CAMBRIDGE UNIVERSITY
ENGINEERING DEPARTMENT

SEQUENTIAL BAYESIAN ESTIMATION AND MODEL SELECTION APPLIED TO NEURAL NETWORKS

C Andrieu[1], JFG de Freitas and A Doucet

CUED/F-INFENG/TR 341

Second version: March 1999

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

E-mail: {ca226,jfgf,ad2}@eng.cam.ac.uk

[1] Authorship based on alphabetical order

Abstract

In this paper, we address the complex problem of sequential Bayesian estimation and model selection. This problem does not usually admit any type of closed-form analytical solutions and, as a result, one has to resort to numerical methods. We propose here an original sequential simulation-based strategy to perform the necessary computations. It combines sequential importance sampling, a selection procedure and reversible jump MCMC moves. We demonstrate the effectiveness of the method by applying it to radial basis function networks.

Contents

1 Introduction

Model selection is a fundamental data analysis task that arises in many fields of science and engineering. Research in this area indicates that the Bayesian paradigm is a natural and powerful approach for dealing with this problem. However, even within this framework, model selection is a difficult problem as it typically involves the evaluation of posterior model probabilities, which, in turn, require the computation of high-dimensional integrals with no closed-form analytical expression. These computational problems have limited the use of Bayesian model selection, except in cases where asymptotic expansions of the Bayes factors are valid (Bernardo and Smith 1994, Djurić 1998).

Recently, a set of powerful stochastic algorithms, known as Markov chain Monte Carlo (MCMC) methods, have enabled us to surmount most of the Bayesian computational problems when the data are available in batches (Gilks, Richardson and Spiegelhalter 1996, Green 1995, Tierney 1994). However, we address here the more challenging problem of sequential Bayesian model selection, that is, we are interested in performing Bayesian model selection recursively in time as the data arrive. In this scenario, MCMC methods are not applicable. Fortunately, Monte Carlo (MC) methods that perform sequential Bayesian estimation have also been developed recently (Berzuini, Best, Gilks and Larizza 1997, de Freitas, Niranjan, Gee and Doucet 2000, Doucet 1998, Liu and Chen 1998), following the seminal article of (Gordon, Salmond and Smith 1993). These methods are not well adapted to Bayesian model selection. One obvious solution is to run a sequential MC method for each candidate model (Djurić 1999). Unfortunately, this approach is very computationally intensive and, in addition, the same computational effort is attributed to each model. This can be very wasteful as some of these models are of no interest in practice because they have very weak posterior model probabilities.

In this paper, we develop an original sequential MC method to perform Bayesian model selection. This on-line simulation-based estimation scheme combines sequential importance sampling, a selection scheme and reversible jump MCMC moves. It allows us to run a single algorithm which concentrates itself on the model orders of interest. We apply this method to the problem of estimating, on-line, the parameters, noise variance, and number of neurons of a hybrid model consisting of a linear model and a radial basis function (RBF) network. We propose a full Bayesian prior that allows us to integrate the basis coefficients and linear regression parameters. As a result of this, we obtain an efficient algorithm that computes the coefficients and linear regression parameters using a stochastic bank of Kalman filters.

The article is organised as follows: Section 2 introduces the model selection problem and describes the approximation and probabilistic models. It also presents the estimation objectives. In Section 3, we propose a generic simulation-based method to perform sequential

Bayesian estimation and model order selection simultaneously and discuss the related implementation issues. In Section 4, we apply the proposed algorithm to sequential Bayesian model order selection of hybrid linear-RBF models observed in Gaussian noise. The experimental results are presented in Section 5. Finally, we draw some conclusions in Section 6 and describe the notation in Appendix A.

2 Problem Statement

We consider a sequential data processing scenario, where the model can change at each time step. In particular, the model active at time t will be denoted $\mathcal{M}_{t,k_t}$, where k_t corresponds to its current dimension. Examples include neural network models where the number of neurons changes over time and autoregressive (AR) processes where the number of lags also varies with time (Andrieu, de Freitas and Doucet 1999b). However, the framework is more general that the one of model order selection in embedded models. As the subscript t indicates, we can have different varieties of models in our formulation. For instance, one could easily treat the problem of neural networks where the type of basis functions changes over time.

For each model, at time t, there is a corresponding set of unknown parameters $\boldsymbol{\varphi}_{k_t,t} \in \boldsymbol{\Omega}_{k_t,t}$. For example, these parameters could correspond to the weights of a neural network or a set of AR coefficients. We use the subscript k_t to emphasise that the dimension of the parameter vector (k_t) changes with time.

If the dimension k_t takes an integer value between 0 and $k_{\max}$ at each time step, then there are $(k_{max}+1)^{t+1}$ possible model trajectories between time 0 and time t. We can describe the finite set of candidate models for this period with $\mathcal{M}_{0:t,k_{0:t}}$, $k_{0:t} \in \{0,\ldots,k_{max}\}^{t+1}$, where $k_{0:t} \triangleq \{k_0, k_1, \ldots, k_t\}$. The corresponding set of parameters is given by $\boldsymbol{\varphi}_{k_{0:t},0:t} \in \boldsymbol{\Omega}_{k_{0:t},0:t}$.

We assume that these parameters and their number are random quantities with given prior distributions. We may as a result define the following joint distribution for these quantities given the data $\mathbf{d}_{1:t}$:

$$p(k_{0:t}, \boldsymbol{\varphi}_{k_{0:t},0:t}, \mathbf{d}_{1:t}) = p(\mathbf{d}_{1:t}|k_{0:t}, \boldsymbol{\varphi}_{k_{0:t},0:t})p(\boldsymbol{\varphi}_{k_{0:t},0:t}|k_{0:t})p(k_{0:t})$$

where $p(\mathbf{d}_{1:t}|k_{0:t}, \boldsymbol{\varphi}_{k_{0:t},0:t})$ is the likelihood function, $p(\boldsymbol{\varphi}_{k_{0:t},0:t}|k_{0:t})$ is the prior parameters distribution and $p(k_{0:t})$ is the prior model distribution. This general framework encompasses many types of signal models. For demonstration purposes, we address the problem of neural network models.

2.1 Example: Radial Basis Functions Networks

Many physical processes may be described by the following nonlinear, multivariate input-output mapping:

$$\mathbf{y}_t = \mathbf{f}(\mathbf{x}_t) + \mathbf{n}_t$$

where $\mathbf{x}_t \in \mathbb{R}^d$ corresponds to a group of input variables, $\mathbf{y}_t \in \mathbb{R}^c$ to the output variables and $\mathbf{n}_t \in \mathbb{R}^c$ to an unknown noise process. In this context, the learning problem involves computing an approximation to the function $\mathbf{f}$ and estimating the characteristics of the noise process each time new data $\mathbf{d}_t \triangleq \{\mathbf{x}_t, \mathbf{y}_t\}$ arrive.

When the exact nonlinear structure of the multivariate function $\mathbf{f}$ cannot be established *a priori*, it may be synthesised as a combination of parametrised basis functions. That is:

$$\widehat{\mathbf{f}}(\mathbf{x}_t, \boldsymbol{\theta}_t) = G_{t,k}\left(\boldsymbol{\theta}_{t,k}; \left(\cdots \sum_j G_{t,j}\left(\boldsymbol{\theta}_{t,j}; \sum_i G_{t,i}(\boldsymbol{\theta}_{t,i}; \mathbf{x}_t)\right)\cdots\right)\right)$$

where $G_{t,i}(\boldsymbol{\theta}_{t,i}; \mathbf{x}_t)$ denotes a multivariate basis function. For the purposes of this paper, we adopt an approximation scheme consisting of a mixture of k radial basis functions (RBFs) and a linear regression term (Holmes and Mallick 1998). However, the work can be straightforwardly extended to other regression models. More precisely, our approximation model $\mathcal{M}$ for the observations at time t is:

$$\begin{array}{lll} \mathcal{M}_{t,0}: & \mathbf{y}_t = \boldsymbol{b}_t + \boldsymbol{\beta}_t'\mathbf{x}_t + \mathbf{n}_t & k_t = 0 \\ \mathcal{M}_{t,k_t}: & \mathbf{y}_t = \sum_{j=1}^{k_t} \mathbf{a}_{j,t}\phi_t(\|\mathbf{x}_t - \boldsymbol{\mu}_{j,t}\|) + \boldsymbol{b}_t + \boldsymbol{\beta}_t'\mathbf{x}_t + \mathbf{n}_t & k_t \geq 1 \end{array} \qquad (1)$$

where $\|\cdot\|$ denotes a distance metric (usually Euclidean or Mahalanobis), $\boldsymbol{\mu}_{j,t} \in \mathbb{R}^d$ denotes the j-th RBF centre for a model with k RBFs, $\mathbf{a}_{j,t} \in \mathbb{R}^c$ the j-th RBF amplitude and $\boldsymbol{b}_t \in \mathbb{R}^c$ and $\boldsymbol{\beta}_t \in \mathbb{R}^d \times \mathbb{R}^c$ the linear regression parameters. The noise sequence $\mathbf{n}_t \in \mathbb{R}^c$ is assumed to be zero-mean Gaussian; its variance changes over time. It is important to mention that although we have not explicitly indicated the dependency of $\boldsymbol{b}_t$, $\boldsymbol{\beta}_t$ and $\mathbf{n}_t$ on k_t, these parameters are indeed affected by the value of k_t.

Depending on our *a priori* knowledge about the smoothness of the mapping, we can choose different types of basis functions (Girosi, Jones and Poggio 1995). The most common choices are linear, cubic, thin plate spline, Gaussian and multiquadric basis functions (Andrieu, de Freitas and Doucet 1999a, Holmes and Mallick 1998). For convenience, we express our approximation model in vector-matrix form[2]:

$$\mathbf{y}_t = \mathbf{D}(\boldsymbol{\mu}_{1:k_t,t}, \mathbf{x}_t)\boldsymbol{\alpha}_{1:c,1:1+d+k_t,t} + \mathbf{n}_t$$

where:

$$\begin{aligned} \mathbf{y}_t &= [\mathbf{y}_{1,t} \cdots \mathbf{y}_{c,t}] \\ \mathbf{D}(\boldsymbol{\mu}_{1:k_t,t}, \mathbf{x}_t) &= [1 \quad \mathbf{x}_{1,t} \cdots \mathbf{x}_{d,t} \quad \phi(\mathbf{x}_{1:d,t}, \boldsymbol{\mu}_{1:d,1,t}) \cdots \phi(\mathbf{x}_{1:d,t}, \boldsymbol{\mu}_{1:d,k_t,t})] \\ \boldsymbol{\alpha}_{1:c,1:1+d+k_t,t} &= \begin{bmatrix} \boldsymbol{b}_{1,t} & \boldsymbol{\beta}_{1,1,t} \cdots \boldsymbol{\beta}_{d,1,t} & \mathbf{a}_{1,1,t} \cdots \mathbf{a}_{1,k_t,t} \\ \boldsymbol{b}_{2,t} & \boldsymbol{\beta}_{1,2,t} \cdots \boldsymbol{\beta}_{d,2,t} & \mathbf{a}_{2,1,t} \cdots \mathbf{a}_{2,k_t,t} \\ & \vdots & \\ \boldsymbol{b}_{c,t} & \boldsymbol{\beta}_{1,c,t} \cdots \boldsymbol{\beta}_{d,c,t} & \mathbf{a}_{c,1,t} \cdots \mathbf{a}_{c,k_t,t} \end{bmatrix}' \end{aligned}$$

[2]We adopt the notation $\mathbf{y}_{1:c,t} \triangleq (\mathbf{y}_{1,t}, \mathbf{y}_{2,t}, \ldots, \mathbf{y}_{c,t})'$ to denote all the output observations at time t. To simplify this notation, $\mathbf{y}_t$ is equivalent to $\mathbf{y}_{1:c,t}$. That is, if one index does not appear, it is implied that we are referring to all of its possible values. We shall favour the shorter notation and only invoke the longer notation to avoid ambiguities and emphasise certain dependencies.

and, finally, the noise process is assumed to be normally distributed as follows:

$$\mathbf{n}_t \sim \mathcal{N}\left(\mathbf{0}_{c\times 1}, diag\left(\boldsymbol{\sigma}_{1,t}^2, \ldots, \boldsymbol{\sigma}_{c,t}^2\right)\right)$$

Once again, we stress that $\boldsymbol{\sigma}_t^2$ depends implicitly on the model order k. The number k_t of RBFs and their parameters $\boldsymbol{\theta}_{k_t,t} \triangleq \{\boldsymbol{\alpha}_{1:c,1:m_t,t}, \boldsymbol{\mu}_{1:d,1:k_t,t}, \log(\boldsymbol{\sigma}_{1:c,t}^2)\}$, with $m_t = 1 + d + k_t$, are assumed to be unknown and need to be estimated sequentially.

2.2 Bayesian model

A state space representation is adopted to describe the network's evolution over time:

$$\begin{aligned}
\mathbf{y}_t &= \mathbf{D}(\boldsymbol{\mu}_{1:k_t,t}, \mathbf{x}_t)\boldsymbol{\alpha}_{1:m_t,t} + \mathbf{n}_t && (2)\\
k_{t+1} &= k_t + \epsilon_k && (3)\\
\boldsymbol{\mu}_{1:k_{t+1},t+1} &= \boldsymbol{\mu}_{1:k_{t+1},t} + \epsilon_{\boldsymbol{\mu}} && (4)\\
\boldsymbol{\alpha}_{1:m_{t+1},t+1} &= \boldsymbol{\alpha}_{1:m_{t+1},t} + \epsilon_{\boldsymbol{\alpha}} && (5)\\
\log(\boldsymbol{\sigma}_{t+1}^2) &= \log(\boldsymbol{\sigma}_t^2) + \epsilon_{\boldsymbol{\sigma}} && (6)
\end{aligned}$$

where the diffusion process ϵ_k is sampled from a discrete distribution:

$$\epsilon_k \sim p(k_{t+1}|k_t)$$

The other processes are sampled from normal distributions:

$$\begin{aligned}
\epsilon_{\boldsymbol{\mu}} &\sim \mathcal{N}(0, \delta_{\boldsymbol{\mu}}^2 \mathbf{I}_d)\\
\epsilon_{\boldsymbol{\alpha}} &\sim \mathcal{N}(0, \delta_{\boldsymbol{\alpha}}^2 \mathbf{I}_{m_t})\\
\epsilon_{\boldsymbol{\sigma}} &\sim \mathcal{N}(0, \delta_{\boldsymbol{\sigma}}^2 \mathbf{I}_c)
\end{aligned}$$

where $m_t = k_t + d + 1$ and $\mathbf{I}_m$ denotes the identity matrix of size $m \times m$. Note that if k_t changes, the dimensions of $\boldsymbol{\alpha}_{1:m_t,t}$ and $\boldsymbol{\mu}_{1:k_t,t}$ also change accordingly. In the example of RBFs, if k_t increases, new basis centres $\boldsymbol{\mu}_t$ are sampled from a Gaussian distribution centred at the data $\mathbf{x}_t$. That is, one stochastically places basis functions where there is more information. On the other hand, if k_t decreases, one samples uniformly from the existing set of basis functions to decide which bases should be deleted.

The joint distribution of the proposed model, with $\boldsymbol{\theta}_{k_{0:t},0:t} \triangleq \{\boldsymbol{\alpha}_{1:m_{0:t},0:t}, \boldsymbol{\mu}_{1:k_{0:t},0:t}, \boldsymbol{\sigma}_{0:t}^2\}$, is:

$$\begin{aligned}
&p(k_{0:t}, \boldsymbol{\theta}_{k_{0:t},0:t}, \mathbf{x}_{1:t}, \mathbf{y}_{1:t}) \propto p(\mathbf{y}_{1:t}|k_{0:t}, \boldsymbol{\theta}_{k_{0:t},0:t}, \mathbf{x}_{1:t}) p(\boldsymbol{\mu}_{1:k_{0:t},0:t}) p(\boldsymbol{\alpha}_{1:m_{0:t},0:t}) p(\boldsymbol{\sigma}_{0:t}^2) p(k_{0:t})\\
&= \left[\prod_{l=1}^{t} p(\mathbf{y}_l|k_l, \boldsymbol{\theta}_{k_l,l}, \mathbf{x}_l) p(\boldsymbol{\mu}_{1:k_l,l}|\boldsymbol{\mu}_{1:k_{l-1},l-1}) p(\boldsymbol{\alpha}_{1:m_l,l}|\boldsymbol{\alpha}_{1:m_{l-1},l-1}) p(\boldsymbol{\sigma}_l^2|\boldsymbol{\sigma}_{l-1}^2) p(k_l|k_{l-1})\right]\\
&\quad \times p(\boldsymbol{\mu}_{k_0,0}) p(\boldsymbol{\alpha}_{m_0,0}) p(\boldsymbol{\sigma}_0^2) p(k_0) \qquad (7)
\end{aligned}$$

Multivariate normal distributions are adopted to represent the priors $p(\boldsymbol{\mu}_{k_0,0})$ and $p(\boldsymbol{\alpha}_{k_0,0})$, a uniform distribution for $p(\boldsymbol{\sigma}_{k_0,0}^2)$ and a discrete uniform distribution for $p(k_0)$. The parameters of these priors should reflect our degree of belief in the initial values of the various quantities of interest (Bernardo and Smith 1994).

In our model, it is clear that $\boldsymbol{\varphi}_{k_{0:t},0:t} = \boldsymbol{\theta}_{k_{0:t},0:t}$ and:

$$\boldsymbol{\Omega}_{k_{0:t},0:t} \triangleq (\mathbb{R}^{d+1+S_{k_t}})^c \times (\mathbb{R}^{S_{k_t}})^d \times \mathbb{R}^{c(t+1)}$$

where $\boldsymbol{\alpha}_{1:c,1:m_{0:t},0:t} \in (\mathbb{R}^{d+1+S_{k_t}})^c$, $\boldsymbol{\mu}_{1:d,1:k_{0:t},0:t} \in (\mathbb{R}^{S_{k_t}})^d$, $\boldsymbol{\sigma}^2_{1:c,0:t} \in \mathbb{R}^{c(t+1)}$ and $S_{k_t} = \sum_{l=0}^{t} k_l$. Figure 1 shows the directed acyclic graphical model representation of the joint distribution for the first four time steps.

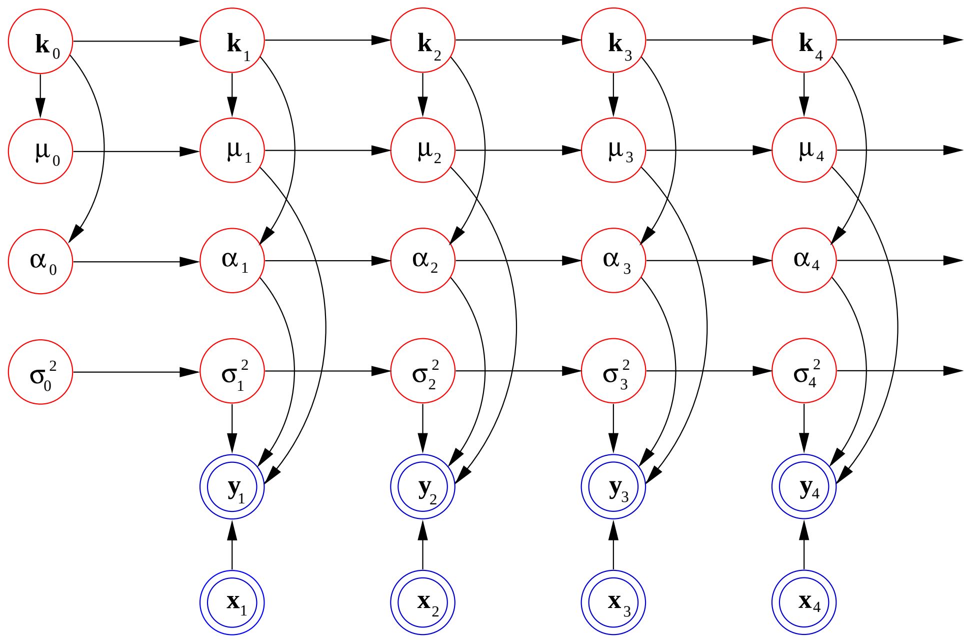

Figure 1: Dynamical directed acyclic graphical model. The circles represent unknowns, while the double circles correspond to the data. The hyper-parameters $\delta^2_{\boldsymbol{\mu}}$, $\delta^2_{\boldsymbol{\alpha}}$, $\delta^2_{\boldsymbol{\sigma}}$ and δ_k have been omitted for clarity.

2.3 Estimation objectives

Our aim is to estimate the joint posterior distribution $p(k_{0:t}, \boldsymbol{\varphi}_{k_{0:t},0:t}|\mathbf{d}_{1:t})$, defined on the space $\boldsymbol{\Omega}_{0:t} = \bigcup_{l=1}^{(k_{max}+1)^{t+1}} \boldsymbol{\Omega}_{k^{[l]}_{0:t},0:t} \times \{k^{[l]}_{0:t}\}$, recursively in time. Note that $k^{[l]}_{0:t}$ represents the l-th possible model trajectory for an arbitrary ordering. According to Bayes' rule, the posterior distribution $p(k_{0:t}, \boldsymbol{\varphi}_{k_{0:t},0:t}|\mathbf{d}_{1:t})$ is given by:

$$\frac{p(\mathbf{d}_t|\mathbf{d}_{1:t-1}, \boldsymbol{\varphi}_{k_{0:t},0:t}, k_{0:t})p(k_{0:t}, \boldsymbol{\varphi}_{k_{0:t},0:t}|\mathbf{d}_{1:t-1})}{\sum_{l=1}^{(k_{max}+1)^{t+1}} \int_{\boldsymbol{\Omega}_{k^{[l]}_{0:t},0:t}} p(\mathbf{d}_t|\mathbf{d}_{1:t-1}, \boldsymbol{\varphi}_{k^{[l]}_{0:t},0:t}, k^{[l]}_{0:t})p(k^{[l]}_{0:t}, \boldsymbol{\varphi}_{k^{[l]}_{0:t},0:t}|\mathbf{d}_{1:t-1})d\boldsymbol{\varphi}_{k^{[l]}_{0:t},0:t}}$$

It is, in most occasions, impossible to derive closed-form analytical expressions for this distribution and its features of interest, such as the posterior model probabilities $p(k_{0:t}|\mathbf{d}_{1:t})$. As a result, one has to resort to computational methods in most situations.

3 Sequential Monte Carlo

The basic idea in Monte Carlo simulation is that a set of weighted particles (samples), drawn from the posterior distribution of the model parameters, is used to map the integrations, involved in the inference process, to discrete sums. When, for simplicity, the model dimension is known and fixed, we may make use of the following Monte Carlo approximation:

$$\widehat{p}(\boldsymbol{\varphi}_{0:t}|\mathbf{d}_{0:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{\varphi}_{0:t}^{(i)}}(\mathrm{d}\boldsymbol{\varphi}_{0:t})$$

where $\boldsymbol{\varphi}_{0:t}^{(i)}$ represents the particles used to describe the posterior distribution and $\delta(\mathrm{d}\cdot)$ denotes the Dirac delta function. Consequently, any expectations of the form:

$$\mathbf{E}[f_t(\boldsymbol{\varphi}_{0:t})] = \int f_t(\boldsymbol{\varphi}_{0:t}) p(\boldsymbol{\varphi}_{0:t}|\mathbf{d}_{1:t})\mathrm{d}\boldsymbol{\varphi}_{0:t}$$

may be approximated by the following estimate:

$$\mathbf{E}[f_t(\boldsymbol{\varphi}_{0:t})] \approx \frac{1}{N}\sum_{i=1}^{N} f_t(\boldsymbol{\varphi}_{0:t}^{(i)})$$

where the particles $\boldsymbol{\varphi}_{0:t}^{(i)}$, $i = 1, \ldots, N$, are drawn from the posterior density function and assumed to be "sufficiently" independent for the approximation to hold. Monte Carlo sampling techniques are an improvement over direct numerical approximation in that they automatically select particles in regions of high probability.

Many researchers in the the statistical and signal processing communities have proposed the use of sequential Monte Carlo (SMC) algorithms; see, for example, (Doucet 1998, Liu and Chen 1998) for reviews on the topic. Figure 2 illustrates the operation of a simple SMC method, where one samples proposal candidates from the transition prior. Only the fittest particles, that is the ones with the highest likelihood, are selected in the update stage. These then proceed to be multiplied, according to their likelihood, in the prediction stage.

Most existing SMC methods belong to a framework where the model order is assumed to be known. We present here a Bayesian importance sampling method to deal with the more general case of unknown model dimension. To accomplish this goal in an efficient way, we introduce reversible jump MCMC steps into the algorithm. In the next subsection, we present a generic version of our algorithm.

3.1 Generic SMC algorithm

Given N particles $\{(\boldsymbol{\varphi}_{k_{0:t-1},0:t-1}^{(i)}, k_{0:t-1}^{(i)}); i \in \{1, \ldots, N\}\}$ at time $t-1$, approximately distributed according to $p(\boldsymbol{\varphi}_{k_{0:t-1},0:t-1}, k_{0:t-1}|\mathbf{d}_{1:t-1})$, the algorithm allows us to compute N particles $(\boldsymbol{\varphi}_{k_{0:t},0:t}^{(i)}, k_{0:t}^{(i)})$, approximately distributed according to the posterior $p(\boldsymbol{\varphi}_{k_{0:t},0:t}, k_{0:t}|\mathbf{d}_{1:t})$, at time t. Before presenting the algorithm, let us first introduce the quantities $(\boldsymbol{\varphi}_{k_t,-t}, k_t)$ to denote the set of variables in $(\boldsymbol{\varphi}_{k_{0:t},0:t}, k_t)$, but not in $(\boldsymbol{\varphi}_{k_{0:t-1},0:t-1}, k_{t-1})$ (note that this set can be empty for some problems). The importance function for this set will be denoted

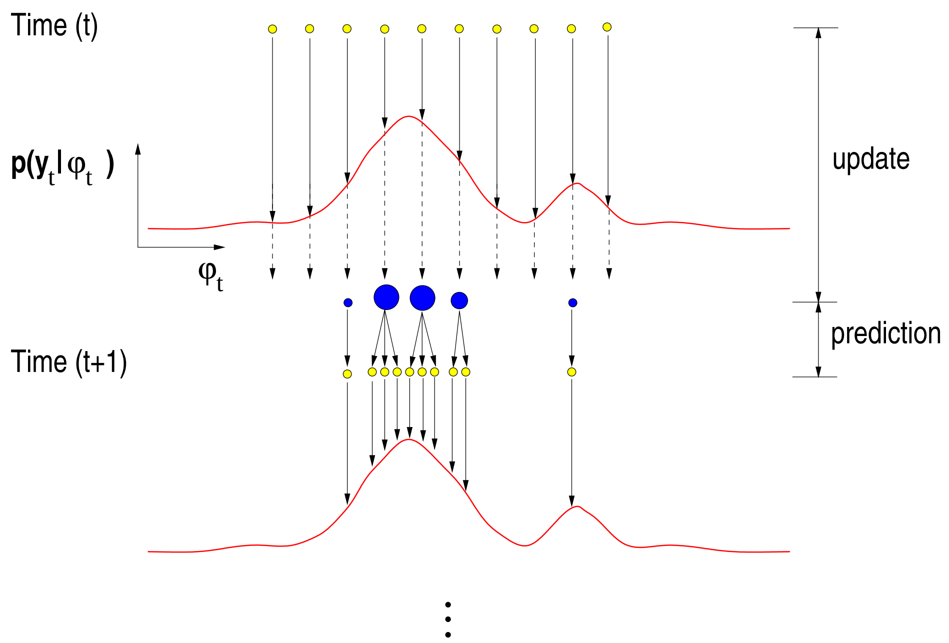

Figure 2: Update and prediction stages in a generic sequential Monte Carlo algorithm with the transition prior as proposal. In the update stage, the likelihood of each particle is evaluated. The size of the circles indicates the likelihood of a particular particle. The particles are then selected according to their respective likelihoods. In this process, the particles with higher likelihood are allowed to have more "children". Subsequently, the algorithm computes the predicted values of the particles by evaluating the transition equations. The end result is that the surviving particles provide a better weighted description of the likelihood function.

$q(\varphi_{k_t,-t}, k_t|\varphi_{k_{0:t-1},0:t-1}, k_{0:t-1}, \mathbf{d}_{1:t})$. This function is such that the support of the distribution $p(\boldsymbol{\varphi}_{k_{0:t},0:t}, k_{0:t}|\mathbf{d}_{1:t})$ is included in the support of the product $p(\boldsymbol{\varphi}_{k_{0:t-1},0:t-1}, k_{0:t-1}|\mathbf{d}_{1:t-1})$ $q(\boldsymbol{\varphi}_{k_t,-t}, k_t|\boldsymbol{\varphi}_{k_{0:t-1},0:t-1}, k_{0:t-1}, \mathbf{d}_{1:t})$. The algorithm proceeds as follows at time t:

SMC Algorithm for Bayesian Model Selection

1. Bayesian importance sampling step

   - For $i = 1, \ldots, N$, sample:

$$(\widehat{\varphi}^{(i)}_{\widehat{k}_t,-t}, \widehat{k}^{(i)}_t) \sim q(\boldsymbol{\varphi}_{k_t,-t}, k_t|\boldsymbol{\varphi}^{(i)}_{k_{0:t-1},0:t-1}, k^{(i)}_{0:t-1}, \mathbf{d}_{1:t})$$

   and set:

$$(\widehat{\varphi}^{(i)}_{\widehat{k}_{0:t},0:t}, \widehat{k}^{(i)}_{0:t}) \triangleq ((\boldsymbol{\varphi}^{(i)}_{k_{0:t-1},0:t-1}, k^{(i)}_{0:t-1}), (\widehat{\varphi}^{(i)}_{\widehat{k}_t,-t}, \widehat{k}^{(i)}_t))$$

- For $i=1,\ldots,N$, evaluate the importance weights up to a normalising constant:

$$w_t^{(i)} = \frac{p(\widehat{\varphi}^{(i)}_{\widehat{k}_{0:t},0:t}, \widehat{k}^{(i)}_{0:t}|\mathbf{d}_{1:t})}{p(\widehat{\varphi}^{(i)}_{\widehat{k}_{0:t-1},0:t-1}, \widehat{k}^{(i)}_{0:t-1}|\mathbf{d}_{1:t-1})q(\widehat{\varphi}^{(i)}_{\widehat{k}_t,-t}, \widehat{k}^{(i)}_t|\varphi^{(i)}_{k_{0:t-1},0:t-1}, k^{(i)}_{0:t-1}, \mathbf{d}_{1:t})} \tag{8}$$

- For $i=1,\ldots,N$, normalise the importance weights:

$$\widetilde{w}_t^{(i)} = w_t^{(i)} \left[\sum_{j=1}^{N} w_t^{(j)}\right]^{-1} \tag{9}$$

2. Selection step

- Multiply/Suppress samples $(\widehat{\varphi}^{(i)}_{\widehat{k}_{0:t},0:t}, \widehat{k}^{(i)}_{0:t})$ with high/low importance weights $\widetilde{w}_t^{(i)}$, respectively, to obtain N random samples $(\widetilde{\varphi}^{(i)}_{\widetilde{k}_{0:t},0:t}, \widetilde{k}^{(i)}_{0:t})$ approximately distributed according to $p(\widetilde{\varphi}^{(i)}_{\widetilde{k}_{0:t},0:t}, \widetilde{k}^{(i)}_{0:t}|\mathbf{d}_{1:t})$.

3. MCMC step

- Apply a Markov transition kernel with invariant distribution given by the product $\prod_{i=1}^{N} p(\varphi^{(i)}_{k_{0:t},0:t}, k^{(i)}_{0:t}|\mathbf{d}_{1:t})$ to obtain $(\varphi^{(i)}_{k_{0:t},0:t}, k^{(i)}_{0:t})$.

■

3.2 Implementation issues

3.2.1 Bayesian importance sampling step

As mentioned earlier, Bayesian importance sampling has been formulated mainly in scenarios where the distribution from which one wishes to sample is defined on a simple set with a unique dominating measure (Geweke 1989). For the sake of clarity, this section describes the more general case that arises when the target distribution is defined on a union of distinct subspaces, that is:

$$\pi(\varphi_{k_{0:t},0:t}, k_{0:t}) = \sum_{l=1}^{(k_{max}+1)^{t+1}} \pi_{k^{[l]}_{0:t}}(\varphi_{k^{[l]}_{0:t},0:t}, k^{[l]}_{0:t})\mathbb{I}_{\mathbf{\Omega}_{k^{[l]}_{0:t},0:t}\times\{k^{[l]}_{0:t}\}}(\varphi_{k_{0:t},0:t}, k_{0:t})$$

Let us introduce the following importance distribution:

$$q(\varphi_{k_{0:t},0:t}, k_{0:t}) = \sum_{l=1}^{(k_{max}+1)^{t+1}} q_{k^{[l]}_{0:t}}(\varphi_{k^{[l]}_{0:t},0:t}, k^{[l]}_{0:t})\mathbb{I}_{\mathbf{\Omega}_{k^{[l]}_{0:t},0:t}\times\{k^{[l]}_{0:t}\}}(\varphi_{k_{0:t},0:t}, k_{0:t})$$

where the support of $q_{k^{[l]}_{0:t}}(\varphi_{k^{[l]}_{0:t},0:t}, k^{[l]}_{0:t})$ includes the support of $\pi_{k^{[l]}_{0:t}}(\varphi_{k^{[l]}_{0:t},0:t}, k^{[l]}_{0:t})$ for $l = 1,\ldots,(k_{max}+1)^{t+1}$. Hence, $\pi(\varphi_{k_{0:t},0:t}, k_{0:t})$ can be rewritten as:

$$\frac{\sum_{l=1}^{(k_{max}+1)^{t+1}} w_{k^{[l]}_{0:t}}(\varphi_{k^{[l]}_{0:t},0:t}, k^{[l]}_{0:t})q_{k^{[l]}_{0:t}}(\varphi_{k^{[l]}_{0:t},0:t}, k^{[l]}_{0:t})\mathbb{I}_{\mathbf{\Omega}_{k^{[l]}_{0:t},0:t}\times\{k^{[l]}_{0:t}\}}(\varphi_{k_{0:t},0:t}, k_{0:t})}{\sum_{l=1}^{(k_{max}+1)^{t+1}} \int_{\mathbf{\Omega}_{k^{[l]}_{0:t},0:t}} w_{k^{[l]}_{0:t}}(\varphi_{k^{[l]}_{0:t},0:t}, k^{[l]}_{0:t})q_{k^{[l]}_{0:t}}(\varphi_{k^{[l]}_{0:t},0:t}, k^{[l]}_{0:t})d\varphi_{k^{[l]}_{0:t},0:t}}$$

where

$$w_{k_{0:t}^{[l]}}(\boldsymbol{\varphi}_{k_{0:t}^{[l]},0:t}, k_{0:t}^{[l]}) = \frac{\pi_{k_{0:t}^{[l]}}(\boldsymbol{\varphi}_{k_{0:t}^{[l]},0:t}, k_{0:t}^{[l]})}{q_{k_{0:t}^{[l]}}(\boldsymbol{\varphi}_{k_{0:t}^{[l]},0:t}, k_{0:t}^{[l]})}$$

represents the unnormalised importance ratios. Let us assume that we have N samples $(\boldsymbol{\varphi}_{k_{0:t},0:t}^{(i)}, k_{0:t}^{(i)})$ distributed according to $q(\boldsymbol{\varphi}_{k_{0:t},0:t}, k_{0:t})$, then the Bayesian importance sampling method gives us the following point mass approximation of $\pi(\boldsymbol{\varphi}_{k_{0:t},0:t}, k_{0:t})$:

$$\widehat{\pi}(d\boldsymbol{\varphi}_{k_{0:t},0:t}, k_{0:t}) = \frac{\sum_{l=1}^{(k_{max}+1)^{t+1}} \sum_{I^{\star}_{k_{0:t}^{[l]}}} w_{k_{0:t}^{[l],(i)}}(\boldsymbol{\varphi}^{(i)}_{k_{0:t}^{[l],(i)},0:t}, k_{0:t}^{[l],(i)}) \delta_{\boldsymbol{\varphi}^{(i)}_{k_{0:t}^{(i)},0:t}, k_{0:t}^{(i)}}(\mathrm{d}\boldsymbol{\varphi}_{k_{0:t},0:t}, k_{0:t})}{\sum_{l=1}^{(k_{max}+1)^{t+1}} \sum_{I^{\star}_{k_{0:t}^{[l]}}} w_{k_{0:t}^{[l],(i)}}(\boldsymbol{\varphi}^{(i)}_{k_{0:t}^{[l],(i)},0:t}, k_{0:t}^{[l],(i)})} \tag{10}$$

with $I^{\star}_{k_{0:t}^{[l]}} = \{i \in \{1, \ldots, N\} / (\boldsymbol{\varphi}^{(i)}_{k_{0:t},0:t}, k_{0:t}^{(i)}) \in \boldsymbol{\Omega}_{k_{0:t}^{[l]},0:t} \times k_{0:t}^{[l]}\}$. Equations (8) and (9) follow.

3.2.2 Selection step

A selection scheme associates to each particle $(\widehat{\boldsymbol{\varphi}}^{(i)}_{\widehat{k}_{0:t},0:t}, \widehat{k}_{0:t}^{(i)})$ a number of "children", say $N_i \in \mathbb{N}$, such that $\sum_{i=1}^{N} N_i = N$. We describe some interesting selection schemes proposed in the literature. These schemes satisfy $\mathbb{E}[N_i] = N\widetilde{w}_t^{(i)}$ but their performance varies in terms of $var[N_i]$.

• *Sampling Importance Resampling (SIR)/Multinomial Sampling procedure*: this procedure, originally introduced by (Gordon et al. 1993), is the most popular one. One samples N times from $\sum_{i=1}^{N} \widetilde{w}_t^{(i)} \delta_{(\widehat{\boldsymbol{\varphi}}^{(i)}_{\widehat{k}_{0:t},0:t}, \widehat{k}_{0:t}^{(i)})}(\mathrm{d}\widetilde{\boldsymbol{\varphi}}_{\widetilde{k}_{0:t},0:t}, \widetilde{k}_{0:t})$ to obtain $((\widetilde{\boldsymbol{\varphi}}^{(i)}_{\widetilde{k}_{0:t},0:t}, \widetilde{k}_{0:t}^{(i)}); i = 1, \ldots, N)$. This is equivalent to drawing jointly $(N_i; i = 1, \ldots, N)$ according to a multinomial distribution of parameters N and $\widetilde{w}_t^{(i)}$. This procedure can be implemented in $\mathcal{O}(N)$ iterations (Doucet 1998, Pitt and Shephard 1999) following the work of (Ripley 1987, pp. 96). However, as pointed out in (Carpenter, Clifford and Fearnhead 1999) and (Liu and Chen 1998), it is better to use selection schemes with a limited variance.

• *Residual Resampling*: this procedure was introduced by (Liu and Chen 1998). It involves the following steps: set $\widetilde{N}_i = \left\lfloor N\widetilde{w}_t^{(i)} \right\rfloor$ and perform an SIR procedure to select the remaining $\overline{N}_t = N - \sum_{i=1}^{N} \widetilde{N}_i$ samples with the new weights $w_t'^{(i)} = \overline{N}_t^{-1}\left(\widetilde{w}_t^{(i)} N - \widetilde{N}_i\right)$, finally add the results to the current $\widetilde{N}_i$. For this scheme $var[N_i]$ is smaller than the one given by the SIR scheme. Moreover, this procedure is computationally cheaper.

• *Systematic Sampling* (Carpenter et al. 1999)*:* this method introduces a variance on N_i even smaller than the residual resampling scheme. Its computational complexity is $\mathcal{O}(N)$.

As a matter of fact, the restriction $\mathbb{E}[N_i] = N\widetilde{w}_t^{(i)}$ is unnecessary to obtain convergence results. So it is possible to use biased but very quick selection schemes.

3.2.3 MCMC steps

After the selection scheme at time t, we obtain N particles distributed marginally approximately according to $p(k_{0:t}, \boldsymbol{\varphi}_{k_{0:t},0:t} | \mathbf{d}_{1:t})$. In practice, a skewed importance weights distribution implies that many particles have had no children, i.e. $N_i = 0$, whereas others have had a large number of children, the extreme case being $N_i = N$ for a particular value i. In this case, there is a severe depletion of samples. A strategy for improving the results consists of introducing MCMC steps of invariant distribution $p(k_{0:t}, \varphi_{k_{0:t},0:t}|\mathbf{d}_{1:t})$ on each particle (Andrieu et al. 1999b, Carpenter et al. 1999, Doucet and Gordon 1999, Gilks and Berzuini 1998, MacEachern, Clyde and Liu 1999). The basic idea is that, by applying a Markov transition kernel, the total variation of the current distribution with respect to the invariant distribution can only decrease. Note, however, that we do not require this kernel to be ergodic. Convergence results for this type of algorithm are presented in (Gilks and Berzuini 1998).

It is possible to generalise this idea and to introduce MCMC steps on the product space with an invariant distribution $\prod_{i=1}^{N} p(\varphi^{(i)}_{k_{0:t},0:t}, k^{(i)}_{0:t}|\mathbf{y}_{1:t})$, that is apply MCMC steps on the *whole population* of particles. In doing so, it allows us to introduce the algorithms developed in parallel MCMC computation. This paper is limited to the use of simple MCMC transitions steps on each particle. In particular, reversible jump MCMC steps (Green 1995) are proposed to allow the particles to move from one subspace to the others.

4 SMC Applied to RBF Networks

This section shows that it is possible to integrate the nuisance parameters $\boldsymbol{\alpha}_{1:m_t,t}$, leading to a more efficient algorithm. Subsequently, it introduces the reversible jump MCMC algorithm in the context of sequential sampling and describes the SMC algorithm for RBFs. Finally, it presents the implementation details for the sampling, selection and MCMC steps.

In the RBF signal model, $(\boldsymbol{\varphi}_{k_t,-t}, k_t) = (\boldsymbol{\theta}_{k_t,t}, k_t) = (\{\boldsymbol{\alpha}_{1:m_t,t}, \boldsymbol{\mu}_{1:k_t,t}, \boldsymbol{\sigma}^2_t\}, k_t)$. It is important to note that for the parameters $\boldsymbol{\alpha}_{1:m_t,t}$, conditionally upon $(\boldsymbol{\mu}_{1:k_t,t}, \boldsymbol{\sigma}^2_t\}, k_t)$, we have a linear Gaussian state space model. Consequently, it is possible to integrate out $\boldsymbol{\alpha}_{1:m_t,t}$, thereby reducing the variance of the estimates (Doucet, Gordon and Krishnamurthy 1999, Propositions 1 and 2). The end result corresponds to a bank of Kalman filters for each network output[3] (Jazwinski 1970, Kalman and Bucy 1961):

$$K^{(i)}_t = (P^{(i)}_{k_t,t-1} + \delta^2_{\boldsymbol{\alpha}}\mathbf{I}_{m_t})\mathbf{D}'(\boldsymbol{\mu}_{1:k_t,t}, \mathbf{x}_t)\Big[\boldsymbol{\sigma}^2_t + \mathbf{D}(\boldsymbol{\mu}_{1:k_t,t}, \mathbf{x}_t)(P^{(i)}_{k_t,t-1} + \delta^2_{\boldsymbol{\alpha}}\mathbf{I}_{m_t})\mathbf{D}'(\boldsymbol{\mu}_{1:k_t,t}, \mathbf{x}_t)\Big]^{-1} \tag{11}$$

$$\bar{\boldsymbol{\alpha}}^{(i)}_{1:m_t,t} = \bar{\boldsymbol{\alpha}}^{(i)}_{1:m_t,t-1} + K^{(i)}_t\big(\mathbf{y}_t - \mathbf{D}(\boldsymbol{\mu}_{1:k_t,t}, \mathbf{x}_t)\bar{\boldsymbol{\alpha}}^{(i)}_{1:m_t,t-1}\big) \tag{12}$$

$$P^{(i)}_{k_t,t} = P^{(i)}_{k_t,t-1} + \delta^2_{\boldsymbol{\alpha}}\mathbf{I}_{m_t} - K^{(i)}_t\mathbf{D}(\boldsymbol{\mu}_{1:k_t,t}, \mathbf{x}_t)(P^{(i)}_{k_t,t-1} + \delta^2_{\boldsymbol{\alpha}}\mathbf{I}_{m_t}) \tag{13}$$

where K_t denotes the Kalman gain and $\bar{\boldsymbol{\alpha}}_{1:m_t,t}$ and $P_{k_t,t}$ correspond to the conditional mean and covariance matrix of $\boldsymbol{\alpha}_{1:m_t,t}$. Note that the sizes of $\bar{\boldsymbol{\alpha}}_{1:m_t,t}$ and $P_{k_t,t}$ vary over time. If k_t decreases, then we should delete, at random, $k_{t-1} - k_t$ basis function coefficients and the

[3]Note that to simplify the notation, the Kalman filter equations are described for a single output. Thus $\boldsymbol{\sigma}^2_t$ is a scalar. In general, we would need to introduce a subindex $j = 1, \ldots, c$ (that is, write explicitly $\boldsymbol{\sigma}^2_{j,t}$, $\bar{\boldsymbol{\alpha}}_{j,1:m_t,t}$ and $P_{j,k_t,t}$) to indicate the j-th output of the neural network.

corresponding row and column entries in the matrix $P_{k_{t-1},t-1}$. If k_t increases, say for example $k_t = k_{t-1}+1$, we can extend $\bar{\boldsymbol{\alpha}}_{1:m_{t-1},t-1}$ and $P_{k_{t-1},t-1}$ as follows:

$$\bar{\boldsymbol{\alpha}}_{1:m_t,t-1} = \begin{bmatrix} \bar{\boldsymbol{\alpha}}_{1:m_{t-1},t-1} & \bar{\boldsymbol{\alpha}}_{1:m_0,0} \end{bmatrix}'$$

$$P_{k_t,t-1} = \begin{bmatrix} P_{k_{t-1},t-1} & 0 \\ 0 & P_{k_0,0} \end{bmatrix}$$

where $\bar{\boldsymbol{\alpha}}_{1:m_0,0}$ and $P_{k_0,0}$ depend on our prior knowledge about the data set being modelled. By integrating out $\boldsymbol{\alpha}_{1:m_t,t}$, we may select the following importance distribution:

$$q(\boldsymbol{\mu}_{1:k_t,t}, \boldsymbol{\sigma}_t^2, k_t | \boldsymbol{\theta}_{k_{0:t-1},0:t-1}, k_{0:t-1}, \mathbf{d}_{1:t}) = p(\boldsymbol{\mu}_{1:k_t,t}|\boldsymbol{\mu}_{1:k_t,t-1}, k_t)p(\boldsymbol{\sigma}_t^2|\boldsymbol{\sigma}_{t-1}^2)p(k_t|k_{t-1})$$

Standard MCMC methods are not able to "jump" between subspaces $\boldsymbol{\Omega}_{k_{0:t},0:t}$ of different dimension. However, recently, Green has introduced a flexible class of MCMC samplers, the so-called reversible jump MCMC, that are capable of jumping between subspaces of different dimensions (Green 1995). Here, the chain must move across subspaces of different dimensions, and therefore the proposal distributions are more complex: see (Green 1995, Richardson and Green 1997) for details. For our problem, the following moves have been selected:

1. Birth of a new basis, *i.e.* proposing a new basis function.
2. Death of an existing basis, *i.e.* removing a basis function chosen randomly.
3. Update the RBF centres.

These moves are defined by heuristic considerations, the only condition to be fulfilled being to maintain the correct invariant distribution. A particular choice will only have influence on the convergence rate of the algorithm. The birth and death moves allow the network to grow from k_t to k_t+1 and decrease from k_t to k_t-1 respectively. By applying the bank of Kalman filters and incorporating reversible jump MCMC steps, the sequential model selection algorithm for RBF networks is as follows:

SMC Algorithm for RBFs Bayesian Model Selection

1. <u>Sequential importance sampling step</u>

   - For $i = 1, \ldots, N$, sample from the prior distributions:

$$\begin{aligned}
\widehat{k}_t^{(i)}|(\boldsymbol{\theta}_{k_{t-1},t-1}^{(i)}, k_{t-1}^{(i)}, \mathbf{d}_{1:t-1}) &\sim p(k_t|k_{t-1}^{(i)}) \\
\widehat{\boldsymbol{\sigma}}_t^{2(i)}|(\widehat{k}_t^{(i)}, \boldsymbol{\theta}_{k_{t-1},t-1}^{(i)}, k_{t-1}^{(i)}, \mathbf{d}_{1:t-1}) &\sim p(\boldsymbol{\sigma}_t^2|\boldsymbol{\sigma}_{t-1}^{2(i)}) \\
\widehat{\boldsymbol{\mu}}_{1:k_t,t}^{(i)}|(\widehat{k}_t^{(i)}, \boldsymbol{\theta}_{k_{t-1},t-1}^{(i)}, k_{t-1}^{(i)}, \mathbf{d}_{1:t-1}) &\sim p(\boldsymbol{\mu}_{1:k_t,t}|\boldsymbol{\mu}_{1:\widehat{k}_t,t-1}^{(i)}, \widehat{k}_t^{(i)})
\end{aligned}$$

   Change the dimension of $\bar{\boldsymbol{\alpha}}_{k_{t-1},t-1}$ and $P_{k_{t-1},t-1}$ to $\bar{\boldsymbol{\alpha}}_{\widehat{k}_t,t-1}$ and $P_{\widehat{k}_t,t-1}$ and set:

$$(\widehat{\boldsymbol{\theta}}_{\widehat{k}_{0:t},0:t}^{(i)}, \widehat{k}_{0:t}^{(i)}) \triangleq ((\boldsymbol{\theta}_{k_{0:t-1},0:t-1}^{(i)}, k_{0:t-1}^{(i)}), (\widehat{\boldsymbol{\theta}}_{\widehat{k}_t,t}^{(i)}, \widehat{k}_t^{(i)}))$$

- For $i=1,\ldots,N$, evaluate the importance weights up to a normalising constant:

$$w_t^{(i)} = p(\mathbf{y}_t|\widehat{\boldsymbol{\mu}}^{(i)}_{1:\widehat{k}_{0:t},0:t}, \widehat{\boldsymbol{\sigma}}^{2(i)}_{0:t}, \widehat{k}^{(i)}_{0:t}, \mathbf{x}_t, \mathbf{d}_{1:t-1})$$

- For $i=1,\ldots,N$, normalise the importance weights:

$$\widetilde{w}_t^{(i)} = w_t^{(i)} \left[\sum_{j=1}^{N} w_t^{(j)}\right]^{-1}$$

2. Selection step

- Use residual sampling to multiply/suppress samples $(\widehat{\boldsymbol{\theta}}^{(i)}_{\widehat{k}_{0:t},0:t}, \widehat{k}^{(i)}_{0:t})$ with high/low importance weights $\widetilde{w}_t^{(i)}$, respectively, to obtain N random samples $(\widetilde{\boldsymbol{\theta}}^{(i)}_{\widetilde{k}_{0:t},0:t}, \widetilde{k}^{(i)}_{0:t})$ approximately distributed according to $p(\widetilde{\boldsymbol{\theta}}^{(i)}_{\widetilde{k}_{0:t},0:t}, \widetilde{k}^{(i)}_{0:t}|\mathbf{d}_{1:t})$.

3. MCMC step

- Sample $u \sim \mathcal{U}_{[0,1]}$.
- If $(u \leq b_{k_t})$
  - then "birth" move (See Section 7.3.2.2).
  - else if $(u \leq b_{k_t} + d_{k_t})$ then "death" move (See Section 7.3.2.2).
  - else update $\widetilde{\boldsymbol{\mu}}_{1:\widetilde{k}_t,t}$ and $\widetilde{k}_t$ (See Section 7.3.2.1).

  End If.
- Update the parameters $\bar{\boldsymbol{\alpha}}^{(i)}_{1:m_t,t}$ and $P^{(i)}_{k_t,t}$ using a bank of Kalman filters (equations (11) to (13)).

■

4.1 Sampling and selection steps

In the prediction stage, samples from the importance function are obtained as follows:

$$\widehat{k}_t|(\boldsymbol{\mu}_{1:k_{t-1},t-1}, \boldsymbol{\sigma}^2_{t-1}, k_{t-1}, \mathbf{d}_{1:t-1}) \sim p(k_t|k_{t-1}) \tag{14}$$

$$\widehat{\boldsymbol{\sigma}}^2_t|(\widehat{k}_t, \boldsymbol{\mu}_{1:k_{t-1},t-1}, \boldsymbol{\sigma}^2_{t-1}, k_{t-1}, \mathbf{d}_{1:t-1}) \sim p(\boldsymbol{\sigma}^2_t|\boldsymbol{\sigma}^2_{t-1}) \tag{15}$$

$$\widehat{\boldsymbol{\mu}}_{1:k_t,t}|(\widehat{k}_t, \boldsymbol{\mu}_{1:\widehat{k}_t,t-1}, \boldsymbol{\sigma}^2_{t-1}, k_{t-1}, \mathbf{d}_{1:t-1}) \sim p(\boldsymbol{\mu}_{1:k_t,t}|\boldsymbol{\mu}_{1:\widehat{k}_t,t-1}, \widehat{k}_t) \tag{16}$$

Applying Bayes' rule, the posterior at time t is:

$$\begin{aligned}
&p(\boldsymbol{\mu}_{1:k_{0:t},0:t}, \boldsymbol{\sigma}^2_{0:t}, k_{0:t}|\mathbf{d}_{1:t}) \propto p(\mathbf{y}_t|\boldsymbol{\mu}_{1:k_{0:t},0:t}, \boldsymbol{\sigma}^2_{0:t}, k_{0:t}, \mathbf{x}_t, \mathbf{d}_{1:t-1}) p(\boldsymbol{\mu}_{1:k_{0:t},0:t}, \boldsymbol{\sigma}^2_{0:t}, k_{0:t}|\mathbf{d}_{1:t-1}) \\
&= p(\mathbf{y}_t|\boldsymbol{\mu}_{1:k_{0:t},0:t}, \boldsymbol{\sigma}^2_t, k_{0:t}, \mathbf{x}_t, \mathbf{d}_{1:t-1}) p(\boldsymbol{\mu}_{1:k_t,t}, \boldsymbol{\sigma}^2_t, k_t|\boldsymbol{\mu}_{1:k_{0:t-1},0:t-1}, \boldsymbol{\sigma}^2_{0:t-1}, k_{0:t-1}, \mathbf{d}_{1:t-1}) \\
&\quad \times p(\boldsymbol{\mu}_{1:k_{0:t-1},0:t-1}, \boldsymbol{\sigma}^2_{0:t-1}, k_{0:t-1}|\mathbf{d}_{1:t-1}) \\
&= p(\mathbf{y}_t|\boldsymbol{\mu}_{1:k_{0:t},0:t}, \boldsymbol{\sigma}^2_{0:t}, k_{0:t}, \mathbf{x}_t, \mathbf{d}_{1:t-1}) p(\boldsymbol{\mu}_{1:k_t,t}|\boldsymbol{\mu}_{1:k_t,t-1}, k_t) p(\boldsymbol{\sigma}^2_t|\boldsymbol{\sigma}^2_{t-1}) p(k_t|k_{t-1}) \\
&\quad \times p(\boldsymbol{\mu}_{1:k_{0:t-1},0:t-1}, \boldsymbol{\sigma}^2_{0:t-1}, k_{0:t-1}|\mathbf{d}_{1:t-1})
\end{aligned}$$

Hence, it follows that the unnormalised importance ratios w_t are given by:

$$w_t = \frac{p(\boldsymbol{\mu}_{1:k_{0:t},0:t}, \boldsymbol{\sigma}^2_{0:t}, k_{0:t}|\mathbf{d}_{1:t})}{q(\boldsymbol{\mu}_{1:k_t,t}, \boldsymbol{\sigma}^2_t, k_t|\boldsymbol{\theta}_{k_{0:t-1},0:t-1}, k_{0:t-1}, \mathbf{d}_{1:t})p(\boldsymbol{\mu}_{1:k_{0:t-1},0:t-1}, \boldsymbol{\sigma}^2_{0:t-1}, k_{0:t-1}|\mathbf{d}_{1:t-1})}$$
$$\propto p(\mathbf{y}_t|\boldsymbol{\mu}_{1:k_{0:t},0:t}, \boldsymbol{\sigma}^2_{0:t}, k_{0:t}, \mathbf{x}_t, \mathbf{d}_{1:t-1}) \tag{17}$$

That is, the importance ratios are the one-step-ahead densities (also known as the evidence or innovations) in the bank of Kalman filters. More precisely,

$$p(\mathbf{y}_t|\widehat{\boldsymbol{\mu}}^{(i)}_{1:\widehat{k}_{0:t},0:t}, \widehat{\boldsymbol{\sigma}}^{2(i)}_{0:t}, \widehat{k}^{(i)}_{0:t}, \mathbf{x}_t, \mathbf{d}_{1:t-1})$$
$$= \mathcal{N}\left(\mathbf{D}(\widehat{\boldsymbol{\mu}}^{(i)}_{1:\widehat{k}_t,t}, \mathbf{x}_t)\bar{\boldsymbol{\alpha}}^{(i)}_{1:\widehat{m}_t,t-1}, \widehat{\boldsymbol{\sigma}}^{(i)2}_t + \mathbf{D}(\widehat{\boldsymbol{\mu}}^{(i)}_{1:\widehat{k}_t,t}, \mathbf{x}_t)(P^{(i)}_{\widehat{k}_t,t-1} + \delta^2_{\boldsymbol{\alpha}}\mathbf{I}_{m_t})\mathbf{D}'(\widehat{\boldsymbol{\mu}}^{(i)}_{1:\widehat{k}_t,t}, \mathbf{x}_t)\right) \tag{18}$$

The selection step is done according to the residual resampling scheme.

4.2 MCMC steps

For our problem, the target distribution is defined as:

$$p\left(k_t, \boldsymbol{\mu}_{1:k_t,t}\middle| \mathbf{x}_{1:t}, \mathbf{y}_{1:t}, k_{1:t-1}, \boldsymbol{\mu}_{1:k_{t-1},1:t-1}, \boldsymbol{\sigma}^2_{1:c,1:t}\right) \tag{19}$$

That is, the past trajectories are kept fixed to reduce the computational requirements. Note, however, that the methodology is more general and that it is possible to sample from the joint distribution of the trajectories $p\left(k_{1:t}, \boldsymbol{\mu}_{1:k_{1:t},1:t}, \boldsymbol{\sigma}^2_{1:c,1:t}\middle| \mathbf{x}_{1:t}, \mathbf{y}_{1:t}\right)$. The target distribution is defined on a union of disconnected spaces, $\left\{\widetilde{k}_{t-1} - 1\right\} \times \boldsymbol{\Omega}_{\widetilde{k}_{t-1}-1} \cup \left\{\widetilde{k}_{t-1}\right\} \times \boldsymbol{\Omega}_{\widetilde{k}_{t-1}} \cup \left\{\widetilde{k}_{t-1} + 1\right\} \times \boldsymbol{\Omega}_{\widetilde{k}_{t-1}+1}$, and admits the following expression:

$$\begin{aligned}
& p\left(k_t, \boldsymbol{\mu}_{1:k_t,t}\middle| \mathbf{x}_{1:t}, \mathbf{y}_{1:t}, \widetilde{k}_{1:t-1}, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},1:t-1}, \boldsymbol{\sigma}^2_{1:c,1:t}\right) \\
= \; & \mathbb{I}_{\left\{\widetilde{k}_{t-1}+1\right\}}(k_t)\, p_1\left(\widetilde{k}_{t-1}+1, \boldsymbol{\mu}_{1:k_{t-1}+1,t}\middle| \mathbf{x}_{1:t}, \mathbf{y}_{1:t}, \widetilde{k}_{1:t-1}, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},1:t-1}, \boldsymbol{\sigma}^2_{1:c,1:t}\right) \\
& + \mathbb{I}_{\left\{\widetilde{k}_{t-1}\right\}}(k_t)\, p_2\left(\widetilde{k}_{t-1}, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}\middle| \mathbf{x}_{1:t}, \mathbf{y}_{1:t}, \widetilde{k}_{1:t-1}, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},1:t-1}, \boldsymbol{\sigma}^2_{1:c,1:t}\right) \\
& + \mathbb{I}_{\left\{\widetilde{k}_{t-1}-1\right\}}(k_t)\, p_3\left(\widetilde{k}_{t-1}-1, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1}-1,t}\middle| \mathbf{x}_{1:t}, \mathbf{y}_{1:t}, \widetilde{k}_{1:t-1}, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},1:t-1}, \boldsymbol{\sigma}^2_{1:c,1:t}\right)
\end{aligned} \tag{20}$$

To sample from this distribution over different models, we have to resort to reversible jump MCMC simulation. In particular, each particle is updated independently using reversible jump MCMC moves. The transition kernel is a mixture of different transition kernels. At each iteration, one of the candidate moves (birth, death or update) is randomly chosen[4]. The probabilities for choosing these moves are b_{k_t}, d_{k_t} and u_{k_t} respectively, such that $b_{k_t} + d_{k_t} + u_{k_t} = 1$ for all $1 \leq k_t \leq k_{\max}$. A move is performed if the algorithm accepts it. For $k_t = 1$ the death move is impossible, so that $d_0 \triangleq 0$. For $k = 0$ the death move is impossible, so that $d_0 \triangleq 0$. For $k = k_{\max}$, the birth is not allowed and therefore $b_{k_{\max}} \triangleq 0$. We set the transition model probabilities as follows $b_{k_t} = p(k_t + 1|k_t)$, $d_{k_t} = p(k_t - 1|k_t)$ and $b_{k_t} = p(k_t|k_t)$.

[4]Note that although we are choosing MCMC moves that allow the number of RBFs to change only by 1, it is possible to design more complex and general moves within this sampling framework.

The update move simply updates the value of the parameters, while the birth and death moves allow the number of RBFs to increase or decrease. To ensure reversibility, we have designed a birth$_1$/death$_1$ move to communicate between the spaces $\left\{\widetilde{k}_{t-1}\right\} \times \mathbf{\Omega}_{\widetilde{k}_{t-1}}$ and $\left\{\widetilde{k}_{t-1}+1\right\} \times \mathbf{\Omega}_{\widetilde{k}_{t-1}+1}$ and a birth$_2$/death$_2$ move to communicate between $\left\{\widetilde{k}_{t-1}\right\} \times \mathbf{\Omega}_{\widetilde{k}_{t-1}}$ and $\left\{\widetilde{k}_{t-1}-1\right\} \times \mathbf{\Omega}_{\widetilde{k}_{t-1}-1}$. For all particles, the Markov chain is initialised at $\widetilde{k}_t$ and it obeys the following transition kernel:

$$\mathcal{K} = u_k K_{update} \mathbb{I}_{\{\widetilde{k}_{t-1}-1,\widetilde{k}_{t-1},\widetilde{k}_{t-1}+1\}}(k_{t,j}) + b_k\{K_{birth_1}\mathbb{I}_{\{\widetilde{k}_{t-1}\}}(k_{t,j}) + K_{birth_2}\mathbb{I}_{\{\widetilde{k}_{t-1}-1\}}(k_{t,j})\}$$
$$+ d_k\{K_{death_2}\mathbb{I}_{\{\widetilde{k}_{t-1}\}}(k_{t,j}) + K_{death_1}\mathbb{I}_{\{\widetilde{k}_{t-1}+1\}}(k_{t,j})\} \quad (21)$$

where the subscript j denotes the j-th iteration of the Markov chain. The expression tells us that if we are in the state $\{\widetilde{k}_{t-1}^{(i)}\}$, we can either execute the update, birth$_1$ or death$_2$ moves. If, for example, the algorithm selects a birth move in the first iteration, then it has to be a birth$_1$ move. If this move is then accepted, at the next iteration, one can only accept the update or death$_1$ move. That is, the death$_1$ move is the reversible move for the birth$_1$ move and the death$_2$ move is the reversible move for the birth$_2$. The possible trajectories of the Markov chain are summarised in Figure 3. In practice, we will only run one iteration of the MCMC sampler. We shall now proceed to describe the various moves in more detail.

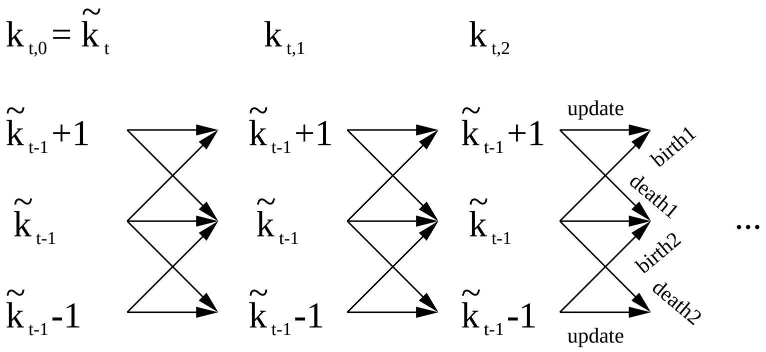

Figure 3: Graph showing the possible transitions of the Markov chain.

4.2.1 Update move

In this move, the dimension is kept fixed, *i.e.* $k_t^{(i)} = \widetilde{k}_t^{(i)}$. One merely samples the RBF centres; a difficult task because the distribution is nonlinear in these parameters. Here, they are sampled using the Metropolis-Hastings (MH) algorithm (Besag, Green, Hidgon and Mengersen 1995, Gilks et al. 1996, Tierney 1994). The target distribution for each particle, at time t, is the posterior distribution:

$$p(\widetilde{\boldsymbol{\mu}}_{1:\widetilde{k}_{0:t},0:t}^{(i)}, \widetilde{\boldsymbol{\sigma}}_{0:t}^{2(i)}, \widetilde{k}_{0:t}^{(i)} | \mathbf{d}_{1:t})$$

while the proposal distribution corresponds to:

$$\boldsymbol{\mu}_{1:k_t,t}^{(i)\star} | \widetilde{\boldsymbol{\mu}}_{1:\widetilde{k}_t,t-1}^{(i)}, \widetilde{k}_t^{(i)}, \mathbf{d}_{t-1} \sim p(\boldsymbol{\mu}_{1:k_t,t} | \widetilde{\boldsymbol{\mu}}_{1:\widetilde{k}_t,t-1}^{(i)}, \widetilde{k}_t^{(i)})$$

Hence, the acceptance probability for the update move is given by:

$$\mathcal{A}(\widetilde{\boldsymbol{\mu}}_{\widetilde{k}_t,t}, \boldsymbol{\mu}^{\star}_{\widetilde{k}_t,t}) = \min\left\{1, \prod_{j=1}^{c} \frac{p(\mathbf{y}_{j,t}|\boldsymbol{\mu}^{(i)\star}_{1:\widetilde{k}_t,t}, \widetilde{\boldsymbol{\sigma}}^{2(i)}_{j,t}, \widetilde{k}^{(i)}_t, \mathbf{x}_t, \mathbf{d}_{1:t-1})}{p(\mathbf{y}_{j,t}|\widetilde{\boldsymbol{\mu}}^{(i)}_{1:\widetilde{k}_t,t}, \widetilde{\boldsymbol{\sigma}}^{2(i)}_{j,t}, \widetilde{k}^{(i)}_t, \mathbf{x}_t, \mathbf{d}_{1:t-1})}\right\} \tag{22}$$

4.2.2 Birth$_1$ and death$_1$ moves

As mentioned earlier[5], the birth$_1$ and death$_1$ moves allow us to jump between the spaces $\left\{\widetilde{k}_{t-1}\right\} \times \boldsymbol{\Omega}_{\widetilde{k}_{t-1}}$ and $\left\{\widetilde{k}_{t-1}+1\right\} \times \boldsymbol{\Omega}_{\widetilde{k}_{t-1}+1}$. Now, suppose that the current state of the Markov chain is in $\left\{\widetilde{k}_t = \widetilde{k}_{t-1}\right\} \times \boldsymbol{\Omega}_{\widetilde{k}_t}$, the birth$_1$ move will then involve the following steps:

Birth$_1$ move

- Propose a new RBF centre at random from $p\left(\boldsymbol{\mu}^{\star}\right)$.
- Evaluate $\mathcal{A}_{birth_1} = \min\left\{1, r_{birth_1}\right\}$, see equation (24), and sample $u \sim \mathcal{U}_{[0,1]}$.
- If $u \leq \mathcal{A}_{birth_1}$ then the state of the Markov chain becomes $\left(\widetilde{k}_{t-1}+1, \left(\boldsymbol{\mu}^{\star}, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1}+1,t}\right)\right)$ else it remains equal to $\left(\widetilde{k}_{t-1}, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}\right)$.

■

The reverse move is described as follows. Suppose that the current state of the Markov chain is in $\left\{\widetilde{k}_t = \widetilde{k}_{t-1}+1\right\} \times \boldsymbol{\Omega}_{\widetilde{k}_t}$ the death$_1$ move will then involve the following steps:

Death$_1$ move

- Remove the basis that was added during the extension of the trajectory in the sampling stage.
- Evaluate $\mathcal{A}_{death_1} = \min\left\{1, r_{death_1}\right\}$, see equation (25), and sample $u \sim \mathcal{U}_{[0,1]}$.
- If $u \leq \mathcal{A}_{death_1}$ then the state of the Markov chain becomes $\left(\widetilde{k}_{t-1}, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}\right)$ else it remains equal to $\left(\widetilde{k}_{t-1}+1, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1}+1,t}\right)$.

■

The acceptance ratios for the proposed moves are deduced from the following expression (Green, 1995):

$$r = (\textit{posterior distribution ratio}) \times (\textit{proposal ratio}) \times (\textit{Jacobian}) \tag{23}$$

For the birth$_1$ move, we have:

$$\begin{aligned} r_{birth_1} &= \prod_{j=1}^{c} \frac{p\left(\mathbf{y}_{j,t}\middle|\,\boldsymbol{\mu}_{1:\widetilde{k}_{t-1}+1,t}, \widetilde{k}_{t-1}+1, \mathbf{x}_t, \mathbf{d}_{1:t-1}\right) p\left(\widetilde{k}_{t-1}+1\middle|\,\widetilde{k}_{t-1}\right)}{p\left(\mathbf{y}_{j,t}\middle|\,\boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}, \widetilde{k}_{t-1}, \mathbf{x}_t, \mathbf{d}_{1:t-1}\right) p\left(\widetilde{k}_{t-1}\middle|\,\widetilde{k}_{t-1}\right)} \\ &\quad \times \frac{p\left(\boldsymbol{\mu}^{\star}, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}\middle|\,\boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t-1}\right)}{p\left(\boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}\middle|\,\boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t-1}\right)} \frac{d_{\widetilde{k}_{t-1}+1}}{b_{\widetilde{k}_{t-1}} p\left(\boldsymbol{\mu}^{\star}\right)} \times J \end{aligned}$$

[5] To alleviate the notation, we shall drop the index $.^{(i)}$ when describing the birth and death moves.

where $p\left(\boldsymbol{\mu}^{\star}\right)$ is a Gaussian proposal distribution centered at the input data $\mathbf{x}_t$. The Jacobian J is clearly equal to 1. The above expression simplifies to:

$$r_{birth_1} = \prod_{j=1}^{c} \frac{p\left(\mathbf{y}_{j,t} \middle| \boldsymbol{\mu}_{1:\widetilde{k}_{t-1}+1,t}, \widetilde{k}_{t-1}+1, \mathbf{x}_t, \mathbf{d}_{1:t-1}\right) p\left(\widetilde{k}_{t-1}+1 \middle| \widetilde{k}_{t-1}\right)}{p\left(\mathbf{y}_{j,t} \middle| \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}, \widetilde{k}_{t-1}, \mathbf{x}_t, \mathbf{d}_{1:t-1}\right) p\left(\widetilde{k}_{t-1} \middle| \widetilde{k}_{t-1}\right)} \frac{d_{\widetilde{k}_{t-1}+1}}{b_{\widetilde{k}_{t-1}}} \tag{24}$$

The corresponding r_{death_1} is simply:

$$r_{death_1} = r_{birth_1}^{-1}$$

That is:

$$r_{death_1} = \prod_{j=1}^{c} \frac{p\left(\mathbf{y}_{j,t} \middle| \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}, \widetilde{k}_{t-1}, \mathbf{x}_t, \mathbf{d}_{1:t-1}\right) p\left(\widetilde{k}_{t-1} \middle| \widetilde{k}_{t-1}\right)}{p\left(\mathbf{y}_{j,t} \middle| \boldsymbol{\mu}_{1:\widetilde{k}_{t-1}+1,t}, \widetilde{k}_{t-1}+1, \mathbf{x}_t, \mathbf{d}_{1:t-1}\right) p\left(\widetilde{k}_{t-1}+1 \middle| \widetilde{k}_{t-1}\right)} \frac{b_{\widetilde{k}_{t-1}}}{d_{\widetilde{k}_{t-1}+1}} \tag{25}$$

4.2.3 Birth$_2$ and death$_2$ moves

The birth$_2$ and death$_2$ moves allows us to jump between the model spaces $\left\{\widetilde{k}_{t-1}\right\} \times \boldsymbol{\Omega}_{\widetilde{k}_{t-1}}$ and $\left\{\widetilde{k}_{t-1}-1\right\} \times \boldsymbol{\Omega}_{\widetilde{k}_{t-1}-1}$. Suppose that the current state of the Markov chain is in $\left\{\widetilde{k}_t = \widetilde{k}_{t-1}-1\right\} \times \boldsymbol{\Omega}_{\widetilde{k}_t}$ the birth$_2$ move will then involve the following steps:

Birth$_2$ move

- Propose a new RBF centre at random from $p\left(\boldsymbol{\mu}^{\star} \middle| \boldsymbol{\mu}_{deleted,t-1}\right)$, where $\boldsymbol{\mu}_{deleted,t-1}$ corresponds to the basis that was deleted during the extension of the trajectory.
- Evaluate $\mathcal{A}_{birth_2} = \min\left\{1, r_{birth_2}\right\}$, see equation (26), and sample $u \sim \mathcal{U}_{[0,1]}$.
- If $u \leq \mathcal{A}_{birth_2}$ then the state of the Markov chain becomes $\left(\widetilde{k}_{t-1}, \left(\boldsymbol{\mu}^{\star}, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}\right)\right)$ else it remains equal to $\left(\widetilde{k}_{t-1}-1, \mu_{1:\widetilde{k}_{t-1}-1,t}\right)$.

■

The reverse move is described as follows. Suppose that the current state of the Markov chain is in $\left\{\widetilde{k}_t = \widetilde{k}_{t-1}\right\} \times \boldsymbol{\Omega}_{\widetilde{k}_t}$, the death$_2$ move will then involve the following steps:

Death$_2$ move

- Choose the basis centre, to be deleted, at random among the $\widetilde{k}_{t-1}$ existing bases.
- Evaluate $\mathcal{A}_{death_2} = \min\left\{1, r_{death_2}\right\}$, see equation (27), and sample $u \sim \mathcal{U}_{[0,1]}$.
- If $u \leq \mathcal{A}_{death_2}$ then the state of the Markov chain becomes $\left(\widetilde{k}_{t-1}-1, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1}-1,t}\right)$ else it remains equal to $\left(\widetilde{k}_{t-1}, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}\right)$.

$\blacksquare$

Following the same steps as in the previous section, we obtain the following expression for r_{birth_2}:

$$r_{birth_2} = \prod_{j=1}^{c} \frac{p\left(\mathbf{y}_{j,t}\middle| \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}, \widetilde{k}_{t-1}, \mathbf{x}_t, \mathbf{d}_{1:t-1}\right) p\left(\widetilde{k}_{t-1}\middle| \widetilde{k}_{t-1}\right)}{p\left(\mathbf{y}_{j,t}\middle| \boldsymbol{\mu}_{1:\widetilde{k}_{t-1}-1,t}, \widetilde{k}_{t-1}-1, \mathbf{x}_t, \mathbf{d}_{1:t-1}\right) p\left(\widetilde{k}_{t-1}-1\middle| \widetilde{k}_{t-1}\right) \frac{1}{\widetilde{k}_{t-1}}}$$
$$\times \frac{p\left(\boldsymbol{\mu}^{\star}, \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}\middle| \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t-1}\right)}{p\left(\boldsymbol{\mu}_{1:\widetilde{k}_{t-1}-1,t}\middle| \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t-1}\right)} \frac{d_{\widetilde{k}_{t-1}} \frac{1}{\widetilde{k}_{t-1}}}{b_{\widetilde{k}_{t-1}-1} p\left(\boldsymbol{\mu}^{\star}\middle| \boldsymbol{\mu}_{deleted,t-1}\right)} \times J$$

which simplifies to:

$$r_{birth_2} = \prod_{j=1}^{c} \frac{p\left(\mathbf{y}_{j,t}\middle| \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}, \widetilde{k}_{t-1}, \mathbf{x}_t, \mathbf{d}_{1:t-1}\right) p\left(\widetilde{k}_{t-1}\middle| \widetilde{k}_{t-1}\right)}{p\left(\mathbf{y}_{j,t}\middle| \boldsymbol{\mu}_{1:\widetilde{k}_{t-1}-1,t}, \widetilde{k}_{t-1}-1, \mathbf{x}_t, \mathbf{d}_{1:t-1}\right) p\left(\widetilde{k}_{t-1}-1\middle| \widetilde{k}_{t-1}\right)} \frac{d_{\widetilde{k}_{t-1}}}{b_{\widetilde{k}_{t-1}-1}} \quad (26)$$

The corresponding death ratio is:

$$r_{death_2} = r_{birth_2}^{-1}$$

That is:

$$r_{death_2} = \prod_{j=1}^{c} \frac{p\left(\mathbf{y}_{j,t}\middle| \boldsymbol{\mu}_{1:\widetilde{k}_{t-1}-1,t}, \widetilde{k}_{t-1}-1, \mathbf{x}_t, \mathbf{d}_{1:t-1}\right) p\left(\widetilde{k}_{t-1}-1\middle| \widetilde{k}_{t-1}\right)}{p\left(\mathbf{y}_{j,t}\middle| \boldsymbol{\mu}_{1:\widetilde{k}_{t-1},t}, \widetilde{k}_{t-1}, \mathbf{x}_t, \mathbf{d}_{1:t-1}\right) p\left(\widetilde{k}_{t-1}\middle| \widetilde{k}_{t-1}\right)} \frac{b_{\widetilde{k}_{t-1}-1}}{d_{\widetilde{k}_{t-1}}} \quad (27)$$

5 Synthetic example

Data[6] was generated from the following univariate, nonlinear and time-varying function using 500 covariate points uniformly distributed on $[0,1]$:

$$y_t = \begin{cases} 4x_t - 2 + \left(2 + \frac{t}{150}\right)\exp(-15(x_t-0.3)^2) + 2\exp(-15(x_t-0.7)^2) + n_t & 1 \leq t \leq 250 \\ 4x_t - 2 + 2\exp(-15(x_t-0.5)^2) + n_t & 250 < t \leq 500 \end{cases}$$

where $n_t \sim \mathcal{N}(0,\sigma^2)$ with $\sigma^2 = 0.01$. Figure 4 shows the generated data. Note that the drift in the size of the Gaussian component at $x_t = 0.3$ is seriously affected by the noise corruption.

In the experiment, we selected Gaussian radial basis functions with the same variance as the Gaussian signal components. We chose $p(k_{t+1}|k_t)$ to have support $\{k_t-1, k_t, k_t+1\}$ and range $\{0.025, 0.95, 0.025\}$ and adopted a uniform prior for $\boldsymbol{\sigma}_0^2$ between 0 and 0.1. Since the domain of the data is the interval $[0,1]$, a multivariate Gaussian prior for $\boldsymbol{\mu}_{1:k_0,0}$ with mean 0.5 and variance 0.1 was adopted. The diffusion parameters $\delta_{\boldsymbol{\mu}}^2$, $\delta_{\boldsymbol{\alpha}}^2$ and $\delta_{\boldsymbol{\sigma}}^2$ were all set to 0.001. The parameter $k_{\max}$ was set to 20. An informative multivariate Gaussian prior, with mean 2 and variance 2, was used for $\boldsymbol{\alpha}_{1:m_0,0}$. The extended means $\overline{\boldsymbol{\alpha}}_{1:m_0,0}$ of the bank of Kalman filters were drawn from the same prior. Finally, the Kalman filter covariance terms $P_{k_0,0}$ were set to 1.

[6] The software for this experiment is available at the following web site: `http://www.cs.berkeley.edu/~jfgf/software.html`.

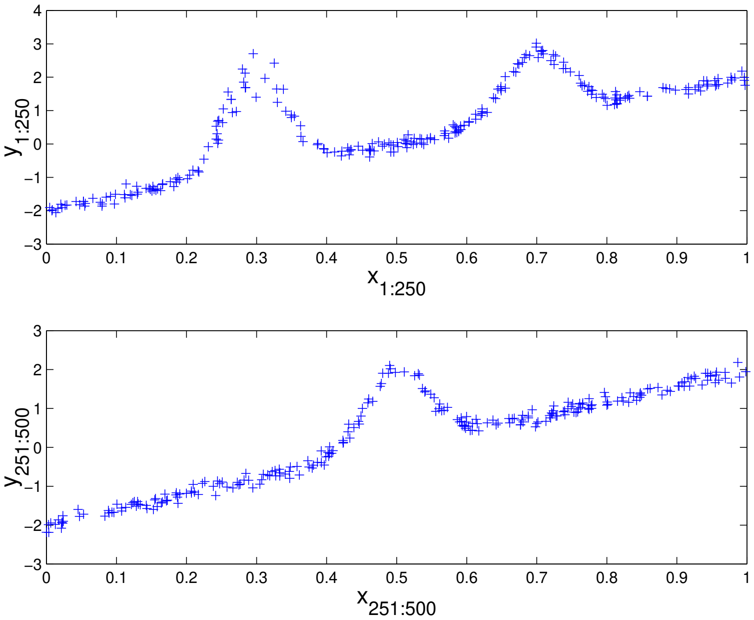

Figure 4: Data generated for the synthetic experiment.

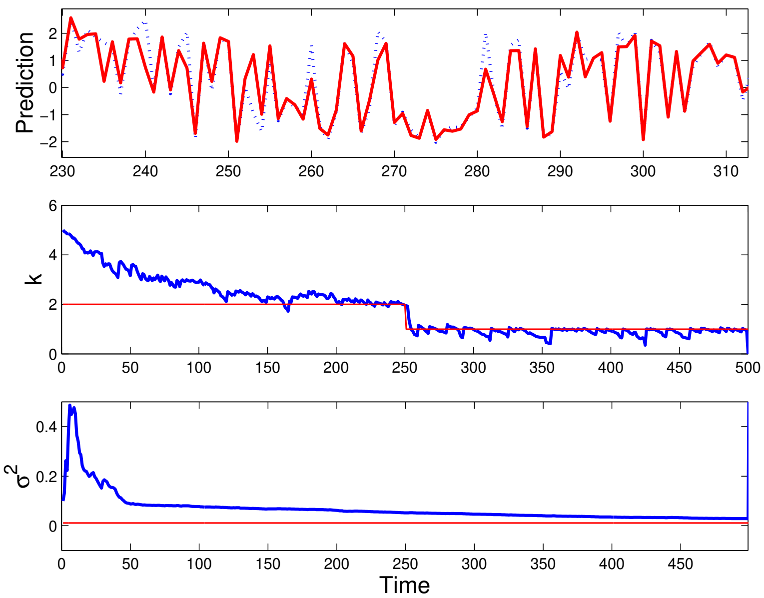

Figure 5: The top plot shows the one-step-ahead output predictions [—] and the true outputs [$\cdots$]. The middle and bottom plots show the true values and estimates of the model order and noise variance respectively.

The simulation results, using 1000 samples, are depicted in Figures 5 and 6. Figure 5 shows that the estimated model orders and noise variances (posterior means) converge to the right values. As a result, the output one-step-ahead predictions are very accurate. Figure 6 shows

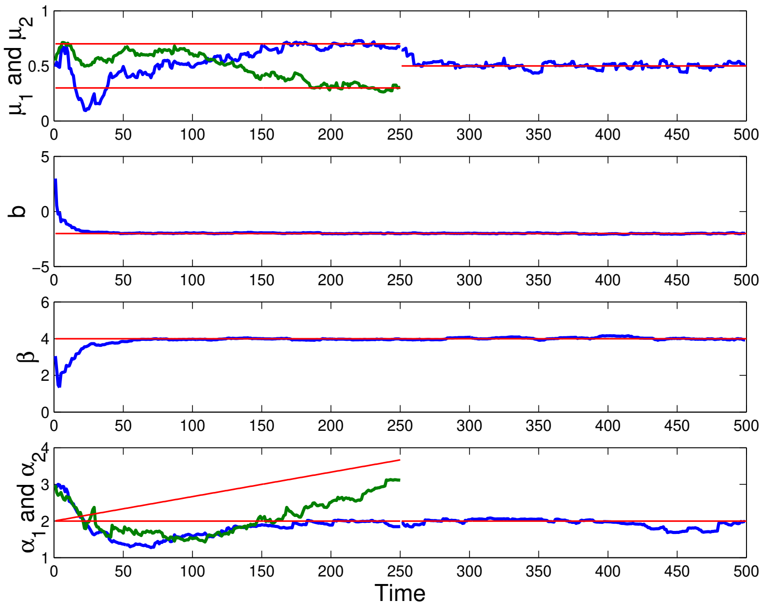

Figure 6: Estimated and true values of the basis centres, linear regression parameters and basis coefficients.

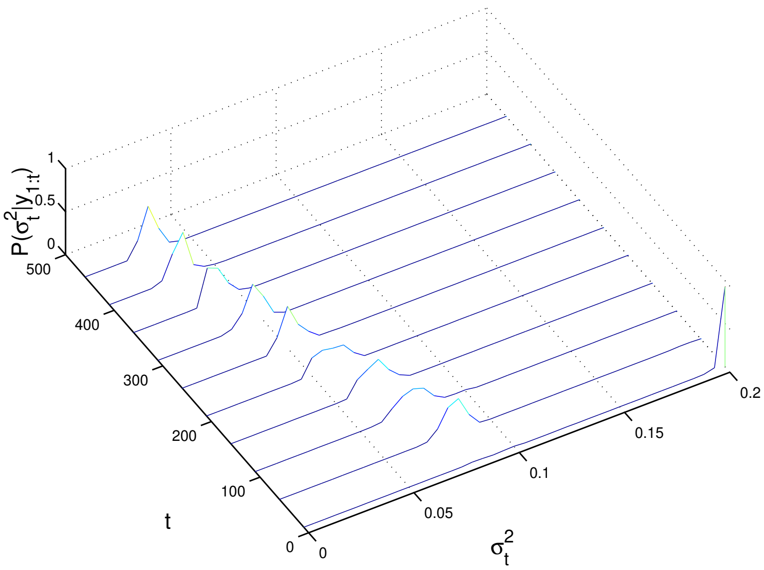

Figure 7: Probabilities of the estimated noise variance.

the estimates of the linear regression coefficients, basis centres and basis coefficients. These posterior mean estimates correspond to the centroids of the distributions computed using the main modes ($k = 2$ from $t = 1$ to $t = 500$ and $k = 1$ afterwards). Note how the integration of the coefficients leads to very fast and accurate estimates of the linear coefficients.

The algorithm performs extremely well when computing the linear regression parameters.

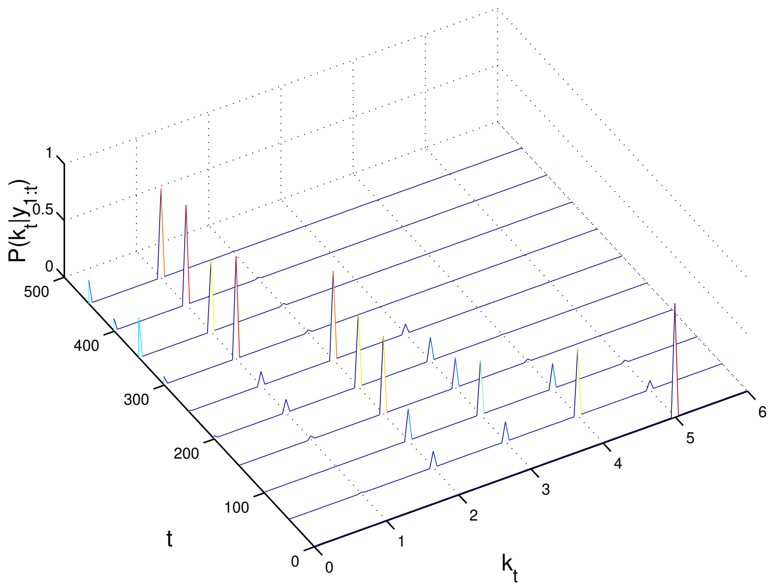

Figure 8: Probabilities of the estimated model order.

However, it is clear that to obtain similar results for the other quantities, it requires more samples. It should also be mentioned that given the amount of noise, it is very difficult to track the drift in the basis function amplitude. Figures 7 and 8 show the histograms of the estimated noise variance and model order.

6 An Application to Financial Time Series

Derivatives are financial instruments whose value depends on some basic underlying cash product, such as interest rates, equity indices, commodities, foreign exchange or bonds (Hull 1997). An option is a particular type of derivative that gives the holder the right to do something. For example, a call option allows the holder to buy a cash product, at a specified date in the future, for a price determined in advance. The price at which the option is exercised is known as the strike price, while the date at which the option lapses is referred to as the maturity time. Put options, on the other hand, allow the holder to sell the underlying cash product. Research results in this field seem to provide clear evidence that there is a nonlinear and non-stationary relation between the options' price and the cash products' price, maturity time, strike price, interest rates and variance of the returns on the cash product (volatility) (Black and Scholes 1973).

Following the work of (Hutchinson, Lo and Poggio 1994) and (Niranjan 1996), we employ neural networks to map the stock's price to the options' price. This approach, in conjunction with the use of volatility smiles and other strategic financial information, might provide an indication of whether an option in the market is either overpriced or underpriced. However, our main interest is to simply demonstrate the use of our method on a typical non-stationary, nonlinear time series problem. The experiment used a call option contract on the FTSE100 index (daily close prices from February 1994 to December 1994) to evaluate the pricing algorithms. For comparison purposes, we also show the results obtained using various training methods for multi-layer perceptrons (MLPs): see (de Freitas 1999) for more details concerning these methods. The MLP architecture consisted of 5 sigmoidal hidden units and a linear output neuron. The sampling algorithms all used 500 samples. We considered the following algorithms:

Trivial : This method simply involves using the current value of the option as the next prediction.

MLP-EKF : MLP trained with the EKF algorithm.

MLP-SMC-G : MLP trained with plain SMC (no MCMC step) under the assumptions of Gaussian process and measurement noises.

MLP-SMC-TT : MLP with t-distributed process and measurement noises.

RBF-SMC : RBF with thin plate spline basis functions trained with the model selection algorithm.

Method	RMS mean	RMS std. dev.
Trivial	0.0064	0
MLP-EKF	0.0037	0.0001
MLP-SMC-G	0.0055	0.0019
MLP-SMC-TT	0.0048	0.0007
RBF-SMC	0.0039	0.0002

Table 1: Means and standard deviations for the root mean square errors obtained over 50 simulations in the options pricing problem.

The results obtained, over 50 runs, are shown in Table 1. One can draw several conclusions from these results:

- Most of the methods will on the average perform better than the trivial approach.
- The simple EKF algorithm seems to outperform the sampling methods. This indicates that in problems that are high-dimensional and where the distribution of the data is unknown, we require to increase the number of samples or design more clever alternatives.
- Using heavy tailed distributions resulted in more accurate sampling methods. This is not surprising considering the nature of the data and standard results from importance sampling.

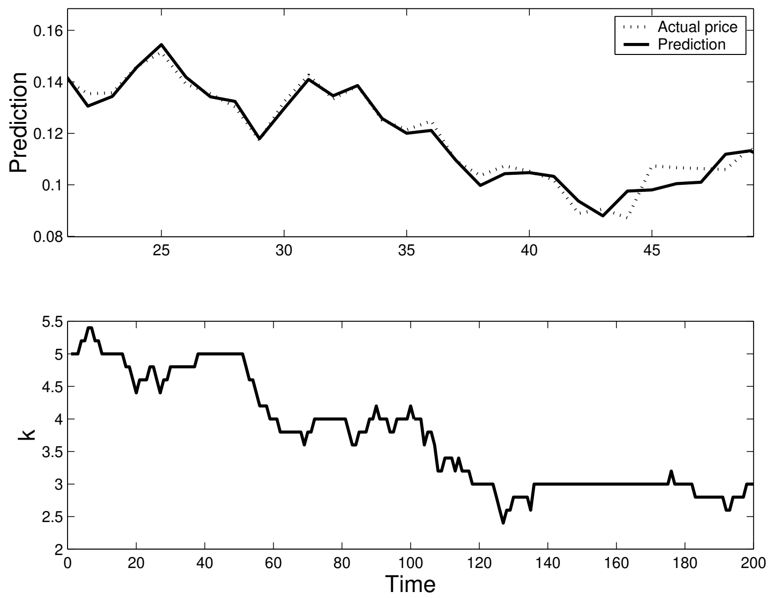

Figure 9: One step ahead predictions of the options prices (above) and posterior mean model order (below) for the RBF network on the options pricing problem.

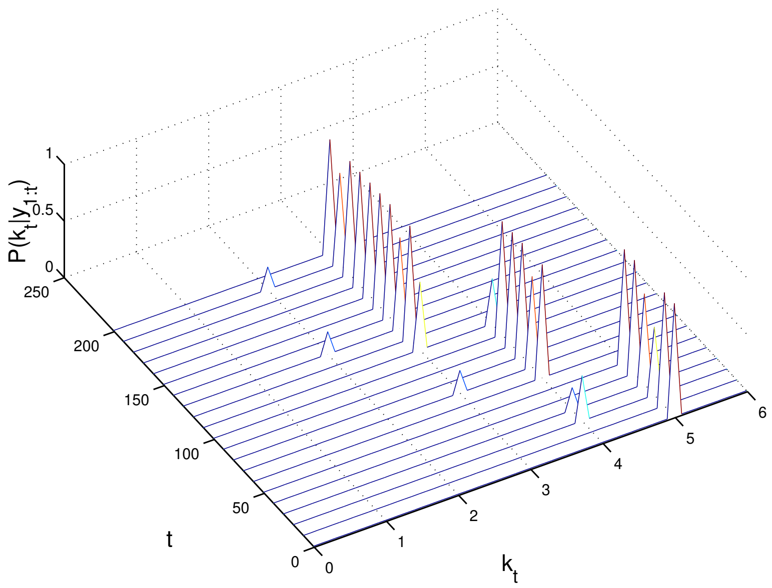

Figure 10: Model order (number of neurons) probabilities for the RBF network on the options pricing problem.

- The RBF model selection strategy performed very well (see also Figures 9 and 10). This demonstrates, that model selection and Rao-Blackwellisation are two good strategies for dealing with high-dimensional problems.

- Although the EKF results look better, we should not forget that the sampling methods are far more general. They allow us to compute noise variances, model order and other important quantities. In the case of the EKF, we had to tune by hand the number of basis functions.

7 Conclusions

We proposed a simulation-based approach to perform sequential Bayesian model selection. This strategy combines sequential Monte Carlo methods and reversible jump MCMC methods. It also makes use of variance reduction techniques. Its only drawback is that it is computationally intensive, yet it can be straightforwardly implemented on parallel computers. We demonstrated the effectiveness of the method on various complex joint estimation and model selection problems. There are many avenues for future research. These include adopting MCMC schemes with forgetting factors to estimate the noise variance, thereby avoiding the need for the diffusion parameter $\delta_{\boldsymbol{\sigma}}^2$; allowing for models with different types of basis functions; performing input variable selection; experimenting with other noise distributions and testing the algorithms on additional problems.

8 Acknowledgements

Christophe Andrieu is financially supported by AT&T Labs, Cambridge. Nando de Freitas is financially supported by two University of the Witwatersrand Merit Scholarships, a Foundation for Research Development Scholarship (South Africa), an ORS award and a Trinity College External Research Studentship (Cambridge). Arnaud Doucet is financially supported by an EPSRC grant, UK.

A Notation

- $\mathbf{A}_{i,j}$: entry of the matrix $\mathbf{A}$ in the i^{th} row and j^{th} column.
- $\mathbf{A}'$: transpose of matrix $\mathbf{A}$.
- $|\mathbf{A}|$: determinant of matrix $\mathbf{A}$.
- If $\mathbf{z} \triangleq (z_1, ..., z_{j-1}, z_j, z_{j+1}, ..., z_k)'$ then $\mathbf{z}_{-j} \triangleq (z_1, ..., z_{j-1}, z_{j+1}, ..., z_k)'$.
- $\mathbf{I}_n$: identity matrix of dimension $n \times n$.
- $\mathbb{I}_E(\mathbf{z})$: indicator function of the set E (1 if $\mathbf{z} \in E$, 0 otherwise).
- $\mathbf{z} \sim p(\mathbf{z})$: $\mathbf{z}$ is distributed according to $p(\mathbf{z})$.
- $\mathbf{z} | \mathbf{y} \sim p(\mathbf{z})$: the conditional distribution of $\mathbf{z}$ given $\mathbf{y}$ is $p(\mathbf{z})$.

Probability distribution	$\mathcal{F}$	$f_{\mathcal{F}}(\cdot)$
Gaussian	$\mathcal{N}(\mathbf{m}, \Sigma)$	$\lvert 2\pi\Sigma \rvert^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{z}-\mathbf{m})' \Sigma^{-1} (\mathbf{z}-\mathbf{m})\right)$
Uniform	$\mathcal{U}_A$	$\left[\int_A d\mathbf{z}\right]^{-1} \mathbb{I}_A(\mathbf{z})$

References

Andrieu, C., de Freitas, J. F. G. and Doucet, A. (1999a). Robust full Bayesian learning for neural networks, *Technical Report CUED/F-INFENG/TR 343*, Cambridge University Engineering Department.

Andrieu, C., de Freitas, J. F. G. and Doucet, A. (1999b). Sequential MCMC for Bayesian model selection, *IEEE Higher Order Statistics Workshop*, Ceasarea, Israel, pp. 130–134.

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*, Wiley Series in Applied Probability and Statistics.

Berzuini, C., Best, N. G., Gilks, W. R. and Larizza, C. (1997). Dynamic conditional independence models and Markov Chain Monte Carlo methods, *Journal of the American Statistical Association* **92**(440): 1403–1412.

Besag, J., Green, P. J., Hidgon, D. and Mengersen, K. (1995). Bayesian computation and stochastic systems, *Statistical Science* **10**(1): 3–66.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities, *Journal of Political Economy* **81**: 637–659.

Carpenter, J., Clifford, P. and Fearnhead, P. (1999). Building robust simulation-based filters for evolving data sets, Unpublished. Department of Statistics, Oxford University.

de Freitas, J. F. G. (1999). *Bayesian Methods for Neural Networks*, PhD thesis, Department of Engineering, Cambridge University, Cambridge, UK.

de Freitas, J. F. G., Niranjan, M., Gee, A. H. and Doucet, A. (2000). Sequential Monte Carlo methods to train neural network models, To appear in *Neural Computation*.

Djurić, P. M. (1998). Asymptotic MAP criteria for model selection, *IEEE Transactions on Signal Processing* **46**(10): 2726–2735.

Djurić, P. M. (1999). Monitoring and selection of dynamic models by Monte Carlo sampling, *IEEE Higher Order Statistics Workshop*, Ceasarea, Israel, pp. 191–194.

Doucet, A. (1998). On sequential simulation-based methods for Bayesian filtering, *Technical Report CUED/F-INFENG/TR 310*, Department of Engineering, Cambridge University.

Doucet, A. and Gordon, N. J. (1999). Simulation-based optimal filter for manoeuvring target tracking, *SPIE Signal and Data Processing of Small Targets*, Vol. SPIE 3809. To appear.

Doucet, A., Gordon, N. J. and Krishnamurthy, V. (1999). Particle filters for state estimation of jump Markov linear systems, *Technical Report CUED/F-INFENG/TR 359*, Cambridge University Engineering Department.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration, *Econometrica* **24**: 1317–1399.

Gilks, W. R. and Berzuini, C. (1998). Monte Carlo inference for dynamic Bayesian models, Unpublished. Medical Research Council, Cambridge, UK.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (eds) (1996). *Markov Chain Monte Carlo in Practice*, Chapman and Hall, Suffolk.

Girosi, F., Jones, M. and Poggio, T. (1995). Regularization theory and neural networks architectures, *Neural Computation* **7**(2): 219–269.

Gordon, N. J., Salmond, D. J. and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEE Proceedings-F* **140**(2): 107–113.

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

Holmes, C. C. and Mallick, B. K. (1998). Bayesian radial basis functions of variable dimension, *Neural Computation* **10**(5): 1217–1233.

Hull, J. C. (1997). *Options, Futures, and Other Derivative*, third edn, Prentice Hall.

Hutchinson, J. M., Lo, A. W. and Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks, *The Journal of Finance* **49**(3): 851–889.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*, Academic Press.

Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction theory, *Transactions of the ASME (Jounal of Basic Engineering)* **83D**: 95–108.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems, *Journal of the American Statistical Association* **93**: 1032–1044.

MacEachern, S. N., Clyde, M. and Liu, J. S. (1999). Sequential importance sampling for non-parametric Bayes models: the next generation, *Canadian Journal of Statistics* **27**: 251–267.

Niranjan, M. (1996). Sequential tracking in pricing financial options using model based and neural network approaches, *in* M. C. Mozer, M. I. Jordan and T. Petsche (eds), *Advances in Neural Information Processing Systems*, Vol. 8, pp. 960–966.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters, *Journal of the American Statistical Association* **94**(446): 590–599.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components, *Journal of the Royal Statistical Society B* **59**(4): 731–792.

Ripley, B. D. (1987). *Stochastic Simulation*, Wiley, New York.

Tierney, L. (1994). Markov chains for exploring posterior distributions, *The Annals of Statistics* **22**(4): 1701–1762.