likelihood or otherwise) and the specific assumptions imposed on the image model. For example, Punithakumar et al (2005); Vo et al (2010);

This work was supported by the Engineering and Physical Sciences Research Council [grant numbers EP/G037590/1, EP/K020153/1.]

Dept. of Engineering, University of Cambridge E-mail: sss40@cam.ac.uk

Papi and Kim (2015) use specific but different Poisson approximations for the MTT posterior which is then approximated using a Particle Filter.

Given images recorded over a length of time, say from time 1 to n, our aim is to jointly infer the target tracks and the MTT model parameters. We adopt a Bayesian approach and one of our main contributions is the design of a new Markov chain Monte Carlo (MCMC) algorithm for an image measurement model that can jointly track and calibrate the model. The MCMC algorithm is a trans-dimensional sampler that combines Particle Markov Chain Monte Carlo (PMCMC) (Andrieu et al, 2010) steps to sample from the *exact* MTT posterior distribution for (entire) target tracks and model parameters. This is in contrast to numerous MTT techniques that use specific variational approximations, e.g. spatial Poisson in the cited references above, of the MTT posterior to simplify inference. We infer entire tracks, as opposed to point estimates of target locations at each time as in Vo et al (2010), because these are then needed to infer the aggregate diffusion characteristics of the tracked molecules (Weimann et al, 2013). Model calibration is needed because in real fluorescence microscopy data, the molecules to be tracked bleach over time and the noise characteristics of the acquired images also drift. These changes can be captured by time varying image model parameters, which are indeed unknown to the analyst, as are other parameters such as those describing the molecule motion model. To the best of our knowledge, our trans-dimensional MCMC tracker for image observations addresses this practical tracking problem in greater generality without major limiting assumptions like well separated molecules with non-overlapping illumination regions (Vo et al, 2010) or aforementioned principled simplifications using a variational approximation of the MTT posterior (Papi and Kim, 2015). In numerical examples we demonstrate the superior performance of our method over the TBD technique of Vo et al (2010) for closely spaced targets. We also apply it to real florescence microscopy data and show it outperforms a method currently used by biologists (Weimann et al, 2013) which pre-processes the images to extract point observations. Such comparisons, which are absent in the literature, highlight the gain in performance by targeting the exact MTT posterior and avoiding simplifications like disallowing overlaps.

We do not advocate that our MCMC method should replace techniques that extract point observations, those that use variational approximations to simplify the posterior (Mahler, 2007; Vo et al, 2010; Papi and Kim, 2015; Schlangen et al, 2016), or those that do not extract point observations but are optimised for non-overlapping targets (Vo et al, 2010). Our MCMC technique should be viewed as a complement to these other techniques. It could be applied in an online tracking scenario by processing a window of data at a time or as a post-processing tool to refine the trajectories identified by any other online algorithm (Jiang et al, 2014). This is similar to the role MCMC plays in the related field of Particle Filtering, which is an online estimation method, where MCMC is used to refine the online estimates (Del Moral et al, 2006; Doucet and Johansen, 2009; Kantas et al, 2015).

MTT methods have very recently been applied to tracking in fluorescence microscopy data in Rezatofighi et al (2015); Schlangen et al (2016). Both works extract point measurements and use a specific but different variational approximation of the MTT posterior distribution which is then approximated in Rezatofighi et al (2015) using Gaussian quadrature and in Schlangen et al (2016) using a Particle filter. The obtained point estimates of molecule locations are then post-processed to obtain tracks. Model calibration is done in a limited sense in Rezatofighi et al (2015), i.e. only some parameters are learned from the data while others are assumed known.

For point measurement models in general, there is a growing literature on using MCMC for tracking as it is recognised that sampling the true MTT posterior, although challenging, is feasible in offline applications and can serve as a track refinement tool in the online setting (Oh et al, 2009; Vu et al, 2014). Oh et al (2009); Kokkala and Sarkka (2015) assume the underlying state-space and observation model is linear and Gaussian, Vu et al (2014); Jiang et al (2015) consider the non-linear and non-Gaussian setting while (Duckworth, 2012; Jiang et al, 2015; Kokkala and Sarkka, 2015) simultaneously estimate the model parameters. Although some of the above works incorporate parameter estimation, it is a topic in MTT that has only recently gained attention, see Singh et al (2011); Yıldırım et al (2014).

The remainder of the paper is organised as follows. Section 2 describes the MTT model and presents the framework for joint state and parameter learning algorithm. In Section 3, we present details of our novel MCMC kernel for detecting and maintaining tracks, which constitutes the core part of our tracking algorithm. (More detailed derivations are given in the Appendix.) Section 4 presents numerical results for both synthetic and real florescent microscopy data.

2 Multiple target tracking model

2.1 The single target model

We commence with a description of the image based tracking problem assuming a single target and then enlarge the model for the multi-target case. Let the discrete time Markov process $\{X_t\}_{t\geq 1} = \{X_1, X_2, \ldots\}$ represent the state values of a single evolving target. In this work it is assumed that $X_t = (X_t(1), \ldots, X_t(5)) \in \mathbb{R}^5$ where $X_t(i)$ denotes its ith component. $X_t(1)$ is the target's *illumination intensity* or amplitude (to be discussed in detail next), $(X_t(2), X_t(3))$ is the spatial coordinate of the target and $(X_t(4), X_t(5))$ are the corresponding spatial velocities. Frequent reference will be made to the intensity (amplitude), spatial coordinate and spatial velocity components of a target state X_t. As such we will denote these components by $A_t = X_t(1)$, $S_t = (S_t(1), S_t(2)) = (X_t(2), X_t(3))$ and $V_t = (V_t(1), V_t(2)) = (X_t(4), X_t(5))$. X_t is a time-homogeneous Markov process,

$$X_1 \sim \mu_\psi(\cdot), \quad X_t | X_{1:t-1} = x_{1:t-1} \sim f_\psi(\cdot | x_{t-1}) \tag{1}$$

where μ_ψ and f_ψ are, respectively, the initial and state transition probability density function (pdf), parameterised by the common real valued vector $\psi \in \Psi \subset \mathbb{R}^{d_\psi}$. (As a rule, a random variable (r.v.) is denoted by a capital letter and its realisation by small case.) In our numerical example we use linear Gaussian state dynamics, $\mu_\psi(x) = \mathcal{N}(x; \mu_b, \Sigma_b)$, $f_\psi(x'|x) = \mathcal{N}(x'; Fx, \Sigma)$, where $\mathcal{N}(\cdot; Fx, \Sigma)$ denotes the Gaussian pdf with mean Fx and covariance matrix Σ. Typically the matrix F is known and thus $\psi = (\mu_b, \Sigma_b, \Sigma)$. ψ and other parameters of the model detailed below are to be learnt from the data.

A two dimensional image measurement model is assumed with m pixels in total. Let

$$Y_t = (Y_{t,1}, \ldots, Y_{t,m}),$$

denote the observed image at time t where $Y_{t,i}$ is the value (illumination intensity) of pixel i. $Y_{t,i}$ is defined as

$$Y_{t,i} = h_i(X_t) + E_{t,i}, \tag{2}$$

where $E_{t,i}$ is the observation noise of pixel i at time t and $h_i(X_t)$ is the illumination of pixel i by a single target with state X_t. As in Vo et al (2010), for $x = (a, s, v) \in \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2$ where a is the intensity, $s = (s(1), s(2))$ the spatial coordinate and $v = (v(1), v(2))$ the spatial velocity, $h_i(x)$ is the point spread function

$$\begin{aligned} h_i(x) &= \mathbb{I}_{L(s)}(i)\ \frac{a\Delta_1\Delta_2}{2\pi\sigma_h^2} \\ &\quad \times \exp\{-\frac{(\Delta_1 r - s(1))^2 + (\Delta_2 c - s(2))^2}{2\sigma_h^2}\} \\ &=:\ a\ \bar{h}_i(s) \end{aligned} \tag{3}$$

where $\mathbb{I}_A$ is the indicator function, (r, c) denotes the row and column number of pixel i, Δ_1 and Δ_2 are constants that map pixel indices to spatial coordinates and σ_h is the blurring parameter. It is assumed that the spatial coordinate of the pixel with index corresponding to row and column number $(0, 0)$ is the origin of $\mathbb{R}^2$. As in Vo et al (2010), we also assume for each state value $x = (a, s, v)$ there is a square truncation region $L(s)$ where $h_i(x) = 0$ if $i \notin L(s)$. Specifically, $L(s)$ is the set of $l \times l$ pixels, l an odd integer, whose centre pixel has spatial coordinate closest to s. Henceforth we assume $\Delta_1 = \Delta_2 = \Delta$.

For later use, (3) defines the function $\bar{h}_i(s)$ to be $h_i(x)/a$ when $x = (a, s, v)$. In addition, extend the domain of the truncation region L and point spread function $\bar{h}_i$ to included pixel indices $j \in \{1, \ldots, m\}$. That is, let (r', c') be the row and column number of pixel j and define

$$L(j) = L(s), \quad \bar{h}_i(j) = \bar{h}_i(s) \quad \text{where } s = (r'\Delta, c'\Delta). \tag{4}$$

Equivalently $L(j)$ is the square of $l \times l$ pixels centered at pixel j.

The pixel noise is assumed to be Gaussian with mean value b_t, representing the background intensity, and variance $\sigma_{r,t}^2$, both time varying but common across pixels, i.e.

$$E_{t,i} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\cdot | b_t, \sigma_{r,t}^2), \quad i = 1, \ldots, m,\ t = 1, \ldots, n.$$

Thus, the conditional pdf of the observed image at time t due to a single target with state X_t is

$$g_t(y_t | x_t) = \prod_{i=1}^{m} \mathcal{N}(y_{t,i}; h_i(x_t) + b_t, \sigma_{r,t}^2).$$

where subscript t of g_t indicates the observation model is time-inhomogeneous. Given n images, all the model parameters, $(\psi, b_1, \sigma_{r,1}, \ldots, b_n, \sigma_{r,n})$, will be estimated. See Fig. 8 for a real captured image and the corresponding synthesised image based on estimated tracks and model parameters.

2.2 The model for multiple targets

In this section we partially adopt the mathematical formulation in Jiang et al (2015) for the MTT model. (Note though that the model in Jiang et al (2015) is for point-observations and not for images as in our case.) In an MTT model, the MTT state at time t is the concatenation of all individual target states at t:

$$\mathbf{X}_t = \left(X_{t,1}, X_{t,2}, \ldots, X_{t,K_t^x}\right) \tag{5}$$

where each sub-vector $X_{t,i}$ is the state (as in (1)) of an individual target. The number of targets K_t^x under surveillance changes over time due to the death of existing targets and the birth of new targets. Independently of the other targets, a target survives to the next time with survival probability p_s and its state evolves according to the transition density f_ψ, otherwise it 'dies'. In addition to the surviving targets, the model introduces K_t^b new targets at each time $t \in \{1, \ldots, n\}$ where K_t^b is distributed according to the Poisson probability mass function with parameter λ_b. The states of the new targets are then sampled independently from μ_ψ, which is the common pdf for the initial states of all new targets. The states of the new born targets and surviving targets from time t make up $\mathbf{X}_{t+1}$.

To describe the evolution of $\mathbf{X}_t$ due to survivals and births, a series of random variables are defined below. Let K_t^x and K_t^b denote the number of targets and new births at time t respectively. We assume that at time $t = 1$ there are only new born targets, i.e. no surviving targets from the past and thus $K_1^x = K_1^b$. For $t > 1$ and $i = 1, \ldots, K_{t-1}^x$, let

$$C_t(i) = \begin{cases} 1 & \text{if the } i\text{th target from } t-1 \text{ survives,} \\ 0 & \text{if it does not survive to time } t. \end{cases} \tag{6}$$

C_t is the $K_{t-1}^x \times 1$ binary vector where 1's indicate survivals and 0's indicate deaths of targets from time $t-1$. Let K_t^s denote the number of surviving targets at time t, thus

$$K_t^s = \sum_{i=1}^{K_{t-1}^x} C_t(i). \tag{7}$$

The K_t^s surviving targets from time $t-1$ evolve to become the *first* K_t^s targets in $\mathbf{X}_t$. Specifically, define the $K_t^s \times 1$ ancestor vector I_t, $t > 1$, as

$$I_t(i) = \min\left\{k : \sum_{j=1}^{k} C_t(j) = i\right\}, \quad i = 1, \ldots, K_t^s, \tag{8}$$

so that $X_{t-1,I_t(i)}$ evolves to $X_{t,i}$ for $i = 1, \ldots, K_t^s$. In addition to the surviving targets $(X_{t,1}, \ldots, X_{t,K_t^s})$, we have K_t^b newly born targets denoted by $X_{t,K_t^s+1}, \ldots, X_{t,K_t^x}$. The state $\mathbf{X}_t$ is formed of the new born targets together with the surviving targets, and thus $K_t^x = K_t^b + K_t^s$. An ordering rule is adopted for the new born targets to avoid labelling ambiguity. Specifically, the new born targets at each time t are labelled in ascending order of their first component value. Let $Z_1 = K_1^b$ and

$$Z_t = \left(C_t, K_t^b\right), \quad t > 1, \tag{9}$$

which is the discrete component of the MTT state at time t. Figure 1 illustrates all the MTT random variables.

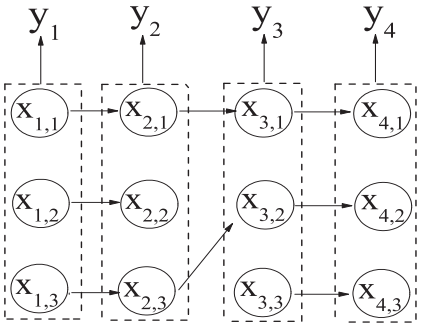

Fig. 1: A realisation from the MTT model. States of a target are connected with arrows and all targets at time t contribute to image y_t.
MTT random variables:
Time $t = 1$: No prior targets ($C_1 = (), K_1^s = 0, I_1 = ()$), three targets are born ($K_1^x = K_1^b = 3$) with states $X_{1,1}, X_{1,2}, X_{1,3}$.
Time $t = 2$: All targets $X_{1,1}, X_{1,2}, X_{1,3}$ survive to become $X_{2,1}, X_{2,2}, X_{2,3}$. Thus $C_2 = (1,1,1)$, $K_2^s = 3$, $I_2 = (1,2,3)$. No new born targets, $K_2^b = 0$, $K_2^x = K_2^s + K_2^b = 3$.
Time $t = 3$: Targets $X_{2,1}$ and $X_{2,3}$ survive to become $X_{3,1}$ and $X_{3,2}$ respectively while $X_{2,2}$ dies. Thus $C_3 = (1,0,1)$, $K_3^s = 2$, $I_3 = (1,3)$. One new born target, $K_3^b = 1$, denoted $X_{3,3}$. $K_3^x = K_3^s + K_3^b = 3$.
Time $t = 4$: All targets survive, no new born, same as time $t = 2$.
MTT variables in the parameterisation of Sec. 2.4:
$t_b^1 = 1$, $\mathbf{X}^1 = (X_{1,1}, X_{2,1}, X_{3,1}, X_{4,1})$; $t_b^2 = 1$, $\mathbf{X}^2 = (X_{1,2}, X_{2,2})$; $t_b^3 = 1$, $\mathbf{X}^3 = (X_{1,3}, X_{2,3}, X_{3,2}, X_{4,2})$; $t_b^4 = 3$, $\mathbf{X}^4 = (X_{3,3}, X_{4,3})$.

2.3 The law of MTT model

The image observation $Y_t = (Y_{t,1}, \ldots, Y_{t,m})$ generated by multiple targets at time t is the superposition of the contributions of all targets at time t, the background intensity and noise, i.e.

$$Y_{t,i} = h_i(\mathbf{X}_t) + E_{t,i}, \quad h_i(\mathbf{X}_t) = \sum_{k=1}^{K_t^x} h_i(X_{t,k}), \tag{10}$$

where $h_i(X_{t,k})$ is the contribution of the k-th target at time t to the illumination of pixel i (see (3)). The MTT

observation model is

$$g_t(y_t|\mathbf{x}_t) = \prod_{i=1}^{m} \mathcal{N}(y_{t,i}; h_i(\mathbf{x}_t) + b_t, \sigma^2_{r,t}). \tag{11}$$

Given the vector of the MTT model parameters

$$\theta = (\psi, p_s, \lambda_b, b_1, \sigma^2_{r,1}, \ldots, b_n, \sigma^2_{r,n}) \tag{12}$$

(recall ψ describes the motion model (1), p_s is the survival probability, λ_b the birth rate of new targets, while the remaining parameters describe the observed image pixel intensities (11)) the law of the MTT model can be expressed with the joint density of $(Z_{1:n}, \mathbf{X}_{1:n}, Y_{1:n})$,

$$p_\theta(y_{1:n}|\mathbf{x}_{1:n}, z_{1:n}) p_\theta(\mathbf{x}_{1:n}|z_{1:n}) p_\theta(z_{1:n}),$$

where $a_{i:j}$, $i \le j$, denotes the sequence $a_i, a_{i+1} \ldots a_j$,

$$p_\theta(y_{1:n}|\mathbf{x}_{1:n}, z_{1:n}) = \prod_{t=1}^{n} g_\theta(y_t|\mathbf{x}_t), \tag{13}$$

$$p_\theta(z_{1:n}) = \mathcal{P}(k^b_1; \lambda_b) \prod_{t=2}^{n} p_s^{k^s_t} (1-p_s)^{k^x_{t-1} - k^s_t} \mathcal{P}(k^b_t; \lambda_b), \tag{14}$$

$$p_\theta(\mathbf{x}_{1:n}|z_{1:n}) = \prod_{t=1}^{n} \Bigg[\prod_{j=1}^{k^s_t} f_\psi(x_{t,j}|x_{t-1,i_t(j)}) \; k^b_t! \mathbb{I}_{\mathcal{O}}(x_{t,k^s_t+1:k^x_t}) \prod_{j=k^s_t+1}^{k^x_t} \mu_\psi(x_{t,j}) \Bigg]. \tag{15}$$

$\mathcal{P}(k;\lambda) = \exp(-\lambda)\lambda^k/k!$, the Poisson probability mass function and $g_\theta(y_t|\mathbf{x}_t)$ in (13) is only dependent on components $(b_t, \sigma_{r,t})$ of θ and is precisely g_t of (11). In (14), k^s_t is the number of surviving targets from time $t-1$ defined in (7) and k^x_t is the total number of (alive) targets at time t (5). Variable $i_t(j)$ in (15) denotes the ancestor of the time-t target j (8). $\mathbb{I}_{\mathcal{O}}$ is the indicator function of the particular ordering rule $\mathcal{O}$ for the new born targets,

$$\mathbb{I}_{\mathcal{O}}(x_{t,k^s_t+1:k^x_t}) = \begin{cases} 1 & \text{if } x_{t,k^s_t+1}(1) < \cdots < x_{t,k^x_t}(1), \\ 0 & \text{else.} \end{cases}$$

The ordering rule $\mathcal{O}$ will allow us to deterministically assign a unique label to each target track; see Section 2.4. Finally, the marginal likelihood of the data $y_{1:n}$ is given by

$$p_\theta(y_{1:n}) = \sum_{z_{1:n}} \int p_\theta(z_{1:n}, \mathbf{x}_{1:n}, y_{1:n}) \mathrm{d}\mathbf{x}_{1:n}. \tag{16}$$

2.4 An equivalent representation of $(Z_{1:n}, \mathbf{X}_{1:n})$

This section introduces an equivalent parameterization for the MTT problem. Essentially, we define a new set of random variables which are an alternative to those defined in Section 2.3 without any loss of information. The idea here is to introduce notation that explicitly isolates the state trajectories of individual targets. We need to use both the parameterization in this section and that in Section 2.3 to adequately describe the MCMC moves and proposal distributions in Section 3.

Let $K = \sum_{t=1}^{n} k^b_t$ denote the total number of targets that have appeared from time 1 to n. Each target appearing in this time span can be assigned a distinct label or index $k \in \{1, \ldots, K\}$ with the convention that targets born earlier are given a smaller label than those born at a later time and targets born at the same time are sorted by the ordering rule $\mathcal{O}$.

Consider a target assigned labelled $k \in \{1, \ldots, K\}$, let its birth time be t^k_b, death time be t^k_d and its life span be $l^k = t^k_d - t^k_b$. (Note $t^k_d - 1$ is the final time of its existence.) The entire continuous state trajectory of this target can be extracted from the MTT state sequence $(\mathbf{X}_{t^k_b}, \ldots, \mathbf{X}_{t^k_d - 1})$ and denote it by

$$\mathbf{X}^k = (X^k_0, \ldots, X^k_{l^k-1})$$

where X^k_{i-1} is the i-th state of target k. Note that $\mathbf{X}^k$ is a Markov process with initial and state transition densities μ_ψ and f_ψ respectively. It is straightforward to extract $\{(k, t^k_b, \mathbf{X}^k)\}_{k=1}^K$ from $(Z_{1:n}, \mathbf{X}_{1:n})$ as illustrated in Figure 1. The main point is that we can use one of the two equivalent descriptions for latent variables of the MTT model, i.e.

$$(Z_{1:n}, \mathbf{X}_{1:n}) \Leftrightarrow \{(k, t^k_b, \mathbf{X}^k)\}_{k=1}^K. \tag{17}$$

On the other hand, $(Z_{1:n}, \mathbf{X}_{1:n})$ can be obtained from $\{(k, t^k_b, \mathbf{X}^k)\}_{k=1}^K$ since the underlying transformation is a bijection. (Again see Figure 1 for an example.)

2.5 Bayesian tracking and parameter estimation for MTT

The inference task is to estimate the discrete variables $Z_{1:n}$, target states $\mathbf{X}_{1:n}$ and the MTT parameter θ given the observations $y_{1:n}$. Regarding θ as a random variable taking values in Θ with a prior density $\eta(\theta)$, the goal is to obtain Monte Carlo samples from

$$p(z_{1:n}, \mathbf{x}_{1:n}, \theta|y_{1:n}) \propto \eta(\theta) p_\theta(z_{1:n}, \mathbf{x}_{1:n}, y_{1:n}). \tag{18}$$

We achieve this by iteratively performing the MCMC sweeps given in Algorithm 1. A single call of Algorithm

1 will transform a current sample $(\theta, Z_{1:n}, \mathbf{X}_{1:n})$ from the posterior to a new sample $(\theta', Z'_{1:n}, \mathbf{X}'_{1:n})$. The entire sequence of samples yielded by the repeated calls to Algorithm 1 will constitute the desired set of Monte Carlo samples from (18). We need though to discard an initial sequence of this set so that the remaining samples retained are correctly distributed. Section 3

Algorithm 1: MCMC for state and parameter learning

Input: Current sample $(\theta, z_{1:n}, \mathbf{x}_{1:n})$, data $y_{1:n}$, number of inner loops n_1.
Output: Updated sample $(\theta', z'_{1:n}, \mathbf{x}'_{1:n})$.

```
for j = 1 : n_1 do
    Update (z_{1:n}, x_{1:n}) by invoking Algorithm 2.
Isolate target trajectories (see (17)) {(t_b^k, x^k)}_{k=1}^K.
for k = 1 : K do
    Update x^k using the CSMC (conditional SMC) smoother with other trajectories (≠ k) and θ fixed.
Call the updated sample (z'_{1:n}, x'_{1:n}).
Conditioned on (z'_{1:n}, x'_{1:n}), update θ to θ' using the Gibbs move for each component.
```

is dedicated to the exposition of the first loop of Algorithm 1 while the second loop is more easily described. The principal aim of the second loop is to resample the continuous state trajectory of each target using the conditional SMC (CSMC) sampler of Andrieu et al (2010) but using the implementation in Whiteley (2010) as detailed in Appendix C. This step enhances our MCMC algorithm's efficiency. The CSMC step is done by first explicitly isolating the state trajectories of individual targets as in Section 2.4 and then updating the targets' trajectories in turn using the CSMC sampler. When conjugate priors are available for the components of θ, as in our numerical examples, it is possible to sample $p(\theta|z_{1:n}, \mathbf{x}_{1:n}, y_{1:n}) \propto p(\theta)p_\theta(z_{1:n}, \mathbf{x}_{1:n}, y_{1:n})$ exactly in the final step; details in Appendix D. Otherwise, one can run a Metropolis-Hastings (MH) algorithm to sample from this pdf. When the MTT parameters are known, the final step of Algorithm 1 is to be omitted and we refer to the resulting algorithm as the *MCMC tracker*.

3 MCMC moves

In this section, we present the MCMC moves to sample $(Z_{1:n}, \mathbf{X}_{1:n})$ for the first loop in Algorithm 1. Notice that the dimension of $\mathbf{X}_{1:n}$, which is proportional to $\sum_{t=1}^n K_t^x$, changes with $Z_{1:n}$. Therefore, the posterior distribution $p_\theta(z_{1:n}, \mathbf{x}_{1:n}|y_{1:n})$ is said to be *trans-dimensional*.

3.1 A brief on trans-dimensional MCMC

A general method for sampling from a trans-dimensional distribution is the reversible jump MCMC (RJMCMC) algorithm of Green (1995). We briefly describe RJMCMC for a (general) distribution $\pi(m, x_m)$ where m is a discrete variable e.g. $m \in \{1, 2 \ldots\}$ known as the model index and $x_m \in \mathbb{R}^{d_m}$. Note though that in general $m' \neq m$ does not imply $d_{m'} \neq d_m$. We will then connect this generic RJMCMC description with the specific moves of the MCMC procedure for tracking, described in Section 3.2 onwards, to aid the latter's exposition.

For each (m, x_m), let $Q(m'|m, x_m)$ be a probability mass function satisfying $\sum_{m'} Q(m'|m, x) = 1$ and $Q(m'|m, x_m) = 0$ if $d_{m'} = d_m$. Furthermore, for each m' such that $d_{m'} > d_m$, let $Q(u|m, x_m, m')$ be a pdf on $\mathbb{R}^{d_{m'}-d_m}$. Q will be the proposal distribution and $Q(m'|m, x_m) = 0$ if $d_{m'} = d_m$ implies Q only proposes moves across dimension.

Let $(m, x_m) \sim \pi$. First sample m' from $Q(\cdot|m, x_m)$ and if $d_{m'} > d_m$, then sample u from $Q(\cdot|m, x_m, m')$, which are the extra so called dimension matching continuous r.v.'s needed to generate the candidate sample $x_{m'} \in \mathbb{R}^{d_{m'}}$. Assume $d_{m'} > d_m$. (The reverse case is considered below.) The candidate sample is obtained by applying a *bijection*, to be chosen by the practitioner just as Q was, to yield $x_{m'} = \beta_{m,m'}(x_m, u)$. The acceptance probability for the proposed sample $(m', x_{m'})$ is $\alpha(m', x_{m'}; m, x_m) = \min\{1, r(m', x_{m'}; m, x_m)\}$ where $r(m', x_{m'}; m, x_m)$ is

$$\frac{\pi(m', x_{m'})}{\pi(m, x_m)} \frac{Q(m|m', x_{m'})}{Q(m', u|m, x_m)} |\nabla \beta_{m,m'}(x_m, u)| \tag{19}$$

where the right most term is the Jacobian of $\beta_{m,m'}$. If however $d_{m'} < d_m$, let $(x'_m, u) = \beta^{-1}_{m',m}(x_m)$ and the candidate sample for the move to the lower dimension model is x'_m. The proposal $(m', x_{m'})$ is accepted with probability $\min\{1, r\}$ where $r(m', x_{m'}; m, x_m)$ is

$$\frac{\pi(m', x_{m'})}{\pi(m, x_m)} \frac{Q(m, u|m', x_{m'})}{Q(m'|m, x_m)} |\nabla \beta_{m',m}(x_{m'}, u)|^{-1} \tag{20}$$

In the MTT model, each target configuration $z_{1:n}$ corresponds to a model index m, $\mathbf{x}_{1:n}$ corresponds to the continuous variable x_m, and $p_\theta(z_{1:n}, \mathbf{x}_{1:n}|y_{1:n})$ corresponds to $\pi(m, x_m)$. The bijections $\beta_{m,m'}$ are permutations of the input variables to preserve the MTT ordering rule $\mathcal{O}$ and thus all Jacobians are 1.

3.2 MCMC to sample $(Z_{1:n}, \mathbf{X}_{1:n})$ in loop 1 of Algorithm 1

Algorithm 2 proposes a change to $(Z_{1:n}, \mathbf{X}_{1:n})$ by selecting one of the following proposals at random:

1. *birth/death* proposal to create or delete a target;
2. multi-step *extension/reduction* proposal to extend or reduce an existing track by multiple time units;
3. one-step *extension/reduction* proposal to extend or reduce an existing track by one time unit;
4. *state* proposal to exchange states between targets.

All but the state proposal changes the dimension of $\mathbf{X}_{1:n}$ and hence the corresponding RJMCMC acceptance probability in (19) needs to be derived. However the bijections are such that the Jacobian in (19) is always 1. Each proposal type is now described in detail in the sub-sections below

Algorithm 2: MCMC moves to sample $(Z_{1:n}, \mathbf{X}_{1:n})$

Input: Sample $(z_{1:n}, \mathbf{x}_{1:n})$, data $y_{1:n}$, parameter θ
Output: Updated sample $(z'_{1:n}, \mathbf{x}'_{1:n})$

1 Randomly select a proposal type from {birth/death, multi-step extension/reduction, state, one-step extension/reduction}.
2 Propose $(z'_{1:n}, \mathbf{x}'_{1:n})$ by executing the chosen proposal.
3 Calculate acceptance prob. $\alpha(z'_{1:n}, \mathbf{x}'_{1:n}; z_{1:n}, \mathbf{x}_{1:n})$ (see (21)/(22), (31)/(32), (34)). Output $(z'_{1:n}, \mathbf{x}'_{1:n})$ with prob. α, otherwise $(z_{1:n}, \mathbf{x}_{1:n})$.

3.3 Birth/Death Proposal

The birth/death proposal of Algorithm 2 initiates a new track or deletes an existing track and thus only generates proposed samples $(z'_{1:n}, \mathbf{x}'_{1:n})$ that move across dimension, i.e. is never intra-dimensional. Birth creates a track sequentially in time, until a stopping rule is met, by using the observed images to increase the probability of acceptance and we describe this sequential process in detail in Sec. 3.4 below.

The current MCMC input sample (Algorithm 2) is $(z_{1:n}, \mathbf{x}_{1:n})$ or $\{(k, t_b^k, \mathbf{x}^k)\}_{k=1}^K$ in the alternative parameterization of 2.4. Sampling from the birth/death proposal commences by first choosing to create or delete a track with probability 0.5. If death is chosen, one of the K targets are randomly deleted, say target k with probability $1/K$. If birth is chosen, a new track with birth time t_b and states

$$\mathbf{x} = [x_0, \ldots, x_{l-1}], \quad x_i = (a_i, s_i, v_i)$$

are proposed, where (a_i, s_i, v_i) denote the intensity, spatial coordinates and velocity components of the state at time $t = t_b + i$. Using $(z_{1:n}, \mathbf{x}_{1:n})$ and the newly created track, the ordering rule of Section 2.4 is invoked to obtain the MTT proposed sample $(z'_{1:n}, \mathbf{x}'_{1:n})$. Assume the newly created target has label k' in the alternative parameterization of $(z'_{1:n}, \mathbf{x}'_{1:n})$. The acceptance probability is $\alpha_1 = \min\{1, r_1\}$ where $r_1(z'_{1:n}, \mathbf{x}'_{1:n}; z_{1:n}, \mathbf{x}_{1:n})$ is

$$\frac{p_\theta(z'_{1:n}, \mathbf{x}'_{1:n}, y_{1:n})}{p_\theta(z_{1:n}, \mathbf{x}_{1:n}, y_{1:n})} \frac{(K+1)^{-1}}{q_{b,\theta}(t_b, \mathbf{x}|z_{1:n}, \mathbf{x}_{1:n}, y_{1:n})} \tag{21}$$

The term $q_{b,\theta}(t_b, \mathbf{x}|z_{1:n}, \mathbf{x}_{1:n}, y_{1:n})$, defined in Sec. 3.4 below, is the pdf of the newly created target states which corresponds to term $Q(m', u|m, x_m)$ in the denominator of (19).

If death is chosen, delete target k of $\{(i, t_b^i, \mathbf{x}^i)\}_{i=1}^K$ with probability K^{-1} and let $(z'_{1:n}, \mathbf{x}'_{1:n})$ be the new MTT state excluding target k. The acceptance probability is $\alpha_1 = \min\{1, r_1\}$ where the function $r_1(z'_{1:n}, \mathbf{x}'_{1:n}; z_{1:n}, \mathbf{x}_{1:n})$ is

$$\frac{p_\theta(z'_{1:n}, \mathbf{x}'_{1:n}, y_{1:n})}{p_\theta(z_{1:n}, \mathbf{x}_{1:n}, y_{1:n})} \frac{q_{b,\theta}(t_b^k, \mathbf{x}^k|z'_{1:n}, \mathbf{x}'_{1:n}, y_{1:n})}{K^{-1}} \tag{22}$$

3.4 Birth proposal for creating a new target

The birth procedure, summarised in Alg. 3, proposes a new target trajectory, or track, by using the *residual* images to first construct the intensity and spatial coordinates of the entire track. The track's velocity values are then sampled conditionally on the created spatial components under Gaussian assumptions.

Algorithm 3: Birth proposal

Input: Sample $(z_{1:n}, \mathbf{x}_{1:n})$, data $y_{1:n}$, parameter θ
Output: A new target trajectory $\mathbf{x} = [x_0, \ldots, x_{l-1}]$ where $x_i = (a_i, s_i, v_i) \in \mathbb{R}^5$.

1 Randomly choose a birth time $t_b \in \{1, \ldots, n\}$, set $t_0 = t_b$, $k = 0$.
2 **repeat**
3 Calculate residual image $y^r_{t_k}$ in (23), match filtered image $y^f_{t_k}$ in (24), local maxima set G_{t_k} in (25).
4 *Exit* loop if G_{t_k} is empty, otherwise randomly choose pixel $i \in G_{t_k}$ and perform test (26).
5 *Exit* loop if test failed, otherwise sample target intensity and position, (a_k, s_k), from (27).
6 Set $t_{k+1} = t_k + 1$, set $k = k + 1$. *Exit* loop with probability $1 - p_s \mathbb{I}[t_k \leq n]$.
7 **until** *Exit*
8 If $t_k = t_b$, exit birth proposal with null output.
9 Set $t_d = t_k$, set $l = k$ and sample velocity components $v_0, \ldots, v_{l-1}$.

3.4.1 Residual and match filtered images

The birth proposal commences by choosing a random birth time and then calculates the residual and matched

filtered images as follows: using the current MCMC input sample $(z_{1:n}, \mathbf{x}_{1:n})$, subtract the contribution of the k_t^x targets and background intensity b_t from the image y_t to get the residual image y_t^r where

$$y_{t,i}^r = y_{t,i} - h_i(\mathbf{x}_t) - b_t, \qquad i = 1, \ldots, m. \tag{23}$$

Match filter y_t^r to get the image y_t^f where the j-th pixel in the filtered image is

$$y_{t,j}^f = \frac{1}{E_{\bar{h}}} \sum_{i=1}^{m} y_{t,i}^r \bar{h}_i(j) \tag{24}$$

where $\bar{h}_i(j)$ is defined in (3)-(4) and $E_{\bar{h}} = \sum_{i=1}^m \bar{h}_i(j)^2$ is the energy of the filter $\{\bar{h}_i(j)\}_{i=1}^m$. (The sum that defines $y_{t,j}^f$ can be truncated to $i \in L(j)$.) The rationale is that the presence of a target at or close to pixel j will likely result in $y_{t,j}^f$ being a local maxima among pixels. Put another way, local maxima of $y_{t,j}^f$ are likely locations of targets. Let

$$G_t = \{1 \le i \le m : y_{t,i}^f \text{ is a local maxima}, y_{t,i}^f \ge \gamma_t(\theta)\} \tag{25}$$

where $\gamma_t(\theta)$ is a time-dependent threshold chosen to avoid peaks that are not likely to be target generated. A definition is given in Sec. 4, the numerical section.

3.4.2 Proposing state values

Line 4 of Alg. 3 randomly chooses a pixel $i \in G_t$. As a local intensity maxima is not necessarily target generated, perform a hypothesis test on the square of $l \times l$ pixels $L(i)$ centered at chosen maxima $i \in G_t$. Let $y_{t,L(i)}^r = \{y_{t,j}^r, j \in L(i)\}$, H_1 the hypothesis that $y_{t,L(i)}^r$ is generated by a new born target and H_0 the converse that it is purely background noise generated. Calculate the test ratio

$$\rho(y_{t,L(i)}^r) = \frac{p(y_{t,L(i)}^r | H_1)}{p(y_{t,L(i)}^r | H_0)} \tag{26}$$

While $p(y_{t,L(i)}^r | H_0) := \prod_{j \in L(i)} \mathcal{N}(y_{t,j}^r; 0, \sigma_{r,t}^2)$ can be calculated analytically, $p(y_{t,L(i)}^r | H_1)$ is intractable but can be estimated, e.g. we use the Laplace approximation. (See Appendix A for details.) H_1 is accepted with probability $\min\{1, \rho(y_{t,L(i)}^r)\}$. The loop of Alg. 3 is exited if H_1 is rejected. If H_1 is accepted, sample the intensity and position components of the target state from

$$(A_t, S_t) \sim \mathcal{N}_{t,i}(\cdot) \tag{27}$$

where $\mathcal{N}_{t,i}(\cdot)$ is a Gaussian derived when calculating the hypothesis test and is given in Appendix A. The subscript (t, i) indicates this approximation is specific to pixel i of time t; recall $i \in G_t$ was the chosen maxima.

The birth loop continues for $t = t_b + k$, $k > 0$, until a stopping criteria is met. The birth loop stops at some time $t = t_d$, yielding a target lifespan $l = t_d - t_b$, when either $t > n$, the target does not survive to the next state with probability $1 - p_s$, set G_t is empty or H_1 is rejected. The output of the birth loop is $(a_0, s_0), \ldots, (a_{l-1}, s_{l-1})$ which is the complete trajectory of intensity and positions of the new born target. The velocity components are now generated to yield

$$\mathbf{x} = [x_0, \ldots, x_{l-1}], \quad x_i = (a_i, s_i, v_i).$$

For linear Gaussian state dynamics, the velocity can be sampled conditionally on the spatial locations $s_0, \ldots, s_{l-1}$ and more generally, a Gaussian approximation technique could be employed. The numerical examples (synthetic and real data) assume Gaussian targets.

3.4.3 Proposal density of the birth move

Denoting $t_k = t_b + k$, $k = 0, \ldots, l$, we can write the pdf for proposing $(t_b, \mathbf{x})$ in the birth move as

$$\begin{aligned} q_{b,\theta}(t_b, \mathbf{x} | z_{1:n}, \mathbf{x}_{1:n}, y_{1:n}) &= q_0(t_b) q_1(a_0, s_0 | y_{t_0}^r) \\ &\times \prod_{k=1}^{l-1} q_2(a_k, s_k | a_{0:k-1}, s_{0:k-1}, y_{t_k}^r) \\ &\times q_3(\text{stop} | a_{0:l-1}, s_{0:l-1}, y_{t_l}^r)\ q_4(v_{0:l-1} | a_{0:l-1}, s_{0:l-1}) \end{aligned} \tag{28}$$

where $q_0(t_b)$ is the probability of choosing the birth time t_b, q_1 corresponds to proposing the target's initial intensity and position, and can be written as

$$\begin{aligned} q_1(a_0, s_0 | y_{t_0}^r) &\propto \mathbb{I}[G_{t_0} \neq \emptyset] \\ &\times \sum_{i \in G_{t_0}} q(i | G_{t_0}) \min(1, \rho(y_{t_0, L(i)}^r)) \mathcal{N}_{t_0,i}(a_0, s_0) \end{aligned} \tag{29}$$

where $\mathbb{I}[G_{t_0} \neq \emptyset]$ is 1 if G_{t_0} is non-empty; $q(i|G_{t_0})$ is the probability of choosing local intensity peak $i \in G_{t_0}$; $\min(1, \rho(y_{t_0,L(i)}^r))$ is the probability of accepting hypothesis H_1 at peak $i \in G_{t_0}$; $\mathcal{N}_{t_0,i}$ is the Gaussian density approximation for the initial value.

The law q_2 is that for adding more states sequentially and may be written as

$$\begin{aligned} q_2(a_k, s_k | a_{0:k-1}, s_{0:k-1}, y_{t_k}^r) &= \mathbb{I}[G_{t_k} \neq \emptyset] \\ &\times p_s \sum_{i \in G_{t_k}} q(i | G_{t_k}) \min(1, \rho(y_{t_k, L(i)}^r)) \mathcal{N}_{t_k,i}(a_k, s_k) \end{aligned} \tag{30}$$

which is similar to q_1 except that the previously created states $a_{0:k-1}$ and $s_{0:k-1}$ are used to calculate $\rho(y_{t_k,L(i)}^r)$,

which is defined as in (26) with the difference here being $p(H_1) = p_s$ (target survival probability.) The likelihood ratio term in (26) is derived in Appendix A, as is the Gaussian term $\mathcal{N}_{t_k,i}$ in (30).

When $t_l = t_b + l > n$ stopping at t_l is certain. Otherwise, the law q_3 corresponds to stopping due to the target not surviving, G_{t_l} being empty, or the hypothesis test failing and is given by

$$q_3(\text{stop}|a_{0:l-1}, s_{0:l-1}, y^r_{t_l}) = 1 - p_s \mathbb{I}[G_{t_l} \neq \emptyset] \sum_{i \in G_{t_l}} q(i|G_{t_l}) \min(1, \rho(y^r_{t_l,L(i)}))$$

For linear and Gaussian state dynamics, conditioned on $a_0, s_0, \ldots, a_{l-1}, s_{l-1}$, the velocity can be sampled (exactly) since q_4 will be a Gaussian distribution as well.

3.5 Multi-step Extension/Reduction proposal

This proposal extends or reduces the trajectory of a randomly chosen target. A target's trajectory is extended (/reduced) by bringing forward its birth (/death) time or delaying its death (/birth) time. The extra state values are then appended (/discarded) accordingly. Like the birth/death proposal, this proposal only moves the MCMC sample across dimension.

The current MCMC input sample (Algorithm 2) is $(z_{1:n}, \mathbf{x}_{1:n})$ or $\{(k, t^k_b, \mathbf{x}^k)\}_{k=1}^K$ in the alternative parameterization of 'sec. 2.4. Sampling from the multi-step extension/reduction proposal commences with choosing between extension and reduction equiprobably and then doing one of the following:

Extension: From the subset of targets with lifetimes less than n, randomly select a target and extension direction. (Without the lifetime restriction the chosen target cannot be extended further.) The direction of extension is chosen equiprobably if both forward and backward extensions are permissible. Assume target k is selected for a forward extension, denoted k_+. A new (delayed) death time and corresponding trajectory extension $\mathbf{x} = (x_1, x_2, \ldots)$ is proposed to yield the new extended trajectory $\hat{\mathbf{x}}^k = (\mathbf{x}^k, \mathbf{x})$. For a backward extension of target k, the event denoted by k_-, a new (earlier) birth time, denoted τ^k_b and trajectory extension $\mathbf{x}$ is proposed to yield $\hat{\mathbf{x}}^k = (\mathbf{x}, \mathbf{x}^k)$. The ordering rule of Section 2.4 is invoked to obtain the MTT proposed sample $(z'_{1:n}, \mathbf{x}'_{1:n})$ from the unaltered targets $\{(t^i_b, \mathbf{x}^i)\}_{i=1,i\neq k}^K$ and the extended target $(\tau^k_b, \hat{\mathbf{x}}^k)$. The acceptance probability, denoted α_2, is $\min\{1, r_2\}$ where function $r_2(z'_{1:n}, \mathbf{x}'_{1:n}; z_{1:n}, \mathbf{x}_{1:n})$ is

$$\frac{p_\theta(z'_{1:n}, \mathbf{x}'_{1:n}, y_{1:n})}{p_\theta(z_{1:n}, \mathbf{x}_{1:n}, y_{1:n})} \frac{q_{r,\theta}(z_{1:n}, \mathbf{x}_{1:n}|z'_{1:n}, \mathbf{x}'_{1:n})}{q_{e,\theta}(k_{+/-}, \mathbf{x}|z_{1:n}, \mathbf{x}_{1:n}, y_{1:n})}. \quad (31)$$

Here $q_{e,\theta}(k_{+/-}, \mathbf{x}|z_{1:n}, \mathbf{x}_{1:n}, y_{1:n})$ is the probability density of choosing target k, the extension direction and states $\mathbf{x}$, which can be calculated similarly to the expression (28) of the birth move and it is not repeated here. We denote the total probability of making the return transition from $(z'_{1:n}, \mathbf{x}'_{1:n})$ to $(z_{1:n}, \mathbf{x}_{1:n})$ via the reduction, described next, with the term $q_{r,\theta}$ above.

Reduction: Randomly select a target from the subset of targets with lifetimes exceeding one and then the reduction direction, either forwards or backwards, equiprobably. Let k_+ denote target k for forward reduction and k_- for backward reduction. In the forward case, a reduction time point $t \in \{t^k_b + 1, \ldots, t^k_d - 1\}$ is chosen randomly and discard the time $\{t, \ldots, t^k_d - 1\}$ section of the track, causing t to be the new death time. If backwards then $t \in \{t^k_b, \ldots, t^k_d - 2\}$ is chosen randomly and the $\{t^k_b, \ldots, t\}$ portion is discarded to yield a new birth time of $t + 1$. Let τ^k_b denote the (possibly new) birth time, $\hat{\mathbf{x}}^k$ the retained trajectory and $\mathbf{x}$ the discarded forward/backward state trajectory of target k. The ordering rule of Section 2.4 is invoked to obtain the MTT proposed sample $(z'_{1:n}, \mathbf{x}'_{1:n})$ from the the unaltered targets $\{(t^i_b, \mathbf{x}^i)\}_{i=1,i\neq k}^K$ and the reduced target $(\tau^k_b, \hat{\mathbf{x}}^k)$. Let $q_{r,\theta}(z'_{1:n}, \mathbf{x}'_{1:n}|z_{1:n}, \mathbf{x}_{1:n})$ denote the total probability of making the transition from $(z_{1:n}, \mathbf{x}_{1:n})$ to $(z'_{1:n}, \mathbf{x}'_{1:n})$ via the described reduction step, then

$$q_{r,\theta}(z'_{1:n}, \mathbf{x}'_{1:n}|z_{1:n}, \mathbf{x}_{1:n}) = \frac{1}{\sum_{j=1}^K \mathbb{I}[l^j > 1]} \frac{1}{2} \frac{1}{l^k - 1}.$$

Assume the reduced target has label k' in the alternative parameterization of $(z'_{1:n}, \mathbf{x}'_{1:n})$. The acceptance probability is $\alpha_2 = \min\{1, r_2\}$ where $r_2(z'_{1:n}, \mathbf{x}'_{1:n}; z_{1:n}, \mathbf{x}_{1:n})$ is

$$\frac{p_\theta(z'_{1:n}, \mathbf{x}'_{1:n}, y_{1:n})}{p_\theta(z_{1:n}, \mathbf{x}_{1:n}, y_{1:n})} \frac{q_{e,\theta}(k'_{+/-}, \mathbf{x}|z'_{1:n}, \mathbf{x}'_{1:n}, y_{1:n})}{q_{r,\theta}(z'_{1:n}, \mathbf{x}'_{1:n}|z_{1:n}, \mathbf{x}_{1:n})} \quad (32)$$

3.6 One-step Extension/Reduction proposal

In addition to the multi-step extension/reduction proposal, a *one-step* extension/reduction proposal is also employed. This proposal proceeds as in multi-step but with the following two differences: i) it extends or truncates the trajectory of the selected target by one time point only. ii) The trajectory extension is sampled from the prior distribution of the target motion model.

Like the birth proposal, the multi-step extension uses the intensity threshold $\gamma_t(\theta)$ (as described in Sec. 3.4) to sample the extended trajectory. This may result in the next state of a momentarily dim target not being detected but one-step extension/reduction will be able to extend the target.

Algorithm 4: One-step Extension/Reduction

Input: Sample $(z_{1:n}, \mathbf{x}_{1:n})$ or $\{(k, t_b^k, \mathbf{x}^k)\}_{k=1}^K$ in alternative parameterisation, parameter θ

Output: A shortened or extended target trajectory $(k, \tau_b^k, \hat{\mathbf{x}}^k)$.

Choose between extension (e) or reduction (r) equiprobably then choose a target k and direction $+$ or $-$. Assume $\mathbf{x}^k = (x_0^k, \ldots, x_{l^k-1}^k)$

If $(k, e, +)$ extend target k's trajectory to $\hat{\mathbf{x}}^k = (\mathbf{x}^k, x)$ by $x \sim f_\psi(x|x_{l^k-1}^k)$. Set $\tau_b^k = t_b^k$.

If $(k, e, -)$ extend target k's trajectory to $\hat{\mathbf{x}}^k = (x, \mathbf{x}^k)$ by $x \sim \frac{\mu_\psi(x) f_\psi(x_0^k|x)}{\int \mu_\psi(x') f_\psi(x_0^k|x')\mathrm{d}x'}$ and set $\tau_b^k = t_b^k - 1$.

If $(k, r, +)$ set $\hat{\mathbf{x}}^k = (x_0^k, \ldots, x_{l^k-2}^k)$ and $\tau_b^k = t_b^k$.

If $(k, r, -)$ set $\hat{\mathbf{x}}^k = (x_1^k, \ldots, x_{l^k-1}^k)$ and $\tau_b^k = t_b^k + 1$.

Note that for a backward extension, the appended state is sampled from initial distribution μ_ψ conditioned on the value of its next state for Gaussian targets. The MTT proposed sample $(z'_{1:n}, \mathbf{x}'_{1:n})$ is the unaltered targets $\{(t_b^i, \mathbf{x}^i)\}_{i=1, i\neq k}^K$ and the altered target $(\tau_b^k, \hat{\mathbf{x}}^k)$. The acceptance probability is $\alpha_3 = \min\{1, r_3\}$ with r_3 is defined similarly as in the multi-step case (31) and (32).

3.7 State proposal

This proposal chooses a pair of targets and swaps a section of their state trajectories. In particular, it randomly chooses a time $t < n$ and then randomly changes I_{t+1}, which is the vector that links targets in $\mathbf{X}_t$ and $\mathbf{X}_{t+1}$, as illustrated in Figure 2. When $X_{t,i}$ has descendant $X_{t+1,g}$, it can propose to swap its descendant with that of $X_{t,j}$ (case 1), or change its descendant to the initial state $X_{t+1,h}$ of a target born at time $t+1$ (case 2), or to delete the link (case 4). When $X_{t,i}$ has no descendant, it can be merged with a new born target at time $t+1$ by linking to its initial state (case 3), or steal another surviving target's descendant (case 5). The state proposal is purely intra-dimensional (or in the context of Sec. 3.1 it moves between models pairs (m, m') satisfying $d_m = d_{m'}$.)

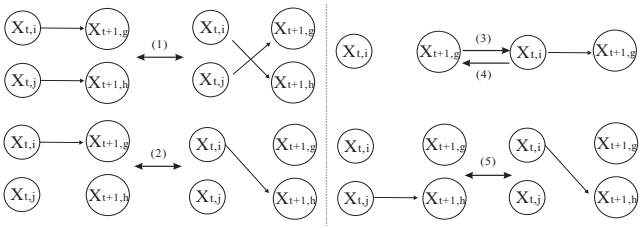

Fig. 2: State move.

The current MCMC input sample (Algorithm 2) is $(z_{1:n}, \mathbf{x}_{1:n})$ or $\{(k, t_b^k, \mathbf{x}^k)\}_{k=1}^K$ in the alternative parameterization of 2.4. Sampling from the state proposal commences with choosing a time $t < n$ and a pair of targets (labels) $U = \{i, j\}$ from the total set of targets $\{1, \ldots, K\}$ subject to targets i and j being alive at times t and $t+1$ respectively. Choosing $i = j$ is permitted (then $U = \{i\}$) and target i must be alive at time t and $t+1$. (For example, in implementation we chose a target i from time t, and a target state value from the set of time $t+1$ targets with probability inversely proportional to the distance from target i's state value at time t.) We denote the probability of a particular selection by $q_{s,\theta}(t, U|\mathbf{x}_{1:n}, z_{1:n})$. If $i = j$ then split target i into two targets as in case 4 of Figure 2. If $i \neq j$, pair the *ancestors* of i with the *descendants* of j in the manner shown in Figure 2 (all cases except 4), i.e. the swap alters trajectories $\mathbf{x}^i$ and $\mathbf{x}^j$ to

$$\begin{aligned} \mathbf{x}^i &\to (x_0^i, \ldots, x_s^i, x_{s'}^j, \ldots, x_{lj-1}^j) =: \hat{\mathbf{x}}^i \\ \mathbf{x}^j &\to (x_0^j, \ldots, x_s^j, x_{s'}^i, \ldots, x_{l^i-1}^i) =: \hat{\mathbf{x}}^j \end{aligned} \tag{33}$$

where $t_b^i + s = t$, $t_b^j + s' = t+1$. Note the birth time (of at most one target) may change. Call the birth times after the swap τ_b^i, τ_b^j

The next part of the state move then proposes a change to the continuous variables of the affected targets $k \in U = \{i, j\}$ form $\hat{\mathbf{x}}^k$ to $\tilde{\mathbf{x}}^k$ to increase the chance of the move being accepted. This is because changing the links between targets at time t and $t+1$ can cause a mismatch in the velocity and intensity of the newly formed links. As such, the state move will then propose a change to the velocity and intensity components of the affected targets while retaining their original spatial position components. Let $q_{s,\theta}(t, U, (\tilde{\mathbf{x}}^k)_{k\in U}|\mathbf{x}_{1:n}, z_{1:n}, y_{1:n})$ denote the joint pdf of selecting (t, U) and the change $(\hat{\mathbf{x}}^i, \hat{\mathbf{x}}^j) \to (\tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^j)$. (See Appendix B for the expression.) Finally, we let $(z'_{1:n}, \mathbf{x}'_{1:n})$ denote MTT state (in the original parameterization) of the combined collection of unaltered $\{t_b^k, \mathbf{x}^k\}_{k\in\{1\ldots,K\}/U}$ and altered targets $\{\tau_b^k, \tilde{\mathbf{x}}^k\}_{k\in U}$. The acceptance probability of the state move is then $\alpha_3 = \min\{1, r_3\}$ where function $r_3(z'_{1:n}, \mathbf{x}'_{1:n}; z_{1:n}, \mathbf{x}_{1:n})$ is

$$\frac{p_\theta(z'_{1:n}, \mathbf{x}'_{1:n}, y_{1:n})}{p_\theta(z_{1:n}, \mathbf{x}_{1:n}, y_{1:n})} \frac{q_{s,\theta}(t, U', (\mathbf{x}^k)_{k\in U'}|\mathbf{x}'_{1:n}, z'_{1:n}, y_{1:n})}{q_{s,\theta}(t, U, (\tilde{\mathbf{x}}^k)_{k\in U}|\mathbf{x}_{1:n}, z_{1:n}, y_{1:n})} \tag{34}$$

U' is the label set of the targets with swapped components in the alternate parameterisation of $(z'_{1:n}, \mathbf{x}'_{1:n})$.

4 Numerical examples

This section presents the two main numerical examples. The first one uses synthetic data and assumes known

MTT parameters so that a fair comparison can be made between our MCMC tracker (Algorithm 1 excluding parameter learning) and the multi-Bernoulli (MB) filter of (Vo et al, 2010). (The MB tracker does not learn parameters.) The numerical results will demonstrate the performance improvements of our method when tracking targets that are close to each other with overlapping illumination regions. The second example is a real-data example that applies Algorithm 1 to track Fab[1] labelled Jurkat T-cells. The tracking method currently used by biochemists (Weimann et al, 2013) extracts point measurements from the images and then connects them to form trajectories using a nearest neighbour type method. Our algorithm will be shown to outperform theirs when tracking dim targets as well as targets with overlapping illumination regions. All simulations were run in Matlab on a PC with an Intel i5 2.8 GHZ $\times 2$ processor.

Gaussian targets and θ: Recall from (1) an individual target's state is $X_t = (A_t, S_t(1), S_t(2), V_t(1), V_t(2))$. For the numerical examples, we use a drifting intensity and near constant velocity motion model,

$$A_t = A_{t-1} + U_t, \quad U_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_i^2),$$
$$[S_t(j), V_t(j)] = [S_{t-1}(j) + \delta V_{t-1}(j), V_{t-1}(j)] + U_{t,j},$$
$$U_{t,1}^{\mathrm{T}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_x^2 \Sigma), U_{t,2}^{\mathrm{T}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_y^2 \Sigma),$$

where δ is the (known) sampling interval and

$$\Sigma = \begin{pmatrix} \delta^3/3 & \delta^2/2 \\ \delta^2/2 & \delta \end{pmatrix}.$$

The initial hidden state is assumed to be Gaussian distributed with mean $\mu_b = (\mu_{bi}, \mu_{bx}, \mu_{by}, 0, 0)^{\mathrm{T}}$ and covariance $\Sigma_b = \mathrm{diag}(\sigma_{bi}^2, \sigma_{bp}^2, \sigma_{bp}^2, \sigma_{bv}^2, \sigma_{bv}^2)$. Note the common variance σ_{bp}^2 for the spatial coordinate. (Similarly for the velocity.) The mean of the initial velocity is set to be 0 in the absence of more information but this still can yield directional motion if the observations support this. All the parameters ψ of the hidden state dynamics are (see (1))

$$\psi = (\mu_{bi}, \mu_{bx}, \mu_{by}, \sigma_{bi}^2, \sigma_{bp}^2, \sigma_{bv}^2, \sigma_i^2, \sigma_x^2, \sigma_y^2). \tag{35}$$

and augmenting ψ with the parameters of the target birth/death and observation models gives

$$\theta = (\psi, p_s, \lambda_b, b_1, \sigma_{r,1}^2, \ldots, b_n, \sigma_{r,n}^2). \tag{36}$$

Prior for θ: All the variance components above have identical independent priors, which is the inverse gamma distribution $\mathcal{IG}(\alpha_0, \beta_0)$ with common shape α_0 and scale β_0 parameters. Given $\sigma_{bi}^2, \sigma_{bp}^2, \sigma_{r,t}^2$ (for $t = 1, \ldots, n$), the priors of $\mu_{bi}, \mu_{bx}, \mu_{by}$ and b_t are $\mu_{bi}|\sigma_{bi}^2 \sim \mathcal{N}(\mu_0, \sigma_{bi}^2/n_0)$, $\mu_{bx}|\sigma_{bp}^2 \sim \mathcal{N}(\mu_0, \sigma_{bp}^2/n_0)$, $\mu_{by}|\sigma_{bp}^2 \sim \mathcal{N}(\mu_0, \sigma_{bp}^2/n_0)$, $b_t|\sigma_{r,t}^2 \sim \mathcal{N}(\mu_0, \sigma_{r,t}^2/n_0)$. The conjugate prior of p_s is the uniform distribution $\mathcal{U}(0,1)$ and for λ_b the Gamma distribution $\mathcal{G}(\alpha_0, \beta_0)$, α_0 is the shape and β_0 the scale parameter α. We set $\alpha_0 \ll 1, \beta_0 \gg 1$ to make the prior less informative.

Intensity threshold $\gamma_t(\theta)$: The illuminated region $L(s)$ is an $l \times l$ square region of pixels centered at s, with $l = 1 + \lceil 4\sigma_h \rceil / \Delta$ where $\lceil \cdot \rceil$ rounds up its argument. The intensity threshold $\gamma_t(\theta)$ is chosen to be $\gamma_t(\theta) = \min(\mu_{bi} - 3\sigma_{bi}, 3\sigma_{r,t}/\sqrt{E_{\bar{h}}})$ using the following rationale. We expect $y_{t,j}^f$ in (24) to exceed $\mu_{bi} - 3\sigma_{b,i}$ (mean birth illumination minus 3 times its standard deviation) with high probability if a target is present in pixel j of the residual image in (23). However, assuming no targets illuminate pixels $L(s)$ of the residual image, $\sigma_{r,t}/\sqrt{E_{\bar{h}}}$ is the standard deviation of $y_{t,j}^f$. With high probability, $y_{t,j}^f$ should not be exceed $3\sigma_{r,t}/\sqrt{E_{\bar{h}}}$ and avoids triggering detection.

4.1 Comparison with the multi-Bernoulli tracker

We compared our algorithm with the MB tracker (Vo et al, 2010). Unlike the subsequent real data example, this synthetic case assumed $b_t = 0$ and $\sigma_{r,t}^2 = \sigma_r^2$ for all t (see (11).) We synthesised 50 frames (images) of 168×184 pixels each using the parameter vector

$$\psi^* = (30, 0, 0, 4, 25, 3, 0.5, 0.3, 0.7) \tag{37}$$
$$\theta^* = (\psi^*, 0.95, 0.3, 1). \tag{38}$$

We set $\sigma_h^2 = 1$, $\Delta = 1$. This gives a 5×5 pixel square for the illuminated region $L(s)$ (see (3)). From (3), define $\mathrm{SNR} = 20 \log(\frac{a\Delta^2/2\pi\sigma_h^2}{\sigma_r})$. For $a = \mu_{bi} = 30$, the initial SNR is 13.6 dB. The synthetic data had 20 targets whose trajectories are shown in Figure 3 along with the trajectories obtained by running the MCMC tracker, i.e. Alg. 1 with θ update omitted, with $n_1 =$ 30and 15 particles per target for the CSMC step.

Figure 3 shows all targets being tracked completely by Alg. 1 in contrast to the MB tracker of Vo et al (2010). The birth process assumed by the MB tracker has four terms each of which has the same initial distribution $\mathcal{N}(\cdot|\mu_b, \Sigma_b)$ and existence probability 0.1. Pruning and merging targets are performed as suggested in Vo et al (2010) to eliminate tracks with existence probability less than 0.01 and merge two tracks when they fall within a fraction (3/4) of a pixel size in distance. The number of particles assigned for each hypothesised target in MB tracker is restricted between 5000 and 8000. In Figure 3, it is seen that some tracks are lost after they cross, which is the main limitation of the MB

[1] Fab (Fragment antigen-binding) is a region on an antibody that binds to antigens.

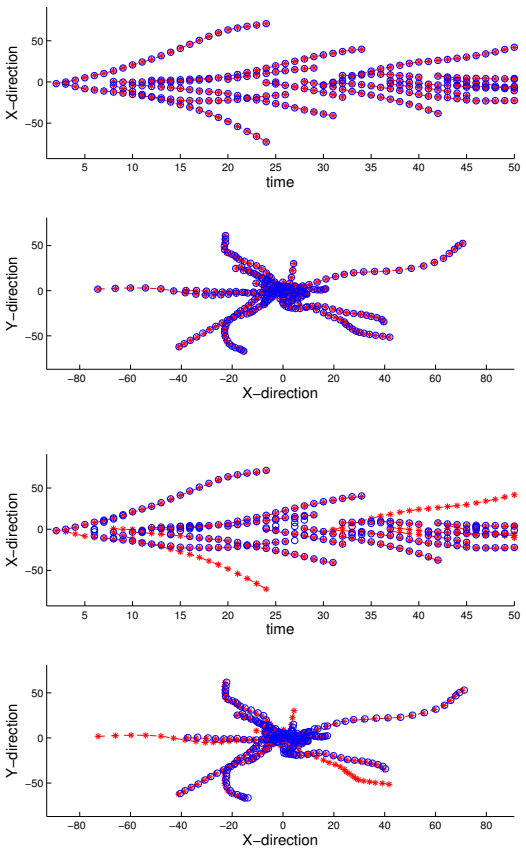

Fig. 3: Left column (top and bottom figures) tracked targets of Alg. 1, right column the multi-Bernoulli filter in Vo et al (2010). Tracks labelled $-*-$ (red) is ground truth while (blue) circles are the estimates. Multi-Bernoulli displays some error in tracking.

tracker as pointed out in Vo et al (2010). This is because crossing targets invalidates the crucial assumption, necessary to derive the MB tracker, that the illuminated region of the targets do not overlap. In terms of the computation time, the MB tracker take less than one minute to run while the MCMC tracker takes 6 minutes. Closely spaced targets are common place in many applications, an example being the real-data experiment reported below. Our MCMC tracker should be viewed as a method applicable to closely spaced targets and not as a competitor of a technique optimized for non-overlapping targets like the MB tracker.

The previous comparison was done assuming known model parameters. The current example revisits the same data set assuming θ^* in (38) is unknown to Algorithm 1, which was re-run with the initial parameter set to

$$\theta^{(0)} = (45, 10, 5, 8, 50, 6, 3, 1, 1.5, 0.6, 1, 4),$$

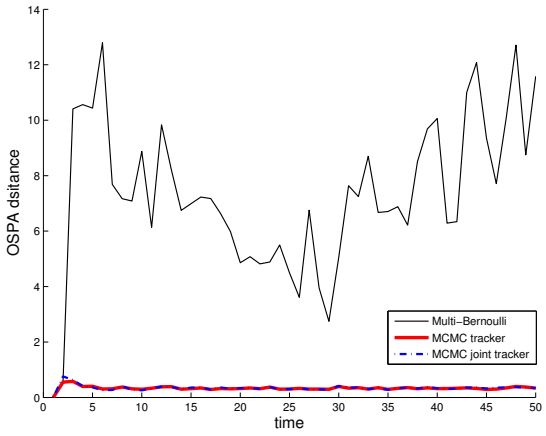

Fig. 4: Comparing the OSPA tracking error of the multi-Bernoulli filter assuming known θ^* (top solid black line) with Alg. 1 (lower traces) when θ^* is known (red) and when θ^* is inferred (blue). OSPA errors of MCMC almost equal for known and inferred θ^*.

while MB output of the previous example was used. (The MB was given the true parameter as it does not incorporated parameter learning. The MCMC outputs of Algorithm 1 will be denoted $(z_{1:n}^{(i)}, \mathbf{x}_{1:n}^{(i)}, \theta^{(i)})$.) OSPA distances (Vo et al, 2010) of the three algorithms are plotted in Figure 4. The OSPA, or optimal sub-pattern assignment, error between two set of points $(u_1, \ldots, u_m)$ and $(v_1, \ldots, v_n)$ in $\mathbb{R}^d$, assuming $n > m$, is the sum of location and cardinality error

$$\min_{j_{1:m} \subset 1:n} \sum_{i=1}^{m} \frac{1}{n} \left(||u_i - v_{j_i}|| \wedge c \right) + c(1 - \frac{m}{n}).$$

Figure 4 shows that the tracking error (penalty $c = 20$) with unknown parameters is similar to the known case reported earlier. A further verification is the probability density values plotted in Figure 5 where $p_{\theta*}(z_{1:n}^{(i)}, \mathbf{x}_{1:n}^{(i)}, y_{1:n})$ was calculated from the previous experiment (Algorithm 1 with known parameters) and $p_{\theta^{(i)}}(z_{1:n}^{(i)}, \mathbf{x}_{1:n}^{(i)}, y_{1:n})$ are the density values from Algorithm 1 with parameter learning.

Figure 6 shows the histograms of 2000 post burn-in parameter samples of Algorithm 1 as the approximation of $p(\theta|y_{1:n})$. The (red) dashed lines show the MLE estimate $\hat{\theta}$ obtained using the true value of the latent variables, i.e. $(z_{1:n}^*, \mathbf{x}_{1:n}^*)$. Specifically, $\hat{\theta}$ is comprised of (see (13), (14), (15))

$$(\hat{p}_s, \hat{\lambda}_b) = \arg\max_{p_s, \lambda_b} p(z_{1:n}^*), \quad \hat{\psi} = \arg\max_{\psi} p(\mathbf{x}_{1:n}^* | z_{1:n}^*),$$
$$\hat{\sigma}_r = \arg\max_{\sigma_r} p(y_{1:n} | \mathbf{x}_{1:n}^*, z_{1:n}^*). \tag{39}$$

As a correctness check, for an uninformative prior, the posterior modes should be consistent with MLE of θ^*.

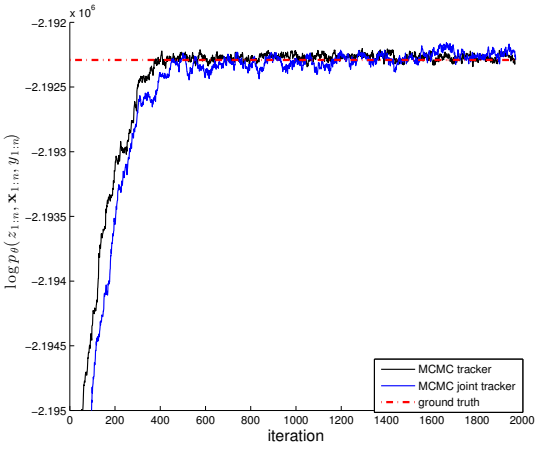

Fig. 5: Plot of $p_{\theta*}(z^{(i)}_{1:n}, \mathbf{x}^{(i)}_{1:n}, y_{1:n})$ of Alg. 1 assuming known θ^* (black line) and $p_{\theta^{(i)}}(z^{(i)}_{1:n}, \mathbf{x}^{(i)}_{1:n}, y_{1:n})$ (blue line) of Alg. 1 when θ^* is also inferred. Horizontal axis is MCMC iteration number i. Horizontal (red) line indicates ground truth $p_{\theta^*}(z^*_{1:n}, \mathbf{x}^*_{1:n}, y_{1:n})$.

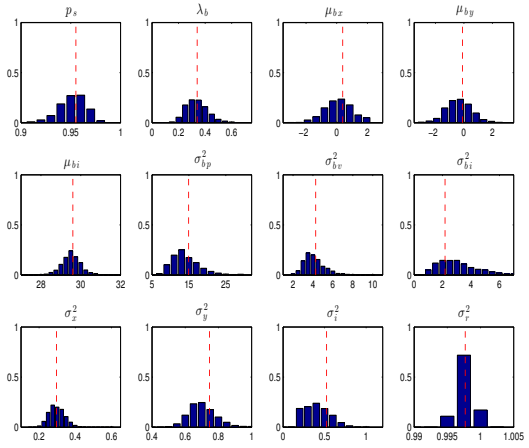

Fig. 6: Histograms of 2000 parameter samples obtained by Algorithm 1. The estimates of model parameters in (35) and (36) shown as histograms. Unlike the subsequent real data example, this synthetic case assumes $b_t = 0$ and $\sigma^2_{r,t} = \sigma^2_r$ for all t. Vertical (red) lines indicate the *idealised* MLE estimate in (39). Agreement of Bayesian and idealised MLE (latter centred in histograms) indicative of correct performance.

The true MLE is $\arg\max_\theta p_\theta(y_{1:n})$, which will be different from (38), is not available as $p_\theta(y_{1:n})$ in (16) is intractable. We use $\hat{\theta}$ of (39) as the surrogate. Note the modes of the posterior do coincide with the surrogate MLE.

4.2 Experiments on Fab labelled Jurkat T-cells

The source of the data were Jurkat T-cells, an immortalised cell line of human T-lymphocytes which plays an important function in immune response. Cells were imaged using a microscopy technique called *total internal reflection fluorescence microscopy*. The cell's molecules of interest were bound to antibodies labelled with a bright green-fluorescent dye (Alexafluo48). It is known that the number of labeled molecules (or 'targets') can be high; at physiological levels there can be several molecules per square micron.

The data is comprised of 20 frames of 115×120 pixel images with a pixel size $176nm$ and a frame rate of 17.8 frame/s. The diffusion coefficient D is expected to be in the region of 0.01-0.1um^2/s. This implies the displacement of molecules between consecutive frames is expected to be in the order of 0.1-1 pixel. The estimated initial SNR here is around 10dB (estimated from the tracking result). Figure 8 (left) shows one frame of the observed images.

4.2.1 Parameter initialisation

In choosing the initial parameter vector $\theta^{(0)}$, some components were chosen arbitrarily while others were informed choices guided by the observed images. (Note that the initialisation step does not need to be overly precise as our algorithm does not depend on specific initial values to work.) We set $p_s = 0.6, \lambda_b = 0.2$ arbitrarily; $\mu^{(0)}_{bx} = \mu^{(0)}_{by} = 60$ to coincide with the image centres since the centres appear much brighter than the periphery; $\mu^{(0)}_{bi} = 70$ is calculated from (3) assuming a target is in the middle of the brightest pixel of the first frame; set $(\sigma^2_{bp})^{(0)} = 400$ by roughly observing that the bright spots are sparse and span the whole image; $(\sigma^2_{bv})^{(0)} = 1$ covers the velocity range of the molecules (0.1-1 pixel per image) and flat enough to allow different possible diffusion coefficients; $(\sigma^2_{bi})^{(0)} = 100$ arbitrarily; $(\sigma^2_x)^{(0)} = (\sigma^2_y)^{(0)} = 0.1$ for small initial driving state noise; $(\sigma^2_i)^{(0)} = 25$ arbitrarily. The time-varying observation noise statistics, mean $(b_t)^{(0)}$ and variance $(\sigma^2_{r,t})^{(0)}$, are initialised to equal the mean and variance of the pixel intensities at that time since bright pixels are sparse in the images.

The point spread parameter σ_h is not estimated and fixed at $\sigma_h = 2$ (normalised with Δ^2). The illuminated region $L(s)$ has 9×9 pixels. The value of σ_h if often known to the experimentalist otherwise, a bit tuning is required.[2] It could also be estimated as part of θ. Algorithm 1 was run for 8000 iterations and its inner

[2] When σ_h is too small, the illuminated region taken into account is smaller than it should be which would cause many

value loop is $n_1 = 50$. Burn-in was observed to have occurred after about 1500 iterations.

4.2.2 Comparison with Weimann et al (2013)

The tracking algorithm of Weimann et al (2013) is a nearest-neighbour method with image pre-processing steps to extract point measurements. Tracks are then created by connecting point measurements nearest to each other. A set of consecutive frames are considered at the same time to allow temporary mis-detections. (See Weimann et al (2013) for more details.) One of the main disadvantages of this heuristic method is that it may miss targets moving close to each other with overlapping illumination regions, as only one point measurement may be extracted from a comparably big bright region. Another disadvantage is the user-defined hard-threshold which may cause the targets with lower intensities to be completely missed. Weimann et al (2013) mentions indeed targets were missed in the marginal regions and Figure 7 also indicates so. Figure 7 compares the tracked positions of the molecules of these two algorithms. As a verification of our result, in the absence of ground truth, we compare in Figure 8 the true and synthesised image (based on our estimated tracks and model parameters) at frame $t = 15$. Their likeness offers some reassurance.

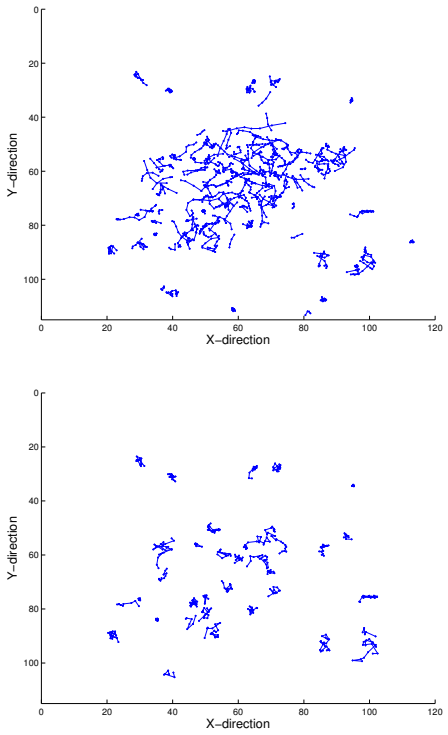

Fig. 7: Real data experiment. Top figure is the tracking result of Alg. 1 (with parameter learning), bottom figure the method in Weimann et al (2013) which appears to miss many tracks.

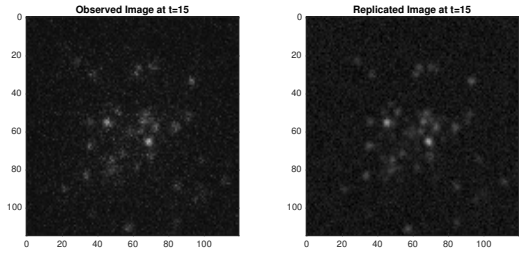

Fig. 8: Left figure is the real observed image at frame $t = 15$. (Expert hand annotated ground truth not available.) Right figure is the synthesised image at frame $t = 15$ based on the inferred tracks and model parameters using Alg. 1 (see Fig. 7 for the tracks.) Their similarity is an indication of correctness of inferred tracks and parameters.

5 Conclusion

We have proposed a new MCMC based MTT algorithm for joint tracking and parameter learning that works directly with image data and avoids the need to pre-process to extract point observations. In numerical examples, we demonstrated improved performance in difficult tracking scenarios involving many targets with overlapping illumination regions, over competing methods (Weimann et al, 2013; Vo et al, 2010), which was achieved by targeting the exact posterior using MCMC. We do not advocate that our MCMC technique should replace an online method or optimised methods for non-overlapping targets such Vo et al (2010). It is an alternative that works without major underlying limiting assumptions (like non-overlapping) and can be used to refine online estimates.

We have constructed all our MCMC moves to obey detailed balance. A measure of the sampler's efficacy is its ability to recover tracks and learn the model parameters in our synthetic data example. Likewise for the real-data experiment, the inferred target locations and model parameters are consistent with the recorded images. However, we have not proven that the chain we have is irreducible for the MTT model. MTT is a very challenging inference problem as it can be viewed as a problem of inferring the states of an unknown number of Hidden Markov Models as each target is a sequence of partially observed Markov states in its own right. Theoretical issues such as identifiability, irreducibility etc are important issues to be pursued in future work. Other possible future works include the design of more efficient proposals for our MCMC routine, parelleliza-

more targets than expected. In that case, we should increase σ_h.

tion and performance optimization for very high density tracking.

A Hypothesis testing

The term of (26) to be calculated is $p(y^r_{t,L(i)}|H_1)$ where $y^r_{t,L(i)} = \{y^r_{t,j}\}_{j\in L(i)}$ and $i \in G_t$ is a pixel that is a local maximum of y^f_t. Let (r, c) be its row and column number and

$$(\bar{a}, \bar{s}) = (y^f_{t,i}, \Delta r, \Delta c).$$

Vector $(\bar{a}, \bar{s})$ can be interpreted as the likely intensity $\bar{a}$ and location $\bar{s}$ of an undetected target. Below we just write L as the set of pixels under consideration instead of $L(i)$.

Let $x = (a, s, v) \in \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2$ where as before a denotes intensity, $s = (s(1), s(2))$ spatial coordinates and $v = (v(1), v(2))$ spatial velocity. (Recall that a pixel illumination is not a function of velocity.) The aim is to calculate

$$\begin{aligned} p(y^r_{t,L}|H_1) &= \int \left(\prod_{j\in L} \mathcal{N}(y^r_{t,j}; a\bar{h}_j(s), \sigma^2_{r,t}) \right) \\ &\qquad \times p(a, s, v|H_1)\mathrm{d}a\mathrm{d}s\mathrm{d}v \\ &= \int p(y^r_{t,L}|a, s)p(a, s|H_1)\mathrm{d}a\mathrm{d}s \end{aligned} \tag{40}$$

where $p(a, s|H_1)$ is either the marginal (or restriction to intensity and spatial position only) of the law of the birth μ_ψ (see (1)) if proposing the initial state of the birth move, or the pdf $p(a, s|a_{0:k}, a_{0:k})$ (to be defined below) if extending the target intensity and position trajectory after having created the initial intensity and position. $p(y^r_{t,L}|a, s)$ is implicitly defined.

Let $p(y^r_{t,L}, a, s|H_1) = p(y^r_{t,L}|a, s)p(a, s|H_1)$. Use the approximation

$$\begin{aligned} \ln p(y^r_{t,L}, a, s|H_1) &\approx \ln p(y^r_{t,L}, \bar{a}, \bar{s}|H_1) \\ &\quad - \frac{1}{2}[(a, s) - (\bar{a}, \bar{s})]D[(a, s) - (\bar{a}, \bar{s})]^T \end{aligned} \tag{41}$$

where $-D$ is the second order derivative $\nabla^2 \ln p(y^r_{t,L}, a, s|H_1)$ evaluated at $(\bar{a}, \bar{s})$. Expression (41) is like the Laplace approximation except that the second order Taylor expansion is computed at $(y^f_{t,i}, \Delta r, \Delta c)$ and not the true maximum $\arg\max_{a,s} p(y^r_{t,L}, a, s|H_1)$ to save on the maximization step, which we find in the numerical examples to be still effective as a component of the birth move. (Moreover, it is a fair simplification for a diffused prior.) Thus

$$p(y^r_{t,L}|H_1) \approx p(y^r_{t,L}, \bar{a}, \bar{s}|H_1)\ \frac{(2\pi)^{3/2}}{\sqrt{|D|}}$$

and the Gaussian distribution in (29) is

$$\mathcal{N}(\cdot|(\bar{a}, \bar{s}), D^{-1}). \tag{42}$$

Sec. 4 (numerical examples) assumes a linear and Gaussian model for the targets in both the synthetic and real data examples. The description is now completed by specifying $p(\bar{a}, \bar{s}|H_1)$.

When the birth move is constructing the initial/first state of the target,

$$p(a, s|H_1) = \int \mu_\psi(a, s, v)\mathrm{d}v \tag{43}$$

and the expression in (26) now simplifies to

$$\rho(y^r_{t,L(i)}) = \frac{p(H_1)}{p(H_0)} \frac{p(y^r_{t,L}|\bar{a}, \bar{s})}{p(y^r_{t,L}|H_0)} p(\bar{a}, \bar{s}|H_1) \frac{(2\pi)^{3/2}}{\sqrt{|D|}}. \tag{44}$$

Finally, we derive $p(a, s|H_1)$ in (40) when the birth move is extending the target intensity and position trajectory after having created the initial intensity position pairs $(a_0, s_0), \ldots, (a_{k-1}, s_{k-1})$ for some $k \geq 1$. For a target with a linear Gaussian model (1), the pdf of v_{k-1} (or $v_{0:k-1}$) conditioned on $s_{0:k-1}$, which is denoted $p_\psi(v_{k-1}|s_{0:k-1})$, is a Gaussian. Thus $p(a, s|H_1)$ is

$$\int f_\psi(a, s, v|a_{k-1}, s_{k-1}, v_{k-1}) p_\psi(v_{k-1}|s_{0:k-1})\mathrm{d}v\mathrm{d}v_{k-1}$$

and the corresponding expression for $\rho(y^r_{t_k,L(i)})$ in (30) is the same as in (44).

B State proposal

The state proposal of Sec. 3.7 alters the state values of the targets whose trajectories have been partially exchanged. This proposal is defined for the Gaussian model in Sec. 4.

Let $\{(k, t^k_b, \mathbf{x}^k)\}_{k=1}^K$ be the MTT state and assume without loss of generality $U = \{1, 2\}$. The state proposal is decomposed into

$$\begin{aligned} &q_{s,\theta}(U, t, \tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2|\mathbf{x}^{1:K}, t^{1:K}_b, y_{1:n}) \\ &= q_{s,\theta}(U, t|\mathbf{x}^{1:K}, t^{1:K}_b) q_{s,\theta}(\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2|\mathbf{x}^{1:K}, t^{1:K}_b, y_{1:n}, U, t) \end{aligned}$$

where the first term is the probability of selecting (U, t) and is not $y_{1:n}$ dependent. Using $y_{1:n}$ and $\{(k, t^k_b, \mathbf{x}^k)\}_{k=3}^K$, generate the residual image as in (23) by subtracting the background intensity and the contribution from all targets except $(t^1_b, \mathbf{x}^1)$ and $(t^2_b, \mathbf{x}^2)$. Let $y^r_t = (y^r_{t,1}, \ldots, y^r_{t,m})$ denote the residual images at time t. The state proposal samples $(\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2)$ from the pdf $q_{s,\theta}(\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2|\hat{\mathbf{x}}^{1:2}, \tau^{1:2}_b, y^r_{1:n})$ where $(\hat{\mathbf{x}}^1, \hat{\mathbf{x}}^2)$ is defined in (33). Note the dependancy on targets $k > 2$ is captured through the residual image.

For brevity, instead of $(\hat{\mathbf{x}}^1, \hat{\mathbf{x}}^2)$, we write $(\mathbf{x}^1, \mathbf{x}^2)$. Also, to highlight the intensity, spatial coordinates and velocity components, $\mathbf{x}^i = (a^i_0, s^i_0, v^i_0, \ldots, a^i_{l^i-1}, s^i_{l^i-1}, v^i_{l^i-1})$ is written as $(\mathbf{a}^i, \mathbf{s}^i, \mathbf{v}^i)$.

The proposal $q_{s,\theta}$ does not alter the spatial components, i.e. $\tilde{\mathbf{s}}^1 = \mathbf{s}^1$ and $\tilde{\mathbf{s}}^2 = \mathbf{s}^2$.

The velocities $(\tilde{\mathbf{v}}^1, \tilde{\mathbf{v}}^2) \sim p_\psi(\tilde{\mathbf{v}}^1|\mathbf{s}^1)p_\psi(\tilde{\mathbf{v}}^2|\mathbf{s}^2)$, i.e. are sampled independently from p_ψ, which is the prior pdf of the velocity conditioned on the spatial coordinates values, which is Gaussian.

If targets 1 and 2 exist at time t then their state values are $x^1_{t-\tau^1_b}$ and $x^2_{t-\tau^2_b}$ and respectively. We assume (for pixel i) $y^r_{t,i} \sim \mathcal{N}(\cdot|0, \sigma^2_{r,t})$ if targets 1 and 2 both do not exist at time t, $y^r_{t,i} \sim \mathcal{N}(\cdot|a^1_{t-\tau^1_b} h_i(s^1_{t-\tau^1_b}), \sigma^2_{r,t})$ if only target 1 exists and $y^r_{t,i} \sim \mathcal{N}(\cdot|a^1_{t-\tau^1_b} h_i(s^1_{t-\tau^1_b}) + a^2_{t-\tau^2_b} h_i(s^2_{t-\tau^2_b}), \sigma^2_{r,t})$ if both exists. The prior probability model (see (1)) for the intensities are independent Gaussians, denoted $p_\psi(\mathbf{a}^1)p_\psi(\mathbf{a}^2)$. Conditioned on $y^r_{1:n}$, $(\tau^1_b, \mathbf{s}^1)$ and $(\tau^2_b, \mathbf{s}^2)$, the posterior pdf for the joint intensities is also a Gaussian, which is denoted by $q_{s,\theta}(\tilde{\mathbf{a}}^1, \tilde{\mathbf{a}}^2|\mathbf{s}^{1:2}, \tau^{1:2}_b, y^r_{1:n})$. Thus

$$\begin{aligned} q_{s,\theta}(\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2|\mathbf{x}^{1:2}, \tau^{1:2}_b, y^r_{1:n}) &= p_\psi(\tilde{\mathbf{v}}^1|\mathbf{s}^1)p_\psi(\tilde{\mathbf{v}}^2|\mathbf{s}^2) \\ &\quad \times q_{s,\theta}(\tilde{\mathbf{a}}^1, \tilde{\mathbf{a}}^2|\mathbf{s}^{1:2}, \tau^{1:2}_b, y^r_{1:n}). \end{aligned}$$

C CSMC step of Algorithm 1

(Initialisation.) Denote target k's trajectory by $(t_b^k, \mathbf{x}^k)$ with $\mathbf{x}^k = (x_0, \ldots, x_{l-1})$. Let $\mathbf{x}'_{1:n}$ denote the MTT state, in the representation of Sec. 2.3, omitting target k. Define target k's state likelihood to be

$$g_t(x) = \prod_{i=1}^{m} \mathcal{N}(y_{t,i}|h_i(\mathbf{x}'_t) + h_i(x) + b_t, \sigma^2_{r,t})$$

(c.f. (11).)

Execute the CSMC algorithm of Whiteley (2010), with backward sampling, for the partially observed Markov model with state law $\mu_\psi(x_0)$, $f_\psi(x_t|x_{t-1})$ (see (1)), state likelihood $g_{t_b^k+t}(x_t)$ above and input trajectory $(x_0, \ldots, x_{l-1})$ to yield an updated target k trajectory $(x'_0, \ldots, x'_{l-1})$. The CSMC implementation used proposes particles from the state law.

D Gibbs step of Algorithm 1 for the MTT model of Sec. 4

Recall K_t^s is a Binomial r.v. with success probability p_s and number of trials K_{t-1}^x. K_t^b is a Poisson r.v. with rate λ_b. Thus their posteriors are the Beta and Gamma pdfs:

$$p_s|z_{1:n}, \mathbf{x}_{1:n}, y_{1:n} \sim \mathcal{B}\Big(1 + \sum_{t=1}^{n} k_t^s,\ 1 + \sum_{t=2}^{n}(k_{t-1}^x - k_t^s)\Big),$$

$$\lambda_b|z_{1:n}, \mathbf{x}_{1:n}, y_{1:n} \sim \mathcal{G}\Big(\alpha_0 + \sum_{t=1}^{n} k_t^b,\ (\beta_0^{-1} + n)^{-1}\Big).$$

The updates for b_t and $\sigma_{r,t}$ require the following empirical means and variances (c.f. (10), (11)): $\bar{e}_t = m^{-1}\sum_{i=1}^m y_{t,i} - h_i(\mathbf{x}_t)$, $\beta_1 = \sum_{i=1}^m (y_{t,i} - h_i(\mathbf{x}_t) - \bar{e}_t)^2$, $\beta_2 = \frac{n_0 m}{n_0+m}(\mu_0 - \bar{e}_t)^2$,

$$\sigma^2_{r,t}|z_{1:n}, \mathbf{x}_{1:n}, y_{1:n} \sim \mathcal{IG}(\alpha_0 + m/2, \beta_0 + \beta_1/2 + \beta_2/2),$$

$$b_t|\sigma^2_{r,t}, z_{1:n}, \mathbf{x}_{1:n}, y_{1:n} \sim \mathcal{N}(\frac{n_0\mu_0 + m\bar{e}_t}{n_0+m}, \frac{\sigma^2_{r,t}}{n_0+m}).$$

The Gibbs update for means and variances of the remaining Gaussians of the MTT model of Sec. 4 follow similar update formulae.

Acknowledgement

We thank Kristina Ganzinger and Professor David Klenerman for providing the real data and the code in Weimann et al (2013) for the comparisons in Section 4.2, and Sinan Yıldırım for his careful reading of this paper.

References

Andrieu C, Doucet A, Holenstein R (2010) Particle Markov chain monte carlo methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72(3):269–342

Bar-Shalom Y, Fortmann TE (1988) Tracking and Data Association. Academic Press, Boston

Boers Y, Driessen J (2004) Multitarget particle filter track before detect application. IEE Proceedings-Radar, Sonar and Navigation 151(6):351–357

Davey SJ, Rutten MG, Cheung B (2007) A comparison of detection performance for several track-before-detect algorithms. EURASIP Journal on Advances in Signal Processing 2008

Del Moral P, Doucet A, Jasra A (2006) Sequential monte carlo samplers. Journal of the Royal Statistical Society 68:411–436

Doucet A, Johansen AM (2009) A tutorial on particle filtering and smoothing: Fifteen years later. Handbook of Nonlinear Filtering 12:656–704

Duckworth D (2012) Monte carlo methods for multiple target tracking and parameter estimation. Master's thesis, EECS Department, University of California, Berkeley, URL http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-68.html

Green PJ (1995) Reversible jump Markov chain monte carlo computation and Bayesian model determination. Biometrika 82(4):711–732

Jiang L, Singh S, Yildirim S (2014) A new particle filtering algorithm for multiple target tracking with non-linear observations. In: Information Fusion (FUSION), 2014 17th International Conference on, pp 1–8

Jiang L, Singh S, Yıldırım S (2015) Bayesian tracking and parameter learning for non-linear multiple target tracking models. IEEE Tran. Signal Proc 63:5733–5745

Kantas N, Doucet A, Singh SS, Maciejowski J, Chopin N (2015) On particle methods for parameter estimation in state-space models. Statist Sci 30(3):328–351, DOI 10.1214/14-STS511

Kokkala J, Sarkka S (2015) Combining particle MCMC with Rao-Blackwellized monte carlo data association for parameter estimation in multiple target tracking. Digital Signal Processing 47:84 – 95

Mahler RP (2007) Statistical Multisource-Multitarget Information Fusion. Artech House, Boston

Oh S, Russell S, Sastry S (2009) Markov chain monte carlo data association for multi-target tracking. IEEE Trans Automat Control 54(3):481–497

Papi F, Kim DY (2015) A particle multi-target tracker for superpositional measurements using labeled random finite sets. IEEE Transactions on Signal Processing 63(16):4348–4358

Punithakumar K, Kirubarajan T, Sinha A (2005) A sequential monte carlo probability hypothesis density algorithm for multitarget track-before-detect. In: Optics & Photonics 2005, International Society for Optics and Photonics, pp 59,131S–59,131S

Rezatofighi SH, Gould S, Vo BT, Vo BN, Mele K, Hartley R (2015) Multi-target tracking with time-varying clutter rate and detection profile: Application to time-lapse cell microscopy sequences. IEEE Transactions on Medical Imaging 34(6):1336–1348

Rutten MG, Gordon NJ, Maskell S (2005) Recursive track-before-detect with target amplitude fluctuations. IEE Proceedings - Radar, Sonar and Navigation 152(5):345–352

Schlangen I, Franco J, Houssineau J, Pitkeathly WTE, Clark D, Smal I, Rickman C (2016) Marker-less stage drift correction in super-resolution microscopy using the single-cluster PHD filter. IEEE Journal of Selected Topics in Signal Processing 10(1):193–202

Singh SS, Whiteley N, Godsill SJ (2011) Approximate likelihood estimation of static parameters in multi-target models. In: Barber D, Cemgil AT, Chiappa S (eds) Bayesian Time Series Models, Cambridge University Press, pp 225–244

Streit RL, Graham ML, Walsh MJ (2002) Multitarget tracking of distributed targets using histogram PMHT. Digital Signal Processing 12(2):394–404

Vo BN, Vo BT, Pham NT, Suter D (2010) Joint detection and estimation of multiple objects from image observations. Signal Processing, IEEE Transactions on 58(10):5129–5141

Vu T, Vo BN, Evans R (2014) A particle marginal Metropolis-Hastings multi-target tracker. IEEE Trans Signal Process 62(15):3953 – 3964

Weimann L, Ganzinger KA, McColl J, Irvine KL, Davis SJ, Gay NJ, Bryant CE, Klenerman D (2013) A quantitative comparison of single-dye tracking analysis tools using monte carlo simulations. PloS one 8(5):e64,287

Whiteley N (2010) Discussion of particle Markov chain monte carlo methods by Andrieu, Doucet and Holenstein. JRSSB 72(3)

Yıldırım S, Jiang L, Singh SS, Dean TA (2014) Calibrating the Gaussian multi-target tracking model. Statistics and Computing pp 1–14