UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA

Lívia Maria Dutra

Exact Bayesian inference for Markov switching Cox processes

Belo Horizonte
2019

Lívia Maria Dutra

Exact Bayesian inference for Markov switching Cox processes

Tese apresentada ao Programa de Pós-graduação em Estatística da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Doutora em Estatística.

Orientador: Flávio Bambirra Gonçalves
Co-orientador: Roger William Câmara Silva

Belo Horizonte
2019

Ficha catalográfica elaborada pelo bibliotecário Célio Resende Diniz - CRB 6ª Região nº 2.403.

Dutra, Lívia Maria.

D978e Exact Bayesian inference for Markov switching Cox processes / Lívia Maria Dutra — Belo Horizonte, 2019.
80 f.: il.; 29 cm.

Tese (doutorado) - Universidade Federal de Minas Gerais – Departamento de Estatística.

Orientador: Flávio Bambirra Gonçalves.
Coorientador: Roger William Câmara Silva.

1. Estatística. – Teses. 2. Teoria bayesiana de decisão estatística – Teses. 3. Markov, Processos de – Teses. I.Orientador. II. Coorientador. III. Título.

CDU 519.2(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS

PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA

UFMG

ATA DA DEFESA DE TESE DA ALUNA
LÍVIA MARIA DUTRA

Realizou-se, no dia 04 de dezembro de 2019, às 09:00 horas, 2025 ICEx, da Universidade Federal de Minas Gerais, a 58ª defesa de tese, intitulada *Exact Bayesian inference for Markov switching Cox processes*, apresentada por LÍVIA MARIA DUTRA, número de registro 2015700476, graduada no curso de ESTATISTICA, como requisito parcial para a obtenção do grau de Doutor em ESTATÍSTICA, à seguinte Comissão Examinadora: Prof(a). Flávio Bambirra Gonçalves - Orientador (DEST/UFMG), Prof(a). Roger William Camara Silva (DEST/UFMG), Prof(a). Dani Gamerman (DEST/UFMG), Prof(a). Marcos Oliveira Prates (DEST/UFMG), Prof(a). Rafael Izbicki (UFSCAR), Prof(a). Daiane Aparecida Zuanetti (UFSCar).

A Comissão considerou a tese:

(X) Aprovada

( ) Reprovada

Finalizados os trabalhos, lavrei a presente ata que, lida e aprovada, vai assinada por mim e pelos membros da Comissão.
Belo Horizonte, 04 de dezembro de 2019.

Prof(a). Flávio Bambirra Gonçalves (Doutor)

Prof(a). Roger William Camara Silva (Doutor)

Prof(a). Dani Gamerman (Doutor)

Prof(a). Marcos Oliveira Prates (Doutor)

Prof(a). Rafael Izbicki (Doutor)

Prof(a). Daiane Aparecida Zuanetti (Doutor)

Agradecimentos

Muitos foram importantes durante essa trajetória e só consigo sentir gratidão!

Ao Flávio, meus mais sinceros agradecimentos. Um professor e orientador nato! Há 7 anos acreditando na minha capacidade, orientando e mantendo a calma sempre quando eu achava que tudo estava perdido e com uma dedicação sem igual. Agradeço imensamente ao Roger, por também sempre acreditar na minha capacidade e pela dedicação. Meus orientadores de fato tem minha admiração e inspiração profissional! Ao Gregório, por abrir meus olhos para a beleza da aleatoriedade. Agradeço a todos os professores por contribuírem para minha formação acadêmica e profissional. Também, aos funcionários do Departamento de Estatística pelos serviços prestados. À CAPES, pelo auxílio financeiro.

Agradeço ao Leandro por ser extremamente paciente e compreensivo, pelo companheirismo e amor! Agradeço a toda minha família pelo apoio e carinho. Aos meus pais por não medir esforços, pelo suporte, incentivo e amor. Aos meus irmãos por acreditarem em mim. À minha amorzinha Marina, por muitas vezes me fazer esquecer dos problemas da tese e me acalmar, mesmo sem saber o que estava acontecendo.

E esse ciclo não poderia ter finalizado com sucesso sem os amigos. Em especial, agradeço a Ju, juntas na alegria e no desespero por tantos anos! À amiga Larissa, pelo exemplo de disciplina e uma boa conversa sempre! À Gabi, sempre acreditando em mim mais do que eu mesma! À Zaidinha pelo exemplo de dedicação, competência e amizade! Agradeço à todos que estiveram na PG-Est juntos comigo por algum momento. Foram tantos conhecimentos, estudos, histórias, risadas, lágrimas, desesperos e alegrias compartilhadas! Aos que saíram antes, foram inspiração e esperança de que o fim chega. Aproveito para deixar o meu apoio aos próximos colegas doutores: "Se está difícil, lembre-se que ao final o esforço será recompensado. Força e coragem!". E, claro, agradeço imensamente aos amigos fora dessa bolha! Essenciais para manter o equilíbrio.

Agradeço ao DECOM-CEFET e colegas pelo apoio e pela receptividade. Agradeço e dedico essa conquista aos meus alunos! Estar dentro de sala de aula é a minha paixão e o que me motiva!

Por último, obrigada UFMG!! O final do doutorado marca também o final de uma vida inteira de discente. Há 12 anos começava a minha história com essa instituição a qual lembrarei com muito carinho e orgulho!

Resumo

A modelagem estatística de dados pontuais é um problema comum e importante em diversas áreas do conhecimento. O processo pontual mais amplamente utilizado e o mais comum é o processo de Poisson e, em particular, em uma de suas generalizações, sua função de intensidade é considerada também como um processo estocástico. Este modelo é conhecido como processo de Cox e diferentes opções para modelar a dinâmica da função de intensidade dão origem a uma ampla gama de modelos. Apresentamos uma nova classe de processos Cox unidimensionais, a qual é um processo de Poisson não-homogêneo em que a função de intensidade se alterna entre diferentes formas funcionais paramétricas de acordo com a trajetória de uma cadeia de Markov em tempo contínuo. Nos referimos a essa nova classe como processos de Cox com mudanças markovianas. Alguns resultados e algoritmos já presentes na literatura são utilizados como base para desenvolver uma metodologia Bayesiana para se realizar inferência exata, através de algoritmos MCMC. A confiabilidade do algoritmo depende de uma variedade de especificações que são cuidadosamente abordadas. Estudos simulados e análise de dados reais são apresentados com o objetivo de investigar a eficiência e aplicabilidade da metodologia proposta.

Palavras-chave: Inferência Bayesiana. Distribuição a posteriori exata. Processo de Cox. Cadeia de Markov em tempo contínuo.

Abstract

Statistical modelling of point patterns is an important and common problem in several areas. Poisson process is the most common process used for this purpose and, in particular, its generalisation considers a stochastic intensity function. This is called a Cox process and different choices to model the dynamics of the intensity give raise to a wide range of flexible models. We present a new class of unidimensional Cox processes in which the intensity function is driven by parametric functional forms that switch among themselves according to a continuous-time Markov chain. We refer to these as Markov switching Cox processes (MSCP). Previous developments in the literature are used to develop a Bayesian methodology to perform exact inference based on MCMC algorithms. The reliability of the algorithm depends on a variety of specifications which are carefully addressed. Simulated and real studies are presented in order to investigate the efficiency and applicability of the proposed methodology.

Keywords: Bayesian inference. Exact posterior distributions. Cox process. Continuous-time Markov chain.

Contents

Chapter 1

Introduction

Statistical modelling of point patterns is an important and common problem in several knowledge areas. Typically, the aim is to model the occurrences of a given event of interest in a given region. The most widely used point process model is the Poisson process (PP), which the number of events in a given region has Poisson distribution. This is the simplest model, since the rate of event occurrences is constant over the observed region. That is, the expected number of events in a sub-region is equal for any sub-region with the same size. Furthermore, the observation points in non-overlapping regions are independent. A possible generalisation of PP is the non-homogeneous Poisson process (NHPP), which allows the rate to vary across the observed region. That is, the rate is a function of the observed region and the expected number of events can be different for distinct sub-regions with the same size. Note that in this process one sub-region may be more likely to have more events than another. For NHPP's, the rate is called intensity function (IF).

Several extensions for point processes have been developed to make flexible the rate of events over the observed region, so that the model application becomes more realistic with the problem. The Cox processes (Cox, 1955), also known doubly stochastic Poisson processes, are an appealing generalisation of the Poisson process. A Cox process (CP) is a non-homogeneous Poisson process in which the intensity function evolves stochastically. While PP has a constant rate for event occurrences and the intensity function of NHPP is a deterministic function, the CP is more flexible to model real data due to the stochastic nature of its IF. These processes have been applied in different contexts, e.g., bursts of rainfall (Smith & Karr, 1983), neuroscience (Amarasingham et al., 2006; Cunningham et al., 2008), finances (Lando, 1998; Dassios & Jang, 2003) and others.

A particular interesting example of a Cox process is the Markov Modulated Poisson Process (MMPP) (Fischer & Meier-Hellstern, 1992), in which the intensity function is driven by a finite state space continuous-time Markov chain (CTMC). This means that a MMPP is a NHPP whose intensity function is a path of a CTMC, that is, the IF is piecewise-constant and switches according to a CTMC transitions. Note that the possible values for this intensity function are the constants in the CTMC state space. MMPP's have been applied in a variety of problems, e.g., web traffic (Scott & Smyth, 2003), occurrence of a rare DNA motif (Fearnhead & Sherlock, 2006) and software reliability (Landon et al., 2013).

An appealing and realistic generalisation of a MMPP is to allow the change of the intensity function form over time, instead of switching only between different constants. Thereby, we propose a novel Cox process model, based on MMPP. Rather than assuming the state of a CTMC, the intensity function switches among different pre-defined parametric functional forms according to a CTMC transitions. We assume that the state space of the CTMC only indicates which functional form the IF should adopt. We call this as Markov switching Cox process (MSCP). Hence, while MMPP alternates stochastically between homogeneous PP's, a MSCP alternates stochastically between parametric NHPP's. Note that a MMPP is an specific case of a MSCP and the proposed model is more flexible than MMPP due to the behaviour of its rate, which can change its form over time. In this work we only consider point processes on $\mathbb{R}$, therefore, the regions will usually be interpreted as time.

We develop a Bayesian methodology to perform exact inference in these processes, via MCMC algorithms. The term exact refers to the fact that no time discretisation approximation is assumed, only Monte Carlo errors that can be controlled. The challenging step is to characterise the posterior distribution of the intensity function. Rao & Teh (2013) developed a general MCMC sampling algorithm that can be applied to MMPP's, which is an exact and computationally efficient algorithm. Unfortunately, their algorithm can not be applied to the class of MSCP proposed in this work. However, we shall develop over the Rao & Teh (2013) algorithm form by adopting the same model augmentation scheme considered by those authors, adding an auxiliary variable. Also, we propose an improvement to sample this auxiliary variable from its posterior distribution.

This thesis is organised as follows. The Chapter 2 presents the proposed model and our

methodology to perform inference. Also, we develop an extension of the first model and make the necessary adjustments to perform inference, which are presented in Chapter 3. In Chapter 4 the methodology is applied to simulated data sets, with different scenarios, and to two real data sets to investigate its efficiency. Also, we compare our methodology with a model that consider non-parametric forms for the intensity function. Final remarks are presented in Chapter 5.

1.1 Markov modulated Poisson processes

The model developed in this thesis was devised from Markov modulated Poisson processes. Therefore, here we present a brief review of this model, with its definition, an example and Bayesian methodologies previously proposed to perform inference.

The definition of a MMPP is following.

Definition 1.1. *A Markov modulated Poisson process* $Y := \{Y(s)\}_{s\in\mathbb{R}^+}$ *is a non-homogeneous Poisson process such that the intensity function* $\lambda := \{\lambda(s)\}_{s\in\mathbb{R}^+}$ *is a continuous-time Markov chain with state space* $E = \{\epsilon_1, \cdots, \epsilon_{|E|}\}$, $\epsilon_i \in \mathbb{R}^+$ $\forall i$, *initial distribution* π_0 *and* Q*-matrix* $Q = \{q_{ij} : i, j \in E\}$.

Figure 1.1 shows an example of a MMPP in $[0, 100]$ in which the underlying CTMC has state space $\{1, 3, 7\}$, uniform initial distribution and Q-matrix

$$Q = \begin{bmatrix} -1/40 & 1/80 & 1/80 \\ 1/40 & -1/20 & 1/40 \\ 1/40 & 1/40 & -1/20 \end{bmatrix}.$$

Also, Figure 1.1 shows the cumulative number of events over time and it is possible to see that the rate of event occurrences is not constant over the entire interval. Furthermore, for each observed change in the slope of the curve in the botton of Figure 1.1 may have a change of the IF. This assumption can be confirmed with the hidden CTMC path that generated these observations.

The transition dynamics of the CTMC is such that the chain remains on state i for an exponentially distributed time with mean $1/|q_{ii}|$. The transition probabilities among the states are given by the normalised off-diagonal values, i.e., $q_{ij}/|q_{ii}|$ is the transition probability from state i to j.

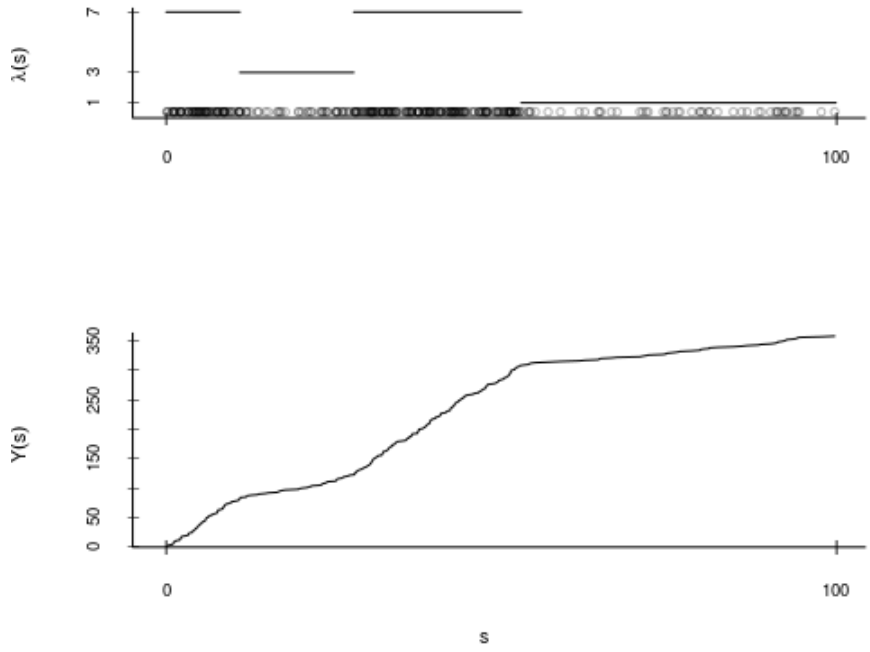


Figure 1.1: Intensity function and realisation of a MMPP. Top: the solid line is the IF trajectory and the small circles are events of the PP. Bottom: Accumulated number of events.

Methodologies to perform inference for MMPP have been developed and applications in several areas show the importance of this model. Based on the observation of the process in a compact time interval, the aim is to estimate the intensity function and the parameters indexing the CTMC distribution. Under the Bayesian paradigm, inference should be based on the posterior distribution of all the unknown quantities of the model. Given the intractability of the posterior, MCMC is the most reasonable way to explore this distribution.

Scott (1999) and Scott & Smyth (2003) propose a Gibbs sampling algorithm based on a discrete time approximation. More specifically, the CTMC is only allowed to jump at event times. The proposed methodology is then applied in examples with data from banking transactions and web traffic.

Fearnhead & Sherlock (2006), on the other hand, propose an exact Gibbs sampler in which the CTMC is sampled in two steps. First, it is sampled only at event times and then in the intervals between them, conditional on the states at those times. The latter is based on the algorithm to sample from CTMC bridges proposed in Hobolth & Stone (2009). The proposed Gibbs sampling has as its main frailty the fact that its cost scales

with the data size, which compromises its application to large datasets. The proposed methodology is used to model the occurrence of a rare DNA motifs along the genome.

Finally, Rao & Teh (2013) develop an exact MCMC algorithm using a data augmentation scheme based on uniformisation algorithm of Hobolth & Stone (2009). The main advantage of this methodology if compared to that from Fearnhead & Sherlock (2006) is that its cost does not scale with the data size, but rather with the smallest mean waiting time of the CTMC. The CTMC is sampled in two steps. First, jump times of an extended CTMC are sampled. Then, the transitions at those times are sampled using a forward-filtering backward-sampling (FFBS) scheme.

The algorithm proposed in this thesis make use of the data augmentation idea from Rao & Teh (2013), but has to deal with a complex sampling problem due to the fact that a FFBS scheme is not viable.

Chapter 2

Markov switching Cox processes

This chapter presents the proposed model and some of its properties as well as the methodology to perform exact Bayesian inference. We named the model as a Markov switching Cox process (MSCP).

The full model specification and some examples are presented in Section 2.1. Section 2.2 presents the inference problem and the general solution for that. A detailed description of all the MCMC steps is presented in Section 2.3. Finally, Section 2.4 brings some strategies to boost MCMC efficiency.

2.1 The proposed model

Let $\{X(s)\}_{s\in\mathbb{R}^+}$ be a continuous-time Markov chain with Q-matrix Q_θ, initial distribution π_0 and state space $E = \{1, 2, \cdots, |E|\}$, $|E| \in \mathbb{N}$, where θ is the vector of parameters indexing Q_θ. For cleanness of notation, we shall refer to the process $\{X(s)\}_{s\in\mathbb{R}^+}$ as X.

Now define $T = (T_1, T_2, \cdots)$ as the jump times of X and $Z = (Z_1, Z_2, \cdots)$ as the corresponding sequence of visited states, i.e., $Z_i = X(T_i)$, $i \in \mathbb{N}$, and $Z_0 = X(0)$. The triplet (Z_0, Z, T) completely characterises the CTMC trajectory.

Let h be the surjective function $h : E \to K$, for $K = \{1, 2, \cdots, |K|\}$, $|K| \in \mathbb{N}$, that assigns a functional form g_k to each of the states of X. Finally, define $\{Y(s)\}_{s\in\mathbb{R}^+}$ to be a non-homogeneous Poisson process with intensity function $\{\lambda(s, X, \psi)\}_{s\in\mathbb{R}^+}$, where $\lambda(s, X, \psi) = g_{h_s}(s, \delta_T(s), \psi)$, $h_s := h(X(s))$, $\delta_T(s) = \max_{i\geq 0}\{T_i : T_i \leq s\}$, $T_0 = 0$ and ψ is a parameter vector indexing the functional forms g_k. We shall use $\lambda(s) := \lambda(s, X, \psi)$, $Y := \{Y(s)\}_{s\in\mathbb{R}^+}$ and $\lambda := \{\lambda(s)\}_{s\in\mathbb{R}^+}$ for cleanness of notation.

The g_k's are the functional forms that the intensity function can assume - $g_1(s, \delta_T(s), \psi_1)$, $\cdots$, $g_{|K|}(s, \delta_T(s), \psi_{|K|})$, and we shall use the notation $g_k := g_k(s, \delta_T(s), \psi_k)$. That is, at time s we look at $h(X(s))$ to decide what functional form the IF assumes. This means that the CTMC path determines the functional form of the IF and when it changes. For that reason, the state space E is just an enumeration to construct the domain of the function h. The need of the auxiliary function h will be made clear further ahead in the text. We refer to process Y as a Markov switching Cox process. We have the following hierarchical representation of the MSCP:

$$
\begin{aligned}
Y &\sim \text{NHPP}(\lambda), && (2.1)\\
\lambda(s) &= g_{h_s}(s, \delta_T(s), \psi), && (2.2)\\
h: &\; E \to K, && (2.3)\\
X &\sim \text{CTMC}(\pi_0, Q_\theta, E). && (2.4)
\end{aligned}
$$

Note that function $\delta_T(s)$ gives the last CTMC jump time preceding time s and defines the required translation of the respective functional form so that each g_k restarts always at the same value. For example, for $s_1 < s_2$ such that $h_{s_1} = h_{s_2} = k$, $\delta_T(s_1) \neq \delta_T(s_2)$ and $s_1 - \delta_T(s_1) = s_2 - \delta_T(s_2)$, we have that $g_k(s_1, \delta_T(s_1), \psi_k) = g_k(s_2, \delta_T(s_2), \psi_k)$.

The sets E and K do not necessarily have the same size. In the simplest case, each jump of X also indicates a change of the functional form and, therefore, $E = K$ and $h(e) = e, \forall e \in E$. Figure 2.1 shows an example of a MSCP with $E = K = \{1, 2\}$, where $g_1 = 1 + 0.2(s - \delta_T(s))$ and $g_2 = 1$.

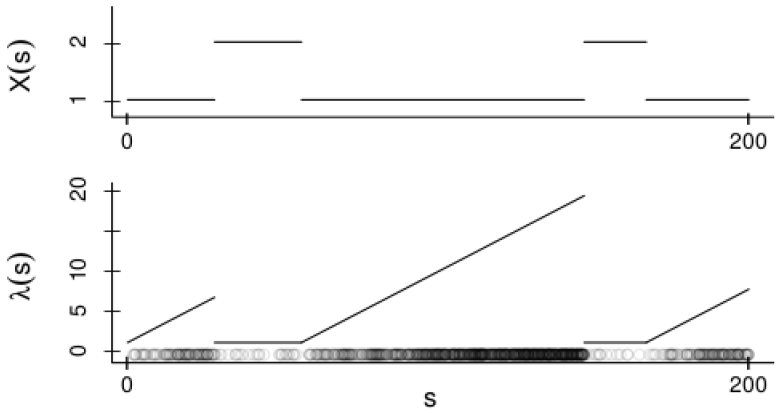


Figure 2.1: Example of a MSCP with a two-state CTMC and two functional forms. Top: CTMC trajectory. Bottom: solid line is the intensity function and circles are the events.

Nevertheless, it also makes sense to have a change from a given functional form to itself, meaning that the same g_k restarts itself. Since the sequence of functions g_k's is chosen according to a CTMC trajectory, this would be equivalent to have jumps to the same state in the CTMC, which cannot be accommodate by a CTMC trajectory. To solve this issue, the state space of X will be extended. We assign the same k to two different states of the CTMC if we want to allow the respective g_k to restart itself. That is, for a g_k allowed to restart itself, the state space E is augmented in such a way that $h^{-1}(k)$ is a subset with two states from E. In particular, if all functional forms can be restarted, we have $|E| = 2|K|$. Finally, note that it does not make sense to restart g_k if this is a constant function.

Figures 2.2 and 2.3 show examples where E is extended. In the first one, $E = \{1, 2, 3\}$, $K = \{1, 2\}$, $g_1 = 1 + 0.2(s - \delta_T(s))$ and $g_2 = 1$, with $h(1) = h(3) = 1$ and $h(2) = 2$. In the second one, $E = \{1, 2, 3, 4\}$, $K = \{1, 2\}$, $g_1 = 0.2(s - \delta_T(s))$ and $g_2 = 15 - 0.2(s - \delta_T(s))$, with $h(1) = h(3) = 1$ and $h(2) = h(4) = 2$.

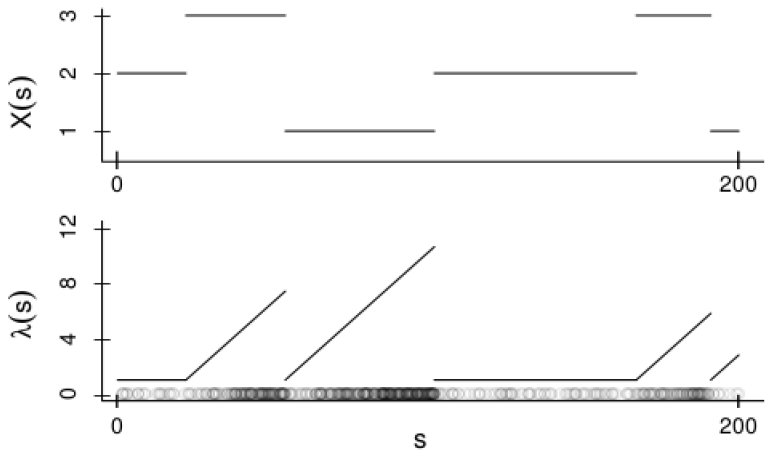


Figure 2.2: Example of a MSCP with a three-state CTMC and two functional forms. Top: CTMC trajectory. Bottom: solid line is the intensity function and circles are the events.

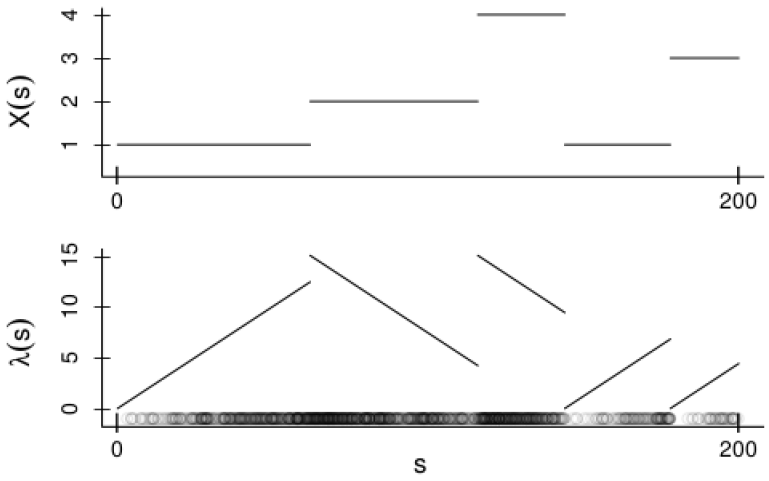


Figure 2.3: Example of a MSCP with a four-state CTMC and two functional forms. Top: CTMC trajectory. Bottom: solid line is the intensity function and circles are the events.

2.2 Bayesian inference

Based on a realisation of Y in $[0, S]$, we want to estimate the intensity function and the unknown parameters indexing the model. The values of $|K|$ and $|E|$ and functional forms g_k are fixed. The unknown parameters of the model are in vectors θ and $\psi = \{\psi_1, \cdots, \psi_{|K|}\}$, such that ψ_k are the parameters indexing the functional form g_k .

Inference is performed under the Bayesian paradigm which means that it should be based on the posterior distribution $\pi(Z_0, Z, T, \theta, \psi|y)$, where y is a realisation of Y.

Bayes Theorem (see Shao, 2003, Theorem 4.1) implies that

$$\pi(Z_0, Z, T, \theta, \psi|y) \propto \pi(y|Z_0, Z, T, \psi)\pi(Z_0, Z, T|\theta)\pi(\theta)\pi(\psi), \tag{2.5}$$

where the π's are densities with respect to suitable dominating measures.

The high level of complexity of the posterior distribution makes of MCMC the only viable solution to explore it. We carefully propose a Gibbs sampling algorithm based on the data augmentation scheme proposed in Rao & Teh (2013). The higher complexity of MSCP when compared to MMPP implies that the full tractability of the algorithm from Rao & Teh (2013) cannot be achieved in our case, so that further non-trivial developments are required. As it is done in Rao & Teh (2013), the CTMC space is extended based on the uniformisation algorithm of Hobolth & Stone (2009).

The Uniformisation algorithm is based on an alternative representation of the CTMC

that extends its space as follows. Let W be a set of times drawn from a homogeneous Poisson process in $[0, S]$ with rate Ω, where $\Omega \geq \max_i\{|[Q_\theta]_{ii}|\}$. The set W constitutes a random discretisation of $[0, S]$ and we shall denote its dimension as $|W|$. Now define $\{V(s)\}_{s\in W}$ as a subordinated (to W) discrete-time Markov chain (sDTMC) with state space E and transition matrix $B = I + \frac{Q_\theta}{\Omega}$. Thus, $V = (V_1, V_2, \cdots, V_{|W|})$ is a sequence of states corresponding to the times W. Note that this representation allows for the occurrence of virtual jumps - when the sDTMC goes from e to e. Finally, we have that (V, W) is a CTMC with state space E and Q-matrix Q_θ. We define $|U|$ as the number of virtual jumps and $U = (U_1, \cdots, U_{|U|})$ as the virtual jump times, which implies that $W = T \cup U$. Figure 2.4 shows an example of this alternative representation of a CTMC.

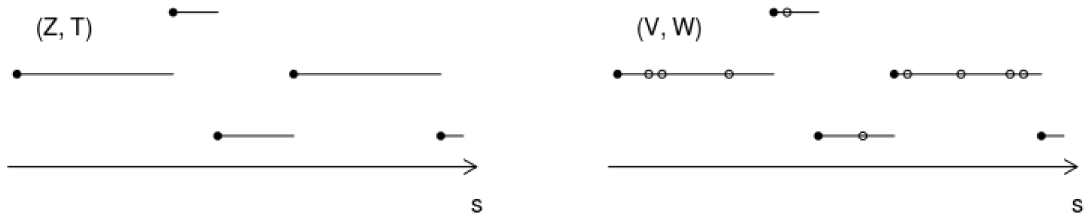


Figure 2.4: Example of a CTMC trajectory. Left: (Z, T) representation. Right: (V, W) representation, circles represent the virtual jumps.

The aim now is to obtain the posterior distribution of $(Z_0, Z, T, \theta, \psi, U, W)$. Note that this could be specified in different ways in terms of its components. For example, although W is fully determined by T and U, making it explicit in (2.6) is crucial to devise and prove the validity of the MCMC algorithm to be proposed. We have that

$$\pi(Z_0, Z, T, \theta, \psi, U, W|y) \propto \pi(y|Z_0, Z, T, \psi)\pi(Z_0, Z, T, U, W|\theta)\pi(\theta)\pi(\psi). \tag{2.6}$$

We define $(Y_1, \cdots, Y_n)$ as the vector of event times from Y in $[0, S]$, where n is the number of observations. The likelihood function $L(Z_0, Z, T, \psi)$ is the density $\pi(Y|Z_0, Z, T, \psi)$ evaluated at the observed y. Note that Y and (θ, U, W) are conditionally independent, given (Z_0, Z, T, ψ). We obtain the density $\pi(Y|Z_0, Z, T, \psi)$ as the Radon-Nikodym derivative of P - the probability measure of a PP with IF λ, w.r.t. the probability measure P_0

of a unit rate PP (see Gonçalves & Franklin, 2019; Andersen et al., 1993):

$$\begin{aligned} L(Z_0, Z, T, \psi) &= \frac{dP}{dP_0}(Y) \\ &\propto e^{-\int_0^S \lambda(s)ds} \prod_{i=1}^{n} \lambda(y_i) \\ &= \exp\left\{ -\int_0^S g_{h_s}(s, \delta_T(s), \psi)ds \right\} \prod_{i=1}^{n} g_{h_{y_i}}(y_i, \delta_T(y_i), \psi). \end{aligned}$$

2.3 The MCMC algorithm

We devise a Gibbs sampling to (approximately) sample from the posterior in (2.6). The following blocking scheme is considered:

$$\{U, W\}, \quad \{Z_0, Z, T, U\}\ , \quad \{\theta\}\ , \quad \{\psi\}.$$

Now note that sampling (Z_0, Z, T, U) from its full conditional posterior distribution is equivalent to sampling from the distribution of $(V_0, V | y, \theta, \psi, W)$. This is the same blocking scheme adopted by Rao & Teh (2013), although it is not made explicit in the paper.

Finally, note that all the full conditional posterior densities are proportional to (2.6).

2.3.1 Sampling (U, W)

We have that

$$\pi(U, W | y, Z_0, Z, T, \theta, \psi) \propto \pi(Z_0, Z, T, U, W | \theta) \propto \pi(U, W | Z_0, Z, T, \theta).$$

Note, from (2.6), that the observations do not depend on the virtual jumps. Also, the full conditional distribution of the virtual jumps depend only on the CTMC trajectory and their parameters and not on the functions g_k's. From Rao & Teh (2013), we have that

$$\pi(U, W | Z_0, Z, T, \theta) \propto \left[\prod_{j=0}^{J} (\Omega + Q_{Z_j})^{|U_j|} \right] e^{-\int_0^S (\Omega + Q_{X(s)})ds}, \tag{2.7}$$

where $|U_j|$ is the number of virtual jumps in $[T_j, T_{j+1})$, with $T_0 = 0$ and $T_{J+1} = S$. To simplify the notation, the diagonal elements of Q_θ are denoted by $Q_i := [Q_\theta]_{ii}$.

The dominating measure in (2.7) is the probability measure of a unit rate PP and,

therefore, the full conditional distribution of (U, W) is a non-homogeneous Poisson process with intensity function $\Omega + Q_{X(s)}$. Moreover, for each interval $[T_j, T_{j+1})$, this distribution is a homogeneous Poisson process with rate $\Omega + Q_{Z_j}$.

Although the augmented representation of the CTMC requires only $\Omega \geq \max_i\{|Q_i|\}$, we actually need to have the strict inequality to guarantee the irreducibility of the MCMC chain (see Rao & Teh, 2013). Details of the algorithm to sample from this full conditional are presented in Appendix A.

2.3.2 Sampling (V_0, V)

We have that

$$\pi(V_0, V|y, \theta, \psi, W) \propto L(V_0, V, W, \psi)\pi_0(V_0)\prod_{j=1}^{|W|}\pi(V_j|V_{j-1}, \theta), \tag{2.8}$$

where $L(V_0, V, W, \psi) = \pi(y|V_0, V, W, \psi)$. Defining $y_{[W_j, W_{j+1})}$ as the observed events in $[W_j, W_{j+1})$, with $W_{|W|+1} = S$. We can factorise the likelihood term as

$$\begin{aligned}\pi(y|V_0, V, W, \psi) &= \pi(y_{[W_0,W_1)}|V_0)\pi(y_{[W_1,W_2)}|V_0, V_1)\cdots\pi(y_{[W_{|W|},S)}|V_0, V_1, \cdots, V_{|W|}) \\ &= L_0(V_0)L_1(V_{0:1})\cdots L_{|W|}(V_0, V),\end{aligned} \tag{2.9}$$

where

$$L_j(V_{0:j}) := \exp\left\{-\int_{W_j}^{W_{j+1}}\lambda(s)ds\right\}\prod_{y_i\in[W_j,W_{j+1})}\lambda(y_i),$$

and all terms in (2.9) are conditional on W and ψ - suppressed for cleanness of notation.

Therefore,

$$\pi(V_0, V|y, \theta, \psi, W) \propto L_0(V_0)L_1(V_{0:1})\cdots L_{|W|}(V_0, V)\pi_0(V_0)\prod_{j=1}^{|W|}\pi(V_j|V_{j-1}, \theta), \tag{2.10}$$

where π_0 and π are densities with respect to the counting measure.

For the Markov modulated Poisson Process, the likelihood terms are independent for each interval $[W_j, W_{j+1})$, that is, $L_1(V_{0:1}) = L_1(V_1)$, $L_2(V_{0:2}) = L_2(V_2)$ and so on. Therefore, a forward-filtering backward-sampling (FFBS) scheme can be devised to sample from the full conditional distribution of (V_0, V), as it is done in Rao & Teh (2013).

In the case of our Markov switching Cox process, the likelihood in the interval $[W_j, W_{j+1})$ depends on the previous effective jump of the CTMC − knowing only V_j is not enough. Remember that the intensity function alternates between different functional forms g_k's and these functions restarts whenever there is an effective jump in V. Therefore, conditioned only on times W, the likelihood in $[W_j, W_{j+1})$ is different for each $(V_{0:j})$ arrangement. As a consequence, an FFBS scheme cannot be devised to sample from this full conditional distribution.

Note, however, that this is a finite dimensional discrete distribution and, therefore, it is analytically possible to compute its probability function. Nevertheless, this requires the computation of the likelihood $L(V_0, V, W, \psi)$ for every possible path of V and, therefore, will typically have an extremely high computational cost - $\mathcal{O}(|E|^{|W|})$. A detailed description of this algorithm is presented in Appendix A.

As an alternative, we propose a Metropolis-Hastings (MH) step in which the proposal distribution samples $(V_j|V_{0:j-1}, \cdot)$ from a probability vector which is proportional to $L(V_{0:j})\pi(V_j|V_{j-1})$, therefore, it can be simulated forward in time, requiring, for each time, the computation of one local likelihood term and one normalising constant. Furthermore, the computation of those normalising constants directly returns the value of the acceptance probability of the MH step. The proposal distribution is as follows.

$$\begin{aligned} q(V_0, V) &= c_0 L_0(V_0)\pi_0(V_0)c_1(V_0)L_1(V_{0:1})\pi(V_1|V_0)\times \\ &\qquad c_2(V_{0:1})L_2(V_{0:2})\pi(V_2|V_1)\cdots c_{|W|}(V_{0:(|W|-1)})L_{|W|}(V_0, V)\pi(V_{|W|}|V_{|W|-1}) \\ &= c_0\pi^*(V_0)c_1(V_0)\pi^*(V_1|V_0)\cdots c_{|W|}(V_{0:(|W|-1)})\pi^*(V_{|W|}|V_{|W|-1}), \end{aligned} \tag{2.11}$$

where π^* is the product of the likelihood term and density π and c_j is the normalising constant of $\pi^*(V_j|V_{j-1})$. Note how the proposal uses information from the data through the local likelihoods, making it a reasonable choice. The acceptance probability of the algorithm is given by

$$\alpha = 1 \wedge \frac{\pi(V_0^*, V^*|\cdot)}{\pi(V_0, V|\cdot)}\frac{q(V_0, V)}{q(V_0^*, V^*)} = 1 \wedge \prod_{j=1}^{|W|}\frac{c_j}{c_j^*}, \tag{2.12}$$

where $c_j = c_j(V_{0:(j-1)})$ and $c_j^* = c_j(V_{0:(j-1)}^*)$.

Note that this MH sub-chain is an independent Metropolis since the proposal does

not depend on the previous iteration of the chain and, therefore, it is either uniformly ergodic or not even geometrically ergodic (Mengersen & Tweedie, 1996). The following proposition establishes that the former is true.

Proposition 2.1. *The Metropolis-Hastings sub-chain defined by (2.11) and (2.12) is uniformly ergodic.*

Proof. As it is shown in (Mengersen & Tweedie, 1996), in order to establish uniform ergodicity for an independent MH chain, it is enough to show that the ratio $\frac{q}{\pi}$ is uniformly bounded away from zero in the state space of π.

$$\frac{q}{\pi}(V_0, V) = \frac{1}{\kappa}\prod_{j=0}^{|W|} c_j > \frac{\beta}{\kappa},$$

where $\kappa > 0$ is the normalising constant of (2.10), $\beta = \min_{\mathcal{V}} \prod_{j=0}^{|W|} c_j$ and $\mathcal{V}$ is the trajectory space of $V|W$. The fact that $c_j > 0, \forall j, \forall V$ completes the proof. □

Further details of the algorithm are presented in Appendix A.

2.3.3 Sampling θ

Let J be the number of jumps of X in $[0, S]$ and define Δ_i as the total time that the CTMC remains in state i, i.e., $\Delta_i = \sum_{j=1;X(T_j)=i}^{J}(T_{j+1} - T_j)$ and let J_i be the number of times that the Markov chain visits state i, so that $\sum_{i=1}^{|E|} J_i - 1 = J$. Finally, let $J_i\bullet = (J_{ij} : j = 1, \cdots, |E|)$ be the vector of the number of jumps from state i to state j and note that $J_{ii} = 0$, $\forall i$.

The parameter vector θ is associated to all entries of the Q-matrix. We define $\theta\bullet = \{\theta_i : i = 1, \cdots, |E|\}$ as the rates of the waiting times and $\theta_i\bullet = \{\theta_{ij} : j = 1, \cdots, |E| \text{ and } j \neq i\}$ as the transition probabilities from i. Thus,

$$\theta = (\theta_1, \cdots, \theta_{|E|}, \theta_{12}, \cdots, \theta_{1|E|}, \theta_{21}, \cdots, \theta_{2|E|}, \cdots, \theta_{|E|1}, \cdots, \theta_{|E|(|E|-1)})$$

and the Q-matrix is given by

$$Q = \begin{bmatrix} -\theta_1 & \theta_1\theta_{12} & \cdots & \theta_1\theta_{1|E|} \\ & & \ddots & \\ & & & -\theta_{|E|} \end{bmatrix}.$$

In order to sample θ, we use a collapsed Gibbs sampling step in which the set of virtual jumps U is integrated out from the full conditional distribution of θ. This is bound to lead to an uniformly better algorithm (see Liu, 1994). We have that

$$\begin{aligned}\pi(\theta|y, Z_0, Z, T, \psi) &\propto \pi(Z_0, Z, T|\theta)\pi(\theta) \\ &\propto \exp\left\{-\sum_{j=1}^{J+1}(T_j - T_{j-1})\theta_{Z_{j-1}}\right\}\left[\prod_{j=1}^{J}\theta_{Z_{j-1}}\theta_{Z_{j-1}Z_j}\right]\pi(\theta). \qquad (2.13)\end{aligned}$$

Prior distributions

Defining $\theta = \{\theta_{\bullet}, \theta_{\bullet\bullet}\}$, where $\theta_{\bullet}$ is the vector of all θ_i's and $\theta_{\bullet\bullet}$ is the vector of all $\theta_{i\bullet}$'s, we assume prior independence between $\theta_{\bullet}$ and $\theta_{\bullet\bullet}$, among the θ_i's and among the $\theta_{i\bullet}$'s, $\forall i \leq |K|$.

The prior density of θ is given by

$$\begin{aligned}\pi(\theta) &= \pi(\theta_{\bullet}, \theta_{\bullet\bullet}) = \pi(\theta_1, \cdots, \theta_{|E|}, \theta_{1\bullet}, \cdots, \theta_{|E|\bullet}) \\ &= \left[\prod_{i=1}^{|K|} f_G(\theta_i; \alpha_i, \beta_i)\mathbb{I}\left\{\theta_i = \theta_{i+|K|}\right\} f_D(\theta_{i\bullet}; \gamma_{i1}, \cdots, \gamma_{i|K|})\mathbb{I}\left\{\theta_{i(i+|K|)} = \theta_{(i+|K|)i}\right\}\right] \times \\ &\qquad \left[\prod_{i=1}^{|K|}\prod_{j=1, j\neq i}^{|K|} \mathbb{I}\left\{\theta_{ij} = \theta_{(i+|K|)j}\right\}\mathbb{I}\left\{\theta_{i(j+|K|)} = \theta_{(i+|K|)(j+|K|)} = 0\right\}\right],\end{aligned}$$

where f_G is the probability density function of a Gamma distribution and f_D is the probability density function of a Dirichlet distribution. The indicator functions give the dependence structure of θ_i's and $\theta_{i\bullet}$'s when there is duplication of states to allow some funcional form restart itself. For model parsimony and identifiability, the respective entries of the Q-matrix are set to be zero so that a change between two different functional forms is allowed only through one of the first $|K|$ states in E.

In order to illustrate the use of those indicator functions, consider a MSCP with $K = \{1, 2\}$ and $E = \{1, 2, 3, 4\}$, where $h(1) = h(3) = 1$ and $h(2) = h(4) = 2$. The

Q-matrix is then written as

$$\begin{bmatrix} -\theta_1 & \theta_1\theta_{12} & \theta_1\theta_{13} & 0 \\ \theta_2\theta_{21} & -\theta_2 & 0 & \theta_2\theta_{24} \\ \theta_1\theta_{13} & \theta_1\theta_{12} & -\theta_1 & 0 \\ \theta_2\theta_{21} & \theta_2\theta_{24} & 0 & -\theta_2 \end{bmatrix}.$$

It is expected that several of the models to be fitted will be likely to have few changes in the underlying CTMC. This means that not much information about parameters θ_i's may be available in the data. In those cases, it is highly recommended to adopt reasonably informative priors, based on the scale of the expected behavior of the waiting times, in order to avoid a non-smooth posterior and, consequently, an unstable MCMC algorithm. This issue will be illustrate in the simulated examples in Chapter 4.

The full conditional distributions

The prior independence of $\theta_{\bullet}$ and $\theta_{\bullet\bullet}$ implies in their independence under the full conditional measure. We then have that

$$\pi(\theta_{\bullet}|y, Z_0, Z, T, \psi) \propto \left[\prod_{i=1, i\neq e}^{|K|} \theta_i^{J_i + J_{i+|K|} + \alpha_i - 1} e^{-\theta_i\left(\Delta_i + \Delta_{i+|K|} + \beta_i\right)} \mathbb{I}\{\theta_i = \theta_{i+|K|}\}\right] \times$$
$$\theta_e^{J_e + J_{e+|K|} + \alpha_e - 2} e^{-\theta_e\left(\Delta_e + \Delta_{e+|K|} + \beta_e\right)} \mathbb{I}\{\theta_e = \theta_{e+|K|}\},$$

where g_e, $e \leq |K|$, is the last visited functional form of the intensity function. If the model does not allow for the IF to jump from g_i to itself, we assume $J_{i+|K|} = \Delta_{i+|K|} = 0$. Therefore, the θ_i's, $i \leq |K|$, are conditionally independent with distribution $\theta_i|\cdot \sim \text{Gamma}(\alpha_i + J_i + J_{i+|K|}, \beta_i + \Delta_i + \Delta_{i+|K|})$, if $i \neq e$, $\theta_i|\cdot \sim \text{Gamma}(\alpha_e + J_e + J_{e+|K|} - 1, \beta_e + \Delta_e + \Delta_{e+|K|})$, if $i = e$.

Furthermore,

$$\pi(\theta_{\bullet\bullet}|y, Z_0, Z, T, \psi) \propto$$
$$= \left[\prod_{i=1}^{|K|}\prod_{j=1, j\neq i}^{|K|} \theta_{ij}^{J_{ij}+J_{(i+|K|)j}+\gamma_{ij}-1}\mathbb{I}\left\{\theta_{ij} = \theta_{(i+|K|)j}\right\}\mathbb{I}\left\{\theta_{i(j+|K|)} = \theta_{(i+|K|)(j+|K|)} = 0\right\}\right] \times$$
$$\left[\prod_{i=1}^{|K|} \theta_{i(i+|K|)}^{J_{i(i+|K|)}+J_{(i+|K|)i}+\gamma_{i(i+|K|)}-1}\mathbb{I}\left\{\theta_{i(i+|K|)} = \theta_{(i+|K|)i}\right\}\right].$$

If the model does not allow for the IF to jump from g_i to itself, we assume $J_{i(i+|K|)} = J_{(i+|K|)i} = J_{j(i+|K|)} = J_{(i+|K|)j} = 0$, for all $j \leq |K|$. Then, the $\theta_{i\bullet}$'s, $i \leq |K|$, are conditionally independent with distribution $\theta_{i\bullet}|\cdot \sim \text{Dirichlet}(J_{i1}+J_{(i+|K|)1}+\gamma_{i1}, \cdots, J_{i|K|}+J_{(i+|K|)|K|}+\gamma_{i(|K|-1)}, J_{i(i+|K|)}+J_{(i+|K|)i}+\gamma_{i|K|})$.

The algorithms to sample from (2.13) are described in Appendix A.

2.3.4 Sampling ψ

We have that

$$\pi(\psi|y, Z_0, Z, T, \theta, U, W) \propto L(Z_0, Z, T, \psi)\pi(\psi)$$
$$\propto e^{-\int_0^S \lambda(s)ds}\left[\prod_{i=1}^{n}\lambda(y_i)\right]\pi(\psi). \tag{2.14}$$

We assume prior independence of the ψ_k's, $\forall k \in K$ and, if g_k is a constant than a Gamma(α_k, β_k) distribution is a conjugated prior for ψ_k. Otherwise, no conjugated closed-form analysis is available and we will typically consider improper uniform priors.

The prior independence of the ψ_k's implies in their independence under the full conditional measure. We then have that, if g_k is a constant,

$$\pi(\psi_k|\cdot) = e^{-\psi_k(\Delta_k+\beta_k)}\psi_k^{n_k+\alpha_k-1} \sim \text{Gamma}(n_k+\alpha_k, \Delta_k+\beta_k),$$

where n_k is the number of Poisson events occurring during the time period that the IF assumes the functional form g_k.

If g_k is not a constant, it is sampled in a Gaussian random walk Metropolis step with proposal covariance matrix Σ_k. The choice of this matrix is crucial to have a good mixing

of the chain and we shall use a strategy based on an adaptive Metropolis algorithm proposed in Roberts & Rosenthal (2009) and it is discussed in Appendix A. The acceptance probability of a move from ψ_k to ψ_k^* is given by

$$\alpha = 1 \wedge \frac{L(Z_0, Z, T, \psi_k^*)\pi(\psi_k^*)}{L(Z_0, Z, T, \psi_k)\pi(\psi_k)}. \tag{2.15}$$

2.4 Improving the proposed MCMC algorithm

The MCMC algorithm proposed in the previous section is liable to have serious mixing problems as S increases. We propose two adaptations to overcome this issue. Firstly, we propose an adaptation to the model augmentation based on uniformisation approach. Secondly, we adapt the sampling step for (V_0, V) by considering a carefully chosen (random) partition of this vector.

2.4.1 Alternative model augmentation scheme

We propose an adaptation of the model augmentation scheme based on the uniformisation representation of a CTMC. As it was mentioned before, in order to have an irreducible MCMC algorithm, we require $\Omega > \max_i |Q_i|$. Now note that, the greater is the value of Ω, the greater the expected number of virtual jumps is and, in turn, the better is the mixing of the MCMC chain. On the other hand, the greater is the number of virtual jumps, the higher is the computational cost to sample (V_0, V). Rao & Teh (2013) provide empirical evidence to suggest that $\Omega = 2\max_i |Q_i|$ is a reasonable choice. Note, however, that the choice of Ω depends only on the state corresponding to the smallest expected waiting time and, therefore, leads to different local mixing properties of the MCMC algorithm with respect to different CTMC states. Based on that, we propose an alternative model augmentation scheme, aiming at a global optimisation of the efficiency of the sampling step for (V_0, V), that basically considers a state-wise adaptation of the uniformisation

idea. More specifically, we define

$$\begin{aligned} V_0 &\sim \pi_0, \\ (W_1|V_0 = i) &\sim \text{Exponential}(\Omega_i), \\ (V_1|V_0 = i) &\sim B_{i\bullet} = \mathbf{1}_i + \frac{1}{\Omega_i} Q_{i\bullet}, \\ (W_2 - W_1|V_1 = j, W_1) &\sim \text{Exponential}(\Omega_j), \\ (V_2|V_1 = j) &\sim B_{j\bullet} = \mathbf{1}_j + \frac{1}{\Omega_j} Q_{j\bullet}, \\ &\vdots \end{aligned} \tag{2.16}$$

where $\Omega_i \geq |Q_i|$, $Q_{i\bullet}$ is the i-th row of Q_θ and $B_{i\bullet}$ is a probability vector such that $\mathbf{1}_i$ is a vector of zeros with the i-th element being equal to one. $Q_{i\bullet}$ is the i-th row of Q_θ.

Proposition 2.2. *For any $\Omega_i \geq |Q_i|$, the process (V_0, V, W) defined in (2.16) is a valid augmented representation of the CTMC (Z_0, Z, T), which has Q-matrix Q_θ and initial distribution π_0.*

Proof. Let $W_{(1)}$ be the first non-virtual jump from W and $V_{(1)}$ the state of V at $W_{(1)}$. The density of $(W_{(1)}|V_0)$ with respect to the Lebesgue measure is

$$\begin{aligned} \pi_{W_{(1)}|V_0=i}(t) &= \sum_{n=1}^{+\infty} \pi_{W_{(1)}}(t|V_{0:n-1} = i, V_n \neq i) P(V_{1:n-1} = i, V_n \neq i|V_0 = i) \\ &= e^{-\Omega_i t} \Omega_i (1 - B_{ii}) \sum_{k=0}^{+\infty} \frac{(\Omega_i t B_{ii})^k}{K!} \\ &= \Omega_i (1 - B_{ii}) e^{-\Omega_i (1 - B_{ii}) t} = |Q_i| e^{-|Q_i| t} \sim \text{Exponential}(|Q_i|),\ i \in E. \end{aligned}$$

Analogous calculations lead to $(W_{(j)} - W_{(j-1)}|V_{(j-1)} = i) \sim \text{Exponential}(|Q_i|),\ i \in E,\ j \in J$.

We also have that

$$\begin{aligned} P(V_{(1)} = j|V_0 = i) &= \sum_{n=1}^{+\infty} P(V_n = j|V_{0:n-1} = i) P(V_{0:n-1} = i|V_0 = i) \\ &= \sum_{n=1}^{+\infty} B_{ij} B_{ii}^{n-1} = B_{ij} \frac{1}{1 - B_{ii}} = -\frac{Q_{ij}}{Q_{ii}}, \forall j \neq i \in E \end{aligned}$$

and analogous calculations establish the result for $(V_{(n)}|V_{(n-1)})$. □

Finally, note that the irreducibility of the MCMC chain is achieved by making $\Omega_i > |Q_i|$.

Now note that, under this new augmentation scheme, we have that

$$\begin{aligned}
\pi(U, W|Z_0, Z, T) &= \frac{\pi(Z_0, Z, T, U, W)}{\pi(Z_0, Z, T)} \\
&= \frac{\pi_0(Z_0)\left[\prod_{j=0}^{J-1}\pi(U^j, T_{j+1}, Z_{j+1}|Z_j, T_j)\right]\pi(U^J, I_S|Z_J, T_J)}{\pi_0(Z_0)\left[\prod_{j=0}^{J-1}\pi(T_{j+1}, Z_{j+1}|Z_j, T_j)\right]\pi(I_S|Z_J, T_J)} \\
&= \frac{\left[\prod_{j=0}^{J-1}\pi(Z_{j+1}|Z_j)\pi(T_{j+1}|Z_j, T_j)\pi(U^j|Z_j, T_j, T_{j+1})\right]\pi(I_S|Z_J, T_J)\pi(U^J|Z_J, T_J, I_S)}{\left[\prod_{j=0}^{J-1}\pi(Z_{j+1}|Z_j)\pi(T_{j+1}|Z_j, T_j)\right]\pi(I_S|Z_J, T_J)} \\
&= \left[\prod_{j=0}^{J-1}\pi(U^j|Z_j, T_j, T_{j+1})\right]\pi(U^J|Z_J, T_J, I_S), \qquad (2.17)
\end{aligned}$$

where U_j is the set of virtual jumps in $[T_j, T_{j+1})$ and $I_S := \mathbb{I}(T_{J+1} > S)$ so that $P(I_S = 1|T_J) = e^{-Q_{Z_J}(S-T_J)}$. Equation (2.17) establishes the conditional independence of U among intervals $[T_j, T_{j+1})$. Furthermore, for $j < J$, the full conditional density of U^j w.r.t. the measure of a unit rate Poisson process is given by

$$\begin{aligned}
\pi(U^j|Z_j, T_j, T_{j+1}) &= \frac{\pi(U^j, T_{j+1}|Z_j, T_j)}{\pi(T_{j+1}|Z_j, T_j)} \\
&= \frac{\Omega_{Z_j}^{|U_j|+1} e^{-\Omega_{Z_j}(T_{j+1}-T_j)} B_{Z_jZ_j}^{|U_j|}(1 - B_{Z_jZ_j})}{e^{-(T_{j+1}-T_j)}|Q_{Z_j}|e^{-|Q_{Z_j}|(T_{j+1}-T_j)}} \\
&= \frac{\Omega_{Z_j}^{|U_j|+1} e^{-\Omega_{Z_j}(T_{j+1}-T_j)} \left(1 - \frac{|Q_{Z_j}|}{\Omega_{Z_j}}\right)^{|U_j|} \left(\frac{|Q_{Z_j}|}{\Omega_{Z_j}}\right)}{e^{-(T_{j+1}-T_j)}|Q_{Z_j}|e^{-|Q_{Z_j}|(T_{j+1}-T_j)}} \\
&= \left(\Omega_{Z_j} + Q_{Z_j}\right)^{|U_j|} \frac{e^{-\left(\Omega_{Z_j}+Q_{Z_j}\right)(T_{j+1}-T_j)}}{e^{-(T_{j+1}-T_j)}},
\end{aligned}$$

which implies that this is a homogeneous Poisson process with rate $\Omega_{Z_j} + Q_{Z_j}$. For the

last interval $[T_J, S]$, we have

$$\pi(U^J|Z_J, T_J, I_S) = \frac{\pi(U^J, I_S|Z_J, T_J)}{\pi(I_S|Z_J, T_J)} = \frac{\Omega_{Z_J}^{|U_J|} e^{-\Omega_{Z_J}(S-T_J)} B_{Z_J Z_J}^{|U_J|}}{e^{-(S-T_J)} e^{-|Q_{Z_J}|(S-T_J)}}$$

$$= (\Omega_{Z_J} + Q_{Z_J})^{|U_J|} \frac{e^{-\left(\Omega_{Z_J}+Q_{Z_J}\right)(S-T_J)}}{e^{-(S-T_J)}},$$

which implies that this is a homogeneous Poisson process with rate $\Omega_{Z_J} + Q_{Z_J}$. It is then straightforward to adapt the Algorithm 1 in Appendix A to sample from the full conditional distribution of (U, W).

2.4.2 A blocking scheme to sample (V_0, V)

Given that an independent proposal distribution is adopted for (V_0, V), the greater its acceptance rate is, the better is the mixing properties of the algorithm. Nevertheless, such rate is bound to get smaller as S gets bigger and, in turn, the efficiency of the MCMC algorithm may be compromised. We mitigate this problem by sampling (V_0, V) in blocks, defined by sub-intervals of $[0, S]$. This is not straightforward though. Firstly, there is a trade off in terms of the number of sub-intervals to be considered since more blocks imply in a higher autocorrelation of the chain. Secondly, a naive strategy to partition $[0, S]$ may defy the original purpose of the blocking strategy. For example, the partition must vary along the iterations of the chain in order to guarantee irreducibility of the chain.

In order to take full computational advantage of the blocking scheme, we adopt two strategies to guarantee that the full conditional distribution of each sub-interval depends on the likelihood function only inside that interval. First, the partition of S is chosen so that the sub-interval edges are of the form $[W_{(b-1)}, W_{(b)})$, where $W_{(b-1)}, W_{(b)} \in T$ which mean that the edges of the sub-intervals are effective jumps of the CTMC X. Second, the proposal distribution has the restriction that $W_{(b)}$ is an actual change (i.e. not a virtual jump). Details concerning when and how to partition $[0, S]$ are discussed in Appendix A. Figure 2.5 shows an example of how the complete observed interval could be split using the first strategy.

Let B be the number of blocks, $[W_{(b-1)}, W_{(b)})$ be the b-th block, $V^b = (V_{(b-1)}, \cdots, V_{(b)-1})$ and $V^{-b} = V \setminus V^b$. If we consider the example in Figure 2.5, we have $B = 3$, $[W_{(0)}, W_{(1)}) = [W_0, W_5)$, $[W_{(1)}, W_{(2)}) = [W_5, W_{10})$, $[W_{(2)}, W_{(3)}) = [W_{10}, W_{13})$ and $V^2 = (V_5, V_6, V_7, V_8, V_9)$.

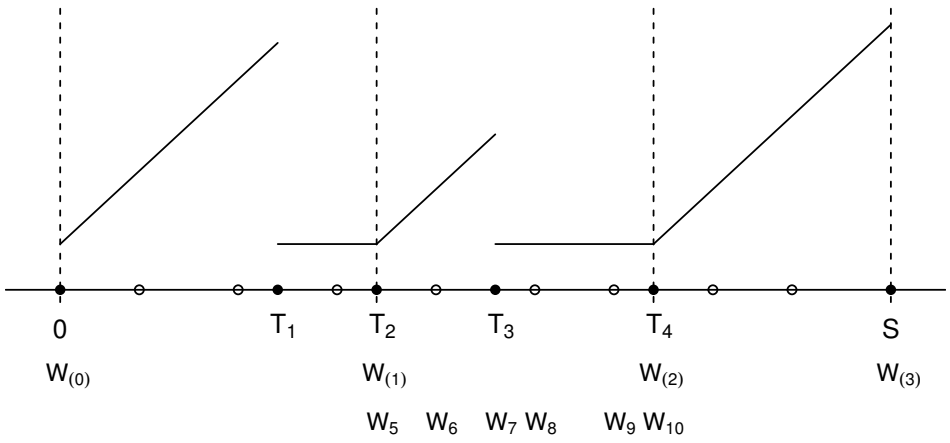


Figure 2.5: Example of a possible partition of the observed interval. Solid line is the IF, the dotted lines define the partition and dots are the set W.

We have that

$$\pi(V^b|y,\theta,\psi,W,V^{-b}) \propto \prod_{j=(b-1)}^{|W|} L_j(V_{0:j}) \prod_{j=(b-1)}^{(b)} \pi(V_j|V_{j-1},\theta). \tag{2.18}$$

The proposal distribution is the one in (2.11) but with the restriction $V_{(b)-1} \neq V_{(b)}$, i.e.,

$$q(V^b) = \left[\prod_{j=(b-1)}^{(b)-1} c_j \pi^*(V_j|V_{j-1},\theta)\right] \pi(V_{(b)}|V_{(b)-1},\theta)\mathbb{I}(V_{(b)-1} \neq V_{(b)}), \tag{2.19}$$

where $c_j = c_j(V_{0:j-1})$ is the normalising constant of π^* and the last constant $c_{(b)-1}$ normalises the product $\pi^*(V_{(b)-1}|V_{(b)-2},\theta)\pi(V_{(b)}|V_{(b)-1},\theta)\mathbb{I}(V_{(b)-1} \neq V_{(b)})$. The acceptance probability of a move V^b to V^{b*} is then given by

$$\alpha = 1 \wedge \prod_{j=(b-1)}^{(b)-1} \frac{c_j}{c_j^*} \tag{2.20}$$

and the uniform ergodicity of the new MH step is preserved.

Another possibility to sample from each block is to consider the direct computation of the probability mass function of the respective full conditional distributions whenever $|E|^{|W_b|}$ is not too large.

Chapter 3

Markov switching Cox processes with varying starting value

The model described in Chapter 2 has the constraint that each functional form has a unique (maybe unknown) starting value. In order to make the model more flexible for real applications, we propose an extension of the MSCP proposed in Chapter 2 to allow for the starting value of each functional form to vary.

3.1 The proposed model

We consider the same notation defined in the previous chapter and define the sequence of starting values $R = (R_0, R_1, \cdots, R_J)$ of the IF, i.e., $R_j = \lambda(T_j)$. Additionally, $R_T(s) := R_{max\{i:T_i \leq s\}}$, that is, $R_T(s)$ is the starting value of the functional form assumed at time s. The Markov switching Cox process with varying starting value (MSCPV) is defined as follows:

$$
\begin{aligned}
Y &\sim \text{NHPP}(\lambda), && (3.1)\\
\lambda(s) &= g_{h_s}(s, \delta_T(s), R_T(s), \psi), && (3.2)\\
h: &\ E \to K, && (3.3)\\
(R_j | h(Z_j) = k) &\overset{ind.}{\sim} \pi_k, \ \forall j, k, && (3.4)\\
X &\sim \text{CTMC}(\pi_0, Q_\theta, E). && (3.5)
\end{aligned}
$$

where π_k is some suitable prior for the starting values of g_k.

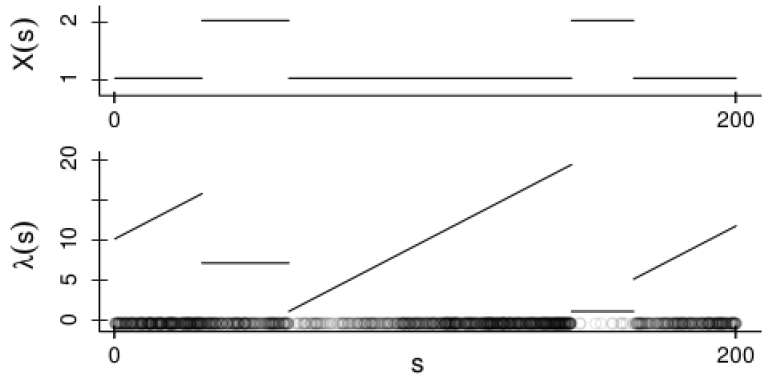


Figure 3.1: An example of a MSCPV. Top: the CTMC trajectory. Bottom: the solid line represents the MSCPV intensity function λ and the circles are the observations of the PP.

Whilst the new model is more flexible than that proposed before, it may also lead to simplifications. For example, in order to consider $m \in \mathbb{N}$ distinct constant levels for the IF in the model of Chapter 2, m g_k's would have to be added to the model whereas, for the new model, only one g_k is enough. Figure 3.1 shows an example of a MSCPV with $E = K = \{1, 2\}$, $g_1 = R_T(s) + 0.2(s - \delta_T(s))$ and $g_2 = R_T(s)$.

3.2 MCMC

We consider the MCMC algorithm proposed in Chapter 2, with the required adaptations, to perform Bayesian inference for the family of MSCPV models. The target posterior density is now given by

$$\pi(Z_0, Z, T, R, \theta, \psi, U, W|y) \propto \pi(y|Z_0, Z, T, R, \psi)\pi(Z_0, Z, T, U, W|\theta)\times \\ \pi(R|Z_0, Z)\pi(\theta)\pi(\psi) \tag{3.6}$$

and we consider the following blocking scheme for the Gibbs sampling algorithm:

$$\{U, W\},\ \{V_0, V, R\},\ \{\theta\},\ \{\psi\}.$$

The algorithms to sample $\{U, W\}$, $\{\theta\}$ and $\{\psi\}$ are the same as before. In the case of $\{V_0, V, R\}$, we have that

$$\pi(V_0, V, R|\cdot) \propto L_0(V_0, R_0)\pi_0(V_0)\prod_{j=1}^{|W|} L_j(V_{0:j}, R)\pi(V_j|V_{j-1}, \theta)\pi(R_j|V_j). \tag{3.7}$$

We consider an adaptation of the MH algorithm proposed in Chapter 2 with proposal distribution given by

$$q(V_0, V, R) = c_0 \frac{\pi_0(V_0)}{c_0(V_0)} c_0(V_0) L_0(V_0, R_0) \pi(R_0|V_0, \tau) \times \prod_{j=1}^{|W|} c_j \frac{\pi(V_j|V_{j-1}, \theta)}{c_j(V_j)} c_j(V_j) L_j(V_{0:j}, R) \left(\pi(R_j|V_j, \tau)\mathbb{I}(V_{j-1} \neq V_j) + \mathbb{I}(V_{j-1} = V_j)\right) \tag{3.8}$$

where $c_j(V_j)$ is the normalising constant of $L_j(V_{0:j}, R)\pi(R_j|V_j, \tau)$ and c_j is the normalising constant of $\frac{\pi(V_j|V_{j-1},\theta)}{c_j(V_j)}$. Note that when V_j corresponds to a virtual jump, there is no R_j variable and $c_j(V_j)$ is the constant that normalises the corresponding likelihood term.

Unless R_j is the starting value of a constant function with a Gamma prior, the normalising constants $c_j(V_j)$ will typically be intractable. For that reason, we consider Gamma or discrete priors for the starting values of a constant g_k and discrete priors for the remaining functional forms. Moreover, unless useful information is available, we assume uniform discrete priors with supports chosen according to the scale of the IF.

We sample from (3.8) forward in time and accept a proposal (V_0^*, V^*, R^*) with probability

$$\alpha = 1 \wedge \frac{\pi(V_0^*, V^*, R^*|\cdot)}{\pi(V_0, V, R|\cdot)} \frac{q(V_0, V, R)}{q(V_0^*, V^*, R^*)} = 1 \wedge \prod_{j=1}^{|W|} \frac{c_j}{c_j^*}. \tag{3.9}$$

The algorithm to sample from (3.7) is described in Appendix A.

The same algorithm can be applied to blocks of (V_0, V, R) defined by suitably chosen sub-intervals. The blocks are defined in the same way as it is described in Section 2.4. We define $R^b = (R_{(b-1)}, \cdots, R_{(b)-1})$ and $R^{-b} = R \setminus R^b$. The proposal density is given by

$$q(V^b, R^b) = \prod_{j=(b-1)}^{(b)-1} c_j \frac{\pi(V_j|V_{j-1}, \theta)}{c_j(V_j)} c_j(V_j) L_j(V_{0:j}, R) \left(\pi(R_j|V_j, \tau)\mathbb{I}(V_{j-1} \neq V_j) + \mathbb{I}(V_{j-1} = V_j)\right) \times \pi(V_{(b)}|V_{(b)-1}, \theta)\mathbb{I}(V_{(b)-1} \neq V_{(b)}),$$

and the acceptance probability is the same as (3.9), considering the corresponding subset of terms in the product.

3.2.1 A more flexible approach

An extra level of hierarchy can be consider for the priors π_k. In the case of the Gamma prior, its parameters τ_k are assumed to be unknown and modeled according to some suitable prior. A Gaussian random walk MH step can be used to sample from the full conditional distribution of those parameters, which is proportional to $\pi(R|Z_0, Z, \tau)\pi(\tau)$.

In the case of discrete priors, we need to set the size of the support - say s_k, so that its elements $\tau_k = \{\tau_{k1}, \ldots, \tau_{ks_k}\}$ are assumed to be unknown. A continuous joint prior is adopted for τ with the truncation $\tau_{k1} < \ldots < \tau_{ks_k}$ to avoid label switching problems. Finally, a Gaussian random walk MH step is performed to update the components of τ_k that are actually assumed as values of the R_j's in the current iteration of the MCMC chain.

Chapter 4

Simulated studies and data analysis

In this chapter we present some analyses for simulated and real data to investigate the efficiency of the methodology developed in this thesis. The algorithms are coded in Ox (Doornik, 2009) and run in a CPU Intel Core i7-3770, 3.40GHz x 8.

4.1 Markov switching Cox processes

We consider several scenarios to apply the proposed methodology. Table 4.1 presents the specifications of the simulated data.

		Time interval	n	K	E	g_k's
Scenario 1	A	$[0, 100]$	346	$\{1, 2\}$	$\{1, 2\}$	$g_1 = 1 + 0.2(s - \delta_T(s))$ $g_2 = 1$
	B	$[0, 200]$	1158			
	C	$[0, 500]$	3016			
	D	$[0, 1000]$	5026			
Scenario 2	A	$[0, 200]$	592	$\{1, 2\}$	$\{1, 2, 3\}$	$g_1 = 1 + 0.2(s - \delta_T(s))$ $g_2 = 1$
	B	$[0, 500]$	1922			
Scenario 3	A	$[0, 200]$	999	$\{1, 2, 3\}$	$\{1, 2, 3\}$	$g_1 = 1 + 0.2(s - \delta_T(s))$ $g_2 = 15 - 0.2(s - \delta_T(s))$ $g_3 = 1$
	B	$[0, 500]$	3209			
Scenario 4	A	$[0, 200]$	1361	$\{1, 2\}$	$\{1, 2, 3, 4\}$	$g_1 = 0.2(s - \delta_T(s))$ $g_2 = 15 - 0.2(s - \delta_T(s))$
	B	$[0, 500]$	3821			
Scenario 5	A	$[0, 200]$	738	$\{1, 2\}$	$\{1, 2, 3, 4\}$	$g_1 = 1 + 0.1(s - \delta_T(s))$ $g_2 = 12e^{-0.1(s-\delta_T(s))}$
	B	$[0, 500]$	2204			

Table 4.1: Specifications of the different scenarios to be considered.

As we have discussed in Section 2.3.3, informative priors for parameters $\theta_{\bullet}$ are usually required to obtain good results. This is a way to mitigate the problem of having low information about those parameters due to the fact that the intensity function visits

each functional form only a few times. Those priors are informative in the sense of simply setting the scale of the problem by basically avoiding very short visiting times and parameter values for the ψ_k's in total disagreement with the scale of the IF. For scenarios with short time intervals ($[0, 100]$ and $[0, 200]$) we used informative prior for at least one of θ_k's, in particular, we set a Gamma$(1, 10)$ prior for those parameters. This means that, at prior, the IF remains in its k-th functional form on average ten units of time. The Figure B.21 in Appendix B shows the density of this prior.

For the examples presented here, the functional forms are chosen to be the same ones used to generate the data. In a real data analysis we shall use the knowledge of an expert or with no prior knowledge about the process, we recommend to base the model specification on some empirical estimation of the IF.

To obtain the estimated IF we define a grid on the observed interval, with size 1000, then the posterior mean and credibility interval are computed for each point in this grid. Figures 4.1 and 4.2 show the estimated IF for scenario 1A and 5B, respectively. The results for all the other scenarios can be found in Appendix B. Results show a good recovery of the IF. It supports that our algorithm works well to estimate the IF of a MSCP with different specifications.

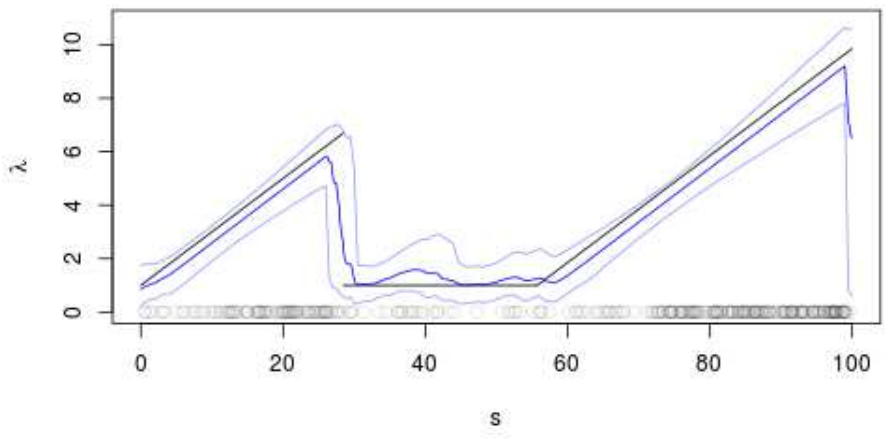


Figure 4.1: Scenario 1A. Blue lines: posterior mean and 95% credibility interval for the IF. Black line: true IF. Circles: data.

We assess the convergence properties of the MCMC chains through the autocorrelation structure of suitable unidimensional chains. We compute the inefficiency factor for each

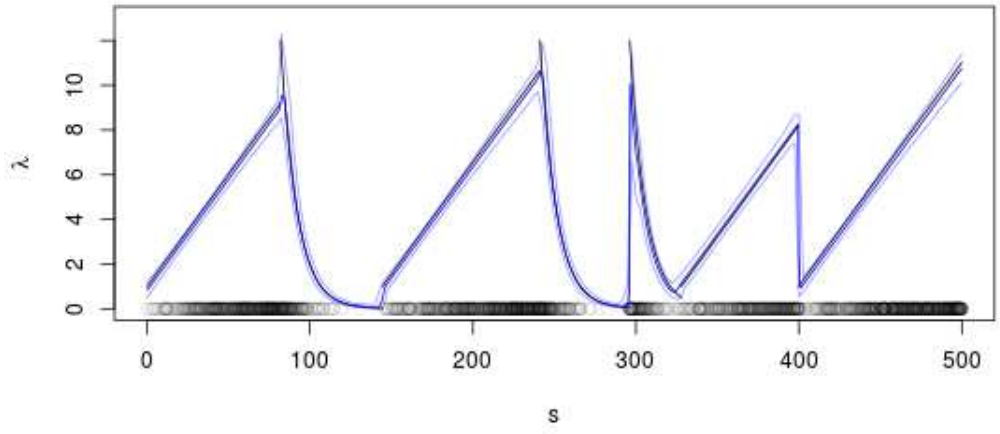


Figure 4.2: Scenario 5B. Blue lines: posterior mean and 95% credibility interval for the IF. Black line: true IF. Circles: data.

of these chains, which is given by

$$1 + 2\sum_{k=1}^{\infty} \rho_k, \tag{4.1}$$

where ρ_k is the autocorrelation of lag k. The effective sample size of a chain of length n is given by

$$n_{\text{eff}} = \frac{n}{1 + 2\sum_{k=1}^{\infty} \rho_k}. \tag{4.2}$$

Thus, n_{eff} represents the size of an i.i.d. sample which has the same variance for the Monte Carlo estimator given by the ergodic average of the chain.

We perform the convergence analysis for the chain of the log posterior density, which is a good unidimensional summary of the whole chain. To calculate the inefficiency factor and the effective sample size we shall use the estimated autocorrelations and truncate the summation term in (4.1) and (4.2) up to the highest lag with significant autocorrelation.

It is not possible to assert that there is a single feature that influences autocorrelation of the chain and computational time. Naturally, the computational time is impacted by number of observations and observed interval length. The total computational time spent to obtain all estimates can be seen in Appendix B.

The highest autocorrelation structure was observed for scenario 5 with observed interval $[0, 500]$ (see Table 4.2 and Figure B.19). Other scenarios with high autocorrelations are 1C, 1D, 4A and 4B (see the ACFs in Appendix B). Although the algorithm may generate chains with higher autocorrelations, this is compensated by the low computational

		Ineff.	n_{eff}	Time for 100 eff.
Scenario 1	A	36.585	820	2.94s
	B	21.413	934	4.26s
	C	72.539	965	14.98s
	D	61.881	808	27.07s
Scenario 2	A	26.596	940	3.53s
	B	16.892	888	6.41s
Scenario 3	A	49.342	912	7.61s
	B	45.767	874	20.40s
Scenario 4	A	64.935	770	14.84s
	B	40.706	737	32.95s
Scenario 5	A	46.536	967	6.86s
	B	181.644	523	58.37s

Table 4.2: Statistics for all the scenarios: inefficiency factor (Ineff.), effective sample size and computational time spent to obtain an effective sample size of 100, in seconds.

costs, making this feasible in many applications. For example, in scenario 5B, an effective sample size of 100 is obtained in less than a minute (see Table 4.2).

The Table 4.3 presents some posterior statistics for all parameters of scenarios 1A and 5B. In particular, we have: mean, standard deviation (SD), percentile 2.5 ($\text{CI}_{0.025}$) and 97.5 ($\text{CI}_{0.975}$), respectively. In addition, in Appendix B, the Table B.1 presents these posterior statistics for the remaining scenarios. Note that these estimates were obtained with respect to the effective sample sizes present in Table 4.2. Generally, the initial values of the g_k's have their posterior means different from their true values and high standard deviations, due to the variance of a Poisson process. Nevertheless all IF's were satisfactorily recovered.

Scenario	Parameter	True value	Mean	SD	$\text{CI}_{0.025}$	$\text{CI}_{0.975}$
1 A	ψ_{11}	1	0.7489	0.3479	0.1941	1.5563
	ψ_{12}	0.2	0.2016	0.0254	0.1513	0.2520
	ψ_{21}	1	1.1330	0.3855	0.3474	1.9464
	θ_1	0.025	0.0582	0.0339	0.0103	0.1404
	θ_2	0.05	0.3267	0.2963	0.0296	1.0834
5 B	ψ_{11}	1	0.8156	0.1741	0.4893	1.1668
	ψ_{12}	0.1	0.0995	0.0041	0.0914	0.1074
	ψ_{21}	0	0.0362	0.0504	-0.0313	0.1601

Table 4.3: Posterior statistics.

Scenario	Parameter	True value	Mean	SD	$CI_{0.025}$	$CI_{0.975}$
	ψ_{22}	12	10.3962	1.3980	7.7258	13.1853
	ψ_{23}	-0.1	-0.1050	0.0095	-0.1254	-0.0883
	θ_1	0.02	0.0123	0.0061	0.0034	0.0267
	θ_2	0.02	0.0296	0.0148	0.0080	0.0652
5 B	θ_{12}	0.5	0.6188	0.2050	0.2001	0.9567
	θ_{13}	0.5	0.3812	0.2050	0.0433	0.7999
	θ_{21}	0.5	0.6298	0.2129	0.1986	0.9717
	θ_{24}	0.5	0.3702	0.2129	0.0283	0.8014

Table 4.3: Posterior statistics.

As expected, the posterior standard deviations for CTMC transition probability parameters are high and their credibility interval are wide. The same is true for CTMC waiting time parameters. This is due to the lack of information on the data. Comparing processes with the same specifications, note how the length of the credibility intervals decrease as the observed intervals increase. Trace plots in Appendix B, suggest good convergence.

Generally speaking, the results obtained in this section strongly suggest that the proposed methodology is very efficient to sample from the posterior distribution. It is efficient in the sense of having good estimates and being a very fast algorithm.

4.2 Markov switching Cox processes with varying starting values

We now consider examples with varying stating values. Scenario 6 considers the following: time interval $[0, 200]$, $K = \{1, 2\}$, $E = \{1, 2, 3, 4\}$, $g_1 = 1 + 0.2(s - \delta_T(s))$ and $g_2 = R_T(s)$. A data set with 1014 events was generated. We only allow the constant functional form to have a varying stating values, for which a discrete uniform prior on $\{0.1, 0.3, \cdots, 5.9, 6.1\}$ is assumed. Informative priors for θ are now imperative, give the extra level of model flexibility. Remember that the priors are informative in the sense of simply setting the scale of the observed interval and waiting times. In particular, we want to avoid lots of

jumps among different constant levels of the IF.

The Figure 4.3 shows the real and estimated IF. The Table 4.4 presents the posterior statistics for all the parameters. The same issue for the initial value of g_1 as before is observed here, but this does not affect the efficiency in estimating the IF.

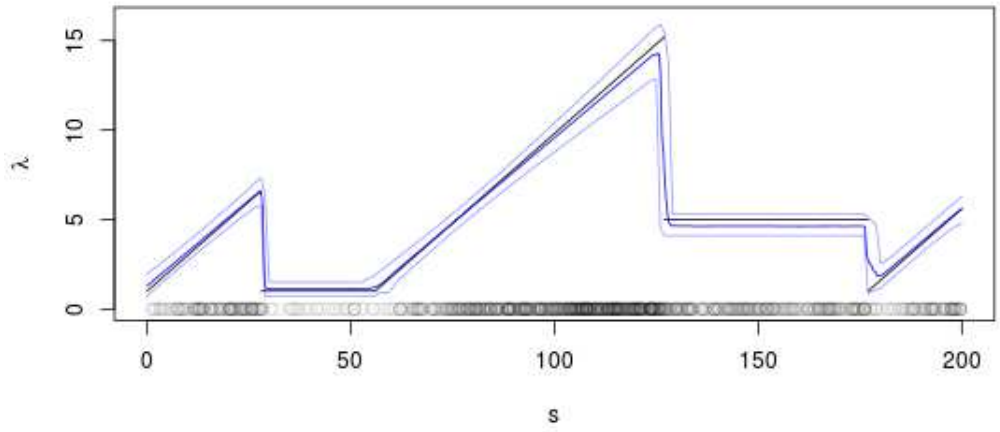


Figure 4.3: Scenario 6. Blue lines: posterior mean and 95% credibility interval for the IF. Black line: true IF. Circles: data.

Parameter	True value	Mean	SD	$CI_{0.025}$	$CI_{0.975}$
ψ_{11}	1	1.2948	0.3107	0.7281	1.9460
ψ_{12}	0.2	0.1911	0.0136	0.1649	0.2181
θ_1	0.035	0.0254	0.0147	0.0052	0.0613
θ_2	0.035	0.0366	0.0212	0.0074	0.0889
θ_{12}	0.5	0.7542	0.1918	0.2995	0.9918
θ_{13}	0.5	0.2458	0.1918	0.0082	0.7005
θ_{21}	0.5	0.7532	0.1924	0.2939	0.9917
θ_{24}	0.5	0.2468	0.1924	0.0083	0.7061

Table 4.4: Posterior statistics for scenario 6.

The MCMC presents high autocorrelation structure (see the ACF in Figure B.20). The inefficiency factor was equal 82.54 and for a 100k MCMC samples of the log posterior density the effective sample size was 1212. Although it presents high inefficiency factor, the algorithm takes around 1.28 minutes to draw 100 effective samples, which is quite good, specially given the high number of observations.

4.3 Real data analysis

4.3.1 Coal mining disasters

We apply the proposed methodology to the classic coal mining disasters data of Jarrett (1979), consisting of the dates of 191 explosions in coal mines that killed ten or more men in Britain between 15th March 1851 and 22th March 1962 (rescaled to $[0, 112]$, yearly).

We observe an empirical estimation of the IF to define the functional forms and other feature of the model to be fit (see Figure B.23 in Appendix B). The following is assumed: $K = \{1, 2\}$, $E = \{1, 2, 3\}$, $g_1 = R_T(s)$, $g_2 = \psi_{21} + \psi_{22}(s - \delta_T(s))$, $h(1) = h(3) = 1$, $h(2) = 2$ and $(R_j|h(Z_j) = 1) \sim \text{Uniform}(0.1, 0.3, \cdots, 2.9, 3.1)$. Thus, the model has a constant functional form with varying value and a decreasing line with a fixed (unknown) intercept. To avoid identifiability issues, we adopt suitable informative priors for θ and ψ's, more specifically: $\psi_{21} \sim \text{Normal}(3, 0.5^2)$, $\psi_{22} \sim \text{Normal}(-0.1, 0.02^2)$, $\theta_1 \sim \text{Gamma}(1, 10)$, $\theta_2 \sim \text{Gamma}(1, 5)$. The priors for the θ_i.'s are Uniforms on the simplex. The densities of these priors can be found in Figure B.21 in Appendix B. Although the priors for ψ's seem very informative, these choices have a reasonable variability for the intercept and the slope according to the scale of the problem. It would be reasonable to also consider some increasing functional form due to the behavior of the IF around times 60 and 85, nevertheless, under the chosen specification, we can investigate how efficient the varying constant g_1 is to accommodate this behavior.

The MCMC chain runs for 50k iterations. The inefficiency factor of the log posterior density is 20.08, leading to an effective sample size of 2490 (see the ACF in Figure B.25). The algorithm takes around 5.52 seconds to draw 100 effective samples. The Table 4.5 and Figure B.25 (in Appendix B) present the posterior statistics and trace plots for the parameters, respectively.

Parameter	Mean	SD	$\text{CI}_{0.025}$	$\text{CI}_{0.975}$
ψ_{21}	3.1205	0.6913	1.7872	4.4700
ψ_{22}	-0.1098	0.0864	-0.3208	-0.0053
θ_1	0.0270	0.0199	0.0025	0.0766
θ_2	0.1643	0.1493	0.0150	0.5639

Table 4.5: Posterior statistics for the analysis of the coal mine data.

We compare the results obtained with the MSCPV to those obtained with a model that assumes a non-parametric structure for the IF, proposed by Gonçalves & Gamerman

(2018). More specifically, the IF is assumed to be a continuous function of a Gaussian process trajectory. Whilst the autocorrelation of the log posterior density is considerably lower for the non-parametric model, its computational cost is significantly higher, resulting in a less computational efficient algorithm - 3.93 minutes to draw 100 effective samples. This class of model allows the most flexible estimation for the IF and, on the other hand, it has high computational costs. The Figure 4.4 shows the estimated IF for both models. The estimates are similar and, naturally, smoother for the non-parametric model for which, as expected, a better performance is observed to estimate the IF around times 70 and 100. If we consider the posterior mean of the integrated IF over the whole interval, this is approximately 192 for both models.

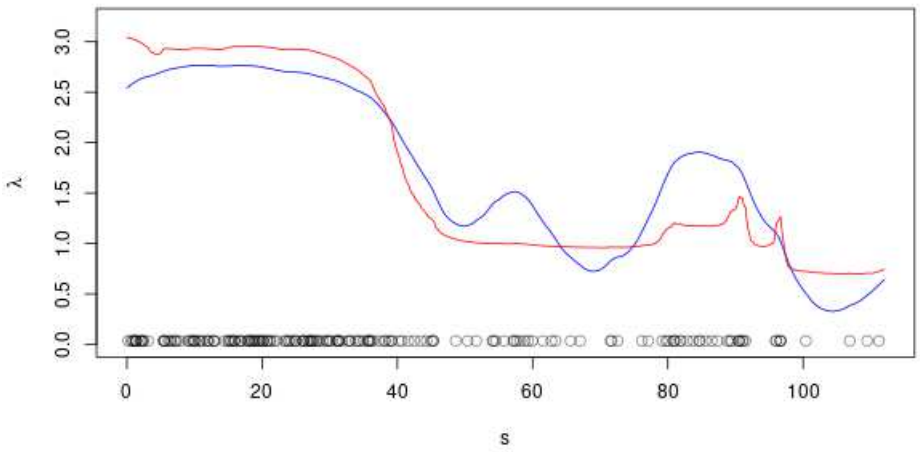


Figure 4.4: Estimates of the IF for coal mine data - posterior means. Blue line: non-parametric model. Red line: MSCPV. Circles: data.

4.3.2 Daily exchange rate variance

The second dataset that we apply the proposed methodology is the currency exchange rate of United States dollar to Brazilian real. The dataset consists of 670 dates that the daily exchange rate varied by more than 1%, between January 2009 and December 2018 (rescaled to [0, 120], monthly).

The following is assumed: $K = \{1, 2\}$, $E = \{1, 2\}$, $g_1 = R_T(s) + \psi_{11}(s - \delta_T(s))$, such that $\psi_{11} < 0$, $g_2 = \psi_{21} + \psi_{22}(s - \delta_T(s))$, such that $\psi_{22} > 0$, $h(1) = 1$, $h(2) = 2$ and $(R_j|h(Z_j) = 1) \sim \text{Uniform}(6, 6.2, \cdots, 10.8, 11)$. Thus, the model has a decreasing line with varying starting value and a increasing line with a fixed (unknown) starting

value. The empirical estimation of the IF can be seen in Figure B.26 (in Appendix B). We adopted the following priors for θ and ψ's: $\psi_{11} \sim \text{Normal}(-0.3, 0.1^2)$, $\psi_{21} \sim \text{Normal}(1, 0.5^2)$, $\psi_{22} \sim \text{Normal}(0.5, 0.1^2)$, $\theta_1 \sim \text{Gamma}(1, 15)$, $\theta_2 \sim \text{Gamma}(1, 20)$. The priors for the θ_i.'s are Uniforms. The densities of these priors can be found in Figure B.22 in Appendix B.

The MCMC chain runs for 40k iterations (after burn-in) and the inefficiency factor of the log posterior density is 66.57 (see the ACF in Figure B.27). The algorithm takes around 1.49 minutes to draw 100 effective samples. Although it is a computational time longer than that performed with previous data, it can also be considered quite fast, given the high number of observations. The Table 4.6 and Figure B.27 (in Appendix B) present the posterior statistics and trace plots for the parameters, respectively, where the trace plots suggest a good convergence. The estimation of the IF is presented in Figure 4.5.

Parameter	Mean	SD	$\text{CI}_{0.025}$	$\text{CI}_{0.975}$
ψ_{11}	-0.3241	0.0380	-0.3903	-0.2321
ψ_{21}	1.3887	0.5610	0.4348	2.5477
ψ_{22}	0.5097	0.0525	0.4062	0.6117
θ_1	0.0645	0.0288	0.0219	0.1330
θ_2	0.1339	0.0707	0.0357	0.3049

Table 4.6: Posterior statistics for the analysis of the exchange rate data.

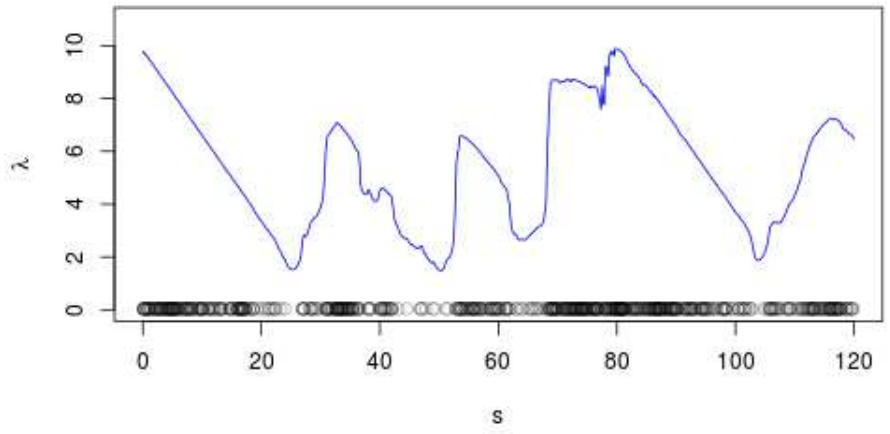


Figure 4.5: Estimate of the IF for exchange rate data. Blue line: posterior mean. Circles: data.

With the proposed methodology it is also possible to make predictions, through the posterior predictive distribution. To assess the model's prediction we use the exchange

rate data from January 2019 to November 2019 (11 months) for comparison. During this time period, 49 dates in which the daily exchange rate varied by more than 1% were observed. Note that these data are not used for the model estimation. The mean and standard deviation of the posterior predictive distribution for the number of events in this time period are 38.64 and 12.97, respectively. The prediction of the number of events for a future period can be considered good, since the true value is less than one standard deviation from the posterior predictive mean.

Chapter 5

Final remarks

Motivated by the importance of modeling point pattern data, we proposed a novel class of unidimensional Cox processes, referred to as Markov switching Cox process. This class stems from the simpler, yet quite interesting, Markov modulated Poisson process. Whilst the MMPP considers the intensity function of a non-homogeneous Poisson process to switch among constant levels according to a continuous-time Markov chain, the MSCP allows the IF to switch among pre-determined parametrised functional forms.

We developed a Bayesian methodology to perform exact inference in this class of point process by devising an MCMC algorithm to sample from the posterior distribution. The algorithm uses the data augmentation idea from Rao & Teh (2013), which is crucial for the efficiency of the algorithm. We developed upon this idea to propose an strategy to boost the efficiency of the algorithm. It was also proposed an extension to consider varying starting values and provided the required adaptations from the previous MCMC algorithm for this case.

Simulated studies were performed to investigate the efficiency of the proposed methodology, under different specifications. Results strongly suggested a very good performance of the MCMC algorithm in terms of estimation and computational cost. In particular, the computational cost is feasible to deal with very large data sets. For real data analysis, the proposed methodology also leads to a very fast MCMC algorithm and it obtained good results for prediction.

Although the form of the IF in the proposed models is not entirely smooth, due to the switches, smooth estimates may be obtained through the posterior mean, since this averages over the possible switches. This could be clearly noticed in some of the examples

provided and in the comparison with a model that considers a continuous IF to fit a real data set.

Appendix A

Algorithm from Section 2.3.1

Algorithm 1 samples from the density in (2.7).

Algorithm 1 Full conditional of (U, W)

Input: S, (Z_0, Z, T), Q_θ, Ω.
Output: U.

1: **for** $j = 0 \to J$ **do**
2: Sample $|U_j| \sim \text{Poisson}(\Omega + Q_{Z_j})$.
3: Sample U_j from $|U_j|$ independent Uniform(T_j, T_{j+1}) r.v.'s.
4: **end for**
return $U = (U_0 \cup \cdots \cup U_J)$ (in increasing order).

Algorithm from Section 2.3.2

Algorithm 2 samples from the density in (2.10) directly. If a proposal (V_0, V) that leads to a negative IF is sampled, this is immediately rejected.

Algorithm 2 Full conditional of (V_0, V)

Input: y, (Z_0, Z, T), E, π_0, g_k's, ψ_k's, W, B.
Output: (V_0, V).

1: Define a $(|E|^{1+|W|}) \times (1 + |W|)$ matrix M with all possible (V_0, V) arrangements (each row represents a possible path).
2: **for** $i = 1 \to |E|^{1+|W|}$ **do**
3: Compute $\lambda(s)$, for all $s \in [0, S]$, for $(V_0, V) = M_{i\bullet}$.
4: **if** $\lambda(s) < 0$ for some $s \in [0, S]$ **then**, make $p_i = 0$.
5: **else** compute p_i from (2.10), for $(V_0, V) = M_{i\bullet}$.
6: **end for**
7: Sample $V' \sim \text{Multinomial}\left(1, \frac{(p_1, \cdots, p_{|E|^{1+|W|}})}{\sum_{i=1}^{|E|^{1+|W|}} p_i}\right)$.
return $(V_0, V) = M_{V'\bullet}$.

Algorithm 3 performs the MH step to sample from (2.10), with proposal (2.11) and acceptance probability (2.12). The proposal is sampled forward in time, following the sequence $V_j^*|V_{j-1}^*, j = 1, \cdots, |W|$, and constants c_j^* are computed along the way. Occasionally, given the parameter vector ψ and $V_{0:j-1}$, at $V_j = i$, for some $i \in E$, the likelihood function may not be defined. For a proposal that defines a negative IF, we define $\pi^*(V_j = i|V_{j-1}^*) = 0$.

Algorithm 3 MH step for (V_0, V)

Input: (V_0, V), y, (Z_0, Z, T), E, π_0, g_k, ψ_k's, W, B.
Output: (V_0^*, V^*).

Compute $c = (c_0, \cdots, c_{|W|})$ from (2.11), for the current trajectory (V_0, V).
Define $p = c_0^*(\pi^*(V_0 = 1), \cdots, \pi^*(V_0 = |E|))$.
Sample $V_0^* \sim$ Multinomial$(1, p)$.
for $j = 1 \to |W|$ **do**
    if $g_{h(i)}(s, \delta(s, T), \psi) < 0$, for some $s \in [W_j, W_{j+1})$, given ψ and the sequence $V_{0:j-1}$ **then** Set $\pi^*(V_j = i|V_{j-1}^*) = 0$.
    Define $p = c_j^*(\pi^*(V_j = 1|V_{j-1}^*), \cdots, \pi^*(V_j = |E| \ |V_{j-1}^*))$.
    Sample $V_j^* \sim$ Multinomial$(1, p)$.
end for
Sample $u \sim$ Uniform$(0, 1)$.
if $u < \left(1 \wedge \prod_{j=1}^{|W|} \frac{c_j}{c_j^*}\right)$ **then return** $(V_0^*, V_1^*, \cdots, V_{|W|}^*)$.
else return (V_0, V).

In order to avoid numerical problems when computing the acceptance probability in Algorithm 3, we compute the logarithm of this probability as follows.

$$\ln c_j = \ln \left[\sum_{i=1}^{|E|} \exp \left\{ \ln c_{ji} - \ln c_{jm} \right\} \right] + \ln c_{jm}.$$

where $c_{jm} = \max\{c_{j1}, \cdots, c_{j|E|}\}$ and $c_{ji} = \pi^*(V_j = i|V_{j-1})$.

Algorithm from Section 2.3.3

Algorithm 4 samples from the density in (2.13) in the general case in which jumps between the same functional form is allowed.

Algorithm 4 Full conditional of θ

Input: S, (Z_0, Z, T), $\alpha = \{\alpha_i : i \in K\}$, $\beta = \{\beta_i : i \in K\}$, $\gamma = \{\gamma_{ij} : i \in K, j \in K\}$.
Output: θ.

for $i = 1 \to |K|$ **do**
  Define $\Delta_i = \sum_{j=1}^{J} (T_{j+1} - T_j) (\mathbb{I}\{Z_j = i\} + \mathbb{I}\{Z_j = i + |K|\})$.
  Define $J_i = \sum_{j=0}^{J} (\mathbb{I}\{Z_j = i\} + \mathbb{I}\{Z_j = i + |K|\})$.
  Define $J_{i\bullet} = \Big(\sum_{j=1}^{J} (\mathbb{I}\{Z_j = 1 | Z_{j-1} = i\} + \mathbb{I}\{Z_j = 1 | Z_{j-1} = i + |K|\}), \cdots,$
    $\sum_{j=1}^{J} (\mathbb{I}\{Z_j = |K| | Z_{j-1} = i\} + \mathbb{I}\{Z_j = |K| | Z_{j-1} = i + |K|\}),$
    $\sum_{j=1}^{J} (\mathbb{I}\{Z_j = i + |K| | Z_{j-1} = i\} + \mathbb{I}\{Z_j = i | Z_{j-1} = i + |K|\}) \Big)$.
  if $V_J = i$ or $V_J = i + |K|$ **then** Sample $\theta_i \sim \text{Gamma}(\alpha_i + J_i - 1, \beta_i + \Delta_i)$.
  else Sample $\theta_i \sim \text{Gamma}(\alpha_i + J_i, \beta_i + \Delta_i)$.
  Sample $\theta^*_{i\bullet} \sim \text{Dirichlet}(J_{i\bullet} + \gamma_{i\bullet})$.
  if $i + |K| \in E$ **then**
    Set $\theta_{(i+|K|)i} = \theta_{i(i+|K|)}$.
    for $j = 1 \to |K|$ **do**
      Set $\theta_{(i+|K|)j} = \theta_{ij}$.
      Set $\theta_{i(j+|K|)} = \theta_{(i+|K|)(j+|K|)} = 0$.
end for
return $\theta = (\theta_{\bullet}, \theta_{\bullet\bullet})$.

Algorithm from Section 2.3.4

Each ψ_k is sampled separately. If g_k is a constant function, the full conditional distribution of ψ_k is a Gamma distribution. For the other cases, ψ_k is sampled in a Gaussian random walk MH step as described in Algorithm 5. If a proposal ψ_k that leads to a negative IF is sampled, this is immediately rejected.

Algorithm 5 MH step for ψ_k

Input: ψ_k, proposal covariance matrix Σ_k, y, S, (Z_0, Z, T).
Output: ψ_k.

Sample $\psi^*_k \sim \text{Normal}(\psi_k, \Sigma_k)$.
if $g_k(s, \delta(s, T), \psi^*_k) < 0$, for some $s \in S_k$ **then** Set $L(Z_0, Z, T, \psi^*_k) = 0$.
Sample $u \sim \text{Uniform}(0, 1)$.
if $u < \left(1 \wedge \frac{L(Z_0,Z,T,\psi^*_k)\pi(\psi^*_k)}{L(Z_0,Z,T,\psi_k)\pi(\psi_k)}\right)$ **then return** ψ^*_k.
else return ψ_k.

The choice of the covariance matrix is based on an adaptive random walk Metropolis algorithm proposed in Roberts & Rosenthal (2009). We calibrate the MH proposal to obtain nearly reasonable acceptance rates - from around 0.44 in one dimension to around 0.234 in high (>4) dimensions.

Basically, the covariance matrix of the proposal is calibrated to be proportional to the empirical covariance matrix of the last N iterations of the chain, for some suitable N. This is done until a reasonable acceptance rate is achieved.

Algorithm from Section 2.4.2

The partition of the observed time interval to sample (V_0, V) is based on the MH global acceptance rate. First, we draw N MCMC iterations sampling (V_0, V) over the whole time interval and compute the empirical acceptance rate. If this is less than 0.25, we make $B = B + 1$ and run another N MCMC iterations. This strategy is repeated until an acceptance rate greater than 0.25 is achieved. Depending on the example a threshold other than 0.25 may be more suitable.

In order to define a partition with B blocks, we take the T_j's which are the closest to the W_i's with respective values of i that are the closest to $|W|/B,\ 2|W|/B, \ldots,\ (B-1)|W|/B$.

Algorithm details of Section 3.2

Algorithm 6 MH step for (V_0, V, R)

Input: (V_0, V, R), y, E, π_0, g_k, ψ_k's, τ_k's, W, B.
Output: (V_0, V, R).

Compute the vector $c = (c_0, \cdots, c_{|W|})$ from (3.8) for the current values of (V_0, V, R).
for $k = 1 \to |K|$ **do**
    Compute $c_0^*(V_0 = k)$.
end for
Define $p = c_0^* \left(\frac{\pi_0(V_0=1)}{c_0(V_0=1)}, \cdots, \frac{\pi_0(V_0=|K|)}{c_0(V_0=|K|)} \right)$.
Sample $V_0^* \sim$ Multinomial$(1, p)$.
Sample R_0^* from the density or probability vector $c_0(V_0)L_0(V_0, R_0)\pi(R_0|V_0, \tau)$.
for $j = 1 \to |W|$ **do**
    for $k = 1 \to |K|$ **do**
        Compute $c_j^*(V_j = k)$.
    end for
    Define $p = c_j^* \left(\frac{\pi(V_j=1|V_{j-1},\theta)}{c_j(V_j=1)}, \cdots, \frac{\pi(V_j=|K||V_{j-1},\theta)}{c_j(V_j=|K|)} \right)$.
    Sample $V_j^* \sim$ Multinomial$(1, p)$.
    Sample R_j^* from the density or probability vector $c_j(V_j)L_j(V_{0:j}, R)(\pi(R_j|V_j, \tau)\mathbb{I}(V_{j-1} \neq V_j) + \mathbb{I}(V_{j-1} = V_j))$.
end for
Sample $u \sim$ Uniform$(0, 1)$.
if $u < \left(1 \wedge \prod_{j=1}^{|W|} \frac{c_j}{c_j^*}\right)$ **then return** $(V_0^*, V_1^*, \cdots, V_{|W|}^*, R_1^*, \cdots, R_J^*)$.
else return (V_0, V, R).

Appendix B

This appendix presents some results from the simulation studies and real data discussed in Chapter 4.

The Table B.1 presents the posterior statistics for all the parameters of each scenario. Tables B.2 − B.6 show some results of the algorithm specifications - the number of iterations to calibrate the random walk proposals; the number of blocks obtained from splitting the observed time interval; acceptance rates for (V_0, V) and ψ_k's after calibration; total number of iterations; burn-in ; total computational time. Note that if there is more than one interval, we report the acceptance rate of V^b for each interval.

Scenario	Parameter	True value	Mean	SD	$CI_{0.025}$	$CI_{0.975}$
1 A	ψ_{11}	1	0.7489	0.3479	0.1941	1.5563
	ψ_{12}	0.2	0.2016	0.0254	0.1513	0.2520
	ψ_{21}	1	1.1330	0.3855	0.3474	1.9464
	θ_1	0.025	0.0582	0.0339	0.0103	0.1404
	θ_2	0.05	0.3267	0.2963	0.0296	1.0834
1 B	ψ_{11}	1	0.6581	0.2443	0.2548	1.1985
	ψ_{12}	0.2	0.1947	0.0109	0.1724	0.2160
	ψ_{21}	1	1.0976	0.2791	0.5790	1.6855
	θ_1	0.025	0.0412	0.0205	0.0105	0.0898
	θ_2	0.05	0.2831	0.2215	0.0418	0.8567
1 C	ψ_{11}	1	0.6970	0.3434	0.2036	1.5264
	ψ_{12}	0.2	0.2095	0.0064	0.1971	0.2224
	ψ_{21}	1	1.0300	0.0806	0.8765	1.1931

Table B.1: Posterior statistics.

Scenario	Parameter	True value	Mean	SD	$CI_{0.025}$	$CI_{0.975}$
	θ_1	0.025	0.0234	0.0110	0.0073	0.0500
	θ_2	0.05	0.0360	0.0256	0.0082	0.1047
	ψ_{11}	1	0.7328	0.2930	0.2465	1.3551
	ψ_{12}	0.2	0.2034	0.0052	0.1934	0.2135
1 D	ψ_{21}	1	1.0417	0.0531	0.9392	1.1463
	θ_1	0.025	0.0214	0.0075	0.0093	0.0384
	θ_2	0.05	0.0269	0.0110	0.0108	0.0534
	ψ_{11}	1	1.3918	0.6141	0.4191	2.7641
	ψ_{12}	0.2	0.2176	0.0312	0.1682	0.2938
	ψ_{21}	1	1.1485	0.1335	0.8995	1.4167
2 A	θ_1	0.035	0.0621	0.0308	0.0174	0.1352
	θ_2	0.035	0.0480	0.0341	0.0095	0.1363
	θ_{12}	0.5	0.6362	0.2366	0.1550	0.9837
	θ_{13}	0.5	0.3638	0.2366	0.0163	0.8450
	ψ_{11}	1	1.7559	0.4479	1.0238	2.7733
	ψ_{12}	0.2	0.1981	0.0106	0.1775	0.2185
	ψ_{21}	1	1.0422	0.0765	0.9054	1.1889
2 B	θ_1	0.035	0.0353	0.0132	0.0144	0.0651
	θ_2	0.035	0.0256	0.0121	0.0078	0.0543
	θ_{12}	0.5	0.6549	0.1976	0.2464	0.9723
	θ_{13}	0.5	0.3451	0.1976	0.0277	0.7535
	ψ_{11}	1	0.7171	0.4950	0.0727	1.9143
	ψ_{12}	0.2	0.2345	0.0205	0.1926	0.2729
	ψ_{21}	15	14.1803	0.8781	12.4642	15.8678
	ψ_{22}	-0.2	-0.1878	0.0233	-0.2313	-0.1415
	ψ_{31}	1	1.0082	0.1592	0.6984	1.3255
	θ_1	0.025	0.0449	0.0278	0.0067	0.1117
3 A	θ_2	0.025	0.0356	0.0247	0.0044	0.0974
	θ_3	0.05	0.0690	0.0502	0.0117	0.1985

Table B.1: Posterior statistics.

Scenario	Parameter	True value	Mean	SD	$CI_{0.025}$	$CI_{0.975}$
	θ_{12}	0.5	0.3080	0.2350	0.0112	0.8649
	θ_{13}	0.5	0.6920	0.2350	0.1351	0.9888
	θ_{21}	0.5	0.4034	0.2750	0.0155	0.9490
	θ_{23}	0.5	0.5966	0.2750	0.0510	0.9845
	θ_{31}	0.5	0.6582	0.2256	0.1655	0.9825
	θ_{32}	0.5	0.3418	0.2256	0.0175	0.8345
	ψ_{11}	1	1.1197	0.3279	0.5508	1.8495
	ψ_{12}	0.2	0.2102	0.0104	0.1893	0.2304
	ψ_{21}	15	15.2679	0.5536	14.1510	16.3505
	ψ_{22}	-0.2	-0.2100	0.0141	-0.2374	-0.1810
	ψ_{31}	1	0.9998	0.1098	0.7854	1.2151
	θ_1	0.025	0.0296	0.0131	0.0095	0.0603
3 B	θ_2	0.025	0.0191	0.0109	0.0040	0.0453
	θ_3	0.05	0.0438	0.0260	0.0110	0.1071
	θ_{12}	0.5	0.4711	0.1943	0.1251	0.8551
	θ_{13}	0.5	0.5289	0.1943	0.1449	0.8749
	θ_{21}	0.5	0.4946	0.2521	0.0500	0.9503
	θ_{23}	0.5	0.5054	0.2521	0.0497	0.9500
	θ_{31}	0.5	0.6989	0.1985	0.2552	0.9832
	θ_{32}	0.5	0.3011	0.1985	0.0168	0.7448
	ψ_{11}	0	0.3122	0.1893	0.0249	0.7396
	ψ_{12}	0.2	0.1990	0.0149	0.1689	0.2282
	ψ_{21}	15	13.5801	0.7167	12.1284	14.9576
	ψ_{22}	-0.2	-0.1855	0.0315	-0.2499	-0.1282
4 A	θ_1	0.035	0.0306	0.0179	0.0063	0.0748
	θ_2	0.035	0.0463	0.0272	0.0099	0.1127
	θ_{12}	0.5	0.4793	0.2249	0.0829	0.9008
	θ_{13}	0.5	0.5207	0.2249	0.0992	0.9171
	θ_{21}	0.5	0.4229	0.2248	0.0646	0.8862

Table B.1: Posterior statistics.

Scenario	Parameter	True value	Mean	SD	$CI_{0.025}$	$CI_{0.975}$
	θ_{24}	0.5	0.5771	0.2248	0.1138	0.9354
	ψ_{11}	0	0.2996	0.1765	0.0331	0.6902
	ψ_{12}	0.2	0.2042	0.0089	0.1851	0.2203
	ψ_{21}	15	14.2334	0.4240	13.3766	15.0505
	ψ_{22}	-0.2	-0.1799	0.0117	-0.2022	-0.1563
4 B	θ_1	0.035	0.0261	0.0103	0.0098	0.0498
	θ_2	0.035	0.0235	0.0103	0.0079	0.0475
	θ_{12}	0.5	0.6426	0.1823	0.2655	0.9456
	θ_{13}	0.5	0.3574	0.1823	0.0544	0.7345
	θ_{21}	0.5	0.5915	0.2071	0.1838	0.9488
	θ_{24}	0.5	0.4085	0.2071	0.0512	0.8162
	ψ_{11}	1	0.5760	0.2628	0.1244	1.1158
	ψ_{12}	0.1	0.1089	0.0073	0.0944	0.1229
	ψ_{21}	0	0.0591	0.0823	-0.0641	0.2578
	ψ_{22}	12	8.6111	2.7194	4.2712	14.4711
	ψ_{23}	-0.1	-0.1076	0.0190	-0.1496	-0.0737
5 A	θ_1	0.02	0.0148	0.0106	0.0018	0.0412
	θ_2	0.02	0.0337	0.0240	0.0041	0.0945
	θ_{12}	0.5	0.6557	0.2385	0.1500	0.9863
	θ_{13}	0.5	0.3443	0.2385	0.0137	0.8500
	θ_{21}	0.5	0.6567	0.2383	0.1531	0.9866
	θ_{24}	0.5	0.3433	0.2383	0.0134	0.8469
	ψ_{11}	1	0.8156	0.1741	0.4893	1.1668
	ψ_{12}	0.1	0.0995	0.0041	0.0914	0.1074
	ψ_{21}	0	0.0362	0.0504	-0.0313	0.1601
	ψ_{22}	12	10.3962	1.3980	7.7258	13.1853
	ψ_{23}	-0.1	-0.1050	0.0095	-0.1254	-0.0883
5 B	θ_1	0.02	0.0123	0.0061	0.0034	0.0267
	θ_2	0.02	0.0296	0.0148	0.0080	0.0652

Table B.1: Posterior statistics.

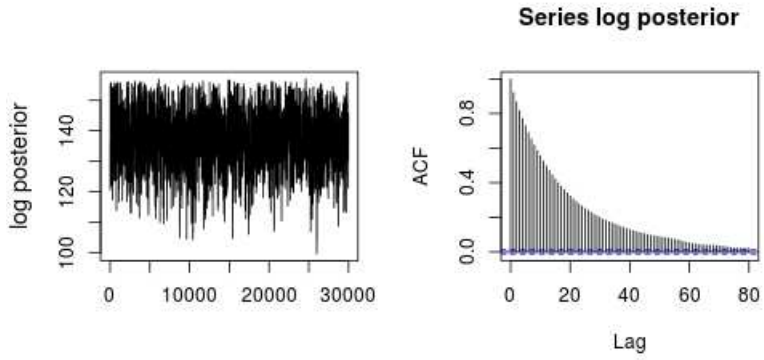
Scenario	Parameter	True value	Mean	SD	$CI_{0.025}$	$CI_{0.975}$
	θ_{12}	0.5	0.6188	0.2050	0.2001	0.9567
	θ_{13}	0.5	0.3812	0.2050	0.0433	0.7999
	θ_{21}	0.5	0.6298	0.2129	0.1986	0.9717
	θ_{24}	0.5	0.3702	0.2129	0.0283	0.8014

Table B.1: Posterior statistics.

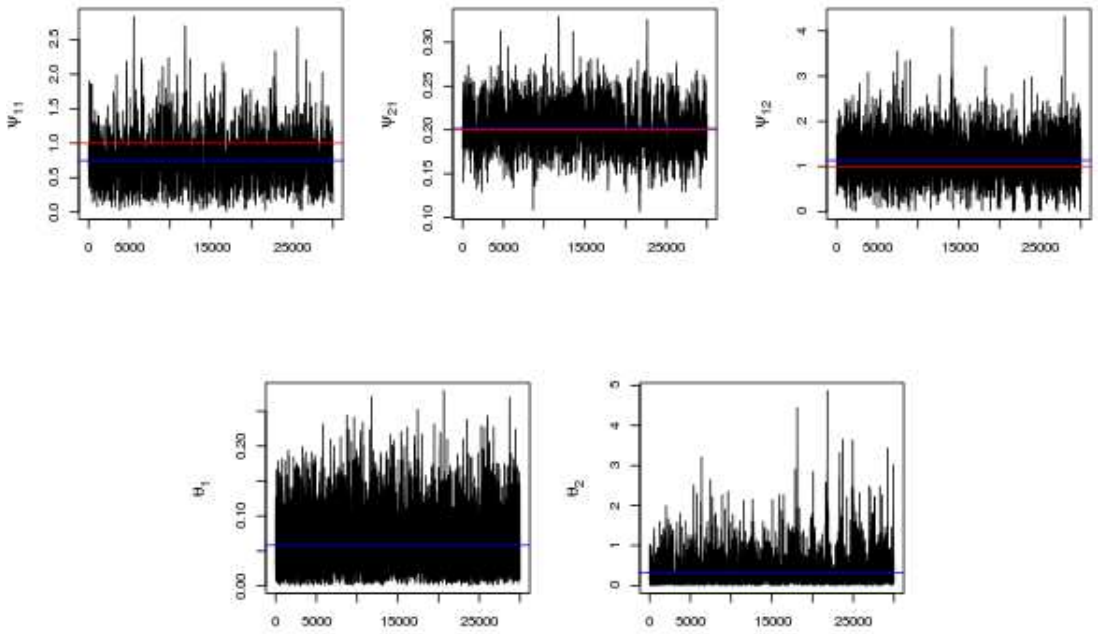
B.1 Scenario 1

	A	B	C	D
iterations	5k	11k	5k	7k
interval blocks	1	2	2	2
accept. rate (V_0, V)	0.3523	1^{st}: 0.5401 2^{nd}: 0.4714	1^{st}: 0.58750 2^{nd}: 0.73839	1^{st}: 0.4831 2^{nd}: 0.4850
accept. rate ψ	0.2597	0.2975	0.3440	0.3393
iterations MCMC	30k	25k	70k	60k
burn-in	0	5k	0	10k
Total computational time	24.11s	39.79s	2m24.57s	3m38.71s

Table B.2: Computational results for scenario 1.

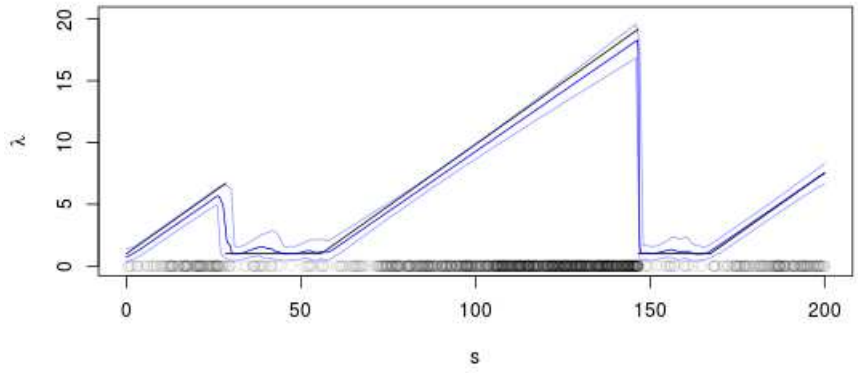


(a) Trace plot and ACF for the log posterior density.

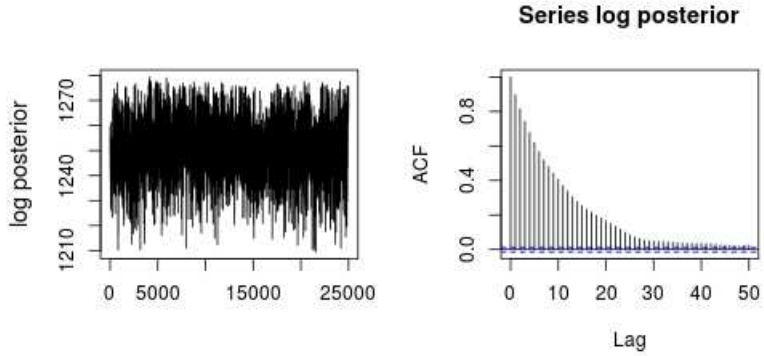


(b) Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.

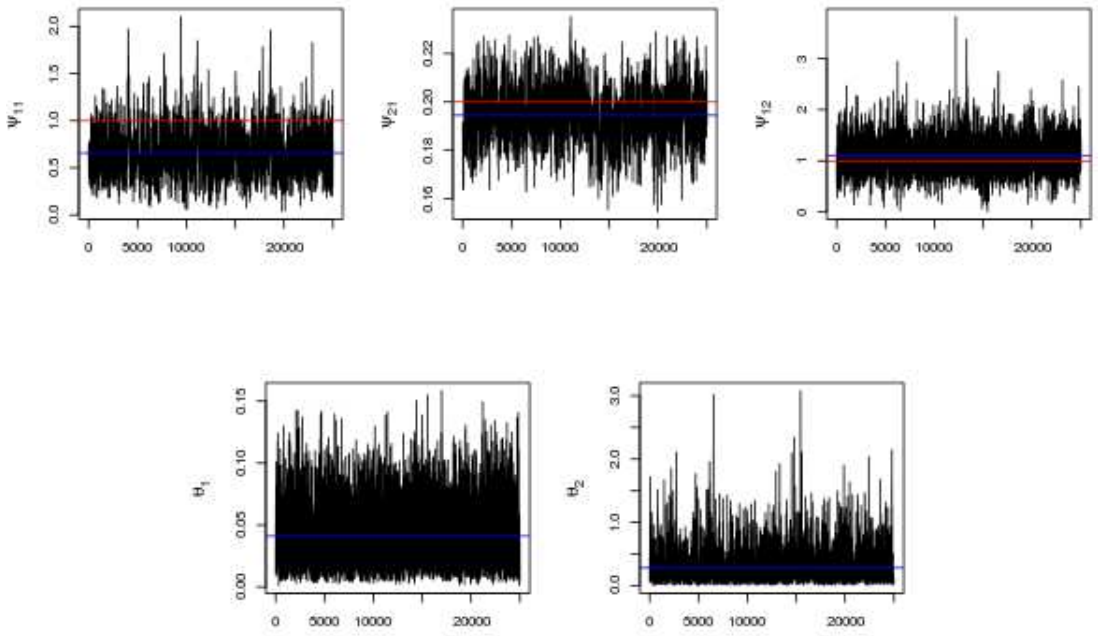
Figure B.1: Results for scenario 1A.



(a) Posterior mean and 95% credibility interval (blue lines) for the IF. The black line is the true IF.

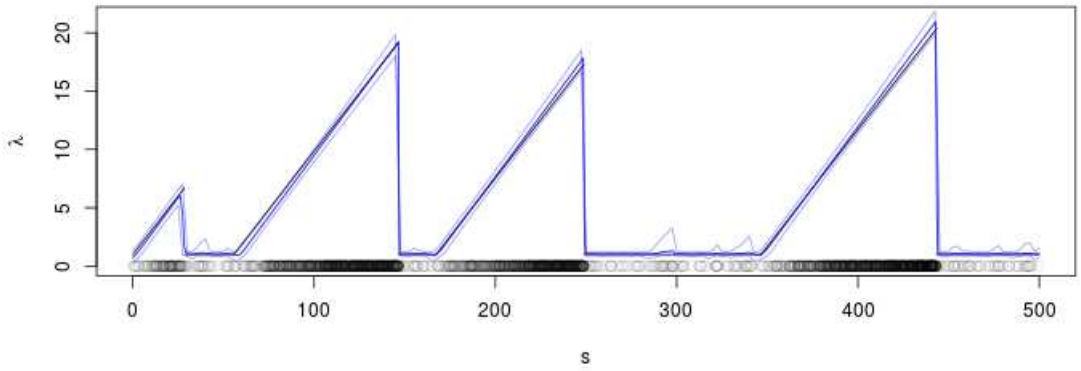


(b) Trace plot and ACF for the log posterior density.

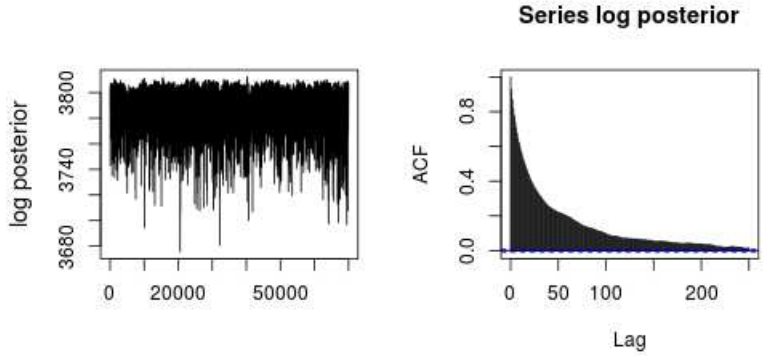


(c) Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.

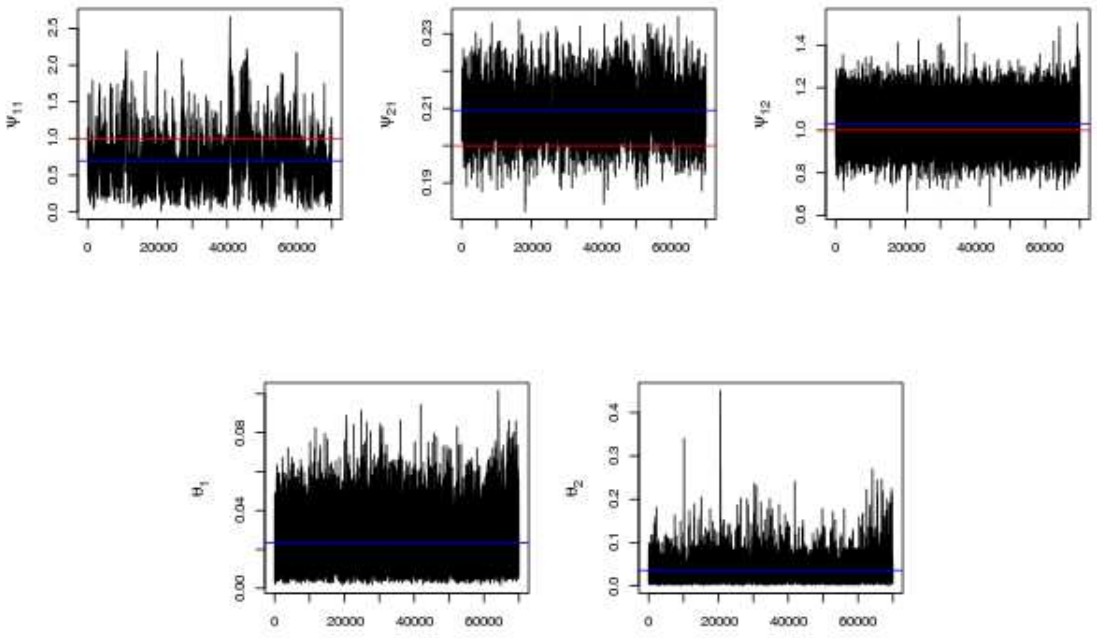
Figure B.2: Results for scenario 1B.



(a) Posterior mean and 95% credibility interval (blue lines) for the IF. The black line is the true IF.

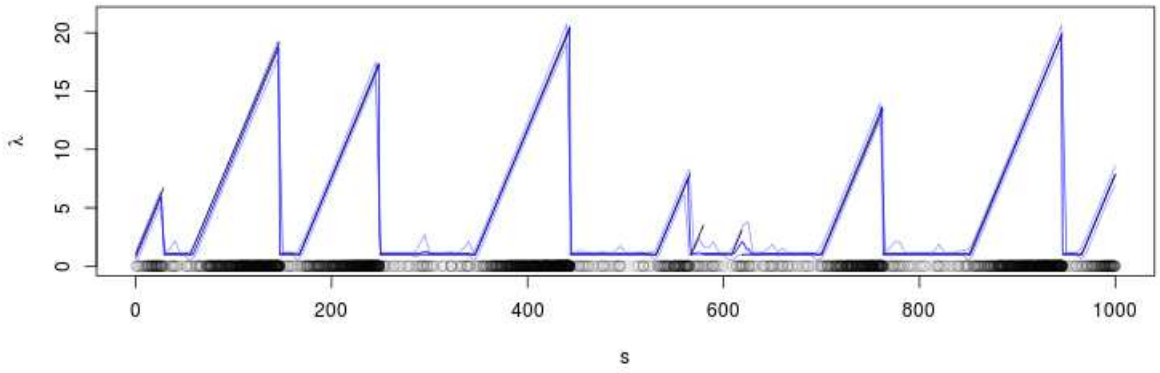


(b) Trace plot and ACF for the log posterior density.

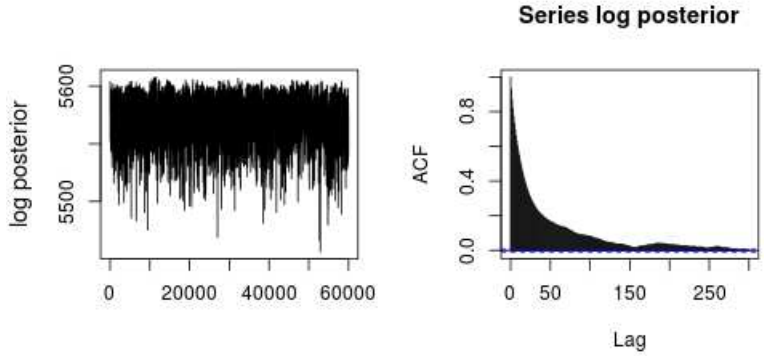


(c) Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.

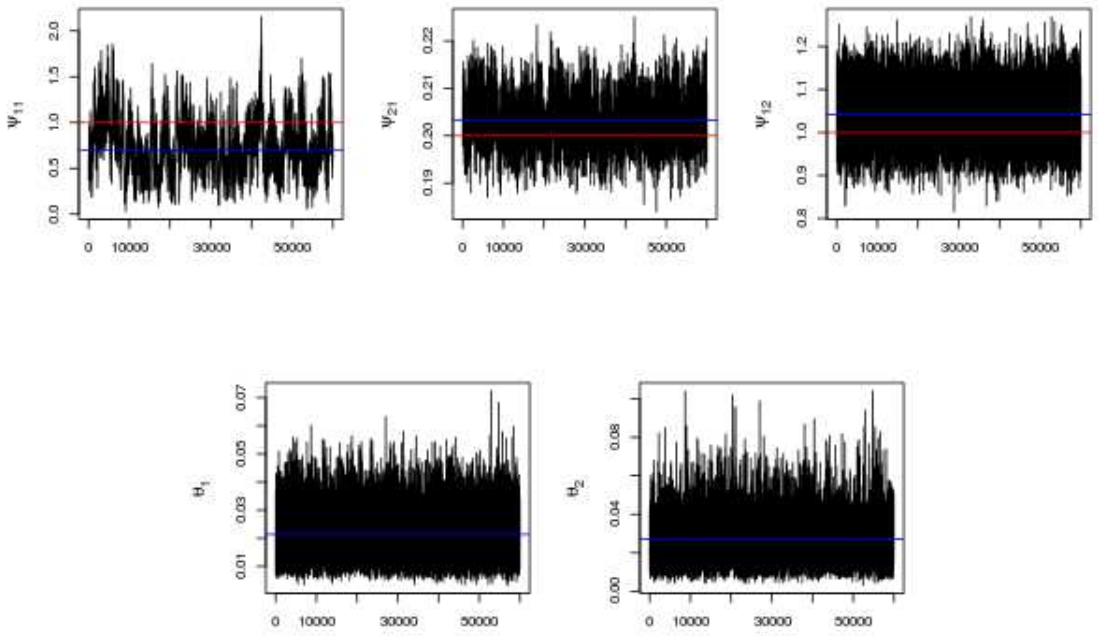
Figure B.3: Results for scenario 1C.



(a) Posterior mean and 95% credibility interval (blue lines) for the IF. The black line is the true IF.



(b) Trace plot and ACF for the log posterior density.



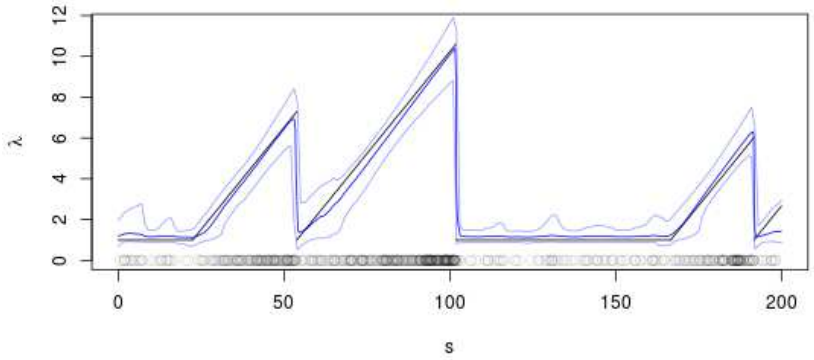
(c) Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.

Figure B.4: Results for scenario 1D.

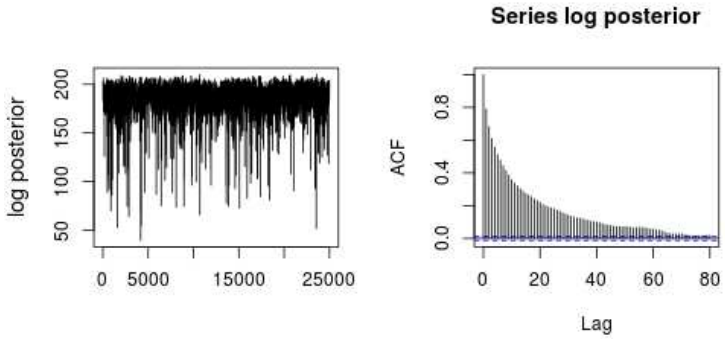
B.2 Scenario 2

	A	B
iterations	7k	5k
interval blocks	2	2
accept. rate (V_0, V)	1^{st}: 0.6082; 2^{nd}: 0.6499	1^{st}: 0.6125; 2^{nd}: 0.6554
accept. rate ψ	0.3198	0.3571
iterations MCMC	25k	25k
burn-in	0	10k
Total computational time	33.14s	56.94s

Table B.3: Computational results for scenario 2.



(a) Posterior mean and 95% credibility interval (blue lines) for the IF. The black line is the true IF.



(b) Trace plot and ACF for the log posterior density.

Figure B.5: Results for scenario 2A.

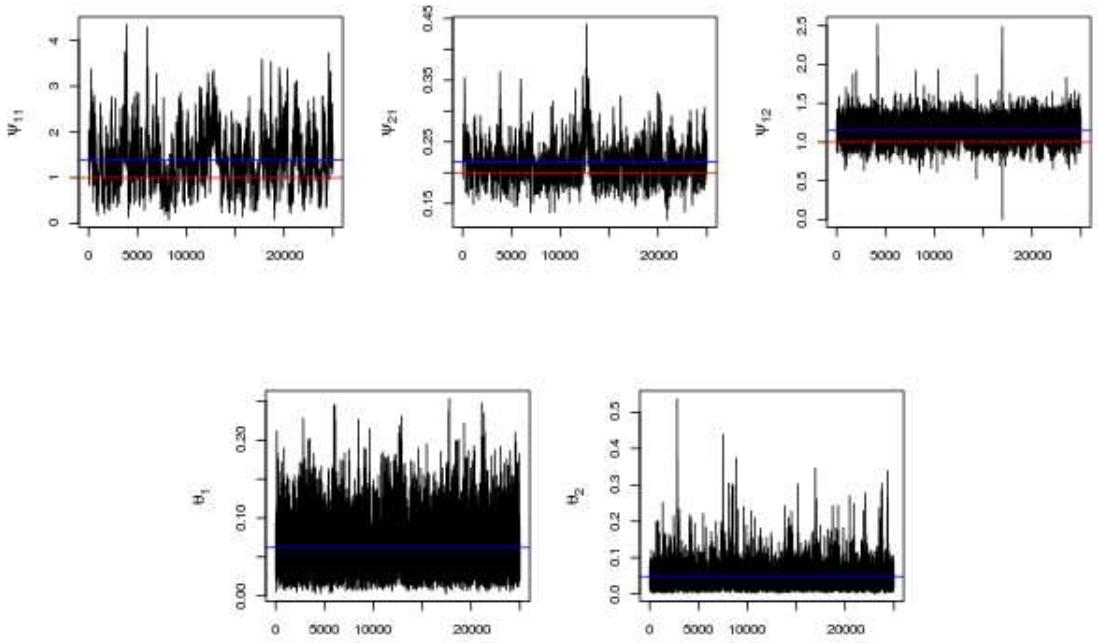


Figure B.6: Results for scenario 2A: Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.

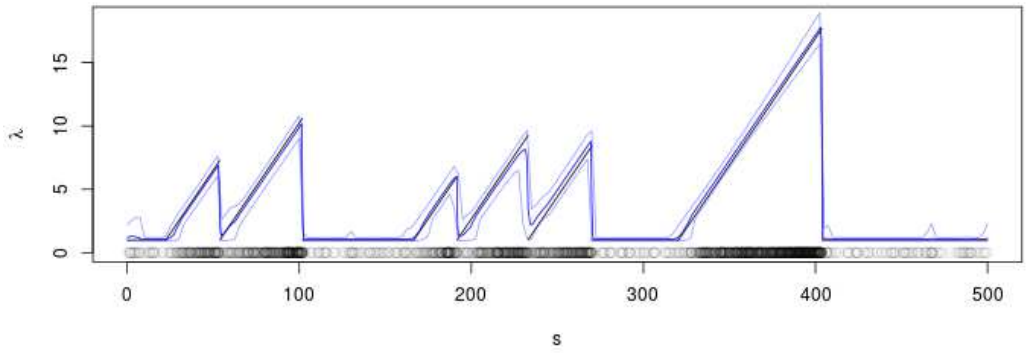
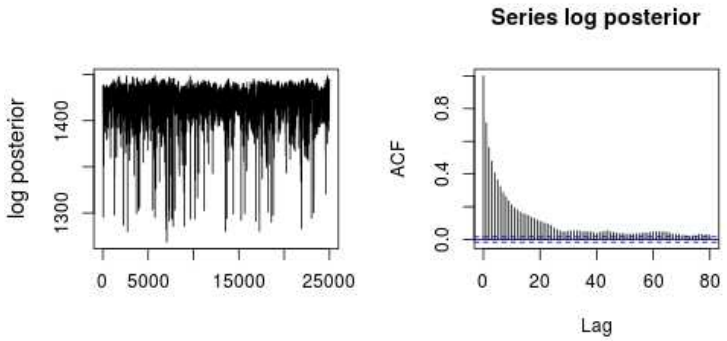
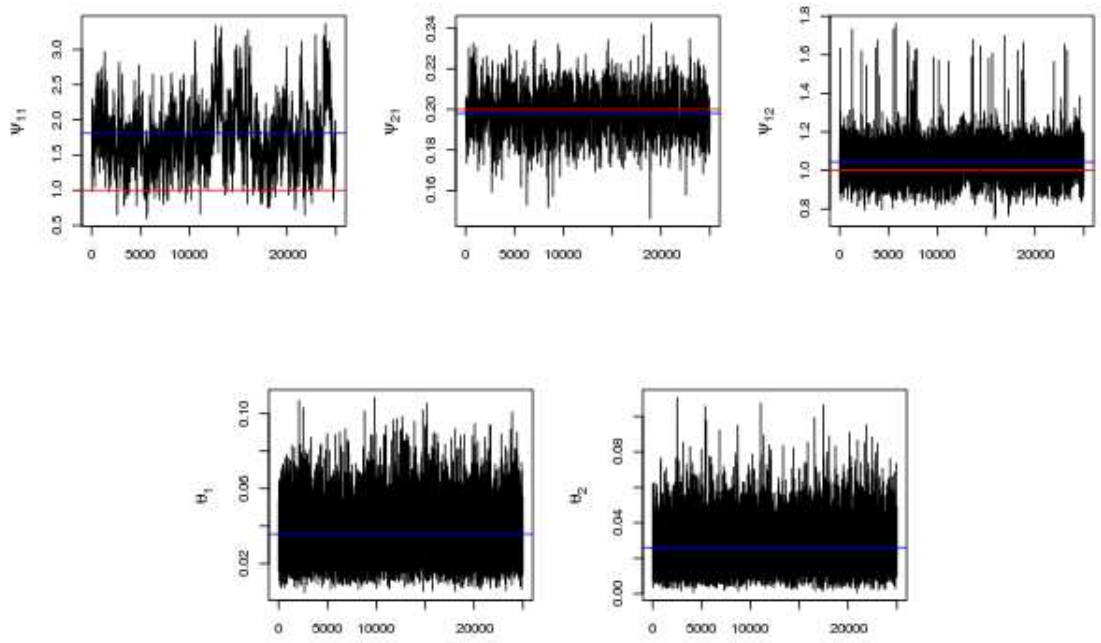


Figure B.7: Results for scenario 2B: Posterior mean and 95% credibility interval (blue lines) for the IF. The black line is the true IF.



(a) Trace plot and ACF for the log posterior density.



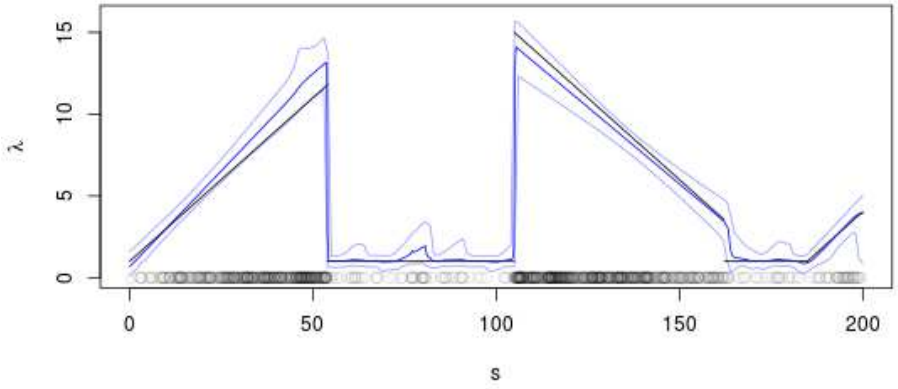
(b) Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.

Figure B.8: Results for scenario 2B.

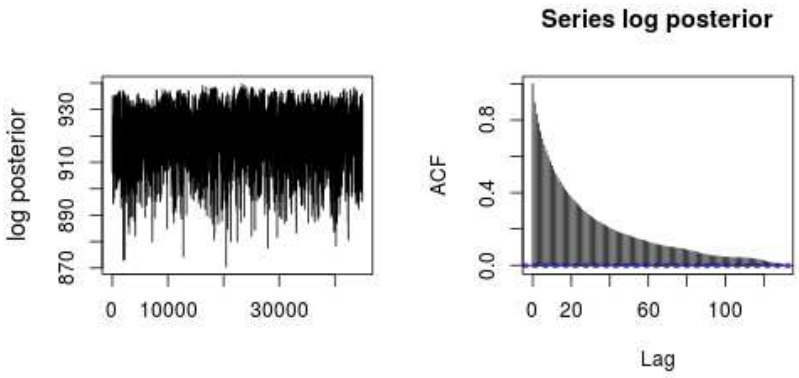
B.3 Scenario 3

	A	B
iterations	9k	9k
interval blocks	1	1
accept. rate (V_0, V)	0.4595	0.3369
accept. rate ψ	ψ_1: 0.3129; ψ_2: 0.3520	ψ_1: 0.3184; ψ_2: 0.3782
iterations MCMC	45k	55k
burn-in	0	15k
Total computational time	1m09.40s	2m58.27s

Table B.4: Computational results for scenario 3.



(a) Posterior mean and 95% credibility interval (blue lines) for the IF. The black line is the true IF.



(b) Trace plot and ACF for the log posterior density.

Figure B.9: Results for scenario 3A.

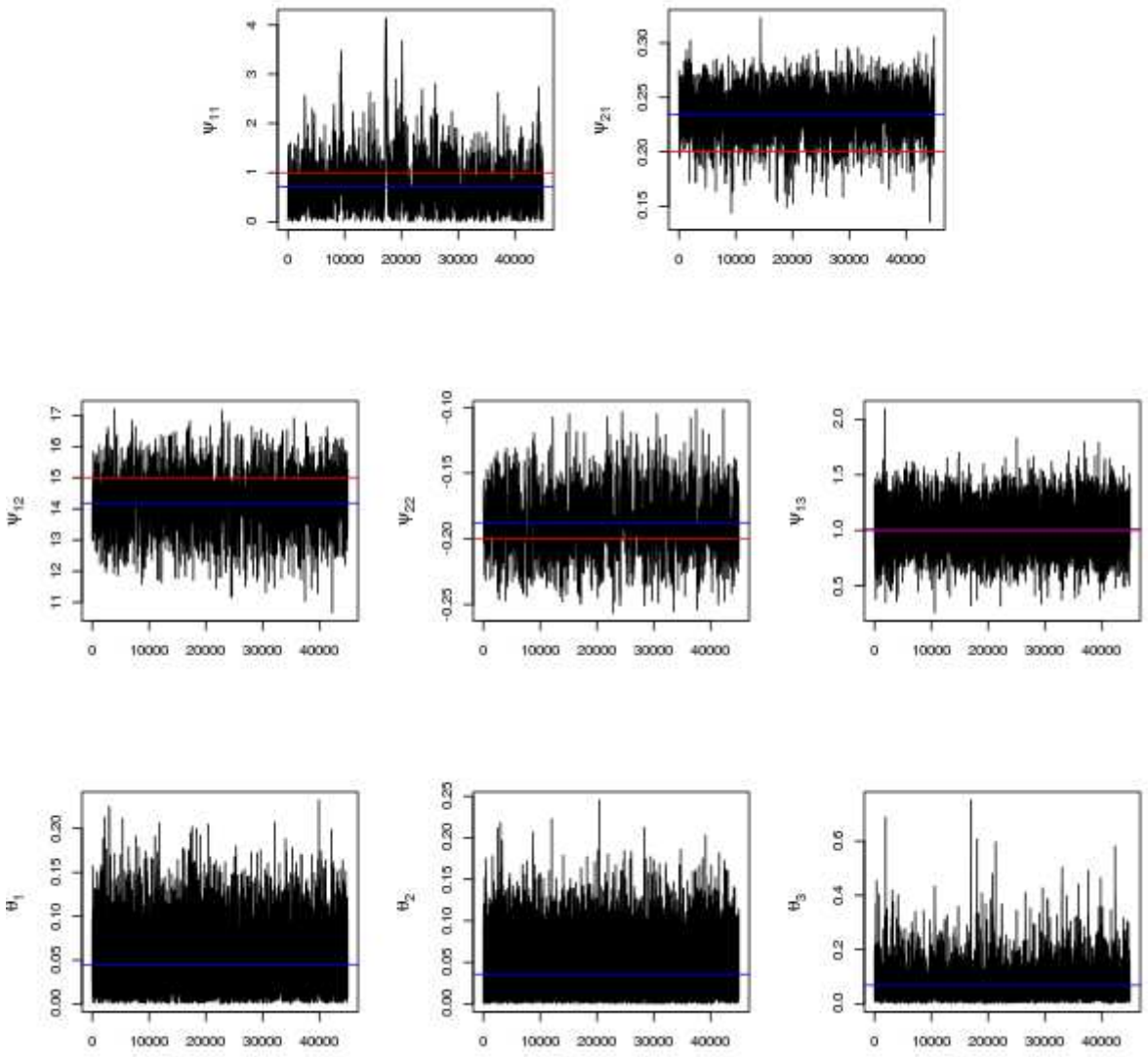
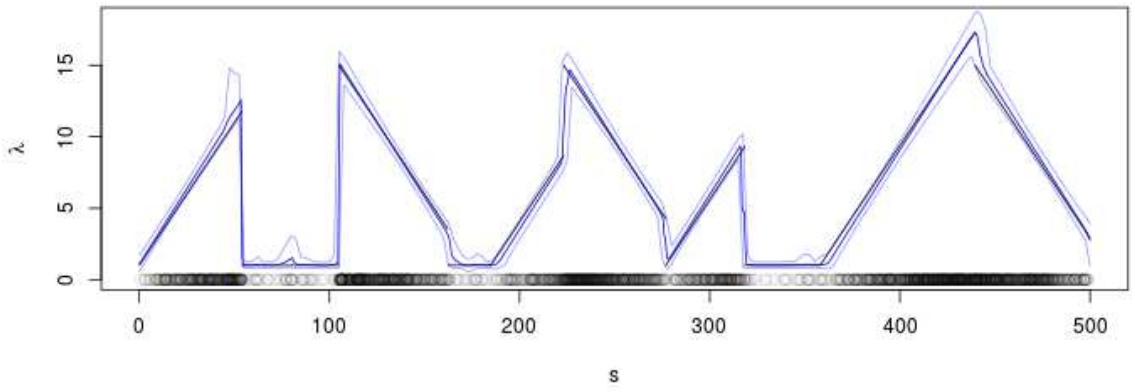
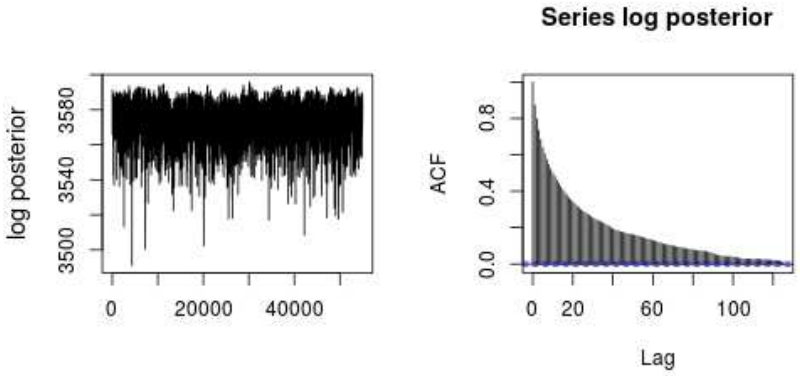


Figure B.10: Results for scenario 3A: Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.



(a) Posterior mean and 95% credibility interval (blue lines) for the IF. The black line is the true IF.



(b) Trace plot and ACF for the log posterior density.

Figure B.11: Results for scenario 3B.

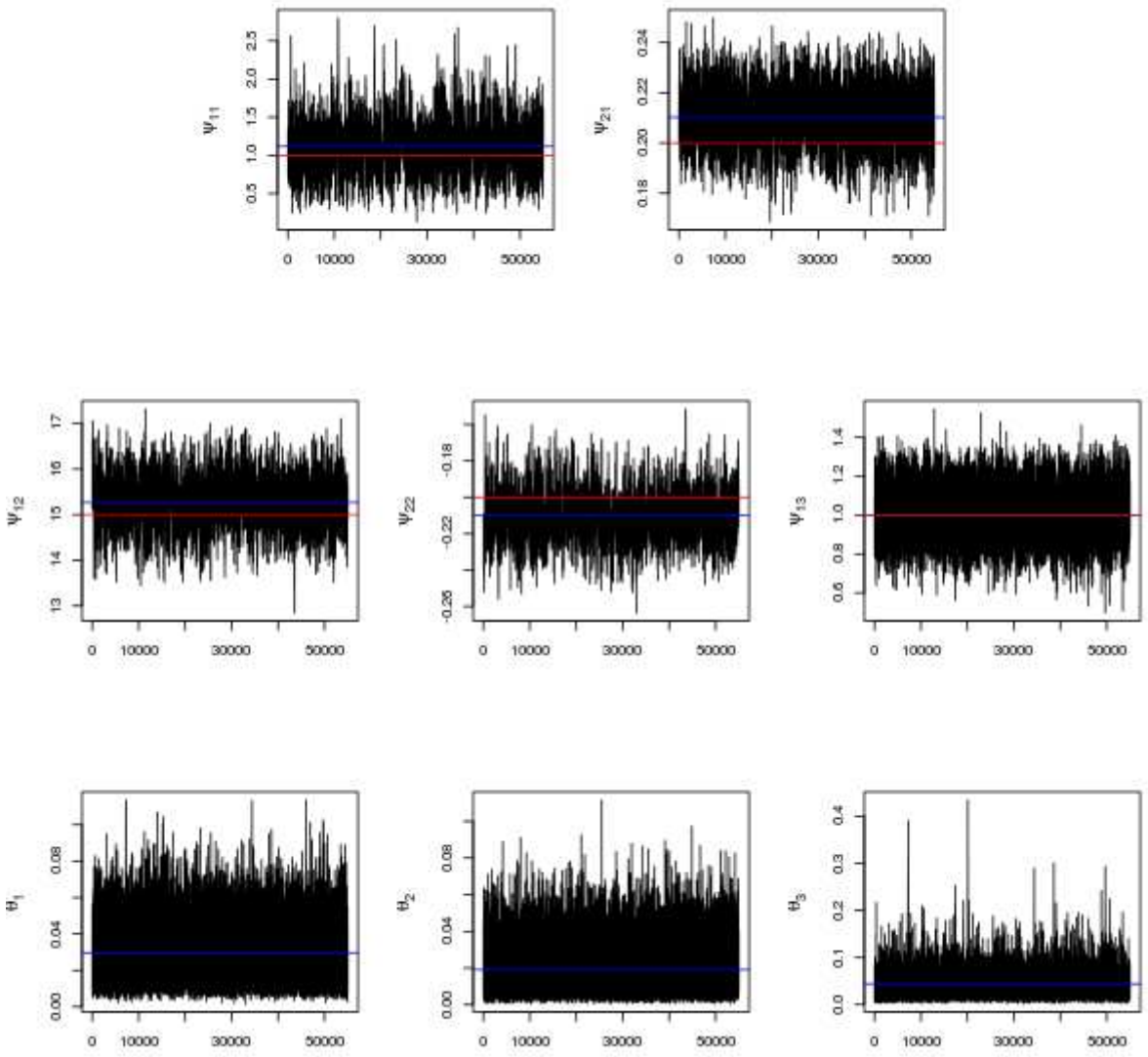
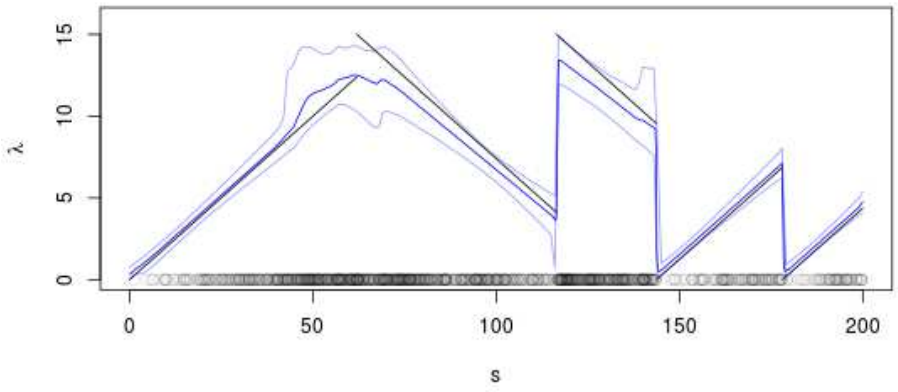


Figure B.12: Results for scenario 3B: Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.

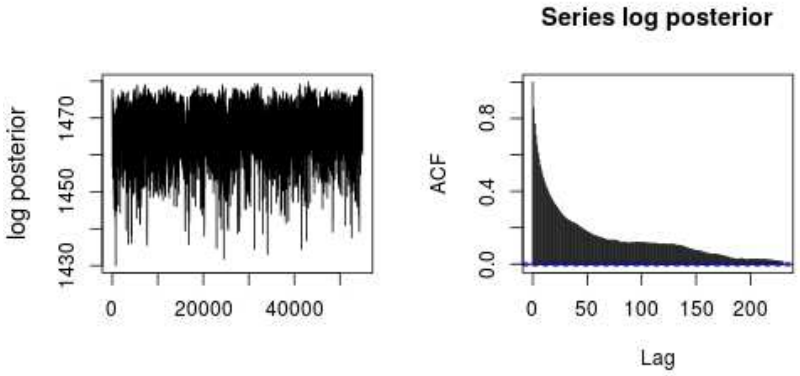
B.4 Scenario 4

	A	B
iterations	15k	17k
interval blocks	1	2
accept. rate (V_0, V)	0.5168	1^{st}: 0.6420; 2^{nd}: 0.6540
accept. rate ψ	ψ_1: 0.3738; ψ_2: 0.3042	ψ_1: 0.2705; ψ_2: 0.3510
iterations MCMC	55k	50k
burn-in	5k	20k
Total computational time	1m54.26s	4m02.81s

Table B.5: Computational results for scenario 4.



(a) Posterior mean and 95% credibility interval (blue lines) for the IF. The black line is the true IF.



(b) Trace plot and ACF for the log posterior density.

Figure B.13: Results for scenario 4A.

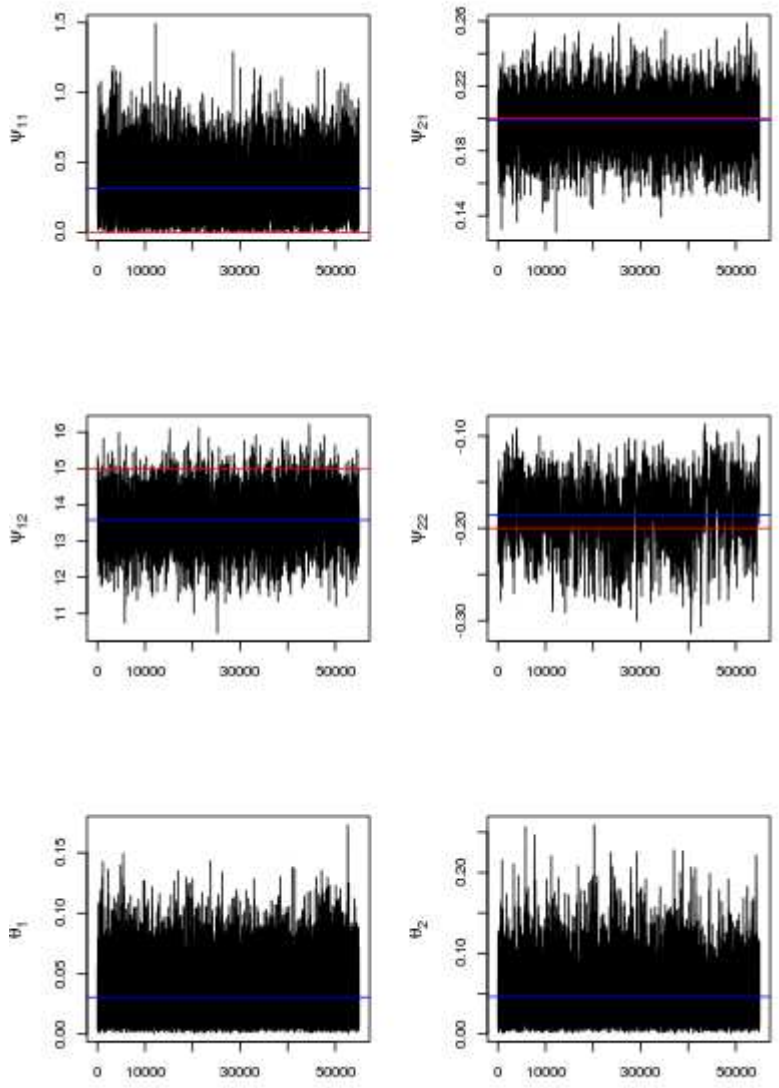
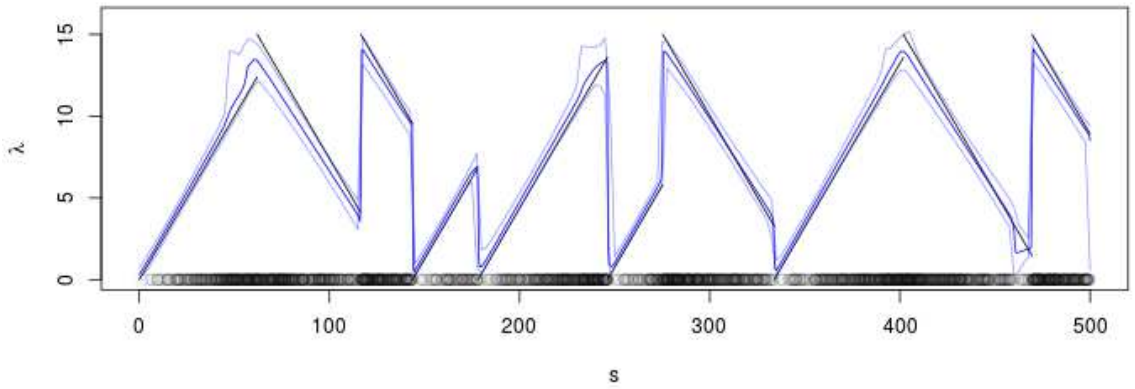
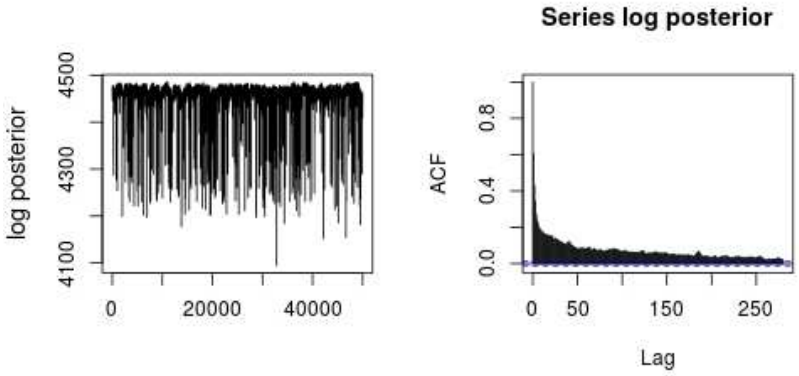


Figure B.14: Results for scenario 4A: Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.



(a) Posterior mean and 95% credibility interval (blue lines) for the IF. The black line is the true IF.



(b) Trace plot and ACF for the log posterior density.

Figure B.15: Results for scenario 4B.

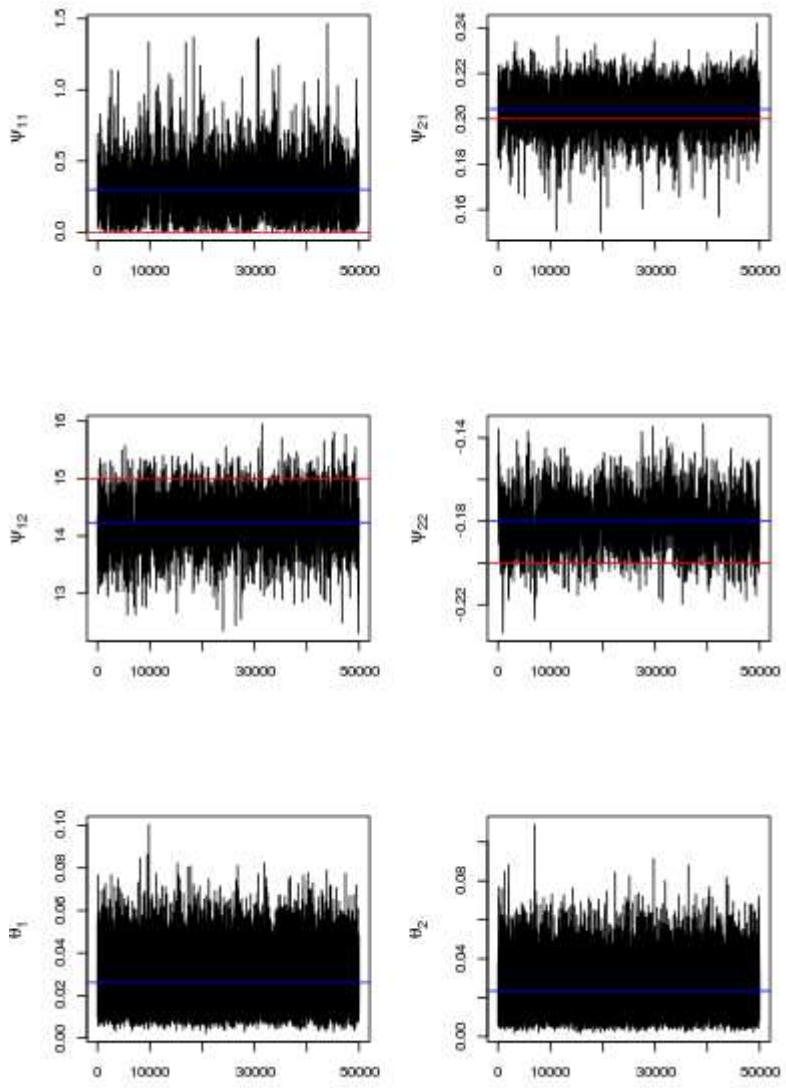
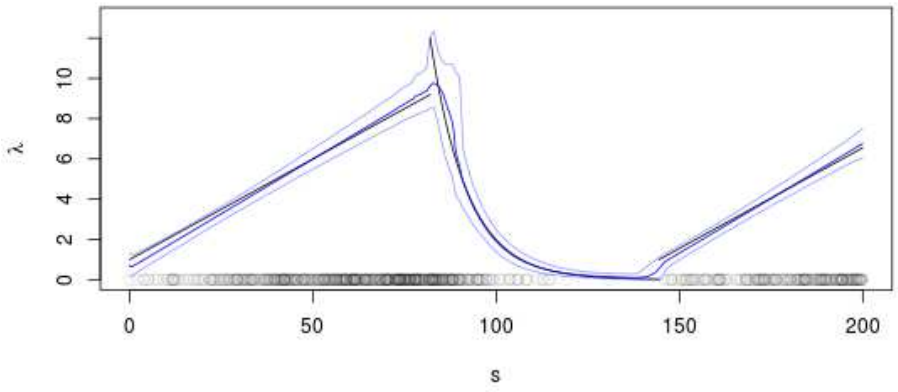


Figure B.16: Results for scenario 4B: Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.

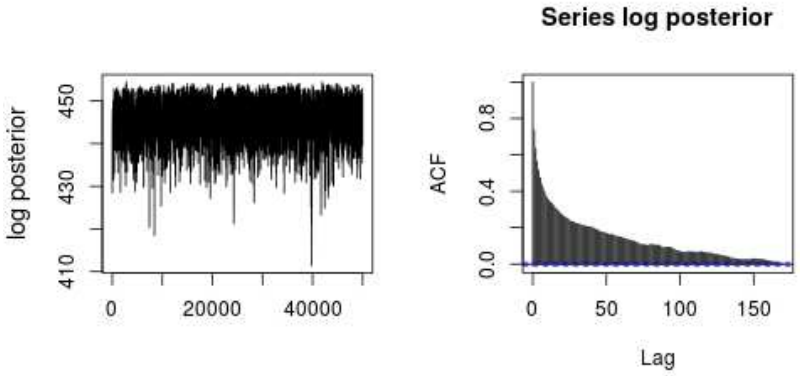
B.5 Scenario 5

	A	B
iterations	11k	5k
interval blocks	1	1
accept. rate (V_0, V)	0.7580	0.5653
accept. rate ψ	ψ_1: 0.3731; ψ_2: 0.3574	ψ_1: 0.3921; ψ_2: 0.3555
iterations MCMC	50k	100k
burn-in	5k	5k
Total computational time	1m06.38s	5m05.27s

Table B.6: Computational results for scenario 5.



(a) Posterior mean and 95% credibility interval (blue lines) for the IF. The black line is the true IF.



(b) Trace plot and ACF for the log posterior density.

Figure B.17: Results for scenario 5A.

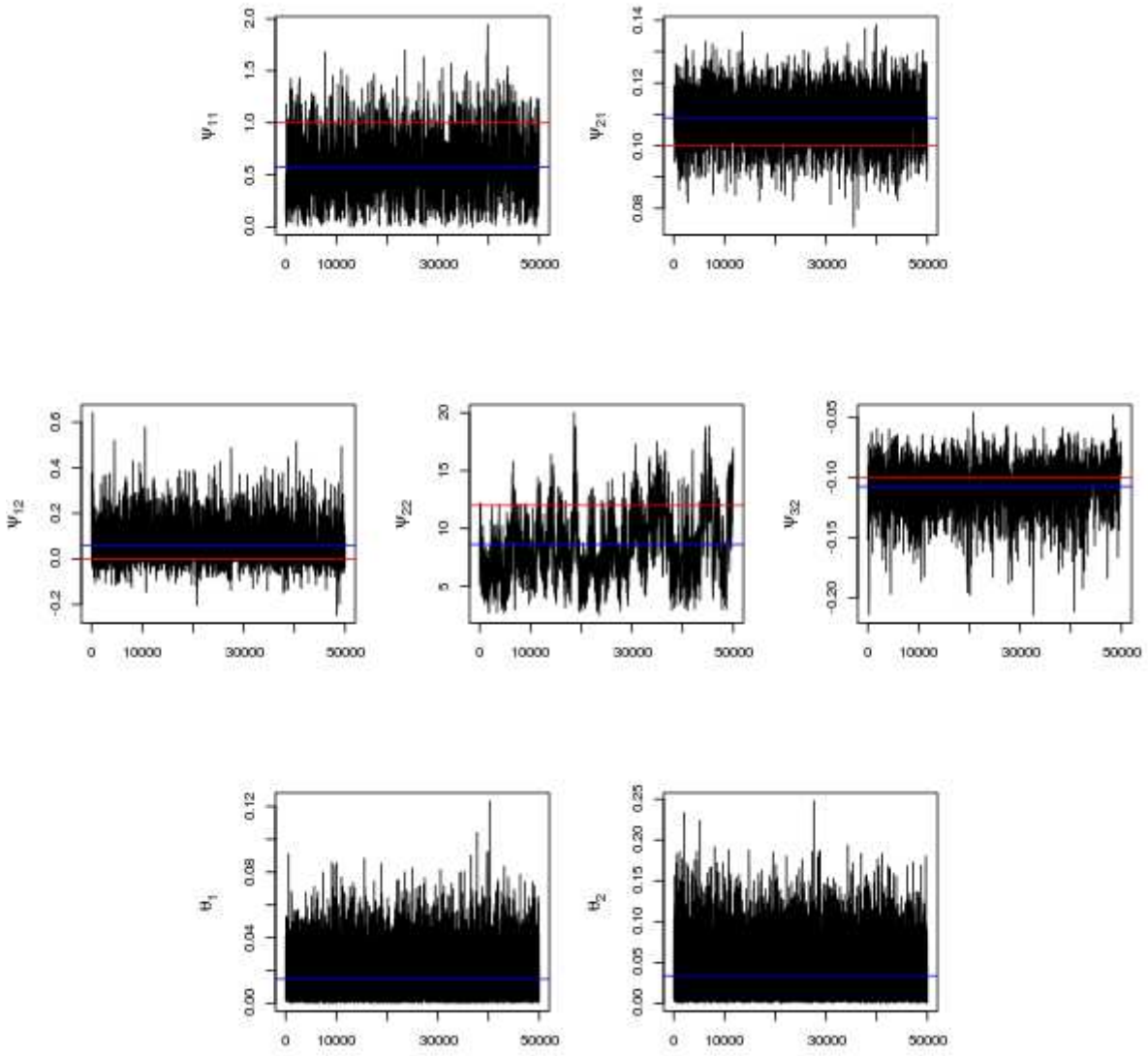
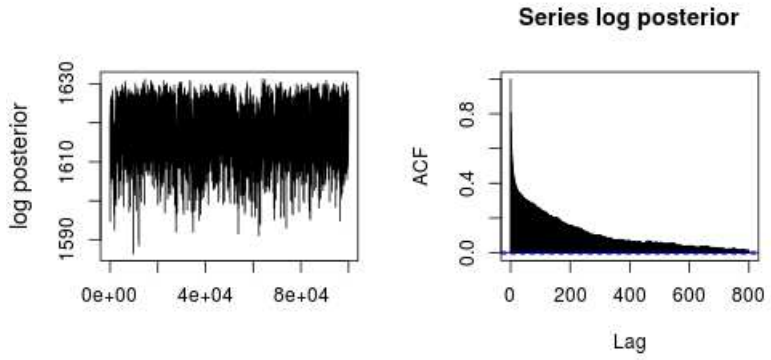
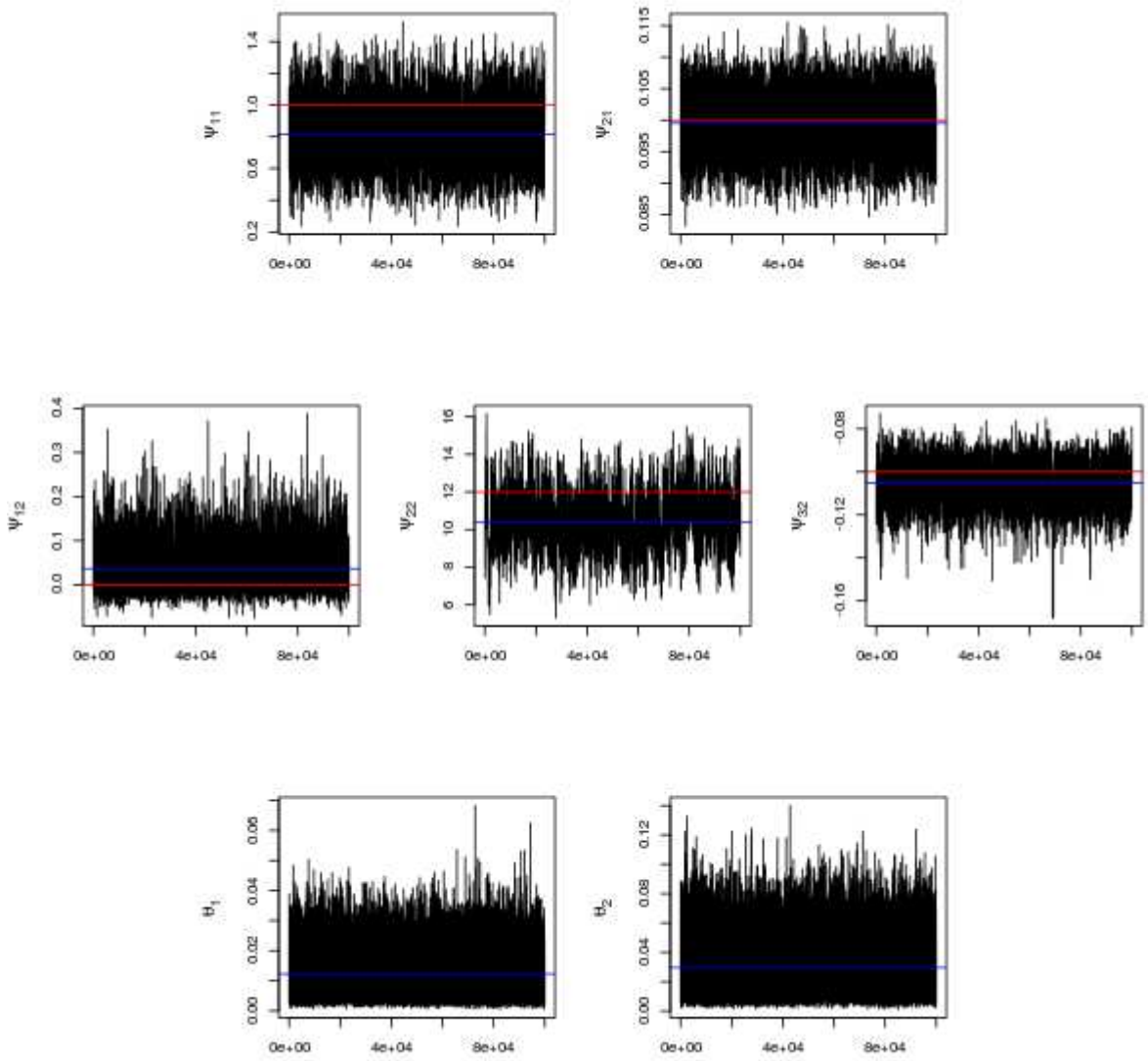


Figure B.18: Results for scenario 5A: Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.



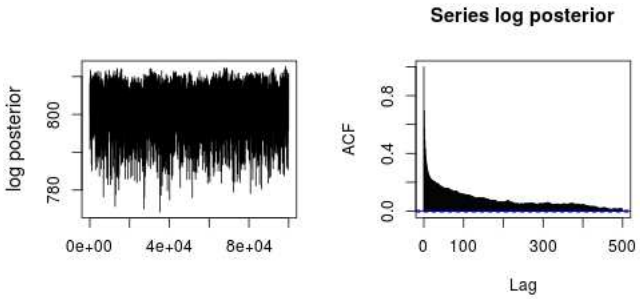
(a) Trace plot and ACF for the log posterior density.



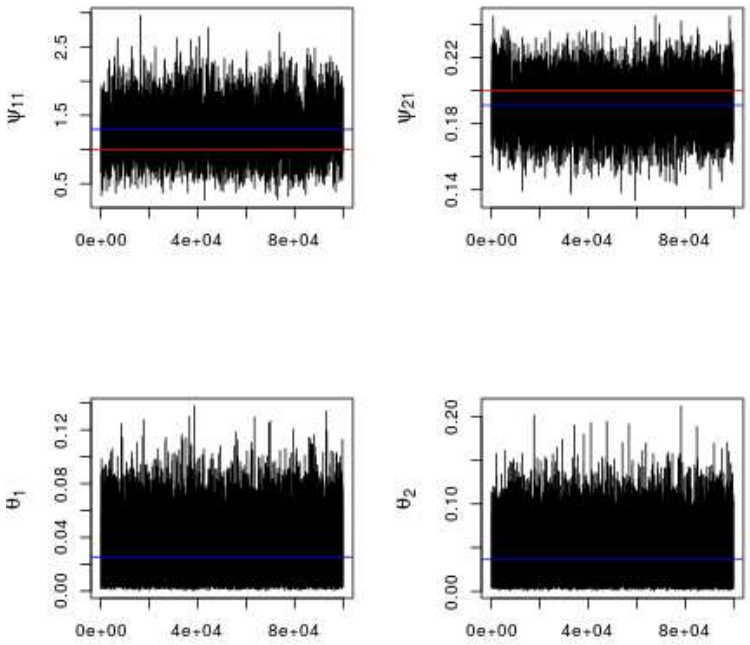
(b) Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.

Figure B.19: Results for scenario 5B.

B.6 Scenario 6



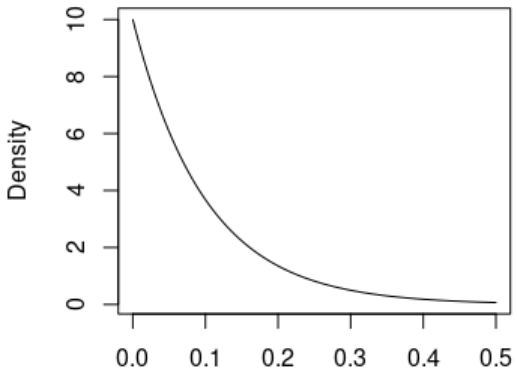
(a) Trace plot and ACF for the log posterior density.



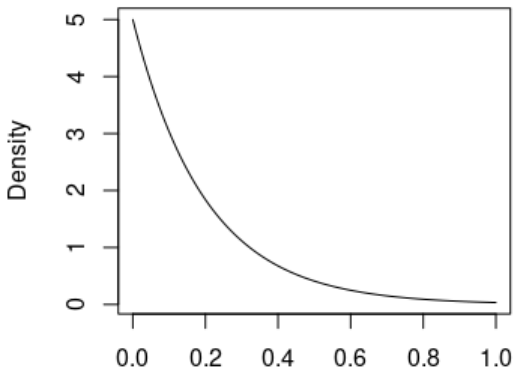
(b) Trace plots for parameters ψ and θ. The red line is the parameter true value and the blue line is the posterior mean.

Figure B.20: Results for scenario 6.

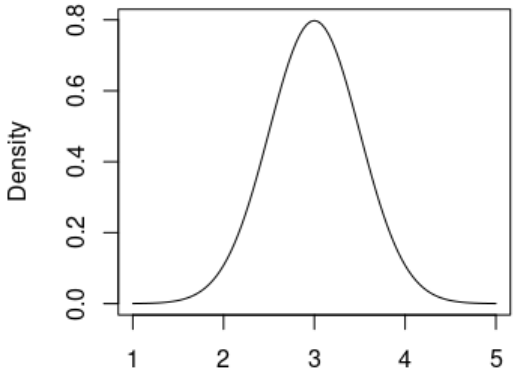
B.7 Prior specifications



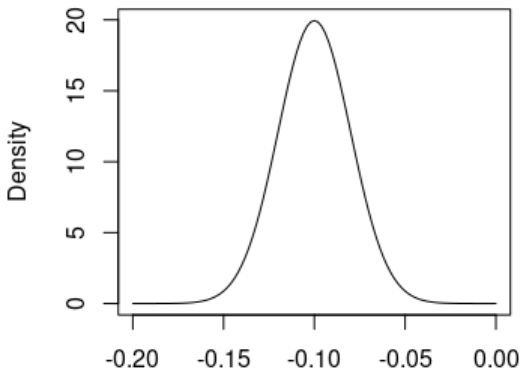
(a) Gamma$(1, 10)$.



(b) Gamma$(1, 5)$.

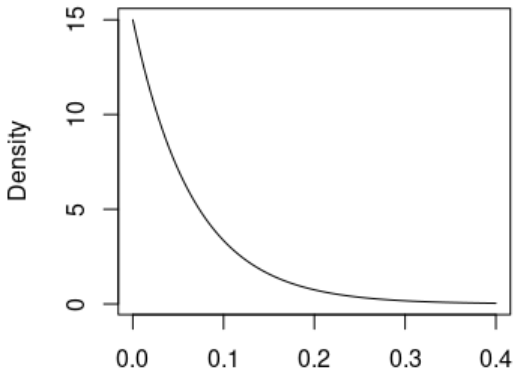


(c) Normal$(3, 0.5^2)$.

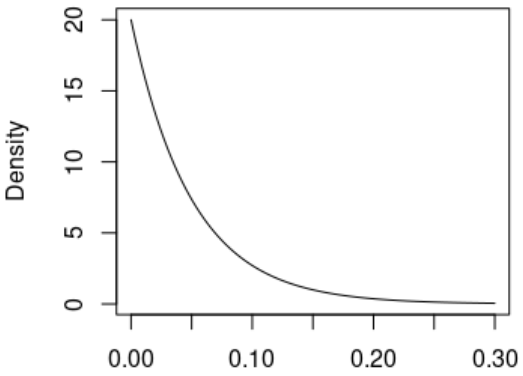


(d) Normal$(-0.1, 0.02^2)$.

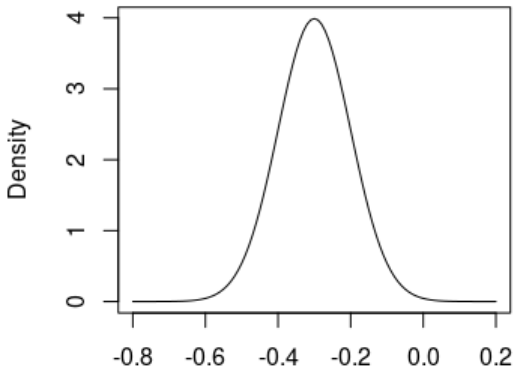
Figure B.21: Probability densities.



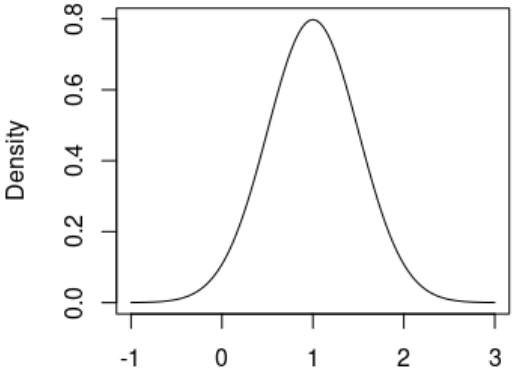
(a) Gamma$(1, 15)$.



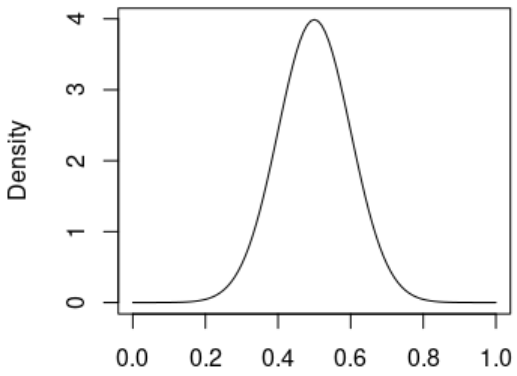
(b) Gamma$(1, 20)$.



(c) Normal$(-0.3, 0.1^2)$.



(d) Normal$(1, 0.5^2)$.



(e) Normal$(0.5, 0.1^2)$.

Figure B.22: Probability densities.

B.8 Real data analysis

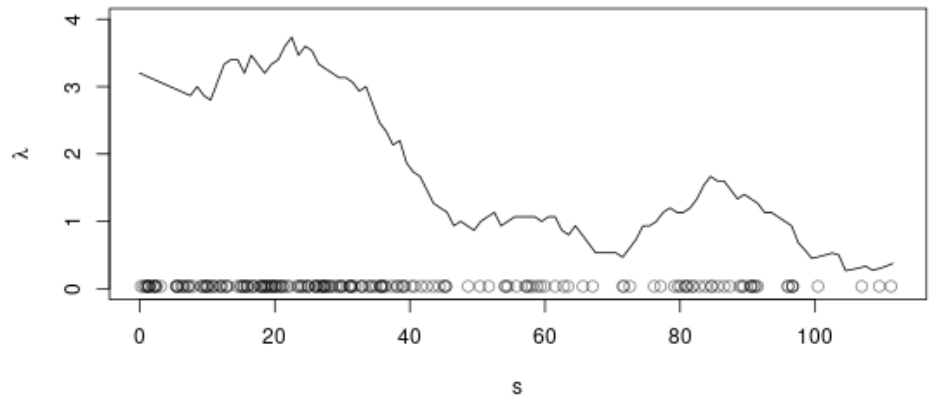


Figure B.23: Empirical estimation of the IF for coal mine data.

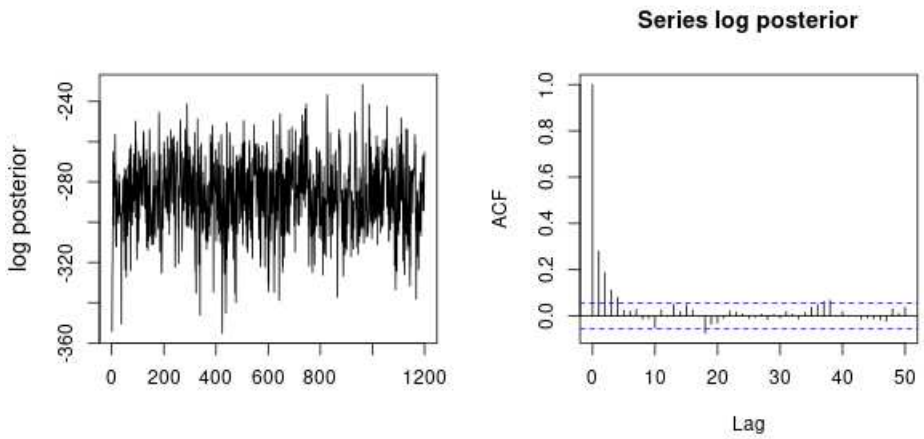
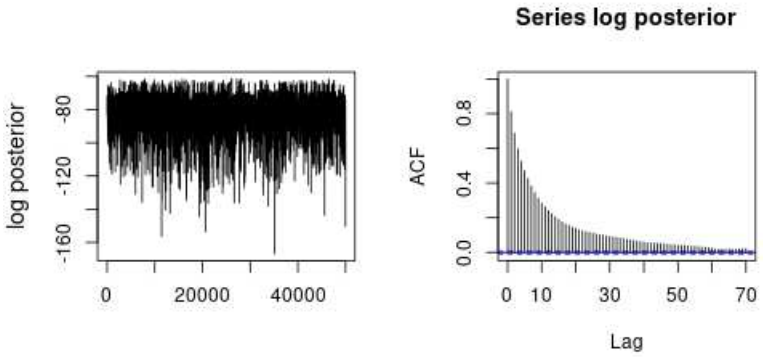
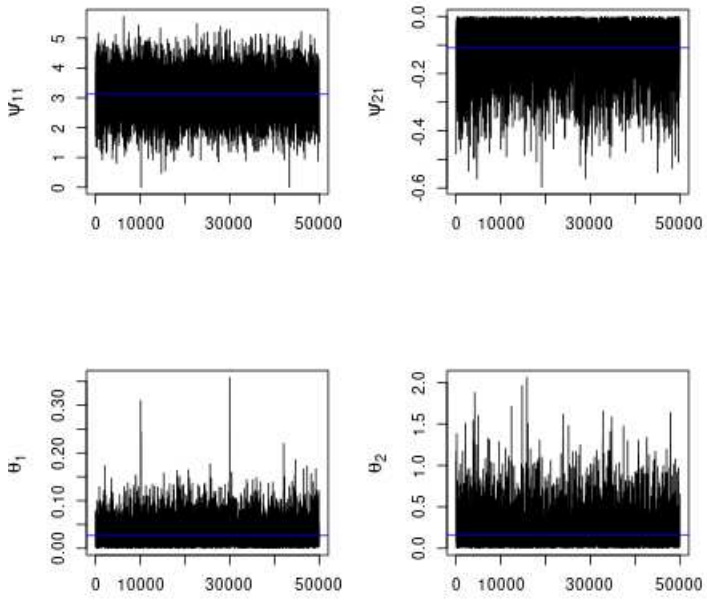


Figure B.24: Trace plot and ACF for the log posterior density using a non-parametric model for the coal mine data.



(a) Trace plot and ACF for the log posterior densities.



(b) Trace plots for parameters ψ and θ. The blue line is the posterior mean.

Figure B.25: Results for the coal mine data.

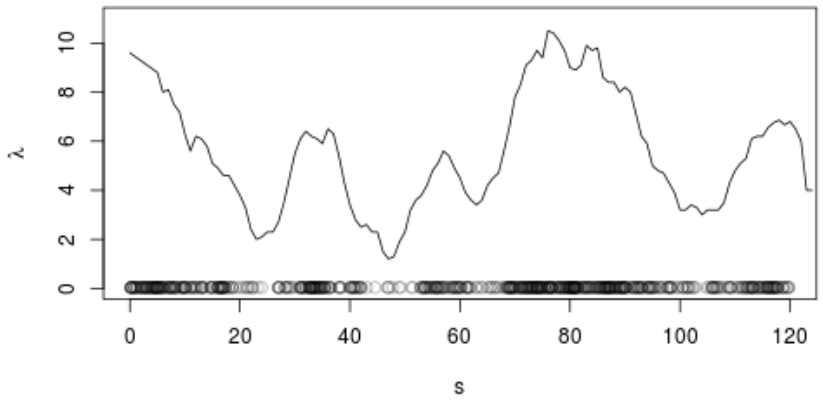
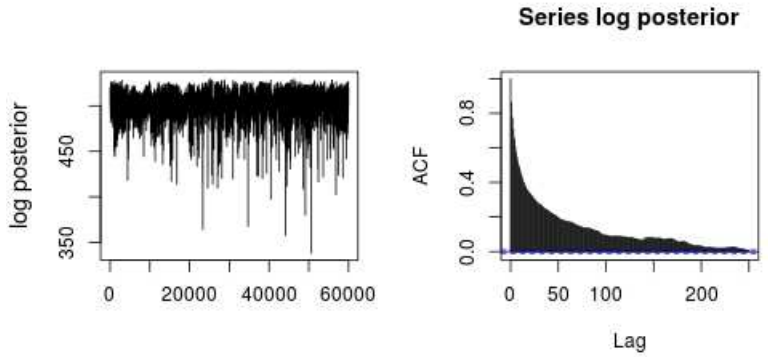
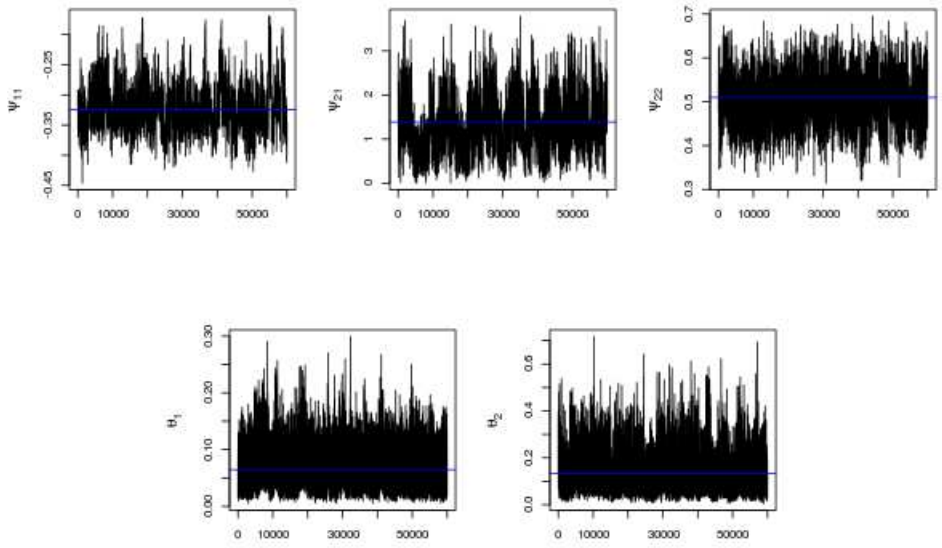


Figure B.26: Empirical estimation of the IF for exchange rate data.



(a) Trace plot and ACF for the log posterior densities.



(b) Trace plots for parameters ψ and θ. The blue line is the posterior mean.

Figure B.27: Results for the exchange rate data.

Bibliography

Amarasingham, A., Chen, T.-L., Geman, S., Harrison, M. T., & Sheinberg, D. L. (2006). Spike count reliability and the Poisson hypothesis. *Journal of Neuroscience*, *26*(3), 801–809.

Andersen, P. K., Borgan, O., Gill, R. D., & Keiding, N. (1993). *Statistical models based on counting processes*. Springer, New York.

Cox, D. R. (1955). Some statistical methods connected with series of events. *Journal of the Royal Statistical Society. Series B (Methodological)*, 129–164.

Cunningham, J. P., Byron, M. Y., Shenoy, K. V., & Sahani, M. (2008). Inferring neural firing rates from spike trains using Gaussian processes. *Advances in Neural Information Processing Systems*, 329–336.

Dassios, A., & Jang, J.-W. (2003). Pricing of catastrophe reinsurance and derivatives using the Cox process with shot noise intensity. *Finance and Stochastics*, *7*(1), 73–95.

Doornik, J. A. (2009). An object-oriented matrix programming language Ox 6.

Fearnhead, P., & Sherlock, C. (2006). An exact Gibbs sampler for the Markov-modulated Poisson process. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(5), 767–784.

Fischer, W., & Meier-Hellstern, K. (1992). The Markov-modulated Poisson process (MMPP) cookbook. *Performance evaluation*, *18*(2), 149–171.

Gonçalves, F. B., & Franklin, P. (2019). On the definition of likelihood function. *arXiv:1906.10733*.

Gonçalves, F. B., & Gamerman, D. (2018). Exact Bayesian inference in spatio-temporal Cox processes driven by multivariate Gaussian processes. *Journal of the Royal Statistical Society, Series B*, *80*, 157–175.

Hobolth, A., & Stone, E. A. (2009). Simulation from endpoint-conditioned, continuous-time Markov chains on a finite state space, with applications to molecular evolution. *The Annals of Applied Statistics*, *3*(3), 1204.

Jarrett, R. G. (1979). A note on the intervals between coal-mining disasters. *Biometrika*, *66*, 191–193.

Lando, D. (1998). On Cox processes and credit risky securities. *Review of Derivatives research*, *2*(2-3), 99–120.

Landon, J., Özekici, S., & Soyer, R. (2013). A Markov modulated Poisson model for software reliability. *European Journal of Operational Research*, *229*(2), 404–410.

Liu, J. S. (1994). The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, *89*(427), 958–966.

Mengersen, K. L., & Tweedie, R. L. (1996). Rates of convergence of the Hastings and Metropolis algorithms. *The Annals of Statistics*, *24*(1), 101–121.

Rao, V., & Teh, Y. W. (2013). Fast MCMC sampling for Markov jump processes and extensions. *The Journal of Machine Learning Research*, *14*(1), 3295–3320.

Roberts, G. O., & Rosenthal, J. S. (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, *18*(2), 349–367.

Scott, S. L. (1999). Bayesian analysis of a two-state Markov modulated Poisson process. *Journal of Computational and Graphical Statistics*, *8*(3), 662–670.

Scott, S. L., & Smyth, P. (2003). The Markov modulated Poisson process and Markov Poisson cascade with applications to web traffic data. *Bayesian Statistics*, *7*, 671–680.

Shao, J. (2003). *Mathematical statistics*. Springer texts in statistics.

Smith, J. A., & Karr, A. F. (1983). A point process model of summer season rainfall occurrences. *Water Resources Research*, *19*(1), 95–103.