Adaptive sequential Monte Carlo for multiple changepoint analysis

Nicholas A. HEARD
Department of Mathematics, Imperial College London
& Heilbronn Institute for Mathematical Research
and
Melissa J. M. TURCOTTE*
Advanced Research in Cyber Systems, Los Alamos National Laboratory

April 19, 2016

Abstract

Process monitoring and control requires detection of structural changes in a data stream in real time. This article introduces an efficient sequential Monte Carlo algorithm designed for learning unknown changepoints in continuous time. The method is intuitively simple: new changepoints for the latest window of data are proposed by conditioning only on data observed since the most recent estimated changepoint, as these observations carry most of the information about the current state of the process. The proposed method shows improved performance over the current state of the art.

Another advantage of the proposed algorithm is that it can be made adaptive, varying the number of particles according to the apparent local complexity of the target changepoint probability distribution. This saves valuable computing time when changes in the changepoint distribution are negligible, and enables re-balancing of the importance weights of existing particles when a significant change in the target distribution is encountered.

The plain and adaptive versions of the method are illustrated using the canonical continuous time changepoint problem of inferring the intensity of an inhomogeneous Poisson process, although the method is generally applicable to any changepoint problem. Performance is demonstrated using both conjugate and non-conjugate Bayesian models for the intensity. Appendices to the article are available online, illustrating the method on other models and applications.

Keywords: Adaptive sample size; Particle filters; Online inference; Markov chain Monte Carlo methods

*The second author gratefully acknowledges support from the EPSRC and the Institute for Security Science and Technology, Imperial College London.

1 Introduction

Let $\{y(t) : t \geq 0\}$ be a continuous time stochastic process of observable data on $\mathbb{R}^+$, where the law of $y(t)$ is governed by a second underlying stochastic process $\theta(t) \in \Theta$. A changepoint model for y assumes an unknown number of changepoints $\tau_1 < \tau_2 < \ldots$ that partition $\mathbb{R}^+$ into disjoint, homogeneous segments $[\tau_i, \tau_{i+1})$ such that $\forall t \in [\tau_i, \tau_{i+1}), \theta(t) = \theta_i$ for some constant parameter value θ_i. Changepoint detection is the task of making inference about the occurrence of changepoints in the underlying process $\theta(t)$ based on the observable process $y(t)$.

The motivation behind the sequential Monte Carlo (SMC) algorithm proposed here is detecting changepoints in continuous time, previously considered by Whiteley et al. (2011) under the name *piecewise deterministic processes*. The process data $y(\cdot)$ are assumed to arrive as a continuous stream, while inferences about $\theta(\cdot)$ are made at a discrete sequence of observation times $0 < t_1 < t_2 < \ldots$. At each observation time t_n, $n \geq 1$, the posterior distribution for changepoints in the interval $[0, t_n]$ is sought in order to make such inferences.

The SMC algorithm of Whiteley et al. (2011) is a direct application of the SMC samplers methodology of Del Moral et al. (2006), which is a more general SMC technique for sampling sequentially from *any* sequence of target distributions defined on a common space with a fixed computational cost. The generality of the SMC samplers method is achieved by augmenting the target distributions to ever increasing dimensions in order to avoid the need to integrate over a general transition kernel; whilst that provides the basis for a very general class of samplers, applicable in a broad variety of contexts, the aim of this article is to propose a more bespoke SMC algorithm designed specifically for changepoint analysis.

Further adaptation of the SMC sampler methodology to point process models was provided by Martin et al. (2013), and a fixed computational cost implementation of their method is also included for comparison in this article. In the wider literature of SMC for changepoint detection, but within the context of discrete time changepoint analysis, Fearnhead and Liu (2007) make sequential inference on data where filtering recursions are used to sample exactly from the distribution of the most recent changepoint, and consequently the joint distribution of all changepoints; the computational cost of exact simulation increases linearly with time and so an approximation using particle filtering is proposed. Chopin (2007) and Fearnhead and Clifford (2003) approach discrete changepoint detection in time series by reformulating the changepoint problem as a hid-

den Markov model and use particle filters to propagate forward the distribution of the time since the most recent changepoint. Carrying forward inferences about the most recent changepoint is also an important element of the proposed method.

The next section outlines the Bayesian changepoint model. Section 3 proposes an efficient new SMC algorithm for changepoint problems; Section 4 demonstrates how this algorithm can easily be made adaptive, automatically varying the number of particles according to the complexity of the target, which can potentially be valuable when performing inference on a number of sequences of target distributions in parallel.

2 Bayesian changepoint model

In all of the examples in the main text of this article, the arrivals of changepoints in $\theta(t)$ will be assumed to be a homogeneous Poisson process with constant intensity $\nu \in \mathbb{R}^+$.

Following the notation of Whiteley et al. (2011), let k_n be the number of changepoints over $[0, t_n]$. When $k_n > 0$, let $\tau_{1:k_n} = (\tau_1, \ldots, \tau_{k_n}) \in \mathbb{T}_{n,k_n}$ denote the ordered locations of these changepoints, where $\mathbb{T}_{n,k_n} = \{\tau_{1:k_n} : 0 < \tau_1 < \ldots < \tau_{k_n} < t_n\}$, and let $\tau_0 = 0$ and $\tau_{k_n+1} = t_n$.

For a concise notation, define $\boldsymbol{\tau}_n = (0)$ if $k_n = 0$, and $\boldsymbol{\tau}_n = (k_n, \tau_{1:k_n})$ otherwise; the prior density for $\boldsymbol{\tau}_n$ implied by the Poisson process is

$$p_{[0,t_n]}(\boldsymbol{\tau}_n) = \nu^{k_n} \exp(-\nu t_n) \mathbb{I}_{E_n}(\boldsymbol{\tau}_n), \tag{1}$$

where (E_n) is the sequence of nested transdimensional spaces, $E_n \subset E_{n+1}$, for which

$$E_n = \bigcup_{k_n=0}^{\infty} \{k_n\} \times \mathbb{T}_{n,k_n}.$$

The appendices to this article in the Supplementary Material consider a generalization of (1),

$$p_{[0,t_n]}(\boldsymbol{\tau}_n | s) = \nu^{k_n} \exp\left(-\nu t_n + \nu \sum_{i=1}^{n} \min\{s, t_n - \tau_i\}\right) \mathbb{I}_{[0,t_n]}(\tau_1) \prod_{i=1}^{n-1} \mathbb{I}_{(\tau_i+s,t_n]}(\tau_{i+1}) \tag{2}$$

for $s \geq 0$. This prior enforces a waiting time of at least s after each changepoint event before resuming the constant intensity ν. Note that $p_{[0,t_n]}(\boldsymbol{\tau}_n) \equiv p_{[0,t_n]}(\boldsymbol{\tau}_n | s = 0)$, but for $s > 0$ this

prior places a finite bound on the number of changepoints in any finite interval.

Let $\boldsymbol{\theta}_n = \theta_{0:k_n} = (\theta_0, \ldots, \theta_{k_n}) \in \Theta^{k_n+1}$ be the vector of parameters corresponding to the $k_n + 1$ segments of the partition of $[0, t_n]$ created by the changepoints in $\boldsymbol{\tau}_n$, with prior density $p(\boldsymbol{\theta}_n|\boldsymbol{\tau}_n)$. Over a time interval $I \subset \mathbb{R}^+$, let $y(I)$ be the path of data observed. Assuming a likelihood function $f(y([0, t_n])|\boldsymbol{\tau}_n, \boldsymbol{\theta}_n)$ for the data over $[0, t_n]$ which is known pointwise, the joint density of the changepoints, the parameters and the sample path is

$$\gamma_{[0,t_n]}(\boldsymbol{\tau}_n, \boldsymbol{\theta}_n, y([0, t_n])) = f(y([0, t_n])|\boldsymbol{\tau}_n, \boldsymbol{\theta}_n)\, p(\boldsymbol{\theta}_n|\boldsymbol{\tau}_n)\, p_{[0,t_n]}(\boldsymbol{\tau}_n). \tag{3}$$

The posterior density for the changepoints and parameters is then known up to proportionality by

$$\pi_{[0,t_n]}(\boldsymbol{\tau}_n, \boldsymbol{\theta}_n|y([0, t_n])) \propto \gamma_{[0,t_n]}(\boldsymbol{\tau}_n, \boldsymbol{\theta}_n, y([0, t_n])). \tag{4}$$

If the conditional density of the parameters given the changepoints is the conjugate prior to the likelihood model, then the parameters $\boldsymbol{\theta}_n$ in (3) can be integrated out to give the marginal posterior density for the changepoints up to proportionality,

$$\pi_{[0,t_n]}(\boldsymbol{\tau}_n|y([0, t_n])) \propto \gamma_{[0,t_n]}(\boldsymbol{\tau}_n, y([0, t_n])) = \int_{\Theta^{k_n+1}} \gamma_{[0,t_n]}(\boldsymbol{\tau}_n, \boldsymbol{\theta}_n, y([0, t_n]))d\boldsymbol{\theta}_n. \tag{5}$$

3 Sequential Monte Carlo algorithm for changepoint distributions

For a fast SMC algorithm for changepoint problems, the proposal distributions for the sequence of target distributions $\pi_{[0,t_n]}$, $n = 1,2,\ldots$, will sample changepoints from approximations to the posterior distributions for each update interval $(t_{n-1}, t_n]$ without reference to the sampled changepoints from the previous intervals. Instead, information about changepoints in $[0, t_{n-1}]$ will be summarized through some convenient estimate t^*_{n-1} of $\tau_{k_{n-1}}$, the most recent changepoint (*cf.* Chopin, 2007; Fearnhead and Clifford, 2003). New changepoints will then be proposed within the update interval $(t_{n-1}, t_n]$ based on the process data from $(t^*_{n-1}, t_n]$.

Here t^*_{n-1} will usually be the posterior mean of $\tau_{k_{n-1}}$ estimated from the weighted SMC sample at t_{n-1}. Other choices are possible: the straightforward choice of $t^*_{n-1} = t_{n-1}$ was also

tested, but led to reduced performance in all examples. The extreme case of $t^*_{n-1} = t_n$ will also be considered, since this corresponds to proposing changepoints from the prior and thus provides a fixed-computation cost implementation of the method of Martin et al. (2013).

The motivation behind the proposed algorithm is as follows: If new changepoints within $(t_{n-1}, t_n]$ are to be sampled and appended to the existing changepoint vectors for each particle, then to construct a good proposal distribution it might be sufficient to retain only those data that have been observed since the last changepoint in $[0, t_n]$, as these data provide all of the available information on the current state of $\theta(t)$ as the process enters $(t_{n-1}, t_n]$. The diversity of the particles in terms of their earlier changepoints in $[0, t_{n-1}]$ has no future bearing.

Let $\tilde{k}_n \leq k_n$ be a random variable for the number of changepoints in the *update interval* $(t_{n-1}, t_n]$. Further let $\tilde{\boldsymbol{\tau}}_n = (0)$ if $\tilde{k}_n = 0$ and $\tilde{\boldsymbol{\tau}}_n = (\tilde{k}_n, \tilde{\tau}_{1:\tilde{k}_n})$ otherwise, so that $\tilde{\boldsymbol{\tau}}_n \in \tilde{E}_n$ where,

$$\tilde{E}_n = \bigcup_{\tilde{k}_n=0}^{\infty} \{\tilde{k}_n\} \times \{\tilde{\tau}_{1:\tilde{k}_n} : t_{n-1} < \tilde{\tau}_1 < \ldots < \tilde{\tau}_{\tilde{k}_n} < t_n\}$$

Let $\pi_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n | y((t^*_{n-1}, t_n])))$ be the local posterior distribution of changepoints in $\tilde{E}_n$ obtained when conditioning on the data observed in the extended interval $(t^*_{n-1}, t_n]$. Then

$$\pi_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n | y((t^*_{n-1}, t_n]))) \propto$$
$$\gamma_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n, y((t^*_{n-1}, t_n]))) = \mathbb{I}_{\tilde{E}_n}(\tilde{\boldsymbol{\tau}}_n) \int_{\Theta^{\tilde{k}_n+1}} f\left(y([t^*_{n-1}, t_n]) | \tilde{\boldsymbol{\tau}}_n, \tilde{\boldsymbol{\theta}}_n\right) p\left(\tilde{\boldsymbol{\theta}}_n | \tilde{\boldsymbol{\tau}}_n\right) p_{[t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n) d\tilde{\boldsymbol{\theta}}_n.$$

In general, sampling directly from the densities $\pi_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n | y((t^*_{n-1}, t_n])))$ will not be possible, except in the special case where $t^*_{n-1} = t_n$ (Martin et al., 2013). Instead, a single reversible jump Markov chain Monte Carlo (RJMCMC) (Green, 1995) sampler with target density $\pi_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n | y((t^*_{n-1}, t_n])))$ will provide approximate draws within the SMC algorithm. Importantly, it is likely that convergence of RJMCMC on the subintervals $(t_{n-1}, t_n]$ will be very fast if the update intervals are small, as the prior probability that there would be more than one changepoint is $o(t_n - t_{n-1})$. To demonstrate the validity of the approximation, the appendices in the Supplementary Material present a comparison of RJMCMC against exact (rejection) sampling within the SMC algorithm, which is possible when adopting the alternative model prior (2). The

results from the two approaches are indistinguishable.

Extending a vector of changepoints in the case of a conjugate Bayesian model, with target density (5), is the most straightforward case and will be considered first. Second, the non-conjugate case will be addressed, with extended target density (4). This latter case is more difficult for two reasons: firstly, sampling from more highly parameterized models is generally more cumbersome and inefficient; secondly and more importantly, when appending changepoint vectors from the update interval there is a spare *intercept* parameter that needs to be handled.

3.1 Conjugate models

Algorithm 1 presents the most straightforward form of the SMC algorithm being proposed, when assuming a conjugate Bayesian model for $y(t)$ within each changepoint segment.

Algorithm 1 SMC algorithm for changepoint detection

1: Set $n = 1$, and $w_1^{(i)} = 1$ for $i = 1, \ldots, N$
2: Sample $\{\boldsymbol{\tau}_1^{(i)}\}_{i=1}^N \sim \pi_{[0,t_1]}(\boldsymbol{\tau}_1|y([0,t_1]))$ (either approximately via RJMCMC, or exactly)
3: $n \leftarrow n + 1$
4: Calculate $t_{n-1}^* = \sum_{i=1}^N w_{n-1}^{(i)} \tau_{k_{n-1}}^{(i)} / \sum_{i=1}^N w_{n-1}^{(i)}$
5: Sample $\{\tilde{\boldsymbol{\tau}}_n^{(i)}\}_{i=1}^N \sim \pi_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n|y((t_{n-1}^*,t_n]))$ (either approximately via RJMCMC or exactly)
6: Draw a random permutation σ uniformly from S_N, the symmetric group on N symbols
7: Combine the particles, $\{\boldsymbol{\tau}_n^{(i)} = (k_{n-1}^{(i)} + \tilde{k}_n^{(i)}, \tau_1^{(i)}, \ldots, \tau_{k_{n-1}}^{(i)}, \tilde{\tau}_1^{\sigma(i)}, \ldots, \tilde{\tau}_{\tilde{k}_n}^{\sigma(i)})\}_{i=1}^N$
8: Update the importance weights $\{w_n^{(i)}\}_{i=1}^N$ according to (6)
9: Calculate the effective sample size $\text{ESS} = \sum_{i=1}^N {w_{n-1}^{(i)}}^2 / (\sum_{i=1}^N w_{n-1}^{(i)})^2$
10: **if** $\text{ESS} < \text{ESS}_{\min} = N/3$ **then**
11: Resample $\{\boldsymbol{\tau}_n^{(i)}, w_n^{(i)}\}_{i=1}^N$ according to the weights to obtain unweighted particles $\{\boldsymbol{\tau}_n^{(i)}, 1\}_{i=1}^N$
12: Optionally move each particle according to the $\pi_{[0,t_n]}$-invariant kernel used for RJMCMC sampling, and retain the same importance weights
13: **end if**
14: **goto** 3

At time t_{n-1}, the algorithm assumes a set of N importance weighted particles $\{\boldsymbol{\tau}_{n-1}^{(i)}, w_{n-1}^{(i)}\}_{i=1}^N$ that approximate $\pi_{[0,t_{n-1}]}(\boldsymbol{\tau}_{n-1}|y([0,t_n]))$. Let $t_{n-1}^* = \sum_{i=1}^N w_{n-1}^{(i)} \tau_{k_{n-1}}^{(i)} \Big/ \sum_{i=1}^N w_{n-1}^{(i)}$ be the Monte Carlo estimate of the posterior mean of $\tau_{k_{n-1}}$. Then at time t_n, N new sub-particles are sampled from $\pi_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n|y((t_{n-1}^*,t_n]))$, either approximately via RJMCMC, or by exact sampling

if possible. Then setting $\boldsymbol{\tau}_n = (\boldsymbol{\tau}_{n-1}, \tilde{\boldsymbol{\tau}}_n)$, the implied importance distribution on $[0, t_n]$ from combining the two sets of particles is known up to proportionality through

$$q_{[0,t_n]}(\boldsymbol{\tau}_n) = \gamma_{[0,t_1]}(\boldsymbol{\tau}_1, y([0,t_1])) \prod_{j=2}^{n} \gamma_{(t_{j-1},t_j]}(\tilde{\boldsymbol{\tau}}_j, y((t^*_{j-1}, t_j])).$$

When RJMCMC is used to draw approximate samples from $\pi_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n | y((t^*_{n-1}, t_n]))$ in each update, there will be autocorrelation in each batch of samples. To negate this, before joining the particles together step 6 of the algorithm samples a random permutation for the labels of the sample from the new interval, to break the autocorrelation of the combined particles.

The importance weights are then updated to account for the discrepancy between the importance distribution and the changepoint posterior distribution (5). For $n > 1$ the weight for the ith particle is given by

$$w_n(\boldsymbol{\tau}_n^{(i)}) = \frac{\gamma_{[0,t_n]}(\boldsymbol{\tau}_n^{(i)}, y([0,t_n]))}{q_{[0,t_n]}(\boldsymbol{\tau}_n^{(i)})} = w_{n-1}(\boldsymbol{\tau}_{n-1}^{(i)}) \overline{w}_n(\boldsymbol{\tau}_n^{(i)}), \tag{6}$$

where the incremental weight

$$\overline{w}_n(\boldsymbol{\tau}_n^{(i)}) = \frac{\gamma_{[0,t_n]}(\boldsymbol{\tau}_n^{(i)}, y([0,t_n]))}{\gamma_{[0,t_{n-1}]}(\boldsymbol{\tau}_{n-1}^{(i)}, y([0,t_{n-1}])) \gamma_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n^{(i)}, y((t^*_{n-1}, t_n]))}, \tag{7}$$

is computationally simple to calculate; for example, in the case of $t^*_{n-1} = t_{n-1}$ this is analogous to calculating the probability of accepting an RJMCMC death move for a changepoint at t_{n-1}. Again it should be noted that, strictly, (6) and (7) are only approximate formulas when the update samples have been generated by RJMCMC. However, a comparison with exact sampling presented in the appendices in the Supplementary Material demonstrates any bias introduced by this approximation to appear negligible.

Note that the entire SMC scheme is equivalent to sequential importance sampling on a sequence of distributions defined on nested spaces with a transition kernel approximating

$$K_n(\boldsymbol{\tau}_n | \boldsymbol{\tau}'_{n-1}) \propto \delta_{\boldsymbol{\tau}'_{n-1}}(\boldsymbol{\tau}_{n-1}) \gamma_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n, y((t^*_{n-1}, t_n])).$$

When the *effective sample size* (ESS) (Liu, 1996) of the particles drops below a threshold,

commonly taken to be $N/3$, the systematic resampling approach (Kitagawa, 1996) is used (Algorithm 1, Step 9 to 11). Additionally, a sweep of RJMCMC moves is applied to the particle set after resampling (Algorithm 1, Step 12), as adopted in Del Moral et al. (2006), Whiteley et al. (2011) and Martin et al. (2013).

3.2 Non-conjugate models

The SMC algorithm for non-conjugate models follows the steps of Algorithm 1, with the exception that the marginal posteriors $\pi_{[0,t_n]}(\boldsymbol{\tau}_n|y([0,t_n]))$ from (5) are unavailable. Changepoints can be sampled approximately at each update via RJMCMC, but from the joint distribution of changepoints and parameters $\pi_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n, \tilde{\boldsymbol{\theta}}_n|y((t^*_{n-1},t_n]))$.

Suppose at t_{n-1} a weighted sample of changepoints and parameters $\{\boldsymbol{\tau}^{(i)}_{n-1}, \boldsymbol{\theta}^{(i)}_{n-1}, w^{(i)}_{n-1}\}_{i=1}^N$ has been obtained from $\pi_{[0,t_{n-1}]}(\boldsymbol{\tau}_{n-1}, \boldsymbol{\theta}_{n-1}|y([0,t_{n-1}]))$, and subsequently at time t_n a sample $\{\tilde{\boldsymbol{\tau}}^{(i)}_n, \tilde{\boldsymbol{\theta}}^{(i)}_n\}_{i=1}^N$ is drawn from $\pi_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n, \tilde{\boldsymbol{\theta}}_n|y((t^*_{n-1},t_n]))$. Combining the particles from these two samples is less straightforward, as there is an extra, redundant parameter for $\theta(t)$ for the segment $(\tau_{k_{n-1}}, \tilde{\tau}_1]$. The implied proposal distribution would be over-parameterized, so the parameter pair $(\theta_{k_{n-1}}, \tilde{\theta}_0)$ needs to be combined to form a single parameter θ^*_n for $(\tau_{k_{n-1}}, \tilde{\tau}_1]$.

Let $s_1(\theta_{k_{n-1}}, \tilde{\theta}_0)$ be a suitably chosen function to combine the two model parameters into a single value θ^*_n. As the marginal distribution of θ^*_n implied by the proposal density and s_1 is unlikely to have an analytic solution, a joint change of variables is required. Let $s_2(\theta_{k_{n-1}}, \tilde{\theta}_0)$ be a second transformation such that the pair

$$(\theta^*_n, u_{n-1}) = s(\theta_{k_{n-1}}, \tilde{\theta}_0) = (s_1(\theta_{k_{n-1}}, \tilde{\theta}_0), s_2(\theta_{k_{n-1}}, \tilde{\theta}_0))$$

comprise a one to one mapping $(\theta_{k_{n-1}}, \tilde{\theta}_0) \mapsto (\theta^*_n, u_{n-1})$, and let $|\mathrm{J}_s|$ be the determinant of the Jacobian of s. Following the change of variable s, the implied proposal density is known up to proportionality and satisfies

$$q_{[0,t_n]}(\boldsymbol{\tau}_n, \boldsymbol{\theta}_n, u_{1:n-1}) = q_{[0,t_{n-1}]}(\boldsymbol{\tau}_{n-1}, \boldsymbol{\theta}_{n-1}, u_{1:n-2})\pi_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n, \tilde{\boldsymbol{\theta}}_n, y((t^*_{n-1},t_n]))|\mathrm{J}_s|,$$

where $\boldsymbol{\theta}_n = (\theta_0, \ldots, \theta_{k_{n-1}-1}, \theta^*_n, \tilde{\theta}_1, \ldots, \tilde{\theta}_{\tilde{k}})$. This proposal density generates parameters suited to the concatenated changepoints, but also the nuisance parameters $u_{1:n-1}$. To accommodate

these nuisance parameters, a general solution is to extend the target distribution,

$$\pi_{[0,t_n]}(\boldsymbol{\tau}_n, \boldsymbol{\theta}_n, u_{1:n-1}|y([0,t_n])) \triangleq \pi_{[0,t_n]}(\boldsymbol{\tau}_n, \boldsymbol{\theta}_n|y([0,t_n])) \prod_{j=1}^{n-1} \tilde{\pi}(u_j|\boldsymbol{\tau}_j, \boldsymbol{\theta}_j) \tag{8}$$

where $\tilde{\pi}$ can be *any* density with the correct support for u_j. As the true target (4) is a marginal of (8), standard importance sampling estimates obtained in the extended space can still be used to give an approximation for the true target distribution and its normalizing constant.

The importance weights given in (6) for the non-conjugate case with this extended target are

$$\begin{aligned} w_n(\boldsymbol{\tau}_n^{(i)}, \boldsymbol{\theta}_n^{(i)}, u_{1:n-1}^{(i)}) &= \frac{\gamma_{[0,t_n]}(\boldsymbol{\tau}_n^{(i)}, \boldsymbol{\theta}_n^{(i)}, u_{1:n-1}^{(i)}, y([0,t_n]))}{q_{[0,t_n]}(\boldsymbol{\tau}_n^{(i)}, \boldsymbol{\theta}_n^{(i)}, u_{1:n-1}^{(i)})} \\ &= w_{n-1}(\boldsymbol{\tau}_{n-1}^{(i)}, \boldsymbol{\theta}_{n-1}^{(i)}, u_{1:n-2}^{(i)}) \overline{w}_n(\boldsymbol{\tau}_n^{(i)}, \boldsymbol{\theta}_n^{(i)}, u_{n-1}^{(i)}), \end{aligned}$$

with the incremental weight

$$\overline{w}_n(\boldsymbol{\tau}_n^{(i)}, \boldsymbol{\theta}_n^{(i)}, u_{n-1}^{(i)}) = \frac{\gamma_{[0,t_n]}(\boldsymbol{\tau}_n^{(i)}, \boldsymbol{\theta}_n^{(i)}, y([0,t_n]))\tilde{\pi}(u_{n-1}^{(i)}|\boldsymbol{\tau}_n^{(i)}, \boldsymbol{\theta}_n^{(i)})}{\gamma_{[0,t_{n-1}]}(\boldsymbol{\tau}_{n-1}^{(i)}, \boldsymbol{\theta}_{n-1}^{(i)}, y([0,t_{n-1}]))\gamma_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n^{(i)}, \tilde{\boldsymbol{\theta}}_n^{(i)}, y((t_{n-1}^*, t_n])|\mathbf{J}_s^{(i)}|}.$$

The particular parameter transformation s_1 should be chosen such that if $\theta_{k_{n-1}}$ and $\tilde{\theta}_0$ are samples from the corresponding conditional posterior distributions, then $\theta_n^* = s_1(\theta_{k_{n-1}}, \tilde{\theta}_0)$ should approximate a draw from the posterior for the joined segment. The transformation s_2 is less critical, but should have a distribution that can be loosely identified so as to guide how to extend the target distribution. An example is provided in Section 3.3.2.

3.3 Illustrative examples

Two examples are now presented where $y(t)$ is a Poisson process. In the first example the proposed changepoint SMC algorithm (referred to as SMC_CP) is demonstrated on the coal-mining disaster data analyzed by Raftery and Akman (1986), Green (1995) and Del Moral et al. (2006) among others. Performance is compared with the SMC samplers algorithm (SMC_S) used in Del Moral et al. (2006) and the fixed-cost approximation of the prior-sampling method of Martin et al. (2013) using the SMC_CP algorithm with $t_{n-1}^* = t_n$ (SMC_CP_P). The second example demonstrates the non-conjugate extension of the algorithm using data simulated from a shot noise cox

process taken from Whiteley et al. (2011); performance is compared with the piecewise deterministic processes particle filter (SMC_PDP) of Whiteley et al. (2011) when applied to the same model. The data and C++ code for both examples are available in the Supplementary Material.

In SMC_CP, RJMCMC is used to obtain approximate samples from each of the update intervals $(t_{n-1}, t_n]$, using standard Metropolis Hastings moves of birth (uniformly on the subinterval), death or movement (uniformly) of a changepoint, selected with equal probability when available. Starting from an initial particle with no changepoints, a burn-in of 5,000 samples is discarded.

For both examples, full RJMCMC on the whole interval $[0, t_n]$ is used as an alternative method to sample from the sequence of posteriors $\pi_{[0,t_n]}(\boldsymbol{\tau}_n|y([0,t_n]))$, with the *maximum a posteriori* sample obtained from $\pi_{[0,t_{n-1}]}$ serving as a starting value for sampling from $\pi_{[0,t_n]}$. For gauging performance, this "sequential MCMC" (denoted SMCMC) approach can be considered as a *gold standard* way to generate samples from the posterior given the data from $[0, t_n]$, although carrying a much higher computational cost than SMC.

3.3.1 Coal data

The coal mining disaster data consist of the dates of coal-mining disasters in the UK between 1851 and 1962 and are a popular data set for applying changepoint analysis. It is assumed that the disasters follow an inhomogeneous Poisson process with piecewise constant intensity function. The intensity was estimated in a sequential time frame in Del Moral et al. (2006), and here a comparison will be made with results from the SMC samplers (SMC_S) algorithm from that article using the code provided therein.

For a vector of changepoints $\boldsymbol{\tau}_n$ at time t_n, define the parameter vector $\boldsymbol{\lambda}_n = (\lambda_0, \ldots, \lambda_{k_n})$ such that the intensity of the process $\lambda(t) = \sum_{i=0}^{k_n} \lambda_i \mathbb{I}_{(\tau_i, \tau_{i+1}]}(t)$. If $r_i = \int_{\tau_i}^{\tau_{i+1}} \mathrm{d}y(t)$ is the number of disasters occurring in $(\tau_i, \tau_{i+1}]$, then the likelihood of the observed process data is

$$f(y([0,t_n])|\boldsymbol{\tau}_n, \boldsymbol{\lambda}_n) = \prod_{i=0}^{k_n} \lambda_i^{r_i} \exp\{-\lambda_i(\tau_{i+1} - \tau_i)\},$$

If independent conjugate $\Gamma(\alpha, \beta)$ priors are assigned to the intensity levels,

$$p(\boldsymbol{\lambda}_n|\boldsymbol{\tau}_n) = \prod_{i=0}^{k_n} \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_i^{\alpha-1} \exp(-\beta\lambda_i), \tag{9}$$

then for posterior inference the intensities $\boldsymbol{\lambda}_n$ can be integrated out to obtain the posterior distribution for the changepoints (5) up to proportionality through

$$\gamma_{[0,t_n]}(\boldsymbol{\tau}_n, y([0,t_n]) = \nu^{k_n} \exp(-\nu t_n) \prod_{i=0}^{k_n} \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\Gamma(\alpha + r_i)}{(\beta + \tau_{i+1} - \tau_i)^{\alpha + r_i}}.$$

Conditional on the changepoints, the intensity levels λ_i have independent posterior distributions

$$\lambda_i | \boldsymbol{\tau}_n, y([0,t_n]) \sim \Gamma(\alpha + r_i, \beta + \tau_{i+1} - \tau_i). \tag{10}$$

Del Moral et al. (2006) also treated changepoints as a Poisson process, but assumed a non-conjugate prior for the intensities with $\lambda_0 \sim \Gamma(\alpha_0, \beta_0)$ and $\lambda_i | \lambda_{i-1} \sim \Gamma(\lambda_{i-1}^2/\chi, \lambda_{i-1}/\chi)$,

$$p_{\text{SMC\_S}}(\boldsymbol{\lambda}_n | \boldsymbol{\tau}_n) = \frac{{\beta_0}^{\alpha_0}}{\Gamma(\alpha_0)} \lambda_0^{\alpha_0 - 1} e^{-\beta_0 \lambda_0} \prod_{i=1}^{k_n} \frac{(\lambda_{i-1}/\chi)^{{\lambda_{i-1}}^2/\chi}}{\Gamma({\lambda_{i-1}}^2/\chi)} \lambda_i^{{\lambda_{i-1}}^2/\chi - 1} e^{-\lambda_{i-1}\lambda_i/\chi}. \tag{11}$$

To make inference under (11), whilst still adopting the conjugate priors for the intensity levels (9) for ease of sampling, the following particle re-weighting is proposed: Given a (possibly) weighted sample of changepoints $\{\boldsymbol{\tau}_n^{(i)}, w_n^{(i)}\}_{i=1}^N$ obtained from Algorithm 1 or sequential MCMC, $\boldsymbol{\lambda}_n^{(i)}$ can first be sampled from (10) for each particle to give a weighted sample of changepoints and intensities $\{\boldsymbol{\tau}_n^{(i)}, \boldsymbol{\lambda}_n^{(i)}, w_n^{(i)}\}_{i=1}^N$. Second, this augmented sample can be simply reweighted to give an approximate sample from the non-conjugate model of Del Moral et al. (2006), with new weights $\bar{w}_n^{(i)}$ given by

$$\bar{w}_n^{(i)} = w_n^{(i)} \frac{p_{\text{SMC\_S}}(\boldsymbol{\lambda}_n^{(i)} | \boldsymbol{\tau}_n)}{p(\boldsymbol{\lambda}_n^{(i)} | \boldsymbol{\tau}_n^{(i)})}.$$

Del Moral et al. (2006) chose to perform inference annually, which implies a sequence of 112 changepoint densities where the nth density concerns the date range $[1851, 1851 + n]$. The prior intensity parameter for the occurrence of changepoints was chosen to be $\nu = 2/112$. For the non-conjugate model intensity priors, following Del Moral et al. (2006) $\alpha_0 = 4.5$ and $\beta_0 = 1.5$, while $\chi = 5$; for the conjugate model, uninformative priors are used with $\alpha = \beta = 0.1$. Again following Del Moral et al. (2006), the overall number of particles $N = 10{,}000$. For SMCMC, $1{,}000{,}000$ samples are drawn from each posterior to give reliable posterior estimates

for comparison.

Figure **??** shows the average over 5,000 runs of the online, or *filtered*, intensity function for the non-conjugate model, estimated each year using SMC_CP, SMC_S, SMC_CP_P and SMCMC. The SMC_CP and SMC_CP_P methods perfectly track the target SMCMC curve, which represents the best possible inference. In particular, SMC_CP and SMC_CP_P perform much better than SMC_S for the first fifty years of densities, and after that from 1900 the performance of the algorithms are more comparable. Figure **??** also shows the average mean square error and ESS for the SMC algorithms. The ESS for SMC_CP shows good stability across the 112 year period, whereas particularly SMC_S initially suffers from persistent weight degeneracy. In total SMC_S performs resampling due to the ESS dropping below the threshold an average of 29.0 times, compared with SMC_CP performing resampling only 7.6 times in total, and SMC_CP_P 9.7 times.

Overall, the improvement of SMC_CP over SMC_S is very clear for this example, but the performance gain of SMC_CP over the SMC_CP_P strategy of sampling from the prior for each interval (Martin et al., 2013) is less pronounced. A likely reason for this is that the coal data are a low frequency data set, with an average of less than two data points arriving per update interval, and so the differences between the prior and local posterior distributions will be more slight. In Appendix B of the Supplementary Material, a higher frequency synthetic Poisson process data set is analyzed, and under these conditions the performance of sampling from the prior deteriorates.

For computation time, a single run of SMC_CP on the coal data took 8.25s on a notebook with 1.1 GHz Intel Core processor, which is less that a tenth of a second per update interval. SMC_CP_P took even less time, 2.25s, since no MCMC or thinning is required. In contrast SMC_S took 58s, although part of the faster run times of SMC_CP and SMC_CP_P could be due to the efficient implementation of the code (available in the Supplementary Material).

3.3.2 Shot noise Cox process

Now suppose $y(t)$ is a shot noise Cox process, where changepoints $\tau_{1:k_n}$ now correspond to shots (positive jumps) in the intensity function

$$\lambda(t) = \sum_{i=0}^{k_n} \lambda_i \exp\{-\kappa(t-\tau_i)\}\mathbb{I}_{(\tau_i,\tau_{i+1}]}(t),$$

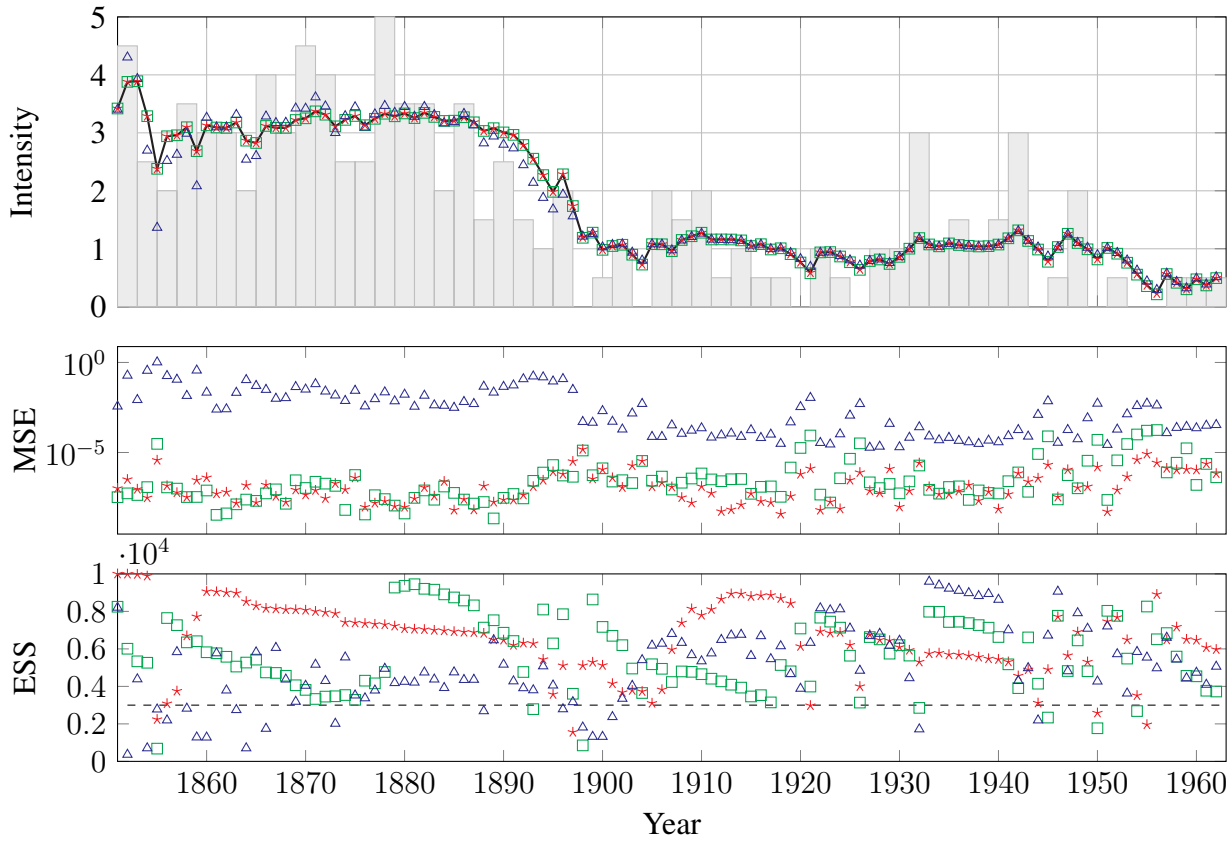

Figure 1: Averaged online estimated intensity function (top), mean square error (middle, log-scale) and effective sample size (bottom) for the coal-mining disaster data using SMCMC (**??**), SMC_CP (**??**), SMC_CP_P (**??**) and SMC_S (**??**) for $N = 10{,}000$ particles with an ESS threshold of $3{,}000$ (**??**). The bar-plot shows the empirical intensity.

where $\kappa > 0$ is a fixed decay parameter for the decrease in intensity between shots. The parameters $\boldsymbol{\lambda}_n = \lambda_{0:k_n}$ are the random intensity levels immediately after the shots, $\lambda_i \triangleq \lambda(\tau_i)$ and are constrained such that the shots are always positive; that is, $\lambda_i > \lambda_i^-$ where $\lambda_i^- \triangleq \lambda_{i-1}\exp\{-\kappa(\tau_i - \tau_{i-1})\}$ is the intensity just before τ_i.

Following Whiteley et al. (2011), the prior density for the intensity levels is

$$p(\boldsymbol{\lambda}_n|\boldsymbol{\tau}_n) = \alpha^{k_n+1}\exp(-\alpha\lambda_0)\prod_{i=1}^{k_n}\exp\{-\alpha(\lambda_i - \lambda_i^-)\}\mathbb{I}_{(\lambda_i^-,\infty)}(\lambda_i). \tag{12}$$

The likelihood of the observed process data is

$$f(y([0,t_n])|\boldsymbol{\tau}_n, \boldsymbol{\lambda}_n) = \exp\left\{-\sum_{i=0}^{k_n}(\lambda_i - \lambda_{i+1}^-)/\kappa + \int_{t=0}^{t_n}\log\lambda(t)\mathrm{d}y(t)\right\}.$$

A conjugate model approximation to the shot noise Cox process would need to forgo the constraint $\lambda_i > \lambda_i^-$. Particularly for low values of κ, which constitute the harder inference problems, reweighting samples from the unconstrained conjugate model in SMC would be an unreliable approach, since the proportion of particles obeying the required constraint, and therefore having non-zero weight according to the target model, would diminish over time. So for this application the non-conjugate algorithm of Section 3.2 is favored.

When performing a "birth" move in RJMCMC, the location of the new changepoint is drawn from the proposal distribution specified in Whiteley et al. (2011). This simply puts higher probability of proposing a changepoint at regions in the process $y(\cdot)$ where an increased rate of occurrence of events was observed, which might correspond to a shot. To follow the non-conjugate SMC algorithm, the intensity parameters λ_i are also sampled during RJMCMC. Note that, conditional on all other parameters, the λ_i corresponding to interior changepoints ($0 < i < n$) have both lower and an upper constraints: Necessarily, $\lambda_i \in (\lambda_i^-, \exp\{\kappa(\tau_{i+1} - \tau_i)\}\lambda_{i+1})$. Under (12), the full conditional distribution of λ_i is a truncated gamma distribution

$$\pi(\lambda_i|\cdot) \propto \lambda_i^{r_i}\exp\{-(\alpha\kappa + 1)z_i\lambda_i\}\mathbb{I}_{(\lambda_i^-,\exp\{\kappa(\tau_{i+1}-\tau_i)\lambda_{i+1})\}}(\lambda_i),$$

where r_i is the number of y events in $(\tau_i, \tau_{i+1}]$ and $z_i = [1 - \exp\{-\kappa(\tau_i - \tau_{i-1})\}]/\kappa$.

For the SMC algorithm, it can be supposed that two sets of particles have been obtained at

time t_n:

- $\{\boldsymbol{\tau}_{n-1}^{(i)}, \boldsymbol{\lambda}_{n-1}^{(i)}, w_{n-1}^{(i)}\}_{i=1}^{N}$, a weighted sample from $\pi_{[0,t_{n-1}]}(\boldsymbol{\tau}_{n-1}, \boldsymbol{\lambda}_{n-1}|y([0,t_{n-1}]))$;
- $\{\tilde{\boldsymbol{\tau}}_n^{(i)}, \tilde{\boldsymbol{\lambda}}_n^{(i)}\}_{i=1}^{N}$, sampled from $\pi_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n, \tilde{\boldsymbol{\lambda}}_n|y((t_{n-1}^*, t_n]))$.

To combine the two intensity parameters $\lambda_{k_{n-1}}$ and $\tilde{\lambda}_0$ to a single intensity parameter λ_n^* for the merged interval $(\tau_{k_{n-1}}, \tilde{\tau}_1]$, an attractive function would be

$$s_1(\lambda_{k_{n-1}}, \tilde{\lambda}_0) = \frac{[\lambda_{k_{n-1}}(1 - \exp\{-\kappa(t_{n-1} - \tau_{k_{n-1}})\}) + \tilde{\lambda}_0(1 - \exp\{-\kappa(\tilde{\tau}_1 - t_{n-1})\})]}{(1 - \exp\{-\kappa(\tilde{\tau}_1 - \tau_{k_{n-1}})\})},$$

which preserves the cumulative intensity over $(\tau_{k_{n-1}}, \tilde{\tau}_1]$. However, this function might propose an illegal intensity according to the constraints of the model. To ensure a legitimate proposal, it is easiest to work with the parameterization provided by the shots, $\theta_i = \lambda_i - \lambda_i^-$. Working in this parameter space, to guarantee a legitimate move the simplest choice of s is then the bivariate identity function, implying $\theta_n^* = \theta_{k_{n-1}} = \lambda_{k_{n-1}} - \lambda_{k_{n-1}}^-$ and $u_{n-1} = \tilde{\theta}_0 = \tilde{\lambda}_0$. This proposal has the potential to work well, since the proposal density for changepoints in $(t_{n-1}, t_n]$ assumes the last shot in $[0, t_{n-1}]$ was at t_{n-1}^*, and carries forward the data from $t > t_{n-1}^*$.

Finally, to extend the target distribution, $\tilde{\pi}(u_{n-1}|\tau_{1:k_n}, \lambda_{0:k_n})$ can be defined to be the full conditional from which the parameter $\tilde{\theta}_0$ was originally proposed.

Viewed from the intensity parameterization, the proposed parameters at t_n will be:

$$\{\boldsymbol{\tau}_n^{(i)} = (\tau_1^{(i)}, \ldots, \tau_{k_{n-1}}^{(i)}, \tilde{\tau}_1^{(i)}, \ldots, \tilde{\tau}_{\tilde{k}_n}^{(i)})\}_{i=1}^{N}$$
$$\{\boldsymbol{\lambda}_n^{(i)} = (\lambda_0^{(i)}, \ldots, \lambda_{k_{n-1}}^{(i)}, \tilde{\lambda}_1^{(i)} + \delta_n^{(i)}, \ldots, \tilde{\lambda}_{\tilde{k}_n}^{(i)} + \delta_n^{(i)}\}_{i=1}^{N},$$

where $\delta_n^{(i)} = \lambda_{k_{n-1}}^{(i)} \exp\{-\kappa(\tilde{\tau}_1^{(i)} - \tau_{k_{n-1}}^{(i)})\} - \tilde{\lambda}_0^{(i)} \exp\{-\kappa(\tilde{\tau}_1^{(i)} - t_{n-1})\}$.

For a true comparison against the piecewise deterministic process (SMC_PDP) algorithm in Whiteley et al. (2011), the same shot noise parameter values are used: $\nu = 1/40$, $\kappa = 1/100$ and $\alpha = 2/3$. The data obtained from the code provided in Whiteley et al. (2011) are simulated over $[0, 2000]$, with 40 update intervals each of length 50. The total number of particles $N = 500$ and the ESS resampling threshold is set to 200. Again as a comparison to both SMC algorithms, the slower but accurate SMCMC algorithm is used to provide a "gold standard" of inference.

Figure **??** shows the online filtering estimate for the intensity function using the three different algorithms SMC_CP, SMC_PDP and SMCMC for the shot noise Cox process as well as a histogram of the data over the time period. Although both SMC algorithms perform well, the SMC_CP algorithm tracks the SMCMC curve more reliably, with SMC_PDP slightly overestimating some of the shots in the intensity.

Figure **??** also shows the ESS at each time point for both SMC_CP and SMC_PDP, and SMC_CP shows much better stability in terms of the variability of the weights. The proposal distribution specified in Whiteley et al. (2011) only allows birth of changepoints within each update interval, so initially the algorithm may be overfitting changepoints causing high variability in the weights.

Whiteley et al. (2011) also plotted the number of unique particles, over time, that eventually survived to the final iteration of the SMC algorithm. The third panel of Figure **??** plots this quantity both before and after resampling, as although there may be many unique particles the importance weights may have high variance, implying a low quality particle approximation, Whiteley et al. (2011). Here SMC_CP shows much more of a diverse particle set further back in time then SMC_PDP both before and after resampling. However, it should be noted that when ESS falls below the threshold, SMC_CP performs MCMC on the whole of $\tilde{\tau}_n$ according to the $\pi_{[0,t_n]}$-invariant kernel whereas SMC_PDP only perform MCMC on τ_{k_n}. The fourth panel of Figure **??** shows the change in performance if SMC_CP is disallowed from performing any MCMC, disadvantaging the method against SMC_PDP; then the performance is much more comparable.

4 Adaptive sequential Monte Carlo

In many applications such as finance or security, there can be cause to make sequential inference about many independent target probability distributions in parallel. In finance, such problems could arise in automated trading, where beliefs about the future prices of many stocks will be continually updated; and in security, statistical models can be used for monitoring each entity in a large network for unusual behavior. Notationally, suppose there is a collection of m sequences of target distributions $\{\pi^j_{[0,t_n]}\}$, $j = 1, \ldots, m$, such that inference is to be made about each target distribution $\pi^j_{[0,t_n]}$ at the same sequence of update times $t_1 < t_2 < \ldots$.

For SMC algorithms, when updating beliefs about each of these target distributions it is de-

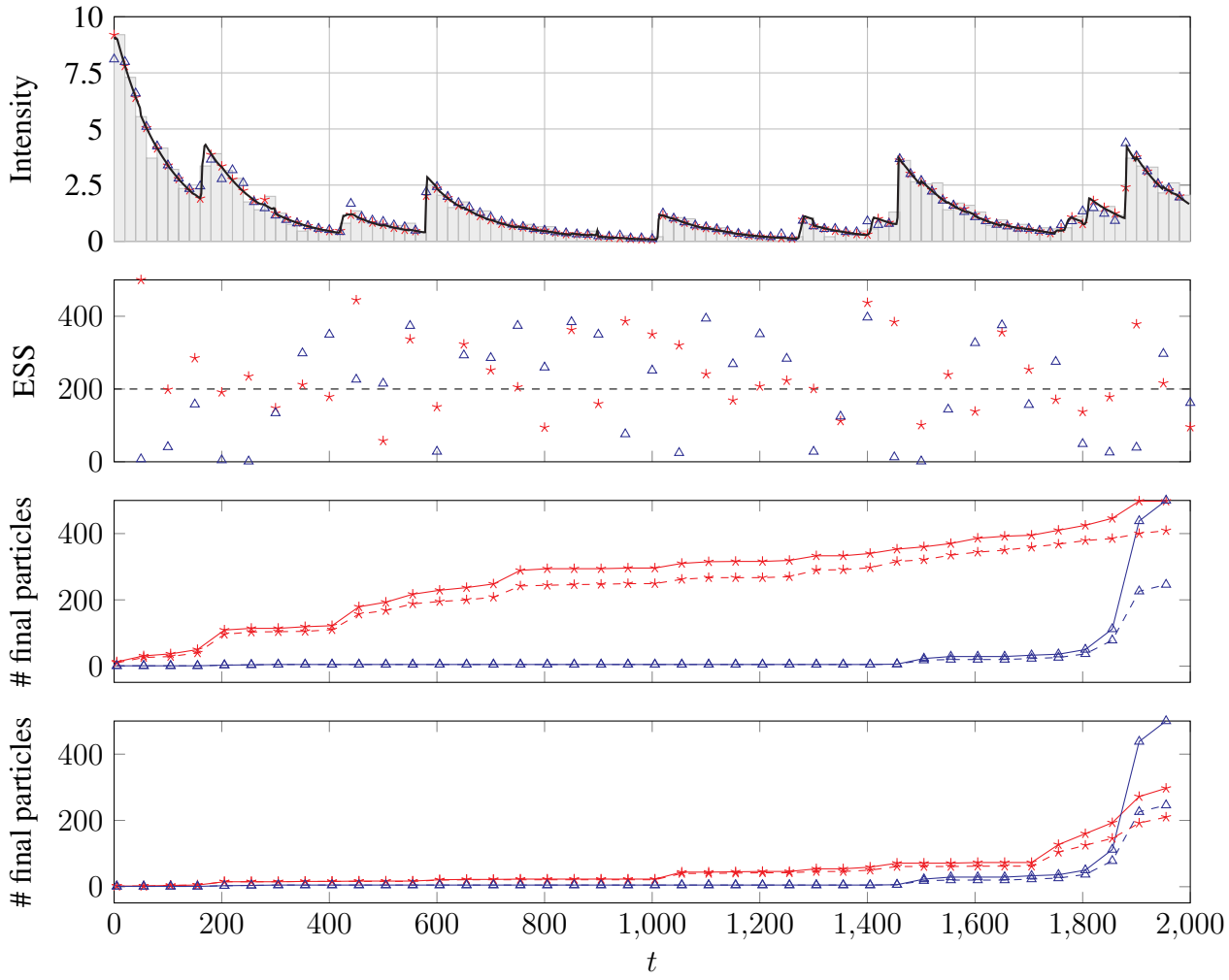

Figure 2: Results for the shot-noise model using SMCMC (**??**), SMC_CP (**??**) and SMC_PDP (**??**). Row 1: Online estimated intensity function. Row 2: Effective sample size, $\text{ESS}_{\text{min}} = 200$ (**??**). Rows 3-4: Number of unique particles present in the final sample before (dashed line) and after (solid line) resampling, with (row 3) and without (row 4) MCMC sampling for SMC_CP when $\text{ESS} < \text{ESS}_{\text{min}}$.

sirable to allocate more computational resource to those target distributions that appear to have changed the most. For existing SMC methods this idea is problematic and has not been widely explored, since the number of particles N is typically fixed from the start; these particles are either refined or resampled as the target distribution evolves, but the same number of particles is always maintained. Indeed, the effective sample size typically drops at precisely those times when the target distribution undergoes the most change, and resampling is required to ensure that the N particles are still relevant, at the cost of a loss of diversity of the particles.

As a prequel to this article, Heard and Turcotte (2015) derived a sequential approach to determining sample sizes during sampling, based on the apparent relative entropy of the different distributions. By estimating the Kullback-Leibler (K-L) divergence of each sample from its target, samples can be allocated optimally to minimize the overall discrepancy. Similarly, Fox (2003) considered estimating the K-L divergence of the sample from a target, although with respect to the predictive distribution of a particle filter for a hidden Markov model.

In the context of SMC, such an adaptive sample size strategy can be applied at each update when sampling from the proposal distributions $\pi^j_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n|y((t^{*j}_{n-1},t_n]))$. The rationale behind this approach is as follows. By conditioning on all data since the estimated most recent changepoint t^{*j}_{n-1}, the proposal density $\pi^j_{(t_{n-1},t_n]}(\tilde{\boldsymbol{\tau}}_n|y((t^{*j}_{n-1},t_n]))$ was chosen such that

$$\pi^j_{[0,t_n]}(\cdot,\cdot|y([0,t_n])) \approx \pi^j_{[0,t_{n-1}]}(\cdot,\cdot|y([0,t_{n-1}]))\pi^j_{(t_{n-1},t_n]}(\cdot,\cdot|y((t^{*j}_{n-1},t_n]))$$

Hence this proposal density will have higher entropy when more probability is assigned to the existence of multiple new changepoints of uncertain location within $(t_{n-1},t_n]$, which in turn implies a larger distance between the old and new target distributions. So by taking more samples during the current update interval, the uncertainty surrounding the new changepoints will be better captured. Whereas if another target distribution j' strongly appears to have no new changepoints during the same update window, it will be acceptable to take fewer samples to represent this portion of the distribution.

At time t_{n-1}, suppose there were N^j weighted samples approximating the jth target distribution $\pi^j_{[0,t_{n-1}]}$. Then, following the algorithm of Heard and Turcotte (2015) or some other adaptive strategy, suppose M^j samples are obtained from $\pi^j_{(t_{n-1},t_n]}(\boldsymbol{\tau}_n|y((t^{*j}_{n-1},t_n]))$, the update proposal at time t_n. Typically $M^j \neq N^j$, so at step 7 of the SMC Algorithm 1 there are two unequal

sized groups of particles to combine. To redress this imbalance, copies need to be made of some particles in the smaller sample so that the two sample sizes are equal. When $N^j < M^j$, this task of replicating particles can be used advantageously to reduce the variability of the weights of the old particles and increase the effective sample size. The next section outlines a simple procedure for determining how many copies to make of each particle, and how the particles are consequently reweighted. It can be noted that the same algorithm can equally be applied for increasing the number of new particles when $N^j > M^j$, but this trivially reduces to assigning $\boldsymbol{\tau}_n^{(i)} = \boldsymbol{\tau}_n^{(i \mod M^j)}$ for $i > M^j$.

4.1 Replicating particles

Suppose there are currently N particles for the region $[0, t_{n-1}]$ that need to be paired with $M > N$ particles from $(t_{n-1}, t_n]$ following an increased sampling allocation. When considering duplicating particles from the weighted particle set, it is important to note that there may already be duplicated particles, perhaps from previous iterations. For continuous time changepoints, duplicates also arise when particles with no changepoints are sampled. For simplicity of notation, assume now that the weighted particle set has been labeled such that the first $N' < N$ particles are unique. For $1 \leq i \leq N$, let $m_0^{(i)}$ be the number of replicates of $\boldsymbol{\tau}_{n-1}^{(i)}$ in the N particles, and define $\bar{w}^{(i)} = w_{n-1}^{(i)} m_0^{(i)}$. Then note that $\{\boldsymbol{\tau}_{n-1}^{(i)}, \bar{w}^{(i)}\}_{i=1}^{N'}$ is an equivalent representation of the full weighted particle set, since

$$\sum_{i=1}^{N'} \bar{w}^{(i)} \delta_{\boldsymbol{\tau}_{n-1}^{(i)}}(\boldsymbol{\tau}_{n-1}) \equiv \sum_{i=1}^{N} w_{n-1}^{(i)} \delta_{\boldsymbol{\tau}_{n-1}^{(i)}}(\boldsymbol{\tau}_{n-1}).$$

It is necessary to work with this reduced representation, as otherwise the algorithm would admit the possibility of making different numbers of copies of the same particle. Assume that each unique particle i will be replicated $m^{(i)}$ times, so that $\sum_{i=1}^{N'} m^{(i)} = M$. Then in order to minimize the sum of the squared weights and ensure that any Monte Carlo estimates are the same after the particle set has been increased, the revised weight for particle i is $\bar{w}^{(i)}/m^{(i)}$. The important implication here is that replicating highly weighted particles will reduce those weights, which will make the weights more uniform and therefore boost the effective sample size in step 9 of Algorithm 1.

Algorithm 2 Increasing particle set from N to M

1: Set $m^{(i)} = m_0^{(i)}$ and $\bar{w}^{(i)} = m_0^{(i)} w^{(i)}$ for $i = 1, \ldots, N'$. Let $m = \sum_{i=1}^{N'} m^{(i)} = N$
2: Calculate $\delta_i = (\bar{w}^{(i)})^2 / \{(m^{(i)} + 1) m^{(i)}\}$ for $i = 1, \ldots, N'$
3: Let $i^* = \text{argmax}_i \{\delta_i : i = 1, \ldots, N'\}$
4: **while** $m < M$ **do**
5: Let $i^{**} = \text{argmax}_i \{\delta_i : i = 1, \ldots, N', i \neq i^*\}$
6: Let $x_{N'} = \min \left(M - m, \lceil \sqrt{(\bar{w}^{(i^*)})^2 / \delta_{i^{**}} + 0.25} - 0.5 - m^{(i^*)} \rceil \right)$
7: Let $m^{i^*} = m^{i^*} + x_{N'}$ and $m = m + x_{N'}$
8: Let $\delta_{i^*} = (\bar{w}^{(i^*)})^2 / \{(m^{(i^*)} + 1)(m^{(i^*)})\}$
9: Let $i^* = i^{**}$
10: **end while**
11: Let $i' = 1$
12: **for** $i = 1 : N'$ **do**
13: **for** $j = 1 : m^{(i)}$ **do**
14: $\boldsymbol{\tau}_{n-1}^{*(i')} = \boldsymbol{\tau}_{n-1}^{(i)}$
15: $w_{n-1}^{*(i')} = \bar{w}^{(i)} / m^{(i)}$
16: $i' = i' + 1$
17: **end for**
18: **end for**
19: $(\boldsymbol{\tau}_{n-1}^{(i)}, w_{n-1}^{(i)}) \leftarrow (\boldsymbol{\tau}_{n-1}^{*(i)}, w_{n-1}^{*(i)})$ for $i = 1, \ldots, M$

Choosing optimal values $\{m^{(i)}\}_{i=1}^{N'}$, so that the resulting sum of squared weights is minimized is a complex optimization problem, and solving this directly would add too much computational burden to the overall SMC algorithm. So instead, Algorithm 2 presents a sequential optimization method.

The quantity δ_i, calculated in step 2, represents the decrease in the sum of the squared weights if the ith particle is replicated once, and this is used to identify the next particle to replicate, i^*. The number of replicates of particle i^* that are then made, $x_{N'}$, calculated in step 6, is the largest integer solving the inequality

$$\frac{(\bar{w}^{(i^*)})^2}{(m^{(i^*)} + x_{N'} + 1)(m^{(i^*)} + x_{N'})} < \delta_{i^{**}},$$

since this is the smallest number of replicates that are required for i^* not to remain the optimal particle to replicate.

4.2 Example: The VAST data

The IEEE VAST 2008 Challenge data are synthetic data comprising information of mobile call records for a small community of 400 mobile phones, over a 10 day period. The challenge was aimed at social network analysis, with the aim of uncovering anomalous behavior within the social network. The data can be obtained from `www.cs.umd.edu/hcil/VASTchallenge08`.

A successful approach taken in Heard et al. (2010) was to monitor the incoming call patterns of each phone to detect changes from their normal patterns, and thereby obtain a much smaller subset of potentially anomalous IDs that can further be investigated. After correcting for diurnal effects on normal behavior as in Heard et al. (2010), this approach reduces to the online detection of changepoints of 400 processes which can be assumed to follow a Poisson process with a conjugate prior for the intensity as detailed in Section 3.3.1. Furthermore, it was later shown in Heard and Turcotte (2015) that for a fixed computational effort, more accurate inference could be obtained from the Poisson process model of these data by using the adaptive sampling strategy presented in that paper.

The SMC Algorithm 1 can be deployed to simulate a real time changepoint analysis of the incoming call data for each phone number in the network. Each phone number is reanalyzed each hour over the ten days of data, which corresponds to 240 update intervals. Furthermore, to illustrate the adaptive version of the SMC algorithm, a variable number of particles are assigned to each process in each update window according to the complexities of their distributions using the Algorithm given in Heard and Turcotte (2015) for minimizing the total estimation error of the distributions. On each interval, each process is given a minimum of 500 particles, but the total number of samples to be adaptively allocated across the processes $m^* = 4{,}000{,}000$; so equal sample sizes would correspond to $10{,}000$ particles for each process. The data and C++ code for both examples are available in the Supplementary Material.

For each update window, Figure **??** shows a box plot of the sample sizes N^j allocated to the 400 processes. The dotted line shows the sample size that would be allocated to each process under a fixed sample size strategy, where $N^j = 10{,}000$. It is interesting to note that on some update intervals, most of the samples sizes are bunched together (making the box appear as a single horizontal line); this is particularly the case at the start of the observation period, when little is known about any of the distributions.

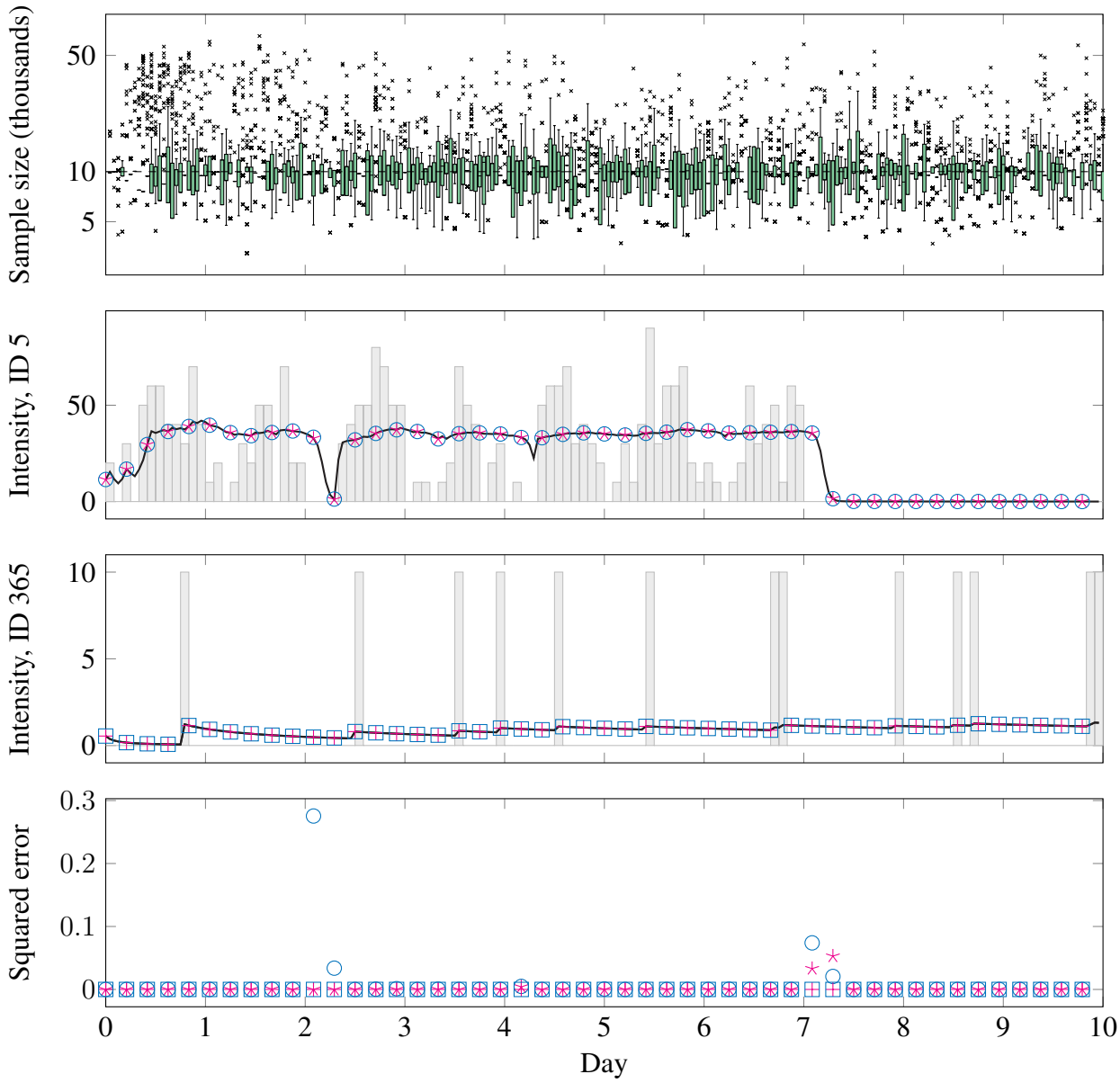

Figure 3: Top: Box plot of the sample sizes $\{N^j\}$ allocated by adaptive SMC to the individuals in the VAST data for each of the 240 update intervals. Equal allocation corresponds to $N^j = 10{,}000$ (**??**). Rows 2-3: Binned empirical intensity of incoming call data and SMCMC (**??**) estimated online intensity (after correcting for diurnal changes in intensity); Row 2: Adaptive (**??**) and fixed (**??**) sample size estimates for the ID allocated the most samples in adaptive SMC; Row 3: Adaptive (**??**) and fixed (**??**) sample size estimates for the ID allocated fewest samples. Bottom: squared errors of estimates from the mean intensity from SMCMC.

In Ye et al. (2008) it was shown that the main anomalous activity involved at least four actors with respective phone IDs 1, 2, 3 and 5, who change their handsets on the eighth day of the data period. Interestingly, IDs 5 and 1 get the highest and second highest total allocation of sampling effort in the adaptive SMC algorithm, and the other anomalous IDs get the eleventh and fourteenth highest allocations. Figure **??** shows the estimated intensity functions for the IDs obtaining the highest (ID 5) and lowest (ID 365) overall number of samples. It can be seen that ID 5 has much more event data than ID 365, and a correspondingly more interesting intensity function.

For each of these IDs, the second and third plots in Figure **??** show the estimated intensities obtained from SMCMC and SMC_CP using either adaptive or fixed sample sizes, and on this scale the differences in the estimates are difficult to detect. However, the bottom panel of Figure **??** shows just the squared errors of these intensity functions from the SMCMC estimate, and there the adaptive approach is shown to have promising benefits.

5 Discussion

A new SMC algorithm for changepoint analysis has been presented, and shown to outperform existing SMC methods. The computational effort of the algorithm does not increase over time. Effective sample size (ESS) thresholding has been used to control diversity of the particles in all examples; other standard techniques for improving SMC performance can also be applied to Algorithm 1, such as the Resample-Move algorithm of Gilks and Berzuini (2001), where MCMC transition kernels are applied to the particle set after ESS resampling to introduce diversity. The algorithm has also been shown to be adaptive in the number of particles used over time, which further improves upon the computational savings that SMC methods offer.

The simplicity of the method derives from summarizing existing particles at each time step with a single point estimate, before sampling changepoints for the new interval conditioning only upon that point estimate. Cases where this estimate has low variance are when this approach will work best, whereas when beliefs about the most recent changepoint are more diffuse this summary will be an over-simplification and should lead to faster weight degeneration. Possible extensions to the method could therefore be to calculate several point estimates, such as a set of quantiles of the distribution for the last changepoint, and then take a mixed sample from each of

the implied proposals.

6 Supplementary Material

All supplemental files are contained in a single archive.

Appendices : Appendix A presents an analysis using the changepoint prior (2) which permits exact simulation within the SMC algorithm. Appendix B considers simulated data from a more challenging Poisson process with 500 changepoints. Appendix C demonstrates application of the algorithm to two different changepoint data problems: piecewise constant Gaussian regression with unknown mean and variance, and Poisson regression.

C++ code : The code and the data for the examples in Section 3.3.1, 3.3.2, 4.2 and the other applications presented in the appendices. See the README file for details on how to run the code for the examples. The code and the data are also available to download from `https://github.com/mjmt05/rjmcmc.git`.

References

Chopin, N. (2007). Dynamic detection of change points in long time series. *Annals of the Institute of Statistical Mathematics*, 59:349–366.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 68:411–436.

Fearnhead, P. and Clifford, P. (2003). On-line inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 65:887–899.

Fearnhead, P. and Liu, Z. (2007). On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 69:589–605.

Fox, D. (2003). Adapting the sample size in particle filters through KLD-sampling.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63:127–146.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732.

Heard, N. and Turcotte, M. (2015). Convergence of Monte Carlo distribution estimates from rival samplers. *Statistics and Computing, to appear*.

Heard, N. A., Weston, D. J., Platanioti, K., and Hand, D. J. (2010). Bayesian anomaly detection methods for social networks. *Annals of Applied Statistics*, 4:645–662.

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25.

Liu, J. S. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6:113–119.

Martin, J., Jasra, A., and McCoy, E. (2013). Inference for a class of partially observed point process models. *Annals of the Institute of Statistical Mathematics*, 65(3):413–437.

Raftery, A. E. and Akman, V. E. (1986). Bayesian analysis of a Poisson process with a change-point. *Biometrika*, 73:85–89.

Whiteley, N., Johansen, A. M., and Godsill, S. (2011). Monte Carlo filtering of piecewise deterministic processes. *Journal of Computational and Graphical Statistics*, 20(1):119–139.

Ye, Q., Zhu, T., Hu, D., Wu, B., Du, N., and Wang, B. (2008). Cell Phone Mini Challenge Award: Social Network Accuracy - Exploring temporal communication in mobile call graphs. In *Proceedings of IEEE VAST Symposium*, pages 207–208. IEEE.