UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING

School of Electronics and Computer Science

Inference and Learning in State-Space Point Process Models: Algorithms and Applications

by

Ke Yuan

Thesis for the degree of Doctor of Philosophy

May 2013

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
School of Electronics and Computer Science

Doctor of Philosophy

INFERENCE AND LEARNING IN STATE-SPACE POINT PROCESS MODELS: ALGORITHMS AND APPLICATIONS

by Ke Yuan

Physiological signals such as neural spikes and heart beats are discrete events in time, driven by a continuous underlying system. A recently introduced data driven model to analyse such systems is the state-space model with point process observations (SSPP), parameters of which and the underlying state sequence are simultaneously identified in a maximum likelihood setting using an approximate expectation-maximization (EM) algorithm. This thesis provides a detailed study on the property of SSPP under the EM setting. The results strongly suggest that the Bayesian treatment is more appropriate to avoid biased estimation. For this we develop the variational methods, and a range of efficient Markov chain Monte Carlo methods. The performance of these inference mechanisms is throughly tested on both synthetic and real world datasets.

DECLARATION OF AUTHORSHIP

I, Ke Yuan, declare that the thesis entitled *Inference and Learning in State-Space Point Process Models: Algorithms and Applications* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as:
  - Yuan, K, Girolami, M. and Niranjan, M. (2012). Markov chain Monte Carlo methods for state-space models with point process observations. *Neural Computation*, 24(6): 1462–1486.
  - Zammit Mangion, A., Yuan, K., Kadirkamanathan, V., Niranjan, M., and Sanguinetti, G. (2011). Online variational inference for state-space models with point process observations. *Neural Computation*, 23(8):1967–1999.
  - Yuan, K. and Niranjan, M. (2010). Estimating a state-space model from point process observations: A note on convergence. *Neural Computation*, 22(8):1993–2001.

Signed..Date...

Contents

List of Figures

List of Tables

Acknowledgements

I would like to gratefully acknowledge my supervisor Prof. Mahesan Niranjan. My PhD training has benefited enormously from his expertise, enthusiasm, and encouragement. Throughout the years, Niranjan is not only a mentor but also a colleague and a friend. His excitement about and commitment to research set a strong example that I hope to emulate as I grow in academia.

I have been very fortunate to receive regular advice and ideas from Dr. Chunqi Chang, Dr. Guido Sanguinetti and Prof. Mark Girolami. Their contributions have brought this thesis to a higher level. I would like to thank Dr. Andrew Zammit Mangion for many hours of intriguing discussions on a variety of topics.

This work is carried out at the CSPC group at the University of Southampton. Many group members provide me with kindness help during my PhD. I would like to thank Dr. Ivan Markovsky for the invaluable discussions and guidance that have helped me move forward. I am also benefited hugely from the discussions with Dr. Adam Prugel-Bennett and Dr. Srinandan Dasmahapatra.

I would like to thank my colleagues in the group, who offered me their constant support, advice and good humors, in particular Abdullah Alrajeh, Tayyaba Azim, Chathurika Dharmagunawardhana, Dr. Bassam Farran, Dr. Mustansa Ali Ghazanfar, Dr. Ali Hassan, Dr. Sung Uk Jung, Wanmu Liu, Xin Liu, Shaobai Li, Dr. Yizhao Ni, Dr. Amirthalingam Ramanan, Dr. Somjet Suppharangsan, Daisy Tong, Dr. Salih Tuna and others who are not mentioned here.

Special thanks to my wife Wei Liu and for the happiness life she shares with me. Having her beside is best thing ever happened to me. I would like to thank my parents for their patience and support help me on every step of the way.

Finally, I would like to acknowledge the financial support from School of Electronics and Computer Science.

To my wife Wei and my parents Xiaowei and Xiaolu

Chapter 1

Introduction

1.1 Problem statement

Many biomedical signal processing problems, such as neural spikes and heartbeats, are concerned with discrete events in time, separated by seemingly random intervals. They are often driven by continuous processes relating to the organ's physiology, whose charge-and-fire type behaviour results in observed discrete events.

The neural spikes are indicators of neural activities. They are measured by electrodes implanted into the subject's cortex surface[1], while some controlled stimulations are applied. Figure 1.1(a) shows an illustration of such an experiment on a monkey. Understanding the relation between spikes and stimuli is a grand task in neuroscience, which could at some stage enable "mind reading"; see for example figure 1.2.

Likewise, heartbeats are also a binary sequence, if only the R-waves (the big pulse on ECG) are considered. In this case, the variations between beat-to-beat intervals or heart rates are known to be the results of underlying control inputs from the autonomic nervous system. Such a regulation mechanism is demonstrated in figure 1.1(b). Being able to estimate the conditions under which regulation occurs could lead to a diagnostic tool for detecting heart diseases.

Towards a mathematical framework for characterising the above two physiological problems, Smith and Brown (2003) proposed a state-space model with point process observations (SSPP), which avoids the somewhat artificial change to inter-event times and handles the discrete events directly. This model assumes a first-order autoregressive process driven by an exogenous stimulus as state dynamics and an approximate Bernoulli process with a parameterised intensity function as its observation model. The states

[1] The operation does serious damage to the experimental subject. Therefore this type of experiment is rarely performed on humans. The common subjects are fly, rat and monkey.

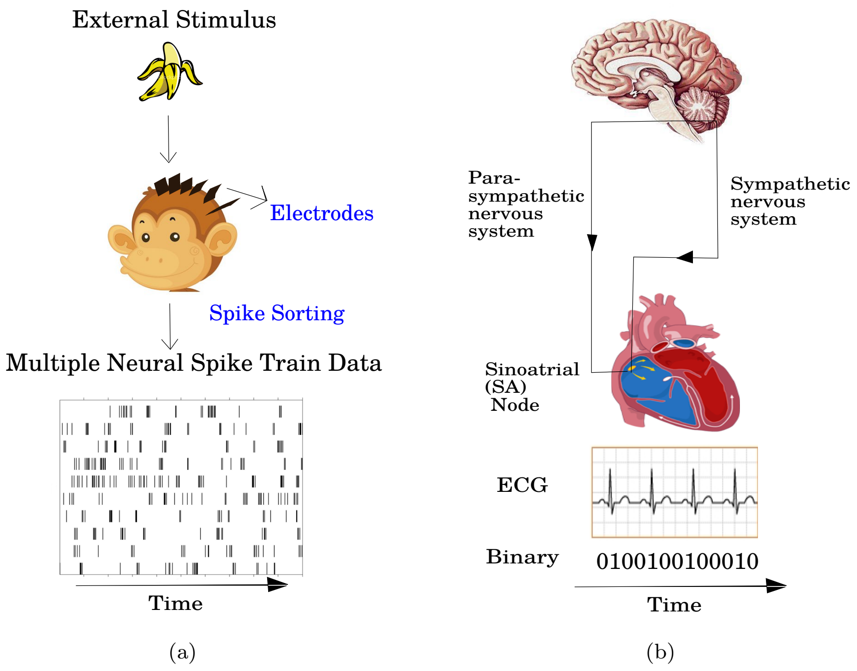


Figure 1.1: (a): An illustration of a typical experiment investigating neural spikes (or neural spike trains) in response to external stimuli. (b): An illustration of autonomic nerves systems control circuits.

and parameters, serving as discriminating features, could potentially provide solutions to the two problems.

For simultaneous estimation of states and parameters of SSPP, Smith and Brown derived an expectation-maximization (EM) algorithm (Dempster et al., 1977). The quality of estimation results from EM in SSPP is yet fully understood. This thesis takes this as its starting point, and provides a detailed study of the convergence of EM in SSPP. A particular property, the highly skewed likelihood function, motivates the development of a range of more powerful Bayesian methods, including the variational Bayes (VB) methods and Markov chain Monte Carlo (MCMC) methods.

1.2 Contributions

This thesis has made the following four contributions:

- We have established that the filtering and smoothing distributions of SSPP are concave. This is done by identifying three sufficient conditions that ensure an arbitrary state-space model to have concave filtering and smoothing distributions. Moreover, due to the fact that the expectation of the log-joint data likelihood is highly skewed in most parameters, the estimation is biased. These results pave

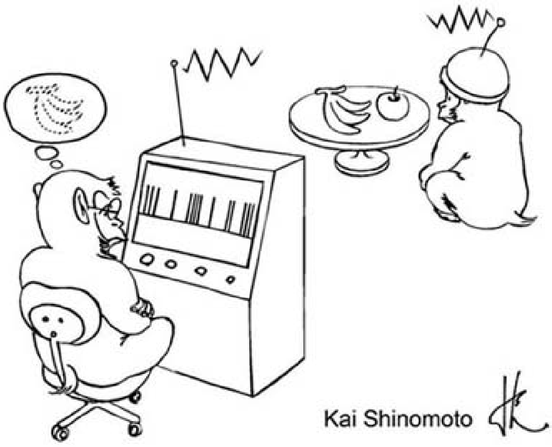


Figure 1.2: A cartoon example of mind reading, adapted from Shigeru Shinomoto's homepage: `http://www.ton.scphys.kyoto-u.ac.jp/~shino/english.html`

the way for more sophisticated Bayesian approaches. A paper based on this contribution is that of Yuan and Niranjan (2010).

- In a collaboration with Andrew Zammit Mangion, we have developed VB methods for SSPP in both offline (batch) and online (sequential) settings. The results on synthetic datasets are compared with a Gibbs sampler (offline) and a particle filter (online). Further, the methods are tested on rat taste response dataset. The estimated parameters successfully separate different taste stimuli. A publication based on this work is that of Zammit Mangion et al. (2011b).

- A wide range of efficient MCMC (approximate MCMC) methods are proposed for SSPP, including the latest particle MCMC (Andrieu et al., 2010) and geometry based MCMC (Girolami and Calderhead, 2011). In addition, two approximate MCMC methods targeting the marginal likelihood of SSPP are also developed. The efficiency of these methods is assessed on a synthetic dataset, where the geometry-based MCMC shows the best performance. Further, MCMC and VB are compared on two real datasets. Part of this work is published as Yuan et al. (2012).

- A point process generalised linear model is developed for human heartbeat. The candidate frequency components of autonomic nervous system controls and history information of the heartbeats are related to the observed heartbeats sequence. The model is examined on datasets from healthy subjects and patients who have suffered sudden cardiac death. The estimated parameters show comparable results on separating the two groups with some traditional features.

1.3 Thesis organisation

The reminder of this thesis is organized as follows. Chapter 2 gives a literature review of neural spikes data analysis, heart rate variability, state-space models and inference and learning algorithms. Chapter 3 introduces the SSPP and its properties under the EM setting. In chapter 4, we present a variational Bayes framework for SSPP. Chapter 5 introduces a range of MCMC methods for SSPP. Subsequently, the case studies with real neural spikes and human heartbeat datasets are illustrated in chapter 6. Finally, chapter 7 concludes the thesis, and outlines some directions for further study.

Chapter 2

Literature review

In this chapter, we review some related works on modelling neural spikes and human heartbeats. In addition, a wide range of state-of-the-art inference and learning algorithms for state-space models are also discussed.

2.1 Binary physiological time series

2.1.1 Neural spikes

Understanding neural response to sensory input is one of the central problems in neuroscience. This can be traced all the way back to Adrian (1928), who established the fact that neural activities are expressed through series of electrical pulses, formally known as action potentials. These action potentials, or spikes (and spike trains), are sole indicator of neural activities. This means that, when evoking stimulus, a neuron will either produce a spike or do nothing. Further, the duration of each spike is very short, typically $<$ 1ms; as a result, the difference between the wave forms of spikes tends to be ignored. Figure 2.1 shows an example of action potentials, from different neurons under different stimulation, appearing to be the same individually. The timing of spikes, however, becomes the only relevant information (Dayan and Abbott, 2001). To this end, the spikes can be seen as the neural coding for the sensory stimuli (Rieke et al., 1997).

Throughout the development of neuroscience, mathematical modelling has played a vital role in gaining insight on the neural response. A representative example is the Hodgkin-Huxley model which successfully characterises the dynamic properties of the spikes in the giant axon of squid (Hodgkin and Huxley, 1952). The Hodgkin-Huxley model is a classical differential equation model. In other words, it is designed to describe the physical or phenomenal nature of the spikes. However, this type of model normally misses out on the randomness of the spikes. Such randomness is clearly present in

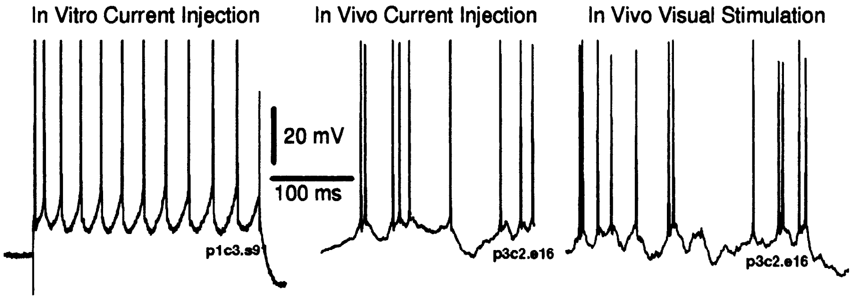


Figure 2.1: Adapted from Holt et al. (1996) (also from Dayan and Abbott (2001)). Action potentials from cat V1 neurons in response to three different stimulus conditions. Particularly, current injection in vitro (*left*), the current injection in *vivo* (*centre*), and moving visual image stimuli in *vivo* (*right*).

the timing of the spikes. To this end, many statistical models are heavily involved in modelling neural spikes (Kass et al., 2005).

As the spikes are identical to each other, many models tend to convert them into rates or the firing rates in spikes per second unit. The simplest of this kind is the prei (or post)-stimulus time histogram (PSTH). In detail, it counts the spike number in different length of time slots which is essentially a firing rate histogram (Rieke et al., 1997; Dayan and Abbott, 2001; Kass et al., 2005). Of course, one can use traditional signal processing methods such as the sliding window to obtain the firing rate estimates as well. Once the spike train is converted to a continuous-valued firing rate time series, it is then easy to model it with a linear regression framework; for example, the reverse regression approach where the spike count can be seen as given input to predict stimulus (Stanley et al., 1999; Warland et al., 1997). Particularly, Stanley et al. (1999) used this method to decode natural scenes from cat lateral geniculate nucleus neurons.

Quite often, the spike trains are available in vector form, meaning there are multiple neural activities recorded simultaneously from an array of electrodes as shown in figure 2.2 for example. Examples include the cat experiment mentioned above (Stanley et al., 1999), decoding positional representation from rat hippocampal neurons (Brown et al., 1998; Zhang et al., 1998), decoding velocity from fly H1 neural activities (Bialek et al., 1991). These multiple neural spike trains are known as the *population coding* (Georgopoulos et al., 1986). To this end, a successful framework is the population vector algorithm (PVA) proposed by Georgopoulos et al. (1982, 1986), which has been used to model cortical control of arm movement (Moran and Schwartz, 1999; Ruiz et al., 1995; Schwartz et al., 2004). Specifically, the PVA is still a linear regression model, but with a more conventional setting, which takes some preferred directions of the arm movement as inputs. The regression outputs (or predictions) are the firing rates.

In addition, there are more traditional signal processing methods based on firing rate, including Fourier analysis of the spikes (Brillinger, 1992; Jarvis and Mitra, 2001) and

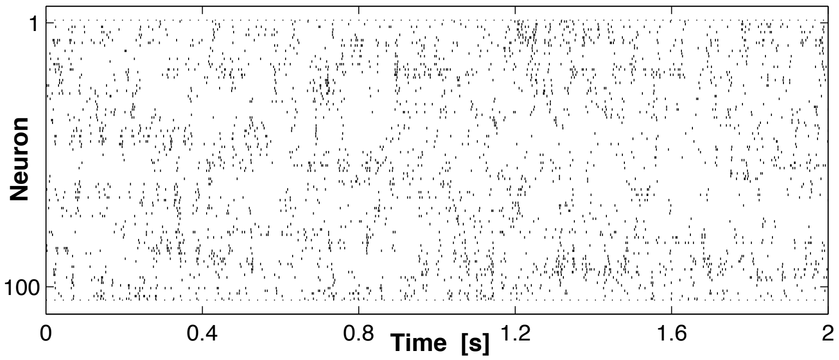


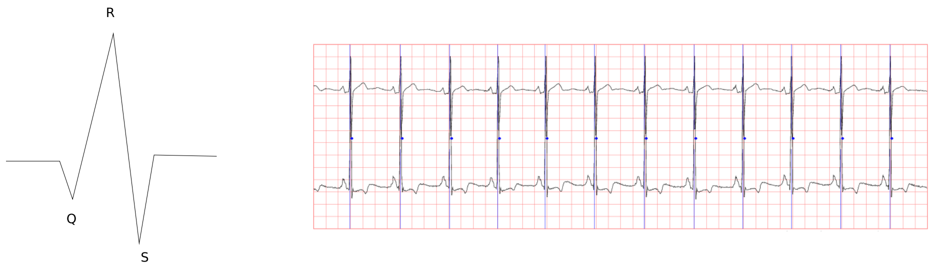
Figure 2.2: Adapted from Kreuz et al. (2011). Spike trains from 105 retinal ganglion cells (neurons) in response to the same stimulation. Each dot represents a spike.

wavelet-based methods (Percival and Walden, 2002). These methods are used for analysing association between spike trains. A good summary of these methods can be seen in Brown et al. (2004).

Obtaining the firing rates itself is not, however, an easy task. Ideally, one would want the firing rates to capture the underlying neural dynamics, meaning that what really matters is the instantaneous information of the firing rates. In practice, the firing rates are often computed over predefined time bins, or converted from inter-spike intervals. Either case suffers loss of instantaneous information of the spiking activities. Therefore more sophisticated methods to estimate the firing rates are necessary.

Alternatively, the spikes can be treated as binary codes, and modelled with point processes (Cox and Isham, 1980; Daley and Vere-Jones, 2003), which allows the spikes to be handled directly. In this framework, the firing rates become a time-varying parameter. More specifically, this parameter is called *conditional intensity function* (CIF) which is often formulated as a linear combination between some basis functions and parameters. This formulation is also known as the generalised linear model (GLM) (McCullagh and Nelder, 1989). In the context of spike train modelling, this model is referred to as the point process GLM (PPGLM). Under the PPGLM setting, the basis functions represent some given inputs or stimuli (e.g. arm movement direction). The parameters, however, need to be estimated. To this end, Paninski (2004); Okatan et al. (2005); Chen et al. (2010); Ahmadian et al. (2011) illustrated how the estimation can be done in both frequentist and Bayesian fashion.

Towards a more flexible model, the basis function can be a stochastic process as well, forming the *doubly stochastic point process* (DSPP) or Cox process (Daley and Vere-Jones, 2003). The state-space models with point process observation (SSPP) proposed by Smith and Brown (2003) belong to this family. Another class of methods consider the parameter to be time-varying under the PPGLM setting. The estimation becomes a



(a) (b)

Figure 2.3: (a): Schematic representation of normal QRS complex on ECG. (b): A 10 seconds ECG data from record itdb/14046 in MIT-BIH long-term ECG database in PhysioNet database.

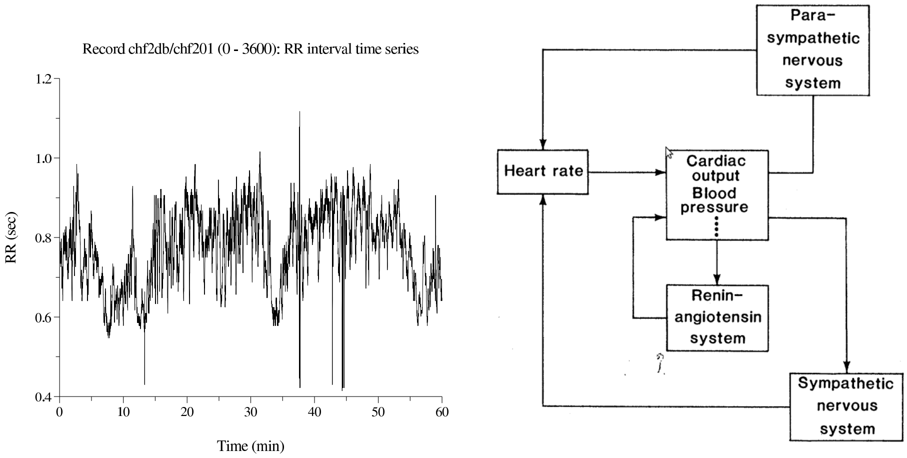
filtering problem in state-space model as per, for example, Eden et al. (2004); Ergün et al. (2007). These methods can be also seen as SSPP, however, they should be categorized as online estimation for PPGLM.

2.1.2 Human heartbeats

Heartbeats, heart rates and heart rate variability (HRV) play a key role in cardiovascular study. Heartbeats are indicated by the QRS complex, which is a combination of the Q-wave, R-wave and S-wave, on the electrocardiography (ECG), also known as a cardiac cycle. Figure 2.3 shows a schematic diagram for a standard QRS complex and a ECG segment. Within the QRS complex, the R-wave represents a successful heartbeat. Based on the timing of the heartbeats, heart rate is traditionally measured by two approaches: Firstly, the average of the reciprocal of the R-R intervals within a specified time window. Secondly, the number of R-wave events in unit time on ECG. As a contrast to the neural spikes, heart rate rarely responses to the outside world directly. Instead, the variations in heart rates, formally called heart rate variability (HRV)[1], are determined by regularisation from the autonomic nervous system (ANS). These ANS controls are split into two branches: the sympathetic and para-sympathetic nervous systems. Akselrod et al. (1981) proposed a model of the regularisation mechanism and estimated the two ANS inputs with spectral method. Figure 2.4 presents an example HRV and the regulatory circuits proposed by Akselrod et al. (1981).

Four decades ago, Hon and Lee (1963) showed that the foetal distress is associated with appreciable changes in HRV. After this significant discovery, traditional signal processing methods for HRV modelling are employed to assess many diseases. For example, elementary statistics such as mean and standard deviation of R-R intervals or heart rates, over both short term period (5 minutes) and long term period (24 hours) have become a benchmark test for any heart rate time series (Malik et al., 1996). Other methods focus on spectral analysis of HRV, linking HRV to ANS controls in frequency domain

[1] HRV is a term for variations in both heart rates and R-R intervals (Malik et al., 1996).



(a) (b)

Figure 2.4: (a): An hour R-R interval data from record chf2db/chf201 in congestive heart failure R-R interval database in PhysioNet database. (b): An ANS control model for heart rate proposed by Akselrod et al. (1981) (figure also adapted from Akselrod et al. (1981)).

(Akselrod et al., 1981; Ivanov et al., 1996; Goldberger et al., 2002). These methods are employed to measure the efficiency of therapy in diagnosing diseases that affect nervous systems. Examples include Guillain-Barre syndrome, multiple sclerosis, Parkinson's diseases, cardiac sudden death and congestive heart failure studies (Ewing et al., 1984; Binkley et al., 1991; Freeman et al., 1991; Dougherty and Burr, 1992).

Similar to the case in neural spikes, traditional methods based on analysing the heart rate do not use instantaneous information. To overcome this problem, Barbieri et al. (2005) proposed a history-dependent inverse Gaussian (HDIG) point process model which treats heartbeats as a binary sequence. The model parameters are estimated with maximum likelihood in Barbieri et al. (2005). Later, the filter algorithm in Eden et al. (2004) is adapted to this model by Barbieri and Brown (2006), facilitating online estimation of the parameters.

2.2 State-space models

State-space models are a class of model frequently used for time series analysis. In this approach, the system of interest consists of two time-varying signals: An unobserved sequence of variables called states. An observed sequence of variables called observations. A state-space model is a specified relation between states and observations. General reference on this subject can be found in Bar-Shalom and E. (1987); Kitagawa and Gersch (1996); Durbin and Koopman (2001).

2.2.1 Linear dynamical systems

The simplest and perhaps the most widely used class of state-space model is the linear dynamical systems (LDS). In LDS, the state and observation at each time point k are described as

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k, \tag{2.1}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k, \tag{2.2}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the state transition matrix and $\mathbf{C} \in \mathbb{R}^{m \times n}$ is the observation matrix. $\mathbf{w}_k$ and $\mathbf{v}_k$ are Gaussian random variables from $\mathcal{N}(\mathbf{0}, \mathbf{Q})$ and $\mathcal{N}(\mathbf{0}, \mathbf{R})$, respectively.

There are countless applications of linear dynamical systems across the fields of control, signal processing, statistics and machine learning among others. For example, LDSs are the primary model in control engineering for characterizing systems. In the field of signal processing, a larger range of algorithms are concentrated on solving estimation problems in LDSs (Kailath et al., 2000; Haykin, 2002). In machine learning/statistics, the time index k often solely represents the index of data points. These data do not have to be time-varying. Consequently, as summarised by Roweis and Ghahramani (1999), many classical problems in machine learning can be formulated with LDSs, including the factor analysis (FA) model, principal component analysis (PCA) and independent component analysis (ICA) among others.

LDSs also found itself useful in computational biology and biomedical engineering as well; for example, modelling gene expression data (Beal et al., 2005; Sanguinetti et al., 2006), where the states denote the transcription factor (TF) activities and the observations are the microarray data. The state transition matrix $\mathbf{A}$ denotes the interactions between TFs. The observation matrix $\mathbf{C}$ presents the regulatory network between transcription factors and genes. These networks are difficult to measure in biological experiments. The LDSs, however, provides a framework to estimate them confidently.

Likewise, the electroencephalography (EEG) signals are often treated with LDSs (Sanei and Chambers, 2007; Cheung et al., 2010). In this case, the observations are the multichannel of EEG signals. The hidden states are some unmeasurable sources in the cortex. The parameters in this case can be used to characterise the cortical connectivities.

2.2.2 General state-space models

Nonlinear model with additive noise The immense complexity of the real world problems naturally motivates state-space models beyond the LDS setting. The first extension is the nonlinear state-space model:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_k \tag{2.3}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) + \mathbf{v}_k \tag{2.4}$$

where $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$. $\mathbf{w}_k$ and $\mathbf{v}_k$ are the same Gaussian random variables as the ones in the LDS. The most distinct feature of this nonlinear setting is that the relations between $\mathbf{y}_k$ and $\mathbf{x}_k$, $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ are characterised by two general functions, g and f. The LDSs can be easily fitted in this setting by choosing g and f to be linear.

Example 2.1. *A well studied nonlinear state-space model is the model considered in Kitagawa and Gersch (1996); Arulampalam et al. (2002):*

$$x_k = \frac{1}{2}x_{k-1} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos 1.2k + w_k, \tag{2.5}$$

$$y_k = \frac{x_n^2}{20} + v_k, \tag{2.6}$$

where $w_k \sim \mathcal{N}(0, \sigma_w^2)$ and $v_k \sim \mathcal{N}(0, \sigma_v^2)$. Put it in the language of nonlinear state-space model. Thus, we have

$$f(x_{k-1}) = \frac{1}{2}x_{k-1} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos 1.2k, \tag{2.7}$$

$$g(x_k) = \frac{x_n^2}{20}. \tag{2.8}$$

This model is also known as the nonlinear growth model.

Nonlinear model with multiplicative noise The nonlinear model can be further extended. For example, the Gaussian noise can be multiplicative rather than additive. The *stochastic volatility* (SV) model is an example of such a kind.

Example 2.2. *The SV model plays an important role in characterising financial time-series data, such as the log of the variance of returns on assets (Kim et al., 1998). A particularly successful application lies in modelling the daily exchange rate of Sterling/-Dollar, which motivates some major ideas on designing efficient Bayesian inference methods in general state-space models (Shephard and Pitt, 1997; Chen et al., 2001; Chib et al., 2002; Girolami and Calderhead, 2011).*

Specifically, the SV model consists of the following two equations:

$$x_k = \phi x_{k-1} + \eta_k, \tag{2.9}$$

$$y_k = \varepsilon_k \beta \exp(\frac{x_k}{2}) \tag{2.10}$$

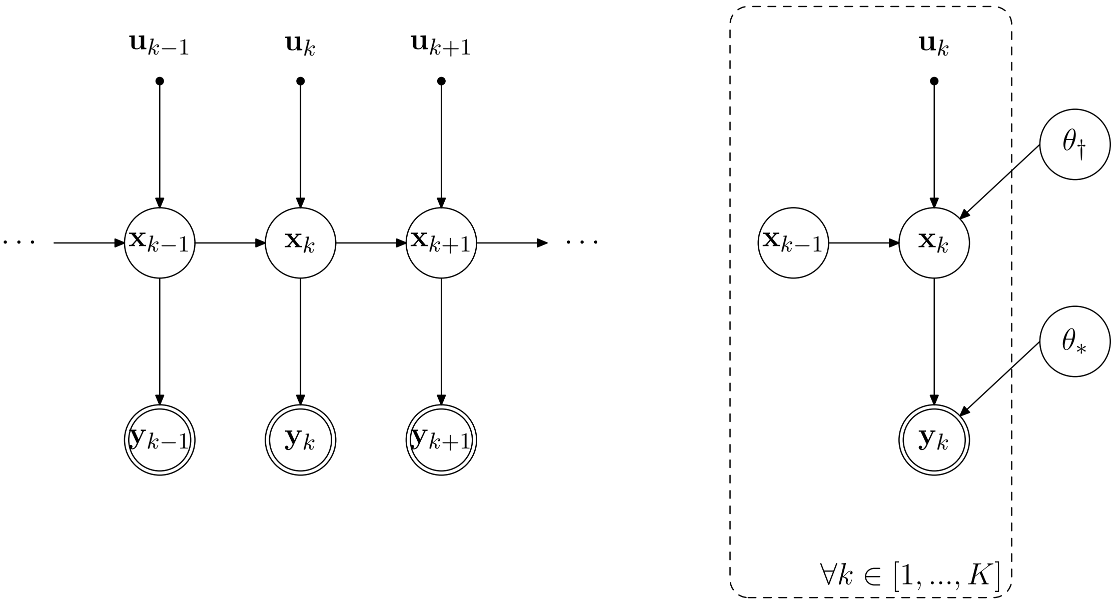


Figure 2.5: Graphical representations of general state-space models. *Left*, the traditional graphical representation of state-space models. *Right*, a compact graphical representation for state-space models including parameters. Both versions consider the presence of given inputs (denote as $\mathbf{u}_k$).

where $\eta_k \sim \mathcal{N}(0, \sigma^2)$ *and* $\varepsilon_k \sim \mathcal{N}(0, 1)$. *In the context of the nonlinear model formulation in the previous case,* $f(x_k) = \phi x_{k-1}$, $g(x_k) = \beta \exp(x_k)$. *The noise in the state model* η_k *works in the same way as in the previous case. The noise in the observation model,* ε_k, *however is multiplied to* $g(x_k)$.

A unified approach From a statistical perspective, both state and observation models can be described in terms of probability distributions (Geweke and Tanizaki, 2001). Specifically, assuming $\mathbf{x}_k \in \mathbb{R}^n, \forall k \in [0, \cdots, K]$, and $\mathbf{y}_k \in \mathbb{R}^m, \forall k \in [1, \cdots, K]$, we can write the state-space models as:

$$\mathbf{x}_0 \sim p(\mathbf{x}_0 | \boldsymbol{\theta}_\dagger), \tag{2.11}$$

$$\mathbf{x}_k \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}, \boldsymbol{\theta}_\dagger), \tag{2.12}$$

$$\mathbf{y}_k \sim p(\mathbf{y}_k | \mathbf{x}_k, \boldsymbol{\theta}_*), \tag{2.13}$$

where $\mathbf{x}_0$ is the initial state. Some state-space models use $\mathbf{x}_1$ as their initial state, which can be easily adapted to the above framework. $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \boldsymbol{\theta}_\dagger)$ is called the state transition distribution. $p(\mathbf{y}_k | \mathbf{x}_k, \boldsymbol{\theta}_*)$ is known as the observation distribution. Any state-space model can be written in term of these three probability distributions. The relation between the variables in these distributions is often described graphically, such as the two versions in figure 2.5 Sometimes, state-space models are considered with external inputs. More specifically, the inputs which denoted as $\mathbf{u}_k$, are involved in the state transition distribution; that is, $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)$.

In the following examples, we convert the formulations of the three models discussed above into the unified approach.

Example 2.3. *LDS in the form of probability distributions:*

$$p(\mathbf{x}_0|\boldsymbol{\theta}_\dagger) = \mathcal{N}(\mathbf{x}_0|\boldsymbol{\pi}_0, \boldsymbol{\Sigma}_0), \tag{2.14}$$

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}_\dagger) = \mathcal{N}(\mathbf{x}_k|\mathbf{A}\mathbf{x}_{k-1}, \mathbf{Q}), \tag{2.15}$$

$$p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\theta}_*) = \mathcal{N}(\mathbf{y}_k|\mathbf{C}\mathbf{x}_k, \mathbf{R}), \tag{2.16}$$

where $\boldsymbol{\theta}_\dagger = \{\boldsymbol{\pi}_0, \boldsymbol{\Sigma}_0, \mathbf{A}, \mathbf{Q}\}$*,* $\boldsymbol{\theta}_* = \{\mathbf{C}, \mathbf{R}\}$*.*

Example 2.4. *The nonlinear growth model in the form of probability distributions:*

$$p(x_0|\boldsymbol{\theta}_\dagger) = \mathcal{N}(x_0|\pi_0, \sigma_0^2), \tag{2.17}$$

$$p(x_k|x_{k-1}, \boldsymbol{\theta}_\dagger) = \mathcal{N}(x_k|ax_{k-1} + \frac{bx_{k-1}}{1 + x_{k-1}^2} + c\cos 1.2k, \sigma_w^2), \tag{2.18}$$

$$p(y_k|x_k, \boldsymbol{\theta}_*) = \mathcal{N}(y_k|\frac{x_n^2}{20}, \sigma_v^2), \tag{2.19}$$

where $\boldsymbol{\theta}_\dagger = \{\pi_0, \sigma_0^2, a, b, c, \sigma_w^2\}$*,* $\boldsymbol{\theta}_* = \sigma_v^2$*.*

Example 2.5. *The SV model in the form of probability distributions:*

$$p(x_0|\boldsymbol{\theta}_\dagger) = \mathcal{N}(x_0|0, \frac{\sigma^2}{1 - \phi^2}), \tag{2.20}$$

$$p(x_k|x_{k-1}, \boldsymbol{\theta}_\dagger) = \mathcal{N}(x_k|\phi x_{k-1}, \sigma^2), \tag{2.21}$$

$$p(y_k|x_k, \boldsymbol{\theta}_*) = \mathcal{N}(y_k|0, \beta^2 \exp(x_k)). \tag{2.22}$$

where $\boldsymbol{\theta}_\dagger = \{\phi, \sigma^2\}$*,* $\boldsymbol{\theta}_* = \beta$*.*

In practice, apart from the observations $\{\mathbf{y}_k\}$, both $\{\mathbf{x}_k\}$ and $\boldsymbol{\theta} = (\boldsymbol{\theta}_\dagger, \boldsymbol{\theta}_*)$ are unknown. In the following, we review a range of estimation algorithms of states and parameters under the unified approach formulation.

2.3 Inference in state-space models: Filtering

The state estimation problem, statistically, can be seen as an *inference* problem, in which the target is a posterior distribution of states given observations. Mathematically, such a distribution can be written as

$$p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}) = \frac{p(\mathbf{y}_{1:K}|\mathbf{x}_{1:K})p(\mathbf{x}_{0:K})}{p(\mathbf{y}_{1:K})} \tag{2.23}$$

where $\mathbf{y}_{1:K} = \{\mathbf{y}_k\}_{k=1}^{K}$, $\mathbf{x}_{1:K} = \{\mathbf{x}_k\}_{k=0}^{K}$

$$p(\mathbf{y}_{1:K}|\mathbf{x}_{1:K}) = \prod_{k=1}^{K} p(\mathbf{y}_k|\mathbf{x}_k), \quad p(\mathbf{x}_{0:K}) = p(\mathbf{x}_0) \prod_{k=1}^{K} p(\mathbf{x}_k|\mathbf{x}_{k-1}). \tag{2.24}$$

Remark 2.1. *During the state estimation process, all the parameters are given.*

The inference problem in state-space models is often split into two parts, namely *filtering* and *smoothing* (Kitagawa and Gersch, 1996).

2.3.1 Kalman filter

The most famous and perhaps the most widely used filter algorithm is the *Kalman filter* (or Kalman-Bucy filter in continuous time models) (Kalman, 1960; Kalman and Bucy, 1961) for LDS.

Algorithm 2.1. The Kalman filter (Kalman, 1960; Kalman and Bucy, 1961).

```
Input: $\mathbf{x}_{0|0}, \mathbf{\Sigma}_{0|0}, \{\mathbf{y}_k\}, \boldsymbol{\theta}$
Output: $\{\mathbf{x}_{k|k}\}, \{\mathbf{\Sigma}_{k|k}\}$
for $k = 1$ to $K$ do
    $\mathbf{x}_{k|k-1} = \mathbf{A}\mathbf{x}_{k-1|k-1}$
    $\mathbf{\Sigma}_{k|k-1} = \mathbf{A}\Sigma_{k-1|k-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}$
    $\mathbf{K}_k = \mathbf{\Sigma}_{k|k-1}\mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\mathbf{\Sigma}_{k|k-1}\mathbf{C}^{\mathrm{T}} + \mathbf{R}\right)^{-1}$
    $\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k(\mathbf{y} - \mathbf{C}\mathbf{x}_{k|k-1})$
    $\mathbf{\Sigma}_{k|k} = \mathbf{\Sigma}_{k|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{\Sigma}_{k|k-1}$
end for
```

$\mathbf{x}_{k|k}$ and $\mathbf{\Sigma}_{k|k}$ are the estimated state and its covariance at time k. $\mathbf{x}_{k|k-1}$ and $\mathbf{\Sigma}_{k|k-1}$ are called the prediction and prediction covariance.

2.3.2 Extended Kalman filter

The original Kalman filter formulation is incapable of solving the state estimation problem in the nonlinear state-space models (e.g. equations (2.3)-(2.4). A modified filter for such models is the extended Kalman filter (EKF), which linearises the nonlinear functions. More precisely, it takes the Taylor expansion of $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ around $\mathbf{x}_{k-1|k-1}$ and $\mathbf{x}_{k|k-1}$ respectively.

$$\mathbf{f}(\mathbf{x}_{k-1}) \simeq \mathbf{f}(\mathbf{x}_{k-1|k-1}) + \mathbf{F}_k(\mathbf{x}_{k-1} - \mathbf{x}_{k-1|k-1}) + \text{high order terms}, \tag{2.25}$$

$$\mathbf{g}(\mathbf{x}_k) \simeq \mathbf{g}(\mathbf{x}_{k|k-1}) + \mathbf{G}_k(\mathbf{x}_k - \mathbf{x}_{k|k-1}) + \text{high order terms}, \tag{2.26}$$

where

$$\mathbf{F}_k = \left.\frac{\partial \mathbf{f}(\mathbf{x}_{k-1})}{\partial \mathbf{x}_{k-1}}\right|_{\mathbf{x}_{k-1}=\mathbf{x}_{k-1|k-1}}, \quad \mathbf{G}_k = \left.\frac{\partial \mathbf{g}(\mathbf{x}_k)}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k=\mathbf{x}_{k|k-1}}. \tag{2.27}$$

Substituting equations (2.25) and (2.26) into equations (2.3) and (2.4), respectively, and ignoring the high-order terms, we have a LDS formulation:

$$\mathbf{x}_k = \mathbf{F}_k\mathbf{x}_{k-1} + \mathbf{d}_k + \mathbf{w}_k, \tag{2.28}$$

$$\mathbf{y}_k = \mathbf{G}_k\mathbf{x}_k + \mathbf{e}_k + \mathbf{v}_k, \tag{2.29}$$

where

$$\mathbf{d}_k = \mathbf{f}(\mathbf{x}_{k-1|k-1}) - \mathbf{F}_k\mathbf{x}_{k-1|k-1}, \quad \mathbf{e}_k = \mathbf{g}(\mathbf{x}_{k|k-1}) - \mathbf{G}_k\mathbf{x}_{k|k-1}. \tag{2.30}$$

Consider the $\mathbf{d}_k$ and $\mathbf{e}_k$ as given input terms at time k, one can write a Kalman filter recursion for the newly formed LDS as the following.

Algorithm 2.2. The extended Kalman filter (Anderson and Moore, 1979; Haykin, 2002).

```
Input: $\mathbf{x}_{0|0}, \mathbf{\Sigma}_{0|0}, \{\mathbf{y}_k\}, \boldsymbol{\theta}$
Output: $\{\mathbf{x}_{k|k}\}, \{\mathbf{\Sigma}_{k|k}\}$
for $k = 1$ to $K$ do
    $\mathbf{x}_{k|k-1} = \mathbf{f}(\mathbf{x}_{k-1|k-1})$
    $\mathbf{\Sigma}_{k|k-1} = \mathbf{F}_k\Sigma_{k-1|k-1}\mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}$
    $\mathbf{K}_k = \mathbf{\Sigma}_{k|k-1}\mathbf{G}_k^{\mathrm{T}}\left(\mathbf{G}_k\mathbf{\Sigma}_{k|k-1}\mathbf{G}_k^{\mathrm{T}} + \mathbf{R}\right)^{-1}$
    $\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k(\mathbf{y} - \mathbf{g}(\mathbf{x}_{k|k-1}))$
    $\mathbf{\Sigma}_{k|k} = \mathbf{\Sigma}_{k|k-1} - \mathbf{K}_k\mathbf{G}_k\mathbf{\Sigma}_{k|k-1}$
end for
```

2.3.3 Bayesian optimal filtering

Using the unified approach to general state-space models, one can formulate the filtering problem from a Bayesian perspective. That is, transferring the filter problem from obtaining point estimates to inferring the posterior distribution over $\mathbf{x}_k$ based on the observations up to time k. Specifically,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{k-1})}, \tag{2.31}$$

where

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int_{\mathbf{x}_{k-1}} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1}, \tag{2.32}$$

is known as the *Chapman-Kolmogorov* equation. $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ can be seen as the prediction distribution of $\mathbf{x}_k$. The denominator,

$$p(\mathbf{y}_k|\mathbf{y}_{k-1}) = \int_{\mathbf{x}_k} p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k, \tag{2.33}$$

serves as a normalising constant that ensures $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ being a proper probability density function. The product of the denominator over all time points forms the marginal likelihood, that is,

$$p(\mathbf{y}_{1:K}) = p(\mathbf{y}_1)\prod_{k=1}^{K} p(\mathbf{y}_k|\mathbf{y}_{k-1}). \tag{2.34}$$

As we shall see later in this chapter, this term is the objective (or score) function for estimating the parameters $\boldsymbol{\theta}$.

Interestingly, as noticed by many authors (Kitagawa and Gersch, 1996; Minka, 1999; Bishop, 2006), both Kalman filter and EKF can be derived from the Bayesian optimal filtering framework. Firstly, let us see the Kalman filter derivation.

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int_{\mathbf{x}_{k-1}} \mathcal{N}(\mathbf{x}_k|\mathbf{A}\mathbf{x}_{k-1},\mathbf{Q})\mathcal{N}(\mathbf{x}_{k-1}|\mathbf{x}_{k-1|k-1},\boldsymbol{\Sigma}_{k-1|k-1})d\mathbf{x}_{k-1} \tag{2.35}$$

$$= \mathcal{N}(\mathbf{x}_k|\mathbf{x}_{k|k-1},\boldsymbol{\Sigma}_{k|k-1}) \tag{2.36}$$

where, using the Gaussian conditioning rule,

$$\mathbf{x}_{k|k-1} = \mathbf{A}\mathbf{x}_{k-1}, \quad \boldsymbol{\Sigma}_{k|k-1} = \mathbf{A}\boldsymbol{\Sigma}_{k-1|k-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}. \tag{2.37}$$

Expanding equation (2.31) in the logarithm domain, we have

$$\begin{aligned}
\ln p(\mathbf{x}_k|\mathbf{y}_{1:k}) &= \ln p(\mathbf{y}_k|\mathbf{x}_k) + \ln p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) - \ln p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) \\
&= -\frac{1}{2}(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k) - \frac{1}{2}(\mathbf{x}_k - \mathbf{x}_{k|k-1})^{\mathrm{T}}\boldsymbol{\Sigma}_{k|k-1}^{-1}(\mathbf{x}_k - \mathbf{x}_{k|k-1}) \\
&\quad + \text{const} \\
&= -\frac{1}{2}\mathbf{x}_k^{\mathrm{T}}\left(\mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{C} + \boldsymbol{\Sigma}_{k|k-1}^{-1}\right)\mathbf{x}_k + \mathbf{x}_k^{\mathrm{T}}\left(\mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}_k + \boldsymbol{\Sigma}_{k|k-1}^{-1}\mathbf{x}_{k|k-1}\right) \\
&\quad + \text{const}
\end{aligned} \tag{2.38}$$

The "const" denotes terms that are independent from $\mathbf{x}_k$. Using the technique of completing the square, one can rewrite $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ as a Gaussian distribution $\mathcal{N}(\mathbf{x}_k|\mathbf{x}_{k|k},\boldsymbol{\Sigma}_{k|k})$, where, based on the first term in equation (2.38) and the matrix inversion lemma, the covariance is written as

$$\begin{aligned}\mathbf{\Sigma}_{k|k} &= \left(\mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{C} + \mathbf{\Sigma}_{k|k-1}^{-1}\right)^{-1}, \\ &= \mathbf{\Sigma}_{k|k-1} + \mathbf{K}_k\mathbf{C}\mathbf{\Sigma}_{k|k-1}.\end{aligned} \tag{2.39}$$

$\mathbf{K}_k$ is commonly known as the *Kalman gain*, taking the form of

$$\mathbf{K}_k = \mathbf{\Sigma}_{k|k-1}\mathbf{C}^{\mathrm{T}}\left(\mathbf{R} + \mathbf{C}\mathbf{\Sigma}_{k|k-1}\mathbf{C}^{\mathrm{T}}\right)^{-1}. \tag{2.40}$$

The mean of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, $\mathbf{x}_{k|k}$ can be deduced from the second term in equation (2.38); more precisely,

$$\mathbf{x}_{k|k} = \mathbf{\Sigma}_{k|k}\left(\mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}_k + \mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{x}_{k|k-1}\right). \tag{2.41}$$

Substituting $\mathbf{\Sigma}_{k|k}$ with equation (2.39), we have

$$\begin{aligned}\mathbf{x}_{k|k} &= \left(\mathbf{\Sigma}_{k|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{\Sigma}_{k|k-1}\right)\mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}_k + \left(\mathbf{\Sigma}_{k|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{\Sigma}_{k|k-1}\right)\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{x}_{k|k-1} \\ &= \mathbf{\Sigma}_{k|k-1}\mathbf{C}^{\mathrm{T}}\left(\mathbf{I} - \left(\mathbf{R} + \mathbf{C}\mathbf{\Sigma}_{k|k-1}\mathbf{C}^{\mathrm{T}}\right)^{-1}\mathbf{C}\mathbf{\Sigma}_{k|k-1}\mathbf{C}^{\mathrm{T}}\right)\mathbf{R}^{-1}\mathbf{y}_k + \left(\mathbf{I} - \mathbf{K}_k\mathbf{C}\right)\mathbf{x}_{k|k-1},\end{aligned} \tag{2.42}$$

where $\mathbf{I}$ is the identity matrix. Using the matrix identity

$$\left(\mathbf{I} - (\mathbf{A} + \mathbf{B})^{-1}\mathbf{A}\right)\mathbf{B}^{-1} = (\mathbf{A} + \mathbf{B})^{-1}. \tag{2.43}$$

$\mathbf{x}_{k|k}$ finally becomes

$$\begin{aligned}\mathbf{x}_{k|k} &= \mathbf{\Sigma}_{k|k-1}\mathbf{C}^{\mathrm{T}}\left(\mathbf{R} + \mathbf{C}\mathbf{\Sigma}_{k|k-1}\mathbf{C}^{\mathrm{T}}\right)^{-1}\mathbf{y}_k + (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{x}_{k|k-1} \\ &= \mathbf{K}_k\mathbf{y}_k + (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{x}_{k|k-1} \\ &= \mathbf{x}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}\mathbf{x}_{k|k-1}).\end{aligned} \tag{2.44}$$

This completes the derivation of the Kalman filter in algorithm 2.1.

Since a EKF transfers a nonlinear system into a LDS, the derivation of EKF follows the same with Kalman filter.

2.3.4 Laplace Gaussian filter

Let us now apply EKF to the SV model. After linearisation, the observation equation becomes

$$y_k = (g_k x_k + e_k)\varepsilon_k, \tag{2.45}$$

where $g_k = \frac{1}{2}\beta\exp(\frac{x_{k|k-1}}{2})$, $e_k = \beta\exp(\frac{x_{k|k-1}}{2}) - g_k x_{k|k-1}$. This is still not of the standard LDS formulation; therefore EKF is not suitable for general models beyond additive Gaussian noise.

Start from the unified approach, Laplace Gaussian filter (Koyama et al., 2010) provides more general method to construct Gaussian approximation to $p(\mathbf{x}_k|\mathbf{y}_{1:k})$. Precisely, take the Taylor expansion around the maximum of $\ln p(\mathbf{x}_k|\mathbf{y}_{1:k})$.

$$\ln p(\mathbf{x}_k|\mathbf{y}_{1:k}) \simeq \ln p(\mathbf{x}_{k|k}|\mathbf{y}_{1:k}) + \mathbf{g}^{\mathrm{T}}(\mathbf{x}_{k|k})(\mathbf{x}_k - \mathbf{x}_{k|k}) \tag{2.46}$$

$$+ (\mathbf{x}_k - \mathbf{x}_{k|k})^{\mathrm{T}}\mathbf{H}(\mathbf{x}_{k|k})(\mathbf{x}_k - \mathbf{x}_{k|k}) + \text{high order terms} \tag{2.47}$$

where

$$\mathbf{g}(\mathbf{x}_{k|k}) = \left.\frac{\partial \ln p(\mathbf{x}_k|\mathbf{y}_{1:k})}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k=\mathbf{x}_{k|k}}, \quad \mathbf{H}(\mathbf{x}_{k|k}) = \left.\frac{\partial^2 \ln p(\mathbf{x}_k|\mathbf{y}_{1:k})}{\partial \mathbf{x}_k \partial \mathbf{x}_k^{\mathrm{T}}}\right|_{\mathbf{x}_k=\mathbf{x}_{k|k}}. \tag{2.48}$$

Further, as $x_{k|k}$ corresponds to the peak of $\ln p(\mathbf{x}_k|\mathbf{y}_{1:k})$, the first-order term in the Taylor expansion vanishes. Ignoring the high-order terms, the remaining second-order forms a Gaussian density function. As such, we obtained a Gaussian approximation of the filtering distribution, $p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \mathcal{N}(\mathbf{x}_k|\mathbf{x}_{k|k}, \mathbf{\Sigma}_{k|k})$, where, $\mathbf{\Sigma}_{k|k} = \mathbf{H}^{-1}(\mathbf{x}_{k|k})$. The outline of LGF is described in the following pseudo codes:

Algorithm 2.3. The Laplace Gaussian filter

Input: $\mathbf{x}_{0|0}$, $\mathbf{\Sigma}_{0|0}$ and $\boldsymbol{\theta}$

Output: $\{\mathbf{x}_{k|k}\}$ and $\{\mathbf{\Sigma}_{k|k}\}$

```
for $k = 1$ to $K$ do
    $\mathbf{x}_{k|k} = \arg\max_{\mathbf{x}_k} \ln p(\mathbf{x}_k|\mathbf{y}_{1:k})$
    $\mathbf{\Sigma}_{k|k} = \mathbf{H}^{-1}(\mathbf{x}_{k|k})$
end for
```

Many authors used the similar idea for filtering in the context of neuroscience (Brown et al., 1998, 2001; Smith and Brown, 2003; Truccolo et al., 2005; Czanner et al., May 2008).

2.3.5 Particle filter (SMC)

The *particle filter* or *sequential Monte Carlo* (SMC) approaches (Liu and Chen, 1998; Doucet et al., 2000b, 2001; Arulampalam et al., 2002) take a different view on the Bayesian optimal filtering problem. It uses the *Monte Carlo approach* to obtain sample based representation of the filtering distribution rather than the functional form such as the Kalman filter, or approximate functional form such as EKF and LGF. Precisely, the sample-based representation writes as

$$P_M(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{1}{M}\sum_{i=1}^{M} \delta\left(\mathbf{x}_k - \mathbf{x}_k^{(i)}\right), \tag{2.49}$$

where $\delta(\cdot)$ is the Dirac delta function. $x_k^{(i)} \sim p(\mathbf{x}_k|\mathbf{y}_{1:k})$. Often, this distribution is involved in computing expectation over some function of interest, $h(\mathbf{x}_k)$; that is

$$I(h) = \int_{\mathbf{x}_k} h(\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k})d\mathbf{x}_k. \tag{2.50}$$

Using the sample representation of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, $I(h)$ can be estimated as

$$\hat{I}_M(h) = \frac{1}{M}\sum_{i=1}^{M} h\left(\mathbf{x}_k^{(i)}\right). \tag{2.51}$$

Following the strong law of large numbers (LLN), this estimation has an appealing accuracy, which states as,

$$\hat{I}_M(h) \xrightarrow{a.s.} I(h), \quad \text{as } M \to +\infty, \tag{2.52}$$

where $\xrightarrow{a.s.}$ denotes almost sure convergence. Further, Let $h(\mathbf{x}_k)$ has a finite second moment,

$$\mathbb{E}_{p(\mathbf{x}_k|\mathbf{y}_{1:k})}[h^2(\mathbf{x}_k)] < +\infty. \tag{2.53}$$

Then, a central limit theorem (CLT), also known as the Monte Carlo CLT, holds

$$\sqrt{M}\left(I_M(h) - I(h)\right) \xrightarrow{d} \mathcal{N}(0, \sigma_h^2), \tag{2.54}$$

where $\xrightarrow{d}$ denotes convergence in distribution. σ_h^2 is the variance of $h(\mathbf{x}_k)$, more precisely,

$$\sigma_h^2 = \mathbb{E}_{p(\mathbf{x}_k|\mathbf{y}_{1:k})}[h^2(\mathbf{x}_k)] - h^2(\mathbf{x}_k). \tag{2.55}$$

The most appealing property of the Monte Carlo CLT is that, the convergence rate is independent from the dimensionality of $\mathbf{x}_k$. Oppositely, other methods (such as grid-based methods) often suffers slow convergence as the dimension of $\mathbf{x}_k$ increase, which is known as the *curse of dimensionality* (Bishop, 2006).

The Monte Carlo CLT can be generalized to the joint posterior distribution $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, which gives rise to the *sequential importance sampling* (SIS) approach (Doucet et al., 2001; Arulampalam et al., 2002), and eventually facilitates the particle filter.

The SIS approach belongs to a larger family of methods known as the *importance sampling* (Liu, 2001; Robert and Casella, 2004). The Monte Carlo CLT relays on the fact that one is able to obtain samples from $p(\mathbf{x}_k|\mathbf{y}_{1:K})$ or $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$. In practice, such sampling is often infeasible. As such, IS is a means to perform the Monte Carlo estimation in the absent of direct samples from the target distribution.

Suppose, instead of direct sampling from the target distribution, one can obtain samples from an alternative distribution $q(\mathbf{x}_k)$[2] (or $q(\mathbf{x}_{0:k})$, when $\mathbf{x}_{0:k}$ is of interest). To a larger extend, $q(\mathbf{x}_k)$ is often called as *proposal distribution.* Rewrite $I_M(h)$ as,

$$I_M(h) = \int_{\mathbf{x}_k} h(\mathbf{x}_k)\frac{p(\mathbf{x}_k|\mathbf{y}_{1:k})q(\mathbf{x}_k)}{q(\mathbf{x}_k)}d\mathbf{x}_k. \tag{2.56}$$

Then, the ratio between $p(\cdot)$ and $q(\cdot)$ is defined as the importance weight,

$$w(\mathbf{x}_k) = \frac{p(\mathbf{x}_k|\mathbf{y}_{1:k})}{q(\mathbf{x}_k)}. \tag{2.57}$$

Consequently, the Monte Carlo estimation becomes

$$\hat{I}_M(h) = \frac{1}{M}\sum_{i=1}^{M} h\left(\mathbf{x}_k^{(i)}\right) w\left(\mathbf{x}_k^{(i)}\right), \quad \mathbf{x}_k^{(i)} \sim q(\mathbf{x}_k), \tag{2.58}$$

which expresses the estimation as the weighted sum of $h(\mathbf{x}_k)$. When $\mathbf{x}_{0:k}$ is of interest, the IS is

$$\hat{I}_M(h) = \frac{1}{M}\sum_{i=1}^{M} h\left(\mathbf{x}_{0:k}^{(i)}\right) w\left(\mathbf{x}_{0:k}^{(i)}\right), \quad \mathbf{x}_{0:k}^{(i)} \sim q(\mathbf{x}_{0:k}). \tag{2.59}$$

Under weak assumptions, we have the same the strong LLN for $\hat{I}_M(h) \xrightarrow{a.s.} I(h)$. Further, the Monte Carlo CLT holds still, providing some additional assumptions (Geweke, 1989). As a result, the convergence rate of $\hat{I}_M$ still only scales with the number of samples.

The SIS method considers the scenario which targets on $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$. In this case, the importance weight writes as

$$w_k(\mathbf{x}_{0:k}) = \frac{p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})}{q(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})}, \tag{2.60}$$

where the proposal distribution is assumed to be data dependent.

To facilitate a sampling procedure, the SIS further restricts the proposal distribution to be factorised as

$$q(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = q(\mathbf{x}_0)\prod_{n=1}^{k} q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{1:n}). \tag{2.61}$$

[2] $q(\mathbf{x}_k)$ can be any distribution. Particularly, $q(\mathbf{x}_k|\mathbf{y}_{1:k})$ is popular given its appealing data-dependent nature.

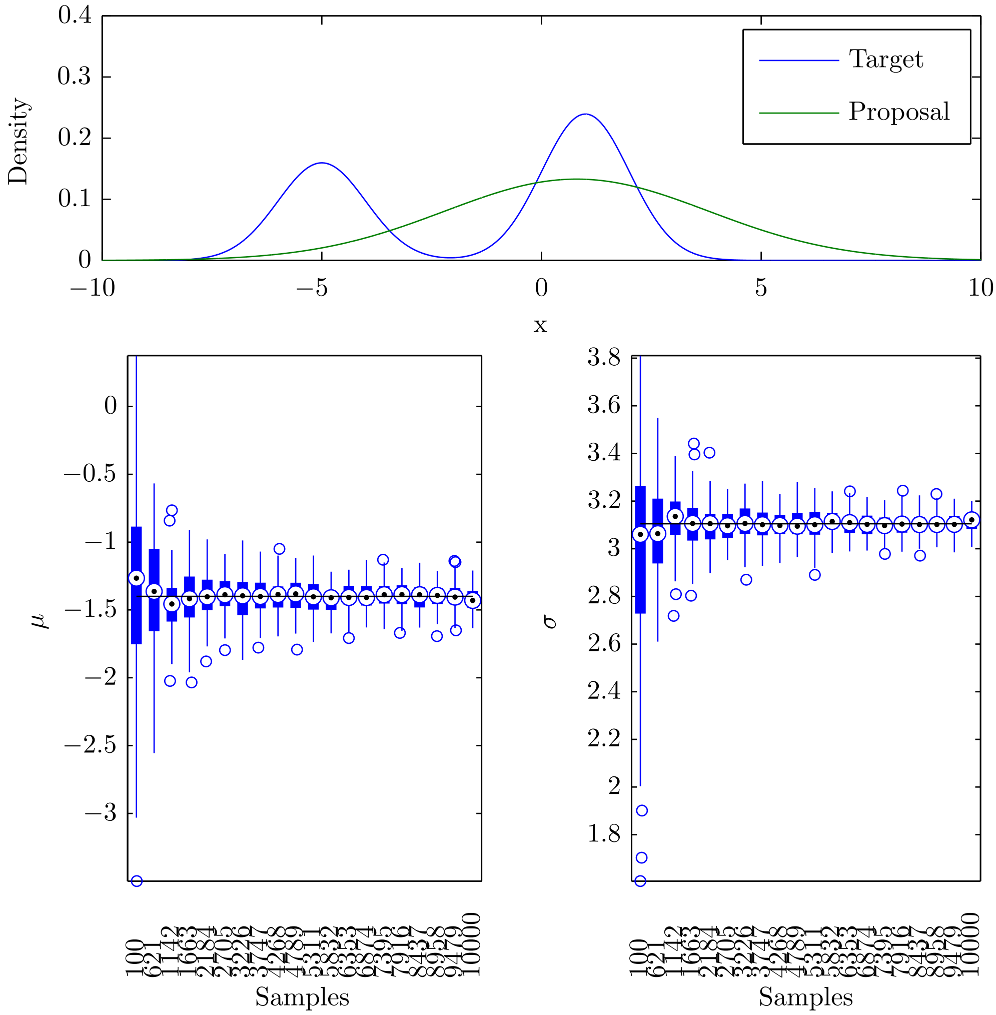


Figure 2.6: Illustration of the importance sampling

Such a factorisation allows samples to be proposed in a recursive fashion. Particularly, at time k, we have

$$q(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = q(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}). \tag{2.62}$$

Substitute equation (2.62) into (2.60)

$$w_k(\mathbf{x}_{0:k}) = \frac{p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})}{q(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k})}. \tag{2.63}$$

With a little additional manipulations, we obtain a recursive formula for updating the weights; that is

$$\begin{aligned} w_k(\mathbf{x}_{0:k}) &= \frac{p\left(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}\right)}{q(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})} \frac{p(\mathbf{y}_k|\mathbf{x}_{0:k}, \mathbf{y}_{1:k-1})p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k})} \\ &\propto w_{k-1}(\mathbf{x}_{0:k-1})\frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})}{q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k})} \end{aligned} \tag{2.64}$$

The SIS method results in the generic particle filter (Arulampalam et al., 2002), in which the proposal is further restricted to $q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k)$. Consequently, the generic particle filter is shown in the following pseudo-code:

Algorithm 2.4. The generic particle filter (Arulampalam et al., 2002)
Input: $\mathbf{x}_0^{(1:M)}$, $\boldsymbol{\theta}$
Output: $\mathbf{x}_{1:K}^{(1:M)}$

```
for $k = 1$ to $K$ do
    Draw $\mathbf{x}_k^{(1:M)} \sim q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k)$
    Compute weights as
        $$w_k^{(m)} = w_{k-1}^{(m)} \frac{p\left(\mathbf{y}_k \middle| \mathbf{x}_k^{(m)}\right) p\left(\mathbf{x}_k^{(m)} \middle| \mathbf{x}_{k-1}^{(m)}\right)}{q\left(\mathbf{x}_k^{(m)} \middle| \mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k\right)}, \quad \forall m \in [1, \cdots, M] \tag{2.65}$$
    Compute the normalised weights
        $$W_k^{(m)} = \frac{w_k^{(m)}}{\sum_{m=1}^{M} w_k^{(m)}}, \quad \forall m \in [1, \cdots, M]. \tag{2.66}$$
    Resample $\mathbf{x}_k^{*(1:M)}$ according to the normalised weights $W_k^{(1:M)}$, and let $\mathbf{x}_k^{(1:M)} = \mathbf{x}_k^{*(1:M)}$.
end for
```

The resampling procedure is introduced to fight a pathology in the SIS method. The distribution of weights becomes increasingly skewed over time. This means only a small portion of the particles appears to be important in the empirical distribution. This problem is known as the *degeneracy* problem (Doucet et al., 2001; Arulampalam et al., 2002). Note that, after resampling for each time point, the weights become uniformly distributed, or $w_k^{(m)} = 1/M$ (Gordon et al., 1993). Hence, equation (2.65) can be reduced to

$$w_k^{(m)} = \frac{p\left(\mathbf{y}_k \middle| \mathbf{x}_k^{(m)}\right) p\left(\mathbf{x}_k^{(m)} \middle| \mathbf{x}_{k-1}^{(m)}\right)}{q\left(\mathbf{x}_k^{(m)} \middle| \mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k\right)}. \tag{2.67}$$

Many variations have been derived based on the generic particle filter framework in algorithm 2.4. The simplest and perhaps the most popular adaptation of the generic framework is the *sampling importance resampling* (SIR), or the *bootstrap* filter (Gordon et al., 1993). Specifically, the SIR filter adopts the state transition distribution to be the proposal; that is

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1}). \tag{2.68}$$

Substituting equation (2.68) into (2.67), we have a new weight updating rule, in which the weight is solely determined by the observation model,

$$w_k^{(m)} = p\left(\mathbf{y}_k \middle| \mathbf{x}_k^{(m)}\right). \tag{2.69}$$

The resampling procedure on its own cannot completely solve the degeneracy problem. In fact, sometimes it also gives rise to problems similar to the degeneracy problem. Specifically, the same particle could get resampled too many times. Consequently, the particle population loses its diversity (Arulampalam et al., 2002). In this regard, more sophisticated algorithms are proposed to continue the battle with degeneracy. These include the resample-move algorithm proposed by Gilks and Berzuini (2001) and the auxiliary particle filter due to Pitt and Shephard (1999). At the other end of the spectrum, authors employ the EKF and Unscented Kalman filter (UKF) and LGF to construct proposal distribution, which also significantly improves the efficiency of particle filter.

2.4 Inference in state-space models: Smoothing

The smoothing problem is the quest of $p(\mathbf{x}_k|\mathbf{y}_{1:K})$[3] (Kitagawa and Gersch, 1996). Under the unified representation of state-space models, such a distribution can be obtained by

$$p(\mathbf{x}_k|\mathbf{y}_{1:K}) = p(\mathbf{x}_k|\mathbf{y}_{1:k}) \int_{\mathbf{x}_{k+1}} \frac{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:K})p(\mathbf{x}_{k+1}|\mathbf{x}_k)}{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})} d\mathbf{x}_{k+1}. \tag{2.70}$$

Equation (2.70) forms a backward recursion, as $p(\mathbf{x}_k|\mathbf{y}_{1:K})$ is obtained by manipulating $p(\mathbf{x}_{k+1}|\mathbf{y}_{1:K})$. In the context of LDS, such smoothing distributions are Gaussians, which motivate the *Rauch-Tung-Striebel* (RTS) smoother[4] (Rauch et al., 1965), shown in algorithm 2.5.

Algorithm 2.5. The RTS smoother

Input: $x_{K|K}$, $\mathbf{\Sigma}_{K|K}$

Output: $\{x_{k|K}\}$, $\{\mathbf{\Sigma}_{k|K}\}$

```
for $k = K - 1$ to 1 do
    $\mathbf{J}_k = \mathbf{A}^{\mathrm{T}}\mathbf{\Sigma}_{k|k}\mathbf{\Sigma}_{k+1|k}^{-1}$
    $\mathbf{x}_{k|K} = \mathbf{x}_{k|k} + \mathbf{J}_k(\mathbf{x}_{k+1|K} - \mathbf{x}_{k+1|k})$
    $\mathbf{\Sigma}_{k|K} = \mathbf{\Sigma}_{k|k} + \mathbf{J}_k\left(\mathbf{\Sigma}_{k+1|K} - \mathbf{\Sigma}_{k+1|k}\right)\mathbf{J}_k^{\mathrm{T}}$
end for
```

[3] $p(\mathbf{x}_k|\mathbf{y}_{1:K})$ belongs to a common smoothing paradigm known as the *fixed interval smoothing*. There are another two types of smoothing techniques; namely *fixed point smoothing* (Anderson and Moore, 1979) and *fixed lag smoothing* (Clapp and Godsill, 1999).

[4] The RTS smoother is normally used following a Kalman filter; therefore, it is sometime called the Kalman smoother or RTS smoother (including a Kalman filter).

Derivation of the RTS smoother based on equation (2.70) can be found in Kitagawa and Gersch (1996); Minka (1999).

Interestingly, according to equation (2.70), the observation distribution $p(\mathbf{y}_k|\mathbf{x}_k)$ plays no part in the computation of $p(\mathbf{x}_k|\mathbf{y}_{1:K})$. This naturally gives us the following insight.

Remark 2.2. *For state-space models with Gaussian state transition distribution, if the filtering distribution is assumed to be Gaussian, then the RTS smoother gives the optimal solution to equation (2.70), regardless of the observation model.*

This remark justifies the method of choice in Smith and Brown (2003), which is discussed in a later chapter. For more asymptotically exact solutions, we have to resort to the Monte Carlo approaches, or some modified SMC methods for smoothing purposes, e.g. see Clapp and Godsill (1999); Doucet et al. (2000b); Kitagawa (1987, 1996).

2.5 Learning in state-space models: EM

In some cases, model parameters are also not available. Estimating these unknown parameters in state-space models, is named as *learning* by Roweis and Ghahramani (1999). In control, this is also known as the *system identification* problem (Ljung, 1999).

From a statistical perspective, these unknown parameters can be estimated by MLE; that is,

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \ \ln p(\mathbf{y}_{1:K}|\boldsymbol{\theta}), \tag{2.71}$$

where $p(\mathbf{y}_{1:K}|\boldsymbol{\theta})$, the *marginal likelihood* writes

$$p(\mathbf{y}_{1:K}|\boldsymbol{\theta}) = \int_{\mathbf{x}_{0:K}} p(\mathbf{y}_{1:K}, \mathbf{x}_{0:K}|\boldsymbol{\theta}) d\mathbf{x}_{0:K}. \tag{2.72}$$

Equation (2.72) is a computationally expensive (or quite often intractable) integral for many models. Consequently, directly maximise the log-marginal likelihood could be problematic.

A significant milestone in solving problems of this kind, is the *expectation-maximization* (EM) algorithm proposed by Dempster et al. (1977). In detail, the maximisation in equation (2.71) is split into two steps; namely, the (E)xpectation-step and the (M)aximisation-step.

Particularly, the E-step computes the expectation of the log-joint likelihood w.r.t. the posterior distribution of all states given all observations and parameters. This expectation is often known as the $\mathcal{Q}$-function, which writes

$$\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)}) = \mathbb{E}_{p\left(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\boldsymbol{\theta}^{(l)}\right)}\left[\ln p(\mathbf{y}_{1:K}, \mathbf{x}_{0:K}|\boldsymbol{\theta})\right], \tag{2.73}$$

where

$$p(\mathbf{y}_{1:K}, \mathbf{x}_{0:K}|\boldsymbol{\theta}) = p(\mathbf{x}_0|\boldsymbol{\theta}_\dagger)\prod_{k=1}^{K} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}_\dagger)p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\theta}_*). \tag{2.74}$$

The M-step is fairly straightforward. It maximises the $\mathcal{Q}$-function w.r.t. $\boldsymbol{\theta}$. That is,

$$\boldsymbol{\theta}^{(l+1)} = \arg\max_{\boldsymbol{\theta}}\ \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)}). \tag{2.75}$$

The EM algorithm has an appealing property: The iteration between E-step and M-step always ensures that the marginal likelihood is not decreasing (Dempster et al., 1977; Wu, 1983; McLachlan and Krishnan, 1997). To see this, we expand the log-marginal likelihood according to (a special case of) the *Jensen's inequality* (Jensen, 1906),

$$\ln p(\mathbf{y}_{1:K}|\boldsymbol{\theta}) = \int_{\mathbf{x}_{0:K}} p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta}^{(l)}) \ln\frac{p(\mathbf{y}_{1:K}, \mathbf{x}_{0:K}|\boldsymbol{\theta})}{p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta})} d\mathbf{x}_{0:K} \tag{2.76}$$

$$= \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)}) - \int_{\mathbf{x}_{0:K}} p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta}^{(l)}) \ln p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta}) d\mathbf{x}_{0:K}. \tag{2.77}$$

Note that the term after $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)})$ is commonly known as the entropy of $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta}^{(l)})$ (Cover and Thomas, 1991).

To ease the presentation, let $\mathcal{H}(p|p^{(l)})$ denotes the entropy term. Considering the log-marginal likelihood evaluated at iteration l, and possible solution for the next step $\boldsymbol{\theta}$, the difference between them expands as

$$\begin{aligned}\ln p(\mathbf{y}_{1:K}|\boldsymbol{\theta}) - \ln p(\mathbf{y}_{1:K}|\boldsymbol{\theta}^{(l)}) &= \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)}) - \mathcal{Q}(\boldsymbol{\theta}^{(l)}|\boldsymbol{\theta}^{(l)}) + \mathcal{H}(p|p^{(l)}) - \mathcal{H}(p^{(l)}|p^{(l)}),\\ &= \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)}) - \mathcal{Q}(\boldsymbol{\theta}^{(l)}|\boldsymbol{\theta}^{(l)})\\ &\quad + \int_{x_{0:K}} p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta}^{(l)}) \ln\frac{p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta}^{(l)})}{p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta})} d\mathbf{x}_{0:K},\\ &= \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)}) - \mathcal{Q}(\boldsymbol{\theta}^{(l)}|\boldsymbol{\theta}^{(l)}) + \mathrm{KL}[p^{(l)}||p].\end{aligned} \tag{2.78}$$

$\mathrm{KL}[p^{(l)}||p]$ is the *KullbackLeibler divergence* between $p^{(l)}$ and p (Kullback and Leibler, 1951). According to the *Gibbs inequality*, the KL divergence is alway non-negative (Mackay, 2003). As such, in order to ensure the increase of the log-marginal likelihood, the only condition is

$$\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)}) - \mathcal{Q}(\boldsymbol{\theta}^{(l)}|\boldsymbol{\theta}^{(l)}) \geq 0. \tag{2.79}$$

Recall that the M-step finds the maximum of $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(l)})$; therefore, the iteration of EM algorithm ensures a non-decreasing marginal likelihood.

As noted by many authors (Wu, 1983; McLachlan and Krishnan, 1997; Bishop, 2006), however, the EM algorithm converges to local maximums, rather than a global maximum.

2.6 Learning in state-space models: Bayesian approaches

From a Bayesian perspective, inference in a general state-space model targets the joint posterior distribution of parameters and hidden states, denoted as $p(\boldsymbol{\theta}, \mathbf{x}_{0:K}|\mathbf{y}_{1:K})$. In this section, we review two major method classes of this kind.

2.6.1 Variational Bayes methods

The EM algorithm requires exact computation of the $\mathcal{Q}$-function; or at least, it should be possible to obtained $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta})$. Unfortunately, for many models, computing the $\mathcal{Q}$-function is intractable.

Neal and Hinton (1998) provided an alternative interpretation for EM, which relaxes the condition of exact computation of $\mathcal{Q}$-function. In detail, suppose that we do not have access to $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta})$, but instead an approximation $q(\mathbf{x}_{0:K})$ is available. Then, we expand the log-marginal likelihood with the general case of Jensen's inequality,

$$\ln p(\mathbf{y}_{1:K}|\boldsymbol{\theta}) \geq \int_{\mathbf{x}_{0:K}} q(\mathbf{x}_{0:K}) \ln \frac{p(\mathbf{y}_{1:K}, \mathbf{x}_{0:K})|\boldsymbol{\theta})}{q(\mathbf{x}_{0:K})} d\mathbf{x}_{0:K}. \tag{2.80}$$

The r.h.s. of equation (2.80) is known as the *variational lower bound* or *variational free energy* denotes as $\mathcal{F}(q, \boldsymbol{\theta})$. Similar to the original form of EM, $F(q, \boldsymbol{\theta})$ can be split into two components,

$$F(q, \boldsymbol{\theta}) = \mathcal{Q}(q, \theta) + \mathcal{H}(q), \tag{2.81}$$

where

$$\begin{aligned} \mathcal{Q}(q, \boldsymbol{\theta}) &= \int_{\mathbf{x}_{0:K}} q(\mathbf{x}_{0:K}) \ln p(\mathbf{y}_{1:K}, \mathbf{x}_{0:K}|\boldsymbol{\theta}) d\mathbf{x}_{0:K}, \\ \mathcal{H}(q) &= -\int_{\mathbf{x}_{0:K}} q(\mathbf{x}_{0:K}) \ln q(\mathbf{x}_{0:K}) d\mathbf{x}_{0:K}. \end{aligned} \tag{2.82}$$

Note that this formulation of $\mathcal{Q}$-function does not use the posterior $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta})$.

With these formulations, the EM algorithm can be seen as a *coordinate descent* method to maximise $\mathcal{F}(q, \boldsymbol{\theta})$; that is,

$$\begin{aligned} &\text{E-step:} \quad q^{(l)}(\mathbf{x}_{0:K}) = \arg\max_{q} \mathcal{F}(q, \boldsymbol{\theta}^{(l)}), \\ &\text{M-step:} \quad \boldsymbol{\theta}^{(l+1)} = \arg\max_{\boldsymbol{\theta}} \mathcal{F}(q^{(l)}, \boldsymbol{\theta}). \end{aligned} \tag{2.83}$$

Equation (2.83) is often known as the *generalised EM* (GEM) or *variational EM* algorithm. It is shown by Neal and Hinton (1998) that the GEM iteration also keeps the marginal likelihood non-decreasing. In addition the lower bound $\mathcal{F}(q, \boldsymbol{\theta})$ shares the local and global optimums with the marginal likelihood.

The variational Bayes (VB) methods (Attias, 1999; Beal, 2003) adopt the GEM framework, and takes a step forward towards a Bayesian paradigm. That is, instead of computing a point estimate in the M-step, VB produces the posterior distribution of the parameters. Specifically, VB computes an approximation $q(\mathbf{x}_{0:K}, \boldsymbol{\theta}) = q_{\mathbf{x}}(\mathbf{x}_{0:K})q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ to the actually posterior distribution, $p(\mathbf{x}_{0:K}, \boldsymbol{\theta}|\mathbf{y}_{1:K})$. This factorisation is known as the *mean-field approximation* (Jordan et al., 1999) (which will be discussed in chapter 4). To facilitate the approximation, VB maximises a variational lower bound $\mathcal{F}(q_{\mathbf{x}}, q_{\boldsymbol{\theta}})$ to the evidence of the model, $\ln p(\mathbf{y})$[5].

$$\begin{aligned} \ln p(\mathbf{y}) &\geq \int_{\mathbf{x}_{0:k}} \int_{\boldsymbol{\theta}} q_{\mathbf{x}}(\mathbf{x}_{0:K}) q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln \frac{p(\mathbf{y}_{1:K}, \mathbf{x}_{0:K}, \boldsymbol{\theta})}{q_{\mathbf{x}}(\mathbf{x}_{0:K}) q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} d\mathbf{x}_{0:K} d\boldsymbol{\theta} \\ &= \mathcal{F}(q_{\mathbf{x}}, q_{\boldsymbol{\theta}}). \end{aligned} \tag{2.84}$$

Similarly to the vEM algorithm, VB is a coordinate descent method for maximising $\mathcal{F}(q_{\mathbf{x}}, q_{\boldsymbol{\theta}})$ w.r.t $q_{\mathbf{x}}$ and $q_{\boldsymbol{\theta}}$ in turn. The resulting formulae are the iterations between VB E and M-steps for state-space models in the following (Beal, 2003):

$$\begin{aligned} &\text{VBE-step:} \quad q_{\mathbf{x}}^{(l)}(\mathbf{x}_{0:K}) \propto \exp\left(\mathbb{E}_{q_{\boldsymbol{\theta}}^{(l)}(\boldsymbol{\theta})}[\ln p(\mathbf{y}_{1:K}, \mathbf{x}_{0:K}, \boldsymbol{\theta})]\right), \\ &\text{VBM-step:} \quad q_{\boldsymbol{\theta}}^{(l+1)}(\boldsymbol{\theta}) \propto \exp\left(\mathbb{E}_{q_{\mathbf{x}}^{(l)}(\mathbf{x}_{0:K})}[\ln p(\mathbf{y}_{1:K}, \mathbf{x}_{0:K}, \boldsymbol{\theta})]\right). \end{aligned} \tag{2.85}$$

The derivation of equation (2.85) can be found in chapter 4.

The VB method was first introduce by Attias (1999) in the context of *Gaussian mixture models* (GMM) (Bishop, 2006). The GMM and state-space model belong to a larger family of models known as the *hierarchical models* (Gelman et al., 2004).

The hierarchical models are a class of models which have three basic elements: observed data, hidden variables and parameters. The hidden variables are the distinct factor, compared to normal regression models. In fact, the hierarchical models generalise the

[5] The reason of doing such an maximisation is discussed in chapter 4.

regression models, by assuming the regression basis functions are unknown. These unknown basis functions play a role as the hidden variables. In practice, a prior is added to regularise the hidden variables. The state-space models use a Markov structured transition model to describe the relation between hidden variables. In contrast, the GMM assumes that the hidden variables are independent of each other.

Example 2.6. *(Bishop, 2006) The GMM model has joint (or complete) data likelihood as*

$$\begin{aligned} p(\mathbf{X},\mathbf{Z}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) &= p(\mathbf{X}|\mathbf{Z},\boldsymbol{\mu},\boldsymbol{\Lambda})p(\mathbf{Z}|\boldsymbol{\pi}) \\ &= \prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}}\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k^{-1}\right)^{z_{nk}}, \end{aligned} \tag{2.86}$$

where z_{nk}, being the hidden variable, is a binary indicator for the nth data point belonging to the kth Gaussian component. Moreover, $\sum_{k=1}^{K} z_{nk} = 1$. $\pi_k \in [0,1]$ is the portion of the kth Gaussian component within the mixture, and $\sum_{k=1}^{K}\pi_k = 1$. $\boldsymbol{\mu}_k$ and $\boldsymbol{\Lambda}_k$ are the mean and precision matrix of each Gaussian component. Particularly, we have

$$p(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}}, \qquad p(\mathbf{X}|\mathbf{Z},\boldsymbol{\mu},\boldsymbol{\Lambda}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k^{-1}\right)^{z_{nk}}. \tag{2.87}$$

The prior settings for the parameters are

$$p(\boldsymbol{\pi}) = Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}_0), \tag{2.88}$$

$$\begin{aligned} p(\boldsymbol{\mu},\boldsymbol{\Lambda}) &= p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) \\ &= \prod_{k=1}^{K}\mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_0,(\beta_0\boldsymbol{\Lambda}_k)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_k|\mathbf{W}_0,\upsilon_0), \end{aligned} \tag{2.89}$$

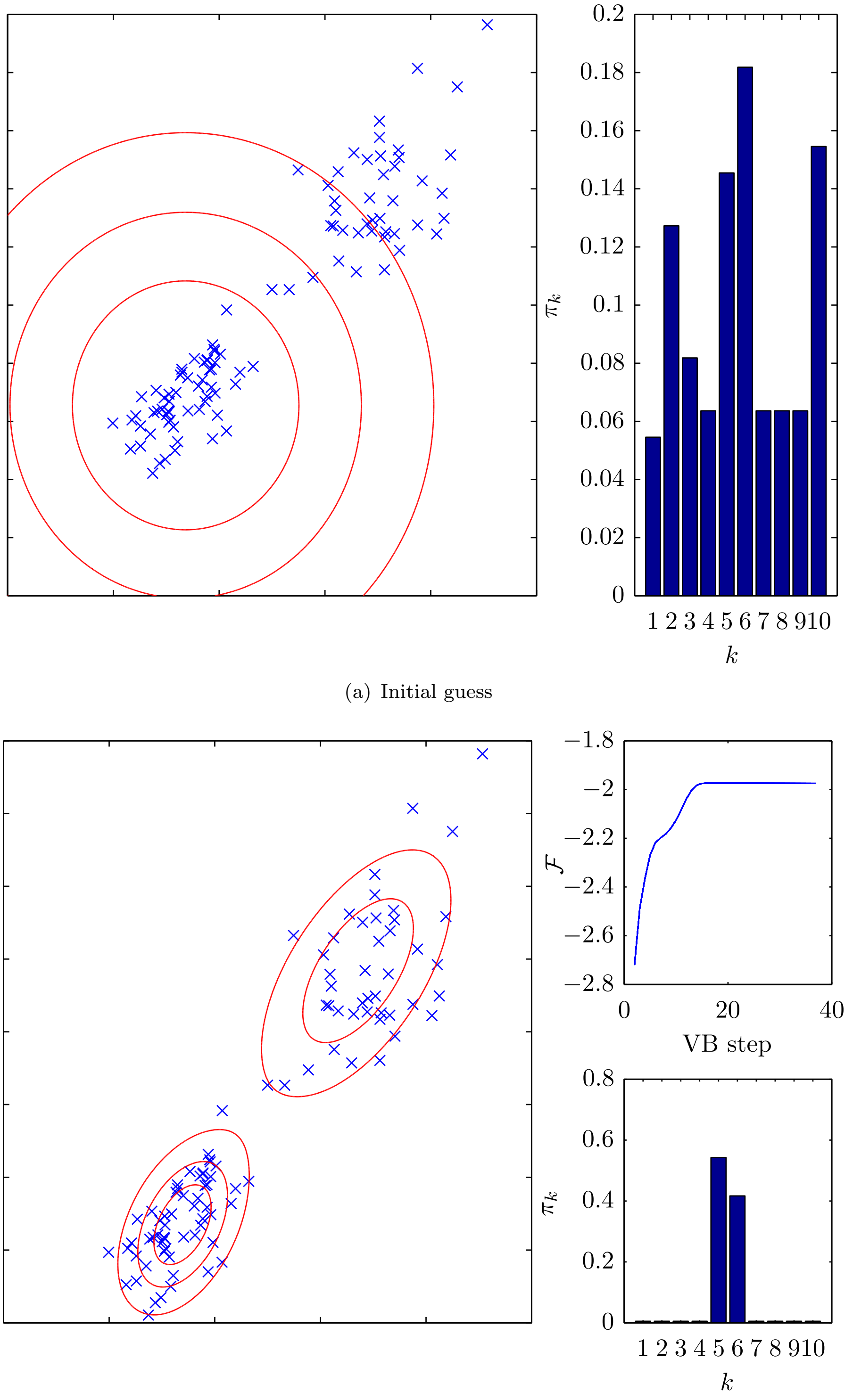
and where $\mathcal{W}$ is the Wishart distribution.

With these settings, the VB approximation to the posterior distribution $p(\mathbf{Z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}|\mathbf{X})$ is the following[6]:

$$\begin{aligned} \textit{VBE-step:}\quad & q_{\mathbf{Z}}(\mathbf{Z}) \propto \exp\left(\mathbb{E}_{q_{\boldsymbol{\pi}}q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}}[\ln p(\mathbf{X},\mathbf{Z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda})]\right), \\ \textit{VBM-step:}\quad & q_{\boldsymbol{\pi}}(\boldsymbol{\pi}) \propto \exp\left(\mathbb{E}_{q_{\mathbf{Z}}q_{\boldsymbol{\mu},\boldsymbol{\Lambda}}}[\ln p(\mathbf{X},\mathbf{Z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda})]\right), \\ & q_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\mu},\boldsymbol{\Sigma}) \propto \exp\left(\mathbb{E}_{q_{\mathbf{Z}}q_{\boldsymbol{\pi}}}[\ln p(\mathbf{X},\mathbf{Z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda})]\right). \end{aligned} \tag{2.90}$$

Figure 2.7 demonstrates a numerical example based on the above formulae. The Dirichlet prior introduces the strong shrinkage effect on π_k. Specifically, the data are generated with from a mixture of two Gaussians. When estimating the model from data, it is assumed that there are 10 *Gaussian components. At the end of the estimation, only two Gaussian components are left with significant portion.*

[6] More details of these approximations can be found in Bishop (2006)



(a) Initial guess

(b) Final convergence

Figure 2.7: Illustration of VB for GMM. For both (a) and (b), on the *left*, we have reconstructed Gaussian mixture density shown as *red contour* and observed data points in *blue* crosses. On the *right*, the expected portions $\{\mathbb{E}_{q_\pi}[\pi_k]\}$ are displayed. (a): The initial guess on the parameters of the Gaussian mixtures. (b): The convergence results. The value of the lower bound $\mathcal{F}$ across iterations is also demonstrated.

2.6.2 Markov chain Monte Carlo (MCMC) methods

As a contrast to the VB method, the MCMC methods focus on generating samples from the joint posterior. A detailed review on this subject can be found in Fearnhead (2010). A Gibbs sampler, iteratively drawing samples from $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\mathbf{x}_{0:K}, \mathbf{y}_{1:K})$, is the most popular method to sample from such a posterior distribution. In practice, sampling from $p(\boldsymbol{\theta}|\mathbf{x}_{0:K}, \mathbf{y}_{1:K})$ is often easy, whereas designing a sampler for $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta})$ is trickier, due to the fact that the states are highly correlated and can have large variation in scale.

The simplest implementation of such a sampling approach is a single-site update Gibbs sampler for both hidden states and parameters, where the components of $\mathbf{x}_{0:K}$ and $\boldsymbol{\theta}$ are updated one at a time (see Geweke and Tanizaki (2001) for details). For sampling states, a sequential sampler which updates each state conditioning on all the rest of the states is used. Such an approach is easy to implement, since the conditional distribution of each state given all the others reduces to one conditioning only on its two adjacent states: $p(\mathbf{x}_k|\mathbf{y}_{1:K}, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \boldsymbol{\theta})$. However, due to the severe correlation between states, such a sampler may lead to slow mixing (such slow mixing is evident in the SSPP; empirical results are shown in section 7).

To overcome this, Shephard and Pitt (1997) propose a block Gibbs sampler, in which instead of single-site updating, the states are grouped into many blocks and updated simultaneously. In this case, the conditionals on states change to the density of each block of states given the two neighbouring states of the block: $p(\mathbf{x}_{k:s}|\mathbf{y}_{1:K}, \mathbf{x}_{k-1}, \mathbf{x}_{s+1})$, where $k < s < K$. Ideally, one needs the block to be as large as possible; however, when the block size is too large, it is hard to sample from the conditional in most general state space models. If the block is not large enough, the sampler still suffers from state dependency issues. A balance between the extremes is often difficult to strike.

In the case of block size set equal to the total time points in the model, the whole state sequences are updated simultaneously from $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta})$. Such updates can be performed exactly only in the linear Gaussian models using the Kalman filter (Carter and Kohn, 1994) and discrete hidden Markov model using the forward-backward method (Scott, 2002). However, recent developments in MCMC provide flexible means for updating the whole state sequence for more general state-space models. In chapter 5, we introduce several such efficient sampling schemes that can be applied to the SSPP model.

2.7 Discussions

In this chapter, we reviewed a range of models which are designed to handle neural spikes and heartbeats. Whilst most of these models translate the discrete events into counts and continuous-valued intervals, the point process models directly handle the discrete events. Together with a state-space approach, the state-space models with point process observations show great potentials for characterising this type of data.

We then set out to show traditional state-space models and their mathematical formulations. In particular, a probabilistic formulation, which nicely generalises linear/nonlinear and Gaussian/non-Gaussian models, is highlighted.

From an algorithmic point of view, we presented a roadmap of methods for solving inference and learning problems in traditional state-space models. On estimating the hidden states, starting from the well-known Kalman/extended Kalman filters, the method of choice converges to a more general Bayesian filtering framework which can be applied to many models. Likewise, we showed an evolution of approaches for learning the unknown parameters, from the EM algorithm which gives point estimates to the more powerful Bayesian methods including variational Bayes and MCMC.

Chapter 3

State-space models with point process observations

In this chapter, we introduce the general framework of state-space models with point process observations (SSPP), and their relation with other point process models. An approximate expectation-maximisation algorithm for inference and learning is studied both theoretically and empirically, by which we identify several properties of SSPP and state-space models in general. We noted that the approximate filtering approach has led to biases in parameter estimation. This motivates the Bayesian treatments in later chapters.

3.1 Point process models

3.1.1 Data formats

In probability theory, a point process is considered as a sequence of discrete events occurring in continuous or discrete time. As shown in the previous chapter, for signals like neural spike train and heartbeat, the events are often represented by pulses of which the amplitude or height is not of interest. Naturally, such a sequence of pulses can be described by several format:

- *Pulse occurring time.* The pulse occurring time (or pulse time) is normally considered as the original format.
- *Inter-pulse interval (IPI).* A sequence of waiting time before the next pulse occurs. Traditionally, the IPI is the main data format of heartbeats.
- *Counting function.* A function that counts how many pulses happened over time. Also known as counting process.

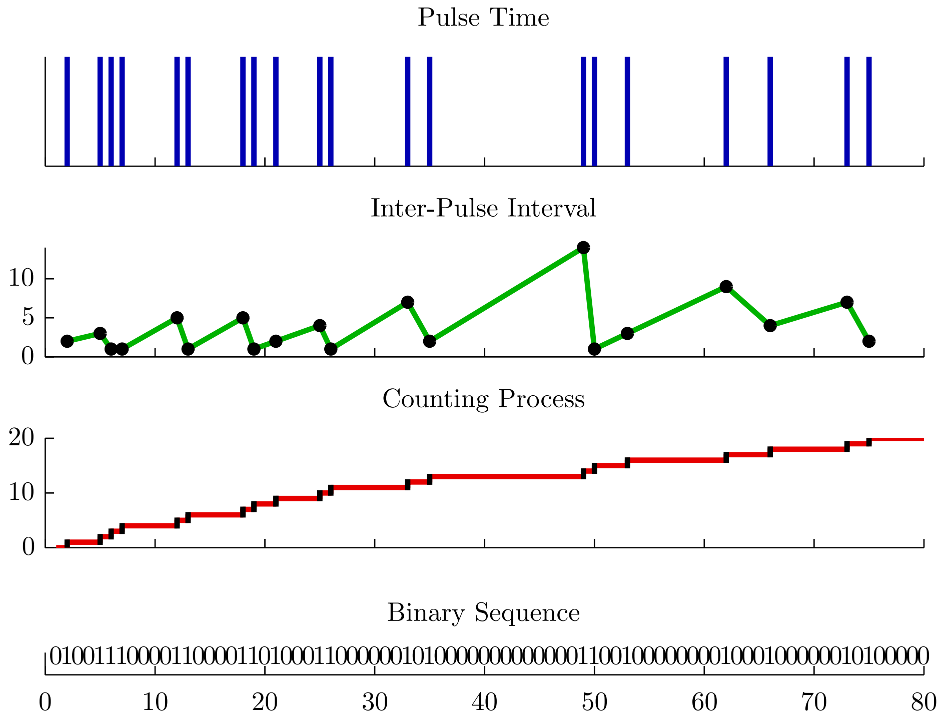


Figure 3.1: Four different descriptors of a toy point process dataset. The dataset is constructed as 20 events happened at randomly selected time locations within total 80 unitless time points. We treat the pulse time (*blue*) as the original format of the point process. The inter-pulse interval (*green*) is the difference between adjacent pulse times. The counting process (*red*) is a cumulative function of the pulse number. The binary sequence can be seen as either indicators of pulse or counting differences between adjacent time points.

- *Discrete binary sequence.* An indicator function of pulse time.

To visualise these signal formats, a toy point process dataset is shown in figure 3.1.

Although the pulse time carries all information about the system of interest, it is not straightforward to model it directly. The other three signal formats are all functions (or transformations) of the pulse time. Throughout this thesis, we focus on modelling the *Binary Sequence* format, whereas the other formats will appear as supporting blocks of the framework.

3.1.2 Continuous time likelihood

We begin with a mathematical definition of a point process. Given an observation interval $(0, T]$, let an ensemble $\{s_j; 0 < s_1 < \cdots < s_j < \cdots < s_J \le T\}$ present the pulse time locations. The binary representation of a point process , $y(t)$, can be written as

$$y(t) = \begin{cases} 1, & \text{if } t = s_j \\ 0, & \text{if } t \neq s_j \end{cases} \quad \forall j \in [1, \ldots, J]. \tag{3.1}$$

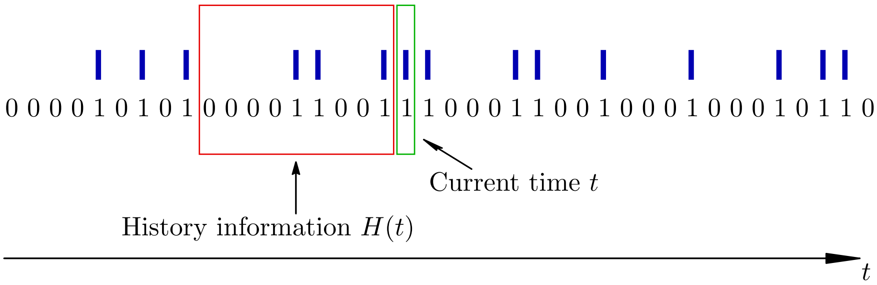


Figure 3.2: An example of history information $H(t)$ in relation to the current time t.

Subsequently, the counting process (or function) is denoted as $Y(t)$, such that $Y(t+\Delta)-Y(t)=y(t)$. Now we have set up the necessary notations for a probabilistic framework of point processes. Before we introduce the likelihood of a point process, we first introduce the idea of the *Conditional Intensity Function*(CIF). According to Daley and Vere-Jones (2003), any point process can be fully characterised by the CIF, $\lambda(\cdot)$, which has the expression

$$\lambda\left(t|H(t)\right)=\lim_{\Delta\to 0}\frac{\Pr\left(Y(t+\Delta)-Y(t)=1|H(t)\right)}{\Delta}. \tag{3.2}$$

The term "conditional" is expressed through $H(t)$ which represents the history information of a point process involved at current time t. Such information are essentially the previous pulses shown as figure 3.2. Note that it is possible to have a conditional intensity function involving no history information. In this case, the "condition" could represent any information of interest.

A concrete example of the CIF is the rate function of a Poisson process[1]. A useful consequence due to equation (3.2) is that

$$\Pr\left(y(t)=1|H(t)\right)=\lambda(t|H(t))\Delta, \tag{3.3}$$

where $y(t)=Y(t+\Delta)-Y(t)$ is the increment of the counting process; and Δ is the time resolution. The above equation is crucial in simulating the point process (Brown et al., 2002), and as we shall see later in this chapter, equation (3.3) is the only parameter required for simulation.

Now we can write a likelihood function for a point process in continuous time. Given the CIF, let $y_{(0,t]}:=\{y(t)|t\in(0,t]\}$ and $H_{(0,t]}:=\{H(t)|t\in(0,t]\}$ the likelihood is

$$p\left(y_{(0,t]}|H_{(0,t]}\right)=\exp\left(\int_0^t\ln\lambda\left(t|H(t)\right)dY(t)-\int_0^t\lambda\left(t|H(t)\right)dt\right). \tag{3.4}$$

[1] The rate function of the Poisson process does not involve any history information.

The above function has different derivations; for instance, Daley and Vere-Jones (2003) derive the formula from a generalised Poisson process, while Brown et al. (2003) show a derivation from correlated Bernoulli trials.

The summation version of the continuous likelihood can be written as the following (Brown et al., 2003; Truccolo et al., 2005)

$$\lim_{\Delta \to 0} p\left(y_{(0,k\Delta]} | H_{(0,k\Delta]}\right) = \lim_{\Delta \to 0} \exp\left(\sum_{i=1}^{k} y(i\Delta) \ln\left(\lambda\left(i\Delta | H(i\Delta)\right)\right) - \sum_{i=1}^{k} \lambda\left(i\Delta | H(i\Delta)\right) \Delta\right) + \text{const},$$

where Δ is the discretisation resolution which divided $(0, t]$ into k intervals.

3.1.3 Discrete time likelihood

To work with a discrete time model, we choose K large to divide the observation interval $(0, T]$ into K intervals with equal width $\Delta = T/K$. This naturally yields a time index $k \in [1, \ldots, K]$. Similar to the continuous time model, in the discrete time model, we have y_k for binary sequence, Y_k for counting function, λ_k for the CIF and H_k for history information.

The discrete time likelihood can be derived from both Poisson and Bernoulli processes. Let $\mathbf{y} := [y_1, \ldots, y_K]^{\mathrm{T}}$ and $\boldsymbol{\lambda} := [\lambda_1, \ldots, \lambda_K]^{\mathrm{T}}$. By the definition of the inhomogeneous Poisson process, we have y_k from a Poisson distribution Pois($\lambda_k \Delta$); moreover y_ks are independent, provided that the discretised intervals do not overlap. Hence the likelihood function $p(\mathbf{y}|\boldsymbol{\lambda})$ can be written as a product of K number of Poisson distributions.

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\lambda}) &= \exp\left(\sum_{k=1}^{K} \left(y_k \ln(\lambda_k \Delta) - \lambda_k \Delta - \ln(y_k!)\right)\right), && (3.5) \\ &\propto \exp\left(\sum_{k=1}^{K} \left(y_k \ln(\lambda_k \Delta) - \lambda_k \Delta\right)\right). && (3.6) \end{aligned}$$

Recall that, we assume that Δ is very small; such that for each bin there is at maximum 1 spike. This means the probability of y_k being larger than 1 can be ignored. In addition, if Δ is sufficiently small, and the probability of having a spike is small; then, in an approximate fashion, y_ks can be seen as a independent Bernoulli random variable with Ber($\lambda_k \Delta$). We therefore write the likelihood function as a product of Bernoulli distributions,

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\lambda}) &= \exp\left(\sum_{k=1}^{K} \ln\left((\lambda_k \Delta)^{y_k} (1 - \lambda_k \Delta)^{(1 - y_k)}\right)\right) && (3.7) \\ &= \exp\left(\sum_{k=1}^{K} \left(y_k \ln\left(\frac{\lambda_k \Delta}{1 - \lambda_k \Delta}\right) + \ln(1 - \lambda_k \Delta)\right)\right). && (3.8) \end{aligned}$$

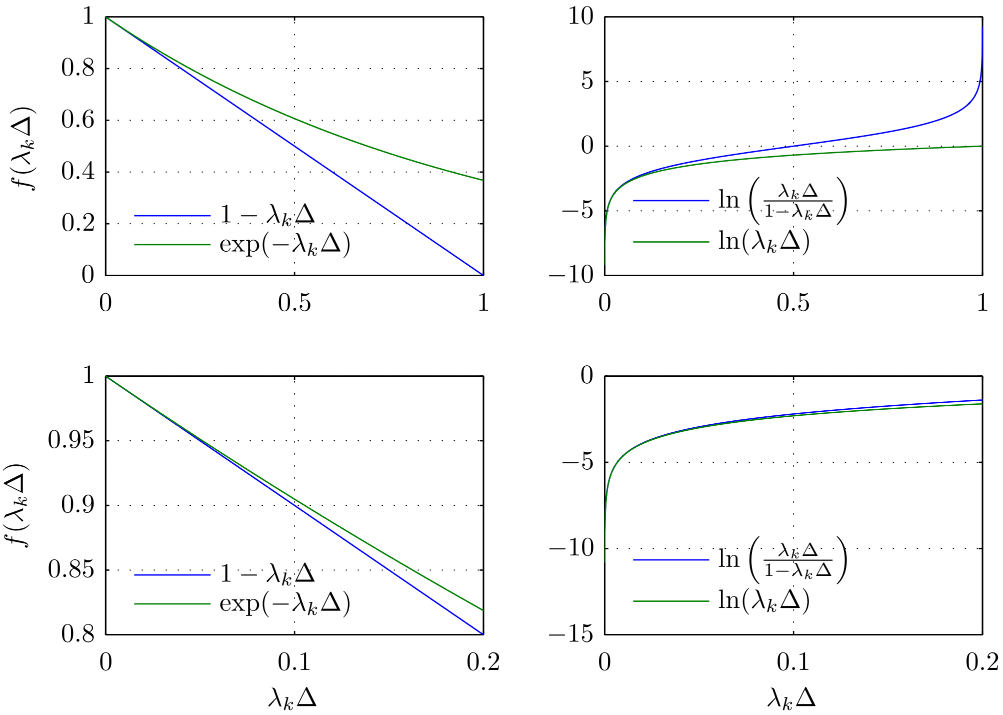


Figure 3.3: An illustration of the approximations in equation (3.9) (*left column*) and (3.10) (*right column*). The *top row* shows all possible values of $\lambda_k\Delta$, while the *bottom row* displays zoom-in version where $\lambda_k\Delta$ takes from 0 to 0.2. For the approximations to be valid, $\lambda_k\Delta$ has to be small. For both cases, performance drops as $\lambda_k\Delta$ increases.

Using the fact that the probability of firing a spike is small ($\lambda_k\Delta$ is small), equations 3.7 and 3.5 are equivalent. To show this, we use the following two approximations (Brown et al., 2003; Truccolo et al., 2005):

$$1-\lambda_k\Delta \approx \exp(-\lambda\Delta), \tag{3.9}$$

$$\ln\left(\frac{\lambda_k\Delta}{1-\lambda_k\Delta}\right) \approx \ln(\lambda_k\Delta). \tag{3.10}$$

The effect of these approximations is illustrated in figure 3.3. As expected, their validity depends on $\lambda_k\Delta$ being small.

We rewrite the Bernoulli likelihood–equation (3.7) with the approximations, which yields

$$p(\mathbf{y}|\boldsymbol{\lambda}) = \exp\left(\sum_{k=1}^{K}(y_k\ln(\lambda_k\Delta)-\lambda_k\Delta)\right). \tag{3.11}$$

Hence, we have a discrete likelihood for the general point process. As we shall see later in this chapter, the approximation is mathematically convenient when the model is treated in a hierarchical fashion. In addition, these likelihood models can be summarised by the directed graphical models (also known as direct acyclic graphs) shown as figure 3.4.

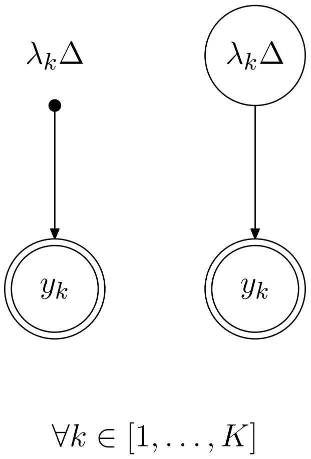


Figure 3.4: Directed graphical models for point process models defined by equation (3.11), in which $\lambda_k\Delta$ is a nonrandom variable (*left*) or a random variable (*right*). In both cases, variables at different time indexes are independent.

3.1.4 Conditional intensity function

In the previous subsection, we have built a conditional probability model $p(\mathbf{y}|\boldsymbol{\lambda})$, in which the CIF, $\boldsymbol{\lambda}$, serves as a parameter. Various ways of treating the CIF are the main difference between members of the point process family. Fundamentally, there are two categories: either seeing $\boldsymbol{\lambda}$ as a non-random variable, or as a random variable. The first falls into the theme of *generalised linear model* (GLM), therefore named as point process GLM (PPGLM). The latter is called the *doubly stochastic point process* (DSPP) (Daley and Vere-Jones, 2003).

To see how the CIF is formulated in these two different settings, we start with its definition. Specifically, the CIF is a function of some time-varying features. These features can be considered as a d-dimensional *basis function*[2] vector, $\boldsymbol{\phi}_k \in \mathbb{R}^d$, where d is the number of the features. Accordingly, a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$ is defined to be the weights by which the basis functions are scaled. With these terminologies, the CIF can be written as

$$\lambda_k = f(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k). \tag{3.12}$$

The difference between the two modelling strategies lies in the treatments on the basis functions $\boldsymbol{\phi}_k$. In particular, PPGLM treats $\boldsymbol{\phi}_k$ as non-random variables, whereas in the DSPP setting, $\boldsymbol{\phi}_k$ are assigned with probability densities; in other words, *priors*. For both cases, the parameter $\boldsymbol{\theta}$ can be either non-random or random, representing the frequentist and Bayesian approaches within the two model categories. These differences have been summarised graphically in figure 3.5.

[2] A concrete example of the basis function is sinusoid function, in which case the parameters are considered as Fourier coefficients. Radial basis function and sigmoid function are also commonly adopted.

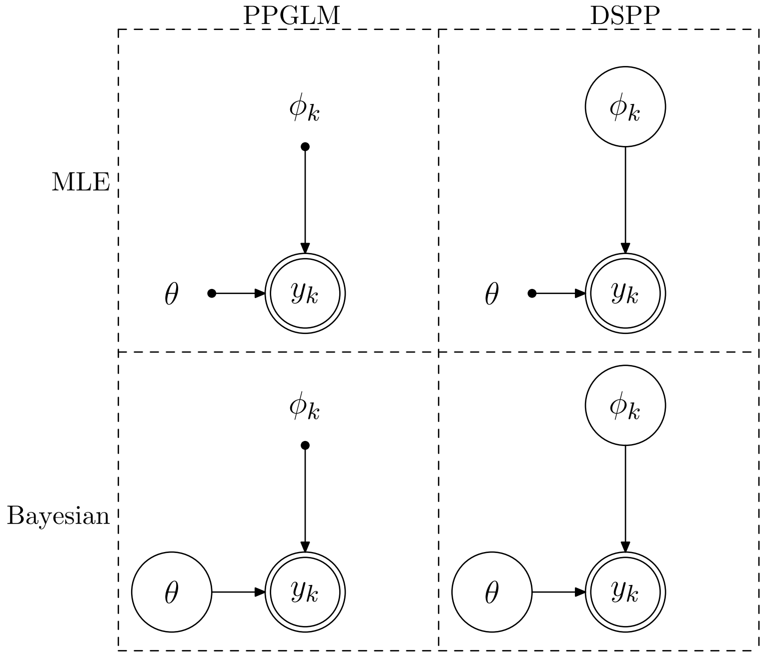


Figure 3.5: Graphical models for PPGLM and DSPP with Bayesian and MLE treatment to the parameter.

In principle, the function $f(\cdot)$, can take any form that keeps the equation (3.11) likelihood function within the exponential family. The exponential function and the sigmoid function are the most popular choices. In particular, the exponential function is preferred in the Poisson and general likelihood. The sigmoid is the primary choice for the Bernoulli likelihood. These choices give the likelihood model a well-known property:

Theorem 3.1. *In the following settings for likelihood functions: (i). Equation (3.6) with $\lambda_k = \exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k)$; (ii). Equation (3.7) with $\lambda_k\Delta = \sigma(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k)$, where $\sigma(a) = \frac{1}{1+\exp(-a)}$; (iii). Equation (3.11) with $\lambda_k = \exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k)$. Given $\boldsymbol{\phi}_k$, $\forall k$, the log-likelihoods are concave in the parameter $\boldsymbol{\theta}$ and vice versa.*

Proof. The detailed proof is given in appendix. □

Remark 3.2. *Theorem 3.1 ensures the uniqueness of the maximum likelihood estimation (MLE) of $\boldsymbol{\theta}$ in the PPGLM setting, in which $\boldsymbol{\phi}_k$ is given.*

3.1.5 Point process generative model

The generative model for point process is also known as the Poisson spike generator (Dayan and Abbott, 2001). It is based on the idea in equation (3.3) that the probability of y_k being 1 is $\lambda_k\Delta$. The algorithm for generating a channel of point process data is the following:

Algorithm 3.1. Discrete process process generator

```
Input: $\lambda_k, \forall k \in [1, \cdots, K]$, $\Delta$
Output: $y_k = \{0, 1\}, \forall k \in [1, \cdots, K]$
for $k = 1$ to $K$ do
    Draw $u \sim \mathcal{U}(0, 1)$
    if $u < \lambda_k \Delta$ then
        $y_k = 1$
    else
        $y_k = 0$
    end if
end for
```

It is easy to recognise that this procedure is similar to the accept/reject step in a Metropolis-Hastings algorithm, reflecting the fact that at each time point the system produces 1 with probability $\lambda_k\Delta$. Let us, for the moment, consider an example which adopts the PPGLM paradigm with the CIF written as weighted sum of some fixed basis functions.

Example 3.1. *Generate three binary point process datasets on a* $6s$ *time interval with resolution* $\Delta = 10ms$ *(resulting* $K = 600$ *time points) using algorithm 3.1. In this example, the CIF is chosen be a scaled sigmoid function, such that* $\lambda_k\Delta = \sigma\left(\theta_0 + \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k\right)$, $\forall k \in [1, \cdots, K]$, *where* θ_0 *is a base intensity/rate and set to be* -3*. In practice, this term can be added into the parameter vector. Accordingly, the basis function vector will include an additional* 1 *. The parameter* θ *is drawn from the standard normal distribution. Each data stream is produced by the sinusoidal, radial and sigmoidal basis functions, respectively, in which the details of these functions are the following:*

$$\textit{Sinusoidal: } \phi_{i,k} = \sin(\omega_i k\Delta + \psi_i), \ \forall i \in [1, \cdots, N], \tag{3.13}$$

$$\textit{Radial: } \phi_{i,k} = \exp\left(-\frac{(i\Delta - k\Delta)^2}{2l^2}\right), \ \forall i \in [1, \cdots, N = K | i \neq k], \tag{3.14}$$

$$\textit{Sigmoidal: } \phi_{i,k} = \sigma\left(\frac{i\Delta - k\Delta}{l}\right), \ \forall i \in [1, \cdots, N = K | i \neq k]. \tag{3.15}$$

For the sinusoid case, the phase terms are further linearised and merged into the parameter $\boldsymbol{\theta}$*, by letting* $\sin(\omega_i k\Delta + \psi_i) = \sin\psi_i \cos\omega_i k\Delta + \cos\psi_i \sin\omega_i k\Delta$*. To ensure visual clarity,* 5 *frequency components uniformly ranging from* 0 *to* $\pi/2$ *are selected. For both radial and sigmoidal cases, the dimension of the basis function vector is* K*. Further, the hyperparameter* l *is set to be* $\sqrt{0.08}$*. With these settings, figure 3.6 presents a realisation of the data simulation.*

With example 3.1 and figure 3.6, we have seen the effect of external basis functions. Here, we discuss a different type of the basis function, which is the history information of the point process data. More explicitly, $\boldsymbol{\phi}_k = [y_{k-1}, \cdots, y_{k-N}]^{\mathrm{T}}$, $\forall k \in [1, \cdots, N | N < k]$. This can be seen as a self-exited point process or an autoregressive point process which

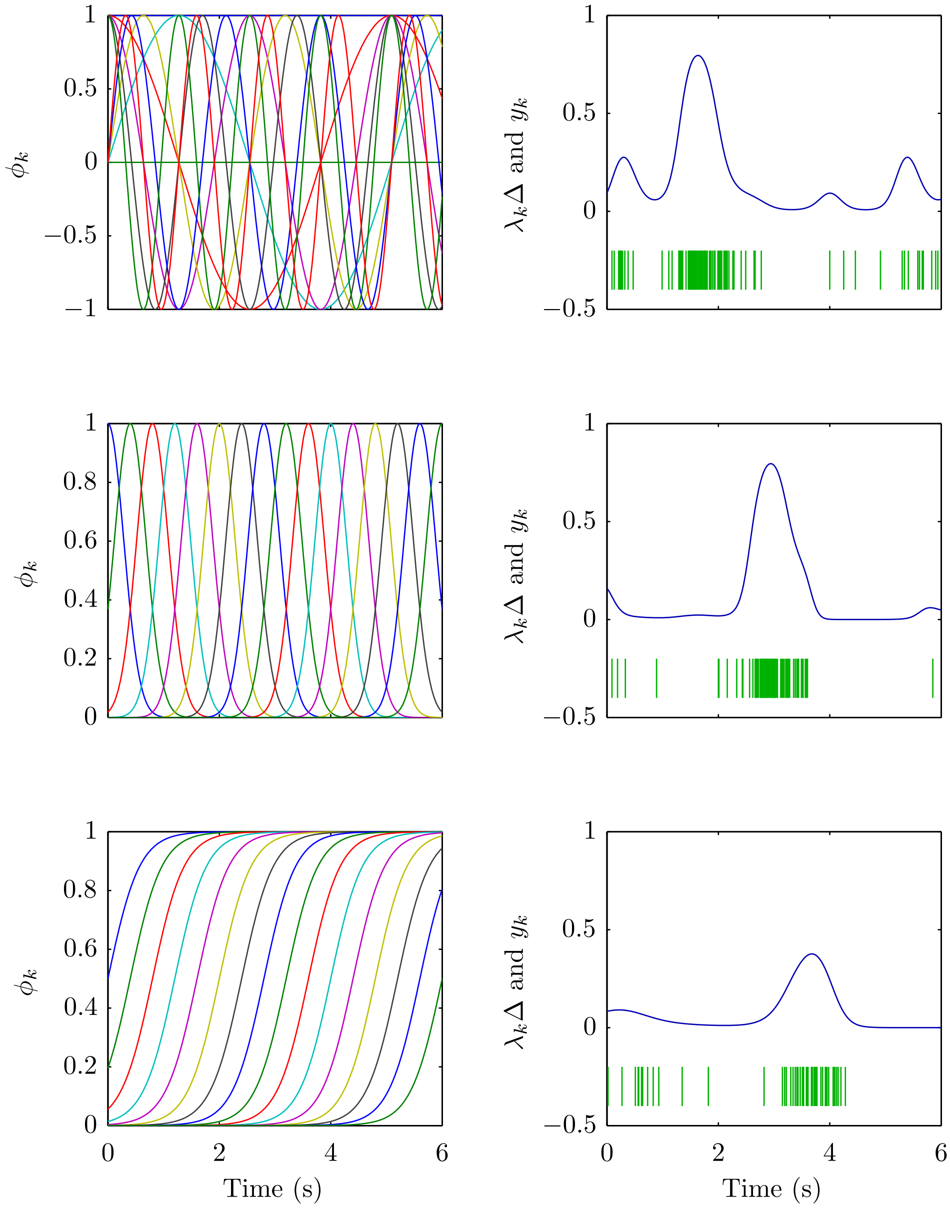


Figure 3.6: Illustration of synthetic data as described in example 3.1. Showing on the *left* are three types of basis function $\boldsymbol{\phi}_k$ ranging from sinusoidal (*top*), radial (*middle*) and sigmoidal (*bottom*) basis functions, while on the *right* are the corresponding CIF λ_k (*blue curve*) and point process data y_k (*green bars*). For radial and sigmoidial basis function, only 15 components (every 40th component) are displayed for visual clarity. Further, $\boldsymbol{\theta}$ is the same during the simulations with these two cases.

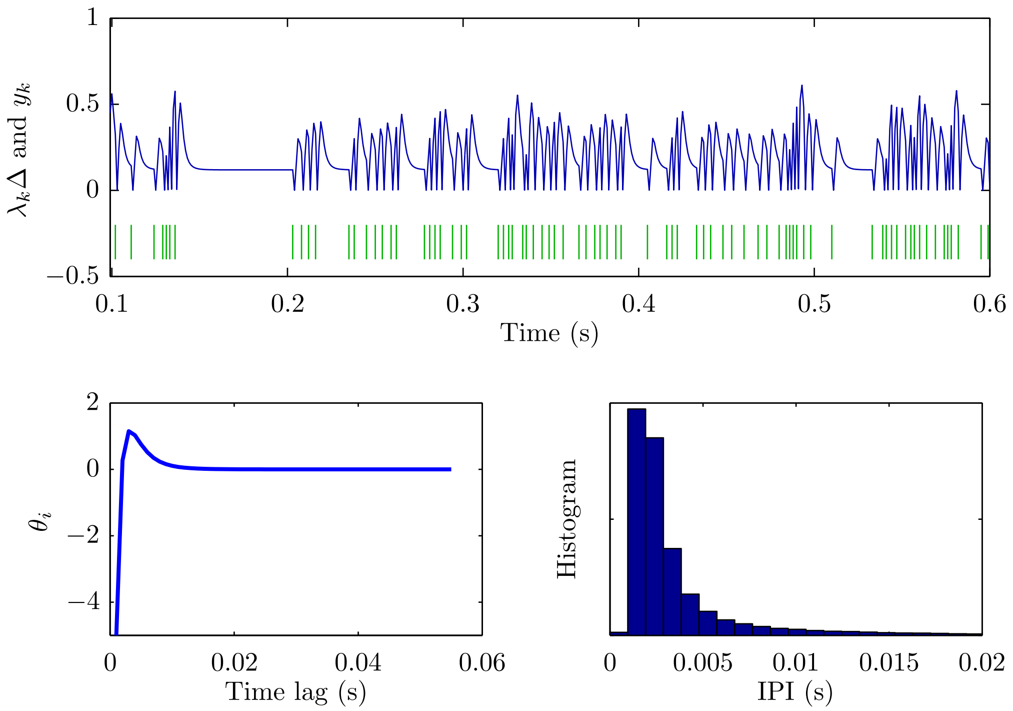


Figure 3.7: Illustration of a synthetic self-excited point process. *Top*: A 0.6s segment ($\Delta = 1$ms) of CIF λ_k (*blue curve*) and point process data y_k (*green bars*), where $\lambda_k = \sigma(\cdot)$ with a baseline intensity $\theta_0 = -2$. *Bottom left*: The $\boldsymbol{\theta}$ used for this simulation. *Bottom right*: the histogram of IPIs (total 115,062 pulses during a 500s simulation). Note that, due to parameter setting, $\lambda_k\Delta$ sharply drops after every pulse.

is similar to the autoregressive models for continuous time series. The CIF which spans $\boldsymbol{\phi}_k$ is

$$\lambda_k = f\left(\sum_{i=1}^{N} \theta_i y_{k-i}\right) \tag{3.16}$$

The above parameterisation enables the point process generative model to capture some essential patterns in real physiological data. Of particular interest are *burst* and *refractory* behaviour. The burst behaviour is often seen in neural spike trains, in which over a short period of time, a neuron or a group of neurons exhibits intense spiking activities. Such behaviour can be easily implemented by setting a set of parameters with large values. The refractory behaviour is more commonly observed in many organisms, where the system is incapable of firing or producing a electrical pulse, immediately or over a short period after a successful spiking activity. This can be realised by choosing parameters with negative values. Figure 3.7 presents simulations with this setting.

Comparing figures 3.6 and 3.7, apparently, the CIFs based on external basis functions are much smoother than the self-excited ones. Therefore, these basis functions are good at modelling some internal dynamics in which high frequency variations are believed to be rare. In practice, it often of interest to combine the two types of modelling approach, constructing a larger $\boldsymbol{\phi}_k$ contains both external inputs and spiking history (Truccolo et al., 2005).

3.2 The state-space model with point process observations

In the previous section, we have discussed several examples from the PPGLM family. Here, the focus moves on to an important member of the DSPP paradigm, the *state-space model with point process observations* (SSPP) (Smith and Brown, 2003).

We use the standard notation $\mathbf{x}_k := [x_{1,k}, \cdots, x_{N,k}]^{\mathrm{T}}$ to denote the hidden state at time point k. Let $\mathbf{u}_k := [u_{1,k}, \cdots, u_{M,k}]^{\mathrm{T}}$ be the term for the exogenous input, which represents the controlled external stimulus in a typical neurophysiological experiment. An first-order autoregressive (AR(1)) describing the state dynamics is the following:

$$x_{i,k} = \boldsymbol{\rho}_i^{\mathrm{T}} \mathbf{x}_{k-1} + \boldsymbol{\alpha}_i^{\mathrm{T}} \mathbf{u}_k + \varepsilon_{i,k}, \qquad \forall k = [1, \cdots, K], \tag{3.17}$$

where $\boldsymbol{\rho}_i = [\rho_{i,1}, \cdots, \rho_{i,N}]^{\mathrm{T}}$ is the AR coefficient, $\boldsymbol{\alpha}_i = [\alpha_{i,1}, \cdots, \alpha_{i,M}]^{\mathrm{T}}$ is the weight of the input, and ε_k is the process noise drawing form $\mathcal{N}(0, \sigma^2_{\varepsilon_i})$. As a result, a conditional density of a element in the current vector $x_{i,k}$, given $\mathbf{x}_{k-1}$ and parameter $\boldsymbol{\theta}_i := \{\boldsymbol{\rho}_i, \boldsymbol{\alpha}_i, \sigma^2_{\varepsilon_i}\}$, arises as:

$$x_{i,k} | \mathbf{x}_{k-1}, \boldsymbol{\theta}_i \sim \mathcal{N}(\boldsymbol{\rho}_i^{\mathrm{T}} \mathbf{x}_{k-1} + \boldsymbol{\alpha}_i^{\mathrm{T}} \mathbf{u}_k, \sigma^2_{\varepsilon_i}) \tag{3.18}$$

For the sake of convenience, we define the initial state $x_0 \sim \mathcal{N}(\boldsymbol{\pi}_0, \boldsymbol{\Sigma}_0)$. With these density settings, a joint density of states $\mathbf{x}_{0:K} := \{\mathbf{x}_k\}_{k=0}^{K}$ is the product of the conditional density at each time point:

$$p(\mathbf{x}_{0:K} | \boldsymbol{\theta}_{\dagger}) = p(\mathbf{x}_0 | \boldsymbol{\pi}_0, \boldsymbol{\Sigma}_0) \prod_{k=1}^{K} \prod_{i=1}^{N} p(x_{i,k} | \mathbf{x}_{k-1}, \boldsymbol{\theta}_i), \tag{3.19}$$

where $\boldsymbol{\theta}_{\dagger} = \{\boldsymbol{\pi}_0, \boldsymbol{\Sigma}_0, \boldsymbol{\theta}_{1:N}\}$ and $\boldsymbol{\theta}_{1:N} = \{\boldsymbol{\theta}_i\}_{i=1}^{N}$. Equation (3.19) defines a prior over the basis function at all time points.

The likelihood is defined over C number of channels of binary sequences. Mathematically, we add a superscript c for channel index to the CIF λ_k and binary sequence y_k. We use the general point process likelihood equation (3.11) with exponential CIF in this framework. To picture the state process with the general CIF in equation (3.12), let $\boldsymbol{\theta}_c := [\mu_c, \boldsymbol{\beta}_c^{\mathrm{T}}]^{\mathrm{T}}$ and $\boldsymbol{\phi}_k = [1, \mathbf{x}_k^{\mathrm{T}}]^{\mathrm{T}}$, the CIF becomes:

$$\lambda_{c,k} = \exp(\boldsymbol{\theta}_c^{\mathrm{T}} \boldsymbol{\phi}_k) = \exp(\mu_c + \boldsymbol{\beta}_c^{\mathrm{T}} \mathbf{x}_k), \tag{3.20}$$

where μ_c and $\boldsymbol{\beta}_c = [\beta_{1,c}, \cdots, \beta_{N,c}]^{\mathrm{T}}$ are the log of the background firing rate and the weight on the states for each channel. As a result, the likelihood or the observation

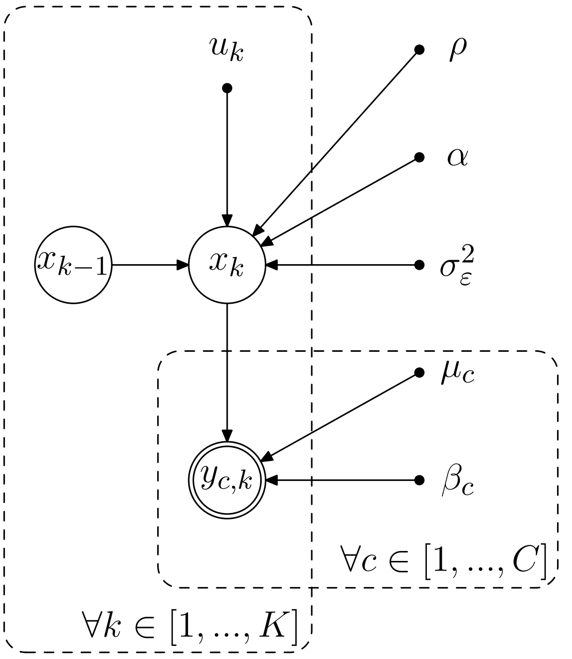


Figure 3.8: A graphical representation of the SSPP, in which the parameters are considered as non-random variables. The panels correspond to the index of the variables.

distribution can be written as the following:

$$p(y_{c,k}|\mathbf{x}_k, \boldsymbol{\theta}_c) = \exp(y_{c,k}(\mu_c + \boldsymbol{\beta}_c^{\mathrm{T}}\mathbf{x}_k + \log\Delta) - \exp(\mu_c + \boldsymbol{\beta}_c^{\mathrm{T}}\mathbf{x}_k)\Delta), \tag{3.21}$$

Let $\mathbf{y}_k := \{y_{c,k}\}_{c=1}^C$ and $\mathbf{y}_{1:k} = \{\mathbf{y}_k\}_{k=1}^k$, we can write down the complete data likelihood for the model:

$$p(\mathbf{x}_{0:K}, \mathbf{y}_{1:K}|\boldsymbol{\theta}) = p(\mathbf{x}_{0:K}|\boldsymbol{\theta}_\dagger) \prod_{k=1}^{K}\prod_{c=1}^{C} p(y_{c,k}|\mathbf{x}_k, \boldsymbol{\theta}_c), \tag{3.22}$$

where $\boldsymbol{\theta} := \{\boldsymbol{\theta}_\dagger, \boldsymbol{\theta}_{1:C}\}$ and $\boldsymbol{\theta}_{1:C} = \{\boldsymbol{\theta}_c\}_{c=1}^C$. Now we have completely setup the SSPP model. The parameters in this chapter are considered as non-random variables. A graphical representation for the SSPP is showing as figure 3.8. With the above formulations, we present an example of SSPP as a generative model.

Example 3.2. *Generate a* 10*-channel point process data in a* 20*s time interval with* $\Delta = 10ms$ *(resulting* $K = 2,000$ *time points). The input are with* 1 *at every second. Defined a SSPP with the following parameter settings:* $\rho = 0.8$*,* $\alpha = 4$*,* σ_ε^2*,* $\mu = 0$ *and* $\beta_{1:C} = [0.5, \cdots, 1]^T$*. Figure 3.9 illustrates a realisation with these settings.*

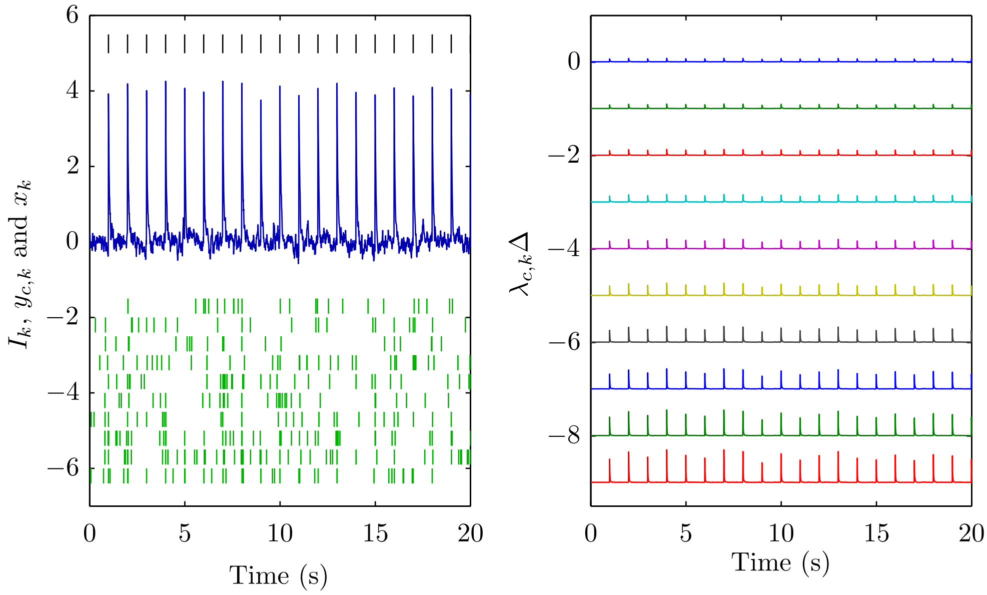


Figure 3.9: Illustration of synthetic data with SSPP as described in example 3.2. *Left*: Inputs (*black bars*), states (*blue curve*) and pulses (*green bars*). Artificial scales are added to visualise data in each channel. *Right*: The CIFs of each channel (different in colors), top-down ranked corresponds to the pulses on the left panel.

3.3 An approximate EM algorithm

To simultaneously estimate the states and parameters, Smith and Brown (2003) proposed an approximate EM algorithm. In the section, we discuss this algorithm in detail. Recall that the EM algorithm maximises a lower bound to the log-marginal likelihood:

$$\begin{aligned}\ln p(\mathbf{y}_{1:K}|\boldsymbol{\theta}) &\geq \mathcal{F}(q,\boldsymbol{\theta}) \\ &= \underbrace{\int_{\mathbf{x}_{0:K}} q(\mathbf{x}_{0:K})\ell(\mathbf{y}_{1:K},\mathbf{x}_{0:K}|\boldsymbol{\theta})d\mathbf{x}_{0:K}}_{\mathcal{Q}(\theta,q)} - \underbrace{\int_{\mathbf{x}_{0:K}} q(\mathbf{x}_{0:K})\ln q(\mathbf{x}_{0:K})d\mathbf{x}_{0:K}}_{-\mathcal{H}(q)},\end{aligned} \tag{3.23}$$

where $\ell(\mathbf{y}_{1:K},\mathbf{x}_{0:K}|\boldsymbol{\theta})$ denotes for the log-complete data likelihood for SSPP. An exact EM algorithm refers to the case when one can make $q(\mathbf{x}_{0:K}) = p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\boldsymbol{\theta})$, and the lower bound equals to the marginal likelihood (for example, the EM for Gaussian mixture model and linear dynamical system). The formulations in Smith and Brown (2003) set $q(\mathbf{x}_{0:K}) = \prod_{k=0}^{K}\mathcal{N}(\mathbf{x}_{k|K},\boldsymbol{\Sigma}_{k|K})$ which is an approximation to $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\boldsymbol{\theta})$.

The approximate EM can be interpreted as:

$$\begin{aligned}\textbf{E-step:}\quad & \mathcal{Q}(q,\boldsymbol{\theta}) = \mathbb{E}_{q^{(l)}}[\ell(\mathbf{y}_{1:K},\mathbf{x}_{0:K}|\boldsymbol{\theta}^{(l)})], \\ \textbf{M-step:}\quad & \boldsymbol{\theta}^{(l+1)} = \arg\max_{\boldsymbol{\theta}}\ \mathcal{Q}(q^{(l)},\boldsymbol{\theta}).\end{aligned} \tag{3.24}$$

where $q^{(l)} = q(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta}^{(l)})$.

Define $\mathbf{P} = [\boldsymbol{\rho}_i, \cdots, \boldsymbol{\rho}_N]^{\mathrm{T}}$, $\mathbf{A} = [\boldsymbol{\alpha}_i, \cdots, \boldsymbol{\alpha}_N]^{\mathrm{T}}$ and $\boldsymbol{\Sigma}_\varepsilon = \mathrm{diag}(\sigma^2_{\varepsilon_1}, \cdots, \sigma^2_{\varepsilon_N})$ to be the state transition, input weight, and noise covariance matrices. The $\mathcal{Q}$-function has the following expression:

$$\begin{aligned}
\mathcal{Q}(q, \boldsymbol{\theta}) &= \mathbb{E}_q[\ell(\mathbf{y}_{1:K}, \mathbf{x}_{0:K}|\boldsymbol{\theta})] \\
&= \sum_{k=1}^{K}\sum_{c=1}^{C}\left(y_{c,k}\left(\mu_c + \boldsymbol{\beta}_c^{\mathrm{T}}\mathbf{x}_{k|K} + \ln\Delta\right) - \exp\left(\mu_c + \boldsymbol{\beta}_c^{\mathrm{T}}\mathbf{x}_{k|K} + \frac{1}{2}\boldsymbol{\beta}_c^{\mathrm{T}}\boldsymbol{\Sigma}_{k|K}\boldsymbol{\beta}_c\right)\Delta\right) \\
&\quad - \frac{K}{2}\log|\boldsymbol{\Sigma}_\varepsilon| - \frac{1}{2}\sum_{k=1}^{K}\mathbb{E}_q\left[(\mathbf{x}_k - \mathbf{P}\mathbf{x}_{k-1} - \mathbf{A}\mathbf{u}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_\varepsilon^{-1}(\mathbf{x}_k - \mathbf{P}\mathbf{x}_{k-1} - \mathbf{A}\mathbf{u}_k)\right] \\
&\quad - \frac{1}{2}\log|\Sigma_0| - \frac{1}{2}\mathbb{E}_q\left[(\mathbf{x}_0 - \boldsymbol{\pi}_0)^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}(\mathbf{x}_0 - \boldsymbol{\pi}_0)\right] + \mathrm{const}
\end{aligned} \tag{3.25}$$

The first line of the detailed expansion is from $\mathbb{E}_q[\ln p(\mathbf{y}_{1:K}|\mathbf{x}_{0:K}, \boldsymbol{\theta})]$, and the other terms are from $\mathbb{E}_q[\ln p(\mathbf{x}_{0:K}|\boldsymbol{\theta}_\dagger)]$. These expectations are taken with respect to $q(\mathbf{x}_{0:K})$.

At this point, it is worthy to emphasise the need for the approximate likelihood in equation (3.11) and functional form of CIF being exponential, with the following remark.

Remark 3.3. *If the Bernoulli likelihood in equation (3.7) with $\lambda_{c,k}\Delta = \sigma(\boldsymbol{\theta}_c^{\mathrm{T}}\boldsymbol{\phi}_k)$ is chosen. $\mathbb{E}[\ln p(\mathbf{y}_{1:K}|x_{0:K}, \boldsymbol{\theta})]$ requires computing expectation over sigmoid function. More specifically, a term $\mathbb{E}[\ln(1 + \exp(-\boldsymbol{\theta}_c^{\mathrm{T}}\boldsymbol{\phi}_k))]$ arises, to which exact evaluation is analytically intractable. In practice, as suggested by Saul et al. (1996), a lower bound is often employed to approximate this term:*

$$\begin{aligned}
\mathbb{E}_q[\ln(1 + \exp(-\boldsymbol{\theta}_c^{\mathrm{T}}\boldsymbol{\phi}_k))] \leq &- \xi_k\mathbb{E}_q[\boldsymbol{\theta}_c^{\mathrm{T}}\boldsymbol{\phi}_k] + \ln\left(\mathbb{E}_q[\exp(\xi_k\boldsymbol{\theta}_c^{\mathrm{T}}\boldsymbol{\phi}_k)]\right. \\
&\left.+\mathbb{E}_q[\exp((\xi_k - 1)\boldsymbol{\theta}_c^{\mathrm{T}}\boldsymbol{\phi}_k)]\right),
\end{aligned} \tag{3.26}$$

where, the parameter ξ_k controls the approximation quality and needs to be optimised. The optimisation has to be taken place at every time point, leading additional computational overheads. By contrast, the likelihood model in equation 3.11 with the exponential CIF, bypasses the above pathology, and therefore, acts as the primary setting in SSPP.

3.3.1 (E)xpectation-step

To evaluate the $\mathcal{Q}$-function, one needs the following statistics:

$$\mathbf{x}_{k|K} \equiv \mathbb{E}_q\left[\mathbf{x}_k\right], \quad \boldsymbol{\Sigma}_{k|K} \equiv \mathbb{E}_q\left[(\mathbf{x}_k - \mathbf{x}_{k|K})(\mathbf{x}_k - \mathbf{x}_{k|K})^{\mathrm{T}}\right], \tag{3.27}$$

$$\mathbf{W}_k \equiv \mathbb{E}_q\left[\mathbf{x}_k\mathbf{x}_k^{\mathrm{T}}\right], \quad \mathbf{W}_{k,k-1} \equiv \mathbb{E}_q\left[\mathbf{x}_k\mathbf{x}_{k-1}^{\mathrm{T}}\right]. \tag{3.28}$$

Equation (3.27-3.28) are obtained in the E-step of the approximate EM algorithm (Smith and Brown, 2003). The E-step is formed of a Laplace Gaussian filter to compute the

$\mathbf{x}_{k|k}$ and $\boldsymbol{\Sigma}_{k|k}$; a backward fixed interval smoothing (FIS) algorithm to compute the $\mathbf{x}_{k|K}$ and $\boldsymbol{\Sigma}_{k|K}$; and a state covariance algorithm to estimate $\mathbf{W}_k$ and $\mathbf{W}_{k,k-1}$.

The *Laplace Gaussian filter* (LGF) (Koyama et al., 2010), computes the Gaussian approximation to the filtering density $p(\mathbf{x}_k|\mathbf{y}_{1:k}, \boldsymbol{\theta})$ via the Laplace's method. Specifically, Taking the Taylor expansion on the $\ln p(\mathbf{x}_k|\mathbf{y}_{1:k}, \boldsymbol{\theta})$ around the peak point, denoting as $\mathbf{x}_{k|k}$, we have,

$$\ln p(\mathbf{x}_k|\mathbf{y}_{1:k}, \boldsymbol{\theta}) \simeq \ln p(\mathbf{x}_{k|k}|\mathbf{y}_{1:k}, \boldsymbol{\theta}) + (\mathbf{x}_k - \mathbf{x}_{k|k})^{\mathrm{T}} \mathbf{H}(\mathbf{x}_{k|k})(\mathbf{x}_k - \mathbf{x}_{k|k}) + \cdots , \tag{3.29}$$

where $\mathbf{H}(\mathbf{x}_{k|k}) = \frac{\partial^2 \ln p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \boldsymbol{\theta})}{\partial \mathbf{x}_k \partial \mathbf{x}_k^{\mathrm{T}}}|_{\mathbf{x}_k = \mathbf{x}_{k|k}}$ is the Hessian matrix. The quadratic term fits a Gaussian density $\mathcal{N}(\mathbf{x}_{k|k}, \boldsymbol{\Sigma}_{k|k})$ with $\boldsymbol{\Sigma}_{k|k} = -\mathbf{H}(\mathbf{x}_{k|k})^{-1}$. Such a Gaussian density is the approximated filtering density.

Based on the above principal, the LGF recursion for SSPP is shown in the following: $\forall k \in [1, \cdots, K]$, with the initial condition $\mathbf{x}_{0|0}$ and $\boldsymbol{\Sigma}_{0|0}$,

$$\mathbf{x}_{k|k-1} = \mathbf{P}\mathbf{x}_{k-1|k-1} + \mathbf{A}\mathbf{u}_k, \tag{3.30}$$

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{P}\boldsymbol{\Sigma}_{k-1|k-1}\mathbf{P}^{\mathrm{T}} + \boldsymbol{\Sigma}_{\epsilon}, \tag{3.31}$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \boldsymbol{\Sigma}_{k|k-1} \sum_{c=1}^{C} \boldsymbol{\beta}_c \left(y_{c,k} - \exp(\mu_c + \boldsymbol{\beta}_c^{\mathrm{T}} \mathbf{x}_{k|k}) \Delta \right), \tag{3.32}$$

$$\boldsymbol{\Sigma}_{k|k} = -\left(-\boldsymbol{\Sigma}_{k|k-1}^{-1} - \sum_{c=1}^{C} \boldsymbol{\beta}_c \exp(\mu_c + \boldsymbol{\beta}_c \mathbf{x}_{k|k}) \Delta \boldsymbol{\beta}_c^{\mathrm{T}} \right)^{-1} . \tag{3.33}$$

Note that equation (3.32) does not have close form solution. In practice, it can be solved by Newton's method or the fixed point iteration. When $\mathbf{x}_k$ is a high dimensional vector, the fixed point iteration is much preferred, since it doesn't involve computing the Hessian. Due to the fact that $p(\mathbf{x}_k|\mathbf{y}_{1:k}, \boldsymbol{\theta})$ is log-concave[3] in $\mathbf{x}_k$, both methods converges in small number of steps (typically 5 or 6).

Once the filtering density is assumed to be Gaussian, the Bayesian smoothing backward recursion will leads to the standard RTS smoother, regardless of the likelihood model. Specifically, with initial conditions $\mathbf{x}_{K|K}$ and $\boldsymbol{\Sigma}_{K|K}$, for $k = K - 1$ to 1,

$$\mathbf{J}_k = \mathbf{P}^{\mathrm{T}} \boldsymbol{\Sigma}_{k|k} \boldsymbol{\Sigma}_{k+1|k}^{-1}, \tag{3.34}$$

$$\mathbf{x}_{k|K} = \mathbf{x}_{k|k} + \mathbf{J}_k (\mathbf{x}_{k+1|K} - \mathbf{x}_{k+1|k}), \tag{3.35}$$

$$\boldsymbol{\Sigma}_{k|K} = \boldsymbol{\Sigma}_{k|k} + \mathbf{J}_k \left(\boldsymbol{\Sigma}_{k+1|K} - \boldsymbol{\Sigma}_{k+1|k} \right) \mathbf{J}_k^{\mathrm{T}}. \tag{3.36}$$

[3] The log-concavity of $p(\mathbf{x}_k|\mathbf{y}_{1:k}, \boldsymbol{\theta})$ will be formally established in the later section in this chapter.

The remaining terms $\mathbf{W}_k$ and $\mathbf{W}_{k,k-1}$ can be computed as the following:

$$\boldsymbol{\Sigma}_{k,k-1|K} = \mathbf{J}_k \boldsymbol{\Sigma}_{k-1|K}, \tag{3.37}$$

$$\mathbf{W}_k = \boldsymbol{\Sigma}_{k|K} + \mathbf{x}_{k|K}\mathbf{x}_{k|K}^{\mathrm{T}}, \tag{3.38}$$

$$\mathbf{W}_{k,k-1} = \boldsymbol{\Sigma}_{k,k-1|K} + \mathbf{x}_{k|K}\mathbf{x}_{k-1|K}^{\mathrm{T}}. \tag{3.39}$$

3.3.2 (M)aximisation-step

In the M-step, each parameter is estimated by solving,

$$\frac{\partial \mathcal{Q}(q,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0. \tag{3.40}$$

Explicitly, it yields the following estimates:

- The state transition matrix and input weight matrix:

$$\begin{bmatrix}\mathbf{P}^{(l+1)} & \mathbf{A}^{(l+1)}\end{bmatrix} = \sum_{k=1}^{K}\begin{bmatrix}\mathbf{W}_{k,k-1} & \mathbf{x}_{k|K}\mathbf{u}_k^{\mathrm{T}}\end{bmatrix}\left(\sum_{k=1}^{K}\begin{bmatrix}\mathbf{W}_{k-1} & \mathbf{x}_{k-1|K}\mathbf{u}_k^{\mathrm{T}} \\ \mathbf{u}_k\mathbf{x}_{k-1|K}^{\mathrm{T}} & \mathbf{u}_k\mathbf{u}_k^{\mathrm{T}}\end{bmatrix}\right)^{-1}. \tag{3.41}$$

- The noise covariance matrix:

$$\boldsymbol{\Sigma}_{\varepsilon}^{(l+1)} = \frac{1}{K}\left(\mathbf{W}_k - \mathbf{P}^{(l+1)}\mathbf{W}_{k,k-1}^{\mathrm{T}} - \mathbf{A}^{(l+1)}\mathbf{u}_k\mathbf{x}_{k|K}^{\mathrm{T}}\right). \tag{3.42}$$

  Alternatively, one can directly compute the diagonal terms:

$$\sigma_{\varepsilon_i}^{2\ (l+1)} = \frac{1}{K}\left(w_{i,i,k} - \left(\boldsymbol{\rho}_i^{(l+1)}\right)^{\mathrm{T}}\mathbf{w}_{i,k,k-1} - \left(\boldsymbol{\alpha}_i^{(l+1)}\right)^{\mathrm{T}}\mathbf{u}_k x_{i,k|K}\right), \tag{3.43}$$

  where, $w_{i,i,k}$ is the $(i,i)^{\mathrm{th}}$ element of $\mathbf{W}_k$, and $\mathbf{w}_{i,k,k-1}^{\mathrm{T}}$ is the i^{th} row of $\mathbf{W}_{k,k-1}$.

- Background firing rate and state weight vector:

  Unfortunately, μ_c and $\boldsymbol{\beta}_c$ do not have closed-form solutions. Instead, we use the Newton's iterations. Let $f(\boldsymbol{\theta}_c) = \mathbb{E}_q[\ln p(\mathbf{y}_{1:K}|\mathbf{x}_{1:K},\boldsymbol{\theta}_{1:C})]$ and r denotes the index to Newton step.

$$\boldsymbol{\theta}_c^{(r+1)} = \boldsymbol{\theta}_c^{(r)} - \mathbf{H}\left(\boldsymbol{\theta}_c^{(r)}\right)^{-1}\mathbf{g}\left(\boldsymbol{\theta}_c^{(r)}\right), \tag{3.44}$$

  where the gradient,

$$\mathbf{g}\left(\boldsymbol{\theta}_c^{(r)}\right) = \frac{\partial f(\boldsymbol{\theta}_c)}{\partial \boldsymbol{\theta}_c}\Big|_{\boldsymbol{\theta}_c=\boldsymbol{\theta}_c^{(r)}} = \begin{bmatrix} g(\mu_c) \\ \mathbf{g}(\boldsymbol{\beta}_c)\end{bmatrix}, \tag{3.45}$$

with,

$$g\left(\mu_c^{(r)}\right) = \sum_{k=1}^{K}\left(y_{c,k} - \exp\left(\mu_c^{(r)} + \left(\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\mathbf{x}_{k|K} + \frac{1}{2}\left(\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_{k|K}\boldsymbol{\beta}_c^{(r)}\right)\Delta\right), \tag{3.46}$$

$$\begin{aligned}\mathbf{g}\left(\boldsymbol{\beta}_c^{(r)}\right) = \sum_{k=1}^{K}\Bigg(&\mathbf{x}_{k|K}y_{c,k} - \left(\mathbf{x}_{k|K} + \boldsymbol{\Sigma}_{k|K}\boldsymbol{\beta}_c^{(r)}\right)\exp\Bigg(\mu_c^{(r)} + \left(\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\mathbf{x}_{k|K}\\ &+\frac{1}{2}\left(\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_{k|K}\boldsymbol{\beta}_c^{(r)}\Bigg)\Delta\Bigg).\end{aligned} \tag{3.47}$$

And the Hessian,

$$\mathbf{H}\left(\boldsymbol{\theta}_c^{(r)}\right) = \frac{\partial^2 f(\boldsymbol{\theta}_c)}{\partial\boldsymbol{\theta}_c\partial\boldsymbol{\theta}_c^{\mathrm{T}}}\Big|_{\boldsymbol{\theta}_c=\boldsymbol{\theta}_c^{(r)}} = \begin{bmatrix} h\left(\mu_c^{(r)}\right) & \mathbf{h}\left(\mu_c^{(r)},\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}} \\ \mathbf{h}\left(\mu_c^{(r)},\boldsymbol{\beta}_c^{(r)}\right) & \mathbf{H}\left(\boldsymbol{\beta}_c^{(r)}\right)\end{bmatrix}, \tag{3.48}$$

with,

$$h\left(\mu_c^{(r)}\right) = -\sum_{k=1}^{K}\exp\left(\mu_c^{(r)} + \left(\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\mathbf{x}_{k|K} + \frac{1}{2}\left(\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_{k|K}\boldsymbol{\beta}_c^{(r)}\right)\Delta, \tag{3.49}$$

$$\begin{aligned}\mathbf{h}\left(\mu_c^{(r)},\boldsymbol{\beta}_c^{(r)}\right) = -\sum_{k=1}^{K}&\left(\mathbf{x}_{k|K} + \boldsymbol{\Sigma}_{k|K}\boldsymbol{\beta}_c^{(r)}\right)\exp\Bigg(\mu_c^{(r)} + \left(\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\mathbf{x}_{k|K}\\ &+\frac{1}{2}\left(\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_{k|K}\boldsymbol{\beta}_c^{(r)}\Bigg)\Delta,\end{aligned} \tag{3.50}$$

$$\begin{aligned}\mathbf{H}\left(\boldsymbol{\beta}_c^{(r)}\right) = -\sum_{k=1}^{K}\Bigg(&\boldsymbol{\Sigma}_{k|K}\exp\left(\mu_c^{(r)} + \left(\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\mathbf{x}_{k|K} + \frac{1}{2}\left(\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_{k|K}\boldsymbol{\beta}_c^{(r)}\right)\Delta\\ &+\left(\mathbf{x}_{k|K} + \boldsymbol{\Sigma}_{k|K}\boldsymbol{\beta}_c^{(r)}\right)\exp\Bigg(\mu_c^{(r)} + \left(\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\mathbf{x}_{k|K}\\ &+\frac{1}{2}\left(\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_{k|K}\boldsymbol{\beta}_c^{(r)}\Bigg)\Delta\left(\mathbf{x}_{k|K} + \boldsymbol{\Sigma}_{k|K}\boldsymbol{\beta}_c^{(r)}\right)^{\mathrm{T}}\Bigg).\end{aligned} \tag{3.51}$$

- Initial state mean and covariance

$$\boldsymbol{\pi}_0 = \mathbf{x}_{0|K}, \qquad \boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_{0|K}. \tag{3.52}$$

Figure 3.10 and 3.11 illustrate typical result of the approximate EM, with a dataset generated by example 3.2.

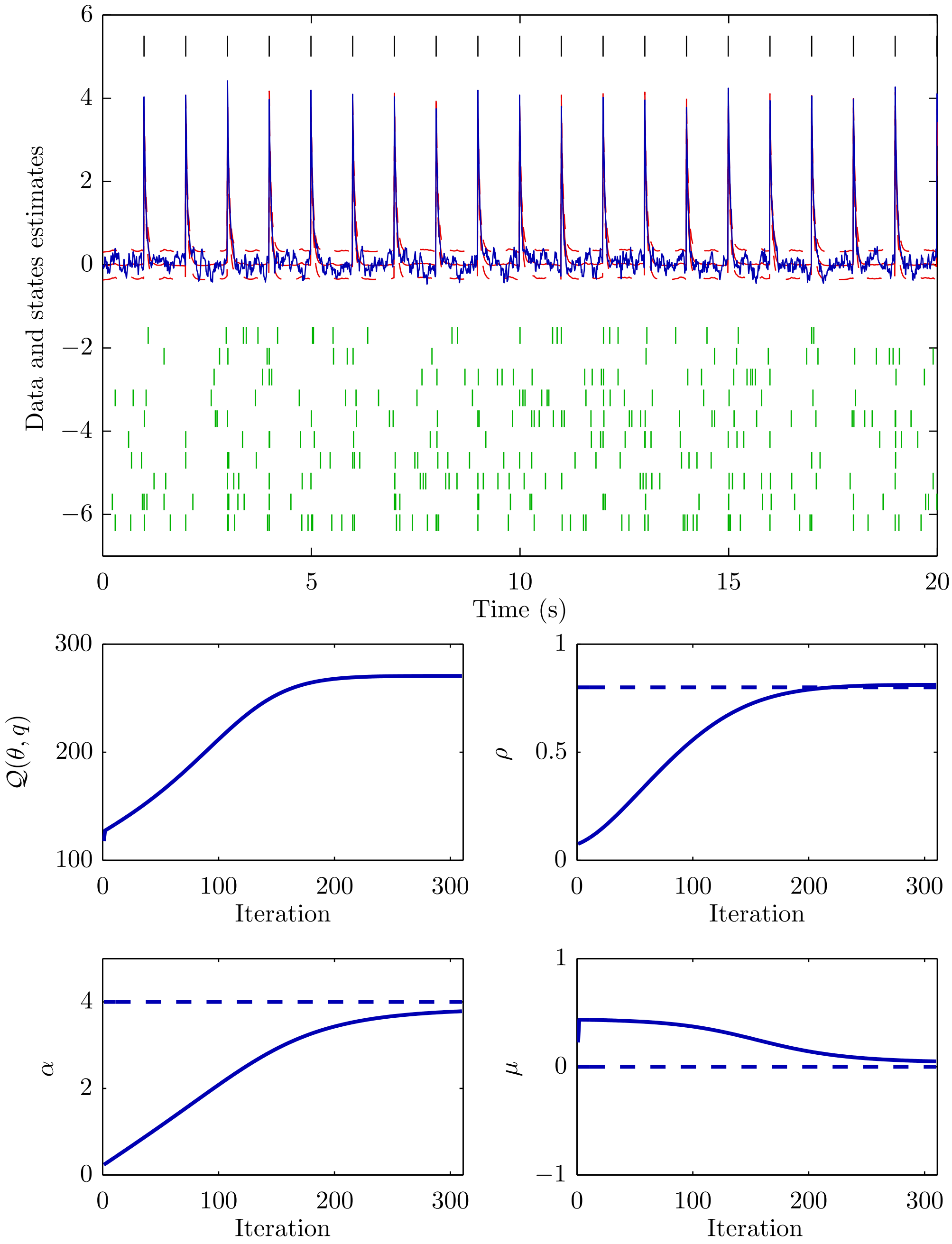


Figure 3.10: Results on states and parameter estimation via EM algorithm, with σ_ε^2 and β_c fixed to their true values. *Top*: Inputs as *black bars* scaled for visualisation and true states as *blue solid line*, observed 10-channel binary sequences as *green bars*, the mode of the smoothing density $x_{k|K}$ (*red solid line*) and 95% confident intervals (*red dashed lines*) computed as $x_{k|K} \pm 1.96\sigma_{k|K}$. *Bottom panels*, ordering from *left* to *right* are the learning curves of $\mathcal{Q}$-function, ρ, α and μ, in which the *dashed lines* denote for true values.

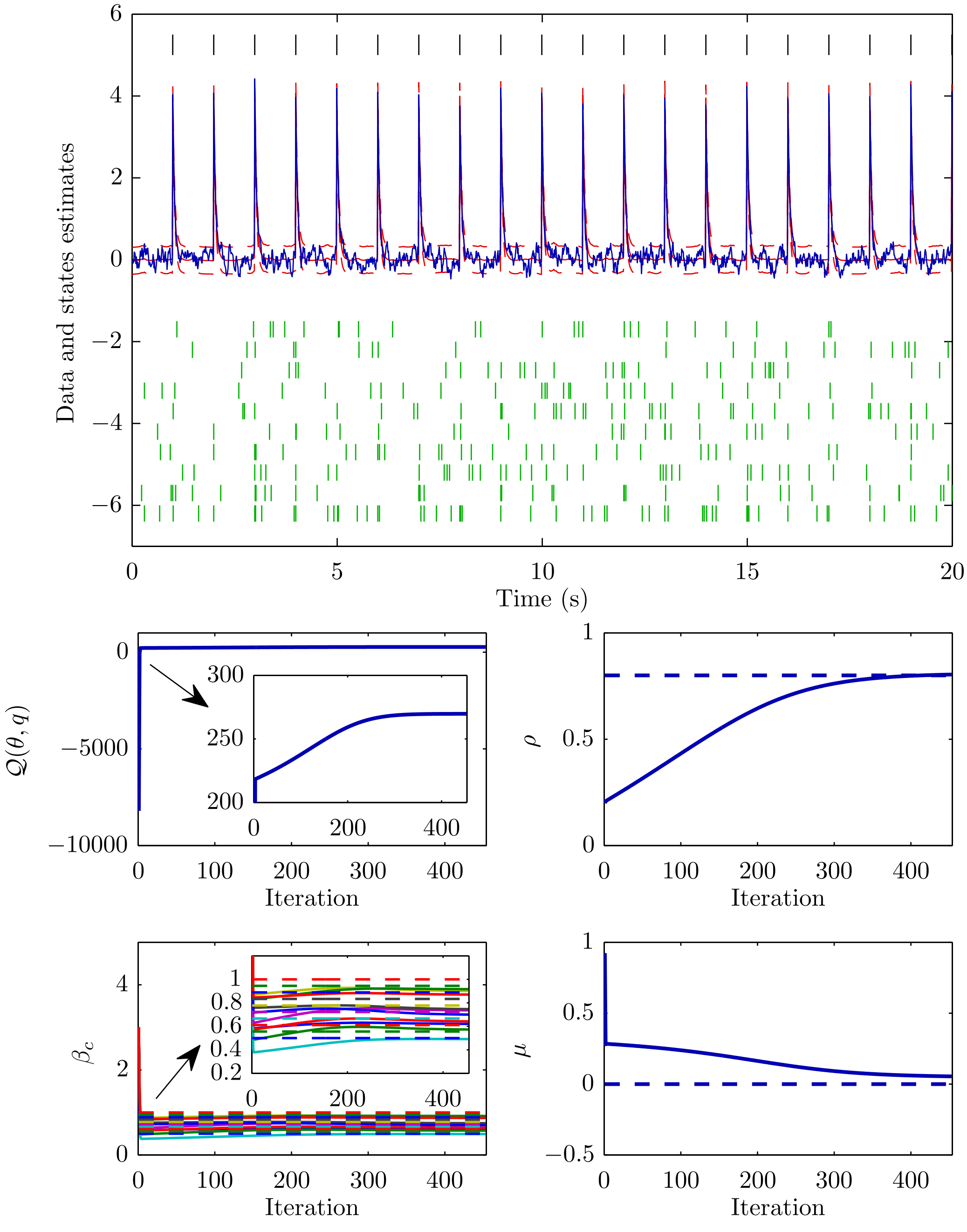


Figure 3.11: Results on states and parameter estimation via EM algorithm, with σ_ε^2 and α fixed to their true values. *Top*: Inputs as *black bars* scaled for visualisation and true states as *blue solid line*, observed 10-channel binary sequences as *green bars*, the mode of the smoothing density $x_{k|K}$ (*red solid line*) and 95% confident intervals (*red dashed lines*) computed as $x_{k|K} \pm 1.96\sigma_{k|K}$. *Bottom panels*, ordering from *left* to *right* are the learning curves of $\mathcal{Q}$-function, ρ, β_c and μ, in which the *dashed lines* denote for true values.

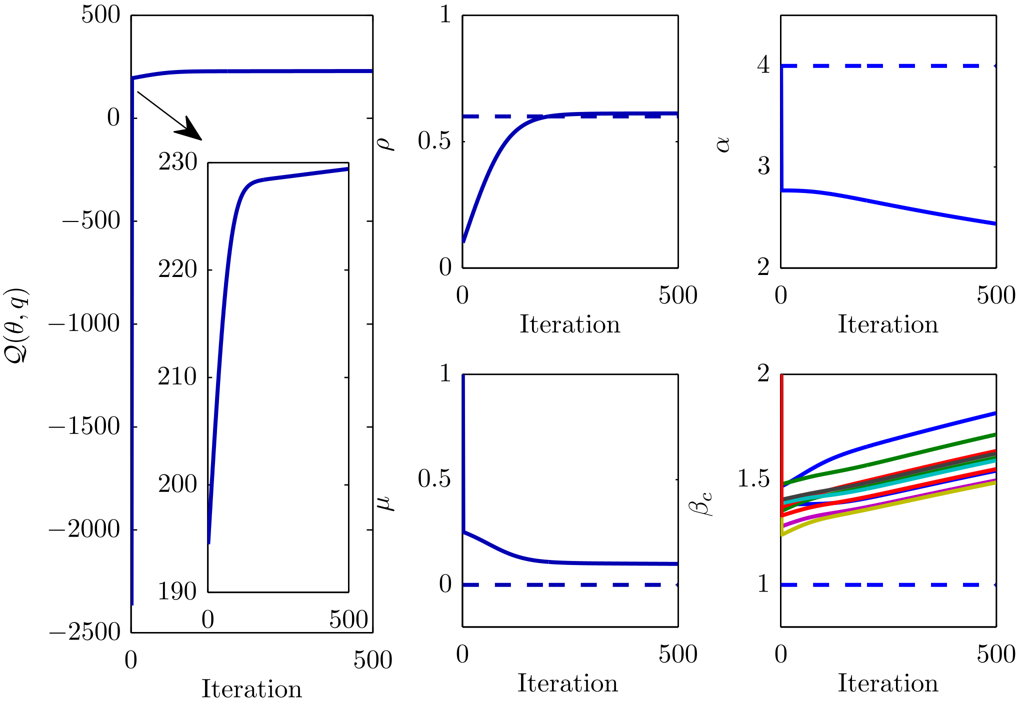


Figure 3.12: Illustration of the identifiability problem in SSPP. Convergence of $\mathcal{Q}(q, \boldsymbol{\theta})$, ρ, α, μ and β_c. Note that α and β_c compensate each other while keeping $\mathcal{Q} - (q, \boldsymbol{\theta})$ stable.

3.3.3 Identifiability

Although we have the formulas for learning each of the five parameters, the model is generally over-parameterised. This arises from the fact that parameter $\boldsymbol{\beta}_c$ appears in the likelihood only via the product $\boldsymbol{\beta}_c^{\mathrm{T}}\mathbf{x}_k$, and the term $\mathbf{A}$. This makes $\mathbf{A}$ and $\boldsymbol{\beta}_c$ difficult to estimate jointly even the hidden states are considered as scalars. To see this, we construct a dataset using the following parameter setting:

Example 3.3. *In time interval $T = 20s$, we set the parameters of the model to be: $\rho = 0.8$, $\alpha = 4$, $\sigma_\epsilon^2 = 10^{-2}$, and $x_0 = 0$ with the spike input u_k fires at every $1s$ interval. With $\mu_c = 0$, $\beta_c = 1$ $\forall c$, and time resolution: $\Delta = 10ms$, 10 channels of binary data is simulated.*

Figure 3.12 clearly demonstrates the identifiability problem, based on a realisation of example 3.3. In practice, to deal with this problem, it is preferred to fix one and estimate the other (e.g. figure 3.10 and 3.11). When $\boldsymbol{\beta}_c$ is fixed the formula for μ_c becomes available in closed-form:

$$\mu_c = \ln\left(\sum_{k=1}^{K} y_{c,k}\right) - \ln\left(\sum_{k=1}^{K} \exp\left(\boldsymbol{\beta}_c^{\mathrm{T}}\mathbf{x}_{k|K} + \frac{1}{2}\boldsymbol{\beta}_c^{\mathrm{T}}\boldsymbol{\Sigma}_{k|K}\boldsymbol{\beta}_c\right)\Delta\right) \tag{3.53}$$

3.4 Analysis of EM

3.4.1 Log-concavity of the filtering and smoothing densities

As $q(\mathbf{x}_{0:K})$ is an approximation. Investigating the shape of the true filtering and smoothing distributions is vital to understand how close they are to the approximation quality. For this, we establish that both the optimal filtering and smoothing density in SSPP are log-concave. As a result, the Gaussian approximations produced by LGF are appropriate.

The analysis outline is the following: First, with a technical theorem we identify the conditions for an arbitrary state-space model's filtering and smoothing density to be log-concave. Then, we show that SSPP satisfies these conditions, and hence, has log-concave filtering and smoothing densities.

Firstly, recall the definition of the concave function:

Definition 3.4. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is concave if the domain of f is a convex set and if for all $\mathbf{x}, \mathbf{y} \in$ the domain of f, and θ with $0 < \theta < 1$, we have*

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) < \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y}). \tag{3.54}$$

A useful tool for checking whether a function is concave is the second order condition. That is, if a function f is twice differentiable, then f is concave if and only if the domain of f is convex and its Hessian is negative semidefinite: $\frac{\partial^2 f(\mathbf{x})}{\partial\mathbf{x}\partial\mathbf{x}^{\mathrm{T}}} \preceq 0$.

Theorem 3.5. (Prékopa, 1973) *Let $f(\mathbf{x}, \mathbf{y})$ be a log-concave function on $\mathbb{R}^n \times \mathbb{R}^m$, then*

$$g(\mathbf{x}) = \int_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}) d\mathbf{y} \tag{3.55}$$

is a log-concave function in $\mathbf{x}$ on $\mathbb{R}^n$.

Proof. The proof can be found in Prékopa (1973). □

Theorem 3.6. *For an arbitrary state-space model with state transition distribution $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, observation distribution $p(\mathbf{y}_k|\mathbf{x}_k)$, $\forall k \in [1, \cdots, K]$, and a initial state distribution $p(\mathbf{x}_0)$, where $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{y}_k \in \mathbb{R}^m$. If the following conditions hold,*

1. *$p(\mathbf{x}_0)$ is log-concave in $\mathbf{x}_0$ on $\mathbb{R}^n$,*
2. *$p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is log-concave in $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ on $\mathbb{R}^n \times \mathbb{R}^n$,*
3. *$p(\mathbf{y}_k|\mathbf{x}_k)$ is log-concave in $\mathbf{x}_k$ on $\mathbb{R}^n$.*

Then, the filtering distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is log-concave in $\mathbf{x}_k$ on $\mathbb{R}^n$, $\forall k \in [1, \cdots, K]$.

Proof. Recall that, the filtering density is expressed as:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{k-1})}, \tag{3.56}$$

where,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int_{\mathbf{x}_{k-1}} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1}, \tag{3.57}$$

$$p(\mathbf{y}_k|\mathbf{y}_{k-1}) = \int_{\mathbf{x}_k} p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k. \tag{3.58}$$

Since, $p(\mathbf{y}_k|\mathbf{y}_{k-1})$ is not a function of $\mathbf{x}_k$, the log-concavity of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is purely depends on $p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$. Consider the case $k = 1$, we have

$$p(\mathbf{x}_1|\mathbf{y}_1) \propto p(\mathbf{y}_1|\mathbf{x}_1) \int_{\mathbf{x}_0} p(\mathbf{x}_1|\mathbf{x}_0)p(\mathbf{x}_0)d\mathbf{x}_0. \tag{3.59}$$

Given the condition of $p(\mathbf{x}_1|\mathbf{x}_0)$ and $p(\mathbf{x}_0)$ being log-concave on $\mathbb{R}^n \times \mathbb{R}^n$ and $\mathbb{R}^n$, respectively, $p(\mathbf{x}_1, \mathbf{x}_0) = p(\mathbf{x}_1|\mathbf{x}_0)p(\mathbf{x}_0)$ is log-concave on $\mathbb{R}^n \times \mathbb{R}^n$. Then apply theorem 3.5, we obtain $p(\mathbf{x}_1) = \int_{\mathbf{x}_0} p(\mathbf{x}_1|\mathbf{x}_0)p(\mathbf{x}_0)d\mathbf{x}_0$ is log-concave on $\mathbb{R}^n$. Use the other condition, $p(\mathbf{y}_1|\mathbf{x}_1)$ being log-concave, and due the fact that product of log-concave function preserves log-concavity. We conclude that $p(\mathbf{x}_1|\mathbf{y}_1)$ is log-concave.

Now, consider $k = 2$, we have

$$p(\mathbf{x}_2|\mathbf{y}_{1:2}) \propto p(\mathbf{y}_2|\mathbf{x}_2) \int_{\mathbf{x}_1} p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_1|\mathbf{y}_1)d\mathbf{x}_1. \tag{3.60}$$

Similarly, use the conditions: $p(\mathbf{x}_2|\mathbf{x}_1)$ is log-concave on $\mathbb{R}^n \times \mathbb{R}^n$, and $p(\mathbf{x}_1|\mathbf{y}_1)$ is log-concave in $\mathbf{x}_1$ on $\mathbb{R}^n$, we obtain that $p(\mathbf{x}_2, \mathbf{x}_1|\mathbf{y}_1)$ is log-concave on $\mathbb{R}^n \times \mathbb{R}^n$. Then using theorem 3.5, we have $p(\mathbf{x}_2|\mathbf{y}_1)$ is log-concave in $\mathbf{x}_2$ on $\mathbb{R}^n$. Finally, due to the face $p(\mathbf{y}_2|\mathbf{x}_2)$ is log-concave in $\mathbf{x}_2$ on $\mathbb{R}^n$, we conclude that $p(\mathbf{x}_2|\mathbf{y}_{1:2})$ is log-concave in $\mathbf{x}_2$ on $\mathbb{R}^n$.

Use the same proof, we can easily establish that $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is log-concave in $\mathbf{x}_k$ on $\mathbb{R}^n$, $\forall k$. This completes the proof. □

Theorem 3.7. *For an arbitrary state space model with $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{y}_k|\mathbf{x}_k)$, $\forall k \in [1, \cdots, K]$, and a initial state distribution $p(\mathbf{x}_0)$, where $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{y}_k \in \mathbb{R}^m$. If the following conditions holds,*

1. *$p(\mathbf{x}_0)$ is log-concave in $\mathbf{x}_0$ on $\mathbb{R}^n$,*
2. *$p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is log-concave in $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ on $\mathbb{R}^n \times \mathbb{R}^n$,*
3. *$p(\mathbf{y}_k|\mathbf{x}_k)$ is log-concave in $\mathbf{x}_k$ on $\mathbb{R}^n$.*

Then, the smoothing density $p(\mathbf{x}_k|\mathbf{y}_{1:K})$ is log-concave $\forall k \in [1, \cdots, K]$.

Proof. Recall that the Bayesian smoothing density is

$$p(\mathbf{x}_k|\mathbf{y}_{1:K}) = p(\mathbf{x}_k|\mathbf{y}_{1:k}) \int_{\mathbf{x}_{k+1}} \frac{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:K})p(\mathbf{x}_{k+1}|\mathbf{x}_k)}{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})} d\mathbf{x}_{k+1}. \tag{3.61}$$

Consider the case $k = K - 1$, the smoothing density becomes

$$p(\mathbf{x}_{K-1}|\mathbf{y}_{1:K}) = p(\mathbf{x}_{K-1}|\mathbf{y}_{1:K-1}) \int_{\mathbf{x}_K} \frac{p(\mathbf{x}_K|\mathbf{y}_{1:K})p(\mathbf{x}_K|\mathbf{x}_{K-1})}{p(\mathbf{x}_K|\mathbf{y}_{1:K-1})} d\mathbf{x}_K. \tag{3.62}$$

The denominator in the integral can be rewritten as:

$$p(\mathbf{x}_K|\mathbf{y}_{1:K-1}) = \frac{p(\mathbf{x}_K|\mathbf{y}_{1:K})p(\mathbf{y}_K|\mathbf{y}_{1:K-1})}{p(\mathbf{y}_K|\mathbf{x}_K)}. \tag{3.63}$$

Substitute back into the integral, we have

$$\begin{aligned} p(\mathbf{x}_{K-1}|\mathbf{y}_{1:K}) &= p(\mathbf{x}_{K-1}|\mathbf{y}_{1:K-1}) \int_{\mathbf{x}_K} \frac{p(\mathbf{y}_K|\mathbf{x}_K)p(\mathbf{x}_K|\mathbf{x}_{K-1})}{p(\mathbf{y}_K|\mathbf{y}_{1:K-1})} d\mathbf{x}_K & (3.64) \\ &= p(\mathbf{x}_{K-1}|\mathbf{y}_{1:K-1}) \frac{p(\mathbf{y}_K|\mathbf{x}_{K-1})}{p(\mathbf{y}_K|\mathbf{y}_{1:K-1})}, & (3.65) \end{aligned}$$

where $p(\mathbf{x}_{K-1}|\mathbf{y}_{1:K-1})$ according to theorem 3.6 is log-concave in $\mathbf{x}_{K-1}$ on $\mathbb{R}^N$, given the conditions. Apply theorem 3.5, $p(\mathbf{y}_K|\mathbf{x}_{K-1})$ is log-concave in $\mathbf{x}_{K-1}$ on $\mathbb{R}^N$. And, with $p(\mathbf{y}_K|\mathbf{y}_{1:K-1})$ is not a function of $\mathbf{x}_K$, these facts establish that $p(\mathbf{x}_{K-1}|\mathbf{y}_{1:K})$ is log-concave in $\mathbf{x}_K$ on $\mathbb{R}^N$.

Consider the cases $k = K-2, \cdots, 1$, we can rewrite $p(\mathbf{x}_{k+1}|\mathbf{y}_{1:K})$ and $p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})$ with Bayes' rule,

$$p(\mathbf{x}_{k+1}|\mathbf{y}_{1:K}) = p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k+1}, \mathbf{y}_{k+2:K}) = \frac{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k+1})p(\mathbf{y}_{k+2:K}|\mathbf{y}_{1:k+1})}{p(\mathbf{y}_{k+2:K})}, \tag{3.66}$$

$$p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) = \frac{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k+1})p(\mathbf{y}_{k+1}|\mathbf{y}_{1:k})}{p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})}. \tag{3.67}$$

Substitute equation (3.66) and (3.67) into equation (3.61), we have

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{y}_{1:K}) &= p(\mathbf{x}_k|\mathbf{y}_{1:k}) \int_{\mathbf{x}_{k+1}} b p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}) d\mathbf{x}_{k+1} & (3.68) \\ &= p(\mathbf{x}_k|\mathbf{y}_{1:k}) b p(\mathbf{y}_{k+1}|\mathbf{x}_k), & (3.69) \end{aligned}$$

where $b = \frac{p(\mathbf{y}_{k+2:K}|\mathbf{y}_{1:k+1})}{p(\mathbf{y}_{k+2:K})p(\mathbf{y}_{k+1}|\mathbf{y}_{1:k})}$ is not a function of $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$. Use the given conditions and theorem 3.6, we have $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ being log-concave in $\mathbf{x}_k$ on $\mathbb{R}^N$. With the given conditions and theorem 3.5, we have $p(\mathbf{y}_{k+1}|\mathbf{x}_k)$ being log-concave in $\mathbf{x}_k$ on $\mathbb{R}^N$. Hence, the product $p(\mathbf{x}_k|\mathbf{y}_{1:K})$ is also log-concave in $\mathbf{x}_k$ on $\mathbb{R}^N$.

Finally, when $k = K$, $p(\mathbf{x}_K|\mathbf{y}_K)$ as a filtering density is log-concave in $\mathbf{x}_K$ on $\mathbb{R}^N$. This completes the proof. □

Remark 3.8. *Both theorem 3.6 and 3.7 hold based on the model parameters being given.*

Remark 3.9. *For state-space models with $\mathbf{x}_1$ as the initial state, both theorem 3.6 and 3.7 holds, by changing the condition 1 in both theorems to be $p(\mathbf{x}_1)$ is log-concave in $\mathbf{x}_1$ on $\mathbb{R}^N$.*

Theorem 3.10. *The filtering density $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ and the smoothing density $p(\mathbf{x}_k|\mathbf{y}_{1:K})$ in SSPP are both log-concave in $\mathbf{x}_k$ on $\mathbb{R}^N$, $\forall k \in [1, \cdots, K]$.*

Proof. The initial states density in SSPP is $\mathcal{N}(\mathbf{x}_0|\boldsymbol{\pi}_0, \boldsymbol{\Sigma}_0)$, which log-concave in $\mathbf{x}_0$ on $\mathbb{R}^N$. Due to theorem 3.1, the observation model $p(\mathbf{y}_k|\mathbf{x}_k)$ is log-concave in $\mathbf{x}_k$ on $\mathbb{R}^N$. The state transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ has the expression

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) \propto \exp\left(-\frac{1}{2}(\mathbf{x}_k - \mathbf{P}\mathbf{x}_{k-1} - \mathbf{A}\mathbf{u}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_{\varepsilon}^{-1}(\mathbf{x}_k - \mathbf{P}\mathbf{x}_{k-1} - \mathbf{A}\mathbf{u}_k)\right). \tag{3.70}$$

Let $\mathbf{z} = [\mathbf{x}_k^{\mathrm{T}}, \mathbf{x}_{k-1}^{\mathrm{T}}]^{\mathrm{T}}$, write $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ as $p(\mathbf{z})$, then the Hessian matrix of $\ln p(\mathbf{z})$ is

$$\mathbf{H}(\mathbf{z}) = \begin{bmatrix} \mathbf{H}(\mathbf{x}_k) & \mathbf{H}(\mathbf{x}_k, \mathbf{x}_{k-1}) \\ \mathbf{H}(\mathbf{x}_k, \mathbf{x}_{k-1})^{\mathrm{T}} & \mathbf{H}(\mathbf{x}_{k-1}) \end{bmatrix}. \tag{3.71}$$

where,

$$\mathbf{H}(\mathbf{x}_k) = -\boldsymbol{\Sigma}_{\varepsilon}^{-1}, \quad \mathbf{H}(\mathbf{x}_k, \mathbf{x}_{k-1}) = \boldsymbol{\Sigma}_{\varepsilon}^{-1}\mathbf{P}, \quad \mathbf{H}(\mathbf{x}_{k-1}) = -\mathbf{P}^{\mathrm{T}}\boldsymbol{\Sigma}_{\varepsilon}^{-1}\mathbf{P}. \tag{3.72}$$

Then, let $\mathbf{v} = [\mathbf{v}_1^{\mathrm{T}}, \mathbf{v}_2^{\mathrm{T}}]^{\mathrm{T}}$, in which $\mathbf{v}$ is a arbitrary $2n$-dimensional nonzero vector, $\mathbf{v}_1$ and $\mathbf{v}_2$ are both n-dimensional nonzero vectors. We have,

$$\begin{aligned} \mathbf{v}^{\mathrm{T}}\mathbf{H}(\mathbf{z})\mathbf{v} &= \begin{bmatrix} \mathbf{v}_1^{\mathrm{T}} & \mathbf{v}_2^{\mathrm{T}} \end{bmatrix} \mathbf{H}(\mathbf{z}) \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} && (3.73) \\ &= -\mathbf{v}_1^{\mathrm{T}}\boldsymbol{\Sigma}_{\varepsilon}^{-1}\mathbf{v}_1 + \mathbf{v}_2^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\left(\boldsymbol{\Sigma}_{\varepsilon}^{-1}\right)^{\mathrm{T}}\mathbf{v}_1 + \mathbf{v}_1^{\mathrm{T}}\boldsymbol{\Sigma}_{\varepsilon}^{-1}\mathbf{P}\mathbf{v}_2 - \mathbf{v}_2^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\boldsymbol{\Sigma}_{\varepsilon}^{-1}\mathbf{P}\mathbf{v}_2 && (3.74) \\ &= -(\mathbf{v}_1 - \mathbf{P}\mathbf{v}_2)^{\mathrm{T}}\boldsymbol{\Sigma}_{\varepsilon}^{-1}(\mathbf{v}_1 - \mathbf{P}\mathbf{v}_2) < 0 && (3.75) \end{aligned}$$

Hence, $\ln p(\mathbf{z})$ is concave on $\mathbb{R}^N \times \mathbb{R}^N$, so does $\ln p(\mathbf{x}_k|\mathbf{x}_{k-1})$. Subsequently, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is log-concave on $\mathbb{R}^N \times \mathbb{R}^N$. Now, apply theorem 3.6 and 3.7, we have that, in SSPP $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ and $p(\mathbf{x}_k|\mathbf{y}_{1:K})$ are both log-concave in $\mathbf{x}_k$ on $\mathbb{R}^N$ $\forall k \in [1, \cdots, K]$, as desired. □

Now, we have formally establish that both the filtering and smoothing densities are log-concave. As a result, both the filtering and smoothing densities have single mode. In this regard, the Gaussian approximations in the E-step are likely to be appropriate; in the sense that, they cover a relative large region of the true density. However, if the true density is heavy tailed, Gaussian approximate might perform poorly.

3.4.2 $\mathcal{Q}$-function: Theoretical landscapes

The EM algorithm is known to converge to a local optimum closest to the initial guess of the parameters (Wu, 1983). Hence the landscape of the likelihood is an important consideration in the application of EM based algorithms. Usually, when the likelihood is multimodal, it is desirable to resort to Monte Carlo sampling methods, rather than a point estimate of parameters for this reason. Here, we study the shape of $\mathcal{Q}$-function theoretically.

Lemma 3.11. *Given $\mathbf{x}_{0:K}$, $\mathbf{y}_{1:K}$ and $\boldsymbol{\Sigma}_\varepsilon$, rewrite the log-complete data likelihood as $\ell(\boldsymbol{\theta}_*)$, where $\boldsymbol{\theta}_* = (\boldsymbol{\pi}_0, \{\boldsymbol{\rho}_i, \boldsymbol{\alpha}_i\}, \{\mu_c, \boldsymbol{\beta}_c\})$. Then $\ell(\boldsymbol{\theta}_*)$ is concave in $\boldsymbol{\theta}_*$ on its domain.*

Proof. First, let $\ell_0(\boldsymbol{\pi}_0) = \ln p(\mathbf{x}_0|\boldsymbol{\pi}_0, \boldsymbol{\Sigma}_0)$ when $\mathbf{x}_0$ and $\boldsymbol{\Sigma}_0$ are given. $\ell(\theta_*)$ can be write as:

$$\ell(\boldsymbol{\theta}_*) = \ell(\boldsymbol{\pi}_0) + \sum_{k=1}^{K} \ell_k\left(\{\boldsymbol{\rho}_i\}, \{\boldsymbol{\alpha}_i\}, \{\mu_c\}, \{\boldsymbol{\beta}_c\}\right), \tag{3.76}$$

where,

$$\ell_k\left(\{\boldsymbol{\rho}_i\}, \{\boldsymbol{\alpha}_i\}, \{\mu_c\}, \{\boldsymbol{\beta}_c\}\right) = \sum_{i=1}^{N} \ell_{x_{i,k}}(\boldsymbol{\rho}_i, \boldsymbol{\alpha}_i) + \sum_{c=1}^{C} \ell_{y_{c,k}}(\mu_c, \boldsymbol{\beta}_c). \tag{3.77}$$

with $\ell_{x_{i,k}}(\boldsymbol{\rho}_i, \boldsymbol{\alpha}_i) = \ln p(x_{i,k}|\mathbf{x}_{k-1}, \boldsymbol{\theta}_i)$ and $\ell_{y_{c,k}}(\mu_c, \boldsymbol{\beta}_c) = \ln p(y_{c,k}|\mathbf{x}_k, \mu_c, \boldsymbol{\beta}_c)$.

Now, given the fact that, summation with positive weights preserves concavity, and summation with positive weights between concave functions with different variables, is concave in these variables jointly. To prove $\ell(\boldsymbol{\theta}_*)$ is concave, we only need to show that, $\ell_0(\boldsymbol{\pi}_0)$, $\ell_{x_{i,k}}(\boldsymbol{\rho}_i, \boldsymbol{\alpha}_i)$ and $\ell_{y_{c,k}}(\mu_c, \boldsymbol{\beta}_c)$ are concave.

- $\ell_0(\boldsymbol{\pi}_0)$ is concave in $\boldsymbol{\pi}_0$ on $\mathbb{R}^N$.

  To see this, let us look at the Hessian,

  $$\mathbf{H}_{\ell_0}(\pi_0) = -\boldsymbol{\Sigma}_0 \prec 0. \tag{3.78}$$

  Hence, $\ell_0(\boldsymbol{\pi}_0)$ is concave.

- $\ell_{x_{i,k}}(\boldsymbol{\rho}_i, \boldsymbol{\alpha}_i)$ is concave in $\boldsymbol{\rho}_i$ and $\boldsymbol{\alpha}_i$ on $\mathbb{R}^N \times \mathbb{R}^M$.

  Let $\mathbf{d}_i = [\boldsymbol{\rho}_i^{\mathrm{T}}, \boldsymbol{\alpha}_i^{\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{z}_{k-1} = [\mathbf{x}_{k-1}^{\mathrm{T}}, \mathbf{u}_k^{\mathrm{T}}]^{\mathrm{T}}$, then $\ell_{x_{i,k}}(\mathbf{d}_i)$ has the expression:

  $$\ell_{x_{i,k}}(\mathbf{d}_i) = -\frac{\left(x_{i,k} - \mathbf{d}_i^{\mathrm{T}}\mathbf{z}_{k-1}\right)^2}{2\sigma_\varepsilon^2} + \text{const.} \tag{3.79}$$

The Hessian matrix is:

$$\mathbf{H}_{\ell_{x_{i,k}}}(\mathbf{d}_i) = -\frac{1}{\sigma_\varepsilon^2}\mathbf{z}_{k-1}\mathbf{z}_{k-1}^{\mathrm{T}} \preceq 0. \tag{3.80}$$

Hence, $\ell_{x_{i,k}}(\mathbf{d}_i)$ is concave, and $\ell_{x_{i,k}}(\boldsymbol{\rho}_i, \boldsymbol{\alpha}_i)$ is concave.

- $\ell_{y_{c,k}}(\mu_c, \boldsymbol{\beta}_c)$ is concave in μ_c and $\boldsymbol{\beta}_c$ on $\mathbb{R} \times \mathbb{R}^N$.

  Rewrite $\ell_{y_{c,k}}(\mu_c, \boldsymbol{\beta}_c)$ as $\ell_{y_{c,k}}(\boldsymbol{\theta}_c)$. Simply apply theorem 3.1, we have, with the given conditions, the concavity of $\ell_{y_{c,k}}(\boldsymbol{\theta}_c)$, so does $\ell_{y_{c,k}}(\mu_c, \boldsymbol{\beta}_c)$.

This completes the proof. □

Theorem 3.12. *Given $q(\mathbf{x}_{0:K})$, $\mathbf{\Sigma}_0$ and $\mathbf{\Sigma}_\varepsilon$. $\mathcal{Q}(q, \boldsymbol{\theta}_*)$ is concave in $\boldsymbol{\theta}_*$ on its domain.*

Proof. Recall that $\mathcal{Q}(q, \boldsymbol{\theta}_*)$ has the following expression:

$$\mathcal{Q}(q, \boldsymbol{\theta}_*) = \int_{\mathbf{x}_{0:k}} q(\mathbf{x}_{0:K})\ell(\mathbf{x}_{0:K}, \theta_*)d\mathbf{x}_{0:K}. \tag{3.81}$$

As $q(\mathbf{x}_{0:K})$ is a probability density function, which is positive everywhere, and due to lemma 3.11, $\ell(\mathbf{x}_{0:K}, \theta_*)$ is concave in θ_*. The integral operation preserves the concavity. Hence, $\mathcal{Q}(q, \boldsymbol{\theta}_*)$ is concave in $\boldsymbol{\theta}_*$, as desired. □

Now, let us consider another case whereby $\boldsymbol{\theta}_*$ is given and $(\mathbf{\Sigma}_0, \mathbf{\Sigma}_\varepsilon)$ is unknown. As $(\mathbf{\Sigma}_0, \mathbf{\Sigma}_\varepsilon)$ being the covariance matrices, Gaussian likelihood function is not concave in them. Hence, we apply a change of variable, letting $\mathbf{S}_0 = \mathbf{\Sigma}_0^{-1}$ and $\mathbf{S}_\varepsilon = \mathbf{\Sigma}_\varepsilon^{-1}$.

Lemma 3.13. *Given $\mathbf{x}_{0:K}$, $\mathbf{y}_{1:K}$ and $\boldsymbol{\theta}_*$, rewrite the log-complete data likelihood as $\ell(\mathbf{S}_0, \mathbf{S}_\varepsilon)$, where $\boldsymbol{\theta}_* = (\boldsymbol{\pi}_0, \{\boldsymbol{\rho}_i, \boldsymbol{\alpha}_i\}, \{\mu_c, \boldsymbol{\beta}_c\})$. Then $\ell(\mathbf{S}_0, \mathbf{S}_\varepsilon)$ is concave on $\mathbb{R}_{++}^{N\times N} \times \mathbb{R}_{++}^{N\times N}$.*

Proof. Let $\ell_0(\mathbf{S}_0) = \ln p(\mathbf{x}_0|\boldsymbol{\pi}_0, \mathbf{\Sigma}_0)$ when $\mathbf{x}_0$ and $\boldsymbol{\pi}_0$ are given. Similarly, $\ell(\mathbf{S}_\varepsilon)$ be the log-complete data likelihood, which can be write as:

$$\ell(\mathbf{S}_0, \mathbf{S}_\varepsilon) = \ell_0(\mathbf{S}_0) + \sum_{k=1}^{K} \ell_{\mathbf{x}_k}(\mathbf{S}_\varepsilon) + \mathrm{const} \tag{3.82}$$

where the const refers to the likelihood term which is not a function of $(\mathbf{S}_0, \mathbf{S}_\varepsilon)$. To prove $\ell(\mathbf{S}_0, \mathbf{S}_\varepsilon)$ is concave, we only need to show that, $\ell_0(\mathbf{S}_0)$ and $\ell_{x_{i,k}}(\mathbf{\Sigma}_\varepsilon)$ are concave on $\mathbb{R}_{++}^{N\times N}$.

- $\ell_0(\mathbf{S}_0)$ is concave.

With the change of variable, $\ell_0(\mathbf{S}_0)$ becomes

$$\ell_0(\mathbf{S}_0) = \frac{1}{2}\ln(|\mathbf{S}_0|) - \frac{1}{2}\mathbf{tr}\left(\mathbf{S}_0\mathbf{X}_0\right) + \text{const}, \tag{3.83}$$

where, $\mathbf{X}_0 = (\mathbf{x}_0 - \boldsymbol{\pi}_0)(\mathbf{x}_0 - \boldsymbol{\pi}_0)^{\mathrm{T}}$. $\ln(|\mathbf{S}_0|)$ is concave on $\mathbb{R}_{++}^{N\times N}$ (two versions of proof can be found in Cover and Thomas (1988); Boyd and Vandenberghe (2004)), and $\mathbf{tr}\left(\mathbf{S}_0\mathbf{X}_0\right)$ is both concave and convex on $\mathbb{R}_{++}^{N\times N}$, so does $-\mathbf{tr}\left(\mathbf{S}_0\mathbf{X}_0\right)$. We conclude that, the positive weighted sum of the two terms $\ell_0(\mathbf{S}_0)$ is concave on $\mathbb{R}_{++}^{N\times N}$.

- $\ell_{\mathbf{x}_k}(\boldsymbol{\Sigma}_\varepsilon)$ is concave on $\mathbb{R}_{++}^{N\times N}$.

  Similar to $\ell_0(\mathbf{S}_0)$, $\ell_{\mathbf{x}_k}(\boldsymbol{\Sigma}_\varepsilon)$ can be written as

$$\ell_{\mathbf{x}_k}(\mathbf{S}_\varepsilon) = \frac{1}{2}\ln(|\mathbf{S}_\varepsilon|) - \frac{1}{2}\mathbf{tr}\left(\mathbf{S}_\varepsilon\mathbf{X}_k\right) + \text{const}, \tag{3.84}$$

  where, $\mathbf{X}_k = (\mathbf{x}_k - \mathbf{P}\mathbf{x}_{k-1} - \mathbf{A}\mathbf{u}_k)(\mathbf{x}_k - \mathbf{P}\mathbf{x}_{k-1} - \mathbf{A}\mathbf{u}_k)^{\mathrm{T}}$. Use the same proof for $\ell_0(\mathbf{S}_0)$, we establish the concavity of $\ell_{\mathbf{x}_k}(\boldsymbol{\Sigma}_\varepsilon)$ on $\mathbb{R}_{++}^{N\times N}$.

This completes the proof. □

Theorem 3.14. *Given $q(\mathbf{x}_{0:K})$ and $\boldsymbol{\theta}_*$, $\mathcal{Q}(q, \mathbf{S}_0, \mathbf{S}_\varepsilon)$ is concave in $(\mathbf{S}_0, \mathbf{S}_\varepsilon)$ on $\mathbb{R}_{++}^{N\times N} \times \mathbb{R}_{++}^{N\times N}$.*

Proof. Use lemma 3.13, on $\mathbb{R}_{++}^{N\times N} \times \mathbb{R}_{++}^{N\times N}$, we have $\ell(\mathbf{x}_{0:K}, \boldsymbol{\theta})$ is concave in $(\mathbf{S}_0, \mathbf{S}_\varepsilon)$. Then, the proof follows the same as theorem 3.12. □

Theorem 3.12 and 3.14 establish that $\mathcal{Q}(q, \boldsymbol{\theta})$ is concave in two parameter settings, under which lead to unique solutions. However, for cases both $q(\mathbf{x}_{0:K})$ and $\boldsymbol{\theta}$ are unknown, the $\mathcal{Q}$-function is not concave. This means that, we can only ensure that, in the M-step, the solutions under the two parameter settings are unique. Combining the E-step and M-step, there is no such guarantee.

3.4.3 $\mathcal{Q}$-function: Empirical landscapes

Since no theoretical guarantee is available for the shape of the $\mathcal{Q}$-function in q and $\boldsymbol{\theta}$ jointly, we have to investigate it numerically. To this end, we plot the landscapes $\mathcal{Q}$-function under two settings. Firstly, let the $\mathcal{Q}$-function be a function of the parameters only, while the sufficient statistics of states are fixed. Such setting reflects the scenarios considered theoretically in the previous subsection. Secondly, let the $\mathcal{Q}$-function be a function jointly varying in q and $\boldsymbol{\theta}$, and show the variations in $\boldsymbol{\theta}$ only. This means q is recomputed for every parameter variation.

We restrict ourself to the scalar states case and use three test datasets:

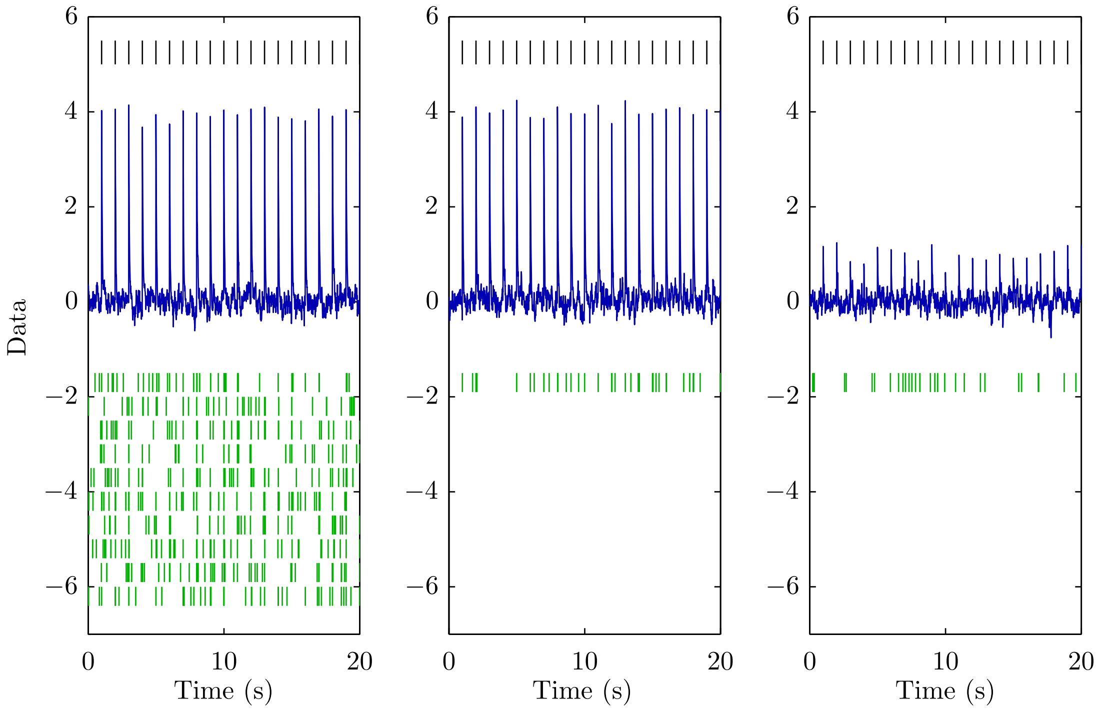


Figure 3.13: Test datasets: **D1** (*left*), **D2** (*middle*) and **D3** (*right*). For each panel, the *black bars*, *blue line* and *green bars* denotes for inputs, states, and observations, respectively.

1. **D1**: A realisation of the parameter and dimension settings in example 3.3. Results are shown in figure 3.14 and 3.15.
2. **D2**: A realisation obtained by reducing the number of channels to 1, while keeping the other settings in example 3.3. Results are shown in figure 3.16 and 3.17.
3. **D3**: A realisation based the settings in **D2**, but reducing the weight of the inputs to 1. Results are shown in figure 3.18 and 3.19.

Remark 3.15. *The essence behind these three experiments is that, the information about the underlying states significantly shrinks in the binary observations, as we reduce the number of channels and weight of the inputs. Such feature is clearly visible in figure 3.13.*

Figure 3.14, 3.16 and 3.18 confirm the theorems in the previous section, that is, if q is fixed, $\mathcal{Q}(q, \boldsymbol{\theta})$ is always concave in $\boldsymbol{\theta}$. Whether the observation is informative to the state or not, does not matter. However, the information strongly effects the convergence speed. We see that the maxima of the $\mathcal{Q}$-function at 1th, 15th and 50th iteration clearly differ from each other in **D1**. Such differences become much smaller in **D2** and **D3** (This is especially obvious in ρ and α). Another important point conveys by these three figures is that, the $\mathcal{Q}$-function is a skewed in most parameters. Recall that, in the EM, lower bound is lifted at each iteration until it touches the real objective function – marginal

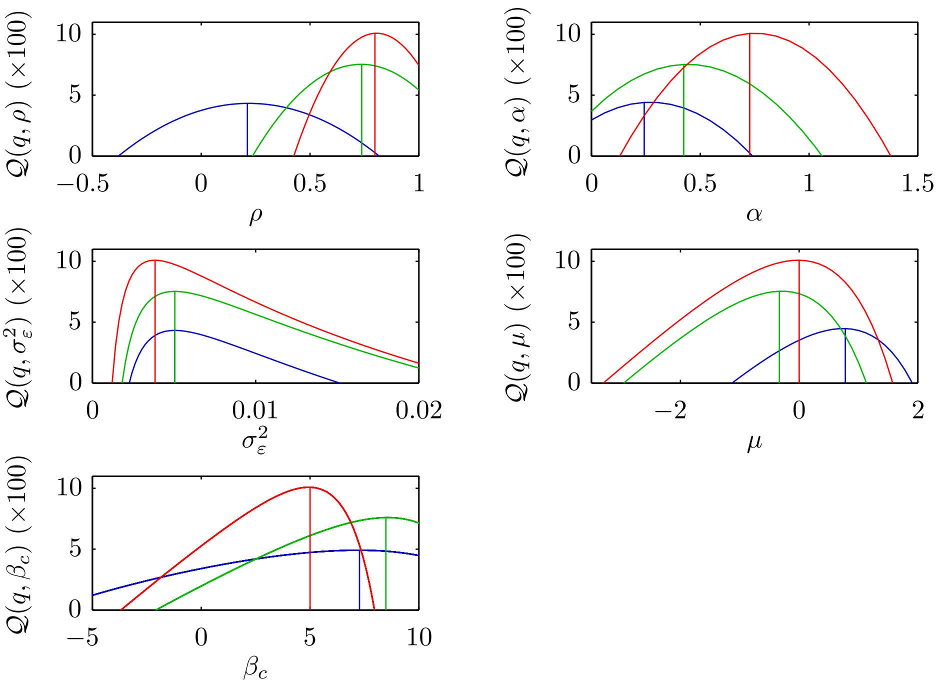


Figure 3.14: Based on test dataset **D1**, $\mathcal{Q}(q, \boldsymbol{\theta})$ at iteration 1 (*blue*), 15 (*green*) and 50 (*red*), in which all five parameters are estimated jointly. The colored vertical bars represent the maximum points. Each panel is drawn by fixing the other parameters to their estimates at the previous iteration, and state sufficient statistics are fixed by their values at the current iteration.

likelihood. As the lower bound being skewed, it raises the possibility of bias during the maximisation process.

Such biases are evident in figure 3.15, 3.17 and 3.19, and increase as the observations becomes less and less informative to states in **D2** and **D3**. In particular, in **D3**, local optimals appear in ρ and β_c. And the global optimals are far from true value because EM severely over fits the data. In addition, throughout **D1**, **D2** and **D3**, the maxima of the $\mathcal{Q}$-function is alway towards zero, regardless of the true value.

These empirical findings provide a guideline for the learning the parameters in SSPP:

1. It is advisable to always fixing σ_ε^2.

2. It is advisable to fix one between α and β_c.

3. When dealing with single channel data. It is advisable to fix ρ and β at reasonable values.

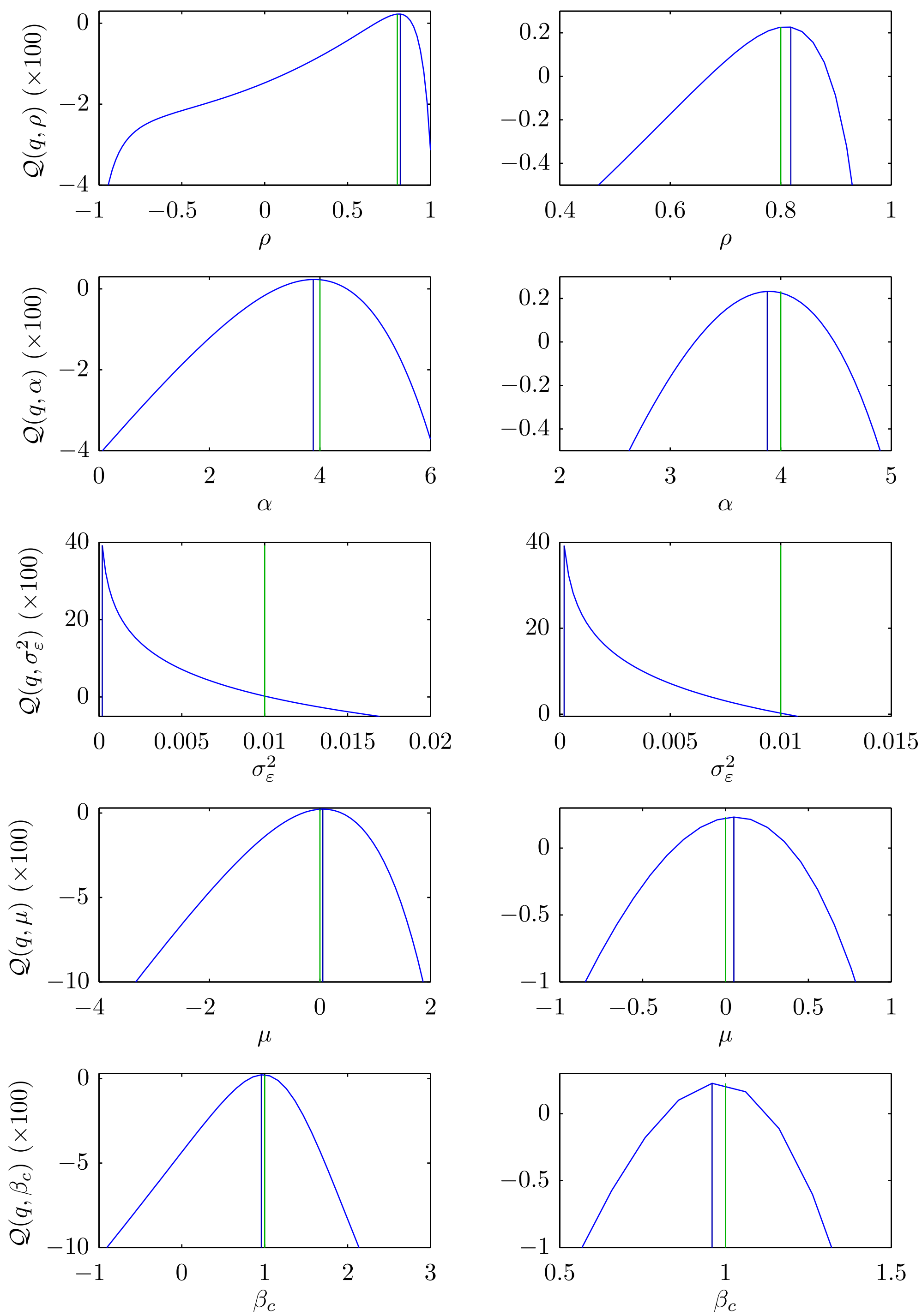


Figure 3.15: Based on test dataset **D1**, $\mathcal{Q}(q,\boldsymbol{\theta})$ in each of the five parameters, where $q(\cdot)$ is recomputed for every parameter variation. The *right panels* are the zoomed in version to their *left* counterparts. Each panel is computed by fixing the other four parameters to their true value. The maximum $\mathcal{Q}(q,\boldsymbol{\theta})$ and the true values are indicated by *blue* and *green* vertical lines. ρ is chosen in $(-1,1)$, following by the stationary condition of AR(1) model.

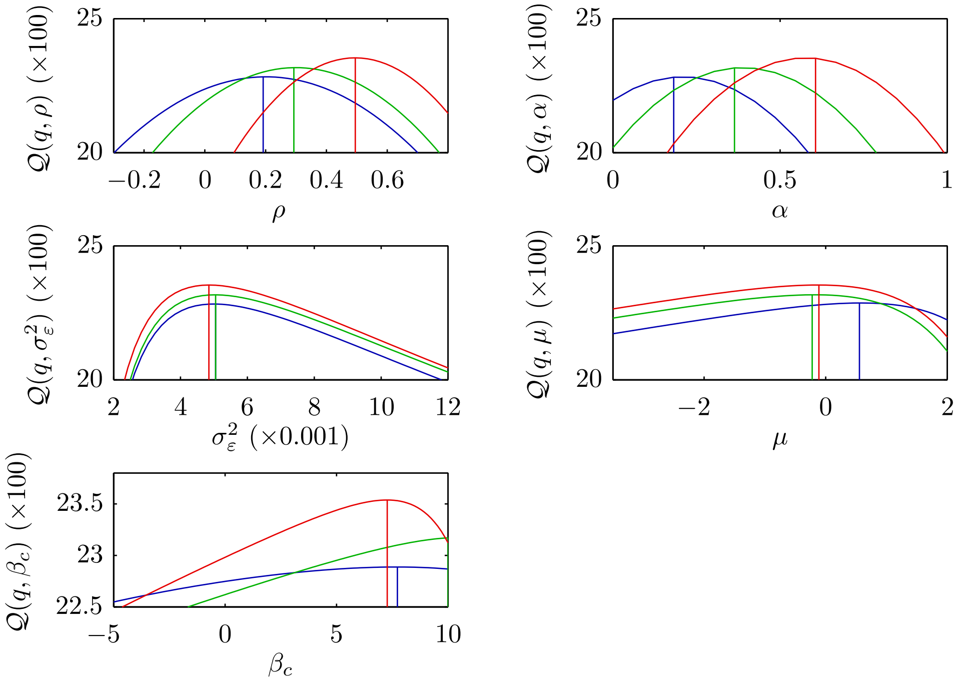


Figure 3.16: Based on test dataset **D2** $\mathcal{Q}(q,\boldsymbol{\theta})$ at iteration 1 (*blue*), 15 (*green*) and 50 (*red*), in which all five parameters are estimated jointly. The colored vertical bars represent the maximum points. Each panel is drawn by fixing the other parameters to their estimates at the previous iteration, and state sufficient statistics are fixed by their values at the current iteration.

3.5 Assessing model goodness-of-fit

Being a data-driven model, it is always important to evaluate how well the model describes the spikes data, that is, assessing model goodness-of-fit. For this, Brown et al. (2002) proposed the time-rescaling theorem which allows the *Kolmogorov-Smirnov* (KS) test. The theorem is the following:

Theorem 3.16. (Time-Rescaling Theorem (Brown et al., 2002).) *Let* $0 < s_1 < s_2 < \cdots < s_J < T$ *be a realisation from a point process with a conditional intensity function* $\lambda(t|H_t)$ *satisfying* $0 < \lambda(t|H_t) < \infty$ *with probability one,* $\forall t \in (0,T]$*. Define a transformation:*

$$\Lambda(s_j) = \int_0^{s_j} \lambda(v|H_t)dv, \tag{3.85}$$

for $j = 1, \cdots, J$*. Then* $\{\Lambda(s_j)\}$ *are a Poisson process with unit rate.*

Proof. The proof can be found in Brown et al. (2002). □

A simple consequence of theorem 3.16 is that, if the model is correct, then difference between two $\Lambda(s_j)$s denoting as τ_j are independent exponential random variables with

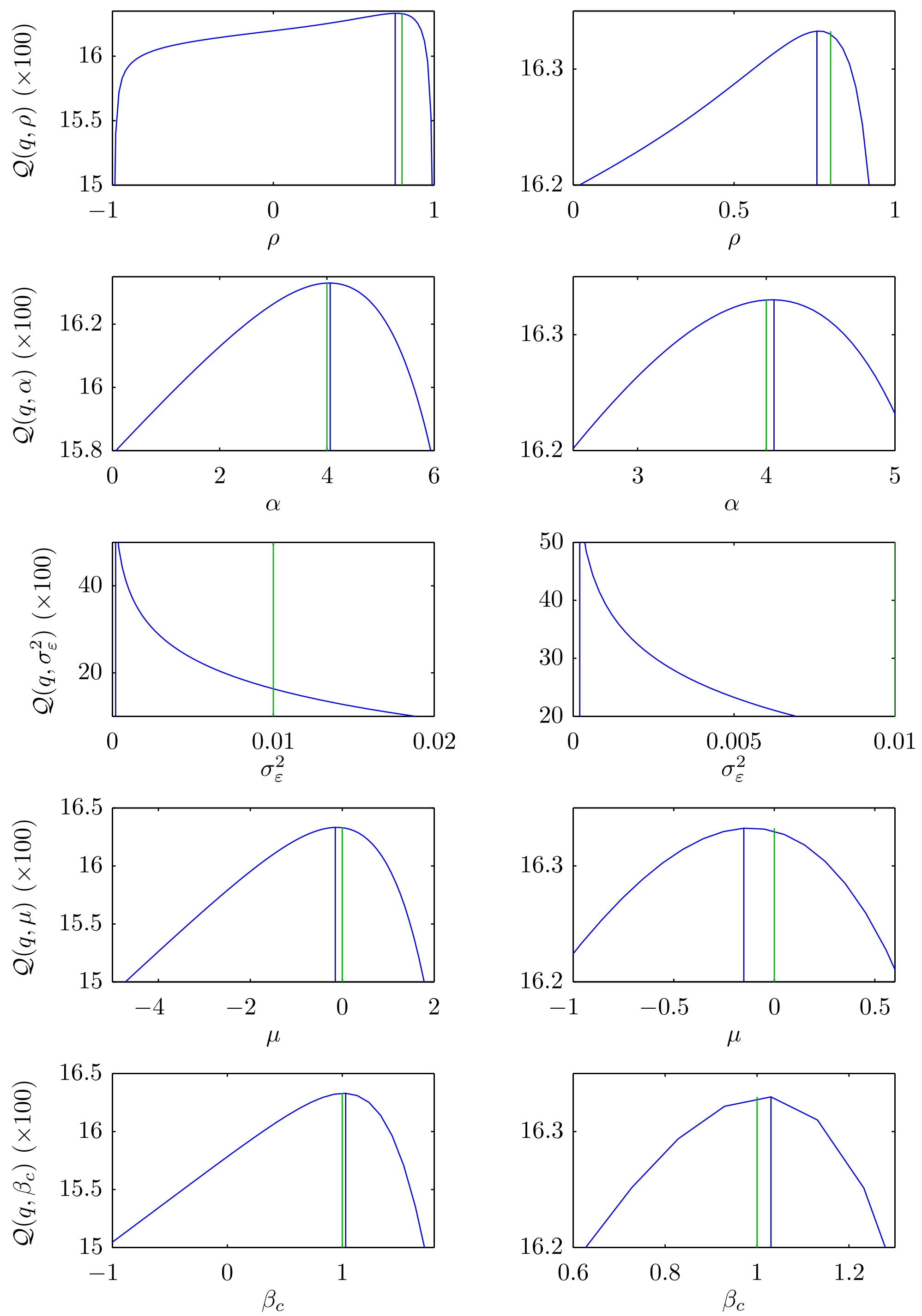


Figure 3.17: Based on test dataset **D2**, $\mathcal{Q}(q,\boldsymbol{\theta})$ in each of the five parameters, where $q(\cdot)$ is is recomputed for every parameter variation. The *right panels* are the zoomed in version to their *left* counterparts. Each panel is computed by fixing the other four parameters to their true value. The maximum $\mathcal{Q}(q,\boldsymbol{\theta})$ and the true values are indicated by *blue* and *green* vertical lines. ρ is chosen in $(-1,1)$, following by the stationary condition of AR(1) model.

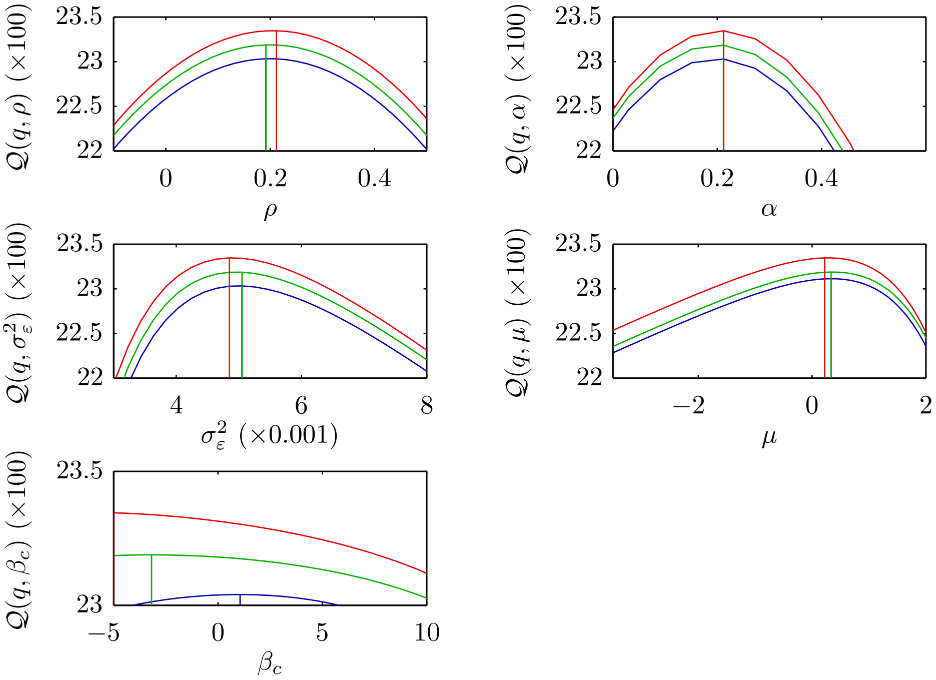


Figure 3.18: Based on test dataset **D3**, $\mathcal{Q}(q, \boldsymbol{\theta})$ at iteration 1 (*blue*), 15 (*green*) and 50 (*red*), in which all five parameters are estimated jointly. The colored vertical bars represent the maximum points. Each panel is drawn by fixing the other parameters to their estimates at the previous iteration, and state sufficient statistics are fixed by their values at the current iteration.

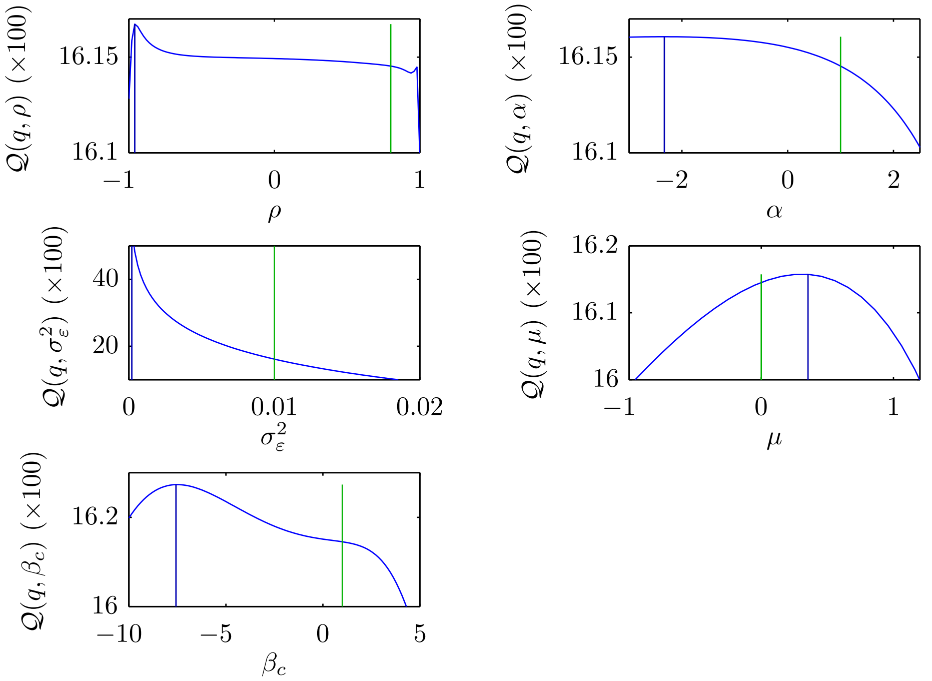


Figure 3.19: Based on test dataset **D3**, $\mathcal{Q}(q, \boldsymbol{\theta})$ in each of the five parameters, where $q(\cdot)$ is recomputed for every parameter variation. Each panel is computed by fixing the other four parameters to their true value. The maximum $\mathcal{Q}(q, \boldsymbol{\theta})$ and the true values are indicated by *blue* and *green* vertical lines. ρ is chosen in $(-1, 1)$, following by the stationary condition of AR(1) model.

a unit rate. Use this fact, one can compute τ_j with the estimated model as:

$$\tau_j = \Lambda(s_j) - \Lambda(s_{j-1}) = \int_{s_{j-1}}^{s_j} \lambda(v|\hat{\mathbf{x}}, \hat{\boldsymbol{\theta}})dv \tag{3.86}$$

where $\hat{\mathbf{x}}$ and $\hat{\boldsymbol{\theta}}$ are the estimated hidden states and parameters. Then, if we make a further transformation:

$$z_j = 1 - \exp(-\tau_j), \tag{3.87}$$

z_js will be independent uniform random variables on the interval $[0, 1]$. Thus, a KS test can be performed to measure agreement between z_js and an uniform probability density that are constructed from the observed data.

Specifically, the *cumulative density function* (CDF) of the uniform density is defined as:

$$b_j = \frac{j - 0.5}{J} \tag{3.88}$$

for $j = 1, \ldots, J$, Then, we order the z_js in ascending order. Plotting the ordered z_js against the b_js, if the model is correct, the points should lie on the line of 45 degree. Moreover, the 95% confident interval can be computed as $\frac{b_j \pm 1.36}{\sqrt{J}}$. These procedures are summarised in algorithm 3.2. An empirical example based on the experiment in figure 3.11 is shown in figure 3.20.

Algorithm 3.2. Time-rescaling theorem based KS test

```
Input: $y_k$ $\lambda_k, \forall k \in [1, \cdots, K]$, $\Delta$
Output: $b_j$ and $z_j$, $\forall j \in [1, \cdots, J]$
Let $j = 0$ and $s_0 = 0$
for $k = 1$ to $K$ do
    if $y_k = 1$ then
        $s_j = k\Delta$
        $j = j + 1$
    end if
end for
for $j = 1$ to $J$ do
    $\tau_j = \sum_{s_{j-1}/\Delta}^{s_j/\Delta} \hat{\lambda}_k\Delta$
    Compute $z_j$ and $b_j$ with equation (3.87) and (3.88)
end for
```

3.6 Discussion

Vector states vs. scalar states. In this chapter, we provide a generalised formulation for SSPP in vector state space, and the approximate EM algorithm for inference and

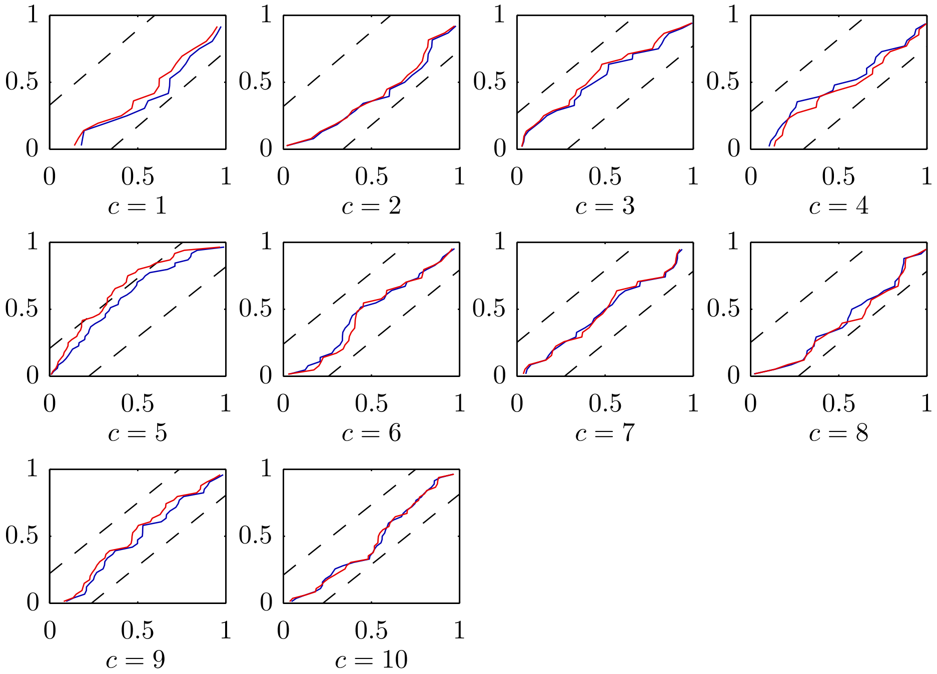


Figure 3.20: An example of the time rescaled theorem based KS test for 10-channel SSPP, estimated by EM. Each panel corresponds to the test for a individual channel labelled by c. Y-axis is CDF b_j and x-axis is quintile z_j. *Blue line* represents the KS test for the true model and *red line* represents the KS test for the estimated model. The two *black dashed lines* are the 95% confident intervals. In this case, the performance of EM is mostly reliable and in good agreements with the true model.

learning. However, the numerical results are with scalar states case. One may wonder what happens if the states are vectors. The answer to this question can be somehow deduced from the empirical landscapes of the $\mathcal{Q}$-functions. That is whether the observation is informative enough to support a reliable estimation. To see this, we use the following example:

Example 3.4. *Over an observation interval* $(0, T]$, *where* $T = 20s$, *choose* $\Delta = 10ms$, *such that* $K = 2000$. *Let* $\mathbf{x}_k \in \mathbb{R}^2$ $\mathbf{y}_k$ *be a* 10*-dimensional binary vector and* $\mathbf{u}_k$ *be a* 3*-dimensional binary vector. Specifically, the input is applied at every* 1, 1.5 *and* 2 *second for* $u_{1,k}$, $u_{2,k}$ *and* $u_{3,k}$, *respectively. The parameters in the states dynamics are:*

$$\mathbf{P} = \begin{bmatrix} 0.8 & 0 \\ -0.2 & 0.9 \end{bmatrix}, \qquad \mathbf{A} = \begin{bmatrix} 0.5 & 2 & 1.2 \\ 1.1 & 1.3 & 1.19 \end{bmatrix}, \qquad \mathbf{\Sigma}_\varepsilon = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}. \tag{3.89}$$

The parameters in the observation model are:

$$\mu_c = 0, \forall c, \qquad [\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_{10}] = \begin{bmatrix} 0.5 & \cdots & 1 \\ 0.5 & \cdots & 1 \end{bmatrix}. \tag{3.90}$$

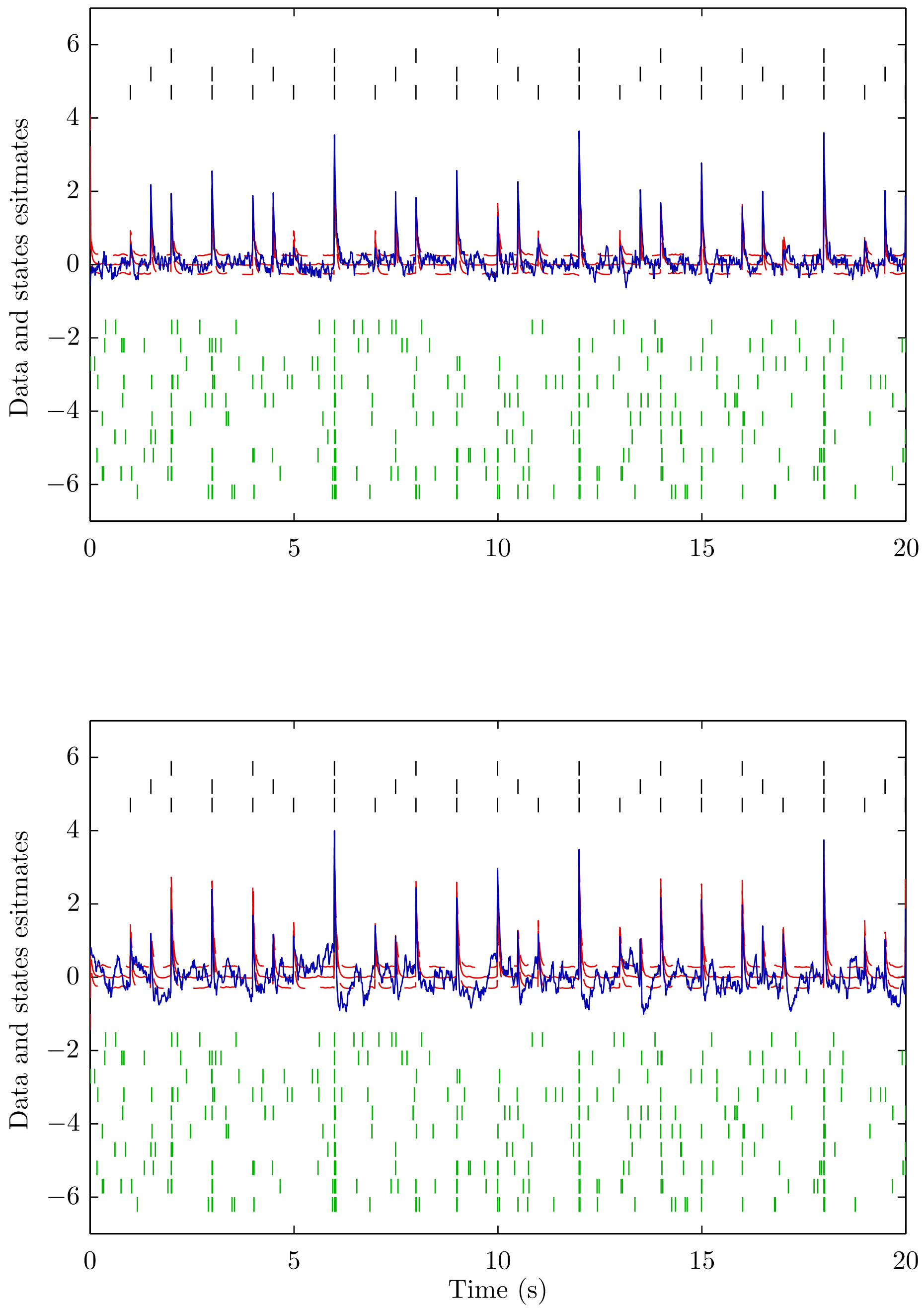


Figure 3.21: Results on EM algorithm for vector states SSPP descried in example 3.4. *Top*: $x_{1,k}$. *Bottom*: $x_{2,k}$ Inputs as *black bars* scaled for visualisation and true states as *blue solid line*, observed 10-channel binary sequences as *green bars*, the mode of the smoothing density $x_{k|K}$ (*red solid line*) and 95% confident intervals (*red dashed lines*) computed as $x_{k|K} \pm 1.96\sigma_{k|K}$.

Based on this parameter setting, we applied the approximate EM algorithm in which $\mathbf{\Sigma}_\varepsilon$ *and* $\boldsymbol{\beta}_c$*s are fixed. Results are shown in figure 3.21 and 3.22.*

In figure 3.21, the state estimation is still in good agreement to the truth, whereas the learned parameter 3.22 is far away from the truth. Selecting the dimension of the states is a model selection problem. One should resort to Akaike information criteria

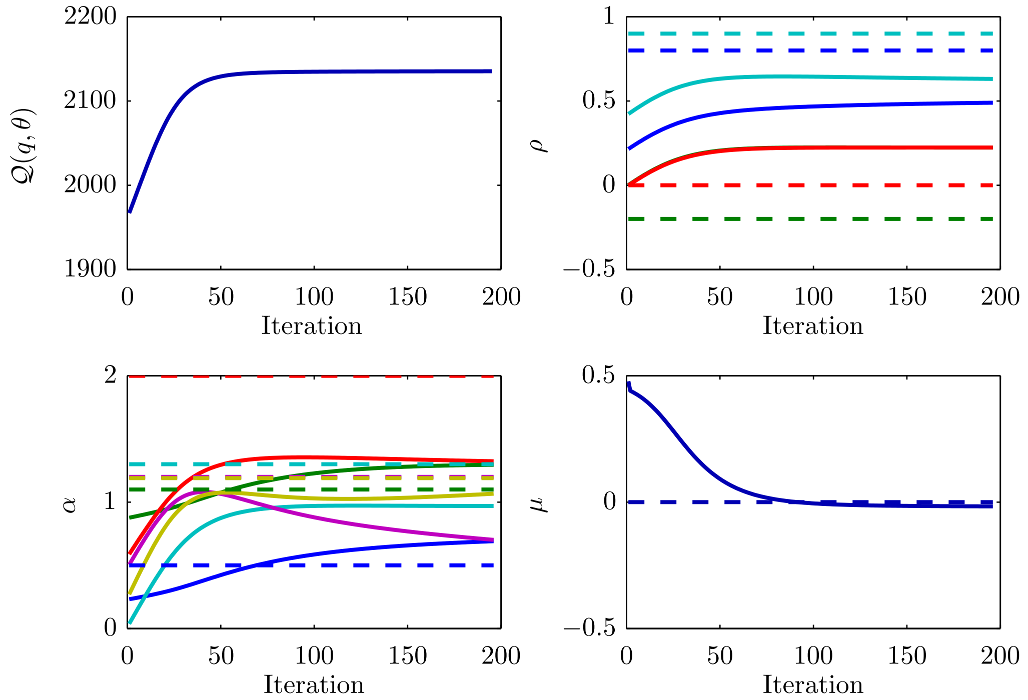


Figure 3.22: Results on EM algorithm for vector states SSPP descried in example 3.4. *Top*: Inputs as *black bars* scaled for visualisation and true states as *blue solid line*, observed 10-channel binary sequences as *green bars*, the mode of the smoothing density $x_{k|K}$ (*red solid line*) and 95% confident intervals (*red dashed lines*) computed as $x_{k|K} \pm 1.96\sigma_{k|K}$.

(AIC), Bayesian information criteria (BIC) or Bayes factor. In this case, it is clear that unless there are massive information available (meaning large dataset), or clear modelling objects (e.g. tracking multiple underlying biological processes), the scalar state systems are preferred. Further, even with scalar states, the model is able to deal with scenarios that, multiple stimuli are applied simultaneously during a multi-channel recording.

Accuracy of LGF. The theorem 3.10 guarantees the log-concavity for both filtering and smoothing densities in SSPP. According to Koyama et al. (2010), the accuracy of the LGF for both filtering and smoothing in SSPP is $\mathcal{O}(\gamma^{-1})^4$, where $\gamma = \Delta \sum_{c=1}^{C} e^{\mu_c}||\boldsymbol{\beta}_c||^2 + ||\boldsymbol{\Sigma}_\varepsilon^{-1}||$. Numerically, Koyama et al. (2010) show that, the LGF is more accurate than the particle filter with 10,000 particles. This motivates the possibility of approximate the marginal likelihood at each time point $p(\mathbf{y}_k|\mathbf{y}_{1:k-1})$ within LGF, which makes LGF as a full package filtering method competitive with Kalman filters and particle filters. We show this in later chapter under a sampling setting.

Bayesian inference for both states and parameters. The high skewness is indicative of parameter posteriors where simple maximum likelihood estimates of the parameters may be quite far from the actual posterior means, which requires full Bayesian

[4] In this thesis, only the first-order Laplace approximation is considered. If the second-order Laplace approximation is applied, the accuracy increases to $\mathcal{O}(\gamma^{-2})$

inference methods. From the modelling nature, MLE is also not satisfying, due to the fact that it does not provide the degrees of believe on the parameter values, which conveys the curial information about the biological state. When such information are used to perform classification or prediction, the results could be inaccurate. In the following two chapters, we investigate Bayesian treatments for SSPP thoroughly.

Chapter 4

Variational methods

In this Chapter [1]*, we propose a variational Bayesian (VB) approach to solve this problem, extending results of Beal (2003) to the SSPP case to obtain a variational smoother which offers a good compromise between distributional accuracy and computational efficiency. The developed techniques are demonstrated on a synthetic data set, showing good performance when compared to EM and fully Bayesian approaches based on Gibbs sampling. The details of a VB filter are also given, using ideas taken from dual filtering (Wan and Nelson, 2001) whereby parameters are allowed to evolve to track changes in the system's mode of operation.*

4.1 Basics of variational method

Suppose we have a posterior distribution of interest, $p(\mathbf{X}|\mathbf{Y})$, where $\mathbf{X} \in \mathbb{R}^{M\times N}$ stands for the variables of interest (e.g. parameters or hidden variables and etc), $\mathbf{Y} \in \mathbb{R}^{D\times N}$ refers to the observed data. Without loss of generality, $\mathbf{X}$ and $\mathbf{Y}$ are set to have the same column number, which matches the scenario in state-space models, where N stands for total time points. It is obvious that exact computation of the posterior is not feasible, given the fact that $\mathbf{X}$ is continuously valued and with high dimensionality. The variational method aims to find an approachable distribution $q(\mathbf{X})$ which is the closest one to $p(\mathbf{X}|\mathbf{Y})$. The degree of closeness is described by the *Kullback-Leibler* (KL) divergence between the two,

$$\mathrm{KL}(q||p) = \int_{\mathbf{X}} q(\mathbf{X}) \ln \frac{q(\mathbf{X})}{p(\mathbf{X}|\mathbf{Y})} d\mathbf{X}. \tag{4.1}$$

[1] Part of the work in this Chapter was done in collaboration with Andrew Zammit Magnion, during his visit to Southampton (May 2010). This builds on his previous work on deriving a Variationla Bayes filter for integro-difference equations (Zammit Mangion et al., 2011a). The derivations for the SSPP model were done jointly with him., and published in Zammit Mangion et al. (2011b).

The variational approach can be formed as an optimization problem over *functionals* as,

$$\min_{q} \mathrm{KL}(q||p), \quad \text{s.t.} \quad q \in \mathscr{Q} \tag{4.2}$$

where $\mathscr{Q}$ denotes for the set of functions that q can take. In this way, the inference problem is transformed into an seemly approachable optimization problem. However, it is still not clear how to solve it in the absence of the true posterior $p(\mathbf{X}|\mathbf{Y})$.

Fortunately, the KL divergence is closely related to the marginal likelihood $p(\mathbf{Y})$, which makes the target posterior intractable. As a result, the inference problem can be further transformed. Precisely, we have,

$$\ln p(\mathbf{Y}) = \mathcal{F}(q) + \mathrm{KL}(q||p), \tag{4.3}$$

where,

$$\mathcal{F}(q) = \int_{\mathbf{X}} q(\mathbf{X}) \ln \frac{p(\mathbf{X}, \mathbf{Y})}{q(\mathbf{X})} d\mathbf{X}, \tag{4.4}$$

which is the objective function of EM algorithm. Given the fact that $\ln p(\mathbf{Y})$ is a constant, minimizing $\mathrm{KL}(q||p)$ is equivalent to maximizing the so called *variational free energy*, $\mathcal{F}(q)$.

$$\begin{aligned} &\min_{q} \mathrm{KL}(q||p) \leftrightarrow \max_{q} \mathcal{F}(q) \\ &\text{s.t.} \quad q \in \mathscr{Q}. \end{aligned} \tag{4.5}$$

At this point, the variational method is yet an approximation method to the exact posterior. Since if $q(\mathbf{X}) = p(\mathbf{X}|\mathbf{Y})$, $\mathrm{KL}(q||p) = 0$. The true posterior is always the optimal solution. The approximation is introduced by constraints on $\mathscr{Q}$.

Practically, $q(\mathbf{X})$ is often restricted to be a product over functionals of components (Jordan et al., 1999), such as,

$$q(\mathbf{X}) = \prod_{j=1}^{M} q_j(\mathbf{X}_j). \tag{4.6}$$

Each component can contain an arbitrary number of variables, and this number can be different across components. The only requirement over this setting is that, each $q_j(\mathbf{X}_j)$ should be a distribution with known form, e.g. Gaussian, Dirichlet and etc. Such a factorization, which is not necessarily true in the target distribution, is inevitably an approximation. In physics literature, this approach is known as the *mean-field theory*, and widely used to compute energy functions known as the (Parisi, 1988).

Given the factorization, we can now maximize $\mathcal{F}(q)$ w.r.t. each q_j in turn. For this, one can take the *functional derivative* (Gelfand and Fomin, 1963) of $\mathcal{F}(q)$ w.r.t. q_j, and solve for setting it to 0. As a result, the optimal $q_j(\mathbf{X}_j)$ takes the form:

$$q_j(\mathbf{X}_j) \propto \exp\left(\mathbb{E}_{q_{\setminus j}}[\ln p(\mathbf{Y}, \mathbf{X})]\right), \tag{4.7}$$

where $q_{\setminus j}$ denotes for all $\{q_i, i \neq j\}$. This is applicable for any probabilistic model, as long as $p(\mathbf{Y}, \mathbf{X})$ is well defined.

Alternatively, as shown in (Bishop, 2006), the same optimal q_j can be obtained by manipulating $\mathcal{F}(q)$,

$$\begin{aligned}\mathcal{F}(q) &= \int_{\mathbf{X}} \prod_i q_i \left(\ln p(\mathbf{Y}, \mathbf{X}) - \sum_i \ln q_i\right) d\mathbf{X} \\ &= \int_{\mathbf{X}_j} q_j \left(\int_{\mathbf{X}_{\setminus j}} \ln p(\mathbf{Y}, \mathbf{X}) \prod_{\setminus j} q_i d\mathbf{X}_{\setminus j}\right) d\mathbf{X}_j - \int q_j \ln q_j d\mathbf{X}_j + C_0 \\ &= \int_{\mathbf{X}_j} q_j \ln \tilde{p}(\mathbf{Y}, \mathbf{X}_j) - \int q_j \ln q_j d\mathbf{X}_j + C_0 \\ &= -\mathrm{KL}(q_j || \tilde{p}) + C_0,\end{aligned} \tag{4.8}$$

where,

$$C_0 = -\sum_{\setminus j} \int_{\mathbf{X}_i} q_i \ln q_i d\mathbf{X}_i, \quad \ln \tilde{p}(\mathbf{Y}, \mathbf{X}_j) = \mathbb{E}_{q_{\setminus j}}[\ln p(\mathbf{Y}, \mathbf{X})] + C_1, \tag{4.9}$$

C_1 is a normalizing constant that ensures $\tilde{p}(\mathbf{Y}, \mathbf{X})$ being a proper distribution. Consequently, the optimization problem for each q_j is further transformed as:

$$\max_{q_j} \mathcal{F}(q) \leftrightarrow \min_{q_j} \mathrm{KL}(q_j || \tilde{p}). \tag{4.10}$$

Finally, the KL divergence is minimized when

$$\begin{aligned} q_j(\mathbf{X}_j) &= \tilde{p}(\mathbf{Y}, \mathbf{X}_j) \\ &= \exp\left(\mathbb{E}_{q_{\setminus j}}[\ln p(\mathbf{Y}, \mathbf{X})] + C_1\right) \\ &\propto \exp\left(\mathbb{E}_{q_{\setminus j}}[\ln p(\mathbf{Y}, \mathbf{X})]\right),\end{aligned} \tag{4.11}$$

which is the same as equation (4.7).

Example 4.1. *(Bishop, 2006) Suppose our target distribution is a two dimensional Gaussian, $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{S}^{-1})$, where $\mathbf{x}, \boldsymbol{\mu} \in \mathbb{R}^2$, and $\mathbf{S} \in \mathbb{R}^{2\times 2}$ is the precision matrix. Applying the variational method, we let $q(\mathbf{x}) = q(x_1)q(x_2)$. According to equation (4.7),*

$q(x_1)$ can be obtained as

$$\begin{aligned}\ln q_1(x_1) &= \mathbb{E}_{q_2}[\ln p(\mathbf{x})] + const \\ &= \mathbb{E}_{q_2}\left[-\frac{1}{2}(x_1-\mu_1)^2 s_{11} - (z_1-\mu_1)s_{12}(x_2-\mu_2)\right] + const \\ &= -\frac{1}{2}x_1^2 s_{11} + x_1\mu_1 s_{11} - x_1 s_{12}(\mathbb{E}_{q_2}[x_2]-\mu_2) + const.\end{aligned} \tag{4.12}$$

By completing the square, $q_1(x_1)$ writes as a Gaussian distribution, $\mathcal{N}(x_1|m_1, s_{11}^{-1})$, where,

$$m_1 = \mu_1 - s_{11}^{-1}s_{12}(\mathbb{E}_{q_2}[x_2]-\mu_2). \tag{4.13}$$

Likewise, $q_2(x_2) = \mathcal{N}(x_2|m_2, s_{22}^{-1})$, where,

$$m_2 = \mu_2 - s_{22}^{-1}s_{21}^{-1}(\mathbb{E}_{q_1}[x_1]-\mu_1). \tag{4.14}$$

The solutions are coupled, since $\mathbb{E}_{q_1}[x_1] = m_1$ and $\mathbb{E}_{q_2}[x_2] = m_2$.

Figure 4.1 shows an example, where $q(\mathbf{x})$ at four different iterations ($l = 0, 5, 10, 40$) are compared. The intermediate approximation $q^{(l^*)}(\mathbf{x}) = q_1^{(l)}(x_1)q_2^{(l-1)}(x_2)$ and the final product $q^{(l)}(\mathbf{x}) = q_1^{(l)}(x_1)q_2^{(l)}(x_2)$ at each iteration, reveal the effect of computing q_1 and q_2 in turn. Note that the difference between $q^{(l^*)}(\mathbf{x})$ and $q^{(l)}(\mathbf{x})$ gradually disappears.

4.2 Batch VBEM for SSPP

4.2.1 Model

Before dive into the variational method for SSPP, let us first recall the model specification with scaler state in the following. For this, we use the terminology in Smith and Brown (2003), in particular,

$$\begin{aligned}\text{State model: } & x_k = \rho x_{k-1} + \alpha I_k + \epsilon_k, && (4.15)\\ \text{Observation model: } & p(y_k^c|x_k,\mu,\beta^c) = \exp(y_k^c \ln(\lambda_k^c\Delta) - \lambda_k^c\Delta), && (4.16)\end{aligned}$$

where I_k and y_k^c are the binary variables. Δ is the time resolution, which is set to ensure 1 spike per time bin. ϵ_k is additive white Gaussian noise with mean 0 and variance $\sigma_\epsilon^2 \in \mathbb{R}^+$. The initial state x_0 is assumed to be normally distributed with known mean $x_{0|0}$ and variance $\sigma_{0|0}^2$. The parameters $\rho \in \mathbb{R}$ and $\alpha \in \mathbb{R}$ are the propagation constant and input gain respectively.

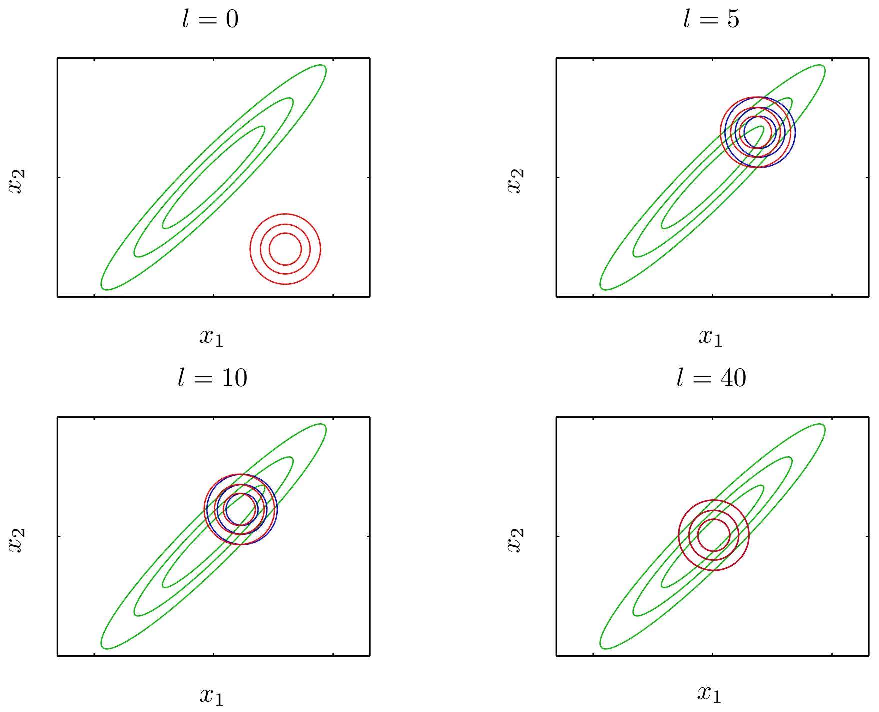


Figure 4.1: Illustration of the variational approximation $q(\mathbf{x}) = q(x_1)q(x_2)$ to the target posterior $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{S}^{-1})$. The panels show $q(\mathbf{x})$ at four different iterations: $l = 0, 5, 10, 40$. The *green* contour denotes the target distribution, the *blue* and *red* contours represent $q_1^{(l)}(x_1)q_2^{(l-1)}(x_2)$ and $q_1^{(l)}(x_1)q_2^{(l)}(x_2)$, respectively.

The CIF, λ takes the exponential form:

$$\lambda_k^c = \exp(\mu + \beta^c x_k), \tag{4.17}$$

where μ is the background firing rate. Note that, here μ is set to be the same across all channels. β^c is the weight of states for each channel. In practice, both the parameters governing the firing rate μ and $\boldsymbol{\beta} = \{\beta^c\}_{c=1}^{C}$, and the governing state equation parameters α and ρ are unknown. In this work the noise variance σ_ϵ^2 is assumed to be fixed, and we are hence faced with the problem of having to estimate a set of unknown parameters $\boldsymbol{\theta} \in \mathbb{R}^d, d = C + 3$ with $\boldsymbol{\theta} = \{\alpha, \rho, \mu, \beta^1, \beta^2, \ldots, \beta^C\}$ in addition to an underlying hidden state x_k. Note that, unlike the previous chapter, we jointly estimation α and β^c. The identifiability problem can be solved with a very informative prior on α or β^c.

4.2.2 Variational posteriors

The variational framework for the inference in the SSPP is developed in a similar way to Beal (2003). Let $\mathcal{X}_K, \mathcal{Y}_K$ be the set of states and observed data points respectively, $\mathcal{X}_K = \{x_i\}_{i=0}^{K}$ and $\mathcal{Y}_K = \{\mathbf{y}_i\}_{i=1}^{K}$, where $\mathbf{y}_k = \{y_k^c\}_{c=1}^{C}$. The problem pivots on finding an approximation to the true posterior $p(\mathcal{X}_K, \boldsymbol{\theta}|\mathcal{Y}_K) \approx q(\mathcal{X}_K, \boldsymbol{\theta})$ such that the variational free energy (or log marginal likelihood) is maximised (Attias, 1999). The approximation is carried out by imposing independence between partitioned variables in the joint distribution. This is a well-known drawback when employing variational Bayesian methods; however, the ensuing factorisation is rewarded with significant computational savings.

In this work, the approximate (joint) posterior is assumed to be a product of Gaussian distributions

$$q(\mathcal{X}_K, \boldsymbol{\theta}) = q(\mathcal{X}_K)q(\boldsymbol{\theta}) = q(\mathcal{X}_K)q(\rho|\alpha)q(\alpha)q(\mu)\prod_{i=1}^{C} q(\beta^i). \tag{4.18}$$

The dependency between the ρ and α parameters is retained since the interaction terms between them which appear when deriving the log posterior distribution are relatively easy to compute. As a result, α and ρ are dealt with jointly and without loss in generality we redefine the set $\boldsymbol{\theta} = \{(\alpha, \rho), \mu, \beta^1, \beta^2, \ldots, \beta^C\}$. The optimal choice for the variational posteriors $q(\mathcal{X}_K)$ and $q(\boldsymbol{\theta})$ is then given by optimal solution for variational method in equation 4.7,

$$q(\mathcal{X}_K) \propto \exp(\langle \ln p(\mathcal{X}_K, \mathcal{Y}_K, \boldsymbol{\theta})\rangle_{q(\boldsymbol{\theta})}), \tag{4.19a}$$

$$q(\theta^i) \propto \exp(\langle \ln p(\mathcal{X}_K, \mathcal{Y}_K, \boldsymbol{\theta})\rangle_{q(\mathcal{X}_K)q(\boldsymbol{\theta}^{/i})}), \tag{4.19b}$$

where θ^i is the i^{th} component in $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{/i}$ is the set of all $\boldsymbol{\theta}$ excluding θ^i. The notation $\langle\cdot\rangle_{p(x)}$ denotes the expectation operator taken with respect to $p(x)$. In the standard case of linear-Gaussian dynamical systems, the variational posteriors can be computed exactly (Beal, 2003). For the model under consideration, because of the form of the observation process, this is not possible. However, the non-Gaussian densities which become apparent in the subsequent derivations are unimodal with respect to the underlying states and parameters, and simulation studies have shown that they can be reasonably approximated by Gaussian densities. We take advantage of this property and introduce approximations in a way similar to Smith and Brown (2003) (see also Friston et al., 2007) to obtain analytically tractable forward and backward passes for state distribution updates and the subsequent parameter distribution updates.

4.2.3 Batch update of states

Evaluating equation 4.19a and linearising as in Smith and Brown (2003), one obtains the following equations governing the forward pass (see section B.1.1 of appendix B)

$$x_{k|k} = \tilde{x}_k + \tilde{\sigma}_k^2 \sum_{c=1}^{C} \left\{ \langle \beta^c \rangle_{q(\beta^c)} y_k^c - \Delta \langle \exp \mu \rangle_{q(\mu)} \frac{d}{dx_k} \left[\langle \exp x_k \beta^c \rangle_{q(\beta^c)} \right] \Big|_{x_k = x_{k|k}} \right\}, \tag{4.20a}$$

$$\sigma_{k|k}^2 = \left(\tilde{\sigma}_k^{-2} + \sum_{c=1}^{C} \left\{ \Delta \langle \exp \mu \rangle_{q(\mu)} \frac{d^2}{dx_k^2} \left[\langle \exp x_k \beta^c \rangle_{q(\beta^c)} \right] \Big|_{x_k = x_{k|k}} \right\} \right)^{-1}, \tag{4.20b}$$

where,

$$\begin{aligned}\frac{\tilde{x_k}}{\tilde{\sigma}_k^2} &= \Bigg((\sigma_{k-1|k-1}^{-2} + \langle\rho^2\rangle\sigma_\epsilon^{-2})^{-1}\langle\rho\rangle\sigma_\epsilon^{-2}\left[x_{k-1|k-1}\sigma_{k-1|k-1}^{-2} - \langle\rho\alpha\rangle I_k\sigma_\epsilon^{-2}\right] \\ &\qquad + \langle\alpha\rangle I_k\sigma_\epsilon^{-2}\Bigg), \\ \tilde{\sigma}_k^2 &= (\sigma_\epsilon^{-2} - \langle\rho\rangle^2(\sigma_{k-1|k-1}^{-2} + \langle\rho^2\rangle\sigma_\epsilon^{-2})^{-1}\sigma_\epsilon^{-4})^{-1}.\end{aligned} \tag{4.21}$$

Equation 4.20a is composed of two terms, the first pertaining to the underlying linear dynamical model, and the second to the observation point process. Considering the nonlinear form of equation 4.20a, it can be shown that if each $\beta^c > 0$ and $\langle{\beta^c}^2\rangle_{q(\beta^c)} \approx \langle\beta^c\rangle_{q(\beta^c)}^2$ the forward estimate tends to be lowered by a lack of events (indicative of a decreasing intensity). On the other hand, $y_k^c = 1$ tends to increase the estimated $x_{k|k}$. The effect of the number of output channels C is also apparent by evaluating the sum in equation 4.20b, from which it is easily seen that the precision $\sigma_{k|k}^{-2}$ increases with increasing C (assuming β^c is constant across all channels).

The forward state update equations do not depend on the actual values of the parameters, rather on their first and second moments under the approximating distribution. This averaging, which will be evident in all of the following update equations, is at the core of the 'mean field' variational algorithms which originated in statistical physics, where the interdependence between states were replaced by a dependence on the average (mean) value of the states. For conciseness, in equation 4.21 and in some of the following equations, the distributions with which the expectations are taken with respect to are omitted. The normal assumption for the variational distributions allow analytical computation of the expectations involved in the above equations.

In a similar fashion, a backward recursion on the data is computed in order to obtain variational smoothed state estimates (see section B.1.2 in appendix B). The resulting equations are given as

$$x_{k|K} = \sigma_{k|K}^2(x_{k|k}\sigma_{k|k}^{-2} + x_k^*\sigma_k^{*^{-2}}), \qquad \sigma_{k|K}^2 = (\sigma_{k|k}^{-2} + \sigma_k^{*^{-2}})^{-1}, \tag{4.22}$$

where,

$$\begin{aligned}\frac{x_k^*}{\sigma_k^{*2}} &= \Bigg(\langle\rho\rangle x'_{k+1}(\sigma_\epsilon^{-2} + \sigma'^{-2}_{k+1})^{-1}\sigma_\epsilon^{-2}\sigma'^{-2}_{k+1} + (\sigma_\epsilon^{-2} + \sigma'^{-2}_{k+1})^{-1}\langle\rho\rangle\langle\alpha\rangle I_{k+1}\sigma_\epsilon^{-4} \\ &\qquad - \langle\rho\alpha\rangle I_{k+1}\sigma_\epsilon^{-2}\Bigg), \\ \sigma_k^{*^2} &= (\langle\rho^2\rangle\sigma_\epsilon^{-2} - (\sigma_\epsilon^{-2} + \sigma'^{-2}_{k+1})^{-1}\langle\rho\rangle^2\sigma_\epsilon^{-4})^{-1},\end{aligned} \tag{4.23}$$

and where,

$$\begin{aligned} x'_{k+1} \;\; = \;\; & x_{k+1|k+1} + \sigma'^2_{k+1}\Bigg(\frac{x^*_{k+1} - x_{k+1|k+1}}{\sigma^{2*}_{k+1}} + \sum_{c=1}^{C} \Bigg\{ \langle \beta^c \rangle_{q(\beta^c)} y^c_{k+1} \\ & - \Delta \langle \exp \mu \rangle_{q(\mu)} \frac{d}{dx_{k+1}} \; \left[\langle \exp x_{k+1} \beta^c \rangle_{q(\beta^c)} \right] \Big|_{x_{k+1} = x_{k+1|k+1}} \Bigg\} \Bigg), \end{aligned} \tag{4.24a}$$

$$\sigma'^2_{k+1} = \left(\sigma^{*^{-2}}_{k+1} + \sum_{c=1}^{C} \left\{ \Delta \langle \exp \mu \rangle_{q(\mu)} \frac{d^2}{dx^2_{k+1}} \; \left[\langle \exp x_{k+1} \beta^c \rangle_{q(\beta^c)} \right] \Big|_{x_{k+1} = x_{k+1|k+1}} \right\} \right)^{-1} . \tag{4.24b}$$

In equations 4.24a and 4.24b, the Gaussian approximation is carried out around the filtered estimate to give a closed form solution. As a consequence of this the forward and backward passes need to be carried out sequentially. On completing the backward pass, if the initial state distribution is not known it may be updated by setting $x_{0|0} = x_{0|K}$ and variance $\sigma^2_{0|0} = \sigma^2_{0|K}$ (see Beal, 2003).

Equation 4.20a is not available in closed form and needs to be solved by a deterministic optimisation method. One can take advantage of the facts that the equation has a unique solution, and that the prior $x_{k|k-1}$ (obtained from the predictive density) can be used as a good initialisation for $x_{k|k}$ to solve the optimisation method in an efficient manner. In practice it was found that replacing the state variable on the right hand side by the prior (to obtain a closed form solution) gave very good results and a marked decrease in computational requirements.

The required statistics needed for updating the parameter variational posteriors are $\langle x_k x_{k+1} \rangle_{q(\mathcal{X}_K)}$, $\langle x_k^2 \rangle_{q(\mathcal{X}_K)}$ and $\langle x_k \rangle_{q(\mathcal{X}_K)}$ for all time k. The only quantity which is not readily available from the above is the first of these expectations. To obtain this we invert the precision of the approximate pairwise marginal $p(x_k, x_{k+1} | \mathcal{Y}_K)$ to get

$$\langle x_k x_{k+1} \rangle_{q(\mathcal{X}_K)} = \langle \rho \rangle \sigma_\epsilon^{-2} \left((\sigma^{-2}_{k|k} + \langle \rho^2 \rangle \sigma_\epsilon^{-2}) \sigma^{l^{-2}} - \langle \rho \rangle^2 \sigma_\epsilon^{-4} \right)^{-1} + x_{k+1|K} x_{k|K}, \tag{4.25}$$

where

$$\sigma^{l^2} = \left(\sigma^{*^{-2}}_{k+1} + \sigma_\epsilon^{-2} + \sum_{c=1}^{C} \left\{ \Delta \langle \exp \mu \rangle \; \frac{d^2}{dx^2_{k+1}} \; \left[\langle \exp x_{k+1} \beta^c \rangle_{q(\beta^c)} \right] \Big|_{x_{k+1} = x_{k+1|K}} \right\} \right)^{-1} . \tag{4.26}$$

After computing the state sufficient statistics, one can update the parameter variational posteriors as described next.

4.2.4 Batch update of parameters

Equation 4.19b gives the updates for the parameter distributions. As a direct consequence of the underlying linear state evolution model, the optimal variational estimates for α and ρ become identical to those in a linear dynamical system and so we refer the reader to Beal (2003) for details. The estimation of μ and β_c is somewhat more involved and we refer the reader to section B.2 of appendix B for their treatment. Denoting the means and variances of μ and β_c as $\hat{\mu}, \hat{\beta}_c$ and $\sigma^2_\mu, \sigma^2_{\beta_c}$ respectively we have that

$$\hat{\beta^c} = \beta^c_p + \sigma^2_{\beta^c_p} \sum_{i=1}^{K} \left(y^c_i \langle x_i \rangle_{q(\mathcal{X}_K)} - \Delta \langle \exp \mu \rangle_{q(\mu)} \frac{d}{d\beta^c} \left[\langle \exp x_i \beta^c \rangle_{q(\mathcal{X}_K)} \right] \Big|_{\beta^c = \hat{\beta}^c} \right), \tag{4.27a}$$

$$\sigma^2_{\beta^c} = \left(1/\sigma^2_{\beta^c_p} + \Delta \langle \exp \mu \rangle_{q(\mu)} \sum_{i=1}^{K} \left[\frac{d^2}{d\beta^{c^2}} \langle \exp x_i \beta^c \rangle_{q(\mathcal{X}_K)} \Big|_{\beta^c = \hat{\beta}^c} \right] \right)^{-1}, \tag{4.27b}$$

and

$$\hat{\mu} = \mu_p + \sigma^2_{\mu_p} \sum_{i=1}^{K} \sum_{c=1}^{C} \left(y^c_i - \Delta \exp(\hat{\mu}) \langle \exp(\beta^c x_i) \rangle_{q(\mathcal{X}_K) q(\beta^c)} \right), \tag{4.28a}$$

$$\sigma^2_\mu = \left(1/\sigma^2_{\mu_p} + \Delta \exp(\hat{\mu}) \sum_{i=1}^{K} \sum_{c=1}^{C} \langle \exp(\beta^c x_i) \rangle_{q(\mathcal{X}_K) q(\beta^c)} \right)^{-1}, \tag{4.28b}$$

where the subscript p denotes *prior*. For this study we have taken Gaussian prior distributions over all parameters, with hyperparameters assumed to be known.

All expectations in the above equations are standard except for $\langle \exp(\beta^c x_i) \rangle_{q(\mathcal{X}_K) q(\beta^c)}$, which can be calculated using moment generating functions (see section B.2 of appendix B). As is standard in VB estimation, updates for specific variables depend on the expectations of the remaining variables, leading to a natural iterative algorithm. Convergence can be easily assessed by monitoring changes in the free energy or in the statistics of the variational distributions.

4.3 Online VB for SSPP

The above off-line VB algorithm can be extended for use in an online scenario with some modifications. Using a standard technique in dual filtering (Wan and Nelson, 2001), a time evolution model for the parameters is introduced

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \mathbf{e}_k, \tag{4.29}$$

where $\mathbf{e}_k \in \mathbb{R}^d$ is additive white Gaussian noise with diagonal covariance matrix $\mathbf{\Sigma^e_k} \in \mathbb{R}^{d \times d}$, which is also time varying (see below). Let $\Theta_k = \{\boldsymbol{\theta}_i\}_{i=1}^k$. Equations 4.16 and 4.15 now become

$$\lambda^c_k = \exp(\mu_k + \beta^c_k x_k), \tag{4.30}$$

$$x_k = \rho_k x_{k-1} + \alpha_k I_k + \epsilon_k. \tag{4.31}$$

The *online variational posteriors* are given as

$$q(\mathcal{X}_k) \propto \exp(\langle[\ln p(\mathcal{X}_k, \mathcal{Y}_k, \Theta_k)]\rangle_{q(\Theta_k)}), \tag{4.32a}$$

$$q(\theta_k^i) \propto \exp\left[\langle[\ln p(\mathcal{X}_k, \mathcal{Y}_k, \Theta_k)]\rangle_{q(\mathcal{X}_k)q(\Theta_k^{/\theta_k^i})}\right], \tag{4.32b}$$

where $q(\Theta_{\backslash i,k})$ is the joint $q(\Theta_k)$ without the variable $\theta_{i,k}$. We choose the following variational posteriors

$$\begin{aligned} q(\mathcal{X}_k, \Theta_k) &\approx q(\mathcal{X}_k) \prod_{j=1}^{k} q(\boldsymbol{\theta}_j) \\ &= q(\mathcal{X}_k) q(\Theta_k), \end{aligned} \tag{4.33}$$

that is, the parameters are approximated to be conditionally independent in time through the product distribution $q(\Theta_k)$. To facilitate recursion, the parameter variational posteriors are further restricted to be the filtered distributions. We hence redefine $q(\Theta_k)$ as follows

$$q(\Theta_k) = \prod_{j=1}^{k} q(\boldsymbol{\theta}_j | \mathcal{Y}_j). \tag{4.34}$$

At each time step the distribution $q(\mathcal{X}_k)$ and $q(\boldsymbol{\theta}_k)$ are variational posteriors in the conventional sense. We refer to $\{q(\boldsymbol{\theta}_j)\}_{j=1}^{k-1}$ as the *restricted* variational posteriors, as is typical in restricted variational Bayes methods (Šmídl and Quinn, 2006). A novel result for dual VB filtering is presented in the following theorem.

Theorem 4.1. *For the SSPP described by equations 4.16 and 4.15, given the factorisation in equation 4.33, the restriction in equation 4.34 and the maximisers in equations 4.32a and 4.32b, the recursive updates for the state and parameter variational distributions $q(\mathcal{X}_k)$ and $q(\boldsymbol{\theta}_k)$ are given by*

$$q(x_k) \propto \int dx_{k-1} q(x_{k-1}) \exp(\langle \ln p(x_k | x_{k-1}, \boldsymbol{\theta}_k) p(\mathbf{y}_k | x_k, \boldsymbol{\theta}_k)\rangle_{q(\boldsymbol{\theta}_k)}), \tag{4.35a}$$

$$\begin{aligned} q(\theta_k^i) \propto \exp(\langle \ln p(\mathbf{y}_k | x_k, \boldsymbol{\theta}_k) p(x_k | x_{k-1}, \boldsymbol{\theta}_k)\rangle_{q(\mathcal{X}_k)q(\boldsymbol{\theta}_k^{/i})}) \\ \times \quad \exp(\langle \ln p(\theta_k^i | \theta_{k-1}^i)\rangle_{q(\theta_{k-1}^i)}), \quad i = 1 \dots d. \end{aligned} \tag{4.35b}$$

Proof. To show that the above holds we start off by the new joint distribution of $p(\mathcal{X}_k, \Theta_k, \mathcal{Y}_k)$, which writes

$$p(\mathcal{X}_k, \Theta_k, \mathcal{Y}_k) = p(x_0, \boldsymbol{\theta}_0) \prod_{t=1}^{k} p(x_t | x_{t-1}, \boldsymbol{\theta}_k) p(\mathbf{y}_t | x_t, \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}).$$

Now, consider the variational approximation of the state marginal:

$$\begin{aligned} q(x_k) &\propto \int d\mathcal{X}_{k-1} \exp(\langle \ln p(\mathcal{X}_k, \Theta_k, \mathcal{Y}_k)\rangle) \\ &= \exp(\langle \ln p(\mathbf{y}_k|x_k, \boldsymbol{\theta}_k)\rangle) \int d\mathcal{X}_{k-1} \exp(\langle \ln p(x_k|x_{k-1}, \boldsymbol{\theta}_k) p(\mathcal{X}_{k-1}, \Theta_k, \mathcal{Y}_{k-1})\rangle). \end{aligned} \tag{4.36}$$

Due to fact that $\boldsymbol{\theta}_k$ is conditionally independent with $\mathcal{X}_{k-1}$ and $\mathcal{Y}_{k-1}$ given $\boldsymbol{\theta}_{k-1}$, we have

$$p(\mathcal{X}_{k-1}, \Theta_k, \mathcal{Y}_{k-1}) = p(x_0, \boldsymbol{\theta}_0) p(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}) \prod_{t=1}^{k-1} p(x_t|x_{t-1}, \boldsymbol{\theta}_k) p(\mathbf{y}_t|x_t, \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$$

The second term of the integrand in equation 4.36 can also be expanded, and by treating the conditional parameter distributions as constants relative to the distribution of interest, it can be shown that

$$\begin{aligned} q(x_k) \propto &\exp(\langle \ln p(\mathbf{y}_k|x_k, \boldsymbol{\theta}_k)\rangle) \int dx_{k-1} \Bigg(\exp(\langle \ln p(x_k|x_{k-1}, \boldsymbol{\theta}_k)\rangle) \\ &\times \Bigg[\exp(\langle \ln p(\mathbf{y}_{k-1}|x_{k-1}, \boldsymbol{\theta}_{k-1})\rangle) \int d\mathcal{X}_{k-2} \exp(\langle \ln p(x_{k-1}|x_{k-2}, \boldsymbol{\theta}_{k-1})\rangle) \\ &\times \exp(\langle \ln p(\mathcal{X}_{k-2}, \Theta_{k-1}, \mathcal{Y}_{k-2})\rangle) \Bigg] \Bigg). \end{aligned} \tag{4.37}$$

Recall that since the approximate parameter posteriors have been restricted to be conditional on the data up to the instant in which they were estimated, the distributions of the parameters do not need to be recomputed using the latest data which is available. In particular for any function $\psi(\cdot)$

$$\mathbb{E}_{q_{\Theta_k}}[\psi(\boldsymbol{\theta}_{k-1})] = \mathbb{E}_{q_{\boldsymbol{\theta}_{k-1}|\mathcal{Y}_{k-1}}}[\psi(\boldsymbol{\theta}_{k-1})], \tag{4.38}$$

which was computed at the previous time step. Hence, in comparison to equation 4.36, it is clear that the terms in the square brackets of equation 4.37 constitute the exact variational posterior marginal of the state at the previous time instant to give equation 4.35a. Equation 4.35b follows by application of the chain rule on equation 4.32b where the joint $p(\mathcal{X}_{k-1}, \Theta_{\backslash i,k}, \mathcal{Y}_{k-1})$ is constant relative to the distribution of interest. □

The above does not constitute an online algorithm in the strictest sense since equations 4.35a and 4.35b are evidently coupled, and, as in the off-line case, some form of iteration between the solutions is required for convergence. However, iterations are required only between the marginals at the last time instant, making the algorithm fast and efficient, and in practice few iterations often suffice. It should also be noted that the online algorithm does not necessarily maximise the variational free energy as the restricted VB assumption is an approximation to the correct update rule. Based on this result one can find the update equations for the variational posteriors of interest.

4.3.1 Online update of states

By comparing equation 4.32a to equation 4.19a, it is evident that $q(\mathcal{X}_k)$ is updated exactly in the same way as in the off-line case, the only differences being that

(i) the expectations are in this case taken with respect to the parameters at the *present* time instant.

(ii) from equation 4.35b it is evident that only the variational posteriors over the pair (x_k, x_{k-1}) are required to be evaluated at each time step.

The parameter distribution updates require the smoothed distribution of x_{k-1} at each time instant, and the cross-covariance between (x_k, x_{k-1}) (see section B.1.2 of appendix B). The required sufficient statistics are denoted as follows

$$U_k = I_k^2,\ \ G_k = I_k \mathbb{E}_{q_{x_{k-1}}}[x_{k-1}],\ \ M_k = I_k \mathbb{E}_{q_{x_k}}[x_k],\ \ W_k = \mathbb{E}_{q_{x-1}}[x_{k-1}^2],\ \ S_k = \mathbb{E}_{q_{x_k} q_{x_{k-1}}}[x_k x_{k-1}]\rangle. \tag{4.39}$$

4.3.2 Online update of parameters

The variational posteriors can be obtained using similar computations to those for the off-line case. The only alteration is the time evolution of the parameters driven by the noise $\mathbf{e}_k$. Following standard practice in signal processing (Wan and Nelson, 2001), $\mathbf{e}_k$ is modelled to have zero mean and slowly varying variance

$$\langle e_k^{i^2} \rangle = (\eta^i)^{-1} \sigma^2_{\theta^i_{k-1}}, \quad i = 1 \ldots d, \tag{4.40}$$

where the term $\eta^i \in (0, 1], i \in \{\alpha, \rho, \mu, \beta^1, \beta^2, \ldots, \beta^C\}$, is a user-defined forgetting factor. Effectively the prior is no longer fixed (although an additional fixed prior can be introduced), rather, according to the parameter evolution equation, it is a Gaussian distribution with the mean of the previous estimate and a precision weighted by η.

Online update of $q(\alpha_k)$: The joint distribution $q(\rho_k, \alpha_k)$ is first found from equation 4.35b. The conditional distribution may then be obtained from $q(\rho_k, \alpha_k) = q(\rho_k|\alpha_k)\bar{p}(\alpha_k)$. Ignoring terms independent of ρ_k, this is given as

$$\ln q(\rho_k|\alpha_k) = \langle \ln p(\rho_k|\rho_{k-1}) \rangle + \langle \ln p(x_k|x_{k-1}, \rho_k, \alpha_k) \rangle, \tag{4.41}$$

from which the following expressions are obtained

$$\begin{aligned}\sigma^2_{\rho_k|\alpha_k} &= \left[\frac{1}{\eta^{\rho^{-1}}\sigma^2_{\rho_{k-1}}} + \frac{W_k}{\sigma^2_\epsilon}\right]^{-1}, \\ \langle\rho_k\rangle_{q(\rho_k|\alpha_k)} &= \sigma^2_{\rho_k|\alpha_k}\left[\frac{S_k}{\sigma^2_\epsilon} + \frac{\hat{\rho}_{k-1}}{\eta^{\rho^{-1}}\sigma^2_{\rho_{k-1}}} - \frac{\alpha_k G_k}{\sigma^2_\epsilon}\right].\end{aligned} \tag{4.42}$$

The marginal $q(\alpha_k)$ may be found by marginalising ρ_k from the $q(\rho_k|\alpha_k)\bar{p}(\alpha_k)$. This is given by

$$\begin{aligned}\sigma^2_{\alpha_k} &= \left(\frac{1}{\eta^{\alpha^{-1}}\sigma^2_{\alpha_{k-1}}} + \frac{U_k}{\sigma^2_\epsilon} - \frac{\sigma^2_{\rho_k|\alpha_k}G_k^2}{\sigma^4_\epsilon}\right)^{-1}, \\ \hat{\alpha}_k &= \sigma^2_{\alpha_k}\left(\frac{\hat{\alpha}_{k-1}}{\eta^{\alpha^{-1}}\sigma^2_{\alpha_{k-1}}} + \frac{M_k}{\sigma^2_\epsilon} - \frac{G_k}{\sigma^2_\epsilon}\left[\frac{S_k\sigma^2_{\rho_k|\alpha_k}}{\sigma^2_\epsilon} + \frac{\sigma^2_{\rho_k|\alpha_k}\hat{\rho}_{k-1}}{\eta^{\rho^{-1}}\sigma^2_{\rho_{k-1}}}\right]\right).\end{aligned} \tag{4.43}$$

Online update of $q(\rho_k)$: The statistics over ρ_k are obtained by marginalising α_k from the joint distribution as follows

$$q(\rho_k) = \int d\alpha_k q(\rho_k|\alpha_k)q(\alpha_k). \tag{4.44}$$

The variational posterior $q(\rho_k|\alpha_k)$ is computed from equation 4.42 and $q(\alpha_k)$ is known from equation 4.43. The marginalisation is straightforward to give the following expressions

$$\begin{aligned}\sigma^2_{\rho_k} &= \sigma^2_{\rho_k|\alpha_k} + \frac{\sigma^2_{\alpha_k}\sigma^4_{\rho_k|\alpha_k}G_k^2}{\sigma^4_\epsilon}, \\ \hat{\rho}_k &= \sigma^2_{\rho_k|\alpha_k}\left[\frac{S_k}{\sigma^2_\epsilon} + \frac{\hat{\rho}_{k-1}}{\eta^{\rho^{-1}}\sigma^2_{\rho_{k-1}}} - \frac{G_k\hat{\alpha}_k}{\sigma^2_\epsilon}\right].\end{aligned} \tag{4.45}$$

Online update of $q(\mu_k)$: Following equation 4.35b and ignoring terms independent of μ_k, we have that

$$\begin{aligned}\ln q(\mu_k) &= \langle \ln p(\mu_k|\mu_{k-1})\rangle + \langle \ln p(\mathbf{y}_k|x_k, \mu_k, \boldsymbol{\beta}_k)\rangle, \\ &= -\frac{\langle(\mu_k - \mu_{k-1})^2\rangle}{2\eta^{\mu^{-1}}\sigma^2_{\mu_{k-1}}} + \langle\sum_{c=1}^{C} y_k^c[\mu_k + \beta_k^c x_k] - \exp(\mu_k)\exp(\beta_k^c x_k)\Delta\rangle.\end{aligned} \tag{4.46}$$

where the state evolution density is omitted since it is independent of μ_k. On expanding and approximating around $\hat{\mu}_k$, the following update equations are obtained

$$\begin{aligned}
\hat{\mu_k} &= \hat{\mu}_{k-1} + \eta^{\mu^{-1}} \sigma^2_{\mu_{k-1}} \sum_{c=1}^{C} \left(y_k^c - \Delta \langle \exp(\beta_k^c x_k) \rangle \exp(\hat{\mu}_k) \right), \\
\sigma^2_{\mu_k} &= \left(\eta^{\mu} \sigma^{-2}_{\mu_{k-1}} + \Delta \exp(\hat{\mu_k}) \sum_{c=1}^{C} \langle \exp(\beta_k^c x_k) \rangle \right)^{-1} .
\end{aligned} \tag{4.47}$$

Online update of $q(\beta_k^c)$: Following the same reasoning as that for updating $q(\mu_k)$ the resulting equations are given as

$$\begin{aligned}
\hat{\beta}_k^c &= \hat{\beta}_{k-1}^c + \eta^{\beta^{-1}} \sigma^2_{\beta^c_{k-1}} \left(y_k^c \langle x_k \rangle - \Delta \langle \exp \mu_k \rangle \frac{d}{d\beta_k^c} \left[\langle \exp x_k \beta_k^c \rangle \right] |_{\beta_k^c = \hat{\beta}_k^c} \right), \\
\sigma^2_{\beta_k^c} &= \left(\eta^{\beta^c} \sigma^{-2}_{\beta^c_{k-1}} + \Delta \langle \exp \mu_k \rangle \left[\frac{d^2}{d\beta_k^{c2}} \langle \exp x_k \beta_k^c \rangle |_{\beta_k^c = \hat{\beta}_k^c} \right] \right)^{-1} .
\end{aligned} \tag{4.48}$$

4.4 Results

4.4.1 Offline VB

We first considered the *off-line* inference problem illustrated by Smith and Brown (2003) and Yuan and Niranjan (2010), where outputs from multiple neurons sharing a common hidden state were simulated. We set the number of neurons $C = 20$ and considered the response to a spike input applied every 1s over a time interval of $T = 10$s with a sampling rate of 100Hz. We set $\rho = 0.8, \alpha = 4, \mu = 0$ and β^c to a randomly generated number in the interval [0.9 1.1].

All priors on the parameters and states, except for that over β^c, were set to normal distributions with variances: $\sigma^2_{\rho_p} = 5, \sigma^2_{\alpha_p} = 50, \sigma^2_{\mu_p} = 1$. The prior over β^c was set to a normal distribution centred at 1 with a 99% confidence between 0.7 and 1.3; this was done to remedy the identifiability issues stemming from the fact that the likelihood (equation 4.16) involves only the product $\beta^c x_k$ (a problem related to the parameter offsetting observed by Smith and Brown (2003)).

The estimation of the state variational posterior describing the latent process using the VBEM algorithm can be seen in Figure 4.2 where at each time step the variational posterior's mean and 99% confidence limits are given. Graphical results for the corresponding estimation of the 23 unknown parameters are shown in Figure 4.3, showing rapid convergence to good estimates.

We further compared our results to those obtained using EM (Smith and Brown, 2003) and those given by a Gibbs sampler on the same data set (see chapter 5 for details).

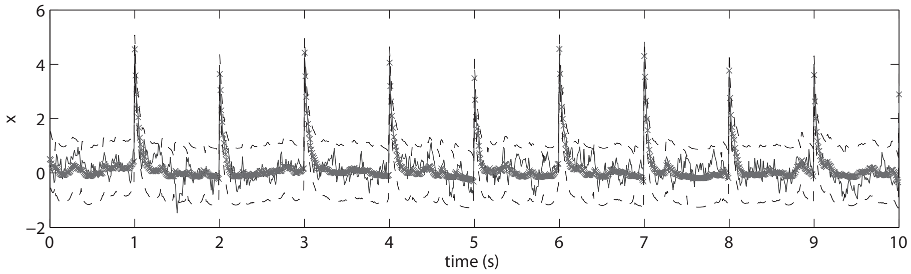
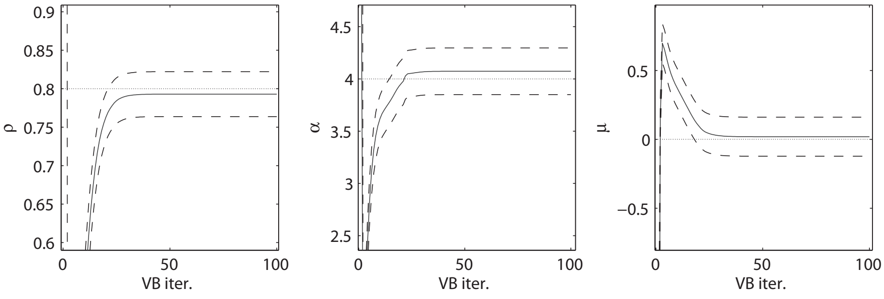
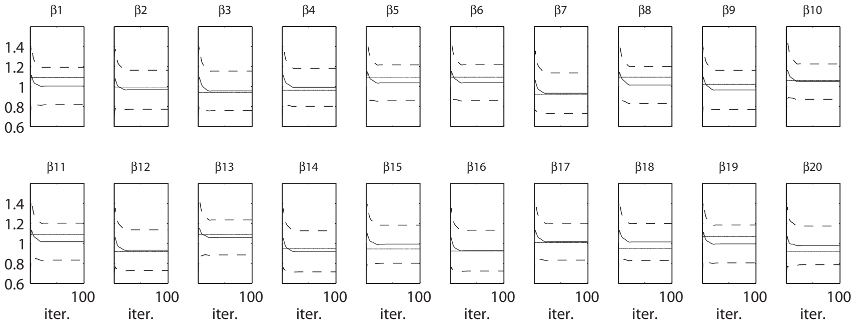


Figure 4.2: True state (continuous unmarked line) and mean estimated state (marked line) as given by the batch VB algorithm in the final iteration. The true state lies consistently within the 99% confidence intervals (dashed line).



(a)



(b)

Figure 4.3: Mean estimates (continuous varying line) and 99% confidence intervals (dashed line) over 100 VBEM iterations for the parameters (a) $\rho, \alpha,$ μ and (b) $\beta^c, c = 1 \ldots 20$ using the batch VB algorithm. The parameters converge in distribution to reasonable estimates irrespective of the initial conditions and the true (solid level line) values are seen to lie well within the 99% confidence intervals at steady-state.

Table 4.1: Parameter estimation by the EM algorithm, Gibbs sampler and VBEM algorithm. Unless stated, $\boldsymbol{\beta}$ was fixed to the true value during simulation.

θ	True	EM	Gibbs	VBEM	VBEM (free $\boldsymbol{\beta}$)
ρ	0.80	0.82	0.79 ± 0.06	0.79 ± 0.03	0.79 ± 0.03
α	4.00	4.08	3.81 ± 0.48	4.04 ± 0.22	4.07 ± 0.22
μ	0.00	-0.19	0.06 ± 0.24	0.01 ± 0.14	0.02± 0.14
avr($\boldsymbol{\beta}$)	1.00	-	-	-	0.99 ± 0.19

Table 4.2: Mean squared maximum KS distances for the 20 neurons with different event-rate models (lower is better) for one data set. Unless stated, $\boldsymbol{\beta}$ was fixed to the true value during simulation.

	Gibbs	VBEM	EM	VBEM (free $\boldsymbol{\beta}$)	EM (free $\boldsymbol{\beta}$)	SW
MSE	0.0046	0.0055	0.0076	0.0077	0.0136	0.0336

To avoid identifiability issues, we also ran experiments with $\boldsymbol{\beta}$ fixed to its true value. Table 4.1 shows that all methods are effective in estimating parameters for this data, with Gibbs and VBEM also providing confidence intervals which are in good agreement with the true values. It took 5s for the EM algorithm (50 iterations), 12s for the VBEM algorithm (50 iterations), and 279s for the Gibbs sampler (5000 iterations) to converge.[2]

A more informative test of the model's performance is its ability to capture the spike train distribution. A quantitative measure of this can be achieved using the time-rescaling theorem of Brown et al. (2002) in conjunction with a Kolmogorov-Smirnov (KS) test, following the same procedure as Smith and Brown (2003) and Barbieri et al. (2005). As a goodness of fit measure, the mean squared maximum distance between the model rate and the true rate over all output channels was found. The results for this KS measure on a synthetic data set are given in Table 4.2; for completeness we also compare with a sliding window (SW) empirical rate-estimator of 100ms width which is often used in these applications (Riehle et al., 1997). The Bayesian methods (VBEM and Gibbs sampler) are seen to obtain a considerable better goodness of fit than the EM algorithm (which in turn is much better than the simple SW heuristic), indicating that retaining distributional information over the parameters does lead to an improvement in the modelling of the spike distribution. To further validate the result we ran a 2-sample t-test on the KS-measures from 20 different data sets. The mean-square maximum KS distance for all these runs was 0.0070 for the VBEM algorithm (fixed $\boldsymbol{\beta}$) and 0.0089 for

[2] Simulations carried out on an Intel®Core$^{\text{TM}}$2 Quad Q6600 @ 2.40GHZ with 4GB of RAM

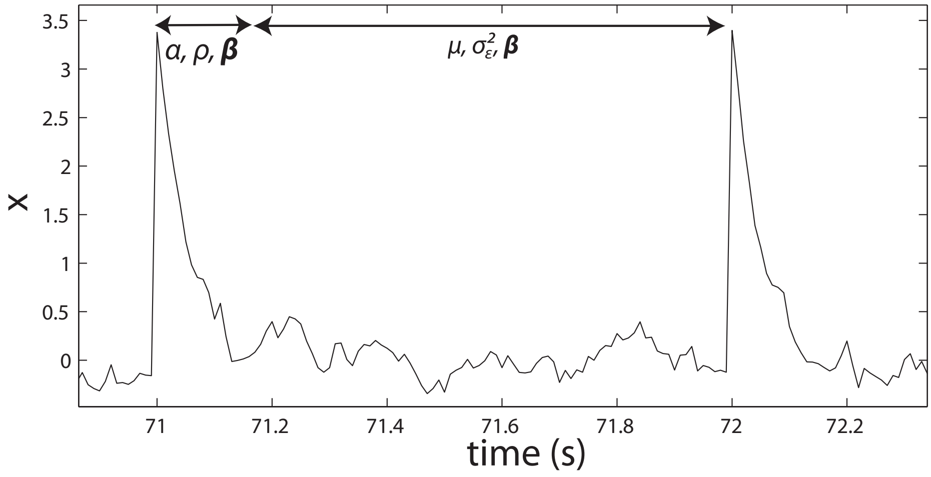


Figure 4.4: Selective updating of parameter estimates in an online framework is carried out in accordance to the areas where the state bears most information about the relevant parameters of interest. In this case, the narrow stretch close to an input spike bears a lot of information on the state decay factor ρ and the input gain α. The noise parameters μ and σ_ϵ^2, on the other hand, are more evident in regions of no input.

the EM algorithm (also with fixed $\boldsymbol{\beta}$). The test rejected the null hypothesis that the decrease in error occurred by chance at the 5% significance level.

4.4.2 Online VB

In this section we present a simulation study of the VB online algorithm derived in section 4.3. The nature of the data typical in these types of models requires some further intervention for correct estimation when using filters. In regions where no input is present, the observed events in the output are predominantly due to the background firing rate μ and state noise σ_ϵ^2 and there is little or no information about ρ and α in these regions. On the other hand, the deterministic component of the hidden state governs the output in time intervals close to an input. In these areas there is significant information about ρ and α. Parameter distributions were thus updated only in regions where there is ample information about the relevant parameters, as illustrated in Figure 4.4. This procedure is standard in online filtering in other areas, e.g. speech enhancement by spectral subtraction, in which noise levels are estimated in regions of the signal where speech is not present (Boll, 1979).

For this study we assumed $C = 20$ and that $\boldsymbol{\beta}$ and μ were predetermined from a previous off-line analysis and assumed to be constant. The choice of the forgetting factors was carried out by trial and error such that a parameter change could be tracked without compromising stability in the online estimates. We subsequently chose $\eta^\rho = 0.8$ and

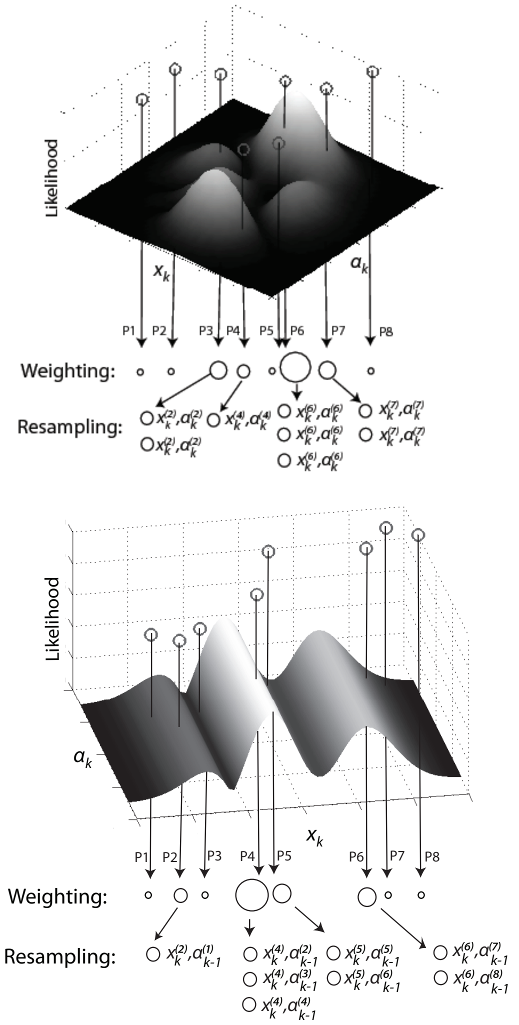

Likelihood
x_k
α_k
P1
P2
P3
P4
P5
P6
P7
P8
Weighting:
Resampling:
$x_k^{(2)},\alpha_k^{(2)}$
$x_k^{(2)},\alpha_k^{(2)}$
$x_k^{(4)},\alpha_k^{(4)}$
$x_k^{(6)},\alpha_k^{(6)}$
$x_k^{(6)},\alpha_k^{(6)}$
$x_k^{(6)},\alpha_k^{(6)}$
$x_k^{(7)},\alpha_k^{(7)}$
$x_k^{(7)},\alpha_k^{(7)}$
Likelihood
α_k
x_k
P1
P2
P3
P4
P5
P6
P7
P8
Weighting:
Resampling:
$x_k^{(2)},\alpha_{k-1}^{(1)}$
$x_k^{(4)},\alpha_{k-1}^{(2)}$
$x_k^{(4)},\alpha_{k-1}^{(3)}$
$x_k^{(4)},\alpha_{k-1}^{(4)}$
$x_k^{(5)},\alpha_{k-1}^{(5)}$
$x_k^{(5)},\alpha_{k-1}^{(6)}$
$x_k^{(6)},\alpha_{k-1}^{(7)}$
$x_k^{(6)},\alpha_{k-1}^{(8)}$

Figure 4.5: (top) The likelihood function is used to appropriately weight the particles (P#) representing the posterior distribution which are then resampled into N particles of equal weight. (bottom) In this case the likelihood is practically independent of α_k and thus the weighing and resampling steps solely depend on the x_k component of the particles. In order to maintain the posterior distribution with fewer particles than would be necessary otherwise, after resampling, the prior particle parameter set is redistributed with equal weight among the resampled particles. The figures (top) and (bottom) correspond to the two areas marked in Figure 4.4 respectively (likelihood surfaces shown are for illustration only and do not represent actual surfaces).

$\eta^{\alpha} = 0.9$. The dual VB filter was compared to a standard particle filter (PF) which makes use of an augmented state vector $\mathbf{z}_k = [x_k, \rho_k, \alpha_k]^T$ and implements what is effectively a standard sequential importance sampling with resampling (SISR) algorithm (see Kitagawa, 1998; Doucet et al., 2001; de Freitas et al., 2000, and details therein). The prior distribution was chosen as the importance distribution so that the weights were updated in time according to the likelihood. That is, if $w_k^{(i)}$ denotes the weight of the i^{th} particle at time k, and $\mathbf{z}_k^{(i)}$ the i^{th} particle at time k, the weight update is given as

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(\mathbf{y}_k|\mathbf{z}_k^{(i)}) \tag{4.49}$$

The selective estimation process described above was adapted to the PF by using *selective* SISR as shown in Figure 4.5. In this figure, the case where only the input gain α_k and the state x_k are to be estimated at one time instant is shown. In regions where α_k does not affect the likelihood (or *importance factor*), propagation and subsequent resampling only takes place in the state-space. The respective parameter marginal distribution is retained and propagated through time unchanged. Formally, after resampling, in this region we have that the full joint distribution is given by

$$p(\alpha_k, x_k|\mathcal{Y}_k) \approx \frac{1}{N}\sum_{i=1}^{N} \delta\begin{pmatrix} x_k - x_k^{(i)} \\ \alpha_k - \alpha_{k-1}^{(i)} \end{pmatrix}, \tag{4.50}$$

and the subsequent marginal distribution by

$$p(\alpha_k|\mathcal{Y}_k) = \int dx_k p(\alpha_k, x_k|\mathcal{Y}_k) \approx \frac{1}{N}\sum_{i=1}^{N} \delta(\alpha_k - \alpha_{k-1}^{(i)}) \approx p(\alpha_{k-1}|\mathcal{Y}_{k-1}), \tag{4.51}$$

where N denotes the number of particles and $\delta(\cdot)$ the delta Dirac mass. Finally, for both VB filter and PF the initial state prior is set to be $\mathcal{N}(0, \sigma_\varepsilon^2/(1 - rho_0^2))$, where ρ_0 corresponds to a initial guess. This setting is the same as the one in Smith and Brown (2003)

The result for the successful tracking a sudden change in the true value of ρ from 0.8 to 0.6 by both the VB filter, and the PF with $N = 5000$ particles, is shown in Figure 4.6 (the number of particles chosen was the minimum required for consistent posterior distribution approximations across several trials). The results corroborate each other, indicating that the VB filter gives a realistic description of what can be termed the ground truth.[3] Complete results are shown in Table 4.3. Despite the parameter distributions estimated being very similar, the PF took on the order of 10 times longer than the VB filter to execute. Indeed, the computational time required by the PF in this

[3] Since filters are particularly sensitive to the chosen parameter evolution model, the online parameter posterior distribution is highly dependent on i) the forgetting factor in the VB filter and ii) the corresponding parameter noise statistics in the PF. In the latter case, the variance was tuned to give a similar learning rate as that of the VB filter.

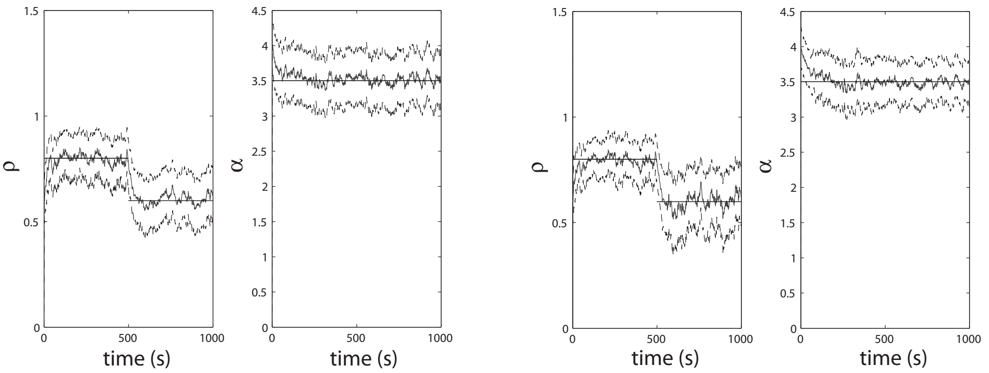


Figure 4.6: Online tracking of a sudden change in the true parameter (level black line) ρ at time $t = 500s$. In this example μ and $\boldsymbol{\beta}$ where assumed constant and known from previous off-line analysis of the system. The 99% confidence intervals (outer traces) are seen to enclose the true value upon the filter reaching a steady behaviour both for the (left) VB filter and (right) particle filter with 5000 particles.

Table 4.3: Comparison between the VB filter and a PF for SSPP with 5000 particles

	ρ	α	$mean(\hat{\rho_k})$	$mean(\sigma_{\rho_k})$	$mean(\hat{\alpha_k})$	$mean(\sigma_{\alpha_k})$
VB (t $\leq$ 500)	0.8	3.5	0.799	0.037	3.52	0.13
PF (t $\leq$ 500)			0.797	0.031	3.49	0.12
VB (t > 500)	0.6	3.5	0.607	0.041	3.51	0.12
PF (t > 500)			0.602	0.049	3.49	0.10

example was more than the duration of the data stream itself, rendering it impractical for the real-time application of this case study scenario.

4.5 Discussion

In this chapter, we proposed a variational Bayesian method for filtering and smoothing within state-space models with point process observations. This class of models provides a physiologically plausible signal processing framework for event-based observations, and has proved a popular framework for analysing and decoding spike-train data. Experiments on realistic simulated data show that the Bayesian treatment (either by VB or by computationally expensive sampling methods) does indeed lead to an improvement in the modelling of the spike train distribution, while retaining very good accuracy in estimating the parameter posteriors. A major contribution of this work is the introduction of an online estimation framework. This allows considerable computational savings, potentially paving the way for real-time biomedical applications. It also allows for the monitoring of online changes in system mode of operation, as exemplified in our case study of neural responses to different taste stimuli.

Filtering of doubly stochastic point process may be carried out directly in continuous time (Snyder and Miller, 1991), in which case the stochastic intensity is generally assumed to be a function of a diffusion (Segall et al., 1975; Solo, 2000). Solutions are given as normalised or unnormalised conditional intensities which take the form of partial differential equations. Analytical solutions can be found in special cases, such as when the intensity is given as the square of an Ornstein-Uhlenbeck process (Boel and Benes, 1980). Nonetheless, in the general case, computational expensive numerical methods are still required for implementation purposes. The case is similar in discrete-time. Manton et al. (1999), for instance, showed that an exact (strictly) finite-dimensional filter exists for equation 4.15 with $\lambda_k^c = (\gamma_k x_k)^2, \gamma_k > 0.\forall k$, but the treatment quickly becomes intractable for different forms of the intensity. This work, and most of the literature which focuses on state estimation from point process observations, utilise models where the parameters are assumed to be known. This motivates the investigation into new, more versatile methods such as that first proposed by Smith and Brown (2003), now extended into a variational setting in this paper.

VB provides a neat, deterministic way for approximating the joint posterior distribution online. We have compared the performance of the VB filter to a stochastic approximation method through a standard PF and seen that it performs very well comparatively with a marked decrease in computational requirements. Previous to this work, sequential Monte Carlo (SMC) methods had already been applied to the state estimation problem in the SSPP framework. In Ergün et al. (2007), the underlying states dynamics were modeled by a random walk process but the underlying parameters were assumed to be known. The authors introduced point process adaptive filters (Eden et al., 2004) for proposing new particles to increase the computational efficiency. The method showed good performance both on a synthetic dataset and on a real dataset, where the problem of tracking the evolution of a hippocampal spatial receptive field was studied. The extension of these results to online parameter learning SMC approaches (see also Storvik, 2002) was thus a natural step. It should be noted that the highly linear substructure (through the underlying AR latent process) also allows for Rao-Blackwellised PFs (Doucet et al., 2000a) to be applied. In this case the state forward filtering step may by approximated by that of Smith and Brown (2003) or Fahrmeir and Tutz (1994). However, preliminary results show that even in this case, SMC methods may still prove to be too time consuming for any interesting biomedical application where data needs to be handled in real time.

Online variational Bayes was first proposed for model selection of static conjugate-exponential (CE) models by Sato (2001), where the recursive updates at each time step describe the solution to successive maximisations of a discounted free energy. Unlike the online VB algorithm presented here, Sato's approach has the advantage that the algorithm behaves as a stochastic approximator for the maximum expected free energy for a fixed amount of data points, obviating the requirement of VB iterations at each datum. However, Sato's algorithm relies on the favourable properties of the family of static CE models which SSPP clearly do not form part of. Moreover, it is envisioned that the algorithm proposed in this work finds potential in its application to a continuous stream of data, where the maximisation of a fixed objective functional loses its appeal. In the proposed solution we have made use of a static forgetting factor to discount the use of "old" information in the estimation process. This bears similarity to the time-varying discount factor for variable learning rate as used in Sato's work.

The application to online tracking suggests naturally an extension to consider state-space models with switching parameters, which would formally incorporate abrupt changes in mode of operation into the model. These have proved a popular tool in biomedical applications (see, for instance, Quinn et al., 2008), and would also be suitable for the application described in chapter 7. This additional complexity is likely, however, to come at some computational cost. A further interesting extension would be to improve on the observation model by using more advanced models for spike generation, such as integrate and fire; parameter estimation within these models has recently been explored using search-type algorithms (MacGregor et al., 2009), but the complexity of the likelihood model means that it is likely that considerable work will be needed before they can be used in signal processing applications.

Chapter 5

Markov chain Monte Carlo methods

In this chapter, we use the more powerful Markov chain Monte Carlo (MCMC) methods (see for example Neal (1993)), which offer asymptotically exact posteriors, to explore different approximate schemes of inference for SSPP models. For this, we consider a number of variants of MCMC methods suitable for SSPP models, thereby enriching the array of tools for inference and parameter estimation. In particular, we examine two recently advanced MCMC methods – particle marginal Metropolis-Hastings (PMMH) algorithm (Andrieu et al., 2010) and Riemann manifold Hamiltonian Monte Carlo (RMHMC) method (Girolami and Calderhead, 2011), as well as the traditional Hamiltonian Monte Carlo (HMC) method (Duane et al., 1987). These methods are demonstrated on a synthetic dataset, showing significant efficiency improvement when compared with a commonly used single-site update Gibbs sampler. In these simulations, RMHMC outperforms the others with high efficiency scores and comparable computational costs. In addition, we provide two approximate MCMC methods, embedding deterministic approximations into the sampling procedure.

5.1 Basic MCMC methods

5.1.1 The Metropolis-Hastings algorithm

Assume $p(x)$ is a target distribution, also known as the *stationary distribution.* Let x^* denote every state that can be reached from the current state x, with a conditional probability density function $q(x^*|x)$. Define an acceptance ratio,

$$a(x^*, x) = 1 \wedge \frac{p(x^*)q(x|x^*)}{p(x)q(x^*|x)} = \min\left(1, \frac{p(x^*)q(x|x^*)}{p(x)q(x^*|x)}\right). \tag{5.1}$$

The Metropolis-Hastings (MH) algorithm does the following (Metropolis et al., 1953; Hastings, 1970):

Algorithm 5.1. The Metropolis-Hasting algorithm

Input: $x^{(0)}, q(\cdot|\cdot)$ and M (number of MCMC iterations)

Output: $x^{(1:M)}$

```
for $i = 1, \dots, M$ do
    Draw $x^* \sim q(x^*|x^{(i)})$
    Compute $a(x^*, x^{(i)}) = 1 \wedge \frac{p(x^*)q(x^{(i)}|x^*)}{p(x^{(i)})q(x^*|x^{(i)})}$
    Draw $u \sim \mathcal{U}(0, 1)$
    if $u < a(x^*, x^{(i)})$ then
        $x^{(i+1)} = x^*$
    else
        $x^{(i+1)} = x^{(i)}$
    end if
end for
```

The MH procedure in algorithm 5.1 serves as a fundamental building block of the MCMC family. Most MCMC methods can be seen as special cases of MH, where $q(x^*|x)$ is constructed in special ways to achieve specific goals. Among them, the simplest, and perhaps the most widely used one, is when $q(\cdot|\cdot)$ satisfies $q(x^*|x) = q(x|x^*)$ and the acceptance ratio reduces to $1 \wedge \frac{p(x^*)}{p(x)}$. This procedure is known as the Metropolis algorithm (Metropolis et al., 1953). A particular example is the Gaussian $q(x^*|x) = \mathcal{N}(x^*|x, \sigma^2)$, which is known as the random-walk MH (RWMH).

5.1.2 Detailed balance condition

Equation 5.1 gives an appealing property to the MH algorithm, that is it respects the *detailed balance condition.* If the produced Markov chain is both aperiodic and irreducible, this condition is *sufficient* (not necessary) to ensure that the Markov chain converges to a stationary distribution $p(x)$. In the following, we explain the condition in detail, which follows the standard description in Geyer (2010).

Further, let $K(x^*|x)$ be the kernel defining the transition probability for the Markov chain constructed by MH, and $p(x)$ be the stationary distribution. The detailed balance is that, if

$$p(x)K(x^*|x) = p(x^*)K(x|x^*), \tag{5.2}$$

then the Markov chain is reversible w.r.t. $p(x)$. To see it leaves the stationary distribution invariant, integrating over x^* for both sides of equation (5.2), we have

$$p(x) = \int_{x^*} p(x^*)K(x|x^*)dx^*. \tag{5.3}$$

Then, the stationary distribution $p(x)$ is also known as the equilibrium distribution of the Markov chain constructed by $K(x^*|x)$.

More precisely, the MH transition kernel expands to

$$K(x^*|x) = q(x^*|x)a(x^*|x) + \delta(x^* - x)\int_{x^*} q(x^*|x)(1 - a(x^*|x))dx^*. \tag{5.4}$$

When a proposed x^* is rejected, then $x^* = x$. The condition is naturally satisfied. When a proposed x^* is accepted, the detailed balance condition translates to

$$p(x)q(x^*|x)a(x^*, x) = p(x^*)q(x|x^*)a(x, x^*). \tag{5.5}$$

Expanding the right hand side of the equation, we have

$$\begin{aligned} p(x)q(x^*|x)a(x^*, x) &= p(x)q(x^*|x)\left(1 \wedge \frac{p(x^*)q(x|x^*)}{p(x)q(x^*|x)}\right) \\ &= p(x)q(x^*|x) \wedge p(x^*)q(x|x^*) \\ &= p(x^*)q(x|x^*)\left(1 \wedge \frac{p(x)q(x^*|x)}{p(x^*)q(x|x^*)}\right) \\ &= p(x^*)q(x|x^*)a(x, x^*). \end{aligned} \tag{5.6}$$

It is clear that the acceptance probability $a(\cdot, \cdot)$, or more specifically, the Hastings ratio

$$\begin{aligned} r(x^*, x) &= \frac{p(x^*)q(x|x^*)}{p(x)q(x^*|x)} \\ &= \frac{1}{r(x, x^*)}, \end{aligned} \tag{5.7}$$

which holds the equality.

5.1.3 The Gibbs sampler

Introduced by Geman and Geman (1984); Gelfand and Smith (1990), a particular choice of $q(x^*|x)$, and perhaps the most popular one, leads to the Gibbs Sampler. Consider the same target distribution; this time the variable of interest is a vector $\mathbf{x} = [x_1, \cdots, x_n]^{\mathrm{T}}$. Let the MH proposal be

$$q(\mathbf{x}^*|\mathbf{x}) = p(x_j^*|\mathbf{x}_{\backslash j})\mathbb{I}(\mathbf{x}^*_{\{\backslash j\}} = \mathbf{x}_{\{\backslash j\}}), \tag{5.8}$$

where $\mathbb{I}$ is the indicator function. The subscript $\backslash j$ presents the set $[1, \cdots, j-1, j+1, \cdots, n]$. Plugging in such an proposal distribution, the acceptance probability becomes

$$\begin{aligned} a(\mathbf{x}^*, \mathbf{x}) &= 1 \wedge \frac{p(\mathbf{x}^*)q(\mathbf{x}|\mathbf{x}^*)}{p(\mathbf{x})q(\mathbf{x}^*|\mathbf{x})} \\ &= 1 \wedge \frac{p(\mathbf{x}^*)p(x_j|\mathbf{x}_{\backslash j})}{p(\mathbf{x})p(x_j^*|\mathbf{x}_{\backslash j}^*)} \\ &= 1 \wedge \frac{p(\mathbf{x}_{\backslash j}^*)}{p(\mathbf{x}_{\backslash j})} = 1. \end{aligned} \tag{5.9}$$

Hence, the proposed move is always accepted. Specifically, the Gibbs sampler does the following:

Algorithm 5.2. The Gibbs sampler

Input: $\mathbf{x}^{(0)}$

Output: $\mathbf{x}^{(1:M)}$

```
1: for $i = 1, \dots, M$ do
2:     Draw $x_1^{(i+1)} \sim p(x_1|x_2^{(i)}, x_3^{(i)}, \cdots, x_n^{(i)})$
3:     Draw $x_2^{(i+1)} \sim p(x_2|x_1^{(i+1)}, x_3^{(i)}, \cdots, x_n^{(i)})$
4:         ⋮
5:     Draw $x_j^{(i+1)} \sim p(x_j|x_1^{(i+1)}, \cdots, x_{j-1}^{(i+1)}, x_{j+1}^{(i)}, \cdots, x_n^{(i)})$
6:         ⋮
7:     Draw $x_n^{(i+1)} \sim p(x_n|x_1^{(i+1)}, x_2^{(i+1)}, \cdots, x_{n-1}^{(i+1)})$
8: end for
```

The distribution $p(x_j|\mathbf{x}_{\backslash j})$ is called the full conditional distribution, which, according to Bayes' rule is:

$$p(x_j|\mathbf{x}_{\backslash j}) = \frac{p(x_j, \mathbf{x}_{\backslash j})}{\int_{x_j} p(x_j, \mathbf{x}_{\backslash j}) dx_j} \propto p(x_j, \mathbf{x}_{\backslash j}) \tag{5.10}$$

Example 5.1. *Take simple linear regression as an example, which fits a line to n data points with $\{x_i, y_i\}$ coordinates. Particularly, y_i is the response to the input x_i, obeying*

$$y_i|x_i, \beta_0, \beta_1, \sigma_\varepsilon^2 \sim \mathcal{N}(y_i|\beta_0 + \beta_1 x_i, \sigma_\varepsilon^2). \tag{5.11}$$

The inference problem is: Given the data points $\mathbf{x} = [x_1, \cdots, x_n]^{\mathrm{T}}$ and $\mathbf{y} = [y_1, \cdots, y_n]^{\mathrm{T}}$, what is the posterior distribution of the parameters? Assume only β_0 and β_1 are of interest, define $\boldsymbol{\beta} = [\beta_0, \beta_1]^{\mathrm{T}}$, then the posterior is written as

$$p(\boldsymbol{\beta}|\mathbf{x}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\beta})p(\boldsymbol{\beta})}{p(\mathbf{y}|\mathbf{x})}. \tag{5.12}$$

If we use a prior $\mathcal{N}(\boldsymbol{\beta}|\mathbf{0}, \sigma_0^2\mathbf{I})$, the posterior distribution $p(\boldsymbol{\beta}|\mathbf{x}, \mathbf{y})$ becomes available in closed form:

$$p(\boldsymbol{\beta}|\mathbf{x}, \mathbf{y}) = \mathcal{N}(\boldsymbol{\beta}|\mathbf{m}, \boldsymbol{\Sigma}) \tag{5.13}$$

where

$$\mathbf{m} = \begin{bmatrix} m_0 \\ m_1 \end{bmatrix} = \sigma_\varepsilon^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{y}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \Sigma_{00} & \Sigma_{01} \\ \Sigma_{10} & \Sigma_{11} \end{bmatrix} = \left(\sigma_0^{-2}\mathbf{I} + \sigma_\varepsilon^{-2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}. \tag{5.14}$$

$\Phi = [\boldsymbol{\phi}(x_1), \cdots, \boldsymbol{\phi}(x_n)]^{\mathrm{T}}$, *where* $\boldsymbol{\phi}_i = [1, x_i]^{\mathrm{T}}$. *Such formulation represents a basis function view of the simple linear regression problem.*

The posterior is a Gaussian which can be directly sampled from. However, let us assume that we cannot achieve that. The MH algorithm and the Gibbs sampler are the only options.

For MH, the RWMH is employed, such that $q(\boldsymbol{\beta}^*|\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta}^*|\boldsymbol{\beta}, \sigma^2\mathbf{I})$. The acceptance probability becomes

$$\begin{aligned} a(\boldsymbol{\beta}^*, \boldsymbol{\beta}) &= 1 \wedge \frac{p(\boldsymbol{\beta}^*|\mathbf{x}, \mathbf{y})q(\boldsymbol{\beta}|\boldsymbol{\beta}^*)}{p(\boldsymbol{\beta}|\mathbf{x}, \mathbf{y})q(\boldsymbol{\beta}^*|\boldsymbol{\beta})} \\ &= 1 \wedge \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\beta}^*)p(\boldsymbol{\beta}^*)q(\boldsymbol{\beta}|\boldsymbol{\beta}^*)}{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\beta})p(\boldsymbol{\beta})q(\boldsymbol{\beta}^*|\boldsymbol{\beta})} \\ &= 1 \wedge \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\beta}^*)p(\boldsymbol{\beta}^*)}{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\beta})p(\boldsymbol{\beta})} \end{aligned} \tag{5.15}$$

Here, the Hastings ratio shows another appealing feature; that is, the normalising constant $p(\mathbf{y}|\mathbf{x})$, which makes the Bayesian inference intractable, cancels out. The computational effort at each MCMC iteration reduces to the product of likelihood and prior. Both terms are normally well-defined[1] *and easy to compute.*

For the Gibbs sampler, the target distribution is split into two full conditionals, $p(\beta_0|\beta_1, \mathbf{x}, \mathbf{y})$ and $p(\beta_1|\beta_0, \mathbf{x}, \mathbf{y})$, each of which can be easily obtained from the Gaussian conditional rule.

$$\beta_0|\beta_1, \mathbf{x}, \mathbf{y} \sim \mathcal{N}(\beta_0|m_{0|1}, \Sigma_{0|1}), \quad \beta_1|\beta_0, \mathbf{x}, \mathbf{y} \sim \mathcal{N}(\beta_1|m_{1|0}, \Sigma_{1|0}), \tag{5.16}$$

where

$$\begin{aligned} m_{0|1} &= m_0 + \Sigma_{01}\Sigma_{11}^{-1}(\beta_1 - m_1), \quad \Sigma_{0|1} = \Sigma_{00} - \Sigma_{01}\Sigma_{11}^{-1}\Sigma_{10}, \\ m_{1|0} &= m_1 + \Sigma_{10}\Sigma_{00}^{-1}(\beta_0 - m_0), \quad \Sigma_{1|0} = \Sigma_{11} - \Sigma_{10}\Sigma_{00}^{-1}\Sigma_{01}, \end{aligned} \tag{5.17}$$

[1] The likelihood of ODE/PDE-based models are normally not available in closed form. In such a scenario, both MCMC and VB methodologies failed. One needs to resort to the *approximate Bayesian computation* (ABC) methods.

To illustrate the performances of these two methods, we generate 100 *data points with a true model:* $\beta_0 = 2$*,* $\beta_1 = 7$*,* $\sigma_\varepsilon^2 = 10$ *and* $\sigma_0^2 = 20$*. In addition,* $q(\boldsymbol{\beta}^*|\boldsymbol{\beta})$ *is set to be* $\mathcal{N}(\boldsymbol{\beta}^*|\boldsymbol{\beta}, 0.2\mathbf{I})$*. For both samplers,* 200 *samples are collected after* 100 *burn-in samples. The inference results are shown in figure 5.1 where RWMH and Gibbs differ fundamentally in approaching to the target posterior distribution* $p(\boldsymbol{\beta}|\mathbf{x}, \mathbf{y})$*. The RWMH moves jointly in both* β_0 *and* β_1 *guided by* $q(\boldsymbol{\beta}^*|\boldsymbol{\beta})$*. In this case,* $q(\boldsymbol{\beta}^*|\boldsymbol{\beta})$ *appears as a circle, which means the new proposed moves in the two directions are identically probable. Such* $q(\boldsymbol{\beta}^*|\boldsymbol{\beta})$ *is said to be isotropic. On the other hand, the Gibbs sampler is clearly more adaptive to the underlying target distribution, due to the fact that the full conditionals encoded information form the data. This is evident from equation 5.17. As a result, in this example, the Gibbs sampler provides better coverage to the posterior with a small number of samples.*

5.2 MCMC methods for SSPP

The full Bayesian treatment for the state-space point process model targets the joint posterior distribution of the underlying states and parameters, given all observations $p(x_{0:K}, \boldsymbol{\theta}|\mathbf{y}_{1:K})$. Following Bayes's rule, such a distribution can be obtained by

$$p(x_{0:K}, \boldsymbol{\theta}|\mathbf{y}_{1:K}) = \frac{p(\mathbf{y}_{1:K}|x_{0:K}, \boldsymbol{\theta})p(x_{0:K})p(\theta)}{p(\mathbf{y}_{1:K})}. \tag{5.18}$$

To approach this posterior distribution, two extensions of the Gibbs sampler are crucially important.

The blocked Gibbs sampler The first extension is known as the *Blocked Gibbs sampler*, in which a set of variables are updated jointly in turn, while leaving the stationary distribution invariant. In the context of SSPP or the state-space model in general, this feature provides a blueprint for constructing inference mechanisms.

Algorithm 5.3. The blocked Gibbs sampler for state-space models

Input: $x_{1:K}^{(0)}$, $\boldsymbol{\theta}^{(0)}$

Output: $x_{1:K}^{(1:M)}$, $\boldsymbol{\theta}^{(1:M)}$

```
for $m = 1$ to $M$ do
    Draw $x_{1:K}^{(m)} \sim p(x_{1:K}|y_{1:K}, \boldsymbol{\theta}^{(m-1)})$
    Draw $\boldsymbol{\theta}^{(m)} \sim p(\boldsymbol{\theta}|y_{1:K}, x_{1:K}^{(m-1)})$
end for
```

This procedure converges to a equilibrium distribution $p(x_{0:K}, \boldsymbol{\theta}|y_{1:K})$, which is the desired posterior in any state-space model. In addition, the blocking procedure mitigates a problem in the Gibbs sampler. That is, the Gibbs sampler, which updates one variable

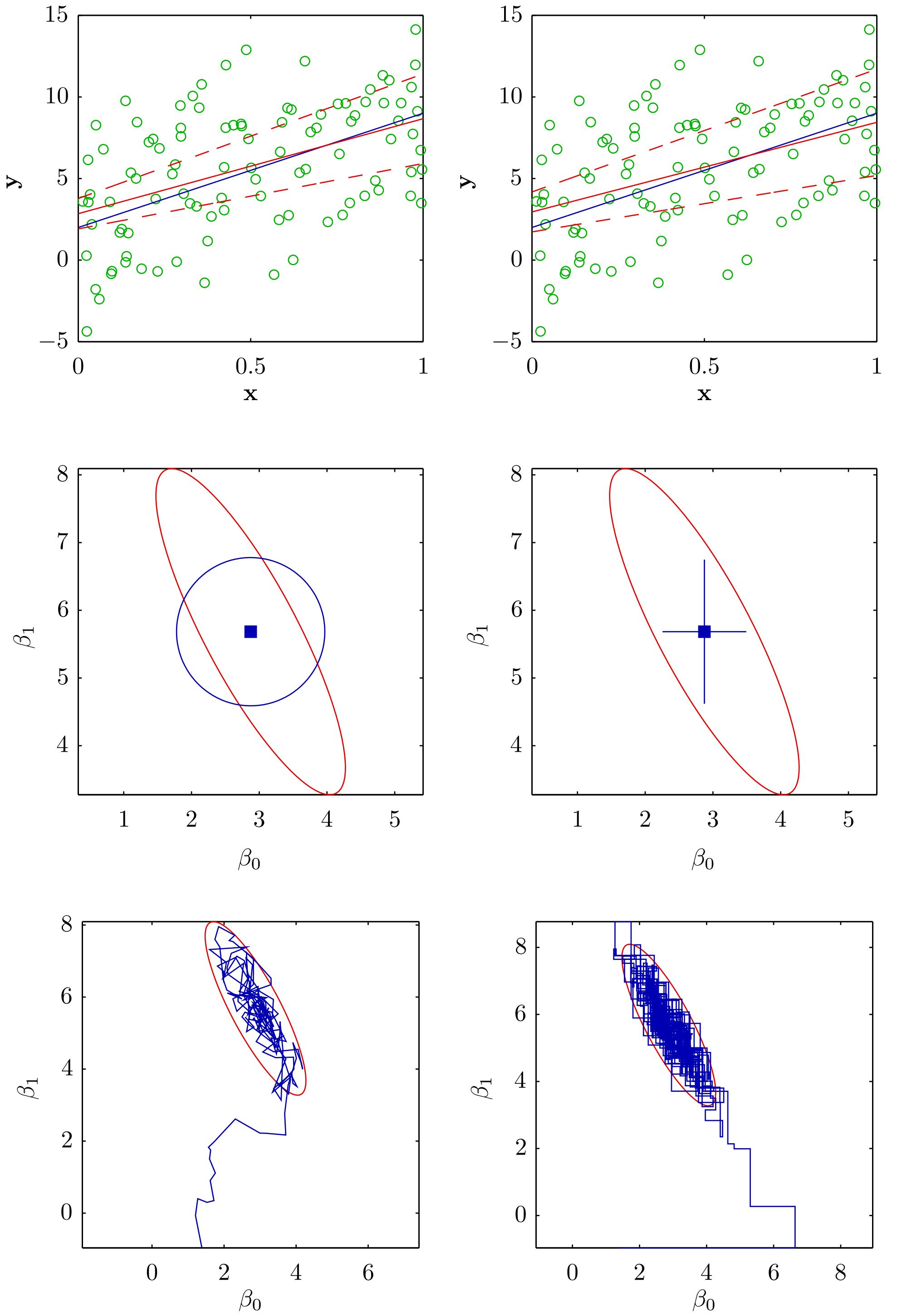


Figure 5.1: *Top*: Data points (*green points*), true model (*blue line*), inferred model from posterior mean (*red line*) and 95% confident intervals (*red dashed lines*). *Middle*: Illustration of the moving mechanisms for RWMH (*left*) and Gibbs sampler (*right*). The target posterior $p(\boldsymbol{\beta}|\mathbf{x},\mathbf{y})$ evaluated at 95% confident level is shown as *red ellipse* in both cases. The small *blue square* in both left and right panels indicates the starting point a particular move. For the RWMH, the *blue circle* is $q(\boldsymbol{\beta}^*|\boldsymbol{\beta})$ evaluated at 95% confidence intervals. For the Gibbs sampler, the *blue lines* show the 95% confidence intervals of $p(\beta_0|\beta_1,\mathbf{x},\mathbf{y})$ and $p(\beta_1|\beta_0,\mathbf{x},\mathbf{y})$. *Bottom*: Realisations of two chains constructed by RWMH (*left*) and Gibbs sampler (*right*).

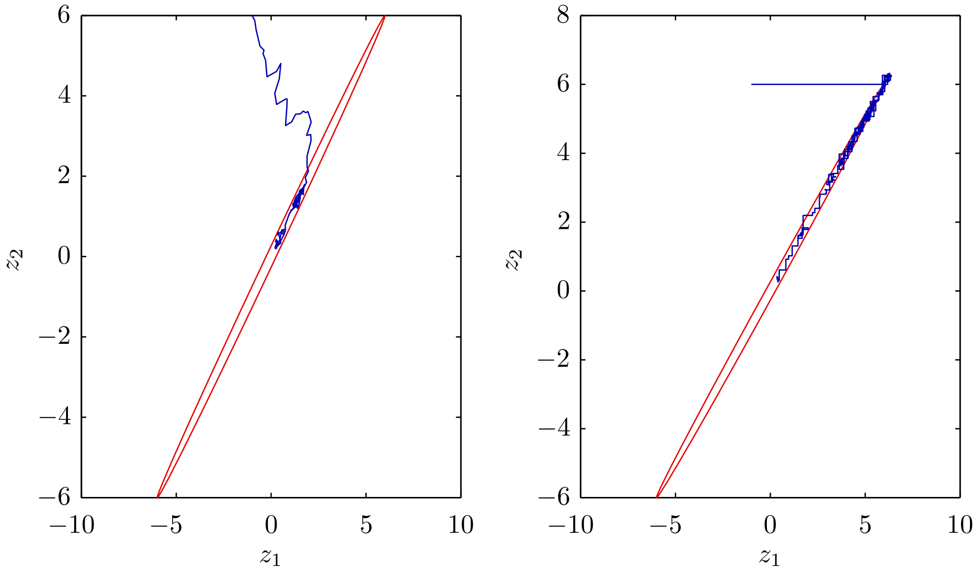


Figure 5.2: Illustrations of RWMH (*left*) and Gibbs sampler (*right*) trapped in a local region of a 2-dimensional Gaussian distribution as described in example 5.2. The trajectories in both cases represent 200 samples. The red ellipse is the 95% confident level of target distribution.

at a time, moves slowly in narrow target distributions. This is illustrated by example 5.2 and figure 5.2.

Example 5.2. *In this example, we sample from a* 2*-dimensional Gaussian distribution in which the two variables* $\mathbf{z} = [z_1, z_2]^{\mathrm{T}}$ *are highly correlated (correlation factor:* 0.999*). Precisely, the log-density is written as*

$$\log p(\mathbf{z}) = -\frac{1}{2}(\mathbf{z}-\mathbf{m})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{z}-\mathbf{m}) + const, \tag{5.19}$$

$$\mathbf{m} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} 6 & 0.999 \times 6 \\ 0.999 \times 6 & 6 \end{bmatrix}. \tag{5.20}$$

Figure 5.2 shows that samples produced by both RWMH and the Gibbs sampler are highly concentrated in a local region of the target distribution.

The Metropolis-within-Gibbs sampler The block structure often makes direct sampling from the full conditional distributions infeasible. This problem also arises when the likelihood and prior are not a conjugate pair. For problems of this kind, one can use the *Metropolis-within-Gibbs sampler* (MWG):

Algorithm 5.4. The Metropolis-within-Gibbs sampler

Input: $\mathbf{x}^{(0)}$

Output: $\mathbf{x}^{(1:M)}$

```
for $i = 1$ to $M$ do
    for $j = 1$ to $n$ do
```

```
        Draw $x_j \sim p(x_j|\mathbf{x}_{\backslash j})$ via MH algorithm
    end for
end for
```

5.2.1 The single-site update Gibbs sampler

According to the scheme in algorithm 5.3, the target distribution splits into two full conditional distributions: $p(x_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\mathbf{y}_{1:K}, x_{0:K})$. A convenient approach to drawing samples from these two distributions is the *single-site update Gibbs sampler* (ssGibbs) (Geweke and Tanizaki, 2001). The update procedure of this sampler can be read from its name, such that the variables are updated one at a time. When targeting the state posterior, the desired full conditionals are $p(x_k|x_{\backslash k}, \mathbf{y}_{1:K}, \boldsymbol{\theta})$.

$$
\begin{aligned}
& p(x_k|x_{\backslash k}, \mathbf{y}_{1:K}, \boldsymbol{\theta}) \\
& \propto \begin{cases} p(x_k|x_{k-1}, \alpha, \rho, \sigma_\varepsilon^2) p(x_{k+1}|x_k, \alpha, \rho, \sigma_\varepsilon^2) \prod_{c=1}^{C} p(y_{c,k}|x_k, \mu, \beta) & \text{if } k = 1, \ldots, K-1 \\ p(x_k|x_{k-1}, \alpha, \rho, \sigma_\varepsilon^2) \prod_{c=1}^{C} p(y_{c,k}|x_k, \mu, \beta) & \text{if } k = K \end{cases}
\end{aligned}
\tag{5.21}
$$

According to the SSPP formulation, direct sampling from the above is infeasible. We use a MWG sampler with a random walk proposal to facilitate the sampling. [2].

Likewise, the parameters posterior $p(\boldsymbol{\theta}|x_{0:K}, \mathbf{y}_{1:K})$ is also approached by single-site updates. The conditionals in this case are

$$p(\rho|\mathbf{y}_{1:K}, x_{0:K}, \alpha, \sigma_\varepsilon^2, \mu, \beta) \propto p(x_0) \prod_{k=1}^{K} p(x_k|x_{k-1}, \rho, \alpha, \sigma_\varepsilon^2) p(\rho), \tag{5.22}$$

$$p(\alpha|\mathbf{y}_{1:K}, x_{0:K}, \rho, \sigma_\varepsilon^2, \mu, \beta) \propto p(x_0) \prod_{k=1}^{K} p(x_k|x_{k-1}, \rho, \alpha, \sigma_\varepsilon^2) p(\alpha), \tag{5.23}$$

$$p(\mu|\mathbf{y}_{1:K}, x_{0:K}, \rho, \alpha, \sigma_\varepsilon^2, \beta_{1:C}) \propto \prod_{c=1}^{C} \prod_{k=1}^{K} p(y_{c,k}|x_k, \mu, \beta_c) p(\mu), \tag{5.24}$$

$$p(\beta_c|\mathbf{y}_{1:K}, x_{0:K}, \rho, \alpha, \sigma_\varepsilon^2, \mu) \propto \prod_{k=0}^{K} p(y_{c,k}|x_k, \mu, \beta_c) p(\beta_c). \tag{5.25}$$

Given the above conditionals the ssGibbs is shown in algorithm 5.5. Additionally, with the parameter settings in example 5.3, we illustrate the performance of ssGibbs for SSPP.

Algorithm 5.5. The single-site update Gibbs sampler for the SSPP

```
Input: $x_{1:K}^{(0)}$, $\boldsymbol{\theta}^{(0)}$
Output: $x_{1:K}^{(1:M)}$, $\boldsymbol{\theta}^{(1:M)}$
  for $m = 1, \ldots, M$ do
    for $k = 1, \ldots, K$ do
```

[2] Another Using the state transition density as the proposal gives similar results in this case.

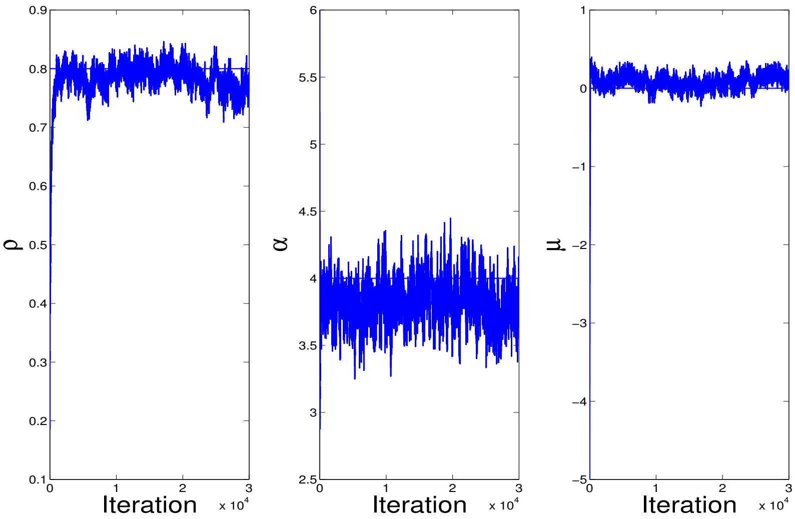


Figure 5.3: The trajectory of the ssGibbs sampler for the unknown parameters ρ, α, and μ. The solid level line denotes the true parameter value.

```
            Sample $x_k^{(m)} \sim p(x_k | x_{\backslash k}^{(m-1)}, \mathbf{y}_{1:K}, \boldsymbol{\theta}^{(m-1)})$
        end for
        Sample $\boldsymbol{\theta}^{(m)} \sim p(\boldsymbol{\theta} | \mathbf{y}_{1:K}, x_{1:K}^{(m)})$
    end for
```

Example 5.3. $\rho = 0.8$, $\alpha = 4$,$\sigma_\varepsilon^2 = 0.01$, $\mu = 0$ *and* β_c *are random variables from* $[0.9, 1.1]$. *The time interval for the model is set to be* $10ms$, *and total observation time length is* $20s$. *The external stimulus is given at every second.*

In this chapter we focus on the case when σ_ε^2 and β_c are fixed.

Figure 5.3 shows the performance with 30000 samples. The burn-in period of the sampler is about 5000 iterations in both estimation tasks. The autocorrelation function (ACF) of the obtained samples, as shown in figure 5.4, reveals that the chains are poorly mixed.

Specifically, as seen in to figure 5.4, ρ converges poorly when comparing α and μ. This is due to the poor mixing performance in sampling from the state posterior, which is evident in the effective sample size performance (shown later in this chapter). The ACFs of α and μ decrease fast and become stationary around 0 within 200 lags. Especially, the ACF of μ drops fastest, reflecting that μ is independent from the states and other parameters.

Joint state sampling via filtering and smoothing From previous discussions, we know that the ssGibbs mixes slowly in SSPP. This is due to fact that the states are highly correlated. The ssGibbs tends to explore a narrow high dimensional distribution with inefficient local moves. On designing a better inference strategy, a immediate option is to take the idea of blocking variables together in the state sampling stage.We push this

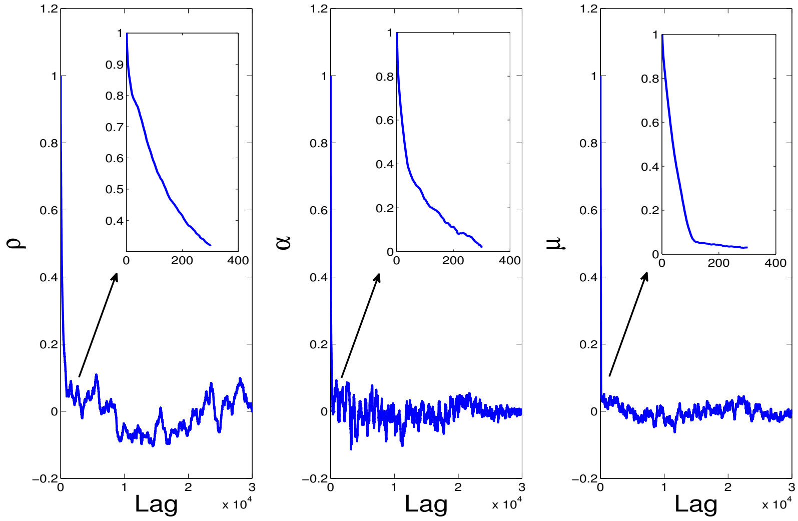


Figure 5.4: he ACF of each sample path obtained by ssGibbs. Zoom-in version for the first 300 lags is also plotted.

idea to its extreme version, that is, drawing $x_{0:k}^{(i)} \sim p(x_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta})$. To this end, it seems natural to use filtering and smoothing in the sampling. In the context of linear dynamical systems, Carter and Kohn (1994) proposed such a sampling approach by factorising the joint state posterior as $p(x_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta}) = p(x_K|\mathbf{y}_{1:K}, \boldsymbol{\theta}) \prod_{k=1}^{K} p(x_k|x_{k+1}, \mathbf{y}_{1:k}, \boldsymbol{\theta})$. In their method, the sampling of the states is split into two stages: first, a Kalman filter is used to obtain the sufficient statistics of the filtering densities $p(x_k|\mathbf{y}_{1:k})$; second, they treated x_{k+1} as an additional observation for x_k. Then, the state transition equation can be used as an observation equation in a Kalman filter recursion. Such a filter estimates the mean and variance of $p(x_k|x_{k+1}, \mathbf{y}_{1:k}, \boldsymbol{\theta})$ from which direct sampling is easy.

Carter and Kohn's method is not exact when applying to the SSPP. The reason is that the Laplace Gaussian filter for SSPP is an approximate filter. Consequently, a Metropolis correction step is necessary at each MCMC iteration. In our experiment, the method gives huge rejection rate ($\approx$ 90%). The reason of such a high rejection rate is intuitive; specifically, Carter and Kohn's method aims to draw sample from $p(x_k|x_{k+1}, y_{1:k}, \boldsymbol{\theta})$. The dependency of x_{k+1} is introduced by using a sample $x_{k+1}^{(i)}$ from $p(x_{k+1}|xk+2, \mathbf{y}_{1:k}, \boldsymbol{\theta})$. In linear Gaussian models, this sampling scheme can be easily implemented by a standard Kalman filter with the state transition equation as an observation equation (x_{k+1} becomes an additional observation of x_k). In SSPP, the Kalman filter has to be replaced by the Laplace Gaussian filter; as a result, the sample $x_{k+1}^{(i)}$ is no longer exactly from $p(x_{k+1}|xk+2, \mathbf{y}_{1:k}, \boldsymbol{\theta})$. One could imagine an error term being accumulated over time. The proposed state sequence is therefore likely to be rejected.

To solve this problem one might have to reduce the number of states that are simultaneously being updated. This approach will reduce the accumulated error by simply reducing the number of accumulation. Strictly speaking, a more proper modification should employ a Metropolis correction step at each time point; then the proposed sample will be exact. This is in contrast with our aiming, which is to update all the states in on go. Due to the fact that we have to correct state samples at each time point, the method is essentially a single-site update scheme with a sophisticated proposal density for states. Therefore, we resort to other methods for joint state updating.

5.2.2 Riemann manifold Hamiltonian Monte Carlo

An alternative class of efficient MCMC methods are gradient based methods, in which the gradient of the underlying distribution is used to assist large moves. A representative of this class is the Hamiltonian Monte Carlo (HMC) method (Duane et al., 1987). HMC employs a Hamiltonian dynamical system as a proposal mechanism, the proposed variables are adjusted by a Metropolis step (see a recent review in Neal (2010)). However, the effective use of HMC requires a high level of tuning, which is not feasible with high dimensional problems. Girolami and Calderhead (2011), by considering the manifold structure of the distribution of interest, propose a novel algorithm: Riemann manifold Hamiltonian Monte Carlo (RMHMC) method, to automatically tune HMC. We first introduce RMHMC on a general problem setting.

Assume we are interested in sampling from a probability density function $p(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^D$, let $\mathcal{L}(\mathbf{x})$ denotes the logarithm of $p(\mathbf{x})$. By introducing an auxiliary variable $\mathbf{p} \in \mathbb{R}^D$ with density $p(\mathbf{p}) = \mathcal{N}(\mathbf{0}, \mathbf{G}(\mathbf{x}))$, we can write down the negative joint log-density of $p(\mathbf{x}, \mathbf{p})$ as

$$H(\mathbf{x}, \mathbf{p}) = -\mathcal{L}(\mathbf{x}) + \frac{1}{2} \log \left((2\pi)^D |\mathbf{G}(\mathbf{x})| \right) + \frac{1}{2} \mathbf{p}^T \mathbf{G}(\mathbf{x})^{-1} \mathbf{p}. \tag{5.26}$$

Following Duane et al. (1987), $H(\mathbf{x}, \mathbf{p})$ can be interpreted as a Hamiltonian in physics, which consists of the sum of a potential energy function $-\mathcal{L}(\mathbf{x})$ at position $\mathbf{x}$, and a kinetic energy function $\frac{1}{2}\mathbf{p}^T \mathbf{G}(\mathbf{x})^{-1}\mathbf{p}$ with momentum variable $\mathbf{p}$ and a mass matrix $\mathbf{G}(\mathbf{x})$. In the traditional HMC paradigm, the mass matrix is a constant, $\mathbf{M}$, which needs to be tuned for good performance – often simply set to the identity matrix. Clearly, when the dimensionality of $\mathbf{x}$ is high, tuning the elements in $\mathbf{M}$ is difficult, and using the identity matrix may lead to poor performance.

In the RMHMC method, the target distribution $p(\mathbf{x})$ is to be defined on a Riemann manifold. The mass matrix $\mathbf{G}(\mathbf{x})$ becomes a metric tensor on the manifold. Assume we have a joint function of data, $\mathbf{z}$, and parameters, $\mathbf{x}$, $p(\mathbf{z}, \mathbf{x})$. The metric tensor is the expected Fisher information matrix:

$$\begin{aligned} \mathbf{G}(\mathbf{x})_{ij} &= -\mathbb{E}_{p(\mathbf{z}|\mathbf{x})} \left[\frac{\partial^2}{\partial x_i \partial x_j} \log p(\mathbf{z}|\mathbf{x}) \right] \\ &= \mathbb{E}_{p(\mathbf{z}|\mathbf{x})} \left[\frac{\partial}{\partial x_i} \log p(\mathbf{z}|\mathbf{x}) \frac{\partial}{\partial x_j} \log p(\mathbf{z}|\mathbf{x}) \right]. \end{aligned} \tag{5.27}$$

To see this clearly, we expand the r.h.s of the first equality. Let D_i denotes the partial derivative $\frac{\partial}{\partial x_i}$ and D_{ij} denotes second derivative $\frac{\partial^2}{\partial x_i \partial x_j}$

$$
\begin{aligned}
\mathbf{G}(\mathbf{x})_{ij} &= -\int p(\mathbf{z}|\mathbf{x}) D_{ij} \log p(\mathbf{z}|\mathbf{x}) d\mathbf{z} \\
&\overset{1}{=} -\int p(\mathbf{z}|\mathbf{x}) D_i \left(\frac{D_j p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} \\
&= -\int p(\mathbf{z}|\mathbf{x}) \left(\frac{D_{ij} p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})} - \frac{D_i p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})} \frac{D_j p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} \\
&\overset{1}{=} -\int D_{ij} p(\mathbf{z}|\mathbf{x}) d\mathbf{z} + \int p(\mathbf{z}|\mathbf{x}) D_i \log p(\mathbf{z}|\mathbf{x}) D_j \log p(\mathbf{z}|\mathbf{x}) d\mathbf{z} \\
&\overset{2}{=} \mathbb{E}_{p(\mathbf{z}|\mathbf{x})} \left[D_i \log p(\mathbf{z}|\mathbf{x}) D_j \log(p(\mathbf{z}|\mathbf{x}) \right],
\end{aligned}
$$

where $\overset{1}{=}$ uses the fact that $D_i \log p(\mathbf{z}|\mathbf{x}) = \frac{D_i p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})}$. $\overset{2}{=}$ exploits the following identity (under some mild conditions):

$$
\int D_{ij} p(\mathbf{z}|\mathbf{x}) d\mathbf{z} = D_{ij} \int p(\mathbf{z}|\mathbf{x}) d\mathbf{z} = D_{ij} 1 = 0.
$$

Using the expected Fisher information matrix as an metric tensor was initially proposed in Rao (1945), and triggered intensive studies on the use of Riemann geometry in statistical inference afterwards (e.g. Amari and Nagaoka (2000); Kass (1989) and etc.).

The Hamiltonian dynamical system based on equation (5.26) is therefore given by

$$
\begin{aligned}
\frac{dx_i}{d\tau} &= \frac{\partial H}{\partial p_i} = \{\mathbf{G}(\mathbf{x})^{-1}\mathbf{p}\}_i \\
\frac{dp_i}{d\tau} = -\frac{\partial H}{\partial x_i} &= \frac{\partial \mathcal{L}}{\partial x_i} - \frac{1}{2}\mathrm{tr}\big(\mathbf{G}(\mathbf{x})^{-1} \frac{\partial \mathbf{G}(\mathbf{x})}{\partial x_i}\big) + \frac{1}{2} \mathbf{G}(\mathbf{x})^{-1} \frac{\partial \mathbf{G}(\mathbf{x})}{\partial x_i} \mathbf{G}(\mathbf{x})^{-1}\mathbf{p}.
\end{aligned}
\tag{5.28}
$$

The system of partial differential equations, equation (5.28), is solved by a generalized leapfrog integrator, such that, the properties of volume preservation and reversibility is maintained:

$$
\mathbf{p}(\tau + \frac{\varepsilon}{2}) = \mathbf{p}(\tau) - \frac{\varepsilon}{2} \nabla_{\mathbf{x}} H\big(\mathbf{x}(\tau), \mathbf{p}(\tau + \frac{\varepsilon}{2})\big) \tag{5.29}
$$

$$
\mathbf{x}(\tau + \varepsilon) = \mathbf{x}(\tau) + \frac{\varepsilon}{2} \Big(\nabla_{\mathbf{p}} H\big(\mathbf{x}(\tau), \mathbf{p}(\tau + \frac{\varepsilon}{2})\big) + \nabla_{\mathbf{p}} H\big(\mathbf{x}(\tau + \varepsilon), \mathbf{p}(\tau + \frac{\varepsilon}{2})\big) \Big) \tag{5.30}
$$

$$
\mathbf{p}(\tau + \varepsilon) = \mathbf{p}(\tau + \frac{\varepsilon}{2}) - \frac{\varepsilon}{2} \nabla_{\mathbf{x}} H\big(\mathbf{x}(\tau + \tau), \mathbf{p}(\tau + \frac{\varepsilon}{2})\big)) \tag{5.31}
$$

These properties of the Hamiltonian system leave the target distribution invariant thereby ensuring a correct MCMC algorithm.

Solutions to equation (5.29)-(5.31), which are obtained by fixed point iterations in practice, yield a trajectory of position variable $\mathbf{x}$ and momentum variable $\mathbf{p}$. Let $\mathbf{x}^*$ and $\mathbf{p}^*$ denote the end of the trajectory, with $\mathbf{x}^*$ becoming the newly proposed variable. Let

$\mathbf{x}^{(i-1)}$ and $\mathbf{p}$ be the starting pair of the trajectory, with $\mathbf{x}^{(i-1)}$, the previous sample. Then, $\mathbf{x}^*$ is accepted or rejected according to the ratio

$$\min\left[1, \exp\left(-H(\mathbf{x}^*, \mathbf{p}^*) + H(\mathbf{x}^{(i-1)}, \mathbf{p})\right)\right].$$

Note that when the metric tensor $\mathbf{G}$ is not a function of the position $\mathbf{x}$, the generalized leapfrog integrator reduces to the standard leapfrog integrator of the HMC method.

$$\mathbf{p}(\tau + \frac{\varepsilon}{2}) = \mathbf{p}(\tau) + \frac{\varepsilon}{2}\frac{\partial \mathcal{L}(\mathbf{x}(\tau))}{\partial \mathbf{x}}, \tag{5.32}$$

$$\mathbf{x}(\tau + \varepsilon) = \mathbf{x}(\tau) + \varepsilon \mathbf{G}^{-1}\mathbf{p}(\tau + \frac{\varepsilon}{2}), \tag{5.33}$$

$$\mathbf{p}(\tau + \varepsilon) = \mathbf{p}(\tau + \frac{\varepsilon}{2}) + \frac{\varepsilon}{2}\frac{\partial \mathcal{L}(\mathbf{x}(\tau + \varepsilon))}{\partial \mathbf{x}}. \tag{5.34}$$

In this scenario, the RMHMC is the same as a HMC with an optimally tuned mass matrix. Example 5.2 belongs to such category

Example 5.4. *Assume that we are trying to sample from the Gaussian distribution in example 5.2. The corresponding gradient is* $-\mathbf{\Sigma}^{-1}(\mathbf{z} - \mathbf{m})$*. The metric tensor in this case is* $\mathbf{\Sigma}^{-1}$*, which is not a function of* $\mathbf{z}$*. For HMC, the mass matrix is set to be the identity matrix.*

Figure 5.5 visualise the moving mechanism of HMC and RMHMC, in which 10 *steps leapfrog integrator is employed for both HMC and RMHC. In particular, the steps size for HMC is* 0.055*. For RMHMC, the step size is* 0.7*. These settings result in approximately* 94% *acceptance rate for both HMC and RMHMC. In addition, for a fair comparison, the random seeds are the same during the course of sampling.*

The advantage of using the metric tensor is evident. With only 100 *samples, the RMHMC is able to provide significantly good coverage over the target distribution. The metric tensor helps the RMHMC to propose very large moves (see e.g. the 2nd and 3rd rows of figure 5.5), while maintaining the accuracy of the integrator, such that, most of the proposed moves are accepted.*

For the application of sampling from the joint posterior $p(x_{0:K}, \boldsymbol{\theta}|y_{1:K})$ of the SSPP model, we adopt the blocked Gibbs sampler paradigm in algorithm 5.3, where RMHMC is applied in states sampling (which jointly updates the whole states sequence), and parameter sampling respectively. The metric tensors in the two sampling stage have two different forms.

Metric tensor for states For sampling the states, the metric tensor of the likelihood is a diagonal matrix in which the entries on the diagonal are $\sum_{c=1}^{C} \beta_c^2 \exp(\mu + \beta_c x_k)\Delta$. The negative Hessian of the log-prior has the same form as stochastic volatility models. Therefore the metric tensor G is a tridiagonl matrix whose diagonal elements are

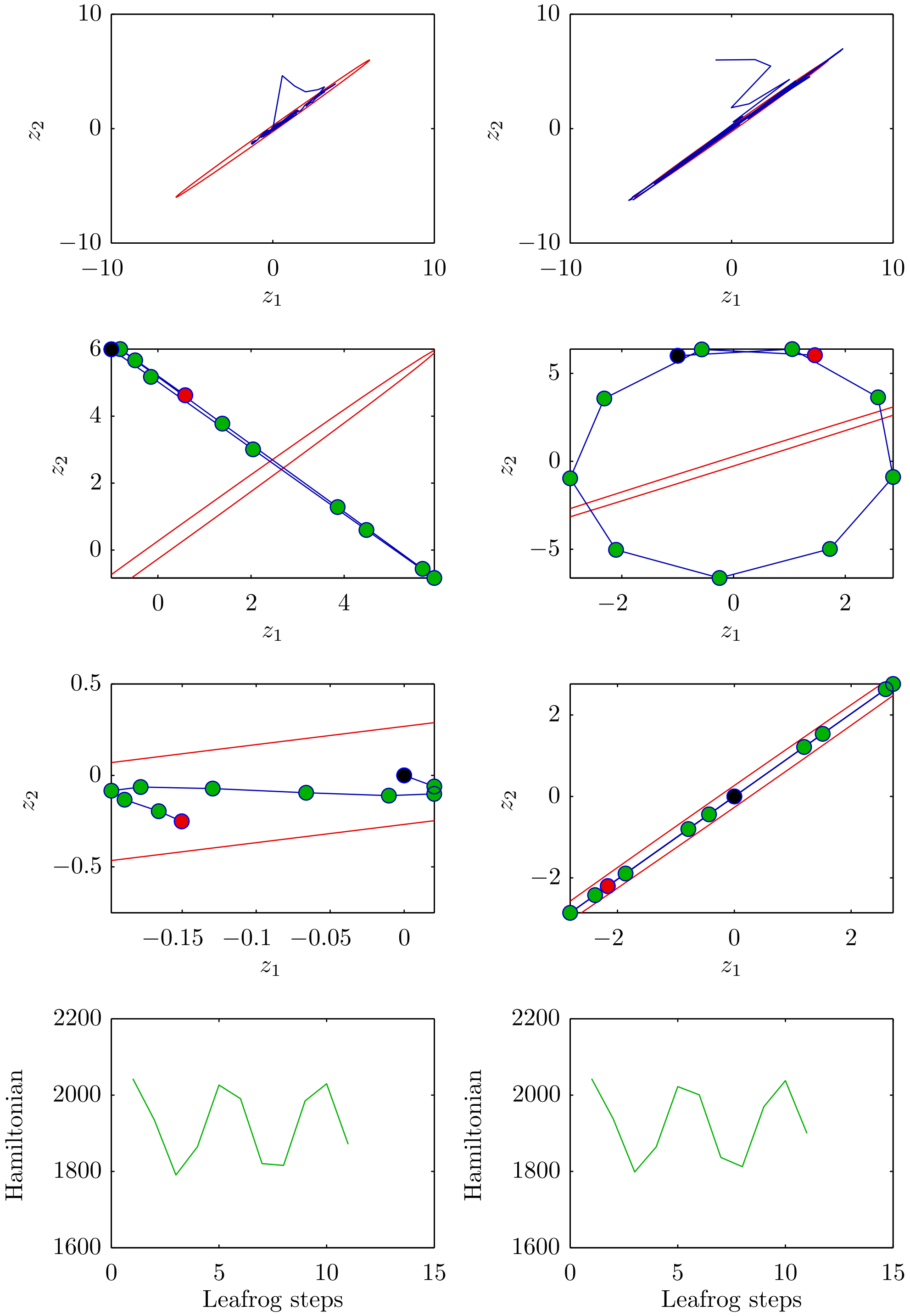


Figure 5.5: Illustration of HMC (*left columns*) and RMHMC (*right columns*), sampling from a 2-dimensional Gaussian distribution described in example 5.2. *The 1st row*: Starting from $[-1, 6]^{\mathrm{T}}$, 100 samples and the 95% confident level of the target distribution as *red ellipse*. *The 2nd row*: The trajectory of the leapfrog integrator in HMC and RMHMC with the starting and ending points marked in *black* and *red*. *The 3rd row*: The trajectory of the leapfrog integrator in HMC and RMHMC with a different initial point. *The 4th row*: The corresponding changes in the Hamiltonian of the leapfrog integrator.

$\big[\frac{1}{\sigma_\varepsilon^2}, \sum_{c=1}^C \beta_c^2 \exp(\mu + \beta_c x_1)\Delta + \frac{1+\rho^2}{\sigma_\varepsilon^2}, \cdots, \sum_{c=1}^C \beta_c^2 \exp(\mu + \beta_c x_{K-1})\Delta + \frac{1+\rho^2}{\sigma_\varepsilon^2}, \sum_{c=1}^C$ $\beta_c^2 \exp(\mu + \beta_c x_K)\Delta + \frac{1}{\sigma_\varepsilon^2}\big]$. Elements on the superdigonal and subdiagonal are $-\frac{1}{\sigma_\varepsilon^2}$.

Further, we integrate out the states, obtaining a constant metric tensor for sampling states. Therefore, the generalized leapfrog algorithm reduces to the standard one in HMC. The formulation of the metric tensor changes accordingly, in particular, the likelihood terms on the diagonal changes to $\sum_{c=1}^C \beta_c^2 \exp(\mu + \beta_c \mathbb{E}[x_k] + \frac{\beta_c^2}{2}\mathrm{Var}[x_k])\Delta$, where $\mathbb{E}[x_k]$ and $\mathrm{Var}[x_k]$ denote the mean and variance of x_k, and are obtained by equation (5.35) and (5.36) respectively.

The benefit of using the averaged metric tensor is merely computational. The computational cost at each MCMC iteration reduces significantly as the generalised Leapfrog integrator become the standard Leapfrog integrator in HMC. Potentially, this procedure might affect the efficiency of the sample. To more specific, the metric tensor is no longer adaptive to the location of the variable. As a result, only the averaged local geometric information is used to assist the proposal mechanism. However, the convergence will not be compromised due to the fact that a Metropolis correction step is applied at the end of each MCMC iteration.

Metric tensor for parameters We only consider three parameters ρ, α and μ, while β_c and σ_ε^2 are fixed to ensure strong identifiability. To constrain the AR process to be stable, ρ is subject to the transformation $\rho = \tanh(\gamma)$. We first obtain the expected value of states $\mathbb{E}[x_k]$ and $\mathrm{Var}[x_k]$:

$$\mathbb{E}[x_k] = \alpha(I_k + \rho I_{k-1} + \cdots + \rho^{k-1} I_1), \tag{5.35}$$

$$\mathrm{Var}[x_k] = \frac{\sigma_\varepsilon^2}{1-\rho^2}. \tag{5.36}$$

Hence, the non-zero terms of the metric tensor (equation 5.27) can be derived as:

$$\begin{aligned}
\mathbb{E}[\frac{\partial^2 \mathcal{L}}{\partial \gamma^2}] &= -2\rho^2 - K(1-\rho^2) - \frac{1-\rho^2}{\sigma_\varepsilon^2}\sum_{k=1}^K \mathbb{E}[x_{k-1}]^2, \\
\mathbb{E}[\frac{\partial^2 \mathcal{L}}{\partial \gamma \alpha}] &= -\frac{1-\rho^2}{\sigma_\varepsilon^2}\sum_{k=1}^K \mathbb{E}[x_{k-1}] I_k, \\
\mathbb{E}[\frac{\partial^2 \mathcal{L}}{\partial \alpha^2}] &= -\sum_{k=1}^K \frac{I_k^2}{\sigma_\varepsilon^2}, \\
\mathbb{E}[\frac{\partial^2 \mathcal{L}}{\partial \mu^2}] &= -\sum_{k=0}^K \sum_{c=1}^C \exp(\mu + \beta_c \mathbb{E}[x_k] + \frac{1}{2}\beta_c^2 \mathrm{Var}[x_k])\Delta.
\end{aligned}$$

The derivatives of the above metric tensor terms w.r.t each parameter, needed in the generalized leapfrog algorithm, are straightforward to carry out. These formulae are provided in appendix C.

With these formulae, we show that, under the parameter settings in example 5.3, the performance of HMC and RMHMC on estimate the hidden states in figure 5.6.

5.2.3 Particle marginal Metropolis-Hastings algorithm

Differing the blocked Gibbs sampler in algorithm 5.3, Andrieu et al. (2010) propose a particle marginal Metropolis-Hastings (PMMH) algorithm, which not only jointly samples states, but also updates parameters simultaneously with the states.

One may use a proposal mechanism joint in states and parameters as below:

$$q(\{\boldsymbol{\theta}^*, x_{0:K}{}^*\}|\{\boldsymbol{\theta}, x_{0:K}\}) = q(\boldsymbol{\theta}^*|\boldsymbol{\theta})p(x_{0:K}{}^*|y_{1:K}, \boldsymbol{\theta}^*).$$

where the superscript $*$ denotes for proposed variables. Such a proposal mechanism requires an efficient sampling approach for the states, so that the proposed $x^*_{0:K}$ is linked to the proposed $\boldsymbol{\theta}^*$ in a "deterministic" fashion. The only remaining degree of freedom is in the parameter proposal process. Thus, the MH acceptance ratio reduces to the following:

$$\frac{p(x^*_{0:K}, \boldsymbol{\theta}^*|y_{1:K})q(\{\boldsymbol{\theta}, x_{0:K}\}|\{\boldsymbol{\theta}^*, x^*_{0:K}\})}{p(x_{0:K}, \boldsymbol{\theta}|y_{1:K})q(\{\boldsymbol{\theta}^*, x^*_{0:K}\}|\{\boldsymbol{\theta}, x_{0:K}\})} = \frac{p(y_{1:K}|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{p(y_{1:K}|\boldsymbol{\theta})p(\boldsymbol{\theta})q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}. \tag{5.37}$$

There are two key issues with this algorithm. Firstly, how to directly draw samples from the smoothing distribution $p(x_{0:K}|y_{1:K}, \boldsymbol{\theta})$; and secondly, how to evaluate the marginal likelihood $p(y_{1:K}|\boldsymbol{\theta})$. For SSPP and many general state-space models, exact computation of the marginal likelihood is not possible and one needs to perform approximations.

The PMMH algorithm, by employing the sequential Monte Carlo (SMC) approach (see e.g. Doucet et al. (2001)), provides an integrated solution to both the above problems. It is straightforward to use SMC for sampling hidden states of general state-space models. Moreover, SMC also estimates the marginal likelihood by importance sampling.

The marginal likelihood $p(y_{1:K}|\boldsymbol{\theta})$ can be decomposed as follows:

$$p(y_{1:K}|\boldsymbol{\theta}) = p(y_1|\boldsymbol{\theta})\prod_{k=2}^{K} p(y_k|y_{1:k-1}, \boldsymbol{\theta}), \tag{5.38}$$

where each component takes the form

$$p(y_k|y_{1:k-1}, \boldsymbol{\theta}) = \int p(y_k|x_k, \boldsymbol{\theta})p(x_k|y_{1:k-1}, \boldsymbol{\theta})dx_k. \tag{5.39}$$

With the SMC algorithm, one can simply add up the unnormalized weights of each particle for time k to obtain an estimate of $p(y_k|y_{1:K}, \boldsymbol{\theta})$. Further, multiplying all components yields an estimate of $p(y_{1:K}|\boldsymbol{\theta})$.

The PMMH algorithm can be described in pseudocode as follows:

Algorithm 5.6. The PMMH algorithm (Andrieu et al., 2010).

```
Input: $\boldsymbol{\theta}^{(0)}$, $x_{0:K}^{(0)}$ and $\hat{p}(y_{1:K}|\boldsymbol{\theta}^{(0)})$
Output: $\boldsymbol{\theta}^{(1:M)}$, $x_{0:K}^{(1:M)}$
for $i = 1$ to $M$ do
    Draw $\boldsymbol{\theta}^*$ from $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(i-1)})$
    Use SMC to sample ${x_{0:K}}^* \sim p(x_{0:K}|y_{1:K}, \boldsymbol{\theta}^*)$ and compute $\hat{p}(y_{1:K}|\boldsymbol{\theta}^*)$
    Compute acceptance ratio
        $a = \frac{\hat{p}(y_{1:K}|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^*)}{\hat{p}(y_{1:K}|\boldsymbol{\theta}^{(i-1)})p(\boldsymbol{\theta}^{(i-1)})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(i-1)})};$
    Draw $u \sim \text{Uniform}[0, 1]$
    if $u < a$ then
        $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^*$, ${x_{0:K}}^{(i)} = {x_{0:K}}^*$ and $\hat{p}(y_{1:K}|\boldsymbol{\theta}^{(i)}) = \hat{p}(y_{1:K}|\boldsymbol{\theta}^*)$
    else
        $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$, ${x_{0:K}}^{(i)} = {x_{0:K}}^{(i-1)}$ and $\hat{p}(y_{1:K}|\boldsymbol{\theta}^{(i)}) = \hat{p}(y_{1:K}|\boldsymbol{\theta}^{(i-1)})$
    end if
end for
```

The PMMH provides a mathematically rigorous sampling approach. Andrieu et al showed that, with any $N \geq 1$, the sampler leaves the target distribution invariant, provided that the resampling scheme being unbiased, the marginal posterior of states $p(x_{1:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta})$ being approachable by a SMC-obtained importance density, also the $q(\cdot|\boldsymbol{\theta}^{(i-1)})$ being irreducible and aperiodic (more details can be found in the section 4 of Andrieu et al. (2010)). This theoretical guarantee shows that PMMH is correct MCMC method, even when the number of particle N is small.

The computational cost of PMMH scales as $\mathcal{O}(NTM)$, where N is the number of the particles used in SMC, and T and M are the total numbers of time points and MCMC iterations, respectively. For neural spike train modelling with SSPP models, the length of time series is often long. Moreover, in order to achieve acceptable performance of SMC, thousands of particles are needed. As a result, computational considerations may be high for the PMMH algorithm. Finally, in figure 5.6, we demonstrate the performance in state sampling stage.

5.3 Approximate methods for SSPP

The previous two methods perform exact MCMC which ensures convergence to the target distribution. However, these methods are computationally expensive in scenarios

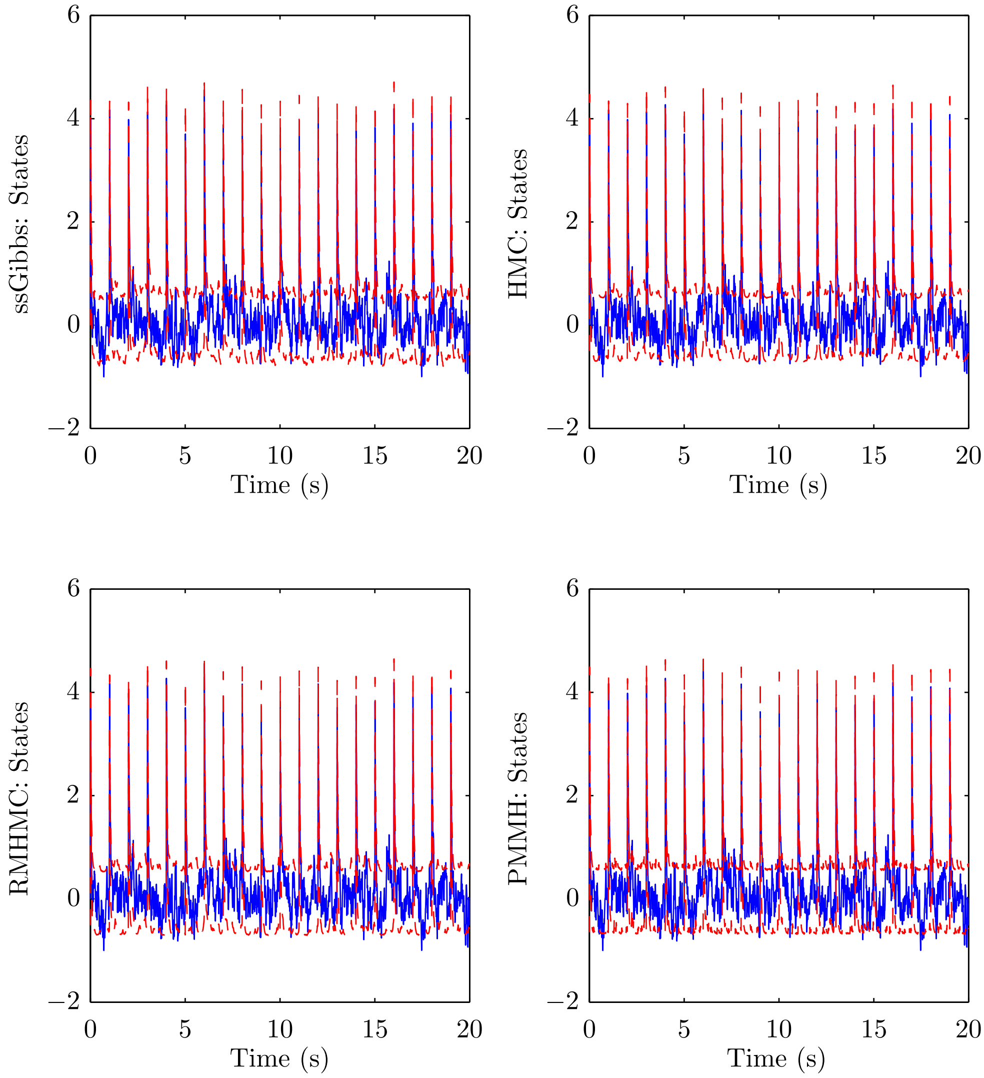


Figure 5.6: 2-standard derivation of the marginal posterior $p(x_k|\mathbf{y}_{1:K}, \boldsymbol{\theta})$ *red dashed lines*, and the true states (*blue lines*).

of long data records. An additional issue for PMMH has to do with degeneracy of SMC which can also be acute when processing large datasets. Further, RMHMC is not ideally suited for data in which the input stimuli are continuous. The reason is that, the generalised leapfrog algorithm, which involves numerical iterations, requires equation (5.35) to be updated many times for each state. For periodic sparse input signals the computation can be implemented efficiently, whereas when the inputs are dense or continuous, updating will be computationally expensive.

To provide workarounds for SSPP in the above situations, in this section, we introduce two MCMC methods, which employ approximation schemes in ways different from approximate Bayesian methods such as VB. The key idea in these is to use Laplace approximation to the posterior over states, while treating the inference relating to parameters by sampling. Such an approach allows rather relaxed approximations, whereas VB methods are restricted to the conjugate exponential family. Laplace approximation, being a second order method, has its limitations and needs to be deployed with caution when good accuracies are required.

5.3.1 Rao-Blackwellised Gibbs sampler

The first method we introduce, is within the framework of Rao-Blackwellization of the Gibbs sampler with respect to the states, replacing their dependencies by sufficient statistics. A discussion of the Rao-Blackwellization approach in a general MCMC setting is given in Casella and Robert (1996). For application in state-space models, such an idea has been proved to be successful on improving particle filters by Doucet et al. (2000a). Dewar et al. (2010) also have a scheme which has conceptual similarities.

Ideally, we seek to perform marginalization on the full conditional of $\boldsymbol{\theta}$, such that:

$$p(\boldsymbol{\theta}|y_{1:K}) = \int p(\boldsymbol{\theta}|x_{0:K}, y_{1:K})p(x_{0:K}|y_{1:K})dx_{0:K}. \tag{5.40}$$

The left hand side of equation (5.40) is proportional to $p(y_{1:K}|\boldsymbol{\theta})p(\boldsymbol{\theta})$, where

$$p(y_{1:K}|\boldsymbol{\theta}) = \int p(x_{0:K}, y_{1:K}|\boldsymbol{\theta})dx_{0:K}. \tag{5.41}$$

We ignore the prior $p(\boldsymbol{\theta})$ for the moment, since it does not depend on the states. Assuming q is an arbitrary function over $x_{0:K}$, we can obtain a variational lower bond $\mathcal{F}(q, \boldsymbol{\theta})$ for the integral in equation (5.41) in the log domain (See Neal and Hinton (1998); Roweis and Ghahramani (1999) for the same analysis for EM).

$$\begin{aligned}\mathcal{F}(q, \boldsymbol{\theta}) &= \int q(x_{0:K}) \ln \frac{p(x_{0:K}, y_{1:K}|\boldsymbol{\theta})}{q(x_{0:K})} dx_{0:K} &(5.42)\\ &= \int q(x_{0:K}) \ln p(x_{0:K}, y_{1:K}|\boldsymbol{\theta})dx_{0:K} - \int q(x_{0:K}) \ln q(x_{0:K})dx_{0:K}. &(5.43)\end{aligned}$$

Following Neal and Hinton (1998), the lower bound is maximised and is equivalent to $\log p(y_{1:K}|\boldsymbol{\theta})$ when

$$q(x_{0:K}) = p(x_{0:K}|y_{1:K}, \boldsymbol{\theta}). \tag{5.44}$$

Therefore, sampling parameter from $p(\boldsymbol{\theta}|y_{1:K})$, when condition equation (5.44) holds, is equivalent to sampling from the exponential of the lower bound with prior $\exp(\mathcal{F}(q, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}))$. Due to the fact that the second term in equation (5.43) does not depend on $\boldsymbol{\theta}$ when $q(x_{0:K})$ is fixed, the target distribution is further equivalent to replacing the lower bound with the first term in equation (5.43), which is conventionally denoted as $\mathcal{Q}(q, \boldsymbol{\theta})$. We use $\mathcal{P}(\boldsymbol{\theta})$ to denote the above target distribution, which has the following expression:

$$\mathcal{P}(\boldsymbol{\theta}) = \exp(\mathcal{Q}(q, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})).$$

Hence, in the parameter sampling stage, RBG draws samples from the function $\mathcal{P}(\boldsymbol{\theta})$.

However, for most general state-space models, including the SSPP model, it is not possible to find a $q(x_{0:K})$ that satisfies equation (5.44), which we tackle here by introducing approximations. Therefore, the RBG for SSPP samples form an approximated distribution, denotes as $\hat{\mathcal{P}}(\boldsymbol{\theta})$.

For models where $p(x_{0:K}|y_{1:K}, \boldsymbol{\theta})$ is unimodal, one can approximate it by a Gaussian centered at its mode. As the SSPP model falls into such a category, we will use this Gaussian approximation, which has also been used in the EM algorithm of the original paper that introduced the SSPP model (Smith and Brown, 2003).

For this, $q(x_{0:K})$ is computed by a forward Laplace Gaussian filter (LGF), referred to as nonlinear recursive filter in Smith and Brown (2003), and a backward Kalman smoother (RTS smoother). LGF is a recent term coined by Koyama et al. (2010), who gave a rigorous treatment, while the formulation itself has wide application in multiple neural spike trains and heartbeat modelling (Eden et al., 2004; Ergün et al., 2007; Wang et al., 2009).

Let $x_{k|K} = \mathbb{E}_{p(x_k|y_{1:K})}[x_k]$, $\sigma^2_{k|K} = \mathrm{Var}_{p(x_k|y_{1:K})}[x_k]$ $W_k = \mathbb{E}_{p(x_k|y_{1:K})}[x_k^2]$ and $W_{k,k-1} = \mathbb{E}_{p(x_k|y_{1:K})}[x_k x_{k-1}]$ be the sufficient statistics that are needed for computing $\hat{\mathcal{P}}(\boldsymbol{\theta})$ and S denote the ensemble of these quantities. Using these notations, our RBG algorithm is given by the pseudo code below:

Algorithm 5.7. Rao-Blackwellised Gibbs sampler for SSPP model.

Input: $\boldsymbol{\theta}^{(0)}$, $x_{0:K}^{(0)}$ and S

Output: $\boldsymbol{\theta}^{(1:M)}$ and $x_{0:K}^{(1:M)}$

```
for $i = 1$ to $M$ do
    Use LGF and RTS smoother to obtain $\{x_{k|K}\}_{k=0}^K$, $\{\sigma^2_{k|K}\}_{k=0}^K$, $\{W_k\}_{k=0}^K$ and
    $\{W_{k,k-1}\}_{k=1}^K$
    Let $S^{(i)} = \Big(\{W_k\}_{k=0}^K, \{W_{k,k-1}\}_{k=1}^K, \{x_{k|K}\}_{k=0}^K, \{\sigma^2_{k|K}\}_{k=0}^K\Big)$
    for $k = 0$ to $K$ do
        Draw $x_k^{(i)} \sim \mathcal{N}(x_{k|K}, \sigma^2_{k|K})$
        Let $x_{0:K}^{(i)} = \{x_k^{(i)}\}_{k=0}^K$
        Draw $\boldsymbol{\theta}^{(i)}$ from the $\hat{\P}(\boldsymbol{\theta})$
    end for
end for
```

The RBG algorithm is conceptually closely related the collapsed Gibbs sampler due to Liu (1994), in the sense of using marginalization to improve the sampler efficiency. By doing this, in the application to SSPP or general state-space model, it leads to a combination of variational and MCMC methods, drawing benefits from both sides. Another related work is that of Rosti and Gales (2004), who propose a method for switching linear dynamical systems using the same terminology. The purpose of their marginalisation is improved state sampling, with the parameters treated within a maximum likelihood estimator. Ours, however, is a full Bayesian approach of a more sophisticated model. The RBG algorithm follows the same idea of the sampler designed in Campbell and Godsill (1998). The difference is that our method is developed for a non-Gaussian model.

5.3.2 Laplace marginal Metropolis-Hastings algorithm

The second approximate method we introduce, is a variant of the PMMH algorithm, which we will refer to as Laplace marginal Metropolis-Hastings (LMMH) algorithm. It follows the same paradigm as PMMH, using a Laplace approximation to the marginal likelihood $p(\mathbf{y}_{1:K}|\boldsymbol{\theta})$ and obtaining sufficient statistics of filtering densities via LGF.

Let $x_{k|k}$ and $\sigma^2_{k|k}$ be the mode and variance of $p(x_k|y_{1:k}, \boldsymbol{\theta})$; $x_{k|k-1}$ and $\sigma^2_{k|k-1}$ be mode and variance of the one-step prediction $p(x_k|y_{1:k-1})$. The Laplace approximation of equation (5.39) is:

$$\hat{p}(y_k|y_{1:k-1}, \boldsymbol{\theta}) = \mathcal{N}(x_{k|k}|x_{k|k-1}, \sigma^2_{k|k-1})p(y_k|x_{k|k}, \boldsymbol{\theta})\sqrt{2\pi\sigma^2_{k|k}},$$

where $p(y_k|x_{k|k}, \boldsymbol{\theta}) = \prod_{c=1}^{C} p(y_k^c|x_{k|k}, \mu, \beta_c)$. Applying the same factorization in equation (5.38), we obtain an estimate of the marginal likelihood $\hat{p}(y_{1:K}|\boldsymbol{\theta})$. LMMH, in pseudocode form, is given below:

Algorithm 5.8. LMMH for SSPP model.

```
Input: $\boldsymbol{\theta}^{(0)}$, ${x_{0:K}}^{(0)}$ and $\hat{p}(y_{1:K}|\boldsymbol{\theta}^{(0)})$
Output: $\boldsymbol{\theta}^{(1:M)}$ and $x_{0:K}^{(1:M)}$
for $i = 1$ to $M$ do
    Draw $\boldsymbol{\theta}^*$ from $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(i-1)})$
    Use LGF and RTS smoother to compute $\{x_{k|k}\}_{k=0}^K$, $\{\sigma^2_{k|k}\}_{k=0}^K$, $\{x_{k|K}\}_{k=0}^K$ and $\{\sigma^2_{k|K}\}_{k=0}^K$
    for $k = 0$ to $K$ do
        Draw $x_k^* \sim \mathcal{N}(x_{k|K}, \sigma^2_{k|K})$
    end for
    Let $x_{0:K}^* = \{x_k^*\}_{k=0}^K$
    Compute $\hat{p}(y_{1:K}|\boldsymbol{\theta}^*)$ by Laplace's method
    Compute acceptance ratio
```

$$a = \frac{\hat{p}(y_{1:K}|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^*)}{\hat{p}(y_{1:K}|\boldsymbol{\theta}^{(i-1)})p(\boldsymbol{\theta}^{(i-1)})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(i-1)})}; \tag{5.45}$$

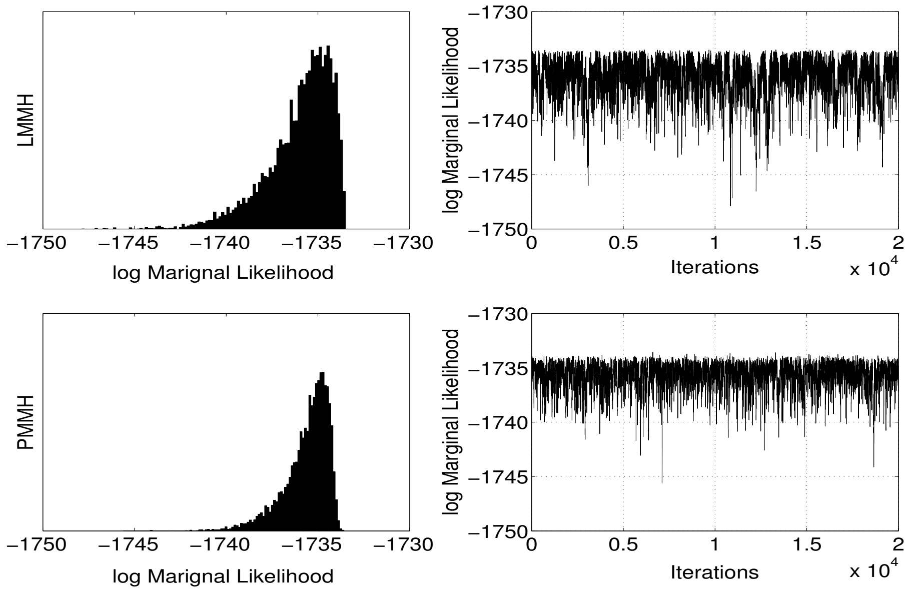


Figure 5.7: Marginal likelihood estimates from LMMH (top row) and PMMH (bottom row), where the right column shows the trace, and the left column corresponds to its histogram.

```
    Draw $u \sim \mathcal{U}(0,1)$
    **if** $u < a$ **then**
        $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^*$, $x_{0:K}^{(i)} = x_{0:K}^*$ and $\hat{p}(y_{1:K}|\boldsymbol{\theta}^{(i)}) = \hat{p}(y_{1:K}|\boldsymbol{\theta}^*)$
    **else**
        $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$, $x_{0:K}^{(i)} = x_{0:K}^{(i-1)}$ and $\hat{p}(y_{1:K}|\boldsymbol{\theta}^{(i)}) = \hat{p}(y_{1:K}|\boldsymbol{\theta}^{(i-1)})$
    **end if**
**end for**
```

In this algorithm, the Laplace approximation to the marginal likelihood is carried out at each point in time, replacing the particle based estimation of PMMH. This achieves a drastic reduction in computational cost. The effect of the approximation is that the acceptance ratio equation (5.45) will be different from the true ratio, equation (5.37), which the PMMH can asymptotically achieve with infinite number of particles. Moreover, the Laplace's method requires MAP estimation of the hidden x_k, which would be inappropriate for multimodal distributions, which, as discussed earlier, is not an issue for SSPP.

In figure 5.7, we compare Laplace approximation (LMMH) and importance sampling (PMMH) to the marginal likelihood of the SSPP model. The outcome of the two methods are in good agreement. Despite the fact that the Laplace approximation gives some lower estimates than PMMH, the results across 20,000 draws are consistently in good agreement between the two methods.

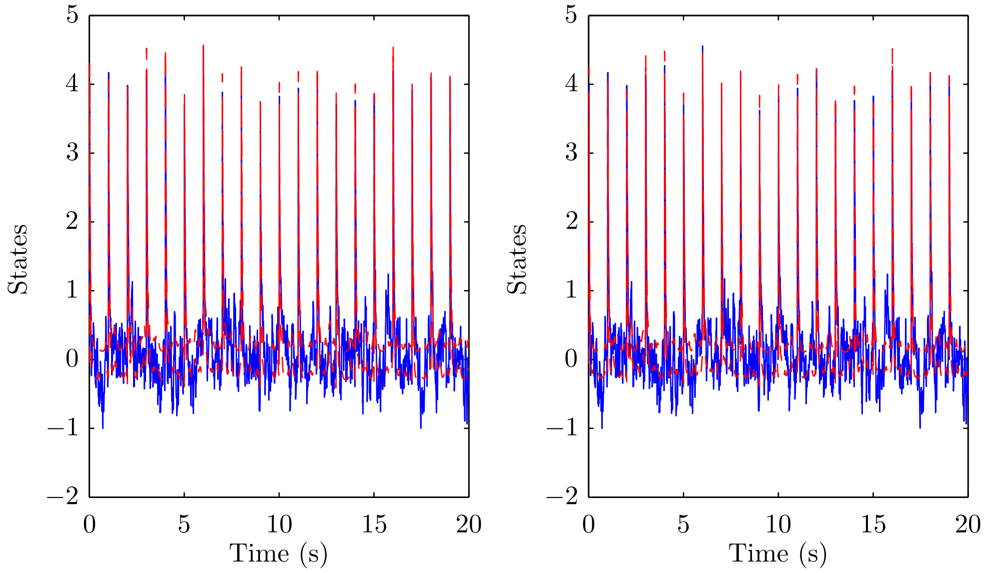


Figure 5.8: 2-standard derivation of the marginal posterior $p(x_k|\mathbf{y}_{1:K}, \boldsymbol{\theta})$ *red dashed lines*, and the true states (*blue lines*). Ordering from *left* to *right*, represent samples obtained by RBG and LMMH.

Finally, in figure 5.8, the state estimation performance of RBG and LMMH are demonstrated. In both cases, the obtained state posteriors are more compact as compare to the exact MCMC methods.

5.4 Quantitative efficiency comparison

In this section, we examine, under the SSPP model, the efficiency of the five MCMC and the two approximate MCMC methods. All simulations were carried out with MATLAB®on an Intel®Core$^{\mathrm{TM}}$2 Quad Q6600 @ 2.40GHZ with 4GB RAM computer.

The parameter settings used for generating the synthetic dataset are the same in example. As mentioned previously, to ensure strong identifiability, we fixed β_c and σ_ε^2 to their true values. Hence the inference task is focused on the states and parameters ρ, α and μ. In addition, each of the three parameters is assigned a flat prior.

The implementation details of the seven methods are the following.

- *ssGibbs* uses the state transition density proposal for each state and random walk proposals for the parameters, in particular, $\mathcal{N}(\theta^{(i-1)}, 0.01^2)$ for ρ and $\mathcal{N}(\theta^{(i-1)}, 0.1^2)$ for both α and μ.
- *PMMH* uses the same proposals for parameters as the single-site Gibbs sampler. The particles of SMC algorithm are proposed by the state transition density with a population of $1,000$.

Table 5.1: Acceptance rates of all five methods.

ssGibbs	HMC	RMHMC
$30\% - 60\%$	$80\% - 90\%$	$85\% - 99\%$
PMMH	RBG	LMMH
$40\% - 55\%$	$35\% - 65\%$	$40\% - 55\%$

- *HMC* uses an identity mass matrix for which is further scaled by step sizes. Specifically, in the state sampling stage, we employ 34 integration steps with a step size of 0.03 . For the parameters, 67 integration steps with each step size of 0.015 are chosen. On top of the above settings, we also use random integration directions to ensure reversibility.

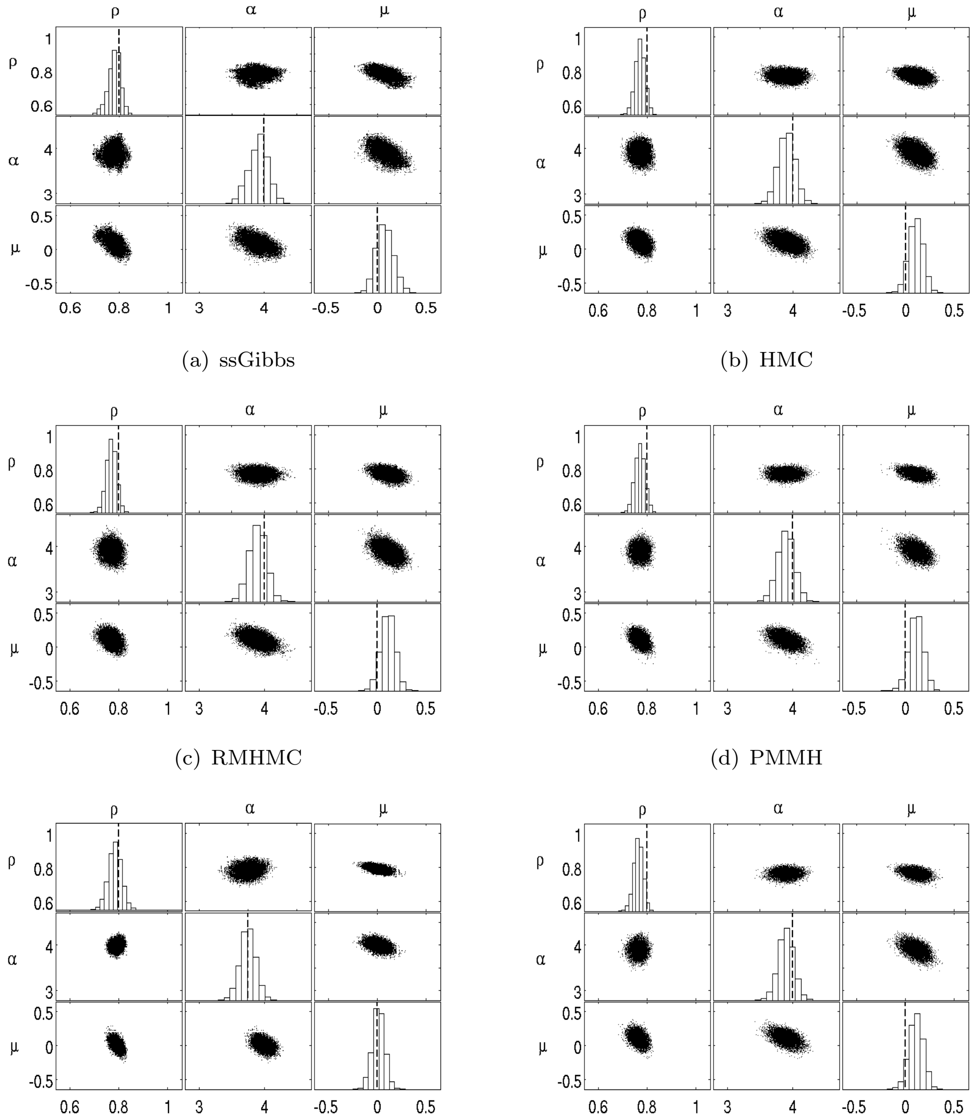
- *RMHMC* uses a step size of 0.2 and 25 integration steps for the states, and a step size of 0.8 and 5 integration steps for parameters. Again, a random integration direction is applied at each generalized leapfrog loop.

- *RBG* and *LMMH* both use the same proposals for parameters as the single-site Gibbs sampler.

HMC is tuned in the light of making a trade-off between acceptance rate and number of the leapfrog steps within each Monte Carlo iteration. In other words, we aim to integrate over a certain distance with small number of integration steps, without rejecting too many proposals. Likewise, we tuned RMHMC in the same spirit. In addition, simulations show that, benefiting from the use of local geometric structure, with the same number of integration steps, RMHMC is able to make much larger moves while maintaining high acceptance rate, consistent with the findings in Girolami and Calderhead (2011). Based on this, one can achieve fast mixing with less integration steps. With the above settings, as expected, the acceptance rates of HMC and RMHMC shown in Table 5.1 are much higher than the other two random walk proposal based methods.

Figure 5.9 shows the posterior distributions obtained by each of the seven MCMC methods. While PMMH, HMC and RMHMC faithfully capture the posterior distributions, the single-site Gibbs sampler produces some ad hoc shapes within the clouds of samples, implying the chosen burn-in period is not sufficiently long.

It is evident that the posteriors of both two approximate methods are in good agreement with those produced by exact methods (ssGibbs, PMMH, HMC and RMHMC). In particular, LMMH shows almost identical results as the exact methods. A small shift on the RBG posterior marginals is observed, however such shift leads to a more accurate estimation in terms of the posterior mean, and as expected, due to the Rao-Blackwellisation, the uncertainty in RBG is notably smaller than those obtained by the other methods.



(a) ssGibbs

(b) HMC

(c) RMHMC

(d) PMMH

(e) RBG

(f) LMMH

Figure 5.9: Full posterior distribution of parameters obtained by four methods, where true value of each parameter is indicated by dashed line. There are $20,000$ (after $1,000$ burn-in) posterior samples for each of the three parameters.

In addition to the posterior profile, by comparing the $\sqrt{\hat{R}}$ statistic from Gelman and Rubin (1992), we further assess each method on the time for convergence to the stationary distribution in Fig.5.10. This test is carried out by considering 5 chains with different initializations. Since we have a total of 2004 variables, we only show the statistics for parameters which capture the overall convergence status well. We observe that the Markov chain obtain by single-site Gibbs sampler is poorly mixed in ρ. Whereas, RMHMC consistently shows the fastest convergence performance.

Table 5.2 shows relative performances of the MCMC methods in terms of effective sample size (ESS) and processing time. Such criteria (ESS and processing time) were also used

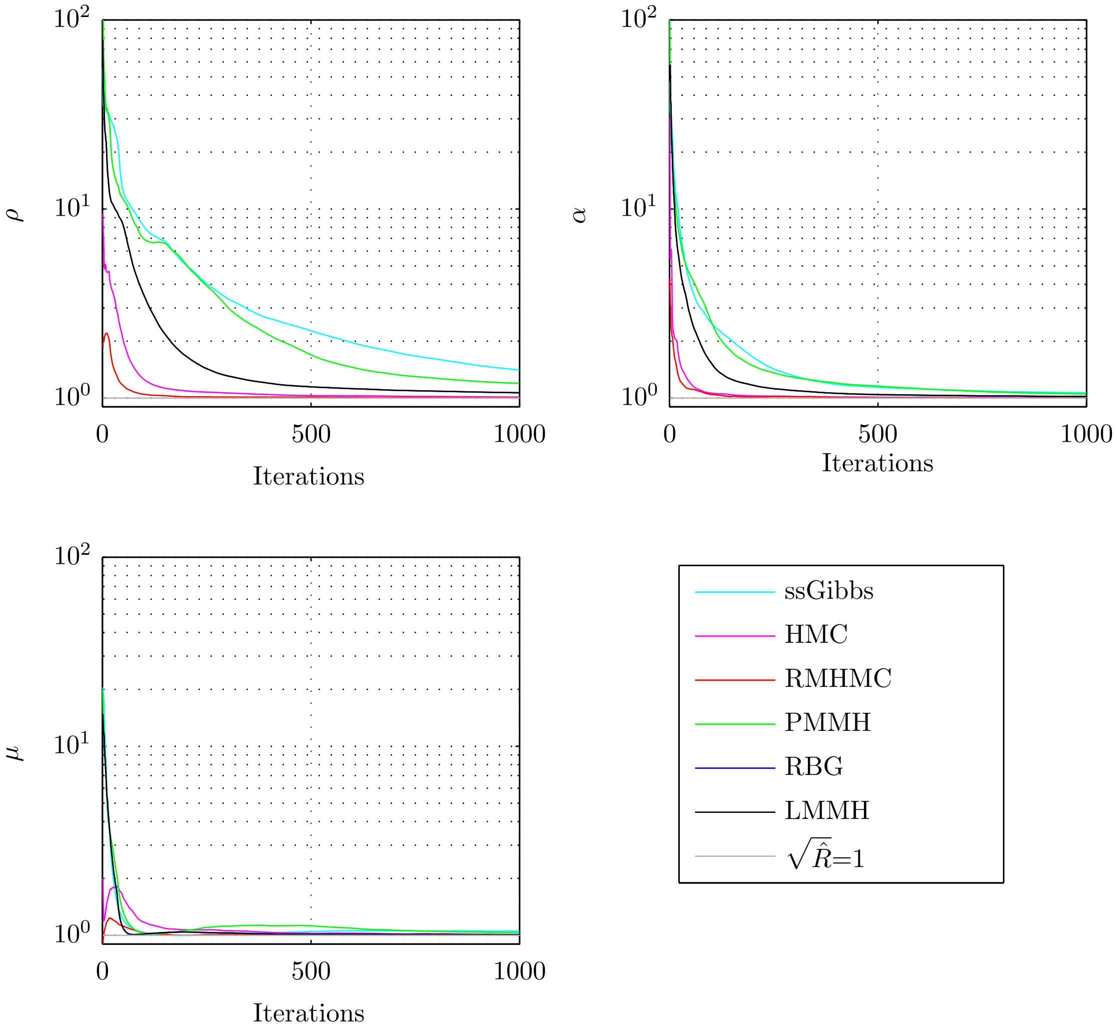


Figure 5.10: Logarithm $\sqrt{\hat{R}}$ statistics (see e.g. (Gelman and Rubin, 1992)) for ρ, α and μ. Convergence corresponds to an $\sqrt{\hat{R}}$ value close to 1.

in Girolami and Calderhead (2011) (see Liu (2001) for more details on ESS). In order to make a fair assessment, each method is run 10 times on the same dataset and averages tabulated. We note that RMHHC shows the highest ESS scores for both states and parameters, and ranks second in processing speed. HMC shows the second highest ESS score on states, yet the parameter ESS (in ρ and μ) are similar to PMMH. Further, all methods show significant improvement on ESS when compared to the baseline single-site Gibbs sampler. Finally, results on autocorrelation function (ACF) performance in Fig.5.11 also lend additional supports to the above findings.

The processing time can be unreliable, instead, we give estimates of the orders of computational complexities of the different sampling algorithms used. Let K and M be the time points and the number of Monte Carlo iterations, respectively. N denotes the number of particles used in PMMH. L_1 and L_2 are the number of leapfrog steps for states and parameters in HMC. Superscript $*$ indicates that the number of leapfrog steps in RMHMC are different from those in HMC. In addition, we assume the cost

Table 5.2: ESS and processing time comparison based on 20, 000 posterior samples (1000 burn-in) of states and parameters obtained by single-site Gibbs, PMMH, HMC and RMHMC on synthetic dataset. Each attribute is averaged over 10 runs.

Methods	ESS (ρ,α,μ)	States ESS (Min, Median, Max)	Time(s)
ssGibbs	$16, 94, 38$	$46, 98, 210$	2594
PMMH	$458, 939, 1055$	$567, 2132, 5129$	115341
HMC	$340, 1590, 930$	$1152, 4045, 10979$	4225
RMHMC	$1072, 1593, 2326$	$4060, 20000, 20000$	3136
RBG	$392, 613, 670$	$532, 18018, 19648$	2347
LMMH	$482, 1027, 1113$	$603, 4830, 5172$	2661

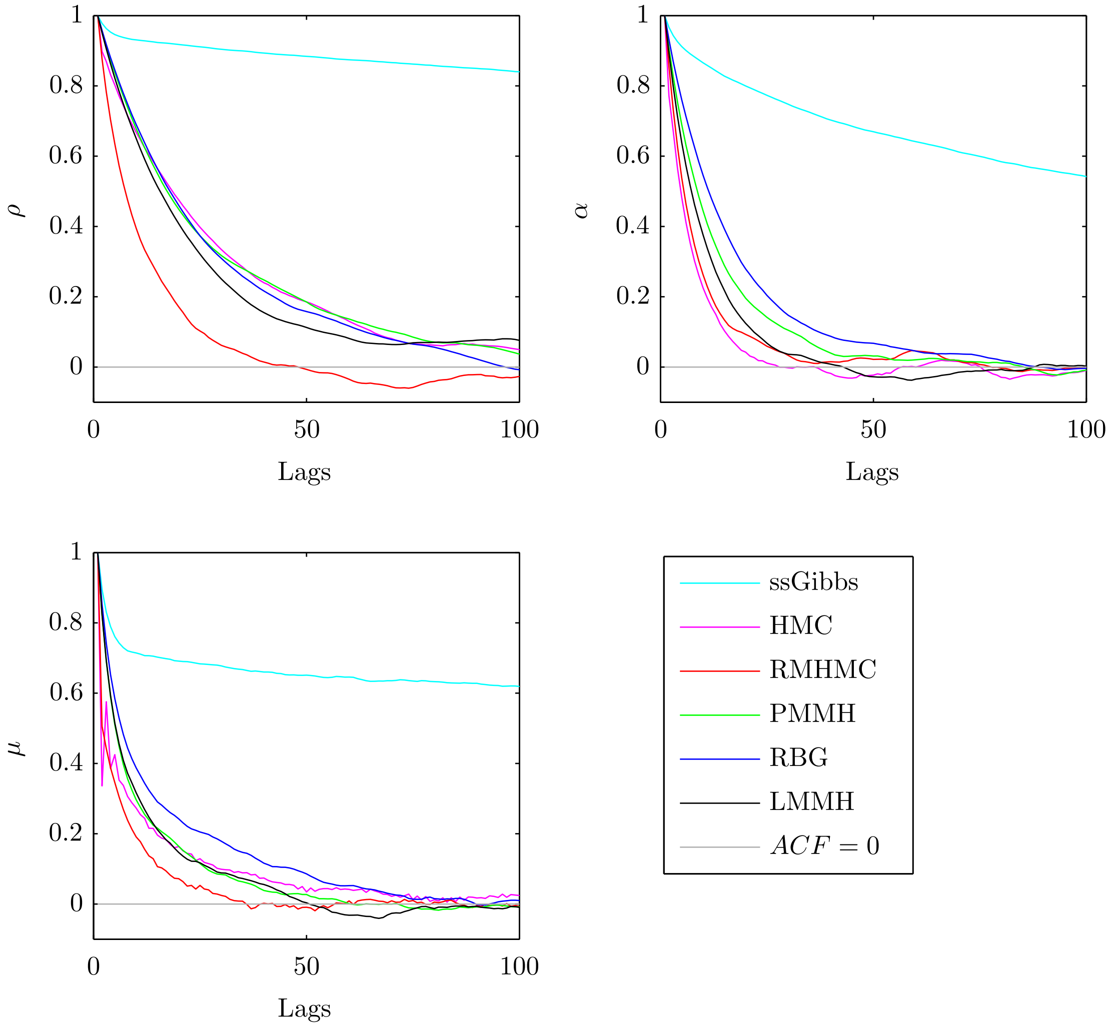


Figure 5.11: The first 100 lags of autocorrelation values of different MCMC methods for each parameter. RMHMC outperforms other methods in ρ and μ, whereas in α HMC drops faster than others, indicating that a unit tensor in α may be appropriate.

of each parameter updating and other inner calculations to be 1. Let F and S denote the cost of filtering and smoothing at each time point. With these notations, Table 5.3

Table 5.3: General computational cost comparison.

ssGibbs	HMC	RMHMC
$\mathcal{O}((K+1)M)$	$\mathcal{O}((L_1K+L_2)M)$	$\mathcal{O}((L_1^*K+L_2^*)M)$
PMMH	RBG	LMMH
$\mathcal{O}(NKM)$	$\mathcal{O}((K(F+S)+1)M)$	$\mathcal{O}((K*F+1)M)$

shows the estimated complexities.

In summary, from the comparisons carried out on a synthetic dataset, we conclude that RMHMC is a clear winner in terms of its sampling and computational efficiencies. The performance of PMMH can be further improved by increasing the number of particles used in the SMC stage, or adding sophisticated tricks like auxiliary variables (Pitt and Shephard, 1999) and resample-move algorithm (Gilks and Berzuini, 2001). The computational costs of PMMH, however, higher by a factor of three in the current setting, make PMMH strand less appealing. Such costs could be even higher with real applications, in which the length of data records may be substantial in comparison to the synthetic data we have used.

5.5 Discussion

In this chapter, we study Bayesian inference and learning in a state-space model with point process observations (SSPP) with a wide range of state-of-the art MCMC methods. While all methods we considered converge and produce the correct inference, their efficiencies differ significantly, with RMHMC outperforming the others. The reason is that, by using the gradient of the posterior and benefiting from the volume preservation properties of the Hamiltonian dynamic system, RMHMC is able to propose large moves while maintaining high acceptance rate. These moves are guided by a metric tensor which takes advantage of the underlying manifold structure of the posterior distributions. As for the state-space models and SSPP in particular, the metric tensor takes the form of expected Fisher information matrix which is analytically available. Moreover, due to the previously noted unimodality property of the SSPP model, such a metric tensor is guaranteed to be positive definite for both states and parameters, which justifies the suitability of using RMHMC.

We further presented two approximate methods to improve the efficiencies of a single-site update Gibbs sampler and demonstrated that efficient convergence can be achieved while maintaining posterior distributions similar to the exact methods. Further improvements to the approximate methods considered may be possible by the use of second order Laplace approximation as suggested by Koyama et al. (2010), who give rigorous proofs of approximation bounds in both filtering and smoothing contexts. In addition, as previously noted, the low acceptance rate problem due to the random walk proposal in those methods (also in PMMH), can be easily solved by considering a HMC or RMHMC targeting on the marginal likelihood (PMMH and LMMH) and its variational lower bound (RBG).

For PMMH, a feasible extension would be using the Metropolis adjusted Langevin algorithm (MALA) or manifold MALA method (Girolami and Calderhead, 2011), as the proposal mechanism. Despite the severe computational overheads, this idea is algorithmically attractive, since they offer highly efficient proposal mechanisms to tackle problems with high correlations between states and parameters.

Relating the framework descried in this chapter, another popular model for characterizing neural spike trains is the point process generalized linear model (Truccolo et al., 2005; Okatan et al., 2005), in which the stimuli are treated as canonical parameters in a likelihood model which is similar to the one considered in SSPP. For inference on this model, Paninski (2004) provides a maximum likelihood formulation, whereas for Bayesian perspective, both MCMC and VB approaches have been recently studied and shown good results on decoding the spike trains (Ahmadian et al., 2011; Chen et al., 2010). The SSPP differs from such a paradigm by assuming an latent dynamic process which cooperates the stimuli as input sources, resulting in a physiologically plausible parameterization. Its inference frameworks including EM (Smith and Brown, 2003), VB (Zammit Mangion et al., 2011b) and MCMC discussed in this work also show good performance on decoding the spike train, while the inferred parameters may serve as discriminant attributes of different physiological states.

Chapter 6

Applications

This chapter demonstrates several applications in neural spikes and heartbeats modelling. For neural spikes, we tested a dataset of taste response in rat. The inferred parameter posteriors successfully separates four chemical stimuli. Similar results are also obtained on a dataset of visual response in monkey. Further, these two datasets are used for comparing posteriors obtained by VB and MCMC. The results suggest that the performances are generally similar. We later use a PPGLM for heartbeat analysis. This framework is tested on both synthetic and real heartbeat datasets, showing promising improvement, when comparing with time-frequency features which are dominate in clinical practice.

6.1 Neural data analysis

In this section, we applied the Bayesian SSPP framework to two neural spike datasets: Taste response in rat and visual response in monkey. These two datasets are from neurodatabase.org - a neuroinformatics resource funded by the Human Brain Project.

6.1.1 Modelling taste response in rat

First, we modelled spiking patterns of taste-response cells in the nucleus tractus solitarii (NTS) of Sprague-Dawley rats following the application of different taste stimuli (Di Lorenzo and Victor, 2003). The experimental data was obtained from trials where different compounds dissolved in distilled water were delivered to the oropharyngeal area. On the neurodatabase.org, data from three cells (Cell 4, 9 and 11) are available from this experiment.

The attraction of the SSPP is that it offers a framework for capturing the neural dynamics, while discriminating various chemical stimuli with the estimated parameters.

Although the SSPP was primarily developed for implicit stimuli in Smith and Brown (2003), it provides a neat way of parameterising a dynamic CIF to model variable rate neural responses to explicit stimuli. Such as the case considered here, where ample evidence suggests that for some of the cells in the NTS, rate coding is used for interstimulus discrimination (Roussin et al., 2008).[1]

Some of these are so fine-tuned to different stimuli that one can use spike count alone to discriminate between different tastes (e.g. cell 9 in the study). Others, on the other hand, are not so fine-tuned and spike count cannot be used to discriminate between the tastants (e.g. cell 11). Nonetheless, spike count gives no information on the time-varying event rate (or rate envelope) itself. Moreover, many alternatives (such as the conventional sliding window) do not provide a plausible model for the underlying neural dynamics. The SSPP applied to these cells can not only give the descriptive powers required for taste discrimination, but also additional information which may be of physiological use. Here we also show how the VB filter can infer the varying SSPP parameters governing the underlying dynamics, which for the same neuron appear to vary in a structured manner with the application of different stimuli.

Data pre-processing Each experimental trial consisted of three phases: i) a 10s baseline period in the absence of any stimulus, ii) 5s of stimulus presentation, and iii) a 5s wait. Each trial was separated by rinsing and a 1.5min wait. The data used in the analysis was that recorded in the second and third phases (10s segments), in which the neural response to the four tastants used, NaCl, HCl, quinine and sucrose, (each of which represents a different taste quality; salty, sour, bitter and sweet respectively), is present. The learning data set was formed by first grouping the 10s segments according to stimulus, and then concatenating them into four sets (1 per stimulus). Combinations of these spike trains were then joined together to form the data sets on which learning was carried out.

Data was gathered at a resolution of 1ms and we hence initially organised the spikes into bins of $\Delta = 1$ms. However we then increased the bin size to $\Delta = 10$ms to speed up the algorithm. This resulted in some bins ($< 5\%$) containing more than one output spike (e.g. for cell 9 - max. HCl with 3.4% and min. sucrose with 1.5%) which were subsequently repositioned to the closest empty bin in forward time. Pre-analysis of the data was carried out by studying the post-stimulus histograms (PSTHs) of the responses to the four stimuli. These histograms suggested an approximate linear increase in firing rate for the first 250ms, and also a response latency which was not considered in the simulation study. To cater for these effects, we treated the input signal as a pulse of width 250ms. The resulting data contained $23,000$ time points in Cell 9, and 16000 time points in Cells 4 and 11.

[1] as opposed to temporal coding where the specific arrangement in time is of particular relevance to discrimination and deemed to play an important role, particularly in the initial (phasic) phase of the response.

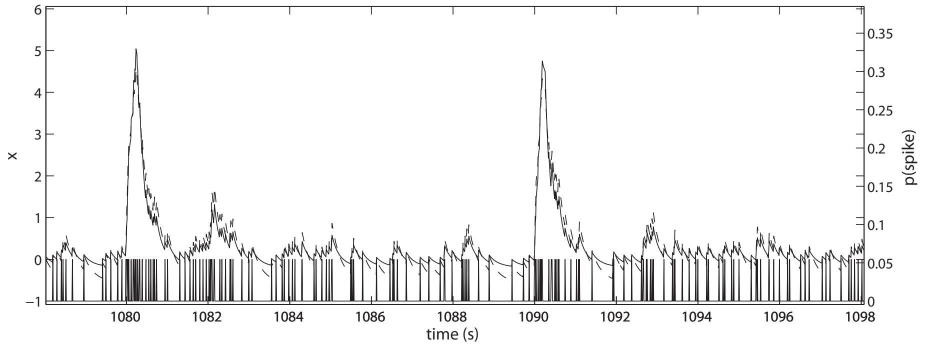


Figure 6.1: A 20s segment of training data taken from the cell 9 response to NaCl. The time duration shown spans across two trials with the rinsing and phase 1 periods removed. The estimated state (*dashed line*) and probability of spike occurring (*solid line*) are seen to be indicative of the frequency of spike events (shown on the bottom axis).

Inference It is evident, from preliminary studies, that the dominant rate coding characteristics which differed across tastants were attributed to the input gain α and the background firing rate μ. We thus chose to monitor these two parameters (in addition to the underlying state) with both online and offline methods, in order to study the response behaviour whilst discriminating between the four tastants. In particular, the online VB, offline VB and the RMHMC methods.

For all experiment, we chose to fix the unknown parameters $\beta = 0.5$, $\sigma_\epsilon^2 = 0.05$ and $\rho = 0.95$, which ensure nice convergence performance for all methods. Some additional details, regarding to implementations are the following:

- *Online VB*, the relevant forgetting factors were set to $\eta^\mu = 0.999$ and $\eta^\alpha = 0.9$ respectively. To ensure convergence, the online parameter updates are carried out only in the regions where ample information is present, so that α was only updated in regions of input application and μ in regions between the application of the respective inputs.
- *Offline VB*, for all experiments, convergence within 100 iterations with relative changes in parameter less than 10^{-3}.
- *RMHMC*, uses a step size of 0.05 and 60 integration steps for the states, and a step size of 0.8 and 8 integration steps for parameters. Again, a random integration direction is applied at each generalised leapforg loop. On top of these settings, for all experiments, $20,000$ posterior samples are collected after 1000 burn-in period.

Results First, we demonstrate the effectiveness of the online VB method, for which a representative filtered state and output probability of a spike occurring for the tastant NaCl in cell 9 is shown in figure 6.1. Note how the firing probability adequately captures the behaviour of the spike train.

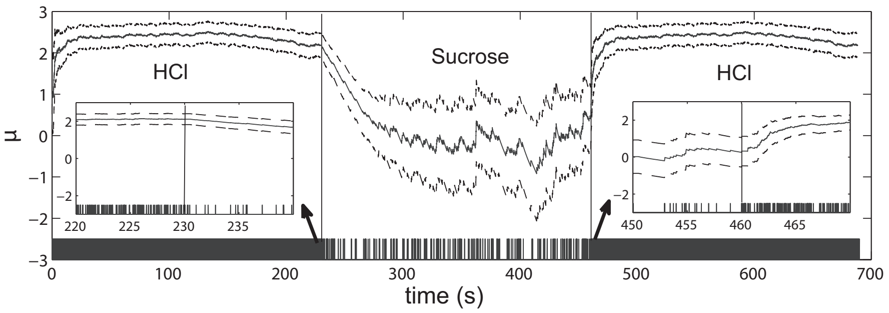


Figure 6.2: Tracking the mean (solid) and corresponding 99% intervals (dashed) of μ indicating a change in stimulus from HCl to sucrose and back to HCl in cell 9. The parameter change is indicative of a change in the spike train pattern (inset) when the stimulus is changed. The solid vertical lines indicate where the change in applied chemical stimulus took place. For this trial α was fixed to 0.1.

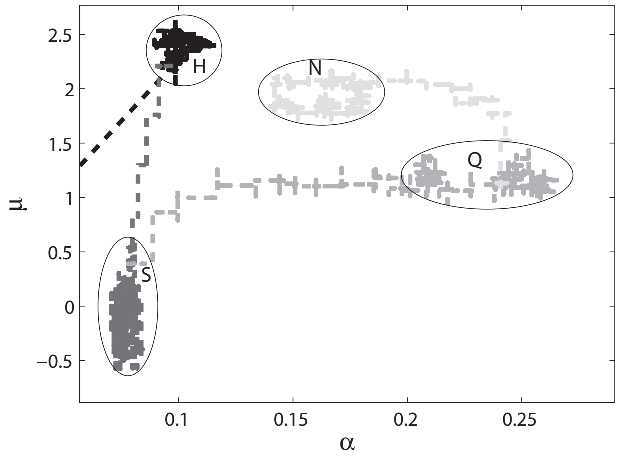


Figure 6.3: Cell 9; temporal progression of the estimated mean of α and μ indicating a change of stimulus from (in order of decreasing contrast) HCl (H) to sucrose (S) to quinine (Q) to NaCl (N). Although the cell is, overall, less responsive (μ) to quinine, the immediate effect of its application (α) is more relatively substantial than in the case of both HCl and NaCl. The ellipses define arbitrarily chosen classification boundaries.

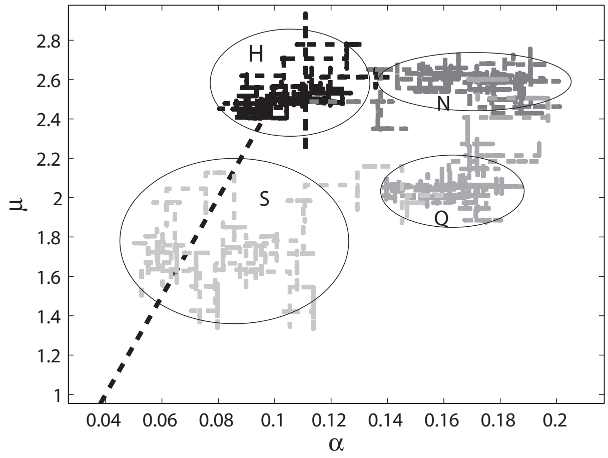


Figure 6.4: Cell 11; temporal progression of the estimated mean of α and μ indicating a change of stimulus from (in order of decreasing contrast) HCl (H) to NaCl (N) to quinine (Q) to sucrose (S). From this chart it is evident that α or μ on their own cannot capture the difference in response to the different tastants. The ellipses define arbitrarily chosen classification boundaries.

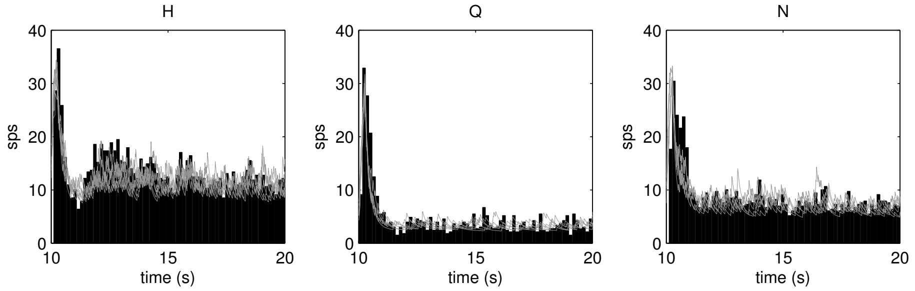
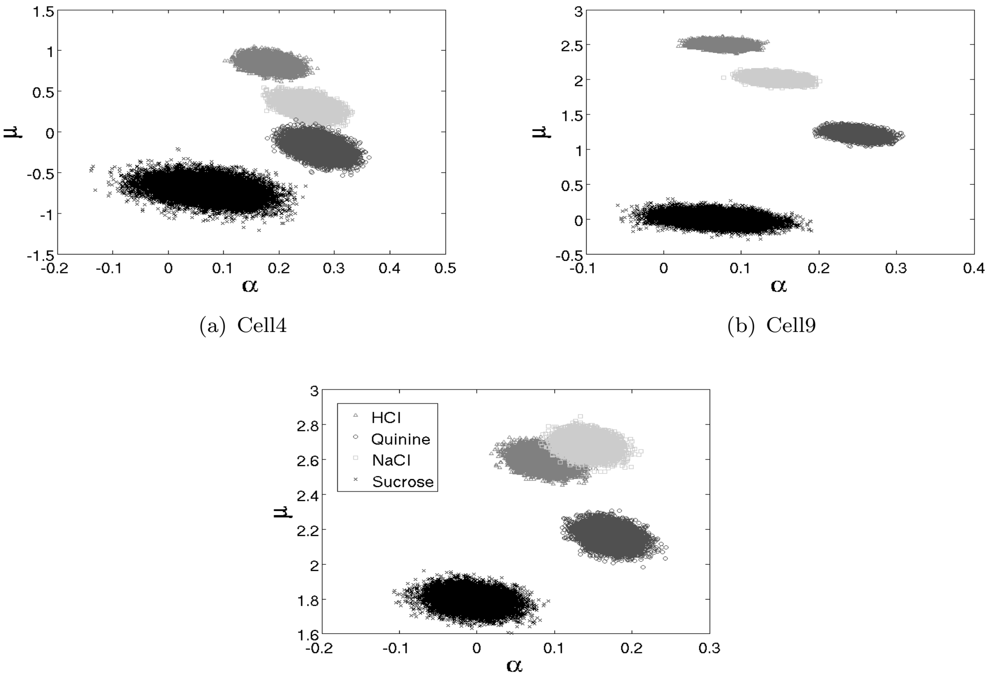


Figure 6.5: The estimated firing rate in spikes per second (sps) from five randomly selected trials (grey) in the online data, overlaying the PSTH (black) of responses to the respective stimuli, H (HCl), Q (quinine) and N (NaCl) in cell 9. The approximate firing rate is computed as $p(y_k = 1|x_k, \boldsymbol{\theta}_k)/\Delta$.

Both the change in α and that in μ were very evident across the different experiments. In some cases, monitoring μ is sufficient to characterise the difference in response to different tastants (see figure 6.2 for a comparison of sucrose with HCl in cell 9). However, this is not the general case, as shown by the trajectories of the mean parameter estimates of α and μ in figure 6.3 and 6.4. For instance, whilst μ seems to vary across tastants in cell 9 (figure 6.3), the background firing rate in response to NaCl and HCl for cell 11 are fairly similar (figure 6.4). It is the input gain α which is different between these two responses. By monitoring the parameters μ and α, the responses are seen to cluster in distinct and separate regions characteristic to the stimulus being applied.

These results are confirmed by the RMHMC method in figure 6.6, showing the robustness of the online approach. One interesting finding is that, For cell 4, in which the number



(a) Cell4

(b) Cell9

(c) Cell11

Figure 6.6: Posterior distributions of α and μ given the observed spike trains in Cells 4, 9 & 11. The parameter space shows good separation of the four tastes.

of spikes is significantly smaller than it in cell 9 and 11 (this can be seen in figure 6.9), the online does not produce the nice separation as RMHMC in figure 6.6(a).

It is also interesting to note that, except for sucrose, neither response can be considered to be passive, *i.e.* has both a low α and a low μ. The responses exhibit prominent activity either in either the initial stage, or the steady-state stage (the phasic and tonic stages respectively), or both. The considerable activity in the initial stage even when the overall reponse μ is low (particularly with quinine), is also somewhat of a testimony to the hypothesis that the initial neural response to every tastant may contain some additional information, encoding for instance a measure of taste acceptance (known as the hedonic value, see Di Lorenzo and Victor, 2003).

Finally, we conclude by showing how the online algorithm also manages to accurately give a rate envelope over the responses as indicated by the multiple overlays on the PSTHs in figure 6.5. The VB filter manages to approximate the PSTH in each trial validating the appropriateness of this model for characterisation of the rate encoding properties of this neuron.

Model goodness-of-fit We also assess the model goodness-of-fit in figures using the time rescaled theorem based KS test (for details, see Brown et al. (2002)), and use it to compare the posterior distributions obtained by VB and MCMC (RMHMC) methods. The results show that while Precisely, the time rescaled theorem based KS test is designed within maximum likelihood (or EM in SSPP) setting. In order to be able to adapt to compare posteriors, we compute the rate in a Bayesian fashion,

$$\lambda_c(k\Delta|\mathbf{y}_{1:K}) = \int_{x_k}\int_{\mu} \exp(\mu + \beta_c x_k) p(x_k, \mu|\mathbf{y}_{1:K}) dx_k d\mu. \tag{6.1}$$

In the VB setting, the Bayesian rate is computed as

$$\lambda_c(k\Delta|\mathbf{y}_{1:K}) = \exp(\hat{\mu} + \frac{1}{2}\sigma_{\mu}^2 + \beta_c x_{k|K} + \frac{1}{2}\beta_c^2 \sigma_{k|K}^2), \tag{6.2}$$

where,

$$\begin{aligned} \hat{\mu} &= \mathbb{E}_{q(\mu)}[\mu], & \sigma_{\mu}^2 &= \mathbb{E}_{q(\mu)}[\mu^2] - \hat{\mu}^2, \\ x_{k|K} &= \mathbb{E}_{q(x_k)}[x_k], & \sigma_{k|K}^2 &= \mathbb{E}_{q(x_k)}[\mathbf{x}_k^2] - x_{k|K}^2. \end{aligned} \tag{6.3}$$

where, $q(\mu)$ and $q(x_k)$ are the variational posteriors, more precisely, Gaussian approximations. In the online setting, the $q(\mu_k)$ and $q(x_k)_{online}$ are employed to computed the Bayesian rate.

In the RMHMC setting, the Bayesian rate is computed as

$$\lambda_c(k\Delta|\mathbf{y}_{1:K}) = \frac{1}{M}\sum_{m=1}^{M} \exp(\mu^{(m)} + \beta_c x_k^{(m)}), \tag{6.4}$$

where m and M are the sample index and total number of samples produced by RMHMC, respectively.

Once the rates are computed, the algorithm 3.2 can be used to perform the time rescaled KS test. The results are shown in figures 6.7 and 6.8, in which RMHMC obtained a slightly better fit in Cell 4, whereas in Cells 9 and 11 RMHMC and VB have the similar performance. The comparison is mainly concerned with offline methods. The results of online VB are also shown in the figures, however, compare them with offline methods with a fitting criteria is not fair, since the online approach has additional freedoms to fit the data.

In figure 6.9, we show the expected spiking probability over states and parameter posteriors obtained from VB (online and offline) and RMHMC. Note the increases expected probability synchronous with the appearence of spikes. This comparison suggests that in data with a significant number of spikes, VB appears to perform better and the MCMC approach is better suited to data in which the spikes are sparse. Our intuition on this is that when spike count is low, the uncertainty within the posterior is relatively high (see e.g figure 6.6(a) and 6.6(b)). MCMC, therefore is more flexible to handle the uncertainty. VB methods, on the other hand, often underestimates the uncertainty within the state transition process due to the independent assumption of the mean-field approximation. (Turner and Sahani, 2010). Note that the number of data points is large in this dataset,

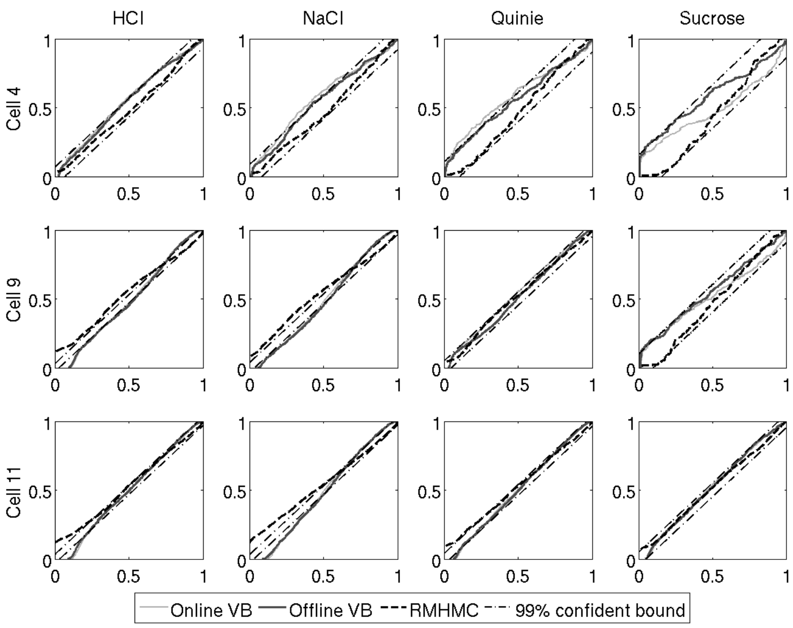


Figure 6.7: Q-Q plot based on time rescaling theorem (Brown et al., 2002) of inferred model by RMHMC, offline VB and online VB. x-axis shows the quantiles and y-axis shows empirical cumulative rate function. 99% confident intervals are indicated by dash line in each figure. 45^o line indicates a perfect match.

and given the fact that the posteriors are unimodal, it is reasonable to expect MCMC and VB showing similar performance.

The next subsection shows results from a different dataset in which the data record is much shorter in time and spiking is sparse.

6.1.2 Parvocellular neuron dataset

Data We now consider another dataset from Victor et al. (2007), where the response variability of marmoset parvocellular neurons under drifting sinusoidal luminance gratings stimulus is considered. Single cell spiking activities are recorded, where the luminance modulation (LUM) stimuli are presented at 10 different ascending contrast levels [2]. Each contrast is repeated 13 times within a 3.5s period for 3 trials. We treat the 3 trials as 3 parallel channels of spike trains driven by the same stimulus. The time resolution is set to 0.002s, which guarantees one spike per time bin and yields 1750 time points for each channel.

Inference Offline VB and RMHMC are employed to target the posterior distributions of α, μ and hidden states given observed spike trains. We fix $\rho = 0.8$, $\sigma_\varepsilon^2 = 0.05$ and $\beta = 1$ for each channel. Other implementation details are the following:

[2] 0, 0.0156, 0.0312, 0.0625, 0.0937, 0.125, 0.25, 0.375, 0.5 and 1. Data is from cell MY107.

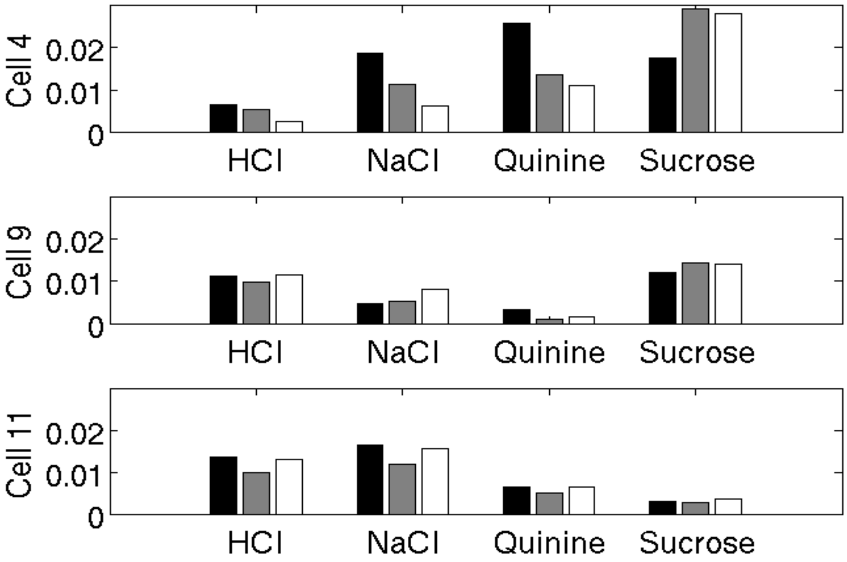


Figure 6.8: Maximum KS distance for each cell with each taste stimulus. Where each block of vertical bars correspond to offline VB, online VB and RMHMC from left to right.

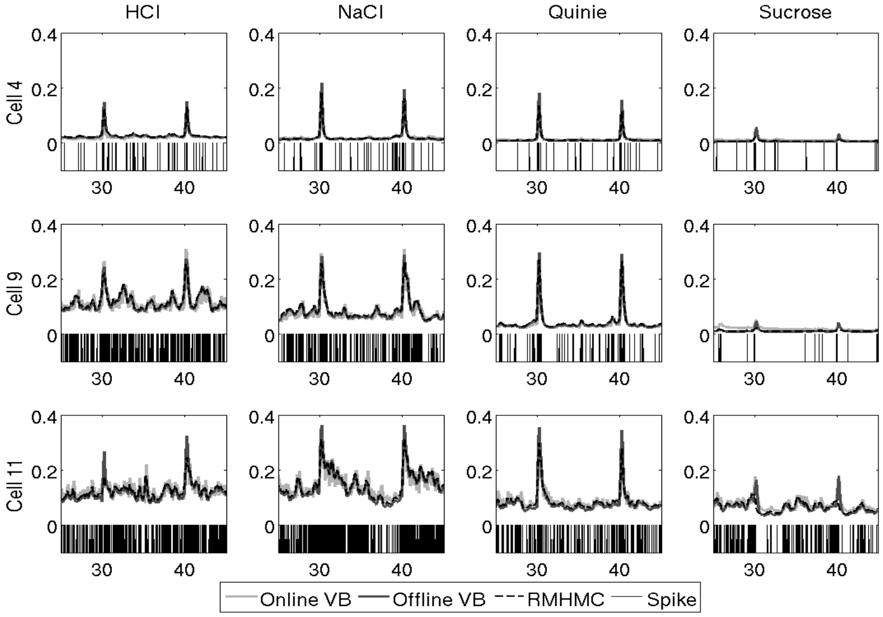


Figure 6.9: 20s segment expected spiking probability with respect to state and parameter posteriors obtain by RMHMC, offline VB and online VB (graphs overlap because the differences between the methods are small). For each panel, x-axis denotes time with unit in seconds, y-axis denotes the expected spiking probability measure. The observed spike train is also shown in black bars.

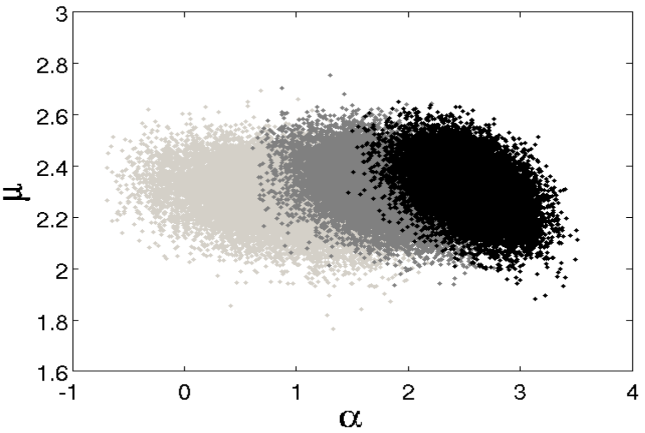


Figure 6.10: Joint α and μ posteriors; clusters from right to left correspond to contrast values of 1, 0.5 and 0.35.

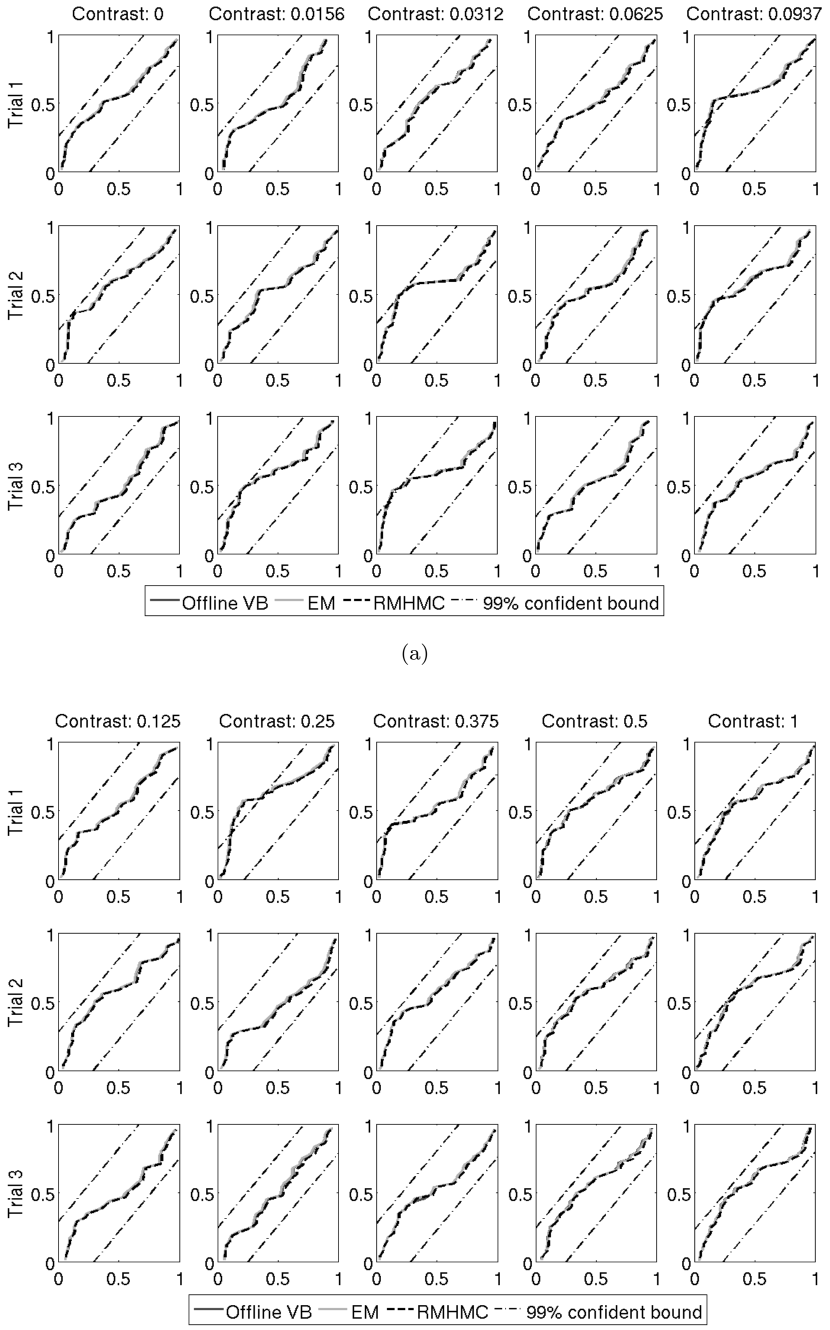
- *VB*, for all experiments, convergence within 1000 iterations with relative changes in parameter less than 10^{-3}. Note that due the fact that the observation is less informative about the states, VB needs longer time to converge. This slow convergence problem is shared with the approximate EM.

- *RMHMC*, uses a step size of 0.2 and 15 integration steps for the states, and a step size of 0.8 and 5 integration steps for parameters. Again, a random integration direction is applied at each generalised leapforg loop. On top of these settings, for all experiments, $20,000$ posterior samples are collected after 1000 burn-in period.

Results As shown in figure 6.10, the resulting posteriors overlap heavily even for the strongest three contrast levels, therefore it is not easy to distinguish between trials with different stimulus types within posterior space. However, the inferred model is still able to characterize the data quite well according the KS test results, as shown in figures 6.11 and 6.12. In this case, the amount of the data is significantly less than the taste response dataset considered previously. The RMHMC consistently outperforms both EM and VB in terms of the model goodness-of-fit across each of the 10 contrast levels. Finally, the expected spiking probabilities are consistent with the data (figure 6.13).

6.1.3 Discussion

With two real neural spike train datasets, the SSPP framework is able to modelling the neural dynamics while providing discriminative features for spike classification. In addition, these two datasets are used for comparing VB and MCMC representations of the posteriors in SSPP. Specifically, we use a model fitting measure as the comparison criteria. The results match our intuition, that is, when more uncertainties present in the data, MCMC is a better option, whereas, when the data is rich and informative about inference target, VB and MCMC have the similar performance. However, our approach



(a)

(b)

Figure 6.11: Q-Q plot based on time rescaling theorem, (a) and (b) show results on contrast from 0 to 0.0937 and 0.125 to 1, respectively. Offline VB, EM and RMHMC are drawn, and overlap heavily. 99% confident intervals are shown as dashed lines.

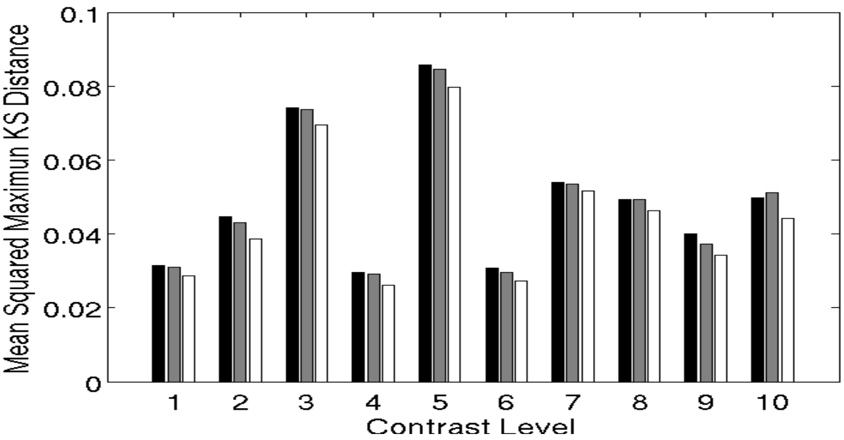


Figure 6.12: The mean squared maximum KS distance of each contrast level. Contrast level 1 to 10 denote 0.0156 to 1 in the dataset. Each block of vertical bars correspond to EM, Offline VB and RMHMC from left to right.

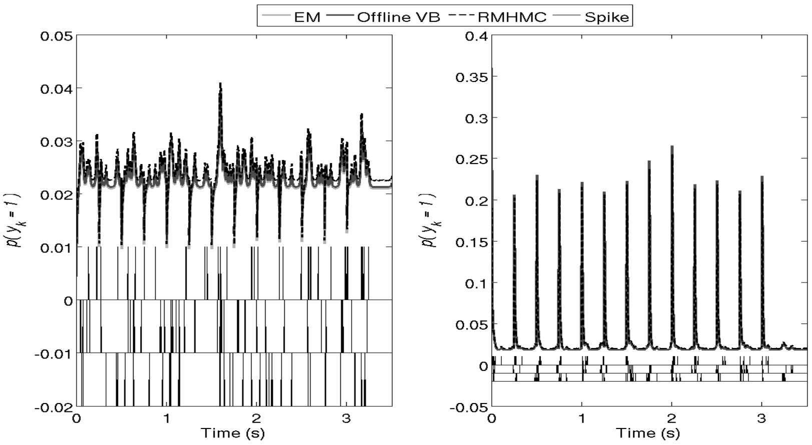


Figure 6.13: Expected spiking probability with respect to state and parameter posteriors obtained by RMHMC, EM and offline VB. The left panel and right panel correspond to contrasts of 0 and 1.

is not entirely satisfying in a Bayesian perspective. As such, the model goodness-of-fit is not a target of Bayesian inference, in fact, intuitively, Bayesian trends choose the many suboptimal model fittings. In other words, it tries to average over many possible solutions to the inference problem. Comparing the how well the obtained model fits the data is somehow opposite to such a spirit. Technically, the time rescaled KS test does not compare high order (higher than second moments) statistics of the posterior, and correlation between time steps or correlation between states and parameters. Potentially, these are measures on which MCMC methods may outperform VB, due to its highly structural assumptions.

For spike train classification, Salimpour et al. (2011) show an interesting approach of using the likelihood based on filtered estimates as a discriminator for spike trains responding to various stimuli. Their model treats parameters of the CIF as states in deriving an extended Kalman filter estimator. This differs from the model we considered, which has a separate underlying dynamical state process that can be used to capture the underlying neural dynamics. Algorithmically, Salimpour et al. (2011)'s work has similarities to the Laplace approximation-based adaptive filters (Smith and Brown, 2003; Eden et al., 2004; Koyama et al., 2010).

6.2 Heartbeats data analysis

In this section, we consider an application of heartbeat analysis. The purpose of the modelling task is to relate heartbeat to the autonomic nervous system (ANS) inputs: sympathetic and para-sympathetic inputs. These two inputs are also known as the sympathetic nervous system (SNS) and the para-sympathetic nervous (PSNS), because they influence not only the heartbeat, but also several other organisms including bronchi and kidney. Normally, it is difficult to measure these inputs directly; instead, they are concerned as harmonic signals with certain frequency bands. More specifically, the SNS is believed to be centered within $0.04 - 0.15$Hz frequency band, and the PSNS is centred within the $0.15 - 0.4$Hz band.

There are two major differences between heartbeat (or heart rate) and neural spikes. i) The inter-pulse intervals (IPIs) in heartbeat are long and strongly regulated, suggesting long and severe negative history dependency. As a contrast, IPIs in neural spikes are much shorter and exhibit large variations. ii) Sometimes neural spikes are available in multi-channel format, which provides more data for inference. Unfortunately, such amount of information is not available in heartbeat (heart rate) signals. It is difficult to apply SSPP to heartbeat with these two differences; since the information in heartbeats is not enough to allow us to identify both hidden states and parameters. As a result, we replace the SSPP with a point process generalised linear model (PPGLM), in which the parameters still act as discriminative features.

6.2.1 A point process generalised linear model

To introduce the point process generalised linear model (PPGLM), we use the following notations. Let Δ be the time resolution. The observation interval is written as $t = (0, k\Delta, \cdots, K\Delta]$. Δ is set to be small, such that there is only one R-wave per each time bin. Define $\mathbf{y} = \{y_k\}_{k=1}^{K}$ as a vector of R-wave indicators over the observation interval. Specifically, at the kth time bin, $y_k = 1$ if a R-wave occurs and 0 otherwise. $\mathbf{u} = \{\mathbf{u}_k \in \mathbb{R}^2\}_{k=1}^{K}$ and $\mathbf{h} = \{\mathbf{h}_k = \{y_{k-j}\}_{j=1}^{M}\}_{k=1}^{K}$ to be the ensemble of inputs and

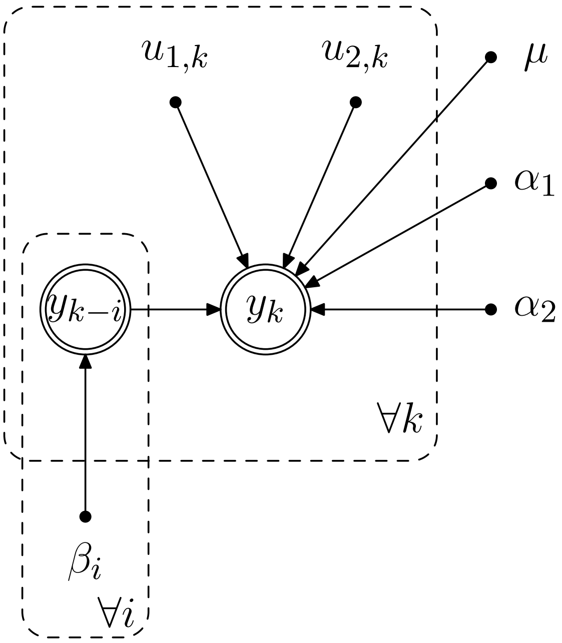


Figure 6.14: A graphical representation of the point process GLM for heartbeats.

history informations. The total likelihood of $\mathbf{y}$ can be written as

$$p(\mathbf{y}|\mathbf{u},\mathbf{h},\boldsymbol{\theta}) = \prod_{k=1}^{K} \exp\big(y_k \log(\lambda(k\Delta|\mathbf{u}_k,\mathbf{h}_k,\boldsymbol{\theta})\Delta) \\ - \lambda(k\Delta|\mathbf{u}_k,\mathbf{h}_k,\boldsymbol{\theta})\Delta\big), \tag{6.5}$$

where $\boldsymbol{\theta}$ is the model parameters vector. As described in chapter 3, equation 6.5 is an inhomogeneous Poisson likelihood model.

The CIF in this case is formed as a function of two time varying contributors: exogenous inputs and historical pulses. At each time point, we have

$$\lambda(k\Delta|\mathbf{u}_k,\mathbf{h}_k,\mu,\boldsymbol{\alpha},\boldsymbol{\beta}) = \exp\big(\mu + \sum_{i=1}^{2} \alpha_i \sin(\omega_i k\Delta + \psi_i) \\ + \sum_{j=1}^{M} \beta_j y_{k-j}\big), \tag{6.6}$$

where μ is the background contributor or log-baseline firing rate. The inputs are two harmonic functions with certain frequencies ω_i and phases ψ_i. In particular, the lower frequency (LF) (around 0.1Hz) stands for the SNS input, and the high frequency (HF) (around 0.3Hz) stands for the PSNS input. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the weights of $\mathbf{u}_k$ and $\mathbf{h}_k$, respectively. This formulation can be summarised by the graphical model in figure 6.14.

In practice, it is difficult to estimate the phase terms directly, a standard approach is to linearise them. For this, each harmonic function is expanded into two components. As a result, we redefine

$$\mathbf{u}_k = [\cos(\omega_1 k\Delta), \sin(\omega_1 k\Delta), \cos(\omega_2 k\Delta), \sin(\omega_2 k\Delta)]^{\mathrm{T}}, \tag{6.7}$$

$$\boldsymbol{\alpha} = [c_1, c_2, c_3, c_4]^{\mathrm{T}}. \tag{6.8}$$

These parameters introduce no additional difficulty to the estimation, since α can be simply recovered by taking the ℓ_2-norm of $[c_1, c_2]$ for α_1 ($[c_3, c_4]$ for α_2). Let $\boldsymbol{\theta} = [\mu, \boldsymbol{\alpha}^{\mathrm{T}}, \boldsymbol{\beta}^{\mathrm{T}}]^{\mathrm{T}}$ and define a basis function vector $\boldsymbol{\phi}_k = [1, \mathbf{u}_k^{\mathrm{T}}, \mathbf{h}_k^{\mathrm{T}}]^{\mathrm{T}}$. The CIF can be rewritten as a exponential function of a weighted sum of basis functions:

$$\lambda(k\Delta) = \exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k). \tag{6.9}$$

The total likelihood $p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\theta})$ is

$$p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\theta}) = \prod_{k=1}^{K} \exp(y_k(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k) - \exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k)\Delta) \tag{6.10}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K]$.

In principle, there is no restriction on the functional form of the CIF, as long as it is a positive one. Exponential function, as the most popular choice has the advantage of ensuring the total likelihood function to be concave in the parameters given stimulus and historical pulses. The maximum likelihood estimation of the parameter is therefore always unique. Given such a appealing property, the estimated parameters may act as identity features for different heartbeat (heart rate) signals. In this work, we choose the weights of the two harmonic excitations $[\alpha_1, \alpha_2]$ as our major discriminator. As they represent the intensity of SNS and PSNS inputs, we are aiming to identify or explain heartbeat with different physiological conditions, through the control from ANS.

6.2.2 Maximum likelihood estimation (MLE)

The maximum likelihood estimation for GLM is often obtained by the iteratively reweighted least squares (IRLS) (Green, 1984). This method transforms the nonlinear optimisation problem into a weighted least squares problem embedded in an Newton's method. For PPGLM, the problem can be formulated as the following:

$$\min_{\boldsymbol{\theta}\in\Theta} -\log p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\theta}). \tag{6.11}$$

The above optimisation problem can be solved by the Newton's method with an update equation written as

$$\boldsymbol{\theta}^{(l+1)} = \boldsymbol{\theta}^{(l)} - \mathbf{H}^{-1}(\boldsymbol{\theta}^{(l)})\mathbf{g}(\boldsymbol{\theta}^{(l)}), \tag{6.12}$$

where $\mathbf{H}(\boldsymbol{\theta})$ and $\mathbf{g}(\theta)$ are the Hessian and gradient of the negative likelihood in equation (6.11) evaluated at $\boldsymbol{\theta}^{(l)}$. In particular, they take the following forms:

$$\mathbf{g}(\boldsymbol{\theta}) = \boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{y} - \mathrm{diag}(\mathbf{W}(\boldsymbol{\theta}))), \tag{6.13}$$

$$\mathbf{H}(\boldsymbol{\theta}) = \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{W}(\boldsymbol{\theta})\boldsymbol{\Phi}, \tag{6.14}$$

where $\mathbf{W}(\boldsymbol{\theta})$ is a K-by-K diagonal matrix with elements $W_{kk} = \exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k)\Delta$. According to the standard IRLS procedure, finding $\boldsymbol{\theta}^{(l+1)}$ is equivalent to solving a weighted least square problem,

$$\min_{\boldsymbol{\theta}\in\Theta} \left|\left|(\mathbf{W}^{\frac{1}{2}}(\boldsymbol{\theta}^{(l)})\boldsymbol{\Phi})\boldsymbol{\theta}^{(l+1)} - \mathbf{W}^{\frac{1}{2}}(\boldsymbol{\theta}^{(l)})\mathbf{z}\right|\right|_2^2 \tag{6.15}$$

where $\mathbf{z} = \boldsymbol{\Phi}\boldsymbol{\theta}^{(l)} + \mathbf{W}^{-1}(\boldsymbol{\theta}^{(l)})(\mathbf{y} - \mathrm{diag}(\mathbf{W}(\boldsymbol{\theta}^{(l)})))$. The above problem can be easily solved by a conjugate gradient (CG) method. In addition, the Hessian of the negative likelihood is also called the observed information matrix, which can be used to compute the standard error of the maximum likelihood estimation:

$$\mathrm{std}[\theta_{\mathrm{ML}}] = \sqrt{\mathrm{diag}(\mathbf{H}^{-1}(\boldsymbol{\theta}_{\mathrm{ML}}))}. \tag{6.16}$$

6.2.3 Synthetic examples

Simulation We firstly demonstrate that the model is able to generate heart rate time series which are close to the real ones. For this, we use the following settings:

Example 6.1. *The time resolution $\Delta = 5ms$, $\mu = 2$, and $\boldsymbol{\alpha} = [0.9, 0, 0.3, 0]^{\mathrm{T}}$. The two frequency components are chosen as $[\omega_1, \omega_2] = [0.1, 0.3]Hz$. Note that the ω_1 and ω_2 are given in the estimation. The window of history information is set to be $1s$, resulting $\boldsymbol{\beta} \in \mathbb{R}^{200}$. These βs are chosen as $\boldsymbol{\beta} = [-30, \cdots, -0.1]./[1, \cdots, 2]$. The two intervals are equally spaced and the symbol ./ denotes elementwise division.*

The synthetic data are generated by the following simple rule in algorithm 3.1 in chapter 3. Specifically, for each point in time, y_k is a sample from Bernoulli distribution with parameter $\lambda_k\Delta$.

With these settings, figure 6.15 shows 10min synthetic R-wave events. Four attributes are displayed for assessing the validity of approximating heartbeats. Firstly, the heart rate time series, which are obtained from the reciprocals of the R-R intervals and rescaled in beat per minute (bpm) unit, are valued within the normal heart rate region $50-100$bpm. Secondly, the distribution of the heart rates is similar to the normal heart rates. Specifically, the distribution appears to be close to Gamma or inverse Gaussian distribution, with a mode around 65bpm. Thirdly, we examine the frequency profile of the simulated heart rates. Due to the fact that the heart rate time series is not evenly

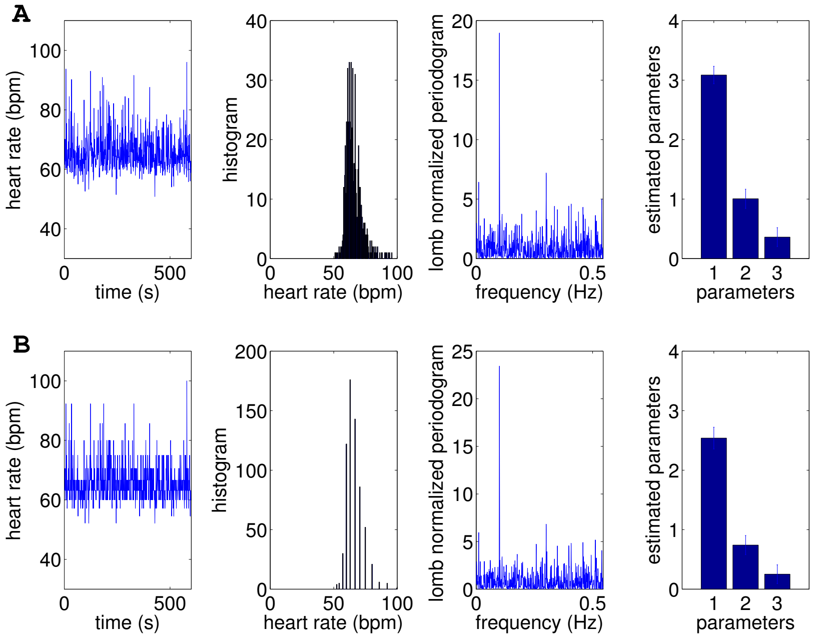


Figure 6.15: 10min heart rate generated by the PPGLM model. The parameters are set as described in example 6.1. (A) The time resolution is 5ms. (B) The time series in (A) is down-sampled such that the time resolution becomes 50ms, and for each time bin, there is still only one pulse. For both (A) and (B), the panels listed from left to right are: Time domain signal; Histogram; Spectrum of the signal obtained by the Lomb-Scargle method and the MLE of μ, α_1 and α_2 with standard deviation.

spaced, we use the Lomb-Scargle method as suggested by Moody (1993). Finally, the IRLS is employed to estimate back the background rate and the weights of the two ANS inputs. The discretisation of the observation interval or time axis in example 6.1 leads to $120,000$ data points. Here, we substantially down sample the generated R-wave by a factor of 10, resulting in the time resolution being 50ms and only $12,000$ data point. Further, unlike the case in neural spike, the down-sampling doe not bring in any cases which two R-waves fall into the same time bin. With the down-sampled R-waves, the above four attributes are also tested and shown in figure 6.15. The results do not change much, indicating that $\Delta = 50$ms is a decent choice for discretisation for R-wave indicators.

Estimation quality In figure 6.15, we have shown the performance of MLE for μ and $\boldsymbol{\alpha}$ from a single realisation, in which the MLE is shown to be accurate. This is what we expected, given the fact that the likelihood is concave and MLE is known to be asymptotically unbiased. However, in practice, it is still not clear how long a time series we need to confidently estimate the parameters, and use them as discriminative features. To address this question, we conducted a simulation, in which four time-length (5, 10, 30 and 60 minutes) settings are tested against the relative estimation error, computed

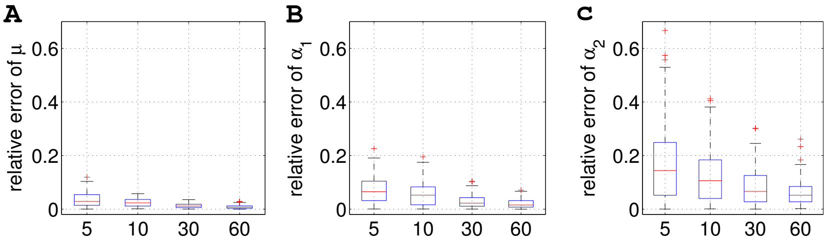


Figure 6.16: Relative error (computed as $|\hat{\theta} - \theta|/|\theta|$) in the DC coefficients μ, the first and the second harmonic coefficients α_1 and α_2, as a function of the length of the time series in minute unit. Each attribute on x-axis is constructed with 100 realisations.

as $|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}|/|\boldsymbol{\theta}|$. In addition, the robustness of the MLE is also tested based on 100 realisations under each setting. The parameters are the same as those in example 6.1, except where $\Delta = 50$ms. The results are shown in figure 6.16. Most errors in μ and α_1 are less than 20% of the true value, when only 5min data is available. In fact the mean of these errors is less than 10% of the true value. For α_2, the quality is worse, but the mean of the errors is still less than 20% of the true value. As the data length increases, it improves the estimation quality. In summary, the quality check tells us that the framework can be effectively applied to short-term and long-term heart rate time series.

So far, we have not discussed the estimation quality of $\boldsymbol{\beta}$. In theory, MLE should be always able to find the optimal value. However, when dealing with heartbeats, the problem appears ill-posed to some extent. Specifically, in order to ensure that the probability of R-wave occurring is strongly suppressed during a period immediately after a R-wave event, the β_js governing the history information within such a period have to be very small.

In the previous experiment, $\{\beta_j\}_{j=1}^{20} \in [-30, -0.5]$. Accurate estimation of these small β_js is difficult. As illustrated in figure 6.17, the overall performances are almost identical across the four time-length settings. For β_js that are larger than -5, the estimation is accurate and robust, as the true mean and estimated mean computed from 100 realisations are in strong agreement. An interesting phenomenon of the estimation in the case is that the estimated $\boldsymbol{\beta}$ is unable to reach a value below -10.

In this case, 10 β_js have a true value which is smaller than -10. In other words, during simulation, no R-wave will occur within the 0.5s right after each R-wave. Consequently, there are no data carrying information about how these 10 β_js are different from each other. In fact, according to figure 6.17, the IRLS tends to consistently estimate them with similar values; precisely, around -10 with very tight error bars across 100 realisations.

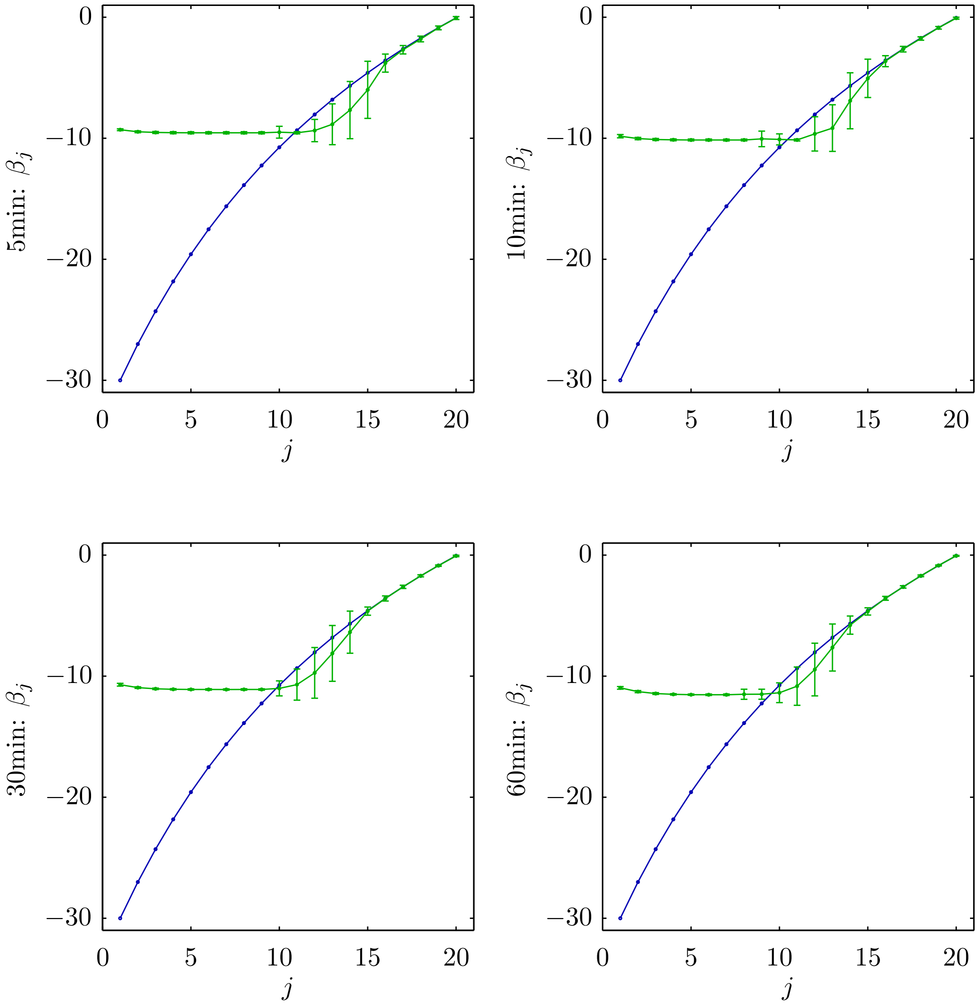


Figure 6.17: The performance of estimating $\boldsymbol{\beta}$ from heartbeats with four time-lengths (5, 10, 30 and 60min). The *blue line* is the truth, the *green line* with error bars are the mean estimated $\boldsymbol{\beta}$ with 1 standard derivation from 100 realisations.

The results in figures 6.16 and 6.17 are based on the very same datasets. Despite the estimation of the low-valued $\boldsymbol{\beta}$ being not accurate, the estimation of $\boldsymbol{\alpha}$ and μ is still reliable. This is true for $5-60$min-long heart rate series. According to Malik et al. (1996), 5min trials are often used for diagnostic analysis, since it is long enough to assess the frequency components from $0.04-0.4$Hz[3]. Hence, we can confidently use $\boldsymbol{\alpha}$ to be the candidate discriminative feature R-wave binary sequences.

Remark 6.1. *The estimation quality is assessed for fixed basis functions; in other words, for given frequency components.*

Separations Here, we show that with the synthetic R-waves, the PPGLM-based features do a better job on discriminating sequences with different simulation settings. For this, we construct simulations with five different settings on $\boldsymbol{\alpha}$ as the following:

[3] The time periods corresponding to 0.04Hz and 0.4Hz components are 25s and 0.25s respectively. 5min is long enough to confidently estimate the components within the frequency band.

S1: $\boldsymbol{\alpha} = [0.9, 0, 0.3, 0]^{\mathrm{T}}$

S2: $\boldsymbol{\alpha} = [0.9, 0, 0.9, 0]^{\mathrm{T}}$

S3: $\boldsymbol{\alpha} = [0.3, 0, 0.3, 0]^{\mathrm{T}}$

S4: $\boldsymbol{\alpha} = [0.3, 0, 0.9, 0]^{\mathrm{T}}$

S5: $\boldsymbol{\alpha} = [0.6, 0, 0.6, 0]^{\mathrm{T}}$

The parameters μ and $\boldsymbol{\beta}$ are the same as those in example 6.1. Based on these settings, 20 realisations of 5min R-wave events with $\Delta = 50$ms are generated for each of the five settings, which presents R-waves under different regulations from SNS and PSNS. With these data, we compare the visual separations in the PPGLM-based features and the traditional features.

In clinical practice, the most frequently used features for heart rate time series are the time and frequency features (Malik et al., 1996).

- **Time domain features:** The average of the normal-to-normal beat intervals (AVNN), and the standard derivation of the normal-to-normal beat intervals (SDNN). Here, a normal beat means a successful R-wave event.
- **Frequency domain features:** The total frequency amplitude within low frequency (LF) band: $0.04 - 0.15$Hz, and the total frequency amplitude within high frequency (HF) band: $0.15 - 0.4$Hz. On obtaining the frequency domain features, the Lomb-Scargle has been shown to be adequate (Moody, 1993).

The period or length of the recordings for computing these features is 5min as short term or 24 hours as long term. As mentioned above, for our purpose, which is the effect from two ANS inputs acting within the $0.04 - 0.4$Hz frequency band, the short-term 5min recordings are of interest.

Based on the five groups of simulated data, figure 6.18 shows the separations in the PPGLM feature space, α_1 vs α_2, and time-frequency domain features, AVNN vs SDNN, and total LF vs total HF. Evidently, the weights of two harmonic inputs separate the five groups well. This is not a surprise, since the data are simulated from the PPGLM model. The interesting observation is the performance of the time-frequency feature. The five groups appeal in the α_1 and α_2 space; These five clusters have a pattern with two apparent diagonals. Each time-frequency domain feature separates one of the diagonal clusters. In particular, the time domain features separate the diagonal, blue-magenta-red clusters. The black-magenta-green clusters are heavily overlapped. In the frequency domain, the black-magenta-green clusters are separated. Opposite to time domain features, the blue-magenta-red clusters are not visually separable.

To combine the time-frequency features, we group the four features together as a new feature vector for each recording. Then, the newly combined feature matrix is centred and projected onto its first and second principal components with principal component analysis. The results are also shown in figure 6.18, in which the separation has been improved, comparing time or frequency-only features. However, the α_1,α_2 space still provides a better separation. Moreover, the estimation of α_1 and α_2 comes with confident levels. Finally, being the weights and SNS and PSNS inputs, they are highly interpretable features, whereas the principal components cannot be easily connected to the underlying physiological knowledge.

6.2.4 Analysis of normal and sudden cardiac death patients

In this subsection, we examine the PPGLM features using heartbeats data from normal patients and sudden cardiac death patients. Both datasets are taken from the Physionet database (Goldberger et al., 2000). Specifically, the nsr2db dataset (54 subjects, 24hr long recording each) for normal patients and the sddb dataset (23 subjects, various recording length) for patients who suffered sudden cardiac death are chosen.

Frequency component selection When dealing with real data, the frequency components of the SNS and PSNS inputs, ω_1, ω_2, are unknown. On estimating ω_1 and ω_2, one can straightforwardly use the joint likelihood, equation (6.5), as a objective function. Optimising equation (6.5) with respect ω_1 and ω_2 could be problematic, since multiple local optimals exit. Confronting such a problem, we use a grid search method. Specifically, the search range for ω_1 is set to be $0.04 - 0.15$Hz, and the search range for ω_2 is set to be $0.15 - 0.4$Hz. In practice, we use a 20×20 grid, and for each point in the grid, compute the maximum likelihood solution for the other parameters. Finally, the optimal solution for the frequency components and parameters are those that correspond to the global optimal.

Data preprocessing The original data consists of a time-length of R-R intervals in both precise time and discrete sample index formats. Each R-wave event is annotated as normal or as other types. Here, only the normal beats (labelled as “N”) are taken into account for both nsr2db and sddb.

The Physionet provides a software package WFDB[4] to download and convert their data. In the following, we give examples of how the recordings for each subject are obtained with shell scripts.

- Normal-nsr2db:

```
ann2rr -r nsr2db/nsr001 -a ecg -f 0 -t 43200 -p N -c >
```

[4] http://www.physionet.org/physiotools/wfdb.shtml

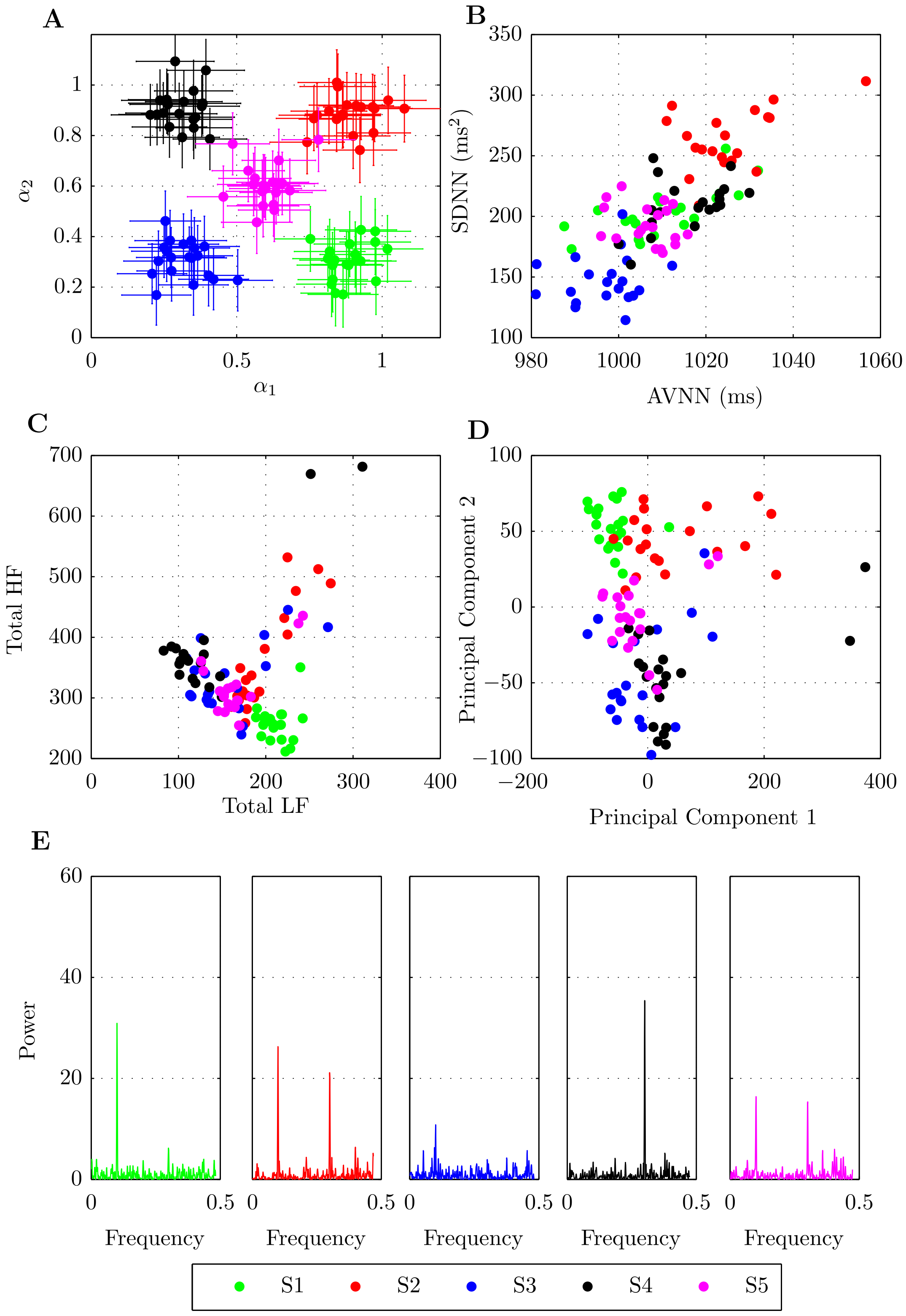


Figure 6.18: Separation between 5 groups of 5min heart rate time series generated from the PPGLM. The details of parameter settings are in the main text. (A): Estimated amplitudes of two frequency components: α_1 and α_2. The error bars are the standard deviations of the estimates obtained by equation (6.16). (B): AVNN vs SDNN. (C): Total power of the LF and HF. (D): Combination of the time-frequency features in (B) and (C), projected onto the first and second principal components. (E): A realisation of each group in the frequency domain via the Lomb-Scargle method.

```
"./nsr2db/rawdata/1st12hr/original/nsr001.rr.txt"
```

- Sudden death-sddb:

```
ann2rr -r sddb/30 -a ari -f 0 -t 43200 -p N -c >
"./sddb/rawdata/1st12hr/original/sd30.rr.txt"
```

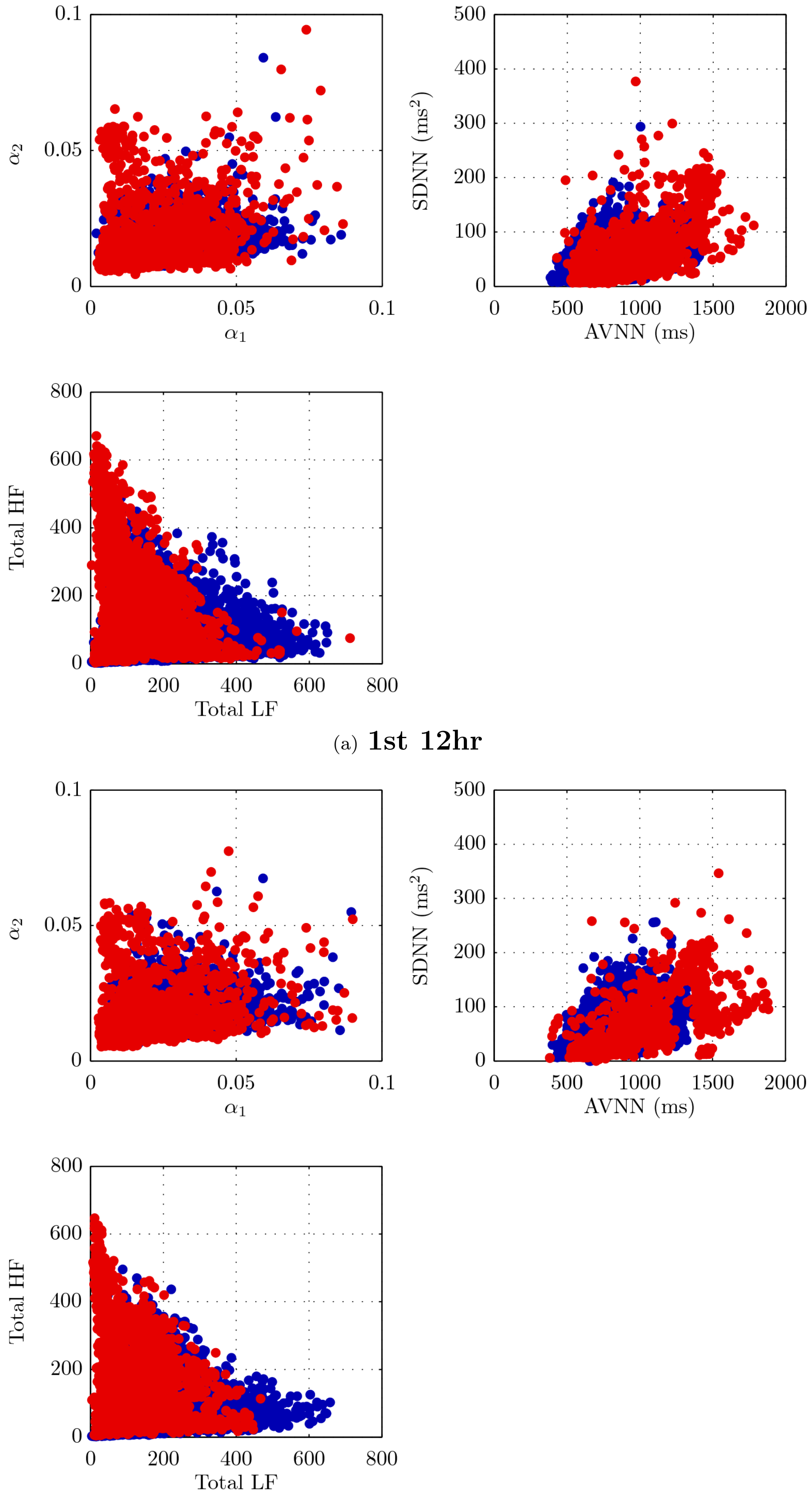
where ann2rr is a part of the WFDB package, handling transforming annotated data into R-R intervals. The 24hr recordings are split into two 12hr files. These procedures produce data files with sample-based R-R intervals. The time resolutions Δ are 7.8ms for nsr2db and 4ms for sddb. Then, the obtained data are cleaned by a strategy (as described by Moody (1993)) that removes abnormal instantaneous heart rates with extremely large or small intervals. The instantaneous heart rates are computed as $\mathrm{ihr} = \frac{60}{\mathrm{rr}\Delta}$.

The resulting R-R intervals are further split into 5min segments to extract time-frequency domain features. For the PPGLM features, the R-R intervals are converted to binary sequences and down-sampled by a factor of 10. As a result, the time resolution Δ changes to 78ms for nsr2db and 40ms for sddb. Similarly, 5min segments are chosen as the data-length within which to perform learning. Finally, the history dependency window is set to be 0.3s for all experiments.

Separation On separating all of the patients within nsr2db and sddb, none the three features achieves significant separation. This is demonstrated in figure 6.19. However, individual-wise, separation is much more evident. For this, we show three examples: nsr-48 vs sd-43 in figure 6.20, nsr-49 vs sd-40 in figure 6.21 and nsr-52 vs sd-31 in figure 6.22.

In figure 6.20, the trends of heart rate time series of the two subjects are almost identical, which means they cannot be separated by AVNN. The variance, however, is discriminative. The heart rate variance of the normal subject 48 (nsr2db-48) is considerably higher than that of the sudden death subject 43 (sddb-43), implying that, compared with sddb-43, nsr2db-48 has a larger spectral response in terms of amplitude. This can be read from the total LF vs. total HF space; also the α_1 vs α_2 which represent two single frequency component responses. In this case, the separations in AVNN & SDNN and α_1 &α_2 are better than the total LF & HF features.

Figure 6.21 shows another individual separation between normal subject 49 (nsr2db-49) and sudden death subject 40 (sddb-40). In this case, all features are able to show considerable differences between the two subjects. In particular, the AVNN of sddb-40 is larger than nsr2db-49, meaning that the sudden death subject 40 has a much slower heart rate. Again the heart rate variability in the sudden death subject is smaller than that in the normal subject. In the frequency domain, the two subjects can be easily



(a) **1st 12hr**

(b) **2nd 12hr**

Figure 6.19: Overall separation between 5min heartbeat recordings from nsr2db and sddb. For all panels, *red* and *blue* dots denote sddb and nsr2db, respectively. (a): Segments from the 1st 12hr recordings. (b) Segments from the 2nd 12hr recordings.

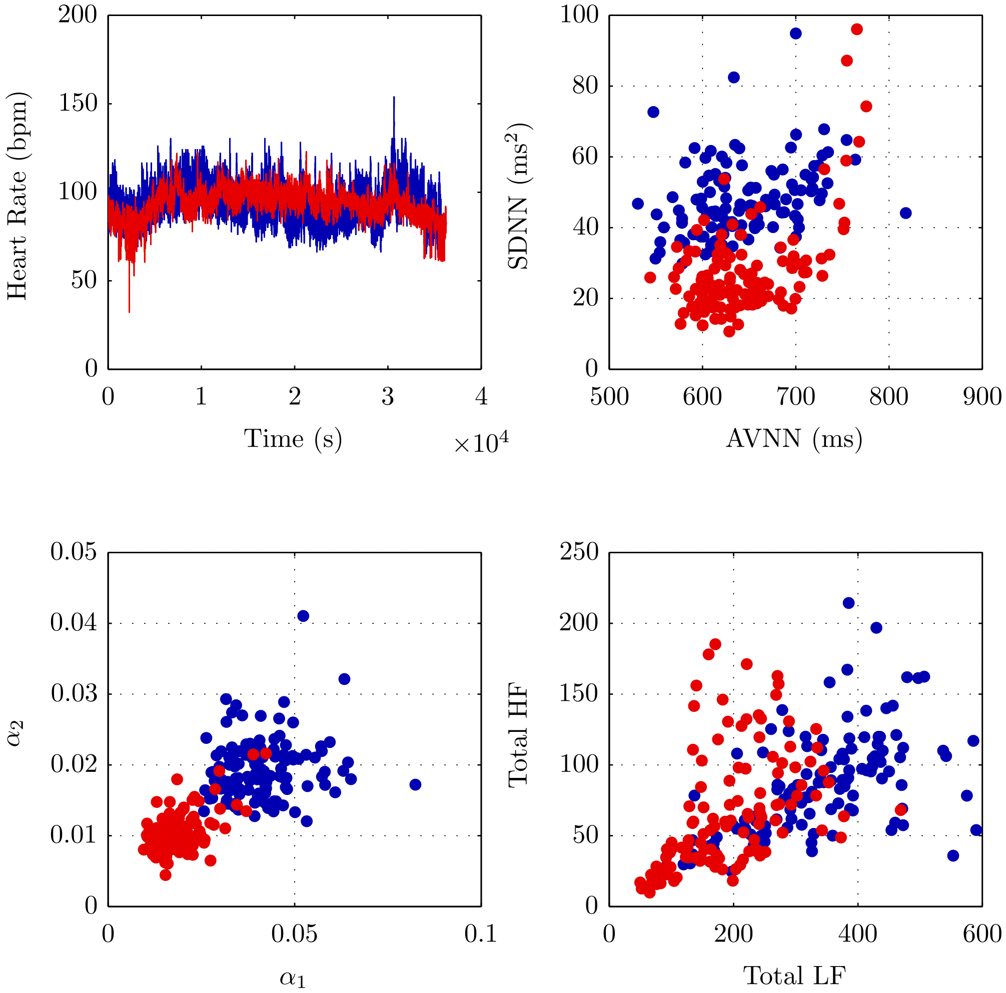


Figure 6.20: Separation between subjects nsr2db-48 and sddb-43 within the first 12hr recording, in heart rate time series, AVNN & SDNN, α_1 & α_2 and total LF & total HF. For each panel, the *red* and *blue* dots denote sddb and nsr2db subjects, respectively. Parameters are estimated using window length of 5mins.

separated by an evident shift between LF and HF. Similarly, totally opposite response of the LF/HF ratio can be observed in the α_1 and α_2 space. These indicate that the regulation from the SNS and PSNS swapped in their intensities. Particularly, in nrs2db-49, the SNS activation is stronger than the PSNS activation, whereas, in sddb-40, the PSNS becomes the major regulator.

The separation between normal subject 53 and sudden death subject 31 is also evident across the three discriminative features, as shown in figure 6.22. In this case, the separation is similar to those in figure 6.20. Particularly, the overall variation in the normal subject is much higher than in the sudden death subject. From the α_1&α_2, we can read that the overall intensity of the SNS and PSNS regulation is stronger in the normal subject. In terms of the total LF&HF comparison, a swapping pattern can also be observed. More examples like these can be found from the two datasets. For this, we

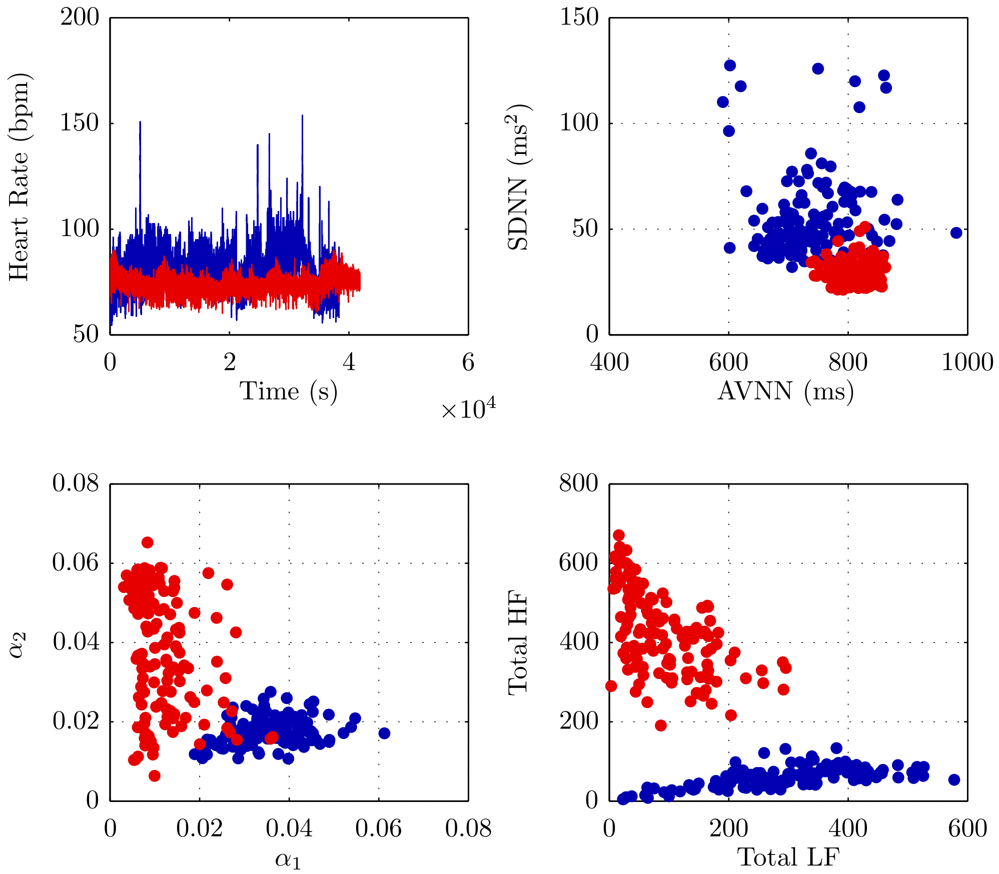


Figure 6.21: Separation between subjects nsr2db-49 and sddb-40 within the first 12hr recording, in heart rate time series, AVNN & SDNN, α_1 & α_2 and total LF & total HF. For each panel, the *red* and *blue* dots denote sddb and nsr2db subjects, respectively.

present the estimated $\alpha_1 \& \alpha_2$ for all subjects in nsr2db and sddb, as time series across the recording time, in figures 6.23-6.26.

Beside the heartbeat recordings, the age and gender information about the 54 subjects in nsr2db is available in table 6.1, from which two age-groups can be found; i.e. a group of 46 subjects (nsr2db-1 to nsr2db-46) who are older than 55 and a group of 8 subjects (nsr2db-47 tp nsr2db-54) who are younger than 40. According to figures 6.23 and 6.25, α_1 in the younger age group appear to be stronger than α_2.

To see this more clearly, we show the ratio $\frac{\alpha_1}{\alpha_2}$ in figure 6.27. Evidently, the mean ratios of the 8 subjects in the younger group are above or at least close to the line of 2, whereas the mean ratios of the majority of the older group lie in the interval $[1, 1.5]$. Comparing the $\frac{\alpha_1}{\alpha_2}$ ratio in the sddb subjects, most of the sddb subjects have relatively smaller valued ratio.

Almost all subject in nsr2db have mean ratios that are larger than 1, whereas, in sddb, for some cases, the ratio is smaller than 1. As α_1 and α_2 represent the activation of SNS and PSNS, respectively, these results imply that the sudden death patients may suffer

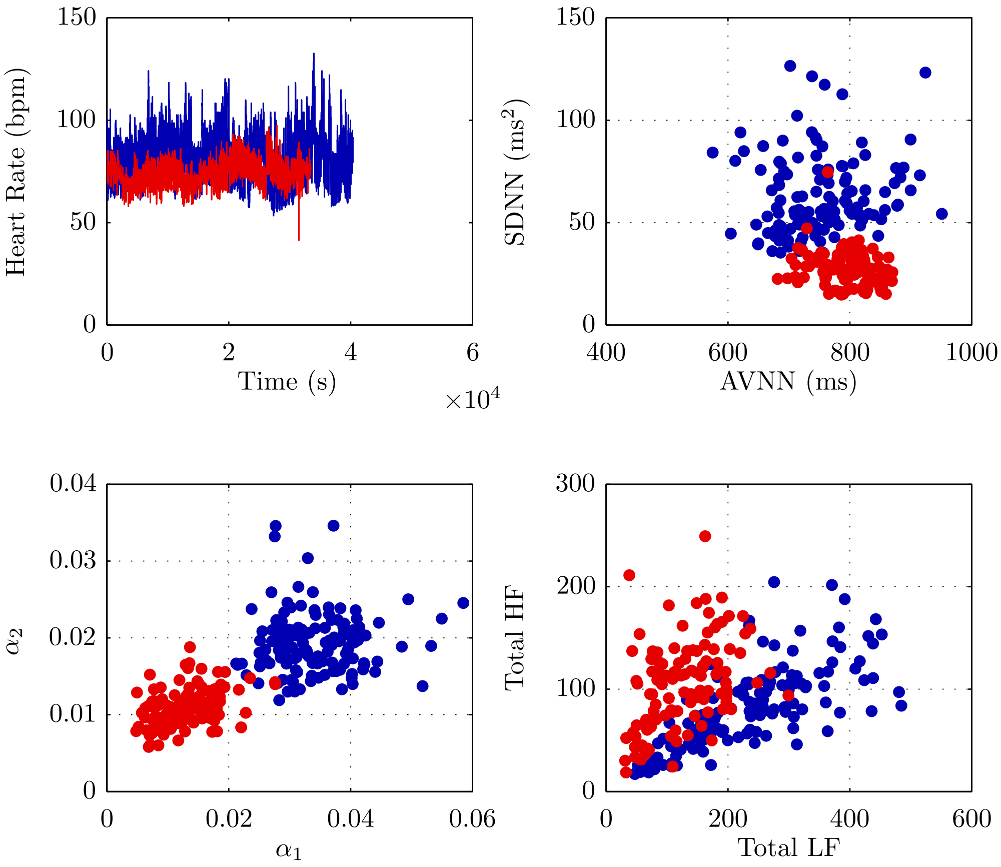


Figure 6.22: Separation between subjects nsr2db-53 and sddb-31 within the first 12hr recording, in heart rate time series, AVNN & SDNN, α_1 & α_2 and total LF & total HF. For each panel, the *red* and *blue* dots denote sddb and nsr2db subjects, respectively.

abnormal regulations from SNS and PSNS. In other cases, the regulation intensities of SNS and PSNS are completely opposite to those in the normal subjects. Since it is well known that the SNS increases the heart rate, and the PSNS activation decreases the heart rate (Guyton and Hall, 1991), we expected to see that the sddb subjects have relative larger AVNN or slower heart rate. This can be seen in figures 6.20-6.22.

6.2.5 Discussion

To model human heartbeat, we replace the SSPP framework with a PPGLM model, in which a sequence of R-wave event indicators are related to two harmonic signals, representing the SNS and PSNS activations. The parameters, particularly the weights or the intensities of the SNS and PSNS activations, can serve as discriminative features for diagnostics. The above results show that the performance of the PPGLM features are competitive with the traditional time-frequency domain features.

Algorithmically, when dealing with real data, how to determine the frequency components of SNS and PSNS is the major challenge. Currently, we are simply using a

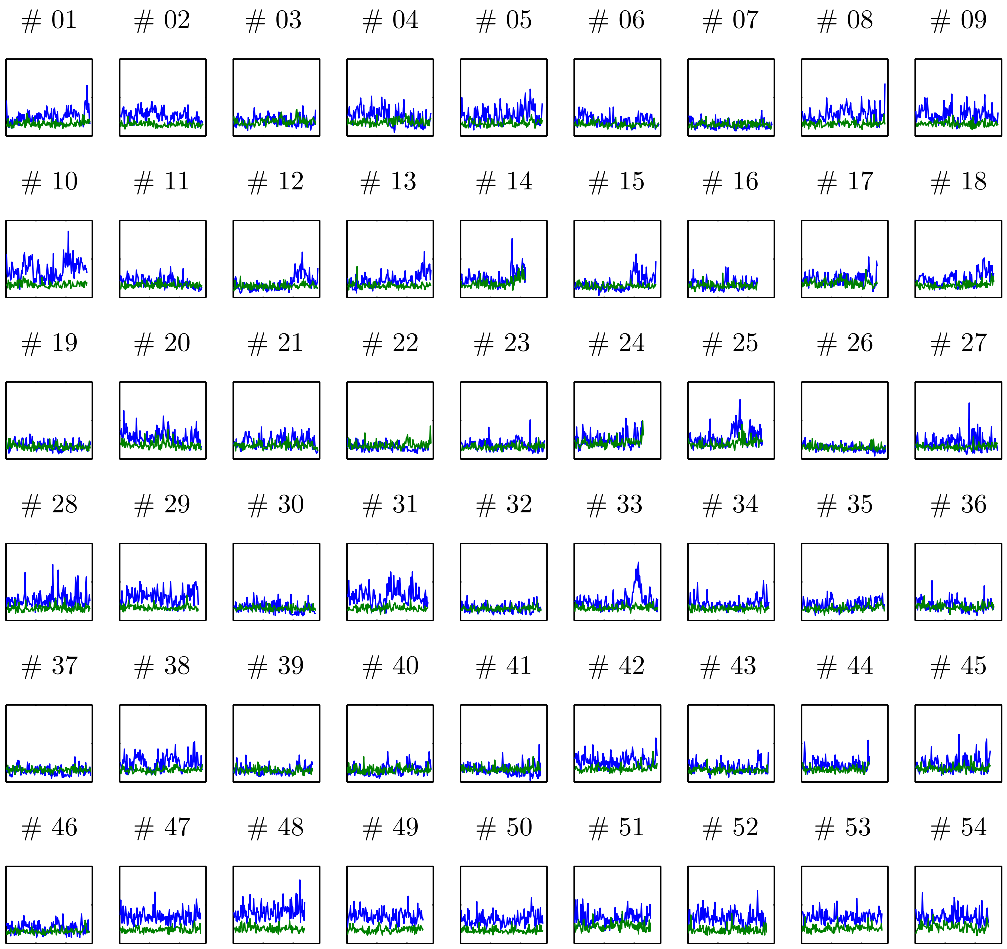


Figure 6.23: Estimated α_1 *blue lines* and α_2 *green lines* from 5min segments across the first 12hr recordings of the 54 subjects in nsr2db.

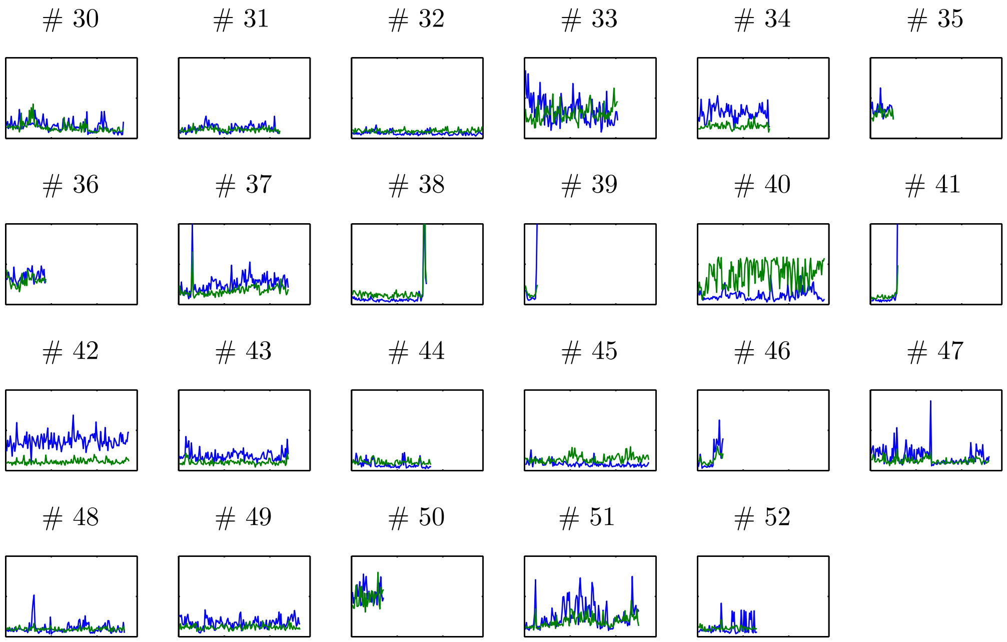


Figure 6.24: Estimated α_1 *blue lines* and α_2 *green lines* from 5min segments across the first 12hr recordings of the 23 subjects in sddb.

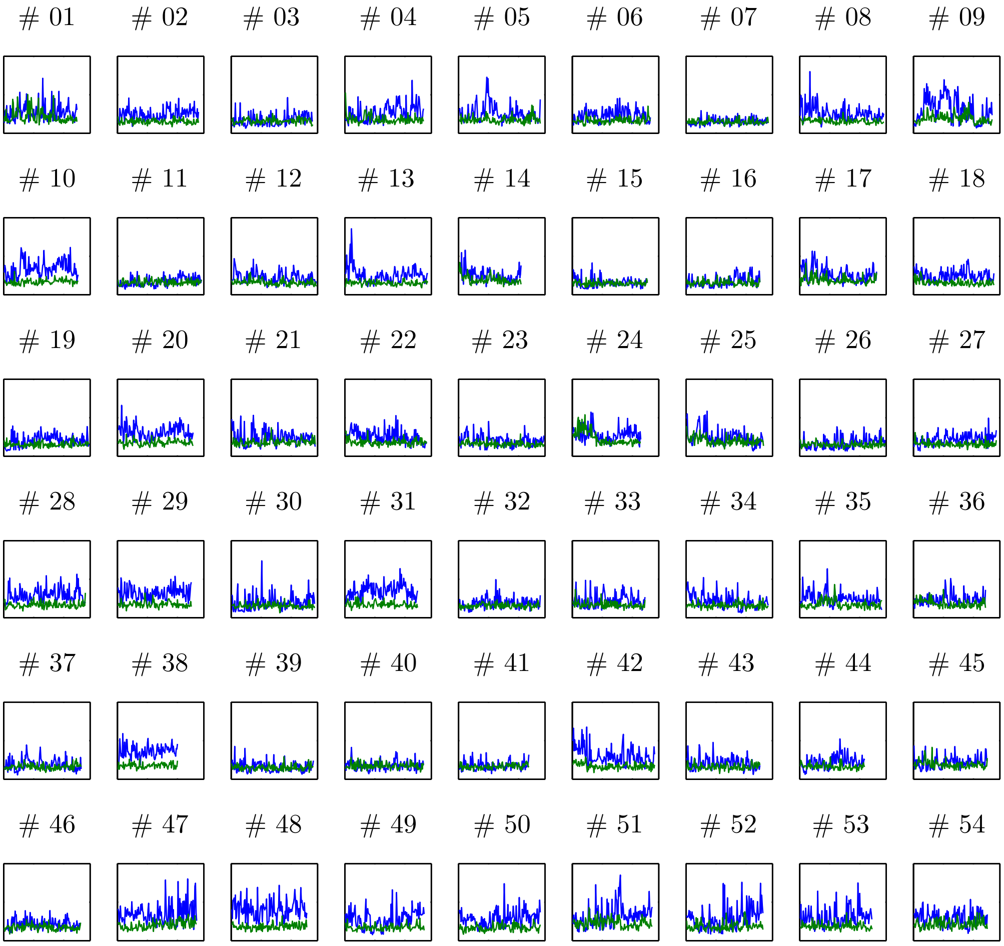


Figure 6.25: Estimated α_1 *blue lines* and α_2 *green lines* from 5min segments across the second 12hr recordings of the 54 subjects in nsr2db.

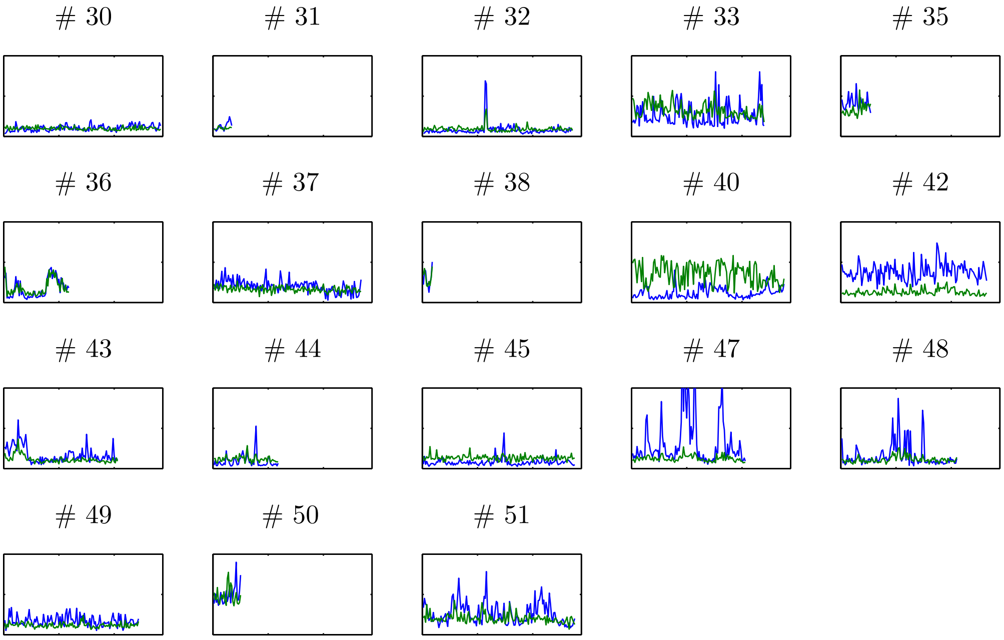


Figure 6.26: Estimated α_1 *blue lines* and α_2 *green lines* from 5min segments across the second 12hr recordings of 18 subjects in sddb.

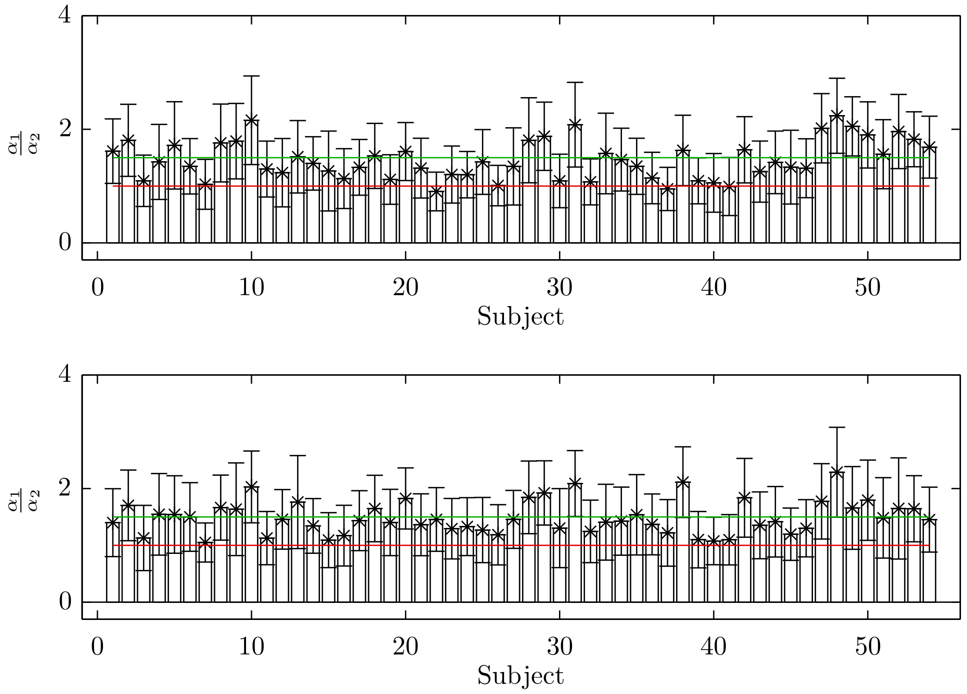


Figure 6.27: The $\frac{\alpha_1}{\alpha_2}$ of 54 subjects in nsr2db for the 1st 12hr (*top*) and 2nd 12hr (*bottom*) recordings. For each subject, the mean with one standard deviation is presented. The red and green lines denote 1 and 2, respectively.

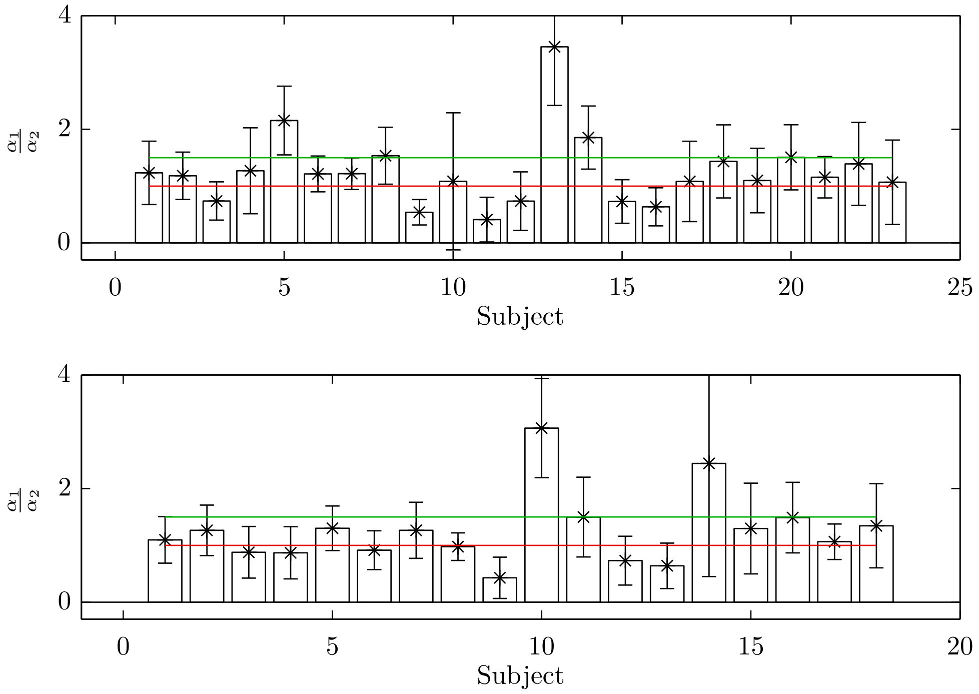


Figure 6.28: The $\frac{\alpha_1}{\alpha_2}$ of 23 subjects in sddb for the 1st 12hr (*top*) and 2nd 12hr (*bottom*, only 18 subjects have recordings within the 2nd 12hr) recordings. For each subject, the mean with one standard deviation is presented. The red and green lines denote 1 and 2, respectively.

Table 6.1: Age and Gender information about the subjects of the nsr2db dataset on Physionet.

Subject	Age	Gender	Subject	Age	Gender
1	64	F	28	65	M
2	67	M	29	63	M
3	67	F	30	70	F
4	62	F	31	67	M
5	62	F	32	68	M
6	64	M	33	65	M
7	76	M	34	67	M
8	64	F	35	66	M
9	66	M	36	60	F
10	61	F	37	63	M
11	65	F	38	62	M
12	66	M	39	70	M
13	63	F	40	63	F
14	65	F	41	64	F
15	74	M	42	68	F
16	73	F	43	66	M
17	71	F	44	65	F
18	68	M	45	67	F
19	65	F	46	63	F
20	58	F	47	28	M
21	59	M	48	38	M
22	68	M	49	39	M
23	66	F	50	29	M
24	63	F	51	40	M
25	75	M	52	39	M
26	72	M	53	35	M
27	64	M	54	35	M

grid-search method with a relatively sparse grid (20×20). As a result, the obtained frequency component estimates may be unreliable. For this reason, our results are mainly preliminary.

In practice, the additional computational overheads prohibit us from using very dense grids to deal with the quality concern. One promising approach could be to increase the number of regression basis components. In other words, instead of two frequency components, one could use a branch of frequency components within the LF and HF bands. To some extent, this is similar to the least square spectral method of Lomb-Scagle method, but with a different objective function. To this end, in order to capture the frequency components of SNS and PSNS, it should include as many frequency components as possible. However, increasing the number basis functions may lead the problem becoming ill-posed. In this regard, one can assume the parameters to be sparse. In fact, only two frequency components are of interests. The sparsity constrain can be easily added to the original the objective function, resulting a new problem similar to the ℓ_1 regularised logistic regression in Lee et al. (2006); Koh et al. (2007):

$$\min_{\boldsymbol{\theta} \in \Theta} - \log p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\theta}) + \gamma ||\boldsymbol{\theta}||_1, \tag{6.17}$$

where γ is the sparsity coefficient. This coefficient is normally determined by cross-validation. Additionally, since the historic R-waves also serve as parts of the basis functions in PPGLM, ℓ_1 regularisation effectively permits automatic selection of the history window length.

Finally, it is important to emphasise that the PPGLM framework can be easily extended to relate more features to the heartbeats. These features may include blood pressure and respiratory conditions among others. With the help from ℓ_1 regularisation, the resulting parameters may reveal regulatory networks of ANS activities and physiological conditions, and their effect on or contribution to heart diseases in patients.

Chapter 7

Conclusion and future work

7.1 Conclusion

This thesis demonstrates computational approaches to extracting discriminative features from event-based signals, such as neural spikes and heartbeats. These signals are concerned with discrete events in time, separated by seemingly random intervals. They are often driven by continuous processes relating to the organ's physiology, whose charge-and-fire type behaviour results in observed discrete events. The state-space model with point process observations (SSPP) proposed by Smith and Brown (2003), which avoids the somewhat artificial change to inter-event times, and handles directly the discrete events. The model parameters are highly interpretable while showing promises on discriminating signals under different experimental or physiological scenarios. Estimating these parameters from data is challenging, since the hidden states are also unknown. We discuss and develop a broad range of inference algorithms in this thesis to tackle this challenge.

For simultaneous estimation of states and parameters of SSPP model, Smith and Brown (2003) derived an approximate expectation-maximization (EM) algorithm. In chapter 3, we extend their framework in which the hidden states are treated as scalers to vector systems. To fully understand the estimation quality under the approximate EM setting, the model structure has been studied both theoretically and empirically, which led to the following findings: (i) The filtering and smoothing densities of the states of SSPP are log-conave, which allows a Laplace approximation based smoother to provide accurate estimations. We further identify several conditions that allow an arbitrary state-space model to have log-conave filtering and smoothing densities. This insight was found to be helpful in designing the inference strategy. (ii) The EM objective function, the $\mathcal{Q}$-function, when the summary statistics of the smoothing density are fixed, is skewed and concave in most of the parameters. This creates biases while the $\mathcal{Q}$-function being maximized. And in some extreme cases, this leads to multimodal landscapes with maxima far away from the truth. These results paved the way for more sophisticated Bayesian approaches to be applied for parameter estimation and inference from SSPP.

Based on the insights form EM for SSPP, we provide the variational methods for such Bayesian treatment. The methods construct approximate posterior distributions for both states and parameter. As a result, confident levels of the parameter estimation can be drawn, whereas in the EM setting only point estimates are available. We further derived an online variational methods for SSPP, similar to the dual extended Kalman filter (Wan and Nelson, 2001), yet in a Bayesian setting. The performances show good agreement with asymptotic exact methods such as Gibbs sampler and particle filter, in batch and sequential formulations. While the variational methods strike a decent balance between approximation accuracy and computational cost, they are still based on some unrealistic assumptions of variables being independent, which justifies a through exploration of Markov chain Monte Carlo (MCMC) methods in chapter 5.

Starting from a baseline single-site update Gibbs sampler, we showed the major challenges for MCMC methods for SSPP: (i) High correlation between states, and (ii) the lack conjugate prior to support direct sampling. To tackle these two challenges, we adopted several efficient schemes including the newly proposed methods, which exploit local geometries of the underlying distributions, and particle filter based method, that allow joint sampling between states and parameters. Quantitative efficiency comparison of these methods indicates that the geometry based MCMC shows the best performance for SSPP. We also explore possible approximate schemes sitting between functional approximation and sampling, particularly, embedding variational and Laplace's method into the sampling procedure.

In chapter 6, we demonstrate applications of the above framework in real neural spike train and heartbeat data analysis, using publicly available datasets. In particular, the posteriors of parameters show systematic difference between neural spikes recordings from rats that are stimulated with four taste chemicals. In addition, together with a dataset from monkey with various image contrast level stimuli, we investigate the how close are the variational approximation and MCMC samples, in terms of a fitting criteria. The results show that the approximation are generally similar, especially when more data is available. When less spikes are sparsely populated with a short period of time, MCMC shows superior performance. We further focus on human heartbeats, in which a shallow model is employed by switching off the state process. The model become a regression model relating two harmonic inputs representing para-sympathetic and sympathetic nerves systems, which calibrate the heart rate, and a window of history heartbeats, with heartbeats. Some early results show that the learned regression parameters capture variability between normal and patients suffered sudden cardiac death much better than traditional time-frequency domain features which are primarily used in clinical practise.

7.2 Future work

An immediate extension would be deriving Bayesian inference mechanisms for SSPP with vector states. This brings the need for model selective methods such as sparsity encouraging regularizations, Bayes factor or the reversible jump MCMC to strike balance between exploration and exploitation. Further, the power of efficient MCMC methods allow us to relax the restrictions introduced to the likelihood model of SSPP. For example, it could become state-space models with logistic likelihood model, which is more accurate on dealing with the firing probability. With these modeling tools, we could be in a better position to characterize more complicated experimental settings. Examples include, predicting the effect of multiple stimuli, and detecting unknown biological processes that are significant contributors to variation in the observed spiking activities.

Algorithmically, it is demanded to further cross-fertilize advanced MCMC methods for superior performance. Of particular interest is combining the Riemann manifold Langevin method and the particle MCMC, in which the geometric information can be used for efficiently guiding the particles. Additionally, to make these methods scalable for large-scale problems, it is desirable to parallelize these powerful yet computationally expensive MCMC methods. An example is the circularly-coupled scheme proposed by Neal (2002), in which the large number of samples produced by some traditional MCMC methods can be spilt into several coupled parallel chains. The coupling procedure allows the majority of the chains to automatically discard the burn-in period. Therefore the target distribution can be approached much rapidly. To this end, designing such type of schemes for the advanced MCMC methods could be an exciting line of research.

Another interesting direction could be the validation of approximate inference results. Comparing difference approximation of posterior is a challenge, since the truth is genuinely unknown. This makes ideal measures such as KL divergence difficult to compute. Normally, the results from MCMC methods are considered as ground truth. However, this is questionable, since the convergence of MCMC methods under finite sample number is poorly understood. For these reasons, it is of interest to develop effective validation tools for comparing different approximations inference results. In chapter 6, we tried a model fitting criteria to compare MCMC and variational methods. However, this is somehow contradictory to the essence of Bayesian inference, which tends to average over models, rather than fit them. Alternatively, Eaton (2011) provides a promising possibility with what he called “a conditional game” approach, where the outcome of the game can be used to selective better approximations.

Appendix A

Appendix for chapter 3

A.1 Proof for theroem 3.1

Let $\ell(\boldsymbol{\theta})$ be the log-likelihood function, and examine the three settings in theorem 3.1

- Consider the first scenario, the likelihood is equation (3.6) and $\lambda_k = \exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k)$.

  $$\ell(\boldsymbol{\theta}) = \sum_{k=1}^{K}(y_k(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k) - \exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k)). \tag{A.1}$$

  Denote $\mathbf{H}(\boldsymbol{\theta})$ as the Hessian matrix of $\ell(\boldsymbol{\theta})$.

  $$\mathbf{H}(\boldsymbol{\theta}) = \sum_{k=1}^{K} -\boldsymbol{\phi}_k \exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k)\boldsymbol{\phi}_k^{\mathrm{T}} \preceq 0. \tag{A.2}$$

  The negative-semidefinite Hessian matrix $\mathbf{H}(\boldsymbol{\theta})$ implies that $\ell(\boldsymbol{\theta})$ is concave in $\boldsymbol{\theta}$.

- Consider the second scenario, the likelihood is equation (3.7) and $\lambda_k\Delta = \sigma(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k)$, where $\sigma(a) = \frac{1}{1+\exp(-a)}$.

  $$\ell(\boldsymbol{\theta}) = \sum_{k=1}^{K}\left(y_k \ln(\lambda_k\Delta) + (1-y_k)\ln\left(1-\lambda_k\Delta\right)\right). \tag{A.3}$$

  The $\mathbf{H}(\boldsymbol{\theta})$ writes as,

  $$\mathbf{H}(\boldsymbol{\theta}) = \sum_{k=1}^{K} -\boldsymbol{\phi}_k\lambda_k(1-\lambda_k)\boldsymbol{\phi}_k^{\mathrm{T}} \preceq 0. \tag{A.4}$$

  Likewise, this establishes that ℓ is concave in $\boldsymbol{\theta}$ given $\boldsymbol{\phi}_k$ and vice versa.

- For the third case, the log-likelihood $\ell(\boldsymbol{\theta})$ becomes,

$$\ell(\boldsymbol{\theta}) = \sum_{k=1}^{K} (y_k(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k + \ln \Delta) - \exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k)\Delta). \tag{A.5}$$

  The only difference between equation (A.5) and (A.1) is the additional terms $\ln \Delta$ and Δ. These two terms are constant w.r.t to $\boldsymbol{\theta}$.

  The Hessian $\mathbf{H}(\boldsymbol{\theta})$ becomes,

$$\mathbf{H}(\boldsymbol{\theta}) = \sum_{k=1}^{K} -\boldsymbol{\phi}_k \exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_k)\boldsymbol{\phi}_k^{\mathrm{T}} \preceq 0. \tag{A.6}$$

  Then, the $\ell(\boldsymbol{\theta})$ is concave in $\boldsymbol{\theta}$ given $\boldsymbol{\phi}_k$.

Additional, given the fact that $\boldsymbol{\theta}$ and $\boldsymbol{\phi}_k$ is exchangeable, the proof for ℓ is concave in $\boldsymbol{\phi}_k$ follows the same. However, ℓ is not concave joint in $\boldsymbol{\theta}$ and $\boldsymbol{\phi}_k$. This completes the proof. □

Appendix B

Appendix for chapter 4

B.1 Derivation of the update equations for states

B.1.1 The forward pass

Initialise $x_{0|0}$ and set $\sigma^2_{0|0} = \kappa$ where κ is indicative of the uncertainty on the initial state. The forward pass is given by Beal (2003)

$$p(x_k|\mathcal{Y}_k) \propto \int dx_{k-1} p(x_{k-1}|\mathcal{Y}_{k-1}) \exp \langle \ln p(x_k|x_{k-1}, \boldsymbol{\theta}) p(\mathbf{y}_k|x_k, \boldsymbol{\theta}) \rangle,$$

where $p(x_k|x_{k-1}, \boldsymbol{\theta}) = \mathcal{N}_{x_k}(\rho x_{k-1} + \alpha I_k, \sigma^2_\epsilon)$ and $p(x_{k-1}|\mathcal{Y}_{k-1}) = \mathcal{N}_{x_{k-1}}(x_{k-1|k-1}, \sigma^2_{k-1|k-1})$. The product $p(x_{k-1}|\mathcal{Y}_{k-1}) \exp \langle \ln p(x_k|x_{k-1}, \boldsymbol{\theta}) \rangle$ is normal in x_{k-1} with precision $\overline{\sigma}^{-2}_{k-1} = \sigma^{-2}_{k-1} + \langle \rho^2 \rangle \sigma^{-2}_\epsilon$ and mean

$$\overline{x}_{k-1} = \overline{\sigma}^2_{k-1}(x_{k-1|k-1}\sigma^{-2}_{k-1|k-1} + \langle \rho \rangle x_k \sigma^{-2}_\epsilon - \langle \rho\alpha \rangle I_k \sigma^{-2}_\epsilon).$$

Marginalising out x_{k-1} we get

$$p(x_k|\mathcal{Y}_k) \propto \mathcal{N}_{x_k}(\tilde{x_k}, \tilde{\sigma}^2_k) \exp(\langle \ln p(\mathbf{y}_k|x_k, \boldsymbol{\theta}) \rangle),$$

where $\tilde{\sigma}^{-2}_k = \sigma^{-2}_\epsilon - \langle \rho \rangle^2 \overline{\sigma}^2_{k-1} \sigma^{-4}_\epsilon$ and

$$\tilde{x_k} = \tilde{\sigma}^2_k \left(\overline{\sigma}^2_{k-1} \langle \rho \rangle \sigma^{-2}_\epsilon [x_{k-1|k-1}\sigma^{-2}_{k-1|k-1} - \langle \rho\alpha \rangle I_k \sigma^{-2}_\epsilon] + \langle \alpha \rangle I_k \sigma^{-2}_\epsilon \right).$$

Since the observation equation is nonlinear we choose to approximate the product of the distributions to a Gaussian with Laplace's method so that

$$\mathcal{N}_{x_k}(\tilde{x_k}, \tilde{\sigma}^2_k) \exp(\langle \ln p(\mathbf{y}_k|x_k, \boldsymbol{\theta}) \rangle) \approx \mathcal{N}_{x_k}(x_{k|k}, \sigma^2_{k|k}),$$

where, we recall

$$p(\mathbf{y}_k|x_k,\boldsymbol{\theta}) = \prod_{c=1}^{C} \Delta \exp(\mu+\beta^c x_k)^{y_k^c} \exp(-\exp(\mu+\beta^c x_k)\Delta).$$

As shown in the main text, a nonlinear optimiser is needed to evaluate $x_{k|k}$.

B.1.2 The backward pass

Initialise with $\sigma^{*^2} = \kappa$ where κ is large and $x_K^* = x_{K|K}$ if carried out after the forward pass (see below). The backward pass is given by the recursion as in Beal (2003)

$$p(\mathbf{y}_{k+1:K}|x_k) = \int dx_{k+1} p(\mathbf{y}_{k+2:K}|x_{k+1}) \exp\langle \ln p(x_{k+1}|x_k,\boldsymbol{\theta}) p(\mathbf{y}_{k+1}|x_{k+1},\boldsymbol{\theta})\rangle,$$

where $p(x_{k+1}|x_k,\boldsymbol{\theta}) = \mathcal{N}_{x_{k+1}}(\rho x_k + \alpha I_{k+1}, \sigma_\epsilon^2)$ and $p(\mathbf{y}_{k+2:K}|x_{k+1}) = \mathcal{N}_{x_{k+1}}(x^*_{k+1}, \sigma^{*^2}_{k+1})$. We find $p(\mathbf{y}_{k+2:K}|x_{k+1})\exp(\langle \ln p(\mathbf{y}_{k+1}|x_{k+1},\boldsymbol{\theta})\rangle) \approx \mathcal{N}_{x_{k+1}}(x'_{k+1}, \sigma'^2_{k+1})$ by taking the quadratic Taylor expansion around an arbitrary $\hat{x}_{k+1}$ to obtain the expressions

$$\begin{aligned} x'_{k+1} &= \hat{x}_{k+1} + \sigma'^2_{k+1}\Bigg(\frac{x^*_{k+1}-\hat{x}_{k+1}}{\sigma^{*^2}_{k+1}} + \sum_{c=1}^{C}\Bigg\{\langle\beta^c\rangle_{q(\beta^c)} y^c_{k+1} \\ &\qquad - \Delta\langle\exp\mu\rangle_{q(\mu)}\frac{d}{dx_{k+1}}\,\left[\langle\exp x_{k+1}\beta^c\rangle_{q(\beta^c)}\right]\big|_{x_{k+1}=\hat{x}_{k+1}}\Bigg\}\Bigg), \\ \sigma'^2_{k+1} &= \left(\sigma^{*^{-2}}_{k+1} + \sum_{c=1}^{C}\left\{\Delta\langle\exp\mu\rangle_{q(\mu)}\frac{d^2}{dx^2_{k+1}}\,\left[\langle\exp x_{k+1}\beta^c\rangle_{q(\beta^c)}\right]\big|_{x_{k+1}=\hat{x}_{k+1}}\right\}\right)^{-1}. \end{aligned}$$

The choice of $\hat{x}_{k+1}$ bears a lot of weight on the performance of the algorithm. One can set $\hat{x}_{k+1} = x'_{k+1}$ resulting in a nonlinear optimisation problem. On the other hand, one can linearise around the filtered estimate $x_{k+1|k+1}$ instead and this is what is done in the main text. The advantage is that no nonlinear optimisation is required to compute the backward pass; the drawback is that the backward pass can no longer be carried out in parallel with the forward pass.

The next step is to find the product of this approximate distribution with $\exp(\langle\ln p(x_{k+1}|x_k,\boldsymbol{\theta})\rangle)$ which is easily shown to be proportional to

$$\begin{aligned} \exp\Bigg(&- x^2_{k+1}(\sigma_\epsilon^{-2} + \sigma'^{-2}_{k+1})/2 + x_{k+1}\left[\langle\rho\rangle x_k\sigma_\epsilon^{-2} + \langle\alpha\rangle I_{k+1}\sigma_\epsilon^{-2} + x'_{k+1}\sigma'^{-2}_{k+1}\right] \\ &- \langle\rho^2\rangle x_k^2\sigma_\epsilon^{-2}/2 - \langle\rho\alpha\rangle x_k I_{k+1}\sigma_\epsilon^{-2}\Bigg). \end{aligned}$$

The required normal distribution in x_k with mean x^*_k and variance $\sigma^{*^2}_k$ is found by marginalising out x_{k+1}. The smoothed estimate is computed by considering the product distribution of the forward pass and the backward pass. In particular we find that

$$p(x_k|\mathcal{Y}_K) \propto p(x_k|\mathcal{Y}_k)p(\mathbf{y}_{k+1:K}|x_k).$$

Since this is a product of Gaussian distributions the state estimate conditioned on all the data can be found and can be readily computed in the backward pass if this is carried out sequential to the forward pass. The results are shown in the main text. The pairwise marginals are given as

$$p(x_k, x_{k+1}|\mathcal{Y}_K) \propto p(x_k|\mathcal{Y}_K)p(\mathbf{y}_{k+2:K}|x_{k+1}) \exp(\langle \ln p(x_{k+1}|x_k, \boldsymbol{\theta})p(\mathbf{y}_{k+1}|x_{k+1}, \boldsymbol{\theta})\rangle).$$

We expand the logarithm of this quantity and approximate it to a multivariate normal distribution about the smoothed state estimate. The required second moment is then found by adding the product of the smoothed pair to the cross-covariance. The result is shown in the main text.

B.2 Derivation of the update equations for CIF parameters

Batch update of $p(\mu)$: The variational posterior over μ, ignoring terms independent of μ, is given by

$$\ln q(\mu) = \ln p(\mu) + \langle \sum_{c=1}^{C} \sum_{i=1}^{K} y_i^c[\mu + \beta^c x_i] - \exp(\mu)\exp(\beta^c x_i)\Delta \rangle,$$

where $p(\mu)$ is the prior over μ with mean μ_p and variance σ_p^2. We restrict the variational posterior to be Gaussian with mean $\hat{\mu}$ and variance σ_μ^2. By application of the standard Laplace's method we obtain the expressions given in the main text. In these expressions it is required to evaluate the quantity $\langle \exp(x_i\beta^c)\rangle$. From moment generating functions we know that

$$\int \exp(x_i\beta^c)\mathcal{N}_{x_i}(x_{i|K}, \sigma_{i|K}^2)dx_i = \exp(x_{i|K}\beta^c + \sigma_{i|K}^2\beta^{c^2}/2).$$

However, we are concerned with the quantity

$$\begin{aligned}
\langle \exp(x_i\beta^c)\rangle &= \int dx_i \left[\int d\beta^c \mathcal{N}_{\beta^c}(\hat{\beta}^c, \sigma_{\beta^c}^2) \right] \mathcal{N}_{x_i}(x_{i|K}, \sigma_{i|K}^2) \\
&= \int dx_i \exp(\hat{\beta}^c x_i + \sigma_{\beta^c}^2 x_i^2/2)\mathcal{N}_{x_i}(x_{i|K}, \sigma_{i|K}^2) \\
&= \frac{1}{\sqrt{2\pi\sigma_{i|K}^2}} \int dx_i \exp(\hat{\beta}^c x_i + \sigma_{\beta^c}^2 x_i^2/2 - (x_i - x_{i|K})^2/2\sigma_{i|K}^2).
\end{aligned}$$

After marginalising out x_i and some algebraic manipulation the final result is obtained as

$$\langle \exp(\beta^c x_i) \rangle_{q(\mathcal{X}_K)q(\beta^c)} = \sqrt{\frac{1}{1 - \sigma^2_{\beta^c}\sigma^2_{i|K}}} \exp\left(\frac{x^2_{i|K}\sigma^2_{\beta^c} + \hat{\beta^c}^2 \sigma^2_{i|K} + 2\hat{\beta}^c x_{i|K}}{2(1 - \sigma^2_{\beta^c}\sigma^2_{i|K})} \right).$$

Batch update of $p(\beta^c)$: The variational posterior over β^c, ignoring terms independent of β^c, is given by

$$\ln q(\beta^c) = \ln p(\beta^c) + \langle \sum_{i=1}^{K} y_i^c [\mu + \beta^c x_i] - \exp(\mu) \exp(\beta^c x_i) \Delta \rangle,$$

where $p(\beta^c)$ denotes the prior over β^c. Effecting the required derivatives we once again restrict the variational posterior to be Gaussian with mean and variance as given in the main text. The expectations required in this case are those of log normal distributions which are easy to compute. In particular we have that

$$\langle \exp(\beta^c x_i) \rangle_{q(\mathcal{X}_K)} = \exp(\beta^c x_{i|K} + {\beta^c}^2 \sigma^2_{i|K}/2),$$

and $\langle \exp(\mu) \rangle_{q(\mu)} = \exp(\hat{\mu} + \sigma^2_\mu/2)$.

Appendix C

Appendix for chapter 5

C.1 The gradient of the joint likelihood

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \gamma} = -\rho + \frac{\rho x_0^2(1-\rho^2)}{\sigma_\varepsilon^2} + \frac{(1-\rho^2)}{\sigma_\varepsilon^2}\sum_{k=1}^{K} x_{k-1}(x_k - \rho x_{k-1} - \alpha I_k), \tag{C.1}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \alpha} = \frac{1}{\sigma_\varepsilon^2}\sum_{k=1}^{K}(x_k - \rho x_{k-1} - \alpha I_k) I_k, \tag{C.2}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \mu} = \sum_{k=1}^{K}\sum_{c=1}^{C}\left(y_{c,k} - \exp(\mu + \beta_c x_k)\Delta\right). \tag{C.3}$$

C.2 The derivative of metric tensor

The derivative of the metric tensor has the general form as,

$$\nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta}) = \begin{bmatrix} \nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta})_{1,1} & \nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta})_{1,2} & \nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta})_{1,3} \\ \nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta})_{2,1} & \nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta})_{2,2} & \nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta})_{2,3} \\ \nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta})_{3,1} & \nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta})_{3,2} & \nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta})_{3,3} \end{bmatrix}, \tag{C.4}$$

where $\nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta})_{1,3}$ and $\nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta})_{3,1}$ are always 0. In the following subsections, only the nonzero elements of $\nabla_{\boldsymbol{\theta}}\mathbf{G}(\boldsymbol{\theta})$ are stated.

In addition, we have the derivative of $\mathbb{E}_p\left[x_{k-1}\right]$ and $\mathrm{Var}_p\left[x_{k-1}\right]$ w.r.t. ρ,

$$\frac{\partial \mathbb{E}_p\left[x_{k-1}\right]}{\partial \rho} = \alpha(I_{k-1} + 2\rho I_{k-1} + \cdots + (K-1)\rho^{K-2} I_1), \tag{C.5}$$

$$\frac{\partial \mathrm{Var}_p\left[x_{k-1}\right]}{\partial \rho} = -\frac{2\rho\sigma_\varepsilon^2}{(1-\rho^2)^2}. \tag{C.6}$$

C.2.1 The derivative of metric tensor w.r.t. γ

$$\nabla_\gamma \mathbf{G}(\boldsymbol{\theta})_{1,1} = (1-\rho^2)\Bigg(4\rho - 2K\rho - \frac{4\rho(1-\rho^2)}{\sigma_\varepsilon^2}\sum_{k=1}^{K}\mathbb{E}_p[x_{k-1}^2] \tag{C.7}$$

$$+\frac{(1-\rho^2)^2}{\sigma_\varepsilon^2}\sum_{k=1}^{K} 2\mathbb{E}_p\left[x_{k-1}\right]\frac{\partial \mathbb{E}_p\left[x_{k-1}\right]}{\partial \rho}\Bigg), \tag{C.8}$$

$$\nabla_\gamma \mathbf{G}(\boldsymbol{\theta})_{1,2} = \nabla_{\boldsymbol{\theta}} \mathbf{G}(\boldsymbol{\theta})_{2,1} \tag{C.9}$$

$$= (1-\rho^2)\left(-\frac{2\rho}{\sigma_\varepsilon^2}\sum_{k=1}^{K}\mathbb{E}_p[x_{k-1}]I_k + \frac{(1-\rho^2)}{\sigma_\varepsilon^2}\sum \frac{\partial \mathbb{E}_p\left[x_{k-1}\right]}{\partial \rho} I_k\right), \tag{C.10}$$

$$\nabla_\gamma \mathbf{G}(\boldsymbol{\theta})_{3,3} = (1-\rho^2)\Delta\sum_{k=1}^{K}\sum_{c=1}^{C}\Bigg(\exp\left(\mu + \beta_c \mathbb{E}_p\left[x_k\right] + \frac{\beta_c^2}{2}\mathrm{Var}_p\left[x_k\right]\right) \tag{C.11}$$

$$\times\left(\beta_c \frac{\partial \mathbb{E}_p\left[x_k\right]}{\partial \rho} + \frac{\beta_c^2}{2}\frac{\partial \mathrm{Var}_p\left[x_k\right]}{\partial \rho}\right)\Bigg). \tag{C.12}$$

C.2.2 The derivative of metric tensor w.r.t. α

$$\nabla_\alpha \mathbf{G}(\boldsymbol{\theta})_{1,1} = \frac{(1-\rho^2)^2}{\sigma_\varepsilon^2}\sum_{k=1}^{K} 2\mathbb{E}_p\left[x_{k-1}\right]\frac{\partial \mathbb{E}_p\left[x_{k-1}\right]}{\partial \alpha}, \tag{C.13}$$

$$\nabla_\alpha \mathbf{G}(\boldsymbol{\theta})_{1,2} = \nabla_\alpha \mathbf{G}(\boldsymbol{\theta})_{2,1} \tag{C.14}$$

$$= \frac{(1-\rho^2)}{\sigma_\varepsilon^2}\sum_{k=1}^{K}\frac{\partial \mathbb{E}_p\left[x_{k-1}\right]}{\partial \alpha} I_k, \tag{C.15}$$

$$\nabla_\alpha \mathbf{G}(\boldsymbol{\theta})_{3,3} = \Delta\sum_{k=1}^{K}\sum_{c=1}^{C}\left(\exp\left(\mu + \beta_c \mathbb{E}_p\left[x_k\right] + \frac{\beta_c^2}{2}\mathrm{Var}_p\left[x_k\right]\right)\beta_c\frac{\partial \mathbb{E}_p\left[x_k\right]}{\partial \alpha}\right). \tag{C.16}$$

C.2.3 The derivative of metric tensor w.r.t. μ

$$\nabla_\mu \mathbf{G}(\boldsymbol{\theta})_{3,3} = \sum_{k=1}^{K}\sum_{c=1}^{C}\left(\exp\left(\mu + \beta_c \mathbb{E}_p\left[x_k\right] + \frac{\beta_c^2}{2}\mathrm{Var}_p\left[x_k\right]\right)\Delta\right). \tag{C.17}$$

References

E. D. Adrian. *The basis of sensation.* WW Norton & Co, 1928.

Y. Ahmadian, J. W. Pillow, and L. Paninski. Efficient Markov chain Monte Carlo methods for decoding neural spike trains. *Neural Computation*, 23(1):46–96, 2011.

S. Akselrod, D. Gordon, F. A. Ubel, D. C. Shannon, A. C. Berger, and R. J. Cohen. Power spectrum analysis of heart rate fluctuation: a quantitative probe of beat-to-beat cardiovascular control. *Science*, 213(4504):220–222, 1981.

S. Amari and H. Nagaoka. *Methods of Information Geometry.* Oxford University Press, Oxford, 2000.

B. D. O. Anderson and J. B. Moore. *Optimal filtering.* Prentice-hall, 1979.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72:269–342, 2010.

M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.

H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 21–30, 1999.

Y. Bar-Shalom and Fortmann T. E. *Tracking and data association.* Academic Press Professional, Inc., 1987.

R. Barbieri and E. N. Brown. Analysis of heartbeat dynamics by point process adaptive filtering. *Biomedical Engineering, IEEE Transactions on*, 53(1):4–12, 2006.

R. Barbieri, E. C. Matten, A. A. Alabi, and E. N. Brown. A point-process model of human heartbeat intervals: new definitions of heart rate and heart rate variability. *Am J Physiol Heart Circ Physiol*, 288(1):H424–435, 2005.

M. J. Beal. *Variational algorithms for approximate Bayesian inference.* PhD thesis, University College London, London, UK, 2003.

M. J. Beal, F. Falciani, Z. Ghahramani, C. Rangel, and D. L. Wild. A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 21 (3):349–356, 2005.

W. Bialek, F. Rieke, R. R. de Ruyter van Steveninck, and D. Warland. Reading a neural code. *Science*, 252(5014):1854–1857, 1991.

P. F. Binkley, E. Nunziata, G. J. Haas, S. D. Nelson, and R. J. Cody. Parasympathetic withdrawal is an integral component of autonomic imbalance in congestive heart failure: demonstration in human subjects and verification in a paced canine model of ventricular failure. *J Am Coll Cardiol*, 18(2):464–472, 1991.

C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer-Verlag, New York, 2006.

R. Boel and V. Benes. Recursive nonlinear estimation of a diffusion acting as the rate of an observed poisson process. *IEEE Transactions on Information Theory*, 26(5): 561–575, 1980.

S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(2):113–120, 1979.

S. P. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.

D. R. Brillinger. Nerve cell spike train data analysis: a progression of techniques. *J. Amer. Stat. Assoc.*, 87:260–271, 1992.

E. N. Brown, R. Barbieri, U. T. Eden, and L. M. Frank. Likelihood methods for neural data analysis. In J. Feng, editor, *Computational Neuroscience: A Comprehensive Approach*, pages 253–286. CRC, London, 2003.

E. N. Brown, R. Barbieri, V. Ventura, R. E. Kass, and L. M. Frank. The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation*, 14(2):325–346, 2002.

E. N. Brown, L. M. Frank, D. Tang, M. C. Quirk, and M. A. Wilson. A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *The Journal of Neuroscience*, 18(18): 7411–7425, 1998.

E. N. Brown, R. E. Kass, and P. P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, 7(5):456–461, 2004.

E. N. Brown, D. P. Nguyen, L. M. Frank, M. A. Wilson, and V. Solo. An analysis of neural receptive field plasticity by point process adaptive filtering. *Proceedings of the National Academy of Sciences*, 98(21):12261–12266, 2001.

C. Campbell and S. Godsill. On a new stochastic version of the EM algorithm. In *Proceedings of the EUSIPCO 1998*, 1998.

C. K. Carter and R. Kohn. On Gibbs sampling for state space models. *Biometrika*, 81 (3):541–553, 1994.

G. Casella and C. P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83 (1):81–94, 1996.

L. Chen, Z. Qin, and J. S. Liu. Exploring hybrid Monte Carlo in Bayesian computation. In *Proceedings of the ISBA 2000*, 2001.

Z. Chen, F. Kloosterman, M. A. Wilson, and E. N. Brown. Variational Bayesian inference for point process generalized linear models in neural spike trains analysis. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 2086 –2089, march 2010.

B. L. P. Cheung, B. A. Riedner, G. Tononi, and B. Van Veen. Estimation of cortical connectivity from eeg using state-space models. *Biomedical Engineering, IEEE Transactions on*, 57(9):2122–2134, 2010.

S. Chib, F. Nardari, and N. Shephard. Markov chain monte carlo methods for stochastic volatility models. *Journal of Econometrics*, 108(2):281–316, 2002.

T. C. Clapp and S. J. Godsill. Fixed-lag smoothing using sequential importance sampling. In *Bayesian Statistics 6: Proceedings of the Sixth Valencia International Meeting*, volume 6, pages 743–752, 1999.

T. M. Cover and J. A. Thomas. Determinant inequalities via information theory. *SIAM Journal on Matrix Analysis and Applications*, 9(3):384–392, 1988.

T. M. Cover and J. A. Thomas. *Elements of information theory*, volume 6. Wiley Online Library, 1991.

D. R. Cox and V. Isham. *Point Processes*, volume 12. Chapman & Hall/CRC, 1980.

G. Czanner, U. T. Eden, S. Wirth, M. Yanike, W. A. Suzuki, and E. N. Brown. Analysis of between-trial and within-trial neural spiking dynamics. *Journal of Neurophysiology*, 99(5):2672–2693, May 2008.

D. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Process,*. Springer Verlag, New York, 2nd edition, 2003.

P. Dayan and L. F. Abbott. *Theoretical neuroscience: Computational and mathematical modeling of neural systems.* MIT Press, Cambridge, 2001.

J. F. G. de Freitas, M. Niranjan A. H. Gee, and A. Doucet. Sequential Monte Carlo methods to train neural network models. *Neural Computation*, 12(4):955–993, 2000.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

M. Dewar, V. Kadirkamanathan, M. Opper, and G. Sanguinetti. Parameter estimation and inference for stochastic reaction-diffusion systems: application to morphogenesis in D. melanogaster. *BMC Systems Biology*, 4(1):21, 2010.

P. M. Di Lorenzo and J. D. Victor. Taste response variability and temporal coding in the nucleus of the solitary tract of the rat. *Journal of Neurophysiology*, 90:1418–1431, 2003.

A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice.* Springer-Verlag, New York, 2001.

A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filters for dynamic Bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 176–183, 2000a.

A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000b.

C. M Dougherty and R. L Burr. Comparison of heart rate variability in survivors and nonsurvivors of sudden cardiac arrest. *Am J Cardiol*, 70:441–8, 1992.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216 – 222, 1987.

J. Durbin and S. J. Koopman. *Time series analysis by state space methods.* Oxford University Press, 2001.

F. Eaton. A conditional game for comparing approximations. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11), Ft. Lauderdale, FL, USA*, 2011.

U. T. Eden, L. M. Frank, R. Barbieri, V. Solo, and E. N. Brown. Dynamic analysis of neural encoding by point process adaptive filtering. *Neural Computation*, 16(5): 971–998, 2004.

A. Ergün, R. Barbieri, U. T. Eden, M. A. Wilson, and E. N. Brown. Construction of point process adaptive filter algorithms for neural systems using sequential Monte carlo methods. *IEEE Transactions on Biomedical Engineering*, 54(3):419–428, 2007.

D. J. Ewing, J. M. Neilson, and P. Travis. New method for assessing cardiac parasympathetic activity using 24 hour electrocardiograms. *Br Heart J*, 55:396–402, 1984.

L. Fahrmeir and G. Tutz. Dynamic stochastic models for time-dependent ordered paired comparison systems. *Journal of the American Statistical Association*, 89(428):1438–1449, 1994.

P. Fearnhead. MCMC for state space models. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*. CRC, Boca Raton, 2010.

R. Freeman, V. Lirofonis, W. B. Farquhar, and M. Risk. Spectral analysis of heart rate in diabetic autonomic neuropathy. *Arch Neurol*, 48:185–190, 1991.

K. Friston, J. Mattout, N. Trujillo-Barreto, J. Ashburner, and W. Penny. Variational free energy and the laplace approximation. *NeuroImage*, 34:220–234, 2007.

A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409, 1990.

A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis*. CRC press, 2004.

A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science*, 7(4):457–472, 1992.

I. M. Gelfand and S. V. Fomin. *Calculus of variations*. Prentice Hall, 1963.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):721–741, Nov. 1984.

A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *The Journal of Neuroscience*, 2(11):1527–1537, 1982.

A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.

J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica: Journal of the Econometric Society*, pages 1317–1339, 1989.

J. Geweke and H. Tanizaki. Bayesian estimation of state-space models using the Metropolis-Hasting algorithm within Gibbs sampling. *Computational Statistics and Data Analysis*, 37(2):151 – 170, 2001.

C. J. Geyer. Introduction to Markov Chain Monte Carlo. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*. CRC, Boca Raton, 2010.

W. R. Gilks and C. Berzuini. Following a moving target–Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146, 2001.

M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73:123–214, 2011.

A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

A. L. Goldberger, L. A. N. Amaral, J. M. Hausdorff, P. Ch. Ivanov, C.-K. Peng, and H. E. Stanley. Fractal dynamics in physiology: Alterations with disease and aging. *Proceedings of the National Academy of Sciences of the United States of America*, 99 (Suppl 1):2466–2472, 2002.

N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *Radar and Signal Processing, IEE Proceedings F*, volume 140, pages 107–113, 1993.

P. J. Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 149–192, 1984.

A. C. Guyton and J. E. Hall. *Textbook of medical physiology.* Saunders Philadelphia, PA, 8th edition, 1991.

W. K. Hastings. Monte carlo sampling methods using markov chain and their applications. *Biometrika*, 57:97–109, 1970.

S. Haykin. *Adaptive Filter Theory.* Prentice-Hall, 4th edition, 2002.

A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117 (4):500–544, 1952.

G. R. Holt, W. I. R. Softky, C. Koch, and R. J. Douglas. Comparison of discharge variability in vitro and in vivo in cat visual cortex neurons. *Journal of Neurophysiology*, 75(5):1806–1814, 1996.

E. H. Hon and S. T. Lee. The fetal electrocardiogram. I. the electrocardiogram of the dying fetus. *Am J Obstet Gyecol*, 87:804–813, 1963.

P. Ch. Ivanov, M. G. Rosenblum, C.-K. Peng, J. Mietus, S. Havlin, H. E. Stanley, and A. L. Goldberger. Scaling behaviour of heartbeat intervals obtained by wavelet-based time-series analysis. *Nature*, 383:323–327, 1996.

M. R Jarvis and P. P. Mitra. Sampling properties of the spectrum and coherency in sequences of action potentials. *Neural Computation*, 13, 2001.

J. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, 1906.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

T. Kailath, A. H. Sayed, and B. Hassibi. Linear estimation. *Upper Saddle River, NJ*, 2000.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. Am. Soc. Mech. Eng., Series D, Journal of Basic Engineering*, 82:35–45, 1960.

R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Trans. Am. Soc. Mech. Eng., Series D, Journal of Basic Engineering*, 83:95–108, 1961.

R. E. Kass. The geometry of asymptotic inference. *Statistical Science*, 4:188–234, 1989.

R. E. Kass, V. Ventura, and E. N. Brown. Statistical issues in the analysis of neuronal data. *Journal of Neurophysiology*, 94(1):8–25, 2005.

S. Kim, N. Shephard, and S. Chib. Stochastic volatility: Likelihood inference and comparison with ARCH models. *The Review of Economic Studies*, 65(3):361–393, 1998.

G. Kitagawa. Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American statistical association*, pages 1032–1041, 1987.

G. Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, pages 1–25, 1996.

G. Kitagawa. A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215, 1998.

G. Kitagawa and W. Gersch. *Smoothness Priors Analysis of Time Series.* Springer-Verlag, New York, 1996.

K. Koh, S. J. Kim, and S. Boyd. An interior-point method for large-scale ℓ_1-regularized logistic regression. *Journal of Machine learning research*, 8(8):1519–1555, 2007.

S. Koyama, L. Castellanos Pérez-Bolde, C. R. Shalizi, and R. E. Kass. Approximate methods for state-space models. *Journal of the American Statistical Association*, 105 (489):170–180, 2010.

T. Kreuz, D. Chicharro, M. Greschner, and R. G. Andrzejak. Time-resolved and time-scale adaptive measures of spike train synchrony. *Journal of neuroscience methods*, 195(1):92–106, 2011.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

S. I. Lee, H. Lee, P. Abbeel, and A. Y. Ng. Efficient l_1 regularized logistic regression. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-06)*, 2006.

J. S. Liu. The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem. *Journal of the American Statistical Association*, 89 (427):958–966, 1994.

J. S. Liu. *Monte Carlo Strategies in Scientific Computing.* Springer-Verlag, New York, 2001.

J. S. Liu and R. Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, pages 1032–1044, 1998.

L. Ljung. *System identification.* Wiley Online Library, 1999.

D. J. MacGregor, C. K. I. Williams, and G. Leng. A new method of spike modelling and interval analysis. *Journal of Neuroscience Methods*, 176(1):45–56, 2009.

D. J. C. Mackay. *Information Theory, Inference, and Leaning Algorithms.* Cambridge University Press, Cambridge, 2003.

M. Malik, J. T. Bigger, A. J. Camm, R. E. Kleiger, A. Malliani, A. J. Moss, and P. J. Schwartz. Heart rate variability: Standards of measurement, physiological interpretation, and clinical use. *Eur Heart J*, 17(3):354–381, 1996.

J. Manton, V. Krishnamurthy, and R. Elliott. Discrete time filters for doubly stochastic poisson processes and other exponential noise models. *nternational Journal of Adaptive Control and Signal Processing*, 13(5):393–416, 1999.

P. McCullagh and J. A. Nelder. *Generalized linear models.* Chapman & Hall/CRC, 1989.

G. J. McLachlan and T. Krishnan. *The EM algorithm and extensions*, volume 274. Wiley New York, 1997.

N. Metropolis, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machine. *Journal of Chemical Physics*, 21:1087–1091, 1953.

T. P. Minka. From hidden Markov models to linear dynamical systems. Technical report, `http://vismod.media.mit.edu/tech-reports/TR-531.pdf`, 1999.

G. B. Moody. Spectral analysis of heart rate without resampling. In *Computers in Cardiology 1993, Proceedings.*, pages 715 –718, 1993.

D. W. Moran and A. B. Schwartz. Motor cortical activity during drawing movements: population representation during spiral tracing. *Journal of neurophysiology*, 82(5): 2693–2704, 1999.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical report, CRG-TR-93-1. Department of Computer Science, University of Toronto, 1993.

R. M. Neal. Circularly-coupled Markov chain sampling. Technical report, Technical Report No. 9910 (revised), Department of Statistics, University of Toronto, 2002.

R. M. Neal. MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*. CRC, Boca Raton, 2010.

R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse and other variants. In M. I. Jordan, editor, *Learning in graphical models*, pages 355–368. Kluwer, Dordrecht, 1998.

M. Okatan, M. A. Wilson, and E. N. Brown. Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. *Neural Computation*, 17(9):1927–1961, 2005.

L. Paninski. Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15(4):243–262, 2004.

G. Parisi. *Statistical field theory*. Addison Wesley Publishing Company, 1988.

D. B. Percival and A. T. Walden. *Wavelet Methods for Time series analysis*. Cambridge University Press, Cambridge, 2002.

M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94:590–599, 1999.

A. Prékopa. On logarithmic concave measures and functions. *Acta Scientiarum Mathematicarum*, 34:335–343, 1973.

J. A. Quinn, C. K. I. Williams, and N. McIntosh. Factorial switching linear dynamical systems applied to physiological condition monitoring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1537–1551, 2008.

C. R. Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bull. Calc. Math. Soc.*, 37:81–91, 1945.

H.E. Rauch, F. Tung, and CT Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

A. Riehle, S. Grün, M. Diesmann, and A. Aertsen. Spike synchronization and rate modulation differentially involved in motor cortical function. *Science*, 278:1950–1953, 1997.

F. Rieke, D. Warland, R. R. de Ruyter van Steveninck, and W. Bialek. *Spike: Exploring the Neural Code.* MIT Press, Cambridge, 1997.

C. P. Robert and G. Casella. *Monte Carlo statistical methods.* Springer Verlag, 2004.

A. V. I. Rosti and M. J. F. Gales. Rao-blackwellised gibbs sampling for switching linear dynamical systems. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 1, pages I – 809–12 vol.1, 2004.

A. T. Roussin, J. D. Victor, J.-Y. Chen, and P. M. Di Lorenzo. Variability in responses and temporal coding of tastants of similar quality in the nucleus of the solitary tract of the rat. *J Neurophysiol*, 99:644–655, 2008.

S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345, 1999.

S. Ruiz, P. Crespo, and R. Romo. Representation of moving tactile stimuli in the somatic sensory cortex of awake monkeys. *Journal of neurophysiology*, 73(2):525–537, 1995.

Y. Salimpour, H. Soltanian-Zadeh, S. Salehi, N. Emadi, and M. Abouzari. Neuronal spike train analysis in likelihood space. *PLoS ONE*, 6(6):e21256, 06 2011.

S. Sanei and J. Chambers. *EEG signal processing.* Wiley-Interscience, 2007.

G. Sanguinetti, N. D. Lawrence, and M. Rattray. Probabilistic inference of transcription factor concentrations and gene-specific regulatory activities. *Bioinformatics*, 22(22): 2775–2781, 2006.

M. Sato. On-line model selection based on the variational Bayes. *Neural Computation*, 13(7):1649–1681, 2001.

L. K. Saul, T. Jaakkola, and M. I. Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996.

A. B. Schwartz, D. W. Moran, and G. A. Reina. Differential representation of perception and action in the frontal cortex. *Science*, 303(5656):380–383, 2004.

Steven L. Scott. Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 97(457):337–351, 2002.

A. Segall, M. H. A. Davis, and T. Kailath. Nonlinear filtering with counting observations. *IEEE Transactions on Information Theory*, 21(2):143–149, 1975.

N. Shephard and M. K. Pitt. Likelihood analysis of non-Gaussian measurement time series. *Biometrika*, 84(3):653–667, 1997.

A. C. Smith and E. N. Brown. Estimating a state-space model from point process observations. *Neural Computation*, 15(5):965–991, 2003.

D. L. Snyder and M. I. Miller. *Random Point Processes in Time and Space.* Springer Verlag, New York, 1991.

V. Solo. 'Unobserved' Monte Carlo method for identification of partially observed nonlinear state space systems. Part II: Counting process observations. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 4, pages 3331–3336, 2000.

G. B. Stanley, F. F. Li, and Y. Dan. Reconstruction of natural scenes from ensemble responses in the lateral geniculate nucleus. *The Journal of Neuroscience*, 19(18): 8036–8042, 1999.

G. Storvik. Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing*, 50(2):281–289, 2002.

W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of Neurophysiology*, 93(2):1074–1089, 2005.

R. E. Turner and M. Sahani. Two problems with variational expectation maximisation for time-series models. In D. Barber, A. T. Cemgil, and S. Chiappa, editors, *Inference and Learning in Dynamic Models.* Cambridge University Press, 2010.

J. D. Victor, E. M. Blessing, J. D. Forte, P. Buzás, and P. R. Martin. Response variability of marmoset parvocellular neurons. *The Journal of Physiology*, 579(1):29–51, 2007.

V. Šmídl and A. Quinn. The restricted variational Bayes approximation in Bayesian filtering. In *Proceedings of the IEEE Nonlinear Statistical Signal Processing Workshop*, pages 224–227, 2006.

E. A. Wan and A. T. Nelson. Dual extended Kalman filter methods. In S. Haykin, editor, *Kalman Filtering and Neural Networks*, pages 123–173, New York, USA, Oct. 2001. John Wiley & Sons, Inc.

Y. Wang, A. R. C. Paiva, J. C. Príncipe, and J. C. Sanchez. Sequential Monte Carlo point-process estimation of kinematics from neural spiking activity for brain-machine interfaces. *Neural Computation*, 21(10):2894–2930, 2009.

D. K. Warland, P. Reinagel, and M. Meister. Decoding visual information from a population of retinal ganglion cells. *Journal of Neurophysiology*, 78(5):2336–2350, 1997.

C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

K. Yuan, M. Girolami, and M. Niranjan. Markov chain Monte Carlo methods for state-space models with point process observations. *Neural Computation*, 24(6):1462–1486, 2012.

K. Yuan and M. Niranjan. Estimating a state-space model from point process observations: A note on convergence. *Neural Computation*, 22(8):1993–2001, 2010.

A. Zammit Mangion, G. Sanguinetti, and V. Kadirkamanathan. A Variational Approach for the Online Dual Estimation of Spatiotemporal Systems Governed by the IDE. In *Proceedings of the IFAC World Congress 2011*, 2011a.

A. Zammit Mangion, K. Yuan, V. Kadirkamanathan, M. Niranjan, and G. Sanguinetti. Online variational inference for state-space models with point process observations. *Neural Computation*, 23(8):1967–1999, 2011b.

K. Zhang, I. Ginzburg, B. L. McNaughton, and T. J. Sejnowski. Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells. *Journal of Neurophysiology*, 79(2):1017–1044, 1998.