Lecture notes

HMM analysis of general state-space models

Martin W. Pedersen

Abstract

These notes explain how to use a hidden Markov model (HMM) approach for analysing possibly highly nonlinear time series data in a state-space formulation. The text introduces the general state-space model and gives an overview of other methods for filtering and smoothing ranging from the simple linear and Gaussian case to the fully general case. A discretization of the state-space is instrumental to the HMM approach and the choice of discretization is therefore discussed. The filter and smoothing recursions for the hidden Markov model are presented and applied to the standard benchmark model known from the literature on nonlinear time series. Finally, as an example, the parameters of a stochastic volatility model are estimated with maximum likelihood and the results are compared with an Monte Carlo based estimation procedure.

1 Introduction

State-space models (SSMs) cover the broad range of time series models where the aim is to estimate the state of an unobservable (and in our case) random process $\{C_t\}$ with $t \in \{1, 2, \ldots, T\}$ from an observed data set $\{X_t\}$. In the general state-space formulation, the dynamics of the state, C_t, are governed by the system (or process) equation

$$C_t = a(C_{t-1}, t, u_t) \tag{1}$$

and the link from the unobserved state to the observed data x_t is described by the observation equation

$$X_t = b(C_t, t, v_t), \tag{2}$$

where u_t and v_t are random disturbances (or errors) with known probability density functions (pdfs) $p(c_t|c_{t-1})$ and $p(x_t|c_t)$. This formulation implies that C_t is a Markov process since future values only depend on the current state and are independent of the past.

The filtering steps related to this SSM are the state update (time update)

$$p(c_t|\mathcal{X}_{t-1}) = \int p(c_{t-1}|\mathcal{X}_{t-1}) p(c_t|c_{t-1})\, dc_{t-1}, \tag{3}$$

and the Bayes' update (data update)

$$p(c_t|\mathcal{X}_t) = p(c_t|\mathcal{X}_{t-1})\frac{p(x_t|c_t)}{p(x_t|\mathcal{X}_{t-1})} \tag{4}$$

where $\mathcal{X}_t = \mathbf{X}^{(t)} = \mathbf{x}^{(t)}$. Equation (3) marginalizes the joint density $p(c_t, c_{t-1}|\mathcal{X}_{t-1})$ using the relation $p(c_t|c_{t-1}, \mathcal{X}_{t-1}) = p(c_t|c_{t-1})$ by an argument of conditional independence between c_t and $\mathcal{X}_{t-1}$ given c_{t-1}. The data update in (4) applies Bayes' rule.

The main challenge of the recursions is to compute the integral (3) which rarely has a closed form expression for nonlinear pdf's and must therefore be approximated either numerically (deterministically) or empirically (stochastically).

1.1 Previous filtering approaches

The integral (3) occurs in some variant in any filtering problem and ways to obtain its solution (or some sort of approximation thereof) is therefore an often studied topic in modern time series analysis.

Kalman derived the analytical solution for the case where $p(c_t|\mathcal{X}_t)$ and $p(c_t|c_{t-1})$ are Gaussian and the system and observation dynamics are linear (Kalman, 1960). In this case the filtered and smoothed densities are Gaussian as well and therefore fully represented by their mean and variance.

Much work has focused on extending the ordinary Kalman filter to deal with nonlinear (but unimodal) distributions resulting in the development of the extended Kalman filter (first order Taylor accuracy) (Jazwinski, 1970; Welch and Bishop, 1995), a third-order moment filter (Wiberg and DeWolf, 1993) and the unscented Kalman filter (third order Taylor accuracy) (Julier and Uhlmann, 1997). These methods, however, are stil limited to parametric representations of the probability densities and are for this reason mostly appropriate for mild nonlinearities. Another variant is the ensemble Kalman filter (Evensen, 2003) which by analysing randomly perturbed versions of the data generates an ensemble of possible filter solutions. The ensemble Kalman filter is often used for very complex systems where the cost of computing the system dynamics can be considerable e.g. ocean modelling, meteorology etc.

The Gaussian sum filter (Alspach and Sorenson, 1972) represents the posterior distribution as a sum of multiple Gaussian distributions and therefore enables closed form expressions of (3) to be obtained even for multimodal distributions. This approach applies a Kalman filter for each term in the Gaussian sum. As expected, for highly nonlinear models, this filter performs

well when compared to the extended Kalman filter, but like many other nonlinear filtering approaches it is computationally demanding and may even diverge if the number of Gaussian terms is too small.

Markov chain Monte Carlo (MCMC) (Gilks and Spiegelhalter, 1996) is a method for simulating random numbers from a probability distribution and is particularly useful for high-dimensional distributions. Simulating the high-dimensional posterior distribution of an SSM, $p(c_1, \ldots, c_T|\mathcal{X}_T)$, is therefore a task well suited for MCMC. By generating multiple outcomes the posterior distribution can be approximated empirically arbitrarily well without applying any restrictions to the general model (1) and (2). For strongly nonlinear distributions such as multimodal distributions the performance of the MCMC routine may depend on the initialization and the number of MCMC iterations. A major issue with MCMC is deciding when to stop further simulation i.e. to assess the time when the chain has converged. Diagnostics can be obtained regarding the quality of the estimation, but for highly nonlinear models reliable diagnostics are not always available.

Another simulation based method is the sequential Monte Carlo (SMC) approach (or more popularly the "particle filter"). The method approximates the densities $p(c_t|\mathcal{X}_t)$ empirically by sequential simulation of outcomes (particles) from a proposal density weighted by taking the observation y_t into account. The initial attempts to apply this scheme in practice failed because the weights of the particles degenerated rapidly i.e. the weights became very small for all but a few particles. Gordon et al. (1993) resolved this problem by introducing a resampling step in the iterations such that all resampled particles had equal weights. Now, a method for filtering general nonlinear time series was available that was claimed to avoid the curse of dimensionality. This claim, however, does not hold in general as pointed out by Daum and Co (2005).

Kitagawa (1996) presented the theory of parameter estimation and smoothing for SMC methods. For estimation quality and reliability, the most important aspect of the SMC method is the proposal distribution which propagates the particles from one time step to the next. The most natural choice of proposal distribution is to use the process model itself i.e. the density $p(c_t|c_{t-1})$. At times where the behaviour of the next observation departs from what is predicted by the proposal distribution (i.e. an outlier) the number of useful particles may be reduced and the density approximation which holds asymptotically may be invalid. Even when the number of effective particles is replenished in the resampling step, all the remaining particles may stem from only a few "mother" particles and are therefore unlikely to meaningfully approximate the true density. Owing to this fact much work has been invested to develop better proposal distributions which,

among many other approaches, has resulted in the auxiliary particle filter Pitt and Shephard (1999). A broad overview of SMC methods can be found in the review paper Cappé et al. (2007).

Another general approach to nonlinear filtering is the one initially described by Bucy and Senne (1971) which uses a point mass representation of the probability densities. The integration problem is viewed as a problem of solving the partial differential equation that describes the time evolution of the state probability density (Kolmogorov's forward equation). At the time of publication, however, the computational requirements of the approach severely limited the practical use of the method, but in turn required the authors to discuss ways of mitigating this difficulty. They considered "floating" or adaptive finite difference grids which improved performance significantly although not to the extent that the curse of dimensionality could be overcome.

The method was "reinvented" by Kitagawa (1987) in a modern version which, in addition to filtering, also presented smoothing, parameter estimation and model selection via Akaike's Information Criterion (AIC). In Kitagawa's approach the probability densities were represented by piecewise linear functions on a finite number of intervals thus approximating (3) by a sum of a finite number of integrals. The results of the Kitagawa (1987) publication lead to a supplementary discussion (see *Journal of the American Statistical Association*, vol. 82, no. 400, 1987, p. 1041-1063) where many questions and much criticism were raised about the method. The majority of the critique was directed towards the computational requirements (and thereby limited practical use) of the method as compared to existing methods for nonlinear time series analysis in particular for high dimensional state-spaces. Kitagawa's reply to this was that the specialized nature of the alternative methods prohibited model selection thus leading to less flexible methods as compared to the more general nonlinear modelling framework he presented. Many authors have recognized that for state-spaces of dimension higher than four, point mass approaches (or similar discretization based approaches) become impractical.

1.2 This text

The method presented in the present text is fundamentally very similar to the point-mass approach (Kitagawa, 1987), but formulated in the framework of hidden Markov models (HMMs) (Zucchini and MacDonald, 2009). HMMs require that the state-space is finite and discrete. The SSM is defined on a continuous infinite state-space and does therefore not fit immediately

into the HMM framework, however in most cases a solution on a discretized version of the continuous state-space will suffice as an approximation. The advantage of this approach is that complex models become very easy to handle with respect to filtering, smoothing, parameter estimation and optimal state estimation (MAP estimation of the state). As with other discretization methods it is limited to low-dimensional state-vectors.

The text is constructed as follows: Section 2 explains the method and presents the equations regarding filtering, smoothing and parameter estimation. Section 3 illustrates the method by considering a few examples that are classical nonlinear problems in time series analysis. Section 4 discusses the presented method as compared to existing techniques and with respect to computational requirements.

2 Method

Assume that the state-space is partitioned (typically uniformly) into m parts such that $\{\Omega_i : i \in I\}$, where $I = \{1, 2, \ldots, m\}$. The probability distribution of the state given the observations $\mathcal{X}_t$ is $\Pr(C_t \in \Omega_i | \mathcal{X}_t) = \phi_t(i)$ which are collected in the row vector $\boldsymbol{\phi}_t$ (see p. 46 in Zucchini and MacDonald (2009)). The transition probability of jumping from Ω_i to Ω_j is

$$\gamma_{ij}(t) = \Pr(C_{t+1} \in \Omega_j | C_t \in \Omega_i) = \int_{\Omega_j} p_{C_{t+1}|C_t}(c_{t+1}|C_t \in \Omega_i)\, dc_{t+1}. \quad (5)$$

For a one-dimensional problem Ω_i is an interval on the line, in two dimensions Ω_i is an area and analogously for higher dimensions. Note that the $m \times m$ probability transition matrix $\mathbf{\Gamma}_t = \{\gamma_{ij}(t)\}$ is not homogeneous, i.e. the transition probabilities may change as a function of t as indicated by (1).

2.1 Choosing the discretization

It is not immediately obvious how the discretization of the state-space should be constructed as it depends on the problem at hand. However, some guide lines can be given at least for simple problems. We have two different cases: online analysis and offline analysis.

For online analysis we are interested in estimating the present and sometimes also in predicting the future, but we are less interested in the past. For this reason it usually advantageous to apply adaptive gridding of the state-space that follows the data such that computational resources are focused in ares of interest (this is not possible for offline analysis since adaptive gridding

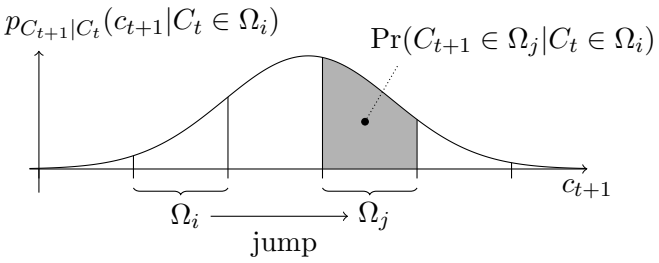

Figure 1: Probability of a jump (transition) from the state Ω_i to the state Ω_j in the time interval from t to $t+1$ in an HMM. The shaded area corresponds to the integral in (5).

complicates matters of smoothing, parameter estimation and MAP state estimation). Previous work concerning adaptive gridding includes Bucy and Senne (1971); Challa and Bar-Shalom (2000); Šimandl et al. (2006). We will not consider online analysis further in this text.

For offline analysis the whole data series is available before the analysis is started, and we can define the grid using information from the data. In the offline case we are typically interested in smoothing and MAP estimation which prohibit the utility of adaptive gridding routines. The link between c_t and y_t is established via the function b in (2) so we should inspect b^{-1} to find information about c_t. In this respect we do not require that b is a monotone function since we are only interested in assessing the domain of c from the range of y. There exist unfortunate cases where b includes a multiplicative noise term for which $\Pr(v_t = 0) \neq 0$, leading to the inverse mapping $c_t = b^{-1}(y_t)$ being undefined (divide by zero) or the case where b is periodic. For many practical cases b will, however, have a physical interpretation where natural bounds on c_t exist.

The resolution of the discretization (as determined by m) is another important issue that needs to be addressed before the HMM filter can be applied. The value of m obviously influences the accuracy of the end result and the choice of m therefore depends on the demanded accuracy. A rather *ad hoc* but straightforward approach is to make an initial analysis using a coarse grid and then gradually refine the grid until the required accuracy of the results is attained. The upper limit of the resolution is usually defined by the maximum allowed computing time of the problem which for high-dimensional (> 4) state vector can be reached even for coarse grids and thus render the HMM approach impractical.

2.2 Approximating the integral

The integral in (5) can be readily calculated if a functional expression is available for the cumulative distribution function of the transitions. If this is not the case, (5) must be approximated numerically. To this end any quadrature procedure as known from the literature can be employed. Speed, however, must be prioritized since the integral must be computed m^2 times to construct $\mathbf{\Gamma}_t$ and for inhomogeneous models $\mathbf{\Gamma}_t$ must be reconstructed at each time step. Typically for relatively fine grids a first order approximation to the integral using the trapeziodal rule provides a sufficiently accurate solution. We then get the following expression for the integral

$$\int_{\Omega_j} p_{C_{t+1}|C_t}(c_{t+1}|C_t \in \Omega_i)\, dc_{t+1} = \frac{\Delta_j}{d} \sum_{c \in S_j} p_{C_{t+1}|C_t}(c|C_t \in \Omega_i), \qquad (6)$$

where S_j is a set containing the edges of Ω_j, d is the number of terms in S_j and Δ_j is the size of Ω_j (i.e. interval length in 1D, area in 2D etc.). As an example, in the case where the state is one-dimensional, S_j contains the two end-points of the interval and thus $d = 2$.

2.3 The HMM filter

The filter iterates between time and data-update steps in a way similar to the Kalman filter. The time-update and data-update step together become

$$\boldsymbol{\phi}_t = \frac{\boldsymbol{\phi}_{t-1}\mathbf{\Gamma}\mathbf{P}(x_t)}{\psi_t}, \quad \text{where} \qquad (7)$$

$$\psi_t = \boldsymbol{\phi}_{t-1}\mathbf{\Gamma}\mathbf{P}(x_t)\mathbf{1}'. \qquad (8)$$

This step propagates the state probabilities forward in time (as in 3) and uses Bayes' rule to condition on the next observation, x_t (as in 4). The state-dependent distribution matrix (or equivalently the matrix containing the likelihood of the observations) $\mathbf{P}(x_t)$ is a diagonal matrix with the elements $p_i(x_t)$ in the diagonal (see p. 31 in Zucchini and MacDonald (2009)).

2.3.1 Initial distribution

Often when analysing SSMs it is unknown if it is reasonable to assume that the underlying Markov chain is stationary. If stationarity cannot be assumed the distribution at time $t = 0$ can be omitted. Instead, the first data point can be used to calculate an estimate at $t = 1$:

$$\boldsymbol{\phi}_1 = \frac{\mathbf{1}\mathbf{P}(x_1)}{\psi_1}.$$

2.4 Likelihood estimation

The parameters of the SSM are gathered in the vector $\boldsymbol{\theta}$. The likelihood of $\boldsymbol{\theta}$ is the joint density of the observations

$$L_T(\boldsymbol{\theta}) = p(\mathcal{X}_T|\boldsymbol{\theta}). \tag{9}$$

Using Bayes' rule it can be shown that $\psi_t = p(x_t|\mathcal{X}_{t-1}, \boldsymbol{\theta})$ and therefore it follows that

$$L_T(\boldsymbol{\theta}) = \prod_{t=1}^{T} \psi_t. \tag{10}$$

The maximum likelihood parameter estimate of the model is

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}), \tag{11}$$

which can be found with numerical optimization as described in Chapter 3 in Zucchini and MacDonald (2009).

2.5 HMM smoother (local decoding)

To incorporate all observations in each state estimate, i.e. local decoding the procedure using backward probabilities as described in Chapter 5 in Zucchini and MacDonald (2009) can be followed.

An alternative way to calculate the state probabilities $\varrho_t(i) = \Pr(C_t = i|\mathbf{X}^{(T)} = \mathbf{x}^{(T)})$ is termed the smoothing step in Kalman filtering terminology. The smoothed state estimates are not only conditioned on data observed by t but also on future measurements. The terms $\varrho_t(i)$ are collected in the row vector $\boldsymbol{\varrho}_t = (\varrho_t(1), \ldots, \varrho_t(m))$.

The smoothing procedure is as follows:

1. Compute the vector

$$\boldsymbol{\eta}_{t+1} = \boldsymbol{\varrho}_{t+1}\boldsymbol{\Upsilon}_{t+1}, \tag{12}$$

   where $\boldsymbol{\Upsilon}_{t+1}$ is a diagonal matrix of size $m \times m$ with the following elements in the diagonal $1/\Pr(C_{t+1} = i|\mathcal{X}_t)$. These terms are known from the HMM filter.

2. Multiply $\boldsymbol{\eta}_{t+1}$ with the transposed transition matrix to step backwards in time

$$\boldsymbol{\xi}_t = \boldsymbol{\eta}_{t+1}\boldsymbol{\Gamma}_t^T. \tag{13}$$

3. Get the smoothed state estimate at t with the $m \times m$ matrix $\boldsymbol{\Xi} = \text{diag}(\boldsymbol{\xi}_t)$:

$$\boldsymbol{\varrho}_t = \boldsymbol{\phi}_t \boldsymbol{\Xi}_t. \tag{14}$$

The recursion is initiated with the last estimated distribution from the final iteration of the forward filter, $\boldsymbol{\varrho}_T = \boldsymbol{\phi}_T$ which is also a smoothed estimate.

The proof of the above recursion is omitted here, but see Kitagawa (1987) for a derivation of the general smoothing recursion.

2.6 Global decoding

Global decoding will not be covered in this text as the algorithm is described in detail in Section 5.3.2 in Zucchini and MacDonald (2009). An alternative reference is Viterbi (2006), which also provides some background for the development of the algorithm.

2.7 Pseudo code

Here is pseudo code describing the structure of an implementation of the HMM method for analysing state-space models. First for likelihood estimation of parameters:

```
LOAD data
# Discretization of state-space (gridding)
DEFINE grid resolution
COMPUTE extent of grid
# Parameter estimation
likfun <- FUNCTION
{
  INITIALIZE state vector
  for t in 1:T
  {
    COMPUTE state-dependent distribution
    COMPUTE transition matrix (eq. 5)
    UPDATE state vector (eq. 7)
    STORE normalisation constant (eq. 8)
  }
COMPUTE likelihood value (eq. 10)
RETURN likelihood value
}
MAXIMISE likfun OVER parameters
```

With the estimated parameters then run the filter and the smoother.

```
# Filter
INITIALIZE state vector
for t in 1:T
{
  COMPUTE state-dependent distribution
  COMPUTE transition matrix (eq. 5)
  UPDATE state vector (eq. 7)
}

# Smoother
INITIALIZE smoothed state vector
for t in T:1
{
  COMPUTE Upsilon
  COMPUTE eta (eq. 12)
  COMPUTE transition matrix (eq. 5)
  COMPUTE xi (eq. 13)
  UPDATE smoothed state vector (eq. 14)
}
```

Note that the filter uses the same procedure as was used in calculating the likelihood function. Pseudo code using the forward and backward probabilities is not shown here. Instead refer to the R scripts in Zucchini and MacDonald (2009).

3 Examples

Here we present two examples: the benchmark example known from the nonlinear time series literature and an ARCH (autoregressive conditional heteroskedasticity) model which is essentially an example of a stochastic volatility model.

3.1 A nonlinear time series model - state estimation

First we analyze a simple nonlinear time series model which is widespread as a benchmark example in the SMC literature Kitagawa (1987); Gordon et al. (1993); Cappé et al. (2007). The system and observation equations are

$$C_t = \frac{1}{2}C_{t-1} + 25\frac{C_{t-1}}{1+C_{t-1}^2} + 8\cos(1.2t) + u_t, \tag{15}$$

$$Y_t = \frac{1}{20}C_t^2 + v_t \tag{16}$$

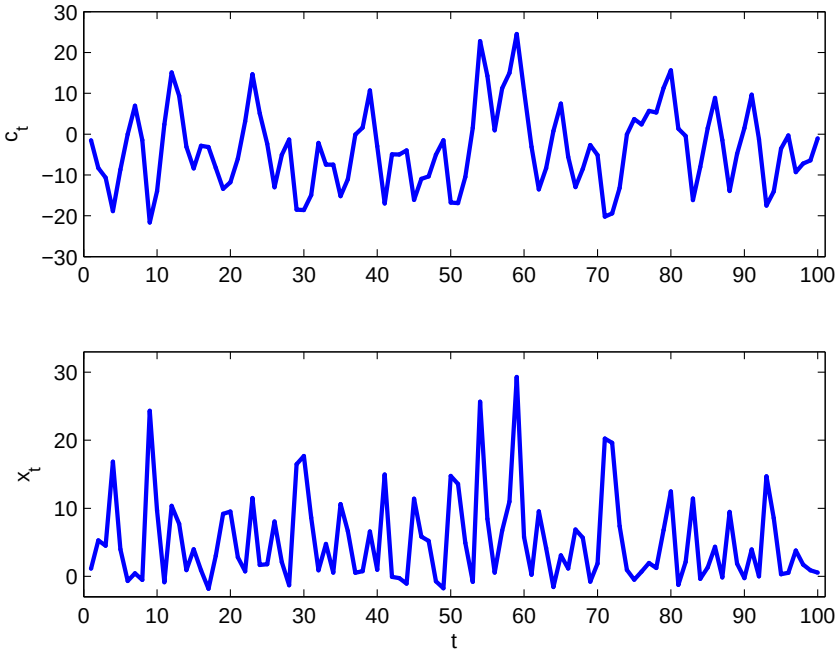

Figure 2: Simulated data from the nonlinear time series model. c_t is the unobserved state and y_t is the observed data.

where u_t and v_t are Gaussian white noise processes with variances $\sigma_u^2 = 10$ and $\sigma_v^2 = 1$. This model is difficult to filter with ordinary methods because the observation likelihood is bimodal owing to the squared term in the observation equation. For $t = 1, \ldots, T$, $T = 100$ we simulated c_t and generated the observed y_t with noise, see Figure 2.

In determining the domain of c_t we considered

$$c_t = b^{-1}(x_t, v_t) = \pm\sqrt{20(x_t - v_t)}. \tag{17}$$

We need to evaluate this expression for the values of x_t and v_t that result in the extreme values of c_t. For x_t we define $\widehat{x}_t = \max\{x_1, \ldots, x_{100}\}$. For v_t we choose the 99.99% quantile in the distribution of v_t (in this case 3.719). Inserting the extreme values of x_t and v_t in (17) we get

$$c_t \in [-25.70;\ 25.70]. \tag{18}$$

In this example we set $m = 500$ which is relatively fine, but not a problem computationally since the state vector is one-dimensional.

The entries in the probability transition matrices are now easily computed

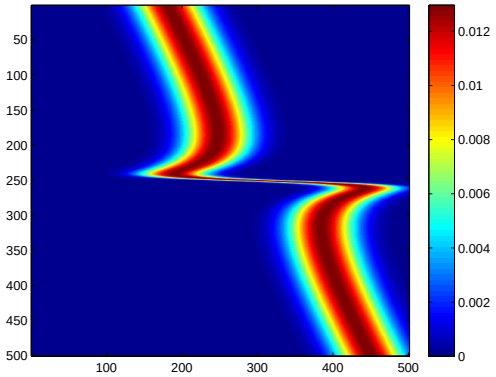

Figure 3: Probability transition matrix for the nonlinear time series in example 1, $t = 100$, $m = 500$.

$$\Pr\left(C_t \in \Omega_j | C_{t-1} \in \Omega_i\right) = \int_{\Omega_j} N_{pdf}\left(c_t, \mu_{t-1}^{(i)}, \sigma_u^2\right) \, dc_t,$$

where

$$\mu_{t-1}^{(i)} = \frac{1}{2}c^{(i)} + 25\frac{c^{(i)}}{1 + \left(c^{(i)}\right)^2} + 8\cos(1.2t)$$

and $c^{(i)}$ is the midpoint of Ω_i and therefore $\Omega_i = [c^{(i)} - \frac{1}{2}\Delta,\, c^{(i)} + \frac{1}{2}\Delta]$. Using the trapezoidal rule for integration we get the approximation

$$\Pr\left(C_t \in \Omega_j | C_{t-1} \in \Omega_i\right) \approx \frac{\Delta}{2}\left(N_{pdf}\left(c^{(j)} - \frac{1}{2}\Delta, \mu_{t-1}^{(i)}, \sigma_u^2\right) + N_{pdf}\left(c^{(j)} + \frac{1}{2}, \mu_{t-1}^{(i)}, \sigma_u^2\right)\right).$$

Calculating all entries in $\mathbf{\Gamma}_t$ requires $m(m+1)$ evaluations of a univariate Gaussian pdf which means that the computing effort scales with the square of the number of grid cells. See Figure 3 for an illustration of the transition matrix for $t = 100$.

We assume all the parameters in the model are known and run the filter and smoothing routines, see Figure 4 for a plot of the resulting probability distributions. From the plot it is clear that the smoothed estimates have much narrower confidence limits because the density estimates are conditioned on all available observations. The estimates shown in Figure 4 are optimal conditional on the discretization and the quadrature procedure (in this case trapeziodal rule). As $m \to \infty$ the HMM smoothed estimates converge to the optimal smoothed estimates.

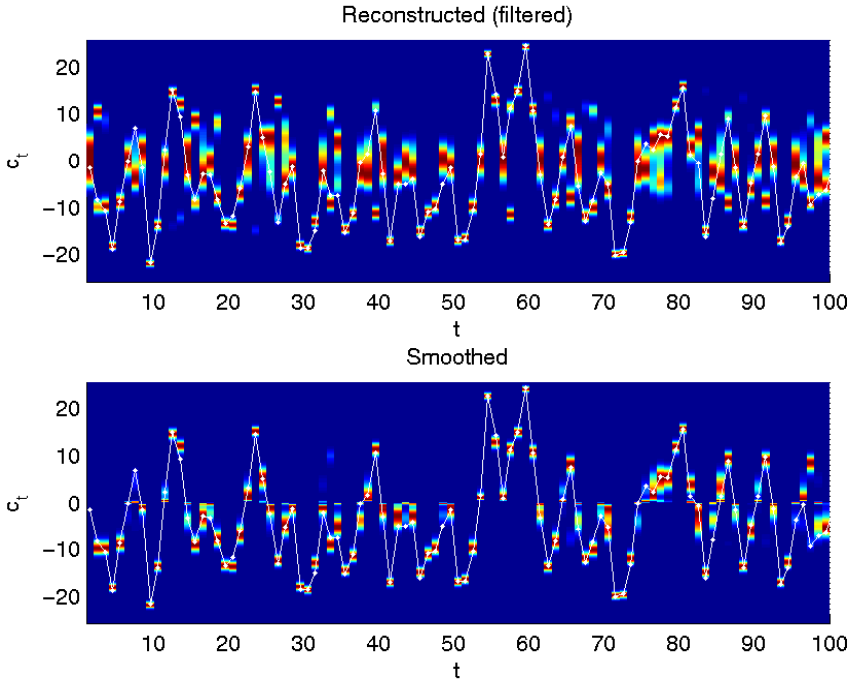

Figure 4: Top panel: filtered density estimates, bottom panel: smoothed density estimates. Since the smoothed estimates are conditional on all available data (including future values) they are more accurate than the filtered estimates.

In this example with $m = 500$ and $T = 100$ the total computing time for both filter and smoother was 11.29 seconds on the `hms2` high-memory server at DTU Informatics using Matlab 7.7 and one core. The vast majority of the computing time was spent on evaluating the normal density (6.85 seconds) and calculating the integrals (5) (3.674 seconds) for constructing $\mathbf{\Gamma}_t$. This time would have doubled had we used the cumulated density function (cdf) for calculating integrals, since the cdf is more expensive to evaluate for the Gaussian case (but not necessarily in general). Less than half a second was spent on the actual filtering and smoothing recursions.

3.2 Stochastic volatility model - parameter estimation

We now consider a model which is part of the ARCH (autoregressive conditional heteroskedasticity) class of models which are often used within econometrics for modelling stochastic volatility. The aim of this example is to estimate the parameters of the model by maximizing the likelihood returned by the HMM filter as described above. The likelihood is optimized using Matlab's build-in optimizer `fmincon`. The model has previously been anal-

ysed by Pitt and Shephard (1999); Doucet and Tadić (2003). Doucet and Tadić (2003) presented a recursive maximum likelihood (RML) using particle filters which gave nice estimation results. Specifically, the model we analyse in this example has the system equation

$$C_t = \phi C_{t-1} + \sigma V_t, \quad C_0 \sim N\left(0, \frac{\sigma^2}{1-\phi^2}\right) \tag{19}$$

and observation equation

$$X_t = \beta \exp\left(\frac{C_t}{2}\right) W_t, \tag{20}$$

where $V_t \sim N(0,1)$ and $W_t \sim N(0,1)$ are independent white noise sequences. We therefore have the parameter vector $\boldsymbol{\theta} = (\phi, \sigma, \beta)$ on the domain $\boldsymbol{\Theta} = (-1,1) \times (0,100) \times (0,100)$. The fixed parameters used for simulation were $\boldsymbol{\theta}^* = (0.8, 0.5, 1)$.

For this example the observation model does not provide information about c_t through b^{-1} since the noise term is multiplicative and $\Pr(W_t = 0) \neq 0$. Instead we inspect the system model itself to find reasonable bounds on c_t. From (19) it is clear that $C_t \sim N[0, \sigma^2/(1-\phi^2)]$. In practice this does not help us, however, since we do not know the values of σ and ϕ and the variance of C_t is unbounded for $|\phi| = 1$. In such a case we need to use physical knowledge about the system. That knowledge is not available here since this is a theoretical example so we use $\boldsymbol{\theta}^*$ to determine the bounds of the discretized state domain. The 99.99% bounds correspond to the domain $c_t \in [-3.10, 3.10]$.

We estimate the parameters for different values of m (see verbatim below) to investigate the performance of the method and its convergence and behavior as the grid becomes finer.

```
CONVERGENCE FOR INCREASING m

m = 100, T = 300
esttime =   60.8008
phi:   [ 0.4411  0.7815  0.9253] True: 0.8
sigma: [ 0.2898  0.5192  0.9301] True: 0.5
beta:  [ 1.0420  1.0928  1.1460] True: 1.0

m = 150, T = 300
esttime =  147.6289
phi:   [ 0.3581  0.7768  0.9354] True: 0.8
sigma: [ 0.2641  0.5241  1.0401] True: 0.5
beta:  [ 0.9880  1.0873  1.1966] True: 1.0

m = 200, T = 300
esttime =  229.5432
phi:   [ 0.3699  0.7762  0.9333] True: 0.8
```

```
sigma: [ 0.2699  0.5247  1.0201] True: 0.5
beta:  [ 0.9519  1.0867  1.2405] True: 1.0

m = 300, T = 300
esttime =  488.7552
phi:   [ 0.5649  0.7800  0.8958] True: 0.8
sigma: [ 0.3434  0.5207  0.7895] True: 0.5
beta:  [ 1.0476  1.0911  1.1365] True: 1.0

m = 500, T = 300
esttime = 1.2447e+03
phi:   [ 0.4349  0.7763  0.9225] True: 0.8
sigma: [ 0.3025  0.5247  0.9100] True: 0.5
beta:  [ 1.0464  1.0868  1.1288] True: 1.0
```

The results show that the HMM method is capable of estimating the model parameters somewhat reliably (in this example the true value of β is outside the confidence bounds, but not by much) for $m = 100$ and $T = 300$ spending only a minute of computing time. In contrast Doucet and Tadić (2003) had to run their model for $T = 20000$ with $N = 10000$ particles to obtain steady estimates. Surprisingly, the confidence bands do not change significantly as m increases and there is not a big difference for $m = 100$ as compared to the case where $m = 500$.

4 Discussion

What is also the conclusion from previous discussions of the HMM method or similar (Kitagawa, 1987) its greatest disadvantage is the increase in computer resources required as the dimension of the state-space increases, i.e. the method suffers from the curse of dimensionality. This cannot be easily overcome if at all. Several attempts have been made (Bucy and Senne, 1971; Šimandl et al., 2006), but these are typically very involved and make evaluation of the estimation performance difficult. It is hard to evaluate if the benefits of an adaptive gridding procedure outweighs the increased implementation effort required, but we conjecture it does reduce the flexibility of the HMM method because smoothing and parameters estimation are no longer possible in general.

A few things can be done to reduce the computational requirements of the HMM method. For inhomogeneous time series constructing the probability transition matrix dominates the CPU usage and quick computation of (5) should therefore be prioritized. For high dimensional state-spaces many state transitions might be improbable leading to $\mathbf{\Gamma}_t$ having a sparse structure. It may therefore be advantageous to use sparse matrix routines to avoid spending CPU time on transitions with zero probability.

Unfortunately the authors of Doucet and Tadić (2003) do not comment on the computational requirements of their estimation method, but they do mention that "...our algorithm requires a substantial amount of data to converge" which is probably because the estimation is recursive and therefore the filter is only run once. Also, the implementational complexity of the RML using a particle filter seems extensive as compared to the HMM approach. Moreover, the HMM method also computes an estimate of the Hessian of the likelihood function which can be translated into estimates of the covariance matrix of the estimates. This is highly convenient and not a feature of the particle RML.

It is commonly asserted that the particle filter avoids the curse of dimensionality. This can be true if one is interested in estimating the conditional mean, median, quantiles etc. of the posterior distribution. These statistics can be estimated empirically with the particle filter with computing time scaling linearly with the number of state dimensions. However, if one is interested in the full posterior density the computation time of the particle filter estimate is no longer linear in the state dimension. In such case a kernel density estimation is required to obtain the posterior distribution. This adds parameters to the estimation procedure, increased complexity and indeed the requirement of a discrete grid to represent the distribution.

References

Alspach, D. and H. Sorenson. 1972. Nonlinear Bayesian estimation using Gaussian sum approximations. IEEE transactions on automatic control **17**:439–448.

Bucy, R. S. and K. D. Senne. 1971. Digital synthesis of nonlinear filters. Automatica **7**:287–298.

Cappé, O., S. Godsill, and E. Moulines. 2007. An overview of existing methods and recent advances in sequential Monte Carlo. Proceedings of the IEEE **95**:899–924.

Challa, S. and Y. Bar-Shalom. 2000. Nonlinear filter design using Fokker-Planck-Kolmogorov probabilitydensity evolutions. IEEE Transactions on Aerospace and Electronic Systems **36**:309–315.

Daum, F. and R. Co. 2005. Nonlinear filters: beyond the Kalman filter. IEEE Aerospace and Electronic Systems Magazine **20**:57–69.

Doucet, A. and V. Tadić. 2003. Parameter estimation in general state-space models using particle methods. Annals of the Institute of Statistical Mathematics **55**:409–422.

Evensen, G. 2003. The ensemble Kalman filter: Theoretical formulation and practical implementation. Ocean dynamics **53**:343–367.

Gilks, W. and D. Spiegelhalter. 1996. Markov chain Monte Carlo in practice. Chapman & Hall/CRC.

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

Jazwinski, A. 1970. Stochastic processes and filtering theory. Academic Pr.

Julier, S. and J. Uhlmann, 1997. A new extension of the Kalman filter to nonlinear systems. Page 26 *in* Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, volume 3. Citeseer.

Kalman, R. E. 1960. A new approach to linear filtering and prediction problems. Journal of Basic Engineering **82**:35–45.

Kitagawa, G. 1987. Non-Gaussian state-space modeling of nonstationary time series. Journal of the American statistical association **82**:1032–1041.

Kitagawa, G. 1996. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. Journal of computational and graphical statistics **5**:1–25.

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

Šimandl, M., J. Královec, and T. Söderström. 2006. Advanced point-mass method for nonlinear state estimation. Automatica **42**:1133–1145.

Viterbi, A. J. 2006. A personal history of the Viterbi Algorithm. IEEE Signal Processing Magazine **23**:120–142.

Welch, G. and G. Bishop. 1995. An introduction to the Kalman filter. University of North Carolina at Chapel Hill, Chapel Hill, NC .

Wiberg, D. and D. DeWolf. 1993. A convergent approximation of the continuous-time optimal parameterestimator. IEEE Transactions on Automatic Control **38**:529–545.

Zucchini, W. and I. L. MacDonald. 2009. Hidden Markov Models for Time Series. Chapman & Hall/CRC.