Dynamic models for nonstationary signal segmentation

William D. Penny and Stephen J. Roberts

w.penny@ic.ac.uk, s.j.roberts@ic.ac.uk
Neural Systems Research Group,
Department of Electrical and Electronic Engineering,
Imperial College of Science, Technology and Medicine,
London SW7 2BT., U.K.

January 15, 1999

Abstract

This paper investigates Hidden Markov Models (HMMs) in which the observations are generated from an autoregressive (AR) model. The overall model performs nonstationary spectral analysis and automatically segments a time series into discrete dynamic regimes. Because learning in HMMs is sensitive to initial conditions, we initialise the HMM model with parameters derived from a cluster analysis of Kalman filter coefficients. An important aspect of the Kalman filter implementation is that the state noise is estimated on-line. This allows for an initial estimation of AR parameters for each of the different dynamic regimes. These estimates are then fine-tuned with the HMM model. The method is demonstrated on a number of synthetic problems and on electroencephalogram (EEG) data.

1 Introduction

Autoregressive (AR) models are a well-known parametric technique for the spectral estimation of stationary signals [1]. The standard AR model can also be used for the spectral estimation of nonstationary signals by embedding it into a Kalman filter. The so-called 'Kalman filter AR model' (KF-AR) achieves this by having a state which is a vector of AR coefficients [2]. The evolution of this state allows for the tracking of nonstationarities and, in the Kalman filter, this state evolves continuously.

In this paper we show that the Hidden Markov AR model (HMM-AR) [3], which was originally developed for the purpose of time series prediction, can also be used for the spectral estimation of nonstationary signals. It can also be used to segment a signal into different sections within which the signal is considered to be stationary. The HMM-AR model is identical to the KF-AR model except that in the HMM-AR model the states are discrete whereas in the KF-AR model they are continuous.

Whereas Kalman filters have been widely used in the biomedical signal processing community there are relatively few applications using Hidden Markov Models (HMMs). As the suitability of the dynamic model (HMM or Kalman filter) depends solely on whether the underlying system dynamics are discrete or continuous this imbalance seems unjustified and, in this paper, we attempt to redress this imbalance by giving examples involving discrete dynamics i.e. by applying HMM-AR models to biomedical data sets.

A possible shortcoming of the HMM approach, however, is that HMMs are usually trained with an Expectation-Maximisation (EM) algorithm [4] which, being a local search method, is susceptible to becoming trapped at local maxima. It is therefore important to initialise the HMM models carefully. We address this issue by describing an initialisation method based on a cluster analysis of AR vectors where the AR vectors are derived from a KF-AR model. The KF-AR model, even though it is intrinsically suited to continuous dynamics, can be set up to detect and follow discrete dynamic changes by updating a 'state noise' parameter on-line using the Jazwinski algorithm [5, 6]. This allows the different dynamic regimes to be approximately identified. This identification is sufficient to initialise the HMM-AR model which then fine tunes the AR models (and the HMM state transition matrix) using the EM algorithm.

In sections 2 and 3 of this paper we describe the KF-AR and HMM-AR models and in section 4 we describe the HMM initialisation process. The methods are then demonstrated on synthetic problems and on EEG data in section 5.

2 Kalman filter AR models

An autoregressive model is a linear predictor used for modelling time series. It is a time-series version of linear regression whereby an $AR(p)$ model predicts the next value in a time series as a linear combination of the p previous values

$$y_t = -\sum_{i}^{p} a_i y_{t-i} + v_t \tag{1}$$

where a_i are the AR coefficients and v_t is independent identically distributed (IID) Gaussian noise with zero mean and variance σ^2. We define a *time-varying* autoregressive model or Kalman filter AR model by the following state-space equations

$$\boldsymbol{q}_t = \boldsymbol{q}_{t-1} + \boldsymbol{w}_t, \qquad \boldsymbol{w}_t \sim N(0, \boldsymbol{W}_t) \tag{2}$$

$$y_t = \boldsymbol{F}_t \boldsymbol{q}_t + v_t, \qquad v_t \sim N(0, \sigma_t^2) \tag{3}$$

where $\boldsymbol{F}_t = -[y_{t-1}, y_{t-2}, ..., y_{t-p}]$ and $N(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2. The state variables, $\boldsymbol{q}_t = [a_{1,t}, a_{2,t}, ..., a_{p,t}]^T$, are time-varying AR coefficients. The vector $\boldsymbol{w}_t$ is state noise and the scalar v_t is observation noise. This model is identical to the generic Kalman filter model [7] with the state transition matrix (which acts on $\boldsymbol{q}_{t-1}$) equal to the identity matrix. This simplification means that we have one less parameter matrix to estimate (see end of this section) but, because of the additive state noise, the model is still capable of tracking non-stationarities.

The Kalman filter may be viewed as a Bayesian estimator of the state vector. We can write the probability distribution for the current state vector *before* the observation y_t is made, as the prior distribution

$$p(\boldsymbol{q}_t) = N(\hat{\boldsymbol{q}}_{t-1}, \boldsymbol{\Sigma}_{t-1} + \boldsymbol{W}_t) \tag{4}$$

The mean of the prior distribution is simply the state estimate from the previous time step, $\hat{\boldsymbol{q}}_{t-1}$, and the covariance is the covariance of the state estimate from the previous time step, $\boldsymbol{\Sigma}_{t-1}$, plus the state noise covariance. This second term arises because we are adding on state noise to the previous state estimate (see Eq. 2). The likelihood of making an observation, y_t, given state vector $\boldsymbol{q}_t$ can be inferred from an inspection of Eq. 3. It is

$$p(y_t \mid \boldsymbol{q}_t) = N(\boldsymbol{F}_t \hat{\boldsymbol{q}}_{t-1}, \sigma_t^2) \tag{5}$$

The mean of the likelihood distribution is the predicted value, $\hat{y}_t = \boldsymbol{F}_t \hat{\boldsymbol{q}}_{t-1}$. The difference between the observed value and the predicted value is the prediction error, e_t. The variance of the likelihood distribution is simply the observation noise variance, σ_t^2.

The likelihood term acts to reduce the prior distribution to a posterior distribution according to Bayes rule

$$p(\boldsymbol{q}_t \mid y_t) = \frac{p(y_t \mid \boldsymbol{q}_t) p(\boldsymbol{q}_t)}{\int p(y_t \mid \boldsymbol{q}_t) p(\boldsymbol{q}_t) d\boldsymbol{q}_t} \tag{6}$$

Because the prior is Gaussian and the likelihood is Gaussian the posterior is also Gaussian. The mean and covariance of the posterior can be calculated from the above equation. These calculations, though conceptually simple, are somewhat lengthy. We refer the interested reader to the tutorial by Meinhold and Singpurwalla [7] who show that the posterior is given by

$$p(\boldsymbol{q}_t | y_t) = N(\hat{\boldsymbol{q}}_t, \boldsymbol{\Sigma}_t) \tag{7}$$

where the mean and covariance are

$$\begin{aligned} \hat{\boldsymbol{q}}_t &= \hat{\boldsymbol{q}}_{t-1} + \boldsymbol{K}_t e_t & (8) \\ \boldsymbol{\Sigma}_t &= \boldsymbol{\Sigma}_{t-1} + \boldsymbol{W}_t - \boldsymbol{K}_t \boldsymbol{F}_t (\boldsymbol{\Sigma}_{t-1} + \boldsymbol{W}_t) & (9) \end{aligned}$$

where $\boldsymbol{K}_t$ is the 'Kalman gain' vector which is calculated as follows

$$\begin{aligned} \boldsymbol{K}_t &= \frac{(\boldsymbol{\Sigma}_{t-1} + \boldsymbol{W}_t) \boldsymbol{F}_t^T}{\sigma_{\hat{y}_t}^2} & (10) \\ \sigma_{\hat{y}_t}^2 &= \sigma_t^2 + \boldsymbol{F}_t \boldsymbol{\Sigma}_{t-1} \boldsymbol{F}_t^T + \boldsymbol{F}_t \boldsymbol{W}_t \boldsymbol{F}_t^T & (11) \end{aligned}$$

where $\sigma_{\hat{y}_t}^2$ is the estimated prediction variance. This is composed of three terms; the observation noise, σ_t^2, the component due to the uncertainty in the state variables from the last estimation step and the component due to the injection of state noise.

A useful quantity is the likelihood of an observation given the model parameters before they are updated

$$p(y_t) = N\left(\hat{y}_t, \sigma_{\hat{y}_t}^2\right) \tag{12}$$

In Bayesian terminology this is also known as the evidence for the data point [6]. Data points with low evidence correspond to periods when the statistics of the underlying system are changing (non-stationarity) or, less consistently, to data points having large target noise components.

It is also interesting to monitor the effective learning rate of a Kalman filter model because it provides a check that the filter is converging to stationary solutions, when given stationary data, and is adapting to new regimes, when given nonstationary data. DeFreitas and Niranjan[6] show that the Kalman filter has a learning rate *matrix* equal to

$$\boldsymbol{\rho} = \frac{\boldsymbol{\Sigma}_{t-1} + n_t \boldsymbol{I}}{\sigma_{\hat{y}_t}^2} \tag{13}$$

The learning rate is a matrix because each state variable (AR coefficient) adapts at a different rate to each different AR input. It is possible to get an idea of the overall learning rate, however, by simply averaging the learning rate over all state variables [8]

$$\rho_{av} = \frac{1}{p} \frac{Tr\left(\boldsymbol{\Sigma}_{t-1} + n_t \boldsymbol{I}\right)}{\sigma_{\hat{y}_t}^2} \tag{14}$$

where $Tr()$ denotes the trace of the covariance matrix and p is the number of state variables.

To apply the Kalman filter model we make initial estimates for the AR parameters, $\hat{\boldsymbol{q}}_0$, and their covariance, $\boldsymbol{\Sigma}_0$, and use Eq.'s 8 to 11. The initial estimates can be obtained by passing a static AR model over an initial portion of the time series. We must also plug in estimates for the state noise covariance, $\boldsymbol{W}_t$, and the observation noise variance, σ_t^2.

The state noise is estimated on-line using Jazwinski's algorithm [5] which is reviewed in [6]. This method requires either that the state noise is calculated over a window of samples or that the estimates are smoothed from sample to sample [8]. The observation noise is set to a fixed value estimated by passing a static AR model over an initial portion of the time-series.

3 Hidden Markov AR Models

We use the Hidden Markov AR models described by Fraser and Dimitriadis [3] where the hidden states evolve with the usual Markovian dynamics [4] but the observations are generated by AR models. The HMMs have N discrete hidden states, $[S_1, S_2, ..., S_N]$ and the state-space equations are

$$\begin{aligned} P(q_t = S_j | q_{t-1} = S_i) &= r_{ij} && (15) \\ P(\boldsymbol{y}_t | q_t = S_i) &= N(\boldsymbol{F}_t \hat{\boldsymbol{a}}_i, \sigma_i^2) && (16) \end{aligned}$$

where q_t is the state at time t, r_{ij} are the state transition probabilities which are stored in the state transition matrix $\boldsymbol{A}$, $\boldsymbol{F}_t = -[y_{t-1}, y_{t-2}, ..., y_{t-p}]$, $\hat{\boldsymbol{a}}_i$ is the (column) vector of AR coefficients for the ith state and σ_i^2 is the estimated observation noise for the ith state. The prediction for the ith state is $\hat{y}_t^i = \boldsymbol{F}_t \hat{\boldsymbol{a}}_i$. The above two equations can be compared with the Kalman filter equations; Eq.'s 2 and 3. The only difference is that in HMMs the state is discrete whereas in the Kalman filter it is continuous (the Kalman filter state is a continuous vector, $\boldsymbol{q}_t$, containing AR coefficients, whereas the HMM state is a discrete flag, q_t, which indexes a vector of AR coefficients, $\hat{\boldsymbol{a}}_{q_t}$). The initial state is set according to

$$P(q_1 = S_j) = \pi_j \tag{17}$$

To train an HMM it is necessary to be able to calculate the probability of being in state S_i at time t given a particular observation sequence

$$P(q_t = S_i | \boldsymbol{y}_1^T) = \gamma_t(i) \tag{18}$$

where $\boldsymbol{y}_1^T$ denotes all observations in a length T sequence. It is also necessary to calculate the probability of state S_j succeeding state S_i given a particular observation sequence

$$P(q_t = S_i, q_{t+1} = S_j | \boldsymbol{y}_1^T) = \xi_t(i,j) \tag{19}$$

The two variables $\gamma_t(i)$ and $\xi_t(i,j)$ can be calculated using the forward-backward algorithm [4]. An Expectation-Maximisation (EM) algorithm can then be used to estimate the model parameters; the initial state probabilities and the state transition matrix are estimated in the usual way [4] and the AR parameters are set so as to minimise [3]

$$E_i = \sum_t^T \gamma_t(i)(y_t - \hat{y}_t^i)^2 \tag{20}$$

This is a weighted least squares problem. Each 'weight', $\gamma_t(i)$, is the probability that the HMM is in state i at time t. If $\boldsymbol{C}_i = diag(\sqrt{\gamma_1(i)}, \sqrt{\gamma_2(i)}, ..., \sqrt{\gamma_T(i)})$, $\boldsymbol{Y}_i$ is the column vector whose tth element is y_t and $\boldsymbol{X}_i$ is a matrix whose tth row is $[-y_{t-p}, ..., -y_{t-1}]$, then by defining $\tilde{\boldsymbol{X}}_i = \boldsymbol{C}_i \boldsymbol{X}_i$ and $\tilde{\boldsymbol{Y}}_i = \boldsymbol{C}_i \boldsymbol{Y}_i$ we can express the solution in standard least squares form

$$\hat{\boldsymbol{a}}_i = \left(\tilde{\boldsymbol{X}}_i^T \tilde{\boldsymbol{X}}_i \right)^{-1} \tilde{\boldsymbol{X}}_i^T \tilde{\boldsymbol{Y}}_i \tag{21}$$

which can be efficiently computed using singular value decomposition [9]. The noise variance for each state is then calculated as

$$\hat{\sigma}_i^2 = \frac{\sum_t \gamma_t(i)(y_t - \hat{y}_t^i)^2}{\sum_t \gamma_t(i)} \tag{22}$$

To summarise, the HMM model is trained by running the forward backward algorithm to obtain values for $\gamma_t(i)$ and $\xi_t(i,j)$. The initial state probabilities and the state transition matrix are then updated in the usual way [4] and the AR models are updated using Eq.'s 21 and 22. This constitutes one iteration of the EM training algorithm. The computational demands of each training iteration are therefore a single pass of the forward-backward algorithm and the solution of K weighted least squares problems. Typically, training requires tens of iterations.

After the HMM-AR models have been trained, they can be applied to novel data sets and the most likely sequence of hidden states identified using the Viterbi algorithm [4].

4 HMM Initialisation

HMMs trained using the EM algorithm are guaranteed to reach only a local maximum likelihood solution (or, less likely, a saddle point of the likelihood function). Because there are many local maxima in the likelihood landscape [4] HMMs are sensitive to how they are initialised. For this reason we initialise the HMM-AR models rather carefully using the following procedure.

The Kalman filter AR model, as described in section 2, is passed over the data with the state noise smoothing coefficient, α, set to a value such that the model is predisposed to detecting sharp state transitions (see [8]). In this paper we use a value of $\alpha = 0.1$ for all experiments.

This produces a set of AR vectors at each time point, $\{\boldsymbol{a}_t\}$. AR vectors corresponding to low evidence models (see Eq. 12) are then removed from this data set. We do this because low evidence models are, by definition, poor models of the original time series.

A Gaussian Mixture Model (GMM) [9] is then used to perform a cluster analysis of the set of remaining AR vectors. This is performed using full-covariance matrices for each of the Gaussians in the GMM as this is the most general cluster model. The centre of each Gaussian cluster is then used to initialise the AR coefficients in each state of the HMM-AR model. The overall initialisation process is shown schematically in Figure 1.

It is also important to initialise the HMM state transition matrix. The approach adopted in this paper is to use prior knowledge from the problem domain about average state duration densities, i.e. for how long is the signal stationary ? If d is the average number of samples we wish the HMM to remain in state i for, then the corresponding diagonal entry in the state transition matrix is set to $r_{ii} = 1 - \frac{1}{d}$. EEG data for example, is known to be stationary for a period of the order of half a second [10]. For 125Hz data this leads to diagonal entries of 0.984. The off-diagonal entries soak up the remaining probabiltiy.

4.1 Window methods

An alternative approach to using Kalman filters for nonstationary spectral estimation is to use Short-Time Fourier Transform or Short-Time AR models. In these methods the time series is split up into a number of discrete overlapping blocks and a Fourier Transform or AR model applied to each.

It is then possible to apply an HMM with a Guassian or Gaussian Mixture observation model, not to the original data, but to the sequence of derived feature values. This has the advantage of reducing the dimensionality of the space in which the HMM operates and therefore requires much less computation. This is the approach used in speech recognition [4].

A problem with this approach, however, is that the resulting estimates of state and state transition can be less than accurate. This is because windows which straddle a state transition will contain samples from two different states. The resulting coefficients will be representative of neither state but of a new spurious state which is a combination of the two. It is therefore likely that the optimal number of states will be overestimated and that the state estimates will be somewhat 'smeared'.

5 Simulations

In order to validate the dynamic modelling techniques we first analyse, in sections 5.1 and 5.2, time series with known spectral content and segmentation. We then apply the methods to EEG data in sections 5.3 and 5.4. In each of the experiments the order of the AR model was chosen, arbitrarily, to be $p = 8$. Although we made no attempt to optimise the model order, for each experiment and for each state of the HMM, we note that all the usual model order selection methods [11] can be applied.

5.1 Spectral Analysis

This example compares spectral estimates from the Kalman filter AR model and the Hidden Markov AR model. It also demonstrates the basic functioning of the Kalman filter model, with a particular focus on the on-line noise estimation algorithm.

A time series was generated consisting of a 10Hz sine wave in the first second, a 20Hz sinewave in the next second and a 30Hz sine wave in the third second. All signals contained additive Gaussian noise with standard deviation 0.1. One hundred samples were generated per second.

Static AR models were derived from several overlapping blocks of data in order to estimate the average observation noise in the time series. This estimate of observation noise was then used in the Kalman filter AR model. The Kalman filter state noise was initially set to zero and was adapted using Jazwinski's algorithm. The model was initialised using linear regression; the first p data points were regressed onto the $p + 1$th data point using least squares. The initial state, q_{p+1}, was initialised to the resulting least squares solution. The initial state covariance matrix was set to the linear regression covariance matrix, $\Sigma_{p+1} = \sigma_t^2 \boldsymbol{F}_{p+1} \boldsymbol{F}_{p+1}^T$.

Figures 2 (a) and (b) show the original time series and the evidence of each point under the Kalman filter model. Data points occuring at the transitions between different dynamic regimes have low evidence. Figure 2(c) shows the evolution of the state noise parameter. The state noise is increased to account for the shortfall between the estimated prediction error and the actual prediction error. The increase in state noise leads to an increase in the learning rate, as shown in Figure 2(d), and the filter is seen to quickly adapt to each new dynamic regime.

The Kalman filter model was then used to initialise the HMM-AR model, as described in section 4. The spectra were estimated directly from the AR coefficients as described in [1].

Figure 3 shows the spectral estimates obtained from the Kalman filter AR model and the HMM-AR model. Although the Kalman filter is able to track rapid state transitions it is not fundamentally suited to doing so. It does, however, provide a useful initialisation for the HMM-AR model which when trained with EM performs the required state and spectral estimates.

5.2 State reconstruction

This example demonstrates the ability of the HMM-AR model to recover underlying state dynamics from a nonstationary signal. We generate data from a three state HMM where data in each state is generated according to

$$y_t = a_1 \sin(2\pi 10 t) + a_2 \sin(2\pi 25 t) + e_t \tag{23}$$

where e_t is zero mean IID Gaussian noise with standard deviation 0.3. The weighting coefficients were set differently for each state. For state 1 we have $a_1 = 0$, $a_2 = 1$, for state 2 we have $a_1 = 0.5$, $a_2 = 0.5$ and for state 3 we have $a_1 = 1$, $a_2 = 0$. The initial state is state 1 and the state transition matrix was set to

$$\boldsymbol{P} = \begin{pmatrix} 0.98 & 0.02 & 0.00 \\ 0.00 & 0.98 & 0.02 \\ 0.02 & 0.00 & 0.98 \end{pmatrix} \tag{24}$$

The diagonal elements were chosen such that the average state duration density was half a second. The off-diagonal elements were chosen arbitrarily (there is no significance in the fact that we have a cycle of transitions).

Forty seconds of data were generated at a sampling rate of 125Hz. A Kalman filter AR model was then applied to the data set. Figure 4 shows 10 seconds of data along with the true hidden state and evidence of data points under the Kalman filter model.

A data set was formed from the AR coefficients corresponding to high evidence predictions. The cut-off was $p(y_t) > 0.5$ (other choices of cut-off values, in the range 0.1 to 0.9, were also tried and resulted in only marginally different initial spectral estimates). Out of 5000 AR vectors only 1807 were used. A three-state GMM was then applied to the data as described in section 4. An HMM-AR model was initialised to the centre of each cluster and the model was trained further with the EM algorithm. The resulting HMM-AR models had the state transition matrix

$$\hat{\boldsymbol{P}} = \begin{pmatrix} 0.968 & 0.032 & 0.000 \\ 0.000 & 0.983 & 0.017 \\ 0.023 & 0.000 & 0.977 \end{pmatrix} \tag{25}$$

Application of the Viterbi decoding algorithm resulted in the estimated state sequence shown in Figure 4 (Row 3). Figure 5 shows the estimated spectra from the Kalman filter AR model and HMM-AR model, for a short portion of the signal.

The HMM-AR model thus recovers the correct state transition matrix, state spectra and hidden state sequence. We also applied the windowed HMM method described in section 4.1. Because of the problem of overlapping windows described earlier, the true 25Hz state was estimated to also have a significant 10Hz component and the true 10Hz state was estimated to also have a significant 25Hz component. This led to errors in estimating the transition matrix, where, for example, state 3 was mistakenly inferred to exit to state 2 instead of to state 1.

5.3 EEG hand movement data

In this section we describe the results of applying an HMM-AR model to a section of EEG data recorded whilst a subject performed cued hand movements [12]. Although recordings were made from 22 electrodes we focus on just one channel as an example. Seconds 30 to 70 of a 90 second EEG recording are shown as the top trace in Figure 6. The dashed vertical lines, preceding seconds 36, 50 and 64 indicate the movement cues which signified to the subject that they should open and close their hand.

A Kalman filter AR model was applied to the data and a two-state Gaussian Mixture model was derived from the resulting AR coefficients. This was used to initialise a two-state HMM-AR model. Two states were used as we wished to build a model in which one state corresponds to 'hand movement' (after the cue) and the other state corresponds to 'no hand movement' (before the cue). We also initialised the state transition matrix such that the average state duration density was half a second. We then trained the HMM-AR model in the usual way. The middle trace in Figure 6 shows the HMM-AR state as calculated by the Viterbi algorithm. The bottom trace in Figure 6 shows the state as estimated by a two-component Gaussian Mixture model (GMM) trained on the Kalman filter coefficients. The classifications from the GMM, which use full covariance matrices, are equivalent to those obtainable from a quadratic discriminant classifier [9]. The GMM is, however, a static model i.e. the estimate of state at time t is not dependent on the estimate of state at other times. For this reason, the GMM state estimates have many spurious transitions. The HMM-AR model, however, correctly partitions the data into 'hand movement' and 'no hand movement' sections.

Figure 7 shows the spectra associated with each AR model. The 'no hand movement' state is identified with a 10Hz rhythm and the 'hand movement' state with a low-power broad-band frequency spectrum. These two states correspond to before and after the desynchronisation of the 10Hz rhythm, the 'mu' rhythm, that is known to accompany finger and hand movement [12]. Importantly, however, our partitioning of the data was achieved without the knowledge that we should be looking for 10Hz activity.

5.4 EEG sleep spindle data

We now describe the results of applying the HMM-AR model to EEG data recorded during a sleep experiment. The aim of the analysis is to identify 'sleep spindles' from background EEG. For this reason we use a two-state model.

The analysis was performed in the usual way. The Kalman filter AR model was applied to a 40-second section of data and high evidence AR-coefficients were used in a two-state Gaussian Mixture Model. This 'unsupervised' method was used to initialise the HMM.

We also tried a 'supervised' initialisation scheme whereby an expert's labelling of the data was used to partition the data into two sets and an AR model was applied to each.

A third initialisation method was also investigated. This consisted of running an AR model over the whole time series and setting the AR model of each state to the global AR solution plus a small noise perturbation vector. This initialisation method is referred to as 'None', as it is rather ad-hoc and we include it only to be able to quantify the worth of using more sophisticated initialistion methods (eg. clustering of Kalman filter AR coefficients).

The state transition matrix was initialised such that the average state duration density was half a second as this is known a priori to be compatible with sleep spindle time scales. We then tried the different initialisation methods in combination with different training schemes whereby the state transition matrices and observation models were either both updated, neither were updated, or only the state transition matrix was updated. The likelihoods of some of the resulting models are summarised in Table 1.

From this table we can conclude that, at least for this data set, (1) unsupervised initialisation is better than no initialisation at all, (2) unsupervised initialisation leads ultimately to the same EM solution as supervised initialisation and (3) the EM solution is a higher likelihood labelling of the data than is the expert's labelling (model 4).

We also tried varying the initial average state duration densities. Any value above about 0.1s led to the same full EM solution (models 1 or 2). Values below 0.1s led to a sub-optimal solution, with a likelihood of around -540, which involved many spurious state transitions.

Figures 8 (a) and (b) show the spectra associated with the supervised and unsupervised intialisation methods (models 4 and 6). The supervised method identifies the background EEG as being primarily low frequency activity (1-5 Hz) and the sleep spindles as being primarily 12-13Hz activity with a small low frequency component. The unsupervised spectra are similar, but the 12-13Hz peak for the sleep spindle state is much smaller. Figures 9 (a) and (b) show the spectra after training the HMM-AR model with EM and updating both the state transition matrix and observation model (the 'full' EM solution; models 1 and 2). Both initialisation methods led to the *same* solution. The background EEG state is relatively unchanged but the spindle state has two equal magnitude peaks; one at 2-3Hz and the other at 12-13Hz.

Figure 10 shows the EEG data, the expert's labelling and the labelling from Viterbi decoding of the full EM solution. These two labellings are largely in agreement, but it could be argued that the expert has missed a number of regions of spindle activity (see seconds 17 and 27 for example).

6 Discussion

We have shown that HMM-AR models, originally proposed for the purpose of time series prediction [3], can also be used for the segmentation of nonstationary biomedical signals. The signals are segmented into stationary regimes characterised by different AR models. This type of segmentation is attractive because each AR model is associated with a particular spectral trace [1] allowing the segmentations to be readily interpreted.

We have also focussed on the issue of HMM initialisation. This is important because there are many sub-optimal solutions in the likelihood landscape that the EM algorithm, which employs a local search strategy, may be drawn to. The AR coefficients in the HMM-AR models can be initialised via Gaussian Mixture clustering of AR coefficients derived from a Kalman filter model. This 'unsupervised' initialisation method can be crucial to finding good solutions and, surprisingly, was shown, on one of our EEG data sets, to be as good as a 'supervised' initialisation method which used a pre-existing data labelling. The HMM state transition matrix can be initialised by defining the period over which the time series may be considered to be stationary. This provides a natural way of incorporating prior knowledge from the problem domain into the statistical model, and can also have a crucial influence on the solutions obtained.

We have incorporated our prior knowledge, from the problem domain and from Kalman filter modelling, by initialising an HMM model. A more rigorous approach is to use a full Bayesian analysis involving Dirichlet priors over the state transition matrix [13] and Gaussian priors over the AR coefficients [14] and to find maximum posterior solutions instead of maximum likelihood solutions. The 'full' Bayesian approach involving sampling from the posterior distribution may be achieved with Markov Chain Monte Carlo [15]. Application of these methods to HMM-AR models is an area for future research.

We also note that the HMM-AR approach is readily extendible to Multivariate Autoregressive (MAR) models. The resulting 'HMM-MAR' models perform non-stationary *cross-spectral* analysis, where switching between hidden states causes switching between stationary regimes. This type of model is suited, for example, to *multi-channel* EEG analysis.

Another key area for future work is the design of algorithms for the automatic selection of the number of states in an HMM, and the application of existing methods [16] to HMM-AR or HMM-MAR models.

7 Acknowledgements

The authors would like thank Iead Rezek for many useful discussions regarding AR models. We would also like to thank Nahid Haque who recorded the EEG data for the hand movement experiment and Prof. Rappelsberger, Prof. Trenker, Prof. Zeitlhofer and members of the SIESTA project (funded by the European Community Commision, DG XII, Biomed-2 PL962040) for supplying the sleep spindle data. W. D. Penny was supported by the UK Engineering and Physical Sciences Research Council (EPSRC), grant number GR/K79062, and equipment and technical support for the EEG hand-movement experiments was funded by the Living Again Trust.

References

[1] J. Pardey, S. Roberts, and L. Tarassenko. A Review of Parametric Modelling Techniques for EEG Analysis. *Med. Eng. Phys.*, 18(1):2–11, 1996.

[2] D. W. Skagen. Estimation of Running Frequency Spectra Using a Kalman Filter Algorithm. *Journal of Biomedical Engineering*, 10:275–279, May 1988.

[3] A.M. Fraser and A. Dimitriadis. Forecasting Probability Densities by Using Hidden Markov Models with Mixed States. In A.S. Weigend and N.A. Gershenfeld, editors, *Time Series Prediction: Forecasting the Future and Understanding the Past*, pages 265–282. Addison-Wesley, 1994.

[4] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[5] A.H. Jazwinski. Adaptive Filtering. *Automatica*, 5:475–485, 1969.

[6] J.F.G. DeFreitas, M. Niranjan, and A.H. Gee. Hierarchical Bayesian-Kalman Models for Regularisation and ARD in Sequential Learning. Technical Report CUED/F-INFENG/TR 307, Department of Engineering, Cambridge University, 1998. Also avaiable from http://svr-www.eng.cam.ac.uk/ jfgf/publications.html.

[7] R.J. Meinhold and N.D. Singpurwalla. Understanding the Kalman Filter. *The American Statistician*, 37(2):123–127, May 1983.

[8] W.D. Penny and S.J. Roberts. Dynamic Linear Models, Recursive Least Squares and Steepest Descent Learning. Technical Report TR-98-13, Department of Electrical Engineering, Imperial College, 1998. Available from http://www.ee.ic.ac.uk/research/neural/wpenny/publications/publications.html.

[9] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, 1995.

[10] P.L. Nunez. *Neocortical Dynamics and Human EEG Rhythms.* Oxford University Press, 1995.

[11] C. Chatfield. *The Analysis of Time Series: An Introduction.* Chapman and Hall, 1996.

[12] S. J. Roberts, W. Penny, and I. Rezek. Temporal and Spatial Complexity Measures for EEG-based Brain-Computer Interfacing. *Medical and Biological Engineering and Computing*, 1998. To appear.

[13] P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach.* MIT Press, 1998.

[14] D.J.C. Mackay. Bayesian Interpolation. *Neural computation*, 4(3):415–447, 1992.

[15] G. Celeux C.P. Robert and J. Diebolt. Bayesian Estimation of Hidden Markov Models. *Statistics and Probability Letters*, 16:77–83, 1993.

[16] T. Ryden. Estimating the Order of Hidden Markov Models. *Statistics*, 26(4):345–354, 1995.

Table 1: *Likelihood of HMM-AR models on portion of EEG sleep spindle data. Models 1 and 2 (the 'full' EM solutions) are actually the same model but are reached from different initialisations. The likelihood values for the models with no initialisation are median values calculated from 5 trials. For all of the models the state transition matrix was initialised such that the state duration density was half a second.*

Model	Initialisation	Update transition matrix	Update AR models	Likelihood
1	Unsupervised	Yes	Yes	-417
2	Supervised	Yes	Yes	-417
3	Supervised	Yes	No	-455
4	Supervised	No	No	-487
5	None	Yes	Yes	-545
6	Unsupervised	No	No	-811
7	None	No	No	-1830

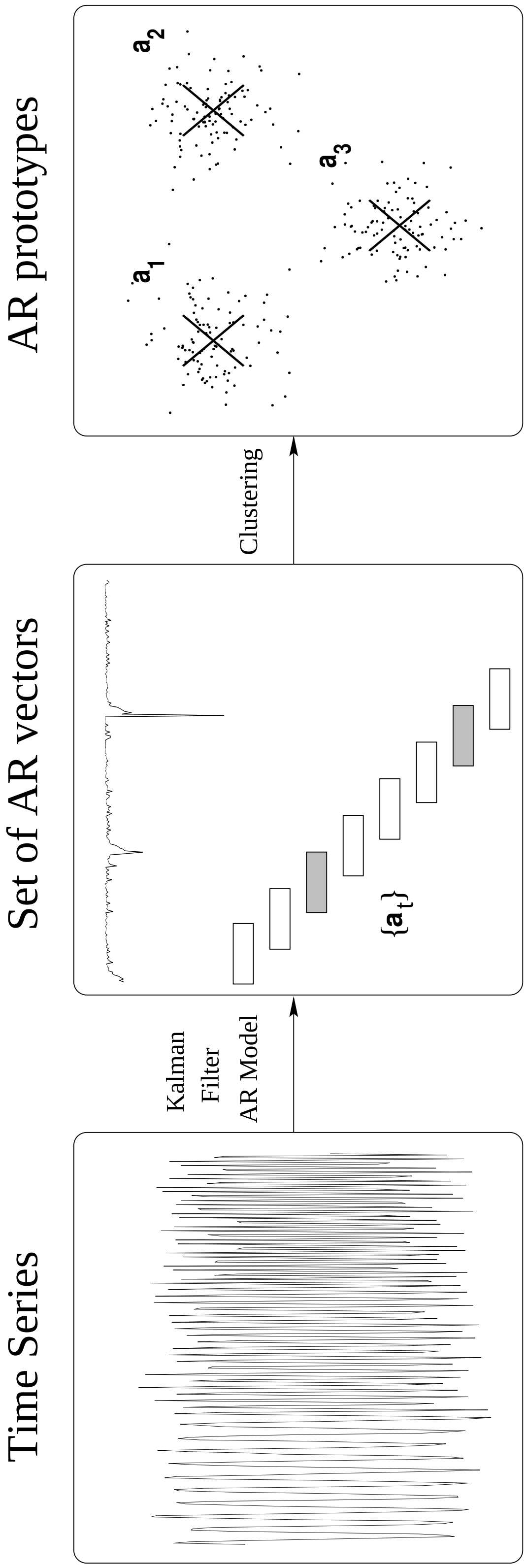


Figure 1: Initialisation of AR coefficients in the HMM-AR model. A Kalman Filter AR model is passed over the original time series to obtain a set of AR vectors at each time step, $\{\boldsymbol{a}_t\}$. If the evidence (the top plot in the second box) of a prediction is low, the corresponding AR vector is removed from this data set (these vectors are shaded). Clustering is then performed using a Gaussian mixture model which identifies a number of prototype AR-vectors which are then used to initialise the HMM-AR model.

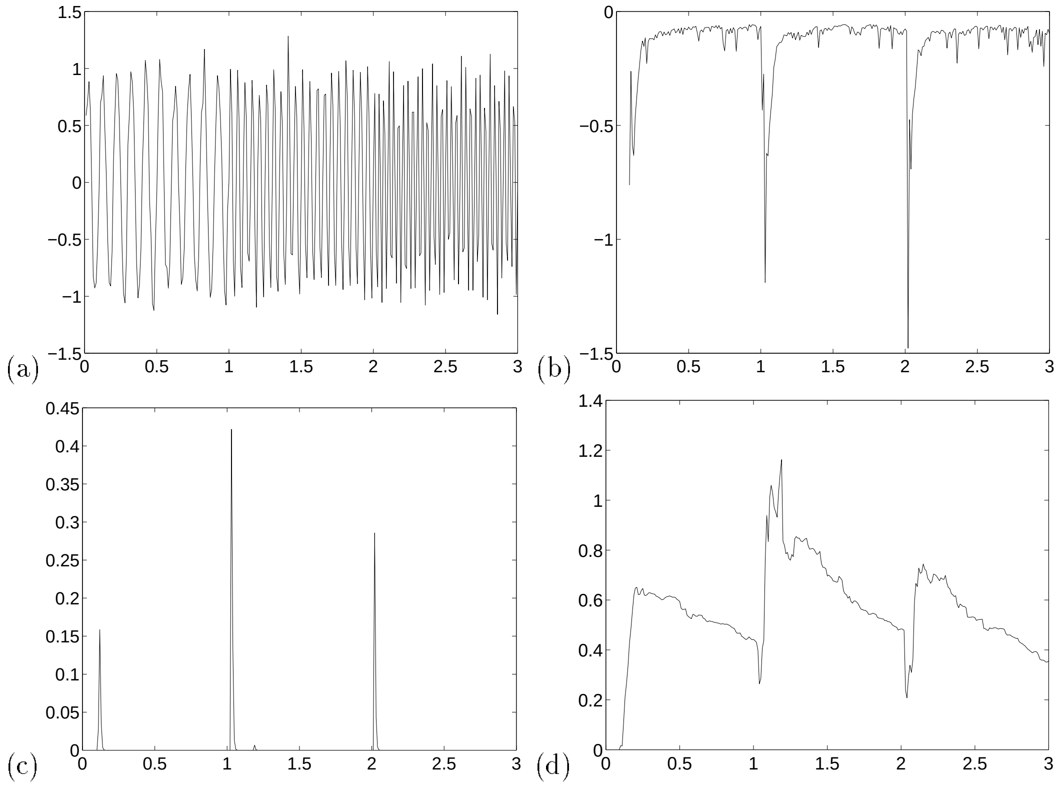
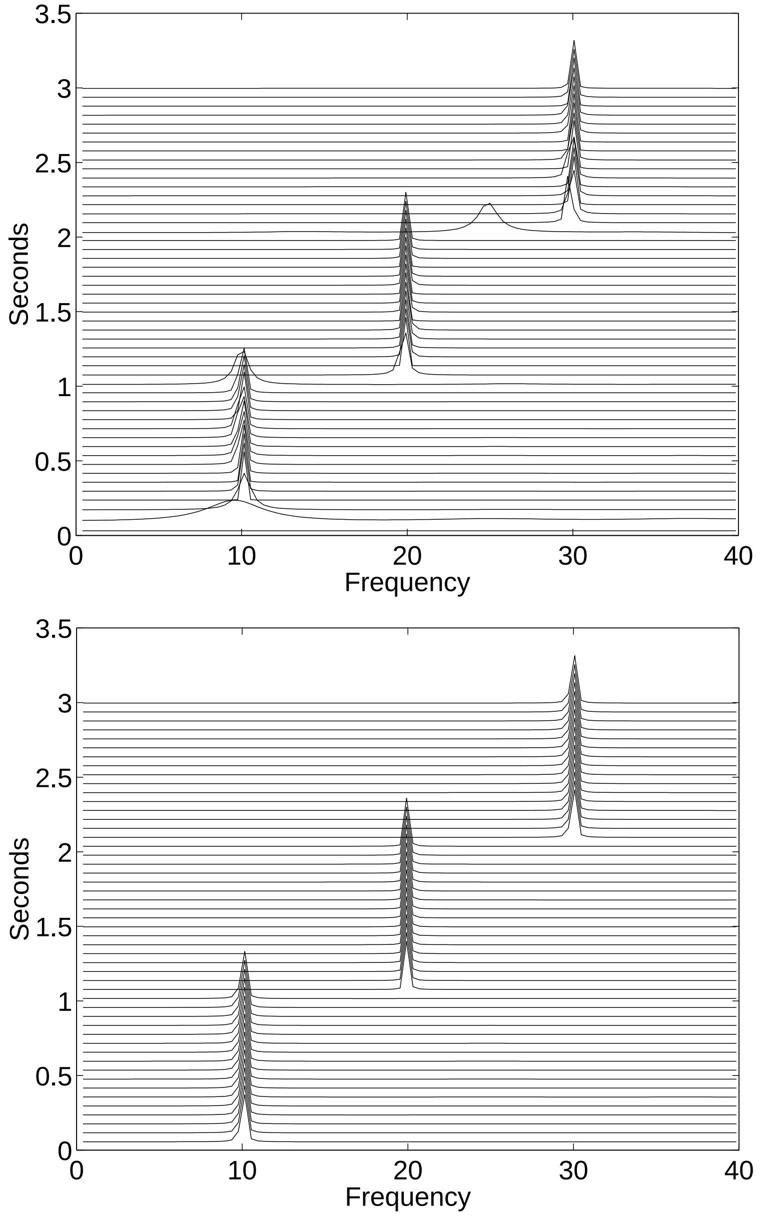


Figure 2: Example 1: (a) Original time series (b) Log evidence of data points under Kalman filter AR model, $\log_{10}(p(y_t))$ (see Eq. 12), (c) Estimate of state noise parameter and (d) effective learning rate, from Eq. 14. The Kalman filter responds to a new dynamic regime by abruptly increasing the estimate of the state noise. This has the effect of increasing the effective learning rate, so that the model adapts to the new data. As the model is then exposed to more data from the same stationary regime, the learning rate reduces in proportion to the reducing uncertainty in model parameters.



(a)

(b)

Figure 3: Example 1: Spectrograms from (a) Kalman filter AR model (b) Hidden Markov AR model.

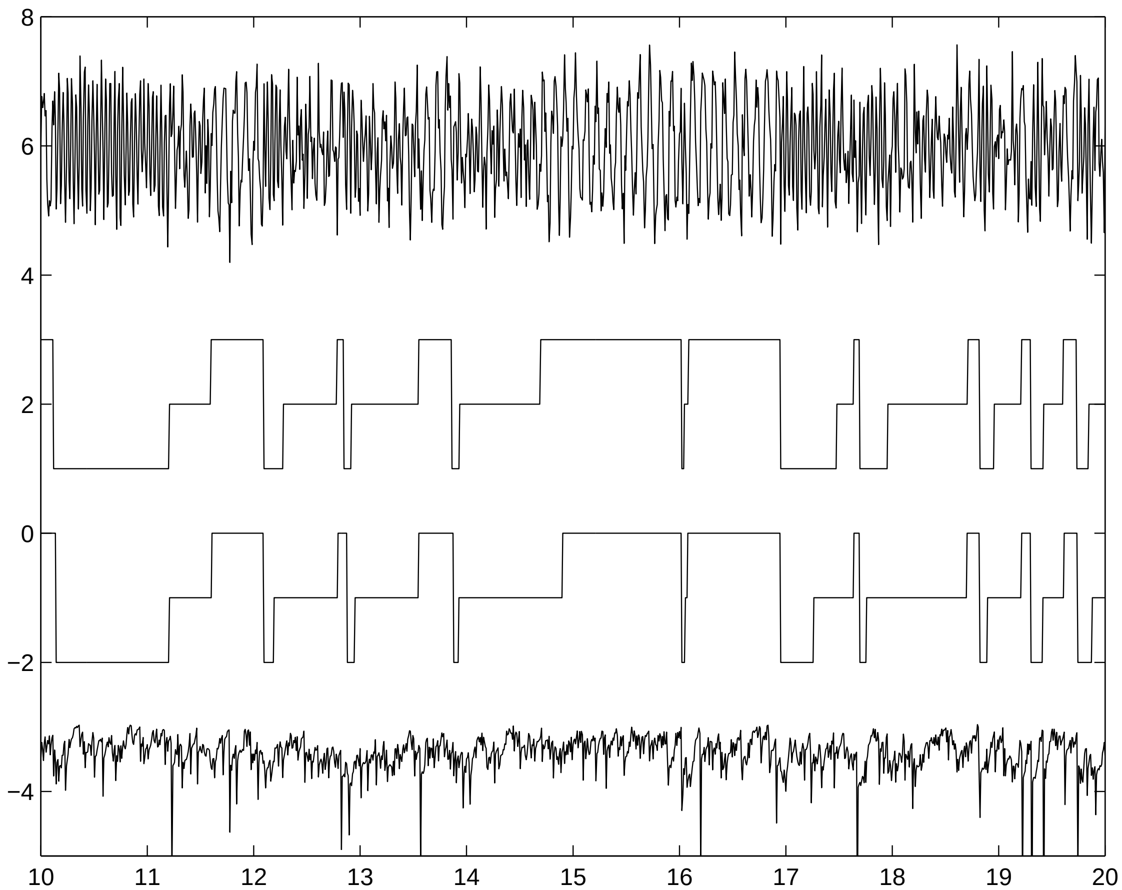


Figure 4: Example 2: (Row 1) Ten seconds of data generated from 3-state sinusoidal HMM, (Row 2) true state, (Row 3) state as estimated by HMM-AR model, (Row 4) log evidence of Kalman filter AR model.

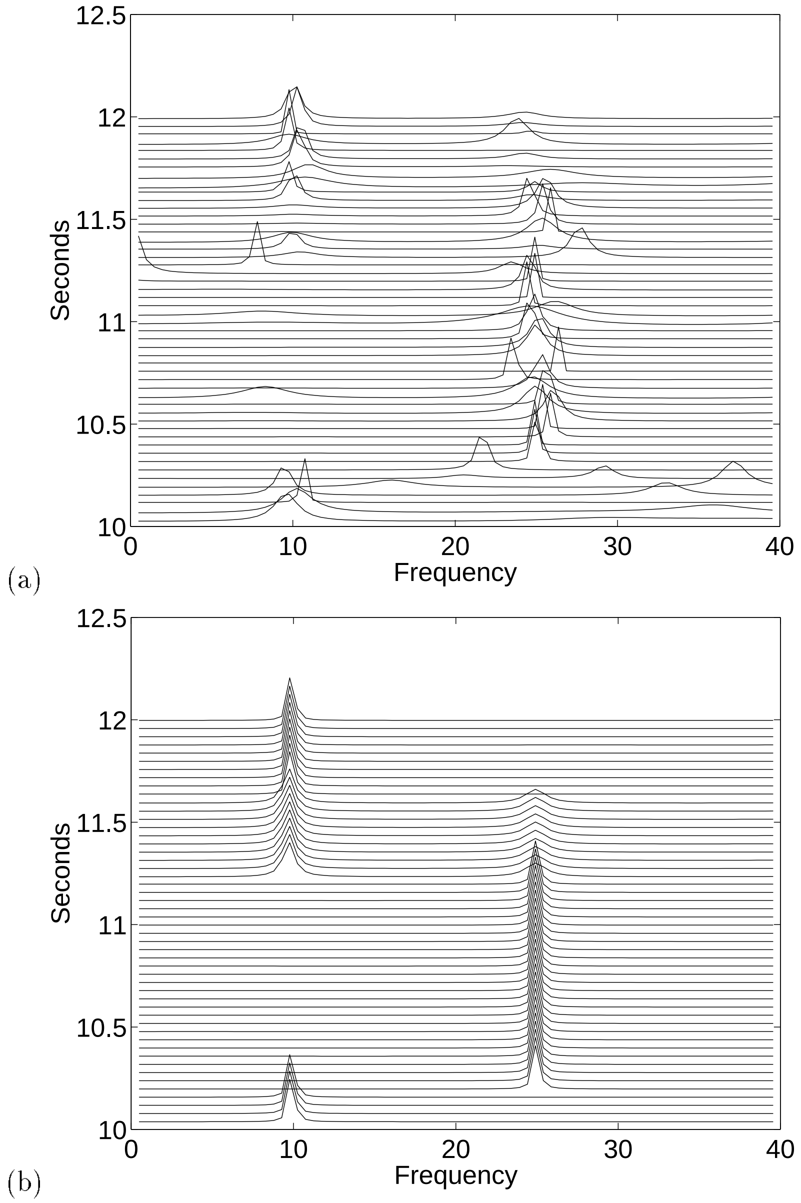


Figure 5: Example 2: Spectrograms from (a) Kalman filter AR model (b) Hidden Markov AR model for seconds 10 to 12.

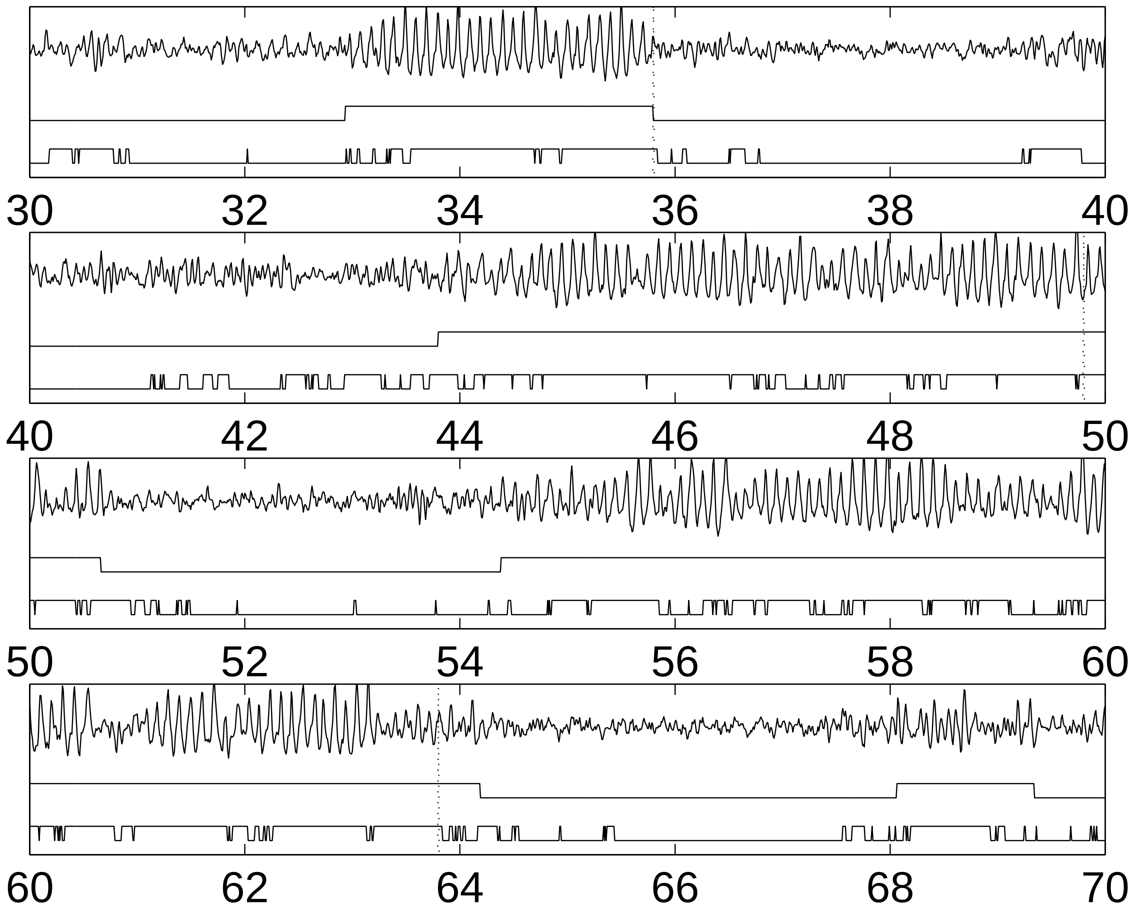
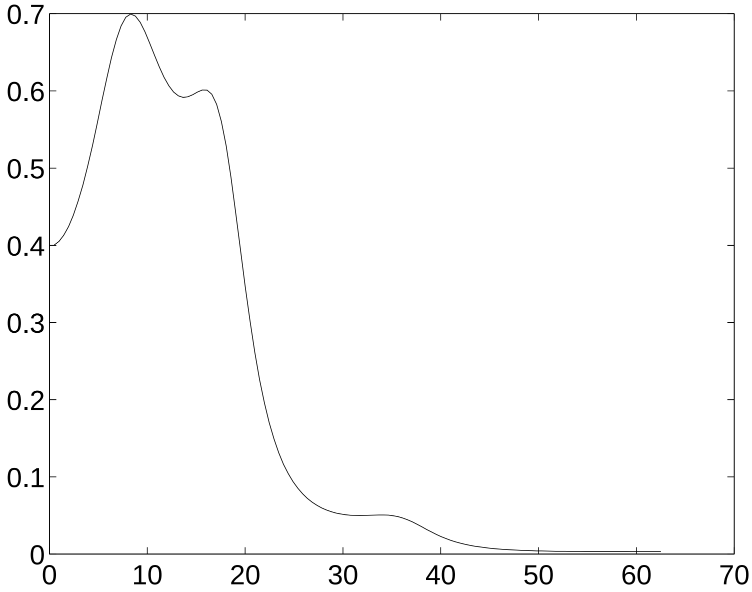
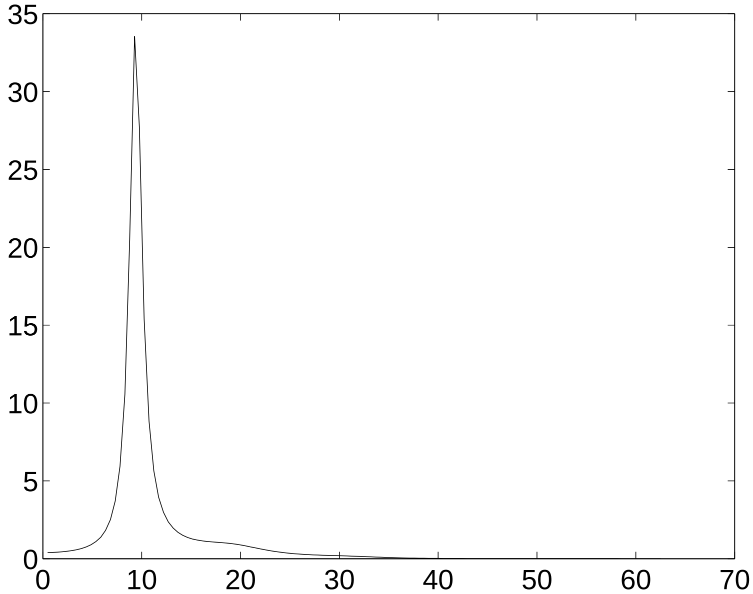


Figure 6: Example 3: (Row 1) Forty seconds of EEG data from a cued hand movement experiment. The mu-rhythm, a 10Hz synchronization in the sensorimotor cortex, can be detected during motor relaxation. When the subject moves their hand (after the cue), the mu-rhythm disappears. The dashed vertical lines, preceding seconds 36, 50 and 64 indicate the movement cues, (Row 2) state as estimated by HMM-AR model, (Row 3) state as estimated by a Gaussian Mixture Model trained on Kalman filter coefficients.

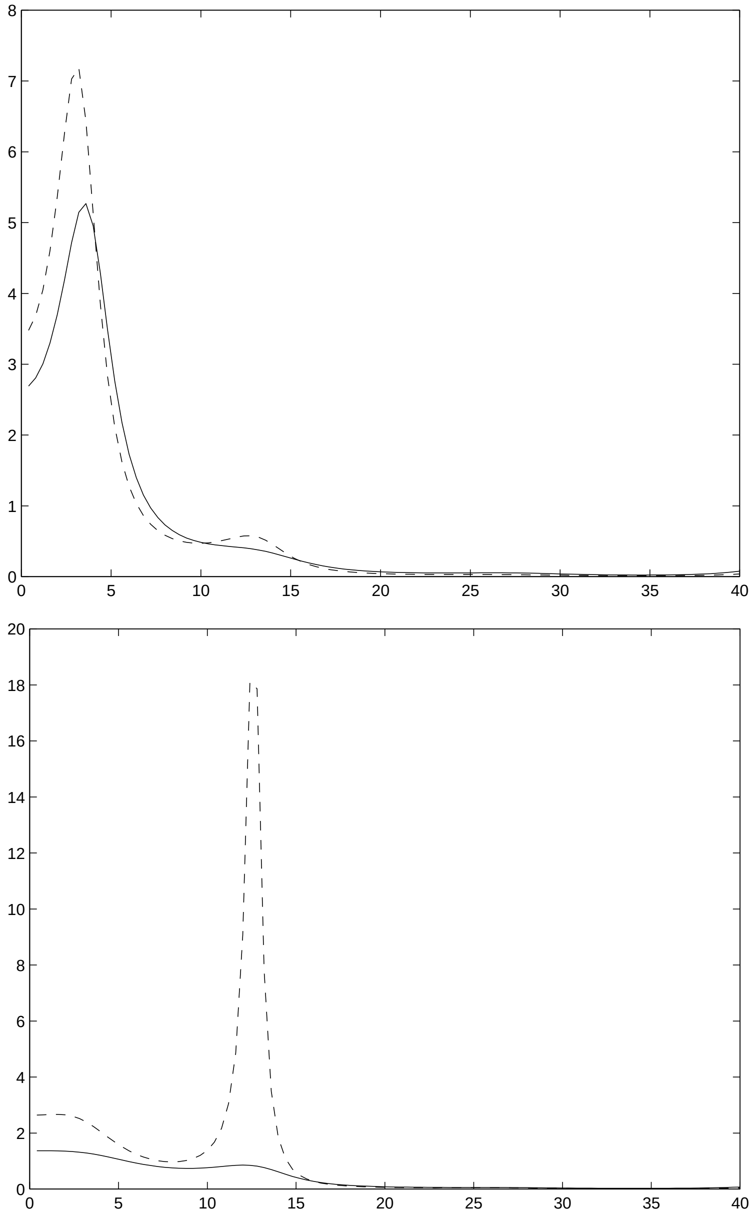


(a)



(b)

Figure 7: Example 3: Spectra associated with (a) state 1 ('hand movement') and (b) state 2 ('no hand movement') of an HMM-AR model trained on the EEG data.



(a)

(b)

Figure 8: Example 4: (a) state 1 and (b) state 2 spectra used to initialise HMM-AR model. The dashed lines are for supervised initialisation (using the expert's classification) and the solid lines are for unsupervised initialisation. Both initialisation schemes led to the same HMM-AR solution (see Figure 9).

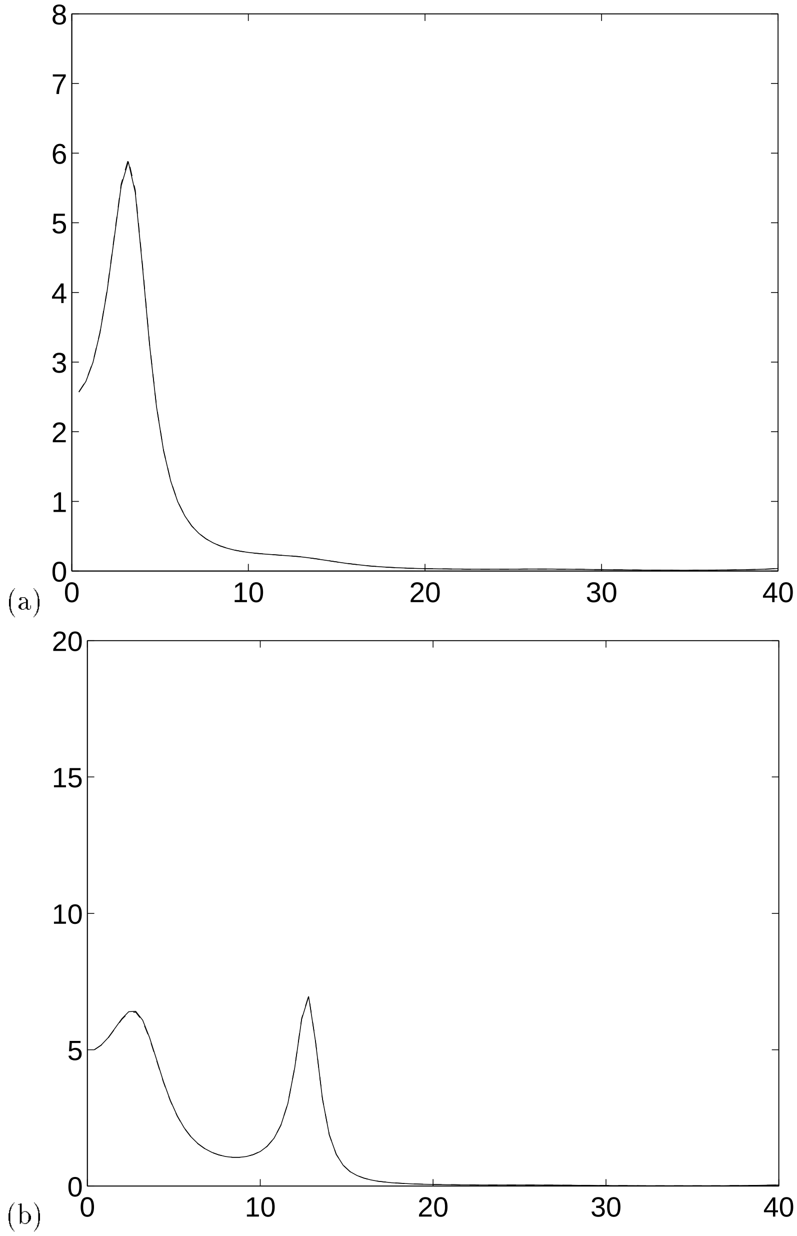


Figure 9: Example 4: (a) state 1 and (b) state 2 spectra after full EM training of the HMM-AR model. Both initialisation schemes led to this solution.

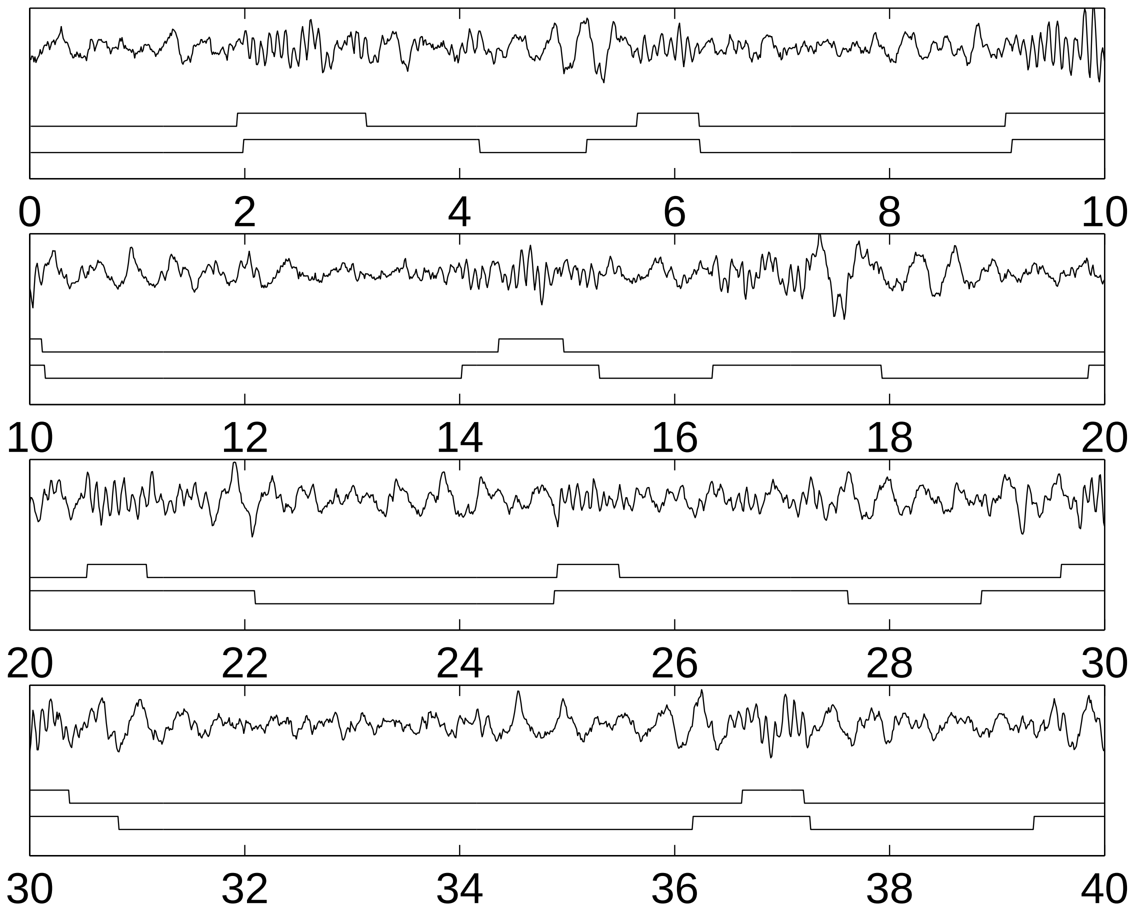


Figure 10: Example 4: (Row 1) Forty seconds of EEG data from a sleep experiment, (Row 2) state estimation by expert, (Row 3) state as estimated by HMM-AR model.