Approximate forward-backward algorithm for a switching linear Gaussian model

Hugo Hammer[1,*]

Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway

Håkon Tjelmeland

Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway

Abstract

A hidden Markov model with two hidden layers is considered. The bottom layer is a Markov chain and given this the variables in the second hidden layer are assumed conditionally independent and Gaussian distributed. The observation process is Gaussian with mean values that are linear functions of the second hidden layer. The forward-backward algorithm is not directly feasible for this model as the recursions result in a mixture of Gaussian densities where the number of terms grows exponentially with the length of the Markov chain. By dropping the less important Gaussian terms an approximate forward-backward algorithm is defined. Thereby one gets a computationally feasible algorithm that generates samples from an approximation to the conditional distribution of the unobserved layers given the data. The approximate algorithm is also used as a proposal distribution in a Metropolis–Hastings setting, and this gives high acceptance rates and good convergence and mixing properties. The model considered is related to what is known as Switching linear dynamical systems. The proposed algorithm can in principle also be used for these models and the potential use of the algorithm is therefore large. In simulation examples the algorithm is used for the problem of seismic inversion. The simulations demonstrate the effectiveness and quality of the proposed approximate algorithm.

Keywords: approximation, forward-backward algorithm, hidden Markov model,

[*] Corresponding author

[1] Present address: Faculty of Engineering, Oslo University College, PO box 4 St. Olavs plass, 0130 Oslo, Norway. Phone number: +47 452 10 184, Fax: (+47) 22 45 32 05 (Att: Hugo Hammer)

1. Introduction

In a hidden Markov model (HMM) the observations are incomplete and noisy functions of an underlying unobserved process, where the latent process is assumed to be Markov. The goal is typically to restore the underlying process from the noisy observations and possibly also to estimate unknown parameters in both the latent and observation processes. For some HMM the underlying process consists of two layers, where Switching linear dynamical systems (SLDS) (Bar-Shalom and Li, 1998) is a typical example. In SLDS the bottom layer of the underlying unobserved process is a discrete Markov chain and conditioned on this the second unobserved layer is a Gaussian Markov process. Given the two unobserved layers the observations are assumed independent Gaussian. The mean vector and covariance matrix for the observed value at any time index are functions of the two unobserved states at the same time index. The goal is to restore the unobserved layers. SLDS have been used in many applications, e.g. fault detection in planetary rovers (Dearden and Clancy, 2002), speech recognition (Rosti and Gales, 2004), dancing of bees (Oh et al., 2005), econometrics (Kim, 1994) and machine learning (Lerner et al., 2000; Ghahramani and Hinton, 1998). Larsen et al. (2006) considers the problem of seismic inversion and a model similar to SLDS, but allow the observations to be a function of both past and future values of the hidden Gaussian process. The goal is again to restore the unobserved layers.

Recursive algorithms for HMM have successfully been used in many areas, see the discussions and references in MacDonald and Zucchini (1997), Künsch (2000), Scott (2002) and Cappé et al. (2005). Generalizations to hidden semi-Markov models are discussed in Guédon (2007) and Bulla et al. (2010). When an HMM only has one unobserved layer modelled as a discrete Markov chain, efficient recursive computations known as the forward-backward algorithm can be used. If the observations and the unobserved layer both are Gaussian, then the forward-backward algorithm corresponds to the famous Kalman filter. Forward-backward algorithms for HMM with two unobserved layers have been considered by Bar-Shalom and Li (1998), Barber (2006) and Zoeter and Heskes (2006). The forward-backward recursions can also be formulated for these models, but are not computationally feasible as they involve a mixture of Gaussian distributions where the number of terms grows exponentially with the length of the Markov chain. In the first set of references given above, approximate forward recursions are defined by substituting the Gaussian mixture by a single Gaussian term. Larsen et al. (2006) define approximate recursions by approximating the marginal distribution for the unobserved continuous process by a product of Gaussian densities.

We consider a model close to the model in Larsen et al. (2006) and separate from SLDS in that the observations can be a function of both past and future values. To get a computationally feasible algorithm, we construct an approximate forward-backward algorithm. In the forward recursions we propose to drop terms associated with small weights in the Gaussian mixture. Thus, our approximation is less dramatic than previous suggestions, but with a corresponding higher computational cost. Clearly, the quality of the approximation depends on the number and importance of the terms that are dropped. Using the approximate forward-backward algorithm as a proposal distribution in a Metropolis–Hastings algorithm (Smith and Roberts, 1993; Dellaportas and Roberts, 2003) we correct for the induced approximation. Moreover, we use the Metropolis–Hastings acceptance rate as a measure for the quality of the approximation. The proposed algorithm can in principle be used for SLDS and other similar models with the Markov property. The potential use of the algorithm is therefore large.

An alternative strategy to cope with HMM with two hidden layers are sequential Monte Carlo algorithms. Chen and Liu (2000) define a sequential Monte Carlo algorithm for what they call conditional dynamic linear models (CDLM). Our model defined in Section 2 can be rephrased to a CDLM by redefining the state variables. However, Chen and Liu (2000) only consider the filtering problem, whereas our focus is mainly the smoothing problem. Doucet et al. (2000) also use sequential Monte Carlo for a model similar to the one in Chen and Liu (2000), but the focus is again on filtering. Godsill et al. (2004) use sequential Monte Carlo to solve the smoothing problem for a state-space model. To generalize the procedure in Godsill et al. (2004) to handle a CDLM constitutes an alternative avenue for solving the problem we discuss.

In Hammer et al. (2010) a slightly modified variant of the simulation algorithm we present have successfully been used to invert real seismic data from an oilfield offshore Norway. Ulvmoen and Hammer (2009) also focus on the seismic inversion problem and use the procedure proposed in the present article to evaluate the quality of a much faster, but rougher approximation strategy. We also consider the problem of seismic inversion, but the focus here is to evaluate the efficiency of our proposed algorithm. In the seismic inversion setting, the Markov chain represents lithology-fluid classes along a vertical trace through the underground, the intermediate Gaussian layer represents elastic parameters of the rock along the same trace, and the observations are seismic data. The focus is to restore the unobserved Markov chain. Parameter estimation is clearly also of interest, but not considered here.

The paper is organized as follows. Section 2 introduces necessary notation for our hidden Markov model. In Section 3 we develop the approximate forward-backward algorithm. Section 4 gives a brief introduction to the seismic inversion application and explains how the hidden Markov model is the core part of the resulting model. We also evaluate the algorithm in simulation examples. Finally, Section 5 provides conclusions.

2. The switching linear Gaussian model

We represent (multivariate) Gaussian distributions in its canonical form, as this simplifies the forward-backward recursions. A Gaussian distribution with mean vector $\mu \in \mathbb{R}^r$ and covariance matrix $\Sigma \in \mathbb{R}^{r \times r}$ is then parameterized by the precision matrix $Q = \Sigma^{-1}$ and the vector $q = Q\mu$, and we use $\mathrm{N}(q, Q)$ to denote this distribution. The corresponding density we denote by $\mathrm{N}(u|q, Q)$, which reads

$$\mathrm{N}(u|q,Q) = \frac{\sqrt{|Q|}}{(2\pi)^{\frac{r}{2}}} \exp\left\{-\frac{1}{2}q^T Q^{-1} q\right\} \exp\left\{-\frac{1}{2}\left[u^T Q u - 2q^T u\right]\right\}. \tag{1}$$

Consider a three layer hidden Markov model $\{(x_i, y_i, z_i)\}_{i=1}^n$ as visualized in Figure 1, where $x_i \in \{1, \ldots, L\}$, $y_i = (y_{i1}, \ldots, y_{ir})^T \in \mathbb{R}^r$ and $z_i = (z_{i1}, \ldots, z_{is})^T \in \mathbb{R}^s$ for $i = 1, \ldots, n$. We call this a switching linear Gaussian model. We require the number of possible values for x_i, L, to be small. In the seismic data example in Section 4 we have $L = 4$. We let $x_{1:n} = (x_1, \ldots, x_n)^T$ be a stationary, aperiodic

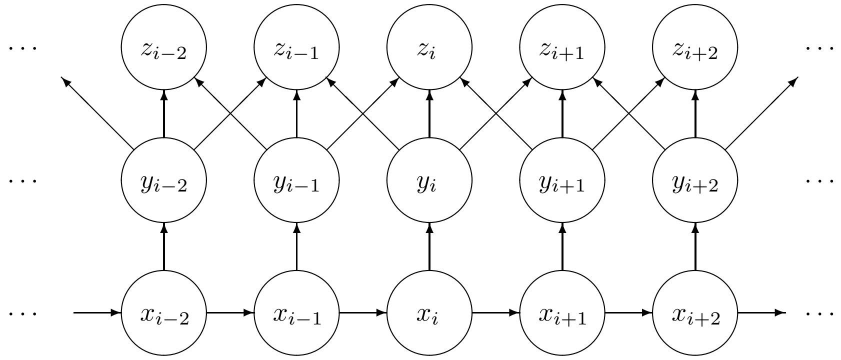

Figure 1: Directed acyclic graph (DAG) representation of the hidden Markov model discussed in Sections 2 and 3.

and ergodic Markov chain with transition matrix

$$\mathrm{P} = [p(x_i|x_{i-1})]_{x_{i-1},x_i=1}^{L}. \tag{2}$$

Thus, the marginal distribution of x_1, which we denote by $p(x_1)$, equals the limiting distribution induced by P. Conditioned on $x_{1:n}$ we assume the elements of $y_{1:n} = (y_1, \ldots, y_n)^T$ to be independent and Gaussian distributed, where the mean vector and precision matrix of y_i are (known) functions of x_i and denoted by $\mu(x_i)$ and $Q(x_i)$, respectively. Thus,

$$y_i|x_{1:n} \sim \mathrm{N}(q(x_i), Q(x_i)) \text{ , where } q(x_i) = Q(x_i)\mu(x_i). \tag{3}$$

Given $y_{1:n}$ we assume the elements of $z_{1:n} = (z_1, \ldots, z_n)^T$ to be independent and Gaussian, and the mean vector and precision matrix of z_i are $a_i^T y_{i-1} + b_i^T y_i + c_i^T y_{i+1}$ and R_i, respectively, i.e.

$$z_i|y_{1:n} \sim \mathrm{N}\left(A_i^T y_{i-1} + B_i^T y_i + C_i^T y_{i+1}, R_i\right), \tag{4}$$

where $A_i = a_i R_i$, $B_i = b_i R_i$ and $C_i = c_i R_i$. Note that we allow the coefficient matrices a_i, b_i and c_i to vary with i, and in particular we require $a_1 = c_n = 0$.

3. The forward-backward algorithm

In this section we define an approximate forward-backward algorithm for the model described in Section 2. We first derive the exact forward recursions. Starting with $\pi(x_{1:n}, y_{1:n}|z_{1:n})$ we integrate out y_i and x_i for $i = 1, \ldots, n$ in turn to get the distributions $\pi(x_{i:n}, y_{i+1:n}|z_{1:n})$ and $\pi(x_{i:n}, y_{i:n}|z_{1:n})$ for $i = 1, \ldots, n$. This is the basis for the backward simulation part of the algorithm, which sequentially for $i = n, \ldots, 1$ generates x_i from $\pi(x_i|x_{i+1:n}, y_{i+1:n}, z_{1:n}) \propto \pi(x_{i:n}, y_{i+1:n}|z_{1:n})$ and y_i from $\pi(y_i|x_{i:n}, y_{i+1:n}, z_{1:n}) \propto \pi(x_{i:n}, y_{i:n}|z_{1:n})$.

3.1. Forward integration

The conditional distribution of interest is $\pi(x_{1:n}, y_{1:n}|z_{1:n})$. However, to avoid notationally cumbersome special cases for $i = n-1$ and n we also introduce x_{n+1}, y_{n+1}, y_{n+2} and z_{n+1}. We make these auxiliary variables independent of the variables of interest by setting $p(x_{n+1}|x_n) = 1/L$ and also adopting (3) for $i = n+1$ and $n+2$ and (4) for $i = n+1$ with $A_{n+1} = B_{n+1} = C_{n+1} = 0$ and $R_{n+1} = I$. As a function of $x_{1:n}$ and $y_{1:n}$ we then have

$$\pi(x_{1:n}, y_{1:n}|z_{1:n}) \propto \pi(x_{1:n+1}, y_{1:n}, z_{1:n+1}|y_{n+1:n+2}) \propto p(x_1) \prod_{i=2}^{n+1} p(x_i|x_{i-1}) \cdot \\ \prod_{i=1}^{n} \mathrm{N}\left(y_i|q(x_i), Q(x_i)\right) \cdot \prod_{i=1}^{n+1} \mathrm{N}\left(z_i|A_i^T y_{i-1} + B_i^T y_i + C_i^T y_{i+1}, R_i\right). \tag{5}$$

To get a more compact notation in the development of the forward recursions we define

$$T_0(y_{1:2}) = \mathrm{N}(z_1|B_1^T y_1 + C_1^T y_2, R_1) \tag{6}$$

and

$$T_i(x_i, y_{i:i+2}) = \mathrm{N}(y_i|q(x_i), Q(x_i))\mathrm{N}(z_{i+1}|A_{i+1}^T y_i + B_{i+1}^T y_{i+1} + C_{i+1}^T y_{i+2}, R_{i+1}), \tag{7}$$

for $i = 1, \ldots, n$, so that

$$\pi(x_{1:n}, y_{1:n}|z_{1:n}) \propto p(x_1) T_0(y_{1:2}) \prod_{i=1}^{n} p(x_{i+1}|x_i) T_i(x_i, y_{i:i+2}). \tag{8}$$

Starting with this expression we integrate and sum out y_i and x_i for $i = 1, \ldots, n$ in turn and define $U_i(x_i, y_{i:i+1})$ and $V_i(x_i, y_{i+1:i+2})$ for $i = 1, \ldots, n$ so that the result becomes

$$\pi(x_{i:n}, y_{i:n}|z_{1:n}) \propto U_i(x_i, y_{i:i+1}) \prod_{j=i}^{n} p(x_{j+1}|x_j) T_j(x_j, y_{j:j+2}) \tag{9}$$

and

$$\pi(x_{i:n}, y_{i+1:n}|z_{1:n}) \propto V_i(x_i, y_{i+1:i+2}) p(x_{i+1}|x_i) \prod_{j=i+1}^{n} p(x_{j+1}|x_j) T_j(x_j, y_{j:j+2}). \tag{10}$$

The following theorem gives the relation between the U_i and V_i functions.

Theorem 1. *Consider the hidden Markov model defined in Section 2 and the notation introduced above. We then have*

$$U_1(x_1, y_{1:2}) = p(x_1)T_0(y_{1:2}), \tag{11}$$

and the recursions

$$U_{i+1}(x_{i+1}, y_{i+1:i+2}) = \sum_{x_i=1}^{L} V_i(x_i, y_{i+1:i+2})p(x_{i+1}|x_i),\ i = 1, \ldots, n-1 \tag{12}$$

and

$$V_i(x_i, y_{i+1:i+2}) = \int U_i(x_i, y_{i:i+1})T_i(x_i, y_{i:i+2})\, dy_i,\ i = 1, \ldots, n. \tag{13}$$

The theorem is proven as follows. By comparing (8) and (9) for $i = 1$ we get (11). Next, by summing out x_i in (10) and comparing with (9) we get (12). Finally integrating out y_i in (9) and comparing with (10) one gets (13).

In the following we use the notation

$$D_1 = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},\ D_3 = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I} \end{bmatrix},\ D_{12} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \text{ and } D_{23} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{14}$$

where $\mathbf{0}$ and $\mathbf{I}$ are an $r \times r$ matrix with all elements equal to zero and the r-dimensional identity matrix, respectively. Then the following theorem gives how to compute the U_i and V_i functions recursively.

Theorem 2. *Consider the hidden Markov model defined in Section 2 and the notation introduced above. We then have*

$$U_i(x_i, y_{i:i+1}) \propto \sum_{j=1}^{N_i} \gamma_{ij}(x_i)\, \mathrm{N}(y_{i:i+1}|g_{ij}, G_{ij}) \tag{15}$$

and

$$V_i(x_i, y_{i+1:i+2}) \propto \sum_{j=1}^{N_i} \kappa_{ij}(x_i)\, \mathrm{N}(y_{i+1:i+2}|k_{ij}(x_i), K_{ij}(x_i)) \tag{16}$$

for $i = 1, \ldots, n$, where $N_i = L^{i-1}$ and $\gamma_{ij}(x_i) \in \mathbb{R}$, $g_{ij} \in \mathbb{R}^{2r\times 1}$, $G_{ij} \in \mathbb{R}^{2r\times 2r}$, $\kappa_{ij}(x_i) \in \mathbb{R}$, $k_{ij}(x_i) \in \mathbb{R}^{2r\times 1}$ and $K_{ij}(x_i) \in \mathbb{R}^{2r\times 2r}$ can be computed recursively. Initial values $\gamma_{11}(x_1)$, g_{11} and G_{11} are

$$\gamma_{11}(x_1) = p(x_1),\ \ g_{11} = \begin{bmatrix} B_1 \\ C_1 \end{bmatrix} z_1,\ \ G_{11} = \begin{bmatrix} B_1 \\ C_1 \end{bmatrix} R_1^{-1} \left[B_1^T\ C_1^T\right]. \tag{17}$$

For $i = 1, \dots, n$ and $j = 1, \dots, N_i$ the $\kappa_{ij}(x_i)$, $k_{ij}(x_i)$ and $K_{ij}(x_i)$ can be obtained from $\gamma_{ij}(x_i)$, g_{ij} and G_{ij} by

$$\begin{aligned}
&\kappa_{ij}(x_i) = \gamma_{ij}(x_i)\sqrt{\frac{|G_{ij}| \cdot |Q(x_i)|}{|H_{ij}(x_i)|}} \cdot \\
&\exp\left\{-\frac{1}{2}\left[g_{ij}^T G_{ij}^{-1} g_{ij} + q(x_i)^T Q(x_i)^{-1} q(x_i) - h_{ij}(x_i)^T H_{ij}(x_i)^{-1} h_{ij}(x_i)\right]\right\}
\end{aligned} \tag{18}$$

$$k_{ij}(x_i) = K_{ij}(x_i) D_{23}^T H_{ij}(x_i)^{-1} h_{ij}(x_i) \tag{19}$$

and

$$K_{ij}(x_i) = \left(D_{23}^T H_{ij}(x_i)^{-1} D_{23}\right)^{-1}, \tag{20}$$

where

$$h_{ij}(x_i) = D_1 q(x_i) + D_{12} g_{ij} + \begin{bmatrix} A_{i+1}^T & B_{i+1}^T & C_{i+1}^T \end{bmatrix}^T z_{i+1} \tag{21}$$

and

$$\begin{aligned}
H_{ij}(x_i) =& D_1 Q(x_i) D_1^T + D_{12} G_{ij} D_{12}^T + \delta(i \geq n-1) D_3 D_3^T \\
&+ \begin{bmatrix} A_{i+1}^T & B_{i+1}^T & C_{i+1}^T \end{bmatrix}^T R_{i+1}^{-1} \begin{bmatrix} A_{i+1}^T & B_{i+1}^T & C_{i+1}^T \end{bmatrix},
\end{aligned} \tag{22}$$

where $\delta(\cdot)$ is the indicator function, i.e. $\delta(E) = 1$ when E is true and $\delta(E) = 0$ otherwise. Finally, for $i = 2, \dots, n$, $j = 1, \dots, N_{i-1}$ and $l = 1, \dots, L$

$$\gamma_{i,j+(l-1)N_{i-1}}(x_i) = p(x_i|l)\kappa_{i-1,j}(l), \tag{23}$$

$$g_{i,j+(l-1)N_{i-1}} = k_{i-1,j}(l) \tag{24}$$

and

$$G_{i,j+(l-1)N_{i-1}} = K_{i-1,j}(l). \tag{25}$$

The theorem is proved by induction. Reordering terms in $T_0(y_{1:2})$ straightforwardly gives (15) for $i = 1$ and initial values (17). Starting with (13), rearranging terms and using well known properties of the multivariate Gaussian distribution gives (16) and (18) through to (22). We represent Gaussian distributions in the canonical form and not by the mean vector and covariance matrix and this causes the somewhat unfamiliar expressions in (19) and (20).

The $D_3 D_3^T$ term that appears in (22) for $i = n-1$ and n ensures that the $H_{n-1,j}$ and H_{nj} matrices are invertible. The $D_3 D_3^T$ term does not influence the variables of interest, $x_{1:N}$ and $y_{1:n}$, only the auxiliary variables y_{n+1} and y_{n+2}. Finally, (12) straightforwardly gives (15) and (23) through to (25) by a reordering of the terms.

As the number of terms in (15) and (16) grows exponentially with i the recursive algorithm is computationally feasible only for small values of n. In the next section we propose to approximate $U_i(x_i, y_{i:i+1})$ and $V_i(x_i, y_{i+1:i+2})$ by ignoring less important terms.

3.2. Approximate forward integration algorithm

In this section we propose an approximate, but computationally feasible version of the recursions developed above. We first compute the (exact) representations of $U_1(x_1, y_{1:2})$ and $V_1(x_1, y_{2:3})$ as given in Theorem 2. The starting point for finding an approximation for $U_i(x_i, y_{i:i+1})$ and $V_i(x_i, y_{i+1:i+2})$ is an approximation of $V_{i-1}(x_{i-1}, y_{i:i+1})$ on the form

$$\widetilde{V}_{i-1}(x_{i-1}, y_{i:i+1}) \propto \sum_{j=1}^{\widetilde{N}_{i-1}(x_{i-1})} \widetilde{\kappa}_{i-1,j}(x_{i-1}) \mathrm{N}(y_{i:i+1} | \widetilde{k}_{i-1,j}(x_{i-1}), \widetilde{K}_{i-1,j}(x_{i-1})), \tag{26}$$

where we use tilde to distinguish approximate quantities from exact ones. The approximate representation is of the same form as (16), except that in (26) the number of terms may depend on the value of x_{i-1}. Of course, for $i = 2$ we use $\widetilde{V}_{i-1}(x_{i-1}, y_{i:i+1}) = V_{i-1}(x_{i-1}, y_{i:i+1})$. For $i > 2$ we define $\widetilde{V}_i(x_i, y_{i+1:i+2})$ from $\widetilde{V}_{i-1}(x_{i-1}, y_{i:i+1})$ in to steps. First we use the recursions in Theorem 2 to find an approximation $U_i^\star(x_i, y_{i:i+1})$ to $U_i(x_i, y_{i:i+1})$ and a first approximation $V_i^\star(x_i, y_{i+1:i+2})$ to $V_i(x_i, y_{i+1:i+2})$. Thereafter we drop the less important terms in $V_i^\star(x_i, y_{i+1:i+2})$ to get a final approximation $\widetilde{V}_i(x_i, y_{i+1:i+2})$. More precisely, we set

$$U_i^\star(x_i, y_{i:i+1}) \propto \sum_{j=1}^{N_i^\star} \gamma_{ij}^\star(x_i) \mathrm{N}(y_{i:i+1} | g_{ij}^\star, G_{ij}^\star) \tag{27}$$

and

$$V_i^\star(x_i, y_{i+1:i+2}) \propto \sum_{j=1}^{N_i^\star} \kappa_{ij}^\star(x_i) \mathrm{N}(y_{i+1:i+2} | k_{ij}^\star(x_i), K_{ij}^\star(x_i)), \tag{28}$$

where $N_i^\star = \sum_{l=1}^{L} \widetilde{N}_{i-1}(l)$. Corresponding to (23) through to (25), $\gamma_{ij}^\star(x_i)$, $g_{ij}^\star$ and $G_{ij}^\star$ are defined by

$$\gamma_{i,j+\sum_{t=1}^{l-1} \widetilde{N}_{i-1}(t)}^\star(x_i) = p(x_i|l)\widetilde{\kappa}_{i-1,j}(l), \tag{29}$$

$$g_{i,j+\sum_{t=1}^{l-1} \widetilde{N}_{i-1}(t)}^\star = \widetilde{k}_{i-1,j}(l) \tag{30}$$

and

$$G^{\star}_{i,j+\sum_{t=1}^{l-1}\widetilde{N}_{i-1}(t)} = \widetilde{K}_{i-1,j}(l), \tag{31}$$

for $i = 2,\ldots,n$, $j = 1,\ldots,\widetilde{N}_{i-1}(l)$ and $l = 1,\ldots,L$. Finally, $\kappa^{\star}_{ij}(x_i)$, $k^{\star}_{ij}(x_i)$ and $K^{\star}_{ij}(x_i)$ are defined by replacing $\kappa_{ij}(x_i)$, $k_{ij}(x_i)$, $K_{ij}(x_i)$, $\gamma_{ij}(x_i)$, g_{ij}, G_{ij} and N_i with corresponding starred quantities in (18) through to (22).

Which terms in (28) that are of less importance is not obvious as the terms are functions of $y_{i+1:i+2}$ which is still unspecified when the decision about what terms to drop has to be made. Natural strategies are either to maximize over or to integrate out $y_{i+1:i+2}$ before comparing the terms. Maximizing over $y_{i+1:i+2}$ is obtained by evaluating the Gaussian densities in (28) at their mean values. Thus, for a threshold value ε this gives that we should drop terms in (28) that have

$$\frac{\kappa^{\star}_{ij}(x_i)\mathrm{N}\left(\mu^{\star}_{ij}(x_i)\middle|\, k^{\star}_{ij}(x_i), K^{\star}_{ij}(x_i)\right)}{\max_{k=1,\ldots,N^{\star}_i(x_i)}\left\{\kappa^{\star}_{ik}(x_i)\mathrm{N}\left(\mu^{\star}_{ik}(x_i)\middle|\, k^{\star}_{ik}(x_i), K^{\star}_{ik}(x_i)\right)\right\}} < \varepsilon, \tag{32}$$

where $\mu^{\star}_{ij}(x_i) = K^{\star}_{ij}(x_i)^{-1}k^{\star}_{ij}(x_i)$. With the second strategy, integrating out y_{i+1} and y_{i+2}, only $\kappa^{\star}_{ij}(x_i)$ remains to compare. Thus, again for a given threshold ε, we drop all terms that corresponds to a $\kappa^{\star}_{ij}(x_i)$ that have

$$\frac{\kappa^{\star}_{ij}(x_i)}{\max_{k=1,\ldots,N^{\star}_i(x_i)}\left\{\kappa^{\star}_{ik}(x_i)\right\}} < \varepsilon. \tag{33}$$

In the simulation examples in Section 4 we adopt the first strategy, but we do not expect the second strategy to behave much differently. One should note that we decide what terms to drop separately for each possible value of x_i, and as a result the number of remaining terms, $\widetilde{N}_i(x_i)$, becomes a function of x_i.

Clearly, alternative term dropping strategies may be defined. First, one may use the term dropping step for $U^{\star}_i(x_i, y_{i:i+1})$ instead, but we do not expect this to make much difference. Second, instead of choosing a specific threshold value ε, one may fix the number of terms we want to keep and drop the necessary number of small terms. Thereby the memory requirements for running the algorithm will be known in advance, but the quality of the approximation may be more variable than with the strategy we have chosen.

3.3. Backward simulation

When the (exact or approximate) forward integration is done and necessary quantities stored in memory, backward simulation is straight forward. Here we give the necessary equations for the approximate, computational feasible algorithm. We initiate auxiliary variables $x_{n+1}, y_{n+1:n+2}$ and z_{n+1} with arbitrary values and sequentially for $i = n, \ldots, 1$ first simulate x_i from

$$\pi^\star(x_i|x_{i+1:n}, y_{i+1:n}, z_{1:n}) \propto V_i^\star(x_i, y_{i+1:i+2})p(x_{i+1}|x_i) \tag{34}$$

and then y_i from

$$\pi^\star(y_i|x_{i:n}, y_{i+1:n}, z_{1:n}) \propto U_i^\star(x_i, y_{i:i+1})T_i(x_i, y_{i:i+2}). \tag{35}$$

The first is a discrete distribution and the second a mixture of r-variate Gaussian densities, so both are easy to sample from. The resulting realization is thereby simulated from an approximation to the conditional distribution $\pi(x_{1:n}, y_{1:n}|z_{1:n})$,

$$\begin{aligned} &\pi^\star(x_{1:n}, y_{1:n}|z_{1:n}) = \\ &\prod_{i=1}^{n} \left[\pi^\star(x_i|x_{i+1:n+1}, y_{i+1:n+2}, z_{1:n+1})\pi^\star(y_i|x_{i:n+1}, y_{i+1:n+2}, z_{1:n+1})\right]. \end{aligned} \tag{36}$$

One should note that evaluating $\pi^\star(x_{1:n}, y_{1:n}|z_{1:n})$ is straight forward for a generated sample $(x_{1:n}, y_{1:n})$, but to do this correctly one must of course remember to include the normalizing constants in the two conditional distributions $\pi^\star(x_i|x_{i+1:n+1}, y_{i+1:n+2}, z_{1:n+1})$ and $\pi^\star(y_i|x_{i:n+1}, y_{i+1:n+2}, z_{1:n+1})$.

3.4. Simulation from the hidden Markov model

The error introduced by the approximation discussed above may be corrected for by adopting $\pi^\star(x_{1:n}, y_{1:n}|z_{1:n})$ as a proposal distribution in an independent proposal Metropolis–Hastings scheme. The resulting acceptance rate can then also be used as a measure for the quality of the approximation.

4. Simulation examples

We study the approximate forward-backward algorithm in a number of simulation exercises. We implement the algorithm in C++, where a list is used to store the Gaussian mixture. Each element in the list is a term in the Gaussian mixture. Using a list makes it is easy to remove terms with small weight following Section 3.2.

In the simulation examples we focus on the problem of seismic inversion from the petroleum industry. Our objective here is to demonstrate that our approximate algorithm is able to solve a problem of significant practical importance. Here we consider simulated data only, but in Hammer et al. (2010) we also apply it on real seismic data. Seismic inversion is the discipline of predicting lithology-fluid characteristics in a reservoir from seismic data. Numerous introductory books to seismic terminology and inversion exist, see for example Sheriff and Geldart (1995) and references therein.

Seismic data is created by an explosion which sends sound waves into the ground. Parts of the waves are reflected, returned upwards and observed by microphones (geo- or hydrophones). These observations are the basis for the seismic data. A forward model, describing what we observe for given lithology–fluid characteristics, is known from physics theory. In seismic inversion we are interested in the corresponding inverse problem.

The simulation example is organized as follows. In Section 4.1 we present the seismic model and in Section 4.2 we present our choices of parameters in the seismic model. Further in Section 4.3 we link the seismic model to the switching model in Section 2 and explain how we simulate efficiently by taking advantage of the approximate forward-backward algorithm in Section 3. Finally in Sections 4.4 and 4.5 we present simulation results.

4.1. Seismic model

Our forward model is similar to the ones in Buland et al. (2003) and Larsen et al. (2006). When dealing with seismic data depth is typically not referenced by distance, but time used by the sound wave from the surface to a location in the underground and back, called two way travel time. An important problem not considered here is how to convert travel times to depths. Following Buland et al. (2003) and Larsen et al. (2006) we discretize the travel time and formulate the problem in a Bayesian setting. Let $i = 1, \ldots, n$ denote n two way travel times along a vertical profile and let x_i denote the lithology-fluid class in location i. As prior for $x_{1:n} = (x_1, \ldots, x_n)^T$ we adopt a Markov chain as specified by (2). Assuming an isotropic and elastic medium, the material properties at a location i are uniquely defined by the P-wave velocity (α_i), S-wave velocity (β_i) and density (ρ_i) at that location. Let $y_i = (\ln \alpha_i, \ln \beta_i, \ln \rho_i)^T$. The distribution of $y_{1:n} = (y_1, \dots, y_n)^T$ given $x_{1:n}$ is based on a rock physics model (Avseth et al., 2005) and we assume a Gaussian distribution as specified by (3).

We consider seismic data for s offset values, or angles, $\theta_1, \dots, \theta_s$. For each depth location i and offset value θ_j a reflection coefficient r_{ij} results from $y_{1:n}$. For this we use what is known as a weak contrast approximation to the Zoeppritz equations (Aki and Richards, 1980; Buland and Omre, 2003) and get for $r_i = (r_{i1}, \dots, r_{is})^T$,

$$r_i = \Gamma \frac{y_{i+1} - y_{i-1}}{2} \text{ for } i = 2, \dots, n-1, \tag{37}$$

where

$$\Gamma = \begin{bmatrix} \gamma_\alpha(\theta_1) & \gamma_\alpha(\theta_2) & \cdots & \gamma_\alpha(\theta_s) \\ \gamma_\beta(\theta_1) & \gamma_\beta(\theta_2) & \cdots & \gamma_\beta(\theta_s) \\ \gamma_\rho(\theta_1) & \gamma_\rho(\theta_2) & \cdots & \gamma_\rho(\theta_s) \end{bmatrix} \quad , \quad \begin{array}{l} \gamma_\alpha(\theta) = \frac{1}{2}\left(1 + \tan^2(\theta)\right), \\ \gamma_\beta(\theta) = -4\overline{\beta/\alpha}^2 \sin^2(\theta), \\ \gamma_\rho(\theta) = \frac{1}{2}\left(1 - 4\overline{\beta/\alpha}^2 \sin^2(\theta)\right) \end{array} \tag{38}$$

and one has assumed the ratio β_i/α_i to have an approximately constant value $\overline{\beta/\alpha}$ in the reservoir. The difference in (37) is an approximation to a derivative in the corresponding continuous model. For $i = 1$ and n we correspondingly use forward and backward differences, respectively. Finally, seismic observation d_{ij} is obtained for each location i and offset θ_j through a convolution of the reflection coefficients,

$$d_{ij} = \sum_{u=-k}^{k} \omega_{uj} r_{i-u,j} + \varepsilon_{ij}, \tag{39}$$

where $\{\omega_{uj}\}_{u=-k}^{k}$ defines a wavelet for each offset θ_j and ε_{ij} is Gaussian observation noise. Similar to Buland and Omre (2003) we assume the main part of the noise to have a correlation structure corresponding to the wavelet. The argument for this is that both the signal and noise parts are the results of sound waves going through the (same) underground. More precisely, we set

$$\varepsilon_{ij} = \sum_{u=-k}^{k} w_{uj} \varepsilon^1_{i-u,j} + \varepsilon^2_{ij}, \tag{40}$$

where ε^1_{ij} and ε^2_{ij} are independent Gaussian white noise with $\text{Var}(\varepsilon^1_{ij}) = \sigma_1^2$ and $\text{Var}(\varepsilon^2_{ij}) = \sigma_2^2$.

4.2. Parameter values

Our base case parameter values are chosen to be realistic for the seismic inversion application and are based on the values adopted in Larsen et al. (2006). We have $L = 4$ classes for x_i, where $x_i = 1, 2$ and 3 represent gas-, oil- and brine (water) saturated sandstone, respectively, and $x_i = 4$ represents shale. Sandstone is porous and allows flow of gas, oil and water, whereas the shale porosity is negligible and thereby acts as a barrier to fluid flow. Our choice of transition matrix P is based on values used in Larsen et al. (2006), but we consider a coarser seismic resolution than done there. Numbering the nodes from bottom to top, we use

$$\mathrm{P} = \begin{bmatrix} 0.9441 & 0 & 0 & 0.0559 \\ 0.0431 & 0.9146 & 0 & 0.0424 \\ 0.0063 & 0.0230 & 0.9422 & 0.0284 \\ 0.0201 & 0.0202 & 0.1006 & 0.8591 \end{bmatrix}. \tag{41}$$

The zero elements are important in the seismic application as these represent known physical properties. Water has a higher density than oil, which again has a higher density than gas. Thus, water can not be above gas or oil and oil can not be above gas, unless separated by a non-porous shale layer. The corresponding marginal probabilities for x_i are $[0.24, 0.16, 0.38, 0.22]$.

As discussed in Section 4.1 we use $y_i \in \mathbb{R}^3$, where the three elements represent logarithms of P- and S-wave velocities and density, respectively. In Larsen et al. (2006) the distribution of $y_i|x_i$ is represented as empirical distributions given by a set of corresponding x_i and y_i values. We use the same set of (x_i, y_i) values to estimate mean vectors and covariance matrices for the four assumed Gaussian distributions. The resulting mean vectors are $\mu(1) = [8.052, 7.492, 7.688]^T$, $\mu(2) = [8.071, 7.472, 7.730]^T$, $\mu(3) = [8.121, 7.467, 7.746]^T$ and $\mu(4) = [8.166, 7.546, 7.846]^T$, and for each value of x_i, the diagonal and off-diagonal entries in the following matrices give corresponding standard deviations and correlations

$$\begin{bmatrix} 0.031 & 0.876 & 0.322 \\ 0.876 & 0.033 & 0.271 \\ 0.322 & 0.271 & 0.012 \end{bmatrix}, \quad \begin{bmatrix} 0.027 & 0.891 & 0.384 \\ 0.891 & 0.032 & 0.295 \\ 0.384 & 0.295 & 0.009 \end{bmatrix}, \tag{42}$$

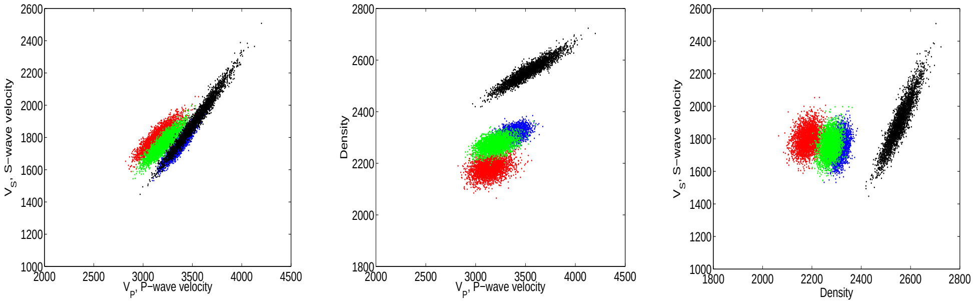

Figure 2: Scatter plots of P- and S-wave velocities and density of samples from the distribution adopted for $y_i|x_i$ in BC. Red, green, blue and black is used for gas-, oil- and brine-saturated sandstone and shale, respectively.

$$\begin{bmatrix} 0.022 & 0.912 & 0.453 \\ 0.912 & 0.032 & 0.317 \\ 0.453 & 0.315 & 0.008 \end{bmatrix}, \quad \begin{bmatrix} 0.044 & 0.982 & 0.935 \\ 0.982 & 0.068 & 0.917 \\ 0.935 & 0.917 & 0.015 \end{bmatrix}. \tag{43}$$

Figure 2 shows scatter plots of simulated P- and S-wave velocities and density according to the specified distributions. Here and in all the following we use red, green, blue and black for gas-, oil- and brine-saturated sandstone and shale, respectively. We observe that shale is well separated from the other classes and that gas-saturated sandstone is reasonably well separated from oil- and brine-saturated sandstone, whereas there is more overlap between oil- and brine-saturated sandstone.

To specify the model for $d_{1:n}|y_{1:n}$ we must give what offsets to use, the wavelet for each offset, and the variances σ_1^2 and σ_2^2. Still following Larsen et al. (2006) we use $s = 5$ offsets $\theta = 0°, 10°, 20°, 30°$ and $40°$ and adopt an offset independent Ricker wavelet,

$$w(u,\theta) = \left\{1 - 2\left(\pi\phi u\right)^2\right\} \exp\left\{-\left(\pi\phi u\right)^2\right\}, \; u = -k, \ldots, k, \tag{44}$$

with $\phi = 0.11$ and $k = 10$. For the error variances we use $\sigma_1^2 = 0.015^2$ and $\sigma_2^2 = \sigma_1^2/10^4$, which also corresponds to values used in Larsen et al. (2006).

The above defines our base case parameter set, which we refer to as BC. We define four more parameter sets, which are small modifications of BC. First the variances of the noise terms ε_{ij}^1 and ε_{ij}^2 are scaled to give higher and lower signal-to-noise ratios, and we denote these by LN and MN, respectively. We use $\sigma_1^2 = 0.0085^2$ and 0.026^2 for LN and MN, respectively, still keeping $\sigma_2^2 = \sigma_1^2/10^4$. The next two

Parameter set	BC	LN	MN	RL	RM
Signal-to-noise-ratio	1.34	2.22	0.53	1.30	1.36
ε	$2.5 \cdot 10^{-3}$	$2.5 \cdot 10^{-3}$	$1.8 \cdot 10^{-3}$	$1.7 \cdot 10^{-3}$	$1.7 \cdot 10^{-3}$
acceptance rate	0.44	0.43	0.37	0.46	0.42

Table 1: Signal-to-noise ratios, value used for the tuning parameter ε, and the resulting Metropolis–Hastings acceptance rate for the various parameter sets.

cases are obtained from BC by a scaling of the covariance matrices in the rock physics model, $y_i|x_i$. We define rock physics models with less variance (RL) and more variance (RM) by multiplying the covariance matrices defined by (42) and (43) by 1/2 and 2, respectively. We want the signal-to-noise ratios for RL and RM to be about the same as for BC and obtain this by modifying the noise variance σ_1^2, still keeping $\sigma_2^2 = \sigma_1^2/10^4$. When defining the signal-to-noise ratio we consider variability in $d_{1:n}$ originating from $x_{1:n}$ as signal and the remaining variability in $d_{1:n}$ as noise, see Hammer (2008) for the precise definition. This gave $\sigma_1^2 = 1.65 \cdot 10^{-2}$ and $1.10 \cdot 10^{-2}$ for RL and RM, respectively. Table 1 gives the resulting signal-to-noise ratios for all five parameter sets.

4.3. Simulating from the seismic model

We want to simulate $x_{1:n}$ and $y_{1:n}$ conditioned on $d_{1:n}$ in the seismic model defined above. A key point in the construction of an effective simulation algorithm is to take advantage of the approximate forward-backward algorithm defined in Section 3. We achieve this by introducing the additional variable $z_{1:n} = (z_1, \ldots, z_n)^T$, where $z_i = (z_{i1}, \ldots, z_{is})^T \in \mathbb{R}^s$ for $i = 1, \ldots, n$,

$$z_i = r_i + \varepsilon_i^1 \tag{45}$$

and $\varepsilon_i^1 = (\varepsilon_{i1}^1, \ldots, \varepsilon_{is}^1)^T$. The distributions for $x_{1:n}, y_{1:n}$ and $z_{1:n}$ is then as specified in Section 2. In (4) we have $A_i = -\Gamma R_i/2$, $B_i = \mathbf{0}$ and $C_i = \Gamma R_i/2$ for $i = 2, \ldots, n-1$, and using forward and backward difference at the boundaries $B_1 = -\Gamma R_1$, $C_1 = \Gamma R_1$, $A_n = -\Gamma R_n$ and $B_n = \Gamma R_n$. Finally we have $R_i = \sigma_1^{-2} I$ for $i = 1, \ldots, n$, where I is the identity matrix. Combining (39), (40) and (45) we get the relation between $z_{1:n}$ and $d_{1:n}$,

$$d_{ij}|z \sim \mathrm{N}\left(\sigma_2^{-2}\sum_{u=-k}^{k} w(u,\theta_j)\cdot z_{i-u,j}, \sigma_2^{-2}I\right). \tag{46}$$

We construct a Metropolis–Hastings algorithm (Smith and Roberts, 1993; Dellaportas and Roberts, 2003) consisting of two updates in each iteration. The first update is a block Gibbs update for $y_{1:n}$ and $z_{1:n}$. The joint full conditional for these are Gaussian and therefore easy to sample from. The second update in each iteration is a joint Metropolis–Hastings update for $x_{1:n}$ and $y_{1:n}$ by using the approximate forward-backward algorithm as the proposal distribution.

4.4. Evaluation of the approximate forward-backward algorithm

In this section we report the results for one Metropolis–Hastings run for each of the five parameter sets defined in Section 4.2 with $n = 100$ and use this to evaluate the performance of the proposed approximate forward integration algorithm. In each case we first simulate $x_{1:n}$, $y_{1:n}$, $z_{1:n}$ and $d_{1:n}$ according to the model specified in Sections 2 and 4.1 and thereafter use the algorithm proposed in Section 4.3 to sample from the resulting posterior distribution $\pi(x_{1:n}, y_{1:n}, z_{1:n}|d_{1:n})$. We evaluate the quality of the approximate algorithm by the acceptance rates and the convergence and mixing properties of the simulated Markov chains.

Following the optimal strategies for choice of Metropolis–Hastings tuning parameters found in Roberts et al. (1997) and Roberts and Rosenthal (1998) we find a value for our tuning parameter ε for each of the five parameter sets by aiming at a Metropolis–Hastings acceptance rate of about 0.4. It should be noted that our situation differs from what is discussed in the two references, so it is not clear that this is an optimal strategy in our situation. However, we found it to be a reasonable first try and it has worked satisfactory in all our runs. Table 1 reports both the ε values used and the resulting acceptance rates.

Figures 3 to 7 present simulation results for each of the five parameter sets. The upper rows show the simulated “true” values. Note that we use the same realization of $x_{1:n}$ in all cases to make comparison easier. The lower rows consist of three parts. To the left the “true” $x_{1:n}$ is replotted for easier comparison, in the middle each state of the Metropolis–Hastings run is plotted side by side, and the plots to the

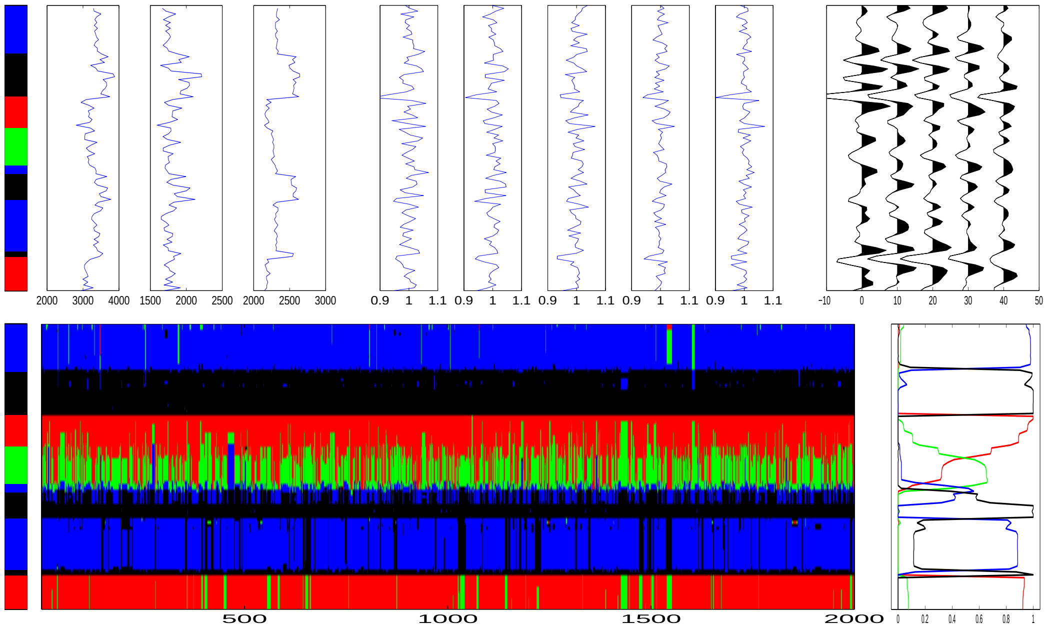

Figure 3: Simulation results for parameter set BC: The upper row shows, from left to right, the simulated "true" $x_{1:n}$, elastic parameters $\exp(y_{1:n})$, $z_{1:n}$ and $d_{1:n}$. The lower row contains posterior simulation results. From left to right, the lower row shows the true $x_{1:n}$ (replotted for easier comparison), each state of the Metropolis–Hastings run plotted side by side, and the resulting estimated marginal posterior probabilities.

right show the resulting estimated marginal probabilities for each node i. The runs shown are all initiated by setting all $x_i = 1$ and drawing $y_{1:n}$ and $z_{1:n}$ values from the corresponding full conditional. In all the runs the initial state is left within very few iterations and the burn-in phases are not even visible in the figures. We have also tried starting with all $x_i = 4$ and other initial values, but without experiencing any burn-in problems. The results clearly show that the approximate forward-backward algorithm gives a good approximation to the distribution of interest and produces very good mixing properties when used as a proposal distribution in a Metropolis–Hastings setting.

For the data shown in Figures 3 to 7 we have also tried the algorithm described in Section 3.4 for simulating from $\pi(x_{1:n}, y_{1:n}|z_{1:n})$. Again we tuned the value of ε as described above. The results indicated quite good convergence and mixing properties, but here some stickiness in the runs could be observed. As the algorithm is an independent proposal procedure, the latter should come as no surprise. Figure 8 shows the total number of Gaussian terms stored for each node i. Comparing the

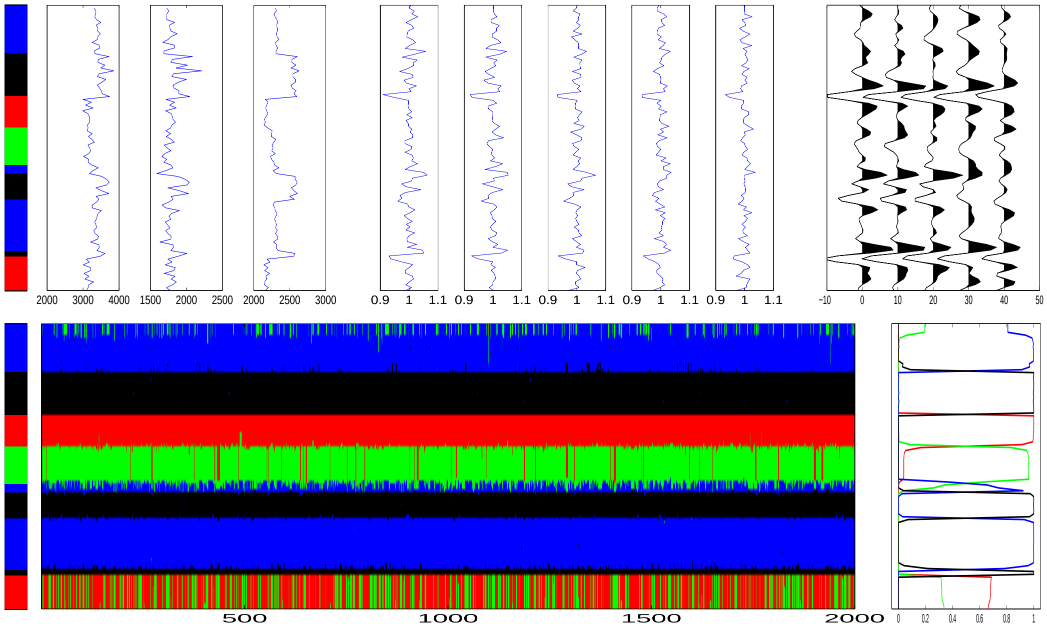

Figure 4: Simulation results for parameter set LN: See Figure 3 for an explanation of the different parts of the figure.

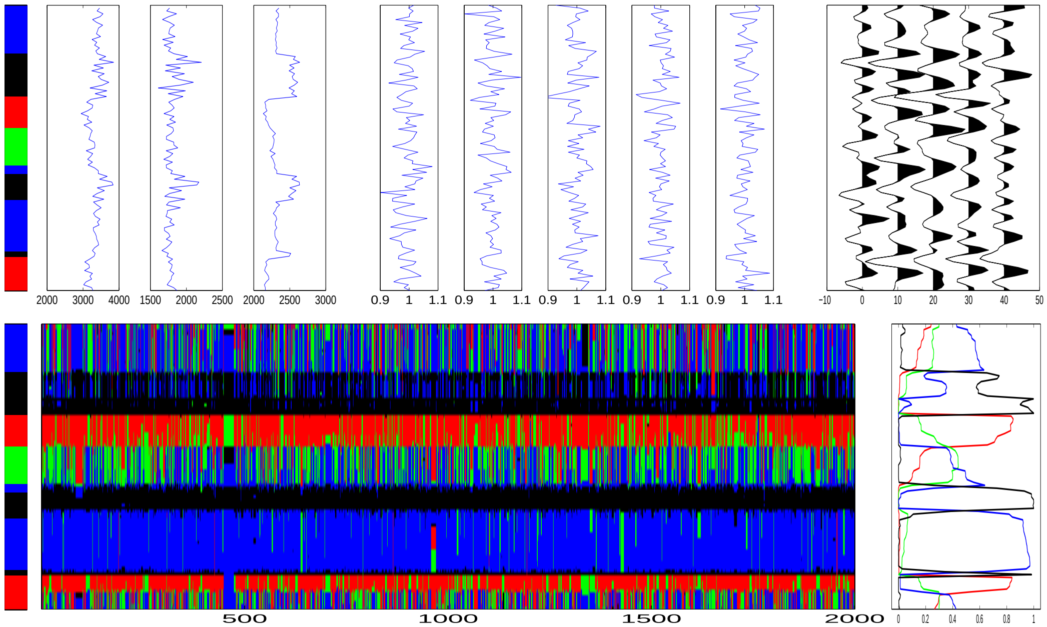

Figure 5: Simulation results for parameter set MN: See Figure 3 for an explanation of the different parts of the figure.

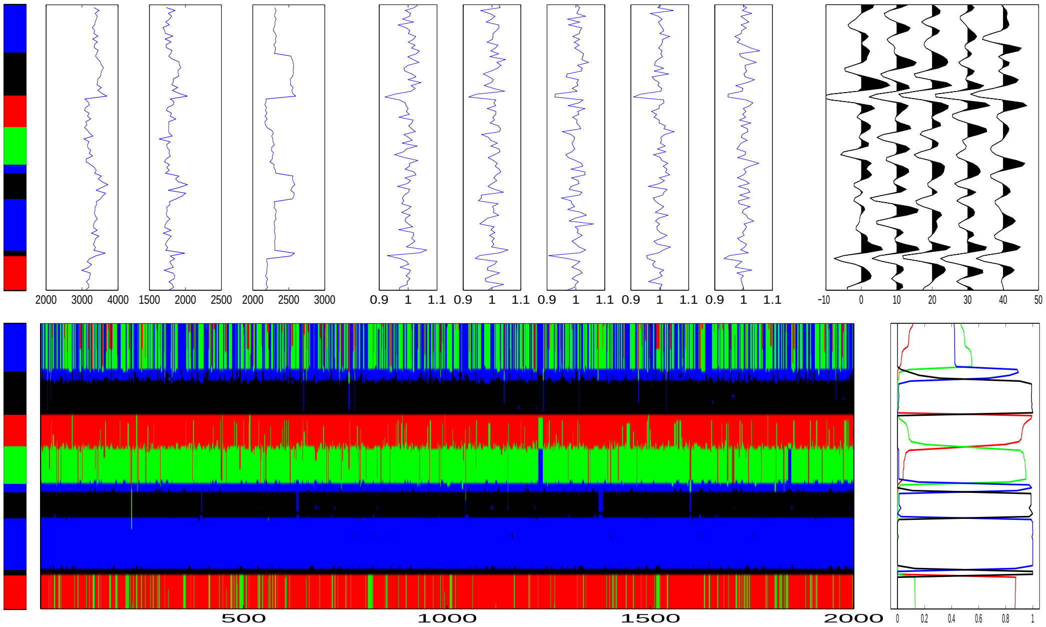

Figure 6: Simulation results for parameter set RL: See Figure 3 for an explanation of the different parts of the figure.

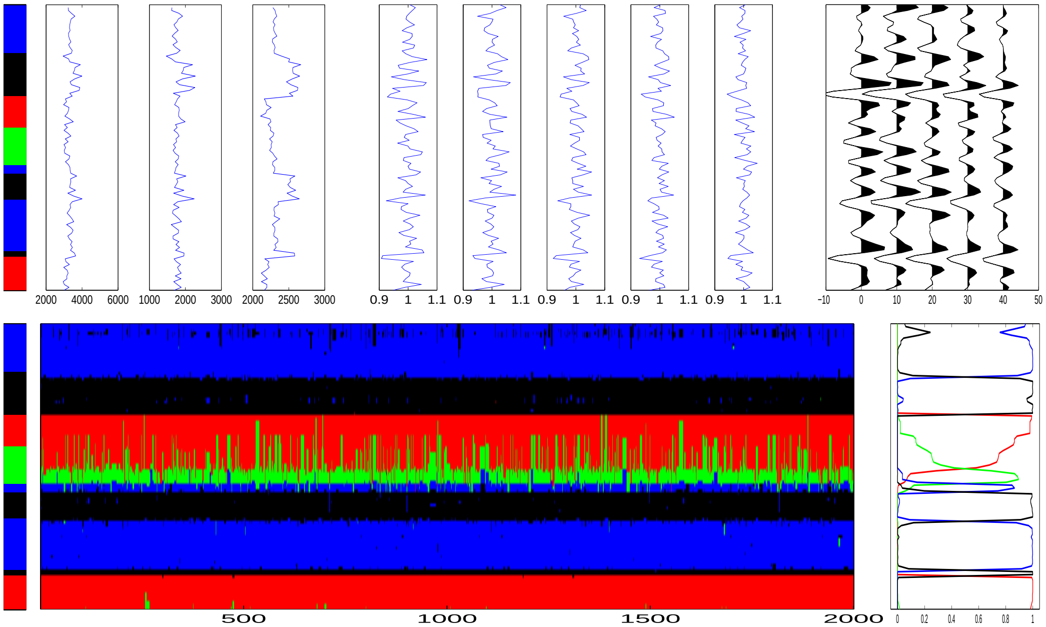

Figure 7: Simulation results for parameter set RM: See Figure 3 for an explanation of the different parts of the figure.

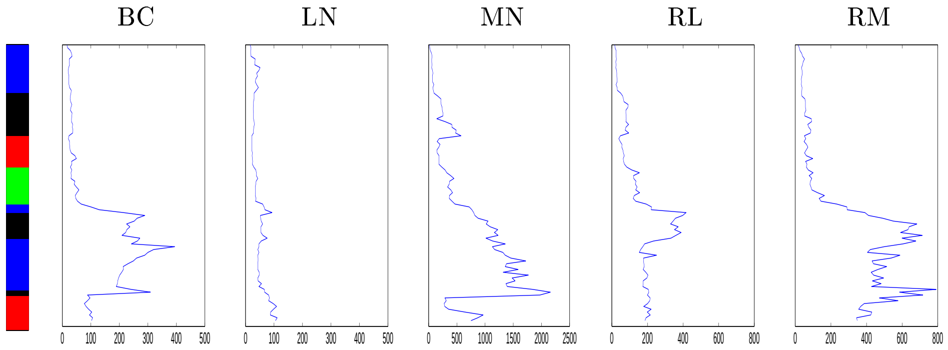

Figure 8: Simulation results for each of the five parameter sets: Number of terms stored in the approximate forward integration algorithm, $\sum_{x_{i-1}} \widetilde{N}_i(x_{i-1})$, as a function of node number i when simulating $x_{1:n}, y_{1:n}|z_{1:n}$. Note the different horizontal scales.

five models we observe that more terms are required for noisy models. This is also as one should expect, in the extreme case when the noise level goes to infinity the importance of the terms are only decided by the prior. Studying the five graphs separately we observe a growing trend in the number of terms, but it grows much slower than the exponential increase of all terms. We have also tried runs for $n > 100$ and the results there supports this conclusion. We also note the large fluctuations in the number of terms and that many of the abrupt changes happens close to where the true x_i changes value.

4.5. Inversion results

For each of our five parameter sets we repeated the simulation exercise shown in Figures 3 through to 7 ten times, now also randomizing over the value of the true $x_{1:n}$. For each run we used the MCMC results to estimate a confusion matrix $[c_{ij}]$ where c_{ij} is the average posterior probability for class j in nodes with true class i. Table 2 shows the resulting confusion matrices. As one would expect we observe a tendency that higher signal-to-noise ratios gives better predictions. However, LN and RM have comparable performances, demonstrating that the model noise in $y_{1:n}|x_{1:n}$ is informative. We also tried an alternative definition of the signal-to-noise ratio by considering variability originating from both $x_{1:n}$ and $y_{1:n}$ as signal, see the discussion in Hammer (2008), but this did not produce a better explanation of the figures in Table 2. Studying the individual confusion matrices we see that shale is

BC

	gas	oil	brine	shale
gas	0.668	0.324	0.004	0.004
oil	0.353	0.581	0.063	0.003
brine	0.005	0.064	0.891	0.040
shale	0.001	0.005	0.028	0.967

LN

	gas	oil	brine	shale
gas	0.837	0.163	0.000	0.000
oil	0.173	0.824	0.002	0.001
brine	0.000	0.136	0.833	0.093
shale	0.007	0.003	0.013	0.977

MN

	gas	oil	brine	shale
gas	0.611	0.279	0.102	0.009
oil	0.285	0.323	0.367	0.025
brine	0.047	0.115	0.740	0.098
shale	0.015	0.014	0.130	0.842

RL

	gas	oil	brine	shale
gas	0.787	0.197	0.016	0.001
oil	0.156	0.530	0.297	0.022
brine	0.006	0.059	0.918	0.017
shale	0.000	0.001	0.035	0.964

RM

	gas	oil	brine	shale
gas	0.881	0.112	0.005	0.001
oil	0.104	0.768	0.122	0.006
brine	0.001	0.033	0.924	0.042
shale	0.001	0.006	0.063	0.930

Table 2: Confusion matrices for the various parameter sets. In the tables element (i, j) is the estimated average posterior probability for class j for nodes where the true class is i.

most frequently classified correctly in all five cases. Considering Figure 2 this comes as no surprise. The misclassification between gas- and oil-saturated sandstone is significant in all cases. Oil-saturated sandstone is frequently misclassified to brine-saturated sandstone in MN, RL and RM, whereas such a misclassification is less frequent in BC and rare in LN.

5. Closing remarks

We have revisited the seismic inversion problem as a hidden Markov model with both continuous and discrete hidden variables. We split the model into a switching linear Gaussian model and a Gaussian linear model. To handle the first part computationally we propose an approximate forward-backward algorithm. In a number of simulation exercises we demonstrate the effectiveness of the approximation and how this makes inversion of the seismic model computationally feasible. The approximate algorithm includes a tuning parameter ε. To choose a value for ε one must compromise between memory usage and computation time on one side and approximation accuracy on the other. We have found no automatic way to set the value of ε, but our experience is that it is relatively easy to find a reasonable value by trial and error. What makes the choice of ε non-trivial is that it is used to decide what terms to drop in the forward recursions when information from the data is available from one side only. The importance of the various terms becomes available first when the following backward recursions have been done.

We think the inversion problem in the switching linear Gaussian model for seismic inversion is harder than the problems previously considered for switching linear dynamical systems (Zoeter and Heskes, 2006; Bar-Shalom and Li, 1998) and switching state space models (Barber, 2006). Within an interval with the same value for $x_{1:n}$, the seismic data does not depend on the mean value of the continuous variables. By the difference taken in (37) the mean value of the continuous variables influence the data only when the value of $x_{1:n}$ is changing. This induces larger posterior uncertainty in $x_{1:n}$ and it becomes correspondingly more important to have an approximate forward-backward algorithm that realistically represents this uncertainty. Thus, we think the importance of including more Gaussian terms in the forward recursion is larger for the seismic model than for the cases previously considered in Zoeter and Heskes (2006), Bar-Shalom and Li (1998) and Barber (2006).

We define an approximate forward recursion by dropping Gaussian terms with small weights. In the references mentioned above an approximation is obtained by taking a single Gaussian density that (approximately) represents the whole Gaussian mixture. It is clearly also possible to define an approximate forward recursion by following an intermediate strategy, finding groups of terms in the Gaussian mixture that have similar mean and covariance and approximate these by a single Gaussian term. However, the computational cost of finding what terms to merge is quadratic in the number of terms, whereas the cost of finding what terms to drop grows linearly with the number of terms. Thus, unless the number of Gaussian terms necessary to obtain a sufficiently good approximation is dramatically reduced when using the merging strategy, our simple dropping strategy is preferable. We have done a little experimentation with the merging strategy for our seismic inversion model, but without success. However, we think the merging strategy may have a potential if the continuous variable y_i is univariate.

The focus of the simulation examples of this paper is the computational problem associated with the hidden Markov seismic model. We have not considered inversion of real seismic data. To answer a real inversion problem one must also solve the associated parameter estimation problem. Preliminary experimentation with maximum likelihood estimation from simulated data indicates that it is not possible to estimate all the model parameters only from seismic data. Either one must adopt a Bayesian view with informative priors, or information about (at least some of) the parameters must be obtained from other data sources.

References

Aki, K. and Richards, P. G. (1980). *Quantitative seismology: Theory and methods*, W. H. Freeman and Company.

Avseth, P., Mukerji, T. and Mavko, G. (2005). *Quantitative seismic interpretation -Applying rock physics tools to reduce interpretation risk*, Cambridge University Press.

Bar-Shalom, Y. and Li, X.-R. (1998). *Estimation and Tracking: Principles, Techniques and Software*, Artech House, Norwood, MA.

Barber, D. (2006). Expectation correction for smoothed inference in switching linear dynamical systems, *Journal of machine learning research* **7**: 2515 – 2540.

Buland, A., Kolbjørnsen, O. and Omre, H. (2003). Rapid spatially coupled AVO inversion in the Fourier domain, *Geophysics* **68**: 824–836.

Buland, A. and Omre, H. (2003). Bayesian linearized AVO inversion, *Geophysics* **68**: 185–198.

Bulla, J., Bulla, I. and Nenadić, O. (2010). hsmm – an R package for analyzing hidden semi-Markov models, *Computational Statistics & Data Analysis* **54**: 611–619.

Cappé, O., Moulines, E. and Rydén, T. (2005). *Inference in Hidden Markov Models*, Springer.

Chen, R. and Liu, J. S. (2000). Mixture Kalman filters, *Journal of the Royal Statistical Society, Series B* **62**: 493–508.

Dearden, R. and Clancy, D. (2002). Particle filters for real-time fault detection in planetary rovers, *Proceedings of the 13th International Workshop on Principles of Diagnosis (DX02)*, pp. 1–6.

Dellaportas, P. and Roberts, G. O. (2003). An introduction to MCMC, *in* J. Møller (ed.), *Spatial Statistics and Computational Methods*, number 173 in *Lecture Notes in Statistics*, Springer, Berlin, pp. 1–41.

Doucet, A., Godsill, S. and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering, *Statistics and Computing* **10**: 197–208.

Ghahramani, Z. and Hinton, G. E. (1998). Variational learning for switching state-space models, *Neural Computation* **12**: 963–996.

Godsill, S. J., Doucet, A. and West, M. (2004). Monte Carlo smoothing for nonlinear time series, *Journal of the American Statistical Association* **99**: 156–168.

Guédon, Y. (2007). Exploring the state sequence space for hidden Markov and semi-Markov chains, *Computational Statistics & Data Analysis* **51**: 2379–2409.

Hammer, H., Kolbjørnsen, O., Tjelmeland, H. and Buland, A. (2010). Lithology and fluid prediction from prestack seismic data using a Bayesian model with Markov process prior, *Technical report*, Norwegian Computing Center. In preparation.

Hammer, H. L. (2008). *Topics in stochastic simulation, with an application to seismic inversion*, PhD thesis, Norwegian University of Science and Technology. Thesis number 73:2008.

Kim, C. J. (1994). Dynamic linear models with Markovswitching, *Journal of Econometrics* **60**.

Künsch, H. (2000). State space models and hidden Markov models, *in* O. Barndorff-Nielsen, D. Cox and C. Klüppelberg (eds), *Complex Stochastic Systems*, number 87 in *Monographs on Statsitics and Applied Probability*, Chapman & Hall/CRC.

Larsen, A. L., Ulvmoen, M., Omre, H. and Buland, A. (2006). Bayesian lithology/fluid prediction and simulation on the basis of a Markov-chain prior model, *Geophysics* **71 issue 5**: R69–R78.

Lerner, U., Parr, R., Koller, D. and Biswas, G. (2000). Bayesian fault detection and diagnosis in dynamic systems, *In Proc. AAAI*, pp. 531–537.

MacDonald, I. and Zucchini, W. (1997). *Hidden Markov and Other Models for Discrete-Valued Time Series*, Chapman and Hall.

Oh, S. M., Rehg, J. M., Balch, T. and Dallaert, F. (2005). Data-driven mcmc for learning and inference in switching linear dynamic systems, *Proc. 20th National Conference on Artificial Intelligence (AAAI-2005), Pittsburgh, PA.*

Roberts, G. O., Gelman, A. and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms, *Annals of Applied Probability* **7**: 110–120.

Roberts, G. O. and Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions, *Journal of the Royal Statistical Society. Series B* **60**: 255–268.

Rosti, A.-V. and Gales, M. (2004). Rao-blackwellised Gibbs sampling for switching linear dynamical systems, *In Intl. Conf. Acoust., Speech, and Signal Proc. (ICASSP)* **1**: 809–812.

Scott, A. L. (2002). Bayesian methods for hidden Markov models: Recursive compution in the 21st century, *Journal of the American Statistical Association* **97**: 337–351.

Sheriff, R. E. and Geldart, L. P. (1995). *Exploration Seismology*, Cambridge University Press.

Smith, A. and Roberts, G. O. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods (with discussion), *Journal of the Royal Statistical Society, Series B* **55**: 2–24.

Ulvmoen, M. and Hammer, H. (2009). Bayesian lithology/fluid inversion – comparison of two algorithms, *Computational Geosciences* **14**: 357–367.

Zoeter, O. and Heskes, T. (2006). Deterministic approximate inference techniques for conditionally Gaussian state space models, *Statistics and Computing* **16**: 279–292.